龙芯中科
LOONGSON TECHNOLOGY

JPEG 模块 说明

2023 年 3 月 10 日

龙芯中科技术有限公司

修订历史

序号	更新日期	版本号	更新内容
1	2021-5-9	V0.1	文档框架

目　　录

目录 ..... **i**

第一章　JPEG 编码器 **1**

1.1　需求 1

1.2　原理 1

1.2.1　JPEG 文件结构 1

1.2.2　JPEG 编码生成流程 1

1.3　实现 5

1.3.1　软件部分 5

1.3.2　硬件部分 6

此页留空

表 目 录

1.1 JPEG 编码器的需求 ····· 1
1.2 常见 JPEG 文件的组成 ····· 1
1.3 位宽与幅值 ····· 5
1.4 32 位 RGB ····· 6
1.5 采样软硬件配置 ····· 6

此页留空

图 目 录

1.1 灰度 JPEG 编码生成过程 2
1.2 采样示意图 3

此页留空

第一章　JPEG 编码器

1.1　需求

JPEG 编码模块为扫描，复印等功能提供支持。在参考了行业需求方提供的软件接口和文档后，将 JPEG 编码器的分别从软件接口和硬件支持角度需求归纳如表1.1所示：

表 1.1: JPEG 编码器的需求

软件接口	硬件支持
到内存读写接口 (DMA)	RGB 到 YCbCr 转换
配置寄存器接口	离散余弦变换 (DCT)
连续编码	量子化, 可配量化表
产生/清除中断	哈夫曼编码
软件复位	多种采样类型

1.2　原理

1.2.1　JPEG 文件结构

通常，以.jpeg 或.jpg 为后缀名的图像文件由按表1.2顺序出现的部分组成，有些组件可能会不止一次出现，如 DQT，DHT 等。

表 1.2: 常见 JPEG 文件的组成

序号	名称	字段标记	释义
0	SOI	0xffd8	Start of Image / 图像开始
1	APP	0xffe0	Application / 应用程序保留标记
2	DQT	0xffdb	Define Quantization Table / 定义量化表
3	SOF	0xffc0	Start of Frame / 帧图像开始
4	DHT	0xffc4	Define Huffman Table / 定义哈夫曼表
5	SOS	0xffda	Start of Scan / 扫描开始
6	EOI	0xffd9	End of Image / 图像结束

1.2.2　JPEG 编码生成流程

JPEG 编码技术在上世纪 90 年代开始流行，其目的在于对 RGB 格式大量的数据进行可不同程度恢复的压缩。一般意义上，JPEG 编码指从 SOS 到 EOI 之间的表示图像的编码部分。一个用 RGB 文件生成灰度图片 JPEG 编码的过程如图1.1所示：

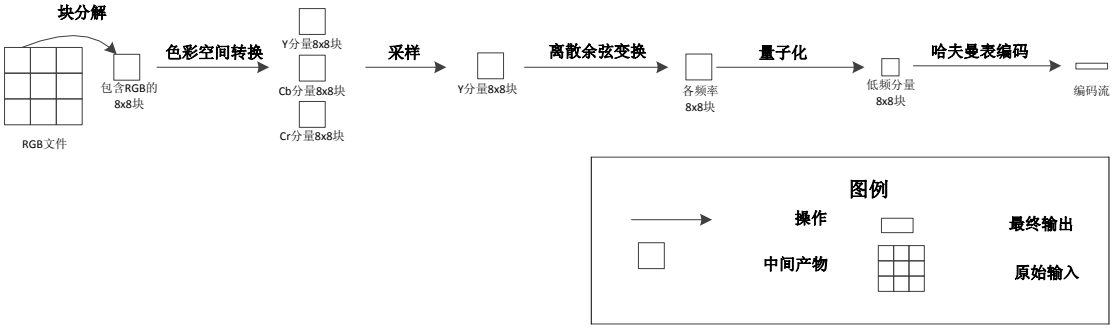


图 1.1: 灰度 JPEG 编码生成过程

1.2.2.1 块分解

包含 RGB 的原始文件可以按照行或列生成与存储，但是在转换之前必须要按照每行 8 个像素，总共 8 列，先行后列的方式进行处理。如果文件的行列像素值不能被 8 所整除，则应将最后的行或列复制 8 -（像素数 mod 8）次，以保证像素信息在转换时不至于丢失。

1.2.2.2 色彩空间转换

将 RGB 转为 Y(亮度)，Cb(蓝色度)，Cr(红色度) 的公式如下：

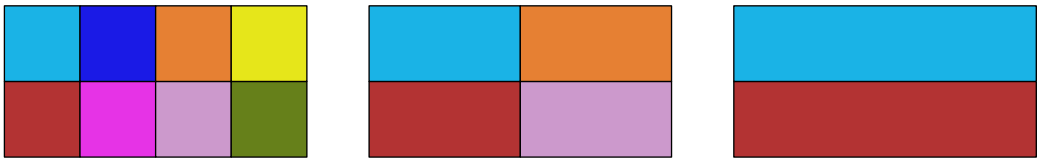
$$Y = 0.299 * R + 0.587G - 0.114B$$

$$Cb = -0.1687R - 0.03313G + 0.5B + 128$$

$$Cr = 0.5R - 0.4187G - 0.0813B + 128$$

1.2.2.3 采样

由于人眼对亮度的感知远高于色度，因此将亮度色度分离之后，在最大限度保留亮度的同时对色度进行采样，可以达成对质量影响不大而文件所占空间变小的目的。如果完全丢弃色度信息，则会生成灰度 (grayscale) 图像。本设计实现了四种采样方式：4:4:4，4:2:2，4:1:1，4:0:0。4:a:b 的含义如图1.2所示，对于每 2 行 4 列像素，亮度信息取全部提取，第一行取 a 个色度信息，第二行取 b 个色度信息。



(a) 4:4:4 (b) 4:2:2 (c) 4:1:1

图 1.2: 采样示意图

1.2.2.4 离散余弦变换

离散余弦变换是一种傅立叶变换的形式，将离散的数据表示为有限的的各频率余弦函数之和。输入的 8x8 矩阵，每个单元表示色彩信息，坐标对应其位置信息；转化后的 8x8 矩阵，每个单元表示其频率分量，坐标对应频率。公式如下：

$$F(u,v) = \frac{1}{4}C(u)C(v)[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y) * cos\frac{(2x+1)u\pi}{16} cos\frac{(2x+1)v\pi}{16}]$$

where

$$C(u), C(v) = 1/\sqrt{2}\ for\ u, v = 0;$$

$$C(u), C(v) = 1\ otherwise.$$

1.2.2.5 量子化

根据人眼对不同分量感知不同，将较高频分量滤除，保留较低的频率的过程。将离散余弦变换产生的 8x8 矩阵的每个元素，除以量化表对应位置的元素，再将结果舍入为整数，就会得到量化后的 8x8 矩阵。有损编码机制产生信息损失主要在此步骤发生。如果保留所有分量可以使用全 1 量化表，此时接近于无损。

1.2.2.6 哈夫曼编码

哈夫曼编码过程可以分为 3 个步骤

1. 矩阵元素按以下矩阵所示的曲折 (zig-zag) 顺序读入。

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

2. 矩阵的第一行第一列为 DC 分量值，当前 DC 分量值减去上一个，首个 DC 分量值则减去 0，得到的差进行可变长度编码（variable-length code），再转换为哈夫曼编码。例如，$\Delta DC = 3$，先进行 VLC，用（size, amplitude）的格式表示为（2,3）。其位数（size）为 2，将位数对应直流范式哈夫曼表 (DC Canonical Huffman Table) 为三位二进制数 011，幅度 (amplitude) 为 3，二进制为 11，所以此部分将会生成码流 01111。位数幅度转换关系见表1.3。
3. 矩阵的剩余 63 个元素为 AC 分量值，先判断该值是否为 0，是 0 则做计数累加，非 0 则将此前累加的 0 的个数 (run-length) 和该值所占的位数 (size) 相组合，根据交流范式哈夫曼表 (AC Canonical Huffman Table) 将其转换为编码；再 AC 分量值转写成对应位数行所在的位置 (amplitude) 输出。需要注意的是，run-length 只有 4 位，当需要表示连续 16 个零则应转化作（15,0）。另外，如果从某个位置开始，直到最后一个值也没有非 0 值可供转换输出，则直接输出 EOB 组合（End of Block）。

   以下列矩阵为例：

$$\begin{matrix} 15 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

   第一个单元 15 为直流分量，按照曲折的读取顺序，第 2 个值为 0, 第 3 个值为 -2，第 4 个值为 -1，第 5 个值为 -1, 第 6 个值为 -1, 第 7 个值为 0, 第 8 个值为 0, 第 9 个值为 -1, 剩下值全部为 0。用 (run-length, size)(value) 表示结果为:

   (1,2)(-2), (0,1)(-1), (0,1)(-1), (0,1)(-1), (2,1)(-1) (EOB)

   将 (run-length, size) 对照交流范式哈夫曼表：

$$(0,1)\ 00$$

$$(1,2)\ 11011$$

$$(2,1)\ 11100$$

$$(EOB)\ 1010$$

   DC 和 AC 的幅度转换都根据表1.3的规则进行。幅度值不能有负数，所以其对应

该行所在的位置，所占位数就是 size（从 0 开始计数）。

3 11

−2 01

−1 0

结合上一步的 DC 码值，则此 8x8 转换后的结果为：

01111110110100000000001110001010

表 1.3: 位宽与幅值

SIZE	AMPLITUDE
1	-1, 1
2	-3, -2, 2, 3
3	-7..-4, 4..7
4	-15..-8, 8, 15
5	-31..-16, 16..31
6	-63..-32, 32..63
7	-127..-64, 64..127
8	-255..-128, 128..255
9	-511..-256, 256..511
10	-1023..-512, 512..1023

1.3 实现

1.3.1 软件部分

扫描应用场景中，输入的结合需求得出软件编程一般场景：

1. 将量化表和 RGB 原始文件按块准备就绪，并把二者在内存的地址配入相应寄存器。
2. 配置生成码存放在内存位置的寄存器。
3. 配置采样模式，可从 4:4:4，4:2:2，4:1:1，灰度四种选取其一。
4. 配置 RGB 文件包含的块 (block) 数。
5. 配置是否需要接续上一次编码，是则配置上次编码的最后字节及包含位数。
6. 清除状态位或中断标志。
7. 开始编码。
8. （可选）发出软复位，停止编码。
9. 等待完成中断或状态位。
10. 从字节计数器中读出生成总字节数。

11. 将生成的编码放在 JPEG 文件 SOS 到 EOI 之间，再与表1.2其他部分组合成 JPEG 文件。

1.3.2 硬件部分

1.3.2.1 AXI-DMA

1. R-DMA
   每次按顺序取 64 个 32 位字，对应一个 8x8 块对应的 RGB 数据，因此，软件必须在开始之前按照顺序将一个 RGB 原始文件的顺序按块整理，字内部排序见表1.4。另外，要求起始地址 16 字节对齐，规避 AXI 规范要求的 4K 边界问题。

表 1.4: 32 位 RGB

Byte3	Byte2	Byte1	Byte0
31:24	23:16	15:8	7:0
Reserved	R	G	B

2. W-DMA
   哈夫曼模块生成的编码为 32 位字，完成一个编码单元（MCU）之后，会发出信号要求 W-DMA 向内存输出编码。传输时，先查看将要写的地址是否 16 字节对齐，如果是则直接将待传输数据及长度写入；如果不是按 16 字节对齐，查看总传输量能否大于使其对齐的量，如果可以，则先传输使其对齐的字节，不可以则传输所需传输数目字节。

1.3.2.2 色彩空间转换单元

将1.2.2.2的公式由浮点数变为整数进行操作。

1.3.2.3 采样单元

采样单元的配置需要软件和硬件相互配合才能正常生成 JPEG 文件。四种模式软硬件配合对应表1.5。

表 1.5: 采样软硬件配置

采样方式	SOF.color_info	SOF.color	SOS.color_type
4:4:4	0x11 0x11 0x11	3	3
4:2:2	0x12 0x11 0x11	3	3
4:1:1	0x14 0x11 0x11	3	3
灰度	0x11	1	1

1.3.2.4 离散余弦变换单元

将1.2.2.4的公式由浮点数变为整数进行操作。

1.3.2.5 量化单元

通常，量化表可分为亮度和色度两种，所以此处量化单元按照这样分类具备两种。需要软件配合预处理，将 4096 除以量化值的商取 13 位整数按照 32 位对齐连续地放入内存中，起始地址 16 字节对齐。内部的实现方式为提前将数据向右移动 12 位，然后乘以 4096/Q，如果按 1 量化，则又乘以 4096，此时量化造成数据损失最小。化除为乘以降低时间和面积消耗。

1.3.2.6 哈夫曼编码单元

1. 将取得 64 个量化后数据由线性排列换为曲折排列（zig-zag）。
2. 将 DC 数据转换为 (size, amplitude) 形式，AC 数据转换为 (size,run-length)(amplitude) 形式。
3. 查表将 DC 和 AC 数据转换为哈夫曼编码
4. 如果为接续编码，则将上一次剩余编码和本次生成一个块的哈夫曼编码流收集，按 32 位输出；没有剩余结果或者非接续编码，则只收集本次生成码流。
5. 如果有余下数据，则保留作为下一次继续编码的输入。