

# JPEG 软件解码及其在 ARM7 上的实现

廖小情, 陈小平

(苏州大学 电子信息学院, 江苏 苏州 215006)

**摘要:** 介绍了一种用 ARM7 实现 JPEG 控制解码的方案。采用 LPC2478 软件解码 JPEG 图像并将图像实时显示在具有 LVDS 接口的 TFT\_LCD 显示器上。在硬件设计中, 解决了 LVDS 信号与 LPC2478 的 LCD 控制器输出信号的转换问题, 并对解码及显示所需内存大小作了详细的分析; 针对 JPEG 解码中的 Huffman 解码、IDCT 变换等步骤提出了优化方法, 使解码速度在一定程度上有了较大的提升, 实现了在主频不很高的 LPC2478 上解码并显示 JPEG 图像。

**关键词:** LPC2478; LVDS; JPEG; 软件解码; 算法优化

中图分类号: TP751

文献标识码: A

文章编号: 1674-7720(2011)09-0039-03

## JPEG software decoding and the implementation on ARM7

Liao Xiaoqing, Chen Xiaoping

(Electronic Information Institute, Soochow University, Suzhou 215006, China)

**Abstract:** This paper introduced a scheme to achieve JPEG decoding controlled by ARM7, using LPC2478 to implement the decoding JPEG images by software and display images on the TFT\_LCD with LVDS interface. In the hardware design, the signal conversion problem was resolved between the LVDS signal and the output signal of LCD controller of LPC2478, and a detail analysis about the decoded and display's memory size was made. As to JPEG decoding, optimization methods for Huffman decoding and IDCT were proposed. The decoding speed was promoted in a certain extent. It realized decoding and displaying JPEG images on LPC2478 when the work frequency is not very high.

**Key words:** LPC2478; LVDS; JPEG; software decoding; algorithm optimization

JPEG 作为一种成熟的图像压缩标准, 具有压缩效率高、恢复的图像质量好、易于实现等优点<sup>[1]</sup>, 已经得到了广泛的运用。随着现代嵌入式技术的快速发展, 各类电子产品为了进一步降低成本, JPEG 技术也越来越多地受到各种小型嵌入式产品的青睐。目前嵌入式系统方面对 JPEG 图像软件解码较多地应用在 DSP 和多核等高速处理器上<sup>[2-3]</sup>。

本文提出一种基于 ARM7 处理器 LPC2478 的软件解码 JPEG 方案, 同时将图像显示在高分辨率 10.4 英寸的、具有 LVDS 接口的 TFT\_LCD 液晶显示器 AA104SG02 上。

### 1 硬件设计及实现

以 LPC2478 为控制核心的小型嵌入式系统硬件实现框图如图 1 所示。

LPC2478 是一款 NXP 公司生产的基于 ARM7 的微控

制器<sup>[4]</sup>, 拥有丰富的片上资源, 在嵌入式控制领域得到了广泛的运用。LPC2478 的内核为 32 bit 的 ARM7TDMI 处理器, CPU 工作频率最高可达 72 MHz, 片上 Flash 为 512 KB, SDRAM 为 96 KB, Flash、SRAM、SDRAM 可扩展 (SDRAM 最大可扩展到 1 024 MB), 带 SSI、USB、UART 等多种外设接口。此外, LPC2478 还带有 LCD 控制器, 支持 TFT 及 STN 显示模式。

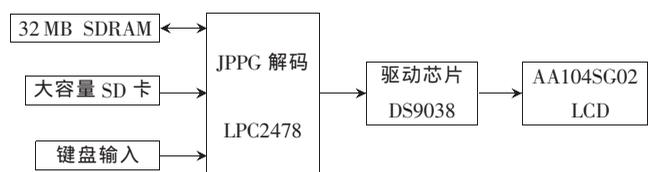


图1 系统硬件设计框图

AA104SG02 为三菱公司推出的 10.4 型 SVGA TFT\_LCD 显示屏<sup>[5]</sup>, 分辨率为 800×600, 亮度为 400, 对比度

为 500:1, 输入电压为 3.3 V, 标准工作频率为 40 MHz, 支持多种数据格式, 电气接口为 LVDS 信号。LVDS 信号是一种能够满足大数据传送需求的差分信号技术, 具有功耗低、可靠性高及高速传送等特点。

在 LPC2478 控制 LCD 显示模式下, LCD 显示数据的位宽度为 16 bit, 采用数据格式为 R:G:B=5:6:5。LCD 控制器输出的信号有 RGB 数字信号、像素点时钟 (PCLK)、行同步时钟 (HSYNC)、场同步时钟 (VSYNC) 及其他控制信号, 这些信号均以并行方式输出, 因此, LPC2478 的 LCD 控制器的输出并行信号与 AA104SG02 的 LVDS 信号不能直接连接, 必须在两者之间加驱动电路, 将并行信号转换为 LVDS 差分信号。驱动芯片 DS90383 是一款信号传输转换芯片, 可将接收到的 TTL/CMOS 数据转换为 LVDS 数据流, 支持 VGA、SVGA 等更高分辨率, 输入时钟为 20 MHz~60 MHz, 3.3 V 单电源供电。因此, 利用 DS90383 可实现两者之间的信号转换。

在液晶控制器中, 需要为显示数据提供相应大小的缓存。屏幕为 800×600 像素, 每个像素占 2 B, 则需要 960 000 B 的 SDRAM 作为显示缓存。此外, 在 JPEG 解码过程中, 需要缓存原始 JPEG 图像数据、JPEG 解码中间数据等, 对于一幅 800×600 的 JPEG 图片来说, 解码需要 2 MB 左右的缓存空间。而 LPC2478 的片上 SDRAM 远远不能满足空间要求, 因此, 本系统外扩了一个 32 MB 的 SDRAM。与此同时, 考虑到在实际应用中需要更换不同的显示图片, 但又不可能把很多幅图片都存储在芯片内, 为此, 还外接了一个大容量 SD 卡。当需要显示某个图片时, LPC2478 可从 SD 卡中读取图片, 解码后显示在 LCD 上。

## 2 JPEG 软件解码算法优化及实现

JPEG 解码技术主要包括获取图像基本信息、建立解码码表、Huffman 解码、反量化、反 Zig-Zag、IDCT 及颜色空间转换。JPEG 解码流程图如图 2 所示。



图 2 JPEG 解码流程图

### 2.1 图像基本信息获取

在 JPEG 图像解码之前, 需要通过 JPEG 文件获取图像解码时所需的各种信息, 包括图像的大小、采样因子、量化表以及 Huffman 表等。由于 JPEG 文件的存储格式为 JFIF 格式(主要为标记码和压缩数据), 因此, 可以根据不同的标记码获得相应的信息。在扫描开始(SOS)标记段结束后才是真正的图像压缩数据流。

### 2.2 解码码表建立

在 Huffman 解码前, 要建立解码码表。对于一幅图像, 一般有 4 个 Huffman 表(直流 0 号表、直流 1 号表、交流 0 号表、交流 1 号表), 每个码表都建立各自的三张解码码

表, 分别为: 不同位数的码字的数量表(Code\_bitsnum\_table)、码权值表(Code\_weight\_table)、码字表(Code\_value\_table)。不同位数码字的数量表和码权值表可直接从 Huffman 表中获得。而码字表则是通过 Huffman 表建立 Huffman 树获得, 且码字均自动按从小到大排列, 而且没必要保存每个码字的长度(以下简称码长)。因为每个码长可由不同位数码字的数量表与码字表推算而得。建立码字表的流程图如图 3 所示。

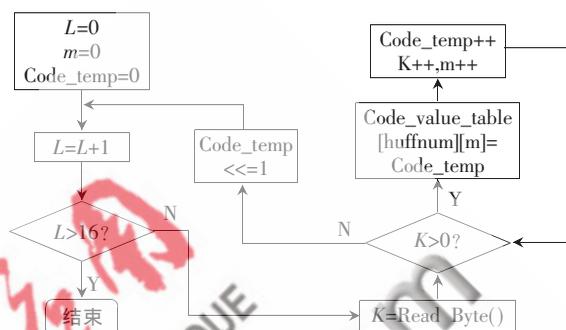


图 3 码字表流程图

图中,  $L$  表示码字的位数(最长为 16 bit);  $huffman$  为 Huffman 表的序号;  $Read\_Byte()$  为读取 Huffman 表的值, 表示对应  $L$  位码字的个数, 如果等于 0, 表示没有  $L$  位的码字。第一个码字值必定为 0, 若有  $K$  个码字数相同, 则当前码字为它前面的码字加 1; 但如果它的位数比它前面的码字数大, 则当前码字是前面码字加 1 后再在后面添若干个 0, 直到满足位数长度位置为止。

### 2.3 Huffman 解码优化

在 JPEG 文件中, 图像压缩数据流是逐位存储的, 因此解码时也必须逐位地读入。Huffman 解码的过程主要是解码码表的查找过程。简单的解码方法为直接查找解码码表直到找到正确的 Huffman 码, 此时, 逐位移入码流获得的编码值都需要从解码码表的头开始查找, 比较码字是否相同、码长是否相等。这将需要大量的存取操作, 不仅会增加代码量, 同时还将耗费较多的 CPU 时间。

为降低代码开销及尽可能减少 CPU 时间, 本文对该解码过程进行了优化。获得编码值及编码值长度后, 根据不同位数码字的数量表可直接定位到码字表中第一个码长相等的码字, 若编码值不小于该码字且不大于相同码长的码字的最大值, 则可以找到与编码值相同的码字; 否则不能找到, 须重新获得新的编码值后再按上述步骤查找。根据查找出的相同的码字在码字表中的位置, 由码权值表获得该码字的权值。当这个码字为直流码字时, 权值的大小即为直流分量数值的二进制位数, 再根据 Huffman 直流译码表译码; 当这个码字为交流码字时, 权值的高 4 位表示当前值前面有多少个连续的 0, 低 4 位为该交流分量数值的二进制位数, 再根据 Huffman 交流译码表译码。优化后的一次 Huffman 解码流

程图如图 4 所示。

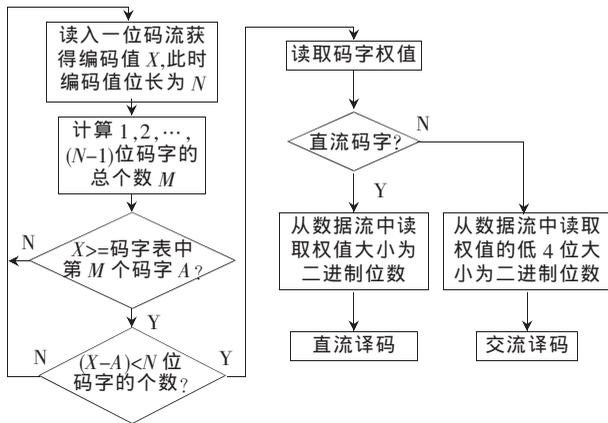


图 4 优化后的 Huffman 解码流程图

在 JPEG 文件中存储了两张量化表，分别为亮度量化表和色度量化表。反量化运算过程即为经 Huffman 解码得到的系数矩阵与相应的量化矩阵相乘。由于数据是按  $8 \times 8$  矩阵的“之”字行排列，所以要对反量化运算的结果进行反 Zig-Zag 操作。

在解码  $8 \times 8$  的数据块过程中发现，右下角的高频分量存在大量的 0。如果完成 Huffman 解码之后，再进行反量化及反 Zig-Zag 排序，必然存在许多乘 0 运算及重复的数据存取操作，而且同时必须至少开辟两块  $8 \times 8$  大小的 SDRAM 来存储这些数据，这样不仅浪费了时间也浪费了空间，不利于在小型嵌入式系统上的应用。因此，采取 Huffman 解码、反量化及反 Zig-Zag 三步合一的优化措施，Huffman 解码出一个值后立即进行反量化、反 Zig-Zag 操作。但为了对应原来的存储顺序，需要建立一个对应存储位置的数组：

```
Place_Order[64]={
    0, 1, 8, 16, 9, 2, 3, 10,
    17, 24, 32, 25, 18, 11, 4, 5,
    12, 19, 26, 33, 40, 48, 41, 34,
    27, 20, 13, 6, 7, 14, 21, 28,
    35, 42, 49, 56, 57, 50, 43, 36,
    29, 22, 15, 23, 30, 37, 44, 51,
    58, 59, 52, 45, 38, 31, 39, 46,
    53, 60, 61, 54, 47, 55, 62, 63};
```

#### 2.4 IDCT 及浮点转整点运算

在图形解码运算中，IDCT 变换是主要运算，占用大量的 CPU 时间。若直接进行 IDCT 变换计算，一次二维 IDCT 变换需要 1 024 次乘法和 896 次加法，这样，在 ARM7 上解码  $800 \times 600$  大小图像会相当慢，因此要提高解码效率，必须利用 IDCT 的快速算法。为此，本文采用了参考文献[6] ARAZ 等人提出的 IDCT 快速算法。该算法的主要思想是：

(1) 将二维的 IDCT 分解为两个一维的 IDCT 降维处理，先后在行方向、列方向上做 IDCT 变换。

(2) IDCT 通过数学变换转化为离散傅里叶逆变换 (IDFT)，然后利用矩阵变换简化计算。

(3) 减少乘法和加法运算次数。其结果是完成一次 2D-IDFT 仅需要 10 次乘法和 29 次加法。

原 IDCT 变换运算过程包含了 5 步计算，而本系统将其简化为 3 步，最大限度地减少数据存取操作次数。三角函数及一些固定的系数采用查表方式，除法运算的除数均为 2 的  $n$  次幂，可直接转换为右移操作，以减少计算量。但乘法和加法运算均为有符号浮点运算，而 ARM7 只支持整点运算，因此还需要将浮点运算转化为整点运算。

在 IDCT 运算中，大部分的浮点乘法运算为一个数乘以一个余弦值（即乘数小于 1），根据这个特点，本文设计了如下浮点转整点运算方法：

(1) 将浮点数乘以 256，得到的 32 位整点数格式为  $0xXXXXXX\ XX$ ，高 24 位保留了浮点的整数部分及符号位，低 8 位保留到了小数部分的第 4 位。

(2) 乘法运算：被乘数转化为如上格式的 32 位整数（如  $0xAAAAAABB$ ），乘数要求小于 1。转化后为 8 位整数，如  $0xCC$ 。为了防止溢出，乘法运算做如下更改：

$$0xAAAAAABB \times 0xCC = (0xAAAAAA \times 0xCC) + ((0xBB \times 0xCC) \gg 8)$$

该方法能确保整个运算过程中数据不溢出。

#### 2.5 色彩空间转换

通过解码出来的 Y(亮度)CrCb(色度)信号，要在屏幕上显示图像，还必须转化为 RGB 信号。其转换公式为：

$$R = Y + 1.402 \times Cb + 128$$

$$G = Y - 0.344 \ 14 \times Cr - 0.714 \ 14 \times Cb + 128$$

$$B = Y + 1.772 \times Cr + 128$$

计算时，乘法运算使用上述的浮点转整点运算法则，但需要作如下修改：

$$R = Y + Cb + Cb \times 0.402 + 128$$

$$G = Y - Cr \times 0.344 \ 14 - Cb \times 0.714 \ 14 + 128$$

$$B = Y + Cr + Cr \times 0.772 + 128$$

解码后的 RGB 信号直接存入显示缓存区，将实时显示在屏幕的对应像素点上，实现边解码边显示，直到整个图像全部显示在屏幕上。解码效果如图 5 所示，解码优化前耗时为 30 s，优化后耗时为 15 s，速度提高了一倍。



图 5 解码效果图

本系统实现了 JPEG 图像通过 LPC2478 解码显示在像素为 800×600 的 AA104SG02 液晶显示器上,完成了基于 LPC2478 的图像解码及显示的硬件设计,成功将 JPEG 解码算法移植到本系统上,优化了解码算法。通过将优化后的 Huffman 解码与反量化、反 Zig-Zag 相结合,减少了乘零运算及数据存取操作、降低了内存消耗;引入快速 IDCT 算法,减少了乘法和加法运算次数;将浮点运算转换为整点运算,加快了运算速度、提高了运算精度。使复杂的 JPEG 图像解码在主频并不很高的小型嵌入式产品上得到了很好的实现。本系统可应用于控制系统的液晶桌面图片显示,为人机交互提供了友好的操作界面。

#### 参考文献

- [1] 张益贞,刘滔.VisualC++实现 MPEG/JPEG 编解码技术[M].北京:人民邮电出版社,2002.

- [2] 薛永林,刘珂,李凤亭.并行处理 JPEG 算法的优化[J].电子学报,2002(2).
- [3] 董岚,李丽,张宇昂.基于 AMBA2AHB 总线多核平台的 JPEG 解码[J].电子测量与仪器学报,2009,23(2).
- [4] NXP Semiconductors. LPC24XX user manual. Rev. 04-26, August, 2009.
- [5] Mitsubishi Electric Corp. Technical specification Aa104sg01. Mar.9, 2006.
- [6] ARAI Y, AGUI T, NAKAJIMA M. A fast DCT-SQ scheme for images [J]. IECE Transactions 1990,71(11): 1095-1097.

(收稿日期:2010-12-13)

#### 作者简介:

廖小情,女,1988年生,硕士,主要研究方向:智能信号处理。

电子技术应用  
APPLICATION OF ELECTRONIC TECHNIQUE  
www.chinaAET.com