第35卷 第20期 计 算 机 工 程 2009年10月
Vol.35 No.20 Computer Engineering October 2009
·工程应用技术与实现· 文章编号：1000—3428(2009)20—0219—03 文献标识码：A 中图分类号：TP368.1

基于FSL总线的JPEG解码协处理器

李庆诚[1]，白振轩[1,2]，刘 洋[1]，胡海军[1]
(1. 南开大学信息技术科学学院，天津 300071；
2. 解放军后勤工程学院，重庆 400016)

摘 要：介绍一种面向嵌入式应用，能与Micrblaze处理器较好地异步协同工作，不需要专用双口RAM的JPEG解码协处理器。该JPEG解码器采用Verilog语言实现，能很好地处理JFIF格式的JPEG压缩文件，并且在只有少量缓冲空间的FIFO上能正确工作。验证实验结果表明，该处理器在xilinx公司的XUP Virtex II Pro开发板上，采用FSL总线与Microblaze处理器相连，工作效率较高。
关键词：FSL总线；JPEG解码器；协处理器；异步FIFO

JPEG Decoding Coprocessor Based on FSL Bus

LI Qing-cheng[1], BAI Zhen-xuan[1,2], LIU Yang[1], HU Hai-jun[1]
(1. College of Information Technical Science, Nankai University, Tianjin 300071;
2. Logistical Engineering University of PLA, Chongqing 400016)

【Abstract】This paper introduces an embedded system oriented JPEG decoding coprocessor, which doesn't need dual-port RAM and works well with Microblaze asynchronously. This decoder can decode JPEG file whose format is JFIF perfectly with few FIFO memory space by verilog. Certified in xilinx's XUP Virtex II Pro development board, it works well through connection with Microblaze by FSL bus.
【Key words】FSL bus; JPEG decoder; coprocessor; asynchronously FIFO

联合图像专家小组(Joint Photographic Experts Group, JPEG)是由国际电话与电报咨询委员会CCITT与国际标准化组织ISO联合成立的小组，负责制定静态数字图像的编码标准。JPEG算法是国际通用标准，主要用于静态图像的编码。Microblaze是xlinx公司开发的32位高性能嵌入式处理器，它以软核形式提供，培植简单而且灵活性强，在成本和性能之间达到很好的平衡，它采用RISC指令集，Harvard体系机构，很适合嵌入式开发应用[1]。

1 JPEG解码协处理器的总体设计

常用的JPEG压缩文件存储方式是JPEG文件交换格式JFIF(JPEG File Interchange Format)[2]，该种文件分为2个部分：标记和图像数据流。本文提出的JPEG解码协处理器针对JFIF格式，能处理完整的JFIF格式的文件，标记和图像数据流都不需要软件做任何预处理。

在JPEG标准中采用的色彩模式不是常见的RGB模式，而是采用*YCbCr*色彩系统[2]。其中，*Y*表示亮度；*Cb*表示色度；*Cr*表示饱和度。JPEG压缩主要利用人眼对低频的数据比高频的数据要敏感得多的现象。事实上人眼对于亮度的改变要敏感得多，即*Y*成分的数据比较重要。因此，为提高压缩比，在对图片质量要求不高的前提下，可对*Y*, *Cb*, *Cr*进行不同的采样，JPEG中常用的2种采样方式是YUV411和YUV422[3]。本文主要围绕4:1:1的采样比展开讨论。

JPEG解码器的解压缩过程如图1所示。其中，所有的解码都是基于16×16的图像块，也就是一个MCU(Minimum Coded Unit)进行的。解码器主要有文件格式解析模块、huffman解码模块、反量化模块与ZigZag模块、IDCT模块、色彩空间转换模块。

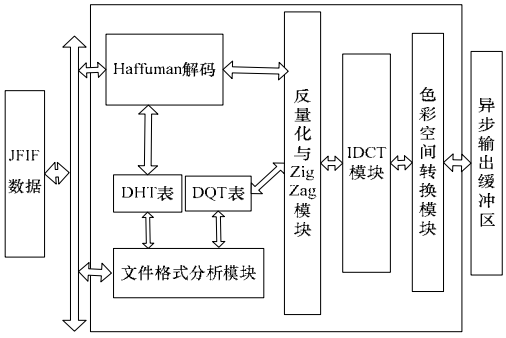

图1 JPEG解压缩过程

1.1 一维IDCT

*N*点一维IDCT的公式定义如下：

$$X(n)=\sqrt{\frac{2}{N}}\times\sum_{k=0}^{N-1}a_k\times Y(k)\times\cos(\frac{(2n+1)k\pi}{2N}) \quad (3)$$

其中，$a_0=\frac{\sqrt{2}}{2}$；$a_k=1,\ k=1,\ 2,\cdots,N-1;\ n=0,\ 1,\cdots,\ N-1$。

二维IDCT可以分解为2个一维的IDCT[4]，如图2所示。首先进行行IDCT，此时转换的结果保存到一个中间RAM中，待8行IDCT转换完毕以后，再使用同一个一维IDCT处理

基金项目：天津市科技支撑计划基金资助重点项目(08ZCKFGX01400)
作者简介：李庆诚(1964－)，男，教授、博士、博士生导师，主研方向：嵌入式操作系统，数字版权保护体系，电子图书与产业对策；白振轩，硕士研究生；刘 洋，博士研究生；胡海军，硕士研究生
收稿日期：2009-02-04 **E-mail：**bzhx@yahoo.cn

单元进行列 IDCT 转换。行列 IDCT 采用同样的设计。通过将二维 IDCT 转换为一维 IDCT 虽然在运行速度上有一定降低，但是较大幅度地节省了 FPGA 门数的消耗[5]。

1.2 异步输入输出

由于 JPEG 解码器输出速度非常快，基本上一个时钟周期一个像素点，并且没有专用的输出 RAM，只能将数据传输给处理器，处理器是不可能进行这么快的数据处理，因此输出必须是异步的。

同理，JPEG 解码器必须等待处理器将数据送到以后才能进行处理。为了减少颠簸，输入采用的是异步缓冲。

2 异步接口的实现

2.1 FSL 接口

FSL 总线是基于 FIFO 的单向点对点通信总线，主要用于 MicroBlaze 处理器与协处理器之间的高速数据传输[6]。该接口的主要特点为：(1)单向的点对点通信；(2)非共享的无仲裁通信机制；(3)支持控制位与数据分离的通信；(4)基于 FIFO 的通信模式；(5)可配置的数据宽度；(6)高速的通信性能(独立运行达到 600 MHz)。

2.2 FSL 总线与协处理器的接口

协处理器器需要读取 Microblaze 处理器通过 FSL 总线传输过来的 JPEG 文件数据，同时在解码完毕以后，还要通过 FSL 总线将数据传输到 Microblaze，总共需要 2 条 FSL 总线连接。采用此连接方式， Microblaze 与协处理器实现了双向通信。其中一条专门用于将数据从 Microblaze 传送到协处理器中，另一条专门用于数据从协处理器到 Microblaze 的传输。

实验证明，采用一个 Microblaze 与协处理器连接处理数据需要异步阻塞，比较麻烦。可采用 2 个 Microblaze 处理器与一个 JPEG 解码器相连的方法，本文验证部分给出了模型。

2.3 异步输出的实现

协处理器的输出是以 MCU 为单位的，协处理器会缓存数据到一个 RAM 缓冲区中，只有当一整块数据都解码完毕以后才进行输出。由于 FSL 缓冲区有限，不可能容纳下整个图片，因此协处理器的输出必须是异步的，也就是能够随时停下来，直到 FSL 缓冲区中有了空间再进行输出。如图 2 所示，本文采取了协处理器在解码完一个 MCU 以后停下来，直到整个块完整输出以后才继续进行解码工作。

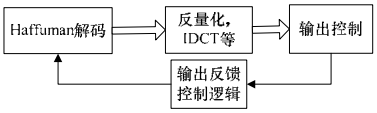

图 2 异步反馈

在 Huffman 解码过程中一旦识别到已经解码了一个完整的 MCU 以后就停止；直到输出控制模块，反馈一个信号让其继续进行 Huffman 解码。而输出控制保证在一个 MCU 都输出完毕以后才反馈此信号。

异步输出控制逻辑的状态切换图如图 3 所示。

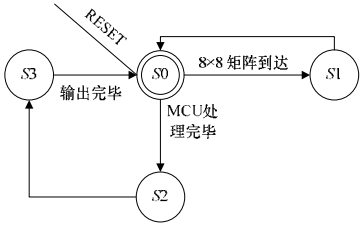

图 3 异步输出状态机

*S*0 状态：系统复位以后就进入该状态，等待 Huffman 解码的输出。该状态对于 8×8 矩阵块进行计数。以行列采样比 4:1:1 的图片为例，包括 4 个 *Y* 矩阵，1 个 *Cb* 矩阵，1 个 *Cr* 块。也就是说计数器到达 6 以后就转到状态 S2 。而 *S*0 到状态 *S*1 之间切换主要是依据 Huffman 解码时输出的信号 *HmDecCount*，当 *HmDecCount*=63 时就是一个完整的 *Y*/*Cb*/*Cr* 8×8 矩阵块到达了。

*S*1 状态：等待下一个 8×8 矩阵块的开始。上一个 8×8 矩阵块解码结束以后，过几个时钟周期以后，下一个块才开始。

*S*2 状态：等待输出控制逻辑反馈的开始输出信号。一旦输出模块开始输出就进入 *S*3。

*S*3 状态：等待输出控制逻辑反馈的输出完毕信号，一旦输出完毕就清零计数器，同时转到状态 *S*0。

Huffman 解码模块只在 *S*0 和 *S*1 模块进行解码工作，而输出模块可以随时被中断，这样就保证了异步输出。

2.4 输入缓冲的实现

协处理器在处理输入的时候也必须等待数据到来以后才能进行数据处理，由于处理器的速度与协处理器速度的不匹配，最坏情况下将使解码器在每次获取数据的时候都必须等待，这就造成了系统的颠簸，尤其对于 JPEG 解码器这样的多级流水线的结构，停下来的后果造成的损失比较大，因此在输入模块本文提出了缓冲以减少系统颠簸的处理方案。具体来说就是将接收到的数据保存到一个固定大小(大小可以根据测试调整，以达到最佳性能)的缓冲区中，只有缓冲区填满以后，解码器再从缓冲区中读取数据。图 4 给出了缓冲区的工作原理。由于添加了一层外围控制，从状态机上可以保证一旦解码过程中出现了问题，因此可以在数据处理送完以后进行复位，从而不影响后续工作。

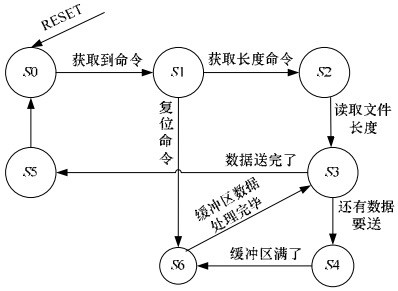

图 4 输入缓冲状态机

对图 4 中个状态说明如下：

状态 *S*0：等待处理器送第 1 个数据，一旦有数据读取并就进入 *S*1。

状态 *S*1：对该数据(命令)进行分析，如果表示下一个要送的数据是图片长度，就转入状态 *S*2，如果是复位命令就进入 *S*6。对于所有其他不能识别的命令直接忽略，仍然处于 *S*1。

状态 *S*2：获取图片长度，这里的长度指的是 JPEG 压缩文件的大小，然后转入 *S*3。

状态 *S*3：如果数据没有送完(也就是到达的数据数量小于等于图片长度)，就进入 S4，否则进入 S1。

状态 *S*4：读取数据进行缓存，如果缓冲区满了就转入 *S*5，否则一直读取。

状态 *S*5：在解码控制逻辑部分需要数据时送入数据，直到将缓冲区中数据全部送入，然后转入状态 *S*3。

状态 *S*6：复位内部解码逻辑，然后进入 *S*0。

3 验证实验

本文采用3种方案对比试验结果：(1)采用1个Microblaze核通过FSL总线与协处理器相连。(2)采用2个Microblaze核通过FSL总线与协处理器相连，其中1个负责接收，另外一个负责发送。方案3采用纯软件解析JPG图片。同时，给出相应的仿真时间，即协处理器理论上最优的时间。

方案1发送接收都在一个处理器上进行，软件设计有一定复杂度，经测试如采用Xilkernel对于性能影响比较大，因此，并没有采用多线程设计。图5为方案1的软件工作流程。

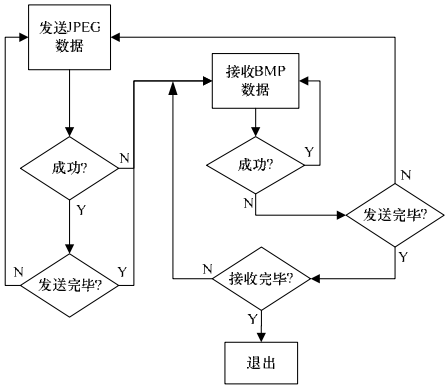

图5　数据接收流程

方案2使用2个处理器核测试，该方案软件比较简单，一个负责发送，另一个负责接收，不存在阻塞而导致无法运行的问题。但硬件架构方面还需要一个专门的FIFO_LINK用于2个处理器之间的通信。方案2的硬件架构如图6所示。

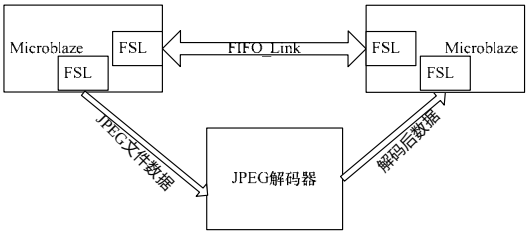

图6　双核硬件架构

在该方案中需要的FIFO_LINK实际上就是一个简单的基于FSL的数据通道。它用VHDL描述的实现过程如下：

```
architecture EXAMPLE of fifo_link is
begin
  FSL_M_Data <= FSL_S_Data;
  FSL_M_Write <= FSL_S_Exists and (not FSL_M_Full);
  FSL_S_Read <= FSL_S_Exists and (not FSL_M_Full);
end architecture EXAMPLE;
```

方案3软件解码器来自Geldreich R编写的Small JPEG Decoder Library[7]，经过少量修改以适应平台需要。

表1为几种解决方案图像处理的时间的对比，统计了对于不同大小的图片，不同解决方案下花费的时间。

表1　不同方案处理时间

图片分辨率	处理时间/ms			
	方案(1)	方案(2)	方案(3)	仿真
128×128	6.8	6.8	78.9	5.9
512×256	54.7	42.9	238.7	566.3
700×500	145.5	105.1	566.3	104.3

其中，Microblaze工作在100 MHz，而JPEG解码协处理器工作在10 MHz，所有图片都是4∶1∶1 JFIF格式的JPEG压缩文件。从实验结果可以看出，方案1和方案2明显优于方案3。虽然方案3工作频率是100 MHz，10倍于协处理器的工作频率，但由于方案1和方案2都采用了专用的协处理器，使其速度大幅度提升。而方案1与方案2相比，由于方案2采用2个Microblaze来负责JPEG解码器数据的发送和接收，不必考虑阻塞问题，方案1只有1个Microblaze需要处理发送阻塞和接收阻塞，以及在发送和接收2个状态之间切换，相对耗时，图片越大越明显，而图片较小时基本不会阻塞，因此在图片较小的时候两者之间的差异很小。

仿真采用ModelSim 6.1c，并且假定协处理器读取时总有数据，输出时总后缓冲区，也就是不存在阻塞问题，因此也就是理论上的最优时间。当图片较大时，方案1在时间上与理论时间差距比较大，而方案2则几乎一直相等(须将测试程序本身的运行时间考虑进去)。

但方案(1)只需一个处理器，这在很多嵌入式系统中是可行的，方案(2)在实际应用中很少采用。但通过方案2可以看出JPEG处理器由于输入输出阻塞而降低的时间。

4　结束语

本文介绍的JPEG协处理器在V2P30上ISE综合后报告可以在80 MHz的频率下正常工作。经过验证在66.7 MHz下可以正常稳定地工作，但本文提供的测试数据都是在10 MHz下测出的。经测试该协处理器可以与Microblaze无缝结合，在FSL缓冲区只有1个字(32 bit)的情况下仍可正常工作。

在嵌入式产品中，图像处理一直是其性能提高的瓶颈，本文提出的标准JFIF格式的JPEG文件解码协处理器能大幅提升嵌入式产品的性能，同时由于协处理器工作在较低的频率，该处理器也能降低产品运作的功耗。

参考文献

[1] Xilinx Inc.. MicroBlaze Processor Reference Guide[Z]. (2004-08-24). http://www.xilinx.com/support/documentation/sw_manuals/edk63i_mb_ref_guide.pdf.

[2] Hamilton E. JPEG File Interchange Format[Z]. Miarosystems. 1992.

[3] 云　风. JPEG 简易文档[Z]. (2007-02-05). http://www.codingnow.com.

[4] 来蒲军, 桑红石, 陈朝阳. 一种用于JPEG IP核的DCT设计[J]. 计算机与数字工程, 2006, 35(3): 148-150.

[5] 钟文荣, 陈建发. 二维DCT算法的高速芯片设计[J]. 厦门大学学报: 自然科学版, 2005, 44(2): 68-70.

[6] 李庆诚, 张　杰, 汤建军. FSL总线IP核及其在MicroBlaze系统中的应用[J]. 单片机与嵌入式系统应用, 2005, (6): 135-137.

[7] Geldreich R. Small JPEG Decoder Library[Z]. (2000-04-23). http://www.voicenet.com/~richgel.

编辑　金胡考