

高性能全流水线可控参数 JPEG-LS 编码器实现

刘嘉晗^{1,2}, 赵 岩², 王宇心², 赵军锁²

(1 中国科学院 研究生院, 北京 100049; 2 中国科学院 软件研究所, 北京 100190)

摘 要: 提出了一种高性能的 JPEG-LS 无损/近无损图像压缩算法 VLSI 实现结构. 通过对 JPEG-LS 算法瓶颈的分析, 针对算法中不利于流水线实现的场景缓存部分, 采用了一种信号量集机制避免流水线等待. 全流水线结构保证了算法实现可以满足高速图像传感器系统的吞吐量需求. 同时通过高度参数化的设计, 系统可以动态调整和优化算法参数, 使压缩效果和效率适应不同的运行环境. 算法在 FPGA 平台通过验证, 并得到了接近甚至超过其他 ASIC 实现的性能.

关键词: 图像处理; 无损压缩; JPEG-LS; VLSI; FPGA

中图分类号: TN911

文献标识码: A

文章编号: 1000-7180(2010)02-0034-04

A High Performance, Fully Pipelined and Parameter Controllable Implementation of JPEG-LS Encoder

LIU Jia-han^{1,2}, ZHAO Yan², WANG Yu-xin², ZHAO Jun-suo²

(1 Graduate School of Chinese Academy of Science, Beijing 100049, China;

2 Institution of Software, Chinese Academy of Science, Beijing 100190, China)

Abstract: A high-performance VLSI implementation of JPEG-LS algorithm for lossless/near-lossless image compression was proposed in this paper. After analyzing the bottlenecks in JPEG-LS, a semaphore collection mechanism was employed to avoid pipeline interruption in the part of context buffer. The fully-pipelined structure ensured its feasibility to high-speed image processing system, in which ultra-high-throughput is required. Benefiting from its highly-parameterized design, host systems could dynamically adjust and optimize algorithm parameters, in order to adapt compression result and efficiency to different environments. The implementation was verified on FPGA platform, and the result shows high performance that was close to or exceeding other ASIC implementation.

Key words: image processing; lossless compression; JPEG-LS; VLSI; FPGA

1 引言

JPEG-LS 无损/近无损压缩算法在医疗和遥感图像等对图像细节有苛刻要求的领域获得了重要应用. 在目前几种性能优秀的 JPEG-LS 的硬件实现中^[1], 流水线机制是提高数据吞吐量的主要手段. 同时, 硬件实现中使用并行运算^[2], 使之相对于软件算法的性能优势更加明显. 然而, JPEG-LS 算法也存在不利于硬件实现的方面, 即数据依赖问题, 这一问题已成为制约算法性能的瓶颈.

在遥感和工业应用中, 多传感器的应用要求图

像压缩单元能够接受和处理不同分辨率和精度的图像. 由于应用领域的复杂, 算法对于不同特征的图像 (熵值, 灰度分布等) 具有不同的压缩效果, 很难保证算法的某一特定实现始终满足系统的功能和性能要求 (压缩率、处理速度等). 因此, 对算法参数进行动态调整从而达到对图像特性的自适应也是一个重要的需求.

文中提出了一种全流水线 JPEG-LS 硬件实现, 该实现结构避免了大多数情况下数据依赖造成的流水线空闲等待, 是目前数据吞吐量最高的 JPEG-LS 实现. 另外, 该结构还实现了可变输入格式和参数动

态调整,使其能够适应复杂应用环境的需求。

2 JPEG-LS 算法分析

从 JPEG-LS 算法的流程可以看出,在场景统计信息和误差值之间存在数据环路:误差值的修正依赖于场景统计信息,而场景统计信息的更新依赖于误差值的结果^[3]。两者的计算不能够并行进行,这成为高性能 JPEG-LS 实现,尤其是基于流水线的实现中需要克服的重要困难。

算法的一个重要参数是阈值集(T_1, T_2, T_3),它的选择决定了场景建模中像素模板到场景空间的映射。尽管适当选择的阈值集能够满足多数情况下场景建模的需要,在运行环境中如果可以调整阈值集的组合,就能够解决很多由于场景区分度产生的码率和实时性问题^[4]。

另一个重要参数是精度损失 NEAR。JPEG-LS 算法支持近无损模式,此模式能够在保证图像损失低于给定范围 NEAR 的情况下提高压缩比,即压缩损失是可控的,这在对图像恢复损失要求严格的领域受到欢迎。在另一些系统中,则可以通过控制损失参数 NEAR 调整图像码率使之适应传输信道的需要。

3 实现结构

JPEG-LS 算法包括较多顺序运算步骤,如果直接按照算法实现为硬件逻辑,将会产生很大的延迟时间,严重阻碍时钟频率的提高。文中提出的 JPEG-LS 实现结构以提高最大吞吐量为主要目标,通过多级流水线降低每一级运算的延迟,从而满足更严格的时序约束。经过优化后的算法框图如图 1 所示。除流水线机制外,通过在较细的粒度上(运算符级)实现运算的并行,提高了每一级运算的速度。

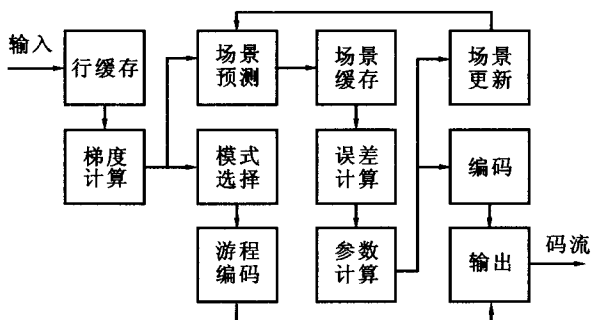


图 1 JPEG-LS 实现框图

3.1 场景缓存结构

JPEG-LS 场景建模需要维护深度为 365 的场景缓存,其中存储了各场景的统计参数。在基于流水

线的实现方案中,由于场景参数的计算存在反馈环路,场景缓存的读取需要在场景更新完成后才能继续进行,引起流水线中断,成为运算瓶颈。为了解决这一问题,文中采用了一种基于信号量集的硬件互斥机制,使不同场景的读取操作能够连续进行,仅当访问已读取但尚未更新的场景时才中断等待,消除了大多数情况下不必要的流水线中断。该互斥机制可以表示为两个原子操作 $\text{load}(Q)$ 和 $\text{update}(Q)$, 以及一个深度为 365 的互斥信号量集 $m[0..364]$:

```
load(Q){
    wait until  $m[Q] = 0$ ;
     $m[Q] = 1$ ;
}
Update(Q){
     $M[Q] = 0$ ;
}
```

该互斥机制的硬件实现如图 2 所示,使用了 365x1 位的双口 RAM 实现信号量集。

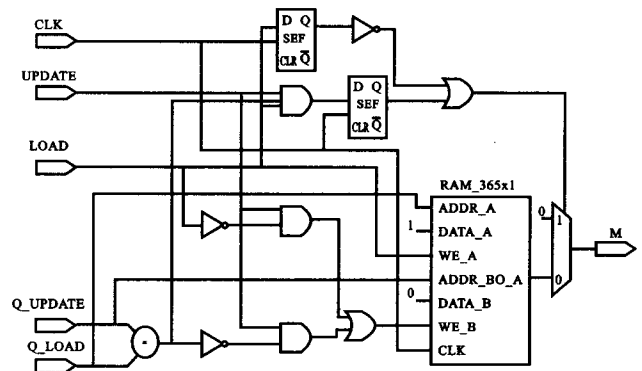


图 2 互斥机制硬件结构

场景缓存的数据宽度根据场景分布参数 A, B, C, N 确定。 N 为场景计数,其精度为 n 位。 C 为预测纠正参数,一般取值为 $-128 \sim 127$ 。图像精度为 $\text{bpp}(\text{bits per sample})$, 根据各参数之间的依赖关系,可以得到 A, B 的精度分别为 $n + \text{bpp} - 1, n + 1$, 场景缓存 RAM 的大小为 $365 \times (\text{bpp} + 3 \times n + 8)$ 位。研究表明^[3], n 取 5~8 位即可满足绝大多数应用环境的需求^[5]。

3.2 行缓存结构

JPEG-LS 算法预测级的计算需要同时访问当前行和前一像素,但由于计算是在空间上顺序进行,实际上只需要缓存一行像素。为了提高性能,行缓存的操作按三级流水线执行:第一级读取 RAM 获得 R_a ;第二级计算 R_a, R_b, R_c 各像素的值;第三级将 R_a 写入 RAM。行缓存的结构如图 3 所示。因为 I_x, R_a 起到了两个存储位置的作用,实际上需要

的 RAM 深度为 $spl - 2$. spl 为每行图像的像素数量 (samples per line).

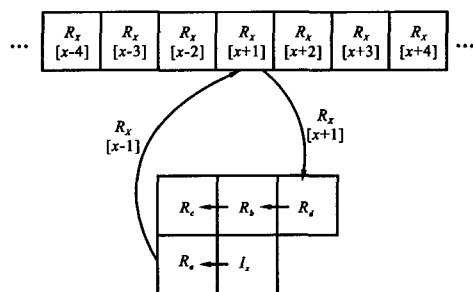


图3 行缓存结构

由于缓存深度只与图像列数或宽度有关, 根据系统需要支持的图像最大列数 X , 可以得到行缓存 RAM 的大小为 $(X - 2) \times bpp$ 位。

3.3 流水线结构

文中提出的实现结构将 JPEG-LS 运算流程划分为三条流水线: 通用流水线、常规模式流水线和游程模式流水线。通用流水线操作包括行缓存, 梯度计算, 模式选择, 误差修正, 像素重建, Golomb 参数计算, Golomb 编码和数据输出; 常规模式流水线操作包括预测, 建模, 场景存取, 预测值修正, 误差计算, 误差映射和场景更新; 游程模式流水线操作包括建模, 场景存取, 游程编码, 误差映射和场景更新。三条流水线在时间域上相互重叠, 并行执行。带有时域信息的流水线结构示意图如图 4 所示, 图中每条虚线代表流水线中的一个时隙。

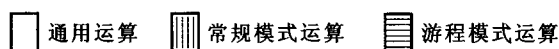
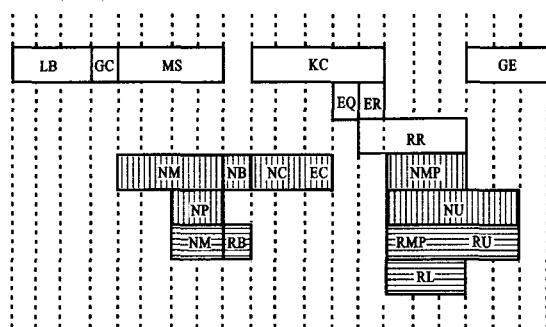


图4 流水线结构

3.4 动态参数调整

与其他 JPEG-LS 实现不同, 文中不假定应用对象为某一类特征的图像, JPEG-LS 算法中的各项参数 ($MAXVAL$ 、 T_1 、 T_2 、 T_3 、 $NEAR$) 以及图像规格 (bpp 、 spl) 都实现为运行期间可调整的方式。算法参数保存在寄存器文件中, 系统在运行期间通过控制接口设定和变更参数值。采用这一设计后, 系统在运行期间可以根据环境情况如曝光度、噪声、码率限制

等条件动态调整算法, 使处理结果在性能、损失和码率之间达到最佳平衡。

3.5 低功耗设计

高速数字系统中功耗的主要部分来自逻辑电平切换引起的电荷充放电, 因而功耗的大小与系统内部电平切换的活动程度成正比。低功耗设计的一个主要途径是减少异步电路中不等长时延造成的信号抖动, 或毛刺, 减少电荷充放, 这一方法称为平衡电路 (Balanced Circuit)。

基于流水线的结构更多地实现了平衡电路, 尽管由于寄存器的增加使静态功耗略有上升, 但由于减少了电路中的异步状态切换, 其动态功耗特性优于非流水线结构。现有研究表明, 针对 JPEG-LS 算法, 相对于非流水线结构, 流水线结构能够节省 25% 的功耗^[1]。

为了进一步降低动态功耗, 在三条流水线的各个层级的输入寄存器中加入门控机制, 在数据无效时禁用输入寄存器。由于常规模式和游程模式不会同时执行, 引入门控后, 减少了大量的无用状态切换, 从而达到降低动态功耗的目的。

4 实现结果

算法采用可综合 RTL 代码实现, 在 FPGA 平台上通过验证。文中面向 SOC 应用, 集成了 AMBA, OPB 和 WISHBONE 系统接口 (可配置)。高可复用性和可移植性使其易于系统集成, 并可以简单地移植到 ASIC 实现。文中方法具有良好的可配置性, 通过调整配置可将其应用于特定目标的系统, 并在多个 FPGA 平台上对其多种配置进行了测试, 其结果如表 1 所示。以在 XC4VLX15 平台上的 8 位实现为基准, 与其他几种 JPEG-LS 的 VLSI 实现的比较如表 2 所示。

表1 JPEG-LS 算法 FPGA 实现结果

目标工艺	bpp 上限	spl 上限	逻辑量/ slices	存储空间/ (kb/BRAM)	工作频率/ MHz
XC2V500	8	640	1 894 (61%)	18.9/4 (12%)	154.3
XC2V500	16	2 048	2 529 (82%)	49.6/5 (15%)	152.7
XC4VLX15	8	640	2 294 (37%)	18.9/4 (8%)	197.5
XC4VLX15	16	2 048	2 965 (48%)	49.6/5 (10%)	195.8
XC5VLX30	8	2 048	1 353 (28%)	49.6/4 (12%)	225.1

文献[6]中提出了一种低功耗的实现方案,通过划分多个时钟域的方法,将瓶颈计算在高速时钟域中执行,其他操作在低速时钟域中执行,以提高性能和降低功耗.这一方法的缺点是没有真正避免数据依赖的问题,高速时钟的引入使整个系统的时序约束更加紧张,无法达到更高的运行频率.文中给出的10MHz的实现结果证明了这一点.另外,多时钟的

管理也更加复杂.

文献[2]中采用了一种改进的算法 N2C-EX,实现了并行运算,达到了很高的性能,但文中给出的结果表明,N2C-EX 是以两倍的缓存空间和牺牲一定程度的压缩比为代价的,而且 N2C-EX 与 JPEG-LS 并不能实现数据兼容^[7].

表2 与其他 JPEG-LS 实现的特性比较

文献	目标工艺	逻辑量	存储 空间	工作频 率/MHz	性能 对比/%
文献[5]	XCV50	—	—	12.0	6
文献[1]	XC4VLX15	2 596slices	6 BRAM	102.6	52
文献[6]	ASIC0.18 μ m	17.6k eq. gates	17 530 bits	10.0	5
文献[1]	ASIC 0.18 μ m	27.7k eq. gates	20 815 bits	183.0	93
文献[2] ^①	ASIC 0.13 μ m	17.3k eq. gates	13 984 bits ^②	333.0	169
文献[1]	ASIC 0.13 μ m	21.5k eq. gates	15 695 bits ^②	263.0	133

①基于 N2C-EX 算法;②仅包括场景缓存

文献[1]中提出的是一种高性能的实现,通过对变量 $C[Q]$ 的超前预测解决数据依赖问题,避免了流水线等待.该实现保证了算法的最差性能,适用于实时性要求比较高的领域.该方法的局限在于,超前预测和比较选择的逻辑限制了流水线的级数,在该结构上进一步提高时钟频率比较困难^[8].

5 结束语

文中通过对 JPEG-LS 算法瓶颈的分析,提出了一种利用信号量集机制避免流水线等待的方法,并应用到算法的硬件实现中.文中的 JPEG-LS 实现在保证最大吞吐量的基础上提供了对多图像格式和可控参数的支持,实现了高性能和高适用性的统一,适用于对压缩速率和码率有严格要求,需要对压缩效果有自适应调整能力的多传感器复杂系统.

参考文献:

- [1] Papadonikolakis M E, Kakarountas A P. Efficient high-performance implementation of JPEG-LS encoder [J]. Journal of Real-time Image, 2008(3): 303-310.
- [2] Ferretti M, Boffadossi M. A parallel pipelined implementation of LOCO-I for JPEG-LS[C]// ICPR'04. UK, Cambridge, 2004(1): 769-772.
- [3] JPEG/JBIG. FCD14495. Lossless and near-lossless coding of continuous tone still images[EB/OL]. [2000-11-10] [2000-08-15]. <http://www.jpeg.org/public/>

fcd14495p.pdf.

- [4] Weinberger M J, Sapiro G, Seroussi G. The LOCO-I lossless image compression algorithm: principle and standardization into JPEG-LS[J]. IEEE Trans. on Image Processing, 2000(9):1309-1324.
- [5] Klimesh M, Stanton V, Watola D. Hardware implementation of a lossless image compression algorithm using a field programmable gate array[R]. NASA JPL TMO Progress Report 42-144. Pasadena, CA: NASA, 2001.
- [6] Li Xiaowen, Chen Xinkai, Xie Xiang, et al. A low power, fully pipelined implementation of LOCO-I for JPEG-LS [C]// ICME'07. China, Beijing, 2007: 1906-1909.
- [7] Savakis A, Piorun M. Benchmarking and hardware implementation of JPEG-LS[C]// ICIP'02. NY Rochester, 2002(2): 949-952.
- [8] 杨武,董世都,张光建.一种基于小波系数上下文模型的图像压缩方法[J].微电子学与计算机,2006,23(4): 153-159.

作者简介:

刘嘉晗 男,(1983-),硕士研究生.研究方向为数字集成电路、嵌入式系统.

赵岩 女,(1983-),助理工程师.研究方向为图像处理、嵌入式系统.

王宇心 男,(1969-),副研究员,硕士生导师.研究方向为虚拟现实、计算机仿真.

赵军锁 男,(1973-),高级工程师.研究方向为星载计算机、嵌入式系统.