ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.7, July 2006, pp.1553−1560
DOI: 10.1360/jos171553

E-mali: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

高性能的 EBCOT 编码及其 VLSI 结构*

刘 凯[1+], 李云松[2], 吴成柯[2]

[1](西安电子科技大学 计算机学院,陕西 西安 710071)

[2](西安电子科技大学 ISN 国家重点实验室,陕西 西安 710071)

A High Performance EBCOT Coding and Its VLSI Architecture

LIU Kai[1+], LI Yun-Song[2], WU Cheng-Ke[2]

[1](School of Computer, Xidian University, Xi'an 710071, China)

[2](National Key Lab. of Integrated Service Networks, Xidian University, Xi'an 710071, China)

+ Corresponding author: Phn: +86-29-88203110, Fax: +86-29-88203116, E-mail: kailiu@mail.xidian.edu.cn

Liu K, Li YS, Wu CK. A high performance EBCOT coding and its VLSI architecture. *Journal of Software*, 2006,17(7):1553−1560. http://www.jos.org.cn/1000-9825/17/1553.htm

Abstract: This paper proposes an efficient architecture composed of bit plane-parallel and pass-parallel coder for EBCOT (embedded block coding with optimized truncation) entropy encoder used in JPEG2000.After the detailed analysis of EBCOT architecture in JPEG2000, the coding information of each bit plane and the corresponding passes can be obtained simultaneously. Therefore, bit plane-parallel and pass-parallel coding (BPPP) is proposed, and its VLSI architecture is shown in details. The analysis and the corresponding experimental results show that the proposed architecture reduces the processing time greatly compared with others, and a FPGA prototype chip is designed and can process 512×512 gray level images with 30 frames per second at 65MHz working frequency. The quality of images reaches the results released by JPEG2000.

Key words: EBCOT (embedded block coding with optimized truncation); bit plane-parallel and pass-parallel; block encoder

摘 要: 提出了比特平面与编码过程全并行处理的 EBCOT(embedded block coding with optimized truncation)编码结构.通过分析 JPEG2000 和国内外提出的 EBCOT 编码结构,指出不仅每一个比特平面,而且对应的编码过程的编码信息可以同时获得,从而给出了比特平面与编码过程全并行处理的块编码方法,并且详细说明了实现的 VLSI 结构.理论分析以及具体实验结果表明,比特平面与编码过程全并行处理所需的时钟周期最少,FPGA 原型系统最高时钟频率可达 65MHz,对于 512×512 的灰度图像,处理速度可达 30fps,完全可以实时处理,图像质量达到了公布的 JPEG2000 标准.

关键词: EBCOT 算法;比特平面与编码过程全并行;块编码

中图法分类号: TP391 文献标识码: A

* Supported by the National Natural Science Foundation of China under Grant Nos.60532060, 60507012 (国家自然科学基金)
Received 2004-05-31; Accepted 2005-05-23

JPEG2000[1]作为新一代的静止图像压缩国际标准,具有丰富的特性,能够进行单分量或多分量的有损和无损编码,还同时支持 SNR 和分辨率渐进传输、感兴趣区(ROI)编码、码流随机访问,提供灵活的文件格式,支持内置用户信息(如版权信息)、图像序列(motion JPEG),并具有良好的抗误码特性.然而,如此多优良特性的代价是实现的复杂度提高.作为 JPEG2000 核心部分的 EBCOT(embedded block coding with optimized truncation)[2],编码模块就占了整个编码时间的一半以上.因此,EBCOT 的设计与实现就成为进行 JPEG2000 芯片设计的关键问题,对能否进行实时处理起着至关重要的作用.

1 引 言

针对 EBCOT 硬件结构,不少学者提出了自己的算法和实现结构,其中主要有 David Taubman[3]给出的基于像素点的实现结构,Andra[4,5]提出的状态机结构的比特平面编码,J.-S. Chiang[6]提出的编码过程并行(pass-parallel)结构,以及 Chung-Jr Lian[7]提出的基于列(column-based)结构等.这些实现结构主要根据 EBCOT 编码本身的特点,提高实现的并行度和减少时钟的浪费.例如:J.-S. Chiang 即将 3 个编码过程(coding passes 或称为子平面编码 fractional bit-plane coding)同时处理生成上下文进行算术编码,称为编码过程并行的上下文模型(pass-parallel context modeling,简称 PPCM);而 Chung-Jr Lian 则采用像素跳过(sample skipping,简称 SS)和列组跳过(group-of-column skipping,简称 GOCS),避免了没有上下文输出的像素点以及列组,从而提高了处理速度.通过对这些结构以及对 EBCOT 的仔细分析,这些实现结构并没有完全利用 EBCOT 本身内部具有的并行特性,即不但编码过程可以并行处理,而且编码平面也可以并行处理.正是基于此,我们提出了比特平面与编码过程全并行的上下文模型(bit plane-parallel and pass-parallel context modeling,简称 BPPP)硬件结构.该结构充分利用了 EBCOT 并行度,包括比特平面和编码过程实现的并行,其中编码模块完全由组合逻辑实现,而且不会在单像素位置产生时钟浪费,因此编码速度和效率有显著改善.下面,我们首先给出 EBCOT 码块编码的具体分析,然后给出 BPPP 编码模型以及硬件实现结构,并且对 BPPP 模型所进行的具体的时序分析给出相应的实验结果.

2 EBCOT 编码及 BPPP 模型

2.1 EBCOT编码过程

JPEG2000 的编码系统基本结构[1]如图 1 所示.

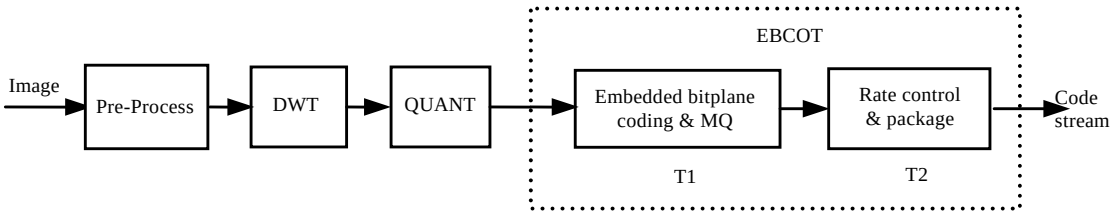

Fig.1 Block diagram of the JPEG2000 encoder

图 1 JPEG2000 的编码系统基本结构框图

EBCOT 算法分为 T1 和 T2 两部分.T1 由内嵌比特平面编码和自适应算术编码器 MQ 组成,T2 部分完成率控制和码流组织.EBCOT 编码时,各小波子带划分为更小的码块(如 64×64),以码块(code-block)为单位独立作 T1 编码.不同的码块产生的比特流长度是不同的,它们对恢复图像质量的贡献也是不同的.因此,对于所有码块产生的比特流,T2 部分采用了率失真优化技术进行后压缩处理(PCRD optimization: post compression rate distortion optimization),即对各码块的码流,按对恢复图像的质量贡献分层,完成码流的率控制和组织.

码块 B_i 的比特平面编码是按下列 3 个编码过程(passes 或者 fractional bit-planes)顺序进行编码的:重要性传播(significance propagation pass)、幅值细化(magnitude refinement pass)和清理更新(cleanup pass).设 $P_i^{p,1}$, $P_i^{p,2}$ 和

$P_i^{p,3}$ 是码块 B_i 在比特平面 p 时这 3 个编码过程的编码信息,则如图 2 所示为比特平面编码的流程结构.其中最高比特平面 M_i–1 所有状态变量为 0,只进行清理更新.

Fig.2 Bit plane coding sequence

图 2 比特平面编码步骤

在每个过程中使用了“零编码(ZC)”、“符号编码(SC)”、“幅值细化(MR)”和“游程编码(RLC)”,4 种编码原语进行上下文形成操作.为了形成上下文,对应码块中每一个像素位置使用了 4 个状态变量,它们分别是:重要性状态位(σ),用以表示系数的重要性状态;符号位(χ),用以表示系数的符号状态;访问位(η),用以表示系数在当前比特平面是否编码过状态(当该系数在该比特平面某个编码过程中进行了编码后置为‘1’,否则置为‘0’)和细化状态位(σ'),用以表示系数是否进行了 MR 编码,以及 1 位用于表示系数在该比特平面的幅值位(ν)状态.原语的选择,就是根据系数 $s_i[m,n]$所处的编码阶段以及与其直接相邻的 8 个系数状态来决定.

具体的比特平面编码算法如下:

(1) 重要性传播, $P_i^{p,1}$:对于不重要但有重要邻居的系数进行 ZC 编码,如果系数变成重要,则进行 SC 编码;

(2) 幅值细化, $P_i^{p,2}$:对于上一位平面已经重要的系数,进行 MR 编码.

清理更新, $P_i^{p,3}$:对于不重要且未编码过的系数进行 ZC 或 RLC 编码,如果有系数变成重要,则进行 SC 编码;返回第(1)步进行下一个比特平面 p–1 的编码.

从以上分析我们可以看出:EBCOT 是按比特平面串行处理,在每一个比特平面内也是按照编码过程串行处理完成整个编码,整个码块的处理时间为 $N\times N\times((B-1)\times 3+1)$,$N\times N$ 表示码块大小,B 表示参与编码的比特平面数.我们设 N=64,B=10,则处理一幅 512×512 灰度图像的时间周期为 7 340 032,若进行实时处理(25 帧/秒),系统时钟至少为 183.5008MHZ.显然,如此高的时钟是很难实现的,必须对 EBCOT 进一步分析以减少处理周期,避免没有上下文生成的像素位置时钟的浪费.

2.2 BPPP编码模型

针对 EBCOT 本身的处理模式,主要考虑从并行度和避免时钟浪费两方面来提高速度.EBCOT 两个层次的串行处理是整个系统的瓶颈所在,即比特平面和编码过程的串行处理.那么,如果将这两个串行处理变为并行处理,则处理速度可以有很大的提高.为此,我们提出了比特平面与编码过程全并行的上下文模型.从以下分析我们会看到,状态变量的处理将是实现 BPPP 的关键.

2.2.1 比特平面的并行处理

由比特平面编码算法的分析可以看出,编码过程是从最高比特平面(MSB)开始,依次向低比特平面顺序进行的,各个比特平面之间的关系是通过状态变量联系的,而每一个编码点在各个比特平面的状态变量是可以预先得到的,这就意味着比特平面的 EBCOT 编码可以不采用串行方式,而采用并行方式.为了达到比特平面并行,就必须在开始编码时,各个比特平面得到关于编码位置的正确状态变量值.然而在 4 个状态变量中,只有重要性状态和细化状态是与比特平面位置有关的.当然,各个比特平面的幅值是不相同的,但幅值位是不会改变的,所以不必考虑幅值位对并行编码的影响.对 p^{th} 平面位置$[m,n]$处的重要性状态,为该平面以上各平面对应位置幅值位的或运算,而细化位则是 p+1th 平面位置$[m,n]$处的重要性状态.即

$$\sigma_p = \sum_{i=p+1}^{MSB} \nu_i[m,n], \qquad \sigma'_p = \sigma_{p+1} \quad (p \text{ 指平面号}) \tag{1}$$

其中,加法表示逻辑或运算.

其余的状态位可以直接从小波系数获得.至此,每一个比特平面的状态信息可以同时获得,这样就保证了比特平面并行编码的可行性.

2.2.2 编码过程的并行处理

在处理每一个比特平面时,为了达到 3 个编码过程的并行,就必须获得对应的正确状态变量值,具体来讲就是:在幅值细化时,要得到像素位置经过重要性传播后的状态变量值;清理更新时,要得到重要性传播和幅值细化之后的状态变量结果.在 4 个状态变量中,随编码过程发生变化的只有重要性状态位(σ)和访问位(η),其中访问位(η)可以通过组合逻辑进行预判决得到;重要性状态位(σ)只在系数变为重要时发生改变.图 3 给出了具体像素处理位置.

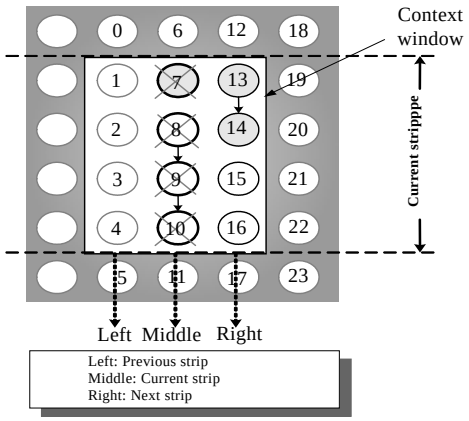

Fig.3 Pixel processing position

图 3 像素处理位置

由于 JPEG2000 是按 4 行为单位的条带扫描处理,图中位置 1,2,3,4 为已处理的前 4 个像素点;7,8,9,10 为当前编码的 4 个像素点;13,14,15,16 为下一个编码的 4 个像素点;19,20,21,22 为辅助编码点;0,6,12,18 为当前条带的上边界点;5,11,17,23 为当前条带的下边界点.其中,1~4,7~10, 13~16 构成了当前上下文形成窗.为了编码过程的并行,必须得到当前编码点(7~10)和下一编码点(13~16)在重要性传播后的重要性状态,以供幅值细化和清理更新过程同时使用.通过分析以及 JPEG2000 标准,我们得到关于这 8 个点重要性状态的更新公式如下:

$$\left.\begin{aligned}
\sigma_{7new} &= \sigma_7 + \nu_7(\sigma_0+\sigma_1+\sigma_2+\sigma_6+\sigma_8+\sigma_{12}+\sigma_{13}+\sigma_{14})\\
\sigma_{8new} &= \sigma_8 + \nu_8(\sigma_1+\sigma_2+\sigma_3+\sigma_{7new}+\sigma_9+\sigma_{13}+\sigma_{14}+\sigma_{15})\\
\sigma_{9new} &= \sigma_9 + \nu_9(\sigma_2+\sigma_3+\sigma_4+\sigma_{8new}+\sigma_{10}+\sigma_{14}+\sigma_{15}+\sigma_{16})\\
\sigma_{10new} &= \sigma_{10} + \nu_{10}(\sigma_3+\sigma_4+\sigma_5+\sigma_{9new}+\sigma_{11}+\sigma_{15}+\sigma_{16}+\sigma_{17})\\
\sigma_{13new} &= \sigma_{13} + \nu_{13}(\sigma_6+\sigma_{7new}+\sigma_{8new}+\sigma_{12}+\sigma_{14}+\sigma_{18}+\sigma_{19}+\sigma_{20})\\
\sigma_{14new} &= \sigma_{14} + \nu_{14}(\sigma_{7new}+\sigma_{8new}+\sigma_{9new}+\sigma_{13new}+\sigma_{15}+\sigma_{19}+\sigma_{20}+\sigma_{21})\\
\sigma_{15new} &= \sigma_{15} + \nu_{15}(\sigma_{8new}+\sigma_{9new}+\sigma_{10new}+\sigma_{14new}+\sigma_{16}+\sigma_{20}+\sigma_{21}+\sigma_{22})\\
\sigma_{16new} &= \sigma_{16} + \nu_{16}(\sigma_{9new}+\sigma_{10new}+\sigma_{11}+\sigma_{17}+\sigma_{15new}+\sigma_{21}+\sigma_{22}+\sigma_{23})
\end{aligned}\right\} \quad (2)$$

其中,加法表示逻辑或运算,乘法表示逻辑与运算.

在上式中,下标指示编码位置;$\sigma_{7new},\sigma_{8new},\sigma_{9new},\sigma_{10new}$以及$\sigma_{13new},\sigma_{14new},\sigma_{15new}$和$\sigma_{16new}$分别为幅值细化和清理更新过程时,位置 7~10,13~16 点的重要性状态;σ为对应位置在重要性传播时的重要性状态;ν为对应的幅值状态.

2.2.3 上下文生成

由于采用 4 点为一列的处理单元模式,那么,4 点(位置 7~10)在每个编码过程的上下文可以同时获得.在重要性传播过程中,可能生成的上下文数目为 8 个,分别是位置 7~10 的 ZC 和 SC 上下文;在幅值细化过程中,可能生成的上下文数目为 4 个,分别是位置 7~10 的 MR 上下文;在清理更新过程中,可能生成的上下文数目为 8+3 个,分别是位置 7~10 的 ZC 和 SC 上下文,以及 RLC 的 3 个上下文.这些上下文的生成完全可以根据 JPEG2000 标准中的要求,以组合逻辑的形式给出,同时设置相应的上下文有效信号,指示对应的上下文是否有效.这样,3 个编码过程的上下文产生部分完全可以由组合逻辑实现.同样,如果 4 个点中有不输出上下文的位置时,可以通过上下文有效指示在读取时跳过,实现了像素跳过和列跳过(column skipping,简称 CS),避免了单像素位置的时钟浪费.与文献[7]相比,我们实现的 SS 和 CS 方法不需要特定判决逻辑部分,实现简单.

3 BPPP 硬件结构

通过以上分析我们可以看出,在进行比特平面编码时,像素位置在各个比特平面以及编码过程对应的状态

可以经过简单的逻辑预先获得.这样,我们就可以根据 BPPP 模型,进行具体的实现 EBCOT 编码的硬件体系结构设计.下面,我们给出 BPPP 的实现细节.

3.1 整体结构

图 4 为整个 BPPP 模型的结构框图.

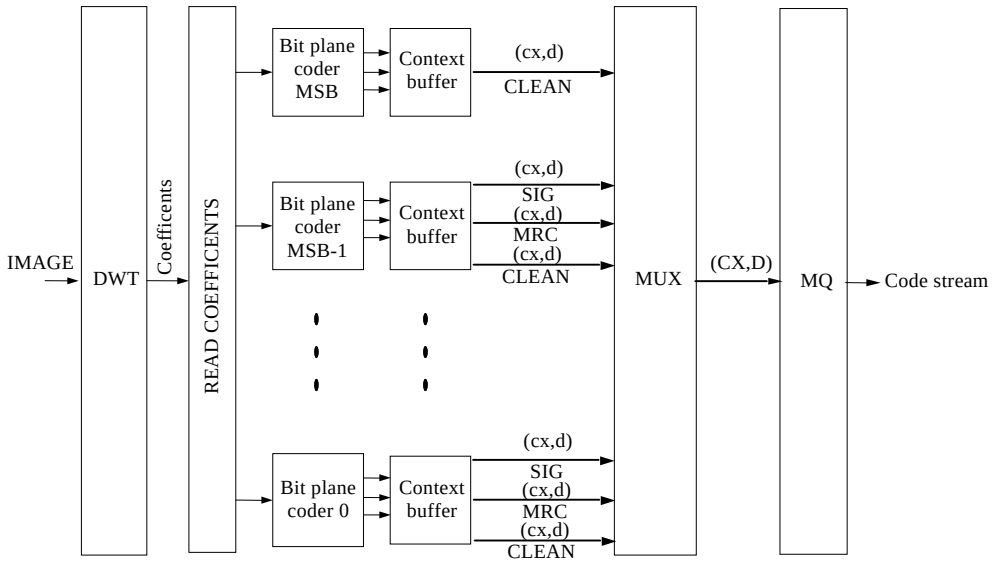

Fig.4 Architecture of BPPP coding system

图 4 BPPP 编码系统结构

原始图像首先进行离散小波变换(discrete wavelet transform),得到频率域的小波系数.这里,我们选择 5/3[8] 小波滤波器,小波系数的量化步长为 1,小波级数为 3.接下来读系数模块(read coefficients),根据分辨率渐进要求,将小波系数按级数和频带读出,送给各个比特平面编码器,具体为(LL HL LH HH)n(n=0~2).其中 0 对应第 1 级小波,2 对应第 3 级小波.同时,根据小波系数在各个比特平面的值,组织该系数在对应平面的状态变量.每一层的比特平面编码器接收小波系数和状态变量在该平面的值,分别输入到 3 个编码过程形成部件,按照算法的编码要求同时生成这 3 个编码过程对应的上下文和判决信息,即(CX,D)sig,(CX,D)mrc 和(CX,D)clp.生成的上下文和判决分别存放到每个平面对应的 3 个上下文缓冲中,各个平面 3 个编码过程产生的上下文和数据,通过多路选择器(MUX)输出到算术编码器(MQ)进行压缩编码.MQ 产生的压缩码流提供给 T2 编码模块进行优化截取,最终形成压缩码流.这里,由于上下文的标号范围 0~18 判决为二值判断,因此,上下文和判决采用 5+1 位表示即可.

3.2 扫描处理方式及状态变量组织

JPEG2000 提供了两种可选的条带扫描顺序:Regular 和 Vertical Causal.为了有效实现编码过程的并行处理,采用了 Vertical Causal 模式的扫描方式.这种方式下,避免了下一条带对当前处理条带的影响,从而简化了上下文生成部分逻辑.具体来讲就是,在该模式下认为,图 3 中,5,11,17,23 这 4 个下一条带像素位置为不重要的.那么,对于每一列只需考虑 5 个位置的状态变量,其中 4 个为当前处理条带的 4 个点以及一个上一条带的位置点.在 4 个状态变量中,访问位可以在编码时判断,无须存储,上一条带位置点只需存储符号位和重要位.这样,一列需要存储的信息为(4×3+2)+4 位,包括 4 位对应的幅值位,共 18 位.表 1 给出了状态寄存器的组织.

Table 1 State register
表 **1** 状态位寄存器

10				9				8				7				6	
σ'	σ	χ	ν	σ'	σ	χ	ν	σ'	σ	χ	ν	σ'	σ	χ	ν	σ	χ

表 1 中,第 1 行为图 4 中当前处理条带的位置标号;第 2 行为对应位置的状态信息位.

3.3 编码过程并行的比特平面编码器

每一个比特平面编码的内部结构,可如图 5 所示.

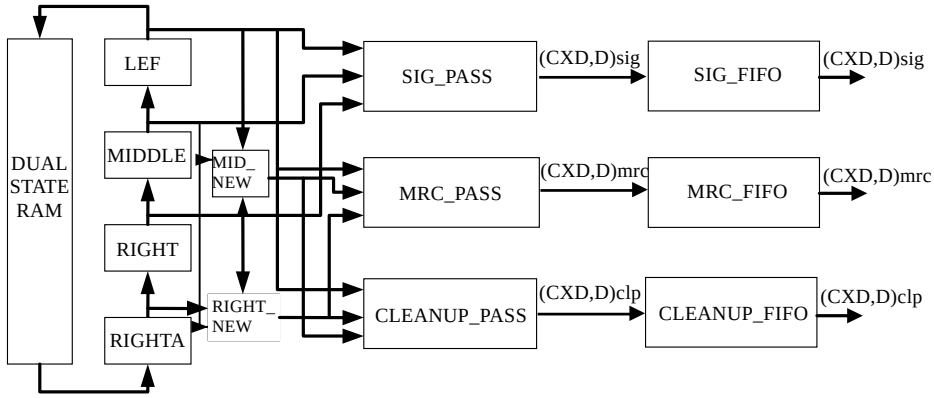

Fig.5 Architecture of pass-parallel bit plane coder
图 5 编码过程并行的比特平面编码器结构

DUAL STATE RAM 是用于存储码块状态信息的双端口存储器,大小为$(N\times N/4)\times 18$ 位,N 为码块的宽度.LEFT,MIDDLE,RIGHT 和 RIGHTA 是 18 位宽的寄存器,分别存放图 3 中位置 0~4,6~10,12~16 和 18~22 的状态信息.MID_NEW 和 RIGHT_NEW 是用于实现式(2)以供幅值细化和清理更新过程使用的组合逻辑门阵列.SIG_PASS,MRC_PASS 和 CLEANUP_PASS 是 3 个编码过程的上下文生成逻辑模块,内部完全按 JPEG2000 标准要求以组合逻辑实现,并经卡诺图进行优化.SIG_FIFO,MRC_FIFO 和 CLEANUP_FIFO 分别缓存对应编码模块生成的上下文和判决,FIFO 的深度由所处的编码过程决定.若码块大小 $N\times N$,对于重要性传播过程至多产生 $N\times N\times 2$ 个上下文和判决,即每一位置都输出 ZC 和 SC 上下文;对于幅值细化过程至多产生 $N\times N$ 个上下文,即每一位置都输出 MR 上下文;对于清理更新过程至多产生 $N\times N\times 2+N\times N/4$ 个上下文,即每一位置都输出 ZC 和 SC 上下文,每一列都输出 RLC 上下文.这样,3 个 FIFO 的大小可以分别为 $N\times N\times 2\times 5$ 位、$N\times N\times 5$ 位和 $(N\times N\times 2+N\times N/4)\times 5$ 位.实际上,每一个像素位置在 3 个编码过程中只进行一次编码,3 个 FIFO 的总体平均利用率只有 33.33%,因此在工程实现时,可以根据实际需要调整 FIFO 大小.

原始图像经离散小波变换(DWT)后,小波系数按比特平面组织成对应的状态变量写入 DUAL STATE RAM 内,其中每一地址单元内容以表 1 的顺序来组织.当一个码块所有条带均写入后,启动编码过程并行的比特平面编码器,进行 EBCOT 的码块编码.首先,依次从 DUAL STATE RAM 中读出 3 个地址单元至 MIDDLE,RIGHT 和 RIGHTA 寄存器,这时 LEFT 寄存器位全为 0;然后,组合逻辑模块按上下文产生规则生成对应上下文和判决以及上下文有效信号;最后,将 3 个编码过程生成的上下文和判决(CX,D)写入对应的 FIFO 中.然后,继续从 DUAL STATE RAM 中读出数据,将 MIDDLE 值赋给 LEFT,RIGHT 值赋给 MIDDLE 以及 RIGHTA 值赋给 RIGHT,读出的数据写入 RIGHTA 后继续编码,直到该平面的所有像素编码完毕.当所有参与编码的比特平面均完成编码时,从 MSB 平面开始,按照 $\text{CLEANUP\_FIFO}_{msb}$,$\text{SIG\_FIFO}_{msb-1}$,$\text{MRC\_FIFO}_{msb-1}$,$\text{CLEANUP\_FIFO}_{msb-1}$… SIG_FIFO_{lsb},MRC_FIFO_{lsb},$\text{CLEANUP\_FIFO}_{lsb}$ 顺序,依次读出上下文和判决进行算术编码.自此,一个码块完成了 EBCOT 码块编码.

3.4 状态变量的更新

根据重要性传播过程要求,对每一个条带处理时要用到最新的邻近条带状态变量,因此,状态变量的及时更新就非常必要.由图 3 可见,当前条带位置 10 也是下一个条带的上边界点(相当于位置 6).当位置 10 的重要性发生变化时,这种变换也必须反映在下一条带的处理过程中,即对下一条带相应位置列的状态变量进行更新.这种更新,具体来讲就是对 DUAL STATE RAM 的回写过程.由图 5 可见,LEFT 寄存器的输出是连接到 DUAL STATE RAM 的输入端,由此来完成对状态变量的更新过程.那么,在后续的编码过程中所使用的状态变量就是最新的,以确保编码正确.

4 时间分析和实验结果

设码块大小为 $N\times N$,参与编码的比特平面数为 B,该码块产生的上下文数目为 C,每个子平面产生的上下文数为

$$Cclp_{bp-1},Csig_{bp-2},Cmrc_{bp-2},Cclp_{bp-2},\ldots,Csig_0,Cmrc_0,Cclp_0,$$

其中,sig 代表重要性传播;mrc 代表幅值细化;clp 代表清理更新.

$$C_{\max}=\mathrm{Max}(Cclp_{bp-1},Csig_{bp-2},Cmrc_{bp-2},Cclp_{bp-2},\ldots,Csig_0,Cmrc_0,Cclp_0).$$

那么,采用 Sample-Based[3]方式编码一个码块的时钟数目为

$$Total_{sb}=N\times N\times((B-1)\times 3+1).$$

采用 PPCM[6]方式编码一个码块的时钟数目为

$$Total_{ppcm}=(N\times N/4)\times B+C_{\max}+C.$$

其中,第 1 项是所有平面访问列所需的时钟数;第 2 项是所有编码过程输出上下文的上限;第 3 项是上下文输出的时间.

采用 SS+GOCS[7]方式编码一个码块的时钟数目为

$$Total_{ss+gocs}=(N\times N/4)\times B+\lambda\times(N\times N/4)\times(2\times(B-1))+C.$$

其中,第 1 项是所有平面访问列所需的时钟数;第 2 项是在幅值细化和清理更新时,由于采用列组跳过技术而剩余有上下文产生的列数.λ为比例因子,取值范围是 0–1.当λ=0 时,表示所有列都没有上下文生成;当λ=1 时,表示所有列都有上下文生成(这是两个极端情况,实际上是不会发生的);第 3 项是上下文输出的时间.

采用 BPPP 方式编码一个码块的时钟数目为

$$Total_{bppp}=(N\times N/4)+C_{\max}+C.$$

其中,第 1 项是一个比特平面访问列所需的时钟数;第 2 项是所有编码过程输出上下文的上限;第 3 项是上下文输出的时间.

根据以上分析,我们选取了 5 幅标准的测试图像,在 Windows XP 环境下,利用软件分别采用 Sample-Based, PPCM,SS+GOCS 和 BPPP 方式对一个码块所需时钟进行了实测,测试结果见表 2.

Table 2 Execution time (clock cycles) of single code block

表 2 单码块执行时钟周期数

Block size 32×32 8 bit planes	Lena (512×512)	Barbara (512×512)	Woman (2560×2048)	Bike (2560×2048)	Café (2560×2048)
Sample-Based	22 528	22 528	22 528	22 528	22 528
PPCM	10 357	11 485	9 504	11 364	11 579
SS+GOCS (λ=0.00)	9 398	10 318	8 563	10 348	10 205
SS+GOCS (λ=0.25)	10 294	11 214	9 459	11 244	11 101
SS+GOCS (λ=0.50)	11 190	12 110	10 355	12 140	11 997
SS+GOCS (λ=0.75)	12 086	13 006	11 251	13 036	12 893
SS+GOCS (λ=1.00)	12 982	13 902	12147	13 932	13 789
BPPP	8 565	9 693	7 712	9 572	9 787

从以上的实验数据可以看出,我们提出的 BPPP 结构进行 EBCOT 码块编码所需时钟数最少;对于 PPCM 和

SS+GOCS 方法,当可以跳过的列组较多(λ较小)时,SS+GOCS 优于 PPCM;相反地,当可以跳过的列组较少(λ较大)时,PPCM 优于 SS+GOCS.

5 结束语

本文提出了编码过程以及编码平面并行处理的 EBCOT 编码结构,给出了具体的硬件实现结构,并且采用 Xilinx 公司 Virtex2 系列(2v3000bg728-6)FPGA 芯片,设计实现了 BPPP 结构的 EBCOT 编码模块,系统综合频率可达 65.994MHz,验证了编码过程以及编码平面并行处理的可行性.从逻辑分析仪上实测,对于 512×512 的灰度图像,其处理速度可达 30fps,完全可以实时处理.同时,图像质量也达到了 JPEG2000 公布的结果.从而可以看出,BPPP 结构以其具有很高的并行度,成为 JPEG2000 芯片化优选的方案.

致谢 在此,我们向对本文的工作给予支持和建议的同行,尤其是西安电子科技大学图像所的老师和同学们表示感谢.

References:

[1] JPEG2000 part I final draft Int'l standard. ISO/IEC JTC1/SC29/WG1 N1890, 2000.

[2] Taubman D. High performance scalable image compression with EBCOT. IEEE Trans. on Image Processing, 2000,9(7):1158−1170.

[3] Taubman D, Ordentkich E, Weinberger M, Seroussi G. Embedded Block Coding in JPEG 2000. Technical Report, HPL-2001-35, Palo Alto: HP Labs., 2001.

[4] Andra K, Acharya T, Chakrabarti C. Efficient VLSI implementation of bit plane coder of JPEG 2000. In: Proc. of the SPIE Applications of Digital Image Processing XXIV. 2001. 246−257. http://enws155.eas.asu.edu:8001/papers.html

[5] Andra K, Chakrabarti C, Acharya T. A high-performance JPEG 2000 architecture. IEEE Trans. on Circuits and System for Video Technology, 2003,13(3):209−218.

[6] Chiang JS, Lin YS, Hsieh CY. Efficient pass-parallel architecture for EBCOT in JPEG 2000. IEEE Int. Symp. Circuits and Systems, 2002. 773−776. http://ieeexplore.ieee.org/xpl/abs_free.jsp?arNumber=1009955

[7] Lian CJ, Chen KF, Chen HH, Chen LG. Analysis and architecture design of block-coding engine for EBCOT in JPEG 2000. IEEE Trans. on Circuits and System for Video Technology, 2003,13(3):219−230.

[8] Adams MD, Reversible FK. Integer to integer wavelet transform for image compression: Performance evaluation and analysis. IEEE Trans. on Image Processing, 2000,9(6):1010−1024.

刘凯(1977−),男,陕西西安人,博士生,讲师,主要研究领域为图像编码,VLSI 设计.

吴成柯(1938−),男,教授,博士生导师,主要研究领域为计算机视觉,视频编码.

李云松(1974−),男,博士,副教授,主要研究领域为图像编码,遥感图像处理.