JPEG2000算术解码器的VLSI实现

方 晗，黄全平，周荣政，洪志良

(复旦大学专用集成电路国家重点实验室 上海 杨浦区 200433)

【摘要】介绍了JPEG2000编解码流程以及JPEG2000算术编码的原理。针对传统算术解码器过慢的情况，提出了一种动态的流水线算术解码器结构，给出了相应的硬件实现的框图，该结构通过FPGA验证。采用了TSMC 0.25 μm工艺，进行了ASIC的实现。

关 键 词 JPEG2000; 算术编码器; 流水线; MQ编码
中图分类号 TN47 文献标识码 A

VLSI Implement of JPEG2000 Arithmetic Decoder

FANG Han，HUANG Quan-ping，ZHOU Rong-zheng，HONG Zhi-liang

(ASIC & System State Key Lab, Fudan University, Yang pu Shanghai 200433)

Abstract This paper introduces JPEG2000 encode and decode flow, and also the principle of JPEG2000 arithmetic coding. Traditional arithmetic decoder is very slow that may be the bottleneck of the JPEG2000 system. To solve this problem, a pipeline arithmetic decoder is present here, together with the hardware chart. The arithmetic decoder is verified with FPGA and is implemented in TSMC 0.25 μm technology.

Key words JPEG2000; arithmetic coding; pipeline; MQ coder

文献[1]指出“ 一个码串可以看作指向某个子区间的二进制分数 ”。文献[2]把这个概念用到了对区间的连续划分中。文献[3-4]分别将其以LIFO和FIFO的形式引入。文献[5]则改进成Q编码，该算法省去了复杂的乘法运算，同时也更适合于硬件的实现[1]。目前，Q编码被应用于许多图像压缩标准中，例如JPEG、JBIG以及JPEG2000等。

JPEG2000标准采用MQ算术编码作为其压缩核心算法，MQ算术编码虽然省去了乘法运算，但是由于其算法本身的复杂性，处理速度偏慢，需要3～4个时钟周期才能完成一个比特数据的编码或者解码，成为了整个JPEG2000系统的瓶颈。因此，研究并设计高速的算术编解码器对于高速的JPEG2000图像处理系统是很有意义的。

1 PEG2000算术编码原理

JPEG2000算术编码只对两个符号进行编码，分别称为大概率符号(MPS)和小概率符号(LPS)。若Q_e是LPS所对应的概率值，则MPS概率值P_e等于$1-Q_e$。编码时算术编码器将当前区间按照MPS、LPS概率值划分为大概率区间和小概率区间。当前区间用A、C两个值表示，C为当前区间下限，A为当前区间长度，则当前区间可以表示为[C，$C+A$]。大概率区间可以表示为[$C+AQ_e$，$C+A$]，小概率区间可以表示为[C，$C+AQ_e$]。为了避免乘法，JPEG2000规定当前区间长度必须满足$0.75 \le A<1$，则AQ_e近似的等于Q_e。近似处理后大概率区间可以表示为[$C+Q_e$，$C+A$]，相应的小概率区间可以表示为[C，$C+Q_e$]。因此可以看出A、C满足以下递推关系：

MPS编码：

$$C_{i+1} = C_i + Q_e \tag{1}$$

$$A_{i+1} = A_i - Q_e \tag{2}$$

LPS编码：

$$C_{i+1} = C_i \tag{3}$$

收稿日期：2004 - 08 - 31
基金项目：国家863计划资助项目(2002 AAIZ1450)
作者简介：方 晗(1979 –)，男，硕士，主要从事图像与通信系统的专用集成电路等方面的研究.

$$A_{i+1} = Q_e \tag{4}$$

编码时，若待编码数据为大概率符号，则选择大概率区间为下一次编码的当前区间。若待编码数据为小概率符号，则选择小概率区间为下一次编码的当前区间。当前区间的任意值都可表征编码数据，JPEG2000规定编码完毕输出C作为压缩结果进行传递。解码与之类似，算术解码器根据输入的位流以及上下文关系CX，判断位流属于大概率区间还是小概率区间，从而解得相应的数据。

2　流水线结构算术解码器硬件实现

图1为JPEG2000算术解码的流程，图中前端解包程序解出压缩数据CD，提供给算术解码器。算术解码器根据位平面解码器提供的当前数据的CX，查找I表，找出当前概率索引值以及MPS值；根据当前概率索引值，算术解码器查找Q表，得到Q_e、NMPS、NLPS和SWITCH值；算术解码器根据当前区间A、C以及Q_e值判断，进行MPS解码或者LPS解码，解出当前位的数值传递给位平面解码器。在MPS和LPS解码时，有时候还需要进行重整化以及读入压缩数据；位平面解码器则根据传递过来的数据，进行运算，得到下一位数据的CX值传递给算术解码器，如此反复，直到所有数据解码完毕。

由于在解包时，解包程序处理数据的能力远远快于算术解码器，为了减少解包程序的等待时间，在解包程序与算术解码器之间。采用一个FIFO作为缓冲存储器，算术解码器与位平面解码器之间是实时交换，一个工作时，另一个模块必须等待，因此没有使用缓冲器而是使用握手信号进行控制；I表采用19*7的RAM实现，Q表采用47*29的ROM实现，得到的算术解码器硬件结构如图1所示。

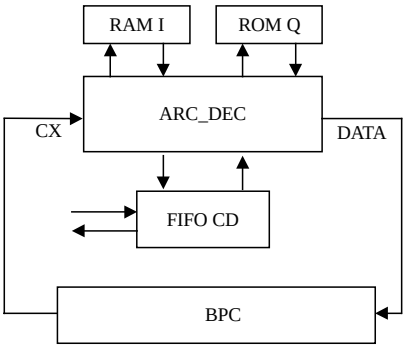

图1　算术解码器硬件结构

传统的算术解码器在解码时，要依次经过读I表、读Q表、MPS/LPS编码、写I表四个阶段，此外算术解码器还需时刻监视寄存器A的内容，若其跳出范围之外，则必须进行重整化，这样一个比特数据解码出来需要4～5个时钟周期。为了提高解码器处理数据的能力，流水线是一种行之有效的方法。由于处理每个比特数据的时间是不相同的，为了避免不必要的空闲等待，本文使用了一种动态流水线结构的算术解码器。

对JPEG2000的解码流程做进一步的详细分析，可以发现在位平面解码器工作时，算术解码器处于等待状态，反之算术解码器工作时，位平面解码器则处于等待状态。若把位平面解码器工作的这一段时间利用起来，显然可以有效地提高解码速度。在算术解码流程中，从位平面解码器传递CX到来开始，算术解码器依次经过读I表、Q表、MPS/LPS解码、写I表、重整化五个过程。而解码数据实际在MPS/LPS过程中就已经得到了，可见写I表和重整化两个进程完全可以利用位平面解码器工作的这一段时间进行。

若对算术解码的流程做进一步的详细分析，可以发现，读写I表、Q表只对RAM I和ROM Q进行操作，重整化则只对寄存器A、C和CT进行操作，它们完全可以同时进行。实际上，重整化所需要的时间可能比较长，当位平面解码完毕后，重整化可能还未完成。这时为了提高解码速度，可以使重整化和读I、Q表两个进程同时进行。在读Q表结束后，算术解码器查询重整化是否完成，若已经完成，则进行MPS/LPS解码，反之则等待重整化完成以后再进行MPS/LPS解码。

最终得到的流水线算术解码器工作流程如图2所示。若在同一竖轴上同时有若干个进程则表示这些进程是同时并行工作的。采用这种流水线结构，可以使费时最大的重整化进程几乎不占用额外的系统时间。实际上，只在极少数情况下，重整化所需时间会大于位平面解码、读I表、读Q表三个进程的时间总和，改进后的算术解码器速度提高接近50%。

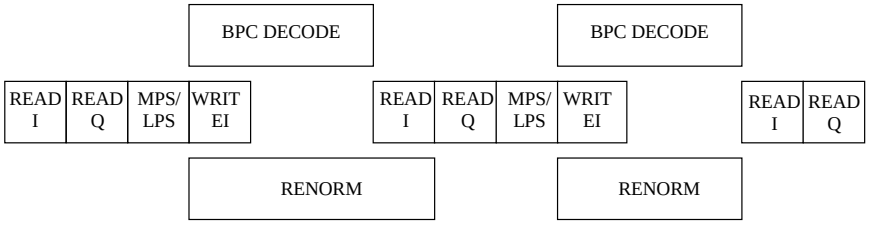

图2　流水线算术解码器流程

在JPEG2000算术编解码中，最关键的问题是重整化和部分输出、位填充，以及动态概率估计。

2.1 位填充

图3为位填充算法用于解决重整化和部分输出过程中所引入的另一新问题：进位翻转问题。基本方法为：设置一个字长为8 bit的缓冲寄存器B，用以保存最近的8 bit码字。其中，B为辅助寄存器，C寄存器第27位c为进位。位填充方法为：

(1) 若B≠0xFF且进位c=0

进位c=0说明没有进位，则将B寄存器中数据输出，将C中第26～19位移到B中，置ct=8，说明下一次在移位8次后输出。

(2) 若B=0xFF

将B寄存器中数据输出，将C中包括进位的第27～20位数据转移到B中，置ct=7，说明下一次在移位7次后输出，以致不遗漏本次输出剩下的第19位。

(3) 若B≠0xFF且进位c=1

将进位c加到B中，然后根据B的值进行判断。若B≠0xFF，则根据规则(1)输出；若B=0xFF，则根据规则(2)输出。

解码时，解码器将检测所有的数据为0xFF的字节后面的第一位，如果这一位是1，则解码器知道这里产生了一个进位，需要超前一位相加。

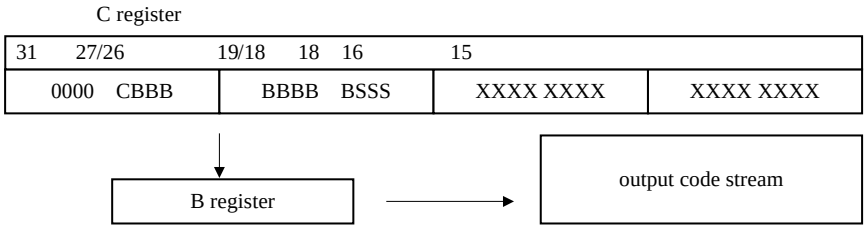

图3 位填充寄存器及辅助寄存器

2.2 动态概率估计

在JPEG2000中，为了达到更好的压缩比，MPS和LPS所对应的概率值是动态更新的。算术编码器根据当前输入CX，查表得到当前概率值索引I和当前MPS，然后再根据I值查Q表得到Q_e值以及表征下一索引的NLPS以及NMPS值，此外Q表中还包括一标志是否交换当前MPS和LPS值的标志SWITCH。若当前输入数据为MPS，则将当该CX对应的索引寄存器值更新为NMPS，反之则更新为NLPS。若SWITCH为1 ，则交换MPS和LPS编码中的值。

在硬件实现时，第一次查表用一个小RAM I来实现。因为共有19种CX和47种概率索引值，所以该RAM的大小为19*7，其中每个地址中前6位存储该CX对应的概率值索引，最后一位为MPS值。

因为Q表中存储的值是不变的，所以第二次查表用一个ROM Q实现。该ROM大小为47*29，每个地址中前16位为Q_e值，NMPS、NLPS各为1位，SWITCH为1位。RAM I和ROM Q结构如图4所示。

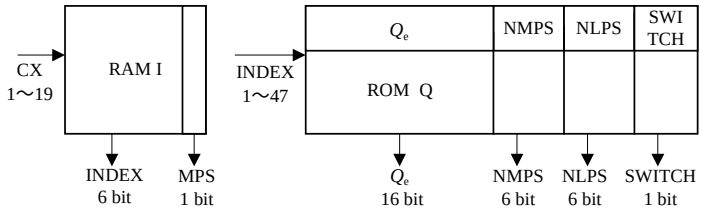

图4 RAM I和ROM Q结构

3 仿真及测试结果

采用Verilog硬件描述语言对此算术解码器进行行为级描述，并采用Xilinx Foundation进行布局布线，最高工作频率为73 MHz。整个JPEG2000编解码流程经FPGA验证，其结果与软件仿真结果一致，从而也间接

证明了该结构的可行性。

对该JPEG2000行为级模型自动综合和布局布线，并采用TSMC0.25 μm工艺进行了流片，该JPEG2000专用处理芯片的面积为3.7 mm×3.3 mm，最高工作频率为48 MHz。测试结果表明算术解码器能够完成的解码功能，结果与软件仿真和FPGA测试结果一致。验证结果表明该结构在功能上的正确性，可以大幅度提高算术解码器解码速度，能满足JPEG2000系统要求。

参　考　文　献

[1] Shannon C E. A mathematical theory of communication[J]. Bell Syst. Tech. J., 1948, 27: 379-423, 623-656.
[2] Eliasp P. Information theory and coding[M]. New York: McGraw-Hill, 1963.
[3] Rissanen J. Generalized Kraft inequality and arithmetic coding[J]. IBM J. Res. Devel., 1976, 20: 198-203.
[4] Pasco R C. Source coding algorithms for fast data compression[D]. California: Standford Univ., 1976.
[5] Pennebaker W B, Mitchell J L, Langdon G G, et al. An overview of the basic principle of the Q-coder adaptive binary arithmetic coder[J]. IBM J. Res. Devel., 1988, 32(6): 717-726.

编　辑　刘文珍

(上接第919页)

4　讨论与总结

相对于以前的一些量子密钥分配协议，例如BB84协议等，本协议的特点是高效性。在BB84协议里[1]，寻求建立共享的密钥的双方必须各自随机选择测量基$\{|0>,|1>\}$或者$\{\frac{1}{\sqrt{2}}(|0>+|1>),\ \frac{1}{\sqrt{2}}(|0>-|1>)\}$来测量手中的量子位。只有双方恰巧选择了相同的测量基的前提下，它们才能得到相同的测量结果，进而建立起一个共享的密钥位。平均来说，双方选择相同测量基的概率为50%，因此，只有一半的量子位能够对密钥的产生有所贡献。换句话说，BB84协议的效率最大为50%，因此，BB84协议的效率是很低的。而高效量子密钥分配协议中，不存在测量基的随机选择，双方的测量结果都是唯一对应的，除了用来检错的量子位之外，所有其他量子位都对密钥有所贡献。因此，高效量子密钥分配协议是相对高效的。

本文提出了一个建立在纠缠态内部关联基础上的量子密钥分配协议。通信双方通过交换量子位和贝尔测量来建立起共享的密钥。除了用作检错的部分之外，所有的EPR对都对生成密钥有贡献，因此它是高效的。本文还证明了在可能的攻击下，它是安全的。

参　考　文　献

[1] Bennet C H, Brassard G. Quantum cryptography: Public-key distribution and tossing[C]// IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, 1984.
[2] Ekert A K. Quantum cryptography based on Bell's theorem[J]. Physical Review Letters, 1991, 67: 661-663.
[3] Bennttt C H, Brassard G, Mermin N D. Quantum cryptography without Bell's theorem[J]. Physical Review Letters, 1992, 68: 557-559.
[4] Bennett C H. Quantum cryptography using any two nonorthogonal states[J]. Physical Review Letters, 1992, 68: 3121-3124.
[5] Huttner B, Imoto N, Gisin N,et al. Quantum cryptography with coherent states[J]. Physical Review A, 1995，51: 1863-1869.
[6] Goldenberg L, Vaidman L. Quantum cryptography based on orthogonal states[J]. Physical Review Letters, 1995, 75: 1239-1243
[7] Cabello A. Quantum key distribution in the holevo limit[J]. Physical Review Letters, 2000, 85: 5635-5638.
[8] Xue P, Li C F, Guo G C. Conditional efficient multiuser quantum cryptography network[J]. Physical Review A, 2002, 65: 022317.
[9] Long G L, Liu L S. General scheme for superdense coding between multiparties[J]. Physical Review A, 2002, 65: 032305.
[10] Kimura T, Nambu Y. Single-photon interference over 150km transmission using silica-based integrated-optic interferometers for quantum cryptography[J/OL]. http://www. eprints: quant-ph, 2006-06-18.
[11] Kim Y H, Kulik S P, Shih Y. Quantum teleportation of a polarization state with a complete bell state measurement[J]. Physical Review Letters, 2000, 86: 1370-1373.
[12] Cinelli C, Barbieri M, Martini F D, et al. Realization of hyperentangled two-photon states[J]. International Journal of Laser Physics, 2005, 15(1): 124-128.

编　辑　熊思亮