

# 基于FPGA的JPEG2000自适应算术编码器设计

陈 玮, 杨名利

(西南交通大学 信息科学与技术学院, 四川 成都 610031)

**摘 要:**文中研究 JPEG2000 标准中自适应算术编码器的硬件实现问题,采用并行结构的 FPGA 设计,并用 Modelsim5.8 对其作仿真验证。设计使用 VerilogHDL 语言在 RTL 级描述,并以 Xilinx VertexII 系列中的 xc2v250-6fg256 器件为基础在 ISE6.1 下完成综合。

**关键词:**JPEG2000;算术编码器;FPGA;图像压缩

**中图分类号:**TN919.81;TP391.41

**文献标识码:**A

**文章编号:**1673-629X(2006)10-0211-03

## Design of JPEG2000 Arithmetic Encoder Based on FPGA

CHEN Wei, YANG Ming-li

(School of Information Science & Technology, Southwest Jiaotong University, Chengdu 610031, China)

**Abstract:** FPGA implementation of the adaptive arithmetic encoder in JPEG2000 standard is investigated. A parallel architecture is presented, which is simulated and verified with Modelsim5.8 on FPGA. The design is described with VerilogHDL at RTL level. Based on Xilinx VertexII xc2v250-6fg256, synthesis are conducted with ISE6.1.

**Key words:** JPEG2000; arithmetic encoder; FPGA; image compression

### 0 引 言

在新一代静止图像压缩标准 JPEG2000 中,提高图像压缩性能的关键技术之一就是基于上下文的自适应算术编码。与其它熵编码方法相比,算术编码能够取得更好的压缩效果。JPEG2000 在算术编码的设计中运用了很多技巧。文中就是在对算术编码进行具体的分析后,结合静态图像实时处理的应用,综合考虑前级小波系数位平面编码和自适应算术编码器互连关系<sup>[1]</sup>,提出了一种自适应算术编码器的 FPGA 并行设计思想,并对其进行仿真验证。

### 1 自适应算术编码器算法

JPEG2000 自适应算术编码模型如图 1 所示,输入信号是 D 和 CX,其中 D 表示当前待编码的符号,为二进制:0 或 1;CX 表示上下文,共有 19 个上下文(0~18),它和数据对是由前一级的系数位模型产生的;CD 表示压缩后的数据。



图1 算术编码器输入输出模型

递归编码间隔细分<sup>[2]</sup>是自适应算术编码的基础,在编码的过程中,输入的二进制数据流 0,1 被分为大概率符号

(MPS)和小概率符号(LPS)。概率区间被划分为 MPS 的编码间隔和 LPS 的编码间隔,其长度由每个信源符号的概率决定。自适应算术编码器采用一个可以对原始数据快速适应的概率自动估计模型表<sup>[3]</sup>,共有 47 项。LPS 的编码间隔应该总是小于 MPS 的编码间隔。每当 LPS 区间比 MPS 区间大时就交换 MPS 和 LPS 区间;当编码 MPS 时,就把 LPS 子区间加到码流上;当编码 LPS 时,码流不变。

编码时设置两个专用寄存器 A 和 C, A 寄存器中的数值为子区间的宽度, C 寄存器中的数值为子区间的起始位置,用  $[C, C+A]$  表示它的编码区间。

为了有利于硬件实现,因而采用固定精度的整数运算做编码操作,且用整数代表分数值,即 0x8000 相当于小数 0.75。区间 A 保持在  $0.75 \leq A < 1.50$  的范围内,当整数值降到小于 0x8000 时,通过重整化把它加倍。重整化发生时,调用概率估计模型,为当前正被编码的上下文确定一个新的概率估计。

假定当前区间是 A, LPS 概率的当前估计值是  $Q_e$ , 为了提高编码器运算速度,对子区间进行了近似处理: A -  $Q_e$  为 MPS 的概率子区间;  $Q_e$  为 LPS 的概率子区间。编码过程中有关寄存器的结构如表 1 所示。

A 寄存器中的 a 位是小数位,码寄存器 C 中的 x 位是小数位, s 位是分隔位, b 位指明从 C 寄存器移出的整个字节原来的位置, c 位是进位位。

每当编码一个大概率符号时,有  $C = C + Q_e$ , 即小概

收稿日期:2005-12-20

作者简介:陈 玮(1978-),男,贵州铜仁人,硕士研究生,研究方向为图像处理;杨名利,教授,研究方向为自动控制。

率符号的概率要加到码字 C 上,这就可能在 C 中产生一个进位。为了保持 A 和 C 有统一的尺度,每次 A 寄存器中的值左移一位,C 寄存器中的值也左移一位。因为 C 寄存器中存储的是编码的结果,这就需要有一个缓冲区来保存从 C 寄存器中移出的数据位,这里使用了一个位填充寄存器 B 保存 C 寄存器的高位。每当 C 寄存器满时,就从 B 寄存器把一个字节的数据输出到外面的压缩数据流中。

表 1 寄存器 A, C 的结构

	MSB			LSB
寄存器	0000	0000	aaaa	aaaa
A	0000	0000	aaaa	aaaa
寄存器	0000	bbbb	xxxx	xxxx
C	cbbb	bsss	xxxx	xxxx

## 2 自适应算术编码器 FPGA 的设计

### 2.1 自适应算术编码器框图

自适应算术编码器框图如图 2 所示。

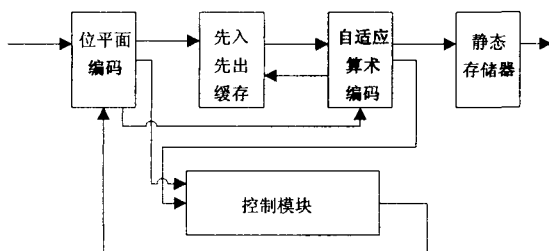


图 2 自适应算术编码器功能框图

控制模块的控制下,位平面编码模块读入小波变换后的系数,并对系数进行位编码,编码后的系数写入 FIFO,通过检测 FIFO 的缓存为空的指示信号 empty 变为 0;自适应算术编码器知道 FIFO 中有数据,于是给 FIFO 发出读数据信号,读入数据,对位平面编码数据进行编码,编码后的数据写入静态外存 RAM 中。在小波系数位平面编码模块与自适应算术编码模块之间采用 FIFO,通过 FIFO 缓存,两个模块之间可以间接通信。如果 FIFO 为空,但位平面编码没有结束,自适应编码模块要等待,直到 FIFO 有数据;另一方面,如果 FIFO 满了,位平面暂停编码,等到自适应编码模块从 FIFO 中读出了数据,FIFO 不满之后,位平面编码模块才继续编码,这样两个模块之间几乎是并行工作。

### 2.2 概率估计存储结构的设计

根据 JPEG2000 协议<sup>[4]</sup>,LPS 的概率  $Q_e$  是由上下文对应的索引值决定的,如表 2 所示。在自适应编码初始化时,要对一些特定的上下文赋索引初值。随着编码的进行,同一个值的上下文对应的索引值( $I(CX)$ )不是一成不变的,某些上下文对应的索引值要不断地用表 1 当前索引值对应的下一个较小概率符号索引(NLPS)或下一个较大概率符号索引(NMPS)的值来更新<sup>[5]</sup>。用可以读写的 FPGA 片内 RAM 保存对应每个上下文的  $I(CX)$  值,该 RAM 由全局时钟同步,地址线为 6 位,地址值为上下文值,输入

输出都为  $I(CX)$  的值,在系统工作时,要对该 RAM 进行初始化,初始化时在地址 0 单元写入 4;地址 17 写入 3;地址 18 写入 46;其它地址写入 0。

由 JPEG2000 协议知道,由于  $Q_e$ , NMPS, NLPS 以及 SWITCH(交换信号)的地址是相同的<sup>[6]</sup>,因此,可以使用一片宽总线的片内 ROM 来实现。其中,数据总线宽度为上述 4 种数据的组合,为 28 位:  $dout(28) = Q_e(15) + NMPS(6) + NLPS(6) + SWITCH(1)$

表 2 概率估计表

Index	Qe 值		NMPS	NLPS	SWITCH
	十六进制	十进制			
0	0x5601	0.503937	1	1	1
1	0x3401	0.304715	2	6	0
...					
45	0x0001	0.000023	45	43	0

地址总线为 6 位,识别 47 种不同状态。ROM 在时钟同步信号作用下,如果使能信号有效,就读出对应的  $dout$  值。由于 RAM 和 ROM 的地址线都为 6 位,两者都可以存储 64 个数据,但实际上只用到了 47 个数据,地址值大于 46 以上的存储单元初始化时置 0。

### 2.3 自适应算术编码器状态机的设计

自适应算术编码器是功能、时序复杂的时序逻辑电路。为了便于利用硬件描述语言进行设计,将这一复杂的时序逻辑抽象成有限状态机,并利用有限状态机实现复杂的设计。根据自适应算术编码协议,可以得到 FPGA 设计的状态机,将数据从读入到判断其是大概率符号还是小概率符号的过程与其自适应编码过程分开(如图 3 所示),两者并行执行,有利于提高系统的运行速度。

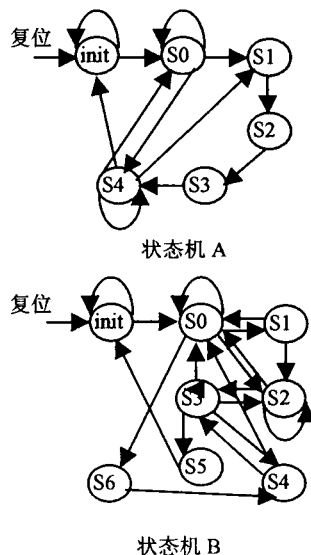


图 3 自适应算术编码器状态机

A 状态机 init 状态只对 MPS(CX) 信号、start\_sig、flush、ready 清零。start\_sig、flush、ready 用来控制状态机 B, ready 为 1 表示数据已准备好。初始化完成后,如果系数位平面编码模块发来了开始编码的指示(start = 1),并且 FIFO 不空,则向 FIFO 发出读数据请求,开始编码。之后,状态机进入 S0 状态。在 S0 状态,判断在发出一个读

信号后, FIFO 是否为空, 有 3 种结果:

①如果为空, 但小波系数位平面编码模块传来的 bitmodel\_over 信号为低电平, 说明编码还没有结束, 只是 FIFO 还没有数据, 于是继续向 FIFO 发读信号, 状态机下一个状态仍然处在 S0;

②如果 FIFO 为空, bitmodel\_over 信号为高电平, 说明编码结束, 则下一个状态转向 S4, 并置位 flush;

③如果 FIFO 不空, 则转到 S1 状态, 对数据进行处理。

这里把 JPEG2000 中的判别条件进行了归纳整理, 化简为条件: D 不等于 MPS(CX) 是否成立, 成立为小概率符号, 否则为大概率符号。若为小概率符号, 且 SWITCH(CX) = 1, 则将 MPS(CX) 值取反。设计中采用同步 ROM, 从地址和使能信号输入到数据输出的延迟大于一个周期, 所以需要 S1, S2, S3 状态完成以上工作。在 S0 状态若判断该次自适应编码结束, 在 S3 判断一个数据是大概率还是小概率符号后, 置为相应的大小概率标识信号 (mps), 并使 start\_sig = 1, 数据准备好信号 ready = 1 (如图 4 所示)。如果 FIFO 不空, 向 FIFO 发读信号, 读取下一个输出数据, 同时转到状态 S4。在 S4 循环等待状态机 B 的接收应答信号 receive。若 receive = 1, 则置 ready 为 0, 结束循环。如果 S3 状态已发过读 FIFO 数据信号, 则从 S4 转到状态 S1, 否则, 转 S0 并发读 FIFO 信号, 此外, 如果此时状态机 B 传来了块编码结束信号, 则转 init 状态, 等待下一个数据块的到来。

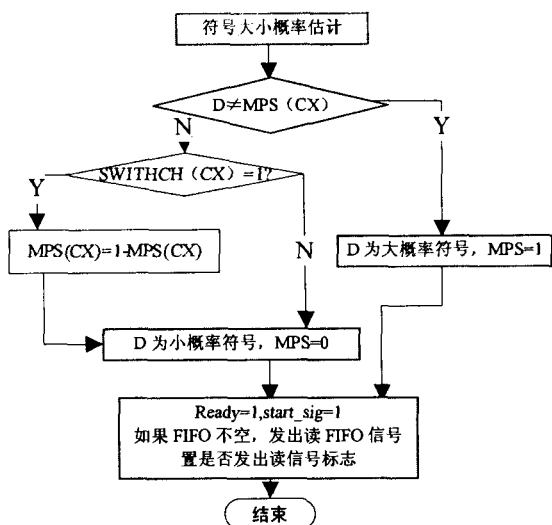


图 4 较大较小概率估计流程

状态机 B 在 init 状态对计数器 CT 和 A, C 寄存器及片内 RAM 初始化, 如果 start\_sig = 1, 则状态机 B 转到 S0 开始编码, 在 S0 状态, 若收到 flush = 1 信号后转 S6 状态; 否则, 接收到 ready 有效信号后发应答信号 receive, ready 无效仍然在 S0 状态等待 ready 信号。若 mps = 0, 在 S0 进行较小概率符号编码, 否则转 S1 进行较大概率符号编码。

在 S2 状态开始进行区间重整化。重整化时, A 寄存器和 C 寄存器中的值同时左移, 直到 A 寄存器中的值不

小于 0x8000。CT 记录移位数, 当 CT = 0, 输出一个字节数据, 具体过程见文献[2]中图 C-8。显然, 该模块中存在一个循环操作, 循环操作次数是不确定的, 此外还要在重整化操作中, 调用字节输出模块 BYTEOUT, 该模块的功能不能在一个时钟中完成。和小波系数位平面编码统一考虑, 为了提高整个 JPEG2000 编码器时钟运行频率, 采用流水结构, 让该模块在 2 个时钟周期(状态 S2, S3)中完成, 在状态 S2 完成寄存器 A 和 C 的移位以及计数器 CT 减 1 的操作。如果计数器 CT 不等于 0, 且寄存器 A 中的值小于 0x8000, 则下个状态转到 S2, 否则转到 S0, 并发出读 FIFO 信号。如果计数器 CT 等于 0, 下一状态转到 S3。在 S3 状态完成编码字节输出(BYTEOUT)操作后, 若寄存器 A 的第 15 位为 0, 再转向 S2; 否则转 S0 并发出读 FIFO 信号。

根据编码情况在状态 S6 完成寄存器 C 重新设置操作(SETBITS)。寄存器 C 移位是在状态 S4 完成的, 移位完成后转到状态 S3, 进行编码字节输出后又回到状态 S4, 移位寄存器 C 中余下的数据在状态 S3 再次输出编码字节。每次编码模块调用后, 输出的总是寄存器 B 原来寄存的数, 之后寄存器 C 又对寄存器 B 进行比特填充, 同时根据情况对 CT 重新赋值为 7 或 8。在状态 S5 完成编码器清空, 如果 B 寄存器中是非 0xFF 数据, 则将其输出, 否则将数据抛弃。

整个数据块编码完成后, 编码器回到初始状态, 等待下一个数据块的到来。

#### 2.4 减少设计面积——资源共享

协议中, 区间重整化、编码器清空 3 次调用编码字节输出(BYTEOUT)模块。工作方式采用资源共享设计思想, 用状态机来分析和设计整个流程, 编码字节输出只在一个状态机中完成, 供其它模块调用, 大大节约了设计的面积。

### 3 电路的模拟与验证

采用 VerilogHDL 在 RTL 级对 JPEG2000 自适应算术编码器进行描述, 并以 xc2v250-6fg256 为基础, 在 ISE6.1 下分别完成设计综合, 综合前后用 Modelsim5.8 模拟, 在 JPEG2000 编码系统设计的背景下, 最高工作频率达 66MHz。其中并行工作方式自适应算术编码器速度约为 0.25bit/cycle。综合结果在经布线后进行后模拟, 得到的结果与 C 语言的软件结果及前仿真结果一致。

### 4 结 论

在 JPEG2000 编码系统设计的背景下, 针对自适应算术编码器, 为了进一步提高运算速率, 提出了硬件设计并行方式工作方案。该方案采用了流水线设计思想以提高系统时钟的工作频率, 采用资源共享思想, 节约了设计面积, 并且在 FPGA 上得以实现。结果不仅验证了该设计在

(下转第 216 页)

对试验进度按计划单编号、材料牌号、任务来源、下达日期结合计划单状态进行网上检索,对试验项目表信息进行网上检索。

### 3 系统特点

#### 3.1 系统结构特点

在整个开发过程中,根据 workflow 管理的内涵,以过程为中心,结合底层实现技术确定了计划的执行的内在逻辑顺序和触发条件以及计划状态的转换机制。在系统功能模块的划分上,采用了结构化程序设计思想,具备 5 大特点:可读性,即每个模块的功能明确,模块间关系简单清晰;独立性,即每个模块内部的修改尽可能不影响其他模块;可验证性,即每个模块的正确性容易验证;可扩充性,即模块有良好的接口,容易扩充;可维护性,即每个模块相对独立、便于修改和维护。在数据库应用软件编码时,采用 C/S 和 B/S 两种模式相结合的开发方法以及面向对象的程序设计,考虑了软件的易用性、程序运行的效率和稳定性。实现了测试数据信息的发送、接收及反馈成完整的闭环流动,体现了网络化的优势。在数据库设计时,使各数据表符合关系数据库第三范式的原则,以减少数据冗余,提高访问效率。

#### 3.2 系统功能特点

系统主要功能包括试验计划进度的管理,试验计划的接收,试验数据的录入、编辑、查询及试验报告的输出,工作模式符合当前工作流程,过程明确、数据流清晰闭环流动且运行稳定。并具有三大优点:其一,由于采用了模块化设计方法使系统具备独立性强、维护方便的特点;其二,本着实用的思想,试验项目表的信息包括试验条件、试样根数、指标值,实现了一次录入,全程使用,减轻了工作量,界面设计合理、直观、易操作;其三,系统具有一定的逻辑纠错能力,提高了信息的准确率。本系统通过用户的分级操作权限的安全机制来保证数据存放的安全。为了增强系统的柔度,数据库和应用软件的设计时充分考虑了可扩展性,预留了相应的接口,可以和其它系统连接。

#### 3.3 运用的编程技术特点

在实现本系统时,本着实用、易用的思想,着眼于界面

的直观性和用户的易操作性。系统采用面向对象的程序设计方法和分布式结构交互层(用户界面)、业务逻辑层、数据层。在数据层,把数据访问逻辑封装在类中,运用了类的继承、重载、共享成员等技术,并大量使用了 ADO.NET 中的数据适配器和数据集、数据表(数据行和数据列的集合)、数据视图,来对数据记录的信息进行离线操作和动态生成统计分析表。其中数据视图其内容、排序和成员关系实时反映基础数据表,基于筛选器表达式来筛选数据,它是数据绑定的理想选择。在交互层使用多文档界面,大量使用了 DataGrid 网格控件、选项卡,其中 DataGrid 网格控件是一个多列数据绑定网格,数据源通常为数据表及数据视图,在这个网格里具有对数据源填充的数据进行浏览、维护和对数据库数据物理更新功能并可以定制显示的样式和列的顺序,由于这些控件的运用使得从而使系统具备了实用、方便、稳定的特点。并且在程序中运用了 .NET 的 Exception 类的出错捕获机制,来保证数据的入库的正确性、系统的友好性。

### 4 结 论

本系统是根据国内航空材料研究和应用需求的新形势,及航空材料数据应用的具体情况和特点,参考国际上成功的数据库系统,采用了主流计算机软、硬件平台和开发工具,建立的面向航空工业材料性能试验室数据库系统。它结合了试验室 workflow,采用网络数据库平台。经测试,系统运行稳定。

#### 参考文献:

- [1] 王兴鹏. 工作流管理系统与 ERP 系统的集成应用[J]. 微机发展, 2003, 13(12): 124 - 126.
- [2] Goode C, Kauffman J. ASP.NET 1.0 入门经典[M]. 北京: 清华大学出版社, 2002.
- [3] Riordan R M. ADO.NET 应用开发指南[M]. 北京: 清华大学出版社, 2002.
- [4] Anderson B, Francis B. ASP.NET 高级编程[M]. 北京: 清华大学出版社, 2002.
- [5] 刘猛玉, 张开春. Visual Basic .NET 数据库开发教程[M]. 北京: 清华大学出版社, 2003.
- [6] Langodin G, Rissanen J. Compression of Black/White Images with Arithmetic Coding[J]. IEEE Transactions on Communications, 1981, 29(6): 591 - 598.
- [7] Boliek M, Christopoulos C, Majani E. JPEG 2000 part 1 final draft international standard[S]. 2000.
- [8] Witan H, Neal R M, Clearly J G. An Arithmetic IC Coding for Data Compression[J]. Communications of the ACM, 1987, 30(6): 5202 - 5401.
- [9] ISO/IEC. ISO/IEC 15444 - 1. Information technology J PEG 2000 image coding system system - Part 1: Core coding system[S].
- [10] Taubman D. High Performance Scalable Image Compression with EBCOT[A]. Proc IEEE Int Conference on Image processing (ICIP) [C]. Japan: IEEE, 1999. 344 - 348.
- [11] Howard P G, Vitter J S. Arithmetic Coding for Data Compression[J]. Proc IEEE, 1994, 82(5): 857 - 865.

(上接第 213 页)

功能上的正确性,同时表明该设计在速度和规模上都能够适应 JPEG2000 编码系统的需要。

#### 参考文献:

- [1] Langodin G, Rissanen J. Compression of Black/White Images with Arithmetic Coding[J]. IEEE Transactions on Communications, 1981, 29(6): 591 - 598.
- [2] Boliek M, Christopoulos C, Majani E. JPEG 2000 part 1 final draft international standard[S]. 2000.
- [3] Witan H, Neal R M, Clearly J G. An Arithmetic IC Coding for