

高吞吐率的高效视频编码熵编码 并行硬件架构设计

周小朋, 梁峰, 李冰

(西安交通大学微电子学院, 710049, 西安)

摘要: 针对超高清视频编码的实时性需求以及高效视频编码熵编码系统中存在的数据吞吐率瓶颈,提出了一种基于波前并行处理(WPP)技术的硬件架构,在提高编码并行度的同时保留了行与行的依赖性,提升了编码系统的数据吞吐率与压缩效率。该架构可以进行多线程编码树单元同步编码,提高了近3倍的编码并行性;采用同步更新单元保证线程之间概率模型的同步更新;单线程内的概率模型支持多个二元的符号的并行更新;初始化单元支持对多种模式下编码单元的概率初始化;支持在线模式和离线模式,可以更好地平衡熵编码与其他模块的吞吐差异。本设计实现的熵编码器单周期处理的平均数据量为5 b,在SMIC 55 nm工艺下,其综合频率达到300 MHz,吞吐率为1 512 Mb/s,与参考设计相比,数据吞吐率提升了89%。本设计提出的熵编码器性能满足超高清视频的实时编码需求。

关键词: 高效视频编码;熵编码;波前并行处理

中图分类号: TP302 **文献标志码:** A

DOI: 10.7652/xjtuxb202007021 **文章编号:** 0253-987X(2020)07-0180-07



OSID 码

A High-Throughput Parallel Very-Large-Scale Integration Architecture of Entropy Coding in High Efficiency Video Coding

ZHOU Xiaopeng, LIANG Feng, LI Bing

(School of Microelectronics, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: Due to the real-time requirements of ultra-high-definition video encoding and the data throughput bottleneck in the entropy encoding system, a hardware architecture based on wavefront parallel processing (WPP) technology is proposed, which improves the parallelism of coding while retaining row-to-row dependency, and provides higher throughput and compression efficiency. This architecture allows three-threaded coding tree units to be encoded simultaneously, thus improves the parallelism by nearly three times. The synchronous update unit ensures the probability models between threads can be updated simultaneously. The probability models in a single thread support parallel updating of multiple binary symbols. Initialization unit can initialize the coding units in multiple modes. Online mode and offline mode are both supported, which can balance the throughput between entropy encoder and other modules. This entropy encoder delivers an average of 5 b per cycle. The design is synthesized in SMIC 55 nm and it can work at a speed of 300 MHz. And the throughput is 1 512 Mb/s, which is 89% higher than that of the reference design. The result shows that the performance of the

entropy encoder proposed in this design meets the real-time encoding requirements of ultra-high-definition video.

Keywords: high-efficiency video coding; entropy encoding; wavefront parallel processing

随着视频技术和应用的发展,尤其是高清、超高清、3D 等视频技术的兴起,视频流信息已深入到人们工作和生活的各个方面。海量的视频传输需求对网络带宽以及传输能力都是极大的挑战,因此,实现视频信息的有效压缩是解决这一矛盾的重要技术途径之一^[1]。

高效视频压缩(HEVC)标准是由视频编码联合组在 2013 年推出的视频编解码标准^[2]。HEVC 的编码原理和前一代视频编解码标准 H.264/AVC 基本一致,在保持预测编码与变换编码的混合编码框架的前提下引入了诸多新的技术来改善编码效率,包括针对不同应用场合的图像组(GOP)编码方式、条和片的划分、多种预测模式、基于四叉树单元的块分割结构、自适应的运动矢量补偿、可变尺寸的离散余弦变换等。相比于以往的视频编解码标准,HEVC 标准在压缩率、图像质量、并行处理等方面有着无可比拟的优势。

HEVC 编码过程主要包括帧内编码、帧间编码、变换量化、滤波、自适应补偿和熵编码等,其编码框架如图 1 所示。

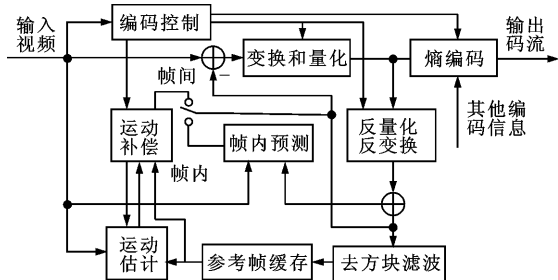


图 1 HEVC 编码框架

不考虑信道传输的情况下,熵编码处于整个视频压缩编码系统的最后一级。熵编码基于算术编码算法实现,其在编码过程中引入的数据依赖严重地限制了硬件设计的并行化程度,使得熵编码成为整个编码器的吞吐率瓶颈所在^[3]。

针对熵编码存在的瓶颈,众多学者进行了大量研究工作,提出了高吞吐率的硬件设计架构和优化方案,使得单个熵编码器的吞吐率有了较大提升。文献[3]针对熵编码的数据依赖性进行了研究,提出了预归一化、双状态转换等技术,降低了关键路径延时;文献[4]提出了几种提高熵编码吞吐率的方案;

文献[5]采用了一种四级流水的二进制算术编码结构,提高了编码器吞吐率;文献[6]提出了一种按位预先取反技术,降低了算术编码过程中区间更新的路径延时,提高了设计的工作频率和吞吐率;文献[7]采用静态随机存取存储器(SRAM)构建上下文存储器,在降低一定性能的基础上减小了芯片的面积,考虑到性能与面积的折衷,在不影响编码性能的前提下,采用 SRAM 代替寄存器可以降低成本;文献[8]采用门控时钟等低功耗技术,将功耗降低了 25% 以上,可以采用文献[8]中的低功耗技术来降低设计功耗。

考虑到超高清视频编码的巨大运算量,HEVC 提供了基于条和片的编解码处理机制^[1],将图像划分为多个可以独立熵编码的单元,可以同时处理多个条或片来提升吞吐率。文献[9]采用两个熵编码器来满足 4K 超高清视频的编码需求。多个熵编码器固然会带来更大的面积开销,但其所带来的吞吐率的提升是单个熵编码器不能比拟的。波前并行处理(WPP)技术允许多行编码单元同时进行编码,与基于条片的并行处理机制相比,WPP 技术在编码的同时不需要打破编码树单元(CTU)直接预测的依赖性,保证了行与行之间上下文信息的连贯。针对熵编码硬件设计中由于存在数据依赖而导致的吞吐率瓶颈,本文提出了一种基于波前并行处理技术的硬件架构,并进行了硬件实现与性能评估。

1 HEVC 熵编码概述

熵编码是按照信息熵原理进行的无损编码方式,HEVC 标准中采用基于上下文的自适应算术编码(CABAC)作为熵编码的算法,通过统计输入符号的变化特性,建立符号的上下文分布模型,利用图像块与块之间的相关性消除符号之间的冗余,实现对视频信息的压缩^[4]。熵编码主要包括语法元素(SE)的二值化、上下文建模和二进制算术编码 3 个步骤^[2]。CABAC 将帧内、帧间预测信息以及变换量化后的残差信息编码为二进制字符串,并为每一个二进制字符建立概率模型表,根据不同字符对应的上下文索引,在概率模型表中读取概率状态值进行后续的二进制算术编码,最后将算术编码过程中移位产生的比特流信息以字节的形式打包输出。

1.1 熵编码过程

CABAC 的编码流程如图 2 所示^[1],其输入为原始输入视频经过预测变换后产生的语法元素,输出为经过编码压缩的比特流信息。二值化模块将非二进制的语法元素转换为二进制字符串,随后进行算术编码。算术编码有常规编码、旁路编码两种模式。常规编码模式包括上下文建模和二进制算术编码。在算术编码之前,二进制符号进入上下文建模模块并根据输入符号的上下文索引值选择与其对应的概率模型,模型的选择依赖于先前已经编码的语法元素。确定概率模型后,进入二进制算术编码模块,并根据编码的结果对相应的上下文模型进行更新。旁路编码模式可以视为常规编码过程的简化。在该模式下,输入的二进制符号 0 和 1 为等概率分布,与常规编码模式相比,旁路编码具有更快的编码速度,但编码效率一般低于常规编码。

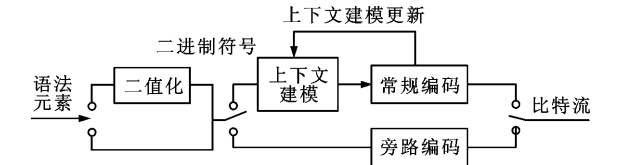


图 2 CABAC 编码流程

1.2 二值化

二值化的输入是语法元素,输出是二进制字符串。若输入的语法元素就是二进制符号,则跳过二值化的处理,直接通过旁路送到下一级处理。在实际应用中,二值化采用了一元码(U)、截断一元码(TU)、 k 阶指数哥伦布编码(EGk)和定长码(FL)、截断莱斯码(TR)等方法。HEVC 标准协议中对每个语法元素都指定了固定的二值化方法,因此可以在不保存和传输任何码表的情况下恢复原值^[1]。

1.3 上下文建模

上下文概率模型的建模过程,即为每个编码符号选择并更新概率模型的过程。对于不同语法元素或者同一语法元素不同位置的二进制字符,其对应的上下文模型均不同。HEVC 标准为每个语法元素都给出了上下文索引,确定了上下文索引后,通过查表可以得到唯一的上下文模型的起始值。上下文模型主要包括概率状态索引(pStateIdx)和大概率符号(Mps)两个参数。

随着编码的进行,需要对上下文模型参数及时更新,图 3 给出了部分状态更新的示意图。当前编码符号为大概率符号(Mps)时,原来的概率状态索引增加;编码符号为小概率符号(Lps)时,状态

索引减少。当概率状态索引为 0 时,Lps 和 Mps 的概率相等,若再出现 Lps 符号,Mps 与 Lps 的值需要互换。

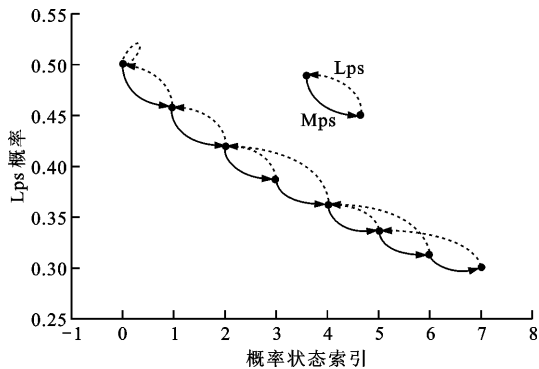


图 3 概率状态更新

为了提高熵编码的数据吞吐率,本文采用了并行化 WPP 技术,每一行编码树单元(CTU)都有独立的概率模型^[10]。图 4 给出了三线程并行处理的过程,T1、T2、T3 表示独立的处理线程^[2]。除了第 1 行第 1 个 CTU 的初始化需要查表得到初始值以外,其他行的 CTU 的初始概率模型均以前一行第 2 个 CTU 的概率模型为基准。当前 CTU 的预测和运动信息需要依赖上方和右上方 CTU 的数据,新的编码线程比上一线程延迟两个 CTU 时,可以使用更多的上下文信息,提供更好的压缩性能。

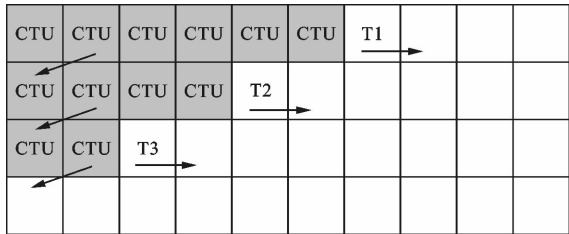


图 4 WPP 编码过程及概率更新

1.4 二进制算术编码

二进制算术编码基于区间递归划分原理实现,以编码符号 110 为例,其编码过程如图 5 所示。初始编码区间长度 R 和区间下限 L 分别设置为 510 和 0。编码过程根据输入的二元符号及其上下文模型不断对编码区间进行划分,当编码区间长度 R 为 76 时,位于(256,512)区间范围之外,需要对区间进行归一化,也即是对区间范围和区间下限进行移位,使得区间长度保持在(256,512)范围内。在归一化的移位过程中,会输出已经确定的区间下限的高位比特“01”。编码过程中,不断重复上述步骤,最终完成二进制算术编码。

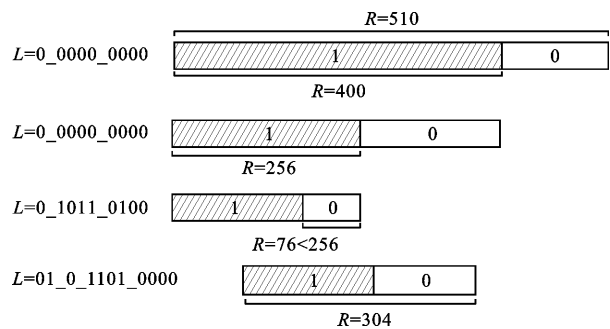


图 5 二进制算术编码示意图

2 熵编码 VLSI 架构

图 6 为熵编码整体硬件架构,熵编码主要包括数据预处理、二值化、上下文建模和算术编码 4 个模块。语法元素存储单元和控制总线为上级模块,其中语法元素存储单元存放上级模块处理完成的语法元素信息,控制总线控制熵编码模块的工作状态。熵编码模块启动后,首先进行概率表的初始化,编码帧第一个 CTU 的初始化数据来源为只读存储器 ROM,其余 CTU 的初始化数据来源均为概率表 3。初始化完成后,开始从语法元素存储单元读取对应 CTU 的数据,数据预处理模块和二值化模块将读取的 CTU 的语法元素处理并编码为二进制符号串。

二值化与上下文建模之间加入了数据缓冲 FIFO 单元以平衡模块之间的吞吐差异。根据不同的配置信息,FIFO 中的二进制符号串进入不同的上下文建模和算术编码模块进行编码并输出比特流,最终将编码后的数据通过高级可扩展接口 (AXI) 总线存储到双倍速率同步动态随机 (DDR) 存储器。为了获取较高的性能,本文采用流水线形式并且支持三行 WPP 并行技术。编码线程开始时,同步进行相关编码单元的概率模型更新,行与行之间的上下文模型更新无需采用额外的处理逻辑,且后续编码的 CTU 也无需额外的数据等待周期,可以直接进入流水线进程中编码。前一个 WPP 三行处理线程与当前 WPP 三行处理线程之间的初始化模型传递采用独立的存储空间概率表 3 保存,将上下文模型的初始化与处理过程中的上下文模型的更新分离,省去了复杂的上下文存储器切换操作。

本文设计的熵编码器同时支持在线编码和离线编码两种模式。在线编码过程中,熵编码处理的数据直接来自上级处理单元,可进行视频传输过程中的流式计算。离线编码时,编码数据来源为 DDR,通过 AXI 总线将数据从 DDR 搬移到数据解析单元,数据解析单元再将数据按照固定格式存储到语法元素存储单元。在线编码与离线编码的结合,更

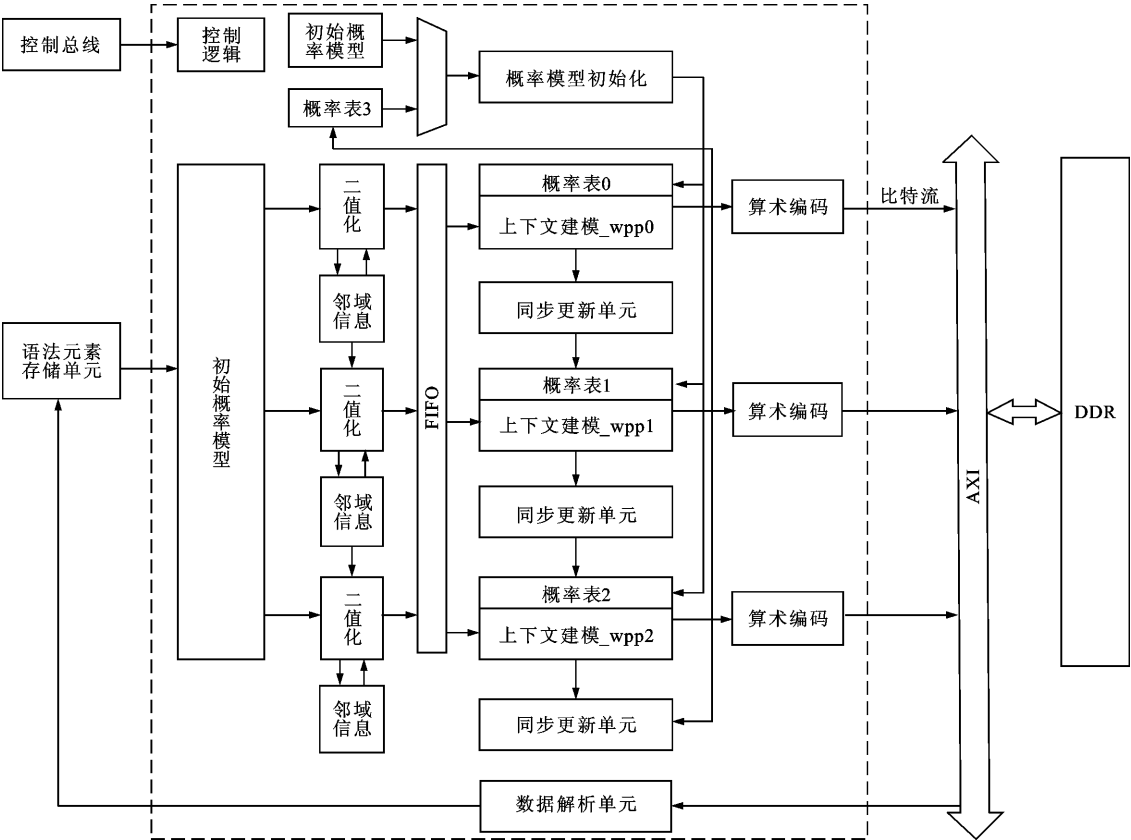


图 6 熵编码硬件架构

好地平衡了整个设计系统的吞吐性能。

2.1 数据解析单元

数据解析单元结构如图 7 所示,主要完成离线模式下数据从 DDR 到语法元素存储单元的搬移。

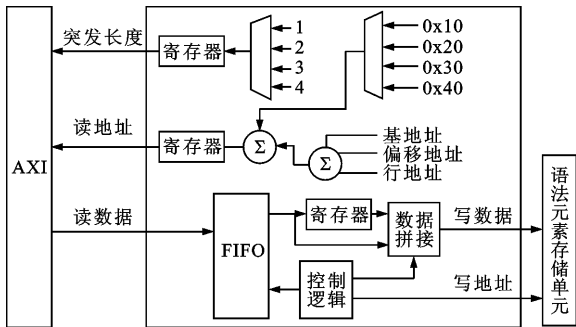


图 7 数据解析单元结构

DDR 与数据解析单元通过 AXI 总线进行通信,数据解析单元计算数据的突发长度及读地址后,通过寄存器输出到 AXI 总线模块,读地址根据当前编码单元的位置以及系统给出的熵编码起始地址计算得到。由于 AXI 总线位宽为 128 b,而语法元素存储单元数据位宽为 256 b,因此数据拼接模块需要完成数据的位宽匹配。同时,控制逻辑根据当前语法元素的类型给出其在语法元素存储单元的地址。

2.2 预处理模块

预处理模块硬件结构如图 8 所示,其根据协议的语法规则完成对输入的语法元素的规整,并输出到二值化模块进行二值化处理。

熵编码主要进行编码单元(CU)、预测单元(PU)、变换单元(TU)的编码处理,因此预处理模块主要包括处理这 3 个单元的语法元素的功能单元。

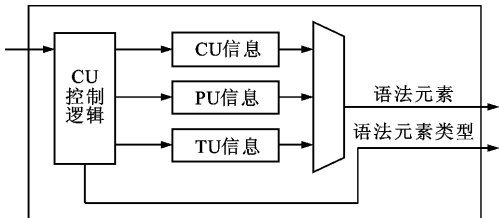


图 8 预处理模块硬件结构

2.3 二值化模块

在 HEVC 标准中,主要包括 FL、TR、EGK 3 种二值化方法。对于一些特殊的语法元素,HEVC 制定了特别的定制(Custom)二值化方法^[11]。二值化模块硬件结构如图 9 所示,模块输入为语法元素及语法元素的类型,控制逻辑根据编码单元的深度信息遍历整个编码树单元,并通过有限状态机(FSM)完成不同语法元素处理逻辑的切换,最终将模块输

出的二值化后的二进制符号 Bin、上下文索引 ctxIdx 及语法元素对应的编码类型 code_mode 存储到 FIFO 缓冲单元中。二值化按照 Z 字形的扫描方式完成对一帧图像的处理。

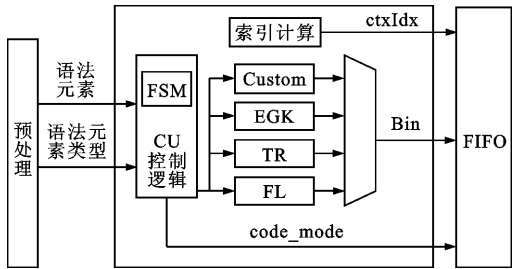


图 9 二值化模块硬件结构

2.4 上下文建模模块

编码过程中,需要对每一个语法元素建立概率模型,上下文建模模块硬件结构如图 10 所示,其主要输入为来自二值化模块的二进制 Bin、上下文索引 ctxIdx、编码模式 code_mode 以及当前 CTU 所在行的索引 wpp_index。输出为二进制符号 Bin、大概率符号 Mps、概率状态索引 pStateIdx 以及编码模式 code_mode。

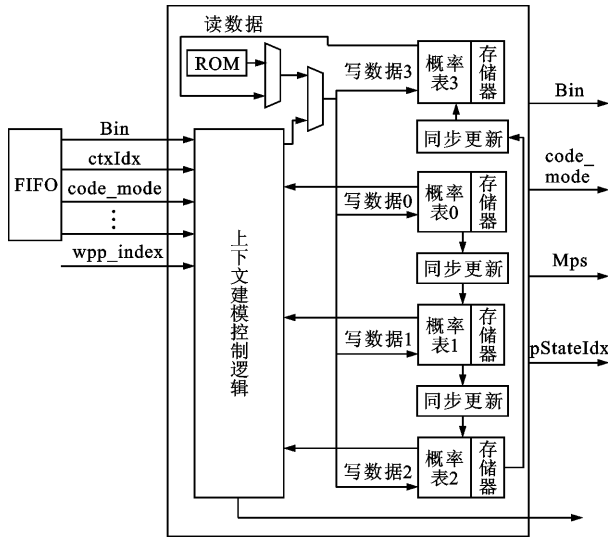


图 10 上下文建模模块硬件结构

熵编码启动后,首先进行概率表的初始化,一帧图像的第 1 个 CTU 的初始化数据为 ROM 中存储的协议规定的初始值,其余情况下的初始化的值均为上一行第 2 个 CTU 的概率模型,也即是概率表 3 中存储的概率值。编码过程中,按照配置信息,在相应的概率表中查找概率值,并输出到二进制算术编码模块。

行概率模型同步更新单元结构如图 11 所示。在编码开始前,通过行索引 wpp_index 和当前编码

的 CTU 块的横向编码 lcu_x 确定 4 个概率表的读写规则,读写使能数据从高位到低位分别代表概率表 0 到概率表 3 的读写使能信号。读写地址由基地址与所更新概率表的地址偏置相加得到,初始化过程的地址偏置默认为 1,编码过程中地址偏置根据当前所编码符号的类型动态变化。

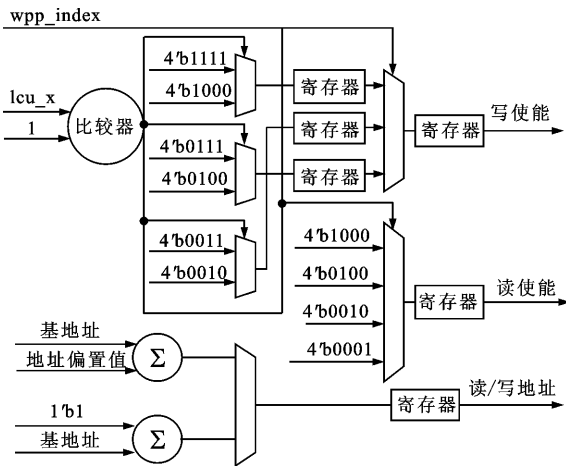


图 11 行概率模型同步更新单元结构

上下文建模中单线程概率更新结构如图 12 所示。此部分电路根据当前二元符号 Bin 的值及其上下文索引 ctxIdx,从对应的概率表中读出概率状态,并完成概率状态的更新,此部分电路采用两级流水线实现。二元符号之间的概率状态存在数据依赖,因此当连续周期或者同一周期内符号状态索引值相同时,需要采用多路选择器在当前读出的数据和更新的数据之间选择正确的模型。

2.5 二进制算术编码

二进制算术编码模块的硬件结构如图 13 所示,输入为上下文建模模块输出的二进制 Bin、大概率符号 Mps、概率状态索引 pStateIdx 以及编码模式 code_mode,输出为最终编码后的码流。本文的二进制算术编码采用 4 级流水线结构:第 1 级,根据概

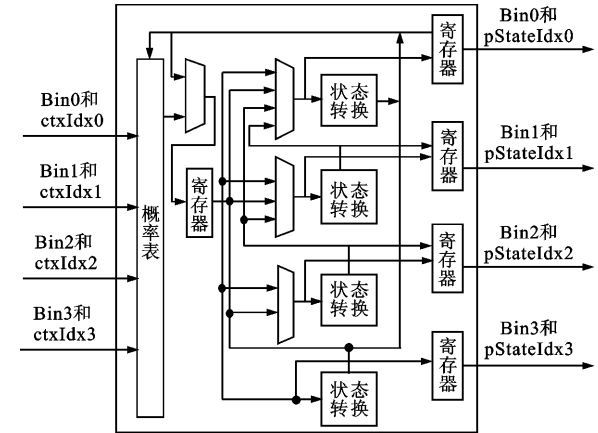


图 12 单线程概率更新结构

率状态索引得到当前的小概率符号 Lps 的区间 rLps;第 2 级,实现编码区间的更新;第 3 级,实现编码区间下限的更新;第 4 级,将码流按照字节的格式打包输出。

3 结果分析

数据吞吐率是衡量视频编码系统的关键性能指标。表 1 为 HEVC 标准下本文与其他设计的性能对比。

表 1 HEVC 标准下本文与其他设计的性能对比

设计来源	工艺特征尺寸/nm	频率/MHz	吞吐率/(Mb·s ⁻¹)	平均单周期编码符号数	逻辑门数/10 ³
文献[7]	90	200	800	4	54.7
文献[8]	45	280	1 120	4	
文献[13]	28	312	1 062	3.4	83.5
本文	55	300	1 512	5	130.7

从表 1 可以看出,本文在数据吞吐率方面有了较大的提升。文献[7-8,13]均采用单个熵编码处理单元,本文在采用流水线技术及 WPP 并行架构后,

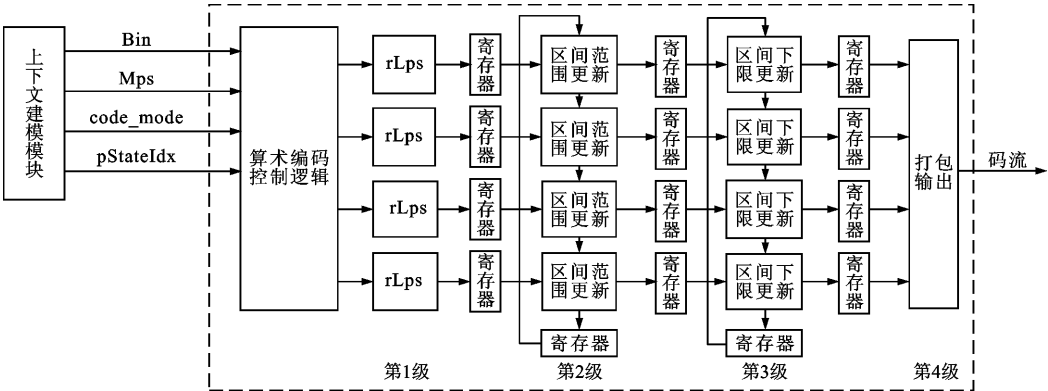


图 13 二进制算术编码模块的硬件结构

熵编码器可以进行多线程处理单元同步编码,工作频率为 300 MHz 时,平均单周期编码符号为 5 个二进制码,数据吞吐率为 1 512 Mb/s,满足超高清视频的编码需求。相比于表 1 中的其他方法,本文方法的数据吞吐率分别提升了 89%、35%、42%,但是也带来了较大的面积资源占用,考虑到设计与面积的折中,为了获取高的数据吞吐率,牺牲一定面积是不可避免的。

4 结 论

超高清视频的发展对视频编码标准提出了更高的需求,相比于上一代视频编码标准,HEVC 标准具有更高的压缩性能,同时也提升了编解码的复杂度。本文研究的熵编码器位于 HEVC 视频编码系统的最后一级,其采用的二进制算术编码算法中存在很强的数据依赖,限制了熵编码硬件设计的并行化程度,使得熵编码成为整个视频编码系统的瓶颈。

本文在研究熵编码器中二值化、上下文建模和二进制算术编码过程的基础上,将整个编码系统划分为 4 个模块,包括数据预处理、二值化、上下文建模以及二进制算术编码模块,最终设计并实现了一种 WPP 并行编码硬件架构。本文所采用的架构充分利用了硬件电路的并行计算的优势,极大地提高了熵编码器的数据吞吐率。经过验证,本设计工作频率可达 300 MHz,吞吐率为 1 512 Mb/s,满足超高清视频的实时编码需求。

参考文献:

- [1] 朱秀昌,刘峰. H. 265/HEVC 视频编码新标准及其扩展 [M]. 北京:电子工业出版社,2016:151-176.
- [2] ISO/IEC. High efficiency video coding, ITU-T Recommendation H. 265; ISO/IEC 23008-2 [S]. Geneva, Switzerland: ISO, 2013.
- [3] ZHOU Dajiang, ZHOU Jinjia, WEI Fei, et al. Ultra-high-throughput VLSI architecture of H. 265/HEVC CABAC encoder for UHD TV applications [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2015, 25(3): 497-507.
- [4] SZE V, CHANDRAKASAN A P, BUDAGAVI M, et al. Parallel CABAC for low power video coding [C]// Proceedings of the 15th IEEE International Conference on Image Processing. Piscataway, NJ, USA: IEEE, 2008: 2096-2099.
- [5] TSAI C H, TANG C S, CHEN L G. A flexible fully hardwired CABAC encoder for UHD TV H. 264/AVC high profile video [J]. IEEE Transactions on Consumer Electronics, 2012, 58(4): 1329-1337.
- [6] 陈成. HEVC 熵编码器的 VLSI 架构设计 [D]. 合肥:中国科学技术大学,2017:37-56.
- [7] 郭勇,王桂海,范益波,等. 基于 HEVC 标准的全高清 CABAC 编码器设计 [J]. 电视技术,2014,38(9): 71-74.
- [8] GUO Yong, WANG Guihai, FAN Yibo, et al. High definition CABAC encoder design based on HEVC [J]. Video Engineering, 2014, 38(9): 71-74.
- [9] RAMOS F L L, GOEBEL J, ZATT B, et al. Low-power hardware design for the HEVC binary arithmetic encoder targeting 8K videos [C]// Proceedings of the 2016 29th Symposium on Integrated Circuits and Systems Design. Piscataway, NJ, USA: IEEE, 2016: 1-6.
- [10] DING L F, CHEN W Y, TSUNG P K, et al. A 212MPixels/s 4096 × 2160p multiview video encoder chip for 3D/quad HDTV applications [C]// Proceedings of the Solid-State Circuits Conference: Digest of Technical Papers. Piscataway, NJ, USA: IEEE, 2009: 154-155.
- [11] 李姝仪. HEVC 中率失真算术编码与基于 WPP 的并行熵编码器的 VLSI 架构设计 [D]. 西安:西安电子科技大学,2017:47-50.
- [12] VIZZOTTO B, MAZUI V, BAMPI S. Area efficient and high throughput CABAC encoder architecture for HEVC [C]// Proceedings of the 15th IEEE International Conference on Electronics Circuits and Systems. Piscataway, NJ, USA: IEEE, 2015: 572-575.
- [13] ZHANG Yuanzhi, LU Chao. A highly parallel hardware architecture of table-based CABAC bit rate estimator in an HEVC intra encoder [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019, 29(5): 1544-1558.
- [14] TSAI S F, LI C T, CHEN H H, et al. A 1062 Mpixels/s 8192 × 4320p high efficiency video coding (H. 265) encoder chip [C]// Proceedings of the 2013 Symposium on VLSI Circuits. Piscataway, NJ, USA: IEEE, 2013: 188-189.

(编辑 武红江)