

基于 HEVC 的 CABAC 二进制算术编码器的 FPGA 实现

王尧^{1,2,3}, 汤心溢^{1,3}

(1. 中国科学院上海技术物理研究所 中国科学院红外探测与成像技术重点实验室, 上海 200083;
2. 中国科学院大学, 北京 100049; 3. 上海科技大学 信息科学与技术学院, 上海 201210)

摘要: 本文基于 H.265/HEVC 视频编码标准, 实现了 CABAC 编码中二进制算术编码器常规编码模式下的一种硬件流水线结构, 根据算法特性设计并优化了编码器的硬件架构, 将概率状态数据储存在 SRAM 中, 并使用查找表优化概率估计更新运算; 对编码数据进行打包处理, 简化概率估计更新带来的计算, 以优化视频数据流编码速度; 二进制算术编码采用多级流水线结构, 支持四路并行编码。仿真结果表明, 本文的硬件 CABAC 二进制算术编码器平均每时钟周期可以完成 4 个 bin 的编码, 符合较高帧率的 1080p 视频实时编码要求。

关键词: HEVC; 熵编码; CABAC; FPGA; 二进制算术编码器

中图分类号: TN919.81 文献标识码: A 文章编号: 1001-8891(2020)04-0335-06

FPGA Implementation of CABAC Binary Arithmetic Encoder Based on HEVC

WANG Yao^{1,2,3}, TANG Xinyi^{1,3}

(1. Shanghai Institute of Technical Physics, Key Laboratory of Infrared System Detection and Imaging Technology, Chinese Academy of Sciences, Shanghai 200083, China; 2. University of Chinese Academy of Sciences, Beijing 100049, China;
3. School of Information Science and Technology, Shanghai Tech University, Shanghai 201210, China)

Abstract: Based on the H.265/HEVC video coding standard, a hardware pipeline structure is implemented in this study in the regular mode of a binary arithmetic encoder in CABAC coding. Based on the characteristics of the algorithm, the hardware architecture of the coding engine is designed and optimized. The probability state data are stored in a SRAM, and the probability estimation updating operation is optimized using a lookup table. The coding data are packaged to simplify the calculation obtained by the update of the probability estimation to optimize the coding speed of the video data stream. Binary arithmetic coding uses a multistage pipeline structure to support four-way parallel encoding. Simulation results show that the hardware of the CABAC binary arithmetic coder can complete the encoding of four bins per clock cycle, which satisfies the higher frame rate of 1080p video real-time encoding requirements.

Key words: HEVC, entropy coding, CABAC, FPGA, binary arithmetic encoder

0 引言

作为新一代视频编码标准, H.265/HEVC (high efficiency video coding) 可支持 3840 pixel × 2160 pixel 以上级别高清视频的压缩编码, 同时帧率达到 30 帧/s 以上级标准, 要求视频流编码具有更高的压缩率。H.265/HEVC 标准同上一代 H.264/AVC 标准相比, 在相同的图像质量下码流数据可减少 50% 左右^[1], 同时也要求更复杂的编码计算; 同 CAVLC (context-based adaptive variable-length coding) 相比, CABAC (context-based adaptive binary arithmetic coding, 上下

文自适应二进制算术编码) 可节省 9% ~ 14% 的码流数据, 同时要求运算量提高约 40%^[2]。

HEVC 标准的结构框图如图 1 所示, 主要分为预测、量化、滤波和熵编码部分。其中, 熵编码采用 CABAC 编码器, 输入为量化后的残差数据流, 通过 CABAC 编码器后输出比特流, 即最终的 H.265 视频数据流。本文基于 HEVC 标准, 在硬件加速方面对 CABAC 的二进制算术编码器进行了设计实现与优化。

1 CABAC 二进制算术编码器

CABAC 编码器分为 3 部分: 二值化, 上下文建

收稿日期: 2019-03-04; 修订日期: 2019-04-18.

作者简介: 王尧 (1994-), 男, 硕士研究生, 主要研究方向为视频编码技术; E-mail: xnx1994@foxmail.com。

模，以及二进制算术编码器。二值化过程将输入的非二进制残差数据语法元素转化为二进制序列(bins)；在上下文建模过程中，对二值化后的序列中每个符号(bin)出现的概率进行估计，分为最大概率符号(most probability symbol, MPS)和最小概率符号(least probability symbol, LPS)；二进制算术编码器则将此序列通过算术编码的形式编码为最终的比特流符号，完成HEVC编码输出。该算术编码的本质就是用一个码字表示一段数据流，这个码字是对一段数据流不断分割后获得的^[3]。每个bin在编码完成后都需要对其上下文模型进行更新，即概率估计过程，该过程在上

下文建模部分中完成，更新当前概率状态索引 σ 和LPS的值。CABAC算术编码器整体结构如图2所示。

在CABAC编码中，概率状态(state)代表当前符号所对应的LPS出现概率，概率状态估计值由一组有限的集合表示，LPS概率取值区间为[0.01875,0.5]，对应64个概率索引值。概率理论值可由公式(1)计算得出：

$$p_{\sigma} = \alpha p_{\sigma-1}, \quad \sigma = 1, \dots, 63, \\ \alpha = (0.01875/0.5)^{1/63}, \quad p_0 = 0.5 \quad (1)$$

式中： σ 为概率状态索引； p_{σ} 为索引值 σ 对应的LPS出现概率理论值； α 为迭代计算系数。

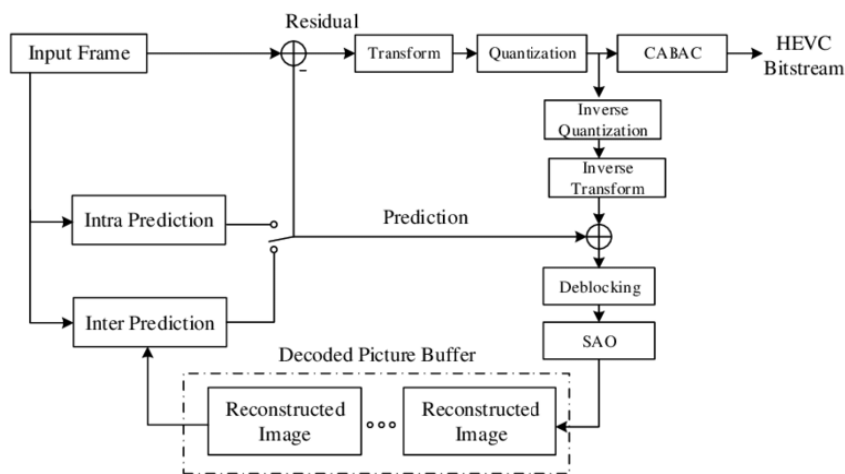


图1 HEVC 编码架构

Fig.1 HEVC coding architecture

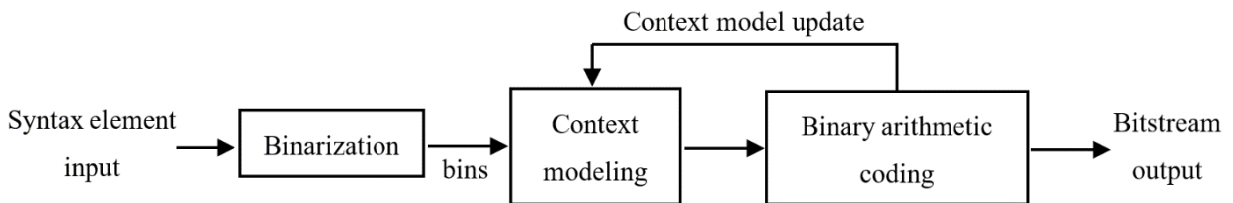


图2 CABAC 算术编码器基本结构

Fig.2 Basic structure of CABAC arithmetic encoder

边界情况下，当MPS出现时，概率状态索引 σ 根据特定的规则跳转增加^[4]，LPS对应的概率降低；当LPS出现时，概率状态索引 σ 跳转降低至对应的下一个值，LPS概率升高。当 $\sigma=0$ 时，LPS与MPS所对应的符号互换。CABAC概率估计更新模型如图3^[4]所示，该更新过程决定了二进制算术编码过程中概率状态索引和LPS的输入值，同时将当前编码符号bin输入至二进制算术编码器中。

时所对应的range更新值，记为rLPS，之后将对low和range的取值进行更新。

在二进制算术编码模块中，输入数据包括当前待编码符号bin，LPS值，以及上下文建模/更新后的概率状态索引 σ 。该模块主要负责编码区间的更新，并将更新好的编码区间打包输出比特流。描述编码区间的参数由low和range构成，low为当前区间的下界值，range为当前区间长度，每输入一个bin，二进制算术编码通过 σ 和目前的range值查表得到bin=LPS

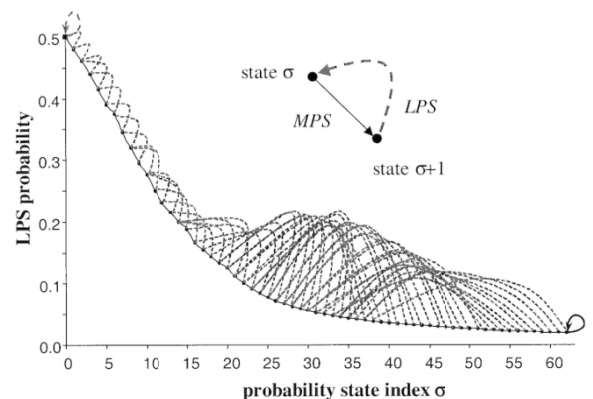


图3 CABAC 的概率估计更新模型

Fig.3 CABAC probability estimation update model

对当前符号bin，如果bin=LPS，则当前区间更

新为:

$$\begin{cases} \text{range} = \text{rLPS} \\ \text{low} = \text{low} + \text{range} - \text{rLPS} \end{cases} \quad (2)$$

如果 bin=MPS, 则当前区间更新为:

$$\begin{cases} \text{range} = \text{range} - \text{rLPS} \\ \text{low} = \text{low} \end{cases} \quad (3)$$

low 和 range 值更新完毕后, 编码数据通过重归一化^[5]进行实时的打包输出, 完成二进制算术编码器的比特流输出。CABAC 二进制算术编码器的算法流程如图 4 所示, 虚线所示流程为概率估计更新的计算过程, 可在上下文建模部分中完成^[6], 其中 σ 的更新计算通常使用查找表的方式代替式(1)的概率转移运算。随着新的编码区间计算完毕, 会产生部分已确定的编码位, 后来输入的符号无法修改这部分编码。重归一化的目的是将已确定的编码位进行及时的打包输出, 并保证更新后的编码区间仍然可以用有限位的精度来表示。

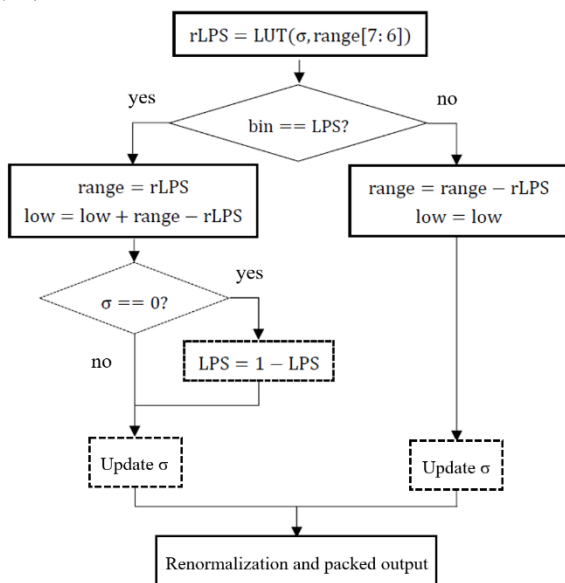


图4 CABAC 二进制算术编码器的算法流程

Fig.4 Algorithm flow of CABAC binary arithmetic encoder

2 编码器算法实现与优化

根据硬件平台的并行运算特性, 该二进制算术编码器采用如图 5 所示的多级流水线的硬件加速结构, 通过牺牲硬件资源提高数据编码速度。编码器主要针对常规编码模式, 支持四路 bin 并行输入, 首先对输入数据进行预处理, 提取编码二进制 bin 值、当前 LPS 取值及索引值 σ ; 之后依据图 4 所示的二进制算术编码器算法流程, 以硬件流水线的形式计算每个 bin 对应的 rLPS、新的 range 值和 low 值, 并对编码区间进行重归一化, 最后以每 8 个比特作为一个字节打包输出编码比特流, 完成编码器输出。

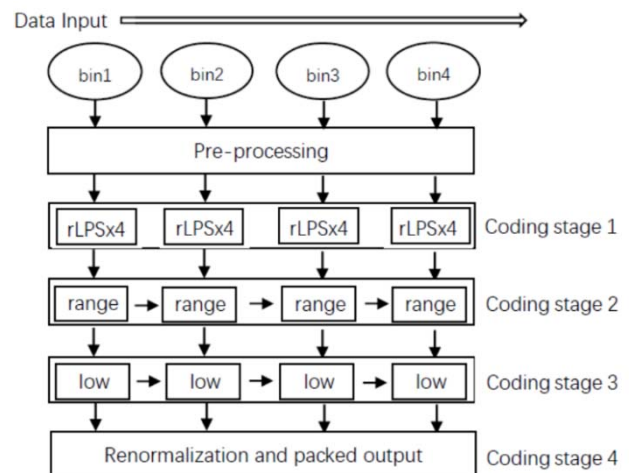


图5 二进制算术编码器多级流水线硬件结构

Fig.5 Multi-stage pipeline hardware structure of binary arithmetic encoder

该二进制算术编码器在流水线结构中分为 4 个编码阶段。在第一阶段中, 对每个 bin, 编码器判断当前字符 bin 是否为 LPS, 并根据输入的概率状态索引 σ 查表得到 rLPS 值。由于此时尚未读取到最新的 range 值, rLPS 的计算依赖于该 range 值的最高两位, 其可能取值有 4 种, 所以需要计算并存储每个 bin 的所有 4 种 rLPS 取值, 记录为 rLPSTab, 并在第二阶段 range 的更新中通过式(4)的规则进行查表读取, 解决了 rLPS 对 range 值的数据依赖问题:

$$\text{rLPS} = \text{LUT}(\sigma, \text{range}[7:6]) \quad (4)$$

在每路 rLPSTab 的计算中, 各使用一路寄存器保存 rLPSTab 的值, 该过程每次需要 1 个时钟周期进行数据更新; 在第二阶段中, 编码器需要通过寄存器维护最新的编码区间长度 range_last 的值, 根据当前 bin 是否为 LPS、range_last 以及通过式(4)查表读取到的 rLPS 值, 通过式(2)、(3)得到更新后的编码区间长度 range_update, 并各使用一路寄存器进行数据暂存。编码阶段 1、2 采用如图 6 所示的电路计算结构, 在 range 计算后需要对 range 值进行重整, 当 bin 为 MPS 时, 在计算后的 range 值 (rMPS) 最高位为 0 时需要对其算术左移一位, 可以通过二选一 MUX 解决; 当 bin 为 LPS 时, range 值即为重整后的 rLPS 值, 此时 range 的重整过程被预存在 rLPSupdateTab 中, rLPSupdateTab 保留了 4 种 rLPS 值所对应的重整结果, 可在编码阶段 1 内预先完成, 需要时从 LUT 中调取, 避免了本级流水线对 range 值的数据依赖。之后, 根据当前 bin 是否为 LPS, 得到编码阶段 2 对应的 range_update 结果, 进行 low 的计算。数据从预处理到计算结果送入编码阶段 3 的过程需要 2 个时钟周期。

在第三阶段中, 编码器需要通过寄存器维护最新

的编码区间下界 low_last 的值, 根据当前 bin 是否为 LPS、rMPS 和 low_last , 通过式(2)、(3)得到更新后的 low_update ; 第四阶段为重归一化过程, 由于 low 的部分高位二进制数值不再随输入而产生变化, 需要重新维护编码区间, 判断已确定的编码比特, 将缓冲比特序列算术左移, 将待输出的 bit 存放在比特暂存区中, 并将已确定的二进制编码数据以 8 个 bit 为一个单位打包输出, 即为二进制算术编码器的输出比特流。由于每个 bin 从输入该模块到输出比特流需要依赖数据打包等待过程, 完成比特输出需要 1~3 个不等的时钟周期, 为防止数据溢出, 设置深度为 64 bit 的比特 $buffer$, 将四路编码进行缓冲存储。编码阶段 3、4 的电路结构如图 7 所示。

在实际运算中, 对概率估计更新过程, 使用查表法代替乘法运算, 可大大减少运算时间, 节省时钟周期和资源占用。采用概率状态转移查找表, 表示每个 σ 对应的下一个概率状态及概率估计值, 将概率估计更新后的概率数据存放在 SRAM 中, 在下次概率估计更新时对 SRAM 进行读取和重新写入, 以完成概率模型的动态更新, 避免了使用寄存器存储大量数据导致的芯片资源和面积消耗。本文的二进制算术编码器对概率估计更新进行了运算效率的优化, 由于编码器模块每次的输入为同一数据块中连续的 4 个 bin , 可

将该并行输入的 4 个 bin 的概率估计更新过程简化为只计算其中第一个 bin 的概率估计更新, 减少硬件资源占用和运算时间; 这一过程对输入的二进制编码数据进行打包处理, 即以 4 个符号的包为单位进行编码 (对应文献[7]中 $N=4$ 的情况), 并将概率估计更新的计算结果在 SRAM 中进行更新。当输入的 4 bit 序列的后三位为交替出现的序列, 即 {LPS,MPS,LPS} 或 {MPS,LPS,MPS} 时, 对于编码效率的影响最小, 此时只有 1 个 bin 的概率估计更新值受到 1 次 σ 跳转的影响^[7]; 而对于最差情况^[8], 输入的 4 bit 序列的后三位为同一符号, 即 {LPS,LPS,LPS} 或 {MPS, MPS, MPS} 时, 此组序列仅有 3 次 σ 跳转的偏差, 整体而言可兼顾运算资源与编码效率的平衡, 并将算术编码器部分的 SRAM 使用数量减少为原始算法^[9]的 2/5, 同时略微提高硬件编码速度。

3 仿真与结果

对本文的二进制算术编码器在 Vivado 2018.2 平台上进行前仿真, 仿真结果如图 8 所示, 在不考虑初始化及中断复位的情况下, 该算术编码器每个时钟周期可以完成 4 个 bin 的编码, 相比文献[3]的结构有显著的编码速度提升。

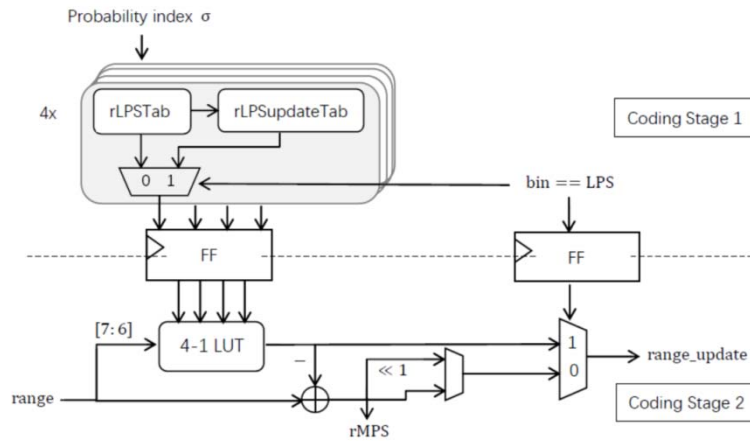


图 6 Range_update 的计算结构

Fig.6 Calculation structure of range_update

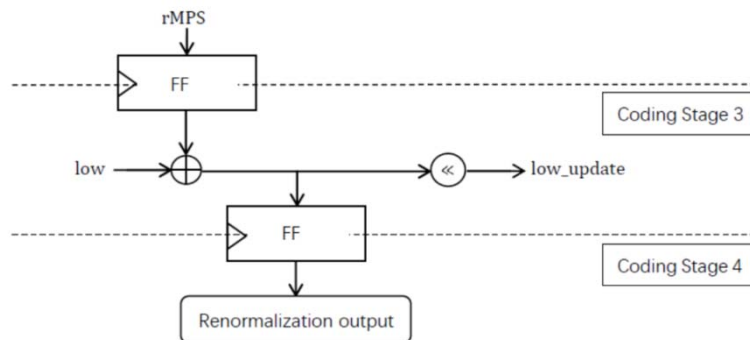


图 7 Low 的计算和比特输出

Fig.7 Calculation and bit output of low

采用 Xilinx 的 xczu9eg-ffvb1156 FPGA 对本文的二进制算术编码器模块进行硬件实现,综合实现后的资源占用情况如表 1 所示。

在本文设计中,将 DSP 乘法器运算全部转化为查找表运算,避免了时序问题的同时提高整体编码速度,数据输出时延为 3~5 个时钟周期。本文与其他参考文献的实现结果对比如表 2 所示,与文献[3]相比,本文的二进制算术编码器消耗了大量的 LUT 资源用于概率索引查找表以及 range、low 的快速计算,

而每个 Slice 可提供 4 个 6 输入的 LUT,在该系列 Xilinx 器件中每个 CLB 逻辑单元对应 2 个 Slice,导致承载 LUT 的 CLB 的数量消耗为文献[3]的 4 倍左右,与后者每时钟周期 0.7 个 bin 的编码速度对比,本文在牺牲了大量片上资源的基础上,获得了更快的编码速度,实时性更强,在相同条件下可支持 1080p 视频数据在 30~60 f/s 帧率下的实时编码,符合基于 HEVC 高效的 CABAC 二进制算术编码器的设计要求。

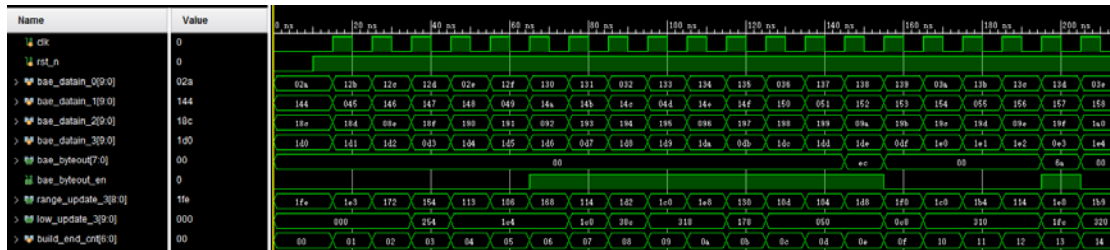


图 8 二进制算术编码器仿真结果

Fig.8 Binary arithmetic encoder simulation results

表 1 二进制算术编码器资源使用情况

Table 1 Binary arithmetic encoder resource usage

On-chip resources	rLPSTab	Range update	Low update	Renormalization and packed output	Total
LUT	401	328	258	1402	2389
FF	364	142	9	220	735
DSP	0	0	0	0	0
CLB	180	56	47	249	532
Slice	360	112	94	498	1064

表 2 与其他文献实现结果对比

Table 2 Comparison with implementation results of other references

	WANG ^[3]	ZHAO ^[10]	This paper
Slice	284	N/A	1064
Bin/cycle	0.7	1	4
Cycle delay	3-5	5	3-5
Coding standard	H.264	HEVC	HEVC

4 总结

本文实现了 HEVC 的 CABAC 二进制算术编码器常规编码模式下的硬件加速结构,基于并行流水线对二进制算术编码器进行硬件结构设计与实现,通过 SRAM 存储概率估计更新数据,使用查找表优化概率估计及编码区间的计算过程,避免 DSP 乘法运算,缩短运行时间。结果表明,本文的硬件 CABAC 二进制算术编码器平均每时钟周期可以完成 4 个 bin 的编码,适用于对 HEVC 视频编码实时性要求较高的场合,功耗相对有限,编码效率高,支持较高帧率的 1080p 视频实时编码,具有较快的数据编码传输能力。

参考文献:

- [1] Goldman M S. High-efficiency video coding (HEVC): the next-generation compression technology[C]//*SMPT E Motion Imaging Journal*, 2015, **121**(5): 27-33.
- [2] Marpe D, Schwarz H, Wiegand T. *Context-based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard*[M]. IEEE Press, 2003.
- [3] 王震, 张延军. 基于 FPGA 的 CABAC 算术编码器设计[J]. *微电子学与计算机*, 2015(6): 167-170.
WANG Zhen, ZHANG Yanjun. The design of CABAC arithmetic encoder based on FPGA[J]. *Microelectronics & Computer*, 2015(6): 167-170.

(下转第 347 页)