文章编号:1674-2974(2019)02-0075-06 DOI:10.16339/j.cnki.hdxbzkb.2019.02.010

一种 HEVC 全系统低冗余 CABAC 解码器

林子明†,梁利平
(中国科学院 微电子研究所,北京 100029)

摘 要:为解决最新一代视频压缩标准 HEVC(High Efficiency Video Coding)中熵解码部分存在的语法元素串行依赖性问题,本文提出一种低时钟数冗余的 CABAC(Context-Based Adaptive Arithmetic Coding)硬件解码器实现方案.核心采用动态码表预处理方式和并行处理电路设计,提高了时钟利用效率,满足 HEVC Level4.1 Main Tier 的全部要求,可以以 40 MHz 的时钟完成 1080HD@60fps 的视频实时解码.

关键词:HEVC;CABAC;解码器;熵解码;低冗余
中图分类号:TN41 **文献标志码:**A

A Full System Low Redundancy CABAC Decoder for HEVC

LIN Ziming†, LIANG Liping
(Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, China)

Abstract: To solve the serial dependency problem that came up with CABAC in the next generation of video coding standard called High Efficiency Video Coding (HEVC), this work proposed a low redundancy full system CABAC decoder solution. By adopting the dynamic codebook and parallel computing design, this work improved the clock efficiency and could satisfy the HEVC Level4.1 Main Tier and decode a 1080HD@60fps video with a clock of 40MHz.

Key words: HEVC; CABAC; decoder; entropy decoding; low redundancy

HEVC 标准在 2013 年正式发布,其核心目标在于:在 H.264 High Profile 的基础上,压缩效率提高 1 倍.HEVC 整体上依然采用 H.264 的预测加变换的混合编码结构,其中在熵编码部分舍弃了变长编码、统一采用基于上下文的自适应算术编码(CABAC).CABAC 压缩算法是一种高压缩效率的算法,其压缩效率较变长编码 CAVLC 提高了 9%~14%,同时也大幅提高了算法复杂度,其串行本质决定了其在高清、超高清应用中实现实时编解码非常困难.

在 CABAC 应用中,编码和解码过程的串行依赖主要来源于两个方面:CABAC 算法依赖和 HEVC 语法结构依赖.CABAC 算法依赖存在于区间信息更新和上下文模型更新,比特间的依赖性导致难以连续多个时钟周期的连续计算.HEVC 语法结构依赖存在于视频协议中,为了提高压缩效率,减少不必

* 收稿日期:2018-03-25
作者简介:林子明(1987—),男,河北唐山人,中国科学院微电子所博士研究生
† 通讯联系人,E-mail:linzmwl@163.com

要的信息量,后续语法元素的选择严重依赖前面的语法元素信息选择,难以并行化处理,导致实际编码和解码过程时钟利用率不高.

在文献[1-3]中,提出优化的常规解码计算引擎,可以高效完成一个比特的解码,但对区间正规化和语法元素间的依赖没有进行优化. 由于算术编码对于区间的划分有精度的限制,在达到一定的区间长度后需要对区间进行归一化处理以保证算法的压缩效率,导致实际应用中效率仍然较低.

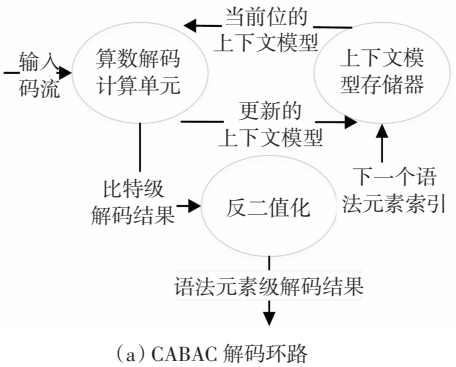
基于并行计算的考虑,文献[4]中的作者提出采用多个解码单元预测计算的技术,通过多个计算单元的并行化处理来提高峰值解码能力,但是并未解决 HEVC 算法中语法元素间的依赖性. 该算法在码流中大量存在小码块无预测的情形下以及亮度、色度计算等无依赖情形下,多个计算单元处理效率较高,在预测帧(P 帧或 B 帧)中或图像内容剧烈变化情形下,语法元素依赖性极高,解码单元利用率降低,平均到每个解码单元的实际解码效率较低.

在文献[4-6]中,虽然解码器实现的峰值解码能力及平均解码速度较高,但是多数解码器单个宏块解码仍需 300 个以上的时钟周期, 最快也需要 118 个周期,可见其解码过程中有很大部分的解码过程存在计算资源等待的情况,造成资源浪费.此外,文献[7]根据不同语法元素统计特性将语法元素分组处理,达到了很高的解码速度,但该方法需专用码流,通用性较差,不具满足一般应用要求.

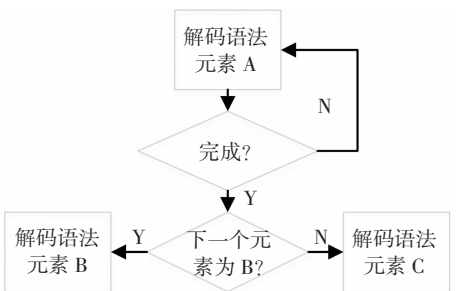
本文针对 CABAC 算法依赖性和 HEVC 语法结构依赖性提出一种低冗余全系统的 CABAC 硬件解码器实现方案.设计单周期的算数解码引擎,高效完成区间信息和上下文模型的更新以及区间正规化过程.采用动态码表预测技术,有效消除语法结构中存在的语法元素的串行依赖,减少解码器解码过程中的时钟周期冗余.本文主要分为以下几部分:第 1 节介绍 CABAC 算法和 HEVC 语法结构依赖的来源,分析其实现的瓶颈;第 2 节提出本文的实现结构以及采用的新算法;第 3 节给出硬件的测试结果并在第 4 节给出本文结论.

1 串行依赖性来源

CABAC 算法基本思想是用 0~1 之间的一个数字表示整个码流,编码方式是区间递进,这种压缩算法的压缩性能更接近熵极限[8].一个典型的 CABAC 解码过程如图 1(a)所示:



(a) CABAC 解码环路



(b)语法结构依赖

图 1 CABAC 解码瓶颈

Fig.1 Bottleneck of CABAC decoding

一个比特的解码过程分为:1)获取上下文模型和待解码比特;2)比特级解码计算;3)更新上下文模型. 且第二个比特的上下文模型选取依赖于第一个比特解码结果,这种串行依赖性严重制约了解码器吞吐率的提高, 是制约整个 CABAC 解码技术过程的瓶颈[7].反二值化过程为语法元素级处理,影响到解码结果输出和下一个上下文模型的选择,如图 1(b)所示,语法元素 A 解码完成后需经过两级判断才能确定下一个语法元素类型, 在此判断过程中,解码计算单元处于空闲状态,这一结构性依赖造成大量时钟浪费. 因此需要优化语法元素调度算法并优化解码电路引擎,提高 CABAC 解码器的吞吐率.

2 硬件实现

根据前面分析, 本设计针对 CABAC 算法依赖和 HEVC 语法结构进行了深入研究,设计单周期解码引擎解决比特级串行依赖,设计预测单元解决语法元素级的串行依赖性, 提高计算资源的利用率,缩短解码时间.

2.1 解码引擎

根据前面对 CABAC 算法分析可知, 一个算数解码运算需经历上下文模型读取、解码计算和上下文模型更新三个阶段. 其中真正的计算状态只有解

码计算过程,上下文模型更新和区间长度更新过程将造成计算单元停滞状态.进一步分析得出如下结论:

1)比特的解码值输出仅与当前输入区间长度信息和偏移量相关;

2)上下文模型可根据当前输入的区间信息和偏移量进行更新;

3）区间的归一化操作仅与当前输入区间信息初始值相关;

4)比特流输入仅有 0 和 1 两种可能性.

因此,本设计将区间信息更新过程、上下文模型更新过程与解码运算过程并行实现,如图 2 所示.

在每个比特的解码阶段同时获取区间信息和偏移量以及 0 和 1 对应的上下文模型,分别计算输

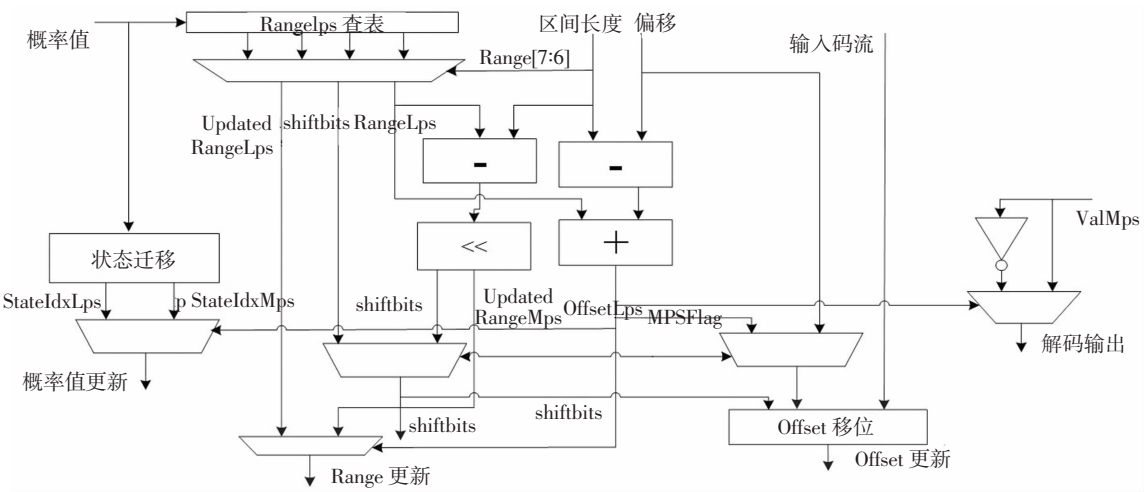


图 2　单周期解码引擎结构

Fig.2　Architecture of single-cycle decoding engine

入码流为 0 和 1 两种可能性下的区间信息更新值和上下文模型更新值,最后通过一级 2 选 1 逻辑输出解码结果. 由于区间信息在整个解码过程中持续更新且不随着语法元素的变化而变化,因此在解码单元内暂存更新后的区间信息和偏移量以减少下个比特解码时的区间信息读取时间.通过以上优化,单个比特的解码过程只需一个时钟周期即可完成,且下一个比特的上下文模型可以在本周期内预测,以减少比特间解码的等待过程.

2.2　块信息预处理

语法结构依赖是 CABAC 解码器设计过程中另一个瓶颈,该问题源于协议开发阶段片面追求高压缩率而忽略了算法复杂度的提高. 通常在连续两个语法元素解码过程中,后续语法元素的存在性以及后续语法元素上下文模型选取需要等待第一个语法元素解码完成并保存相关信息,从而造成解码器空转形成时钟浪费.在 HEVC 中,由于大尺寸图像和大范围变换块处理数量急剧增加,导致残差模块的计算是在整个 CABAC 解码中数据量最多的部分.例如对于一个最小的 4*4 变换块,HEVC 语法结构共定义了 11 个语法元素来表示一个点的值，由于各点信息在实际码流中是否存在对解码器是不可知的,理论上解码器最多需要分析 176 个语法元素.

这种冗余部分的分析过程造成了在文献 [4-6,9-10]所实现的设计中,虽然单个比特的解码速度很高，但实际解码过程时钟开销远大于理论值,解码效率较低的结果.

根据 DCT/DST 特性可知,变换后的能量主要集中在矩阵左上部分,而大多数的位置变换值为 0,以图 3(a)为例:实际码流中只存在左上角 6 个位置的图像信息(阴影部分),其余 10 个位置未编入码流,存在优化的空间.

4	2	1	0
1	1	0	0
1	0	0	0
0	0	0	0

(a)变换块语法结构分解

SubBlock 0	SubBlock 1
SubBlock 2	SubBlock 3

(b)变换块语子块划分

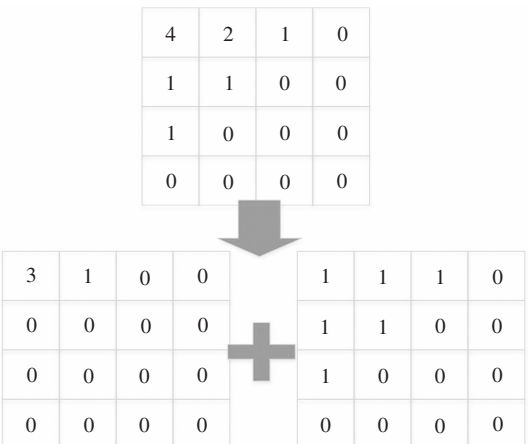
图 3　变换块和残差模块数据冗余分析

Fig.3　Transformation block and residual block parsing

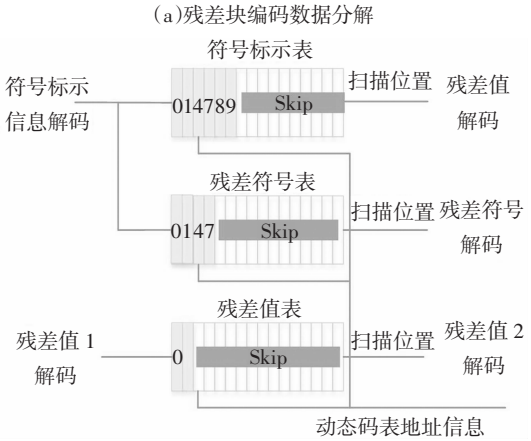
如图 3(b)所示,对于一个变换块,变换后分为有效子块（阴影部分）和无效子块，其有效性由

transform_skip_flag 和 coded_sub_block_flag 两个语法元素指示. 因此在实际解码过程中可在比特级解码阶段根据指示标志预测下一个解码的实际数据块位置信息和语法元素类型,略过子块 2 和 3 的分析过程.

残差块编码数据分解.对于图 3(a)所示的残差块,其有效部分由符号位和数值位组成,分解如图 4(a).



(a)残差块编码数据分解



(b)动态码表预处理

图 4 残差模块数据冗余处理及实现

Fig.4 Realization of redundant data processing in residual block

分解后的信息由符号标志,残差值组成,涉及 last_sig_coeff_x,last_sig_coeff_y,sig_coeff_flag 和 coeff_abs_level 几种语法元素信息,其中通过 last_sig_coeff_x,last_sig_coeff_y 可计算出最后一个有效数据的位置,通过 sig_coeff_flag 和 coeff_abs_level 可计算出有效残差值位置的信息.

基于以上考虑,本设计提出一种动态的码表技术,如图 4(b)所示.在解码过程中,每个块根据当前块数据标志信息和位置信息动态预测后续码流中实际需要解码的语法元素位置信息,并更新对应的动态码表.在后面解码对应的语法元素时,仅需要从表中读取需要解码的位置信息,而不需要再对每个位置进行分析判断语法元素是否存在,以此大幅度减少码流解码过程的分析过程.在实现中,采用三组寄存器作为动态码表,在解码过程中分别存储有效符号位置信息,以图 4(b)为例,有效残差值位置信息和残差幅度大于 2 的位置信息. 通过该预处理机制, 可以在协议分析过程中预先剔除无效位置,使解码器始终处于解码计算状态.

通过以上预处理技术,可有效降低变换块和残差块中无效数据位置的处理过程,大幅度压缩实际解码时间.处理效果如图 5 所示.

2.3 语法元素预测机制

对于非变换块部分的语法元素,仍存在大量的图 1(b)所示情形,下一个语法元素类型的选择取决于前一个语法元素的最终结果,在此切换期间存在着较多的计算单元空闲状态.针对此情况,本设计采用转移预测的思想进行优化.

视频文件的起始帧均为 I 帧, 一般来讲对于一个视频文件, 尤其是在高分辨率高帧率的应用下,相邻图像间和相邻块之间差异很小,因此相同语法元素连续出现的概率较高.基于以上思想,本设计中增加了一组语法元素记录器,对涉及条件分支的判

图 5 动态码表预处理后的效果

Fig.5 Result after dynamic codebook processing

断的语法元素共记录 3 次,如图 6 所示.

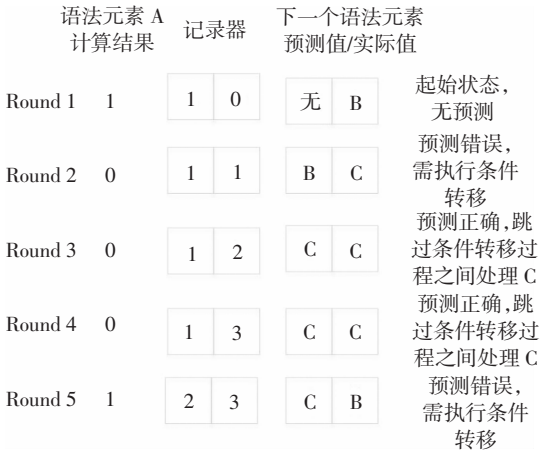


图 6　语法元素预测机制

Fig.6　Mechanism of syntax element prediction

对于起始状态,无转移预测,从第二次开始进行条件转移预测,预测成功时跳过图 1(b)中的条件判断过程,预测失败时进行条件转移判断.记录器根据当前记录结果进行预测,记录值大的预测可能性更高,相同时,下次预测以本次实际值为方向.基于以上算法, 可进一步降低语法元素间的串行依赖性,降低计算资源的空闲率.

3　测试结果

本设计提出全系统 CABAC 解码器, 整个解码器完成从码流分析,参数集解析,到全部语法元素解码并输出全部过程. 以 Altera Stratix III 为平台进行测试.

整个解码系统以参考软件 HM10.0 作为参照进行解码正确性比对, 完成 7 种不同 1080HD@60 fps 码流的解码测试如表 1 所示.

表 1　本设计在不同码流下测试结果

Tab.1　Test result on different streams

视频源	码流特征	测试帧数	原始结果 Bins/cycle	动态码表结果 Bins/cycle	最终结果 Bins/cycle
Bluesky	Low_delay	120	0.332	0.456	0.655
Station	Low_delay	40	0.294	0.385	0.645
Station	Low_delay	20	0.294	0.417	0.645
Station	Intra	10	0.446	0.518	0.703
Bluesky	Intra	30	0.427	0.501	0.703
Station	MainStill	1	0.535	0.649	0.828
Bluesky	Randomaccess	180	0.323	0.400	0.608

原始结果为未进行优化时解码器性能. 由以上测试可以看出,采用动态码表去除冗余空拍后可以节省 20%左右的时钟周期,采用语法元素预测机制最终实现高于 0.6bin/cycle 的解码速度.经过仿真测试,平均每个编码单元仅需 85 个时钟周期,远低于文献[11]中实现的效果.文献[7]虽然实现很高的解码能力,但是代价是大量的解码引擎资源浪费,平均到每个解码单元的处理能力并不高.另外文献[12–14]提出不同的解码器实现,都需要 100 MHz 以上的时钟频率完成 1920×1080@30 fps 的视频解码,而本实现仅需不超过 40 MHz 的时钟即可完成 1920×1080@60 fps 的视频解码, 完全满足 HEVC Level4.1 Main Tier 的速度要求. 本设计在 FPGA 上最高可实现 65 MHz 的解码速度, 每秒完成 90 帧的 1080HD 视频解码. 表 2 所示为本设计与之前设计的实现结果.本设计未能及时流片,DC 65 nm 工艺综合速度为 95 MHz,可满足 1920×1080@120 fps 的视频解码需求.

表 2　本设计与其他设计性能比较

Tab.2　Comparison between this design and other designs

	CSVT[12]	CSVT[13]	ISCAS[14]	SCIENCE[11]	JSSC[7]	This work
Target	1920×1080@30 fps	1920×1080@30 fps	1920×1080@30 fps	QFHD	QFHD	1920×1080@90 fps
Frequency	105 MHz	225 MHz	264 MHz	483 MHz	125 MHz	65 MHz
Bins/cycle	0.86(2cores)	0.25	1.83(2cores)	1	24.11(80 cores)	0.6
Average cycle/MB	N/A	N/A	N/A	500	N/A	85

4 结 论

本文提出一个低冗余的 HEVC 全系统 CABAC 解码实现方案，实现高效率的 HEVC 熵解码器,提高了时钟利用效率.本设计中,预测处理和并行化设计起到了重要作用，实现一个全系统 CABAC 解码器,可以以 40 MHz 时钟完成 1920*1080 @60 fps 的视频解码，完全满足 HEVC Main Profile Level 4.1 Main Tier 的速度要求.

参考文献

[1] 朱敏,刘雷波,魏少军. 一种 CABAC 解码引擎的芯片实现[J]. 电路与系统学报,2013,18(2):6—11.
ZHU M,LIU L B,WEI S J. An implementation of CABAC decoding engine[J]. Journal of Circuits and Systems,2013,18(2):6—11.(In Chinese)

[2] HUANG K,MA D,YAN R J,*et al*. High throughput VLSI architecture for H.264/AVC context -based adaptive binary arithmetic coding (CABAC) decoding [J]. Journal of Zhejiang University (Science C),2013,14(6): 449—463.

[3] 田晓华. 一种高吞吐率 CABAC 硬件编码器设计[J]. 武汉理工大学学报(信息与管理工程版),2013,35(4): 490—495.
TIAN X H. A high throughput CABAC encoder design [J]. Journal of Wuhan University of Technology (Information & Management Engineering),2013,35(4): 490—495.(In Chinese)

[4] YU W,HE Y. A high performance CABAC decoding architecture [J]. IEEE Transactions on Consumer Electronics,2005,51 (4): 1352—1359.

[5] YANG Y C,LIN C C,CHANG S C,*et al*. A high throughput VLSI architecture design for H.264 context -based adaptive binary arithmetic decoding with look ahead parsing [C]//Multimedia and Expo,2006 IEEE International Conference. Toronto,Ont: IEEE , 2006: 357—360.

[6] ZHENG Y,ZHENG S B,HUANG Z H,*et al*. A time and storage optimized hardware design for context -based adaptive binary arithmetic decoding in H.264/AVC [C]//Multimedia and Expo, IEEE International Conference. Beijing: IEEE,2007: 1567—1570.

[7] SZE V,CHANDRAKASAN A P. A highly parallel and scalable CABAC decoder for next generation video coding[J]. IEEE Journal of Solid-state Circuits,2012,47(1):8—22.

[8] GONZALEZ R C,WOODS R E. 数字图像处理[M]. 第三版.北京: 电子工业出版社,2011.
GONZALEZ R C,WOODS R E. Digital image processing [M]. Third Edition. Beijing: Publishing House of Industry,2011. (In Chinese)

[9] CHEN J W,LIN Y L. A high -performance hardwired CABAC decoder for ultra-high resolution video [J]. IEEE Transactions on Consumer Electronics,2009,55(3):1614—1622.

[10] 蒋洁. 利用平滑区域检测的 HEVC 帧内编码快速算法[J]. 西安电子科技大学学报,2013,40(3):194—200.
JIANG J. Fast intra coding algorithm using smooth region detection for HEVC [J]. Journal of Xidian University,2013,40(3):194—200. (In Chinese)

[11] SONG R,CUI H,LI Y,*et al*. High -efficiency pipeline design of binary arithmetic encoder[J]. Science China Information Sciences, 2014,57(9): 1—8.

[12] YANG Y C,GYO J I. High-throughput H. 264/AVC high-profile CABAC decoder for HDTV applications [J]. IEEE Transactions on Circuits and Systems for Video Technology,2009,19 (9):1395—1399.

[13] YI Y S,PARK I C. High-speed H. 264/AVC CABAC decoding[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2007,17(4):490—494.

[14] LIAO Y H,LI G L,CHANG T S. A high throughput VLSI design with hybrid memory architecture for H. 264/AVC CABAC decoder [C]//Circuits and Systems ISCAS,Proceedings of 2010 IEEE International Symposium. Paris: IEEE,2010:2007—2010.