

H.264 标准中的 CAVLC 编码算法与 FPGA 实现

张玲,李芳,何伟

ZHANG Ling,LI Fang,HE Wei

重庆大学 通信工程学院,重庆 400030

Department of Communication,Chongqing University,Chongqing 400030,China

E-mail:leefang_cqu@126.com

ZHANG Ling,LI Fang,HE Wei.CAVLC and its FPGA realization for H.264.Computer Engineering and Applications, 2008,44(17):78-81.

Abstract: CAVLC is adopted as a entropy coding method in baseline and extended profile in H.264/AVC standard,but the detailed syntax on which is not provided.A profound analysis on the CAVLC coding algorithm in H.264 standard is performed based on the principle of CALVC decoding method.A high-speed and low power-consumption CAVLC coder for H.264 standard is presented according to the former analysis.And the detailed design and FPGA realization method on each sub-blocks are also concerned.Besides,some optimizations on algorithm and architecture of complex sub-blocks are also performed to reduce complexity of hardware realization.Finally,FPGA verification and realization indicates that the maximum coding system clock can up to 100 MHz,which can adequately meet the needs of some high-speed and real-time applications.

Key words: H.264;CAVLC;entropy coding;encoder;FPGA

摘要:H.264 视频编码标准在基本档次和扩展档次采用 CAVLC(基于上下文的自适应可变长编码)熵编码方法,但标准并未给出详细的 CALVC 编码句法。从 CALVC 的解码原理出发,详细分析了 H.264 视频编码标准中的 CAVLC 编码算法,提出了一种应用于 H.264 标准的快速低功耗 CAVLC 编码器结构,给出了各个功能模块的详细设计原理与 FPGA 实现方法,并对较复杂的几个模块进行了算法和结构上的优化,降低了实现的复杂度。FPGA 实验验证表明,该方案编码系统时钟可达 100 MHz,能满足对高速、实时应用的编码要求。

关键词:H.264 标准;CAVLC;熵编码;编码器;FPGA

DOI:10.3778/j.issn.1002-8331.2008.17.024 **文章编号:**1002-8331(2008)17-0078-04 **文献标识码:**A **中图分类号:**TN918.81

1 引言

H.264/AVC 是 ITU-T 和 MPEG 组织共同推出的最新一代视频压缩标准,其压缩率较 H.263 和 MPEG-4 Simple Profile 有显著提高。在相同的重建图像质量下,H.264 比 H.263 节约 50%左右的码率。因其更高的压缩比、更好的 IP 和无线信号适应性,在数字视频通信和存储领域得到了越来越广泛的应用。

H.264 标准以编解码运算复杂度的增加,来换取对视频压缩性能上的提升,然而,随着实时、高清视频应用的日益广泛,采用专门的 FPGA 硬件来实现一些计算相对密集,复杂度较高的算法,对于实现实时 H.264/AVC 标准清晰度(SD)或者高清清晰度(HD)分辨率的编解码应用方案具有很高的实用价值。

近年来,国内外虽有较多文献致力于研究和设计适于 H.264/VAC 标准的快速算法或硬件加速方案,但大多是关于预测和解码方面的研究^[3-6],专注于 H.264/AVC 熵编码方面的研究文献尚不多见。本文主要研究 H.264 标准中的熵编码算法及高

速熵编码器的 FPGA 实现。

H.264 标准支持两种熵编码方案,一种是基于上下文的自适应可变长编码(CAVLC)的低复杂度技术,一种是计算要求更高的基于上下文的自适应二进制算术编码(CABAC)算法。CAVLC 是 H.264 标准的基本熵编码方法,被基本档次和扩展档次所采用。

文献[7]采用直接映射的方式实现了 CAVLC 编码,但由于未对算法进行优化,造成硬件资源的浪费,同时系统的编码速率也仅能达 32 MHz;文献[8]采用了两级流水方式实现对两个 4×4 块的并行处理,虽然编码速率可达 100 MHz,但硬件资源消耗达 23.6 K 个逻辑单元;文献[9]针对移动环境下的视频传输提出了一种 CAVLC 硬件实现方案,虽然消耗硬件资源较少,但编码速率仅达 50 MHz;本文从解码原理出发,详细分析了 CAVLC 编码算法原理,并对编码算法进行了合理的优化,综合采用了流水处理与并行处理技术,提高了系统编码速率,并采

作者简介:张玲(1964-),女,副教授,主要研究领域为图像信息处理,智能信号处理以及专用芯片设计等研究;李芳(1981-),女,在读研究生,主要研究领域为图像信息处理,基于 FPGA 的嵌入式系统设计等应用技术研究;何伟(1964-),男,副教授,主要研究领域为集成化芯片设计,图像信息处理及嵌入式系统设计等应用技术研究。

收稿日期:2007-09-17 **修回日期:**2007-12-17

用算术运算替换部分静态码表,节约了系统的存储资源消耗。

2 CAVLC 算法编码原理

H.264 规定了基本档次、主要档次和扩展档次等三种档次,每个档次都支持一组特定的编码功能,并支持一类特定的应用。其中,基本档次利用 I 片和 P 片支持帧内和帧间编码,支持基于上下文的自适应变长熵编码(CAVLC);主要应用于可视电话、会议电视、无线通信等实时视频通信^[1,2]。

如图 1 所示,H.264 编码器主要由帧间预测(运动估计 ME,运动补偿 MC)、帧内预测、变换、量化以及熵编码模块组成,本文主要研究 CAVLC 熵编码器的原理与硬件实现。

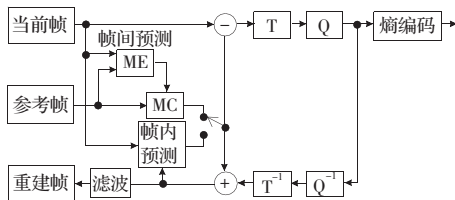


图 1 H.264 编码器框图

CAVLC 用于亮度和色度残差数据的编码,残差经过变换量化后,非零系数主要集中在低频部分,而高频系数大部分是零;量化后的数据经过锯齿形(Zig-Zag)扫描,DC 系数附近的非零系数值较大,而高频位置上的非零系数值大部分是+1和-1,并且相邻的 4×4 块的非零系数的数目是相关的,CAVLC 正是利用残差经过整数变换、量化后数据所表现的稀疏特性进行压缩,可进一步减小数据中的冗余信息;同时,CAVLC 利用相邻已编码符号的相关性,为所要编码的符号选择合适的上下文模型,从而可以大大降低符号间的冗余度。在 CAVLC 中,上下文模型的选择主要体现在两个方面,即非零系数编码所需码表的选择和拖尾系数后缀长度的更新。

与 CALVC 解码相对应,CAVLC 编码主要包含以下 5 个主要步骤:

(1)coeff_token: 对非零系数的数目(TotalCoeffs)以及拖尾系数的数目(TrailingOnes)进行编码,编码过程中根据 NC (Number Current,当前块值)值选择 6 个变长码表之一进行编码,其中 NC 值依赖于当前块上面 4×4 块的非零系数数目 NA 和当前块左边 4×4 块的非零系数数目 NB,以及当前输入块的系数类型进行确定,这也正体现出基于上下文的编码思想。具体如表 1 所示。

表 1 NC 值确定方式

| NA | NB | NC |
|-------------|----|-------------------|
| √ | √ | $(NA+NB+1) \gg 1$ |
| √ | × | NA |
| × | √ | NB |
| 色度直流系数=(-1) | | -1 |

(2)trailing_ones_sign_flag: 对每个拖尾系数的符号进行编码,+1 编码为‘0’,-1 编码为‘1’。

(3)levels: 对除了拖尾系数之外的非零系数的幅值(Levels)进行编码,这是 CAVLC 编码的难点,也是编码运算复杂度最集中的部分。非零系数的幅值编码由前缀(level_prefix)和后缀(level_suffix)两部分组成,levelSuffixSize 和 suffixLength 是编

码过程中需要使用的两个变量,用以表征 level_suffix 的比特位数,在一般情况下,levelSuffixSize 和 suffixLength 的值相等,但有如下两种意外:

①if level_prefix==14 && suffixLength==0
levelSuffixSize=4;

②if level_prefix>=15
levelSuffixSize=level_prefix-3;

变量 suffixLength 的更新是基于上下文模式自适应的,suffixLength 的更新与当前的 suffixLength 值以及当前已编码完的非零系数值有关,其初始化按如下句法进行:

if TotalCoeff>10 && TrailingOnes<3

suffixLength=1;

else

suffixLength=0;

end

在后续编码过程中则需要根据非零系数的幅值与标准所提供的阈值表的关系自适应的更新 suffixLength,具体句法描述如下:

if suffixLength==0

suffixLength=1;

else if abs(level(n))>(3<<(suffixLength-1)) && suffixLength<6

suffixLength=suffixLength+1;

end

end

(4)total_zeros: 对最后一个非零系数前零的数目(TotalZeros)进行编码,该非零系数为正向扫描的最后一个系数,可根据相应参数查找变长表编码。

(5)run_before: 对每个非零系数前零的个数(RunBefore)进行编码,编码按照反序进行,从最高频的非零系数开始,但需要注意其编码的终止条件:

①逆序最后一个非零系数前零的个数不需编码;

②如果没有剩余的零需要再继续编码,即 $\sum RunBefore = TotalZeros, ZeroLeft[n]=0$ 时,停止编码。

上述终止条件可用句法如下描述:

if RunBefore(n)==ZeroLeft(n) && n==length(RunBefore)||ZeroLeft(n)==0

break;

else

run_before;

end

3 CAVLC 硬件实现

所提出的 CAVLC 编码器结构框图如图 2 所示,主要包含 Zig-Zag、控制模块、参数统计与编码预处理、NC、Coeff-Token、Levels、TotalZeros、RunBefore、码字组合与输出控制以及编码时钟控制等 10 个功能模块组成。

在设计中对 CAVLC 的编码器结构按照功能进行了合理划分,以便于实现流水处理,提高系统的处理速度,同时,由于 CAVLC 为变长编码的扩展,各个句法元素的编码长度不确定,因此在设计中对各个句法的编码码字采用最大可能字长方式并行输出(*_code),并提供相应的数据状态指示信号

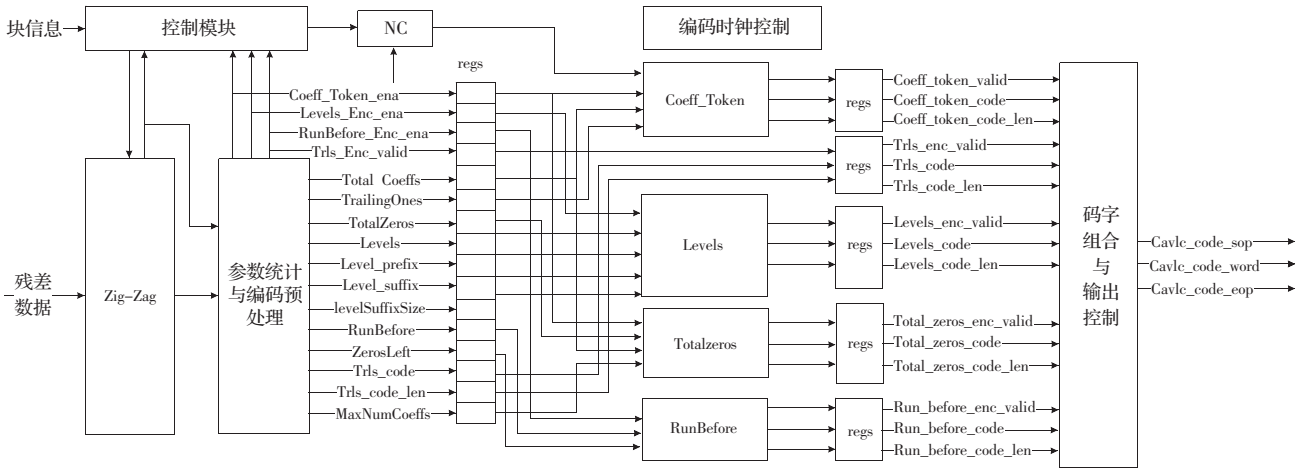


图2 CAVLC 编码器硬件结构框图

(*_enc_valid)以及当前句法的编码码字长度(*_code_len),下面根据第2部分分析的 CAVLC 算法的编码原理,具体介绍主要功能子模块的设计原理与优化方法。

(1) Zig-Zag

CALVC 实现对亮度和色度残差数据的编码,残差经过变换量化后的非零系数呈现的系数特性是 CALVC 实现数据压缩的基础,但在进行正确的编码操作前,需要对量化后的数据进行锯齿形(Zig-Zag)扫描,以使得 DC 系数附近的非零系数值最大,而高频位置处的非零系数值大部分是+1 和-1,因此 Zig-Zag 是实现 CAVLC 正确编码的前提。在具体实现上采用 DP-DRAM 对输入的残差数据块进行顺序重排。

(2)参数统计与编码预处理

与解码相对应,CALVC 需要对 TotalCoeff、TrailingOnes、TrailingOnes_Sign_Flag、Levels、TotalZeros 以及 RunBefore 等 6 个句法元素进行编码,该模块即完成对上述参数的统计以及部分参数的直接编码输出。具体结构如图3所示。

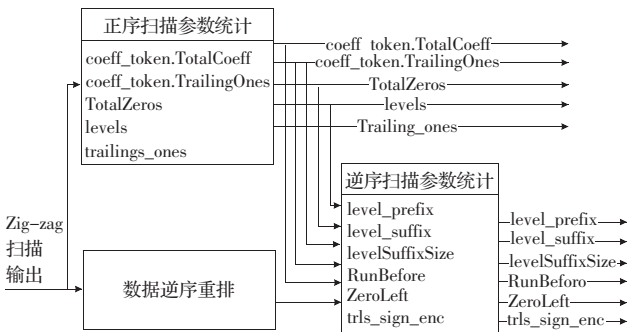


图3 参数统计与编码预处理模块结构框图

Zig-Zag 扫描输出的残差数据一方面送入正序扫描参数统计模块对非零系数的数目 (TotalCoeff)、拖尾系数的数目 (TrailingOnes)、最后一个非零系数前零的数目 (TotalZeros) 以及相应的非零系数和拖尾系数进行统计输出;另一方面,由于 RunBefore 和 ZeroLeft 的统计需要逆序扫描,因此还需要对残差数据进行逆序扫描,与此同时可以对当前模块的拖尾系数符号进行编码输出,以减小系统的编码时延。

level_prefix、level_suffix 和 levelSuffixSize 的计算是本模块设计的难点所在,也是 CALVC 编码系统中计算复杂度密集的部分,根据第 2 部分的算法原理描述,给出具体的计算过程如图 4 所示。

部分,根据第 2 部分的算法原理描述,给出具体的计算过程如图 4 所示。

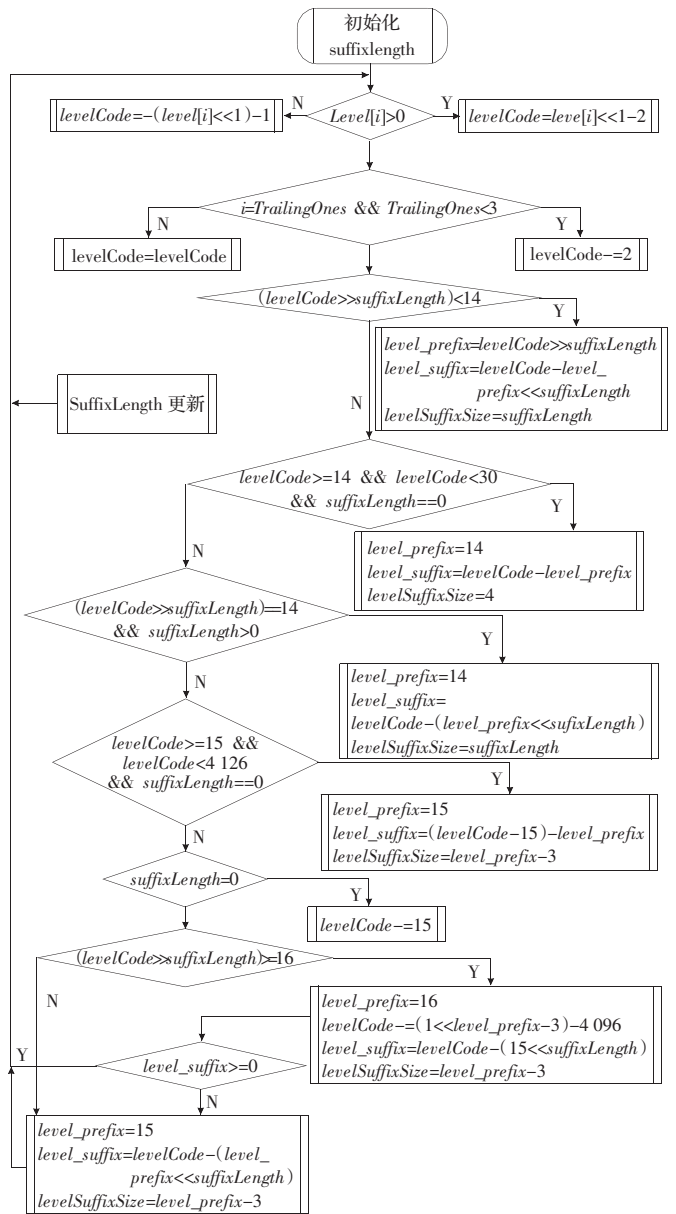


图4 Levels 编码参数计算过程

(3)NC

NC用于在进行coeff_token编码时选择相应变长码表,具体选择方式如表1所示,同时,该模块还要缓存相关已编码块的非零系数个数,以获取后续编码块的NC值。

(4)Coeff-Token、Levels、TotalZeros、RunBefore

在完成了相关编码参数的统计和计算之后,后续的编码处理则可以根据参数查找相应的变长码表实现编码,同时提供相应的编码状态指示信号(*_enc_valid)以及当前句法的编码码字长度。

由变长编码所固有的伸缩特性,各个模块编码输出的码字长度存在不一致性,其中,Coeff-Token的最大码长为16位,TotalZeros的最大码长为9位,RunBefore的最大码长为11位,而Levels编码复杂度高,其码字长度具有不确定性。如图5给出了非零系数幅值变化情况下,编码长度随suffixLength的变化关系,其中非零系数的幅值取自范围为[-600~600],suffixLength的取自范围为[0~6],对应H.264标准规定的阈值递增表。可以看出,随着阈值的增加,Levels编码长度呈现缓慢正常的趋势,且编码的最大长度为28,这也正体现出使用短码长编码高概率系数,而使用长码长编码小概率系数的由变长编码所固有的伸缩特性,各个模块编码输出的码字长度存在不一致性,其中,Coeff-Token的最大码长为16位,TotalZeros的最大码长为9位,RunBefore的最大码长为11位,而Levels编码复杂度高,其码字长度具有不确定性。如图5给出了非零系数幅值变化情况下,编码长度随suffixLength的变化关系,其中非零系数的幅值取自范围为[-600~600],suffixLength的取自范围为[0~6],对应H.264标准规定的阈值递增表。可以看出,随着阈值的增加,Levels编码长度呈现缓慢正常的趋势,且编码的最大长度为28,这也正体现出使用短码长编码高概率系数,而使用长码长编码小概率系数的分配原则,使码流长度得到很好的压缩,提高对数据的压缩性能。

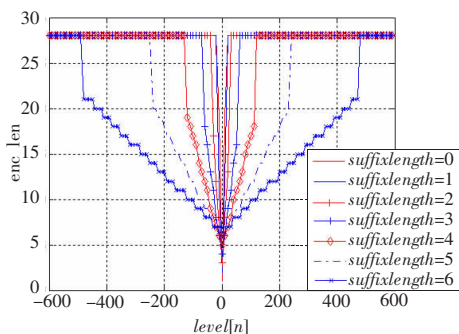


图5 Levels码长随阈值的变化关系

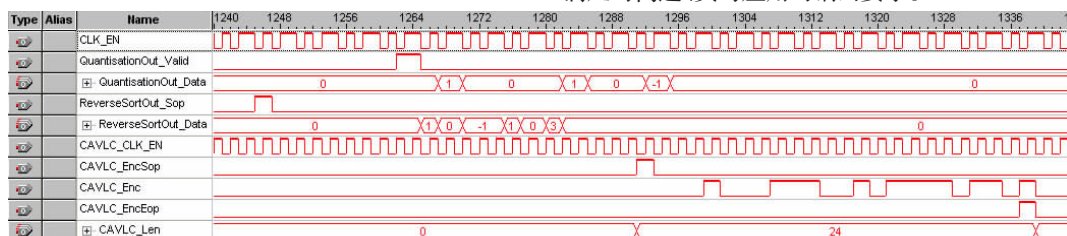


图7 CAVLC编码器硬件验证时序

(5)码字组合与输出控制

码长的变化导致各个句法要素的编码起止时刻以及残差块的输出码流长度存在不确定性,因此残差块的编码输出是异步的。该模块根据前端各编码模块的编码输出以及相应的码长指示信号对各个句法要素的码字进行组合,产生当前块的最终CAVLC码流,同时提供对应块的码流状态信息(包含CAVLC_enc_sop, CAVLC_enc,CAVLC_enc_eop和CAVLC_enc_len等)。

4 CAVLC硬件验证

使用VerilogHDL设计并实现了提出的CAVLC编码器,并使用ALTERA Quartus II对设计进行综合、适配、布局布线,最后使用ALTERA Cyclone EP1C12Q240C8 FPGA硬件对设计进行了验证,设计综合结果如图6所示,系统编码时钟高达100 MHz,能满足对HDTV(1920×1088-30 f/s)实时编码要求,与文献[7]~[9]相比,提出的硬件实现方案在系统性能和硬件资源上取得了更好的平衡。

| | |
|-------------------------|--|
| Flow Status | Successful - Wed Aug 15 17:50:12 2007 |
| Quartus II Version | 5.1 Build 176 10/26/2005 SJ Full Version |
| Revision Name | H264_CAVLC_EntropyEncode |
| Top-level Entity Name | H264_CAVLC_EntropyEncode |
| Family | Cyclone |
| Device | EP1C12Q240C8 |
| Timing Models | Final |
| Met timing requirements | No |
| Total logic elements | 9,416 / 12,060 (78 %) |
| Total pins | 32 / 173 (18 %) |
| Total virtual pins | 0 |
| Total memory bits | 58,198 / 239,616 (24 %) |
| Total PLLs | 1 / 2 (50 %) |

图6 CAVLC编码器综合结果信息

如图7为输入4×4残差块[0 3 -1 0; 0 -1 1 0; 1 0 0 0; 0 0 0],NC为0时的CAVLC编码测试,其输出码流为[0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 1 1 1 1 0 1 1 0 1],码流长度为24比特,这与理论编码结果是一致的。该时序为ALTERA公司SignalTap II嵌入式逻辑分析仪测试的硬件实时时序,验证了硬件设计的正确性。

5 结束语

本文从CAVLC的解码原理出发,对H.264视频编码标准中的CAVLC编码算法作了详细的分析,提出了一种应用于H.264标准的快速低功耗CAVLC编码器结构,并对各功能子模块的设计原理与方法进行了详细的论述。最后通过FPGA硬件实现,验证了设计的正确性,同时,编码器的系统时钟高达100 MHz,能满足对高速、实时应用的编码要求。