

高效的 H. 264 指数哥伦布解码器设计

颜 明, 陈咏恩

(同济大学 通信软件及专用集成电路设计中心 上海 200092)

摘 要: H. 264 视频压缩标准凭借高压缩比和较好的图像质量, 已经作为一种新型的标准被广泛接受。由于 H. 264 的解码复杂度很高, 软件实现难以满足实时性的要求, 所以需要采用硬件解码。本文提出了一种针对 H. 264 视频编码标准的可变长指数哥伦布解码的硬件设计结构, 给出了一种系统解码时间消耗与系统资源占用较少的硬件设计方案, 最后给出了设计最终的仿真以及后端设计的结果。

关键词: H. 264 标准; 指数哥伦布码; 移位寄存器; 优先级编码器

中图分类号: T P274

文献标识码: B

文章编号: 1004 - 373X(2007) 03 - 071 - 03

Design of Efficient Exp - Golomb Decoder for H 264

YAN Ming, CHEN Yong'en

(Design Center of Communication Software and ASIC, Tongji University, Shanghai, 200092, China)

Abstract: Based on its high compression ratio and better image quality, the new standard of video compression, H. 264 has been widely accepted and used in many applications. Because the decoding of H. 264 is very complex, software implementation can't decode the video in real-time. We implement the H. 264 decoder in hardware. We propose a hardware architecture of Exp - Golomb decoder for H. 264. This design has less decoding time and uses less logical resource. Finally the simulation and back end design results are also presented in this paper.

Keywords: H. 264; Exp - Golomb; shift register; priority decoder

1 引言

ITU - TVCEG(视频编码专家组)和 ISO/IECMPEG(运动图像编码专家组)两个专家组联合组成的 JVT(联合视频组), 相互协作、共同研究开发出了新一代视频编码算法——H. 264, 同时也被称为 MPEG - 4 的第十部分或者 MPEG - 4 AVC(高级视频编码标准)。1998 年 1 月开始草案征集, 1999 年 9 月, 完成第一个草案, 2001 年 5 月制定了其测试模式 TML - 8, 2002 年 6 月的 JVT 第 5 次会议通过了 H. 264 的 FCD 板。2003 年 3 月正式发布。

H. 264 标准采用了大量最新的视频编码技术, 获得了令人满意的图像质量, 但是其压缩效率的提高也是以压缩算法复杂度的提高为代价的, 其解码复杂度大约是 MPEG - 2 的 2~3 倍, 这造成了单纯用软件方法解码难以达到很高的性能。尤其对实时性要求较高的应用中, 高清晰度数字视频难以实现实时解码播放。这样应用环境下就需要利用硬件加速或者设计专门的硬件解码电路以达到实时解码。H. 264 专用的硬件解码电路使得视频码流不仅能在计算机上进行解码, 还能在数字电视机、DVD 机等设备上播放, 其应用范围非常广泛。

H. 264 在 entropy 编码上提供 CAVLC(Context Adaptive Variable Length Coding)与 CABAC(Context Adaptive Binary Arithmetic Coding)两种编码方式。Base

line profile 规格的 entropy 编码是将剩余区块以 CAVLC 编码, 其他信息则以 Exp - Golomb VLC 编码。CAVLC 比传统的 Run - Length 编码更能有效消除统计上之重复性, 而 Exp - Golomb VLC 之特点为规则、简单。Main profile 规格的 entropy 编码则是将信息全部采用 CABAC 编码, 以得到更高的压缩率。

2 指数哥伦布算法简介

Exp - Golomb 是一种使用一定规则构造码字的可变长编码。他的具体算法如下:

(1) 对于一个值为 code 的变量, 通过式(1)确定 i 的值:

$$\sum_{j=0}^{i-1} 2^{j+k} \leq \text{code} < \sum_{j=0}^i 2^{j+k} \quad (1)$$

(2) 用以“0”为符号的一组数表示 i 。即 i 个“0”组成码字的前缀, 其中“0”只是符号, 只有个数有意义。

(3) 插入一个分隔符“1”。与第(2)步相同, “1”只是符号, 只需与第 2 步的符号不同即可。

(4) 用式(2)计算出二进制的码字尾部:

$$\text{code} - \sum_{j=0}^{i-1} 2^{j+k} \quad (2)$$

表 1 列出 $k=0$ 的头 9 个码字, 这也是 H. 264 的采用的参数。从这里可以清楚地看出码字的构造遵循一种逻辑规则。每一个码字按如下方式构造:

[M zeros][1][INFO]

INFO 为 M 比特, 包含有效信息。第一个码字不包含前缀零和后缀零; 码字 1 和 2 有一个 1 比特的 INFO 场; 码字 3~ 6 有 2 比特的 INFO 场, 依次类推。每个码字的长度为 $(2M + 1)$ 比特。

$$M = \log_2(\text{code_num} + 1) \quad (3)$$

$$\text{INFO} = \text{code_num} + 1 - 2^M \quad (4)$$

表 1 $k = 0$ 时 Exp-Golomb 码字

Code_num	Codeword
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000
8	0001001
...	...
...	...

一个待编码的参数 v 映射到 code_num , 有 3 种映射方式:

(1) $ue(v)$: 无符号直接映射, $\text{code_num} = v$ 。应用在宏块类型、参考帧索引等参数上。

(2) $se(v)$: 符号映射, 应用于运动矢量差值, 量化参数差值等, 映射规则如下:

$$\begin{aligned} \text{code_num} &= 2|v| & (v < 0) \\ \text{code_num} &= 2|v| - 1 & (v > 0) \end{aligned}$$

(3) $me(v)$: 标记映射, 参数 v 根据标准中定义的一张表格映射到 code_num 。这种映射方式用于 $\text{coded_block_pattern}$ 参数。

每一种映射方式 (ue, se, me) 都被尽量设计成对出现频率高的值使用较短的码字而对出现频率低的值使用较长的码字。例如, 宏块模式 Pred_L0_16x16 被映射到 code_num_0 , 因为这种预测方式出现频率高, 而 Pred_8x8 被映射为 code_num_3 , 因其出现频率较低。

传统的 H. 264 解决方案将熵编码归入软件解码部分, 如图 1 所示。

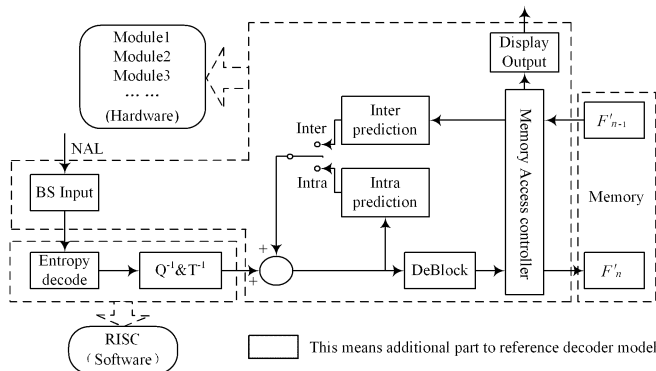


图 1 H. 264 解码过程的软硬件划分

对 H. 264/ AVC 解码器的复杂度分析显示, 运动补偿和去块效应这两个部分是解码器最费时的两个模块, 大概占了解码器 CPU 时间的一半以上。由于 H. 264/ AVC 的高复杂度, 即使把这两个模块用硬件来实现并在硬件系统上达到了高清分辨率的解码速率要求, 基于 RISC CPU 的软件解码部分却不能达到解码指标。要想使整个 SoC 解决方案能够完成高清分辨率的解码速率, 就把原来用软件实现的一些模块用硬件实现。比如现在的软硬件划分里面处于软件解码的 IDCT, 熵解码 (包括 CABAC, CA VLC), 这些模块都是可以用硬件实现来提高解码速度。

3 硬件设计

变长码即采用不同的位数对数据进行编码对那些经常出现的数据指定较少的位数表示, 而那些不常出现的数据指定较多的位数表示。从总的效果看还是节省了存储空间, 这样编码得到的代码码长就是不固定的, 故称为变长码。与定长码相比变长码提高了压缩率, 但是其码长由其码字内容决定, 他的解码不能像解定长码那样预先就知道确定的码长, 而是需要逐位读取码字来确定码长后才能解析出码字的值, 这样一来对硬件处理来说既浪费了时间节拍, 后果是可能引起不能实时解码, 同时复杂了解变长码部件的控制, 后果是所需硬件资源的增加。本文针对指数哥伦布码提出了一种优化的硬件结构, 以保证解码的实时性。

3.1 码流读入模块 (Bits Stream Reader)

如果码流直接引入解码计算模块, 很可能在解码当前参数时, 不能准备好解码下个码字的码流。因为只有获得当前码字长度, 才能继续下一个码字的解码。所以在码流接入后, 引入一个缓冲模块 (如图 2 所示)。由于 H. 264 的码字长度不会超过 32 b, 所以寄存器 A, B, C 的长度都为 32 位。寄存器的读取由输出控制器 (output controller) 决定。外部的累加器计算码字长度的总和。当码字长度小于 32 b 时, 将 A 或 B 中的值移入 32 位外部寄存器, 填充 C; 当码字长度大于 32 b, 小于 64 b, 填充 A, 同时读取 B, C; 大于 64 b, 填充 B, 读取 A, C。总之, 要保证 32 位外部寄存器已解码的数据移出, 留下的都是未解码的数据。

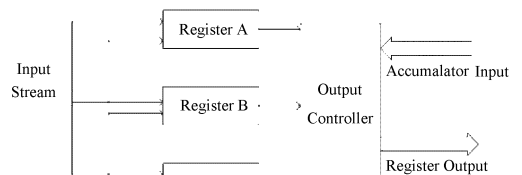


图 2 码流读入模块系统框架

3.2 长度检测模块 (Length Detector)

由于 H. 264 的码字不超过 32 位, 同时它具有对称性, 所以表示长度的全 0 位数不会超过 16 位。把最低的

16 位输入长度检测模块, 检测最先出现的‘1’的位置。这个模块主要包括 2 个优先级编码器。一个用来检测输入的 4 位组哪一组具有第一个‘1’另一个用来检测这 4 个位哪一位具有第一个‘1’。然后把 2 个结果综合起来, 就得到第一个‘1’位置 M , 码字长度 Len 根据式(5) 得到。

$$Len = 2 * M + 1 \quad (5)$$

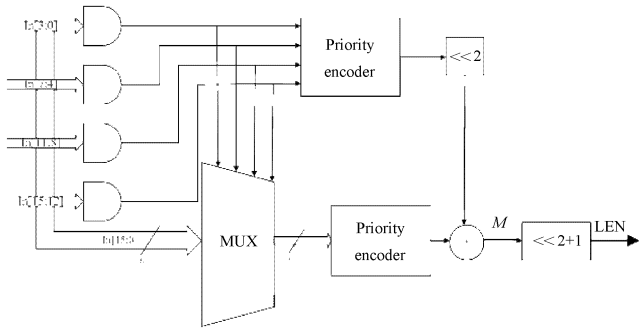


图3 长度检测模块系统框架

3.3 系统整体设计

如图4所示, 码流读入模块将 32 位数据填入移位寄存器, 移位寄存器的前 16 位输入长度检测器, 得到码字长度。码字的长度送入累加器, 提醒输入模块更新移位寄存器。

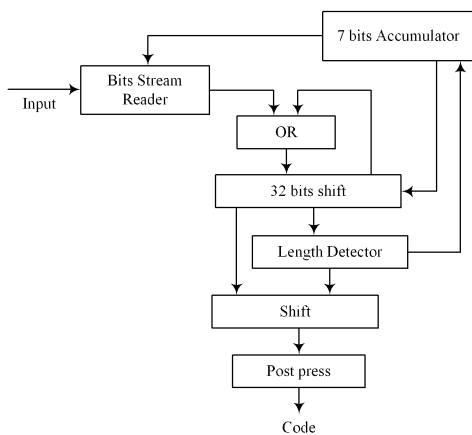


图4 硬件整体框架

同时, 寄存器将码字存入一个临时寄存器。后处理模块主要是根据编码类型来解码。如果是无符号的数, 就将移位寄存器低位 $Len+1$ 位数据-1, 如果有符号数, 先

将移位寄存器低位 $Len+1$ 位数据-1, 得到数值 Val , 如果 Val 为奇数, 则将 $(Val+1)$ 左移一位; 如果为偶数, 则输出 $-Val > > 1$ 。

例如 8, 无符号编码后是 00101, 得到后 3 位 101, 减 1 得到 100, 即是 4; 如果采用有符号方法编码-3, 得到 00111, 后三位 111, 减 1 得到 110, 为偶数, 则输出 $-6/2 = -3$; 如果采用有符号方法编码 3, 得到 00100, 后 3 位 100, 减 1 得到 011, 为奇数, 则输出 11 就是 3。

4 结 语

本文提出了一种针对 H. 264 视频编码标准的可变长指数哥伦布码解码的硬件设计结构, 系统解码时间和资源占用的比较少。采用优先级编码器检测码字的长度, 有效降低了硬件的复杂度。硬件采用 Verilog HDL 语言描述, 采用 0.25 μm 的库综合后的规模在 4 000 门左右。设计的验证采用 Xilinx Vertex II 6 000 开发板, 综合频率可以达到 85 MHz。由此可以看到, 设计能够符合实时解码的要求, 为以后的工作打下了良好的基础, 具有一定的现实意义。

参 考 文 献

- [1] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Proposed Draft Errata List with Revision - Marked Corrections for H. 264/AVC, 2003, 12.
- [2] 毕厚杰. 新一代视频压缩编码标准—H. 264/AVC[M]. 北京: 人民邮电出版社, 2005.
- [3] Wn Di, Gao Wen, Hu Ming Zeng, et al. An Exp - Golomb Encoder and Decoder Architecture for JVT/AVS, IEEE 5th International Conference Proceeding, 2003.
- [4] Yong Ho Moon, Gyu Yeong Kim, Jae Ho Kim, An Efficient Decoding of CAVLC in H. 264/AVC Video Coding Standard. IEEE International Conference on Multimedia and Expo, 2004.
- [5] Min K Y, Clouig J W. A Motion Efficient VLC Decoder Architecture for MPEG - 2 Application[J]. IEEE Workshop on Signal Processing Systems, 2000.
- [6] Clouig S B, Lee M H. High Speed Pattern Matching for a Fast Huffman Decoder[J]. IEEE Trans on Consumer Electronics, 1995, 41(11): 97 - 103.

作者简介 颜 明 男, 上海人, 2004 年在同济大学电子与信息工程学院获得电子信息工程专业学士学位, 现为同济大学电子与信息工程学院信号与信息处理专业研究生。主要研究方向为多媒体编解码以及集成电路芯片设计。

陈咏恩 男, 教授, 1983 年 1 月在德国亚琛大学电子工程系通信工程专业获得德国工程博士学位, 博士研究成果获得德国 Borchers 奖, 1987 年任同济大学教授, 博士生导师, 兼同济大学通信软件及专用集成电路设计中心主任。