H.265 视频编码器在 TMS320C6678 上的优化实现

刘贤梅,任 重
(东北石油大学 计算机与信息技术学院,黑龙江 大庆 163318)

摘 要:H.265 是新一代视频编码标准,与第二代视频标准 AVS 和 H.264 相比可提升 1 倍的编码效率,但其复杂度也随之提升了 4 倍以上,给 H.265 的 DSP 嵌入式实时实现带来挑战。文中为实现 DSP 实时视频编码,通过对 H.265 视频编码算法复杂度进行分析,并基于 TI 公司的 TMS320C6678 DSP 芯片,从使用快速算法、针对芯片特点进行结构优化和使用汇编加速指令及重新安排指令流水线三个层次进行优化实现。实现结果表明,所提出的优化方法可以获得 25 倍的视频编码速度的提升,实现了 D1 分辨率视频的 H.265 实时编码处理。该方法可进一步扩展获得更高的编码速度的提升。

关键词:TMS320C6678;H.265;汇编优化;结构优化
中图分类号:TP391 **文献标识码**:A **文章编号**:1673-629X(2015)03-0171-04
doi:10.3969/j.issn.1673-629X.2015.03.039

Optimized Realization on TMS320C6678 of H.265 Video Encoder

LIU Xian-mei, REN Zhong
(College of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China)

Abstract: The H.265 standard is the newest video coding standard, which can increase one time compared with H.264 and AVS which is the second generation video standard, but the time complexity is also improved more than 4 times, which brings a challenge to the real-time implementation of DSP for H.265. In this paper, to achieve the real-time video encoding on DSP platform, through the analysis of complexity for H.265 video encoding algorithm, and based on the TMS320C6678 DSP chip of TI company, the optimized implementation is carried out from three levels that are including conducting structure adjustment in accordance with chip feather, using assembly accelerating instruction and rearranging the pipeline. The results show that the proposed optimization method can improve the speed of encoding 25 times, and can support H.265 real-time encoding by D1 resolution. The proposed method can be further extended to obtain higher encoding speed.

Key words: TMS320C6678; H.265; assembly optimization; structural optimization

0 引 言

随着计算机技术的飞速发展,目前主流的视频编码标准主要有 H.264 和 AVS 协议[1-3]。随着视频分辨率的不断提高和无线视频的广泛应用,对视频编码效率和网络适应性提出了更高的要求。2010 年 1 月,ITU-T VCEG(Video Coding Experts Group)和 ISO/IEC MPEG(Moving Picture Experts Group)联合成立了 JCT-VC(Joint Collaborative Team on Video Coding),统一制定下一代编码标准:HEVC(High Efficiency Video Coding),ITU 将其定名为 H.265[4-5]。H.265 编码效率比 H.264 平均提高 50%。

然而,视频标准的高压缩比是以高计算/存储复杂度为代价的,H.265 编码压缩效率较 H.264 提升 1 倍的同时对计算量的需求会增加 4 倍以上。过高的计算/存储复杂度对 H.265 视频编码的实时实现是一个巨大挑战。DSP(数字信号处理器)是一种特别适合进行数字信号处理运算的微处理器。目前,DSP 芯片的各大厂商中,TI(德州仪器)公司是技术最领先的 DSP 芯片厂商之一,其 TMS320C66x[6] 系列芯片是一款高性能多核 DSP 处理器,芯片内集成了 8 个 C66x 内核和一个网络协处理器,兼具同构和异构的特点,在最高频率 1.25 GHz 下,单核可达 40 GMAC 的定点计算性

收稿日期:2014-03-15 **修回日期**:2014-06-18 **网络出版时间**:2014-10-23
基金项目:黑龙江省教育科学技术研究项目(12511011)
作者简介:刘贤梅(1968-),女,山东日照人,教授,硕导,CCF 高级会员,研究方向为多媒体信息处理、计算机图形学;任 重(1986-),男,黑龙江大庆人,硕士研究生,研究方向为信息智能分析与处理。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20141027.1429.001.html

能和 20 GFLOP 的浮点运算速度。

文中首先针对 DSP 平台的特点进行编码器结构优化,然后分析 H. 265 算法复杂度,根据分析结果对耗时最多、优化效果最明显的模块进行优化,最后给出优化结果。

1 结构优化

根据 H. 265 编码特点[7],为充分利用 TMSC6678 DSP 平台的 8 核并行处理能力,采用 1 个核做调度,另外 7 个核做并行编码,见图 1。为降低核间通信开销,采用 Slice 级并行编码,即将一帧待编码图像等分为 7 个 Slice,每个核独立编码一个 Slice。采用 Slice 级编码而不采用 CU 级或帧级编码的原因主要是因为:

(1)Slice 介于帧和 CU 之间。因为视频编码各 CU(编码单元)会参考同一 Slice 中的其他 CU,因此采用 CU 级编码会极大地增加核间通讯压力。

(2)各 Slice 之间没有数据关联性,各 Slice 完全独立,待编码单元无需参考其他 Slice 即可完成编码。

(3)帧级编码对 DDR 和 Cache 有极高的要求,但 DSP 平台硬件资源有限,帧级编码无法实现。

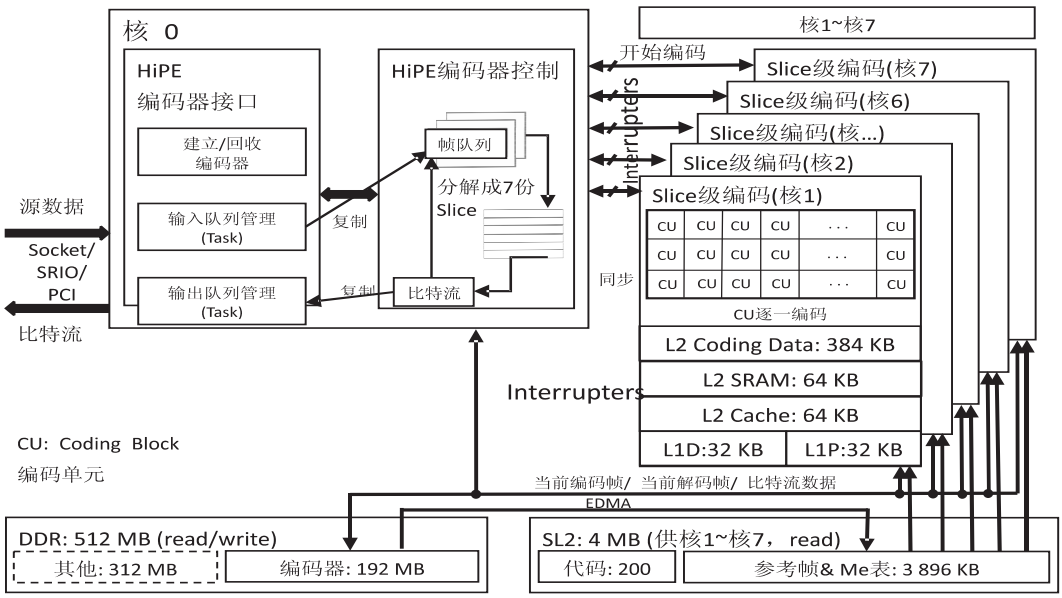

图 1 基于 TI 6678 DSP 芯片的视频编码结构

结构优化后,编码器主要工作过程如下:核 0 用于编码控制及预处理。对于待编码帧,核 0 完成初始化后,发中断给其他各核启动编码;各核完成编码后进行同步后由核 0 完成码流拼接。因为各 Slice 没有数据相关性,因此 7 个编码核独立编码,各编码核之间无依赖关系,可以使它们尽最大可能并行运行。

2 关键算法优化

2.1 算法分析

H. 265 标准与 H. 264 和 AVS 类似,仅仅规定了解码器,对编码器的运动估计和模式决策等算法不进行严格规定,可通过设计它们的快速算法来降低处理复杂度。快速算法可以实现微弱的视频编码质量降低带来编码处理速度的大幅度提高。运动估计算法是编码器中复杂度最高的部分,文中采用文献[8]中所用的改进型六边形快速搜索算法实现运动估计算法,该算法已经比较成熟。与参考代码中全搜索算法相比复杂度得到成倍的降低。

与 H. 264 和 AVS 标准相比,不同之处在于帧内和帧间预测方面。H. 265 采用了变块大小预测技术,支持{2N×2N,2N×N,N×2N,N×N}/{2n×2n,2n×n,n×2n,n×n}(其中 $n=N/2$)递归层次划分形式。虽然快速算法和结构调整可以大幅提高编码速度,但是对于 D1 每秒 25 帧的视频实时编码还不够,需要进一步对关键操作进行汇编优化,以充分利用 DSP 的指令级并行处理能力。

H. 265 使用更多的块大小和编码模式,导致编码算法复杂度较 H. 264/AVS 视频标准增加 4 ~ 5 倍。H. 265 的最终目标是比 H. 264 提高一倍的编码效率,实现这一目标的核心技术之一就是块大小变换技术。H. 264 中最大处理单元是 16×16 像素,而 H. 265 的最大处理单元达到 64×64,编码单元可以选择从最小的 8×8 到最大的 64×64[9]。由此带来的是运动估计和模式决策算法复杂度大幅度提升,运算量的增加对硬件平台提出了更高的要求。为了能够在 DSP 平台有限的资源下实现 H. 265 实时编解码处理,需要对编码器进行优化,提高运算处理效率。

通过 Intel VtuneAmplifer 工具统计热点函数及其所占用时间,可以得出编码过程耗时最多的就是运动估计(Motion Estimation,ME)和模式决策(Mode Deci-

sion,MD),占整个系统运算量的 80% ~93%(不同视频序列比例不同),详见表 1。帧内预测和帧间预测都属于运动估计,运动估计又可以分为两类:整像素运动估计和分像素运动估计。整像素运动估计(IME)是将图像序列的每一帧分成许多不重叠的宏块,先假设宏块内所有像素的位移量都相同,然后对每个宏块到参考帧中某一给定的搜索范围内,根据一定的匹配准则,找出与当前块最相似的块,即匹配块;匹配块与当前块的相对位移即为运动矢量。分像素运动估计(FME)的意义,是由于自然物体运动的连续性,相邻两帧之间的块的运动矢量不一定是以整像素为基本单位的,运动的真实位移量是 1/2 像素或 1/4 像素。IME 主要完成粗精度的帧间匹配搜索,FME 在 IME 的基础上,再对宏块分割进行 1/2 像素和 1/4 像素精度匹配搜索,最终得到宏块的编码模式和运动矢量。模式决策就是寻找最佳匹配块的过程,将每个宏块视为矩阵,通过矩阵运算,寻找差值最小的矩阵,将其对应的宏块视为最佳匹配块。

表 1　编码器各模块复杂度统计

编码算法	复杂度(所占比例)	主要基本操作
帧内预测	30% ~35%	35 种亮度预测+SAD 或 SSD 4 ~5 种色度预测+SAD 或 SSD
帧间预测	45% ~50%	通过 SAD 或 SSD 选择最优整像素,插值后再次进行 SAD 或 SSD 运算
模式决策	10%	SAD 或 SSD 运算后的编码模式决策
变换量化	4% ~5%	数值运算,蝶形算法,用加减法代替矩阵乘法,含 DCT,IDCT,QIQ
环路滤波	2% ~3%	数值运算,滤波,滤波系数
熵编码	1% ~2%	位操作、查表

整像素运动估计的主要操作是 SAD(Sum of Absolute Differences)和 SSD(Sum of Squared Differences),分别如式(1)和式(2)所示;分像素运动估计耗时最多、最常用的操作是双线性插值;模式决策中最主要的操作也是 SAD 和 SSD。因此,SAD、SSD 和双线性插值的优化是文中重点研究的内容。

$$SAD_{M\times N} = \sum_{i=0}^{M\times N} | X_i - Y_i | \quad (1)$$

$$SSD_{M\times N} = \sum_{i=0}^{M\times N} (X_i - Y_i)^2 \quad (2)$$

式中,X 代表参考帧像素;Y 代表目标帧像素;i 代表对应的像素点位置。

在 H.265 视频标准中,$M\times N$ 可能的取值有:4×16,8×8,8×16,8×32,12×16,16×4,16×8,16×12,16×16,16×32,16×64,24×32,32×8,32×16,32×24,32×32,32×64,48×64,64×16,64×32,64×48,64×64。

2.2　IME 和 MD 算法优化

由于 IME 和 MD 算法中,大部分操作都是 SAD 和 SSD 运算,考虑到 SAD 运算的特性,因此可以使用 DSP 平台的 SIMD 加速指令[10-11]来实现优化。SAD 操作需要进行大量重复性的数学运算,指令级流水线可充分利用 DSP 提供的并行处理能力,实现处理速度的成倍提高。但是,在 H.265 视频编码处理过程中,SAD 运算都是矩阵运算,矩阵运算的特点是每行数据地址都是连续的,但是不同行的数据地址不连续。因此实现 SAD 运算需要在同一行进行运算偏移后进行判断,到了矩阵行尾的时候跳转到下一行头继续进行运算,是双循环嵌套结构。为了使 SAD 的指令级流水线能够最大程度并行运行,需要将矩阵数据进行重组,然而 H.265 变块大小技术产生多种不同大小的矩阵运算,为数据组织模式重用带来困难。

文中根据 H.265 各种块大小进行 SAD 和 SSD 操作的特点,将块大小分类,根据块矩阵行数据数量来进行分类组织,最终分成 4×N、8×N、12×N、16×N、32×N、48×N、64×N 共 7 种情况。相同的行大小带来的好处在于,每次行循环的循环次数和偏移量是固定的,这样可以将判断控制行大小内层循环去掉,变双重循环为单循环,使流水线在整个块的 SAD 和 SSD 计算过程中不会被判断打断,同时偏移量数量的减少也使被占用的寄存器数量减少。

数据重组是为了使数据运算通过 SIMD 指令并行运算,而流水线的建立是为了使 SIMD 指令充分并行执行。文中根据上述 7 种情况构建 7 套 SAD 和 SSD 计算的指令级流水线,如图 2 所示。

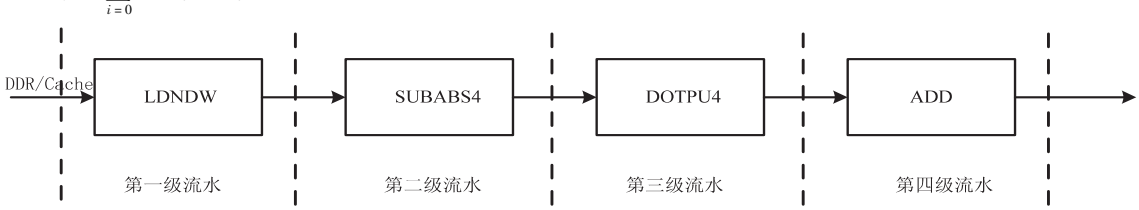

图 2　四级流水线示意图

具体操作如下:

第一级流水:完成从 DDR 或者 Cache 取数到寄存器,将通过调用 LDNDW 两次分别获取 8 个被减数和减数到不同的寄存器中;

第二级流水:将获取的前四个被减数和减数通过 SUBABS4 计算差的绝对值,该流水级运算 2 次,计算

完第一级流水获取的所有 8 个差的绝对值;

第三级流水:将第二级流水产生的 8 个差的绝对值分为两组,并调用两次 DOTPU4 获得两组绝对差和;

第四级流水:将第三级产生的绝对差和或者平方差和通过调用 ADD 进行累加获得 8 个数的绝对差和或者平方差和,然后再调用 ADD 将该和与前一组 8 个数的和相加,获得最终的累加和。

流水线分为建立、流水和释放三个工作阶段:

(1)流水线建立阶段:第一级流水需要 5 个 cycle(指令周期)可建立好,第三级流水需要 4 个 cycle 可建好,第二级流水和第四级流水沟为 1 个 cycle,因此需要 11 个 cycle 建立;

(2)正常流水阶段:正常流水 1 个 cycle 计算出 4 个绝对差和或者平方差和;

(3)流水线释放阶段:与建立阶段相同,需要 11 个 cycle 排空流水线。

以 64×64 大小块的 SAD 为例,共计需要:64×64(像素点)/4(像素点/cycle)+11+11=1046 cycle,平均 0.25 个 cycle 处理一个像素点的绝对差。

2.3 FME 算法优化

分像素估计采用双线性插值的方法实现。主要通过汇编语言优化来实现双线性插值的优化。由于双线性插值的对象是宏块,宏块的尺寸是从 8×8 到 64×64,都是 4 或 8 的倍数,因此考虑用 SIMD 指令,通过多数据并行方式来实现[11-12]。C 语言实现双线性插值,主要是通过用乘法、加法和移位操作,来实现取平均值操作。使用 DSP 的 SIMD 指令运算,可以大量节省运算时间。

如图 3 所示,*ABCD* 代表一个 32×32 大小的宏块中的 4 个整像素点,其值是已知的[13]。数字 1 至 15 代表 15 个需要进行插值的分像素点,其值需要通过已知的整像素点运算取平均值得到。可以知道,点 1 的平均值为(3*A*+*B*)/4,点 2 的平均值为(*A*+*B*)/2,点 3 的平均值为(*A*+3*B*)/4。依次类推,可以用 *ABCD* 这 4 个整像素点计算出中间全部 15 个分像素点。通过分析这 15 个点的计算方法,可以看出其实这些点可以分为两类,其中点 1、2、3、4、8、12 均只需两个整像素点(*AB* 或 *AC*)来进行运算,其他点需要四个整像素点参与运算[14]。因此可以考虑将 15 个分像素点拆成两组函数参与运算,一组是 1/2 插值,一组是 1/4 插值。1/2 插值可以参照 SAD 优化方法,根据待插分像素点的位置,将两个系数组合,将 4 组系数 pack 后放入一个寄存器中,然后采用点乘指令,一个周期内同时计算 4 个相同位置的分像素点,实现单指令多数据的优化。1/4 插值因为同时用到 4 个整像素点,即每组包含 4 个系数,1 个寄存器内只能放两组系数,如果采用点乘实现,单指令周期只能实现 2 个点的运算,因此考虑采用 DSP 平台的 Avgu4 指令实现,替代掉 C 语言中的乘法后右移的取平均值方式。Avgu4 指令可以实现 1 个指令周期取 4 对数据的平均值,大大减少了运算时间。同时由于 Avgu4 指令可以并行执行,且由于是针对块的运算,其行数都是 4 的倍数即行数能均分为 2 份,因此考虑使用两个指针使得两行数据能够并行运算,减少判断是否到行尾的判断次数,使得流水线更加紧凑、流水效率更高。

图 3 双线性插值示意图

3 实验结果

在 TI CCSV5.4 开发环境上实现,并在 TI TMSC6678 EVM 板上验证,优化后可以正常进行 H.265 视频编码,且码率不变。实验结果表明,结构优化后整个编码器速度比参考代码的运算速度提升了 7 倍,关键算法优化后整个编码器编码速度提升了 3.8 倍左右,其中优化后的 IME 算法速度平均提升 6.2 ~ 7.1 倍,FME 算法速度平均提升 10.4 倍,MD 算法速度平均提升 4.2 ~ 5.1 倍。

4 结束语

文中对 H.265 视频编码算法复杂度进行了分析,并进一步基于 TI 公司的 TMS320C6678 DSP 芯片,对 H.265 视频编码算法、结构进行优化,重点对占视频处理时间较多的运动估计和模式决策通过汇编加速指令进行优化,实现 3.8 倍左右的处理速度提升。实现结果表明,所提出的优化方法可以获得 25 倍以上的视频编码速度提升,可支持 D1 分辨率视频的实时编码处理。该方法尚可进一步扩展获得更高的速度提升,如将 *M*×*N* 矩阵数据组织成 Aligned 类型以提高 load 指令效率,更大范围内的数据共享以减少 load 指令执行次数,将指令级汇编应用到变换、量化等其他编码算法加速。

参考文献:

[1] 安维嵘. H.264 解码算法的研究及其在 DSP 平台的实现[D]. 北京:清华大学,2004.

[2] 黄铁军. 我视频编码国家标准 AVS 与国际标准 MPEG 的比较[EB/OL]. 2012. http://www.avs.org.cn/.

(下转第 179 页)

的标准读入 tablereport 中,字典将会根据此表在数据库中生成新标准的数据结构,这样新的数据结构就会应运而生,相当于完成了初始化工作,这样体现了数据架构变化程序不变的便捷之处。

3.3　元数据批量处理

将元数据应用关系数据库管理的另外一个优势便是批量处理。它跟元数据模板定制的思想如出一辙。如一百幅图件的元数据由普通的文本文件存储管理,如果查询、修改则需要打开文件上百次,相当耗费精力。一旦采用关系数据库管理,便可按照一定的检索条件呈现符合要求的元数据,其查询、更改等操作也可按照用户自行设置,这样的管理效率更高,数据的准确性也会大大提升。另外一个优势是将所有元数据集中管理有利于数据挖掘技术的使用。当存储大量数据后,由专业人员进行简单的查询就能反应其中一些比较明显的规律。如各省的地质背景情况,某区域与某种矿物存在的关系以及延伸到空间上的联系等规律,这些特质在以往的文本文件管理中是体现不出来的,因此可进一步利用这些规律来反向检查图件质量。

4　结束语

地学元数据管理系统,以一种新的管理方式即通过关系数据库来管理元数据,同时通过非规范化技术和数据字典技术来弥补数据录入输出的效率问题,这样巧妙地抵消了输入输出的时间消耗问题,反而在后期体现出它的价值,利用已有数据为后续元数据编辑,定制提供了部分数据源和参考。它不仅仅是元数据管理软件,同时也是一款元数据编辑器,为不同行业的数据进行元数据编制,方便而快捷。

参考文献:

[1] 万江波,邹逸江. 空间数据仓库元数据的认知过程[J]. 测绘科学,2008,33(1):58-61.

[2] 陈惠荣,游　雄. 地理空间元数据及其相关技术的探讨[J]. 测绘学院学报,2003,20(4):290-292.

[3] 地质信息元数据标准[S]. 北京:中国地质调查局,2006.

[4] 赵改善,曹邦功. 元数据:勘探开发数据管理的一种新工具[J]. 石油物探,2002,41(2):236-242.

[5] Abraham S,Henry F K. Database system concepts[M]. New York:McGraw Hill Higher Education,2010.

[6] Powell G. Beginning database design[M]. Birmingham:Wrox,2007.

[7] 孙　睿,刘　磊,邵明珠. 数据库的规范化理论与非规范化设计[J]. 电脑知识与技术:学术交流,2006(4):23-24.

[8] Dorsey P,Hudicka J R. Oracle8 database design using UML object modeling[M]. New York:Oracle Press,1998.

[9] 萨师煊,王　珊. 数据库系统概论[M]. 北京:高等教育出版社,2000.

[10] 马小刚,汪新庆,毋丽红,等. 应用数据字典实现多源地质空间数据的通用管理[J]. 矿业研究与开发,2007,27(1):37-40.

[11] 杨圣伟,汪新庆. 数据字典在煤炭数据发布平台中的应用[J]. 煤田地质与勘探,2008,36(6):17-19.

[12] 葛　艳,汪新庆. 用 ASP 和数据字典技术解决网络数据库中通用动态查询的问题[J]. 计算机与现代化,2004(5):75-77.

[13] 韩志军,汪新庆. 数据库系统数据字典的设计与实现[J]. 微机发展(现更名:计算机技术与发展),1999,9(2):30-32.

(上接第 174 页)

[3] 梁　凡. AVS 视频标准的技术特点[J]. 电视技术,2005(7):12-15.

[4] Sullivan G J,Ohm Jens-Rainer,Han Woo-Jin,et al. Overview of the High Efficiency Video Coding (HEVC) standard[J]. IEEE Transactions on Circuits and Systems for Video Technology,2012,22(12):1649-1668.

[5] 蔡晓霞,崔岩松,邓中亮,等. 下一代视频编码标准关键技术[J]. 电视技术,2012,36(2):80-84.

[6] Texas Instruments. TMS320C66x CorePac user's guide[EB/OL]. 2011. http://www.ti.com/lit/ug/sprugw0c/sprugw0c.pdf/.

[7] Pescador F,Chavarrias M,Garrido M J,et al. Complexity analysis of an HEVC decoder based on a digital signal processor [J]. IEEE Transactions on Consumer Electronics,2013,59(2):391-399.

[8] 郭晓珉. 基于 FPGA 的 H.264 运动估计算法优化与实现[D]. 南京:南京航空航天大学,2012.

[9] 刘　昱,胡晓爽,段继忠. 新一代视频编码技术 HEVC 算法分析及比较[J]. 电视技术,2012,36(20):45-49.

[10] Texas Instruments. TMS320C6000 optimizing compiler v7.4 User's guide[EB/OL]. 2012. http://www.ti.com/lit/ug/spru187u/spru187u.pdf/.

[11] Texas Instruments. Hand-tuning loops and control code on the TMS320C6000[EB/OL]. 2010. http://www.ti.com/lit/an/spra666/spra666.pdf/.

[12] Texas Instruments. TMS320C66x DSP CPU and instruction set reference guide[EB/OL]. 2012. http://www.ti.com/lit/ug/sprugh7/sprugh7.pdf/.

[13] Pescador F,Garrido M J,Juarez E,et al. On an implementation of HEVC video decoders with DSP technology[C]//Proc of IEEE international conference on consumer electronics. [s.l.]:IEEE,2013:121-122.

[14] 曾接贤,郑大芳,符　祥. 基于运动矢量空间相关性的 H.264 分像素运动估计[J]. 计算机工程与应用,2013,49(15):175-178.

H .265视频编码器在TMS320 C6678上的优化实现

作者： 刘贤梅， 任重， LIU Xian-mei， REN Zhong
作者单位： 东北石油大学 计算机与信息技术学院,黑龙江 大庆,163318
刊名： 计算机技术与发展 ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2015(3)