

基于 DM642 的 H.264 视频编解码器的实现

王 卿, 李 广, 陈 曦

(中国电子科技集团公司第 54 研究所, 河北 石家庄 050081)

摘 要 简要介绍了 TI 的定点多媒体处理芯片 TMS320DM642 的特点及 H.264 编解码器实现的硬件系统结构。说明了系统软件任务流程和工作原理, 并重点阐述了在基于 DM642 平台上的 H.264 视频编解码器的实现和优化。实验结果表明, 优化后可以实现 CIF 格式图像的实时处理, 并能保持较高的图像质量和压缩效率。

关键词 H.264; 代码优化; 实时编解码; DM642;

中图分类号 TN919 **文献标识码** A **文章编号** 1003-3106(2007)05-0052-03

Implementation of H.264 Realtime Video Codec Based on DM642 Platform

WANG Qing, LI Guang, CHEN Xi

(The 54th Research Institute of CETC, Shijiazhuang Hebei 050081, China)

Abstract The paper briefly introduces the features of Video/Imaging Fixed-Point Digital Signal Processor TMS320DM642 and the hardware structure of H.264 codec, and analyzes the software task flowchart and principle. Then the implementation and optimization of Video codec on TMS320DM642 is mainly discussed. Experiment results show that the realtime processing of CIF images can be realized while the high image quality and compression efficiency are preserved.

Key words H.264; code optimization; realtime codec; DM642

0 引言

H.264 标准实现编解码的压缩效率高, 而且对网络传输具有更好的支持功能, 但是 H.264 在提高编码效率的同时, 其算法复杂度也大有提高, 因而要满足图像压缩的实时性要求, 需要对现有的 H.264 编解码器进行优化^[2]。本文主要讨论 H.264 系统的硬件平台和任务流程, 并针对基于 DM642 硬件平台的特点, 介绍从代码级对算法进行优化, 进一步提高编码算法的运算速度, 实现 H.264 实时编解码。

1 硬件平台

1.1 TMS320DM642 介绍

TMS320DM642 是 TI 公司 C6000 系列的一款新型高性能 DSP, 基于 C64x 内核, 具有 64 个 32 位通用寄存器, 8 个独立计算功能单元并行运行, 可工作在 600 MHz 时钟速率, 可达到 4 800 MIPS 的峰值计算速度。DM642 采用 2 级缓存结构 L1 和 L2, 并能够提供 1 个时钟周期内 2 个 64 bit 的读取或存储, 在图像处理算法方面的能力有了很大的提高, 相对于 C62xDSP 最多有 19 倍的性能改善^[1]。

DM642 芯片还集成了各种片内外设。它有 3 个

可配置的双通道视频端口, 最多可处理 6 路视频采集数据; 具有 64 个独立通道的 EDMA (扩展的直接存储器访问) 控制器, 负责片内 L2 与其他外设之间的数据传输。DM642 还有 64 bit 的外部存储接口 (EMIF)、10/100 Mbps 以太网 MAC、多通道音频串行端口 (McASP)、主机接口 (HPI) 和多通道缓存串口 (McBSP) 等外设。可见, DM642 是目前构建数字多媒体处理应用的一个理想平台。本文选用 DM642 作为开发平台。

1.2 系统结构框架

H.264 编解码器的系统结构框图如图 1 所示。视频模拟信号输入经 SAA7115 ADC, 数字信号进入 DM642 进行处理, DM642 扩展 2 片片外 SDRAM 和 1 片 FLASH 存储数据和程序, 并且通过 I2C 总线对模块进行配置来协调和控制整个系统。输出的数字视频信号经过视频端口的内部 FIFO 缓冲后, 由 DM642 通过 EDMA 将数据送到片外 SDRAM 中, 供视频处理程序使用。编码算法处理后的码流可以经过 HPI 口传输, 也可以以实时重建回放的方式输出给 SAA7129 DAC, 用于监视器上显示, 还可以将压缩码

收稿日期: 2006-12-17

流经串口输出用于测试。

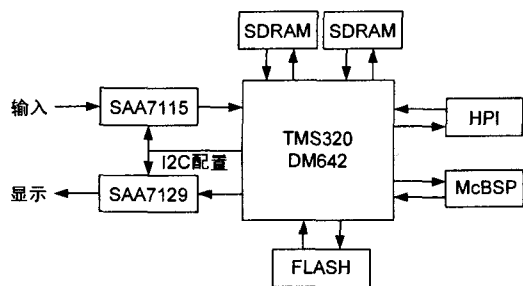


图1 H.264 编解码器系统结构

2 软件任务流程

TI 公司的 DSP 开发软件包括集成开发环境 (CCS)、实时操作系统 (DSP/BIOS) 和第三方算法库标准 (eXpressDSP)。DSP/BIOS 是一个用户可裁减的实时操作系统,包含内存管理、线程管理、中断和任务调度等功能。在本系统中,采用基于 DSP/BIOS 的 TI 参考框架 RF5 (Reference Framework 5)^[4] 来协助实现系统流程中各个环节的交互和协调同步。

如图 2 所示,将整个核心流程分为采集、编码、解码和显示 4 个部分,其中编码线程 thrEncode 负责视频压缩编码,解码线程 thrDecode 负责相应的解码重建过程,是系统的 2 个核心线程。采集线程 thrcapture 主要负责对输入数据的获取,完成捕获数字视频信号的工作,然后将 4:2:2 格式重采样为 4:2:0 格式,等待交由 thrEncode 线程处理。显示线程 thrDisplay 则等待获得 thrDecode 线程处理完毕后的数据对其进行反向的重采样得到 4:2:2 格式的数据,然后通过设备驱动向 VPORT 口写出数据以供输出端的 DAC 进行处理。

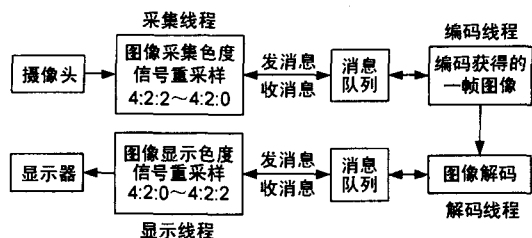


图2 核心任务流程

3 编解码算法优化

编解码算法是 H.264 系统的核心,其实现流程

一般分 3 个阶段:第 1 阶段产生和评估 C 代码;第 2 阶段优化和评估 C 代码;第 3 阶段编写和评估线性汇编^[1,3]。这 3 个阶段不是必须经过,当在某一阶段获得了期望的性能,就不必进行下一阶段的优化。

3.1 产生和评估 C 代码

要进行代码优化,首先必须找出程序中的瓶颈所在,即占用了大部分 CPU 时间的代码。在 CCS 环境下,使用代码剖析工具 Profiler 或者 clock 系统时钟及 printf 函数统计出某段函数耗时,对关键函数进行优化。首先关掉 P 帧,在 DSP 上优化 I 帧的编解码,然后加入 P 帧编解码,这样可以减少优化代码的工作量,并且为优化 P 帧增加经验。表 1 为 I 帧编解码各部分耗时。

表 1 各编码部分耗时所占比例

编码部分	编码耗时所占比例
模式选择	9.1%
编码	6.8%
去块滤波	40.5%
子像素插值	41.2%
其他	2.4%

由以上的数据得到主要的几个耗时比较大的部分,其中模式选择、编码、边缘滤波以及半像素的插值耗时较多,分别针对这几个部分选定优化的方向。

3.2 优化 C 代码

要针对图像处理的特点对 DSP 进行优化编程,充分发挥 DSP 的运算能力,必须从它的硬件结构出发,最大限度地利用 8 个功能单元,使用软件流水线尽量让程序无冲突地并行执行。根据代码分析结果,主要采用以下方法优化 C 代码:

① 程序中一般的循环体都满足并行处理的条件,并且循环体往往是程序中耗时最长的,而且循环语句不能充分利用 C6000 的软件流水线,因此进行优化时循环体是重点。将多重循环展开成单层循环,调整语句,提高程序并行执行的能力^[2];

② DSP 的指令多为单周期指令,但是转移类指令却通常要耗费较多的时钟周期,每个跳转都有 5 个延迟间隙,另外循环内跳转也使软件流水受到

专题技术与工程应用

阻塞,因此应尽可能地减少程序中的分支。使用逻辑判断语句代替 if ...else ... 语句可以去除不必要的跳转;

③ 充分利用 C6000 系列 DSP 的超长指令字特点,尽可能用多位的指令来访问少位的数据。如:使用 int 型(32 位)访问 2 个 short(16 位)型数据,将其分别放在 32 位寄存器的高 16 位和低 16 位字段。这样数据读取效率可以提高 1 倍,从而减少内存访问次数^[2];

④ 对于复杂运算语句,用查表法实现,以节省耗时。例如,在量化和反量化程序中移位位数的处理,先计算出所有可能的值,后来的运算就可以只需查表得到数值;

⑤ C6000 编译器提供的 intrinsic 可快速优化 C 代码。内联函数定义在 CCS 所在的 C6000 \ CGTOOLS \ Include 目录下的 C6x.h 文件中。在 C 语言的调试全部通过以后,将尽可能多的语句使用内联函数改编,尤其对循环体和循环展开配合使用,可以大幅度减少执行时间^[2];

⑥ DSP 对片内寄存器和 RAM 的访问速度比片外快很多,因此合理地配置和使用存储空间,对系统整体效率影响很大,尽可能地将访问频繁的代码段和常数表放入片内 RAM。访问片外数据时,将数据从片外数据存储器读到片内^[3],程序执行时直接从片内调用相应的数据,大大提高程序的运行速度;

⑦ QDMA 是 C6000 提供的一种快速 DMA 传输方式,在视频实时性和数据速率要求高的场合,数据搬移部分适合采用 QDMA。对于正在编码的帧,通过 QDMA 功能将当前帧中当前宏块搬移到片内。同时在片内开辟 2 个缓冲区,采用 ping-pang 模式存放和处理当前宏块,实现搬移和处理数据的同时进行,以提高编解码速度。

4 实验结果

在给出的硬件平台上对优化前和优化后的性能进行了测试。几个主要函数优化结果如表 2,可见,用循环展开、内联函数等技术改写 C 代码,能够大大提高处理速度。

表 2 几个主要函数优化前后比较

函数	所用时钟数	
	优化前	优化后
DCT4x4	272	38
IDCT4x4	322	48
量化 4x4_c	238	56
帧内预测模式 0	272	29
SAD 计算 4x4	191	26

使用一些典型的标准测试序列,如 football.cif、news.cif 及 mobile.cif 对编解码速度进行了测试。分别取测试序列 100 帧,码率 384 kbps,测试编解码时间和 psnr 值结果如表 3 所示。

表 3 编解码时间和 psnr 测试结果

测试序列	编解码时间/ms		平均 PSNR/dB		
	编码	解码	Y	U	V
Forman.qcif	561.5	291.9	33.4	38.5	39.3
Foot.cif	2 621.2	935.7	33.6	38.0	39.2
Mobile.cif	2 208.7	1 046.0	33.1	37.8	38.8

试验结果表明,码率为 384 kbps 时,编解码速度达到 30 fps 左右,平均峰值信噪比(PSNR)33 dB 以上,实现了 CIF 分辨率图像的实时编解码。

5 结束语

介绍了 H.264 视频编解码器实现的硬件平台和软件实现任务流程,讨论了在基于 DM642 的硬件平台上对核心编解码算法实现和优化的方法,针对 DM642 芯片的特点和 H.264 视频编解码程序自身的特点,对程序进行了优化。测试结果表明,优化效果明显,经过优化后的编解码器能够满足 CIF 格式图像的实时性处理要求。 ❖

参考文献

- [1] 李方慧. TMS320C6000 系列 DSPs 原理与应用[M]. 北京: 电子工业出版社, 2003.
- [2] WIEGAND T, SULLIVAN G J. Overview of the H.264 / AVC Video Coding Standard [J]. IEEE Trans. on Circuits and Systems. 2003, 13(7): 1 - 18.
- [3] 张彤宇, 苏睿. H.264/AVC 视频编码器在 DM642 平台上的实现与优化[J]. 微电子学与计算机, 2005, 22(12): 165 - 168.
- [4] 司群, 臧英新. TMS320DSP 算法标准(XDAIS)及参考构架 RF5 综述[J]. 舰船电子工程, 2006, 2(2): 27 - 31.
- [5] 王嵩, 周祥平. 新标准 H.264 的核心技术与视频移动通信研究[J]. 无线电工程, 2005, 35(5): 21 - 27.

作者简介

王 卿 女, (1982 -), 中国电子科技集团第 54 研究所在读硕士研究生。主要研究方向: 通信与信息系统、视频编解码。