



基于 TMS320DM642 的 H. 264 解码器的实现与优化

刘羽楠 国澄明

(天津大学电子信息工程学院 天津 300072)

摘要: 本文描述了基于 TMS320DM642 DSP 多媒体平台的 H. 264 解码器的实现和优化。简单介绍了该解码器的硬件平台之后,讲述了移植解码器的过程,包括将基于 PC 的 H. 264 解码程序 DSP 化;随后介绍了针对各解码模块的优化方法。实验结果表明:与原始的 H. 264 解码器(JM74)相比,该解码器提高了运行速度,可以进行实时应用。

关键词: H. 264/AVC; 解码器; 移植; 算法优化

中图分类号: TN919 **文献标识码:** A

Implementation and optimization of H. 264 decoder software based on TMS320DM642

Liu Yunan Guo Chengming

(School of Electronic Information Engineering, Tianjin University, Tianjin 300072)

Abstract: This paper describes the implementation and optimization of H. 264 decoder based on TMS320DM642 DSP platform. After briefly introducing the hardware system, the procedure of porting the decoder is described that includes the digital signal processing on decoder program based on H. 264 of PC. Then the optimization methods for every decoding module are explained. The experiment results indicate that comparing with JM74, the running speed of decoder is increased and the performance of the decoder in real-time applications is good.

Keywords: H. 264/AVC; decoder; porting; algorithm optimization

0 引言

H. 264^[1-2]是由 ITU-T 视频编码专家组(VCEG)和 ISO/IEC 移动图像专家组(MPEG)共同提出的国际视频编码标准。它在 H. 264、H. 263 视频压缩标准的基础上,进行了改进和扩展,其目的是在不降低图像质量的情况下进一步降低编码率,提高压缩效率。此外,该标准提供一个友好的网络接口,使得视频码流更适合在网络上传送。由于该标准可以提供更低的码率,所以更适合应用于多媒体通信领域。但与此同时,H. 264 性能上的改进也带来了复杂度的明显提升。人们普遍认为,H. 264 解码器比起其他标准需要更多的运算量和存储空间,因此在嵌入式环境中实现高效的 H. 264 解码器是一个极具挑战性的工作。

1 TMS320DM642 多媒体平台^[3-5]

本文所讨论的 H. 264 解码器的移植和优化工作,是在我们已研制成功的 DM642 多媒体处理平台上进行的。该系统以 DM642 为核心处理器,具有音、视频捕获显示通道以及以太网通信模块。除了 DM642 内部可配置的 256 KB L2CACHE/SRAM 外,系统中还扩展了 16 MB 的片外 SDRAM 存储器,可以满足本 H. 264 解码器对存储空间的

需要。

DM642 多媒体处理硬件平台如图 1 所示。本系统的工作流程为:(1)通过网络接收 H. 264 码流;(2)H. 264 解码;(3)将解码后视频图像输出给显示终端。

2 H. 264 解码器程序移植与优化

本文重点介绍 H. 264 解码程序在上述硬件平台上的实现。本文的做法是从联合视频专家组(JVT)提供的参考软件 JM7. 4 出发,主要进行两方面的工作:(1)将该程序移植到该硬件平台上;(2)为了满足实时性的要求,必须对移植后的程序进行相应的优化。

2.1 移植^[6-7]

JM7. 4 是用 C 语言编写的基于 PC 的参考代码。TI 公司 DSP 有其自己的 ANSIC 编译器和优化器,并有自己的语法规则和定义。因此,在 DSP 上实现 H. 264 的算法要把 PC 上 C 语言编写的 H. 264 代码进行改动,使其完全符合 DSP 中 C 的规则。这些改动包括:去除所有的文件操作;去除多余的命令行参数;合理安排内存空间的预留和分配;规范数据类型——即修改一些 CCS 编译器所不支持的数据类型,如 longlong 可根据情况改为 double,而且要注意,某些数据类型长度在 VC 编译器中的定义和在

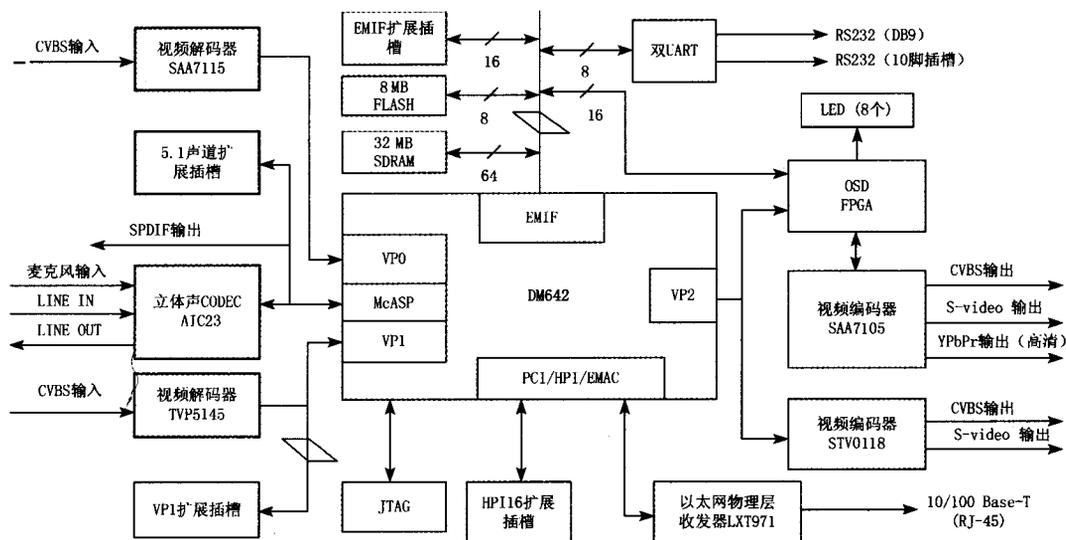


图 1 DM642 多媒体处理平台框图

CCS 编译器中的有所不同,如 int 类型的长度在 VC 编译器中为 16 b,而 CCS 中则为 32 b,所以要进行相应修改以避免此类问题带来的负面影响。

以上工作仅是将基于 PC 的 H.264 解码程序 DSP 化。考虑到实际硬件平台的某些特点还进行了下面的修改以提高程序的效率。

(1)在程序数据输入接口的设计中引入乒乓机制。考虑到 DM642 只有有限的片内存储空间,压缩数据、参考帧图像均要存储在片外存储器中。此外,DM642 具有 64 个独立的 EDMA 通道,所以在程序输入接口的设计中引入乒乓机制,从而提高 CPU 的工作效率并降低数据存取的时间开销。

乒乓机制实际是一种使用 EDMA 的方案,所谓“乒乓”是指在 DM642 片内存储器中开辟并分别称为 ping 和 pong 的两个缓冲区。当 CPU 对缓冲区 ping 中的数据进行处理时,EDMA 由片外存储器向缓冲区 pong 搬数据;而当 EDMA 向缓冲区 pong 搬完数据且 ping 中的数据处理完毕后,CPU 就转而处理缓冲区 pong 中的数据;与此同时,EDMA 再向缓冲区 ping 搬数据。如此反复运行就可完成 CPU 对一大块数据的处理。这样做的好处是 EDMA 与 CPU 并行工作,节省了大量花费在数据搬运上的时间。

(2)为节省 DSP 有限的片内存储空间引入“两行机制”。

根据 H.264 标准,要正确解码某块不仅需要当前块的解码信息,还往往需要当前块左面和上面的邻近块的一些参数或像素,这些参数或像素包括:运动矢量(mv);参考帧序号(ref_idx);非零系数的个数;帧内编码时要用到的邻近像素。在 JM7.4 中需要把当前帧中全部的上述参数和像素都保存起来,这将占用相当大的内存空间,如一帧

CIF 图像的运动矢量约占用 50 KB。为此,本文设计了“两行机制”,即只保存两个宏块行的参数或相关像素。这两个宏块行就是当前宏块所在和当前宏块上面的宏块行,使用该机制可以大大减少内存占用。

2.2 优化^[8]

2.2.1 汇编实现反变换模块

本文把 JM7.4 解码器中的 itrans 函数用汇编语言进行了重写。函数 itrans 的输入是反量化后的残差块和预测块,输出是解码后的 YUV 信号。H.264 在变换与反变换中都使用 16 b 运算,因此可以充分使用 C64x 提供的强大的包处理指令。使用这些指令,可同时处理 2 个 16 位数据和 4 个 8 位数据。C64x 还提供了 STDW (STNDW)、LDDW (LDNDW)等指令,可一次存取连续的 64 位数据,两组 D 功能单元(.D1 和 .D2)一次最多可存取 128 位数(即 8 个 short 型数据)。

在使用汇编语言重写了反变换模块后,该模块的运行效率有了明显的提升,如表 1 所示。

表 1 用汇编语言重写 itrans 函数前后的效率对比

	JM74 中的 函数 itrans	汇编实现的 函数 itrans
执行一次的平均 CPU 周期使用数	323	32

2.2.2 新的 Cavlc 解码算法

原始 JM7.4 程序采取了遍历码表的方式来进行 CAVLC 解码,效率很低。通过研究 CAVLC 码表,本文利用了 CAVLC 码表中的一些规律从而大大提高了这方面的解码效率。首先,根据码字开始处零的个数把一个很大的 H.264 码表分成若干小码表,从而缩小搜索范围;然后,利用小码表中的规律快速在码表中定位。表 2 所示的是以

解码 QCIF 的 foreman 序列为例,对比新旧两种算法的效率。

表 2 新旧 CAVLC 解码算法的效率对比

	JM7.4 CAVLC 解码算法		新的 CAVLC 解码	
	I 帧	P 帧	I 帧	P 帧
非零系数与 Trailing 1s 的个数	14 144	14 106	176	166
TotalZero	14 467	14 409	139	155
Zerorun	14 118	14 127	102	104

注:表中索引数字为平均周期数

2.2.3 图像边界扩展

JM7.4 在进行运动补偿时,使用运动向量在参考图像中寻找预测块。当运动向量指向参考图像以外时,预测块超出参考图像边界的那些样点通过图像外扩技术来获得。在 JM7.4 中通过对数组下标进行判断,确定参考样点是否超过图像边界,增加了处理时间。为此,考虑将参考图像的边界向四周各复制若干各像素。

H.264 标准中宣称拥有“无限制”的运动向量,但是通过对 JM7.4 编码器的阅读发现它的运动向量并不是无限地指向图像之外。编码器首先通过相邻块的运动向量来预测当前块的运动向量,并以此作为当前运动向量的搜索中心,然后依次在整数(搜索半径最大为 16)、1/2 像素、1/4 像素位置进行搜索,最终确定运动向量的值。当确定预测搜索中心后,编码器先判断其是否跨越图像边界,若跨越边界则将其值限定在图像边界处,然后至多向外搜索 16 个像素。这是因为当运动向量使计算一个预测块所需要像素完全处于参考图像之中,伸多远,所得到的预测块都是相同的。因此,将扩展宽度设为 16 个像素就可以了,如图 2 所示。扩展后原图像的坐标原点发生了移动,需要对原程序中的地址指针进行修订。

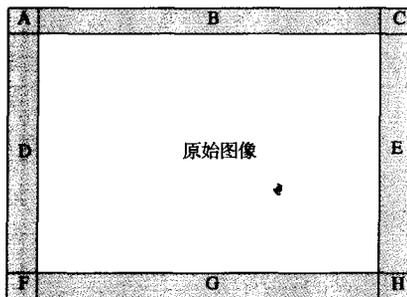


图 2 图像边界扩展

的是分数位置样点。aa、bb、cc、dd、ee、ff、gg、hh 是计算这些样点所需要的中间值。应用权(抽头)值为(1, -5, 20, 20, -5, 1)的六抽头滤波器得出 1/2 像素位置的亮度预测样点。通过对整像素位置和 1/2 像素位置的亮度样点取平均值或者将 1/2 像素样点进行二次插值 2 得出 1/4 像素位置的亮度预测样点。

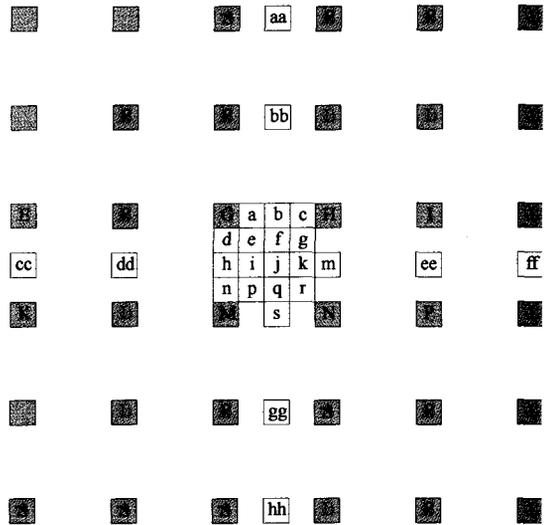


图 3 分像素与整数像素位置关系图

使用汇编流水线技术改写插值函数,优化前后效率如表 3 所示。表 3 提供了优化前后,获取 3 个有代表性的位置(G、b、j)预测样点所需运算量的比较数据。G 点是整数相位样点,这个位置不需要进行插值计算就可以得到参考样点。b 点为 1/2 像素相位样点,需要进行一维插值运算。j 点为 1/4 像素相位样点,需要进行二维插值运算得到参考样点。

表 3 插值函数优化前后的效率比较

样点位置	o1 优化,包含边界判断 (cycle)	经过图像边界处理并使用汇编优化后 (cycle)
G	2 600	167
b	9 875	393
j	44 437	1 829

2.2.5 运动补偿与数据预取

一个算法的复杂度,除了包括最基本的处理开销,还要包括数据载入的开销。因此对运动补偿的数据输入方式做了修改,以降低数据载入的时间开销。

JM7.4 中每个 4×4 块进行运动补偿时为了获得相应的参考样点需要读取一个 9×9 大小的块进行插值运算,为了提高数据的利用率本解码器使用了新的运动补偿方案。

在解当前宏块时,先从参考帧中(存储在片外)读入一个数据窗(存储在片内);在进行运动补偿时,将原来从参

2.2.4 汇编语言优化插值模块

图 3 为分数像素与整数像素位置关系图,图中用大写字母表面的阴影块位置表示的是二维亮度矩阵中的整数像素亮度样点,这些样点用来生成亮度预测样点。a~s 代表

考帧中读入每个子块插值所需数据,变为在预取窗中读取。若所需的 9×9 块完全包含在预取窗中,则从该窗中读取;若不在窗中,则作为特例情况仍使用原来的方式从参考帧中读取。

显然一个需要解决的问题就是如何选取窗口的大小。经过实验,最终方案采用的是 48×48 的窗口。这相当于一个 3×3 宏块大小的窗,如图 3 所示。在这种情况下,特例情况(插值所需 9×9 块不完全在窗中)所占比重大约为 0.3%。当处理每一行的第一个宏块时程序读入 9 个宏块,以后依次处理本行其他宏块时只需要再读入 3 个宏块就可以了,因为有 6 个宏块(见图 4:2、3、6、8、9 和当前块)已经位于窗中,只要将它们向左平移就可以了。这样解一帧 QCIF 图像实际的运动补偿数据读取量(从片外到片内)为: $99 \times 3 \times 16 \times 16 + 9 \times 16 \times 16 \times 9 = 77\ 328$ 字节。而 JM7.4 原方案所需数据量为 $99 \times (9 \times 9) \times 16 = 128\ 304$ 字节。可见新方案数据载入量为原来的 60.3%。同时,新方案可以在宏块解码初始化时就启动 EDMA 将这些数据预取到片内,从而大大节省了载入数据的时间。色度与亮度使用的方法一致,所不同的时窗口尺寸大小为 24×24 像素。



图 4 运动补偿预取窗

3 结束语

目前,经过上述的优化措施,表 4 所表示的是本解码器在解码两个测试码流时的运行速度。

表 4 解码器

序列	帧	所需周期数/周期
Foreman(QCIF)	I	7 321 722
	P	4 614 783
Tempete(CIF)	I	30 351 869
	P	21 029 748

由于 TMS320DM642 的时钟频率是 600 MHz,从表 4 可以计算出:本系统的 H. 264 解码器每秒种可以解码 100~120 帧 QCIF 图像,或 20~30 帧 CIF 图像,这已经达到了实时解码的要求。作为在嵌入式环境中实现新一代视频压缩标准解码器的有益尝试,可以应用于网络多媒体终端、数字电视机顶盒、视频监控、可视电话、视频会议系统等多项领域。

参考文献

- [1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H. 264 \ ISO/IEC 14496-10AVC) Joint Video Team (JVT)[Z]. Mar. 2003,Doc. JVT-G050.
- [2] 钟阳,李文元. 高效视频编码标准—H. 264[J]. 电子测量技术,2004(5):22-23.
- [3] Texas Instruments. TMS320DM642 Technical Overview[Z]. Application Report SPRU615,Sept. 2002.
- [4] Texas Instruments. TMS320DM642 Video/Imaging Fixed-Point Digital Signal Processor Data Manual[Z]. SPRS200A, Apr. 2003.
- [5] 刘强,国澄明. 数字媒体处理器 TMS320DM642 及其应用[J]. 电子测量技术,2003(6):55-56.
- [6] Texas Instrument. TMS320C6000 Programmer's Guide [Z]. 2000.
- [7] Texas Instrument. TMS320C6000 Code Optimization Partitioning Strategies to Improve Inner Loop Performance[Z]. 2000.
- [8] Texas Instrument. TMS320C6000 CPU and Instruction Set Reference Guide[Z]. 2000.

作者简介



刘羽楠,女,天津大学电信学院信号与信息处理专业,硕士研究生。
电话:022-27408404

作者简介



张小兵,男,1978 年出生,燕山大学信息学院硕士研究生,主要研究方向为智能天线自适应算法。
电话:0335-8515840; E-mail: bj2008_xb@yahoo.com.cn

(上接第 54 页)

- [9] HAYKIN T. Adaptive filter theory[M]. 3rd Edition. Prentice Hall,2002.
- [10] ABOULNASR T, MAYAS K. A robust variable step-size LMS type algorithm: analysis and simulation[J]. IEEE Trans. Signal Processing,1997,45(3):631-639.