

基于 AVS-M 和 DM642 视频服务器的研究

The research of Video Sever based on TMS320DM642 and AVS-M

(1.哈尔滨工业大学(威海);2.吉林电子信息职业技术学院)周广禄¹ 王翠华¹ 孙鹏娇² 郭恒业¹
Zhou,Guanglu Wang,Cuihua Sun,Pengjiao Guo,Hengye

摘要:本文完成了视频服务器的硬件设计,针对如何充分发挥 DM642 硬件平台的处理能力,提出了关于 AVS-M 算法的编码优化方案,该方案是对软件框架流程进行仔细考虑后提出的,避免了冗余操作,针对存储系统对各部分的数据结构进行了设计,而且通过 DMA 实现了计算与数据传输的并行处理。

关键字:AVS-M; DM642; 视频服务器; 实时编码

中图分类号:TP39 **文献标识码:**B

Abstract: This paper realized the hardware design of video server. After considering the software framework process carefully, we advance the optimization of AVS-M coding, which is based on how to bring into full play handling ability of the hardware platform of DM642, this scheme avoid redundant operation, design the data structure of the storage system in each part, moreover, realize the parallel processing of calculation and data transmission by DMA.

Key words: AVS-M; DM642; video server; real time encoding

1 引言

AVS是具有自主知识产权的数字音视频编解码技术标准,其包括系统、视频、音频、数字版权管理等四个主要技术标准和一致性测试等支撑标准。其中,移动视频标准 AVS-M (AVS 第七部分)适用范围包括视频会议、可视电话、移动多媒体等领域。

TMS320DM642 是 TI 公司开发研制的一款专门面向多媒体应用的专用数字信号处理芯片,使用此 DSP 芯片并利用 AVS-M 算法来进行视频压缩,可大幅提高视频压缩率,减少传输流量,即使在低带宽情况下也可以有效保证实时性和监控需求。

我们开发的 Imlab6421 视频服务器,是基于 Internet 的视/音频监控设备。核心 DSP 芯片采用 DM642 芯片。而应用软件、视频压缩算法是根据 AVS-M 标准进行设计、优化的。下面针对这款视频服务器的系统结构、视频优化的软件设计等进行详细的介绍。音频压缩的优化本文不做介绍。

2 硬件设计

视频服务器 Imlab6421 原理框图如图 1 所示,系统采用 TMS320DM642 用于音视频压缩。DM642 芯片包含一个 64 位的外部存储器接口,可驱动 4 个片选地址空间(CE0, CE1, CE2, CE3),它支持 8, 16, 32, 64 位宽度的同步和异步访问。我们在 DM642 的片外扩展

了 16M Bytes 的 SDRAM, 位于 EMIF 的 CE0 地址空间,用于存放程序和数据。SDRAM 工作时钟为 100MHz,是由 DM642 芯片的 CPU 工作时钟 6 分频产生的。数据更新由 DM642 自动完成。还在 DM642 板上设计有 512K Bytes 的 Flash 存储器,位于 DM642 的 CE1 地址空间,宽度为 8 bits。另外能实现与 Internet 连接的以太网处理器采用的是 CRYSTAL 公司的 CS8900A,它高度集成设计使其不再需要其它以太网控制器所必需的昂贵外部器件。

视频编码工作原理大致为:输入的模拟视频信号经 TVP5150(支持 PAL 和 NTSC 两种制式)被数字化为 YUV4:2:2 的数字视频格式,经由 I2C 总线被送至输入缓冲区(采用三缓冲机制),DM642 的 CPU 把捕捉到的视频数据从一个输入缓冲区中取出待编码图像数据进行压缩编码处理,形成的压缩码流放到输出缓冲区,然后打包通过网口直接传输到 Internet。

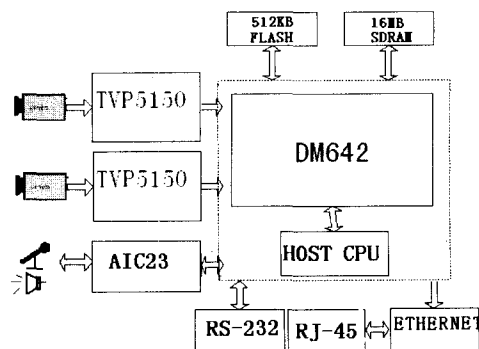


图 1 Imlab6421 硬件原理框图

周广禄:讲师 工学硕士

国家自然科学基金重点项目(项目编号:60333020)

3 软件设计

DSP嵌入式程序受硬件资源的限制,对程序流程和数据组织需要从硬件资源和代码运行效率上做仔细的考虑。通过分析 AVS-M 编码器的程序流程,借助实验中积累的经验,本文给出了 AVS-M 编码器的优化方案,主要介绍 Cache 性能优化、存储空间的分配以及 CPU 与 DMA 的并行性设计等。

3.1 存储结构及 CACHE 性能优化

(1) 存储结构:DM642 的存储器系统由片内内存 L1、L2 和片外外存两部分组成,L1、L2 和片外 SDRAM 构成了整个存储器系统的三级层次结构,如图 2 所示。其中,片内内存采用两级缓存结构,第一级由 L1P 和 L1D 组成,L1 距离 DSP 核最近,数据访问速度最快,只需一个时钟周期,只能作为不能寻址的 Cache 使用。第二级 L2 是一个统一的程序/数据空间,可以整体作为 SRAM 映射到存储空间,也可以整体作为第二级 Cache,或是二者按比例进行组合。第三级是片外外存,一般由 SDRAM 构成。L1P cache 大小为 16KB,直接映射,每行大小 32 字节;L1D cache 大小 16KB,2 路映射,每行大小 64 字节。L2 是 L1 和外存储器的中间层,容量较大有 256KB,访问速度较慢,根据 L2 配置为 Cache 或 SRAM 的不同选择,访问速度需 8 个或 6 个时钟周期。片外存储器容量很大但访问速度很慢,一般都会远远大于 8 个时钟周期。

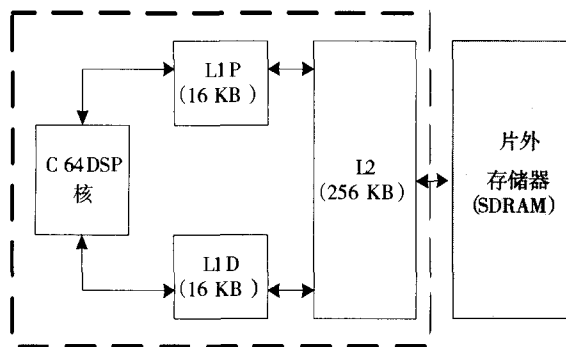


图 2 三级存储系统

(2)CACHE 性能优化:要优化 Cache 的使用性能需了解 Cache 的具体结构,如 Cache 容量、行大小、组相联数等。下面总结了一些优化 Cache 性能的方法:合理配置 L2;合理布置程序代码段和数据段的内存布局,为防止有效代码、数据在缓冲存储器中相互排挤,应尽量把顺序执行的代码、同时使用的数据放在相互邻接的物理空间当中;若函数模块和数据包含在一个循环中,循环体的大小应和 Cache 的容量相吻合,以便能把整个循环体全部放入 Cache 中。为了提高 Cache 中数据的重复利用率,把数据操作构成一条数据处理链,链中的下一级操作就能直接使用上一级操作留在 Cache 中的数据。此外还可以根据 Cache 行数据宽度信息调节数据在物理内存中的存放位置,从而利用数

据预取增加 Cache 的命中率;挖掘 L1D 的不命中流水处理能力,加速待使用数据的读入速度;通过合理的数据填充策略,避免同一时钟周期对相同存储体的读写操作将造成存储器的存取冲突。

3.2 存储空间的分配

在 DSP 上由于内存空间有限,需要合理分配内存空间,这对于程序的运行效率十分重要。使用的一个原则是:应尽量把数据和代码放入片内存储器。因为外存比 CPU 工作的速度要慢很多,如果用 CPU 来处理访问外部存储器的工作,大量时间将浪费在存取等待上。

DM642 的 L2 片内存储器可以配置为 SRAM 或 Cache。由于编码器的数据流程是有规律的,因此我们考虑用程序控制 DMA 控制器来进行内存和外存之间的数据交换,这样比硬件自动地来处理效率要高。

由于片内存储器容量的限制,不可能将编码器的所有数据都放入片内存储器。原始图像和重构图像是无法完全放到片内存储器中的。事实上,没有必要将这些数据放在片内,因为编码器的处理过程是以宏块为单位的,我们只需要在片内维护一个宏块的数据结构,CPU 访问这些数据进行计算。每编码一个宏块的时候把该宏块需要的数据从外存调入内存,填到相应的这些数据结构中。利用 DM642 提供的 QDMA 机制,CPU 发出 QDMA 请求后就可以继续对其它数据进行计算,由 DMA 负责将数据从外存调到内部存储器。因此如何设计使 CPU 与 DMA 之间协调工作很重要,本文 2.3 部分将详细讨论这个问题。

需要注意的问题是当前宏块编码过程中需要用到前面编码已经获得的一些信息。参考代码中是保留所有宏块的编码信息,这样的做法是不适合 DSP 实现的,需要的存储空间太大,片内存储器无法容纳。实际上编码当前宏块只需要参考它上面和左面的宏块。因此设计编码器中各模块的局部数据结构如图 3 所示。该数据结构保留上面一行的值和左边宏块的值,每编码完一个宏块,确定当前宏块的信息后更新这些缓冲区,这些数据可以放在 L2 中,不用访问外存。而且实验证明用来维护这样的数据结构所需要的计算时间很小。

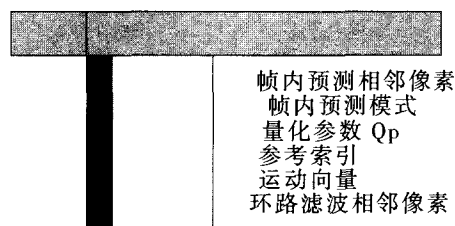


图 3 模块的局部数据结构

经过优化的程序和常用的数据结构的大小可以放在 L2 中。所以按照上面的分析将 L2 配置为 256KB SRAM, 将程序代码段(.text)、变量初值表(.cint)、常量字符串(.const)、全局变量静态变量(.bss/.far)、堆栈段(.stack)等放入 L2 SRAM 当中, 全局堆(.system 用于动态存储器分配)置于外部存储器。表 1 总结了编码器所要用到的存储空间分配情况。

表 1 编码器存储空间的分配

数据名称	大小(字节)	所属段名	存放位置
程序	130K	.text	L2 SRAM
整像素运动估计参考缓冲区	9.5K	.far	L2 SRAM
分像素运动估计参考缓冲区	10K	.far	L2 SRAM
编码常用数据结构	40K	.far	L2 SRAM
编解码表	9K	.const	L2 SRAM
函数调用栈	8K	.stack	L2 SRAM
当前编码帧	图像大小	.system	片外 SDRAM
当前重构帧	图像大小	.system	片外 SDRAM
整像素参考帧	图像大小 (两参考帧)	.system	片外 SDRAM
分像素参考帧	图像大小的四倍 (两参考帧)	.system	片外 SDRAM

其中整像素运动估计参考缓冲区包括亮度和色度。因为参考帧有两个, 整像素运动估计参考缓冲区也有两个。分像素运动估计参考缓冲区也是两个: 一个用来调入 SKIP 编码模式的预测值, 一个用来做分像素运动估计。

3.3 CPU 与 DMA 并行性设计

I 帧编码可以说是 P 帧编码的特例, 如果 P 帧中不用运动估计的话, 则与 I 帧编码流程相同。因此下面对于 CPU 与 DMA 的并行性的讨论只针对 P 帧。

我们要解决的问题是 CPU 什么时候发 QDMA 请求, 命令 DMA 控制器将需要的数据调入内存中。而且这种调度方式要保证 CPU 发命令之后可以进行其它的计算, 等 CPU 需要这些数据的时候, DMA 已经将其调入内存中了。

为了解决这个问题需要了解编码器各个模块的运行时间, 以及 DMA 调度数据到内存所需要的时间。通过在 DM642 上运行优化过的程序, 一个参考帧情况下测得各部分占程序运行时间的比例大致如表 2 所示:

表 2 程序各部分运行时间所占比例

模块名称	所占比例
1/2 插值	10%
初始化	2%
整像素运动估计	35%
分像素运动估计	15%
帧内亮度预测	15%
重构和扫描	10%
环滤波	6%
熵编码	4%
其它	3%

图 4 中由 CPU 指向 DMA 的箭头表示启动 QDMA 传输。每个 DMA 传输所用的时间相对于程序运行的时间比例是: 传输原始像素占 1%, 传输 SKIP 和分像素运动估计参考区各占 3%, 传输整像素亮度和色度参考区共 15%, 传输环路滤波结果 5%。整个 DMA 传输的时间大概占 CPU 计算时间的 30%。通过这些数据可以看出, 按照图 4 进行安排可以达到上面所述目标。

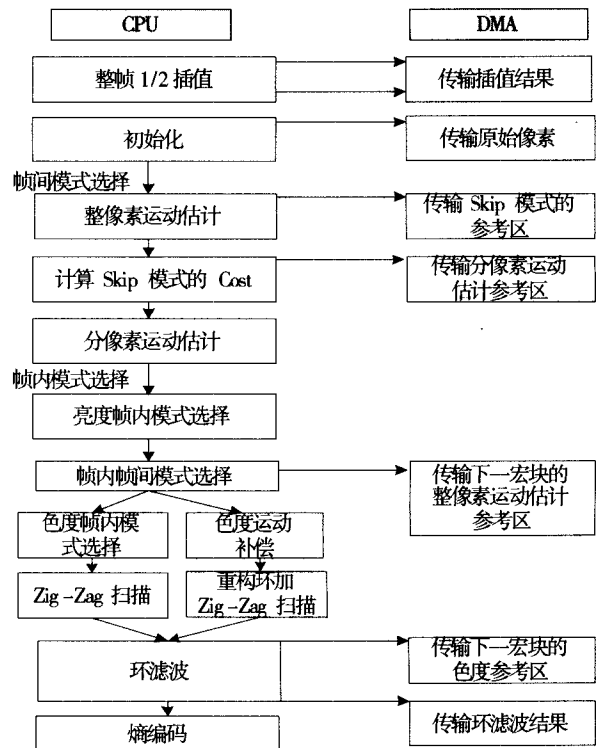


图 4 CPU 与 DMA 并行工作

只依靠上面这些方法进行优化, 视频压缩还不能达到实时要求, 还需要进行算法级优化, 以及对编码器中各个模块进行程序代码级的优化。常通过采用内联函数、软件流水、线性汇编优化等方法, 以及合理使用针对视频处理而设计的特殊指令集, 充分利用 DM642 内部的并行计算单元, 提高了程序的运行速度。由于篇幅有限, 对这些优化方法本文不再重点论述。

4 结论

结合 AVS-M 视频压缩处理流程的特点, 本文完成了一个基于 DM642 平台的编码器的设计与实现。通过对编码流程的合理安排使得 CPU 能与 DMA 控制器并行工作, CPU 不用等待数据, 需要的数据已经被 DMA 调到内存中。实验表明通过系统级优化、程序级优化、汇编级优化、算法级优化等优化之后, 基于这款视频服务器(实物图见图 5), 能达到 2 路 CIF352x288 格式实时视频压缩, 以及音频实时编码、解码处理, 且图像主观效果及音频效果良好。(下转第 295 页)

技术创新

时我们认为是猛踩刹车,就要对图像进行飘动处理,进行飘动处理的时候要当前行车的速度进行判断,速度大图像就飘动得厉害,否则飘动就小一点。

3 结论

本文主要研究的主要研究目标是改进虚拟现实系统中的三维图像的生成技术。现将工作总结如下:

特殊环境行车(夜间行车、雾天行车、雨天行车、雪地行车)的三维场景的绘制。这是通过使用基于建模的方法来实现的,即绘图工作是在建立好模型的基础上完成的。在图像的绘制过程中主要用到了 OpenGL 的一些特殊功能如纹理贴图、雾化、光照、材质等。绘制好的场景通过和汽车采样信号的配合比较逼真的模拟了以上所述的几种特殊环境下驾驶汽车的情形:雾天行车和夜间行车分别再现了白雾茫茫以及漆黑一片的朦胧效果,使得驾驶者不由得集中精力小心避让前方以及后方的来车;雨天行车再现了一个雨天的世界,由于可视度不高,驾驶员在雨中行车时也要十分小心;雪地行车再现的是大雪过后的那种天寒地冻的效果,驾驶员必须注意谨慎操作。所作三维图像虽然在实际中已经够用,但是实际的视觉效果不是十分逼真,与实际现实还存在一定的距离。图像中存在锯齿现象,图像的纹理欠逼真,路边的树木的设计上还存在缺陷。

本文作者创新点:在 Visual C++ 环境下通过三维绘图工具 OpenGL,在图像的绘制过程中用到了纹理贴图、雾化、光照、材质等特殊功能,完成了汽车模拟驾驶系统中特殊环境行车(夜间行车、雾天行车、雨天行车、雪地行车)的三维场景的绘制。

参考文献:

- [1]汪成为,高文,王行仁. 灵境(虚拟现实)技术的理论、实现及应用 清华大学出版社 1996;
 - [2]黄干,徐学军.基于机器视觉的汽车智能驾驶系统研讨[J].微计算机信息,2004,6
 - [3]王俊修.用 VC++.Net 实现退化图像的恢复.微计算机信息[J].2004,4
 - [4]李锦涛等.虚拟现实技术概论专题.中国计算机报,1998,1,17
 - [5]何克抗.运用新一代 VR 技术建立高质量教育与培训系统的研究.1997 年度国家自然科学基金申请书
 - [6]Joshua Eddings,How Virtual Reality Works, 电子工业出版社 1994 年中文译本
 - [7]Burdea G,Virtual Reality systems and Applications,Electro'93 International Conference
 - [8]Wexelblat A Virtual Reality:Applications and Explorations, Boston,MA:Academic Press Professional, 1993
- 作者简介:杜飞,男,1982 年 11 月生,汉族,北京科技大学电子信息系系统工程专业硕士,主要研究领域为虚拟现实技术 E-mail:fly416@sina.com;钟延炯,男,1934 年生,汉族,江西南昌人,教授、博士生导师,主要从事电子技术在汽车上的应用、基于虚拟现实的汽车模拟

驾驶系统的研究。

Biography:DuFei, male, Han Nationality, 1982.11,master, system engineering major, mostly undertake the Virtual Reality.

(100083 北京科技大学信息工程学院)杜飞 李巍巍 钟延炯 唐祥宝

(School of information and engineering, University of Science and Technology Beijing, Beijing 100083) Du Fei, Li Weiwei, Zhong Yanjiong,Tang Xiangbao

通讯地址:(100083 北京科技大学 102# 信箱)杜飞

(收稿日期:2006.2.23) (修稿日期:2006.3.18)

(上接第 48 页)

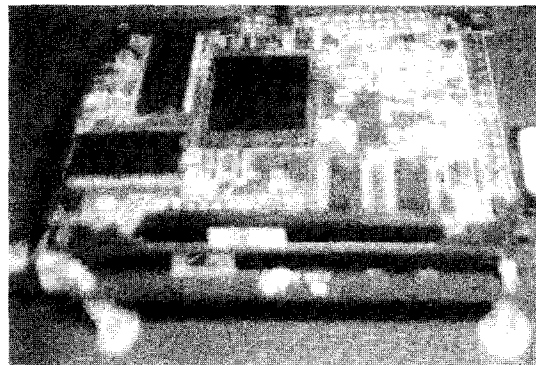


图 5 Imlab6421 视频服务器实物图

本文创新点是:把具有自主知识产权的数字音视频编解码技术标准第七部分(AVS-M)应用于视频服务器的视频压缩,目前市场上还没有采用此压缩标准的产品,此产品具有极高得性价比,采用此压缩标准还可以避免产品产业化之后知识产权之争,具有很好的应用前景。

参考文献:

- [1]周大山,李华,张淑芳等. AVS-M 视频解码器设计和实现. 电视技术,2005, 8:10-11
- [2]李方慧,王飞,何佩琨.TMS320C6000 系列 DSPs 原理及应用.电子工业出版社,2003.
- [3]许海燕,闫健恩,陈静. H.264 视频编码在 DM642 上的实现与优化,微计算机信息,2006,8-2:160-162

作者简介:周广禄,男,讲师,硕士,哈尔滨工业大学(威海)计算机科学与技术学院,E-mail:zhouguanglu@sohu.com,主要研究方向数字图像处理;郭恒业,男,教授,哈尔滨工业大学,计算机学院,研究方向:网络安全、嵌入式系统、图形图像压缩、识别。

Biography:Zhou,Guanglu,male,graduate, Haerbin Institute of Technology,major in DSP.

(264209 哈尔滨工业大学(威海)计算机科学与技术学院)周广禄 王翠华 郭恒业

(264209 School of Computer Science and Technology,Harbin Institute of Technology at Weihai)Zhou,Guang-lu

通讯地址:(264209 哈尔滨工业大学(威海)计算机科学与技术学院)周广禄

(收稿日期:2006.1.15)(修稿日期:2006.2.15)