

DM642 在网络多媒体开发平台中的应用

刘星星

(深圳巨龙科教高技术有限公司研发部 深圳 518057)

摘要 随着宽带网络的普及和高效音、视频标准的不断推出,利用互联网进行音、视频节目的实时传输已经成为多媒体技术应用领域一个重要的研究方向。DM642 是美国德州仪器公司推出的最新的一款媒体处理器,它为多媒体产品应用提供了理想的解决方案。DM642 的性能在已推向市场的各种媒体处理器中位于领先地位,将会广泛应用在网络多媒体开发平台的设计中。

关键词 DM642, 媒体处理器, 多媒体开发

The Study of Network Multimedia Developing Platform Designs Based on DM642

LIU Xing-Xing

(Department of R and D, Shenzhen Julong Science and Education High Technology Limited Company, Shenzhen 518057)

Abstract With the popularization of the broadband network and the introduction of many high-efficient audio and video compression standards, the real-time transmitting of video and audio data over internet has become an important research field of multimedia technology. DM642 is a new media processor provided by Texas instrument Company, it offers ideal solutions for high performance multimedia applications. DM642 makes a leading performance among all the media processors introduced into market, and will be widely used in network multimedia develop platform designs.

Keywords DM642, Media processor, Multimedia develop

1 DM642 性能结构简介

随着多媒体应用需求越来越大,市场上对专用于多媒体处理的 DSP 处理器的需求越来越大。通常,视频处理应用对处理器和系统有如下要求:高速的运算能力;高速的数据传送;大容量的数据存储;灵活专用的视频接口等。作为 DSP 业界的行业巨头,TI(美国德州仪器)公司日益看好这个市场,近年来对专用于媒体处理的 DSP 进行了重点投入和开发,并规划了专用于视频和图像应用的 DSP 产品路线图。数字多媒体处理器 DM642 是 TI 公司 C6000 系列的一款新型高性能 DSP,基于 C64x 内核,扩展的高级甚长指令字(VelociTI)体系结构,具有 64 个 32 位通用寄存器,8 个独立计算功能单元(2 个乘法器,6 个算术逻辑单元)可以并行运行,因此多条指令可同时执行,可工作在 600MHz 时钟速率,每个指令周期可并行运行 8 条 32 位指令,因此可达到 4800MIPS 的峰值计算速度。DM642 采用两级缓存结构,L1P, L1D, L2。DM642 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。容量较大的两级缓存和 EDMA 通道是 DM642 高性能的体现之一,若能合理使用和管理,将能大幅度提高程序的运行性能。

DM642 具有丰富的外围设备接口。包括三个可配置的双通道视频端口 video port,每个 video port 又分成 A、B 两个通道,A/B 通道可以分别处理一路视频采集,因此 DM642 最多可以处理 6 路视频采集数据(不带音频)。如果将 video port 配置成用于视频输出,则只能在 A 通道输出,B 通道不可以,因此 DM642 最多可支持 3 路视频输出(不带音频)。如果同时处理音频,每一个 video port 可以处理两路立体声,以

及 64bit 的外部内存接口(EMIF)可连接异步或同步存储器,如 SDRAM。总共有 1024M 外部可寻址存储空间。还包括符合 IEEE 802.3 规范的 10/100M 以太网 MAC,具有媒体独立接口(MII),可直接与 PHY 器件连接。包含 8 个独立传输(TX)和 8 个独立接收(RX)通道。此外,还具有用户可配置的 16bit 或 32bit 的主机接口(HPI16/HP 工 32)、两个多通道缓存串口(McBSP)、I2C 总线模块、数据管理输入输出通道(MDIO),以及通用 GPIO 端口等外设接口。可见,DM642 特别适用于开发基于以太网的网络多媒体通信设备。功耗较低的高性能 DM642 能同时处理四路帧率为 30 帧、分辨率为 D1 的 MP@ML MPEG2 解码,或分辨率为 D1 的最新视频标准 H264, WMV9 的实时编解码,是目前构建数字多媒体处理应用的一个理想平台。

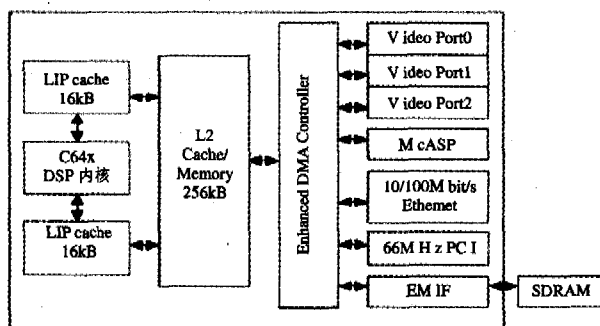


图 1 DM642 系统体系结构

2 DM642 网络多媒体开发平台硬件系统的设计

2.1 视频输入输出模块设计

刘星星 研究方向为计算机应用方面的网络编程、音频开发。

2.1.1 视频捕捉方案

TI公司的视频解码芯片 TVP5145 内部有两个 10 位的 A/D 转换器。它支持模拟的复合视频基带信号 (CVBS)、YPbPr 分量信号以及 S-video 输入,输出的信号格式可以是 16 位或 8 位的 4:2:2 格式,或者是 8 位的数据流内自带同步信息的 BT656 格式。TVP5145 的输出信号包括数字视频信号、同步信号、消隐信号、行有效信号、奇偶场信号和时钟信号。TVP5145 的主机端口可配置成四种模式: I2C, PHI 的 MODE_A, MODE_B 和 MODE_C。通过主机端口,可以配置 TVP5145 的工作参数,调节 TVP5145 输出视频的亮度、色度和饱和度等。本系统中, TVP5145 输入信号为 CVBS 信号,输出为 8-bit BT656 格式。DM642 通过 I2C 接口与 TVP5145 进行通信。上电复位时, TVP5145 检测管脚 GLCO, PAL1 和 FID 的电平值,以确定主机端口模式采用哪种配置方式。在该系统中, TS 流和图像匹配处理机需要的原始视频数据的捕获也是通过 VP1 进行的。

2.1.2 视频显示方案

PAL/NTSC 视频编码器 STV0118 从 DM642 的 VP2 接收到的是 8-bit BT656 格式的数据。STV0118 从 BT656 数据流中提取 SAV 和 EAV 标志码以产生编码所需要的同步信号。由于没有使用 VP2 提供的行、场同步信号,与 DM642 的接口相对比较简单。DM642 视频端口的数据输出是由引脚 CLK0 输入的时钟驱动的,但是 STV0118 并不提供 27M 的时钟输出,因此在选用该方案时,必须为 VP2 的 CLK0 引脚提供一个 27MHz 的时钟,可以选择时钟恢复电路产生的 27MHz 时钟为该引脚输入。

2.2 音频模块的设计

DM642 的 McASP (多通道音频串行口) 是一个通用的串行音频接口,可以和多种音频设备连接。它支持 IIS (integrated sound) 协议和 DIT (digital audio interface transmission) 音频接口标准。McASP 包含音频输入、输出模块,两个模块可以以不同的传输模式,独立地按照主时钟、位时钟、和帧同步脉冲工作。发送和接收模块也可以工作在同步模式。系统设计中没有用到的 McASP 引脚均可以配置为通用 I/O 引脚。McASP 的外部接口非常灵活,可以和音频 ADC、DAC、音频 CODEC 等多种器件实现无缝接口。

在进行音频模块的设计时,采用 TI 公司的数字音频编解码芯片 TLU320AIC23 进行音频信号的输入和输出。AIC23 将麦克风输入的模拟音频信号或者 LINE IN 输入的立体声模拟信号进行 AD 变换,转换为 DSP 可以处理的数字音频信号;或者从 DSP 接收数字音频信号进行 DA 变换,输出模拟音频信号到耳机或扬声器。同时, AIC23 可以将输入的模拟音频信号在进行 AD 变换之前旁路到输出通道,和 DA 转换器输出的音频信号进行混音。将 DM642 通过 I2C 总线与 AIC23 进行通信,设置 AIC23 的工作模式和各项参数。

2.3 网络传输模块设计

EMAC/MDIO 是 DM642 的片内外设,它包括三个部分: EMAC 控制模块、EMAC 模块、LV1D10 模块。EMAC 控制模块是 DSP 核与 EMAC 模块和 MDIO 模块之间的接口,其主要功能是将 EMAC 和 MDIO 的寄存器映射入 DSP 的配置空间,使 EMAC 模块可通过 DSP 的存储器传输控制器存取 DM642 的片内和片外存储器。EMAC 模块提供了 DSP 核与以太网收发器间的高效接口,支持 10BASE-T 和 100BASE-TX 的全双工或半双工数据传输 MDIO 模块与 EMAC 模块

配合使用,它通过 MDIO 总线与物理层器件连接,可轮询最多 32 个 MDIO 地址,即管理 32 个器件。

将 DM642 的 MDIO, EMAC 与网络收发器 LXT971ALC、网络隔离变压器、RJ-45 插头相连接,共同构成系统的网络通信块,实现 DM642 的以太网通信模块。其中 LXT971A 是以太网物理层收发器,支持三种物理层标准: 10BASE-T, 100BASE-TX, 100BASE-FX。

3 DM642 网络多媒体开发平台的软件系统

3.1 DSP/BIOS 实时操作内核

DSP/BIOS 是为 TI 公司 TMS320C5000 和 TMS320C6000 系列 DSP 设计的一个可裁减的实时操作内核, DSP/BIOS 包括抢占式多任务调度器、外设管理工具、实时分析工具、图形配置工具等,是 TI 集成软件开发工具 CCS 的一个组成部分。DSP/BIOS 为 DSP 应用程序开发者提供了一个标准化的结构,围绕这个结构来组织软件模块和单元,开发者就可以回避在项目中很多底层的软件开发,而尽可能地把精力集中到应用要求的特定算法上。

DSPBIOS 支持交互式的程序开发流程。用户可以首先创建基本的程序框架,使用替代函数模拟 DSP 算法的 CPU 负载情况来检验程序框架的正确性,在这个过程中,用户可以很容易地改变系统各线程的优先级和类型。等程序框架开发成熟后再将 DSP 算法添加到应用程序中。

3.2 片级支持库 CSL

在 DSP 应用系统中,一般会涉及到大量对 DSP 器件外设特别是对片上外设的编程处理工作,在开发初期消耗用户较多精力。TI 公司 CCS 开发环境中,提供了 DSP 片级支持库 CSL 作为一个自包容组件被归于 DSP/BIOS 中,多数 CSL 模块都由对应函数、宏、类和表示符号构成,在不调用其他 DSP/BIOS 组件的情况下也可以简单方便地完成对 DSP 器件片上外设的配置和控制,从而简化了 DSP 片上外设编程工作,缩短了开发周期,而且可以达成标准化控制管理片上外设的能力,减少硬件特殊性对应用程序代码的影响,方便用户代码在不同器件间的移植。

每个 CSL 模块都用有单独的支持符号,若选定的 DSP 器件支持这一 CSL 模块,则模块的支持符号在编译过程中置为 1,不支持则置 0。例如 DM642 支持 VP 模块,则与 VP 对应的支持符号 VP_SUPPORT 在编译的过程中置 1。用户必须在 CCS 的编译器中添加编译参数“-d”来设置所选择的芯片类型,编译器根据设置的芯片类型来判断哪些 CSL 模块的支持符号被设为 1,哪些被设为 0。对于 DM642,需要在编译器参数中添加“-d CHIP_DM642”。

3.3 网络开发工具包 NDK

TI 公司提供的 C6000 TCP/IP 网络开发工具包 NDK (Network Developer's Kit) 是 DSP/BIOS 和 CCS 的一个附件。它采用紧凑的设计方法,仅用 200~250kB 程序空间和 95kB 数据空间即可支持常规的 TCP/IP 服务,包括应用层的 telnet, DHCP, HTTP 等。为了最大限度地减少资源消耗, NDK 采用了许多特殊技巧,其中比较重要的包括底层驱动程序与 TCP/IP 协议栈之间通过指针传递数据,不对包进行复制拷贝。另外, NDK 还设置专门的线程清除存储器碎片和检查存储器泄漏,这些线程由 NDK 提供,不必由程序员编写。

与常规的 TCP/IP 开发不同,在以 NDK 为基础为系统增加网络通信功能时,程序开发者必须对网络环境和应用需求

作更细致的设置。需要完成的主要工作包括:第一,创建网络初始化线程;第二,调用初始化函数 NC_SystemOpen()。该函数完成对协议栈及其所需要的内存的初始化;第三,创建系统配置,该系统配置用于对协议栈的控制和管理,可用 CfgNew()和 CfgLoad()等函数操作;第四,调用 NETCTRL 函数 NC_NetStart()启动网络。NC_NetStart()函数的参数中包含三个回调函数指针,分别处理“Start”、“Stop”和“IP Address Event”事件,其中“Start”和“Stop”只执行一次,“IP Address Event”则响应每次 IP 地址的变化。由于 NDK 已经提供了完整的 TCP/IP 库函数,程序员开发的代码只须按需要进行配置即可。

参 考 文 献

1 李方慧,王飞,何佩琨著. TMS320C6000 系列 DSPs 原理与应用

(第 2 版). 北京:电子工业出版社,2003

- (美)Incorporated TI 著, TI DSP 集成化开发环境 (CCS) 使用手册. 彭启琮, 张诗雅, 常冉, 等编译. 北京:清华大学出版社, 2005
- TMS320DM642 Video/Imaging Fixed-Point Signal Processor Manual. May 2003. Texas Instruments Incorporated
- DSP/BIOS Workshop: A Real-time Software Designer's workshop (student guide). July 2003. Texas Instruments Incorporated
- Texas Instrument. TMS320C6000 Chip Support Library API Reference Guide, June 2003
- Wiegand T, Sullivan C J, Bjontegaard G, Luthra A. Overview of the H. 264/AVC Video Coding Standard. IEEE Transactions on Circuits and Systems for Video Technology, July 2003

(上接第 99 页)

精确匹配策略, 很难匹配成功:

- 处于正交关系的“Desc”和“Place”属性颠倒, 这需要“对称翻转匹配”策略的支持;
- 服务查询条件中缺少“Effect”属性描述, 这需要“部分匹配”策略的支持;
- 查询表达式与服务描述中“Language”属性值虽有语义包含关系, 但无法精确匹配, 这需要“模糊匹配”策略的支持。

基于 XPath 的查询很好地解决了“对称翻转匹配”与“部分匹配”的问题。但“模糊匹配”却因为常常与语义相关, 尚未被现有的服务发现和定位相关工作所涉及。我们为提高服务定位过程的准确性和效率, 引入了如下基于模糊匹配的服务查询策略:

(1) 为服务描述中的每个属性根据属性重要性及对匹配精确度的要求设置一个权重值 W , 取值于 $[0, 1]$ 区间, 其中, 固定标签的服务属性通常定义一个较大的权重值, 而自定义标签的属性通常定义一个较小的权重值; 易精确匹配的服务属性, 如服务类型通常赋予较大的权重值, 而难于精确匹配的服务属性; 如服务描述通常赋予较小的权重值;

(2) 匹配过程中, 对每个属性的匹配程度 M 提供形式化度量, M 也取值于 $[0, 1]$ 区间:

a) 对于服务描述中出现, 而查询条件中未出现的服务属性, 设置匹配程度 $M=1$, 以提供对部分查询的支持;

b) 对于数值描述的服务属性, 提供属性标准值 v 及偏差范围 π , 若查询条件中属性取值为 λ , 则以 $M=1-|\lambda-v|/\pi$ 计算属性匹配程度;

c) 对于文字描述的服务属性, 假设服务描述中属性值字符串长度为 l_1 , 查询条件中属性值字符串长度为 l_2 , 计算属性描述与查询条件的最大匹配子串, 设其长度为 l_3 , 则属性匹配程度 M 可以通过如下公式计算: $M = \sqrt{\frac{l_3}{l_2} * \frac{l_3}{l_1}} = \frac{l_3}{l_1 * l_2} * \sqrt{l_1 * l_2}$

(3) 设置属性权重值的边界值 ω , 对于 $W \geq \omega$ 的所有属性, 其匹配程度 M 不得小于阈值 ϵ ; 若 $W \geq \omega$ 且 $M < \epsilon$, 则认为该属性节点匹配失败, 匹配过程结束。

(4) 若遍历了 XPath 表达式的所有查询条件, 均未匹配失败, 则计算总的查询匹配相似度 $Z = (\sum_{i=1}^N (W_i * M_i)) / N$, N 为服务描述中总的属性个数。设定匹配相似度阈值 k , 若 $Z \geq$

k , 则认为找到了与查询条件相匹配的服务资源, 其匹配相似度为 Z 。

(5) 按匹配相似度由大到小的顺序向用户返回匹配的服务资源, 供用户选择。

采用上述模糊匹配策略, 可以弱化服务信息描述与用户请求描述无法精确一致带来的影响, 提高服务资源匹配的 success 率和准确性。

小结 本文分析了移动环境对分布式条件下服务资源定位和访问机制带来的影响, 介绍了一个在移动性条件下、具有服务资源主动适配能力的服务适配系统: Service CatalogNet, 研究了其中的关键技术问题。服务适配系统在广域网环境下的扩展, 具有分级结构的代理覆盖网络, 以及代理服务器上的并发访问、负载均衡和失效恢复机制将是我们的继续研究的重点。

参 考 文 献

- Lee Choonhwa, Helal S. Protocols for Service Discovery in Dynamic and Mobile Networks. International Journal of Computer Research, 2002, 11(1): 1~12
- Sun Microsystems. Jini Community Resources; Jini Technology Architectural Overview. January 1999. <http://www.sun.com/jini/whitepapers/architecture.html>
- Microsoft Corporation. Universal Plug and Play Device Architecture Version 1.0. June 8, 2000. <http://www.upnp.org/download/UPnPDA10-20000613.htm>
- Guttman E. Service Location Protocol: Automatic Discovery of IP Network Services. IEEE Internet Computing, 1999, 3(4): 71~80
- Adjie-Winoto W, Schwartz E, Balakrishnan H, et al. The Design and Implementation of an Intentional Naming System. 17th ACM Symposium on Operating System Principles (SOSP'99), published as Operating System Review, 1999, 34(5): 186~201
- Balazinska M, Balakrishnan H, Karger D. INS/Twine, A Scalable Peer-to-Peer Architecture for Intentional Resource Discovery. In: Proceedings of the First International Conference on Pervasive Computing. Switzerland, August 2002. 195~210
- Czerwinski S E, et al. An Architecture for a Secure Service Discovery Service. In: Fifth Annual International Conference on Mobile Computing and Networks (MobiCom '99). Seattle, WA, 1999. 24~35