

DM642 在网络视频传输系统中的应用

Application of DM642 in Network Video Transmission System

王 玮 张卫宁 张大志 史 英

(山东大学信息科学与工程学院, 济南 250100)

摘 要: 数字信号处理器 TMS320DM642 在多媒体通信领域有着广泛的应用。以利用 XDAIS 算法在 DM642 上实现网络视频编码器为例,研究了 DM642 在网络视频传输系统开发中的一般过程,介绍了系统的硬件和软件模块的功能、工作原理和 RF5 在程序中的使用方法,并详细分析了使用 NDK 进行网络功能开发的过程。实践证明,所用方法是正确有效的。该系统模型可作为网络视频监控系统的通用平台。

关键词: XDAIS 算法 参考框架 5 网络开发工具包

中图分类号: TP368.1 **文献标志码:** A

Abstract: Digital signal processor TMS320DM642 has been widely used in multimedia communications. With realizing network video encoder on DM642 with XDAIS algorithm as example, the general process of developing network video system with DM642 is studied. The functions and working principles of hardware and software modules and the use of RF5 in program are presented. The procedures of developing network function with NDK are analyzed in detail. The practice shows that the method is correct and effective. The system model can be used as a universal platform for network video monitoring system.

Keywords: XDAIS algorithm Reference framework 5 (RF5) Network developer's kit (NDK)

0 引言

数字视频的编解码及网络传输是视频通信的关键技术,采用专用的编解码算法芯片虽能保证实时性,但系统的灵活性却大大降低,不利于算法升级;纯软件实现的系统虽易于升级和移植,但实时性能较差;TI 公司专为多媒体应用而设计的芯片 TMS320DM642(简称 DM642)则很好地解决了这些问题,其丰富的外围接口使得它近乎是一个多媒体嵌入式系统的单芯片硬件平台,它的完全可编程性,又使其能兼容正在发展的各种多媒体信号处理标准,构成通用的软件平台。利用嵌入式处理芯片为核心处理器的软硬件相结合的方法已成为一种有效的视频编解码实现途径。本文即以利用 XDAIS 算法在 DM642 上实现网络视频编码器为例,研究 DM642 在网络视频传输系统中的应用。

1 DM642 多媒体处理器

DM642 是 TI 公司推出的一款面向数字多媒体应用的高性能 32 位定点 DSP 芯片^[1],该芯片采用基于 C64X 的 DSP 内核,工作频率最高达 720 MHz,处理性

能可达 5 760 MIPS。DM642 配有丰富的外围设备和接口,包括专为视频处理而增加的视频口(简称 VP 口),具有比较完整网络功能的以太网接口及 64 位外存存储器接口 EMIF 等众多外设接口,特别适用于开发基于以太网的网络多媒体通信设备。

DM642 具有两级 Cache 结构,其中第一级 Cache 分为 L1P(16 kB)和 L1D(16 kB),而第二级 L₂(256 kB)既可整体作为 SRAM 映射到存储空间,也可整体作为第二级 Cache,或是二者按比例组合使用。此外,DM642 有 64 个独立的 EDMA 通道,负责片内 L₂ 与其它外设之间的数据传输,每个通道都有一个特定的事件与之关联,由这些事件触发相应通道的传输,EDMA 可以在没有 CPU 参与的情况下,完成映射存储空间中的数据搬移,极大地提高了系统的并行性能。容量较大的两级 Cache 和 EDMA 通道是 DM642 高性能的体现之一。

2 系统硬件部分

DM642 上集成了 3 个 VP 口^[2],每个 VP 口分为 A 和 B 两个通道,每个通道都可被配置为视频输入或输出口,但 A、B 两个通道只能同时被配置为输入或输出口,所以 DM642 最多支持 6 路视频输入。每个 VP 口都有一个 5 120 字节的视频输入/输出缓冲区,当 VP

第一作者王玮,男,1983 年生,现为山东大学信息科学与工程学院在读硕士研究生;主要研究方向为数字视频处理与通信和基于 DSP 的嵌入式系统设计。

口作为输入口时,缓冲区按照 A、B 两个通道分为两个(各为 2 560 字节),每个通道的缓冲区又分为 Y 缓冲区(1 280 字节)、Cb 缓冲区(640 字节)和 Cr 缓冲区(640 字节)。VP 口作输入或输出时,均支持 8/10 位的 ITU - R BT. 656、16/20 位的 Y/C、8/10/16/20 位的 raw 等多种视频格式。本系统中,使用 VP1 的 A 通道作为输入口并采用 8 位的 ITU - R BT. 656 格式。

DM642 的网络接口主要用来支持物理层网络器件(简称 PHY)与 DSP 的连接,由 EMAC (Ethernet media access controller)与 MDIO (management data input/output)两部分组成。EMAC 为网络的数据通路,它控制 PHY 与 DSP 之间数据包的交换;MDIO 是 EMAC 的状态及控制接口,它控制 PHY 的配置并监测其状态。网络接口符合 IEEE802.3 协议,支持传媒无关接口(MII),并具有 8 个独立的发送与接收通路,支持同步的 10/100 Mb 的数据操作并支持广播及多帧的传送。

VP 口缓冲区与片内 L_2 存储器间的数据传递通过 EDMA 实现^[3],BT. 656 格式的数据流按照 Cb_0 、 Y_0 、 Cr_0 、 Y_1 、 Cb_2 、 Y_2 、 Cr_2 、 Y_3 、 Cb_4 、 Y_4 、 Cr_4 、 Y_5 ……的串行传输顺序,经由 BT656 捕获通道分别进入到各自的缓冲区中并打包成 64 位的双字,当双字大小增至缓冲区的设定阈值时,即触发 EDMA 事件,其中 VP1 口 A 通道的 Y、Cb、Cr 缓冲区数据的 EDMA 传输对应的通道号分别为 56、57 和 58。EDMA 适合完成 L_2 与外设之间固定周期的数据传输,如果需要由 CPU 直接控制搬移一块数据,则更适合采用 QDMA(快速 DMA),QDMA 尤其适用于需要快速传递数据的场合,如紧耦合循环代码中的数据搬移任务。

系统实现了单路视频信号的采集及输入,在 DM642 上对视频信号进行 H. 263 编码,通过网络接口实现编码流的发送,由此实现网络视频编码器。采用图像解码芯片 TVP5150 来完成视频图像的 A/D 转化以及对水平同步和垂直同步等信号的分离,TVP5150 将 PAL 制的模拟视频信号转化为 8 位 BT656、4:2:2 的数字视频流后发送给 DM642 的 VP1 口的 A 通道。网络接口的 EMAC 属于数据链路层,因此需外接物理层网络器件才能进行网络通信,PHY 选用 Broadcom 公司的 BCM5221。系统结构如图 1 所示。

3 系统软件部分

3.1 XDAIS 及参考框架 RF5

XDAIS(eXpressDSP algorithm standard)^[3]是 TI 针对 TMS320 DSP 的代码可重复利用提出的编程准则和

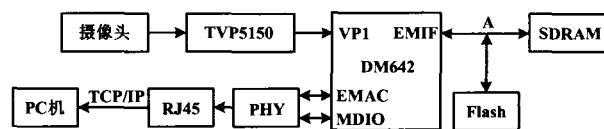


图 1 系统总体结构图

Fig.1 Overall structure of system

建议,其目的是使按照该标准开发的 XDAIS 算法能独立于具体的应用系统,使算法无需或只需很少的修改就可集成到新的 DSP 系统中。目前已有用于语音编码的 G. 723、G. 729 和用于图像编解码的 JPEG、H. 263 等 XDAIS 算法。XDAIS 算法需在 CCS 集成开发环境下,利用 DSP/BIOS 及 TI 的第三方提供的标准软件模块库才能使用。所有的 XDAIS 算法都有一个算法实例的抽象接口 IALG 用来创建算法实例对象和定义存储器资源需求;使用 DMA 设备的算法还有一个 IDMA 抽象接口,使用该接口的算法又必须使用接口 ACPY(算法复制模块),因为 IDMA 可能用到此接口。

RF(reference framework)^[4]是一种使用 DSP/BIOS 及 XDAIS 的通用初始化代码,将应用程序、XDAIS 算法和底层结构结合在一起形成的一个框架空间。参考框架标准 5(RF5)专用于由 C6X 实现的高端系统,适用于包含大量算法、多线程、多通道的应用,如图像处理、多媒体应用等。RF5 中的数据处理分为四部分:任务(task)、通道(channel)、核(cell)和 XDAIS 算法,如图 2 所示。

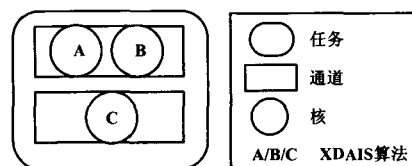


图 2 RF5 的基本数据元素

Fig.2 Basic data elements of RF5

每个任务通常包含一个或多个通道(简单的任务也可以不包含任何通道),任务顺次执行所包含的通道,并与设备驱动及其它任务进行数据通信;通道是核的集合,每个通道对象可以包含多个核;每个核包含一个 XDAIS 算法,真正的数据处理在 XDAIS 算法,核只是提供一个调用算法的接口,通道即通过调用这些算法接口,依次顺序地执行核内的算法。

本系统即使用 RF5,创建输入、编码、网络三个任务,其中的编码任务使用一个通道,通道中包含一个核,核内为 H. 263 编码 XDAIS 算法。算法使用时可根据需要对编码参数(传输码率、图像格式等)进行设置。

3.2 软件模块介绍

系统涉及的主要软件模块有 FVID、SCOM、Cache 和 DAT 等模块。

FVID 模块^[5]为基于 DSP/BIOS 的应用程序提供 API 函数,该模块以芯片支持库(CSL)和设备驱动为基础,实现帧图像的获取和显示。模块集中完成了 CSL 中 EDMA、VP、IRQ 和 I²C 四个模块的配置和控制,实现了 VP 口与 L₂ 之间固定周期的 EDMA 传输/中断以及 VP 口中断这些固定操作,极大地简化了编程工作,而 FVID 的 API 主要任务是管理设备驱动和程序之间缓冲区的所有权。大多数 DSP/BIOS 设备驱动的数据缓冲区的最初所有权都由应用程序掌握,数据的输入/输出通过传递缓冲区的地址到设备驱动实现;而 FVID 模块的设备驱动掌握数据缓冲区的最初所有权,应用程序按需求来分配缓冲区,这是因为视频采集/显示系统通常需要较大且复杂的缓冲区,而且对数据读取速度的要求要远高于对缓冲区分配灵活性的要求所致。模块使用时首先用 FVID_create 创建并初始化一个 FVID 通道,之后通过 FVID_control 发送控制命令到设备驱动以配置相应的视频解码器(TVP5150)并启动视频帧图像的采集,设备驱动工作后,利用 FVID_alloc 为应用程序分配视频帧缓冲区,当应用程序处理完成后,再用 FVID_exchange 完成应用程序与设备驱动间缓冲区的交换以为后续处理作准备。

Cache 模块^[3]用于配置和管理数据及程序 Cache,由于 CPU 对 Cache 进行直接访问,因此 L₂ 中 Cache 的设置及使用直接影响系统性能,其中 Cache_setL₂Mode 用于设置二级 Cache 大小;而 Cache_enableCaching 则设置二级 Cache 可映射的外部存储器的地址空间。

DAT 模块^[3]是 CSL 的一个抽象模块,其实现的功能是通过 DMA/EDMA 进行数据传递。对于有 EDMA 的 DSP,模块则用 QDMA 进行传输,DM642 使用的就是 QDMA。使用该模块需首先用 DAT_open 打开模块,之后用 DAT_copy 完成线性数据块的拷贝传送。

SCOM 模块^[4]是 RF5 中不同任务间传递消息的通信模块。SCOM 消息在任务间的传递通过 SCOM 队列实现,SCOM 队列由 SCOM_create 创建,任务用 SCOM_putMsg 把消息放到 SCOM 队列中完成消息的发送,而接收消息的任务则用 SCOM_getMsg 将消息从队列中取出,由此实现任务间的通信。

3.3 网络功能的软件实现

3.3.1 NDK 的结构框架

系统使用 NDK(network developer's kit)^[6]开发网

络功能,NDK 是 TI 公司配合其推出的 TMS320C6000 系列 DSP 的 TCP/IP 协议栈而开发的新型网络开发工具包,其目的是提供一个完整的 TCP/IP 功能环境。NDK 采用紧凑的设计方法,实现了用较少的资源耗费支持 TCP/IP,仅用 200 ~ 250 kB 的程序空间和 95 kB 的数据空间即可支持常规的 TCP/IP 服务,是 DSP 网络开发的上等之选。

图 3 为 NDK 中 TCP/IP 协议栈的流程框图,其中五个网络功能模块对应的函数库分别为 OS.LIB、HAL.LIB、STACK.LIB、NETTOOL.LIB 和 NETCTRL.LIB。NDK 通过编程接口与本地操作系统(DSP/BIOS)和底层硬件相互隔离,其中 DSP/BIOS 被抽象为一个操作系统适应层,底层硬件抽象为一个硬件抽象层,它们是 NDK 与本地操作系统和底层硬件的接口;TCP/IP 协议栈库是主要的 TCP/IP 网络功能库,该库针对 DSP/BIOS,实现了从上层套接口层至底层链路层的所有功能;网络工具库包含了 NDK 提供的所有基于套接字的网络功能及一些开发网络程序的其它工具;网络控制及调度模块是协议栈的配置、初始化和事件调度的核心,它控制着 TCP/IP 与外界的联系及互动,它负责初始化协议栈和底层设备驱动、加载并保持系统的配置、与底层设备驱动接口并安排底层事件传递进入协议栈以及退出时将系统配置导出并清理底层驱动。

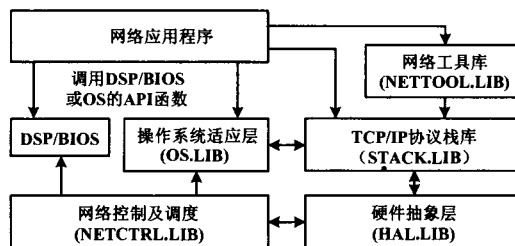


图 3 TCP/IP 协议栈的流程图

Fig. 3 Flowchart of TCP/IP stack

3.3.2 TCP/IP 协议栈的配置和初始化

协议栈的正确配置及初始化是应用程序网络功能正确执行的前提。通常用 DSP/BIOS 配置工具静态或者用 DSP/BIOS 的 API 函数动态创建一个“网络主线程”,它作为网络的入口,首先在其中进行协议栈的配置和初始化,之后它最终成为协议栈的事件调度线程,只有在应用程序结束时该线程才返回。该线程并不是真正意义上的网络任务线程,而应用程序的任务,无论是否网络相关,都要开辟新的线程,而不会在该线程中执行。协议栈的典型配置和初始化流程为:

```
NC_SystemOpen() -> CfgNew() -> CfgLoad()/CfgAddEntry() -> NC_NetStart()
```

其中 NC_SystemOpen() 用来初始化协议栈及其使用的存储器环境; CfgNew() 产生新创建配置的句柄, 该句柄是配置信息的入口; 之后用 CfgLoad() 将已有的配置从非易失性存储器中加载至配置入口或用 CfgAddEntry() 向配置入口逐条添加配置信息。典型的配置参数包括: 网络主机名、IP 地址、子网掩码、默认路由器 IP 地址、需要的服务 (HTTP、DHCP、DNS) 等。

当需要的配置被装载后, 通过 NC_NetStart() 启动网络事件调度, 该函数会自动调用“开始”、“结束”和“IP 地址事件”三个函数, “开始”函数只在系统初始化后准备执行网络程序时调用一次; “结束”函数在系统要结束时调用一次; 而“IP 地址事件”函数可被多次调用, 例如当本地的 IP 地址从系统中添加或删除时, 都会被调用。

除非系统关闭, 否则 NC_NetStart() 不会返回, 该函数的返回值是一个关闭代码, 系统的关闭方式会决定协议栈是否需要重新启动, 例如以下代码可以实现配置改变后的自动重启:

```
do
{
    rc = NC_NetStart( hcfg, NetworkOpen, NetworkClose,
NetworkIPAddr);
} while( rc > 0);
```

该例子中, 如果 NC_NetStart() 的返回值大于 0, 就重新启动协议栈。

若是协议栈关闭, 还需要用 CfgFree() 和 NC_SystemClose() 来删除配置和关闭系统。

3.3.3 网络通信模块设计

“网络主线程”用于完成协议栈的配置、初始化及网络事件调度, 而真正的网络任务线程则通过 NDK 的 API 函数 TaskCreate() 创建, 这里称为“服务器任务”线程, 该线程用“客户端/服务器”网络通信模式实现, 将 DM642 设计为视频服务器端, 利用套接字 (socket)^[7] 编程实现, 服务器负责响应客户端的请求并发送编码流; 同时在 PC 端用 VC++ 6.0 开发了基于 WinSock^[7] 技术的客户端, 负责向服务器端发起视频请求并按帧接收数据、H. 263 解码及回放。

由于数字视频的数据量大, 传输层协议若采用可靠的、面向连接的 TCP 协议必然会消耗过多的网络资源, 且不能保证实时性; 采用不可靠的、无连接的 UDP 协议传输数字视频信号则更符合实际应用的实时性要求, 因此系统选用 UDP 协议, 相应的套接字使用数据报套接字。系统网络通信部分的基本流程如图 4 所示。

对于“服务器任务”: 首先用 socket 建立一个套接

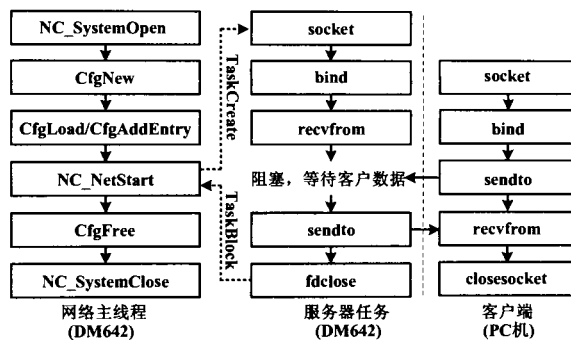


图 4 系统网络通信基本流程图

Fig. 4 Basic flowchart of system network communication

字以进行数据传输, 再用 bind 将本地 IP 地址和通信使用的端口号与该套接字绑定, 之后通过 recvfrom 阻塞等待客户端的服务请求, 当收到客户端的请求后, 再将编码后的数据流利用 sendto 发送给客户端。客户端也要先建立套接字并将本地 IP 及通信的端口号与其绑定, 之后使用 sendto 向服务器发送服务请求, 服务器应答后, 再通过 recvfrom 接收服务器发来的数据流。

3.4 程序流程

本系统利用 DM642 实现网络视频编码器, 软件框架采用 RF5, 利用 DSP/BIOS 创建输入任务、H. 263 编码任务和网络任务。在进入 DSP/BIOS 的调度程序之前, 需要执行三种类别的初始化: ①处理器和系统板的初始化, 包括初始化 BIOS 环境、CSL 以及 Cache 和 EDMA 的设置; ②RF5 模块的初始化, 包括通道模块、SCOM 模块等; ③FVID 通道的建立和启动; ④H. 263 编码 XDAIS 算法实例对象的建立。初始化完成后即进入 DSP/BIOS 调度程序管理下的任务, 任务之间通过 SCOM 模块传递消息。输入任务负责从输入设备获得视频图像, 利用 FVID_exchange 从输入设备获得一帧最新的 4:2:2 格式的 YUV 图像, 并将其采样为 YUV 4:2:0 格式, 将包含图像数据指针的消息发送给编码任务。编码任务将收到的 4:2:0 格式的 YUV 图像利用 XDAIS 算法进行 H. 263 编码, 之后将包含编码流大小和指针的消息发送给网络任务。网络任务包括“网络主线程”和在其中创建的“服务器任务”, 前者作为后者的基础和准备, 真正的数据发送则由后者完成。系统的软件流程如图 5 所示。

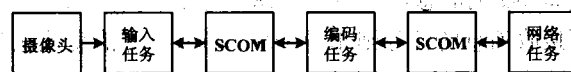


图 5 系统的软件流程

Fig. 5 Software process of system

(下转第 25 页)

使用 CP1613 网卡将以太网与工程师站或操作员站连接,将现场采集的数据传给上位工控机,进行数据的通信和变量传输。使用 CP343-1 网卡将以太网和西门子 S7 300 系列 PLC 连接,接收上位工控机的各种数据和控制命令以及进行程序的下载和调试。

3 系统的软件设计

3.1 软件系统结构

为了方便工作人员的操作,我们在操作员站上利用 WinCC 组态软件编制工艺流程画面、参数设定画面、历史趋势曲线、实时趋势曲线、报警画面、实时报表、系统参数等画面。不同的画面之间通过鼠标进行切换,操作灵活、方便(三个操作员站分别对应三个车间)。

工程师站上的利用 Step7 编写的 PLC 应用程序包括主程序、参数设定子程序、PID 控制子程序、模糊控制子程序、各种机泵的控制方式子程序等各种程序。

3.2 控制算法

在整个工艺中,需要进行很多数字量的逻辑控制(例如各种机泵的起停控制)和模拟量的 PID(控制各种液体的流量、温度、各种槽的液位等)。

使用 PLC 的数字量输入输出模块就可以满足数字量的逻辑控制要求。

而对于一些模拟量使用 FB41 的 PID 控制规律(即比例积分微分控制)就可以满足系统要求。介绍 PID 控制模块 FB41 中的位置式 PID 算法^[4],其算法如式(1)所示:

$$u(k) = K_p \left[e(k) + \frac{T}{T_i} \times \sum_{i=0}^k e(i) + T_D \times \frac{e(k) - e(k-1)}{T} \right] \quad (1)$$

式中: $K_p = 1/\delta$,为比例系数; T_i 为积分时间常数; T_D 为微分时间常数; $u(k)$ 为控制量; $e(k)$ 为偏差。

在本系统中还需要对密度进行控制,而密度属于非线性、大滞后变量,采用常规的控制方法难以达到满意的控制效果,所以决定采用 PID 参数自组织模糊控制器的方法来对密度进行控制^[3]。PID 参数自组织模糊控制系统结构原理如图 3 所示。

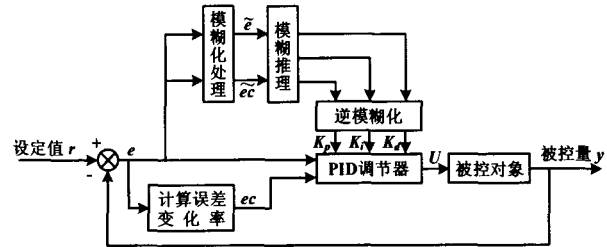


图3 系统结构原理图

Fig.3 Structure principle of system

图3中: e 为误差; ec 为误差变化率; \tilde{e} 为模糊化后的误差; \tilde{ec} 为模糊化后的误差变化率。

4 结束语

系统经过现场调试运行,结果令人满意,各项技术指标都能达到技术要求。提高了生产效率和经济效益;系统显示画面丰富、直观、操作界面简洁、使用方便。为现场操作人员创造了高效率的工作条件,实现了较为先进的过程控制和管理技术水平。

参考文献

- 1 崔坚. 西门子工业网络通信指南[M]. 北京:机械工业出版社, 2005.
- 2 于海生, 潘松峰, 于培仁, 等. 微型计算机控制技术[M]. 北京:清华大学出版社, 2003.
- 3 廉小亲. 模糊控制技术[M]. 北京:中国电力出版社, 2003.
- 4 吕俊白, 施敏芳. PLC 梯形图可视化编辑与语句表的自动生成[J]. 自动化仪表, 2005, 26(3): 28-30.

(上接第22页)

4 结束语

本文以网络视频编码器为例,介绍了DM642在网络多媒体通信应用中的一般开发过程。同时,系统所实现的网络视频编码器可直接作为网络监控系统的一部分,使用H.263算法能在低带宽上实现高质量的视频流传输,该系统模型不仅可用于基于局域网的区域监控,也可通过Internet用于跨地区的远程监控。利用

该模型还可方便地将H.264、MPEG4等视频编解码算法应用在视频通信系统中,其灵活性要远优于一些专用芯片的编解码系统,且利用DSP可以保证系统的实时性,可以将该系统模型作为网络视频监控系统的—个通用平台。

参考文献

- 1 李方慧, 王飞, 何佩琨. TMS320C6000系列DSPs原理与应用[M]. 北京:电子工业出版社, 2002: 390-410.
- 2 张广辉, 徐江华, 邵慧鹤. 基于交换式以太网的嵌入式控制系统[J]. 自动化仪表, 2003, 24(6): 5-7.