

并行 DSP 系统消息传递软件框架

王希敏¹, 王 哲², 胡 艺¹

(1 海军工程大学, 湖北 武汉 430033; 2 92664 部队指挥自动化工作站, 山东 青岛 266031)

摘 要: 设计实现了一个可复用的面向并行 DSP 应用的消息传递软件框架. 为了实现 DSP 软件的编码能够独立于不同的并行处理器体系结构, 设计分层的软件框架结构和各层中类间的关系. 为了 DSP 应用的开发能够独立于底层的数据通信路径, 提高并行系统的可扩展与可配置性, 设计基于处理器间数据流模型和硬件平台拓扑模型的节点路由表结构和路由算法. 以 TigerSHARC DSP 构成的并行 DSP 系统为例, 说明软件框架应用于特定的多处理器平台的方法, 实现并验证了消息传递软件框架.

关键词: 并行 DSP 系统; 消息传递; 软件框架; 路由表; TigerSHARC DSP

中图分类号: TP391.9

文献标识码: A

文章编号: 1000-7180(2010)10-0008-05

A Software Framework of Message Passing for Parallel Computing on Distributed DSP Systems

WANG Xi-min¹, WANG Zhe², HU Yi¹

(1 Naval University of Engineering, Wuhan 430033, China;

2 Command Automation Workstation of Unit 92664 of PLA, Qingdao 266031, China)

Abstract: This paper presents a layered message passing software framework and an efficient mechanism that decouples signal processing algorithm design from details of data communication for special-purpose parallel DSP system. We also described the structure of each layer and the interaction between layers. Inter-processor communication model and hardware platform topology model are used for the design of shortest path route algorithm for the needs of creating transferring table in which the router information is stored for each DSP node. Finally, the framework is applied to a TigerSHARC DSP multi-processors platform.

Key words: parallel DSP system; message passing; software framework; route table; TigerSHARC DSP

1 引言

并行 DSP 处理器体系结构主要有两种: 共享存储器体系结构和消息传递体系结构^[1-2]. 对于 DSP 系统, 一般有链路口和共享总线两种 DSP 间的数据传输方式, 可实现对这两种不同架构模型的支持: 通过链路口互连实现消息传递体系结构. 另外多个 DSP 通过总线连接一个容量较大的全局共享外部存储器. 这个共享存储器可以作为 DSP 间交换数据的大型缓冲区. DSP 处理器间的互联能力为构成不

同的 DSP 硬件拓扑结构提供了基础^[3-5].

消息传递和共享存储代表两种完全不同的编程模型. 消息传递模型通过显式的方式调用 API 函数进行点对点通信, 适用消息传递体系结构系统; 共享存储模型是通过读写共享存储器的方式进行数据通信. 因此, 并行软件的设计依赖于并行 DSP 处理器体系结构. 为了屏蔽这两种硬件体系结构的差异, 为上层并行应用软件提供统一的编程接口, 即将共享存储模型统一到与消息传递模型相同的显式的 API 函数调用方式, 需要底层硬件支持和上层通信软件

框架的设计与实现.

2 并行 DSP 系统消息传递软件框架

2.1 框架的分层结构与功能

DSP 系统消息传递软件框架的设计采用分层架构模式. 框架中类之间的关系如图 1 所示.

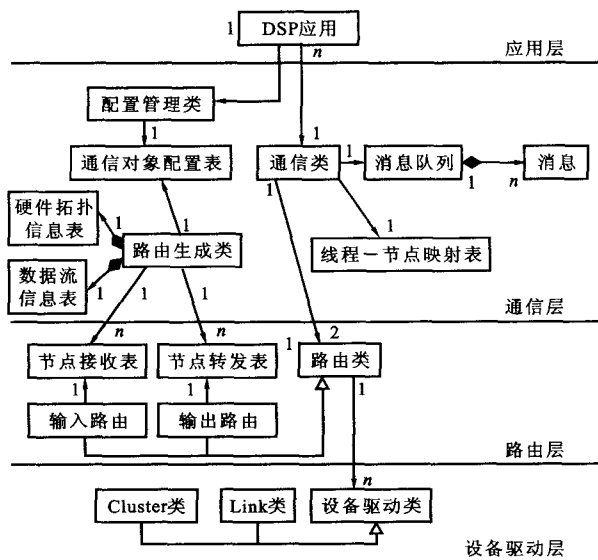


图 1 消息传递框架类图

框架主要分为三层:通信层,为上层应用提供通信服务;路由层,提供寻径和消息转发服务,为通信层交给的任务选择合适的传输路径,然后交给驱动层处理;设备驱动层为上层提供底层硬件设备的接口.设备驱动层封装了 DSP 间硬件连接的差异,并为路由层提供一个统一的接口.满足了 DSP 应用软件与两种并行 DSP 多处理器体系结构模型无关的需求.

通信层中的通信对象关联一个消息队列,用于存放具体的消息对象,通信对象通过调用与之关联的路由对象完成上层 DSP 应用交给的任务.通信类中的主要成员是消息发送与接收函数.消息发送函数的功能是将要发送的消息结构变量的值映射给路由层的消息结构变量;设置消息缓冲区标志;调用输出路由类对象的方法发送消息.消息接收函数则是调用输入路由类对象方法接收消息,并返回由消息头指针.

路由层中的主要成员是路由函数,对于输出路由对象,该函数的功能是根据路由层消息结构中的目的处理器属性值,调用路由表搜索函数查找处理器转发表;设置硬件设备对象的属性值;调用设备驱动类发送函数发送消息.路由对象通过分析消息头

信息,框架中的路由生成类创建与生成处理器路由表.对于复杂的 DSP 应用,通常需要将应用分解成多个任务,再将任务分配到各 DSP 处理器.因此任务间的数据通信路径和处理器上通信对象配置依赖于任务到处理器上的映射和处理器互连的拓扑结构.因此,路由生成类聚集了系统硬件拓扑信息和数据流信息属性.

2.2 支持时延包容的消息结构设计

对于并行 DSP 系统,实时性要求较高,为了提高性能,节省资源,采取消息头和有效数据分离的消息结构,如图 2 所示,消息结构由消息头 (MsgHead) 和有效数据 (MsgData) 两部分组成.消息头 MsgHead 用于存放有关该消息的相关信息,有效数据 MsgData 存储在数据缓冲区中.

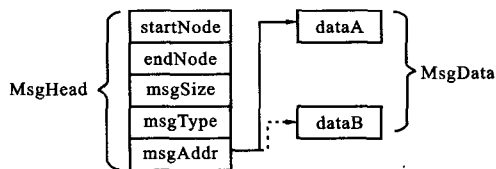


图 2 消息数据结构图

DSP 处理器间的消息传输方式有 DMA (Direct Memory Access) 传输和内核直接传输两种. DMA 方式不需要内核干预,比较适合大量数据传输的传输;在传输少量数据时,可以直接通过 DSP 内核进行访问,效率较高.为了充分利用 DSP 芯片的这种特点,实现通信与计算操作的重叠;支持数据通信时延包容,为消息缓冲区分配两个数据内存块: dataA 和 dataB. 上层应用根据缓冲区标志 buffFlag 设置消息头结构中的指针 * msgAddr 指向缓冲区 dataA 或 dataB.

3 路由表的设计与生成

路由表是每个 DSP 处理器上数据接受与发送路径的依据. DSP 系统中处理器的路由表由两个表组成,一个是处理器转发表,用于存储当前处理器发送数据到其他处理器的最短路径信息,供消息发送时查询;一个是处理器接收表,用于存储该处理器与接收其他处理器数据的最短路径信息,供消息接收时查询.

3.1 路由表结构

为了提高 DSP 资源的利用率,减少资源冗余,在并行 DSP 系统中,处理器路由表中只需存储其邻接处理器的相关信息即可.

发送消息时,该处理器的转发表中的每一条信

息只需包含当前处理器、目的处理器、下一跳处理器以及相关的链路口信息;接收消息时,该处理器的接收表中的每一条信息,需要存储当前处理器、源处理器、当前处理器的前驱处理器以及相关的链路口号.路由表可定义一个整块内存单向链表,链表中信息结构如图 3 所示.

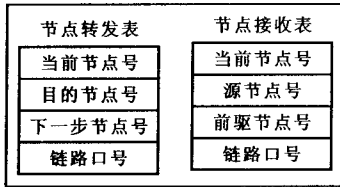


图 3 处理器路由表数据结构

3.2 路由表的生成

数据路由与硬件拓扑结构有关,而路由表所需存储的路径是与具体应用的数据流有关.路由表通过路由算法生成,一个好的路由算法不但可以为系统通信提供准确的路径信息而且还可以节省系统处理时间和存储空间,提高系统的实时性.路由算法分为两步:(1)首先由最短路径算法计算所有数据流的最短路径;(2)再分析最短路径得到数据流经过的所有处理器的路由表.

3.2.1 并行 DSP 系统硬件拓扑模型

假设已知某并行 DSP 系统有 n 个处理器, n 个处理器的硬件拓扑结构可表示为有向图 $P = (V_p, E_p, L)$, 其中, $V_p = \{v_i \mid 0 \leq i \leq n-1, n \text{ 为系统处理器数}\}$; $E_p = \{e_{ij} \mid e_{ij} = \langle v_i, v_j \rangle, v_i, v_j \in V_p\}$; $L = \{l(v_i, v_j), v_i, v_j \in V_p\}$. $l(v_i, v_j)$ 表示处理器 v_i 和 v_j 物理数据链路上 v_i 的输出端口.

在软件实现中,采用邻接表法描述硬件拓扑信息.硬件拓扑图可表示为处理器和与其关联的邻接表的关联:(处理器,邻接表),在标准 C++ 中实现中,用单向链表存储邻接表,链表中的结点表示图中与某一处理器相连的边的信息,结构可定义如下:

```
struct node_port
{
    vertex v; //相邻处理器,
    int link; //端口号
};
```

可用 STL 模板库中的 map 容器存储这些关联,处理器标识作为关键值.硬件拓扑信息可表示为

```
typedef map<vertex, list<node_port>>
Map;
```

3.2.2 数据流信息 SDF 描述

硬件拓扑图反映出系统处理器间的物理连接,

而处理器的数据流图反映的是处理器间数据通信状况,是一种逻辑连接,两个处理器之间有连接说明这两个处理器之间存在数据传输.

采用 SDF(Synchronous dataflow)图描述信号处理系统的信号处理流^[6]. SDF 图中节点表示生产或消费固定大小块的数据的 DSP 处理器,图中的边表示数据流向和数据块大小.定义 $G = (V_G, E_G, W)$ 来描述处理器间的数据流, V_G 是系统处理器集合, E_G 是数据流图的有向边的集合,即处理器间的数据流集,箭头表示数据流向. $E_G = \{e_{ij} \mid e_{ij} = \langle v_i, v_j \rangle, v_i, v_j \in V_G\}$, $W = \{w(v_i, v_j), v_i, v_j \in V_G\}$, $w(v_i, v_j)$ 是图 G 的任一边上的权值,表示固定数据块大小.

在软件实现中用关联(处理器,数据流集合)描述 SDF 图,用 map 容器存储这些关联.

3.2.3 数据流的最短路径

根据硬件拓扑和处理器间的数据流信息,计算各数据流的最短路径.基于 Dijkstra 算法,设计用 SDF 图作为算法搜索条件的最短路径算法.主要思路如下:

(1) 在算法开始时,首先对 V_p 中的源处理器 v 的数据流集合 $E_G(v)$ 进行判断,如果为空,表明 v 不和系统中任何处理器进行数据通信,结束算法,不必进行其他查找;如果不为空,继续执行;

(2) 一条路径信息为一条通路,根据硬件拓扑信息和处理器间的数据流信息运用最短路径算法求得所有数据流通路信息,包括从源点到终点经过的所有中间处理器、使用的链路口设备以及每条路径的数据流大小.

一条完整的数据路径信息暂存在链表中,该链表中的处理器存储一段链路,结构定义如下:

```
struct path
{
    int node;
    int link;
    int size;
    path * next;
};
```

其中,变量 node 表示本处理器的标识,变量 link 表示 node 的前驱到达 node 通过的链路口号. size 表示该路径上的数据流大小.

3.3 转发表和接收表生成算法

根据所有数据流的最短路径生成路由表.在实际 DSP 应用中,一个处理器并不需要知道它将消息

发送到目的处理器或从源处理器接收消息所通过的完整路径信息,而只需知道发送消息的下一跳处理器或接收消息的前驱处理器.因此,搜索所有数据路径信息暂存的链表,可得到处理器数据转发和接受所需的信息,并将这些信息分别插入到处理器转发表和接收表中.

例如 node1 到 node6 的一条通路的完整信息为: node1 -- link0 → node0 -- link2 → node4 -- link1 → node6, 遍历这条路径将被得到 3 条转发表信息:

- (1) node1 -- link0 → node0 → node6;
- (2) node0 -- link2 → node4 → node6;
- (3) node4 -- link1 → node6;

3.4 处理器路由表查找算法

程序运行时,以目的处理器为关键值搜索转发表,以源处理器为关键值搜索接收表,即可找到发送或接收数据的相关路径信息,完成处理器的数据传送.

4 软件框架应用于 ADSPTS101 DSP 平台

将框架应用于由多片 ADSP-TS101 DSP 组成的通用信号处理平台^[7].硬件结构如图 4 所示. ADSP-TS101 DSP 处理器有两种类型的多处理器间数据接口: Link 口和簇总线. ADSP-TS101S 芯片的簇总线提供了传输速率高达 800Mbyte/s 的多处理器间的数据通道.可通过共享总线方式构成多处理器. Link 口传输速率高达 250Mbyte/s、支持点对点通信.根据应用系统的需求,可通过 Link 口和簇总线,灵活地构成各种拓扑结构的多处理器系统.可以通过多处理器空间互相访问通过簇总线互联的 TigerSHARC DSP 片内存储.这种处理器间的互访机制对于统一两种硬件体系结构,为上层并行应用软件提供统一的编程接口提供了硬件基础.

另外多个 DSP 通过局部总线连接一个容量较大的全局共享外部存储器 SDRAM. 这个共享存储器可以作为 DSP 间数据交换的大型缓冲区.在与主机通信方面,所有 DSP 以及外部共享全局存储器都通过共享的局部总线及 PCI 接口控制器连接到 PCI 总线.这样主机可以直接访问任何一个 DSP 处理器的内部存储器及外部共享全局存储器.主机通过 PCI 设备驱动函数库访问 DSP 内部存储器及片外 SDRAM.设备驱动程序是基于 WinDriver 开发实现.

软件可分为两部分: DSP 程序和主机程序.在

AD 公司提供的集成开发环境 VisualDSP++ 3.5 下,用标准 C++ 编程语言设计完成软件框架与应用实例.主机的操作系统采用 WindowsXP,主机软件开发环境是 VC++ 6.0.在主机上运行路由算法程序,产生处理器的转发表和接收表,并通过广播将这些信息写到 DSP 板的各个处理器片内存储.

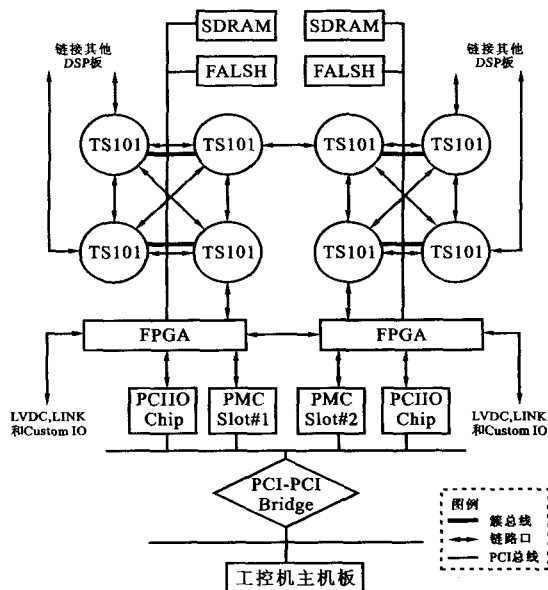


图 4 并行 DSP 系统硬件结构图

4.1 设备驱动层数据收发的同步

将软件框架的应用于特定的处理器平台,主要的问题是设备驱动类的实现以及数据收发的同步.

ADSP-TS101 有 4 个物理 Link 口,因此可定义 4 个 Link 口类对象 Link0, Link1, Link2, Link3. 簇总线与 Link 口设备驱动类的实现有所不同,为了实现数据传输,收发双方需要在所有处理器片内或共享存储器的固定地址区域定义多个消息头缓冲区和同步标志.消息头和同步标志是由发送方写,接收方读.消息的传递都采取非阻塞的同步方式.

为了可靠地进行消息的收发,在设备驱动层的收发程序中实现同步协议.即由接收方控制数据的收发,数据包括消息头和有效数据两部分.同步协议流程如图 5 所示.其中的有效数据发送与接收采取 DMA 传送方式,同步标志和消息头发送与接收采取内核写方式.

4.2 应用实例

以 Link 口方式传输为例给出消息传递框架的验证实例.图 6 给出应用软件框架实现多个 DSP 节点间进行消息传递的示意图, DSP1 向 DSP5 传输消息需要经过 DSP0 和 DSP4 的转发,图 7 中给出

DSP1 要传输的消息头和有效数据,DSP5 完整无误地接收到了从 DSP1 传输进来的数据,该框架在该通用信号处理机中成功运行。

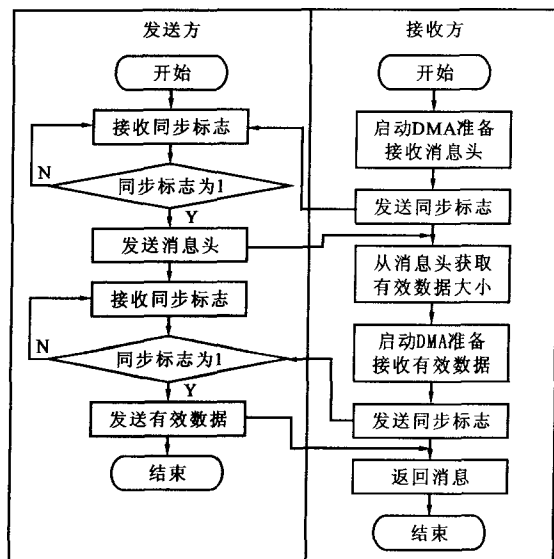


图5 收发双方同步关系图

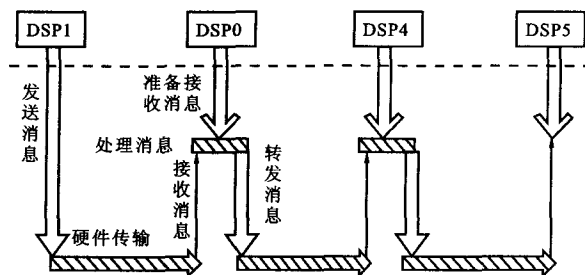


图6 DSP1和DSP5间点消息传递示意图

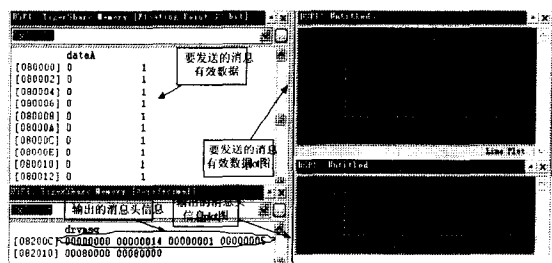


图7 DSP1发送的数据

5 结束语

为了解决DSP软件的移植性问题,设计实现了可复用的并行DSP系统消息传递软件框架.对于研究开放的电子信息系统,设计独立于平台的、可复用的嵌入式实时DSP应用软件,具有实际的应用价值。

参考文献:

- [1] 朱彦伊,徐胜,魏宗坤,等.基于DSP的多传感器数字视频监控[J].微电子学与计算机,2008,25(11):202-205.
- [2] 姚仲欢,王高才,孙野,等.传感器网络中基于最短路径树的低延时节能路由算法[J].微电子学与计算机,2007,24(9):174-180.
- [3] Agbaria A, Kang Dong-in, Singh K. LMPI: MPI for heterogeneous embedded distributed systems [C]//Proceedings of the 12th International Conference on Parallel and Distributed Systems (ICPADS'06). Minnesota, 2006.
- [4] Kohout J, George A D. A high-performance communication service for parallel computing on distributed DSP systems[J]. Parallel Computing, 2003, 29(7): 851-878.
- [5] Saha S, Bhattacharyya S S, Wolf W. A communication interface for multiprocessor signal processing systems[C]//Proceedings of the 2006 IEEE/ACM/IFIP Workshop on Embedded Systems for Real Time Multimedia. Seoul, 2006:127-132.
- [6] Wayne Wolf. 高性能嵌入式计算[M]. 北京:机械工业出版社,2007:200-206/361-370.
- [7] Analog Devices Inc. ADSPTS101 tiger SHARC processor hardware reference [Z]. America, 2004.

作者简介:

王希敏 女,(1963-),副教授.研究方向为军用仿真系统体系结构.

王哲 女,(1981-),助理工程师.研究方向为DSP应用软件设计.

胡艺 男,(1984-),硕士研究生.研究方向为软件工程与中间件技术.