

基于SMP的总线式软运动控制引擎设计*

孙好春 张承瑞 胡天亮

(山东大学机械工程学院, 济南 250061)

摘要: 提出了基于SMP实时架构的EtherMAC实时以太网网站的设计方法,开发了一种运行于内核空间的运动控制引擎,并给出了一种通过在共享内存建立内存堆栈方式的跨平台编程接口设计方法。通过设计总线抽象层将总线控制资源抽象,降低了运动控制引擎与总线之间耦合性,增强了总线系统或从站设备的可替换性。原型机进行功能验证表明,该引擎实现了较好的易用性和可移植性,并间接增强了EtherMAC的实时性能。

关键词: 实时以太网 软件运动控制 运动控制引擎 实时拓展

中图分类号: TP393 **文献标识码:** A **文章编号:** 1000-1298(2014)03-0288-05

引言

为了满足高性能的多轴运动控制系统对总线数据传输的高速率需求,运动控制总线的通信载体逐渐由传统的低速串行通信接口向高速的以太网过渡。由于普通以太网的通信架构的实时性能不能满足运动控制系统的要求,因此需要对以太网通信结构的链路层进行修改以增强以太网总线的实时性能^[1-3]。EtherMAC (Ethernet for manufacture automation control) 是山东大学数控技术研究中心研发的面向运动控制领域的实时以太网总线,其从站周期定时机制和主从站双缓存的方法,虽然降低了对主站定时精度的要求,并可以实现高精度插补^[3-4],但是缓存机制会造成计算延时,不能满足实时插补、全闭环运算等功能要求。因此,本文提出基于SMP (Symmetric multiprocessing) 实时架构的EtherMAC主站设计方法,以增强主站的实时运算性能。基于该结构开发的运动控制程序必须运行在内核空间才能保证实时性,但是内核空间的程序开发需要开发者具有较高的编程能力,因此内核空间编程不适宜作为普通控制程序开发环境。为了提高该平台控制程序开发效率,本文在内核空间设计一种面向总线抽象层的全软件运动控制引擎,并通过在共享内存建立用于传递参数的内存堆栈的方式设计跨平台编程接口。总线抽象层将总线控制资源抽象,并向运动控制引擎提供统一的控制资源供其使用,从而降低运动控制引擎与总线的耦合性,提高软

件的复用性和移植性。

1 总线主站的SMP实时平台设计

系统的实时性要求系统能够对外来事件在限定时间内做出反应^[5]。SMP架构的多核平台不仅提升了系统的运算性能,而且也可实现更好的系统实时性能^[6-9]。

1.1 EtherMAC性能瓶颈分析

一般总线系统采取主站管理总线的定时和同步机制,而EtherMAC则是由从站的嵌入式硬件实现,这种机制降低了对主站定时精度的要求,并且可以实现更加稳定精确的定时和同步精度^[3]。但是要求主站对网口外部中断信号的系统响应时间小于通信周期,如果要想实现200 μs 的通信周期,则此系统响应时间必须小于200 μs 。最小通信周期是总线实时性能的主要指标之一,目前EtherMAC最小通信周期仅达到1 ms。因此只有进一步减小主站对网口中断的系统响应时间,才能实现更小的EtherMAC最小通信周期。

1.2 SMP实时架构的原理

目前多核计算机已经在工业领域普及,出于性能因素的考虑,大部分多核计算机多处理器的硬件架构均为SMP架构^[9]。该架构中每个物理或逻辑CPU可以平等地使用相同的硬件资源,因此每一个CPU都可以独立的运行,并可以在不同的CPU并行运行独立的操作系统。

在计算机上并行运行RTOS (Real-time operating

收稿日期: 2013-09-13 修回日期: 2013-11-28

* 国家科技重大专项资助项目(2012ZX04010-021-002-005)

作者简介: 孙好春, 博士生, 主要从事运动控制技术研究, E-mail: sunvhao@163.com

通讯作者: 张承瑞, 教授, 博士生导师, 主要从事机电控制基础理论和先进制造研究, E-mail: zhangchengrui@gmail.com

system)与GPOS(General purpose operating system),通过功能互补可使系统既具有良好的实时性能又兼有丰富的软件开发资源支持,这也是改进GPOS实时性能的主要方法之一^[10]。在单CPU架构下,由于GPOS与RTOS共用同一个CPU,有时会因为争夺计算资源互相影响,从而导致系统实时性能下降^[11],如图1a所示。而SMP架构可以让GPOS与RTOS各自运行于不同CPU,且其他硬件资源也可以配置为其中一个操作系统专用,因此在运行时互补干扰,可以实现稳定且更强的实时性能^[12-15],如图1b所示。

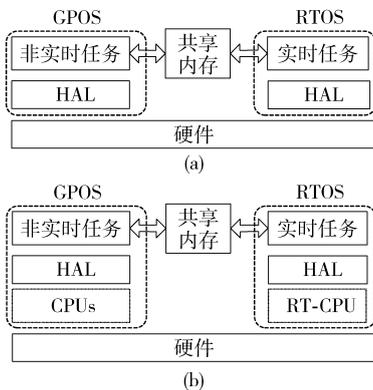


图1 SMP实时架构原理图

Fig.1 SMP real-time architecture

1.3 Kithara RTS

Kithara RTS(Real-time Suite)是德国Kithara软件公司出品的一款Windows实时拓展软件。其原理是通过在Windows外并行运行子实时系统Kithara RTS-Kernel增强系统的实时性能,并且在SMP架构下使能Dedicated模块后Kithara RTS将独占一个CPU核心。Kithara RTS具有丰富的功能模块,其中Packet模块可以直接控制网卡在数据链路层收发数据,无需开发或修改网卡驱动,非常适合作为实时以太网开发平台。

2 主站的系统响应性能测试

系统响应时间是系统发出处理要求到系统给出应答信号的时间,是衡量系统实时性能的一个综合指标,其数值越小则系统实时性能越好^[4]。这里的系统响应时间是指对以太网卡中断信号的响应时间。通过在未使用和使用Kithara RTS的Dedicated模块两种模式下进行对比测试,验证SMP实时架构的系统响应性能。

2.1 测试方法设计

由于PC自身的时间测量精度较低,仅能达到 μs 级,因此本文采用外部辅助硬件记录时间的方法进行测试^[16]。测试方法原理如图2所示,首先PC

通过以太网对外部硬件测试装置进行初始化,然后测试装置按周期时间 T 向PC的网口上发数据包,PC在接收到数据时立刻下发数据包。上发数据包内容包括测试装置计算的相关的发送与接收之间的时间差 t_c ,PC在每个周期从接收的数据中解析并保存 t_c 。由于网络传输时间微小且稳定,因此 t_c 可客观反映系统响应时间 t_{sa} 的大小。EtherMAC的从站定时机制即为外部硬件定时,可通过修改从站的FPGA程序实现系统响应时间的测试。

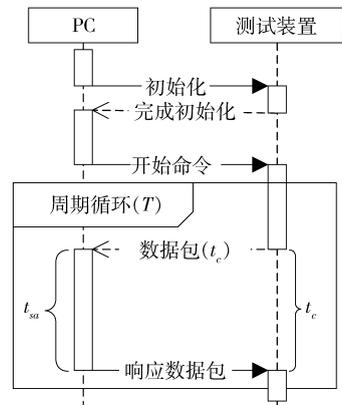


图2 系统响应时间测试方法示意图

Fig.2 Method for measuring response time

2.2 测试数据分析

用于测试的PC选用Intel多核处理器i3-2100,内存容量为2GB,网卡型号为RTL8168C,整个测试环境在Windows7操作系统上搭建。关闭Intel处理器特有的超线程技术,构建一个标准的SMP多核平台,设置 T 为1ms。分别在两种模式下各测试10次,每次的测试时间1h,同时开启Prime95计算机性能测试软件使CPU和内存使用率保持在100%。

每次测试中仅记录响应时间最大值^[10],在未使用Dedicated模块时,测得的时间波动较大,最大值接近 $145\mu\text{s}$;而使用Dedicated模块后,测得的时间最大值波动较小,最大值不超过 $14\mu\text{s}$ 。

由此可知,相比于普通Windows系统,在共用CPU的情况下增加RTOS后的系统响应性能得到了很大的提升,而使用SMP实时结构后系统(EtherMAC主站)的系统响应性能可以得到更大的提升,因此采用SMP实时架构的主站时,EtherMAC总线系统的最小通信周期可以达到 $100\mu\text{s}$ 。

3 运动控制引擎设计

SMP实时架构中并行运行了Windows和Kithara RTS两个操作系统,而运动控制程序为了保证运行的实时性能则必须运行于Kithara RTS的内核空间。但是内核层的程序编码要求较高,开发调试过程常伴随系统崩溃,不适宜在内核空间直接开

发控制程序。因此从易用性考虑,在内核空间设计一个运动控制引擎,并提供跨平台编程接口,只需在 Windows 下调用此编程接口即可完成控制程序开发。

3.1 软件方案设计

运动控制引擎的软件方案如图 3 所示。由于该引擎为全软件架构,因此运行于 Kithara 内核空间的运动控制引擎可以静态复用 PC 原有的软件功能模块,如插补模块、PLC 模块等。控制程序则可复用 Windows 平台下丰富的程序功能模块,如 HMI 模块、代码解析模块、配置模块和信息集成模块等。

由函数调用机制可知,函数参数是通过内存堆栈传入函数中,而并行运行的多个操作系统内存管理是相互独立的,因此 Windows 程序无法直接调用 Kithara RTS 的程序接口,只能通过共享物理内存的方式进行数据交互,因此需要设计一个 API 层使 Windows 程序可以直接调用 Kithara RTS 程序接口。本文提出的跨平台 API 层设计原理如下:首先将内存堆栈设置在共享内存区域,先由 Windows 程序接口函数把调用的函数参数写入内存堆栈,然后触发 Kithara RTS 内核空间的程序函数接口从共享内存的内存堆栈中取出函数参数进行函数运算,将函数返回结果也存入此内存堆栈中,由 Windows 程序接口函数取回函数返回结果。

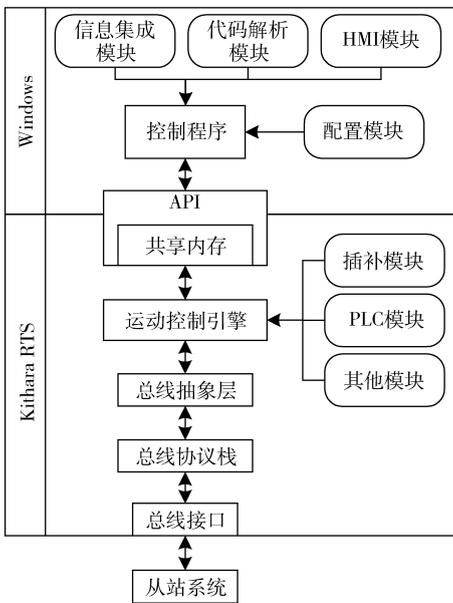


图 3 软件方案结构图

Fig.3 Structure of software design

在计算机体系结构中,硬件抽象层能让操作系统适应不同结构的硬件系统,以增加操作系统的通用性。本文借鉴硬件抽象层的思想,在运动控制引擎和总线协议栈之间增加了一个总线抽象层。总线抽象层的结构如图 4 所示,其将接收的总线数据转

换成状态数据供运动控制引擎周期运算使用,同时将运动控制引擎下发的控制数据转化为各节点数据,然后下发到总线中。总线抽象层降低运动控制引擎和总线之间的耦合性,不用修改总线抽象层的上层软件即可适应新的总线系统或从站设备。例如用一个可以控制 3 个电动机的从站替换原有的可控制两个电动机的从站,虽然功能有冗余但仍然可以满足功能要求,仅需要修改总线配置文件即可,总线抽象层会把新的总线从站系统抽象,运动控制引擎检测到可控制资源没有减少,则控制程序仍然可以正常运行。此外,也可只更改总线抽象层的配置文件使其适应不同类型的总线系统。

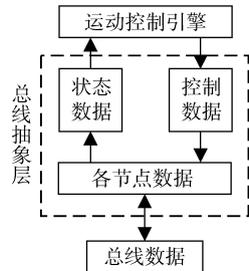


图 4 总线抽象层功能图

Fig.4 Function diagram of bus abstraction layer

3.2 软件流程设计

运动控制引擎的运算周期等于总线通信周期,即每次总线数据更新时进行一次任务运算。在每个运算周期内需要完成的任务包括处理外部命令、进行任务处理、接收和下发控制数据等。

每个周期解析控制命令和执行任务的机制如图 5 所示。首先从总线抽象层读取状态数据作为本周期运算的主要数据依据。然后解析控制命令表中的外部命令,修改任务表中的运动控制任务。任务表以 MotionProc 抽象类作为任务加载接口,将具体的运动控制任务类实例化加载到任务表中,如直线插补、圆弧插补和曲线插补等任务。在引擎逻辑运算之后判断任务表中任务的可执行状况,如果没有可执行任务则本周期结束;如果有则执行任务表中的运动控制任务然后更新控制数据。

4 原型机验证

以 EtherMAC 实时以太网作为运动控制总线,设计了一种原型机以验证本文设计的运动控制引擎的功能可行性。

选用两种从站节点模块以级联的方式组成从站系统连接 PC,一种为可控制 1 个伺服电动机控制模块,另一种为具有 8 个开关量 I/O 的控制模块。在 Windows 下只使用开发效率高但程序执行效率较低的 CLR 高级编程语言 C#开发控制程序,通过互操

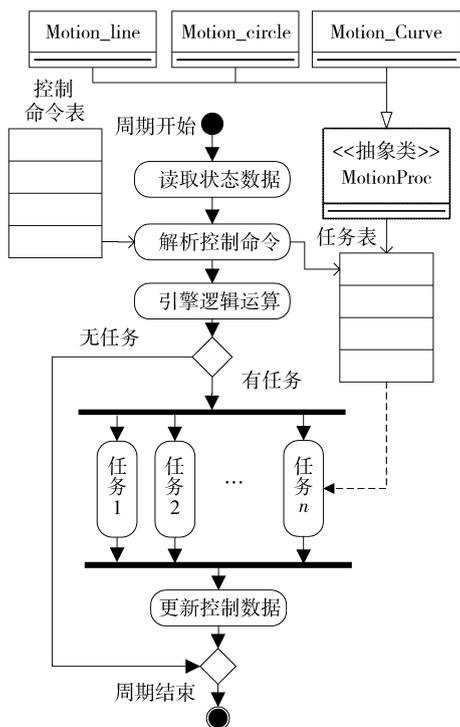


图 5 运动控制引擎任务处理过程图

Fig. 5 Activity diagram of task processing

作技术调用非 CLR 的运动控制引擎的编程接口。使能 Kithara RTS 的 Dedicated 模块, 设置通信周期为 $100 \mu\text{s}$, 实际运行效果如图 6 所示。结果发现电动机稳定到达指定位置, 并没有出现数据丢失现象, 表明 EtherMAC 总线可以实现 $100 \mu\text{s}$ 的通信周期, 而只使用 C# 开发控制程序的方式表明该引擎具有较好的易用性。

然后采用一个可控制 3 个电动机并带有 16 个开关量 I/O 的控制模块, 代替前面的 2 个从站节点,

可见控制资源并没有减少, 然后修改总线配置文件, 控制程序仍可以正常工作。表明总线抽象层的增加有效增强了设备之间的替换性。



图 6 原型机实物图

Fig. 6 Image of prototype

5 结束语

SMP 实时架构的主站增强了 EtherMAC 实时以太网总线的实时性能, 并基于此架构设计了一种面向总线抽象层的全软件运动控制引擎, 其跨平台编程接口简化了控制程序开发过程, 降低了控制程序的开发难度, 实现了较好的易用性。而面向总线抽象层的设计方法降低了运动控制引擎与总线之间耦合性, 增强了总线系统之间或从站设备之间的可替换性, 使系统实现了较好的复用性和移植性。改进后的 EtherMAC 与该运动控制引擎结合, 可以组成一个具有良好开发环境的高速精运动控制开发平台, 具有较好的应用价值。

参 考 文 献

- 1 Felser M. Real-time ethernet-industry prospective[J]. Proceedings of the IEEE, 2005, 93(6): 1118 - 1129.
- 2 Lian F L, Moyne J, Tilbury D. Network design consideration for distributed control systems[J]. IEEE Transactions on Control Systems Technology, 2002, 10(2): 297 - 307.
- 3 姬帅, 胡天亮, 张承瑞. 基于实时以太网的高速精插补算法及其实现[J]. 计算机集成制造系统, 2012, 10(10): 2211 - 2216. Ji Shuai, Hu Tianliang, Zhang Chengrui. High speed fine interpolation algorithm and its implement based on real-time Ethernet [J]. Computer Integrated Manufacturing Systems, 2012, 10 (10): 2211 - 2216. (in Chinese)
- 4 Wang Ke, Zhang Chengrui, Xu Xun, et al. A CNC system based on real-time Ethernet and Windows NT [J]. The International Journal of Advanced Manufacturing Technology, 2013, 65(9 - 12): 1383 - 1395.
- 5 罗蕾. 嵌入式实时操作系统及应用开发[M]. 3 版. 北京: 北京航空航天大学出版社, 2011.
- 6 McKenney P. SMP and embedded real-time[J]. Linux Journal, 2007(153): 52 - 57.
- 7 Bruzzone G, Caccia M, Ravera G, et al. Standard Linux for embedded real-time robotics and manufacturing control systems[J]. Robotics and Computer-Integrated Manufacturing, 2009, 25(1): 178 - 190.
- 8 周凯. 基于实时以太网的高性能多轴数控系统[J]. 制造技术与机床, 2011(3): 157 - 161. Zhou Kai. High-performance multi-axis NC system based on real-time ethernet[J]. Manufacturing Technology & Machine Tool, 2011(3): 157 - 161. (in Chinese)
- 9 Dual core processors lower system costs for embedded real-time applications [OL]. http://download.intel.com/technology/advanced_comm/TenasysforLightReading.pdf
- 10 Koren Y, Pasek Z J, Ulsoy A G, et al. Real-time open control architectures and system performance [J]. CIRP Annals-

- Manufacturing Technology, 1996, 45(1): 377 - 380.
- 11 陈宗雨,郭伟,王立峰. 基于 Windows NT 与实时扩展的开放式数控系统的研究[J]. 计算机集成制造系统,2006,12(4): 568 - 572,640.
Chen Zongyu, Guo Wei, Wang Lifeng, et al. Research on open CNC system based on Windows NT & RTX[J]. Computer Integrated Manufacturing Systems, 2006,12(4):568 - 572,640. (in Chinese)
- 12 徐跃,王太勇,赵艳菊. 基于现场总线的开放式网络化数控系统的设计[J]. 天津大学学报,2008, 41(6): 678 - 683.
Xu Yue, Wang Taiyong, Zhao Yanju. Design of open and networked CNC system based on fieldbus[J]. Journal of Tianjin University, 2008, 41(6): 678 - 683. (in Chinese)
- 13 周凯. PC 数控原理系统及应用[M]. 北京:机械工业出版社,2007.
- 14 毛军红,李黎川,吴序堂. 机床数控软件化结构体系[J]. 机械工程学报, 2000, 36(7): 48 - 51.
Mao Hongjun, Li Lichuan, Wu Xutang. Hardware independent architecture for CNC machine tools reconfiguration[J]. Chinese Journal of Mechanical Engineering, 2000,36(7): 48 - 51. (in Chinese)
- 15 李攀,裴海龙. 基于 Soft DMC 圆弧插补器的研究和设计[J]. 机械设计与制造,2011(4):37 - 39.
Li Pan,Pei Hailong. Arc interpolator research and design based on Soft DMC[J]. Machinery Design & Manufacture, 2011(4): 37 - 39. (in Chinese)
- 16 赵立业,张激,游夏. 实时操作系统的性能分析和评估[J]. 计算机工程, 2008, 34(8): 283 - 285.
Zhao Liye, Zhang Ji, You Xia. Analysis and evaluation of real-time operating system performance[J]. Computer Engineering, 2008, 34(8): 283 - 285. (in Chinese)

Design of Bus-style Soft Motion Control Engine Based on SMP

Sun Haochun Zhang Chengrui Hu Tianliang

(School of Mechanical Engineering, Shandong University, Ji'nan 250061, China)

Abstract: The design method of real-time Ethernet EtherMAC master based on SMP architecture was proposed, while a field bus oriented motion control engine was developed, which was running in kernel space. To make the procedure of motion control program development easy, a cross-platform programming interface was created by making memory stack in a shared memory for transferring parameters. To reduce the coupling between the motion control engine and the field bus, and to enhance the adaptability of the engine to any other field bus, a bus abstraction layer was proposed and designed. The result from prototype shows that the motion control engine has a better ease of use and a better portability, and indirectly enhances the EtherMAC real-time performance.

Key words: Real-time Ethernet Soft motion control Motion control engine Real-time extension