

# 一种用于机器人力反馈依从控制的 计算机体系结构

邹逢兴

(自动控制系)

**摘要** 本文提出了一种用于机器人力反馈依从 (Compliance) 控制的计算机体系结构——MIMD型的多微机并行处理系统,并从硬件系统结构和系统软件两方面予以了说明。该系统是为完成一项实际科研任务而设计的,它能有效地实现机器人力/位置混合控制中多任务的并行处理,它的实现将使伺服控制周期缩短到5ms以下,能满足各种机器人控制任务的需要。

**关键词** 机器人计算机系统, 并行处理/力反馈控制, 计算机体系结构, 多微机系统

**分类号** TP302, TP242

力反馈依从控制技术和力控制器是研究和开发智能机器人的关键性基础技术之一。由于机器人力反馈依从控制是以操作空间法为基础的力/位置混合控制,实现它比实现以关节级的PID算法为基础的纯位置控制要复杂得多,计算量要大得多,需对控制计算机的计算速度和存储容量等性能提出更高的要求。为此,近几年来许多国家的专家们都在结合力/位置混合控制器的研究,对机器人控制的计算机体系结构进行探索和研究。

实践证明,采用目前的单处理机系统来实现机器人力反馈依从控制是肯定不行的,即使采用16位以至32位的超级小型机也无法满足它的计算需要;必须采用多机并行处理体系结构,实现机器人力反馈依从控制中多重任务的并行处理和控制,才能满足要求。

从目前所见,多机并行处理系统用于机器人控制方面,主要有三种结构型式,即流水线式结构,并行处理机结构和多处理机结构。但综合起来看,还是以采用多指令流多数据流(MIMD)的多处理机结构更合适、更普遍;而在机器人控制的多处理机结构中,

1989年10月7日收稿

operating system function supporting the design of the distributed library function, and some problems which have to be emphasized in the design process are described. A simple distributed program example is also given.

**Key words** programming, distributed computer, robot/distributed library function, distributed concurrent processes, primitives

采用的又大都是功能分布式的并行处理，各处理机与机器人各关节或各部分之间具有固定的对应关系，因而各处理机具有固定的计算任务。我们结合科研任务的需要，突破这种传统方式，设计了一种由 OS 集中、统一管理，划分负载动态平衡的多微机并行处理系统，用于实现机器人力/位置的混合控制，可望将机器人的伺服控制速率提高到 200Hz（即周期为 5ms）以上，从而提高机器人控制的实时性、精确性和稳定性。本文将从硬件和软件两方面对该系统的体系结构予以说明。

## 1 硬件系统结构

我们设计的力反馈依从控制器的计算机系统结构如图 1 所示。这是一种以实现任务级并行处理为基础的总线式同构型 MIMD 紧耦合多处理机结构。其中：

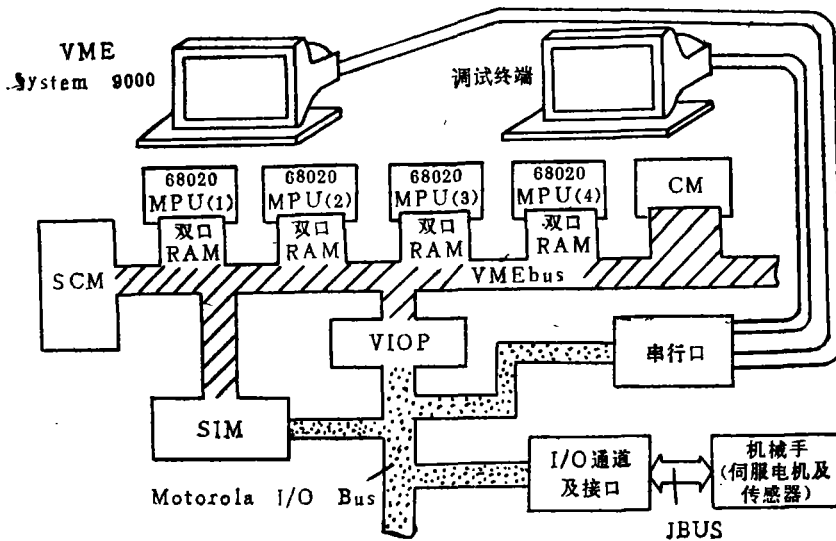


图 1 系统结构

VME System 9000 是 32 位超级微机系统，作为整个机器人控制系统的前端机，主要用于程序加载、人机交互、文件 I/O 服务和提供程序开发环境等，其主 CPU 为 MC 68020。

挂在 VMEbus 上的四块 68020 MPU 模块、一块 CM 模块、一块 I/O 处理机模块 VIOP 和系统控制模块 SCM、系统中断模块 SIM，构成多机并行处理系统，是真正实现力/位置伺服控制的后台处理机。各 68020 模块作为并行处理单元，CM 用作共享存储器，VIOP 负责与机械手、前端机、调试终端等有关的 I/O 处理。多机并行处理系统具有统一的多机并发操作系统，各 68020 模块与机械手各关节并没有固定的对应关系，也没有固定的计算任务，各关节腕力传感器参数和关节位置参数被采集后，由多机操作系统根据力反馈依从控制数学模型的内在并行性和各处理机的忙/闲、良/故情况，统一调度和管理，统一分配处理单元进行并行处理，并根据处理结果统一发出各关节的伺服控制信号。

为了尽量减化硬件设计，提高可靠性，我们尽量选用能满足要求的，同时又是标准

的、在市场上能买到的VMEbus模块和器件来组建系统，而避免设计一套全新的硬件。如我们选用了美国VME技术公司制造的VMPU—32模块作为并行处理单元，上面有带指令cache的68020MPU，68881协处理器，256KB EPROM，1MB双端口RAM，以及存储管理器MMU等；选用了该公司的VIOP模块作为I/O处理器，它以MC68000为CPU，上面有64KB EPROM，512KB双端口RAM，还有VMEbus接口和I/O总线接口，以及两种总线间的适配逻辑等。我们自己只设计了市场上难于合适买到的系统控制逻辑、系统中断逻辑和I/O通道。采用这种技术路线设计出来的系统，虽然在某些方面可能不很理想，但总的说来能满足我们需要，特别对于基础研究性实验，甚至不失为一种较好的途径。

### 1.1 存储器组织结构

该多机并行处理系统的存储器组织，采用的是既有共享存储器CM，又有本地存储器LM的邮箱共享存储结构。LM存储工作程序和专用数据、局部变量，而CM只存储公共变量、有关表格和进行数据交换，起邮箱作用。工作时基本遵循就地处理原则，只在少数情况下才需访问CM。这样，系统几乎可以处于一种高度的模块“自治”并行处理之中，可以尽量减少通信量，降低对CM的争用，提高处理速度。这种存储器结构如图2所示。

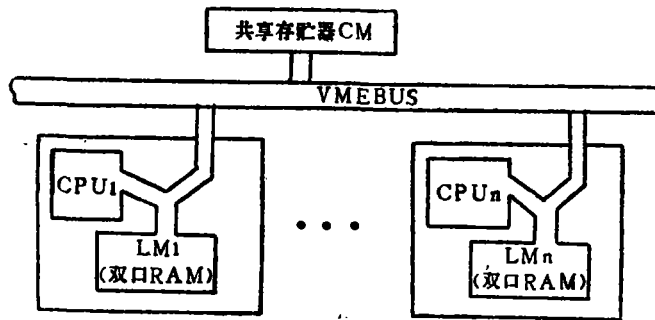


图2 存储器结构

### 1.2 系统控制器

系统控制器提供如下功能：

- 16MHZ系统时钟；
- 处理总线仲裁和总线定时输出；
- 提供上电、掉电和本地或远程的复位功能；
- 指示总线状态。

其中核心功能是总线仲裁。它的主要任务是有效地分解总线冲突，即在有多个主设备同时请求控制总线时，将总线控制权按一定算法分配给具有最高优先权的主设备。根据多机并行处理系统的工作特点和VME总线的规范，我们设计的总线仲裁算法的构成如图3所示。可以看出，这是一种将独立请求法（又称并行法）和串链法相结合的二维总线判优机构。各主设备对总线控制权的请求首先经由图中左框内的独立请求机构，由并行总线仲裁器按循环选择算法，决定与BR<sub>0</sub>~BR<sub>3</sub>中哪根请求线相连的主设备获得总线控制权，然后再用串链法来决定BR<sub>3</sub>上的设备是68020MPU(1)还是VIOP应该获得总线控制权。其中独立请求机构采用循环选择算法，可以使各并行处理单元地位平等，占用总线机会均等，避免发生某个主设备独占总线的现象。采用这种二维判优系统，既可在VMEbus规范的前提下满足要求，又有较好的灵活性，便于系统扩充。

我们将总线控制器的全部逻辑设计成一块专门的VMEbus模块，插在总线底板上的第一槽。

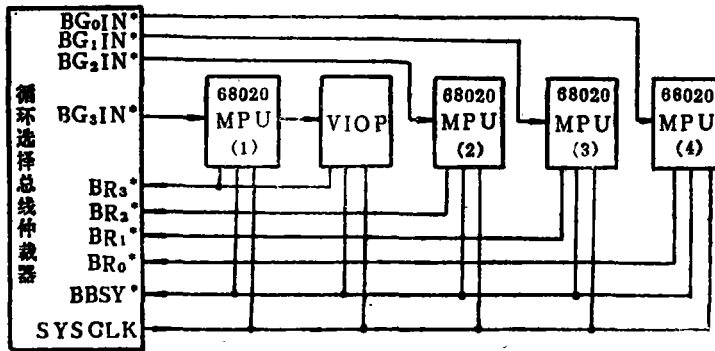


图3 总线仲裁器的构成

### 1.3 系统中断逻辑

为了保证机器人控制多处理机系统的正常运转，协调系统内部各部分之间的工作及其与机械手、操作员的关系，系统设置了一个中断子系统。所有主设备(包括68020MPU和VIOP)间的通讯均利用这些板上提供的邮箱中断，通过软件来实现。所有外部中断，如I/O通道中断、各种外部信号中断等，采用按优先顺序排队的分布式多级向量中断结构，使用VMEbus或I/O BUS中的中断线，且自己设计相应的中断请求逻辑，与各主设备板上的中断处理逻辑相结合来实现。中断子系统的多级优先向量中断结构如图4所示。

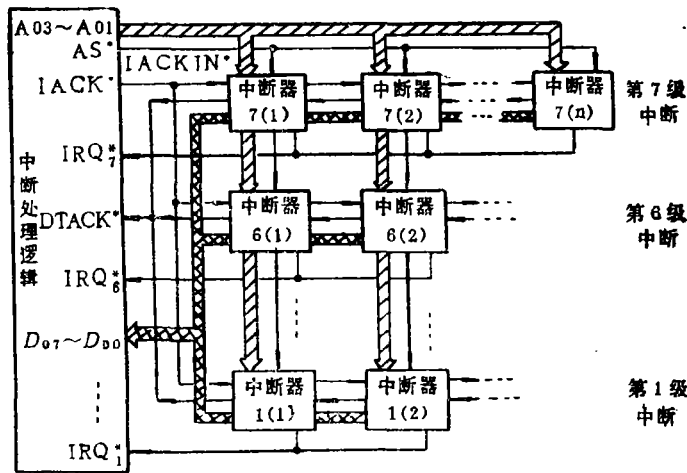


图4 中断结构

### 1.4 I/O接口子系统

在多机并行处理系统与机器人之间，我们配置了“智能I/O子系统”，目的是使并行处理节点机能摆脱繁重的I/O处理任务，以便高效地集中执行其预定的内部程序和数据处理等主要任务，而I/O数据的采集、预处理和传送控制等任务则专门由智能I/O子系

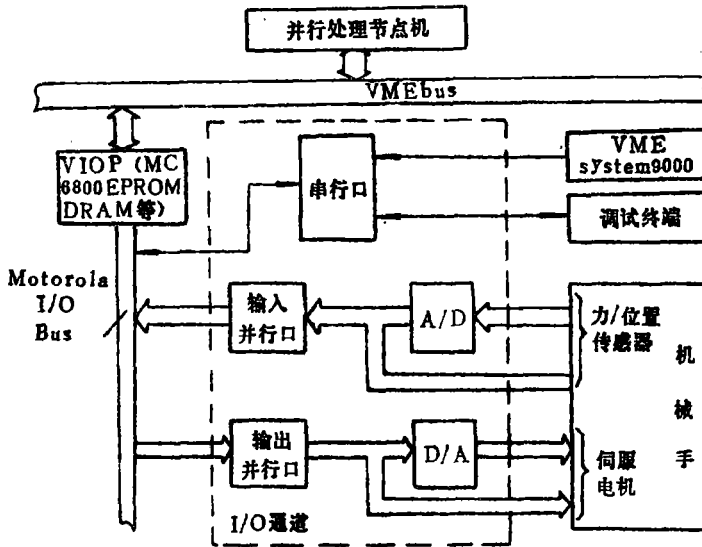


图5 I/O子系统结构

统承担。这样，IOP操作和并行处理节点机的操作是并发进行，有利于提高整个工作效率和处理速度。I/O子系统结构如图5所示，其中虚线框内的I/O通道由自己设计。

## 2 系统软件

我们的多微处理机系统的系统软件主要包括引导程序、多机并发操作系统和简单动作级机器人语言等部分。

### 2.1 引导程序

引导程序主要完成多机系统的测试诊断和自举。引导程序固化在各处理机本地存储器的EPROM中。系统在上电，或者按本地或远程“RESET”按钮时被复位，复位使所有处理机进入事先规定好的状态，然后每一处理机开始执行在它们的EPROM中从某一固定地址开始的引导程序。其中的自举程序使各模块初始化，然后等待用户指定的主处理机的进一步命令。在初始化期间，主处理机是控制机，它负责完成多处理机系统自身局部存储器测试，再对系统中其它模块一一进行测试。先测这些模块的存在性，再测试其复位状态，看其是否初始化，以及测试它们是否有失效中断请求等。初始化之后，装入程序便从主处理机开始执行，然后主处理机根据并行处理的需要，启动各处理机投入运行，共同完成装入程序的并行处理。

### 2.2 多机并发操作系统

作为一个机器人控制的专用多处理机系统，我们设计多机操作系统的目标以能支持单道作业、多个任务的并行处理为原则。严格地说，它只是多机操作系统的内核，它主要提供对下列功能的支持：实现协同进程的同步与通讯；实现并发进程对临界区的互斥访问，保护共享数据的完整性；实现并发任务的派生和会合；完成中断处理；实现对多处理机和共享存储器的管理；提供必要的故障探测和容错功能。

为了提高系统的运行效率和对用户的透明性，操作系统设计时我们采用了如图6所

示的层次结构和模块化结构，并且规定外层模块可以调用内层模块，反之则不允许，这样可以避免循环的相互依赖关系，利于软件的调试和维护。

从机器人控制的实时性约束和易于设计这两者的折衷考虑出发，我们在OS设计时，采用C语言与汇编语言相结合的过程编码，同时按响应时间的要求设计了访问机构和数据结构。

多机操作系统内核的功能，采用分布式执行原则，即每个处理机上都驻留一个OS内核副本，内核中包含一个用于机间通讯的模块VMPUCOM，该模块提供两条原语：P—SEND（发送进程）和P—RECEIVE（接受进程）。机间通讯的规约，采用集中式的组织方式，将信件组织于CM中，各处理机通过对CM的读写来交换信件。机间通讯的链路，用建立于CM中的邮箱来提供，每个处理机建立一个邮箱登记表。

鉴于我们的多处理机系统是专为机器人控制而设计的，无需用编译程序去自动分析和识别应用程序中的并发关系，而是用由用户在作业中使用显式标记的方式来描述同一问题的多个任务间的并行性。为此，我们在多机OS的并发机制中引入了两条用于实现任务派生和汇合的原语：FORK和JOIN。

此外，为了实现进程控制，实现进程对临界区中共享表格的互斥访问，实现对处理机和存储器的管理等，OS内核中还设置了创建、撤消，阻塞、唤醒，开锁、关锁，以及分配、回收进程控制块和消息缓冲区等原语。

### 2.3 简单动作级机器人语言

我们设计的多微机系统，对于实现机器人控制将具有一定的通用性，能灵活地适应不同的机器人控制对象和控制任务，为此，有必要设计一个实用的简单动作级机器人语言，以便为用户提供比较方便、良好的编程环境。

我们设想，简单动作级机器人语言的设计原则是：以C语言为基础，引入若干反映机器人动作的语句，其中每个动作语句实质上包含一个机器人运动控制的函数包，即用C语言或汇编语言编写而成的、能完成机器人控制任务中某种基本功能的子程序。我们可以将机器人控制（包括力控制）中各种常用的基本函数和功能都用C语言或汇编语言编成子程序，构成应用程序库，然后设计相应的语句和命令供用户编程时调用。

用户进行控制任务编程时，只需综合考虑任务本身的内在并行性、数据相关性、并行算法和同步、通讯开销等因素，按照模块内任务并行的思想，将应用程序划分成若干大小适当的模块，然后系统以串并结合的方式运行各个模块，能并行执行的模块并行执行，不能并行执行的模块串行执行，如此以模块为单位，顺序向前推进。

## 3 结 论

本文以实际科研为背景，提出了一种用于机器人力反馈依从控制的计算机体系结构，即总线结构的MIMD型多微机并行处理系统。首先从总线结构框图及其各部分功



图6 OS层次结构

能、存储器组织、系统仲裁逻辑、中断逻辑和 I/O 接口等方面说明了硬件系统结构,接着从引导程序、多机并发操作系统内核和简单动作级机器人语言等方面说明了系统软件的基本思想。该多处理机系统能有效地实现机器人力/位置混合控制中多任务的并行处理,将伺服控制速率提高到 200HZ 以上。

### 参 考 文 献

- [1] Chen J Bradley, Fearing Ronald S, Armstrong Brian S, Burdick Joel W. NYMPH: A Multiprocessor for Manipulation Applications. IEEE International Conference on Robotics and Automation, April, 1986
- [2] Kametani M, Watanabe T. Hardware and software of A Multi-Microprocessor System Applied for Robot control. IECON 1984:749~753
- [3] Liu C H, Chen Y M. Multimicroprocessor-Based Cartesian Space Control Techniques for A Mechanical Manipulator. In proc. IEEE Int. Conf. on Robotics and Automation, 1986: 823~827
- [4] Backes Paul G, Leinger Gary G, Chun-Hsien Chung. Real Time Cartesian Coordinate Hybrid Control of A PUMA 580 Manipulator. In proc. IEEE Int. Conf. on Robotics and Automation, 1985: 815~821
- [5] Hwang Kai, Briggs Faye A. Computer Architecture and Parallel Processing. McGraw-hill Book Company
- [6] 邹逢兴.关于机器人力控制器计算机系统的设计方案探讨.国防科技大学学报, 1990, 12(2): 73~79
- [7] 区益善.多微机系统与并行处理.华南工学院出版社, 1987
- [8] [美]J. B. 约翰逊, S. 卡塞尔著; 张玉轩, 梁适, 钟仲凯译. 多总线设计手册. 科学出版社, 1987

## A Computer Architecture Used in Robot Force-Feedback Compliance Control

Zou Fengxing

(Department of Automatic Control)

### Abstract

A new computer architecture-MIMD multimicroprocessor parallel processing system used in robot force-feedback compliance control—is described from the views of system hardware and software. The system is designed for fulfilling a practical research project. It can efficiently and parallelly process the multitask in the robot force/position mixed control. The realization of the system will make the servo-control period shorten to 5 ms and meet various needs of robot control tasks.

Key words robot, computer system, parallel processing/force—feedback control, computer architecture, multimicroprocessor system