

智能工具机系统结构*

王 朴 张晨曦 胡守仁

(计算机系)

摘 要 本文提出了一种面向 PROLOG 的计算机系统, 它采用传统计算机做为主机, 后端机采用通用 CPU 芯片和支持 PROLOG 高速执行的专用硬件。该系统结构简单, 性能高。性能模拟表明, 该系统具有 500KLIPS 的逻辑推理速度和 17MIPS 的数值计算速度。

关键词 计算机系统, 人工智能, PROLOG/PROLOG 机, WAM 模型, 一致化操作, dereference 操作

分类号 TP302.1, TP387

PROLOG 语言具有描述能力强, 语法简单, 使用方便等优点, 已在人工智能领域获得广泛应用。实践证明, PROLOG 是当今几种最重要的人工智能语言之一。

国外已研制出许多 PROLOG 系统和 PROLOG 机, 从发展趋势看, PROLOG 系统已由解释型向编译型发展。PROLOG 机已由直接面向 PROLOG 语句的解释执行向较低级抽象指令的编译执行方向发展。这种发展的推动力是后者能够充分利用传统计算机中的加速技术, 从而获得较高的执行速度。

现有顺序 PROLOG 机, 按其体系结构可分为独立型、依附型和集成型三种。

独立型 PROLOG 机, 以 PSI 为代表。由于它完全独立, 需要开发所有管理软件和程序设计环境, 工作量大, 而且独立型 PROLOG 机不能很好地支持数值计算, 应用受到了限制。

依附型 PROLOG 机把 PROLOG 机做为后端机, 附加在某一种传统机器上。这样可以充分利用传统机上的管理软件和程序设计环境, 也可利用传统机上的数值计算能力, 因而研制开销少, 大多数 PROLOG 机都采用这种体系结构。由于数值计算在主机上, 逻辑推理在后端机上, 使得主机和后端机通信频繁, 开销增加。

集成型 PROLOG 机 (如 IPP), 把数值计算和逻辑推理集成在一起, 按数值计算设计 CPU, 并增加支持推理操作的硬件, 如支持标志测试, 标志分解, 标志合并和多路转移的硬件, 支持一致化 (unification) 操作的硬件, 从而克服了独立型支持数值计算和依附型通信开销大的不足。但是, 仍然需要开发系统管理软件和程序设计环境, 研制开销大的问题并未解决。

本文提出的智能工具机, 采用了较低的指令抽象级, 其体系结构集中了依附型和集

* 本项研究得到国家八六三计划的资助

1990年1月16日收稿

成型两种体系结构的优点，克服了各自的缺点。

本文第一节介绍智能工具机的系统结构，第二节介绍后端机PP，第三节介绍后端机的专用部件，最后一节为几点结论。

1 智能工具机的系统结构

为了支持逻辑程序设计语言的高速执行，智能工具机中的语言分三个层次(图1)：最高层是扩充了的PROLOG语言，中间层是扩充了的Warren抽象指令，最低层是通用CPU芯片的汇编语言ASM。

扩充了的PROLOG语言，是在PROLOG的基础上扩充了等词逻辑和面向对象的功能，使其兼有逻辑、函数和面向对象三种程序设计语言风格。

扩充了的Warren抽象指令集，是在

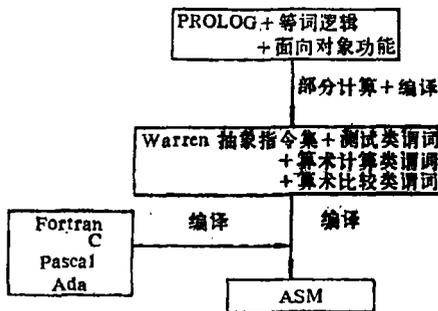


图1 智能工具机的语言层次结构

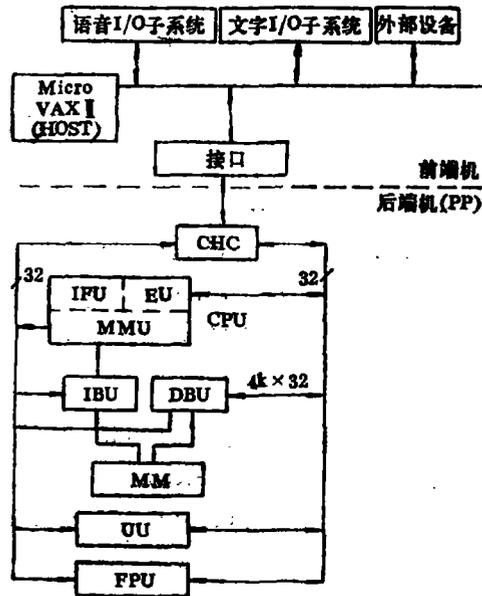


图2 智能工具机的系统结构

IFU: 指令取部件, EU: 执行部件, MMU: 存储管理部件, IBU: 指令缓冲部件, DBU: 数据缓冲部件, MM: 主存储器, UU: 一致化部件, FPU: 浮点处理部件, CHC: 通道控制部件

Warren抽象指令集的基础上，增加支持非逻辑成份操作的指令，和调用其它语言写的软件接口，扩充了测试类谓词、算术比较类谓词、算术计算类谓词等，从而加速了这些谓词的操作。

此外，通用CPU芯片还支持一般的程序设计语言Fortran, C, Pascal, Ada等。

智能工具机的体系结构如图2所示，它包括前端机和后端机两大部分。

前端机采用传统的Micro-VAX I计算机，程序的修改和调试，文件的编辑和管理，I/O的控制，外设的控制和管理，都将由前端机完成，从而充分地利用了Micro-VAX I的系统管理功能和环境。为了提供友善的用户介面，前端机将增加汉语语音I/O功能和文字I/O功能。

后端机是面向逻辑程序设计语言的PROLOG处理器(PP)，由通用的CPU芯片和支持逻辑程序运行的专用硬件DBU和UU，一个大容量主存储器(MM)，一个指令Cache，一个与前端机接口的通道控制器(CHC)和一个可选的浮点处理部件(FPU)组成。后端机既能支持逻辑程序的高速运行，又支持高速的数值计算。

智能工具机的软件配置包括操作系统，数据库管理系统，人工智能程序设计环境等。

操作系统OS和子句编译器(命令解释程序、通讯程序VAXCOM除外)常驻在后端机上,其它软件常驻在前端机上。

2 后端处理机(PP)

2.1 PP的执行模型

WAM模型已成为许多高性能的PROLOG机的基础,但WAM不能有效地支持非逻辑成分的操作,例如数据库操作。为此,PP执行模型以WAM模型为基础,扩充了WAM—PLUS模型中非逻辑成分执行机制。

许多WAM机,采用微程序实现,其好处是可增加执行中的并行性。但由于现有微程序设计的芯片时钟频率都比较低,最高时钟频率只有10MHZ,而且用微程序设计方法,工作量大,最终获得的逻辑推理速度并不高。因此,从简化设计出发,我们利用通用CPU芯片的简单指令集来实现WARREN的抽象指令。由于通用CPU芯片的时钟频率高,可以弥补并行性的不足,最终获得的逻辑推理速度反而比采用微程序设计方法高。

在智能工具机中,PROLOG源程序经第一级编译后产生类似于Warren代码的中间代码,经第二级编译再产生可执行的机器代码。虽然在代码库中直接存放中间代码、用存放在ROM中的机器指令序列来解释中间代码的方法具有节省代码空间和便于实现代码库的管理等优点,但由于中间代码被解释时,其操作数是作为数据取入CPU进行处理的,对速度有较大的影响(降低)。因此,代码库中管理的将是机器指令形式出现的代码。

PP的执行模型具有下列特点:

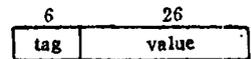
(1) 实现了PROLOG数据库的动态修改。我们简化了WAM—PLUS中的管理算法,使得既能实现动态修改,又不会花太大的修改开销。以过程为基本单位管理数据库,即代码库和源库在过程一级耦合,具有较高的性能。

(2) 实现了一致的PROLOG代码语义。

(3) 采用了代码分类,数据库操作内部谓词的操作模式等优化技术。

2.2 PP的数据类型与表示方法

PP的数据字长为32位,采用带标志的表示方式,6位标志域,26位值域(图3)。标志表示数据的类型,值域表示值或指针,取决于数据的类型。



PP中的基本数据类型有11种,即变量、原子、空表、表、结构、串、短整数、长整数、单精度实数、双精度实数、函数。下面着重说明两点:

图3 PP中的数据字

(1) PROLOG程序中的变量有两种,一种是未约束变量(unbound),另一种是引用变量(reference)。在PP中,这两种变量采用同一个标志。这样做的好处是,提高了运行效率。因为若采用不同的tag,则当把一个未约束变量拷贝到另一个变量或寄存器,需要进行tag的修改操作(或屏蔽操作)。这对于通用CPU来说,开销是较大的(至少需增加一条指令)。而设置不同的tag所带来的好处是使从存储器开始的dereference操作减少一次循环。这种情况最多不超过dereference所有情况的25%,而更多的dereference操

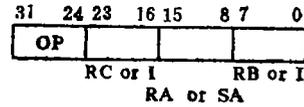
作是从寄存器开始。

采用同一标志以后,两种变量的唯一区别是其pointer,未约束变量的pointer指向自身单元的地址,引用变量的pointer指向被引用数据单元的地址。PP的专用硬件支持这种地址判别,从而dereference操作很容易区分这两种变量。

(2) 函词省缺了标志位。这是考虑到函词总是出现在结构指针所指的单元中,是隐含已知的。省去其tag,可把函词类型压缩到一个数据字内,减少占用的空间和访存次数。

2.3 PP的指令集

指令集分为9类,即:整数算术操作类指令,比较操作类指令,逻辑操作类指令,移位操作类指令,传送操作类指令,常数操作类指令,浮点
 数操作类指令,转移操作类指令,其它操作类指令。



PP中的所有指令字长都是32位,分为四个域,每个域都是8位,如图4所示。

图4 PP中的指令字格式

图4中的符号说明如下:

OP 操作码; RA, RB, RC 通用寄存器号; SA 专用寄存器号; I 直接操作数

2.4 PP的硬件结构

PP的硬件组成已在图2中给出,本节描述各部件的功能与特点。

采用的通用CPU芯片,具有很好的性能,其内部结构可分为三部分,即取指令部件(IFU),执行部件(EU),存储管理部件(MMU)。IFU能自动地完成指令的预处理,使得转移指令能有效地完成,执行部件能在一个时钟周期执行一条指令(按流水方式工作)。EU提供了一组通用寄存器文件,还有一组专用寄存器用于系统控制和某些数据的处理。大的寄存器文件,能保证大多数指令能在一个时钟周期内完成,采用延迟指令方法,能使转移指令、子程序调用指令均只占一个时钟周期。提供的成组读写指令,加速了成组读写操作。读写指令的控制域,允许CPU能灵活地访问外部部件,允许用户进行适当的扩充。MMU提供了地址变换和进程管理机构,便于实现虚拟存储器管理。大容量的寄存器文件,快速的转移指令执行机制,成组读写指令的设置,高的时钟周期频率,都有利于逻辑程序的高速执行。

IBU是一个指令高速缓冲部件,其目的是保证每一时钟周期能向CPU提供一条指令,减轻系统对主存贮器的要求。IBU的容量为4K字。

MM是一个16M字的主存贮器,采用模块化结构。大容量的存贮器,能满足大多数人工智能问题对存储空间的要求。模块化结构能满足不同用户的需要。

CHC是一个通道控制部件,是PP与前端机的接口。CHC能与远距离总线相连,为PP与多种前端机相连创造了条件。

FPU是一个浮点处理部件,可根据用户的需要选用。若采用FPU,将加速浮点操作的执行,选用专用芯片,可达5MFLOPS。若不用FPU时,CPU采用软件模拟的方法完成浮点操作,浮点操作执行速度降低一个数量级。

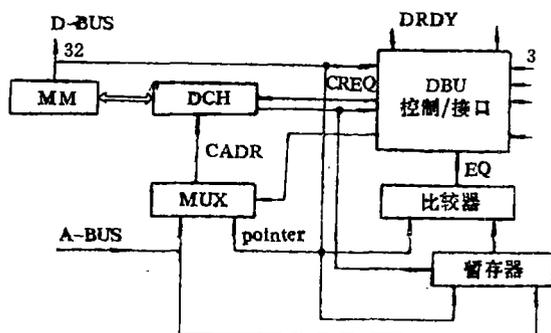
DBU和UU是支持逻辑程序运行的专用硬件,DBU主要支持快速的dereference操作,UU支持快速的一致化操作,并实现快速多路转移。下一节将进一步说明。

3 PP的专用部件

PP的专用部件主要指为支持逻辑程序快速运行而设计的部件, 包括DBU和UU.

3.1 DBU部件

dereference操作是PROLOG程序编译实现的基本操作, DBU应该能够提供递归访问功能; 由于PP中的数据类型将未约束变量和引用变量的标志合成一个, DBU还应能提供地址比较功能, 以判断dereference操作何时结束. DBU同时应该是一个快速的数据Cache, 应能使数据命中率达到90%以上, 而且在非dereference操作时, 不影响数据访问速度.



DCH: 数据Cache, MM: 主存储器, CREQ: 对Cache的请求
CADR: 访问Cache的地址, CRDY: Cache数据就绪

图5 DBU的框图

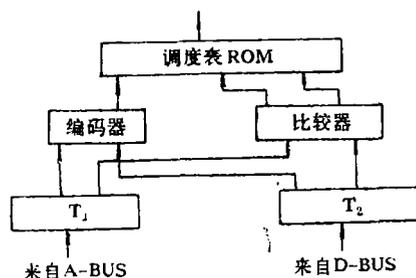


图6 UU的框图

图5是DBU的框图. dereference操作终止条件的判定与Cache的访问是相互独立的两部分. 暂存器暂存上一次访问Cache的地址, 用于判断dereference的终止条件. 控制/接口电路用于产生控制DBU信号和与CPU的接口信号.

若不采用DBU, 实现dereference操作要 $4(1+n)$ 个时钟周期 (n 为dereference深度), 采用DBU后只需 $2+2n$ 个时钟周期, 提高速度一倍.

此外, 由于约束变量和引用变量采用同一标志, 省却了某些标志修改, 标志拼接操作, 也提高了执行速度.

3.2 UU部件

一致化操作是PROLOG程序中的基本操作, 不少PROLOG机都从硬件上支持一致化操作 (如PEK). UU的设计, 一方面是支持简单数据类型的比较, 如原子, 变量, 短整数, 空表. 因为这几项的一致化 (变量除外), 就是比较操作. 另一方面, UU要根据两个项的标志类型, 迅速转移到对复合项进行一致化操作的子程序入口, 快速地完成多路转移.

图6给出UU的框图. 其中的比较器用于在某些情况 (例如T1和T2中的内容都是原子或短整数) 判断一致化是否成功, 以及用于在两者都为变量的情况下判断哪个变量的地址小. 编码器的作用是减少ROM的空间大小.

根据典型PROLOG程序的统计, 子句体目标和子句头中的自变量为原子的情况,

分别占50~60%，44~87%。由于引用变量经过 dereference 操作后，又有一部分是原子，所以实际参加一致化的项为原子的比例更高。采用 UU，可使大多数一致化操作在六个时钟周期内完成。若没有 UU 的支持，CPU 根据两个项的类型标志实现多路转移则需要一个复杂的指令序列才能完成。

4 结 论

(1) 智能工具机的系统结构，具有依附型和集成型两种系统结构的优点，克服了两者各自的缺点，使整个性能提高。

(2) PP 采用通用 CPU 芯片加专用硬件，既能支持高速的数值计算，又能支持高速的逻辑推理，同时简化了设计，使 PP 结构十分简单，而性能很高。数值计算速度平均为 17MIPS 以上；模拟表明，逻辑推理速度平均为 500KLIPS 左右。

(3) 在 PP 的设计中，把未约束变量和引用变量的标志合一，利用专用硬件支持 dereference 操作和一致化操作，有效地提高了逻辑程序的执行速度，而所增加的硬件量极少。

感 谢

智能工具机课题组的同志都参加了智能工具机的总体方案讨论，在此致谢。

参 考 文 献

- [1] 王朴. PROLOG 与 PROLOG 机. 见: 高庆狮主编, 智能技术与系统基础. 北京大学出版社, 1990, 216~252
- [2] 张晨曦. 基于 Warren 抽象机的 PROLOG 实现技术的研究. 工学博士论文, 国防科技大学研究生院, 1987
- [3] 王朴, 张晨曦, 朱海滨. 智能工具机的性能分析. 国防科技大学报, 1990, 12(4)
- [4] Touati H, Despain A. An Empirical Study of the Warren Abstract Machine. IEEE, 1987
- [5] Nakajima K, Nakashima H, et al. Evaluation of PSI Micro-Interpreter. The 13th Annual International Symposium on Computer Architecture
- [6] Tick R and Warren D H D. Towards a Pipelined Prolog Processor. New Gener. Comput. 1984(2):323~345
- [7] Taki K, Yokota M, et al. Hardware Design and Implementation of the Personal Sequential Inference Machine (PSI). In: Proc. Int. Conf. on Fifth Generation Computer Systems, Tokyo, 1984; ICOT, 1984, 398~409
- [8] Kaneda Y et al. Sequential Prolog Machine PEK. New Gener. Comput. 1986, (4), 51~66
- [9] Yamaguchi S et al. Architecture of High Performance Integrated Prolog Processor IPP. In: Proc. of Fall Joint Computer Conf. Dallas, 1987; IEEE Society Press, 1987, 175~181
- [10] Sriniv V P et al. A COMS chip for PROLOG. In: Proc. Int. Conf. on Computer Design, New York, 1987

多体存贮器有效频带分析

吴涛 陈福接

(计算机系)

摘要 本文以排队论为工具建立了紧耦合多处理机系统存贮器的排队模型。通过对存贮体忙期的分析, 得出了存贮器有效频带的解析表达式, 研究了在保证存贮器有效频带最大的情况下, 处理机的访存请求率 λ 、存贮体的服务率 μ 和存贮体数 m 三者之间的关系。

关键词 存贮器, 排队论, 模型/有效频带, 排队模型

分类号 TP301

计算机已经进入多处理机时代。多处理机系统有紧耦合和松散耦合两种结构。特别是共享主存的紧耦合多处理机系统对存贮器提出了更高的要求。如著名的巨型机系统Cray X—MP, Cray-2等都对存贮器的结构和性能做了很好的设计和优化。存贮器在多处理机系统中占有重要的位置。

在多处理机系统中, 存贮器是由多个存贮体组成的, 系统通过对多个体的并行访问

1989年10月5日收稿

The System Architecture of an Intelligent Machine

Wang Pu Zhang Chenxi Hu Shouren

(Department of Computer Science)

Abstract

This paper puts forward the architecture and organization of a high performance prolog-oriented intelligent machine system, which comprises a conventional computer as its host. A powerful prolog processor supported by a special hardware is its back-end machine. Although the architecture of the system is simple, the performance is very high. Simulation results show that the performance of the system is about 500 KLIPS for logical inference and 17 MIPS for numerical calculations.

Key Words computer system, artificial intelligence, PROLOG/prolog machine, WAM mode, unification operation, dereference operation