

第一部份 RISC 技术综述

第一章 RISC 技术的发展

RISC (精简指令集计算机) 技术是计算机体系结构发展史上又一次重大变革。RISC 体系结构为计算机系统带来了巨大的性能价格比优势, 深刻地改变了整个计算机工业的面貌, 从 80 年代中期至今, 单 CPU RISC 计算机每隔 12 至 18 个月运算速度翻一番, 从 86 年的 10MIPS 展到 91 年的 50MIPS, 这是迄今计算机发展史上性能增长最快的时期。目前, RISC 系统的性能已全面超过了 CISC (复杂指令集计算机) 系统。

进入 90 年代, RISC 技术的发展势头依然不减, 主要的 RISC 公司继续投入大量人力、资金进行开发, 在 1991 年底都已先后推出了第三代 RISC 微处理器, 其主要技术特征是采用超级流水线、超标量等指令级并行技术, 使整数运算部件和浮点运算部件各自的指令流水线在每个时钟周期内执行两条指令 ($CPI < 0.5$), 使单片 CPU 运算速度达 50 至 150MIPS 第三代 RISC 微处理器芯片把 RISC 性能价格比推向新的水平。从更长期的发展趋势来看 (如到 2000 年), 单 CPU 微处理器的性能可达到 1000MIPS

在计算机市场上, 广大用户正在由专用、封闭的系统结构迅速地转向通用、开放式系统, RISC 系统产品顺应并推动了这一发展潮流。性能高的且开放性好的 RISC 体系结构能满足各种计算平台的需要, 而被工业界广泛接受为标准。计算机发展史上将第一次出现能够实现从笔记本型机、膝上机、台式机到大型机全功能范围, 从而使它们互相兼容的单一体系结构—RISC 体系结构。

由美国加州大学伯克萊分校于 80 年代初研制成 RISC-I 和 RISC-II 至今虽然还不到十年, 可是由于采用 RISC 设计技术的处理器和计算机系统因其具有结构简单、紧凑, 工艺易于实现, 研制周期短, 性能价格比高等特点, 而得到各半导体制造商和计算机系统公司的青睐, 现已推出了几十种各种功能的 RISC 处理器芯片, 其中包括以工程工作站起家的 SUN 公司的 SPARC 芯片, MIPS 公司的 R3000 芯片; Intel 公司的 i80860 芯片; Motorola 公司的 MC88000 芯片和 IBM 公司的 RS/6000 芯片。同时, 还有更多的计算机制造厂商也不断地选用或自制 RISC 微处理器来设计制造各种高性能的计算机系统, 以满足日益增长的应用要求。随着 UNIX 操作系统普及, 编译技术的成熟和工艺技术的进步, RISC 微处理器的性能将越来越高, 甚至有的产品已可构成一个计算机系统了, 其应用范围也将不断扩大, RISC 处理器除已在工程工作站产品中占主导地位外, 现已逐渐用于小型计算机、中、大型计算机系统, 甚至也扩大到巨型计算机系统中, RISC 技术已遍及整个计算机领域。

1.1 RISC 技术的由来

1.1.1 复杂指令集计算机 (CISC)

长期来, 计算机性能的提高往往是通过增加硬件的复杂性来获得, 随着集成电路技术, 特别是 VLSI 技术的迅速发展, 为了软件编程方便和提高程序的运行速度, 硬件工程师采用的办法是不断增加可实现复杂功能的指令和多种灵活的编址方式, 甚至某些指令可支持高级语言语句归类后的复杂操作, 致使硬件越来越复杂, 造价也相应提高。为实现复杂操作, 微处理器除向程序员提供类似各种寄存器和机器指令功能外, 还通过存于只读存储器 (ROM) 中的微程序来实现其极强的功能, 微处理在分析每一条指令之后执行一系列初级指令运算来完成所需的功能, 这种设计的型式被称为复杂指令集计算机 (Complex Instruction Set Computer—CISC) 结构, 一般 CISC 计算机所含的指令数目至少 300 条以上, 有的甚至超过 500 条。

1.1.2 精简指令集计算机 (RISC)

采用复杂指令系统的计算机有着较强的处理高级语言的能力, 这对提高计算机的性能是有益的, 当计算机的设计沿着这条道路发展时, 有些人没有随波逐流, 他们回过头去看一看过去走过的道路, 开始怀疑这种传统的做法: IBM 公司设在纽约 Yorktown 的 JhomasI.Wason 研究中心于 1975 年组织力量研究指令系统的合理性问题, 因为当时已感到, 日趋庞杂的指令系统不但不易实现, 而且还可能降低系统性能。1979 年以帕特逊 (David A. Patterson) 教授为首的一批科学家也开始在美国加州大学伯克萊分校开展这一研究。结果表明, CISC 存在许多缺点, 首先, 在这种计算机中, 各种指令的使用率相差悬殊: 一个典型程序的运算过程所使用的 80% 指令, 只占一个处理器指令系统的 20%。事实上, 最频繁使用的指令是取、存和加这些最简单的指令。这样一来, 长期致力于复杂指令系统的设计, 实际上是在设计一种难得在实践中用得上的指令系统的处理器, 同时, 复杂的指令系统必然带来结构的复杂性, 这不但增加了设计的时间与成本还容易造成设计失误。此外, 尽管 VLSI 技术现在已达到很高的水平, 但也很难把 CISC 的全部硬件做在一个芯片上, 这也妨碍单片计算机的发展。在 CISC 中, 许多复杂指令需要很复杂的操作, 这类指令多数是某种高级语言的直接翻版, 因而通用性差, 由于采用二级的微码执行方式, 它也降低那些最频繁调用的简单指令系统的运行速度。因而, 针对 CISC 的这些弊病, 帕特逊等人提出了精简指令的设想即指令系统应当只包含那些使用频率很高的少量指令, 并提供一些必要的指令以支持操作系统和高级语言, 按照这个原则发展而成的计算机被称为精简指令集计算机 (Reduced Instruction Set Computer—RISC) 结构, 简称 RISC。

1.2 RISC 技术的类型

在众多的 RISC 处理器中, 如何了解它们的性能和设计技术的差别, 以便正确选择和应用它们, 这对系统设计者来讲是至关重要的, 由于早期 RISC 的发展基本上起源于一些著名大学和公司的研究计划, 虽然 RISC 的处理器品种很多, 制造厂商也不相同, 但就其性能和设计技术的差异可分四类: 它们的典型代表是 IBM 公司的 801 小型计算机; 帕克莱分校的 RISC I、RISC II; 斯坦福大学的 MIPS R2000、R3000 和 R4000 以及克雷公司的 CRAY-1, 目前几乎所有各大公司研制或生产的 RISC 处理器产品都分别隶属于这四种类型, 图 1-1 示出了 RISC 处理器的分类和发展, 它们的主要特性分述如下

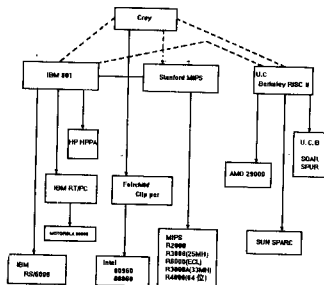


图 1-1 RISC 微处理器分类和发展

1.2.1 IBM 801 小型计算机

801 小型计算机是 IBM 公司于 70 年代末期开始研制的 32 位小型机, 它排除了通常的两级微码控制方式, 取而代之的是在硬件中直接执行一组精选的简单指令和优化的编译程序, 这种特殊的编译程序通过对程序详细执行步骤进行重新安排, 以便更有效地利用 CPU 资源来提高程序的运行速度, 因为它放弃了对复杂指令系统执行微码的方式, 这就有利于用硬连线控制系统来执行为数不多的简单指令, 为了达到单周期时间执行一条指令的目的, 存储结构使 CPU 不需等待访问存储器的时间, 提供指令和数据高速缓存 (Cache), 有 32 个 32 位寄存器, 所有指令在寄存器中执行, 同时编译器分寄存器和全局优化及存取校验, 还具有延迟转移功能, 是一台采用紧耦合硬件和编译技术相结合的 RISC 计算机。

1.2.2 帕克莱分校的 RISC I、RISC II 型

RISC I, RISC II 是帕克莱分校分别于 1982 年和 1984 年研制成功的 32 位微处理器, 它们的设计专注于最有效地利用大规模集成电路 (VLSI) 芯片的区域研究, 因为 RISC 设计的有限个简单指令只需要相对少量的芯片控制逻辑电路, 而芯片上更多的区域可以用作系统功能, 以加强处理器的性能并使它的功能多样化。它们采用重叠窗口技术, 有高达 138×32 位寄存器, 组成 8 个重叠窗口, 在相邻界面上有 6 个寄存器以提供窗口间的运行参数和结果。它们是专为相邻两个窗口共享的, 在处理器中, 寄存器作为临时数据存贮的快速存取区, 通过这些寄存器保持信息参数, 它们从一个程序传送到另一程序 (由于程序调用实现)。在 CISC 计算机中, 子程序调用之前, 参数是存于系统内存中, 这样非常浪费时间, 而 RISC I, RISC II 利用寄存器来传送参数很有效地降低了执行子程序调用时所需要的时间。指令数为 31 个, RISC II 在整数算术运算方面比现存的复杂指令的 VAX II / 780 还快。

1.2.3 MIPS R $\times$ 000 系列型

斯坦福大学的 MIPS 计划, 它不采用帕克莱分校的 RISC 窗口技术而采用消除流水线各级互锁的微处理器 MIPS (Microprocessor Without Interlocking Pipeline Stage) 技术。它将 IBM 公司对优化编译程序的研究和帕克莱分校的大规模集成电路的思想结合起来。流水线是现代各类微处理器都采用的指令执行技巧, 即将若干条指令的取指令、译码和执行过程部分重叠在流水线中同时执行, 但在 CISC 计算机中, 由于指令多而复杂, 处理每条指令的所需时间不固定, 当后面指令需要前条指令的结果, 往往造成指令互锁, 斯坦福大学的 MIPS 计划就是在编译过程中, 利用编译程序优化处理器的流水线, 以求提高处理器流水线的效率, 由于采用硬连线控制来执行为数不多的简单指令, 且能重组软件流水线, 这样就减少硬件复杂性。

1.2.4 CRAY-I 型

克雷 (CRAY) 公司的巨型机 CRAY-1, 它是 70 年代中期开发成功的最先采用面向寄存器操作的精简指令的高速计算机, 公司创始人 Seymour CRAY 一开始就反对过在他的巨型计算机系列的设计中采用增加指令的建议, 因此 CRAY-1 除大量采用标量, 向量和地址寄存器外还采用流水线技术以达到高速运算的能力, 做到了一个机器周期时间里执行一条指令的目标, 所以有人讲 CRAY-1 是 RISC 技术的鼻祖。

由于这些研究计划的实施, 奠定了 RISC 设计基础, 通过这些处理器成功的实验和由它们构成的计算机系统结果表明 RISC 技术优于 CISC, 深得广大用户的青睐。世界上很多计算机制造厂商和半导体生产厂都不断研制和推出各种性能的 RISC 微处理器, 在众多 RISC 的微处理器中, 根据其性能和设计技术间的区别均可分属上述四类: Sun 公司的 SPARC (scaleable processor architecture) 微处理器芯片和 AMD29000 微处理器属帕克

来分枝的 RISC I, RISC II 结构; IBM 公司的 Power RS / 6000, Romp / PC-RT 和 HP 的 PA (Precision Architecture) 属 IBM801 结构; MIPS 公司的 R2000, R3000, R6000, R4000 等都属斯坦福大学的 MIPS 结构, 而 Intel 公司的 i80860 和 Fairchild 公司的 Cippier 属 CRAY-1 结构。

1.3 RISC 技术的设计技巧

RISC 的设计原则是使系统设计达到最高的有效速度, 将那些能对系统性能产生净增益的功能用硬件实现, 其余大部分都用软件实现, 它排除了那些实现复杂功能的复杂指令, 所谓“精简”并不是简单的简少, 而是保留经验证的能提高机器性能的指令, 另外还将编译器作为机器的功能, 而且 RISC 微处理器使编译器能够直接访问基本的硬件功能, RISC 设计的基本目的在于使计算机结构更加简单、更加合理、更加有效, 指令经过精简后, 计算机体系结构自然趋于简单, 在这个基础上, 还必须克服 CISC 的许多缺点, 使得计算机速度更快, 程序运行时间缩短。这样, RISC 才能以崭新的面貌出现。

自从 80 年代中期 RISC 结构的计算机商品化以来, 全世界几乎所有计算机系统制造厂商都竞相采用, 半导体厂商也不断研制性能越来越强、集成度越来越高的 RISC 微处理器, 甚至一向以发展 CISC 而著称的 Motorola 和 Intel 公司也同时发展 RISC 微处理器产品系列。RISC 不是一个产品而是一种设计技术, 人们可以根据不同要求选择使用这些设计技术用于各类计算机系统中, 以利于改善和提高机器性能。RISC 技术的主要设计技巧及其特点分述如下。

1.3.1 短周期时间

为了指令的快速执行, 就必须快速选择译码和减小寄存器存取时间, 尽量采用先进工艺技术以缩短机器周期时间, 也即提高机器的工作频率。

1.3.2 单周期执行指令

由于 RISC 微处理器的指令经过精选, 所有指令长度都相同, 大多数指令都能在一个机器周期内执行完, 实际上, 大多数 RISC 微处理器在没有增加并行技术时, 平均每条指令执行时间为 1.25~2 个机器周期时间, 而 CISC 微处理器在相同工艺技术条件下平均执行每条指令需 5~8 个机器周期时间, 如 DEC VAX II / 780 小型机执行每条指令平均需 5 个机器周期时间, 而 Motorola 公司的 M68020 微处理器执行每条指令平均需 8 个机器周期时间, RISC 微处理器力求达到每一个机器周期时间执行一条指令, 单周期执行指令是 RISC 微处理器性能增强的基础, 必须简化指令系统和采用流水线技术。

1.3.3 Load (取) / Store (存) 结构

每当一条指令需要访问主存信息时, 机器的执行速度将会降低。RISC 的 Load

/Store 指令只有在访问内存时才使用,所有其它的指令都是在寄存器内对数据进行运算。一条存取数指令 (Load) 从内存将数据取出放到寄存器中,在那里可以对数据进行快速处理,并把它暂存在寄存器里,以便将来还要使用。在适当的时候,一条存数指令 (Store) 可将这个数据送回到它在内存中的地址中去。CISC 微处理器支持那些直接从内存处理信息的指令,这些指令需要多个机器周期时间才能完成。RISC 的设计技术与 CISC 的设计技术相比,有大量寄存器。由于允许数据在寄存器中保留较长的时间,这样就减少了存/取指令对内存访问的需要。在寄存器中,每当再被使用时不必再次访问内存。这种 Load/Store 结构通过寄存器对寄存器进行操作的方式乃是获得单周期执行的关键。

1.3.4 简单固定格式的指令系统

所有指令采用 32 位固定长度,寻址方式不超过三种,简化了逻辑和缩短译码时间,确保单周期执行指令,同时也有利于流水线操作的执行,这是由于指令的固定格式保证指令译码和取操作数同时进行。

1.3.5 不用微码技术

由于 RISC 的设计采用简单、合理的指令系统和简化的寻址方式,所以排除微码设计技术,也即不采用微码只读存储器 (ROM),而是直接在硬件中执行指令,这意味着省去将机器指令转换成原始微码这一中间步骤,这也就减少了执行一条指令所需要的机器周期个数,这也就节省了芯片的空间使得可以利用这些节省下来的芯片空间扩展微处理器功能。

1.3.6 大寄存器堆

RISC 微处理器中大量的计算都在 ALU 高速寄存器中执行,由编译器产生、分配和优化寄存器的使用,从而简化流水线结构和使指令周期降到最小,同时又不访问内存,允许调用的嵌套执行,但这也增加 ALU 周期中的寄存器存取时间和一些选址机构,因此在任务变换中需要较高的开销。

1.3.7 哈佛 (Harvard) 总线结构

采用指令和数据高速缓存 (Cache),利用双总线动态访问机构,填入执行程序有利于单周期执行指令,又可双倍增加数据带宽以提高数据吞吐量,在片高速缓存容量的增加将占较大芯片空间,而脱片 Cache 也将增加存取延迟时间。

1.3.8 高效的流水线操作

当前不论什么结构的微处理器都毫无例外地采用流水线技术,以达到高速执行指令的能力,因为流水线的每一级都负责执行一个单个的操作段,比如,指令译码或取操作数,实际上一个流水线微处理器是在同一时刻对数条指令进行操作,CISC微处理器执行指令时效率低,甚至有时会使得执行过程处于短暂的停滞状态,例如,当处理器遇到一条执行时间比预定时间要长的指令,它必须延长这条指令的操作,这样就要阻止其它指令在流水线中正常执行流水线操作,直到这条长指令的完成。这种状态除了降低了执行指令效率外,还迫使设计者将处理器的微结构在硬件上设计得更加复杂,以便对付这些问题。而在RISC微处理器设计中,它具有对指令执行时间的预测能力,因此它能使流水线在高效率状态下运行。

1.3.9 延迟转移

这可解决指令转移相关问题,它把转移指令与前一条指令交换位置以避免浪费一个机器周期时间,这不是靠程序员去完成,而是由RISC的编译程序自动完成,另外还有的RISC微处理器芯片采用可存放128个转移目标地址的高速缓冲存储器,这样就能确保在一个机器周期内执行一条转移指令。这也是RISC设计技术一个重要特性。

1.3.10 硬连线控制

采用少量、简单、固定的硬连线控制逻辑替代微码以实现减少指令系统,保证短周期,单周期执行指令,但不能处理复杂指令,除在特定状态机(State machine)或使用垂直微码外,不能处理多个Load/Store指令。

1.3.11 重叠寄存器窗口技术

为了简单有效的支持高级语言,RISC设计中把大寄存器堆分成多个重叠寄存器窗口,用以在执行高级语言中的过程调用和返回子程序的直接转换参数,这样就减少了调用和返回访问主存所消耗的计算时间,在RISC机器中,复杂指令是用子程序来实现的,因此RISC程序中的调用数量必然大超过CISC程序中的调用数量,帕克莱分校的巴斯克特(F.Baskett)提出用重叠寄存器窗口(Overlapping Register Window)技术的设计思想,这样就大大减少调用和返回子程序访问的次数,这是一个极其有效的设计技术,不少RISC微处理器都采用这种技术。

1.3.12 优化编译程序

编译程序能够分析数据流和控制流,并在这个基础上调整指令的执行顺序,巧妙安排

寄存器的用法，在 RISC 的设计中，内存访问和条件转移都可能与流水线相关的问题，而优化编译器可以替代用复杂、昂贵的硬件来解决的难题，例如，在访问内存引起的时间延迟，可以通过合理地利用寄存器使之达到最小影响程度。当一个寄存器的内容为随后的运算所利用，而又无须重新从内存那里去取时，优化编译程序可以识别出这种状态来。当遇到这样一些指令，访问内存不可避免时，编译程序能够重新排列这些指令，使得微处理器在等待把数据调入寄存器的这个时间里，其它有效工作照样执行，并不需要等待取数据时间。类似地，一个优化编译程序也可以通过使用“延迟转移”的方法来处理无法预测的条件转移，这个技术也是重新安排指令顺序，当微处理器在判断转移条件时，允许在条件转移后面的指令先执行，虽然流水线和优化编译程序并不是 RISC 技术所独有，但是这个技术与 RISC 相结合比与 CISC 相结合更加有效。例如，CISC 编译程序的设计者必须开发复杂的结构才能保证编译程序从正在被编译的程序的各个不同点上选择最有效的寻址方式和指令，它们不能像 RISC 可以对设计相关联的大寄存器进行存取。RISC 设计的大量寄存器，少而精的指令系统及单一寻址方式使得它比较容易做到一个较低限度地访问内存的、能够识别顺序计算的和最有效地利用流水线指令的编译程序。对于 RISC 机器的优化编译程序能够比较简单的测算由优化带来的节省，因为所有指令执行都只需一个机器周期时间。另一方面 CISC 机器的优化不得不把指令分成若干种可选择的指令顺序，而每一种顺序的执行都对应不同的执行周期时间。总之，对面向寄存器的指令，Load/Store 指令和转移指令的简单指令系统由优化编译程序处理非常合理。通过优化使用寄存器，减少访问内存时间或使寄存器间的操作和访问内存并行执行。优化编译程序可以解决流水线中的相关处理。同时还可提前执行检测条件，可节省许多不必要的测试操作。

1.3.13 增强存贮管理功能

由于 RISC 设计提高了 CPU 的性能，这就非常有必要提供一个快速有效存贮管理部件 (MMU) 来使各存贮层次与处理器的效率相匹配。在 RISC 设计中存贮层次由大量的芯片寄存器，用于分离数据存贮区和指令区的高速缓冲存贮器 (Cache)，写缓冲区以及芯片内存管理器等部分组成，随着半导体存贮器的价格下降，密度和速度的提高，促使要设计高性能的存贮层次来支持 RISC 处理器的速度。存贮管理的目的不仅要处理地址变换和提高高速缓冲的命中率，而且要能支持多任务环境下的大虚拟空间。

1.3.14 面向高级语言

在 RISC 设计中，不用微码，高级语言可直接由硬件实现，通过一次编译直接生成由硬件执行的机器代码。这样可提高执行速度。不过程序员就没有可读的汇编语言，适合人工编码。

以上列举的 RISC 技术的设计原则和设计技巧，已广泛用于现有的微处理器中，虽然应用的程序不完全一样，但这些设计方法的选用对提高机器的性能是有效的，它们已逐渐被越来越多的人所接受，而且已渗透到 CISC 计算机结构中，如美国国家半导体公司的 NSC32532 微处理器，它是 CISC 结构的处理器，加入上述某些设计技术，在其芯片上采

用指令和数据 Cache, 流水线和转移预拟逻辑, 简单指令用硬连线控制, 复杂指令保留微码控制, 结果在采用 25MHz, 1.25 μ m CMOS 工艺条件下, 平均每条指令执行时间从原来的 6 个机器周期时间降到小于 2.4 个机器周期时间, 机器处理速度高达 10—12MIPS, 接近 RISC 机器水平, Intel 公司 CISC 微处理器 Intel 80486 和 Motorola 公司的 CISC 微处理器 MC68040 也都吸取上述更多的 RISC 设计技术, 使每条指令平均执行时间小于 2 个机器周期时间, 尽管 RISC 的设计技术对计算机系统结构产生了巨大影响, 但 RISC 结构也存在一些不足之处, 如编译程序较 CISC 结构难写, 对虚存和浮点指令的处理也不如 CISC 理想, 因此在 RISC 的结构中也采用一些 CISC 技术中行之有效的设计方法以获得它们两者之间的完美结合, 有人提出集中 RISC 和 CISC 优点的新的计算机结构称 CRISC。

1.4 新一代 RISC 微处理器的发展

简单、固定长、单周期执行指令的 RISC 微处理器与复杂、可变长、并行（指令级）执行指令的 CISC 微处理器, 经过近年来的角逐, 已显示出 RISC 的性能明显优于 CISC。在相同的工艺制造技术条件下, RISC 的处理速度是 CISC 的 2~5 倍, 其中 RISC 的整数处理速度是 CISC 的 1~5 倍, 而 RISC 的浮点处理速度是 CISC 的 5 倍。因此 RISC 技术的研究和开发工作得到了越来越多的半导体厂商和计算机制造商的重视, 由 RISC 微处理器构成的计算机受到更多用户的欢迎。

自从 1985 年 Sun 公司和 MIPS 公司把它们的 SPARC 和 MIPS R2000 芯片作为商业产品投放市场以来, 晶体管集成度由几个增加到 100 万个, 工作频率由 8MHz 增加到 40MHz 以上; 芯片功能已从单纯的整数处理部件扩展到包括整数处理, 浮点处理、Cache 存储器 and 图形控制等多种部件。随着半导体微细加工工艺和新器件的进步以及 RISC 体系结构的新发展, 按照 80 年代的发展速度, 90 年代 RISC 芯片的集成度将每 3 年增长 4 倍, 处理速度每 3 年提高一个数量级, RISC 微处理器每 3 年更换一代。因此, 90 年代末期的 RISC 将是并行多处理器的体系结构, 处理速度高达 1000MIPS (即 1GIPS = 10 亿次指令/秒), 将广泛用于大、中、小型机, 高速服务器, 大小巨型机以及 PC、工程工作站和嵌入式控制器中。

1.4.1 90 年代的 RISC 微处理器

计算机进步的主要推动力历来就是高速逻辑器件的开关速度及其高集成度。由于先进半导体技术、存储介质、软件技术和其它有关工艺技术的发展, 计算机的处理速度基本上按指数曲线上升。过去的巨型机现在已可做成个人台式计算机; 过去用在大型计算机系统上的 Cache 存储、协处理、流水线和 CPU 并行处理等技术, 现已集成到单 RISC 芯片中; 而近来发展的数字信号处理、多机处理、阵列处理、专用算法处理、自适应处理、神经网络计算机和模糊逻辑运算等新技术也已开始集成在 RISC 芯片中。如日立公司在去年底首次制成圆片规模集成 WSI (Wafer Scale Integrated) 的神经网络计算机。采用 0.8 μ m 的 CMOS 工艺, 集成度高达 2000 万个晶体管。它由 60 个高密度的门阵

组成, 其中 48 个门阵构成神经元, 12 个门阵由 120 万个门组成, 由此构成的神经网络计算机可以模拟人脑动态并行处理功能。

90 年代 RISC 微处理器的发展趋势见表 1.1。

表 1.1 90 年代 RISC 微处理器的发展

时间	90 年代初	90 年代中	90 年代末
设计精度 (μm)	0.8~1.0	0.5~0.8	0.25~0.5
集成晶体管数 (万)	100	1000	10000
处理速度 (MIPS)	50	200	500~1000
体系结构	指令级并行处理*	多处理器	多处理器
Cache	高速多级 Cache	多机 Cache	多机 Cache
工艺技术	CMOS/BiCMOS	BiCMOS	GaAs
地址空间	32 位	32 位/64 位	64 位
规模	VS (Very Scale)	US (Ultrascle)	GS (Gigascale)

* 指令级并行处理主要指超标量, 超流水线和超长指令字 VLIW

1.4.2 采用并行处理技术

90 年代大部分 RISC 微处理器是通过简单高效的指令系统, 固定长度的指令格式, 简洁的硬件连线控制逻辑等技术而获得单周期执行指令的性能。但事实上, RISC 的 CPI (每条指令周期数) 一般只能做到 1.2~1.7。为了使 $\text{CPI} < 1$, 而产生了在一个时钟周期内执行多条指令的并行处理技术, 如超标量 (superscalar)、超流水线 (superpipeline) 和超长指令字 VLIW (Very Long Instruction Word)。超标量和 VLIW 都是采用多指令在多处理部件中并行处理的结构, 因此 CPI 可做到小于 1。而超流水线则通过增加流水线级数, 一般从通常的 4 级增加到 8 级, 使 RISC 处理性能提高。

80860 微处理器采用 VLIW 并行技术在 40MHz 的工作频率下, 处理速度高达 40MIPS 或 80MFLOPS。intel 80960CA 微处理器和 intel Wrap 微处理器分别采用超标量和 VLIW 的并行技术, iWrap 芯片每个时钟周期执行 9 条指令, 可用于信号处理的并行计算机中。

90 年代开发的高性能 RISC 微处理器都将采用超标量或超流水线并行处理技术。由于 VLIW 虽然有更高的并行处理能力, 但与早期开发的 RISC 微处理器不容易兼容, 而且对编译程序要求高, 所以大多数公司推出的大都是采用超标量技术的产品。典型的产品有 IBM 公司的 RISC/6000, Motorola 公司的 MC88110, Intergraph 公司的 C-400 和 AMD 公司的 AMD29050 等。而 MIPS 公司的 R4000 则采用超流水线的并行技术。

1.4.3 支持多处理机技术

超标量、VLIW 和超流水线等结构都是在单处理器中的指令级并行技术，它们受器件在特定时间里制造精度和工作频率的限制。目前采用 CMOS 工艺的 RISC 芯片可做到 40MHz，而用 ECL 工艺也只能做到 60MHz（如 MIPS 的 R6000）到 80MHz（BIT 的 SPARC B5000）。估计 90 年代中期可实现把双处理器集成在一个芯片上；到 90 年代末期可把 4 个处理器集成在一个芯片上；近期将在新开发的 RISC 微处理器产品上增强多处理功能。

绝大多数 RISC 微处理器构成的多机系统都将采用对称多处理 SMP（Symmetric Multiprocessing）技术。因为在 SMP 系统中，处理器有自己的局部存储器（通常是 Cache）、共享公共主存、外设和一个拷贝的操作系统。微处理器之间通过共享 Load 进行信息交换，速度快，因而能适应微处理器更新、升级换代的要求。由于采用大量廉价的微处理器成本低、性能高，同时还有多处理器的支持，因此 SMP 将逐步成为一个性能价格比较高的标准多处理机结构。

目前已推出具有高速处理性能的小小型机和服务器产品。例如：美国以生产小小型机而得名的 Alliant 公司的 FX/2828 多处理机，由 28 个 40MHz 的 intel 80860 RISC 芯片构成，整数处理速度为 672MIPS，双精度浮点运算能力为 1.1GFLOPS CPU 与 Cache 之间的传输速率为 1.2GB/s，主存与 Cache 之间的传输速率也高达 640MB/s，操作系统为 Concentrix，并与 Unix4.3BSD 和 Unix system V4.0 兼容，还支持 Ethernet，FDDI，HPPI（High performance Parallel Interface）等网络标准和 VME 总线标准。

又如美国以生产图形工程工作站而闻名的 SGI 公司的 Iris4D/280s 多处理机，由 8 个 25MHz 的 MIPS R3000 RISC 芯片组成，整数处理速度为 160MIPS，浮点运算能力为 28MFLOPS，每秒能描绘由 10 象素组成的三维向量 100 万个，主存传输速率为 100MB/s，操作系统为 Irix，也与 Unix system V4.0 兼容。

又如 intel 公司的 iPSC/860 多处理机，是一个具有强大处理功能而又可缩放的（scaleable）计算机系列产品。根据所用 i80860 RISC 芯片数量的不同（8、16、32、64、128 个），组成五个型号的系列品种，其处理速度为 480MFLOPS~7.6GFLOPS。它可在很广的范围内升级，还可集成到高速计算机网络中，它支持网络队列服务 NOS（NetworkQueueing Services）、TCP/IP、Ethernet、NFS 和 intel 的 VMSLINK。iPSC/860 的最高处理性能已达到巨型机 CRAY-Y-MP 的水平，而成本仅为其 1/10。

因此，90 年代采用 RISC 微处理器构成的多机系统将是一个重要发展方向。为了支持 SMP 功能，很多公司都在修改自己的 UNIX 操作系统以适应 SMP 环境。两大 UNIX 组织 OSF 和 U1 也将公布支持 SMP 的操作系统版本，很多计算机公司都推出了 SMP UNIX 系统，如 DEC 的 VAX6000 系统、DG 的 Aviiion 系列、Pyramids 的 Masever，SGI 的 Power 系列等。其它像 HP、AT&T NCR、Sequent 计算机系统公司、Encore 计算机公司等都已推出或即将推出 SMP UNIX 系统。

1.4.4 多级 Cache 存储器结构

历来计算机体系结构的革新往往都是围绕着快速 CPU 和慢速主存之间速度的差异进行的。对 RISC 微处理器的结构来讲, 仅有 Load/Store 指令才访问主存, 通过 Cache 与 CPU 中的寄存器堆进行寄存器与寄存器之间的高速运算, 但是存在着如何保证一条 Load 指令返回的结果与最近的 Store 指令的给出的相同地址结果的一致性问题, 即 Cache-coherence。通常 CPU 平均指令执行时间 T 可表示成: $T = T_{cp} + M(A + W)$ 。

这里, T_{cp} —Cache 命中的时钟周期时间, M —指令访问 Cache 未命中的次数, A —主存存取时间, W —存储器竞争的等待时间, 最理想的情况是 $M=0$, 即指令全命中, 但这是不可能的。为了尽量减小 M 就需要增加 Cache 容量。但考虑到高速 Cache 的成本高, 往往采用二级或多级 Cache, 以减轻对高速 Cache 容量的要求, 这样可提高 Cache 的命中率以达到 CPU 的高速处理能力, 为了保证 Cache 的一致性还必须考虑 Cache 的组织结构、替代算法和写控制策略。

在单 RISC 微处理器中, 采用统写及同时更新 Cache 和主存, 基本上可实现 Cache 的一致性。但对处理器系统, 就要同时更新所有处理器的 Cache 内容以保证 Cache 的一致性。同时还要考虑进程的动态调度和 I/O 共享系统总线的复杂关系。因此多处理机系统中的 Cache 一致性问题就更加复杂。目前, 对由数量不多的微处理器构成的共享主存的紧耦合多机系统, 大都采用总线动态监视功能 (bus snooping)。而对较多数量微处理器构成的多机系统则采用软件控制方式或软硬件结合方式, Motorola 公司的 RISC 微处理器 MC88000 就是用总线动态监视硬件功能支持多处理器系统的 Cache 一致性问题, 它大大减轻了编程负担且对软件透明。intel 公司的 i80860 RISC 微处理器和 MC88000 都支持软件控制方法。

另外 NCR 公司采用基于目录 (Direction-based) 原理, 通过存储器系统的标志信息和消息传送 (Message passing) 的通信协议来获得 Cache 一致性的方法, 它已用于 NCR 的基于 UNIX 系统 V4.0 操作系统的 NCR 对称多处理机系统中。由于多处理机的 Cache 一致性是直接影响系统性能的核心技术, 因而受到计算机厂商的重视, 可以肯定多机系统的 Cache 一致性技术将会得到进一步的发展。图 1-2 给出了单 RISC 和多 RISC 处理机的存储结构。新一代 RISC 微处理器大都采用二级或多级 Cache, 单 RISC 微处理器 MIPS R6000 就是采用两级 Cache 控制, 多 RISC 处理机将同样采用外扩 Cache 的两级 Cache 结构, 这是一种经济有效的方法。

1.4.5 RISC 在嵌入式控制系统中的应用

长期来, 嵌入式控制系统都采用 MC68000 或类似的 CISC 微处理器芯片。由于嵌入式控制系统的日益复杂, 对外界事件的接收和处理中断的响应时间要求也越来越高, 编程工作量相应增加, 精度要求也越来越高, 因此, 迫切需要高速的 32 位微处理器, 而 RISC 微处理是它的最佳选择。目前, 美国空军已选用 MIPS R3000 作为其指挥系统的微处理器; 美国海军选用 AMD 29000 RISC 微处理器用于它的 C³I 系统中。另外还有很多

RISC 芯片生产厂商正在积极投入人力物力开发用于嵌入式控制器的微处理器产品。如 Intel 公司 Intel80960 系列, SGS—Thomson Microelectric 公司的 Transputer 以及 IDT 公司的 R3051 / 3052 微处理器等, 表 1.2 给出了嵌入式控制所用的 CPU 的发展趋势。

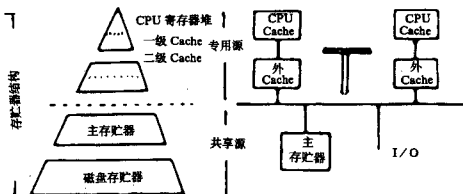


图 1-2 单 RISC 和多 RISC 处理机的存储结构

根据统计和预测, 全世界 RISC 微处理器在嵌入式控制器中的用量, 1989 年有 61 万个, 占总用量的 12%; 1990 年有 150 万个, 占总用量的 30%; 1991 年将有 335 万个, 占总用量的 45%, 1992 年将有 603 万个, 占总用量的 73%。图 1-3 给出了 RISC 微处理器应用的销售金额。

表 1.2 嵌入式控制所用 CPU 的发展趋势

年代	1970 年代	1980~1987 年	1988~1991 年
CPU 性能	0.1MIPS	1MIPS	5~100MIPS
CPU 结构	CISC	CISC	RISC
存储器容量	4~64KB	~1MB	~64MB
开发语言	Assemble	C 语言	C 语言 / Ada
通信手段	RS-232C	RS-232C, ETHERNET	ENTE

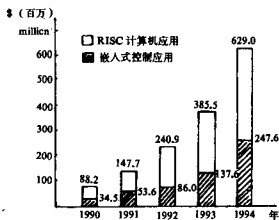


图 1-3 RISC 微处理器应用的销售金额

RISC 微处理器除用于工作站和其它各类计算机外, 嵌入式控制器的应用是十分广阔的市场, 在相当长的一段时间里, 嵌入式计算机 (Embedded Computer) 是指被嵌入到武器系统、战术系统、飞机、舰船、导弹、宇宙飞船和某些指挥与控制系统等的数据处理机, 而现在范围扩大到激光高速打印机、声音处理和通信、DMA 控制器、机器人控制器等方面, 这是一个值得重视的应用领域。

90 年代, RISC 将成为微处理器发展的主流技术, 今后 RISC 微处理器将必须采用更微细的半导体加工技术来提高芯片的工作频率和集成度, 并尽量集成更多的功能部件于同一芯片上, 如整数处理部件 (IPU), 浮点处理部件 (FPU), 图形处理部件 (GPU) 存储管理部件 (MMU), 高速缓冲存储器 (Cache) 和通信链路部件 (CLU) 等, 利用超标量、超流水线和 VLIW 的并行技术实现高速、高性能的新一代 RISC 微处理器的体系结构。

90 年代刚刚开始, 全世界各主要计算机半导体厂商纷纷在各自第一代 RISC 微处理器系列产品的基础上, 推出新一代 RISC 处理器产品, 令人瞩目的有 MIPS 公司的 R6000、R4000; Sun 公司的 SPARC 结构的 MB86903; MOTOROLA 的 MC88110; intel 公司的 intel 80960; Intergraph 的 C400; AMD 公司的 AMD29050; IBM RISC / 6000 以及 SGS-THOMSON 的 Transputer H1 和 E1 型芯片等。

除上述新一代 RISC 微处理器产品外, 还有 HP-PA 型 RISC 微处理器, 工作频率已达 90MHz, 用其构成的 HP3000 / 980 计算机, 处理速度达 50MIPS ~ 100MIPS, 集成度为 47.9 万个晶体管, 功耗为 6.3w。

随着 RISC 微处理器的广泛应用, 可以肯定在今后几年里将会推出更高性能的, 高集成度的 RISC 微处理器产品。

RISC 技术缩短了计算机的设计周期, 提高了设计可靠性, 还使机器的性能价格比大大超过以往的 CISC。这些优良特性充分显示了 RISC 的强大生命力, 作为一种新兴事物, RISC 技术将会继续丰富和发展, 还有许多新技术在等待人们去发掘。

第二章 典型的 RISC 微处理器

由于 RISC 具有指令简单; 指令长度固定; 大多数指令都在一个机器周期里完成; 只有存取与取数指令才访问存储器; 其余指令均在寄存器之间高速执行; 转移指令采用延迟转移方式, 以简单有效的方式支持高级语言, 不专门面向某种编程语言等特点, 这样大大提高了计算机的系统速度。因此, RISC 微处理器的发展格外迅速, 品种不断增加, 现介绍几种典型的 RISC 微处理器产品, 它们有 Sun 公司 SPARC; Motorola 公司的 MC88000; Intel 公司的 80860; IBM 公司的 RS/60C3; INMOS 公司的 SPARC Transputer T414 和 T800 以及 MIPS 公司的 R2000/R3000/R4000/R6000 等。

2.1 Sun 公司 SPARC RISC 微处理器

Sun 公司 SPARC (Scaleable Processor Architecture) RISC 微处理器是 1986 年推出的 32 位 RISC 微处理器, 其结构基于帕克莱分校的研究成果而设计的, 它是一种可定标处理器体系结构, 它强调的是可应用于各类计算机系统中, 采用开放式策略, 允许众多生产厂商生产其芯片, 且可按不同制造工艺, 不同精度生产 SPARC 芯片, 同时还提供丰富的软件, 深受用户欢迎, 像世界著名的 Cypress、LSI 逻辑、Fujitsu、TI、BIT 等半导体公司都生产 SPARC 芯片。

SPARC 芯片基本上是一个 32 位整数执行部件, 可以直接连到分离的浮点处理部件。整数处理部件中有大量寄存器存储区, 采用 4 级流水线技术以便使大多数指令在一个时钟周期内完成, 外部存储管理部件用以提供虚拟存储器管理, 在 SPARC 结构中, 整数和浮点部件的操作是并行的, 整数执行部件由指令流中取出需加浮点运算的操作, 并排队等候浮点处理部件来处理, 而浮点单元则通过一组浮点算术部件来执行浮点运算。图 2-1 是 SPARC 实现的方块图。

2.1.1 SPARC 的体系结构

SPARC 是基于帕克莱分校的 RISC I, RISC II 的体系结构, 在 CPU 中有 120×32 位寄存器堆作为工作存储器, 其中 8 个寄存器 R0~R7 作为全局存储器, 可用于所有时间里, 其余 112 个寄存器是设计成每个有 24 个寄存器构成的七个重叠窗口, 一个 24 个寄存器窗口在任何给定时刻内可以工作, 在每个窗口中, 8 个寄存器 R8~R15 是用于输出参数到下个进程, 8 个窗口 R15~R23 是为当前进程用作局部存储, 最后 8 个窗口 R24~R31 提供给来自前级进程的输入参数, 所有窗口是链接成一个循环链, 实际上, 一个窗口的输入寄存器就是前级窗口的输出寄存器, 在链中, 输出寄存器提供给下个窗口的输入, 因此, 当窗口 0 被选定时, 则窗口 6 的寄存器就是窗口 0 的输入寄存器, CPU 的状态寄存器包含当前窗口指针字段 (CWP), 在任何给定时刻, CWP 允许处理器保持跟踪那个有效窗口, 系统操作相当存储器堆栈结构, 所以, 当一个新进程被调用时, 窗口号将

递减, 而从该进程返回的窗口号将递增以便重存前级窗口, 寄存器的操作在单时钟周期内, 可读 2 个寄存器和写入一个寄存器。图 2-2 是 6 个寄存器窗口的 SPARC 的结构, 大型寄存器窗口的应用, 使处理器通过对大量重叠组形式出现的寄存器的访问, 编译程序可自动在寄存器内贮存数据和传递参数。

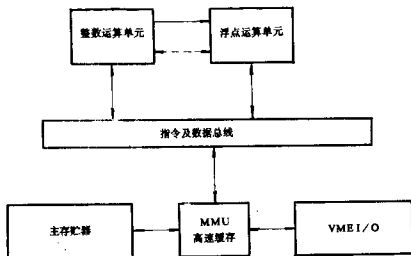


图 2-1 SPARC 实现的方块图

2.1.2 SPARC 的指令系统

SPARC 的指令有 64 条基本指令, 只有 Load/Store 指令可访问内存, 除 Load/Store 指令和浮点指令外, 所有指令皆可在一个机器周期内完成, 指令的操作在寄存器之间进行, 所有指令皆为 32 位宽, 且字与字之间在存储器内对齐, 仅有三种指令格式, 操作码与地址码在指令中的位置固定, 整数执行部件可实现五种类型的指令:

- 取/存 (Load/Store) 指令;
- 算术/逻辑/移位指令;
- 控制转移指令;
- 读/写控制寄存器指令;
- 其它指令 UMHP (Unimplemented) 和 IFLUSH (Instruction Cache Flush)。

SPARC 结构还定义了一组浮点指令, 在整数执行部件控制下由浮点处理器完成。另外, 还可以有一组由用户定义的协处理器指令 (CP), 它们也是在整数执行部件控制下实现。除了一般的算术运算指令外, SPARC 还包含带标志的算术运算指令, 它们操作数的最低两位称为标志 (Tag)。当它不为“0”时, 置条件码。人工智能语言用它来表明操作类型, 从而使 LISP 和 Prolog 语言运行得更有效。在 Load/Store 类指令中, 有两条支持

处理器操作的指令，它们是 LDSTUB (Atomic Load/Store unsigned byte) 和 SWAP (Swap Register with Memory)。LDSTUB 从存储器读出一个字节送入整数执行部件的寄存器堆中，并向存储器的同一单元写入一个全“1”字节。SWAP 在存储器与寄存器堆之间交换一个字。这两条指令在执行过程中不允许中止或打断。

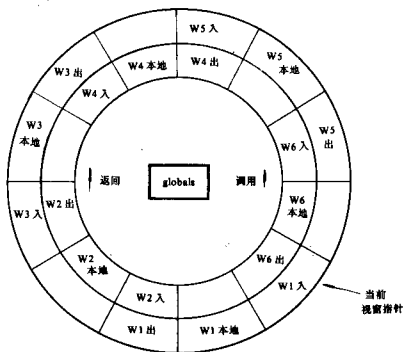


图 2-2 六个寄存器窗口的 SPARC 结构

2.1.3 流水线技术

在执行指令时，SPARC 采用 4 级流水线，即取指令 (Fetch)，译码 (Decode)，执行 (Execute) 和写结果 (Write back)。在正常操作中，处理器将在每个时钟周期内接收一条新指令，这样以确保每个时钟周期执行一条指令，在执行阶段的操作结果保存在寄存器中，而在写回阶段则把结果写入目的寄存器中。数据和指令 Cache 是合一的，由系统管理，因此 Load 指令需 2 个机器周期时间，而 Store 指令则需 3 个机器周期时间，典型的产品 MB86901 由日本富士通生产的，在 $1.2\mu\text{m}$ CMOS 工艺条件，25MHz，处理速度高达 15MIPS 或 3.3MFLOPS，Cypross 的 Cy601，采用 $0.8\mu\text{m}$ CMOS 工艺技术，33MHz 工作频率，处理速度高达 20MIPS 或 4MFLOPS。