第 17 卷第 4 期
2004 年 12 月

海南师范学院学报(自然科学版)
JOURNAL OF HAINAN NORMAL UNIVERSITY(NATURAL SCIENCE)

Vol. 17 No. 4
Dec. 2004

文章编号:1671 - 8747(2004)03 - 0347 - 05

超标量微处理器研究

莫壮坚[1],李 振[2]

(1. 海南师范学院 计算机科学与教育技术系,海南 海口 571158;
2. 西北工业大学 航空微电子中心,陕西 西安 710072)

摘 要:介绍了超标量体系结构的基础,研究了超标量体系结构的基本特点和性能分析,探讨了超标量体系中遇到的数据相关问题以及解决相关问题的 Tomasulo 算法.

关键词:超标量;微处理器;Jomasulo 算法

中图分类号:TP3 **文献标识码**:A

为了提高微处理器的性能,人们在流水线技术的基础上提出了超标量(Superscalar)方法. 超标量是指在 CPU 中有一条以上的流水线,并且每时钟周期内可以完成一条以上的指令的技术. 超标量体系结构很早就出现了,而且在高性能的通用微处理器中也得到了普遍应用,例如 Intel Pentium 系列处理器[Intel95]和 IBM PowerPC 系列处理器[IbmPc97]. 本文介绍了超标量体系结构的基础,深入研究了超标量体系结构的基本特点和性能分析,详细地探讨了超标量体系中遇到的数据相关和结构相关的问题以及解决相关问题的 Tomasulo 算法及计分牌算法.

1 超标量体系结构基础

早期的单发射结构微处理器的流水线设计目标是做到每个周期能平均执行一条指令,但这一目标不能满足处理器性能增长的要求,为了提高处理器的性能,要求处理器具有每个周期能发射执行多条指令的能力. 与超长指令字(VLIW, Very Long Instruction Word)结构的数字信号处理器相似,超标量结构的处理器每个时钟周期也并行发射和执行多条指令. 区别在于,VLIW 的处理器采用静态调度,而超标量结构的处理器采用动态指令调度,在指令执行时根据资源、数据相关等情况,决定是否并行执行指令. 超标量结构是当代多发射微处理器所广泛采用的微体系结构.

在超标量微处理器中,每个周期可同时发射执行多条指令,但指令的高发射频率也意味着相关[1]所发生的频率也很高;而且其结构决定了相关的复杂性. 因此,相关的检测和解决策略的优劣将直接影响超标量处理器的性能. 为了有效地处理相关,需采用静态和动态调度技术相结合的方法. 静态调度可在编译过程中减少相关的产生;而动态调度可根据处理器的动态信息,发掘出更多的 ILP. 动态调度简化了编译器的设计,减小了编译代码对硬件的依赖,但却是以大量的硬件开销为代价的.

在超标量处理器中,指令的发射策略是指在指令的发射过程中,所采取的相关检测方法和

收稿日期:2003 - 12 - 28
作者简介:莫壮坚(1974 -),男,海南万宁人,助教,主要从事计算机应用方面的研究.

相关处理措施,决定指令队列中指令的发射顺序,其算法效率的优劣将直接影响超标量处理器的性能.设计高效的指令发射策略是实现高性能超标量处理器的前提.在超标量处理器中,由于指令乱序执行,判断指令执行结束的条件更为复杂,因此设计超标量处理器时,应考虑采取一定的措施以实现精确中断[1].

2 超标量微处理器的结构

超标量微处理器每个周期可同时发射执行多条指令,图 1 所示为超标量微处理器的一般结构.

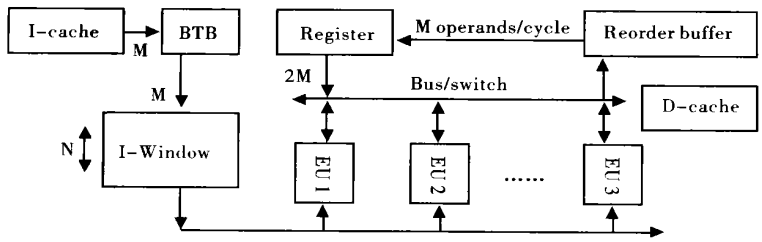


图 1 超标量处理器的一般结构

在超标量微处理器中,指令由指令 Cache 取回放入指令窗口中等待发射.每个周期可发射的最大指令数,即发射频率称为超标量微处理器的度(degree of superscalar)[2].图 1 所示的超标量微处理器的指令窗口为深度 N,度为 M,且 N(M)度是超标量微处理器中的一个很重要的概念,是衡量超标量微处理器性能的一个重要指标.为了增加发射指令的组合,执行单元的数目比处理器的度要大,如 Pentium,HP PA - 8000 等.图 1 中所示的超标量处理器结构中其执行单元数与处理器的度一致.

由于超标量处理器中各执行单元所处理的指令类型不同,指令执行所需的操作步骤也不相同,因此通常在超标量处理器中,各执行单元的流水线组织不相同,这种超标量结构称为非统一的超标量结构(nonuniform superscalar).相应地,各执行单元流水线级数相同的超标量结构称为统一的超标量结构(uniform superscalar).几乎所有的当代超标量微处理器都采用非统一的超标量结构.图 2(b)为一非统一的超标量结构,图 2(a)为相应的标量结构.

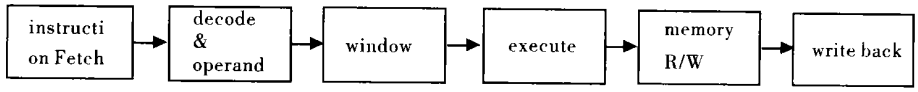
在微处理器中,由于每个周期要同时执行多条指令,需要增大寄存器文件的频宽.在最坏情况下,对于度为 M 的微处理器,寄存器文件的访问端口需设置为 3M.

超标量微处理器对指令 Cache 的设计提出了更高的要求,通常有两种方法来提供快速的指令访问:其一是使用宽字的 Cache,其二是使用高速的指令队列.同时处理器对数据 Cache 的速度要求也很高,在 Cache 不失效的条件下,要求在一个周期内提供操作数;但由于数据 Cache 同指令 Cache 相比每周期的访问频率相对较低,因此可采用交叉 Cache 结构来实现数据 Cache.

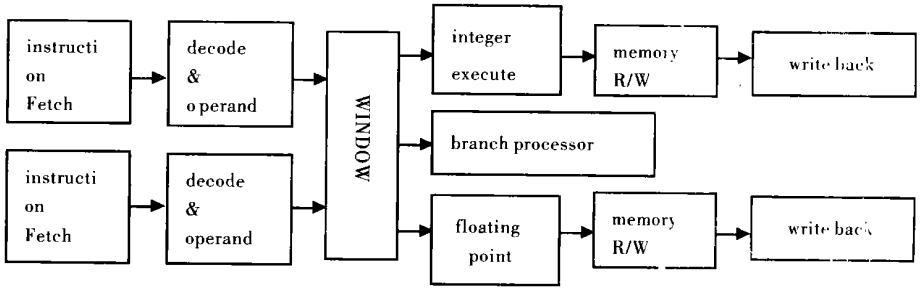
3 标量体系结构的性能分析

为了分析超标量微处理器同相应的标量处理器的性能增益情况,以及影响超标量微处理器性能的因素,可以采用基于超标量微处理器流水线的预约表,用来表示建立超标量微处理器的性能模型.图 3 为一度为 2 的统一结构的超标量处理器的预约表表示.

为了简化超标量微处理器性能模型的建立,认为:处理器的结构为统一的超标量结构,流



(a) 基准的标量结构



(b) 非统一的超标量结构

图 2　非统一的超标量结构

水线的级数为 n,度为 σ时钟周期为相应标量处理器的时钟周期,流水线每级的操作延迟为 1 个周期,则处理器处理连续 k 条指令所需的时钟周期数为:SScycles = $n + k/\sigma - 1$.

每条指令的平均执行周期数 CPI 为 CPI(SS) = $1/\sigma + (n - 1)/k$. 超标量微处理器同相应的标量处理器的性能加速比为 SSspeedup = $\frac{n + k - 1}{n - 1 + k/\sigma}$ 当 $k \to \infty$, speedup $\to \sigma$ 时,若 $\sigma \to \infty$,则 speedup $\to 1 + k/(n - 1)$.

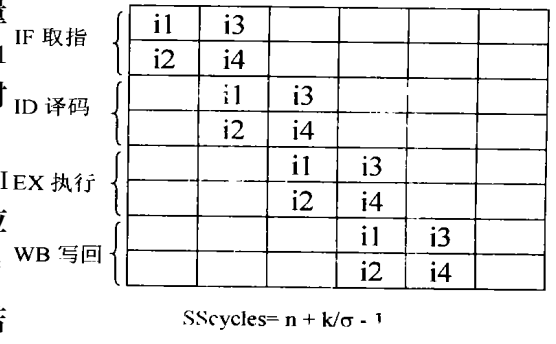


图 3　度为 2 的超标量处理器的预约表表示

由此可见,只要程序中可开发的 ILP 足够高(即可处理的指令序列足够长),则增加超标量处理器的度就可提高处理器的性能;另一方面,由于相关的影响,使得程序中可开发的 ILP 有限,则增加超标量处理器的度并不能明显提高处理器的性能. 为了充分发挥和提高超标量处理器的性能,应从解决相关、开发程序中的 ILP 入手.

4　标量体系结构的相关问题

在超标量体系结构中,由于每个周期发射多条指令会使控制相关在每次发射中出现的频率高于标量处理器,而且频率增加的比率大致与处理器的发射频率成正比. 当发射频率增加时,数据相关发生的频率也会上升. 同时由于指令在多条流水线中并行执行,相关会发生在将要发射的指令之间或将要发射的指令与正在执行的指令之间. 因此,相关出现的情况要复杂得多. 在超标量微处理器中有三种情况的相关:数据相关、控制相关、结构相关. 本文将只讨论超标量微处理器中数据相关和控制相关的有关问题.

4.1　**数据相关**　在超标量体系结构中存在三种数据相关:RAW 相关、WAR 相关、WAW 相关.

RAW 相关又称真相关(true dependence). 若两条指令之间存在真相关,则两条指令不能同

时或完全重叠执行,因此真相关在一定程度上限制了程序中 ILP 的开发.避免真相关发生的基本方法是采用代码调度技术,硬件和软件技术都可实现.

WAR 相关(又称反相关)与 WAW 相关(又称输出相关)这两种相关的产生都是由于指令的乱序执行所引起的.其中,将输出相关作为引起指令冲突的原因有些不合理,这是因为若指令 i 产生的结果没有被后面的指令引用,而被后面的指令 j 覆盖,则指令 i 实际上是一条冗余的指令,应该由编译器来在编译优化时消除.因此,输出相关一般仅存在于代码优化很差的环境下.若编译器能保证输出相关不发生,则可获得更高的 ILP.

三种相关中,真相关是在一条指令的源操作数依赖于前一条指令的结果的条件下发生的.真相关的解决必须有操作数的流动,出现相关的两条指令不能同时或完全重叠执行.而对于反相关和输出相关,相当于是由于资源问题所引发的相关,其实质是由于使用了共同的寄存器资源保存了不同的变量值所引发的,若采用更名的方法用两个不同的寄存器来保存这两个变量,则出现相关的两条指令就可并发执行,不会引起性能损失.因此反相关和输出相关又称为伪相关(false dependence),可通过更名来解决.更名可以通过软件和硬件的方法来实现.

5 解决数据相关的动态调度策略

在超标量体系结构中,数据相关的解决可通过两种途径:静态调度和动态调度[1].静态调度由编译器将相关的指令尽量调度开来,以减少相关的发生和相关所造成的性能损失;动态调度由硬件重新安排指令的执行顺序,允许指令乱序执行,以减少相关所引起的性能损失.动态调度一方面可处理编译过程中无法预知的动态相关问题,另一方面将有助于简化编译器的设计、降低编译代码对硬件的依赖程度,但是以增加硬件开销为代价的.动态调度是指根据处理器的运行状态,发射指令到其相应执行单元的过程.根据动态调度过程中相关的检测时机的不同,动态调度分为两种方法:控制流调度方法和数据流调度方法.在控制流调度方法中,数据相关和结构相关的检测在译码级集中进行,若出现相关,则指令停止发射;在数据流调度方法中,指令在译码级被发射至相应执行单元的缓冲区中,相关的检测将以分散的方式在各执行单元的缓冲区中进行,若指令操作数及其相应执行单元有效,则分配指令到相应执行单元中执行.

最为常用的数据流调度方法是 Tomasulo 算法,Tomasulo 算法的主要思想是,将记分牌方法同基于预约站的寄存器更名方法有机的结合在一起,有效地避免了 WAW 和 WAR 相关[3];同时基于预约站提供了分散的相关检测和指令执行控制.在 Tomasulo 算法体系结构中(图 4),预约站用于缓存发射的指令及其操作数以等待指令的执行.当指令发射到预约站时,若源操作数有效,则从寄存器文件中读取源操作数送入预约站中缓存;否则以产生源操作数的预约站标识替换源寄存器标识.这一过程成为"寄存器更名":更名使每个寄存器对应的预约站标识为最后更改寄存器的预约站标识,更名消除 WAW 相关和 WAR 相关.

每个执行单元之前都设置了多个预约站,每个预约站有 6 个域:Op —指令执行的操作;Busy —表明预约站和相应的执行单元被占用;Qj —产生源操作数的预约站标识,为 0 则表示源;Qk —操作数已在 Vj,Vk 域中,或不需要操作数;Vj —源操作数的值,每个操作数中只有 V;Vk —域或 Q 域其一有效.

寄存器堆中每个寄存器都有一个 Qi 域:Qi —产生结果的预约站标识,为 0 则表示寄存器内容有效.

在 Tomasulo 算法中,设置了 Load 缓冲区和 Store 缓冲区用于保存访问存贮器的地址或数

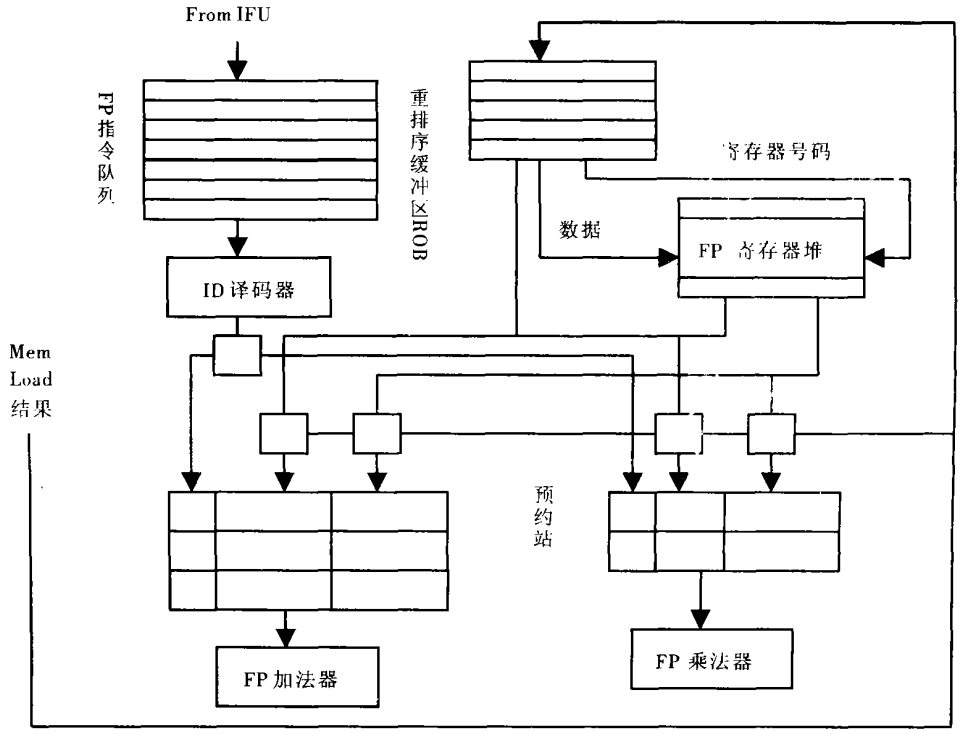


图 4　Tomasulo 结构示意图

据.

Store 缓冲区中每个缓冲器都有一个标识域:Qi —产生结果的预约站标识,为 0 则表示无指令计算结果用于存贮在该缓冲器中;V —用于存贮的操作数值;Busy —指示该缓冲器是否被占用;Address —访存地址.

Load 缓冲区中每个缓冲器都有一个标识域:Busy —指示该缓冲器是否被占用;Address —访存地址.

当指令发射时,若相应执行单元的预约站有空闲的位置,则发射指令到对应位置,更新预约站及相应寄存器的标识域;否则,因为预约站已满,就出现了结构相关,需要停止指令发射. 若指令为 load/store 指令,则更新相应的 load/store 缓冲器的标识域. 这一阶段的操作执行了寄存器更名,避免了 WAR 和 WAW 相关的发生. 在 Tomasulo 算法中,所有执行单元的结果和访问存贮器的结果都送入公共数据总线(CDB —Common Data Bus). 在预约站中缓存的指令,若源操作数无效,则监视 CDB,等待操作数有效后,放入相应的预约站中. 当所有源操作数有效且执行单元有效,则分配指令到相应的执行单元执行. 这一阶段的操作动态解决了 RAW 相关. 当执行单元执行结束后,将结果送入 CDB,该结果从 CDB 送入相应的寄存器或预约站中.

Tomasulo 算法的特点总结如下:①相关的检测和指令执行的控制是分散进行,由每个单元的预约站分别完成指令相关的检测和控制指令的执行;②采用了基于预约站的寄存器更名避免了 WAR 和 WAW 相关的发生,可动态解决 RAW 相关;③执行单元的结果通过 CDB 送入等待操作数的预约站中,同时送入相应的结果寄存器中;④将 Load 和 Store 作为基本的功能单元;⑤Tomasulo 算法实现指令动态调度,硬件逻辑开销很大.　(下转第 355 页)

[6] Shabalovskaya S A ,Anderegg J W. Surface spectroscopic characterization of TiNi nearly equiatomic shape memory alloys for implants[J]. J Vac Sci Technol ,1995 ,A 13(5) :2 624 - 2 632.
[7] Lu Y M ,Huang W S ,Yang J S. Effects of substrate temperature on the resistivity of non-stoichiometric sputtered NiO films[J]. Surface and Coatings Technology , 2002 ,155 :231 - 235.
[8] Wever D J ,Veldhuizen A G,Vries J D ,et al. Electrochemical and surface characterization of a nickel-titanium alloy [J]. Biomaterials ,1998 ,19 :761 - 769.

The effect of surface oxidation of NiTi alloy on nickelion release

HUA Ying-jie , WANG Chong-tai
(Department of Chemistry , Hainan Normal University , Haikou 571158 , China)

Abstract :Atom adsorption spectometer was used to study the release rate of mechanically polished and oxidized NiTi alloy in the simulated body liquid. The result shows that the oxidation film on NiTi alloy surface can reduce the release of nickelion effectively.

Key words :NiTi shape memory alloy ; nickel ; simulated body liquid

(上接第 351 页)

因此在设计超标量处理器时,若设计目标是实现高性能的超标量处理器,则建议采用的动态调度策略为 Tomasulo 算法;这一建议可从当代的多数高性能超标量处理器都采用基于 Tomasulo 算法的动态调度策略来得到证实.

参考文献:

[1] 金正谊,白英彩. 流水处理中转移开销的分析[J]. 小型微型计算机系统 ,1992 ,(10) :11 - 22.
[2] 李学干 ,苏东庄. 计算机系统结构[M]. 西安 :西安电子科技大学出版社 ,1991.
[3] 李　涛 ,高德远 ,樊晓桠. 高性能微处理器性能模型设计[J]. 航空电子技术 ,2001 ,(2) :25 - 28.

Studies on superscalar microprocessors

MO Zhuang-jian[1] , LI Zhen[2]
(1. Department of Computer Science and Educational Technology , Hainan Normal University , Haikou 571158 , China ;
2. Aerial Microelectron Center , Northwest Industrial University , Xi'an 710072 , China)

Abstract :In this paper , the basics of the superscalar microprocessor structure are introduced ,followed by a disussion on the characteristics of its structure and the analysis of its performance. Besides , this paper focuses on the data dependence and control dependence and explains the Tomasulo algorithm and scoreboard algorithm specifically to solve relevant problems.

Key words : superscalar ; microprocessors