

40-42

VLIW 体系结构微处理器的一种设计方法^{*}

A Designing Method of VLIW Architecture Microprocessor Processor

王昭顺 张建林 曹文彬 TP332.02

(首都师范大学计算机系 北京100037)(北京科技大学计算机系 北京100083)

Abstract In this paper, based on analyzing structure features, technical advantage and key technical of VLIW, we point out and discuss some critical issues required to solve under VLIW architecture. Then we put forward a method to solve these issues in designing a microprocessor model schema.

Keywords Microprocessor architecture, VLIW, EPIC, Code structure rules

微处理器体系结构的发展经历了三个不同的阶段,以 Intel 早期 X86 产品为代表的 CISC 体系结构微处理器;以 MIPS、PA-RISC、SPARC、ALPHA、PowerPC 等为代表的 RISC 体系结构微处理器;以 Intel 近期产品为代表的 CISC-RISC 混合型体系结构微处理器。RISC 和 CISC 由于其实现技术的复杂性,逐步退出历史的潮流,国内外关于微处理器体系结构的研究表明,新一代微处理器体系结构将是 VLIW 一统天下。

VLIW (Very Long Instruction Word, 超长指令字)体系结构是一种让多条指令进入流水线执行的微处理器体系结构,借助编译器将能同时发送的操作安排在一个很长(一般为128~1024位)的超长指令字中。

1 VLIW 体系结构的特征分析

微处理器体系结构从 CISC 到 RISC 的发展是一个飞跃。RISC 技术不仅突破了 Flynn 瓶颈,还实现了指令级并行(ILP)。但 RISC 体系结构微处理器的内部控制逻辑也越来越复杂,开销也越来越大。Fisher 从水平微码得到启示,提出超长指令体系结构(VLIW)^[1],希望它既具有支持较高的指令级并行性的能力,又能够用较为简单的控制逻辑实现。但由于 VLIW 相关技术的不成熟,VLIW 体系结构微处理器至今未能占据通用 CPU 芯片市场,只在科学计算和信号处理等专用领域占有一席之地。但是,随着 Intel 下一代产品 IA-64 (或称作为 Merced)芯片的出现,有专家预测 VLIW 体系结构微处理器将开拓微处理器新时代^[2],成为微处理器体系结构发展的又一个飞跃。

•VLIW 的结构特征 VLIW 体系结构微处理器具有如下结构特征:

- 1)长的指令字(128~1024BIT);
- 2)指令包含多个独立的并行操作;
- 3)单个操作具有确定的执行周期;
- 4)一个控制单元每周发送一条长指令;
- 5)用全局共享寄存器堆连接多个功能单元。

所述 VLIW 体系结构微处理器的指令字格式:

特征域	操作段1	操作段2		操作段 n
-----	------	------	-------	-------

其中,特征域可用于表示各个操作段的组成方式(如每个操作段在指令字中的位置),也可用于表示各个操作段的执行方式(如各个操作段规定的操作之间的串并

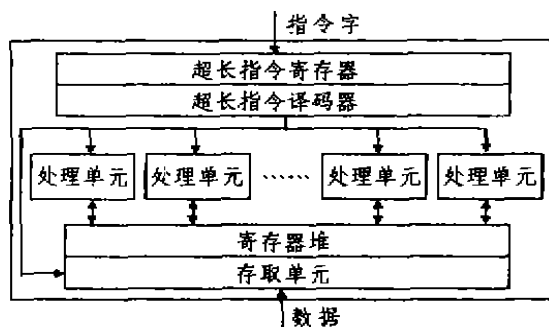


图1

行执行顺序)等。各操作段编码具体表示了要执行操作的语义。

* 本文得到教育部优秀年轻教师基金资助。

所述 VLIW 体系结构微处理器硬件结构如图1。

指令字取入后,经过译码器的译码,形成对应于各个处理单元、寄存器堆、存取单元的控制信号,同时控制数据存取、数据处理、以及寄存器传送等操作。

•VLIW 的技术优势 与采用超标量技术的 RISC 处理器相比,VLIW 微处理器具有两个方面的技术优势:

1)显式并行指令控制 EPIC(Explicit Parallelism Instruction Control)。由 VLIW 字显式表明并行操作,不需要微处理器内部复杂的指令调度,降低了微处理器的实现复杂度。

2)更大的指令级并行 IPL。现代超标量 RISC 微处理器的 ILP 一般不超过4,而 VLIW 微处理器可以有更大的 ILP,可以在10~30的范围内(随 VLIW 指令字宽不同而不同)^[1]。

•VLIW 存在的问题 VLIW 体系结构微处理器通过优化编译器静态地将可并行执行的操作安排在一个超长指令字中的各个指令段上。由于这一特点,超长指令字译码器逻辑简单,不需要操作相关判定、指令调度、操作的乱序处理等控制逻辑。但是,这一特点同时也带来了一些问题:

1)灵活性差,使得实际的操作并行性较低,资源的利用率较差。由于程序动态执行过程在编译时无法准确确定,只能通过一些预测算法将各个操作排列在超长指令字流中,且在程序执行过程中又不可能再调整各个操作的执行顺序,因此,若预测执行过程与实际执行过程不一致,将会大大增加程序执行的时空代价。

2)主存瓶颈严重,原因有两个方面,一是处理器内部处理单元较多,需要与之匹配的供数能力,从而产生对存贮器的压力;另一方面,由于指令字较 RISC 指令宽,因此,取指也增加了对存贮器的压力。

3)代码不兼容。原因是芯片中支持各个基本操作的处理单元的 Latency 直接确定了超长指令字流中各个操作之间的相对位置,因此,不但在处理单元或其他资源的数量和功能发生变化时,指令代码不兼容,而且就是在某个处理单元的 Latency 发生变化时,指令代码也不兼容。

2 VLIW 体系结构的研究现状与方法

VLIW 体系结构微处理器的优势与问题推动着 VLIW 的研究,目前主要的研究内容有:

1)VLIW 的体系模型研究,例如:IBM 公司提出基于树的 VLIW 体系模型^[3],麻省理工学院提出基于基本块的二维模型^[4];

2)基于体系模型的优化编译器研究^[3,5,6];

3)VLIW 体系结构微处理器设计的性能分析,例

如:麻省理工学院正在研究基本块个数与性能的关系等^[1]。

4)VLIW 兼容性研究,例如:IBM 的 DAISY 项目^[3]提出了一种实现二进制代码兼容的方案。

从研究环境上分析,几乎所有的 VLIW 研究都是在用程序设计语言(如:C 语言等)实现的指令级模拟环境下进行,而不是在用硬件描述语言描述的硬件模型、或物理仿真器上进行,即:利用程序设计语言描述 VLIW 微处理器的体系结构,建立一个微处理器模拟环境,该模拟环境具备所有体系特征,且可能通过参数设置进行调整、修改。在该环境下可以运行优化编译器产生的任何一个程序的可执行代码(不受程序长度的限制),并能够统计各种资源的利用率,发现体系的瓶颈。

3 VLIW 微处理器设计的相关技术与难点分析

VLIW 微处理器用优化编译器代替复杂的指令调度,降低了微处理器的复杂性,因此 VLIW 微处理器的性能依赖于优化编译器。除需要传统的 CISC 和 RISC 编译技术,以及超级计算机的向量化技术之外,VLIW 微处理器还需要有其特有的编译技术,主要有:

1)路径调度(Trace Scheduling)技术 VLIW 微处理器的特征之一是可以发挥更大的 ILP,传统超标量微处理器的优化编译器在基本块内挖掘 ILP,不能满足 VLIW 微处理器的 ILP。为发挥更大的 ILP,需要对超出基本块的代码进行 ILP 挖掘。路径调度技术首先选择出最有可能执行的操作序列——路径,将这条路径上的操作压缩在尽可能少的超指令字中。

2)循环展开(Loop Unrolling)技术 对编译时可以确定循环次数的循环,通过把循环体重复多次,不仅减少循环转移的开销,而且可以挖掘更大的 ILP。循环展开技术将循环体重复多次,并对其进行压缩,产生超指令字序列。

3)软件流水技术 循环展开将使 VLIW 目标代码体积爆炸,为解决这一问题,人们提出软件流水技术。软件流水在不改变程序语义的前提下,使循环的不同迭代中的操作重叠,以充分利用硬件资源的并行性,缩短循环程序的执行。

优化编译器生成 VLIW 目标代码的过程是将源程序的串行指令压缩成超长指令字序列的过程,即在一定的资源和操作之间数据等相关条件的约束下将串行操作序列合并为并行指令序列,以尽可能开发操作的并行性。因此,VLIW 微处理器设计的关键是如何将编译器发现的并行操作压缩在超长指令字中,即 VLIW 微处理器指令字的构造规则。

我们认为,VLIW 微处理器指令字的构造规则应遵循以下设计原则:

·**并行控制原则**:能够并行执行的操作能尽可能拼装在同一条 VLIW 指令字中。

·**高效译码原则**:VLIW 指令字的译码可以用规模小、延时小的电路实现。

·**指令吞吐率原则**:VLIW 微处理器能有效利用机器的吞吐能力,尽可能多取入需要执行的操作。

基于以上原则,VLIW 指令字的构造规则有两类:并行规则和先后顺序规则。所谓并行规则,就是满足并行控制原则和高效译码原则,即在资源不相关、数据不相关和指令字长的约束下,尽可能取入需要并行执行的操作。所谓先后顺序规则,就是满足指令吞吐率原则和高效译码原则,即在资源相关或数据相关的情况下,能充分利用指令字长资源,取入以后将要执行的操作。

在微处理器的设计中,以并行规则为主来构造指令字。当指令字不饱满时,用先后顺序规则来填充指令字的空操作段,可以采用代码填充技术实现,即当 VLIW 的指令字中有空操作段时,将后续要执行的操作编码预先取入,填充在缓冲器中,使用时直接从缓冲器取操作编码;也可以采用延时技术实现,即当 VLIW 的指令字中有空操作段时,将后续要执行的操作编码预先取入,延时若干周期执行。基于这种 VLIW 指令字构造规则,VLIW 微处理器指令格式如下:

CBFF	CONTROL	OPERATOR
------	---------	----------

其中 CBFF:指令格式形态控制域;CONTROL:操作控制域,根据 OPERATOR 的分段分为多个段,控制每个 OPERATOR 段的操作类型和执行方式;OPERATOR:操作编码域,分为多个操作段。

VLIW 微处理器硬件基本结构如图2。

图2中,程序的执行过程为:第一周期执行 I1 指令字的 F1、F3、F5 操作编码相应的操作,F2 操作编码暂存在内部缓冲器中,F4 作为延时操作编码送内部延时器;第二周期执行 I2 指令字的 F1、F2、F3、F4 操作编码和 I1 指令字的 F2、F4 操作编码,I2 指令字的 F5 操作编码作为延时操作编码送内部延时器;第三周期执行 I3 指令字的 F1、F2、F3、F4、F5 操作编码,I1 指令字的 F2 操作编码,和 I2 指令字 F5 的操作编码。

采用这种 VLIW 指令字构造规则,一方面能够充分利用 VLIW 指令字长资源,使指令尽可能饱满,缩短 VLIW 目标代码长度;另一方面可以提高 VLIW 微

处理器的 ILP 度,如图2中,指令字包含五个操作段,但第三周期执行了七个操作。

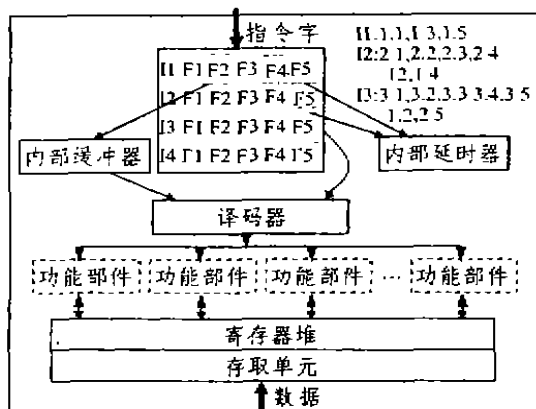


图2

4 设计结果

采用上述 VLIW 微处理器结构,我们设计了一个用 Verilog 描述的微处理器逻辑模型,指令字宽为128位,一条指令可以执行4~8个操作。仿真验证结果较满意,在80M 主频的条件下浮点峰值达160MFLOPS,几个典型 Benchmark 测试程序的性能指标如下:

Benchmark 程序	性能指标
Livermore	105MFLOPS
Linpack	120MFLOPS
Dhrystone	429MIPS

这个微处理器模型目前已通过富士通0.35 μ 工艺库的网表仿真,计划在年内生产样片,相关优化编译器、固件、操作系统正在研制过程中,希望能构成一个实用的计算机系统。

参考文献

- 1 Fisher J A. Very Ling Instruction Word Architecture and the ELI. ACM Conf. 1983:512
- 2 林钟官. Merced 芯片将开拓微处理器新时代. 微处理机, 1999,2
- 3 Available at: <http://www.research.ibm.com/vliw/>
- 4 Available at: <http://cag.lcs.mit.edu/raw/>
- 5 Available at: <http://sunl.stanford.edu/>
- 6 Available at: <http://www.goof.com/pcg/epic/>