

标量处理机流水线性能分析研究

陈利军,高秀峰,崔 静,李 芳

(军械工程学院 信息工程系,河北 石家庄 050003)

摘 要:目前标量处理机提高指令执行速度的途径除提高处理机工作频率外,常采用流水线技术和资源重复技术等。为研究这些技术对标量处理机性能的影响和这些技术使用中潜在的问题,利用一款具有32位精简指令集的虚拟处理器WinDLX对这些技术进行仿真研究。对WinDLX虚拟处理器的寄存器、指令集等结构特征进行详细介绍,然后对标量流水线处理机的工作原理、潜在问题和WinDLX对这些问题的解决办法进行说明。在此基础上,通过一个矩阵乘法运算程序分析了设置相关专用通路和提前形成条件码两种针对数据相关的解决办法及其实际效果。该实例揭示了提高标量处理机指令执行速度的有效途径,为设计和改进流水线结构及提高流水线的执行效率提供了参考依据。

关键词:标量处理机;流水线;WinDLX;虚拟处理器;精简指令集

中图分类号:TP391.9

文献标识码:A

文章编号:1673-629X(2016)11-0082-04

doi:10.3969/j.issn.1673-629X.2016.11.018

Analysis and Research on Pipelining of Scalar Processor

CHEN Li-jun, GAO Xiu-feng, CUI Jing, LI Fang

(Department of Information Engineering, Ordnance Engineering College, Shijiazhuang 050003, China)

Abstract: Now, besides increasing the operation frequency, the pipelining and the resource replication are used to accelerate the scalar processor. In order to study these technologies' effect and potential problems to the scalar processor, it uses the WinDLX virtual processor which has 32bit reduced instruction set. The structure characteristics of the WinDLX virtual processor like registers and instruction set are introduced. Then the operational principle of the scalar processor's pipelining, the potential problems and the settlements by WinDLX are discussed. After that, a matrix multiplication program is used to analyze the solutions to data correlation, one is correlation special channel, the other is early formation condition code. This example reveals an effective way to improve the execution speed of the scalar processor instructions, which provides a reference for the design and improvement of the pipeline structure and the improvement of the efficiency of the pipeline.

Key words: scalar processor; pipelining; WinDLX; virtual processor; RIS

0 引言

标量处理机是目前最为通用的一种处理机,在标量处理机中只有标量数据表示和标量指令系统。为提高标量处理机的指令执行速度,除提高处理机的工作频率,采用更好的算法和设计更好的功能部件外,最常用的途径就是开发指令级并行技术。指令级并行技术分为时间上的并行和空间上的并行。时间上的并行即流水线技术,空间上的并行即资源重复技术。在流水线处理机中由于指令并行执行,因此会出现指令顺序执行不会出现的一些问题,这些问题必须通过合理有效的办法加以解决才能发挥出流水线应有的效能^[1]。为研究流水线处理机的潜在问题及其解决办法,文中

利用WinDLX虚拟处理器进行了仿真研究。

1 WinDLX虚拟处理器的结构特征

1.1 WinDLX处理器的基本结构

WinDLX处理器是一种虚拟处理器,具有32位微处理机系统结构和多元未饱和型指令集。该处理器不仅体现了当今多种机器(AMD29K、DEC station 3100、HP850、IBM801、Intel i860等)指令集结构的共同特点,而且还体现了未来一些机器的指令集结构的特点。

WinDLX处理器有32个32位的通用寄存器 R_0, R_1, \dots, R_{31} 和一套浮点寄存器,浮点寄存器既可作为32个32位单精度寄存器使用 F_0, F_1, \dots, F_{31} ,也可作为16个双精度寄存器使用 F_0, F_2, \dots, F_{30} 。处理器支持

收稿日期:2016-03-25

修回日期:2016-06-28

网络出版时间:2016-09-19

基金项目:国家自然科学基金资助项目(61271152)

作者简介:陈利军(1973-),男,硕士,副教授,研究方向为计算机系统结构、嵌入式系统应用。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20160919.0843.068.html>

的数据类型包括 8 位、16 位和 32 位整型以及 64 位 IEEE754 浮点数。WinDLX 内存采用 32 位地址字节寻址,指令采用 32 位定长指令。

WinDLX 处理器的指令集属于简单的 Load/Store 指令集,指令注重流水效率和简化译码。WinDLX 提供了寄存器寻址、立即值寻址、偏移寻址和寄存器间接寻址 4 种方式,指令操作分为 Load/Store 操作、ALU 操作、分支和跳转操作以及浮点操作。所有的 ALU 指令都是寄存器—寄存器型指令,其运算包含了简单的算术和逻辑运算。浮点操作指令的操作数来源于浮点寄存器,能完成浮点加、减、乘、除操作。

1.2 WinDLX 平台构成

WinDLX 是一个基于 Windows 的模拟器,具有 Register, Code, Pipeline, Clock Cycle Diagram, Statistics 和 Breakpoints 共 6 个模块。在 WinDLX 中将每条指令分为取指 (IF)、译码 (ID)、执行 (EX)、访存 (MEM) 和写回 (WB) 5 个功能段,此外对于浮点操作还专门设置了浮点加、浮点乘和浮点除部件。在 WinDLX 中,“Clock Cycle Diagram”时钟周期图模块能准确地描绘出指令在 DLX 流水线中的调度及功能段的使用情况。通过该模块能有效地分析出指令流水过程中出现的结构相关、数据相关和控制相关。

2 流水线处理机潜在的问题

在流水线中,由于处理机被分为若干个功能段,如取指、分析和执行功能段,指令在这些功能段中流水前进^[2-3]。

在这样的流水线中由于多个功能段同时工作,因此会出现指令顺序执行中不会出现的一些问题。比较明显的就是访存冲突问题。例如,取指功能段需要访问主存取指令,分析功能段需要从主存取操作数,执行功能段需要向主存写回运算结果。当这 3 个功能段同时工作时,就会出现访存冲突^[4]。为了解决该问题,标量处理机采用了先行控制技术。通过增设先行指令缓冲栈、先行读数栈和后行写数栈,实现对指令流和数据流的预处理和缓冲,尽量使取指、分析和执行部件独立工作而不会出现访存冲突^[5]。

WinDLX 处理器的指令集属于简单的 Load/Store 指令集,只有 Load/Store 指令要访存,而所有的 ALU 指令都是寄存器—寄存器型指令,不需要访存,这样就能极大降低访存冲突。

除访存冲突问题外,在流水线实际运行中还会出现结构相关、数据相关和控制相关等问题^[6-7]。

结构相关是指指令在并行解释过程中,前、后指令要求使用同一个功能部件所引起的资源冲突。

数据相关是指后续指令若用到前面指令的执行结

果,则必须等待前面指令执行完成、写入结果后,后续指令才能执行。

控制相关是指流水线遇到分支或转移指令,导致流入流水线的指令作废而产生的相关。

流水线中的这几种相关通常同时存在,一旦出现相关,必然会影响指令的执行速度,降低流水线的执行效率。

3 WinDLX 针对相关问题的解决办法

3.1 结构相关的解决方法

在标量流水线处理机中,针对结构相关可通过资源重复技术加以解决,即通过重复设置易发生使用冲突的功能部件,让同时解释执行的多条流水指令独立使用重复设置的多个功能部件,从而解决该问题。WinDLX 提供对浮点运算部件的重复设置功能,每个运算部件最多可重复设置 8 套^[8]。

3.2 数据相关的解决方法

针对数据相关,标量流水线处理机通过建立相关专用通路加以解决。相关专用通路的基本原理是数据重定向。图 1 为 4 条指令组成的完成 1 个简单算术四则运算的程序段。

k : LOAD F₁, A; F₁ ← A
k+1: FADD F₁, F₂; F₁ ← F₁+F₂
k+2: FMUL F₁, F₃; F₁ ← F₁×F₃
k+3 STROE F₁, B; B ← F₁

图 1 算术四则运算程序

该程序段执行时的数据流程如图 2 所示。

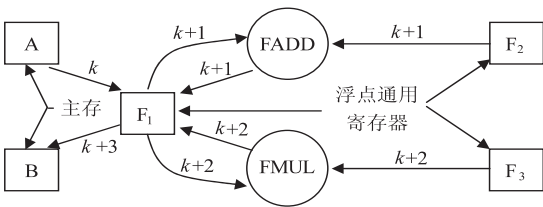


图 2 原数据流程

图中,F₁,F₂,F₃代表浮点寄存器,A 和 B 代表主存单元,FADD 和 FMUL 代表运算部件,带有箭头的线表示数据传送路径。

由图 2 看出,指令 k,k+1,k+2 和 k+3 依次存在“先写后读”相关性;指令 k,k+1,k+2 依次存在“写—写”相关性。这些相关性的出现必然造成流水线中后续相关指令的等待,降低流水线的执行效率。

对于“先写后读”相关性,可设置 A→FADD,FADD→FMUL 和 FMUL→B 这 3 条专用通路,同时撤消 F₁→FADD 与 F₁→FMUL 的通路。

对于“写—写”相关性,可撤消 A→F₁及 FADD→F₁的通路。

通过以上专用通路的设置,解决了数据相关性的

问题。

设置了相关专用通路后的数据流程如图 3 所示。

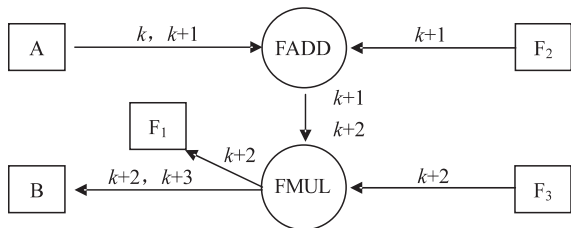


图 3 重定向后的数据流程

相关专用通路的设置解决了因数据相关性而产生的指令等待,保证了最终写入数据的正确性,从而提高了流水线的执行效率^[9]。

WinDLX 由于所有的 ALU 指令都是寄存器—寄存器型指令,因此数据相关基本都是涉及通用寄存器的数据相关,针对通用寄存器数据相关采用相关专用通路是有效的解决办法。在 WinDLX 中通过 Configuration-EnableForwarding 菜单项可设置或取消流水线中的相关专用通路。

3.3 控制相关的解决方法

针对控制相关,通常可以采取转移预测技术等方法加以解决。转移预测主要是对条件转移指令转移成功还是转移不成功进行预测,让预测分支进入流水线^[10-11]。

在 WinDLX 中,解决控制相关的主要办法是从硬件上提前得到转移的目标地址。WinDLX 中将计算 PC 值的多路选择器 MUX 提前到 IF 段,保证对 PC 的写操作只在一个流水段内出现,避免了在遇到分支转移时,流水线中的转移成功和转移不成功 2 条分支指令都试图在不同的流水段写 PC 值从而引发冲突^[12]。

4 求阶乘程序的对比分析

为了检验 WinDLX 对相关问题的解决办法和效率,编制了一个矩阵乘法运算程序, $C = A \times B$ 。如图 4 所示,设 $A = (a_{ij})_{m \times k}$, $B = (b_{ij})_{k \times n}$, 则 $C = (c_{ij})_{m \times n}$ 。其中,取 $m = 5, k = 4, n = 5$ 。

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \\ 17 & 18 & 19 & 20 \end{pmatrix} \times \begin{pmatrix} 21 & 22 & 23 & 24 & 25 \\ 26 & 27 & 28 & 29 & 30 \\ 31 & 32 & 33 & 34 & 35 \\ 36 & 37 & 38 & 39 & 40 \end{pmatrix}$$

图 4 矩阵 $A \times B$

由于两个原始矩阵各有 20 个分量,运行结果有 20 个分量,共计 60 个分量,超出了 WinDLX32 个通用寄存器的数量,因此该程序采用存储器来存放原始矩阵和结果矩阵的每个分量^[13]。

首先通过循环程序对连续的 40 个存储单元赋初值,然后通过三重循环嵌套实现矩阵乘法运算并将各结果分量存入存储器,最后通过系统调用 Trap 5 实现对运行结果的显示。其中对 40 个连续的存储单元赋初值的循环程序如图 5 所示。

```

addi    r1,r0,0
addi    r2,r0,0
addi    r4,r0,0
addi    r5,r0,0

chushi:
addi    r2,r2,1
sw      4(r1),r2
addi    r1,r1,4
addi    r4,r4,1;   循环控制变量
subi    r5,r4,40
beqz    r5,mul;    初始化完成
j       chushi;    未初始化完

mul:    ...

```

图 5 对存储单元循环赋初值

在 WinDLX 的流水线时钟周期图中可以看到,由于 $\text{addi } r_4, r_4, 1$ 指令与 $\text{subi } r_5, r_4, 40$ 指令存在针对 r_4 寄存器的先写后读相关; $\text{subi } r_5, r_4, 40$ 指令与 $\text{beqz } r_5, \text{mul}$ 指令存在针对 r_5 寄存器的先写后读相关,在未设置数据专用通路的情况下,造成了后续指令的读操作数延迟 R-Stall。在实现矩阵乘法运算的三重循环嵌套中也存在类似问题。程序在未设置数据专用通路的情况下最终执行时间为 52 s,程序总执行周期数为 4 730,先写后读相关 (RAW) 共 2 161 次,占总周期数的 45.7%。

如果改进处理机结构,增设数据专用通路,则 $\text{addi } r_4, r_4, 1$ 指令与 $\text{subi } r_5, r_4, 40$ 指令之间由于相关性造成的 R-Stall 延迟得到解决, $\text{subi } r_5, r_4, 40$ 指令与 $\text{beqz } r_5, \text{mul}$ 指令之间相关性造成的 R-Stall 延迟得到部分解决。整个程序的执行周期数为 3 150,先写后读相关 (RAW) 降到 570 次,占总周期数的 18.1%。

设置相关专用通路只让 $\text{subi } r_5, r_4, 40$ 指令与 $\text{beqz } r_5, \text{mul}$ 指令之间的相关性问题得到了部分解决而不是全部解决,因为 $\text{beqz } r_5, \text{mul}$ 指令属于一般条件转移指令,该指令在指令译码功能段 (ID) 就要真正用到 r_5 进行判断,而不像 $\text{subi } r_5, r_4, 40$ 指令在执行功能段 (intEX) 才会真正用到 r_4 。因此即便是设置了相关专用通路,只要 $\text{subi } r_5, r_4, 40$ 指令与 $\text{beqz } r_5, \text{mul}$ 指令相邻,它们之间的 R-Stall 延迟不可避免。

为了完全解决这种一般条件转移指令造成的先写后读相关,可以对程序进行优化,提前形成条件码^[14]。在该例中,可以将 $\text{addi } r_4, r_4, 1$ 指令与 $\text{subi } r_5, r_4, 40$ 指令提前,如图 6 所示。

```
addi    r1,r0,0
addi    r2,r0,0
addi    r4,r0,0
addi    r5,r0,0

chushi:
addi    r4,r4,1;    循环控制变量
subi    r5,r4,40
addi    r2,r2,1
sw      4(r1),r2
addi    r1,r1,4
beqz    r5,mul;    初始化完成
j       chushi;    未初始化完

mul:    ...
```

图 6 优化后的程序

这样 beqz r₅, mul 指令的判断条件会提前得到,该数据相关性问题会得到彻底解决。对矩阵乘法运算中的三重循环嵌套也采用类似办法优化代码,最后程序的执行时间仅需 32 s,整个程序的执行周期数为 2 990,先写后读相关(RAW)降到 400 次,占总周期数的 13.38%。

5 结束语

通过上述程序可以看到,设置相关专用通路和提前形成条件码是 WinDLX 解决数据相关的有效途径,能极大提高流水线的执行效率。WinDLX 同时支持资源重复设置,对浮点加法器、浮点乘法器和浮点除法器可进行重复设置。在浮点运算程序中通过重复设置浮点运算部件可极大提高流水线的执行效率。由此可见,WinDLX 虚拟处理器为设计、研究和改进标量处理机流水线结构提供了一个方便、直观的平台,研究该平台有助于对计算机系统结构的学习和理解。

参考文献:

[1] Hennessy J L, Patterson D A. Computer architecture – a quantitative approach[M]. 4th ed. San Francisco: Morgan Kauf-

mann Publishers, 2007.

[2] 张晨曦. 计算机系统结构[M]. 第 4 版. 北京: 高等教育出版社, 2006.

[3] Sherwood T, Perelman E, Hamerly G, et al. Discovering and exploiting program phases[J]. IEEE Micro, 2003, 23(6): 84–93.

[4] 薛 杨. 流水线技术性能评价与最佳段数选择[J]. 吉林省教育学院学报, 2011, 27(10): 141–144.

[5] 刘 博, 张盛兵, 黄嵩人. 静态超标量 MCU–DSP 内核的 Load 先行访存调度[J]. 计算机应用研究, 2013, 30(2): 450–453.

[6] 王晓勇, 张盛兵, 黄嵩人. 一种多发射 DSP 的数据相关控制[J]. 微型电脑应用, 2011, 27(11): 56–58.

[7] 柴晓东. 指令流水线阻塞问题分析[J]. 濮阳职业技术学院学报, 2014, 27(5): 143–144.

[8] 潘琢金, 郑彩平, 杨 华. 流水线前端资源分配及其性能影响研究[J]. 计算机工程, 2010, 36(14): 275–277.

[9] 朱博元, 刘高辉, 李政运, 等. RISC 指令集众核处理器功能验证与实现[J]. 计算机工程与应用, 2014, 50(21): 54–58.

[10] Ho C Y, Chng K F, Yau C H, et al. A study of dynamic branch predictors: counter versus perceptron[C]//Proc of the international conference on information technology. [s. l.]: [s. n.], 2007: 528–563.

[11] Jimenez D A, Lin C. Dynamic branch prediction with perceptron[C]//Proc of the 7th international symposium on high performance computer architecture. [s. l.]: [s. n.], 2001: 197–206.

[12] 赖兆馨, 潘 明, 许 勇, 等. 嵌入式五级流水线 CPU 核的设计与实现[J]. 微计算机信息, 2008, 24(29): 32–34.

[13] Johnsonhaugh R, Schaefer M. Algorithms[M]. Beijing: Tsinghua University Press, 2007: 195–197.

[14] Pan Y, Mitra T. Characterizing embedded applications for instruction-set extensible processors[C]//Proc of the 2004 international conference on compilers. [s. l.]: ACM, 2004: 67–78.

(上接第 81 页)

(2): 141–149.

[9] R–Core–Team. R: a language and environment for statistical computing R foundation for statistical computing[M]. [s. l.]: [s. n.], 2013.

[10] Zhang Dongchen, Wang Shoufeng. A novel long term traffic forecast algorithm and case study for China[C]//Proc of international conference on advanced information networking and applications workshops. [s. l.]: [s. n.], 2014: 425–430.

[11] Sommer M, Tomforde S, Haner J. Using a neural network for forecasting on an organic traffic control management system [C]//Proc of international workshop on embedded self-organizing systems. [s. l.]: [s. n.], 2013.

[12] Wang Haryan, Wang Youzhen. Vessel traffic flow forecasting

with the combined model based on support vector machine [J]. Transportation Information and Safty, 2015, 3: 695–698.

[13] Chan K Y, Dillon T S, Singh J, et al. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm [J]. IEEE Transactions on Intelligent Transportation Systems, 2012, 13(2): 644–654.

[14] Lippi M, Bertini M, Frasconi P. Short-term traffic flow forecasting: an experimental comparison of time-series analysis and supervised learning [J]. IEEE Transactions on Intelligent Transportation Systems, 2013, 14(2): 871–882.

[15] Dunne S, Ghosh B. Weather adaptive traffic prediction using neurowavelet models [J]. IEEE Transactions on Intelligent Transportation Systems, 2013, 14(1): 370–379.