

乱序超标量处理器核的功耗优化^{*}

孙彩霞, 李文哲, 高 军, 王永文

(国防科学技术大学计算机学院, 湖南 长沙 410073)

摘 要:为了追求更高的性能,处理器核的主频不断提升,处理器核的设计日益复杂,随之而来的是功耗问题越来越突出。除了在工艺级和电路级采用低功耗技术外,在逻辑设计阶段通过分析处理器核各个功能模块的特点并采用相应的技术手段,也可以有效降低功耗。对一款乱序超标量处理器核中功耗比较突出的模块——寄存器文件和再定序缓冲——进行了逻辑设计优化,在程序运行性能几乎不受影响的情况下明显减少了面积,降低了功耗。

关键词:乱序;超标量;性能;功耗

中图分类号:TP302.2

文献标志码:A

doi:10.3969/j.issn.1007-130X.2017.01.006

Power optimization of an out-of-order superscalar processor core

SUN Cai-xia, LI Wen-zhe, GAO Jun, WANG Yong-wen

(College of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: To maximize the performance, the frequency of the processor core becomes increasingly higher, and the design of the processor core gets much more complex. As a result, power issues become a challenge and need special care. Beside process-level and circuit-level low power technologies, adopting some strategies according to the features of modules during the RTL design can reduce power consumption. We analyze the power of an out-of-order superscalar processor core and optimize the designs of register files and the reorder buffer based on the analysis. Evaluation results show that the area and power of the processor core decrease with no obvious performance penalty.

Key words: out-of-order; superscalar; performance; power

1 引言

功耗问题以前只是在嵌入式和移动计算领域被高度重视,但是现在高性能处理器的设计也日益受到功耗问题的制约。一方面,处理器主频的提升速度虽然不快,但是总的趋势依然是上涨;另一方面,为了追求更高的性能,高性能处理器采用更加复杂的设计,包括分支预测、前瞻执行、乱序执行、宽发射、大容量片上 Cache 等,而且为了更大可能地开发指令级并行,各种资源数目越来越多。比如,Intel Xeon 系列处理器的几代 Tock 架构,乱序

执行资源数目呈不断上升趋势^[1-5]。

功耗问题已经成为继续提升处理器性能的首要障碍,进而得到了广泛关注^[6-8],低功耗设计也开始贯穿在整个处理器的研制过程中。除了在工艺级和电路级降低功耗,逻辑设计阶段通过分析处理器各个功能模块的特点并采用相应的技术手段,也可以有效降低功耗。

本文以一款乱序超标量处理器核 XMC 的 RTL(Register Transfer Level)代码为基础,首先对其功耗进行分析,然后从功耗比较突出的模块着手,通过分析其功能特点来采取逻辑设计优化,在尽可能小地牺牲程序运行性能的前提下减少面积、

^{*} 收稿日期:2016-08-13;修回日期:2016-10-15

基金项目:国家自然科学基金(61103011)

通信地址:410073 湖南省长沙市国防科学技术大学计算机学院

Address: College of Computer, National University of Defense Technology, Changsha 410073, Hunan, P. R. China

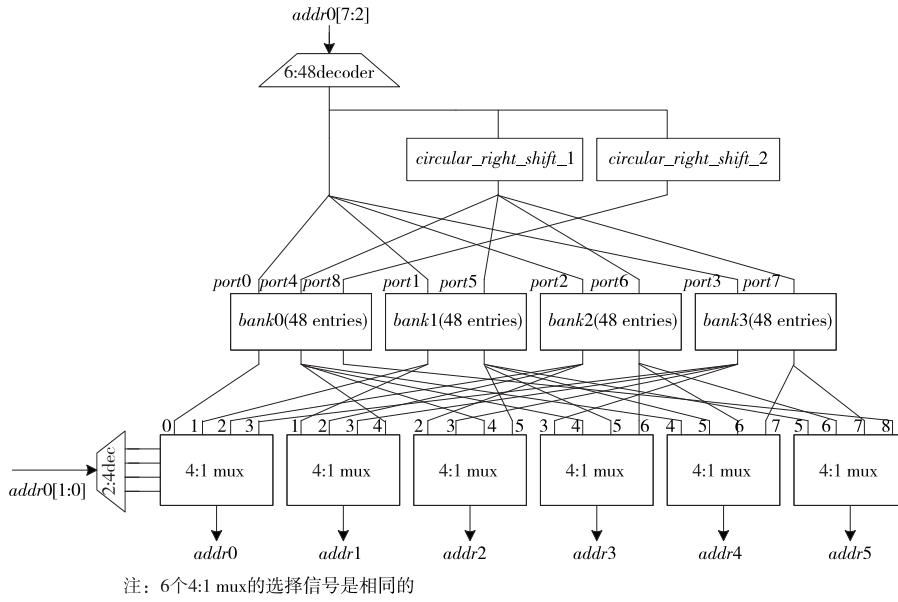


Figure 3 Banking of the rename register file

图3 重命名寄存器文件的分体设计结构示意图

的寄存器的数据,体1(*bank1*)存放的是索引模4为1的寄存器的数据,体2(*bank2*)存放的是索引模4为2的寄存器的数据,体3(*bank3*)存放的是索引模4为3的寄存器的数据。结果写回的6个地址是连续的,只对地址0(*addr0*)的高6位进行译码,得到 one-hot 编码,然后循环右移得到其他索引的 one-hot 编码;4个体共有9个读端口,读出的9组数据分别送往6个4选1开关,这些4选1开关根据 *addr0* 的低两位选出所需数据。

从图3不难看出,*bank0* 的读端口数目为3,而其余3个体的读端口数目仅为2,与未分体时的6个相比,逻辑量减少明显,我们将在后面的章节进行定量评估。

除此之外,还对 ROB 的某些数据体进行了分体设计,由于设计思路和重命名寄存器文件的分体

设计是相同的,所以不再赘述。

3.2 寄存器文件的读端口共享

XMC 每拍分派4条指令,分派指令时会读取寄存器文件获取源操作数。每条指令最多可能有4个源操作数,因此寄存器文件的读端口数按照最大可能需要16个。但是,我们通过统计发现^[10],同一个周期分派的4条指令的源操作数总个数大多数时候都不超过12,因此可以通过共享的方式减少读端口数目,并且不会对性能造成太大的影响。端口共享方法在有关文献中也有提及^[9],但是如何共享则需要考虑特定处理器的实现。

通过运行典型应用程序,统计同时分派的指令源操作数的情况,我们确定了如图4所示的寄存器文件读端口共享策略。寄存器文件设置12个读端口,其中6个端口不共享,分别分配给指令0的源

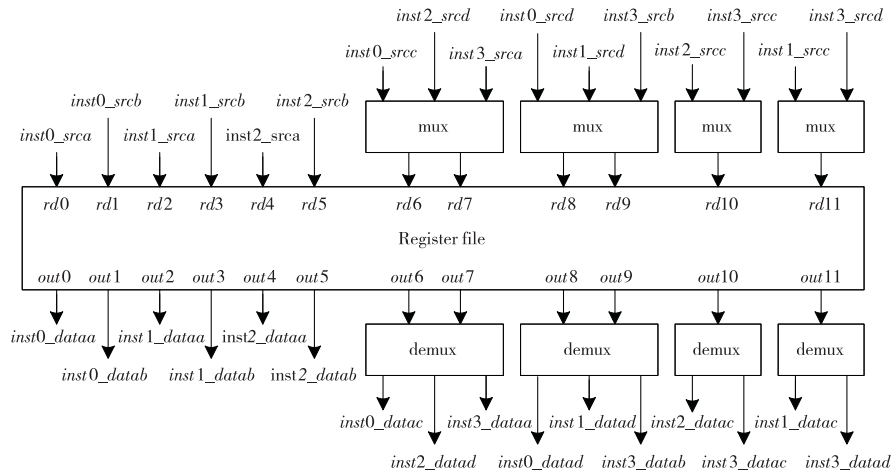


Figure 4 Read ports sharing of the register file

图4 寄存器文件读端口共享设计

操作数 $a(inst0_srca)$ 、指令 0 的源操作数 $b(inst0_srcb)$ 、指令 1 的源操作数 $a(inst1_srca)$ 、指令 1 的源操作数 $b(inst1_srcb)$ 、指令 2 的源操作数 $a(inst2_srca)$ 和指令 2 的源操作数 $b(inst2_srcb)$ ，其余 6 个端口由指令 0 的源操作数 $c(inst0_srcc)$ 、指令 0 的源操作数 $d(inst0_srcd)$ 、指令 1 的源操作数 $c(inst1_srcc)$ 、指令 1 的源操作数 $d(inst1_srcd)$ 、指令 2 的源操作数 $c(inst2_srcc)$ 、指令 2 的源操作数 $d(inst2_srcd)$ 、指令 3 的源操作数 $a(inst3_srca)$ 、指令 3 的源操作数 $b(inst3_srcb)$ 、指令 3 的源操作数 $c(inst3_srcc)$ 和指令 3 的源操作数 $d(inst3_srcd)$ 共享。我们选取了一种性价比较高的交叉开关实现方式，即 6 个共享端口分成四组，第 1 组两个端口，由 $inst0_srcc$ 、 $inst2_srcd$ 和 $inst3_srca$ 共享，第 2 组两个端口，由 $inst0_srcd$ 、 $inst1_srcd$ 和 $inst3_srcb$ 共享，第 3 组一个端口，由 $inst2_srcc$ 和 $inst3_srcc$ 共享，第 4 组一个端口，由 $inst1_srcc$ 和 $inst3_srcd$ 共享。共享端口优先满足指令 0/1/2。

采用端口共享方式，寄存器文件的源操作数读端口数目从 16 降到了 12，逻辑量明显减少。但是，当同时分派的 4 条指令的源操作数总个数超过 12 时，或者共享同一端口的所有源操作数都有效时，会导致第 4 条指令无法分派，进而可能会影响程序运行性能。我们将在后续章节对性能、面积和功耗进行定量评估。

4 优化设计的评估

从逻辑综合结果、功耗和程序运行时的性能三个方面评估功耗优化设计。

4.1 逻辑综合结果

在某特定工艺库下，我们对优化前后的 RTL 代码进行了逻辑综合，在同等约束条件下，优化后代码并未带来时序的恶化。另外，对优化前后被优化模块的组合单元数、时序单元数和标准单元总数进行了对比，如表 1 所示，表中给出的数据为相对于优化前，优化后模块的单元数目减少的百分比。

重命名寄存器文件的分体设计(以下简称优化 1)主要针对重命名寄存器文件的状态阵列($rename_rf_status$)进行优化，分体设计使得该模块的组合单元数目减少了 25.41%，时序单元数目减少了 20.03%，标准单元总数减少了 24.87%。时序单元数目减少是因为未做分体设计时，192 选 1 的逻辑分成两拍实现，因此有大量的中间结果寄存

器，分体设计时我们将寄存器文件的读取在一拍内完成，节约了中间结果寄存器，同时并未导致时序恶化。

寄存器文件的读端口共享(以下简称优化 2)同时应用在重命名寄存器文件($rename_rf$)和体系结构寄存器文件(包括 int_rf 和 fp_rf)上，该优化与原设计相比，减少了读取数据时的选择逻辑，读取前后增加了必要的交叉开关，对时序单元数目没有任何影响。

ROB 中某些数据体的分体设计(以下简称优化 3)令 rob_queue 模块的标准单元总数减少了 9.49%，时序单元数目只有轻微的变化。

最后，将所有优化合并，对 XMC 处理器核优化前后的标准单元数目进行对比，如表 1 最后一栏所示，XMC 处理器核的标准单元总数减少了 6.08%。可见，在编写 RTL 代码时，仔细分析逻辑功能，并根据其特点进行逻辑设计，能带来更加优化的实现。

Table 1 Comparison of the logic synthesis results before and after optimization

表 1 优化前后逻辑综合结果对比

	被优化模块	组合单元数/%	时序单元数/%	标准单元总数/%
优化 1	$rename_rf_status$	25.41	20.03	24.87
	$rename_rf$	14.56	0	13.90
优化 2	int_rf	16.15	0	14.53
	fp_rf	18.85	0	16.10
优化 3	rob_queue	11.21	0.19	9.49
所有优化	XMC 处理器核	6.81	1.39	6.08

4.2 功耗

使用 PowerArtist 工具，导入 Dhrystone 测试程序的波形文件，对优化前后 XMC 处理器核的功耗进行了评估，结果如表 2 所示，表中给出的数据为相对于优化前，优化后模块的功耗减少的百分比。

分体设计令重命名寄存器文件状态阵列的功耗降低很显著，尤其是动态功耗，降低了 33.04% 之多。这主要是因为分体设计不但简化了读取逻辑，而且读取在一拍内完成，节省了大量的中间结果寄存器。

寄存器文件的读端口共享主要降低的是重命名寄存器文件和体系结构寄存器文件的静态功耗，而动态功耗反而有轻微增加。这是因为：读取寄存器文件的有关信号进行了细粒度门控，当源操作数无效时，读取不会发生，以节省动态功耗。因此，完

成同样的指令,无论端口是否共享,要读取的源操作数个数都是相同的。但是,读端口共享时,寄存器文件读取前后各增加了一些必要的交叉开关,这些交叉开关会产生一些动态功耗。

分体设计令 rob_queue 模块的功耗也有一定的降低。

将所有优化合并后,XMC 处理器核的静态功耗相比优化前降低了 4.38%,动态功耗降低了 4.84%,总功耗降低了 4.61%。

Table 2 Comparison of power

表 2 优化前后功耗对比

	被优化模块	静态功耗/%	动态功耗/%	总功耗/%
优化 1	rename_rf_status	19.87	33.04	28.15
	rename_rf	13.75	-2.26	6.37
优化 2	int_rf	14.71	-2.42	8.90
	fp_rf	15.68	-2.20	12.25
优化 3	rob_queue	4.16	2.97	3.52
所有优化	XMC 处理器核	4.38	4.84	4.61

4.3 性能

重命名寄存器文件和 ROB 的分体设计只是改变了逻辑实现方式,对程序运行性能没有影响,但是寄存器文件的读端口共享是会影响性能的。我们借助 FPGA 原型系统,运行 SPEC CPU2000 测试程序,对优化前后的性能进行了对比,结果如图 5 所示,纵坐标给出的是优化后性能相对于优化前性能变化的百分比。

可以看出,对于大部分程序,性能变化都在 0.13%以内,可以认为几乎没有受到影响。有个别程序的性能提升了,我们仔细分析了这一现象出现的原因。分析发现,由于端口共享改变了指令分派的时机,从而影响到整个程序的访存流,具体而言,指令 Cache 失效请求和数据 Cache 失效请求的调度次序发生了变化。不过从结果看,提升最多的 164.zip 也仅为 0.66%。总的来说,性能还是有所

下降的,平均降低了 0.18%,降低最多的是 177.mesa,为 1.67%。

5 结束语

本文从功耗比较突出的模块着手,对 XMC 处理器核的功耗进行了优化。优化包括:根据结果写回时读取重命名寄存器文件的索引是连续的这一特点,对重命名寄存器文件进行分体设计;根据同时分派的 4 条指令源操作数总个数大多数时候都不超过 12 这一统计,实现了寄存器文件读端口共享;此外,还对 ROB 的某些数据体进行了分体设计。评估结果表明,这些优化令 XMC 处理器核组合单元数据减少了 6.81%,时序单元数目减少了 1.39%,标准单元总数减少了 6.08%;运行 Dhrystone 程序时的静态功耗降低了 4.38%,动态功耗降低了 4.84%,总功耗降低了 4.61%;在运行 SPEC CPU2000 测试程序时,优化后的代码令性能降低了 0.18%,综合考虑逻辑综合结果和功耗,优化设计对性能的影响是可接受的。

参考文献:

[1] Jain T, Agrawal T. The Haswell microarchitecture—4th generation processor[J]. International Journal of Computer Science and Information Technologies, 2013, 4(3): 477-480.

[2] Intel Skylake microarchitecture detailed. TechPowerUp cite [EB/OL]. [2015-08-19]. <http://www.techpowerup.com>.

[3] The Intel Skylake mobile and desktop launch, with architecture analysis. AnandTech cite [EB/OL]. [2015-09-01]. <http://www.anandtech.com>.

[4] Intel Sandy Bridge Microarchitecture. Real word technologies cite [EB/OL]. [2010-09-25]. <http://www.realworldtech.com>.

[5] Sandy Bridge: Intel's next-generation microarchitecture revealed. ExtremeTech cite [EB/OL]. [2010-09-18]. <http://www.extremetech.com>.

[6] Perais A, Seznec A, Michaud P, et al. Cost-effective speculative scheduling in high performance processors[C]// ACM Si-

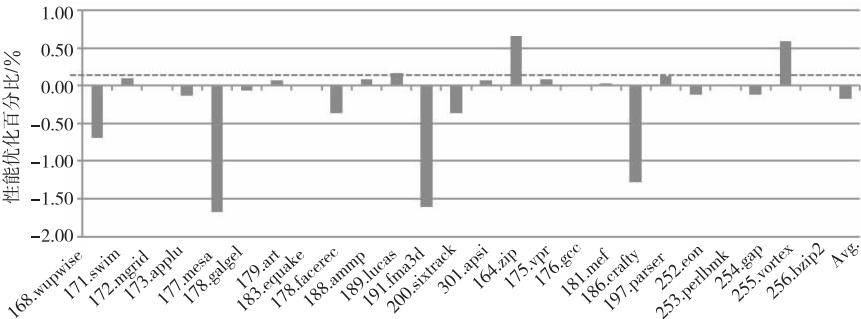


Figure 5 Comparison of performance

图 5 优化对 SPEC CPU2000 程序的性能影响

- garch Computer Architecture News, 2016,43(3):247-259.
- [7] Folegnani D, Gonzalez A. Energy-effective issue logic[C]//ACM Sigarch Computer Architecture News, 2001, 29(2): 230-239.
- [8] Moshovos A. Power-aware register renaming[R]. Toronto: University of Toronto, 2002.
- [9] Sun Cai-xia, Sui Bing-cai, Wang Lei, et al. Counters based performance analysis and optimization of a processor core[C]//Proc of the 19th National Conference on Computer Engineering and Technology, 2015:1. (in Chinese)
- [10] Li Wen-zhe. Design and organization of the register renaming mechanism in an out-of-order superscalar processor [D]. Changsha: National University of Defense Technology, 2015. (in Chinese)

附中文参考文献:

- [9] 孙彩霞, 隋兵才, 王蕾, 等. 基于计数器的处理器核性能分析与优化[C]//第十九届计算机工程与工艺学术年会, 2015:1.
- [10] 李文哲. 乱序超标量处理器寄存器重命名机制的设计与优化[D]. 长沙: 国防科学技术大学, 2015.

作者简介:



孙彩霞 (1979 -), 女, 黑龙江绥化人, 博士, 副研究员, 研究方向为计算机体系结构和微处理器设计。E-mail: cxsun@nudt.edu.cn

SUN Cai-xia, born in 1979, PhD, asso-

ciate research fellow, her research interests include computer architecture, and microprocessor design.



李文哲 (1989 -), 男, 湖南张家界人, 硕士, 研究方向为微处理器设计。E-mail: liwenzheyihao@yeah.net

LI Wen-zhe, born in 1989, MS, his research interest includes microprocessor design.



高军 (1978 -), 男, 甘肃高台人, 博士, 副研究员, 研究方向为计算机体系结构和微处理器设计。E-mail: gjun78@nudt.edu.cn

GAO Jun, born in 1978, PhD, associate research fellow, his research interests include computer architecture, and microprocessor design.



王永文 (1977 -), 男, 山东泰安人, 博士, 研究员, CCF 会员 (12958M), 研究方向为计算机体系结构和微处理器设计。E-mail: yongwen@nudt.edu.cn

WANG Yong-wen, born in 1977, PhD, research fellow, CCF member (12958M), his research interests include computer architecture, and microprocessor design.