

## 一种 VLIW-Superscalar 混合微处理器结构

陈峰扬, 杜 勇, 郭德源, 何 虎

(清华大学 微电子学研究所, 北京 100084)

**摘 要:** 描述了一款同时支持超长指令字(VLIW)与超标量(Superscalar)的混合处理器结构. 该结构在一条流水线上, 通过分发级(DDP)控制逻辑的灵活部署, 支持在软件层面使用指令控制该流水线在两种结构模式间的切换. 详细描述了该混合结构在 Lily2 处理器上的设计实现. 对该处理器结构的性能评测由一款基于 Open64 的 Lily2 编译器提供支持. 评测结果表明该处理器在信号处理领域和通用处理领域都有较强的性能优势.

**关键词:** 超长指令字; 超标量; 混合结构; 微处理器; 指令级并行

**中图分类号:** TP368.1

**文献标识码:** A

**文章编号:** 1000-7180(2013)11-0001-05

## A VLIW-Superscalar Hybrid Microprocessor Architecture

CHEN Feng-yang, DU Yong, GUO De-yuan, HE Hu

(Institute of Microelectronics, Tsinghua University, Beijing 100084, China)

**Abstract:** This paper proposed a hybrid microprocessor architecture that combines VLIW and Superscalar executing mode. By flexibly deploying control logic of the dispatch stage in the original single VLIW pipeline and implementing two mode-switching instructions, this architecture can be switched between the two executing mode by programmer or compiler. This paper describes the design and implementation of this architecture on Lily2 ISA. The evaluations based on an open64 compiler toolchain show that this architecture performs well in both signal processing applications and general calculating applications.

**Key words:** VLIW; superscalar; hybrid architecture; microprocessor; instruction level parallelism

### 1 引言

计算过程的指令之间存在着潜在的并行性, 即在同一个周期内可以有多个指令或多个操作同时进行, 这种并行性称为指令级并行. 发掘指令级并行的方法主要有两种, 一种依靠硬件动态地发掘指令级并行性, 另一种依赖于软件, 在编译阶段静态地发掘并行性. 超长指令字(VLIW)<sup>[1]</sup>和超标量(Superscalar)正是建立在发掘指令级并行方法上的两种主流微处理器结构<sup>[2]</sup>. Superscalar 具有较好的代码兼容性和实时性, 但硬件设计具有较高的复杂度. 而 VLIW 硬件设计复杂度较低, 但缺乏对现有代码的兼容性和对实时事务的良好处理能力<sup>[2]</sup>.

Superscalar 和 VLIW 这两种结构之间存在着优势互补, 为了同时发挥两种结构的优点, 本文提出的方案基于一款成熟的 VLIW 处理器结构, Lily DSP<sup>[3]</sup>.

### 2 Lily2 VLIW-SuperScalar 混合结构

#### 2.1 VLIW-SuperScalar 混合结构流水线

Lily2 的整体设计继承于其上一代处理器 Lily. 如图 1 所示, Lily2 的程序存储管理模块(PMMU)和数据存储管理模块(DMMU)分别控制在相应 Cache 和 Memory 中的指令和数据的读取. Lily2 拥有 6 路并行发射能力, 其执行部分被设计为双簇(X 簇和 Y 簇)结构, 每一簇包含 3 个指令执行单元和

一个独占的寄存器堆,并且共享一个全局寄存器堆. 能简介如下:

Lily2 为 10 级流水线设计,如图 2 所示. 各流水级功

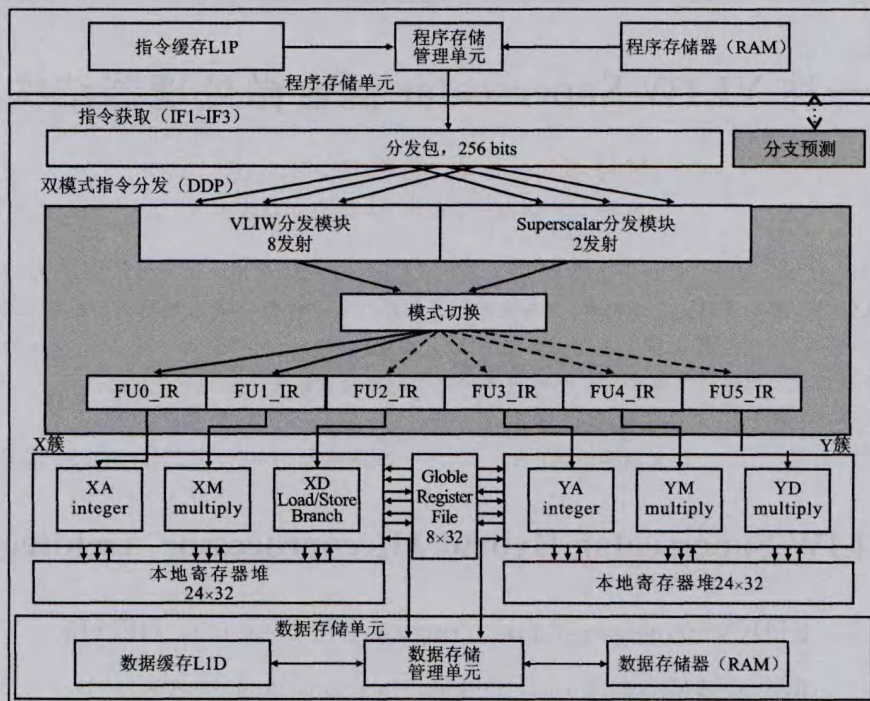


图 1 Lily2 的整体结构

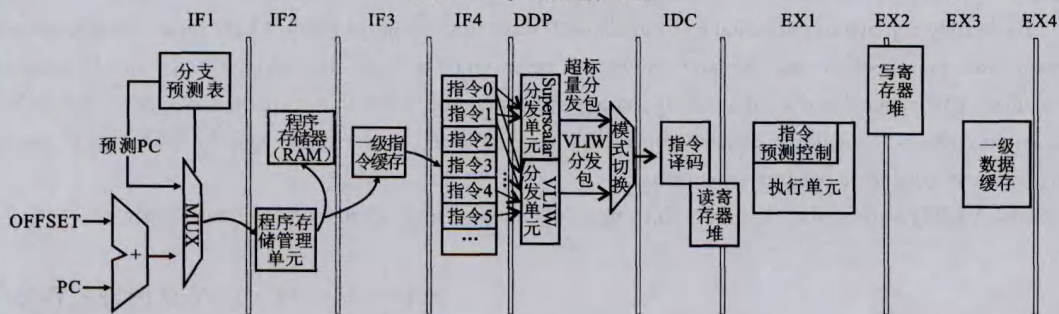


图 2 Lily2 的流水线

IF1(取指 1 级): SuperScalar 模式下的分支预测、PC 生成;

IF2(取指 2 级): 指令地址传递;

IF3(取指 3 级): 等待指令存储器读取 256 位宽的指令包;

IF4(取指 4 级): 获取指令包;

DDP(双模式分发级): 与 Lily 相比, Lily2 在结构上的最大改进在该流水级体现: 两种分发模式分别对应两种不同的分发策略——VLIW 分发策略和 Superscalar 分发策略, 并且实现为两个独立的模块. 可以通过指令直接控制分发策略切换单元选择不同的模块, 以实现两种分发策略的灵活切换;

IDC(指令译码级): 对指令译码, 产生分支指令

的地址;

EX1(执行 1 级): ①执行指令功能, 如加减法运算(乘法、除法等多周期指令在后续流水级中继续执行); ②指令预测控制模块在 Superscalar 模式下判定分支预测是否成功, 并将结果反馈到前级流水. 该模块与 IF1 级的分支预测模块结合实现了 Lily2 SuperScalar 模式的分支预测功能;

EX2~EX4(执行 2、3、4 级): 除了完成多周期指令的执行并写入目的寄存器外, Load、Store 的数据读、写操作也在这几个流水级完成.

## 2.2 VLIW-Superscalar 混合结构分发模式的切换

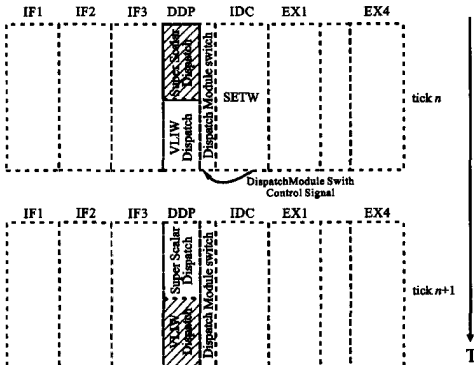
Lily2 的单一指令流中可混合两种模式的指令片段, 每一指令片段的入口为一条相应的切换指令

(SETW/SETR),由该指令控制 DDP 级切换到指定的分发模式,处理器将以该模式继续执行后续指令。

图 3(a)左边是一段典型的 Lily2 双模式混合指令流模式切换代码示例,在 SETW(指令 3)未进入 IDC 级之前,处理器使用 Superscalar 分发策略分发执行 SETW 及之前的指令,当 SETW 进入 IDC 级,译码模块识别出并反馈信号到 DDP 级,如图 3(b)所示,DDP 级中的切换模块根据反馈信息将分发模式切换为 VLIW 模式。随后,处理器开始使用 VLIW 分发策略分发执行 SETW 之后的指令(指令 4~10),直至 SETR 指令进入 IDC 级,DDP 接收 IDC 级解码 SETR 的反馈信号并将分发模式切换回 Superscalar 模式。无论 B.32(指令 9)跳转到 Lable\_1 或 Lable\_2,后续指令将在 Superscalar 模式下执行。

| 分发模式                        | Superscalar 模式 | VLIW 模式 |
|-----------------------------|----------------|---------|
| 1 MOV.K.32(.XM) G3, #15     |                |         |
| 2 MOV.K.32(.XM) G4, #15     |                |         |
| 3 SETW.32(.YD)              |                |         |
| 4 ADD.32(.XA) G2, G3, G4    |                |         |
| 5 ADD.32(.XM) G3, G3, G4    |                |         |
| 6 ADD.32(.XD) G4, G3, G3    |                |         |
| 7 CMPE.32(.XD) G0, G2, G4   |                |         |
| 8 MOV.K.32(.XM) G0, #0      |                |         |
| 9 SETR.32(.YD)              |                |         |
| 10 [!G0] B.32(.XD) @Lable_2 |                |         |
| Lable_1:                    |                |         |
| 11 ADD.32(.XA) X22, X22, #1 |                |         |
| 12 B.32(.XD) @test_done     |                |         |
| Lable_2:                    |                |         |
| 13 ADD.32(.XA) X23, X23, #1 |                |         |
| 14 B.32(.XD) @test_fail     |                |         |

(a) Lily2双模式混合指令流模式切换



(b) 指令控制分发模块切换

图 3 分发模式切换

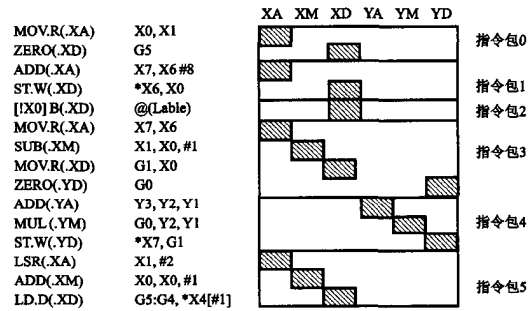
## 2.3 VLIW-SuperScalar 混合结构分发策略

### 2.3.1 VLIW 模式分发策略

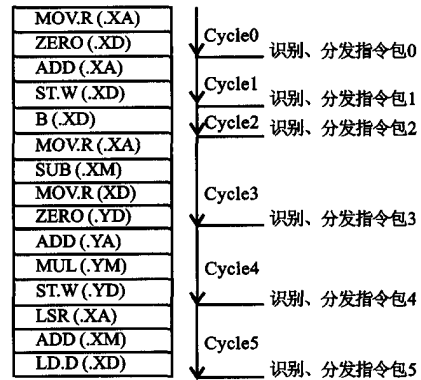
VLIW 模式下的指令调度由面向 Lily2 的 VLIW 模式的 Open64 编译器完成。编译器静态分配执行单元,并将每个并行指令组按照固定的执行单元顺序(XA, XM, XD, YA, YM, YD)升序排列成一个分发包。

列成一个分发包。

图 4(a)所示为编译器生成 VLIW 指令包的方法。Lily2 的 VLIW 模式分发策略继承于 Lily: VLIW 分发单元在指令流中检测按单元升序的连续指令组,从而识别出由编译器生成的各个分发包,并按序将其中的指令分发到对应的单元,如图 4(b)所示。该分发策略与编译器功能相配合,是典型的 VLIW 分发策略。由于寄存器数据依赖、硬件资源限制等问题都已经由编译器解决,在硬件设计中分发策略的实现较为简单。



(a) Lily2 VLIW编译器生成指令包



(b) Lily2 VLIW分发模块识别、分发指令包

图 4 VLIW 模式分发

### 2.3.2 Superscalar 模式分发策略

Lily2 的 Superscalar 模式的分发策略使用的是双发射设计,单次最多发射两条指令。当 Lily2 运行在 Superscalar 模式下,只有 X 簇的执行单元会被使用,相应地面向 Lily2 的 Superscalar 模式的 Open64 编译器仅使用了 X 簇的功能单元来生成可执行代码。

Lily2 的 Superscalar 模式为经典的 Superscalar 结构设计,编译器负责生成串行的指令序列。Lily2 的硬件部分实现指令的寄存器依赖检测、功能单元的冲突检测等功能,如图 5 所示。

寄存器依赖检测:

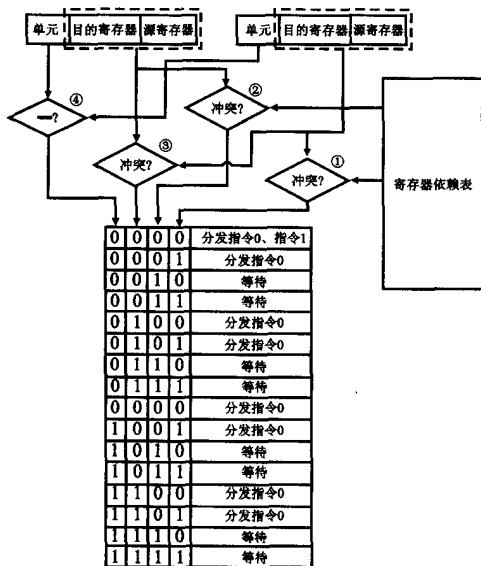


图5 Superscalar模式的指令冲突检测

为了实现程序中前后指令间的寄存器依赖检测, Lily2 的 DDP 流水级中为 Superscalar 模式引入了一张寄存器表. 在 Superscalar 模式下, 每一条被判定为可分发的指令都将被预解码, 得到其目的寄存器的编址和寄存器写入时间(最短 1 拍, 最长 3 拍). 寄存器编址被写入寄存器表, 相应的寄存器写入时间决定该寄存器编码在寄存器表中的留存时间. 在其留存期间, 后续待分发指令的源寄存器均需要与表中的留存项进行冲突检测, 作为后续指令是否可以被分发的部分决策, 如图 5 中①、②所示.

在分发决策过程中, 检测模块同样会对取指包中的前两条指令(指令 0 和指令 1)间的寄存器依赖进行检测, 以此决定是否同时分发两条指令, 如图 5 中③所示.

在 VLIW 和 Superscalar 两种模式的相互切换过程中, 分发单元都会暂停分发 3 周期(目的寄存器的延迟写入时间最长为 3 周期), 避免在模式切换后发生寄存器冲突. 采用这种方式的原因是 VLIW 模式下没有对寄存器依赖进行检查的硬件机制, 也不会将指令的目的寄存器写入寄存器表.

#### 执行单元冲突检测:

Lily2 的 Superscalar 模式在分发决策过程中对取指包前两条指令(指令 0 和指令 1)之间的执行单元冲突进行检测, 如图 5 中④所示. 如果存在执行单元冲突, 最多只有第一条指令(指令 0)会被分发.

#### 2.3.4 Superscalar 模式的 2bit 分支预测设计

Lily2 引入了 2bit 分支预测功能<sup>[4]</sup>, 该预测功

能运行在 Superscalar 模式下, 使处理器性能得到了很好的提升. 分支预测的功能模块主要分布于 IF1 和 EX1 流水级, 如图 6 所示. 在 IF1 中, 分支预测表用来记录跳转指令的目标地址. 当 IF1 级取得分支指令, 该模块根据分支预测表项对该指令的目标地址进行预测(根据目标指令在分支预测表中的有无选择下一 PC 值). 在 EX1 级中, 指令预测控制模块将 IF1 级预测得出的跳转地址和 IDC 级中译码得出的真实跳转地址进行对比以判断是否预测成功, 并将结果反馈给之前的流水级. 如果预测成功, 则流水线仍按原指令流顺序执行; 如果预测失败, 则需要清空流水线、重新调整分支预测表并按真实跳转地址重新取指令.

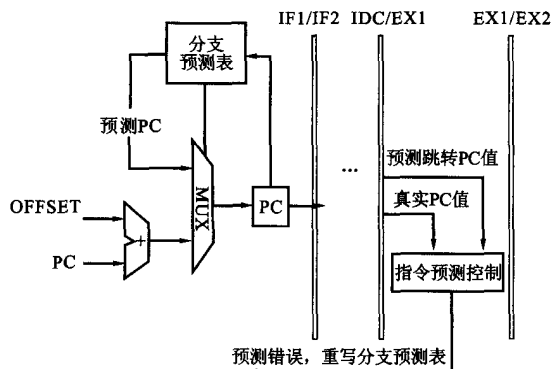


图6 2 bit 分支预测模块

### 3 性能评测

采用 TSMC 65 nm 工艺, Lily2 稳定工作在 310 MHz 频率下. 分两步对 Lily2 的性能进行评测.

#### 3.1 对 Lily2 的 VLIW 模式的评测

表 1 所示为 Lily2 工作在 VLIW 模式下(Lily2-VLIW)的性能评测, 该评测使用了多个面向信号处理的 benchmark kernel<sup>[5]</sup>, 并与 3 款主流的 DSP 核心进行对比. 可以看出拥有 6 发射的 Lily2-VLIW 在信号处理领域有良好的性能表现.

表1 DSP 核心的性能评测比较(cycles)

| DSP core                            | Ceva  | StarCore | TI c64x | Lily2— |
|-------------------------------------|-------|----------|---------|--------|
| benchmark                           | x1640 | C1400    |         | VLIW   |
| Two biquad IIR                      | 9     | 9        | 16      | 13     |
| Vector dot product                  | 19    | 16       | 25      | 19     |
| Vector add                          | 18    | 19       | 27      | 21     |
| vector maximum                      | 22    | 27       | 36      | 19     |
| 40-sample, 16-tap complex block FIR | 1 333 | 675      | 674     | 448    |
| Radix-2, 256-point complex FFT      | 1 248 | 1 631    | 1 246   | 1 472  |

3.2 用 Dhrystone Benchmark 对 Lily2 的混合双模式性能进行评测

该评测的最终可执行程序分两部分实现:

使用基于 Open64 的 Lily2 编译器将 Dhrystone 编译为 Lily2 Superscalar 模式的目标程序.

将 Dhrystone 中使用到的 strcmp、strcpy 等函数实现为 Lily2 VLIW 模式的库函数,并对其进行手工优化.

使用 Lily2 的 binutils 二进制工具进行链接,生成最终用于仿真的可执行程序. 表 2 所示为 Lily2 与主流 ARM 处理器核的 Dhrystone 评测对比 (DMIPS/MHz). 从评测结果来看,采用 Superscalar-VLIW 混合双模式结构的 Lily2 处理器具有优秀的性能表现,甚至可以比肩最出色的主流通用处理器.

表 2 Dhrystone 评测结果比较

| Microprocessor Core  | Dhrystone result<br>(DMIPS/MHz) |
|----------------------|---------------------------------|
| ARM7-ARM710a         | 0.68                            |
| ARM8-ARM810          | 1.16                            |
| ARM11-ARM1176JZ(F)-S | 1.25                            |
| Lily2                | 1.26                            |
| Cortex-A5            | 1.57                            |

4 结束语

本文介绍了一款 Superscalar-VLIW 混合双模式结构处理器的设计,该处理器能够在单指令流的执行过程中灵活切换执行模式,从而充分发挥两种执行模式各自的优势. VLIW 模式下,该处理器利用

软件层面的静态调度以及最多 6 条指令的并行发射能力,在与主流 DSP 核的评测对比中显示出了在信号处理领域的优秀性能. 对该处理器两种模式混合执行的性能评测结果表明,该 DSP 的综合性能甚至可以媲美最优秀的主流通用处理器.

参考文献:

[1] Joseph Fisher. Very long instruction word architecture and the ELI-512 [C]//Computer Architectur (ISCA-10), Computer Architecture News. New York, 1983, 11(3):140-150.

[2] Michael J Flynn, Kevin W Rudd. Parallel architectures [J]. ACM Computing Surveys, 1996, 28(1): 67-70.

[3] Shen Zheng, He Hu. Architecture design of a variable length instruction set VLIW DSP[J]. Tsinghua Science and Technology, 2009,14(5):561-569.

[4] John K F Lee, Alan Jay Smith. Branch prediction strategies and branch target buffer design[J]. Computer, 1984, 17(1):6-22.

[5] Berkeley Design Technology, Inc. BDTI benchmark results[EB/OL]. (2005-11-10)[2003-11-10]. <http://www.bdti.com>.

作者简介:

陈峰扬 男,(1988-),硕士. 研究方向为微处理器结构设计、微处理器结构软件仿真模型.

杜 勇 男,(1989-),硕士. 研究方向为低微处理器结构设计.

郭德源 男,(1984-),硕士,工程师. 研究方向为软件工具链设计.

何 虎 男,(1974-),副教授. 研究方向为高性能数字信号处理器系统设计.