

Косарев Н.Е.^{1,2}

¹ Московский физико-технический институт

² ЗАО «Интел А/О»

Исследование алгоритмов выбора инструкций в устройстве планирования для микропроцессора с векторным счетчиком инструкций

За последние два десятилетия исследователи и разработчики микропроцессоров достигли значительного увеличения производительности за счет одновременного исполнения нескольких инструкций в рамках одного программного потока (использование параллелизма на уровне инструкций, ILP). Однако в настоящее время становится очевидным, что ограничения на использование ILP не позволяют в ближайшем будущем выпустить процессор, который бы был значительно быстрее (особенно для целочисленного кода), проще или энергоэффективнее [1]. Тенденции в разработке микропроцессоров изменяются в сторону распределенных архитектур, работающих в тесном взаимодействии с компилятором [2], [3].

Одной из реализаций данного типа архитектуры является микропроцессор с векторным счетчиком инструкций. Увеличение производительности достигается за счет программного разбиения потока инструкций на несколько цепочек (strands), каждая из которых обрабатывается параллельно, и кластеризации тесно связанных исполнительных устройств. Цепочкой называется последовательность инструкций, образуемая компилятором, в которой каждая следующая инструкция зависит от предыдущей, в то время как зависимости по данным между инструкциями из разных цепочек минимизированы. Инструкции в каждой цепочке исполняются в прямом порядке (in-order execution), что отличает их от потоков (threads) в современном суперскалярном процессоре, инструкции в которых исполняются в произвольном порядке (out-of-order execution).

Устройство планирования микропроцессора с векторным счетчиком инструкций распределяет инструкции из нескольких цепочек на блоки исполнительных устройств по мере готовности операндов. В процессе выполнения программы число готовых инструкций может существенно превышать количество исполнительных устройств, которое в значительной мере ограничено длинами линий связи и обратной передачи

(bypass). В результате возникает конфликт на уровне использования ресурсов (resource conflict): на каждое исполнительное устройство претендует несколько инструкций.

Для разрешения выше обозначенного конфликта необходим оптимальный динамический алгоритм выборки готовых инструкций на исполнение, который бы максимизировал IPC (число инструкций, исполняемых за один такт) и был относительно прост для реализации в аппаратуре. Стоит отметить, что прямое использование алгоритма, используемого в современных суперскалярных микропроцессорах, который отдает приоритет самым «старым» инструкциям, невозможно, так как он основан на изначально известном аппаратуре программном порядке, который теряется при разбиении потока инструкций на отдельные цепочки. Сравнение динамических алгоритмов выборки инструкций для типовых out-of-order суперскалярных архитектур описано в [4].

В настоящей работе были исследованы следующие методы выборки инструкций для микропроцессора с векторным счетчиком инструкций:

- Циклическая выборка (round robin)
- Выборка в фиксированном порядке (fixed priority)
- Выборка в произвольном порядке (random)
- Циклическая выборка с произвольного начала (random rotate)
- Выборка в программном порядке (восстанавливается аппаратурой с использованием дополнительных тегов, передаваемых компилятором с каждой инструкцией)
- Выборка цепочек с наименьшей длиной (определяется динамически в процессе выполнения программы)
- Выборка цепочек с динамически изменяемым приоритетом (изменяется в зависимости от исполнения цепочки; первоначальное значение приоритета может быть задано компилятором)

Хотя эти методы не являются исчерпывающими, они представляют собой широкий диапазон возможных эвристик. Для некоторых методов выборки инструкций была произведена оценка сложности реализации в аппаратуре (анализ алгоритма на уровне логических элементов и оценка критического пути для заданной частоты микропроцессора).

Каждый алгоритм был запрограммирован в потактовой модели микропроцессора с векторным счетчиком инструкций. В докладе приводятся

результаты сравнения производительности микропроцессора на основе данных потактовой модели с использованием вышеперечисленных алгоритмов выборки на двух группах тестов. Первая группа включает в себя тесты, ориентированные на получение пиковой производительности модели (максимальная загрузка арифметических устройств и устройств обращения в память). Вторая группа представляет собой выборочные тесты производительности из набора SPEC2000, разбитые с использованием компилятора на цепочки инструкций.

СПИСОК ЛИТЕРАТУРЫ

1. *Hennesy J.L., Patterson D.A.* Computer Architecture: A Quantitative Approach. Forth edition. — San Francisco: Morgan Kaufmann Publishers, 2006. — Chapter 3.
2. *Kim H.S., Smith J.E.* An Instruction Set and Microarchitecture for Instruction Level Distributed Processing. // The 29th International Symposium on Computer Architecture (ISCA 2002). — 2002.
3. *Caulfield I.* Complexity-effective superscalar embedded processors using instruction-level distributed processing. // Technical Report UCAM-CL-TR-707, Computer Laboratory, University of Cambridge. — 2007.
4. *Butler M., Patt Y.N.* An Investigation of the Performance of Various Dynamic Scheduling Techniques. // In Proceedings of the 25th Annual International Symposium on Microarchitecture (MICRO'1992). — 1992. — P. 1-9.