

Повышение энергоэффективности аппаратуры подкачки команд суперскалярного микропроцессора в режиме одновременной многопоточности

И. В. Смирнов^{1,2}, О. В. Шимко^{1,2}

¹Московский физико-технический институт (государственный университет)

²АО «Интел А/О»

В настоящее время наибольшее распространение в современных высокопроизводительных ЭВМ получили суперскалярные микропроцессоры с поддержкой технологии одновременной многопоточности (simultaneous multi-threading, SMT), позволяющей повысить эффективность утилизации доступных ресурсов процессора за счет дополнительного уровня параллелизма на уровне потоков исполнения (thread-level parallelism, TLP).

Одним из основных методов программирования для таких систем является создание параллельных single-program multiple-data приложений (SPMD-приложений), особенностью которых является исполнение всеми потоками одного и того же алгоритма над различным набором данных. Несмотря на то, что все потоки приложения исполняют один и тот же код, аппаратура подкачки команд современных многопоточных микропроцессоров закачивает инструкции независимо для каждого потока, что приводит к снижению энергоэффективности микропроцессора.

Для решения проблемы избыточной загрузки команд могут использоваться программно-аппаратные [1] и чисто аппаратные [2] техники, основанные на синхронизации программных адресов (program counter, PC) потоков исполнения и дальнейшем объединении идентичных инструкций аппаратурой подкачки команд. Преимуществом аппаратного подхода является отсутствие требования к перекомпиляции программ и как следствие более широкая область применимости. Основу аппаратного механизма синхронизации составляет буфер истории ветвления (branch history buffer, ВНВ), хранящий адреса переходов инструкций ветвления, встреченных каждым потоком исполнения. В случае, если один и тот же адрес перехода был пройден каждым потоком, включается режим “синхронизации”, в котором “отстающий” поток получает приоритет на подкачку своих инструкций. В момент, когда оба исполняемых потока указывают на один и тот же программный адрес, аппаратура подкачки команд переходит в режим “объединения”, в котором только один из потоков исполнения осуществляет загрузку команд из кэша инструкций микропроцессора. Все инструкции содержат дополнительный признак-идентификатор, сообщающий аппаратуре аллокации о том, что инструкция является общей для двух потоков.

Данная работа посвящена адаптации и усовершенствованию аппаратного механизма синхронизации и объединения идентичных инструкций [2] для двухпоточного микропроцессора суперскалярной архитектуры с отдельными для каждого потока исполнения промежуточными буферами аппаратуры подкачки команд (рис. 1). Разделение промежуточных буферов позволяет микропроцессору продолжать загрузку (аллокацию) команд в аппаратуру внеочередного исполнения даже в случае остановки одного из потоков (например, из-за нехватки ячеек в буфере переупорядочивания инструкций, выделенных для данного потока).

Необходимым условием функциональной корректности рассматриваемого механизма, гарантирующем программный порядок следования инструкций в конвейере, является добавление промежуточного режима очистки всех очередей инструкций одного из потоков (режим “освобождения”) в момент перехода из режима “синхронизации” в режим “объединения”. Аппаратный алгоритм синхронизации, дополненный режимом “освобождения”, рассматривается в качестве базового технического решения для микропроцессора с отдельными промежуточными буферами.

Результаты моделирования, проведенного на потактовом симуляторе суперскалярного микропроцессора, показали, что данный механизм способен объединить в среднем 17% идентичных команд, снизив потребление энергии микропроцессора в среднем на 3.8% (с учетом энергопотребления дополнительной аппаратуры, составляющего порядка 0.4%). Однако общее снижение производительности является существенным и составляет порядка 2.8% (рис. 2). Основной причиной уменьшения производительности является замедление одного из потоков в случае длительного нахождения в режиме синхронизации.

На основании полученных данных авторами был разработан ряд модификаций, нацеленных на снижение негативного эффекта режима синхронизации потоков исполнения. К данным улучшениям относятся принудительный выход из режима синхронизации по истечении заданного временного

интервала, сохранение инструкций одного из потоков в случае ошибки предсказания перехода в другом потоке, а также принудительное разделение объединенных инструкций в случае, если один из потоков не имеет свободных ресурсов.

Результаты моделирования показали, что модифицированный алгоритм удаляет в среднем 16% идентичных команд, сокращая общее энергопотребление микропроцессора на 3.5% при существенно меньшем, по сравнению с базовым механизмом объединения инструкций, негативном влиянии на производительность: среднее снижение производительности составляет 1.1%. Благодаря наличию механизма принудительного выхода из режима синхронизации, модифицированный алгоритм способен отсекав приложения с слишком долгим этапом синхронизации, сокращая среднее время нахождения микропроцессора в режиме синхронизации с 13% до 8%.

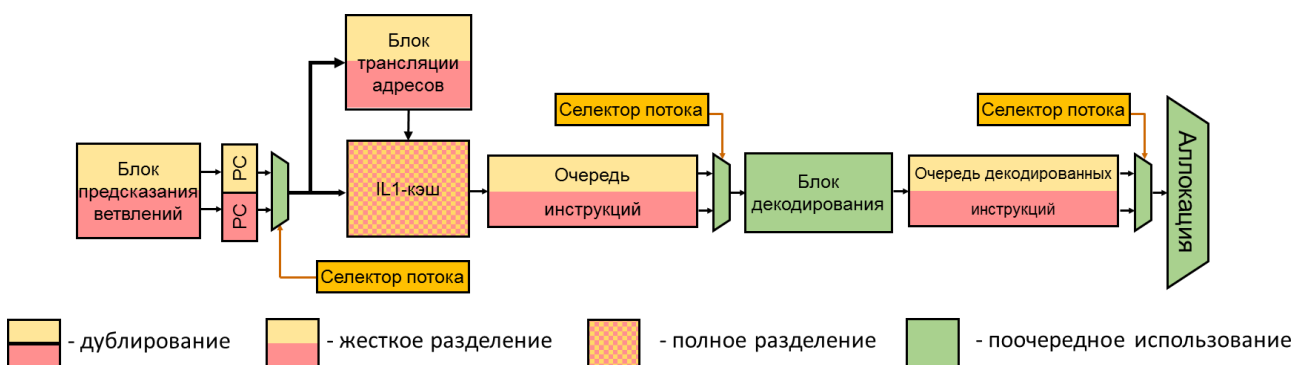


Рис. 1. Схема разделения ресурсов аппаратуры подкачки команд моделируемого микропроцессора в режиме многопоточного исполнения

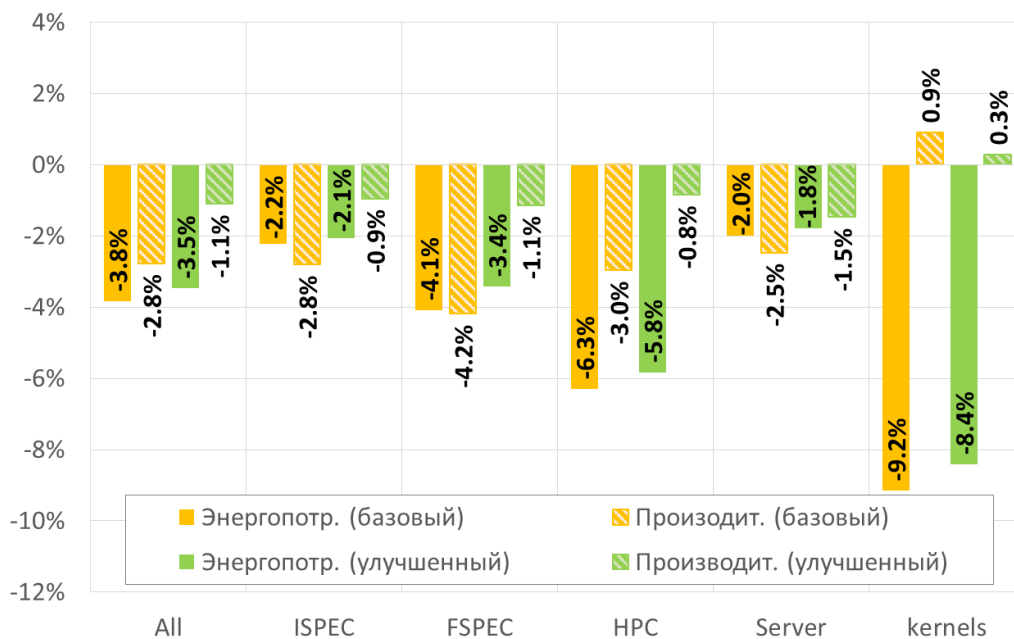


Рис. 2. Изменение энергопотребления и производительности моделируемого микропроцессора при применении базового и улучшенного алгоритмов объединения идентичных инструкций

Литература

1. Rakvic R., González J., Cai Q., Chaparro P., Magklis G., González A. Energy efficiency via thread fusion and value reuse // IET Computers & Digital Techniques. 2010. V. 4, N 2. P. 114-125.
2. Long G., Franklin D., Biswas S., Ortiz P., Oberg J., Fan D., Chong F. Minimal Multi-Threading: Finding and Removing Redundant Instructions in Multi-Threaded Processors // Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture. 2010. P. 337-348.