

Виртуализация тредовых устройств в мультитредово-поточковом процессоре

А.С. Фролов, Л.К. Эйсымонт

*ОАО «НИЦЭВТ», Москва
frolov@nicevt.ru, verger@nicevt.ru*

1. Введение

В ОАО «НИЦЭВТ» ведутся работы по созданию отечественного мультитредово-поточкового микропроцессора (МТП-процессор) для суперкомпьютера стратегического назначения (СКСН) «Ангара» с глобально адресуемой памятью [1,2,3]. СКСН «Ангара» предназначен для решения задач с интенсивной нерегулярной работой с памятью большого объема (от сотен Тбайт до нескольких Пбайт).

В данной работе рассматривается расширение архитектуры МТП-процессора для поддержки механизма виртуализации тредовых устройств. Виртуализация тредовых устройств — механизм, позволяющий за счет аппаратной поддержки переключения контекста реальных (аппаратных) тредовых устройств искусственно увеличить их количество, видимое задачам. Переключение виртуальных тредовых устройств происходит во время выполнения обращений к памяти, вызвавших промах в кэш-данных или ушедших по коммуникационной сети к другим узлам, то есть в случае обращений к памяти удаленных узлов.

Таким образом, при достаточно высокой скорости переключения виртуализация тредовых устройств могла бы ограничить количество реальных тредовых устройств в МТП-процессоре (8-16 устройствами), и получить максимальную реальную производительность задачи за счет большого количества виртуальных тредовых устройств (64-128 устройств). Виртуализация могла бы решить и другую задачу, при количестве аппаратных тредовых устройств 64-128 увеличить их выборку до 256-512, но это будет целесообразно, если пропускная способность интерфейсов с памятью и сетью окажется для них достаточной.

Известны проекты мультитредовых микропроцессоров с поддержкой виртуализации тредовых устройств: разрабатываемый в Intel микропроцессор EXOCHIP [4], в калифорнийском университете (UoC, San Diego) — микропроцессоры BMT и STMP [5], когда за счет виртуализации на процессорах с малой мультитредовостью увеличивается количество тредовых устройств.

2. Архитектура МТП-процессора

МТП-процессор — базовый элемент СКСН «Ангара». Работа ведется по двум вариантам МТП-процессора: упрощенный — микропроцессор J7; полный — микропроцессор J10. Эти микропроцессоры отличаются функциональными возможностями и характеристиками, но программно совместимы от J7 к J10. Рассматриваемый механизм виртуализации тредовых устройств одинаково может быть реализован как в J7, так и в J10. Далее ограничимся вариантом J7, его упрощенная структура J7 приведена на Рис. 1.

Микропроцессор J7 состоит из двух мультитредовых ядер с 64 аппаратными тредовыми устройствами в каждом и возможностью работы в каждом ядре с четырьмя доменами защиты программ и данных. Каждому тредовому устройству доступны 32 64-разрядных регистров общего назначения, 32 64-разрядных регистров для хранения чисел с плавающей точкой, 8 однобитовых регистров признаков, 8 36-разрядных регистров для хранения адресов передачи управления. В одном ядре в J7 может одновременно выполняться до 512 команд обращений к памяти (по 8 обращений на одно тредовое устройство), а во всем процессоре до 1024 одновременно обрабатываемых обращений.

Это обеспечивает толерантность к задержкам выполнения операций с памятью и коммуникационной сети.

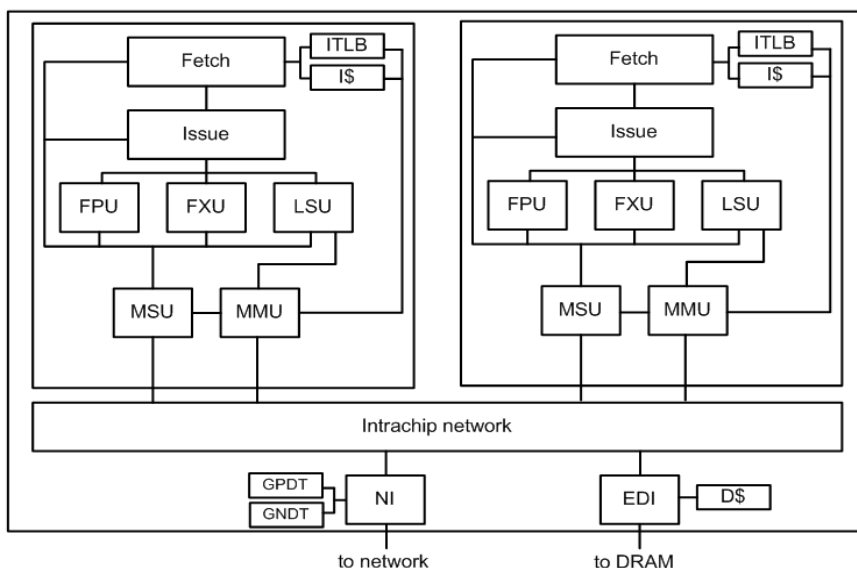


Рисунок 1. Структурная схема МТП-процессора J7.

3. Виртуализация тредовых устройств

Тредовое устройство может быть виртуализировано, то есть представлено в виде некоторой абстракции, обладающей всеми свойствами аппаратного тредового устройства, но не являющейся частью аппаратуры. Выделяются три уровня абстракции: программный тред, виртуальное тредовое устройство и аппаратное тредовое устройство (см. Рис. 2).

Отображение программных тредов на виртуальные тредовые устройства (M:N) осуществляет библиотека программных тредов `jthreads`. Отображение виртуальных тредовых устройств на аппаратные (N:P) осуществляет механизм виртуализации тредовых устройств. Если виртуализация тредовых устройств не используется, тогда программные треды отображаются непосредственно на аппаратные тредовые устройства (M:N=P), так и сделано в существующем варианте J7. Далее рассмотрим, как можно сократить количество аппаратных тредов за счет виртуализации.

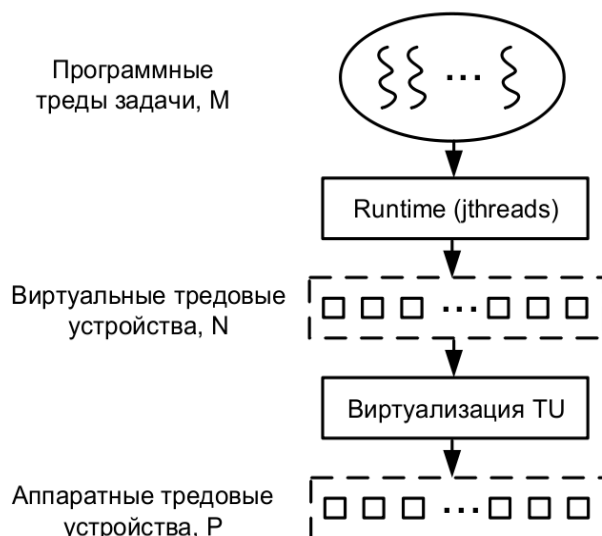


Рисунок 2. Двухуровневая виртуализация тредов.

3.1 Виртуальные тредовые устройства

Виртуальное тредовое устройство обладает таким же набором регистров, что и аппаратное. На Рис. 3 приведена диаграмма состояний виртуального тредового устройства. Как видно из рисунка, виртуальное тредовое устройство может находиться в одном из следующих состояний:

1. *Свободно* — виртуальное тредовое устройство не используется, на него не отображен ни один программный тред. Такие виртуальные тредовые устройства считаются недействительными и не могут быть загружены на аппаратные тредовые устройства.
2. *Пассивно (готово)* — виртуальное тредовое устройство используется приложением, то есть на него отображен один из программных тредов, и при первой возможности будет загружено на аппаратное тредовое устройство.
3. *Пассивно (не готово)* — виртуальное тредовое устройство используется приложением, но не готово к тому, чтобы стать активным, по причине незавершенности выданных ранее LSU-команд (команд обращения к памяти и удаленного вызова процедур - RPC).
4. *Активно* — виртуальное тредовое устройство выполняется на одном из аппаратных тредовых устройств в ядре МТП-процессора.



Рисунок 3. Диаграмма состояний виртуального тредового устройства.

3.2 Трансляция номера виртуального тредового устройства

Состояние неактивных виртуальных тредовых устройств отображено в специальной таблице Virtual Thread Table (VTT). Размер записи VTT составляет 1 Кбайт. В общем случае VTT размещена в DRAM памяти, однако в варианте реализации с ограниченным количеством виртуальных тредовых устройств возможно размещение VTT в сверхбыстрой SRAM памяти внутри ядра процессора.

Для вычисления адреса записи VTT, содержащей дескриптор заданного виртуального тредового устройства (см. Рис. 4), к значению поля BASE регистра DTD (регистр уровня домена защиты), смещенному на 20 разрядов влево (начало VTT), добавляется номер виртуального тредового устройства, смещенный на 10 разрядов влево. Процедура трансляции может выполняться как программно, так и аппаратно.

Запись в VTT содержит следующие поля: (1) *Valid* — флаг действительности виртуального тредового устройства (действительно/недействительно); (2) *Active* — флаг активности виртуального тредового устройства (активно/неактивно); (3) *Ready* — флаг готовности виртуального тредового устройства; (4) *PTN* — номер аппаратного тредового устройства, на которое загружено виртуальное тредовое устройство; (5) *Virtual TU State* — состояние виртуального тредового устройства (TSW, TDS, TRS, r/f/t/q-регистры).

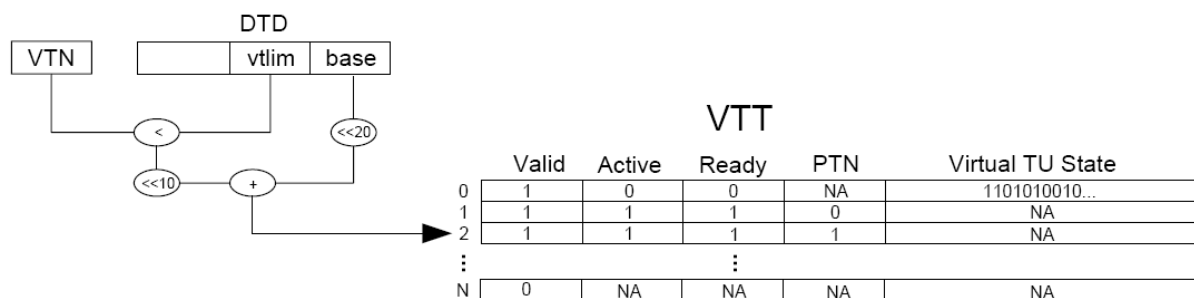


Рисунок 4. Трансляция номера виртуального тредового устройства.

3.3 Выполнение виртуальных тредовых устройств

Каждое виртуальное тредовое устройство имеет 10-разрядный виртуальный номер, однозначно идентифицирующий его в пределах домена защиты. Для хранения номера загруженного виртуального тредового устройства в аппаратном тредовом устройстве введен регистр VTN (см. Рис. 5).

Все LSU-команды выдаются на выполнение вместе с номером виртуального тредового устройства, полученным из регистра VTN. В блоке LSU этот номер запоминается на время выполнения команды вместе с остальной информацией о ней (код операции, операнды, номер аппаратного тредового устройства, выдавшего команду, номер регистра результата и т.д.). На выполнение остальных команд (арифметические операции, команды переходы и др.) режим виртуализации не влияет.

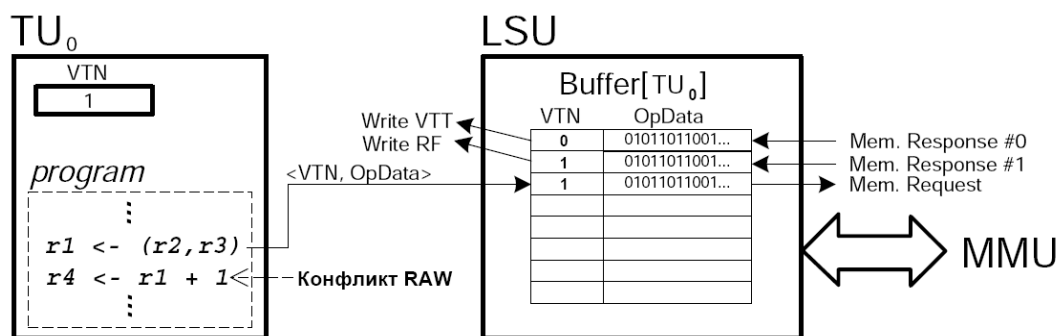


Рисунок 5. Выдача и завершение LSU-команд в режиме виртуализации тредовых устройств.

В буфере LSU для каждого аппаратного тредового устройства выделяется 8K записей, где K — отношение максимального количества виртуальных тредовых устройств к количеству аппаратных тредовых устройств, размещенных в ядре процессора. Таким образом, если одно аппаратное тредовое устройство до виртуализации могло выдать 8 LSU-команд, то теперь это число увеличивается до 8K.

Механизм смены виртуального тредового устройства на аппаратном тредовом устройстве состоит в следующем.

Если при выдаче команды с аппаратного тредового устройства был дан отказ по неготовности регистра (конфликт read-after-write, RAW), и если этот регистр был зарезервирован LSU-командой, то происходит снятие виртуального тредового устройства с аппаратного и затем загрузка другого виртуального тредового устройства. При этом выданные ранее LSU-команды снятого виртуального тредового устройства не прерываются, а продолжают "висеть" в буфере LSU и реально выполняются.

При завершении выполнения LSU-команд происходит сравнение содержимого регистра VTN с полем VTN в записи LSU буфера. Если значения совпадают, команда

завершается стандартным образом — результат записывается в регистровый файл. Если значения не совпадают, результат записывается непосредственно в таблицу VTT. После завершения последней LSU-команды, виртуальное тредовое устройство будет переведено в состояние «Пассивно (готово)» и при первой возможности загружено на аппаратное тредовое устройство для дальнейшего выполнения.

4. Исследование эффективности

Оценка эффективности механизма виртуализации тредовых устройств на разработанной программной имитационной модели СКЧН «Ангара» проводилась на тесте умножения разреженной матрицы на вектор (SpMV). Для теста SpMV характерна интенсивная нерегулярная работа с памятью. Размер матрицы — 75000x75000, среднее количество ненулевых элементов в строке — 6800 (9,1%). Формат хранения разреженной матрицы — CRS.

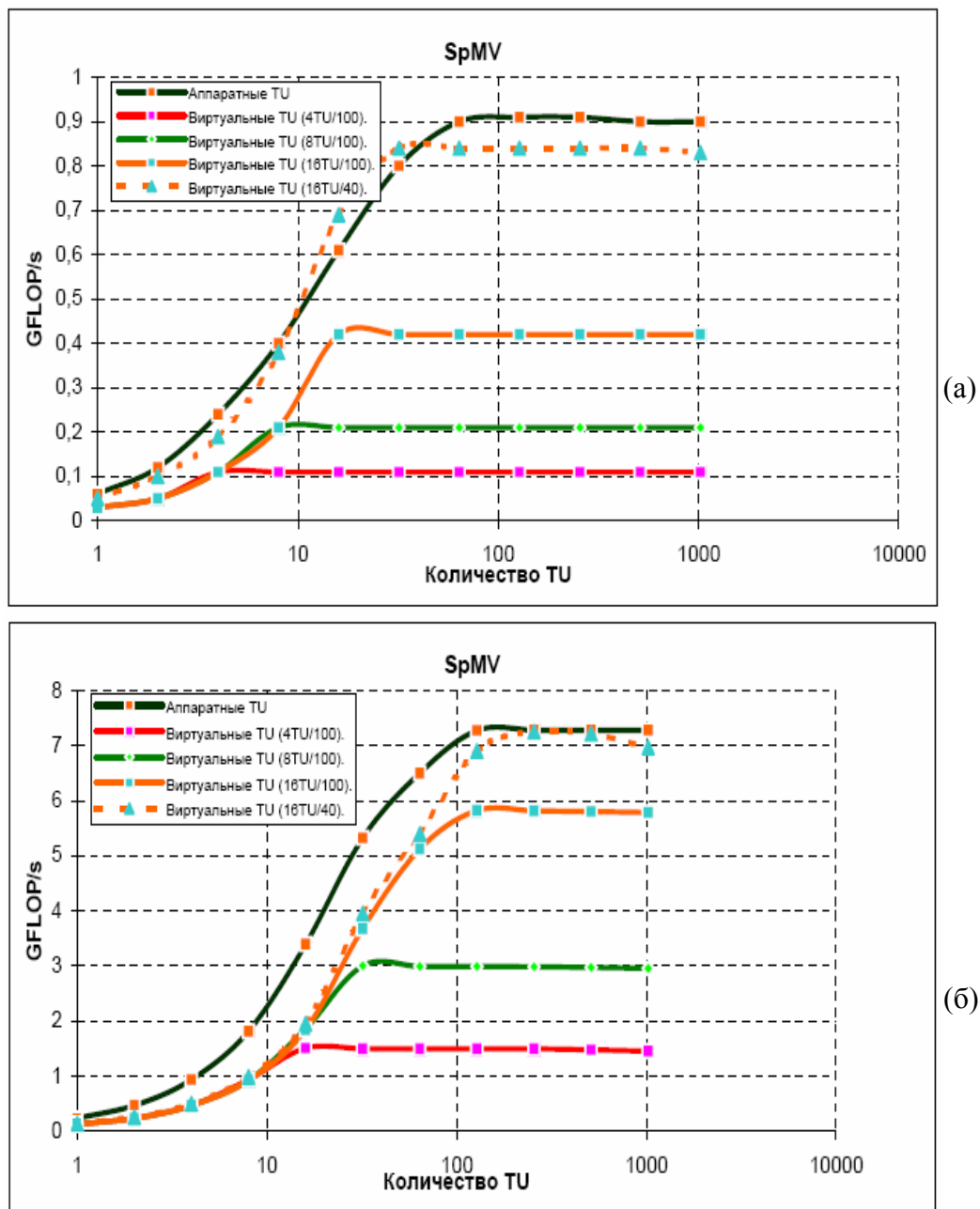


Рисунок 6. Результаты выполнения теста SpMV на одном (а) и 16 МТП-процессорах (б)

Измерялись следующие величины:

- 17) $P(tu)$ — зависимость производительности от количества аппаратных тредовых устройств (tu) в режиме без виртуализации;
- 18) $P^v(vtu, tu, t)$ — зависимость производительности от количества виртуальных тредовых устройств (vtu), количества аппаратных тредовых устройств (tu) и времени переключения виртуальных тредовых устройств (t).

На Рис. 6 приведены результаты, полученные на тесте SpMV. Механизм виртуализации тредовых устройств на одном МТП-процессоре (см. Рис. 6(а)) сравним по эффективности с полностью аппаратной мультитредовой архитектурой только в случае, если время переключения виртуальных тредовых устройств не превышает 40 тактов. При этом $P^v(1024, 16, 100)$ оказалось всего на 7% меньше, чем $P(1024)$.

Результаты SpMV, полученные на 16-узловой конфигурации СКСН «Ангара» (см. Рис. 6(б)), показали, что при t , равном 100 тактам, 16 тредовых устройств с включенной виртуализацией, по производительности уступают аппаратным тредовым устройствам менее 20%. В случае, если время переключения составляет 40 тактов, то значение $P^v(256, 16, 40)$ оказалось равным $P(256)$.

5. Заключение

Основные результаты работы состоят в следующем:

1. Предложен механизм виртуализации тредовых устройств, позволяющий практически при сохранении производительности сократить количество тредовых устройств в 4-8 раз. Сокращение количества тредовых устройств упрощает микроархитектуру МТП-процессора, в особенности реализацию регистровых файлов.
2. Механизм виртуализации тредовых устройств реализован в имитационной модели СКСН «Ангара». При этом обеспечена полная прозрачность виртуализации, что позволило провести тестирование с использованием ранее разработанных программ.
3. Проведено оценочное тестирование разработанного механизма на тесте SpMV. Эксперимент показал, что для достижения положительного эффекта от применения виртуализации тредовых устройств на одном МТП-процессоре время переключения виртуальных тредовых устройств должно быть не более 40 тактов процессора, для 16-узловой конфигурации суперкомпьютера «Ангара» — не более 100 тактов. При этом количество аппаратных тредовых устройств может быть уменьшено с 64 до 16 без потери производительности.

Литература

1. Слуцкий А., Эйсымонт Л. Российский суперкомпьютер с глобально адресуемой памятью. – Открытые системы, 2007, №9, с.42–51.
2. Митрофанов В., Слуцкий А., Эйсымонт Л. Суперкомпьютерные технологии для стратегически важных задач. - ЭЛЕКТРОНИКА: Наука, Технология, Бизнес, 2008, №7, с.66-79.
3. Семенов А. С., Соколов А.А., Эйсымонт Л. К. Архитектурные особенности и реализация глобально адресуемой памяти мультитредово-поточкового суперкомпьютера. // Электроника: Наука, Технология, Бизнес. – 2009. – №1. – С. 50-61.
4. Wang, P. H., Collins, J. D., China, G. N., Lint, B., Mallick, A., Yamada, K., and Wang, H. 2007. Sequencer virtualization. In Proceedings of the 21st Annual international Conference on Supercomputing (Seattle, Washington, June 17 - 21, 2007). ICS '07. ACM, New York, NY, 148-157.
5. Brown, J. A. and Tullsen, D. M. 2008. The shared-thread multiprocessor. In Proceedings of the 22nd Annual international Conference on Supercomputing (Island of Kos, Greece, June 07 - 12, 2008). ICS '08. ACM, New York, NY, 73-82.