

Магистерская диссертация

**Аппаратная поддержка
порядка операций обращения
в память в системе на
кристалле «Эльбрус-2S»**

Выполнил: Петрыкин Д.А., ФРТК, 613 гр.
Научный руководитель: к.т.н. Груздов Ф.А.
Рецензент: Слесарев М.В.

Москва 2012

Семантика памяти в мультипроцессорах

- Мультипроцессор – многопроцессорная система с общей памятью
- Хаотичная борьба процессоров за доступ к модулям памяти
- Хранение копий данных в кэш-памяти
- Правила работы с общим ресурсом
- Модель согласованности памяти (memory consistency model) – совокупность всех правил работы процессоров с памятью
- Свойства сильных моделей:
 - Строгий порядок обращений в память – неявная синхронизация системы
 - Простота реализации
- Свойства слабых моделей :
 - Легче масштабируются
 - Больше возможностей для параллелизма

Архитектура x86

- Архитектура типа CISC (Complex Instruction Set Computer)
- Последовательная модель согласованности памяти
 - Порядок обращений в память может отличаться от программного, определяется аппаратурой и все процессоры видят его одинаково
- Принципы для однопроцессорной системы:
 - Чтения не обгоняют другие чтения
 - Записи не обгоняют предыдущие чтения
 - Записи не обгоняют другие записи
 - Чтения могут обгонять предыдущие записи по непересекающимся адресам
- Принципы для многопроцессорной системы:
 - При обращениях отдельного процессора к памяти применяются принципы однопроцессорной системы
 - Записи одного процессора не упорядочены по отношению к записям из других процессоров
 - Порядок выполнения обращений в память подчиняется принципам причинности (транзитивная видимость)
 - Любые две записи видны всем процессорам в одинаковом порядке

Проблема совместимости с x86

- Архитектура x86 – лидер на рынке продаж
- Написано огромное количество приложений под x86
- Разработчики других архитектур стремятся обеспечить корректное исполнение всех приложений x86 на своих машинах
- Программные и аппаратные средства поддержки совместимости с Intel x86

Решение проблемы в других архитектурах

- Itanium (IA-64):
 - Тип EPIC (Explicitly Parallel Instruction Computing)
 - Слабая модель согласованности памяти
 - Аппаратная трансляция всех команд x86
 - Дополнительные стадии в конвейере команд
 - Набор префиксов для последовательного исполнения команд
- Transmeta:
 - Тип EPIC (Explicitly Parallel Instruction Computing)
 - Слабая модель согласованности памяти
 - Использование Code Morphing Software
 - Оптимизация транслированного кода для повышения производительности
 - Использование программно-аппаратных контрольных точек и возможность восстановления системы
 - Использование «теневых» рабочих регистров
 - Адаптируемая ретрансляция программного кода
- Godson:
 - Тип RISC (Reduced Instruction Set Computer)
 - Последовательная модель согласованности памяти
 - Поддержка более 200 инструкций x86 в системе команд
 - Аппаратные механизмы для «ускорения» бинарной трансляции (EFlags, регистровый стек для чисел с плавающей точкой, режим сегментной адресации и другие)

Место проблемы в архитектуре «Эльбрус»

- Тип VLIW (Very Long Instruction Word)
- Слабая модель согласованности памяти
- Статическое планирование команд
- Совместимость с Intel x86:
 - Требуются дополнительные средства синхронизации
 - Используются не все возможности широкой команды
 - Потери производительности
- Наряду с программными требуются аппаратные механизмы поддержки совместимости с архитектурой Intel x86

Решение проблемы в проекте «Эльбрус-S»

- Реализация архитектуры «Эльбрус» в системе с четырьмя одноядерными процессорами
- Поддержка некоторых операций Intel x86 в системе команд
- Реализация управляющих регистров Intel x86
- Использование расширенного формата представления чисел с плавающей точкой
- Оптимизация процессорного доступа к переменным, механизм преимущественного хранения данных в регистровом файле и устройство Memory Lock Table (MLT)
- Организация виртуальной памяти аналогично Intel x86
- Поддержка порядка операций и модели памяти Intel x86 только для однопоточных приложений:
 - Записи упорядочены относительно других записей и чтений
 - Порядок чтений не соблюдается

Требования к проекту «Эльбрус-2S»

- Реализация архитектуры «Эльбрус» в системе с четырьмя процессорными узлами
- Каждый узел включает в себя по четыре независимых ядра
- Высокопроизводительная совместимость с x86:
 - Требуется обеспечить корректную работу многопоточных x86 программ
 - Потери производительности не должны быть существенными

Анализ задачи

- Основные аппаратные механизмы для совместимости с Intel x86 реализованы в проекте «Эльбрус-S»
- Для корректной работы неявных средств синхронизации необходимо поддержать модель памяти Intel x86:
 - Почти все принципы однопроцессорной системы Intel x86 реализованы в проекте «Эльбрус-S»
 - Почти все принципы многопроцессорной системы Intel x86 выполняются при поддержке когерентности данных в кэш-памяти
 - Строгий порядок чтений понижает почти на порядок производительность архитектуры «Эльбрус»
 - **Соблюдение порядка чтений – основная проблема**

Принятое в проекте решение

Поддержка порядка чтений

- Порядок чтений важен только тогда, когда считываемые переменные модифицируются другим ядром
- Модификация переменной другим ядром – конфликт очередности обращений в память
- Программное отслеживание конфликтов очередности
 - Аппаратура работает в неоптимизированном режиме
 - Работа с общими данными на разных ядрах может быть разнесена по времени с помощью программных средств
- Аппаратное отслеживание конфликтов очередности
 - Аппаратура работает в оптимизированном режиме
 - Использование принципов транзакционной памяти
 - При обнаружении конфликта транзакция выполняется заново
 - Преимущество транзакций: относительно простая реализация
 - Недостаток транзакций: частые конфликты приводят к существенным потерям производительности

Принятое в проекте решение

Положения неоптимизированного режима

- Упорядочивание записей относительно других записей и чтений:
 - Присвоение уникального порядкового номера всем записям и чтениям в соответствии с программным кодом на начальной стадии исполнения операции
 - Использование буфера записей
 - Использование буфера порядка записей
 - Буфер порядка записей по номерам операций следит, чтобы записи не обгоняли ни чтения, ни другие записи
- Порядок чтений относительно записей:
 - Чтения могут обгонять любые записи
 - Чтения просматривают буфер записей и могут спекулятивно использовать новые данные

Принятое в проекте решение

Положения оптимизированного режима

- Используются все аппаратные механизмы неоптимизированного режима
- Механизм разбиения операций на транзакции:
 - Программно-аппаратные контрольные точки
- Механизм восстановления к контрольной точке:
 - Записи завершаются только после завершения своей транзакции и только в соответствии с порядком x86
 - Все записи прерванной транзакции отменяются
- Механизм слежения за конфликтами очередности:
 - Использование буфера чтений (LDGT)
 - Объединение нескольких транзакций в группу
 - Все чтения одной группы завершаются логически одновременно
 - Спекулятивное использование результатов чтений
 - Техника перемещения чтений в более ранние группы

Аппаратная реализация принятых решений

Неоптимизированный и оптимизированный режимы:

- Упорядочивание записей относительно других записей и чтений
 - ▢ Разработка устройства ORDER с использованием старых решений из проекта «Эльбрус-S»
- Порядок чтений относительно записей
 - ▢ Использование буфера записей (STMB)

Оптимизированный режим:

- Обнаружение конфликтов очередности
 - ▢ Разработка буфера чтений (LDGT)
 - ▢ Использование устройства контроля спекулятивных чтений (DAM)
- Аппаратная поддержка транзакций
 - ▢ Сохранение регистров IP и BR в Control Unit (CU)
 - ▢ **Управление транзакциями** (CU и ORDER)
- Аппаратная поддержка восстановления системы к последней контрольной точке
 - ▢ **Управление восстановлением** (CU и ORDER)
 - ▢ Отмена записей в буфере записей (STMB)
 - ▢ **Управление отменой записей** (ORDER)

Схема устройства ORDER

SCALE

WCO

UCO

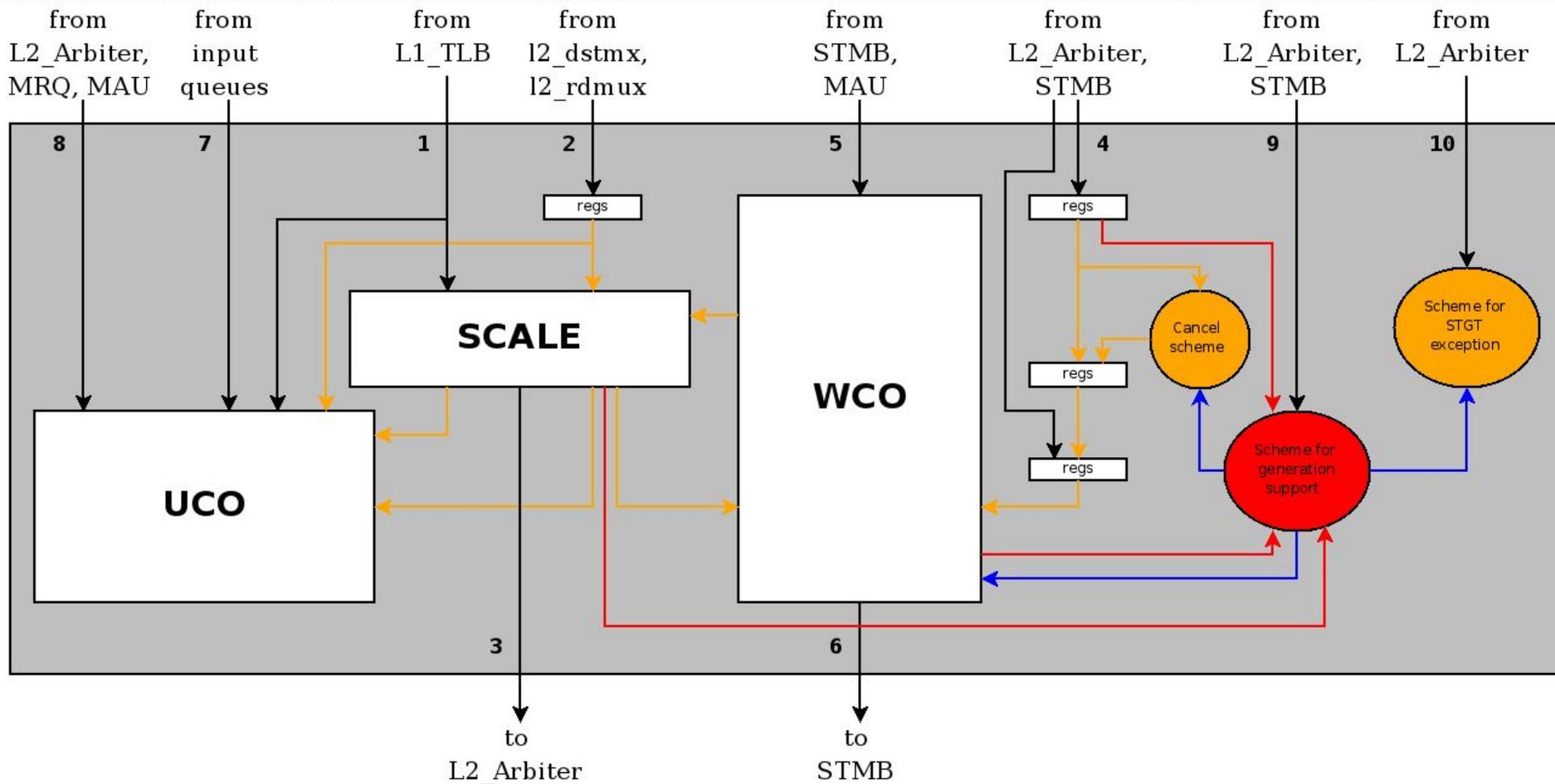
Cancel
scheme

Scheme for
generation
support

Scheme for
STGT
exception

- Модуль нумерации операций (Scale)
- Модуль упорядочивания и завершения записей (WCO)
- Модуль упорядочивания I/O обращений (UCO)
- Схема отмены записей прерванной транзакции
- Схема поддержки и переключения транзакций
- Схема выработки асинхронных прерываний

Схема устройства ORDER



Legend:

— external interfaces

— intermodule signals

— internal signals for generation support

— internal signals of generation support

Функции отдельных модулей устройства ORDER

- Модуль Scale
 - Выдача порядковых номеров всем обращениям в память в соответствии с программным кодом
 - Освобождение номеров по мере завершения операций
 - Вычисление номера самой «старой» операции
 - Нормирование шкалы для определения «возраста» операций
- Модуль UCO
 - Буферизация I/O обращений
 - Ожидание завершения всех предыдущих обращений в память до начала исполнения I/O операции
 - Блокирование всех следующих обращений в память до момента завершения I/O операции
- Модуль WCO
 - Занесение всех операций записи в буфер
 - Завершение операции записи, если завершены все предыдущие чтения и записи и завершено поколение операций записи
 - Управление отменой записей в буфере STMB

Функции отдельных схем устройства ORDER

- Схема отмены записей прерванной транзакции (Cancel scheme):
 - Управление отменой записей в арбитраже L2 кэша
- Схема поддержки и переключения транзакций (Scheme for generation support):
 - Обработка операций установки и сброса контрольной точки
 - Режим поколений (транзакций) и режим отката
 - Транзакционный конфликт – любое прерывание в режиме поколений
- Схема выработки асинхронных прерываний (Scheme for STGT exception):
 - Опасность зависания системы (deadlock)
 - Прерывание при нехватке ресурсов (переполнены Scale, WCO, UCO, STMB или другие буферы кэш-памяти)
 - Прерывание в случае некорректной программной ситуации (I/O обращение в режиме поколений)

Результаты работы

- Проведено исследование и анализ:
 - Используемых в современных мультипроцессорах различных моделей согласованности памяти
 - Основных принципов модели памяти Intel x86
 - Решений проблемы совместимости с Intel x86 в других архитектурах и предыдущих проектах «Эльбрус»
- С целью поддержки совместимости с Intel x86 в системе «Эльбрус-2S» разработана функциональная схема устройства ORDER
 - Упорядочивание записей относительно чтений и других записей
 - Аппаратная поддержка режима транзакций
 - Аппаратная поддержка восстановления системы к контрольной точке
- Разработано RTL-описание устройства ORDER
- Устройство ORDER прошло следующие этапы проектирования:
 - Отладка
 - Автономное тестирование
 - Системное тестирование
 - Первичный физический дизайн
- В настоящее время ожидаются результаты программных тестов



Спасибо за
внимание!