Федеральное государственное автономное образовательное учреждение высшего образования «Московский физико-технический институт (государственный университет)»

# Разработка механизма аппаратной предподкачки элементов таблицы страниц в буфер ассоциативной трансляции для процессоров «Эльбрус»

Выпускная квалификационная работа
(магистерская диссертация)

Студент: Фирюлин Дмитрий Иванович
Консультант: Знаменский Дмитрий Валерьевич
Научный руководитель: Груздов Фёдор Анатольевич

Москва, 2023

# Введение

В процессорах «Эльбрус» исполнение — в программном порядке (in-order execution).

- Операции чтения из памяти могут иметь большое или трудно прогнозируемое время исполнения; такие операции влекут за собой блокировку конвейера на потребителе.
- Среднее время доступа в память (AMAT – Average Memory Access Time) существенно снижается благодаря кэшированию.
- AMAT зависит от miss rate кэша, который связан с изначальным отсутствием данных в кэше.
- Чтобы снизить влияние данного фактора на AMAT, можно угадать расположение данных в памяти и подгрузить их заранее в кэш.

# Проблема

Проблема заключается в ощутимом влиянии механизма трансляции виртуальных адресов на время доступа в память при определённых сценариях работы.

- В современных процессорах аппаратно поддержан механизм виртуальной памяти, реализации содержат кэши (TLB - Translation Lookaside Buffer).
- В случае промаха в этих кэшах, необходимо выполнить поиск по таблице страниц (несколько запросов в память).

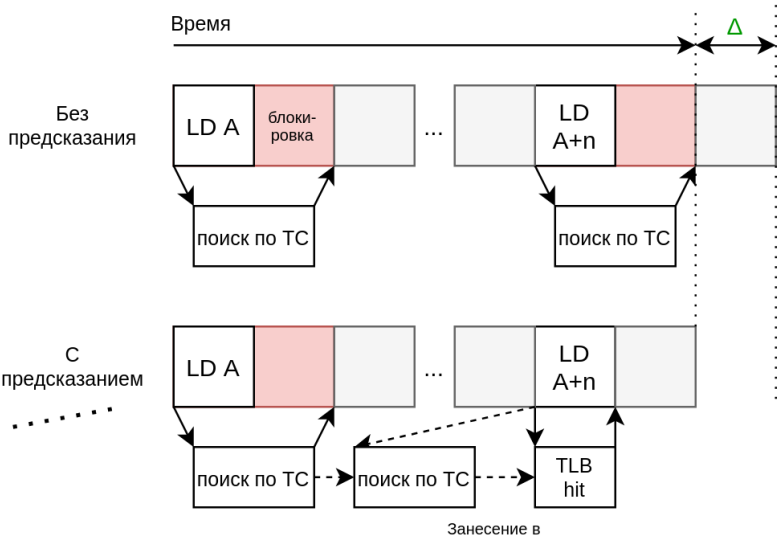
# Проблема (продолжение)

Пример: длинная последовательность обращений в память. Каждое обращение к новой странице запускает новую трансляцию виртуального адреса в случае промаха в TLB.

| Задача | Доля блокировок (обслуживание TLB miss) |
|---|---|
| 171.swim | 20% |
| Stream SET | 36% |
| PostreSQL (PGBench) | 13% |
| x11perf | 18% |

Задачи, работающие с большими непрерывными областями в памяти, являются целевыми.

# Предподкачка (prefetching)




Возможное решение: распознавать паттерны обращений к таблице страниц и выполнять поиск заранее.

# Цель работы

Разработка и внедрение аппаратного механизма предподкачки элементов таблицы страниц в TLB микропроцессора «Эльбрус».

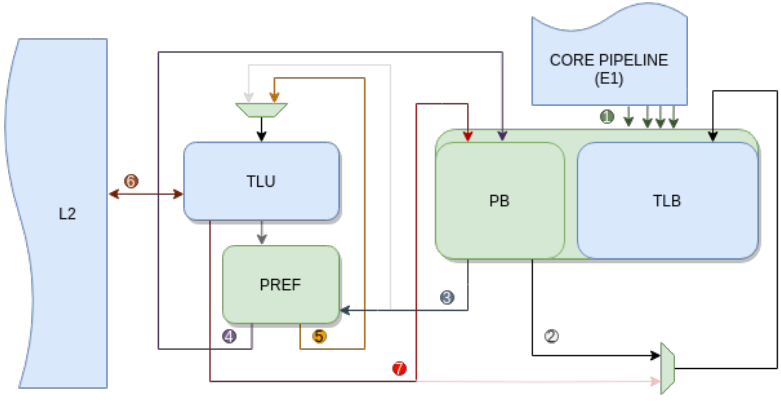
Задачи:

- Разработка RTL-описания и внедрение механизма предподкачки с базовым алгоритмом предсказания в существующую подсистему памяти.
- Реализация дополнительных алгоритмов предсказания в составе разработанной инфраструктуры префетчера TLB.
- Инженерная верификация разработанного решения при помощи RTL-симулятора и FPGA-прототипа.
- Исследование влияния дополнительных алгоритмов на производительность подсистемы памяти.

# Разработка RTL-описания механизма предподкачки

## Общая схема и алгоритм работы

Механизм встраивается в существующую подсистему памяти.
Работа механизма в случае попадания в PB.



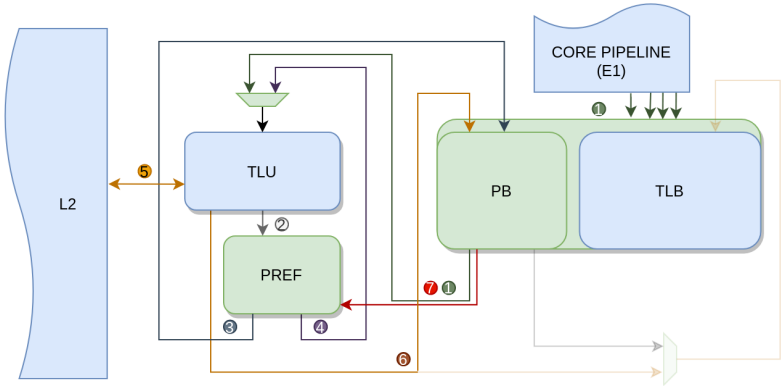

1. Чтение TLB & PB -- PB hit
2. Перенос записи из PB в TLB
3. Запрос PB на предподкачку
4. Проверка наличия записи в PB
5. Запуск поиска предсказанной записи
6. Поиск по таблице страниц
7. Запись результата поиска в PB

PB – буфер предподкачки, TLU – блок поиска по таблице страниц, L2 – кэш данных

# Разработка RTL-описания механизма предподкачки

## Общая схема и алгоритм работы

Работа механизма в случае промаха в PB.




1. Чтение TLB & PB -- miss
2. Запрос TLU на предподкачку
3. Проверка наличия записи в PB
4. Запуск поиска предсказанной записи
5. Поиск по таблице страниц
6. Запись результата поиска в PB
7. Отслеживание потока PB hit/miss

PB – буфер предподкачки, TLU – блок поиска по таблице страниц, L2 – кэш данных второго уровня.

# Разработка RTL-описания механизма предподкачки
## Интеграция с подсистемой памяти

Потребовалась существенная доработка TLU для поддержки новой спекулятивной операции – поиска PTE (Page Table Entry) по предсказанному адресу.
Основные требования:

- Минимальное количество побочных эффектов
- Прерывания, отсутствие access bit, запрос из конвейера ядра – останавливают предподкачку.
- Минимальная задержка при переключении между поиском по предсказанному адресу и обязательному поиску.

# Разработка RTL-описания механизма предподкачки

**Буфер предподкачки**
Полностью ассоциативный буфер, хранящий тег запроса и PTE.

- 6 портов чтения (4 канала + предпросмотр + очистка)
- 1 порт записи
- размер: 32 элемента

**Менеджер предподкачки**
Выполняет арбитраж доступа предсказателей к TLU.

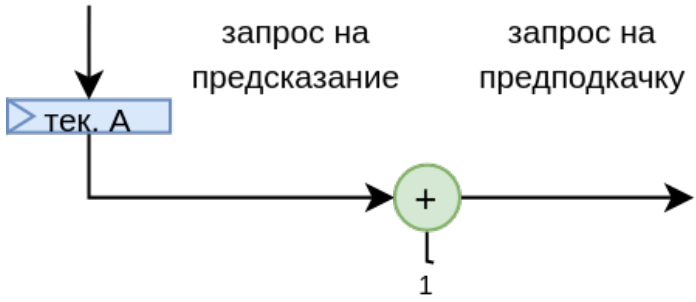
- одновременная работа до 4-х типов предсказателей
- предпросмотр предсказанного тега в PB
- остановка предсказателей при обязятельном поиске
- управление предсказателями

# Алгоритмы предсказания

## Последовательностная предвыборка, базовый алгоритм (stride)

В качестве базового алгоритма реализована предвыборка каждой последующей страницы (stride).

$$A_{\text{след}} = A_{\text{текущ}} + 1$$




A – адрес запроса.

# Алгоритмы предсказания

## Последовательностная предвыборка с историей и буфером потоков (seq_h)

Потоком называется последовательность запросов с инкрементирующимся адресом.

$$A_{\text{след}} = A_{\text{текущ}} + 1, A_{\text{текущ}} \in S$$

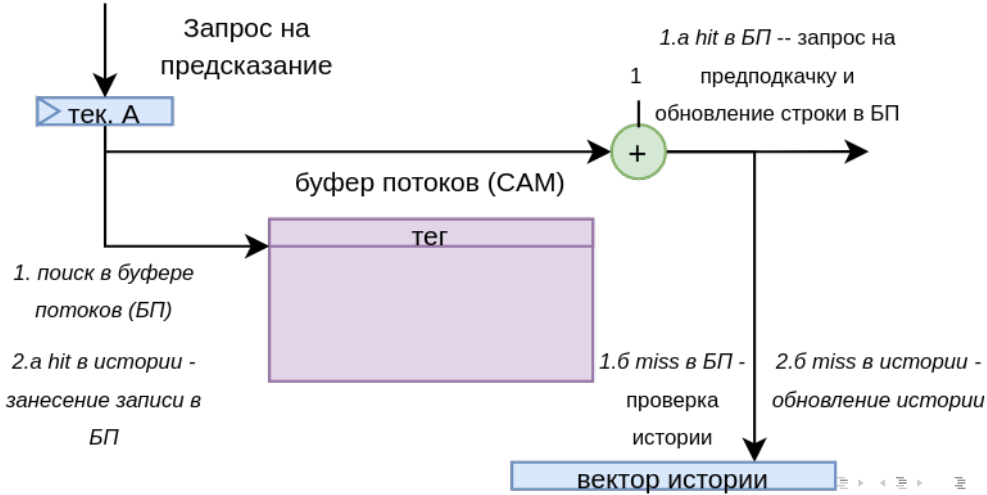
Где S – буфер потоков. Предвыборка следующей страницы выполняется, если запрос является частью потока адресов.

Пример последовательности (цветами обозначены запросы соответствующих потоков)

$$...10, 32, 56, 11, 33, 67, 42, 12, 85, 34...$$

# Алгоритмы предсказания

## Последовательностная предвыборка с историей и буфером потоков (seq_h)



# Алгоритмы предсказания
## Предвыборка с произвольным шагом (asp)

Запросы потока отличаются на фиксированный шаг (расстояние) и имеют одинаковый IP.

$$A_{\text{след}} = A_{\text{текущ}} + d, A_{\text{текущ}} \in S$$

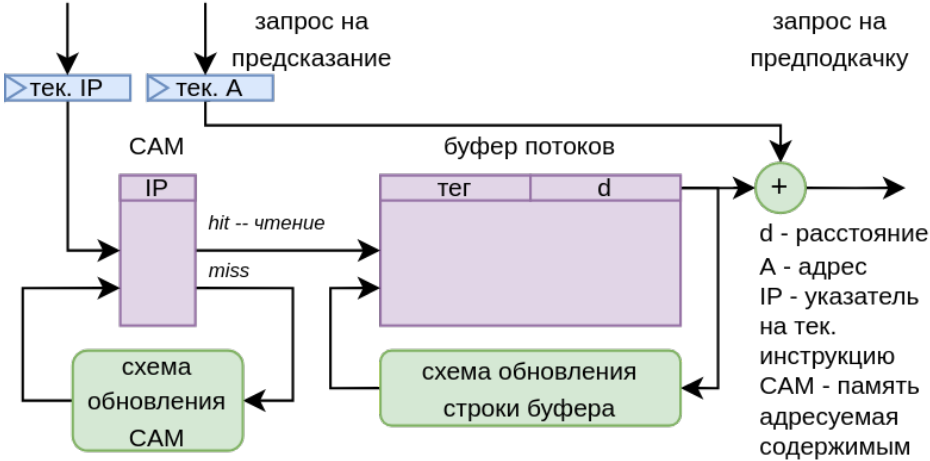
Где S – буфер потоков.
Шаг вычисляется между двумя последними запросами, привязанными к счётчику команд. Адрес предсказанного запроса формируется как сумма адреса текущего и шага.

Пример последовательности (цветами обозначены запросы соответствующих потоков)

$$...10, 11, 12, 20, 21, 22, 13, 14, 23, 24...$$

# Алгоритмы предсказания

## Предвыборка с произвольным шагом (asp)



# Инженерная верификация

Помимо RTL-описания было разработано тестовое окружение или ассемблерные тесты в зависимости от этапа.

- Реализованы автономное тестовое окружение для всего механизма предподкачки, генератор случайных последовательностей запросов с различными конфигурациями потоков (шаг, паттерн, количество, случайные компоненты)
- Создан набор ассемблерных тестов для проверки в составе ядра МП при помощи симулятора
- Проведена отладка на FPGA-прототипе – использованы тесты memperf и stream set

# Исследование влияния предсказателей на производительность подсистемы памяти (Синтетические тесты)

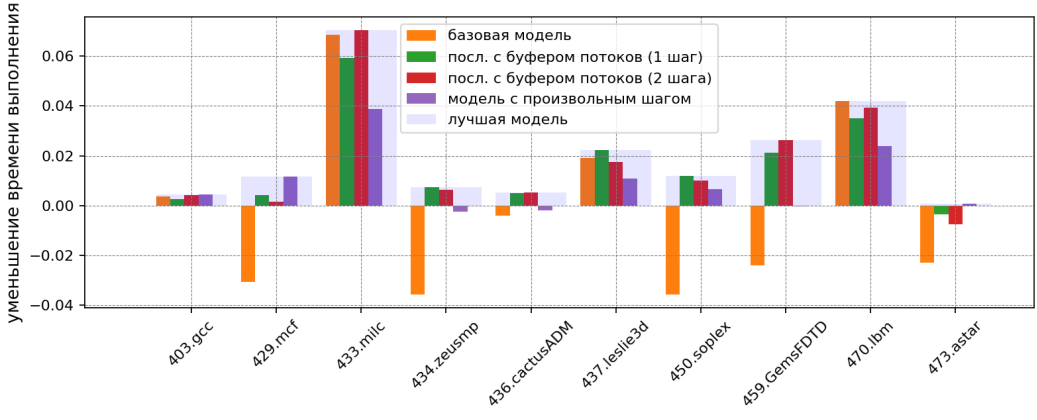
Задачи stream: последовательная работа с протяжённым участком памяти с наибольшей плотностью запросов в память.

| Задача stream | Ускорение |
|---|---|
| set | 9.9% |
| read | 16.0% |
| copy | 9.1% |
| add | 8.4% |

Измерения производились в режиме 4-х КиБ виртуальных страниц.

# Исследование влияния предсказателей на производительность подсистемы памяти (SPEC CPU 2006)

Измерения производились на задачах, для которых наблюдается существенная доля простоя конвейера из-за обслуживания промаха в TLB.

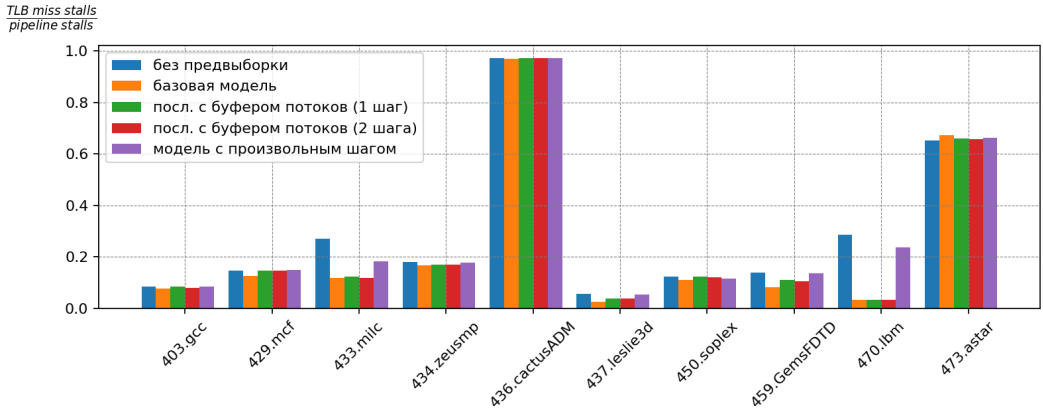


# Результаты работы

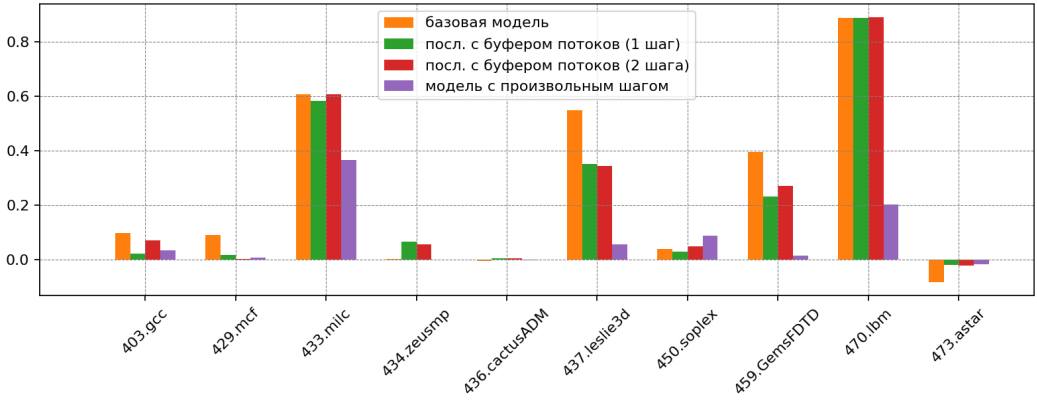
- Разработано и отлажено RTL-описание механизма предподкачки
- Произведено внедрение в существующую подсистему памяти МП «Эльбрус»
- Реализованы алгоритмы предсказателей
- Произведены измерения производительности на SPEC CPU 2006 для различных типов предсказателей.

Получено ускорение на тестах SPEC CPU от -0.5% до 7%.
На задачах типа stream наблюдается ускорение от 8.4% до 16.0%.

# Приложение 1: остановки конвейера из-за TLB miss для SPEC 2006

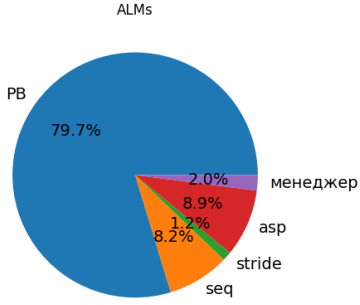


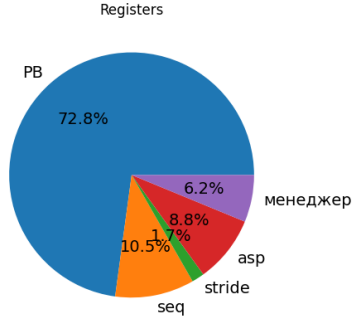
# Приложение 2: уменьшение времени простоя конвейера из-за TLB miss



# Приложение 3: использование аппаратуры

По результатам синтеза в среде Quartus.




| Устройство | ALMs | Registers |
|---|---|---|
| TLB | 135419 | 113677 |
| TLB prefetcher | 8789 | 3917 |