Московский физико-технический институт
(государственный университет)
Физтех-школа радиотехники и компьютерных технологий
Факультет радиотехники и кибернетики
Кафедра информатики и вычислительной техники

Магистерская диссертация

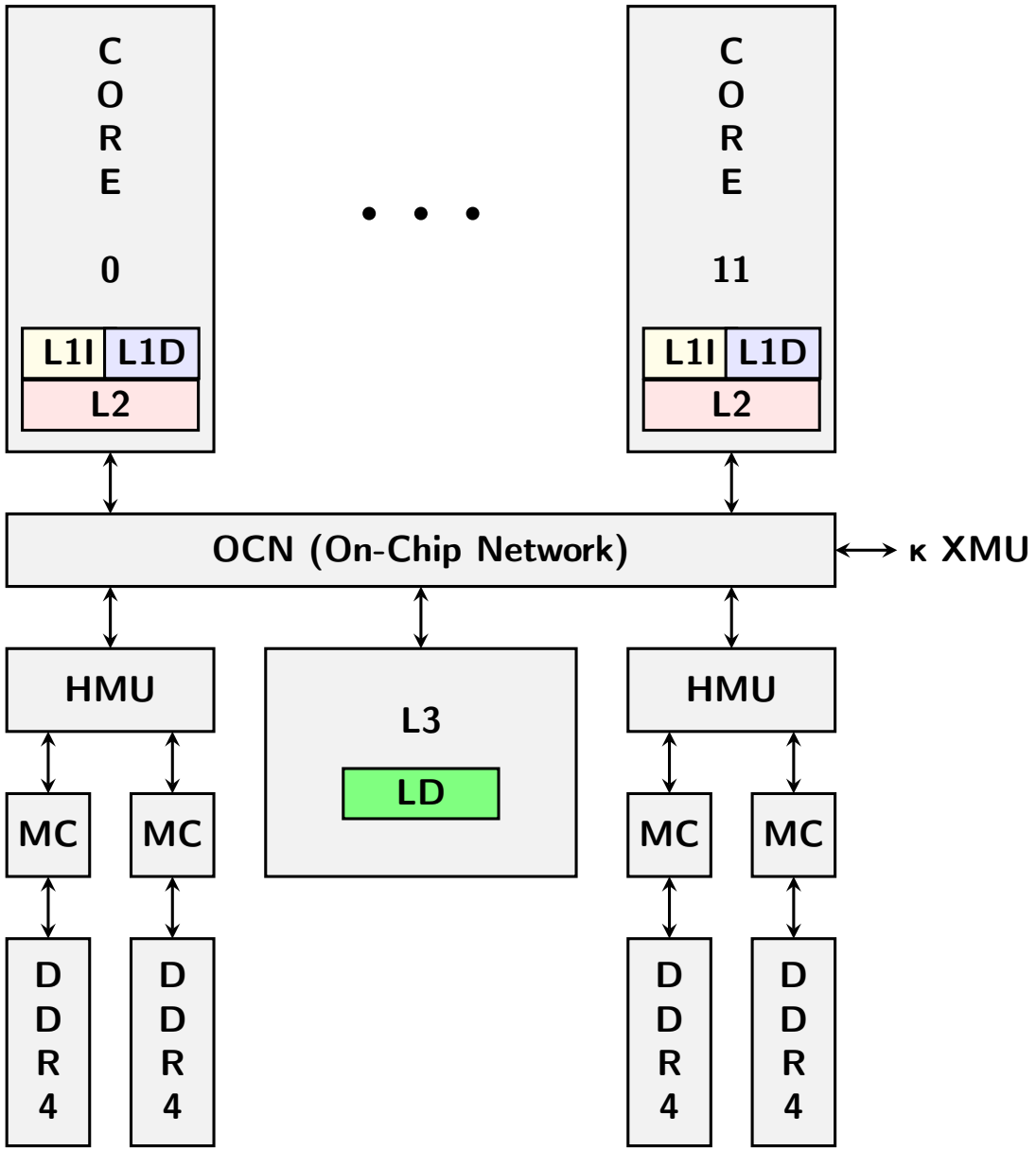
Разработка генератора тестов подсистемы памяти микропроцессора Эльбрус-12С

Агафонов Владимир Андреевич, гр. 213

Научный руководитель: к.т.н. **Мешков Алексей Николаевич**
Научный консультант: **Фролов Павел Викторович**

Москва
2018

Подсистема памяти МП Эльбрус-12С



CORE – вычислительно ядро архитектуры «Эльбрус» версии 6

HMU – устройство доступа к оперативной памяти

L1I – кэш-память инструкций 1-го уровня

L1D – кэш-память данных 1-го уровня

L2 – кэш-память 2-го уровня

L3 – кэш-память 3-го уровня

LD – локальный справочник

MC – контроллер памяти

XMU – устройство доступа к внешней памяти

Цель работы

- Разработать генератор псевдослучайных ассемблерных тестов для верификации подсистемы памяти микропроцессора Эльбрус-12С

Требования к генератору

- Генерация самопроверяющихся тестов
- Гибкая настройка параметров генерации

Требования к тестам

Верифицируемая функциональность

- Обращения в память различных типов
- Механизмы поддержки когерентности памяти
- Функционирование отдельных кэш-памятей
- Исполнение плотных широких команд

В тестах должны быть

- инструкции обращения в память с различными параметрами
- работа нескольких ядер с разделяемой памятью
- последовательности инструкций обращения в память по адресам, приводящим к вытеснению блоков из кэш-памятей
- широкие команды с несколькими инструкциями обращения в память

Генерируемые типы обращений

ld<формат> [<адрес>] <MAS>, <регистр назначения>
st<формат> <регистр-источник>, [<адрес>] <MAS>
stmqp <регистр-источник>, [<адрес>,<маска>] <MAS>

Форматы обращений: **b**, **h**, **w**, **d**, **q**, **qp**

MAS определяет

- функциональность операции,
- режим размещения в кэш-памятях разных уровней,
- тип памяти.

Функциональность операции	**Размещение в кэш-памятях**		
Простая операция	**L1**	**L2**	**L3**
Без трансляции			
Подкачка данных	✓		
Без проверки блокировок		✓	
Ввод-вывод			✓
Барьер типа acquire		✓	✓
Барьер типа release	✓	✓	✓

Типы памяти
GC (Когерентная кэшируемая)
GnC (Когерентная некэшируемая)
XC (Некогерентная)

Генерация инструкций обращения в память

Множество типов обращений:

$$T = I \times F \times M,$$

где I — множество опкодов $\{\texttt{ld},\texttt{st},\texttt{stm}\}$,
F — множество форматов $\{\texttt{b},\texttt{h},\texttt{w},\texttt{d},\texttt{q},\texttt{qp}\}$,
M — множество MAS.

Алгоритм генерации инструкции обращения в память

1. Выбор адреса обращения
2. Генерация типа обращения $type \in \widetilde{T}$,
   где $\widetilde{T}$ — множество типов обращений для данного адреса
3. Генерация маски записи (только для `stmqp`)
4. Выбор регистра-источника / регистра назначения

Карта памяти – структура, ставящая в соответствие каждому фрагменту памяти ядро-владелец и допустимые типы обращений.

Атрибуты фрагмента памяти

- Диапазон физических адресов – $\mathbf{A}^s = [A^s_{start}, A^s_{end}]$
- Множество доступных типов обращений – T
- Ядро-владелец – C
- Приоритет использования – P^u
  позволяет управлять частотой обращений в данный фрагмент
- Приоритет чтений – P^r
- Приоритет записей – P^w

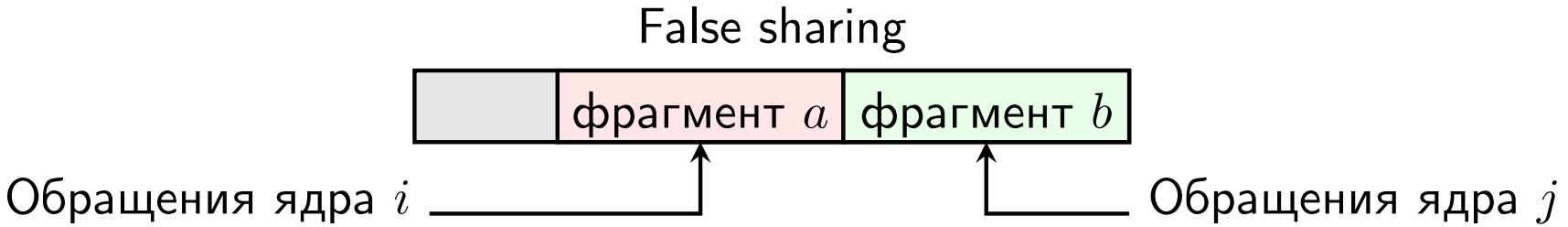
Работа нескольких ядер с разделяемой памятью

Работа с разделяемой памятью организована по типу **false sharing**:

кэш-строка разбивается на непересекающиеся фрагменты, с которыми работают разные ядра.

Достоинства подхода

+ Верификация когерентности
+ Не нужно следить за порядком выполнения обращений разных потоков



Вытеснения из кэш-памятей

Для формирования карты памяти, нацеленной на создание вытеснений в заданном уровне кэш-памяти, реализован генератор карты памяти.

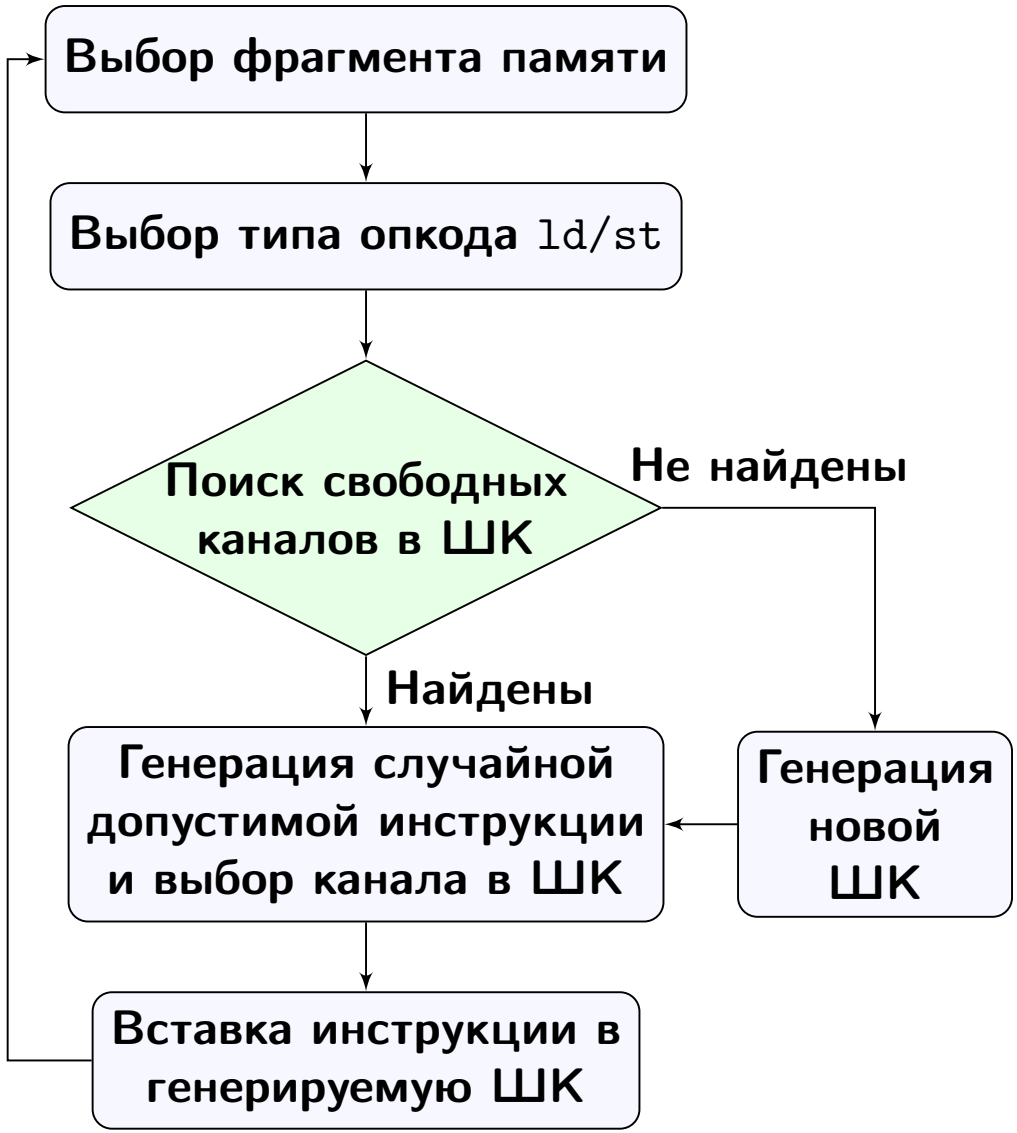
Входные параметры генератора

- описание адресации в целевой кэш-памяти
- минимальный размер выделяемой памяти
- атрибуты для сгенерированных фрагментов памяти

Алгоритм генерации карты памяти

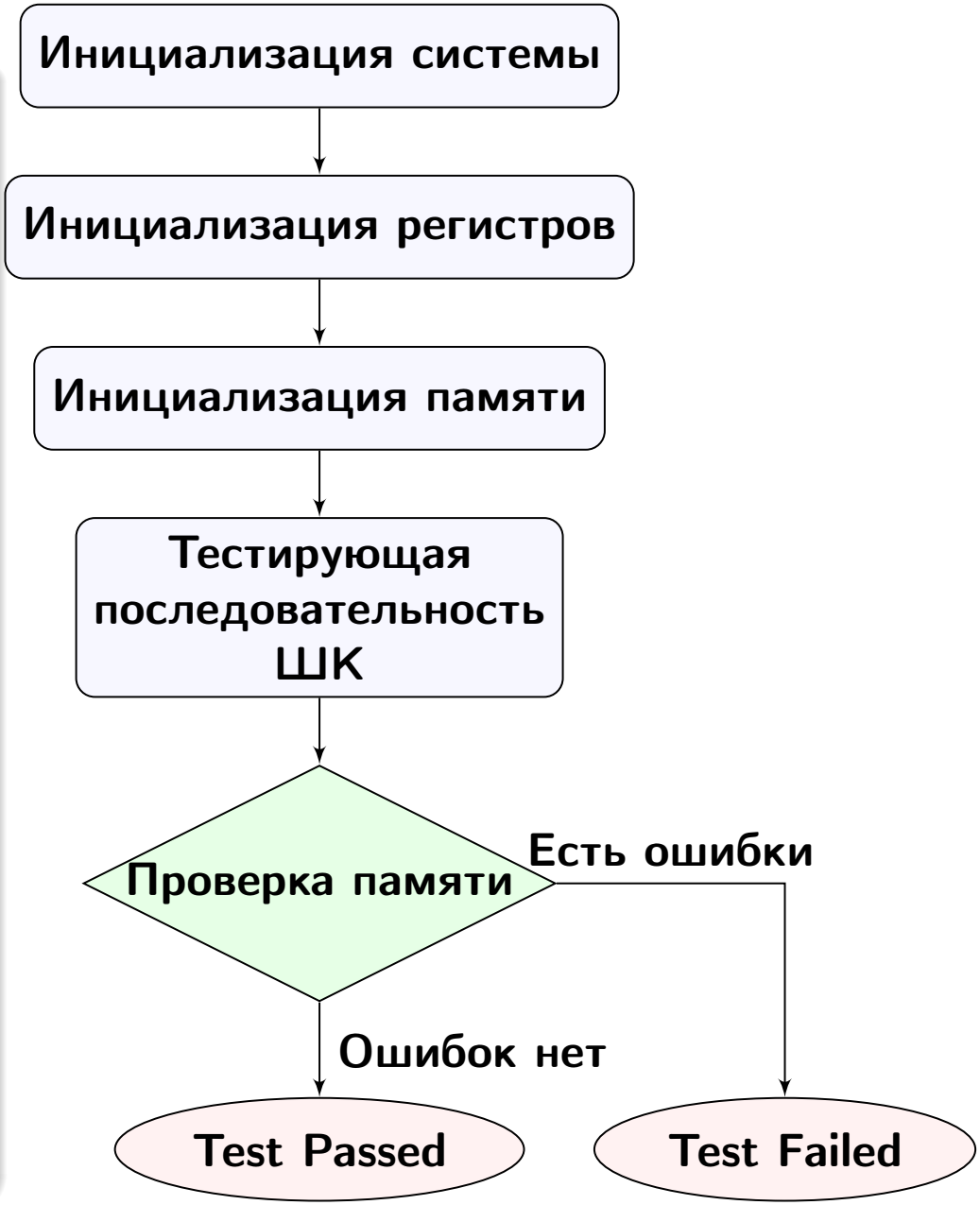
Генерация тестирующей последовательности ШК

- Генерация тестирующей последовательности широких команд (ШК) происходит независимо и одинаково для каждого ядра по карте памяти
- Выбор фрагмента памяти и типа опкода осуществляется случайно по заданному распределению
- Распределение фрагментов определяется их приоритетами P^u
- Распределение типов опкода определяется проритетами P^r, P^w



Алгоритм теста

- Тестирующая последовательность широких команд (ШК) реализует обращения, описанные в карте памяти
- Для обеспечения корректной работы тестирующей последовательности регистры и память инициализируются случайными значениями
- Критерием успешного завершения теста является соответствие данных в памяти по завершении теста эталонному значению



Генерация кода самопроверки
selfcheck-gen

- Для генерации кода самопроверки использован готовый инструмент — *selfcheck-gen*
- Данный инструмент обеспечивает генерацию кода сравнения значений регистров с эталонными значениями
- Источник эталонных значений – функциональная модель верифицируемого микропроцессора
- Для проведения проверки значений в памяти по окончании теста производится считывание фрагментов памяти в регистры

Параметры конфигурации

- Множество запускаемых ядер
- Количество инструкций в одной ШК
- Карта памяти

определяются конфигурационным файлом и строкой запуска

Разработан интерфейс конфигурирования карты памяти.

Интерфейс позволяет

- проводить ручную конфигурацию карты памяти;
- автоматизировать формирование фрагментов памяти, выбор ядер-владельцев и присваивание атрибутов обращений;
- генерировать карту памяти, нацеленную на создание вытеснений в заданной кэш-памяти.

Результаты работы

1. На языке C++ разработан генератор псевдослучайных ассемблерных тестов для верификации подсистемы памяти МП Эльбрус-12С, предоставляющий возможность гибкой настройки и генерирующий тесты с
   - кодом самопроверки;
   - различными инструкциями обращения в память;
   - работой нескольких ядер с разделяемой памятью;
   - последовательностями инструкций обращения в память по адресам, приводящим к вытеснению блоков из кэш-памятей;
   - широкими командами с несколькими инструкциями обращения в память.
2. Генератор тестов применён для верификации подсистемы памяти МП Эльбрус-12С. С помощью сгенерированных тестов обнаружены ошибки в L2, L3, MAU и TLB.

Автоматизация фрагментирования областей памяти

Атрибуты области памяти

- Диапазон физических адресов – $\mathbf{A}^s$
- Приоритет использования – P^u
- Множество допустимых размеров фрагментов кэш-строк – f
- Множество доступных типов обращений – T
- Множество ядер – S
- Приоритет чтений – P^r
- Приоритет записей – P^w

Алгоритм фрагментирования

1. Разделение каждой области памяти на кэш-строки
2. Корректировка MAS кэш-строки: работа со строкой должна осуществляться либо когерентными, либо некогерентыми обращениями
3. Разделение каждой кэш-строки на фрагменты допустимых размеров случайным образом
4. Формирование подмножества инструкций, не превышающих размер фрагмента памяти, для каждого фрагмента
5. Случайный выбор ядра $C \in S$ для каждого фрагмента