Санкт-Петербургский государственный университет
Кафедра системного программирования

Оптимизация алгоритма CRC–32 под RISC-V

Николай Алексеевич Пономарев, группа 21.Б10-мм

Научный руководитель: ст. преподаватель кафедры ИАС К. К. Смирнов

Санкт-Петербург
2022

Постановка задачи

Целью работы является реализация оптимизации CRC–32 с использованием аппаратных инструкций умножения многочленов на архитектуре RISC-V

Задачи:

- Изучить варианты оптимизации алгоритма CRC–32
- Выбрать целевую платформу для проведения измерений с учетом необходимых расширений процессора
- Адаптировать одну из существующих реализаций под RISC-V
- Выполнить замеры производительности оптимизированного кода

Алгоритм CRC–32 и существующие оптимизации

Алгоритм CRC (Cyclic Redundancy Check) используется для проверки целостности сообщения при передаче данных. В данной работе речь идет о его варианте — CRC-32. Известные способы оптимизации:

- Заранее вычисленные таблицы значений
- Использование аппаратных инструкций вычисления CRC-32
- Использование инструкций умножения многочленов над $\mathbb{F}_2$

Необходимые возможности процессора

- RISC-V обладает модульной архитектурой
- Инструкции для умножения многочленов над $\mathbb{F}_2$ содержатся в расширении B[1]
- Можем использовать только 64-битную базу

[1] Standard Extension for Bit Manipulation

Платформы для измерений

- Нет ни одного доступного процессора с расширением B
- Необходим симулятор, были найдены:
  - Spike
  - gem5
- Для измерений подходит только gem5

Реализация

- В качестве базовой реализации была выбрана реализация из ядра Linux
- Адаптированная реализация обрабатывает 128 бит за раз

Эксперимент

Эксперименты проводились на симуляторе gem5 со следующими характеристиками:

- MinorCPU с частотой 1 ГГц и размером кэша L1 в 64 Кб
- 512 Мб ОЗУ DDR4 с частотой 2400 МГц

Исходные файлы компилировались с флагами `-O3 -static`, а затем запускались на симуляторе.

Объем данных, байт	Стандартный алгоритм, тиков	Оптимизированный алгоритм, тиков
128	$308.5 \cdot 10^3$	$83 \cdot 10^3$
1024	$2277.5 \cdot 10^3$	$424 \cdot 10^3$
8192	$18 \cdot 10^6$	$3.6 \cdot 10^6$
65536	$191 \cdot 10^6$	$30.5 \cdot 10^6$

Таблица: Результаты измерений

Результаты

- Изучены существующие способы оптимизации алгоритма CRC–32
- Выбрана целевая платформа для проведения измерений с учетом необходимых расширений процессора
- Проведена адаптация реализации для x86 под RISC-V с использованием инструкции clmul
- Произведены измерения быстродействия оптимизированного кода на симуляторе gem5.