

Лабораторная работа №4

Расширение возможностей учебного процессорного ядра schoolRISCV

1 Описание задания

В лабораторной работе вам предлагается разобраться во внутреннем устройстве простейшего процессорного ядра архитектуры RISC-V. Результатом изучения микроархитектуры процессорного ядра и системы команд RISC-V станут ваши функциональные и нефункциональные модификации ядра.

Основное задание:

1. Расширить систему команд процессора двумя новыми командами, в соответствии с вашим вариантом;
2. Подготовить тестовое окружение системного уровня и убедиться в корректности вашей реализации путём запуска симуляционных тестов.

Примечание: При непосредственном описании ваших модификаций в коде проекта, запрещено использовать симуляционные конструкции и арифметические операции, отличные от сложения и вычитания (то есть, умножение, деление и возведение в степень реализуйте сами посредством описания любого, понравившегося вам, алгоритма). В тестовом окружении использовать симуляционные конструкции и всевозможные арифметические операции можно (и даже нужно).

2 Цели работы

- Ознакомиться с архитектурой RISC-V;
- Получить базовое понимание работы микропроцессорных ядер;
- Получить навыки работы с системами «средней» сложности;

3 Указания к выполнению работы

Процессорное ядро, с которым вам предлагается поработать, находится по следующему адресу: <https://github.com/zhelnio/schoolRISCV>. На текущий момент, актуальная конфигурация располагается в ветке **00_simple** на коммите **c84f949**. Первым делом вам необходимо ознакомиться с материалами, которые предоставил автор проекта. В частности вас будут интересовать пункты:

- Videos;
- Slides;
- New Instruction Example.

Проект удобен тем, что практически вся необходимая вам информация уже представлена автором в удобном для усвоения виде. Для успешного выполнения задания лабораторной работы вам необходимо подробно изучить:

- Теоретическую информацию о функционировании вычислительных процессоров общего назначения;
- Описание уже реализованных в рассматриваемом процессорном ядре команд;
- Структуру изучаемого проекта;
- Функциональные особенности каждой структурной единицы;
- Процедуру генерации «программы» и её загрузки в ROM.

3.1 Порядок выполнения работы

Процесс выполнения лабораторной работы делится на два этапа.

Первый этап — разработка микроархитектуры вашего решения. Перед тем, как приступить к написанию кода, вам необходимо представить микроархитектурную схему, включающую в себя ваши модификации. Проще всего будет дополнить существующую схему, предоставленную автором. Не вдавайтесь в детали блока вашей арифметической операции, но обязательно отобразите все внешние, по отношению к нему, контрольные сигналы и регистры. Ваша схема должна явно показывать, как изменится структура процессорного ядра, после расширения его функциональных возможностей.

Второй этап отводится непосредственно на разработку. Основная часть:

1. Для начала внесите модификации для поддержки команды из таблицы с вариантами (они проще);
2. Выполните тестирование и убедитесь в корректности работы вашей команды;
3. Скомпилируйте программу, с вашей инструкцией. Убедитесь, что она корректно выровнена и содержит корректные поля;
4. Затем измените структуру процессорного ядра таким образом, чтобы он мог работать с многотактовыми микрооперациями. На данном этапе свой блок можно заменить простой симуляционной заглушкой, вносящей небольшую задержку;
5. Выполните тестирование и убедитесь в том, что процессор остался работоспособным после ваших замечательных модификаций;
6. Затемстройте свой арифметический блок и проведите комплексное тестирование. Тестирование не должно быть полным — достаточно будет того, чтобы оно покрывало все исключительные ситуации и краевые точки ваших областей эквивалентности.

4 Варианты

Все варианты должны реализовать две команды. Первая команда должна выполнять функцию из второй лабораторной работы. Операнды для этой команды берутся из двух разных регистров, разрядность операндов не меняется (8 бит). Результат записывается в отдельный третий регистр; разрядность результата не превышает 32 бита. Вторая команда берется из таблицы, согласно вашему варианту.

Вариант	Команда №2	Пояснения
1	BLT	Стандартная команда из набора RV32I
2	BGE	Стандартная команда из набора RV32I
3	BLTU	Стандартная команда из набора RV32I
4	BGEU	Стандартная команда из набора RV32I
5	ANDI	Стандартная команда из набора RV32I. Реализовать в АЛУ.
6	XORI	Стандартная команда из набора RV32I. Реализовать в АЛУ.
7	SRLI	Стандартная команда из набора RV32I. Реализовать в АЛУ.
8	SRAI	Стандартная команда из набора RV32I. Реализовать в АЛУ.

5 Требование к отчету

1. Титульный лист;
2. Описание задания в соответствии с вашим вариантом;
3. Микроархитектурная схема модифицированного вами процессорного ядра, с указанием:
 - (a) Всех контрольных сигналов и логики их формирования;
 - (b) Всех новых регистров и соответствующих им сигналов разрешений;
 - (c) Всех новых входных и выходных сигналов;
 - (d) Блока конечного автомата и всех сопутствующих ему сигналов (если есть);
4. Описание микроархитектурной схемы и всех реализованных вами изменений;
5. Описание алгоритма функционирования микропроцессорного ядра в части исполнения ваших инструкций;
6. Рассчитанные значения количества тактов:
 - (a) Для самой быстрой инструкции;
 - (b) Для самой медленной инструкции;

7. Временные диаграммы, подтверждающие корректность рассчитанных вами значений из предыдущего пункта;
8. Результаты из отчёта по временным параметрам для вашей проекта (для этого необходимо выполнить синтез проекта);
9. Выводы по работе, в которых вам необходимо:
 - (a) Описать достоинства и недостатки ваших модификаций;
 - (b) Привести альтернативные решения для ваших модификаций (для дополнительного задания необязательно)
 - (c) Дать ответ на вопрос, чем ваше решение лучше альтернатив и почему для реализации было выбрано именно оно;
 - (d) Перечислить все возникшие в ходе выполнения работы вопросы.