

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. Модифицировать процессорное ядро, в соответствии с вашим вариантом;
2. Подготовить тестовое окружение системного уровня и убедиться в корректности вашей реализации путём запуска симуляционных тестов.

Дополнительное задание:

1. Выполнить основное задание;
2. Разработать интерфейсы для UART-приёмника и подключить его к проекту;
3. Реализовать возможность заполнения памяти команд данными с UART-приёмника;
4. Реализовать возможность вывода значений регистров на семисегментные индикаторы и новую команду для такого вывода;
5. Добиться корректной работы вашего проекта на FPGA Nexys4DDR.

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 Варианты

Варианты с буффером/очередью должны реализовать две команды.

Первая команда - **push xN** - должна загружать данные из младшей части регистра xN в буффер/очередь.

Вторая команда - **pop xN** - должна выгружать данные в младшую часть регистра xN.

Для очередей никаких дополнительных адресов для выгрузки данных не требуется, однако для буфферов LRU/MRU адрес выгружаемой ячейки должен храниться в

младшей части регистра **xN**. По результату операции **ror xN**, применительно к буферам LRU/MRU, младшая часть регистра **xN** должна быть перезаписана с значения адреса на данные из буфера, соответствующие этому адресу.

Варианты с арбитром должны реализовать одноктактовую выборку двух инструкций и их арбитраж в остальную часть процессорного ядра. Вместо того, чтобы выбирать и исполнять одну инструкцию - процессорное ядро должно выбирать две инструкции одновременно и управлять их доступом к исполняемому ресурсу ядра.

3 Цели работы

- Ознакомиться с архитектурой RISC-V;
- Получить базовое понимание работы микропроцессорных ядер;
- Получить навыки работы с системами «средней» сложности;
- *Получить навыки FPGA-прототипирования проектов «средней» сложности.

4 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 **472b4a2**. 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.

4.1 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, предоставленную автором, схему. Не вдавайтесь в детали вашего блока, будь то буфер, очередь или блок выборки, 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 для вариантов с буферами и очередями:

1. Для начала внесите модификации в контрольную логику процессорного ядра для поддержки команд `push` и `pop`. На данном этапе встраиваемый блок очереди/буфера можно заменить простой симуляционной заглушкой, вносящей небольшую задержку;
2. Выполните тестирование и убедитесь в корректности работы контрольной логики, задействованной в исполнении новых команд;
3. Скомпилируйте программу, с вашими инструкциями. Убедитесь, что она корректно выровнена и содержит корректные поля;
4. Если ваш блок буфера/очереди многотактовым, то измените структуру процессорного ядра таким образом, чтобы он мог работать с многотактовыми операциями;
5. Выполните тестирование и убедитесь в том, что процессор остался работоспособным после ваших замечательных модификаций;
6. Затемстройте свой блок и проведите комплексное системное тестирование. Тестирование не должно быть полным — достаточно будет того, чтобы оно покрывало все исключительные ситуации и краевые точки ваших областей эквивалентности.

Основная часть для вариантов с арбитром:

1. Внесите модификации в процессорное ядро, позволяющие ему выбирать две инструкции из памяти команд за один такт. Так как процессорное ядро многотактовое, то **сформируйте симуляционную задержку** между подачей первой и второй инструкции в остальную часть ядра, таким образом, чтобы команды не терялись;
2. Скомпилируйте программу, с арифметическими инструкциями. Убедитесь, что она корректно выровнена и содержит корректные поля. Проведите тестирование с использованием этой программы. По результатам тестирования вы должны быть уверены в корректности внесенных модификаций;

3. В рамках следующего этапа рекомендуется ввести логику арбитража двух выбранных инструкций, **заменив искусственно-введенную симуляционную задержку**. Встройте свой модуль из предыдущей лабораторной работы, сократив количество потоков до двух. Проведите тестирование.
4. Разберитесь с тем, какие конфликты могут возникать при арбитраже двух команд и, если нужно, внесите в модуль арбитра логику, разрешающую такие конфликты;
5. Проведите комплексное системное тестирование модифицированного процессорного ядра. Тестирование не должно быть полным — достаточно будет того, чтобы оно покрывало все исключительные ситуации и краевые точки ваших областей эквивалентности

Дополнительная часть:

Выполнение дополнительного задания рекомендуется начать с реализации возможности вывода данных регистров на семисегментные индикаторы. Это позволит выполнять удобную отладку проекта на стадии работы с UART. Необходимо добавить в процессорное ядро команду, которая будет иметь как минимум один аргумент — регистр, данные из которого необходимо вывести на семисегментные индикаторы и отображать либо до прихода сигнала сброса, либо до исполнения следующей такой команды.

Особенности выполнения второй части задания заключаются в том, что необходимо изменить ROM таким образом, чтобы разрешить не только возможность чтения, но и записи. Помимо этого, процесс прототипирования необходимо разделить на две части:

1. Запрет выполнения команд для процессорного ядра, но разрешение загрузки команд в память команд;
2. Запрет загрузки команд в память и разрешение выполнения команд для процессорного ядра.

Таким образом, необходимо предусмотреть соответствующую этим режимам работы логику.

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. Результаты из отчёта по временным параметрам для вашей проекта (для этого необходимо выполнить синтез проекта);
 7. 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.