# Список задач по проекту разработки CPU 2015

Необходимо разработать RISC процессор собственной архитектуры подобной архитектуре MIPS.

Предлагается реализовать разработку для платы Atlys с ПЛИС Spartan6 на борту. Данная плата содержит оперативную память DDR, что позволяет организовать взаимодействие процессора с памятью для исполнения пользовательских программ. Загрузка инструкций и данных в память предполагается от ПК через интерфейс USB-UART. Соответствующие аппаратные блоки уже имеются на плате.

Конечным результатом работы должно быть наблюдение исполнения пользовательских программ на спроектированной системе.

## Разработка ассемблера

Необходимо разработать программу, которая будет переводить ассемблерный код в машинный. Кроме того, в процессе разработки ассемблера необходимо проработать механизмы организации данных в памяти. Для примера можно привести два варианта:

1. Данные программы загружаются в оперативную память через инструкции процессора. Если имеется только такая организация, то загрузка данных занимает долгое время, однако это существенно упрощает ассемблер, так как не нужно вводить дополнительные конструкции;
2. Данные программы могут загружаться в оперативную память напрямую через USB-UART интерфейс. В ассемблерном коде устанавливается какие данные и по каким адресам должны быть сохранены в память напрямую, не через инструкции процессора. Такая организация значительно упрощает вид программ, так как не нужно описывать процедуры загрузки данных, но усложняет сам ассемблер.

Ассемблер должен быть разработан на языке высокого уровня (Python, C++).

Результатом работы является: программное обеспечение, которое обрабатывает ассемблерную программу и создаёт файл с машинным кодом, который в дальнейшем будет передан на плату. Кроме того необходимо создать документацию использования данного ассемблера: как писать программы, какие конструкции можно использовать, как загружать данные напрямую в память и т.п.

## Разработка модели CPU

Необходимо разработать функциональную модель CPU. Данная задача подразумевает использование языков высокого уровня C/С++ и взаимодействие с тестовым окружением, написанном на языке SystemVerilog.

Модель должна отражать внутреннюю структуру процессора и периферийных устройств: регистры, память. Кроме того она должна быть способна исполнять все инструкции из файла с машинным кодом.

## Взаимодействие платы Atlys с ПК

Целью данной задачи является получение готового программно-аппаратного решения передачи данных от ПК к плате Atlys. Предлагается использовать USB-UART мост так как на стороне ПК существуют готовые драйверы и программы, способные передавать данные на плату, а со стороны платы протокол UART намного проще, чем того же Ethernet.

## Разработка основных блоков CPU

Необходимо спроектировать и описать на Verilog HDL основные блоки процессора: регистровый файл, АЛУ, блок управления и т.п. и соединить их в единую систему.

Микроархитектура должна соответствовать микроархитектуре процессора из книги Digital design and computer architecture (Harris & Harris). Предлагается использовать 5-тактовое неконвейризованное ядро: одна инструкция исполняется за несколько тактов.

Кроме описания модулей необходимо простейшим способом проверить их функционал.

## Работа с памятью

Необходимо подключить оперативную память на плате к нашей разработке. Исследовать её возможности и предложить варианты работы с ней: как со стороны процессора так и со стороны внешнего интерфейса.

Кроме того требуется организовать доступ процессора в память: стадия доступа за данными в память и стадия извлечения инструкции. Нужно учесть следующие моменты:

1. При невозможности получить инструкцию за такт, организовать преподкачку инструкций во время исполнения предыдущий;
2. Возможность инициирования передача данных от памяти к внешнему интерфейсу (совместно с пунктом взаимодействия платы с ПК)

Данная задача связана с задачи разработки процессора и передачей данных от платы к ПК.

## Разработка тестового окружения

Необходимо разработать тестовое окружение на языке SystemVerilog и проверить корректность работы разрабатываемого устройства. Верификация должна проводится с использованием модели и результатов работы ассемблера (файла с машинным кодом).

# План работ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Время | Задача | Исполнитель | Примечание | |
| **1** | **11.03 – 29.04** | **Разработка ассемблера** | Балыкина Наталья | |  |
| 1.1 |  | Разработка ассемблера |  |  | |
| 1.2 |  | Создание документации использования ассемблера |  |  | |
| **2** | **11.03 – 15.04** | **Разработка модели процессора** | Макарова Светлана | |  |
|  |  |  |  | |  |
|  |  |  |  |  | |
|  |  |  |  |  | |
| **3** | **11.03 – 15.04** | **Разработка основных блоков процессора** | Калтахчян Даниэль | |  |
| 3.1 | 11.03 – 18.03\*  28.03 – 1.04 | Разработка регистрового файла, блока декодирования и АЛУ |  | Появились файлы АЛУ и регистрвого файла | |
| 3.2 | 18.03 – 1.04 | Разработка блока управления |  |  | |
| 3.3 | 1.04 – 8.04 | Соединение блоков вместе |  |  | |
| 3.4 | 11.03 – 8.04 | Предоставление блок-схемы ядра с описанием основных блоков |  |  | |
|  |  |  |  |  | |
| **4** | **11.03 – 29.04** | **Взаимодействие платы Atlys с ПК** | Глазков Валерий | |  |
| 4.1 |  | Изучение описания платы Atlys, изучение USB-UART моста. Анализ необходимых hardware блоков, изучение существующего софта. |  |  | |
| 4.2 |  | Передача файла от ПК к плате и обратно |  | Необходимо разработать модуль “loop back”, который будет возвращать к ПК каждый принятый байт. Тем самым можно убедится, что все данные дошли. | |
| 4.3 |  | Разработка или использование существующего асинхронного FIFO |  | Разные частоты блоков, необходима синхронизация | |
| 4.3 |  | Создание управляющей логики, для запросов к памяти на чтение данных |  |  | |
| 4.4 |  | Обёртка интерфейсом Wishbone для взаимодействия с памятью |  |  | |
|  |  |  |  |  | |
| **5** |  | **Работа с памятью** | Гордеев Владимир | |  |
| 5.1 |  | Разработка ROM |  | Программа начальной загрузки и реализация памяти на verilog | |
| 5.2 |  | Проработка архитектуры контролёра памяти |  | Документация, блок схема, основные блоки, карта памяти | |
| 5.3 |  | Реализация ip-блока контроллера доступа к DDR2 |  | Xilinx IP-core | |
| 5.4 |  | Разработка блоков определение «окон» памяти, маршрутизация запросов |  |  | |
|  |  |  |  |  | |
| **6** | **11.03 – 29.04** | **Функциональная верификация** | Болотников Александр | |  |
| 6.1 |  | Установка программы моделирования |  |  | |
| 6.2 |  | Создание плана верификации |  | Какую сборку верифицировать? Как верифицировать? Что необходимо покрыть? | |
| 6.3 |  | Разработка основных компонент тестового окружения |  |  | |