Федеральное государственное автономное образовательное учреждение  
высшего образования   
"Национальный исследовательский университет   
"Высшая школа экономики"

Московский институт электроники и математики им. А. Н. Тихонова

Департамент компьютерной инженерии

Отчёт

о выполнении практической работы № 3

Тема работы: «Разработка и программирование Soft-процессорных ядер с архитектурой однотактный MIPS. Часть 3»

по курсу «Системное проектирование цифровых устройств»

Выполнила:

Бригада № 10

Власов Р. В. БИВ174

Сегида Т. О. БИВ174

Принял

ассистент Американов А. А.

Оценка:

Москва 2020 г.

Оглавление

[1. Добавление памяти 3](#_Toc52227206)

[1.1. Добавление IP-блока памяти 3](#_Toc52227207)

[1.2. Добавление памяти средствами Verilog 3](#_Toc52227208)

[2. Самостоятельная работа 3](#_Toc52227209)

[2.1. Задание № 1 3](#_Toc52227210)

[2.1.1. Добавить блок памяти 3](#_Toc52227211)

[2.1.2. Добавить команды LW и SW в тракт передачи данных 3](#_Toc52227212)

[2.2. Задание № 2 5](#_Toc52227213)

[2.2.1. Разработать программу на ассемблере, которая инвертирует все нечетные числа в памяти 5](#_Toc52227214)

[2.3. Задание № 3 5](#_Toc52227215)

[3. Выводы 6](#_Toc52227216)

[4. Список литературы 6](#_Toc52227217)

1. Добавление памяти
   1. Добавление IP-блока памяти

Результаты генерации находятся в lab\_3\schoolMIPS\board\de10\_standard\generated\.

* 1. Добавление памяти средствами Verilog

Результаты находятся в lab\_3\schoolMIPS\board\de10\_standard\written\.

1. Самостоятельная работа
   1. Задание № 1
2. Добавить в тракт передачи данных блок памяти на 32 (переключателей 5, почему бы не сделать тогда на 32) 32-хбитных слов с двумя входами/выходами (1-ый вход-выход используется для работы процессора, адресация в памяти начинается с 0 адреса, 2-ой вход-выход используется для подключения внешнего управления (адресный вход получает значение с переключателей, меняя значения которых можно просматривать содержимое памяти, выход данных не подключен, вход разрешения записи не подключен))
3. Добавить команды LW и SW в тракт передачи данных (память инициализируется hex-файлом, содержащим 32 32-хбитных слова)

Данное задание выполнялось на версии процессора 00\_simple.

* + 1. Добавить блок памяти

На 32 32-хбитных слова с двумя входами/выходами. Используется модуль ram\_2\_port, сгенерированный в предыдущей части лабораторной работы

Для дальнейшего выполнения этого пункта задания целесообразно перейти к следующему пункту.

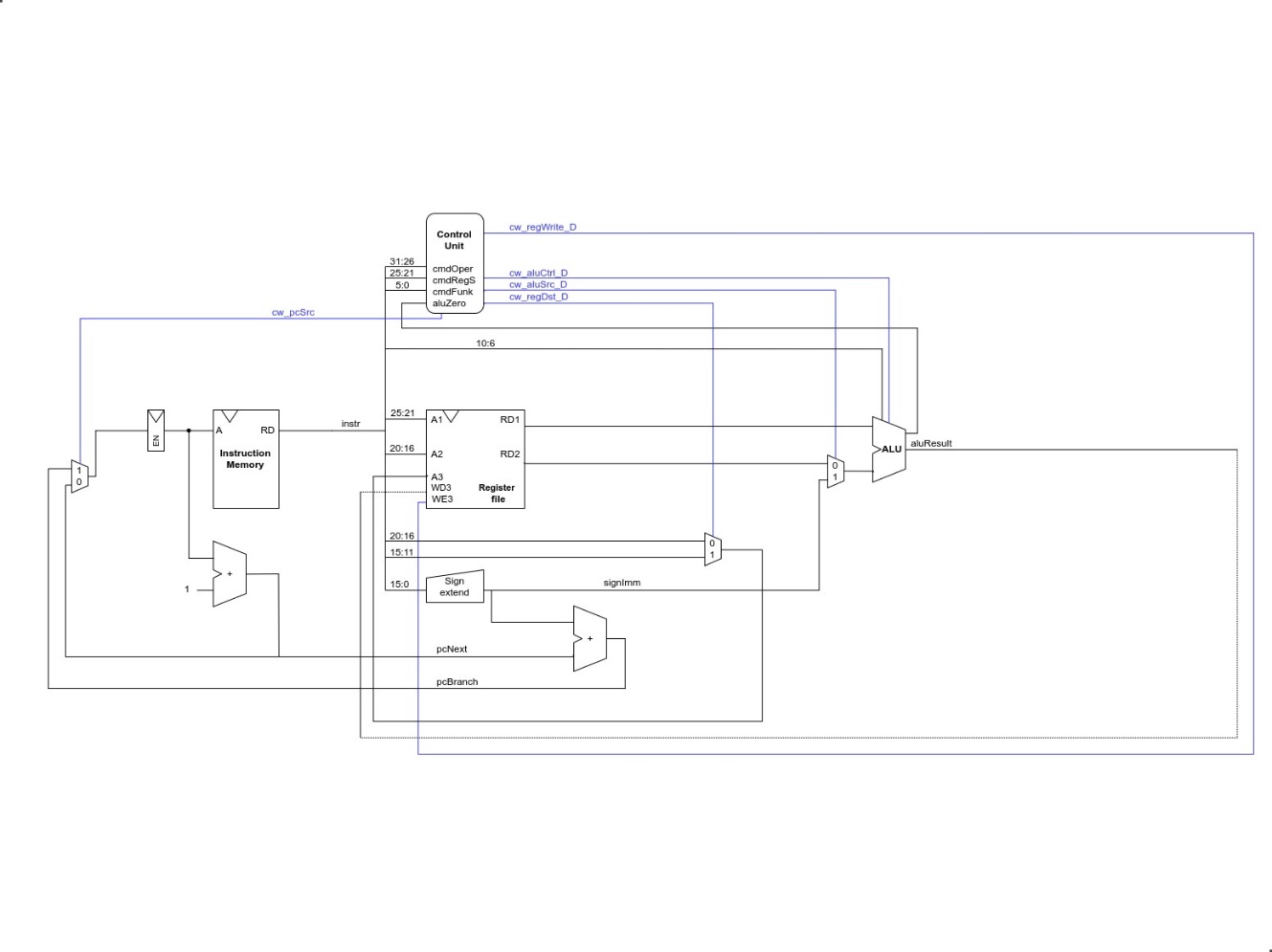
* + 1. Добавить команды LW и SW в тракт передачи данных

Память инициализируется hex-файлом, содержащим 32 32-хбитных слов. Данный файл находится в lab\_3\schoolMIPS\program\03\_task\_2\_1\_1\.

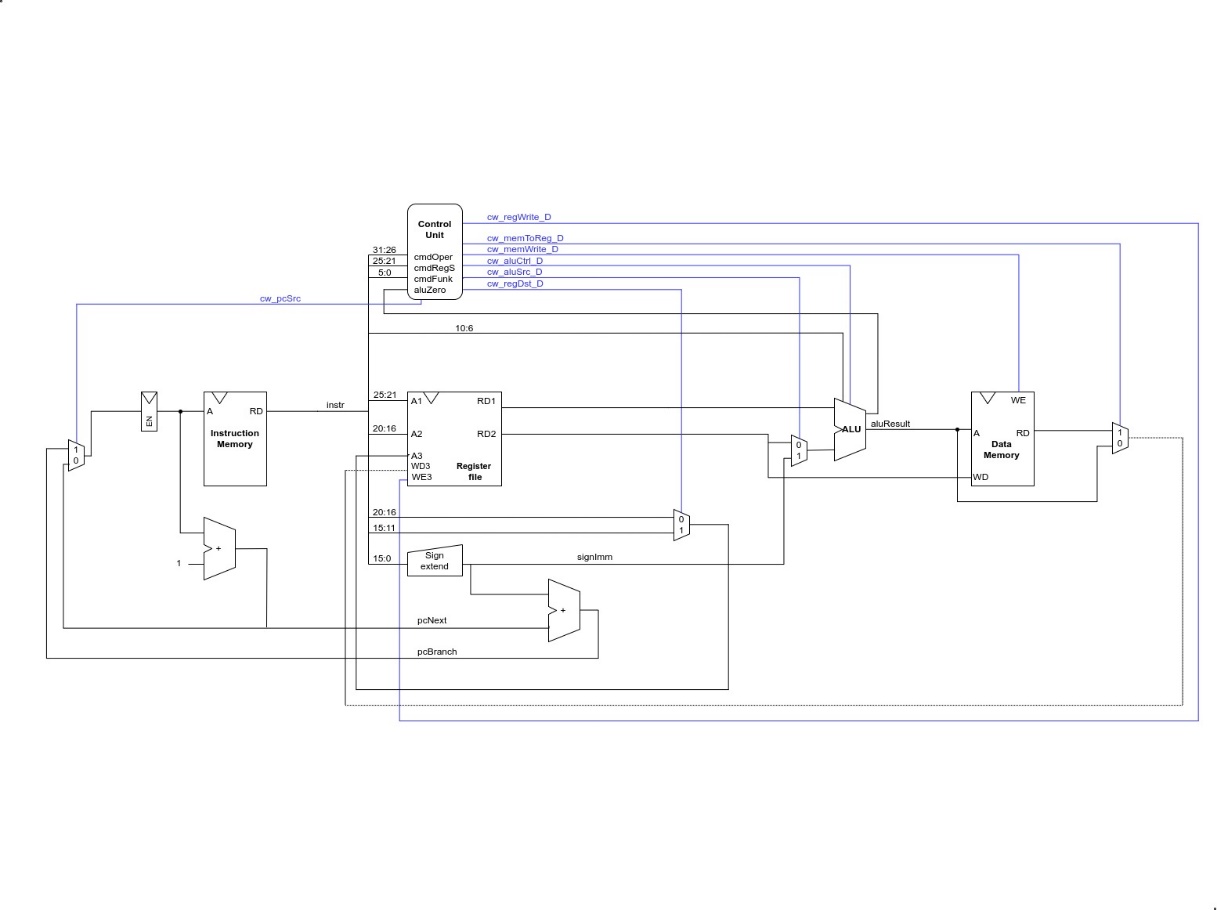
1-ый вход-выход используется для работы процессора, адресация в памяти начинается с 0 адреса. Сделано также, как в книге H&H в главе 7.3.1, чтобы память была как-то задействована для работы процессора.

Чтобы упростить задачу, сравним версии процессора 00\_simple и 01\_mmio. Заметим, что вторая версия архитектурно отличается только наличием необходимой нам памяти (и реализаций, в следствие, команд lw и sw). Как видно на рисунках ниже, все, что нужно для добавления памяти в процессор это:

1. добавить саму память данных в разрыв между регистровым файлом и АЛУ(ram\_2\_port)
2. добавить мультиплексор перед входом WD3, который выбирает между выходом RD памяти и шиной AluResult
3. протянуть из порта RD2 регистрового файла шину в память данных в порт WD
4. добавить управляющий сигнал WE памяти данных (memWrite), разрешающий запись в память данных
5. добавить управляющий сигнал мультиплексору (из пункта 2 этого списка), выбирающий между памятью данных и результатом АЛУ



1. Версия процессора 00\_simple.



1. Версия процессора 01\_mmio.

address\_a – WD

data\_a – A

wren\_a – WE

q\_a – RD

добавили в sm\_cpu.vh (взято из 01\_mmio):

`define C\_LW 6'b100011 // I-type, Load Word

// Rt = memory[Rs + Immed]

`define C\_SW 6'b101011 // I-type, Store Word

// memory[Rs + Immed] = Rt

2-ой вход-выход блока памяти используется для подключения внешнего управления (адресный вход получает значение с переключателей, меняя значения которых можно просматривать содержимое памяти, выход данных не подключен, вход разрешения записи не подключен)

Чтобы не усложнять себе жизнь, приняли такое решение – переключатели, встроенные в плату, отвечавшие за выбор регистра, выводимого на семисегментные дисплеи, будут выбирать какое слово читается из памяти и выводится на семисегментный дисплей. Соответственно, на семисегментные дисплеи выводятся теперь данные из памяти данных, а не из регистров.

* 1. Задание № 2

1. Разработать программу на ассемблере, которая инвертирует все нечетные числа в памяти (5 вариант)
2. Провести прототипирование процессора и выполняемой программы
3. Сравнить данный вариант процессора с версией 01\_mmio (данный вариант – тот, что разработали мы)

Чтение и память данных осуществляется, начиная с нулевого адреса (их всего 32).

* + 1. Разработать программу на ассемблере, которая инвертирует все нечетные числа в памяти

Код программы находится в lab\_3\schoolMIPS\program\04\_task\_2\_2\_1\main.S.

Код программы для запуска в Mars находится в lab\_3\schoolMIPS\program\04\_task\_2\_2\_1\main\_for\_mars.S.



1. Память для теста программы в Mars.



1. Память после работы программы в Mars.

Как видно, программа инвертирует только нечетные значения в памяти.

* + 1. Провести прототипирование процессора и выполняемой программы

Данный пункт выполним только в лаборатории.

* + 1. Сравнить данный вариант процессора с версией 01\_mmio

Дополненная нами версия отличается лишь тем, что память данных является двухпортовой, и второй порт используется для внешнего чтения из памяти. Извне (с переключателей на плате (4..0) приходит адрес памяти для чтения, и на семисегментный дисплей выводится прочитанное из памяти слово.

* 1. Задание № 3

1. Перейти в ветку проекта 03\_pipeline
2. Выполнить программы 00\_counter, 01\_fibonacci, 02\_sqrt, 03\_ram и убедиться, что они работают также, как в предыдущей лабораторной работе
3. Выполнить программы 06\_hz\_forward, 07\_hz\_stall, 05\_hz\_branc и добавить комментарии к программам
4. Объяснить, чем эта версия процессора отличается от базовой
5. Выводы

Во время выполнения данной работы мы познакомились с процессором schoolMIPS MIPS архитектуры. Сравнили две версии данного процессора. Получили навыки расширения функциональности процессора добавлением команд. Получили навыки работы с платой DE10-Standard. Вспомнили процессы дизассемблирования и моделирования.

1. Список литературы
2. SPDS\_Lab\_2\_2020 [Электронный ресурс]. – URL: [https://docs.google.com/document/d/1WuYORvrBLINq-EuFRP1P-HyPyK4WiaPP\_O5PQ25Yehc/edit#heading=h.7wjwmyn0vwth](https://docs.google.com/document/d/1WuYORvrBLINq-EuFRP1P-HyPyK4WiaPP_O5PQ25Yehc/edit%23heading=h.7wjwmyn0vwth)
3. DE10-Standard User Manual [Текст] Terasic inc – 133c.
4. MIPSfpga/schoolMIPS Wiki [Электронный ресурс]. – URL: <https://github.com/MIPSfpga/schoolMIPS/wiki>
5. Цифровая схемотехника и архитектура компьютера [Текст]/ Дэвид М. Хэррис и Сара Л. Хэррис – 1627с.
6. Цифровой синтез [Текст]: практический курс / Панчул Ю., Романов А. / Романов А. – 500с.
7. Introduction to Intel® FPGA IP Cores [Электронный ресурс]. – URL: <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_intro_to_megafunctions.pdf>