

## ПРОЕКТИРОВАНИЕ 8-МИ РАЗРЯДНОГО ОДНОТАКТОВОГО ПРОЦЕССОРА С RISC АРХИТЕКТУРОЙ НА БАЗЕ ПЛИС

*Мананников Никита Александрович, магистрант, КГТУ им. И.Раззакова, Кыргызстан, 720044, г.Бишкек, пр. Мира 66, e-mail: [manannikov\\_nikita@outlook.com](mailto:manannikov_nikita@outlook.com)*

*Исраилова Нелла Амантаевна, доцент кафедры ИВТ КГТУ им. И. Раззакова Кыргызстан, 720044, г.Бишкек, пр. Мира 66, e-mail: [inela.kstu@gmail.com](mailto:inela.kstu@gmail.com)*

**Аннотация.** В статье приведено описание процесса проектирования одноктактного 8-ми разрядного процессора с RISC архитектурой в программе «Logisim» и реализация на FPGA, который в последующим будет внедрен в учебный процесс по подготовке бакалавров направления 710100-“Информатика и вычислительная техника”.

**Ключевые слова:** процессор, CPU, RISC, FPGA+RISC, ПЛИС, Verilog

### *DESIGNING A 8-MULTI-DISCHARGE ONE-STEP PROCESSOR WITH RISC ARCHITECTURE ON THE FPGA BASIS*

*Manannikov Nikita Alexandrovich, master student, KSTU them. I.Razzakova, Kyrgyzstan, 720044, Bishkek, Mira Avenue 66, e-mail: [manannikov\\_nikita@outlook.com](mailto:manannikov_nikita@outlook.com)*

*Israilova Nella Amantaevna, Associate Professor of the Department of ICT in KSTU. I. Razzakova, Kyrgyzstan, 720044, Bishkek city, Mira pr. 66, e-mail: [inela.kstu@gmail.com](mailto:inela.kstu@gmail.com)*

**Annotation.** The article describes the process of designing a single-cycle 8-bit processor with RISC architecture in the "Logisim" program and implementation on FPGA, which in the future will be introduced into the educational process for the preparation of bachelors of direction 710100- "Computer science and computer facilities".

**Keywords:** processor, CPU, RISC, FPGA + RISC, FPGA, Verilog

### **ВВЕДЕНИЕ**

В учебном процессе у студентов возникают сложности в изучении и освоении учебного материала по курсам “Архитектура ЭВМ”, “ЭВМ и ПУ” из за отсутствия качественной и полноценной лабораторной базы и учебных пособий.

Процессор, проектирование и реализация которого описан в данной статье, позволит дать наглядное представление работы как самого процессора, так и его отдельных блоков, что будет способствовать повышению эффективности учебного процесса в целом.

Так же немаловажным является то, что данный процессор можно будет использовать не только в образовательных целях, но и для коммерческих проектов.

Кроме того, появились доступные по цене средства моделирования и макетирования в

виде конструкторских плат, использующих FPGA (Field Programmable Gate Array), называемых ПЛИС (Программируемая логическая интегральная схема).

Такие платы могут быть приобретены любым учебным заведением или даже частным лицом. Используя языки проектирования и описания аппаратуры такие как (Verilog) и предоставленные производителем FPGA инструментальные средства, каждый студент в состоянии самостоятельно спроектировать и построить сложную цифровую систему, включая процессор.

Проектирование процессора включает в себя следующие этапы:

1. Разработка системы команд.
2. Построение тракта данных.
3. Построение устройства управления.
4. Интегрировать проект в среду разработки Quartus.
5. Тестирование в ПЛИС.

#### Разработка системы команд

Одним из важных этапов является, разработка системы команд. Команды были разработаны с фиксированной длиной, что позволяет облегчить разработку процессора и не вводит в заблуждения студентов на начальном этапе изучения, так как не фиксированный формат команд чаще всего используется на CISC архитектуре и требует более высоких знаний в области построения процессоров.

Длина команды составляет 18 бит и выглядит следующим образом:

- **OP** – код операции, имеет размер 4 бита и содержит в себе самую закодированную команду.
- **AR1** – адрес 1-ой магистрали ПОН, имеет размер 2 бита.
- **AR2** – адрес 2-ой магистрали ПОН, имеет размер 2 бита.
- **Address/Data** – непосредственный адрес или данный, имеет размер 8 бит.
- **WR** – выборка регистров или памяти при записи, имеет размер 2 бита.

На данный момент процессор поддерживает 16 команд фиксированной длины (таблица 1).

№	Command	OP	AR1	AR2	Address/Data	WR
1	NOP	0000	XX	XX	XXXXXXXXXX	XX
2	LRB	0001	AA	XX	XXXXXXXXXX	AA
3	LRBI	0010	XX	XX	DDDDDDDD	AA
4	LB	0011	AA	AA	XXXXXXXXXX	AA
5	SB	0100	AA	AA	XXXXXXXXXX	AA
6	InSB	0101	XX	XX	XXXXXXXXXX	AA
7	ADD	0110	AA	AA	XXXXXXXXXX	AA
8	SUB	0111	AA	AA	XXXXXXXXXX	AA
9	MUL	1000	AA	AA	XXXXXXXXXX	AA
10	DIV	1001	AA	AA	XXXXXXXXXX	AA
11	SHL	1010	AA	AA	XXXXXXXXXX	AA
12	SHR	1011	AA	AA	XXXXXXXXXX	AA
13	INC	1100	AA	XX	XXXXXXXXXX	AA
14	DEC	1101	AA	XX	XXXXXXXXXX	AA
15	JE	1110	AA	AA	AAAAAAAA	XX
16	JMP	1111	XX	XX	AAAAAAAA	XX

#### Построение тракта данных

Теперь, когда определены все команды для данного процессора, был построен тракт данных (рис. 1). Так как у нас процессор одноктактный (за один такт выполняется каждая команда) то однозначно он представляет собой гарвардскую архитектуру и имеет раздельную память для инструкций и данных.

Тракт данных представляет собой следующие основные блоки:

- **ROM** – память инструкций содержит разрядность адреса 8-бит и данных 18-бит.
- **Reg** – блок РОН, включает в себя 4-ре восьмиразрядных регистра и мультиплексоры для их коммутации.
- **ALU** – арифметически логическое устройство содержит в себе блок сложения, вычитания, умножения, деления, логические сдвиги в право-лево, инкремент, декремент и проверка на равенство.
- **RAM** – память данных содержит разрядность адреса 8-бит и данных 8-бит.
- **Input A** – 8-ми разрядный регистр для ввода данных через внешний порт.
- **Output A, B** – два 8-ми разрядных регистра для вывода данных на внешние порты.

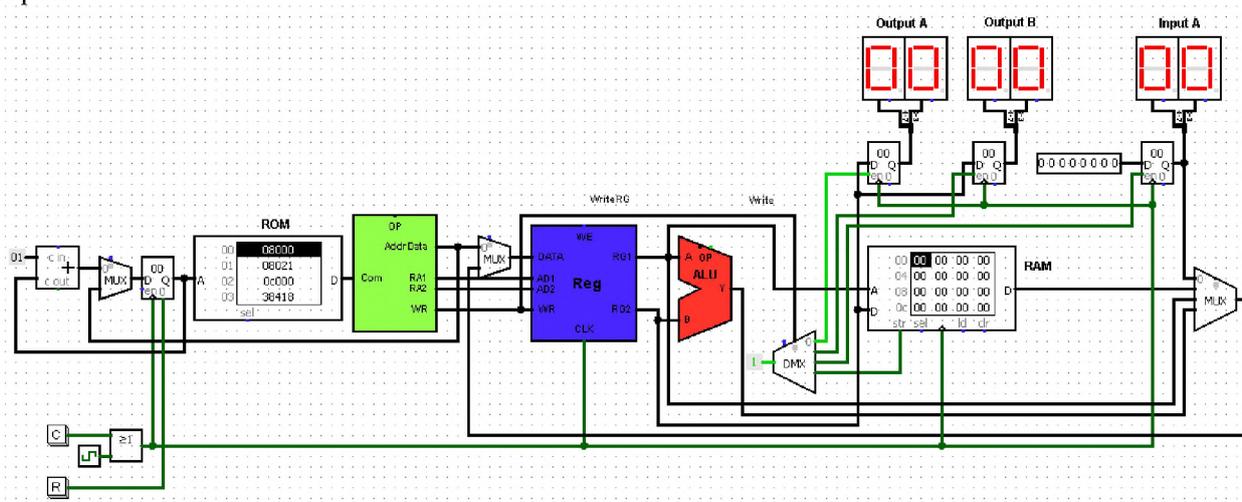


Рис 1. Завершённый тракт данных.

### Построение устройства управления

Следующим этапом был разработан управляющий дешифратор команд (его таблица истинности приведена в таблицы 2). Его основной целью служит, чтобы в нужный момент времени активировать те блоки, которые нужны для той или иной инструкции в момент её выполнения.

Внутренняя архитектура дешифратора приведена на рис 2.

№	ASSEM	OP	JMP	J	MUX	WRG	ALU OP	WRIT	RESULT
1	NOP	0000	0	0	0	0	000	0	00
2	LRB	0001	0	0	1	1	000	0	10
3	LRBI	0010	0	0	0	1	000	0	00
4	LB	0011	0	0	0	0	000	1	00
5	SB	0100	0	0	1	1	000	0	01
6	InSB	0101	0	0	1	1	000	0	00
7	ADD	0110	0	0	1	1	000	0	11
8	SUB	0111	0	0	1	1	001	0	11
9	MUL	1000	0	0	1	1	010	0	11
10	DIV	1001	0	0	1	1	011	0	11
11	SHL	1010	0	0	1	1	100	0	11
12	SHR	1011	0	0	1	1	101	0	11
13	INC	1100	0	0	1	1	110	0	11
14	DEC	1101	0	0	1	1	111	0	11
15	JE	1110	0	1	0	0	000	0	00
16	JMP	1111	1	0	0	0	000	0	00

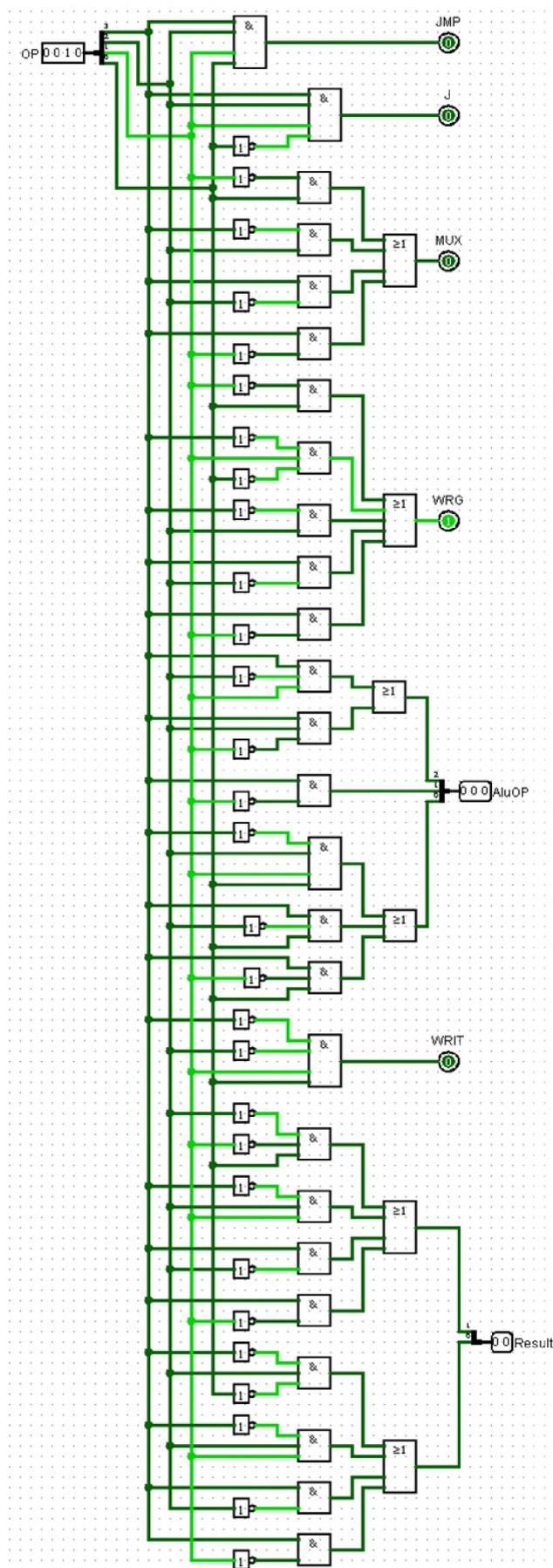


Рис 2. Архитектура дешифратора команд.

После того как был разработан дешифратор команд, его нужно соединить с трактом данных.

Теперь, когда были проделана все эти манипуляции, мы получили законченную структурную схему 8-ми разрядного процессора с поддержкой всех 16-ти команд (рис 3).

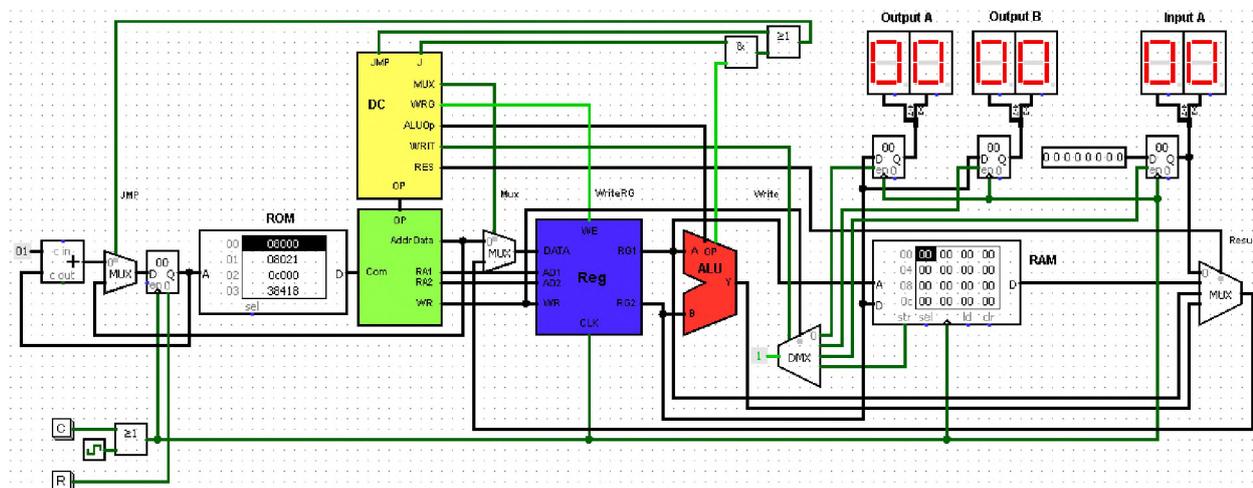


Рис 3. Законченная схема 8-ми битного процессора в программе «Logisim».

### Интегрировать проект в среду разработки «Quartus»

Интеграция процессора в ПЛИС позволила дать еще один шаг на разработку и убедиться в том, что данный процессор не только способен работать в программе Logisim, но и также в реальном железе.

Процессор был спроектирован на языке Verilog. RTL схема приведена на рис 4.

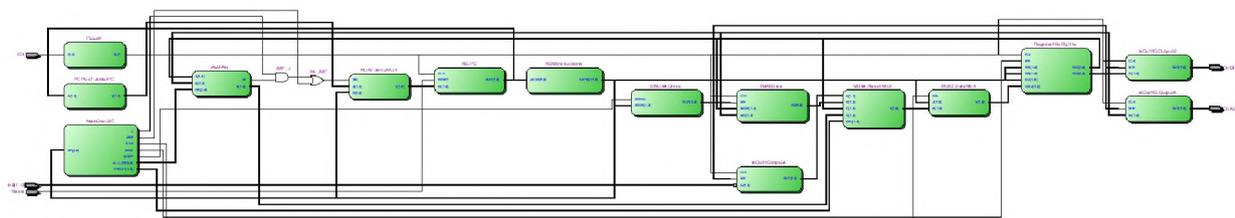


Рис 4. RTL схема в программе Quartus.

### Тестирование в ПЛИС

Тестирование проводилось на относительно дешевой отладочной плате RZ-EasyFPGA A2.1 на микросхеме ALTERA Cyclone IV EP4CE6E22C8N. Данная плата содержит в себе всю необходимую периферию с которой в последующем данный процессор сможет взаимодействовать.

Для теста была написана программа, для управления дисплеям LCD1602. Данная программа сначала проводила инициализацию дисплея, потом выводила надпись «HELLO» и в последующем опрашивала порт ввода (с помощью опроса), которому были назначены четыре кнопки, с 0 по 3 разряд.

Временные диаграммы данного процессора приведены на рис 5.



### **ЗАКЛЮЧЕНИЕ**

В заключении можно сказать, что начальный этап разработки был преодолён. Мы получили на выходе рабочий восьмиразрядный процессор с поддержкой всех команд, которые были реализованы на данный момент.

Также в ходе тестирования были выявлены некоторые проблемы, связанные с нехваткой некоторых команд. В дальнейшем это проблема будет решена.

Еще одним важным шагом будет реализация компилятора (так как на данный момент все команды вводятся вручную) и внешнего программатора.

Разработка и реализация процессора на базе ПЛИС послужит основой для развития лабораторной базы по специальным курсам аппаратного цикла дисциплин по подготовке бакалавров направления 710100-«Информатика и вычислительная техника»

### **СПИСОК ЛИТЕРАТУРЫ**

1. Э. Таненбаум, Т. Остин, «Архитектура компьютера» 6-е издание, Питер 2013.
2. Д. Паттерсон, Дж. Хеннесси, «Архитектура компьютера и проектирование компьютерных систем», Питер 2012.
3. В.В. Кириллов, «Архитектура базовой ЭВМ», Санкт-Петербург 2010.
4. В.В. Соловьев, «Основы языка проектирования цифровой аппаратуры VERILOG», Москва 2014.
5. Дэвид М. Харрис, Сара Л. Харрис, «Цифровая схемотехника и архитектура компьютера» 2-е издание, Morgan Kaufman 2013.