

РАЗРАБОТКА УПРАВЛЯЮЩЕГО АВТОМАТА МИКРОПРОЦЕССОРНОГО ЯДРА RISC-V ДЛЯ РЕАЛИЗАЦИИ В БАЗИСЕ ПЛИС

А.В. Строгонов¹, О.Л. Бордюжа²

¹ФГБОУ ВО «Воронежский государственный технический университет»

²ФГБОУ ВО «Воронежский государственный лесотехнический университет
имени Г.Ф. Морозова»

В работе рассматриваются вопросы разработки многотактного микропроцессорного ядра RISC-V с управляющим автоматом, созданным с помощью редактора конечных автоматов (State Machine Editor) в САПР Altera Quartus II для последующей реализации в базисе ПЛИС серии Cyclone V.

Ключевые слова: Ядро RISC-V, микропроцессор, программа, САПР, моделирование, конечные автоматы.

DEVELOPMENT OF A CONTROL MACHINE FOR A RISC-V MICROPROCESSOR CORE FOR IMPLEMENTATION IN THE FPGA BASIS

A.V. Strogonov¹, O.L. Bordyuzha²

¹Voronezh State Technical University

²Voronezh State University of Forestry and Technologies named after G.F. Morozov

The paper considers the development of a multi-cycle RISC-V microprocessor core with a control machine created using the State Machine Editor in the Altera Quartus II CAD system for subsequent implementation in the Cyclone V FPGA series.

Keywords: RISC-V core, microprocessor, program, CAD system, modeling, finite state machines.

Преимущество многотактного микропроцессорного ядра RISC-V заключается в том, что выполнение каждой команды происходит в несколько этапов.

Процессор может обходиться общей памятью для команд (ПЗУ) и данных (ОЗУ) а на разных этапах обработки требуется всего лишь один сумматор. У разных команд будет разное количество этапов, так что простые команды смогут выполняться быстрее, чем сложные.

Была поставлена задача реализовать управляющий автомат микропроцессорного ядра RISC-V с помощью редактора конечных автоматов (State Machine Editor) в САПР Altera Quartus II.

Особенностью данного проекта является то, что ПЗУ и ОЗУ микропроцессорного ядра реализованы на блочном ОЗУ ПЛИС размером 128 строк по 32 бита в каждой. Адресное пространство с 0 по 80 строку зарезервировано под тестовую программу (взята из работы [1]) а с 81 строки по 127 под ОЗУ данных. Существует проблема генерации САПР Quartus II конфигурационного файла из VHDL- кода. В ходе экспериментов было установлено, что для ее устранения необходимо между командами вставлять три пустые строки, по причине того, что счетчик команд увеличивает свое содержимое на 4, так как команды в RISC-V являются 4 байтовыми. На рис.1 слева продемонстрирована инициализация ПЗУ непосредственно из VHDL-кода. Используется двоичное представление содержимого памяти команд. Запись в ОЗУ осуществляется синхронно, начиная с адреса старше 80, а чтение, как памяти программ, так и данных идет в асинхронном режиме. По vhd1- коду памяти команд САПР генерирует заполненный значениями конфигурационный mif-файл (рис. 1).

На рис. 2 показан фрагмент схемы проекта многотактного процессорного ядра RISC-V в САПР Quartus II, а на рис.3 представлен управляющий автомат процессора, созданный с помощью редактора конечных автоматов (State Machine Editor). Функциональное моделирование и информационные потоки микропроцессорного ядра RISC-V показаны на примере выполнения первых двух команд типа I с кодами: 00500113 и 00C00193 (рис. 4).

Проект успешно реализуется в ПЛИС серии Cyclone V 5CSXFC6D6F31I7ES. Требуется адаптивных логических модулей – 2528; выделенных триггеров – 4449), 2 блока памяти емкостью 2048 бит. Максимальная тактовая частота синхросигнала, $f_{max\ CLK}$, (модель Slow 1100 мВ, 100 C) составила 67 МГц против 58 МГц для одноктактного процессорного ядра RISC-V в базе ПЛИС Cyclone V.

```

ENTITY MEMORY_SYNTH IS
PORT (CLK, Write_Enable: IN STD_LOGIC;
ADDR : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
DATA : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
OUTP : OUT STD_LOGIC_VECTOR(31 DOWNTO 0));
END MEMORY_SYNTH;

ARCHITECTURE a OF MEMORY_SYNTH IS
    TYPE vec_array IS ARRAY(0 TO 127) OF
STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL RAM : vec_array := (
        0 => "00000000010100000000000100010011",
        1 => "00000000000000000000000000000000",
        2 => "00000000000000000000000000000000",
        3 => "00000000000000000000000000000000",
        4 => "00000000110000000000000110010011",
        5 => "00000000000000000000000000000000",
        6 => "00000000000000000000000000000000",
        7 => "00000000000000000000000000000000",
        ...
        80 => "00000000001000010000000001100011",
        OTHERS => (OTHERS => '0'));
BEGIN
PROCESS (CLK, ADDR, Write_Enable)
BEGIN
    IF rising_edge(CLK) THEN
        IF Write_Enable = '1' THEN
            IF to_integer(unsigned(ADDR)) > 80
THEN
                RAM(to_integer(unsigned(ADDR))) <= DATA;
            END IF;
        END IF;
    ND IF;
    outp <= RAM(to_integer(unsigned(ADDR)));
END PROCESS;
END a;

```

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	00500113	00000000	00000000	00000000	00C00193	00000000	00000000	00000000
8	FF718393	00000000	00000000	00000000	0023E233	00000000	00000000	00000000
16	0041F2B3	00000000	00000000	00000000	004282B3	00000000	00000000	00000000
24	02728863	00000000	00000000	00000000	0041A233	00000000	00000000	00000000
32	00020463	00000000	00000000	00000000	00000293	00000000	00000000	00000000
40	0023A233	00000000	00000000	00000000	005203B3	00000000	00000000	00000000
48	402383B3	00000000	00000000	00000000	0471AA23	00000000	00000000	00000000
56	06002103	00000000	00000000	00000000	005104B3	00000000	00000000	00000000
64	008001EF	00000000	00000000	00000000	00100113	00000000	00000000	00000000
72	00910133	00000000	00000000	00000000	0221A023	00000000	00000000	00000000
80	00210063	00000000	00000000	00000000	00000000	00000000	00000000	00000000
88	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
96	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
104	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
112	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
120	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

Рисунок 1 – Фрагмент VHDL-кода ОЗУ (слева) и Конфигурационный mif-файл ОЗУ созданный САПР Quartus II по vhdл-коду (справа)

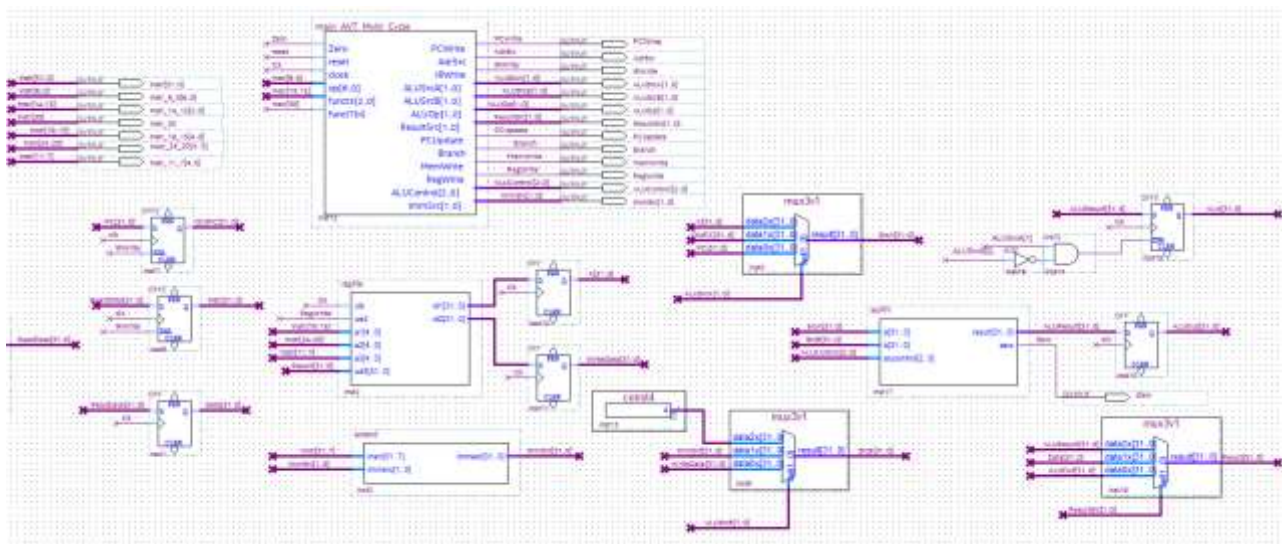


Рисунок 2 – Фрагмент схемы проекта многотактного микропроцессорного ядра RISC-V в САПР Quartus II

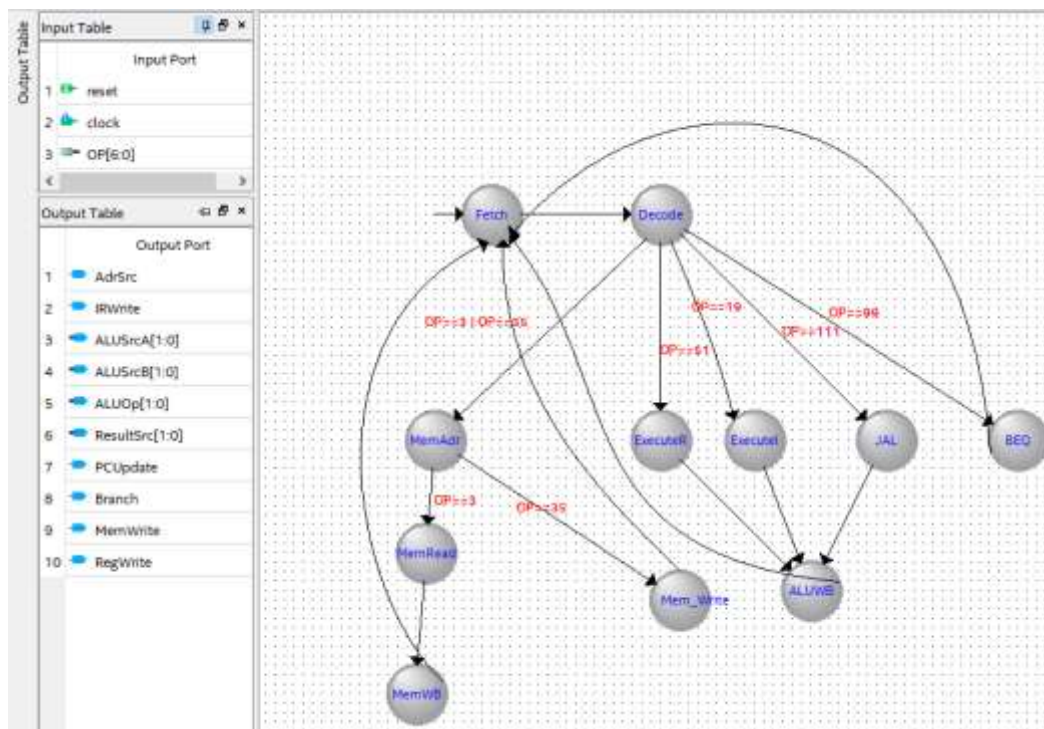


Рисунок 3 – Управляющий автомат многотактного микропроцессорного ядра RISC-V созданный с помощью редактора конечных автоматов (State Machine Editor)

References

1. Harris, Sarah L. Digital Circuit Design and Computer Architecture: RISC-V / Sarah L. Harris, David Harris; trans. from English by V.S. Yatsenkova, A.Yu. Romanov; edited by A. Yu. Romanov. - M.: DMK Press, 2021. - 810 p.
2. Stroganov, A. V. Open Source Software Tools for Designing RISC-V Processor Cores / A. V. Stroganov, O. L. Bordyuzha, A. I. Stroganov // Modeling of Information Systems and Technologies. Proceedings of the International Scientific and Practical Conference. Voronezh, 2024. -P. 582-589.
3. Stroganov, A. V. International Experience in Developing RISC-V Processor Cores and Open Source Software Tools for Their Design / A. V. Stroganov, O. L. Bordyuzha, A. I. Stroganov // Electronics: Science, Technology, Business. 2024. No. 7 (238). -P. 156-164.
4. Gorbunov, V. G. Comparative analysis of the Prometheus methods and fuzzy relations in decision-making / V. G. Gorbunov, O. L. Bordyuzha, A. A. Pak // Modeling of systems and processes. -2024. -V.17, No. 1. - P. 42-56. - DOI 10.12737/2219-0767-2024-17-1-42-56. - EDNLSMBMD/