

## РЕАЛИЗАЦИЯ ПОДМНОЖЕСТВА ИНСТРУКЦИЙ СОФТ ПРОЦЕССОРА АРХИТЕКТУРЫ RISC-V RV32I ДЛЯ ПЛИС ТИПА FPGA

Бурак А.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Станкевич А.В. – канд. техн. наук

Рассматривается реализация RISC-V процессора для ПЛИС типа FPGA. Процессор реализует подмножество команд RV32I за исключением инструкций FENCE, ECALL, EBREAK.

RISC-V — это открытая архитектура со стандартным набором команд (ISA), основанная на концепции RISC (reduced instruction set computer — компьютер с простым набором команд). В отличие от большинства других ISA, спецификация RISC-V ISA доступна для бесплатного использования, реализации в ПЛИС и непосредственно в кремнии.

В спецификации RISC-V ISA определено около 50 инструкций. Так же есть стандартные расширения (C, M, A, F и D). Обязательным для реализации является только подмножество команд I (Integer), в которое входят инструкции по работе с памятью (загрузка в регистр из памяти и сохранение из регистра в память), ветвления, а также инструкции арифметических и логических операций и несколько служебных. Все команды разделены на 6 типов:

- 1) R — операции АЛУ типа регистр – регистр;
- 2) I — операции АЛУ с непосредственным значением в команде;
- 3) S — операции загрузки/сохранения;
- 4) B — условная передача управления;
- 5) U — операции с расширенным непосредственным значением;
- 6) J — безусловная передача управления.

Форматы типов инструкций представлены в таблице 1.

Таблица 1 – Форматы инструкций

Тип	31	25	24	20	19	15	14	12	11	7	6	0
R	funct7		rs2		rs1		funct3		rd		opcode	
I	imm[11:0]				rs1		funct3		rd		opcode	
S	imm[11:5]		rs2		rs1		funct3		imm[4:0]		opcode	
B	imm[12 10:5]		rs2		rs1		funct3		imm[4:1 11]		opcode	
U	imm[31:12]								rd		opcode	
J	imm[20 10:1 11 19:2]								rd		opcode	

В рамках проекта RISC-V были разработаны: 5-ти ступенчатый конвейерный процессор, коммутатор шины данных, контроллер блочной памяти, блок ввода-вывода и универсальный асинхронный приемопередатчик. Структурная схема реконфигурируемой системы с софтверным процессором архитектуры RISC-V на базе системы на кристалле Xilinx Zynq приведена на рисунке 1.

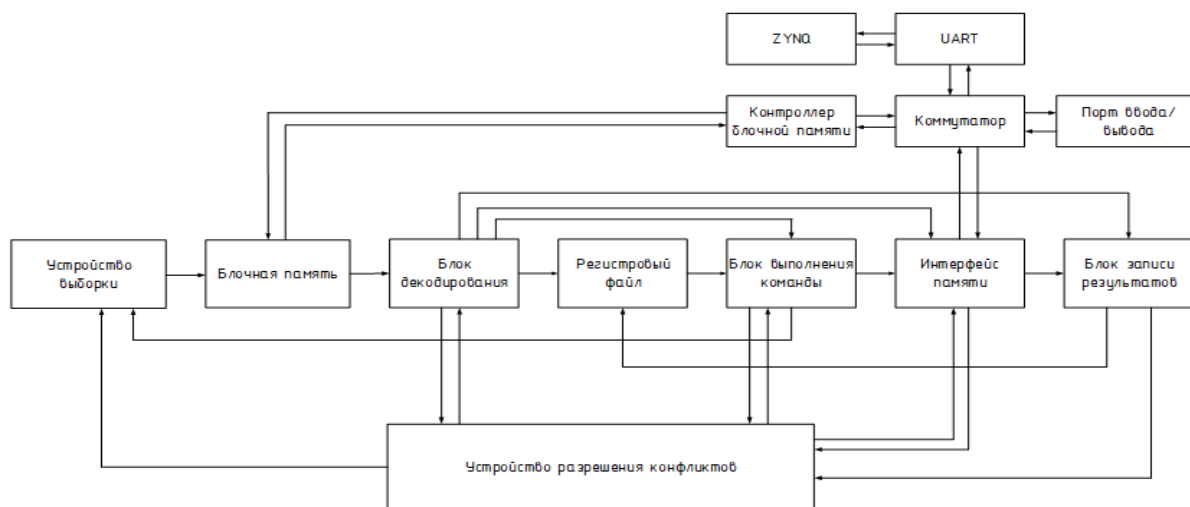


Рисунок 1 – Структурная схема проекта

Обмен данными между процессором и периферийными блоками осуществляется по шине данных. Временная диаграмма операций чтения и записи представлена на рисунке 2, где A0, A1, A2

являются адресами по которым производится чтение данных; D0, D1, D2 соответственно прочитанным данным, A3, A4 и D3, D4 являются соответственно адресами и данными для записи в память; W, HW, B размеры читаемых данных (W – 4 байта, HW – 2 байта, B – байт).

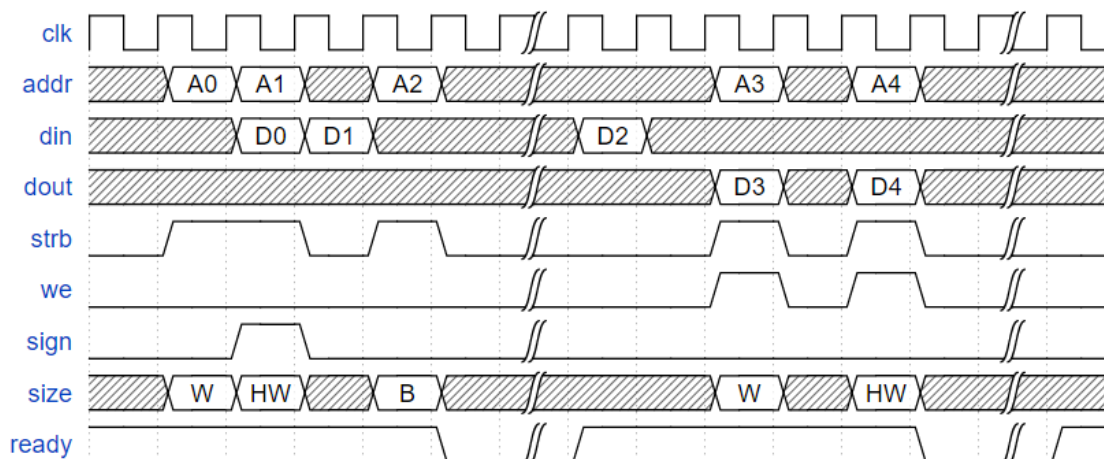


Рисунок 2 – Временная диаграмма операций чтения и записи по шине данных.

Для базовой верификации процессора применялась целенаправленное тестирование. К процессору подключались память команд, содержащая все поддерживаемые команды процессором, а также некоторые их комбинации, как приводящие, так и не приводящие к возникновению конфликтов в конвейере и подключалась внешняя память данных. В случае если после запуска тестовой последовательности содержание памяти данных совпадает с предполагаемым, то процессор можно считать условно рабочим, т.к. правильное выполнение программы маловероятно при наличии ошибок в аппаратуре.

Проект был подготовлен в САПР Vivado 2019.1 для отладочной платы Zybo с кристаллом серии Xilinx Zynq (xc7z010clg400-1). Используя доступные в САПР стратегии синтеза, размещения и трассировки проекта, такие как Flow\_PerfOptimized\_high и Performance\_NetDelay\_high, позволило выполнить ограничение на тактовую частоту работы процессора в 100 МГц. Затраченные ресурсы ПЛИС представлены в таблице 2.

Таблица 1 – Форматы инструкций

Ресурс	Затрачено	Доступно	Затрачено, %
LUT	1798	17600	10.22
LUTRAM	19	6000	0.32
FF	1759	35200	5.0
BRAM	8	60	13.3
IO	8	100	8.0
BUFG	1	32	3.13

Можно оценить производительность процессора на тестовом наборе SPECINT2000 в данном наборе содержится примерно 25% команд загрузки, 10% команд сохранения, 11% команд условного перехода, 2% команд безусловного перехода, и 52% команд типа R. Среднее количество необходимых тактов для выполнения 1-й инструкции (CPI) находится по формуле 1.

$$CPI = \sum_i C_i \cdot N_i \quad (1),$$

где  $C_i$  — количество тактов, необходимых для выполнения  $i$ -го типа инструкций;

$N_i$  — относительное количество  $i$ -х инструкций в тестовом наборе.

Подставив значения в формулу 1 и вычислив, получаем  $CPI = 2,1$ . Производительность может быть оценена по формуле 2.

$$P = \frac{F_{cpu}}{CPI} \quad (2),$$

где  $F_{cpu}$  — тактовая частота процессора.

Подставив значения в формулу 2 получаем что производительность процессора равна примерно 47.6 миллионов операций в секунду.

**Список использованных источников:**

1. Waterman A. The RISC-V Instruction Set Manual Volume I: Unprivileged ISA/ Waterman A., Asanovi'c K.