Выполнение лабораторной работы «**lab\_scr1\_rtl**»

Таблица 1 Условие согласно варианту № 7

|  |  |  |  |
| --- | --- | --- | --- |
| №  вар | Instr | Arch #1 | Arch #2 |
| 7 | **JALR** | **RVIM -FAST\_MUL** | **RVIM +FAST\_MUL** |

**Часть I**

Для выполнения данной части был запущен соответствующий названию инструкции тест *isa/rv32ui/jalr.S*.

На осциллограмме был найден отрезок выполнения исследуемой инструкции (рис. 1).

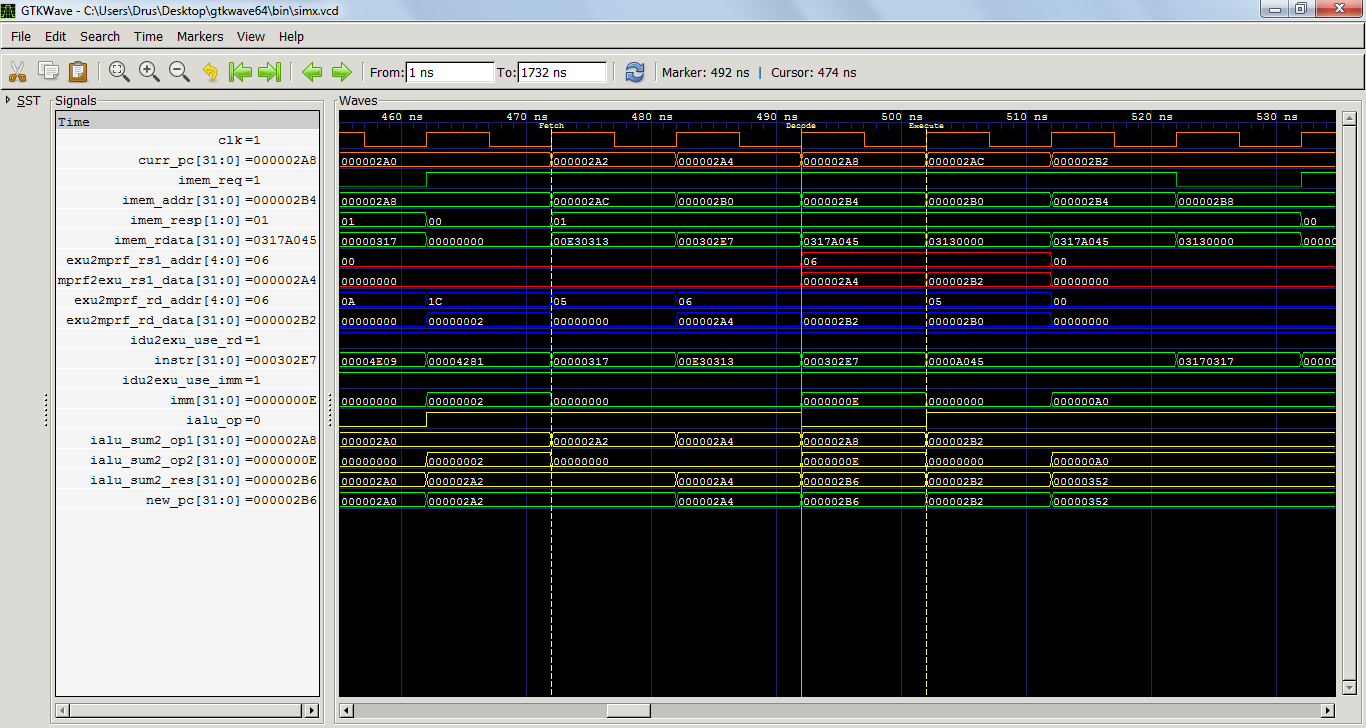


Рисунок 1 Отрезок выполнения инструкции JALR

Этому отрезку соответствует следующая часть дампа теста из файла *.results/jalr.dump*, представленная на рисунке 2, и часть из файла *jalr.s*, представленная на рисунке 3.

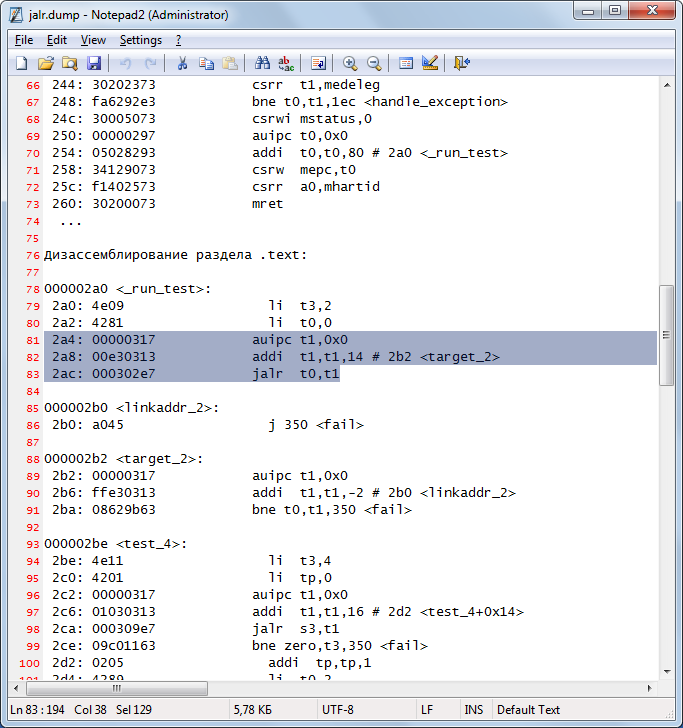


Рисунок 2 Часть дампа выполнения теста

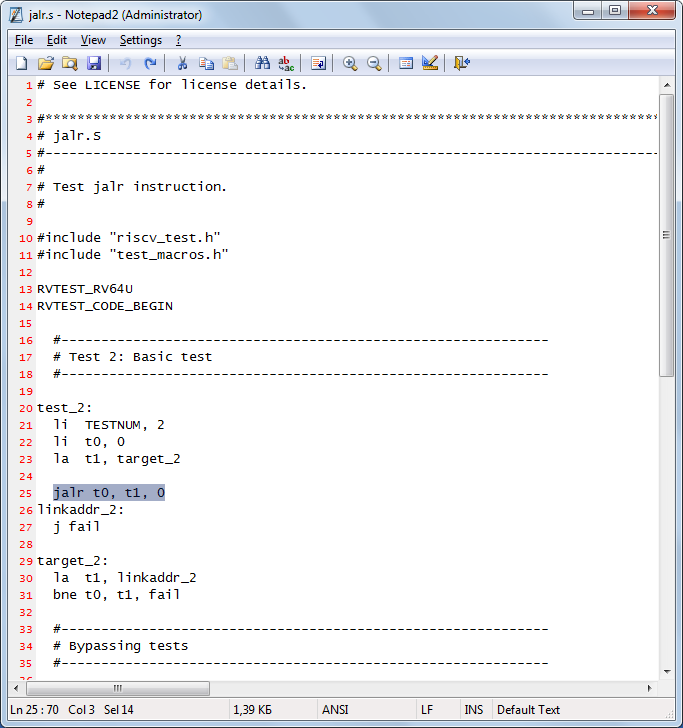


Рисунок 3 Исследуемая часть кода теста

Стадия выборки команды отмечена на осциллограмме маркером «Fetch». На этой стадии получен ответ памяти инструкций (imem\_resp = 01), значение imem\_addr соответствует адресу из дампа (рис. 2), по которому расположена исполняемая инструкция. На следующем такте в imem\_rdata располагается код инструкции. Для подтверждения правильности COP данная инструкция изучена в симуляторе Venus (рисунок 4).

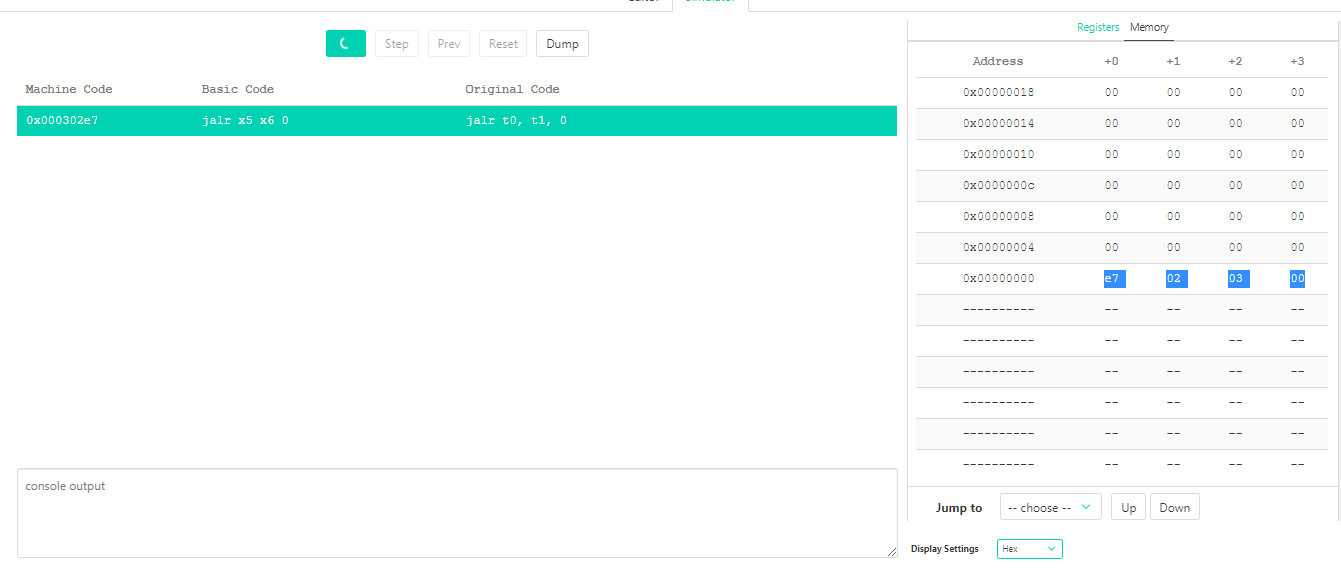


Рисунок 4 Выполнение инструкции в симуляторе Venus для проверки кода операции

Стадия декодирования инструкции отмечена, как «Decode». На осциллограмме можно наблюдать, что адрес регистра rs1 в регистровой памяти (exu2mprf\_rs1\_addr = 06) соответствует регистру x6 (он же t1). Значение адреса регистра назначения rd (exu2mprf\_rd\_addr = 05) x5(t0).

Стадия исполнения инструкции – «Execute». Ее выполнение происходит при curr\_pc = 0x2AC, что соответствует дампу. На этой стадии new\_pc = 0x2B2, что, согласно дампу, является адресом следующей исполняемой команды и метки, на которую был выполнен «jump».

**Часть II**

Для выполнения второй части лабораторной работы установим архитектуру в соответствии с вариантом в файле src/includes/scr1\_arch\_description.svh: закомментируем параметр `define SCR1\_FAST\_MUL (рисунок 5).

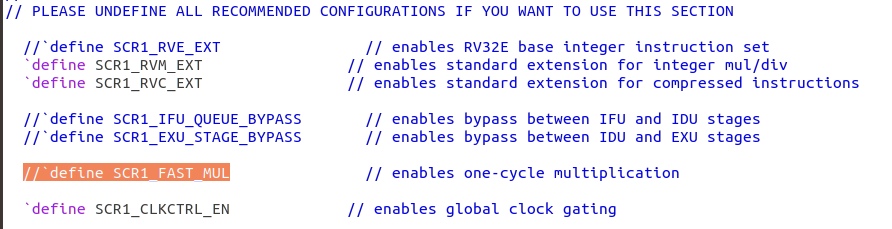


Рисунок 5 Изменение конфигурации ядра в соответствии с Arch #1

Выполним тесты Coremark и Dhrystone, результаты которых представлены на рисунках 6 и 7 соответственно.

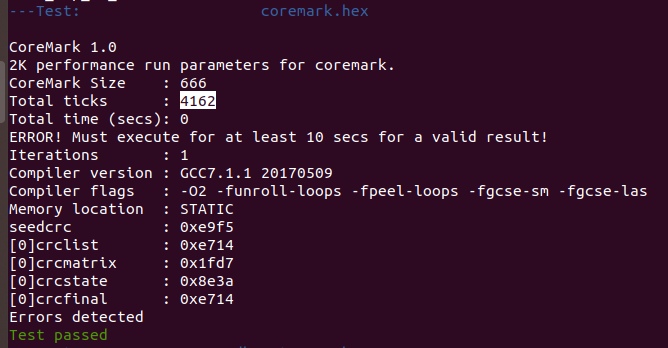


Рисунок 6 Результат выполнения теста Coremark с конфигурацией Arch #1

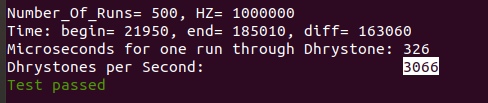


Рисунок 7 Результат выполнения теста Dhrystone с конфигурацией Arch #1

Для выполнения тестирования с конфигурацией Arch #2 необходимо раскомментировать в файле src/includes/scr1\_arch\_description.svh параметр `define SCR1\_FAST\_MUL (рисунок 8).

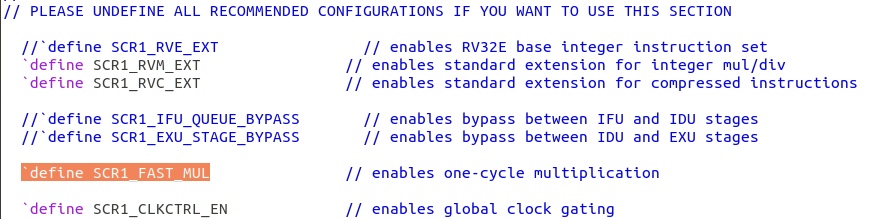


Рисунок 8

Результаты выполнения тестов Coremark и Dhrystone представлены на рисунках 9 и 10 соответственно.

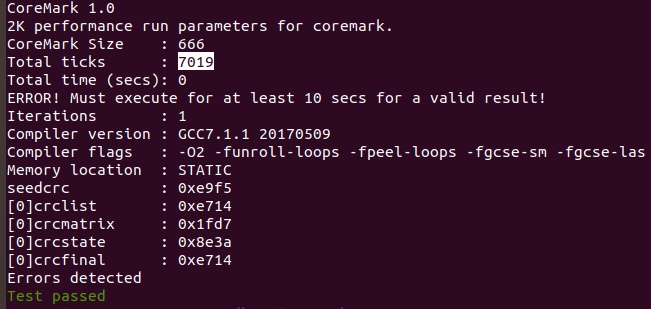


Рисунок 9 Результат выполнения теста Coremark с конфигурацией Arch #2

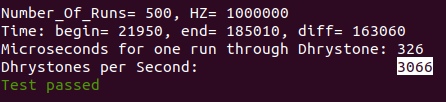


Рисунок 10 Результат выполнения теста Dhrystone с конфигурацией Arch #2

Результаты сравнения тестов при разных конфигурациях приведены в таблице 2.

Таблица 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Архитектура | Производительность | | Объем занимаемой памяти, КБайт | |
| Coremark, total ticks | Dhrystone, dhrystones per second | Coremark | Dhrystone |
| Arch #1 | 4162 | 3066 | 89,9 | 45,6 |
| Arch #2 | 7019 | 3066 | 89,9 | 45,6 |