**Хлыстун Дмитрий, группа P41193**

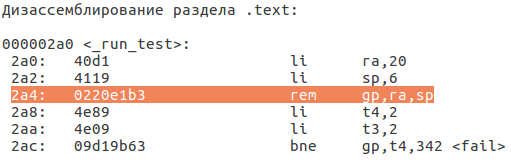
**Lab SCR1 pipeline**

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант | Команда | Arch #1 | Arch #2 |
| 8 | REM | RVIMC | RVIC |

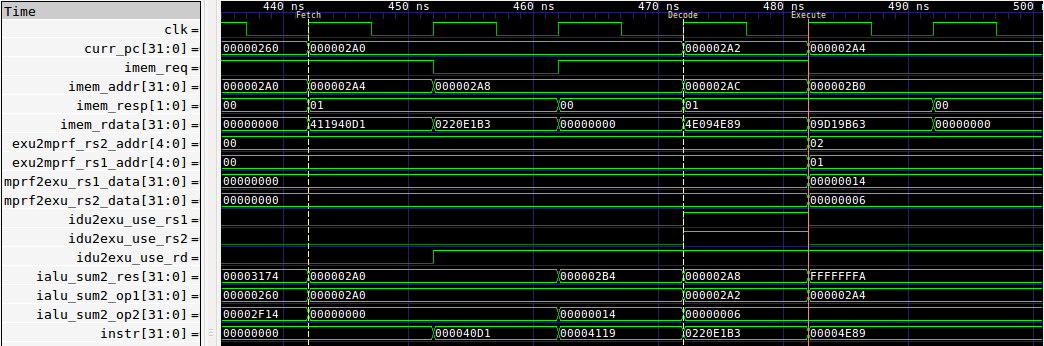
**1 Часть**

Для разборки был выбран файл с тестом «rem.S».

Отрывок из dump-файла, где используется команда rem:

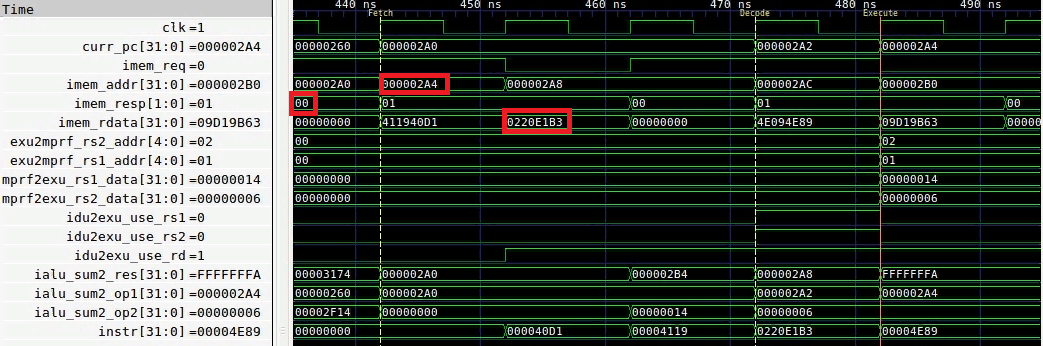


Результирующая wave-form:

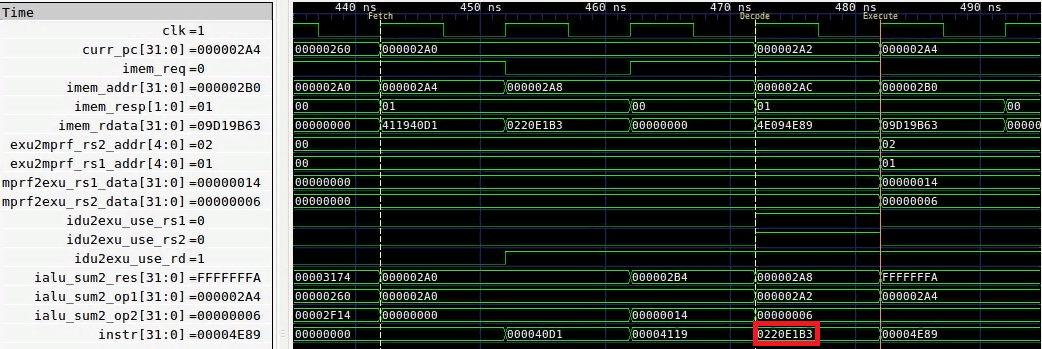


Описание сигналов:

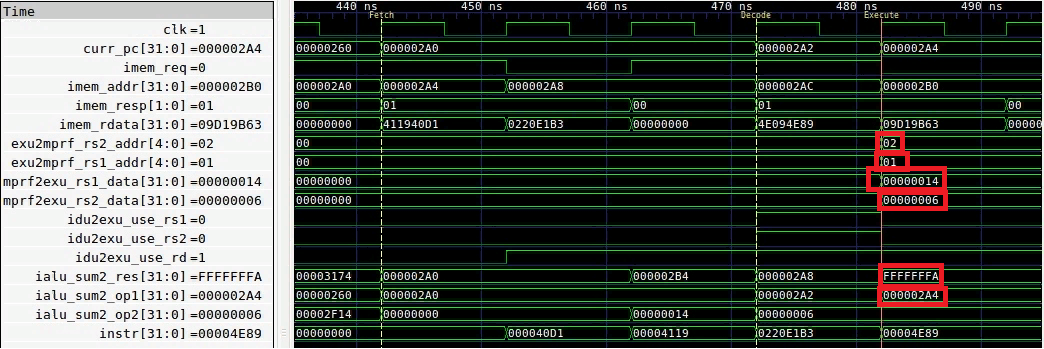
* clk - тактовый импульс
* curr\_pc - текущее значение счетчика команд, соответствует стадии Execution (000002A4)
* набор сигналов для Instruction Fetch:
  + imem\_req - запрос от процессора в память инструкций.
  + imem\_addr - адрес запроса памяти инструкций (000002A4)
  + imem\_resp - ответ памяти инструкций (00)
  + imem\_rdata - данные чтения памяти инструкций (0220Е1В3)



* набор сигналов для Decode:
  + instr – закодированная инструкция (0220Е1В3)
  + idu2exu\_use\_rs1 – сигнал о том, что в команде есть rs1 (0)
  + idu2exu\_use\_rs2 – сигнал о том, что в команде есть rs2 (0)



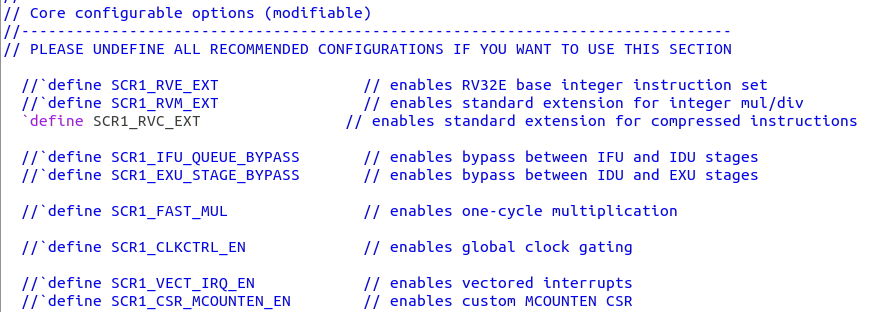
* Набор сигналов для Execute:
  + exu2mprf\_rs2\_addr – номер регистра в котором лежит адрес для записи в память. (02), x2 = sp
  + exu2mprf\_rs2\_data – значение для записи в регистровый файл (00000006).
  + exu2mprf\_rs1\_addr – номер регистра в котором лежит адрес. (01) x1;
  + exu2mprf\_rs1\_data – значение регистра x1 (00000014);
  + ialu\_sum2\_op1 – первый операнд АЛУ (000002A4) Это значение регистра x2(sp);
  + ialu\_sum2\_op2 – второй операнд АЛУ (FFFFFFFA) Это смещение;
  + ialu\_sum2\_res – результат АЛУ.



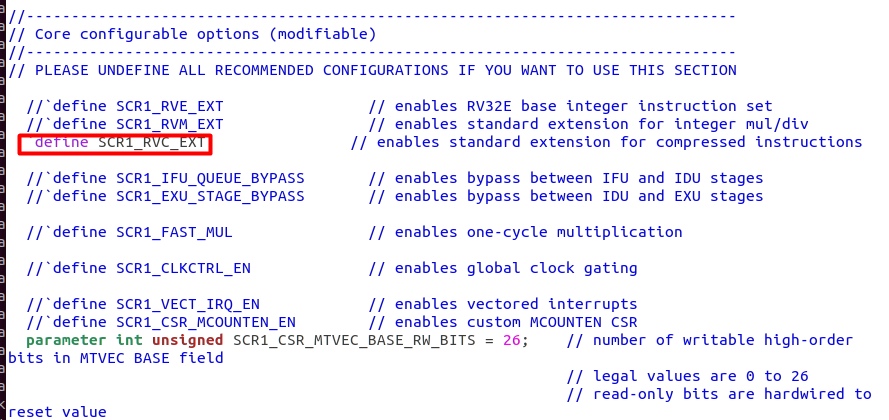
**2 Часть**

Были запущены два теста-бенчмарка Coremark и Dhrystone для двух архитектур по заданию: RVIMC и RVIC. Для этого был изменён файл «scr1\_arch\_description.svh»:

RVIC -IFU\_BYPASS



RVIC



Вызов: make run\_verilator\_wf BUS=AHB ARCH=IC IPIC=0

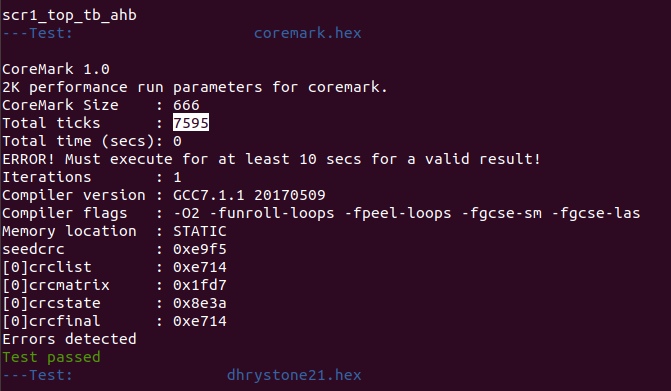
Были получены следующее результаты:

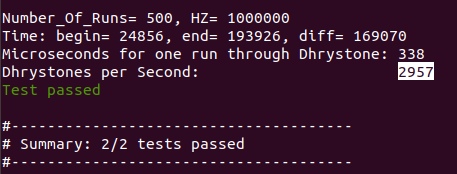
|  |  |  |
| --- | --- | --- |
| Тест | RVIМC | RVIC |
| Coremark (Total ticks) | 7595 Total ticks | 11397 Total ticks |
| Coremark (size memory) | 75,5Кбайт | 78,6 Кбайт |
| Dhrystone (per Seconds) | 2957 per Seconds | 2922 per Seconds |
| Dhrystone (size memory) | 41,6 Кбайт | 45,3 Кбайт |

При выборе разных архитектур, для Dhrystone, количество итераций в секунду изменилось на 35, а размер занимаемой памяти – на 3,7 Кбайт. Для Coremark количество итераций в секунду изменяется на 3802, при это размер занимаемой памяти увеличился.

Из полученных результатов можно сделать вывод, что Coremark занимает почти в два раза больше памяти, чем тест Dhrystone, но при этом количество итераций в секунду выше, по сравнению с Dhrystone.

Скриншоты тест-бенчмарков Coremark и Dhrystone для ARCH = RVIМC





Скриншоты тест-бенчмарков Coremark и Dhrystone для ARCH = RVIC

