



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «ИУ, Информатика и системы управления»

КАФЕДРА «ИУ7, Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа №2 *по дисциплине* **“Архитектура ЭВМ”**

Тема: Изучение принципов работы микропроцессорного ядра RISC-V

Студент: Андреев А.А.

Группа: ИУ7-54Б

Преподаватель: Попов А.Ю.

Москва - 2021 г.

Оглавление

Оглавление	1
1. Функциональная схема разрабатываемой системы на кристалле	2
2. Копия экрана готового модуля в системе проектирования систем на кристалле Altera Qsys	2
3. Таблица распределение адресов модулей в системе на кристалле	3
4. Код программного проекта Nios II Software Build Tools for Eclipse	3
5. Результаты тестирования PSoC на отладочной плате	4

Задание 1.

Были выполнены все подготовительные операции:

- создан каталог для хранения файлов *C://User/Andreev*
- открыт терминал в данном каталоге
- получена копия репозитория при помощи команды *git clone*
<https://gitlab.com/sibragimov/riscv-lab.git>

В результате в каталоге был создан подкаталог *riscv-lab*, внутри него *taiga* и *src*;

В процессе выполнения задания были выполнены следующие действия:

- Ознакомление с теоретической частью и ознакомление с примерами
- Переход в подкаталог *src* командой *cd riscv-lab/src*
- Выполнена сборка при помощи команды *cmake*. Создан файл *test.hex*, содержащий шестнадцатеричное представление программы, а в окне терминала отображается дизассемблированный листинг.
- Создан новый файл, содержащий текст программы по варианту 3. Помещен в *src*. Текст программы сохранен в файл с расширением *.s*.
- Изучен текст программы по 3 варианту. Помещен в отчет код программы (Листинг 1).
- Изменен *makefile* с адресом на файл программы.
- Выполнена компиляция *gmake* и получен *.hex* файл.

Листинг 1: Код программы, Часть 1

```
1.      .section .text
2.      .globl _start;
3.      len = 8 #Размер массива
4.      enroll = 1 #Количество обрабатываемых элементов за
одну итерацию
5.      elem_sz = 4 #Размер одного элемента массива
6.
7. _start:
8.      la x1, _x
9.      addi x20, x1, elem_sz*(len-1) #Адрес последнего
элемента
10.   lp:
11.      lw x2, 0(x1)
12.      add x31, x31, x2 #!
13.      addi x1, x1, elem_sz*enroll
14.      bne x1, x20, lp
15.      addi x31, x31, 1
16.   lp2: j lp2
```

Листинг 1: Код программы, Часть 2

```
17.      .section .data
18.  _x:    .4byte 0x1
19.      .4byte 0x2
20.      .4byte 0x3
21.      .4byte 0x4
22.      .4byte 0x5
23.      .4byte 0x6
24.      .4byte 0x7
25.      .4byte 0x8
```

Задание 2.

В ходе выполнения задания:

- был открыт проект в Quartus, запущен
- выполнен синтез проекта
- запущена симуляция в modelsim при помощи *run 460us* (см. Рисунок 2.3)
- Изучен список сигналов в окне wave
- Получен снимок экрана, содержащий временную диаграмму выполнения стадий выборки и диспетчеризации команды с адресом 80000014 (см. Рисунок 2.2), 1-я для третьего варианта (см. Рисунок 2.1).

Вариант	1	2	3	4	5	6	7	8
Адрес команды, номер итерации	8000000с, 1-я	80000010, 1-я	80000014, 1-я	80000018, 1-я	8000001с, 1-я	80000020, 1-я	80000024, 1-я	80000028, 1-я
Вариант	10	11	12	13	14	15	16	17
Адрес команды, номер итерации	80000030, 1-я	80000034, 1-я	8000000с, 2-я	80000010, 2-я	80000014, 2-я	80000018, 2-я	8000001с, 2-я	80000024, 2-я
Вариант	19	20	21	22	23	—	—	—
Адрес команды, номер итерации	80000028, 2-я	8000002с, 2-я	80000030, 2-я	80000034, 2-я	80000038, 2-я	—	—	—

Рисунок 2.1: Распределение вариантов и адресов

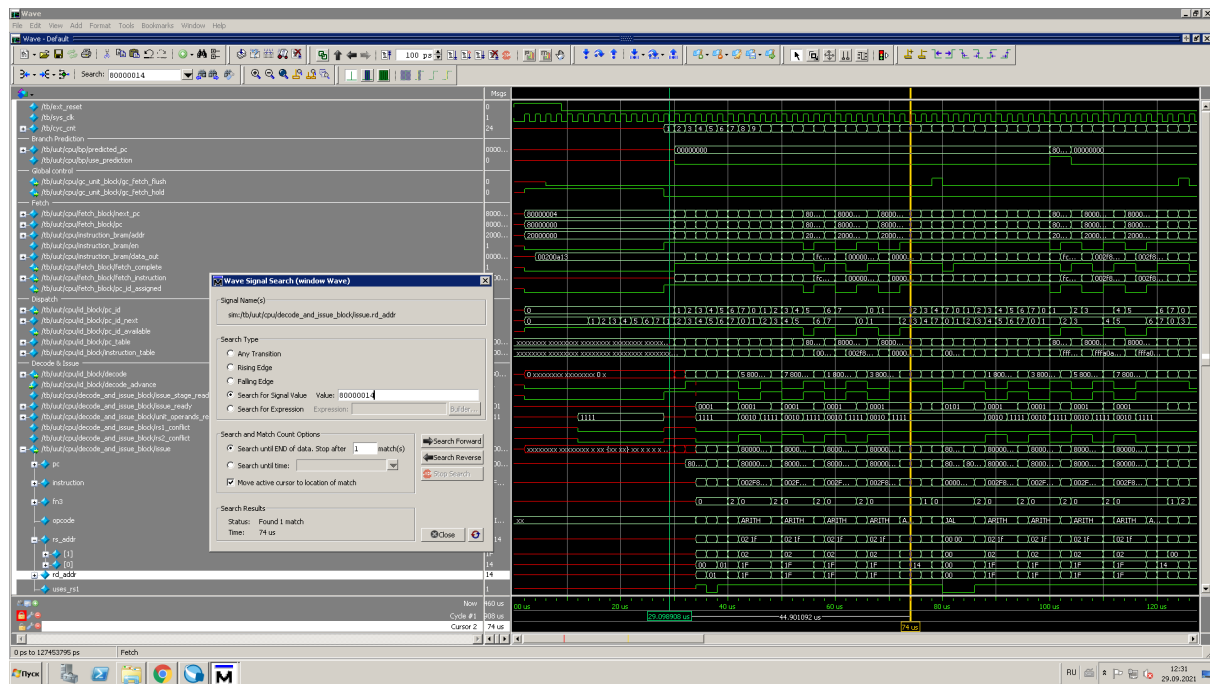


Рисунок 2.2: Снимок экрана с адресом 80000014

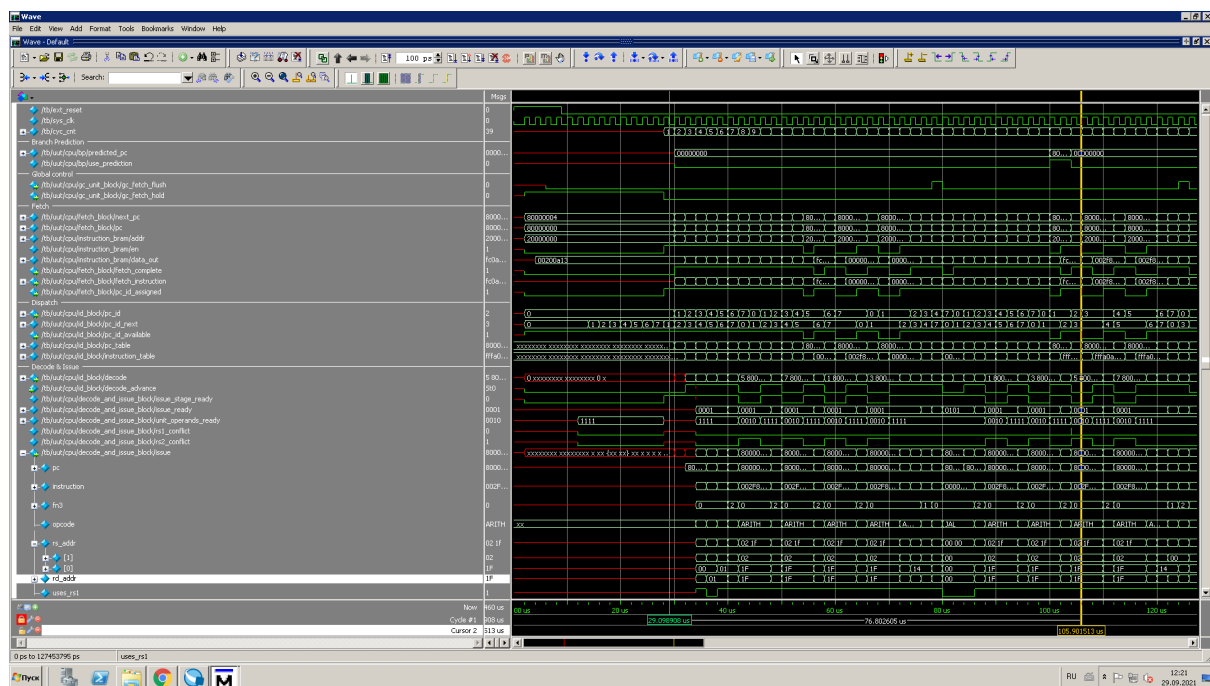


Рисунок 2.3: run 460us

Задание 3.

В ходе выполнения задания:

- В соответствии с таблицей (см. Рисунок 3.1) получен снимок экрана, содержащий стадии декодирования и планирования на выполнение команды с указанным адресом варианта 3 (см. Рисунок 3.2 и Рисунок 3.3).

Вариант	1	2	3	4	5	6	7	8
Адрес команды, номер итерации	80000018, 1-я	8000001с, 1-я	80000020, 1-я	80000024, 1-я	80000028, 1-я	8000002с, 1-я	80000030, 1-я	80000034, 1-я
Вариант	10	11	12	13	14	15	16	17
Адрес команды, номер итерации	80000010, 2-я	80000014, 2-я	80000018, 2-я	8000001с, 2-я	80000020, 2-я	80000024, 2-я	80000028, 2-я	80000030, 2-я
Вариант	19	20	21	22	23	24	25	26
Адрес команды, номер итерации	80000034, 2-я	80000038, 2-я	См. вариант 1	См. вариант 2	См. вариант 3	См. вариант 4	См. вариант 5	См. вариант 6

Рисунок 3.1: Таблица вариантов

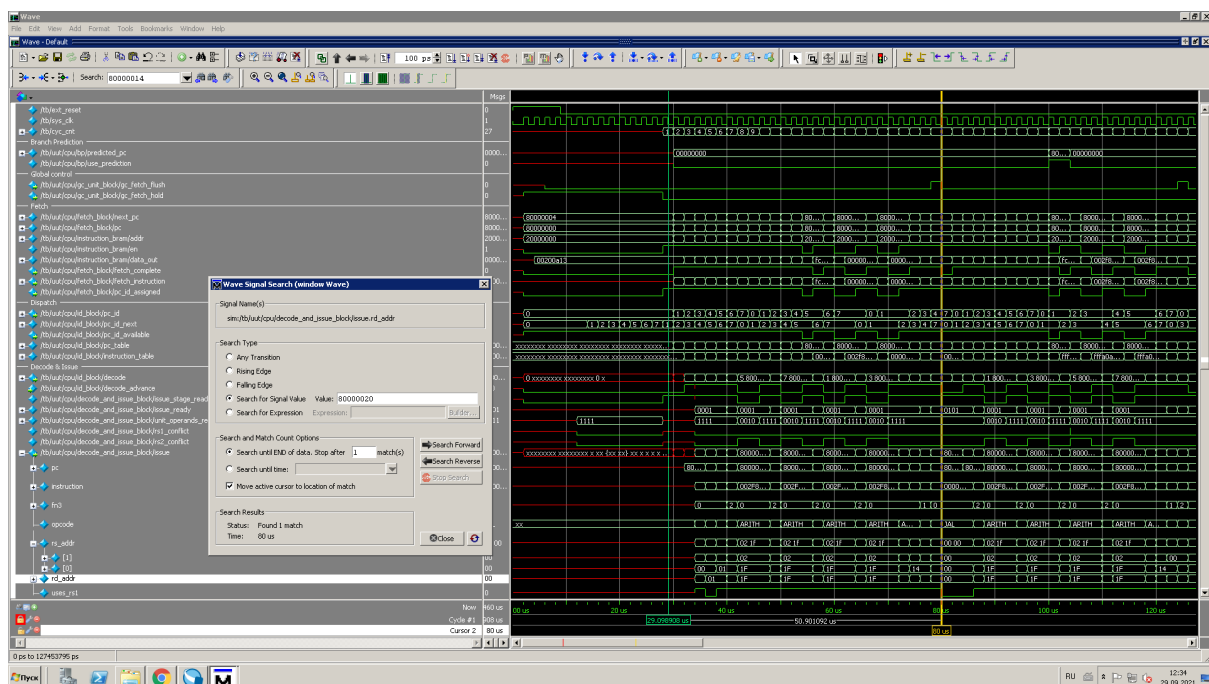


Рисунок 3.2: Снимок экрана со стадиями декодирования по третьему варианту, Часть 1

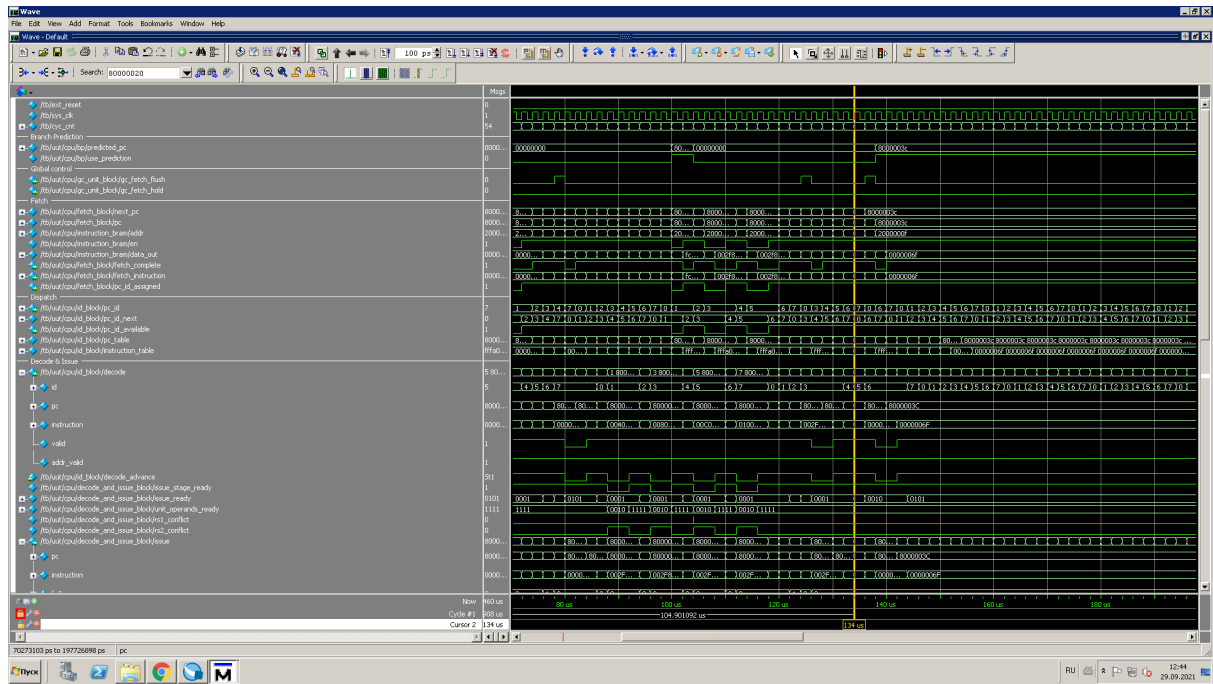


Рисунок 3.3: Снимок экрана со стадиями декодирования по третьему варианту, Часть 2

Задание 4.

В ходе выполнения задания:

- В соответствии с таблицей (см. Рисунок 4.1) получен снимок экрана, содержащий временную диаграмму выполнения команды с указанным адресом варианта 3 (см. Рисунок 4.2).

Вариант	1	2	3	4	5	6	7	8
Адрес команды, номер итерации	80000000	80000004	80000008	8000000с, 1-я	80000010, 1-я	80000014, 1-я	80000018, 1-я	8000001с, 1-я
Вариант	10	11	12	13	14	15	16	17
Адрес команды, номер итерации	80000024, 1-я	80000028, 1-я	8000002с, 1-я	80000030, 1-я	8000000с, 2-я	80000010, 2-я	80000014, 2-я	8000001с, 2-я
Вариант	19	20	21	22	23	24	25	26
Адрес команды, номер итерации	80000020, 2-я	80000024, 2-я	80000028, 2-я	8000002с, 2-я	80000030, 2-я	80000038, 2-я	См. вариант 1	См. вариант 1

Рисунок 3.1: Таблица вариантов

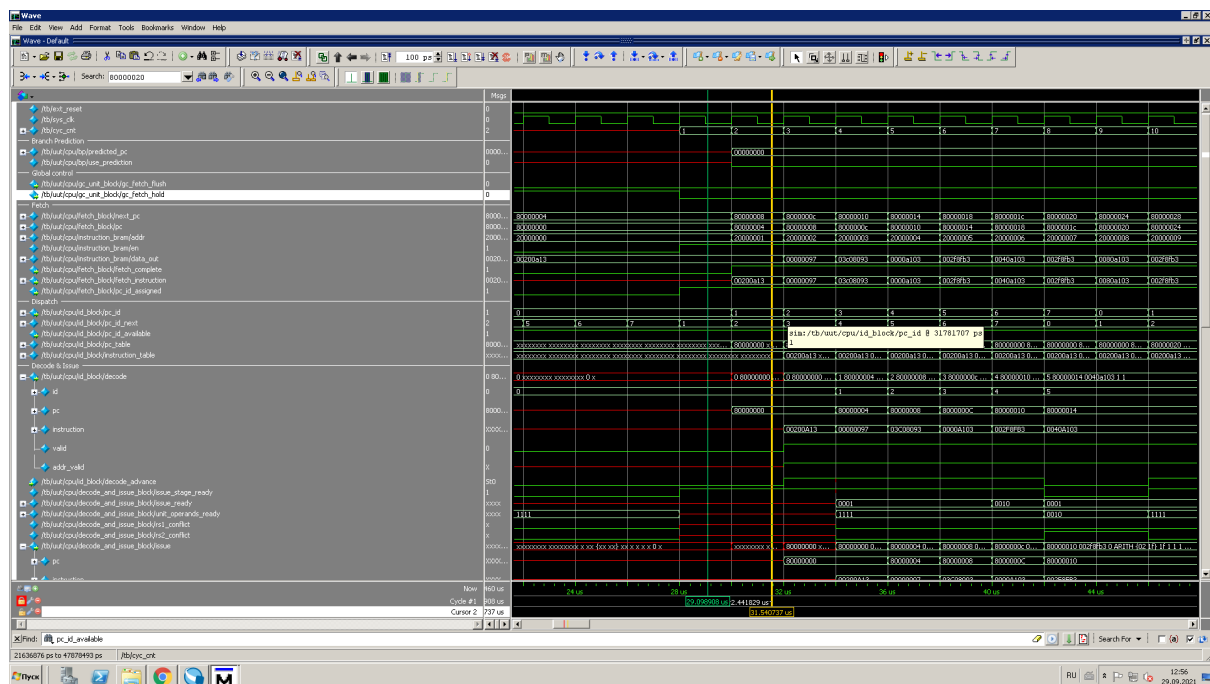


Рисунок 3.2: Снимок экрана со стадиями декодирования по третьему варианту

Задание 5.

В ходе выполнения задания:

- Исправлена 76-я строка файла *taiga/examples/zedboard/taiga_wrapper.sv* так, чтобы там был указан путь к файлу .hex, соответствующему программе по индивидуальному варианту.
- Перекомпилировал исправленный файл.
- Введен в командной строке Modelsim команду `restart; run 460us` для перезапуска симуляции.
- Получена временную диаграмму сигналов выполнения программы третьего варианта (см. Рисунок 5.1 и Рисунок 5.2). `1d=29`

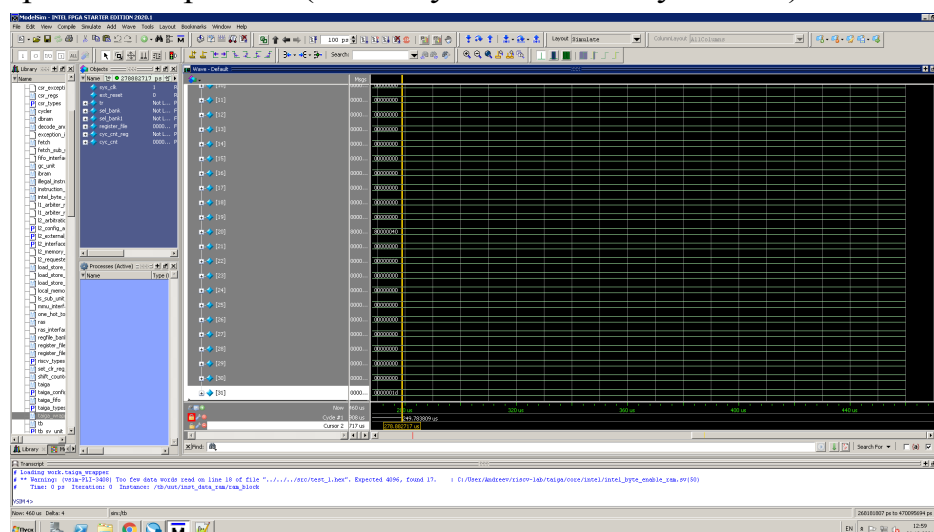


Рисунок 5.1: Временная диаграмма сигналов выполнения программы, Часть 1

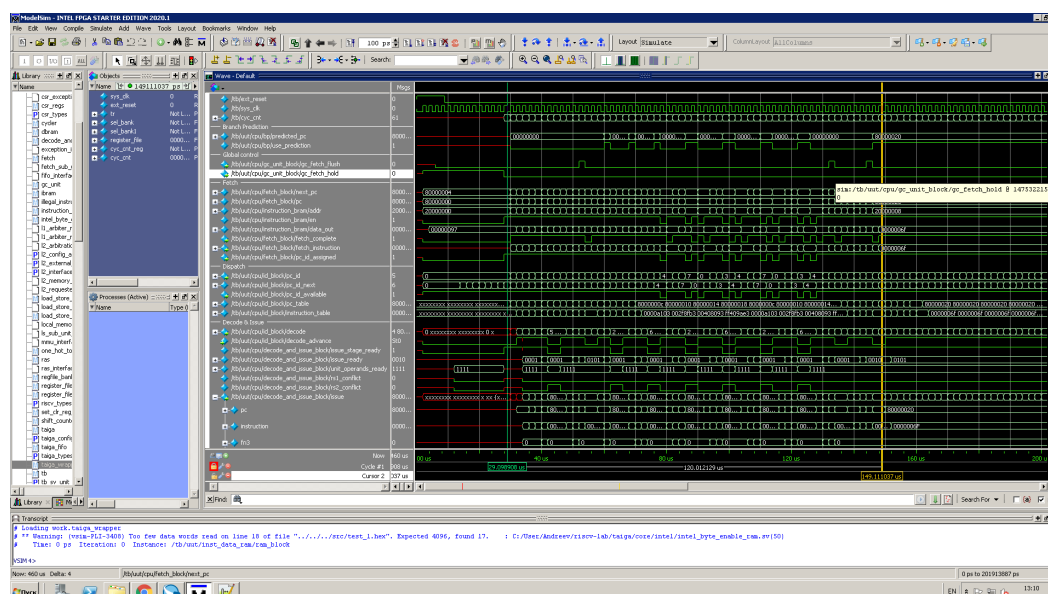


Рисунок 5.2: Временная диаграмма сигналов выполнения программы, Часть 2

- Сравнены значения регистров x31 на момент окончания с 3.1. Получен снимок экрана, содержащий временные диаграммы сигналов, соответствующих в тексте программы #1 (см. Рисунок 5.3, Рисунок 5.4, Рисунок 5.5)

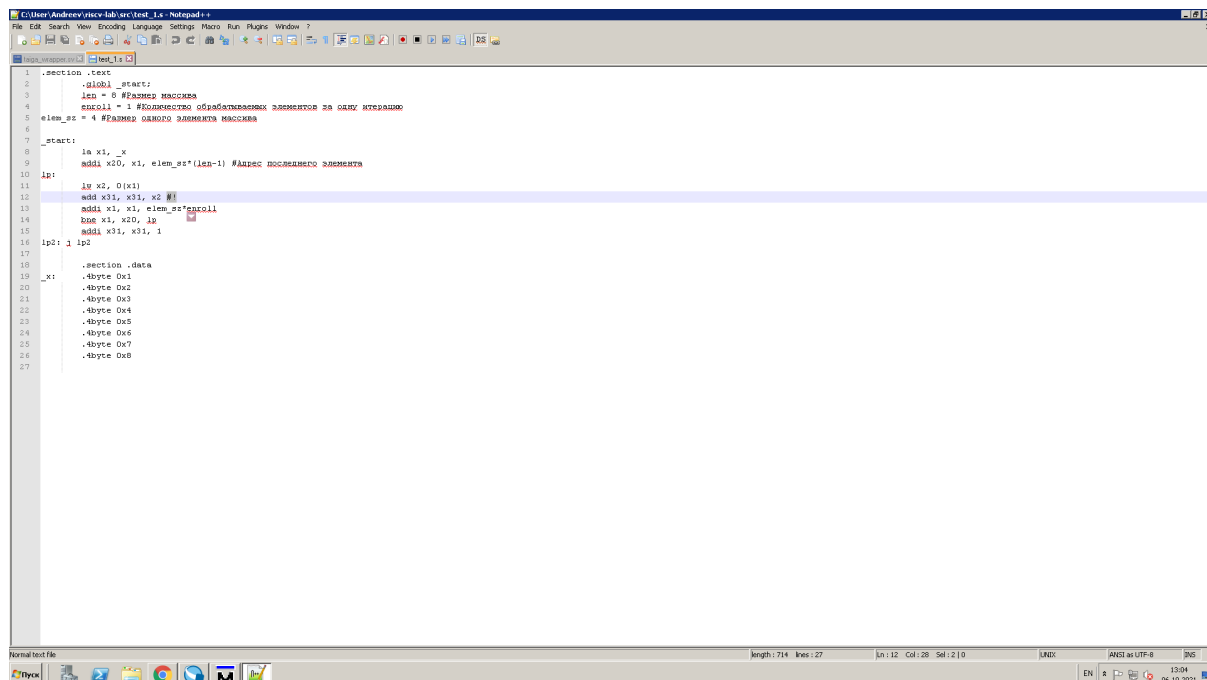


Рисунок 5.3: Отмечены #1

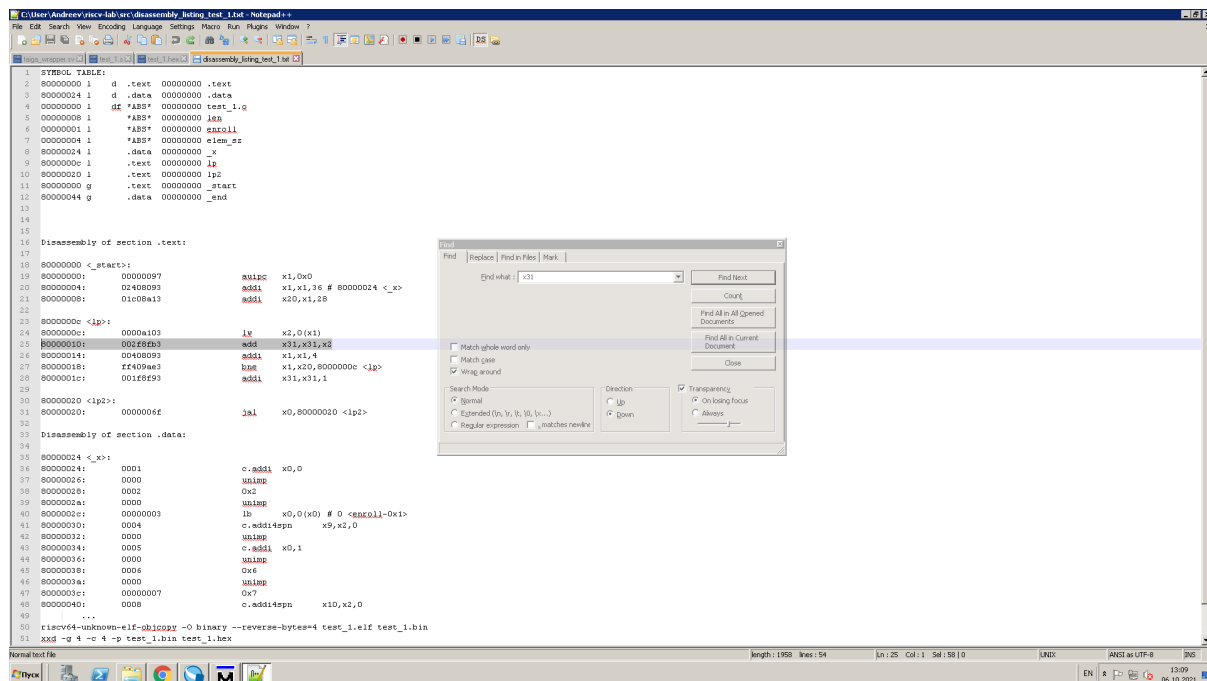


Рисунок 5.4: `add x31, x31, x2, #!`. В дизасемблере.

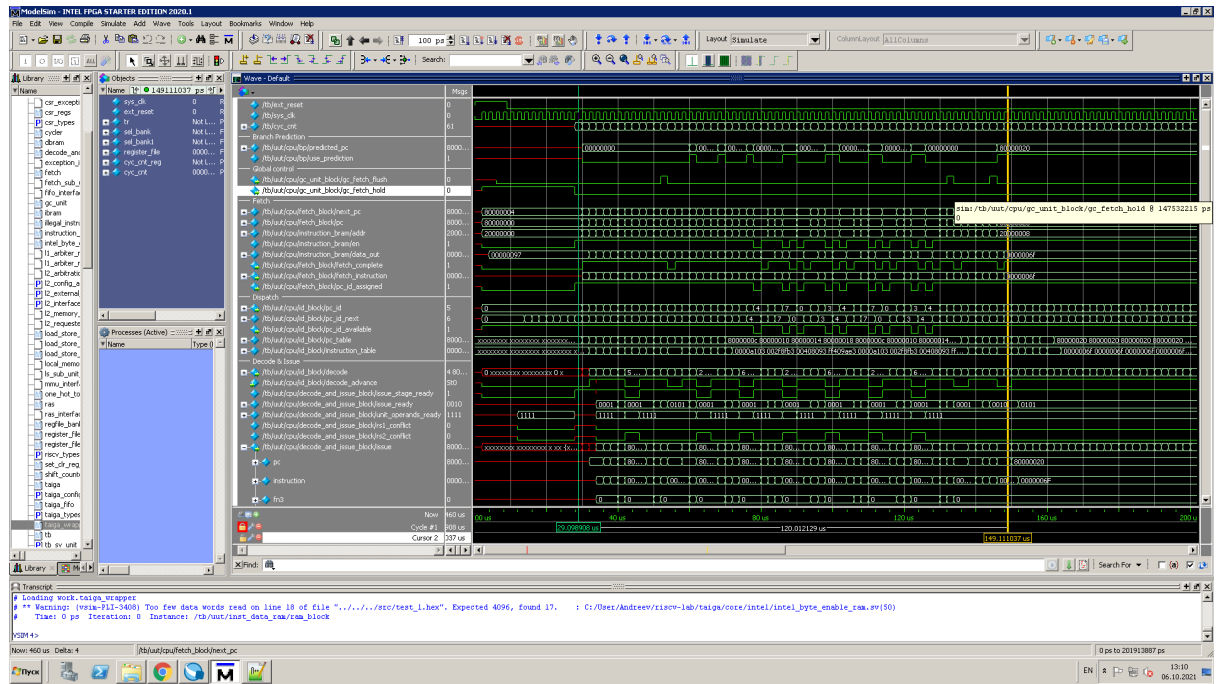


Рисунок 5.5: Временная диаграмма сигналов выполнения программы

- Выполнена трасса программы