



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 «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

по лабораторной работе № 2

по курсу «Архитектура ЭВМ»

на тему: «Изучение принципов работы микропроцессорного ядра RISC-V»

Студент ИУ7-53Б
(Группа)

(Подпись, дата)

Д. В. Недолужко
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

А. Ю. Попов
(И. О. Фамилия)

2022 г.

1 Теоретический раздел

Основной **целью работы** является ознакомление с принципами функционирования, построения и особенностями архитектуры суперскалярных конвейерных микропроцессоров. Дополнительной целью работы является знакомство с принципами проектирования и верификации сложных цифровых устройств с использованием языка описания аппаратуры SystemVerilog и ПЛИС.

2 Практический раздел

2.1 Задание №1

2.1.1 Исходный код

Листинг 2.1 – Исходный код программы

```
.section .text
.globl _start;
len = 8 # array length
enroll = 2 # processed items count for 1 iteration
elem_sz = 4 # item size

_start:
    addi x20, x0, len/enroll
    la x1, _x
lp:
    lw x2, 0(x1)
    add x31, x31, x2
    lw x3, 4(x1)
    add x31, x31, x3 #!
    addi x1, x1, elem_sz*enroll
    addi x20, x20, -1
    bne x20, x0, lp
    addi x31, x31, 1
lp2: j lp2

.section .data
_x:
    .4byte 0x1
    .4byte 0x2
    .4byte 0x3
    .4byte 0x4
    .4byte 0x5
    .4byte 0x6
    .4byte 0x7
    .4byte 0x8
```

2.1.2 Псевдокод

Листинг 2.2 – Псевдокод

```
#define LEN 8
#define ENROLL 2

void
_start()
{
    int sum = 0;
    int i = LEN / ENROLL;
    int arr[] = { 1, 2, 3, 4, 5, 6, 7, 8 };

    do {
        int a = arr[0];
        sum += a;
        int b = arr[1];
        sum += b;
        arr += ENROLL;
        --i;
    } while (i != 0);
    sum += 1;
lp2:
    goto lp2;
}
```

2.1.3 Дизассемблированный листинг

Листинг 2.3 – Дизассемблированный листинг

```
Disassembly of section .text:

80000000 <_start>:
80000000:      00400a13      addi      x20,x0,4
80000004:      00000097      auipc     x1,0x0
80000008:      02c08093      addi      x1,x1,44 #
      80000030 <_x>

8000000c <lp>:
8000000c:      0000a103      lw        x2,0(x1)
80000010:      002f8fb3      add       x31,x31,x2
80000014:      0040a183      lw        x3,4(x1)
80000018:      003f8fb3      add       x31,x31,x3
8000001c:      00808093      addi      x1,x1,8
80000020:      fffa0a13      addi      x20,x20,-1
80000024:      fe0a14e3      bne       x20,x0,8000000c <
      lp>
80000028:      001f8f93      addi      x31,x31,1

8000002c <lp2>:
8000002c:      0000006f      jal       x0,8000002c <lp2>

Disassembly of section .data:

80000030 <_x>:
80000030:      0001          c.addi     x0,0
80000032:      0000          unimp
80000034:      0002          0x2
80000036:      0000          unimp
80000038:      00000003      lb        x0,0(x0) # 0 <
      enroll -0x2>
8000003c:      0004          c.addi4spn      x9,x2,0
8000003e:      0000          unimp
80000040:      0005          c.addi     x0,1
80000042:      0000          unimp
80000044:      0006          0x6
80000046:      0000          unimp
80000048:      00000007      0x7
8000004c:      0008          c.addi4spn      x10,x2,0
```

2.2 Задание №2

Получить снимок экрана, содержащий временную диаграмму выполнения стадий выборки и диспетчеризации команды с указанным адресом. 80000014, 2-я.

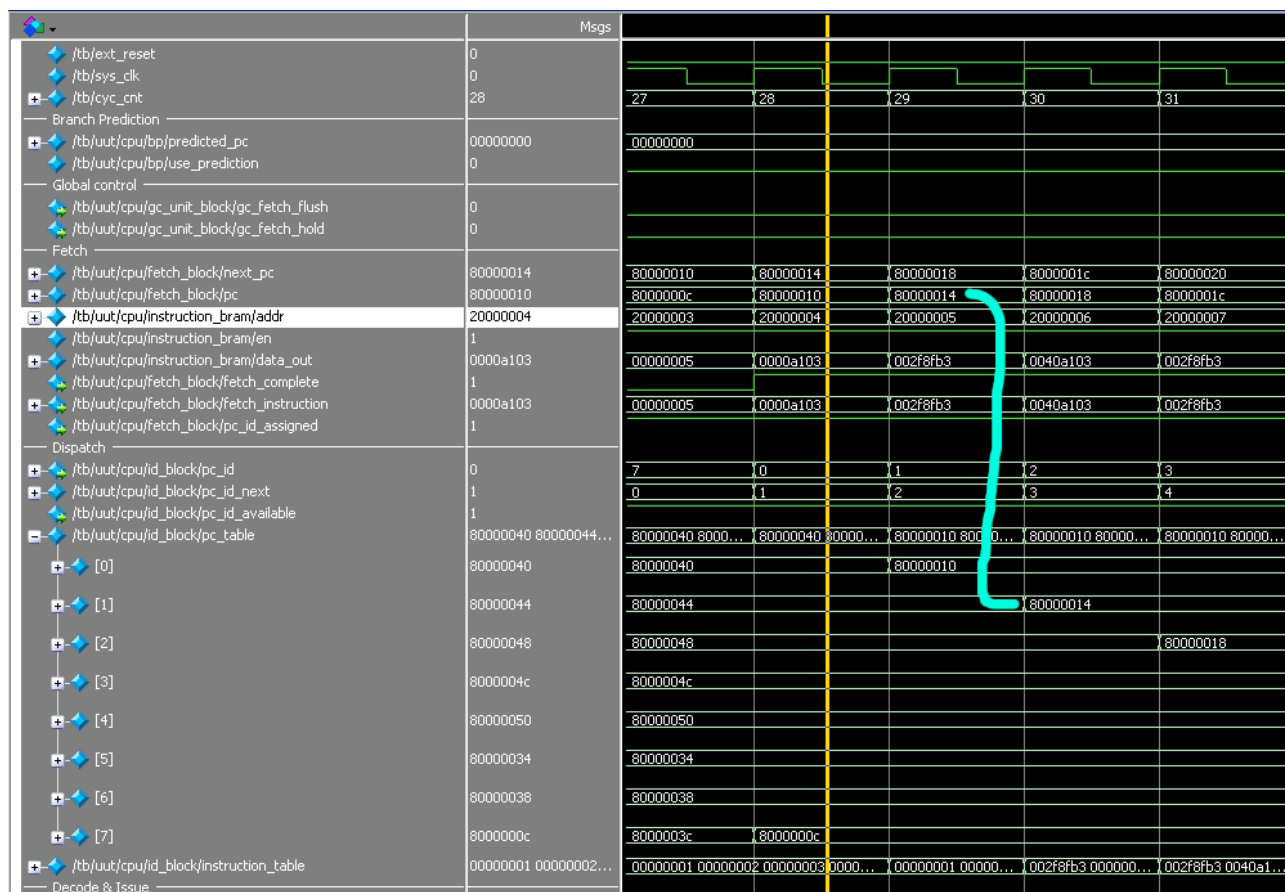


Рисунок 2.1 – Стадии выборки и диспетчеризации команды по 80000014 адреса на 2ой итерации

- Такт 29 - происходит выборка команды с адресом 80000014
- Такт 30 - команда попадает на вход блока управления метаданной; записывается в `pc_table` с `pc_id = 1`.

2.3 Задание №3

Получить снимок экрана, содержащий временную диаграмму выполнения стадии декодирования и планирования на выполнение команды с указанным адресом. 80000020, 2-я.

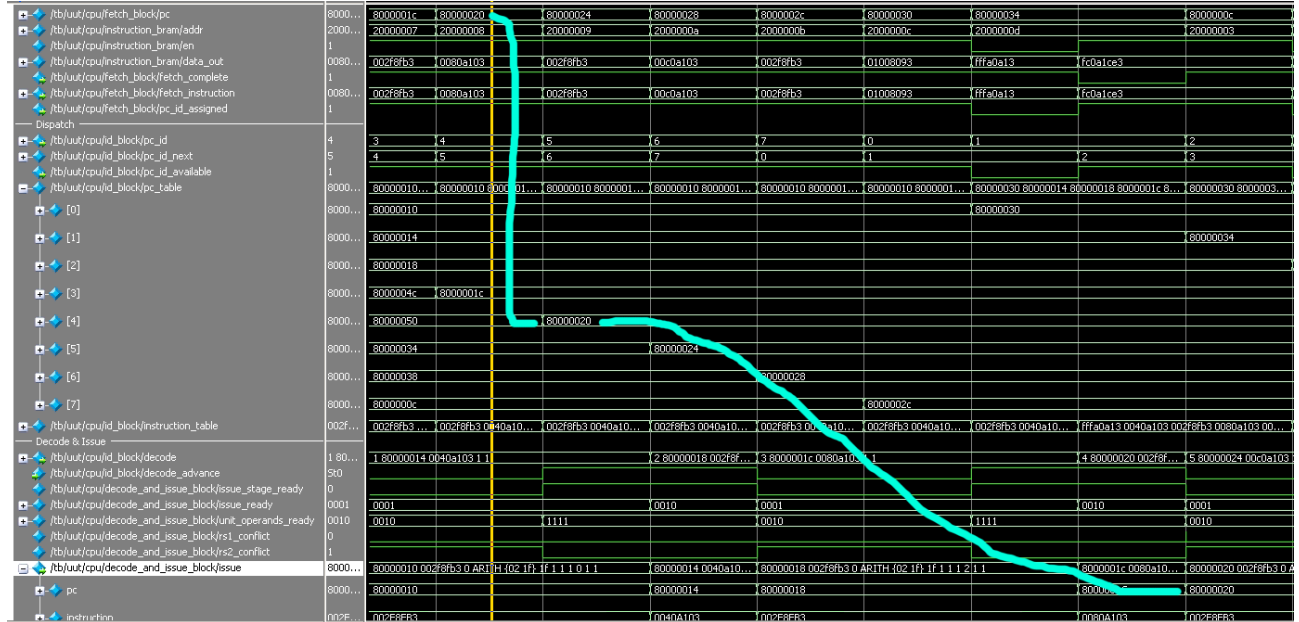


Рисунок 2.2 – Стадии декодирования и планирования на выполнение команды 80000020 на 2ой итерации

- Такт 32 - выборка команды с адресом 80000020
- Такт 33 - диспечеризация. Команда попадает на вход блока управления метаинформацией; записывается в *pc_table* с *pc_id* = 4.
- Такт 34-38 - ожидание
- Такт 39 - планирование на выполнение

Затем 2 такта пребывает в конфликте *rs2_conflict*.

2.4 Задание №4

Получить снимок экрана, содержащий временную диаграмму выполнения стадии выполнения команды с указанным адресом. 8000000с, 2-я. (lw x2,0(x1))

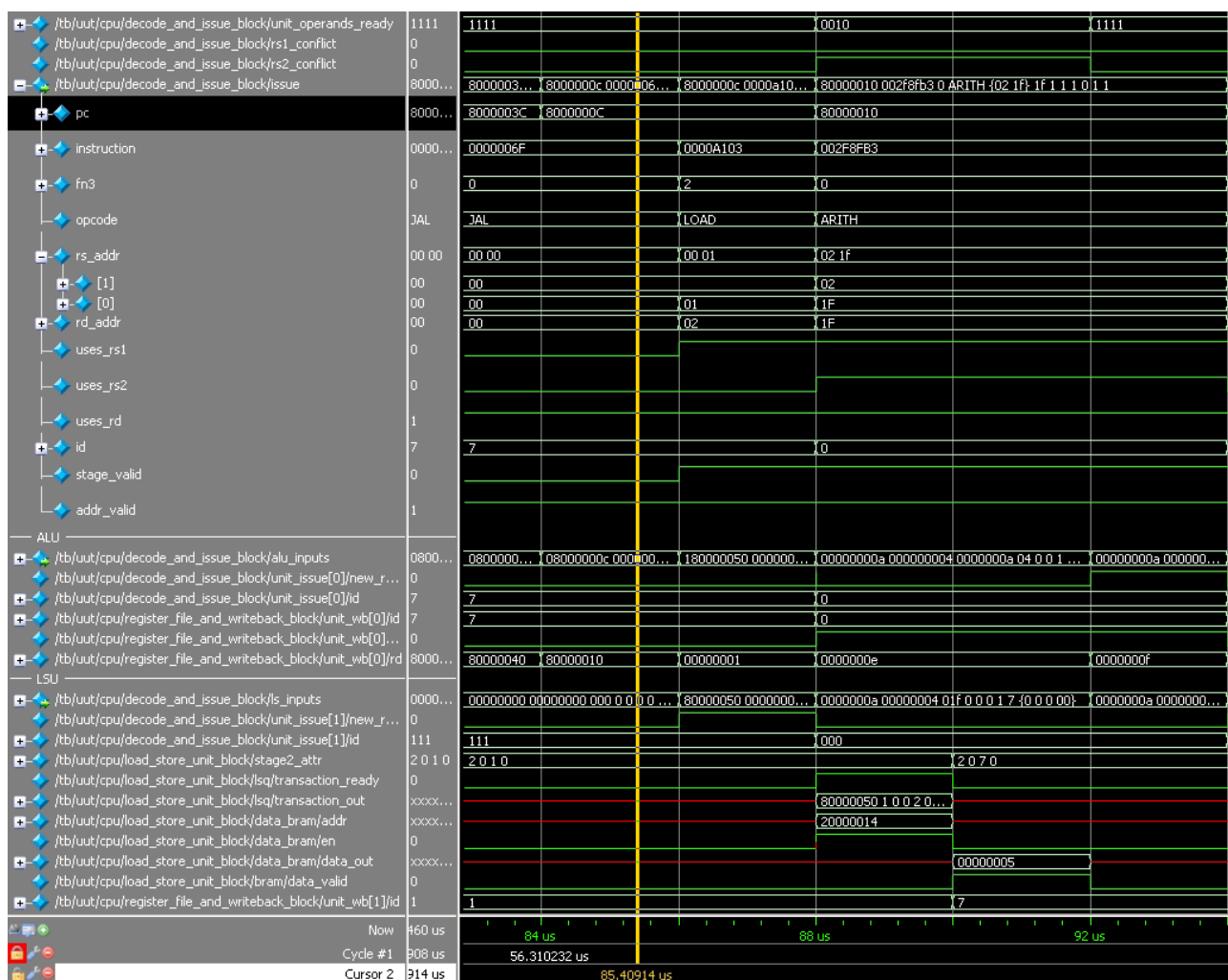


Рисунок 2.3 – Снимок экрана, содержащий временную диаграмму стадии выполнения команды по адресу 8000000с на второй итерации. (lw x2,0(x1))

2.5 Задание №5

Адрес команды: 80000018

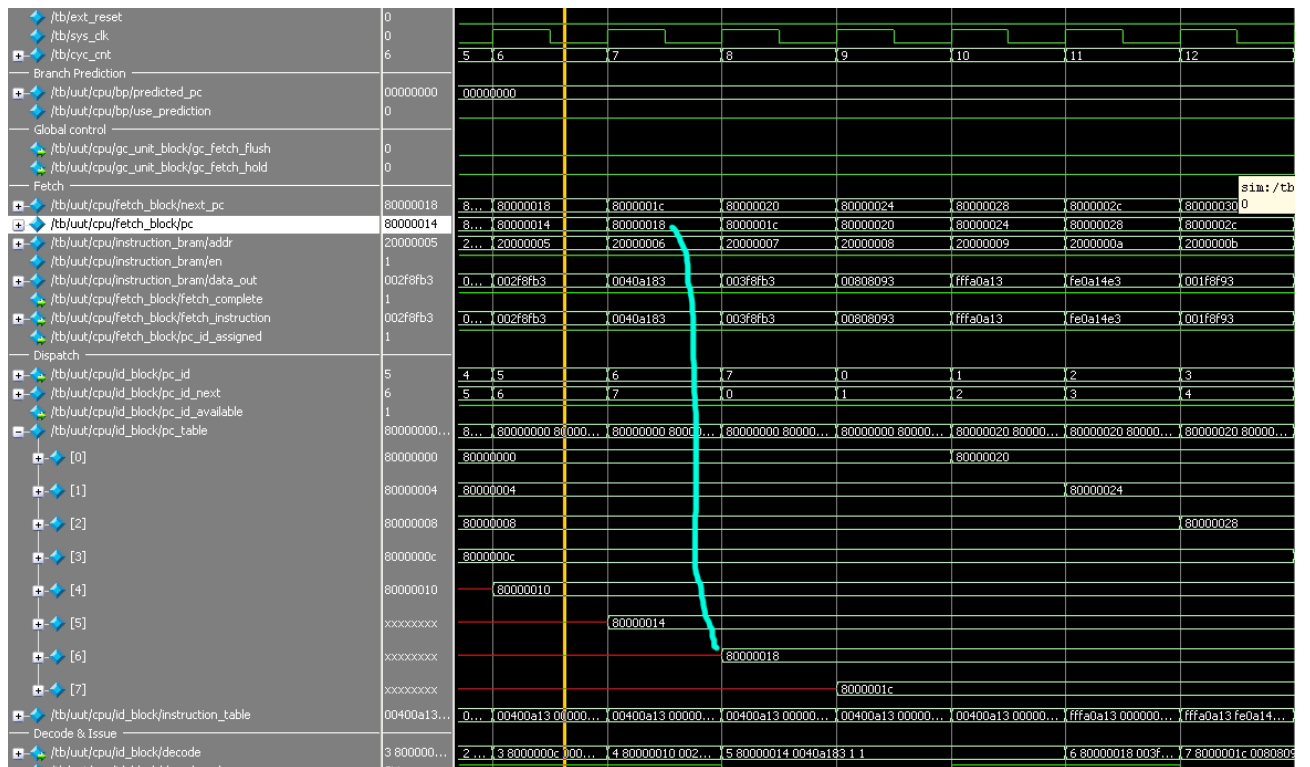


Рисунок 2.4 – Выборка, диспечеризация

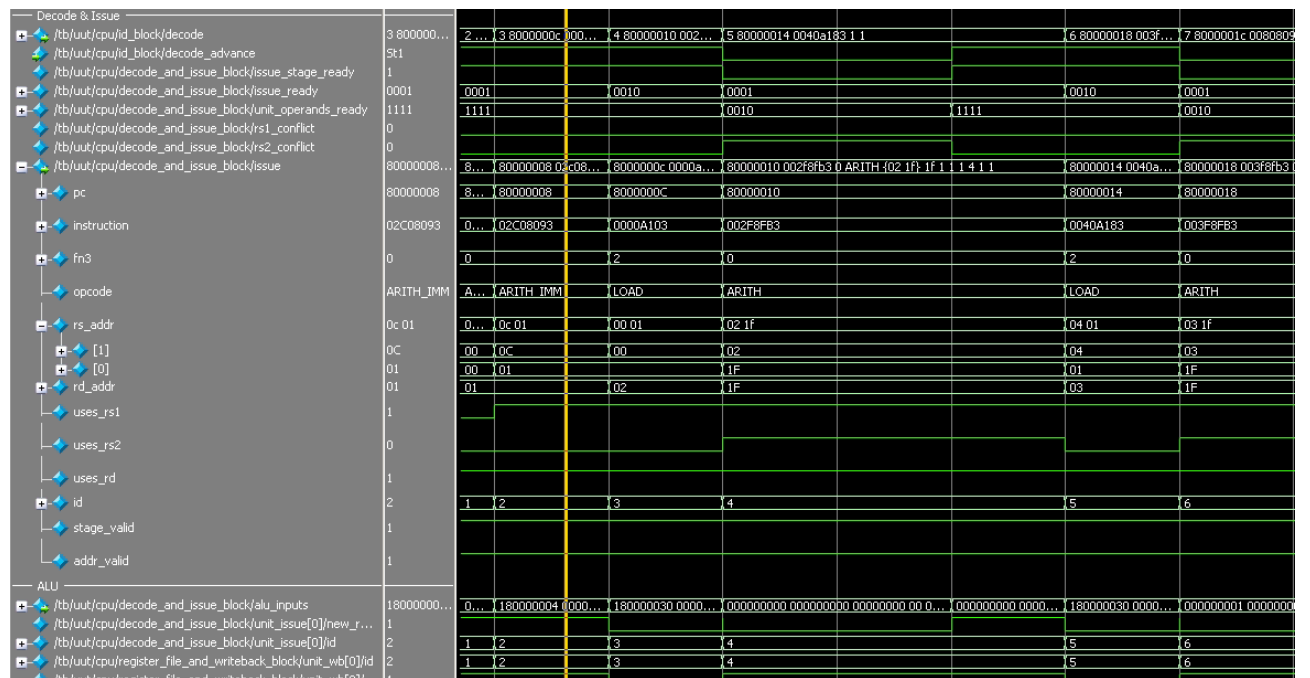


Рисунок 2.5 – Планирование, выполнение

2.5.1 Оптимизация

Можно оптимизировать этот участок:

```
lw x2, 0(x1)
add x31, x31, x2
lw x3, 4(x1)
add x31, x31, x3
```

Каждая следующая команда `add` зависит от предыдущей команды загрузки, из-за чего они простаивают ожидая разрешение конфликта. Это можно оптимизировать путем совместного запроса данных:

```
lw x2, 0(x1)
lw x3, 4(x1)
add x31, x31, x2
add x31, x31, x3
```

[illegible]

Рисунок 2.6 – Оригинальная программа

