



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 № 1 по курсу «Архитектура ЭВМ»

Тема Изучение принципов работы микропроцессорного ядра RISC-V

Студент Красильникова А. И.

Группа ИУ7-56Б

Оценка (баллы)

Преподаватели Попов А.Ю.

Содержание

1	Цели	3
2	Задания	4
2.1	Задание 1	4
2.2	Задание 2	8
2.3	Задание 3	9
2.4	Задание 4	10
2.5	Задание 5	11
2.6	Заключение	14

1 Цели

- 1) Ознакомиться с принципами функционирования построения и особенностями архитектуры суперскалярных конвейерных микропроцессоров;
- 2) Ознакомиться с принципами проектирования и верификации сложных цифровых устройств с использованием языка описания аппаратуры *SystemVerilog* и ПЛИС.

2 Задания

2.1 Задание 1

Содержимое файла *task.s*, который соответствует **варианту 15**:

```
1      .section .text
2      .globl _start;
3      len = 9 #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, x0, (len-1)/enroll
10         lw x31, 0(x1)
11         addi x1, x1, elem_sz*1
12 lp:
13         lw x2, 0(x1)
14         bltu x2, x31, lt
15         add x31, x0, x2 #!
16 lt:
17         add x1, x1, elem_sz*enroll
18         addi x20, x20, -1
19         bne x20, x0, lp
20 lp2: j lp2
21
22     .section .data
23     _x:      .4 byte 0x1
24             .4 byte 0x2
25             .4 byte 0x3
26             .4 byte 0x4
27             .4 byte 0x5
28             .4 byte 0x6
29             .4 byte 0x7
30             .4 byte 0x8
31             .4 byte 0x9
```

Псевдокод:

```
1
```

```

2  #define len 9
3  #define enroll 1
4  #define elem_sz 4
5
6  int _x[]={1,2,3,4,5,6,7,8,9};
7
8  void _start() {
9      int x20 = (len-1)/enroll;
10     int *x1 = _x;
11     int x31 = x1[0];
12     x1++;
13
14     do {
15         int x2 = x1[0];
16         if (x2 >= x31)
17             x31 = x2; // #
18
19         x1++;
20         x20--;
21     } while(x20 != 0);
22
23     while(1){}
24 }

```

Результат выполнения компиляции:

```

1  SYMBOL TABLE:
2  80000000 | d .text 00000000 .text
3  80000030 | d .data 00000000 .data
4  00000000 | df *ABS* 00000000 task.o
5  00000009 | *ABS* 00000000 len
6  00000001 | *ABS* 00000000 enroll
7  00000004 | *ABS* 00000000 elem_sz
8  80000030 | .data 00000000 _x
9  80000014 | .text 00000000 lp
10 80000020 | .text 00000000 lt
11 8000002c | .text 00000000 lp2
12 80000000 g .text 00000000 _start
13 80000054 g .data 00000000 _end

```

Дизассемблирование раздела .text:

```

18
19      80000000 <_start>:
20      80000000: 00000097      auipc x1,0x0
21      80000004: 03008093      addi x1,x1,48 # 80000030 <_x>
22      80000008: 00800a13      addi x20,x0,8
23      8000000c: 0000af83      lw x31,0(x1)
24      80000010: 00408093      addi x1,x1,4
25
26      80000014 <lp>:
27      80000014: 0000a103      lw x2,0(x1)
28      80000018: 01f16463      bltu x2,x31,80000020 <lt>
29      8000001c: 00200fb3      add x31,x0,x2
30
31      80000020 <lt>:
32      80000020: 00408093      addi x1,x1,4
33      80000024: fffa0a13      addi x20,x20,-1
34      80000028: fe0a16e3      bne x20,x0,80000014 <lp>
35
36      8000002c <lp2>:
37      8000002c: 0000006f      jal x0,8000002c <lp2>
38
39      Дизассемблирование раздела .data:
40
41      80000030 <_x>:
42      80000030: 0001      c.addi x0,0
43      80000032: 0000      c.unimp
44      80000034: 0002      c.slli64 x0
45      80000036: 0000      c.unimp
46      80000038: 00000003      lb x0,0(x0) # 0 <enroll-0x1>
47      8000003c: 0004      0x4
48      8000003e: 0000      c.unimp
49      80000040: 0005      c.addi x0,1
50      80000042: 0000      c.unimp
51      80000044: 0006      c.slli x0,0x1
52      80000046: 0000      c.unimp
53      80000048: 00000007      0x7
54      8000004c: 0008      0x8
55      8000004e: 0000      c.unimp
56      80000050: 0009      c.addi x0,2

```

Содержимое файла *task.hex*:

```

1      00000097

```

2	03008093
3	00800a13
4	0000af83
5	00408093
6	0000a103
7	01f16463
8	00200fb3
9	00408093
10	fffa0a13
11	fe0a16e3
12	0000006f
13	00000001
14	00000002
15	00000003
16	00000004
17	00000005
18	00000006
19	00000007
20	00000008
21	00000009

В регистре x31 в конце выполнения программы должно содержаться максимальное значение массива. В нашем случае: **9**.

2.2 Задание 2

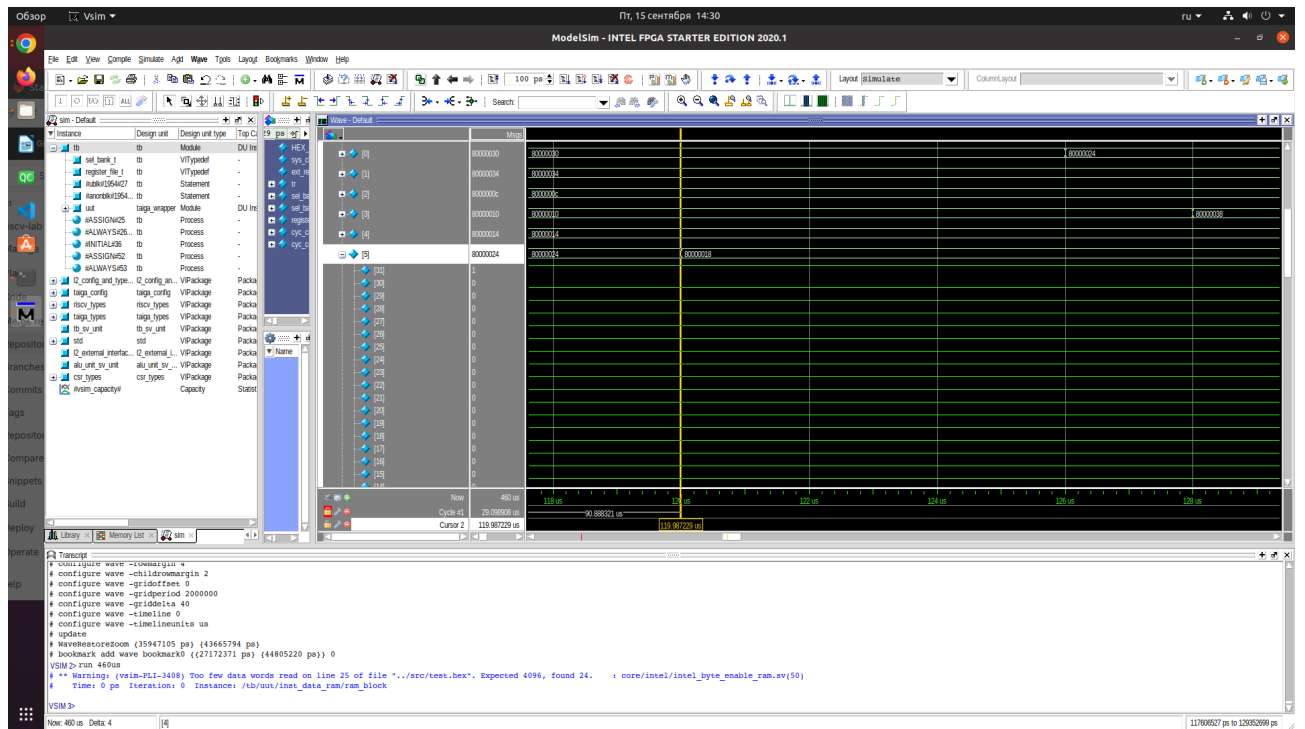


Рисунок 2.1 – Временная диаграмма

2.3 Задание 3

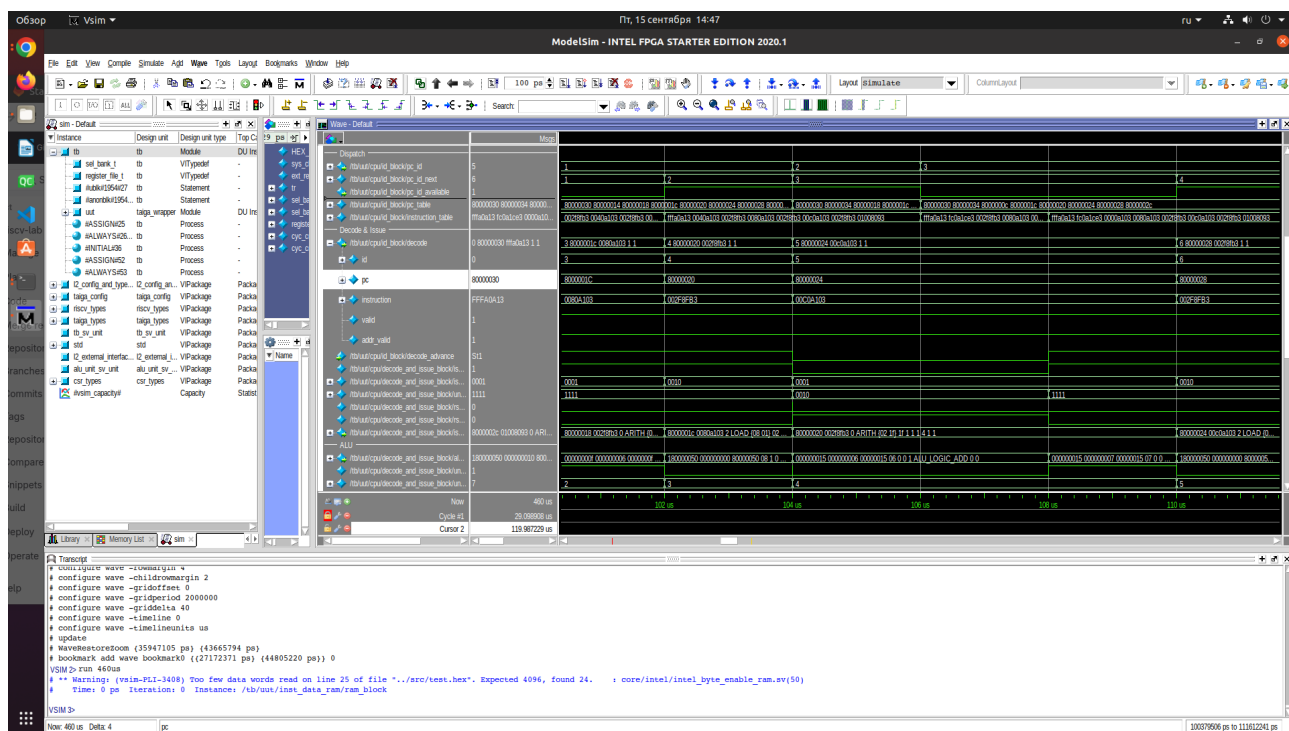


Рисунок 2.2 – Временная диаграмма

2.4 Задание 4

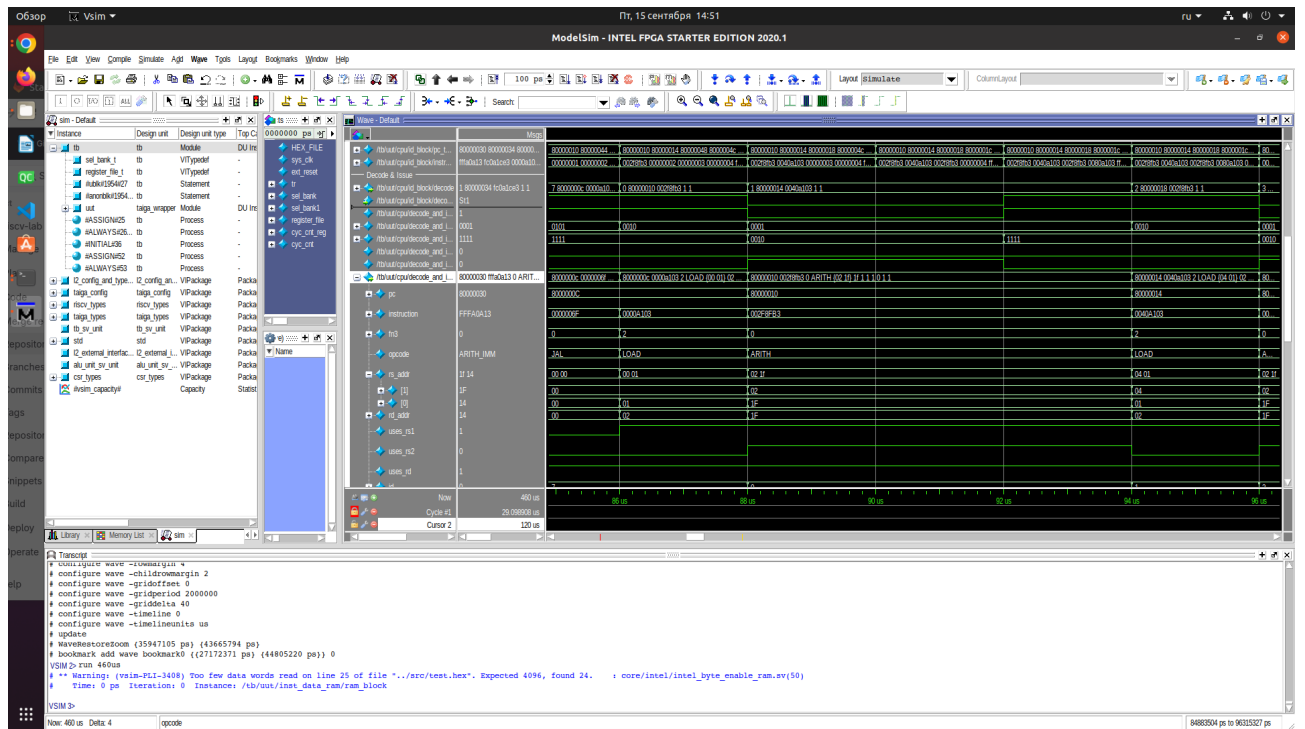
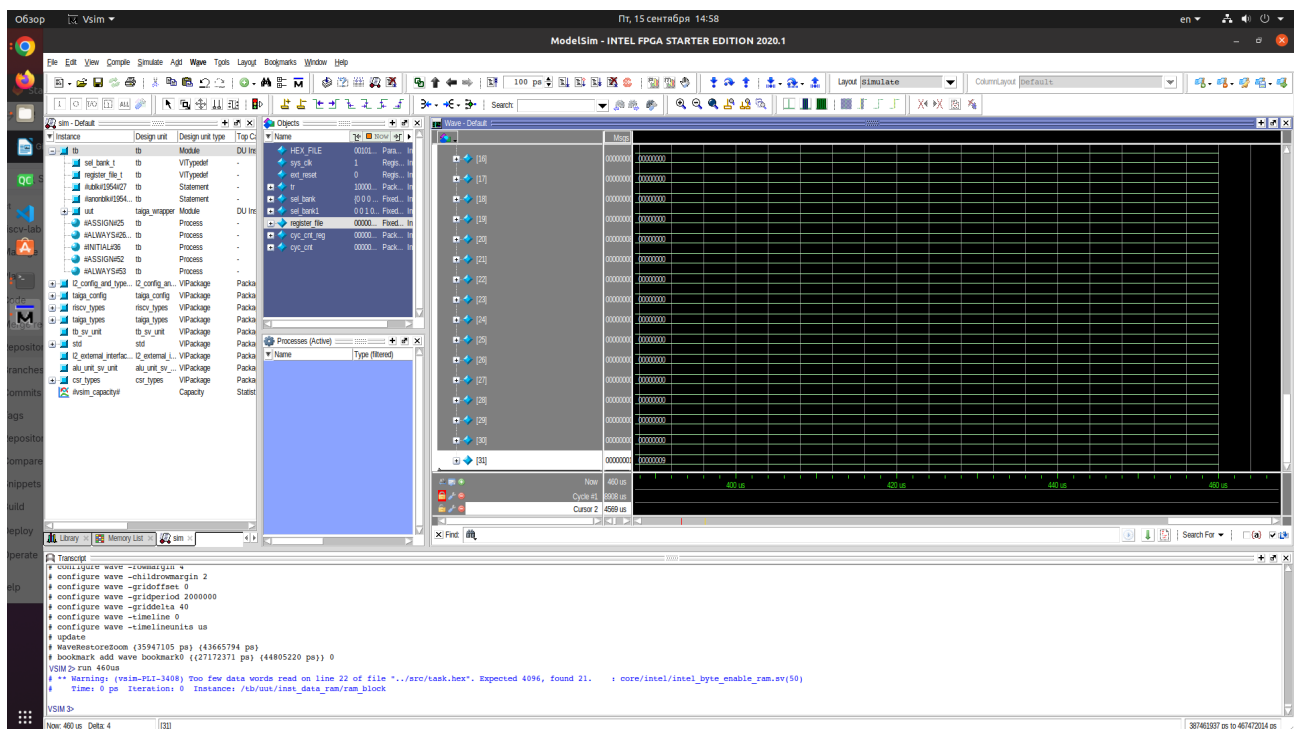
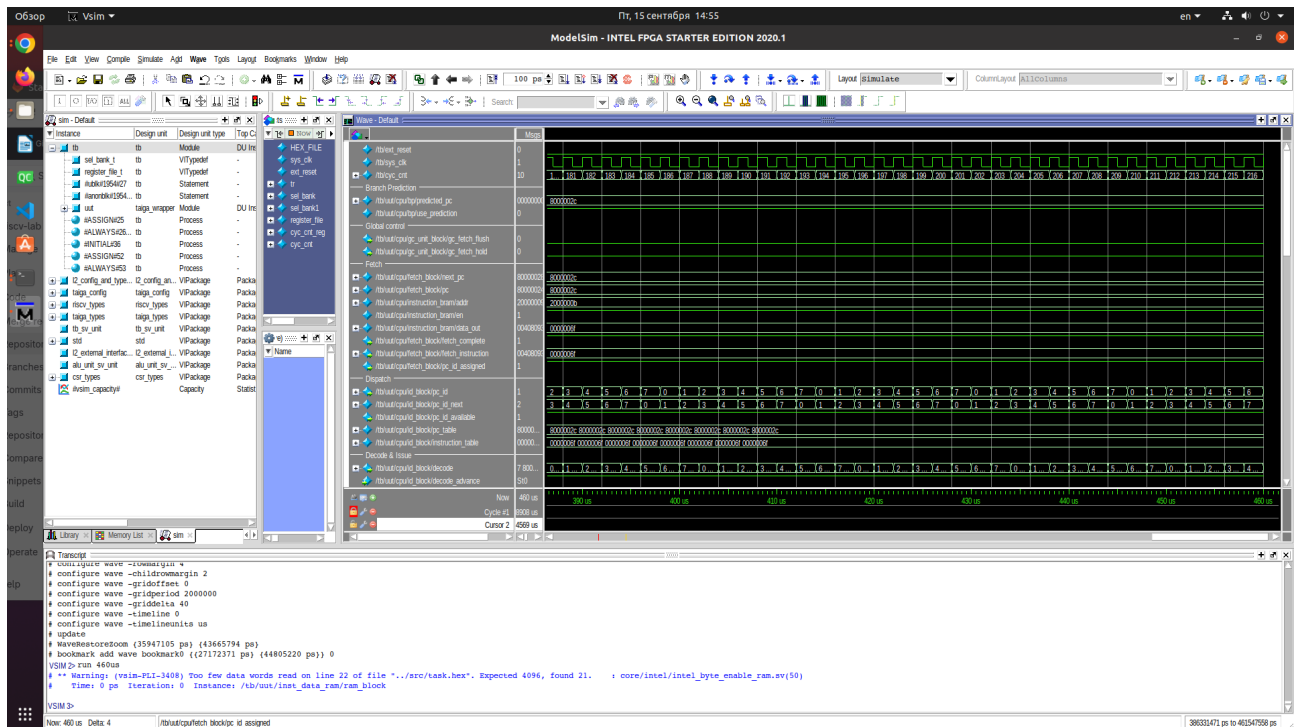


Рисунок 2.3 – Временная диаграмма

2.5 Задание 5



Заметим, что значение из регистра $x31$ совпадает с вычисленным ранее значением.


```

18
19         bltu x2, x31, lt
20         add x31, x0, x2 #!
21 lt:
22         bne x20, x0, lp
23 lp2: j lp2
24
25         .section .data
26 _x:     .4 byte 0x1
27         .4 byte 0x2
28         .4 byte 0x3
29         .4 byte 0x4
30         .4 byte 0x5
31         .4 byte 0x6
32         .4 byte 0x7
33         .4 byte 0x8
34         .4 byte 0x9

```

Трасса работы оптимизированной программы:

[illegible]

Рисунок 2.7 – Трасса работы оптимизированной программы

2.6 Заключение

В результате выполнения задания были изучены основные принципы работы, структура и особенности архитектуры суперскалярных конвейерных микропроцессоров.

Используя полученные знания, был найден метод оптимизации программы.