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 дисциплине «Архитектура ЭВМ»

Тема Изучение принципов работы микропроцессорного ядра RISC-V

Студент Жаворонкова А. А.

Группа ИУ7-56Б

Оценка (баллы)

Преподаватель Попов А. Ю.

Москва — 2023 г.

Содержание

Введение

Цель работы: 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.

Для достижения поставленных целей в настоящей лабораторной работе используется синтезируемое описание микропроцессорного ядра Taiga[1], реализующего систему команд RV32I семейства RISC-V. Данное описание выполнено на языке описания аппаратуры SystemVerilog.

В ходе лабораторной работы используется средство моделирования Modelsim для моделирования работы исследуемого микропроцессора в процессе выполнения программы и наблюдения формы внутренних сигналов.

Все задания выполняются в соответствии с вариантом №10.

1 Задание №1

1.1 Текст программы по индивидуальному варианту

Листинг 1 – Текст программы по индивидуальному варианту

```
1         .section .text
2         .globl _start;
3         len = 8
4         enroll = 4
5         elem_sz = 4
6
7 _start:
8         addi x20, x0, len/enroll
9         la x1, _x
```

[1]https://gitlab.com/sfu-rcl/Taiga, авторы - Eric Matthews, Lesley Shannon

```
10          add x31, x0, x0
11  lp:
12          lw x2, 0(x1)
13          lw x3, 4(x1) #!
14          add x31, x31, x2
15          add x31, x31, x3
16          lw x4, 8(x1)
17          lw x5, 12(x1)
18          add x31, x31, x4
19          add x31, x31, x5
20          addi x1, x1, elem_sz*enroll
21          addi x20, x20, -1
22          bne x20, x0, lp
23          addi x31, x31, 1
24  lp2: j lp2
25
26          .section .data
27  _x:     .4byte 0x1
28          .4byte 0x2
29          .4byte 0x3
30          .4byte 0x4
31          .4byte 0x5
32          .4byte 0x6
33          .4byte 0x7
34          .4byte 0x8
```

1.2 Дизассемблерный листинг кода программы

В результате выполнения компиляции был создан файл с расширением .hex, хранящий содержимое памяти команд и данных. В окне терминала отобразился дизассемблерный листинг, который приведен в листинге 2.

Листинг 2 – Дизассемблированная программа по варианту

```
1     SYMBOL TABLE:
2     80000000 l    d  .text  00000000 .text
3     80000044 l    d  .data  00000000 .data
```

```
00000000 l    df *ABS*  00000000 myprog.o
00000008 l       *ABS*  00000000 len
00000004 l       *ABS*  00000000 enroll
00000004 l       *ABS*  00000000 elem_sz
80000044 l       .data  00000000 _x
80000010 l       .text  00000000 lp
80000040 l       .text  00000000 lp2
80000000 g       .text  00000000 _start
80000064 g       .data  00000000 _end

Дизассемблирование раздела .text:

80000000 <_start>:
80000000:   00200a13            addi    x20,x0,2
80000004:   00000097            auipc   x1,0x0
80000008:   04008093            addi    x1,x1,64 # 80000044 <_x>
8000000c:   00000fb3            add x31,x0,x0

80000010 <lp>:
80000010:   0000a103            lw  x2,0(x1)
80000014:   0040a183            lw  x3,4(x1)
80000018:   002f8fb3            add x31,x31,x2
8000001c:   003f8fb3            add x31,x31,x3
80000020:   0080a203            lw  x4,8(x1)
80000024:   00c0a283            lw  x5,12(x1)
80000028:   004f8fb3            add x31,x31,x4
8000002c:   005f8fb3            add x31,x31,x5
80000030:   01008093            addi    x1,x1,16
80000034:   fffa0a13            addi    x20,x20,-1
80000038:   fc0a1ce3            bne x20,x0,80000010 <lp>
8000003c:   001f8f93            addi    x31,x31,1

80000040 <lp2>:
80000040:   0000006f            jal x0,80000040 <lp2>

Дизассемблирование раздела .data:

80000044 <_x>:
80000044:   0001                    c.addi  x0,0
```

```
80000046:   0000                    c.unimp
80000048:   0002                    c.slli64    x0
8000004a:   0000                    c.unimp
8000004c:   00000003            lb  x0,0(x0) # 0 <elem_sz-0x4>
80000050:   0004                    .2byte  0x4
80000052:   0000                    c.unimp
80000054:   0005                    c.addi  x0,1
80000056:   0000                    c.unimp
80000058:   0006                    c.slli  x0,0x1
8000005a:   0000                    c.unimp
8000005c:   00000007            .4byte  0x7
80000060:   0008                    .2byte  0x8
...
```

1.3 Псевдокод, поясняющий работу программы

На листинге 3 приведен псевдокод, поясняющий работу программы.

Листинг 3 – Псевдокод поясняющий работу программы

```
#define len 8
#define enroll 4
#define elem_sz 4
int _x[] = {1, 2, 3, 4, 5, 6, 7, 8};

void start(){
    int x20 = len/enroll;
    int *x1 = _x;
    int x_31 = 0;

    do
    {
        int x2 = x1[0];
        int x3 = x1[1];
        x31 += x2;
        x31 += x3;

        int x4 = x1[2];
```

```
19              int x5 = x1[3];
20              x31 += x4;
21              x31 += x5;
22
23              x1 += enroll;
24              x20--;
25          } while (x20 != x0)
26
27          x31 += 1;
28
29          while(1) {}
30      }
```

Очевидно, что в регистре x31 после выполнения программы должно содержаться значение:

$$x31 = \sum_{i=1}^{8} i + 1 = 37.$$

2 Задание №2

В результате симуляции, был получен снимок экрана, содержащий временную диаграмму выполнения стадий выборки и диспетчеризации команды с адресом 80000030 (1 итерация). Снимок экрана приведен на рисунке 1.

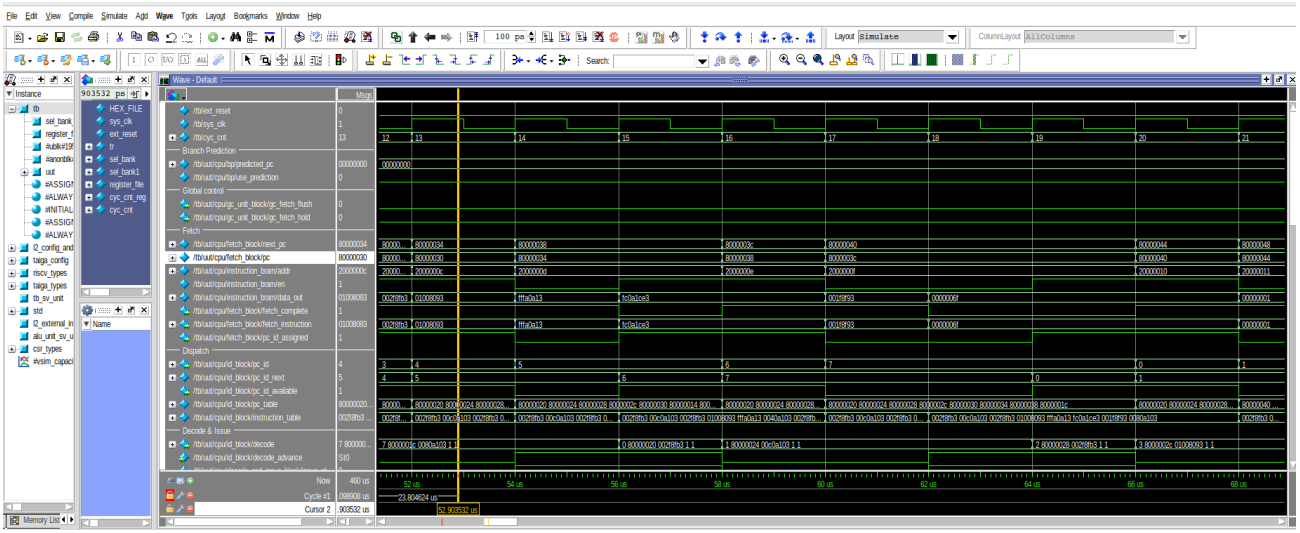

Рисунок 1 – Временная диаграмма выполнения стадий выборки и диспетчеризации команды с адресом 80000030 (1 итерация)

3 Задание №3

В результате симуляции, был получен снимок экрана, содержащий временную диаграмму выполнения стадии декодирования и планирования на выполнение команды с адресом 80000010 (2 итерация). Снимок экрана приведен на рисунке 2.

4 Задание №4

В результате симуляции, был получен снимок экрана, содержащий временную диаграмму стадии выполнения команды с адресом 80000024 (1 итерация). Снимок экрана приведен на рисунке 3.

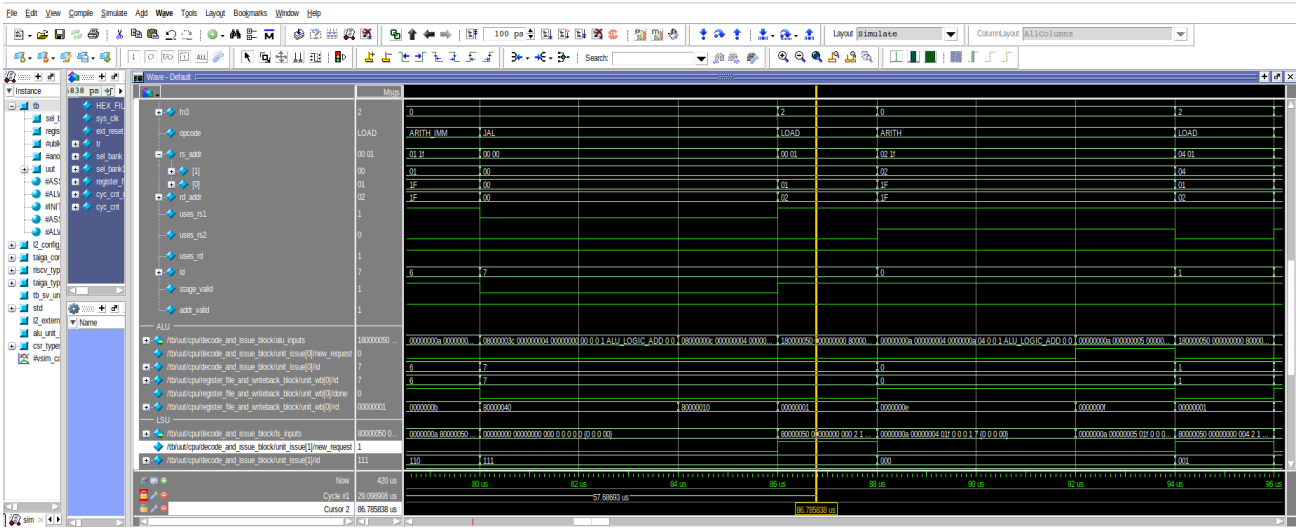

Рисунок 2 – Временная диаграмма выполнения стадий декодирования и планитрования на выполнение команды с адресом 80000010 (2 итерация)

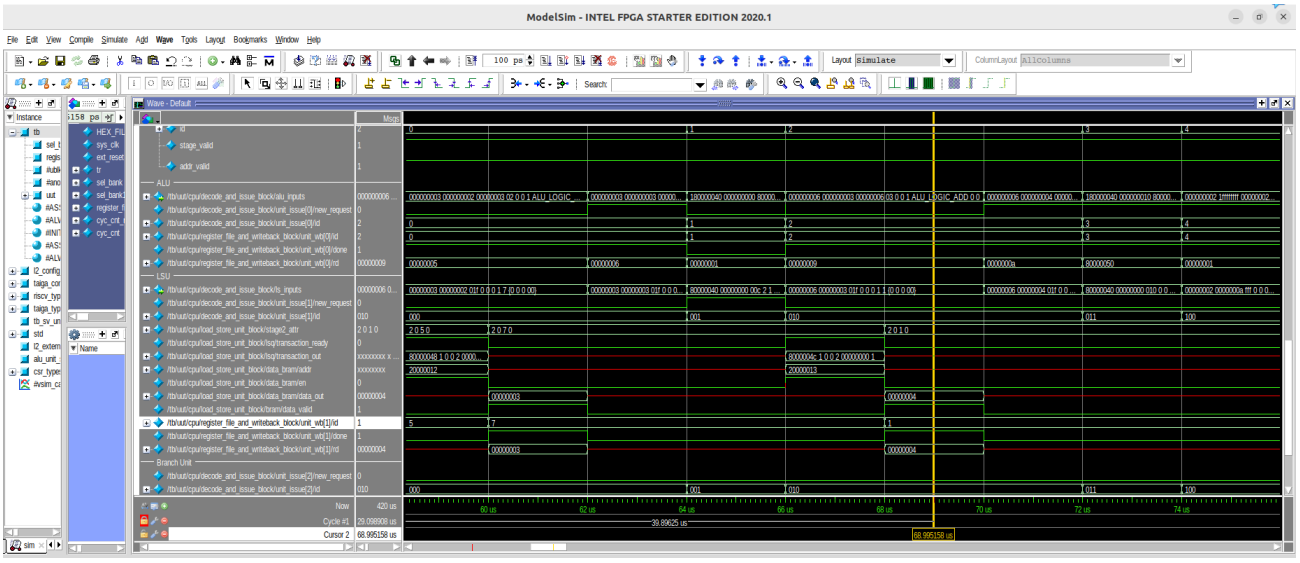

Рисунок 3 – Временная диаграмма стадии выполнения команды с адресом 80000024 (1 итерация)

5 Задание №5

Значение регистра $x31$ в конце выполнения программы равно $25h = 37$, как и предполагалось ранее. Скриншот представлен на рисунке 4.

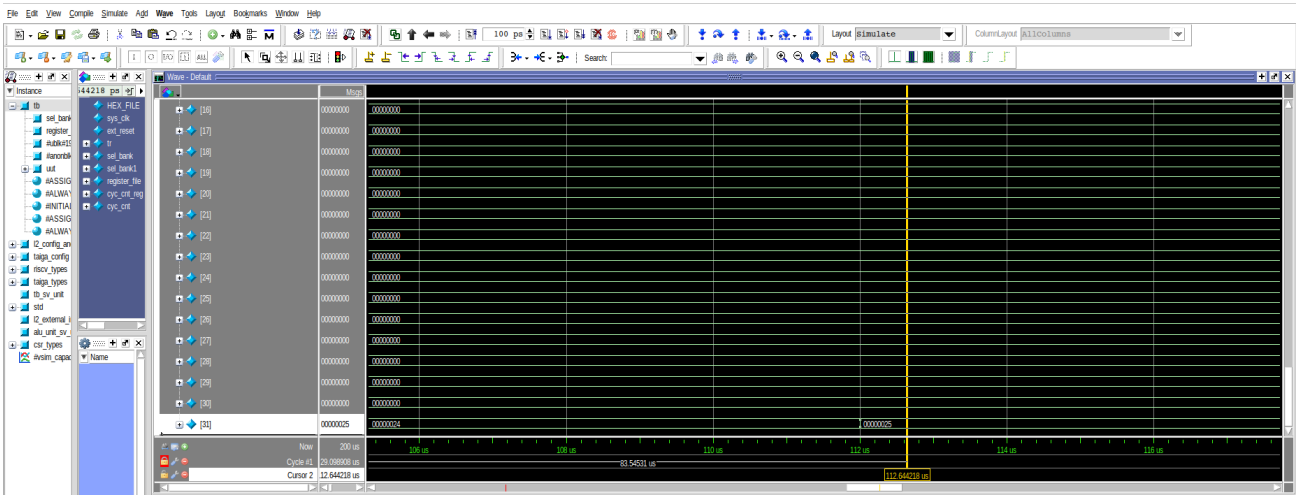

Рисунок 4 – Значение регистра x31 в конце выполнения программы

На рисунках 5 — 7 представлены временные диаграммы сигналов, соответствующих всем стадиям выполнения команды, обозначенной в тексте программы символом #!: команда lw x3,4(x1), имеющая код 0040a183 и адрес 80000014.

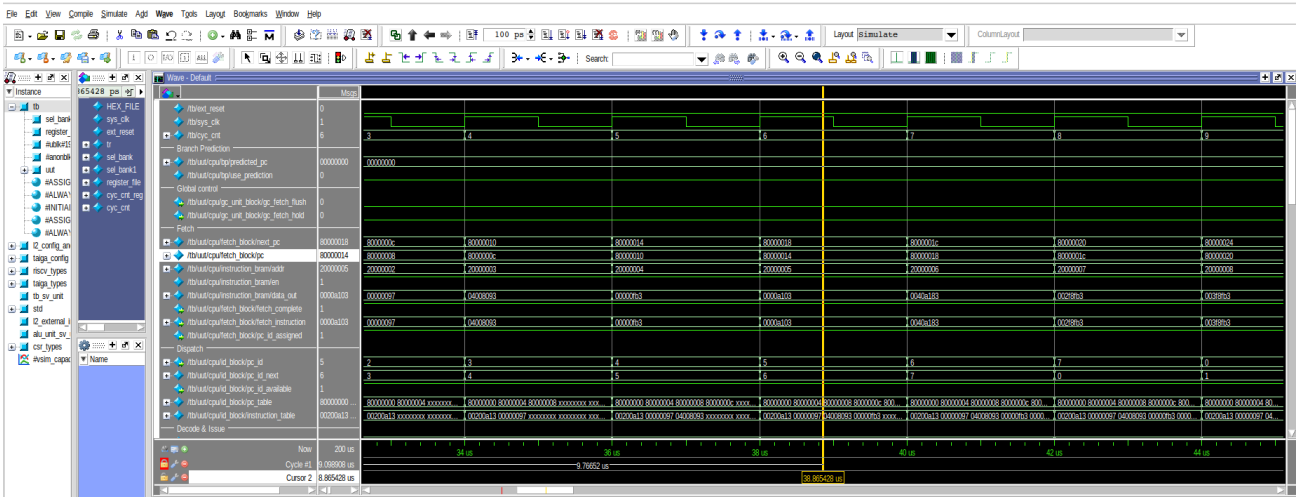

Рисунок 5 – Стадии выборки и диспетчеризации команды lw x3,4(x1)

Трасса неоптимизированной программы представлена на рисунке 8.

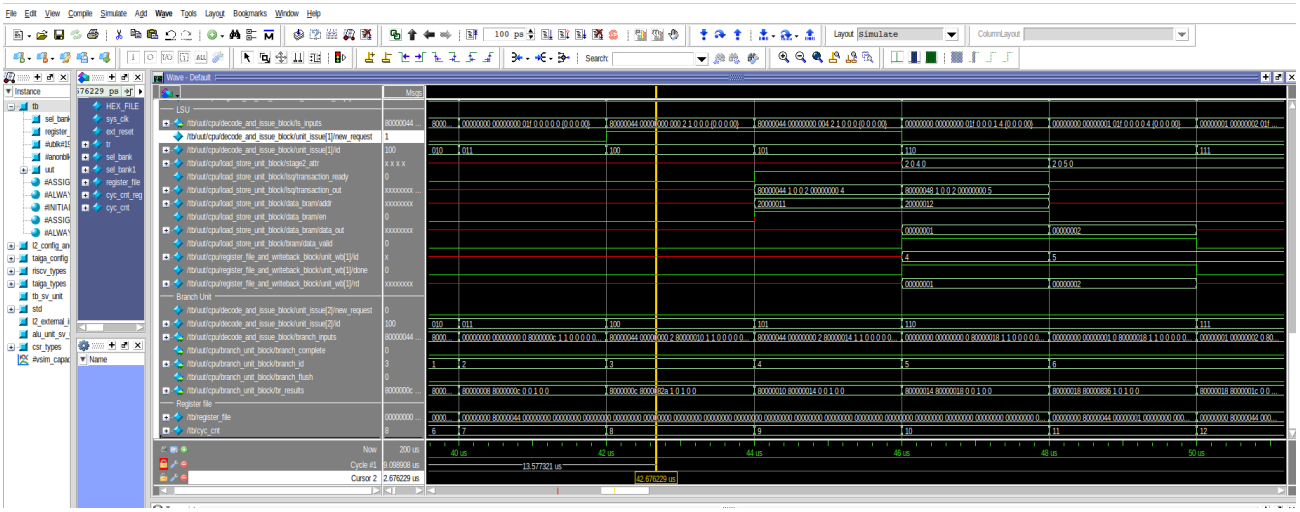

Рисунок 6 – Стадия декодирования команды lw x3,4(x1)

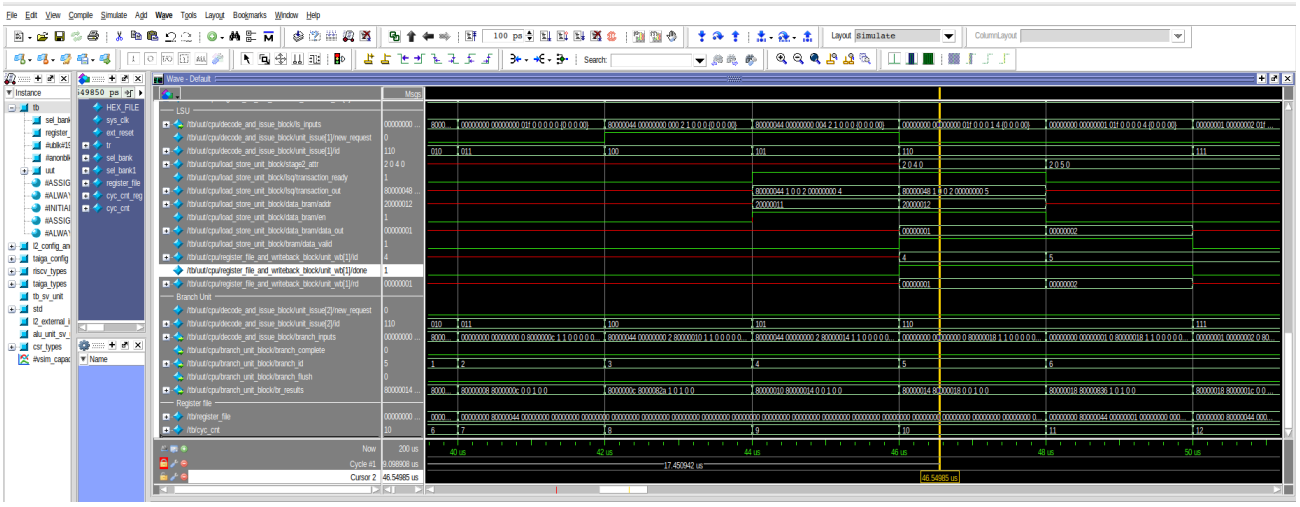

Рисунок 7 – Стадия выполнения команды lw x3,4(x1)

Конфликты возникают при выполнении команды *add* после команды *lw*. Для оптимизации (уменьшения числа конфликтов) перенесем все команды *lw* так, чтобы все они выполнялись раньше команд *add*.

Текст полученной программы представлен в листинге 4, а ее дизассемблерный листинг — в листинге 5

Листинг 4 – Текст оптимизированной программы

```
1     .section .text
2     .globl _start;
3     len = 8
4     enroll = 4
5     elem_sz = 4
6
7     _start:
8     addi x20, x0, len/enroll
9     la x1, _x
10    add x31, x0, x0
11    lp:
12    lw x2, 0(x1)
13    lw x3, 4(x1) #!
14    lw x4, 8(x1)
15    lw x5, 12(x1)
16    add x31, x31, x2
17    add x31, x31, x3
18    add x31, x31, x4
19    add x31, x31, x5
20    addi x1, x1, elem_sz*enroll
21    addi x20, x20, -1
22    bne x20, x0, lp
23    addi x31, x31, 1
24    lp2: j lp2
25
26    .section .data
27    _x:     .4byte 0x1
28    .4byte 0x2
29    .4byte 0x3
30    .4byte 0x4
31    .4byte 0x5
```

```
32      .4byte 0x6
33      .4byte 0x7
34      .4byte 0x8
```

Листинг 5 – Дизассемблированная оптимизированная программа

```
1       SYMBOL TABLE:
2       80000000 l    d  .text  00000000 .text
3       80000044 l    d  .data  00000000 .data
4       00000000 l    df *ABS*  00000000 optcode.o
5       00000008 l       *ABS*  00000000 len
6       00000004 l       *ABS*  00000000 enroll
7       00000004 l       *ABS*  00000000 elem_sz
8       80000044 l       .data  00000000 _x
9       80000010 l       .text  00000000 lp
10      80000040 l       .text  00000000 lp2
11      80000000 g       .text  00000000 _start
12      80000064 g       .data  00000000 _end
13
14
15
16      Дизассемблирование раздела .text:
17
18      80000000 <_start>:
19      80000000:   00200a13           addi    x20,x0,2
20      80000004:   00000097           auipc   x1,0x0
21      80000008:   04008093           addi    x1,x1,64 # 80000044 <_x>
22      8000000c:   00000fb3           add x31,x0,x0
23
24      80000010 <lp>:
25      80000010:   0000a103           lw  x2,0(x1)
26      80000014:   0040a183           lw  x3,4(x1)
27      80000018:   0080a203           lw  x4,8(x1)
28      8000001c:   00c0a283           lw  x5,12(x1)
29      80000020:   002f8fb3           add x31,x31,x2
30      80000024:   003f8fb3           add x31,x31,x3
31      80000028:   004f8fb3           add x31,x31,x4
32      8000002c:   005f8fb3           add x31,x31,x5
33      80000030:   01008093           addi    x1,x1,16
```

```
80000034:   fffa0a13            addi    x20,x20,-1
80000038:   fc0a1ce3            bne x20,x0,80000010 <lp>
8000003c:   001f8f93            addi    x31,x31,1

80000040 <lp2>:
80000040:   0000006f            jal x0,80000040 <lp2>

Дизассемблирование раздела .data:

80000044 <_x>:
80000044:   0001                    c.addi  x0,0
80000046:   0000                    c.unimp
80000048:   0002                    c.slli64    x0
8000004a:   0000                    c.unimp
8000004c:   00000003            lb  x0,0(x0) # 0 <elem_sz-0x4>
80000050:   0004                    .2byte  0x4
80000052:   0000                    c.unimp
80000054:   0005                    c.addi  x0,1
80000056:   0000                    c.unimp
80000058:   0006                    c.slli  x0,0x1
8000005a:   0000                    c.unimp
8000005c:   00000007            .4byte  0x7
80000060:   0008                    .2byte  0x8
...
```

На рисунке 9 представлена трасса оптимизированной программы.

Благодаря оптимизации программы получилось избавиться от конфликтов, а значит сократить время выполнения.

Адрес	Код команды	Команда	id	Номер такта																																														
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
80000000	00200a13	addi x20,x0,2	0	F	ID	D	AL																																											
80000004	00000097	auipc x1,0x0	1		F	ID	D	AL																																										
80000008	04008093	addi x1,x1,64 # 80000044 <_x>	2			F	ID	D	AL																																									
8000000c	00000fb3	add x31,x0,x0	3				F	ID	D	AL																																								
80000010	0000a103	lw x2,0(x1)	4					F	ID	D	M1	M2	M3																																					
80000014	0040a183	lw x3,4(x1)	5						F	ID	D	M1	M2	M3																																				
80000018	002f8fb3	add x31,x31,x2	6							F	ID	D	C	AL																																				
8000001c	003f8fb3	add x31,x31,x3	7								F	ID	W	D	AL																																			
80000020	0080a203	lw x4,8(x1)	0									F	ID	W	D	M1	M2	M3																																
80000024	00c0a283	lw x5,12(x1)	1										F	ID	W	D	M1	M2	M3																															
80000028	004f8fb3	add x31,x31,x4	2											F	ID	W	D	C	AL																															
8000002c	005f8fb3	add x31,x31,x5	3												F	ID	W	W	D	AL																														
80000030	01008093	addi x1,x1,16	4													F	ID	W	W	D	AL																													
80000034	fffa0a13	addi x20,x20,-1	5														F	ID	W	W	D	AL																												
80000038	fc0a1ce3	bne x20,x0,80000010 <lp>	6															F	ID	W	W	D	B																											
8000003c	001f8f93	addi x31,x31,1	7																F	ID	W	W	D	X																										
80000040	0000006f	jal x0,80000040 <lp2>	0																	F	ID	W	W	DX																										
80000044	00000001	<invalid operator>	1																		F	ID	W	X																										
80000048	00000002	<invalid operator>	2																			F	ID	X																										
8000004c	00000003	<invalid operator>	3																				F	X																										
80000050	00000004	<invalid operator>	4																					FX																										
80000010	0000a103	lw x2,0(x1)	0																						F	ID	D	M1	M2	M3																				
80000014	0040a183	lw x3,4(x1)	1																							F	ID	D	M1	M2	M3																			
80000018	002f8fb3	add x31,x31,x2	2																								F	ID	D	C	AL																			
8000001c	003f8fb3	add x31,x31,x3	3																									F	ID	W	D	AL																		
80000020	0080a203	lw x4,8(x1)	4																										F	ID	W	D	M1	M2	M3															
80000024	00c0a283	lw x5,12(x1)	5																											F	ID	W	D	M1	M2	M3														
80000028	004f8fb3	add x31,x31,x4	6																												F	ID	W	D	C	AL														
8000002c	005f8fb3	add x31,x31,x5	7																													F	ID	W	W	D	AL													
80000030	01008093	addi x1,x1,16	0																														F	ID	W	W	D	AL												
80000034	fffa0a13	addi x20,x20,-1	1																															F	ID	W	W	D	AL											
80000038	fc0a1ce3	bne x20,x0,80000010 <lp>	2																																F	ID	W	W	D	B										
80000010	0000a103	lw x2,0(x1)	3																																	F	ID	W	W	D	X									
80000014	0040a183	lw x3,4(x1)	4																																		F	ID	W	W	DX									
80000018	002f8fb3	add x31,x31,x2	5																																			F	ID	W	X									
8000001c	003f8fb3	add x31,x31,x3	6																																				F	ID	X									
80000020	0080a203	lw x4,8(x1)	7																																					F	X									
80000024	00c0a283	lw x5,12(x1)	0																																						FX									
8000003c	001f8f93	addi x31,x31,1	4																																							F	ID	D	AL					
80000040	0000006f	jal x0,80000040 <lp2>	5																																								F	ID	D	B				
80000044	00000001	<invalid operator>	6																																									F	ID	D	X			
80000048	00000002	<invalid operator>	7																																										F	ID	DX			
8000004c	00000003	<invalid operator>	0																																											F	X			
80000050	00000004	<invalid operator>	1																																												FX			
80000040	0000006f	jal x0,80000040 <lp2>	7																																													F	ID	D	B
Адрес	Код команды	Команда	id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
				Номер такта																																														

Рисунок 8 – Трасса работы неоптимизированной программы

Адрес	Код команды	Команда	id	Номер такта																																										
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
80000000	00200a13	addi x20,x0,2	0	F	ID	D	AL																																							
80000004	00000097	auipc x1,0x0	1		F	ID	D	AL																																						
80000008	04008093	addi x1,x1,64 # 80000044 <_x>	2			F	ID	D	AL																																					
8000000c	00000fb3	add x31,x0,x0	3				F	ID	D	AL																																				
80000010	0000a103	lw x2,0(x1)	4					F	ID	D	M1	M2	M3																																	
80000014	0040a183	lw x3,4(x1)	5						F	ID	D	M1	M2	M3																																
80000018	0080a203	lw x4,8(x1)	6							F	ID	D	M1	M2	M3																															
8000001c	00c0a283	lw x5,12(x1)	7								F	ID	D	M1	M2	M3																														
80000020	002f8fb3	add x31,x31,x2	0									F	ID	D	AL																																
80000024	003f8fb3	add x31,x31,x3	1										F	ID	D	AL																															
80000028	004f8fb3	add x31,x31,x4	2											F	ID	D	AL																														
8000002c	005f8fb3	add x31,x31,x5	3												F	ID	D	AL																													
80000030	01008093	addi x1,x1,16	4													F	ID	D	AL																												
80000034	fffa0a13	addi x20,x20,-1	5														F	ID	D	AL																											
80000038	fc0a1ce3	bne x20,x0,80000010 <lp>	6															F	ID	D	B																										
8000003c	001f8f93	addi x31,x31,1	7																F	ID	D	X																									
80000040	0000006f	jal x0,80000040 <lp2>	0																	F	ID	DX																									
80000044	00000001	<invalid operator>	1																		F	X																									
80000048	00000002	<invalid operator>	2																			FX																									
80000010	0000a103	lw x2,0(x1)	0																				F	ID	D	M1	M2	M3																		
80000014	0040a183	lw x3,4(x1)	1																					F	ID	D	M1	M2	M3																	
80000018	0080a203	lw x4,8(x1)	2																						F	ID	D	M1	M2	M3																
8000001c	00c0a283	lw x5,12(x1)	3																							F	ID	D	M1	M2	M3															
80000020	002f8fb3	add x31,x31,x2	4																								F	ID	D	AL																	
80000024	003f8fb3	add x31,x31,x3	5																									F	ID	D	AL																
80000028	004f8fb3	add x31,x31,x4	6																										F	ID	D	AL															
8000002c	005f8fb3	add x31,x31,x5	7																											F	ID	D	AL														
80000030	01008093	addi x1,x1,16	0																												F	ID	D	AL													
80000034	fffa0a13	addi x20,x20,-1	1																													F	ID	D	AL												
80000038	fc0a1ce3	bne x20,x0,80000010 <lp>	2																														F	ID	D	B											
80000010	0000a103	lw x2,0(x1)	3																															F	ID	D	X										
80000014	0040a183	lw x3,4(x1)	4																																F	ID	DX										
80000018	0080a203	lw x4,8(x1)	5																																	F	X										
8000001c	00c0a283	lw x5,12(x1)	6																																		FX										
8000003c	001f8f93	addi x31,x31,1	4																																			F	ID	D	AL					
80000040	0000006f	jal x0,80000040 <lp2>	5																																				F	ID	D	B				
80000044	00000001	<invalid operator>	6																																					F	ID	D	X			
80000048	00000002	<invalid operator>	7																																						F	ID	DX			
8000004c	00000003	<invalid operator>	0																																							F	X			
80000050	00000004	<invalid operator>	1																																								FX			
80000040	0000006f	jal x0,80000040 <lp2>	7																																									F	ID	D	B
Адрес	Код команды	Команда	id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
				Номер такта																																										

Рисунок 9 – Трасса работы оптимизированной программы