

#### 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.
<b>Группа</b> _ <u>ИУ7-56Б</u>
Оценка (баллы)
Преподаватель _ Попов А. Ю.

# Содержание

1	Зад	ание №1	2
	1.1	Текст программы по индивидуальному варианту	2
	1.2	Дизассемблерный листинг кода программы	3
	1.3	Псевдокод, поясняющий работу программы	5
2	Зад	ание №2	7
3	Зад	ание №3	7
4	Зад	ание №4	7
5	Зад	ание №5	9

### 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<sup>1</sup>, 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

<sup>&</sup>lt;sup>1</sup>https://gitlab.com/sfu-rcl/Taiga, авторы - Eric Matthews, Lesley Shannon

```
10
              add x31, x0, x0
11 lp:
12
              lw x2, 0(x1)
              Iw \times 3, 4(\times 1) \#!
13
              add x31, x31, x2
14
15
              add x31, x31, x3
              lw \times 4, 8(\times 1)
16
              lw x5, 12(x1)
17
              add x31, x31, x4
18
              add x31, x31, x5
19
              addi x1, x1, elem sz*enroll
20
21
              addi \times 20, \times 20, -1
22
              bne x20, x0, p
              addi x31, x31, 1
23
24 lp2: j lp2
25
             .section .data
26
27
             .4 byte 0x1
   _ x :
28
              .4 \, \text{byte} \, 0 \, \text{x} \, 2
29
              .4 byte 0x3
30
              .4 byte 0x4
31
              .4 byte 0x5
32
              .4 \, \text{byte} \, 0 \times 6
              .4 byte 0x7
33
34
              .4 byte 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 I d .text 00000000 .text
3 80000044 I d .data 00000000 .data
```

```
4
                      df *ABS*
       00000000 I
                                00000000 myprog.o
5
                         *ABS*
       00000008 I
                                 00000000 len
6
       00000004 I
                         *ABS*
                                00000000 enroll
7
                         *ABS*
                                00000000 elem sz
       00000004 l
8
                         . data
       80000044 l
                                 00000000 x
9
       80000010 I
                         .text
                                 00000000 lp
10
       80000040 l
                         . text
                                 00000000 lp2
11
                         .text
                                 00000000 start
       80000000 g
12
                                 00000000 end
       80000064 g
                         . data
13
14
      Дизассемблирование раздела .text:
15
16
       80000000 < start >:
17
                                          addi
                                                  x20, x0, 2
       80000000:
                    00200a13
18
                                                  x1,0x0
       80000004:
                    00000097
                                          auipc
                                                  x1,x1,64 #80000044 < x>
19
       80000008:
                                          addi
                    04008093
20
       8000000c:
                    00000fb3
                                          add x31,x0,x0
21
22
       80000010 <|p>:
23
       80000010:
                    0000a103
                                          Iw \times 2,0(\times 1)
                                          lw \times 3, 4(\times 1)
24
       80000014:
                    0040a183
                                          add x31,x31,x2
25
       80000018:
                    002f8fb3
26
       8000001c:
                    003f8fb3
                                          add x31,x31,x3
27
                                          Iw x4,8(x1)
       80000020:
                    0080a203
28
                                          lw x5,12(x1)
       80000024:
                    00c0a283
29
       80000028:
                    004f8fb3
                                          add x31,x31,x4
                                          add x31,x31,x5
30
       8000002c:
                    005f8fb3
31
                    01008093
                                          addi x1,x1,16
       80000030:
32
       80000034:
                    fffa0a13
                                          addi
                                                  \times 20, \times 20, -1
33
                    fc0a1ce3
                                          bne \times 20, \times 0,80000010 < lp >
       80000038:
34
       8000003c:
                    001f8f93
                                          addi
                                                  x31,x31,1
35
36
       80000040 < lp2 >:
37
       80000040:
                    0000006 f
                                          jal \times 0,80000040 < lp2>
38
      Дизассемблирование раздела .data:
39
40
41
       80000044 < x>:
42
       80000044:
                    0001
                                              c.addi x0,0
```

```
43
       80000046:
                     0000
                                                 c.unimp
44
       80000048:
                     0002
                                                 c.slli64
                                                               x0
45
       8000004a:
                     0000
                                                 c.unimp
                                            1b \times 0, 0(\times 0) \# 0 < elem \ sz-0x4>
46
       8000004c:
                     0000003
47
       80000050:
                     0004
                                                 .2 byte 0x4
48
       80000052:
                     0000
                                                 c.unimp
                                                 c.addi x0,1
49
       80000054:
                     0005
50
       80000056:
                     0000
                                                 c.unimp
                     0006
                                                 c.slli
51
       80000058:
                                                          x0,0x1
52
       8000005a:
                     0000
                                                 c.unimp
53
                                            .4 byte 0x7
       8000005c:
                     00000007
54
       80000060:
                     8000
                                                 .2 byte 0x8
55
```

#### 1.3 Псевдокод, поясняющий работу программы

На листинге 3 приведен псевдокод, поясняющий работу программы.

Листинг 3 – Псевдокод поясняющий работу программы

```
#define len 8
 1
       #define enroll 4
 2
 3
       #define elem sz 4
       int x[] = \{1, 2, 3, 4, 5, 6, 7, 8\};
 4
 5
       void start(){
 6
            int x20 = len/enroll;
 7
            int *x1 = x;
 8
            int x 31 = 0;
9
10
11
            do
12
            {
13
                 int x2 = x1[0];
                int x3 = x1[1];
14
15
                x31 += x2;
                x31 += x3;
16
17
18
                 int x4 = x1[2];
```

```
19
                 int x5 = x1[3];
                 x31 += x4;
20
                 x31 += x5;
21
22
23
                 x1 += enroll;
                 x20--;
24
            } while (\times 20 != \times 0)
25
26
27
            x31 += 1;
28
            while(1) {}
29
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.

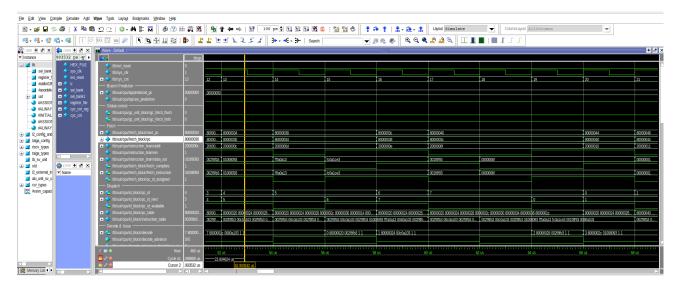


Рисунок 1 – Временная диаграмма выполнения стадий выборки и диспетчеризации команды с адресом 80000030 (1 итерация)

# 3 Задание №3

В результате симуляции, был получен снимок экрана, содержащий временную диаграмму выполнения стадии декодирования и планирования на выполнение команды с адресом 8000010 (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.

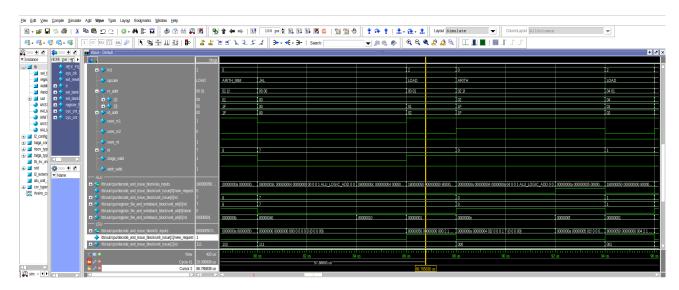


Рисунок 2 — Временная диаграмма выполнения стадий декодирования и планитрования на выполнение команды с адресом 80000010 (2 итерация)

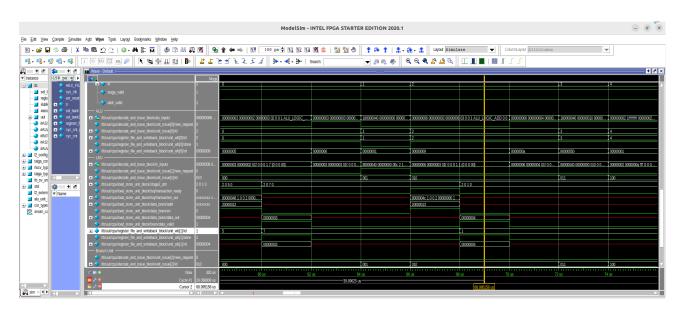


Рисунок 3 — Временная диаграмма стадии выполнения команды с адресом  $8000024~(1~{\rm итерация})$ 

# 5 Задание №5

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

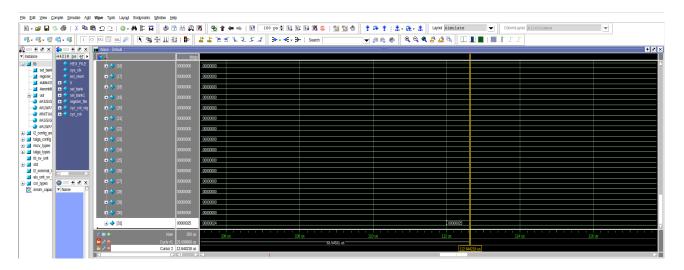


Рисунок 4 – Значение регистра х31 в конце выполнения программы

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.

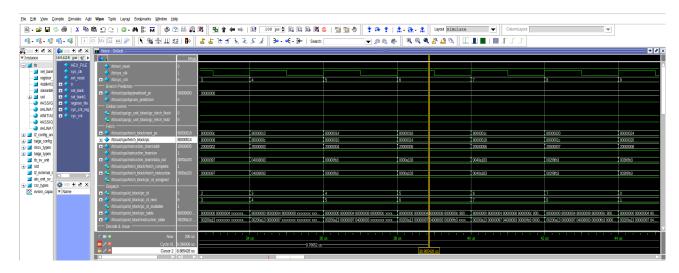


Рисунок 5 — Стадии выборки и диспет<br/>черизации команды lw х3,4(x1)

Трасса неоптимизированной программы представлена на рисунке 8.

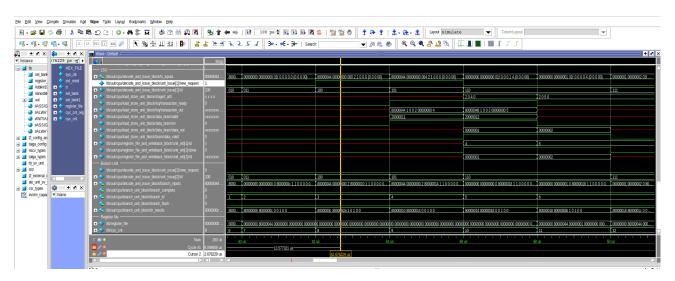


Рисунок 6 — Стадия декодирования команды lw x3,4(x1)

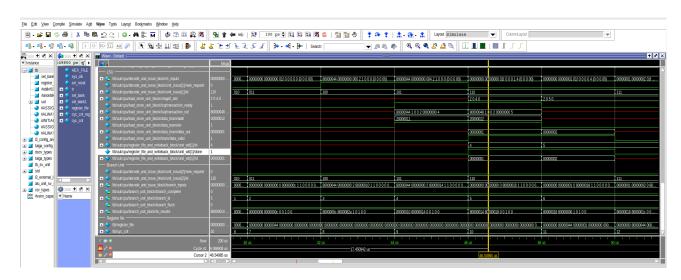


Рисунок 7 — Стадия выполнения команды lw x3,4(x1)

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

```
.section .text
 1
 2
       .globl _start;
 3
       len = 8
       enroll = 4
 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)
       Iw x3, 4(x1) #!
13
14
       Iw x4, 8(x1)
       lw x5, 12(x1)
15
       add x31, x31, x2
16
       add x31, x31, x3
17
18
       add x31, x31, x4
       add x31, x31, x5
19
20
       addi x1, x1, elem sz*enroll
21
       addi \times 20, \times 20, -1
22
       bne x20, x0, p
       addi x31, x31, 1
23
24
       lp2: j lp2
25
26
       .section .data
27
       х:
                 .4 byte 0x1
28
       .4 byte 0x2
       .4 \, \text{byte} \, 0 \, \text{x}3
29
30
       .4 byte 0x4
31
       .4 byte 0x5
```

Листинг 5 – Дизассемблированная оптимизированная программа

```
SYMBOL TABLE:
 1
 2
                                00000000 .text
       80000000 I
                     d .text
 3
       80000044 I
                     d .data
                                00000000 .data
 4
       00000000 I
                     df *ABS*
                                00000000 optcode.o
 5
                         *ABS*
       00000008 I
                                00000000 len
 6
                         *ABS*
       00000004 I
                               00000000 enroll
 7
                         *ABS*
                                00000000 elem sz
       00000004 l
 8
                                00000000 x
       80000044 I
                         . data
 9
       80000010 I
                         . text
                                00000000 lp
10
       80000040 I
                         .text
                                00000000 lp2
11
                         . text
                                00000000 start
       80000000 g
12
                                00000000 end
       80000064 g
                         . data
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
                                                 x1, x1, 64 \# 80000044 < x
       80000008:
                   04008093
                                         addi
22
                                         add x31,x0,x0
       8000000c:
                   00000fb3
23
24
       80000010 <|p>:
25
       80000010:
                   0000a103
                                         lw x2,0(x1)
26
                                         Iw x3,4(x1)
       80000014:
                   0040a183
27
       80000018:
                   0080a203
                                         lw x4,8(x1)
28
                                         lw x5, 12(x1)
       8000001c:
                   00c0a283
29
                                         add x31,x31,x2
       80000020:
                   002f8fb3
30
                   003f8fb3
                                         add x31,x31,x3
       80000024:
                                         add x31,x31,x4
31
       80000028:
                   004f8fb3
32
       8000002c:
                   005f8fb3
                                         add x31,x31,x5
33
       80000030:
                   01008093
                                         addi
                                                 x1, x1, 16
```

```
34
                    fffa0a13
       80000034:
                                           addi
                                                    x20, x20, -1
35
                                           bne x20, x0, 80000010 < lp >
       80000038:
                    fc0a1ce3
36
       8000003c:
                    001f8f93
                                           addi
                                                    x31,x31,1
37
38
       80000040 < lp2 >:
39
       80000040:
                    0000006 f
                                           jal x0,80000040 < lp2 >
40
       Дизассемблирование раздела .data:
41
42
43
       80000044 < x>:
44
       80000044:
                                               c.addi x0,0
                    0001
45
       80000046:
                    0000
                                               c.unimp
                                               c.slli64
46
       80000048:
                    0002
                                                             x0
47
       8000004a:
                    0000
                                               c.unimp
48
       8000004c:
                    0000003
                                           lb
                                               x0,0(x0) \# 0 < elem sz-0x4>
                    0004
49
       80000050:
                                               .2 byte 0x4
50
       80000052:
                    0000
                                               c.unimp
                    0005
51
       80000054:
                                               c.addi x0,1
52
       80000056:
                    0000
                                               c.unimp
53
       80000058:
                    0006
                                               c.slli
                                                        \times 0, 0 \times 1
                    0000
54
       8000005a:
                                               c.unimp
                    0000007
55
                                           .4 byte 0x7
       8000005c:
56
       80000060:
                    8000
                                                .2 byte
                                                        8x0
57
```

На рисунке 9 представлена трасса оптимизированной программы.

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

	Код	1																				_	امدوا	on ·	такт																	_	—	—	_
Адрес	команды	Команда	id	1	2 3	1 4	15	6	7 T	8	9 1	0 1	1 12	13	14	15	16 1	7 18	3 10	9 20	21						7 2	8 20	9 30	9 31	32	33	34	35	36 3	7 38	3 39	40	41	42 4	43 4	4 4	5 46	147	48
80000000	00200a13	addi x20,x0,2			D D			Ĭ		Ť		1	1	1	1	1	10	1	1	1	1		-				1	1			1	1	Ţ.		30 3	1	1		1		<u> </u>				
80000004	00000097	auipc x1,0x0	1		F II					_									Т							т	т											П				т	т	т	
80000008	04008093	addi x1,x1,64 # 80000044 < x>	2					AL																																					
8000000c	00000fb3	add x31,x0,x0	3			F	ID	D.	AL	$\neg$									Т							т	т	т										П				т	т	т	$\Box$
80000010	0000a103	lw x2,0(x1)	4				l F	ID	DI	41	M2 N	13																																	
80000014	0040a183	lw x3,4(x1)	5				Т	F	ID	DΙ	M1 IN	12 l M	3						Т							т	т	т										П				т	т	т	$\Box$
80000018	002f8fb3	add x31,x31,x2	6						F I	ID	D	C A	L																																
8000001c	003f8fb3	add x31,x31,x3	7		т	Т	Т			F	ID	W E	AL			$\neg$	$\neg$		Т							т	т	Т								Т		П			$\neg$	т	т	т	
80000020	0080a203	lw x4,8(x1)	Θ								F ]	DV	I D	M1	M2	МЗ																													
80000024	00c0a283	lw x5,12(x1)	1				Т			$\neg$		FΙ	D W	D	M1	M2 I	мз		Т							т	т	т										П				т	т	т	$\Box$
80000028	004f8fb3	add x31,x31,x4	2									F	II	W	D	C	AL																												
8000002c	005f8fb3	add x31,x31,x5	3				Т			$\neg$			F	ID	W	W	D A	L	Т							т	т	т										П				т	т	т	$\Box$
80000030	01008093	addi x1,x1,16	4											F	ID	W	W	D AL	L																										
80000034	fffa0a13	addi x20,x20,-1	5				T								F	ID	W	N D		L									Т													Т	Т	T	
80000038	fc0alce3	bne x20,x0,80000010 <lp></lp>	6													F :		N W		В																									
8000003c	001f8f93	addi x31,x31,1	7											П			F ]	D W								Т	Т				П											Т		T	
80000040	0000006f	jal x0,80000040 <lp2></lp2>	Θ															FI	) W	W	DX																								
80000044	00000001	<invalid operator=""></invalid>	1							$\neg$								F	I	) W	Х					Т												П				т	Т	Т	
80000048	00000002	<invalid operator=""></invalid>	2																F	ID	X																								
8000004c	00000003	<invalid operator=""></invalid>	3		т	Т	Т			$\neg$		$\top$				$\neg$	$\neg$		Т	F	Х					т	т	Т								Т		П			$\neg$	т	т	т	$\Box$
80000050	00000004	<invalid operator=""></invalid>	4																		FX																								
80000010	0000a103	lw x2,0(x1)	0			Т				$\neg$									Т			F			11 M													П				Т	Т	Т	
80000014	0040a183	lw x3,4(x1)	1																				F :	ID	D M	11 M	12 M	3																	
80000018	002f8fb3	add x31,x31,x2	2							$\neg$									Т						ED I													П				т	Т	Т	
8000001c	003f8fb3	add x31,x31,x3	3																						FI	DI	N E	) AI																	
80000020	0080a203	lw x4,8(x1)	4		т	Т	Т			$\neg$		$\top$				$\neg$	$\neg$		Т	Т								۷ D	M:	1 M2	М3					Т		П			$\neg$	т	т	т	
80000024	00c0a283	lw x5,12(x1)	5																								FΙ	D W	D	M1 D	M2	M3													
80000028	004f8fb3	add x31,x31,x4	6			Т				$\neg$									Т							Т	F	- II	) W	D	C	AL						П				Т	Т	Т	
8000002c	005f8fb3	add x31,x31,x5	7																									F		) W	W	D	AL												
80000030	01008093	addi x1,x1,16	Θ			Т				$\neg$									Т							Т	Т		F	ID	W		D	AL				П				Т	Т	Т	
80000034	fffa0a13	addi x20,x20,-1	1																											F	ID		W		AL										
80000038	fc0alce3	bne x20,x0,80000010 <lp></lp>	2		Т					П		T					Т		Г					Т		Т	Т	Т	Г		F		W	W	DI				П		Т				
80000010	0000a103	lw x2,0(x1)	3																													F	ID			X									
80000014	0040a183	lw x3,4(x1)	4																Т														F	ID	W۱	v D>									
80000018	002f8fb3	add x31,x31,x2	5																															F	ID V	N X									
8000001c	003f8fb3	add x31,x31,x3	6																																	D X									$\Box$
80000020	0080a203	lw x4,8(x1)	7				1														1											1			- 1	F X									
80000024	00c0a283	lw x5,12(x1)	0						T	П	Т					Т	Т						Т	T										ΙП		F)		П							17
8000003c	001f8f93	addi x31,x31,1	4																																		F	ID		AL					
80000040	0000006f	jal x0,80000040 <lp2></lp2>	5																Т					П		Т	Т		П									F	ID		В	Т		T	
80000044	00000001	<invalid operator=""></invalid>	6																																				F		D 2				
80000048	00000002	<invalid operator=""></invalid>	7			Т													Т										Г											F :	ID D	Х	T	T	
8000004c	00000003	<invalid operator=""></invalid>	Θ																																						F 2	K			
80000050	00000004	<invalid operator=""></invalid>	1																Т										Г												F	X		T	
80000040	0000006f	jal x0,80000040 <lp2></lp2>	7																																								= ID		
Адрес	Код	Команда	id	1	2 3	4	5	6	7	8	9 1	0 1	1 12	13	14	15	16 1	7 18	3 19	9 20	21	22	23	24	25 2	6 2	7 2	8 29	9 30	9 31	32	33	34	35	36 3	7 38	39	40	41	42	43 4	4 4	5 46	5 47	48
Адрес	команды	Команда	110																				Ном	ep ·	такт	ra																			

Рисунок 8 — Трасса работы неоптимизированной программы

	Код		Ι	Т																		Hor	мen	тан	ста																	_	_
Адрес	команды	Команда	id	1	2	3	4	5 T	6 7	7 8	9	10	11	12	13	14 1	5 16	5 1	7   18	19	20	21	22	23	24	25	261	27 2	8 2	9 3	0 3	1 3	2 3	3 34	35	36	37	38 3	39 4	0 4	1 42	43	44
80000000	00200a13	addi x20.x0.2	Θ	F		D							i –	H																													
80000004	00000097	auipc x1,0x0	1		F	ID	D A	٩L			Т		П					Т	т											т		т		Т						т		т	
80000008	04008093	addi x1,x1,64 # 80000044 < x>	2						AL.																																		
8000000c	00000fb3	add x31,x0,x0	3						D A	L																														т		т	
80000010	0000a103	lw x2,0(x1)	4						D [	M:	1 M2	M3																															
80000014	0040a183	lw x3,4(x1)	5							D D	M3	L M2	МЗ																											т		т	
80000018	0080a203	lw x4,8(x1)	6						F	= 10			M2	МЗ																													
8000001c	00c0a283	lw x5,12(x1)	7					_		F				M2	мз																									т		т	
80000020	002f8fb3	add x31,x31,x2	Θ								IF	ID	D	AL																													
80000024	003f8fb3	add x31,x31,x3	1					_				F		D	AL																									т		т	
80000028	004f8fb3	add x31,x31,x4	2										F		D	AL																											
8000002c	005f8fb3	add x31,x31,x5	3					Т			Т		П	F	ID	D A	L	Т	т											т		т		Т						т		т	
80000030	01008093	addi x1,x1,16	4												F :		D AI																										
80000034	fffa0a13	addi x20,x20,-1	5								Т						D D											Т				Т	Т	Т									
80000038		bne x20,x0,80000010 <lp></lp>	6														FI																										
8000003c	001f8f93	addi x31,x31,1	7					П									F		D D	Х								Т		Т			Т							Т			
80000040	0000006f	jal x0,80000040 <lp2></lp2>	Θ															F	II	DX																							
80000044	00000001	<invalid operator=""></invalid>	1					П										Т	F	Х								Т		Т			Т							Т			
80000048	00000002	<invalid operator=""></invalid>	2																	FX																							
80000010	0000a103	lw x2,0(x1)	Θ					Т										Т			F	ID		M1						Т										Т	Т		
80000014	0040a183	lw x3,4(x1)	1																			F	ID	D																			
80000018	0080a203	lw x4,8(x1)	2					Т										Т					F	ID	D	M1	M2 I	13		Т										Т	Т		
8000001c	00c0a283	lw x5,12(x1)	3																					F	ID	D	M1 I	12 1	13														
80000020	002f8fb3	add x31,x31,x2	4					Т										Т							F	ID		٩L		Т										Т	Т		
80000024	003f8fb3	add x31,x31,x3	5																							F	ID		\L														
80000028	004f8fb3	add x31,x31,x4	6					Т										Т											D A											Т	Т		
8000002c		add x31,x31,x5	7																											D A													
80000030	01008093	addi x1,x1,16	Θ																												D A												
80000034		addi x20,x20,-1	1																											FI	D D	) AI	L										
80000038		bne x20,x0,80000010 <lp></lp>	2															П										Т				D D											
80000010	0000a103	lw x2,0(x1)	3																												F												
80000014	0040a183	lw x3,4(x1)	4				J																									F		D)									
80000018		lw x4,8(x1)	5																														F										
8000001c	00c0a283	lw x5,12(x1)	6				J																											FΧ									
8000003c		addi x31,x31,1	4																																F	ID	D						
80000040	0000006f	jal x0,80000040 <lp2></lp2>	5																																	F	ID		В				
80000044	00000001	<invalid operator=""></invalid>	6																																		F		D >				
80000048	00000002	<invalid operator=""></invalid>	7																																				ID D				
8000004c	00000003	<invalid operator=""></invalid>	Θ																																				F >				
80000050	00000004	<invalid operator=""></invalid>	1				J																																F	X			
80000040	0000006f	jal x0,80000040 <lp2></lp2>	7																																						ID		
Адрес	Код	Команда	id	1	2	3	4	5	6   7	7 8	9	10	11	12	13	14 1	5   16	5   1	7   18	19	20					25	26	27 2	8   2	9 3	0 3	1   3	2   3	3   34	35	36	37	38	39   4	0 4	1 42	<u> 43</u>	44
Адрес	команды	Nomanga	1-0																			Ho	чер	тан	кта															_		_	

Рисунок 9 – Трасса работы оптимизированной программы