

# 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)»

(национальный исследовательский университет)» (МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика, искус	сственный интеллект и системи	ы управления»
КАФЕДРА «Программное обеспече	ение ЭВМ и информационные т	ехнологии»
	ОТЧЕТ	
по лаб	бораторной работе № 2	
Название: <u>Изучение прини</u>	ипов работы микропроце	ессорного ядра RISC-
Дисциплина: Архитектура	ЭRM	
дисциплина. Архитектура	<u> JDM</u>	
Студент <u>ИУ7-52Б</u> (Группа)	 (Подпись, дата)	
Преподаватель	(, Hara)	А.Ю.Попов

(Подпись, дата)

(И. О. Фамилия)

**Цель работы** — Основной целью работы является ознакомление с принципами функциони- рования, построения и особенностями архитектуры суперскалярных конвейер- ных микропроцессоров. Дополнительной целью работы является знакомство с принципами проектирования и верификации сложных цифровых устройств с использованием языка описания аппаратуры SystemVerilog и ПЛИС.

#### Задание 1

Листинг 1.1. Исходный текст исследуемой программы

```
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.
           add x31, x0, x0
11. lp:
12.
           lw x2, \theta(x1)
13.
           add x31, x31, x2 #!
14.
           addi x1, x1, elem_sz*enroll
15.
         bne x1, x20, lp
16.
           addi x31, x31, 1
17. lp2: j lp2
18.
19.
           .section .data
20. x:
           .4byte 0x1
21. .4byte 0x2
22.
           .4byte 0x3
23.
          .4byte 0x4
24.
           .4byte 0x5
25.
           .4byte 0x6
26.
            .4byte 0x7
27.
           .4byte 0x8
```

## Листинг 1.2. Дизассемблерный листинг

```
1. iscv64-linux-gnu-ld -b elf32-littleriscv -T link.ld task.o -o task.elf
2. riscv64-linux-gnu-objdump -D -M numeric,no-aliases -t task.elf
3.
4. task.elf: формат файла elf32-littleriscv
5.
6. SYMBOL TABLE:
7. 80000000 1 d .text 00000000 .text
8. 80000028 1 d .data 00000000 task.o
10. 00000008 1 *ABS* 00000000 len
11. 00000001 1 *ABS* 00000000 enroll
```

```
12. 00000004 1
                     *ABS* 00000000 elem sz
13. 80000028 1
                     .data 00000000 _x
14. 80000010 1
                     .text 00000000 lp
15. 80000024 1
                     .text 00000000 lp2
16. 80000000 g
                     .text 00000000 _start
17. 80000048 g
                     .data 00000000 _end
18.
19.
20.
21. Дизассемблирование раздела .text:
22.
23. 80000000 <_start>:
24. 80000000: 00000097
                                 auipc x1,0x0
                                 addi x1,x1,40 # 80000028 <_x>
25. 80000004: 02808093
26. 80000008: 01c08a13
                                 addi x20,x1,28
27. 8000000c: 00000fb3
                                 add x31, x0, x0
28.
29. 80000010 <1p>:
30. 80000010: 0000a103
                                 lw x2,0(x1)
31. 80000014: 002f8fb3
                                 add x31,x31,x2
32. 80000018: 00408093
                                 addi x1,x1,4
33. 8000001c: ff409ae3
                                 bne x1,x20,80000010 <lp>
34. 80000020: 001f8f93
                                 addi x31,x31,1
35.
36. 80000024 <1p2>:
37. 80000024: 0000006f
                                 jal x0,80000024 <1p2>
38.
39. Дизассемблирование раздела .data:
40.
41. 80000028 <_x>:
42. 80000028: 0001
                                   c.addi x0,0
43. 8000002a: 0000
                                   c.unimp
44. 8000002c: 0002
                                   c.slli64 x0
45. 8000002e: 0000
                                   c.unimp
46. 80000030: 00000003
                                 lb x0,0(x0) # 0 <enroll-0x1>
47. 80000034: 0004
                                   0x4
48. 80000036: 0000
                                   c.unimp
49. 80000038: 0005
                                   c.addi x0,1
50. 8000003a: 0000
                                   c.unimp
51. 8000003c: 0006
                                   c.slli x0,0x1
52. 8000003e: 0000
                                   c.unimp
53. 80000040: 00000007
54. 80000044: 0008
                                   0x8
55. ...
```

#### Листинг 1.3 Псевдокод программы

```
1. #define len 8
2. #define enroll 1
3. #define elem_sz 4
4. int _x[]={1,2,3,4,5,6,7,8};
5. void _start() {
6. int *x1 = _x;
7. int *x20 = x1 + len-1; // m.e. x20 - адрес последнего элемента массива
```

```
8.
       int x31 = 0;
9.
10.
       do {
11.
     int x2 = x1[0];
12.
         x31 += x2;
13.
         x1 += enroll;
14.
       } while(x20 != x1);
15.
       x31++;
16.
       while(1){}
17.}
```

В конце программы в регистре x31 должна содержаться сумма всех элементов массива, кроме последнего, плюс 1. Таким образом результат будет равен 29 в десятичной системе счисления или 1d в шестнадцатиричной.

#### Задание 2

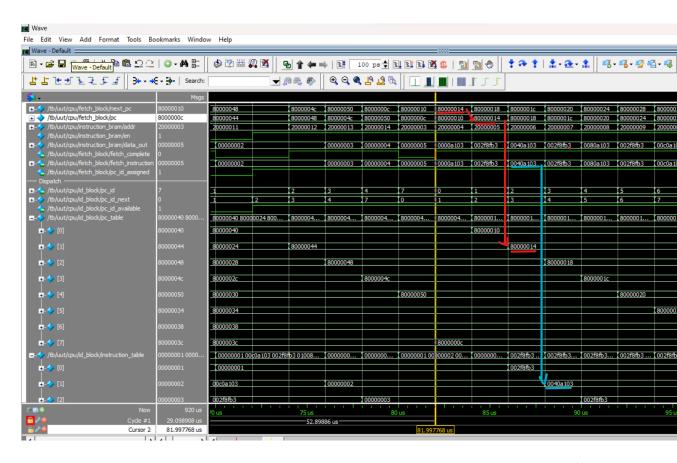


Рисунок 3.1. Временная диаграмма выполнения стадий выборки и диспетчеризации команды с адресом 80000014 для команд, входящих в тело цикла, на 1-ой итерации

#### Задание 3

Получена временная диаграмма стадии декодирования и планирования на выполнение команды с адресом 80000020 для команд, входящих в тело цикла на 1-ой итерации. Конфликта нет (так как значения rs1\_conflict и rs2\_conflict равны 0).

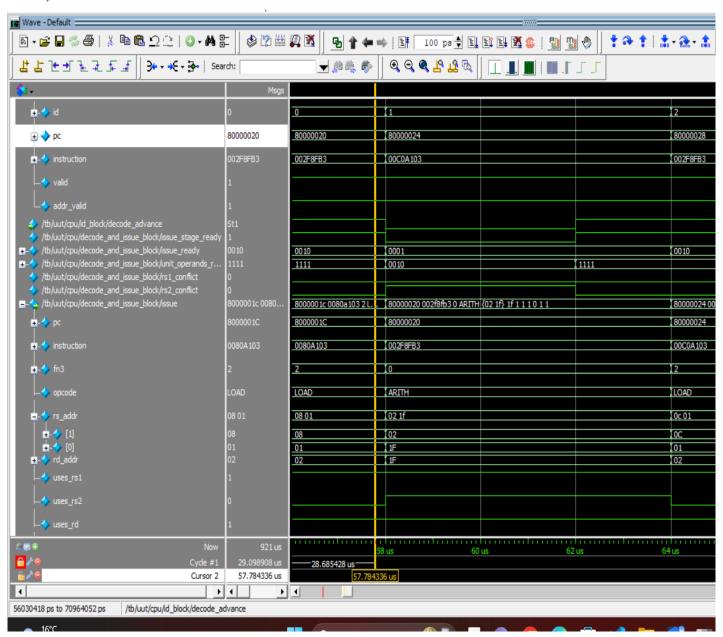


Рисунок 3.2. Временная диаграмма стадии декодирования и планирования на выполнение команды с адресом 80000020 для команд, входящих в тело цикла на 1-ой итерации.

## Задание 4

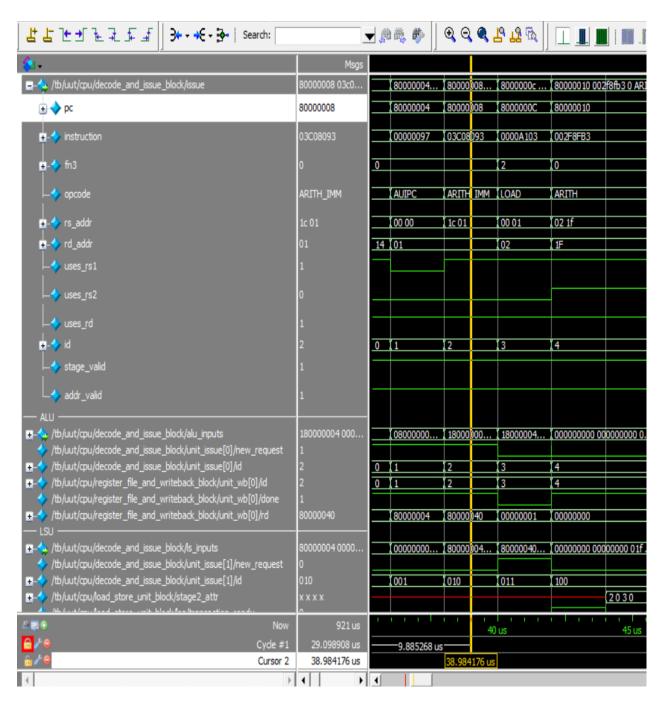


Рисунок 3.3. Временная диаграмма стадии выполнения команды с адресом 80000008.

# Задание 5

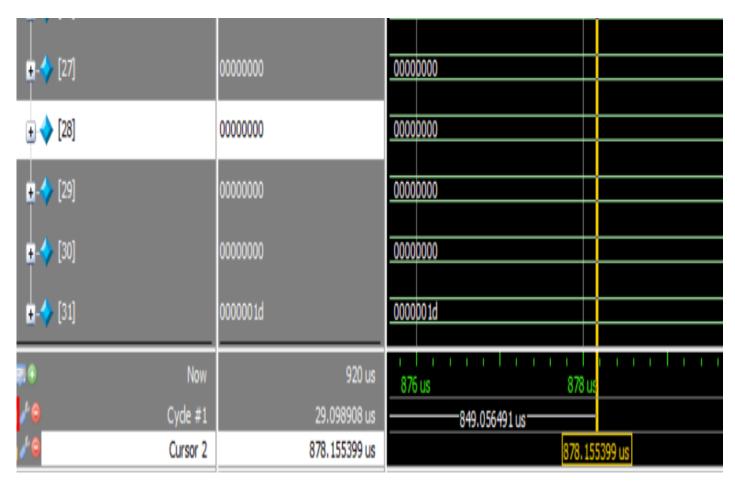


Рисунок 3.4 – Значение регистра х31

Временные диаграммы сигналов, соответствующие всем стадиям выполнения команды add x31, x31, x2 #!.

Из дизассемблированного кода можно узнать адрес данной команды:

1. **80000014**: 002f8fb3 add x31,x31,x2

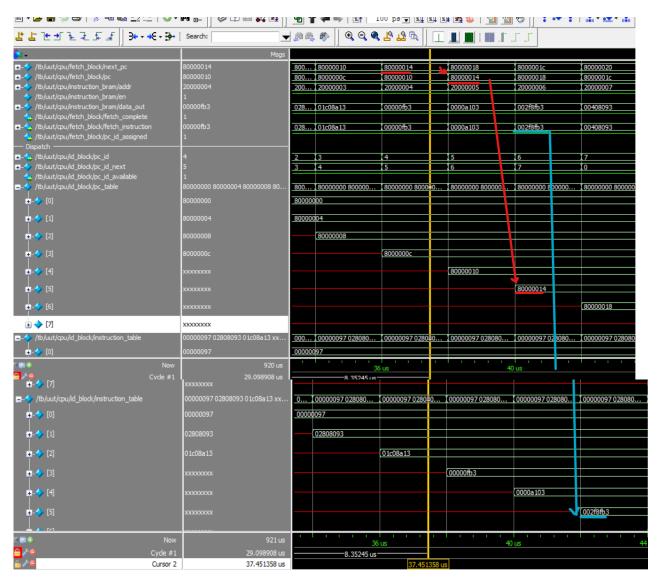


Рисунок 3.5. Временная диаграмма выполнения стадий выборки и диспетчеризации команды с адресом 80000014 для команд, входящих в тело цикла, на 1-ой итерации.

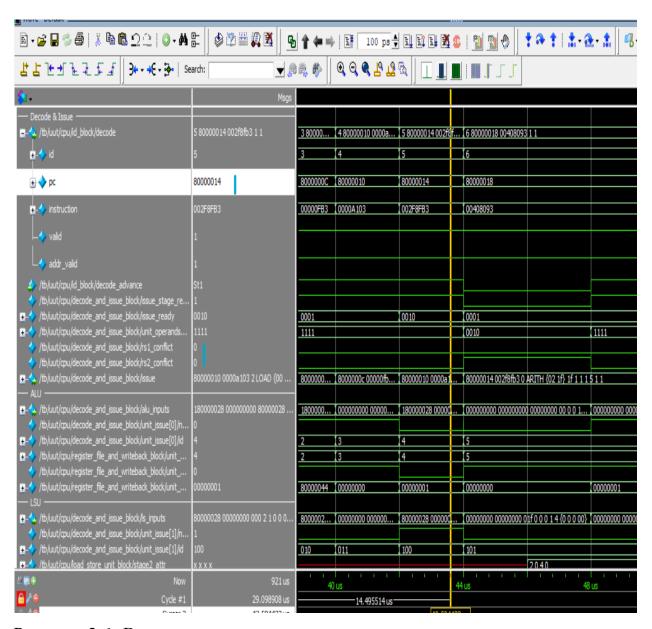


Рисунок 3.6. Временная диаграмма стадии декодирования и планирования на выполнение команды с адресом 80000014 для команд, входящих в тело цикла на 1-ой итерации.

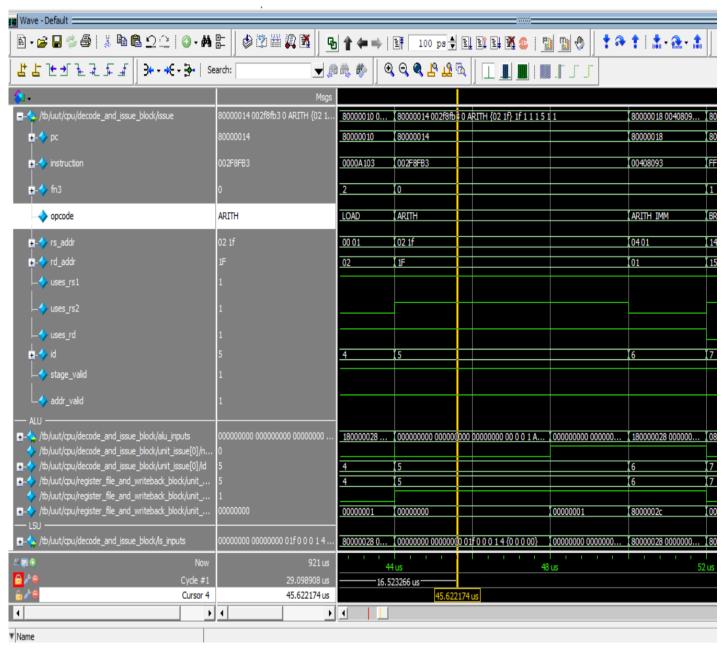


Рисунок 3.7. Временная диаграмма стадии выполнения команды с адресом 80000014 для команд, входящих в тело цикла на 1-ой итерации.

80000004 80000008	команды 00000097												100	1613	200	0 2 2	nn.	25	22 2	9 9 4	100	20	99 m	200	20	31	221																											
0000004		auipc x1,0x0	0 1	P ID	D A		0	6	7	11 11	12	3 14	13	10 1	1	0 13	ZU	21	E & .	3 24	23	Z D	21 2	2 6 2	30	.51	36 .	3 3	9 33	30	21 3	0 35	40 4	1 42	4.5	44 4	3 45	4/	40 4	2 30	1 31	3Z	3.3	29 2	13 3	9 3	1 3	0 .		T	1	2 0	ï	1
0000008	03c08093	addi x1,x1,40 # 80000028	1,1		ID D				1																																									I			-	4
		addi x1,x1,40 # 80000028	2		FI		BT																																											ı				ď
	00000fb3	add x31.x0.x0	Ľ.			ID			-												Н											Н												4		+	Ŧ			Н			7	
		1w x2,0(x1)	2		,			M1	100	en en																																								٠				i
	002181B3 0040a103	add x31,x31,x2	1					D D					Н																			Н												4		+	+			Н	۰		7	
	00408093	addi x1,x1,4	b .		-		_	ID		W D							н				Н				Н							н								н						-	-			٠			_	
	ff409ae3	the state of the s	b					F			D																					Н								Н				4		+	Ŧ			Н			7	
		bne x1,x20,80000010 <1p>	7		4	ш		F	_	_	_	_		-	-		ш				ш			н	ш					ш	-	н			ш					-	н	ш	_	-		_	+	-	-	٠	-	-	_	
	00118193	addi x31,x31,1	a							D W		D X																				Н												4		4				ı			4	
	0000000er	jal x0,80000024 <1p2>	1		4	ш			-		W			_	4		ш				ш				ш	ш				ш		ш			Ш					_		ш	_	_		_	_		_	L		_	_	
	0001	<pre><invalid comman=""></invalid></pre>	2						-	F	ID		_																			Н												4		4				ı			4	
	0002	<invalid comman=""></invalid>	3		ш	Ш			_		_	ED X	_	_	_		ш				Ш				Ш					Ш		ш			Ш			Ш				Ш	_	_		_	$\perp$		_	L		_	╧	
	00000003	<invalid comman=""></invalid>	4						1			FX	_																																					Т			4	
	0004	<invalid comman=""></invalid>	5		Ш				_			F									Ш		_	Ш	ш	ш		_		Ш	_	ш			Ш	_		Ш	_			Ш	_	_	_	_	_	_	_	1	_	_	┸	
	002f8fb3	1w x2,0(x1)	2													1 M2	МЗ																																	ı				
	0040a103	add x31,x31,x2	3		$\perp$				_				Ш		ID I			AL																	Ш									_		$\perp$			1	L	1	1	_	
		addi x1,x1,4	4											1		D W																																						
	ff409ae3	bne x1,x20,80000010 <1p>	7												3	ID			D B																															1				
	002f8fb3	1w x2,0(x1)	a						н							F	ID	W	W D	) M1	M2	М3			П							Ш						Ш		П										Т				
	0040a103	add x31,x31,x2	1										П		Т		F		WW				AL.		П	П												П		Т			П	Т		Т	Т		Τ	Τ				
80000018	00408093	addi x1,x1,4	2															F	ID W	W	W	W	D A	E.																										ı				
8000001c	ff409ae3	bne x1,x20,80000010 <1p>	3		П	П		П	Т	Т	П	Т	П		Т	Т			F II	D W	W	W	W D	В		П			Т			П		Т	П	Т	Т	П	Т	Т			Т	Т	Т	Т	Т	Т	Т	Т	Т	Т	Т	
80000010<1p>	002f8fb3	1w x2,0(x1)	4																F	II	W	W	WW	D	M1	M2 I	мз																							ı				
80000014	0040a103	add x31,x31,x2	5	Т		П	Т	П	Т	Т	П	Т	П	Т	Т	Т				F	ID	W	w w	W	D	c	C 2	L.	Т	П		П		Т	П	Т	Т	П	Т	Т		П	П	т	Т	т	т	Т	Т	Т	Т	Т	Т	ī
80000018	00408093	addi x1,x1,4	6																		F	ID	w w	W	W	W	W	D A	L																					t				
8000001c	ff409ae3	bne x1,x20,80000010 <1p>	7			П		П	т				П	т	т	т				т	П	P	ID W	W	w	w	W	w I	В	П		П			П	т	Т		т	Т		П	П	т		т	т	т	т	Т	т	т	т	ī
80000010<1p>	002f8fb3	1w x2,0(x1)	a																				F	ID	W	W	_	_	_	-	M2 M	3																		t				i
	0040a103	add x31,x31,x2	ri I			П			т				П								П		1	P	TD	W	_	_	_	_					П					П				т		т	т			Т	т		т	f
	00408093	addi x1,x1,4	2																						p	TD	w	w s	e te	w	te te	D	AL																	t				i
	ff409ae3	bne x1,x20,80000010 <1p>	2						т				П													***	P 1			w	W W	w	D E	R										7			т			т			т	ı
		1w x2,0(x1)	1																										_	-	ww	10		D M1	Ma	MO														٠				i
	0040a103	add x31,x31,x2	1						-				Н															-		1	W W	10	W			C A								7		+	T			Н			T	
	00408093	addi x1,x1,4	2																											77	T		W	w w	_	_	D AI													٠				
	ff409ae3	bne x1,x20,80000010 <1p>							-																					F	ID W	W	W											4		+	Ŧ			Н			7	
	002f8fb3	1w x2,0(x1)	, ,																		Н											11		D W																٠			•	
	0040a103	add x31,x31,x2	, u						-																							Н		PII							AL			4		+	+			Н			7	ı
	00408093	add x31,x31,x2 addi x1,x1,4	3						-												Н											н	,		ID							AL					-			٠			_	
	ff409ae3	add1 x1,x1,4 bne x1,x20,80000010 <1p>	4						-												Н											Н		F			D W							4		+	+			Н			7	
			5		-				-						-		ш				Н											н				FI							D :			_	_			٠			_	
		1w x2,0(x1)	6			Н			-				Н																			Н					P									4	+			H			4	
	0040a103	add x31,x31,x2	7		4				4						-		ш				ш				ш							н						F	IDI				W			_	_	_	_	1	_	_	_	
		addi x1,x1,4	a			Н			-				Н								Н											Н							F I	_	_	_	D :	_		4				ı			4	
	ff409ae3	bne x1,x20,80000010 <1p>	5		ш	Ш		ш	4		Ш		ш	_	_		ш				ш			ш	ш	ш				Ш		ш			Ш			Ш		F		_	D :	_		_	_		_	L	_	_	╧	
		1w x2,0(x1)	6																																							_	ID:	_									4	
	0040a103	add x31,x31,x2	7		$\perp$				_				Ш				Ш																										F	_		-	+			L			_	
	00408093	addi x1,x1,4	a																																								7	FΧ										
	001f8f93	addi x31,x31,1	6		4				4																																			_		ID D				L			_	
	00000006£	jal x0,80000024 <1p2>	7																																										F	F II								
	0001	<invalid comman=""></invalid>	2	1													Ш																													F	P I			K			4	
	0002	<invalid comman=""></invalid>	3																																												F	PI		X				
	00000003	<invalid comman=""></invalid>	6		-												Н																														+	F	F' )	K			4	
	0004	<invalid comman=""></invalid>	7																																									4					F	X				1
	001f8f93	addi x31,x31,1	6		-																																													1			D I	
80000024<1p2>	001f8f93	addi x31,x31,1	6		-																				3 3 0								400										_			+	+			+	_	-	D 1	
Адрес	Код комания	Команда	id	1 2	3 4	5	6	8	9 ]	u 11	12	1.3 1.4	1.5	16 1	1 7 1	8 19	20	21	12 2	3 24	25	26	47 2	8 29	30			33[3 Tax	-	36	37 3	н 39	40 4	1 42	4.3	44 4	5 46	47	48 4	5 50	η 51	52	53	54 5	5   6	/b 5	/ 5	8 5	9 6	u t	1 6	2 6	3 6	4

Рисунок 3.8. Трасса неоптимизированной программы

## Оптимизация программы

Для выполнения команды  $1w \times 2$ , 0(x1) необходимо 3 такта, так как происходит обращение к памяти. Только по прошествии этих 3-ёх тактов можно будет работать с регистром x2. Для оптимизации программы можно поместить команду инкремента счётчика цикла ( $addi \times 1, \times 1, elem_sz*enroll$ ) между командами, меняющими регистр x2:  $1w \times 2$ , 0(x1) и  $add \times 31$ , x31, x2.

В результате изменений, время выполнения уменьшилось с 65 до 58 тактов, предположение оказалосьверным.

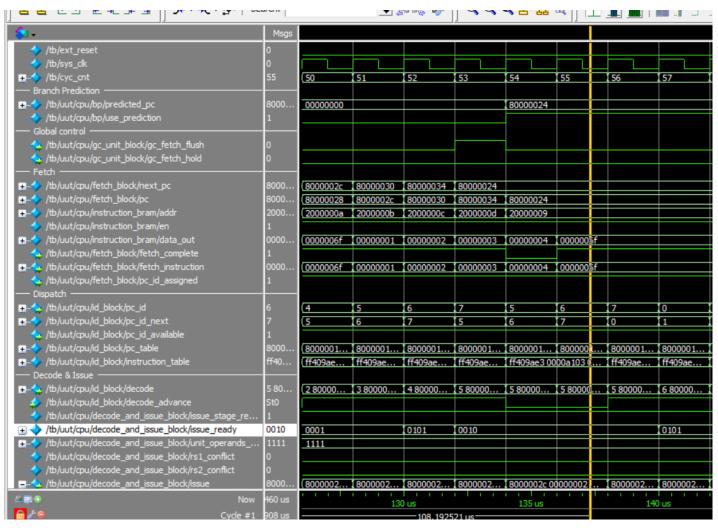


Рисунок 3.9. Завершение программы с оптимизацией

Листинг 2.1. Оптимизированный вариант программы.

```
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. add x31, x0, x0
11. lp:
12.
           lw x2, \theta(x1)
13.
         addi x1, x1, elem_sz*enroll
14.
           add x31, x31, x2 #!
15.
           bne x1, x20, lp
16.
           addi x31, x31, 1
17. lp2: j lp2
18.
19.
           .section .data
20. x:
           .4byte 0x1
21.
         .4byte 0x2
22.
           .4byte 0x3
23.
           .4byte 0x4
24.
           .4byte 0x5
25.
           .4byte 0x6
26.
           .4byte 0x7
27.
           .4byte 0x8
```

## Листинг 2.1. Дизассемблерный листинг оптимизированной программы

```
1. myTest2.elf: file format elf32-littleriscv
2.
3. SYMBOL TABLE:
4. 80000000 1
                d .text
                          00000000 .text
5. 80000028 1 d .data
                          00000000 .data
6. 00000000 1
                 df *ABS*
                          00000000 myTest2.o
7. 00000008 1
              *ABS*
                          00000000 len
8. 00000001 1
                   *ABS*
                          00000000 enroll
9. 00000004 1
                  *ABS*
                          00000000 elem sz
10. 80000028 1
                   .data
                          00000000 x
11. 80000010 1
                   .text
                          00000000 lp
12. 80000024 1
                    .text
                          00000000 lp2
13. 80000000 g
                          00000000 _start
                  .text
14. 80000048 g
                   .data
                          00000000 end
15.
16.
17.
18. Disassembly of section .text:
19.
20. 80000000 <_start>:
```

```
21. 80000000:
                    00000097
                                                     x1,0x0
                                             auipc
22. 80000004:
                    02808093
                                             addi
                                                     x1,x1,40 # 80000028 <_x>
23. 80000008:
                    01c08a13
                                             addi
                                                     x20,x1,28
24. 8000000c:
                    00000fb3
                                             add
                                                     x31,x0,x0
25.
26. 80000010 <lp>:
27. 80000010:
                                                     x2,0(x1)
                    0000a103
28. 80000014:
                    00408093
                                                     x1,x1,4
                                             addi
29. 80000018:
                    002f8fb3
                                                     x31,x31,x2
                                             add
30. 8000001c:
                    ff409ae3
                                             bne
                                                     x1,x20,80000010 <lp>
31. 80000020:
                    001f8f93
                                             addi
                                                     x31,x31,1
32.
33. 80000024 <1p2>:
34. 80000024:
                    0000006f
                                                     x0,80000024 <1p2>
                                             jal
35.
36. Disassembly of section .data:
37.
38. 80000028 <_x>:
39. 80000028:
                    0001
                                             c.addi x0,0
40. 8000002a:
                    0000
                                             c.unimp
41. 8000002c:
                    0002
                                             c.slli64
42. 8000002e:
                    0000
                                             c.unimp
43. 80000030:
                    00000003
                                             lb x0,0(x0) # 0 <enroll-0x1>
44. 80000034:
                    0004
                                             .2byte 0x4
45. 80000036:
                    0000
                                             c.unimp
46. 80000038:
                    0005
                                             c.addi x0,1
47. 8000003a:
                    0000
                                             c.unimp
48. 8000003c:
                    0006
                                             c.slli x0,0x1
49. 8000003e:
                    0000
                                             c.unimp
50. 80000040:
                    00000007
                                             .4byte 0x7
51. 80000044:
                    8000
                                             .2byte 0x8
```

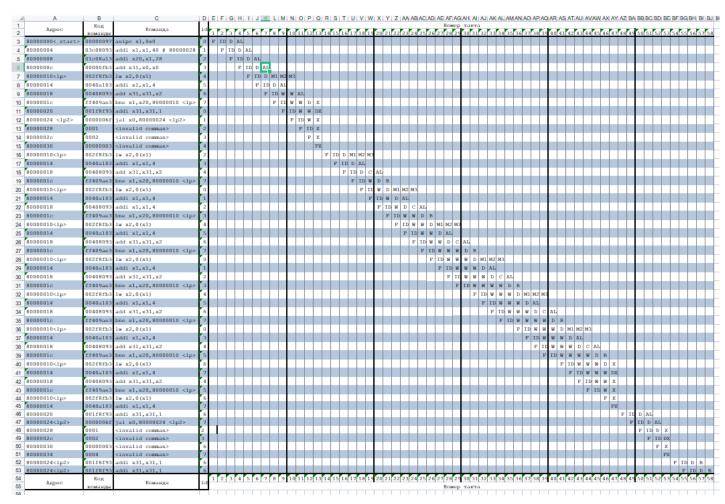


Рисунок 3.10. Трасса оптимизированной программы

# Выводы

Выполнение лабораторной работы позволило ознакомиться с принципами функционирования, построения и особенностями архитектуры суперскалярных конвейерных микропроцессоров. В ходе выполнения лабораторной работы были изучены основны микроархитектуры ядра Taiga. Были получены временные диаграммы стадий команд конвейера Taiga. Также удалось оптимизировать программу, уменьшив время выполнения с 65 до 58 тактов.