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»

Отчёт

по лабораторной работе №2

Название: Изучение принципов работы микропроцессорного ядра RISC-V

Дисциплина: Архитектура ЭВМ

Студент ИУ7-54Б _____ Л.Е.Тартыков

(Группа) (Подпись, дата) (И.О. Фамилия)

Преподаватель _____ А.Ю.Попов

(Подпись, дата) (И.О. Фамилия)

Москва, 2021

Цель работы

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.

1 Основные теоретические сведения

RISC-V является открытым современным набором команд, который может использоваться для построения как микроконтроллеров, так и высокопроизводительных микропроцессоров. Таким образом, термин RISC-V фактически является названием для семейства различных систем команд, которые строятся вокруг базового набора команд, путем внесения в него различных расширений.

В данной работе исследуется набор команд RV32I, который включает в себя основные команды 32-битной целочисленной арифметики кроме умножения и деления.

1.1 Регистровая модель

Набор команд RV32I предполагает использование 32 регистров общего назначения x0-x31 размером в 32 бита каждый и регистр pc, хранящего адрес следующей команды. Все регистры общего назначения равноправны, в любой команде могут использоваться любые из регистров. Регистр pc не может использоваться в командах.

1.2 Модель памяти

Архитектура RV32I предполагает плоское линейное 32-х битное адресное пространство. Минимальной адресуемой единицей информации является 1 байт. Используется порядок байтов от младшего к старшему (Little Endian), то есть, младший байт 32-х битного слова находится по младшему адресу (по смещению 0). Отсутствует разделение на адресные пространства команд, данных и ввода-вывода. Распределение областей памяти между различными устройствами (ОЗУ, ПЗУ, устройства ввода-вывода) определяется реализацией.

1.3 Система команд

Большая часть команд RV32I является трехадресными, выполняющими операции над двумя заданными явно операндами, и сохраняющими результат в регистре. Операндами могут являться регистры или константы, явно заданные в коде команды. Операнды всех команд задаются явно.

Архитектура RV32I, как и большая часть RISC-архитектур, предполагает разделение команд на команды доступа к памяти (чтение данных из памяти в регистр или запись данных из регистра в память) и команды обработки данных в регистрах.

2 Программа по индивидуальному варианту

Задания выполнялись по варианту 16.

2.1 Задание №1

Создать новый файл, содержащий текст программы по индивидуальному варианту. Псевдокод, соответствующий данной программе, представлен на листинге 2.1.

Листинг 2.1 – Исходный текст исследуемой программы

```
1       .section .text
2       .globl _start;
3       len = 9;
4       enroll = 1;
5       elem_sz = 4;
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      add x1, x1, elem_sz*enroll
15      bltu x2, x31, lt
16      add x31, x0, x2 #!
17  lt:
18      addi x20, x20, -1
19      bne x20, x0, lp
20      lp2: j lp2
21
22      .section .data
23  _x:
24      .4byte 0x1
25      .4byte 0x2
26      .4byte 0x3
```

```
27      .4byte 0x4
28      .4byte 0x8
29      .4byte 0x6
30      .4byte 0x7
31      .4byte 0x5
32      .4byte 0x4
```

Дизассемблированный код данной программы представлен на листинге 2.2.

Листинг 2.2 – Дизассемблированный код

```
1  Disassembly of section .text:
2
3  80000000 <_start>:
4  80000000:       00000097            auipc   x1,0x0
5  80000004:       03008093            addi    x1,x1,48 # 80000030
       <_x>
6  80000008:       00800a13            addi    x20,x0,8
7  8000000c:       0000af83            lw      x31,0(x1)
8  80000010:       00408093            addi    x1,x1,4
9
10 80000014 <lp>:
11 80000014:       0000a103            lw      x2,0(x1)
12 80000018:       00408093            addi    x1,x1,4
13 8000001c:       01f16463            bltu    x2,x31,80000024 <lt>
14 80000020:       00200fb3            add     x31,x0,x2
15
16 80000024 <lt>:
17 80000024:       fffa0a13            addi    x20,x20,-1
18 80000028:       fe0a16e3            bne     x20,x0,80000014 <lp>
19
20 8000002c <lp2>:
21 8000002c:       0000006f            jal     x0,8000002c <lp2>
22
23 Disassembly of section .data:
24
25 80000030 <_x>:
26 80000030:       0001                c.addi  x0,0
27 80000032:       0000                unimp
28 80000034:       0002                0x2
29 80000036:       0000                unimp
30 80000038:       00000003            lb      x0,0(x0) # 0 <enroll
```

```
    -0x1>
31 8000003c:        0004                    c.addi4spn      x9,x2,0
32 8000003e:        0000                    unimp
33 80000040:        0008                    c.addi4spn      x10,x2,0
34 80000042:        0000                    unimp
35 80000044:        0006                    0x6
36 80000046:        0000                    unimp
37 80000048:        00000007                0x7
38 8000004c:        0005                    c.addi  x0,1
39 8000004e:        0000                    unimp
40 80000050:        0004                    c.addi4spn      x9,x2,0
```

Код, поясняющий работу данной программы представлен на листинге 2.3.

Листинг 2.3 – Дизассемблированный код

```
1 #define len 9
2 #define enroll 1
3 #define elem_sz 4
4
5 int _x[] = {1,2,3,4,8,6,7,5,4};
6 void _start()
7 {
8     int *x1 = _x;
9     int x31 = x1[0];
10    int x20 = (len - 1) / enroll;
11    int x1 = elem_sz * 1;
12
13    do
14    {
15        int x2 = x1[0];
16        x1 += elem_sz * enroll;
17        if (x2 >= x31){
18            do{
19                x20 -= 1;
20            }
21            while (x2 != x0)
22        }
23    }
24    while(x2 < x31)
25    x31 = x0 + x2;
26    while(1){}
```

```
27 }
```

2.2 Задание №2

Задание: получить снимок экрана, содержащий временную диаграмму выполнения стадий выборки и диспетчеризации команды с адресом 8000002c на первой итерации.

Результат представлен на рисунке 2.1

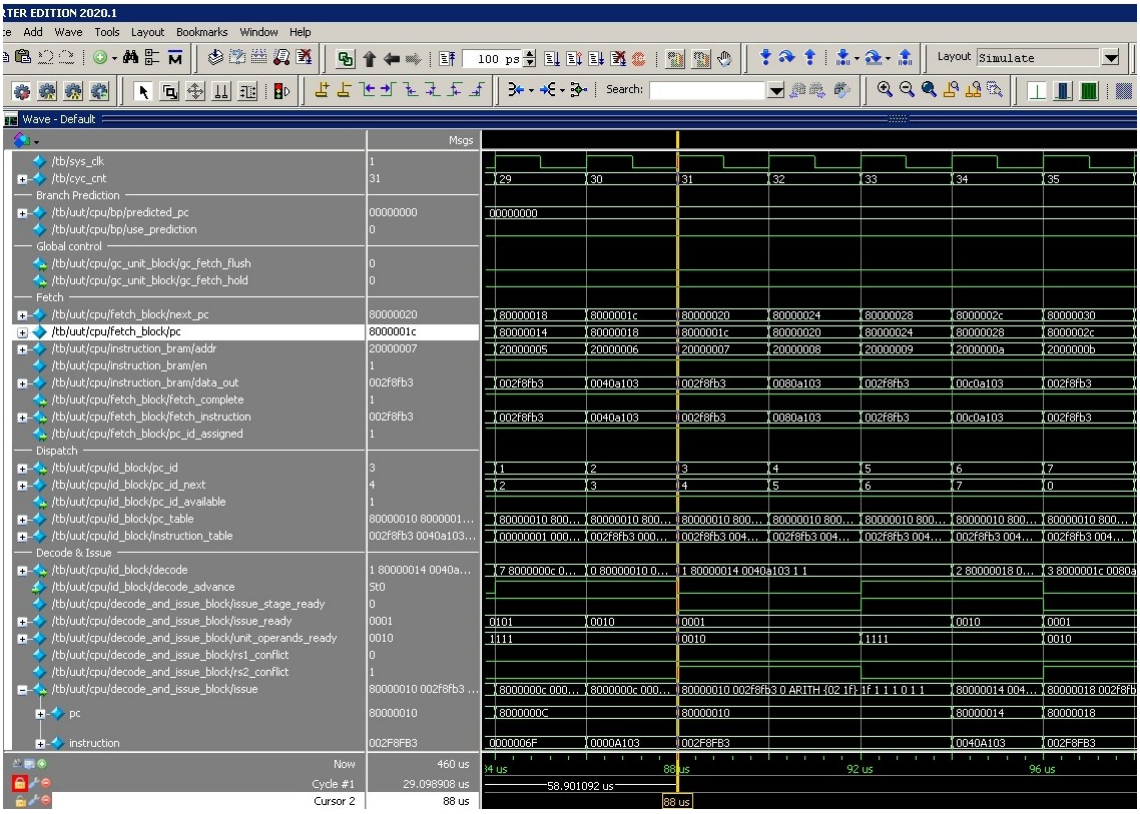

Рисунок 2.1 – Временная диаграмма выполнения стадий выборки и диспетчеризации команды с адресом 8000002c на первой итерации

2.3 Задание №3

Задание: получить снимок экрана, содержащий временную диаграмму выполнения стадии декодирования и планирования на выполнение команды с адресом 8000000c на второй итерации.

Результат представлен на рисунке 2.2

Рисунок 2.2 – Временная диаграмма выполнения стадии декодирования и планирования на выполнение команды с адресом 8000000c на второй итерации

2.4 Задание №4

Задание: получить снимок экрана, содержащий временную диаграмму выполнения стадии выполнения команды с адресом 80000020 на первой итерации.

Результат представлен на рисунке 2.3.

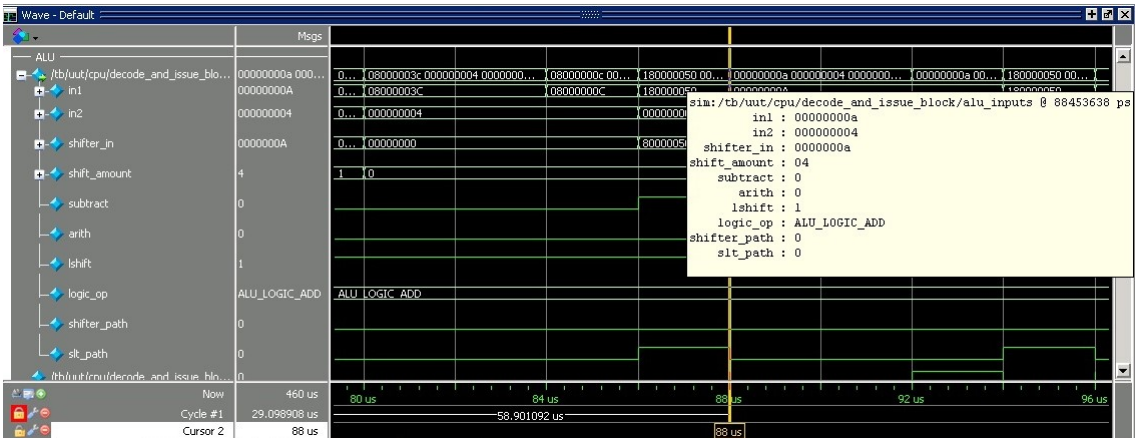

Рисунок 2.3 – Временная диаграмма выполнения стадии выполнения команды с адресом 80000020 на первой итерации

2.5 Задание №5

Задание: получить временную диаграмму сигналов выполнения программы индивидуального варианта. Провести оптимизацию программы путем перестановки команд для устранения конфликтов. Заполнить трассу выполнения оптимизированной программы.

2.5.1 Оптимизация

Конфликт происходит по регистру x20, значение которого изменяется уже после того, как выполнилась обработка над элементом массива. Предлагается сначала уменьшать число оставшихся, необработанных элементов, затем выполнять непосредственно обработку.

Код оптимизированной программы по устранению конфликтов представлен на листинге 2.4.

Листинг 2.4 – Исходный текст исследуемой программы

```
1     .section .text
2     .globl _start;
3     len = 9
4     enroll = 1
5     elem_sz = 4
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    add x1, x1, elem_sz*enroll
15    addi x20, x20, -1
16    bltu x2, x31, lt
17    add x31, x0, x2 #!
18 lt:
19    bne x20, x0, lp
20 lp2: j lp2
```

```
21
22      .section .data
23      _x:     .4byte 0x1
24      .4byte 0x2
25      .4byte 0x3
26      .4byte 0x4
27      .4byte 0x8
28      .4byte 0x6
29      .4byte 0x7
30      .4byte 0x5
31      .4byte 0x4
```

Дизассемблированный код данной программы представлен на листинге 2.5.

Листинг 2.5 – Дизассемблированный код

```
1  Disassembly of section .text:
2
3  80000000 <_start>:
4  80000000:       00000097            auipc   x1,0x0
5  80000004:       03008093            addi    x1,x1,48 # 80000030
       <_x>
6  80000008:       00800a13            addi    x20,x0,8
7  8000000c:       0000af83            lw      x31,0(x1)
8  80000010:       00408093            addi    x1,x1,4
9
10 80000014 <lp>:
11 80000014:       0000a103            lw      x2,0(x1)
12 80000018:       00408093            addi    x1,x1,4
13 8000001c:       fffa0a13            addi    x20,x20,-1
14 80000020:       01f16463            bltu    x2,x31,80000028 <lt>
15 80000024:       00200fb3            add     x31,x0,x2
16
17 80000028 <lt>:
18 80000028:       fe0a16e3            bne     x20,x0,80000014 <lp>
19
20 8000002c <lp2>:
21 8000002c:       0000006f            jal     x0,8000002c <lp2>
22
23 Disassembly of section .data:
24
25 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
    -0x1>
8000003c:       0004                    c.addi4spn      x9,x2,0
8000003e:       0000                    unimp
80000040:       0008                    c.addi4spn      x10,x2,0
80000042:       0000                    unimp
80000044:       0006                    0x6
80000046:       0000                    unimp
80000048:       00000007                0x7
8000004c:       0005                    c.addi  x0,1
8000004e:       0000                    unimp
80000050:       0004                    c.addi4spn      x9,x2,0
```

Трасса исходной программы представлена на рисунке 2.4.

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	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82
80000000 <_start>:	00000097	auipc x1,0x0	0	F	ID	D	AL																																																																														
80000004	03000093	addi x1,x1,48	1		F	ID	D	AL																																																																													
80000008	00800a13	addi x20,x0,8	2			F	ID	D	AL																																																																												
8000000c	0000af83	lw x31,0(x1)	3				F	ID	D	M1	M2	M3																																																																								
80000010	00408093	addi x1,x1,4	4					F	ID	D	AL																																																																										
80000014 <lp>:	0000a103	lw x2,0(x1)	5						F	ID	D	M1	M2	M3																																																																					
80000018	00408093	addi x1,x1,4	6							F	ID	D	AL																																																																								
8000001c	01f16463	bltu x2,x31,80000024 <lt>	7								F	ID	D	C	B																																																																					
80000020	00200fb3	add x31,x0,x2	0									F	ID	W	D	AL																																																																				
80000024 <lt>:	fffa0a13	addi x20,x20,-1	1										F	ID	W	D	AL																																																																			
80000028	fe0a16e3	bne x20,x0,80000014 <lp>	2											F	ID	W	D	B																																																																		
8000002c <lp2>:	0000006f	jal x0,8000002c <lp2>	3												F	ID	W	D	X																																																																	
80000030	00000001	<invalid operation>	4													F	ID	W	DX																																																																	
80000034	00000002	<invalid operation>	3														F	ID	X																																																																	
80000038	00000003	<invalid operation>	4															F	X																																																																	
8000003c <lp>:	00000004	<invalid operation>	5																FX																																																																	
80000014	0000a103	lw x2,0(x1)	6																F	ID	D	M1	M2	M3																																																											
80000018	00408093	addi x1,x1,4	7																	F	ID	D	AL																																																												
8000001c	01f16463	bltu x2,x31,80000024 <lt>	0																		F	ID	D	C	B																																																										
80000020	00200fb3	add x31,x0,x2	1																			F	ID	W	D	AL																																																									
80000024	fffa0a13	addi x20,x20,-1	2																				F	ID	W	D	AL																																																								
80000028 <lp>:	fe0a16e3	bne x20,x0,80000014 <lp>	3																					F	ID	W	D	B																																																							
80000014	0000a103	lw x2,0(x1)	4																						F	ID	W	D	M1	M2	M3																																																				
80000018	00408093	addi x1,x1,4	5																							F	ID	W	D	AL																																																					
8000001c	01f16463	bltu x2,x31,80000024 <lt>	6																								F	ID	W	D	C	B																																																			
80000020	00200fb3	add x31,x0,x2	7																									F	ID	W	W	D	AL																																																		
80000024	fffa0a13	addi x20,x20,-1	0																										F	ID	W	W	D	AL																																																	
80000028 <lp>:	fe0a16e3	bne x20,x0,80000014 <lp>	1																											F	ID	W	W	D	B																																																
80000014	0000a103	lw x2,0(x1)	2																												F	ID	W	W	D	M1	M2	M3																																													
80000018	00408093	addi x1,x1,4	3																													F	ID	W	W	D	AL																																														
8000001c	01f16463	bltu x2,x31,80000024 <lt>	4																														F	ID	W	W	D	C	B																																												
80000020	00200fb3	add x31,x0,x2	5																															F	ID	W	W	W	D	AL																																											
80000024	fffa0a13	addi x20,x20,-1	6																																F	ID	W	W	W	D	AL																																										
80000028 <lp>:	fe0a16e3	bne x20,x0,80000014 <lp>	7																																	F	ID	W	W	W	D	B																																									
80000014	0000a103	lw x2,0(x1)	0																																		F	ID	W	W	W	D	M1	M2	M3																																						
80000018	00408093	addi x1,x1,4	1																																			F	ID	W	W	W	D	AL																																							
8000001c	01f16463	bltu x2,x31,80000024 <lt>	2																																					F	ID	W	W	D	C	B																																						
80000020	00200fb3	add x31,x0,x2	3																																						F	ID	W	W	W	D	X																																					
80000024	fffa0a13	addi x20,x20,-1	4																																							F	ID	W	W	W	DX																																					
80000028 <lp>:	fe0a16e3	bne x20,x0,80000014 <lp>	5																																								F	ID	W	W	X																																					
80000024	fffa0a13	addi x20,x20,-1	6																																													F	ID	D	AL																																
80000028 <lp>:	fe0a16e3	bne x20,x0,80000014 <lp>	7																																														F	ID	D	B																															
80000014	0000a103	lw x2,0(x1)	0																																															F	ID	D	M1	M2	M3																												
80000018	00408093	addi x1,x1,4	1																																																F	ID	D	AL																													
8000001c	01f16463	bltu x2,x31,80000024 <lt>	2																																																	F	ID	D	C	B																											
80000020	00200fb3	add x31,x0,x2	3																																																		F	ID	W	D	X																										
80000024	fffa0a13	addi x20,x20,-1	4																																																			F	ID	W	DX																										
80000028 <lp>:	fe0a16e3	bne x20,x0,80000014 <lp>	5																																																				F	ID	X																										
80000024	fffa0a13	addi x20,x20,-1	6																																																							F	ID	D	AL																						
80000028 <lp>:	fe0a16e3	bne x20,x0,80000014 <lp>	7																																																								F	ID	D	B																					
80000014	0000a103	lw x2,0(x1)	0																																																									F	ID	D	M1	M2	M3																		
80000018	00408093	addi x1,x1,4	1																																																										F	ID	D	AL																			
8000001c	01f16463	bltu x2,x31,80000024 <lt>	2																																																											F	ID	D	C	B																	
80000024	fffa0a13	addi x20,x20,-1	3																																																												F	ID	W	D	AL																
80000028 <lp>:	fe0a16e3	bne x20,x0,80000014 <lp>	4																																																													F	ID	W	D	B															
80000014	0000a103	lw x2,0(x1)	5																																																														F	ID	W	D	M1	M2	M3												
80000018	00408093	addi x1,x1,4	6																																																															F	ID	W	D	AL													
8000001c	01f16463	bltu x2,x31,80000024 <lt>	7																																																																F	ID	W	D	C	B											
80000020	00200fb3	add x31,x0,x2	0																																																																	F	ID	W	W	D	AL										
80000028 <lp>:	fe0a16e3	bne x20,x0,80000014 <lp>	1																																																																		F	ID	W	W	D	B									
80000014	0000a103	lw x2,0(x1)	2																																																																			F	ID	W	W	D	X								
80000018	00408093	addi x1,x1,4	3																																																																				F	ID	W	W	DX								
8000001c	01f16463	bltu x2,x31,80000024 <lt>	4																																																																					F	ID	W	X								
80000020	00200fb3	add x31,x0,x2	5																																																																						F	ID	X								
80000028 <lp>:	fe0a16e3	bne x20,x0,80000014 <lp>	6																																																																							F	X								
80000014	0000a103	lw x2,0(x1)	7																																																																								FX								
8000002c <lp2>:	0000006f	jal x0,8000002c <lp2>	0																																																																									F	ID	D	B				
80000030	00000001	<invalid operation>	1																																																																										F	ID	D	X			
80000034	00000002	<invalid operation>	2																																																																											F	ID	DX			
80000038	00000003	<invalid operation>	3																																																																												F	X			
8000003c <lp>:	00000004	<invalid operation>	4																																																																													FX			
8000002c <lp2>:	0000006f	jal x0,8000002c <lp2>	5																																																																														F	ID	D	B

Рисунок 2.4 – Трасса выполнения программы

Трасса оптимизированной программы представлена на рисунке 2.4.

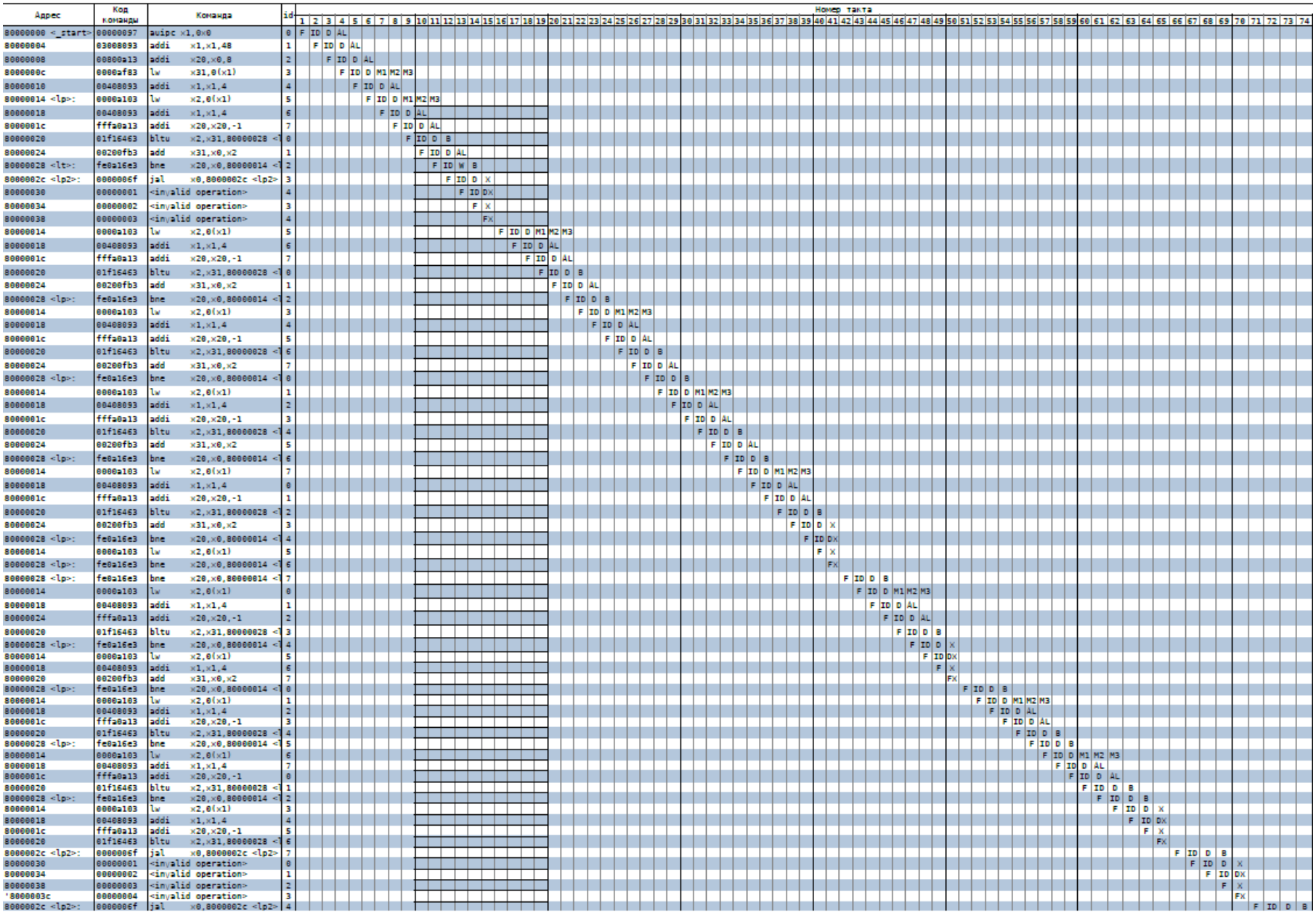

Рисунок 2.5 – Трасса выполнения программы (оптимизированная)

Вывод

В результате выполнения лабораторной работы были изучены принципы функционирования, построения и особенности архитектуры суперскалярных конвейерных микропроцессоров.

Также были рассмотрены принципы проектирования и верификации сложных цифровых устройств с использованием языка описания аппаратуры SystemVerilog и ПЛИС.

На основе изученных материалов был найден способ оптимизации программы.

Поставленная цель достигнута.