

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 1	микропроцессорного	ядра RISC-V»
Дисциплина	«Архитектура ЭВМ»		
Вариант	3		
Студент	ИУ7-55Б		Клименко А.К.
		(подпись, дата)	(Фамилия И.О.)
Преподовате	пь		Ибрагимов С.В.
		(подпись, дата)	(Фамилия И.О.)

Содержание

BB	ведение			3
1	Теори	тически	й раздел	2
	1.1	Архит	гектура набора команд RV32I	2
		1.1.1	Регистровая модель	2
		1.1.2	Модель памяти	2
		1.1.3	Система команд	2
2	Экспе	римента	льный раздел	6
	2.1	Иссле	едование программы	(
		2.1.1	Исходный код программы	6
		2.1.2	Дизассемблированный листинг	7
		2.1.3	Псевдокод	8
		2.1.4	Трасса работы программы	12
		2.1.5	Выводы об эффективности	12
	2.2	Оптим	мизация программы	13
		2.2.1	Исходный код оптимизированной программы	13
		2.2.2	Дизассемблированный листинг оптимизированной программы	14
		2.2.3	Трасса работы оптимизированной программы	15
		2.2.4	Выводы	15
За	ключені	ие		16

Введение

Основной 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.

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

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

Для работы с проектом используется САПР Intel Quartus.

1 Теоритический раздел

RISC-V является открытым современным набором команд, который может использоваться для построения как микроконтроллеров, так и высокопроизводительных микропроцессоров. В связи с такой широкой областью применения в систему команд введена вариативность. 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. В рамках данного набора команд мы не будем рассматривать системные команды, связанные с таймерами, системными регистрами, управлением привилегиями, прерываниями и исключениями.

1.1 Архитектура набора команд RV32I

В настоящем разделе описывается архитектура набора команд, то есть архитектура абстрактной вычислительной машины с точки зрения набора команд без связи с конкретной аппаратной реализацией.

1.1.1 Регистровая модель

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

1.1.2 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.

1.1.3 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 (кроме команды ацірс) 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.

Таким образом, команды RV32I можно разделить на следующие категории:

- 1) Команды обработки данных
 - а) Арифметические и логические команды
 - б) Команды сравнения
 - в) Команды сдвига
 - г) Команды формирования значения в старшей части регистра
- 2) Команды передачи управления
 - а) Команды безусловного перехода с сохранением адреса возврата
 - б) Команды условного перехода
- 3) Команды доступа к памяти
 - а) Команды загрузки
 - б) Команды сохранения
- 4) Системные команды

2 Экспериментальный раздел

2.1 Исследование программы

2.1.1 Исходный код программы

```
.section .text
 1
 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:
            la x1, _x
 8
 9
            addi x20, x1, elem_sz*(len-1) # Адрес последнего элемента
10
    lp:
            lw x2, 0(x1)
11
            add x31, x31, x2 \#!
12
            addi x1, x1, elem_sz*enroll
13
14
            bne x1, x20, lp
15
            addi x31, x31, 1
16
    1p2:
            j 1p2
17
18
            .section .data
19
            .4byte 0x1
    _x:
20
            .4byte 0x2
21
            .4byte 0x3
22
            .4byte 0x4
23
            .4byte 0x5
            .4byte 0x6
24
25
            .4byte 0x7
26
            .4byte 0x8
```

2.1.2 Дизассемблированный листинг

```
Disassembly of section .text:
 2
 3
    80000000 <_start>:
    80000000:
                     00000097
 4
                                               auipc
                                                       x1,0x0
 5
    80000004:
                     02408093
                                               addi
                                                       x1,x1,36 # 80000024 <_x>
    80000008:
                     01c08a13
                                               addi
                                                       x20,x1,28
 7
    8000000c <lp>:
 8
                                                       x2,0(x1)
 9
   800000c:
                     0000a103
                                              lw
    80000010:
                     002f8fb3
                                                       x31,x31,x2
10
                                               add
    80000014:
                     00408093
                                               addi
11
                                                       x1, x1, 4
    80000018:
                                                       x1,x20,8000000c <1p>
12
                     ff409ae3
                                              bne
   800001c:
                                                       x31,x31,1
13
                     001f8f93
                                               addi
14
15
    80000020 <1p2>:
16
    80000020:
                     000006f
                                               jal
                                                       x0,80000020 <1p2>
17
   Disassembly of section .data:
18
19
20
   80000024 <_x>:
21
   80000024:
                     0001
                                               c.addi x0,0
22
    80000026:
                     0000
                                              unimp
23
   80000028:
                     0002
                                               0x2
    8000002a:
                     0000
24
                                              unimp
    8000002c:
                     0000003
                                              1b
25
                                                       x0,0(x0) # 0 < enroll - 0x1 >
   80000030:
                     0004
                                              c.addi4spn
                                                                x9,x2,0
26
27
    80000032:
                     0000
                                              unimp
28
    80000034:
                     0005
                                              c.addi x0,1
29
    80000036:
                     0000
                                              unimp
30
    80000038:
                     0006
                                               0x6
31
   8000003a:
                     0000
                                              unimp
32
   8000003c:
                     0000007
                                               0x7
   80000040:
                     8000
33
                                               c.addi4spn
                                                                x10,x2,0
```

2.1.3 Псевдокод

Листинг 2.1 — Эквивалентный код на С.

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

Задание 2.

Получить снимок экрана, содержащий временную диаграмму выполнения стадий выборки и диспетчеризации команды с указанным адресом.

Адрес команды: 80000014, номер итерации: 1.

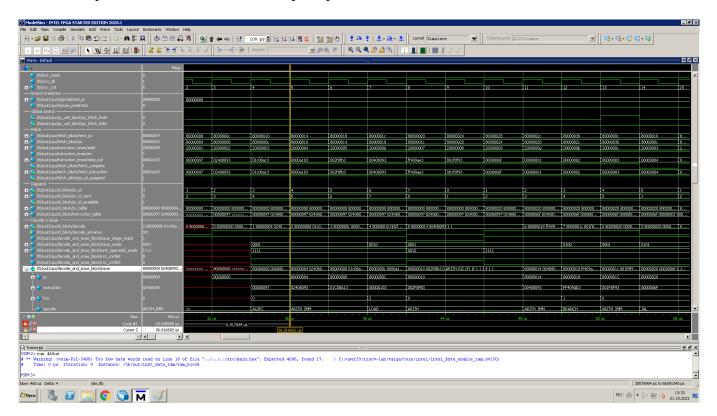


Рисунок 2.1 — Скриншот для задания 2

Задание 3.

Получить снимок экрана, содержащий временную диаграмму выполнения стадии декодирования и планирования на выполнение команды с указанным адресом.

Адрес команды: 80000020, номер итерации: 1.

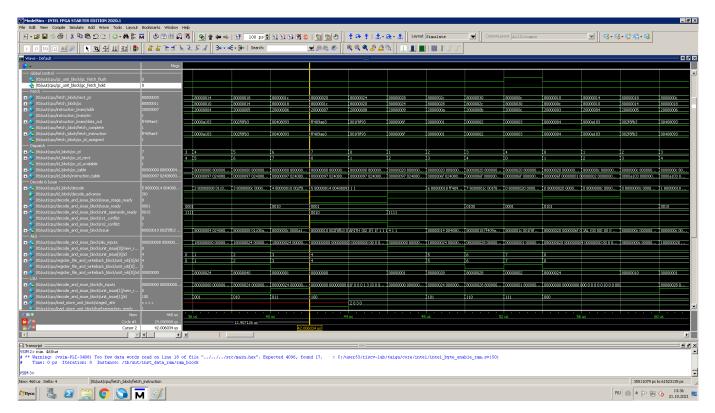


Рисунок 2.2 — Скриншот для задания 3

Задание 4.

Получить снимок экрана, содержащий временную диаграмму выполнения стадии выполнения команды с указанным адресом.

Адрес команды: 80000008.

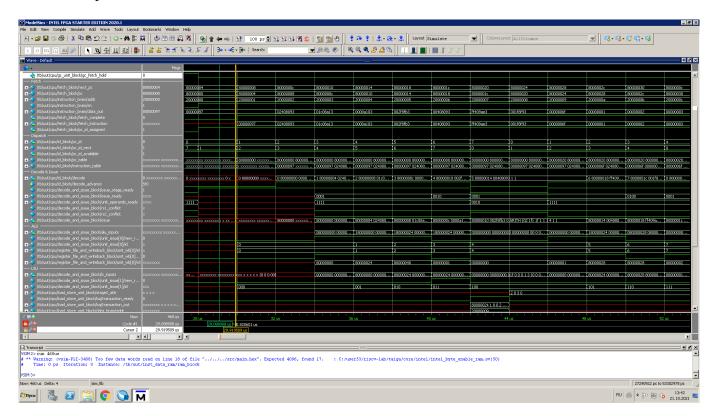


Рисунок 2.3 — Скриншот для задания 4

2.1.4 Трасса работы программы

Composed	Апрес	Код	Команда	номер такта
1 Fid Part	1.4		,,	
Second Compose				
Second Composed Section				
Second S	80000008	01c08a13	addi x20,x1,28#80000024<_x>	2
Second Composed Statistics Second Statis		0000a103	lw x2,0(x1)	3 F ID D M1N2 M3
Second Composed Statistics Second Statis	80000010	002f8fb3	add x31,x31,x2	4
Second Column		00408093	addi x1,x1,4	5 FID W W D AL
Second Column	80000018	ff409ae3	bne x1.x20.8000000c <lp></lp>	is Fitow Win B
Second Company of the Company of t	8999991c	001f8f93	addi x31.x31.1	7 FIDWWD X
Second Company				
Second Compose				
Second Company Seco				
Second S				
Second Colored Second Colored Colore				
Second Color Col				
Seminor Companies Compan				
Second S				
80000014 900000130 brd; 1.1.1.4 5 FID W W W W D AL				
\$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,000,000 \$1,				
\$10000012 \$1,20,3000000cclps				
80000018 002/8F33 add x31,x31,x2 8 8 8 8 8 8 8 8 8	80000018			(6
80000018 002/8F33 add x31,x31,x2 8 8 8 8 8 8 8 8 8	8000000c <lp></lp>	0000a103	lw x2,0(x1)	77
Second S	80000010			
### ### ### ### ### ### ### ### ### ##				
Second S				
800000012 002/f613 add x31,x21,x2 4 000000013 1x,x1,4 5 6 00000013 1x,x1,4 5 7 80000013 1x,x1,4 5 7 80000013 1x,x1,4 1 1 1 1 1 1 1 1 1				
B0000011	000000000			
### PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND ALL STREET OF THE PROVIDED TO BROWN BENEFITS AND AL				
Second S				
Second S				
90000014 9040993 addi x1,x1.4 1 1				
### 10 M W W W W W D B ### 10 M W W W W W D B ### 10 M D M W D W W D M D M D M D M D M D M D				
Second S				
B0000001 002/07/15 and x31,x31,x2				
00000014 00400093 ddi x1,x1,4 5 5 5 5 5 5 5 5 5				3
### 1989-0913				
Second S				
98000014 9024878 3 add x31,x1,x2 9 1 98000015 9024878 3 add x31,x31,x2 9 1 98000016 9024878 3 add x31,x31,x2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	80000018	ff409ae3	bne x1,x20,8000000c <lp></lp>	6
Beassestant				
00000011	80000010	002f8fb3	add x31,x31,x2	9 FIDWWWWDCCAL
## 100 ## ## ## ## ## ## ## ## ## ## ## ## #				
98000014 902.7887.3				
98090010 902/8783 add x31,x1,x2 4 5 FID W W W W W D C CAL 980900101				
S0000014	80000010			
### ### ### ### ### ### ### ### ### ##				
90000014 00408903 doi: 1,1,1,4 0				
80000014 0048093 dof x1,x1,x2 0 FID W W W M XX 80000015 FID W W W M XX 80000016 FID W W W M X 8000016 FID W W W M X 8000016 FID W W W M X 8000016 FID W W W M X 8000016 FID W W W W W M X 8000016 FID W W W W W M X 8000016 FID W W W W W M X 8000016 FID W W W W W W W W W W M X 8000016 FID W W W W W M X 8000016 FID W W W W W M X 8000016 FID W W W W W M X 800016 FID W W W W W W W X 800016 FID W W W W W W W W W W W X 800016 FID W W W W W W X 800016 FID W W W W W X 800016				
S0000014				
80000018	00000010			
	R8688818	ff400093	hne x1.x20.8888888c<1n>	
Second S	8000000c <lp></lp>	0000a103	lw x2.0(x1)	(3)
S0000011	80000010	002f8fb3	add x31,x31,x2	[4]
Second S	80000014	00408093	addi x1,x1,4	
80000024 90000001 kinvalid operations 2 FID 0 > 80000024 FID 0 PROPERTY	B000001c			
	80000020	00000006f	jal x0,80000020 <lp2></lp2>	12
8800802 808090833 invalid operation 4	86666674			F IDD X
-64890024 00000004 invalid operation 5 F 80000026 00000006 in x 0,0000007 in x 0,0000007 20 20 20 20 20 20	00000020	000000002	<invalid operation=""></invalid>	
88000020 80000006f jal x0,8000020<-lp2 - forever 3 80000020 - 00000006f jal x0,8000020<-lp2 - forever 4 4 80000020 - 00000006f jal x0,8000020<-lp2 - forever 4 4 80000020 - 00000006f jal x0,80000020<-lp2 - forever 4 4 8000000000000000000000000000000000				F A
89090020 90090096fjal x8,98090024cjp2 < foreign file	80000020	0000006f	ial x0,80000020 <lp2> <forevers< td=""><td></td></forevers<></lp2>	
	80000020		jal x0,80000020 <lp2> <forever></forever></lp2>	
Aprel Komannia Nomanda 149 Homen Takta	Annec	Код	Команда	1 2 3 4 5 6 7 8 9 10111213141516171819202122232425262728293031323334553637383940414243444546474849505152535455565788596061626364656667686970717273747574
	идрес	команды	поманда	Номер такта

2.1.5 Выводы об эффективности

Из рассмотрения трассы видно, что выполнение программы до окончания команды addi x31,x31,1 заняло 69 тактов (1-69). Из них, в течение 23 тактов (7, 8, 13-15, 17, 18, 23, 24, 29, 30, 35, 36, 41, 42, 47, 48, 53, 54, 60, 65-67) не происходило планирование на выполнение новых команд, то есть, 50% времени было потрачено не эффективно из-за ошибок предсказания ветвлений и конфликтов на конвейере.

Для оптимизации данной программы можно поменять местами команды на строках 12 и 13. После этого конвеер сможет приступить к обработке следующей команды после планирования $1 \times 2.0(x1)$.

2.2 Оптимизация программы

2.2.1 Исходный код оптимизированной 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
   lp:
            lw x2, 0(x1)
11
12
            addi x1, x1, elem_sz*enroll
13
            add x31, x31, x2
            bne x1, x20, lp
14
15
            addi x31, x31, 1
16
    1p2:
            j 1p2
17
18
            .section .data
19
    _x:
            .4byte 0x1
20
            .4byte 0x2
            .4byte 0x3
21
            .4byte 0x4
22
23
            .4byte 0x5
24
            .4byte 0x6
25
            .4byte 0x7
26
            .4byte 0x8
```

2.2.2 Дизассемблированный листинг оптимизированной программы

```
1
        Disassembly of section .text:
 2
 3
        80000000 <_start>:
 4
        80000000:
                         00000097
                                                  auipc
                                                           x1,0x0
 5
        80000004:
                         02408093
                                                           x1,x1,36 # 80000024 <_x>
                                                  addi
        80000008:
                         01c08a13
                                                  addi
                                                           x20, x1, 28
 6
 7
 8
        8000000c <lp>:
                                                           x2,0(x1)
 9
        8000000c:
                         0000a103
                                                  lw
        80000010:
                                                           x1,x1,4
10
                         00408093
                                                  addi
        80000014:
                         002f8fb3
                                                           x31,x31,x2
11
                                                  add
        80000018:
                                                           x1,x20,8000000c <1p>
12
                         ff409ae3
                                                  bne
                                                           x31,x31,1
13
        8000001c:
                         001f8f93
                                                  addi
14
15
        80000020 <1p2>:
        80000020:
16
                         0000006f
                                                  jal
                                                           x0,80000020 <1p2>
17
18
        Disassembly of section .data:
19
20
        80000024 <_x>:
21
        80000024:
                         0001
                                                  c.addi x0,0
22
        80000026:
                         0000
                                                  unimp
23
        80000028:
                         0002
                                                  0x2
24
        8000002a:
                         0000
                                                  unimp
        8000002c:
                         0000003
                                                  lb
25
                                                           x0,0(x0) # 0 < enroll - 0x1 >
26
        80000030:
                         0004
                                                  c.addi4spn
                                                                   x9,x2,0
27
        80000032:
                         0000
                                                  unimp
28
        80000034:
                         0005
                                                  c.addi x0,1
29
        80000036:
                         0000
                                                  unimp
30
        80000038:
                         0006
                                                  0x6
31
        8000003a:
                         0000
                                                  unimp
32
        8000003c:
                         0000007
                                                  0x7
        80000040:
                         8000
33
                                                  c.addi4spn
                                                                   x10,x2,0
```

2.2.3 Трасса работы оптимизированной программы

Адрес	Код	Команда	1d Howep rakta Howep rakta 1 2 3 4 5 6 7 8 9 10 11 22 31 41 51 61 71 81 920 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 64 65 65 65 65 65 65 65 65 65 65 65 65 65
80000000< start>	команды 00000097	auipc x1.0x0	9 F ID D AL
89999994	02408093	addi x1,x1,36	1 F IDD D AL
80000004	01c08a13	addi x1,x1,30 addi x20,x1,28#80000024< x>	2 F IDD D AL
80000008 8000000c <lp></lp>	0000a103	lw x2,0(x1)	2 F 10 0 M1 M2 M3
80000010	00408093	addi x1,x1,4	4 FID DAL
80000014	002f8fb3	add x31,x31,x2	
80000018		bne x1,x20,8000000c <lp></lp>	6 F ID W D B
8000001c	001f8f93		7 F ID W D X
80000020 <1p2>	9999996f	jal x0,80000020 <lp2></lp2>	F ID W DX
89999924	00000001	<invalid operation=""></invalid>	1 FIDX
89999928	00000002	<invalid operation=""></invalid>	2 F X
8000002C	00000003	<invalid operation=""></invalid>	3 FX FX FX FX FX FX FX F
8000000c <lp></lp>	0000a103	lw x2,0(x1)	0
80000010	00408093	addi x1,x1,4	T F ID D AL
80000014	002f8fb3	add x31,x31,x2	12
80000018	ff409ae3	bne x1,x20,8000000c <lp></lp>	3 FIDWD B
8000000c <lp></lp>		lw x2,0(x1)	4 F ID W D M1 M2/M3
80000010	00408093	addi x1,x1,4	F IID W D AL
89999914	002f8fb3	add x31,x31,x2	6 FIDW D C AL
80000014	ff409ae3	bne x1,x20,8000000c <lp></lp>	7 FIDWWD B
80000018 800000018		lw x2,0(x1)	B F IDI W W D IM M2 D M M2 M3
80000000 < 10	00408093	addi x1,x1,4	1 FIDW W D AL
80000010		add x1,x1,4 add x31,x31,x2	2 FIDW W D CAL
80000014	ff409ae3	bne x1,x20,8000000c <lp></lp>	2
8000000c <lp></lp>		lw x2,0(x1)	4 F ID W W W D M1M2M3
80000010	00408093	addi x1,x1,4	5 F ID W W W D AL
80000014	002f8fb3	add x31,x31,x2	6 FIDWWWDCAL
8000018	ff409ae3	bne x1,x20,8000000c <lp></lp>	7 F ID W W W D B
8000000c <lp></lp>		lw x2,0(x1)	0
80000010	00408093	addi x1,x1,4	1 FIDWWWDAL
80990014	002f8fb3	add x31,x31,x2	2
80000018	ff409ae3	bne x1,x20,8000000c <lp></lp>	3
8000000c <lp></lp>	0000a103	lw x2,0(x1)	4 FID W W W D M1 M2 M3
80000010	00408093	addi x1,x1,4	5 FIDW W W D AL
80000014	002f8fb3	add x31,x31,x2	6 FIDWWWDCAL
80000018	ff409ae3	bne x1,x20,8000000c <lp></lp>	7 FDWWWDB
8999999c <lp></lp>	0000a103	lw x2,0(x1)	F ID W W D MIM2M3
80000010	00408093	addi x1,x1,4	1 FIDWWWDAL
89999914	002f8fb3	add x31,x31,x2	12 FIDW W W D C AL
80000014	ff409ae3	bne x1,x20,8000000c <lp></lp>	3 FIDW W W W D B
80000018 800000018	0000a103	lw x2,0(x1)	4 F ID W W W D M1 M2 M3
8000000C <tp></tp>	00408093	addi x1,x1,4	5 F I DW W W D PAL
80000010		add x1,x1,4 add x31,x31,x2	13 F I I DW W W I D PLL
80000014	ff409ae3	bne x1,x20,8000000c <lp></lp>	0
8000000c <lp></lp>		lw x2,θ(x1)	F ID W W W D X
89999919	00408093	addi x1,x1,4	1 F ID W W W DX
80000014		add x31,x31,x2	2 F ID W W X
80000018	ff409ae3	bne x1,x20,8000000c <lp></lp>	3 F ID W X
8000000c <lp>80000010</lp>	0000a103 00408093	lw x2,0(x1) addi x1,x1,4	4 F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F X F
80000010 8000001c	001f8f93	addi x31,x31,1	1 PA
80000010	000110193 00000006f	jal x0,80000020 <lp2></lp2>	12
89999924	00000001	<invalid operation=""></invalid>	13
80000028	00000002	<invalid operation=""></invalid>	14 FIDIX
8999992	00000003	<invalid operation=""></invalid>	5 FX
-64000024	00000004	<invalid operation=""></invalid>	6 FX
80000020	0000006f	jal x0,80000020 <lp2> <forever></forever></lp2>	4 F ID D E
80000020	0000006f	jal x0,80000020 <lp2> <forever></forever></lp2>	5 F. IDI
Адрес	Код	Команда	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 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 33 35 35 35 35 35
, where	команды	s von-nus (paper	Howep TakTa

2.2.4 Выводы

Выполнение оптимизированной программы до окончания каманды addi x31,x31,1 заняло 60 тактов, что на 13% быстрее неоптимизированной программы.

Рассмотрев трассу работы оптимизированной программы, можно обнаружить, что планирование на выполнение новых команд происходило только 13 тактов (8, 12, 13, 17, 22, 27, 32, 37, 42, 47, 52, 56, 57).

Заключение

В результате проделанной работы было получено представление о принципах работы и особенностях суперскалярных конвеерных микропроцессоров.

В ходе выполнения практикума приобретён опыт работы с конвеером команд, потактовой отладкой программ, а также моделирования работы исследуемого микропроцессора в процессе выполнения программы и наблюдения внутренних сигналов.