

# 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)

ФАКУЛЬТЕ′	Г «Информатика и системы управления»
КАФЕДРА	«Программное обеспечение ЭВМ и информационные технологии»

#### ОТЧЕТ

по лабораторной работе № 2 по курсу «Архитектура ЭВМ»

на тему: «Изучение принципов работы микропроцессорного ядра RISC-V»

Студент	ИУ7-53Б		Д. В. Недолужко
	(Группа)	(Подпись, дата)	(И. О. Фамилия)
Преподава	атель		А. Ю. Попов
•		(Подпись, дата)	(И. О. Фамилия)

# 1 Теоретический раздел

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.

## 2 Практический раздел

#### 2.1 Задание №1

#### 2.1.1 Исходный код

Листинг 2.1 – Исходный код программы

```
.section .text
        .globl _start;
        len = 8 # array length
        enroll = 2 # processed items count for 1 iteration
        elem_sz = 4 \# item size
_start:
        addi x20, x0, len/enroll
        la x1, _x
lp:
        lw x2, 0(x1)
        add x31, x31, x2
        lw x3, 4(x1)
        add x31, x31, x3 #!
        addi x1, x1, elem_sz*enroll
        addi x20, x20, -1
        bne x20, x0, lp
        addi x31, x31, 1
1p2: j 1p2
        .section .data
        .4byte 0x1
_x:
        .4byte 0x2
        .4byte 0x3
        .4byte 0x4
        .4byte 0x5
        .4byte 0x6
        .4byte 0x7
        .4byte 0x8
```

# 2.1.2 Псевдокод

# Листинг 2.2 – Псевдокод

```
#define LEN 8
#define ENROLL 2
void
_start()
  int sum = 0;
  int i = LEN / ENROLL;
  int arr[] = { 1, 2, 3, 4, 5, 6, 7, 8 };
  do {
   int a = arr[0];
   sum += a;
   int b = arr[1];
   sum += b;
   arr += ENROLL;
   --i;
  } while (i != 0);
  sum += 1;
1p2:
 goto 1p2;
```

# 2.1.3 Дизассемблированный листинг

Листинг 2.3 – Дизассемблированный листинг

VINCIAM 2.0	изассемолированный листи.	111		
Disassembly of	section .text:			
80000000 <_sta	art>:			
80000000:	00400a13	addi	x20,x0,4	
80000004:	00000097	auipc	x1,0x0	
80000008:	02c08093	addi	x1,x1,44	#
80000030 <_	X >			
8000000c <lp>:</lp>				
8000000c:	0000a103	lw	x2,0(x1)	
80000010:	002f8fb3	add	x31,x31,	x2
80000014:	0040a183	lw	x3,4(x1)	
80000018:	003f8fb3	add	x31,x31,	x3
8000001c:	00808093	addi	x1,x1,8	
80000020:	fffa0a13	addi	x20,x20,	-1
80000024:	fe0a14e3	bne	x20,x0,8	000000c <
1p>				
80000028:	001f8f93	addi	x31,x31,	1
8000002c <1p2>	·:			
8000002c:	0000006f	jal	x0,80000	02c <1p2>
Disassembly of	section .data:			
80000030 <_x>:				
80000030:	0001	c.addi	x0,0	
80000032:	0000	unimp		
80000034:	0002	0 x 2		
80000036:	0000	unimp		
80000038:	0000003	1b	x0,0(x0)	# 0 <
enroll-0x2>				
8000003c:	0004	c.addi4	spn	x9,x2,0
8000003e:	0000	unimp		
80000040:	0005	c.addi	x0,1	
80000042:	0000	unimp		
80000044:	0006	0 x 6		
80000046:	0000	unimp		
80000048:	0000007	0x7		
8000004c:	0008	c.addi4	spn	x10,x2,0
			<b>-</b>	

#### 2.2 Задание №2

Получить снимок экрана, содержащий временную диаграмму выполнения стадий выборки и диспетчеризации команды с указанным адресом. 80000014, 2-я.

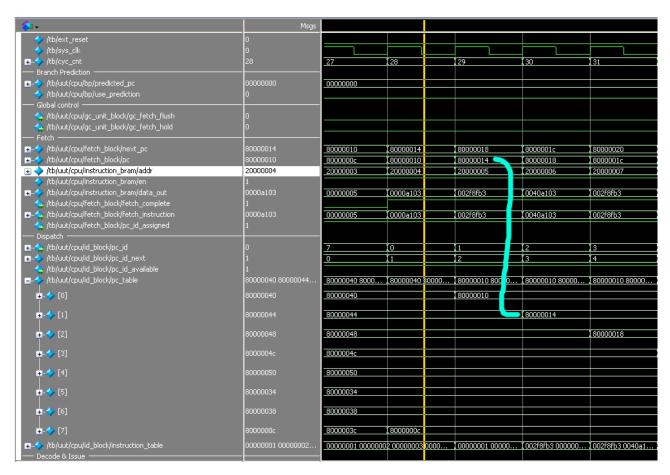


Рисунок 2.1 – Стадии выборки и диспечеризации команды по 80000014 адреса на 20й итерации

- $\bullet$  Такт 29 происходит выборка команды с адресом 80000014
- Такт 30 команда попадает на вход блока управления метаинформацией; записывается в  $pc\_table$  с  $pc\_id=1$ .

## 2.3 Задание №3

Получить снимок экрана, содержащий временную диаграмму выполнения стадии декодирования и планирования на выполнение команды с указанным адресом. 8000020, 2-я.

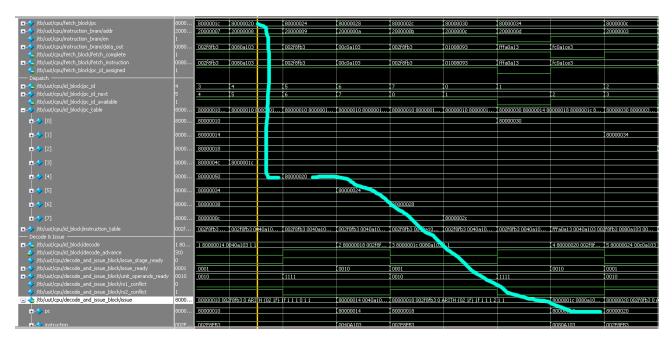


Рисунок 2.2 – Стадии декодирования и планирования на выполнение команды 80000020 на 20й итерации

- Такт 32 выборка команды с адресом 80000020
- Такт 33 диспечеризация. Команда попадает на вход блока управления метаинформацией; записывается в  $pc\_table$  с  $pc\_id=4$ .
- Такт 34-38 ожидание
- Такт 39 планирование на выполнение

Затем 2 такта пребывает в конфликте  $rs2\_conflict$ .

## 2.4 Задание №4

Получить снимок экрана, содержащий временную диаграмму выполнения стадии выполнения команды с указанным адресом. 8000000c, 2-я. (lw x2,0(x1))

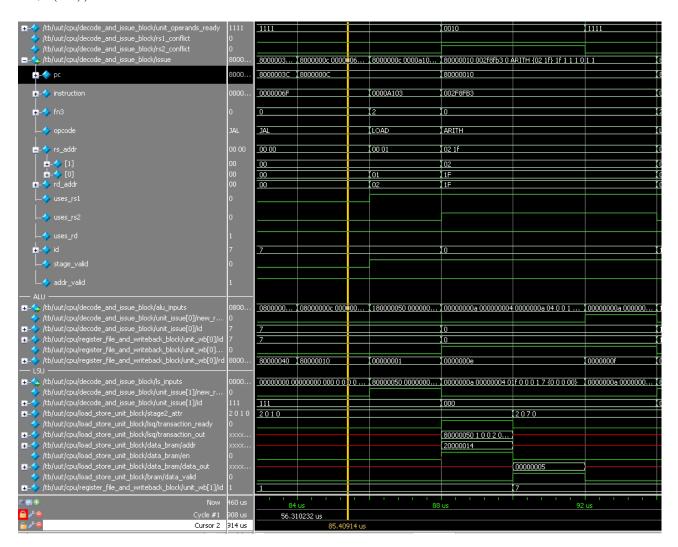


Рисунок 2.3 — Снимок экрана, содержащий временную диаграмму стадии выполнения команды по адресу 800000с на второй итерации. (lw x2,0(x1))

#### 2.5 Задание №5

Адрес команды: 80000018

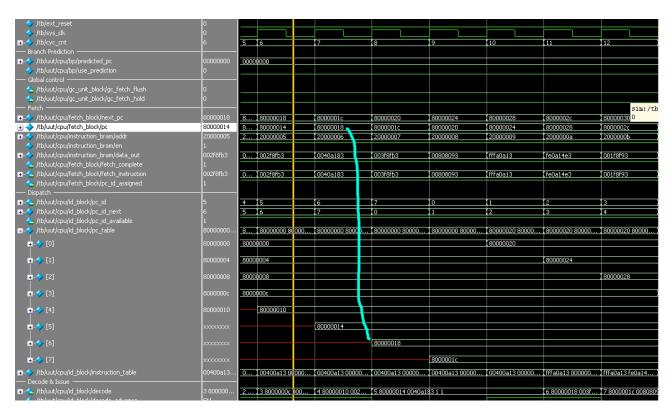


Рисунок 2.4 – Выборка, диспечеризация

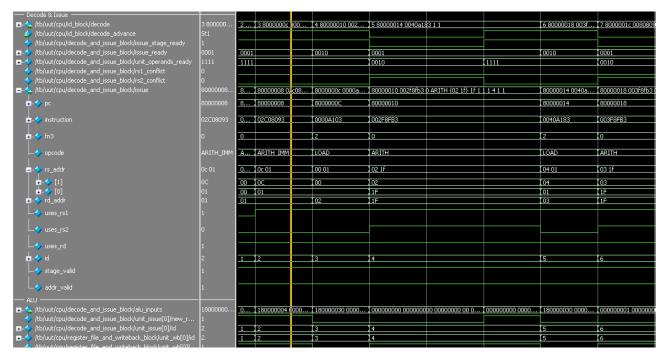


Рисунок 2.5 – Планирование, выполнение

#### 2.5.1 Оптимизация

Можно оптимизировать этот участок:

```
lw x2, 0(x1)
add x31, x31, x2
lw x3, 4(x1)
add x31, x31, x3
```

Каждая следующая команда add зависит от предыдущей команды загрузки, из-за чего они простаивают ожидая разрешение конфликта. Это можно оптимизировать путем совместного запроса данных:

```
lw x2, 0(x1)
lw x3, 4(x1)
add x31, x31, x2
add x31, x31, x3
```

Адрес	Код команды	Команда	id 1	Homep Takra    2   3   4   5   6   7   8   9   1011   1213   1415   1617   1819   2021   22   2425   2627   2829   303   323   333   435   363   738   39   404   424   4445   4647   48149   505   525   535   555   575   59   606   162   63   646     2   3   4   5   6   7   8   9   1011   1213   1415   1617   1819   2021   22   2425   2627   2829   303   323   333   435   363   738   39   404   424   4445   4647   48149   505   525   535   555   575   59   606   162   63   646     2   3   4   5   6   7   8   9   1011   1213   1415   1617   1819   2021   22   2324   2526   27829   303   323   333   435   363   738   39   404   4245   4445   4647   48149   505   525   535
80000000< start>	00400a13	addi x20,x0,4		ID D AL
	00000097	auipc x1,0x0	1	F ID D AL
		addi x1,x1,44#80000030 <x></x>	2	FID D AL
3000000c <loop></loop>	0000a103	w x2,0(x1)	3	F ID D M1M2M3
	002f8fb3	add x31.x31.x2	4	F ID D C C AL
80000014	0040a183	w x3,4(x1)	5	F ID W W D M1M2M3
80000018	003f8fb3	add x31,x31,x3	6	FIDWWWWDAL
8000001c	00808093	addix1,x1,8	7	FIDWWWWDAL
80000020	fffa0a13	addix20,x20,-1	0	F ID W W W W D AL
80000024	fc0a14c3	bne x20,x0,8000000c <loop></loop>	1	F ID W W W D B
80000028	001f8f93	addi x31,x31,1	2	F ID W W W W D X
8000002c <loop2></loop2>	0000006f	ial x0,8000002c <loop2></loop2>	3	FIDWWWWDX
80000030	00000001	<invalid operation=""></invalid>	4	F ID W W W X
80000034	00000002	<invalid operation=""></invalid>	5	F ID W X
80000038	00000003	<invalid operation=""></invalid>	6	F ID X
8000003c	00000004	<invalid operation=""></invalid>	7	F X
	00000005	<invalid operation=""></invalid>	0	FX
	0000a103	w x2,0(x1)	3	F ID D M1M2M3
	002f8fb3	add x31,x31,x2	4	FID D C C AL
80000014	0040a183	w x3,4(x1)	5	F ID W W D M1M2M3
		add x31,x31,x3	6	F ID W W W W D AL
	00808093	addix1,x1,8	7	F ID W W W W D AL
	fffa0a13	addix20,x20,-1	0	FIDWWWD AL
80000024	fe0a14e3	bne x20,x0,8000000c <loop></loop>	1	FIDWWWWD B
	0000a103	w x2.0(x1)	2	F ID W W W W D M1M2M3
		add x31,x31,x2	3	FIDWWWWDCCAL
	002f8fb3	lw x3,4(x1)	4	F ID W W W W W W D M1M2M2
		add x31.x31.x3	5	FIDWWWWWDCCAL
		addix1,x1,8	6	F ID W W W W W W D AL
80000020	fffa0a13	addix20,x20,-1	7	F ID W W W W W D AL
	fc0a14c3	bne x20,x0,8000000c <loop></loop>	0	F ID W W W W W W D B
	0000a103	w x2,0(x1)	1	F ID W W W W W D M1M2M3
			2	F ID W W W W D C C AL
	0040a183	w x3,4(x1)	3	F ID W W W W W W D M1M2M3
		add x31,x31,x3	4	F ID W W W W D C C AL
	00808093	addix1,x1,8	5	F ID W W W W W W D AL
80000020	fffa0a13	addix20,x20,-1	6	F ID W W W W W W D AL
80000024	fc0a14c3	one x20,x0,8000000c <loop></loop>	7	FIDW W W W W W D B
	0000a103	w x2,0(x1)	0	F ID W W W W W D X
	002f8fb3	add x31,x31,x2	1	F ID W W W W DX
	0040a183	lw x3,4(x1)	2	F ID W W W X
		add x31,x31,x3	3	F ID W X
		addix1,x1,8	4	F ID X
	fffa0a13	addix20,x20,-1	5	F X
80000024	fc0a14c3	bne x20,x0,8000000c <loop></loop>	6	FX
80000028	001f8f93	addi x31,x31,1	1	F ID D AL
	0000006f	ial x0,8000002c <loop2></loop2>	2	F ID D B
80000030	00000001	<invalid operation=""></invalid>	3	F ID D X
	00000002	<invalid operation=""></invalid>	4	F ID X
	00000003	<invalid operation=""></invalid>	5	SS X
	00000004	<invalid operation=""></invalid>	6	FX
	0000006f	al x0,8000002c <loop2></loop2>	4	FID D.A
			1	2 3 4 5 6 7 8 9 1011121314151617181920212223242526272829303132333435363738394041424344454647484950515253545556575859606162636466
Адрес	Код команды	Команда	ıa	Номер такта

Рисунок 2.6 – Оригинальная программа

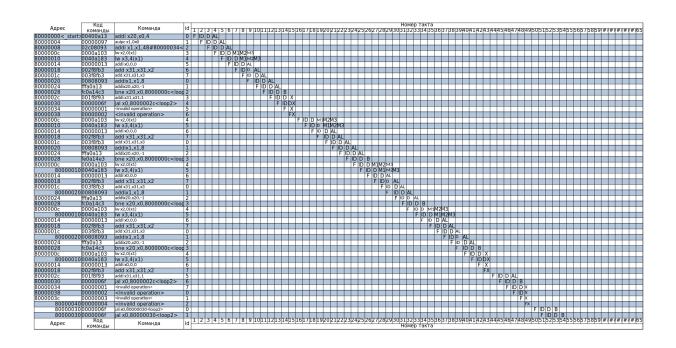


Рисунок 2.7 – Оптимизированная программа