

Преподаватель

## 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 КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

## ОТЧЕТ

	по лаооратор	нои раооте №	
Название: Из	зучение принципов раб	оты микропроцессорно	ого ядра RISC-V
Дисциплина:	Организация ЭВМ и с	истем	
Студент	ИУ6-72Б		С.В. Астахов
	(Группа)	(Подпись, дата)	(И.О. Фамилия)

(Подпись, дата)

(И.О. Фамилия)

## Вариант 1

### Введение

**Цель работы:** 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.

### Задание 1

Скомпилировать исходный код представленной программы, привести дизассемблерный листинг и псевдокод программы.

Исходный код программы:

```
.section .text
        .globl start;
        len = 8 #Размер массива
        enroll = 1 #Количество обрабатываемых элементов за одну
итерацию
     elem sz = 4 #Размер одного элемента массива
start:
        addi x20, x0, len/enroll
        la x1, x
lp:
        lw x2, \theta(x1)
        add x31, x31, x2 #!
        addi x1, x1, elem sz*enroll
        addi x20, x20, -1
        bne x20, x0, lp
        addi x31, x31, 1
lp2: j lp2
        .section .data
        .4byte 0x1
_x:
        .4byte 0x2
        .4byte 0x3
        .4byte 0x4
        .4byte 0x5
        .4byte 0x6
        .4byte 0x7
        .4byte 0x8
```

### Дизассемблерный листинг:

```
Disassembly of section .text:
```

```
80000000 < start>:
80000000:
                 00800a13
                                           addi
                                                   x20,x0,8
8000004:
                 00000097
                                                   x1,0x0
                                           auipc
8000008:
                 02408093
                                           addi
                                                   x1,x1,36 #
80000028 < x>
800000c <lp>:
                 0000a103
                                           lw
                                                   x2,0(x1)
8000000c:
                                                   x31,x31,x2
80000010:
                 002f8fb3
                                           add
80000014:
                 00408093
                                           addi
                                                   x1, x1, 4
                 fffa0a13
                                                   x20,x20,-1
80000018:
                                           addi
                 fe0a18e3
                                                   x20,x0,8000000c
8000001c:
                                           bne
<1p>
80000020:
                 001f8f93
                                           addi
                                                   x31,x31,1
80000024 <lp2>:
80000024:
                 0000006f
                                           jal
                                                   x0,80000024
<1p2>
Disassembly of section .data:
80000028 < x>:
                 0001
                                                   x0,0
80000028:
                                           c.addi
8000002a:
                 0000
                                           unimp
8000002c:
                 0002
                                           0x2
8000002e:
                 0000
                                           unimp
80000030:
                 0000003
                                           lb
                                                   x0.0(x0) # 0
<enroll-0x1>
80000034:
                 0004
                                           c.addi4spn
                                                            x9,x2,0
80000036:
                 0000
                                           unimp
80000038:
                 0005
                                           c.addi
                                                   x0,1
                 0000
                                           unimp
8000003a:
                 0006
                                           0x6
8000003c:
                 0000
                                           unimp
8000003e:
                 00000007
80000040:
                                           0×7
80000044:
                 0008
                                           c.addi4spn
                                                            x10,x2,0
```

## Псевдокод программы:

```
#define len 8
#define enroll 1
#define elem_sz 4
int _x[]={1,2,3,4,5,6,7,8};
void _start() {
   int x20 = len/enroll;
   int *x1 = _x;

   do {
     int x2 = x1[0];
```

```
x31 += x2;
x1 += enroll;
x20--;
} while(x20 != 0);
x31++;
while(1){}
```

Примечание: в результате работы данного кода регистр x31 будет содержать число 37~(0x25).

## Задание 2

Получить снимок экрана, содержащий временную диаграмму выполнения стадий выборки и диспетчеризации команды с указанным адресом (0x800000с, 1-я итерация).

Требуемый снимок экрана представлен на рисунке 1.

			1 7	
k/next_pc	80000018	8000000c	80000010	80000014
k/pc	80000014	80000034	8000000c	80000010
_bram/addr	20000005	2000000d	20000003	20000004
_bram/en	1			
_bram/data_out	002f8fb3	00000003	00000004	0000a103
k/fetch_complete	1			
k/fetch_instruction	002f8fb3	00000003	00000004	0000a103
k/pc_id_assigned	1			
 c_id	2	_		
c_id c_id_next	4	5	1 2	2
c_id_available	l'i	<u> </u>		
c_table	80000020 8000000c 8000	80000020 80.	[ 80000020 80000024 800	000028   80000020 80000000
	THE RESIDENCE			000000000000000000000000000000000000000
	80000020	80000020		
	8000000c	80000024		800000c
	10.000 (0.			
	80000010	80000028		
	8000002c	8000002c		
	80000020	8000002C		
	80000030	80000030		
	10000000			
	80000034	80000014	80000034	
	80000018	80000018		
		00000010		
	8000001c	8000001c		
struction_table	001f8f93 0000a103 00000	001f8f93 00	. 001f8f93 0000006f 0000	00001 00000002 00000003 004
		33110133 0011		1991 920 900 90 90 90 90 90 90 90 90 90 90 90 9

Рисунок 1 – стадия выборки и диспетчеризации

## Задание 3

Получить снимок экрана, содержащий временную диаграмму выполнения стадии декодирования и планирования на выполнение команды с указанным адресом (0x80000018, 1-я итерация).

Требуемый снимок экрана представлен на рисунке 2.

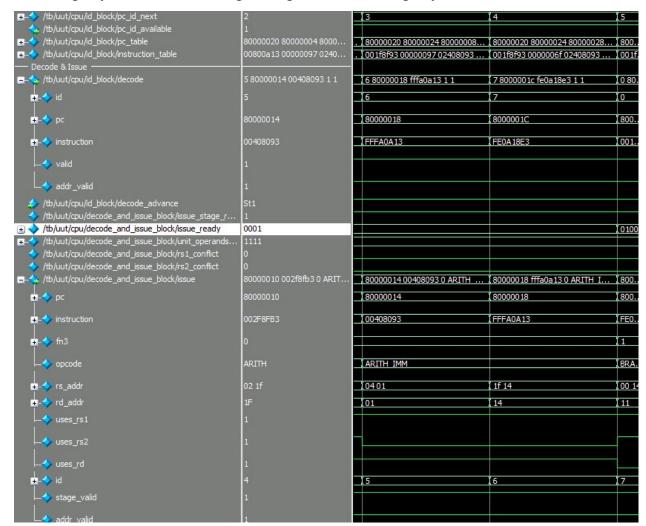


Рисунок 2 – декодирование команды

### Задание 4

Получить снимок экрана, содержащий временную диаграмму выполнения стадии выполнения команды с указанным адресом (0x8000000).

Требуемый снимок экрана представлен на рисунке 3.

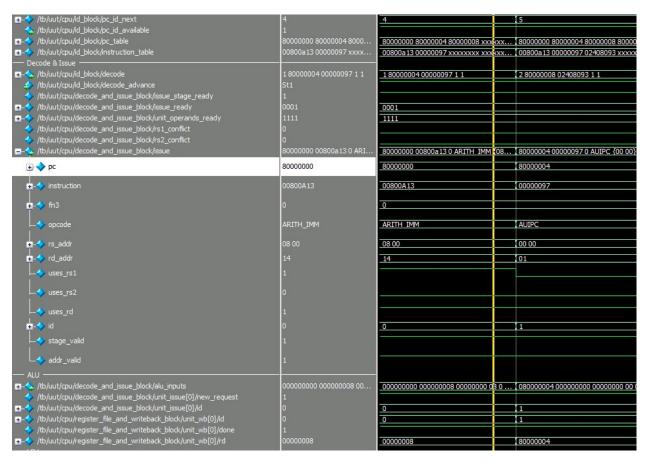


Рисунок 3 – стадия выполнения

### Задание 5.1

Получить снимок экрана, содержащий временные диаграммы сигналов, соответствующих всем стадиям выполнения команды, обозначенной в тексте программы символом #!.

В тексте программы отмечена команда add x31, x31, x2 #! По адресу 0x80000010.

Этапы ее обработки показаны на рисунках 4-6.

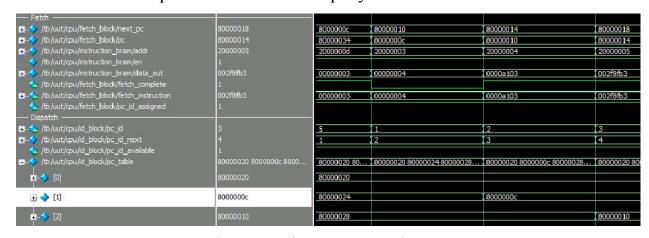


Рисунок 4 – стадия выборки

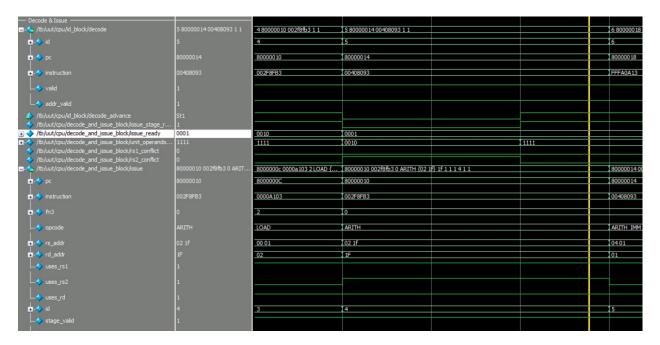


Рисунок 5 – стадия декодирования

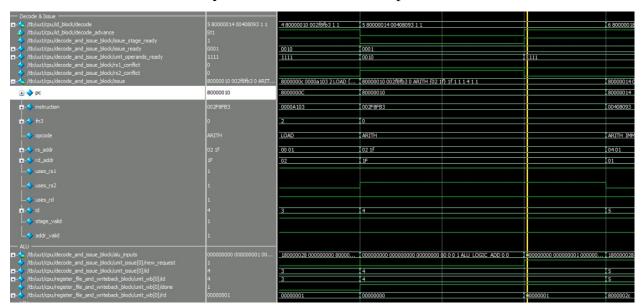


Рисунок 6 – стадия выполнения

## Задание 5.2

Анализируя диаграмму заполнить трассу выполнения программы. Сделать вывод об эффективности выполнения программы и о путях оптимизации.

Трасса программы представлена на рисунке 7.

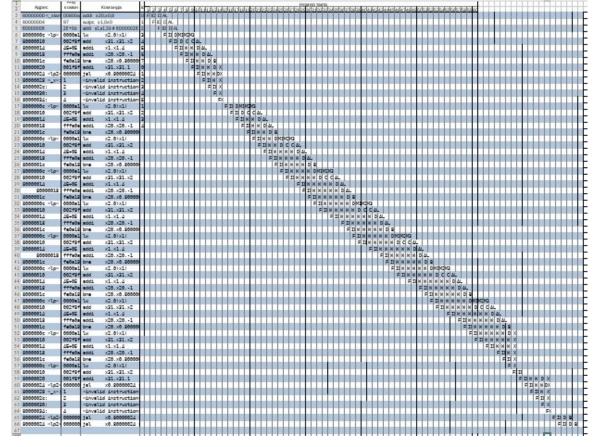


Рисунок 7 – трасса исходной программы

Программа имеет низкую эффективность, ее можно оптимизировать за счет устранения конфликтов на конвейере.

### Задание 5.3

Провести оптимизацию программы путем перестановки команд для устранения конфликтов. Перекомпилировать программу и перезапустить симуляцию. Заполнить трассу выполнения оптимизированной программы. Сравнить трассы выполнения неоптимизированной и оптимизированной версии, сделать выводы.

Будем загружать данные в несколько регистров за цикл, а затем в том же порядке складывать их, чтобы устранить конфликты из-за продолжительной работы с памятью.

Исходный код измененной программы:

```
.section .text
.globl _start;
len = 8 #Размер массива
enroll = 4 #Количество обрабатываемых элементов за одну
итерацию
```

### elem sz = 4 #Размер одного элемента массива

```
start:
        addi x20, x0, len/enroll
        la x1, _x
lp:
        lw x2, 0(x1)
          lw x3, 4(x1)
          lw x4, 8(x1)
          lw x5, 12(x1)
        add x31, x31, x2 #!
          add x31, x31, x3 #!
          add x31, x31, x4 #!
          add x31, x31, x5 #!
        addi x1, x1, elem sz*enroll
        addi x20, x20, -1
        bne x20, x0, lp
        addi x31, x31, 1
lp2: j lp2
        .section .data
        .4bvte 0x1
х:
        .4byte 0x2
        .4byte 0x3
        .4byte 0x4
        .4byte 0x5
        .4bvte 0x6
        .4byte 0x7
        .4byte 0x8
     Дизассемблерный листинг оптимизированной программы:
Disassembly of section .text:
80000000 < start>:
                 00200a13
                                          addi
                                                   x20,x0,2
80000000:
                 00000097
                                                   x1,0x0
80000004:
                                          auipc
                 03c08093
                                          addi
                                                   x1,x1,60 #
80000008:
80000040 < x>
800000c <lp>:
8000000c:
                 0000a103
                                          lw
                                                   x2,0(x1)
                                                   x3,4(x1)
80000010:
                 0040a183
                                          lw
80000014:
                 0080a203
                                          lw
                                                   x4,8(x1)
80000018:
                 00c0a283
                                                   x5,12(x1)
                                          lw
                                                   x31,x31,x2
8000001c:
                 002f8fb3
                                          add
80000020:
                 003f8fb3
                                          add
                                                   x31, x31, x3
                                                   x31,x31,x4
80000024:
                 004f8fb3
                                          add
80000028:
                 005f8fb3
                                          add
                                                   x31,x31,x5
8000002c:
                 01008093
                                          addi
                                                   x1, x1, 16
                                                   x20, x20, -1
80000030:
                 fffa0a13
                                          addi
80000034:
                 fc0a1ce3
                                                   x20,x0,8000000c
                                          bne
```

<1p>

```
80000038:
                 001f8f93
                                           addi
                                                   x31,x31,1
8000003c <1p2>:
8000003c:
                 000006f
                                           jal
                                                   x0,8000003c
<1p2>
Disassembly of section .data:
80000040 < x>:
                 0001
                                           c.addi
                                                   x0,0
80000040:
                 0000
80000042:
                                           unimp
                 0002
80000044:
                                           0 x 2
80000046:
                 0000
                                           unimp
                 0000003
80000048:
                                           lb
                                                   x0,0(x0) # 0
<elem sz-0x4>
8000004c:
                 0004
                                           c.addi4spn
                                                            x9,x2,0
8000004e:
                 0000
                                           unimp
80000050:
                 0005
                                           c.addi
                                                   x0,1
80000052:
                 0000
                                           unimp
80000054:
                 0006
                                           0x6
                 0000
80000056:
                                           unimp
80000058:
                 0000007
                                           0x7
800005c:
                 8000
                                           c.addi4spn
                                                            x10,x2,0
```

## Трасса модифицированной программы представлена на рисунке 8.

•		_	•				•		-					•										•		•						
. A	В		С	D	EF	3 H	I J	K	L M	N O	PC	R	ST	UN	V W	XY	ZA	AAB	ACA	DAE	AF A	GAH	AI A	JAK	AL A	NANA	ACAP	AÇA	RAS	ATAI	UAV	AVA
Адрес	Код команды		Команда	id	1 2	3 4	5 6	7	8 9	111	1111	T.T.	111	1	111	2 2	2   1	2 2	4 4	2 2	H	омер	Tak	a	3 3	73	3 3	3 .	4 4	4 4	7	41
80000000<_start>	00200a13	addi x2	20,x0,2	0	F ID			1111																								
80000004	97	auipc xl	1,0×0	1		D D																										
80000008	03c08093	addi xl	1,x1,60 # 80000040 <_x>	2	- 2	FID	D AL											-					-									
8000000c <lp></lp>	0000a103	lw x2	2,0(x1)	3		F	ID D	M1 P	12 M3								П														П	
80000010	0040a183	lw x3	3,4(x1)	4		1	F ID	DP	11 M2	M3																						
80000014	0080a203	lw x4	4,8(x1)	5			F	ID	D M1	M2 M3	3																					
80000018	00c0a283	lw x5	5,12(x1)	6				FI	D D	M1 M2	M3																					
0 8000001c	002f8fb3	add x3	31,×31,×2	7					FID	D AL																					П	
1 80000020	003f8fb3	add x3	31,×31,×3	0					F	ID D	AL																					
80000024	004f8fb3		31,×31,×4	1		П		П	Т	FID	DA														$\top$	П				$\top$	П	т
3 80000028	005f8fb3	add x3	31,×31,×5	2						F	ID [	AL																				
4 8000002c	1008093	addi xl	1,×1,16	3				П			FI	D D A	AL.				П									$\Box$					П	
5 80000030	fffa0a13	addi x2	20,×20,-1	4							F	ID	D AL																			
80000034	fc0alce3		20.x0.8000000c <lp></lp>	5		П						FI	D D	В										П						$\top$	П	т
80000038	001f8f93		31.×31.1	6									FID	DI	W X																	
8000003c <lp2>:</lp2>	0000006f		0,8000003c <lp2></lp2>	7				П						ID I	D X																П	т
80000040 < x>:	1		instruction>	0											D DX																	
80000042	0		instruction>	1											FX															$\top$	П	т
80000044	2		instruction>	7											FX																	
8000000c <lp></lp>	0000a103		2.0(x1)	0												FID	D M	11 M2	M3												П	
3 80000010	0040a183		3.4(x1)	1															M2 M	13												
80000014	0080a203		4,8(x1)	2															M1 M												П	т
80000018	00c0a283		5,12(x1)	3															DM		M3											
5 8000001c	002f8fb3		31,×31,×2	4															ID C												П	7
7 80000020	003f8fb3		31,x31,x3	5														m		D D	61											
80000024	004f8fb3		31,x31,x4	6										-				-		F ID											$\Box$	_
80000024	005f8fb3		31,×31,×5	7															,			) AL										
0 8000002c	1008093		1,x1,16	0	-				00				-					1000				D D									$\blacksquare$	_
1 80000030	fffa0a13		20.x201	1																		ID.										
80000030	fc0alce3		20,x0,8000000c <lp></lp>	2										-				-			-		ID									-
80000034	0040a183		3,4(x1)	3																		-	FI									
80000014	0040a183		4,8(x1)	4	100	9			85			100	1000	- 10	200		0	1000				100		X			-		9			-
80000014	00c0a283		+,0(x1) 5.12(x1)	5																				FX								
80000018 8000003c <lp2>:</lp2>	0000006f		0.8000003c <lp2></lp2>	6										- 17											E TI	D D						7
8000003C <tp2>:</tp2>	100000061		instruction>	7																						ID		v				
80000040 <_x>:	0		instruction>	0	9 2				100				1000		20			1000							-		ID D	A .				
80000042	2		instruction>																								F ID					
ACCUSED AND ADDRESS OF THE PARTY OF THE PART	2			1										175				-										X		-		7
80000048	0		instruction>	2																								FX		-		
8000004C	0000006f				50 75		0.00	10	all V	100		100	100	- 5	78 0		100			0		100	V 5		100							
2 8000003c <lp2>:</lp2>			0,8000003c <lp2></lp2>	5																					-					D B		
8000003c <lp2>:</lp2>	0000006f	jal x0	0,8000003c <lp2></lp2>	5																									F	ID D	В	
4																									-					-		_

Рисунок 8 – трасса модифицированной программы

# Вывод:

В ходе лабораторной работы произошло ознакомление с принципами функционирования, построения и особенностями архитектуры суперскалярных конвейерных микропроцессоров. Также получено представление о простейших методах оптимизации программ для таких микропроцессоров.