Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное 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»

Лабораторная работа №2
по дисциплине «Архитектура ЭВМ»

Тема Изучение принципов работы микропроцессорного ядра RISCV

Студент Байгарин Алан

Группа ИУ7-52Б

Преподаватели Попов А.Ю., Гейне М.А.

Москва, 2025

СОДЕРЖАНИЕ

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.

Персональный вариант лабораторной работы — 3.

1 Выполнение работы

1.1 Задания по исходной программе

1.1.1 Листинги исходной программы

Исходный текст программы test.s приведен в листинге 1.1

```
  .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
loop:
  lw x2, 0(x1)
  add x31, x31, x2
  lw x2, 4(x1)
  add x31, x31, x2
  lw x2, 8(x1)
  add x31, x31, x2
  lw x2, 12(x1)
  add x31, x31, x2
  addi x1, x1, elem_sz*enroll
  addi x20, x20, -1
  bne x20, x0, loop
  addi x31, x31, 1
forever: j forever

  .section .data
_x: .4byte 0x1
  .4byte 0x2
  .4byte 0x3
  .4byte 0x4
  .4byte 0x5
  .4byte 0x6
  .4byte 0x7
  .4byte 0x8
```

Листинг 1.1 — Исходный код программы text.s

Листинг программы test, полученный через makefile, приведен в листинге 1.2

```
SYMBOL TABLE:
80000000 l    d  .text  00000000 .text
80000040 l    d  .data  00000000 .data
00000000 l    df *ABS*  00000000 test.o
00000008 l       *ABS*  00000000 len
00000004 l       *ABS*  00000000 enroll
00000004 l       *ABS*  00000000 elem_sz
80000040 l       .data  00000000 _x
8000000c l       .text  00000000 loop
8000003c l       .text  00000000 forever
80000000 g       .text  00000000 _start
80000060 g       .data  00000000 _end



Дизассемблирование раздела .text:

80000000 <_start>:
80000000: 00200a13            addi  x20,x0,2
80000004: 00000097            auipc x1,0x0
80000008: 03c08093            addi  x1,x1,60 # 80000040 <_x>

8000000c <loop>:
8000000c: 0000a103            lw  x2,0(x1)
80000010: 002f8fb3            add x31,x31,x2
80000014: 0040a103            lw  x2,4(x1)
80000018: 002f8fb3            add x31,x31,x2
8000001c: 0080a103            lw  x2,8(x1)
80000020: 002f8fb3            add x31,x31,x2
80000024: 00c0a103            lw  x2,12(x1)
80000028: 002f8fb3            add x31,x31,x2
8000002c: 01008093            addi  x1,x1,16
80000030: fffa0a13            addi  x20,x20,-1
80000034: fc0a1ce3            bne x20,x0,8000000c <loop>
80000038: 001f8f93            addi  x31,x31,1

8000003c <forever>:
8000003c: 0000006f            jal x0,8000003c <forever>

Дизассемблирование раздела .data:
```

```
80000040 <_x>:
80000040: 0001                    .insn 2, 0x0001
80000042: 0000                    .insn 2, 0x
80000044: 0002                    .insn 2, 0x0002
80000046: 0000                    .insn 2, 0x
80000048: 00000003              lb  x0,0(x0) # 0 <elem_sz-0x4>
8000004c: 0004                    .insn 2, 0x0004
8000004e: 0000                    .insn 2, 0x
80000050: 0005                    .insn 2, 0x0005
80000052: 0000                    .insn 2, 0x
80000054: 0006                    .insn 2, 0x0006
80000056: 0000                    .insn 2, 0x
80000058: 00000007              .insn 4, 0x0007
8000005c: 0008                    .insn 2, 0x0008
```

Листинг 1.2 — Листинг программы test полученный через makefile

16-ричный набор команд программы test приведен в листинге 1.3

```
00200a13
00000097
03c08093
0000a103
002f8fb3
0040a103
002f8fb3
0080a103
002f8fb3
00c0a103
002f8fb3
01008093
fffa0a13
fc0a1ce3
001f8f93
0000006f
00000001
00000002
00000003
00000004
00000005
00000006
00000007
```

```
00000008
```

Листинг 1.3 — Команды программы test в 16-ричном формате

Псевдокод поясняющий работу программы test приведен в листинге 1.4

```
int len = 8;                    // размер массива
int enroll = 4;                 // обрабатываемых элементов за итерацию
int arr[len] = {1,2,3,4,5,6,7,8};
int sum = 0;

for (int i = 0; i < len; i += enroll) {
    sum += arr[i];
    sum += arr[i + 1];
    sum += arr[i + 2];
    sum += arr[i + 3];
}

sum += 1;                       // добавление 1 к итоговой сумме

while (1);                      // бесконечный цикл
```

Листинг 1.4 — Псевдокод к программе test

1.1.2 Скриншоты к исходной программе

Стадии выборки и диспетчеризации (fetch&dispatch, или F+ID) 1-й итерации команды 80000014 (согласно варианту) приведена на рисунке 1.1

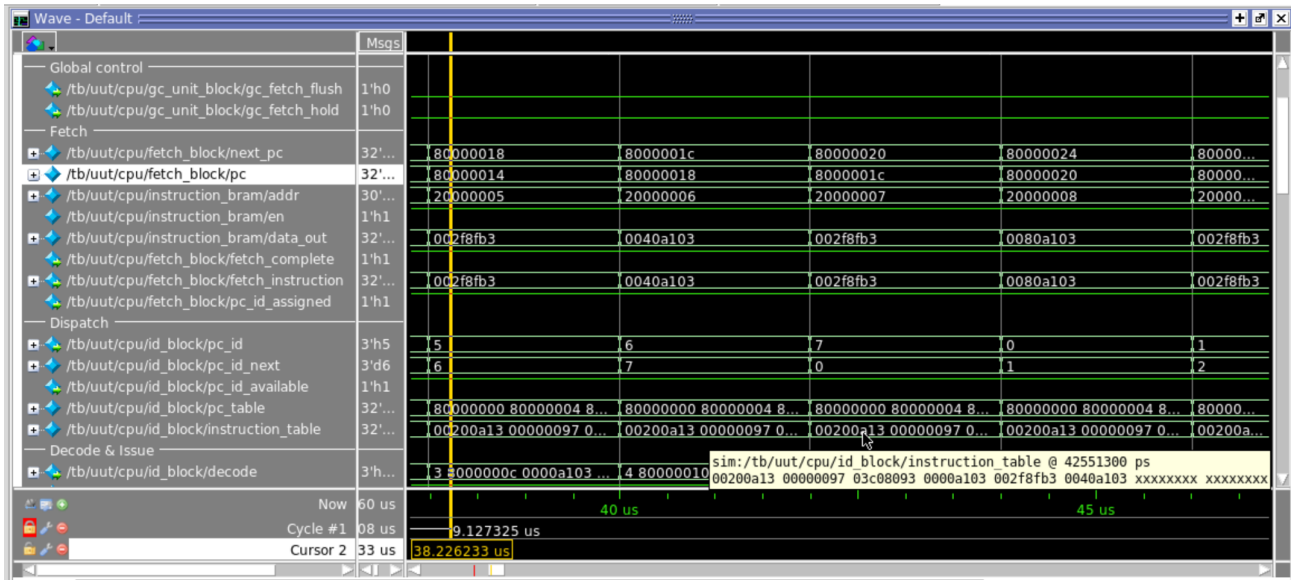


Рисунок 1.1 — Стадии выборки и диспетчеризации 1-й итерации команды 80000014

Стадия декодирования и планирования (decode_and_issue, или D) 1-й итерации команды 80000020 (согласно варианту) приведена на рисунке 1.2

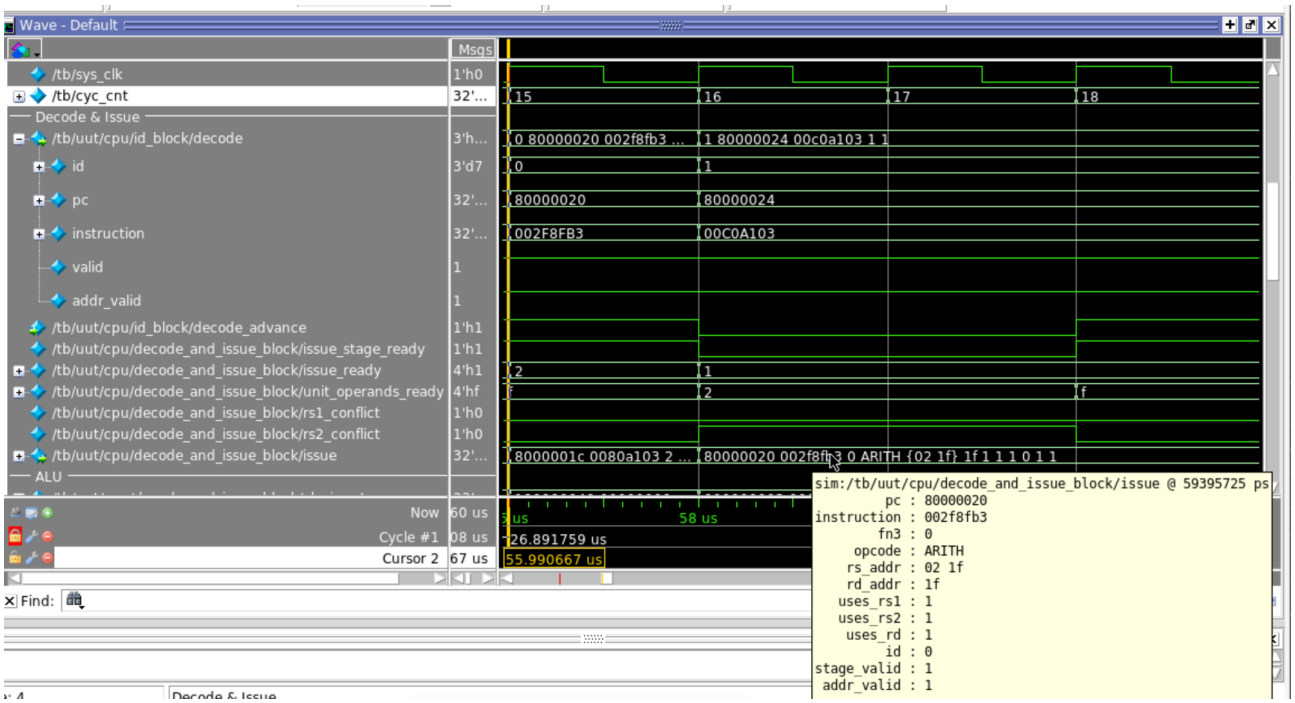


Рисунок 1.2 — Стадия декодирования и планирования 1-й итерации команды 80000020

Стадия выполнения (ALU) команды 80000008 (согласно варианту) приведена на рисунке 1.3

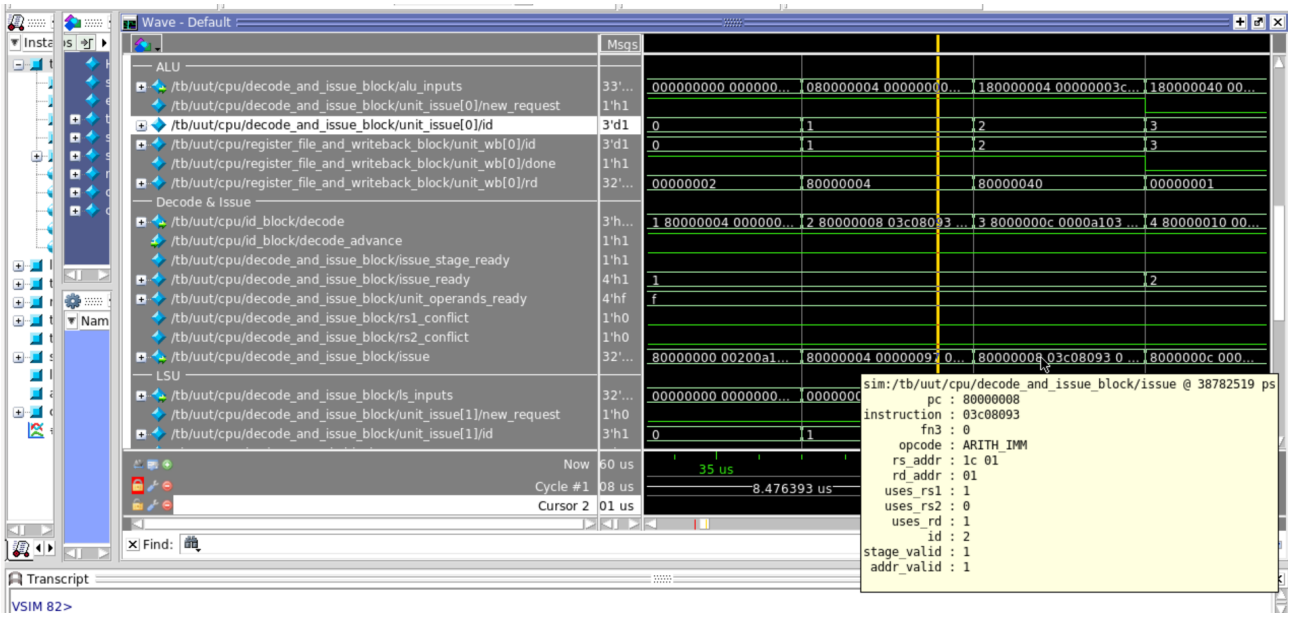


Рисунок 1.3 — Стадия выполнения команды 80000008

1.2 Задания по программе 3-о варианта

1.2.1 Листинг программы

Исходный текст программы 3-о варианта приведен в листинге 1.5

```
        .section .text
        .globl _start;
        len = 8 #Размер массива
        enroll = 1 #Количество обрабатываемых элементов за одну итераци
           ю
        elem_sz = 4 #Размер одного элемента массива

_start:
        la x1, _x
        addi x20, x1, elem_sz*(len-1) #Адрес последнего элемента
        add x31, x0, x0
lp:
        lw x2, 0(x1)
        add x31, x31, x2 #!
        addi x1, x1, elem_sz*enroll
        bne x1, x20, lp
        addi x31, x31, 1
lp2:
        j lp2

        .section .data
_x:     .4byte 0x1
        .4byte 0x2
        .4byte 0x3
        .4byte 0x4
        .4byte 0x5
        .4byte 0x6
        .4byte 0x7
        .4byte 0x8
```

Листинг 1.5 — Исходный код программы по варианту

Листинг программы 3-о варианта, полученный через makefile, приведен в листинге 1.6

```
SYMBOL TABLE:
80000000 l    d  .text  00000000 .text
80000028 l    d  .data  00000000 .data
00000000 l    df *ABS*  00000000 var3.o
```

```
00000008 l       *ABS*  00000000 len
00000001 l       *ABS*  00000000 enroll
00000004 l       *ABS*  00000000 elem_sz
80000028 l       .data  00000000 _x
80000010 l       .text  00000000 lp
80000024 l       .text  00000000 lp2
80000000 g       .text  00000000 _start
80000048 g       .data  00000000 _end



Дизассемблирование раздела .text:

80000000 <_start>:
80000000: 00000097            auipc x1,0x0
80000004: 02808093            addi  x1,x1,40 # 80000028 <_x>
80000008: 01c08a13            addi  x20,x1,28
8000000c: 00000fb3            add x31,x0,x0

80000010 <lp>:
80000010: 0000a103            lw  x2,0(x1)
80000014: 002f8fb3            add x31,x31,x2
80000018: 00408093            addi  x1,x1,4
8000001c: ff409ae3            bne x1,x20,80000010 <lp>
80000020: 001f8f93            addi  x31,x31,1

80000024 <lp2>:
80000024: 0000006f            jal x0,80000024 <lp2>

Дизассемблирование раздела .data:

80000028 <_x>:
80000028: 0001                  .insn 2, 0x0001
8000002a: 0000                  .insn 2, 0x
8000002c: 0002                  .insn 2, 0x0002
8000002e: 0000                  .insn 2, 0x
80000030: 00000003            lb  x0,0(x0) # 0 <enroll-0x1>
80000034: 0004                  .insn 2, 0x0004
80000036: 0000                  .insn 2, 0x
80000038: 0005                  .insn 2, 0x0005
8000003a: 0000                  .insn 2, 0x
```

```
8000003c: 0006                    .insn 2, 0x0006
8000003e: 0000                    .insn 2, 0x
80000040: 00000007              .insn 4, 0x0007
80000044: 0008                    .insn 2, 0x0008
  ...
```

Листинг 1.6 — Листинг программы по варианту полученный через makefile

16-ричный набор команд программы 3-о варианта приведен в листинге 1.7

```
00000097
02808093
01c08a13
00000fb3
0000a103
002f8fb3
00408093
ff409ae3
001f8f93
0000006f
00000001
00000002
00000003
00000004
00000005
00000006
00000007
00000008
```

Листинг 1.7 — Команды программы по варианту в 16-ричном формате

Псевдокод поясняющий работу программы 3-о варианта приведен в листинге 1.8

```
int len = 8;                      // размер массива
int enroll = 1;                   // количество элементов за итерацию
int elem_sz = 4;                  // размер одного элемента (байт)
int arr[len] = {1,2,3,4,5,6,7,8};
int sum = 0;

int* ptr = &arr[0];
int* end_ptr = &arr[len - 1];

while (ptr != end_ptr)            // пока указатель не равен указателю на
    последний
{
```

```
    sum += *ptr;                    // прибавить к сумме значение по адресу
        указателя
    ptr += enroll;
}
                                    // последний не добавится, ибо так уста
                                       новлены указатели
sum += 1;                           // добавляем 1 к сумме

while (1);                          // вечный цикл
```

Листинг 1.8 — Псевдокод к программе по варианту

1.2.2 Стадии выполнения выделенной команды

В исходном файле выделена команда *add x31, x31, x2*, имеющая адрес 80000014

Стадии выборки, диспетчеризации (fetch&dispatch, или F+ID) команды по варианту приведена на рисунке 1.4

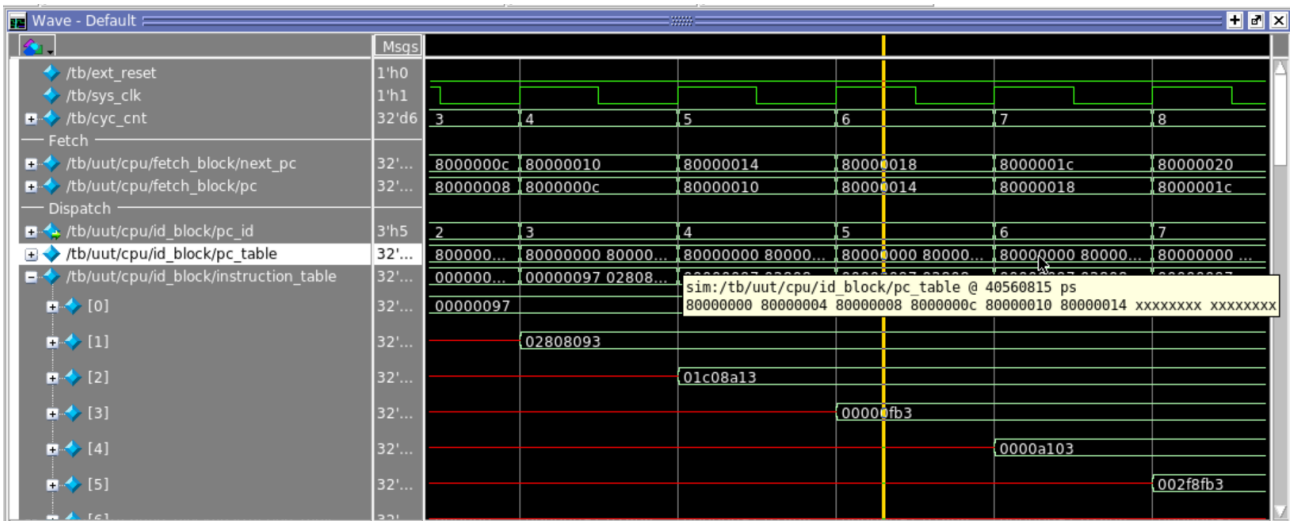


Рисунок 1.4 — Стадии выборки и диспетчеризации команды 80000014

Стадии декодирования и планирования, выполнения (decode_and_issue&ALU, или D+ALU) команды по варианту приведена на рисунке 1.5

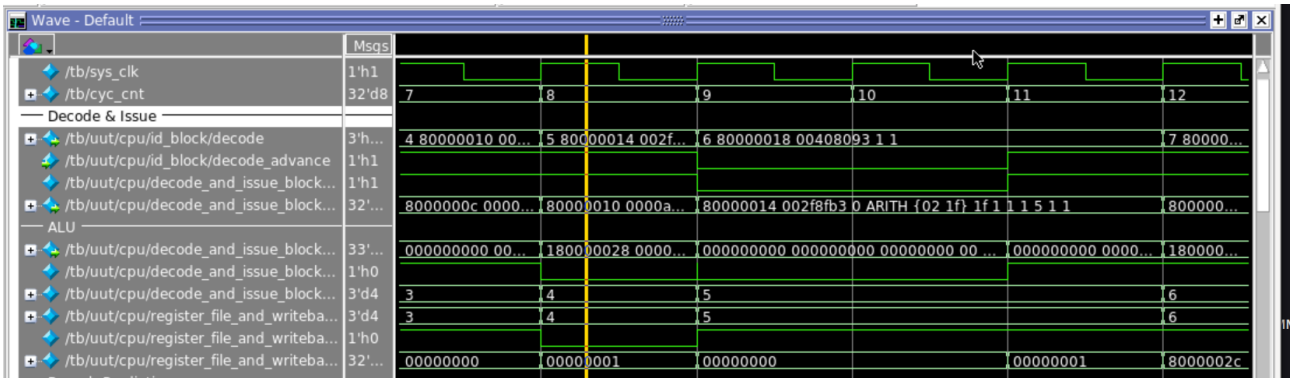


Рисунок 1.5 — Стадия декодирования и планирования команды 80000014

1.2.3 Трасса выполнения программы

Трасса выполнения программы по варианту приведена на рисунке 1.6

Адрес	Код	id	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
80000000	00000097	0	F	ID	D	AL																																																														
80000004	02808093	1		F	ID	D	AL																																																													
80000008	01c08a13	2			F	ID	D	AL																																																												
8000000c	00000fb3	3				F	ID	D	AL																																																											
80000010	0000a103	4					F	ID	D	M1	M2	M3																																																						
80000014	002f8fb3	5						F	ID	D	C	C	AL																																																					
80000018	00408093	6							F	ID	W	W	D	AL																																																				
8000001c	ff409ae3	7								F	ID	W	W	D	B																																																			
80000020	001f8f93	0									F	ID	W	W	D	X																																																		
80000024	0000006f	1										F	ID	W	W	DX																																																		
80000028	00000001	2											F	ID	W	X																																																		
8000002c	00000002	3												F	ID	X																																																		
80000030	00000003	4													F	X																																																		
80000034	unknown	5														FX																																																		
80000010	0000a103	1															F	ID	D	M1	M2	M3																																												
80000014	002f8fb3	2																F	ID	D	C	C	AL																																											
80000018	00408093	3																	F	ID	W	W	D	AL																																										
8000001c	ff409ae3	4																		F	ID	W	W	D	B																																									
80000010	0000a103	5																			F	ID	W	W	D	M1	M2	M3																																						
80000014	002f8fb3	6																				F	ID	W	W	D	C	C	AL																																					
80000018	00408093	7																					F	ID	W	W	W	W	D	AL																																				
8000001c	ff409ae3	0																						F	ID	W	W	W	W	D	B																																			
80000010	0000a103	1																							F	ID	W	W	W	W	D	M1	M2	M3																																
80000014	002f8fb3	2																								F	ID	W	W	W	W	D	C	C	AL																															
80000018	00408093	3																									F	ID	W	W	W	W	W	W	D	AL																														
8000001c	ff409ae3	4																										F	ID	W	W	W	W	W	W	D	B																													
80000010	0000a103	5																												F	ID	W	W	W	W	W	D	M1	M2	M3																										
80000014	002f8fb3	6																													F	ID	W	W	W	W	W	D	C	C	AL																									
80000018	00408093	7																														F	ID	W	W	W	W	W	W	W	D	AL																								
8000001c	ff409ae3	0																																F	ID	W	W	W	W	W	W	D	B																							
80000010	0000a103	1																																		F	ID	W	W	W	W	W	D	M1	M2	M3																				
80000014	002f8fb3	2																																			F	ID	W	W	W	W	W	D	C	C	AL																			
80000018	00408093	3																																				F	ID	W	W	W	W	W	W	W	D	AL																		
8000001c	ff409ae3	4																																						F	ID	W	W	W	W	W	W	D	B																	
80000010	0000a103	5																																								F	ID	W	W	W	W	W	D	M1	M2	M3														
80000014	002f8fb3	6																																									F	ID	W	W	W	W	W	D	C	C	AL													
80000018	00408093	7																																										F	ID	W	W	W	W	W	W	W	D	AL												
8000001c	ff409ae3	0																																												F	ID	W	W	W	W	W	W	D	B											
80000010	0000a103	1																																														F	ID	W	W	W	W	W	D	X										
80000014	002f8fb3	2																																															F	ID	W	W	W	W	W	DX										
80000018	00408093	3																																																F	ID	W	W	W	W	X										
8000001c	ff409ae3	4																																																		F	ID	W	W	X										
80000010	0000a103	5																																																				F	ID	X										
80000014	002f8fb3	6																																																					F	X										
80000018	unknown	7																																																						FX										
80000020	001f8f93	2																																																							F	ID	D	AL						
80000024	0000006f	3																																																								F	ID	D	AL					
80000028	00000001	4																																																									F	ID	D	X				
8000002c	00000002	5																																																										F	ID	DX				
80000030	00000003	6																																																											F	X				
80000034	unknown	7																																																												FX				
80000024	0000006f	5																																																													F	ID	D	AL	
80000024	0000006f	6																																																														F	ID	D	AL

Рисунок 1.6 — Трасса выполнения программы по варианту

1.2.4 Анализ программы

В трассе выполнения 1.6 видно, что арифметическая команда 80000014, зависящая от результата работы команды чтения из памяти 80000010, ожидает 2 такта из-за конфликта по регистру (C), в то время как идущая следом за ней 80000018 выполняет арифметическую операцию с другой переменной. Следовательно, порядок выполнения команд 80000014 и 80000018 можно поменять для уменьшения задержки выполнения первой на 1 такт. Для устранения оставшегося такта ожидания имеет смысл добавить один такт простоя командой NOP между командами 80000010 и 80000014, чтобы полностью исключить возникновение задержек.

1.3 Оптимизация программы по варианту

1.3.1 Листинг программы

Исходный текст оптимизированной программы 3-о варианта приведен в листинге 1.9

```
        .section .text
        .globl _start;
        len = 8 #Размер массива
        enroll = 1 #Количество обрабатываемых элементов за одну итераци
           ю
        elem_sz = 4 #Размер одного элемента массива

_start:
        la x1, _x
        add x31, x0, x0
        addi x20, x1, elem_sz*(len-1) #Адрес последнего элемента
lp:
        lw x2, 0(x1)
        addi x1, x1, elem_sz*enroll     # swap_1
        nop                           # доп.такт задержки
        add x31, x31, x2 #!             # swap_1
        bne x1, x20, lp
        addi x31, x31, 1
lp2:
        j lp2

        .section .data
_x:     .4byte 0x1
        .4byte 0x2
        .4byte 0x3
        .4byte 0x4
        .4byte 0x5
        .4byte 0x6
        .4byte 0x7
        .4byte 0x8
```

Листинг 1.9 — Исходный код оптимизированной программы

Листинг оптимизированной программы 3-о варианта, полученный через makefile, приведен в листинге 1.10

```
SYMBOL TABLE:
80000000 l    d  .text  00000000 .text
8000002c l    d  .data  00000000 .data
00000000 l    df *ABS*  00000000 newVar3.o
00000008 l       *ABS*  00000000 len
00000001 l       *ABS*  00000000 enroll
00000004 l       *ABS*  00000000 elem_sz
8000002c l       .data  00000000 _x
80000010 l       .text  00000000 lp
80000028 l       .text  00000000 lp2
80000000 g       .text  00000000 _start
8000004c g       .data  00000000 _end



Дизассемблирование раздела .text:

80000000 <_start>:
80000000: 00000097            auipc x1,0x0
80000004: 02c08093            addi  x1,x1,44 # 8000002c <_x>
80000008: 00000fb3            add x31,x0,x0
8000000c: 01c08a13            addi  x20,x1,28

80000010 <lp>:
80000010: 0000a103            lw  x2,0(x1)
80000014: 00408093            addi  x1,x1,4
80000018: 00000013            addi  x0,x0,0
8000001c: 002f8fb3            add x31,x31,x2
80000020: ff4098e3            bne x1,x20,80000010 <lp>
80000024: 001f8f93            addi  x31,x31,1

80000028 <lp2>:
80000028: 0000006f            jal x0,80000028 <lp2>

Дизассемблирование раздела .data:

8000002c <_x>:
8000002c: 0001                  .insn 2, 0x0001
8000002e: 0000                  .insn 2, 0x
```

```
80000030: 0002                    .insn 2, 0x0002
80000032: 0000                    .insn 2, 0x
80000034: 00000003              lb  x0,0(x0) # 0 <enroll-0x1>
80000038: 0004                    .insn 2, 0x0004
8000003a: 0000                    .insn 2, 0x
8000003c: 0005                    .insn 2, 0x0005
8000003e: 0000                    .insn 2, 0x
80000040: 0006                    .insn 2, 0x0006
80000042: 0000                    .insn 2, 0x
80000044: 00000007              .insn 4, 0x0007
80000048: 0008                    .insn 2, 0x0008
```

Листинг 1.10 — Листинг оптимизированной программы полученный через makefile

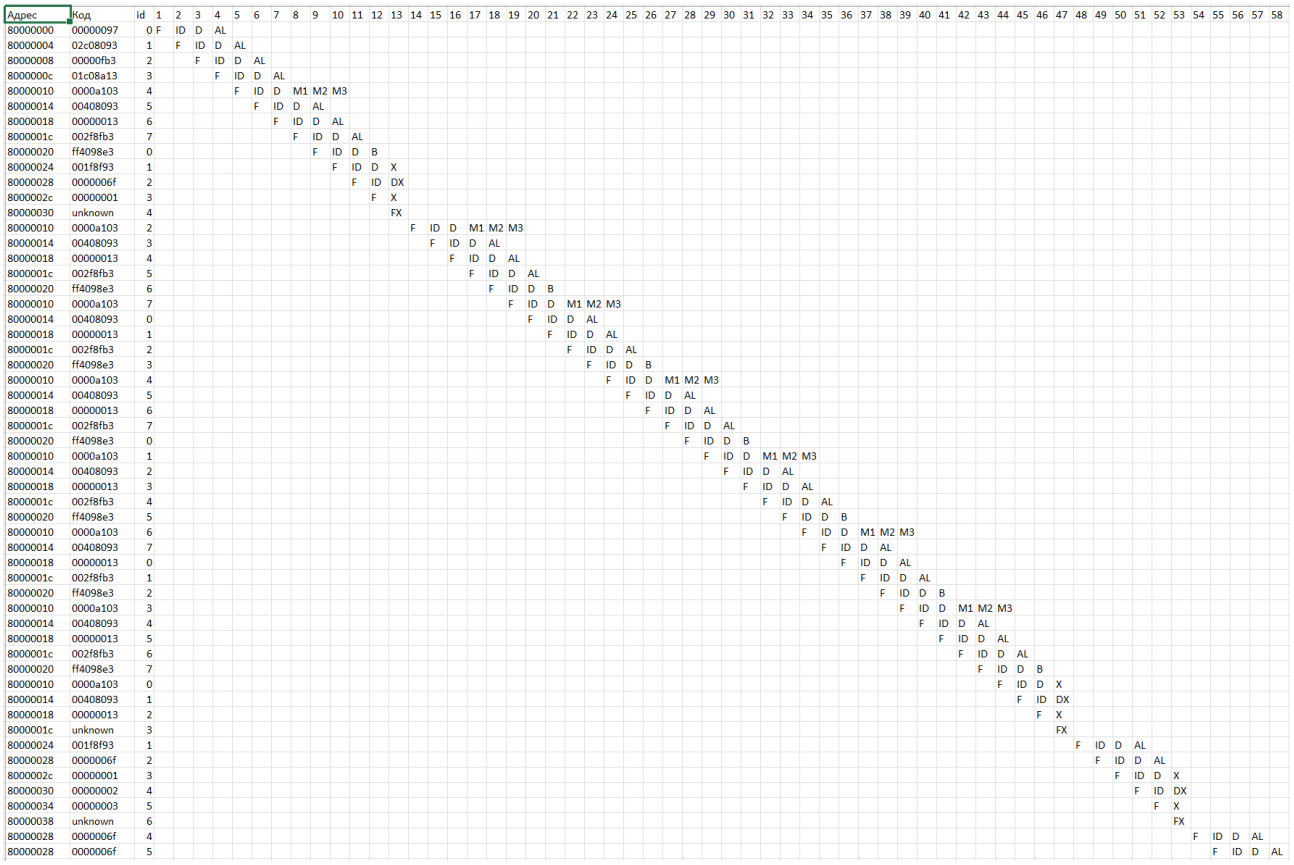
16-ричный набор команд оптимизированной программы 3-о варианта приведен в листинге 1.11

```
00000097
02c08093
00000fb3
01c08a13
0000a103
00408093
00000013
002f8fb3
ff4098e3
001f8f93
0000006f
00000001
00000002
00000003
00000004
00000005
00000006
00000007
00000008
```

Листинг 1.11 — Команды оптимизированной программы в 16-ричном формате

1.3.2 Трасса оптимизированной программы

Трасса выполнения оптимизированной программы приведена на рисунке 1.7



Адрес	Код	id	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
80000000	00000097	0	F	ID	D	AL																																																								
80000004	02c08093	1		F	ID	D	AL																																																								
80000008	00000fb3	2			F	ID	D	AL																																																						
8000000c	01c08a13	3				F	ID	D	AL																																																					
80000010	0000a103	4					F	ID	D	M1	M2	M3																																																		
80000014	00408093	5						F	ID	D	AL																																																			
80000018	00000013	6							F	ID	D	AL																																																	
8000001c	002f8fb3	7								F	ID	D	AL																																																	
80000020	ff4098e3	0									F	ID	D	B																																														
80000024	001f8f93	1										F	ID	D	X																																													
80000028	0000006f	2											F	ID	DX																																													
8000002c	00000001	3												F	X																																													
80000030	unknown	4													FX																																													
80000010	0000a103	2														F	ID	D	M1	M2	M3																																								
80000014	00408093	3															F	ID	D	AL																																									
80000018	00000013	4																F	ID	D	AL																																								
8000001c	002f8fb3	5																	F	ID	D	AL																																								
80000020	ff4098e3	6																		F	ID	D	B																																							
80000010	0000a103	7																			F	ID	D	M1	M2	M3																																				
80000014	00408093	0																				F	ID	D	AL																																					
80000018	00000013	1																					F	ID	D	AL																																				
8000001c	002f8fb3	2																						F	ID	D	AL																																			
80000020	ff4098e3	3																							F	ID	D	B																																		
80000010	0000a103	4																								F	ID	D	M1	M2	M3																																
80000014	00408093	5																									F	ID	D	AL																																
80000018	00000013	6																										F	ID	D	AL																															
8000001c	002f8fb3	7																											F	ID	D	AL																														
80000020	ff4098e3	0																												F	ID	D	B																													
80000010	0000a103	1																													F	ID	D	M1	M2	M3																									
80000014	00408093	2																														F	ID	D	AL																									
80000018	00000013	3																															F	ID	D	AL																							
8000001c	002f8fb3	4																																F	ID	D	AL																						
80000020	ff4098e3	5																																	F	ID	D	B																				
80000010	0000a103	6																																		F	ID	D	M1	M2	M3																
80000014	00408093	7																																			F	ID	D	AL																
80000018	00000013	0																																				F	ID	D	AL														
8000001c	002f8fb3	1																																					F	ID	D	AL												
80000020	ff4098e3	2																																						F	ID	D	B											
80000010	0000a103	3																																							F	ID	D	M1	M2	M3								
80000014	00408093	4																																								F	ID	D	AL										
80000018	00000013	5																																									F	ID	D	AL								
8000001c	002f8fb3	6																																										F	ID	D	AL							
80000020	ff4098e3	7																																											F	ID	D	B						
80000010	0000a103	0																																												F	ID	D	X				
80000014	00408093	1																																														F	ID	DX		
80000018	00000013	2																																															F	X		
8000001c	unknown	3																																																	FX		
80000024	001f8f93	1																																																			F	ID	D	AL	
80000028	0000006f	2																																																			F	ID	D	AL
8000002c	00000001	3																																																				F	ID	D	X
80000030	00000002	4																																																				F	ID	DX
80000034	00000003	5																																																					F	X
80000038	unknown	6																																																						FX
80000028	0000006f	4																																																						F	ID	D	AL	
80000028	0000006f	5																																																					F	ID	D	AL

Рисунок 1.7 — Трасса выполнения программы по варианту

1.4 Вывод

В результате замены порядка арифметических команд и добавления одного такта ожидания через NOP, время выполнения значимой части кода (идущей до вечного цикла) уменьшилось с 54 тактов до 47, т.е. на 13%, а также была полностью устранена нарастающая задержка.

ЗАКЛЮЧЕНИЕ

В ходе оптимизации программы по варианту, скорость выполнения значимой части кода была увеличена на 13%, а также была полностью устранена нарастающая задержка.

Цели работы, а именно основная, заключающаяся в ознакомлении с принципами функционирования, построения и особенностями архитектуры суперскалярных конвейерных микропроцессоров, а также дополнительная, заключающаяся в знакомстве с принципами проектирования и верификации сложных цифровых устройств с использованием языка описания аппаратуры SystemVerilog и ПЛИС, были достигнуты.