Министерство науки и высшего образования Российской Федерации



Федеральное государственное вюджетное 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»
НАПРАВЛЕН!	ИЕ ПОДГОТОВКИ «09.03.04 Программная инженерия»

ОТЧЕТ по лабораторной работе №2

Название:	Изучение прин	ципов работы микропроцессор	ного ядра RISC-V
Дисциплина:		Архитектура ЭВМ	
Студент	ИУ7-54Б Группа	Подпись, дата	С. Д. Параскун и. о. Фамилия
Преподаватель		—————————————————————————————————————	А. Ю. Попов И. О. Фамилия

В данной работе будет выполнен 12 вариант.

Листинг 1.1 – Код рассматриваемой программы

```
.section .text
           .globl _start;
           len = 8 #array length
           enroll = 4 #count of elems in one loop
           elem_sz = 4 #size of one element
  _start:
           la x1, _x
           addi x20, x1, elem_sz*len #address of last elem
10 lp:
           lw x2, 0(x1)
11
           lw x3, 4(x1)
12
           add x31, x31, x2 #!
13
           add x31, x31, x3
           lw x4, 8(x1)
15
           lw x5, 12(x1)
16
           add x31, x31, x4
17
           add x31, x31, x5
           addi x1, x1, elem_sz*enroll
19
           bne x1, x20, lp
20
           addi x31, x31, 1
22 lp2: j lp2
23
           .section .data
           .4byte 0x1
25
           .4byte 0x2
26
           .4byte 0x3
           .4byte 0x4
28
           .4byte 0x5
29
           .4byte 0x6
           .4byte 0x7
31
           .4byte 0x8
```

В регистре x31 в конце программы будет находится сумма элементов массива +1, т.е. значение его будет 37. За один цикл обрабатывается 4 элемента, при этом сначала считываются 2, потом прибавляются к результату, потом считываются еще 2 и прибавляются.

Листинг 1.2 – Псевдокод рассматриваемой программы

```
1 #define len 8
  #define enroll 4
3 #define elem_sz 4
4 int _x[]={1,2,3,4,5,6,7,8};
  void _start() {
      int *x1 = _x;
      int x20 = x1 + len;
      do {
9
        int x2 = x1[0];
         int x3 = x1[1];
11
        x31 += x2;
12
         x31 += x3;
13
         int x4 = x1[2];
14
        int x5 = x1[3];
15
        x31 += x4;
        x31 += x5;
17
        x1 += enroll;
18
      } while(x1 != x20);
      x31++;
20
      while (1) {}
21
22 }
```

Листинг 1.3 – Дизассемблированный листинг рассматриваемой

```
программы
 80000000 <_start>:
2 80000000:
                   00000097
                                                      x1,0x0
                                             auipc
3 80000004:
                   03 c08093
                                                      x1,x1,60 # 8000003c <_x>
                                             addi
  80000008:
                   02008a13
                                             addi
                                                      x20,x1,32
6 8000000c <lp>:
 800000c:
                   0000a103
                                             lw
                                                      x2,0(x1)
8 80000010:
                   0040a183
                                                      x3,4(x1)
                                             lw
9 80000014:
                   002f8fb3
                                             add
                                                      x31,x31,x2
10 80000018:
                   003f8fb3
                                             add
                                                      x31,x31,x3
11 8000001c:
                   0080a203
                                                      x4,8(x1)
                                             lw
12 80000020:
                                                      x5,12(x1)
                   00c0a283
                                             lw
                                                      x31,x31,x4
13 80000024:
                   004f8fb3
                                             add
14 80000028:
                                                      x31,x31,x5
                   005f8fb3
                                             add
15 8000002c:
                   01008093
                                                      x1,x1,16
                                             addi
16 80000030:
                                                      x1,x20,8000000c <lp>
                   fd409ee3
                                             bne
17 80000034:
                                                      x31,x31,1
                   001f8f93
                                             addi
19 80000038 <1p2>:
20 80000038:
                   000006f
                                             jal
                                                      x0,80000038 <1p2>
```

На рисунке ниже рассмотренна временная диаграмма стадий выборки и диспетчеризации команды 8000000с на 2-й итерации цикла. Выборка происходит в 27 такте, так так сигнал fetch_complete сброшен в 0, диспетчеризация заканчивается в 29 (заканчивается записью 0000а103 в instruction_table).

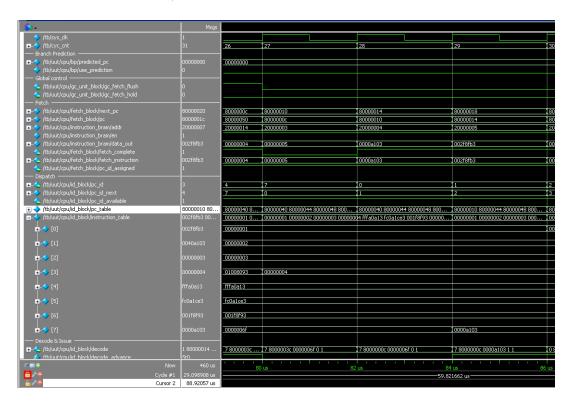


Рисунок 2.1 – Временная диаграмма выполнения стадий выборки и диспетчеризации команды 8000000с на 2-й итерации

На рисунке ниже рассмотренна временная диаграмма стадии декодирования и планирования на выполнение команды 80000018 на 2-й итерации цикла. В 34 такте происходит декодирование команды, в 35 она планируется на выполнение, возникает конфликт по регистру rs2, в 36 такте завершается выполнение предыдущей команды.

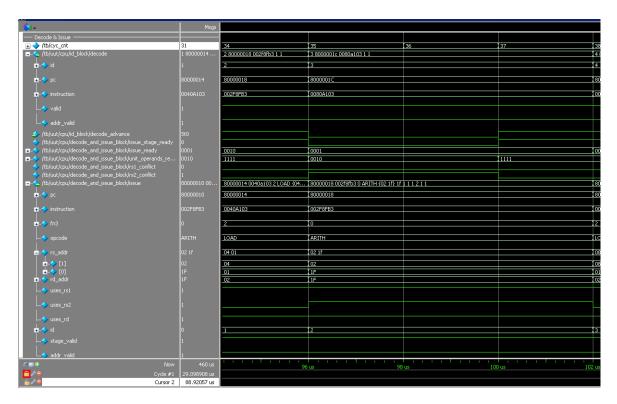


Рисунок 3.1 – Временная диаграмма выполнения стадий декодирования и планирования на выполнение команды 80000018 на 2-й итерации

На рисунке ниже рассмотренна временная диаграмма стадии выполнения команды 8000002с на 1-й итерации цикла. Видно, что в 23 такте данная команда поступает на выполнение, так как это арифметическая операция, она выполняется блоком ALU, в unit_wb[0]/id записывается 3 (как и у исследуемой команды), сигнал unit_wb[0]/done выставлен в 1. Следовательно выполнение данной команды завершается в этом же такте.

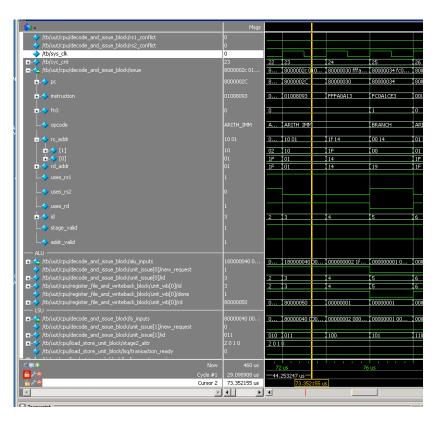


Рисунок 4.1 – Временная диаграмма выполнения стадии выполнения команды 8000002с на 1-й итерации

Для начала узнаем значение регистра x31, из рисунка ниже видно, что вычисленное в первом задании значение совпадает с хранящимся там

$$00000025(hex) = 37(dec) \tag{5.1}$$



Рисунок 5.1 – Значение регистра х31 после выполнения программы

Символом #! в программе помечена команда addi x31, x31, x2. Рассмотрим стадии ее выполнения. На рисунке 5.1 представлены стадии выборки (6 такт), диспетчеризации (7 такт) и декодирования (8 такт).

1.	Msgs				
♦ /tb/sys_clk	0				
⊨ √ /tb/cyc_cnt	9	6	7	[8	=
├� /tb/uut/cpu/bp/predicted_pc	00000000	00000000			
/tb/uut/cpu/bp/use prediction	0				
- Global control					
/tb/uut/cpu/gc_unit_block/gc_fetch_flush	0				
/tb/uut/cpu/gc_unit_block/gc_fetch_hold	0				
Fetch —					
	80000024	80000018	18000001c	180000020	
	80000020	80000014	180000018	18000001c	
- /tb/uut/cpu/instruction bram/addr	20000008	20000005	20000006	20000007	
4 /tb/uut/cpu/instruction bram/en					
	0080a203	0040a183	1002f8fb3	1003f8fb3	
4 /tb/uut/cpu/fetch block/fetch complete					
-4 /tb/uut/cpu/fetch_block/fetch_instruction	0080a203	0040a183	1002f8fb3	I 003f8fb3	
4 /tb/uut/cpu/fetch block/pc id assigned					Ī
Dispatch —					
-4 /tb/uut/cpu/id_block/pc_id	0	5	6	17	
- /tb/uut/cpu/id block/pc id next		6	7	lo	Ē
4 /tb/uut/cpu/id block/pc id available					
⊢/tb/uut/cpu/id_block/pc_table	80000000 80000	80000000 80	. 180000000 80000004 80	180000000 80000004 80	
- /tb/uut/cpu/id block/instruction table	00000097 03c08		. 100000097 03c08093 02		
- Decode & Issue					
-4. /tb/uut/cpu/id_block/decode	6 80000018 003f	3 8000000c	I 4 80000010 0040a183 1 1	I 5 80000014 002f8fb3 1 1	
I		3	14	-	
	P	3	4	5	
■-	80000018	8000000C	[80000010	[80000014	
•	003F8FB3	0000A103	0040A183	1002F8FB3	
└→ valid					
addr_valid					
/b/uut/cpu/id block/decode advance	StO				
tb/uut/cpu/decode_and_issue_block/issue_stage_ready	n -				
tb/uut/cpu/decode and issue block/issue ready	0001	0001	[0010		
tb/uut/cpu/decode and issue block/unit operands re		1111	,00.0		
/tb/uut/cpu/decode_and_issue_block/rs1_conflict	0				Ē
/tb/uut/cpu/decode and issue block/rs2 conflict	1				
. Something the state of the s	460 us	1 1 1	10 us	1 1 1 1	44
/ ⊖ Cycle #1	29.098908 us			32829 us	
Cursor 2	_		10.2		

Рисунок 5.2 – Выборка, диспетчеризация и декодирование команды

В 9 такте команда планируется на выполнение, но так как сигнал rs2_conflict равен 1 возникает конфликт. В следующем такте этот конфликт разрешается, управление передается блоку ALU, о чем свидетельтсвует совпадение

issue/id и unit_wb[0]/id, они равны 5. В 10 такте выполнение завершается, о чем свидетельствует 1 на сигнале unit_wb[0]/done, и в следующем такте переходит к выполнению следующей команды.

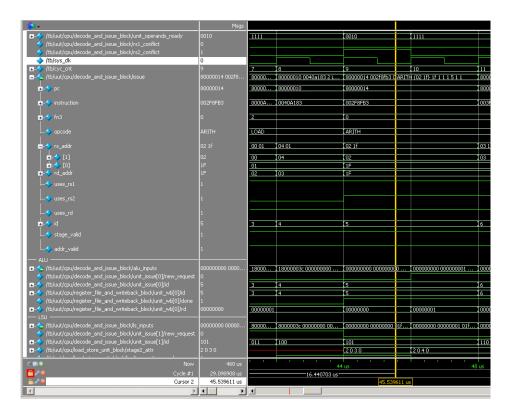


Рисунок 5.3 – Планирование на выполнение и выполнение команды

Ниже представленна временная диаграмма сигналов выполнения программы по варианту 12.

Адрес	Код команды	Команда	id																			мер				_	_	_			_	_	_									
		**	id			4	5	6	7 8	3 9	10	11	12	13	14 1	5 10	6 17	18	19	20 2	21 2	2 23	3 24	25	26	27	28 2	29 3	0 3	1 32	33	34	35 3	6 37	7 38	39	40	41	42	43 4	44	45
80000000<_start>	00000097	auipc x1,0x0	0		ED E						н																															
80000004	03c08093	addi x1,x1,60#8000003c<_x>	1		FΙ				_		ш							_			_							_								ш	ш					_
80000008	00200a13	addi x20,x1,32	2		F		D .																																			
8000000c <lp></lp>	0000a103	lw x2,0(x1)	3			F	ID																																			
80000010	0040a183	lw x3,4(x1)	4				F	ID [D M	1 M2	2 МЗ																	П														
80000014	002f8fb3	add x31,x31,x2	5) C																																
80000018	003f8fb3	add x31,x31,x3	6						FI	D W	D	AL																П														
8000001c	0080a203	lw x4,8(x1)	7			Т			F	10	W	D	M1 I	M2 I	М3	Т		Т					Т					Т							Т							П
80000020	00c0a283	lw x5,12(x1)	0							F	ID	W	D	M1	M2 M	3																										
80000024	004f8fb3	add x31,x31,x4	1	П	Т	Т		Т	Т	Т	F	ID	W	D	C A	L	Т	Т			Т	Т	Т				П	Т	Т	Т			Т	Т	Т				Т	Т	П	Т
80000028	005f8fb3	add x31,x31,x5	2									F	ID	W	WE) AI	L																									
8000002c	01008093	addi x1,x1,16	3	П	Т	т			т	Т	П	П	E :	ID	w v	V D	AL				т		Т	П			Т	Т	Т	Т				Т	Т				т	Т	П	П
80000030	fd409ee3	bne x1,x20,8000000c <lp></lp>	4								L				ID V																											
80000034	001f8f93	addi x31,x31,1	5													D W		D	x																							7
80000034	0000006f	jal x0,80000038 <lp2></lp2>	6								1						D W																									
8000003c	00000001	<pre><invalid operation=""></invalid></pre>	7												-1'		ID																									
80000030	00000001	<invalid operation=""></invalid>	0								Н					ı.		ID																								
80000044	00000002	<pre><invalid operation=""></invalid></pre>	1								Н		Н				Η.	F	X																							7
80000044											н							F	FX																							
	00000004	<invalid operation=""></invalid>	2								н								FX																							4
8000000c <lp></lp>	0000a103	lw x2,0(x1)	6						-		н											M1						-														
80000010	0040a183	lw x3,4(x1)	7								н										_	D D	-	-																		4
80000014	002f8fb3	add x31,x31,x2	0						_		ш							_			F	ID						_													_	_
80000018	003f8fb3	add x31,x31,x3	1								Н											F			D																	
8000001c	0080a203	lw x4,8(x1)	2								ш												F				м1 г															_
80000020	00c0a283	lw x5,12(x1)	3																					F					12 M													
80000024	004f8fb3	add x31,x31,x4	4																						F		W		C A													
80000028	005f8fb3	add x31,x31,x5	5																							F	ID			AL												
8000002c	01008093	addi x1,x1,16	6																											/ D												
8000030	fd409ee3	bne x1,x20,8000000c <lp></lp>	7								П																	FI	DW	W	D	В										
8000000c <lp></lp>	0000a103	lw x2,0(x1)	0								П																	1	F II	w	W	D	Х		Т							П
80000010	0040a183	lw x3,4(x1)	1																										F	ID	W	W	DX									
80000014	002f8fb3	add x31,x31,x2	2	П	Т	Т		Т	Т	Т	П	П	П	\neg		Т	Т	Т			Т	Т	Т				П	Т	Т	F	ID	w	Х	Т	Т				Т	Т	П	Т
80000018	003f8fb3	add x31,x31,x3	3																												F	ID	х									
8000001c	0080a203	lw x4,8(x1)	4			Т			т		П		П			Т		Т					Т				П	Т				F	х	Т	Т				т		П	П
80000020	θθcθa283	lw x5,12(x1)	5																														FX									
80000034	001f8f93	add x31,x31,1	1								П							т					т				7	7						- 10	חומ	ΔΙ			7		7	т
80000038	0000006f	jal x0,80000038 <lp2></lp2>	2																																ID		В					
8000003c	00000001	<pre><invalid operation=""></invalid></pre>	3																															1		ID		Y				
80000030	00000001	<invalid operation=""></invalid>	4								1																								1		ID					
80000040	00000002	<pre><invalid operation=""></invalid></pre>	5																																	1	F					
80000044	00000003	<pre><invalid operation=""></invalid></pre>	6								١.																											FX				
	00000004 0000006f		4																																				F I	ID	D	
80000038 <lp2></lp2>		jal x0,8000003c <forever></forever>	5								1																												_	_	_	_
80000038 <lp2></lp2>	0000006f	jal x0,8000003c <forever></forever>	э	,	2		Ļ	_			120		10	12	14 1	- 1		110	10	20.5	11 0	2 22	104	25	26	27	20.5	000	0 2	1 22	22	-	25 2	6 2	7 20	20	40	-				D
Адрес	Код команды	Команда	lid	Ţ	Z :	4	5	0	/ 8	5 <u>9</u>	110	111	12	13.	14 1	5 I(0 1 /	118	19	20 2	1 2	2 23	5 24	25	26	21	28 2	29 3	10 3	1 32	33	34	30 J	0 3	/ 38	39	40	41	42 4	43 4	44	40

Рисунок 5.4 – Временная диаграмма сигналов выполнения программы

Из трассы видно, что выполнение программы заканчивается выполнением команды add x31, x31, 1 в 39 такте. Тоесть выполнение программы происходит с 1 по 39 такт. Из них планирование на выполнение не происходило в течение 16 тактов (7, 8, 12, 13, 19 - 24, 28, 29, 35 - 38). Это составляет 41%. Чтобы сократить время простоя можно провести оптимизацию путем перестановки команд.

В качестве оптимизации было выбрано перемещение строк загрузки данных друг к другу. Оптимизированный код представлен на листинге ниже.

Листинг 5.1 – Код оптимизированной программы

```
.section .text
           .globl _start;
           len = 8 #array length
           enroll = 4 #count of elems in one loop
           elem_sz = 4 #size of one element
  _start:
           la x1, _x
           addi x20, x1, elem\_sz*len #address of last elem
10 lp:
           lw x2, 0(x1)
11
           lw x3, 4(x1)
12
           lw x4, 8(x1)
13
           lw x5, 12(x1)
14
           add x31, x31, x2 #!
15
           add x31, x31, x3
           add x31, x31, x4
17
           add x31, x31, x5
18
           addi x1, x1, elem_sz*enroll
           bne x1, x20, lp
20
           addi x31, x31, 1
21
22 1p2: j 1p2
23
           .section .data
24
           .4byte 0x1
25
  _x:
           .4byte 0x2
26
           .4byte 0x3
^{27}
           .4byte 0x4
28
           .4byte 0x5
29
           .4byte 0x6
30
           .4byte 0x7
31
           .4byte 0x8
32
```

Временная диаграмма сигналов выполнения представленна ниже.

Адрес	Код команды	Команда	id	1 1	2 2	-	-	_	7	0	0.1	10/	11/2	100	2 -	4 2	E 4	C 1	7 1 1	1 1 2			Ta			or!	20 0	77.0	0 0	010	0 2 2	130	22	24	25/2	1010	77 0	ola.	оT:		
800000000< start>	00000097	auipc x1.0x0	0	1 F I				6	7	8	9	10	11	12 1	3 1	4 1	5 1	ь 1	/ 18	3 19	20	21	22	23	24	25	26 2	1/2	8 2	9 3	9 31	1 32	33	34	35 3	36	37 3	8 3	94	0 4	14
_		addi x1,x1,60#8000003c< x>	0			D D																								1								+			
80000004 80000008	03c08093 00200a13	addi x1,x1,60#8000003c<_x> addi x20,x1,32	1			II		AL																						ı											
			2		1	100	ID														Н									н								+	4		-
8000000c <lp></lp>	0000a103	lw x2,0(x1)	3			F			D																					٠		-						_	4		-
80000010	0040a183	lw x3,4(x1)	4				F				_	_									Н									н								+	4		
80000014	0080a203	lw x4,8(x1)	5					F	ID												ш								-	-		-					-	_	4		_
80000018	00c0a283	lw x5,12(x1)	6							ID				43							Н									н								4	4		
8000001c	002f8fb3	add x31,x31,x2	7	_						F					-	-		-			ш							-	4	-		4					4	4	4	4	_
80000020	003f8fb3	add x31,x31,x3	0								F																			н								4	4		
80000024	004f8fb3	add x31,x31,x4	1	_							_			D A		4		_		_	ш							_	4	_		ш				_	4	_	4	4	_
80000028	005f8fb3	add x31,x31,x5	2											ID																н									4		
8000002c	01008093	addi x1,x1,16	3											F I																\perp		┺						_	1		_
80000030	fd409ee3	bne x1,x20,8000000c <lp></lp>	4													DI														1											
80000034	001f8f93	addi x31,x31,1	5								_				- 1))											_	_		┸					_	_	_	_	_
80000038	0000006f	jal x0,80000038 <lp2></lp2>	6															D X	_																				4		
8000003c	00000001	<invalid operation=""></invalid>	7								_						- 1	· >			_								_	_		┸					_	_	_	_	_
80000040	00000002	<invalid operation=""></invalid>	0															F.																					4		
8000000c <lp></lp>	0000a103	lw x2,0(x1)	6																F				M2									L						\perp	1		\perp
80000010	0040a183	lw x3,4(x1)	7																	F			M1																4		
80000014	0080a203	lw x4,8(x1)	0																		F				M2							L						\perp	┙		
80000018	00c0a283	lw x5,12(x1)	1																		L	F	ID	D	M1	M2	13			Т									4		
8000001c	002f8fb3	add x31,x31,x2	2																				F	ID	D	AL													┸		
80000020	003f8fb3	add x31,x31,x3	3								- 1										ı			F	ID	D	AL.			П									4		
80000024	004f8fb3	add x31,x31,x4	4																						F	ID	D A	IL.		Т								Т	Т		Т
80000028	005f8fb3	add x31,x31,x5	5																		ı					F :	ED I	D A	L	Т											
8000002c	01008093	addi x1,x1,16	6								П																F I	D) A	L	Т							Т	Т		Т
80000030	fd409ee3	bne x1,x20,8000000c <lp></lp>	7																		1							FI	D	В											
8000000c <lp></lp>	0000a103	lw x2,0(x1)	0		Т	Т					П	П		Т	Т	Т	Т	Т	Т	Т	Г							- 1	I	D D	X	Г			\neg	П	Т	Т	Т	Т	Т
80000010	0040a183	lw x3,4(x1)	1																											11	X										
80000014	0080a203	lw x4,8(x1)	2		Т	Т					П	П		Т	Т		Т	Т	Т	Т									Т	F	X	Г			\neg	Т	Т	Т	Т	Т	Т
80000018	00c0a283	lw x5,12(x1)	3																												F	(
80000034	001f8f93	add x31,x31,1	1	Т	Т	Т					П			Т	Т	Т	т	Т	Т	Т	П						Т	Т	Т	Т	Т	F	ID	D	AL	Т	Т	Т	Т	Т	Т
80000038	0000006f	jal x0,80000038 <lp2></lp2>	2												ı																		F	ID	D	В					
8000003c	00000001	<invalid operation=""></invalid>	3												Т		T				Г									T	Т			F	ID	D	х	Т	Т	T	Т
80000040	00000002	<invalid operation=""></invalid>	4															Ĺ		Ĺ										Ĺ					F]						
80000044	00000003	<invalid operation=""></invalid>	5												Т	Т	T				Г									T						F		Т	Т	Т	Т
80000048	00000004	<invalid operation=""></invalid>	6																																		Х		ġ.		
80000038 <lp2></lp2>	0000006f	jal x0,8000003c <forever></forever>	4												T						Г									T								FΙ	D I	D	в
80000038 <lp2></lp2>	0000006f	jal x0,8000003c <forever></forever>	5															Ĺ	İ	Ĺ	L									Ĺ	Ĺ										D E
•		Команда		1	2 3	4	5	6	7	8	9	10	11 :	12 1	3 1	4 1	5 1	6 1	7 18	3 19	20	21	22	23	24	25	26 2	7 2	8 2	9 3	9 31	32	33	34	35 3	36	37 3	8 3			
Адрес	Код команды	команда	Ta								_												та				_					_						_	_	_	_

Рисунок 5.5 – Временная диаграмма сигналов выполнения оптимизированной программы

Из трассы видно, что выполнение программы заканчивается выполнением команды add x31, x31, 1 в 35 такте. Тоесть выполнение программы происходит с 1 по 35 такт. Из них планирование на выполнение не происходило в течение 12 тактов (7, 8, 17 - 22, 31 - 34). Это составляет 31%. Удалось сократить простой на 10%.