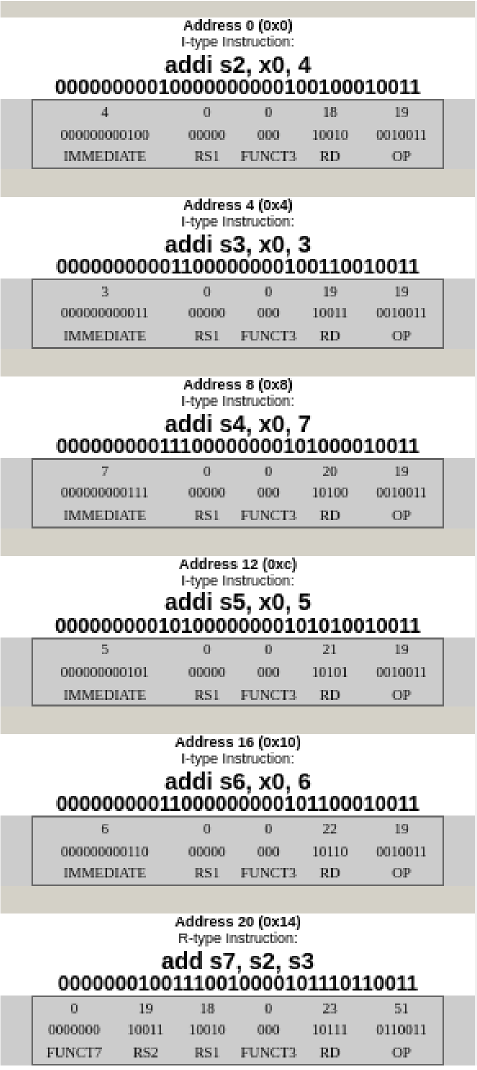
Roteiro 8 - Laboratório de Organização e Arquitetura de Computadores

Aluno: Tarso Jabbes Lima de Oliveira

RESPOSTAS PARA O PROGRAMA 1

1.2.a

Conteúdo da memória de instruções e dos registradores no início da execução:



Address 0 (0x0)
I-type Instruction:
addi s2, x0, 4
00000000010000000000100100010011

4	0	0	18	19
000000000100	00000	000	10010	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 4 (0x4)
I-type Instruction:
addi s3, x0, 3
00000000001100000000100110010011

3	0	0	19	19
000000000011	00000	000	10011	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 8 (0x8)
I-type Instruction:
addi s4, x0, 7
00000000011100000000101000010011

7	0	0	20	19
000000000111	00000	000	10100	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 12 (0xc)
I-type Instruction:
addi s5, x0, 5
00000000010100000000101010010011

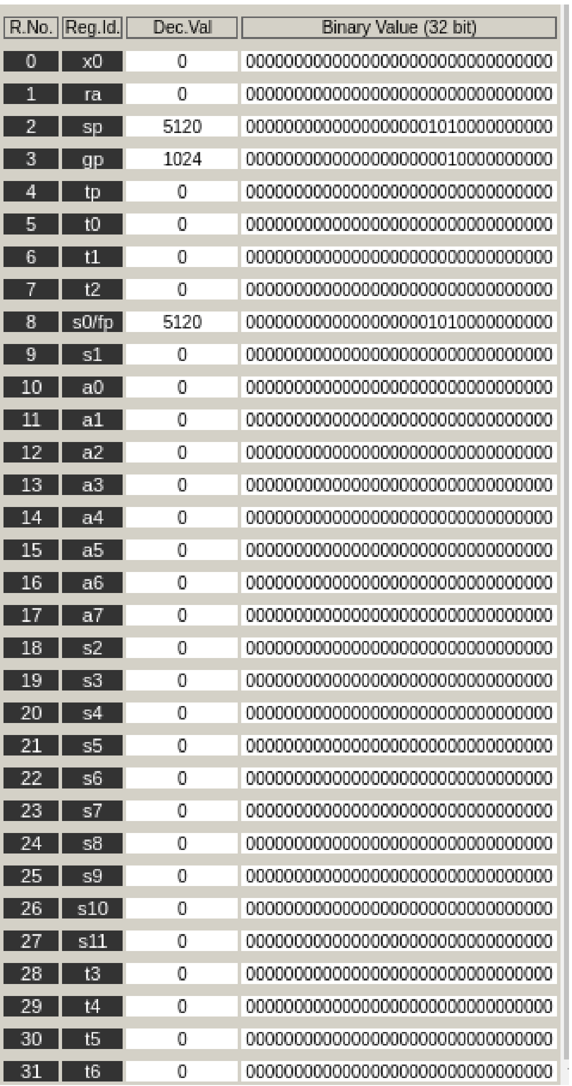
5	0	0	21	19
000000000101	00000	000	10101	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 16 (0x10)
I-type Instruction:
addi s6, x0, 6
00000000011000000000101100010011

6	0	0	22	19
000000000110	00000	000	10110	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 20 (0x14)
R-type Instruction:
add s7, s2, s3
00000001001110010000101110110011

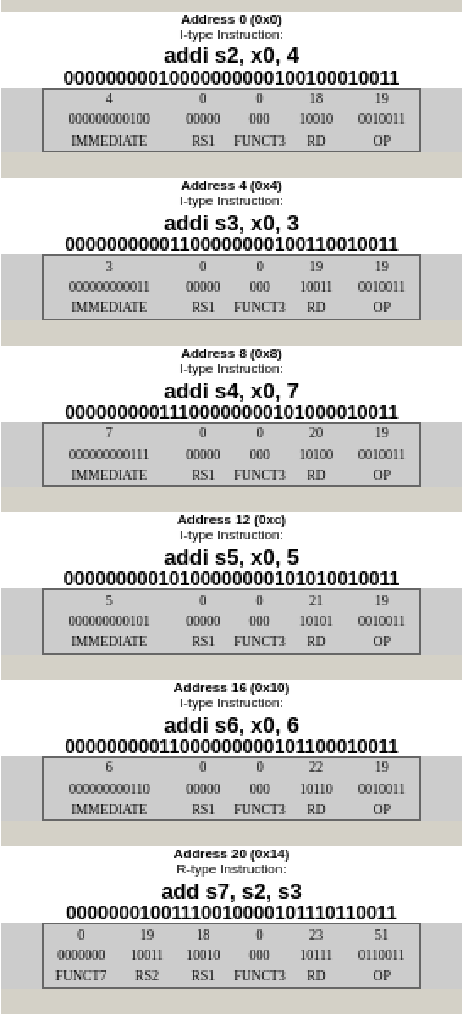
0	19	18	0	23	51
0000000	10011	10010	000	10111	0110011
FUNCT7	RS2	RS1	FUNCT3	RD	OP



R.No.	Reg.Id.	Dec.Val	Binary Value (32 bit)
0	x0	0	00000000000000000000000000000000
1	ra	0	00000000000000000000000000000000
2	sp	5120	00000000000000000001010000000000
3	gp	1024	00000000000000000000010000000000
4	tp	0	00000000000000000000000000000000
5	t0	0	00000000000000000000000000000000
6	t1	0	00000000000000000000000000000000
7	t2	0	00000000000000000000000000000000
8	s0/fp	5120	00000000000000000001010000000000
9	s1	0	00000000000000000000000000000000
10	a0	0	00000000000000000000000000000000
11	a1	0	00000000000000000000000000000000
12	a2	0	00000000000000000000000000000000
13	a3	0	00000000000000000000000000000000
14	a4	0	00000000000000000000000000000000
15	a5	0	00000000000000000000000000000000
16	a6	0	00000000000000000000000000000000
17	a7	0	00000000000000000000000000000000
18	s2	0	00000000000000000000000000000000
19	s3	0	00000000000000000000000000000000
20	s4	0	00000000000000000000000000000000
21	s5	0	00000000000000000000000000000000
22	s6	0	00000000000000000000000000000000
23	s7	0	00000000000000000000000000000000
24	s8	0	00000000000000000000000000000000
25	s9	0	00000000000000000000000000000000
26	s10	0	00000000000000000000000000000000
27	s11	0	00000000000000000000000000000000
28	t3	0	00000000000000000000000000000000
29	t4	0	00000000000000000000000000000000
30	t5	0	00000000000000000000000000000000
31	t6	0	00000000000000000000000000000000

Podemos observar que os únicos valores nos registradores ao início são o de Stack Pointer, Group Pointer e o s0/fp. Além disso, as instruções foram carregadas corretamente na memória de instruções.

Conteúdo da memória de instruções e dos registradores ao final da execução:



Address 0 (0x0)
I-type Instruction:
addi s2, x0, 4
00000000010000000000100100010011

IMMEDIATE	RS1	FUNCT3	RD	OP
4	0	0	18	19
000000000100	00000	000	10010	0010011

Address 4 (0x4)
I-type Instruction:
addi s3, x0, 3
00000000001100000000100110010011

IMMEDIATE	RS1	FUNCT3	RD	OP
3	0	0	19	19
000000000011	00000	000	10011	0010011

Address 8 (0x8)
I-type Instruction:
addi s4, x0, 7
00000000011100000000101000010011

IMMEDIATE	RS1	FUNCT3	RD	OP
7	0	0	20	19
000000000111	00000	000	10100	0010011

Address 12 (0xc)
I-type Instruction:
addi s5, x0, 5
00000000010100000000101010010011

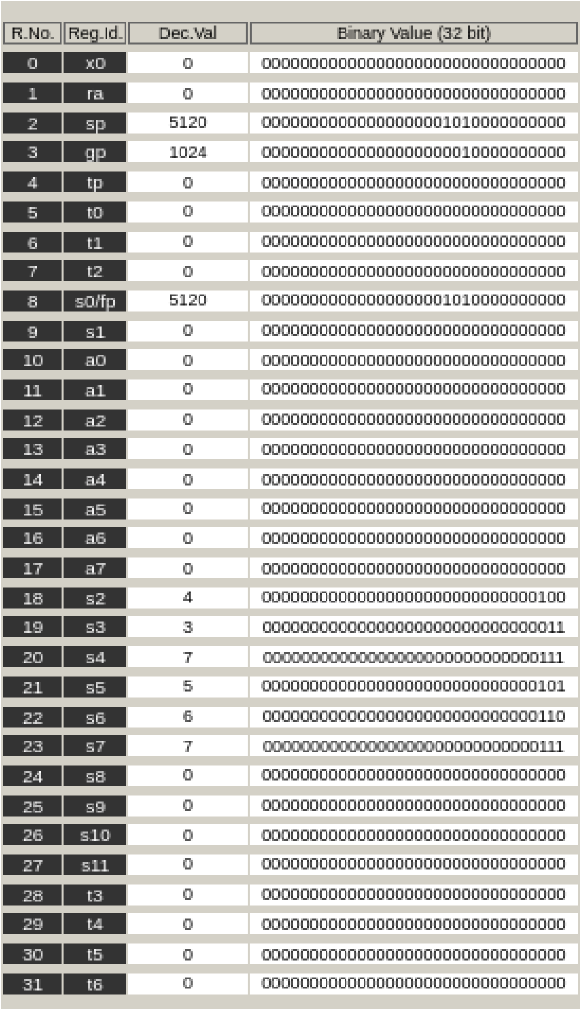
IMMEDIATE	RS1	FUNCT3	RD	OP
5	0	0	21	19
000000000101	00000	000	10101	0010011

Address 16 (0x10)
I-type Instruction:
addi s6, x0, 6
00000000011000000000101100010011

IMMEDIATE	RS1	FUNCT3	RD	OP
6	0	0	22	19
000000000110	00000	000	10110	0010011

Address 20 (0x14)
R-type Instruction:
add s7, s2, s3
00000001001110010000101110110011

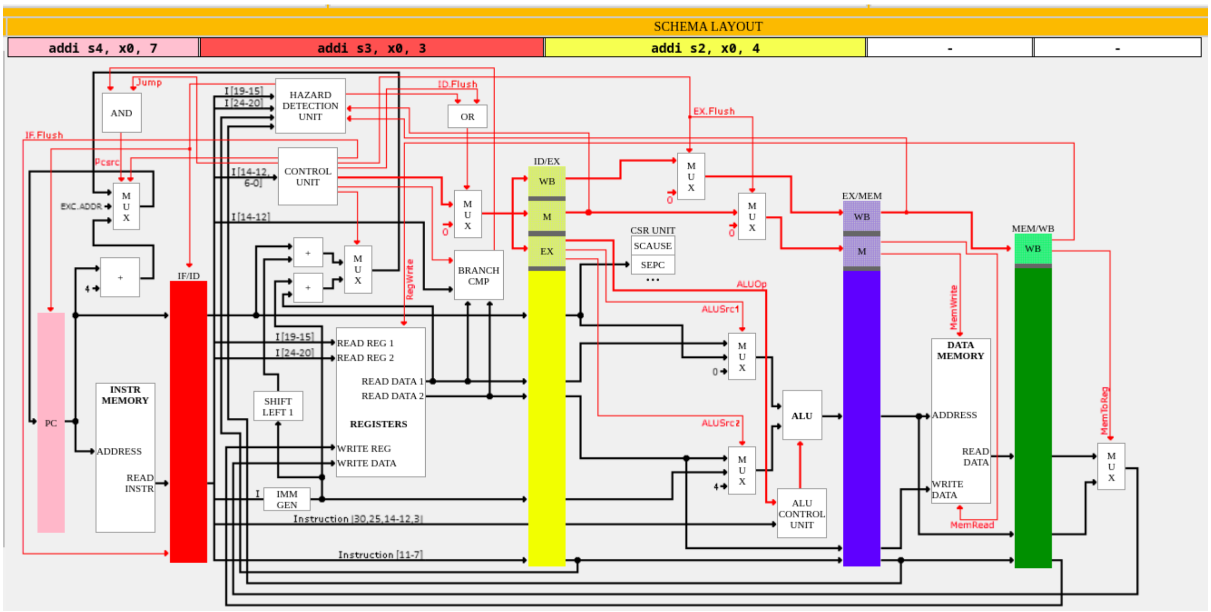
FUNCT7	RS2	RS1	FUNCT3	RD	OP
0	19	18	0	23	51
0000000	10011	10010	000	10111	0110011



R.No.	Reg.Id.	Dec.Val	Binary Value (32 bit)
0	x0	0	00000000000000000000000000000000
1	ra	0	00000000000000000000000000000000
2	sp	5120	00000000000000000001010000000000
3	gp	1024	00000000000000000000010000000000
4	tp	0	00000000000000000000000000000000
5	t0	0	00000000000000000000000000000000
6	t1	0	00000000000000000000000000000000
7	t2	0	00000000000000000000000000000000
8	s0/fp	5120	00000000000000000001010000000000
9	s1	0	00000000000000000000000000000000
10	a0	0	00000000000000000000000000000000
11	a1	0	00000000000000000000000000000000
12	a2	0	00000000000000000000000000000000
13	a3	0	00000000000000000000000000000000
14	a4	0	00000000000000000000000000000000
15	a5	0	00000000000000000000000000000000
16	a6	0	00000000000000000000000000000000
17	a7	0	00000000000000000000000000000000
18	s2	4	00000000000000000000000000000100
19	s3	3	00000000000000000000000000000011
20	s4	7	00000000000000000000000000000111
21	s5	5	00000000000000000000000000000101
22	s6	6	00000000000000000000000000000110
23	s7	7	00000000000000000000000000000111
24	s8	0	00000000000000000000000000000000
25	s9	0	00000000000000000000000000000000
26	s10	0	00000000000000000000000000000000
27	s11	0	00000000000000000000000000000000
28	t3	0	00000000000000000000000000000000
29	t4	0	00000000000000000000000000000000
30	t5	0	00000000000000000000000000000000
31	t6	0	00000000000000000000000000000000

Ao final da execução, os registradores s2, s3, s4, s5, s6 e s7 sofreram alterações.

1.2.b Passagem em três estágios representativos do pipeline

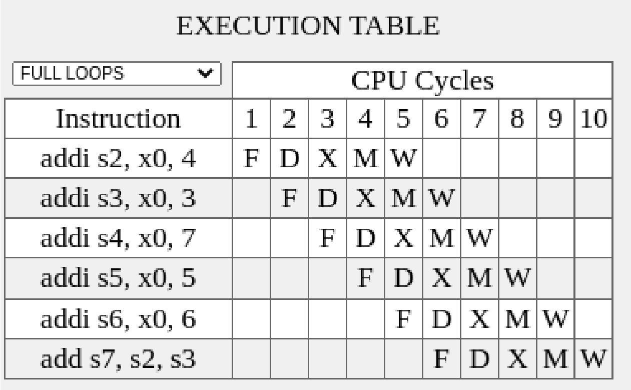


SCHEMA LAYOUT

addi s4, x0, 7	addi s3, x0, 3	addi s2, x0, 4	-	-

Podemos observar que a primeira instrução está no estágio 3 do pipeline (estágio de execução), a segunda instrução está no estágio 2 (estágio de decodificação) e a terceira instrução está no estágio 1 (estágio de *fetch*).

1.2.c



EXECUTION TABLE

FULL LOOPS

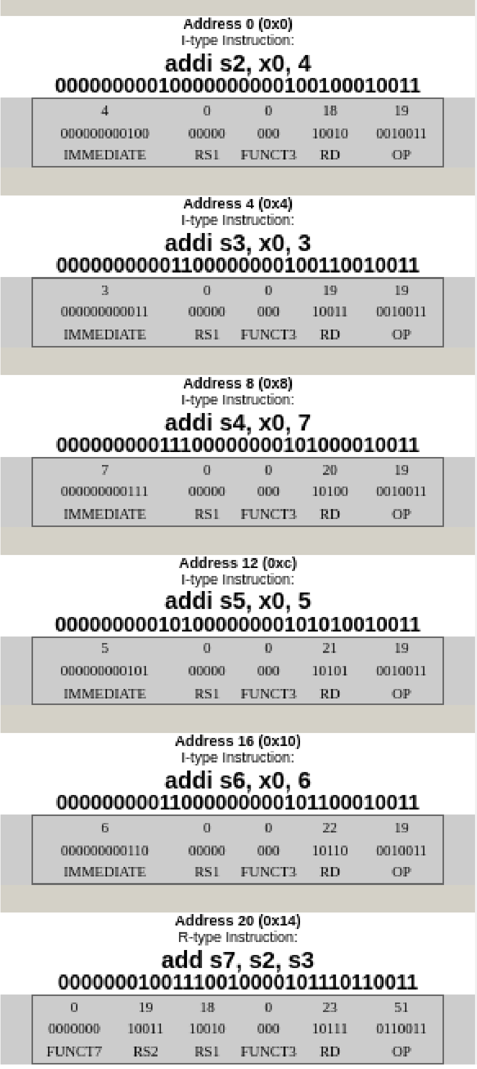
	CPU Cycles									
Instruction	1	2	3	4	5	6	7	8	9	10
addi s2, x0, 4	F	D	X	M	W					
addi s3, x0, 3		F	D	X	M	W				
addi s4, x0, 7			F	D	X	M	W			
addi s5, x0, 5				F	D	X	M	W		
addi s6, x0, 6					F	D	X	M	W	
add s7, s2, s3						F	D	X	M	W

A tabela mostra que não houve conflito na execução das instruções.

1.2.d Ao total, foram gastos 10 ciclos de CPU para que o programa fosse executado.

2.2.a

Conteúdo da memória de instruções e dos registradores no início da execução:



Address 0 (0x0)
I-type Instruction:
addi s2, x0, 4
00000000010000000000100100010011

4	0	0	18	19
000000000100	00000	000	10010	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 4 (0x4)
I-type Instruction:
addi s3, x0, 3
00000000001100000000100110010011

3	0	0	19	19
000000000011	00000	000	10011	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 8 (0x8)
I-type Instruction:
addi s4, x0, 7
00000000011100000000101000010011

7	0	0	20	19
000000000111	00000	000	10100	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 12 (0xc)
I-type Instruction:
addi s5, x0, 5
00000000010100000000101010010011

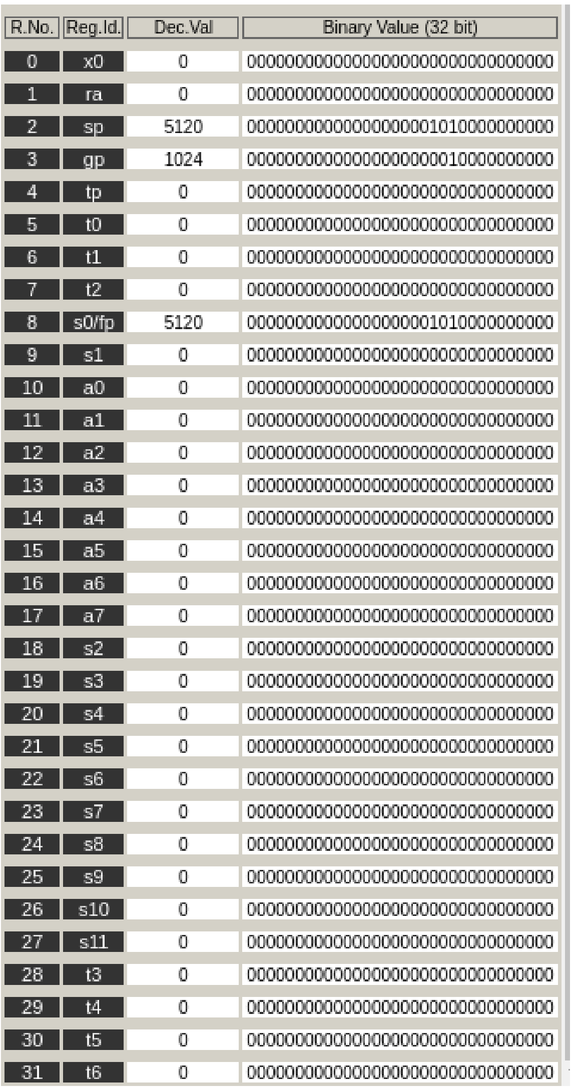
5	0	0	21	19
000000000101	00000	000	10101	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 16 (0x10)
I-type Instruction:
addi s6, x0, 6
00000000011000000000101100010011

6	0	0	22	19
000000000110	00000	000	10110	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

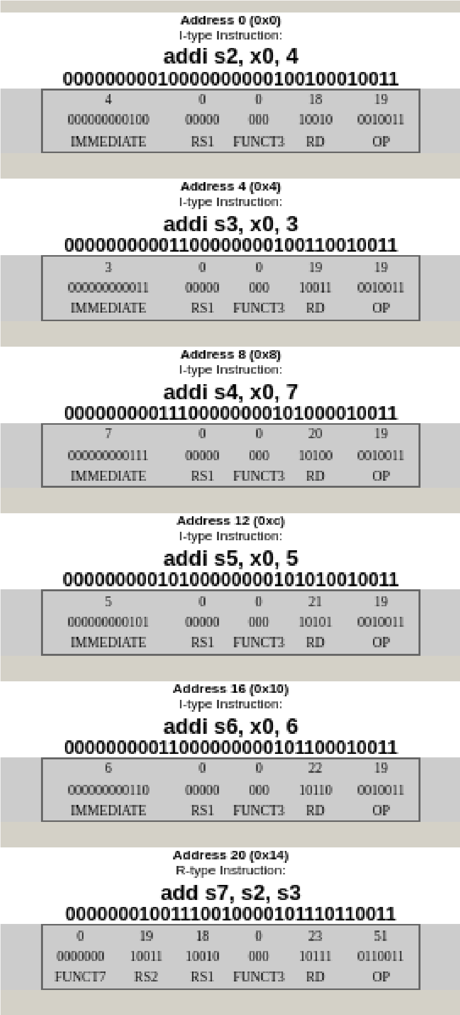
Address 20 (0x14)
R-type Instruction:
add s7, s2, s3
00000001001110010000101110110011

0	19	18	0	23	51
0000000	10011	10010	000	10111	0110011
FUNCT7	RS2	RS1	FUNCT3	RD	OP



R.No.	Reg.Id.	Dec.Val	Binary Value (32 bit)
0	x0	0	00000000000000000000000000000000
1	ra	0	00000000000000000000000000000000
2	sp	5120	00000000000000000001010000000000
3	gp	1024	00000000000000000000010000000000
4	tp	0	00000000000000000000000000000000
5	t0	0	00000000000000000000000000000000
6	t1	0	00000000000000000000000000000000
7	t2	0	00000000000000000000000000000000
8	s0/fp	5120	00000000000000000001010000000000
9	s1	0	00000000000000000000000000000000
10	a0	0	00000000000000000000000000000000
11	a1	0	00000000000000000000000000000000
12	a2	0	00000000000000000000000000000000
13	a3	0	00000000000000000000000000000000
14	a4	0	00000000000000000000000000000000
15	a5	0	00000000000000000000000000000000
16	a6	0	00000000000000000000000000000000
17	a7	0	00000000000000000000000000000000
18	s2	0	00000000000000000000000000000000
19	s3	0	00000000000000000000000000000000
20	s4	0	00000000000000000000000000000000
21	s5	0	00000000000000000000000000000000
22	s6	0	00000000000000000000000000000000
23	s7	0	00000000000000000000000000000000
24	s8	0	00000000000000000000000000000000
25	s9	0	00000000000000000000000000000000
26	s10	0	00000000000000000000000000000000
27	s11	0	00000000000000000000000000000000
28	t3	0	00000000000000000000000000000000
29	t4	0	00000000000000000000000000000000
30	t5	0	00000000000000000000000000000000
31	t6	0	00000000000000000000000000000000

Como não há hazards, os resultados permaneceram os mesmo com o Forwarding ativo.
Conteúdo da memória de instruções e dos registradores ao final da execução:



Address 0 (0x0)
I-type Instruction:
addi s2, x0, 4
00000000010000000000100100010011

4	0	0	18	19
000000000100	00000	000	10010	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 4 (0x4)
I-type Instruction:
addi s3, x0, 3
00000000001100000000100110010011

3	0	0	19	19
000000000011	00000	000	10011	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 8 (0x8)
I-type Instruction:
addi s4, x0, 7
00000000011100000000101000010011

7	0	0	20	19
000000000111	00000	000	10100	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 12 (0xc)
I-type Instruction:
addi s5, x0, 5
00000000010100000000101010010011

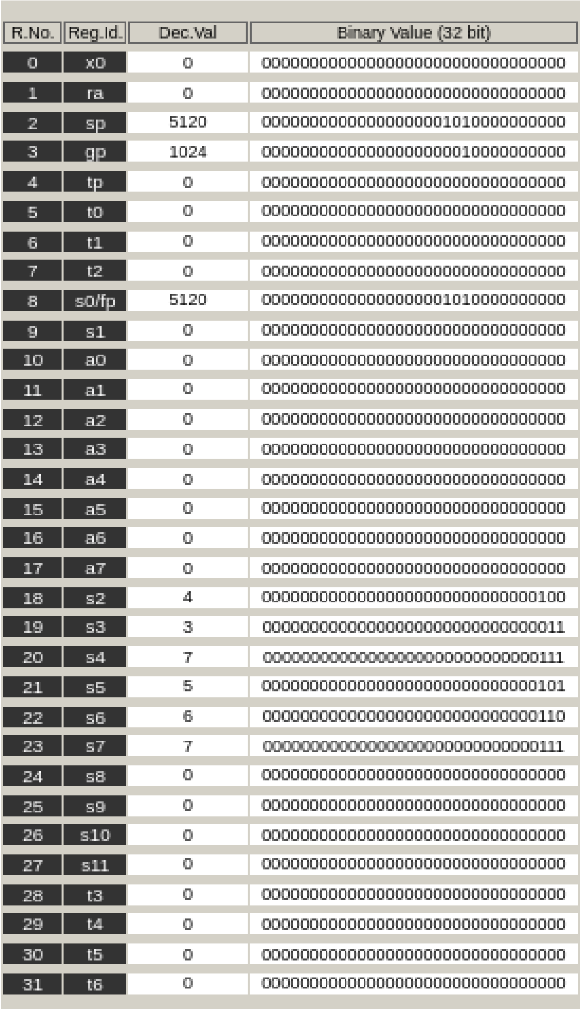
5	0	0	21	19
000000000101	00000	000	10101	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 16 (0x10)
I-type Instruction:
addi s6, x0, 6
00000000011000000000101100010011

6	0	0	22	19
000000000110	00000	000	10110	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 20 (0x14)
R-type Instruction:
add s7, s2, s3
00000001001110010000101110110011

0	19	18	0	23	51
0000000	10011	10010	000	10111	0110011
FUNCT7	RS2	RS1	FUNCT3	RD	OP

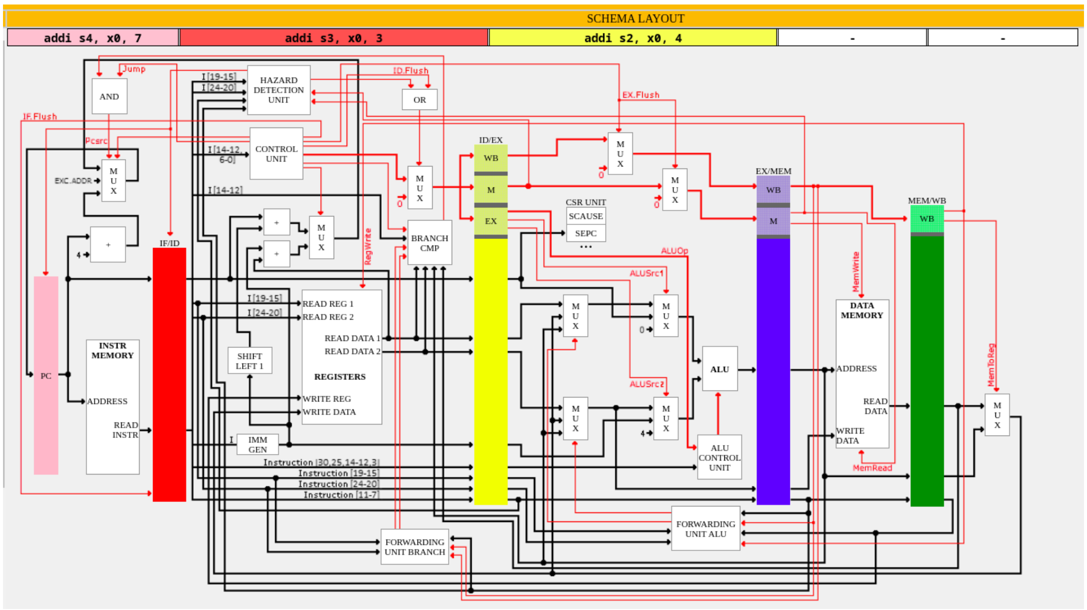


R.No.	Reg.Id.	Dec.Val	Binary Value (32 bit)
0	x0	0	00000000000000000000000000000000
1	ra	0	00000000000000000000000000000000
2	sp	5120	00000000000000000001010000000000
3	gp	1024	00000000000000000000010000000000
4	tp	0	00000000000000000000000000000000
5	t0	0	00000000000000000000000000000000
6	t1	0	00000000000000000000000000000000
7	t2	0	00000000000000000000000000000000
8	s0/fp	5120	00000000000000000001010000000000
9	s1	0	00000000000000000000000000000000
10	a0	0	00000000000000000000000000000000
11	a1	0	00000000000000000000000000000000
12	a2	0	00000000000000000000000000000000
13	a3	0	00000000000000000000000000000000
14	a4	0	00000000000000000000000000000000
15	a5	0	00000000000000000000000000000000
16	a6	0	00000000000000000000000000000000
17	a7	0	00000000000000000000000000000000
18	s2	4	00000000000000000000000000000100
19	s3	3	00000000000000000000000000000011
20	s4	7	00000000000000000000000000000111
21	s5	5	00000000000000000000000000000101
22	s6	6	00000000000000000000000000000110
23	s7	7	00000000000000000000000000000111
24	s8	0	00000000000000000000000000000000
25	s9	0	00000000000000000000000000000000
26	s10	0	00000000000000000000000000000000
27	s11	0	00000000000000000000000000000000
28	t3	0	00000000000000000000000000000000
29	t4	0	00000000000000000000000000000000
30	t5	0	00000000000000000000000000000000
31	t6	0	00000000000000000000000000000000

Os mesmos valores encontrados com o Forwarding desativados foram encontrados agora.

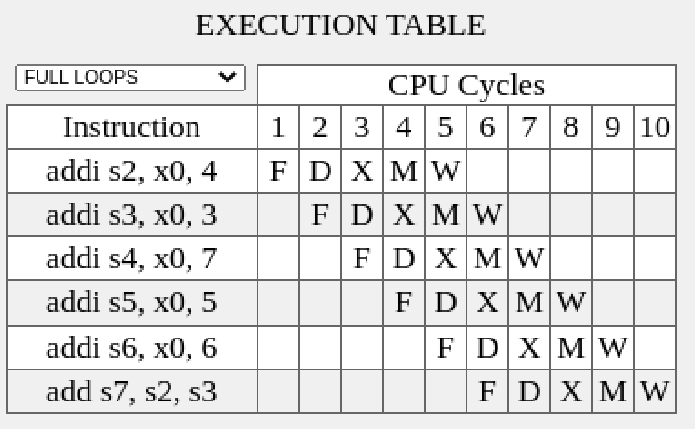
2.2.b

Passagem em três estágios representativos do pipeline



Encontramos os mesmos resultados que anteriormente: a primeira instrução está no estágio 3 do pipeline (estágio de execução), a segunda instrução está no estágio 2 (estágio de decodificação) e a terceira instrução está no estágio 1 (estágio de *fetch*).

2.2.c



EXECUTION TABLE

FULL LOOPS

	CPU Cycles									
Instruction	1	2	3	4	5	6	7	8	9	10
addi s2, x0, 4	F	D	X	M	W					
addi s3, x0, 3		F	D	X	M	W				
addi s4, x0, 7			F	D	X	M	W			
addi s5, x0, 5				F	D	X	M	W		
addi s6, x0, 6					F	D	X	M	W	
add s7, s2, s3						F	D	X	M	W

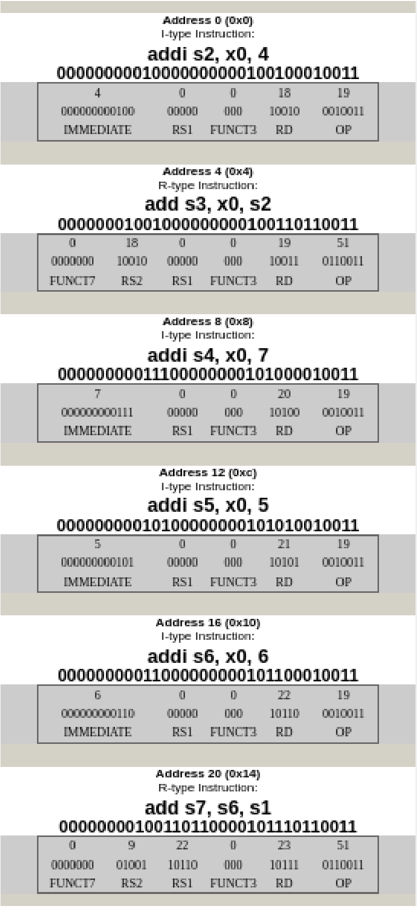
A tabela de execução mostra que não houve conflito durante a execução do programa 1.

2.2.d Ao total, também foram necessários 10 ciclos da CPU.

RESPOSTAS PARA O PROGRAMA 2

1.2.a

Conteúdo da memória de instruções e dos registradores no início da execução:



Address 0 (0x0)
I-type Instruction:
addi s2, x0, 4
00000000010000000000100100010011

4	0	0	18	19
000000000100	00000	000	10010	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 4 (0x4)
R-type Instruction:
add s3, x0, s2
00000001001000000000100110110011

0	18	0	0	19	51
0000000	10010	00000	000	10011	0110011
FUNCT7	RS2	RS1	FUNCT3	RD	OP

Address 8 (0x8)
I-type Instruction:
addi s4, x0, 7
00000000011100000000101000010011

7	0	0	20	19
000000000111	00000	000	10100	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 12 (0xc)
I-type Instruction:
addi s5, x0, 5
00000000010100000000101010010011

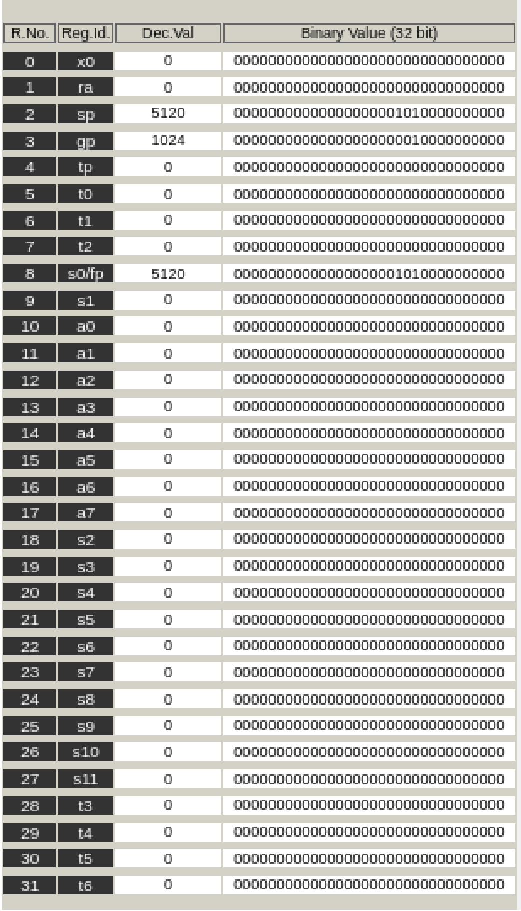
5	0	0	21	19
000000000101	00000	000	10101	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 16 (0x10)
I-type Instruction:
addi s6, x0, 6
00000000011000000000101100010011

6	0	0	22	19
000000000110	00000	000	10110	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 20 (0x14)
R-type Instruction:
add s7, s6, s1
00000000100110110000101110110011

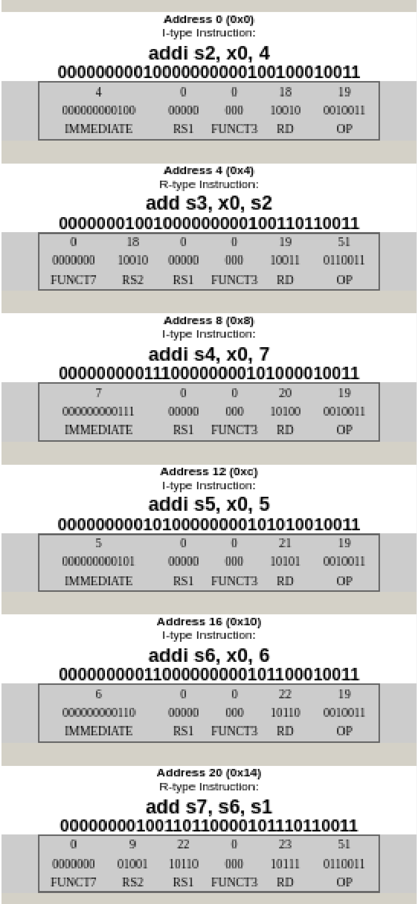
0	9	22	0	23	51
0000000	01001	10110	000	10111	0110011
FUNCT7	RS2	RS1	FUNCT3	RD	OP



R.No.	Reg.Id.	Dec.Val	Binary Value (32 bit)
0	x0	0	00000000000000000000000000000000
1	ra	0	00000000000000000000000000000000
2	sp	5120	00000000000000000001010000000000
3	gp	1024	00000000000000000000010000000000
4	tp	0	00000000000000000000000000000000
5	t0	0	00000000000000000000000000000000
6	t1	0	00000000000000000000000000000000
7	t2	0	00000000000000000000000000000000
8	s0/fp	5120	00000000000000000001010000000000
9	s1	0	00000000000000000000000000000000
10	a0	0	00000000000000000000000000000000
11	a1	0	00000000000000000000000000000000
12	a2	0	00000000000000000000000000000000
13	a3	0	00000000000000000000000000000000
14	a4	0	00000000000000000000000000000000
15	a5	0	00000000000000000000000000000000
16	a6	0	00000000000000000000000000000000
17	a7	0	00000000000000000000000000000000
18	s2	0	00000000000000000000000000000000
19	s3	0	00000000000000000000000000000000
20	s4	0	00000000000000000000000000000000
21	s5	0	00000000000000000000000000000000
22	s6	0	00000000000000000000000000000000
23	s7	0	00000000000000000000000000000000
24	s8	0	00000000000000000000000000000000
25	s9	0	00000000000000000000000000000000
26	s10	0	00000000000000000000000000000000
27	s11	0	00000000000000000000000000000000
28	t3	0	00000000000000000000000000000000
29	t4	0	00000000000000000000000000000000
30	t5	0	00000000000000000000000000000000
31	t6	0	00000000000000000000000000000000

Podemos observar que os únicos valores nos registradores ao início são o de Stack Pointer, Group Pointer e o s0/fp. Além disso, as instruções foram carregadas corretamente na memória de instruções.

Conteúdo da memória de instruções e dos registradores ao final da execução:



Address 0 (0x0)
I-type Instruction:
addi s2, x0, 4
00000000010000000000100100010011

4	0	0	18	19
000000000100	00000	000	10010	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 4 (0x4)
R-type Instruction:
add s3, x0, s2
00000001001000000000100110110011

0	18	0	0	19	51
0000000	10010	00000	000	10011	0110011
FUNCT7	RS2	RS1	FUNCT3	RD	OP

Address 8 (0x8)
I-type Instruction:
addi s4, x0, 7
00000000011100000000101000010011

7	0	0	20	19
000000000111	00000	000	10100	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 12 (0xc)
I-type Instruction:
addi s5, x0, 5
00000000010100000000101010010011

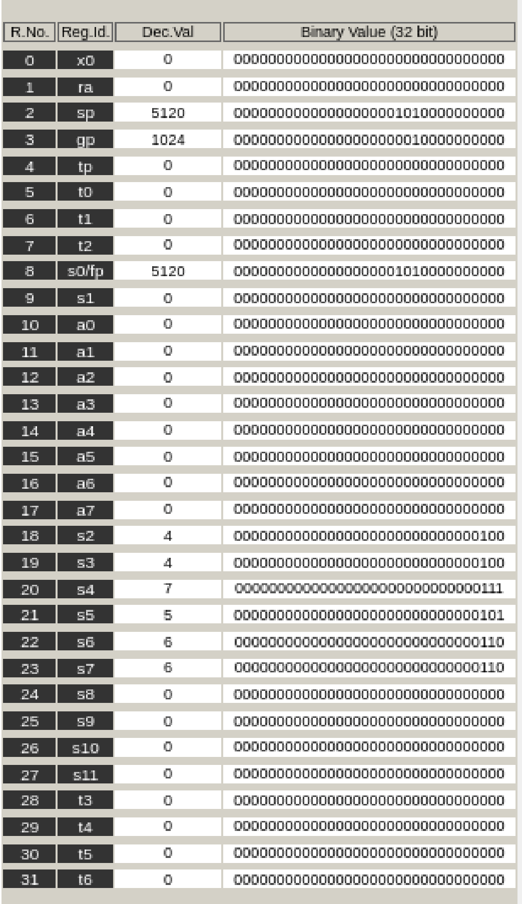
5	0	0	21	19
000000000101	00000	000	10101	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 16 (0x10)
I-type Instruction:
addi s6, x0, 6
00000000011000000000101100010011

6	0	0	22	19
000000000110	00000	000	10110	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 20 (0x14)
R-type Instruction:
add s7, s6, s1
00000000100110110000101110110011

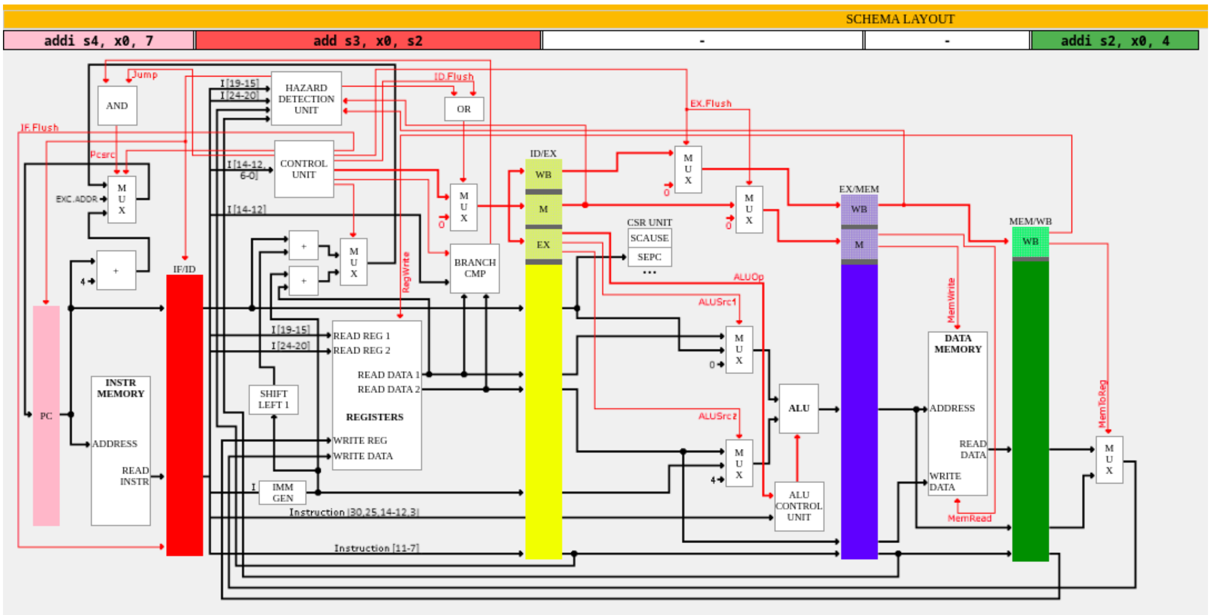
0	9	22	0	23	51
0000000	01001	10110	000	10111	0110011
FUNCT7	RS2	RS1	FUNCT3	RD	OP



R.No.	Reg.Id.	Dec.Val	Binary Value (32 bit)
0	x0	0	00000000000000000000000000000000
1	ra	0	00000000000000000000000000000000
2	sp	5120	00000000000000000001010000000000
3	gp	1024	00000000000000000000010000000000
4	tp	0	00000000000000000000000000000000
5	t0	0	00000000000000000000000000000000
6	t1	0	00000000000000000000000000000000
7	t2	0	00000000000000000000000000000000
8	s0/fp	5120	00000000000000000001010000000000
9	s1	0	00000000000000000000000000000000
10	a0	0	00000000000000000000000000000000
11	a1	0	00000000000000000000000000000000
12	a2	0	00000000000000000000000000000000
13	a3	0	00000000000000000000000000000000
14	a4	0	00000000000000000000000000000000
15	a5	0	00000000000000000000000000000000
16	a6	0	00000000000000000000000000000000
17	a7	0	00000000000000000000000000000000
18	s2	4	00000000000000000000000000000100
19	s3	4	00000000000000000000000000000100
20	s4	7	00000000000000000000000000000111
21	s5	5	00000000000000000000000000000101
22	s6	6	00000000000000000000000000000110
23	s7	6	00000000000000000000000000000110
24	s8	0	00000000000000000000000000000000
25	s9	0	00000000000000000000000000000000
26	s10	0	00000000000000000000000000000000
27	s11	0	00000000000000000000000000000000
28	t3	0	00000000000000000000000000000000
29	t4	0	00000000000000000000000000000000
30	t5	0	00000000000000000000000000000000
31	t6	0	00000000000000000000000000000000

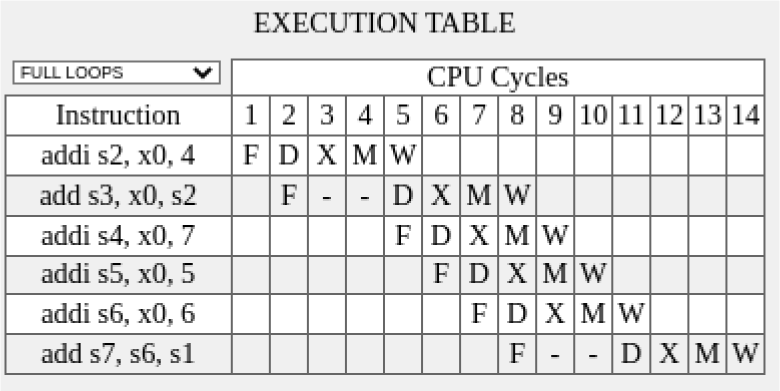
Ao final da execução, os registradores s2, s3, s4, s5, s6 e s7 sofreram alterações.

1.2.b



Como o Forwarding está desativado, só chegamos na etapa de 3 instruções acontecendo simultaneamente bem depois do que no problema 1, já que acontecem alguns conflitos e acontece o “stall” para lidá-los.

1.2.c



EXECUTION TABLE

FULL LOOPS

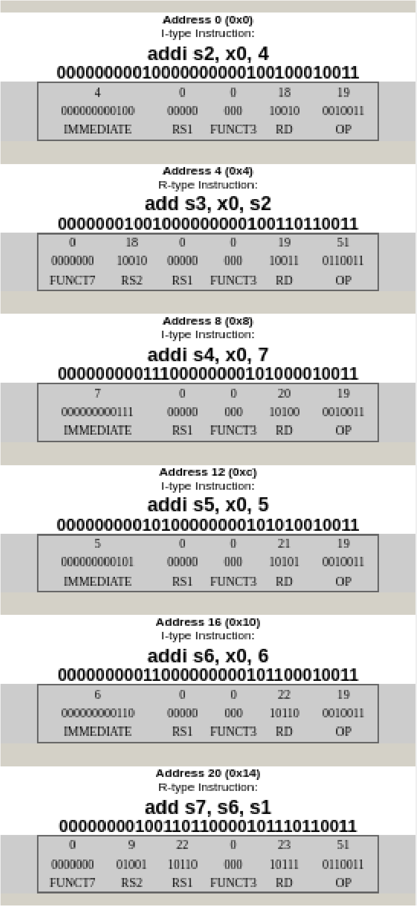
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14
addi s2, x0, 4	F	D	X	M	W									
add s3, x0, s2		F	-	-	D	X	M	W						
addi s4, x0, 7					F	D	X	M	W					
addi s5, x0, 5						F	D	X	M	W				
addi s6, x0, 6							F	D	X	M	W			
add s7, s6, s1								F	-	-	D	X	M	W

A tabela de execução mostra que houve conflito na execução da segunda e da sexta instrução. Como o Forwarding estava desativado, a execução dessas instruções foi pausada até que os conflitos fossem resolvidos.

1.2.d Ao total. foram necessários 14 ciclos de CPU.

2.2.a

Conteúdo da memória de instruções e dos registradores no início da execução:



Address 0 (0x0)
I-type Instruction:
addi s2, x0, 4
00000000010000000000100100010011

4	0	0	18	19
000000000100	00000	000	10010	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 4 (0x4)
R-type Instruction:
add s3, x0, s2
00000001001000000000100110110011

0	18	0	0	19	51
0000000	10010	00000	000	10011	0110011
FUNCT7	RS2	RS1	FUNCT3	RD	OP

Address 8 (0x8)
I-type Instruction:
addi s4, x0, 7
00000000011100000000101000010011

7	0	0	20	19
000000000111	00000	000	10100	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 12 (0xc)
I-type Instruction:
addi s5, x0, 5
00000000010100000000101010010011

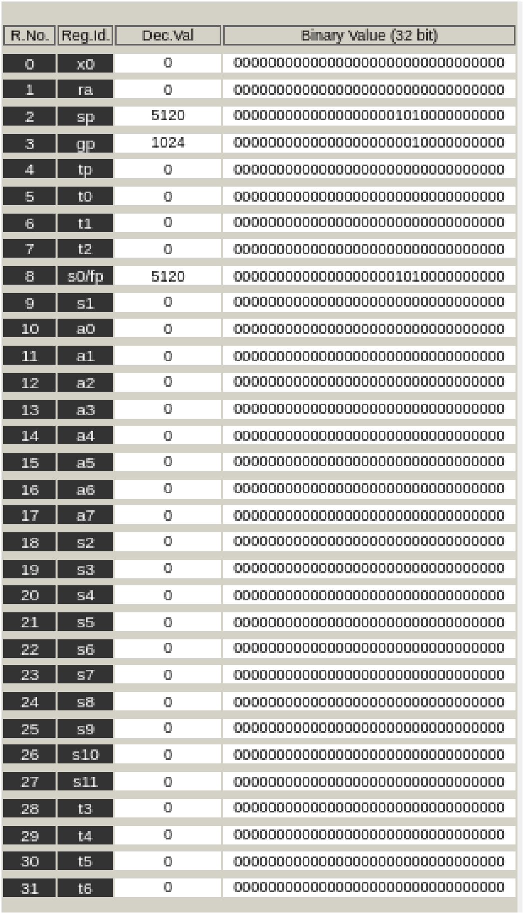
5	0	0	21	19
000000000101	00000	000	10101	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 16 (0x10)
I-type Instruction:
addi s6, x0, 6
00000000011000000000101100010011

6	0	0	22	19
000000000110	00000	000	10110	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 20 (0x14)
R-type Instruction:
add s7, s6, s1
00000000100110110000101110110011

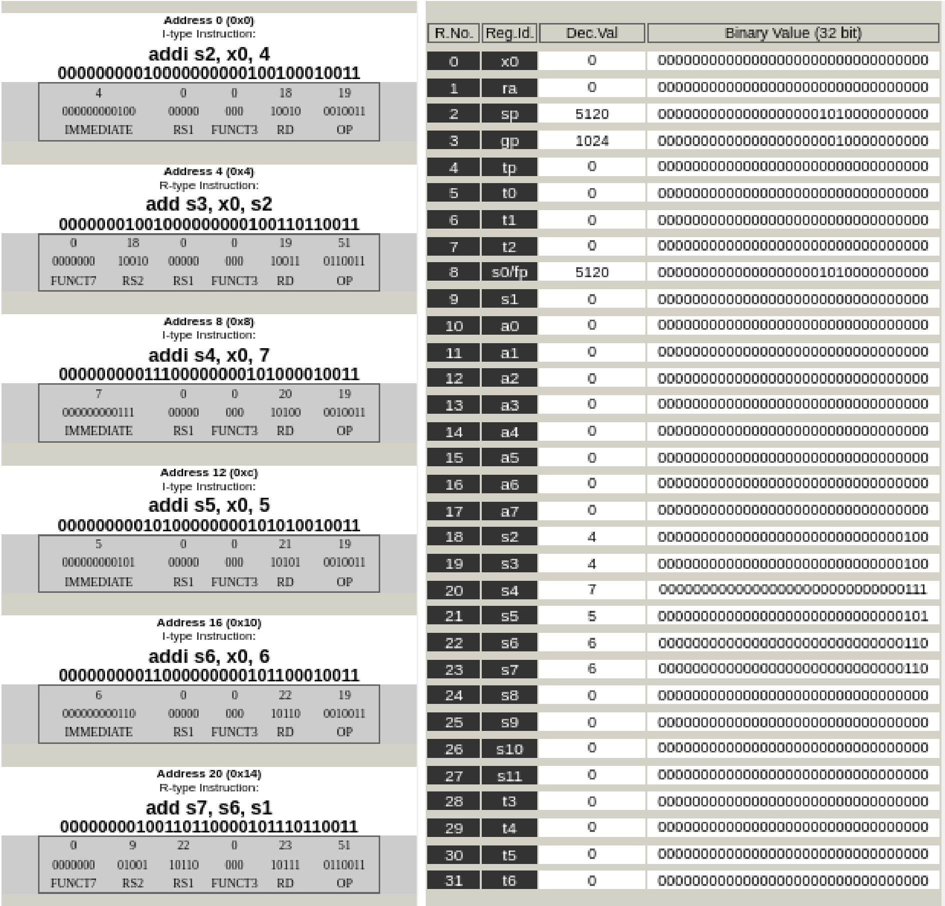
0	9	22	0	23	51
0000000	01001	10110	000	10111	0110011
FUNCT7	RS2	RS1	FUNCT3	RD	OP



R.No.	Reg.Id.	Dec.Val	Binary Value (32 bit)
0	x0	0	00000000000000000000000000000000
1	ra	0	00000000000000000000000000000000
2	sp	5120	00000000000000000001010000000000
3	gp	1024	00000000000000000000010000000000
4	tp	0	00000000000000000000000000000000
5	t0	0	00000000000000000000000000000000
6	t1	0	00000000000000000000000000000000
7	t2	0	00000000000000000000000000000000
8	s0/fp	5120	00000000000000000001010000000000
9	s1	0	00000000000000000000000000000000
10	a0	0	00000000000000000000000000000000
11	a1	0	00000000000000000000000000000000
12	a2	0	00000000000000000000000000000000
13	a3	0	00000000000000000000000000000000
14	a4	0	00000000000000000000000000000000
15	a5	0	00000000000000000000000000000000
16	a6	0	00000000000000000000000000000000
17	a7	0	00000000000000000000000000000000
18	s2	0	00000000000000000000000000000000
19	s3	0	00000000000000000000000000000000
20	s4	0	00000000000000000000000000000000
21	s5	0	00000000000000000000000000000000
22	s6	0	00000000000000000000000000000000
23	s7	0	00000000000000000000000000000000
24	s8	0	00000000000000000000000000000000
25	s9	0	00000000000000000000000000000000
26	s10	0	00000000000000000000000000000000
27	s11	0	00000000000000000000000000000000
28	t3	0	00000000000000000000000000000000
29	t4	0	00000000000000000000000000000000
30	t5	0	00000000000000000000000000000000
31	t6	0	00000000000000000000000000000000

Encontramos os mesmo valores quando o Forwarding estava desativado.

Conteúdo da memória de instruções e dos registradores ao final da execução:



Address 0 (0x0)
I-type Instruction:
addi s2, x0, 4
00000000010000000000100100010011

4	0	0	18	19
000000000100	00000	000	10010	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 4 (0x4)
R-type Instruction:
add s3, x0, s2
00000001001000000000100110110011

0	18	0	0	19	51
0000000	10010	00000	000	10011	0110011
FUNCT7	RS2	RS1	FUNCT3	RD	OP

Address 8 (0x8)
I-type Instruction:
addi s4, x0, 7
00000000011100000000101000010011

7	0	0	20	19
000000000111	00000	000	10100	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 12 (0xc)
I-type Instruction:
addi s5, x0, 5
00000000010100000000101010010011

5	0	0	21	19
000000000101	00000	000	10101	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

Address 16 (0x10)
I-type Instruction:
addi s6, x0, 6
00000000011000000000101100010011

6	0	0	22	19
000000000110	00000	000	10110	0010011
IMMEDIATE	RS1	FUNCT3	RD	OP

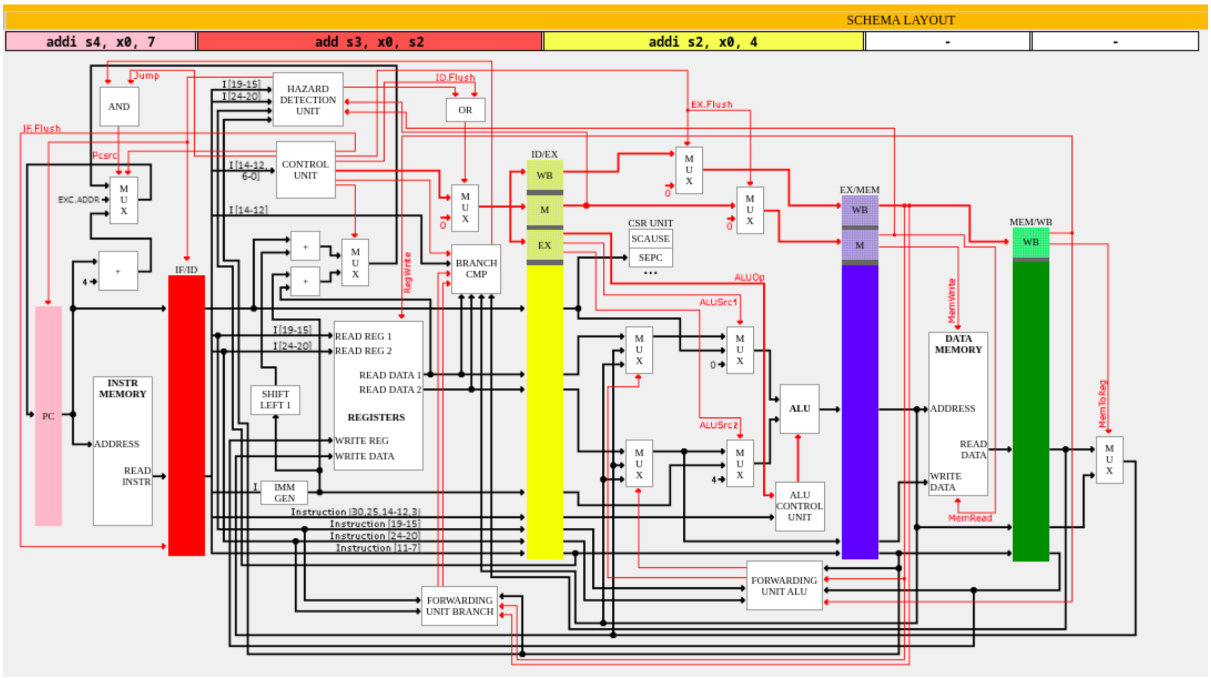
Address 20 (0x14)
R-type Instruction:
add s7, s6, s1
00000000100110110000101110110011

0	9	22	0	23	51
0000000	01001	10110	000	10111	0110011
FUNCT7	RS2	RS1	FUNCT3	RD	OP

R.No.	Reg.Id.	Dec.Val	Binary Value (32 bit)
0	x0	0	00000000000000000000000000000000
1	ra	0	00000000000000000000000000000000
2	sp	5120	00000000000000000001010000000000
3	gp	1024	00000000000000000000010000000000
4	tp	0	00000000000000000000000000000000
5	t0	0	00000000000000000000000000000000
6	t1	0	00000000000000000000000000000000
7	t2	0	00000000000000000000000000000000
8	s0/fp	5120	00000000000000000001010000000000
9	s1	0	00000000000000000000000000000000
10	a0	0	00000000000000000000000000000000
11	a1	0	00000000000000000000000000000000
12	a2	0	00000000000000000000000000000000
13	a3	0	00000000000000000000000000000000
14	a4	0	00000000000000000000000000000000
15	a5	0	00000000000000000000000000000000
16	a6	0	00000000000000000000000000000000
17	a7	0	00000000000000000000000000000000
18	s2	4	00000000000000000000000000000100
19	s3	4	00000000000000000000000000000100
20	s4	7	00000000000000000000000000000111
21	s5	5	00000000000000000000000000000101
22	s6	6	00000000000000000000000000000110
23	s7	6	00000000000000000000000000000110
24	s8	0	00000000000000000000000000000000
25	s9	0	00000000000000000000000000000000
26	s10	0	00000000000000000000000000000000
27	s11	0	00000000000000000000000000000000
28	t3	0	00000000000000000000000000000000
29	t4	0	00000000000000000000000000000000
30	t5	0	00000000000000000000000000000000
31	t6	0	00000000000000000000000000000000

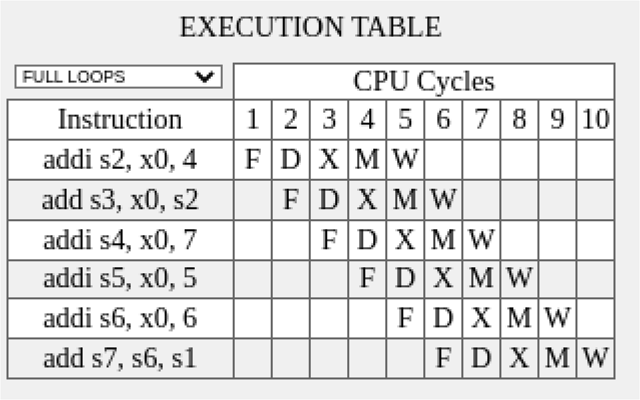
Novamente, os mesmos valores encontrados quando o Forwarding estava desativado.

2.2.b



Com o Forwarding ativo, conseguimos chegar bem antes no estado em que 3 instruções estão sendo executadas em estágios diferentes.

2.2.c



EXECUTION TABLE

FULL LOOPS

	CPU Cycles									
Instruction	1	2	3	4	5	6	7	8	9	10
addi s2, x0, 4	F	D	X	M	W					
add s3, x0, s2		F	D	X	M	W				
addi s4, x0, 7			F	D	X	M	W			
addi s5, x0, 5				F	D	X	M	W		
addi s6, x0, 6					F	D	X	M	W	
add s7, s6, s1						F	D	X	M	W

É possível ver que com o Forwarding ativo não há mais a pausa na execução de um pipeline quando temos um stall. Isso reduziu a quantidade de ciclos de CPU de precisamos para executar as instruções.

2.2.d Ao total, foram necessários 10 ciclos de CPU.