ROTEIRO 08:

.

Código:

addi s2, zero, 4 // Armazena o valor 4 em s2

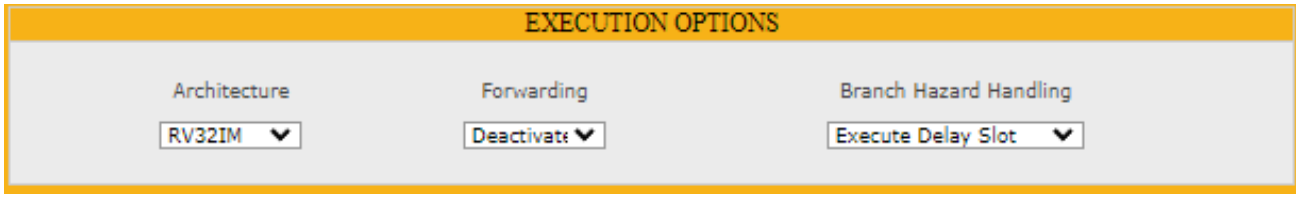
addi s3, zero, 3 // Armazena o valor 3 em s3

addi s4, zero, 7 // Armazena o valor 7 em s4

addi s5, zero, 5 // Armazena o valor 5 em s5

addi s6, zero, 6 // Armazena o valor 6 em s6

add s7, s2, s3 // Armazena a soma de s2 (4) e s3 (3) em s7 (7)



18	s2	4	00000000000000000000000000000100
19	s3	3	00000000000000000000000000000011
20	s4	7	00000000000000000000000000000111
21	s5	5	00000000000000000000000000000101
22	s6	6	00000000000000000000000000000110
23	s7	7	00000000000000000000000000000111

a) Conteúdo da Memória de Instruções (“Instruction Memory”) e dos Registradores (“Registers”), no início e no final da execução do programa.

Inicio do programa:

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

Fim do programa:

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

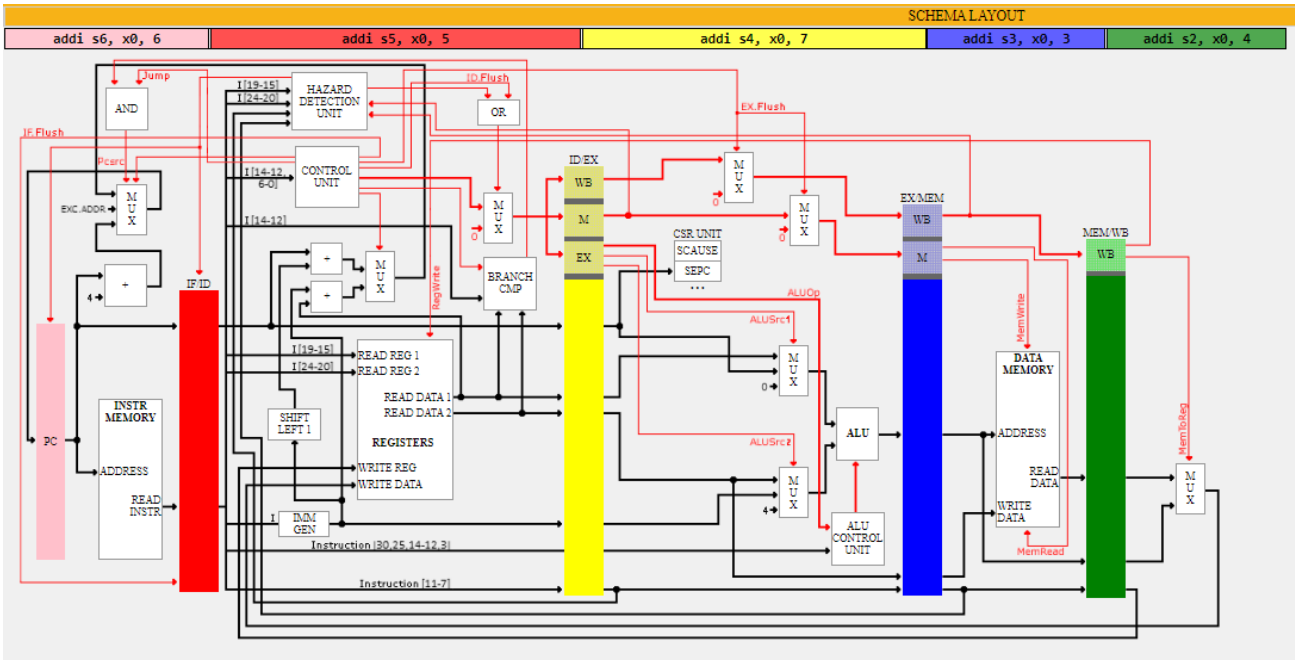
00000001001110010000101110110011

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

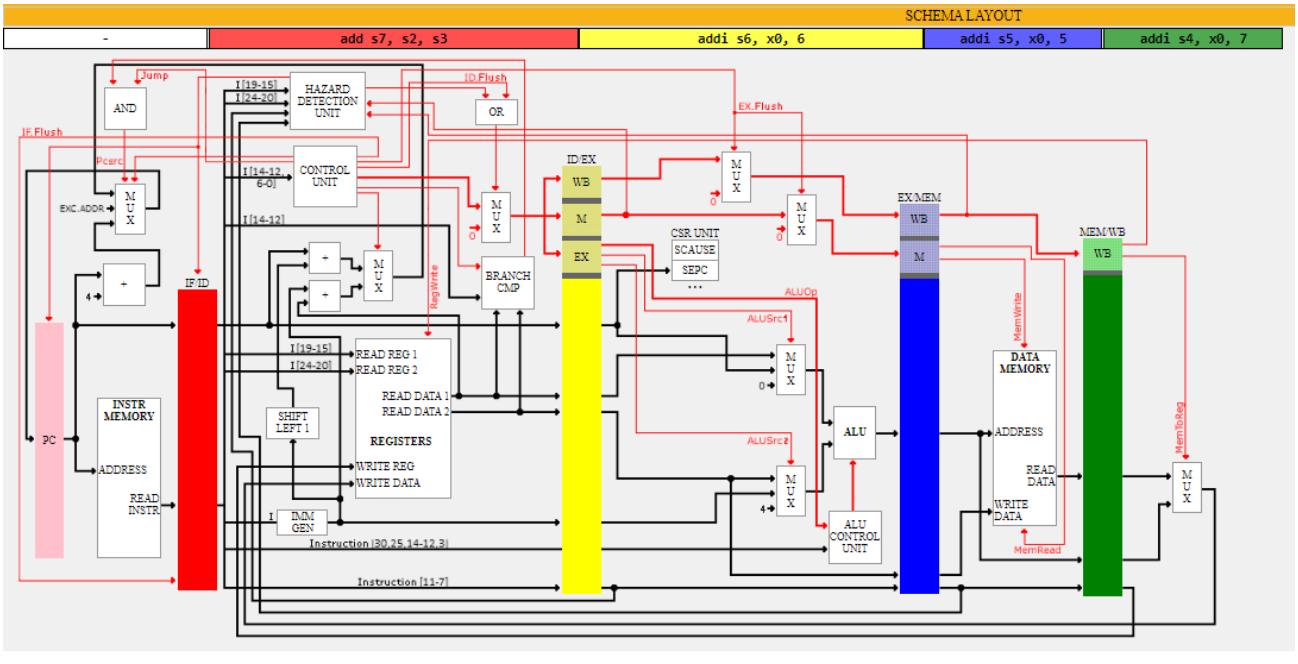
R.No.	Reg.Id.	Dec.Val	Binary Value (32 bit)
0	x0	0	00000000000000000000000000000000
1	ra	0	00000000000000000000000000000000
2	sp	5120	00000000000000000001010000000000
3	gp	1024	00000000000000000000010000000000
4	tp	0	00000000000000000000000000000000
5	t0	0	00000000000000000000000000000000
6	t1	0	00000000000000000000000000000000
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

b) Passagem em três estágios representativos do Pipeline (“SCHEMA LAYOUT”)

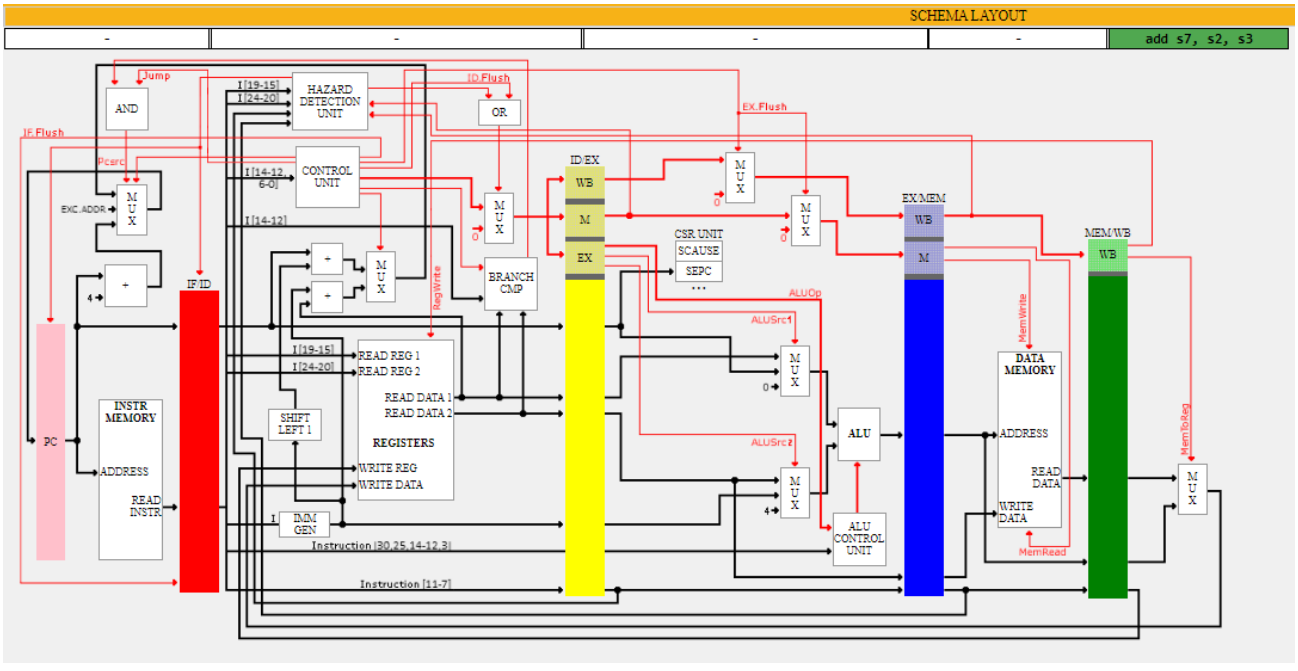
4 sendo armazenado em s2:



7 sendo armazenado em s4:



Soma de s2 e s3 sendo armazenado em s7:



c) Resultado final da execução em Pipeline, por meio da Tabela da Execução do Programa (“EXECUTION TABLE”).

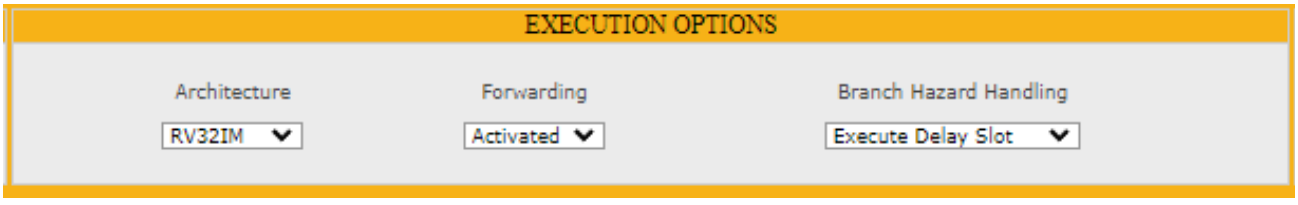
EXECUTION TABLE

FULL LOOPS

Instruction	CPU Cycles 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

d) Ciclos de CPU necessários para executar esse programa.

Como mostra na tabela anterior, não necessários 10 ciclos!



Não muda nada!

Código 2:

addi s2, zero, 4 // Armazena o valor 4 em s2

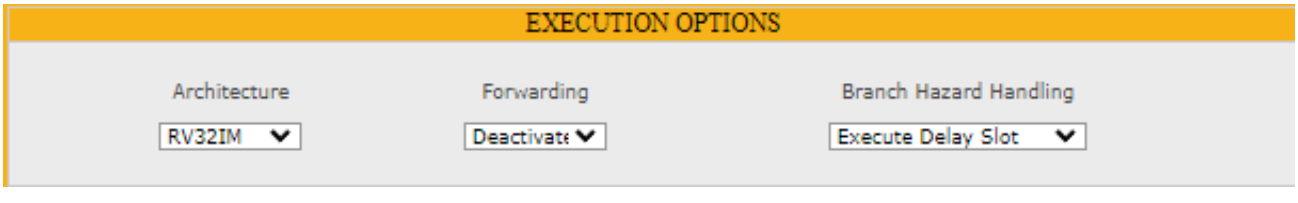
add s3, zero, s2 // Armazena o valor 4 em s3

addi s4, zero, 7 // Armazena o valor 7 em s4

addi s5, zero, 5 // Armazena o valor 5 em s5

addi s6, zero, 6 // Armazena o valor 6 em s6

add s7, s6, s1 // Armazena a soma de s6 (6) e s1 (?) em s7 (?)



a) Conteúdo da Memória de Instruções (“Instruction Memory”) e dos Registradores (“Registers”), no início e no final da execução do programa.

Inicio:

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

Fim:

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

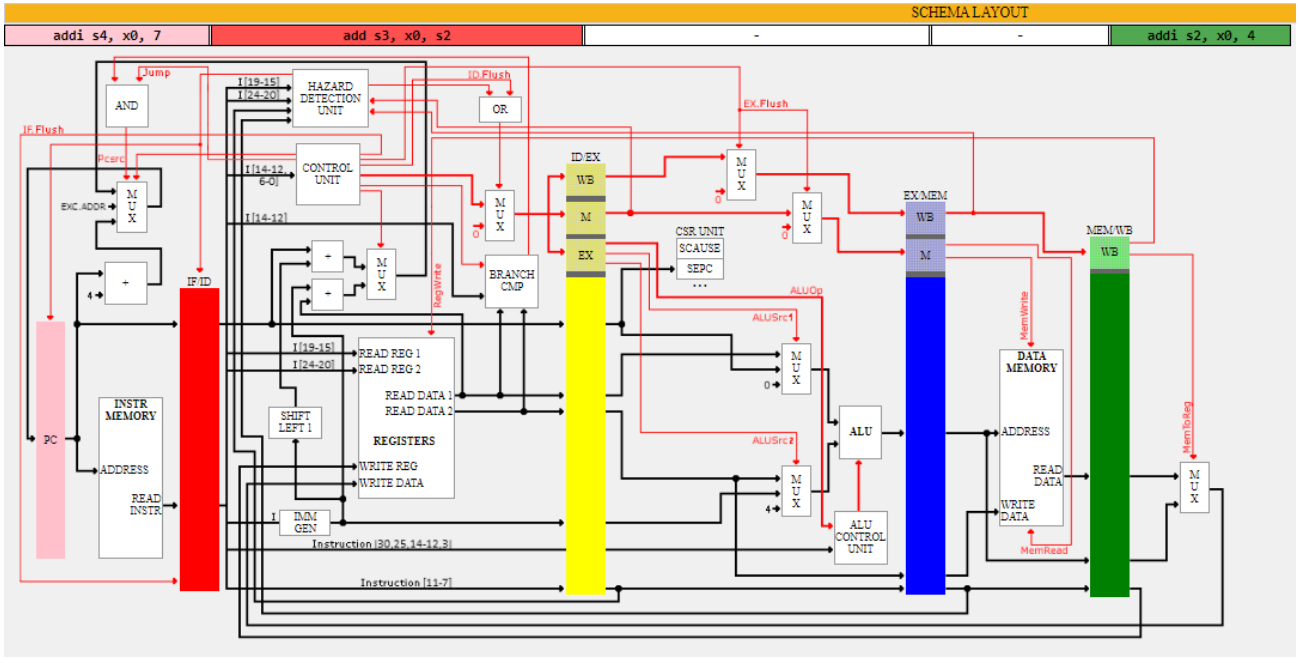
00000000100110110000101110110011

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

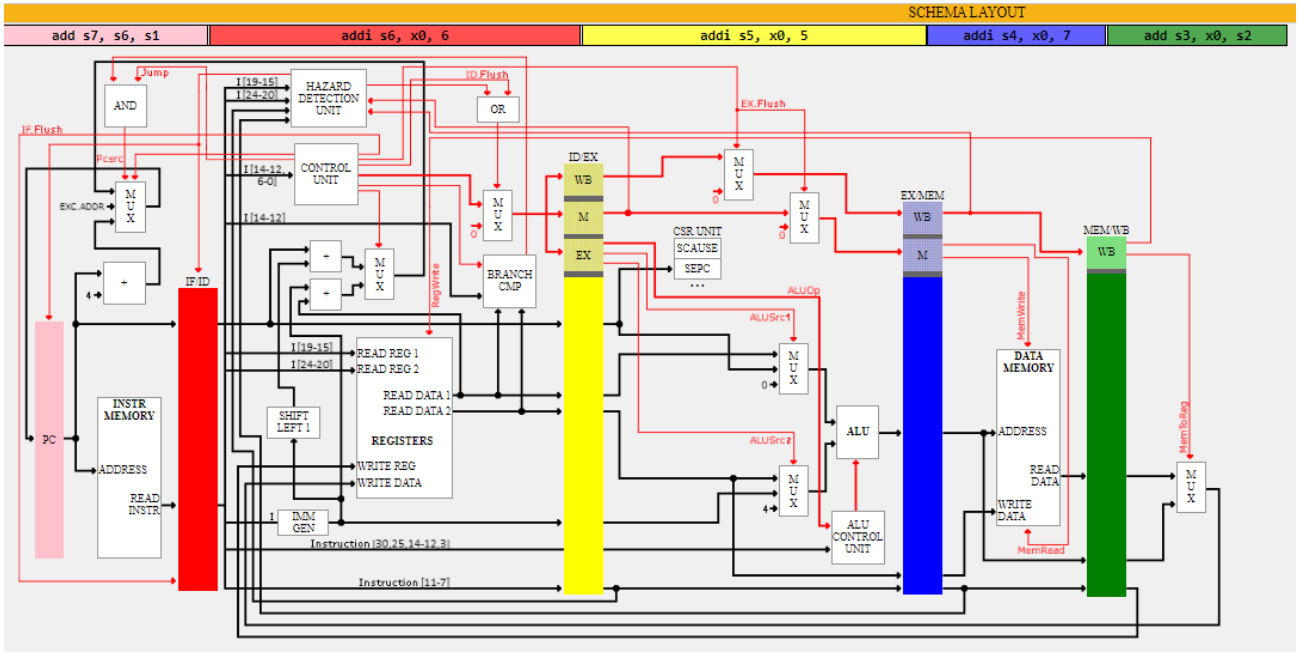
R.No.	Reg.Id.	Dec.Val	Binary Value (32 bit)
0	x0	0	00000000000000000000000000000000
1	ra	0	00000000000000000000000000000000
2	sp	5120	00000000000000000001010000000000
3	gp	1024	00000000000000000000010000000000
4	tp	0	00000000000000000000000000000000
5	t0	0	00000000000000000000000000000000
6	t1	0	00000000000000000000000000000000
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

b) Passagem em três estágios representativos do Pipeline (“SCHEMA LAYOUT”).

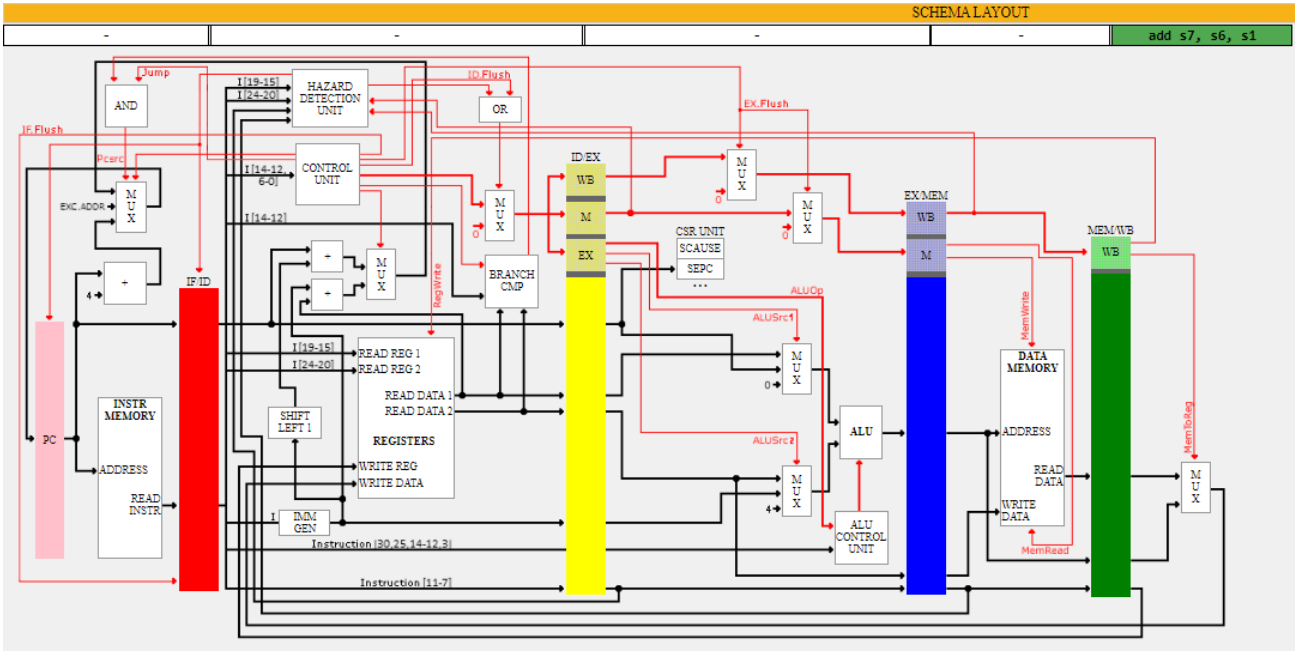
4 armazenado no s2:



4 sendo armazenado em s3, copia do s2:



s6 sendo armazenado em s7:



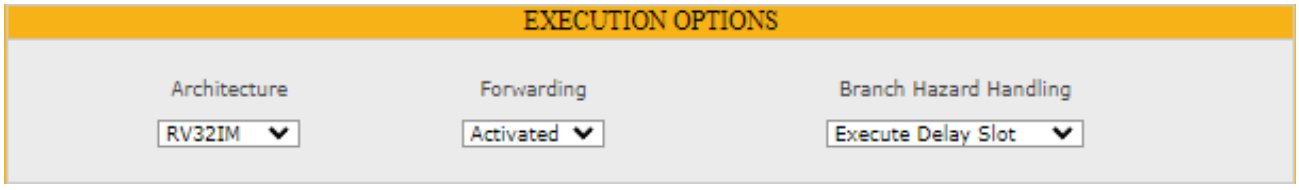
c) Resultado final da execução em Pipeline, por meio da Tabela da Execução do Programa (“EXECUTION TABLE”).

EXECUTION TABLE

FULL LOOPS	CPU Cycles													
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

d) Ciclos de CPU necessários para executar esse programa.

14 ciclos foram necessários, e contamos com 4 esperas de ciclos(2 em cada add).



Apenas a tabela de execução mudou:

EXECUTION TABLE

FULL LOOPS

Instruction	CPU Cycles									
	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

Dessa vez foram necessários 10 ciclos sem esperas.

Resultados continuam os mesmos.