

## Exercícios 06 - Respostas

1.

I1:	lw \$a1, 4(\$sp)	# \$a1 ← Mem[4 + \$sp]
I2:	sw \$a1, 0(\$v0)	# \$a1 → Mem[0 + \$v0]
I3:	addi \$sp, \$sp, -4	# \$sp ← \$sp - 4
I4:	sw \$a0, 8(\$sp)	# \$a0 → Mem[8+\$sp]
I5:	addi \$a0, \$a0, -2	# \$a0 ← \$a0 - 2

Instruções	Registrador
I1 e I2	\$a1
I1 e I3	\$sp
I3 e I4	\$sp
I4 e I5	\$a0

2.

	ciclos														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I1: add R1, R2, R3	I	D	X	M	W										
I2: sub R2, R3, R2		I	D	X	M	W									
I3: add R3, R1, R3			I	-	D	X	M	W							
I4: lw R4, 10 (R7)					I	D	X	M	W						
I5: add R5, R1, R2						I	D	X	M	W					

3.

	ciclos de relógio																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
I1: lw \$2, 100(\$5)	I	D	X	M	W												
I2: add \$1, \$2, \$3		I	-	-	D	X	M	W									
I3: sub \$3, \$2, \$1					I	-	-	D	X	M	W						
I4: sw \$2, 50(\$1)								I	D	X	M	W					
I5: add \$2, \$3, \$3									I	-	D	X	M	W			
I6: sub \$2, \$2, \$4											I	-	-	D	X	M	W

4.

A instrução add \$5, \$2, \$4 depende da instrução sub \$2, \$4, \$6 – no registrador \$2.

Solução: antecipação.

A instrução or \$7, \$2, \$8 depende da instrução sub \$2, \$4, \$6 – no registrador \$2.

Solução: antecipação.

A instrução lw \$9, 10(\$7) depende da instrução or \$7, \$2, \$8 – no registrador \$7.

Solução: antecipação.

A instrução sw \$7, 10(\$5) depende da instrução or \$7, \$2, \$8 – no registrador \$7.

Solução: antecipação – observando que esse circuito de antecipação não é o mesmo circuito de antecipação colocado no estágio EX, e sim, um circuito que deve ser colocado no estágio M.

5.

I.

	Instruction Sequence	With Forwarding	Without Forwarding
<b>a.</b>	I1: ADD R1,R2,R1 I2: LW R2,0(R1) I3: LW R1,4(R1) I4: OR R3,R1,R2	(R1) I3 to I4	(R1) I1 to I2, I3 (R2) I2 to I4 (R1) I3 to I4
<b>b.</b>	I1: LW R1,0(R1) I2: AND R1,R1,R2 I3: LW R2,0(R1) I4: LW R1,0(R3)	(R1) I1 to I2	(R1) I1 to I2 (R1) I2 to I3

II.

	Instruction Sequence	RAW
<b>a.</b>	I1: ADD R1,R2,R1 I2: LW R2,0(R1) I3: LW R1,4(R1) I4: OR R3,R1,R2	(R1) I1 to I2 (30 overrides -1)
<b>b.</b>	I1: LW R1,0(R1) I2: AND R1,R1,R2 I3: LW R2,0(R1) I4: LW R1,0(R3)	(R1) I1 to I2 (0 overrides 4)

III.

	Instruction Sequence with Forwarding Stalls	Execution without Forwarding	Values after Execution
<b>a.</b>	I1: ADD R1,R2,R1 I2: LW R2,0(R1) I3: LW R1,4(R1) Stall I4: OR R3,R1,R2	R1 = 30 (Stall and after) R2 = 0 (I4 and after) R1 = 0 (after I4)  R3 = 30 (after I4)	R0 = 0 R1 = 0 R2 = 0 R3 = 30
<b>b.</b>	I1: LW R1,0(R1) Stall I2: AND R1,R1,R2 I3: LW R2,0(R1) I4: LW R1,0(R3)	R1 = 0 (I3 and after)  R1 = 4 (after I4) R2 = 0 R1 = 0	R0 = 0 R1 = 0 R2 = 0 R3 = 3000