

Sia data la seguente sequenza di istruzioni assembler, dove i dati immediati sono espressi in esadecimale

```
SW $9, 0($1)
LW $1, 7($9)
SUB $9, $1, $8
SW $3, 73($9)
SUBI $9, $3, 9
SW $7, 78($9)
LW $9, A($7)
```

Si consideri la pipeline MIPS a 5 stadi vista a lezione, con possibilità di data-forwarding e con possibilità di scrittura e successiva lettura dei registri in uno stesso ciclo di clock:

- mostrare come evolve la pipeline durante l'esecuzione del codice, spiegando nel dettaglio i motivi di un eventuale stallo o dell'utilizzo di un particolare circuito di by-pass.

## Soluzione

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
SW \$9, 0(\$1)	IF	ID	EXE	MEM	WB											
LW \$1, 7(\$9)		IF	ID	EXE	MEM	WB	MEM/WB.LMD_lw->EXE_TopALU_sub									
SUB \$9, \$1, \$8			IF	ID	ID	EXE	MEM	WB	EXE/MEM.ALUOutput_sub->EXE_TopALU_sw							
SW \$3, 73(\$9)				IF	IF	ID	EXE	MEM	WB							
SUBI \$9, \$3, 9						IF	ID	EXE	MEM	WB	EXE/MEM.ALUOutput_subi->EXE_TopALU_sw					
SW \$7, 78(\$9)							IF	ID	EXE	MEM	WB					
LW \$9, A(\$7)								IF	ID	EXE	MEM	WB				

Sia data la seguente sequenza di istruzioni assembler, dove i dati immediati sono espressi in esadecimale

```
SW $1, A4($2)
LW $2, 90($1)
SUB $1, $2, $8
SW $3, 4($2)
ADDI $1, $3, 4
ADDI $2, $3, 8
LW $2, 15($1)
```

Si consideri la pipeline MIPS a 5 stadi vista a lezione, con possibilità di data-forwarding e con possibilità di scrittura e successiva lettura dei registri in uno stesso ciclo di clock:

- mostrare come evolve la pipeline durante l'esecuzione del codice, spiegando nel dettaglio i motivi di un eventuale stallo o dell'utilizzo di un particolare circuito di by-pass.

## Soluzione

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SW \$1, A4(\$2)	IF	ID	EXE	MEM	WB										
LW \$2, 90(\$1)		IF	ID	EXE	MEM	WB	MEM/WB.LMD_lw->EXE_TopALU_sub								
SUB \$1, \$2, \$8			IF	ID	ID	EXE	MEM	WB							
SW \$3, 4(\$2)				IF	IF	ID	EXE	MEM	WB						
ADDI \$1, \$3, 4						IF	ID	EXE	MEM	WB	MEM/WB.ALUOutput_addi->EXE_TopALU_lw				
ADDI \$2, \$3, 8							IF	ID	EXE	MEM	WB				
LW \$2, 15(\$1)								IF	ID	EXE	MEM	WB			

Sia data la seguente sequenza di istruzioni assembler, dove i dati immediati sono espressi in esadecimale

```

SUB $5, $1, $4
LW $1, 7($5)
ADD $5, $1, $8
LW $3, 73($5)
ADDI $5, $5, 3
SW $7, 78($3)
LW $5, A($7)

```

Si consideri la pipeline MIPS a 5 stadi vista a lezione, con possibilità di data-forwarding e con possibilità di scrittura e successiva lettura dei registri in uno stesso ciclo di clock:

- mostrare come evolve la pipeline durante l'esecuzione del codice, spiegando nel dettaglio i motivi di un eventuale stallo o dell'utilizzo di un particolare circuito di by-pass.

## Soluzione

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
SUB \$5, \$1, \$4	IF	ID	EXE	MEM	WB	EXE/MEM.ALUOutput_sub -> EXE.Top_ALU_input_lw											
LW \$1, 7(\$5)		IF	ID	EXE	MEM	WB	MEM/WB.LMD_lw -> EXE.Top_ALU_input_add										
ADD \$5, \$8, \$1			IF	ID	ID	EXE	MEM	WB	1) EXE/MEM.ALUOutput_add -> EXE.Top_ALU_input_lw 2) MEM/WB.ALUOutput_add -> EXE.Top_ALU_input_add								
LW \$3, 73(\$5)				IF	IF	ID	EXE	MEM	WB	MEM/WB.LMD_lw -> EXE.Top_ALU_input_sw							
ADDI \$5, \$5, 3						IF	ID	EXE	MEM	WB							
SW \$7, 78(\$3)							IF	ID	EXE	MEM	WB						
LW \$5, A(\$7)								IF	ID	EXE	MEM	WB					