

esercizio pipeline

Sia data la seguente sequenza di istruzioni assembler:

```

LB    $3, 30($0)
ADD   $2, $0, $0
LB    $1, 845($2)
ADDI  $1, $1, 4
ADDI  $2, $2, 3
SB    $1, 82($2)
SUB   $4, $3, $2

```

Si consideri la pipeline MIPS a 5 stadi vista a lezione, con possibilità di data-forwarding e di scrittura e successiva lettura dei registri in uno stesso ciclo di clock. Si mostri il diagramma degli stadi della pipeline per l'esecuzione del codice.

Soluzione

istruzione	C I C L I C L O C K													commenti
	1	2	3	4	5	6	7	8	9	10	11	12	13	
LB \$3, 30(\$0)	IF	ID	EX	ME	WB									
ADD \$2, \$0, \$0		IF	ID	EX	ME	WB								fw out-ALU -> in-ALU
LB \$1, 845(\$2)			IF	ID	EX	ME	WB							fw MEM/WB.LMD -> in-ALU
ADDI \$1, \$1, 4				IF	ID	ID	EX	ME	WB					stallo
ADDI \$2, \$2, 3					IF	IF	ID	EX	ME	WB				fw MEM/WB.ALUOut -> in-ALU
SB \$1, 82(\$2)							IF	ID	ID	EX	ME	WB		stallo
SUB \$4, \$3, \$2								IF	IF	ID	EX	ME	WB	

esercizio pipeline

Sia data la seguente sequenza di istruzioni assembler:

```

LB    $3, 450($0)
ADD   $2, $0, $0
LB    $1, 558($2)
ADDI  $2, $2, 5
SUB   $4, $3, $2
ADDI  $1, $1, 7
SB    $1, 58($2)

```

Si consideri la pipeline MIPS a 5 stadi vista a lezione, senza possibilità di data-forwarding, ma con possibilità di scrittura e successiva lettura dei registri in uno stesso ciclo di clock. Si mostri il diagramma degli stadi della pipeline per l'esecuzione del codice.

Soluzione

istruzione	C I C L I C L O C K																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
LB \$3, 450(\$0)	IF	ID	EX	ME	WB												
ADD \$2, \$0, \$0		IF	ID	EX	ME	WB											
LB \$1, 558(\$2)			IF	ID	ID	ID	EX	ME	WB								
ADDI \$2, \$2, 5				IF	IF	IF	ID	EX	ME	WB							
SUB \$4, \$3, \$2							IF	ID	ID	ID	EX	ME	WB				
ADDI \$1, \$1, 7								IF	IF	IF	ID	EX	ME	WB			
SB \$1, 58(\$2)											IF	ID	ID	ID	EX	ME	WB

esercizio pipeline

Sia data la seguente sequenza di istruzioni assembler:

```

LB    $3, 450($0)
ADD   $2, $0, $0
LB    $1, 558($2)
ADDI  $2, $2, 5
SUB   $4, $3, $2
ADDI  $1, $1, 7
SB    $1, 58($2)

```

Si consideri la pipeline MIPS a 5 stadi vista a lezione, con possibilità di data-forwarding e di scrittura e successiva lettura dei registri in uno stesso ciclo di clock. Si mostri il diagramma degli stadi della pipeline per l'esecuzione del codice.

Soluzione

istruzione	C I C L I C L O C K													commenti
	1	2	3	4	5	6	7	8	9	10	11	12	13	
LB \$3, 450(\$0)	IF	ID	EX	ME	WB									
ADD \$2, \$0, \$0		IF	ID	EX	ME	WB								fw out-ALU -> in-ALU
LB \$1, 558(\$2)			IF	ID	EX	ME	WB							fw MEM/WB.ALUOut -> in-ALU
ADDI \$2, \$2, 5				IF	ID	EX	ME	WB						fw out-ALU -> in-ALU
SUB \$4, \$3, \$2					IF	ID	EX	ME	WB					
ADDI \$1, \$1, 7						IF	ID	EX	ME	WB				
SB \$1, 58(\$2)							IF	ID	ID	ID	EX	ME	WB	stallo

esercizio pipeline con data-forwarding

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

SUB \$2, \$7, \$5
 LW \$1, 7(\$2)
 ADD \$2, \$1, \$8
 SW \$3, 73(\$1)
 SUBI \$2, \$3, 4
 ADDI \$7, \$3, 8
 ADD \$1, \$7, \$2

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 \$2, \$7, \$5	IF	ID	EX	MEM	WB											EX.ALUOutput_sub -> EX.Top_ALU_input_lw
LW \$1, 7(\$2)		IF	ID	EX	MEM	WB										MEM.LMD_lw -> EX.Top_ALU_input_add
ADD \$2, \$1, \$8			IF	ID	ID	EX	MEM	WB								
SW \$3, 73(\$1)				IF	IF	ID	EX	MEM	WB							
SUBI \$2, \$3, 4						IF	ID	EX	MEM	WB						MEM.ALUOutput_subi -> EX.Bottom_ALU_input_add
ADDI \$7, \$3, 8							IF	ID	EX	MEM	WB					EX.ALUOutput_addi -> EX.Top_ALU_input_add
ADD \$1, \$7, \$2								IF	ID	EX	MEM	WB				

Esercizio Pipeline 1: Dipendenze

Si consideri il seguente frammento di codice:

LOOP: LW \$1, 0(\$2)	$! R1 \leftarrow mem[0 + [R2]]$
ADDI \$1, \$1, 1	$! R1 \leftarrow [R1] + 1$
SW \$1, 0(\$2)	$! mem[0 + [R2]] \leftarrow [R1]$
ADD \$2, \$1, \$2	$! R2 \leftarrow [R1] + [R2]$
SUB \$4, \$3, \$2	$! R4 \leftarrow [R3] - [R2]$
BENZ \$4, LOOP	$! if([R4] \neq 0) PC \leftarrow indirizzo(loop)$

si individuino le dipendenze **ReadAfterWrite** (RAW) e **WriteAfterWrite** (WAW).

Soluzione

#	codice	R1	R2	R3	R4	commento
1	LOOP: LW \$1, 0(\$2)	W	R			legge R2, scrive R1
2	ADDI \$1,\$1, 1	RW				legge e scrive R1
3	SW \$1, 0(\$2)	R	R			legge R1 e R2
4	ADD \$2, \$1, \$2	R	RW			legge R1, legge e scrive R2
5	SUB \$4, \$3, \$2		R	R	W	legge R2 3 R3, scrive R4
6	BENZ \$4, LOOP				R	legge R4

Linee codice	Spiegazione dipendenza	Tipo
2←1	ADDI legge R1 che è scritto da LW	RAW
2←1	ADDI scrive R1 che è scritto da LW	WAW
3←2, 3←1	SW legge R1 che è scritto da ADDI, e prima da LW	RAW
4←2, 4←1	ADD legge R1 che è scritto da ADDI, e prima da LW	RAW
5←4	SUB legge R2 che è scritto da ADD	RAW
6←5	BENZ legge R4 che è scritto da SUB	RAW

Corso di Architettura degli Elaboratori

Anno Accademico 2016/2017

Esercizi pipeline

es-p1

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 (da compilare)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
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)																				

Commenti alla soluzione:

es-p2

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 (da compilare)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
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)																				

Commenti alla soluzione:

es-p3

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 (da compilare)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
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)																				

Commenti alla soluzione: