Come svolgere un esercizio sulla pipeline MIPS

Riccardo Montagnin

1. **Traduci il linguaggio Assembler.**
   Traduci le istruzioni scritte nel linguaggio Assembler in piccoli schemi, per capire ogni istruzione che cosa fa.

   Es.
   $LW\ \$1,\ 250(\$5) \quad \Rightarrow \quad R1 \leftarrow mem(250 + R5)$
   $ADDI\ \$8, \$1, 1 \quad \Rightarrow \quad R8 \leftarrow R1 + 1$
   $SUB\ \$2, \$8, \$15 \quad \Rightarrow \quad R2 \leftarrow R8 - R15$

2. **Individua le dipendenze.**
   Individua le possibili dipendenze tra un'istruzione e l'altra, ricordando che possono essere di tre tipi:

   (a) Read After Write (RAW)
   Quando in una istruzione si legge un registro che in una precedente istruzione viene scritto, prima che quest'ultima abbia finito di scrivere.

   (b) Write After Write (WAW)
   Quando in una istruzione si deve scrivere in un registro che è stato scritto anche in una istruzione precedente, senza che questa abbia finito.

   (c) Write After Read (WAR)
   Quando in una istruzione si deve scrivere in un registro che è stato letto in una istruzione precedente senza che questa abbia finito (caso raro nella pipeline).

   Dopo aver individuato le dipendenze, tieni conto **solamente** di quelle **RAW** in quanto le WAW si risolvono da sole, e le WAR sono molto rare (negli esercizi non ci sono mai).

3. **Trova come risolvere le dipendenze.**
   Ci sono due modi per risolvere le dipendeze:

   (a) Attraverso il circuito di forward dei dati EX./EX.
   Questo circuito si può utilizzare solo quando vi è una dipendeza RAW e il registro che va in conflitto è coinvolto in due operazioni aritmetiche successive.

   Es. 1
   $ADDI\ \$8, \$1, 1 \quad \Rightarrow \quad R8 \leftarrow R1 + 1$
   $SUB\ \$2, \$8, \$15 \quad \Rightarrow \quad R2 \leftarrow R8 - R15$
   Il registro $R8$ viene scritto nell'istruzione $ADDI$, e letto nell'istruzione SUB (che coinvolge il registro in una operazione aritmetica).

   Es. 2
   $ADDI\ \$1, \$1, 16 \quad \Rightarrow \quad R1 \leftarrow R1 + 16$
   $SW\ \$4, 20(\$1) \quad \Rightarrow \quad mem(R1 + 20) \leftarrow R4$
   Il registro $R1$ viene scritto nell'istruzione $ADDI$, e letto nell'istruzione SW (che coinvolge il registro in una operazione aritmetica).

(b) Attraverso il circuito di forward dei dati ME./EX.
Questo circuito si può utilizzare quando un'istruzione legge un registro che nell'istruzione precedente è il risultato di un'operazione che coinvolge la memoria (LW) (Es. 1) **oppure** quando un'istruzione scrive un registro a seguito di un'operazione aritmetica e lo stesso registro è letto **due operazioni dopo** in un'altra operazione aritmetica (Es. 2).

Es. 1
$LW\ \$1, 10(\$0) \quad \Rightarrow \quad R1 \leftarrow mem(R0 + 16)$
$ADDI\ \$8, \$1, 1 \quad \Rightarrow \quad R8 \leftarrow R1 + 1$
Il registro $R1$ viene scritto nell'operazione di LW (che coinvolge la memoria) e viene letto nell'istruzione ADDI.

Es. 2
$ADD\ \$8, \$1, 1 \quad \Rightarrow \quad R8 \leftarrow R1 + 1$
$LW\ \$3, 0(\$1) \quad \Rightarrow \quad R3 \leftarrow mem(R1 + 0)$
$SUB\ \$3, \$8, \$1 \quad \Rightarrow \quad R3 \leftarrow R8 - R1$
Il registro $R8$ viene scritto nell'istruzione ADD all'interno di un'operazione aritmetica e viene letto due istruzioni dopo nell'istruzione SUB.
In questo caso si sarebbe dovuto utilizzare il circuito di forward dei dati EX./EX. dall'istruzione ADD a quella SUB, ma ciò non è possibile, e si utilizza pertanto quello di ME./EX. che è anche più breve.

4. **Completa la tabella.**
Completa la tabella dell'esercizio, inserendo i vari stadi di ogni istruzione, ricordandoti che:

(a) La sequenza degli stadi è: IF ID EX ME WB
(b) Sotto ogni IF non ci va **nulla** .
(c) Sotto ogni ID ci va un **IF**.
(d) Sotto ogni EX ci va un **ID**.
(e) Sotto ogni ME ci va un **EX** o un **ID**.
(f) Sotto ogni WB ci va un **ME** o un **EX**.
(g) Se usi un circuito di forward le connessioni vanno fatte in **diagonale**.
(h) Se usi un circuito di forward le connessioni vanno fatte sempre **più brevi possibili**.

Es. Vedi esercizio nella prossima pagina *(Ringraziamento: Prof. Sperduti)*.

5. **Scrivi i commenti.**
Quando usi un circuiti di forward scrivi per ogni riga di **destinazione** del circuito il commento opportuno.

Es. Vedi esercizio nella prossima pagina *(Ringraziamento: Prof. Sperduti)*.

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				