

# Esercizio Pipeline 1: Dipendenze

Si consideri il seguente frammento di codice:

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

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