

Esercizio: Dipendenze

- Che tipo di criticità si possono prevedere per il funzionamento della pipeline che corrisponderà a questo codice sorgente?

```
if (a > c) {  
    d = d + 5;  
    a = b + d + e;  
}  
else {  
    e = e + 2;  
    f = f + 2;  
    c = c + f;  
}  
b = a + f;
```

Esercizio Pipeline : Dipendenze

Si consideri il seguente frammento di codice:

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

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

Esercizio pipeline

Si consideri una pipeline a 4 stadi (IF, ID, EI, WO) per cui:

- i salti incondizionati sono risolti (identificazione salto e calcolo indirizzo target) alla fine del secondo stadio (ID)
 - i salti condizionati sono risolti (identificazione salto, calcolo indirizzo target e calcolo condizione) alla fine del terzo stadio (EI)
 - il primo stadio (IF) è indipendente dagli altri
 - ogni stadio impiega 1 ciclo di clock
- Si considerino le seguenti statistiche:
 - 15% delle istruzioni sono di salto condizionale
 - 1% delle istruzioni sono di salto incondizionale
 - Il 60% delle istruzioni di salto condizionale hanno la condizione soddisfatta (prese)

valutare i ritardi nella pipeline introdotti dai salti