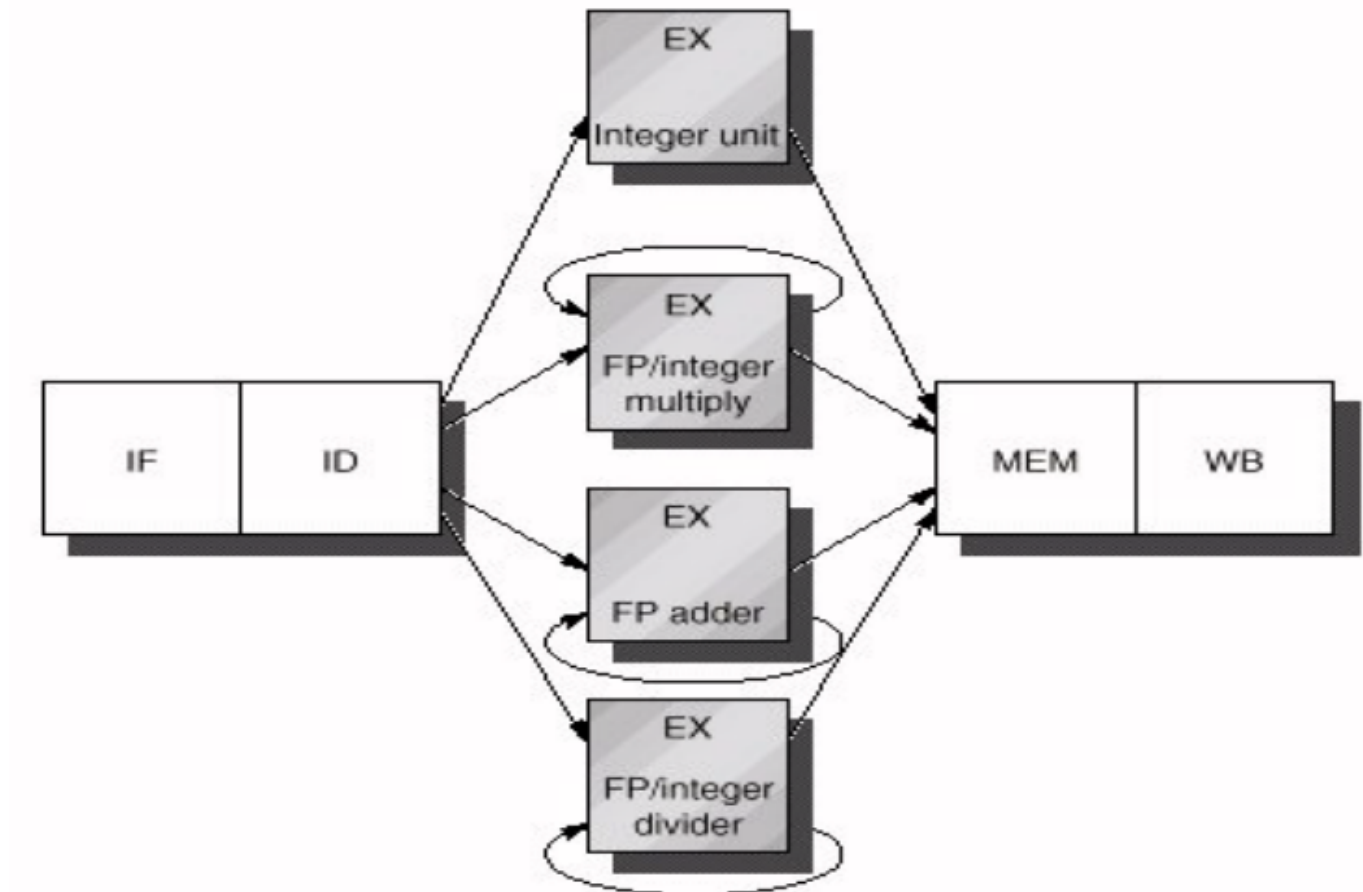


Nel mondo ideale, lo stadio **EX** ci mette un solo clock.  
Tuttavia alcune operazioni richiedono più cicli di clock.

## Floating Operations

Le operazioni con i floating point sono più complesse rispetto a quelle degli interi.



*Nel caso degli interi, c'è solo Integer unit.*

*Nei FP, ci sono diverse unità di esecuzione che ci mettono più cicli di clock*

## Latenza

Numero di clock in più che ci mette oltre allo stadio **EX**.

Ad esempio la latenza dell'**ALU Integer** è 0, mentre le operazioni di memoria hanno 1

## Initiation Interval

Numero di clock dopo il quale è possibile far partire un'istruzione dello stesso tipo

## Hazard 2

A causa delle differenze introdotte nello stadio **EX**, possono capitare più hazard.

## Write after Write (WAW)

Capita quando due istruzioni vogliono scrivere nello stesso momento nella stessa locazione

ADD.D F2,F4,F6	IF	ID	A1	A2	A3	A4	MEM	WB
...		IF	ID	EX	MEM	WB		
...			IF	ID	EX	MEM	WB	
L.D F2,0(R2)				IF	ID	EX	MEM	WB

L.D potrebbe scrivere in F2 prima della ADD.D.

In questo caso, si potrebbe stallare la load.

## Read after Write(RAW)

Capita quando **istruzione 2** legge un registro prima che **istruzione 1** abbia finito di scriverci sopra.

Anche qua, per risolvere, è possibile introdurre uno stallo.

## MIPS R4000 Pipeline

Il MIPS R4000, processore a 64 bit del 1991, ha una pipeline diversa rispetto a quella del MIPS64.

Infatti è una pipeline composta da 8 stadi.

La parte di **IF** è composta da 2 stadi.

La parte di **data memory** anch'essa viene allungata.

Viene aumentato il **load delay slot** a 2 cicli, ovvero il numero di istruzioni che non soffrono di stalli per operazioni che accedono alla memoria.

Il **branch delay slot** viene incrementato a 3 cicli.

**Esempio d'esame: 17/10/2023, seconda parte della lezione.**