

# DATA PATH COMPLETO

ABBIAMO VISTO LE NOSTRE CLASSI DI ISTRUZIONI E I RELATIVI DATAPATH che vengono coinvolti!  
Per completare il mio datapath devo mettere tutto assieme, quello perché io voglio poter eseguire una qualsiasi istruzione.

- L'unità di calcolo finale può essere realizzata unendo insieme i datapath di tutte le singole classi di istruzioni

PERÒ si crea un problema: **ci sono dei percorsi in conflitto.**

Immaginiamo la fase di WRITEBACK dell'istruzione Load e nel caso dell'istruzioni logico aritmetiche. Nel caso di load la fase di WR deve prendere il codice del registro e il dato proveniente dalla cache.

Nel caso di istruzioni logico aritmetica devo prendere il codice del registro da scrivere, ma il dato lo devo prendere dal risultato propagato dalla ALU.

Se prendo questi datapath e li metto insieme ho due sorgenti diverse per la stessa destinazione.

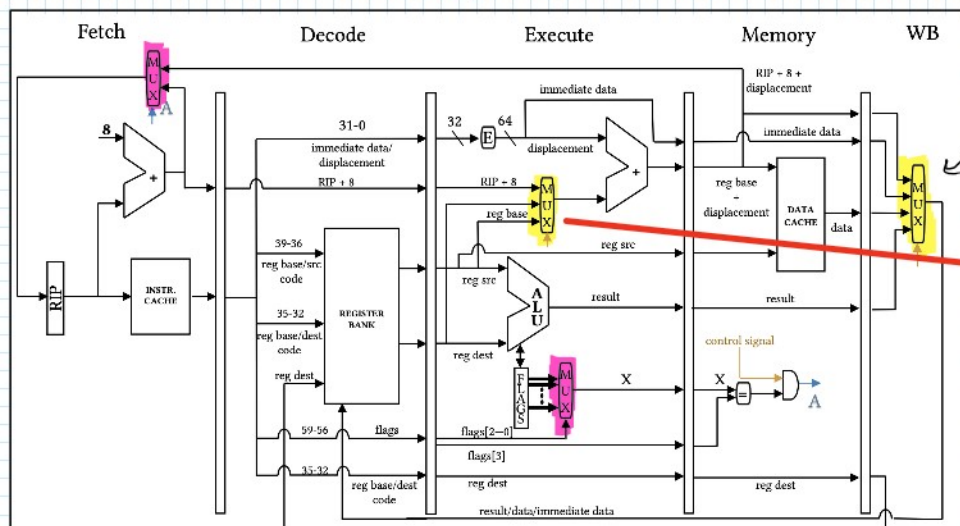
Dove vado a scrivere nel Banco di Registri, prendendo il risultato dalla ALU o dalla memoria?

↓  
(cache)

**DEVO POTER SCEGLIERE.**

**DEVO USARE DEI MULTIPLEXER!** → Dico qual'è la sorgente dati nel caso in cui 2 path sono in conflitto!

- Problema: ci sono alcuni percorsi in conflitto.
  - Esempio: la fase di write back delle istruzioni Load e L/A
  - Nel primo caso, occorre recuperare il valore propagato dalla cache dati
  - Nel secondo caso, occorre recuperare il valore propagato dalla ALU
- Soluzione: utilizzo di multiplexer per selezionare il valore corretto
- Tutti gli altri valori, comunque calcolati dagli stadi della pipeline, vengono ignorati



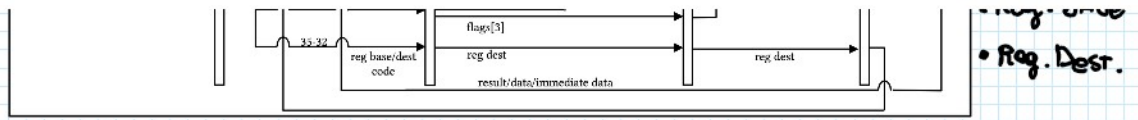
Posso avere più sorgenti connesse per la stessa dest.

Per esempio

questo MUX può sommare sia:

- $(RIP + 8)$
- Reg. base
- Reg. Dest.





chi è che deve guidare questi multiplexer? LA CU, che ancora non abbiamo introdotto; La CU dovrà decidere per tutti quanti gli stadi come devono operare questi multiplexer;

deve provenire dalla fase di decode, perché è lì che abbiamo la rappresentazione binaria della mia istruzione, e da lì sapremo quale segnale di controllo avere per pilotare il multiplexer.

Quando un'istruzione è in fase di decode, l'unità di controllo deve decidere quali sarà il segnale da passare al MUX nella fase di WB.

Ma quando la mia istruzione di load arriva in write back la mia unità di controllo sta decodificando un'altra istruzione.

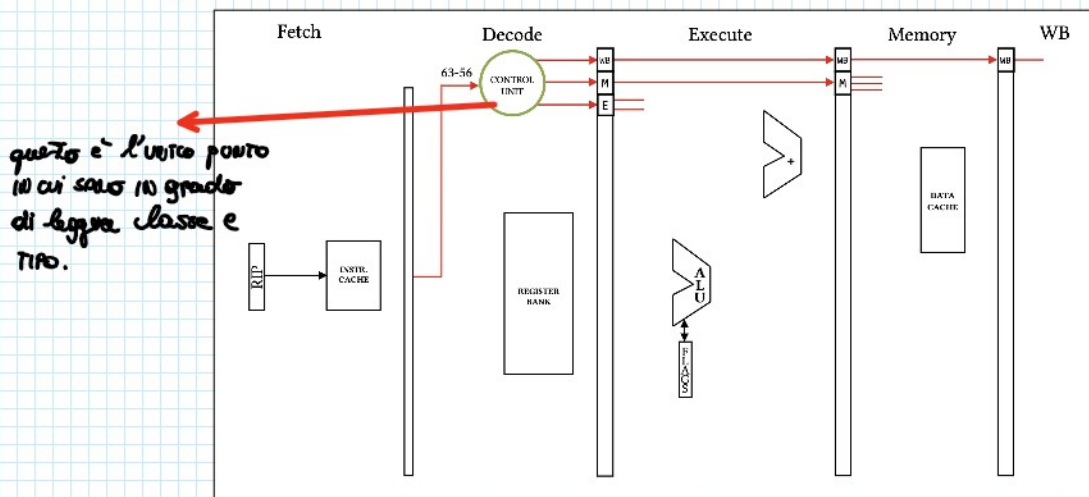
Come risolviamo questo problema? Quando sono in fase di decode decodifico l'istruzione e genero i segnali di controllo per tutte le fasi successive.

Questi segnali di controllo devono essere propagati tramite i Registri di pipeline!

Quindi alla fine i miei registri di pipeline che cosa contengono? Sia i dati necessari a propagare l'esecuzione dell'istruzione, sia i segnali di controllo generati dalla CU in fase di decode.

A ogni colpo di clock tutti i registri di pipeline vengono aggiornati e si passa da una fase precedente a una successiva.

quindi devo organizzare la mia CU in questo modo:



questo è l'unico punto in cui sono in grado di leggere classe e tipo.

A questo punto la CU non è più una macchina a stati come l'organizzazione precedente, ma diventa un circuito combinatorio.

Appena ricevo classe e tipo, genero tutti quanti i segnali di controllo per le fasi successive.