

Istruzioni di salto condizionale: formato

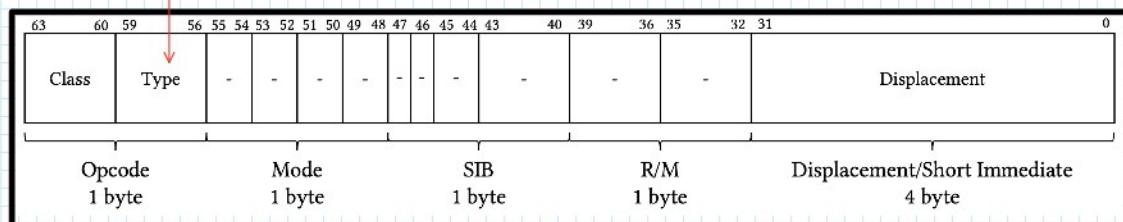
Il Registro RIP e il Registro Flags sono in due stati differenti. Quindi significa che io posso andare a sapere se il salto deve essere eseguito o NO, nella fase di execute, dove c'è il Reg. di FLAGS.

QUINDI UNA FASE SUCCESSIVA ha un IMPATTO SULLA FASE PRECEDENTE. Questa cosa deve essere aperta in qualche modo.

INOLTRE L'ISTRUZIONE di salto condizionale, richiede di sapere qual'è il FLAG che devo andare a controllare e il suo valore.

Dovrò sapere se controllare rispetto al valore zero e rispetto al valore 1.

Questa informazione su qual'è il flag e qual'è il valore del flag che vogliamo andare a vedere, È IL CAMPO TIPO ;



- Il tipo determina sia il flag di interesse, sia il valore di interesse

La classe mi identifica il concetto di salto condizionale, il tipo mi dice quale flag guardare e dove avere questo valore affinché tu possa fare il salto.

Quindi spostiamo la logica di controllo, dalla CU alla rappresentazione binaria della mia istruzione.

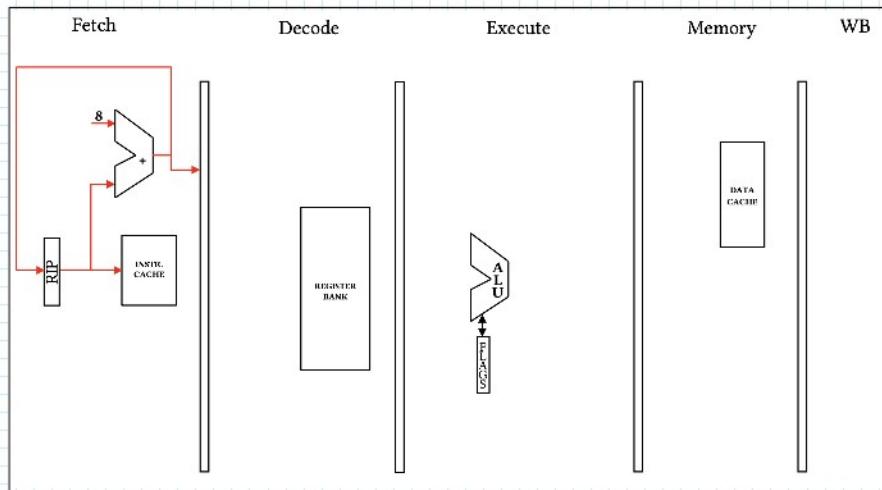
La rappresentazione binaria dell'istruzione, porta con sé già i segnali di controllo che servono a pilotare parte dell'hardware.

Le istruzioni contengono al loro interno dei BIT, che fungono da segnali di controllo !

Per evitare di complicare l'organizzazione della CU che è una macchina a stati.

- Mode: d.c.c.
- SIB: d.c.c.
- R/M: d.c.c.
- D/SI: spiazzamento

Fase di fetch: per ora c'è tutta magia



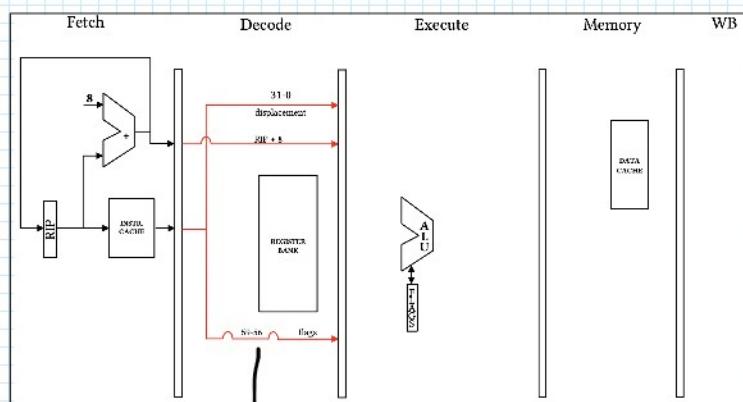
Carcico l'istruzione di salto condizionale, scrivo il valore della mia istruzione nel registro di pipeline, MA LA PIÙ MI DEVO SCRIVERE Rip+8. **PERCHÉ?**

Nel caso di salti lo spiazzamento viene sommato al rip dopo la fase di fetch.

Chi è che ha calcolato Rip dopo la fase di fetch? Il sommatore.

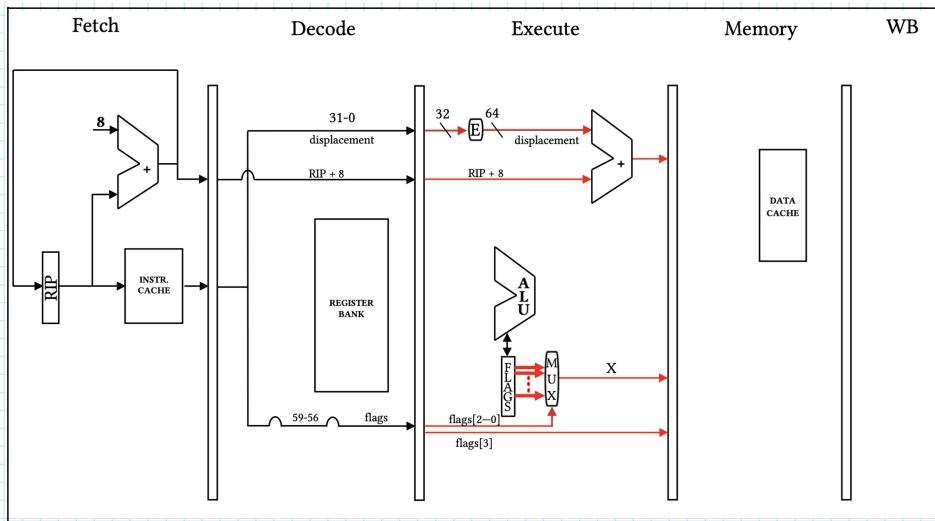
RIP+8 ME LO DEVO PROPAGARE IN AVANTI PERCHÉ DEVO APPLICARE A RIP+8 L'EVENTUALE SPIAZZAMENTO CHE MI SERVIRÀ (SE FACCIO IL SALTO).

Nello stadio successivo mi porto avanti Rip +8, spiazzamento e il tipo!



→ quale flag deve avere quale valore affinché
 $(Rip+8) + \text{spiazzamento}$ debba essere scritto
 all'interno dell'IP.

A questo punto posso utilizzare il sommatore di prima per calcolare displacement + (Rip+8)



Sto usando il sommatore mentre accedo a flags.

Flags è un registro composto da un certo numero di bit di stato, quindi ciascuno dei bit di stato è un segnale che esce da questo registro. Per prendere esattamente il flag che mi interessa ho bisogno di un multiplexer.

→ Mi permette di tirare fuori un solo bit che chiamiamo X , tra gli N bit di flags.

Quale bit prendo? Dipende dal valore che è scritto all'interno di tipo!

Il tipo dell'istruzione va a pilotare il multiplexer per prendere solo il flag di interesse del registro Flags.

Quindi nel registro di interfaccia mi scrivo un bit (0 o 1) t_1

che è il bit che devo andare a controllare per decidere se fare o meno il salto.

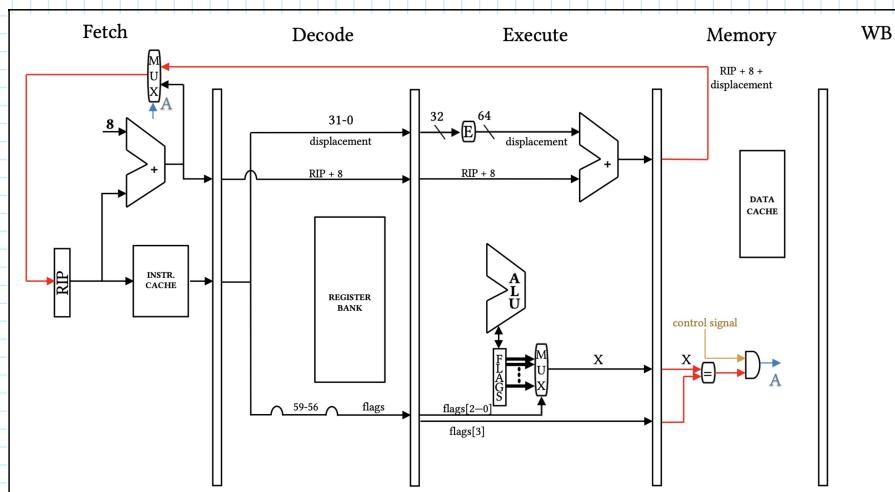
Se il salto lo faccio devo prendere $RIP + displacement$ e riscrivere dunque al registro RIP.

Che valore deve avere questo X affinché debba essere scritto dietro al registro RIP?

dipende dal bit del tipo che mi dice se devo fare un salto a bit impostato o non impostato.

Quindi cosa devo fare per decidere se $(RIP + 8) + displacement$ debba essere scritto nel registro RIP? devo confrontare questi due valori e devo lo essere uguali.

Ho bisogno di un comparatore a 2 bit.



Il bit preso dal tipo è il bit di flag che ho selezionato con il multiplexer devo essere uguale. (Questo mi dà solo informazioni sul salto, non so dire se lo devo fare)

Il bit Atteso preso dal tipo e il bit di flag che ho selezionato con il multiplexer devono essere uguali. (Questo mi dà solo informazioni sul salto, non indica se lo devo fare).

Ora se io voglio fare il salto devo scrivere su Rip, ma in parallelo sulla fase di fetch qualcosa sta calcolando $RIP + \delta$, quindi devo scegliere quale dei due valori scrivere su Rip.

USO UN MULTIPLEXER;

↳ PILOTATO da A, ossia il risultato del mio comparatore.

Quindi $(RIP + \delta) + \text{displacement}$ lo mando ad un multiplexer, il multiplexer o prende $RIP + \delta$ o il vecchio $(RIP + \delta) + \text{displacement}$, e lo scriverà all'interno del registro Rip.

Però mi serve un segnale di controllo che mi dica che sto eseguendo un'istruzione di salto condizionale.

Se sto eseguendo un salto condizionale E il bit di flag ha il valore atteso allora Devo aggiornare il valore di Rip.

Così tramite l'ultimo AND otengo il segnale di controllo A.

NELLA fase di write back non facciamo nulla.