Istr.Load - funzionamento

martedì 28 febbraio 2023 15:05

CARICAMENTO DA MEMORIA

ABBIAMO BISOGNO di UN REGISTRO DI BASE e di UNO SPIAZZAMENTO

E UN REG. DESTINAZIONE perchè CARICHIAMO da MEMORIA PER SCRIVERE all'INTERNO di UN REGISTRO.

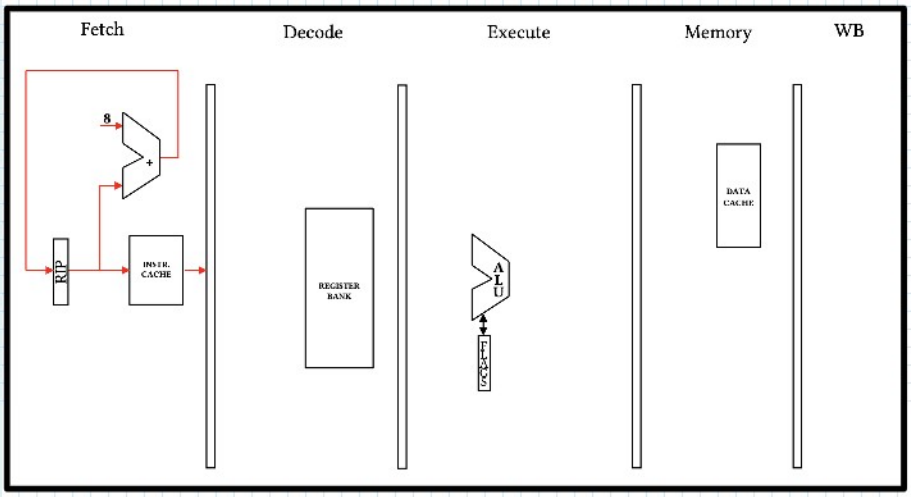
Istruzione Load: formato

1. B_p = 1: utilizziamo il registro base
2. Displacement è una d.c.c.: non si possono usare costanti e non si possono effettuare operazioni in memoria
3. SS/DS = 11: solo operandi a 64 bit
4. DI = 10: è utilizzato uno spiazzamento
5. Mem = 10: il primo operando è in memoria

DOVREMO CARICARCI IL REGISTRO BASE, CARICARCI IL REGISTRO DESTINAZIONE, calcolare BASE+SPIAZZAMENTO, ANDARE IN MEMORIA, CARICARE IL VALORE della memoria all'INDIRIZZO calcolato e SCRIVERE IL RISULTATO NEL REGISTRO DESTINAZIONE.

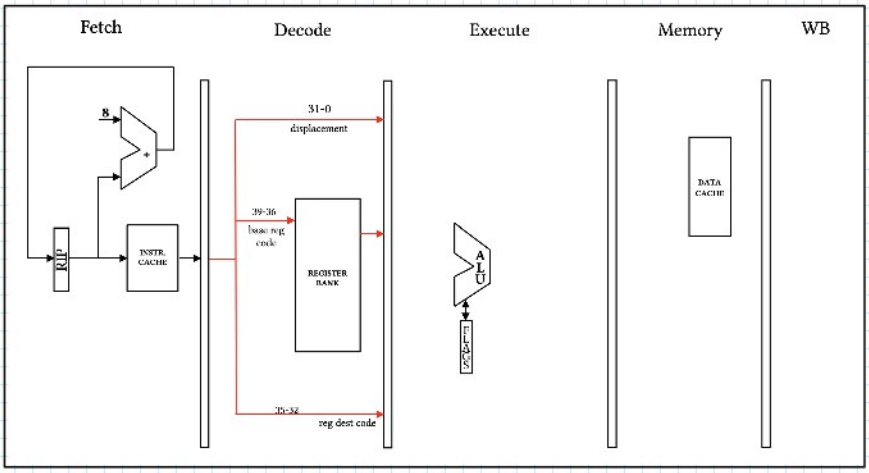
Tutto PARTE CON LA FASE di fetch, che è identica per le OPERAZIONI Logiche ARITMETICHE!

CARICO l'ISTRUZIONE da memoria, SCRIVO IL valore della RAPPRESENTAZIONE BINARIA della mia ISTRUZIONE NEL REGISTRO di pipeline, INCREMENTO RIP.



NELLA FASE di decode ho BISOGNO DEL REGISTRO BASE sorgente, ed è l'unico REGISTRO che devo andare a caricare.

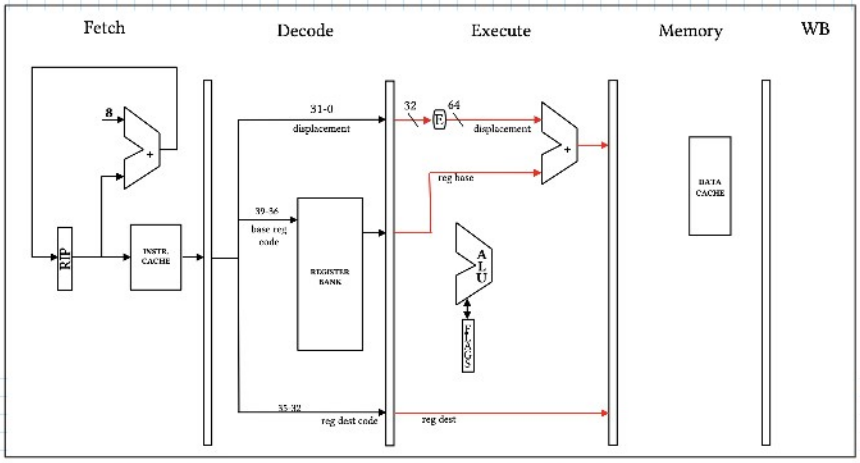
Significa che questo Banco dei REGISTRI mi deve tirare fuori un solo valore. IL codice del REGISTRO di DESTINAZIONE lo dobbiamo propagare come prima, in più dobbiamo propagare anche lo SPIAZZAMENTO, PERCHÈ dobbiamo andare a calcolare BASE + SPIAZZAMENTO.



DOVE CALCOLEREMO BASE + SPIAZZAMENTO? NELLA FASE DI EXECUTE

MA in questo caso, per comodità, aggiungiamo un sommatore dedicato PER NON INFLUENZARE IL REGISTRO FLAGS.

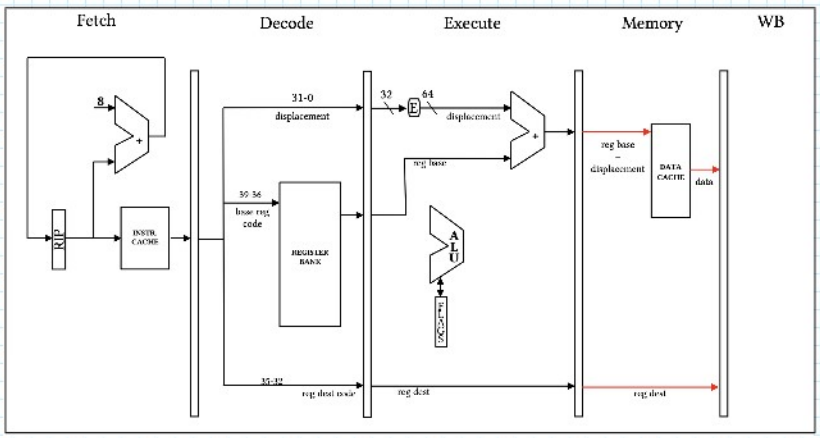
Ogni volta che usiamo la ALU, verrà aggiornato il REGISTRO flags.



Lo spiazzamento è a 32 bit, il contenuto del registro di base è a 64 bit, per poter sommare il registro di base con lo spiazzamento dobbiamo estendere il segno del contenuto dello spiazzamento per passarlo da 32 a 64.

-3 a 32 bit deve rimanere -3 a 64 bit.

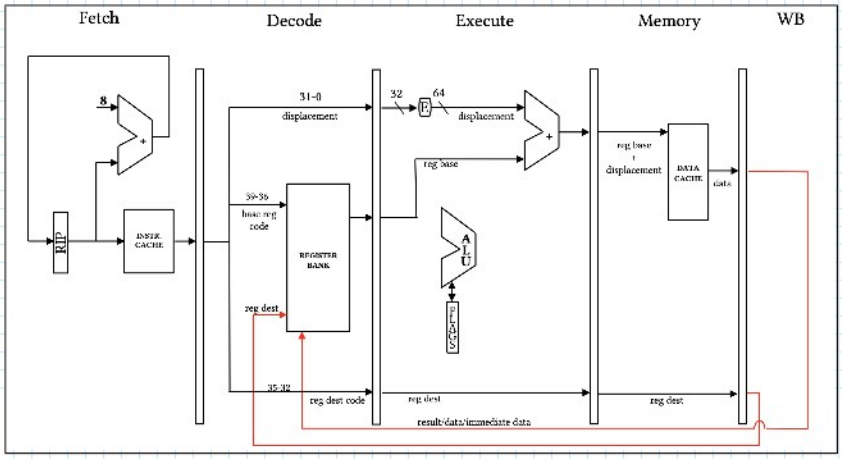
NELLA FASE DI MEMORY leggo DALLA MEMORIA, quindi devo passare alla data cache un indirizzo, ossia il risultato della somma prima effettuata.



Mentre il registro destinazione lo propago in avanti perché mi servirà nella fase di WRITE BACK.

ABBIAMO il valore da scrivere, ABBIAMO il codice del REGISTRO verso cui fare la SCRITTURA,

ABBIAMO IL VALORE DA SCRIVERE, ABBIAMO IL CODICE DEL REGISTRO VERSO CUI FARE LA SCRITTURA, E NELLA FASE DI WRITEBACK, PASSO IL DATO E IL CODICE, E SCRIVO.



NON SI POSSONO MANDARE AVANTI GLI STATI PRECEDENTI SE IL SUCCESSIVO NON È COMPLETO!

IN QUESTA ARCHITETTURA PIPELINE, TUTTO ASPETTA.