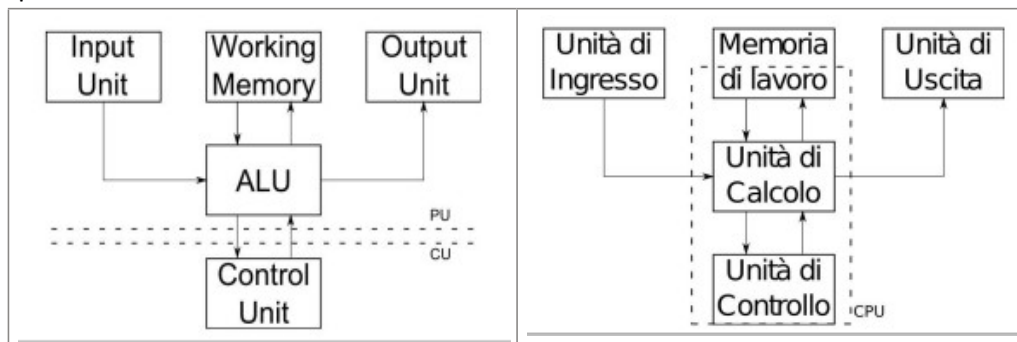


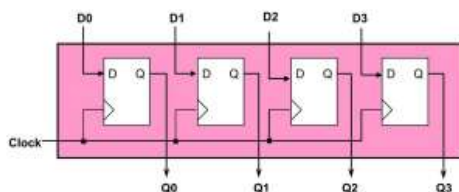
# Lezione 15 Z64

giovedì 2 novembre 2023 15:46

Andiamo ora a vedere il processore Z64 : ricordiamoci che questo processore si basa su architettura di von Neumann . Questo processore implementa un sotto insieme delle funzionalità dei processori intel . Composto da circuiti pilotabili tramite sw . **L'ISA (instruction set architecture)** di un processore definisce tutto quello che può fare e come lo fa. Esistono due famiglie **RISC e CISC**: nella prima si ha un insieme ridotto di istruzioni ,programmi più lunghi , più efficiente dal punto di vista energetico e di più semplice progettazione . Inversamente ai CISC , ci sono i RISC : aumenta la quantità di istruzioni , programmi più compatti , aumento dal punto di vista energetico ed aumenta la complessità della progettazione (processore più grande e circuiteria annessa) . Per quanto riguarda l'architettura , la dobbiamo rivisitare , in quanto se fosse quella tradizionale ed avendo che il processore è molto veloce rispetto ai registri, portiamo un pezzo di memoria di lavoro all'interno del processore . Quindi in sintesi :



Per quanto detto finora , nel processore abbiamo detto che ci va un pezzo della memoria di lavoro , e questo pezzo della memoria di lavoro è costituita da **registri** : blocchetti di memoria di 1 kb , i quali sono organizzati a parole binarie . Sono realizzati tramite FF ( unico circuito che immagazzina un bit in memoria ) : vengono messi in cascata un numero di FF:

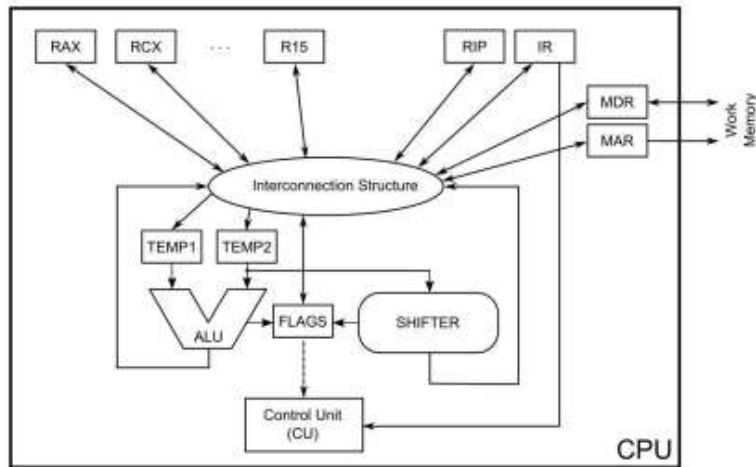


La quale velocità viene decisa dal clock .

I registri fanno parte dell'isa . Non tutti i registri sono esplicitamente utilizzabili : i registri si dividono in :

1. Fondamentali
  - a. Necessari per soddisfare architettura di un processore
2. Visibili al programmatore
  - a. Il programmatore può interagire per letture/scritture dati
3. Invisibili al programmatore
  - a. Il programmatore non può leggere/scrivere: sono le istruzioni che vanno a modificare questi registri.

L'architettura dello z-64 è la seguente:



In dettaglio : i registri RAX,RXB ,...R15 sono 16 registri a 64 bit che possiamo usare : in realtà ne abbiamo 14 da usare. MAR,MDR sono registri tampone che costituiscono interfaccia verso il mondo esterno : velocità molto alte e costituiscono buffer tra 2 reti sequenziali a velocità diversa. RIP (instruction pointer) importante registro che ricorda al processore l'ultima sequenza delle istruzioni : tiene traccia del programma . ALU+Shifter reti iterative che ci permettono di eseguire operazioni su parole ad n bit : nel caso di quelle veloci servono anche i bit di stato , i quali vengono immagazzinati nel registro flags (FF) . Non dimentichiamoci della struttura di interconnessione (fili che fanno viaggiare le istruzioni) . L'unità di controllo è (gigante) rete sequenziale che pilota il nostro hardware , genera i bit di controllo che vanno verso tutti i componenti per configurarli in modo corretto. Vediamo ora come mantenere i dati stabili (vedasi discorso del tempo di propagazione) tra un colpo di clock ed il successivo : usiamo il registro IR (instruction register e registro tampone) il quale mantiene la codifica binaria dell'istruzione che il processore esegue . Vediamo ora come vengono processate le istruzioni :

### 1. Fetch

- a. Vado in memoria a recuperare la codifica binaria della mia istruzione macchina da eseguire e la metto nel registro IR . Conversione/codifica input/output fatta dal processore

### 2. Decode

- a. Input dal registro IR viene interpretato in sequenza

### 3. Execute

- a. Traduzione istruzione macchina in istruzioni interne , quindi esecuzione sequenza passi per eseguire istruzione.