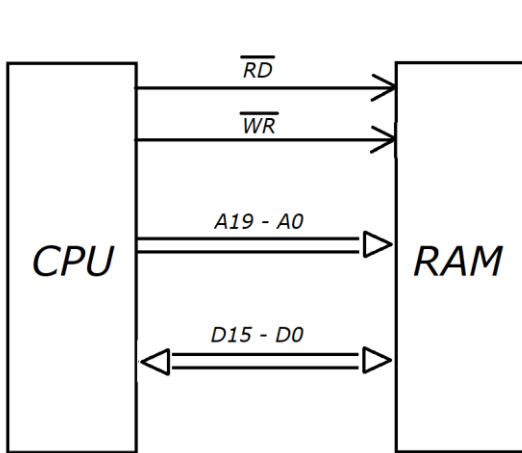


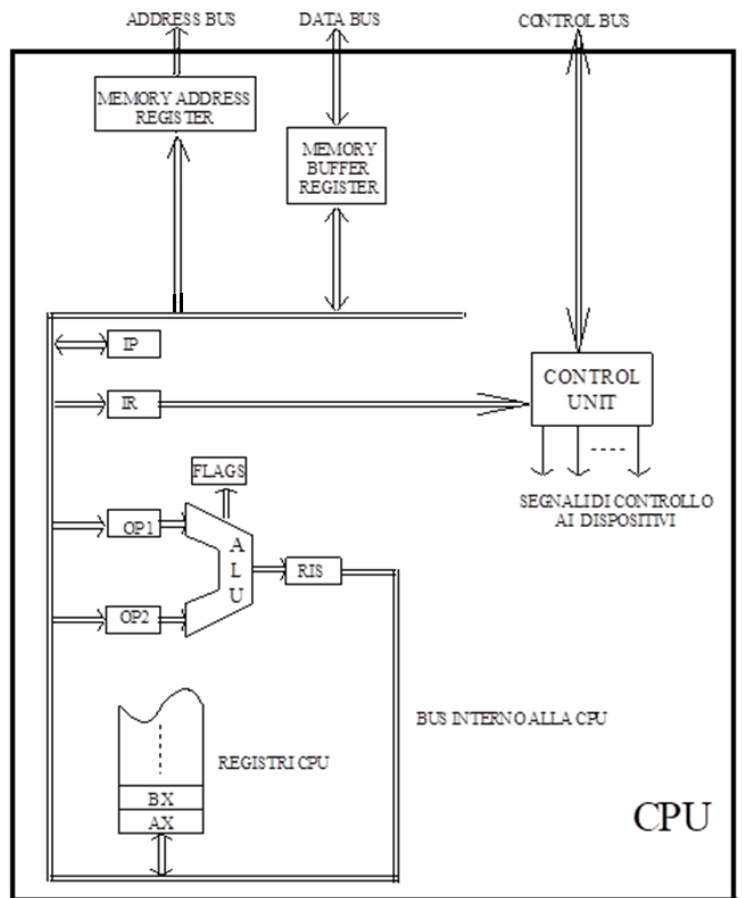
## 1 Il ciclo di Fetch Decode Execute

Il ciclo di esecuzione di un'istruzione o **ciclo di fetch decode execute** esplicita la sequenza di passi che la CPU deve fare per eseguire un'istruzione in linguaggio macchina.



Le entità del microprocessore coinvolte nell'esecuzione di un'istruzione sono:

- la **CU (Control Unit)**, che in seguito alla decodifica dell'istruzione in corso di esecuzione attiva mediante opportuni segnali i dispositivi coinvolti nell'esecuzione dell'istruzione stessa;
- la **ALU (Arithmetic Logic Unit)**, che esegue le operazioni coinvolte nell'esecuzione dell'istruzione;
- il registro **IR (Instruction Register)**, che contiene il codice operativo dell'istruzione in corso di esecuzione;
- il registro **IP (Instruction Pointer)**, che contiene l'offset rispetto all'inizio del Code Segment dell'istruzione successiva a quella in corso di esecuzione;
- i **registri della CPU** che possono comparire come operandi delle istruzioni;
- il **registro MAR (Memory Address Register)**, che costituisce l'interfaccia tra il bus interno della CPU e l'Address Bus esterno;
- il **registro MBR (Memory Buffer Register)**, che costituisce l'interfaccia tra il bus interno della CPU e il Data Bus esterno.



Quando un programma viene mandato in esecuzione, il Sistema Operativo lo carica in memoria centrale ricavando nelle zone di memoria ancora libere spazio sufficiente per contenere il programma.

Dopo aver caricato il programma, il Sistema Operativo memorizza nel registro IP l'indirizzo della prima istruzione del programma e pertanto la successiva istruzione che verrà eseguita sarà effettivamente la prima istruzione del programma mandato in esecuzione.

Il ciclo di esecuzione di un'istruzione si divide in tre fasi distinte: la fase di **fetch** o di *prelievo dell'istruzione*, la fase di **decode** o di *decodifica del codice operativo dell'istruzione* e la fase di **execute** o di *esecuzione dell'istruzione*.

### Fase di fetch

Lo scopo della fase di fetch è di leggere l'istruzione dalla memoria e di memorizzarla nel registro d'istruzione (Instruction Register: IR). La sequenza di passi necessari per raggiungere tale scopo è:

- Trasferimento dell'indirizzo fisico dell'istruzione nel registro MAR (Memory Address Register).** L'indirizzo dell'istruzione contenuto nel registro IP viene trasferito nel MAR che interfaccia la CPU con l'Address Bus;
- Trasferimento del contenuto del MAR sulle linee di Address Bus.** Il contenuto del Memory Address Register viene caricato sulle linee dell'Address Bus esterno e contemporaneamente la CPU genera un segnale di lettura verso la memoria ponendo a livello logico 0 la linea  $\overline{RD}$ ;
- Decodifica dell'indirizzo dell'istruzione da parte della memoria.** La memoria decodifica l'indirizzo presente sulle linee dell'Address Bus, legge l'istruzione nelle locazioni individuate e la pone sulle linee del Data Bus esterno;

- **Caricamento dell'istruzione in IR.** La CPU carica l'istruzione dal Data Bus esterno nel Memory Buffer Register e successivamente nel registro IR;
- **Aggiornamento di IP.** Il registro IP viene incrementato affinché contenga l'indirizzo dell'istruzione successiva all'istruzione in corso di esecuzione; l'aggiornamento avviene sommando al contenuto di IP la dimensione in byte dell'istruzione in corso di esecuzione.

## Fase di decode

Nella fase di decode l'unità di controllo decodifica il codice operativo dell'istruzione per determinare i segnali di controllo da generare successivamente e che abilitano i dispositivi coinvolti nell'esecuzione dell'istruzione stessa. Esistono alcune istruzioni che non richiedono operandi, quale, ad esempio l'istruzione NOP che non ha alcun effetto e viene quindi utilizzata per introdurre delle attese all'interno del programma. In questo caso, appropriati segnali di controllo possono svolgere direttamente l'azione necessaria e quindi il ciclo dell'istruzione termina qui. Se l'istruzione implica qualche operando, allora si entra nella fase di execute.

## Fase di execute

La fase di execute ha come scopo l'esecuzione dell'istruzione con l'eventuale memorizzazione dei risultati e si divide nelle seguenti sottofasi:

- **Determinazione degli indirizzi degli operandi.** Gli operandi coinvolti possono essere memorizzati internamente in un registro della CPU, in una variabile di memoria oppure l'operando può essere una costante. Esaminiamo separatamente i tre casi:
  - **REGISTRO.** Nel caso di operando contenuto in un registro, la CU provvederà ad abilitare in lettura il registro coinvolto nell'operazione.
  - **VARIABILE DI MEMORIA.** Se l'operando dell'istruzione è una variabile di memoria, nel campo operando dell'istruzione viene indicato l'indirizzo della variabile di memoria. Questo indirizzo, in modo del tutto analogo al procedimento seguito nella fase di fetch per l'istruzione, è trasferito nel registro MAR.
  - **COSTANTE.** Un operando costante di un'istruzione è memorizzato direttamente all'interno del codice del programma. Per determinare l'indirizzo fisico della costante viene sottratto 1 o 2 al contenuto di IP a seconda che la costante abbia la dimensione di 1 o 2 byte.
- **Prelievo degli operandi.** Il prelievo degli operandi assume caratteristiche diverse a seconda della locazione in cui è memorizzato l'operando. Esaminiamo i 3 casi:
  - **REGISTRO.** Nel caso di operando contenuto in un registro la CU, dopo aver abilitato in lettura il registro coinvolto nell'operazione, abilita il registro in input alla ALU in scrittura e vi trasferisce il contenuto del registro operando dell'istruzione.
  - **VARIABILE DI MEMORIA.** Il contenuto del Memory Address Register viene caricato sulle linee dell'Address Bus esterno e contemporaneamente la CU genera un segnale di lettura verso la memoria ponendo a livello logico 0 la linea RD\*. Alla ricezione dell'indirizzo, la memoria decodifica l'indirizzo presente sulle linee dell'Address Bus, legge il dato nelle locazioni individuate e lo pone sulle linee del Data Bus esterno. La CPU carica il dato dal Data Bus esterno nel Memory Buffer Register e successivamente nel registro in input alla ALU.
  - **COSTANTE.** Il procedimento è del tutto analogo a quello seguito per le variabili di memoria.
- **Operazioni sugli operandi.** I due operandi sono ora contenuti nei registri in input alla ALU. I segnali di controllo dall'unità di controllo attivano l'ALU, cosicché viene eseguita l'operazione specificata dall'istruzione. Il risultato dell'operazione viene trasferito nel registro in output alla ALU tramite il bus interno. Ciò completa l'esecuzione dell'istruzione nel caso in cui il risultato non debba essere memorizzato, come nel caso dell'istruzione CMP in cui viene eseguita una sottrazione il cui risultato non viene memorizzato perché nell'istruzione di Jump condizionata successiva vengono soltanto esaminati i flags.
- **Memorizzazione del risultato (se necessario).** La memorizzazione del risultato, se richiesta dall'istruzione, avverrà o in un regione o in una locazione di memoria.
  - **REGISTRO.** Nel caso di risultato da memorizzare in un registro la CU abilita in scrittura il registro che dovrà contenere il risultato, consentendo così il trasferimento del valore contenuto all'interno del registro in output alla ALU.
  - **VARIABILE DI MEMORIA.** Dopo aver trasferito in MAR l'indirizzo dell'area di memoria che deve contenere il risultato, il contenuto del Memory Address Register viene caricato sulle linee dell'Address Bus esterno; contemporaneamente la CU genera un segnale di scrittura verso la memoria ponendo a livello logico 0 la linea WR\*. Alla ricezione dell'indirizzo, la memoria decodifica l'indirizzo presente sulle linee dell'Address Bus. La CU pone

successivamente sulle linee di Data Bus il valore contenuto nel registro in output alla ALU. La memoria legge il dato sulle linee del Data Bus esterno e lo memorizza nella locazione individuata con l'operazione di decodifica dell'indirizzo.

A questo punto termina il ciclo di esecuzione di un'istruzione; la CPU ripeterà un nuovo ciclo di esecuzione dell'istruzione eseguendo l'istruzione il cui indirizzo è specificato all'interno del registro IP.