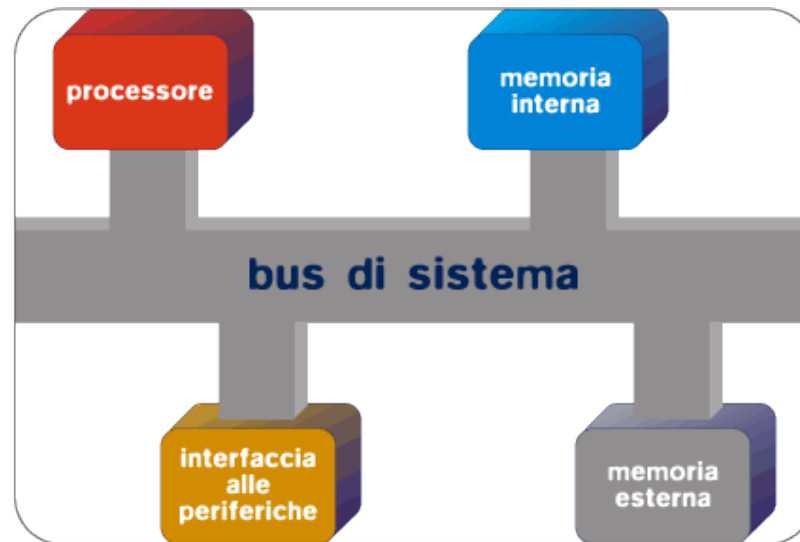


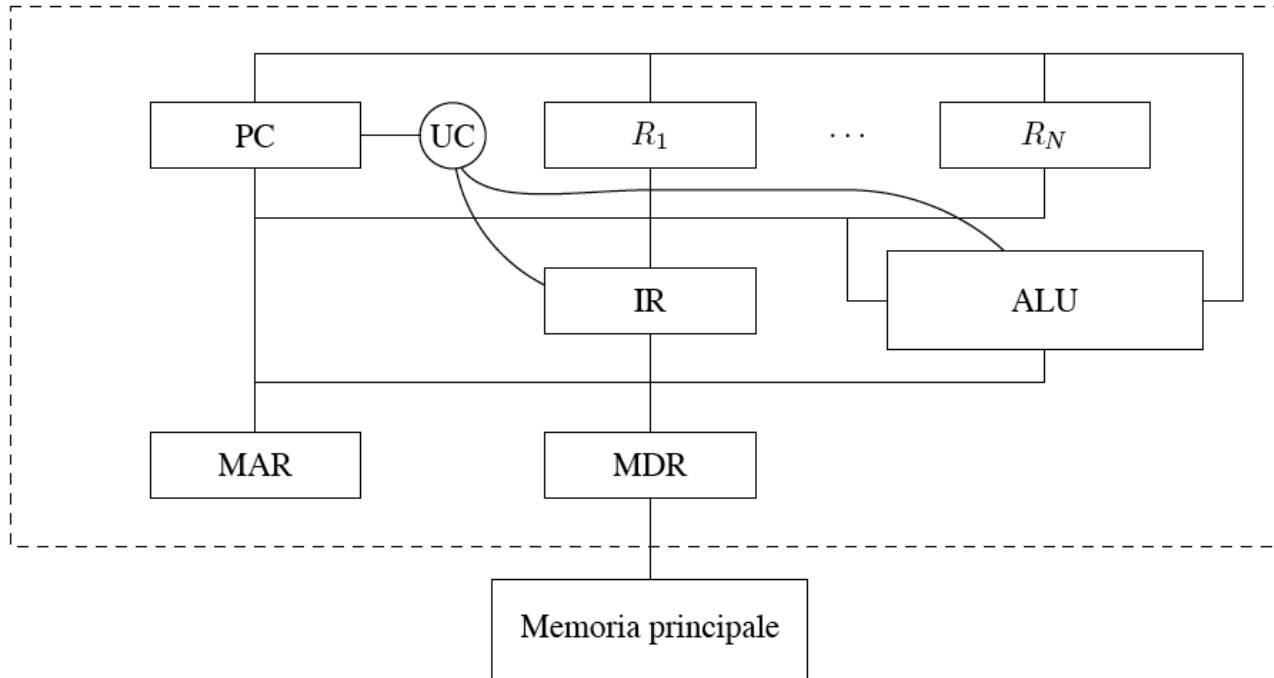
la macchina di von Neumann

Le tecnologie sono basate su una struttura estremamente semplice



...ma anche estremamente flessibile e suscettibile di molte varianti

il processore tipico



può essere complicato a piacere:

architetture parallele

con *piping*

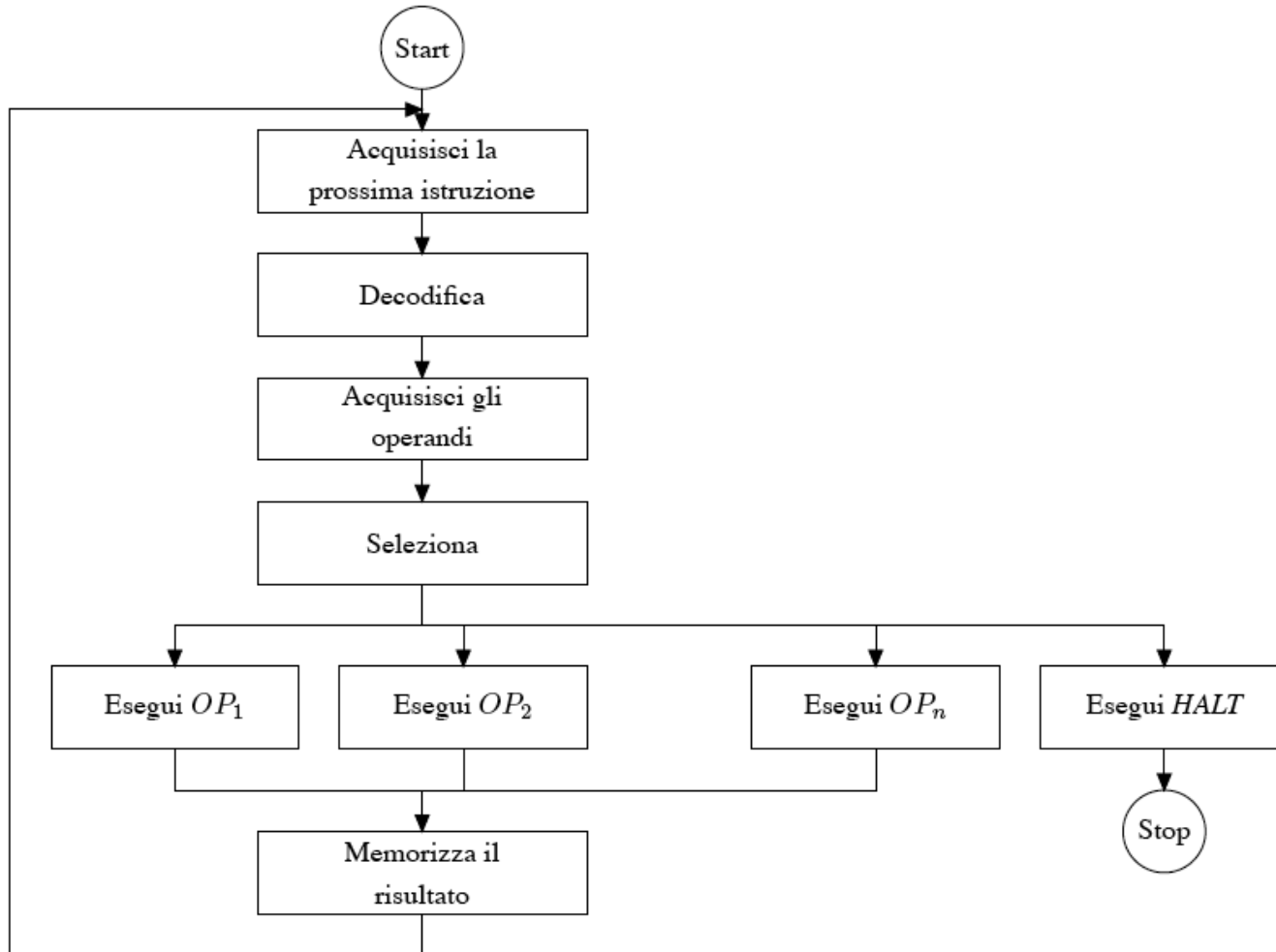
con *prefetch*

La macchina fisica

- **Una** macchina esiste per eseguire il **suo** linguaggio
- Linguaggio e macchina esistono in simbiosi. Ma:
 - una macchina corrisponde ad un linguaggio (il suo)
- Cuore di una macchina:
 - il ciclo fondamentale *fetch-decode-execute*: l'interprete

Una macchina fisica è la realizzazione “a fili” di un particolare algoritmo che, sfruttando alcune strutture dati, è capace di “eseguire” i programmi scritti in un certo linguaggio

Interprete

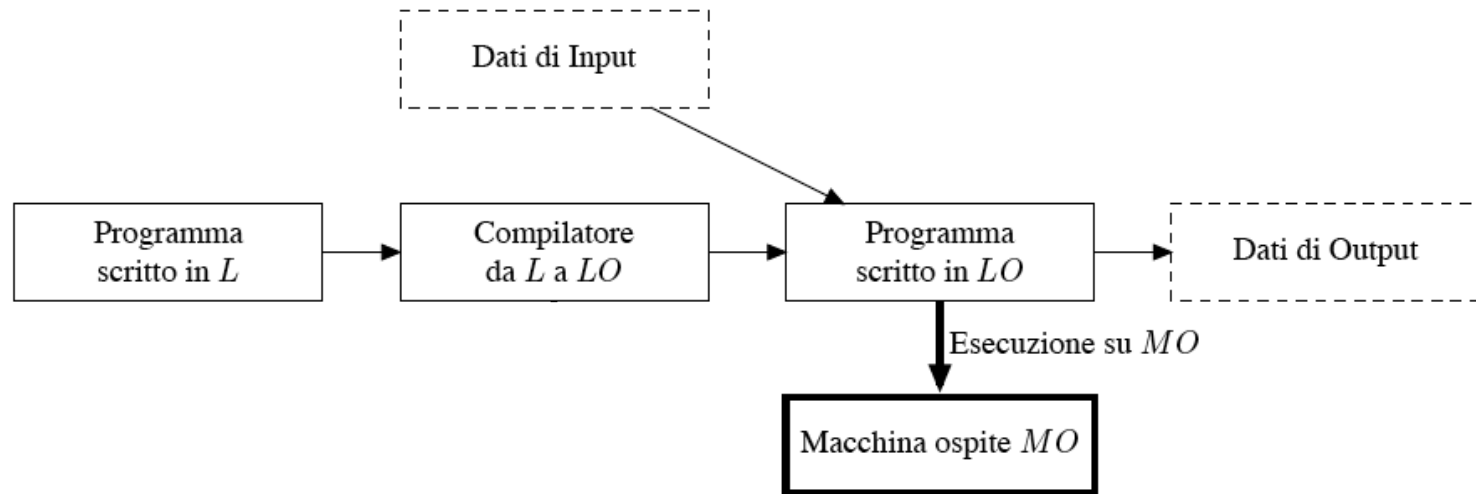


Implementare un linguaggio

- Sono dati:
 - un linguaggio \mathcal{L} da implementare
 - una macchina M_0 (macchina *ospite*)
 - col suo linguaggio \mathcal{L}_0
- Si vuole
 - implementare \mathcal{L} su M_0
- Due modi radicalmente diversi...

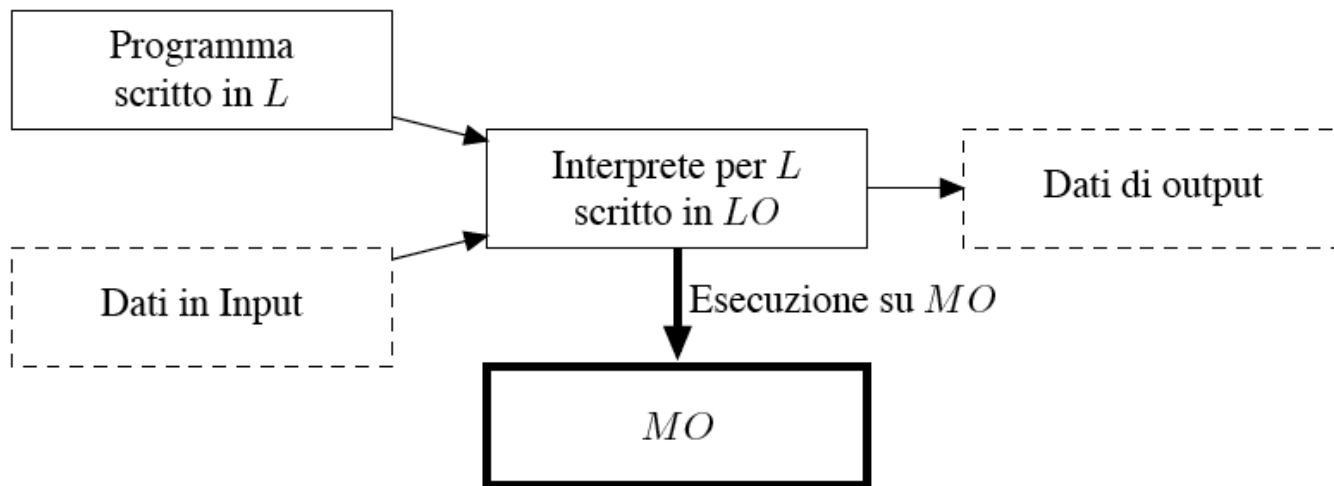
Implementazione compilativa pura

- I programmi in \mathcal{L} sono *tradotti* in programmi *equivalenti* in \mathcal{LO}
- Traduzione effettuata da un altro programma detto compilatore



Implementazione interpretativa pura

- Si realizza scrivendo un *interprete* per \mathcal{L} su MO :

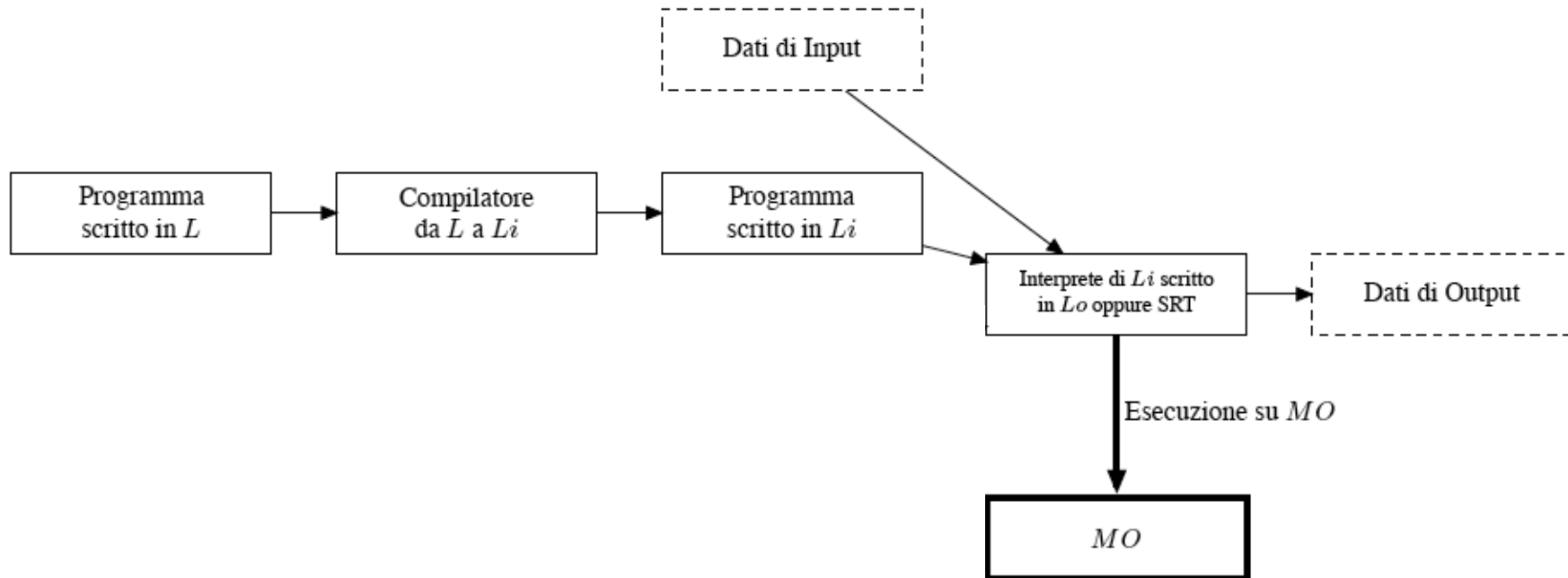


Compilazione o interpretazione ?

- Implementazione **interpretativa** pura:
 - scarsa efficienza
 - buona flessibilità e portabilità
 - facilità di interazione a run-time (es. debugging)
- Implementazione **compilativa** pura:
 - difficile, se \mathcal{L} e \mathcal{LO} sono molto distanti
 - buona efficienza (decodifica a carico del compilatore e ottimizzazioni)
 - scarsa flessibilità e portabilità
 - perdita di informazione sulla struttura del programma sorgente

Nel caso reale

- Entrambe le componenti coesistono favorendo portabilità'



Implementazione di tipo interpretativo



- Linguaggi tipicamente implementati in modo interpretativo
 - LISP, ML, Perl, Postscript, Pascal, Prolog, Smalltalk, Java, Python

Implementazione di tipo compilativo



- Linguaggi tipicamente implementati in modo compilativo
 - C, C++, FORTRAN, Pascal, ADA