

Corso di Calcolatori Elettronici I

Architettura di un calcolatore: introduzione

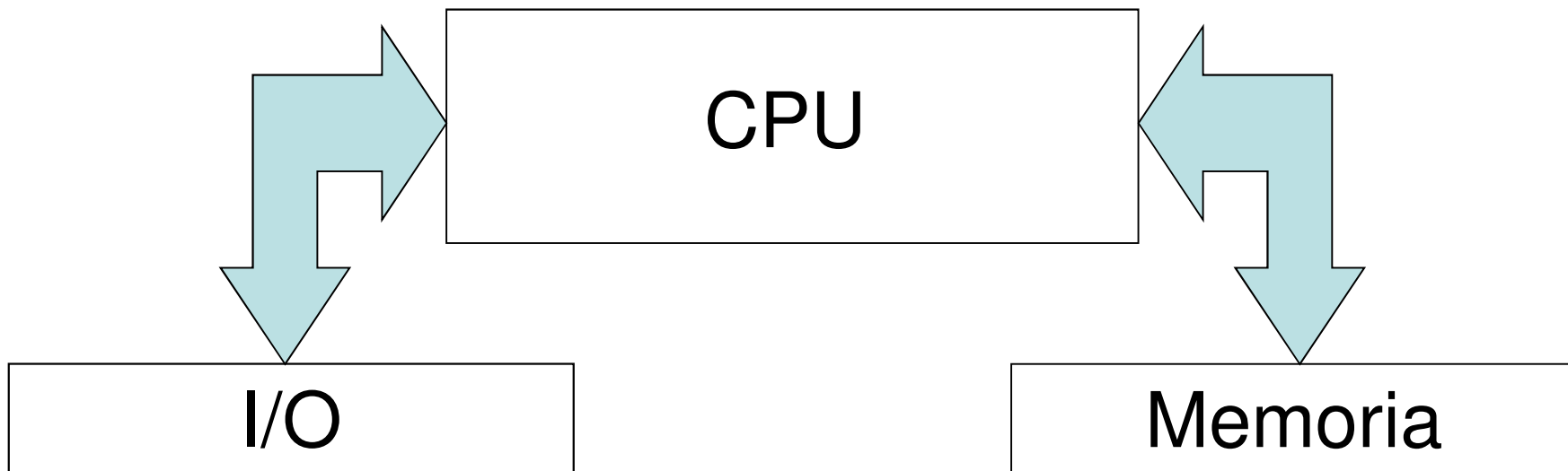
Prof. Roberto Canonico



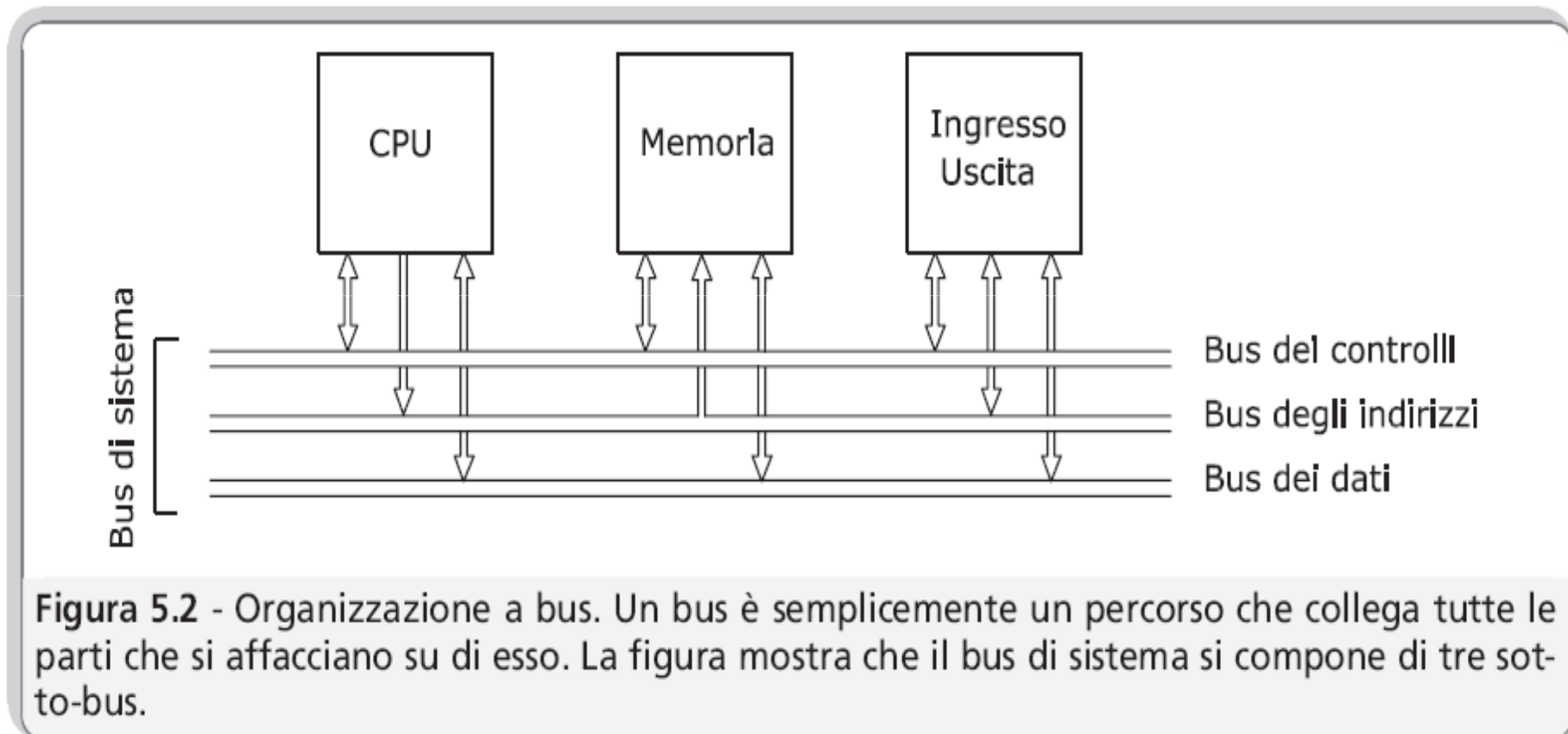
Università degli Studi di Napoli Federico II
Dipartimento di Ingegneria Elettrica
e delle Tecnologie dell'Informazione
Corso di Laurea in Ingegneria Informatica
Corso di Laurea in Ingegneria dell'Automazione

Calcolatore: sottosistemi

- Processore o CPU (*Central Processing Unit*)
- Memoria centrale
- Sottosistema di input/output (I/O)



Calcolatore: organizzazione a bus



Il processore o CPU



Processore o CPU

CPU: struttura interna

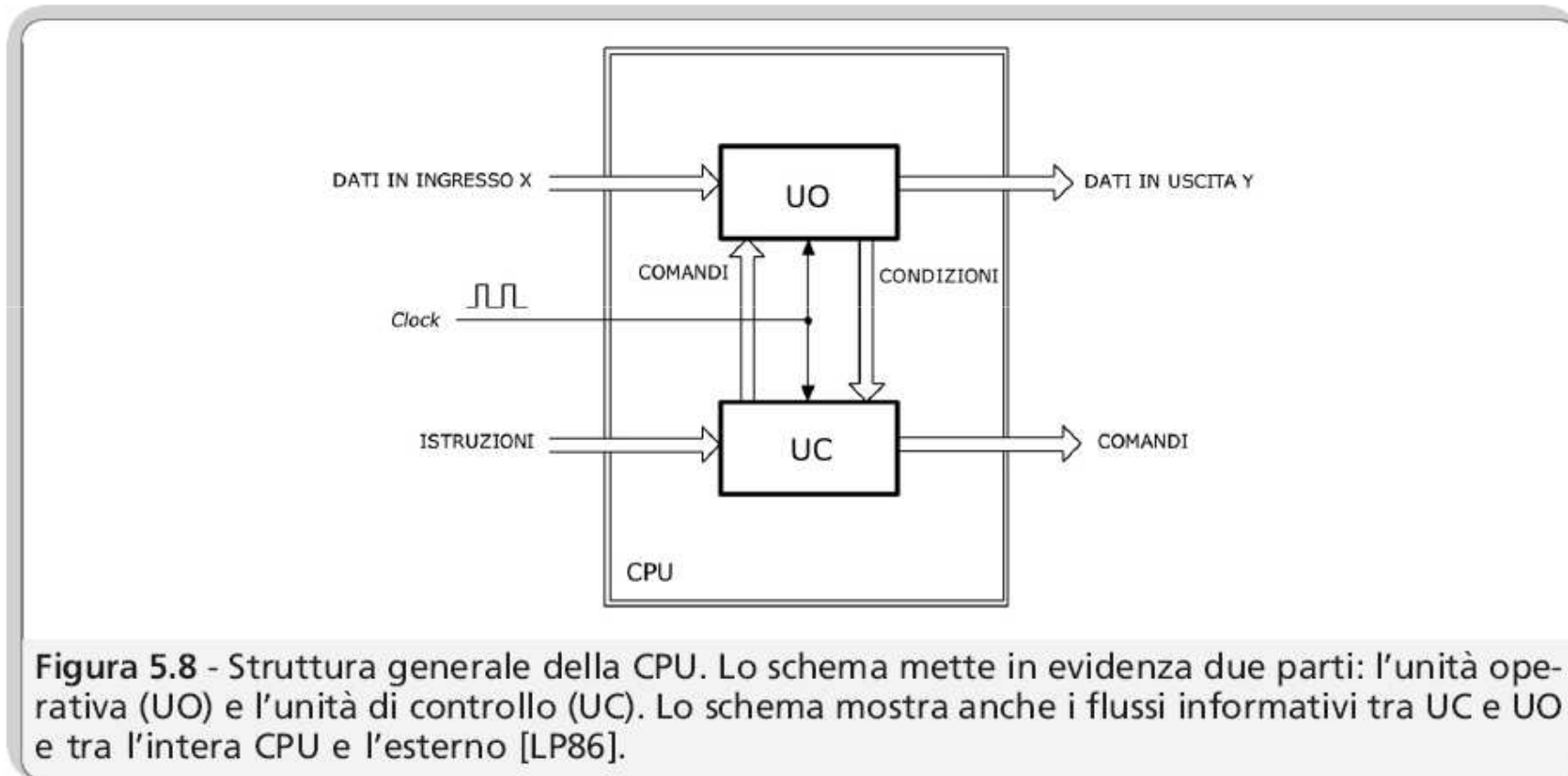


Figura 5.8 - Struttura generale della CPU. Lo schema mette in evidenza due parti: l'unità operativa (UO) e l'unità di controllo (UC). Lo schema mostra anche i flussi informativi tra UC e UO e tra l'intera CPU e l'esterno [LP86].

Il funzionamento della CPU è scandito dal clock.

CPU: struttura interna (2)

- Componenti fondamentali del processore:
 - Unità di controllo
 - registro Program Counter (PC) o Prossima Istruzione
 - Instruction Register o registro di decodifica (IR o D)
 - registri di Macchina
 - Unità aritmetico-logica (ALU)
 - Sezione di Collegamento con la memoria
 - registro degli indirizzi di memoria o Memory Address Register MAR
 - registro di transito dei dati dalla memoria DTR o Memory Buffer MB
 - Sezione di Collegamento con Ingresso-Uscita

 - Il *linguaggio macchina* di un processore è costituito dalla codifica in binario delle istruzioni eseguibili dal processore
-

Registri della CPU

➤ Registri interni

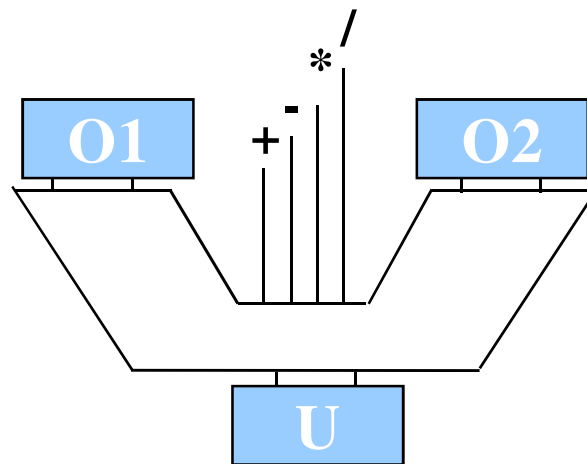
- » Necessari al funzionamento del processore
- » Non direttamente visibili al programmatore
 - » non appartengono al *modello di programmazione*
 - » *Es. MAR, MDR, IR, ...*

➤ Registri di macchina

- » Visibili al programmatore
 - » appartengono al *modello di programmazione*
 - Registri generali (R0, R1, Rn-1)
 - Registri speciali (PC, SR, ...)
-

Unità Aritmetico-Logica (ALU)

- L'Unità di controllo fornisce alla ALU gli operandi, insieme ad un comando che indica l'operazione da effettuare
- Gli operandi sono copiati nei registri di ingresso della ALU (O1, O2)
- La ALU esegue l'operazione e pone il risultato nel registro risultato (U); inoltre, altera il valore dei flag del registro di stato (SR) in funzione del risultato



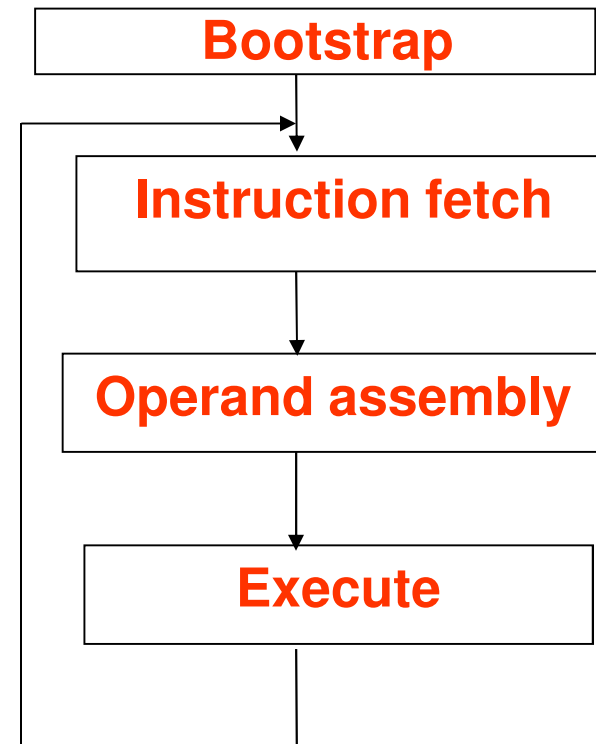
Algoritmo del Processore

- **Prelievo dell'istruzione (Fetch)**
 - La CPU preleva dalla memoria l'istruzione il cui indirizzo è in PC
 - L'istruzione viene copiata nel registro IR
 - **Decodifica / prelievo degli operandi (Operand Assembly)**
 - L'unità di controllo esamina il contenuto di IR e ricava il tipo di operazione ed i relativi operandi
 - Eventuali operandi contenuti in memoria vengono prelevati
 - **Esecuzione dell'istruzione (Execute)**
 - L'unità di controllo richiede all'ALU di effettuare l'operazione specificata nell'istruzione ed invia il risultato ad un registro o alla memoria
-

Algoritmo del processore

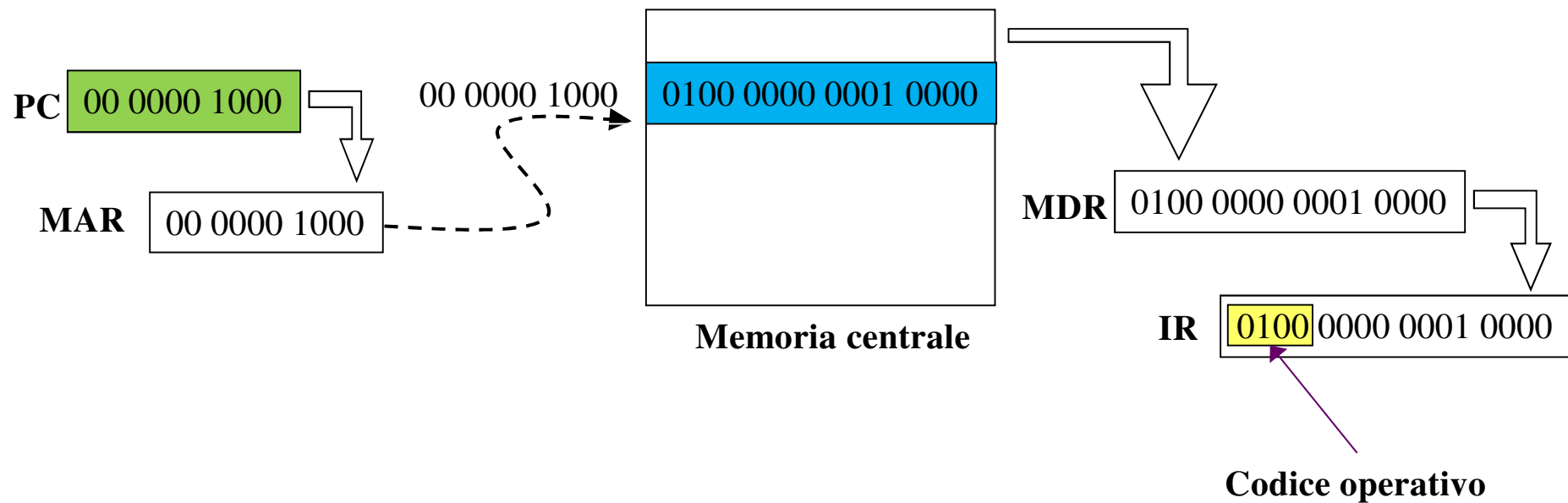
- L'unità di controllo opera in un ciclo infinito:
 1. Prelievo
 2. Preparazione degli operandi
 3. Esecuzione

Nella fase di bootstrap il ciclo viene inizializzato;
viene assegnato un valore iniziale opportuno a PC in modo da avviare l'esecuzione di un programma iniziale in ROM



Fase fetch

- $IR = M[PC]; PC = PC + k$



Fase fetch: sottopassi

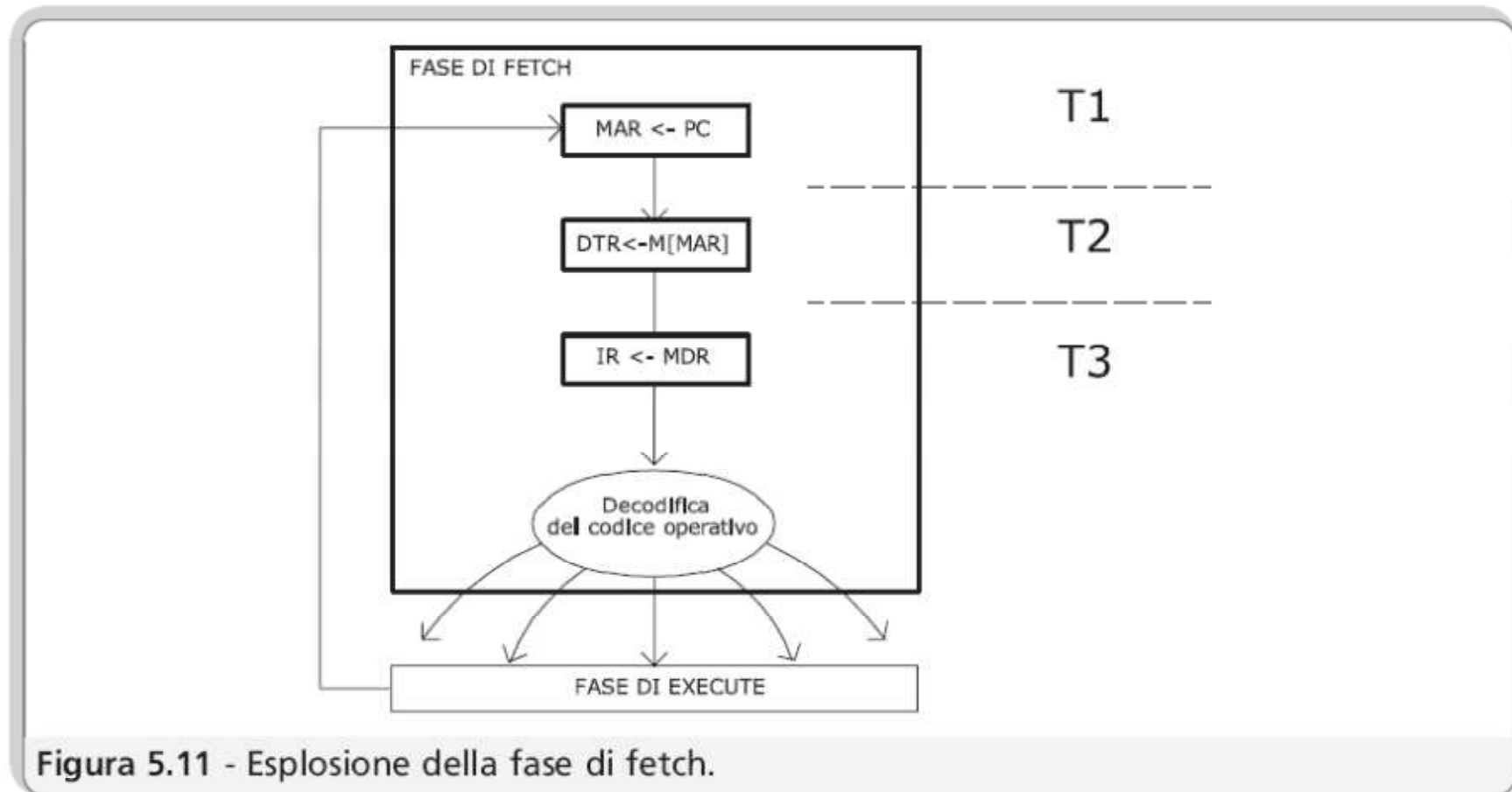


Figura 5.11 - Esplosione della fase di fetch.

Esecuzione sequenziale delle istruzioni

- Alla fine della fase fetch:
 - $PC = PC + k$
 - $k =$ lunghezza istruzioni in byte
- serve a far sì che PC punti all'istruzione posta subito dopo
- Esecuzione delle istruzioni in sequenza così come sono memorizzate
- Per cicli e figure di controllo (if-then, if-then-else, switch) occorrono istruzioni di salto

