



# Sisteme cu microprocesoare

## Cursul 3

---

Structura unui calculator

# Formatul instructiunilor

---

Cod operatie

- Fara operanzi expliciti

Cod operatie

Operand1 - Adresa

- Cu un singur operand

Cod operatie

Operand1 - Adresa

Operand2 - Adresa/Data

- Cu doi operanzi

Cod operatie

Operand1 - Adresa

Operand2 - Adresa/Data

Rezultat - Adresa

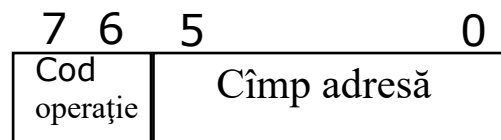
- Cu doi operanzi si rezultat

# Moduri de adresare

---

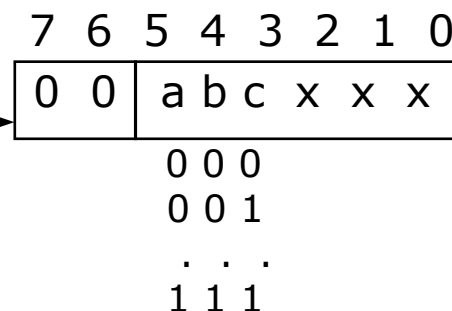
- Cum se determina operandul
  - Adresare imediata – pentru constanta
    - MOV AX, 1234h
  - Adresare directa – pentru variabile simple
    - SUB BX, [100h]
  - Adresare indexata – pentru vectori
    - ADD [SI+500h], AX
  - Adresare bazata – pentru inregistrari
    - AND DX, [BX+200h]
  - Adresare mixta (bazat-indexata) – pentru structuri complexe
    - OR CX, [SI+BX+100h]

# Setul de instructiuni pentru un calculator simplu



Cod  
operație:

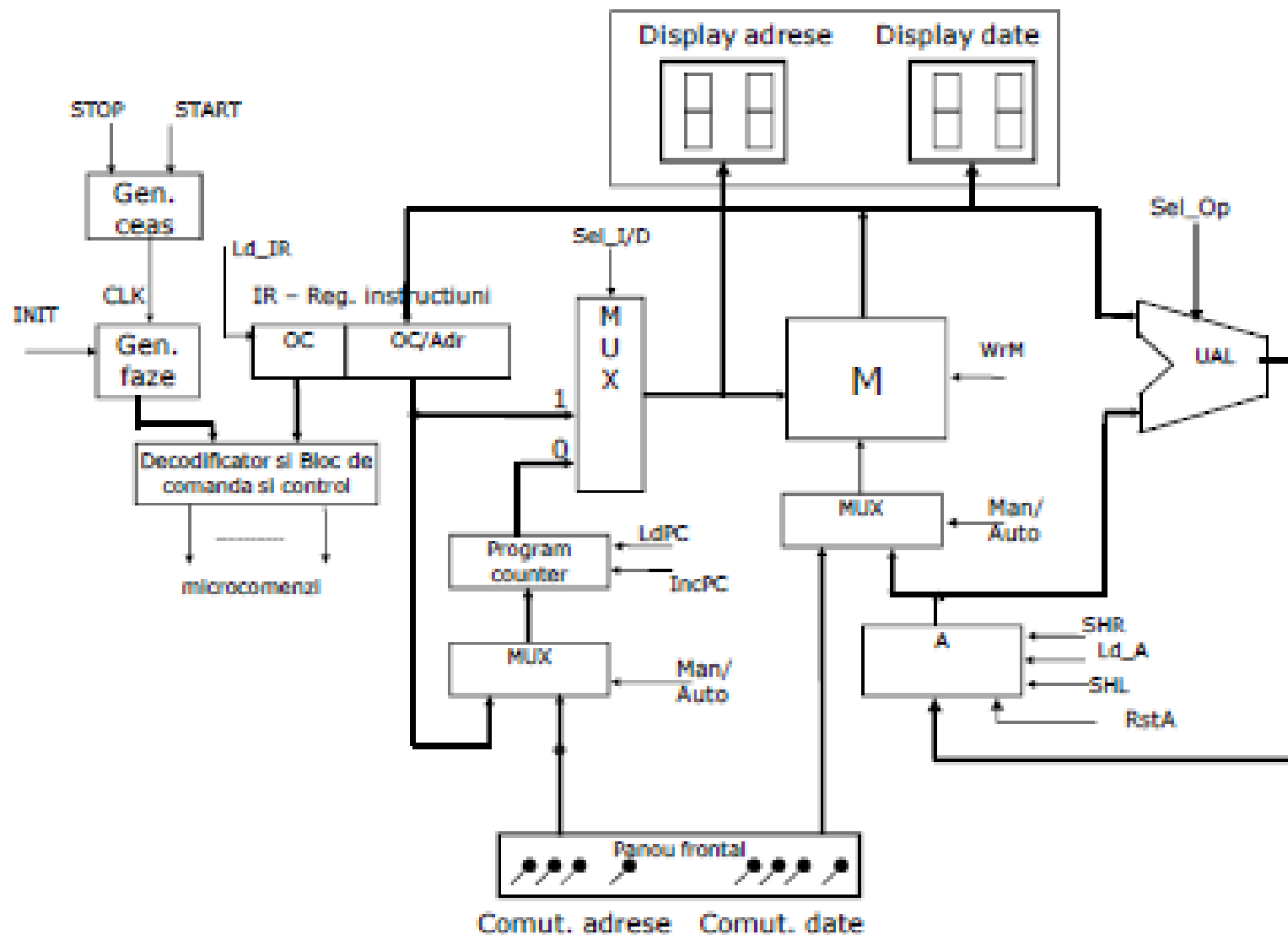
0 0  
0 1  
1 0  
1 1



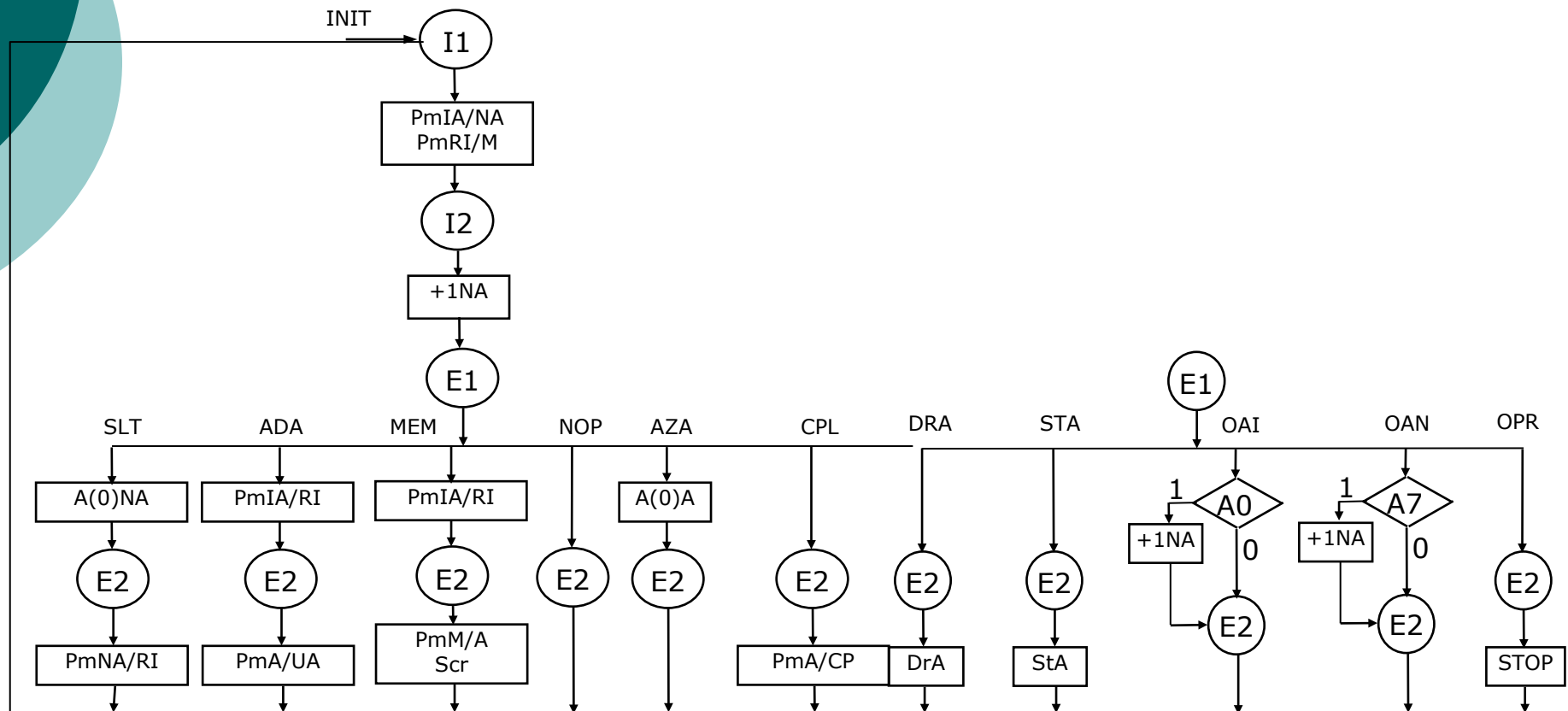
Instr.	Cod instr.	explicatii
SLT	01 adr	NP=adr
ADA	10 adr	A=A+M[adr]
MEM	11 adr	M[adr]=A

NOP	00 000 xxx	-
AZA	00 001 xxx	A=0
CPL	00 010 xxx	A=Not A
DRA	00 011 xxx	A=A>>1
STA	00 100 xxx	A=A<<1
OAI	00 101 xxx	If A <sub>0</sub> =1 NP++
OAN	00 110 xxx	If A <sub>7</sub> =1 NP++
OPR	00 111 xxx	

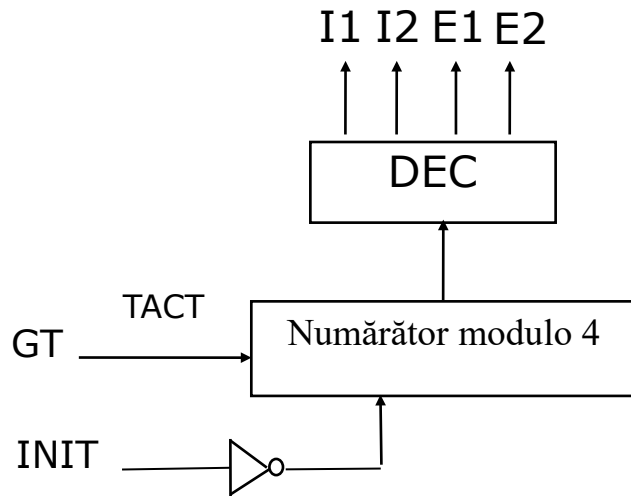
# Schema generala a unui calculator simplu



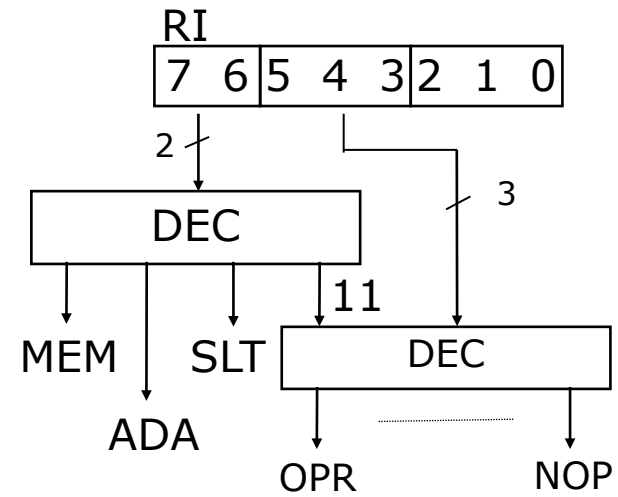
# Descompunerea instructiunilor in faze de executie



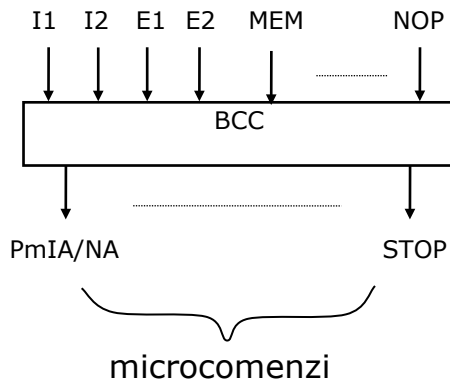
# Generatorul de faze, decodificatorul de instructiuni si blocul de comanda si control



a.

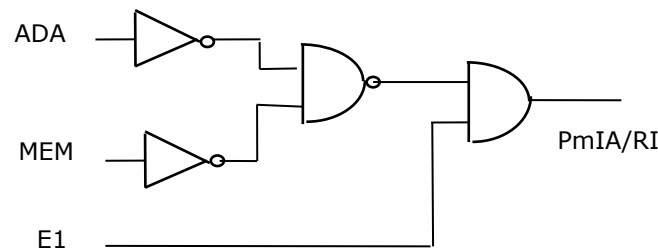


b.



a.

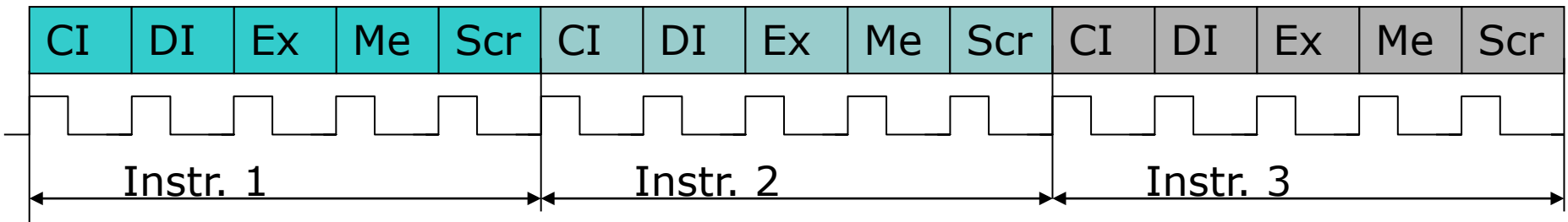
$$PmIA/RI = E1 \cdot (ADA + MEM)$$



b.

# Executia secventiala a instructiunilor (arhitectura scalara)

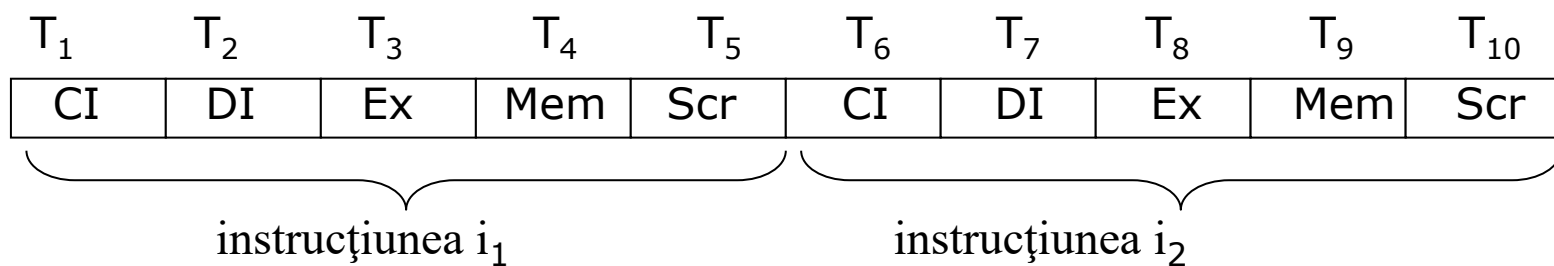
- Fazele de executie ale unei instructiuni:
  1. Citire instructiune (CI)
  2. Decodificare instructiune (ID)
  3. Executie operatie (Ex)
  4. Operatie cu memoria (Mem)
  5. Scriere rezultat (Scr)



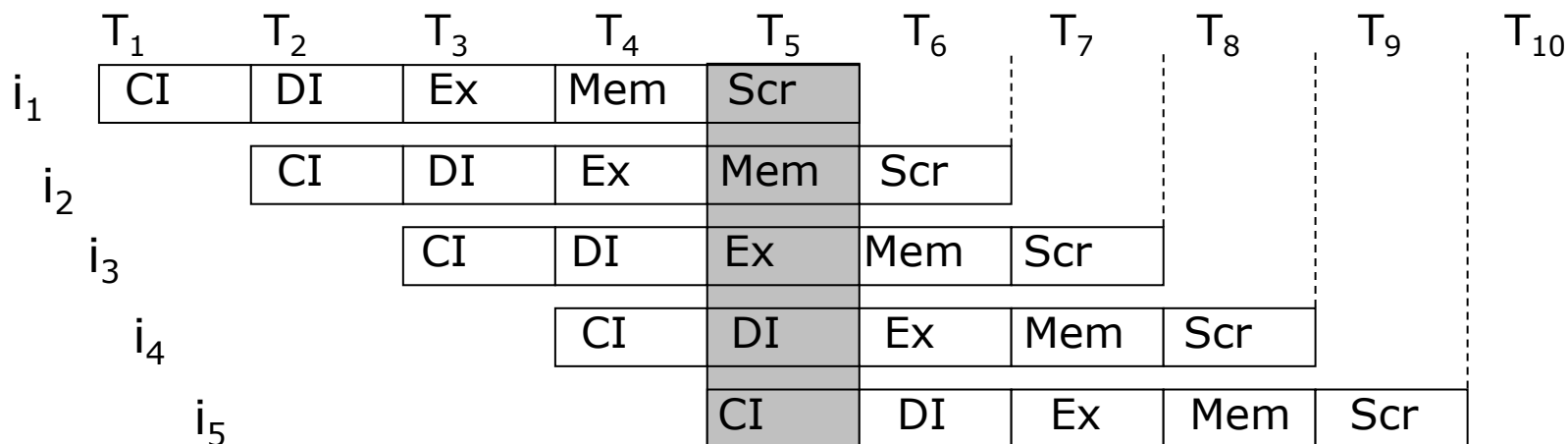
●  $CPI = 5$



# Executia paralela instructiunilor intr-o arhitectura pipeline (banda rulanta)

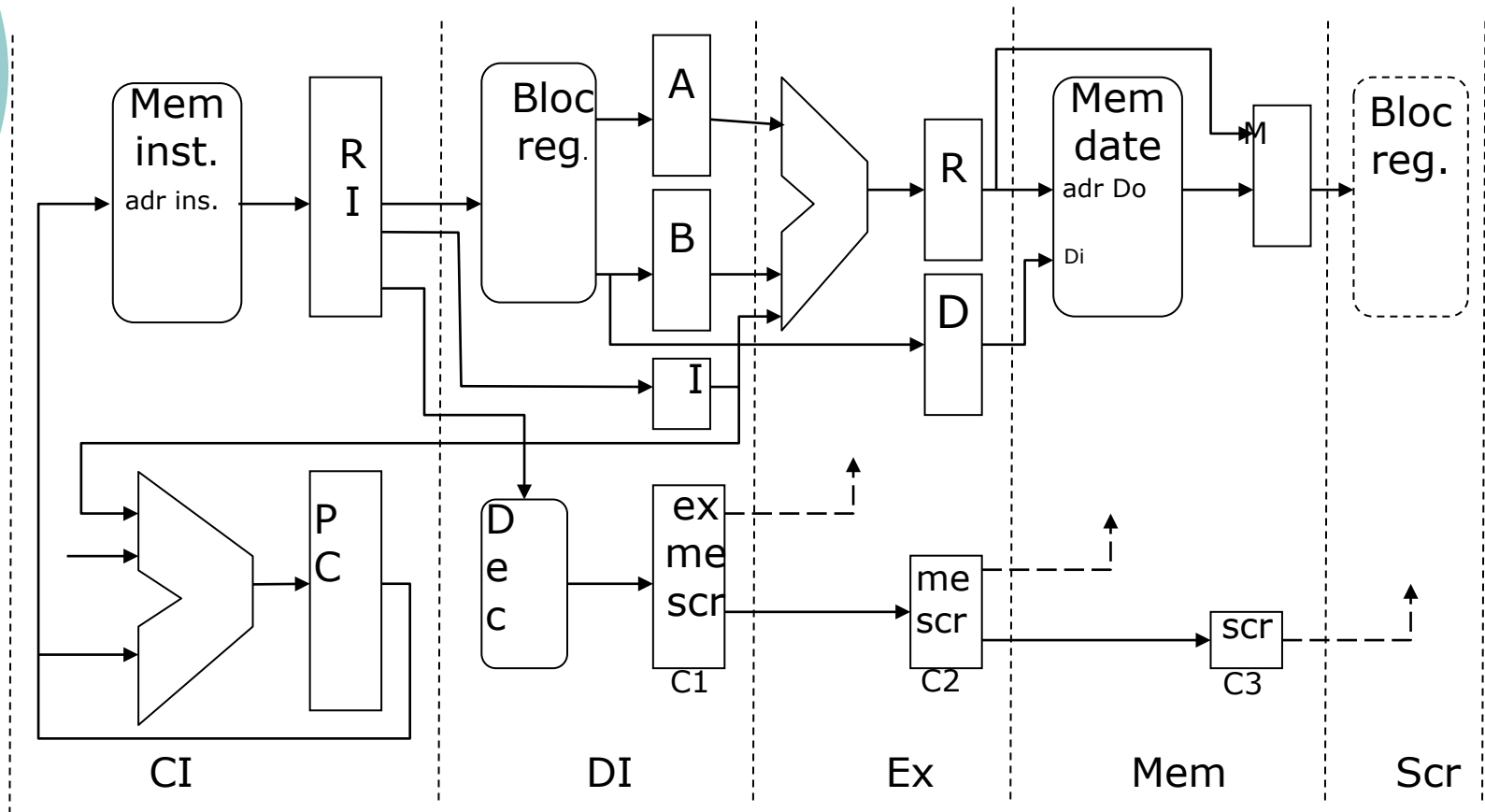


**Execuție secvențială CPI = 5**



**Execuție pipeline CPI=1 (în cazul ideal)**

# Exemplu de arhitectura pipeline

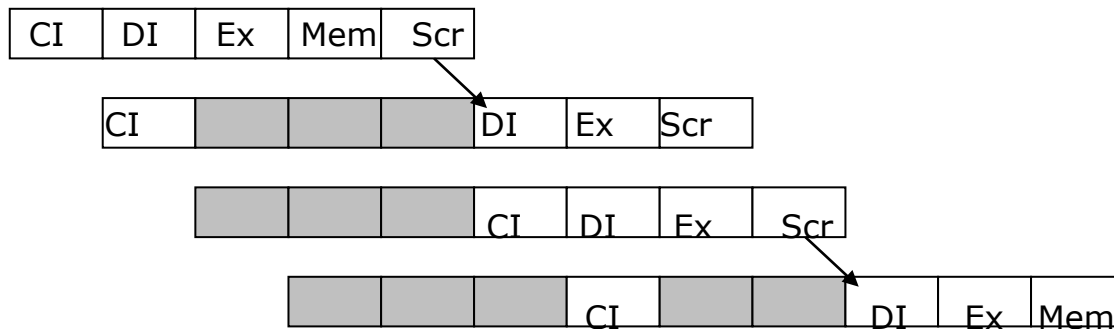


UCP în structură pipeline

# Situatii de hazard in arhitecturile paralele

## ○ Hazardul de date

- Dependenta de date intre instructiuni consecutive
- Solutionare:
  - Stari de asteptare, multiplicarea registrelor, reordonarea instructiunilor



MOV AX, 5

ADD BX, AX

SUB CX, 5

MOV DX, CX

# Situatii de hazard

## ○ Hazardul de control

- Datorat instructiunilor de salt
- Solutionare: predictia salturilor, memorarea salturilor anterioare

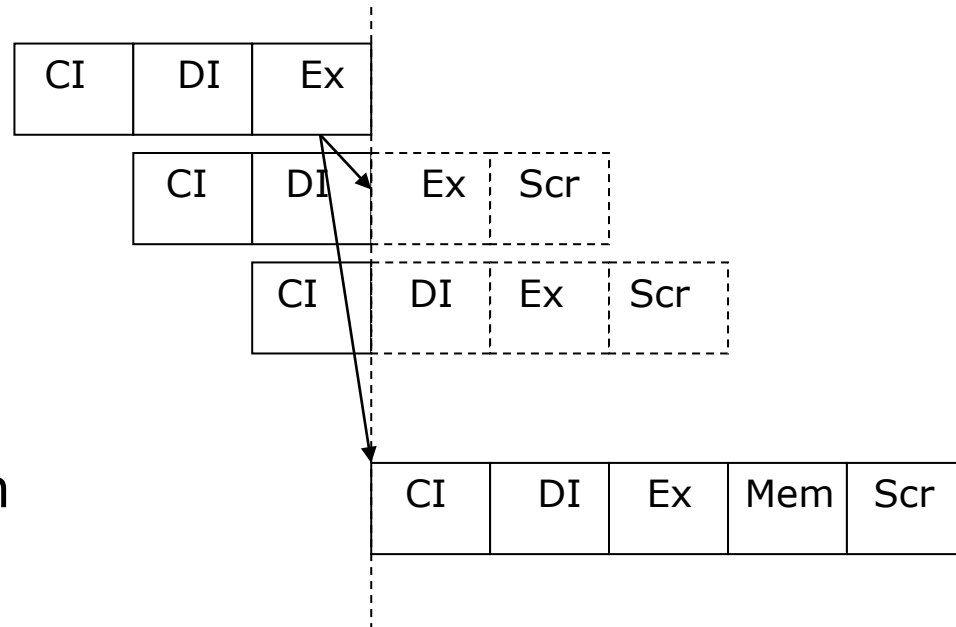
JE et1

ADD AX, BX

SUB CX, DX

.....

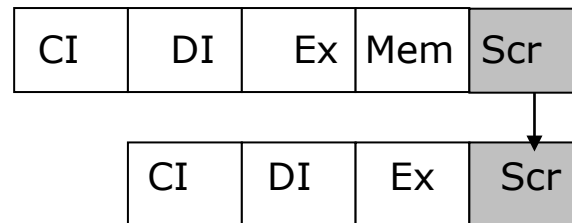
et1: MOV SI, 1234h



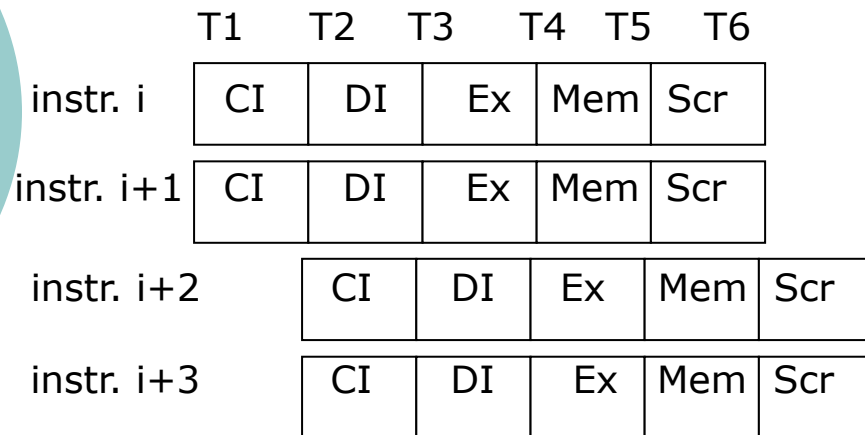
# Situatii de hazard

---

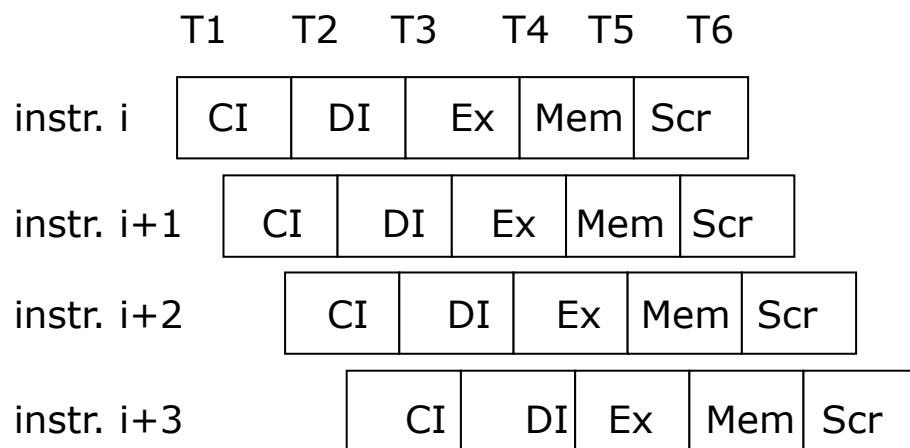
- Hazardul structural
  - Doua faze diferite ale unor instructiuni consecutive solicita aceea si unitate structurala
  - Solutionare: stari de asteptare, reordonarea instructiunilor, duplicarea unitatilor solicitate



# Arhitecturi superscalare și superpipeline



a. arhitectură superscalară  
CPI=1/2



b. arhitectură superpipeline  
CPI=1/2

Figura 6.13 Comparație între arhitectura superscalară și superpipeline

# Planificarea executiei instructiunilor

---

- Obiectiv: reordonarea instructiunilor in scopul evitarii situatiilor de hazard si implicit pentru cresterea performantelor
- Planificarea statica:
  - Reordonarea instructiunilor in faza de compilare
  - Gruparea instructiunilor in super-instructiuni foarte lungi
    - VLIW – Very Long Instruction Word
- Planificarea dinamica:
  - Reordonarea instructiunilor in timpul executiei programului
  - Tehnici folosite: analiza fluxurilor de date, arbori de dependente, predictia salturilor