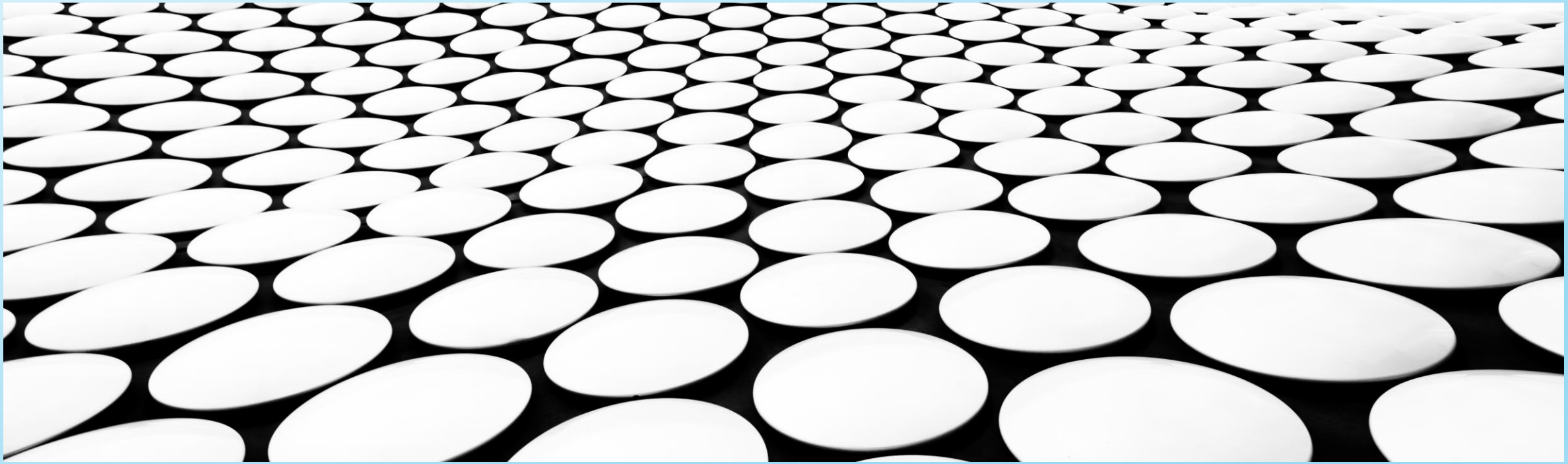
ARHITECTURA SISTEMELOR DE CALCUL

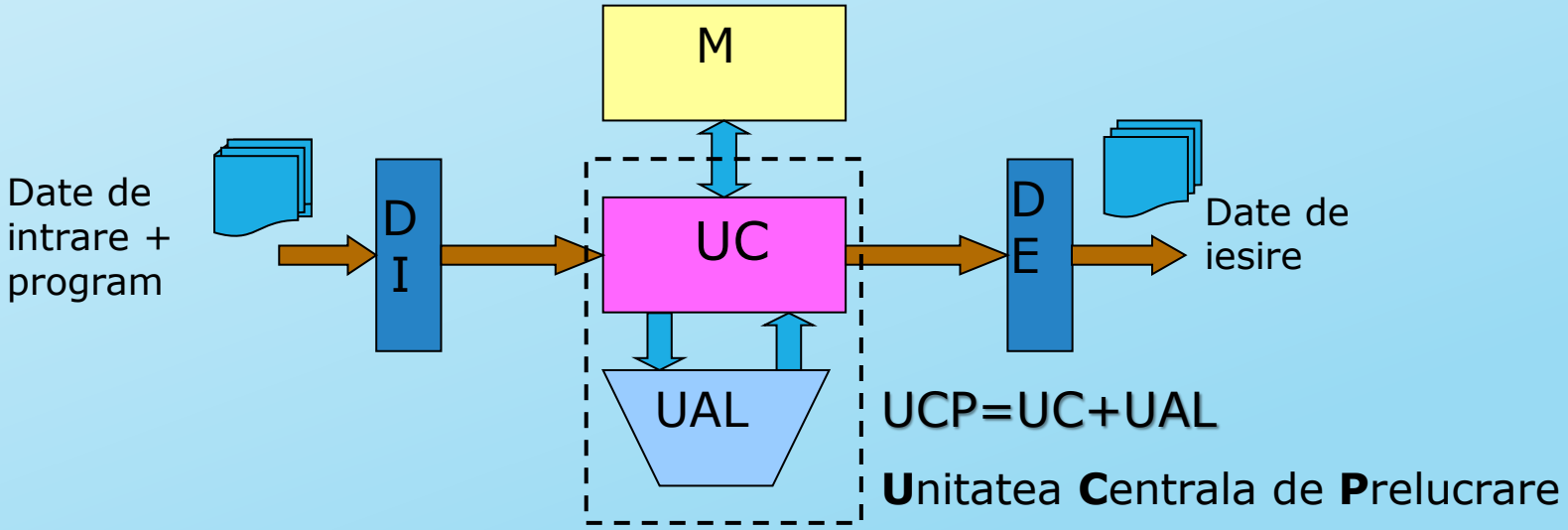
UB, FMI, CTI, ANUL III, 2022-2023



Funcţionarea unui sistem de calcul

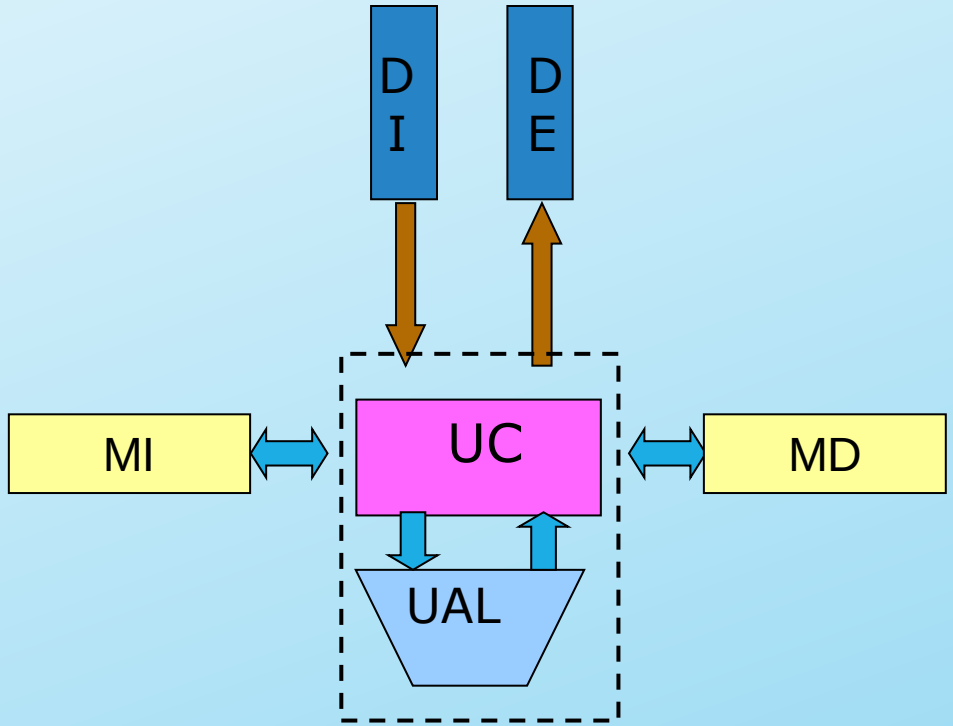
Modelul von Neumann

+ Unitatea de control (UC)
+ Unitatea aritmetico-logica (UAL)
+ Memoria (M)
+ Dispozitiv(e) de intrare (DI)
+ Dispozitiv(e) de iesire (DE)



In engleza CPU (**C**entral **P**rocessing **U**nit)

Modelul Harvard



Memorie pentru instructiuni (MI)
Memorie pentru date (MD)

Componentele unui sistem de calcul

Unitatea centrală de prelucrare(UCP) [CPU]

- Unitatea de Control (UC) [CU]
  - controlează toate componentele, executând instrucțiunile unui program
- Unitatea de calcul Aritmetic si Logic (UAL) [ALU]
  - efectuează calculele aritmetice şi logice

Memoria (UM)

- stocheaza programele în curs de execuție şi datele asociate lor

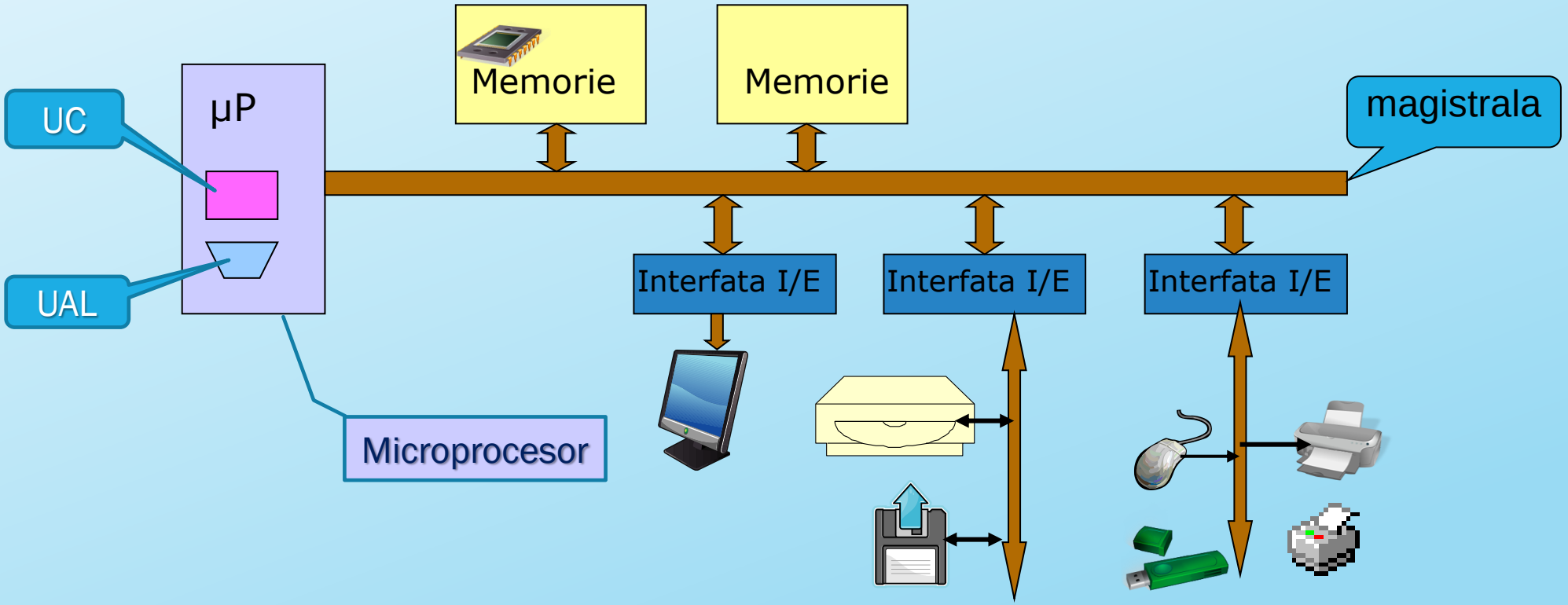
Unitatea de Intrare/Ieşire (U I/E)

Leagă sistemul cu lumea externă prin intermediul unităților periferice:

- ecran,
- tastatură,
- discuri,
- benzi magnetice,
- rețele etc.

Componentele unui sistem de calcul

Sistem de calcul bazat pe un microprocesor



Principiul de funcţionare al unui calculator

- În **UM** există programe, fiecare program având un număr de instrucțiuni, precum si date destinate prelucrarii.
- Execuția unui program înseamnă execuția succesivă a instrucțiunilor din care este alcătuit.
- Fiecare instructiune este executata in mai multe cicluri (etape) succesive.

Ciclurile (etapele) de executie ale unei instructiuni

1. extragerea instrucţiunii
2. identificarea operanzilor
3. transferul operanzilor
4. execuţia propriu-zisă
5. transferul rezultatului

Ciclul ‘extragere instrucţiune’ (instruction fetch).

- UC lanseaza o citire a memoriei la adresa la care se află instrucțiunea. Instrucțiunea are un număr de biți, în funcție de arhitectura calculatorului, de obicei multiplu de 8.
- Instrucțiunea citită este transferata prin magistrală şi depusă într-un registru al UC-ului.

registru = memorie

Ciclul de identificare a operanzilor.

- În această fază trebuie identificate adresele unde se găsesc operanzii. Aceştia se pot găsi în două tipuri de locații:
  - în registrele generale ale UC-ului;
  - la o adresă de memorie.

- La sfârşitul acestui ciclu, în UC trebuie să existe adresele fizice ale operanzilor participanți la instrucțiune.

Ciclul de transfer al operanzilor în UC

- În acest ciclu se aduc operanzii participanți la instrucțiune de la adresele determinate în ciclul anterior. Ei sunt aduşi din registrele generale sau de la adresele de memorie în registrele funcționale.

Ciclul de execuţie propriu-zisă

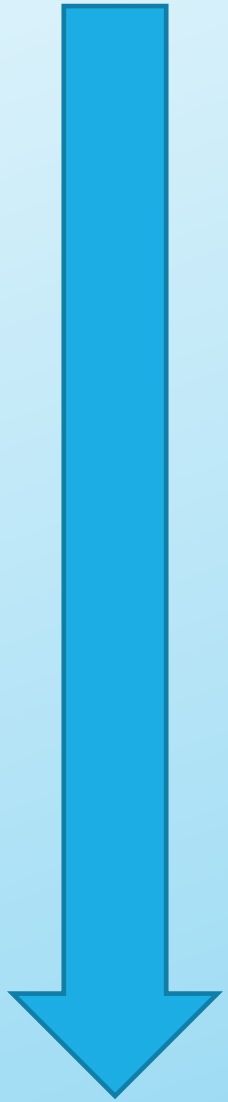
În acest ciclu are loc execuția propriu-zisă a instrucțiunii, dată de codul instrucțiunii

Orice instrucțiune are ca scop final aflarea unui rezultat, care poate fi:

- un **operand** în cazul instrucțiunilor aritmetice (de exemplu suma pentru cod de *adunare*, produsul pentru cod de *înmulțire*) sau:
- poziționarea unor **indicatori** în cazul instrucțiunilor logice
  - de exemplu, în cazul unei instructiuni de comparație între doi operanzi, poziționarea indicatorului z=1 corespunde la identitatea (egalitatea) celor doi operanzi.

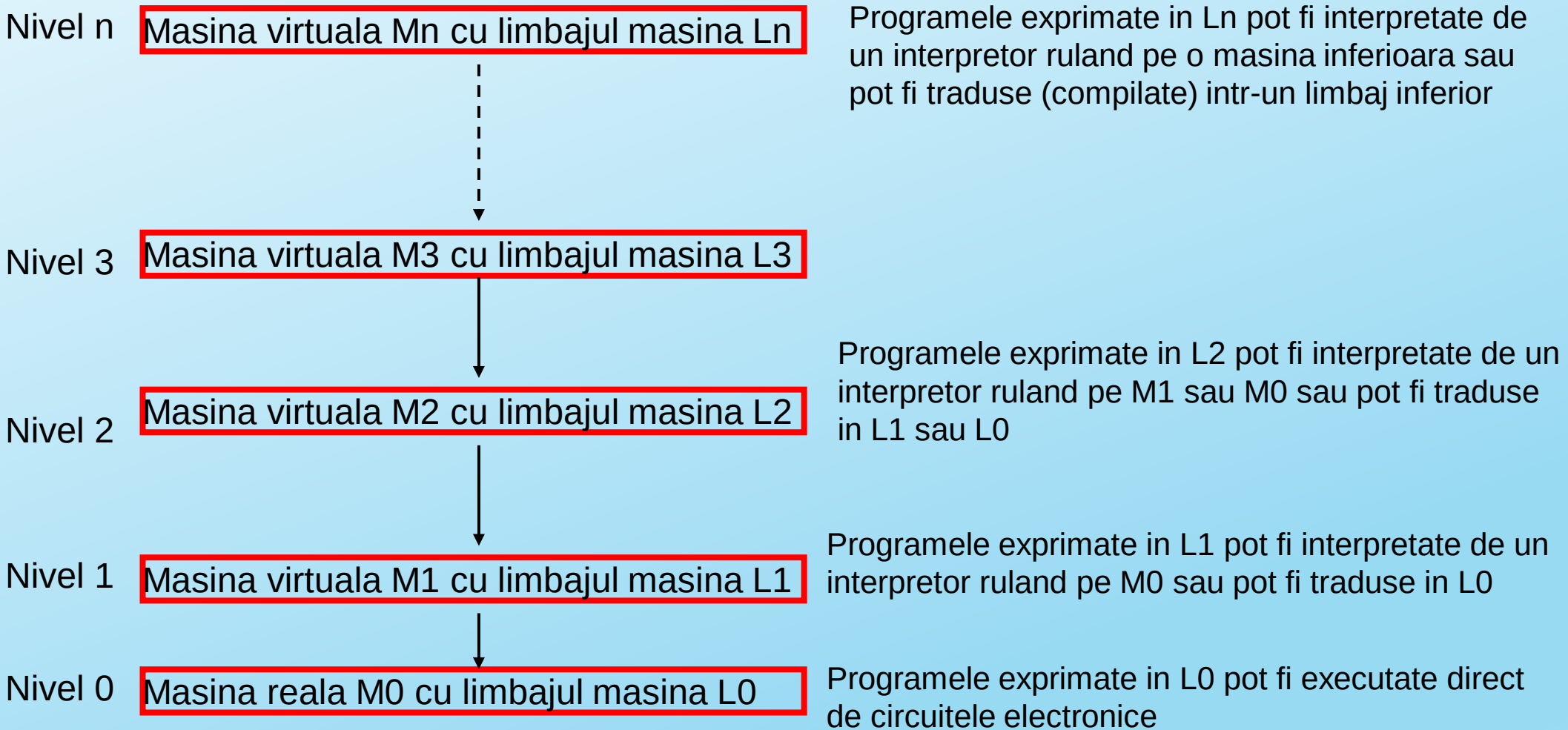
Ciclul de transfer al rezultatului.

- La sfârşitul acestui ciclu, care înseamnă şi sfârşitul executării instrucțiunii,
  - Se calculeaza adresa unde va fi transferat rezultatul
  - Se transfera rezultatul
  - Se calculează adresa de la care va fi adusă instrucțiunea următoare.

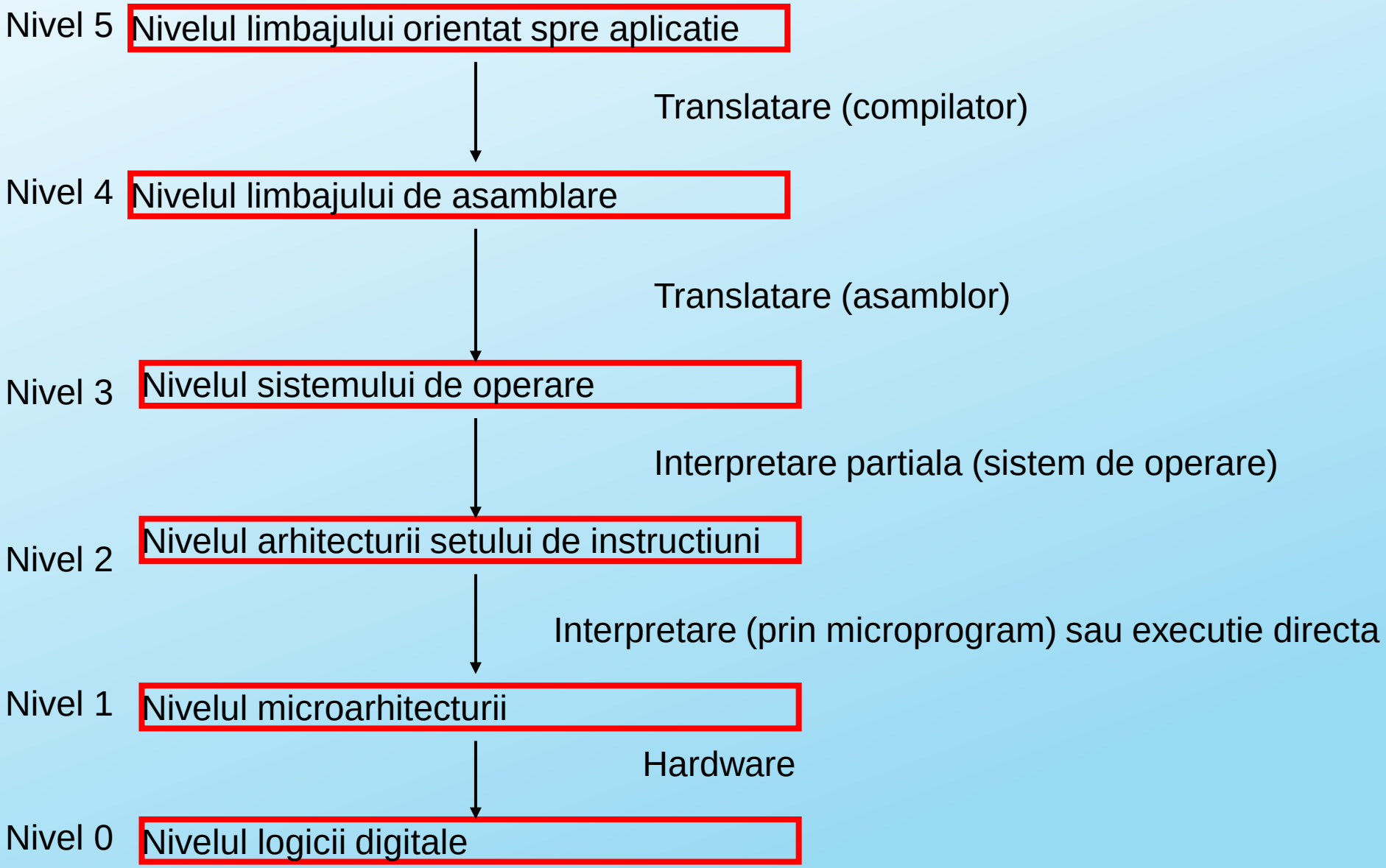


Sistemul de masini virtuale

Exemplul 1



Exemplul 2


Nivel 5
Nivelul limbajului orientat spre aplicatie
Translatare (compilator)
Nivel 4
Nivelul limbajului de asamblare
Translatare (asamblor)
Nivel 3
Nivelul sistemului de operare
Interpretare partiala (sistem de operare)
Nivel 2
Nivelul arhitecturii setului de instructiuni
Interpretare (prin microprogram) sau executie directa
Nivel 1
Nivelul microarhitecturii
Hardware
Nivel 0
Nivelul logicii digitale

Exemplul 3

Nivelul definirii problemei (nivelul 7)

- Se identifica o problema complexa din viata reala si se formuleaza in asa fel incat sa poata fi rezolvata cu calculatorul (abstractizare/modelare)
- Se opereaza simplificari (se neglijeaza anumite detalii)
- Se folosesc modele matematice

Nivelul algoritmului (nivelul 6)

- Se defineste o procedura in pasi succesivi
- Pasii trebuie sa fie bine definiti pentru a putea fi executati fara ambiguitate de o masina (virtuala)
- Dezvoltarea algoritmului este un proces creativ!
- Este necesar un **numar finit de pasi** pentru rezolvarea problemei.

Nivelul limbajului de nivel inalt (nivelul 5)

- Ex: C/C++, Java, Python, FORTRAN, LISP, Ada, etc
- Utilizate de programatorii de aplicatie
- Compilatoarele efectueaza si alte sarcini in afara translatiei.

Nivelul limbajului de asamblare (nivelul 4)

- Cuprinde instructiuni mai primitive (elementare/simple)
- Foloseste o “versiune englezeasca” a limbajului masina

Nivelul arhitecturii setului de instructiuni ISA (nivelul 3)

- Este interfata dintre hardware si software de nivel coborat
  - Avantaje: accepta diverse implementari pentru aceeasi arhitectura
  - Dezavantaje: in unele situatii frânează inovarea
- Exemple de arhitecturi:
  - IA-32(i386), PowerPC, MIPS, SPARC, ARM

- Este nivelul cel mai coborat ce poate fi “vazut” de programator.

Instructiunile

- Reprezinta limbajul masinii
- Sunt specifice platformei
- Se foloseşte un set limitat de comenzi in limbaj masina (ce pot fi intelese de hardware):
  - ex,.: ADD, LOAD, STORE, RET

Nivelul microarhitecturii (nivelul 1)

- Interpretarea este realizata de Unitatea de Control
- Implementarea este realizata folosind resurse tehnice extrem de sofisticate:
  - In Pentium 4 este implementata ISA x86
  - In Motorola G4 este implementata ISA PowerPC
- Implementarea este optimizata in raport cu obiective de tip cost/performanta.

- Acest nivel este impenetrabil pentru programator

Nivelul Logic-Design (nivelul 0)

- (‘nivelul electronicii digitale’)
  - Porti logice
  - Multiplexoare, decodoare, PLA (programmable logic array)

Nivelul de dispozitiv

- Tranzistori, fire de legatura, etc
  - Implementeaza portile logice digitale
- La acest nivel masina se comporta mai de graba analogic dacat digital.
- Exemple de tehnologii:
  - Si CMOS, Si bipolar, GaAs