ARHITECTURA SISTEMELOR DE CALCUL - CURS 0x06

ARHITECTURA CALCULATOARELOR MODERNE

Cristian Rusu

DATA TRECUTĂ

- logică secvențială, exemple
- înmulțirea numerelor întregi binare
- împărțirea numerelor întregi binare
- reprezentarea numerelor în virgulă mobilă
- operații cu numerele în virgulă mobilă

CUPRINS

- **arhitectura de bază a calculatoarelor**

- **câteva detalii despre fiecare componentă hardware …**

STRUCTURA CURSULUI - UNDE SUNTEM

- **circuite digitale**
  - teoria informației și abstractizarea digitală
  - funcții și circuite logice

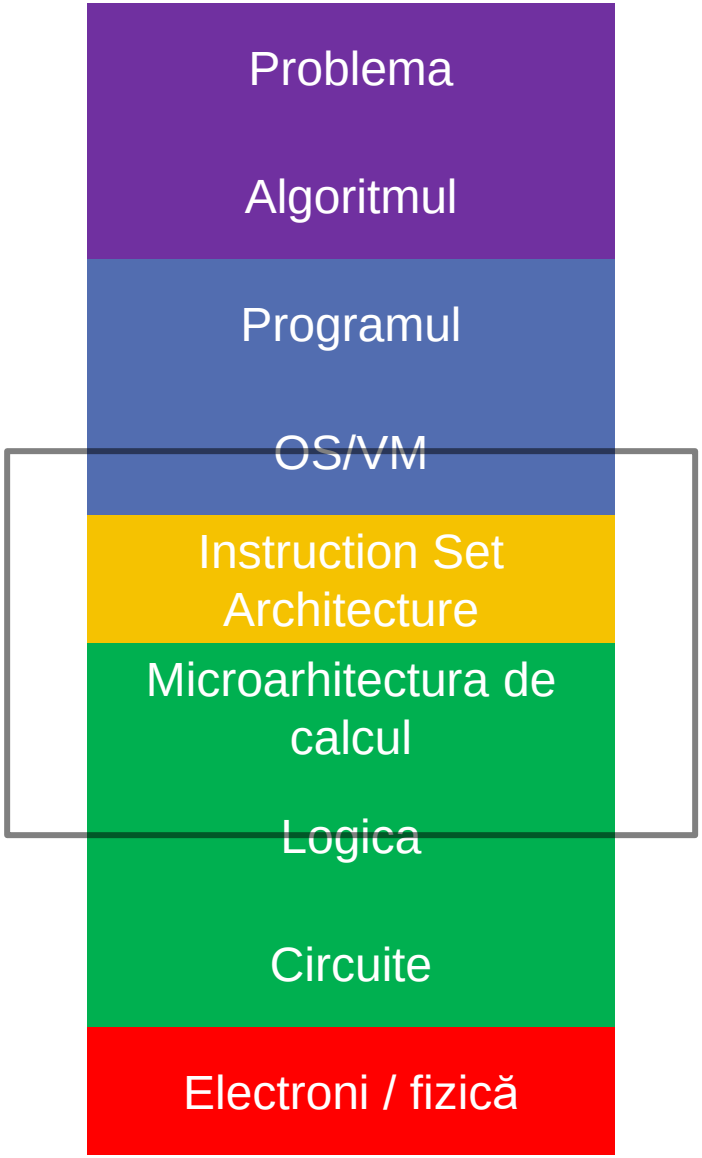
- **arhitecturi de calcul**
  - seturi de instrucțiuni
  - limbajul assembly
  - compilatoare
  - pipelining
  - ierarhia memoriei

- **organizarea calculatoarelor**
  - unitatea de procesare centrală
  - performanța calculatoarelor
  - dispozitive periferice și întreruperi
  - calcul paralel

.

STRUCTURA CURSULUI - UNDE SUNTEM

- **“The purpose of computing is insight, not numbers.” (Richard Hamming)**



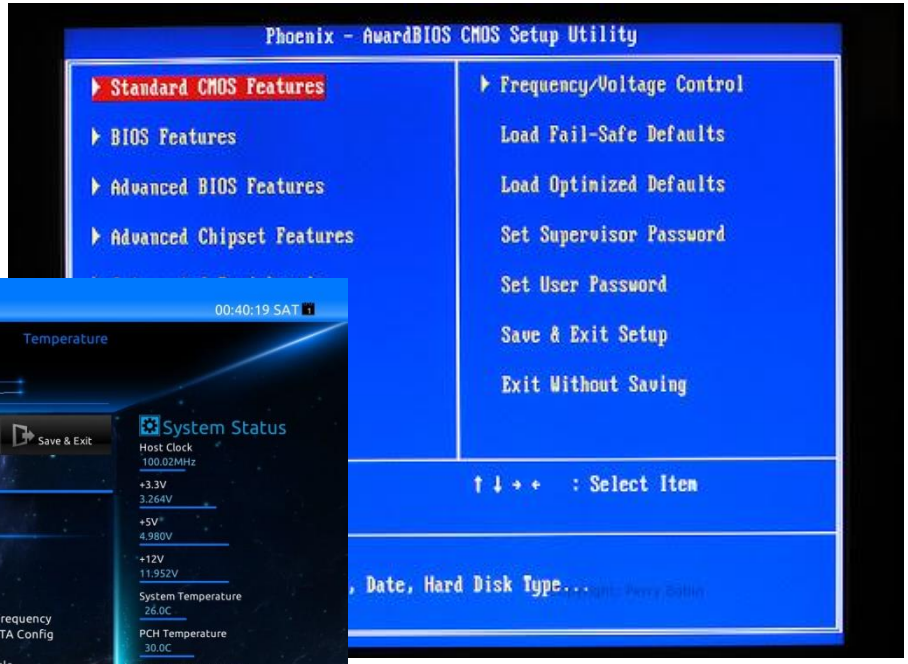
OPERAȚII DE BAZĂ CU SISTEME DE CALCUL

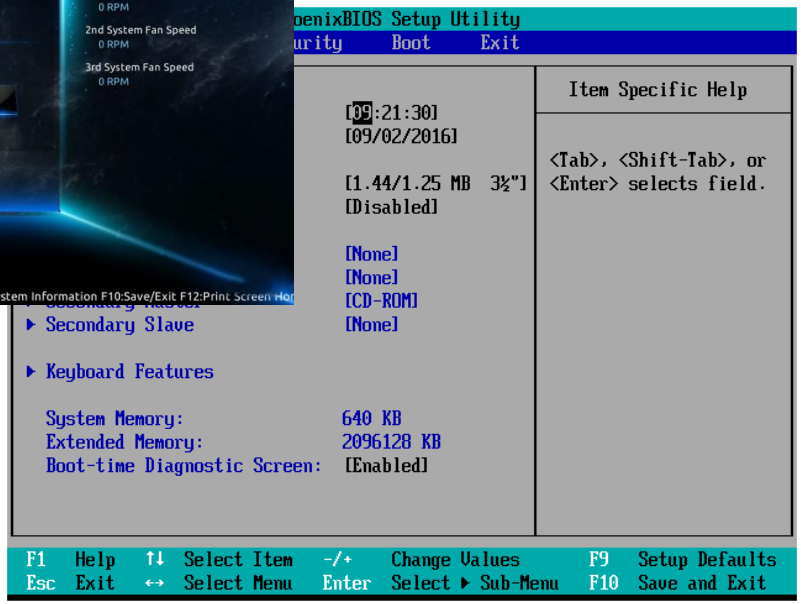
- **pornirea sistemului**
  - în general, un buton de “power on” / “power off” cum funcționează un astfel de buton? atât la pornire cât și la oprire?
  - realizează alimentarea cu electricitate a componentelor
  - **CPU** este activat
  - CPU caută/pornește **BIOS** (Basic Input Output System)
    - testează componentele hardware (RAM, I/O, HD, etc.)
    - BIOS este scris în **ROM** (Read Only Memory) pe placa de bază
    - este scris într-un tip de memorie nevolatilă
    - pentru execuție, BIOS-ul este încărcat în RAM
    - BIOS știe cât e ceasul (**CMOS Real-Time Clock**) și hardware-ul
    - îl accesați automat când porniți calculatorul, fie cu F2 (în general)
  - CPU/BIOS pornesc Boot Code (caută sistemul de operare)
    - sistemul de operare este în general pe HD (poate fi și pe CD, stick)
    - sistemul de operare este încărcat în RAM pentru execuție

https://wiki.osdev.org/CMOS

OPERAȚII DE BAZĂ CU SISTEME DE CALCUL

- **BIOS/UEFI**

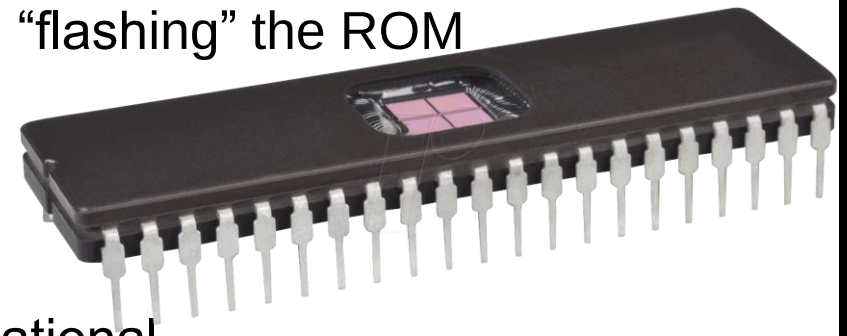




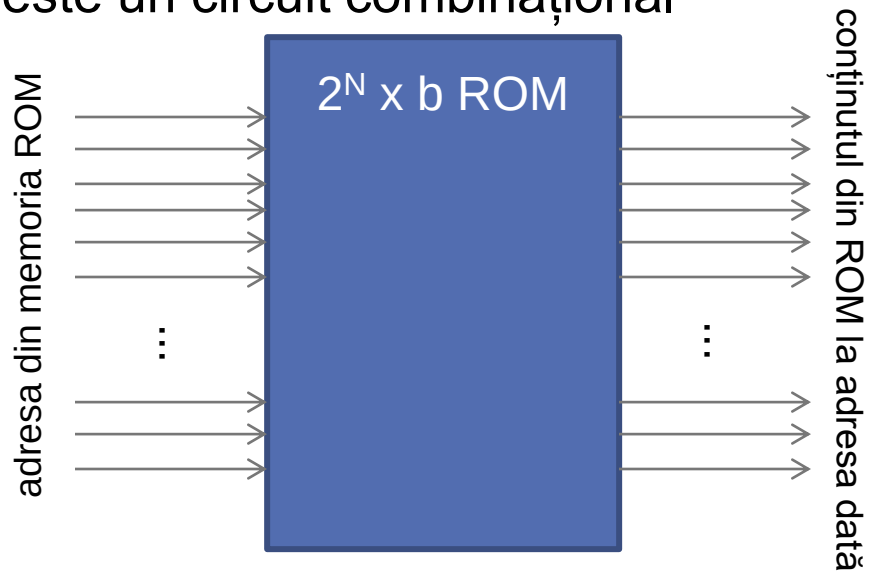
https://en.wikipedia.org/wiki/BIOS
https://en.wikipedia.org/wiki/UEFI

OPERAȚII DE BAZĂ CU SISTEME DE CALCUL

- **BIOS**
  - este scris în ROM (Read Only Memory)
  - câteodată în Programmable ROM / Erasable Programmable ROM / Electrically Erasable Programmable ROM
  - să scriem în ROM: “burning” sau “flashing” the ROM
  - este un tip de *firmware*



  - în general este un circuit combinațional



https://www.reichelt.com/de/en/c-mos-uv-eprom-c-dil-42-2mx8-1mx16-100-ns-27c160-100-p40037.html

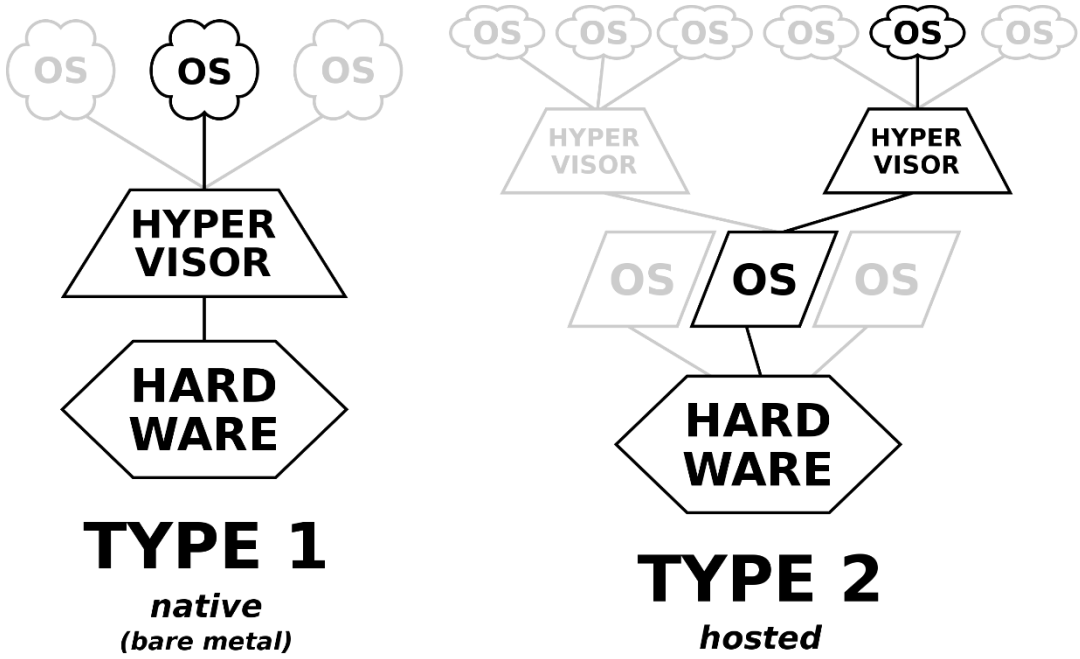
OPERAȚII DE BAZĂ CU SISTEME DE CALCUL

- **mai multe sisteme de operare pe același sistem de calcul**

https://www.gnu.org/software/grub/

OPERAȚII DE BAZĂ CU SISTEME DE CALCUL

- **OS-ul preia controlul de la BIOS**
  - din acest moment doar OS-ul are acces direct la periferice
  - accesul este realizat prin *drivere*
  - virtualizare (hardware)
  - emulare



  - containere (dockere)

https://en.wikipedia.org/wiki/Hypervisor

OPERAȚII DE BAZĂ CU SISTEME DE CALCUL

- **OS-ul preia controlul de la BIOS**
  - din acest moment doar OS-ul are acces direct la periferice
  - accesul este realizat prin *drivere*
  - OS-ul oferă o imagine abstractizată a memoriei pentru fiecare proces pornit

  - din momentul în care OS-ul pornește, sistemul de calcul intră în ciclul obișnuit de procesare (secvența de boot s-a terminat)

ARHITECTURA DE BAZĂ

- **un sistem de calcul trebuie să fie capabil:**
  - să calculeze
    - să execute instrucțiuni
  - să comunice
    - să transfere biți între componente electronice
  - să stocheze
    - date care să fie folosite de instrucțiuni
    - instrucțiuni pentru execuție

ARHITECTURA DE BAZĂ

Periferice Intrare

Periferice Ieșire

BUS

Unitatea Centrală de Procesare (CPU)

Unitatea de Control

Unitatea Artimetică/Logică

Regiștri

Memoria Principală

Memoria de Stocare

ARHITECTURA DE BAZĂ

Periferice Intrare

Periferice Ieșire

BUS

Unitatea Centrală de Procesare (CPU)

Unitatea de Control

Unitatea Artimetică/Logică

Regiștri

Memoria Principală

DMA

Memoria de Stocare

https://en.wikipedia.org/wiki/Direct_memory_access

ARHITECTURA DE BAZĂ

Periferice Intrare

controler

Periferice Ieșire

controler

BUS

Unitatea Centrală de Procesare (CPU)

Unitatea de Control

Unitatea Artimetică/Logică

Regiștri

Memoria Principală

controler

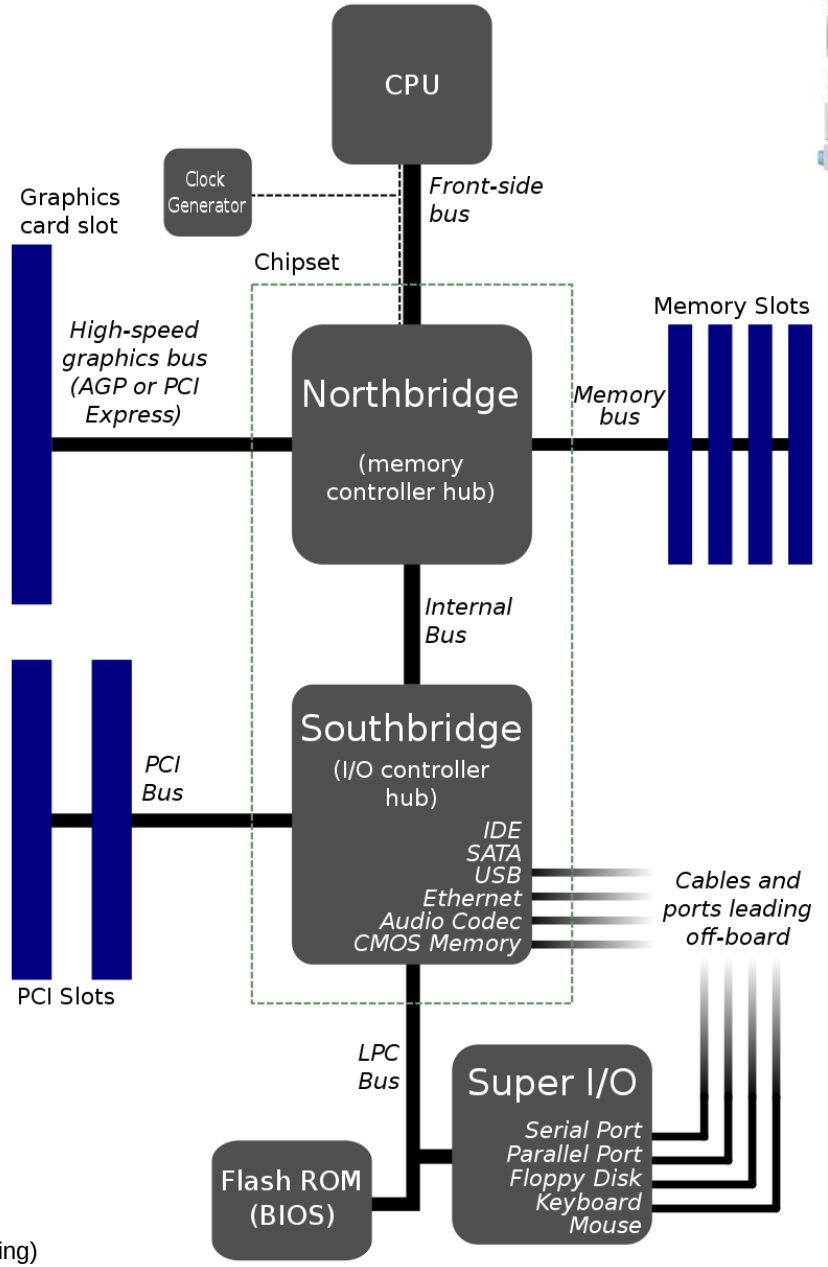
DMA

Memoria de Stocare

controler

https://en.wikipedia.org/wiki/Memory_controller

ARHITECTURA DE BAZĂ

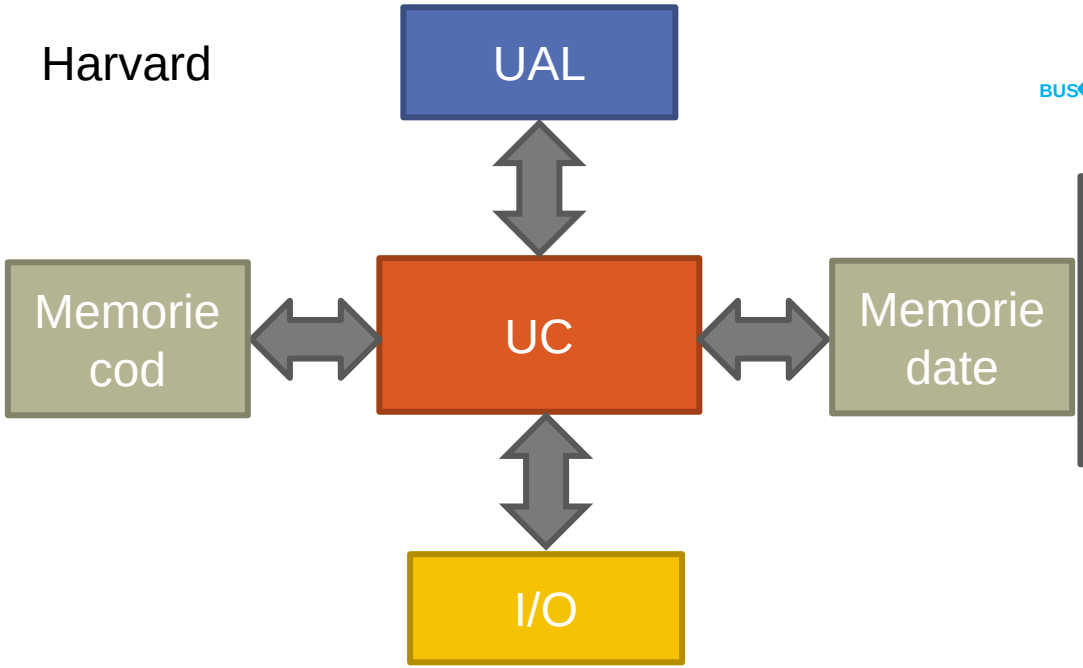


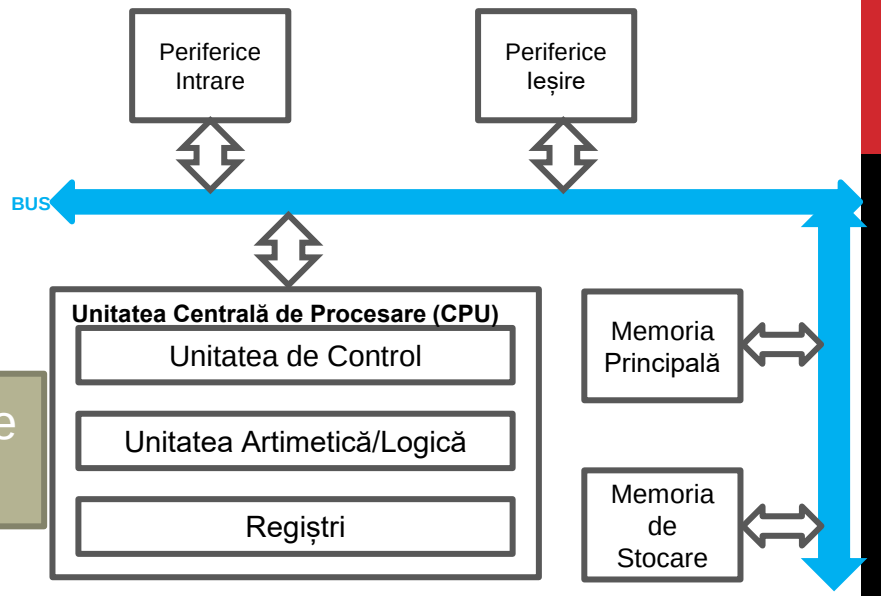
https://en.wikipedia.org/wiki/Northbridge_(computing)

ARHITECTURA DE BAZĂ

- **Tipul arhitecturii de calcul**

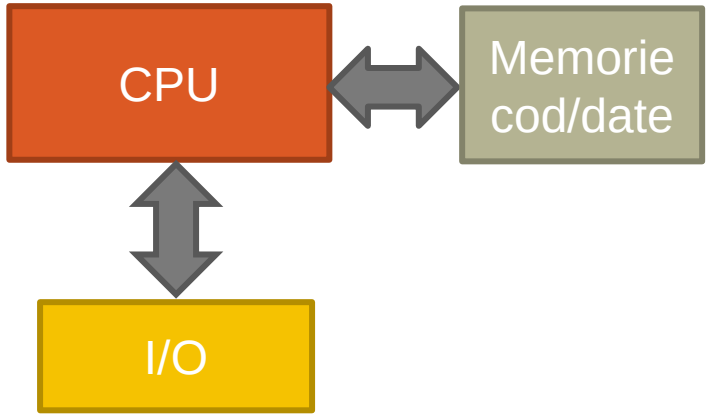
Harvard





---

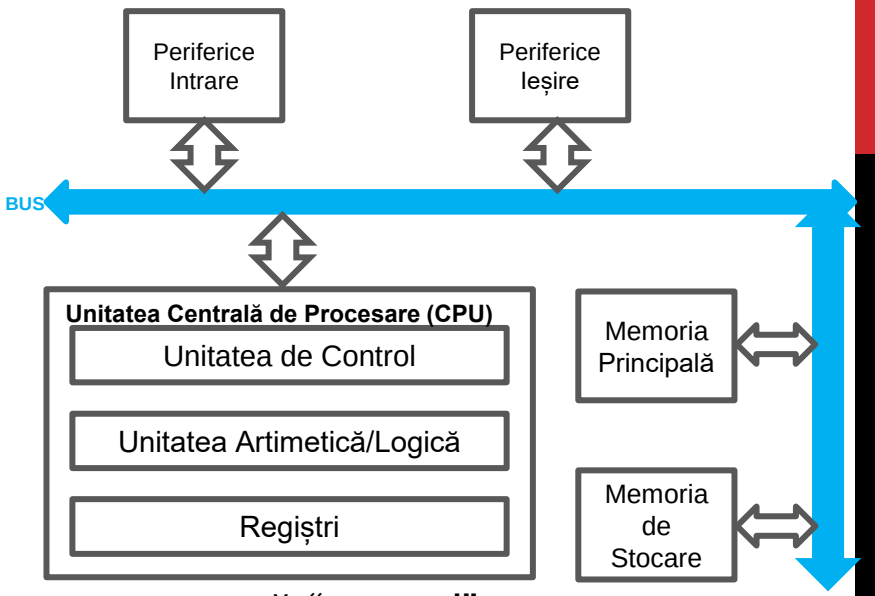
von Neumann



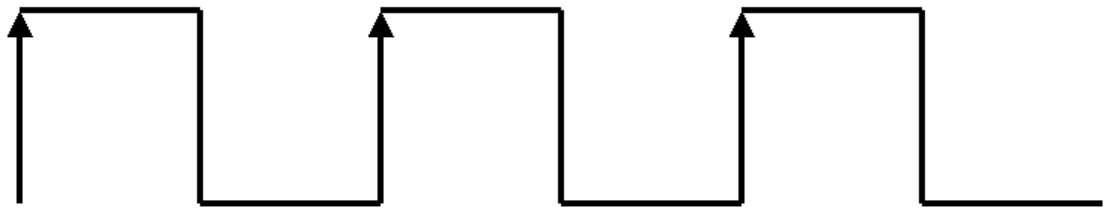
calculatoarele recente încep să nu mai fie von Neumann

ARHITECTURA DE BAZĂ

- **Unitatea Centrală de Procesare**
  - a.k.a. CPU
  - este “creierul” unității de calcul
  - execută instrucțiuni



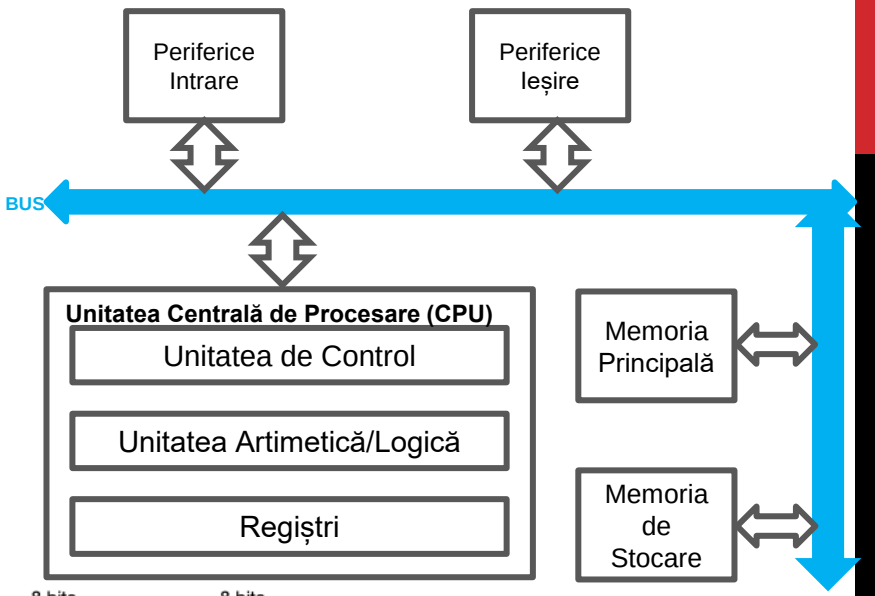
  - 5 componente principale:
    - **Clock**
      - este un circuit special care generează “ceasul”

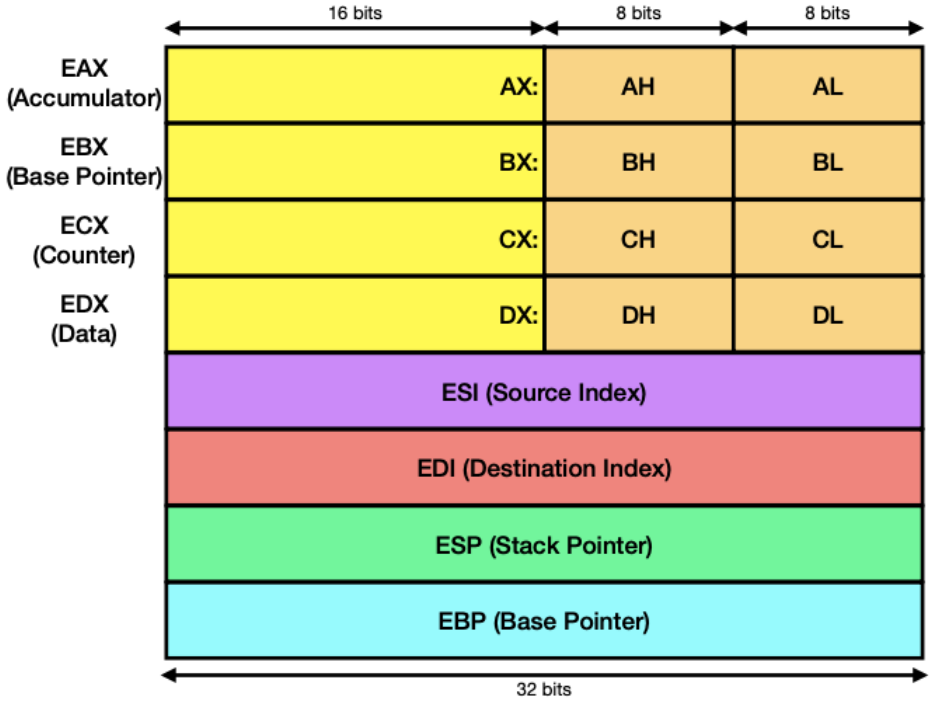


      - este frecvența la care operează (calcule și sincronizarea componentelor secvențiale) CPU-ul
      - cu cât este mai mare frecvența, cu atât mai bine (în general)
      - se măsoară în MHz sau GHz

ARHITECTURA DE BAZĂ

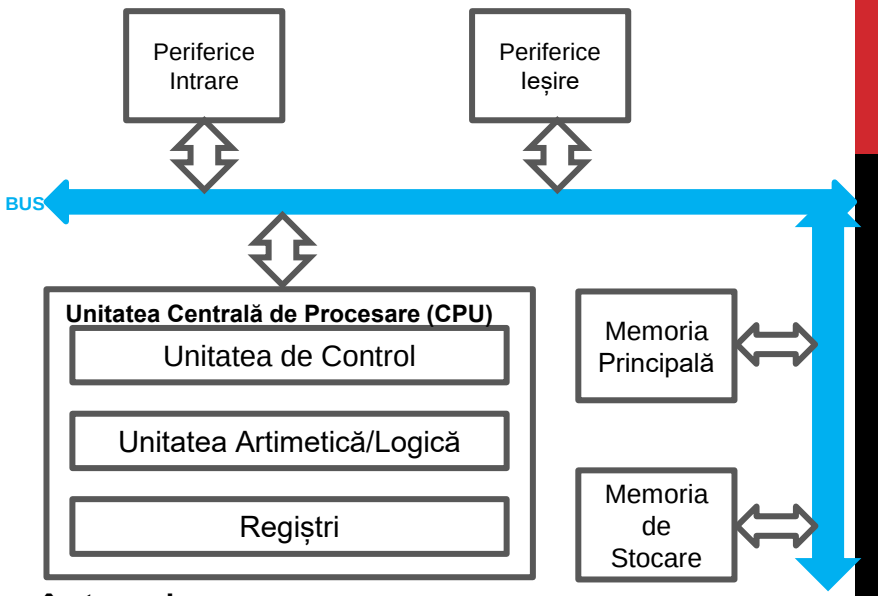
- **Unitatea Centrală de Procesare**
  - a.k.a. CPU
  - este “creierul” unității de calcul
  - execută instrucțiuni
  - 5 componente principale:
    - **regiștri (“memoria”)**



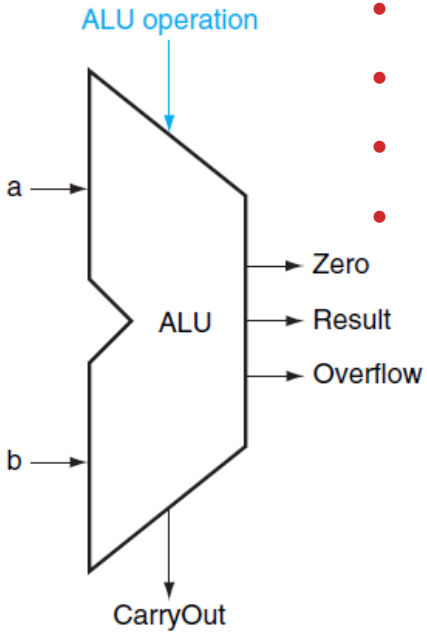


ARHITECTURA DE BAZĂ

- **Unitatea Centrală de Procesare**
  - a.k.a. CPU
  - este “creierul” unității de calcul
  - execută instrucțiuni



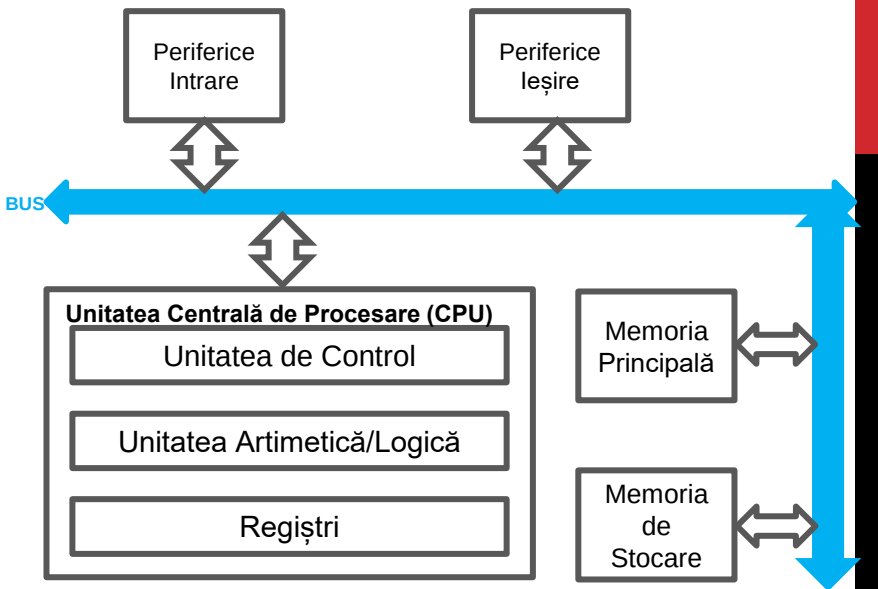
  - 5 componente principale:
    - **UAL (“operații”)**
      - operații aritmetice cu întregi
      - operații logice
      - operații aritmetice cu numere în formatul floating point
      - operații speciale: sqrt, exp, trig



după orice operație, știm “gratuit” dacă rezultatul a fost sau nu zero – este folositor?

ARHITECTURA DE BAZĂ

- **Unitatea Centrală de Procesare**
  - a.k.a. CPU
  - este “creierul” unității de calcul
  - execută instrucțiuni

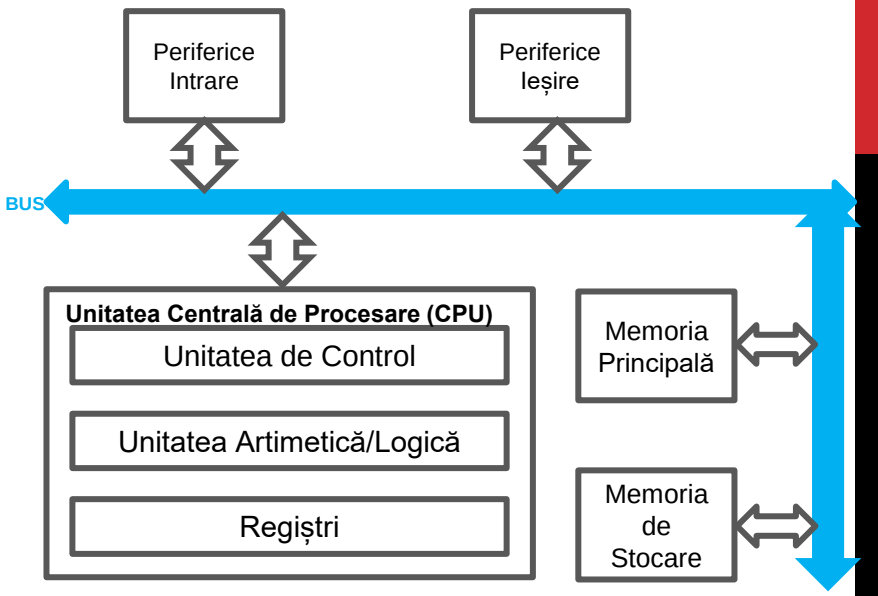


  - 5 componente principale:
    - **BUS**
      - CPU are nevoie de șiruri de biți din memoria principală sau cea de stocare
      - CPU are nevoie să scrie înapoi în memorie rezultate
      - CPU coordonează perifericele

.

ARHITECTURA DE BAZĂ

- **Unitatea Centrală de Procesare**
  - a.k.a. CPU
  - este “creierul” unității de calcul
  - execută instrucțiuni

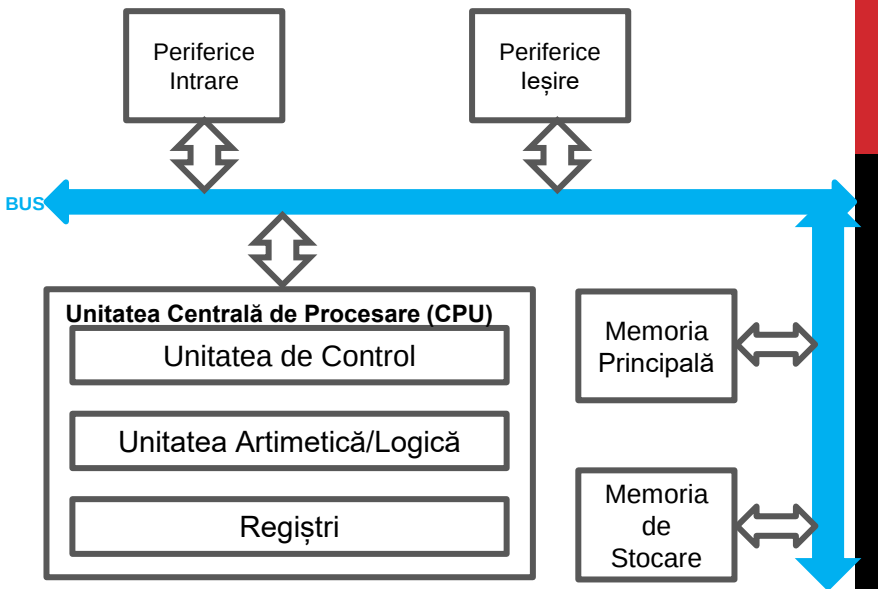


  - 5 componente principale:
    - **UC (“instrucțiunile”)**
      - fetch
        - citim din memorie codul care trebuie executat
        - de unde din memorie? Instruction Pointer ne spune
      - decode
        - circuitul “Instruction Decoder” analizează biții citiți din memorie ca să “înțeleagă” ce să facă cu ei
      - execute
        - execută instrucțiunea decodată
        - poate duce la schimbarea IP sau la transmiterea ceva pe BUS către memorie
      - calculează următorul IP

.

ARHITECTURA DE BAZĂ

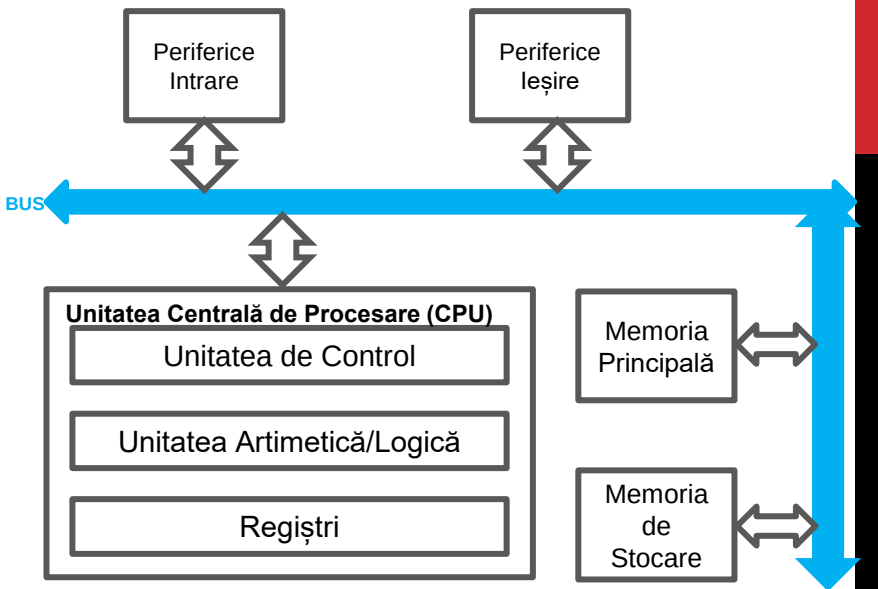
- **Unitatea Centrală de Procesare**
  - a.k.a. CPU
  - este “creierul” unității de calcul
  - execută instrucțiuni



  - 5 componente principale:
    - **UC (“instrucțiunile”)**
      - fetch
        - IP = 10011 (locația în memorie de unde să citim biții)
        - după citire, IP este actualizat
      - decode
        - s-a citit “11000110” care este decodat în
          - opcode = 110, operand1 = 00 operand2 = 110
          - de exemplu: 110 = “adună valoarea imediată A la registrul R”, R = 00 este EAX (prin convenție), A = 110 (adică 6)
      - execute
          - trimite EAX ← EAX + 6 la UAL
          - citește rezultatul din UAL și pune-l în registrul EAX

ARHITECTURA DE BAZĂ

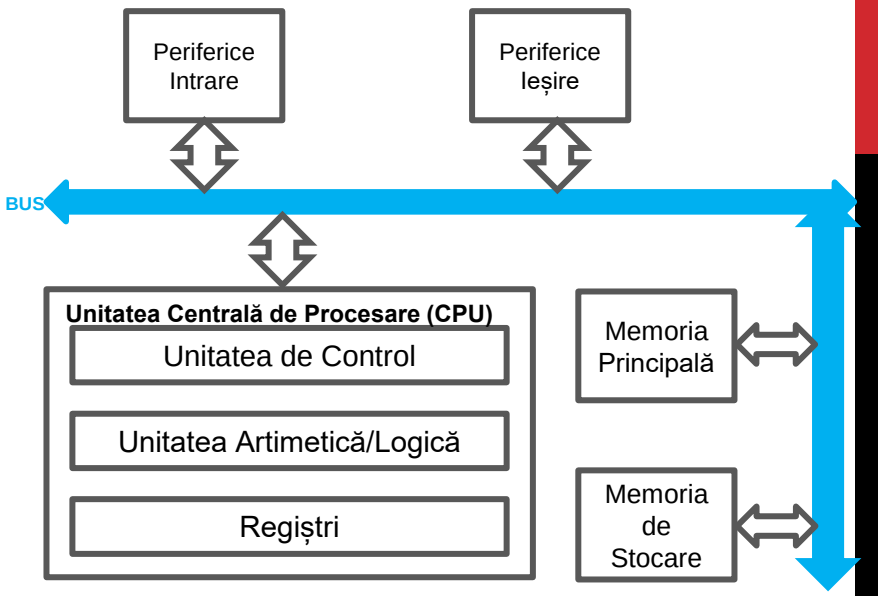
- **Unitatea Centrală de Procesare**
  - a.k.a. CPU
  - este “creierul” unității de calcul
  - execută instrucțiuni

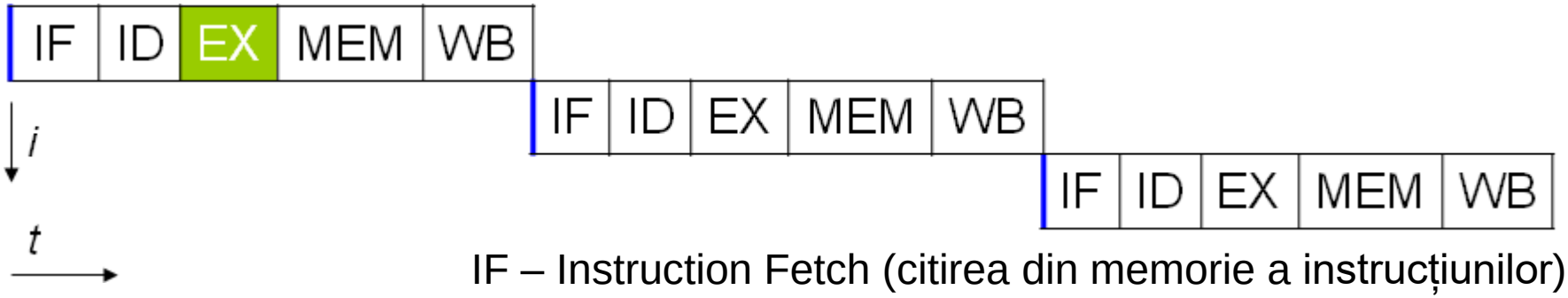


  - 5 componente principale:
    - **UC (“instrucțiunile”)**
      - fetch
        - IP = 10011 (locația în memorie de unde să citim biții)
        - după citire, IP este actualizat
      - decode
        - s-a citit “1110011” care este decodat în
          - opcode = 111, operand1 = 00 operand2 = 11
          - de exemplu: 111 = “adună registrul A la registrul R”, R = 00 este EAX, A = 11 este EDX (prin convenție)
      - execute
          - trimite EAX ← EAX + EDX la UAL
          - citește rezultatul din UAL și pune-l în registrul EAX

ARHITECTURA DE BAZĂ

- **Unitatea Centrală de Procesare**
  - a.k.a. CPU
  - este “creierul” unității de calcul
  - execută instrucțiuni
  - 5 componente principale:
    - **UC (“instrucțiunile”)**





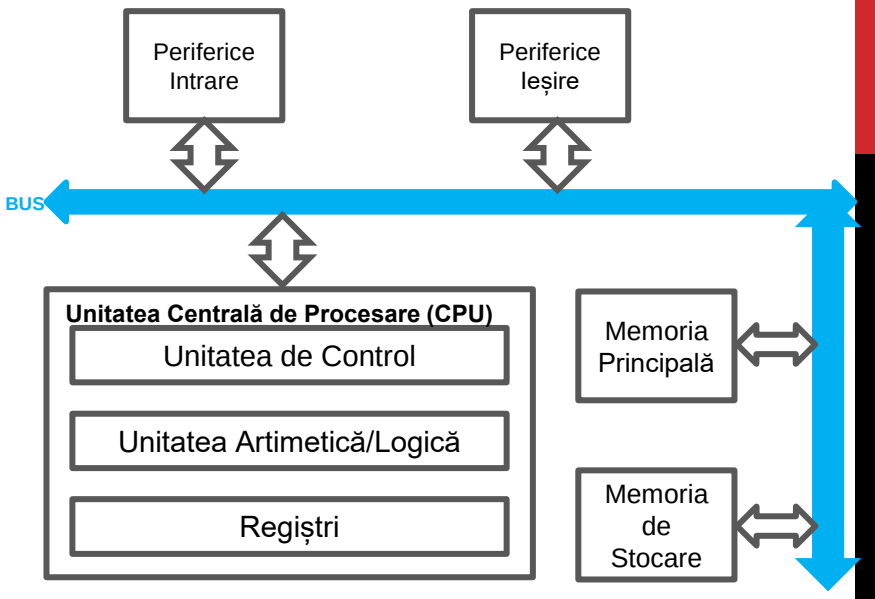
IF – Instruction Fetch (citirea din memorie a instrucțiunilor)
ID – Instruction Decode (circuit secvențial care decodează)
EX – Execute (execuția propriu-zisă)
MEM – Memory Access (orice access memorie)
WB – Write Back (scrie rezultatul înapoi în memorie)

https://en.wikipedia.org/wiki/Central_processing_unit

ARHITECTURA DE BAZĂ

- **Unitatea Centrală de Procesare**
  - a.k.a. CPU
  - este “creierul” unității de calcul
  - execută instrucțiuni

  - 5 componente principale:
    - **Clock**
    - **regiștri (“memoria”)**
    - **UAL (“operații”)**
    - **BUS**
    - **UC (“instrucțiunile”)**

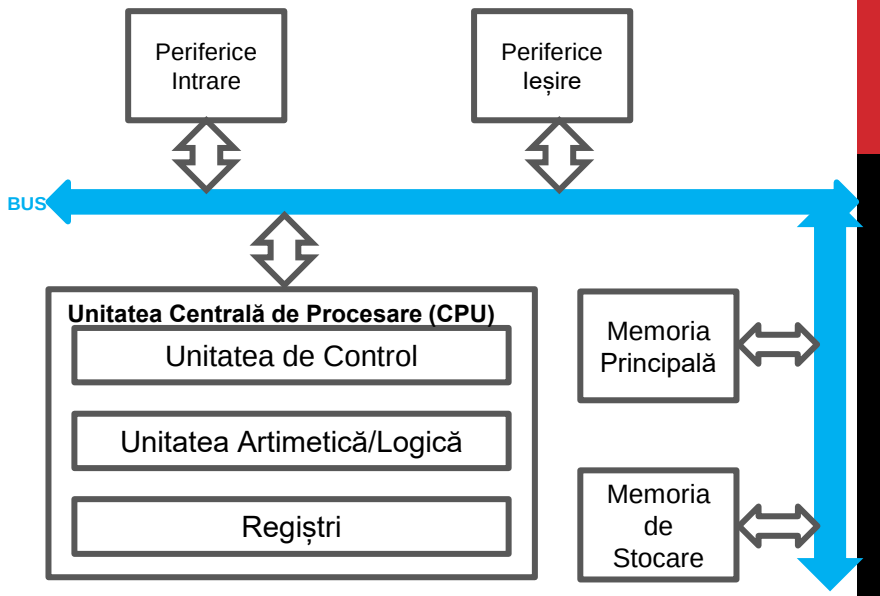


PROCESOR

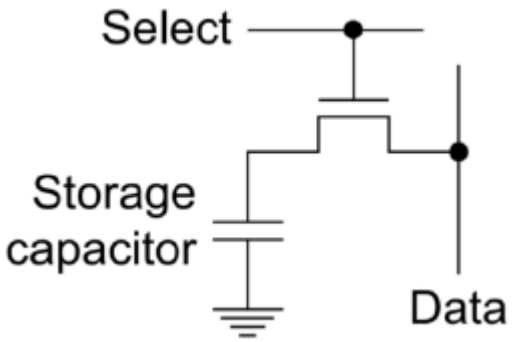
Producator procesor	Intel®
Tip procesor	i9
Model procesor	9880H
Arhitectura	Coffee Lake
Numar nuclee	8
Frecventa nominala	2.3 GHz
Cache	16384 KB
Frecventa Turbo Boost	4.8 GHz
Tehnologie procesor	14 nm
Procesor grafic integrat	Intel® UHD Graphics 630

ARHITECTURA DE BAZĂ

- **Memoria Principală**
  - **conține cod și date**
  - **este volatilă**



  - **Static RAM (SRAM)**
    - **bazată pe flip-flops**
    - **rapid**
    - **scump**
    - **regiștrii din CPU sunt de același tip**



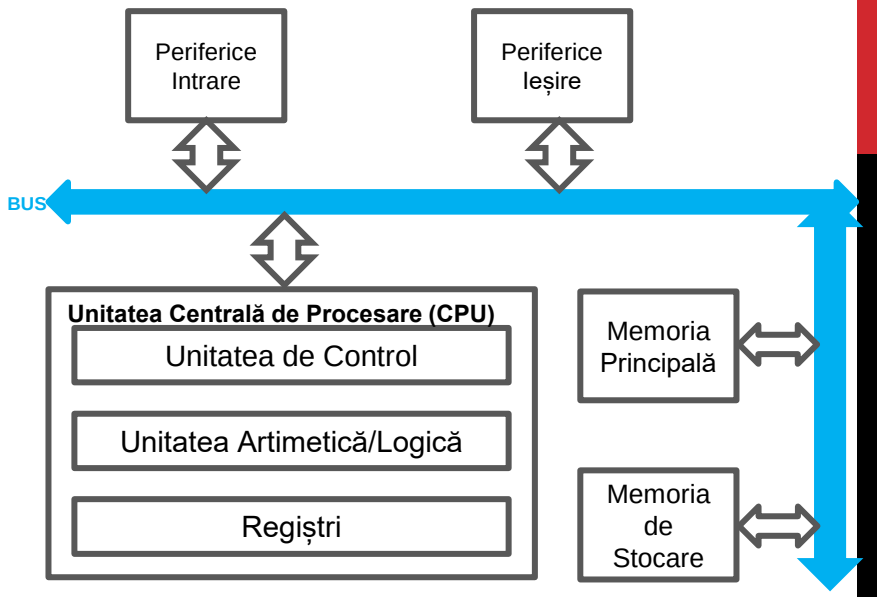
  - **Dynamic RAM (DRAM)**
    - **fiecare bit este reprezentat de o combinație tranzistor + condensator**
    - **condensatoarele suferă de leakeage (scurgeri de tensiune)**
    - **DRAM trebuie actualizat o dată la fiecare câteva zeci de ms**

de ce este DRAM mai ieftin decât SRAM?
are DRAM niște dezavantaje în comparație cu SRAM?

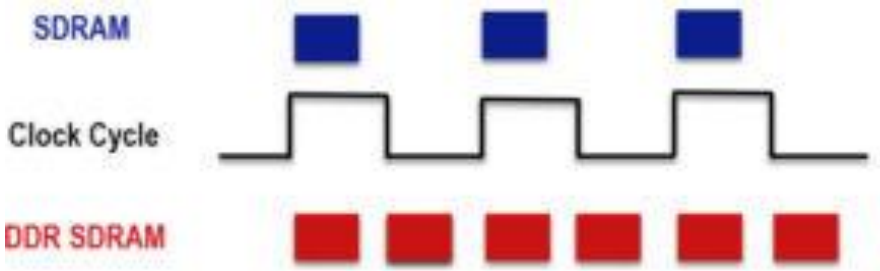
https://en.wikipedia.org/wiki/Dynamic_random-access_memory, https://en.wikipedia.org/wiki/DDR_SDRAM

ARHITECTURA DE BAZĂ

- **Memoria Principală**
  - **conține cod și date**
  - **este volatilă**



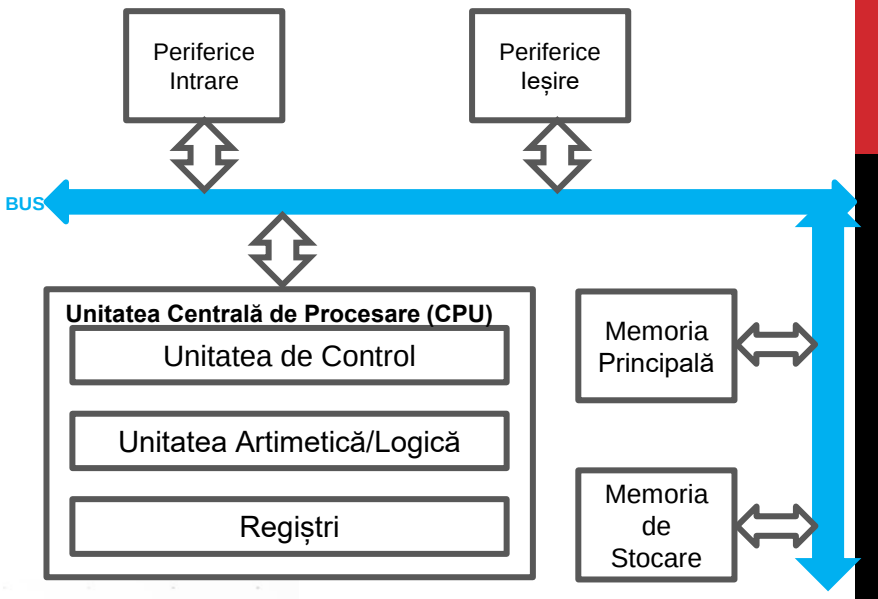
  - **DDR RAM**
    - **Double Data Rate RAM**
    - **.../DDR4/DDR5/DDR6**
    - **performanța este definită de:**
      - **capacitate**
      - **dacă au un sistem intern de corectarea erorilor (ECC)**
      - **timpi de acces (în cât timp de la comanda de citire de biți din RAM avem datele disponibile?, timpul de refresh)**
      - **consumul de energie**

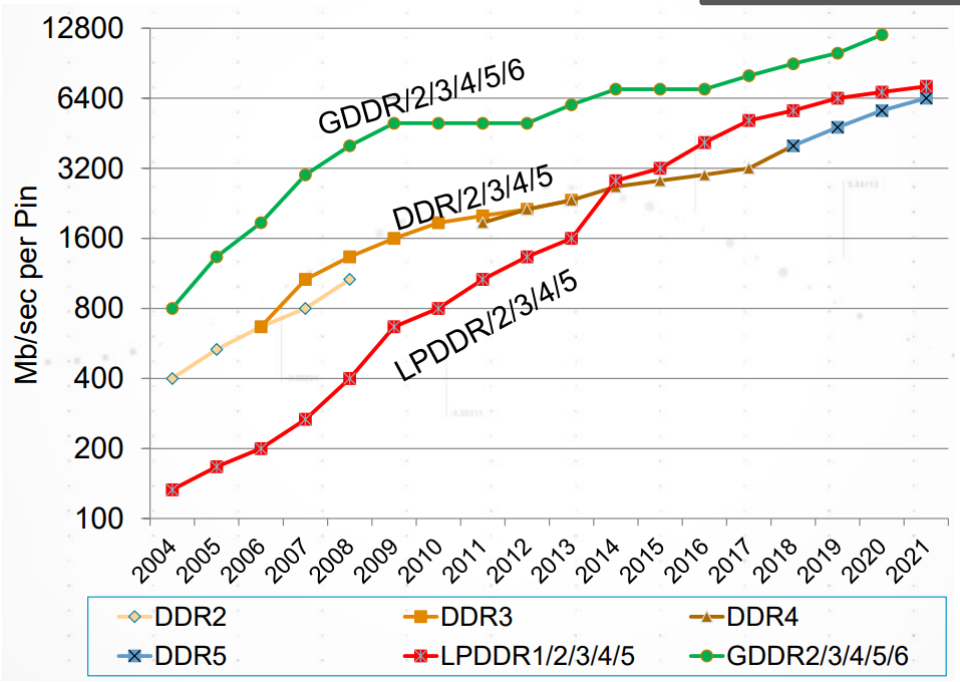


https://www.microcontrollertips.com/understanding-ddr-sdram-faq/

ARHITECTURA DE BAZĂ

- **Memoria Principală**
  - **conține cod și date**
  - **este volatilă**
  - **ce avem azi?**

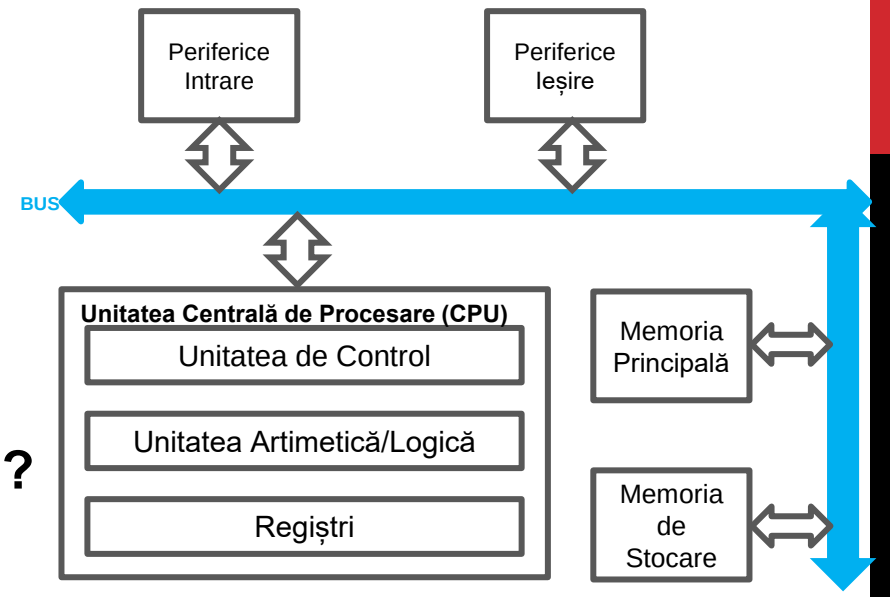




https://resources.altium.com/p/ddr5-vs-ddr6-heres-what-expect-ram-modules

ARHITECTURA DE BAZĂ

- **Memoria Principală**
  - **conține cod și date**
  - **este volatilă**
  - **ce ne interesează la memorie?**



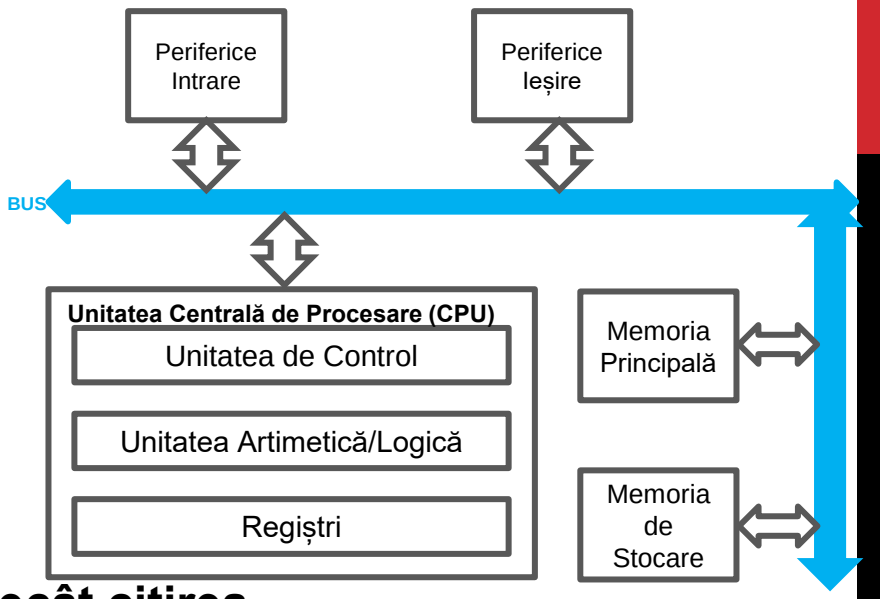
MEMORIE

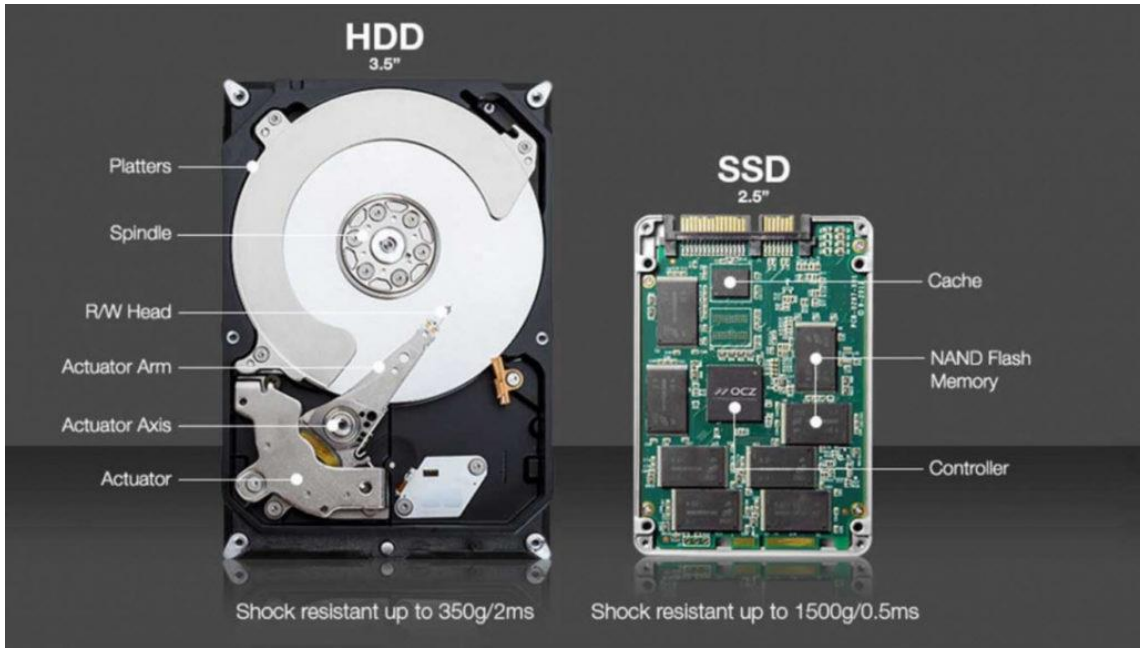
Capacitate memorie	32 GB
Tip memorie	DDR4
Numar sloturi	4
Sloturi ocupate	2
Frecventa	2666 MHz
Capacitate memorie maxima suportata	128 GB

https://www.microcontrollertips.com/understanding-ddr-sdram-faq/

ARHITECTURA DE BAZĂ

- **Memoria de Stocare**
  - **conține cod și date**
  - **este nevolatilă**
  - **SSD (Solid State Disks)**
    - **e memorie flash, rapidă**
    - **azi, e scumpă**
    - **scrierea e mult mai lentă decât citirea**
  - **HDD (Hard Disks)**
    - **mecanic**

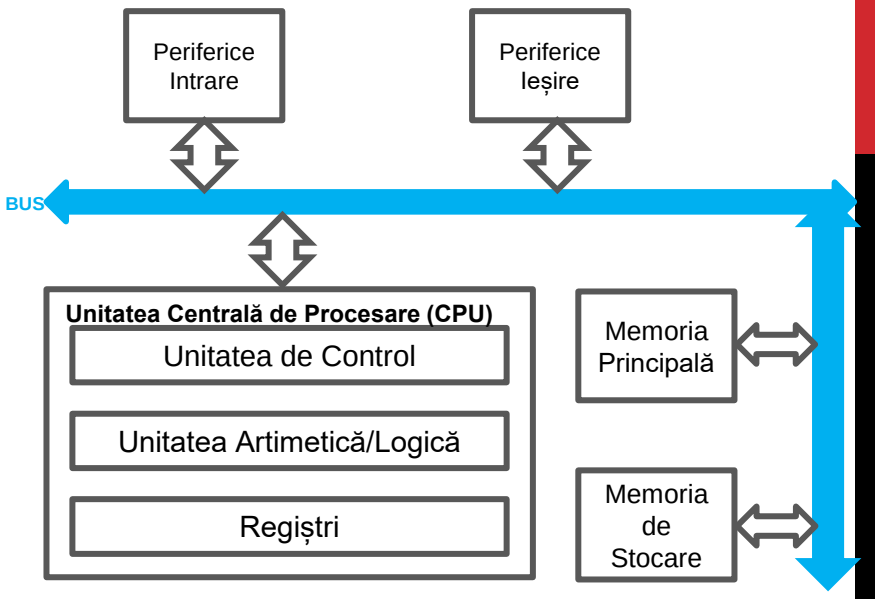




https://www.eeworldonline.com/ssds-vs-hdds-part-2-sand-or-rust/

ARHITECTURA DE BAZĂ

- **BUS**
  - **conectează CPU/memorie**
  - **proprietăți**
    - **capacitatea (bandwidth)**
    - **viteza (MHz)**



ARHITECTURA DE BAZĂ

Periferice Intrare

Periferice Ieșire

BUS

Unitatea Centrală de Procesare (CPU)

Unitatea de Control

Unitatea Artimetică/Logică

Regiștri

Memoria Principală

Memoria de Stocare

un astfel de sistem poate executa *doar cod mașină* pentru un *singur stream*

CE AM FĂCUT ASTĂZI

- **arhitectura de bază a calculatoarelor**

- **Instruction Set Architecture (ISA)**

DATA VIITOARE ...

- continuăm discuția despre arhitectura de bază a calculatoarelor

- de la cod sursă la cod mașină

LECTURĂ SUPLIMENTARĂ

- **PH book**
  - 2.5 Representing Instructions in the Computer
  - 4.1 – 4.4 The Processor

- **Ben Eater, Designing a 7-segment hex decoder, https://www.youtube.com/watch?v=7zffjsXqATg**
- **Ben Eater, Using an EEPROM to replace combinational logic, https://www.youtube.com/watch?v=BA12Z7gQ4P0**

- **Crash Course Computer Science (o descriere grafică intuitivă, corectă):**
  - How Computers Calculate - the ALU, https://www.youtube.com/watch?v=1I5ZMmrOfnA&list=PLH2l6uzC4UEW0s7-KewFLBC1D0l6XRfye&index=6
  - Registers and RAM, https://www.youtube.com/watch?v=fpnE6UAfbtU&list=PLH2l6uzC4UEW0s7-KewFLBC1D0l6XRfye&index=7
  - The Central Processing Unit (CPU), https://www.youtube.com/watch?v=FZGugFqdr60&list=PLH2l6uzC4UEW0s7-KewFLBC1D0l6XRfye&index=8
  - Instructions & Programs, https://www.youtube.com/watch?v=zltgXvg6r3k&list=PLH2l6uzC4UEW0s7-KewFLBC1D0l6XRfye&index=9
- **I Made a Working Computer with just Redstone!, https://youtu.be/CW9N6kGbu2I**

.

.