

Bazele Tehnologiei Informației

Curs introductiv

Facultatea de Cibernetică, Statistică și Informatică Economică
(CSIE) – ASE București

Prof. dr. Răzvan ZOTA

zota@ase.ro

<https://zota.ase.ro/bti>

Conținut curs

1. Introducere. Scurt istoric al calculatoarelor. Arhitectura de bază a unui PC.
2. Elemente de teoria transmisiei informației
3. Bazele numerice ale calculatoarelor
4. Bazele logice ale calculatoarelor
5. Microprocesorul (+ ASM)
6. Memoria și magistralele de sistem
7. Dispozitive periferice
8. Dispozitive de stocare a informației
9. Rețele de calculatoare, IoT, Cloud computing, IA - concepte de bază

Bibliografie

1. Bazele Tehnologiei Informației, Floarea Năstase, Răzvan Zota, Ed. ASE, 2014.
2. <https://mihai-gheorghe.gitbook.io/bti-suport-curs/01-informatia>
3. The Architecture of Computer Hardware, Systems Software, & Networking - An Information Technology Approach 4th ed - I. Englander-Wiley-2009.
4. <https://cs-fundamentals.streamlit.app/>

Conținut seminar

Seminarii 1-7. Aplicații la:

- Teoria transmisiei informației, codificarea informației în calculator
- Bazele numerice ale calculatoarelor, coduri detectoare și corectoare de erori
- Bazele logice ale calculatoarelor
- Intel ASM (Assembler)

Teste după fiecare seminar (cu excepția ultimului)

Modalitate de notare

Nota finală se calculează cu formula $NF = 0,7 \times E + 0,3 \times S$ dacă și numai dacă $E \geq 5$. Dacă $E < 5$, $NF = [E]$; unde:

- NF reprezintă nota finală,
- E reprezintă nota de la examenul scris,
- S reprezintă nota de la seminar.

Pentru promovare este necesar un minim de 5 puncte (înainte de rotunjire) la examenul scris (E).

Scurt istoric al calculatoarelor

Abacul: Primul “Computer Automat”

- ◆ În forma cunoscută astăzi, apare (documentat) pentru prima oară în jurul anului 1200 în China (*suan-pan*) din necesitatea de a automatiza procesul de numărare



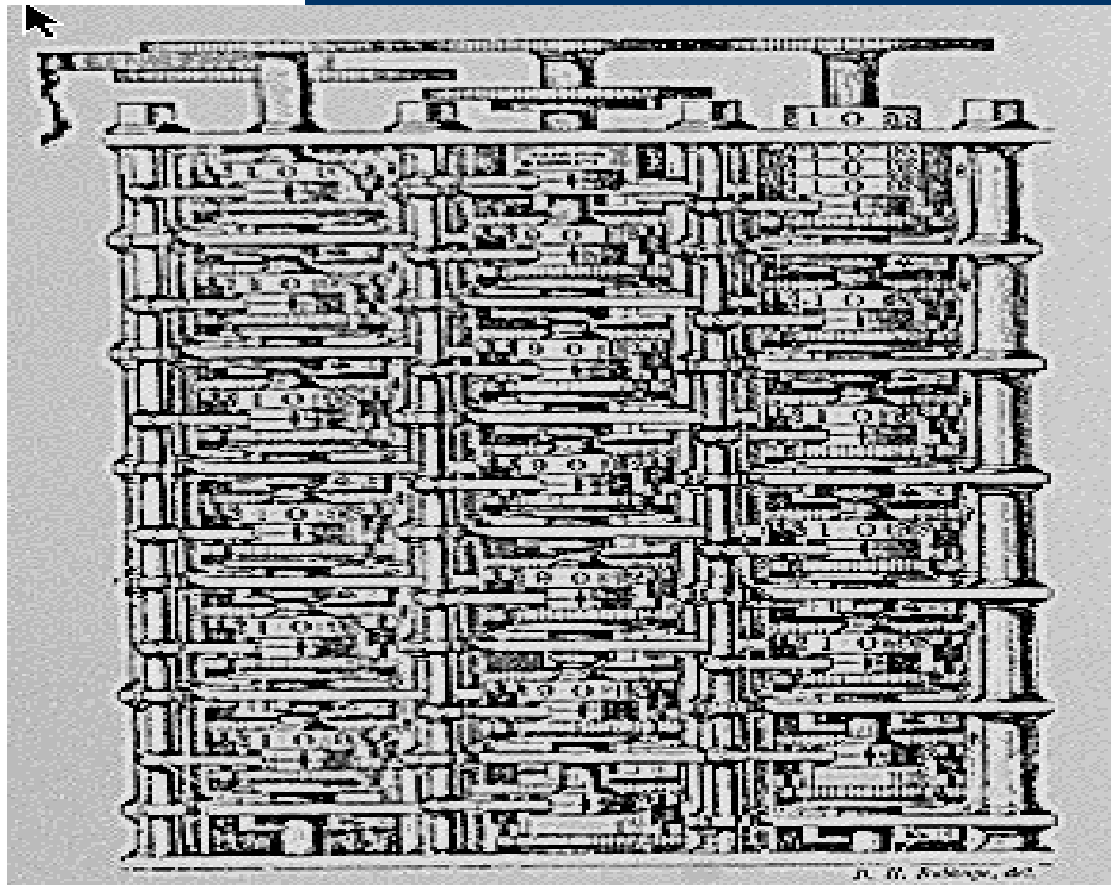
Pionieri - Blaise Pascal (1623-1662)

- ◆ Matematician francez care a inventat prima mașină de calcul operațională
- ◆ “Arithmetic Machine” – introdusă în 1642
 - Adunare și scădere
 - Scăderea se făcea folosindu-se tehnici complementare (similare cu cele utilizate în computerele moderne)
 - Înmulțirea și împărțirea erau implementate prin serii de adunări sau scăderi

Pionieri - Charles Babbage (1791-1871)

- ♦ Matematician britanic care a inventat primul dispozitiv ce poate fi considerat un computer în sensul modern al cuvântului
- ♦ Erau calculate tabele de funcții logaritmice și trigonometrice de către oameni ce se numeau “**computers**”
 - “Difference Engine” (1822) – construită parțial
 - “Analytical Engine” (1830)
- ♦ “Difference Engine” a fost construită ulterior conform desenelor originale de către o echipă la Muzeul Științei din Londra.
 - 4000 componente
 - Cântărea 3 tone, aprox. 3 metri lățime, 2 ½ lungime
 - Echipamentul a efectuat prima secvență de calcule la începutul anilor 1990 obținându-se rezultate cu o precizie de 31 de cifre zecimale

“Difference Engine”



Claude Shannon

- ◆ În jurul anului 1850 matematicianul englez George Boole a inventat Algebra booleană (Algebra Boole)
- ◆ Algebra Boole a rămas relativ necunoscută și neutilizată până în anul 1938
- ◆ Teza de masterat a lui C. Shannon din 1938 a demonstrat cum conceptele lui Boole “TRUE” și “FALSE” pot fi utilizate pentru a reprezenta funcționalitatea comutatoarelor din circuitele electronice

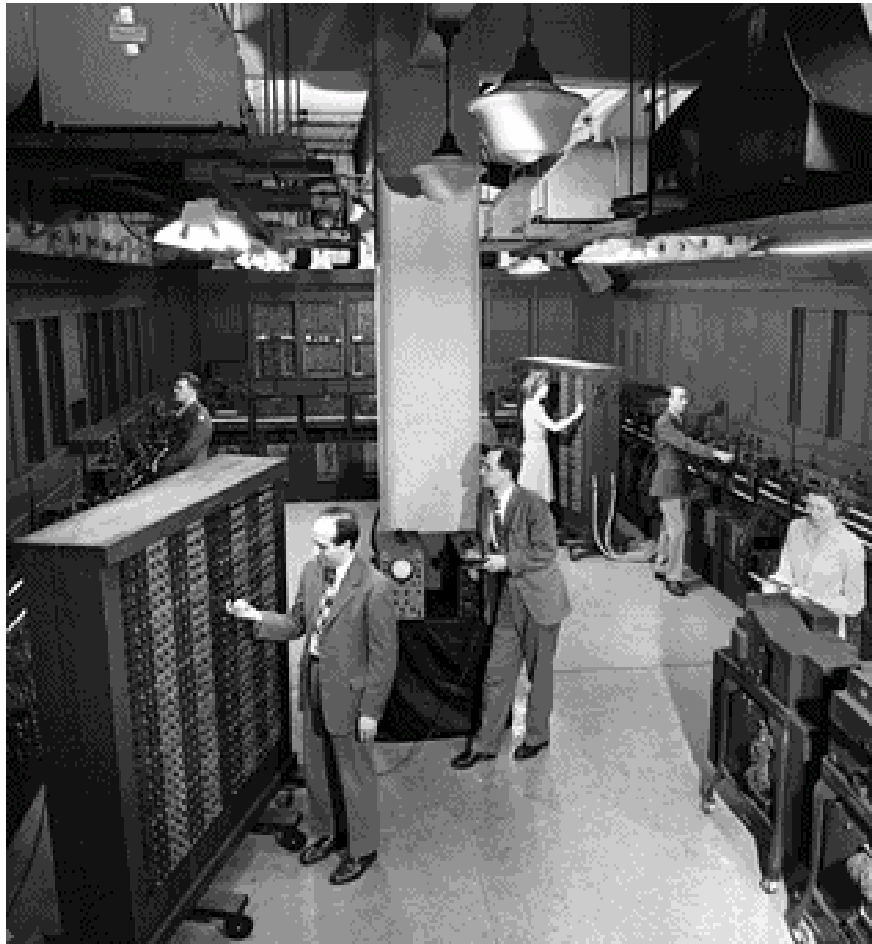
Howard Aiken și calculatorul IBM Harvard Mark I

- ◆ Harvard Mark 1 (*IBM Automatic Controlled Calculator*) a fost construit între anii 1939 și 1944
- ◆ Era format din mai multe calculatoare ce lucrau asupra unor părți ale aceleiași probleme sub supravegherea unei singure unități de control
- ◆ Construit din comutatoare, relee și alte dispozitive mecanice
- ◆ Conținea 750.000 de componente și avea 16 m lungime, 2 ½ înălțime și cântărea 5 tone
- ◆ Numerele erau de 23 de cifre
 - O înmulțire dintre două numere dura 4 secunde
 - O împărțire dura 10 secunde

William Mauchly, J. Presper Eckert - ENIAC - Electronic Numerical Integrator And Computer

- ◆ Construit la Universitatea din Pennsylvania (1943-1946)
- ◆ 3 metri înălțime, 30 mp spațiu, 30 tone
- ◆ 18.000 tuburi cu vacuum
- ◆ 150 kW putere (suficient pentru a ilumina un mic oraș)
- ◆ Problema cheie cu acest tip de calculator era fiabilitatea
 - aprox. 50 tuburi erau înlocuite în fiecare zi
- ◆ 1943 - Eckert și Mauchly au inițiat conceptul de creare a unui program stocat în calculator pentru care era folosită o memorie internă utilizată pentru a stoca atât instrucțiuni cât și date

ENIAC - 1946



Generațiile următoare

- ◆ EDVAC - Electronic Discrete Variable Automatic Computer
 - 4000 tuburi
- ◆ EDSAC - Electronic Delay Storage Automatic Calculator (1949)
 - 3000 tuburi
- ◆ UNIVAC I - Universal Automatic Computer (1951)
 - Primul computer comercial
- ◆ ILLIAC I (1949)
 - Construit la Universitatea Illinois, primul computer deținut de o instituție academică

John Von Neumann

- ◆ Matematicianul Von Neumann a lucrat drept consultant pentru proiectele ENIAC și EDVAC
- ◆ *Prima versiune a unui raport despre EDVAC – 1945* conținea elementele de bază ale unui program stocat în computer
 - **O memorie** ce conținea atât date cât și instrucțiuni
 - **O unitate de calcul** ce putea efectua atât calcule aritmetice cât și logice asupra datelor
 - **O unitate de control** ce putea interpreta o instrucțiune luată din memorie și selecta direcții diferite de acțiune pe baza rezultatelor operațiilor anterioare

Primul tranzistor

- ◆ Bell Laboratories a început cercetarea în domeniul semiconductorilor în 1945
- ◆ William Shockley, Walter Brattain și John Bardeen au creat primul tranzistor pe 23 decembrie 1947
 - Au luat o pauză pentru sărbătorile de Crăciun înainte de a publica evenimentul; de aceea cărțile de referință indică faptul că primul tranzistor a fost creat în 1948

Primul 'supercomputer' cu un hard-disk

- ◆ *In September 1956 IBM launched the 305 RAMAC, the first 'SUPER' computer with a hard disk drive (HDD). The HDD weighed over a ton and stored 5 MB of data.'*



Primul circuit integrat

- ◆ Jack Kilby (Texas Instruments) în 1958 a reușit să combine mai multe componente pe o singură bucată de semiconductor
- ◆ În 1961 Fairchild și Texas Instruments au realizat primele circuite integrate comerciale ce conțineau funcții logice de bază
 - 2 porți logice (4 tranzistori bipolari și 4 rezistoare)
- ◆ În 1970 Fairchild introducea pentru prima oară memoria de 256 biți static RAM



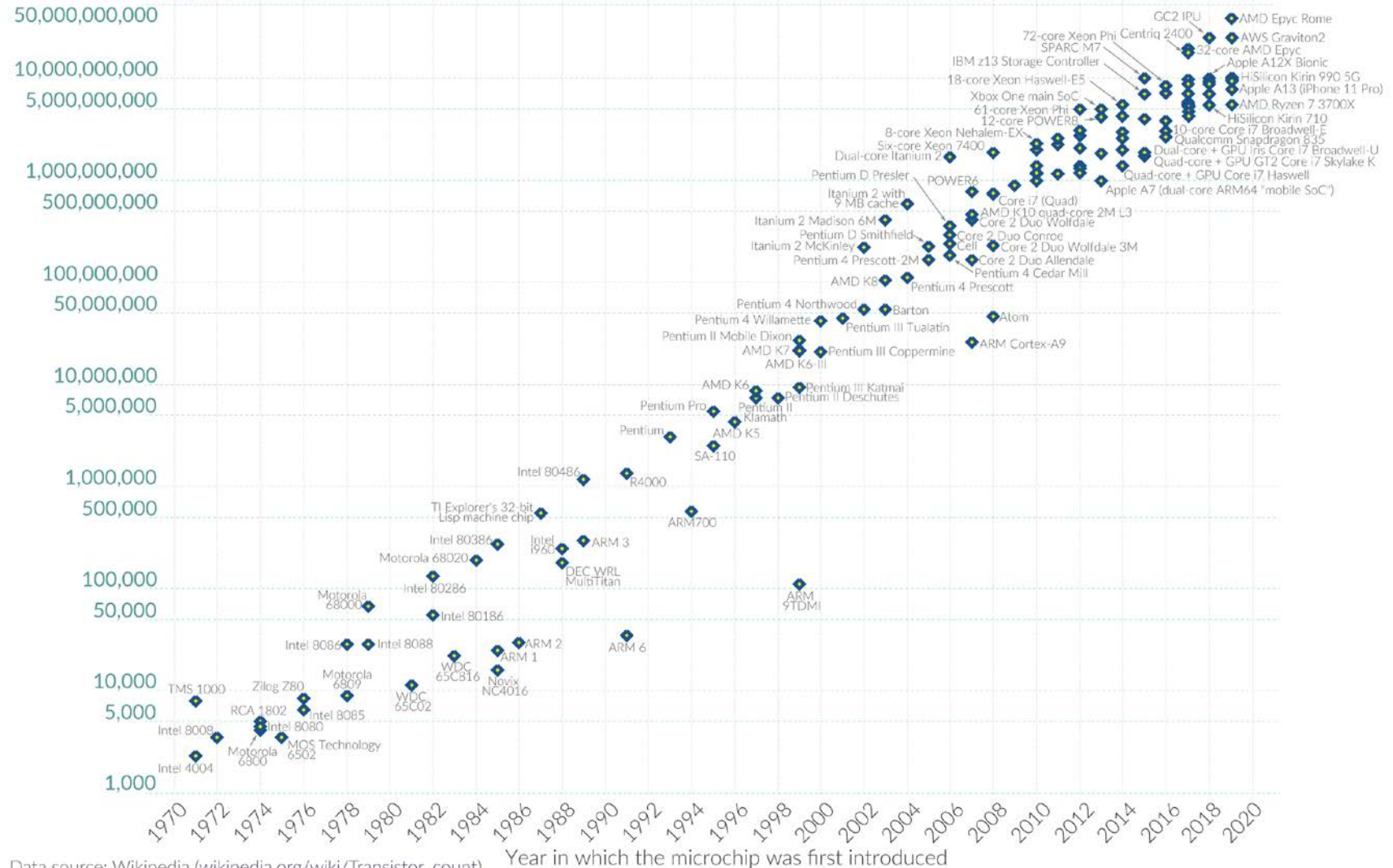
Legea lui Moore

- ◆ În 1965 Gordon Moore a prevăzut că numărul de tranzistori de pe un microprocesor se va dubla într-un interval de aprox. 18-24 luni (vezi slide-ul următor)

Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Transistor count



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

OurWorldinData.org – Research and data to make progress against the world's largest problems.

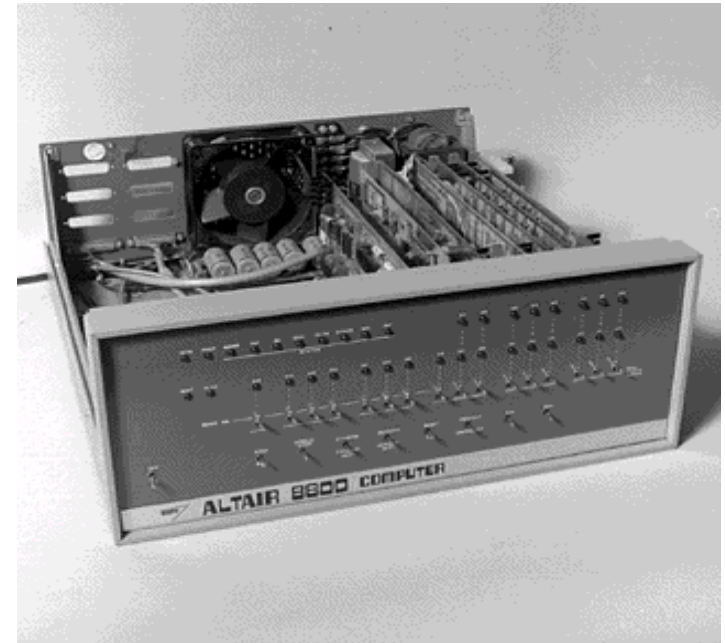
Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

Drumul către primul calculator personal (PC)

- ◆ Computerele încep să utilizeze tranzistoare (anii 1960)
- ◆ Anii “*big iron*”: mainframe-uri IBM
- ◆ În 1970 compania japoneză Busicom de calculatoare a cerut lui Intel un set de 12 CI pentru a le utiliza într-un nou calculator
 - T. Hoff, proiectant la Intel, inspirat de cerea făcută anterior a creat primul microprocesor, denumit 4004
 - 2300 de tranzistoare; 60.000 operații pe secundă
- ◆ Primul microprocesor de uz general - 8080, a fost introdus de Intel în 1974
 - 8-biți, 4500 tranzistori, 200.000 operații pe secundă
- ◆ Alte procesoare: Motorola 6800, MOS Technology 6502, Zilog Z80

Calculatoare personale

- ◆ Ed Roberts concepe Altair 8800 (1974)
 - bazat pe 8080
 - preț 375\$
 - Fără tastatură, ecran, capacitate de stocare
 - 4k memorie, programabil prin intermediul unui panou frontal cu comutatoare
- ◆ Bill Gates și Paul Allen fondează Microsoft (1975)
 - BASIC 2.0 pe Altair 8800
 - Primul limbaj de nivel înalt disponibil pe un calculator personal



Calculatoare personale (cont.)

- ◆ S. Wozniak și S. Jobs:
 - Apple 1 - 1976
 - Apple II - 1977
 - 16k ROM, 4k de RAM, tastatură și display color
 - preț 1300\$, în 1977 afacere de 700.000 \$ iar în 1978 de 7 mil.
- ◆ TRS-80 (bazat pe Z80) de la Radio Shack - 1977
 - 4k ROM, 4k RAM, tastatură și drive de tip casetă
 - preț 600\$
- ◆ Primul PC (Personal Computer) de la IBM - 1981
 - Microprocesor pe 16-biți 8088, ROM BASIC, floppy-disc 360K, DOS 1.0
 - preț 1365 \$

Calculatoare personale (cont.)

- ◆ 1983 IBM XT are hard-disc (10Mb costa 3000\$)
- ◆ 1985 Intel introduce 80386
 - Primul membru pe 32-biți din familia 80x86
- ◆ 1986 Compaq introduce primul sistem bazat pe 80386
- ◆ 1989 Intel introduce 80486, ce includea coprocesor matematic
- ◆ 1992 Intel Pentium (64-biți) magistrala de memorie
 - AMD, Cyrix 486 procesoare compatibile (clonă)
- ◆ 1996 Intel Pentium Pro
- ◆ 1998 Intel Pentium II
- ◆ 2000 Intel Pentium IV la 1.5 GHz

Calculatoare personale (cont.)

- ♦ Pentium M, Celeron M – 2003
- ♦ Intel Core (65nm) Duo/Solo – Ian 2006
- ♦ Dual Core Xeon – 2006
- ♦ Intel Core 2 (65nm) Duo – Iulie 2006
- ♦ **Intel Core i3, i5, i7** (45nm) – 2009 (Arhitectura *Nehalem*) cu variante de 2,4,6,8,10,12 core (*731 milioane de tranzistori pentru varianta quad-core*)
- ♦ 2016 – Intel Core i3, i5, i7, 7th generation (tehnologie de 14 nm)

Calculatoare personale (cont.)

- ◆ Generația a 8-a de la Intel: Core i9



Intel® Core™ i9-8950HK Processor

- 12 MB SmartCache Cache
- 6 Cores
- 12 Threads
- 4.80 GHz Max Turbo Frequency
- HK - High performance graphics, unlocked
- 8th Generation

- ◆ Generația a 9-a de la Intel: Core i9 – 8 cores, 16 MB Cache, 5 GHz Max Turbo Frequency
- ◆ Generația a 10-a Intel: Core i9 – 10 cores, 20 MB Cache
- ◆ Generația a 11-a Intel: Core i9 – 8 cores, 16 MB Cache, 5.20 GHz Max Turbo Frequency

Calculatoare portabile (laptop, PDA, ultrabook, etc.)



iPhone (și nu numai) !



Moștenirea lui John von Neumann

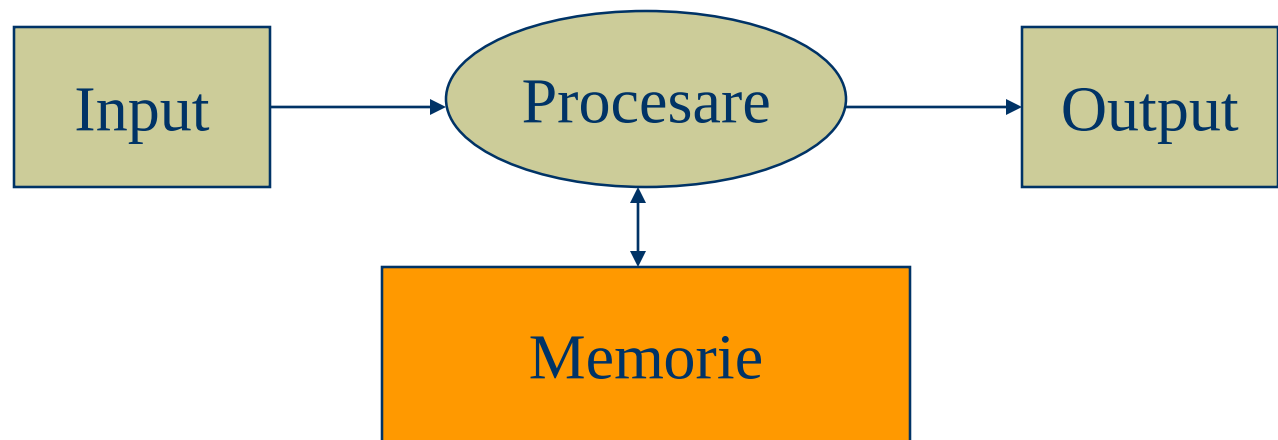
Hardware

Software

- Sisteme de operare (oferă mediul de operare pentru aplicații – utilizat de către acestea pentru a accesa resursele calculatorului)
 - Sunt specifice platformei pentru care sunt proiectate
- Aplicații (procesoare de text, programe pentru baze de date, navigatoare web, etc.)

I/E, procesare, stocare

- ◆ Intrare (input)
- ◆ Procesare
- ◆ ieșire (output)
- ◆ Stocarea informațiilor



Taxonomia lui Flynn – calcul paralel

- ♦ ***SISD - Single Instruction (Stream), Single Data (Stream)***

Toate computerele von Neumann.

Ex. IBM 370, DEC VAX, SUN, IBM PC, MacIntosh.

- ♦ ***SIMD - Single Instruction (Stream), Multiple Data (Stream)***

Un singur flux de instrucțiuni se aplică pe mai multe seturi de date în paralel.

Ex. CRAY-1, DAP CM-1, WARP, CM-2, ILLIAC IV.

- ♦ ***MISD - Multiple Instruction (Stream), Single Data (Stream)***

Mai multe instrucțiuni operează asupra aceluiași set de date

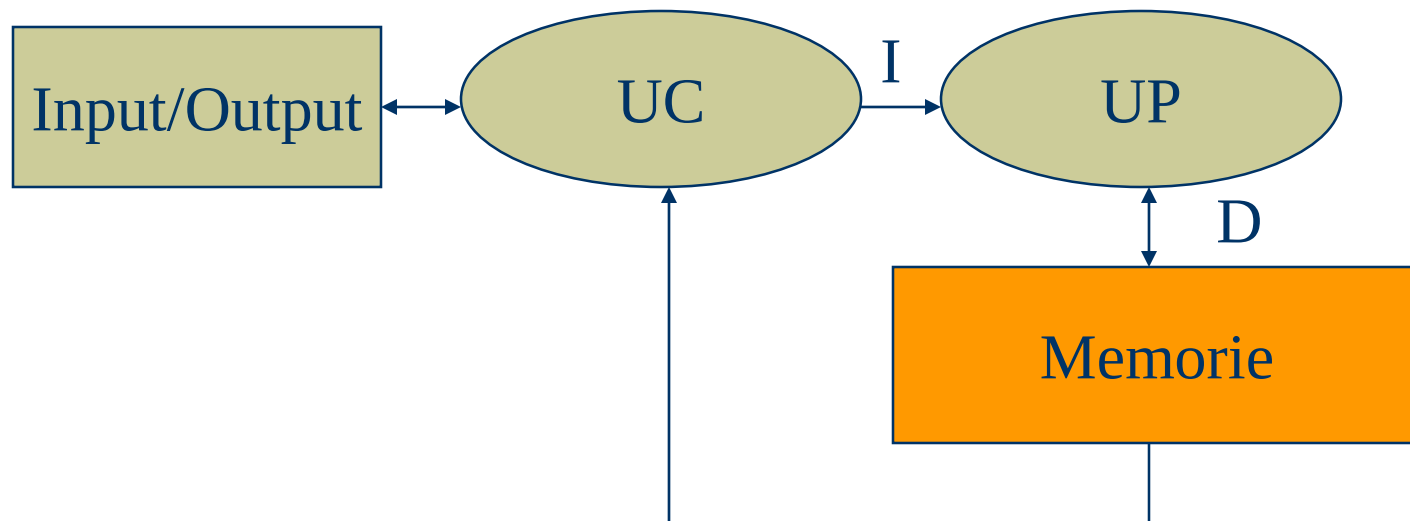
- ♦ ***MIMD - Multiple Instruction (Stream), Multiple Data (Stream)***

Mai multe UCP independente ce operează ca părți ale unui sistem mai mare.

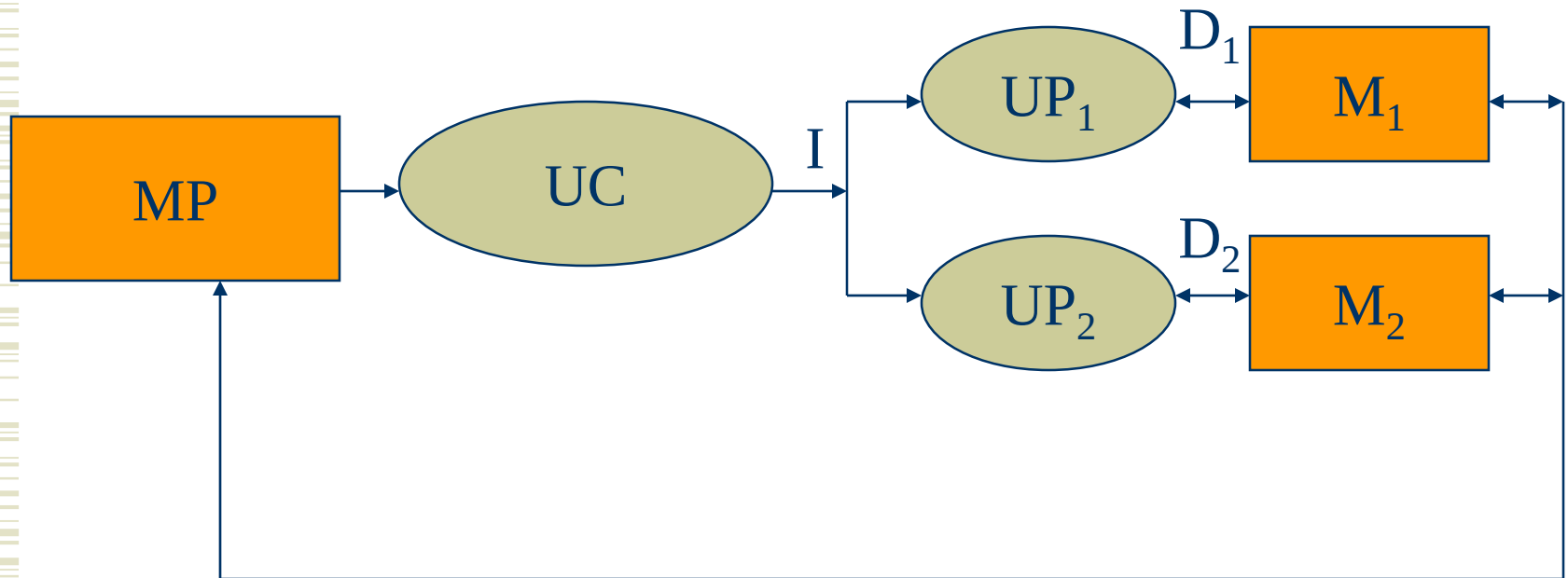
Majoritatea procesoarelor paralele aparțin acestei categorii.

Ex. Transputere, Supernode, DADO, N-cube, Ultracomputer, Butterfly, Alliant, Sequent Balance, CRAY X-MP.

SISD (Single Instruction, Single Data)



SIMD (Single Instruction, Multiple Data)



Obs. Asemănător: SPMD

Exemple SIMD

SIMD (Single Instruction, Multiple Data) este un tip de arhitectură de procesare paralelă care permite unei singure instrucțiuni să efectueze aceeași operație pe mai multe elemente de date simultan.

SIMD este utilizat în mod obișnuit în diverse sisteme informatice, în special în aplicații multimedia și științifice unde procesarea în paralel a unor cantități mari de date este benefică.

Exemple de computere și procesoare care utilizează tehnologia SIMD:

1. **Intel SSE** (extensii SIMD de *streaming*): - Intel a introdus SSE ca o extensie a arhitecturii lor x86. SSE oferă un set de instrucțiuni SIMD pentru sarcini multimedia și de procesare a datelor. - Exemple de procesoare: Intel Pentium 4, seria Intel Core (de exemplu, Core i7) și procesoare Intel ulterioare
2. **AMD SSE** (extensii SIMD de *streaming*): - Procesoarele AMD acceptă și instrucțiuni SSE, oferind capabilități SIMD similare cu procesoarele Intel. - Exemple de procesoare: AMD Athlon 64, seria AMD Ryzen.

Exemple SIMD

3. **ARM NEON:** - Tehnologia NEON de la ARM este o extensie a arhitecturii SIMD pentru procesoarele ARM. Este adesea folosit în dispozitivele mobile, procesarea semnalului digital și aplicațiile multimedia. - Exemple de procesoare: diverse procesoare bazate pe ARM utilizate în smartphone-uri și tablete.

4. **GPU-uri NVIDIA** (Unități de procesare grafică): - Plăcile grafice de la NVIDIA, cum ar fi cele din seriile GeForce și Quadro, utilizează procesarea SIMD pentru a accelera redarea grafică și calculul științific. - Exemple de GPU: seria NVIDIA GeForce RTX 30, seria NVIDIA Quadro P.

5. **GPU-uri AMD Radeon:** - Plăcile grafice AMD Radeon folosesc, de asemenea, arhitectura SIMD pentru procesarea paralelă, beneficiind de jocuri și sarcini de lucru de calcul. - Exemple de GPU: seria AMD Radeon RX 6000.

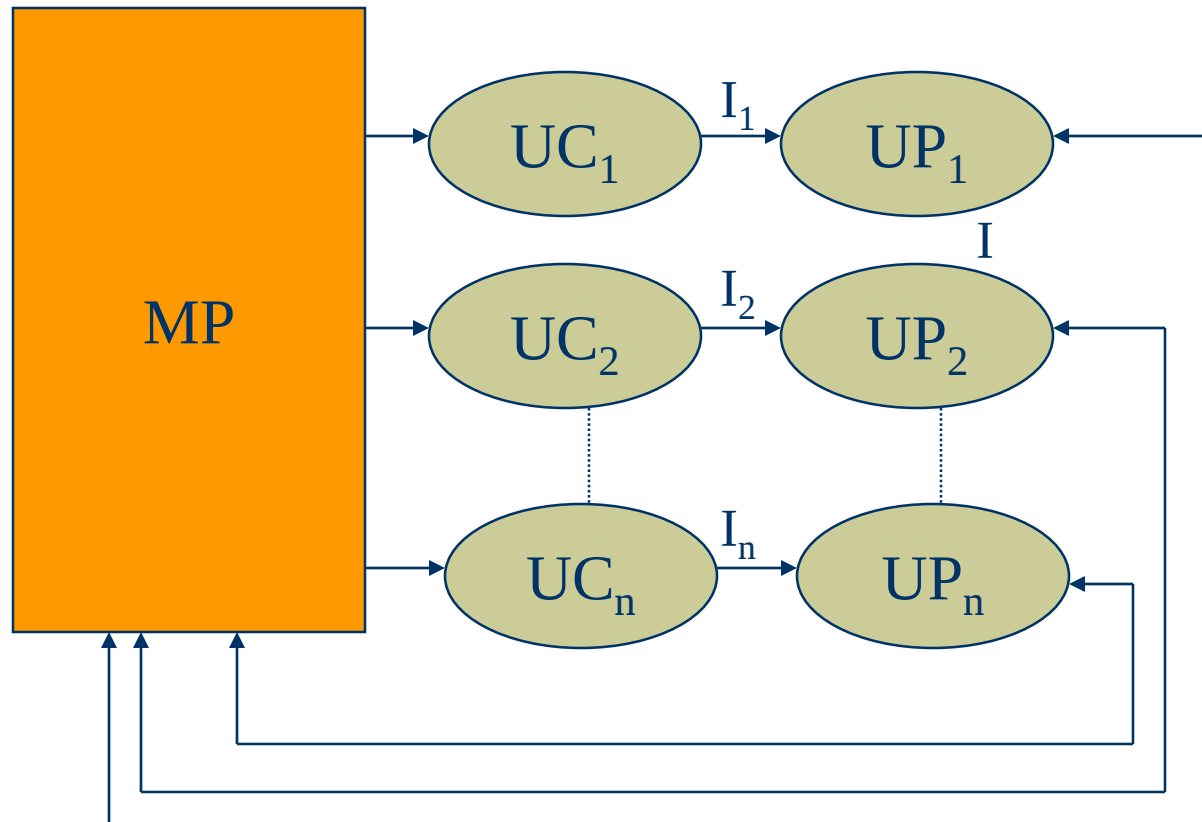
Exemple SIMD

6. Console PlayStation și Xbox: - Consolele de jocuri precum seria PlayStation și Xbox utilizează procesoare SIMD pentru a îmbunătăți grafica și performanța jocurilor.
- Exemple de console: PlayStation 5 (folosește arhitectura AMD RDNA 2), Xbox Series X (folosește arhitectura AMD RDNA 2)

7. Intel Xeon Phi (fostul Knights Landing): - Intel Xeon Phi este o familie de procesoare de înaltă performanță concepute pentru calcule paralele și simulări științifice, utilizând pe scară largă tehnologia SIMD.

Tehnologia SIMD joacă un rol crucial în optimizarea performanței pentru sarcini care necesită operațiuni repetitive pe seturi mari de date, cum ar fi procesarea multimedia, redarea grafică 3D și simulările științifice.

MIMD (Multiple Instruction, Multiple Data)



Exemple MIMD

MIMD (Multiple Instruction, Multiple Data) este o arhitectură de procesare paralelă care permite mai multor procesoare sau nuclee să execute diferite instrucțiuni pe diferite seturi de date în mod independent. Acest tip de arhitectură este utilizat în mod obișnuit în procesoarele multi-core și în sistemele de calcul distribuite.

Exemple de computere și sisteme care utilizează arhitectura MIMD:

1. **Procesoare Intel Multi-Core:** - Multe procesoare Intel, cum ar fi cele din seria Intel Core i și seria Intel Xeon, folosesc arhitectura MIMD cu mai multe nuclee (Intel Core i7, Intel Xeon E5, seria E7).
2. **Procesoare AMD Multi-Core:** - Procesoarele AMD, inclusiv seriile Ryzen și EPYC, utilizează arhitectura MIMD cu mai multe nuclee (AMD Ryzen 7, seria AMD EPYC 7000).
3. **NVIDIA CUDA:** - Arhitectura CUDA a NVIDIA permite procesarea paralelă MIMD pe GPU-urile lor (Graphics Processing Units). Acesta este utilizat pe scară largă pentru simulări științifice.

Exemple MIMD

4. **Sisteme de calcul eterogene:** - Sistemele care combină procesoare și GPU-uri, cum ar fi sistemele CUDA de la NVIDIA sau sistemele eterogene de la AMD, utilizează eficient arhitectura MIMD pentru sarcini de procesare paralelă.
5. **Clustere de calcul distribuite:** - Clusterele de calcul de înaltă performanță constau adesea din mai multe computere conectate împreună pentru a rezolva probleme complexe folosind arhitectura MIMD (clustere Beowulf, clustere Hadoop).
6. **Supercalculatoare:** - Supercomputere precum seria Blue Gene de la IBM și supercalculatoarele Cray folosesc arhitectura MIMD pentru a aborda simulări științifice și de inginerie la scară largă.
7. **Instanțe de cloud computing:** - Mulți furnizori de cloud computing oferă mașini virtuale (VM) și instanțe cu mai multe nuclee CPU, permițând utilizatorilor să valorifice paralelismul MIMD pentru diferite sarcini de lucru.

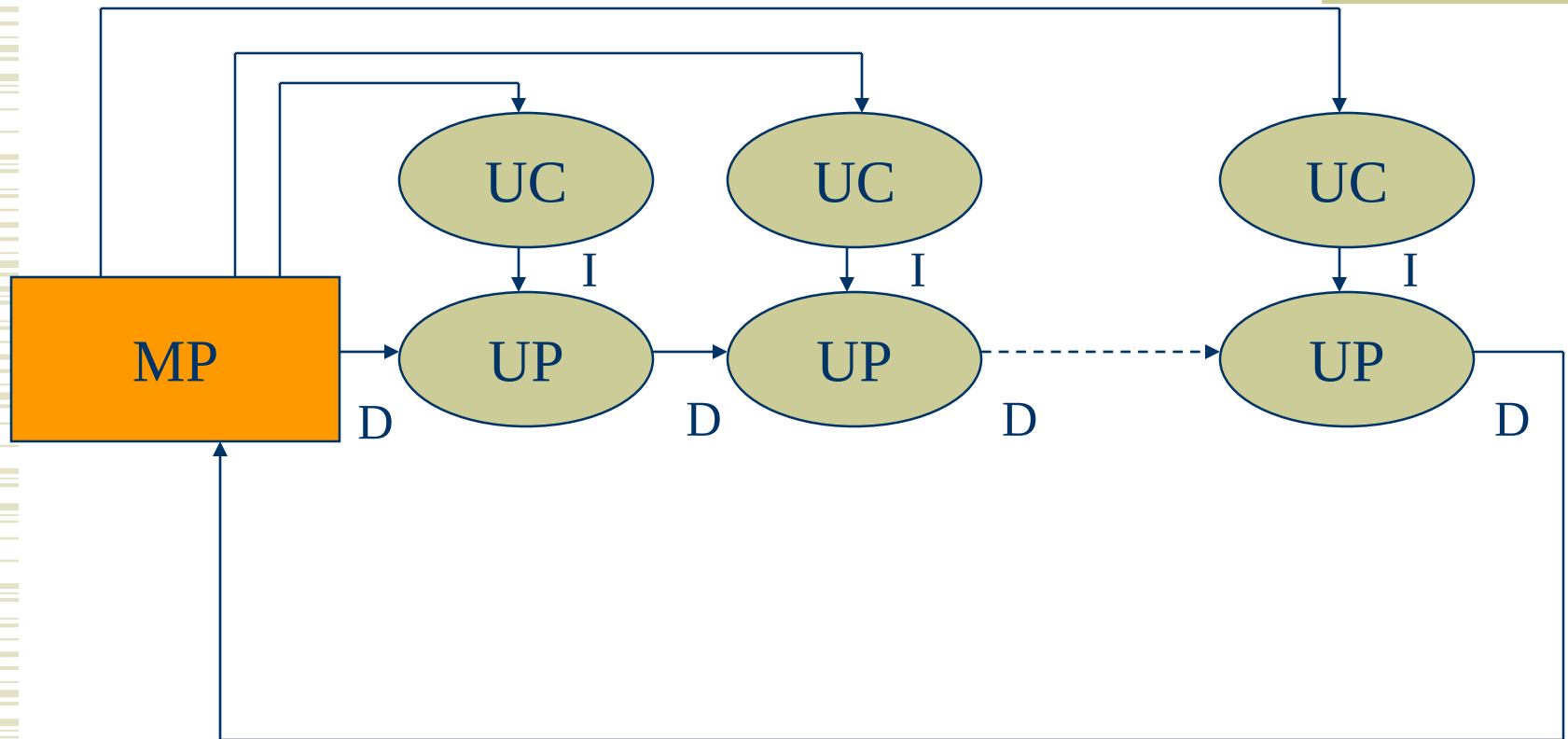
Exemple MIMD

8. Stații de lucru high-end: - Stațiile de lucru de ultimă generație pentru sarcini precum randarea 3D, editarea video și calculul științific au adesea procesoare cu mai multe nuclee, valorificând arhitectura MIMD.

9. Clustere pentru cercetare științifică: - Cercetarea științifică necesită adesea procesarea MIMD pentru sarcini precum modelarea moleculară, simulări meteorologice și secvențierea genomului, folosind grupuri de computere.

Arhitectura MIMD este utilizată pe scară largă în sistemele de calcul moderne pentru a îmbunătăți performanța și eficiența prin împărțirea sarcinilor între mai multe procesoare sau nuclee. Fiecare procesor sau nucleu poate executa propriul set de instrucțiuni pe date separate, făcându-l potrivit pentru sarcini care necesită un grad ridicat de paralelism și procesare concomitentă.

MISD (Multiple Instruction, Single Data)



MISD (Multiple Instruction, Single Data)

MISD (Multiple Instruction, Single Data) este o arhitectură de procesare paralelă mai puțin obișnuită în comparație cu SIMD și MIMD. În MISD, mai multe unități de procesare sau nuclee execută instrucțiuni diferite pe același set de date. Arhitectura MISD este rar folosită în aplicații practice de calcul și este adesea considerată mai mult un concept teoretic. Ca rezultat, există exemple limitate din lumea reală de sisteme MISD.

Un exemplu ipotetic pentru a ilustra conceptul poate fi unul folosit pentru sistemele tolerante la erori. În unele sisteme de calcul cu toleranță la erori, mai multe unități de procesare redundante pot funcționa într-un mod MISD pentru a spori fiabilitatea. Aceste sisteme au ca scop detectarea și corectarea erorilor care pot apărea în timpul procesării datelor. Iată cum ar putea funcționa: presupunem că dispunem de un sistem critic de control aerospațial responsabil de asigurarea siguranței unei nave spațiale în timpul unei misiuni. Acest sistem este conceput pentru a procesa datele senzorilor și a lua decizii pe baza acestor date.

MISD (Multiple Instruction, Single Data)

- 1. Redundanță modulară triplă (TMR):** Aceasta este o tehnică obișnuită tolerantă la erori. Sunt utilizate trei unități de procesare identice (A, B și C). Fiecare unitate rulează același set de instrucțiuni, dar o fac independent.
- 2. Sursă unică de date:** Toate cele trei unități de procesare primesc aceleași date de la senzori, reprezentând un singur set de date.
- 3. Comparatie și vot:** După prelucrarea independentă a datelor, rezultatele fiecărei unități de procesare sunt comparate. Dacă orice unitate produce un rezultat diferit de celelalte două, se folosește un mecanism de vot pentru a determina rezultatul corect. De exemplu, dacă unitățile A și B produc aceeași ieșire, dar unitatea C diferă, sistemul va considera rezultatul de la A și B ca fiind corect.
- 4. Detectarea și corectarea erorilor:** Dacă este detectată o eroare într-una dintre unitățile de procesare (de exemplu, unitatea C produce un rezultat incorect), sistemul poate lua măsuri corective. Poate dezactiva unitatea defectă (C) și poate continua să funcționeze folosind rezultatele celor două unități corecte (A și B).

MISD (Multiple Instruction, Single Data)

În acest scenariu ipotetic, arhitectura MISD este utilizată pentru a îmbunătăți toleranța la erori. Deși nu este o utilizare convențională a MISD, demonstrează conceptul de instrucțiuni multiple (unități de procesare care rulează cod independent) care acționează asupra unui singur set de date (intrări ale senzorului) pentru a obține fiabilitatea și toleranța la erori.

Este important de reținut că MISD nu se găsește în mod obișnuit în aplicațiile de calcul tipice. Majoritatea sistemelor de procesare paralelă din lumea reală utilizează arhitecturi SIMD (Single Instruction, Multiple Data) sau MIMD (Multiple Instruction, Multiple Data), deoarece sunt mai practice și versatile pentru o gamă largă de sarcini.