

1. Bevezetés és alapfogalmak

1.1 Adat és információ

- **Adat:** a világ jelenségeinek leírása számokkal, szöveggel, képekkel vagy mérésekkel.
 - *Példa: **egy szenzor 22,5 °C hőmérsékletet mér → ez adat.** * Információ: **olyan közlés, amely csökkenti a bizonytalanságot, és döntést tesz lehetővé.** Példa: „**A gép túlhevült, mert a hőmérséklet 95 °C**” → ez információ. * Claude Shannon (1939): az információ a bizonytalanság (entrópia) csökkentése. * Bit: a legkisebb információegység, amely két állapotot különböztet meg (0 vagy 1). === 1.2 **Analóg és digitális jelek** === * Analóg jel: folytonos, tetszőleges értéket vehet fel. Példa: **higanyos hőmérő szintje.** * Digitális jel: **diszkrét, előre meghatározott értékeket vesz fel.** Példa: **digitális hőmérő kijelzője.** * A számítógépek digitálisak, mert: - **zajállóbbak,** - **könnyebben feldolgozhatóak,** - **egyszerűbb tárolás és továbbítás.** <mermaid> graph LR A[Valós jelenség] -> B[Analóg jel] A -> C[Digitális jel] B ->|folytonos| D¹⁾ C ->|diszkrét| E²⁾ </mermaid> === 1.3 **Neumann-elv** === Neumann János (1946) öt alapelve: * **Központi vezérlőegység (CPU) irányítja a működést.** * **Programok és adatok közös memóriában tárolódnak.** * **Bináris adatrepresentáció.** * **Utasítás-végrehajtási ciklus: beolvasás → értelmezés → végrehajtás.** * **Soros feldolgozás (egyszerre egy utasítás).** <mermaid> flowchart TD CPU[CPU] -> MEM[Memória] CPU -> IO[I/O perifériák] MEM -> CPU IO -> CPU </mermaid> === 1.4 **Turing-gép** === Alan Turing (1936) megalkotta a Turing-gép modellt. Elemei: * **Szalag (memória): adatok és program.** * **Olvasó/író fej: mozog a szalagon, adatot olvas vagy ír.** * **Vezérlőegység: meghatározza, milyen művelet történjen.** A Turing-gép az algoritmusok elméleti alapja → minden mai számítógép működését leírja. <mermaid> flowchart LR S[Szalag memória] -> F[Olvasó/író fej] F -> V[Vezérlőegység] V -> F F -> S </mermaid> === **Mérnöki alkalmazási példák** === **Adat: szenzor 0,01 mm eltérést mér a gyártás során. Információ: az eltérés nagyobb, mint a megengedett 0,005 mm → selejt.** **Digitális feldolgozás: a PLC a mérést kiértékeli és leállítja a gépet.** **Neumann-elv alkalmazása: a PLC is CPU + memória + I/O elven működik.** ===== 2. **Hardver alapjai** ===== A számítógép hardvere a fizikai részegységekből áll, amelyek együtt biztosítják az adatok feldolgozását, tárolását és megjelenítését. === 2.1 **Központi feldolgozó egység (CPU)** === * **Az utasítások végrehajtásáért felelős "agy".** * **Fő részei:** - **ALU (Aritmetikai-logikai egység): számításokat és logikai műveleteket végez.** - **Regiszterek: a leggyorsabb tárolók, ideiglenes adatokhoz.** - **Vezérlőegység: irányítja a teljes működést.** * **Modern CPU-k többmagosak (quad-core, octa-core), párhuzamos feldolgozásra képesek.** <mermaid> flowchart TD A[CPU] -> B[ALU] A -> C[Vezérlőegység] A -> D[Regiszterek] B ->|Számítások| A C ->|Irányítás| A D ->|Adatok| A </mermaid> === 2.2 **Memóriahierarchia** === * **A CPU különböző sebességű és méretű memóriákat használ:** - **Regiszterek - leggyorsabb, nagyon kicsi.** - **Cache (L1, L2, L3) - gyorsítótár a CPU közelében.** - **RAM - központi memória, ideiglenes adatokhoz.** - **Háttértár (SSD, HDD, NVMe) - tartós tárolás.** * **Elv: minél közelebb van a CPU-hoz, annál gyorsabb, de kisebb a kapacitása.** <mermaid> graph TD R[Regiszterek] -> C1[L1 Cache] -> C2[L2 Cache] -> C3[L3 Cache] -> M[RAM] -> H[Háttértár] </mermaid> === 2.3 **Buszrendszerek** === * **A számítógép részegységei buszokon keresztül kommunikálnak.** * **Fő busztípusok:** - **Adatbusz - adatokat visz a komponensek**

között. - Címbusz - meghatározza, honnan hova kerül az adat. - Vezérlőbusz - irányító jeleket továbbít. * Modern buszok: PCI Express, USB-C, Thunderbolt, NVMe.

<mermaid> flowchart LR CPU[CPU] ->|Adatbusz| MEM[Memória] CPU ->|Címbusz| MEM CPU ->|Vezérlőbusz| IO[I/O perifériák] </mermaid> === 2.4

Példa: Raspberry Pi felépítése === * SoC (System-on-Chip): CPU + GPU + memória vezérlő egyetlen chipben. * GPIO (General Purpose I/O): mérnökök által használt szenzor- és aktuátorvezérlés. * Beépített interfészek: HDMI, USB, Ethernet, kamera csatlakozó.

<mermaid> flowchart TD SOC[System-on-Chip] -> CPU1[CPU magok] SOC -> GPU[Grafikus egység] SOC -> MEMV[Memóriavezérlő] SOC -> GPIO[GPIO csatlakozók] SOC -> USB[USB portok] SOC -> NET[Ethernet/WiFi] SOC -> HDMI[HDMI kimenet] </mermaid> ===== 3.

Számítógépkategóriák és architektúrák ===== A számítógépek többféle kategóriába sorolhatók teljesítményük, méretük és felhasználási területük alapján. Emellett különböző architektúrák léteznek (x86, ARM, RISC-V, SoC), amelyek a mérnöki alkalmazások szempontjából is fontosak. === 3.1

Számítógépkategóriák === * Mikroszámítógép: egyszerű, beágyazott vezérlők (pl. mikrokontroller, Arduino). * Személyi számítógép (PC): általános felhasználásra, mérnöki tervezéshez, szimulációhoz. * Munkaállomás: nagy teljesítményű PC, gyakran erős GPU-val → CAD, FEM, szimuláció. * Szerver: adatfeldolgozás, hálózati szolgáltatások, felhő alapú számítás. * Szuperszámítógép: extrém számítási igényekhez (pl. időjárás-modellezés, molekuláris dinamika).

<mermaid> graph TD A[Mikroszámítógép] -> B[PC] B -> C[Munkaállomás] C -> D[Szerver] D -> E[Szuperszámítógép] </mermaid> === 3.2

Architektúrák === * x86 - Intel és AMD processzorok, PC-k és szerverek alapja. * ARM - energiatakarékos, mobil eszközökben és beágyazott rendszerekben. * RISC-V - nyílt forrású architektúra, gyorsan fejlődő kutatási és ipari terület. * SoC (System-on-Chip) - CPU, GPU, memóriavezérlő, kommunikációs egységek egyetlen chipben.

<mermaid> flowchart TD X[x86 - PC/Szerver] ->|Teljesítmény| P[Munkaállomás] A[ARM - Mobil/Beágyazott] ->|Energiatakarékosság| M[Okostelefonok, IoT] R[RISC-V - Nyílt ISA] ->|Kísérletezés| K[Kutatás, ipar] S[SoC - System on Chip] ->|Integráció| I[Mobil, Raspberry Pi, IoT] </mermaid> === 3.3

Példák mérnöki alkalmazásokban === * Mikrokontroller: egyszerű vezérlési feladatok (pl. motor szabályozás, szenzoradatok gyűjtése). * Munkaállomás: CAD tervezés, végelelemes szimuláció (FEM). * Szerver: ipari adatgyűjtés, gyártási adatok feldolgozása. * Szuperszámítógép: komplex mérnöki számítások (áramlástan, anyagszerkezet szimuláció).

<mermaid> flowchart LR MC[Mikrokontroller] ->|Vezérlés| S1[Szenzor/aktuátor] WS[Munkaállomás] ->|Számítás| FEM[FEM szimuláció] SR[Szerver] ->|Adatfeldolgozás| DB[Gyártási adatok] SC[Szuperszámítógép] ->|Komplex szimuláció| CFD[Áramlástan, anyagtudomány] </mermaid> ===== 4.

Perifériák és interfészek ===== A perifériák a számítógéphez csatlakozó eszközök, amelyek adatbevitelre, megjelenítésre, adattárolásra vagy kommunikációra szolgálnak. Az interfészek biztosítják az összeköttetést a központi egység és a perifériák között. === 4.1

Input (adatbevitel) === * Billentyűzet, egér, érintőképernyő. * Szenzorok (hőmérséklet, nyomás, kamera, mikrofon). * Vonalkód- és QR-olvasó, ipari mérőeszközök.

<mermaid> graph TD IN1[Billentyűzet] -> CPU IN2[Egér] -> CPU IN3[Érintőképernyő] -> CPU IN4[Szenzorok] -> CPU IN5[Videokamera] -> CPU </mermaid> === 4.2

Output (eredménykivitel) === * Monitor, projektor, VR szemüveg. * Nyomtató, 3D nyomtató. * Hangszóró, ipari kijelzők.

<mermaid> graph TD CPU ->

OUT1[Monitor] CPU -> OUT2[Nyomtató] CPU -> OUT3[3D nyomtató] CPU -> OUT4[Hangszóró] </mermaid> === 4.3 Input/Output kombinált eszközök === * Hálózati kártya (Ethernet, WiFi). * Hangkártya (mikrofon + hangszóró). * Érintőképernyő (adatbevitel + megjelenítés). * USB eszközök (pendrive, külső HDD). <mermaid> flowchart LR CPU ↔ NET[Hálózati kártya] CPU ↔ SOUND[Hangkártya] CPU ↔ TOUCH[Érintőképernyő] CPU ↔ USB[USB eszköz] </mermaid> === 4.4 Modern interfészek === * USB-C / USB 3.2 / USB4 - univerzális, gyors adatátvitel és energiaellátás. * Thunderbolt - nagy sávszélesség, külső GPU és kijelzők támogatása. * HDMI / DisplayPort - digitális hang- és videóátvitel. * Ethernet / WiFi / Bluetooth - hálózati kapcsolatok. * I²C, SPI, CAN busz - ipari és beágyazott rendszerek szenzor- és vezérlőcsatolói.

<mermaid> graph TD CPU -> USB[USB-C / USB4] CPU -> TB[Thunderbolt] CPU -> HDMI[HDMI / DisplayPort] CPU -> ETH[Ethernet] CPU -> WIFI[WiFi / Bluetooth] CPU -> I2C[I²C / SPI / CAN] </mermaid> ===== 5. Adattárolás és memória ===== A számítógép működéséhez szükség van gyors, átmeneti és tartós adattárolásra is. Az eltérő tárolók különböző sebességűek, kapacitásúak és feladatúak. === 5.1 Memórahierarchia === * Regiszterek - leggyorsabb, közvetlenül a CPU-ban, nagyon kicsi méret. * Cache (L1, L2, L3) - gyorsítótár, a CPU közelében, kis méret, nagy sebesség. * RAM (Random Access Memory) - központi memória, ideiglenes adattárolás, áramtalanításakor törlődik. * Háttértár - tartós tárolás: SSD, HDD, NVMe. <mermaid> graph LR R[Regiszterek
pár bájtt] -> C1[L1 Cache
~32-64 KB] C1 -> C2[L2 Cache
~256 KB - 1 MB] C2 -> C3[L3 Cache
~4-64 MB] C3 -> RAM[RAM
~8-64 GB] RAM -> SSD[SSD/NVMe
~256 GB - 4 TB] SSD -> HDD[HDD
~1-20 TB] </mermaid> === 5.2 RAM típusai === * DRAM (Dynamic RAM) - olcsóbb, frissítést igényel → fő memória. * SRAM (Static RAM) - gyors, drága → cache. * SDRAM, DDR, DDR4, DDR5 - modern, szinkronizált RAM-típusok. === 5.3 Háttértárak === * HDD (merevlemez): mágneses elven működik, olcsó, nagy kapacitás, de lassú. * SSD (Solid State Drive): flash-alapú, gyorsabb, nincs mozgó alkatrész. * NVMe SSD: PCIe buszon keresztül → extrém sebesség (GB/s). * Optikai tárolók: CD, DVD, Blu-ray - ma inkább archiválásra. * Flash eszközök: pendrive, SD-kártya. <mermaid> flowchart LR HDD[HDD
olcsó, lassú] ->|Kiváltja| SSD[SSD
gyorsabb] SSD -> NVMe[NVMe SSD
PCIe alapú] SSD -> Flash[Flash tárolók
pendrive, SD-kártya] HDD -.→ Archive[Optikai lemezek
CD/DVD/Blu-ray] </mermaid> === 5.4 Virtuális memória === * Az operációs rendszer a háttértár egy részét RAM kiegészítésére használja. * Lehetővé teszi, hogy több program fusson egyszerre, mint amennyi a fizikai RAM-ban elfér. * Hátránya: lassabb, mert a háttértár sebessége korlátozó tényező. <mermaid> flowchart TD CPU -> RAM RAM ->|Ha megtelik| VM[Virtuális memória SSD/HDD] </mermaid> === 5.5 Mérnöki alkalmazási példa === * FEM szimuláció: RAM korlátozhatja a futtatható modell méretét. * Nagy adatgyűjtés (pl. szenzorhálózat): SSD szükséges a gyors íráshoz. * Beágyazott rendszerek: kis méretű RAM és flash → optimalizált programozás kell. ===== 6. Párhuzamos feldolgozás és gyorsítók ===== A számítógépek teljesítményének növelését ma már nem elsősorban az órajel emelése, hanem a párhuzamos feldolgozás és a speciális gyorsítók biztosítják. === 6.1 Többmagos processzorok === * A CPU több maggal rendelkezik → egyszerre több utasítást hajthat végre. * Példák: - Dual-core, Quad-core, Octa-core CPU-k. - Mobil eszközökben: heterogén architektúrák (pl. ARM big.LITTLE). * Előny: több szál (thread) futtatható egyidőben. <mermaid> graph TD CPU[Processzor] -> C1[Mag 1] CPU -> C2[Mag 2] CPU -> C3[Mag 3] CPU -> C4[Mag 4] C1 & C2 & C3 & C4 -> OS[Operációs

**rendszer
többszálúság] </mermaid> === 6.2 Párhuzamos feldolgozási technikák ===** * Pipeline (csővezetékes feldolgozás) - **egy utasítás több részfeladatra oszlik, amelyek átfedésben hajthatók végre.** * SIMD (Single Instruction, Multiple Data) - **egy utasítással több adaton művelet (pl. vektorműveletek).** * MIMD (Multiple Instruction, Multiple Data) - **több mag különböző programrészeket futtat párhuzamosan.** <mermaid> flowchart LR A[Pipeline] ->|lépések egymásra tolva| B[Gyorsabb végrehajtás] C[SIMD] ->|egy utasítás
több adaton| B D[MIMD] ->|különböző utasítások
különböző adatokon| B </mermaid> === 6.3 GPU-k (Grafikus processzorok) === * **Eredetileg grafikai számításokra (3D, játékok).** * **Ma: általános célú számításokra is (GPGPU).** * **Több ezer feldolgozó egység → kiváló párhuzamosítás nagy adathalmazokon.** * **Használat: gépi tanulás, képfeldolgozás, FEM/CFD szimulációk gyorsítása.** <mermaid> graph TD GPU[GPU] -> C1[1000+ mag] GPU -> AI[AI és ML számítások] GPU -> SIM[Szimulációk] GPU -> GFX[Grafika és 3D megjelenítés] </mermaid> === 6.4 Egyéb gyorsítók === * FPGA (Field Programmable Gate Array): **újraprogramozható hardver, ipari alkalmazásokban gyors, testreszabott feldolgozásra.** * TPU (Tensor Processing Unit): **mesterséges intelligenciára optimalizált chip.** * NPU (Neural Processing Unit): **neurális hálókat futtatására mobil eszközökben.** <mermaid> flowchart TD FPGA ->|Rugalmasság| Ipari[Ipari vezérlés] TPU ->|AI felhő| Cloud[Felhőszolgáltatások] NPU ->|Mobil AI| Mobile[Okostelefonok, IoT] </mermaid> === 6.5 Mérnöki alkalmazási példák === * FEM szimuláció - **CPU + GPU együttműködésével gyorsabb számítás.** * Képfeldolgozás - **GPU gyorsítással valós idejű hibadetektálás gyártásban.** * FPGA - **ipari robotvezérlés, valós idejű adatfeldolgozás.** * TPU/NPU - **prediktív karbantartás, IoT érzékelők adatfeldolgozása.** ===== 7. Ipari és mérnöki alkalmazások ===== **A számítástechnika nemcsak irodai környezetben, hanem ipari és mérnöki területeken is kulcsfontosságú. A mérnökök számára a legfontosabb alkalmazási területek: beágyazott rendszerek, ipari vezérlők, IoT, valamint nagy teljesítményű szimulációk.** === 7.1 Beágyazott rendszerek === * **Speciális célú, kisméretű számítógépek.** * **Tipikus eszközök: mikrokontroller, Arduino, ESP32, Raspberry Pi.** * **Alkalmazások: - Szenzoradatok gyűjtése és feldolgozása. - Motorok, aktuátorok vezérlése. - Egyszerű ipari automatizálási feladatok.** <mermaid> flowchart TD S[Szenzorok] -> MCU[Mikrokontroller] MCU -> ACT[Motorok / aktuátorok] MCU -> NET[Kommunikáció pl. \WiFi, Bluetooth\]) </mermaid> === 7.2 PLC-k és ipari vezérlőrendszerek === * PLC (Programmable Logic Controller): **ipari gépek és folyamatok vezérlésére kifejlesztett számítógép.** * **Jellemzők: - Robusztus, megbízható, folyamatos üzemre tervezett. - I/O csatlakozások ipari szabvány szerint (pl. 24V, relékimenet). - Egyszerű logikai programozás (létra diagram).** * **Alkalmazások: gyártósorok, robotok, szállítószalagok.** <mermaid> graph LR SENSOR[Szenzorok] -> PLC[PLC] PLC -> MOTOR[Motorvezérlés] PLC -> HMI[Kezelőpanel / HMI] PLC -> NET[Hálózati kommunikáció] </mermaid> === 7.3 IoT és edge computing === * IoT (Internet of Things): **szenzorok hálózata, amely adatokat gyűjt és továbbít.** * **Edge computing: adatfeldolgozás a hálózat szélén, a szenzor közelében → csökkenti a késleltetést.** * **Példák: - Okosgyár (smart factory). - Prediktív karbantartás (hibák előrejelzése). - Energiagazdálkodás.** <mermaid> flowchart LR S1[Szenzor] -> EDGE[Edge eszköz] EDGE -> CLOUD[Felhő] CLOUD -> USER[Felhasználó / Mérnök] </mermaid> === 7.4 Nagy teljesítményű számítások (HPC) === * **Mérnöki**

tervezésben és kutatásban szükség van extrém számításokra: - FEM (végelelemes módszer) **szilárdságtanhoz.** - CFD (Computational Fluid Dynamics) **áramlástanhoz.** - **Molekuláris szimulációk, anyagvizsgálatok.** * Ezekhez **munkaállomások és szuperszámítógépek szükségesek, gyakran GPU gyorsítással.** <mermaid> flowchart LR MODEL[3D mérnöki modell] -> FEM[FEM szimuláció] MODEL -> CFD[CFD szimuláció] FEM -> HPC[Munkaállomás / Szuperszámítógép] CFD -> HPC </mermaid> === **7.5 Példák mérnöki gyakorlatból** === * Gépészmérnökök: **3D CAD modellezés, végelelemes analízis.** * Villamosmérnökök: **vezérlőelektronika, PLC programozás, ipari kommunikációs hálózatok.** * Anyagmérnökök: **mikroszkópos képfeldolgozás, szimulációs modellek.** * Mechatronikus mérnökök**: robotvezérlés, szenzorhálózatok, IoT integráció.

1)

∞ érték

2)

0 vagy 1

From:

<https://edu.iit.uni-miskolc.hu/> - Institute of Information Science - University of Miskolc

Permanent link:

https://edu.iit.uni-miskolc.hu/tanszek:oktatas:szamitastechnika:hardver_alapismeretek?rev=1758049581

Last update: 2025/09/16 19:06

