

Számítógépek architektúrák

Architektúrák

Bemutakozom ...

Dr. Vadász Dénes egyetemi docens

vadasz@iit.uni-miskolc.hu

<http://www.iit.uni-miskolc.hu/vadasz>

Informatikai Intézet épülete,
I. emelet, 111. szoba

Miskolci Egyetem

Gépészmérnöki és Informatikai Kar

Informatikai és villamosmérnöki tanszékcsoport

Általános informatikai tanszék

<http://www.iit.uni-miskolc.hu>



A tárgy célja

- **Számítógép (computer)**
 - Egy *programozási nyelv* segítségével leírt *számítási feladat* végrehajtására szolgáló *eszköz*¹
- **Architektúra (felépítés, szerkezet)²**
 - Funkcionális specifikáció az irányultság: specifikáció,
 - Megvalósítási célú irányultság: egységek és kapcsolódásuk.
 - Bármelyik irányultságnál különböző részletezettség
- **A tárgy célja:** általános hardverismeret megszerzése, továbbá felhasználói felületek (parancsértelmezős és grafikus) megismerése. Szemléletmód kialakítás.

Az oktatási módszereink ...

- **Vetített képes előadások ...**
 - <http://www.iit.uni-miskolc.hu/~szkovacs/>
 - Oktatási oldalaimon a GEIAL201N kurzus anyaga
 - itt az előadások képei nyomtatható formában (Acrobat Reader)
 - A Számítógépek, számítógép rendszerek c. jegyzet
- **Az előadásokon a lényeges dolgok kiemelése, fontos definíciók, konvencionális szóhasználat stb.**
- **Laboratóriumi gyakorlatok**
 - Kötelező a látogatásuk
- **Önálló feladatok**

A teljesítés feltételei

- **Aláírás és vizsga**
- **Az aláírás feltételei**
 - A gyakorlatokon aktív jelenlét, az ottani feladatok eredményes elvégzése. Legalább 10 gyakorlatot el kell ismertetni! Köztük a „szerelés” kötelező! Köztük évközi zh. kötelező!
 - **Aláírást csak a tanulmányi időszakban szerezhettek!**
- **A vizsga írásbeli és szóbeli vizsga**
 - Írásbeli: rövid kérdésekre rövid válaszok

Az ütemterv

- **A tárgy honlapján, a tanszéki hirdetőtáblán ...**
- **Témák:**
 - Számítási modellek, architektúrák
 - A felhasználó szemlélete. Szolgáltatások. Parancsnyelvek.
 - A CPU
 - Memória
 - Sínek
 - Eszközök: képernyő, billentyűzet, mutatók
 - Háttértárak, nyomtatók
 - Esettanulmányok

Számítógép történet

- **Tanulmányozzák Katona István A számítógép története c. prezentációját!**
http://www.ektf.hu/mediainf/inf/ktoth/konyvtar/Szamitogep-tortenet_elemei/frame.htm
- **Ebből fontos: a Neumann elvű gép**
- **A Neumann elv röviden:**
 - A számítógép legyen teljesen elektronikus, külön vezérlő- és végrehajtó egységgel.
 - Kettes számrendszert használjon.
 - Az adatok és a programok ugyanabban a belső tárban, a memóriában legyenek.
 - A számítógép legyen univerzális Turing-gép¹ (soros utasítás végrehajtás elve érvényesüljön) (Turing 1936)

Fontos mérföldkövek

Év	Név	Készítő	Megjegyzés
1834	Analytical Engine	Babbage	Az első általános célú számítógép
1936	Z1	Zuse	Első működő, relés technikával
1943	COLOSSUS	Brit kormány	Első elektronikus gép. Szupertitkos
1944	Mark I	Aiken	Első amerikai általános célú (Harward arch.)
1946	ENIAC	Ecker/Mauchley	A modern számítógépek története indul
1948		Neumann	A Neumann elv megszületik
1949	EDSAC	Wilkes	Első tárolt program elvű
1952	IAS	Neumann	A mai géptervezés alapjai
1961	1401	IBM	Üzleti célú, népszerű
1962	7094	IBM	Tudományos számításokhoz
1964	360	IBM	Általános célú

Fontos mérföldkövek

Év	Név	Készítő	Megjegyzés
1964	6600	CDC	Első tudományos célú szuperszámítógép
1965	PDP-8	DEC	Tömegesen elterjedt mikroszámítógép
1970	PDP-11	DEC	Tömegesen elterjedt mikroszámítógép
1974	8080	Intel	8 bites lapkán általános célú gép
1974	CRAY-1	Cray	Első vektoros szuperszámítógép
1978	VAX	DEC	32 bites miniszámítógép, 1 MIPS-es
1981	IBM PC	IBM	Személyi számítógépek kora indul
1985	MIPS	MIPS	RISC korszak indul
1987	SPARC	Sun	SPARC alapú munkaállomások
1990	RS6000	IBM	Az első szuperskalár gép

Az utóbbi évtizedek a számítástechnikában

Az évek	60-as	70-es	80-as	90-es
A paradigma	Kötegelt feldolgozás	Időosztás	Asztali gépek	Hálózatok
Hol?	Számítóközpontban	Terminálszobában	Íróasztalon	Mobil
Az adatok	Numerikus adatok	Szövegek + számok	... + rajzok	Multimédia
Fő cél	Számítások	Hozzáférés	Megjelenítés	Kommunikáció

Az utóbbi évtizedek a számítástechnikában 2

Az évek	60-as	70-es	80-as	90-es
A paradigma	Batch	Time sharing	Desktop	Network
Az interfész	Lyuk-kártya	Billentyűzet + CRT	Lásd és kattints	Kérdezd és mondd
Kapcsolódás	Nincs	Terminál vonalak	LAN	Internet
Tulajdonos	Intézeti sz.központ	Osztályok	Osztályok dolgozói	Mindenki

Szemléletek ...

- Számítógép (computer)
 - Egy *programozási nyelv* segítségével leírt *számítási feladat* végrehajtására szolgáló *eszköz*¹
- Architektúra (felépítés, szerkezet)²
 - Funkcionális specifikáció az irányultság: specifikáció,
 - Megvalósítási célú irányultság: egységek és kapcsolódásuk.
 - Bármelyik irányultságnál **különböző részletezettség**
- A tárgy célja: általános hardverismeret megszerzése, továbbá felhasználói felületek (parancsértelmezős és grafikus) megismerése.
- **Különböző szerepköri szemléletek is vannak ...**

Általános felhasználói látásmód

- **Többnyire felső szint**
- **Grafikus v. parancsnyelvi felhasználói felület**
(ikonok, ablakok, eszközök, fájlok, könyvtárak stb.,
parancs, megnyitás, indítás, kattintás vonszolás stb.)
- **Szolgáltatások**
 - Irodai
 - Kommunikációs
 - Információszerző
 - Védelmi, menedzselő
 - Adott célú alkalmazások

Programozói látásmód

- Közeledünk a konkrétumokhoz ...
- Amit a felhasználó lát, azt a programozó is
- Fejlesztői felület (editorok, make, fordító és taszképítő, debugger stb.)
 - Ezt a szemléletet más tárgyakban oktatjuk

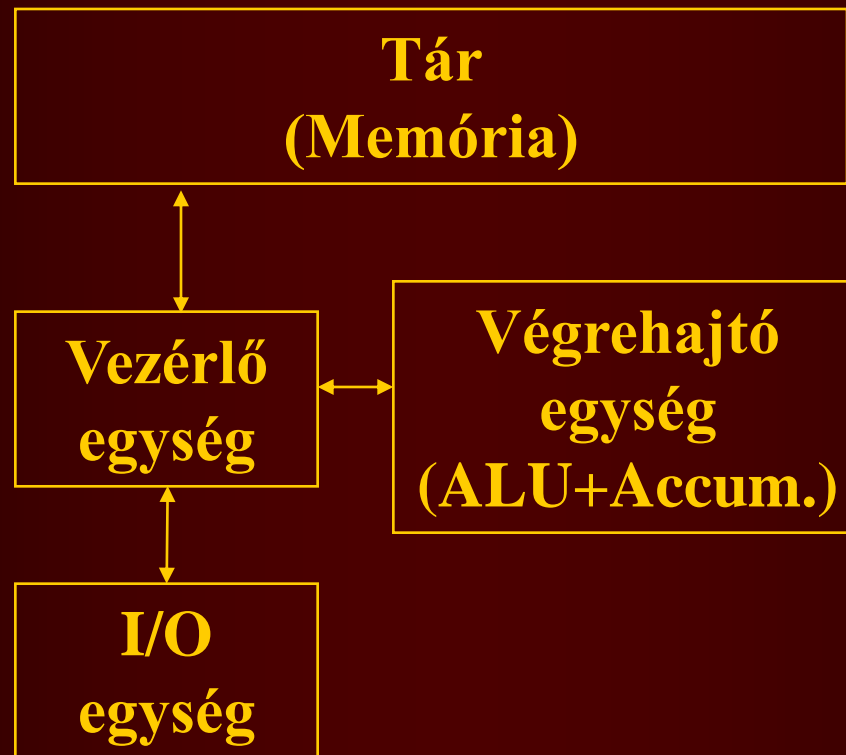
Rendszergazdai látásmód

- Közelebb a valódi géphez
- Az operációs rendszert, szolgáltatásokat, ezek menedzselését ismernie kell

Hardverismeret

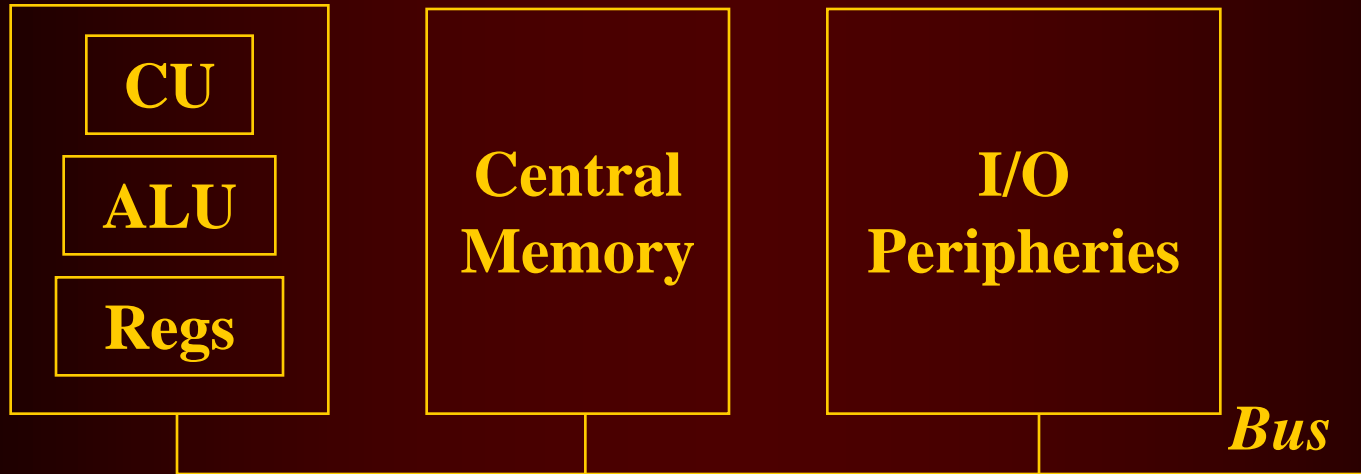
- **Kell-e?**
 - Könnyebb az egyszerű felhasználónak is, ha vannak bizonyos ismeretei
 - Programozónak több ...
 - Rendszergazdának sok ...
 - Hardvereseknek egész sok, villamosmérnökségi szint
- **Figyeljünk fel arra, hogy a egy számítógépet a hardver és szoftver architektúrája együtt határozza meg. Kijelentjük: ez a Neumann elvből származtatható gondolat! (Magyarázzák a származtatást!)**

Az eredeti Neumann gép



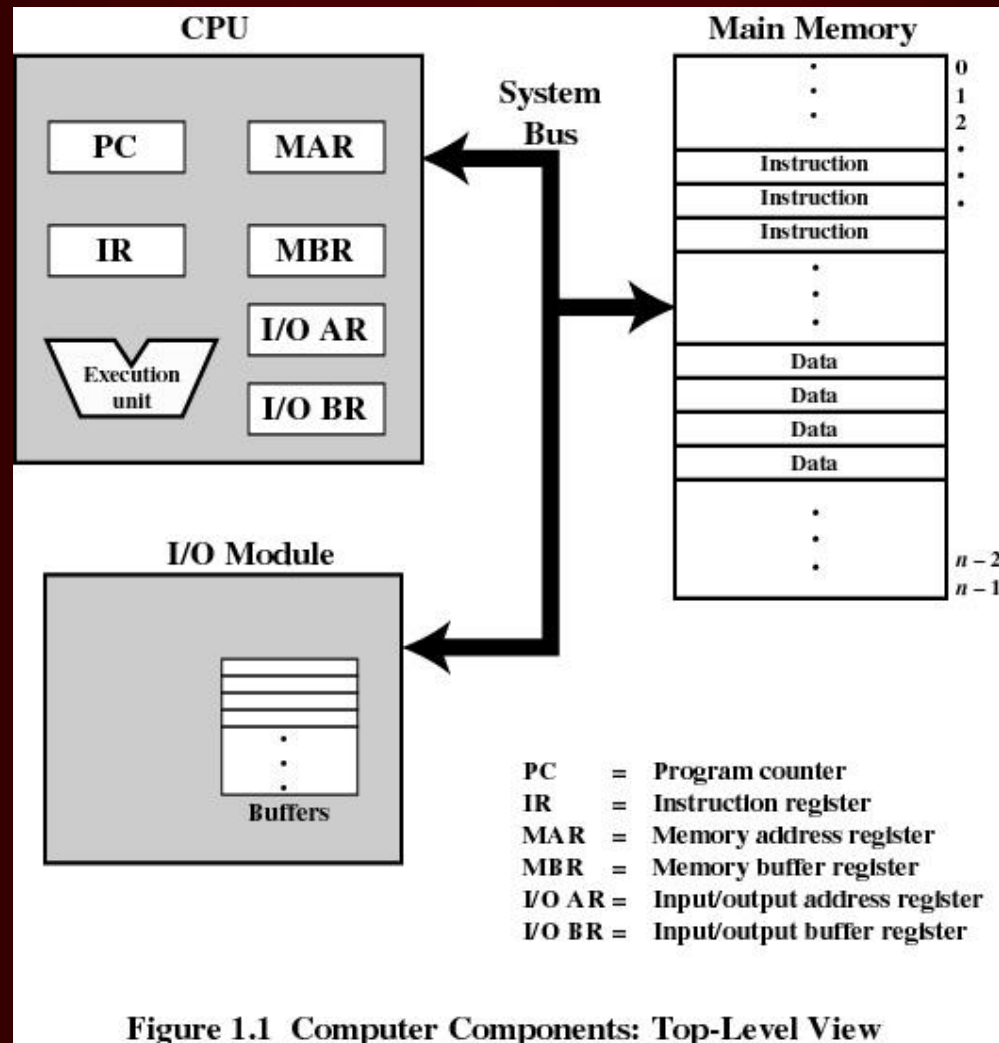
A Neumann architektúra (mai fogalmakkal)

CPU



- **Control Unit:** Vezérlő egység
- **ALU:** Aritmetikai és logikai egység
- **Regs:** Regiszterek
- **Central Memory:** Központi memória, tár
- **I/O Peripherals:** Ki/bemeneti egységek, perifériák
- **Bus:** Sín, adattovábbító áramkörök
- **Central Processing Unit:** Központi feldolgozó egység, processzor

Egy másik funkcionális modell



A központi tár és perifériák

- **A memória**
 - Adatokat (bit, bájt, szó, blokk, mezőkből álló rekord, fájl stb.) és gépi instrukciókat tartalmazó,
 - címezhető cellák (rekeszek) készlete.
 - Valamilyen fizikai hatásra kialakuló állapot, állapotkülönbség (mágnesezettség, töltöttség, feszültség szint, fény törés stb.)
- **A perifériák (ki/bemeneti egységek)**
 - Periféria vezérlő áramkörökkel (controller, adapter) kapcsolódó eszközök (devices).
- **A CPU**
 - A gépi instrukciókat feldolgozó (processzáló) egység, a processzor. Több funkcionális elemből áll (CU, ALU, Regs stb.)

A Neumann gép működése

- A memória rekeszeiben ott vannak a gépi instrukciók (a kód, a program) és az adatok.
- A CPU memóriából felhozza (fetch) a soron következő gépi instrukciót
- A CU elemzi az instrukciót. Értelmezi.
- Ha szükséges, a memóriából felhozza az instrukció operandusát
- Az ALU végrehajtja az instrukciót
- A végrehajtás eredménye a regiszterekbe, esetleg a memória megfelelő rekeszébe kerül
- Folytatódik a soron következő instrukcióval ...

Az állapotterek

- Vegyük észre e következő absztrakciókat
 - A „soron következő instrukció” koncepció egy *instrukció folyam* (instruction stream) képzetet ad
 - Ezen folyamon egy mutató mutathatja a soron következő elemet. Ez a mutató a programszámláló regiszter (PC: Program Counter, IP: Instruction Pointer)
 - Az instrukció folyam instrukcióinak készlete *vezérlési állapotteret* határoz meg. Ebből egy állapotot az ad meg, hogy az *i*-edik lépésben mely instrukciót hajtja végre a processzor
 - Létezik számunkra egy *adat folyam* is: a memória rekeszeknek és regisztereknek az a sora, mely az egymás utáni instrukciókban operandusként szerepelnek.
 - Az adat folyam elemei *adat állapotteret* határoznak meg.
 - Egy-egy instrukció végrehajtása *állapot változást* hoz.

Kis előleg a félév végéről ... Kérdések

- **Csakis egy IS és DS képzelhető el? Dehogy!**
 - **SIMD: ugyanazt az instrukció folyamat változó (különböző, többszörös) adatfolyamon el tudjuk-e képzelni?**
 - **Hol lehetne ezt?**
- **Mi képezi a vezérlési állapotteret?**
 - **Mi a vezérlési állapot (az állapotter egy eleme)?**
- **Adat állapotter? Ennek egy eleme?**
- **Állapot változás? Mi okozza?**

Az instrukció folyam végrehajtása

- A program (az instrukció folyam, a kód) futása **állapot átmenetek láncolatát hozza.**
- A vezérlés állapot átmenet láncolat kulcsjellemezője a programszámláló regiszter egymás utáni értékei: **a vezérlés menete (flow of control).¹**
(thread, szál fogalom)
- A Neumann elvű gépekre jellemző ez az *egy vezérlés menet* (Single Instruction Stream) *egy adat folyamon* (on Single Data Stream): SISD²

Állapotfüggés

- **Hogy egy instrukció milyen állapotba „billent”, az függ az előző állapottól ...**
- **Állapot-érzékeny a Neumann gép**
- **El tudunk-e képzelni állapot-független gépet?**

A Neumann gép és az imperatív programozás

- **Az imperatív nyelvekkel a vezérlés menetét manipuláljuk**
 - **Tedd ezt ezzel, utána ezt stb.**
 - **FORTRAN, C, Pascal, Basic**
- **Ezért az imperatív programozási paradigma jól megfelel a Neumann gépnek**

Hiba és eseménykezelés a Neumann gépen

- **Az eseményekhez kezelő (handler) instrukciófolyam tartozik**
- **Az esemény bekövetkeztekor a vezérlés menete ugorjon a kezelőre (a CPU állapot, a kontextus „lementése” után)**
- **A kezelés után (ha lehetséges) a vezérlés menete térjen vissza a „normál” instrukció folyamra, folytatódjon a processz futása (az állapot, a kontextus vissza-emelése után persze).**
- **Összegezve: a hiba és eseménykezelés a vezérlés menetének manipulálásával történik.**

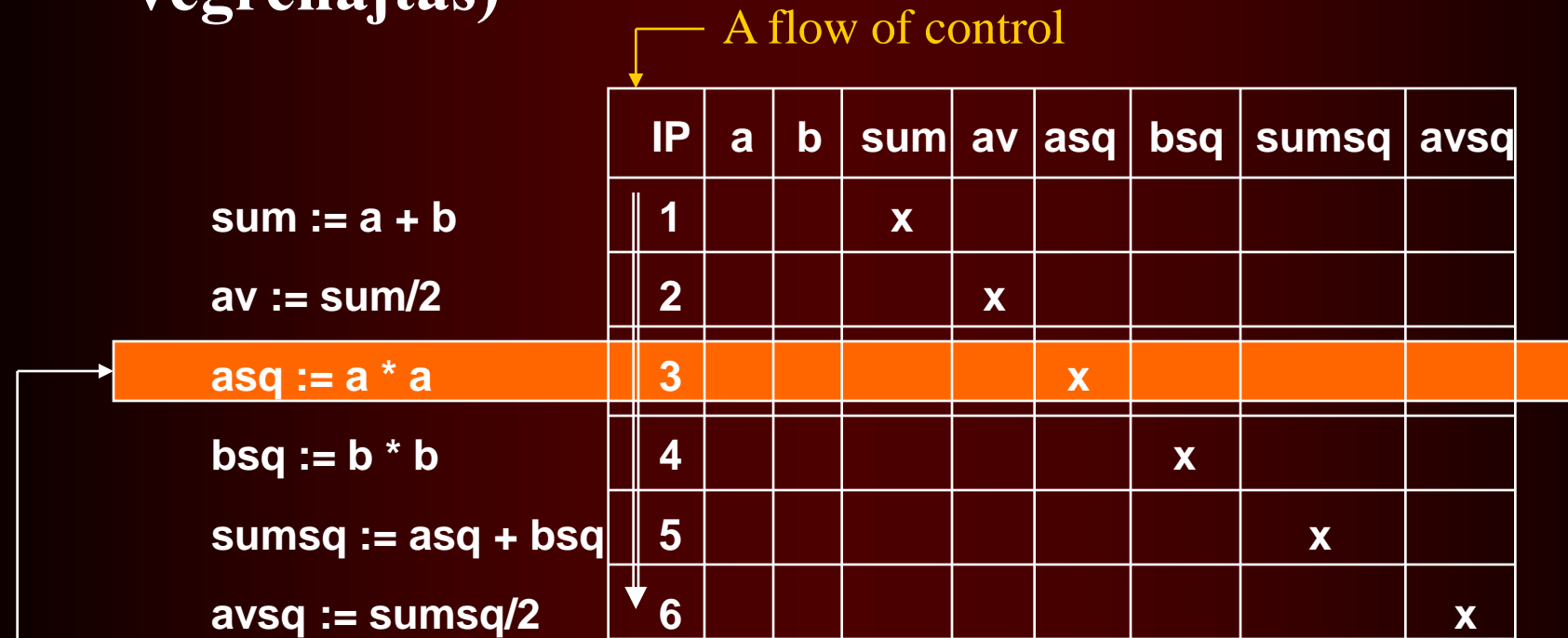
Egy más elvű gép: adatfolyam gép

A Dataflow Machine (kontrasztként) ideája:

- **szeparált processzorok minden operációra (operáció lehet: aritmetikai, logikai, függvényhívás stb.)**
- **Az operációk (processzorok) várnak, míg operandusuk értéke előáll, utána adják eredményüket.**
- **A processzorok (operációk) függetlenek. A legkorábbi lehetséges pillanatban adják az eredményüket.**
- **Az operációk végrehajtásának sorrendje az adatfolyamból adódik.**

Példa: adott a és b, kiszámítandó átlaguk és négyzeteik átlaga

Egy virtuális Neumann gép (szekvenciális végrehajtás)



Az állapotvektor a 3. absztrakt instrukció végrehajtása

után

Gyakran használt fogalmak

- **Virtualitás, virtuális (látszólagos)**
 - Valami, ami valóságosan nem létezik, de mégis úgy használhatjuk, mintha létezne
 - Pl. virtuális meghajtó, emulált terminál, virtuális gép stb.
- **Transzparencia, transzparens (átlátszó)**
 - Valami, ami ott van, de nem látjuk, nem vesszük észre, nem kell törődni vele, mert átlátszó. (Pl. az előbbi virtuális diszk-meghajtót a hálózaton át egy fájlserver biztosítja, akkor a hálózat transzparens, nem kell vele törődni.)
 - Világos, tiszta, nem titkolt ...

Példa: adott a és b, kiszámítandó átlaguk és négyzeteik átlaga

Neumann gép (szekvenciális végrehajtás)

IP		800	804	808	80C	810	814
600	a						
604	b						
608	sum	x					
60C	av		x				
610	asq			x			
614	bsq				x		
618	sumsq					x	
61C	avsq						x
800	ADD a,b,sum						
804	DIV sum, 2, av						
808	MUL a, a, asq						
80C	MUL b, b, bsq						
810	ADD asq, bsq, sumsq						
814	DIV sumsq, 2, avsq						

Számítógépek © Vadász, 2008.

Ea1 31

Példa: adott a és b, kiszámítandó átlaguk és négyzeteik átlaga

Adatfolyam gép

$sum := a + b$

$av := sum/2$

$asq := a * a$

$bsq := b * b$

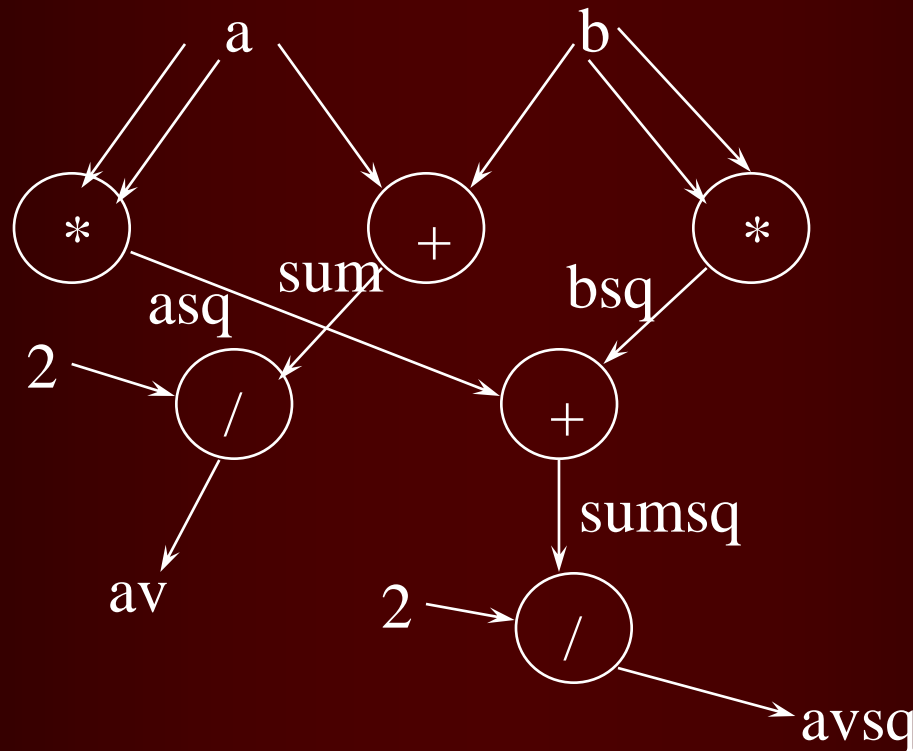
$sumsq := asq + bsq$

$avsq := sumsq/2$

Hibakezelés:

explicit

hibaértékekkel



A nyilak: neves, vagy névnélküli értékek.

A körök: az operációkhoz rendelt processzorok.

Példa: adott a és b, kiszámítandó átlaguk és négyzeteik átlaga

- **Adatfolyam gép**
 - Van 6 processzor a 6 operációhoz,
 - nincsenek “változók” (név, érték, típus, cím),
 - névvel ellátott értékek vannak (a,b,asq,bsq,sum stb).
 - A neves értékek nem definiálhatók át! Redefiníció esetén nem tudnák a processzorok, melyik értékre várjanak!
 - A neves értékeknek típusa és *explicit hibaértéke* van! **Ui.** egy processzor mindenképp kell eredményezzen értéket, legföljebb hibás eredményt! Kaphat hibás inputot is.

Számítógép - nyelvek - számítási modell

- **A számítási modellek összetevői**
 - **a számítás alapelemei**
 - **a problémaleírás modellje**
 - a leírás jellege és
 - módszere;
 - **a végrehajtás modellje**
 - a végrehajtási szemantika
 - a végrehajtás kontrollja

A Neumann modell

- Az **alapelemek**: azonosítható entitásokhoz rendelt (típusos) adatok. (Változók, többszörös értékadás)
- **Problémaleírás**
 - procedurális/imperatív (lépésenként megadva ...)
- A **végrehajtás modellje**
 - a **szemantika**: állapotátmenet szemantika
 - a **kontroll**: közvetlen vezérlés (... a vezérlés menete ...)

Az adatfolyam modell

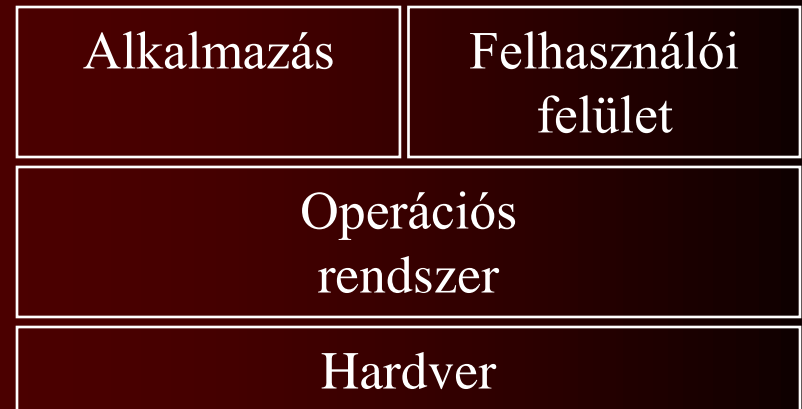
- Az **alapelemek**: azonosítható entitásokhoz rendelt (típusos) adatok. (Egyszeri értékadás)
- **Problémaleírás**
 - deklaratív (az operációk felsorolása pl. függvények használatával ...)
- A **végrehajtás modellje**
 - a **szemantika**: applikatív
 - a **kontroll**: adatfolyam vezérelt

Neumann elvű gép

- **A gép fő részei, követelmények:**
 - ALU, vezérlő egység, memória, perifériák,
 - 2-es számrendszer, elektronikus. Előtte?
- **Tárolt program elv:**
 - A tárban az adatok és a program is. Előtte?
 - Következményei: jók és rosszak.
- **Automatikus működés (közvetlen vezérlés):**
 - program szerint, állapotok, állapotátmenetek, a vezérlés menete, PC/IP szerepe.
- **Babbage Analytical Engine: megfelel?**

Számítógép: hardver és szoftver architektúra

- A legáltalánosabb SW architektúra
- Direkt futtatás, monitor, operációs rendszer
- Az OS fogalma
 - Kiterjesztett gép
 - Erőforrás menedzser



A rétegezethez

(layered architecture)

Egy réteg elrejt az alatta fekvő rétegek részleteit.

Elegendő csak az alattad lévő réteg felületét (interface) ismerni

Operációs rendszer osztályozás

- **Cél szerint: általános, cél**
- **HW “nagyság” szerint: PC, kis, nagy, szuper**
- **Processzorok, processzek, felhasználók száma szerint**
- **Időosztás szerint: szekvenciális, time sharing: kooperatív, beavatkozó**
- **Memóriamenedzselés szerint: valós, virtuális**
- **Fájlrendszer implementáció szerint**

Összefoglalás

- **Bevezetés**
- **Egy kis história ...**
- **A Neumann elvű gép és az**
- **adatfolyam gép.**
- **Számítási modellek**

Számítógépek architektúrák

Architektúrák

VÉGE