Grafikus processzorok általános célú programozása (GPGPU)

Valasek G., Szigeti B., Eichhardt I. és Hajder L.

Eötvös Loránd Tudományegyetem, Informatikai Kar
2025/2026. II. félév

February 19, 2026

Tartalom

1. Adminisztráció
   - Bemutatkozás és elérhetőségek
   - Előadások, gyakorlatok
   - Számonkérés
2. Tematika
3. Számítógép architektúrák gyorsítása
4. Hagyományos számítógép architektúrák fejlődése
5. Hagyományos számítógép architektúrák fejlődése
6. Multimédiás architektúrák
7. GPU architektúrák fejlődése

Tartalom

1. Adminisztráció
   - Bemutatkozás és elérhetőségek
   - Előadások, gyakorlatok
   - Számonkérés
2. Tematika
3. Számítógép architektúrák gyorsítása
4. Hagyományos számítógép architektúrák fejlődése
5. Hagyományos számítógép architektúrák fejlődése
6. Multimédiás architektúrák
7. GPU architektúrák fejlődése

Bemutatkozás és elérhetőségek

- Eichhardt Iván
  - Email: `ivan.eichhardt+gpgpu@gmail.com`
- Szigeti Balázs
  - Email: `szigetibalazs@inf.elte.hu`
  - Szoba: ELTE Déli-tömb 2.704-es szoba
- Valasek Gábor
  - Elérhető: ??? (főleg az iparban dolgozik)
- Hajder Levente
  - Email: `hajder@inf.elte.hu`
  - Szoba: ELTE Déli-tömb 2.704-es szoba

Előadás

- Előadás
  - Heti két óra. Helyszín: Fejér Lipót terem, csütörtök: 16.00-17.30
- Gyakorlat
  - Heti 2 óra három csoportban
    - Grafika labor péntek: 10:15-11:45
    - Grafika labor csütörtök 12:15-13:45
    - Grafika labor csütörtök 14:15-13:45
- A gyakorlat úgyis néha előadás-szerű lesz ☺
  - Ahogy a tananyag megkívánja
- Honlap: `https://cv.inf.elte.hu/index.php/education/gpu-programozas/`

Számonkérés

- Jegyszerzési követelmény: nagyfeladat(ok) bemutatása
  - Feladatkiírások: honlapon, de még bővülhet. Mindegyik feladathoz pontozás tartozik.
  - Feladatválasztás: harmadik hét (február 27.)
  - Megoldási terv, felvett 5 perces videóval (prezentáció): hatodik hétig (max. 10%)
  - Extra pontok: részletek a feladatlista elején
  - Órai aktivitásért is járhatnak extra pontok
- Minimum kritériumok
  - Git verziókövetés
  - Technikai (fejlesztői) dokumentáció
  - Teljesítmény analízis
- Jegy: összpontszám alapján
  - Jeles: 80%, Jó: 70%, Közepes: 60%, Elégséges: 50%

Hol lehet ezt használni?

- Sebesség kritikus a programozási feladatokban
- Felhasználási területek: részecskefizikától a meteorológiai előrejelzéseken át a numerikus optimalizálásig szinte mindenhol. Népszerű alkalmazások:
  - Mélytanulás: objektum-felismerés, nagy nyelvi modellek, ...
  - Kriptovaluta-bányászat
- Az előadók hol használták/használhatnák a tudást:
  - Számítógépes grafika
  - Számítógépes képfeldolgozás
  - Számítógépes 3D látás
  - Geometria/CAD modellezés
  - Matematikai algoritmusok (lineáris algebra)
  - ...

Tartalom

1. Adminisztráció
   - Bemutatkozás és elérhetőségek
   - Előadások, gyakorlatok
   - Számonkérés
2. Tematika
3. Számítógép architektúrák gyorsítása
4. Hagyományos számítógép architektúrák fejlődése
5. Hagyományos számítógép architektúrák fejlődése
6. Multimédiás architektúrák
7. GPU architektúrák fejlődése

Tematika: miről lesz szó a félévben?

- Számítógép architektúrák fejlődése
- Párhuzamos architektúrárák
- GPU történelem, grafikus processzorok fejlődése
- Legelterjedtebb API-k áttekintése
  - OpenCL
  - CUDA
- Speciális algoritmusok párhuzamosítása: iterációk, rekurziók,..stb.
- Esettanulmányok

Tematika: miről lesz szó a félévben?

- Optimalizálási kérdések: hogyan lesz gyorsabb a programunk?
- Esettanulmányok
  - Képfeldolgozás (szűrők megvalósítása)
  - Lineáris algrebra (nagy mátrixok sajátértékeinek számítása)
  - Grafika (?) (renderelés, rekurzív sugárkövetés)

Tartalom

1 Adminisztráció
- Bemutatkozás és elérhetőségek
- Előadások, gyakorlatok
- Számonkérés

2 Tematika

3 Számítógép architektúrák gyorsítása

4 Hagyományos számítógép architektúrák fejlődése

5 Hagyományos számítógép architektúrák fejlődése

6 Multimédiás architektúrák

7 GPU architektúrák fejlődése

A számítógép feldolgozási sebességének határai

- Alapvetően három dolog tud egy számítást lelassítani
  1. A fény túl lassú
     - Laikusok számára ez viccesen hangzik...
  2. Kéne még pár milliárd tranzisztor a lapkára
  3. Rosszul tervezett algoritmusok (emberi tényező)
- Lássuk az okokat egyesével!

Lassú fény

Kezdjük egy ki számolási példával

- A fény sebessége 300.000km/s
- Van egy gépünk AMD A8-6500K 3,5 GHz CPU-val (2013-as!)
- 1 méteres kábellel csatlakoztatunk egy USB merevlemezt.
- A proci 2 órajel alatt képes lebegőpontos összeadásra
- A következő adat a lemezen van, be kell olvasni.
- Hány összeadást tudna a proci addig elvégezni, amíg a lemezről az elektronok (fénysebességgel) megérkeznek a CPU-ba?

Lassú fény

Feladat megoldása:

- A processzor órajele $3,5Ghz = 3.500.000.000Hz$
- Egy órajel $1/3.500.000.000 = 285.7$ psec (pikosec)
- Két órajel (egy összeadás) $571,8$ psec időt vesz igénybe
- A fény ennyi idő alatt $3 \cdot 10^8 m/s \cdot 571,8 \cdot 10^{-12} s = 0,17m$ utat tesz meg
- Méteres kábel esetén: $1m/0,17m \approx 6$
- Tehát a CPU-nk 6 utasításnyi időt malmozott, amíg az adat megérkezett.
  - És akkor egyéb lassító tényezővel nem is számoltunk...
  - Pl. HDD tipikus elérése 20 msec...
  - ...ami alatt a CPU szabadságra is mehet.

Lassú fény

Feladat megoldása:

- A processzor órajele $3,5 Ghz = 3.500.000.000 Hz$
- Egy órajel $1/3.500.000.000 = 285.7$ psec (pikosec)
- Két órajel (egy összeadás) $571,8$ psec időt vesz igénybe
- A fény ennyi idő alatt $3 \cdot 10^8 m/s \cdot 571,8 \cdot 10^{-12} s = 0,17 m$ utat tesz meg
- Méteres kábel esetén: $1 m/0,17 m \approx 6$
- Tehát a CPU-nk 6 utasításnyi időt malmozott, amíg az adat megérkezett.
  - És akkor egyéb lassító tényezővel nem is számoltunk...
  - Pl. HDD tipikus elérése 20 msec...
  - ...ami alatt a CPU szabadságra is mehet.

Lassú fény

Feladat megoldása:

- A processzor órajele $3,5 Ghz = 3.500.000.000 Hz$
- Egy órajel $1/3.500.000.000 = 285.7$ psec (pikosec)
- Két órajel (egy összeadás) $571,8$ psec időt vesz igénybe
- A fény ennyi idő alatt $3 \cdot 10^8 m/s \cdot 571,8 \cdot 10^{-12} s = 0,17 m$ utat tesz meg
- Méteres kábel esetén: $1 m/0,17 m \approx 6$
- Tehát a CPU-nk 6 utasításnyi időt malmozott, amíg az adat megérkezett.
  - És akkor egyéb lassító tényezővel nem is számoltunk...
  - Pl. HDD tipikus elérése 20 msec...
  - ...ami alatt a CPU szabadságra is mehet.

Lassú fény

Feladat megoldása:

- A processzor órajele $3,5Ghz = 3.500.000.000Hz$
- Egy órajel $1/3.500.000.000 = 285.7$ psec (pikosec)
- Két órajel (egy összeadás) $571,8$ psec időt vesz igénybe
- A fény ennyi idő alatt $3 \cdot 10^8 m/s \cdot 571,8 \cdot 10^{-12} s = 0,17m$ utat tesz meg
- Méteres kábel esetén: $1m/0,17m \approx 6$
- Tehát a CPU-nk 6 utasításnyi időt malmozott, amíg az adat megérkezett.
  - És akkor egyéb lassító tényezővel nem is számoltunk...
  - Pl. HDD tipikus elérése 20 msec...
  - ...ami alatt a CPU szabadságra is mehet.

Lassú fény

Feladat megoldása:

- A processzor órajele $3,5Ghz = 3.500.000.000Hz$
- Egy órajel $1/3.500.000.000 = 285.7$ psec (pikosec)
- Két órajel (egy összeadás) $571,8$ psec időt vesz igénybe
- A fény ennyi idő alatt $3 \cdot 10^8 m/s \cdot 571,8 \cdot 10^{-12} s = 0,17m$ utat tesz meg
- Méteres kábel esetén: $1m/0,17m \approx 6$
- Tehát a CPU-nk 6 utasításnyi időt malmozott, amíg az adat megérkezett.
  - És akkor egyéb lassító tényezővel nem is számoltunk...
  - Pl. HDD tipikus elérése 20 msec...
  - ...ami alatt a CPU szabadságra is mehet.

Lassú fény

Feladat megoldása:

- A processzor órajele $3,5Ghz = 3.500.000.000Hz$
- Egy órajel $1/3.500.000.000 = 285.7$ psec (pikosec)
- Két órajel (egy összeadás) $571,8$ psec időt vesz igénybe
- A fény ennyi idő alatt $3 \cdot 10^8 m/s \cdot 571,8 \cdot 10^{-12} s = 0,17m$ utat tesz meg
- Méteres kábel esetén: $1m/0,17m \approx 6$
- Tehát a CPU-nk 6 utasításnyi időt malmozott, amíg az adat megérkezett.
  - És akkor egyéb lassító tényezővel nem is számoltunk...
  - Pl. HDD tipikus elérése 20 msec...
  - ...ami alatt a CPU szabadságra is mehet.

Lassú fény

Feladat megoldása:

- A processzor órajele $3,5Ghz = 3.500.000.000Hz$
- Egy órajel $1/3.500.000.000 = 285.7$ psec (pikosec)
- Két órajel (egy összeadás) $571,8$ psec időt vesz igénybe
- A fény ennyi idő alatt $3 \cdot 10^8 m/s \cdot 571,8 \cdot 10^{-12} s = 0,17m$ utat tesz meg
- Méteres kábel esetén: $1m/0,17m \approx 6$
- Tehát a CPU-nk 6 utasításnyi időt malmozott, amíg az adat megérkezett.
  - És akkor egyéb lassító tényezővel nem is számoltunk...
  - Pl. HDD tipikus elérése 20 msec...
  - ...ami alatt a CPU szabadságra is mehet.

Lassú fény

- Érdekesség: első 3GHz-es CPU
  - Pentium 4
  - Megjelenés: 2005 körül
- Azóta miért nem nőnek radikálisan az órajelek?
  - Előző példa: 3,5 GHz: $\sim$ 8 cm/órajelciklus
  - Pl. 20 Ghz: $< 2cm$. Alig ér át a fény a lapka egyik sarkából a másikba.
- Minél vékonyabb a vezeték, annál nagyobb az ellenállása
  - A hőtermelés arányos az ellenállással

Kevés tranzisztor

- Az algoritmusaink egymás utáni lépésekből állnak
  - Szekvenciális végrehajtás
- Az egymástól független utasításokat egyszerre lehet végrehajtani
- Kulcs: párhuzamosítás
- Párhuzamosítás számtalan módon lehetséges
  - Ennek egyik formája a GPU-k használata
  - Párhuzamosítás több végrehajtó egységet igényel,..
  - ...szaporodnak a tranzisztorok a foglalatban.

Rosszul megírt programok

- Az emberi tényező...
- Az oktatók kudarca...
- A programozók lustasága ...
- ...
- Nem célja a tárgynak, hogy megoldást találjon a problémára.

Megoldások

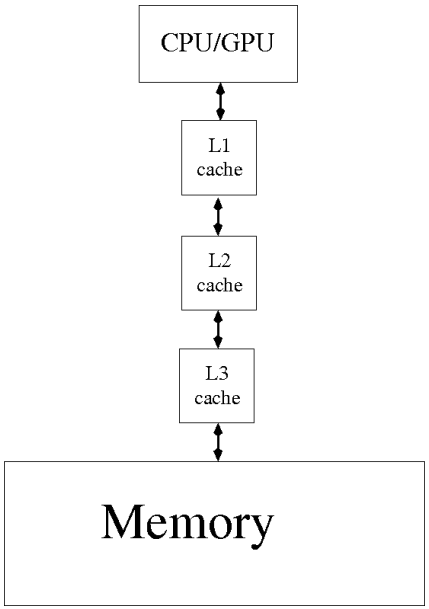
- Ha a fény lassú: utaztassuk kevesebbet az adatokat
- Ha a párhuzamosság gyorsít: tervezzünk párhuzamos architektúrákat
- (Ha rosszul programozunk: programozzunk jobban!)

Lassú fény (1)

- Sajnos a fénysebességen nem tudunk gyorsítani
- Megoldás (1): adatokat (elektronokat) közelebbre kell hozni
  - Erre találták ki a gyorsítótárat (cache)
  - Meg kell mondani előre, hogy milyen adatokra lesz szükség, ...
  - ...vagy ki kell találni.
- Megoldás (2): egyszerre több adatot kell mozgatni
  - Több vezetéken párhuzamosan.
  - Ezért volt értelme a 32 bites processzorokat 64 bitesre cserélni.
  - 32 bit 32 vezetéket jelent, 64 bit a dupláját
- Komoly tanulsága a problémának: **nagy adatmozgatás sok időt vesz igénybe.**

Lassú fény (2)

- Gyorsítótár/cache. (Általában háromszintű, de nem mindig)



Lassú fény (3)

- Cache kezelésre oda kell a programozáskor is figyelni.
- Indirekción keresztül nehéz cache-elni.
- Példa: `http://gameprogrammingpatterns.com/data-locality.html`
  - Második megoldás ötvenszer (!) gyorsabb.
  - A cache előrejelzők pontosságát az indirekciók teljesen tönke tudják tenni.

Lassú fény (4)

Memóriatípusok összehasonlítása

Memória típusa	Elérés	adatátvitel
Regiszterek	$\sim 100psec$	n/a
Gyorsítótár	$\sim 10nsec$	$\sim 100GByte/sec$
RAM	$\sim 500nsec$	$\sim 1Gbyte/sec$
Hálózati gyorsítótár	$\sim 50\mu sec$	n/a
SSD	$\sim 200\mu sec$	$\sim Gbyte/sec$
Merevlemez	$\sim 20msec$	$\sim 200Mbyte/sec$

Lassú fény (4)

- Magát az utasítást is be kell olvasni
- Az utasítás legyen minél rövidebb
  - Ehhez kevesebb utasítást lehet használni
  - Cserébe több regiszter szokásos
- Redukált utasításkészlet: RISC architektúra
  - RISC: Reduced Instruction Set Computer
  - "Hagyományos" CISC architektúra: Complex Instruction Set Computer
  - x64: **CICS**, ARM (Qualcomm, Apple, ...), RISC-V: **RISC**

Lassú fény (4)

- A CISC architektúra sokszor belül RISC.
  - Processzor a processzorban: az utasítást át kell fordítani
  - Egy CISC utasítást több RISC parancs írhat le.
  - Ettől sem lesz kevesebb tranzisztor a gépben, sőt... ☹
- Költői kérdés: miért fogyaszt kevesebbet egy MacBook?
- A low-cost jövő: Qualcomm CPU-k laptopokba?
  - Vagy a jelen...

ARM versus RISC-V

ARM (Zárt/Érett)

- **Üzleti modell:** Licencdíj + darabonkénti jogdíj (Royalty).
- **Ökoszisztéma:** Domináns, „mindenen fut” (Android, iOS, Windows).
- **Kötöttség:** Nem módosítható utasításkészlet (Vendor lock-in).
- **Cél:** Mobil, szerver, autóipar.

RISC-V (Nyílt/Moduláris)

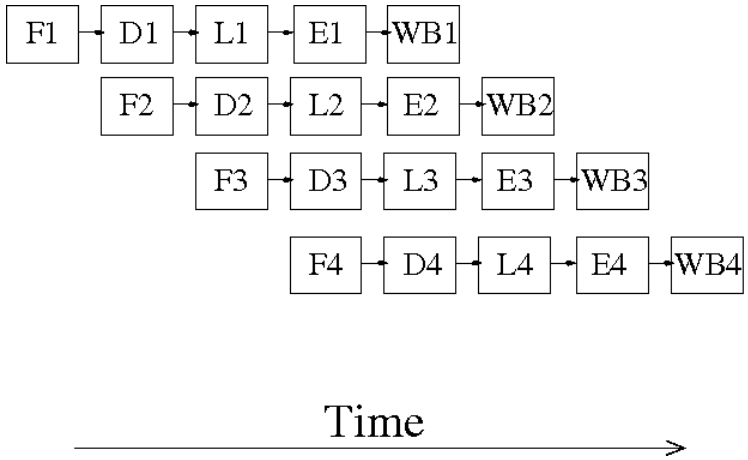
- **Üzleti modell:** Ingyenes ISA (szabadon felhasználható).
- **Testreszabás:** Moduláris + **saját utasítások** adhatók hozzá.
- **Ökoszisztéma:** Gyorsan fejlődő, de még töredezett szoftveres támogatás.
- **Cél:** IoT, MI gyorsítók, kutatás.

Párhuzamos architektúrák (5)

- Egy utasítás életútja (elnagyoltan)
  1. Beolvasás (F: fetch)
  2. Értelmezés (D: decode)
  3. Paraméterek beolvasása (L: load)
  4. Művelet végrehajtása (E: execution)
  5. Eredmény visszaírása (WB: write back)
- Az egyes részek nagyjából függetlenek egymástól.
- Párhuzamosan végrehajthatók

Párhuzamos architektúrák (1)

- Futószalag elv: részfeladatok párhuzamos végrehajtása



Párhuzamos architektúrák (1/2)

- A futószalag-végrehajtást jól megválasztott utasítások segítik
- Kevesebb utasítás jobb szokott lenni
- RISC architektúra ajánlatos
  - Ezért is alakítják át a régi CISC architektúrákat RISC-é belül.
  - Pl. Intel x86/x64 processzorcsalád.

Párhuzamos architektúrák (1/3)

- Ismerkedjünk meg két fogalommal!
- Késleltetés (latency)
  - Az utasítás megkezdése és befejezése között eltelt idő.
- Áteresztőképesség (throughput)
  - Egyszerre hány utasítást tud párhuzamosan végrehajtani
  - Minél több állapotra (stage) bontható az utasítás, annál több.

Párhuzamos architektúrák (2/1)

- Szekvenciális programozás gyenge pontja: feltételes ugrások
  - if/else
  - while/do-while
  - for
  - switch/case
- Feltételeket előre meg kell becsülni (statisztikai felmérések)
  - Utasítás előrejelzés
  - Szuperskalár architektúra

Párhuzamos architektúrák (2/2)

- Független utasítások sorrendjét meg lehet cserélni
  - Soron kívüli végrehajtás
  - angolul:Out of order execution
- Spekulatív végrehajtás: szuperskalár architektúra alapja

Párhuzamos architektúrák (3)

- Bár párhuzamosításról van szó, szekvenciális programokat futtat
  - Minden utasítás saját adatain dolgozik.
  - SISD architektúra: Single Instruction Single Data
- Legkézenfekvőbb párhuzamosítás: több CPU betétele
  - Egyszerre több utasítás a saját adatain dolgozik.
  - MIMD architektúra: Multiple Instruction Multiple Data
- MIMD architektúra két fajtája:
  - Párhuzamosság a processzek szintjén: több független CPU (mag).
  - Párhuzamosság a szálak szintjén: több szál CPU-n (magon) belül. Hyperthreading, Simultaneous Multithreading (SMT)

Párhuzamos architektúrák (5)

- SISD/MIMD architektúrát már ismerjük
- Mi lehet a MISD/SIMD architektúra?
- MISD: Multiple Instruction Single Data?
  - Más utasítás ugyanazon az adaton.
  - Értelme: Ugyanazt több áramkörrel számítjuk.
  - Ha az egyik hibázik, a többi felülbírálhatja: hibatűrő rendszerek
  - Ettől gyorsabb nem lesz a rendszerünk $\rightarrow$ nem foglalkozunk vele

Párhuzamos architektúrák (6)

- SIMD: Single Instruction Multiple Data
  - Elérkeztünk a GPU architektúrához!
- Ugyanazt az utasítást sok adaton kell végrehajtani.
- Példa: Nagy cég számláinak összeadása.
  - Rendkívül jól párhuzamosítható.
- SIMD variánsai
  - Vector processzor (szinoníma)
  - SIMT: Single Instruction Multiple Thread: SIMD többszálú változtata

Kitekintés: miniatürizálás

- Csíkszélesség: milyen vastagok a vezetékek
- Jelenleg: $\sim 2(?) - 5$ nm.
- Csíkszélesség csökkenésének előnyei
  - Több tranzisztor fér el azonos területen
  - Magasabb órajel
- Korlátok:
  - Atomi méreteket hamarosan elérjük. Atom átmérője : $\sim 10 - 100$ pikométer
  - Szivárgó áram

Tartalom

1. Adminisztráció
   - Bemutatkozás és elérhetőségek
   - Előadások, gyakorlatok
   - Számonkérés
2. Tematika
3. Számítógép architektúrák gyorsítása
4. Hagyományos számítógép architektúrák fejlődése
5. Hagyományos számítógép architektúrák fejlődése
6. Multimédiás architektúrák
7. GPU architektúrák fejlődése

Számítógép architektúrák történeti fejlődése

- Őskor: speciális célú áramkörök
  - Konkrét számítási feladatok elvégzésére tervezték
- Ókor: Neumann-elvvel megjelenik a "program" mint fogalom
  - Programot a memóriában tároljuk
  - Központi feldolgozó egység (CPU) megjelenik
- Középkor: CPU gyorsítgatása
  - Futószalagok, gyorsítótárak, RISC, szuperskalár rendszerek
- Újkor: Több CPU egy gépben, majd több mag egy CPU-n belül.
  - Párhuzamos (többprocesszoros) programozás
  - Többszálas programozás
- Legújabb kor: GPU-k bevetése általános számítási feladatokra
- Poszmodern kor
  - GPU magok specializációja (árnyaló, sugárkövető, tenzor, ...)
  - CPU magok specializációja: P/E magok, NPU
  - 3D cache

Tartalom

1 Adminisztráció
- Bemutatkozás és elérhetőségek
- Előadások, gyakorlatok
- Számonkérés

2 Tematika

3 Számítógép architektúrák gyorsítása

4 Hagyományos számítógép architektúrák fejlődése

5 Hagyományos számítógép architektúrák fejlődése

6 Multimédiás architektúrák

7 GPU architektúrák fejlődése

Számítógép architektúrák történeti fejlődése

- És a legújabb trendek?
- CPU: különböző magok alkalmazása
  - energiatakarékos (E)
  - teljesítmény (P)
- GPU: itt is különböző magok
  - hagyományos (hány 'core' van a gépben?)
  - sugárkövetés (sugármetszés): RT-core - RTX kártyák
  - Gépi tanulás: tensor-code
  - TMU: Texture Mapping Unit
  - RoP: Render Output Unit

Tartalom

1. Adminisztráció
   - Bemutatkozás és elérhetőségek
   - Előadások, gyakorlatok
   - Számonkérés
2. Tematika
3. Számítógép architektúrák gyorsítása
4. Hagyományos számítógép architektúrák fejlődése
5. Hagyományos számítógép architektúrák fejlődése
6. Multimédiás architektúrák
7. GPU architektúrák fejlődése

Multimédiás architektúrák

- SIMD architektúra 90-es évek végén bemutatkozott.
- Marketingesek a multimédiás processzor kulcsszót találták ki
  - Intel: SSE, SSE2
  - AMD: 3DNow!
- Multimédia: párhuzamos adatokon végrehajtott utasítások
  - Általában 4 multimédiás regiszter
  - Utasítások: algebrai műveletek, feltételek vizsgálata, konverziók, ..stb.
  - Egész és lebegőpontos számokon egyaránt működik
- GPU-k megjelenésével elvesztette jelentősségét.

Tartalom

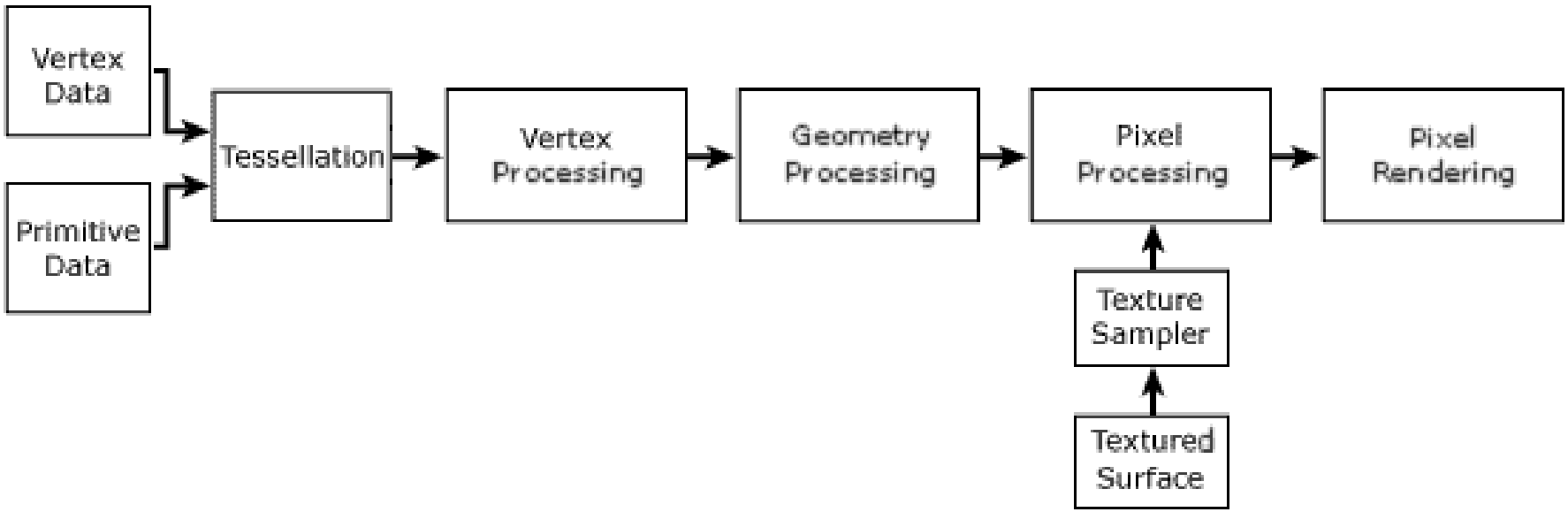
1. Adminisztráció
   - Bemutatkozás és elérhetőségek
   - Előadások, gyakorlatok
   - Számonkérés
2. Tematika
3. Számítógép architektúrák gyorsítása
4. Hagyományos számítógép architektúrák fejlődése
5. Hagyományos számítógép architektúrák fejlődése
6. Multimédiás architektúrák
7. GPU architektúrák fejlődése

GPU architektúrák fejlődése

- 1995 előtt: grafikus processzorok előtti kétdimenziós gyorsítások:
  - 1978: Intel iSBX 275 Video Graphics Controller
    - Hardveres vonal, ív, bitmap...stb. rajzolás
  - 1986: Comodore Amiga : Blitter
    - CPU-tól független grafikus processszorral
  - 1986: Texas Instruments: TMS34010,
    - Általános célú processzor, grafikára hegyezett utasításkészlettel
  - 90-es évek elején egyre több grafikus chipgyártó terméke jelent meg
    - ATI, Matrox, S3 ...stb

GPU architektúrák fejlődése

- Grafikus pipeline rendkívül kiválóan párhuzamosítható



GPU architektúrák fejlődése

- 1995-től: 3D-s gyorsítások
  - Fejlesztés hajtóereje: OpenGL és DirectX API-k megjelenése
    - Háromdimenziós vizualizáció támogatása
    - Inkrementális képszintézis alapján, majd 2019-től sugérkövetés
    - Futószalag elv megjelenése a képszintézisben (jól párhuzamosítható!)
  - Transzformációk és megvilágítás (T&L: Tranform & Ligting)
    - Nvidia GeForce 256
    - Profi kártyák (méregdrágán) is megjelentek
- 2000 táján az első shader-ek megjelentek
  - Vertex-ek és pixelek számítására külön program írható
  - Mai értelemben használt GPU-król itt kezdünk beszélni
  - NVidia GeForce3, Ati Radeon 9700

GPU architektúrák generációi

- Valasek Gábortól megismerhettétek (?)

Köszönjük a figyelmet!