Az SZTE Kutatóegyetemi Kiválósági Központ tudásbázisának
kiszélesítése és hosszú távú szakmai fenntarthatóságának megalapozása
a kiváló tudományos utánpótlás biztosításával”

Eötvös Loránd Kollégium
Informatika Műhely
2011. 11. 28.
GPU Architektúrák
Nagy Antal

TÁMOP-4.2.2/B-10/1-2010-0012 projekt

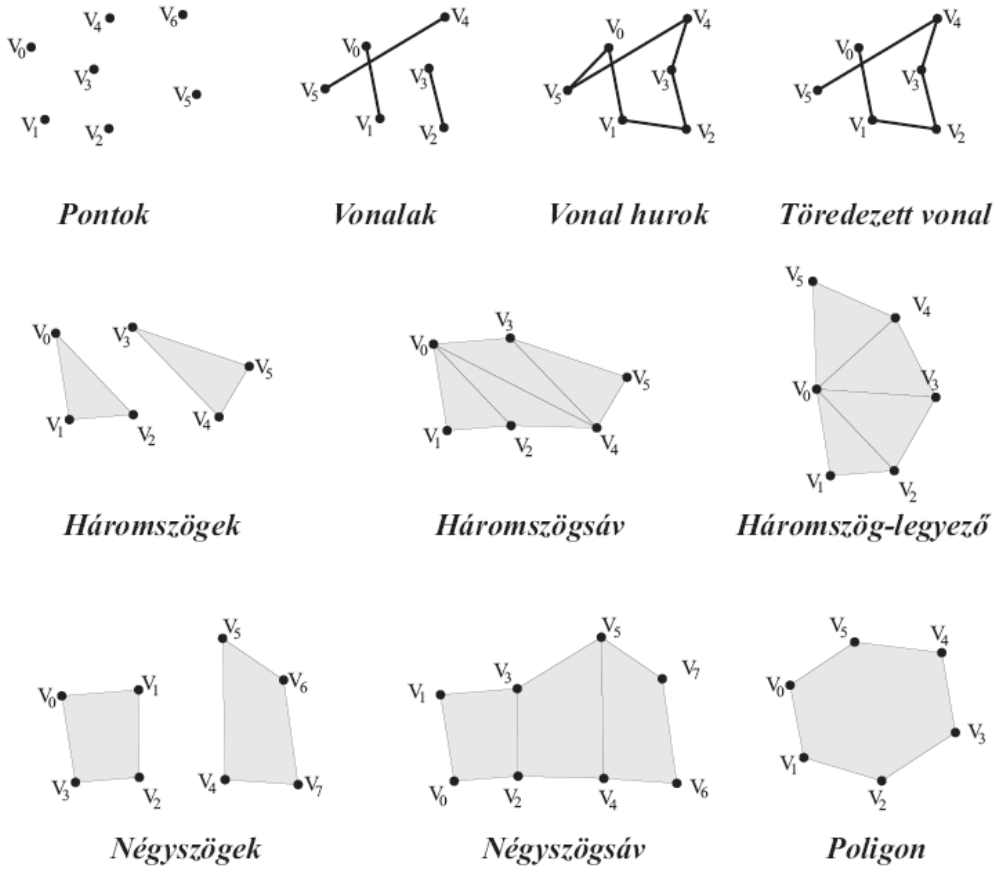
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Tartalomjegyzék

ÚJ SZÉCHENYI TERV
MAGYARORSZÁG MEGÚJUL
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

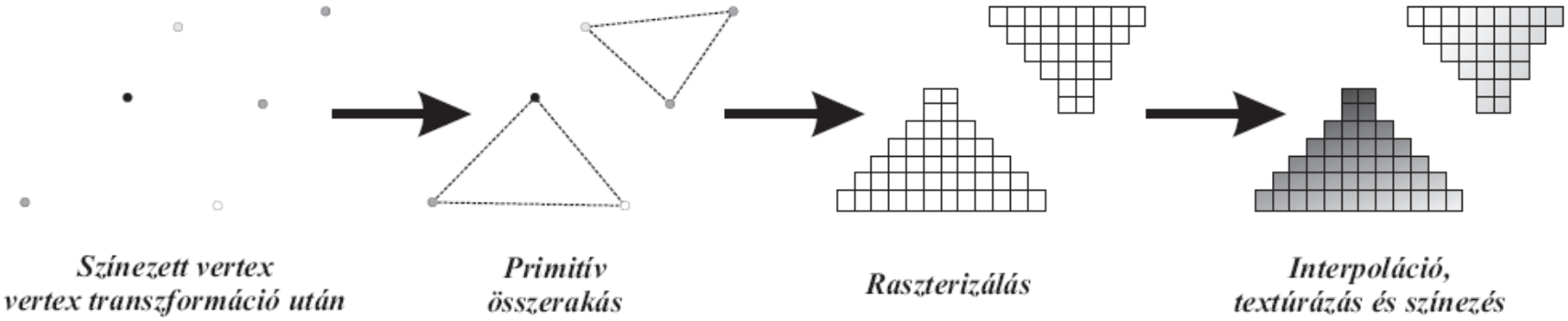
Grafikus csővezeték

- Vertex feldolgozás
  - A vertex-ek egyenként a képernyő térbe vannak transzformálva
- Primitív feldolgozás
  - A vertex-ek primitívekbe vannak szervezve
- Raszterizálás
  - Primitívenként
    - Fragmensek
- Fragmens textúrázás és színezés
  - fragmensenként



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

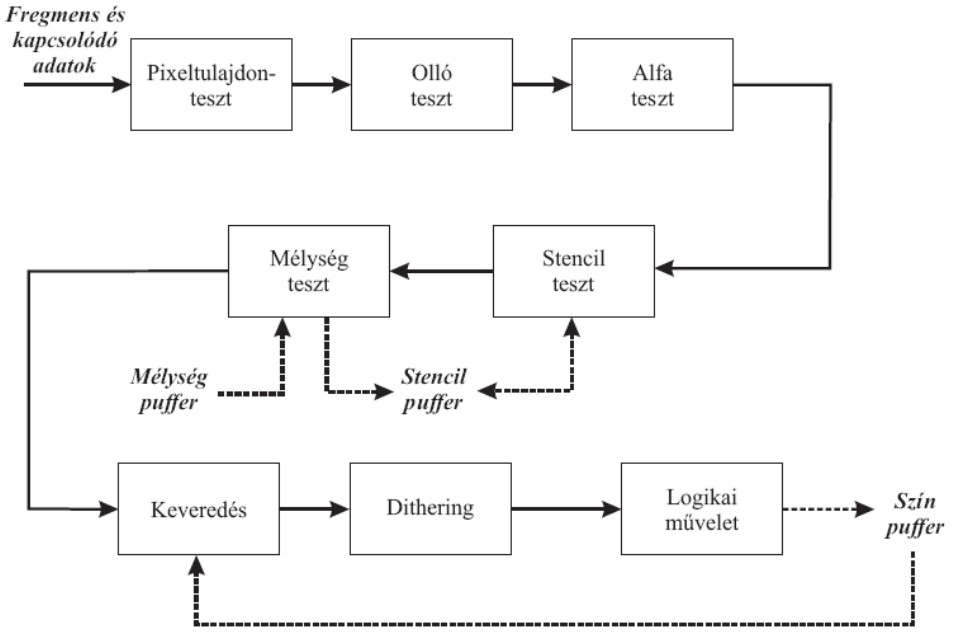
Grafikus csővezeték vizualizálása



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Raszter műveletek

- Pixeltulajdon
  - A képernyő adott pixelére lehet-e írni?
- Olló
  - Téglalap terület
- Alfa
  - Színkeveredés
  - A fragmens végső alfa értékének összehasonlítása egy megadott értékkel
- Stencil
  - Stencil pufferben lévő érték összehasonlítása egy megadott értékkel
  - Műveletek, melyek módosítják a stencil puffert
- Mélység teszt
  - Mélység pufferben lévő értékek
  - Szín puffer frissítése



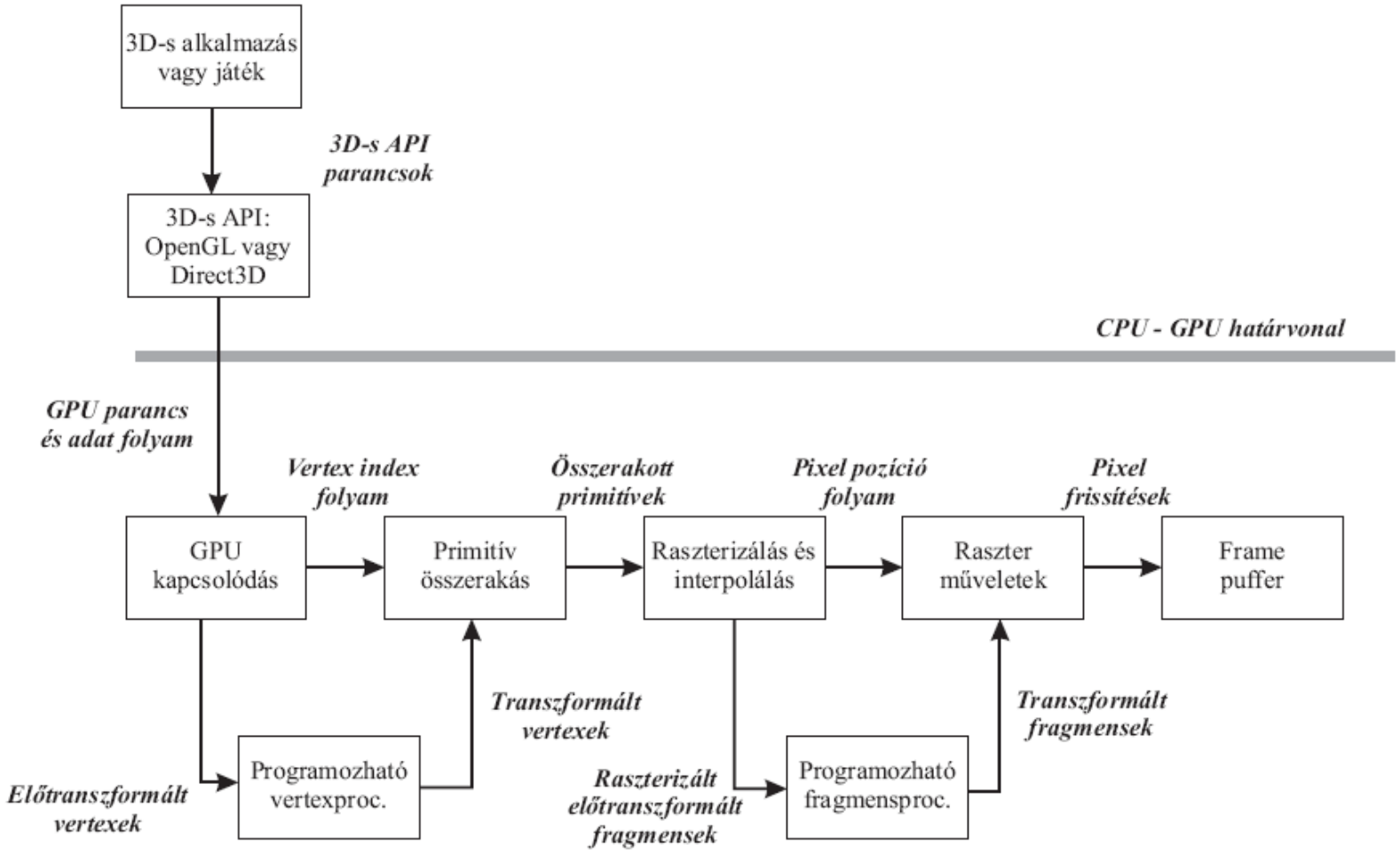
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Raszter műveletek

- Keveredés
  - Fragmens és szín puffer keveredése
- Dithering
  - Belső számítások végrehajtása egy magasabb felbontáson
- Logikai műveletek
  - Egymást kölcsönösen kizárják a keveredéssel
  - Bitminták
    - Forrás és cél pixel adat
  - AND, OR, XOR
    - kurzor rajzolása a háttér tárolása nélkül (XOR)

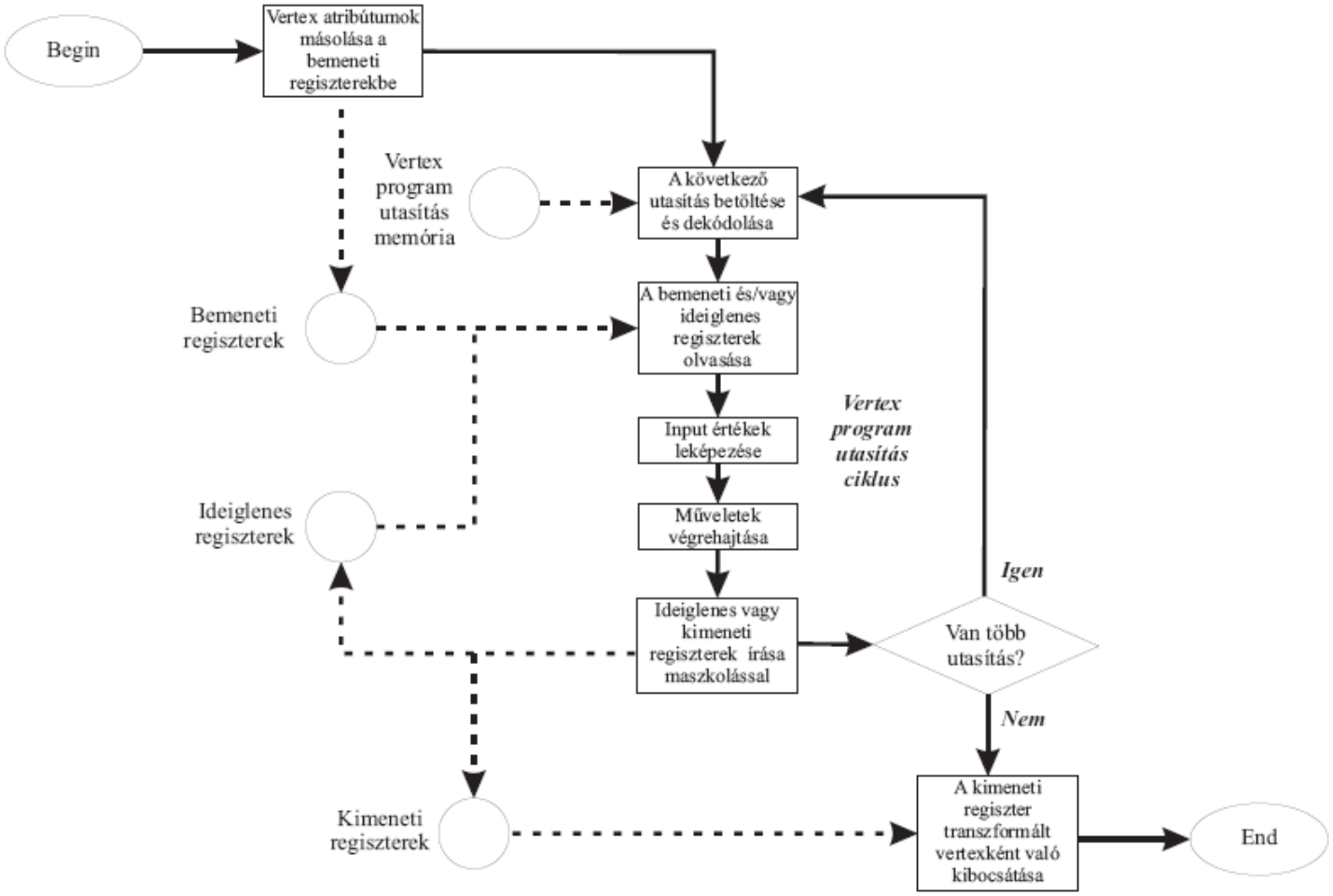
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Grafikus csővezeték

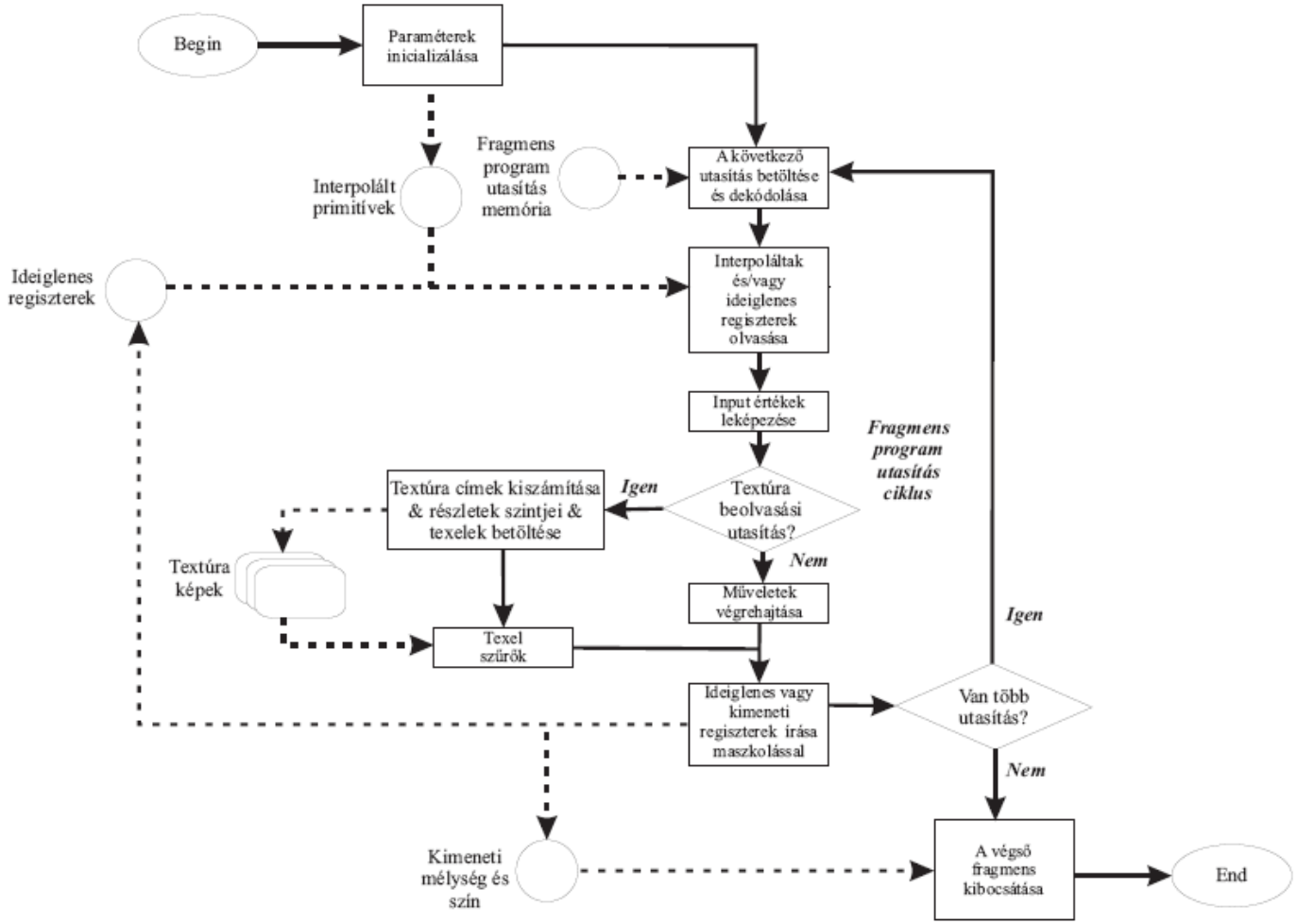


A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Programozható vertexprocesszor



Programozható fragmensprocesszor



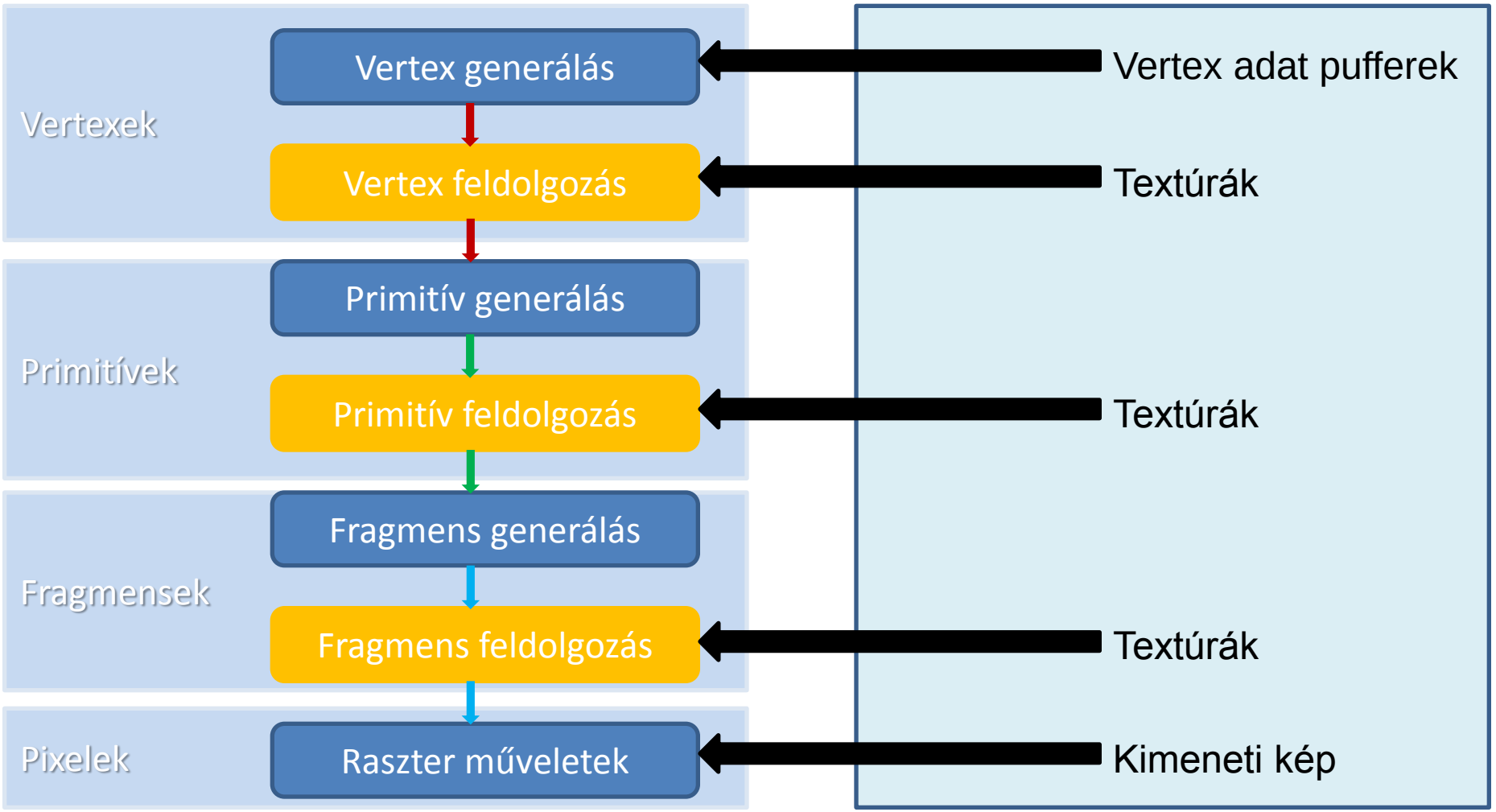
Programozható geometriai processzor

- Direct3D 10
- OpenGL 3.2
- Új grafikus primitívek előállítása az előzetesen a csővezetékbe küldött primitívekből
  - Pontok, vonalak, háromszögek
- Szomszédsági információk

- Felhasználása
  - Pont sprite-ok,
  - geometriai tesszalálás,
  - árnyék térfogat kinyerés,
  - egymenetes renderelés egy cube map-be
- A shader kimenete újból felhasználható a vertex shader-ben
- Általános célú számításokra nem szokás alkalmazni

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Grafikus csővezeték



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Silicon Graphics RealityEngine (1993)

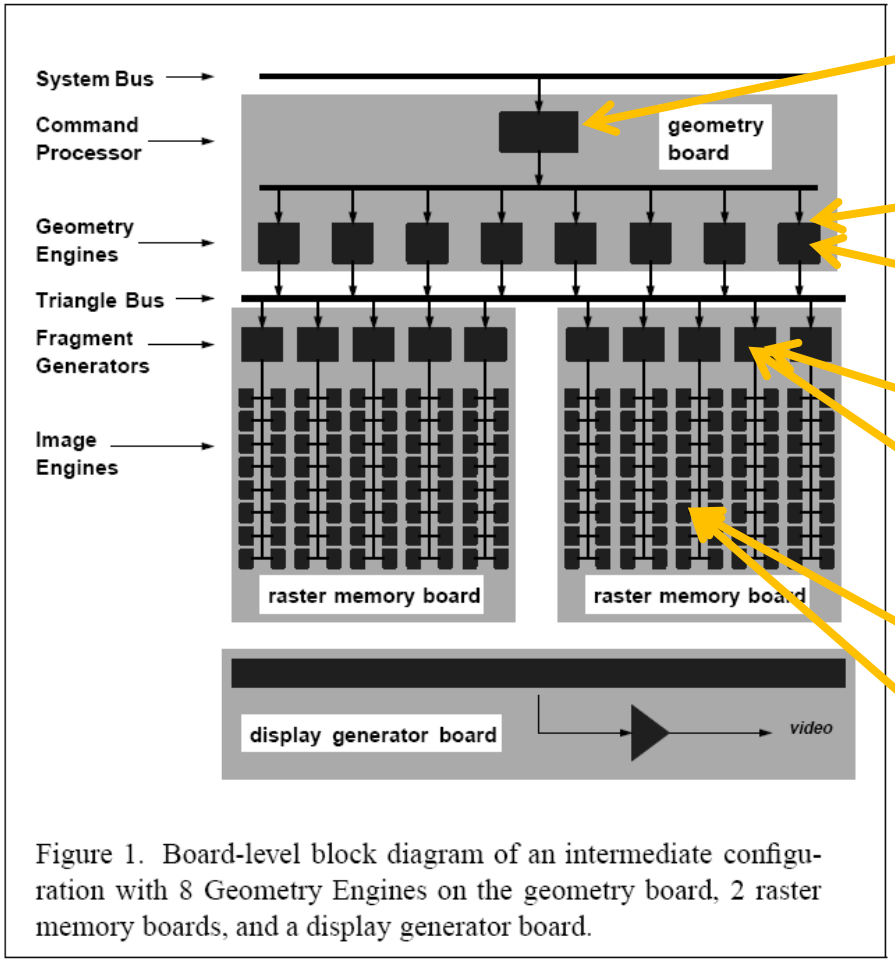
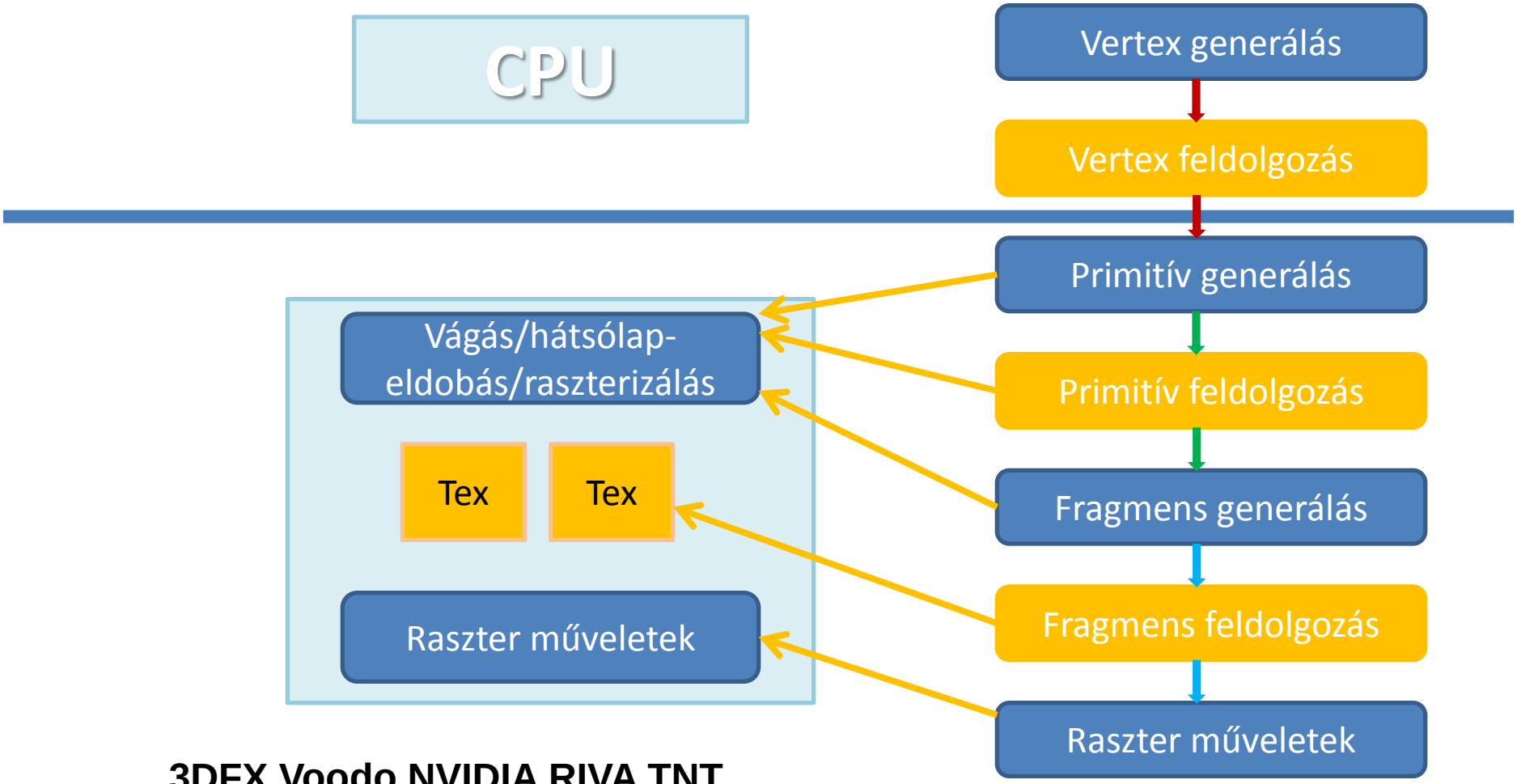


Figure 1. Board-level block diagram of an intermediate configuration with 8 Geometry Engines on the geometry board, 2 raster memory boards, and a display generator board.

- Vertex generálás
- Vertex feldolgozás
- Primitív generálás
- Primitív feldolgozás
- Fragmens generálás
- Fragmens feldolgozás
- Raszter műveletek

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

3D-s grafikus gyorsító 1999 előtt



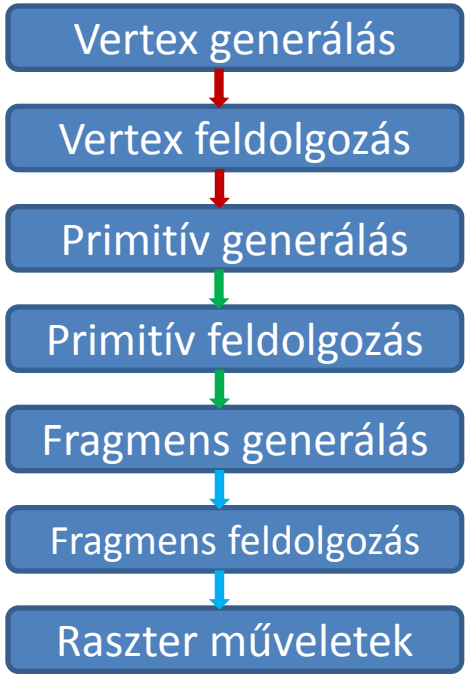
3DFX Voodo NVIDIA RIVA TNT

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

GPU 1999 körül

CPU

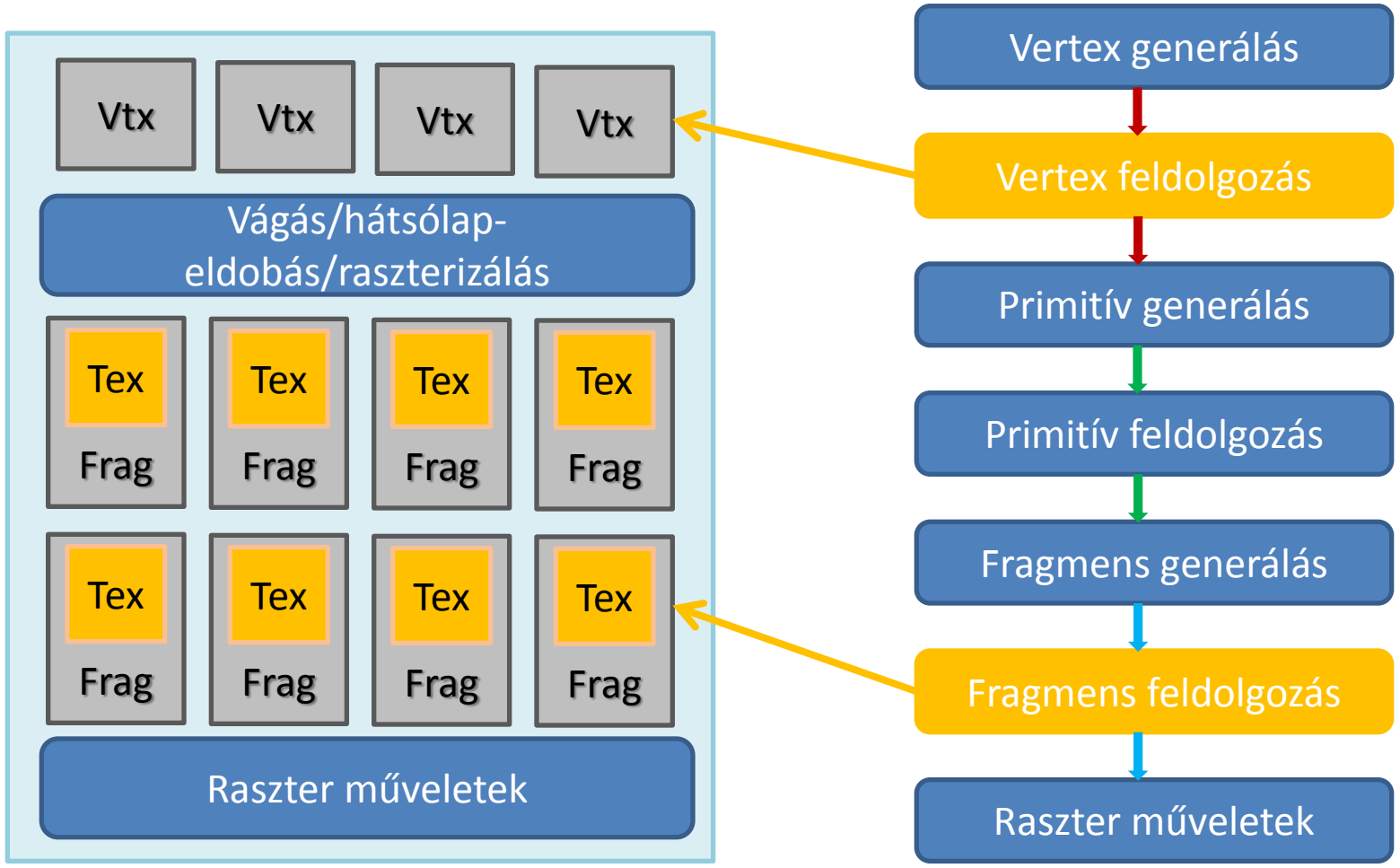
GPU



NVIDIA GeForce 256

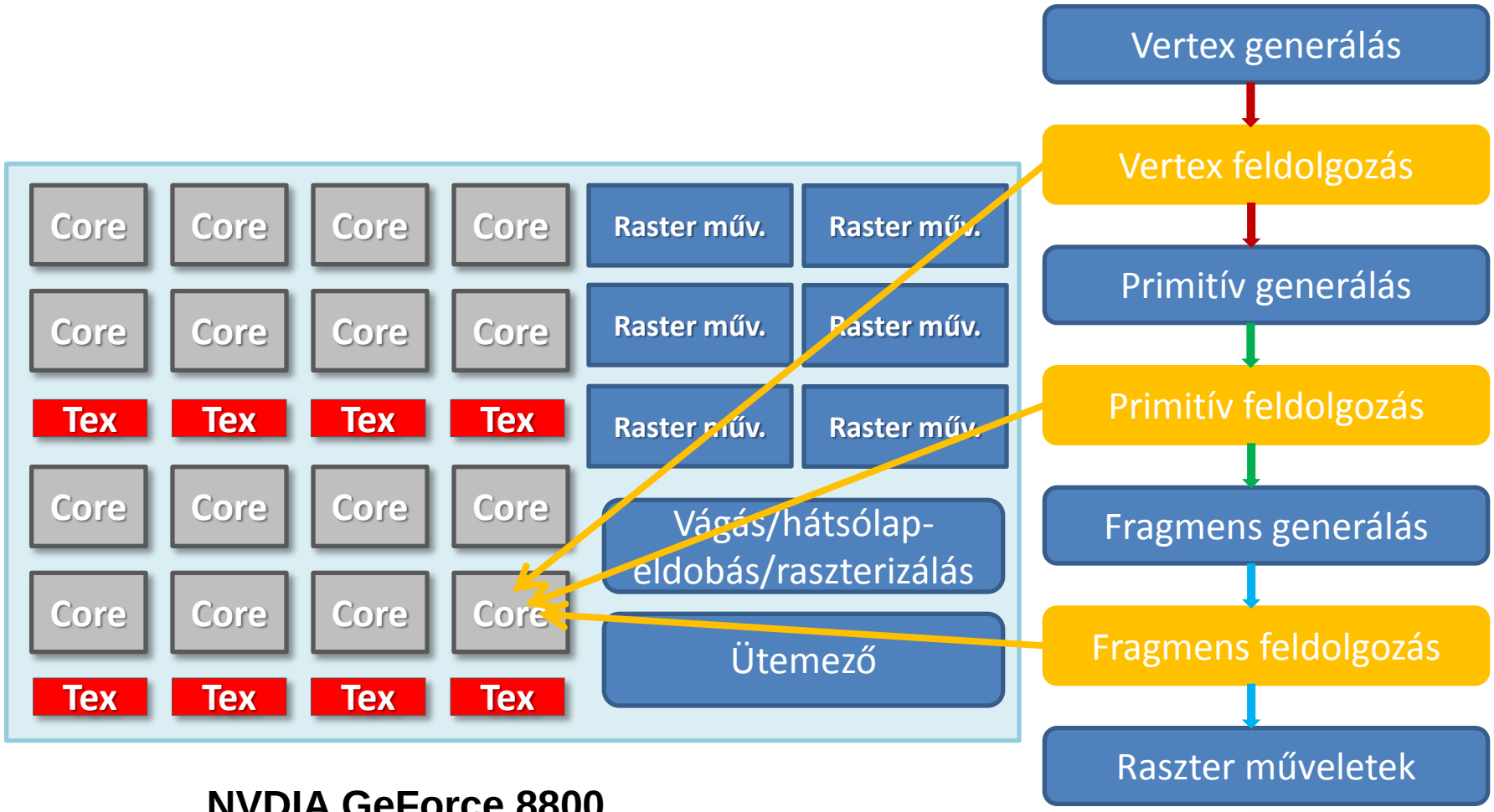
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Direct3D 9 programozhatóság 2002



ATI Radeon 9700

Direct3D 10 programozhatóság 2006



NVDIA GeForce 8800
egységes shader GPU

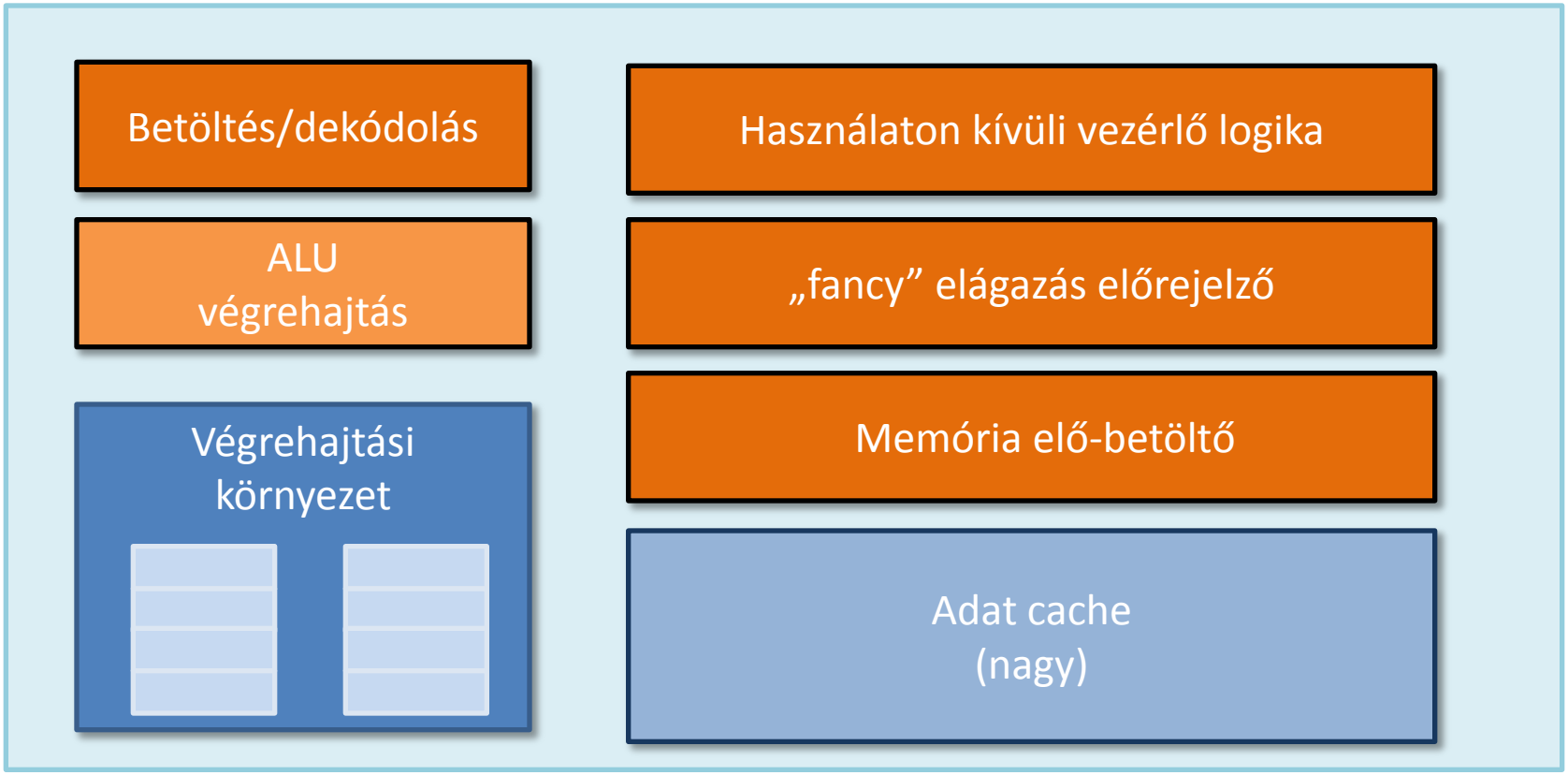
ÚJ SZÉCHENYI TERV
MAGYARORSZÁG MEGÚJUL
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

General-Purpose Computing on Graphics Processing Units (GPGPU)

- Egységes shader modell
  - Shaderek megvalósítása közelebb került egymáshoz
    - Egyszerű
    - Kevés utasítás
    - Általános célú végrehajtóegység
    - Több száz
      - Hatalmas számítási kapacitás

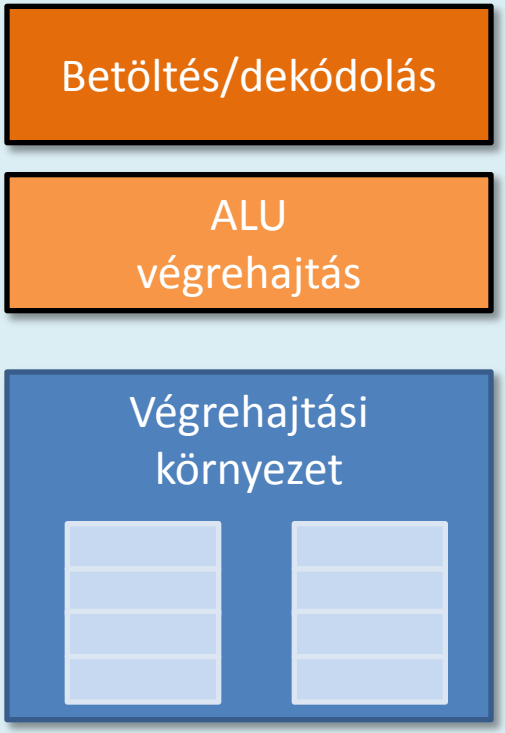
ÚJ SZÉCHENYI TERV
MAGYARORSZÁG MEGÚJUL
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

CPU-stílusú core-ok



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Első ötlet

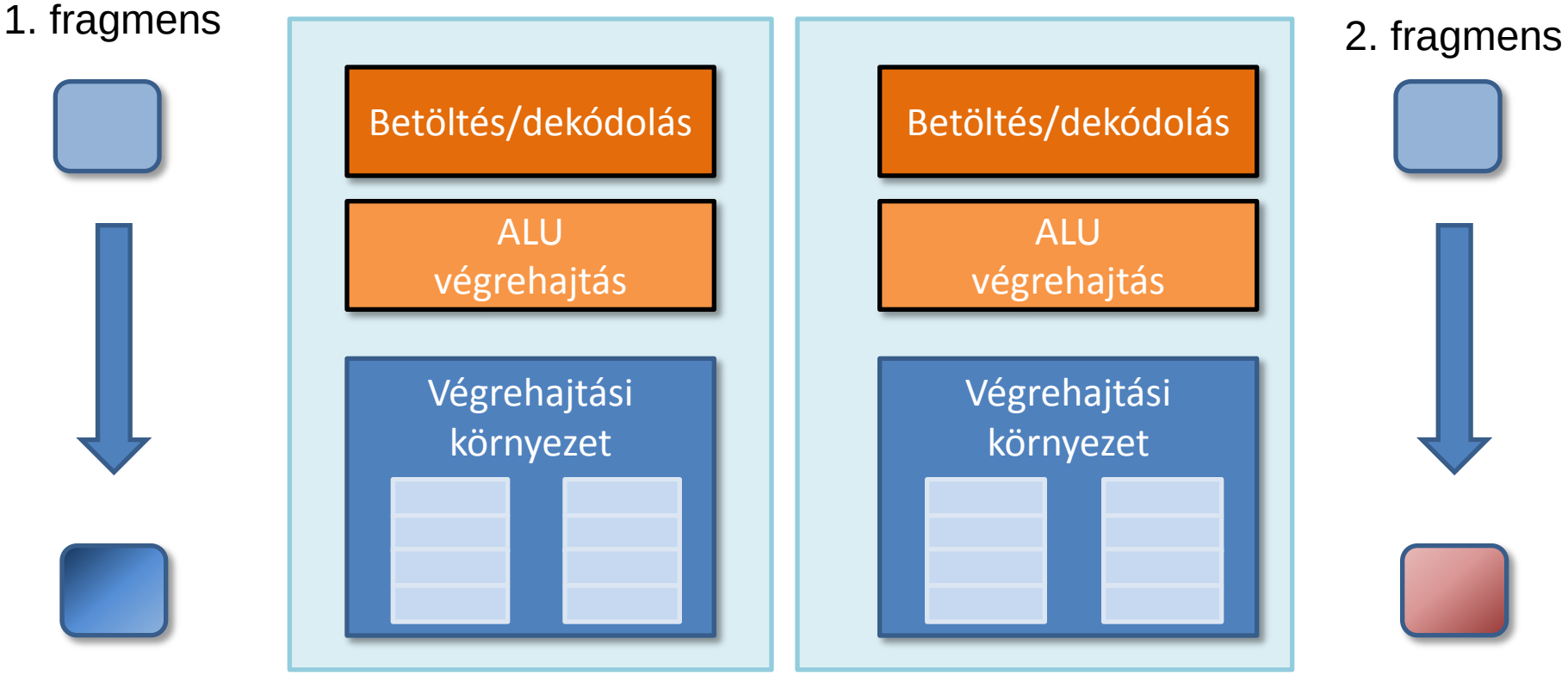


Tüntessük el azokat a komponenseket, amelyek az egyetlen utasítás folyam gyors futását segítik!

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Két mag (core)

két fragmens párhuzamos feldolgozása

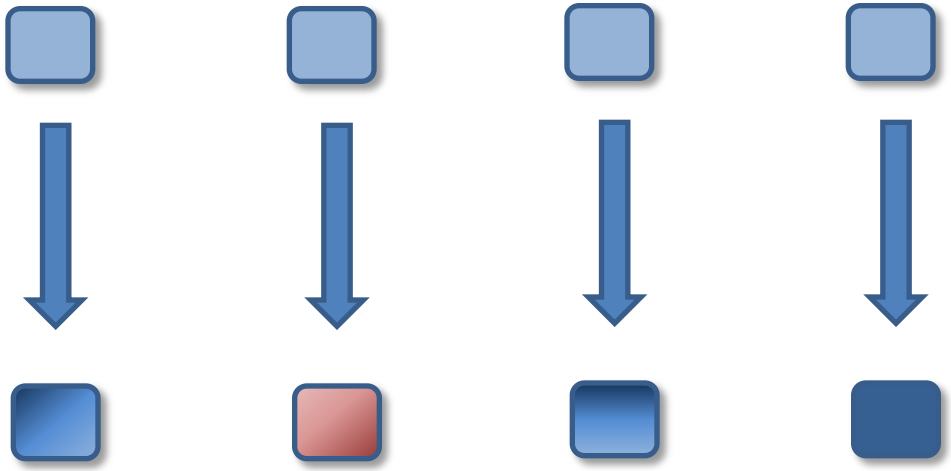


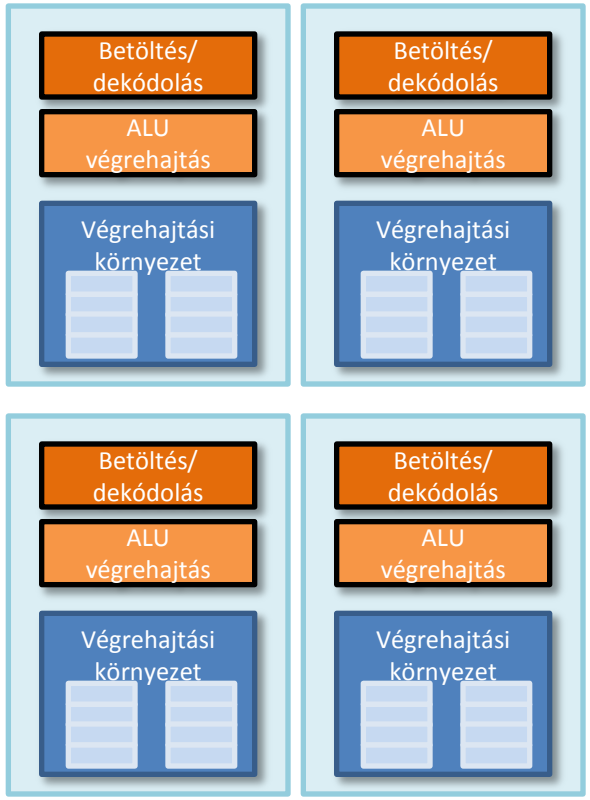
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Négy mag (core)

négy fragmens párhuzamos feldolgozása

1. fragmens 2. fragmens 3. fragmens 4. fragmens

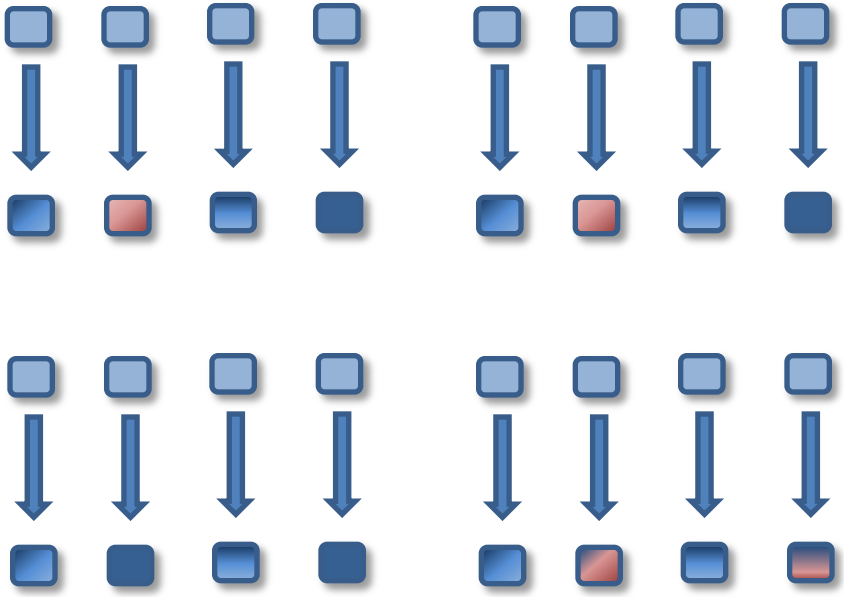




A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Tizenhat mag (core)

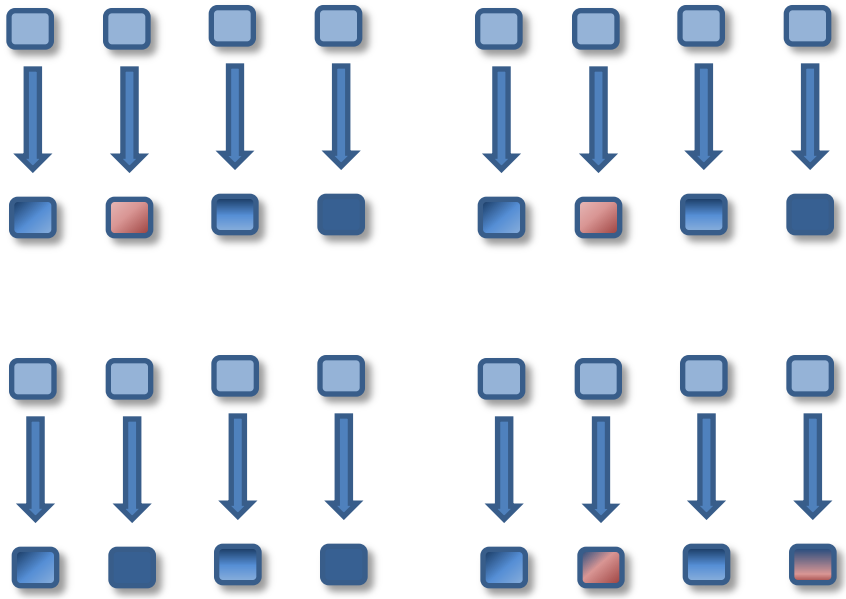
tizenhat fragmens párhuzamos feldolgozása



16 mag = 16 egyidejű utasítás folyam

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

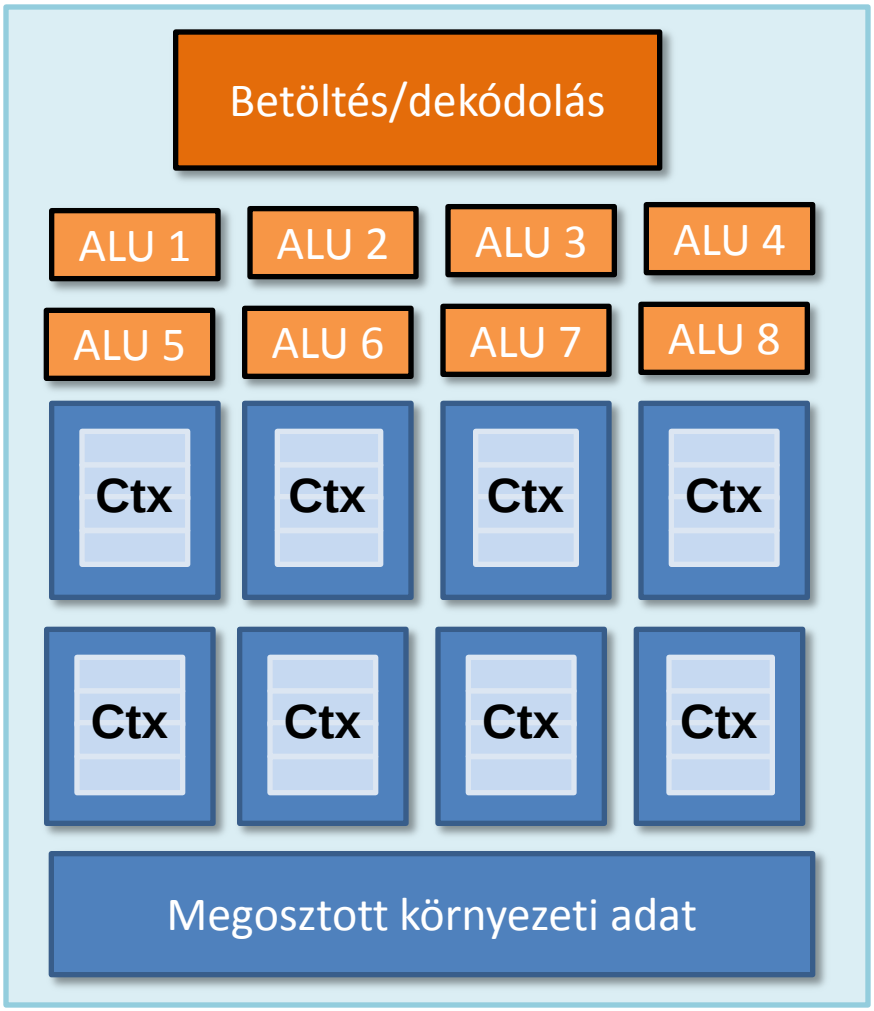
Utasítás folyam megosztás



Szükség van a fragmensek közötti utasítás folyam megosztására

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Második ötlet



Csökkentsük az utasítás folyam kezelésének költségét és összetettségét az ALU-k között

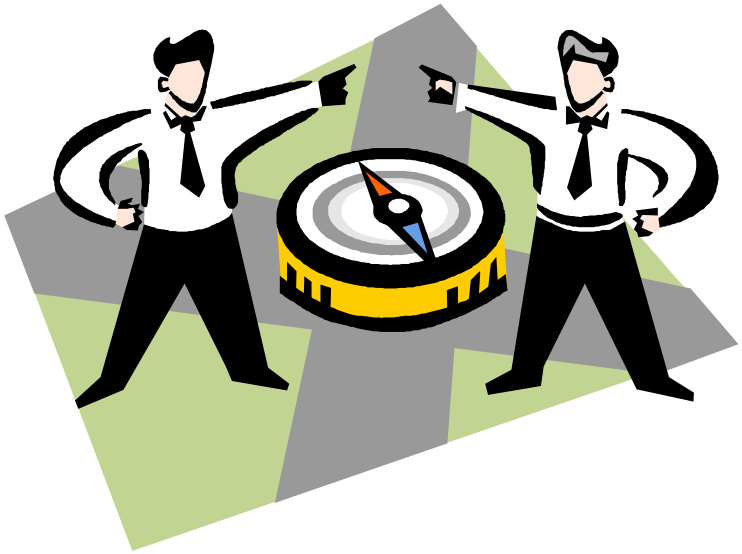
SIMD

Single Instruction Multiple Data

ÚJ SZÉCHENYI TERV
MAGYARORSZÁG MEGÚJUL
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

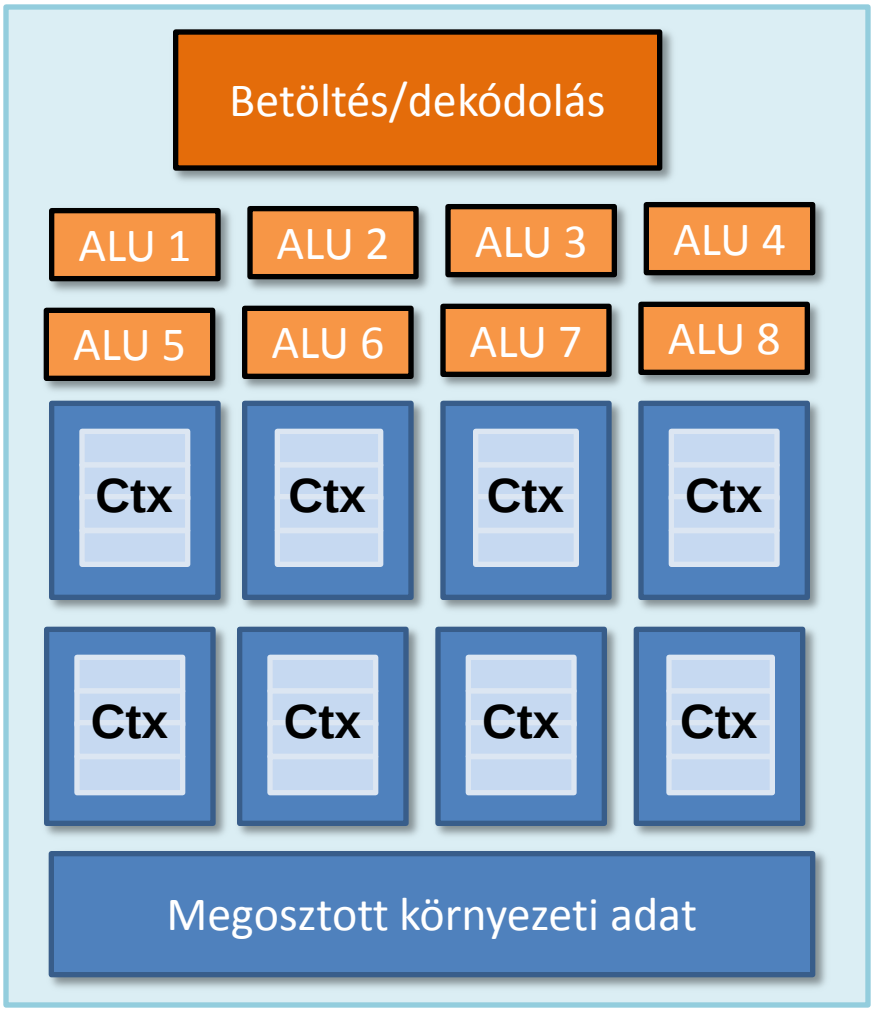
Shader módosítása

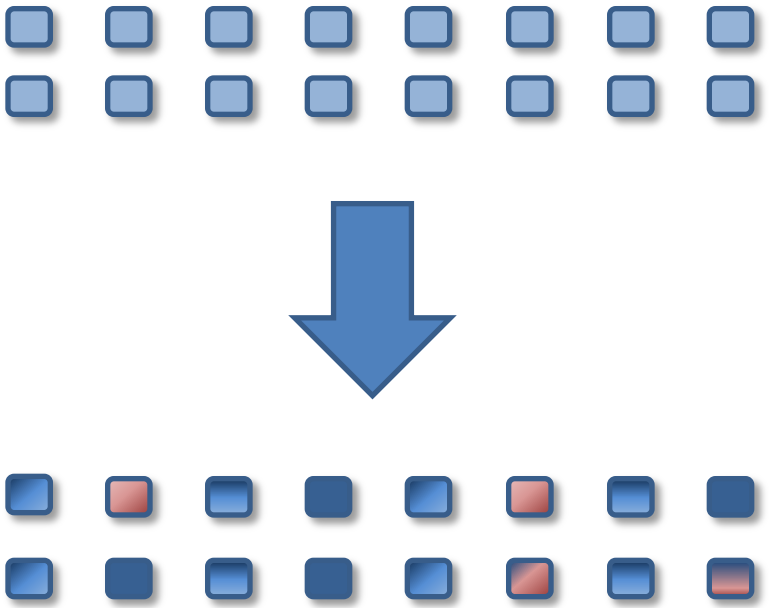
- Előző shader egy fragmenst dolgozott fel
  - Skalár műveletek
  - Skalár operandusok
- Új shader 8 fragmenst dolgoz fel
  - Vektor műveletek
  - Vektor operandusok



ÚJ SZÉCHENYI TERV
MAGYARORSZÁG MEGÚJUL
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Második ötlet





ÚJ SZÉCHENYI TERV
MAGYARORSZÁG MEGÚJUL
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

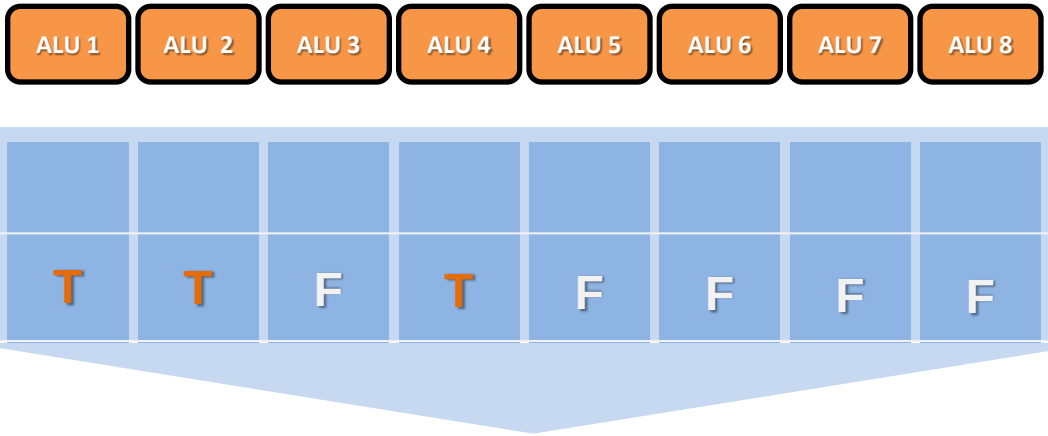
128 fragmens párhuzamosan

- 16 mag = 128 ALU
- 16 egyidejű utasítás folyam
- 128
  - Vertex
  - primitívek
  - Fragmensek

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Mi van az elágazásokkal?

Idő
(tikk-takk)



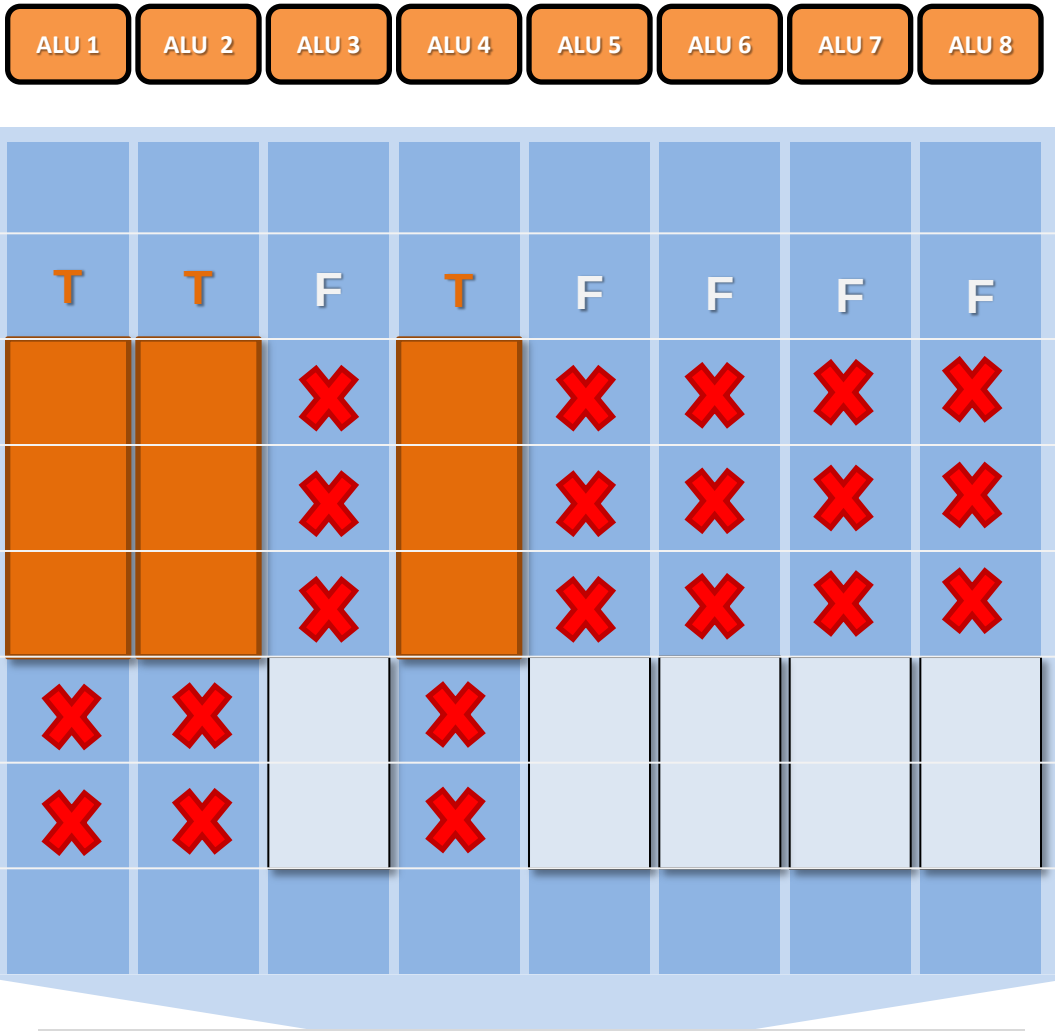
Feltétel nélküli shader kód

```
if (x > 0 {
    y = pow(x, exp);
    y *= Ks;
    refl = y + Ka;
} else {
    x = 0;
    refl = Ka;
}
```


A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Mi van az elágazásokkal?

Idő
(tikk-takk)



Feltétel nélküli shader kód

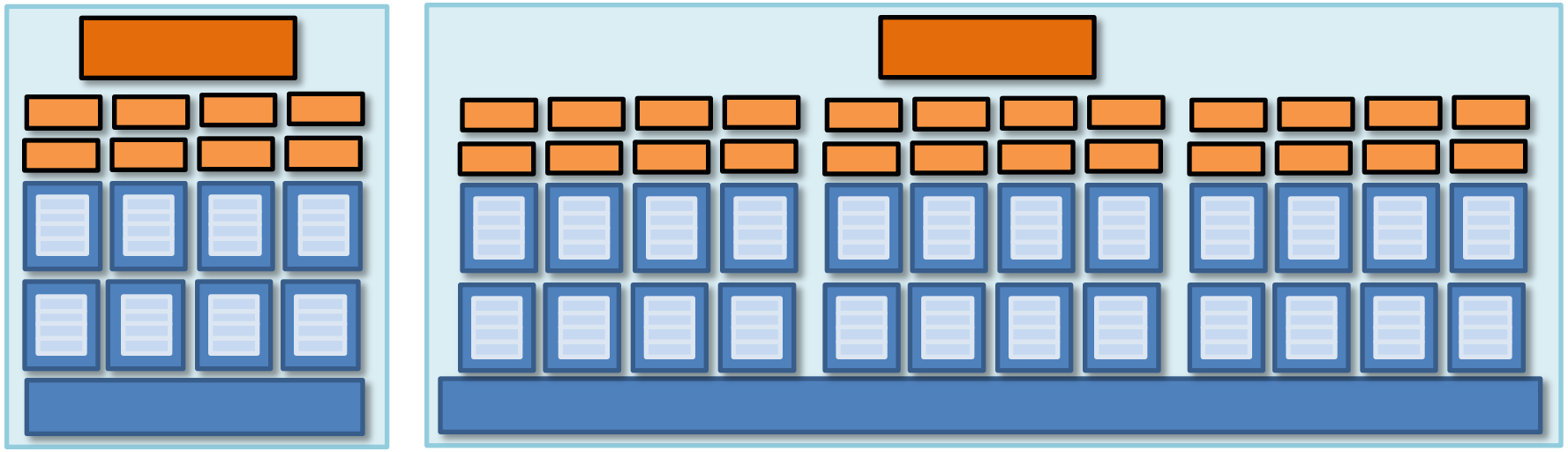
```
if (x > 0 {
    y = pow(x, exp);
    y *= Ks;
    refl = y + Ka;
} else {
    x = 0;
    refl = Ka;
}
```

Nem mindegyik ALU végez hasznos munkát
Legrosszabb eset 1/8

ÚJ SZÉCHENYI TERV
MAGYARORSZÁG MEGÚJUL
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

SIMD feldolgozás a gyakorlatban

16 és 64 fragmens között osztozik egy utasítás folyamon



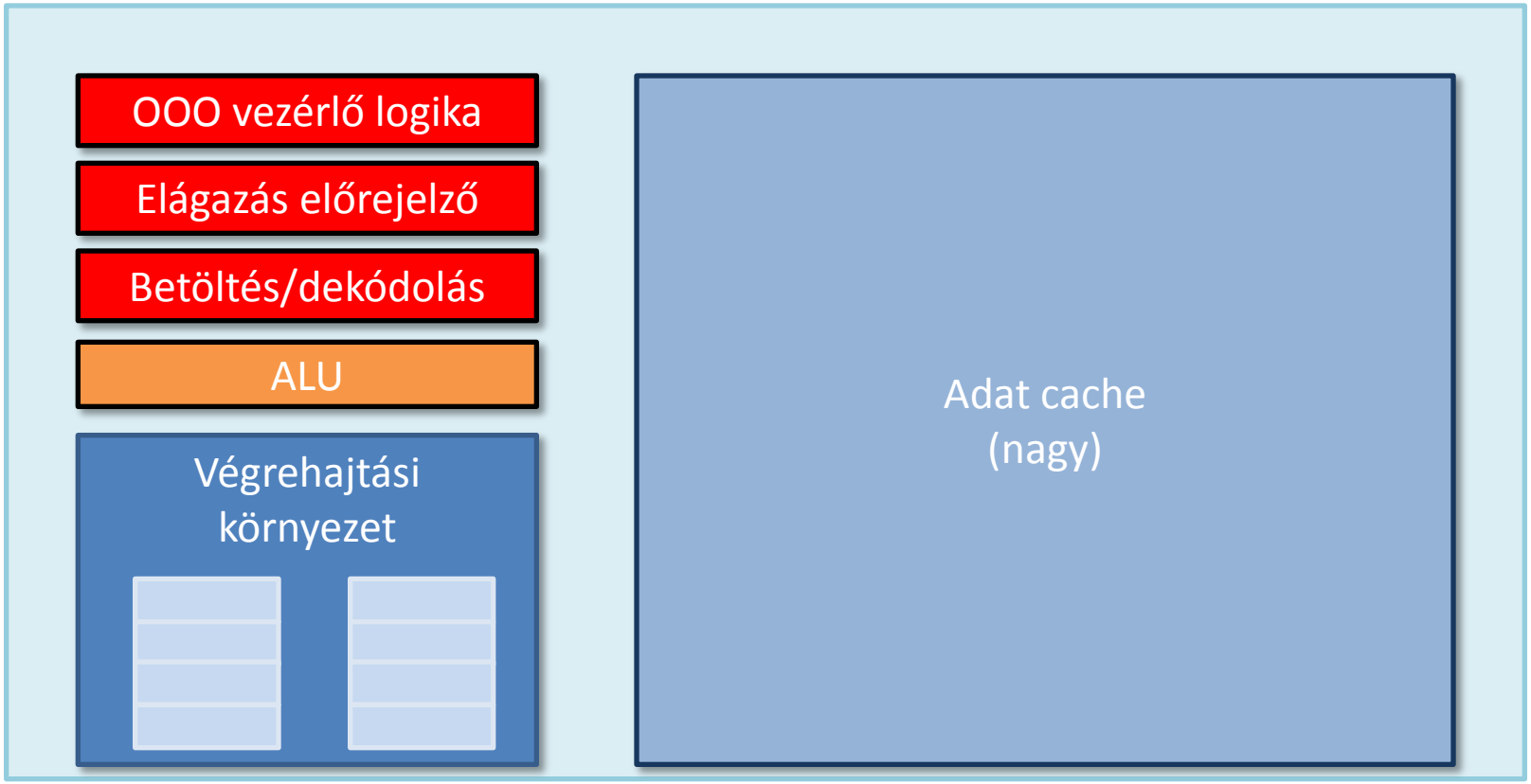
ÚJ SZÉCHENYI TERV
MAGYARORSZÁG MEGÚJUL
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Állások (stalls)

- Állás akkor következik be, amikor egy mag (core) nem tudja futtatni a következő shader utasítást, mivel egy előző utasításra várakozik
  - Függőségek vannak az utasítás folyamban
    - Pl. ADD függ a LOAD befejezésétől
- Késleltetés
  - Adat elérése a memóriából sokszor 1000-nél több ciklust igényel
  - Rossz ötlet volt az első egyszerűsítés?
    - Az eltávolított részek segítenek az állások megoldásában
  - A GPU-k sok független feladatot tételeznek fel
    - Független SIMD csoportok

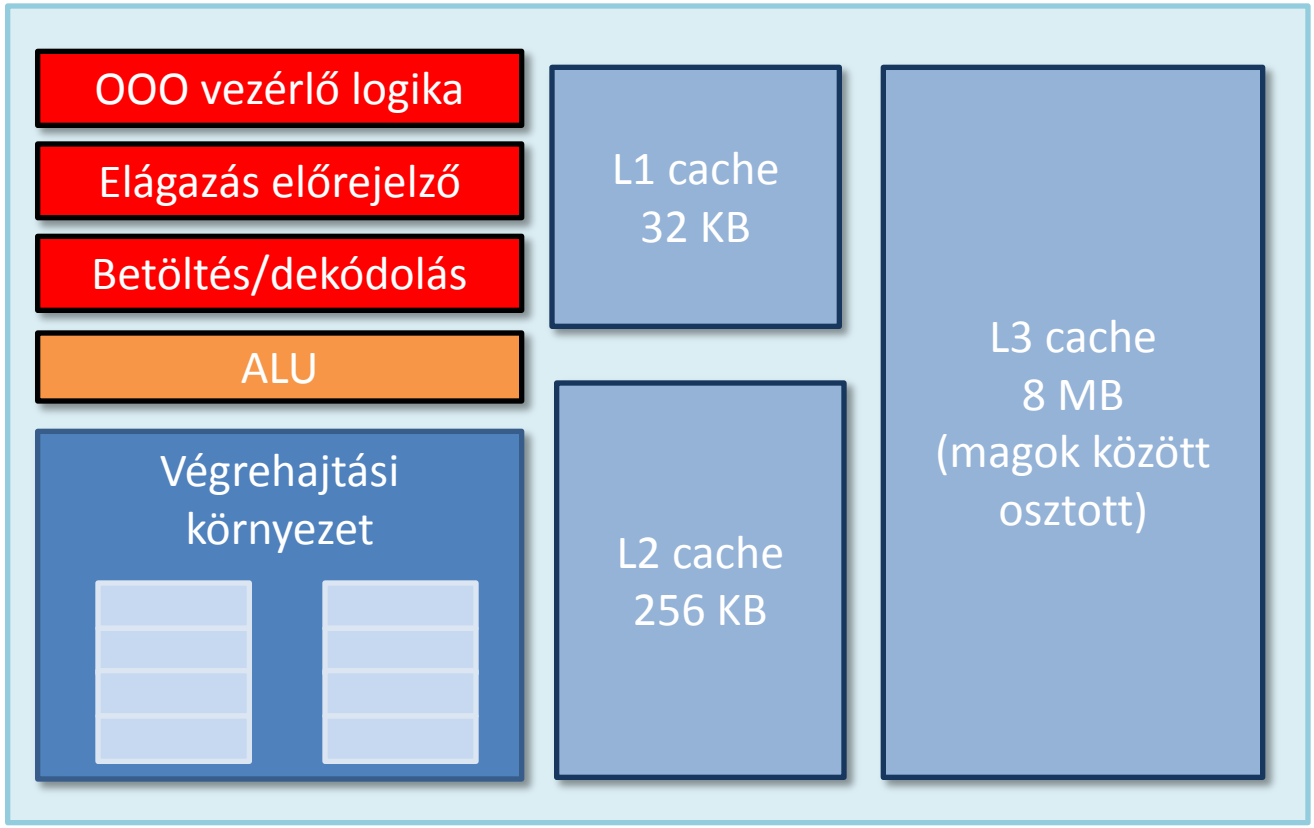
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

CPU-stílusú core-ok



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

CPU-stílusú memória felépítés



25 GB/sec elérés a memóriához

Magok hatékony kihasználása cache-ben lévő adatok esetén (késleltetés csökkentése, nagy sávszélesség)

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

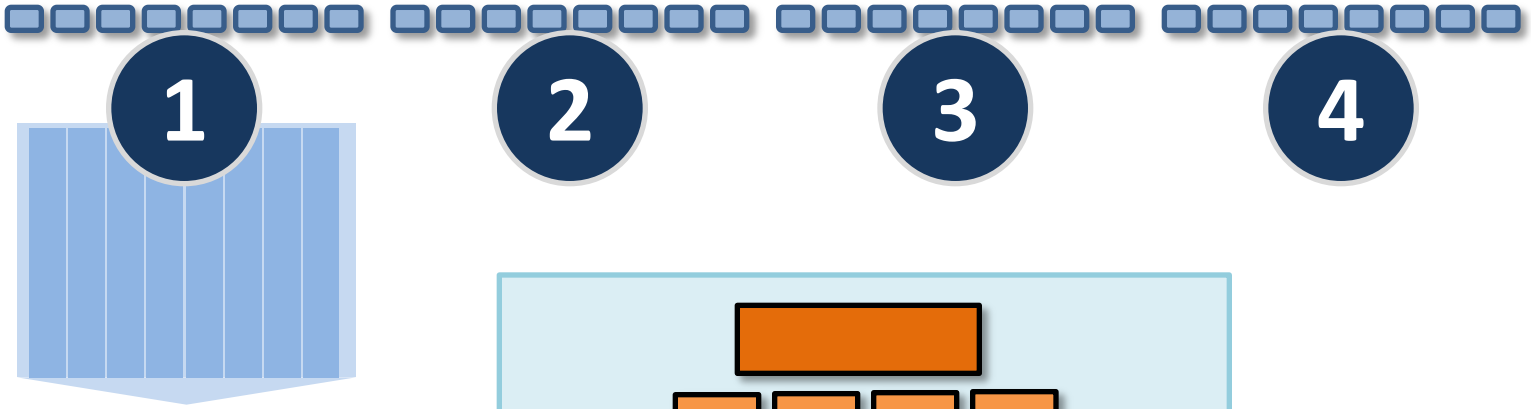
Harmadik ötlet

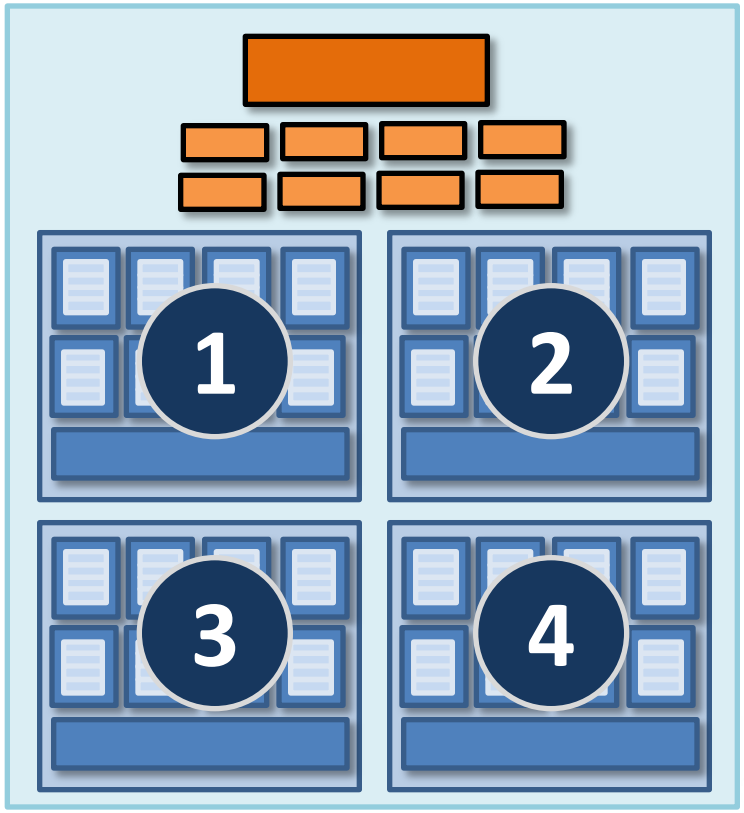
- Sok független fragmensünk van
- Sok fragmens összefésült feldolgozása egy magon
  - Utasítás folyam váltás egy másik (nem álló) SIMD csoportra, ha az aktív csoport áll
  - GPU hardveresen kezeli
    - Overhead mentesen
    - Ideális esetben teljesen láthatatlan
    - Áteresztőképesség maximális

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

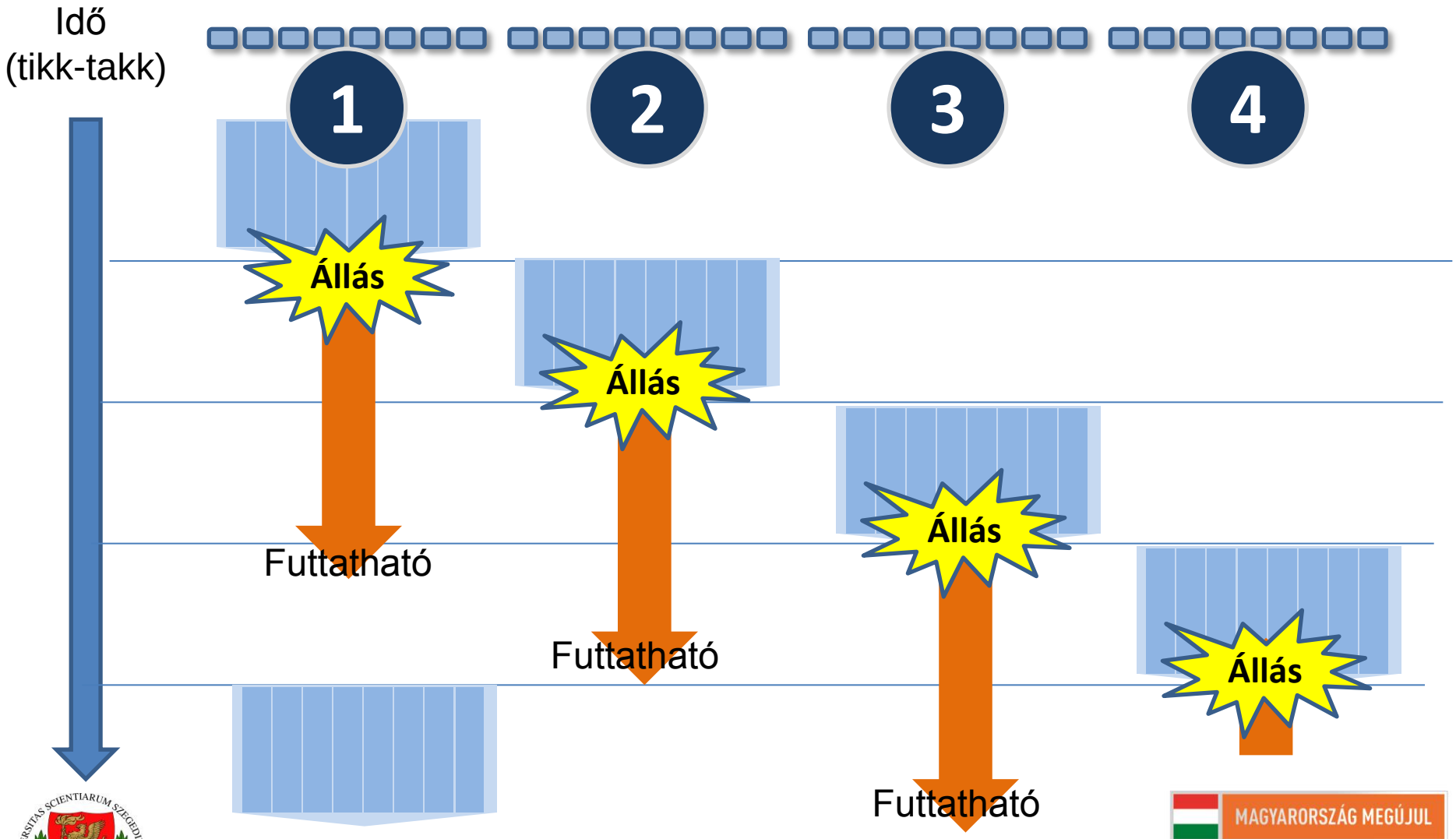
Shader állások elrejtése

Idő
(tikk-takk)

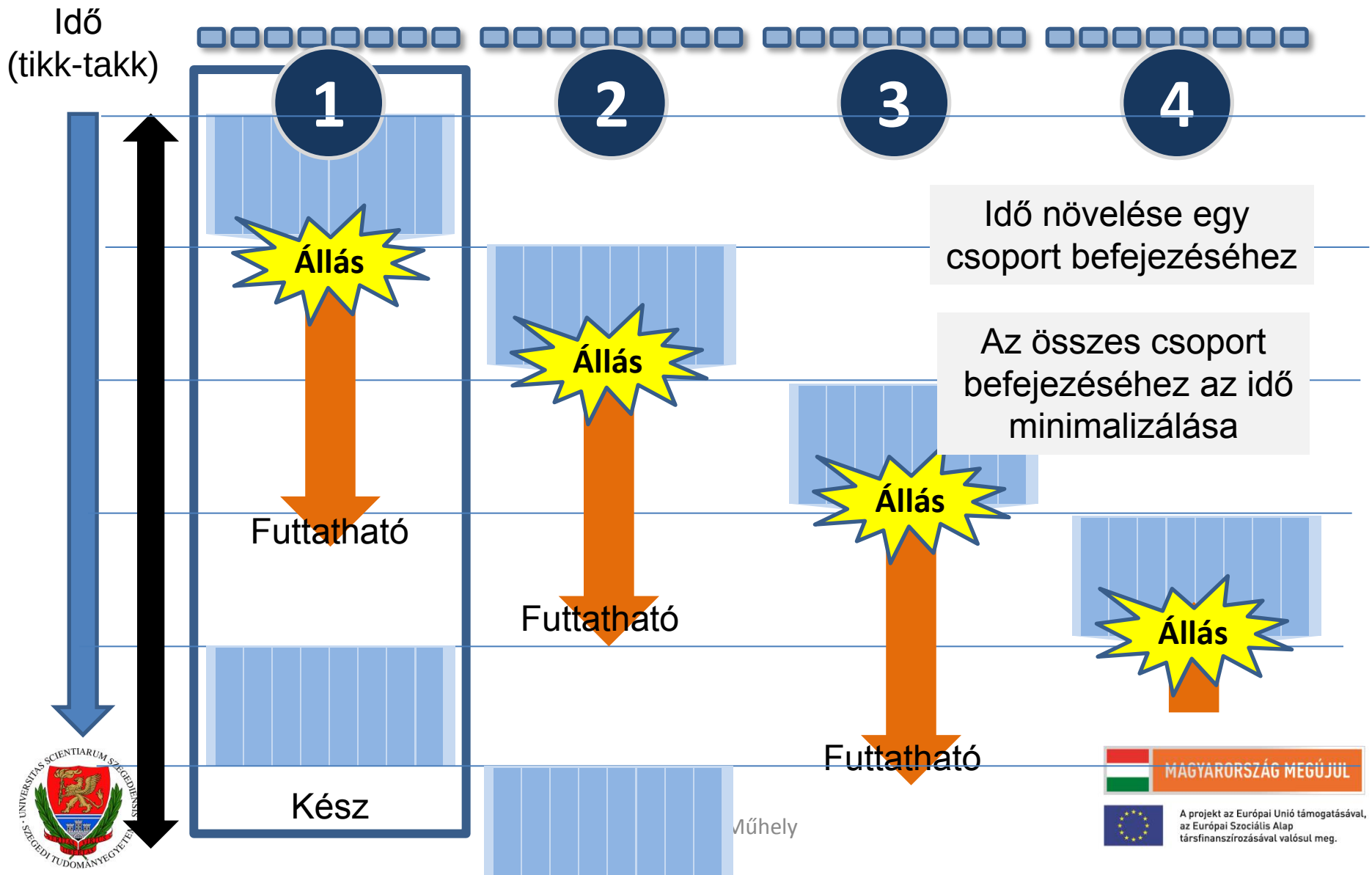




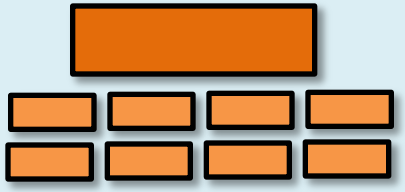
Shader állások elrejtése



Shader állások elrejtése



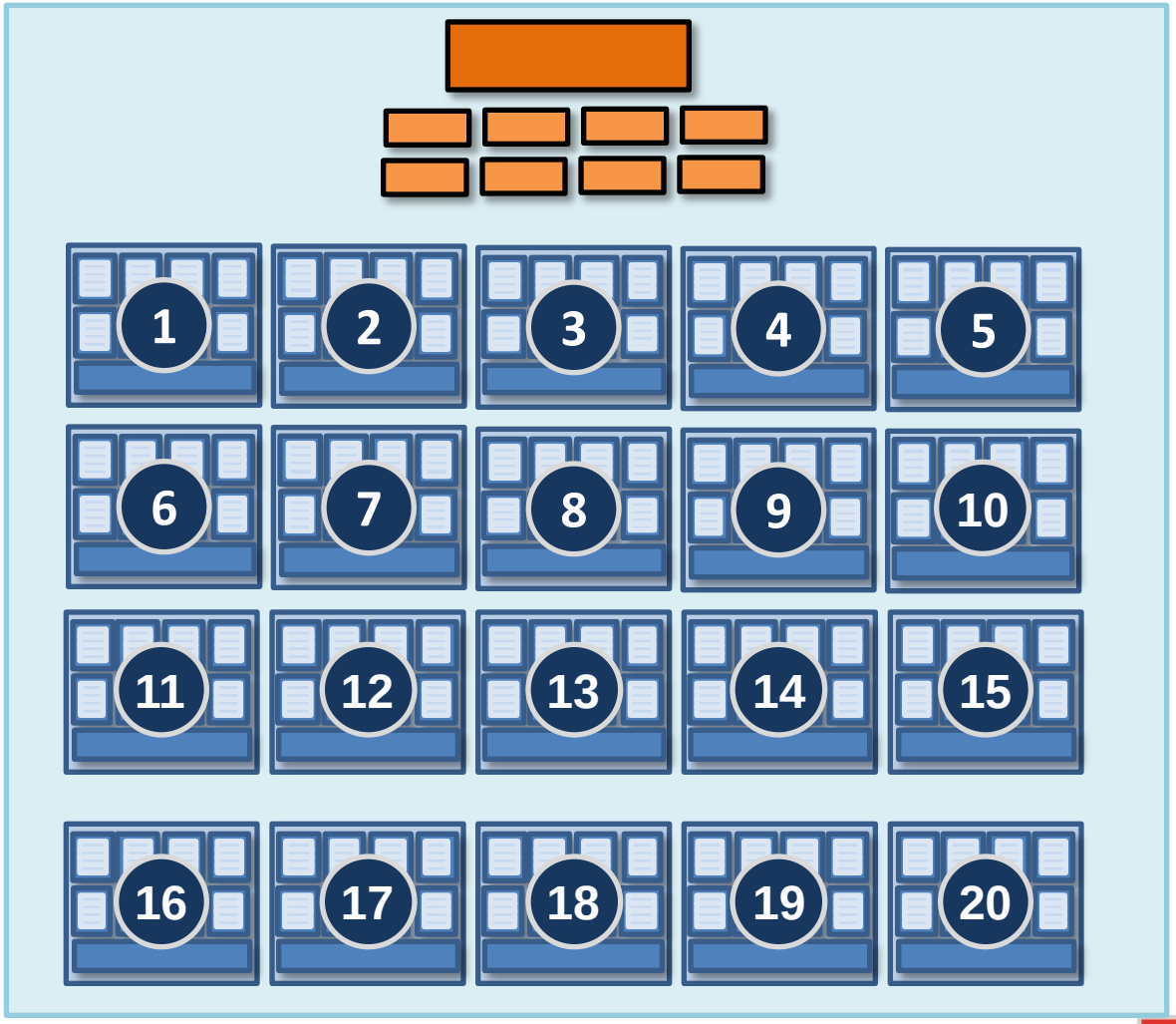
Környezetek tárolása



Közös környezet készlet tárolás
64 KB

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

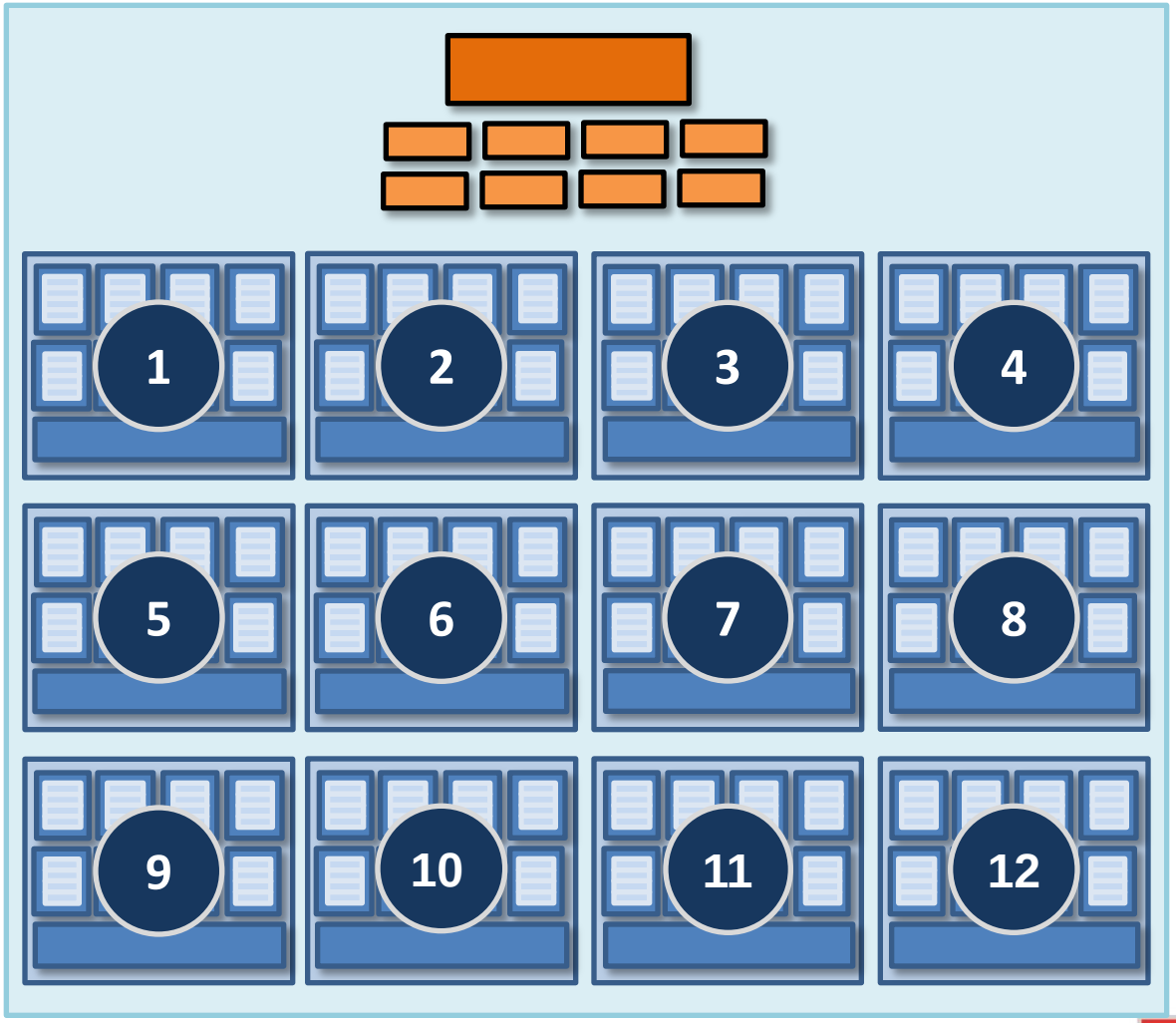
Húsz kicsi környezet



Maximális késleltetés elrejtési képesség

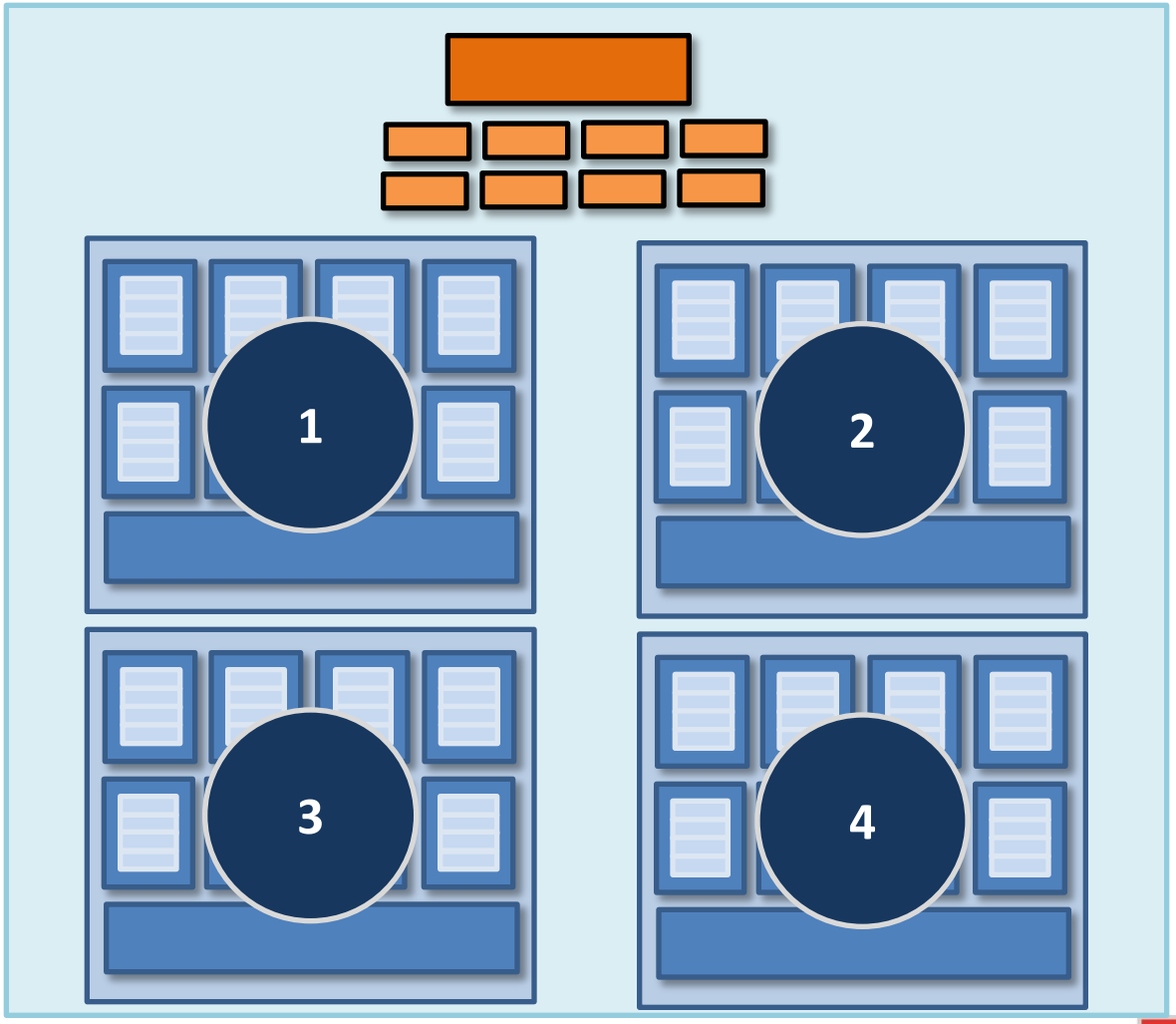
ÚJ SZÉCHENYI TERV
MAGYARORSZÁG MEGÚJUL
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Tizenkét közepes környezet



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Négy nagy környezet



Alacsony késleltetés elrejtési képesség

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

GPU shading rendszer

16 mag

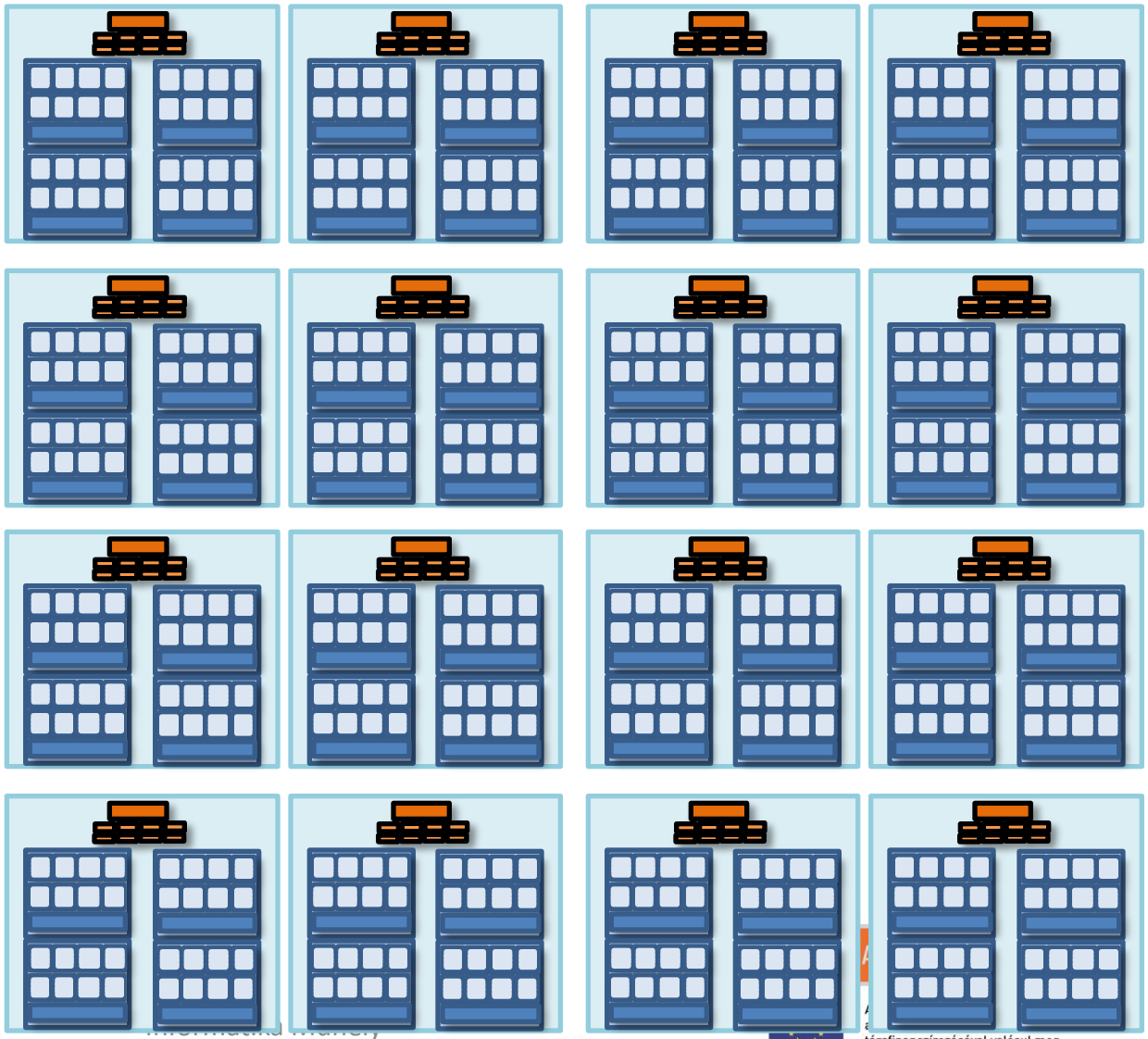
8 mul-add ALU magonként (128 összesen)

16 egyidejű utasítás folyam

64 konkurens (de összefésült) utasítás folyam

512 konkurens fragmens

=256 GFLOPS (@1 GHz)



társfinanszírozásával valósul meg.

Shader mag összefoglalása: három kulcs ötlet

- Használjunk sok karcsúsított magot a párhuzamos futtatáshoz
- A magokat rakjuk tele ALU-kkal
  - Megosztott utasítás folyamokkal fragmensek csoportjainál
- Kerüljük el a késleltetett állásokat több fragmens csoport összefésült végrehajtásával
  - Amikor az egyik csoport áll, akkor dolgozzunk egy másik csoporton

ÚJ SZÉCHENYI TERV
MAGYARORSZÁG MEGÚJUL
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Emlékezzünk erre!

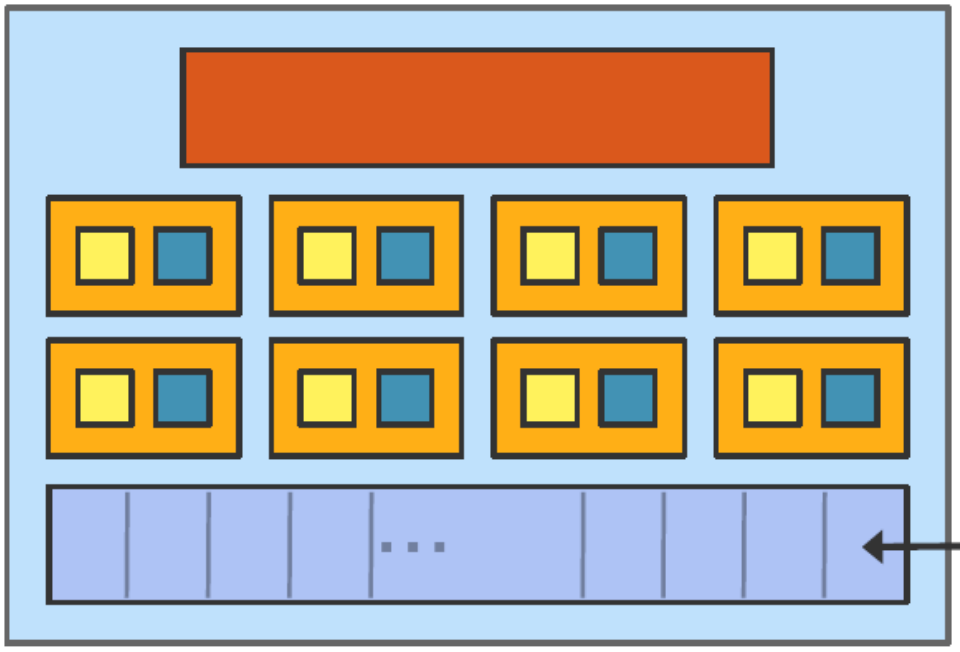
- A GPU-ra úgy gondoljunk, mint egy több-magos processzor, amely arra optimalizáltak, hogy a vertex és fragmens programok maximális áteresztéssel fussanak
- Speciálisan támogatja
  - a raszterizálást, vágást, hátsó oldal eltávolítást, textúrázást…
  - valamint a grafikus csővezeték leképezését ezekre az erőforrásokra

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

NVIDIA GeForce GTX 285

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

NVIDIA GeForce GTX 285 mag



64 KB fragmens környezetek tárolására (regiszterek)

= SIMD funkcionális egység, 8 egység osztott vezérlése

= utasítás folyam dekódolása

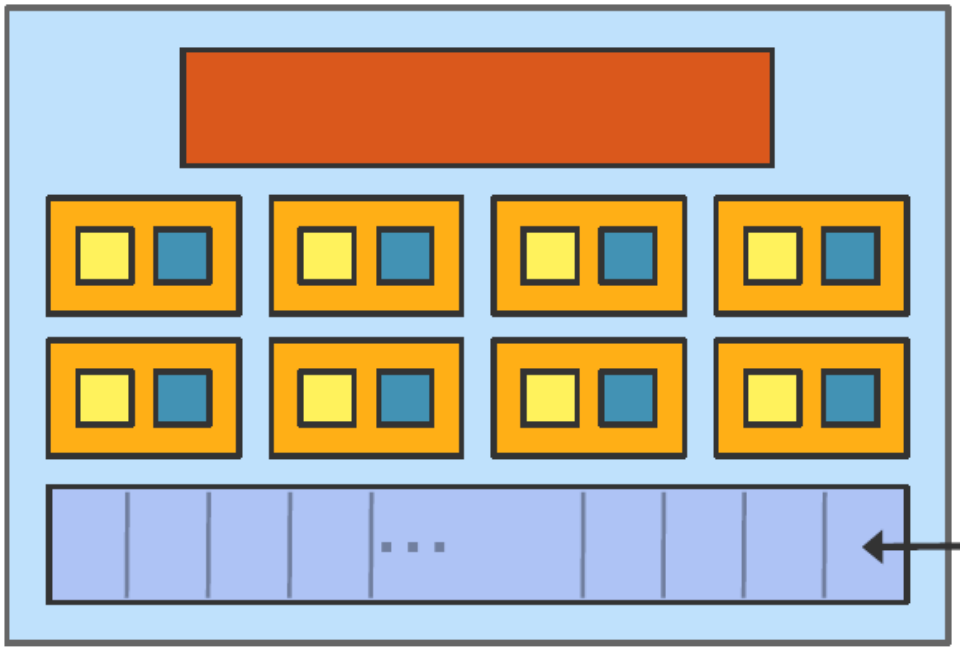
= szorzás - összeadás

= szorzás

= végrehajtási környezet tárolás

ÚJ SZÉCHENYI TERV
MAGYARORSZÁG MEGÚJUL
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

NVIDIA GeForce GTX 285 mag

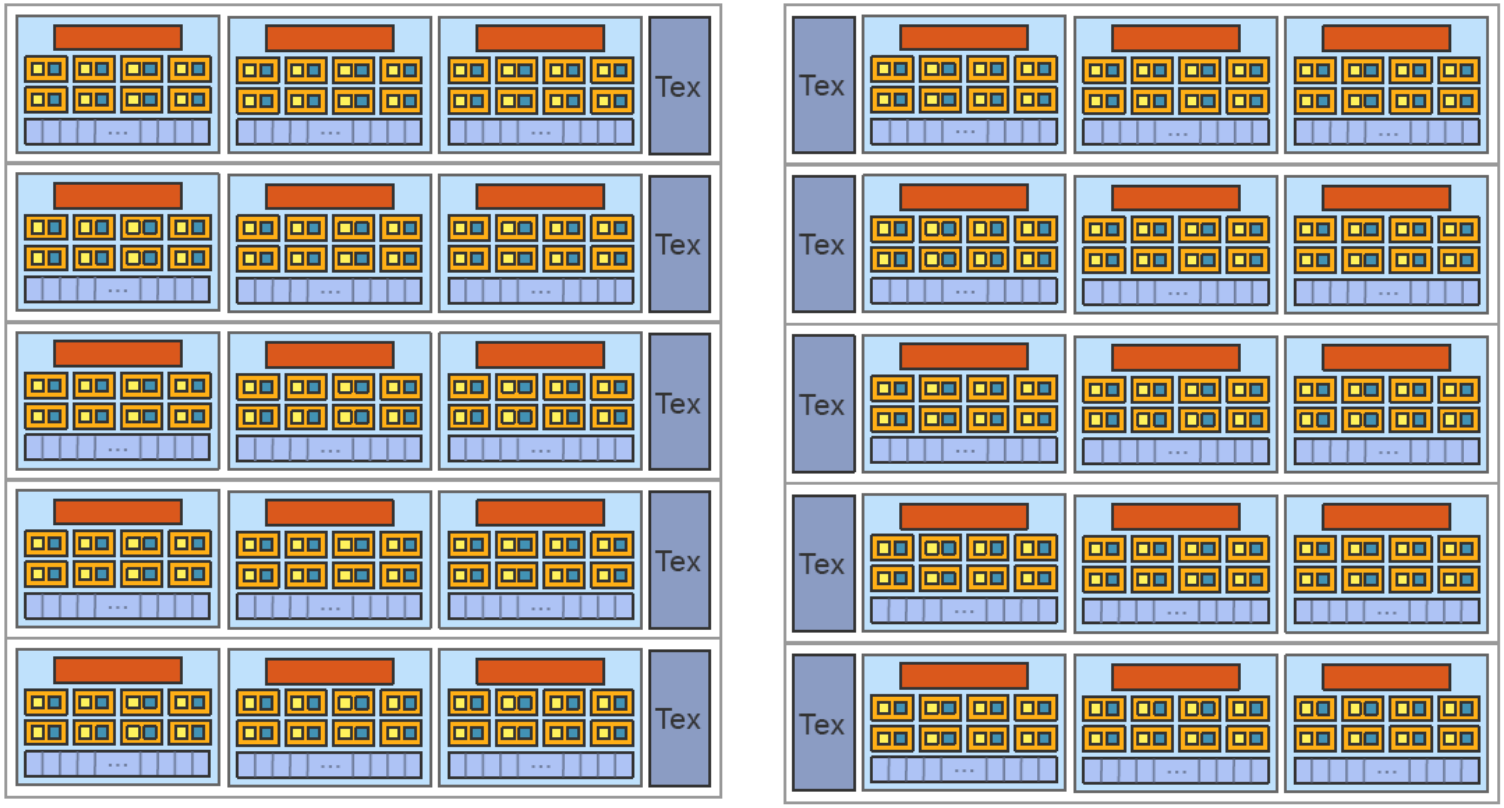


64 KB fragmens környezetek tárolására (regiszterek)

- 32 fragmens/vertex/primitív csoportok osztott utasítás folyammal (WARPS)
- Legfeljebb 32 csoport egyidejű összefésülése
  - Ezért legfeljebb 1024 fragmens környezetet lehet tárolni

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

NVIDIA GeForce GTX 285



- 30 ilyen van
  - 30 000 fragmens

ÚJ SZÉCHENYI TERV
MAGYARORSZÁG MEGÚJUL
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Mostani és jövőbeli GPU

Architektúrák

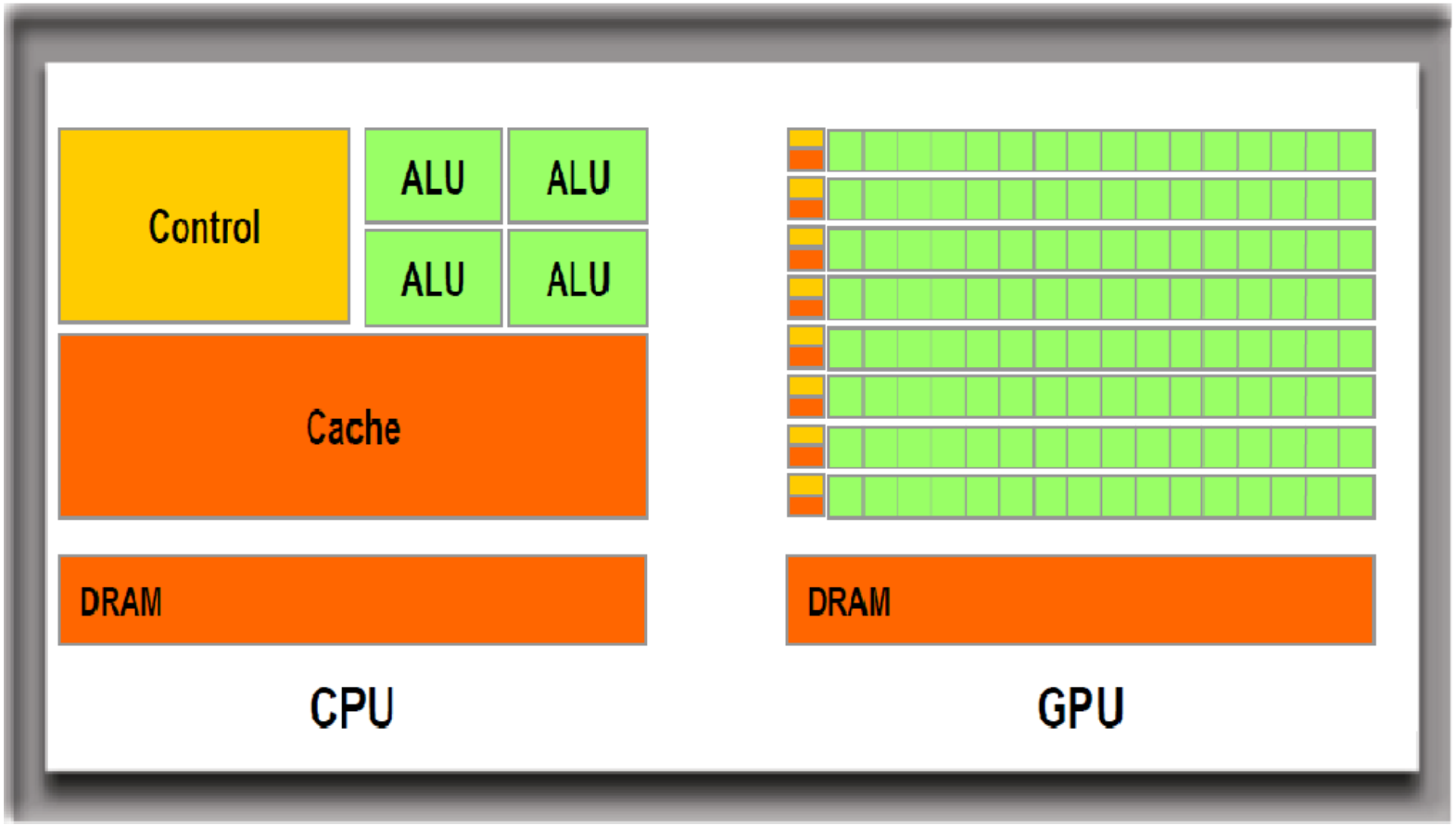
- Nagyobb és gyorsabb
  - Több mag és FLOPS
  - Manapság 2 TFLOPS
- Milyen fix-funkciójú hardvernek kell maradnia?
- Néhány CPU-hoz hasonló tulajdonság hozzáadása
  - Általános R/W cache (Fermi)
  - Szinkronizálás?

Programozása

- Alternatív programozási felületek támogatása
  - Nem-grafikus programozás
    - CUDA, OpenCL, DirectCompute
  - Alkalmazások, amelyek a GPU-t több-processzoros rendszernek tekintik
- Hogyan változik a grafikus csővezeték absztrakció?
  - Direct3D 11 3 új csővezeték szakasz
  - Sugár-követés

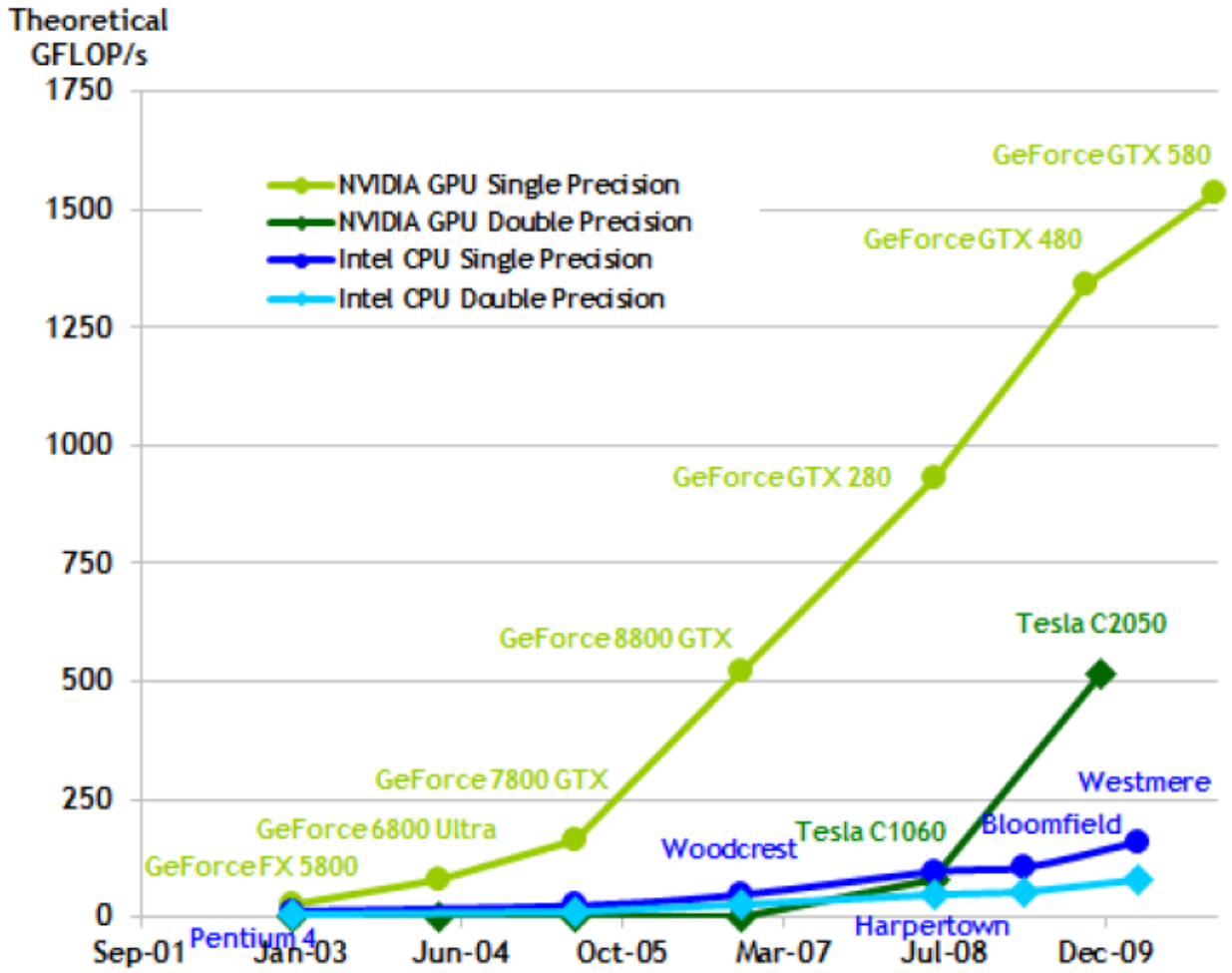
ÚJ SZÉCHENYI TERV
MAGYARORSZÁG MEGÚJUL
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

CPU-GPU összehasonlítás



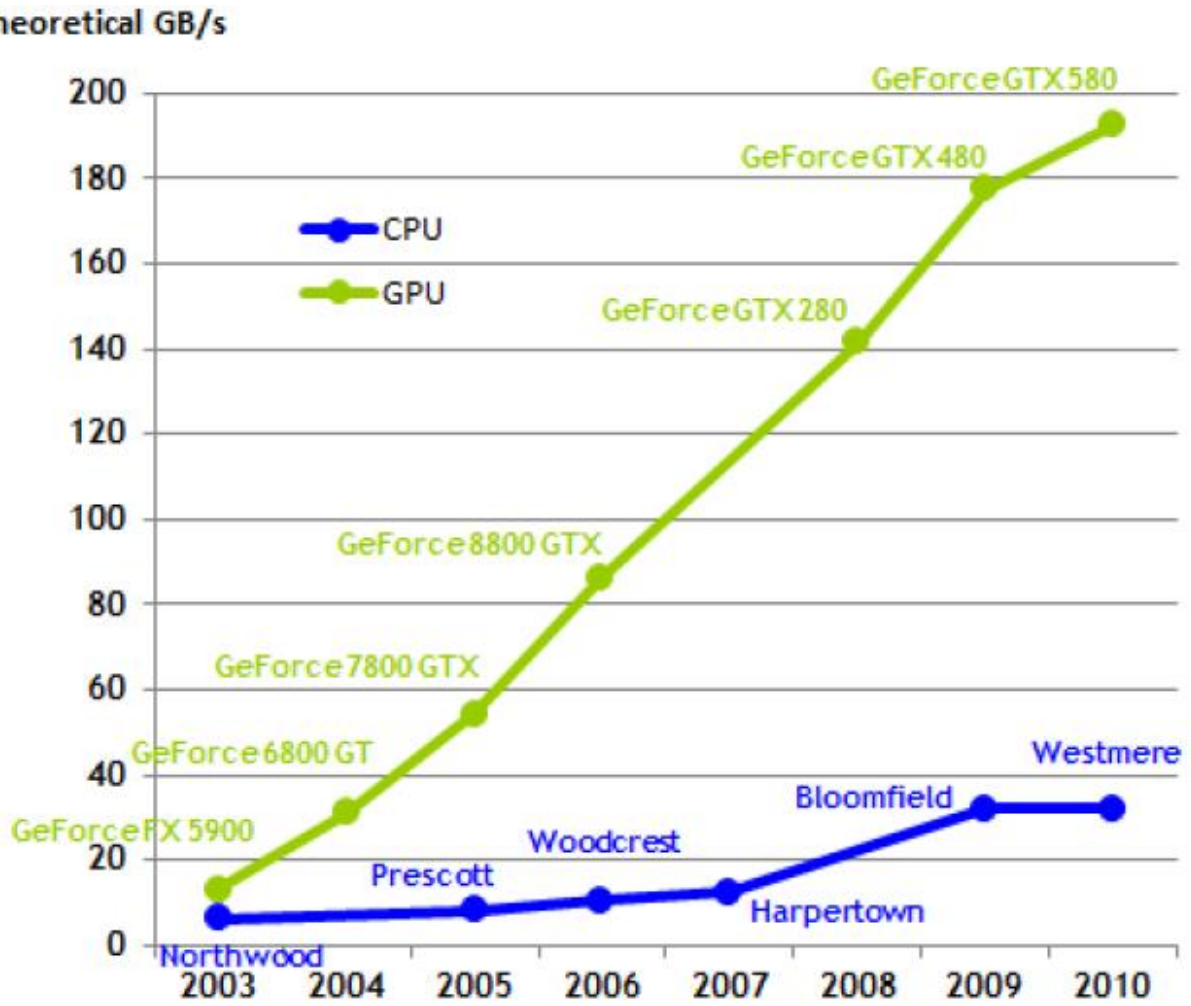
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Teljesítmény FLOPS



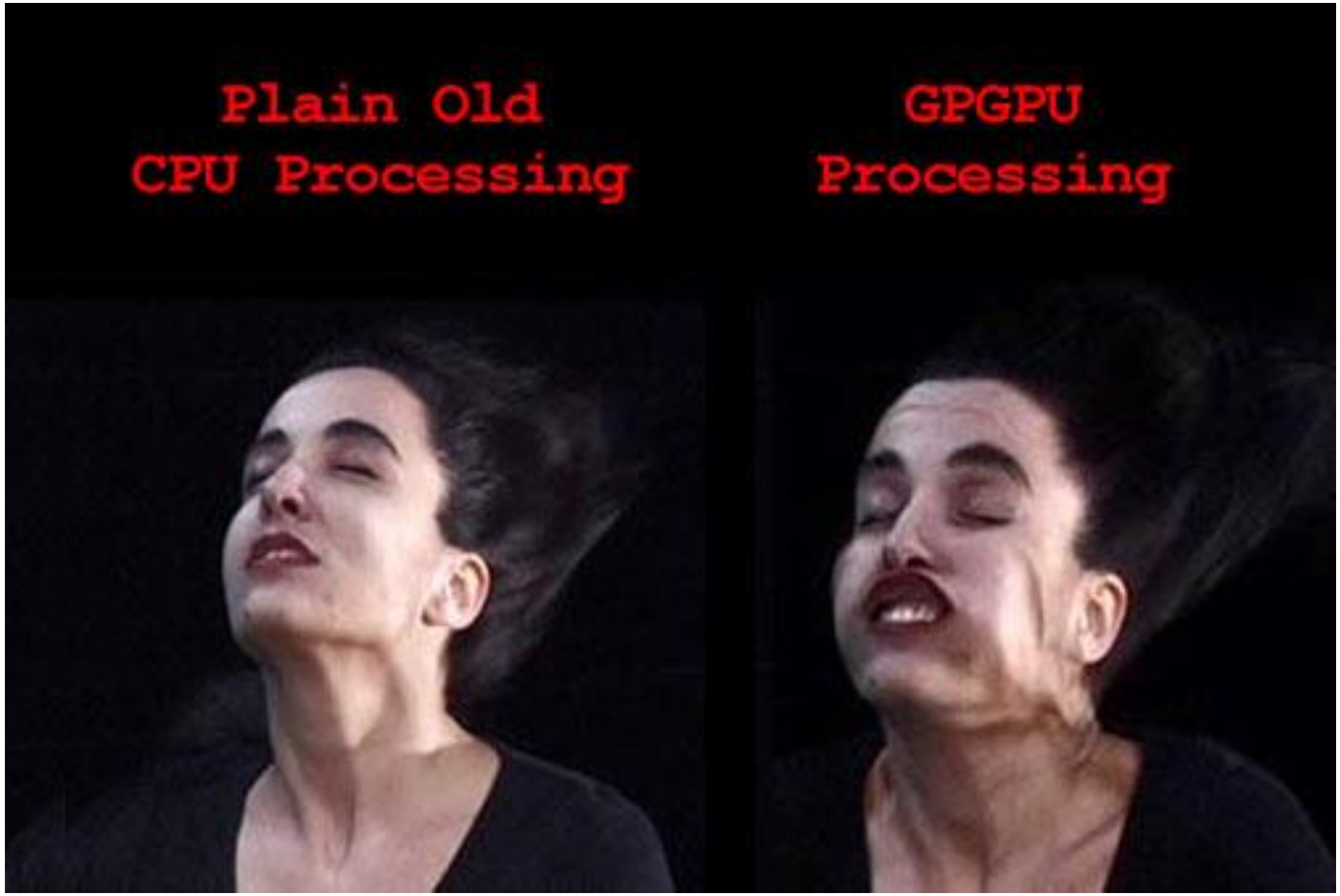
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Teljesítmény memória sávszélesség



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

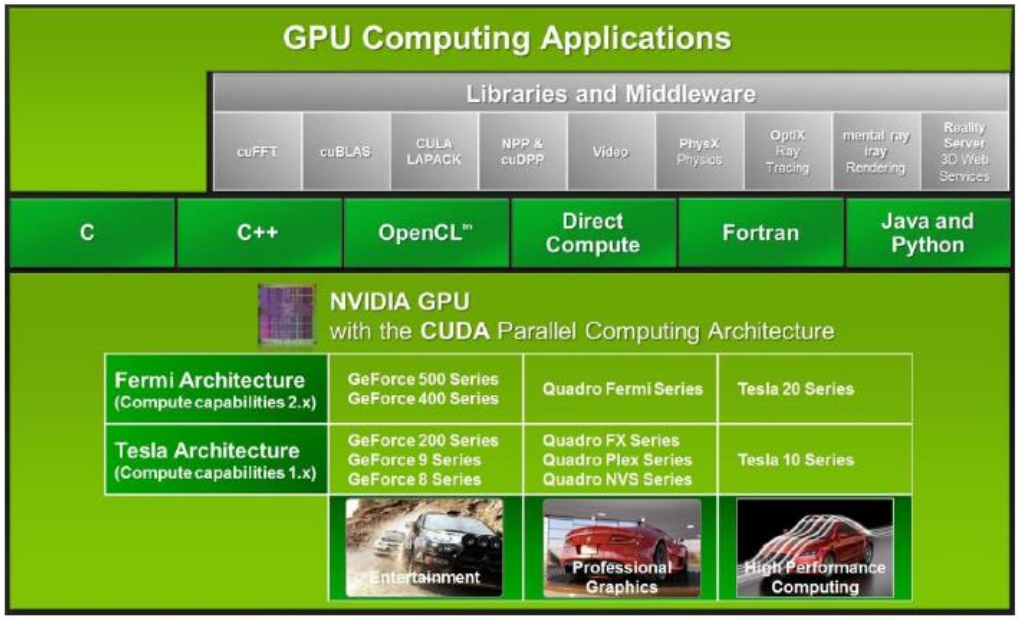
Teljesítmény



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

CUDA

- CUDA
  - Compute
  - Unified
  - Device
  - Architecture
- Programozási modell és utasítás halmaz Szoftveres környezettel
  - C nyelven
  - Magas szintű nyelv

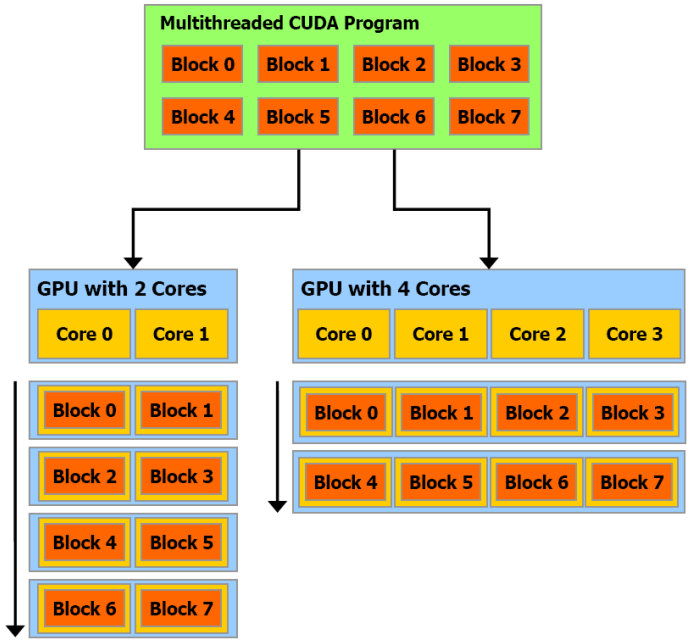


A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

CUDA API

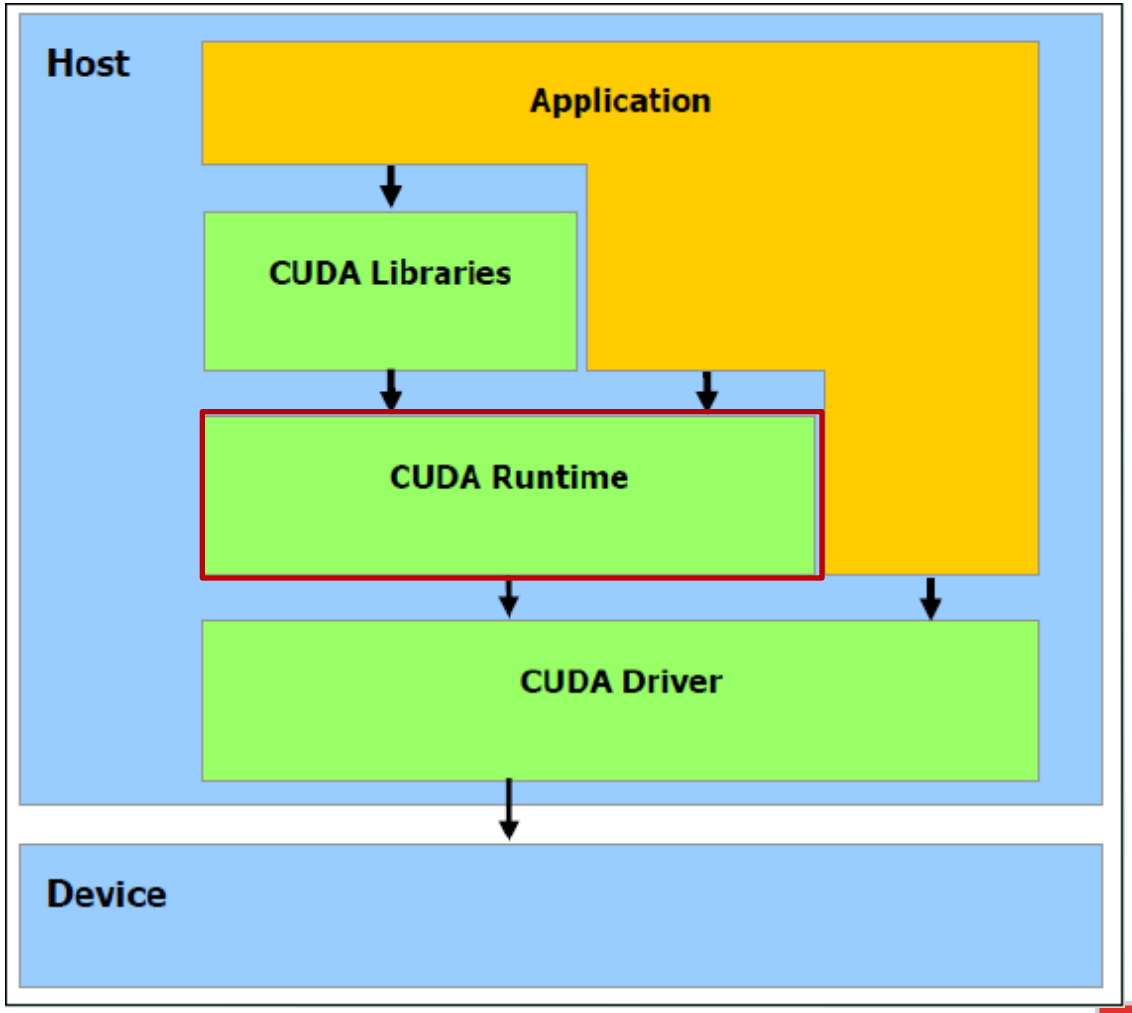
- Fejlesztés
  - Nvidia meghajtó
  - CUDA fordító
  - CUDA debugger
  - CUDA profiler
  - CUDA SDK
- Függvénykönyvtárak
  - FFT, BLAS
- C/C++ nyelv
  - Gazda (host)
  - eszköz (device)
- Más nyelveken is lehetséges
  - Fortran

- Skálázható programozási modell



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

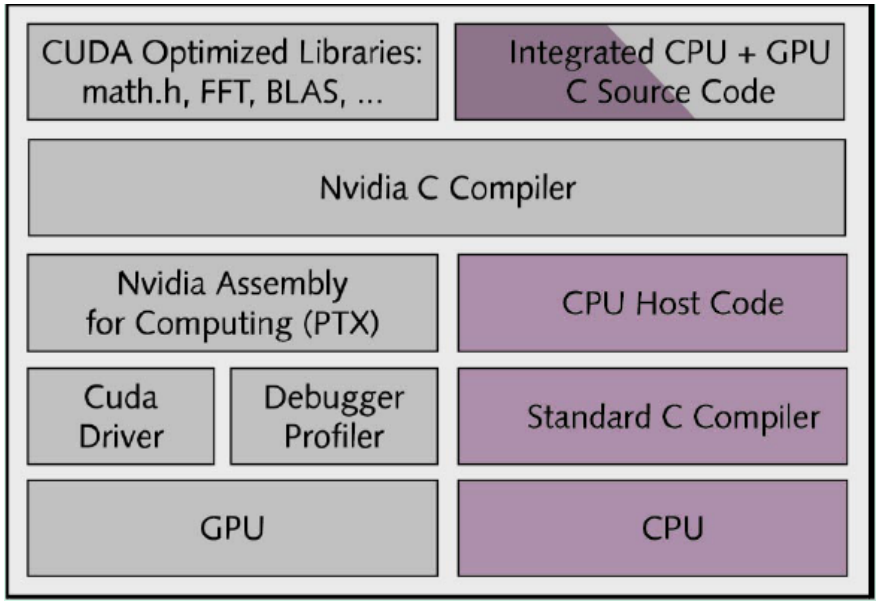
CUDA



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

CUDA

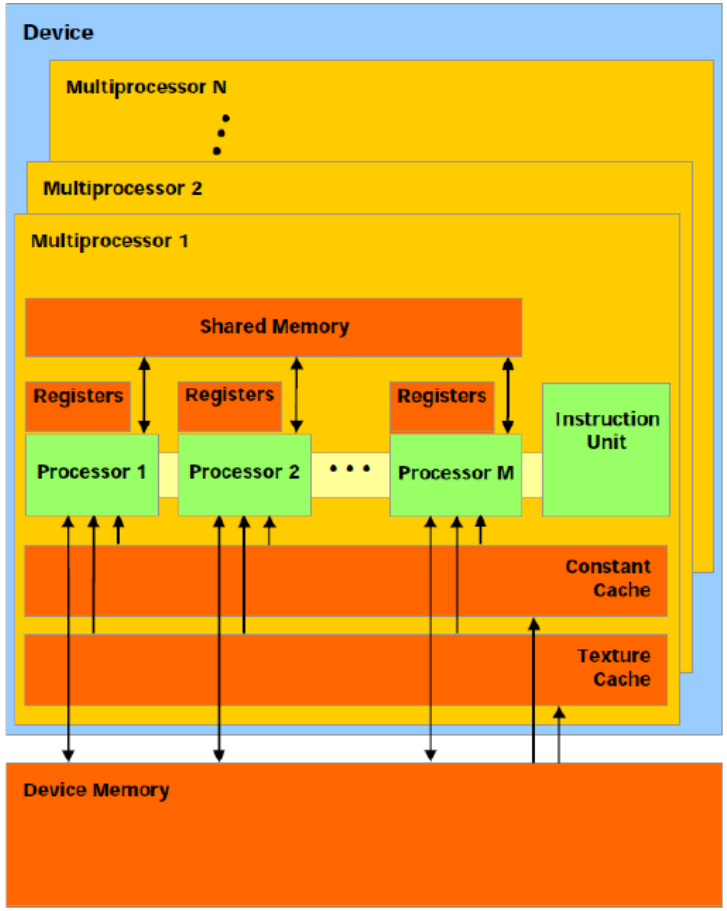
- A forráskód a gazda és az eszköz kódját is tartalmazza
  - Fordító választja szét
  - CPU kódot egy külső C fordító értelmezi



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Memória modell

- Különböző memória típusok
  - Osztott memória
  - Textúra cache
  - Konstans cache
  - Eszköz memória



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Rács, blokkok és szálak

- Kernel-ek
  - C függvények
  - N-szer párhuzamosan lesz végrehajtva N különböző CUDA szállal

```
// Kernel definition
__global__ void VecAdd(float* A, float* B, float* C)
{
    int i = threadIdx.x;
    C[i] = A[i] + B[i];
}

int main()
{
    ...
    // Kernel invocation with N threads
    VecAdd<<<1, N>>>(A, B, C);
}
```


A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Rács, blokkok és szálak

- Szál hierarchia
  - **`threadIdx`**
    - 3 komponensű vektor
    - 1D, 2D és 3D szál index
    - 1D-ben ugyanaz
    - 2D-ben egy 2D-s (D_x, D_y)
      - (x, y)
        - » ($x+y\ D_x$)
    - 3D-ben egy (D_x, D_y, D_z)
      - (x, y, z)
        - » ($x+y\ D_x+z\ D_x\ D_y$)

```
// Kernel definition
__global__ void MatAdd(float A[N][N], float B[N][N],
                       float C[N][N])
{
    int i = threadIdx.x;
    int j = threadIdx.y;
    C[i][j] = A[i][j] + B[i][j];
}

int main()
{
    ...
    // Kernel invocation with one block of N * N * 1 threads
    int numBlocks = 1;
    dim3 threadsPerBlock(N, N);
    MatAdd<<<numBlocks, threadsPerBlock>>>(A, B, C);
}
```

- Limitált blokkonkénti szál szám
  - Mindegyik blokk szál ugyanazon a magon kell lennie és mag memória mérete is korlátozott

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

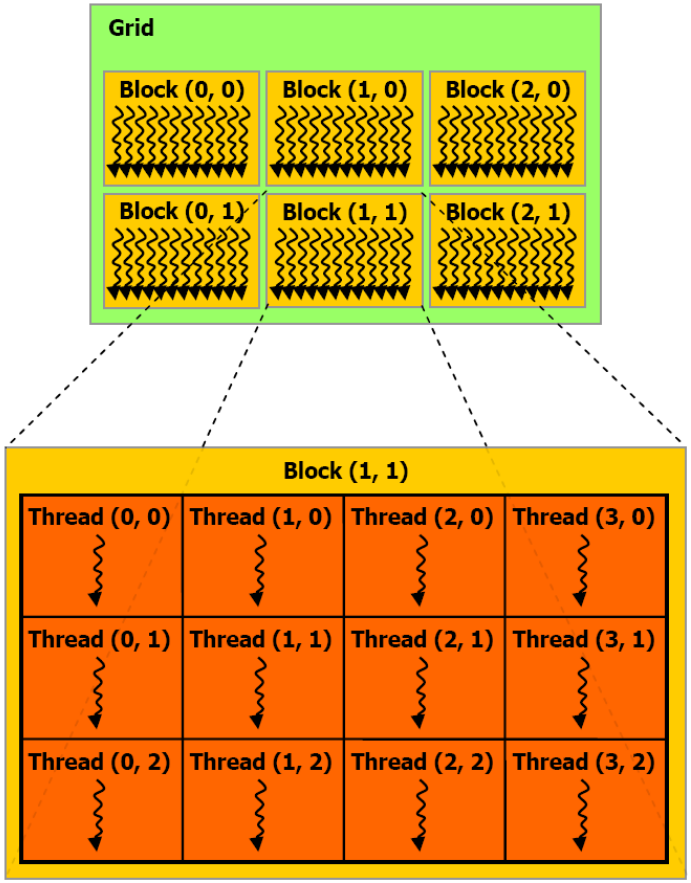
Rács, blokk, szál

- A kernelt végre lehet hajtani több szál blokkon
  - A szálak száma megegyezik a blokkonkénti szálak számával szorzott blokk számmal
- A blokkok 1D-s, 2D-s és 3D-s rácsokba vannak szervezve
  - A blokkok száma függ
    - a feldolgozandó adat mennyiségétől
    - Processzorok számától

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Rács, blokkok és szálak

- Blokkonkénti szálak és a rácsonkénti blokkok száma
  - `<<<…>>>`
  - `int` vagy `dim3`
- Mindegyik blokk 1D-s, 2D-s vagy 3D-s indexeléssel érhető el
  - `blockIdx`



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Rács, blokkok és szálak

- Egymástól függetlenül kell végrehajtani a szál blokkokat
  - Párhuzamosan
- Szinkronizációs pontok
  - Osztott memórián keresztüli adat kezelés
  - **`__syncthreads()`**
    - A blokkban lévő szálaknak várnia kell

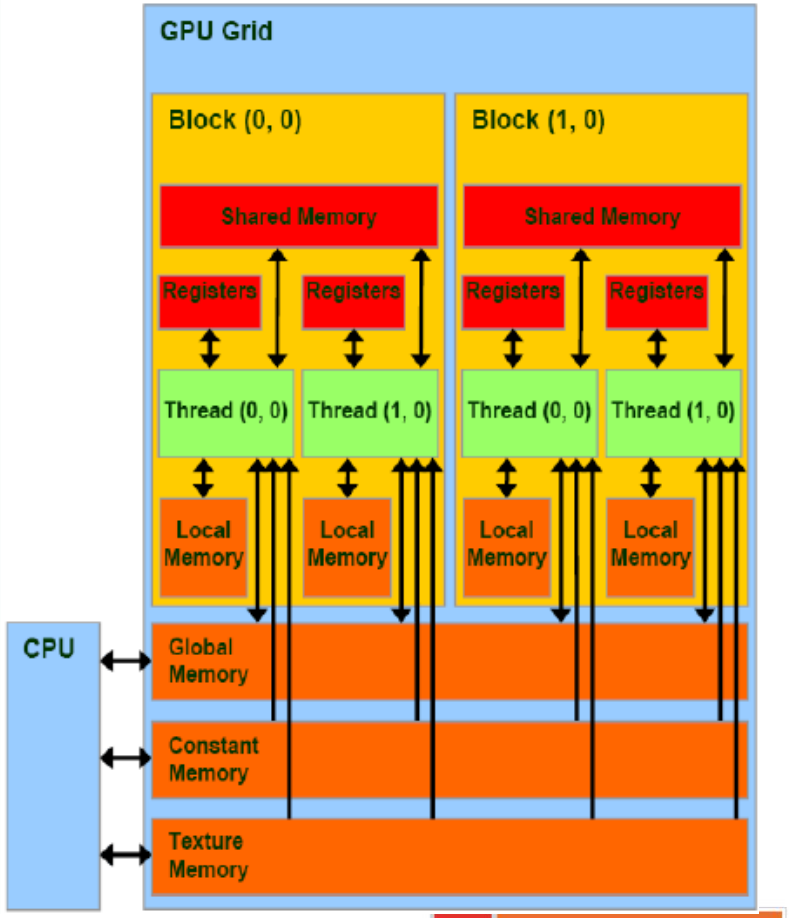
```
// Kernel definition
__global__ void MatAdd(float A[N][N], float B[N][N],
                       float C[N][N])
{
    int i = blockIdx.x * blockDim.x + threadIdx.x;
    int j = blockIdx.y * blockDim.y + threadIdx.y;
    if (i < N && j < N)
        C[i][j] = A[i][j] + B[i][j];
}

int main()
{
    ...
    // Kernel invocation
    dim3 threadsPerBlock(16, 16);
    dim3 numBlocks(N / threadsPerBlock.x, N / threadsPerBlock.y);
    MatAdd<<<numBlocks, threadsPerBlock>>>(A, B, C);
}
```


A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Memória hierarchia

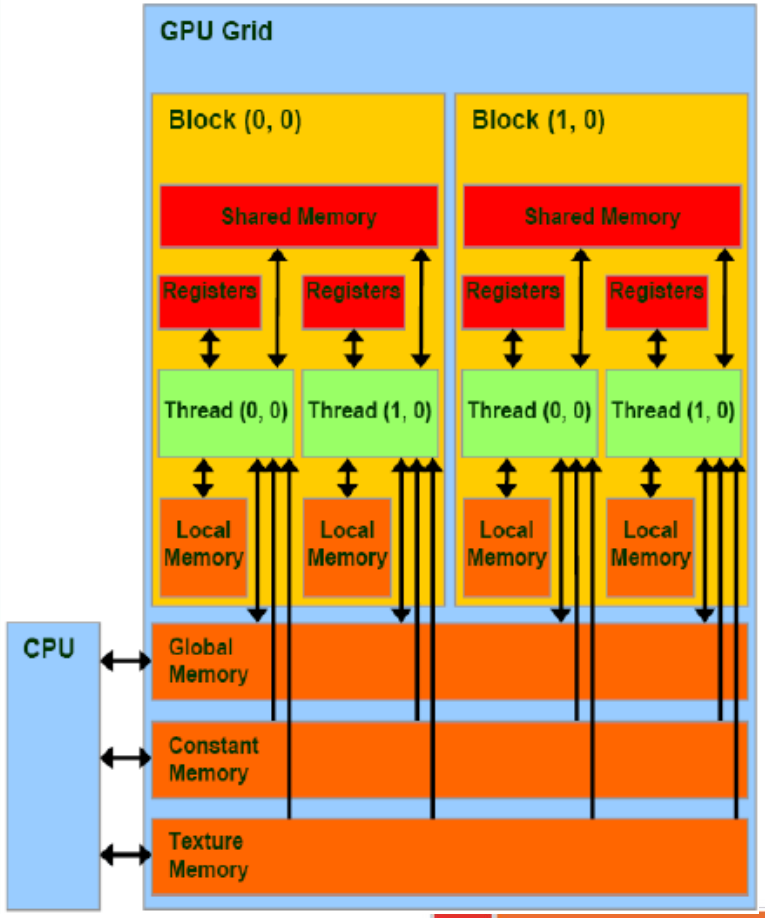
- Memóriaterületek
  - Szál
    - Saját regiszterek (RW)
    - Lokális memória (RW)
  - Blokk
    - Osztott (RW)
    - Konstans (R)
  - Rács
    - Globális (RW)
    - Textúra (R)
- Gazda
  - Globális, konstans, textúra



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Memória hierarchia

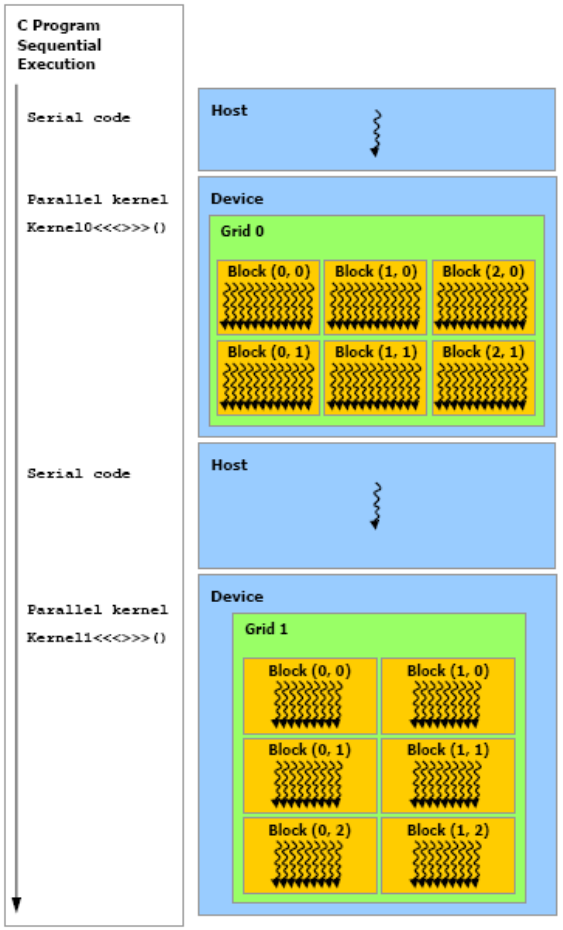
- Memóriaterületek másolása
  - **`cudaMemcpyHostToHost`**
  - **`cudaMemcpyHostToDevice`**
  - **`cudaMemcpyDeviceToHost`**
  - **`cudaMemcpyDeviceToDevice`**



ÚJ SZÉCHENYI TERV
MAGYARORSZÁG MEGÚJUL
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Heterogén programozás

- Egymást követő lépések
- Lehetőség van aszinkron futtatásra
  - Két végrehajtandó egység egy időben futtat programokat



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Lépések

- Párhuzamosítható részek megkeresése
- Érdemes már egy létező algoritmust átírni
  - A memória másolások minimalizálása
  - Kernel hívások minimalizálása
- A CPU és GPU kódok megvalósítása
  - Egymástól függetlenül

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

OpenCV

```
#include <highgui.h>
#include <opencv2/opencv.hpp>
#include <opencv2/gpu/gpu.hpp>
#include <opencv2/gpu/gpumat.hpp>

using namespace cv;
using namespace cv::gpu;

int main(int argc, char** argv)
{

    Mat frame, result;
    gpu::GpuMat img, binResult;
    vector<GpuMat> planes;
    double sum;
```


A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

OpenCV

```
VideoCapture cap(0);
  if(!cap.isOpened())
    return -1;
  for(;;)
  {
    cap >> frame;
                imshow("Original:", frame);
                img = frame;
                gpu::GpuMat smoothPlane0(img.size(), img.type());
                gpu::GpuMat smoothPlane1(img.size(), img.type());
                gpu::GpuMat smoothPlane2(img.size(), img.type());
                gpu::split(img, planes);
                gpu::GaussianBlur(planes[0], smoothPlane0, Size(9,9), 0);
                gpu::GaussianBlur(planes[1], smoothPlane1, Size(9,9), 0);
                gpu::GaussianBlur(planes[2], smoothPlane2, Size(9,9), 0);
                gpu::add(smoothPlane0, smoothPlane1, smoothPlane1);
                gpu::subtract(smoothPlane2, smoothPlane1, smoothPlane2);
                gpu::threshold(smoothPlane2, binResult, 20.0, 255.0, CV_THRESH_BINARY);
                sum = gpu::norm(binResult, NORM_L1);
  ...
```


A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Források

- http://people.maths.ox.ac.uk/gilesm/cuda/
- http://developer.nvidia.com/content/gpu-gems-part-i-natural-effects
- http://developer.nvidia.com/node/17
- http://http.developer.nvidia.com/GPUGems3/gpugems3_part01.html
- http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter30.html
- http://www.cse.ohio-state.edu/~crawfis/cse786/ReferenceMaterial/CourseNotes/Modern%20GPU%20Architecture.ppt
- http://www.stanford.edu/class/ee382a/
- **http://s09.idav.ucdavis.edu/talks/02_kayvonf_gpuArchTalk09.pdf**
- http://www.cis.upenn.edu/~suvenkat/700/
- http://sites.google.com/site/5kk70gpu/materials
- http://s08.idav.ucdavis.edu/luebke-nvidia-gpu-architecture.pdf
- http://www.ece.ubc.ca/~aamodt/papers/wwlfung.micro2007.pdf
- http://www.ece.ubc.ca/~aamodt/papers/gpgpusim.ispass09.pdf
- (http://forums.nvidia.com/lofiversion/index.php?t100074.html)
- http://users.ece.gatech.edu/lanterma/mpg/
- http://developer.amd.com/gpu_assets/Intermediate_Language_Specification--Stream_Processor.pdf
- http://www.google.com/patents/about?id=_jKoAAAAEBAJ
- http://www.google.com/patents/about?id=nsc0AAAAEBAJ
- http://hal.archives-ouvertes.fr/docs/00/37/47/15/PDF/stats_on_instruction.pdf
- http://graphics.stanford.edu/courses/cs448a-01-fall/
- https://graphics.stanford.edu/wikis/cs448s-10/
- **https://graphics.stanford.edu/wikis/cs448s-10/FrontPage?action=AttachFile&do=get&target=05-GPU_Arch_I.pdf**
- **http://panoramix.ift.uni.wroc.pl/~maq/cuda/prezentacja-cuda-eng.pdf**
- **http://nik.uni-obuda.hu/app/APP02.pdf**
- **http://developer.download.nvidia.com/compute/cuda/3_0/toolkit/docs/NVIDIA_CUDA_ProgrammingGuide.pdf**

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.