Double-precision Floating-point on recent GPU

Juraj Starinsky

Overview

- Short Glossary
- Recent GPU
- Architecture overview
- Vector vs Scalar processing
- Theoretical processing power
- Languages and compilation
- Graphics API

Short Glossary

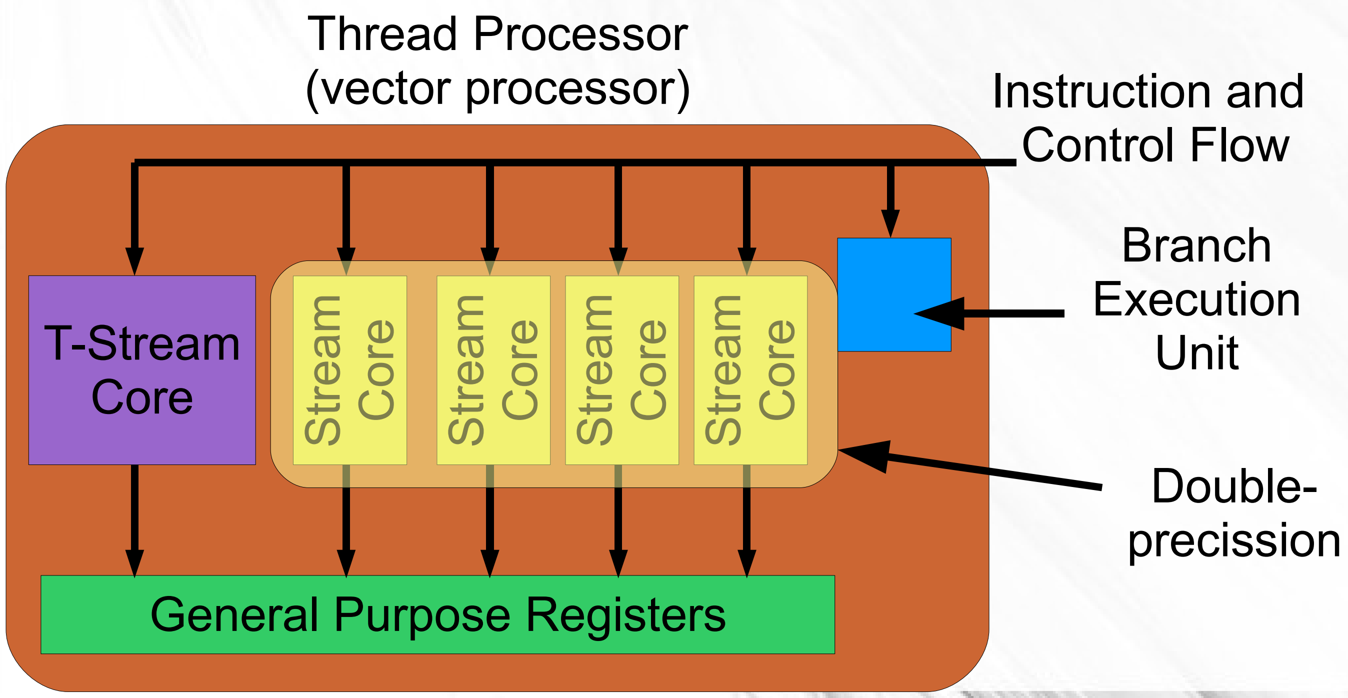
- Stream – a collection of data elements of the same type that can be operated on in parallel
- Stream core – ALU for add, sub, mul, div on int and float
- Kernel – parallel function operating on all elements of input stream
- Thread – one invocation of a kernel corresponding to a single element
- FP32 – 32bit single precision floating point
- FP64 – 64bit double precision floating point

Recent GPU generation

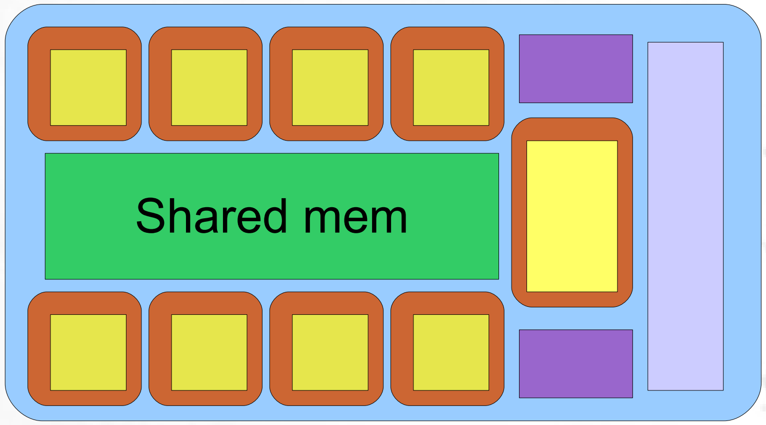
- AMD-ATI
  - RV700 GPU
  - Radeon HD 4000
    - 43x0
    - 45x0
    - 46x0
    - 4830
    - 4850
    - 4870
    - 4870x2
- NVIDIA
  - GT200 GPU
  - GeForce GTX 200
    - GTX260
    - GTX280

AMD Radeon HD 4000

- 4x FP32 core or 1x FP64 “core”
- 1x Transcendental core (sin,cos,log,...)



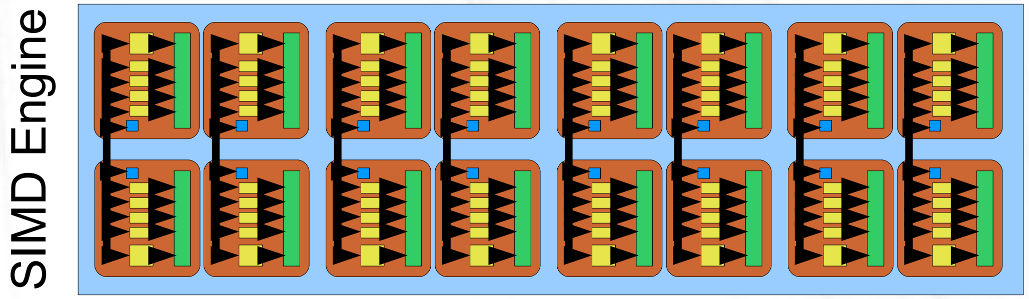
NVIDIA GTX 200

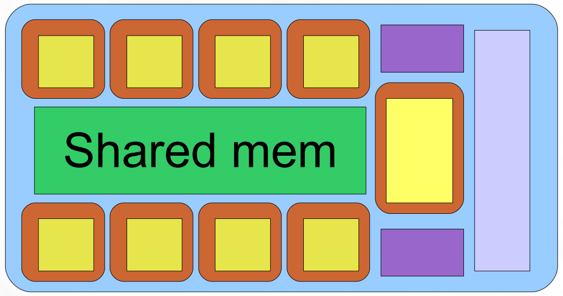


- Multi Processor
  - 8x FP32 scalar core
  - 1x FP64 scalar core
  - 2x SFU
    - Special function unit (sin, cos, log,...)
  - 1x multithreaded instruction unit
  - 16kb Shared mem
  - SIMD execution

SIMD engine vs Multiprocessor

- ATI SIMD engine
  - 16x Thread Vector Processors
  - 80x FP32
  - 16x FP64
  - 4x Texture units
- NVIDIA Multiprocessor
  - 9x Thread scalar processors
  - 8x FP32
  - 1x FP64
  - 16kB shared mem

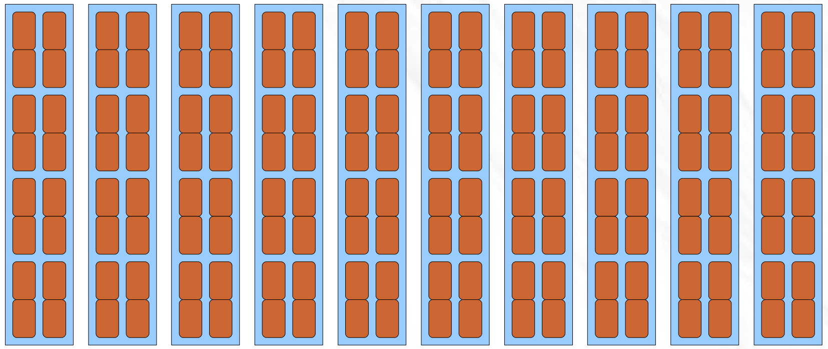


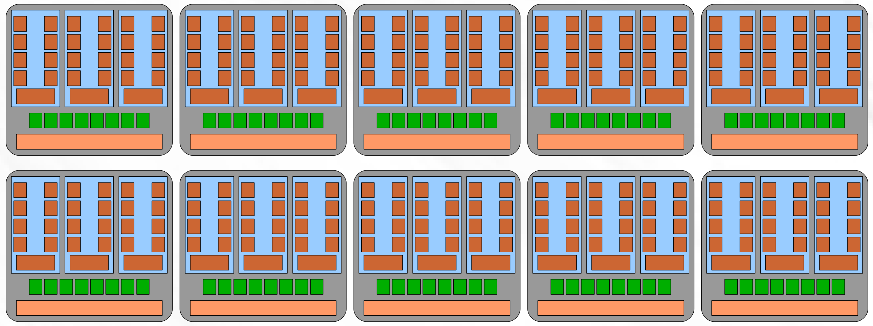


HD 4780 vs GTX 280

- 10x SIMD engines
  - independent
  - 10x16=160 FP64
  - 10x80=800 FP32

- 30x Multiprocessors
  - independent
  - 30x1 FP64
  - 30x8=240 FP32





Vector vs Scalar processing

- More instructions
  - vector and swizzling operations
  - slower decoding
  - faster execution
    - Shorter kernels
- Wasting power
  - Scalar operations
  - Manual vectorization

- Less instruction
  - Faster decoding
  - Slower execution
    - Longer kernels
- No vectorization needed

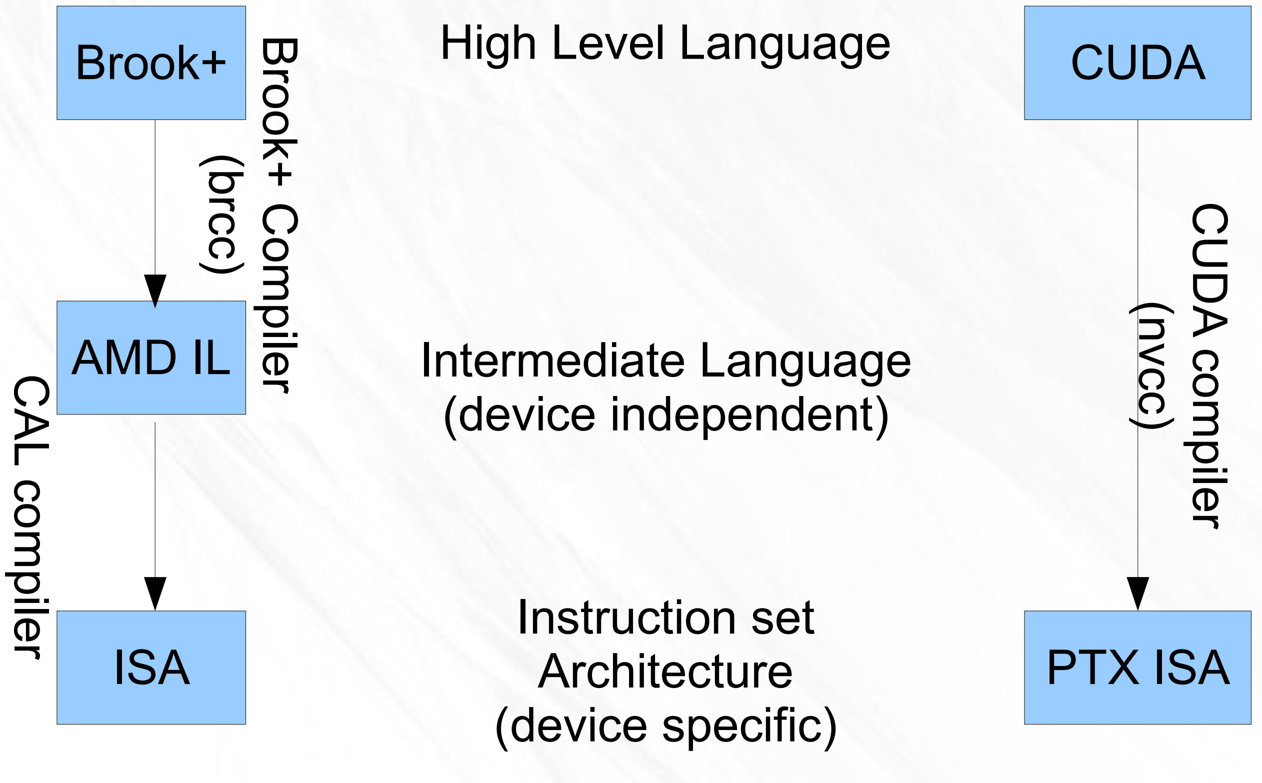
Computational Power

	GPU	Price (euro)	MHz	FP64 #cores	FP64 GFLOPS	FP32 #cores	FP32 GFLOPS
AMD-ATI	**HD 4870**	**260**	**750**	**160**	**120-240**	**640 (800)**	**480-960 (600-1200)**
nVIDIA	**GTX 280**	**380**	**1296**	**30**	**39-78**	**240**	**311-933**

- MAD – mul & add = “2 instructions”
- NVIDIA – 1x MAD + 1xMUL = 3 instructions
- AMD-ATI – 1x MAD

Language hierarchy

- AMD Stream computing
- NVIDIA CUDA



High Level Language

- AMD Brook+
  - C extension
  - Kernel
    - kernel void brook_kernel(args)
  - Device data (stream)
    - double a<20,30>

- NVIDA CUDA
  - C extension
  - Kernel
    - __global__ void cuda_kernel(args)
  - Device data
    - __device__ double a[20][30];

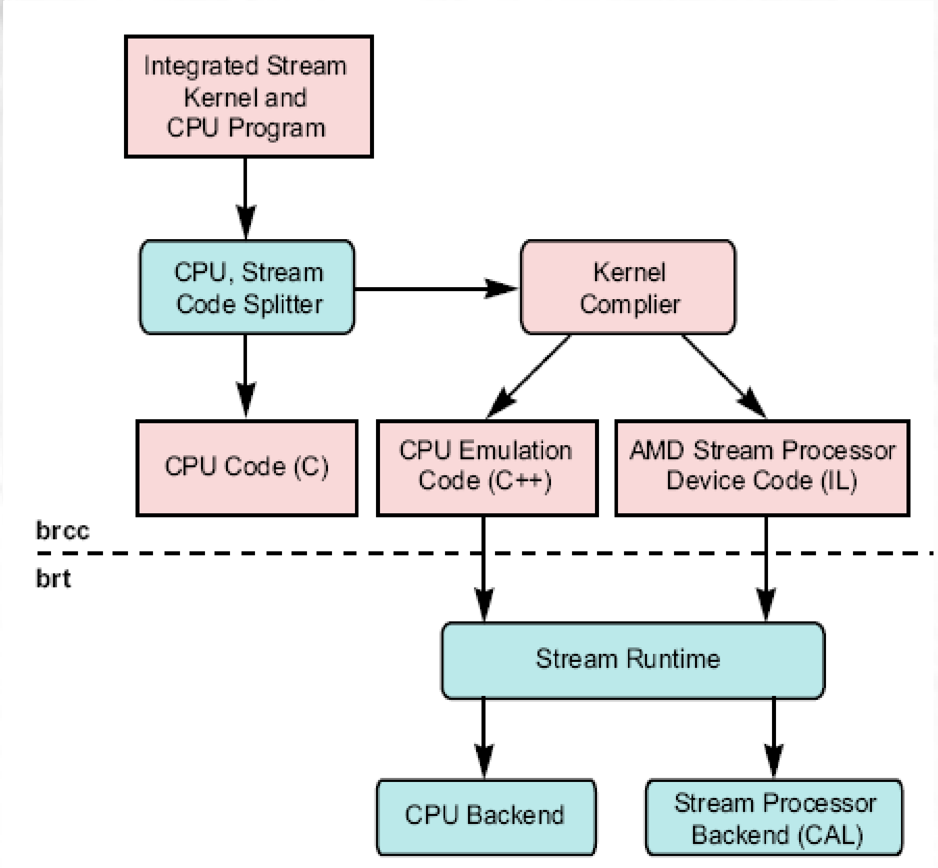
High Level Language

- AMD Brook+
  - Kernel execution
    - brook_kernel(a);
  - Kernel args
    - double i_arg<>
    - out double o_arg<>
  - Within kernel
    - floatN indexof(o_arg)
    - NO SHARED MEM
    - NO SYNCHRONIZATION

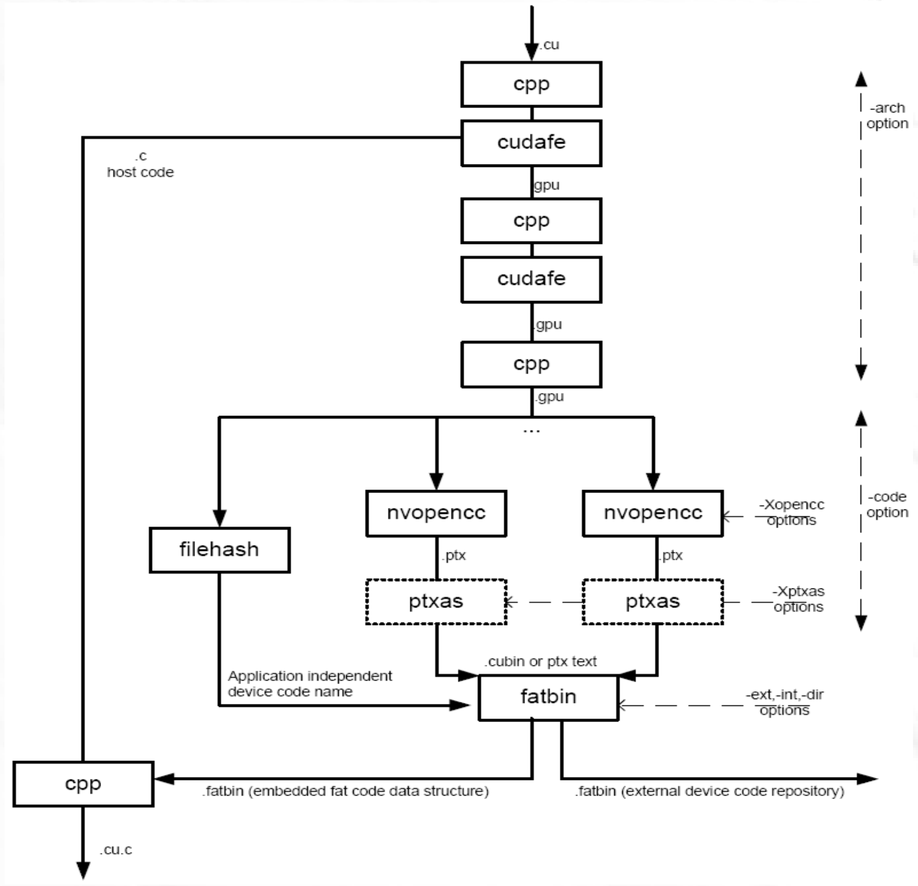
- NVIDA CUDA
  - Kernel execution
    - Dim3 grid(1,0,0),block(20,30,0);
    - cuda_kernel<<<grid,block>>>(a);
  - Kernel args
    - double* arg;
  - Within kernel
    - uint3 threadIdx,blockIdx
    - dim3 blockDim,gridDim
    - int warpSize
    - __shared__ double smem[20];
    - __syncthreads();

HLL compilation

- AMD Brook+


Integrated Stream Kernel and CPU Program
CPU, Stream Code Splitter
Kernel Complier
CPU Code (C)
CPU Emulation Code (C++)
AMD Stream Processor Device Code (IL)
brcc
brt
Stream Runtime
CPU Backend
Stream Processor Backend (CAL)

- NVIDIA CUDA



Graphic API

- OpenGL / DX
  - API - Functions with suffix “d” (GL)
  - Shaders
    - No scatter writing
    - Support within shaders
      - cg,HLSL : double,double2,double3,double4
      - GLSL : double,dvec2,dvec3,dvec4
    - Effective gather reading from textures
    - No FP64 texture support
    - Use of other buffers with FP64 support
      - Vertexbuffer,...
      - No (effective) gather reading

References

- AMD Stream computing user guide
- AMD Entering the golden age of Heterogeneous Computing
- Nvidia CUDA programming guide 2.0
- Nvidia Compute PTX:Parallel Thread Execution ISA Ver. 1.2
- The CUDA compiler driver
- Nvidia Geforce GTX 200 GPU architecture overview

Thank you