GPGPU, 1st Meeting

Mordechai Butrashvily, CEO
GASS

Agenda

- Forming a GPGPU WG
- 1st meeting
- Future meetings
- Activities

Forming a GPGPU WG

- To raise needs and enhance information sharing
- A platform for knowledge exchange and cooperation
- Periodical meetings with updates and news

1st meeting

- To align participants with the technology
- Exposure to latest developments and improvements
- Not too deep – that's saved for future meetings
- Meet the parties working in the Israeli industry

Future meetings

- Software stacks and frameworks by NVIDIA and ATI:
  - CUDA
  - StreamComputing
- Developments and general talks about programming and hardware issues
- More advanced topics
- Looking for ideas ☺

Activities

- Basis for a platform to exchange knowledge, ideas and information
- Cooperation and collaborations between parties in the Israeli industry
- Representing parties against commercial and international companies
- Training, courses and meetings with leading companies

Introduction to GPGPU

Mordechai Butrashvily, CEO
GASS – Driving the GPGPU Revolution
moti@gass-ltd.co.il

Agenda

- Technology and basic terms
- What is GPGPU used for?
- Existing technologies and trends
- Pros, cons of GPGPU and potential
- Community needs and knowledge gaps
- GASS Company
- Summary & Questions

Technology and basic terms

- What is GPGPU?
- GPU hardware explained
- How can computations fit into a GPU?

What is GPGPU?

- GPGPU stands for:

  General **P**urpose computation on **GPU**

  (also referred to as GPU computing)
- A GPU is a Graphics Processing Unit, the same hardware used for games and rendering applications

Continued…

- On 1 hand, an effort to use another computing platform besides the CPU
- Lately, an effort to create a supercomputer in a desktop PC size

Continued…

- How is it different from general graphics operations?
  - In graphics an image is used as input, and the output is an image as well
  - GPGPU – running various kinds of algorithms on a GPU, not necessarily image processing
  - For example: FFT, Monte-Carlo, Data-Sorting, Data mining and the list continues

Continued…

- What technological improvements allowed this?
  - Back in 2001, NVIDIA introduced a programmable GPU
  - In the last 4 years - a GPU can have input and output as floating point data
  - Major hardware improvements in speed and memory access rates

GPU hardware explained

- General characteristics
- Capabilities
- Input and output
- Data flow explained
- Programmability
- Data formats

General characteristics

- Two GPUs for example:

	NVIDIA 9800GTX	ATI HD3870
Cores#	128	320
Processing Power	~0.5 TFlops	~0.5 TFlops
Memory	512 MB GDDR3	512 MB GDDR4
Memory bandwidth	70.4 GB/s	76.8 GB/s
FP/Integer capability	IEEE 754 / 32 bit	IEEE 754 / 32 bit
Bus Interface	PCI Express 2.0	PCI Express 2.0
Power Consumption	168W	105W

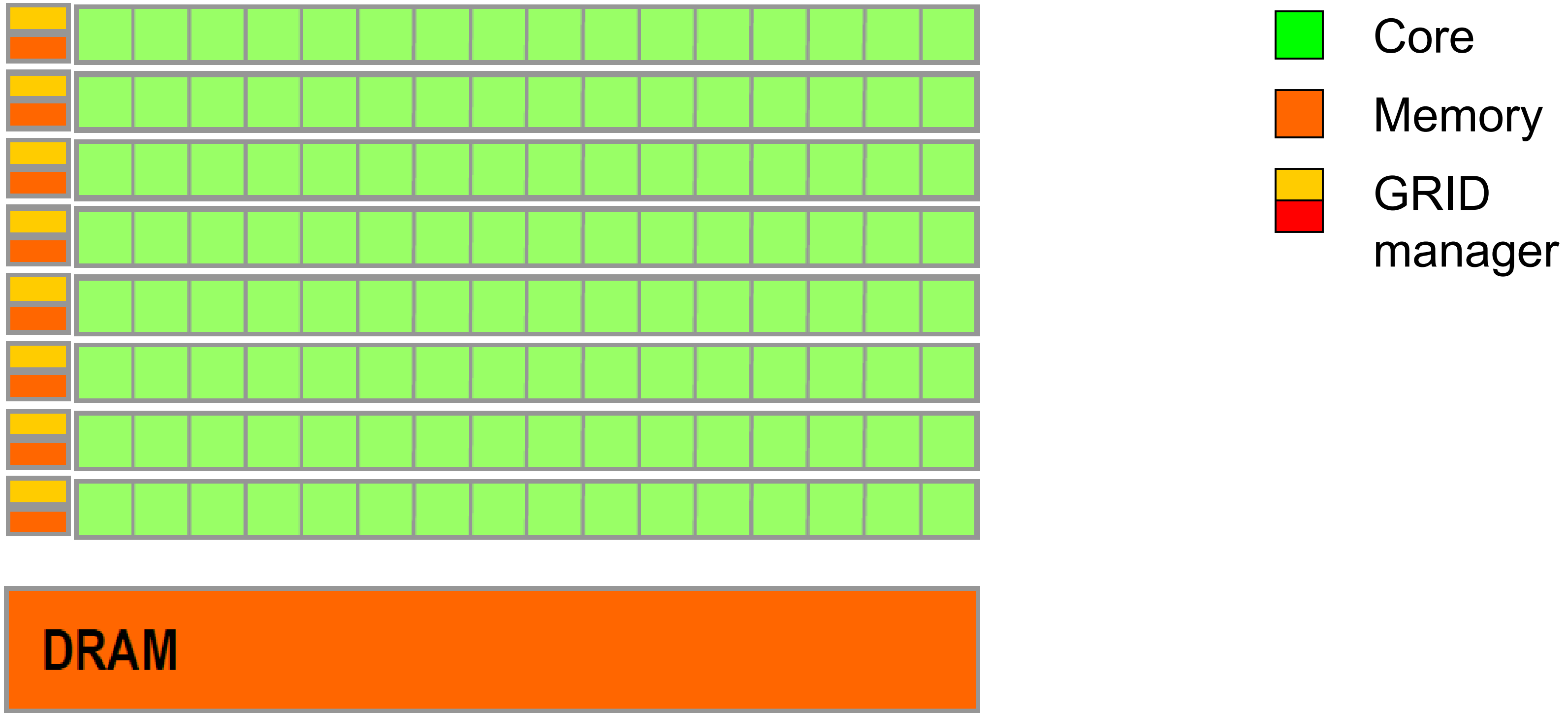
Continued…

- Two High-End GPUs for example:

	NVIDIA 9800GX2	ATI HD3870 X2
Cores#	256	640
Processing Power	~1 TFlops	>1 TFlops
Memory	1024 MB GDDR3	1024 MB GDDR4
Memory bandwidth	128 GB/s	115.2 GB/s
FP/Integer capability	IEEE 754 / 32 bit	IEEE 754 / 32 bit
Bus Interface	PCI Express 2.0	PCI Express 2.0
Power Consumption	197W	225W

Continued…

- A GPU:

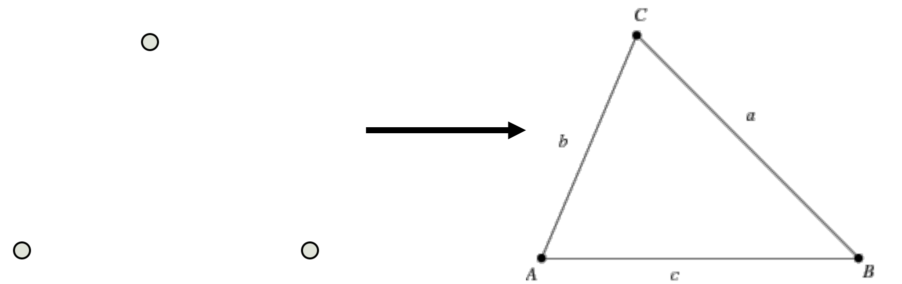


Capabilities

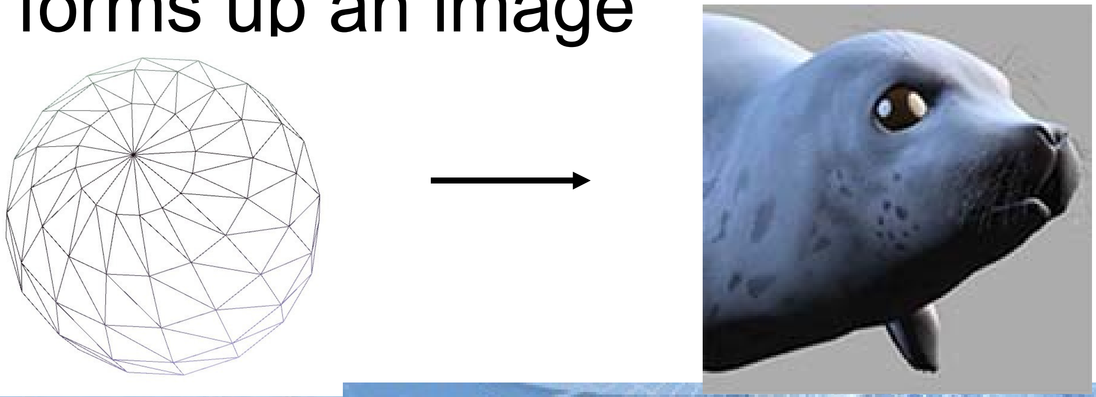
- Basic arithmetic: Integer and floating point
- Vector and matrix operations (dot, mat mul, etc.)
- Providing SIMD architecture – (alternative to SSE), over vector operations
- Decision and loop constructs: If, For etc.

Input and Output

- For general graphics:
  - Vertices (points in 3D space) form up a basic geometry



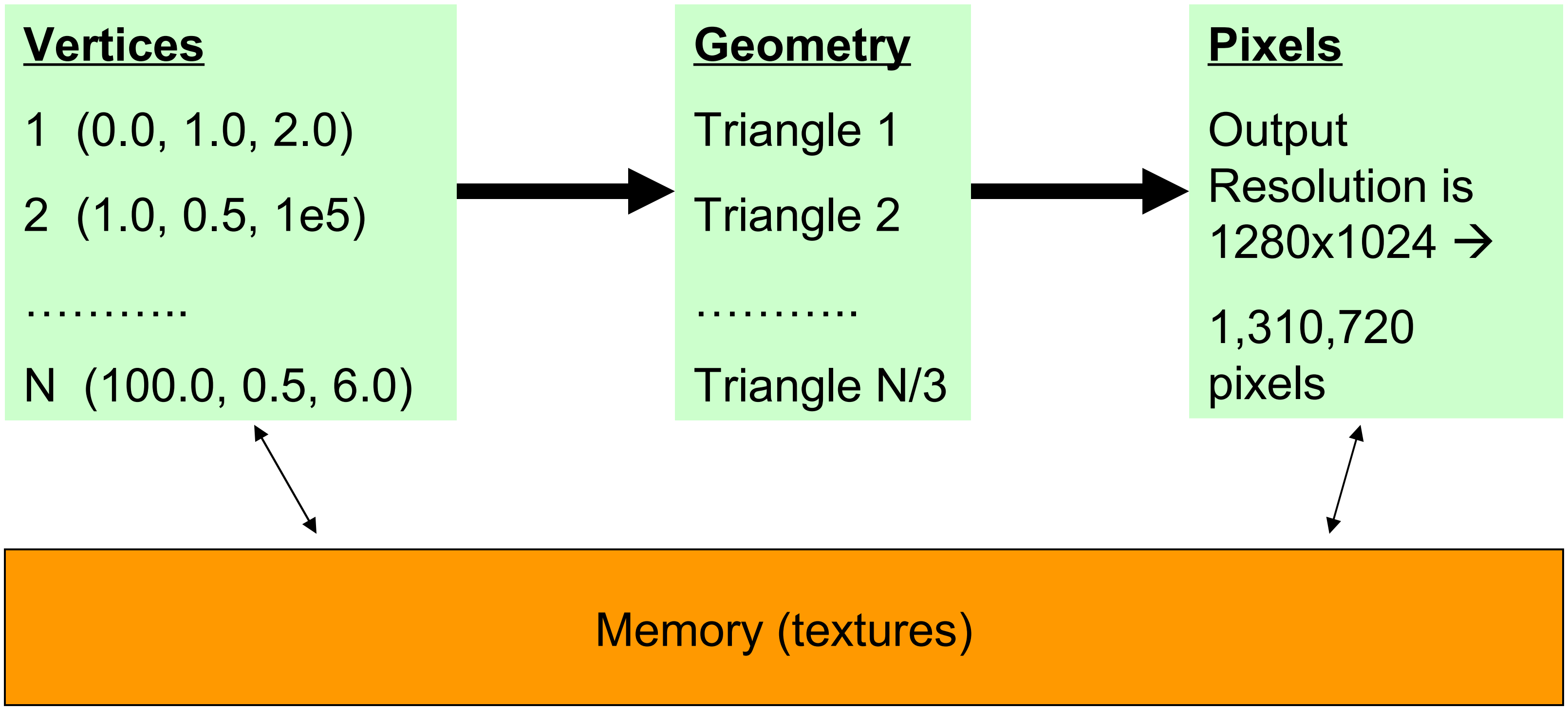
  - That finally forms up an image



Continued…

- A vertex can be represented by 4 elements: x, y, z, w (32 bit floats)
- A collections of vertices form up a geometry (triangle, quad or advanced polygon)
- The GPU will use all that data to generate pixels that create the final image
- A pixel can be 128 bit wide (4 * floats)

Data flow explained



Programmability

- NVIDIA introduced shaders in 2001 for programming a GPU
- There are some type of shaders:
  - Vertex
  - Geometry
  - Pixel
- Each operates on the corresponding element

Continued…

- For example:
  - If a GPU is given 1000 vertices to form a geometry, a vertex shader will fire for each one of them
  - Lets say, that the final image will be 1280x1024, thus 1,310,720 instances of the pixel shader will run
  - That's why a GPU has so many cores

Continued…

- An example for DirectX (HLSL) pixel shader:

```
float4 some_math(float4 data: COLOR): COLOR {
        data.x = sin(data.y) * data.z + data.z;
        return data;
}
```

- Very simple to use, no pointers – designed for math

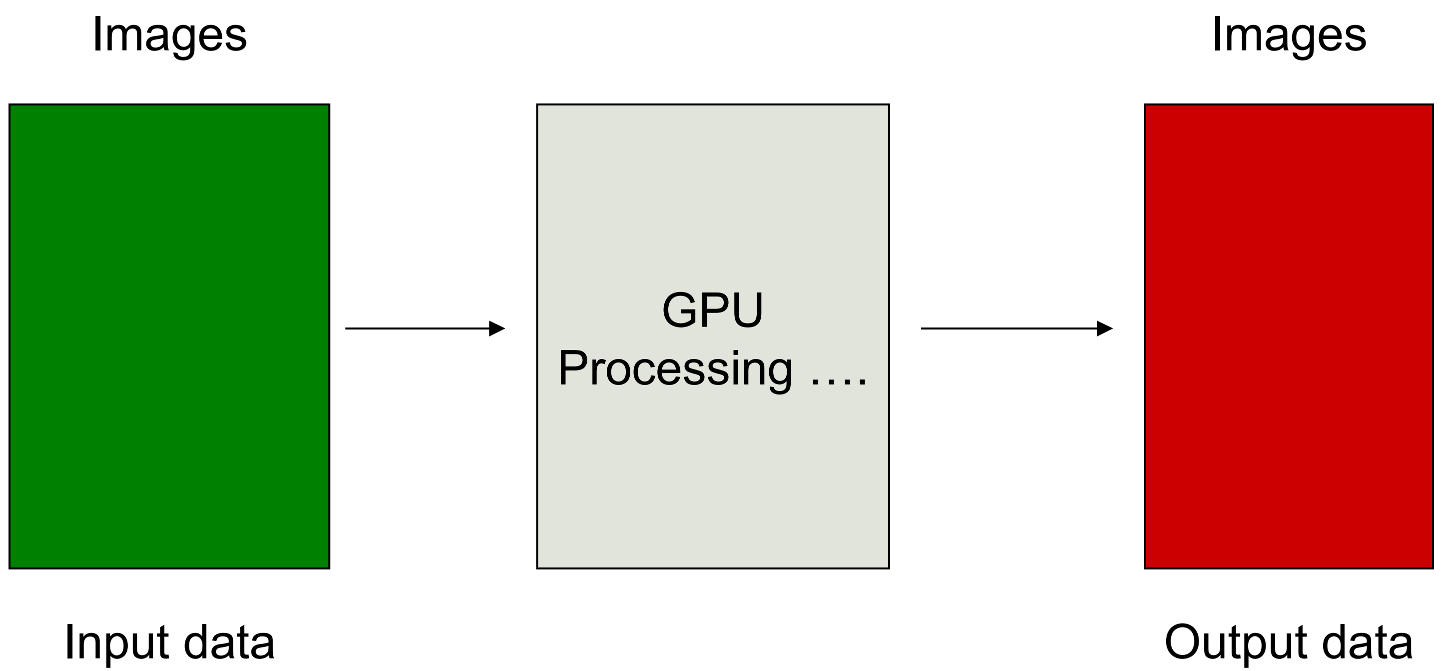
Continued…

- Capabilities:
  - Vector and Matrix operations (up to 4 dimensions)
  - Full IEEE754 support for floats
  - All basic arithmetic constructs and operations are supported (sin, tan, atan, exp etc.)
  - More than 65000 operations for a single shader
  - >300 internal registers to hold constant values

Data Formats

- GPU can support data in various formats:
  - Byte (8 bit)
  - Short (16 bit)
  - Integer (32 bit)
  - Half FP (16 bit)
  - FP (32 bit)
- Whereas a single element sent to the GPU can be composed of up to 4 of the formats above (only the same format is supported)
- The same is for output – 128 bit comes in, 128 bit comes out

An abstract diagram



How computations fit GPUs

- A massive parallel hardware
- Very clean use with great arithmetic support
- Almost every scientific algorithm can fit– specifically parallelized

A massive parallel hardware

- 640 independent cores in high-end cards
- Almost 80 memory reads per clock cycle for every core (128 bit fetches)

Clean use

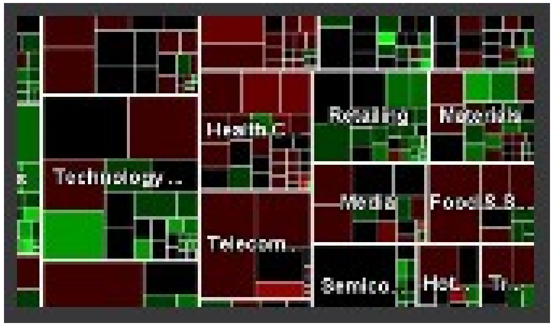
- Besides the graphics API (DirectX, OpenGL) shaders provide a straight forward approach for programming a GPU
- Every problem can be modeled to a GPU
- Not every problem will benefit the GPU

What is GPGPU used for?

- Wide use by international academies and research institutes
- The number is growing every day
- Science and industry fields:
  - Finance
  - Oil & Gas research, meteorology and seismic
  - Medicine and Biophysics
  - Numerical analysis
  - Audio and Signal processing

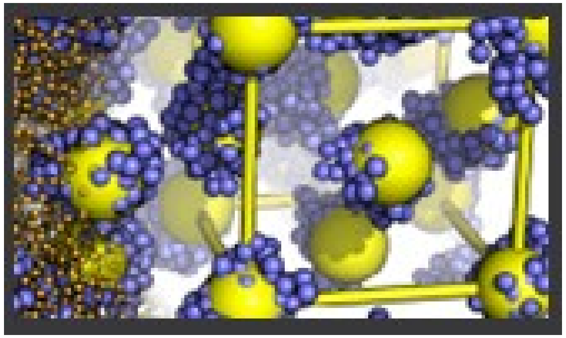
1st Case – Monte Carlo

- Used for option pricing and risk management in finance
- Speedup: 80 times
- Reason: ability of the GPU to process much more samples per second



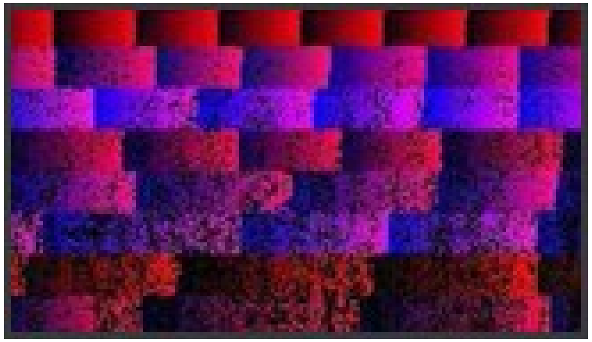
2^{nd} Case – Molecular dynamics

- Calculating forces and interaction between molecules
- Speedup: 30 times
- Reason: ability of the GPU to process large data and many elements



3rd Case – String Matching

- Matching DNA sequences or protein strings
- Speedup: 35 times
- Reason: ability of the GPU to process many sequence start points per second



Academic and Research use

- GPU computing enters many academies in the world
- Universities teaching CUDA:

http://www.nvidia.com/object/cuda_universities.html

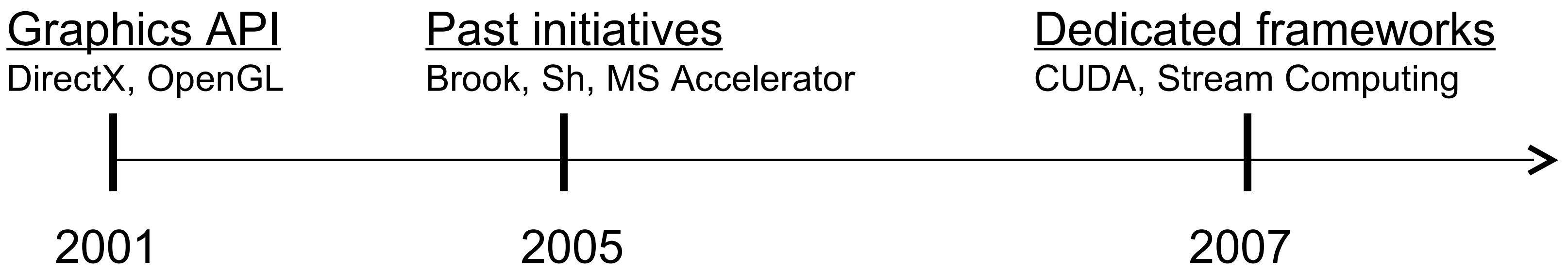
- Or FireStream by ATI
- Non are Israeli – Why?

Existing technologies & trends

- Programmability evolution with time
- GPU hardware evolution
- Latest GPU hardware, for GPGPU
- Upcoming products
- Why select GPU computing

Programmable evolution with time

- Graphics API
- Past initiatives
- Present & Future: CUDA and StreamComputing



Graphics API

- Choosing between DirectX or OpenGL
- The only option for GPGPU in the past
- Support for more data formats, enabling GPGPU (floating point)
- Shader languages improved – from 16 to more than 65000 operations per shader today
- GPU industry is driven by DirectX

Past initiatives

- Some efforts to allow general computing without using graphics API:
  - Brook
  - Sh
  - Microsoft Accelerator
- Writing a program in C, C++ or .NET
- But, in the backstage they all generated shader code in compilation or run time

Present & Future

- But all that was in the past
- No need to use graphics API for GPU computing
- NVIDIA & ATI provide frameworks for GPU computing
- Each framework targets the vendor's hardware

CUDA

- By NVIDIA
- Stands for - Compute Unified Device Architecture
- Provides additional libraries for FFT and BLAS routines
- Programming in C or C++ supporting integration with existing applications

StreamComputing

- By ATI
- Based on Brook and primarily used for processing stream of data
- ATI is porting ACML to GPUs
- Programming in C or C++ supporting integration with existing applications

Continues…

- Using graphics APIs allows to switch vendors without a problem
- Using a specific GPU computing framework bounds to the same vendor
- Future: serving as co-processor, without special compilers or frameworks

GPU hardware evolution

- From dedicated shader cores to universal
- Speed and bandwidth improvements
- Moore’s law?

Examination of ATI X1900

- Manufactured in 2006
- 90nm, 384 million transistors on die
- 48 pixel shader cores
- 8 vertex shader cores

From dedicated to universal shaders

- DirectX 10 (2007) introduced a new approach:
  - No longer specific cores, but universal
  - Each core is capable of running any sort of shader (vertex, geometry, pixel)
  - The GPU can utilize its cores based on the required work – for load balancing

Examination of ATI HD3870

- Manufactured in 2007
- 55nm, 666 million transistors on die
- 320 universal shader cores

Speed and bandwidth

	NVIDIA 7900	NVIDIA GTX 280
Date	2006	Q4 2008
Fabrication	90nm	65nm
Cores# **	20	240
Clock	450 Mhz	602 Mhz
Performance	250 GFlops	~1 TFlops
Bandwidth	42.24 GB/s	141.7 GB/s
Power	85 Watt	236 Watt

Performance metrics have quadrupled in 2 years!

Moore's law

- Claims that GPU hardware are surpassing Moore's law are still correct
- Technology is always improving, thanks to several factors:
  - Competition between ATI and NVIDIA
  - Microsoft (DirectX)
  - Gaming demands
- The quest for more power

Current GPU hardware

- General discussion on GPU hardware
- Both vendors have hardware that competes on the same slot
- GPU hardware is divided into categories:
  - Mobile / Notebook
  - Gaming
  - GPU Computing
  - Industry

Continued…

- Notebook GPUs are weak for GPU computing, but can serve well for specific tasks
- Under industry category, falls ATI FireGL line and NVIDIA Quadro. All devices costs 2000$ and above for a single card
- So our main interest are gaming platforms and GPU computing

Continued…

- Gaming platforms:

NVIDIA	ATI
9800 GTX	HD3870
9800 GX2	HD3870 X2

- GPU Computing:

NVIDIA	ATI/AMD
Tesla C870	FireStream 9170

GPUs Compared:

9800 GTX	HD3870	9800 GX2	HD3870 X2	Tesla C870	FireStream 9170
128	320	256	640	128	320
432 GFlops	496 GFlops	768 GFlops	1056 GFlops	512 GFlops	500 GFlops
512 MB	512 MB	1024 MB	1024 MB	1536 MB	2048 MB
70.4 GB/s	72 GB/s	128 GB/s	115.2 GB/s	76.8 GB/s	>100 GB/s
Single precision	Single precision	Single precision	Single precision	Single precision	**Double precision**
168 watt	105 Watt	197 Watt	225 Watt	171 Watt	150 Watt
270 $	130 $	425 $	300 $	1200 $	~2000 $

Upcoming products

- All products are targeted at Q3-Q4 of 2008

GTX 280	HD4870	Tesla C1060	FireStream 9250
240	480	240	480?
933 GFlops	1008 GFlops	1000 GFlops	1200 GFlops
1024 MB	1024 MB	4096 MB	1024 MB
142 GB/s	125.5 GB/s	102 GB/s	>100 GB/s
		Double precision	Double precision
236 Watt	150 Watt	225 Watt	150 Watt
?	?	?	~1000 $

GPU computing platforms – Why?

- Longer warranty – 3 years and not 1
- Devices are manufactured to avoid hardware arithmetic bugs (in gaming platforms this compensation is OK)
- Better support and drivers
- Devices are dedicated for GPU computing and cannot be attached to a screen

Pros, Cons and potential

- Comparing GPU to CPU
- How is virtualization related?
- Comparing GPU to GRID/Cluster
- What is the potential of GPU computing

GPU vs CPU

	ATI HD3870	Intel X5842	Factor
Core #	320	4	
Performance	500 GFlops	80 GFlops	x62.5
Memory Bandwidth	72 GB/s	10.4 GB/s	x7
Power	105 Watt	150 Watt	x1.4
Price	130 $	**1370 $**	x10
Total			**x6125?**

Virtualization effect on CPUs

- Providing better virtualization in the price of performance
- Better context switches and support for virtualization technologies add more complexity to the CPU
- Transistor growth is already limited (by fabrication)

www.Grid.org.il

MAGNET

GPU vs GRID

	ATI HD3870 X2	Intel X5842 * 100	Factor
Core #	640	400	
Performance	1 TFlops	8 TFlops	x0.125
Memory Bandwidth	115.2 GB/s	21 Gb/s	x5.4
Power	225 Watt	*15000 Watt	x66
Price	300 $	**x,000,000 $**	x∞
Total			**x,148,000?**

Continued…

- We didn't mention the maintenance of a GRID:
  - OS and environment, applying updates
  - Overall power consumption of nodes
  - Cooling solutions

What is the potential?

- Performance will increase as of other demanding industries (games)
- Until reaching engineering boundaries
- Development frameworks are improving
- Acceptance of GPGPU is growing:
  - In environments with limited resources
  - Or that are looking for small sized solutions with better ROI

Community needs and gaps

- Familiarity with hardware and frameworks
- Training, basic and advanced
- Accessible information and knowledge centers

Familiarity with hardware

- What GPUs exist in the market?
- Which is the best for our problem?
- Bus interfaces and integration into existing platforms?

Familiarity with frameworks

- Select CUDA or StreamComputing?
- Decision regarding hardware affects framework selection and vice-versa
- DirectX might be still the answer
- Effective use of the framework
- Optimizations when required

Training, basic and advanced

- Introducing GPGPU to skeptics
- Understanding how it works
- “Hello, world!”
- Advanced programming and using complex constructs and capabilities

Accessible information

- Opening MSDN is easy and serves well
- What about GPU computing?
- Information isn’t always accessible, not speaking of advanced material

Summary

- GPU computing is very mature
- Very cost-effective solutions compared to CPU and GRID
- Both NVIDIA and ATI provide hardware and software solutions:
  - NVIDIA: Tesla, CUDA
  - ATI: FireStream, StreamComputing
- Still more to be done

Summary & Questions