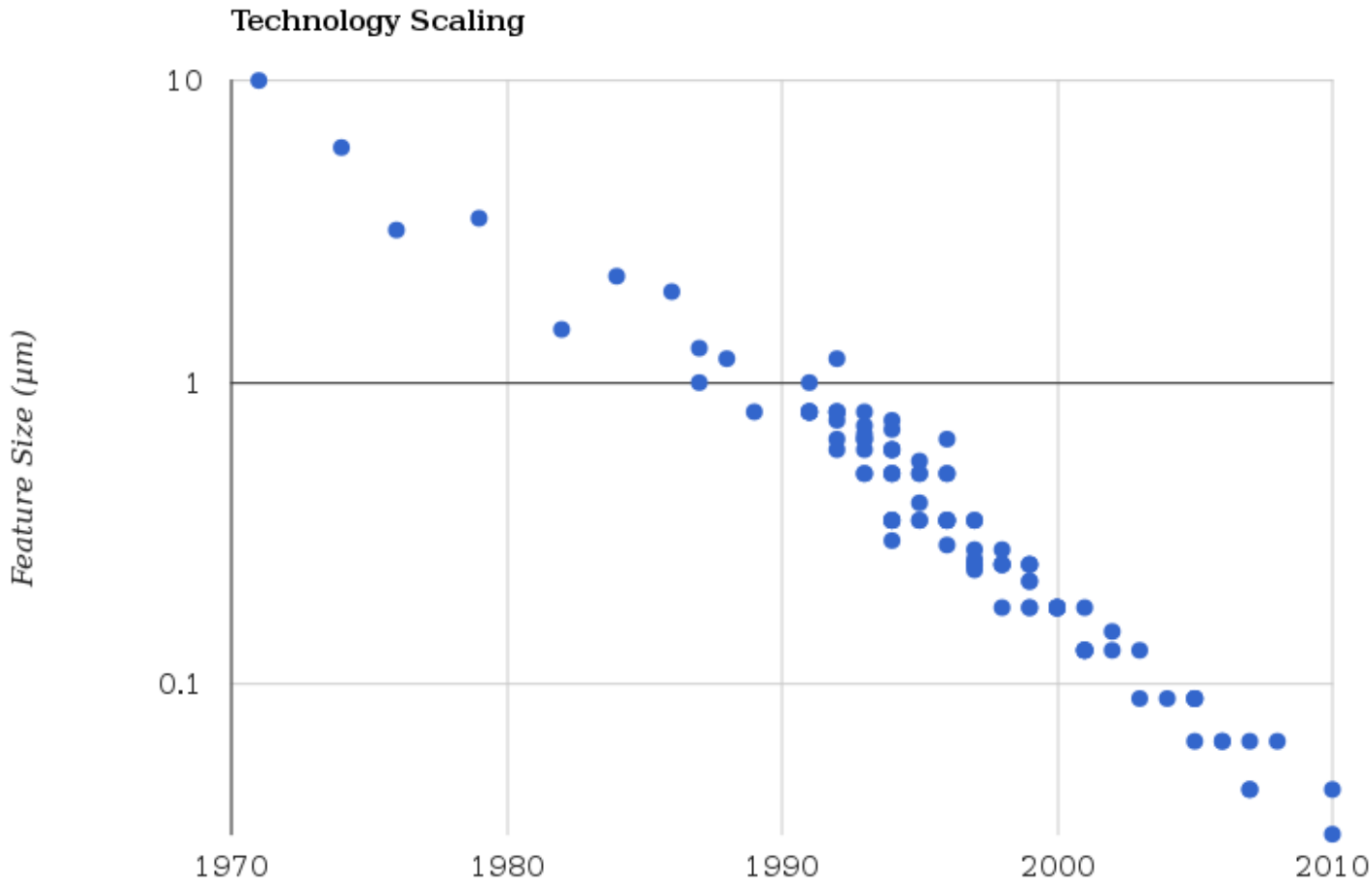
Wprowadzenie do programowania GPU

Marcin Copik

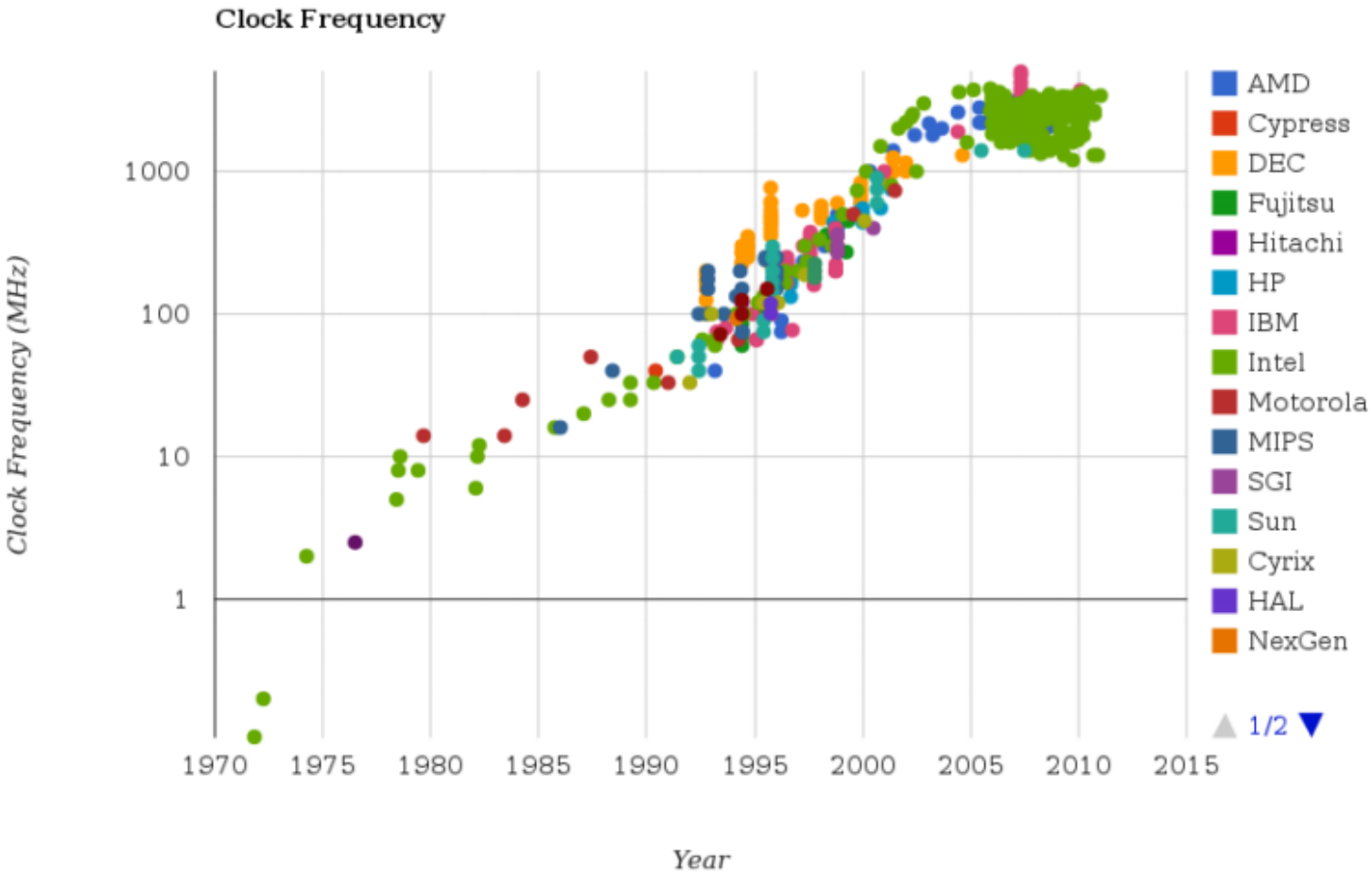
22 stycznia 2017

Spis treści

Zmniejszanie tranzystorów



Częstotliwość zegara



Superkomputery

TOP3 - 2010

Supercomputer	Max speed (TFLOPS)	Processors	Power (kW)
Tianhe-1A	2,566	14,336 Intel Xeon CPUs, 7,168 Nvidia Tesla GPUs	4040.00
Jaguar	1,759	224,256 AMD Opteron CPUs	6950.60
Nebulae	1,271	9,280 Intel Xeon CPUs, 4,640 Nvidia Tesla GPUs	2580.00

Źródło - 'OpenCL in Action',M.Scarpino

2009 - jeden na 20 najlepszych superkomputerów posiadał GPU

2012 - 62 z 500 najlepszych używają GPU,w tym najszybszy

Superkomputery

TOP3 - 2010

Supercomputer	Max speed (TFLOPS)	Processors	Power (kW)
Tianhe-1A	2,566	14,336 Intel Xeon CPUs, 7,168 Nvidia Tesla GPUs	4040.00
Jaguar	1,759	224,256 AMD Opteron CPUs	6950.60
Nebulae	1,271	9,280 Intel Xeon CPUs, 4,640 Nvidia Tesla GPUs	2580.00

Źródło - 'OpenCL in Action',M.Scarpino

2009 - jeden na 20 najlepszych superkomputerów posiadał GPU

2012 - 62 z 500 najlepszych używają GPU,w tym najszybszy

Superkomputery

TOP3 - 2010

Supercomputer	Max speed (TFLOPS)	Processors	Power (kW)
Tianhe-1A	2,566	14,336 Intel Xeon CPUs, 7,168 Nvidia Tesla GPUs	4040.00
Jaguar	1,759	224,256 AMD Opteron CPUs	6950.60
Nebulae	1,271	9,280 Intel Xeon CPUs, 4,640 Nvidia Tesla GPUs	2580.00

Źródło - 'OpenCL in Action',M.Scarpino

2009 - jeden na 20 najlepszych superkomputerów posiadał GPU

2012 - 62 z 500 najlepszych używają GPU,w tym najszybszy

SIMT

SIMD - Single Instruction, Multiple Data
SIMT - Single Instruction, Multiple Thread

Dlaczego nowe pojęcie?

- marketing Nvidii?
- różnica względem np. SSE

SIMT

SIMD - Single Instruction, Multiple Data
SIMT - Single Instruction, Multiple Thread
Dlaczego nowe pojęcie?

- marketing Nvidii?
- różnica względem np. SSE

SIMT

SIMD - Single Instruction, Multiple Data
SIMT - Single Instruction, Multiple Thread
Dlaczego nowe pojęcie?

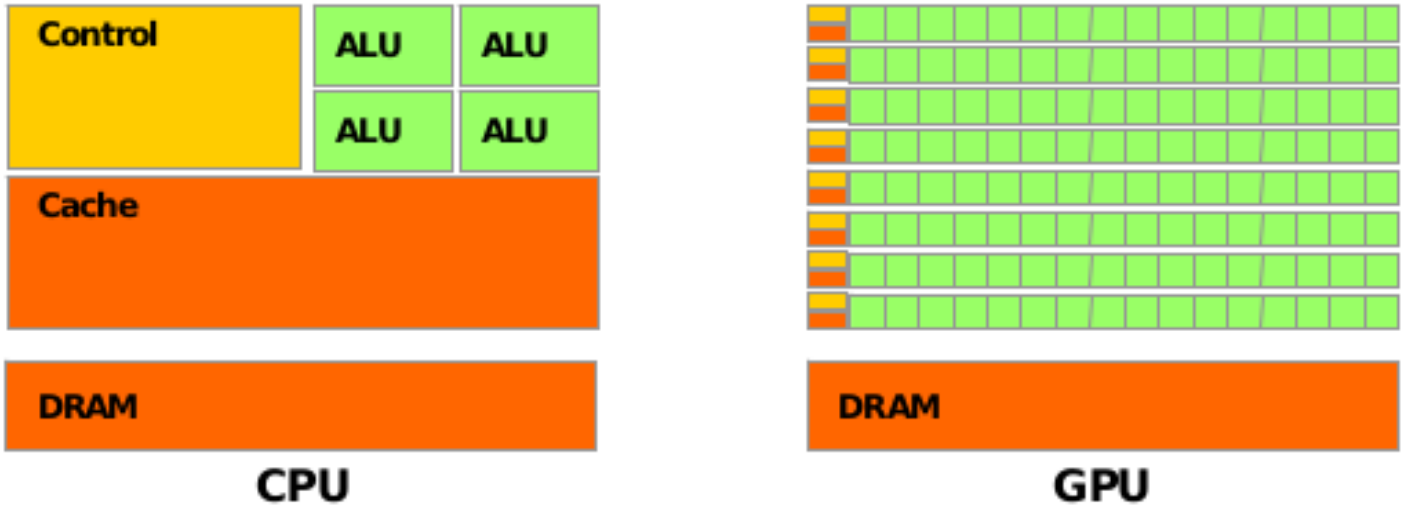
- marketing Nvidii?
- różnica względem np. SSE

SIMT

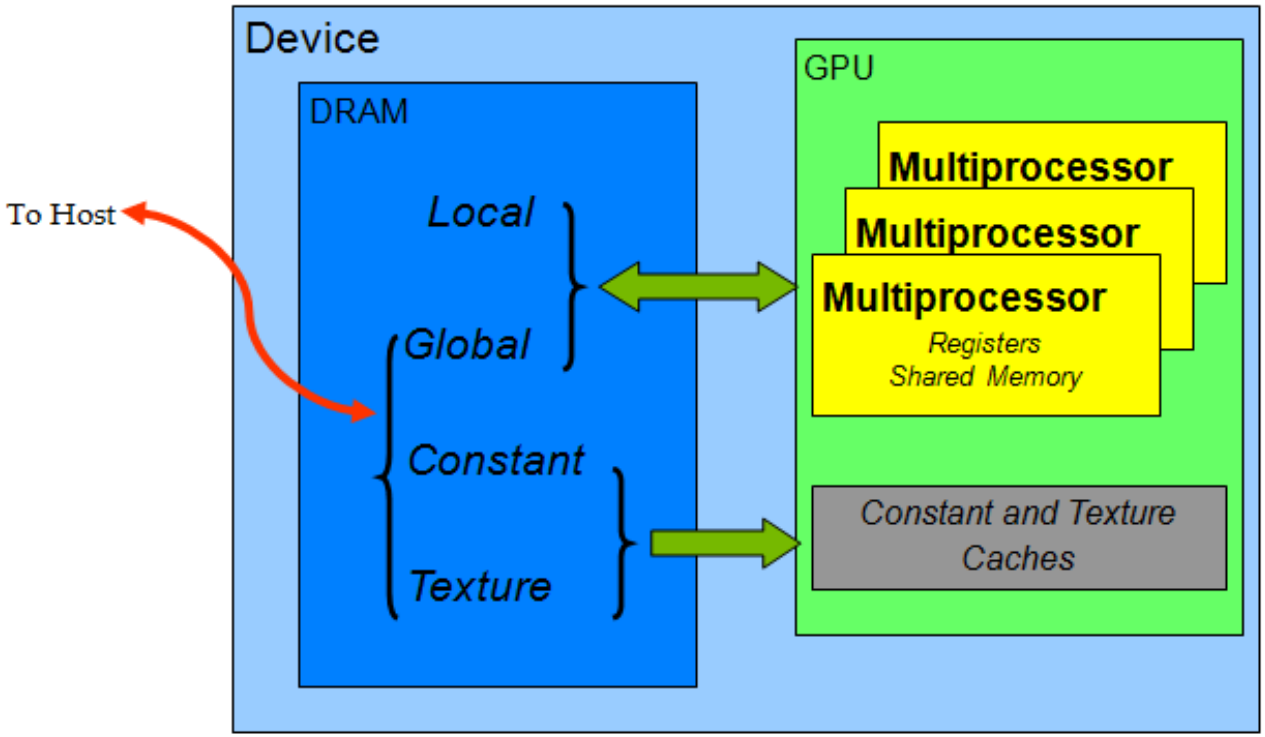
SIMD - Single Instruction, Multiple Data
SIMT - Single Instruction, Multiple Thread
Dlaczego nowe pojęcie?

- marketing Nvidii?
- różnica względem np. SSE

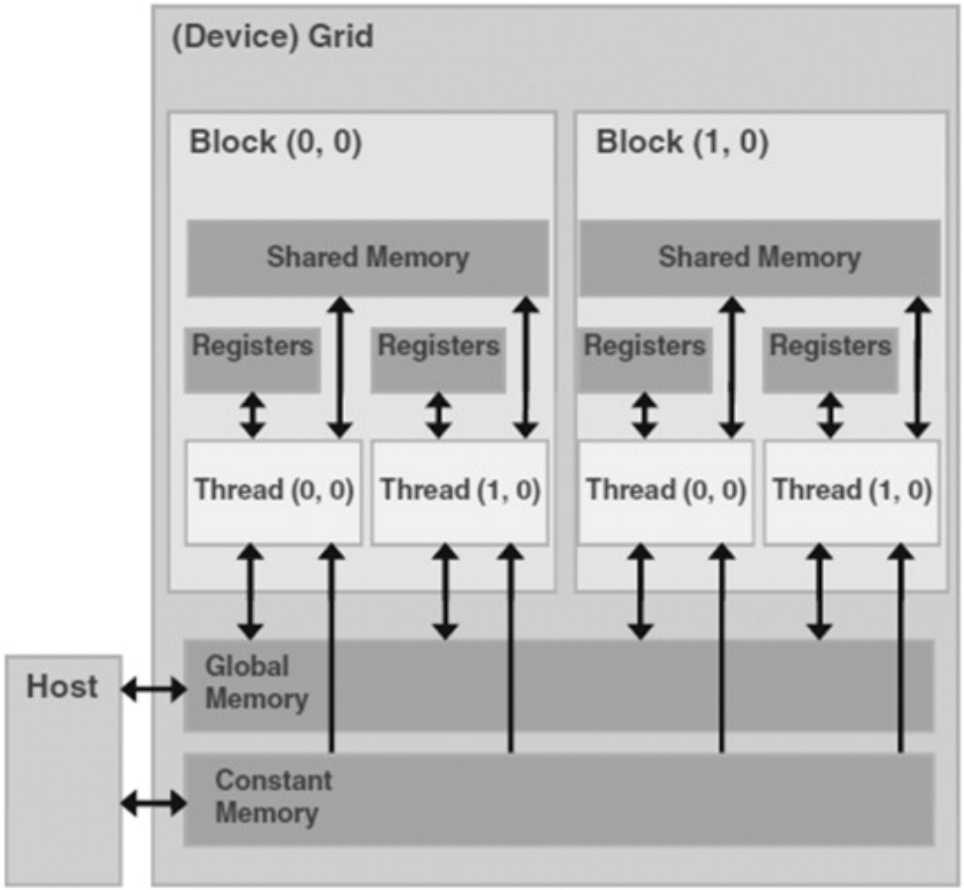
Control-intensive vs data-intensive



Architektura urządzenia



Bloki na urządzeniu



Pamięć

Rodzaje pamięci:

- **rejestry(registers)** - 32-bitowe, dla wątku, 'on-chip',R/W
- **lokalna(local)** - dla wątku, 'off-chip',R/W
- **współdzielona(shared)** - dla bloku, 'on-chip',R/W
- **globalna(global)** - dla urządzenia, 'off-chip',R/W
- **stała(constant)** - dla urządzenia, 'off-chip',R

Pamięć

Rodzaje pamięci:

- **rejestry(registers)** - 32-bitowe, dla wątku, 'on-chip',R/W
- **lokalna(local)** - dla wątku, 'off-chip',R/W
- **współdzielona(shared)** - dla bloku, 'on-chip',R/W
- **globalna(global)** - dla urządzenia, 'off-chip',R/W
- **stała(constant)** - dla urządzenia, 'off-chip',R

Pamięć

Rodzaje pamięci:

- **rejestry(registers)** - 32-bitowe, dla wątku, 'on-chip',R/W
- **lokalna(local)** - dla wątku, 'off-chip',R/W
- **współdzielona(shared)** - dla bloku, 'on-chip',R/W
- **globalna(global)** - dla urządzenia, 'off-chip',R/W
- **stała(constant)** - dla urządzenia, 'off-chip',R

Pamięć

Rodzaje pamięci:

- **rejestry(registers)** - 32-bitowe, dla wątku, 'on-chip',R/W
- **lokalna(local)** - dla wątku, 'off-chip',R/W
- **współdzielona(shared)** - dla bloku, 'on-chip',R/W
- **globalna(global)** - dla urządzenia, 'off-chip',R/W
- **stała(constant)** - dla urządzenia, 'off-chip',R

Pamięć

Rodzaje pamięci:

- **rejestry(registers)** - 32-bitowe, dla wątku, 'on-chip',R/W
- **lokalna(local)** - dla wątku, 'off-chip',R/W
- **współdzielona(shared)** - dla bloku, 'on-chip',R/W
- **globalna(global)** - dla urządzenia, 'off-chip',R/W
- **stała(constant)** - dla urządzenia, 'off-chip',R

Pamięć

Rodzaje pamięci:

- **rejestry(registers)** - 32-bitowe, dla wątku, 'on-chip',R/W
- **lokalna(local)** - dla wątku, 'off-chip',R/W
- **współdzielona(shared)** - dla bloku, 'on-chip',R/W
- **globalna(global)** - dla urządzenia, 'off-chip',R/W
- **stała(constant)** - dla urządzenia, 'off-chip',R

Prawo Amdahla

$$S = \frac{1}{(1 - P) + \frac{P}{N}}$$

Opóźnienia

- pamięć globalna jest wolna!
- pamięć współdzielona ma ok. 100 razy mniejsze opóźnienie
- 'coalescing' dla pamięci globalnej
- 'bank conflicts' dla pamięci współdzielonej
- liczba rejestrów nie jest nieskończona!

Opóźnienia

- pamięć globalna jest wolna!
- pamięć współdzielona ma ok. 100 razy mniejsze opóźnienie
- 'coalescing' dla pamięci globalnej
- 'bank conflicts' dla pamięci współdzielonej
- liczba rejestrów nie jest nieskończona!

Opóźnienia

- pamięć globalna jest wolna!
- pamięć współdzielona ma ok. 100 razy mniejsze opóźnienie
- 'coalescing' dla pamięci globalnej
- 'bank conflicts' dla pamięci współdzielonej
- liczba rejestrów nie jest nieskończona!

Opóźnienia

- pamięć globalna jest wolna!
- pamięć współdzielona ma ok. 100 razy mniejsze opóźnienie
- 'coalescing' dla pamięci globalnej
- 'bank conflicts' dla pamięci współdzielonej
- liczba rejestrów nie jest nieskończona!

Opóźnienia

- pamięć globalna jest wolna!
- pamięć współdzielona ma ok. 100 razy mniejsze opóźnienie
- 'coalescing' dla pamięci globalnej
- 'bank conflicts' dla pamięci wspóldzielonej
- liczba rejestrów nie jest nieskończona!

Rozgałęzienia

- jeden warp może wykonywać jedną instrukcję w danej chwili
- gdy następuje N rozgałęzień, to N-krotnie spada wydajność
- kompilator może dokonać optymalizacji
- rozdzielanie danych na warpy i kernele

Rozgałęzienia

- jeden warp może wykonywać jedną instrukcję w danej chwili
- gdy następuje N rozgałęzień, to N-krotnie spada wydajność
- kompilator może dokonać optymalizacji
- rozdzielanie danych na warpy i kernele

Rozgałęzienia

- jeden warp może wykonywać jedną instrukcję w danej chwili
- gdy następuje N rozgałęzień, to N-krotnie spada wydajność
- kompilator może dokonać optymalizacji
- rozdzielanie danych na warpy i kernele

Rozgałęzienia

- jeden warp może wykonywać jedną instrukcję w danej chwili
- gdy następuje N rozgałęzień, to N-krotnie spada wydajność
- kompilator może dokonać optymalizacji
- rozdzielanie danych na warpy i kernele

Technicznie

- kod kernela w pliku .cu
- kompilator nvcc
- łączymy z aplikacją w C lub C++
- wrappery do Pythona, Javy, Matlaba....

Technicznie

- kod kernela w pliku .cu
- kompilator nvcc
- łączymy z aplikacją w C lub C++
- wrappery do Pythona, Javy, Matlaba....

Technicznie

- kod kernela w pliku .cu
- kompilator nvcc
- łączymy z aplikacją w C lub C++
- wrappery do Pythona, Javy, Matlaba....

Technicznie

- kod kernela w pliku .cu
- kompilator nvcc
- łączymy z aplikacją w C lub C++
- wrappery do Pythona, Javy, Matlaba....

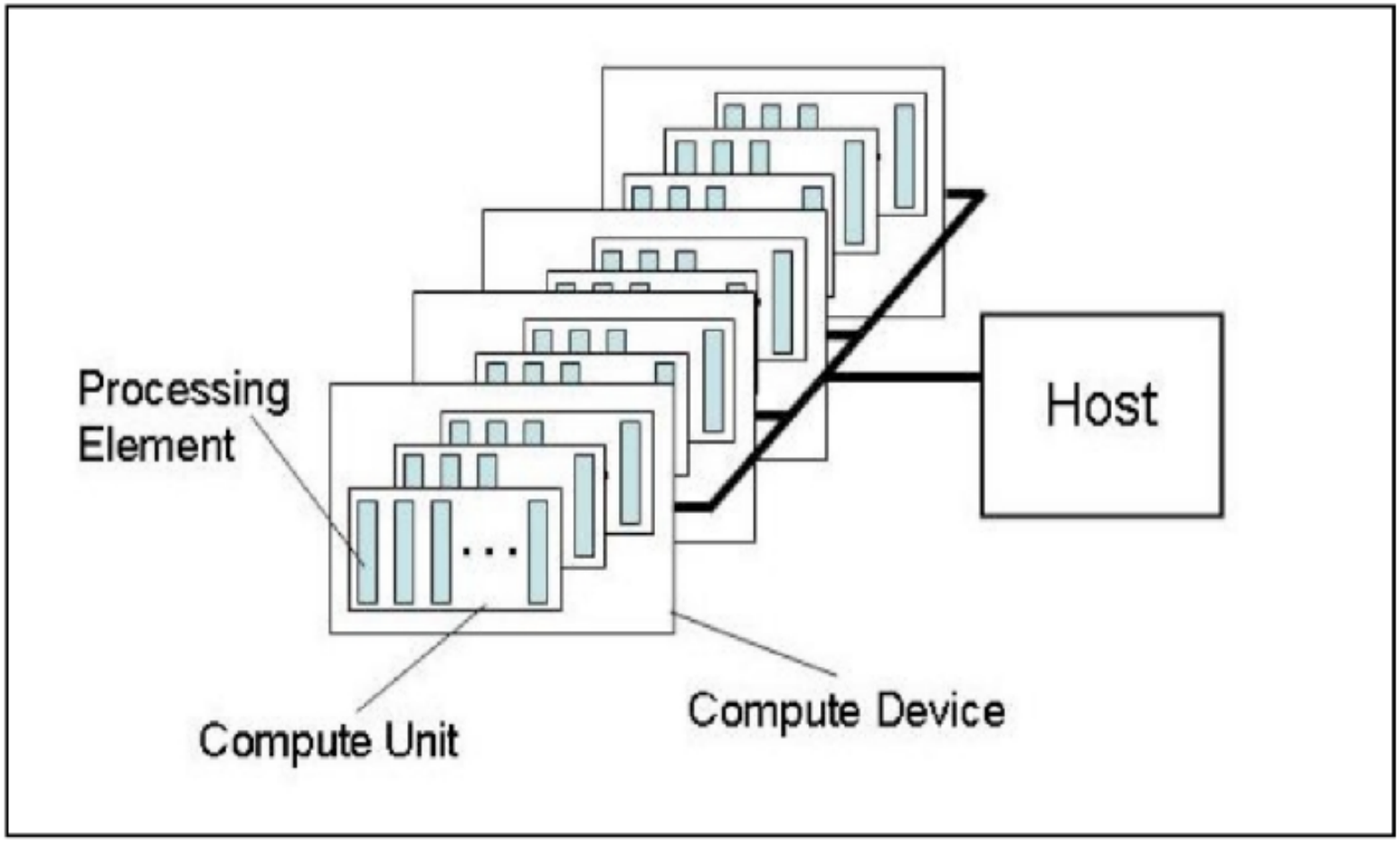
OpenCL


KHRONOS GROUP
OpenCL
3DLABS
ACTIVISION
BLIZZARD
AMD
ARM
BROADCOM
EA
codeplay
ERICSSON
freescale
IBM
intel
Imagination
MOTOROLA
movidia
NOKIA
NVIDIA
Petapath
QNX
RAPIDMIND
S3 GRAPHICS
SAMSUNG
Seaweed SYSTEMS
TAKUMI
TEXAS INSTRUMENTS

Różnice

CUDA term	OpenCL term
GPU	Device
Multiprocessor	Compute Unit
Scalar core	Processing element
Global memory	Global memory
Shared (per-block) memory	Local memory
Local memory (automatic, or local)	Private memory
kernel	program
block	work-group
thread	work item

Model urządzeń



Oficjalne materiały

- 'CUDA C Programming Guide'
- 'CUDA C Best Practices Guide'
- 'OpenCL Programming Guide for the CUDA Architecture'
- 'OpenCL Best Practices Guide'

Oficjalne materiały

- 'CUDA C Programming Guide'
- 'CUDA C Best Practices Guide'
- 'OpenCL Programming Guide for the CUDA Architecture'
- 'OpenCL Best Practices Guide'

Oficjalne materiały

- 'CUDA C Programming Guide'
- 'CUDA C Best Practices Guide'
- 'OpenCL Programming Guide for the CUDA Architecture'
- 'OpenCL Best Practices Guide'

Oficjalne materiały

- 'CUDA C Programming Guide'
- 'CUDA C Best Practices Guide'
- 'OpenCL Programming Guide for the CUDA Architecture'
- 'OpenCL Best Practices Guide'

E-learning

- 'Coursera: Heterogeneous Parallel Programming, University of Illinois'
- 'Udacity: Introduction to Parallel Programming, University of California + NVIDIA'

E-learning

- 'Coursera: Heterogeneous Parallel Programming, University of Illinois'
- 'Udacity: Introduction to Parallel Programming, University of California + NVIDIA'

Książki

- 'CUDA by Example: An Introduction to General-Purpose GPU Programming'
- 'Heterogeneous Computing with OpenCL '
- 'CUDA Application Design and Development'

Książki

- 'CUDA by Example: An Introduction to General-Purpose GPU Programming'
- 'Heterogeneous Computing with OpenCL '
- 'CUDA Application Design and Development'

Książki

- 'CUDA by Example: An Introduction to General-Purpose GPU Programming'
- 'Heterogeneous Computing with OpenCL '
- 'CUDA Application Design and Development'