Ministerstwo Nauki
i Szkolnictwa Wyższego

Regionalna Inicjatywa Doskonałości w Dyscyplinach
Informatyki, Elektrotechniki, Elektroniki, Automatyki i Robotyki
na Politechnice Częstochowskiej

Projekt w ramach programu Regionalna Inicjatywa Doskonałości, decyzja nr 020/RID/2018/19

Architektura Systemów Komputerowych

Rozwój współczesnych systemów komputerowych i podsumowanie materiału

dr hab. inż. Łukasz Szustak, prof. PCz
lszustak@icis.pcz.pl

Katedra Informatyki
Wydział Inżynierii Mechanicznej i Informatyki
Politechnika Częstochowska

Ministerstwo Nauki
i Szkolnictwa Wyższego

Regionalna Inicjatywa Doskonałości w Dyscyplinach
Informatyki, Elektrotechniki, Elektroniki, Automatyki i Robotyki
na Politechnice Częstochowskiej

Projekt w ramach programu Regionalna Inicjatywa Doskonałości, decyzja nr 020/RID/2018/19

Zamieszczane materiały służą wyłącznie do celów indywidualnego kształcenia. Nie wyrażam zgody na ich utrwalanie przekazywanie osobom trzecim ani rozpowszechnianie.

dr hab. inż. Łukasz Szustak, prof. PCz
lszustak@icis.pcz.pl

Katedra Informatyki
Wydział Inżynierii Mechanicznej i Informatyki
Politechnika Częstochowska

Wydział Inżynierii Mechanicznej i Informatyki
The Faculty of Mechanical Engineering and Computer Science

Komputery wielkiej skali: SUPERKOMPUTERY

- **Fugaku** - najszybszy superkomputer na świecie wg rankingu TOP500 z czerwca 2020
  - **7 299 072** rdzeni obliczeniowych
  - **415 530 000 000 000 000** wykonywanych operacji na sekundę

Komputery wielkiej skali: koncepcja budowy

https://www.ibm.com

- Zazwyczaj, pojedynczy węzeł obliczeniowy:
  - Kilka procesorów CPU
  - **Kilka akceleratorów obliczeniowych**
  - Pamięć operacyjna
  - Inne
- Węzły połączone są ze sobą wysokowydajną siecią komputerową, tworząc superkomputer

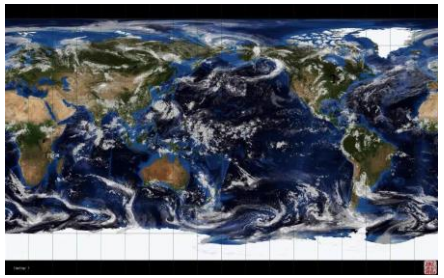
Komputery wielkiej skali: zużycie energii elektrycznej

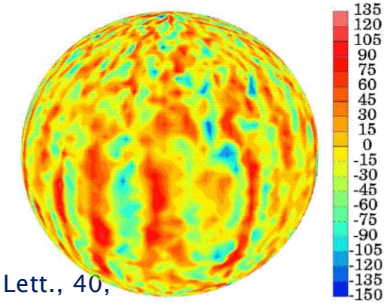
- Obecnie najszybszy superkomputer ***Fugaku*** wg rankingu TOP500 z czerwca 2020, wymaga dostarczenia energii elektrycznej na poziomie **28 MW**
- Zapotrzebowanie Politechniki Częstochowskiej na energię elektryczną wynosi ok **0,5 MW**
- Superkomputer ***Fugaku*** w ciągu 1 godziny pracy zużywa tyle samo energii elektrycznej co ok 14 gospodarstw domowych na przestrzeni roku

Komputery wielkiej skali: zastosowanie

- Nauki ścisłe, ekonomia, finanse, symulacja pogody, przemysł lotniczy i samochodowy i wiele innych

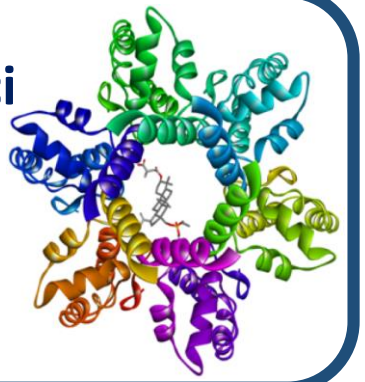
Prognoza pogody



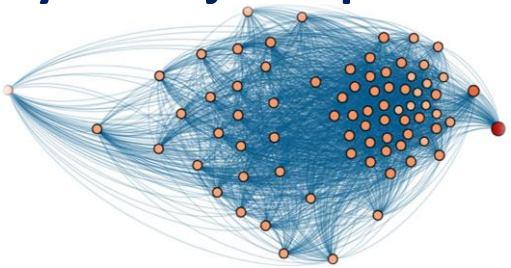


Miyamoto et al (2013) , Geophys. Res. Lett., 40, 4922–4926, doi:10.1002/grl.50944.

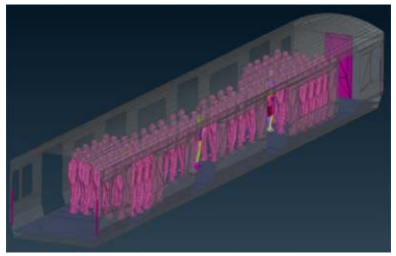
Symulacje właściwości nowych substancji o aktywności przeciwwirusowej



Symulacje rozprzestrzeniania się wirusa



https://www.anl.gov/



https://www.r-ccs.riken.jp/

Uczenie maszynowe

https://www.nvidia.com

Akceleratory obliczeniowe GPU

Co to GPU?

- GPU (Graphics Processing Unit) – specjalizowana jednostka obliczeniowa zaprojektowana do przetwarzania grafiki
- Termin GPU został wprowadzony przez firmę NVIDIA w 1999 roku podczas wprowadzania karty grafiki GeForce 256
- Rozwój grafiki komputerowej skutkuje zwiększonym zapotrzebowaniem na moc obliczeniową
- GPU zostały wprowadzone aby przejąć obliczenia związane z grafiką od CPU i zwiększyć wydajność całego systemu
- GPU może służyć nie tylko do grafiki. Nowoczesne GPU dzięki dużej mocy obliczeniowej określane jako GPGPU (General Purpose GPU – GPU ogólnego przeznaczenia)

Grafika dedykowana

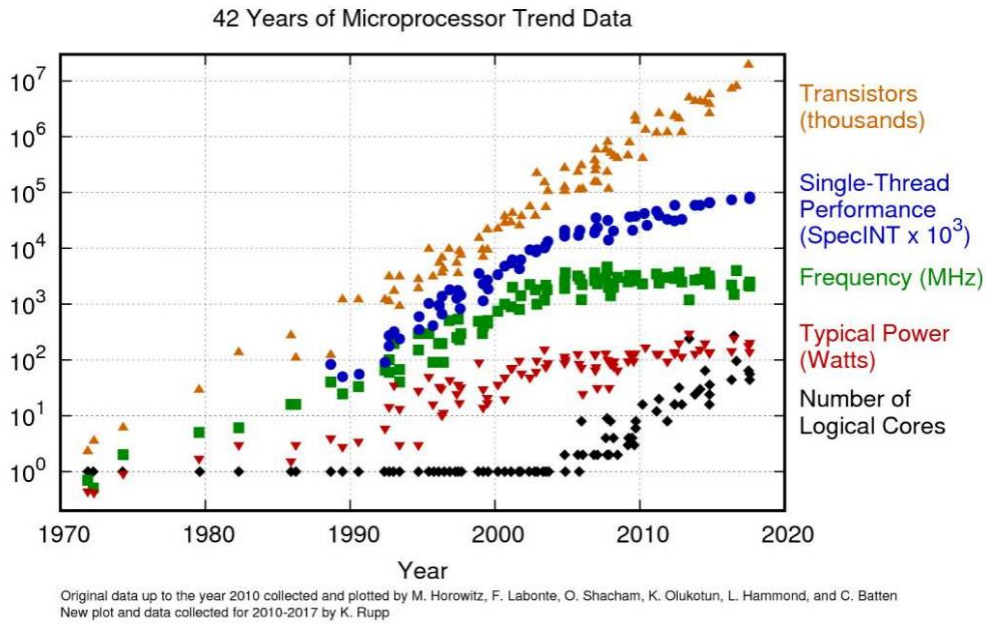
- Montowana na płycie głównej za pomocą złącza PCI –Express
- Duża wydajność
- Własna pamięć RAM
- Wysoka cena sprzętu z dedykowaną grafiką
- Większe zapotrzebowanie na energię

Grafika zintegrowana

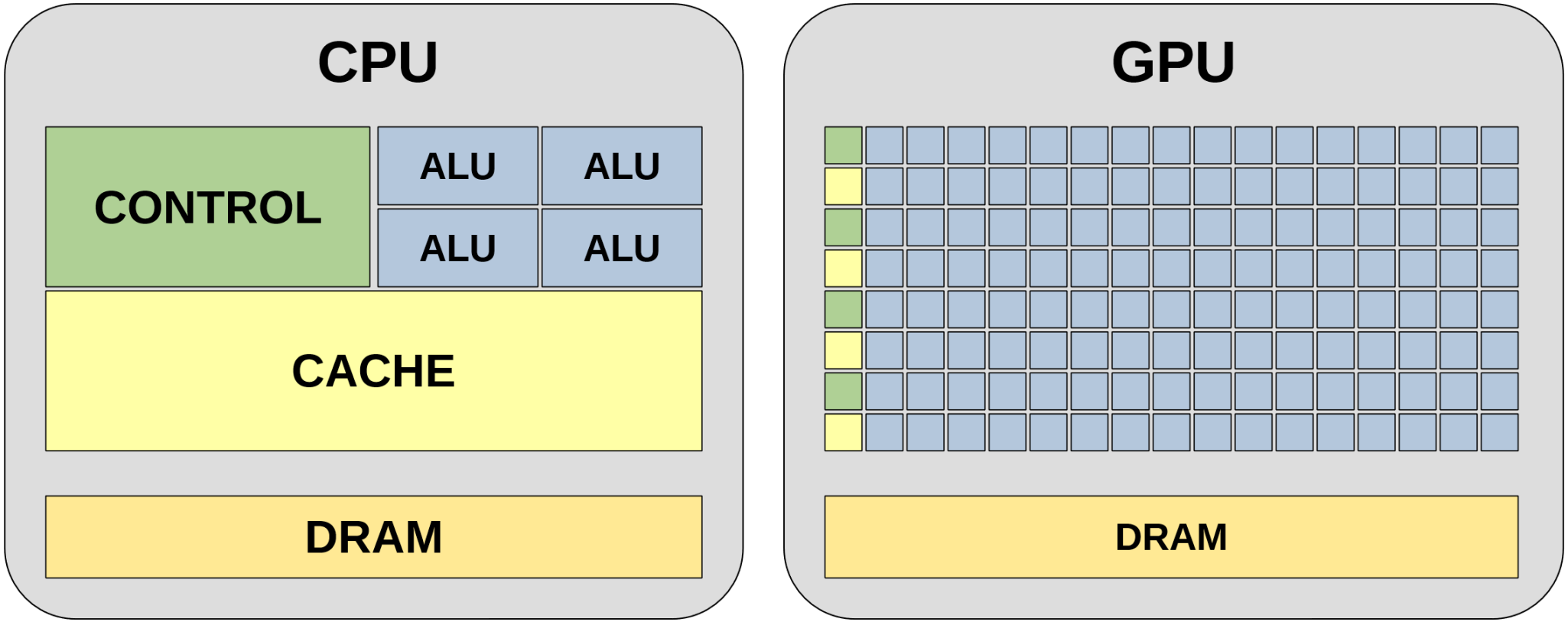
- Grafika zintegrowana oznacza, że układ przetwarzający
  obraz jest wbudowany w chipset
- Zdecydowanie mniejsza wydajność
- Współdzielenie pamięci RAM
- Niska cena sprzętu z grafiką zintegrowaną
- Mniejsze zapotrzebowanie na energię

Prawo Moore’a

- Gordon Moore – jeden z założycieli firmy Intel, w 1965 roku zaobserwował, że liczba tranzystorów w układach scalonych podwaja się co około dwa lata
- Trend ten wydaje się być dalej aktualny
- W obecnych czasach prawo Moore’a często przyjmuje formę: „moc obliczeniowa komputerów podwaja się co 24 miesiące”
- Na przestrzeni ostatnich lat można zaobserwować, że częstotliwość pojedynczego rdzenia CPU osiągnęła limit i jest niemalże niezmienna
- Wydajność pojedynczego wątku rośnie coraz wolniej
- Wydajność systemów obliczeniowych rośnie przede wszystkim z powodu zwiększania liczby rdzeni na procesor



Architektura CPU i GPU



CPU

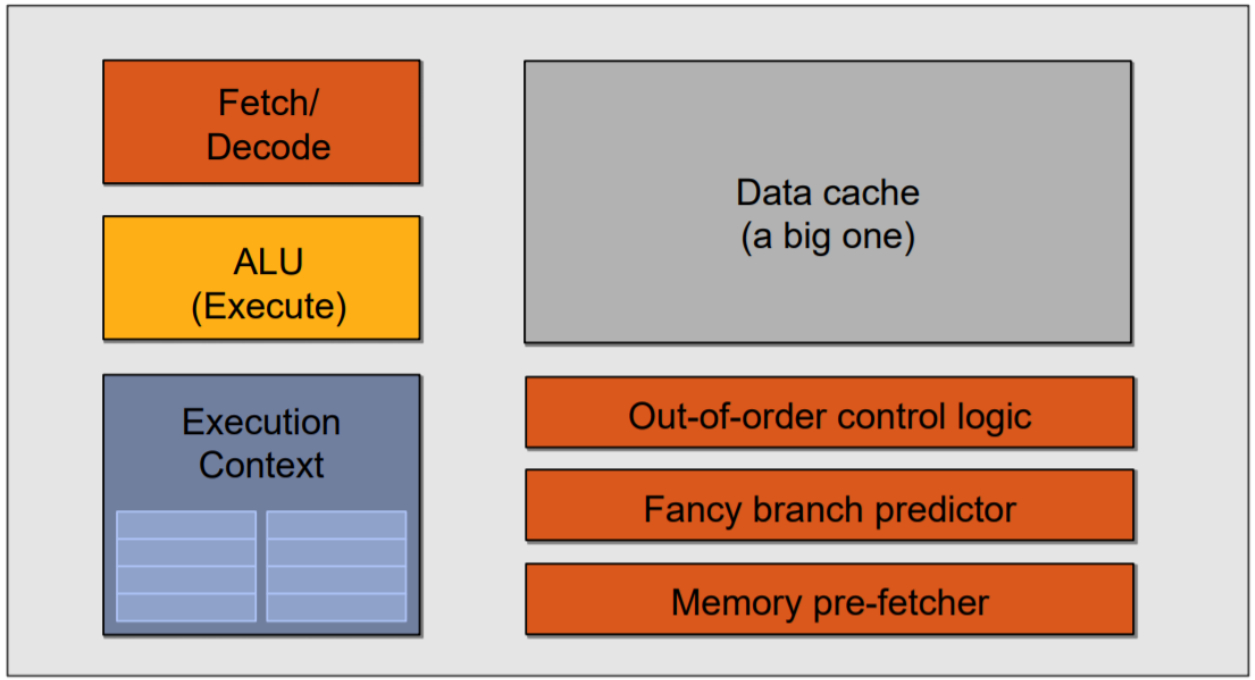
- Priorytetem jest jak **najszybsze** wykonanie strumienia instrukcji
- Niewielka liczba rdzeni
- Dzięki skomplikowanej budowie, rdzenie umożliwiają szybką pracę
- Wielopoziomowa pamięć podręczna
- Rozbudowana potokowość
- Możliwość szybkiego przełączania się między operacjami (Out of order execution)
- Pobieranie danych do pamięci podręcznej z wyprzedzeniem (Prefetching)
- Przewidywanie rezultatów instrukcji warunkowych (Branch prediciton)

GPU

- Priorytetem jest optymalizacja **przepustowości** pozwalająca na wykonanie jak największej liczby zadań jednocześnie
- Architektura składająca się z bardzo dużej liczby, prostszych w budowie rdzeni pozwalających na obliczenia równoległe
- Rdzenie składające się z wielu jednostek ALU (Arithmetic Logic Unit)
- Potrzeba zarządzania hierarchią pamięci
- Liczba poziomów pamięci podręcznej jest mniejsza i charakteryzuje się niewielkim rozmiarem
- Mniejszy rozmiar pamięci podręcznej skutkuje dłuższym czasem dostępu do pamięci
- Jest on jednak ukrywany poprzez równoległe wykonywanie

Szybkość vs Przepustowość

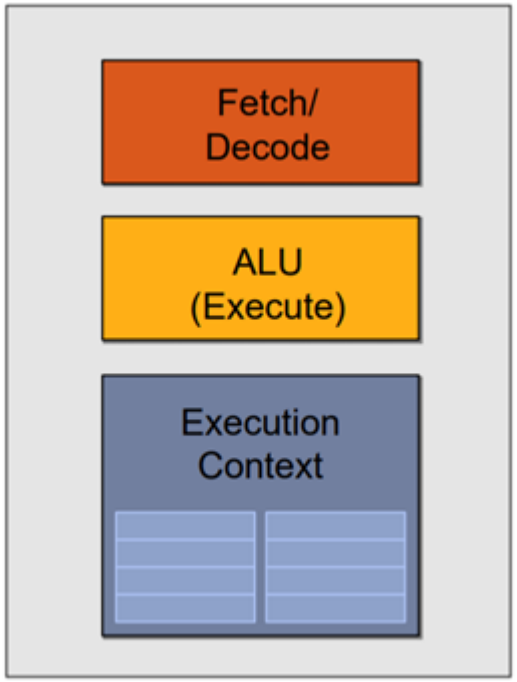
Od CPU do GPU



Źródło: Introduction to GPU Architecture, Ofer Rosenberg, AMD

Od CPU do GPU

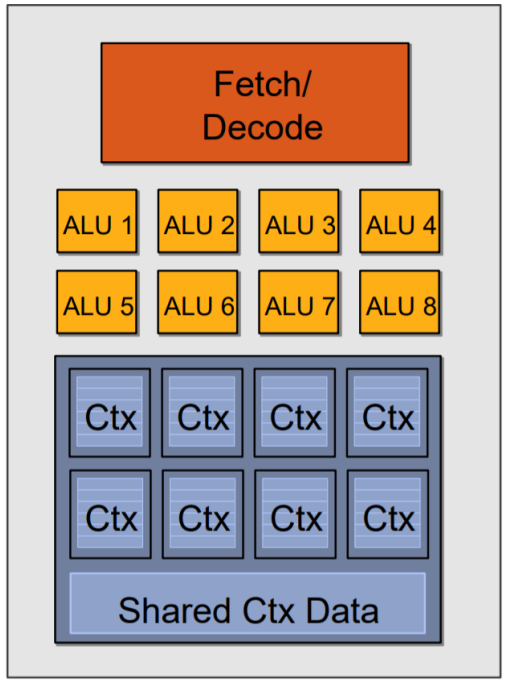
- Usunięcie układów, które wspomagają szybkie wykonanie potoku instrukcji
- Rozmiar rdzenia się zmniejsza, więc liczba rdzeni może być większa
- Koszt produkcji jest niższy z uwagi na uproszczoną budowę



Źródło: Introduction to GPU Architecture, Ofer Rosenberg, AMD

Od CPU do GPU

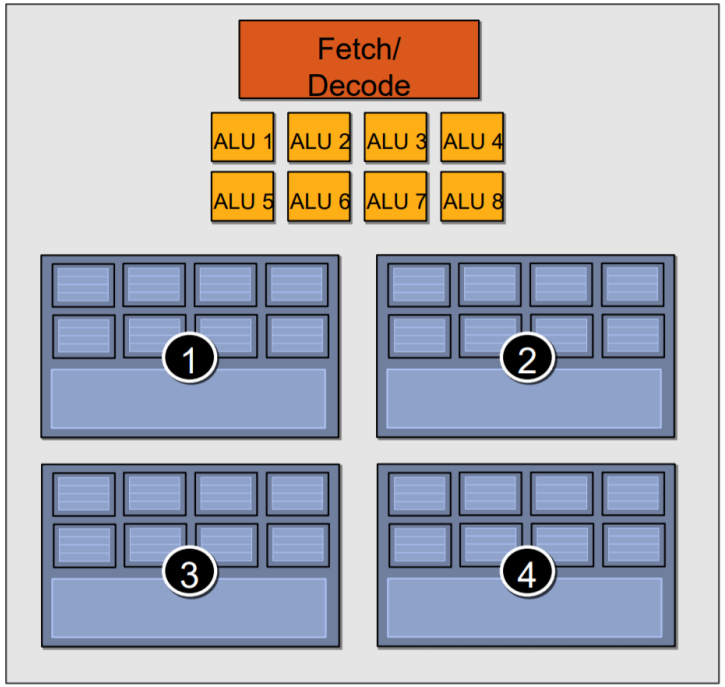
- Przetwarzanie SIMD (Single Instruction Multiple Data)
- Układ pobierania oraz dekodowania może pozostać niezmieniony dzięki wykonywania tych samych instrukcji na różnych danych
- Praca poszczególnych ALU jest wstrzymywana przez oczekiwanie na dane, co skutkuje niską wydajnością
- Układy zapobiegające takim sytuacjom zostały usunięte w poprzednim kroku



Źródło: Introduction to GPU Architecture, Ofer Rosenberg, AMD

Od CPU do GPU

- Rozwiązaniem jest podział kontekstu na grupy
- Przemienna praca poszczególnych grup pozwala na unikanie opóźnień
- Im więcej grup, tym większa możliwość unikania opóźnień
- Przełączanie między kontekstami może być zarządzanie z poziomu sprzętu i/lub oprogramowania

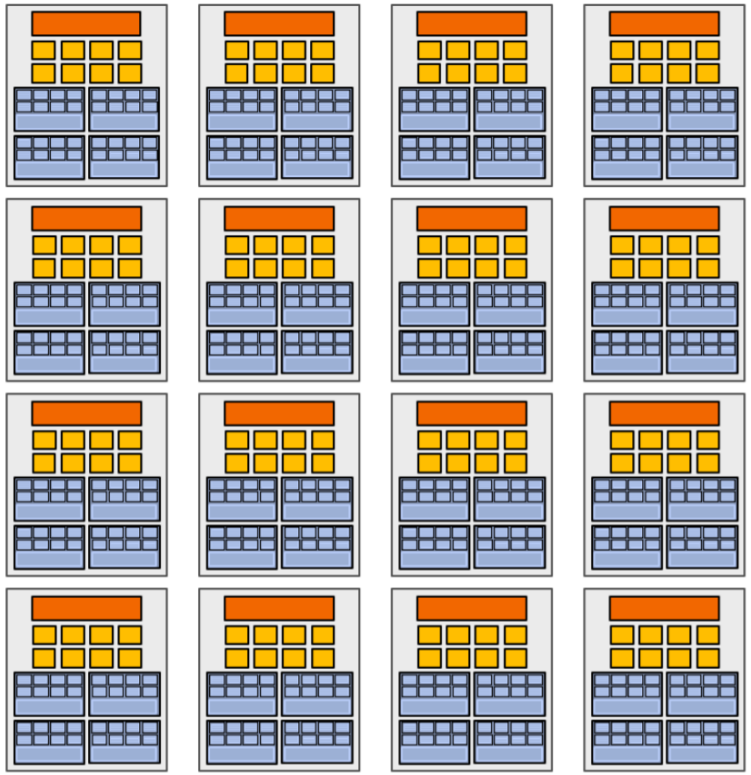


Źródło: Introduction to GPU Architecture, Ofer Rosenberg, AMD

Od CPU do GPU

Przykład

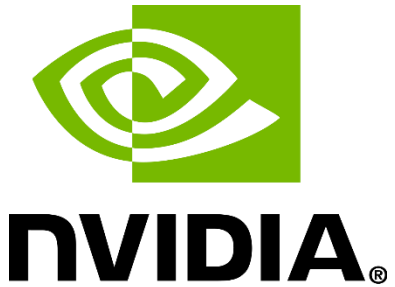
- 16 rdzeni
- 128 ALU (8 w jednym rdzeniu)
- 16 niezależnych strumieni instrukcji
- 64 współbieżne strumienie instrukcji
- 512 współbieżnych kontekstów
- Przy taktowaniu 1 GHz – 256 GFLOPs.

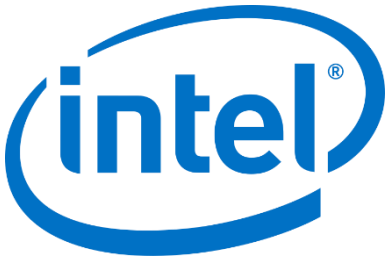


Źródło: Introduction to GPU Architecture, Ofer Rosenberg, AMD

GPU – udział w rynku

- Rynek dedykowanych kart graficznych jest zdominowany przez dwie firmy – Nvidia i AMD
- Liderem jest Nvidia – jej udział w rynku w 2019 roku wyniósł 73%, przy 27% AMD
- W grafice zintegrowanej najlepsze wyniki osiąga Intel – 63 %





NVIDIA – rodziny produktów

- **GeForce** – grafika konsumencka
- **Tegra** – SoC (system on chip) - urządzenia mobilne, systemy wbudowane
- **Tesla** – wysokiej klasy GPGPU – HPC (high performance computing)
- **Quadro** – profesjonalne wizualizacje komputerowe
- **NVS** – grafika biznesowa z wieloma wyświetlaczami
- **nForce** – chipset płyty głównej produkowany dla firmy Intel i AMD

NVIDIA – architektury GPU

- Rozwój technologii przekłada się na coraz wydajniejszy proces technologiczny określony szerokością (w nanometrach) bramki tranzystora

Architektura	Rok wydania	Proces technologiczny [nm]
Tesla	2006	40-90
Fermi	2010	28-40
Kepler	2012	28
Maxwell	2014	28
Pascal	2016	14-16
Volta	2017	12
Turing	2018	12
Ampere	2020	7-8
Hopper	?	5

NVIDIA GA100 GPU

- Ponad 54 miliardy tranzystorów
- Proces technologiczny 7nm

- Architektura podzielona na 8 GPC (GPU Processing Cluster)
- Każdy GPC składa się z 8 TPC (Texture Processing Cluster)
- Każdy TPC zawiera 2 SM (Streaming Multiprocessor)
- Łącznie do dyspozycji jest więc 128 multiprocesorów

- 64 rdzenie CUDA na jeden multiprocesor SM
- Łącznie 8192 rdzenie CUDA w całym GPU

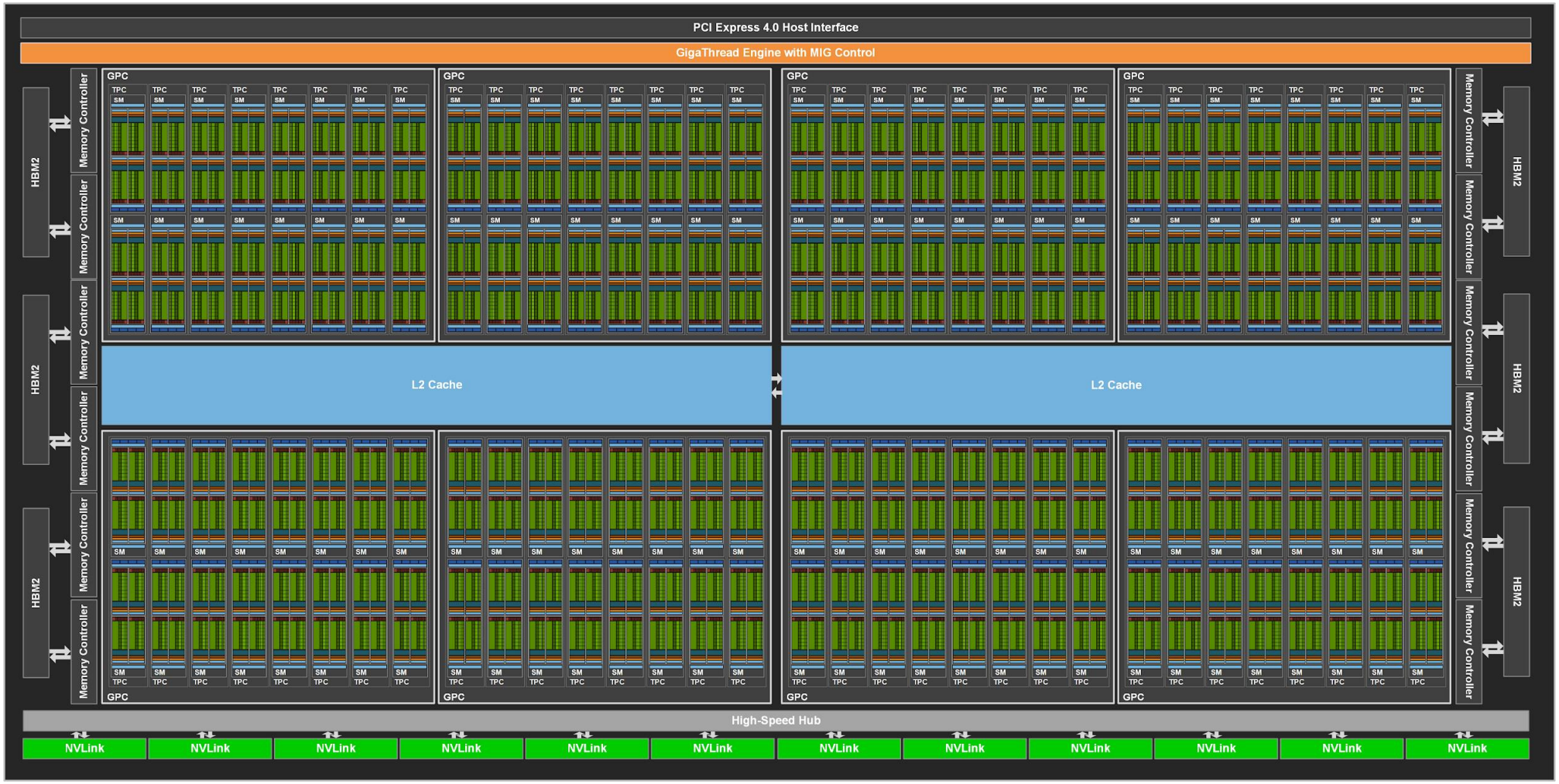
NVIDIA GA100 GPU

- Tensor Core to rdzenie zoptymalizowane do obliczeń na macierzach
- 4 Tensor Cores na multiprocesor SM
- Łącznie 512 Tensor Cores w całym GPU

- 6 HBM2 (interfejsy pamięci RAM)
- 12 512 – bitowe kontrolery pamięci

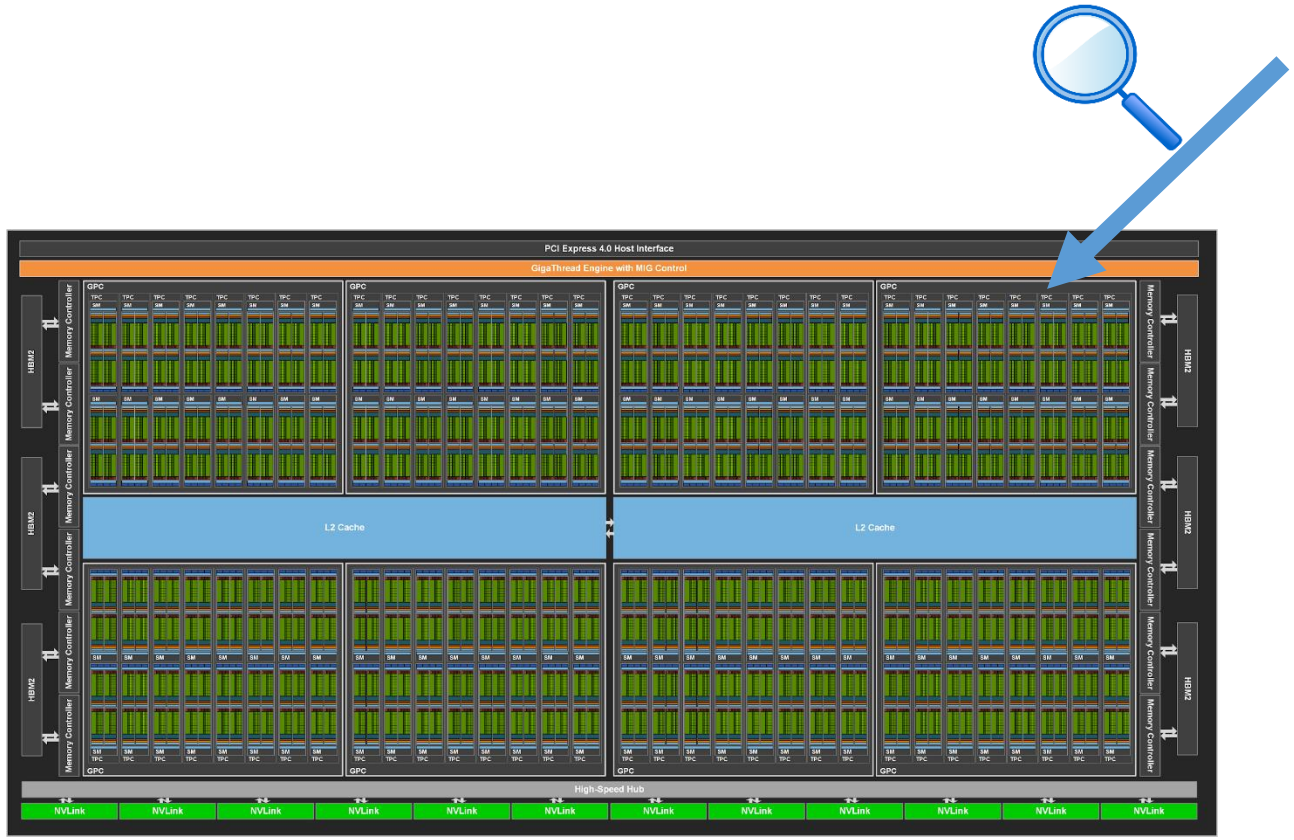
- 40 GB pamięci HBM2.
- 40 BM pamięci L2.
- 192 KB pamięci L1 w każdym multiprocesorze SM.

NVIDIA GA100 GPU



https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/

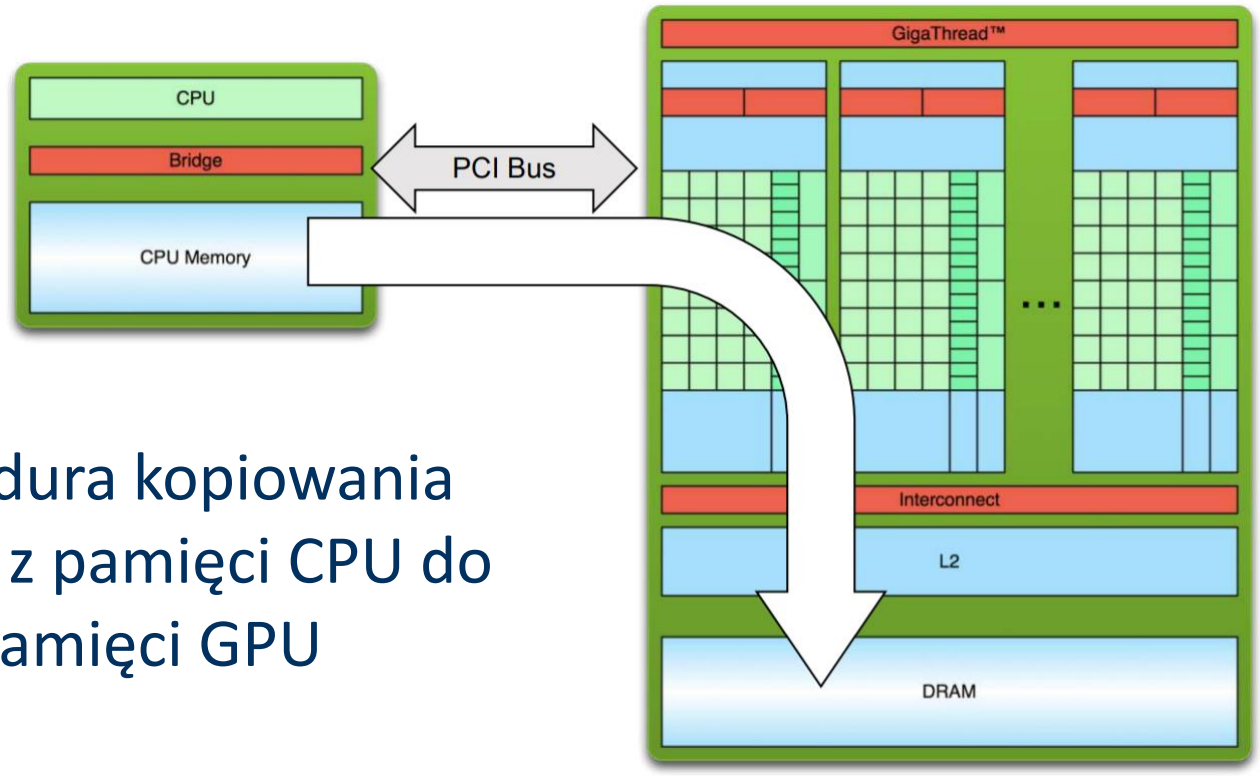
NVIDIA GA100 GPU: pojedynczy SM



https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/

Architektura heterogeniczna

- Polega na współpracy CPU oraz GPU
- CPU określane jest jako host, GPU jako device

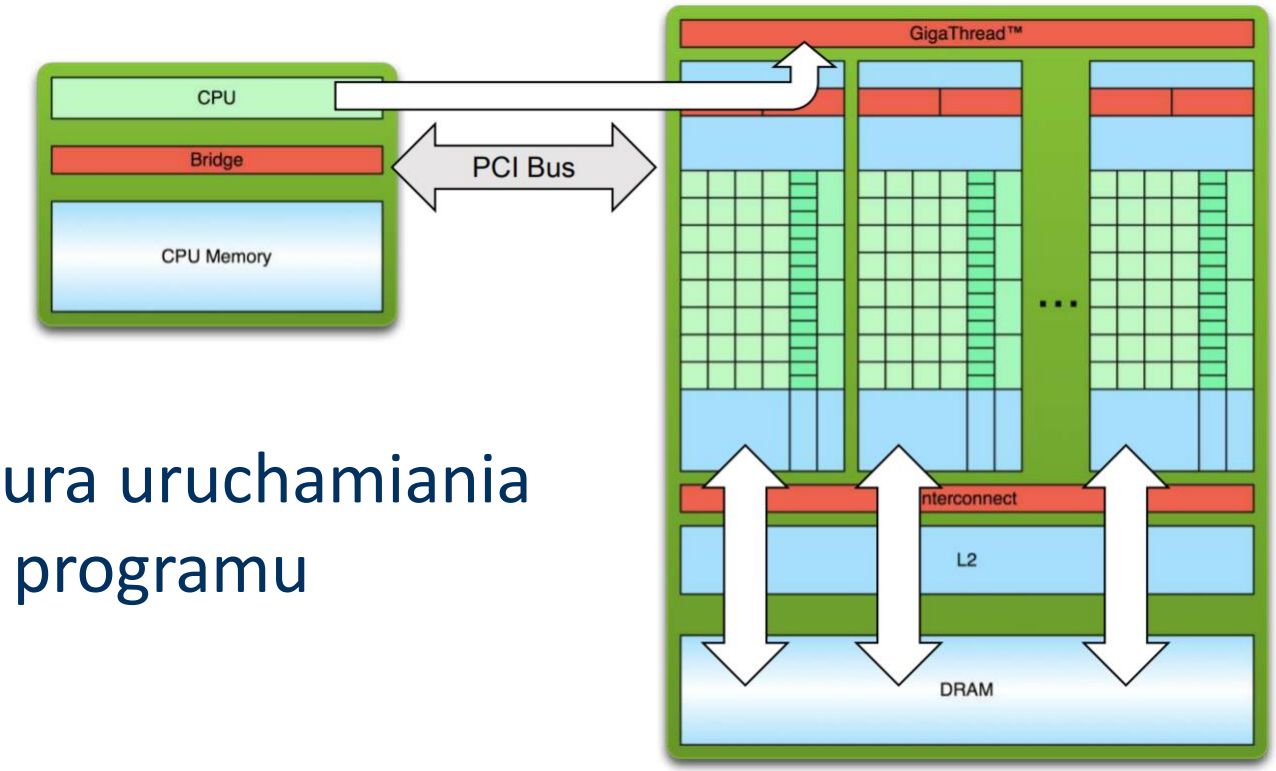


Procedura kopiowania danych z pamięci CPU do pamięci GPU

Źródło: GPU Computing and Programming, Andreas W. Götz

Architektura heterogeniczna

- Polega na współpracy CPU oraz GPU
- CPU określane jest jako host, GPU jako device

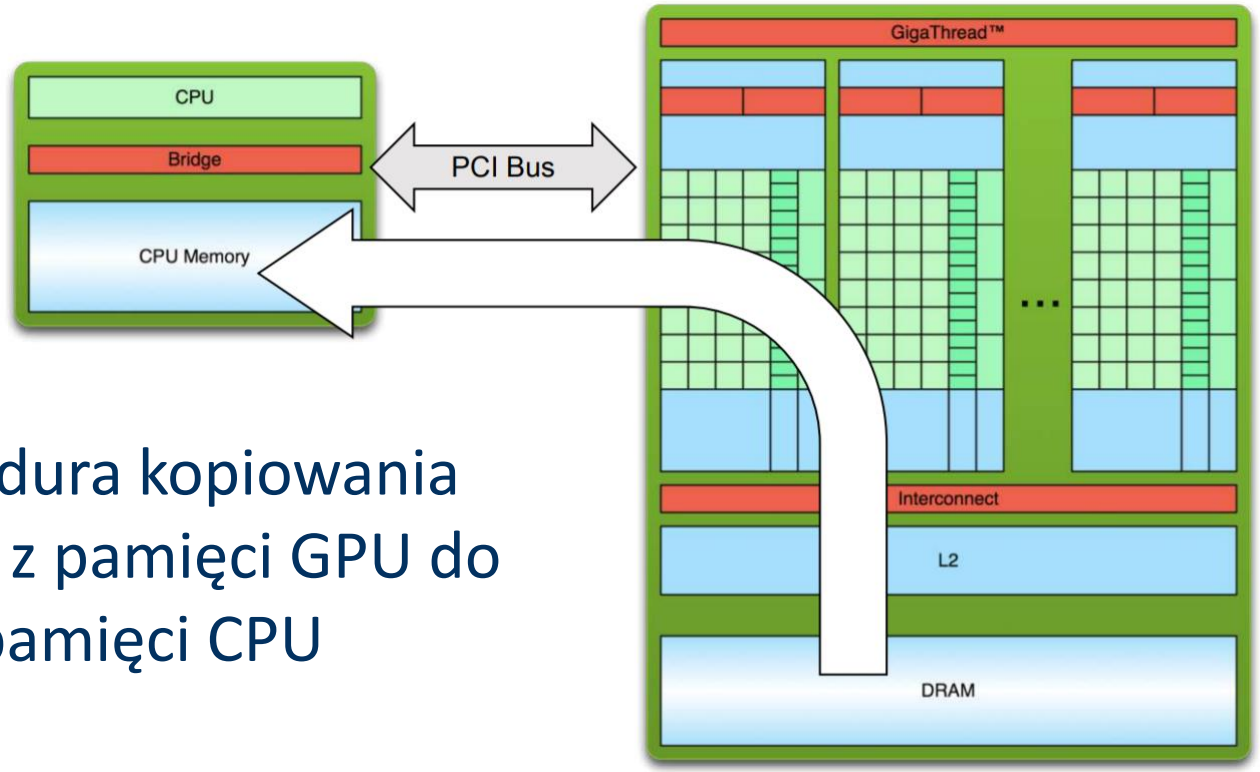


Procedura uruchamiania programu

Źródło: GPU Computing and Programming, Andreas W. Götz

Architektura heterogeniczna

- Polega na współpracy CPU oraz GPU
- CPU określane jest jako host, GPU jako device

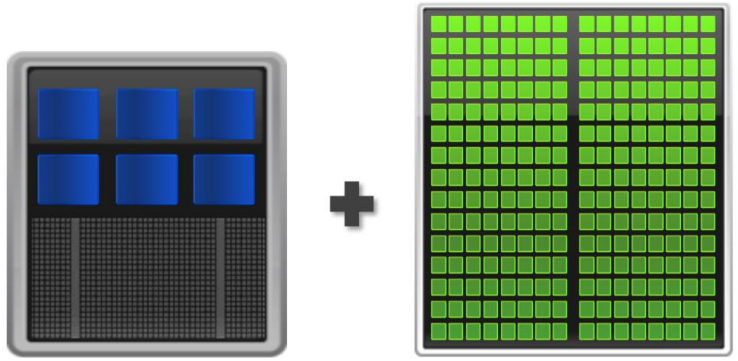


Procedura kopiowania danych z pamięci GPU do pamięci CPU

Źródło: GPU Computing and Programming, Andreas W. Götz

Architektura heterogeniczna

	Zalety	Wady
CPU	• Duża pamięć główna • Rdzenie taktowane wysoką częstotliwością • Opóźnienia zniwelowane dzięki dużej pamięci podręcznej • Wysoka wydajność pojedynczego wątku	• Stosunkowo mała przepustowość pamięci • Słaby stosunek wydajność/energia • Wydajność spada gdy danych nie ma w pamięci podręcznej
GPU	• Duża przepustowość pamięci głównej • Opóźnienia ukrywane przez równoległość • Więcej zasobów obliczeniowych • Korzystny stosunek wydajność/energia	• Stosunkowo mała pojemność pamięci • Niska wydajność pojedynczego wątku



Programowanie GPU

- C, C++, Fortran
- OpenCL (Open Computing Language) – środowisko programistyczne umożliwiające programowanie na heterogenicznych platformach różnych dostawców
- OpenACC (Open Accelerators) – standard programowania oparty na dyrektywach. Umożliwia programowanie na platformach różnych dostawców
- CUDA (Compute Unified Device Architecture) – środowisko programistyczne dostępne tylko dla kart graficznych produkowanych przez firmę NVIDIA
- ….

Dziękuję za uwagę

Kontakt:
dr hab. inż. Łukasz Szustak, prof. PCz
lszustak@icis.pcz.pl
Katedra Informatyki
Wydział Inżynierii Mechanicznej i Informatyki
Politechnika Częstochowska