



KARTA PRZEDMIOTU

Kod przedmiotu	studia stacjonarne:	E-I2G-2006-s1
	studia niestacjonarne:	E-1I22G-1001-s1
Nazwa przedmiotu	Współczesne architektury procesorów graficznych	
Nazwa przedmiotu w języku angielskim	Architecture of modern graphics processor units	
Obowiązuje od roku akademickiego	2022/23	

USYTUOWANIE PRZEDMIOTU W SYSTEMIE STUDIÓW

Kierunek studiów	Informatyka
Poziom kształcenia	II stopień
Profil studiów	Ogólnoakademicki
Forma i tryb prowadzenia studiów	Studia stacjonarne i niestacjonarne
Zakres	Grafika komputerowa
Jednostka prowadząca przedmiot	Katedra Systemów Informatycznych
Koordynator przedmiotu	Dr hab. inż. Roman Deniziak, prof. PŚk
Zatwierdził	Dziekan Wydziału Elektrotechniki, Automatyki i Informatyki dr hab. inż. Roman Deniziak, prof. PŚk

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów		Przedmiot specjalnościowy
Status przedmiotu		Obowiązkowy
Język prowadzenia zajęć		Polski
Usytuowanie w planie studiów - semestr	studia stacjonarne	Semestr I
	studia niestacjonarne	Semestr I
Wymagania wstępne		Brak
Egzamin (TAK/NIE)		Tak
Liczba punktów ECTS		5

Forma prowadzenia zajęć		wykład	ćwiczenia	laboratorium	projekt	Inne
Liczba godzin w semestrze	studia stacjonarne:	30		30		
	studia niestacjonarne:	18		18		

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty uczenia się	Odniesienie do efektów kierunkowych
Wiedza	W01	Student zna struktury współczesnych procesorów graficznych.	INF2_W06
	W02	Student zna zasady programowania równoległego heterogenicznych architektur komputerów opartych o wielordzeniowe procesory uniwersalne i procesory graficzne.	INF2_W06
	W03	Student zna zasad przetwarzania SPMD (Single Program Multiple Data) w procesorach graficznych.	INF2_W06
	W04	Student zna kierunki rozwoju architektur procesorów graficznych.	INF2_W06
Umiejętności	U01	Student potrafi projektowania równoległych algorytmów SPMD oraz ich implementowania w języku CUDA i OpenCL.	INF2_U09, INF2_U10

TREŚCI PROGRAMOWE

Forma zajęć	Treści programowe
wykład	1. Zasady przetwarzania równoległego w procesorach graficznych. 2. Model CUDA: zasady przetwarzania, programowanie w języku C. 3. Język OpenCL: zasady programowania wielowątkowego, zastosowanie do GPGPU. 4. Struktury współczesnych procesorów graficznych stosowanych w komputerach PC, stacjach graficznych, superkomputerach i systemach wbudowanych. 5. Kierunki rozwoju architektur procesorów graficznych oraz komputerów
laboratorium	1. Wstęp do programowania z użyciem GPGPU oraz OpenCL. 2. Sposoby przesyłania oraz przetwarzania tablic jednowymiarowych z użyciem OpenCL. 3. Hierarchia wątków. Architektura SIMT. Sposoby przesyłania oraz przetwarzania tablic dwuwymiarowych z użyciem OpenCL. 4. Hierarchia pamięci i synchronizacja wątków w OpenCL. 5. Implementacja wybranych algorytmów numerycznych w OpenCL. 6. Obsługa oraz podstawowe przetwarzanie obrazów w OpenCL. 7. Wstęp do programowania z użyciem CUDA. Sposoby przesyłania oraz przetwarzania tablic jednowymiarowych z użyciem CUDA. 8. Sposoby przesyłania oraz przetwarzania tablic dwuwymiarowych z użyciem CUDA. 9. CUDA. Dostęp łączony do pamięci GPU. 10. Hierarchia pamięci i synchronizacja wątków w CUDA. 11. Obsługa oraz podstawowe przetwarzanie obrazów w CUDA. 12. Interoperacyjność CUDA z OpenGL. 13. Implementacja prostego silnika fizycznego z użyciem CUDA lub OpenCL.

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów uczenia się					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01		X				
W02					X	
W03					X	
W04		X				
U01			X			X

FORMA I WARUNKI ZALICZENIA

Forma zajęć	Forma zaliczenia	Warunki zaliczenia
wykład	egzamin	Udzielenie poprawnej i wyczerpującej odpowiedzi na wszystkie pytania podstawowe (ocena 3). Poprawne odpowiedzi na pytania zaawansowane umożliwia uzyskanie wyższej oceny.
laboratorium	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów.

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS												
Lp.	Rodzaj aktywności	Obciążenie studenta										Jed- nostka
		studia stacjonarne					studia niestacjonarne					
1.	Udział w zajęciach zgodnie z planem studiów	W	C	L	P	S	W	C	L	P	S	h
		30		30			18		18			
2.	Inne (konsultacje, egzamin)	4		2			4		2			h
3.	Razem przy bezpośrednim udziale nauczyciela akademickiego	66					42					h
4.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	2,64					1,68					ECTS
5.	Liczba godzin samodzielnej pracy studenta	59					83					h
6.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	2,36					3,32					ECTS
7.	Nakład pracy związany z zajęciami o charakterze praktycznym	30					18					h
8.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	1,20					0,72					ECTS
9.	Sumaryczne obciążenie pracą studenta	125					125					h
10.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	5										ECTS

LITERATURA

1. Pharr M., Fernando F. "GPU Gems 3", Addison Wesley, 2008.
2. Edward Kandrot, Fason Sanders, "CUDA w przykładach. Wprowadzenie do ogólnego programowania procesorów GPU, Helion, 2012.
3. Aaftab Munshi, The OpenCL Specification, The Khronos Group Inc., 2011.
<http://www.khronos.org/registry/cl/specs/opencl-1.2.pdf>.
4. Sawerwain Marek, OpenCL. Akceleracja GPU w praktyce, Helion, 2014.
5. Materiały firmowe Nvidia, AMD.