**Nazwa przedmiotu:**

Równoległe implementacje metod numerycznych

**Koordynator przedmiotu:**

Marek Nałęcz i Gustaw Mazurek

**Status przedmiotu:**

Fakultatywny dowolnego wyboru

**Poziom kształcenia:**

Studia II stopnia

**Program:**

Elektronika

**Grupa przedmiotów:**

Przedmioty techniczne - zaawansowane

**Kod przedmiotu:**

RIM

**Semestr nominalny:**

3 / rok ak. 2019/2020

**Liczba punktów ECTS:**

5

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

30 - Obecność na wykładach
3 - Przygotowanie do wykładów (powtórzenie materiału przed każdym wykładem)
15 - Obecność na laboratoriach
20 - Przygotowanie do laboratoriów (5 terminów)
50 - Samodzielna praca nad projektem w domu
15 - Uruchamianie projektu w laboratorium na rzeczywistym sprzęcie
2 - Konsultacje z wykładowcą
8 - Konsultacje z prowadzącym projekt
2 - Konsultacje z prowadzącym laboratoria
9 - Przygotowanie do egzaminu (przy założeniu jednokrotnego zdawania)
1 - Obecność na egzaminie (przy założeniu jednokrotnego zdawania)

RAZEM 155

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

30 - Obecność na wykładach
15 - Obecność na laboratoriach
15 - Uruchamianie projektu w laboratorium na rzeczywistym sprzęcie
2 - Konsultacje z wykładowcą
8 - Konsultacje z prowadzącym projekt
2 - Konsultacje z prowadzącym laboratoria

RAZEM 70 => 2.3 ECTS

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

15 - Obecność na laboratoriach
20 - Przygotowanie do laboratoriów (5 terminów)
50 - Samodzielna praca nad projektem w domu
15 - Uruchamianie projektu w laboratorium na rzeczywistym sprzęcie

RAZEM 100 => 3.3 ECTS

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład:  | 30h |
| Ćwiczenia:  | 0h |
| Laboratorium:  | 15h |
| Projekt:  | 15h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

Metody numeryczne lub Wstęp do metod numerycznych lub podobny przedmiot,

Układy cyfrowe lub Podstawy techniki cyfrowej lub podobny przedmiot,

Cyfrowe przetwarzanie sygnałów lub Cyfrowe przetwarzanie obrazów lub podobny przedmiot

**Limit liczby studentów:**

30

**Cel przedmiotu:**

Celem przedmiotu jest zapoznanie studentów z nowoczesnymi technikami implementacji metod numerycznych i algorytmów przetwarzania sygnałów, w których szczególny nacisk położony jest na efektywność, uzyskaną przez wykorzystanie równoległości obliczeń, zapewnianej przez wielordzeniowe procesory i układy logiki programowalnej.

**Treści kształcenia:**

• Wykład: Ograniczenia klasycznych procesorów jednordzeniowych i sposoby ich sprzętowego pokonywania. Praca potokowa, model obliczeń SIMD, architektura superskalarna i architektura VLIW. Klasyczne rozwiązania wielordzeniowe na przykładzie procesorów x86 i CELL. Procesory systoliczne typu Tilera i SeaForth, procesory graficzne firm Nvidia i AMD. Komputery równoległe, modele programowania równoległego. Problemy zrównoleglania algorytmów, graf zależności. Historyczny rozwój procesorów graficznych. Architektura i model programowania procesorów graficznych, ze szczególnym uwzględnieniem rodziny Tesla 10. Hierarchia pamięci; pamięć stała, pamięć globalna, pamięć dzielona i pamięć tekstur. Podstawy języka PTX. Programowanie w języku CUDA na niskim poziomie, porównanie z językiem OpenCL. Programowanie w języku CUDA na wysokim poziomie (CUDA C). Podstawowe operacje programowania równoległego w języku CUDA. Wykorzystanie bibliotek algebraicznych (BLAS) i sygnałowych (FFT) do języka CUDA. Optymalizacja programów na procesory graficzne. Wzmianka o języku X10. Przykłady zastosowań kart graficznych w obliczeniach: sztuczne sieci neuronowe, modelowanie procesów fizycznych, biologicznych i geologicznych, optymalizacja, symulacje Monte-Carlo, przetwarzanie sygnałów i obrazów biomedycznych i radiolokacyjnych, przeszukiwania dużych baz danych. Równoległe realizacje metod numerycznych i algorytmów przetwarzania sygnałów wykorzystujące układy FPGA. Zaawansowane metody opisu i symulacji sprzętu, język VHDL. Wewnętrzna struktura współczesnych układów FPGA na przykładzie rodziny Xilinx Spartan 3. Typowe układy peryferyjne wykorzystywane w zastosowaniach do cyfrowego przetwarzania sygnałów, współpraca z układami pamięci wykonanymi w różnych technologiach. Ograniczenia możliwości układów FPGA. Metody przyspieszania obliczeń przez ich zrównoleglanie i wprowadzenie przetwarzania potokowego. Przykładowe zastosowania: sposoby implementacji funkcji arytmetycznych, szybkie algorytmy obliczeń przybliżonych, algorytm iteracyjny Newtona-Raphsona (m.in. w operacjach dzielenia i pierwiastkowania).
• Laboratoria:
Laboratorium ma na celu nauczenie wszystkich słuchaczy kursu sprawnego posługiwania się systemami uruchomieniowymi (zarówno w aspekcie sprzętowym, jak i programowym) dla przedstawicieli obu rozważanych platform równoległych (procesorów graficznych Nvidia i układów logiki programowalnej Xilinx). Tematy laboratoriów obejmują pięć ćwiczeń po 3h każde:
1. Implementacja podstawowych obliczeń algebry liniowej w języku CUDA C i z wykorzystaniem biblioteki CUBLAS.
2. Implementacja podstawowych algorytmów cyfrowego prze-twarzania sygnałów w języku CUDA C i z wykorzystaniem biblioteki CUFFT.
3. Implementacja metod optymalizacji i symulacji Monte-Carlo w języku CUDA C.
4. Przybliżone implementacje podstawowych funkcji arytme¬tycznych w układzie FPGA.
5. Implementacja banku filtrów i algorytmu FFT w układzie FPGA.
Laboratorium odbywa się w środkowej części semestru.
• Projekt:
W ramach zajęć projektowych, wykonywanych w drugiej połowie semestru, wiedza pozyskana na laboratoriach będzie wykorzystywana do rozwiązywania konkretnych, choć z konieczności odpowiednio uproszczonych, problemów praktycznych na jednej wybranej platformie (albo procesor graficzny Nvidia, albo układ logiki programowalnej Xilinx), w zespołach 2-4 osobowych.

**Metody oceny:**

o Pięć ćwiczeń laboratoryjnych, ocenianych w skali 0÷8 pkt. każde (razem do 40 pkt.).
o Projekt, którego poszczególne etapy oceniane są następująco:
a. Opis problemu i sposobu rozwiązania – 5 p.
b. Napisanie kodu źródłowego rozwiązania – 10 p.
c. Uruchomienie i testowanie programu – 15 p.
d. Dokumentacja końcowa – 5 p.
(razem do 35 pkt.).
o Testowy egzamin końcowy (0÷25 pkt.).

**Egzamin:**

tak

**Literatura:**

1. Fayez Gebali: Algorithms and Parallel Computing. John Wiley & Sons, 2011 (po angielsku).
2. Jason Sanders, Edward Kandrot: CUDA by Example: An Introduction to General-Purpose GPU Programming. Addison-Wesley, 2011 (po angielsku).
3. Rob Farber: CUDA Application Design and Development. Morgan Kaufmann, 2011 (po angielsku).
4. David B. Kirk, Wen-mei W. Hwu: Programming Massively Parallel Processors: A Hands-on Approach. Morgan Kaufmann, 2010 (po angielsku).
5. A. V. Boreskov, A. A. Kharlamov: Basics of CUDA Technology. DMK Press, Moscow, 2010 (po rosyjsku).
6. Aaftab Munshi, Benedict R. Gaster, Timothy G. Mattson, James Fung, Dan Ginsburg: OpenCL Programming Guide. Addison-Wesley, 2012 (po angielsku).
7. David R. Martinez, Robert A. Bond, M. Michael Vai: High Performance Embedded Computing Handbook – A Systems Perspective. CRC Press, 2008 (po angielsku).
8. A. Karbowski, E. Niewiadomska-Szynkiewicz (red.): Programowanie równoległe i rozproszone. Oficyna Wydawnicza PW, 2009.
9. J. Majewski, P. Zbysiński: Układy FPGA w przykładach. BTC 2007.
10. K. Skahill: Język VHDL. Projektowanie programowalnych układów logicznych, WNT 2001.

**Witryna www przedmiotu:**

https://studia.elka.pw.edu.pl/priv/RIM.A

**Uwagi:**

• Warunkiem zaliczenia przedmiotu jest uzyskanie łącznie co najmniej 50 punktów. Oceny wystawiane są według standardowej skali (pół stopnia co 10 punktów).
• Studenci, którzy w czasie trwania semestru osiągną wyniki wskazujące na osiągnięcie zakładanych efektów kształcenia w stopniu łącznie co najmniej dobrym, mogą zostać przez wykładowcę zwolnieni z obowiązku przystąpienia do egzaminu końcowego.

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka W\_1:**

Student posiada uporządkowaną wiedzę w zakresie wybranych metod przetwarzania równoległego

Weryfikacja:

Egzamin, laboratoria, projekt

**Powiązane charakterystyki kierunkowe:** K\_W03, K\_W04, K\_W05, K\_W06

**Powiązane charakterystyki obszarowe:**

**Charakterystyka W\_2:**

Student posiada uporządkowaną wiedzę w zakresie metodyki i technik implementacji wybranych metod numerycznych i algorytmów przetwarzania sygnałów na wielordzeniowych procesorach graficznych

Weryfikacja:

Egzamin, laboratoria 1, 2 i 3, ew. projekt

**Powiązane charakterystyki kierunkowe:** K\_W04, K\_W05, K\_W06

**Powiązane charakterystyki obszarowe:**

**Charakterystyka W\_3:**

Student posiada uporządkowaną wiedzę w zakresie metodyki i technik implementacji wybranych metod numerycznych i algorytmów przetwarzania sygnałów na układach logiki programowalnej

Weryfikacja:

Egzamin, laboratoria 4 i 5, ew. projekt

**Powiązane charakterystyki kierunkowe:** K\_W04, K\_W05, K\_W06

**Powiązane charakterystyki obszarowe:**

### Profil ogólnoakademicki - umiejętności

**Charakterystyka U\_1:**

Student potrafi napisać w języku strumieniowym i uruchomić prosty program na wielordzeniowy procesor graficzny

Weryfikacja:

Laboratoria 1, 2 i 3, ew. projekt

**Powiązane charakterystyki kierunkowe:** K\_U01, K\_U07, K\_U08, K\_U09, K\_U11, K\_U12

**Powiązane charakterystyki obszarowe:**

**Charakterystyka U\_2:**

Student potrafi napisać w języku opisu sprzętu i uruchomić prosty program dla układu logiki programowalnej

Weryfikacja:

Laboratoria 4 i 5, ew. projekt

**Powiązane charakterystyki kierunkowe:** K\_U01, K\_U07, K\_U08, K\_U09, K\_U11, K\_U12

**Powiązane charakterystyki obszarowe:**

**Charakterystyka U\_3:**

Student potrafi optymalizować kod opracowany w języku strumieniowym lub w języku opisu sprzętu przez odpowiedni dobór bibliotek i wykorzystywanych elementów architektury użytego procesora czy układu

Weryfikacja:

Laboratoria, projekt

**Powiązane charakterystyki kierunkowe:** K\_U08, K\_U10, K\_U11, K\_U12, K\_U13, K\_U07

**Powiązane charakterystyki obszarowe:**

### Profil ogólnoakademicki - kompetencje społeczne

**Charakterystyka KS\_1:**

Student potrafi pracować indywidualnie i w małym zespole nad niedużymi projektami dotyczącymi implementacji wybranych metod numerycznych i algorytmów przetwarzania sygnałów w systemach wbudowanych zawierających wielordzeniowe procesory graficzne lub układy logiki programowalnej

Weryfikacja:

Projekt

**Powiązane charakterystyki kierunkowe:**

**Powiązane charakterystyki obszarowe:**