

# Informacje ogólne

## Wykładowcy

- Marek Nałęcz, p. 446, tel.(22)234-7476  
e-mail: Marek.Nalecz@pw.edu.pl  
konsultacje: śr. N 16:00–16:45, pn. P 16:00–16:45
- Gustaw Mazurek, p. 323, tel.(22)234-7657  
e-mail: Gustaw.Mazurek@pw.edu.pl
- Jacek Gambrych, p. 456, tel.(22)234-7867  
e-mail: Jacek.Gambrych@pw.edu.pl

Regulamin jest w USOS-ie, a na serwerze studia są:

- harmonogram zajęć  
<https://studia.elka.pw.edu.pl/pl/-/103A-ELxxx-MSP-RIM/asn/stud/>
- slajdy wykładowe, *materiały do laboratoriów*, zapisy itp.  
<https://studia.elka.pw.edu.pl/f-pl/-/103A-ELxxx-MSP-RIM/priv/>
- wyniki laboratoriów, projektu i egzaminu(ów)  
<https://studia.elka.pw.edu.pl/pl/-/103A-ELxxx-MSP-RIM/info-one/>

# Cel i zakres merytoryczny zajęć

Celem przedmiotu jest nauczanie podstawowych technik **implementacji i optymalizacji** oprogramowania równoległego na platformach sprzętowych **GPU i FPGA**.

Wymagana jest umiejętność programowania w języku C i C++ oraz znajomość środowiska MATLAB.

# Harmonogram zajęć

- Wykłady – obecność **bardzo** zalecana
  - pn. 14–16, s. 17
- Laboratoria, s. 418 – 3 terminy z **GPU** [Jacek Gambrych] i 2 terminy z **FPGA** [Gustaw Mazurek] – obecność obowiązkowa, usprawiedliwia prowadzący
  - pn.(N) 17-20: 27 X, **cz. 13 XI**, 24 XI, 8 XII, 22 XII
  - śr.(N) 8-11: 29 X, 12 XI, 26 XI, 10 XII, 7 I

# Harmonogram zajęć – c.d.

- Projekty (albo z GPU – NVIDIA GeForce GTX 1650 [Jacek Gambrych], albo z FPGA – [Gustaw Mazurek]) – obowiązkowe konsultacje z prowadzącym na każdym etapie oprócz ostatniego – zapisy na platformę na stronie przedmiotu do 13 X 16:15
  - do 27 X – podział na zespoły (2–4 os.), wybór tematu
  - do 24 XI – opis problemu i sposobu rozwiązania
  - do 22 XII – napisanie kodu źródłowego rozwiązania
  - do 30 I – uruchomienie i testowanie programu w laboratorium **po wcześniejszym uzgodnieniu terminu z prowadzącym**: 12, 14, 19, 21, 26, 28 I (w godzinach laboratorium)
  - do 6 II – dokumentacja końcowa

# Tematyka wykładów – cz. 1: GPU (Nałęcz, Gambrych)

- wprowadzenie do przetwarzania równoległego
- podstawy sprzętowe i programowe technologii CUDA
- CUDA – wielowątkowość i hierarchia pamięci
- optymalizacja programów w języku CUDA
- biblioteki Thrust, CUBLAS, CURAND, CUFFT

# Tematyka wykładów – cz. 2: FPGA (Gustaw Mazurek)

- historia technologii FPGA
- przegląd rodzin układów i tendencji rozwojowych
- zasoby wewnętrzne
- ścieżka projektowa
- podstawy języka VHDL: operatory, instrukcje, style opisu
- wykorzystanie specjalizowanych bloków FPGA
- realizacja operacji arytmetycznych
- koncepcja „arytmetyki rozproszonej”
- szybkie algorytmy obliczeń przybliżonych
- operacje zmiennoprzecinkowe na platformie FPGA
- metodyka projektowania *System on Chip*

# Tematyka wykładów – cz. 3: GPU++ (Marek Nałęcz)

- programowanie niskopoziomowe w języku PTX
- „niezależny” od platformy sprzętowej język OpenCL
- języki programowania Rust i Fortran
- dyrektywy OpenACC
- obliczenia kwantowe

# Wykaz literatury – GPU

- 1 J. Sanders, E. Kandrot: *CUDA w przykładach. Wprowadzenie do ogólnego programowania procesorów GPU*. Helion, 2012
- 2 D. B. Kirk, W. W. Hwu: *Programming Massively Parallel Processors: A Hands-on Approach. Third Edition*. Morgan Kaufmann, 2017
- 3 R. Farber: *CUDA Application Design and Development*. Morgan Kaufmann, 2011
- 4 S. Cook: *CUDA Programming. A Developer's Guide to Parallel Computing with GPUs*. Morgan Kaufmann, 2013
- 5 N. Wilt: *The CUDA Handbook. A Comprehensive guide to GPU Programming*. Addison-Wesley, 2013
- 6 A. B. Боресков, А. А. Харламов: *Основы работы с технологией CUDA*. ДМК, Москва, 2010
- 7 A. Munshi i in.: *OpenCL Programming Guide*. Addison-Wesley, 2012



# Wykaz literatury – FPGA (po polsku)

- ❶ M. Zwoliński: *Projektowanie układów cyfrowych z wykorzystaniem języka VHDL*, WKŁ, 2002
- ❷ K. Skahill: *Język VHDL, Projektowanie programowalnych układów logicznych*, WNT, 2001
- ❸ J. Majewski, P. Zbysiński: *Układy FPGA w przykładach*, BTC, 2007
- ❹ M. Nowakowski, *PicoBlaze. Mikroprocesor w FPGA*, BTC, 2009
- ❺ Z. Hajduk: *Wprowadzenie do języka Verilog*, BTC, 2009

# Wykaz literatury – FPGA (po angielsku)

- ❶ C. Maxfield: *The Design Warrior's Guide to FPGAs Devices, Tools, and Flows*, Elsevier, 2004 (dostępna w sieci jako e-Book)
- ❷ J. Van der Spiegel: *VHDL Tutorial*,  
[http://www.seas.upenn.edu/~ese171/vhdl/vhdl\\_primer.html](http://www.seas.upenn.edu/~ese171/vhdl/vhdl_primer.html)
- ❸ P. J. Ashenden: *The VHDL Cookbook*,  
<http://tams-www.informatik.uni-hamburg.de/vhdl/doc/cookbook/VHDL-Cookbook.pdf>
- ❹ U. Meyer-Baese: *Digital signal processing with field programmable gate arrays*, Springer, 2001

# Wykaz literatury – zagadnienia ogólne

- ❶ P. Krzyżanowski: *Obliczenia inżynierskie i naukowe. Szybkie, skuteczne, efektowne*. Wyd. Naukowe PWN, 2011
- ❷ F. Gebali: *Algorithms and Parallel Computing*. John Wiley & Sons, 2011
- ❸ D. R. Martinez, R. A. Bond, M. M. Vai: *High Performance Embedded Computing Handbook – A Systems Perspective*. CRC Press, 2008
- ❹ A. Karbowski, E. Niewiadomska-Szynkiewicz (red.): *Programowanie równoległe i rozproszone*. Oficyna Wydawnicza PW, 2009
- ❺ R. G. Lyons: *Wprowadzenie do cyfrowego przetwarzania sygnałów*, WKŁ, 2000
- ❻ Ch. Bernhardt: *Obliczenia kwantowe dla każdego*. PWN, Warszawa, 2020

# Zasady zaliczania w trakcie semestru

- 40 p. Laboratoria (5 ćwiczeń po 8 p.) – praca samodzielna
- brak „wejściówki”, ew. praca domowa
  - nieobecni usprawiedliwieni powinni *niezwłocznie* zgłosić się do prowadzącego – ew. będą mogli odrabiać zaległe ćwiczenie za jego zgodą na terminach projektowych
- 35 p. Projekt – praca w zespołach 2–4 osobowych
- 0 p. uzgodnienie tematu
  - 5 p. opis problemu i sposobu rozwiązania
  - 10 p. napisanie kodu źródłowego rozwiązania
  - 15 p. uruchomienie i testowanie programu
  - 5 p. dokumentacja końcowa
- 25 p. Egzamin pisemny (**zwolnienie od 65 p.** z laboratoriów i projektu w ostatnim dniu przed sesją)
- test otwarty zawierający ok. 15 pytań (raczej wykładowych)
  - nie można korzystać z notatek, książek itp.
  - w przypadku niezaliczenia możliwy termin poprawkowy – sesja zimowa i jesienna (z zapisami!)

Maksymalnie 100 p., zaliczenie od 50 p. (co 10 p. pół stopnia).

# Efekty uczenia się – po zaliczeniu RIM student:

- posiada uporządkowaną wiedzę w zakresie wybranych metod przetwarzania równoległego i technik implementacji wybranych metod numerycznych i algorytmów przetwarzania sygnałów na GPU i FPGA
- potrafi napisać w odpowiednim języku i uruchomić prosty program na GPU i FPGA
- potrafi optymalizować kod opracowany w języku strumieniowym lub w języku opisu sprzętu przez odpowiedni dobór bibliotek i wykorzystywanych elementów architektury użytego procesora czy układu
- umie pracować indywidualnie i w małym zespole nad niedużymi projektami dotyczącymi implementacji wybranych metod numerycznych i algorytmów przetwarzania sygnałów w systemach wbudowanych zawierających GPU lub FPGA