GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Parallel Programming, PG_00068272						
Field of study	Automatic Control, Cybernetics and Robotics						
Date of commencement of studies	October 2026		Academic year of realisation of subject	2027/2028			
Education level	first-cycle studies		Subject group	Optional subject group Subject group related to scientific research in the field of study			
Mode of study	Full-time studies		Mode of delivery	at the university			
Year of study	2		Language of instruction	Polish			
Semester of study	4		ECTS credits	3.0			
Learning profile	general academic profile		Assessment form	assessment			
Conducting unit	Department of Decision Systems and Robotics -> Faculty of Electronics Telecommunications and Informatics -> Faculties of Gdańsk University of Technology						
Name and surname of lecturer (lecturers)	Subject supervisor		dr hab. inż. Tomasz Stefański				
	Teachers		dr hab. inż. Tomasz Stefański				
Lesson types	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	15.0	0.0	15.0	0.0	0.0	30
	E-learning hours included: 0.0						
Learning activity and number of study hours	Learning activity	Participation in didactic classes included in study plan		Participation in consultation hours	Self-study		SUM
	Number of study hours	30		5.0	40.0		75
Subject objectives	To familiarize students with parallel programming of multiprocessor systems, i.e. graphics cards, central processing units and their clusters, with particular emphasis on programming in the CUDA environment.						

Learning outcomes

Course outcome	Subject outcome	Method of verification
[K6_U04] can apply knowledge of programming methods and techniques as well as select and apply appropriate programming methods and tools in computer software development or programming devices or controllers using microprocessors or programmable elements or systems specific to the field of study	The student is able to write engineering programs in the CUDA environment.	[SU4] Assessment of ability to use methods and tools [SU1] Assessment of task fulfilment
[K6_U12] can analyze the operation of components, circuits and systems related to the field of study, as well as measure their parameters and examine technical specifications, and plan and conduct experiments related to the field of study, including computer simulations and measurements, and interpret obtained results and draw conclusions	The student is able to analyze the operation of parallel programs written in C/C++ languages using multiprocessor systems.	[SU3] Assessment of ability to use knowledge gained from the subject [SU4] Assessment of ability to use methods and tools
[K6_W04] knows and understands, to an advanced extent, the principles, methods and techniques of programming and the principles of computer software development or programming devices or controllers using microprocessors or programmable elements or systems specific to the field of study, and organisation of systems using computers or such devices	The student understands the principles of parallel programming and is able to write programs that use multi-core processors and graphics cards.	[SW1] Assessment of factual knowledge

Subject contents

Course content – lecture
1. Introduction to parallel programming.
2. Architectures of multiprocessor systems, i.e. graphics cards, central units and their clusters, and their programming standards.
3. OpenMP and MPI programming standards.
4. Introduction to the CUDA environment for programming graphics cards.
5. Discussion of the compiler (nvcc), debugger (cuda-gdb) and profiler in the CUDA environment.
6. Parallel programming model in the CUDA environment: computational kernels, threads, thread blocks, memory hierarchy.
7. CUDA C programming interface.
8. Code optimization in the CUDA environment.
9. Review of programming libraries available within the CUDA environment.
10. Introduction to the OpenCL standard. Heterogeneous multiprocessor systems. Application of the OpenCL standard to programming field-programmable gate arrays (FPGA).
11. Parallel programming model in the OpenCL standard: computational kernels, work units, work groups, memory hierarchy, sequences of operations.

Prerequisites and co-requisites

Student has a basic knowledge of programming.

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
project	40.0%	50.0%
laboratory	40.0%	30.0%
lecture	40.0%	20.0%

Recommended reading

Basic literature	- CUDA C Programming Guide - CUDA Runtime API - CUDA C Best Practices Guide
Supplementary literature	- CUDA Compiler Driver NVCC - CUDA Visual Profiler - CUDA-gdb debugger - CUDA-memcheck memory checker
eResources addresses	

Example issues/ example questions/ tasks being completed

1. CUDA parallel programming model.
2. Memories on the graphics card.
3. The use of shared memory.
4. Streams in CUDA.
5. Events in CUDA.

Practical activites within the subject

Not applicable