POZNAN UNIVERSITY OF TECHNOLOGY
EUROPEAN CREDIT TRANSFER AND ACCUMULATION SYSTEM (ECTS)
pl. M. Skłodowskiej-Curie 5, 60-965 Poznań

COURSE DESCRIPTION CARD - SYLLABUS

Course name

Parallel processing in embedded systems

Course

Field of study

Computer Science

Area of study (specialization)

Computer microsystems

Level of study

Second-cycle studies

Form of study

full-time

Year/Semester

2/3

Profile of study

general academic

Course offered in

polish

Requirements

compulsory

Number of hours

Lecture

20

Tutorials

Laboratory classes

15

Projects/seminars

15

Other (e.g. online)

Number of credit points

4

Lecturers

Responsible for the course/lecturer:

dr inż. Adam Turkot

email: adam.turkot@put.poznan.pl

tel. 61 6652284

ul. Piotrowo 3a, 60-965 Poznań

Responsible for the course/lecturer:

dr inż. Mariusz Naumowicz

email: mnaumowicz@cs.put.poznan.pl

tel. +48 61 665-2364

Wydział Informatykiul.

Piotrowo 3 60-965 Poznań

Prerequisites

A student beginning this course should have basic knowledge of digital electronics, structural and object-oriented programming, as well as knowledge of embedded systems architecture. In addition, the ability to use the parallel system design environment of electronic equipment manufacturers such as NVIDIA (CUDA Toolkit) or Xilinx (ISE, XPS, Vivado) and the ability to obtain information from designated sources is required. It also requires the ability to expand knowledge and work in a team. Due to the social competences, the student should be aware that knowledge in computer science quickly becomes obsolete and requires constant broadening. The student should present an attitude of honesty, creativity, reliability, cognitive curiosity and show respect for other people.

Course objective

Providing students with basic knowledge of parallel processing in embedded systems. Familiarize students with popular software and hardware tools supporting parallel processing in embedded systems. Presentation of case studies illustrating different realizations of parallel processing in embedded systems due to their application. Developing in students the skills of practical application of knowledge in the field of parallel processing in embedded systems for the realization of the set design tasks.

Course-related learning outcomes

Knowledge

1. The graduate knows in detail the construction of tools supporting parallel processing in embedded systems.
2. The graduate knows trends in computer science and other areas that can be useful in designing parallel processing in embedded systems.
4. The graduate shall be familiar with the leading trends in the field of embedded systems as regards the algorithms and languages used to support parallel processing in embedded systems.

Skills

1. The graduate is able to use simulators and other tools and is able to draw conclusions based on the results, which allow to predict the actual operation of the system.
2. The graduate is able to use his/her previous skills in digital electronics design and programming to perform a specific task related to parallel processing in embedded systems.
3. The graduate is able to use modern tools in order to perform a specific task related to parallel processing in embedded systems.
4. The graduate is able to make a critical analysis and introduce his own improvements in the existing embedded system.
5. The graduate is able to solve unusual tasks connected with parallel processing in the embedded system.

Social competences

1. The graduate is ready to continuously broaden his knowledge in the area of embedded systems, especially in the area of changing trends in the IT market development.
2. The graduate understands the necessity of applying the latest solutions during the implementation of microsystems.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formal evaluation:

(a) for lectures: on the basis of answers to questions concerning the material discussed in previous

lectures
(b) for laboratorys/exercises: based on an assessment of the current progress of the tasks,
Summary evaluation:
a) in the scope of lectures, the verification of the assumed educational results is carried out by the assessment of knowledge and skills on the basis of sheets of paper;
b) in the field of laboratories, the verification of the assumed educational results is carried out by: - assessment of the report prepared partly during the classes and partly after their completion; this assessment also includes the ability to work in a team; it concerns design exercises of a reproducible character (the student performs the exercise according to the instruction provided)- assessment of realization of a complex task requiring integration of knowledge and skills acquired during design classes; technical aspects of realization, ability to solve unconventional problems and proficiency in using available design tools are assessed

Programme content

Introduction to parallel processing using the GPU. CUDA API. CUDA memory model. CUDA Tools. Threads. Memory. Memory bank conflicts. Parallel threads. Flow control. Precision
The laboratory exercise programme includes the following issues: Launching the CUDA environment on a prototype plate. Arithmetic actions in the CUDA environment. Support of shared memory. Parallel thread processing. Emulation and project profiling. Simple image filter.
Project implementation program: Using previously acquired knowledge from other lectures in the implementation of a selected project (advanced image filter, advanced sound filter, artificial neural networks...) operating in CUDA environment.

Teaching methods

Teaching methods:
lecture: multimedia presentation, presentation illustrated with examples given on the board, presentation of selected student solutions.
laboratory exercises: practical exercises, performing experiments, discussion, teamwork.

Bibliography

Basic
1. CUDA Programming: A Developer's Guide to Parallel Computing with GPUs, Shane Cook, Morgan Kaufmann Publishers Inc. San Francisco, 2012.

2. GPU Computing Gems Emerald Edition, Wen-mei W. Hwu, Morgan Kaufmann Publishers Inc. San Francisco, 2011.

3. Heterogeneous Computing with OpenCL, Benedict Gaster Lee Howes David Kaeli Perhaad Mistry Dana Schaa, Morgan Kaufmann Publishers Inc. San Francisco, 2012.

Additional

1. Professional CUDA C Programming, John Cheng, Max Grossman, Wrox, ISBN: 978-1-118-73932-7, 2014.

2. https://developer.nvidia.com/

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4
Classes requiring direct contact with the teacher	50	2
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) [1]	50	2

[1] delete or add other activities as appropriate