Hochschule München University of Applied Sciences

Courses in English
Course Description

Department	07 Computer Science and Mathematics
Course title	**Hardware/Software Codesign**
Course number	
Hours per week (SWS)	4
Number of ECTS credits	5
Course objective	Students analyze complex software algorithms Students identify candidates of software for hardware acceleration Students estimate the gain and costs of hardware acceleration Students use modern design tools to write hardware accelerators Students use modern design tools to enhance processor cores Students design optimized algorithms using specific hardware extensions Students apply iterative tools to perform design space exploration
Prerequisites	Basic understanding of IT systems and digital logic
Recommended reading	G. DeMicheli, ”Hardware/Software Co-Design”, 2013 P. Schaumont, ”A Practical Introduction to Hardware/Software Codesign”, 2010 Online resources
Teaching methods	Slides, online resources, problem-based learning
Assessment methods	
Language of instruction	English
Name of lecturer	Prof. Dr. Stefan Wallentowitz
Email	stefan.wallentowitz@hm.edu
Link	
Course content	With the ever increasing complexity of IT systems, the partitioning between hardware and software becomes increasingly important. In particular, the design of System-on-Chip tailored to the needs of specific applications or domains is important to develop platforms that are optimized for performance and power efficiency. Advanced driver assistant systems, machine learning and edge computing are for example domains, where designs can be potentially be optimized for algorithms. Hardware/Software-Codesign are tools and methods that target the partitioning of algorithms between software and hardware, along with the early development of software throughout the hardware design process. In this module, students experience the fundamentals of algorithm design for domain-specific applications. Without knowledge of hardware design, students are able to create optimized hardware elements, either as accelerators or as instruction set extensions. With the help of modern tooling, like model-based design and high-level synthesis, and modern hardware construction languages, students can explore hardware accelerators and automate the design space exploration. Hardware-Prototyping with Field Programmable Gate Arrays (FPGA) are used to prototype optimized hardware-designs and evaluate the impact on an algorithm. Together with case studies from industry, students gain a deep understanding about the tradeoffs of hardware and software partitioning.
Remarks	