

<b>Module title</b> GPU Computing			
<b>Module code</b> tbd	<b>Level</b> Bachelor (B.Sc.)	<b>ECTS credits</b> 5	<b>Duration</b> virtual lecture
<b>Module instructor</b> Dr. Wai-Kong Lee, Gachon University, South Korea	<b>Lecture type</b> Interactive seminar Individual consultations	<b>Prerequisite(s)</b> Good academic standing in 3 <sup>rd</sup> study year	<b>Grading</b> Exam Mini Project Practical Assessment
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>To provide an understanding in the aspects of hardware, software, programming environment and performance profiling for general purpose computing in GPU.</li> <li>To develop the knowledge and skills for designing parallel processing applications using GPU.</li> <li>To study the techniques for optimizing parallel algorithms in GPU platform.</li> </ul>			
<p><b>Content</b></p> <p>This subject introduces the concepts, languages, techniques, and patterns for general purpose GPU computing. GPU can be used as massively parallel co-processor to parallelize many serial algorithms as well as accelerate existing parallel algorithms. It covers GPU architectures, data-parallel programming models, techniques for memory bandwidth optimization and parallel algorithm patterns. The students will learn the techniques to develop parallel applications in GPU platform and evaluate its performance.</p> <ul style="list-style-type: none"> <li>Topic 1: Introduction to parallel programming platforms and system architectures Flynn's Taxonomy; Homogeneous (CPU) and Heterogeneous (CPU + GPU); computing system; Vertical scaling vs. Horizontal scaling; Introduction to Parallel programming languages (CUDA, OpenMP and OpenCL).</li> <li>Topic 2: Introduction to basic parallel programming concepts Sequential programming vs. parallel programming paradigms; Identifying overheads and bottleneck of sequential application.; Data sharing and synchronization; Well known parallel solutions such as partitioning, and divide-and-conquer; Techniques to identify concurrency opportunities.</li> <li>Topic 3: GPU Architecture and Programming Model Introduction to GPU memory model in GPU (global, shared, register, constant and texture memory); Programming model for GPU: Single Instruction Multiple Data (SIMD); Grid, blocks and thread blocks; Introduction to GPU programming language.</li> <li>Topic 4: Performance Metrics for Parallel Systems Parallel performance metrics (total overhead, speedup, efficiency); Amdahl's Law vs. Gustafson's Law; Parallel Overhead; Profiling tools for GPU computing.</li> <li>Topic 5: GPU Memory Model Common techniques for parallelizing serial code in GPU; Global memory bandwidth (coalesced memory access pattern); Shared memory and bank conflict; Constant and texture memory; Register spilling and local memory.</li> <li>Topic 6: Optimization Techniques Identifying bottleneck for parallel program (memory bound or compute bound); Concurrent execution of CPU program, GPU kernel and memory copy process; Thread blocks ordering; Occupancy; Stream programming model.</li> <li>Topic 7: Mini Project - The students will be given a list of algorithms to choose for parallel implementation. The students need to implement and optimize the selected algorithms using GPU. Example algorithms: Encryption: AES, IDEA, Threefish; Hash Function: BLAKE, Keccak, SHA-1, SHA-2; Public Key Cryptography (Montgomery Multiplication, Karatsuba Multiplication); KNN; Binary Tree, Red Black Tree; Pseudorandom Number Generator; Matrix Solver (Dense or Sparse, Direct or Iterative); Various Search and Sort algorithms; Etc.</li> </ul>			
<p><b>Textbook/software</b> Lecturer provided materials on e-learning platform.</p>			

Note: this is not the official course descriptor according to the "Studien- und Prüfungsordnung" (SPO)