University of Tehran
School of Electrical and Computer Engineering

Course:	**8101761 – Parallel Programming**
Course type:	Credit: 3

	EE*						CE*		
	Com	E	P	B	Con	D	SW	HW	IT
Required	☐	☐	☐	☐	☐	☐	☐	☒	☐
Elective	☐	☐	☐	☐	☐	☐	☐	☒	☐

Level:	Undergraduate ■ Graduate ☐
Co-requisite(s):	None.
Prerequisite(s):	Computer Architecture (8101423)
Prerequisite by topic:	
Textbook(s):	[1] Christopher J. Hughes, "Single-Instruction Multiple-Data Execution", Morgan & Claypool Publishers, 2015. [2] Rohit Chandra, Leonardo Dagum, Dave Kohr, Dror Maydan, Jeff McDonald, and Ramesh Menon, "Parallel Programming in OpenMP", Morgan Kaufmann, 2001. [3] Jason Sanders, Edward Kandrot, "CUDA by Example: An Introduction to General-Purpose GPU Programming," Addison Wesley, 2011.
Coordinator:	Saeed Safari
Goals:	• Parallel Architectures • Different levels of parallelism • Parallel programming using different parallel libraries • Using Intel parallel programming tools • GPGPU architectures and programming • Real-life examples
Outcome:	Upon successful completion of the course, students will 1. Get familiar with SIMD architecture 2. Be able to detect the data parallelism in the program and use SIMD programming to exploit data parallelism. 3. Get familiar with multi-core architectures. 4. Get familiar with multi-threaded programming concepts to use multi-core processors. 5. Get familiar with the architecture of all-purpose graphics processors. 6. Get familiar with CUDA programming language for programming all-purpose graphics processors. 7. Learn parallel programming challenges 8. Learn to use Intel's parallel compiler 9. Get familiar with different parallel libraries

Topics:	• Data Level Parallelism (DLP)
	  ○ Intel SIMD Architecture
	  ○ SIMD Programming
	• Thread Level Parallelism (TLP)

	○ Intel Multicore Architectures ○ Process/Thread Architectures ○ OpenMP ○ pThread ○ Intel Integrated Performance Primitives (IPP) • General Purpose Graphic Processor Unit (GPGPU) ○ GPGPU Architecture ○ GPGPU Programming Using CUDA
Computer usage:	The student use C/CUDA compilers They also use different parallel libraries including SIMD, OpenMP, and POSIX to develop parallel programs
Assignments:	Several computer assignments will be assigned during the term to test students skills to exploit different level of parallelism in different classes of applications
Projects:	The final project is dedicated to parallel implementation of a real-life application.
Grading:	Assignments and Projects: 35 % Midterm exams: 25% Final exam: 40 %
Further readings:	[1] Michael Quinn, "Parallel Programming in C with MPI and OpenMP", McGraw-Hill, 2003. [2] Alex Peleg and Uri Weiser, "MMX Technology Extension to The Intel Architecture" IEEE Micro, 1996. [3] Cameron Hughes, Tracey Hughes, "Professional Multicore Programming Design and Implementation for C++ Developers", Wiley Publishing, Inc, 2008. [4] Stewart Taylor, "Intel Integrated Performance Primitives: How to Optimize Software Applications Using Intel IPP", Intel PRESS, 2004. [5] Nicholas Wilt, "The CUDA Handbook: A Comprehensive Guide to GPU Programming", Addison Wesley, 2013. [6] Additional papers, technical reports, and online materials.
Prepared by:	Saeed Safari
Date:	November, 2017

*EE: Electrical Engineering		CE: Computer Engineering	
Com	Communications	SW	Software
E	Electronics	HW	Hardware
P	Power	IT	Information Technology
B	Bioelectronics		
Con	Control		
D	Digital System		