JOINT INSTITUTE
交大密西根学院

Standard Undergraduate Course Profile

COURSE NUMBER: Ve370	**COURSE TITLE:** Introduction to Computer Organization
CREDIT: 4	**PREREQUISITES:** Ve270 and Ve280
TEXTBOOKS/REQUIRED MATERIAL: David Patterson and John Hennessy, Computer Organization and Design - Hardware/Software Interface, 4th edition, Morgan Kaufmann, 2008, ISBN 978-0-12-374493-7	**PREPARED BY:** Gang Zheng **DATE OF PREPARATION:** September 29, 2012 **DATE OF DISCIPLINE GROUP APPROVAL:** **DATE OF UC APPROVAL:** Oct. 30, 2013
CATALOG DESCRIPTION: This course is designed to cover basic concepts of computer organization and hardware. Instructions executed by a processor and how to use these instructions in simple assembly-language programs. Stored-program concept. Datapath and control for multiple implementations of a processor. Performance evaluation, pipelining, caches, virtual memory, input/output.	**COURSE TOPICS:** 1. Introduction to computer 2. MIPS instruction set, operations and operands 3. Instruction encoding, addressing mode 4. Procedures calling conventions, memory usage 5. Assembly programing examples, translating software 6. Single cycle processor 7. Pipelined datapath and control 8. Data hazards 9. Control hazards, exceptions 10. Cache 11. Virtual memory 12. I/Os and interfaces 13. Parallelism, multiprocessors
COURSE STRUCTURE/SCHEDULE: Two 90-minute lectures/week; Six 90-minute discussions/week; Four or more recitation classes.	
COURSE OUTCOMES [Student Outcomes in brackets]	After completing Ve370, students should be able to do the following: 1. Given a simple programming task and an instruction-set architecture, write an assembly language program that implements the task, translate the assembly-language program into machine-level instructions, and trace the execution of the program. [1, 2] 2. Model the computer hardware including datapath and control logic for a given instruction-set architecture, both for a single-cycle and pipelined processor, by using schematic capturing tools or hardware description languages (HDLs). [1, 2, 5, 6] 3. Be able to identify and resolve potential data, control, and structural hazards [1, 2] 4. Understand the memory hierarchy including cache, main memory, hard disk, and how data is stored in that hierarchical structure, and be able to recognize memory hits and misses [1, 2, 6] 5. Understand the memory mapped I/O concept and how I/O devices interface the CPU 6. Be able to use library and internet resources for literature research to learn the current issues, technologies, and future development trends in computing [7]
COURSE OBJECTIVES [Course Outcomes in brackets]	1. To teach students how computers execute machine-level instructions. [1] 2. To teach students how to write assembly language programs and translate them to machine level instructions. [1] 3. To teach students how to design the datapath and control unit for pipelined and non-pipelined processors. [2] 4. To teach students about data and control hazards. [3] 5. To teach students the principles of caches and memory. [4] 6. To teach students how processors, memory, and I/O are combined into a computer. [5] 7. To give students the experience of acquiring new knowledge using available resources [6]
ASSESSMENT TOOLS [Course Outcomes in brackets]	Homework [1-6] Midterm Exam [1,3] Final Exam [1-5] Course projects [1,2,3,4,6]

Rev 1: June 2015; Rev 2: July 2020