University of Texas at El Paso
Course Syllabus

COURSE DESCRIPTION

Dept., Number	CS3432	**Course Title**	Computer Organization
Approval Date	February 2025	**Course Coordinator**	Shirley Moore

CATALOG DESCRIPTION

Compile and assembly processes; machine organization; fetch/decode/execute process; symbolic coding of instructions and data, including instruction types, formats, and addressing modes; implementation of data and control structures, subroutines, and linkage; exception handling.

TEXT BOOK

Kerningham, Brian W & Ritchie, Dennis M. "The C Programming Language, Second edition," Prentice Hall, ISBN: 0-13-115817-1.
David A. Patterson and John L. Hennessy. Computer Organization and Design RISC-V Edition: The Hardware Software Interface, Second edition, Morgan Kaufmann, ISBN: 978-0128203316 OR Instructor notes.

COURSE OUTCOMES

Subject Areas (and abbreviations):
Hardware/software interface (HSI)
Architecture (A)
Numeric Representation (NR)
Linearization (L)
Tools (T)

Level 1: Knowledge and Comprehension:

Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. Upon successful completion of this course, students will able to:

a. Define and explain the purpose of an instruction set architecture (ISA). (A)

b. Explain the relationship and differences between a high-level programming language, assembly language, and machine language. (HSI)

c. Describe the fetch-execute cycle in terms of the hardware-software interface between machine instructions and processor components. (HSI)

d. Describe modifications needed to the classical von Neumann architecture for support of multithreaded execution. (HSI)

e. Explain the relationship between a high-level language basic data type (e.g., signed or unsigned integer, floating point number) and its representation as a bit pattern inside the computer. (NR)

f. Describe the basic components of a processor (e.g., register file, special-purpose registers, control unit, memory) and how they interact with one another. (A)

g. Describe the basic components of a sample heterogeneous node architecture (e.g., multicore CPU + accelerator(s)) and how they interact with one another. (A)

h. Explain how procedures are supported by the ISA and the compiler. (HSI)

i. Explain various ways an operand can be addressed in an assembly language instruction. (HSI)

j. Describe the process of compiling/assembling, linking, loading, and executing a program. (HSI)

k. Explain the machine language encoding of an assembly language instruction. (HSI)

l. Describe how trends in power consumption, data sizes, and application workloads have driven changes in system architectures. (A)

m. Explain the datapath through the processor for a given class of instructions (e.g., arithmetic-logical, memory access, conditional branch). (HSI)

n. Describe instruction-level parallelism using pipelining and how hazards cause stalls in a typical processor pipeline. (A)

Level 2: Application and Analysis:

Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details.
Upon successful completion of this course, students will be able to:

~~a.~~ Convert between different integer data representations (e.g., decimal, binary, hexadecimal). (NR)

~~b.~~ Use bitwise operators to access and manipulate values stored in a subset of bits within a byte or word. (NR)

c. Determine range and precision (if applicable) of numbers that can be stored for a given data type. (NR)

d. Trace the execution of an assembly language program with procedure calls, including allocation and deallocation of stack frames. (HSI)

e. Trace the sequence of events that take place when an exception/interrupt occurs in terms of saving and restoring program state and transfer of control. (HSI)

f. Translate expressions and assignment statements from C to assembly language. (L)

g. Translate Boolean logic and control flow constructs (decisions, loops) from C to assembly language. (L)

h. Translate operations on arrays and pointers from C to assembly language. (L)

i. Translate simple C functions to assembly language procedures. (L)

j. Compute average memory access time under various cache and memory configurations. (A)

k. Compose, compile/assemble, execute, and debug C programs using appropriate modularization and multiple files. (T)

Level 3: Synthesis and Evaluation

Level 3 outcomes are those in which the student can apply the material in new situations. This is the highest level of mastery. Upon successful completion of this course, students will be able to:

a. **Analyze** the performance of alternative algorithms in the context of the cache-memory hierarchy (e.g., open vs. closed hashing). (A)

b. **Analyze** and evaluate key components of a processor's datapath and control unit by analyzing their interactions and assessing their impact on instruction execution efficiency. (A)

c. **Analyze** a sequence of instructions for data, control, and/or structural hazards, and characterize potential performance implications of the hazards. (A)

d. **Design** and implement a program that uses low-level C language features (e.g., pointers, bit operations) to implement a simple module in an ISA toolchain (e.g., tokenizer, parser, code generator, disassembler). (HSI/T)

ABET STUDENT OUTCOMES MAPPING

Course outcomes	Student Outcome
2a, 2b, 2c, 2j	1
3a, 3b, 3c	2 (ABET 1)

3d	3 (ABET 2)
None	4 (ABET 5)
None	5 (ABET 4)
None	6 (ABET 3)
None	7
None	8
2k	9
3d	10 (ABET 6)

PREREQUISITES BY TOPIC

(CS 2302 w/C or better AND EE 2169 w/C or better AND EE 2369 w/C or better AND MATH 2300 w/C or better) OR (CS 2401 w/B or better AND EE 2169 w/B or better AND EE 2369 w/B or better AND MATH 2300 w/B or better)