COURSE SYLLABUS  
CSC10009 – COMPUTER SYSTEM

# GENERAL INFORMATION

|  |  |
| --- | --- |
| Course name: | Computer System |
| Course name (in Vietnamese): | Hệ thống máy tính |
| Course ID: | CSC10009 |
| Knowledge block: | Basic professional knowledge |
| Number of credits: | 2 |
| Credit hours for theory: | 30 |
| Credit hours for practice: | 0 |
| Credit hours for self-study: | Unlimited |
| Prerequisite: |  |
| Prior course: | Introduction of Information Technology, Discrete Mathematics, Introduction of Programming |
| Instructors: |  |

# COURSE DESCRIPTION

This course covers computer system hardware organization and the factors influencing the design of hardware and software elements of a computer system, assembly language programming. The programmer interface with the goal of improving students’ abilities to reason about the execution of their programs, enhance the performance of their program. After this course, the student should be able to understand what happens inside a computer when it executes a program, how the software and hardware interact.

# COURSE GOALS

At the end of the course, students are able to

|  |  |  |
| --- | --- | --- |
| **ID** | **Description** | **Program LOs** |
| G1 | Use the specialized English terminology about information technology, especially relating to computer system. | 2.4.3, 2.4.5 |
| G2 | Describe the basic organization and operation of a computer, including the processor, storage hierarchy, and input/output architecture. | 1.3.3 |
| G3 | Explain the binary representation and operations of integers, floating point numbers and some other types of data on computers. | 1.1.1, 1.2.2 |
| G4 | Construct and illustrate digital circuits at the logic level (logic gates) using simulation tools. | 1.2.1, 1.3.3, 1.3.7, 2.1.1, 2.2.1, 2.2.4, 2.3.2 |
| G5 | Simulate and analyze the process of instruction execution on logic circuit of LEGv8 CPU. Then, explaining the difference in design perspective between RISC and CISC instruction set architectures in general, between LEGv8/ARMv8 and x86 architectures in particular. | 1.3.7, 1.1.3, 1.3.3, 5.1.1, 5.2.2 |

# COURSE OUTCOMES

|  |  |  |
| --- | --- | --- |
| **CO** | **Description** | **I/T/U** |
| G1.1 | Use specialized English terminology. | T |
| G1.2 | Explain English materials related to lectures. | U |
| G2.1 | Describe the structure and operating principles of personal computer components, significant CPU functions, input/output devices, and memory system (especially cache memory). | I, T |
| G3.1 | Explain binary representation of integers (unsigned, sign-magnitude, one's complement, two's complement, biased), floating-point numbers (IEEE standards) and representation principles of some other data types such as characters, colors,… | I, T |

|  |  |  |
| --- | --- | --- |
| G3.2 | Explain and perform basic binary operations on unsigned, two's complement integers and IEEE floating-point numbers. | I, T |

|  |  |  |
| --- | --- | --- |
| G4.1 | Design, construct and experiment by simulation software some basic combinational logic circuits and describe the basic operation of sequential logic circuits. | T, U |
| G5.1 | Explain the LEGv8 architecture (a simple subset of the ARMv8 AArch64 architecture) design point of view. | I, T |
| G5.2 | Translate back and forth between programming language (basic arithmetic/logic statements, control flow, procedures, stacks, etc.) and LEGv8 assembly language / machine code. Then, describe the process of translating and executing the program on the computer. | I, T, U |
| G5.3 | Simulate and explain the process of executing LEGv8 instructions on its CPU logic circuit. | I, T |
| G5.4 | Differentiate design perspectives between LEGv8/ARMv8 and x86 as well as between RISC and CISC architectures. | I, T |

# TEACHING PLAN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Topic** | **Course outcomes** | **Teaching/Learning Activities (samples)** | **Assessments** |
| 1 | Overview of computer systems. | G1.1, G1.2,  G2.1 | **Prepare:**   * Self-studying the lectures.   **Activities:**   * Discussion, Q&A. | * FE |
| 2 | Data representation on computer. | G1.1, G1.2, G3.1, G3.2 | **Prepare:**   * Self-studying the lectures (videos).   **Activities:**   * Discussion, Q&A. * Exercises. | * TE#1 * FE |
| 3 | Microprocessor’s organization and operation. | G1.1, G1.2, G2.1 | **Prepare:**   * Self-studying the lectures.   **Activities:**   * Discussion, Q&A. | * FE |
| 4 | Digital logic circuits. | G1.1, G1.2, G4.1 | **Prepare:**   * Self-studying the lectures (videos).   **Activities:**   * Discussion, Q&A. * Simulate (combinational and sequential) logic circuits (using Logisim). * Exercises. | * TE#2 * FE |
| 5 | LEGv8 architecture. | G1.1, G1.2, G5.1, G5.2, G5.4 | **Prepare:**   * Self-studying the lectures, textbook.   **Activities:**   * Teaching * Case study * Demonstration of LEGv8 assembly program. * Discussion, Q&A. * Exercises. | * TE#3 * FE |
| 6 | Translating and running the program on computer. | G1.1, G1.2, G5.2 | **Prepare:**   * Self-studying the lectures, textbook.   **Activities:**   * Illustrating about static/dynamic linking. * Discussion, Q&A. | * FE |
| 7 | Basic implementation of LEGv8 CPU. | G1.1, G1.2, G5.1, G5.3 | **Prepare:**   * Self-studying the lectures, textbook about the principals/steps of designing and implementing a logic circuit of CPU based on its instruction set architecture.   **Activities:**   * Teaching. * Simulate the execution of LEGv8 instructions on its CPU logic circuit using simulation tools. * Case study. * Discussion, Q&A. * Exercises. | * FE |
| 8 | Compare architectures of x86 and LEGv8/ARMv8.  RISC vs CISC. | G1.1, G1.2, G5.4 | **Prepare:**   * Self-studying the lectures (videos).   **Activities:**   * Discussion, Q&A. | * FE |
| 9 | Memory Hierarchy. | G1.1, G1.2, G2.1 | **Prepare:**   * Self-studying the lectures (videos).   **Activities:**   * Discussion, Q&A. * Exercises. | * TE#4 * FE |
| 10 | I/O System. | G1.2, G1.2, G2.1 | **Prepare:**   * Reading documents.   **Activities:**   * Discussion, Q&A | * FE |
| 11 | Review | G2.1, G3.1, G3.2, G4.1, G5.1, G5.2, G5.3, G5.4 | **Activities:**   * Discussion, Q&A. * Exercises. |  |

# ASSESSMENTS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Topic** | **Description** | **Course outcomes** | **Ratio (%)** |
| **TE** | **Test** |  |  | **40%** |
| TE#1 | Digital data representation on the computer. | Conversion of numeral system (decimal, binary, hexadecimal).  Representation of integers and IEEE floating-point numbers.  Calculation on integers. | G1.1, G1.2, G3.1, G3.2 | 40 / (#TE) % |
| TE#2 | Design combinational logic circuits. | Simplify bool functions using boolean algebra or Karnaugh map.  Design circuits according to bool algebraic function or given true table or requirements. | G1.1, G1.2, G4.1 | 40 / (#TE) % |
| TE#3 | LEGv8 instruction set. | Read and understand a simple LEGv8 assembly language program.  Translate back and forth between simple C programming language and LEGv8 assembly language. | G1.1, G1.2, G5.1, G5.2, G5.4 | 40 / (#TE) % |
| TE#4 | Cache memory. | Organize cache in direct-mapping, fully associative, or set-associative.  Cache memory accessing. | G1.1, G1.2, G2.1 | 40 / (#TE) % |
| **Exam** | **Final Exam** | Quiz | G1.1, G1.2, G2.1, G3.1, G3.2, G4.1, G5.1, G5.2, G5.3, G5.4 | **60%** |

# RESOURCES

# Textbooks

1. David A. Patterson, John L. Hennessy. (2016). ***Computer Organization and Design ARM Edition: The Hardware Software Interface* (1st ed.).** Oxford: Morgan Kaufmann.
2. Randal E. Bryant, David R. O'Hallaron. (2016). ***Computer systems : a programmer's perspective* (3rd ed.)**. Boston, Mass. ; London: Pearson.

# References

1. Nguyễn Minh Tuấn , 2007***, Kiến trúc máy tính,*** ĐH KHTN TpHCM
2. *Paul A. Carter* ***,*** *2019,* ***PC Assembly Language***
3. *W. Stallings, Prentice Hall, 2018.* ***Computer Organization and Architecture: Design for performance (11th ed),*** NewYork: Pearson
4. Patterson, David A., author. Asanović, Krste, contributor. (2019). ***Computer Architecture: A Quantitative Approach* (6th ed.).** Cambridge, MA: Morgan Kaufmann

# Tools

1. Visual Studio .NET
2. Logisim
3. Graphical-Micro-Architecture-Simulator
4. Zoom
5. Kahoot
6. Slack

# Other Resources

1. [MOOC](https://youtube.com/playlist?list=PLvvXmh2iABsLRCucq8APMG6sh6b-mQ90e&si=wbE4i3a0YzzpaVnb)

# GENERAL REGULATIONS & POLICIES

* All students are responsible for reading and following strictly the regulations and policies of the school and university.
* Students who are absent for more than 3 theory sessions are not allowed to take the exams.
* For any kind of cheating and plagiarism, students will be graded 0 for the course. The incident is then submitted to the school and university for further review.
* Students are encouraged to form study groups to discuss the topics. However, individual work must be done and submitted on your own.
* Students prepare lessons, preview documents according to regulations.
* Students need to actively interact in online discussion environments.
* All online accounts must be registered by student email, using the student-ID and full name, the real avatar in online workspace.
* The number of assignments may vary depending on the classroom situation.