Subject Description Form

Subject Code	EE3007 / EE3007A
Subject Title	Computer System Principles
Credit Value	3
Level	3
Pre-requisite/ Co-requisite/ Exclusion	Pre-requisite: ENG2002
Objectives	1. To enable students to establish a broad knowledge of the organization of a computer system and internal architecture of a microprocessor 2. To enable students to understand software development for embedded systems 3. To enable students to utilize a microprocessor or microcontroller to solve engineering problems.
Subject Intended Learning Outcomes	Upon completion of the subject, students will be able to: a. Given the specifications of an application, design the software and hardware to carry out the necessary operations based on a microprocessor or a microcontroller. b. Understand advanced features of the latest microprocessors and understand the functions of basic computer peripherals. c. Understand the basic assembly language programming d. Think logically and be able to analyze data and present results in writing.
Subject Synopsis/ Indicative Syllabus	**Computer Systems Hardware and Operations** 1. ***Microprocessor operations and its internal architecture*****:** Operations of various registers, buses and data path, operations of ALU, arithmetic hardware, and general pipeline architecture. 2. ***Memory organization*****:** Characteristics of memory technologies. Memory hierarchies and memory decoding mechanism. 3. ***Input and output systems*****:** Direct I/O system and memory mapped I/O, interrupt and polling mechanisms. Typical serial data communication mechanisms. 4. ***Introduction to embedded computing systems*****:** System organization and design of input/output system. Programming software for embedded systems. 5. ***Introduction to assembly language programming*** **Laboratory Experiment:** Perform basic input/output operations of an embedded system by Python programming. Applications of different serial communication methods. Software development based on interrupt.
Teaching/Learning Methodology	Lectures and tutorials are the primary means of conveying the basic concepts and theories. Experiences on design, practical applications, and programming are achieved through experiments, in which the students are expected to solve design problems with real-life constraints and to attain feasible solutions with critical and analytical thinking. Interactive laboratory sessions are introduced to encourage better preparation and hence understanding of the experiments. On-the-spot assessments are conducted in the laboratory to provide additional incentives for student's learning. Experiments are designed to supplement the lecturing materials, especially in Python programming, so that the students are encouraged to take extra readings and to look for relevant information.

Teaching/Learning Methodology	Outcomes			
	a	b	c	d
Lectures	✓	✓	✓	
Tutorials	✓	✓	✓	
Experiments	✓		✓	✓

Assessment Methods in Alignment with Intended Learning Outcomes

Specific assessment methods/tasks	% weighting	Intended subject learning outcomes to be assessed			
		a	b	c	d
1. Examination	60%	✓	✓	✓	✓
2. Mid-term quiz	15%	✓	✓	✓	
3. Laboratory performance & report	15%	✓			✓
4. Online assignments and in-class activities	10%	✓		✓	✓
Total	100%				

It is a fundamental computer architecture subject. The outcomes on concepts, design and applications are assessed by the usual means of examination and quiz whilst those on analytical skills, problem-solving techniques and practical considerations of programming, as well as technical reporting, are evaluated via experiments and the report.

Student Study Effort Expected

Class contact:	
▪ Lecture/Tutorial	30 Hrs.
▪ Laboratory	9 Hrs.
Other student study effort:	
▪ Laboratory preparation/report	16 Hrs.
▪ Self-study	50 Hrs.
Total student study effort	105 Hrs.

Reading List and References

Reference books and online materials:

1. J.L. Hennessy and D.A. Patterson, Computer Architecture: A Quantitative Approach, 6th Edition, Elsevier, 2019
2. P. Darche, Microprocessor: Software and Hardware Aspects of Development, Debugging and Testing, John Wiley & Sons, 2021
3. A. Tanenbaum, T. Austin, Structured Computer Organization, Pearson India, 6th Edition, 2016.
4. A.K. Ray, Advanced Microprocessors & Peripherals, McGraw-Hill, 2006
5. A.B. Downey, Think Python: How to Think Like a Computer Scientist, 2nd ed., O'Reilly, 2015
6. S. Monk, Programming the Raspberry Pi Getting Started with Python, McGraw Hill, 2016
7. https://www.raspberrypi.org/documentation/usage/python/