VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI
B.E: Electronics & Communication Engineering / B.E: Electronics & Telecommunication Engineering
NEP, Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
(Effective from the academic year 2021 – 22)

V Semester

Computer Organization & ARM Microcontrollers			
Course Code	**21EC52**	CIE Marks	50
Teaching Hours/Week (L: T: P: S)	(3:0:2:0)	SEE Marks	50
Total Hours of Pedagogy	40 hours Theory + 13 Lab slots	Total Marks	100
Credits	04	Exam Hours	03

Course objectives: This course will enable students to:

1. Explain the basic organization of a computer system.
2. Demonstrate functioning of different sub systems, such as processor, Input/output, and memory.
3. Describe the architectural features and instructions of 32-bit microcontroller ARM Cortex M3.
4. Apply the knowledge gained for Programming ARM Cortex M3 for different applications.
5. Understand the basic hardware components and their selection method based on the characteristics and attributes of an embedded system.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teacher can use to accelerate the attainment of the various course outcomes.

- Lecture method (L) does not mean only traditional lecture method, but different type of teaching methods may be adopted to develop the outcomes.
- Encourage collaborative (Group) Learning in the class.
- Ask at least three HOTS (Higher order Thinking) questions in the class, which promotes critical thinking.
- Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyze information rather than simply recall it.
- Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.
- Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.
- Give Programming Assignments.

Module-1

Basic Structure of Computers: Basic Operational Concepts, Bus Structures, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement.
Text Book 1: Chapter 1 – 1.3, 1.4, 1.6 (1.6.1-1.6.4, 1.6.7), Chapter 2 – 2.2 to 2.10

Input/Output Organization: Accessing I/O Devices, Interrupts – Interrupt Hardware, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces – PCI Bus, SCSI Bus, USB.
Text Book 1: Chapter 4 – 4.1, 4.2, 4.4, 4.5, 4.6, 4.7

Teaching-Learning Process	Chalk and Talk, YouTube videos **RBT Level:** L1, L2, L3

Module-2

Memory System: Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed, Size, and Cost, Cache Memories – Mapping Functions, Replacement Algorithms, Performance Considerations.
Text book 1: Chapter 5 – 5.1 to 5.4, 5.5 (5.5.1, 5.5.2), 5.6

Basic Processing Unit: Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization, Hard-wired Control, Micro programmed Control. Basic concepts of pipelining,

Text book 1: Chapter7, Chapter 8 – 8.1

Teaching-Learning Process	Chalk and Talk, YouTube videos **RBT Level:** L1, L2, L3
Module-3	
ARM Embedded Systems: Introduction, RISC design philosophy, ARM design philosophy, Embedded system hardware – AMBA bus protocol, ARM bus technology, Memory, Peripherals, Embedded system software – Initialization (BOOT) code, Operating System, Applications. ARM Processor Fundamentals, ARM core dataflow model, registers, current program status register, Pipeline, Exceptions, Interrupts and Vector Table, Core extensions. Text book 2: Chapter 1, 2	
Teaching-Learning Process	Chalk and Talk, YouTube videos **RBT Level:** L1, L2, L3
Module-4	
Introduction to the ARM Instruction set: Introduction, Data processing instructions, Load - Store instruction, Software interrupt instructions, Program status register instructions, Loading constants, ARMv5E extensions, Conditional Execution. Text book 2: Chapter 3	
Teaching-Learning Process	Chalk and Talk, Power point presentations, Programming assignments **RBT Level:** L1, L2, L3
Module-5	
Introduction to the THUMB instruction set: Introduction, THUMB register usage, ARM – THUMB interworking, Other branch instructions, Data processing instructions, Stack instructions, Software interrupt instructions. **Efficient C Programming**: Overview of C Compilers and optimization, Basic C Data types, C looping structures. Text book 2: Chapter 4, 5	
Teaching-Learning Process	Chalk and Talk, Power point presentations, Programming assignments **RBT Level:** L1, L2, L3

PRACTICAL COMPONENT OF IPCC

Conduct the following experiments by writing Assembly Language Program (ALP) using ARM Cortex M3 Registers using an evaluation board/simulator and the required software tool.

Sl.No	Experiments
1	Write an ALP to i) multiply two 16-bit binary numbers. ii) add two 64-bit numbers.
2	Write an ALP to find the sum of first 10 integer numbers.
3	Write an ALP to find factorial of a number.
4	Write an ALP to add an array of 16-bit numbers and store the 32-bit result in internal RAM.
5	Write an ALP to find the square of a number (1 to 10) using look-up table.
6	Write an ALP to find the largest/smallest number in an array of 32 numbers.
7	Write an ALP to arrange a series of 32-bit numbers in ascending/descending order.
8	i) Write an ALP to count the number of ones and zeros in two consecutive memory locations. ii)Write an ALP to Scan a series of 32-bit numbers to find how many are negative.

Demonstration Experiments (For CIE only not for SEE) Conduct the following experiments on an ARM CORTEX M3 evaluation board using evaluation version of Embedded 'C' & Keil µvision-4 tool/compiler.	
9	Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction.
10	Interface a DAC and generate Triangular and Square waveforms.
11	Display the Hex digits 0 to F on a 7-segment LED interface, with a suitable delay in between.
12	Interface a simple Switch and display its status through Relay, Buzzer and LED.

Course Outcomes

At the end of the course the student will be able to:

1. Explain the basic organization of a computer system.
2. Demonstrate functioning of different sub systems, such as processor, Input/output, and memory.
3. Describe the architectural features and instructions of 32-bit microcontroller ARM Cortex M3.
4. Apply the knowledge gained for Programming ARM Cortex M3 for different applications.

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures not less than 35% (18 Marks out of 50) in the semester-end examination (SEE), and a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together

CIE for the theory component of IPCC

Two Tests each of **20 Marks (duration 01 hour**)

- First test at the end of 5th week of the semester
- Second test at the end of the 10th week of the semester

Two assignments each of **10 Marks**

- First assignment at the end of 4th week of the semester
- Second assignment at the end of 9th week of the semester

Scaled-down marks of two tests and two assignments added will be CIE marks for the theory component of IPCC for **30 marks**.

CIE for the practical component of IPCC

- On completion of every experiment/program in the laboratory, the students shall be evaluated and marks shall be awarded on the same day. The **15 marks** are for conducting the experiment and preparation of the laboratory record, the other **05 marks shall be for the test** conducted at the end of the semester.
- The CIE marks awarded in the case of the Practical component shall be based on the continuous evaluation of the laboratory report. Each experiment report can be evaluated for 10 marks. Marks of all experiments' write-ups are added and scaled down to 15 marks.
- The laboratory test **(duration 03 hours)** at the end of the 15th week of the semester /after completion of all the experiments (whichever is early) shall be conducted for 50 marks and scaled down to 05 marks**.**

Scaled-down marks of write-up evaluations and tests added will be CIE marks for the laboratory component of IPCC for **20 marks**.

SEE for IPCC

Theory SEE will be conducted by University as per the scheduled timetable, with common question papers for the course (duration 03 hours)

- The question paper will have ten questions. Each question is set for 20 marks.
- There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub-questions), **should have a mix of topics** under that module.
- The students have to answer 5 full questions, selecting one full question from each module.

The theory portion of the IPCC shall be for both CIE and SEE, whereas the practical portion will have a CIE component only. Questions mentioned in the SEE paper shall include questions from the practical component.

- The minimum marks to be secured in CIE to appear for SEE shall be the 12 (40% of maximum marks-30) in the theory component and 08 (40% of maximum marks -20) in the practical component. The laboratory component of the IPCC shall be for CIE only. However, in SEE, the questions from the laboratory component shall be included**.** The maximum of 04/05 questions to be set from the practical component of IPCC, the total marks of all questions should not be more than the 20 marks.

SEE will be conducted for 100 marks and students shall secure 35% of the maximum marks to qualify in the SEE. Marks secured will be scaled down to 50.

Suggested Learning Resources:

Textbooks

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer Organization, 5th Edition, Tata McGraw Hill, 2002. (Listed topics only from Chapters 1, 2, 4, 5, 8).
2. Andrew N Sloss, Dominic System and Chris Wright, “ARM System Developers Guide”, Elsevier, Morgan Kaufman publisher, 1st Edition, 2008.

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

Programming Assignments / Mini Projects can be given to improve programming skills