COURSEPACK (Fall 2025-26)

1. THE SCHEME

<table>
<tr><td>Course Title</td><td colspan="3">Computer Organization and Architecture</td><td colspan="2">Course Type</td><td colspan="4">Theory</td></tr>
<tr><td>Course Code</td><td colspan="3">R1UC305T</td><td colspan="2">Class</td><td colspan="4">B.Tech. III Sem Core and All specialization</td></tr>
<tr><td rowspan="6">Instruction delivery</td><td>Activity</td><td>Credits</td><td>Credit Hours</td><td colspan="4" rowspan="2">Total Number of Classes per Semester</td><td colspan="2" rowspan="2">Assessment in Weightage</td></tr>
<tr><td>Lecture</td><td>3</td><td>3</td></tr>
<tr><td>Tutorial</td><td>0</td><td>0</td><td rowspan="4">Theory</td><td rowspan="4">Tutorial</td><td rowspan="4">Practical</td><td rowspan="4">Self-study</td><td rowspan="4">CIE</td><td rowspan="4">SEE</td></tr>
<tr><td>Practical</td><td>0</td><td>0</td></tr>
<tr><td>Self-study</td><td>0</td><td>0</td></tr>
<tr><td>Total</td><td>3</td><td>3</td><td>40</td><td>0</td><td>0</td><td>0</td><td>50%</td><td>50%</td></tr>
<tr><td>Course Lead</td><td colspan="2">Dr. Sunil Kumar</td><td>Course Coordinator</td><td colspan="6">Dr. Savita Kumari</td></tr>
<tr><td rowspan="2">Names Course Instructors</td><td colspan="4">Theory</td><td colspan="5">Practical</td></tr>
<tr><td colspan="4">Dr. Brijesh Kumar Singh, Dr. D SALANGAI NAYAGI, Dr. Sandeep Mishra, Dr. Pragya Srivastava, Ms. Geeta Gayatri Behera, Dr. Gurmeet Singh, Dr. Manish Verma, Ms. Mullapudi Navyasri, Mr. P.SELVRAJ, Mr. Piyoush Kumar, Dr. Prakash Anand, Mr. Rikendra, Mr. Sandeep Bhatia, Dr. Tarun Maini, Ms. Trapti Shrivastava, Mr. Vimal Singh, Mr. Anshu Praveen</td><td colspan="5">NA</td></tr>
</table>

2. COURSE OVERVIEW

Computer Organization and Architecture is a foundational course in computer science and engineering that explores the internal structure and operational principles of computer systems. The course covers the key concepts and components that make up a computer system, including hardware design, instruction set architecture, memory hierarchy, input/output mechanisms, and performance optimization. Students will gain an understanding of how software interacts with hardware to perform computations and execute programs efficiently.

3. COURSE OBJECTIVES

This course aims to provide a comprehensive understanding of the fundamental principles of computer organization and architecture. Students will learn about the design and function of major computer system components, such as the CPU, memory, and I/O systems, and analyze system performance to explore optimization techniques. The course also emphasizes the interaction between hardware and software in executing instructions, offering practical experience with assembly language programming and hardware simulation tools. This knowledge equips students with the skills needed for advanced studies and careers in hardware and software development.

4. PREREQUISITE COURSE

PREREQUISITE COURSE REQUIRED	Yes	

If, yes please fill in the details

Prerequisite course code	Prerequisite course name
G2UC101B	Introduction to Digital Systems

5. PROGRAM OUTCOMES (POs):

PO No.	Description of the Program Outcome
PO1	Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.
PO2	Problem Analysis: Identify, formulate, review research literature and analyse complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4).
PO3	Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5).
PO4	Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).
PO5	Modern Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6).
PO6	The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).
PO7	Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9).
PO8	Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
PO9	Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences.
PO10	Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments..
PO11	Life-Long Learning: Recognize the need for, and have the preparation and ability for: i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8).

6. PROGRAM SPECIFIC OUTCOMES (PSOs):

Program Specific Outcomes (PSO) are statements that describe what the graduates of a discipline-specific program should be able to do. Two to Three PSOs per program should be designed.

PO No.	Description of the Program-Specific Outcome
PSO1	Have the ability to work with emerging technologies in Computer Science and Engineering requisite to Industry 4.0.
PSO2	Demonstrate Engineering Practice learned through industry internship and research project to solve live problems in various domains.

7. COURSE CONTENT (THEORY)

CONTENT (Syllabus)
THEORY: **Module-1 Register Transfer and Micro-operations** Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro-operations, Logic Micro-operations, Shift Micro-operations, Arithmetic logic shift unit **Module -2 Basic Computer Organizations and Design and CPU Organizations** Instruction Cycle, Memory-Reference Instructions, Register reference instructions, and Input - Output Instructions Central Processing Unit: General Register Organization, Stack Organization, Instruction Formats, and Addressing Modes **Module-3 Computer Arithmetic** Addition and Subtraction (Signed Magnitude and Signed 2's Complement) , Multiplication Algorithms (Binary Multiplication for Signed Magnitude and Booth's Algorithm) , Division Algorithms (Restoring and Non-Restoring). **Module-4 Memory Organization** Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory, Memory Management Hardware **Module-5 Input-Output Organization** Peripheral Devices, Input-Output Interfaces, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, and Direct Memory Access (DMA)

8. COURSE OUTCOMES (COs)

After the completion of the course, the student will be able to:

CO No.	Description of the Course Outcome
R1UC305T.1	To explain and classify the different types of register transfer and micro-operations.
R1UC305T.2	To analyze and compare different computer organizations and CPU components, including instruction formats and addressing modes.
R1UC305T.3	To apply various computer arithmetic algorithms to solve complex problems.
R1UC305T.4	To assess and design strategies for memory organization, including cache and virtual memory management.
R1UC305T.5	To evaluate and interpret the principles of input-output organization.

9. TAXONOMY LEVEL OF THE COURSE OUTCOMES

Mapping of COs with Bloom's Level

CO No.	Remember **KL1**	Understand **KL 2**	Apply **KL 3**	Analyse **KL 4**	Evaluate **KL 5**	Create **KL 6**
R1UC305T.1	√	√				
R1UC305T.2	√	√	√			
R1UC305T.3	√	√	√	√		
R1UC305T.4	√	√	√		√	√
R1UC305T.5	√	√			√	

10. COURSE ARTICULATION MATRIX

COs#/ POs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
R1UC305T.1	3	3	2	-	2	-	-	1	-	-	-	-	-
R1UC305T.2	3	3	2	-	-	2	-	-	-	1	-	-	-
R1UC305T.3	3	3	2	-	2	-	-	1	-	-	-	-	-
R1UC305T.4	3	3	3	3	3	-	1	-	1	-	-	-	-
R1UC305T.5	3	3	3	3	3	2	2	-	1	1	-	-	-

Note: 1-Low, 2-Medium, 3-High \ *first semester first course and first Course Outcome

11. TYPICAL EXAMPLE OF COURSES, CREDIT HOURS AND TEACHING HOURS

Type of Course	**Credits Hours**					**Hours of engagement/ Week**					**12 weeks/ semester**	**Remarks**
	Theory	**Tutorial**	**Practical**	**Self-study**	**Total**	**Theory**	**Tutorial**	**Practical**	**Self-study**	**Total**	**Total no. of classes**	
Theory Course	3	0	0	0	3	3	0	0	0	3	40	40 classes for theory

*1 credit = 3 self-learning hours (Not to mention in the lesson plan)

L. No	T/L	Topics	Tutorial / Practical Plan	Skills	Competency
1	L	Introduction of COA	Theory	Comprehension inter-register transfer using RTL and performing basic operations such as arithmetic, logic, and shifts.	CO1
2	L	Register Transfer Language	Theory		
3	L	Register Transfer	Theory		
4	L	Bus and Memory Transfers	Theory		
5	L	Arithmetic Micro-operations	Theory		
6	L	Logic Micro-operations	Theory		
7	L	Shift Micro-operations	Theory		
8	L	Arithmetic logic shift unit	Theory		
9	L	Arithmetic logic shift unit continued	Theory		
10	L	Instruction Cycle	Theory	Analysis and comparison of different types of computer organizations on the basis of instruction formats and addressing modes.	CO2
11	L	Memory-Reference Instructions	Theory		
12	L	Register reference instructions	Theory		
13	L	Input - Output Instructions	Theory		
14	L	General Register Organization	Theory		
15	L	Stack Organization	Theory		
16	L	Stack Organization Continued	Theory		
17	L	Instruction Formats	Theory		
18	L	Addressing Modes	Theory		
19	L	Addressing Modes Continued	Theory		
20	L	Addition and Subtraction (Signed Magnitude and Signed 2's Complement)	Theory	Application of computer arithmetic algorithms to solve problems	CO3
21	L	Multiplication Algorithms (Binary Multiplication for Signed Magnitude	Theory		
22	L	Booth's Algorithm	Theory		
23	L	Booth's Algorithm Continued	Theory		
24	L	Numerical Based on Booth's Algorithm	Theory		
25	L	Division Algorithms (Restoring and Non-Restoring)	Theory		
26	L	Numerical Based on Division Algorithms (Restoring and Non-Restoring) Continued	Theory		
27	L	Memory Hierarchy	Theory	Designing of various memory system for the computer for provided specifications and requirements	CO4
28	L	Main Memory	Theory		
29	L	Auxiliary Memory	Theory		
30	L	Associative Memory	Theory		
31	L	Cache Memory	Theory		
32	L	Virtual Memory	Theory		
33	L	Memory Management	Theory		

		Hardware			
34	L	Memory Management Hardware Continued	Theory		
35	L	Peripheral Devices	Theory	Evaluation of interpretation the principles of input-output organization	CO5
36	L	Input-Output Interfaces	Theory		
37	L	Asynchronous Data Transfer	Theory		
38	L	Modes of Transfer	Theory		
39	L	Priority Interrupt	Theory		
40	L	Direct Memory Access (DMA)	Theory		

12. BIBLIOGRAPHY

- **TextBook:**

1. Morris Mano, Computer System Architecture, 3rd Edition, Prentice-Hall of India Private Limited, 1999.

- ReferenceBooks:

1. Hayes, John P. Computer architecture and organization. McGraw-Hill, Inc., 2002.
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky Computer Organization, McGraw-Hill, Fifth Edition, Reprint 2012
3. William Stallings, Computer Organization and Architecture-Designing for Performance, Pearson Education, Seventh edition, 2006

- Webliography:

1. GeeksforGeeks COA Tutorial – covers pipelining, microprogrammed control, memory organization, I/O, etc.
2. Wikipedia: Computer Architecture – overview of ISA, design trade-offs, hardware-software abstraction.
3. Wikipedia: Instruction Set Architecture (ISA) – defines ISA and explains RISC vs. CISC, extensions.
4. Wikipedia: Microarchitecture – details microarchitectural design, pipelines, execution units, constraints.
5. Coursera – Computer Architecture course – free modules on ISA, microarchitecture, performance.
6. Wikipedia: Von Neumann Architecture – foundational model and its evolution in computing systems.
7. Stallings/Mano COA Problem PDFs – in-depth descriptions of memory, CPU, I/O frameworks.
8. GitHub curated COA resources – textbooks, RISC-V, conferences, tools for deeper learning.
9. ISCA & ACM SIGARCH – leading-edge research and trends in computer architecture.

SWAYAM/NPTEL/MOOCsCertification:

https://www.coursera.org/learn/comparch

https://nptel.ac.in/courses/106105163

https://www.geeksforgeeks.org/computer-organization-and-architecture-tutorials/

13. COURSE ASSESSMENT

Assessment forms an integral part of curriculum design. A learning-teaching system can only be effective if the student's learning is measured at various stages which means while the student processes learning (Assessment for Learning) a given content and after completely learning a defined content (Assessment of Learning). Assessment for learning is referred to as formative assessment, that is, an assessment designed to inform instruction.

The ability to use and apply the knowledge in different ways may not be the focus of the assessment. With regard to designing assessments, the faculty members must be willing to put in the time required to create a valid, reliable

assessment, that ideally would allow students to demonstrate their understanding of the information while remaining. The following are the five main areas that assessment reporting should cover.

1. **Learning Outcomes**: At the completion of a program, students are expected to know their knowledge, skills, and attitude. Depending on whether it is a UG or PG program, the level of sophistication may be different. There should be no strict rule on the number of outcomes to be achieved, but the list should be reasonable, and well-organized.
2. **Assessable Outcomes**: After a given learning activity, the statements should specify what students can do to demonstrate. Criteria for demonstration are usually addressed in rubrics and there should be specific examples of work that doesn't meet expectations, meets expectations, and exceeds expectations. One of the main challenges is faculty communication whether all faculty agreed on explicit criteria for assessing each outcome. This can be a difficult accomplishment when multiple sections of a course are taught or different faculty members. Hence there is a need for common understanding among the faculty on what is assessed and how it is assessed.
3. **Assessment Alignment**: This design of an assessment is sometimes in the form of a curriculum map, which can be created in something as easy as an Excel spreadsheet. Courses should be examined to see which program outcomes they support, and if the outcome is assessed within the course. After completion, program outcomes should be mapped to multiple courses within the program.
4. **Assessment Planning**: Faculty members need to have a specific plan in place for assessing each outcome. Outcomes don't need to be assessed every year, but faculty should plan to review the assessment data over a reasonable period of time and develop a course of action if the outcome is not being met.
5. **Student Experience**: Students in a program should be fully aware of the expectations of the program. The program outcomes are aligned on the syllabus so that students are aware of what course outcomes they are required to meet, and how the program outcomes are supported. Assessment documents should clearly communicate what is being done with the data results and how it is contributing to the improvement of the program and curriculum.
6. **Designing quality assessment tools** or tasks involves multiple considerations if it is to be fit for purpose. The set of assessments in a course should be planned to provide students with the opportunity to learn as they engage with formative tasks as well as the opportunity to demonstrate their learning through summative tasks. Encouraging the student through the use of realistic, authentic experiences is an exciting challenge for the course faculty team, who are responsible for the review and quality enhancements to assessment practices.

14. FORMATIVE AND SUMMATIVE ASSESSMENT

Assessment Pattern for Theory Course:

Type of Course (T)	CIE			Total Marks		Final Marks CIE*0.5+SEE*0.5
	IA1#	MTE	IA2#	CIE	SEE	
THEORY	25	50	25	100	100	100

#Typical Rubric for the Internal Assessments

Type of Assessment Tools	QUIZ	AAT$/MOOC Certifications
Internal Assessments	10	15
$AAT is Literature survey, Seminar, Assignment, Term Paper, Slip Test (or) MOOC Certificate relevant to the course		

15. PASSING STANDARDS

Passing Criteria for Different Course Types Effective from AY 2022-23 Onwards

S.No.	Course Type	Passing Criterion
1.	**Theory Course (T)**	A student shall secure a minimum of **30% of the maximum marks** in the semester-end examination (SEE/ETE) and **40% of aggregate marks** in the course including Continuous internal examination (CIE) and SEE/ETE marks. i.e., the minimum Passing Grade is "P".

Note: Students unable to meet the overall passing criteria as mentioned shall be eligible for the following options to clear the course:

- Appear in the Back Paper Examinations and have to meet the criteria to score 40% in marks overall
- Appear in summer examinations (Internal +External) to meet the criteria as mentioned.

16. PROBLEM-BASED LEARNING/CASE STUDIES/CLINICS

Exercises in Problem-based Learning (Assignments) (Min 54 Problems)

S.No.	Problem	KL
1	Represent 25H, 42H, and 2000H into binary numbers.	K1
2	Differentiate between `MOV BL, 25H` and `MOV BL, [25H]`.	K2
3	Identify addressing modes used in the following assembly program: `MOV BL, [2000H]`, `MOV CL, [3000H]`, `ADD BL, CL`, `ADD BL, 55H`, `MOV [4000], BL`.	K1
4	Determine address lines and data lines needed for the following memory units: (a) 2K × 16; (b) 64K × 8; (c) 16M × 32; (d) 4G × 64.	K1
5	Calculate number of 128 × 8 memory chips required to provide 4096 × 16 memory capacity.	K1
6	Show value of all bits of a 12-bit register holding decimal 215 in binary, BCD, and hexadecimal.	K2
7	Perform (+42) + (-13) and (-42) - (-13) in signed-2's complement binary representation.	K2
8	For a common bus system of 16 registers (32 bits each) built with multiplexers, determine: (a) selection inputs, (b) multiplexer size, (c) total multiplexers.	K2
9	Given initial values of registers AR, BR, CR, DR, determine values after a sequence of micro-operations.	K3
10	For a CPU with instruction word format containing indirect bit, opcode, register code, and address, determine bit allocation and draw the format.	K3
.11	Identify the next instruction to be fetched and executed given PC, AC, memory contents, and instruction codes.	K1
12	Explain the difference between hardwired control and microprogrammed control; can hardwired control have control memory?	K1
13	Specify instruction format and field sizes for a CPU with multiple addressing modes and registers.	K2
14	Calculate effective addresses for direct, immediate, relative, register indirect, and indexed addressing.	K2
15	Provide five examples each of internal and external interrupts.	K2

16	Differentiate between software interrupts and subroutine calls.	K2
17	Write assembly programs for X = (A + B) × (C + D) using three-, two-, one-, and zero-address instructions.	K3
18	Show step-by-step execution of Booth's multiplication for given signed numbers (+15 × +13) and (+15 × -13).	K3
19	Perform binary division using restoring and non-restoring algorithms for given numbers.	K3
20	Analyze given CPU instruction execution cycle and list register contents after execution.	K3
21	Formulate fetch-execute procedure for CPU with two instructions packed in one memory word.	K3
22	Determine computation time for a weather forecasting task requiring 250 billion flops on a 100 MFLOPS machine.	K3
23	Design pipeline configuration for (Ai + Bi) × (Ci + Di) and list register contents for i = 1 to 6.	K3
24	Draw space-time diagram for 6-segment pipeline processing 8 tasks.	K3
25	Calculate clock cycles required to process 200 tasks in a 6-segment pipeline.	K4
26	Compare daisy-chaining and parallel priority interrupts with diagrams.	K3
27	Draw instruction word format for CPU with given number of registers, opcodes, and addressing modes.	K5
28	Calculate number of chips required to provide 1024 bytes and 16 KB memory using 1024 × 1 RAM chips.	K6
29	For 1200-baud line, calculate characters per second for synchronous and asynchronous modes with different stop bits.	K6
30	Calculate address lines and decoder size for memory capacity of 2048 bytes using 128 × 8 RAM chips.	K6
31	Formulate logical and physical address formats for segmented-paged memory system.	K1
32	For a direct-mapped cache of 32 KB, block size 32 bytes, CPU generating 32-bit addresses, calculate index and tag bits.	K1
33	Design main memory and cache organization for given hit ratio and access time constraints.	K2
34	Explain working of virtual memory with example of page table translation.	K2
35	Analyze impact of cache block size on hit ratio and memory bandwidth.	K2
36	Determine average memory access time given cache hit ratio and memory access times.	K2
37	Develop an associative cache mapping strategy for given constraints.	K3
38	Create control unit design using hardwired approach for specified instruction set.	K3

39	Create microprogram control unit design for specified instruction set.	K3
40	Write micro-operations for implementing an instruction cycle.	K3
41	Simulate CPU execution for given instruction set and pipeline hazards.	K3
42	Identify and resolve structural, data, and control hazards in pipeline execution.	K3
43	Evaluate trade-offs between Harvard and Von Neumann architectures.	K4
44	Compare RISC and CISC architectures in terms of performance, complexity, and application.	K4
45	Assess advantages and disadvantages of pipelining in CPU design.	K5
46	Evaluate effect of increasing number of pipeline stages on performance and complexity.	K5
47	Design bus arbitration logic for a multiprocessor system.	K6
48	Develop DMA controller logic design for a given system specification.	K6
49	Propose enhancements to CPU architecture to improve performance for a given workload.	K6
50	Design memory management unit (MMU) for given paging and segmentation requirements.	K6
51	Create I/O subsystem design for high-speed data acquisition system.	K6
52	Propose optimized instruction set for an embedded application.	K6
53	Design instruction pipeline stages for superscalar processor.	K6
54	Develop power-efficient CPU architecture for mobile devices.	K6

17. STUDENT-CENTERED LEARNING (SELF-LEARNING TOWARDS LIFE-LONG LEARNING)

Self-Learning, self-doing, and application of the knowledge acquired through the course after gaining adequate knowledge

It's a typical course-based project to be carried out by a whole class in groups of four students each; they should exhibit higher level Knowledge Levels (Bloom's Revised Taxonomy). To enhance their skill set in the integrated course, the students are advised to execute course-based **Design projects**.

The students, in a group not exceeding 4, are expected to conceive an idea based on the content (objectives/ outcomes) and apply the suitable knowledge to demonstrate their ability to learn.

A list of 30-40 project statements can be offered to the students to choose or develop their own ideas (teamwork) to define a problem statement, design and develop a product/ process/service/application, and provide a suitable solution (design thinking). They may also upload this Idea on the Yukti Portal (contact the University IIC Team) and also patent the same.