|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | 18CSC203J | **Course Name** | COMPUTER ORGANIZATION AND ARCHITECTURE | **Course Category** | *C* | *Professional Core* | L | T | P | C |
| 3 | 0 | 2 | 4 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Pre-requisite Courses** | *Nil* | | **Co-requisite Courses** | *Nil* | | **Progressive Courses** | *18CSC207J* |
| **Course Offering Department** | | *Computer Science and Engineering* | | | **Data Book / Codes/Standards** | *Nil* | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Learning Rationale (CLR):** | | | *The purpose of learning this course is to:* | |  | **Learning** | | |  | **Program Learning Outcomes (PLO)** | | | | | | | | | | | | | | |
|  |  |
| **CLR-1 :** | *Utilize the functional units of a computer* | | | |  | 1 | 2 | 3 |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| **CLR-2 :** | *Analyze the functions of arithmetic Units like adders, multipliers etc.* | | | |  | Level of Thinking (Bloom) | Expected Proficiency (%) | Expected Attainment (%) |  | Engineering Knowledge | Problem Analysis | Design & Development | Analysis, Design, Research | Modern Tool Usage | Society & Culture | Environment & Sustainability | Ethics | Individual & Team Work | Communication | Project Mgt. & Finance | Life Long Learning | PSO - 1 | PSO - 2 | PSO – 3 |
| **CLR-3 :** | *Understand the concepts of Pipelining and basic processing units* | | | |  |  |
| **CLR-4 :** | *Study about parallel processing and performance considerations.* | | | |  |  |
| **CLR-5 :** | *Have a detailed study on Input-Output organization and Memory Systems.* | | | |  |  |
| **CLR-6 :** | *Simulate simple fundamental units like half adder, full adder etc* | | | |  |  |
|  | |  | | |  |  |
| **Course Learning Outcomes (CLO):** | | | | *At the end of this course, learners will be able to:* | |  |
| **CLO-1 :** | *Identify the computer hardware and how software interacts with computer hardware* | | | | | *2* | *80* | *70* |  | *H* | *H* | *-* | *-* | *-* | *-* | *-* | *-* | *M* | *L* | *-* | *M* | *-* | *-* | *-* |
| **CLO-2 :** | *Apply Boolean algebra as related to designing computer logic, through simple combinational and sequential logic circuits* | | | | | *3* | *85* | *75* |  | *H* | *H* | *H* | *-* | *H* | *-* | *-* | *-* | *M* | *L* | *-* | *M* | *-* | *-* | *-* |
| **CLO-3 :** | *Analyze the detailed operation of Basic Processing units and the performance of Pipelining* | | | | | *2* | *75* | *70* |  | *H* | *H* | *H* | *H* | *-* | *-* | *-* | *-* | *M* | *L* | *-* | *M* | *-* | *-* | *-* |
| **CLO-4 :** | *Analyze concepts of parallelism and multi-core processors.* | | | | | *3* | *85* | *80* |  | *H* | *-* | *-* | *H* | *-* | *-* | *-* | *-* | *M* | *L* | *-* | *M* | *-* | *-* | *-* |
| **CLO-5 :** | *Identify the memory technologies, input-output systems and evaluate the performance of memory system* | | | | | *3* | *85* | *75* |  | *H* | *-* | *H* | *H* | *-* | *-* | *-* | *-* | *M* | *L* | *-* | *M* | *-* | *-* | *-* |
| **CLO-6 :** | *Identify the computer hardware, software and its interactions* | | | | | *3* | *85* | *75* |  | *H* | *H* | *H* | *H* | *H* | *-* | *-* | *-* | *M* | *L* | *-* | *M* | *-* | *-* | *-* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Duration (hour) | | 15 | 15 | 15 | 15 | 15 |
| S-1 | SLO-1 | *Functional Units of a computer* | *Addition and subtraction of Signed numbers* | *Fundamental concepts of basic processing unit* | *Parallelism* | *Memory systems -Basic Concepts* |
| SLO-2 | *Operational concepts* | *Problem solving* | *Performing ALU operation* | *Need, types of Parallelism* | *Memory hierarchy* |
| S-2 | SLO-1 | *Bus structures* | *Design of fast adders* | *Execution of complete instruction, Branch instruction* | *applications of Parallelism* | *Memory technologies* |
| SLO-2 | *Memory locations and addresses* | *Ripple carry adder and Carry look ahead adder* | *Multiple bus organization* | *Parallelism in Software* | *RAM, Semiconductor RAM* |
| S-3 | SLO-1 | *Memory operations* | *Multiplication of positive numbers* | *Hardwired control* | *Instruction level parallelism* | *ROM,Types* |
| SLO-2 | *Memory operations* | *Problem Solving* | *Generation of control signals* | *Data level parallelism* | *Speed,size cost* |
| S  4-5 | SLO-1 | *Lab 1: To recognize various components of PC- Input Output systems*  *Processing and Memory units* | *Lab4:Study of TASM*  *Addition and Subtraction of 8-bit number* | *Lab-7: Design of Half Adder*  *Design of Full Adder* | *Lab-10: Study of Array Multiplier*  *Design of Array Multiplier* | *Lab-13: Study of Carry Save Multiplication*  *Program to carry out Carry Save Multiplication* |
| SLO-2 |
| S-6 | SLO-1 | *Instructions, Instruction sequencing* | *Signed operand multiplication* | *Micro-programmed control-* | *Challenges in parallel processing* | *Cache memory* |
| SLO-2 | *Addressing modes* | *Problem solving* | *Microinstruction* | *Architectures of Parallel Systems - Flynn’s classification* | *Mapping Functions* |
| S-7 | SLO-1 | *Problem solving* | *Fast multiplication- Bit pair recoding of Multipliers* | *Micro-program Sequencing* | *SISD,SIMD* | *Replacement Algorithms* |
| SLO-2 | *Introduction to Microprocessor* | *Problem Solving* | *Micro instruction with Next address field* | *MIMD, MISD* | *Problem Solving* |
| S-8 | SLO-1 | *Introduction to Assembly language* | *Carry Save Addition of summands* | *Basic concepts of pipelining* | *Hardware multithreading* | *Virtual Memory* |
| SLO-2 | *Writing of assembly language programming* | *Problem Solving* | *Pipeline Performance* | *Coarse Grain parallelism, Fine Grain parallelism* | *Performance considerations of various memories* |
| S  9-10 | SLO-1 | *Lab-2:To understand how different components of PC are connected to work properly*  *Assembling of System Components* | *Lab 5: Addition of 16-bit number*  *Subtraction of 16-bit number* | *Lab-8: Study of Ripple Carry Adder*  *Design of Ripple Carry Adder* | *Lab-11: Study of Booth Algorithm* | *Lab-14: Understanding Processing unit*  *Design of primitive processing unit* |
| SLO-2 |
| S-11 | SLO-1 | *ARM Processor: The thumb instruction set* | *Integer division – Restoring Division* | *Pipeline Hazards-Data hazards* | *Uni-processor and Multiprocessors* | *Input Output Organization* |
| SLO-2 | *Processor and CPU cores* | *Solving Problems* | *Methods to overcome Data hazards* | *Multi-core processors* | *Need for Input output devices* |
| S-12 | SLO-1 | *Instruction Encoding format* | *Non Restoring Division* | *Instruction Hazards* | *Multi-core processors* | *Memory mapped IO* |
| SLO-2 | *Memory load and Store instruction in ARM* | *Solving Problems* | *Hazards on conditional and Unconditional Branching* | *Memory in Multiprocessor Systems* | *Program controlled IO* |
| S-13 | SLO-1 | *Basics of IO operations.* | *Floating point numbers and operations* | *Control hazards* | *Cache Coherency in Multiprocessor Systems* | *Interrupts-Hardware, Enabling and Disabling Interrupts* |
| SLO-2 | *Basics of IO operations.* | *Solving Problems* | *Influence of hazards on instruction sets* | *MESI protocol for Multiprocessor Systems* | *Handling multiple Devices* |
| S  14-15 | SLO-1 | *Lab -3To understand how different components of PC are connected to work properly*  *Disassembling of System Components* | *Lab-6: Multiplication of 8-bit number*  *Factorial of a given number* | *Lab-9: Study of Carry Look-ahead Adder*  *Design of Carry Look-ahead Adder* | *Lab-12: Program to carry out Booth Algorithm* | *Lab-15: Understanding Pipeline concepts*  *Design of basic pipeline.* |
| SLO-2 |

|  |  |  |
| --- | --- | --- |
| **Learning**  **Resources** | 1. *Carl Hamacher, ZvonkoVranesic, SafwatZaky, Computer Organization, 5th ed., McGraw-Hill, 2015* 2. *Kai Hwang, Faye A. Briggs, Computer Architecture and Parallel Processing”, 3rd ed., McGraw Hill, 2016* 3. *Ghosh T. K., Computer Organization and Architecture, 3rd ed., Tata McGraw-Hill, 2011* 4. *P. Hayes, Computer Architecture and Organization, 3rd ed., McGraw Hill, 2015.* | 1. *William Stallings, Computer Organization and Architecture – Designing for Performance, 10th ed., Pearson Education, 2015* 2. *David A. Patterson and John L. Hennessy Computer Organization and Design - A Hardware software interface, 5th ed., Morgan Kaufmann,2014* |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Learning Assessment** | | | | | | | | | | | |
|  | Bloom’s  Level of Thinking | Continuous Learning Assessment (50% weightage) | | | | | | | | Final Examination (50% weightage) | |
| CLA – 1 (10%) | | CLA – 2 (15%) | | CLA – 3 (15%) | | CLA – 4 (10%)# | |
| Theory | Practice | Theory | Practice | Theory | Practice | Theory | Practice | Theory | Practice |
| Level 1 | Remember | *20%* | *20%* | *15%* | *15%* | *15%* | *15%* | *15%* | *15%* | *15%* | *15%* |
| Understand |
| Level 2 | Apply | *20%* | *20%* | *20%* | *20%* | *20%* | *20%* | *20%* | *20%* | *20%* | *20%* |
| Analyze |
| Level 3 | Evaluate | *10%* | *10%* | *15%* | *15%* | *15%* | *15%* | *15%* | *15%* | *15%* | *15%* |
| Create |
|  | Total | 100 % | | 100 % | | 100 % | | 100 % | | - | |

# CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc.,

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Designers** |  |  |  |  |  |  | |
| Experts from Industry | | Experts from Higher Technical Institutions | | | | | Internal Experts |
| ***1.*** *T. V. Sankar, HCL Technologies Ltd, Chennai, sankar\_t@hcl.com* | | *1. Prof. A.P. Shanthi, ANNA University Chennai, a.p.shanthi@cs.annauniv.edu* | | | | | *1.Dr. V. Ganapathy, SRMIST* |
|  | |  | | | | | *2. Dr. C. Malathy, SRMIST* |
|  | |  | | | | | *3. Mrs M.S.Abirami, SRMIST* |