Kadi Sarva Vishwavidyalaya

Faculty of Engineering & Technology

Second Year Master of Engineering (Computer Engineering)

(Semester-III)

(With effect from: Academic Year 2018-19)

Subject Code: MECE-303-N-E	**Subject Title:** Advanced Computer Architecture
Pre-requisite	

Teaching Scheme (Credits and Hours)

Teaching Scheme				**Total Credit**	**Evaluation Scheme**					
L	**T**	**P**	**Total**		**Theory**		**Mid Sem Exam**	**CIA**	**Practical**	**Total**
Hours	**Hours**	**Hours**	**Hours**		**Hours**	**Marks**	**Marks**	**Marks**	**Marks**	**Marks**
04	00	02	06	05	03	70	30	20	30	150

Learning Objectives:

- An instruction set architecture (ISA) is the interface between the computer's software and hardware and also can be viewed as the programmer's view of the machine. Computers do not understand high-level programming languages such as Java, C++, or most programming languages used.

Outline of the Course:

Sr. No	**Title of the Unit**	**Minimum Hours**
1	Fundamentals of digital system and Review	07
2	Linear Pipeline processor	08
3	Storage and memory hierarchy	07
4	Instruction level parallelism	06
5	Parallel computer models and program parallelism	10
6	Vector Processor and synchronous parallel processing	10
7	System Interconnect architecture	10
8	Multiprocessor architecture and programming	06
	Total	**64**

Total hours (Theory): 64
Total hours (Lab): 32
Total hours: 96

Detailed Syllabus:

Sr. No	Topic	Lecture Hours	Weightage (%)
1	**Introduction and review**: Fundamentals of digital system and Review	07	11
2	**Pipelining:** Linear Pipeline processor: Nonlinear pipeline processor, instruction pipeline design, Mechanisms for instruction, pipelining, dynamic instruction scheduling, Branch handling techniques, arithmetic pipelining design: Computer arithmetic principles, static arithmetic pipelines, multifunction arithmetic pipelines	08	12
3	**Storage and memory hierarchy:** Register file, Virtual file, Cache memories, cache memory working principles, cache coherence issues, cache performance analysis.	07	11
4	**Instruction level parallelism:** Super-scalar processors	06	09
5	**Parallel computer models and program parallelism:** Classification of machines, SISD, SIMD and MIMD Conditions of parallelism, data and resource dependencies, hardware and software parallelism, program partitioning and scheduling, grain size latency, program flow mechanism, control flow versus data flow, data flow architecture, demand driven mechanisms, comparison of flow mechanisms	10	16
6	**Vector Processor and synchronous parallel processing:** Vector instruction types, vector-access memory schemes vector and symbolic processors SIMD architecture and programming principles: SIMD parallel algorithms, SIMD computers and performance enhancement	10	16
7	**System Interconnect architecture:** Network properties and routing, static interconnection networks, Dynamic interconnection networks multiprocessor system interconnects: Hierarchical bus system, crossbar switch and multi-port memory, multistage and combining network	10	16
8	**Multiprocessor architecture and programming** Functional structure, Interconnection network, Parallel memory organization, Multiprocessor operating system, Exploiting concurrency for multiprocessor	06	09
	Total	64	100

Instructional Method and Pedagogy:

- At the start of course, the course delivery pattern, prerequisite of the subject will be discussed.
- Lectures will be conducted with the aid of multi-media projector, black board, OHP etc.
- Attendance is compulsory in lecture and laboratory which carries 10 marks in overall evaluation.
- One internal exam will be conducted as a part of internal theory evaluation.
- Assignments based on the course content will be given to the students for each unit and will be evaluated at regular interval evaluation.
- Surprise tests/Quizzes/Seminar/tutorial will be conducted having a share of five marks in the overall internal evaluation.
- The course includes a laboratory, where students have an opportunity to build an appreciation for the concepts being taught in lectures.
- Experiments shall be performed in the laboratory related to course contents.

Learning Outcome:

On successful completion of this course, the student should be able to:

- understand and relate the need of parallel computer architecture.
- realize the scope for pipelining and vectorization in parallelism
- design the memory hierarchy for parallel architecture
- evaluate the parallel architecture models

Reference Books:

1. Hennessey & D.A. Patterson, “Computer architecture: A quantitative approach”, International student edition, 3rd edition, 2002, Morgan kaufmaan publisher.
2. Michael J. Flynn,” Computer Architecture: Pipelined and parallel processor design”, 1995, Jones and barlett, Boston.
3. Kai Hawang and Faye A. Briggs, “ Computer architect ure and parallel processing”, International edition, 1993, TMH
4. R.K. Ghose, RajanMoona&Phalfui Gupta, “Foundatio n of parallel processing”; Narosa publication
5. D. Sima, T. Fountain, P. Kasuk, “ Advanced computer architecture – A design space approach”, 1997, Addison Wesley
6. V. Rajaraman, Siva Ram Murthy; “Parallel Computers: Architecture and Programming”; PHI Publiciation. 200
7. Kai Hawang;” Advanced Computer Architecture: Parallelism, Scalability, Programmability” ; TMH; Internaational Edition 1993
8. Denial Tabak; “Advanced Microprocessors”
9. M. Morris Mano, “Computer System Architecture”, Third Edition; PHI Publication, 1993

List of experiments:

Sr. No.	Name of Experiment
1	Implement Booth multiplication algorithm. (Fixed point)
2	Implement addition and subtraction with Signed magnitude data. (Fixed point)
3	Implement Division algorithm for fixed point binary numbers. (Fixed point)
4	Implement Booth multiplication algorithm. (Floating Point)
5	Implement addition and subtraction with Signed magnitude data. (Floating Point)
6	Implement Division algorithm for fixed point binary numbers. (Floating Point)
7	Simulate integer addition in WinDLX.
8	Simulate floating point multiplication in WinDLX.
9	Simulate interger number to floating point number and perform arithmetic operation in WinDLX
10	Simulate looping operation in WinDLX