EVALUATION METHOD TO BE USED:

Continuous assessment	Mid term	End Semester
15(T) + 25 (P)	20	40

CO - PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								
CO2	✓	✓		✓	✓							
CO3	✓		✓	✓	✓							
CO4	✓	✓		✓								
CO5	✓	✓			✓							
CO6	✓	✓	✓	✓								

CS 6107 COMPUTER ARCHITECTURE

Prerequisites for the course: None

OBJECTIVES:

- To identify the requirements of different types of computer systems
- To understand the evaluation of computer systems based on various performance metrics
- To study the characteristics of the ISA and the hardware software co-design
- To trace the execution sequence of an instruction through the processor
- To compare different approaches used for implementing a functional unit
- To understand the fundamentals of memory and I/O systems and their interaction with the processor

COMPUTER ARCHITECTURE	L	T	P	EL	CREDITS
	3	0	2	3	5

MODULE I :	L	T	P	EL
	3	0	2	3

Introduction - Classes of computer systems - Performance - Amdahl's law - The Power wall - Switch from uniprocessors to multiprocessors – Benchmarks.

SUGGESTED ACTIVITIES :

- In Class activity for performance evaluation
- EL - Evolution of computer systems, identification of benchmarks
- Practical – Demonstration - Opening up a computer system and studying the components

SUGGESTED EVALUATION METHODS:

- Assignment problems
- Quizzes

MODULE II :	L	T	P	EL
	3	0	2	3

Hardware Software Interface - ISA - Operations of the computer hardware - Operands - Representing instructions - Instructions for making decisions - Supporting procedures in computer hardware.

SUGGESTED ACTIVITIES :

- Flipped classroom and activity
- EL – Writing simple assembly language programs from high level code
- Practical – Study of an existing standard architectural simulator

SUGGESTED EVALUATION METHODS:

- Assignment problems
- Quizzes

MODULE III :	L	T	P	EL
	3	0	2	3

Addressing modes - Translating and starting a program - Arrays versus pointers - MIPS instruction formats - Assembly language programming.

SUGGESTED ACTIVITIES :

- EL - Familiarising with assembly language programming
- Practical - Study of an existing standard architectural simulator

SUGGESTED EVALUATION METHODS:

- Assignment problems
- Quizzes

MODULE IV :	L	T	P	EL
	3	0	2	3

Integer arithmetic - Binary Parallel adder – Carry Look-ahead Adder - Carry save adder - Binary multiplier - Booth's multiplier - Bit-pair recoding - Binary division.

SUGGESTED ACTIVITIES :

- Flipped Class room
- Some arithmetic algorithms in class and some as EL
- Practical : Study of addressing modes with examples, Tracing the execution sequences, Identifying the timing constraints

SUGGESTED EVALUATION METHODS:

- Assignment problems
- Quizzes

MODULE V :	L	T	P	EL
	3	0	2	3

Floating point arithmetic- Representation - Arithmetic operations on floating point numbers - Parallelism and computer arithmetic.

SUGGESTED ACTIVITIES :

- Flipped class room
- EL – Simulation of the floating point operations
- Practical - Study of the ISA supported by the architectural simulator and running simple programs on the simulator

SUGGESTED EVALUATION METHODS:

- Assignment problems
- Quizzes
- Demonstrate decode and execute for a subset of instructions on the simulator

MODULE VI:	**L**	**T**	**P**	**EL**
	3	**0**	**2**	**3**

Datapath design - Implementation of the basic MIPS ISA - Building the datapath - A simple implementation scheme - Drawbacks.

SUGGESTED ACTIVITIES :

- Introduction in class
- Flipped Classroom for building of datapath for additional instructions
- Practical - Analysing the datapath on the standard simulator

SUGGESTED EVALUATION METHODS:

- Assignment problems
- Quiz in Class or automatic Quizzes for the flipped classroom content

MODULE VII:	**L**	**T**	**P**	**EL**
	6	**0**	**2**	**6**

Instruction Level Parallelism - Pipelining - Overview of pipelining - Performance - Pipeline hazards - Pipelined datapath and control - Handling data hazards and control hazards - Exceptions - Introduction to advanced ILP.

SUGGESTED ACTIVITIES :

- Combinations of in Class & Flipped class rooms
- Practical - Study of the pipelined implementation and analysis of various hazards on the standard simulator

SUGGESTED EVALUATION METHODS:

- Assignment problems involving instruction sequences and real-time scenarios
- Quizzes

MODULE VIII:	**L**	**T**	**P**	**EL**
	6	**0**	**4**	**6**

Need for a hierarchical memory system - The basics of caches - Measuring and improving cache performance. Virtual memory - Paging and segmentation - TLB - Implementing protection with virtual memory. A common framework for memory hierarchies, Associative memories, Introduction to virtual machines.

SUGGESTED ACTIVITIES :

- Flipped classroom
- Practical - Implement a simple functional model of a set-associative cache in C/C++. Study hit/miss rates for various access patterns. Experiment with different replacement policies.

- EL - Writing simple programs to study the behaviour of the memory hierarchy of your own laptop/ PC
  - Analyzing the performance of the memory hierarchy by varying different parameters

SUGGESTED EVALUATION METHODS:

- Assignment problems
- Quizzes
- Practical component evaluation

MODULE IX:	**L**	**T**	**P**	**EL**
	3	**0**	**2**	**3**

Storage and I/O - Dependability, reliability and availability - Disk storage - Flash storage - Connecting processors, memory and I/O devices - Interfacing I/O devices to the processor, memory and the operating system, Designing an I/O system, Parallelism and I/O, RAID.

SUGGESTED ACTIVITIES :

- EL - Survey of storage devices (NAS/SAN/RAID etc.) on different classes of systems
- Practical – Continue with the exercises on memory hierarchy

SUGGESTED EVALUATION METHODS:

- Survey evaluation – mindmap

OUTCOMES:

Upon completion of the course, the students will be able to:

- Evaluate the performance of computer systems
- Design a simple instruction execution unit
- Point out the hazards present in a pipeline and suggest remedies
- Explain the data path and control path implementation of a processor
- Modify some features of an architectural simulator
- Critically analyse the various characteristics of the hierarchical memory and I/O devices and their interface to the processor

TEXT BOOKS:

1. David A. Patterson and John L. Hennessy, “Computer Organization and Design: The Hardware/Software Interface”, Fifth Edition, Morgan Kaufmann / Elsevier, 2013.
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky and Naraig Manjikian, “Computer Organization and Embedded Systems”, Sixth Edition, Tata McGraw Hill, 2012.

REFERENCE BOOKS:

1. William Stallings, “Computer Organization and Architecture – Designing for Performance”, Sixth Edition, Pearson Education, 2003.
2. John P. Hayes, “Computer Architecture and Organization”, Third Edition, Tata McGraw Hill, 1998.
3. V.P. Heuring, H.F. Jordan, “Computer Systems Design and Architecture”, Second Edition, Pearson Education, 2004.
4. Behrooz Parhami, “Computer Architecture”, Oxford University Press, 2007.

EVALUATION METHOD TO BE USED:

Continuous assessment	Mid term	End Semester
15(T) + 25 (P)	20	40

CO - PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓		✓	✓				✓		✓	✓
CO2	✓	✓	✓	✓	✓				✓			✓
CO3	✓	✓	✓	✓	✓				✓			✓
CO4	✓	✓	✓	✓	✓				✓			✓
CO5	✓	✓	✓	✓	✓				✓			✓
CO6	✓	✓										✓

CS 6108 OPERATING SYSTEMS

Prerequisites for the course: None

OBJECTIVES:

- To learn the basic concepts and functions of operating systems
- To learn the mechanisms of operating systems to handle processes and threads and their communication
- To know the components and management aspects of concurrency management
- To study the basic components of scheduling mechanism
- To learn the mechanisms involved in memory management in contemporary OS
- To appreciate the emerging trends in Operating Systems
- To learn programmatically to implement simple OS mechanisms

OPERATING SYSTEMS	L	T	P	EL	TOTAL	CREDITS
	3	0	4	3		6

MODULE I INTRODUCTION TO OPERATING SYSTEMS	L	T	P	EL
	4	0	4	4

Introduction to OS – Operating System Services – Operating System Operations – Virtualization – User and Operating System Interface – System Calls – Operating System Structures - Building and Booting an Operating System

SUGGESTED ACTIVITIES:
PRACTICAL:
l - Shell programming assignments