ADVANCED COMPUTER ARCHITECTURE

1. Introduction

Credits: 3-0-0

2. Prerequisite

Computer Organization, Basics of Computer Architecture, Operating Systems.

3. Course Outline

UNIT - I: Fundamentals of Quantitative Design and Analysis

Classes of computers, Trends in technology, Measuring and Reporting Performance, Power consumption and efficiency as the metric, Dependability, Quantitative Principles of Computer Design.

UNIT - II: Instruction Set Principles

Classifying Instruction Set Architectures, Memory Addressing, Addressing modes, Operations in the instruction set, Instructions for control flow, encoding an instruction set, Role of compilers.

UNIT - III: Memory Hierarchy Design

Cache performance review, Four basic memory hierarchy questions, Six basic optimizations of cache performance, Ten advanced optimizations of cache performance, Protection: virtual memory and virtual machines, memory technology and optimizations: SRAM, DRAM, Flash memory, Graphics Data RAMs.

UNIT - IV: Instruction-Level Parallelism and its Dynamic Exploitation

Instruction level parallelism: concepts and challenges, basics of pipelining, data hazards, structural hazards, control hazards, minimizing data hazards through forwarding, overcoming branch penalties by delayed branches, static and dynamic branch prediction, dealing with exceptions in pipelining, dynamic scheduling, Tomasulo's algorithm, speculative processors, high performance instruction delivery, VLIW approach, static and dynamic superscalar processors.

UNIT - V : Multiprocessors and Thread-Level Parallelism

Multithreading: exploiting thread-level parallelism within a processor, Symmetric shared-memory architectures and their performance, Distributed shared-memory architectures and their performance, synchronization, models of memory consistency.

4. Reading Material

Text Books

1. John L. Hennessey and David A. Patterson. Computer Architecture: A Quantitative Approach, 5E, Morgan-Kaufmann, 2012. ISBN-13: 978-0123838728.

Suggested Assignments

Assignments can primarily be exercises from Hennessy and Patterson with modifi-
cations to them.

1. Problems involving use of Amdahl's law.
2. Problems that use the timing diagram of pipelining to estimate the number of cycles taken for execution of a set of instructions.
3. Problems involving branch prediction and branch target buffers and estimating cycles taken for instruction with and without these optimizations.
4. Problems involving memory operations and improvement in performance through use of different cache optimizations.
5. Problems involving use of dynamic scheduling and speculation to compare the performance of the system with these optimizations versus regular pipelining.
6. Problems involving the centralized and distributed shared-memory architectures and operations involved in accessing different memory locations present in multiple processors.