**NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

| **COURSE OUTLINE TEMPLATE** | | | |
|---|---|---|---|
| **Course Title** | **Parallel Computer Architecture** | | |
| **Course Code** | **CS605** | **No. of Credits** | **3** |
| **Department** | **CSE** | **Faculty** | **Dr. N. Ramasubramanian** |
| **Pre-requisites Course Code** | **NIL** | | |
| **E-mail** | **nrs@nitt.edu** | **Telephone No.** | **0431-2503204** |
| **Course Type** | ☐ **Core course** ☐ **Elective course** | | |

**COURSE OVERVIEW**

Parallel Computer Architecture course gives an emphasis on the design aspects of parallelism that can be achieved throgh different techniques. It also gives a detailed overiew of different ways to improve the performance gain in computer systems. The study of recent processor architectures and architectural support for diffferent classes of processors is theprime focus of this course.

**COURSE OBJECTIVES**

- To understand the principles of underlaying parallel computer architecture
- To understand the design of parallel computer systems including multi-core architectures
- To assess the communication and computing capabilities of parallel architectures and to predict the performance of these systems

**COURSE OUTCOMES (CO)**

| **Course Outcomes** | **Aligned Programme Outcomes (PO)** |
|---|---|
| 1. Ability to evaluate peformance of computer systems & also compare design choices | PO- 1 |
| 2. Ability to evaluate improvement in performance due to various parallel computing techniques | PO-1, PO-6 |
| 3. Ability to design memory subsystems & compare the performance due to various memory automation techniques | PO-8 |

| COURSE TEACHING AND LEARNING ACTIVITIES | | | |
|---|---|---|---|
| **S. No.** | **Week** | **Topic** | **Mode of Delivery** |
| 1 | 1 | Introduction, Defining Computer Architecture, Trends in Technology | Pen-Board |
| 2 | 1 | Trends in Power in Integrated Circuits – Trends in Cost | Pen-Board |
| 3 | 1 | Dependability, Measuring, Reporting and Summarizing Performance | Pen-Board |
| 4 | 2 | Performance equation, Amdhal's Law, Quantitative Principles of Computer Design | Pen-Board |
| 5 | 2 | Problems on Performance equation, Amdhal's Law | PPT |
| 6 | 2 | Basic and Intermediate concepts of pipelining | PPT |
| 7 | 3 | Pipeline Hazards, Dealing with Hazards, Pipelining Implementation issues | PPT |
| 8 | 3 | Instruction-Level Parallelism: Concepts and Challenges | PPT |
| 9 | 3 | Basic Compiler Techniques for Exposing ILP | PPT |
| 10 | 4 | Basic Compiler Techniques for Exposing ILP contd. | PPT |
| 11 | 4 | Problems and discussions | PPT |
| 12 | 4 | Reducing Branch Costs with Prediction – Overcoming Data Hazards with Dynamic Scheduling | PPT |
| 13 | 5 | Dynamic Scheduling: Algorithm and Examples – Hardware-Based Speculation | PPT |
| 14 | 5 | Exploiting ILP Using Multiple Issue and Static Scheduling | PPT |
| 15 | 5 | Exploiting ILP Using Dynamic Scheduling | PPT |
| 16 | 6 | Multiple Issue and Speculation, Problems and discussions | PPT |
| 17 | 6 | Studies of the Limitations of ILP – Limitations on ILP for Realizable Processors | PPT |
| 18 | 6 | Hardware versus Software Speculation | PPT |
| 19 | 7 | Using ILP Support to Exploit Thread-Level Parallelism | PPT |
| 20 | 7 | Problems and discussions | PPT |
| 21 | 7 | Multithreading: Increasing thread level parallelism to improve uniprocessor throughput | PPT |
| 22 | 8 | Case Studies given in end of chapter | PPT |

| | | | |
|---|---|---|---|
| 23 | 8 | Cache Performance – Six Basic Cache Optimizations | PPT |
| 24 | 8 | Ten Advanced Optimizations of Cache Performance | PPT |
| 25 | 9 | Ten Advanced Optimizations of Cache Performance contd. | PPT |
| 26 | 9 | Virtual Memory – Protection and Examples of Virtual Memory | PPT |
| 27 | 9 | Memory Technology and Optimizations | PPT |
| 28 | 10 | Protection: Virtual Memory and Virtual Machines – The Design of Memory Hierarchies | PPT |
| 29 | 10 | Centralized Shared-Memory Architectures – Performance of Shared-Memory Multiprocessors | PPT |
| 30 | 10 | Distributed Shared Memory concept | PPT |
| 31 | 11 | Cache coherence and basics | PPT |
| 32 | 11 | Snoopy Bus Protocol for cache coherence | PPT |
| 33 | 11 | Directory Based Coherence, different types of directories | PPT |
| 34 | 12 | Case studies and problems given on case studies for cache coherence | PPT |
| 35 | 12 | Basics of Synchronization – Models of Memory Consistency | PPT |
| 36 | 12 | Introduction to Warehouse-Scale Computers | PPT |
| 37 | 13 | Programming Models for Warehouse-Scale Computers | PPT |
| 38 | 13 | Computer Architecture of Warehouse-Scale Computers | PPT |
| 39 | 13 | Physical Infrastructure and Costs of Warehouse-Scale Computers | PPT |
| 40 | 13 | Discussions and Problem/ Doubt solving | PPT |

| COURSE ASSESSMENT METHODS | | | | |
|---|---|---|---|---|
| **S.No.** | **Mode of Assessment** | **Week/Date** | **Duration** | **% Weightage** |
| 1. | Cycle Test I | As per Academic Schedule | 1 Hr | 20% |
| 2. | Cycle Test II | As per Academic Schedule | 1 Hr | 20% |
| 3. | Assignment | Before End Sem Exam | ~1 Week | 10% |
| 4. | End Sem Exam | As per Academic Schedule | 3 Hrs | 50% |

**ESSENTIAL READINGS : Textbooks, reference books Website addresses, journals, etc**

1. David.A.Patterson, John L.Hennessy, "Computer Architecture: A Quantitative approach", Elsevier, 5 th Edition 2012.

2. K.Hwang, Naresh Jotwani, “Advanced Computer Architecture, Parallelism, Scalability, Programmability”, Tata McGraw Hill, 2 nd Edition 2010.

**COURSE EXIT SURVEY (mention the ways in which the feedback about the course is assessed and indicate the attainment also)**

On-line Feedback
Live Feedback in class

**COURSE POLICY (including plagiarism, academic honesty, attendance, etc.)**

Academic honesty
Attendance Minimum 75% Mandatory

**ADDITIONAL COURSE INFORMATION**

Students can meet faculty for discussion and queries any time during working hours.

**FOR SENATE’S CONSIDERATION**

**Course Faculty** _____ **CC-Chairperson** _____ **HOD** _____