INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad -500 043

COMPUTER SCIENCE AND ENGINEERING

COURSE INFORMATION SHEET

Course Title	**HIGH PERFORMANCE ARCHITECTURE**			
Course Code	**BCS003**			
Programme	**M.Tech.,**			
Semester	**I**			
Course Type	**Core**			
Regulation	**IARE - R16**			
Course Structure	**Lectures**	**Tutorials**	**Practicals**	**Credits**
	4	**-**	**-**	**4**
Course Coordinator	**Dr. K. Rajendra Prasad, Professor & Head** **Department of Computer Science and Engineering**			
Course Faculty	**Mr. R.M.Noorullah, Associate Professor**			

I. COURSE OVERVIEW:

The Present course concentrates on developing basic understanding about various activities that are involved in increasing the performance of a compiler, the compiling issues for various parallel architecture, transformation techniques for code parallelization and to understand memory management and scheduling for parallel machines. This course enables the student to acquire necessary skills for optimizing compiler performance, and it focuses on all activities involved in data dependence and various data dependences testing methodologies.. In this course students will gain a broad understanding of the discipline of compiling for scalar, super scalar, vector and parallel processors. Student can implement and get knowledge about development of the software and hardware gains knowledge of loop normalization, ZIV, SIV and MIV testing and their appropriate application. A general understanding of fine grained and enhancing fine grained by loop distribution, coarse grained and enhancing coarse grained by privatization, handling control flow by using if-conversion and improving register usage.

II. COURSE PRE-REQUISITES:

Level	Course Code	Semester	Prerequisites	Credits
PG	BCS003	I	High Performance Architecture	4

III. MARKS DISTRIBUTION

Subject	SEE Examination	CIA Examination	Total Marks
High Performance Architecture	70 Marks	30 Marks	100

Semester End Examination (SEE):
The SEE is conducted for 70 marks of 3 hours duration. The syllabus for the theory courses is divided into FIVE units and each unit carries equal weight age in terms of marks distribution. The question paper pattern is as follows: two full questions with ‘either’ ‘or’ choice will be drawn from each unit. Each question carries 14 marks.

Continuous Internal Assessment (CIA):

CIA is conducted for a total of 30 marks, with 25 marks for Continuous Internal Examination (CIE) and 05 marks for Quiz / Alternative Assessment Tool (AAT).

Continuous Internal Examination (CIE):

The CIE exam is conducted for 25 marks of 2 hours duration consisting of two parts. Part–A shall have five compulsory questions of one mark each. In part–B, four out of five questions have to be answered where, each question carries 5 marks. Marks are awarded by taking average of marks scored in two CIE exams.

Quiz / Alternative Assessment Tool (AAT):

Two Quiz exams shall be online examination consisting of 20 multiple choice questions and are be answered by choosing the correct answer from a given set of choices (commonly four). Marks shall be awarded considering the average of two quizzes for every course. The AAT may include seminars, assignments, term paper, open ended experiments, micro projects, five minutes video and MOOCs.

IV. DELIVERY / INSTRUCTIONAL METHODOLOGIES:

√	CHALK & TALK	√	QUIZ	√	ASSIGNMENTS	√	MOOCs
√	LCD / PPT	√	SEMINARS	√	MINI PROJECT	√	VIDEOS
√	OPEN ENDED EXPERIMENTS						

V. ASSESSMENT METHODOLOGIES – DIRECT

√	CIE EXAMS	√	SEE EXAMS	√	ASSIGNEMNTS	√	SEMINARS
√	STUDENT VIVA	√	MINI PROJECT	X	CERTIFICATION	√	TERM PAPER

VI. ASSESSMENT METHODOLOGIES – INDIRECT

√	ASSESSMENT OF COURSE OUTCOMES (BY FEEDBACK, ONCE)	√	STUDENT FEEDBACK ON FACULTY (TWICE)
√	ASSESSMENT OF MINI PROJECTS BY EXPERTS		

VII. COURSE OBJECTIVES:

The course should enable the students to:

I.	Understand the compiling issues for various parallel architectures.
II.	Implementation of transformation technology for code parallelization.
III.	Familiar with the concepts of Data dependence, loop normalization, ZIV,SIV,MIV testing, fine grained for loop distribution, course grained by privatization, handling flow control and improving register reuse.
IV.	Understand the memory management and scheduling for code parallelization.
V.	Understand the optimizing compiler performance for High Performance.

VIII . COURSE LEARNING OUTCOMES:

Students, who complete the course, will have demonstrated the ability to do the following:

BCS003.01	Understand the key concerns that are common to improve the performance of compiler.
BCS003.02	Describe Compiling for scalar, super scalar, VLIW, vector and parallel processor.
BCS003.03	Memorizing Bernstein's condition to execute parallel processing.
BCS003.04	Describe the concept of Data Dependence, types and loop carried and loop independent dependence.

BCS003.05	Understand Loop normalization , parallelization , vectorization and scalar renaming.
BCS003.06	Demonstrate Simple dependence testing and subscript portioning.
BCS003.07	Describe the concept of single subscript and multiple induction variable tests.
BCS003.08	Understand the importance of Delta test in testing coupled group.
BCS003.09	Understand the concept more powerful and multiple simple test.
BCS003.10	Describe the overall dependence testing.
BCS003.11	Memorize fine grained and enhancing fine grained by using loop distribution.
BCS003.12	Understand the principles of loop interchange for vectorization.
BCS003.13	Describe the course grained and enhancing by using privatization.
BCS003.14	Understand loop interchange for parallelization.
BCS003.15	Describe how to handle control flow by using if-conversion.
BCS003.16	Describe the concepts of memory hierarchy used in parallelization for improving performance.
BCS003.17	Understand the concepts of scalar register allocation and cache memory management.
BCS003.18	Implement scalar replacement techniques to optimizing compilers.
BCS003.19	Understand the concept of unroll-and-jam.
BCS003.20	Describe cache blocking and perfecting in increasing performance of parallel architecture.
BCS003.21	Understand to improving register usage by scalar register allocation.
BCS003.22	Understand the concept of data dependence for register reuse.
BCS003.23	Describe the scheduling and tracking in Risk Mitigation, Monitoring and Management Plan.
BCS003.24	Understand loop carried and loop dependent reuse in increasing performance of a compiler.
BCS003.25	Describe pruning dependence graph in register reuse to improve performance.
BCS003.26	Demonstrate dependence spanning multiple iterations and loop inter change for register reuse.
BCS003.27	Possess the knowledge and skills for improving performance of compiler and to succeed in national and international level competitive exams.

IX. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes (POs)		Level	Proficiency assessed by
PO 1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	H	Assignments
PO 2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	H	Assignments

Program Outcomes (POs)		Level	Proficiency assessed by
PO 3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	S	Mini Project
PO 4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	S	Open ended experiments /
PO 5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	S	Mini Project
PO 6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.	N	---
PO 7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	N	---.
PO 8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	N	---
PO 9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	N	---
PO 10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	S	Seminars / Term Paper / 5 minutes video
PO 11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	N	---
PO 12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	S	----

N= None S= Supportive H = Highly Related

X. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes		Level	Proficiency assessed by
PSO 1	**Professional Skills:** The ability to research, understand and implement computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient analysis and design of computer-based systems of varying complexity.	H	Lectures, Assignments
PSO 2	**Problem-Solving Skills:** The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.	H	Projects
PSO 3	**Successful Career and Entrepreneurship:** The ability to employ modern computer languages, environments, and platforms in creating innovative career paths, to be an entrepreneur, and a zest for higher studies.	S	Guest Lectures

N - None S - Supportive H - Highly Related

XI. SYLLABUS:

UNIT I PARALLEL AND VECTOR ARCHITECTURES
Compiling for scalar pipeline, compiling for vector pipeline, super scalar and VLIW processors, compiling for multiple issue processors, processor parallelism, Bernstein's conditions. The role of dependence, Dependence analysis: Concept of dependence, classification of dependences, dependence in loop, deppendence distance, dependence direction, loop carried and loop independent dependences, level of loop carried dependence. Simple dependence testing, vectorization and parallelization. Preliminary transformations required to make dependence testing more accurate, loop normalization, scalar data flow analysis, induction variable substitution, scalar renaming.
UNIT II DEPENDENCE TESTING
Dependence Testing: Introduction, background and terimnology, dependence testing overview, subscript partitioning, merging direction vectors, single-subscript dependence tests, ZIV test, SIV test, MIV test, testing in coupled groups-the Delta test, more powerful multiple subscript test, an empirical study, putting it all together.
UNIT III FINE-GRAINED AND COARSE-GRAINED PARALLELISM
Fine-Grained parallelism: Enhancing fine-grained parallelism using loop distribution. Use of loop interchange for vectorization, scalar and array renaming, use of loop skewing. Coarse-Grained parallelism: Enhancing coarse-grained parallelism using privatization and scalar expansion, loop alignment, loop fusion, use of loop interchange for parallelization.
UNIT IV HANDLING CONTROL FLOW
Types of branches: if-conversion. Management of Memory Hierarchy: scalar register allocation and management of cache memory. Topics include scalar replacement, unroll-and-jam, loop alignment, cache blocking and perfecting.
UNIT V IMPROVING RESGISTER USAGE
Improving Register Usage: Introduction, scalar register allocation, data dependence for register reuse, loop carried and loop independent reuse, a register allocation example, scalar replacement, pruning the dependence graph, simple replacement, handling loop carried dependences, dependence spanning multiple iterations, eliminating scalar copies, loop interchange for register reuse.

TEXT BOOKS:

1.	Allen and Kennedy, " Optimizing compilers for Modern Architectures", Morgan-Kaufman, 1st edition, 2001.
2.	Wolfe, High Performance Compilers for Parallel Computing, Addison-Wesley, 1st edition, 1996.

REFERENCES:

1.	Banerjee, " Dependence Analysis", Kluwer Academic Publishers, 1st edition, 1997.
2.	Wolfe, Optimizing Super Compilers for Supercomputers, MIT Press.
3.	Zima and Chapman, Super Compilers for parallel and Vector Computers. ACM Press.

XII. COURSE PLAN:

The course plan is meant as a guideline. There may probably be changes.

Lecture No	Topic Outcomes	Topic/s to be covered	Reference
1-2	Parallel and Vector Architecture	Compiling for scalar pipeline, compiling for vector pipeline, super scalar and VLIW processors, compiling for multiple issue processors, processor parallelism, Bernstein's conditions. The role of dependence	T1: 1.1-1.5
3-5	Dependence Analysis	Concept of dependence, classification of dependences, dependence in loop, deppendence distance, dependence direction, loop carried and loop independent dependences, level of loop carried dependence.	T1: 2.1-2.8

Lecture No	Topic Outcomes	Topic/s to be covered	Reference
6-10	Vectorization and Parallelization	Preliminary transformations required to make dependence testing more accurate, loop normalization, scalar data flow analysis, induction variable substitution, scalar renaming.	T1: 3.1-3.6
11-12	Dependence Testing	Introduction, background and terimnology, dependence testing overview, subscript partitioning, merging direction vectors	T1: 5.1-5.3
13-15	Single-subscript dependence tests	ZIV test, SIV test, MIV test	T1: 5.4-5.7
16-19	Testing in coupled groups	the Delta test, more powerful multiple subscript test	T2:5.1-5.5
20-22	Empirical Study	an empirical study, putting it all together	T1:10.2.3
23-26	Fine-Grained parallelism	Enhancing fine-grained parallelism using loop distribution. Use of loop interchange for vectorization,	T1:8.1-8.4
24-28	Fine-Grained parallelism	scalar and array renaming, use of loop skewing.	T1:9.1-9.6
29-35	Coarse-Grained parallelism	Enhancing coarse-grained parallelism using privatization and scalar expansion, loop alignment	T1:11.1-11.4
33-40	Coarse-Grained parallelism	loop fusion, use of loop interchange for parallelization	T1:20.1-20.7
41-42	Handling Control Flow	Types of branches: if-conversion. Management of Memory Hierarchy: scalar register allocation	T1:20.8-20.9
42-45	Handling Control Flow	management of cache memory. Topics include scalar replacement, unroll-and-jam, loop alignment, cache blocking and perfecting.	T1:25.1-20.3
47-50	Improving Register Usage Introduction	scalar register allocation, data dependence for register reuse, loop carried and loop independent reuse, a register allocation example	T1:28.1-28.7
50-55	Scalar replacement	scalar replacement, pruning the dependence graph, simple replacement	T1:23.1-23.8
54-60	Handling dependences to improve register reuse	handling loop carried dependences, dependence spanning multiple iterations, eliminating scalar copies, loop interchange for register reuse.	T1:24.1-23.6

XIII. GAPS IN THE SYLLABUS - TO MEET INDUSTRY / PROFESSION REQUIREMENTS:

S NO	DESCRIPTION	PROPOSED ACTIONS	RELEVANCE WITH POs	RELEVANCE WITH PSOs
1	How to collect useful Requirements to build Right Product	Seminars / Guest Lectures / NPTEL	PO 1, PO 2, PO 3	PSO 1, PSO 2
2	Real time Risk management System	Seminars / Guest Lectures / NPTEL	PO 2, PO 3	PSO 1
3	Generation of test cases using ATM machine and Banking Applications	Assignments / Laboratory Practices	PO 1, PO 3, PO 4	PSO 2

XIV. MAPPING COURSE OBJECTIVES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Objectives (COs)	Program Outcomes (POs)												Program Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
I	H	S	S	S									S	H	
II	S	S	S	H	S								H		S
III	H	H	S	S							S	S		H	
IV	H	S	H	S	S						S	S		H	
V	S											H	H		S

S= Supportive **H = Highly Related**

XV. MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Learning Outcomes (CLOs)	Program Outcomes (COs)												Program Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
BCS003.01	H	S												S	
BCS003.02	S	H	S											S	
BCS003.03	S	H		S	S								S		
BCS003.04	H	S	H	S								S		H	
BCS003.05	S	H	S	S								S		H	
BCS003.06		H	S									S	S		
BCS003.07				H								S	S		
BCS003.08	S	H			S									S	
BCS003.09		S	H											S	
BCS003.10		H										S		S	
BCS003.11	S	H	S									S	S	S	S
BCS003.12		S		H									S	S	
BCS003.13	S	S	S										S	S	

BCS003.14		H		S	S							S		H	
BCS003.15	S	S											S	S	
BCS003.16	S	H		S	S							S		H	
BCS003.17	H			S								S		S	
BCS003.18	H			S								S	S	H	
BCS003.19	H	S			S									H	
BCS003.20									S		S	H	S		S

S= Supportive **H = Highly Related**

XVI. DESIGN BASED PROBLEMS (DP) / OPEN ENDED PROBLEM:

I. Suggest a few ways to increase the performance of parallel architecture.

II. Demonstrate how to optimize compiler performance by using dependence and transformation without affecting results.

III. Demonstrate various types of Dependence testing by considering an Empirical Case study.

IV. Suggesting different ways to Improve Register usage by pruning the Dependence graph, spanning multiple iterations and eliminating scalar copies for Register Reuse.

Prepared by
Mr. R.M.Noorullah, Associate Professor

HOD, CSE