CP5008 COMPILER OPTIMIZATION TECHNIQUES

DETAILED SYLLABUS

UNIT I INTERMEDIATE REPRESENTATIONS AND ANALYSIS

Review of Compiler Structure- Structure of an Optimizing Compiler – Intermediate Languages - LIR, MIR, HIR – Control Flow Analysis – Iterative Data Flow Analysis – Static Single Assignment – Dependence Relations - Dependences in Loops and Testing-Basic Block Dependence DAGs – Alias Analysis.

UNIT II EARLY AND LOOP OPTIMIZATIONS

Importance of Code Optimization Early Optimizations: Constant-Expression Evaluation - Scalar Replacement of Aggregates - Algebraic Simplifications and Re-association – Value Numbering - Copy Propagation - Sparse Conditional Constant Propagation. Redundancy Elimination: Common - Subexpression Elimination - Loop-Invariant Code Motion - PartialRedundancy Elimination - Redundancy Elimination and Reassociation - Code Hoisting. Loop Optimizations: Induction Variable Optimizations - Unnecessary Bounds Checking Elimination.

UNIT III PROCEDURE OPTIMIZATION AND SCHEDULING

Procedure Optimizations: Tail-Call Optimization and Tail-Recursion Elimination – Procedure Integration - In-Line Expansion - Leaf-Routine Optimization and Shrink Wrapping. Code Scheduling: Instruction Scheduling - Speculative Loads and Boosting – Speculative Scheduling - Software Pipelining - Trace Scheduling - Percolation Scheduling. Control-Flow and Low-Level Optimizations : Unreachable-Code Elimination - Straightening – If Simplifications - Loop Simplifications -Loop Inversion – Un-switching - Branch Optimizations - Tail Merging or Cross Jumping - Conditional Moves - Dead-Code Elimination – Branch Prediction - Machine Idioms and Instruction Combining.

UNIT IV INTER PROCEDURAL OPTIMIZATION

Symbol table – Runtime Support - Interprocedural Analysis and Optimization: Interprocedural Control Flow Analysis - The Call Graph - Interprocedural Data-Flow Analysis - Interprocedural Constant Propagation - Interprocedural Alias Analysis –

Interprocedural Optimizations - Interprocedural Register Allocation - Aggregation of Global References.

UNIT V REGISTER ALLOCATION AND OPTIMIZING FOR MEMORY

Register Allocation: Register Allocation and Assignment - Local Methods - Graph Coloring – Priority Based Graph Coloring - Other Approaches to Register Allocation. Optimization for the Memory Hierarchy: Impact of Data and Instruction Caches - Instruction-Cache Optimization - Scalar Replacement of Array Elements - Data-Cache Optimization – Scalar vs. Memory-Oriented Optimizations.

REFERENCES:

1. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, "Compilers: Principles, Techniques, and Tools", Addison Wesley, Second Edition, 2007.

2. Andrew W. Appel, Jens Palsberg, "Modern Compiler Implementation in Java", Cambridge University Press, Second Edition, 2002.

3. Keith Cooper, Linda Torczon, "Engineering a Compiler", Morgan Kaufmann, Second Edition, 2011.

4. Robert Morgan ,‖Building an Optimizing Compiler‖, Digital Press, 1998

5. Steven Muchnick, ―Advanced Compiler Design and Implementation‖, Morgan Kaufman Publishers, 1997.

6. Randy Allen and Ken Kennedy, ―Optimizing Compilers for Modern Architectures: A Dependence based Approach‖, Morgan Kaufman, 2001.

OBJECTIVES:

- To be aware of different forms of intermediate languages and analyzing programs.
- To understand optimizations techniques for simple program blocks.
- To apply optimizations on procedures, control flow and parallelism.
- To learn the inter procedural analysis and optimizations.
- To explore the knowledge about resource utilization.