# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bonafide Certificate</td>
<td>ii</td>
</tr>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgement</td>
<td>vi</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xv</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xvi</td>
</tr>
<tr>
<td>List of Symbols, Abbreviations &amp; Nomenclature</td>
<td>xix</td>
</tr>
</tbody>
</table>

## 1. INTRODUCTION

1.1 Parallel Processing System (Parallel Computer) 2
   1.1.1 Parallel Processing –Why? 4
   1.1.2 Hardware Architecture of Parallel Computer 5
   1.1.3 Memory Architecture of Parallel Computer 8
      1.1.3.1 Shared Memory 9
      1.1.3.2 Distributed Memory 13
      1.1.3.3 Hybrid Distributed-Shared Memory 14
   1.1.4 Approaches to Parallel Programming 14

1.2 Motivation 15
   1.2.1 Perfection of DSM System 16
   1.2.2 Need and Importance of Structural Memory Model 16
   1.2.3 Memory Consistency Maintenance 17
1.2.4 Improvisation of Parallelism
1.2.5 Memory Management Technique
1.3 Context
1.4 Thesis Organization and Overview
1.5 Summary of Contribution
1.6 Research Methodology
REFERENCE

2 BACKGROUND
2.1 Designing issue of DSM System
   2.1.1 Relationship with Virtual Memory Manager
   2.1.2 Choice of Consistency Model
   2.1.3 Choice of Granularity
   2.1.4 Naming Scheme
2.2 Implementation of DSM System
   2.2.1 Software DSM Implementation
   2.2.2 Hardware DSM Implementation
   2.2.3 Hybrid Level DSM Implementation
2.3 Classification of DSM System
   2.3.1 Paged Based DSM System
   2.3.2 Shared Variable DSM System
   2.3.3 Object Based DSM System
2.4 Challenges in DSM Architecture
   2.4.1 Implementation Level
   2.4.2 Memory Consistency Model
   2.4.3 Communication Channel
2.5 Memory Coherence in DSM Architecture
2.5.1 Write Invalidate Coherence Protocol 45
2.5.2 Write-Update Coherence Protocol 45

2.6 Memory Consistency in DSM Architecture 47
2.6.1 Classification 49
   2.6.1.1 Strong Memory Consistency 49
   2.6.1.2 Weak Memory Consistency 51
   2.6.1.3 Relaxed Memory Consistency 52

2.6.2 Type of Memory Consistency Model 52
   2.6.2.1 Uniform Memory Model 54
   2.6.2.2 Hybrid Memory Model 54

2.7 DSM System – Some Implementation 55
   2.7.1 Midway 57
   2.7.2 Merlin 58
   2.7.3 RMS 59
   2.7.4 Plus 59
   2.7.5 Galactica Net 60

2.8 Our Approach 61

REFERENCE 64

3 STRUCTURAL UNIFORM MEMORY MODEL 69

3.1 Structural Uniform Memory Model - Design 69
3.2 Defining Uniforms Memory Models 70
   3.2.1 Atomic Consistency (AC) 73
   3.2.2 Sequential Consistency (SC) 74
   3.2.3 Causal Consistency (CC) 76
   3.2.4 Processor Consistency (PC) 78
   3.2.5 Pipelined RAM (PRAM) 80
3.2.6 Cache Consistency(CcC) 81
3.2.7 Slow Memory Consistency 82
3.3 Relating the Uniform Memory Models 83
  3.3.1 SC versus CcC 84
  3.3.2 SC versus PRAM 86
  3.3.3 SC versus CC 88
3.4 Verification of Uniform Memory Models 89
  3.4.1 Verification of Sequential Consistency – VSC 91
  3.4.2 Verification of Casual Consistency – VCC 92
  3.4.3 Verification of Processor Consistency – VPC 93
  3.4.4 Verification of Cache Consistency – VCcC 93
REFERENCE 95

4 STRUCTURAL HYBRID MEMORY MODEL 98
  4.1 Structural Hybrid Memory Model – Design 98
  4.2 Defining Hybrid Memory Models 102
    4.2.1 Weak Consistency (WC) 103
    4.2.2 Release Consistency (RC) 105
      4.2.2.1 Eager Release Consistency (ERC) 107
      4.2.2.2 Lazy Release Consistency (LRC) 107
    4.2.3 Entry Consistency (EC) 108
    4.2.4 Scope Consistency (ScC) 111
    4.2.5 View Based Consistency (VC) 113
  4.3 Relating the Hybrid Memory Models 114
    4.3.1 Weak versus Release Consistency 116
    4.3.2 Release versus Scope Consistency 117
    4.3.3 Entry versus Scope Consistency 118
4.4 Verification of the Hybrid Memory Models

4.4.1 Verification of Weak Consistency – VWC

4.4.2 Verification of Release Consistency – VRC

4.4.3 Verification of Entry Consistency – VEC

4.4.4 Verification of Scope Consistency – VScC

REFERENCE

5 MEMORY CONSISTENCY MAINTENANCE

5.1 Framework to Maintain Memory Consistency

5.1.1 Issue Involved

5.1.2 Basic Concept

5.2 Designing Issue of Framework

5.2.1 Memory Consistency

5.2.2 Coherence Protocol

5.2.2.1 Memory Management

5.2.2.2 Synchronization

5.2.2.3 Ownership Management

5.2.3 Communication Service

5.2.4 Concurrency Control

5.3 Maintaining Consistency

5.3.1 Finding the Destination and Incorporating Updates

5.3.2 Propagation of Updates

REFERENCE

6 PARALLELISM IMPROVEMENT

6.1 Data Replication
6.1.1 Migration algorithm (SRSW) 147
6.1.2 Read-replication Algorithm (MRSW) 148
6.1.3 Full Replication Algorithm (MRMW) 149

6.2 Extending the framework and Implementation 149
6.2.1 Processor Consistency Implementation 150
6.2.2 Write-Update Coherence Protocol Implementation 153

6.3 Working 154
6.3.1 Finding the destination and incorporating updates 155
6.3.2 Propagating updates to all nodes 156
6.3.3 Algorithm for creation of Page Map Table 158
6.3.4 Algorithm for creation of copy list 159
6.3.5 Algorithm for writing by Initiator processor 160
6.3.6 Algorithm for Sending and accepting message 161
6.3.7 Algorithm for updating message 161

6.4 Functional Representation of Algorithms 163

6.5 Degree of Parallelism 164
6.5.1 Time to update one write 165
6.5.2 Total Time to update all write 165

6.6 Performance Analysis 168

REFERENCE 172

7 PAGING AND PERFORMANCE ANALYSIS 175

7.1 Granularity 176
7.1.1 Effect of Data Granularity 176
7.1.2 Effect of Execution Granularity 177

7.2 Thrashing 178

7.3 False Sharing 179
7.4 Paging Implementation 180
7.5 Overhead Analysis 180
   7.5.1 Page Fault Servicing overhead 181
   7.5.2 Coherency Maintenance Overhead 183
   7.5.3 Network Communication Overhead 184
7.6 Page size Calculation 185
7.7 Page Size Analysis 189
REFERENCE 193

8 CONCLUSION 196
   8.1 Summary 198
   8.2 Observation and Limitation 199
   8.3 Future Work 200

Appendix – I Correlation 202
Appendix – II List of Publications 206
Bibliography 210