CHAPTER 1 Fundamentals of Compiling for Parallel Machines **9**

1.1 Introduction to Parallel Architectures **9**

1.1.1 Intra-functional unit parallelism 10
1.1.2 Inter-functional unit parallelism 13
1.1.3 Memory Hierarchy 17

1.2 Compiler Control of High-Performance Hardware **19**

1.2.1 Compiling for Pipelines. 19
1.2.2 Compiling for Asynchronous Parallelism 25
1.2.3 Compiling for Memory Hierarchy 27
1.2.4 A Case Study: Matrix Multiplication 28

1.3 Advanced Compiler Technology **33**

1.3.1 Automatic Parallelization 33
1.3.2 Dependence 34
1.3.3 Transformations 35
1.3.4 Historical Development 36
1.3.5 FORTRAN 90 38

1.4 Chapter Summary **39**

CHAPTER 2 Dependence, Parallelization and Vectorization **41**

2.1 Introduction **41**

2.2 Dependence and its Properties **43**

2.2.1 Load-Store Classification 44
2.2.2 Loop Iterations 45
2.2.3 Dependence and Reordering Transformations 47
2.2.4 Distance and Direction Vectors 49
2.2.5 Loop-carried and Loop-independent Dependences 51
2.2.5.1 Loop-Carried Dependence 52
2.2.5.2 Loop-Independent Dependences 55

2.3 Application to Parallelization and Vectorization **58**

2.3.1 Parallelization 58
2.3.2 Vectorization 58
2.3.3 An Advanced Vectorization Algorithm 61

2.4 Summary **66**

2.5 Historical Notes **67**

CHAPTER 3 Dependence Testing **69**

3.1 Introduction **69**

3.2 Notation **71**

3.2.1 Indices and Subscripts 71

3.2.2 Exact Tests 72
3.2.3 Complexity 72
3.2.4 Separability 73
3.3 Partitioning Subscripts **74**
3.4 ZIV and SIV Tests **77**
3.4.1 ZIV Test 77
3.4.2 SIV Tests 78
3.4.2.1 Strong SIV Subscripts 78
3.4.2.2 Weak SIV Subscripts 80
3.4.2.3 Weak-zero SIV Subscripts 81
3.4.2.4 Weak-crossing SIV Subscripts 82
3.4.3 Complex Iteration Spaces 84
3.4.4 An Exact SIV Test 84
3.5 MIV Tests **88**
3.5.1 Background 88
3.5.2 GCD Test 90
3.5.3 Banerjee Inequality 91
3.5.3.1 Trapezoidal Banerjee Inequality 96
3.5.4 Testing for All Direction Vectors 102
3.5.5 RDIV Test 103
3.6 Symbolic Tests **103**
3.7 Delta Test **104**
3.7.1 Constraints 105
3.7.2 Intersecting Constraints 106
3.7.3 Constraint Propagation 109
3.7.3.1 SIV Constraints 109
3.7.3.2 Multiple Passes 110
3.7.3.3 Improved Precision 111
3.7.3.4 Distance Vectors 112
3.7.3.5 Restricted DIV Constraints 113
3.7.4 Precision and Complexity 114
3.8 Implementation Tradeoffs **115**
3.9 Related Work **115**
3.9.1 Integer and Linear Programming 116
3.9.2 Multiple Subscript Tests 117

CHAPTER 4 Preliminary Transformations **119**
4.1 Introduction **119**
4.2 Necessary Information **122**
4.3 Loop Normalization **123**
4.4 Data Flow Analysis **126**

4.4.1 Definition-Use Chains 126
4.4.2 Dead Code Elimination 129
4.4.3 Constant Propagation 130
4.4.4 Static Single-Assignment Form 132

4.5 Induction Variable Substitution **134**

4.6 Summary **143**

CHAPTER 5 Transformations and Vector Code Generation **145**

5.1 Loop Interchange **145**

5.1.1 Introduction 145
5.1.2 Safety of Loop Interchange} 147
5.1.3 Effects of Loop Interchange 149
5.1.4 Loop Interchange and Vectorization 151
5.1.4.1 Loop Shifting 151
5.1.4.2 General Loop Interchange 154
5.1.4.3 A Loop Selection Heuristic 156

5.2 Recurrence-Breaking Transformations **158**

5.2.1 Scalar Expansion 159
5.2.2 Forward Substitution 163
5.2.3 Scalar and Array Renaming 163
5.2.4 Node Splitting 166
5.2.5 Index-set Splitting 168
5.2.5.1 Threshold Analysis 168
5.2.5.2 Loop Peeling 169
5.2.5.3 Section-based Splitting 170
5.2.5.4 Run-time Symbolic Resolution 172
5.2.6 Loop Skewing 173
5.2.7 Pattern Recognition 176
5.2.8 Transformations in a Vector Code Generator 179

CHAPTER 6 Code Generation for Parallel Computers **181**

6.1 Loop Interchange and Parallelization **181**

6.2 Single-Loop Methods **184**

6.2.1 Tiling 185
6.2.2 Privatization 185
6.2.3 Alignment 185
6.2.4 Code Replication 188

6.3 Perfect Loop Nests **192**

6.3.1 Simple Loop Selection 192
6.3.2 Code Generation with Loop Skewing 196
6.3.3 Nests with Fixed Distance Vectors 199

6.4 Fusion and Distribution **201**
6.4.1 General Approach 201
6.4.2 Single-Loop Methods 202
6.4.2.1 Typed Fusion 207
6.4.2.2 Unordered and Ordered Typed Fusion 214
6.4.3 Multilevel Loop Fusion 216
6.4.4 Profitability-Based Methods 220
6.4.4.1 Memory-Based Code Generation 221
6.5 A Code Generation Algorithm **223**
6.6 An Extended Example **229**

CHAPTER 7 Control Dependence **231**
7.1 Introduction **231**
7.2 If Conversion **232**
7.2.1 Forward Branches 232
7.2.2 Branch Classification 235
7.2.3 Exit Branches 236
7.2.4 Backward Branches 240
7.2.5 Iterative Dependences 244
7.2.6 Conversion to WHILE Loops 248
7.2.7 Simplification 248
7.2.8 IF reconstruction 252
7.2.9 Interaction with Other Transformations 252
7.3 Control Dependence **252**
7.3.1 Control Dependence Construction 255
7.3.2 Impact on Code Generation 255

CHAPTER 8 Compiler Memory-Hierarchy Management **257**
8.1 Introduction **257**
8.2 Scalar Register Allocation **258**
8.2.1 Scalar Replacement 258
8.2.2 Unroll-and-Jam 258
8.3 Vector Register Allocation **258**
8.4 Cache Management **258**
8.4.1 Loop Interchange 258
8.4.2 Blocking 258
8.5 Effectiveness **258**
8.6 Summary **258**

CHAPTER 9 Scheduling **259**

9.1 List Scheduling **259**

9.2 Software Pipelining **259**

9.3 Control Flow **259**

CHAPTER 10 Interprocedural Analysis and Optimization **261**

10.1 Interprocedural Analysis **261**

10.1.1 Interprocedural Problems 261
10.1.2 Algorithms 261
10.1.3 Advanced Methods 262

10.2 Interprocedural Optimization **262**

10.2.1 Inline Substitution 262
10.2.2 Procedure Cloning 262
10.2.3 Hybrid Optimizations 262

CHAPTER 11 Compiling High-Performance Fortran **263**

11.1 Introducation to HPF **263**

11.2 Basic Loop Compilation **263**

11.2.1 Loop Dependence Analysis 263
11.2.2 Iteration Partitioning 263
11.2.3 Communication Generation 263

11.3 Optimization **264**

11.3.1 Communication Vectorization 264
11.3.2 Overlapping Communication and Computation 264
11.3.3 Pipelining 264
11.3.4 Identification of Recurrences 264
11.3.5 Storage Management 264

11.4 Interprocedural Optimization **264**

11.5 Summary **264**

CHAPTER 12 Case Studies **265**

12.1 The Rice Parallel Fortran Converter **265**

12.2 The Ardent Titan Compiler **265**

12.3 The Rice Fortran D Compiler **265**

APPENDIX A Fundamentals of Fortran 90 **267**

A.1 Array Assignment **267**

A.2 Array Sections **268**

A.3 Parallel Loops **269**

A.4 Conditional Assignment **270**

A.5 Library Functions **270**