

Contents

Preface	xvii
Foreword	xix
Acknowledgments	xxi
1 Introduction	1
1.1 Scope of the Book	1
1.2 Instruction-Level Parallelism	3
1.3 Outline of Topics	5
2 Overview of ILP Architectures	9
2.1 Historical Perspective	9
2.2 Superscalar and VLIW Machines	13
2.3 Early ILP Architectures	15
2.4 ILP Architectures in the 80's	20
2.5 ILP Architectures in the 90's	24
2.6 Itanium	35
2.6.1 The EPIC Philosophy	36
2.6.2 Itanium Architecture	38
3 Scheduling Basic Blocks	43
3.1 Introduction	43
3.2 Basic Concepts	44

3.3	Unlimited Resources	47
3.3.1	ASAP Algorithm	47
3.3.2	ALAP Algorithm	49
3.4	Limited Resources	50
3.4.1	List Scheduling	50
3.4.2	Linear Analysis	52
3.5	An Example	53
3.6	More Algorithms	58
3.6.1	Critical Path Algorithm	58
3.6.2	Restricted Branch and Bound Algorithm	62
3.6.3	Force-Directed Scheduling	66
3.7	Limited Beyond Basic Block Optimization	73
4	Trace Scheduling	79
4.1	Introduction	79
4.2	Basic Concepts	82
4.2.1	Program Model	82
4.2.2	Traces	85
4.2.3	Dependence	87
4.2.4	Schedules	87
4.2.5	Program Transformation	89
4.3	Traces without Joins	90
4.4	General Traces	100
4.5	Trace Scheduling Algorithm	109
4.6	Picking Traces	112
5	Percolation Scheduling	117
5.1	Introduction	117
5.2	The Core Transformations	118
5.2.1	Delete Transformation	120
5.2.2	Move-op Transformation	121
5.2.3	Move-test Transformation	122
5.2.4	Unify Transformation	123

5.3	Remarks	126
5.3.1	Termination	127
5.3.2	Completeness	128
5.3.3	Confluence	129
5.4	Extensions	129
5.4.1	Migrate Transformation	129
5.4.2	Trailblazing	130
5.4.3	Resource-Constrained Percolation Scheduling	131
6	Modulo Scheduling	133
6.1	Introduction	133
6.2	Unrolling	135
6.3	Preliminaries	138
6.4	Modulo Scheduling Algorithm	140
6.4.1	Remarks	146
	Sufficiency of simple cycles	146
	Infeasibility of MII	148
6.4.2	Limitations	148
6.5	Modulo Scheduling with Conditionals	150
6.5.1	Hierarchical Reduction	150
6.5.2	Enhanced Modulo Scheduling	152
6.5.3	Modulo Scheduling with Multiple Initiation Intervals	155
6.6	Iterative Modulo Scheduling	156
6.6.1	The Algorithm	157
	Determining Scheduling Priority	159
	Determining Earliest Start Time	160
	Determining Candidate Time Slots	160
6.7	Optimizations	161
6.7.1	Modulo Variable Expansion	161
6.7.2	Using Loop Unrolling to Enhance Modulo Scheduling	162

7	Software Pipelining by Kernel Recognition	167
7.1	Introduction	167
7.1.1	Basic Idea	169
7.2	The URPR Algorithm	170
7.3	OPT: Optimal Loop Pipelining of Innermost Loops . . .	174
7.4	General Handling of Conditionals	180
7.4.1	Perfect Pipelining	180
	Compaction	181
	The Algorithm	183
7.4.2	Enhanced Pipeline-Percolation Scheduling . . .	193
7.4.3	Optimal Software Pipelining with Control Flow	199
7.5	Nested Loops	200
7.6	Procedure Calls	201
8	Epilogue	205
	Bibliography	209
	Index	247

Instruction Level Parallelism

Aiken, A.; Banerjee, U.; Kejariwal, A.; Nicolau, A.

2016, XXI, 255 p. 78 illus., 30 illus. in color., Hardcover

ISBN: 978-1-4899-7795-3