

# Contents

<b>1</b>	<b>Introduction and Overview</b>	<b>9</b>
1.1	Introduction . . . . .	9
1.2	Overview . . . . .	10
<b>2</b>	<b>Synchronous Neural Network Models</b>	<b>13</b>
2.1	Neural Network Taxonomy . . . . .	13
2.2	Neural Network Classification . . . . .	15
2.2.1	Topology and Dynamics in the Recall Phase . . . . .	15
	Feed Forward Nets . . . . .	15
	Recurrent Nets . . . . .	16
2.2.2	Neural Network Learning . . . . .	17
	Supervised Learning . . . . .	17
	Unsupervised Learning . . . . .	17
2.3	Neural Network Models . . . . .	18
	Jordan- and Elman Nets . . . . .	18
	Recurrent Backpropagation . . . . .	18
	Cascade Correlation Nets . . . . .	18
	Self-Organizing Maps . . . . .	18
	Counterpropagation . . . . .	19
	Hopfield Nets . . . . .	19
	Boltzmann-Machines . . . . .	19
	BAM . . . . .	20
	Adaptive Resonance Theory . . . . .	20
	Neocognitron . . . . .	20
	Other Neural Network Models . . . . .	20
2.4	Generalization of Neural Network Models . . . . .	20
<b>3</b>	<b>Massively Parallel Hardware Concepts</b>	<b>23</b>
3.1	SIMD Architectures . . . . .	23
3.2	MIMD Architectures . . . . .	23
3.3	Target Architecture . . . . .	24

<b>4</b>	<b>State of the Art</b>	<b>27</b>
4.1	State of the Art in Parallel Simulation of ANNs	27
4.1.1	Hardware Implementations	27
4.1.2	Software Implementations	27
	MIMD Simulations	28
	Pattern Parallel Simulation	28
	Systolic Implementations	28
	Identify Equivalence Classes	29
	Integrated Systems	29
	SIMD Simulations	30
4.2	Significance of the presented approach	30
<b>5</b>	<b>Scheduling and Mapping</b>	<b>31</b>
5.1	Introduction	31
5.1.1	Scheduling	31
5.1.2	Mapping	32
5.1.3	Combining Scheduling and Mapping	33
5.2	Related Work	33
5.2.1	Static Scheduling	33
	Operation Scheduling	34
	Overviews and Frameworks to Task Scheduling	34
	Critical Path Method	35
	Unbounded Number of Processors	36
	Bounded Number of Processors	37
	Limited Interconnections	39
	Other Approaches	41
	Scalable Tasks	42
5.2.2	Drawing of Directed Acyclic Graphs	43
	Bandwidth Reduction	44
	Reorder Algorithms	44
5.2.3	Graph Partitioning	47
	Bipartitions	48
	Multi Partitions	51
5.2.4	Static Mapping	54
	Regular Graph (Families)	54
	General Approaches	55
	Agglomerative Approaches	55
	Divisive Approaches	56
	Configurative Approaches	58
5.2.5	Dynamic Scheduling and Mapping	59
	Hierarchical approaches	59
	Mapping of Geometrical Oriented Graphs	60
5.2.6	Mapping Neural Networks	61

<b>6</b>	<b>A Conceptual View</b>	<b>65</b>
6.1	Introduction . . . . .	65
	Construction . . . . .	65
	Analysis . . . . .	65
	Mapping . . . . .	65
	Simulation . . . . .	66
6.2	Construction . . . . .	66
6.3	Analysis of the Network Graph . . . . .	67
6.4	Mapping . . . . .	67
6.5	Outline of the Simulation Model . . . . .	71
<b>7</b>	<b>Analysis of the Neural Network Graph</b>	<b>73</b>
7.1	Compression . . . . .	73
7.2	Group Collection . . . . .	76
<b>8</b>	<b>Mapping</b>	<b>79</b>
8.1	Problem Specification . . . . .	80
8.1.1	Task Graph . . . . .	80
8.1.2	Processor Graph . . . . .	80
8.1.3	Cost Function . . . . .	80
8.1.4	Target Architecture . . . . .	81
8.1.5	Computation Model . . . . .	82
8.1.6	Execution Model . . . . .	83
8.2	The Algorithm . . . . .	84
8.2.1	Subproblems . . . . .	84
8.2.2	Overview . . . . .	84
8.2.3	Decomposition of the Problem Graph . . . . .	87
8.2.4	Construction of the Schedule Graph for Non-Cyclic Groups . . . . .	88
	Overview . . . . .	88
	Introduction . . . . .	88
	Random Networks . . . . .	91
	Random Schedule Experiments . . . . .	94
	Smallest Layer Scheduling Algorithms . . . . .	100
	Border Node Experiments . . . . .	104
	Smallest Layer Schedule Experiments . . . . .	108
	Cutting oversized nodes . . . . .	110
8.2.5	Reordering Nodes of the Schedule graph . . . . .	119
	Overview . . . . .	119
	Introduction . . . . .	119
	Quality Measure . . . . .	120
	Quality of Schedule Graphs . . . . .	122
	Reordering Algorithms . . . . .	123
	Experiments . . . . .	125

8.2.6	Number of Processors . . . . .	127
8.2.7	Construction of the Virtual Processor Graph for Non-Cyclic Groups . . . . .	130
8.2.8	Partitioning the Cyclic Group . . . . .	130
	Introduction . . . . .	131
	Overview . . . . .	131
	Algorithm for Partitioning Graphs . . . . .	132
	Decomposing Grids with Random Load . . . . .	137
	Experiments concerning Random Graphs . . . . .	142
	Conclusion on Partitioning . . . . .	144
8.2.9	Mapping of Virtual Processor Graphs . . . . .	144
8.2.10	Mapping onto Regular Processor Graphs . . . . .	150
	Generating Regular Processor Graphs . . . . .	150
	Mapping Virtual Processor Graph onto Regular Processor Graphs . . . . .	150
8.2.11	Construction of Adapted Processor Graphs . . . . .	150
	Degree Reduction Procedure . . . . .	150
	Edge Labeling . . . . .	151
	Edge Subset Selection . . . . .	155
	Create Unique Tree . . . . .	158
	Insert Remaining Edges . . . . .	159
	Abbreviate Long Paths . . . . .	160
	Example . . . . .	160
	Experiments . . . . .	162
<b>9</b>	<b>Simulation</b> . . . . .	<b>165</b>
9.1	Introduction . . . . .	165
9.2	Simulation Model . . . . .	166
9.2.1	Recurrent Networks . . . . .	167
9.2.2	Optimization of the Simulation Model . . . . .	168
9.2.3	Networks without loops . . . . .	170
9.2.4	Learning . . . . .	170
	Supervised Learning . . . . .	170
	Unsupervised Learning . . . . .	171
<b>10</b>	<b>Implementation</b> . . . . .	<b>173</b>
10.1	Components . . . . .	173
10.1.1	Talker . . . . .	173
	Store-and-Forward Deadlock Free Routing . . . . .	174
10.1.2	Organizer . . . . .	178
10.1.3	Simulator . . . . .	178
10.2	Software Architecture . . . . .	179
<b>11</b>	<b>Conclusion and Further Work</b> . . . . .	<b>181</b>

# List of Figures

2.1	Multi Layer Perceptron . . . . .	16
6.1	Overview . . . . .	70
7.1	Collecting blocks to equivalence classes . . . . .	74
7.2	Assigning blocks to four groups . . . . .	76
8.1	Unbalanced edge probabilities ( $n = 50$ ) . . . . .	92
8.2	Smallest Layer Schedule: (SLS) . . . . .	101
8.3	Smallest Layer Schedule (Schedule Interval, Border Nodes): SLS(SI,BN) . . . . .	103
8.4	Example neural net, the width of the nodes corresponds to the <b>number of neurons</b> . . . . .	113
8.5	Schedule of the example net, the width corresponds to the <b>load</b> of each node . . . . .	114
8.6	Schedule of the example net, oversized nodes cut, the width corresponds to the load of each node . . . . .	115
8.7	Schedule of a small graph, oversized nodes cut, all nodes rescheduled . . . . .	117
8.8	Example of communicating nodes in the schedule graph . . . . .	120
8.9	Determining the reorder value of a single node . . . . .	124
8.10	Assigning node $u$ to set $S_j$ . . . . .	133
8.11	Selecting new initial nodes of set $S_i$ . . . . .	134
8.12	Pacman algorithm, <i>candidates selection/candidate test</i> procedure . . . . .	136
8.13	Partitioning into 4 Sets . . . . .	137
8.14	Partitioning into 16 Sets . . . . .	138
8.15	Cycle edges and back edges . . . . .	152
8.16	Edge labeling and sorting step . . . . .	153
8.17	Example graph, containing nodes of high degree . . . . .	154
8.18	Breadth first search trees, rooted at $o$ and $s$ . . . . .	155
8.19	Edge subset selecting step . . . . .	156
8.20	Unique tree creating step . . . . .	158
8.21	Inserting remaining edges step . . . . .	159
8.22	Abbreviating long paths step . . . . .	160
8.23	Maximum degree reduced to 4 . . . . .	161
10.1	Processorgraph and corresponding buffer-graph . . . . .	175

10.2	Extension of the approach by dynamic buffer-queues (Fig. b): pools of free buffers are omitted . . . . .	177
10.3	Architecture of the simulator . . . . .	178
10.4	Overview of the entire system . . . . .	179

# List of Tables

8.1	Groups of Random Graphs: Overview . . . . .	93
8.2	Random, ASAP and ALAP schedules compared to lower bound . . . . .	95
8.3	Average Schedule Profiles . . . . .	97
8.4	Random Schedule compared to lower bound, ASAP and ALAP . . . . .	98
8.5	SLS(BN,SI) Algorithms: Overview . . . . .	104
8.6	Border Node Results for Different Bounds . . . . .	105
8.7	Schedule profiles for different bounds . . . . .	106
8.8	Results of different algorithms, Part I . . . . .	108
8.9	Results of different algorithms, Part II . . . . .	109
8.10	Schedule profiles for all considered algorithms and one graph group . . . . .	111
8.11	Schedule profiles for three algorithms and all considered graph groups . . . . .	112
8.12	Results of different algorithms, only cut portions are rescheduled . . . . .	116
8.13	Results of different algorithms, all nodes are rescheduled . . . . .	118
8.14	Schedule sequence compared to random sequence . . . . .	122
8.15	Iterating the median reordering algorithms on <i>randomized neurons, balanced edge prob.</i> graphs . . . . .	126
8.16	Iterating the balanced forces reordering algorithms on randomized neurons, balanced edge prob. graphs . . . . .	127
8.17	Best results . . . . .	128
8.18	Partitioning $8 \times 8$ and $16 \times 16$ grids with random load and random communication . . . . .	139
8.19	Partitioning random graphs . . . . .	143
8.20	Results of the mapping algorithms, applied to map a graph on itself . . . . .	147
8.21	Results of the mapping algorithms . . . . .	149
8.22	Results of the degree reduction algorithm compared to mapping onto Generalized DeBruijn Nets, by applying the appropriate algorithm . . . . .	163