

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Reconfigurable Computing . . . . .	2
1.2	Contribution of the Thesis . . . . .	4
1.3	Abstracting Layered Approach . . . . .	5
1.4	Outline of the work . . . . .	5
<b>2</b>	<b>Reconfigurable Computing</b>	<b>7</b>
2.1	Introduction . . . . .	7
2.1.1	Evolution . . . . .	7
2.1.2	Makimoto's Wave . . . . .	11
2.1.3	Remainder of the Chapter . . . . .	12
2.2	Technical Aspects . . . . .	12
2.2.1	Reconfigurable versus Programmable . . . . .	13
2.2.2	Granularity . . . . .	14
2.2.3	Field Programmable Gate Arrays . . . . .	15
2.2.4	Programming FPGAs . . . . .	20
2.2.5	Run-Time Reconfiguration . . . . .	22
2.2.6	Coupling . . . . .	28
2.3	Fields of Application . . . . .	29
2.3.1	ASIC Design . . . . .	29
2.3.2	Replacement of Dedicated Circuits . . . . .	30
2.3.3	Adaptive Processing . . . . .	31
2.4	Design Approaches for Reconfigurable Systems . . . . .	34
2.4.1	Execution Environments/Architectures . . . . .	34
2.4.2	Placement/Scheduling Methods . . . . .	37
2.4.3	Comprehensive Design Systems/Design Methods . . . . .	39
2.4.4	Miscellaneous Concepts . . . . .	40
2.5	Lesson Learned . . . . .	40
2.6	Summary . . . . .	42
<b>3</b>	<b>Two-Slot Framework</b>	<b>43</b>
3.1	Introduction . . . . .	43
3.1.1	Layered Approach . . . . .	45
3.1.2	Organization of the Chapter . . . . .	45
3.2	Concept . . . . .	45
3.2.1	Problem Abstraction . . . . .	46

3.2.2	Problem Solution Strategy . . . . .	48
3.3	Run-Time Architecture . . . . .	49
3.3.1	Fundamental Design Constraints . . . . .	50
3.3.2	Intermodule Communication . . . . .	51
3.4	Partitioning . . . . .	53
3.4.1	Simple Temporal Partitioning . . . . .	53
3.4.2	Spectral Based Partitioning . . . . .	54
3.5	Scheduler . . . . .	59
3.6	Experiment . . . . .	61
3.6.1	Proof of Concept Implementation . . . . .	61
3.6.2	Cryptography Example . . . . .	63
3.7	Extensions . . . . .	65
3.7.1	Low Power Considerations . . . . .	65
3.7.2	The Two Slot Framework as an IP Core . . . . .	69
3.8	Lesson Learned . . . . .	70
3.9	Related Work . . . . .	71
3.10	Summary . . . . .	73
<b>4</b>	<b>Specification Graph Approach for Reconfigurable Fabrics</b>	<b>75</b>
4.1	Introduction . . . . .	75
4.1.1	Layered Approach . . . . .	76
4.1.2	Organization of the Chapter . . . . .	76
4.2	Concept . . . . .	77
4.2.1	Problem Abstraction . . . . .	77
4.2.2	Background . . . . .	78
4.2.3	Problem Solution . . . . .	79
4.3	Problem Graph . . . . .	80
4.4	Architecture Graph . . . . .	82
4.5	Mapping . . . . .	83
4.5.1	Basic Mapping Edges . . . . .	83
4.5.2	Extensions . . . . .	84
4.6	Design Space Exploration . . . . .	85
4.7	Experiment . . . . .	91
4.8	Lesson Learned . . . . .	94
4.9	Related Work . . . . .	95
4.10	Summary . . . . .	96
<b>5</b>	<b>Reconfiguration Port Scheduling</b>	<b>97</b>
5.1	Introduction . . . . .	97
5.1.1	Layered Approach . . . . .	98
5.1.2	Remainder of the Chapter . . . . .	99
5.2	Concept . . . . .	99
5.2.1	Execution Environment . . . . .	99
5.2.2	Problem Abstraction . . . . .	100

5.2.3	Reconfiguration Port Scheduling . . . . .	101
5.2.4	Parallel Machine Problems with a Single Server . . . . .	104
5.3	Aperiodic Task Scheduling . . . . .	104
5.3.1	Synchronous Arrival . . . . .	104
5.3.2	Asynchronous Arrival . . . . .	108
5.3.3	Experimental Results . . . . .	111
5.4	Fixed Priority Periodic Task Scheduling . . . . .	111
5.4.1	Schedulability Analysis . . . . .	111
5.4.2	A Server for <i>Full Load of Slots</i> Sections . . . . .	112
5.4.3	Resource Access Protocol for <i>Full Reconfiguration Capacity</i> Sections . . . . .	114
5.4.4	DM + SS + PIP Schedulability Test . . . . .	116
5.4.5	Experiment . . . . .	116
5.5	Caching . . . . .	117
5.5.1	Offline caching methods . . . . .	119
5.5.2	Dynamic/On-line Caching Methods . . . . .	122
5.5.3	Combination of the Methods . . . . .	124
5.5.4	Implementation . . . . .	125
5.6	Experiment . . . . .	125
5.7	Lesson Learned . . . . .	129
5.8	Related Work . . . . .	130
5.9	Summary . . . . .	132
<b>6</b>	<b>Algorithmic Skeletons for Dynamic Reconfiguration</b>	<b>133</b>
6.1	Introduction . . . . .	133
6.1.1	Layered Approach . . . . .	136
6.1.2	Remainder of the Chapter . . . . .	136
6.2	Concept . . . . .	136
6.2.1	Algorithmic Skeletons . . . . .	137
6.2.2	Application for Reconfigurable Computing . . . . .	139
6.3	Run-time Execution Environment . . . . .	140
6.4	Stream Parallelism . . . . .	144
6.4.1	Farm Paradigm . . . . .	144
6.4.2	Pipeline Paradigm . . . . .	145
6.5	Dynamic Reconfiguration . . . . .	147
6.5.1	Dynamic Reconfiguration on a Tile-Based Execution Environment . . . . .	147
6.5.2	Dynamic Reconfiguration on a Skeleton-centric Execution Environment . . . . .	148
6.6	Experiment . . . . .	151
6.6.1	Cryptography Experiment . . . . .	151
6.6.2	Application-centric . . . . .	153
6.7	Lesson Learned . . . . .	157
6.8	Related Work . . . . .	158
6.9	Summary . . . . .	159

<b>7 Conclusion and Outlook</b>	<b>161</b>
7.1 Conclusion . . . . .	161
7.2 Outlook . . . . .	164
<b>A The Design Tool Part-E</b>	<b>165</b>
<b>Author's Own Publications</b>	<b>169</b>
<b>Bibliography</b>	<b>173</b>

# List of Figures

1.1	Performance vs. Flexibility . . . . .	2
1.2	Layered Model . . . . .	5
2.1	Makimoto's Wave . . . . .	11
2.2	Typical layout of a modern FPGA . . . . .	16
2.3	Schematic view of a configurable logic block of an FPGA . . . . .	17
2.4	Slice of a Xilinx Virtex 4 FPGA . . . . .	18
2.5	Busmacro of the Xilinx Application Note 290 . . . . .	26
2.6	Narrow busmacro of the Xilinx early access design flow . . . . .	27
2.7	Coupling of reconfigurable devices . . . . .	28
2.8	The Erlangen Slot Machine . . . . .	36
3.1	Layered approach for the two-slot framework . . . . .	45
3.2	Reconfiguration ( <i>RT</i> ) and execution ( <i>EX</i> ) phase, simple example. . . . .	46
3.3	Benefit of partial reconfiguration . . . . .	46
3.4	Scheduling example of <i>EX</i> and <i>RT</i> phases . . . . .	47
3.5	Draft of the two slot architecture. . . . .	49
3.6	Communication focused floorplan of the architecture . . . . .	51
3.7	Simple partitioning, encapsulation of cycles . . . . .	54
3.8	Partitioning continued . . . . .	55
3.9	Including a NOP-node . . . . .	55
3.10	ASAP schedule of a data flow graph . . . . .	56
3.11	2D spectral placement of the data flow graph of Fig. 3.10 . . . . .	56
3.12	Nodes of ASAP level 1 are assigned to processing elements . . . . .	57
3.13	Result of the ASAP and spectral based node assignment . . . . .	57
3.14	Complete schedule for a coarse grained device . . . . .	57
3.15	Clustering of a task graph . . . . .	58
3.16	Sequence diagram of the slots and the controller entities . . . . .	59
3.17	Layout of the two slot architecture on a Virtex-II Pro FPGA . . . . .	62
3.18	Layout of the two slot environment on a Virtex-4 FPGA . . . . .	64
3.19	Example of produced cluster formation . . . . .	68
4.1	Layered model . . . . .	76
4.2	Schematic of an exemplary runtime environment . . . . .	77
4.3	Task dependence graph $G$ . . . . .	80
4.4	Task dependence graph and Gantt chart . . . . .	81
4.5	Problem graph with intervals/life cycles. . . . .	82

4.6	Architecture graph . . . . .	83
4.7	Mapping Edges . . . . .	84
4.8	Mapping Edges II . . . . .	85
4.9	Specification graph . . . . .	86
4.10	Gantt chart of a schedule of Fig. 4.9 . . . . .	86
4.11	Task graph with communication and reconfiguration phases in ASAP and ALAP ordering . . . . .	88
4.12	Image filtering using limited resources . . . . .	92
4.13	Gantt chart of the image filtering example . . . . .	93
5.1	Layered Approach . . . . .	98
5.2	Execution environment having homogeneous slots . . . . .	100
5.3	Example occupation of three slots. . . . .	101
5.4	Scheduling according to $d$ (left) and $d^*$ (right), synchronous arrival times. . . . .	102
5.5	Scheduling two tasks according to $d^*$ . . . . .	103
5.6	Scheduling task sets according to EDD . . . . .	104
5.7	Full load of slots . . . . .	105
5.8	EDD can fail to produce a feasible schedule. . . . .	108
5.9	EDF schedule on a three slot machine using preemption. . . . .	109
5.10	Full reconfiguration capacity of an EDF schedule . . . . .	109
5.11	Fixed priority example. . . . .	112
5.12	Server for <i>fls</i> : Worst Case. . . . .	113
5.13	Blocking Time for <i>frc</i> : Worst Case. . . . .	115
5.14	Simulator . . . . .	117
5.15	Caching: Priority Based Slot Reservation . . . . .	119
5.16	Consecutive Task Combination with modification of the release times of tasks $\tau_1$ and $\tau_2$ . . . . .	120
5.17	Scheduling Look Back . . . . .	122
5.18	Scheduling Look Ahead . . . . .	123
5.19	Applying different scheduling algorithms . . . . .	124
5.20	Conveyer belt . . . . .	126
5.21	Set of conveyer belts connected to a central processing unit. . . . .	126
5.22	Experimental set-up on the Erlangen Slot Machine . . . . .	127
5.23	The execution environment implemented on the ESM . . . . .	128
6.1	Layered model . . . . .	136
6.2	Two tile-based run-time execution environments . . . . .	141
6.3	Skeleton-centric execution environment . . . . .	142
6.4	Farm parallelism . . . . .	144
6.5	The pipeline paradigm . . . . .	144
6.6	Two possible execution schemes of applications using the farm skeleton. . . . .	145
6.7	Two implementations of a pipe skeleton with different area requirements. . . . .	146
6.8	Run-time reconfiguration of different skeletons . . . . .	148
6.9	Dynamic reconfiguration on a skeleton-centric environment . . . . .	149

6.10	Scheduling example of a farm and pipe skeleton . . . . .	150
6.11	Triple DES given as a pipe and farm skeleton . . . . .	151
6.12	Skeleton-centric platform on a Xilinx Virtex-4 FPGA . . . . .	153
6.13	Mapping the triple DES application . . . . .	154
6.14	Xilinx FPGA editor output of the triple DES application . . . . .	155
6.15	Detailed view of the farm circuit . . . . .	156
A.1	Screenshot of Part-Y . . . . .	166
A.2	Reduced class diagram of the model. . . . .	167
A.3	Screenshot of Part-E . . . . .	168