

Contents

1	Introduction	7
1.1	Controller Design	8
1.2	Related Work	10
1.3	CHaRy: The C-LAB Hard Real-Time System	11
1.4	Overview of Contents	13
2	Longest-Executable-Path Search	17
2.1	Introduction	17
2.2	Related Work on Timing Analysis	20
2.3	False Path Problem Heuristic	22
2.3.1	Control Flow Graph	23
2.3.2	Handling of Loops	24
2.3.3	Handling of Function Calls	25
2.3.4	The Algorithm	26
2.3.5	Handling of Caching and Pipelining	30
2.4	Test-Case Evaluation	31
2.5	Definition-Use Chains and K -Longest Path Search	32

2.6	Conclusions on Timing Analysis	34
3	Allocation of Periodic Hard Real-Time Tasks	35
3.1	Introduction	35
3.2	Static Hard-Real-Time Allocation	39
3.2.1	Basic Assumptions	40
3.2.2	Optimal Algorithms	42
3.2.3	Non-Guided Search Techniques	42
3.2.3.1	Steepest Descent	44
3.2.3.2	Tabu Search	45
3.2.3.3	Simulated Annealing	46
3.2.3.4	Threshold Accepting	47
3.2.3.5	Genetic Algorithms	48
3.2.4	Constructive Heuristics	50
3.2.4.1	List Processing	51
3.2.4.2	The Slack Method	53
3.2.4.2.1	Graph Reduction	54
3.2.4.2.2	Transaction Mapping	59
3.2.4.2.3	LCM Handling	61
3.2.4.2.4	Replications and Attached Tasks	62
3.3	Test-Case Evaluation	63
3.3.1	Evaluation of the New Configuration Concept	64
3.3.2	Evaluation of Different Non-Guided-Search Methods	65
3.3.3	Evaluation of List Processing	66

3.3.4	Evaluation of the Slack Method	68
3.4	Conclusions on Allocation	70
4	Schedulability Analysis of Event-Driven Systems	71
4.1	Introduction	71
4.2	Related Work on Schedulability Analysis	74
4.3	The Schedulability Test	81
4.3.1	Mapping the Precedence Graph to DMS	83
4.3.2	Worst-Case-Response-Time Computation	84
4.3.3	I/O Jitter Reduction and Cyclic Executives	92
4.3.4	Implementation Remarks	93
4.3.5	Example	95
4.4	Test-Case Evaluation	97
4.5	Conclusions on Schedulability Analysis	99
5	Conclusions	101
5.1	Summary of Contents	101
5.2	Transferability	102
5.3	Open Problems and Future Work	102
	Bibliography	105
A	Example Model	115
A.1	Client-Server Model	115

List of Figures

1.1	Automation of controller design	8
1.2	CHaRy: The C-LAB Hard Real-Time System	11
2.1	A simple example program	18
2.2	Example of a control flow graph	24
2.3	Control flow graph of the example program of Fig. 2.1	26
2.4	The WCET algorithm	27
2.5	Application of the algorithm to the example program of Fig. 2.3	29
2.6	The timing-analysis process	30
2.7	A simple example program to illustrate definition-use chains	33
3.1	The advantage of the new configuration approach	36
3.2	An example task graph	41
3.3	Overview of the combination of constructive and non-guided-search heuristics including the LCM handling	50
3.4	A simple list-processing example	52
3.5	Overview of the Slack Method	53
3.6	Critical paths for the example of Fig. 3.2	56
3.7	The corresponding CP-graph of Fig. 3.6	56

3.8	Reducing the CP-graph of Fig. 3.7	58
3.9	The union of CP_1 and CP_3 , and CP_2 and CP_4	58
3.10	A CP-graph embedded into a processor subnet	59
3.11	Overview of the transaction mapping	60
3.12	LCM Handling (i.e. subnet joining)	61
4.1	Task model of an event-driven system	72
4.2	Precedence graph of a communicating task set	79
4.3	Two different schedules of the example task set of Fig. 4.2	79
4.4	Precedence graph of a communicating task set including deadlines	83
4.5	Relation of offsets and response times	87
4.6	The task set of Fig. 4.4 including offset values	88
4.7	The four different cases of computing the transaction interference	89
4.8	Deadline computation algorithm	94
4.9	Worst-case-response-time computation algorithm	95
4.10	Simulation of the example task set of Fig. 4.4	97
A.1	Client-server architecture	115
A.2	Event-driven model of the client-server architecture of Fig. A.1	116