Compositional Performance Analysis
for Complex Embedded Applications

Von der Gemeinsamen Fakultät für Maschinenbau und Elektrotechnik
der Technische Universität Carolo-Wilhelmina zu Braunschweig
zur Erlangung der Würde
eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von: Marek Jersak
aus: Bern, Schweiz

ingereicht am: 28. Mai 2004
mündliche Prüfung am: 24. Juni 2004

Referent: Prof. Dr.-Ing. Rolf Ernst
Referent: Prof. Dr.-Ing. Lothar Thiele
Vorsitzender: Prof. Dr.-Ing. Harald Michalik

2005
# Contents

1 INTRODUCTION  
1.1 Motivation  
1.2 Outline  

2 SYSTEM-LEVEL DESIGN  
2.1 Executable Specifications  
\hspace{0.5em} 2.1.1 Dataflow Process Networks  
\hspace{0.5em} 2.1.2 Globally Synchronous Models  
\hspace{0.5em} 2.1.3 Globally Asynchronous Models  
\hspace{0.5em} 2.1.4 Multi-Language Design  
\hspace{0.5em} 2.1.5 Commercial Tools  
\hspace{0.5em} 2.1.6 System-Level and Implementation Languages  
2.2 Target Architectures  
\hspace{0.5em} 2.2.1 Systems-on-Chip  
\hspace{0.5em} 2.2.2 Distributed Systems  
2.3 Performance Constraints  
2.4 Function Implementation and Integration  
\hspace{0.5em} 2.4.1 Task Activation and Scheduling  
2.5 Integration Problems  
2.6 The SPI Project  
2.7 Summary and Conclusion  

3 PERFORMANCE ANALYSIS  
3.1 Scheduling Analysis  
\hspace{0.5em} 3.1.1 Event Models  
3.2 Single-Processor Scheduling Analysis
### Contents

3.3 Homogeneous Multi-Processor Scheduling Analysis  32  
3.4 Holistic Scheduling Analysis  33  
3.5 Compositional Performance Analysis  34  
3.6 Composition Using Standard Event Models  35  
  3.6.1 Task Model  35  
  3.6.2 Output Event Model Calculation  36  
  3.6.3 Analysis Composition  39  
  3.6.4 Starting Point  40  
  3.6.5 System-Level Analysis Iteration  40  
  3.6.6 Event Stream Adaptation  41  
  3.6.7 Communication Buffers  42  
3.7 Composition Using Packet Flows and Resource Streams  43  
3.8 Performance Analysis for Complex Applications  45  
3.9 Other Performance Analysis Techniques  46  
  3.9.1 Model Checking-Based Scheduling Analysis  46  
  3.9.2 Tightly Coupled System Model and Scheduling Analysis  47  
3.10 Summary and Conclusion  48  

4 TASKS WITH MULTIPLE INPUTS  51  
4.1 AND-Activation  52  
  4.1.1 Calculation of Activating Event Functions  53  
  4.1.2 AND-Activation Incurred Delay and Backlog  55  
  4.1.3 AND-activation for periodic with jitter Input Event Models  56  
  4.1.4 Example  58  
4.2 OR-Activation  60  
  4.2.1 Calculation of Activating Event Functions  60  
  4.2.2 OR-activation for periodic with jitter Input Event Models  61  
  4.2.3 Example  65  
  4.2.4 OR-activation for sporadic with jitter Input Event Models  66  
4.3 Combination of AND- and OR-Activation  66  
4.4 Multiple Outputs  67  
4.5 Summary and Conclusion  67
5 RATE TRANSITIONS BETWEEN TASKS

5.1 Fixed Data Rates
5.1.1 Token Functions
5.1.2 Calculation of Activating Event Functions
5.1.3 Data Rate Transitions for periodic with jitter Event Models
5.1.4 Example
5.1.5 Data Rate Transitions for sporadic with jitter Input Event Models
5.1.6 Rate Transition Incurred Delay and Backlog
5.1.7 Special Cases

5.2 Data Rate Intervals
5.2.1 Data Rate Transitions for periodic with jitter Event Models
5.2.2 Data Rate Transitions for sporadic with jitter Input Event Models
5.2.3 Rate Interval Transition Incurred Token Delay and Backlog

5.3 Combination with Multiple Inputs
5.4 Chaining of Rate Transitions, Multiple Inputs and EAFs
5.5 Combined Token Buffering
5.6 Summary and Conclusion

6 CYCLIC TASK DEPENDENCIES

6.1 Single-Rate Cycles
6.2 Analysis Idea
6.3 Cycles with one initial token
6.3.1 Buffer Calculation
6.4 Cycles with two or more initial tokens
6.4.1 Buffer Calculation
6.5 Analyzability Condition
6.6 Cycles with more available than required initial tokens
6.7 Cycles with fewer available than required initial tokens
6.8 Self-Cycles
6.9 System Startup
6.10 Nested Cycles
6.11 Cycles with Multiple Inputs
6.12 Multi-Rate Cycles
6.13 Summary and Conclusion

7 SYSTEM-LEVEL ANALYSIS EXAMPLE

8 CONTEXT-AWARE ANALYSIS

8.1 Intra Event Stream Contexts
8.2 Inter Event Stream Contexts
8.3 Combination of Contexts
8.4 Example
8.5 Summary and Conclusion

9 PERFORMANCE ANALYSIS FOR SIMULINK

9.1 Simulink Model of Computation
9.2 Code Generation from Simulink
  9.2.1 Tick Scheduling
  9.2.2 Rate Monotonic Scheduling
9.3 Simulink Code Generation Issues
  9.3.1 Multi-Processor Implementation
  9.3.2 Multi-Language Design
9.4 Model Relaxation
  9.4.1 Single-Rate Designs
  9.4.2 Multi-Rate Designs
  9.4.3 Implications for Scheduling
9.5 Scheduling Analysis
  9.5.1 Schedulability Condition
  9.5.2 Analysis Approach
  9.5.3 Triggered and Enabled Blocks
9.6 Summary and Conclusion

10 SUMMARY AND OUTLOOK

List of Figures
List of Tables
Bibliography