Peter Gliwa

Embedded Software Timing

Methodology, Analysis and Practical Tips with a Focus on Automotive

Springer

Peter Gliwa
Gliwa GmbH
Weilheim, Germany

ISBN 978-3-030-64143-6 ISBN 978-3-030-64144-3 (eBook)
https://doi.org/10.1007/978-3-030-64144-3

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Embedded software makes up only a comparatively small part of the larger topic of computer science. Within this, the topic of "timing" focuses only on one specific aspect. So, is the topic of "Embedded Software Timing" one that is only relevant to a few experts?

At this very moment, billions of embedded systems are in use worldwide. Embedded software is running on every one of those devices with each system having its own set of timing requirements. If those timing requirements are not met due to a software error, the range of possible outcomes varies enormously. Depending on the product and situation, this may range from not being noticed, to being an annoyance for the user, to costing lives.

A good understanding for the timing challenges of embedded systems enables the development of better, more reliable embedded software. In addition, it is not only safety and reliability that can be improved. There are also considerable cost savings to be had across the entire development life cycle. These are not purely theoretical, as the practical examples in Chapter 6 highlight. The potential for cost savings extends across the various phases of development:

- Early consideration for the issue of timing in the design of an embedded system and its software helps decisively in increasing development efficiency and prevents timing problems from arising in the first place.

  See, among others, Sections 3.3, 6.2, and 8.1 and Chapter 9.
- Timing analysis can save time and money if the correct timing analysis technique for the given application is used. Chapter 5 provides an overview of the different techniques. Each has its own phases that describe its functional principle and workflow, highlighting use cases and limitations. In addition, an interview with one or two experts in the respective domain completes these descriptions. This livens up the topic and provides some objectivity. If the milk has already spilled—that is, if a project is already facing acute problems—troubleshooting often resembles the search for a needle in a haystack, especially in the case of timing problems. Here, too, the use of the optimal timing analysis technique delivers decisive advantages.
- Automated tests help to save costs: this is a truism. Unfortunately, existing testing all too often lacks explicit timing-related tests and focuses only on functional aspects. Section 9.6 provides recommendations in the form of concrete measures

to counteract this by ensuring embedded software timing can be verified in a well-structured manner.

- If a project reaches the point where the processor is permanently or sporadically overloaded, measures must be taken to relieve the load. This often occurs when deadlines are approaching, and, therefore, a task force is formed to relieve the situation. Chapter 8 offers knowledge that can serve as a valuable basis for such task forces. Section 8.4, which rounds off the chapter, can be used as a starting point.

The focus throughout the book is to provide a close proximity to practice. Theory is always illustrated with examples, and there are plenty of concrete tips for design, implementation, debugging, and verification.

The chapters are structured in such a way that they can be read most easily in the given order. Nevertheless, while writing this book, an attempt has been made to give each chapter a certain degree of independence, so that the reader is not lost when it is used to look something up or when reading selectively.

I would be grateful for any suggestions, criticism, and hints regarding mistakes and also welcome contact for direct professional discussion.

With that, I wish you lots of enjoyment and technical insight while reading.

Weilheim, Germany Peter Gliwa
May 2020
peter.gliwa@gliwa.com

Acknowledgments

Time is the theme and focus of this book. To organize it in such a way that no major timing problems arose when writing over 300 pages, and creating over 100 illustrations, was sometimes a challenge.

I could not have undertaken and mastered this without the active support of many dear people.

First of all, there is Nick (Dr. Nicholas Merriam), from whom I learned a lot, such as an understanding of caches, pipelines, and spinlocks. In general, a lot of my knowledge about multi-core and runtime optimization has Nick as its source and this can now be found in the book. Thanks a lot for that, Nick!

I would also like to thank all the interview partners, not only for the time they took for the interviews but also for our collaborative work in standardization committees—they would be much less fun without you.

A big thanks goes to Stuart Cording (www.cordingconsulting.com) who not only brought the English in this book to an acceptable level but also found several flaws in its contents. Many thanks!

I would like to thank Birgit Tamkus, Peter Stief, Christian Herget, Mark Russell, and Christian Wenzel-Benner very much for their reviews and suggestions. Many thanks to you all also for supporting me in many aspects of my daily work during my writing-intensive periods.

I would like to thank my publisher *Springer*—especially, Mr. Ralf Gerstner—for their pleasant, uncomplicated, and constructive cooperation.

I still remain today grateful to Hans Sarnowski of BMW for encouraging me, back in 2002, to found a company that, from the very beginning, specialized in embedded software timing. We have also achieved many successes together on the front line of timing problems—and each and every one of them was a lot of fun.

Finally, I would like to express my greatest thanks to my wife, Priscilla, without whom I could not have written the book. Especially in the spring of 2020—and despite the Corona crisis—she freed up time for me, took care of our four children, and made it possible for me to spend many weekends and holidays in peace and quiet at the office. Many thanks for that!

Contents