Hardware/Firmware Interface Design

Best Practices for Improving Embedded Systems Development

Gary Stringham

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Newnes is an imprint of Elsevier

Newnes

Newnes is an imprint of Elsevier
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA
The Boulevard, Langford Lane, Kidlington, Oxford, OX5 1GB, UK

Notices
Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data
Application Submitted.

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

ISBN: 978-1-85617-605-7

Typeset by diacriTech, Chennai, India

Printed in the United States of America
08 09 10 10 9 8 7 6 5 4 3 2 1

Contents

Preface

You can find books written by hardware engineers teaching hardware engineers how to design hardware. You can find books written by firmware engineers teaching firmware engineers how to write firmware. This book is written by a firmware engineer but is directed primarily to hardware engineers.

Many engineers have experienced problems when trying to get firmware working on hardware. They are designed generally in isolation from each other and then are expected to work when brought together. But problems and defects appear. At times it is unknown where the defect is located—in hardware or firmware, or maybe the documentation.

There is very little written about how to get hardware and firmware to work well together. This book attempts to fill that niche. It addresses the interface between the hardware and firmware domains and provides practices that will reduce the time and effort required to produce an embedded systems product. It covers all aspects of development surrounding the hardware/firmware interface, including the process of development, the high-level design, and the detailed design.

A key feature of this book are the 300+ Best Practices that give detailed instructions for various aspects of the development process and design. These best practices apply perfectly, but only for a given situation. They should be scrutinized for applicability in a given situation. Throughout this book, the emphasis is for engineers to develop their own set of best practices. They may start with these 300, but the set should evolve to be made their own, as this increases the likelihood of success within their organization.

To help engineers understand the 300+ Best Practices, and to help them create their own set, Seven Principles are presented that provide overarching guidelines and direction. These principles, when internalized, will help engineers work in the right direction, even if there is no specific best practice for that situation. Following the Seven Principles and 300+ Best Practices will improve the design teams' ability to produce successful embedded systems products.

Chapter Summaries

The following chapter summaries provide an overview of the book and help the reader to navigate through the book.

1. **Introduction:** This chapter establishes the foundation for the book. It discusses various types of hardware and how they impact the hardware/firmware interface. It defines principles and best practices, the target audience, and the product life cycle. It also presents a case study used throughout the book.
2. **Principles:** This chapter presents the Seven Principles and provides a high-level view and reasoning for the direction of this book. Understanding these principles is key to understanding why the best practices are stated as they are.
3. **Collaboration:** Of key importance to the success of an embedded product is the proper and sufficient collaboration between hardware and firmware engineers. This chapter defines roles and processes in such an effort.
4. **Planning:** Before starting a project, planning must be done to determine and agree what direction should be taken with the new product. This chapter covers several areas that should be visited when planning a new project.
5. **Documentation:** Most engineers assigned to write documentation do not like the task. And most engineers reading documentation get frustrated with incomplete and incorrect documentation. This chapter discusses the types of documentation, when to write them, how to review them, and what types of details to include in them.
6. **Superblock:** This chapter introduces the concept of a block that can do everything within its own domain. It discusses why a superblock is good and how to set it up to be used where needed. But it also discusses the reality of practical limitations and how to handle those.
7. **Design:** Various design aspects are discussed in this chapter, such as events, power-on sequences, communication, and control.
8. **Registers:** Registers are the fundamental interface between hardware and firmware. This chapter discusses them in great detail, including addresses, bit locations, and types of bits.
9. **Interrupts:** Given a lack of consistency among interrupt designs used in the industry, this chapter focuses in great detail how interrupts from hardware into firmware should be managed. This chapter also contains a proposal for an interrupt standard and discusses the proposal in detail.
10. **Aborts, etc.:** Too often very little thought is given to errors and how to recover from them. This chapter discusses design elements necessary to allow firmware to abort hardware operations, recover, and resume processing.

11. **Hooks:** Logic analyzers cannot probe the internals of a chip but knowledge of what is occurring inside is important when trying to get firmware working on hardware. Having firmware-accessible hooks inside the chip allows firmware to retrieve information for engineering analysis. This chapter contains many possible hooks that could be included.
12. **Conclusion:** This chapter wraps up the book. It also contains a couple of cartoon illustrations used to help illustrate the concepts in the book.

Appendices

A. **Best Practices:** This appendix collects all the best practices in the book into one place.

B. **Block Specification:** This appendix is a documentation template as explained and described in Chapter 5, Documentation.

C. **Using This Book in a University:** This appendix provides suggestions on how to use this book to teach hardware and firmware engineering students that have to work together on a project.

D. **Glossary:** Given that this book addresses two different engineering disciplines, hardware and firmware engineering, it covers terms from one domain that might not be understood by the other.

Conventions Used in This Book

The bulk of the text in this book discusses the concept at hand. Interspersed in the text are one or more of these elements: figures, listings, register maps, best practices, and tales from the trenches.

Figures

Figure 0.1 is an example figure.

Firmware Listings

Listing 0.1 shows an example listing of firmware source code written in C.

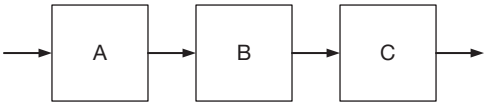

Figure 0.1: Example figure.

Listing 0.1: Example C code listing.

```
/* Read the current list of pending interrupts */
interrupts = *interruptRegister;
```

Hardware Circuits

A few hardware circuits are illustrated in the book. Both a schematic drawing and its equivalent Verilog listing will be given. Figure 0.2 is the schematic and Listing 0.2 is the corresponding Verilog code for an example circuit.

Register Maps

This diagrammatic form is used in discussions about registers, how various bits are mapped into the register, and the mode of operation and reset values of these bits. A detailed explanation of this format is given in Chapter 5, Documentation.

	MSB																										LSB	
	Daily Register – 0x0004																											
Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	B	A
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A This bit means one thing.

B This bit means another.

C And this bit means something else.

Best Practices

The book contains 300+ Best Practices related to the concepts being taught. In addition to presentation throughout the book, these practices are collected in Appendix A, thereby

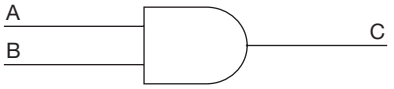

Figure 0.2: Schematic for example circuit.

Listing 0.2: Verilog code for example circuit.

```
// A simple AND gate.
assign c = a & b;
```

providing a concise checklist that can be used during chip design projects. They are also provided in a spreadsheet available online at the publisher's website, elsevierdirect.com/companions, and at the author's website, garystringham.com/hwfwbook.

Each best practice has an ID number, X.Y.Z, which is used in the body of the book, in Appendix A, and in the spreadsheet.

> ☑ **Best Practice**
>
> 1.1.1 Best Practices of Hardware/Firmware Interface Design.

Like the book, the Excel spreadsheet database is copyrighted material. Purchasers of this book are entitled (and encouraged) to start with the database and modify it to suit the needs of their design team, but some restrictions apply. See Appendix A for more details on the database and its copyright permissions.

> ☑ **Best Practice**
>
> 1.1.2 Copyright © 2009, Gary Stringham & Associates, LLC. All rights reserved. Do not distribute beyond your team.

Tales from the Trenches

Scattered throughout this book are real-life stories that help illustrate the impact of the topic at hand. These are stories from real engineers (mostly me) in the trenches, working away designing and solving problems. The following is an example tale (not a real one).

> 🕮 **Tales from the Trenches**
>
> I remember hearing a story from a friend of a friend, who heard that an engineer had said that he heard a manager tell her subordinate that–according to the rumor she had heard–it was already broken to start with.

Companion Website

This book has a companion website at elsevierdirect.com/companions/9781856176057, where you will find links to the spreadsheet database for the 300+ Best Practices, the document template discussed in Chapter 5, Documentation, and other related content. Please

visit the author's website at garystringham.com/hwfwbook for the same tools, plus additional links to his work in this area and details of how to contact him directly.

How to Contact Me

If you have any questions about the content of this book or about your hardware/firmware interface design, feel free to contact me at gary@garystringham.com.

Acknowledgments

I would like to thank Jack Meador and Mike Merrell, the two unlucky hardware engineers who had to put up with my constant questions, issues, and requests as we worked through the project that was the catalyst for this book. They provided valuable insight and help from within their hardware domain. They, along with other hardware and firmware engineers within the organization and from other companies, provided much of the input used in many of the best practices and tales from the trenches in this book.

I would also like to thank my immediate managers at the time, Warren Johnson and Tracy Sauerwein, and managers above them, Sandy Lieske and Von Hansen. The book is finally published—your unwarranted support, while tracing my progress from sandy to smooth, was not in vain.

Me badly english writin' was greatly improved through the patient tutelage of my technical writing coach, Joel Saks. He has a gift with words that is way beyond my abilities. In addition, he was a valuable resource for critical analysis of my concepts, pushing me to clearly articulate and justify what seemed obvious to me.

I would also like to thank John Blyler, Clive "Max" Maxfield, Jack Meador, Mike Merrell, Joel Saks, and three others (who wish to remain anonymous) for reviewing all or parts of my book. Your comments provided valuable input and suggestions, making the book better than otherwise. Thanks to Mike Merrell for his help with the Verilog code and to Kevin Falk for drawing the car illustrations. And thanks to the many others who have given me suggestions and enthusiastic encouragement during the 5 years it took me to complete this project.

Most of all, I want to thank my wife and children  for their patience and long suffering as I spent evenings and weekends working on this book instead of making repairs on the house, driving the children to their activities, and vacationing with the family.