

Numbers and Computers

Ronald T. Kneusel

Numbers and Computers

Second Edition



Springer

Ronald T. Kneusel
Broomfield, Colorado, USA

ISBN 978-3-319-50507-7 ISBN 978-3-319-50508-4 (eBook)
DOI 10.1007/978-3-319-50508-4

Library of Congress Control Number: 2016960676

© Springer International Publishing AG 2015, 2017

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

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

*To my parents, Janet and Tom, for fostering
my love of science.*

Preface to the Second Edition

This book is about numbers and how they are represented and manipulated by computers. In this second edition, we expand on the coverage of the first edition by adding three new chapters and new material to several sections.

Broomfield, CO, USA
November 2016

Ronald T. Kneusel

Preface

This is a book about numbers and how those numbers are represented in and operated on by computers.

Of course, numbers are fundamental to how computers operate because, in the end, everything a computer works with is a number. It is crucial that people who develop a code understand this area because the numerical operations allowed by computers and the limitations of those operations, especially in the area of floating point math, affect virtually everything people try to do with computers. This book aims to help by exploring, in sufficient, but not overwhelming, detail, just what it is that computers do with numbers.

Who Should Read This Book

This book is for anyone who develops software including software engineers, scientists, computer science students, engineering students, and anyone who programs for fun.

If you are a software engineer, you should already be familiar with many of the topics in this book, especially if you have been in the field for any length of time. Still, I urge you to press on, for perhaps you will find a gem or two which are new to you.

If you are old enough, you will remember the days of Fortran and mainframes. If so, like the software engineers above, you are probably also familiar with the basics of how computers represent and operate on numbers, but, also like the software engineers above, you will likely find a gem or two of your own. Scientists in particular should be aware of the limitations and pitfalls associated with using floating point numbers since few things in science are restricted to integers.

Students need this book because it is essential to know what the computer is doing under the hood. After all, if you are going to make a career of computers, why would you not want to know how the machine works?

How to Use This Book

This book consists of two main parts. The first deals with standard representations of integers and floating point numbers, while the second details several other number representations which are nice to know about and handy from time to time. Either part is a good place to start, though it is probably best if the parts themselves are read from start to end. Later, after the book has been read, you can use it as a reference.

There are exercises at the end of each chapter. Most of these are of the straightforward pencil and paper kind, just to test your understanding, while others are small programming projects meant to increase your familiarity with the material. Exercises that are (subjectively) more difficult will be marked with either one or two stars (* or **) to indicate the level of difficulty.

Example code is in C and/or Python version 2.7 though earlier 2.x versions should work just as well. Intimate knowledge of these programming languages is not necessary in order to understand the concepts being discussed. If something is not easy to see in the code, it will be described in the text. Why C? Because C is a low-level language, close to the numbers we will be working with, and because C is the grandfather of most common programming languages in current use including Python. In general, code will be offset from text and in a monospace font. For readers not familiar with C and/or Python, there are a plethora of tutorials on the web and reference books by the bookcase. Two examples, geared toward people less familiar with programming, are *Beginning C* by Ivor Horton and *Python Programming Fundamentals* by Kent Lee. Both of these texts are available from Springer in print or e-book format.

At the end of each chapter are references for material presented in the chapter. Much can be learned by looking at these references. Almost by instinct we tend to ignore sections like this as we are now programmed to ignore advertisements on web pages. In this former case, resist temptation; in the latter case, keep calm and carry on.

Acknowledgments

This book was not written in a vacuum. Here, I want to acknowledge those who helped make it a reality. First the reviewers, who gave of their time and talent to give me extremely valuable comments and friendly criticism: Robert Kneusel, M.S.; Jim Pendleton; Ed Scott, Ph.D.; and Michael Galloy, Ph.D. Gentlemen, thank you. Second, thank you to Springer, especially my editor, Courtney Clark, for moving ahead with this book. Lastly, and most importantly, thank you to my wife, Maria, and our children: David, Peter, Paul, Monica, Joseph, and Francis. Without your patience and encouragement, none of this would have been written.

Broomfield, CO, USA
December 2016

Ronald T. Kneusel
AM+DG

Contents

Part I Standard Representations

1	Number Systems	3
1.1	Representing Numbers	3
1.2	The Big Three (and One Old Guy)	8
1.3	Converting Between Number Bases	10
1.4	Chapter Summary	16
	Exercises	16
	References	17
2	Integers	19
2.1	Bits, Nibbles, Bytes, and Words	19
2.2	Unsigned Integers	21
2.2.1	Representation	21
2.2.2	Storage in Memory: Endianness	22
2.3	Operations on Unsigned Integers	25
2.3.1	Bitwise Logical Operations	25
2.3.2	Testing, Setting, Clearing, and Toggling Bits	30
2.3.3	Shifts and Rotates	33
2.3.4	Comparisons	37
2.3.5	Arithmetic	41
2.3.6	Square Roots	52
2.4	What About Negative Integers?	54
2.4.1	Sign-Magnitude	54
2.4.2	One's Complement	55
2.4.3	Two's Complement	55
2.5	Operations on Signed Integers	56
2.5.1	Comparison	56
2.5.2	Arithmetic	58
2.6	Binary-Coded Decimal	67
2.6.1	Introduction	67
2.6.2	Arithmetic with BCD	69

2.6.3	Conversion Routines	70
2.6.4	Other BCD Encodings	73
2.7	Chapter Summary	76
	Exercises	77
	References	79
3	Floating Point	81
3.1	Floating-Point Numbers	81
3.2	An Exceedingly Brief History of Floating-Point Numbers	84
3.3	Comparing Floating-Point Representations	85
3.4	IEEE 754 Floating-Point Representations	89
3.5	Rounding Floating-Point Numbers (IEEE 754)	97
3.6	Comparing Floating-Point Numbers (IEEE 754)	100
3.7	Basic Arithmetic (IEEE 754)	102
3.8	Handling Exceptions (IEEE 754)	105
3.9	Floating-Point Hardware (IEEE 754)	108
3.10	Binary Coded Decimal Floating-Point Numbers	110
3.11	Chapter Summary	113
	Exercises	114
	References	115
4	Pitfalls of Floating-Point Numbers (and How to Avoid Them)	117
4.1	What Pitfalls?	117
4.2	Some Experiments	119
4.3	Avoiding the Pitfalls	130
4.4	Chapter Summary	134
	Exercises	135
	References	135
 Part II Other Representations		
5	Big Integers and Rational Arithmetic	139
5.1	What is a Big Integer?	139
5.2	Representing Big Integers	140
5.3	Arithmetic with Big Integers	146
5.4	Alternative Multiplication and Division Routines	158
5.5	Implementations	167
5.6	Rational Arithmetic with Big Integers	171
5.7	When to Use Big Integers and Rational Arithmetic	177
5.8	Chapter Summary	180
	Exercises	180
	References	181
6	Fixed-Point Numbers	183
6.1	Representation (Q Notation)	183
6.2	Arithmetic with Fixed-Point Numbers	188
6.3	Trigonometric and Other Functions	194

6.4	An Emerging Use Case	204
6.5	When to Use Fixed-Point Numbers	211
6.6	Chapter Summary	212
	Exercises	212
	References	213
7	Decimal Floating Point	215
7.1	What is Decimal Floating-Point?	215
7.2	The IEEE 754-2008 Decimal Floating-Point Format	216
7.3	Decimal Floating-Point in Software	225
7.4	Thoughts on Decimal Floating-Point	232
7.5	Chapter Summary	233
	Exercises	234
	References	234
8	Interval Arithmetic	235
8.1	Defining Intervals	235
8.2	Basic Operations	237
8.3	Functions and Intervals	253
8.4	Implementations	258
8.5	Thoughts on Interval Arithmetic	262
8.6	Chapter Summary	263
	Exercises	263
	References	263
9	Arbitrary Precision Floating-Point	265
9.1	What is Arbitrary Precision Floating-Point?	265
9.2	Representing Arbitrary Precision Floating-Point Numbers	265
9.3	Basic Arithmetic with Arbitrary Precision Floating-Point Numbers	270
9.4	Comparison and Other Methods	273
9.5	Trigonometric and Transcendental Functions	274
9.6	Arbitrary Precision Floating-Point Libraries	278
9.7	Thoughts on Arbitrary Precision Floating-Point	290
9.8	Chapter Summary	291
	Exercises	291
	References	292
10	Other Number Systems	293
10.1	Introduction	293
10.2	Logarithmic Number System	293
10.3	Double-Base Number System	307
10.4	Residue Number System	324
10.5	Redundant Signed-Digit Number System	332
10.6	Chapter Summary	339
	Exercises	340
	References	341
	Index	343