

# ARM Assembly Language with Hardware Experiments

Ata Elahi • Trevor Arjeski

# ARM Assembly Language with Hardware Experiments

 Springer

Ata Elahi  
Southern Connecticut State University  
New Haven  
Connecticut  
USA

Trevor Arjeski  
Southern Connecticut State University  
New Haven  
Connecticut  
USA

ISBN 978-3-319-11703-4

ISBN 978-3-319-11704-1 (eBook)

DOI 10.1007/978-3-319-11704-1

Library of Congress Control Number: 2014955658

Springer Cham Heidelberg New York Dordrecht London

© Springer International Publishing Switzerland 2015

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

Springer is part of Springer Science+Business Media ([www.springer.com](http://www.springer.com))

# Preface

ARM is one of the leading suppliers of microprocessors for the entire world. ARM has designed and developed a CPU that partner companies can manufacture and add more peripherals to the processor. An ARM processor has a wide range of application in today's technology, such as mobile phones, tablets, televisions, and automobiles. Learning the ARM instruction set and ARM assembly programming is an essential tool in the development of low-level applications for the ARM processor. Engineers will benefit significantly from the understanding of computer architecture and assembly language, especially if they are working in an industry where performance is crucial or hardware is being developed.

**Organization** This book contains seven chapters. The reader does not require any background in ARM assembly language to understand material of this book.

Chapters one and two of this book form a foundation for the rest of the chapters.

Chapter 1 covers some necessary knowledge of digital signals, analog signals, number systems and transmission methods.

Chapter 2 covers logic gates, registers and an introduction to computer architecture.

Chapters 3 and 4 cover the ARM processor architecture with its instructions.

Chapter 5 covers ARM assembly language programming using Keil development tools.

Chapter 6 covers ARM Cortex-M3 processor architecture, the MBED NXP LPC1768 and basic GPIO Programming.

Chapter 7 covers lab experiments that include:

- Creating a binary counter using onboard LEDs
- Configuring an Analog-To-Digital Converter (ADC)
- Creating a voltmeter with an ADC
- Configuring Digital to Analog Converter (DAC)
- Converting binary to output for a hexadecimal display
- Configuring a Real-Time Clock (RTC)

**Intended Audience** This book is written primarily as an introduction to assembly language for students who are studying computer science, computer engineering,

or hobbyists who are simply interested in learning ARM assembly programming with hands-on experiments. This book can be used as a first course in computer system which covers numbers systems, Digital Logics, Introduction to Computer Architecture and Assembly language for computer science and computer technology students.

# Contents

<b>1</b>	<b>Number Systems and Data Communication</b> .....	1
1.1	Introduction .....	1
1.2	Analog Signals .....	1
1.3	Digital Signals.....	4
1.4	Number System.....	4
1.5	Coding Schemes.....	10
1.6	Clock .....	12
1.7	Transmission Modes.....	13
1.8	Transmission Methods .....	14
<b>2</b>	<b>Logic Gates and Introduction to Computer Architecture</b> .....	17
2.1	Introduction.....	17
2.2	Logic Gates .....	17
2.3	Integrated Circuit (IC) Classification.....	21
2.4	Registers.....	22
2.5	Introduction to Computer Architecture .....	22
2.6	Memory .....	27
2.7	Multiplexer and Decoder.....	30
<b>3</b>	<b>ARM Instructions Part I</b> .....	35
3.1	Introduction .....	35
3.2	Instruction Set Architecture (ISA) .....	38
3.3	ARM Instructions .....	39
3.4	Register Swap Instructions (MOV and MVN).....	42
3.5	Shift and Rotate Instructions.....	43
3.6	ARM Unconditional Instructions and Conditional Instructions.....	46
3.7	ARM Data Processing Instruction Format .....	47
3.8	Stack Operation and Instructions .....	49
3.9	Branch (B) and Branch with Link Instruction (BL).....	51
3.10	Multiply (MUL) and Multiply-Accumulate (MLA) Instructions .....	53

- 4 ARM Instructions and Part II** ..... 57
  - 4.1 ARM Data Transfer Instructions ..... 57
  - 4.2 ARM Addressing Mode..... 59
  - 4.3 Data Transfer Instruction Format ..... 61
  - 4.4 Block Transfer Instruction and Instruction Format ..... 62
  - 4.5 Swap Memory and Register (SWAP)..... 62
  - 4.6 Bits Field Instructions ..... 63
  - 4.7 Data Representation and Memory..... 65
  
- 5 ARM Assembly Language Programming Using Keil**
  - Development Tools Introduction**..... 69
    - 5.1 Introduction ..... 69
    - 5.2 Keil Development Tools for ARM Assembly ..... 69
    - 5.3 Program Template ..... 76
    - 5.4 Programming Rules..... 76
    - 5.5 Directives ..... 77
  
- 6 ARM Cortex-M3 Processor and MBED NXP LPC1768**..... 83
  - 6.1 Introduction ..... 83
  - 6.2 MBED NXP LPC1768 ..... 86
  - 6.3 Basic GPIO Programming..... 88
  - 6.4 Flashing the NXP LPC1768 ..... 95
  
- 7 Lab Experiments**..... 97
  - 7.1 Introduction ..... 97
  - 7.2 Lab#1 Binary Counter Using Onboard LEDs..... 97
  - 7.3 Lab2: Configuring the Real-Time Clock (RTC) ..... 100
  - 7.4 Lab#3 Configuring Analog-To-Digital Converter (ADC) ..... 104
  - 7.5 Lab #4: Digital to Analog Converter (DAC)..... 113
  - 7.6 Experiment #5: Binary to Hexadecimal Display ..... 116
  - 7.7 Universal Asynchronous Receiver/Transmitter (UART) ..... 118
  
- Solution to the Problems and Questions** ..... 123
  
- References**..... 139