

1. Record Nr.	UNINA9911019832003321
Autore	Vasavada Darshak S
Titolo	The Microprocessor : A Practical Introduction Using the Arm Cortex-M0 Processor
Pubbl/distr/stampa	Newark : , : John Wiley & Sons, Incorporated, , 2025 ©2025
ISBN	1-394-24531-9 1-394-24532-7
Edizione	[1st ed.]
Descrizione fisica	1 online resource (531 pages)
Collana	Wiley-Arm Professional Computing and Electrical Engineering Series
Altri autori (Persone)	SinhaS. K
Disciplina	004.16
Soggetti	Microprocessors
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	First place of publication from the publisher's website.
Nota di contenuto	Cover -- Title Page -- Copyright -- Contents -- List of Exercises -- Preface -- About This Book -- How to Read This Book -- Acknowledgements -- About the Companion Website -- Introduction -- CHAPTER 1 Microprocessor System -- 1.1 Introduction -- 1.2 Processor -- 1.2.1 Word size -- 1.2.2 Clock speed -- 1.2.3 Instruction set -- 1.2.4 Number of cores -- 1.3 Memory -- 1.3.1 Content of memory -- 1.3.2 Address -- 1.3.3 Memory mapping -- 1.4 Memory Devices -- 1.4.1 ROM devices -- 1.4.2 RAM devices -- 1.4.3 Onchip and offchip memory -- 1.4.4 Primary and secondary memory -- 1.5 Bus -- 1.5.1 Bus operation -- 1.5.2 Bus fault -- 1.6 IO Ports -- 1.7 Reset, Clock and Power Management -- 1.7.1 Reset -- 1.7.2 Clock -- 1.7.3 Power management -- 1.8 Overview of Arm CortexM0 Processor -- 1.9 Summary -- CHAPTER 2 Instruction Set Architecture -- 2.1 Introduction -- 2.2 Registers -- 2.3 Instruction Set -- 2.4 Structure of an Instruction -- 2.4.1 Instruction in assembly language -- 2.4.2 Instruction code -- 2.5 DataProcessing Instructions -- 2.5.1 Structure of a dataprocessing instruction -- 2.5.2 Arithmetic instructions -- 2.5.3 Logic instructions -- 2.6 MemoryAccess Instructions -- 2.6.1 Simple load and store -- 2.6.2 Loadstore with base and offset -- 2.6.3 Word length -- 2.6.4 Multiple load and store -- 2.6.5 Stack instructions -- 2.7 ProgramControl Instructions -- 2.7.1 Branch -- 2.7.2 Indirect branch -- 2.7.3 Conditional branch -- 2.7.4 Function call

and return -- 2.8 Summary -- CHAPTER 3 Program Execution -- 3.1 Introduction -- 3.2 Program -- 3.3 Inside the CPU -- 3.4 Fetch Unit -- 3.5 Decode Unit -- 3.5.1 Dataprocessing instructions -- 3.5.2 Load store instructions -- 3.6 Execution Unit -- 3.6.1 Register bank -- 3.6.2 Arithmetic logic unit -- 3.6.3 Loadstore unit -- 3.7 Instruction Execution -- 3.8 Processor Pipeline -- 3.9 Summary.

CHAPTER 4 Assembly Programming -- 4.1 Statements -- 4.1.1 Comments -- 4.1.2 Spaces and casesensitivity -- 4.2 Labels -- 4.3 Sections -- 4.4 Text Section -- 4.5 Data Sections -- 4.5.1 Readonly data -- 4.5.2 Initialized data -- 4.5.3 Uninitialized data -- 4.6 Summary -- CHAPTER 5 Arithmetic Operations -- 5.1 Arithmetic Instructions -- 5.2 Unsigned Integers -- 5.3 Signed Integers -- 5.4 APSR Flags -- 5.4.1 Zero (Z) flag -- 5.4.2 Negative (N) flag -- 5.4.3 Overflow (V) flag -- 5.4.4 Carry (C) flag -- 5.5 Condition Codes -- 5.5.1 Unsigned comparison -- 5.5.2 Signed comparison -- 5.6 Summary -- CHAPTER 6 BitLevel Operations -- 6.1 Boolean Instructions -- 6.1.1 AND operation -- 6.1.2 OR operation -- 6.1.3 ExclusiveOR operation -- 6.1.4 NOT operation -- 6.2 Bit Manipulation -- 6.2.1 Setting bits -- 6.2.2 Clearing bits -- 6.2.3 Testing bit status -- 6.3 Shift Operations -- 6.3.1 Logical shift -- 6.3.2 Arithmetic shift -- 6.3.3 Rotate -- 6.4 WordLength Extension -- 6.4.1 Zeroextension -- 6.4.2 Signextension -- 6.4.3 Instructions for wordlength extension -- 6.5 Byte Ordering Instructions -- 6.6 Summary -- CHAPTER 7 Load and Store Operations -- 7.1 Introduction -- 7.2 Alignment -- 7.2.1 Alignment in code access -- 7.2.2 Unaligned accesses -- 7.3 Endianness -- 7.3.1 Endianness in instruction codes -- 7.4 Basic Load and Store Operations -- 7.4.1 Bus fault -- 7.5 Data Types -- 7.5.1 Storing halfwords and bytes -- 7.5.2 Loading unsigned halfwords and bytes -- 7.5.3 Loading signed numbers -- 7.6 Offset Addressing -- 7.6.1 Immediate offset -- 7.6.2 Register offset -- 7.6.3 PCrelative addressing -- 7.6.4 Multiple loadstore operations -- 7.7 Summary -- CHAPTER 8 Branch and Subroutine -- 8.1 ProgramControl Instructions -- 8.2 Branch -- 8.3 Conditional Branch -- 8.4 Indirect Branch -- 8.5 Subroutines -- 8.6 Nested Subroutines -- 8.7 Summary.

CHAPTER 9 Stack Operations -- 9.1 Introduction -- 9.2 What Is a Stack? -- 9.3 Stack in CortexM0 -- 9.4 Stack Operations -- 9.5 Creating a Stack -- 9.6 Using the Stack -- 9.7 Local Variables -- 9.8 Summary -- CHAPTER 10 Exceptions -- 10.1 Introduction -- 10.1.1 Reset -- 10.1.2 Faults -- 10.1.3 Supervisor call -- 10.1.4 Interrupts -- 10.2 Exception Model -- 10.3 Vector Table -- 10.4 Reset -- 10.5 Faults -- 10.6 Exception Stack Frame -- 10.7 Summary -- CHAPTER 11 SysTick and Core Interrupts -- 11.1 Introduction -- 11.2 SysTick -- 11.3 SysTick Programming Model -- 11.3.1 SysTick Current Value Register -- 11.3.2 SysTick Reload Value Register -- 11.3.3 Control and Status Register -- 11.4 Programming SysTick -- 11.5 Using SysTick -- 11.6 Polling with SysTick -- 11.7 SysTick Interrupts -- 11.8 Interrupt Masking -- 11.9 Nonmaskable Interrupt (NMI) -- 11.10 Summary -- CHAPTER 12 Introduction to C Programming -- 12.1 Programming Languages -- 12.2 Structure of a C Program -- 12.2.1 C source files -- 12.2.2 Object files -- 12.2.3 Library files -- 12.2.4 Header files -- 12.2.5 Compilation process -- 12.3 Functions -- 12.4 Data -- 12.5 Header Files -- 12.6 Overview of C Language -- 12.6.1 Datarelated keywords (20) -- 12.6.2 Functionrelated keywords (12) -- 12.6.3 Study map -- 12.7 Summary -- CHAPTER 13 Basic Data Types -- 13.1 Characters and Strings -- 13.2 Integers -- 13.2.1 Arithmetic operations -- 13.2.2 Bit level operations -- 13.2.3 Sign and wordlength -- 13.3 FloatingPoint Numbers -- 13.4 FloatingPoint Operations -- 13.5 TypeCasting -- 13.6 Summary -- CHAPTER 14 Functions -- 14.1 Function Calls --

14.2 Arguments -- 14.3 Local Variables -- 14.4 Conditional Execution
-- 14.5 Selection -- 14.6 Loops -- 14.6.1 for loop -- 14.6.2 while loop
-- 14.6.3 dowhile loop -- 14.6.4 break and continue -- 14.6.5 goto
statement -- 14.7 Summary.
CHAPTER 15 Extended Data Types -- 15.1 Arrays -- 15.2 Structures --
15.3 Pointers -- 15.4 Arrays and Pointers -- 15.5 Pointer to a Structure
-- 15.6 Pointer Arithmetic -- 15.7 Uninitialized Pointers -- 15.8
Summary -- CHAPTER 16 Compilation Process -- 16.1 Overview of the
Compilation Process -- 16.1.1 Preprocessing -- 16.1.2 Compilation --
16.1.3 Assembly -- 16.1.4 Linking -- 16.2 Preprocessing -- 16.3
Compilation -- 16.3.1 C statements -- 16.3.2 Translation process --
16.4 Assembler -- 16.4.1 Translation -- 16.4.2 Relative address
assignment -- 16.5 Linker -- 16.5.1 Section placement -- 16.5.2
Absolute address assignment -- 16.5.3 Symbol resolution -- 16.6
Executable Program -- 16.7 Summary -- CHAPTER 17 Microcontroller
-- 17.1 Introduction -- 17.2 Block Diagram -- 17.2.1 CortexM0 --
17.2.2 Bus -- 17.2.3 Memory -- 17.2.4 Reset, clock and power
management -- 17.2.5 Peripheral (IO) ports -- 17.3 Pin Diagram --
17.4 Reset, Clock and Power Management -- 17.4.1 Reset control unit
-- 17.4.2 Clock control unit -- 17.4.3 Power management -- 17.4.4
API functions -- 17.5 Peripheral Interrupts -- 17.5.1 Interrupts in AM0
-- 17.5.2 Nested Vectored Interrupt Controller (NVIC) -- 17.5.3
Interrupt handling -- 17.6 Peripheral Registers -- 17.6.1 Register
definition -- 17.6.2 Bitlevel operations -- 17.7 Peripheral API --
17.7.1 System API -- 17.7.2 Digital IO -- 17.7.3 Timer -- 17.7.4
Analog -- 17.7.5 UART -- 17.7.6 Interrupts -- 17.8 Summary --
CHAPTER 18 IO Programming -- 18.1 IO Devices -- 18.2 GPIO --
18.2.1 Registers -- 18.2.2 Peripheral API -- 18.2.3 Implementation --
18.3 Timer -- 18.3.1 Registers -- 18.3.2 Peripheral API -- 18.3.3
Implementation -- 18.4 ADC -- 18.4.1 Registers -- 18.4.2 Peripheral
API -- 18.4.3 Implementation -- 18.5 DAC -- 18.5.1 Registers --
18.5.2 Peripheral API -- 18.5.3 Implementation -- 18.6 UART --
18.6.1 Registers.
18.6.2 Peripheral API -- 18.6.3 Implementation -- 18.7 Interrupts --
18.7.1 Registers -- 18.7.2 Peripheral API -- 18.7.3 Implementation --
18.8 Summary -- CHAPTER 19 Microprocessor Applications -- 19.1
LED Brightness Controller -- 19.1.1 Hardware configuration -- 19.1.2
Code -- 19.2 Ambient Light Sensor -- 19.2.1 Hardware configuration
-- 19.2.2 Code -- 19.3 EnergyEfficient Coding -- 19.4 Temperature
Monitoring -- 19.4.1 Hardware configuration -- 19.4.2 Code -- 19.5
Motor Speed Control -- 19.5.1 Rotary encoder -- 19.5.2 Speed control
algorithm -- 19.5.3 Code -- 19.5.4 Practical considerations -- 19.6
Summary -- APPENDIX A Programming Environment -- A.1
Introduction -- A.2 Keil MDK -- A.2.1 ToolChain -- A.2.2 Target
Processor -- A.2.3 Workflow for the Exercises -- A.3 Assembly
Programming Setup -- A.4 Writing and Building Assembly Programs --
A.5 Debugging an Assembly Program -- A.5.1 Register Window -- A.
5.2 Source Window -- A.5.3 Disassembly Window -- A.5.4 Memory
Window -- A.6 C Programming Setup -- A.7 Writing and Building C
Program -- A.8 Debugging A C Program -- A.8.1 Source Window -- A.
8.2 Disassembly Window -- A.8.3 Watch Window -- A.9 Debugger --
A.9.1 Loading the Program -- A.9.2 Working with the Data -- A.9.3
Execution Control -- A.9.4 Breakpoints and Watchpoints -- APPENDIX B
Advanced Topics -- B.1 SystemControl Instructions -- B.1.1 Processor
Control Instructions -- B.1.2 ProcessorSynchronization Instructions --
B.2 OS Support -- B.2.1 Processor Modes -- B.2.2 Privileges -- B.2.3
Supervisor Call (SVCall) -- B.2.4 PendSV -- B.2.5 Stacks -- APPENDIX C
Startup Code -- C.1 OSLess System -- C.1.1 Vector Table -- C.1.2

Exception Handlers -- C.1.3 Startup Code -- C.1.4 Main Code -- C.1.5
ReadOnly Data -- C.1.6 ReadWrite Data -- C.1.7 Heap -- C.1.8 Stack
-- C.2 Startup Code -- C.3 Linker Script File -- APPENDIX D AM0
Header Files.
D.1 Register Header File.

Sommario/riassunto

"The Microprocessor offers a supremely accessible and user-friendly introduction to microprocessor basics: instruction set, the exception model, system architecture and microcontroller programming. Explaining the working principles with simplified models, this first-level book builds the base for all onward courses at intermediate and advanced levels. Filled with exercises that can be executed on the free version of Keil Vision MDK without any hardware, the book explains the essential aspects of microprocessor architecture with simple programming examples in assembly and C. By blending conceptual knowledge with practical exercises, the book offers valuable insights that equip readers to engage with real-world applications in the fields of microprocessor architecture and embedded systems."--
