

Practical

Embedded Controllers

Design and Troubleshooting with the Motorola 68HC11

John Park



Newnes
An imprint of Elsevier
Linacre House, Jordan Hill, Oxford OX2 8DP
200 Wheeler Road, Burlington, MA 01803

First published 2003

Copyright © 2003, IDC Technologies. All rights reserved

No part of this publication may be reproduced in any material form (including photocopying or storing in any medium by electronic means and whether or not transiently or incidentally to some other use of this publication) without the written permission of the copyright holder except in accordance with the provisions of the Copyright, Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London, England W1T 4LP. Applications for the copyright holder's written permission to reproduce any part of this publication should be addressed to the publisher

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 07506 58029

For information on all Newnes publications, visit our website at www.newnespress.com
--

Typeset and Edited by Vivek Mehra, Mumbai, India
(vivekmehra@tatanova.com)

Printed and bound in Great Britain

Special thanks to

Industrial Automation
www.cs.jcu.edu.au/~gregory/hc11/

All photos in this book courtesy of

Cursor Magic
www.cursormagic.com
photos@cursormagic.com

Practical Embedded Controllers: Design and Troubleshooting with the Motorola 68HC11

Titles in the series

Practical Cleanrooms: Technologies and Facilities (David Conway)

Practical Data Acquisition for Instrumentation and Control Systems (John Park, Steve Mackay)

Practical Data Communications for Instrumentation and Control (Steve Mackay, Edwin Wright, John Park)

Practical Digital Signal Processing for Engineers and Technicians (Edmund Lai)

Practical Electrical Network Automation and Communication Systems (Cobus Strauss)

Practical Embedded Controllers (John Park)

Practical Fiber Optics (David Bailey, Edwin Wright)

Practical Industrial Data Networks: Design, Installation and Troubleshooting (Steve Mackay, Edwin Wright, John Park, Deon Reynders)

Practical Industrial Safety, Risk Assessment and Shutdown Systems for Instrumentation and Control (Dave Macdonald)

Practical Modern SCADA Protocols, DNP3, 60870.5 and Related Systems (Gordon Clarke, Deon Reynders)

Practical Radio Engineering and Telemetry for Industry (David Bailey)

Practical SCADA for Industry (David Bailey, Edwin Wright)

Practical TCP/IP and Ethernet Networking (Deon Reynders, Edwin Wright)

Practical Variable Speed Drives and Power Electronics (Malcolm Barnes)

Preface

From microwave ovens to alarm systems to industrial programmable logic controllers (PLCs) and distributed control systems (DCSs), embedded controllers are running our world.

Embedded controllers are used in most items of electronic equipment today. They can be thought of as intelligent electronic devices used to control and monitor devices connected to the real world. This can be a PLC, DCS or a smart sensor. These devices are used in almost every walk of life today. Most automobiles, factories and even kitchen appliances have embedded controllers in them.

The microcontrollers that are at the heart of these and many more devices are becoming easier and simpler to use. But when these devices fail, the solution to the problem needs to be found and repairs done quickly.

This book will help technicians, engineers and even the casual user understand the workings of microcontrollers, along with the most common problems and their solutions.

This book covers all aspects of embedded controllers but is biased towards troubleshooting and design. The book also covers design, specification, programming, installation, configuration and troubleshooting.

After reading this book we hope you will have learnt how to:

- Design, set up and program a complete embedded controller development system
- Apply the latest techniques in programming these versatile devices
- Apply troubleshooting tips and tricks for microcontrollers
- Apply the best techniques for installation of microcontrollers
- Fix problems due to electrical noise and interference
- Design correctly the first time to avoid grounding and EMC problems
- Choose and configure the correct software

Typical people who will find this book useful include:

- Electronic technicians and engineers
- Instrumentation and control engineers and technicians
- Process control engineers and technicians
- Electrical engineers
- Consulting engineers
- Process development engineers
- Design engineers
- Control systems sales engineers

A basic knowledge of electrical principles is useful in understanding the concepts outlined in the book, but the contents are of a fundamental nature and are easy to comprehend.

The structure of the book is as follows.

Chapter 1: Introduction. This chapter gives a brief overview of the main components of a microcontroller.

Chapter 2: Microcontroller basics. A review of the basics of this device with a discussion on number systems, Boolean logic, accumulators, registers, data communications, power systems, crystals and oscillators, is done in this chapter.

Chapter 3: Microcontroller programming. A review of the simple techniques involved in programming a microcontroller with a discussion on the various programming issues such as programming structures, addressing modes, operations and finally a short comparison of C++ and BASIC, is done in this chapter.

Chapter 4: Microcontroller memory. The main types and techniques in the effective use of memory such as user RAM, BUFFALO routines, interrupts, control registers, and EEPROM are assessed here.

Chapter 5: Microcontroller inputs and outputs. Analog and digital inputs, keypad and LCD interfacing are described here.

Chapter 6: Data communications. This important topic is broken down into a discussion on the fundamentals, the OSI model, modes of communication and RS-232 and RS-485.

Chapter 7: Noise reduction. This chapter gives an overview of noise reduction and a discussion on conductive, capacitive, and magnetically coupled noise.

Chapter 8: EMC grounding solutions. The most important features of grounding (and protection from lightning) to protect the microcontroller from the effects of EMC are discussed here.

Chapter 9: Installation and troubleshooting. This chapter is a short discussion on connections, cable runs and trays, wire management and troubleshooting techniques.

Chapter 10: End notes. A wrap discussion on the issues discussed in the earlier chapters with a few words on assembly language programming, memory, inputs and outputs, data communication, noise reduction and grounding solutions and finally installation techniques.

Contents

Preface

xiii

1	Introduction	1
1.1	Microcontroller introduction	1
1.2	Microcontroller design and functions	3
1.3	Assembly language programming	5
1.4	Inputs and outputs	7
1.5	Data communication	8
1.6	Noise reduction	9
1.7	Grounding solutions	10
1.8	Installation techniques	11
1.9	Conclusion	12
2	Microcontroller basics	13
2.1	Introduction	13
2.2	Number systems – binary, hex, and decimal	17
2.2.1	Bits, dibits, nibbles, bytes, words and long words	17
2.2.2	Conversion from binary to hexadecimal	18
2.2.3	ASCII, hexadecimal and BCD	18
2.2.4	Converting from ASCII to BCD and HEX	19
2.3	Gates – AND, OR, XOR and NOT gates	20
2.3.1	AND gates – physical and virtual	20
2.3.2	OR gates – physical and virtual	21
2.3.3	XOR gates – physical and virtual	21
2.3.4	NOT gates – physical and virtual	22
2.4	Accumulators, A, B and D	22
2.4.1	Addressing modes	22
2.4.2	Loading, storing and transferring accumulators	23
2.4.3	Add, subtract, compare, increment and decrement A and B	24
2.4.4	AND, OR and test bits	24
2.4.5	Arithmetic and logical shifting and rotating	24
2.4.6	Data and bit testing	24
2.5	Registers – X, Y, the stack and ports	25
2.5.1	The X and Y registers	25
2.5.2	The stack	28
2.5.3	Ports A, B, C, D and E	27
2.6	Communications synchronous and asynchronous	28

2.6.1	Asynchronous character communications	28
2.6.2	Synchronous packet communications	29
2.6.3	Specifying a system – asynchronous vs synchronous	30
2.7	Power systems	30
2.7.1	Hardware vs software	31
2.7.2	Hardware reset design	32
2.7.3	COP watchdog (Woof)	33
2.7.4	Power failure and brownout protection	34
2.8	Crystals and oscillator	34
2.8.1	Values vs baud rate	35
2.8.2	EMC and PCB crystal clock design	35
2.9	Conclusion	37
<hr/>		
3	Microcontroller programming	39
<hr/>		
3.1	Introduction to programming the microcontroller	39
3.2	Programming structure and specifications	41
3.2.1	Programming structures	42
3.2.2	Inputs	42
3.2.3	Manipulation of data	43
3.2.4	Outputs	44
3.2.4	Flow charts	44
3.2.6	Loading a program into an evaluation module (EVM)	48
3.2.7	Setting up the EVM	49
3.3	Addressing modes	49
3.4	Load, stores and transfers	50
3.5	Arithmetic operations	51
3.6	Logical operations	52
3.7	Shifts and rotates	53
3.8	Index registers and the stack	54
3.9	Condition code register	57
3.10	Branches, jumps, interrupts and calls	57
3.11	BASIC and C++	58
3.11.1	BASIC	58
3.11.2	Using C++ in embedded programming	58
3.12	Conclusion	58
<hr/>		
4	Microcontroller memory	60
<hr/>		
4.1	Introduction to memory	60
4.2	User RAM	61
4.2.1	Microcontroller internal RAM	62

4.2.2	External RAM	62
4.3	BUFFALO routines, memory map and vectors	64
4.3.1	BUFFALO as a development tool	64
4.3.2	BUFFALO utility subroutines	65
4.3.3	BUFFALO memory map	65
4.3.4	BUFFALO interrupt pseudo-vectors	66
4.4	Interrupts, vectors and pseudo-vectors	66
4.4.1	Software vs hardware interrupts	67
4.4.2	Maskable vs non-maskable interrupts	68
4.5	Control registers	70
4.5.1	Memory mapped I/O	70
4.5.2	Accessing and using control registers	70
4.6	EEPROM	71
4.6.1	Clearing the EEPROM example	72
4.6.2	Writing to the EEPROM example	73
4.7	Conclusion	74
5	Microcontroller inputs and outputs	76
5.1	Introduction to inputs and outputs	76
5.2	Single ended vs differential inputs	77
5.2.1	Single ended analog circuits	77
5.2.2	Single ended digital circuits	78
5.2.3	Differential analog circuits	79
5.2.4	Differential digital circuits	79
5.3	Digital inputs	80
5.3.1	Switch sensing and de-bounce	80
5.3.2	Normally open (NO) and normally closed (NC) switches	81
5.3.3	Electronic switches	81
5.4	Digital outputs	82
5.4.1	Digital control	82
5.4.2	Back EMF causes and solutions	83
5.5	Analog inputs	84
5.5.1	Voltage, current and resistive measurement	84
5.5.2	Analog and digital filtering and amplification	85
5.5.3	Nyquist and the sample rate	86
5.5.4	Resolution management	87
5.6	Digital control of analog devices	87
5.6.1	Basic stepper motors	87
5.6.2	Stepper motor control and communication	88
5.7	Keypad interfacing	88
5.7.1	Connecting the keypad to the evaluation modules (EVM)	88
5.7.2	Reading the keypad in software	89

5.8	LCD interfacing	91
	5.8.1 LCD software setup	92
	5.8.2 Writing to the LCD	94
5.9	Conclusion	95
6	Data communications	96
6.1	Introduction to data communication	96
6.2	Basics of serial data communication	97
	6.2.1 History of serial data communications	97
	6.2.2 Three parts of data communications	98
6.3	Open system interconnection model	100
	6.3.1 Application layer	101
	6.3.2 Session, presentation, transport and network layers	102
	6.3.3 Datalink layer	102
	6.3.4 Physical layer	102
	6.3.5 Protocols and the three layer model	103
6.4	Modes of communications	103
	6.4.1 Simplex	103
	6.4.2 Half-duplex	103
	6.4.3 Full-duplex	104
	6.4.4 The master slave bus	104
	6.4.5 The CSMA/CD bus	105
	6.4.6 The token bus system	106
	6.4.7 Timed systems	106
6.5	RS-232	107
	6.5.1 Introduction to RS-232	107
	6.5.2 Function of the lines	108
	6.5.3 RS-232 installation and troubleshooting	108
6.6	RS-485	111
	6.6.1 Introduction to RS-485	111
	6.6.2 RS-485 vs RS-422	112
	6.6.3 RS-485 installation and troubleshooting	113
6.7	Fiber optic cables	114
6.8	Fieldbus protocols used in controllers	115
6.9	Conclusion	116
7	Noise reduction	118
7.1	Introduction to noise reduction	118
	7.1.1 The decibel	119
	7.1.2 Signal to noise ratio	119
	7.1.3 Sources of noise – internal vs external	119
	7.1.4 Single ended or grounded circuits	121

7.1.5	Single ended measurement of grounded sources	121
7.1.6	Single ended grounded equipment	122
7.1.7	Differential noise circuits	123
7.1.8	Differential test equipment	124
7.1.9	Common mode noise problems	125
7.1.10	Low impedance drops as noise sources	126
7.1.11	Types of externally induced noise	127
7.2	Conductive coupled noise	127
7.2.1	Conductive noise from external equipment	128
7.2.2	Conductive noise from transmission lines	128
7.3	Capacitive coupled noise	129
7.3.1	Capacitive noise from adjacent equipment	129
7.3.2	Capacitive noise from communication lines	130
7.4	Magnetically coupled noise	130
7.4.1	Magnetically induced noise from adjacent cables	131
7.4.2	Magnetically induced noise from adjacent equipment	132
7.5	EMC and noise reduction in PCB design	132
7.5.1	Placement of analog, digital and power supply circuits	133
7.5.2	Digital circuit decoupling	134
7.5.3	Ground planes	135
7.5.4	1D and 3D Faraday shields	135
7.6	Conclusion	136
8	EMC grounding solutions	137
8.1	Introduction to EMC grounding solutions	137
8.2	EMC grounding	138
8.2.1	Ground specifications	138
8.2.2	Types of earth grounds	141
8.3	EMC grounding on a PCB	143
8.3.1	PCB design recommendations	143
8.3.2	Track placement	147
8.3.3	Faraday boxes	148
8.4	Protecting a PCB from lightning	150
8.4.1	Placement of protection on the PCB	151
8.4.2	The GDT, MOV and transorb	152
8.5	Microcontroller equipment ground	153
8.6	Enclosure or safety ground	154
8.6.1	Spiked earth grounds	154
8.6.2	Cable trench grounds	155
8.6.3	Tower lightning protection	156
8.7	Conclusion	157

9	Installation and troubleshooting	159
9.1	Introduction to installation and troubleshooting	159
9.2	Connections – screw, crimp and solder	160
9.2.1	Screw connectors	162
9.2.2	Crimp connectors	163
9.2.3	Soldering connections	164
9.2.4	Connector problems and solutions	164
9.3	Cable runs and trays	166
9.3.1	Metal vs plastic runs and trays	166
9.3.2	Vertical runs and trays	167
9.3.3	Horizontal runs and trays	168
9.4	Cable ties and mounting	169
9.5	Cooling, heating and air conditioning	170
9.6	Wire management in a cable run	171
9.7	Conduit installation	172
9.8	Troubleshooting techniques	173
9.9	Safety considerations	174
9.10	Conclusion	175
10	End notes	176
10.1	Conclusion	176
10.2	CPU design and functions	176
10.3	Assembly language programming	177
10.4	Memory	178
10.5	Inputs and outputs	178
10.6	Data communication	179
10.7	Noise reduction	180
10.8	Grounding solutions	181
10.9	Installation techniques	181
10.10	Final words	182
	Practicals	183
	Practical 1: Setting up the 68HC11 emulator board	183
	Practical 2: Activating LEDs on the EVM	191
	Practical 3: Reading switches on the EVM	197
	Practical 4: Sending characters to an LCD display	204
	Practical 5: Reading keypad input	212

Practical 6: Using the PAT software	221
Practical 7: Viewing character data transmission	226
Practical 8: Troubleshooting a data communication system	233
Practical 9: Troubleshooting a protocol problem	236
 Bibliography	 242
 Index	 243

Introduction

Objectives

When you have completed this chapter, you will be able to:

- Describe the basic parts and functions of microcontrollers
- Explain what assembly language is and how it is used
- Describe memory mapping
- Describe the basics of inputs and outputs
- Describe what types of data communications controllers use
- Explain noise reduction and its relationship to good signals
- Describe potential grounding problems

1.1 Microcontroller introduction

Embedded controllers are used in most commercial and industrial electronic equipment. The sheer volume of embedded controllers used in the world drives us to understand how they work and then how to troubleshoot and repair them. The microcontrollers and support chips used in these controllers are becoming smarter and easier to use. This is bringing the design and use of embedded controllers to more and more engineers hence the need for a good understanding of what embedded controllers are and how to troubleshoot them.

Embedded controllers are intelligent electronic devices used to control and monitor devices connected to the real world. This can be a microwave oven, programmable logic controller (PLC), distributed control system (DCS) or a smart sensor. These devices are used in almost every walk of life today. Most automobiles, factories and even kitchen appliances have embedded controllers in them. As time goes on and electronic devices get smarter and smaller, the embedded controller will be in or associated with everything we touch throughout the day.

Early embedded controllers contained a CPU (central processing unit) and a multitude of support chips. As time went on, support chips were included in the CPU chip until it became a microcontroller. A microcontroller is defined as a CPU plus random access memory (RAM), electrically erasable programmable read only memory (EEPROM),

inputs/outputs (I/O) and communications (Comms). The embedded controller is a microcontroller with peripherals such as keypads; displays and relays connected to it and is often connected to other embedded controllers by way of some type of communication system.

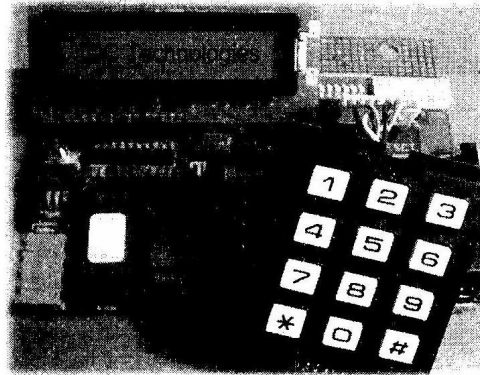


Figure 1.1
Embedded controller development board

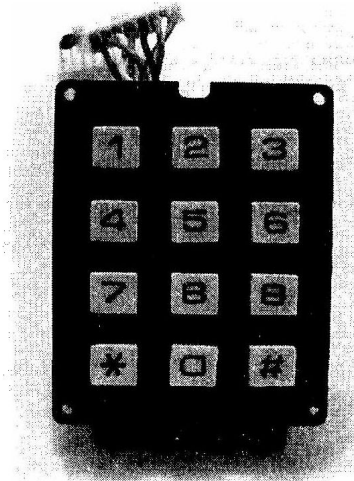


Figure 1.2
Keypad for embedded controller

Electronic equipment is becoming more and more susceptible to noise and other outside influences that can cause catastrophic problems. To be able to troubleshoot and ultimately repair the embedded controller it is not only necessary to understand the inter-workings of the embedded controller but also the external forces that can affect the normal operation of the controller. This may be noise, bad connections or incorrect installation of the system. Often simple things like bad grounds or incorrectly made connections can cost the user hundreds, if not thousands of dollars in down-time. Although the embedded controller ultimately can be a complicated device, when disassembled into its basic parts it becomes simple, clear and easy to understand.

1.2 Microcontroller design and functions

The microcontroller is a direct descendent of the CPU, in fact every microcontroller has a CPU as the heart of the device. It is therefore important to understand the CPU in order to ultimately understand the microcontroller and embedded controller.

The central processor unit (CPU) is the brain of the microcontroller. The CPU controls all functions and uses the program that resides in RAM, EEPROM or EPROM to function. The program may reside in one or more of these devices at the same time. Part of the program might be in RAM while another might be in EEPROM.

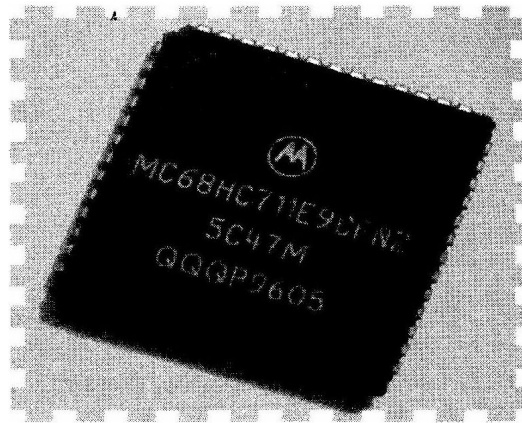


Figure 1.3
68HC11 CPU

A program is a sequence of instructions that tell the CPU what to do. These instructions could be compared to instructions a teacher may give to a student to get a desired result. The instructions sent to the CPU are very, very simple and it usually takes many instructions to get the CPU to do what is necessary to accomplish a task. Upper level programming languages like BASIC and C++ include multiple instructions in one command to speed up the process of programming the CPU. Just like the human brain the CPU is made up of regions that have specific functions. These components are controlled by the program instructions.

The main components of the microcontroller are as follows:

- CPU
- External address bus
- External data bus
- External control bus
- Internal RAM
- Internal ROM
- Internal EPROM
- Internal EEPROM
- Internal registers
- Digital inputs
- Counter inputs
- Digital outputs
- Analog inputs
- Serial data communications
- Parallel ports

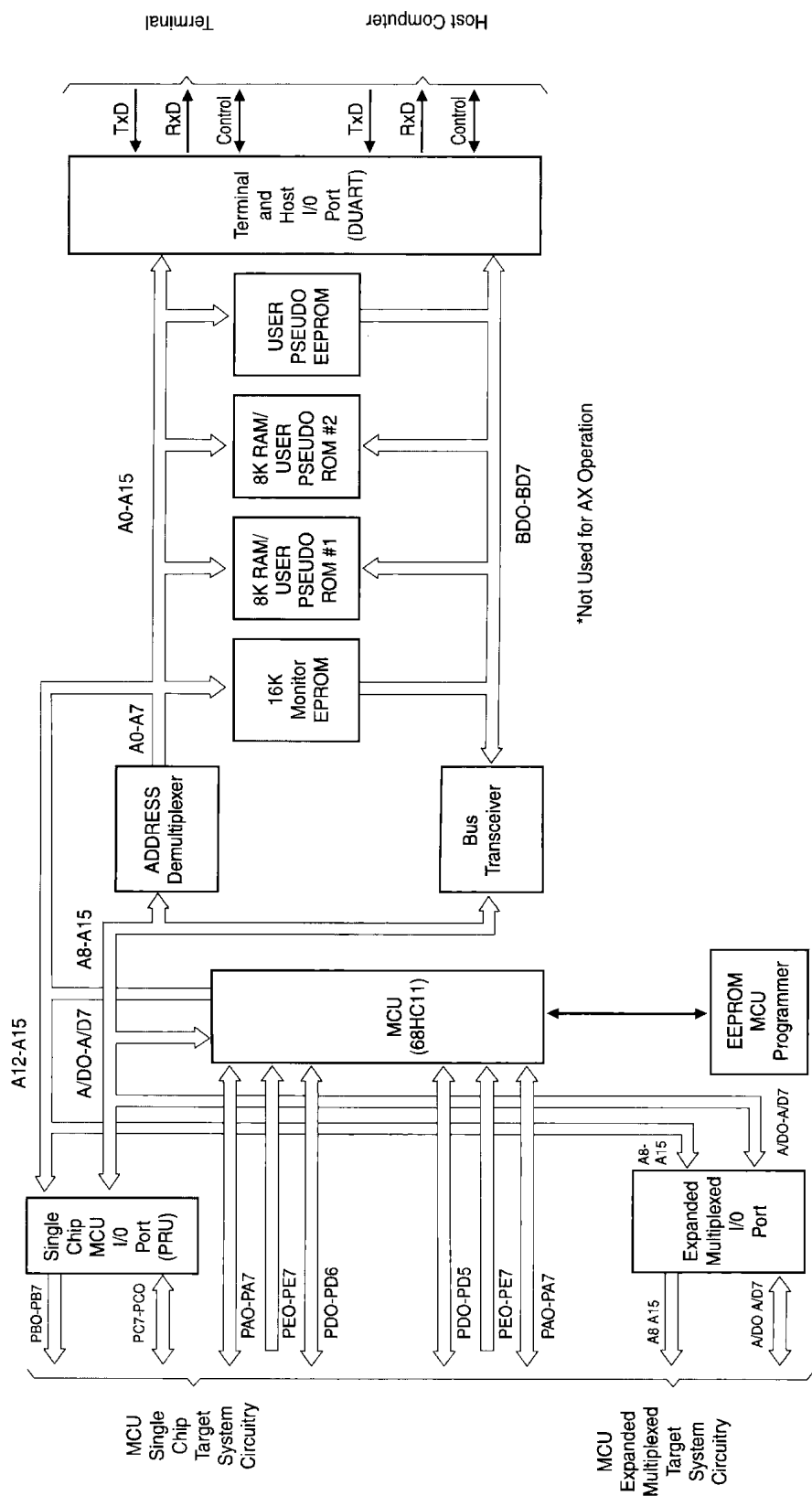


Figure 1.4
Block diagram of a microcontroller