Principles of MICROCOMPUTERS and MICROCONTROLLER ENGINEERING

SECOND EDITION
International Version

FREDRICK M. CADY
Department of Electrical and Computer Engineering
Montana State University

OXFORD
UNIVERSITY PRESS

OXFORD
UNIVERSITY PRESS

YMCA Library Building, Jai Singh Road, New Delhi 110001

Oxford University Press is a department of the University of Oxford.
It furthers the University's objective of excellence in research, scholarship,
and education by publishing worldwide in

Oxford New York
Auckland Cape Town Dar es Salaam Hong Kong Karachi
Kuala Lumpur Madrid Melbourne Mexico City Nairobi
New Delhi Shanghai Taipei Toronto

With offices in
Argentina Austria Brazil Chile Czech Republic France Greece
Guatemala Hungary Italy Japan Poland Portugal Singapore
South Korea Switzerland Thailand Turkey Ukraine Vietnam

Oxford is a registered trade mark of Oxford University Press
in the UK and in certain other countries.

Adapted from a work originally published by Oxford University Press Inc.

This international version has been customized for South and South-East Asia and
is published by arrangement with Oxford University Press, Inc. It may not be sold elsewhere.

Second edition 2009
This international version 2009

ISBN-13: 978-0-19-806226-4
ISBN-10: 0-19-806226-5

Printed in India by Saurabh Printers Pvt. Ltd, Noida 201301
and published by Oxford University Press
YMCA Library Building, Jai Singh Road, New Delhi 110001

Contents

1 Introduction

1.1 Computers, Microprocessors, Microcomputers, Microcontrollers

A computer system is shown in Figure 1-1. We see a *CPU,* or *central processor unit*, *memory (ROM* and *RAM)*, containing the program and data, an *I/O interface* with associated *input and output ports*, and three *buses* connecting the elements of the system together. The organization of the program and data into a single memory block is called a *von Neumann* architecture, after John von Neumann, who described this general-purpose, stored-program computer in 1945. In Figure 1-1 the data, address, and control buses consist of many wires, for example 8, 16, 32

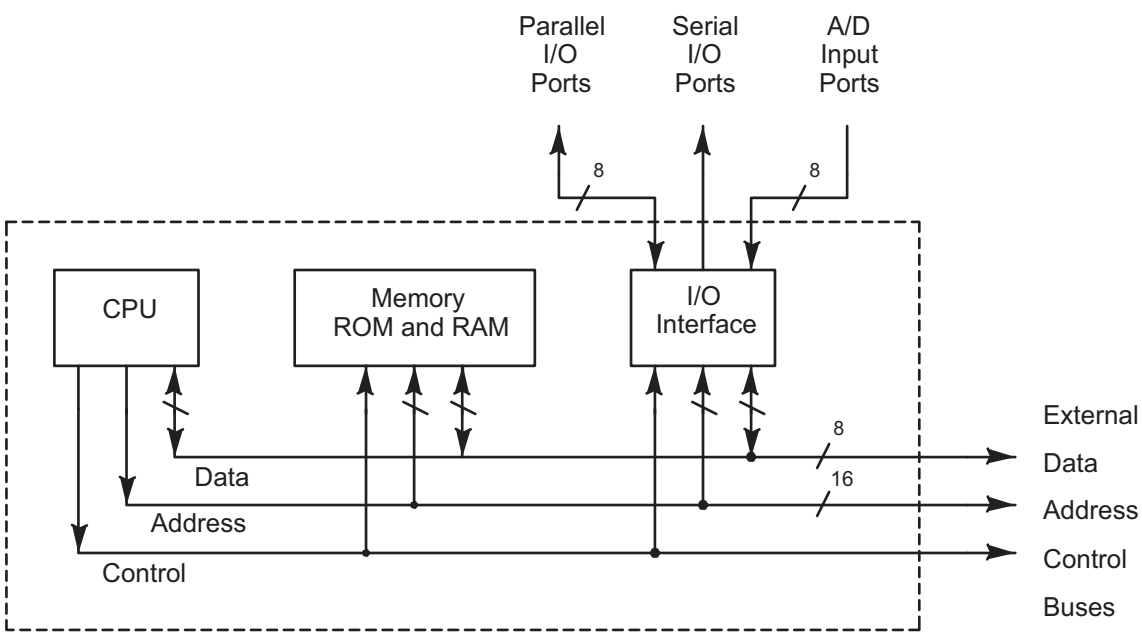

Figure 1-1 Von Neumann computer architecture.

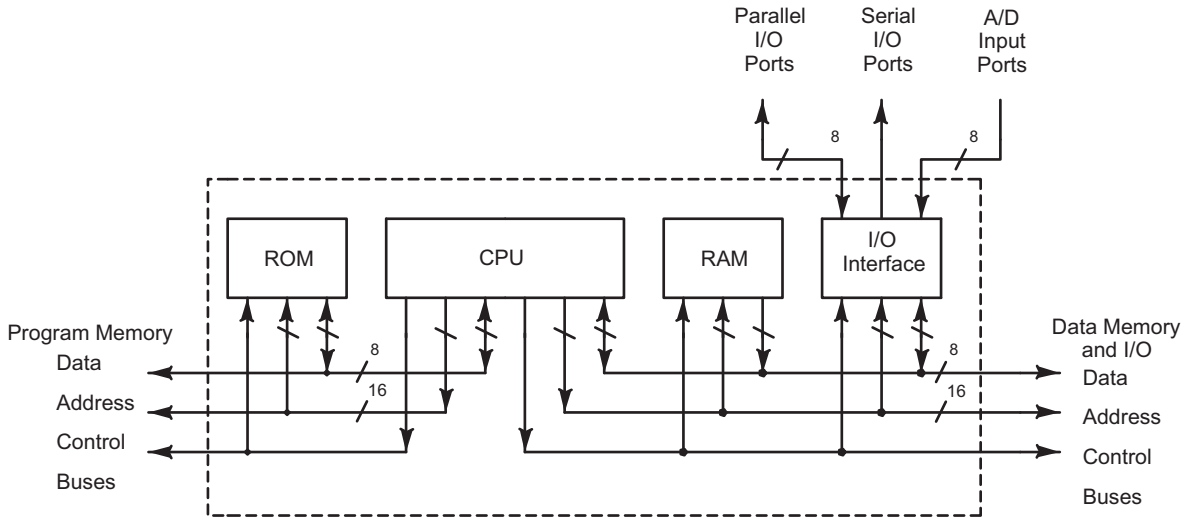

Figure 1-2 Harvard computer architecture.

or more, that carry binary signals from one place to another in the computer system. This is a classical computer system block diagram, and all computers discussed in this text have this basic architecture.

There is another major computer architecture type called the *Harvard* architecture in which two completely separate memories are used—one for the program and one for the data. This architecture is often found in digital signal processing (DSP) chips and some other microcontroller chips such as Microchip Technology PIC microcontrollers (Figure 1-2).

A *microcomputer* is a microprocessor with added memory and I/O.

Until 1971, when the Intel Corporation introduced the first microprocessor, the 4004, the CPU was constructed of many components. Indeed, in 1958 the Air Force SAGE computer required 40,000 square feet and 3 megawatts of power; it had 30,000 tubes with a 4K x 32 bit word magnetic core memory. The first mass-produced minicomputer, the Digital Equipment Company's PDP-8, appeared in 1964. This was the start of a trend toward less expensive, smaller computers suitable for use in nontraditional, non–data processing applications. Intel's great contribution was to integrate the functions of the many-element CPU into one (or at most a few) integrated circuits. The term *microprocessor* first came into use at Intel in 1972[1] and, generally, refers to the implementation of the central processor unit functions of a computer in a single, large scale integrated (LSI) circuit. A *microcomputer*, then, is a computer built using a microprocessor and a few other components for the memory and I/O. The Intel 4004 allowed a four-chip microcomputer consisting of a CPU, a read-only memory (ROM) for program, read/write memory (RAM) for data (using the Harvard architecture), and a shift register chip for output expansion.

The Intel 4004 was a 4-bit microprocessor and led the way to the development of the 8008, the first 8-bit microprocessor, introduced in 1972. This processor had 45 instructions, a 30-microsecond average instruction time, and could address 16 kilobytes of memory. Today, of course, we have advanced far beyond these first microcomputers. Table 1-1 gives a summary time line of many of the important developments leading to our microcontrollers of today.

[1] R. N. Noyce and M. E. Hoff Jr., *A History of Microprocessor Development at Intel*. IEEE MICRO, February 1981.

Table 1-1 Microcomputer Development Time Line

Year	Computer	Event
mid-1800s	Charles Babbage difference engine	A difference engine was completed in 1991 at the Science Museum in London to Babbage's original plans. It had around 4000 parts and weighed almost 3 tons. It successfully calculated a result to 31 digits.
1944	IBM Automatic Sequence Controlled Calculator	Also called the Harvard Mark I computer, it introduced the Harvard architecture with separate data and program memory. Built with switches, relays, and other mechanical components, it had over 700,000 components, and weighed 10,000 pounds.
1945	von Neumann machine described	While working on the EDVAC computer project, John von Neumann described a stored-program computer with data and program in the same memory.
1946	ENIAC	Electronic Numerical Integrator and Computer. With over 17,000 vacuum tubes and 7200 crystal diodes, it weighed 27 tons and consumed 150 kW of power.
1947	Point contact transistor invented	John Bardeen and Walter Brattain at AT&T Bell Labs.
1948	Junction transistor invented	William Shockley at AT&T Bell Labs.
1951	EDVAC	The Electronic Discrete Variable Automatic Computer was a successor to ENIAC. It computed in binary instead of decimal.
1951	Magnetic core memory invented	Jay Forrester at MIT based his invention on work by An Wang at Harvard University in 1949.
1958	Integrated circuit invented	Jack Kilby at Texas Instruments.
1960	MOS transistor invented	John Atalla and Dawon Kahng at AT&T Bell Labs and Robert Noyce at Fairchild Semiconductor.
1963	CMOS transistor invented	C. T. Sah and Frank Wanlass; Fairchild R & D Laboratory.
1964	First static RAM	64-bit memory, from Fairchild Semiconductor.
1964	PDP-8	Digital Equipment Corporation's first mass-produced minicomputer.
1964	Control Data Corporation CDC 6600	First reduced instruction set computer (RISC).
1965	Moore's law proposed	Gordon Moore at Fairchild Semiconductor predicted that the number of components per chip would double every one to two years.
1970	Intel 1103	First dynamic RAM chip, 1 Kbit.
1970	Three-state logic invented	National Semiconductor (now identified by trademark name Tristate)
1971	Intel 4004	First microprocessor: 2300 transistors, 740 kHz clock.
1971	Intel 1702	First erasable programmable read-only memory (EPROM); 256 x 8 bits.
1972	Intel 8008	First 8-bit microprocessor: 3500 transistors, 800 kHz clock.
1972	Hewlett-Packard HP-35	First pocket scientific calculator.
1973	IMP-16	First multichip 16-bit microprocessor; from National Semiconductor. It used five integrated circuits.
1974	PACE	First single-chip, 16-bit microprocessor; from National Semiconductor.
1974	Intel 8080	6000 transistors, 2 MHz clock.
1975	MIT's Altair 8800 computer	First hobbyist computer based on the Intel 8080. It had 4K and 8K BASIC, 4 K RAM, and introduced the S-100 bus standard. The complete kit, including extra memory and I/O, cost $1400 ($5800 in 2008 currency adjusted for inflation).
1976	RCA 1802	RCA COSMAC, the first CMOS microprocessor, was used in space flights in the 1970s.
1977	Commodore Pet	First all-in-one home computer with 4–8 K RAM, a 20 x 25 character display, and built-in cassette for data storage. It used the Mostek 6502 processor. It cost $795 ($2800 in 2008 currency adjusted for inflation).
1977	Apple II computer	Preceded by the Apple I in 1976, this became Apple's highly successful home computer. It cost $1298 with 4 K RAM and $2638 with 48K ($4680 and $9509, respectively, in 2008 currency).

continued

Table 1-1 *Continued*

Year	Computer	Event
1977	Radio Shack TRS-80	One of the first mass-produced home computers. It cost $600 ($2030 in 2008 currency).
1978	Intel 8086	Intel's first 16-bit microcontroller: 29,000 transistors, 4.77 MHz clock.
1978	Motorola 6801	First microcontroller: 3500 transistors with 2 MHz clock. It was the first integration of an 8-bit CPU with 128 bytes of RAM, 2 Kbyte of ROM, a 16-bit timer, and serial I/O interface.
1978	First EEPROM	Intel 2816: 2 Kbyte.
1979	Motorola 68000	First 32-bit microprocessor: 68,000 transistors, 8 MHz clock. It had 32-bit registers but 16-bit internal and external data bus and 24-bit address bus.
1980	BELLMAC-32A	First single-chip, 32-bit microprocessor at AT&T Bell Labs; 146,000 transistors.
1980	Intel 8087	Math coprocessor to do floating point arithmetic.
1981	IBM Personal Computer introduced	Intel 8088 with 4.7 MHz clock, ROM BASIC, up to 640K RAM, CGA display adapter, and cassette. A 160 Kbyte floppy was optional. Its $3000 cost in 1981 is equivalent to $7400 in 2008.
1981	iAPX432	Intel's first 32-bit microprocessor. Three chips with a total of 200,000 transistors. It had an 8 MHz clock.
1981	Osborne I	First commercially successful portable computer. It weighed 23.5 pounds and had the CP/M II operating system, a 5-inch display, 64K memory, and 5.25-inch floppy disk. It cost $1795 ($4460 in 2008 currency).
1982	First RISC processor	Reduced instruction set computer produced by the RISC Project at the University of California at Berkeley; 44,500 transistors.
1982	Intel 80286	16-bit microprocessor: 134,000 transistors, 6 MHz clock.
1983	Compaq Portable	First IBM PC compatible portable computer. It cost $3950 ($8400 in 2008 currency) and weighed 28 pounds.
1984	Flash EEPROM developed	Toshiba.
1984	First Apple Macintosh computer	It used an 8 MHz Motorola 68000 microprocessor, 128K RAM, and a 400 Kbyte 3.5-inch floppy. It cost $2495 ($5130 in 2008 currency).
1984	Motorola 68020	32-bit version of the 68000 microprocessor fabricated in CMOS: 190,000 transistors and 16 MHz clock.
1985	Intel 80386	32-bit microprocessor: 275,000 transistors, 16 MHz clock.
1989	Intel 80486	32-bit microprocessor: 1.2 million transistors, 25 MHz clock.
1990	FCC Part 15, Subpart B	Rules governing radiofrequency emissions for electronic equipment including personal computers. These federal rules require testing and certification of electronic equipment.
1992	IBM PowerPC	First single-chip PowerPC reduced instruction set computer; 32 bits 2.8 million transistors, 68 MHz clock.
1996	DEC Alpha 21064	Digital Equipment Corporation, 64-bit pipelined processor, 9.7 million transistors, 500 MHz clock.
2000	Intel Pentium IV	64-bit microprocessor: 42 million transistors, 1.4 GHz clock.
2005	AMD Athlon 64	64-bit microprocessor: 200 million transistors, 2.6 GHz clock.
2008	AMD Phenom	64-bit microprocessor: 450 million transistors, 3 GHz clock.

1.2 Moore's Law

Table 1-1 shows a remarkable, exponential growth rate in the size and speed of the integrated circuits used in microprocessors and microcontrollers. In 1965 Intel cofounder Gordon Moore observed this phenomenon and predicted that the growth would continue doubling every 18 to 24 months. Although some observers claim this is a self-fulfilling prophecy because

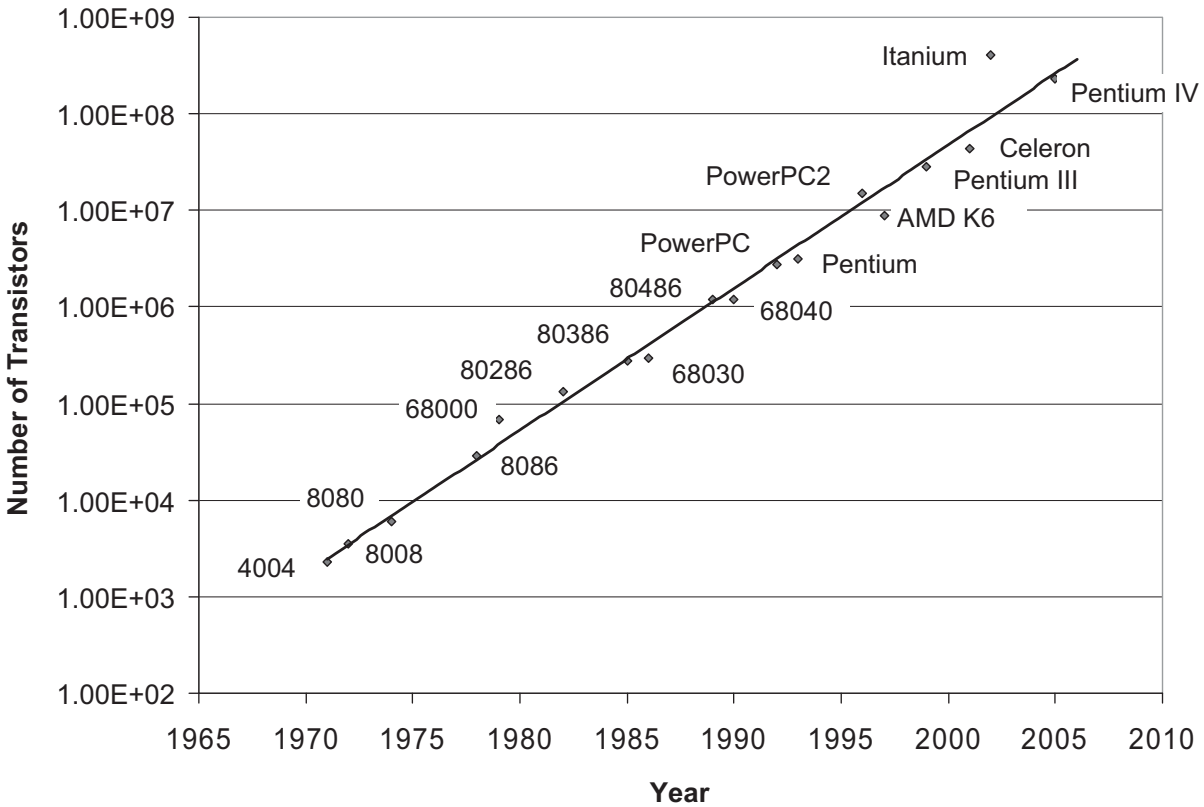

Figure 1-3 Growth in number of transistors in microprocessors from late 1960s to first decade of the twenty-first century.

manufacturers concentrate on improving their technology, Moore's now four-decade-old observation has continued to be true, as shown in Figures 1-3 and 1-4.

1.3 Microcontrollers

A *microcontroller* is a computer with *CPU, memory,* and *I/O* in one integrated circuit chip.

This text primarily is about using computers in applications where the system is dedicated to performing a single task or a single group of tasks. These are called embedded applications, and examples are found almost everywhere in products from microwave ovens and toasters to automobiles. These are often *control* applications and make use of microcontrollers. A *microcontroller* is a microcomputer with its memory and I/O integrated into a single chip. In 1991 the chip manufacturers delivered over 750 million 8-bit microcontrollers; by 2004 the industry's annual total was 6.8 billion microcontroller units.[2]

[2] http://www.instat.com/press.asp?ID=1445&sku=IN0502457SI

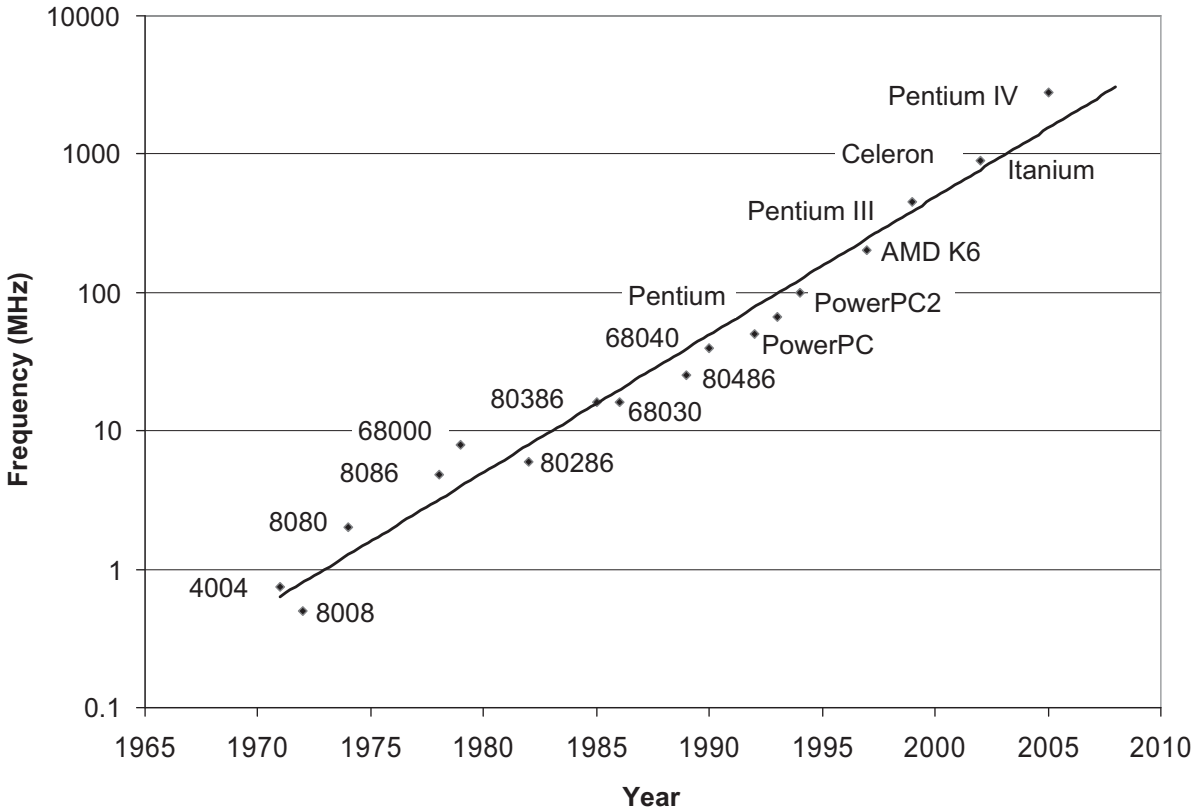

Figure 1-4 Improvements in microprocessor clock frequency for the same period.

1.4 Some Basic Definitions

Throughout this text we use the following digital logic terminology.

Active high: Used to define a signal whose assertion level is logic high.

Active low: This term defines a signal whose assertion level is logic low. For example, the signal `READ_L` is asserted low. Although many data sheets and schematic diagrams make use of an overbar or some other notation, in this text we will denote active-low signals by adding the "_L" suffix to the signal name.

Assembly/Compile time: The time at which our programs are assembled or compiled. Quantities known at this time can be saved as constants in program memory (ROM). In an embedded system, variable data must not be initialized at assembly/compile time.

Assert: Logic signals, particularly signals that control a part of the system, are asserted when the control, or action named by the signal, is being done. A signal may be low or high when it is asserted. For example, the signal `WRITE` indicates assertion when the signal is logic high.

Byte: A byte is 8 bits.

Device loading: The device loading is an indication of what is connected to a device's output. It determines the output voltage and current requirements of the device.

EEPROM: Electrically erasable programmable read-only memory—pronounced "*double e prom*". This is an EPROM that can be erased by an electrical signal, eliminating the need to remove the chip from its circuit and exposing it to UV light, as is the case for EPROM.

EPROM: Erasable programmable read-only memory. First introduced by Intel in 1971, this PROM could be erased by exposing it to ultraviolet (UV) light. Erasable PROMs have a quartz window to allow the UV light into the package.

Fan-out: Fan-out is the number of similar devices one device's output can drive.

Flash EEPROM: EEPROM may be erased and written to one byte at a time. Flash allows data to be erased and written in blocks and is thus faster than EEPROM. Flash is used mostly for program memory and EEPROM for variable data that must be retained when the power is removed. Note that Flash is sometimes called Flash EEPROM.

Logic high: The higher of the two voltages defining logic true and logic false. The value of a logic high depends on the logic family. For example, in the HCMOS family, logic high (at the input of a gate) is signified by a voltage greater than 3.15 V. This voltage is known as V_{ihmin}.

Logic low: The lower of the two voltages defining logic true and false. In HCMOS, a logic low (at the input of a gate, V_{ilmax}) is signified by a voltage less than 1.35 V.

Logical complement: The complement of a logical signal is an operator. We will use the overbar to donate the complementation. Thus, $\overline{\texttt{PUMP\_ON}}$ is the complement of the active-high signal `PUMP_ON`.

Mixed-polarity notation: The notation used by most manufacturers of microcomputer components defines a signal by using a name, such as `WRITE`, to indicate an action, and a polarity indicator to show the assertion level for the signal. Thus, the signal `WRITE` indicates that the CPU is doing a write operation when the signal is high. `READ_L` denotes that a read operation is going on when the signal is low.

Nibble: A nibble is 4 bits. There are two nibbles for each byte.

OTP EPROM: One-time-programmable EPROM. This is an EPROM without the quartz window; thus it cannot be erased after it has been programmed.

Positive and negative edge trigger: Data latches may operate on a level or edge-triggered basis. There are positive (rising) and negative (falling) edge-triggered devices.

PROM: Programmable read-only memory. Memory that can be programmed by the user instead of at the factory, as must be done for ROM.

RAM: Random access memory. This memory can be read from and written to and is used in the microcontroller for variable data storage. The memory contents are lost when the power is removed. Therefore the memory is said to be volatile.

ROM: Read-only memory. The contents of this memory is programmed once, at the time of manufacture, and is nonvolatile. That is, the memory contents persist when the power is removed. ROM is used in microcontrollers for program storage.

Run time: This is when our program executes. Any variable data with initial values must be initialized at run time.

Tristate or three-state: A logic signal that can neither source nor sink current. It presents a high impedance load to any other logic device to which it is connected.

Word: A word is 16 bits.

Table 1-2 Notation

0x	Hexadecimal numbers are denoted by a leading 0x (e.g., 0xFFFF is the hexadecimal number FFFF).
	When two memory locations are to be identified, the starting and ending addresses are given as 0xFFFE:FFFF.
$	Hexadecimal numbers in Freescale assembly language examples use a $ to denote a hexadecimal number. $0F = 15.
%	Binary numbers are denoted by a leading %. For example, 0xF may be written %1111.
@	A base-8 or octal number is preceded by @. Thus 0xF = @17.
	Base 10 is the default base; unlike hexadecimal, binary or octal, it has no base indicator. Thus 0xF = 15.
0b	In C programs, the 0b prefix is used to signify a binary number.
x	An "x" indicates a don't-care bit—that is, the bit may be zero or one.
*	The "*" indicates a pointer in a C program.
_L	A signal whose assertion level is low is followed by "_L."

1.5 Notation

Throughout this text, the notation shown in Table 1-2 is used.

1.6 Study Plan

The designs of embedded application systems and other more general-purpose computers are very similar. Our goal for this course is not to make you an expert in using a specific processor, but to give you the knowledge and tools to be able to effectively apply any processor in any application. We will do that by first studying the general principles necessary to understand each part of the system. You may then turn to the user's manual for a specific processor and be able to more easily understand the information there and apply it in an application.

The basic operation of a stored-program, general-purpose computer is to be studied first. You'll learn about registers, the arithmetic and logic unit, and how a computer works. Because much of your work in an introductory microprocessor/microcontroller course is likely to be learning the language and programming exercises, we introduce you to structured program design in Chapter 3. Designing software before writing it is vital in developing debuggable application software. We will guide you through an introduction to the central processor unit and how it addresses memory in Chapters 4 and 5 and introduce assembly language programming in Chapter 6. You will need to study your own processor in parallel while reading these chapters.

Many embedded applications are written in C, which you may have learned in another programming class. A program written in C for an embedded application, however, has some significant differences from one written for a desktop computer. Chapter 7 will help you learn about these differences. Chapter 8 discusses debugging techniques helpful for assembly and C language programs.

Chapters 9 through 15 cover the basics of parallel and serial I/O, interrupts, memory, analog I/O, timers, and interfacing techniques for single-chip microcontrollers. Chapter 16 touches on real-time operating systems.