MICROPROCESSORS and their OPERATING SYSTEMS

A Comprehensive Guide to 8-, 16- and 32-Bit Hardware, Assembly Language and Computer Architecture

R. C. HOLLAND

PERGAMON PRESS

Microprocessors and their Operating Systems

A Comprehensive Guide to 8-, 16- and 32-Bit Hardware, Assembly Language and Computer Architecture

R. C. HOLLAND
West Glamorgan Institute of Higher Education, Swansea, UK

PERGAMON PRESS
OXFORD · NEW YORK · BEIJING · FRANKFURT
SÃO PAULO · SYDNEY · TOKYO · TORONTO

U.K.	Pergamon Press plc, Headington Hill Hall, Oxford OX3 0BW, England
U.S.A.	Pergamon Press, Inc., Maxwell House, Fairview Park, Elmsford, New York 10523, U.S.A.
PEOPLE'S REPUBLIC OF CHINA	Pergamon Press, Room 4037, Qianmen Hotel, Beijing, People's Republic of China
FEDERAL REPUBLIC OF GERMANY	Pergamon Press GmbH, Hammerweg 6, D-6242 Kronberg, Federal Republic of Germany
BRAZIL	Pergamon Editora Ltda, Rua Eça de Queiros, 346, CEP 04011, Paraiso, São Paulo, Brazil
AUSTRALIA	Pergamon Press Australia Pty Ltd., P.O. Box 544, Potts Point, N.S.W. 2011, Australia
JAPAN	Pergamon Press, 5th Floor, Matsuoka Central Building, 1-7-1 Nishishinjuku, Shinjuku-ku, Tokyo 160, Japan
CANADA	Pergamon Press Canada Ltd., Suite No. 271, 253 College Street, Toronto, Ontario, Canada M5T 1R5

First English edition 1989

Library of Congress Cataloging-in-Publication Data

Holland, R. C.
Microprocessors and their operating systems:
a comprehensive guide to 8-, 16- and 32-bit hardware, assembly language and computer architecture
R. C. Holland.—1st ed.
p. cm.—(Applied electricity and electronics series)
Includes index.
1. Microprocessors. 2. Operating systems (Computers). I. Title. II. Series.
QA76.5.H633 1989 004.16—dc 19 88-38877

British Library Cataloguing in Publication Data

Holland, R. C.
Microprocessors and their operating systems:
a comprehensive guide to 8-, 16- and 32-bit hardware, assembly language and computer architecture
(Applied electricity and electronics)
I. Title II. Series
004.16

ISBN 0–08–037189–2 Hardcover
ISBN 0–08–037188–4 Flexicover

Printed in Great Britain by BPCC Wheatons Ltd, Exeter

Microprocessors and their Operating Systems

A Comprehensive Guide to 8-, 16- and 32-Bit Hardware, Assembly Language and Computer Architecture

APPLIED ELECTRICITY AND ELECTRONICS SERIES

Editor: P. HAMMOND, *University of Southampton, UK*

Pergamon Titles of Related Interest

DHEERE
Universal Computer Interfaces

HAMMOND
Electromagnetism for Engineers, 3rd Edition

HINDMARSH
Electrical Machines & Drives, Worked Examples, 2nd Edition
Electrical Machines & their Applications, 4th Edition

HOLLAND
Illustrated Dictionary of Microelectronics & Microcomputers
Integrated Circuits & Microprocessors
Microcomputers & their Interfacing

KUFFEL & ZAENGEL
High Voltage Engineering

YORKE
Electric Circuit Theory, 2nd Edition

Pergamon Related Journals (*free specimen copy gladly sent on request*)

Computers & Electrical Engineering

Electric Technology USSR

Microelectronics & Reliability

Robotics & Computer-integrated Manufacturing

Solid State Electronics

PREFACE

This book is an attempt to bring together all the essential features of each of the major microprocessor families that have evolved through the 1970s and 1980s. The main characteristics of each of the 8-bit, 16-bit and 32-bit microprocessors are presented, together with descriptions of applications and software considerations.

The book excludes some of the less-frequently applied microprocessors, but the concentration on the most common devices provides descriptions of the processors that are used in at least 95% of current microcomputer systems.

Most books on similar topics tend to concentrate on a single device. The comprehensive coverage of the subject presented here should provide a useful reference for the reader who requires a good general background to the whole subject. The later chapters in the book concentrate on programming languages (BASIC, Pascal and C), and in particular on operating systems (the main program in a multi-program microcomputer system). The descriptive notes on the facilities available within the most common operating systems should enable a user of a disc-based microcomputer to perform a complete range of operator functions.

An introductory chapter on microcomputer principles is provided to assist a reader who is new to the subject, and a glossary of terms at the end of the book should prove helpful.

The author wishes to thank several of his colleagues for helpful discussions, and his long-suffering students (past and present) who have often provided an experimental platform for the presentation style of much of the material.

CONTENTS

CHAPTER 1

MICROCOMPUTER PRINCIPLES

1.1 Microcomputer Architecture

Computers are generally classified as follows:

(a) mainframe computers, which support a large number of user terminals and peripherals and which are used for large data-handling systems, e.g. payroll, customer accounts;
(b) minicomputers, which support a smaller number of user terminals (typically up to 40) and which are used in small-company business applications as well as factory control systems;
(c) microcomputers, which normally support a single user (although latest systems have edged into the minicomputer market) and which are applied in a wide number of domestic, commercial and industrial control applications.

The distinguishing feature of a microcomputer is that its CPU (central processor unit) is a single IC (integrated circuit). This central component is termed "microprocessor". It is supported by memory and input/output ICs to constitute a complete microcomputer.

The basic architecture of all types of computer is shown in Fig. 1.1. The CPU, or microprocessor in the case of a microcomputer, generates the three buses, which are a set of circuit connections. The computer program that the CPU executes is held in memory, which in microcomputers is invariably arranged in 8-bit (a bit is 0 or 1) locations. The CPU extracts each instruction in turn from memory, examines it and executes it. The transfer of each instruction is as follows:

(1) CPU sets the memory address of the next instruction on the address bus;
(2) memory location presents the contents of the selected memory address on the data bus;
(3) CPU reads in the instruction and implements it.

The control bus carries signals that are required to activate the above transfers.

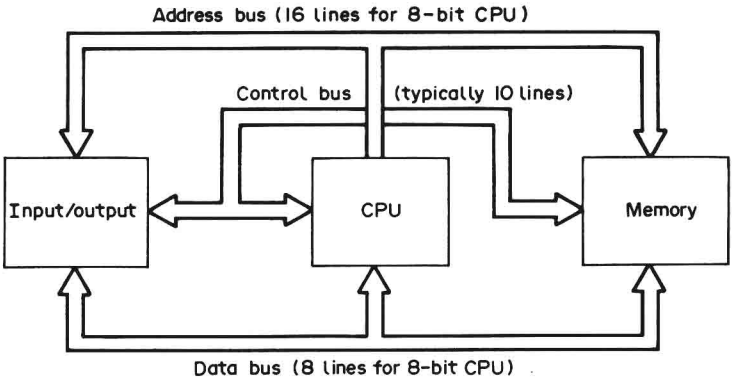


FIG. 1.1. Computer architecture.

Memory also contains data values in addition to program instructions. Data values can be numbers (represented in binary), or characters for display or printing purposes (represented by a unique code—the "ASCII" code).

The input/output module contains circuits that transfer programs and data to and from peripheral devices, e.g. backing store (such as floppy disc), printer and VDU (user terminal).

In the simplest forms of microcomputer, e.g. pocket calculator or washing machine controller, the three modules (CPU, memory and input/output) are contained within a single IC.

1.2 CPU (Central Processor Unit)

The typical internal organisation of a CPU is shown in Fig. 1.2. All actions are triggered by a clock pulse. The functions of the internal modules are as follows:

(a) The *program counter* contents are placed on the address bus so that the next program instruction can be transferred from memory to the CPU.
(b) The *instruction register* receives the instruction.
(c) The *control unit* decodes the instruction and activates transfers around the CPU in order to execute the instruction.
(d) The *work registers* hold data items that can be processed within the instruction (a specialised work register that can receive the results of ALU operations is termed an "accumulator").
(e) The *ALU* is the processing part of the CPU, i.e. it can alter data values using arithmetic (add, subtract, etc.) and logical (OR, AND, etc.) functions.

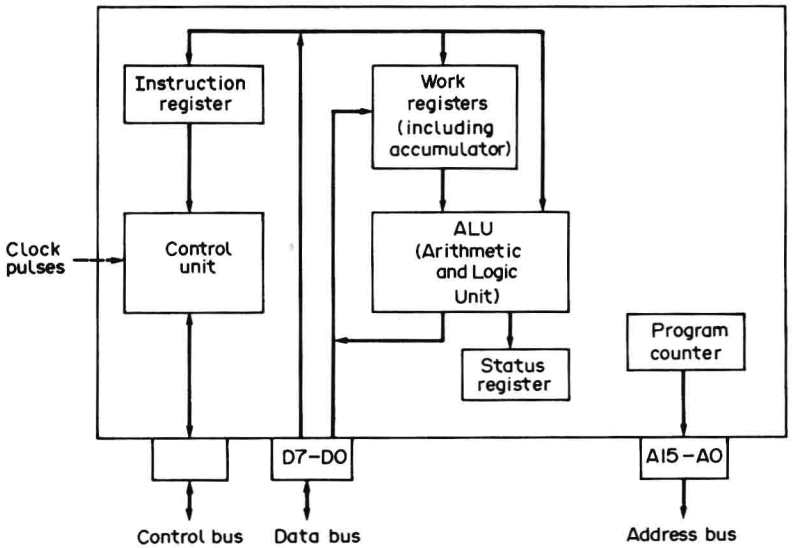


FIG. 1.2. Internal organisation of CPU.

(f) The *status register* contains "flags" that mark the result of ALU operations, e.g. zero result.

The word length of a CPU is expressed in terms of the number of bits that are processed in a single ALU operation, e.g. an 8-bit CPU can perform an arithmetic add operation on two 8-bit numbers. Microprocessor IC packages can be grouped as follows:

(a) 8-bit invariably 40-pin DIL (dual-in-line, i.e. pins on two sides only) package
(b) 16-bit normally 40- to 64-pin DIL package
(c) 32-bit between 60 and 120 pins in a "chip carrier" (pins on all four sides of package), or PGA (pin grid array).

The operation of each instruction is conveniently divided into two operations—"fetch" and "execute". Whilst the fetch part of the fetch–execute cycle is the same for every instruction, the execute part may or may not involve further memory transfers. Some instructions require only ALU operations in the execute part of the cycle, rather than slow memory transfer operations. Therefore the instruction time is variable and depends on the nature of the instruction. The following examples illustrate this:

Instruction Example 1—Add register B to register A
Step 1—Fetch instruction
Step 2—Execute (add B to A, the accumulator)

Instruction Example 2—Load register A from memory
Step 1—Fetch instruction
Step 2—Execute 1 (read first half of memory address into CPU)
Step 3—Execute 2 (read second half of memory address into CPU)
Step 4—Execute 3 (read data byte from memory address into A)

The second instruction takes approximately twice as long to operate as the first instruction. Notice that, in an 8-bit microprocessor, the first instruction is one byte (8 bits) long, i.e. it occupies one memory location. The second instruction is three bytes long, because the second and third bytes contain the 16-bit memory address of the data value that is being accessed. The second and third bytes in this case are termed the "operand" of the instruction, whilst the first byte that specifies the nature of the instruction is termed the "opcode".

The instruction set that can be obeyed by a CPU may number from 50 to several hundred different instructions. Instructions can be conveniently grouped into four categories, as follows:

(1) Data move, e.g. between a CPU register and another CPU register, a memory location or an input/output channel
(2) Data modify, i.e. when the ALU alters a data value
(3) Jump, i.e. transfer program control to another memory location
(4) Miscellaneous, e.g. to stop program execution.

Within (1) and (2) there are various ways of accessing a data value, and these are termed the "addressing modes" that can be specified within an instruction. The most common addressing modes are:

(a) Direct Register—data value is held in a CPU register
(b) Direct Memory—data value is held in a memory location
(c) Indirect Register—data value is held in a memory location specified within a CPU register (register-pair for an 8-bit CPU)
(d) Immediate—data value is held within the instruction (in the second byte for an 8-bit CPU)
(e) Indexed—similar to (c) except that a displacement within the instruction is added to the contents of the indexing register to compute the memory address.

1.3 Memory

Memory ICs are either ROM (read-only memory), which can only be read, or RAM (random access memory), which can be read from and written to. The pin connections of a typical ROM device are shown in Fig. 1.3. The 10 address lines give:

$2^{10} = 1024 \quad (1024 = 1K)$

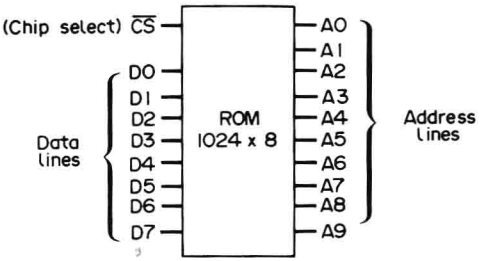


FIG. 1.3. Typical ROM.

memory locations, and the 8 data lines indicate that 8 bits are stored in each memory location. The "memory organisation" of the device is therefore declared to be:

1024×8

A byte is only read out of the device when the $\overline{CS}$ (Chip Select) signal is set to 0 (the bar above the CS legend denotes inverse logic, i.e. the signal is set when it is at logic 0).

The pin-compatible family of ROM devices is:

ROM—mask-prgrammed by the chip manufacturer
PROM—programmable ROM that is programmed by the user
EPROM—erasable PROM that can be erased using UV light and re-programmed.

The pin connections of RAM devices are similar, except that an additional CPU control bus signal (R/$\overline{W}$—Read/Write) must be connected to select the direction of byte transfer. There are two subdivisions of RAM:

Static RAM—holds its bit pattern until dc power is removed (i.e. it is "volatile").
Dynamic RAM—holds its bit pattern for only 2 ms unless a refresh circuit is applied

A complete microcomputer memory circuit is shown in Fig. 1.4. The ROM and RAM devices are connected to the data bus and also to as many of the least significant address bus lines as required. The Chip Select signals are generated by a "2 to 4 decoder", which ensures that only one memory device in the system can be selected at any time. The truth table for the decoder is shown in Fig. 1.5(a). Only one of the four outputs can be set to 0 at any time, and the $\overline{G}$ (Enable) input signal on the decoder must be set to 0 otherwise no

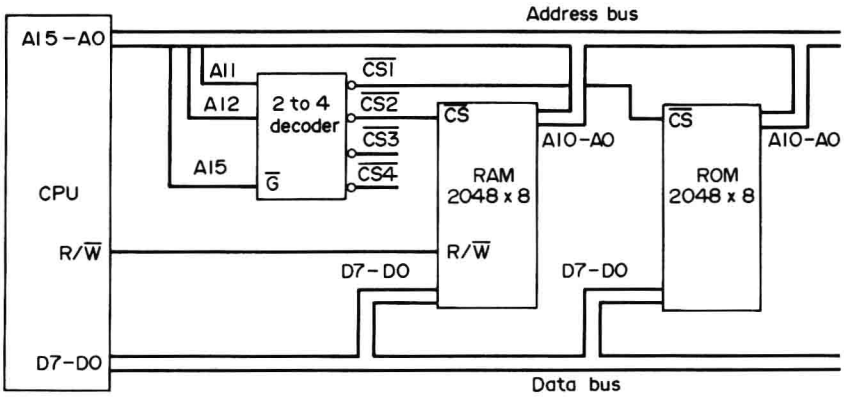


FIG. 1.4. Simple microcomputer memory circuit.

otherwise no output can be set. This particular memory circuit produces device addresses as follows:

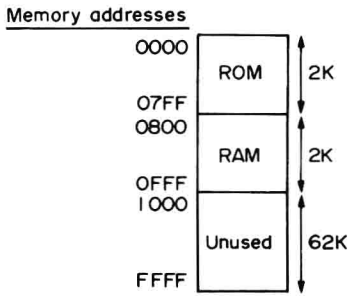
	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
ROM start	0	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	Hex 0000
ROM end	0	X	X	0	0	1	1	1	1	1	1	1	1	1	1	1	Hex 07FF
RAM start	0	X	X	0	1	0	0	0	0	0	0	0	0	0	0	0	Hex 0800
RAM end	0	X	X	0	1	1	1	1	1	1	1	1	1	1	1	1	Hex 0FFF

Decoder enable — 2 to 4 decoder — X = not used (assume set to 0)

Grouping the 16 bits into 4 hexadecimal characters gives the addresses shown on the right of this table. These addresses can be shown on a "memory map" for the computer as indicated in Fig. 1.5(*b*), and this map serves as a guide to the computer programmer to indicate where he can place his programs and data lists.

A12	A11	$\overline{CS4}$	$\overline{CS3}$	$\overline{CS2}$	$\overline{CS1}$
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

(a) Truth table for 2 to 4 decoder in Fig. 1.4



(b) Memory map

FIG. 1.5. Address calculations for Fig. 1.4.

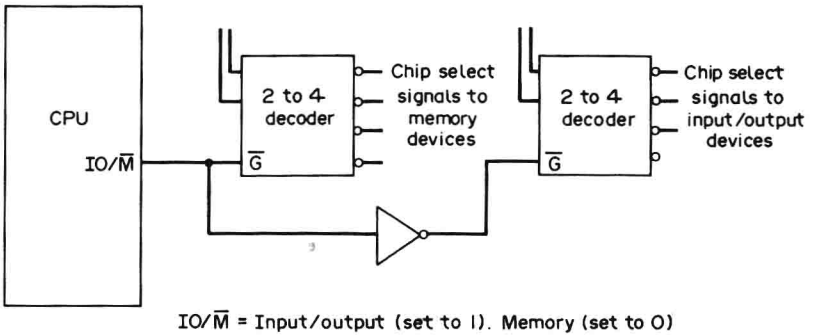


FIG. 1.6. Separate decoding circuits for memory and input/output.

If more than four memory devices exist in a memory circuit, a 3 to 8 decoder can be used in place of a 2 to 4 decoder, and the unused higher-order address bus lines must be connected at its inputs.

Several microprocessors do not possess input/output instructions, and input/output chips must be connected into circuit as if they are memory devices and addressed using memory transfer instructions. This arrangement is termed "memory mapped input/output". If a CPU does possess input/output instructions, memory and input/output devices must use separate decoders as shown in Fig. 1.6. In this arrangement memory and input/output chips can possess the same addresses, i.e. the memory map can overlap the input/output map. However no bus conflict can occur because the CPU sets the IO/$\overline{M}$ signal to 0 when performing memory transfers (to select the first decoder) and to 1 when performing input/output transfers (to select the second decoder).

1.4 Input/Output

Input/output chips are used to connect the microcomputer to a wide variety of peripheral equipment (backing store, printer, VDU), data links to other computers, instrumentation and electrical equipment in control applications, displays of various types, and other devices. Whilst memory chips (ROM and RAM) are supplied by a range of manufacturers who do not normally manufacture the CPU to which they connect, input/output devices are invariably tailored to the CPU and are offered as support devices by the CPU manufacturer.

The principal input/output chips are described in the following sections:

(a) PIO (Parallel Input/Output)

The function of a PIO is to pass data through 8-bit "ports" to and from remote devices. Figure 1.7 illustrates the pin functions of a typical PIO. The

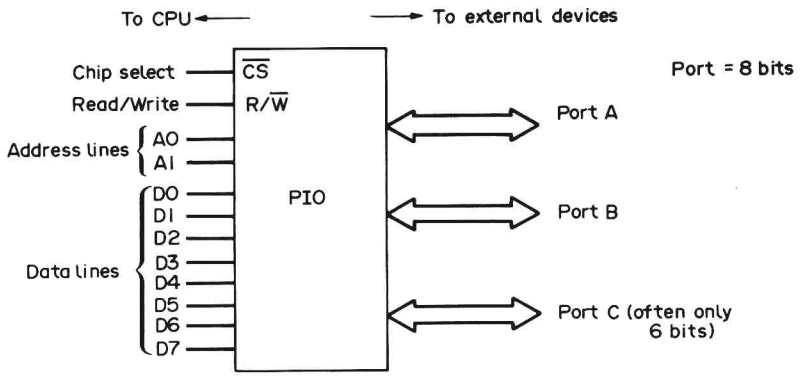


FIG. 1.7. PIO pin functions.

signal connections to the CPU buses are similar to that of a RAM chip. A data byte is transferred via the data bus, and typically two address bus lines are connected to provide four addresses on the PIO as follows:

A1	A0
0	0—Port A
0	1—Port B
1	0—Port C
1	1—Control register

The control register is used to select the directions of the ports, i.e. the programmer must place a control byte in the control register before bytes are transferred through the ports.

It should be noted that microprocessors, memory chips and programmable input/output chips are manufactured using the MOS (or CMOS) technology—this technology is often called VLSI (very large scale integration). However, a non-programmable alternative to a PIO is available in the form of an 8-bit register from the TTL family, which comprises ICs for gating and general digital handling circuits. A typical device that can be utilised as an input port is a SN74244, which possesses:

8 input lines —connect to external device/peripheral, e.g. 8 pushbuttons
8 output lines—connect to data bus
2 enable lines—one used as Chip Select and connected to address decoder

Typical applications of parallel output ports are:

(1) 8-bit connection to 7-segment display (+ decimal point)
(2) 1 to 8 bits to drive indicator LEDs/lamps or to drive electrical motor/heater/solenoid