MICROCOMPUTERS IN THE PROCESS INDUSTRY

E R ROBINSON, MA, PhD, CEng, FIChemE, MIEE
Head of Department of Chemical Engineering
North East London Polytechnic
Dagenham, Essex

ELLIS HORWOOD LIMITED
Publishers Chichester
Halsted Press a division of
JOHN WILEY & SONS
New York Chichester Brisbane Toronto

First published in 1985 by
ELLIS HORWOOD LIMITED
Market Cross House, Cooper Street, Chichester, West Sussex, PO19 1EB, England

The publisher's colophon is reproduced from James Gillison's drawing of the ancient Market Cross, Chichester.

Distributors:
Australia, New Zealand, South-east Asia:
Jacaranda-Wiley Ltd., Jacaranda Press,
JOHN WILEY & SONS INC.,
G.P.O. Box 859, Brisbane, Queensland 40001, Australia
Canada:
JOHN WILEY & SONS CANADA LIMITED
22 Worcester Road, Rexdale, Ontario, Canada.
Europe, Africa:
JOHN WILEY & SONS LIMITED
Baffins Lane, Chichester, West Sussex, England.
North and South America and the rest of the world:
Halsted Press: a division of
JOHN WILEY & SONS
605 Third Avenue, New York, N.Y. 10016, U.S.A.

British Library Cataloguing in Publication Data
Robinson, E.R.
Microcomputers in the process industry. –
(Ellis Horwood series in industry and applied technology)
1. Chemical engineering – Data processing
2. Microcomputers
I. Title
660.2'81 TP155

ISBN 0-85312-723-9 (Ellis Horwood Limited)
ISBN 0-470-20189-4 (Halsted Press)

Typeset by Ellis Horwood Limited.
Printed in Great Britain by Unwin Brothers of Woking.

MICROCOMPUTERS IN THE PROCESS INDUSTRY

ELLIS HORWOOD SERIES IN APPLIED SCIENCE AND INDUSTRIAL TECHNOLOGY

Series Editor: Dr D. H. SHARP, OBE, lately General Secretary, Society of Chemical Industry; formerly General Secretary, Institution of Chemical Engineers; and former Technical Director, Confederation of British Industry

Published and in active publication

PREPARATION, PROPERTIES AND INDUSTRIAL APPLICATIONS OF ORGANOFLUORINE COMPOUNDS
Editor: R. E. BANKS, Department of Chemistry, University of Manchester, Institute of Science and Technology
CERAMIC WHITEWARES
A. DINSDALE, lately Director of Research, British Ceramic Research Association
MATCHMAKING: Science, Technology and Manufacture
C. A. FINCH, Managing Director, Pentafin Associates, Chemical, Technical and Media Consultants, Stoke Mandeville, and S. RAMACHANDRAN, Senior Consultant, United Nations Industrial Development Organisation for the Match Industry
WATER – ABSTRACTION, STORAGE, TREATMENT, DISTRIBUTION
J. JEFFERY, General Manager, North Surrey Water Company
PAINT AND SURFACE COATING: Theory and Practice
Vol. 1: Colloid; Surface and Polymer Chemistry of Modern Surface Coatings
Vol. 2: Physical Properties of Liquid Coating Compositions and of Derived Films
Editor: R. LAMBOURNE, lately Section Manager, Research Department, ICI plc Paints Division
CROP PROTECTION CHEMICALS
B. G. LEVER, Development Manager, ICI plc Plant Protection Division
COMMERCIAL ASPECTS OF OILS AND FATS
R. J. F. LEYSEN, Market Manager, American Soybean Association, Belgium
HANDBOOK OF MATERIALS HANDLING
Translated by R. G. T. LINDKVIST. MTG, Translation Editor: R. ROBINSON, Editor, Materials Handling News. Technical Editor: G. LUNDESJO, Rolatruc Limited
FERTILIZER TECHNOLOGY
G. C. LOWRISON, MSc, FIChemE, MBIM, lately of Fison's Fertilisers Ltd.
NON-WOVEN BONDED FABRICS
Editor: J. LUNENSCHLOSS, Institute of Textile Technology of the Rhenish-Westphalian Technical University, Aachen and W. ALBRECHT, Wuppertal
MICROCOMPUTERS IN THE PROCESS INDUSTRY
E. R. ROBINSON, Head of Chemical Engineering, North East London Polytechnic
POLYMERS IN PACKING TECHNOLOGY
J. STEPEK, Czechoslovakia. Translation Editor: G. E. J. REYNOLDS, Surrey
REFRACTORIES TECHNOLOGY
C. STOREY, Consultant, Durham, One time General Manager, Refractories, British Steel Corporation
PETROLEUM TECHNOLOGY
A. H. SWEATMAN, Visiting Professor, Imperial College and G. A. HOGG, Natural Gas Consultant, lately of British Petroleum Company plc
PERFUMERY TECHNOLOGY 2nd Edition
F. V. WELLS, Consultant Perfumer and former Editor of Soap, Perfumery and Cosmetics, and M. BILLOT, former Cheif Perfumer to Houbigant-Cheramy, Paris, President d'Honneur de la Societe Technique des Parfumeurs de la France
DETERGENTS, SOAPS AND GLYCERINE
E. WOOLLAT, Consultant, lately of Unilever plc

Table of Contents

Introduction

The nature of process control and data acquisition systems has changed dramatically over the past 10 years with the development of microprocessor-based hardware. The use of this equipment has required process engineers to become more familiar with equipment which was very much the sole interest of the electical engineer. Most undergraduate courses in process and automatic control are concerned with the mathematical analysis of control systems, with little or no discussion of hardware. This book discusses the general nature of control and data acquisition equipment, with many examples of applications.

The book is divided into four parts covering basic principles, instruments and interfaces, applications and case studies. After the final chapter a list of texts and articles is given to allow the reader to take the subject further. This list is presented by topic rather than by chapter, although the sequence closely follows the order in which material is covered in the text.

Part I covers the basic areas of computer structures, necessary electronics, computer software and mathematical algorithms, with one chapter devoted to each. Part II covers instruments and interfaces and devotes one chapter to each type of instrument, classified according to the nature of the output signal. The interfaces required for each type of output are covered in some detail.

Parts III and IV are devoted to process applications including both monitoring and closed loop control. Part III deals with application in a general way whilst Part IV deals in detail with four case studies. Both the applications and the case studies reflect the author's interest in fluid systems, particularly those associated with the petroleum industry.

It is hoped that both process and electrical engineers will find the book of value, although each group will probably make greater use of different sections The book does not attempt to provide theoretical studies in control although it is anticipated that by an understanding of the great possibilities which computers offer, wider use may be made of advances in control theory

Part I

FUNDAMENTALS

The objective of this first part is to cover the fundamental aspects of real time computing These are seen as the computer structure, the electronic devices used to create system interfaces, the languages in which programs are written for real time applications and the type of mathematical operations to be carried out

A single chapter is devoted to each of these subjects

1

Computer structures

THE STRUCTURE OF COMPUTERS

A computer may be defined as a general-purpose electronic calculating machine. The basic structure of all computers is shown in Fig. 1.1. The heart of the computer is its central processing unit (CPU) in which all arithmetic and control processes are carried out. Associated with the central processing unit is an area of intermediate storage in which numbers in calculations may be deposited, and which is also used for program storage. Storage may be regarded as a large number of pigeon holes, each of which may be addressed by a location number, allowing the contents of any pigeon hole to be identified and used.

Arrangements must be made for information to enter the computer and for the computer to transmit information out. For on-line uses, instruments such as temperature transmitters and flowmeters will provide the input devices, whilst control valves or electrical motor drives may be the output devices. For all systems, whether on-line or off-line some form of device for user input must be provided. This may be provided either by a typewriter type keyboard or a simple numerical keypad or in larger installations by magnetic tape readers.

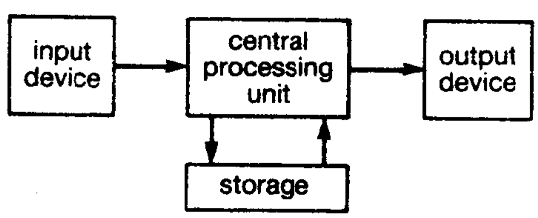

Fig. 1.1 – Computer structure.

Output devices will also include printers and visual display units, which will enable the user or operator to receive information from the computer.

Irrespective of the size of a computer, Fig. 1.1 will be applicable. A large number of input devices and a large number of output devices may be involved, but the basic machine configuration will conform to this arrangement. Nowadays the central processing unit is invariably a microprocessor and it is in this area that concentration will be given in later discussions.

BINARY ARITHMETIC

Computers are based upon the use of binary arithmetic using numbers to the base 2, rather than to the base 10, as we conventionally use. The use of base 2 numbers means that only 2 digits are required: 0 and 1. These can be represented electrically as particular voltage levels and it is the ability to switch electrical voltages in very short times that has made electronic computers possible. Voltages may be switched in times of nanosecond (10^{-9} second) duration, meaning that many million voltage switches may be carried out in a second. During recent years the switching operation has been advanced in speed terms, but even greater advance has taken place in the number of switches which may be compressed into a minute volume of material. Very large-scale integration (VLSI) allows an entire computer central processor with millions of switches to be made on a piece of silicon less than 0.1 cubic centimetre in size. This is the basis of today's so-called 'microcomputer revolution'.

In order to understand the operation of the computer it is necessary to understand that binary numbers can be used to carry out all the operations that decimal arithmetic permits.

Table 1.1 gives a comparison between decimal numbers and binary numbers. The essential difference is that although the columns within decimal numbers represent units (10^0), tens (10^1), hundreds (10^2) etc., binary columns represent units (2^0), twos (2^1), fours (2^2), eights (2^3), sixteens (2^4), etc. The operations of addition and subtraction may be carried out using similar rules to those for decimal operations. Figure 1.2 shows a binary addition process and a binary subtraction process. Following the addition sequence, the rules for binary addition are:

$$0 + 0 = 0 \text{ with } 0 \text{ carry}$$
$$0 + 1 = 1 \text{ with } 0 \text{ carry}$$
$$1 + 1 = 0 \text{ with } 1 \text{ carry}$$
$$\text{carry } 1 + 1 + 1 = 1 \text{ with } 1 \text{ carry}$$

For subtraction the rules are:

$$0 - 0 = 0 \text{ with } 0 \text{ carry}$$
$$1 - 0 = 1 \text{ with } 0 \text{ carry}$$
$$0 - 1 = 1 \text{ with } 1 \text{ carry}$$
$$1 - 1 = 0 \text{ with } 0 \text{ carry}$$

These rules have been applied in each of the applications of Fig. 1.2.

A binary digit is referred to as a bit, and 8 bits are termed a byte. The two numbers used for addition and subtraction are both 8-bit or 1-byte numbers

Table 1.1
Binary/decimal number system

Decimal	Binary
$10^1\,10^0$	$2^3\,2^2\,2^1\,2^0$
0 0	0 0 0 0
0 1	0 0 0 1
0 2	0 0 1 0
0 3	0 0 1 1
0 4	0 1 0 0
0 5	0 1 0 1
0 6	0 1 1 0
0 7	0 1 1 1
0 8	1 0 0 0
0 9	1 0 0 1
1 0	1 0 1 0

```
         10110101
         01011101 +
         ---------
answer  100010010
         ---------
carry     1111101

         10110101
         01011101 -
         ---------
answer   01011000
         ---------
carry     1011000
```

Fig. 1.2 – Addition/subtraction operations.

THE CENTRAL PROCESSING UNIT

The nature of the central processing unit will vary from computer to computer As an illustration of the working principle of the central processing unit a hypothetical machine will be considered. This machine will be assumed to be a 4-bit machine, which means that 4 binary digits will be operated on simultaneously.

Binary numbers are used, not only for the representation of numbers as data, but also for the program instructions This often presents conceptual difficulties when computers are introduced, however, this difficulty should be overcome as the full operating procedure is explained

The heart of the central processing unit is a register called an accumulator which is used to keep a record of arithmetic operations Most of the programming instructions deal with this accumulator A set of instructions has been devised for the hypothetical computer, which will allow its operation to be followed Table 1.2 shows this hypothetical instruction set together with the binary number which is used to represent the operation This binary number is termed the machine code for the instruction

Table 1.2
Hypothetical computer instruction set

Machine code	Meaning
0000	Stop
0001	Load the accumulator with a number
0010	Load the accumulator with memory contents
0011	Add a number to the accumulator
0100	Add memory contents to the accumulator
0101	Store the accumulator in memory location
0110	Shift the accumulator to the right
0111	Shift the accumulator to the left
1000	Jump if zero
1001	Jump if carry set
1010	Clear the accumulator
1011	Increment the accumulator
1100	Subtract a number from the accumulator
1101	Subtract memory contents from the accumulator
1110	Jump to memory location

Using this table a program in machine code may be written to add the numbers 0101 and 1001 and to store the answer in memory location 1111 The program is shown in Fig. 1.3 but deserves a fuller explanation. The program loads the number 0101 into the accumulator, adds to it the number 1001, stores the result in memory location 1111 and finally stops. In each of the instructions except the last, 'stop', the machine code is made up of two 4-bit numbers. The first four bits is the code for the operation and the second is the number or memory location to which the first part applies. This second part of the code is often called the operand. Most instructions are therefore composed of an operation code and an operand code. To load a number in the accumulator requires the code for loading a number (0001) followed by the number (0101).

Machine code	Meaning
0001 0101	load the number 0101 into the accumulator
0011 1001	add the number 1001 to the accumulator
0101 1111	store the contents of the accumulator in memory location 1111
0000	stop

Fig. 1.3 – Program to add numbers

Similarly to add a number requires this code (0011) followed by the number (1001) Storing the accumulator in memory requires this code (0101) followed by the memory location number (1111). The instruction for stop requires only the code 0000.

In order for the program to operate within the computer it must first be loaded into the memory. In general it can be placed anywhere within the computer memory. In Fig. 1.4 the program is shown beginning at memory location 0010. Also in this figure are shown three other important parts of the

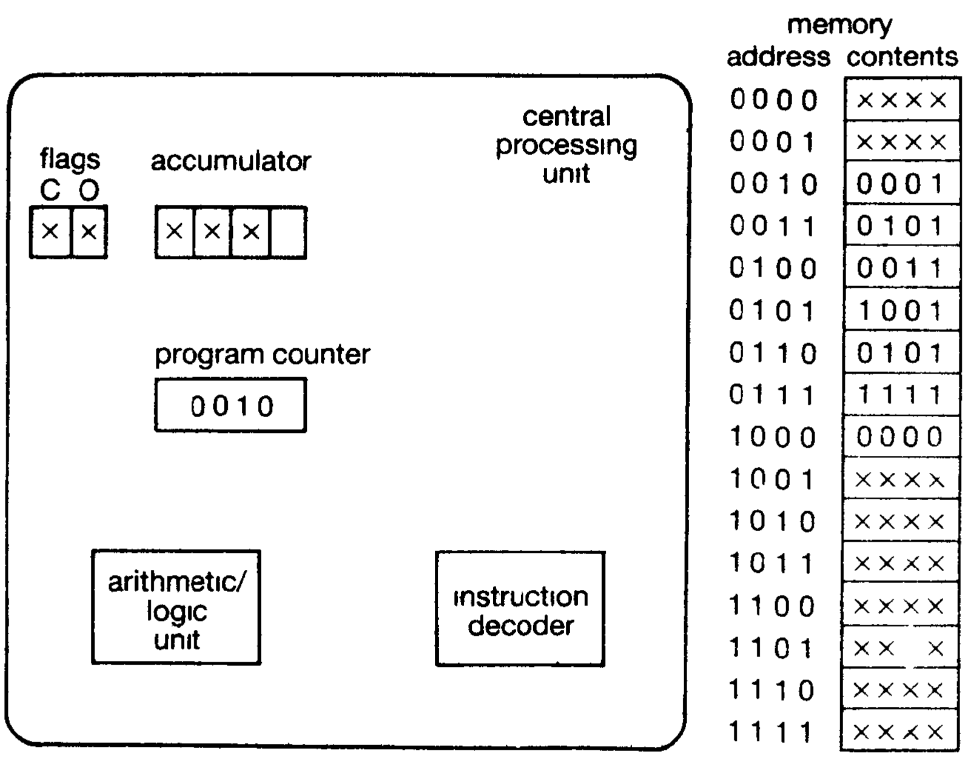

Fig. 1 4 – Program in memory

central processing unit. The program counter is a register which keeps account of the progress of a program by storing the memory address of the next program statement to be serviced. The instruction decoder is electronic circuitry which is able to decode an instruction, to enable it to be executed. The arithmetic/logic unit contains the electronic circuitry to actually carry out the instruction

At the start of program execution the program counter must contain the starting address of the program. The accumulator and the memory locations, except those containing the program, will have numbers left from the previous program or random settings, if the computer has not been used since switching on. These are shown as XXXX and will not affect the program execution.

Once the computer is put into the run mode with the program counter loaded with 0010 the sequence is.

(i) The program counter points at location 0010. The contents of this location are copied to the instruction decoder without altering memory This instruction is decoded as a load memory instruction. The program counter advances to 0011.

(ii) The number 0101 pointed at by the program counter is loaded into the accumulator via the arithmetic/logic unit. The program counter moves on to 0100.

(iii) The instruction at location 0100 is copied to the instruction decoder and decoded as an addition instruction. The program counter advances to 0101.

(iv) The number 1001 at location 0101 is added via the arithmetic/logic unit to the accumulator giving 1110. The program counter increases to 0110

(v) The instruction at memory location 0110 is copied from memory to the instruction decoder and decoded as a store instruction The program counter advances to 0111.

(vi) The memory location for storage is pointed at by the program counter The contents of the accumulator are copied to location 1111 and the program counter is advanced to 1000

(vii) The instruction at location 1000 is copied to the decoder and decoded as a stop instruction The program counter advances to 1001.

(viii) The program execution stops

From this example it may be seen that within the program some of the binary numbers represent operations whilst others represent operands, as either numbers or as memory locations The actual meaning of the binary number is determined only by the position within the program and in the case of operands, by the preceding operation code. Before leaving the hypothetical computer there are further operations which can be carried out which need to be discussed The range of instructions will vary from computer to computer but there will usually be instructions for shifting the accumulator to the left and also to the right A left shift is equivalent to multiplication by 2, whilst a right shift is equivalent to a division by 2. There will also be instructions for subtraction, clearing and incrementing.

A further and most important class of instructions are those which allow branching within the program. Two conditions on which branching may arise with all machines are when a zero is detected or when a carry is detected. Associated with the accumulator of the hypothetical machine will be two flags, one to detect a zero and the other to detect a carry. When the appropriate condition is detected the corresponding flag will be set, which means it will have the value 1. When the condition is not met the flag will have the value 0. These

```
  11
  11 ×
  ----
  11   top
  ----
  11   add to zero
  ----
 110   top shifted left
  ----
1001   add to previous to give answer
  ----
```

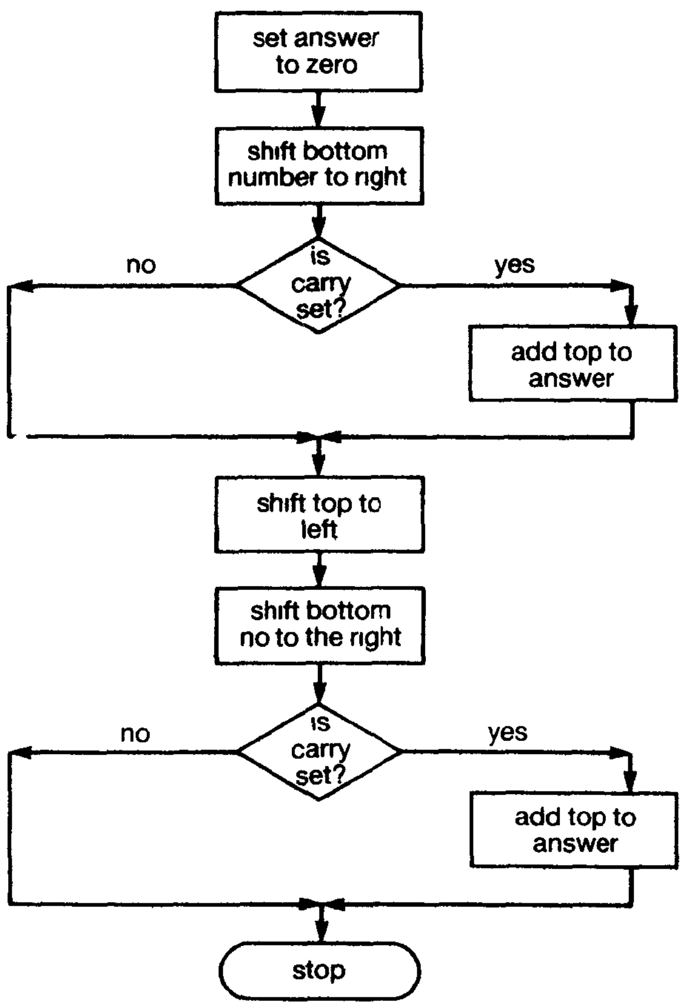

Fig. 1.5 – Multiplication procedure.