DOCUMENT RESUME

ED 168 588 IR 007 148

AUTHOR	Young, Micki Jo; And Others
TITLE	Introduction to Minicomputers in Federal Libraries.
INSTITUTION	Library of Congress, Washington, D.C.
SPONS AGENCY	Federal Library Committee, Washington, D.C.
PUB DATE	78
CONTRACT	A76-234
NOTE	155p.; Not available in hard copy due to small print size of original document
AVAILABLE FROM	Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402 (LC 1.2:M 66/2)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	Computer Programs; *Computers; Computer Storage Devices; *Government Libraries; Library Administration; *Library Automation; Library Role; Man Machine Systems; Systems Approach; *Systems Development
IDENTIFIERS	*Minicomputers

ABSTRACT

This book for library administrators and Federal library staff covers the application of minicomputers in Federal libraries and offers a review of minicomputer technology. A brief overview of automation explains computer technology, hardware, and software. The role of computers in libraries is examined in terms of the history of computers and current evolving technology. An examination of microcomputers as a solution to library problems focuses on hardware and peripherals, including mass storage devices and man/machine interface devices. Systems software is discussed, with emphasis on program development aids, file management programs, operating systems, and applications. Criteria for system selection are identified, and library applications in the areas of acquisitions, cataloging, serials, circulation, interlibrary loan, reference and information services, and administration are explored. Based on these applications and requirements for space and personnel, guidelines are given for system selection and evaluation. The roles of the library, agency, federal government, and private sector in the application of minicomputers are discussed, as well as future trends. A glossary and systems specifications for procurement of minicomputer systems from the Library of Congress are appended. (CWM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED168588

INTRODUCTION TO MINICOMPUTERS IN FEDERAL LIBRARIES

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

BEST COPY AVAILABLE

By
MICKI JO YOUNG
With
FRANK A. PEZZANITE
And
J. CHRIS REISINGER

Prepared under contract A76-234
for the
Federal Library Committee

Library of Congress Washington 1978

IR007148

Library of Congress Cataloging in Publication Data

Young, Micki Jo.
Introduction to minicomputers in Federal libraries.

"Prepared under contract A76-234 for the Federal Library Committee, Library of Congress."
Bibliography: p.
Supt. of Docs. no.: LC 1.2:M 66/2
1. Libraries—United States—Automation.
2. Minicomputers. 3. Libraries, Governmental, administrative, etc.—United States. I. Pezzanite, Frank A., joint author. II. Reisinger, J. Chris, joint author. III. United States. Federal Library Committee. IV. Title.
Z678.9.Y68   027.5'028'54   78-1652

For sale by the Superintendent of Documents, U.S. Government Printing Office
Washington, D.C. 20402
Stock No. 030-000-00097-8

TABLE OF CONTENTS

Page

PREFACE vii

THE AUDIENCE vii
THE SCOPE vii
TIMELINESS viii
THE SPONSORS viii

CHAPTER ONE: OVERVIEW OF AUTOMATION 1

HOW A COMPUTER WORKS 1
  HARDWARE 1
    Central Processing Unit 2
    Memory 6
    Input 8
    Output 8
    The Configuration 9
  SOFTWARE: COMPUTER PROGRAMS 9
    Input/Output Instructions 13
    Internal Data Movement Instructions 13
    Arithmetic Instructions 13
    Testing and Comparison Instructions 13
    Jumping or Branching Instructions 14
    Programming Example 14
      Assembly Language Terms 15
      High-Level Language Terms 15
  OPERATING MODE 16
    Dedicated, Shared, or Time-Shared 16
    Batch, Remote Batch, Online Interactive, or Time-Sharing 16
ROLE OF THE COMPUTER 16

CHAPTER TWO: ROLE OF COMPUTERS IN LIBRARIES 19

DIRECTIONS IN LIBRARY AUTOMATION 19
  HISTORY OF COMPUTERS IN LIBRARIES 20
  CURRENT EVOLVING TECHNOLOGY 22
    Minicomputers 22
    Cooperative Processing 22
    Online Files 23
    Packaged Software 23
    Commercial Services 23
    Computer-Output Microform 24
    Telecommunications 24
  FUTURE APPLICATIONS OF NETWORKS IN LIBRARIES 24
    Communications 25
    Computers 26
    Information 26
  PROBLEMS IN CURRENT LIBRARY APPLICATIONS 27
    Poor Systems Design 27
    Poor Librarian/Computer Personnel Interface 27
    Poor Support 28
    Other Problems 28
  FEDERAL LIBRARY AUTOMATION APPLICATIONS 28

- HOW THE MINICOMPUTER COULD HELP ... 28
  - HISTORY AND DEVELOPMENT ... 29
  - CHARACTERISTICS ... 29
  - DEFINITIONS ... 30
  - BASIC USES AND APPLICATIONS ... 30
    - *Industrial Process Control* ... 30
    - *Peripheral Control* ... 31
    - *Data Acquisitions* ... 31
    - *Communications* ... 31
    - *Computation* ... 32
  - MINICOMPUTERS IN LIBRARIES ... 32
- SUMMARY ... 33

CHAPTER THREE: MINICOMPUTERS—DESCRIPTIONS ... 35

- HARDWARE ... 36
  - MAINFRAME ... 36
    - *CPU* ... 36
      - Instruction Set ... 36
      - Addressing Capabilities ... 37
      - Instruction Execution Speed ... 37
      - Register Configuration ... 37
      - Interrupts ... 38
      - Optional Features ... 38
    - *Main Memory* ... 38
    - *Input/Output Control* ... 39
  - PERIPHERALS ... 39
    - *Mass Storage Devices* ... 39
      - Punch Cards ... 40
      - Paper Tape ... 40
      - Cassette or Cartridge Tape ... 40
      - Industry-Standard Magnetic Tape ... 40
      - Diskette, or Floppy Disk ... 41
      - Fixed-head Disk ... 41
      - Movable-head Disk ... 41
    - *Man/Machine Interface Devices* ... 41
      - Punch Cards ... 42
      - Paper Tape ... 42
      - Key-to-Magnetic-Medium Stations ... 42
      - Transaction Recorders ... 42
      - Printers ... 43
      - Interactive Man/Machine Interface Devices ... 43
      - Input/Output Connections ... 45
- SOFTWARE ... 45
  - SYSTEMS SOFTWARE ... 46
    - *Program Development Aids* ... 46
      - Assemblers ... 46
      - Loaders ... 46
      - Editors ... 46
      - Debuggers ... 47
      - Diagnostics ... 47
      - Utilities ... 47
      - Subroutine Libraries ... 47
      - Compilers ... 47
    - *File Management Programs* ... 48
    - *Operating Systems* ... 48

APPLICATIONS SOFTWARE 50
CONFIGURATIONS 50
CLASSES OF MINICOMPUTERS 50

CHAPTER FOUR: MINICOMPUTERS—LIBRARY APPLICATIONS 53

SYSTEM SELECTION: THE BRIDGE 53
COLUMN ONE: DESIGN CHARACTERISTICS 53
Means of Inputting 54
Types of Output Products 55
File Structure and Size 55
Transaction/Volume 56
Applications Characteristics 56
Interface with Other Systems 57
COLUMNS TWO AND THREE: HARDWARE AND SOFTWARE IMPACTS 57
COLUMN FIVE: CLASS OF MINICOMPUTER 65
COLUMN FOUR: LIBRARY SYSTEM SPECIFICATIONS 65
System Requirements—One End of the Bridge 65
Systems Design—The Other End of the Bridge 67
LIBRARY APPLICATIONS 69
GENERAL 69
ACQUISITIONS 70
Case Study 82
Problem Area 82
Library Environment 82
Goals 82
Proposed Flow 82
System Requirements 82
Alternative Designs 82
The Bridge: Review of Design Characteristics and Their Impacts 82
The Analysis 86
CATALOGING 86
SERIALS 88
CIRCULATION 90
INTERLIBRARY LOAN 91
REFERENCE AND INFORMATION SERVICES 92
ADMINISTRATION 94
SUMMARY 95

CHAPTER FIVE: SELECTION CRITERIA 97

FACTORS INFLUENCING THE DECISION 97
COMPARISON TO A MANUAL SYSTEM 97
COMPARISON TO A LARGE-SCALE SYSTEM 98
SERVICE CONSIDERATIONS 98
COST/BENEFIT ANALYSES 98
SELECTING THE SYSTEM 99
REVIEW OF HARDWARE/SOFTWARE 100
CPU 101
Peripherals 102
Mass Storage Devices 103
Man/Machine Interface Devices 105
Software 107
Systems 107
EVALUATION 108
Turnkey Systems 110
REVIEW OF OTHER RESOURCES/REQUIREMENTS 112

Physical Requirements 112
Space 112
Power 112
Noise 112
Maintenance 112
Personnel 113
Time Frame 113
THE DECISION 113
Federal ADP Procurement 113
CHAPTER SIX: CONCLUSIONS 117
INITIATING MINICOMPUTER PROJECTS IN FEDERAL LIBRARIES 117
THE LIBRARY'S ROLE 117
THE AGENCY'S ROLE 117
FEDERAL GOVERNMENT ROLE 118
ROLE OF THE PRIVATE SECTOR 118
TRENDS IN FUTURE APPLICATIONS OF MINICOMPUTERS IN FEDERAL LIBRARIES 118
IN CLOSING 119
APPENDIX A. GLOSSARY 121
APPENDIX B. LIBRARY OF CONGRESS SYSTEMS SPECIFICATIONS FOR PROCUREMENT OF MINICOMPUTER SYSTEMS 133
BIBLIOGRAPHY 151

PREFACE

THE AUDIENCE

This book is directed to Federal libraries and the librarians who staff them. The main audience will be library administrators; middle-level managers and section heads also will benefit from this material. Parts of the book also may be of value for staff training late in the development of minicomputer projects.

The library director, as the decisionmaker, should be familiar with minicomputers and their applications in libraries. After deciding to use minicomputers, the director must either direct the project personally or assign a project director.

The qualifications required for successful project direction are not rigid. However, the following characteristics are very important:

- Ability to communicate well
- Sound knowledge of library operations
- Familiarity with at least basic fundamentals of data processing technology
- Ability to write clearly
- Sound constitution and steady nerves.[1]

The project will probably involve a systems team that includes data processing experts experienced with hardware, systems analysis, and programming. The team may draw from agency personnel or from outside contractors. Whichever procedure is used, the librarian must be a working member of the team and must control the major decisions. It is the intent of this book to prepare librarians for this role.

The entire staff should be kept informed of the progress of the automation effort because their high morale and support of the system are vital to the success of the project. Lack of knowledge often prompts fear of the unknown. In addition to constant communication, the staff should receive training, at least in the form of general orientation to library automation and minicomputers.

In fact the library staff should play an integral part as contributors to the development of the system. The library director alone cannot give the necessary detailed input to the system designers. It has been pointed out that often only the staff are "in a position to recognize a serious flaw in an operating procedure. Unless they have been given the opportunity to see what the purposes are, they can hardly begin to bring out what would otherwise be only intuitive feelings that 'something's not right'."[2] To participate fully and meaningfully, the staff may have to be trained or at least briefed on systems analysis, library automation, and minicomputers.

THE SCOPE

This book covers the application of minicomputers in Federal libraries. The first four chapters, Part I, form a general textbook treatment of automation, library automation, minicomputers, and library applications of minicomputers. In Part II, Chapters 5 and 6, guidelines for selecting and implementing a minicomputer system are presented.

The chapters of Part I provide the necessary background for reviewing library needs and making decisions in applying minicomputer technology. If the reader is already familiar with automation and minicomputers, he or she may begin at Part II. In-depth explanations of computer technology are not possible in this book, and are, indeed, unnecessary for most of the librarians who will read this book.

An understanding of systems analysis is important to the readers of this book but is not covered in depth here. The library director making the decisions on applying minicomputers should have completed the initial stages of systems analysis before applying the guidelines in this book. The director may follow *Guidelines for Library Automation*,[3] sponsored by the Federal Library Committee, for a thorough treatment of library systems analysis directed specifically to Federal libraries.

---

[1] Barbara Evans Markuson et al., *Guidelines for Library Automation: A Handbook for Fedeial and Other Libraries* (Santa Monica, Calif.: System Development Corporation, 1972), p. 24.

[2] Robert M. Hayes and Joseph Becker, *Handbook of Data Processing for Libraries*, 2d ed. (Los Angeles: Melville Publishing Co., 1974), p. 141.

[3] Markuson et al., *Guidelines for Library Automation.*

TIMELINESS

The time for minicomputers in libraries has come. The *Annual Review of Information Science and Technology*[4] had a chapter on minicomputers for the first time in 1975. The September 1976 issue of *Library Technology Reports*[5] featured a description of three commercial library systems that use minicomputers. There is a need for this book, though some sections may be out of date before it is published. This is true of any work that deals with specific makes and models of equipment. In this case, the problem is compounded by the state of minicomputer technology. This infant area is just beginning to mushroom and grow. Changes in basic methodologies or breakthroughs in techniques can occur at any moment and drastically change capabilities and applications. It is all for the better, but one is warned to stay on top of this dynamic field.

THE SPONSORS

The Federal Library Committee (FLC) has recognized the trend to minicomputer applications in libraries and their value to Federal libraries. In response FLC has organized a working group on minicomputers to serve as a forum for Federal librarians to share knowledge and experience; FLC also contracted with Informatics Inc. to write this book. The Library Information Services section of Informatics Inc.'s Information Services Group is singularly qualified to perform this task. Under the administration of Jack A. Speer, Director, Library Information Services has supplied library technical services for a number of Federal libraries, including acquisitions services, feasibility and design studies, and production of bibliographic products. It has also developed a commercial minicomputer system to support cataloging.

Dr. Micki Jo Young was selected by Mr. Speer to write the book. The background of Dr. Young includes expertise in library technical services management; large-scale computer systems; minicomputers as a cataloging utility tool; and systems design. Frank A. Pezzanite and J. Chris Reisinger of Informatics' senior staff provided technical expertise for this book. Mr. Pezzanite is the Technical Director of Library Information Services and was in charge of project development for MINI-MARC, the firm's commercial minicomputer module for cataloging. Mr. Reisinger is Technical Director for Electronic Composition Services in the Information Processing Services Division. He is responsible for computer-based text-processing, using both large-scale computer systems and minicomputer systems. Together, these three persons bring a well-rounded professional view of minicomputers in Federal library applications.

June 1977

---

[4] Karl M. Pearson, Jr., "Minicomputers in the Library," in *Annual Review of Information Science and Technology*, vol. 10, eds. Carlos A. Cuadra and Ann W. Luke (Washington, D.C.: American Society for Information Science, 1975), pp. 139-63.

[5] "Questions and Answers Department," *Library Technology Reports* 13 (September 1976): 465-9.

CHAPTER ONE

OVERVIEW OF AUTOMATION

HOW A COMPUTER WORKS

In general there are three possible modes for library operations: manual, mechanized, and automated. At one time all catalog cards were written by hand, and penmanship called library hand was taught in library school. The typewriter, and then multilith, offset, and photocopy processes mechanized card production. The next step, automation, involves use of a computer.

Ned Chapin defined the computer as follows: "An automatic computer is a machine that manipulates symbols in accordance with given rules, in a predetermined and self-directed manner."[1] The key word is "self-directed." This is what distinguishes automation from mechanization. Sometimes this concept is called the "stored program." Cox, Dews, and Dolby wrote, "A librarian interested in the potential usefulness of a computer must first have an idea of what the machinery and associated programs can do."[2] It is for this purpose that the following general description of computers, their functions and roles, is presented.[3]

There are two main types of computers: analog and digital.[4] An analog computer represents variables by analogies. That is, it represents numerical quantities by means of physical variables such as translation, rotation, voltage, or resistance. Analog computers are used mainly in industry and for scientific applications requiring simulation of nature or natural properties. A digital computer expresses variables in discrete, countable form usually by means of coded characters such as numbers, signs, or symbols. Digital computers are best suited to arithmetic and logical operations in business and engineering applications. Because of the nature of library operations, digital computers are used exclusively for library automation and the remaining discussion will deal with digital computers only.

To understand computers, one must consider both the equipment (hardware) and the instructions or programs that make them self-directing (software).

HARDWARE

Five separate functions can be performed by computer hardware:[5]

- *Conversion* from one form of representation to another
- *Storage* for varying periods of time
- *Communication* by movement of data
- *Logical* and arithmetic *processing*
- *Display* in human-sensible form.

No one machine performs all five functions. Rather, a number of devices are combined. The group of components required to perform the five functions consists of:[6]

- An *input* unit
- The *memory*
- The *control* unit
- The *arithmetic* unit
- An *output* unit.

Each of these components is not necessarily a separate piece of equipment. Some components have multiple uses (e.g., both input and output) combining more than one unit in one device. Also, under any one category there is a variety of devices from which to choose. For example, there are several types of input units: punched card readers, paper tape readers, optical character recognition devices, and key-to-disk stations, among others.

---

[1] Ned Chapin, *An Introduction to Automatic Computers* (New York: Van Nostrand, 1957) cited by Robert M. Hayes and Joseph Becker, *Handbook of Data Processing for Libraries*, 2d ed. (Los Angeles: Melville Publishing Co., 1974), p. 237.

[2] N. S. M. Cox, J. D. Dews, and J. L. Dolby, *The Computer and the Library; The Role of the Computer in the Organization and Handling of Information in Libraries* (Newcastle upon Tyne, Eng.: University of Newcastle upon Tyne Library, 1966), p. 27.

[3] A number of general textbooks on automation are available for further study; for example, Marilyn Bohl, *Information Processing*, 2d ed. (Chicago: Science Research Associates, 1976) and Elias M. Awad and Data Processing Management Association, *Automatic Data Processing; Principles and Procedures*, 3d ed. (Englewood Cliffs, N.J.: Prentice-Hall, 1973). Hayes and Becker's *Handbook of Data Processing for Libraries* is a comprehensive treatment of library automation while *Computer & Systems; An Introduction for Librarians* by John Eyre and Peter Tonks (London: Clive Bingley, 1971) is a much simpler description.

[4] Edward M. Heiliger and Paul B. Henderson, Jr., *Library Automation: Experience, Methodology, and Technology of the Library as an Information System* (New York: McGraw-Hill Book Co., 1971), pp. 168, 173.

[5] Ibid., p. 191.

[6] William R. Corliss, *Computers*, Rev., Understanding the Atom Series (Oak Ridge, Tenn.: U.S., Atomic Energy Commission, Division of Technical Information, 1967), pp. 13–14.

Central Processing Unit

The central processing unit (CPU) is the heart of the computer. It combines two of the five main functions; it is composed of the arithmetic and the control units and is closely tied to the main memory.

In a digital computer the main units of computation are discrete, countable codes, most commonly numbers. Even more basic than number codes is the simple on-off distinction. This is the principle on which digital computer operation is based. The on and off states are represented numerically by 0 and 1.

0 (zero) = "off"
1 (one) = "on".

In digital computers arithmetic computations and numerical expressions are based on this binary (base-two) number system rather than the decimal system because binary is easier to manipulate by computer. Figure 1 shows a simple comparison of the binary and decimal number systems. Further references are recommended for additional study of binary mathematics.[7] At this point the thing to remember is that the 0–1/off-on condition is the basis of all computer operations. The instructions or commands of the program that allow the computer to be "self-directed" are represented by 0–1 codes, as are the data to be processed or manipulated.

There are several standard 0–1 codes in which separate arithmetic values are used to represent specific letters, numbers, or symbols. Some codes have been standardized throughout the computer industry and can be used on any manufacturer's equipment: BCD, EBCDIC, and ASCII are the three most common. ASCII (American Standard Code for Information Interchange) is used throughout the Federal Government as the standard and is the code for the Library of Congress MARC II distribution tapes.

Table 1 gives the seven-digit (seven-bit) binary pattern for each letter, number and symbol in the ASCII code. The order of the codes represents the binary (arithmetic) values in ascending numerical order; if data expressed in ASCII code is sorted in numerical order, it will result in this sequence. Programmers often review and manipulate data in binary form as programs are developed and corrected. Representing each character with seven digits would be cumbersome and slow. To provide a shorthand method of handling these binary patterns, hexadecimal notation is sometimes used. Hexadecimal is a base-16 system in which there are 15 decimal units before the 10 (no units, one 16) is reached. Because there aren't enough numerals to express all the units, the letters A through F are used:[8]

TABLE 1—Comparison of Decimal, Hexadecimal, and Binary Systems

Decimal System	Hexadecimal System	Binary System 8 4 2 1 bit value
0	0	0 0 0 0
1	1	0 0 0 1
2	2	0 0 1 0
3	3	0 0 1 1
4	4	0 1 0 0
5	5	0 1 0 1
6	6	0 1 1 0
7	7	0 1 1 1
8	8	1 0 0 0
9	9	1 0 0 1
10	A	1 0 1 0
11	B	1 0 1 1
12	C	1 1 0 0
13	D	1 1 0 1
14	E	1 1 1 0
15	F	1 1 1 1

When hexadecimal notation is used for ASCII code, every four bits are represented by a hexadecimal character. Since ASCII is a seven-bit code, the eighth bit (reading from right to left) is assumed always to be zero:

ASCII G = _1000111 = Hex 47
Hex 0111 = 7
Hex 0100 = 4

ASCII k = _1101011 = Hex 6B
6 B

This allows eight characters to be reduced to two characters for efficiency, but it must be remembered that the eight-bit patterns are what reside in main

---

[7] Most textbooks on computers and programming describe binary mathematics. Two particularly detailed treatments are found in "Appendix A, Primer of Number Systems," Richard H. Eckhouse, Jr. *Minicomputer Systems: Organization and Programming (PDP-11)* (Englewood Cliffs, N.J.: Prentice-Hall, 1975), pp. 277–99 and "Chapter 5, Data Representation" and "Chapter 6, Computer Arithmetic," Awad and Data Processing Management Association, *Automatic Data Processing*, pp. 96–130.

[8] International Business Machines, *Student Text Introduction to IBM Data Processing Systems*, 2d ed. (White Plains, N.Y.: International Business Machines Corp., Technical Publications Department, 1968), p. 22.

(A)

To Count:	DECIMAL SYSTEM base ten 0 through 9					BINARY SYSTEM* base two 0 and 1					
	ten thousands	thousands	hundreds	tens	units	thirty-seconds	sixteens	eights	fours	twos	units
●					1						1
●●					2					1	0
●●●					3					1	1
●●●●●● ●●●●●●				1	2			1	1	0	0
●●●●●● ●●●●●● ●●●●●● ●●●				2	1		1	0	1	0	1
●●●●●● ●●●●●● ●●●●●● ●●●●●● ●●●●●●				3	0		1	1	1	1	0

(B)

DECIMAL NUMBER	BINARY NUMBER	COMPUTER IMPULSE PATTERN (off=o, on=x)
0	0000	oooo
1	0001	ooox
2	0010	ooxo
3	0011	ooxx
4	0100	oxoo
5	0101	oxox
6	0110	oxxo
7	0111	oxxx
8	1000	xooo
9	1001	xoox
10	1010	xoxo
11	1011	xoxx
12	1100	xxoo
13	1101	xxox
14	1110	xxxo
15	1111	xxxx

(C)

ADDITION FOLLOWS THE SAME RULES AS DECIMAL ADDING, INCLUDING CARRYING

1	00001	2	00010	13	01101
+ 2	00010	+ 3	00011	+ 11	01011
3	00011	5	00101	24	11000

FIGURE 1. Comparison of binary and decimal number systems

TABLE 2—Hexadecimal Notation for ASCII Code

Character	ASCII	Hexadecimal	Character	ASCII	Hexadecimal	Character	ASCII	Hexadecimal
!	0100001	21		1000000	40		1100000	60
"	0100010	22	A	1000001	41	a	1100001	61
#	0100011	23	B	1000010	42	b	1100010	62
$	0100100	24	C	1000011	43	c	1100011	63
%	0100101	25	D	1000100	44	d	1100100	64
&	0100100	26	E	1000101	45	e	1100101	65
'	0100111	27	F	1000110	46	f	1100110	66
(	0101000	28	G	1000111	47	g	1100111	67
)	0101001	29	H	1001000	48	h	1101000	68
*	0101010	2A	I	1001001	49	i	1101001	69
+	0101011	2B	J	1001010	4A	j	1101010	6A
,	0101100	2C	K	1001011	4B	k	1101011	6B
-	0101101	2D	L	1001100	4C	l	1101100	6C
.	0101110	2E	M	1001101	4D	m	1101101	6D
/	0101111	2F	N	1001110	4E	n	1101110	6E
0	0110000	30	O	1001111	4F	o	1101111	6F
1	0110001	31	P	1010000	50	p	1110000	70
2	0110010	32	Q	1010001	51	q	1110001	71
3	0110011	33	R	1010010	52	r	1110010	72
4	0110100	34	S	1010011	53	s	1110011	73
5	0110101	35	T	1010100	54	t	1110100	74
6	0110110	36	U	1010101	55	u	1110101	75
7	0110111	37	V	1010110	56	v	1110110	76
8	0111000	38	W	1010111	57	w	1110111	77
9	0111001	39	X	1011000	58	x	1111000	78
:	0111010	3A	Y	1011001	59	y	1111001	79
;	0111011	3B	Z	1011010	5A	z	1111010	7A
<	0111100	3C	[	1011011	5B	{	1111011	7B
=	0111101	3D	\	1011100	5C	|	1111100	7C
>	0111110	3E	]	1011101	5D	}	1111101	7D
?	0111111	3F		1011110	5E		1111110	7E
				1011111	5F			

memory and are actually manipulated. Table 2 gives the hexadecimal notation for each ASCII code.[9]

The arithmetic unit is where the manipulation of the data occurs.[10] Whether the original data were figures or words, the binary codes are manipulated by arithmetic or pseudo-arithmetic operations. The on-off combinations of the electronic impulses are called circuit elements and represent combinations of the basic logical operations AND, OR, and NOT.[11] These circuit elements allow binary numbers to be added, subtracted, multiplied, divided, and compared. Figure 2 shows several circuit elements, and the logical operations they represent.[12]

The instructions or program give detailed commands that control the pattern and sequence of the circuit elements. These must be explicit to the last detail; the computer is not an electronic brain but rather an electronic slave.[13] The impact of this detail will be discussed in the section on software.

Four main classes of computer instructions are carried out by the CPU:[14]

1. *Arithmetic and logical instructions.* These instructions add numbers together, compare patterns of letters or numbers, reorganize the coding of characters, etc. [Sometimes this class is divided into two classes, one strictly for arithmetic, and the other for logical operations.]
2. *Fetch and store instructions.* These instructions are used for moving information about in the memory or store.

---

[9] International Business Machines, *IBM System/360 Operating System Basic Telecommunications Access Method*, 3d ed. (IBM Systems Reference Library) (White Plains, N.Y.: International Business Machines Corp., Data Processing Division, 1968), p. 177.

[10] Hayes and Becker, *Handbook of Data Processing for Libraries*, p. 249.

[11] A more detailed description of the basic logical operations is found in "Appendix B: Primer of Logic Operations," of Eckhouse's *Minicomputer Systems*, pp. 300-305.

[12] Figure 2 is based on Figure 10.7 in Hayes and Becker's *Handbook of Data Processing for Libraries*, pp. 252-3 and an illustration in Corliss, *Computers*, p. 28.

[13] Cox, Dews, and Dolby, *Computers and the Library*, p. 30.

[14] Ibid.

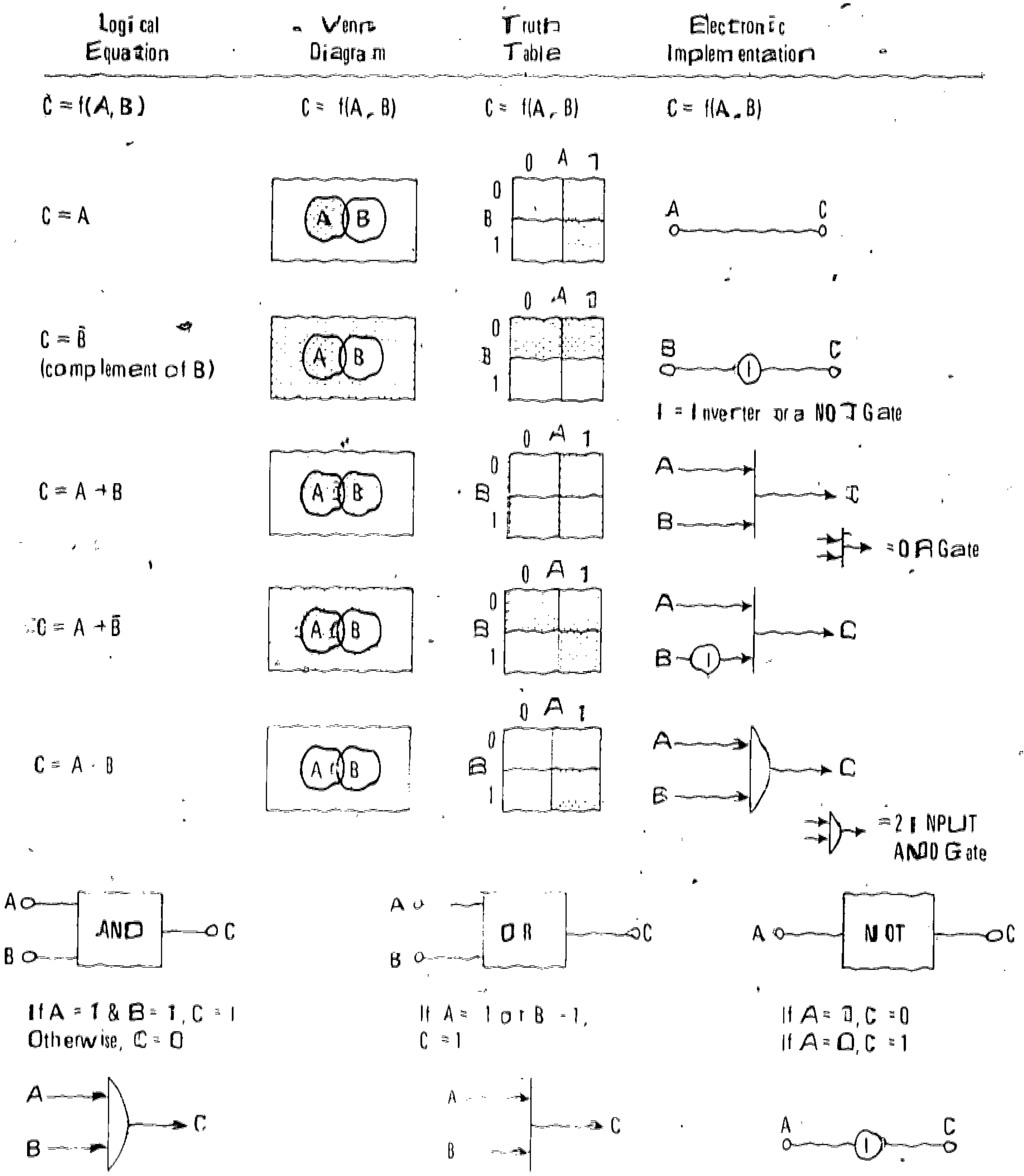

Logical Equation
Venn Diagram
Truth Table
Electronic Implementation
C = f(A, B)
C = f(A, B)
C = f(A, B)
C = f(A, B)
C = A
C = B̄
(complement of B)
I = Inverter or a NOT Gate
C = A + B
= OR Gate
C = A + B̄
C = A · B
= 2 INPUT AND Gate
0 A 1
0 B 1
A
B
C
I
AND
OR
NOT
If A = 1 & B = 1, C = 1
Otherwise, C = 0
If A = 1 or B = 1,
C = 1
If A = 1, C = 0
If A = 0, C = 1

FIGURE 2. Logical relationships expressed as an equation, Venn diagram, truth table and circuit element

3. *Jump (or branch) instructions*. These instructions allow the sequence of operations to vary according to the data being examined.
4. *Input and output instructions*. These instructions are used to fetch blocks of information into the memory from input units and send information from the memory to output devices.

The actual data or codes manipulated by the arithmetic and logic unit are generally held in that unit in a register. A register is a semiconductor device of electromechanical flip-flop switches that holds the data.[15] The register is where the actual operations on the data are performed, such as analyzing, shifting, and performing arithmetic on the numbers in the register.[16] Usually there are multiple registers, and these can be called accumulators. A register has a fixed size and is referred to by name, not by its location in the memory.

The control unit in the CPU handles the switching and flow of the instructions and data in the CPU and in the computer as a whole. It sets the sequence of operations, decodes instructions, and provides the control signals to coordinate the various units of the computer.[17] At a simplistic level the data represented as a pattern of on-off electronic impulses travel through a series of circuits or gates that alters the flow and thus the patterns of the impulses. The circuits or gates are chosen according to the operation to be performed.

The control unit regulates the flow of impulses and the setting of the gates in response to the instructions in the proper order. There are two main registers in the control unit: the instruction register and the location counter. The control operates in cycles. The basic cycles are (1) the instruction cycle (I-time), during which the instruction register receives the next instruction in the program from the main memory, the instruction is interpreted into circuits, and the location counter receives the address of the next instruction, and (2) the execution cycle (E-time) during which the instruction is performed.[18]

Memory

The memory stores information received through an input unit or developed during the processing of data, so that it can be brought from storage for use without being destroyed.[19] There are two kinds of memory or storage. One is main memory, also called working storage or internal storage, and the other is mass storage or auxiliary memory.

The main memory is closely related to the arithmetic and control units. It is sometimes considered part of the central processing unit or mainframe (see Figure 8). The main memory holds the data manipulated by the arithmetic unit. Because of the speed of processing, the data must be readily available, so they are moved from mass storage into main memory. The data and instructions are given unique addresses, or locations. The instructions include the address of the data to be manipulated as well as the address of the next instruction. The computer, thus, is self-directing. In a few processors, arithmetic and logic operations are performed directly on the data in the main memory. In general, however, data are moved from the main memory to a register in either the arithmetic and logic unit (for data) or the control unit (for instructions).

The main memory in most current computers is made up of magnetic cores (small rings of a ferromagnetic material) wired together in an array.[20] Each core can hold one binary digit — either zero or one — by its magnetic state. (The internal memory also is called the core.)

The main memory is described in two ways: in terms of its addressability and its size. At the lowest level a single binary code (either one or zero) is called a *bit*. To express logical characters such as a number or a letter, several bits are grouped in a unit called a *byte*. A byte has a fixed length depending on the hardware design; six-bit and eight-bit bytes are common. The smallest addressable unit in internal memory is called a *word*. Words can be made up of from one to eight bytes, depending on the manufacturer's design.

The capacity of the main memory is expressed in terms of the number of addressable units it can hold, either words or bytes. Capacities are expressed in thousands (K); common capacities are 4K, 8K, 16K, and 32K. With the unit addressability added, the memory is expressed as 4K words, 16K bytes, or 32K bytes (which can be the same as 8K words if the words are 4 bytes or 32 bits each).

Auxiliary or mass storage is used to retain large amounts of data after they are converted into machine-readable form. The storage medium can vary. The storage may be in mechanical, electronic, or magnetic form. The standard punch card is a common mechanical storage medium. Electronic storage is now generally limited to storage during transportation, e.g., telecommunications. Magnetic storage is most common and uses tapes, disks, drums, or strips.

---

[15] William Barden, Jr., *How to Buy & Use Minicomputers & Microcomputers* (Indianapolis: Howard W. Sams & Co., 1976), p. 215.

[16] Decima M. Anderson, *Computer Programming FORTRAN IV* (New York: Appleton-Century-Crofts, 1966), pp. 26-27.

[17] *Auerbach On Minicomputers* (New York: Petrocelli Books, 1974), p. 16.

[18] Anderson, *Computer Programming FORTRAN IV*, pp. 27-28.

[19] Ibid., p. 16.

[20] Corlis, *Computers*, pp. 22-24.

Mass storage data generally are organized in files. There is a hierarchy of data.[21]

Bit: a single binary digit, either 0 or 1.

Byte: a group of bits (usually eight) that represents a logical character such as a letter, number, or symbol.

Field: characters combined into logical units or data elements. Fields can be divided into subfields if appropriate to the intellectual content of the data. For example,

Field — personal name
Subfield 1 — last name
Subfield 2 — first name and middle initial.

Fields can be defined by the programmer as fixed-length (always n number of characters) or variable-length (as many characters as necessary to express the logical unit).

Record: related fields combined into a complete, logical unit.

File: like records combined for the system.

Data base: a group of files related to a system application.

The order of the records in the file and therefore in the main memory device is called file organization. Available methods of file organization include[22]

1. *Sequential (contiguous allocation).* The records occur in a linear order sequenced usually by a control number (Figure 3a). To update the file by adding or deleting a record, a new sequence would have to be compiled. For example, deleting #2341 and adding #2350 would produce something like Figure 3b.

2. *Linked (linked list allocation).* The records do not occur in any order on the file, but the address of the next logical record is carried in the record to link the two together (Figure 4a). To update the file by adding or deleting a record, one need modify only the address of the next record or re-link the records. For example, deleting #2341 and adding #2350 would produce a file like that shown in Figure 4b.

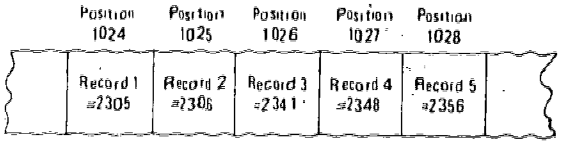

Position 1024
Position 1025
Position 1026
Position 1027
Position 1028
Record 1 #2305
Record 2 #2306
Record 3 #2341
Record 4 #2348
Record 5 #2356

FIGURE 3.a. Records in a sequential file structure

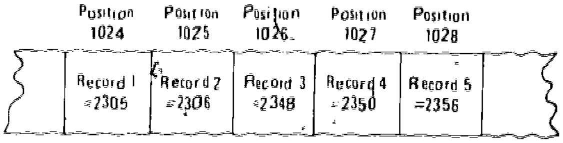

Position 1024
Position 1025
Position 1026
Position 1027
Position 1028
Record 1 #2305
Record 2 #2306
Record 3 #2348
Record 4 #2350
Record 5 #2356

FIGURE 3.b. Updated record sequence

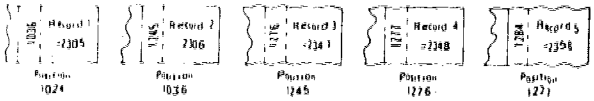

1036
Record 1 #2305
1245
Record 2 2306
1276
Record 3 #2341
1277
Record 4 #2348
1284
Record 5 #2356
Position 1024
Position 1036
Position 1245
Position 1276
Position 1277

FIGURE 4.a. Records in a linked file structure

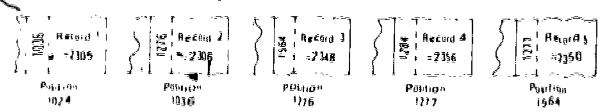

1036
Record 1 #2305
1276
Record 2 #2306
1564
Record 3 #2348
1284
Record 4 #2356
1277
Record 5 #2350
Position 1024
Position 1036
Position 1276
Position 1277
Position 1564

FIGURE 4.b. Updated file

3. *Indexed Sequential.* The records are stored sequentially in units, or blocks, and the locations of key records are entered in an index (Figure 5). This index shortens the time required to locate a record. With the methods described above, each record had to be scanned or read either to match the control number or to read the address of the link. With index sequential organization the index is scanned by a key, and a pointer is given to the area of the file where the record is located. The computer then reads only those records in the unit or range. To allow convenient addition of new records, space is usually left in the file at various points.[23] On multiplatter disks, the block or unit where

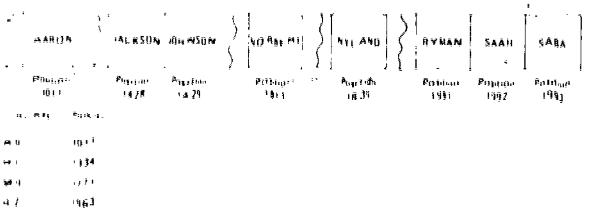

AARON
JACKSON
JOHNSON
NYLAND
RYMAN
SAAH
SABA
Position
1991
1992
1993

FIGURE 5. Records in an indexed sequential file structure

the sequenced records are stored often is tied to the cylinder, or sector, size of the disk.

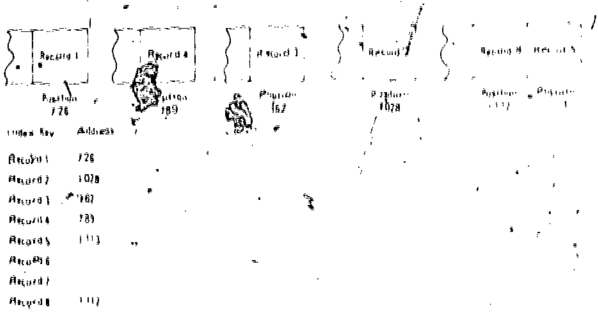
---

[21] U.S., Civil Service Commission, Bureau of Training ADP Management Training Center, "Management Introduction to Automated Data Bases," Washington, D.C., n.d., p. A-2. (Mimeographed.)

[22] Eckhouse, *Minicomputer Systems*, pp. 229–31.

[23] Hayes and Becker, *Handbook of Data Processing for Libraries*, p. 541.

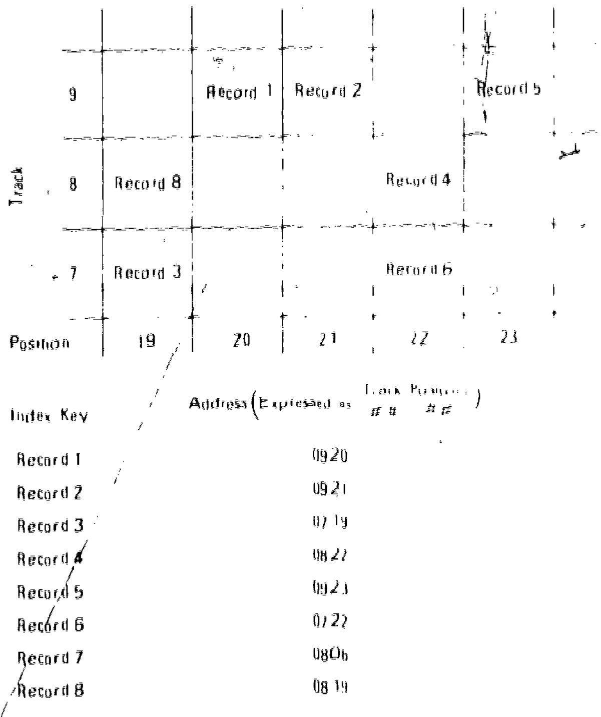
4. *Direct Access.* The record is stored at any available location and the key to the record and its address (location) are entered in an index (Figure 6).[24]



Record 1
Record 4
Record 3
Record 2
Record 8
Record 5
Position 726
Position 789
Position 167
Position 1078
Position 1113
Index Key
Address
Record 1 726
Record 2 1078
Record 3 167
Record 4 789
Record 5 1113
Record 6
Record 7
Record 8

FIGURE 6. Records in a direct access file structure

The illustrations to this point have been made linear for clarity, but disks are two-dimensional; in a random file a record could be stored in any position on any track, as illustrated in Figure 7.[25]



Track
9
8
7
Record 1
Record 2
Record 5
Record 8
Record 4
Record 3
Record 6
Position
19
20
21
22
23
Index Key
Address (Expressed as Track Position)
##
Record 1 0920
Record 2 0921
Record 3 0719
Record 4 0822
Record 5 0923
Record 6 0722
Record 7 0806
Record 8 0819

FIGURE 7. Two-dimensional depiction of a direct access file structure

The physical arrangement of the data on the medium, the form they take, the speed with which they can be accessed, and the storage capacity all vary, and these factors must be considered when a computer configuration is being assembled.

Input

In the discussions of manipulation and storage of data so far, data have been described as being in binary (on-off) or machine-readable form. A person who wishes the data to be processed begins with information in its "natural," original form. This information is human-readable and is generally called the source document. The information must be converted to machine-readable form, and this step is accomplished via an input device. Two processes are involved: converting the human symbol into code and converting the code into electronic impulses.

Numerous techniques are used, and many pieces of equipment are available. Some require a two- or three-stage conversion, and others allow direct input. Punched holes in cards or paper tape are "read" by a machine that allows current to flow wherever there is a hole, giving an "on" or "one" impulse. Magnetic ink is used and "read" by machine to produce impulses. A light pen can be used to "read" a bar code; the lines are used to produce impulse patterns. Optical character recognition is more complicated, but it is similar to the light-pen system. The lines of the character are "read" onto a matrix and the resulting pattern is compared to a reference table; when a match is made, the impulses for that character are created.

A human symbol, a letter or number, must appear as an electronic impulse in the binary, on-off scheme of the computer. Most input devices allow the person to use human symbols and then convert the symbols to machine-readable codes and ultimately to impulses. The initial input device used is often offline and independent (standalone), like a keypunch machine. The converted data it produces can be "read" and added to the computer storage as desired.

Output

Just as source data must be converted to machine-readable form to be manipulated by the CPU, so too the processed data must be converted to human-sensible form to be meaningful to the system user. In most cases only a portion of the data in the system is needed at any one time. The output must be selective, formated as required, and expressed in a suitable medium. The output is also called a "display."[26] Most displays are visual and in a printed form (although computers can produce oral responses). The most common is the printout on continuous-form paper, printed on a fast, computer-driven printer. Photocomposition is possible as an output. A computer tape is produced that is formated and ready to use in an automatic typesetting machine like the Government Printing Office's Linotron. Computer-output microforms (COM) are also a form of printed display. "Soft" displays are produced on a cathode-ray tube like a television screen. This gives the user human-sensible data immediately but leaves no permanent record of the data.

---

[24] Eckhouse, *Minicomputer Systems*, p. 229.

[25] R[ichard] T. Kimber, *Automation in Libraries*, 2d ed. (Oxford: Pergamon Press, 1974), pp. 50-51.

The Configuration

The input, output, and storage units, together with the central processing unit, combine to form the computer hardware system or configuration. In the literature, systems are described in various ways. Some sources show the entire system as "the computer"; others call only the central processing unit the computer and all the other devices peripherals. The CPU, encompassing the arithmetic and logic unit, the control unit, and main memory, is also called the mainframe. The block diagrams in Figure 8 represent some common depictions of computer configuration.[27] However, the computer is depicted, the system is operated by the CPU, specifically the control unit.

The connection of these units and their interfacing can be very complicated and involve both hardware and software control. The speed of operation of the several units can vary within one system. A card reader may input 80 characters of data at a much slower rate than a tape drive can read them, one printer may print 300 lines per minute, while another operates at 1250 lines per minute. Some devices are farther removed physically from the CPU than others, and the time required for their electrical impulses to travel is greater. Some processing operations take longer than others. The control unit must consider all these elements and must sequence and time the impulses and events accordingly.

To the human, the speed of the CPU is so great that operations may seem to occur simultaneously. This is not true, however; the computer operates in a linear, one-step-at-a-time manner. The CPU instructs an input/output controller to read or write information, and the I/O controller performs the actual transfer of data between internal memory and the peripheral devices. There actually may be several I/O controllers in a system, each suited to a particular class of input/output device. Sometimes the controller is an integral part of the input/output device. A buffer is often used as part of the interface. It is a means of temporarily storing data until they are needed. The buffer storage can be designated as part of the main memory or as part of the controller. In all, the interface physically connects the units of the configuration and serves to compensate for differences in the speed of data flow, in the timing of the movement of data from one device to another, and in the kinds of codes or formats involved.

SOFTWARE: COMPUTER PROGRAMS

A computer is directed to manipulate data through predefined sequences of instructions called *programs*.[28] Computer instructions at their most fundamental level are expressed in binary codes and are referred to as the machine's language.

Machine language is extremely crude (low semantic), and it is difficult and time-consuming to manipulate. To ease the burden of program preparation, special programs are provided to allow humans to express problem solutions in functional terms with the object of constructing the machine language required to implement the program. These special programs are called assemblers, compilers, or interpreters, and the statements one codes to express problem solutions are themselves called programming languages. Some programming languages are very close to that of a computer's own language and are suited for use only on that computer. These are called assembly languages. Other programming languages are structured to express a problem's solution in human or near-human language. These are the compilers and interpreters and are called high level programming languages. COBOL, FORTRAN, and BASIC are among the most commonly used high-level languages. High-level languages have the advantages of ease of program development and machine independence; a program

---

[26] Heiliger and Henderson, *Library Automation*, p. 199.

[27] Joseph Becker and Robert M. Hayes, *Information Storage and Retrieval: Tools, Elements, Theories* (New York: John Wiley & Sons, 1963), p. 113; Eckhouse, *Minicomputer Systems*, p. 4; Hayes and Becker, *Handbook of Data Processing for Libraries*, p. 45.

[28] Charles H. Davis has prepared a workbook type book on programming library applications using PL/1 on a large scale computer, see *Illustrative Computer Programming for Libraries, Selected Examples for Information Specialists* (Westport, Conn.: Greenwood Press, 1974).

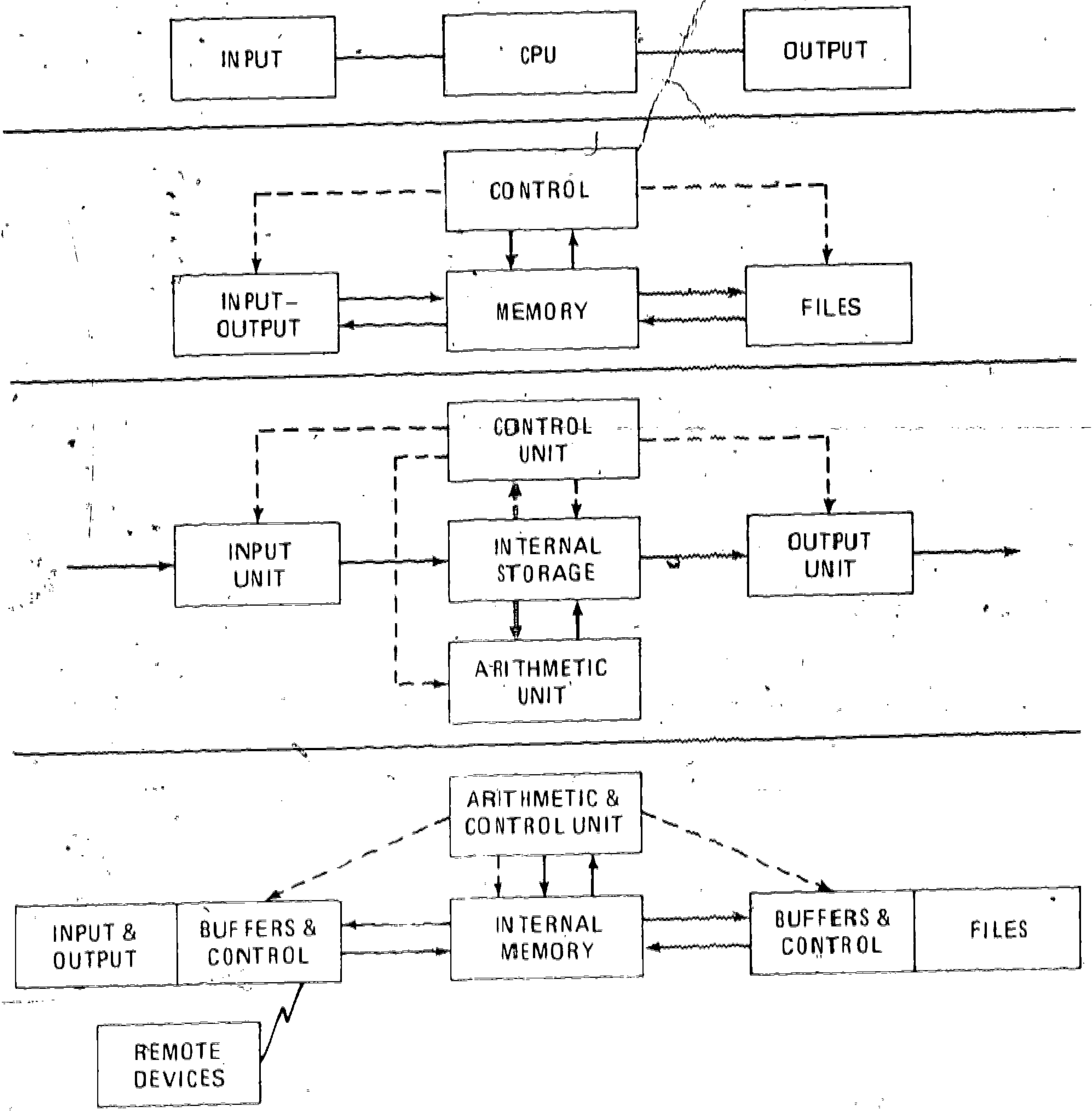

INPUT
CPU
OUTPUT
CONTROL
INPUT-OUTPUT
MEMORY
FILES
CONTROL UNIT
INPUT UNIT
INTERNAL STORAGE
OUTPUT UNIT
ARITHMETIC UNIT
ARITHMETIC & CONTROL UNIT
INPUT & OUTPUT
BUFFERS & CONTROL
INTERNAL MEMORY
BUFFERS & CONTROL
FILES
REMOTE DEVICES

FIGURE 8. Sample block diagrams of a basic computer configuration

written in one of the high-level languages may be run on any computer having a compiler or interpreter for that language.

In preparing a computer program, the objective must be defined and the procedure to accomplish the objective established. These items usually are expressed in a flow chart. Using the flow chart, the steps to be followed to accomplish the objective can be expressed graphically. Some procedures are simple linear paths (Figure 9). Some call for branch logic based on decisions (Figure 10). Some require repetitive steps, accomplished by using a loop (Figure 11).

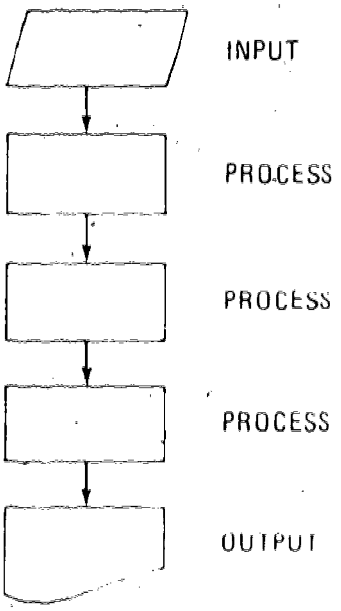


FIGURE 9. Simple linear path flow chart

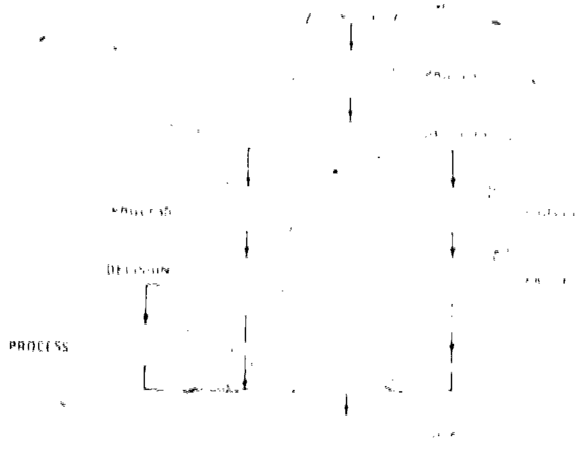


FIGURE 10. Branched logic paths flow chart

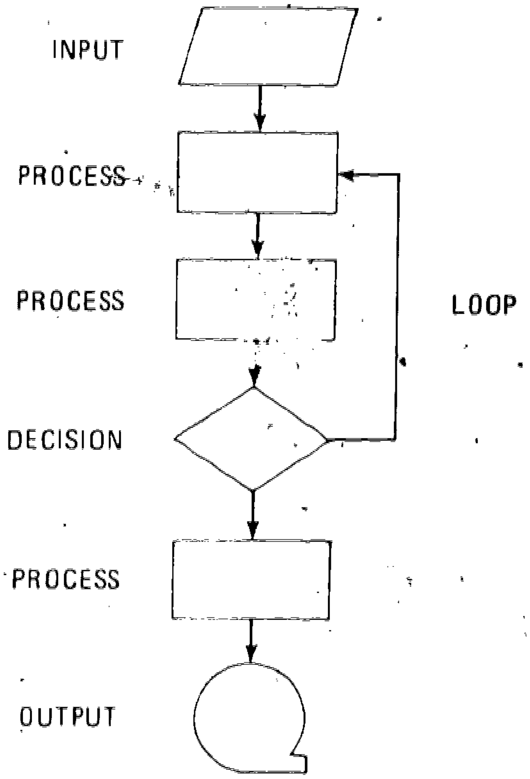


FIGURE 11. Loop flow chart

The more complicated the procedure the more complex the flow chart (Figure 12).[29]

Through a series of flow charts, each more specific and detailed, the procedure finally is reduced to the elementary steps in the proper sequence. This final flow chart is what is expressed in the terms of the programming language. Several routine functions must be considered with every program; their form depends on the conventions of the programming language used.

Whether it is written in assembly or high-level language, a program must ultimately be in the machine's language. Computers understand five types of instructions.

- Input/output instructions
- Internal data movement instructions
- Arithmetic instructions
- Testing and comparison instructions
- Jumping or branching instructions.

---

[29] Heiliger and Henderson, *Library Automation*, p. [illegible]

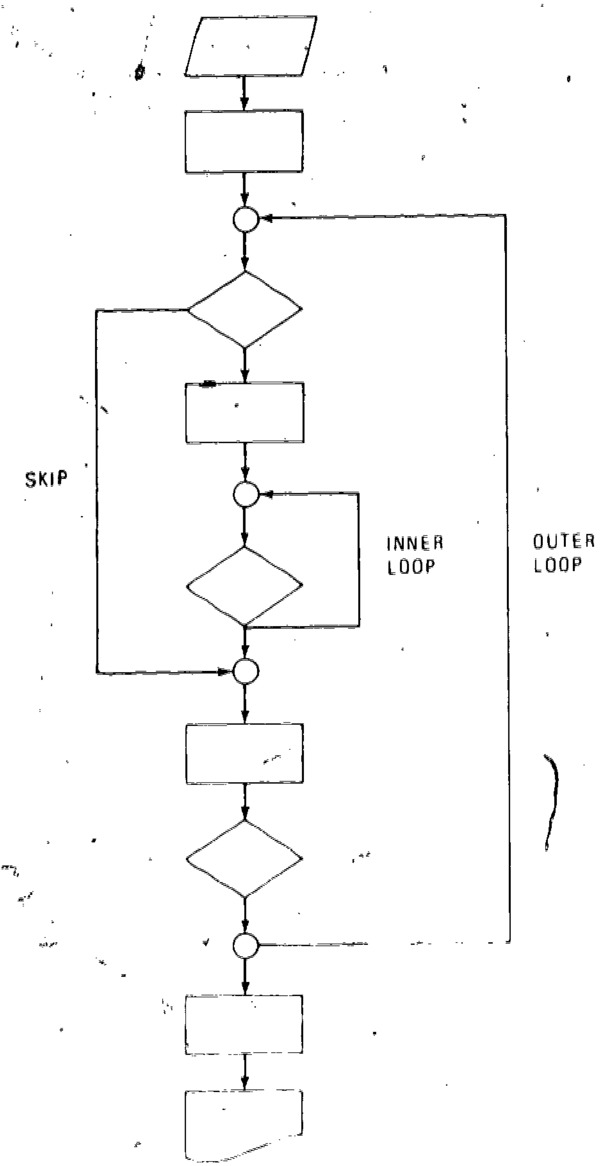


FIGURE 12. Repetitive loops flow chart

The number and power of instructions in each category vary significantly from one computer to another. While one computer may understand 20 arithmetic instructions, another may understand only 2. The complete set of instructions in each category plays a major role in the suitability of a computer for a particular application.

The basic structure of a computer instruction is as shown in Figure 13. The *op* part of a computer instruction defines the operation to be performed, and the address(es) part defines the memory location of the data to be manipulated. Some operations may require a single address, and others may require two or more.

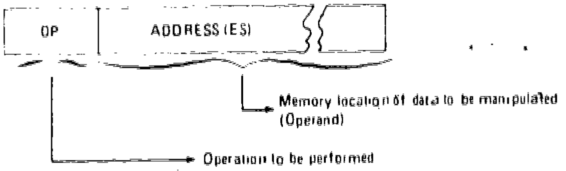

OP
ADDRESS(ES)
Memory location of data to be manipulated (Operand)
Operation to be performed

FIGURE 13. Basic structure of computer instruction

Instruction addresses in some computers may be expressed in terms of the sum of two or more values. Typically these types of addresses cause the content of the address portion to be incremented or decremented by one or more special machine accumulators called index or general registers. Still other computers provide for specification of an address that locates not the data to be manipulated but rather the address of the data. This feature is called an indirect address. Thus a computer instruction address may appear as in Figure 14.

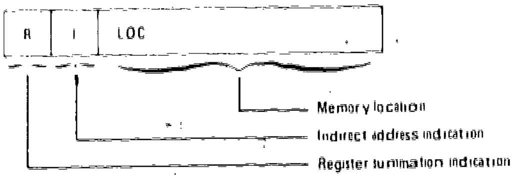


FIGURE 14. Address portion of a computer instruction using the indirect addressing technique

Some computers have no provision for *R* or *I*, and others provide for only one or the other. If neither indexing nor indirect addressing is provided, programming can be very difficult indeed.

In high-level programming languages the ability of the computer to index or indirectly address is unimportant to the programmer because the high-level language requests functional operations rather than detailed machine-level instructions. The compiler or interpreter produces the code required to perform the requested functions, based on the capabilities and limitations of the computer for which it compiles or interprets.

Certain computer instructions cause activity that must be communicated to the program. As an example, a data input instruction should communicate to the program whether or not an error has occurred during the reading of data. Thus, computers generally maintain a special internal status indicator called the condition code. Usually the condition code is only four bits long, and each bit indicates a special condition based on the last instruction executed. Table 3 illustrates the meaning of the bits of the condition code based on representative instructions.

Table 3—Breakdown of Condition Codes for Representative Instructions

	Condition Code			
Instruction	Bit 0	Bit 1	Bit 2	Bit 3
Input	End of file	Error		
Output	End of Media	Error		
Compare	A = B	A B	A B	
Add	Result 0	Result +	Result	Overflow
Subtract	Result 0	Result +	Result -	

In machine-level or assembly language the condition code may be tested following an instruction, and the sequence of executing instructions may be altered as a result. The condition code is not apparent to the high-level programmer because the functional statements make no specific reference to it.

Input/Output Instructions

Data are transferred between memory and I/O devices through specially designed I/O instructions. Machine language and/or assembler language programming can become very difficult in this area unless the programmer has access to preprogrammed routines that perform these functions. Most I/O processes require that the operation go through the following phases:

1. Ready the device
2. Start the transfer
3. Wait for completion
4. Verify successful transfer

Unless preprogrammed routines are available these processes must be programmed. They can be more difficult to program accurately than the application itself.

High-level languages allow the transfer of data to be specified at a functional level (READ or WRITE), with the compiler or interpreter automatically producing the machine-level code required for all phases of the operation

Internal Data Movement Instructions

Data may be moved from one memory location to another, from memory to registers, from registers to memory, and from register to register. The computer must therefore have instructions to perform each type of movement, and the machine-level or assembler language programmer must be aware of these

With high-level languages, instructions to transfer data to or from registers are not necessary. The computer makes the transfers automatically. Therefore, high-level languages do not need to distinguish these differences. The high-level language programmer need only specify the requirement for a transfer, and the compiler or interpreter will produce the machine-level instructions to see that it occurs. Furthermore, the high-level language programmer need not be concerned with the fact that some computers must move data a single character or word at a time while others may move blocks. Again, the compiler or interpreter produces the requisite machine instructions as needed.

Arithmetic Instructions

Arithmetic instructions perform addition, subtraction, multiplication, division, and at machine level, certain other functions such as changing sign or shifting. Some computers provide special instructions for floating-point operations.

Arithmetic typically is performed in specially designed accumulators or registers (see above in indexing) but sometimes may be performed in memory. All modern computers perform binary arithmetic, and most provide decimal arithmetic or conversion from one form to the other.

High-level languages allow programmers to define their own accumulators upon which arithmetic functions are to be performed. The compiler or interpreter will provide the necessary binary-decimal conversion machine instructions. Further, high-level languages usually provide statements that define the equation to be solved, including sine, cosine, and square root functions. The machine-level solution to these statements may be very complex and may require literally dozens of individual instructions.

Testing and Comparison Instructions

These instructions are used to establish the condition code reflecting the relationship of one datum to another. At the machine and assembler levels, programming these instructions would be followed by conditional jump or branch instructions that alter the sequence of program execution based on the settings in the condition code.

High-level languages provide sophisticated testing and comparison techniques, which typically allow the programmer to write such statements as IF, THEN, and ELSE to express conditional processes. Typically, the IF statements allow construction of AND and OR functions that establish complex relationships to determine processing. Such functions require many separate machine- or assembler-level instructions that successively test and jump.

Jumping or Branching Instructions

Jumping, or branching, instructions are used to alter, conditionally or unconditionally, the sequence of program execution by the CPU. For example, one may follow an instruction to COMPARE with an instruction to JUMP IF EQUAL or JUMP IF HIGH. Unconditional jumps are used to transfer control regardless of the setting of the condition code.

High-level languages usually provide a GO TO type of instruction to perform unconditional changes in processing sequences, but handle conditional changes through IF, THEN, and ELSE instructions.

Programming Example

To illustrate the concepts of programming, consider the problem of converting a two-digit month number (01–12) to a three-character abbreviation for the month (Jan–Dec). Program this function using machine, assembly, and high-level languages. The languages used are not actual but are representative of those available.

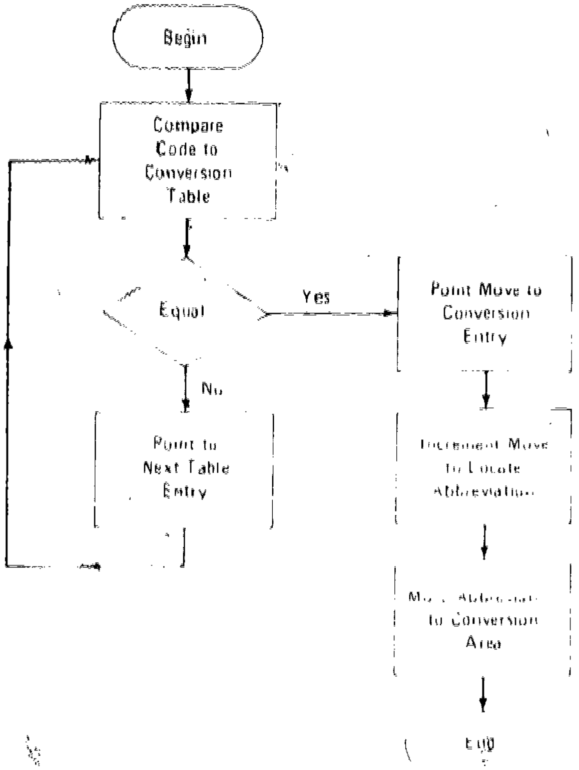

Begin
Compare Code to Conversion Table
Equal
Yes
No
Point to Next Table Entry
Point Move to Conversion Entry
Increment Move to Locate Abbreviation
Move Abbreviation to Conversion Area
End

FIGURE 15. Program flow chart of month number to month abbreviation conversion

Figure 15 is a flow chart of the program in machine language terms. Tables 4 and 5 give the basic elements of this program. The text following explains their significance.

TABLE 4—Memory Definition

Memory Location	Data Content	
001	01JAN	TABLE OF MONTHS
006	02FEB	
.	.	
056	12DEC	
061	03	TWO DIGITS FOR CONVERSION
063	. . .	OUTPUT OF CONVERSION

TABLE 5—Machine Language Program Elements

Instruction Number	Memory Location	OP*	Data Length	Address(es)	
1	100	CP	02	061	001
2	110	JE		130	
3	115	AD	03	200	107
4	125	JU		100	
5	130	MV	03	107	154
6	140	AD	03	203	154
7	150	MV	03	000	063
	200	005			
	203	002			

*CP = Compare
JE = Jump if equal
AD = Addition
JU = Jump unconditionally
MV = Move data

In this program, memory locations 001–060 contain a table of two digit months with their three-character abbreviations. Memory locations 061–062 contain the two-digit code (03) to be converted, and locations 063–065 are reserved to contain the converted results. The instructions begin at location 100, and are performed as follows:

1. Compare two positions at 061–062 with the data at the address shown (initially 001).
2. Jump if equal to location 130.
3. Add the three-digit code in locations 200–202 (005) to the second address of instruction 1, thus causing the instruction to compare on the next entry in the month table.

4. Unconditionally jump to location 100 (the compare instruction as now modified by instruction 3).
5. When the appropriate conversion table entry is located, move the second address of instruction 1 (locations 107–109) to the first address in instruction 7. Thus instruction 7 is constructed to point to the month abbreviation table entry.
6. Add locations 203–205 (002) to the first address of instruction 7. Thus instruction 7 locates the abbreviation.
7. Move the abbreviation to locations 063–065.

Assembly Language Terms

Table 6 gives the elements of this program in assembly language terms.

Table 6—Assembly Language Program Elements

Reference	Symbol	OP	Address(es)
1	TABLE	DC	'01JAN'
		DC	'02FEB'
			DEFINE MONTH TABLE
2	INCODE	DC	'12DEC'
3	OUT-CODE	DC	'_____' RESULT FIELD
4	LOOP	CP	INCODE(2),TABLE
5		JE	FOUND
6		AD	FIVE(3),LOOP + 7
7		JU	LOOP
8	FOUND	MV	LOOP + 7(3),LOOP + 4
9		AD	TWO,SAVE + 4
10	SAVE	MV	0(3),OUTCODE
11	FIVE	DC	'005'
12	TWO	DC	'002'

This assembly language program parallels the machine language coding required in the first example. Notice, however, that the actual memory locations of data in the addresses need not be coded. Instead, assign symbolic names to the data the assembly language program will use to supply the actual memory addresses. Still, one assembler instruction for each machine instruction must be coded, and only instructions that are actually part of the repertoire for the computer can be coded. The following explains the program:

1. The symbol TABLE is assigned to the first entry DC (define characters) for the Month Table. This is followed by eleven other DC's that complete the table.
2. INCODE is the symbol assigned to the characters representing the code to be converted.
3. OUTCODE is the symbol assigned to the result field.
4. LOOP is the symbolic name given to the compare instruction. Note that the instruction is coded using the symbolic names of the fields it acts upon. Also note the (2) defining the length of the fields to be compared.
5. JE alters control if the condition code is set to EQUAL after the comparison. Note the use of the symbolic name FOUND instead of an actual memory address.
6. Here add 5 to the second address of LOOP. These instructions parallel exactly those required by the computer in the machine language example. However, the memory addresses in symbolic terms can be written, using LOOP + 7 to locate the positions to be incremented and FIVE to locate the amount of the addition.
7. Here return to LOOP for continued searching.
8. Match the code in INCODE to one in TABLE, and move the second address of the compare instruction to the first address of the SAVE instruction.
9. Add 2 to the address in SAVE to locate the first of the three characters to be saved.
10. Move the abbreviation to the result area. Note that this instruction is filled in by instructions 8 and 9.
11,12. These are the numeric values needed to perform the additions in instructions 6 and 9.

High-Level Language Terms

Table 7 gives the elements of the program in terms of a high-level language.

Table 7—High-Level Language Program Elements

Reference	Commands			
1	MONTH	DEFINE	'03'	MONTH CODE
2	RESULT	DEFINE	'___'	RESULT FIELD
3	TABLE	DEFINE	(2,3) '01JAN' '02FEB' '12DEC'	MONTH TABLE
4	LOOKUP MONTH IN TABLE GIVING RESULT			

Using the high-level language concept this problem can be solved through functionally related statements as follows:

1. Define the month code to be converted.
2. Define the conversion output location.
3. Define the conversion table with the characteristics of two-digit code = three-character output.

4. Specify the conversion by LOOKUP of MONTH IN TABLE GIVING RESULT. That is, the compiler will produce the code necessary to search TABLE and move the located abbreviation to RESULT. The code will be essentially that shown in the Machine Terms example.

OPERATING MODE

In all computer systems the hardware and software work together to perform the functions. However, the way they work together can vary. The system can be batch, remote batch, online interactive, or time-sharing. The facility can be dedicated, shared, or time-shared. These conditions all affect the final outcome or product.

Dedicated, Shared, or Time-Shared

A *dedicated* facility is one used for only one customer or unit.

A *shared* facility serves more than one customer or unit. The computer must be scheduled to meet the various customers' requirements. The equipment configuration, the software, and the staff must be able to meet the requirements of all users, which can be diverse.

A *time-shared* facility is often a commercial service or company in business to serve users on an equal basis. The computer is removed from the user, and access or control is remote. The software is very general and often not suited to special applications.

Batch, Remote Batch, Online Interactive, or Time-Sharing

A *batch* operation is one in which a number of transactions to be processed are accumulated and processed together. Usually they are sorted into sequential order and matched against affected files.[30] In most facilities a computer operator initiates the operation: he calls up the program, mounts the tapes for auxiliary storage, starts the input device, and readies the output device(s). When processing is complete, the operator wraps up the operation and goes on to the next customer's job.

A *remote batch* operation is a combination of online and batch processing. The data are gathered online and held in a queue; then the processing of the data is performed in a batch mode according to a priority schedule.

In an *online* operation, the input and output units are in direct, continuous communication with the central processing unit.[31] The data are acquired online and used in conjunction with direct-access files so that, as a transaction is received by the computer, relevant data are retrieved from the files and processed, and the results are transmitted immediately to the user. Most online systems are interactive: the user and the computer are in direct communication, and a dialog can occur in real time.

The type of computer operation, the type of computer facility, the hardware configuration, and the software all are interrelated and each affects the others. The entire system must be kept in balance and harmony. For example, some input devices are inappropriate for online or interactive operations. Some storage devices are too slow for online operations. Some programming languages are for batch operations only. Some operating systems are suited for one type of processing only. The structure of the files for batch processing may be inappropriate for online processing.

In computer operations such factors as the speed of the computer, the size of the storage capacity, the size of the internal memory, the speed and capabilities of the peripheral units, the mode of computer operation, and the flexibility and sophistication of the software should be considered. These items must be evaluated in terms of cost and measured against the objectives and requirements of the system applications.

ROLE OF THE COMPUTER

Computers were developed to do complex numerical analyses and computations in the scientific and engineering fields.[32] Their next application was in business and industry for routine processing of data involving little sophisticated computation but many repetitive operations. The third application was in the area of symbol or language manipulation, including natural language, which is based on logical operations. This area is still being developed, and its potential has not been realized. Heuristic applications are also possible, due to a computer's speed and ability to alter its stored programs. This type of application includes the use of computers to play chess and checkers and to write music and poetry. The final application is in control. Computers are used to control industrial processes, military systems, and even learning, as in computer-assisted instruction. More applications no doubt will be developed as the human imagination expands and technology advances.

---

[30] Hayes and Becker, *Handbook of Data Processing for Libraries*, p. 661.

[31] Ibid., p. 664.

[32] Ibid., pp. 241–4.

The use of computers has become pervasive in society and commonplace in daily life. Paychecks, bank statements, creditcard bills, form letters, and bulk mail are computer-produced. Computer terminals are used in airports and hotels to make reservations, and grocery stores and department stores have cash register terminals. Names on tax rolls, voting lists, social security lists, census lists, and car registration lists are all in computer files. The electricity in homes is controlled by a computer system, and the telephone system is totally automated.

In all these cases the computer provides accuracy, consistency, reduction of duplicated efforts, speed, control, cost reductions, increased accountability, and more efficient use of human skills.

1. *Accuracy*. Once data are entered correctly, they remain correct and precise. The human errors of mistyping, misfiling, misadding, or transposing letters or numbers are eliminated.
2. *Consistency*. The same type of data will be handled the same way every time. The same logic is applied and the same decisions made every time.
3. *Reduction of duplicated effort*. The data in a computer can be recorded once in a master file and manipulated and sorted repeatedly as required.
4. *Speed*. The speed of the computer is unequalled. Even output is faster on a computer than most similar kinds of manual output would be.
5. *Cost reduction*. In most applications the use of a computer is cost-effective when compared with the cost of the same tasks performed manually.
6. *Control*. Because a computer system is tightly ordered by the software, control of the flow is possible, the status of the system is always determinable, and exception reporting is easily accomplished.
7. *Accountability*. Because of the way computer programs work, the computer can easily count operations. Statistics are easily gathered, which can give management valuable information for decisionmaking.
8. *Efficient use of human skills*. Because data can be entered once and used over and over, as in producing address labels for repeated mailouts, human time can be put to more productive use. Clerical staff can usually be reduced or reassigned to new tasks.

Yet computers are not a cure-all for the ills of the world. The computer is only as smart as its programmer. Output data are only as valid as the input data — the expression is "garbage in, garbage out." An operation can be paralyzed if equipment breaks or a system goes down and backup is not available. Systems can be misapplied; software designed for one purpose is sometimes used for another purpose, with the result of a mismatch of requirements and results. Ultimately, the computer is as effective as the human beings who manage it.

CHAPTER TWO

ROLE OF COMPUTERS IN LIBRARIES

DIRECTIONS IN LIBRARY AUTOMATION

Like other segments of society, libraries have used computers in a number of ways. In general, though, full utilization of automation in libraries is yet to come. The question of how to apply computers to the functions of a library is answered differently by various writers. Usually, though, the traditional division of the library into public services, technical services, and administration is replaced by new arrangements of roles.

Cox, Dews, and Dolby classify library activities as *acquisition, processing,* and *dissemination.*[1] The library acquires materials, processes them to reveal as much as possible of the information they contain, and then makes the information available to those who need it, and even to some who never knew they needed it. The major effort involved is in handling the records of the materials rather than the materials themselves, and there are many different records for each item.[2] "The functioning of the library depends on the speedy and efficient handling of these records — their creation, consultation, amendment and updating,"[3] which creates an enormous data handling problem and one that is well-suited for computer application.

Swihart and Hefley recognize three main areas of library operations: library administration, library management, and library science.[4] Library administration encompasses acquisition, cataloging, and circulation. Library management involves the staffing, budget, and reporting aspects of acquisition, cataloging, and circulation, as well as the other areas of the library. Library science involves the materials, the catalog, the rules and schemes applied to the materials and catalog, and services to the users. Swihart and Hefley conclude that "automation is of most value and is applied primarily to [library administration]."

Hayes and Becker place computer applications in libraries into three categories: clerical functions, information storage and retrieval, and operations research.[5] Computers can be applied to the routine clerical functions of technical processing and circulation work to reduce the clerical burden and increase the library's ability to perform more work. Computers are applied to reference work in the field of information storage and retrieval. At the most complex level "the objective is to develop new methods for automatically aiding various intellectual processes, such as extracting meaning from text and correlating facts or inferring subject relationships from the complete content of articles and books." The areas of operations research and systems analysis require the use of "the computer as an aid in using the principles of scientific management in library administration" through provision of mathematical models and computer simulation.

Heiliger and Henderson see the library as not one system, but three.[6]

1. *Technical processes* are concerned directly with the acquisition of library materials and their cataloging, preparation for later use, and reference to the accumulated holdings.
2. *Control processes* are applied to the library's resources and to the handling of library materials and the processing of data about them, such as circulation, inventory, and work control information.

---

[1] N. S. M. Cox, J. D. Dews, and J. L. Dolby, *The Computer and the Library; The Role of the Computer in the Organization and Handling of Information in Libraries* (Newcastle upon Tyne, Eng.: University of Newcastle upon Tyne Library, 1966), pp. 14–15.

[2] Ibid., p. 11.

[3] Ibid., p. 15.

[4] Stanley J. Swihart and Beryl F. Hefley, *Computer Systems in the Library: A Handbook for Managers and Designers* (Los Angeles: Melville Publishing Co., 1973), pp. 11–15.

[5] Robert M. Hayes and Joseph Becker, *Handbook of Data Processing for Libraries*, 2d ed. (Los Angeles: Melville Publishing Co., 1974), pp. 5–6.

[6] Edward M. Heiliger and Paul B. Henderson, Jr., *Library Automation: Experience, Methodology, and Technology of the Library as an Information System* (New York: McGraw-Hill Book Co., 1971), pp. 7–13.

3. *Administrative processes* are concerned with the organizational structure of the library and its operations; the main emphasis is on providing quick and accurate access to basic records and statistics for sound decisionmaking.

They further suggest that the organization of the library reflect these systems: a professional services department, a data services department, and an administrative services department, as well as a liaison services department to reach the library's public more effectively. They propose that if computer applications would follow this new pattern of functions and services, automation could be used to better advantage.[7]

The essence of all these approaches is that a library should not use a computer simply to automate an existing manual system. The library must look beyond tradition and expand its thinking to take best advantage of automation.[8]

This leads to the question: Why do libraries automate? Allen Veaner wrote that there are three major practical reasons for automating library functions:[9]

1. To do something less expensively, more accurately, or more rapidly;
2. To do something that can no longer be done effectively in the manual systems because of increased complexity or overwhelming volume of operations;
3. To perform some function that cannot now be performed in the manual system — providing always that the administrator actually wants to perform the service, has the resources to pay for it, and is not endangering the performance of existing services for which there is an established demand.

The ultimate goal of all automation in the library should be, directly or indirectly, to improve the service supplied to the library user.

HISTORY OF COMPUTERS IN LIBRARIES

In 1962 a study was made of the potential of applying advanced data processing in a university library. It was noted that[10]

> machine techniques have been applied successfully to a number of data handling and service problems that appear similar to those found in libraries, and it seems worthwhile to study the potential of routines that may suitably be applied to libraries.

The study concluded that[11]

> the speed of computers and the wide range of routines that they can handle offer possibilities for alleviating the pressures of personnel shortages and turnover, of increased volumes of inventory and user activity, and of demand for more up-to-date records.

In general this rationale explains why libraries become involved in automation. Many libraries had their first experiences with automation in the form of the payroll, personnel, and budget control systems used by their parent organizations. Even today these business applications are the only computer uses found in many libraries.

There was little library automation before the 1960s. Until then libraries used ADP (automated data processing) equipment such as punch card readers, sorters, and collators. This approach was used in a library by Ralph Shaw as early as 1948. A survey of Federal libraries showed only five automated systems in operation in 1960 or before.[12] The survey, through 1970, showed applications throughout the library. Table 8 lists these applications and the numbers of libraries using computers to serve them.[13]

A 1971 survey of all types of libraries (academic, public, school, special, State, and Federal) also showed applications throughout the library (Table 9).[14]

Markuson and her collaborators evaluated the two surveys and concluded,[15]

> . . . most large-scale library automation activities occur in an environment that has relatively sophisticated computer equipment and a fairly large and active computing center, and is partly oriented towards research activity.

These conditions occurred most often in academic libraries and in Federal technical libraries. In these facilities, computers were already on hand and in use

---

[7] Ibid., pp. 93–95.

[8] Gerard Salton contends that information science and computer science will never combine effectively into a complete, integrated system in a library unless the library processes are totally restructured based on computer efficiency requirements; see *Dynamic Information and Library Processing* (Englewood Cliffs, N.J.: Prentice-Hall, 1975).

[9] Allen B. Veaner, "Major Decision Points in Library Automation," *College and Research Libraries* 31 (September 1970), cited by Richard Phillips Palmer, *Case Studies in Library Computer Systems* (New York: R. R. Bowker Co., Xerox, 1973), p. 210.

[10] Louis A. Schultheiss, Don S. Culbertson, and Edward M. Heiliger, *Advanced Data Processing in the University Library* (New York: Scarecrow Press, 1962), p. 22.

[11] Ibid., p. 20.

[12] Barbara Evans Markuson et al., *Guidelines for Library Automation. A Handbook for Federal and Other Libraries* (Santa Monica, Calif.: System Development Corporation, 1972), p. 154.

[13] Ibid., p. 153.

[14] Ibid., p. 326.

[15] Ibid., p. 327.

TABLE 8—Automation Applications in Federal Libraries

Applications	Libraries	Number of Operational Systems
Acquisitions	10	7
Cataloging	32	27
Circulation	18	13
Serials	31	25
Information Retrieval	18	14
Bibliographic Publications	13	10
Selective Dissemination of Information	12	7
Abstracting and Indexing	4	3
Indexes to Special Collections	9	6
Other	3	3

Total Libraries = 59 Total operational systems = 115

TABLE 9—Automation Applications in All Types of Libraries

Applications	Total
Acquisitions	129
Cataloging	104
Circulation	149
Serials	169
Administration and Management	39
Abstracting and Indexing	23
Bibliography and Special Cataloging	87
Dissemination	40
Information Retrieval	34

for support of the research efforts — generally sophisticated applications in science and engineering. The research environment fostered a parent group receptive to the use of technology. The library thus had to serve sophisticated, demanding users. An additional factor contributing to the favorable conditions was the infusion of funds for scientific research in the 1960s — much more than was available for the humanities or public service activities.

It should be pointed out that much of the early library automation was in nontraditional areas, that is, other than acquisitions, cataloging, and circulation of books and periodicals. In fact more than one-third of the operational systems in Federal libraries, as reported by Markuson, were classed as information retrieval, bibliographic publications, selective dissemination of information, abstracting and indexing, and indexes to special collections. These systems dealt mainly with nonbook materials such as "research and development reports; journal articles; patents; trade literature, including catalogs; laboratory notebooks; pictures and photographs; maps; reprints; archival items; and technical correspondence."[16] Many of these early efforts were performed in documentation centers or information centers, not libraries. The techniques used were those of documentation and indexing rather than cataloging. The two important standardized developments in this area were (1) the ASTIA/DDC thesaurus of descriptors and (2) the COSATI standard for document description.

Library automation underwent evolutionary research and development.[17] Standard business systems could not be applied wholesale to the library; special software had to be created. Computer personnel had more experience in areas such as book ordering and fund control, but librarians were more interested in automating circulation, serials control, and cataloging.[18] A substantial learning period was required before much progress could occur. Standard input and output peripherals were often inadequate to handle the library's data. For example, they lacked upper- and lower-case characters and diacritical marks. Equipment had to be adapted and new technology developed in some cases, e.g., the American Library Association (ALA) print train.

In most of the earliest efforts each library worked independently to develop its systems. Redundant effort and incompatibility resulted.

Several developments helped advance library automation efforts:[19] (1) The organization of such groups as the Council on Library Resources, the Office of Science Information Service of the National Science Foundation, and the Committee on Scientific and Technical Information; (2) appropriations from major Federal library legislation; (3) formation of such associations as the Information Science and Automation Division of the American Society for Information Science; (4) development of the Library of Congress MARC II Communications Format for bibliographic data and the National Library of Medicine's MED-

---

[16] Lucille J. Strauss, Irene M. Strieby, and Alberta L. Brown, *Science and Technical Libraries; Their Organization and Administration* (New York: Interscience Publishers, John Wiley & Sons, 1964), p. 171.

[17] Allen B. Veaner, "Perspective: Review of 1968–1973 in Library Automation," in *Library Automation: The State of the Art II*, eds. Susan K. Martin and Brett Butler, papers presented at the Preconference Institute on Library Automation sponsored by the Information Science and Automation Division of the American Library Association at Las Vegas, Nevada, 22–23 June 1973 (Chicago: American Library Association, 1975), pp. 2–5.

[18] Heiliger and Henderson, *Library Automation*, pp. 13–14.

[19] Hayes and Becker, *Handbook of Data Processing for Libraries*, pp. 28–73.

LARS. These developments focused attention on library automation, promoted new efforts, made existing efforts more visible, exposed more librarians to automation and educated them in its uses, and provided the methods and funds to allow more libraries to participate in automation.

The single development with the most impact on library automation is the MARC II format. The MARC format is a standard, and standardization is the key to future advances.

> Without a standardized format for the transmission of bibliographic data, we might not exactly be no place in library automation, but we would certainly all be riding off in different directions, dissipating our resources, and enjoying not the remotest possibility of interchanging bibliographic data or building networks.[20]

MARC allows full expression of the nature of the bibliographic entity being described and is structured so that full access is possible. The tag structure is designed for maximum flexibility and allows each library to manipulate bibliographic data as it chooses.

The availability of Library of Congress cataloging in MARC format through the distribution system has caused more libraries to accept Library of Congress cataloging and has reduced the amount of duplicated effort in libraries throughout the United States. The level of effort necessary to convert catalog data into machine-readable form has been reduced. The availability of library cataloging in MARC format has made it incumbent on most libraries to have their computer systems designed to handle the MARC format. This brings about more compatibility among computer systems, at least as a means of sharing data, and it allows greater cooperation among libraries.

Acquisitions has a standard — the International Standard Book Number (ISBN). The ISBN is not as well developed and widely implemented as MARC and, likewise, automation in the acquisitions area is not as well developed as in cataloging.

CURRENT EVOLVING TECHNOLOGY

The last 4 or 5 years have greatly advanced library automation.[21] Computer hardware now features greater speed and storage capacity. More sophisticated operating and management systems allow more efficient use of the storage and CPU. Online data processing is widespread, and multiprocessing is common. Advanced and economical telecommunications techniques have been developed. Output printers now can be equipped with a special print font for upper- and lower-case characters and diacritical marks. Computer-output microfilm (COM) equipment to handle upper- and lower-case characters is available from more sources. Reliable, economical CRT terminals that can handle the idiosyncracies of library data are available. Light-pen and optical-character-recognition technologies have been refined. For the library these advances mean more efficient and economical automated services.

In a state-of-the-art review of library automation, Diana Delanoy identified four trends of the 1970s.[22]

1. Minicomputers for library applications as turnkey systems
2. Cooperative processing using networks
3. Online large-file searching of major bibliographic files
4. Packaged software for library applications.

Minicomputers

Minicomputers are used in libraries in the form of both turnkey and custom-designed systems. This book expresses the importance of minicomputers to the trend toward library automation. Subsequent sections will detail the uses of minicomputers in libraries.

Cooperative Processing

The network approach to processing has been tried by academic libraries and by State and regional organizations. The end of the 1960s saw the formation of the Colorado Academic Libraries Book Processing Center, the New England Library Information Network (NELINET), and the State of Washington's automated State library system. These networks were developed with economy in mind to allow greater numbers of libraries to benefit from the resources. Development and operating costs of the networks can be divided among all participants, and libraries that could not afford their own systems now have access to the speed and efficiency of automated systems.

---

[20] Veaner, "Perspective: Review of Library Automation," p.

[21] Herman H. Fussler's book, *Research Libraries and Technology; A Report to the Sloan Foundation*, has been called the most thoughtful, authoritative, and comprehensive treatment of this subject; (Chicago: University of Chicago Press, 1973). To review the more recent history of library automation, refer to the last two Aslib bibliographies for sources: Lynne Tinker, compiler, *An Annotated Bibliography of Library Automation 1968–1972* (London: Aslib, 1973) and Maxine MacCafferty, compiler, *An Annotated Bibliography of Automation in Libraries and Information Systems 1972–1975* (London: Aslib, 1976).

[22] Diana Delanoy, "Technology: Present Status and Trends in Computers," in *Library Automation: The State of the Art II*, eds. Susan K. Martin and Brett Butler, papers presented at the Preconference Institute on Library Automation sponsored by the Information Science and Automation Division of the American Library Association at Las Vegas, Nevada, 22–23 June 1973 (Chicago: American Library Association, 1975), pp. 20–23. See also Robert DeGennaro's four main lines of development in library automation in the mid-1970's which closely parallel Delanoy: "Library Automation: Changing Patterns and New Directions," *Library Journal* 101 (January 1976): 180–3.

The early networks did not all succeed. Some lasted only by changing course along the way. The success of the Ohio College Library Center (OCLC) was the single most important factor that established the feasibility of networks. The reasons for its success can be debated.[23] Factors mentioned include the dynamic personality of Frederick Kilgour, the "father" of OCLC. The initial participation of top administrators of the organizations whose libraries were to be involved had an effect. Other reasons cited are that standard products and procedures were provided with few individual user choices and that outside funds and grants were infused at propitious times. OCLC monographic cataloging services are well established, and serials cataloging services are being implemented. Acquisitions and serials control services have been studied but are not yet underway.[24]

Online Files

Online large-file searching of major bibliographic files is available to almost all libraries. The development of this technology depended on several factors.[25] Important hardware developments include the online, time-sharing, third-generation computer, expanded direct access storage capacity, acoustical couplers for reliable direct-distance-dial telephone connections, and cable and microwave communications for transmitting data over great distances. Important software developments are in the area of natural language (nonnumerical) processing, as well as techniques. The availability of large machine-readable data bases originally built to support publication efforts was vital. Commercial, service bureau telecommunication systems such as TYMSHARE and Telenet made access available by telephone lines with local or nearby long-distance charges; the leasing of costly dedicated lines became unnecessary.

The factor drawing these elements together is the commercial intervention of such firms as Systems Development Corporation and Lockheed in making multiple data bases available through one terminal and one hookup. The competition among these commercial services has kept prices within reach of almost any library with a terminal.[26]

Packaged Software

There still is not much packaged software for library applications. The presence of any, however, is a step forward. Previously, each library had to bear all of its own development costs. Transferring software between libraries has occurred, but not without problems. Programs developed by manufacturers and made available to other customers have been used, but again without widespread success. An example is the System 7 software developed by IBM for a circulation system that is available as a Field Developed Program (FDP).[27] In most cases, additional programming is required for the individual library application. Software developed for the Federal government, and thus in the public domain, is available at no cost. For example, the NASA library package is available from COSMIC.[28] Commercial firms offer software packages for lease or sale.[29] Blackwell North America has library processing software that can be purchased, as do Baker and Taylor and Systems Development Corporation, with their acquisitions modules.

Commercial Services

Closely related to the packaged software available for libraries is the number of software packages commercial firms use to support the library services they sell. Most of the major library jobbers and book wholesalers have access to the MARC tapes and sell computer-produced catalog card sets with overtyped headings, spine labels, and circulation book cards at reasonable prices, generally less than $1 per title. The smallest library thus can benefit from library automation. A set of cards ordered from the Library of Congress currently costs $.45 for eight unit cards (which may or may not be enough), all of which must have file headings added, entailing the effort of typing and proofreading each. A little figuring shows the saving from the commercial sets. Additional library automated services from commercial vendors are the book and COM catalogs. Several vendors, such as Science Press, use MARC tapes for input to computer-produced catalogs. The input keying required is decreased, saving time and effort.

---

[23] Brett Butler presented his views on the reasons of OCLC's success in "State of the Nation in Networking," *Journal of Library Automation* 8 (September 1975): 215–6.

[24] De Gennaro, "Library Automation," p. 181.

[25] Joseph Becker, "A Brief History of On-Line Bibliographic Systems," speech given at the UCLA Conference on Information Systems and Networks, 27–29 March 1974.

[26] Delanoy, "Technology," p. 22.

[27] Lois M. Kershner, "Management Aspects of the Use of the IBM System/7 in Circulation Control," in *Applications of Minicomputers to Library and Related Problems*, ed. F. Wilfrid Lancaster, papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974 (Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974), p. 44.

[28] Grants were given to two libraries for automation projects to be designed for transfer to other libraries, specifically the University of Chicago system and the University of Minnesota Bio-Medical Library system. DeGennaro, "Library Automation," pp. 177, 182.

[29] David L. Weisbrod, "Acquisitions Systems: 1973 Applications Status," in *Library Automation: The State of the Art II*, eds. Susan K. Martin and Brett Butler, papers presented at the Preconference Institute on Library Automation sponsored by the Information Science and Automation Division of the American Library Association at Las Vegas, Nevada, 22–23 June 1973 (Chicago: American Library Association, 1975), p. 91.

Computer-Output Microform

Computer-output microform (COM) is growing in importance in libraries.[30] COM is a cheaper, faster substitute for computer hardcopy output. COM catalogs are replacing card catalogs and book catalogs.[31] Elimination of binding and printing costs and of production turnaround time makes COM attractive and cost-effective compared to book catalogs, even considering the initial cost of purchasing viewers. The availability of COM with upper- and lower-case characters, bold and light faces, and alternate fonts such as italics has improved the quality of the image and made users more satisfied. COM reports are used in circulation systems and in acquisitions systems (for staff use where operations are batch or where permanent copy is needed for archival purposes).[32]

Telecommunications

The latest technology in library automation is telecommunications.[33] Mention has been made of this technology in terms of networks and online data base searching, for it is a major component of each. But telecommunications are developing to a point where further library applications are possible. Means of sending data efficiently and effectively are being developed all the time.[34] Computer data can be sent over voice-grade telephone lines. Commercial vendors are entering the field, and the resulting competition speeds advances in technology, improves services, and lowers prices. Microwave transmission and communications satellites, as well as cable television, are in use.

What does this mean for libraries? For many libraries, the telephone is the sole method of communications. Some libraries within the framework of larger systems have Teletype terminals for interlibrary loan transactions. For example, some States link academic libraries and State libraries via Teletype lines. OCLC studies of uses of its terminals indicate heavy use of the data base as a finding tool or union catalog for interlibrary loan searching. This connection among libraries will grow stronger.[35] The sharing of resources by facsimile transmission is inevitable[36] due to increasing production of information (the so-called information explosion) and decreasing funds for libraries. Facsimile transmission equipment has been tested in libraries but its use has been limited by its costs and the small number of libraries with the needed equipment.[37]

FUTURE APPLICATIONS OF NETWORKS IN LIBRARIES

> If someone walked up to you on the street and asked "What's happening in library automation today?", and you were limited to a one-word answer, your response would probably be NETWORKS.[38]

The word "network" means different things to different people. By its most basic definition, a network is "an interconnected or interrelated group of nodes," where a node is "an end point of any branch of a network, or a junction common to two or more branches of a network."[39] It can be disciplinary or problem-oriented or serve a general function. A computer network, which is the main interest of this book, is "an interconnection of assemblies of computer systems, terminals, and communications facilities."[40] It consists of links and nodes arranged in a given topology. At the simplest level a computer network may consist of a point-to-point connection of a host computer and a single communications input/output device. At the other end of the scale of complexity, a computer network can be made up of an interconnected group of computers, including processing systems (host, or main-site, computers and remote computers), communications control systems (e.g., data concentrators, message switchers, and front-end processors), and a variety of remote terminals and the transmission channels that connect the components.[41]

---

[30] For more detail on COM, refer to a book such as Robert F. Gildenberg, *Computer-Output-Microfilm Systems* (Los Angeles: Melville Publishing Co., 1974).

[31] Lois M. Kershner, "User Services: 1973 Applications Status," in *Library Automation: The State of the Art II*, eds. Susan K. Martin and Brett Butler, papers presented at the Preconference Institute on Library Automation sponsored by the Information Science and Automation Division of the American Library Association at Las Vegas, Nevada, 22–23 June 1973 (Chicago: American Library Association, 1975), pp. 45–47.

[32] Weisbrod, "Acquisitions Systems," pp. 93–94.

[33] *ARIST* has featured two essays that provide a background on telecommunications: R. L. Simms and Edward Fuchs, "Communications Technology," in *Annual Review of Information Science and Technology*, vol. 5, eds. Carlos A. Cuadra and Ann W. Luke (Chicago: Encyclopaedia Britannica, 1970, pp. 113–39, and Donald A. Dunn, "Communications Technology," in *Annual Review of Information Science and Technology*, vol. 10, eds. Carlos A. Cuadra and Ann W. Luke (Washington, D.C.: American Society for Information Science, 1975), pp. 165–93.

[34] Delanoy, "Technology," pp. 21–22.

[35] Hayes and Becker, *Handbook of Data Processing for Libraries*, pp. 6–24, discuss the concepts of library network applications and information network applications as two approaches for connection among libraries.

[36] Ibid., p. 23.

[37] Fussler devoted a chapter to this subject: "Chapter 4, Shared Resources, Photocopying, and Facsimile Transmission," pp. 30–50, in Fussler, *Research Libraries and Technology*.

[38] Bruce H. Alper, "Library Automation," in *Annual Review of Information Science and Technology*, vol. 10, eds. Carlos A. Cuadra and Ann W. Luke (Washington, D.C.: American Society for Information Science, 1975), pp. 205-6. For an overview of library networks, see the proceedings of the 10th Clinic on Library Applications of Data Processing held by the University of Illinois Graduate School of Library Science: F. Wilfrid Lancaster, ed., *Networking and Other Forms of Cooperation*, papers presented at the 1973 Clinic on Library Applications of Data Processing, 29 April to 2 May 1973 (Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1973).

[39] *Introduction to Minicomputer Networks* (Maynard, Mass.: Digital Equipment Corporation, 1974), p. C-18.

In general, networks can be said to provide a means for resource sharing that increases economy and convenience. Various kinds of resource sharing are possible.[42]

Device sharing. The ability to connect to and use the resources of a remotely located computing device as if it were local.

File sharing. The ability to read from, write to, or update files on a remotely located computing system as if it were local.

Program sharing. The ability to send a loadable program to a remotely located computing system to be loaded and executed by that system.

Program data sharing. The ability to open a data path between programs on an interactive basis, so that large tasks may be divided into smaller units for execution at different computer sites in the network.

Communications

The basis of any computer network is the communications link that allows interconnection between remote points. Networks have emerged as communications technology and computer technology have become integrated.[43] A computer network requires communications technology to function, and a data communications network uses computers to control its processes. The communications part of a network connects, coordinates, and integrates the network's various nodes.

Each node can be made up of any configuration of processors, terminals, controllers, and software; the nodes may vary considerably in speed, interfaces, languages, and other characteristics. These differences are handled by the three basic types of functions of a communications system: terminal functions, transmission functions, and switching functions.[44]

> At each source terminal, the originating information must be converted to a form suitable for transmission to the remote point. The signals must travel through whatever transmission media are in the path of the information flow so that, when they reach a destination terminal, it is possible to recover the form of the information sent with some reasonable precision. . . . Switching is needed when information must be sent to any number of destinations and it is advantageous to allocate use of parts of the transmission path to users only when they need them.

The communications network must bring all these components into an efficient system. The arrangement or topology of the network can vary.[45] It can be point-to-point, multipoint, centralized (or "star"), hierarchical (or "tree"), loop (or "ring"), distributed (or "multistar"), or fully distributed. For a library, any of these topologies is possible.

The main impact on libraries is due to the existence of commercial communication systems. Instead of having to own point-to-point private lines, libraries can use a public-switched telephone network such as Data-Phone 50 or can lease private lines from the telephone company or a specialized common carrier (WATS, TELPAK, TELEX, TWX, and MCI).[46] Of real value is a commercial service like TYMSHARE's TYMNET which provides a network-only service to customers who wish to attach their own computers to the network and offer service to their own user group.[47] Users can dial one of 80 metropolitan areas in the United States, Canada, and Europe without incurring charges for long-distance calls.[48] The National Library of Medicine has attached several computers to TYMNET.[49] SDC and Lockheed data bases are accessible through TYMNET.[50] The Federal library-OCLC project also has used TYMNET.

---

[40] A. J. Neumann, *A Guide to Networking Terminology* (Washington, D.C.: U.S., Department of Commerce, National Bureau of Standards, 1974), p. 7.

[41] *Introduction to Minicomputer Networks*, pp. 1-4—1-5.

[42] Ibid., p. 1-7.

[43] Besides the essays on communications technology (see reference number 33 above), *ARIST* presented two essays on computer technology; all four provide background on this subject: Harry D. Huskey, "Computer Technology," in *Annual Review of Information Science and Technology*, vol. 5, eds. Carlos A. Cuadra and Ann W. Luke (Chicago: Encyclopaedia Britannica, 1970), pp. 73-85, and Philip L. Long, "Computer Technology—An Update," in *Annual Review of Information Science and Technology*, vol. 11, ed. Martha E. Williams (Washington, D.C.: American Society for Information Science, 1976), pp. 211-22.

[44] Simms and Fuchs, "Communications Technology," p. 114.

[45] *Introduction to Minicomputer Networks*, pp. 2-1—2-11.

[46] Ibid., "Appendix B, Common Carrier Offerings," pp. B-1—B-4.

[47] Dunn, "Communications Technology," p. 177.

[48] "Tymshare," Brochure prepared by the Tymshare Information Services Division, 1976, p. 4.

[49] Dunn, "Communications Technology," p. 177.

Computers

If communications is the "how" of the network, the computer is the "what." A computer network can provide device sharing, file sharing, and program sharing. The ARPA network of the Department of Defense's Advanced Research Projects Agency was planned and developed as an experimental computer network to demonstrate the feasibility of interconnecting and sharing hardware and software systems.[51] When additional resources are required (e.g., additional storage or a faster printer), the process is shifted to another available node in the network.

A time-sharing computer network such as the General Electric Information Services Network[52] allows many remote users to have access to a computer facility which simultaneously processes many completely different problems and allows each user to act as if he or she had sole control.[53] A library can use a time-sharing service to access a computer if it hasn't one of its own, assuming of course that the necessary software is offered by the service bureau.

Information

Infinite variety is possible in the scope and application of networks. Information networks are used by airlines to search data bases of flight schedules and reservations. There is a network of Census Bureau data. There is also a network of stock market information, including Dow Jones averages, stock prices, statistics on the most active stocks, and historical statistical data on stocks.[54] The National Library of Medicine's MEDLAR network is made up of journal and report citations, not the materials themselves.

The major determining factor in the topology of such a network is the type of participation of the nodes.[55] They can be providers of resources exclusively, users of resources exclusively, or combinations of the two. A library can participate in any of the three ways. When a library signs up as a subscriber to the Lockheed or SDC data bases, it is signing on as a user only. When a library joins OCLC, it functions as a user and a provider, for a member library can contribute records to the data base.

Brett Butler, in his article "State of the Nation in Networking,"[56] defined a library network very strictly, but his main point was the distinction between the network itself and the network resource. The network is the organization and the "network resource is the computer system, data base, or service which is available for use by that organization."[57] In general terms, OCLC, BALLOTS, SOLINET, Washington State Library, NELINET, and AMIGOS, to name a few, are all considered functioning library networks.

Most of the network resources for libraries allow access to bibliographic data on library materials rather than to the materials themselves.[58] Future library networks probably will expand to allow access to the full text of library materials, and ultimately facsimile transmission of the texts.[59] Butler addressed some problems that must be resolved if library networks are to succeed and advance. Grosch selected the following as the major technical issues mentioned by Butler:[60]

- Authority controls on network data bases
- Interfacing of multiple data bases created by different institutions
- Direct network communications — computer to computer
- Access/document delivery coupling
- Data base ownership and standards.

Grosch adds her own assessment of the future of library networks.[61]

> Up to now, networks have addressed, for the most part, the provision of standardized bibliographic records or document delivery systems. The really significant cost benefit of networks will come when they can effectively handle the following: the coordination and development of collections among their members; inter-communication of bibliographic requests both within the network and to adjoining networks; and delivery of significant levels of integrated services of an online nature to their mem-

---

[50] Delanoy, "Technology," p. 22.

[51] Simms and Fuchs, "Communications Technology," pp. 127–9.

[52] Dunn, "Communications Technology," p. 178.

[53] *Introduction to Minicomputer Networks*, p. 1-2.

[54] Joseph C. Marshall, "Distributed Processing on Wall Street," *Datamation* 19 (July 1973): 45–46.

[55] *Introduction to Minicomputer Networks*, pp. 2-1—2-2.

[56] Butler, "State of the Nation in Networking," pp. 200–220.

[57] Ibid., p. 201.

[58] DeGennaro wrote of a service where it is possible to order copies through the online terminal of the items whose citations were retrieved in a search; "Library Automation," p. 181.

[59] Hayes and Becker, *Handbook of Data Processing for Libraries*, p. 23.

[60] Audrey N. Grosch, "Library Automation," in *Annual Review of Information Science and Technology*, vol. 11, ed. Martha E. Williams (Washington, D.C.: American Society for Information Science, 1976), p. 233.

[61] Ibid., pp. 245–6.

> bers.... [They must] come to grips with the following new systems concepts: the new generation of mini- and midicomputer hardware; distributive data processing wherein those functions better performed locally are done at the user's location; and development of replicable multipurpose software/hardware packages available for lease or purchase.

PROBLEMS IN CURRENT LIBRARY APPLICATIONS

Literature reviews indicate that library automation — application of the computer to routine operations and services in a library — is firmly established in the library world.[62] The decision for a library is not so much whether to automate, but rather when and how. Alper suggests that this continued application of computers in libraries is an evolutionary process that dictates continual review of past efforts to determine the characteristics that made them succeed or fail.[63] This information then can be applied in future applications.[64]

Most of the problems in the past have been in three main areas: poor or inadequate systems design; poor communications and misunderstandings between librarians and data processing professionals; and poor or inadequate data processing assignment, software, and access to personnel and equipment.

Poor Systems Design

Poor systems design and resulting project failure is not unique to library automation.[65] Libraries do have a propensity for systems problems because librarians in the past often have used informal and unscientific planning and management techniques. Data processing systems require a much higher degree of standardization, centralization, and uniformity than manual systems.[66] Cox, Dews, and Dolby spoke of a great cause of misunderstanding between the library and the computer center being what is *not* said about the details of various library operations.[67] The librarian assumes, that the implicit details of a library procedure are obvious. This problem can be eliminated if a proper, thorough systems design is performed, and a complete systems document is prepared. The following pitfalls have occurred from improper planning for library automation.[68]

- Setting improper goals
- Making an incomplete analysis
- Selecting wrong applications
- Allowing too little time to complete all phases of the system
- Underrating the cost of the installation or system
- Not keeping employees informed

Related to poor systems design is the mistake of automating manual procedures instead of automating with a view to what the computer can do and to what expanded services and applications are possible.[69]

Poor Librarian/Computer Personnel Interface

"It is abundantly clear that elegant technology cannot be exploited and complex systems cannot be sustained by hostile, fearful, or untrained personnel."[70]

Both library and computer center personnel have been guilty. Computer firms have not recognized the peculiarities of library operations and have to tried to install "another address list program" or "another parts inventory system." Librarians feel ill-prepared to cope with the mechanical, the mathematical, and even the logical aspects of library automation.[71] They often avoid automation in order to avoid feeling uncomfortable or inferior. They often defer too much to technical people, which leads to poorly conceived systems. As Markuson says, a librarian cannot say to someone, "Come automate my library."[72] A team approach must be used with all team members doing their homework. For example, the different definitions and meanings of such terms as "file," "list," and "record" in library and computer jargons must be recognized and dealt with.

---

[62] Ibid., p. 225.

[63] Alper, "Library Automation," p. 226.

[64] Chapter 9 of Salmon's book is entitled "Problems of Library Automation Systems." Patrinostro's book is a series of statements on library automation problems submitted by 55 different libraries. Stephen R. Salmon, *Library Automation Systems* (New York: Marcel Dekker, 1975) and Frank S. Patrinostro, compiler, *A Survey of Commonplace Problems in Library Automation*, The LARC Association's World Survey Series, vol. 11 (Peoria, Ill.: LARC Press, 1973).

[65] Alper, "Library Automation," p. 223.

[66] Schultheiss, Culbertson, and Heiliger, *Advanced Data Processing*, p. 21.

[67] Cox, Dews, and Dolby, *Computer and the Library*, p. 10.

[68] John J. Nicolaus, "Library Automation—How to Begin: Initiating A Library Automation Program," in *Initiating a Library Automation Program*, papers presented at the 1965–1966 meetings of the Documentation Group, Washington, D.C. Chapter, Special Libraries Association (Washington, D.C.: Special Library Association, Washington, D.C. Chapter, 1966), p. 47.

[69] R[ichard] T. Kimber, *Automation in Libraries*, 2d ed. (Oxford: Pergamon Press, 1974), p. 18.

[70] Heiliger and Henderson, *Library Automation*, p. 239.

[71] Palmer, *Case Studies*, p. xv.

[72] Markuson et al., *Guidelines for Library Automation*, p. 20.

Poor Support

Libraries often have suffered from a mismatch between their needs and software capabilities, hardware functions, and access time. At one level, the entire industry could not meet the needs. OCLC had to develop and engineer an extended-character-set CRT terminal,[73] and ALA had to sponsor development of a print train with the proper diacritical marks for printing library catalog cards. Most standard software packages cannot handle library filing. Most service bureaus do not provide programs for library operations.

At a local level, the data processing facility used by a library's parent organization often did not have the proper hardware or software to support the library's requirements. Libraries often found their access to the equipment limited and their priority low. Agency data processing staff had little time and little expertise for library systems. Library systems have suffered when hardware configurations have been changed without considering library needs. These problems have affected all types of libraries: academic, research, Federal, and public — all libraries that must rely on outside data processing support. Yet few libraries can afford their own computer facility.

Other Problems

In existing automated library systems, several problems have arisen that must be resolved before further growth of library automation will be possible. One need is for a technically and economically feasible means to convert cataloging records to machine-readable form and to make widespread access to those converted records possible. Independent efforts are too slow and costly, but full use of online bibliographic files as union catalogs is impossible without conversion.

Another need is for a means to transfer data and systems between libraries and between networks. This problem arises from lack of standardization, which is a result of the typical librarian's mind set and approach to management. It can be predicted safely that technology to accomplish such transfer will arrive before the human mind is psychologically prepared to use it.

FEDERAL LIBRARY AUTOMATION APPLICATIONS

Automation in Federal libraries is at many different stages of development, from nonexistent to avant garde. The purpose of Federal libraries varies from providing leisure reading to supporting the most scholarly research. Size of allocation, size of staff, and size of budget vary from library to library, and administrative structures vary from one single enormous library to a branch structure of more than 150 small units. The problems of automation in the Federal libraries are the same as those in other libraries with the addition of a few peculiar circumstances. Federal libraries have strict budget and staffing requirements, for example, and complex procurement and contracting procedures. It is this environment this book addresses.

HOW THE MINICOMPUTER COULD HELP

What is a minicomputer? How is it different from a regular computer? How can minicomputers help meet a library's automation requirements?[74] These are basic questions, but they are not easily answered.

What is a minicomputer? It is a computer — a machine that manipulates symbols in accordance with given rules in a predetermined and self-directed manner. It is made up of a central processing unit, a memory, and input and output devices. It operates under the control of its programs. What then makes a computer a minicomputer? The distinction is one of size — physical size, size of memory, size of words, size of storage capacity, size of repertoire, size of price. The problem is one of drawing the boundary to distinguish between sizes. One must ask, "Compared to what?" Definitions of minicomputers vary according to the range or boundaries set.

Auerbach (1974):[75]

> A minicomputer is a small, stored-program digital computer that can be programmed in an assembly or higher-level language and which has the following attributes:
>
> 1. Sells for less than $25,000 for a minimum, stand-alone configuration comprised of a central processing unit, memory, input/output equipment, and system software.
> 2. Contains a memory of at least 4000 eight-bit words.

---

[73] Long, "Computer Technology," p. 213.

[74] Grosch explained the rationale for selecting a minicomputer over a large-scale computer to develop an integrated library automation system. See: Audrey N. Grosch, "Mini-Computer Systems for Library Management Applications; A New Approach to Bibliographic Processing," in *Computers in Information Data Centers*, eds. Joe Ann Clifton and Duane Helgeson (Montvale, N.J.: AFIPS Press, 1973), pp. 25-33.

[75] *Auerbach on Minicomputers* (New York: Petrocelli Books, 1974), pp. 2-3.

> 3. Performs normal computer functions (inputs, transfers, stores, processes, and outputs data) under stored-program control.
> 4. Is usable in a broad range of applications.

Demitriades (1974):[76]

> What is a minicomputer? In general terms, it is a physically small, relatively inexpensive, highly reliable computer that has a stored program capability and requires little or no environmental control. However, the range in cost and capabilities is very broad. . . There is, however, a "most common package" covering approximately 80 percent of the minicomputers in use today. Its basic unit price range is from $2000 to $10,000; a workable system with memory and peripherals will cost about $20,000. The size of a television set, it will store about 4000 words of 16-bit memory and process at about 5 millionths of a second. Its modular design allows for easy add-on.

Some definitions reflect the varied uses of the minis.

Snyder (1975):[77]

> An acceptable definition of the minicomputer is elusive. . . . Most quoted of the suggested specifics are: a core size of 4,000 to 8,000 words, and a selling price somewhere in the vicinity of $5,000. . . . they're usually referring to a central processor, with 4 to 8K of core memory and possibly a teletype. . . . [However] it's not a functional machine. When you talk about a true mini business system, you have to include a general purpose computer and the normal input and output devices. . . and considerably more core than the 4 to 8K. . . .[The definition must be qualified to] any computer system that is capable of handling typical business applications — billing, payroll, inventory control, etc. — and that can be purchased for a total cost of less than $100,000 or rented for less than $2,000 a month.

Pearson (1975):[78]

> A computer that, with its associated software and peripheral equipment, is priced low enough (from $65,000 or so) to be affordable by a medium-sized library, and with sufficient capability to support most library data processing needs.

HISTORY AND DEVELOPMENT

The minicomputer was developed in the 1960s as an outgrowth of third-generation computers.[79] The first minicomputers were for special applications. That is, each was designed for one purpose only. The early applications were in industrial control; minicomputers were used as low-cost controllers of discrete and continuous processes in automated laboratory or industrial equipment. Almost all of the minis were purchased by manufacturers to be incorporated in their products.

A number of technological advances made minicomputers possible. The main advance was the development of integrated circuits that were extremely small and could be mass-produced economically. The electronics field advanced from mechanical switches and relays to vacuum tubes to transistors and solid-state circuitry, and at each step computers became smaller and faster.[80] The discrete components of transistors, resistors, and capacitors were replaced by an integrated circuit called a small-scale integration (SSI) chip, and then by the large-scale integration (LSI) chip. (Chips are extremely small. They are usually about ⅜ by ¾ inch and about ⅛ inch thick, and each chip can represent thousands of transistors.)

This miniaturization did not lessen the power of minicomputers, which were more powerful than most second-generation large computers. The industrial applications proved minis to be powerful, fast, low in cost, and very reliable.[81] The success of the mini as a controller and monitor led to other uses. Eventually minicomputers broke free of the special application mold, and general-purpose minicomputers were developed.

CHARACTERISTICS

Minicomputers are noted for their flexibility. In fact, Vosatka suggested that "multicomputer" would be a more appropriate name.[82] The other major characteristic of minicomputers is low cost. In 1974, Auerbach reviewed the characteristics of minicomputers that result in their lower cost compared to that of large computers.[83]

1. All have simple, i.e., limited, instruction sets, both in absolute number and the power of instructions provided.
2. All have small memories . . . .
3. Word length is short, which permits design economies in the central processor electronics.
4. All have simple input/output (I/O) control capabilities.
5. Elimination of such features as real-time clocks and parity checking from the basic standard as opposed to optional system is cost inhibitive.

---

[76] Paul B. Demitriades, "Mini Update," *Journal of Systems Management* 25 (December 1974):

[77] James E. Snyder, "Small Computers for Small Business," *Journal of Systems Management* 26 (August 1975): 26, 28.

[78] Karl M. Pearson, Jr., "Minicomputers in the Library," in *Annual Review of Information Science and Technology*, vol. 10, eds. Carlos A. Cuadra and Ann W. Luke (Washington, D.C.: American Society for Information Science, 1975), p. 142.

[79] G. J. Vosatka, "The Minicomputer—Evolution or Revolution," in *Minicomputer Trends and Applications 1973; Symposium Record,* papers presented at the IEEE Symposium held at National Bureau of Standards, Gaithersburg, Maryland, 4 April 1973 (New York: Institute of Electrical and Electronics Engineers, 1973), p. 1.

[80] William Barden, Jr., *How to Buy & Use Minicomputers & Microcomputers* (Indianapolis: Howard W. Sams & Co., 1976), pp. 8–9.

[81] Vosatka, "Minicomputer—Evolution or Revolution," p. 3.

[82] Ibid., p. 1.

[83] *Auerbach on Minicomputers*, p. 3.

6. Slow and relatively unsophisticated peripheral devices are used.
7. Minimal system engineering support is needed.
8. A limited amount of comprehensive system software is supplied free with a system.

These characteristics are disappearing in some machines due to advances that make greater computing power possible for the same or less money. For example, minis are available with more main memory, longer word lengths, larger instruction sets, simpler and more efficient I/O systems, and more system software provided. The distinction between large-scale computers and minicomputers is blurring; system prices are increasing. The smaller large computers and larger minis overlap, and such names as "megamini" and "super-mini" are used by some to describe these machines. The IBM System 3 and the Burroughs 1700 are two common systems in this borderline category.

In general, the philosophy of minicomputers is different from that of large-scale computers. The minicomputer environment is one of closeness and contact. There is a "minicomputer attitude," which is described thus.[84]

> People with minis have time to get to know the computer. People with minis get to know their application on the computer. People with minis become intimate with the computer.

With the use of minicomputers, there is a shift from centralization to distributed computing, from central control to local control. A standalone computer can be dedicated to a single use and placed at the point of use. Communications delays are eliminated and data entry problems minimized. Mini systems are modular, and thus more easily configured to meet specific requirements than are large computers. The technological advances come faster in the minis because the development cycle is abbreviated and adjustments to new technology are easier. This all adds up to an open, responsive atmosphere that makes automation less forboding and more approachable.

DEFINITIONS

In choosing a definition for the word "minicomputer," it seems wise to be inclusive rather than exclusive. The dollar value is important but cannot be definitive due to inflation and Government procurement procedures. The entire system, hardware and software, must be considered in the definition, and the system must be functional. The hardware should not require special environmental conditions. That is, it should be usable at the point of contact, but physical size or dimensions are not necessary to the definition. That leaves the following definition: *A minicomputer is a physically small, relatively inexpensive, general-purpose computer that can operate in a regular environment with as much peripheral and system support as is necessary to meet the requirements of the application.*

Even though minicomputers are only a little more than a decade old, they too have gone through development phases or generations (Figure 16).[85] The "micro" systems are smaller, cheaper, and less sophisticated than minicomputers.

Microcomputers, microprocessors, microchips — these terms often are used interchangeably.[86] A chip with all the CPU functions is called a microprocessor. Frequently it is used with other chips that handle the I/O and memory functions. A microcomputer is a minicomputer built around a microprocessor. Microcomputers are often used as controllers of other devices, and these often are called microcontrollers.

BASIC USES AND APPLICATIONS

Minicomputers have been used in five main areas of application.[87]

1. Industrial process control
2. Peripheral control
3. Data acquisition
4. Communications
5. Computation.

Industrial Process Control

Minicomputers first were used in the industrial area. The minis control processes and machine tools; operate equipment; and test systems, subsystems, and components for quality control or monitoring and maintenance in the field. A mathematical model of the process is used. Minicomputers often are made a part of a system assembly or other machine by the original equipment manufacturer. Minis are used instead of traditional electromechanical control systems because they can handle more complex systems and can be

---

[84] Walter L. Anderson, "Minis Are Beautiful!; Keynote Address," in *Minicomputers: The Applications Explosion*, ed. David E. Debeau, proceedings of the American Institute of Industrial Engineers Conference held in Washington, D.C., 17–19 November 1975 (Los Angeles: Management Education Corporation, 1975), n.p.

[85] Cay Weitzman, "Micros, Minis and Midis; [speech]," in *Minicomputers: The Applications Explosion*, ed. David E. Debeau, proceedings of the American Institute of Industrial Engineers Conference held in Washington, D.C., 17–19 November 1975 (Los Angeles: Management Education Corporation, 1975), p. 676.

[86] Stanley Runyon, "Microprocessors in Test Equipment," in *Microprocessors: New Directions for Designers*, ed. Edward A. Torrero (Rochelle Park, N.J.: Hayden Book Co., 1975), p. 22.

[87] *Auerbach on Minicomputers*, pp. 69–75.

TIME FRAME	MICRO SYSTEMS			MINI SYSTEMS		MIDI SYSTEMS	
LATE 1960'S				MINICOMPUTER SYSTEMS DEC PDP-8 HONEYWELL 516 VARIAN 620I HEWLETT PACKARD 2114			
EARLY 1970'S				MINICOMPUTER SYSTEMS DATA GENERAL NOVA INTERDATA 70 GENERAL AUTOM. SPC-16		MIDI COMPUTER SYSTEMS DEC PDP-11/45 NOVA 840	
MID 1970'S (NOW)	MICRO CHIP	MICRO CARD	MICRO MINI	STD MINI	MICRO PROGRAMMED MINI	MIDI	MEGA MINI
	AMI GEN. INSTR. INTEL 8080 MOTOROLA M6800 NAT'L IMP-16 RAYTHEON SIGNETICS	I/O DEVICES IOP-8 PRO-LOG DEC	MICRO-ONE GA LSI 12/16 PDP-8/A	NOVA II LOCKHEED SUE	VARIAN 73 1-85 HP-2100 MICRODATA 1600	ECLIPSE HP 3000 XEROX 530 PRIME 300 MICRODATA 3200	MODCOMP IV INTERDATA 7/73

FIGURE 16. A decade of minicomputer development

adapted to changes in materials or processors by merely changing their software.

Peripheral Control

Minicomputers are used instead of hard-wired controllers to control peripheral devices such as terminals, data entry systems, or computer input/output devices. The mini can service interrupts; control input/output; and perform sequencing, data transfers, buffering, editing, and formating so that the main CPU's control unit need not be used for these purposes. This allows better CPU efficiency.

Data Acquisitions

Minicomputers are used to acquire data in industrial and military applications, research and development organizations, hospitals, and laboratories. They provide a real-time interface with sensors, counters, test or measuring devices, satellites, or other data-gathering instruments. They accept data from multiple sources at high rates, and then store, log, edit, format, and/or preprocess the data.

Communications

Communications networks use minicomputer communications control processing in three different ways: as front-end processors, as data concentrators, and as message-switching units.

A front-end processor is located close to the host computer and is placed in the data flow between the host and a number of remote data terminals (or other computers). The front-end processor performs data communication control functions such as line control, error checking, code conversion, automatic answering, polling and addressing, and character-to-message assembly and disassembly.[88] The host processor (the mainframe CPU) thus makes more efficient use of processing time and memory requirements. A minicomputer used as a front-end processor is more flexible and economical than a hardwired device and can serve to maintain network operations if the host computer should go down.

A data concentrator is remote from the host computer and serves to coordinate and optimize transmission of data. It accepts messages from many terminals via low-speed lines and transmits them to the host via a single high-speed line. This reduces line costs by more efficient use of the lines. A minicomputer functioning as a data concentrator can be programmed to perform the data communications control functions as a front-end processor; to accommodate interfaces to special terminals; to buffer input/output differences; and to accommodate changes in data rates, formats, codes, communication procedures, and number of terminal devices.[89]

---

[88] *Introduction to Minicomputer Networks*, p. 3-3.

A message-switching unit is an intermediate point in the data flow. An entire message is transmitted to a message-switching unit, where it is stored for a period of time.[90] It is then transmitted to its destination, which is designated by an address in a header field in each message. The unit accepts messages from multiple sources, logs the messages, routes them to multiple output lines, and verifies their transmission. A minicomputer as a message switcher does more than route traffic. It lets a terminal send a single message with multiple addresses, eliminating retransmissions for each address, and it temporarily stores data on disks or magnetic tape, converts codes, edits, logs, and polls and addresses terminals. It saves line costs by allowing messages to be transmitted at top speed and one line to be shared by several terminals.

Computation

Minicomputers are used for computations or problem solving. This is a standalone application in which the mini is used as a large-scale computer — it performs input, manipulation, and output operations while independent of any other computer. It is used in applications that could be served by a large-scale computer but are more efficiently served by the mini. The mode can be batch or online, dedicated or timesharing. Minicomputers have been applied thus in business, education, engineering, and research.

MINICOMPUTERS IN LIBRARIES

Minicomputers have many potential uses in a library. Except for data acquisition, any of the five main areas of application could be found in the library.

Many libraries already have minicomputers — some unknowingly. A Xerox 9000 has a minicomputer as its control unit. Other common library equipment with microprocessors include magnetic-card typewriters, desk calculators, and microfilm readers. Some libraries that are part of large-scale computer systems have a microprocessor-controlled peripheral, i.e., terminal, or an intelligent terminal that performs front-end editing and error control.

Pearson highlighted the early applications of minicomputers in libraries.[91] Minis were first used as data collection units in batch circulation systems. In this application the information for each transaction is recorded by the mini and stored with other transactions to be transmitted in a group to the host computer for batch processing of the files and production of the reports. Minis next were used as remote processors, as well as being used as locally available small files (e.g., bad borrowers files and hold or reserve book files). The major file updates and reports still are handled by the host computer in a batch mode. Ultimately minis were applied to circulation as standalone units to handle transactions in an online environment.

Minicomputers have been applied in similar ways to support acquisitions, technical processing, cataloging, film booking, and selective dissemination of information.

Minis have also been used in larger system configurations as terminals, communication control units, and/or remote computers. The Stanford University BALLOTS system uses a minicomputer as a communications controller and as a programmable terminal, connected to an IBM 360 mainframe or host computer.[92] The University of Chicago Library data management system uses a minicomputer as a remote concentrator (communications controller) to handle the network's 50 terminals.[93] Information Dynamics Corporation's BIBNET system used a dispersed computing approach, with minicomputers functioning as remote processors.[94]

---

[89] Ibid., pp. 3-5—3-7.

[90] Ibid., pp. 3-10—3-12.

[91] Pearson, "Minicomputers in the Library," pp. 142–4.

[92] Wayne Davison, "Minicomputers and Library Automation: The Stanford Experience," in *Applications of Minicomputers to Library and Related Problems*, ed. F. Wilfrid Lancaster, papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974 (Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974), p. 80.

[93] Charles R. Payne, "The University of Chicago Library Data Management System," in *Applications of Minicomputers to Library and Related Problems*, ed. F. Wilfrid Lancaster, papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974 (Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974), pp. 114–6.

The last major type of library application is the minicomputer "turnkey" system.[95] This commercial, on-the-shelf package is complete and ready to use on installation. It includes minicomputers, peripherals, and software necessary to perform specific library functions on a production basis. The Computer Library Services, Inc.'s (CLSI) LIB 100 circulation system has been installed in more than 50 libraries.

SUMMARY

Minicomputers, although not a panacea, can and do have a place in libraries. Their role in library automation will increase. The attributes of minicomputer systems contribute to the library automation environment in a number of ways.

- *Local Control.* Because minicomputers are small and require no special environmental controls, they can be installed in the point of use (that is, the normal library environment). The library staff thus controls the operation of the system, reducing problems of access, long queues, scheduling time, and low priorities.
- *The Mini Attitude.* Because the mini is housed in the library, it is more approachable for the library staff. There is more hands-on use of the system, which encourages the staff to understand and accept the system.
- *Smaller/Simpler System.* As a rule, the minicomputer is a smaller, simpler "machine." It lends itself to applications and environments that previously were considered too small to justify automation.
- *Modular System Development.* Because a mini lends itself to small, discrete applications, a library can build a total system on a modular block-by-block basis. Individual systems can be integrated into a whole library system.
- *Custom Configurations.* Minicomputer hardware and the minicomputer industry are such that a system configuration can be tailored to meet individual system requirements. There are few one-vendor shops in a minicomputer environment. Hybrid, mixed-breed systems are common.
- *Adaptability.* With large-computer systems it is often so costly and time-consuming to change vendors or upgrade equipment or software that systems remain stagnant or become out of date by default. This is not true of minicomputer systems. Changes can be handled much more easily.
- *Low Cost.* The main characteristic of minicomputers, their low cost, allows more libraries to automate. Although they are not cheap, minicomputer systems are relatively inexpensive compared with large-scale computers, and they bring about a greater dollar return and cost/benefit ratio.

No library can afford to ignore the possibility of applying a minicomputer to its operations. This is true for libraries with no experience with automation as well as for those with functioning, full-scale automated systems, for minicomputers do not have to compete with large-scale computer or time-sharing systems. They can supplement or complement such systems instead. Again, their keyword is flexibility.

---

[94] David P. Waite, "The Minicomputer Its Role in a Nationwide Bibliographic and Information Network," in *Applications of Minicomputers to Library and Related Problems*, ed. F. Wilfrid Lancaster, papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974 (Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974), p. 140

[95] Pearson, "Minicomputers in the Library," p. 144

CHAPTER 3

MINICOMPUTERS—DESCRIPTIONS

In general terms, minicomputers are functionally and operationally the same as large-scale computers. There are differences, though, and this chapter describes these divergences from the general description of computers in Chapter One. The differences stem from the facts that (1) minis are smaller and therefore simpler, (2) they are "younger" and have not evolved as far as large-scale computers, and (3) they are usually designed for single applications, whereas large-scale computers are general-purpose systems.

Minicomputers are approachable. Users can see and touch them. For this reason they need not be viewed as "black boxes." Many people become interested in minicomputer systems as a hobby, much as stereo buffs get into audio components and systems. Many systems are put together as hybrids with equipment from various manufacturers. While this can be a strength, allowing the user to have a system that truly meets his needs, it can make selection of a system technically complicated.

This chapter alone will not prepare librarians to assemble their own systems. References to additional sources are given for users who would like to take this approach.[1] The more general goal is to prepare librarians to deal with system requirements, specifications, and selection criteria as members of teams that include data processing experts. Librarians should seek the advice of experts but should not be intimidated into deferring too much to them.

A block diagram of a minicomputer configuration has the same basic components as that of a large-scale computer configuration (Figure 17). Minicomputer software also is basically the same as that of large-scale computers. It includes library subroutines, assemblers, compilers, file management programs, and operating systems as well as the application programs.

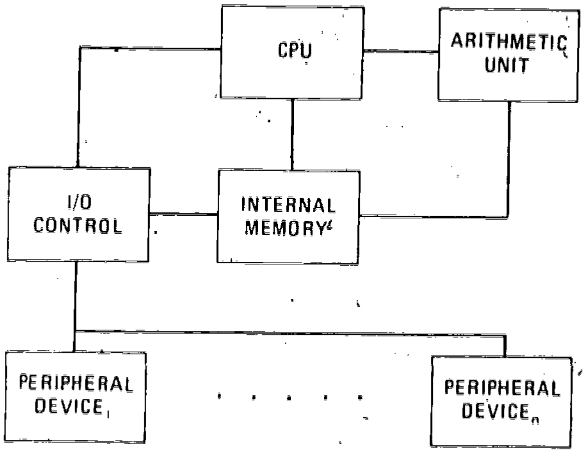


FIGURE 17. Block diagram of a minicomputer configuration

The specific components and configuration are determined by the application. For data communications network applications the main distinction between configurations is in the physical location of the equipment. For example, a minicomputer connected to a host to serve the communications control processing for a number of nodes or terminals can have basically the same components whether it is a front-end processor or a remote concentrator (Figure 18).

---

[1] Cay Weitzman has written a basic text on minicomputers His Preface statement defines the scope which is of the same philosophy as this book: "To insure success the user must not only take the time to determine what the system must accomplish *for him* but also have a fundamental working knowledge of the qualitative and quantitative design approaches to minicomputer systems evaluation, selection and interfacing, and hardware, software, and firmware integration, Quick, poorly supported decisions coupled with lack of knowledge of the various details of minicomputer systems analysis and design will in all probability result in an unsatisfactory selection or solution to the problem. My purpose is, therefore, to give the system designer and/or user an overview of latest minicomputer hardware and software technology, tools, procedures, and approaches used in evaluating and designing minicomputer systems as well as guidelines as to how to implement,maintain, and support these systems" (pp. xi–xii); *Minicomputer Systems, Structure, Implementation, and Application* (EnglewoodCliffs, N.J.: Prentice-Hall, 1974). Eckhouse's book is on computer organization and assembly language programming. It uses the PDP-11 asthe real machine the examples are based on; Richard H. Eckhouse, Jr., *Minicomputer Systems; Organization and Programming (PDP-11)* (EnglewoodCliffs, N.J.: Prentice-Hall, 1975). Barna and Porat have written anintroduction to microcomputers; Arpad Barna and Dan I. Porat, *Introduction to Microcomputers and Microprocessors* (New York: John Wiley& Sons, 1976). The *Microcomputer Dictionary and Guide* is a good reference book for micros and minis; Charles J. Sippl and David A. Kidd (Champaign, Ill.: Matrix Publishers, 1975).

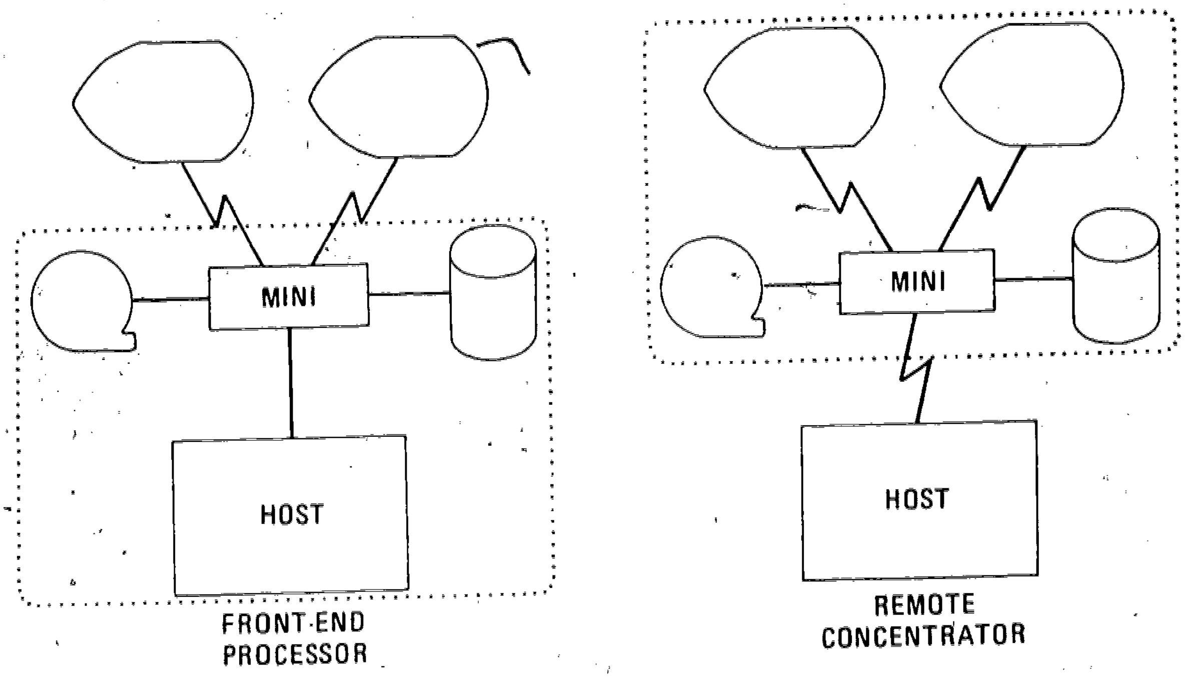


FIGURE 18. Configuration of a front-end processor and a remote concentrator

Front-end minis are located at the same site as the host computer; remote concentrator minis are located at the remote site and are connected to the host by telecommunications lines.[2]

As far as library applications are concerned, the configuration varies according to input requirements, output needs, storage demands, etc. The specific library applications and typical configurations will be reviewed in later sections. In this section the general elements of the hardware and the capabilities of the software will be described, and their roles explained.

HARDWARE

MAINFRAME

CPU

The minicomputer mainframe (CPU plus main memory) determines the power of the system. The main characteristics considered in describing and evaluating the structure, or architecture, of the CPU include the following.[3]

- Instruction set
- Addressing capabilities
- Instruction execution speed
- Number, size, and arrangement of accumulators and/or index registers
- Number of interrupts and methods of handling them
- Optional features

More technical definitions of each of these elements can be found in a number of texts. The impact of any one of these elements can be overcome by augmenting the system with additional hardware or by software design; none can be taken as definitive or pivotal.

Instruction Set

The instruction set is composed of the basic machine language or commands that represent the simplest tasks performed.[4] They include logic statements such as AND and OR; load/store statements such as MOVE DATA and LOAD ACCUMULATOR; arithmetic statements such as ADD, SUBTRACT, TWO'S COMPLEMENT, and SHIFT; and control statements such as COMPARE, SKIP IF EQUAL, UNCONDITIONAL JUMP, and TEST AND SKIP

---

[2] James F. Corey, "Configurations and Software: A Tutorial," in *Applications of Minicomputers to Library and Related Problems*, ed. F. WilfridLancaster, papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974 (Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974), p. 17.

[3] *Auerbach on Minicomputers* (New York: Petrocelli Books, 1974), p. 17.

[4] Ibid., pp. 26–28.

IF SET OR RESET. The simpler CPUs have very few instructions in their sets. In terms of the application, the number of instructions provided is less important than the kind of instructions. In any case a programmer can write routines to accomplish these tasks. It simply takes more development time and makes the program larger and slower.[5]

Addressing Capabilities

The addressing capabilities of the CPU are determined by the length of the instruction word, the processor logic, and the memory size.[6] The word length is the amount of data that can be stored in one memory location; it ranges from 8 to 24 bits. The means of identifying the data in each memory location and making it available for use is called the addressing scheme. The Sams book presents a simple explanation of addressing:[7]

> Memory cells are very similar to post-office boxes. Post-office boxes are numbered in sequence on the outside of the box. We will number our memory cells from 0 to 63, a total of 64 memory cells, each one holding 16 bits of data. When we talk about the contents of memory cell 17, we may visualize a post-office clerk going over to box number 17 and withdrawing a post card with a 16-bit value printed on it. . . The 64 memory cells contain either data values or instructions (or nothing meaningful if a cell is not used). The data values might be used by add or subtract instructions, or for comparisons. Cells might also be set aside, or reserved, for the results of operations. If there were no pieces of data or reserved cells in memory, there would be room for 64 instructions. If there were instructions in locations 0 through 23, there obviously could be no data in these locations at the same time.

The instruction word becomes the key. It must, as explained in Chapter One, specify (1) the operation to be performed and (2) where in the memory is the data to be operated on, where it is to be located after being operated on, and where in the memory is the next instruction. That means an instruction word of 16 bits is divided into a two-part format.[8] The first 4 to 6 bits are assigned as the operation code and the remaining 10 to 12 bits are for addressing.

Most minicomputers use one address instructions which means that one instruction can address only one memory location.[9] Going back to the post office analogy, each memory location has a unique number as its address. With 12 bits representing a base two numbering scheme, only 4096 unique numbers can be specified, or, only 4096 memory locations can be addressed. When memory sizes are greater than 4096 locations, means of addressing memory other than direct addressing must be used. Different manufacturers use different methods. The most common are direct or absolute, immediate absolute, paged, relative, indirect, and indexed addressing. "For these techniques, part of the address field (for example, 3 bits) selects the addressing technique or mode and the rest of the field (9 bits) operates as an address, an address displacement, or an operand."[10] Some techniques use special registers, counters, or stacks to extend the number of locations that can be addressed. The technique used affects the memory size that can be used, the speed of processing (number of cycles required to support an operation), and the difficulty of programming the system.

Instruction Execution Speed

The instruction execution speed depends on memory cycle time as well as on the CPU's internal logic (the way the cycles are used to perform the operations). As a basis for comparing CPUs, the speed is often measured in terms of performing a standard operation, such as the add time as well as the cycle time. Cycle time can be defined as "the time to read (and restore) a single word in memory"[11] or the "minimum time interval, in microseconds, between two successive accesses to a particular storage location."[12] Add time is variously defined, but in general it represents the time required to perform an add operation (C = A + B), including the time used in accessing both operands from memory and storing the results in working storage.[13] Even these times are not always directly comparable, due to the way the manufacturer performed the test. Other factors that affect speed in complicated systems are the load on the system and the use of such techniques as time-slicing.

Register Configuration

The number, size, and arrangement of accumulators and/or index registers really determine the architecture of the CPU. There are two basic types of CPUs, special-register processors and general-register processors.[14] In special-register processors "one of the operands specified by the instruction resides in a main memory location and the other resides in either the accumulator or some other special-purpose register." For special functions there are special registers, such as accumulators; extend registers, which may be logically linked to accumulators; index registers, which are used in addressing; and others for program linkage, etc.

---

[5] Ibid., p. 17.

[6] Ibid.

[7] William Barden, Jr., *How to Buy & Use Minicomputers & Microcomputers* (Indianapolis, Ind.: Howard W. Sams & Co., 1976), pp. 23–24.

[8] *Auerbach on Minicomputers*, p. 26.

[9] Ibid., p. 28.

[10] Ibid.

[11] GML, *Minicomputer Review* [illegible] (Lexington, Mass.: GML [illegible]), p. [illegible]

[12] *Auerbach on Minicomputers*, p. 90.

[13] Ibid., p. 90 and Dennis Hollingworth, *Minicomputers: A Review of Current Technology, Systems, and Applications* (Santa Monica, Calif.: Rand Corporation, 1973), p. 13.

General-register processors do not define the functions of the registers but allow them to be used as accumulators, stack pointers, or index registers depending on program requirements. This gives greater flexibility. Most processors also have nonprogrammable registers that operate as buffers to hold a word temporarily for the processor hardware (e.g., program counters).[15] A variation has been introduced with a mechanism called a stack.[16] The stack is an ordered collection of memory locations or hardware registers with a top or first element, a second element, and so forth, in which only the first element can be accessed. A "pushing" operation loads the elements one at a time in sequence, and a "popping operation" takes the top element off and moves every element up one position. The number and arrangement of the registers affect programming ease and flexibility as well as execution efficiency.[17]

Interrupts

The number of interrupts and methods of handling them allow the processor to interact with the entire computer system. "An interrupt is a signal that causes the processor to suspend execution of instructions in the current program and to branch to a set of instructions that deal with the interrupt condition. When the interrupt has been taken care of, the processor continues to execute the suspended program."[18]

There are two types of interrupts, internal and external. Internal interrupts, or traps, deal mainly with abnormal, error, or occasional conditions (e.g., illegal address, power failure, memory parity error, and overflow from fixed-point arithmetic operation). External interrupts deal with devices external to the processor, such as the peripheral devices. Because the processor speed is very great and the peripheral devices are by comparison very slow, the processor performs other functions until the peripheral sends an interrupt signal to say the input or a character is ready to be transferred. Once the transfer is completed, the processor returns to the program it was executing when the interrupt occurred. Generally the external interrupt expresses the cause or condition that triggered the interrupt, such as error, malfunction, character ready for transfer, operation finished, end of block, etc. The complexity of the interrupt system, called the vectoring mechanism, depends on the number of lines, the number and nature of the devices, and any priority structure used.[19]

Optional Features

Optional features available from many manufacturers may be appropriate for specific applications. The types of features offered include power fail/safe, memory protection, memory parity, floating-point arithmetic, decimal arithmetic, and byte manipulation.[20]

Main Memory

The main memory of a minicomputer is described by its word size and capacity. The word length is the amount of data that can be stored in one memory location, and the capacity is the total number of memory locations or words available. Common capacity sizes are 4K, 8K, 16K, 32K, 64K, 128K, and so on, where K is nominally 1000, but in actual measure each K represents 1024. The size can be thought of in terms of word capacity or character capacity, and a choice can be made based on the best, most efficient combination (e.g., 8K 8-bit words or 4K 16-bit words). Table 10 shows some representative combinations.

TABLE 10—Common Sizes of Main Memory Units

Word Length	Word Capacity	Character Capacity
8 bits	4 K	32,768
8 bits	8 K	65,536
8 bits	16 K	131,072
16 bits	4 K	65,536
16 bits	8 K	131,072
16 bits	16 K	262,144
16 bits	32 K	524,288

---

[14] Hollingworth, *Minicomputers*, pp. 7-8

[15] *Auerbach on Minicomputers*, p. 25

[16] Eckhouse, *Minicomputer Systems*, p. 13

[17] *Auerbach on Minicomputers*, p. 17

[18] Ibid., p. 36

[19] Hollingworth, *Minicomputers*, pp. 20-22

[20] *Auerbach on Minicomputers*, pp. 18-19

The available types of memory are ferrous-core and semiconductor.[21] Ferrous-core memory is the kind described in Chapter One. It is nonvolatile, which means that if power is lost the contents remain stable. The average speed of core memory in a minicomputer ranges from 0.8 to 1.5 microseconds. A semiconductor memory is a solid-state, integrated circuit module usually called a chip. The density of the chip is expressed as LSI (large-scale integration). The material of which the chip is made constitutes an additional means of classification (e.g., bipolar semiconductor memory and metal oxide semiconductor, or MOS, memory). Semiconductor memory is faster than core (MOS averages 0.45 microseconds and bipolar averages 0.3 microseconds), but it is volatile — the contents of memory are lost in the event of a power failure. Most memory has both read and write capability. A special kind of chip is used in some equipment that has read-only capability.

The ROM (read-only memory) is a memory containing permanently available, frequently used programs and/or data. It is designed and sequenced as it is manufactured. It cannot be changed and is relatively slow. A ROM is nonvolatile in that the contents will remain even if the power source is interrupted. A new modification now available is a PROM chip, or a programmable ROM. The programmer can decide on the subroutines, which are entered electronically at the beginning but are not easily changed and are not under program control as other memory is. ROMs and PROMs have figured heavily in miniature systems, controllers, and microcomputers.

Input/Output Control

Input/output control transfers information between the processor of memory and I/O devices in response to signals from the control unit and the external peripheral devices.[22] An I/O system is made up of (1) a controller, which controls such features of the peripheral device itself as electronics for print-head selection, paper motion, and print timing; (2) one or several buses (the actual lines that connect the devices or units); and (3) an interface (the logic circuitry that controls the larger activity required to synchronize data transfer by performing such functions as level conversion, command decoding, multiplexing, and data request logic).[23]

The actual data are transferred by one of two main methods: programmed I/O (PIO) or direct memory access (DMA).[24] All minicomputers have PIO, which is a software technique. Input/output instructions transfer the data and are device-dependent. That is, the instructions are peculiar to each kind of device. PIO is slow and is appropriate for data transfers with slow-speed devices such as paper tape equipment, hard-copy terminals, and low-speed line printers.[25]

DMA, a hardware feature, is not on all minis.[26] It allows data to be transferred at high speed between the device and the memory itself. The DMA interface is sometimes called a channel. This channel contains (1) a memory address register, (2) a word counter to keep track of the number of transfers performed, and (3) logic for gaining access to the memory and providing the necessary timing and control signals. The DMA is used for high-speed devices such as magnetic tapes, disks, and drums. A multiplexer channel is used to handle DMA for several devices. It is somewhat slower than DMA, but it allows a number of data operations to occur simultaneously.[27] The interrupt system described above is also an integral part of the I/O control.

PERIPHERALS

The peripherals are made up of anything attached to the mainframe. Generally classed as input/output devices, they function as mass storage or man/machine interface devices; some can serve both functions.[28]

Mass Storage Devices

Mass storage serves as an extension of computer main memory. The media range from punch card and paper tape to magnetic tape and disks. Each medium has different speed, capacity, and storage characteristics. Selecting the proper device for mass (file) storage is critical to the efficiency of a data processing application.

---

[21] Hollingworth, *Minicomputers*, pp. 16–18.

[22] *Auerbach on Minicomputers*, p. 16.

[23] A. A. J. Hoffman, Robert L. French, and Guy M. Lang, "Minicomputer Interfaces: Know More, Save More," *IEEE Spectrum* 11 (February 1974): 64.

[24] *Auerbach on Minicomputers*, p. 42.

[25] Hollingworth, *Minicomputers*, p. 19.

[26] *Auerbach on Minicomputers*, p. 43.

[27] Hollingworth, *Minicomputers*, p. 20.

[28] There are entire books on peripherals, for example, Ivan Flores, *Peripheral Devices* (Englewood Cliffs, N.J.: Prentice-Hall, 1973). Periodicals feature surveys of peripherals; for example, "Peripherals Make the Mini, *Modern Data*'s Annual Survey of Plug-Compatible Miniperipherals," *Modern Data* 8 (December 1975): 34–43.

Punch Cards

The punch cards contain data represented by patterns of holes punched in the 80 columns of each card.[29] Each column can represent a character. The cards are stored in boxes and can be handled manually — humans can read the interpreted headers and can remove or add cards as desired. Punch cards are inexpensive, but it takes many cards to hold a given number of characters (i.e., records) and therefore a large amount of physical storage space and much manual handling are needed. Data on cards are entered into the system by means of a card reader, which reads the holes and converts them into bit patterns. Card readers are low-speed devices that read 100 to 600 cards per minute. Cards can be prepared by a keypunch machine offline. Computer-driven card punches can be used to punch cards as computer output. Card readers for minis are common and inexpensive. Card punches for minis are rare and expensive.

Paper Tape

A continuous ribbon of paper called a paper tape can be punched like cards with holes representing data.[30] Unique hole patterns in the paper tape represent the symbols. Six-, seven-, and eight-hole paper tape devices are in common use. The number of holes available determines the number of unique symbols that may be coded onto the paper tape. Most paper tape devices provide for extension of code meaning through a technique called shifting. Shifting requires that a special hole pattern be used to indicate the use of upper-case or lower-case shift. A paper tape printing device will recognize the shift codes and take appropriate action. Software in the computer also must recognize shift codes because the symbols can have one of two meanings according to the shift.

The tapes are continuous, unlike cards which are limited to 80 columns. The tape is smaller, more compact, and requires less manual handling than cards. The problem is in editing or correcting errors on the tape. The paper tape readers work similarly to punch card readers but operate at higher speeds, although they are still classed as low-speed devices. Paper tapes commonly are used with minis for storing the simple, basic programs used to start up and operate the CPU.

Cassette or Cartridge Tape

Magnetic tape cassettes that look like common audio cassettes are used on minis.[31] They are recorded and read sequentially, like paper tape. They are compact and relatively inexpensive. Typical recording capacity for a cassette ranges from 110,000 to 400,000 characters. The cassette tape drive reads and records on the cassette. The speed of data transfer on a cassette drive varies from 500 to 1600 characters per second. Once the tape is mounted, little human interface is required. The drive that advances or rewinds the tape is program controlled. Some systems use two tape cassettes, one for the program and one for the data or file.

Cartridge tapes are similar to home audio cartridges but operate in binary as do computer cassette tapes.[32] They generally have multiple tracks and greater length than cassette tapes, which allow greater storage capacity. A cartridge drive is required to record and read the cartridges.

Industry-Standard Magnetic Tape

The common computer tape is ½-inch-wide magnetic tape wound on reels.[33] The information is formated so that each bit of a character fits into a theoretical "track." A single position on a nine-track tape is typically divided into eight data tracks (using an eight-bit code) and a ninth track used for recording a parity bit (a technique used by some manufacturers for error checking). As with all tape, information is recorded and read sequentially. Reels are commonly wound in lengths of 200, 600, 1200, and 2400 feet. The tape drives record data in densities of 556, 800, or 1600 bits per inch (BPI), and recording capacities thus range from 6672 to 9600 characters per foot.

A reel of magnetic tape is continuous and the density of a tape is constant, but determining capacity is not always a matter of multiplying characters per foot times the number of feet. The data on the tape is recorded in blocks, separated on each side by interrecord gaps (IRGs), which help in searching the tape to locate particular records. The number of the IRGs reduces the character capacity of each tape. The reels of tape can be handled easily and stored manually, and they are relatively inexpensive.[34] Because tape is a sequential or serial device, it is most efficient as a storage medium for data that are to be used in a serial fashion. Inserting new data on a tape requires that a new tape be produced. To locate a record the tape drive winds or rewinds the tape until the data are found. This process is time-consuming and inefficient if data are to be used in a random fashion. The tape drives are considered high-speed devices (the tape is read at up to 150 inches per second), but they are still slower than the internal processing speed. Tape drives can be used singly, but two or three tape drives are more common if the data manipulation requires producing new file tapes or sorting data.

---

[29] Robert M. Hayes and Joseph Becker, *Handbook of Data Processing for Libraries*, 2d ed. (Los Angeles: Melville Publishing Co., 1974), pp. 283-9. See also: David C. Bailey, "Requiem for the Punched Card?" *Modern Data* 8 (October 1975): 66-70.

[30] Hayes and Becker, *Handbook of Data Processing for Libraries*, pp. 289-93. See also Horace Lyndes, "Wake for Perf Tape?" *Modern Data* 8 (October 1975): 71-72.

[31] Barden, *Minicomputers & Microcomputers*, pp. 67-68. See also Stephen A. Caswell, ed., "Cassette Drives and Systems," *Modern Data* 8 (October 1975): 58-63.

[32] Barden, *Minicomputers & Microcomputers*, pp. 67-68.

[33] Ibid., pp. 65-67.

Diskette, or Floppy Disk

Disks are also magnetic storage devices, but rather than being serial and linear like tape, they use two-dimensional surfaces, usually like circular plates.[35] Data are stored on both surfaces, top and bottom, and each surface is divided into tracks, which are concentric circles around the axis.[36] The two-dimensional shape is reflected in the way the read/write heads of the drive operate; they can move or be positioned over any part of the surface of the plate (that is, over any track), which allows direct or random, as opposed to sequential, access. This freedom allows greater speed and flexibility in accessing stored data.

The simplest disk medium is the diskette, also called the floppy disk. This single, flexible (hence "floppy") plate is made of a Mylar material. A paper or plastic envelope or jacket houses the disk and is never removed, even when the disk is mounted on the disk drive. The disk is rotated inside its jacket, and a slot in the jacket exposes the tracks to the recording head. Floppy disk drives vary in sophistication and cost from low-speed, low-capacity units to high-speed, high-capacity units with a top capacity of 350,000 bits and a transfer rate of 1.2 million bits per second. Disk drive units are commonly sold for one, two, three, or four floppy disks. The choice depends on the application.

Fixed-head Disk

These disks are rigid, circular plates and are larger than diskettes. The read/write heads are designed so that each track has its own head. The program determines which track to read. Disks are the fastest mass storage medium, but not the largest in storage capacity, and are more expensive on a cost per character basis. Fixed-head disks are mounted individually, like floppy disks, and are used in multiples if the application warrants.

Movable-head Disk

A movable-head disk is so called because one read/write head moves across the disk to the track desired.[37] Then the data are read or written. The time it takes to move the head makes this type slower than fixed-head disks, but once the head is located over the track the data transfer rate is the same. Movable-head disks are generally packaged with multiple disks per unit.[38] A disk cartridge usually has two or four surfaces, and disk packs have 10 or 12 surfaces. Each surface has its own read/write head mounted on an arm assembly. This arm moves as a unit when a track is selected, which means that if one head is over track two on one surface, all arms are over track two on all surfaces. This amount of data — one track of data from each disk surface — is called a cylinder. The storage capacity of each unit can vary up to more than 20 million words.[39]

Man/Machine Interface Devices

Man/machine interface devices are so named to avoid confusion with the general class of peripherals called input/output devices. In computer jargon, an I/O device is anything that introduces digitized data into the mainframe (CPU and main memory) for processing[40] (1) from the mass storage device that reintroduces the data, (2) from another system via a telecommunications line, or (3) from a terminal device with a keyboard to allow a human to introduce data into the system. To the layman, input is not defined so broadly; it means converting source or human data into machine-readable form and transmitting it to the computer for storing and/or processing. There are several ways to categorize man/machine interface devices: low-speed or high-speed; hardcopy or softcopy; offline or online; batch or conversational; input only, output only, or a combination.

---

[34] *Auerbach on Minicomputers*, p. 52

[35] Hollingworth, *Minicomputers*, p. 24. See also, Dan M. Bowers, ed., "Floppy Disk Drives and Systems; Part 1. Historical Perspective," *Mini-Micro Systems* 10 (February 1977): 36–51.

[36] Barden, *Minicomputers & Microcomputers*, pp. 68–69

[37] Hayes and Becker, *Handbook of Data Processing for Libraries*, pp. 343–6. See also Dan M. Bowers, ed., "Removable Disk Storage; Where It's Come From and Where It's Going," *Modern Data* 9 (January 1976): 36–38, and Barbara A. Reynolds, ed., "Removable Disk Cartridge Drives," *Modern Data* 9 (January 1976): 38–43.

[38] *Auerbach on Minicomputers*, p. 53.

[39] Hollingworth, *Minicomputers*, p. 25

[40] Hayes and Becker, *Handbook of Data Processing for Libraries*, p. 280

Punch Cards

Punch cards as a man/machine interface function as described in the mass storage discussion. Keypunch machines are used to convert the source data into card format. The machines are slow and costly in staff time. An average speed is 6,000 keystrokes per hour. One article states that no new system should be designed around keypunch machines.[41] The use of punch cards for transaction records or reentry records still does have merit, however. The card reader usually is used as a remote batch terminal and, as described earlier, is a low-speed device. It can be used for input only. The card punch must be used for punch card output. This use would be limited for mass storage or reentry documents, such as those used in serial and circulation systems.

Paper Tape

For converting source data to machine-readable form, paper tape punches like the Flexowriter are just as outdated as the keypunches. (Paper tape is sometimes combined with Teletype units; this will be discussed later.) Paper tape does not have the transaction or reentry capabilities of punch cards and is not human-readable. The readers are faster than many card readers (2,000 characters per second for the top of the line) but are still considered low-speed devices. Paper tape can handle upper- and lower-case characters but must use shift codes, which slow down conversion and make software more complicated. Paper tape punches driven by the computer are suited only for mass storage or for interface with other systems.

Key-to-Magnetic-Medium Stations

The two main types of key-to-magnetic-medium stations are key-to-tape and key-to-disk stations.[42] There are now key-to-cassette-tape units and key-to-floppy-disk units. These stations can be used as standalone, independent units or in a multistation configuration with a shared processor. They operate offline from the host computer. The input data are gathered and then read into the host system in a batch. These stations can be used for key-to-mass-storage-file purposes if their output is compatible with the system mass storage device(s). Multistation units feed the input into a single storage medium (e.g., tape or disk).

Because input generally must be sequenced before or as it is entered into the host system, multistation key-to-tape units are not as efficient (it is difficult to control the input from the various stations to achieve the proper sequence) as multistation key-to-disk units. Originally called keypunch replacement equipment, key-to-magnetic-medium stations have many more capabilities than keypunches. They perform such functions as error deletion, editing, formating, and merging new and old data. When the shared processor in a multistation key-to-disk unit is a microcomputer, more functions can be performed, especially in the area of arranging data into formats: positioning, generation of blank or skip fields, left zero filling, duplication, and code conversion.

Transaction Recorders

A number of devices are used to capture data at the source or point of contact.[43] They make use of already prepared material, although not all can be called reentry records. A light pen that reads bar-encoded ("zebra") labels is one example. A unique number is assigned to each item (e.g., order, book, or movie projector) and is recorded in a machine-readable file that is part of the data base. The numbers are expressed in unique patterns that are printed on labels affixed to the items. To record a transaction, a light pen is passed over the label and the pattern is translated into electrical impulses, which are matched against a reference table. The number is then transmitted to the mainframe for processing. One available optical-character-recognition (OCR) wand allows online OCR data input from source documents (or labels) printed in special optical-character-recognition type.

Some data collection stations read reentry records and combine the data to create transaction records. One type is the badge/card reader used in circulation systems. The patron's badge is prepunched with a user's number, and the book information on an already prepared punch card is housed in the book. The data from these two source documents are read and transmitted along with any preset or manually keyed data. Several models have different capabilities and levels of sophistication. IBM produces a number of these units, including the 1031, 2790, and 5230.

---

[41] Dan M. Bowers, ed., "Small-Scale Computing: It's Time to Get With the Future, Part 2 Data Entry," *Modern Data* 8 (June 1975): 44.

[42] Hayes and Becker, *Handbook of Data Processing for Libraries*, p. 280. See also Bowers, "Small-Scale Computing; Part 2," pp. 43–49, and Malcolm L. Stiefel, ed., "Source Data Automation; Part 2," *Mini-Micro Systems* 9 (June 1976): 38–43.

[43] For a discussion of source data entry, see Stiefel, "Source Data Automation, Part 2," pp. 38–43, and Bowers, "Small Scale Computing; Part 2," pp. 43–49.

Printers

Printers are one-way (output only) devices.[44] They provide human-readable hardcopy information, often in batches. There are many variations of techniques to achieve printed output; each has its particular advantage in cost or quality. There are also many variations in the fonts of the printers. Not only is the design of characters quite different, but the selection of symbols available also varies. For example, some printers cannot display lower-case data and most cannot display the diacritics required in library applications.

There are two basic types of printers: impact (where ink or carbon is caused by pressure to adhere to paper) and nonimpact (where electrical or electrostatic mechanisms are used).[45] The simplest impact printer is the serial or character printer which is basically a computer-operated typewriter. The characters are printed one at a time, and these printers are very slow (10 to 180 characters per second).

Line printers are so called because they print an entire line at a time instead of a single character. This is accomplished by using multiple print "heads" or elements. A drum, or print wheel, printer is composed of as many print wheels as there are possible character positions in a line.[46] For example, if 120 characters per line were possible there would be 120 rotary print wheels, each with all the possible characters, each capable of rotating independently, and each with its own hammer mechanism. At the time of printing, all 120 print wheels would be positioned to represent the data to be printed in the line. These devices print about 150 lines per minute. Because of the inherently limited dimensions of a drum wheel, these printers are usually limited to upper-case fonts only. Further, the print is generally of lower quality due to poor alinement (the lines usually wave).

Chain, or train, printers generally provide the best print quality at the highest speed.[47] In this type of printer, the print mechanism has a continuous chain of characters which rotates between the ribbon and hammering mechanism. The number of hammering positions determines the number of print columns that may be displayed — most commonly there are 132 characters per line. As a character passes in front of the hammer for the column in which it is to appear, the hammer is activated and the character is pressed against the ribbon, leaving the impression of the character on the paper.

Because chain printers have separate hammers for each column of print, it is possible to increase the speed of print by increasing the availability of characters on the chain. That is, if the chain contains the complete set of symbols a single time, each hammer position potentially could require a complete chain revolution before the character it requires would appear before it. By increasing the chain to two complete sets of symbols, the maximum number of potential rotations is reduced by half.

There are limits on the physical length of the chain which limit the number of complete sets that can be placed on the train. Also, the number of symbols to be represented affects the number of character sets and therefore the speed. Printers with limited fonts are faster than those with more extensive fonts. Typical printing speeds for minicomputer chain printers are 100, 200, 300, and 400 lines per minute. Speeds in excess of 1,100 lines per minute generally are limited to large-scale computers.

Impact matrix printers use rows and columns of wires, the tips of which compose a matrix of dots, to construct the print images. Each character is represented as a specific arrangement of dots, produced by extending the appropriate wires in the matrix against an inked fabric ribbon to print the characters on paper. Common units have 120 characters per line and print at a rate of 500 or 1,000 lines per minute. Nonimpact dot matrix printers commonly use a heat-sensitive print head. A matrix of dots composes the characters. The dots are produced by electrically charging the wire tips, which darkens a special heat-sensitive paper, leaving the outline of the symbol.

Interactive Man/Machine Interface Devices

Interactive man/machine interface devices are a class in themselves in that they must meet certain requirements. An interactive terminal must handle means for inputting information and displaying information back from the mainframe almost immediately. To be interactive means to be under direct control of the CPU. To be online means that the transaction is going into the system and being processed at that time (as opposed to batches of transactions gathered and entered later). It often means being in a time-sharing environment with multiple users.

---

[44] For information on printing and printers, see Irving L. Wieselman, "Printer Technology and Its Future; A Printer Primer," *Modern Data* 8 (November 1975): 33–42; Dan M. Bowers, ed., ["Computer Printers] Manufacturers' Data," *Modern Data* 8 (November 1975): 43–45; Dan M. Bowers, ed., "Printers and Teleprinters," *Mini-Micro Systems* 10 (January 1977): 30–53.

[45] Hayes and Becker, *Handbook of Data Processing for Libraries*, p. 318.

[46] International Business Machines, *Student Text Introduction to IBM Data Processing Systems*, 2d ed. (White Plains, N.Y.: International Business Machines Corp., Technical Publications Department, 1968), p. 57.

[47] Ibid., p. 58.

To be conversational means that a dialog is being executed between the man and the computer programs — questions are asked and answered; prompts are given to aid and control the input; errors are detected and corrections accommodated on the spot. As a group, interactive terminals are not suited for high-volume capability — for either input or output. An individual, one-transaction-at-a-time mode is the most common and most efficient.[48]

Keyboard/Printer Terminals. The most common interactive terminal, and indeed the most common peripheral used on minicomputers, is the Teletype.[49] The Teletype unit is made up of a keyboard and a typewriterlike printer.[50] The keys print in response to the manual keyboard or on signal from the computer. The Teletype is a low-speed device that performs serial data transfer (one character at a time) with full or half duplex transmission. The speed is 110 BAUD for the transmission or 10 characters per second for printing. The input rate is limited by the speed of the keyboard operator, which averages about four characters per second. The Teletype controller converts the serial data into parallel for transfer to the CPU and back from parallel to serial to transfer data from the CPU to the Teletype.

Teletype units come with various options. Some are equipped with punch tape reader/punches, allowing buffer-type input, which can increase the rate of input and allow offline data preparation. Some are equipped with faster printers, such as ball-type Selectric printers.

There are other manufacturers of keyboard/printer terminals, and their units vary in options, functions, speed, noise, and cost. Some are portable and are equipped with acoustical couplers to allow telephones to be used to connect to the mainframe. Some use cassette tapes for even more sophisticated buffering; this type of terminal allows a permanent hardcopy record of the dialog with the system as well as any formated output product, such as a printed report, table, stencil, etc.

Keyboard/Display Terminals. The display for these terminals is "softcopy," or images that appear on a cathode-ray tube (CRT) screen.[51] The principle is similar to that of a television set where electronic impulses are painted on a screen by a "gun." There are two types: alphanumeric and graphic. The alphanumeric type uses a matrix on a mask to format the images on the screen. Only a predetermined character set can be displayed; some CRTs can display uppercase letters only, some can display upper- and lowercase letters and a few special characters, and only a few can handle diacritics.

In general, CRTs are more flexible, faster, and quieter than printers. CRTs have a great range of capabilities and sophistication. The simplest are nothing more than Teletype replacements. They are character-oriented, receiving and transmitting one character at a time. In fact, these CRTs plug into the TTY control board, and the mainframe is unaware that the device is not a Teletype. Data are displayed faster on the CRT than if they were printed on the TTY, because the print mechanism does not slow it down.

The various features or options a CRT can have include the ability to "buffer" characters into blocks for faster, more efficient transmission and the ability to display an entire page at once instead of using the "scroll" method, which displays one line at a time from the bottom of the screen up (pushing the top line off when the screen is full and a new line is displayed). A CRT may have a cursor, or special symbol, that may appear as an underline, a reversed image character, or other symbol superimposed over a line position that indicates the current operative position on the line and can be spaced forward or backward to position the next character to be input. Another CRT feature is a directional-controlled cursor combined with an internal memory to perform edit routines such as

1. Character deletion (the ability to delete one or more characters from the CRT memory)
2. Character insertion (the ability to insert one or more characters in the CRT memory)

---

[48] Hayes and Becker, *Handbook of Data Processing for Libraries*, p. 306. The Intelligent Person's Guide to Choosing a Terminal for

[49] For a description of and discusion on interactive terminals for library use, see Mark S. Radwin, "Choosing a Terminal, Part 2," *Online* 1 (April 1977): 61-73. See also: Bowers, Online Interactive Use; Part 1," *Online* 1 (January 1977): 11, and Mark S. Radwin, "Printers and Teleprinters," pp. 30-53

[50] Barden, *Minicomputers & Microcomputers*, pp. 63-64. "Alphanumeric Display Terminals; Part 1. The CRT

[51] Hayes and Becker, *Handbook of Data Processing for Libraries*, pp. 309-15. See also: Barbara A. Reynolds, ed., "Alphanumeric Display Terminals; Part 2. Market and Technology -- Where They Stand Now," *Modern Data* 9 (February 1976): 44-51, Barbara A. Reynolds, ed., Who's Who in CRTs and Where the Market Is Going," *Modern Data* 9 (March 1976): 44-51, Radwin, "Choosing a Terminal, Part 1," p. 11, Radwin, "Choosing a Terminal; Part 2," pp. 61-73.

3. Character substitution (the ability to substitute characters in the CRT memory)

or to move to any location within the screen in a format mode to enter data, as with a form to "fill out" for data input. A CRT may also have program-controlled use of visual effects such as reverse video, blinking, and foreground-background contrast (high and low intensity). CRTs vary in number of characters per line and number of lines per screen. Some CRTs have no control functions at all and are controlled completely by the CPU. Some have some control functions that are hardwired (for example, function keys). Some have chips (ROMs or PROMs) that have predetermined control functions. Some have addressable internal memory and logic to be user-programmed by a programming language.

The CRTs slide over into the category of intelligent terminals or even microcomputers. Some keyboard/CRT terminals have the ability to have an additional peripheral "hung on" to them. One common option is a printer, so that what is displayed on the screen can be printed on request. Input devices such as OCR wands or light pens often are combined with keyboard/CRT terminals for special applications.

Intelligent Terminals. The term "intelligent terminal" is used in many ways. The following are often called "intelligent terminals": programmable interactive CRT terminals, shared processors for multistation key-to-disk systems, and small, single-application minicomputer or microcomputer systems. For the term to be accurately applied these minimum characteristics must be met.[52]

- Self-contained storage; random access memory
- User interaction with the terminal itself
- Stored program capability
- Processing capability at the terminal through a user-written program
- Capability of online communications with another intelligent terminal
- Human-oriented input, such as a keyboard
- Human-oriented output, such as a printer or a CRT

Intelligent terminals allow some point-of-source, or local, processing. Small files can be accessed on the spot before data input. An intelligent terminal can perform expanded editing functions, and data commonly are transmitted to the mainframe only when completely edited. Also, intelligent terminals often perform the communications control functions, freeing the mainframe processor for other tasks.

Input/Output Connections

In the most basic minicomputer systems, the peripherals are connected to the mainframe by cables or hardwire lines because the mainframe is small and usually located close to the peripherals. Even when the terminal is in another room, a cable connection can be made. Theoretically, there is no limit to the length of the cable or the distance between the peripherals and the mainframe, although some devices have maximum limits. Over great distances, boosters or repeaters are used to amplify the signals.

There are times when a direct cable connection is not feasible; instead, data communications or telecommunications are used.[53] That is, the connection is made over telegraph- or telephone-type lines. The computer signals have to be formated or structured so that they can be transmitted over these lines. In most cases, the lines handle analog-type frequency waves, and the digital signals must be converted. A modem (modulator-demodulator) or data set is used to convert the signals at each end. The transmission must match the bandwidth and the range of allowable bit rates of the line.

The mode of communications can be asynchronous (start-stop) or bisynchronous (binary synchronous), and the transmission can be duplex or half-duplex (one-way or simultaneous two-way communication). The connection of the lines can be on a switched or nonswitched (point-to-point) basis, and contact between "stations" can be established through a contention system or a polling system. The entire data communications process requires special I/O controllers, interfaces, channels, acoustical couplers and/or modems, proper CPU instruction sets, and special systems software.

SOFTWARE

Minicomputer software has been notoriously lacking for general consumption. Until the last 2 or 3 years, a limited amount of software was commercially available; most had to be prepared for each system application. System software provided with the equipment was often sparse, and additional modules had to be purchased as options. This condition has changed somewhat, but on the smaller systems (such as microcomputers) there is still little comprehensive software available.

---

[52] Robert O. Ritchie, "Intelligent Terminals and Distributed Processing," *Computer Decisions* 7 (February 1975): 38.

[53] For a general description and glossary on data communications written especially for librarians, see Mark S. Radwin, "From Nodes to Modes-Duplex and Half-Duplex," *Online* 1 (January 1977): 14-19. For more detailed descriptions, refer to technical manuals such as: International Business Machines, *IBM System/360 Operating System Basic Telecommunications Access Method*, 3d ed., IBM Systems Reference Library (White Plains, N.Y.: International Business Machines Corp., Data Processing Division, 1968), pp. 9-17

The amount of software available affects the development time in preparing an application system. The sophistication of the system software can affect the uses of an installed, ongoing system, even though much of the support software is for use by the programmer/designer.

There are two basic kinds of software: systems and applications. Systems software is an umbrella term that covers programmer-support software, assemblers, compilers, and operating systems. Application software includes general-purpose packages and the final user-specific program.

SYSTEMS SOFTWARE

Corey explained minicomputer manufacturers' software, starting with the simplest versions and moving to the most complex.[54] He classed the simplest software as being made up of four programs: the bootstrap loader, the loader, the assembler, and the debugger. The GML *Minicomputer Review* lists the minimum programs for a minicomputer as diagnostics, binary loaders, debugging and utility routines, and editors.[55] The most basic types of programs are used by the programmer to make the hardware work and to create a new program.

Program Development Aids

Assemblers

An assembler relieves the programmer of the task of coding a machine language program in binary or octal.[56] The assembler translates a rather low-level mnemonic source language program into an object language program, which is the machine language – usually on a one-for-one, instruction-for-instruction basis. The object code is expressed in an absolute address (that is, a specifically assigned memory location). An enhancement available for an assembler is production of a relocatable format or address that can be loaded into and executed from any area in main memory.[57]

Some manufacturers have, as an option, a macroassembler. For this type of assembler, the user-defined mnemonic codes for operations represent multiple instructions in machine language. A macro represents a formal sequence of coded instructions and, when evoked, results in the entire sequence being entered in machine language.[58] The length of the program in memory is not shortened but it does save some of the programmer's time. Cross-system assemblers are available from some manufacturers.[59] These allow the actual assembly of minicomputer programs to be performed on a large-scale computer system and then mounted on the mini for execution. This process is faster and more efficient and can save memory in the mini.[60]

Loaders

The computer hardware "knows" nothing — not how to use a program, and not even how to accept input of the program.[61] When a minicomputer is turned off, or powered down, it returns to this ignorant state. A hardware device can be used to load memory with a program, or a software program called a bootstrap loader can be used. The bootstrap loader is used to specify the data to be deposited and the memory address where the data are to go. It is a short program and actually is used to load the absolute or relocatable loader program, which is a longer, more comprehensive system program used by the programmer to load programs into the memory locations. When this loading is complete, the control is taken over by the regular (i.e., newly loaded) program. A linker-loader or a linkage editor is used to keep track internally of where the various programs are located in main memory and allows them to be linked into modules as required for a total program.

Editors

An editor is of value in creating programs. A set of command words and the instructions they represent allow the programmer to create a new program (more accurately to enter a new program previously designed and coded) with aids and prompts or to make changes to the program previously written without starting from scratch (without reassembling). One type of editor is the line editor which allows lines of the source program to be added, deleted, or modified.[62] Another type is the string editor which allows the programmer to add, delete, or modify character strings. The most comprehensive is the text editor which retrieves lines of text from a file, allows the programmer to correct the text by inserting or deleting characters or whole lines, and returns the corrected lines to the file.[63] Text editors are used online in an interactive mode, which is the most sophisticated mode.

---

[54] Corey, "Configurations and Software," pp. 20-27
[55] GML, *Minicomputer Review 1975*, p. Profile-1
[56] Eckhouse, *Minicomputer Systems*, p. 40
[57] *Auerbach on Minicomputers*, p. 64
[58] Eckhouse, *Minicomputer Systems*, p. 207
[59] Hollingworth, *Minicomputers*, p. 26
[60] Corey, "Configurations and Software," p. 21
[61] Eckhouse, *Minicomputer Systems*, p. 167

Debuggers

A debugger program helps the programmer determine what is wrong with the program. Although the assembler can check for and detect syntactic errors, logical errors are usually found only when run on the computer.[64] The debugger program allows the programmer to view the internal processes and conditions of the computer as the program is run. For example, if the program stops due to a fatal error, the debugger program provides "extensive information about the state of the machine at the time of the failure."[65] An online, or dynamic, debugging program allows the programmer to perform the following tasks.[66]

1. Start a program
2. Suspend its execution at predetermined points
3. Examine and modify the contents of memory words and registers
4. Make additions and corrections to the running program using either symbolic or octal code

This is the most efficient way to debug a new program or repair problems in an existing program.

Diagnostics

A diagnostics program is used to test the equipment and determine if it is functioning correctly. It can be used as part of a maintenance procedure or, if problems have arisen, to determine if there are hardware stresses or malfunctions or memory stresses or problems and to locate and identify the problem areas.

Utilities

Utility programs are programmer aids that handle certain recurring functions, such as[67]

- Moving data from cards to tape or vice versa
- Moving data from cards to disk or vice versa
- Moving data from tape to disk or vice versa
- Moving data from cards, tape, or disk to the printer.

These programs generally perform such routine operations as moving, printing (listing), or dumping data, which can be performed without manipulation or any recognition of specific file format. For example, an 80/80 listing is one in which data on 80-column cards are printed as punched. The programs also perform general system maintenance, such as disk space allocation, system utilization accounting, and construction of program libraries.

Subroutine Libraries

Some small programs or subroutines are written to handle mathematical functions and are called up or linked to the application program as needed.[68] The common ones for minis are for fixed-point arithmetic (multiply, divide, double precision), floating-point arithmetic, conversion of data formats (decimal to binary, fixed point to floating point), and trigonometric functions. These are especially important in minis because they enable software options to replace unavailable hardware features.

Compilers

Although a compiler is definitely a program development aid, it is not a requirement of a support software system. A compiler is similar to an assembler in that it translates a source language into object, or machine, language. Compiler languages are higher level and often designed to be used on any type of machine. The common standard languages are FORTRAN, BASIC, and COBOL. Each is designed for a certain area of application: FORTRAN is designed for scientific applications and any other area requiring high-level mathematical computations; COBOL is a business-oriented language and handles words, text, etc., with great input/output and file format flexibility. Some manufacturers have created their own high-level languages and compilers to best suit their own equipment (for example, DEC's DIBOL).

---

[62] Barden, *Minicomputers & Microcomputers*, p. 57
[63] Corey, "Configurations and Software," p. 25
[64] Eckhouse, *Minicomputer Systems*, p. 221
[65] Corey, "Configurations and Software," p. 25
[66] Eckhouse, *Minicomputer Systems*, p. 223
[67] Corey, "Configurations and Software," p. 23
[68] *Auerbach on Minicomputers*, p. 59

Each language has a set of standard words or instructions that represent common functions or routines — these are similar to macro-assembler instructions. Working storage control is handled by the language conventions, as are file definitions and file handling. Compilers have checking features to diagnose programmer errors in language use, including errors in both semantics (meaning) and syntax (form).[69] Error messages are displayed so the programmer can correct the source code and the program can be recompiled into executable object code.

Compilers are used in several modes. The source program can be compiled once and stored ready to execute. The source program can be compiled and then immediately executed (this is called compile-and-go). Some compilers are in the interpretive mode, where each statement is individually executed. (This mode is typical in conversational language compilers. A common interpreter is in BASIC specifically designed for interactive programming.) Compilers take space in main memory and use main memory as working storage while executing. Some compilers can be used only on CPUs of certain minimum sizes. One source states that BASIC and FORTRAN compilers require a CPU with 4K to 12K memory, while a COBOL compiler requires 8K to 16K memory.[70]

When compilers remain resident in memory at all times, a system designer must allow additional memory for the other programs and data, or the memory would be compiler locked (bound). As stated earlier, compilers are not necessary for systems. Without a compiler the programmer could code the programs in assembler language. More effort would be required (more lines of instruction, more file and memory location control, and less subroutine use, as well as more programming expertise in general), but often a more efficient program can result with maximum use of the CPU and main memory for that specific application.

File Management Programs

Another class of programmer support software is involved with a slightly higher level of data processing of a recurring nature — the basic operations and management of files. These operations include building or establishing a file (including definition of file formats); manipulating files and parts of files, such as combining files, splitting files, adding or deleting parts of files; maintaining files (including adding, changing, and deleting individual records as a result of transactions); searching files to retrieve specified records or fields of data according to request criteria phrased in Boolean form; and sorting files by specific element(s) within the records, using specified sequencing formulas.[71] These operations are written in separate programs so that a programmer can use one or several file management programs as required for the application.

The most comprehensive file management program package is called a data base management system (DBMS).[72] It covers operations such as[73]

- Interface of all applications to an organizational data base;
- Creating and maintaining files;
- Selecting, retrieving, sorting, and routing data for applications;
- Managing and maintaining data files;
- Generating and formating utilization reports;
- Providing for the integrity of the data in the files.

Data base management systems are used for complex applications with many files and file structures and many users.

Operating Systems

Operating systems go beyond program support programs but are not applications programs. In fact operating systems allow application programs to be executed efficiently on the computer hardware.[74] An operating system is not mandatory; the system can be operated manually by the user, but this is not very efficient. An operating system is a software package that provides support in the areas of program coordination, device communication, and internal and auxiliary storage organization. "A complete operating system performs all functions required for programmed system control. It communicates with the operator to request parameters or to report status; loads, links, and sequences programs for execution; assigns I/O devices to programs; performs all I/O operations; and services interrupts."[75]

---

[69] Ibid., p. 66.

[70] GML, *Minicomputer Review 1975*, pp. Profile-1 — Profile-4.

[71] Corey, "Configurations and Software," p. 23, and Hayes and Becker, *Handbook of Data Processing for Libraries*, p. 266.

[72] For more information on DBMS, see: "Data Base for the Mini User," *Mini-Micro Systems* 9 (June 1976): 30.

[73] U.S., Civil Service Commission, Bureau of Training, ADP Management Training Center, "Management Introduction to Automated Data Bases," Washington, D.C., n.d., p. F-1. (Mimeographed.)

[74] Stein and Shapiro describe how an operating system works and evaluate some of the operating systems supplied by minicomputer manufacturers; see: Philip G. Stein and Howard M. Shapiro, "That Makes OS Racing," *Computer Decisions* 6 (November 1974): 46-47.

A complete operating system requires a mass storage device to store programs and data, and each type of mass storage device requires its own specific type of operating system. Thus we have cassette tape operating systems (CTOS), tape operating systems (TOS), disk operating systems (DOS), and virtual (memory) operating systems (VOS or VS).

The heart of the operating system is the supervisor, or monitor, a master control program that remains resident in memory.[76] It is responsible for initiation, maintenance, and termination of all other programs. It processes the communications among the user, the system, and the many control programs. It also acts on monitor calls, validates and transmits I/O calls to device handlers, supervises data and file manipulations, and provides error diagnostics. A main service provided by an operating system is file maintenance, especially in the area of creating and maintaining a directory that contains the location of all the files currently used by the program.[77] It also protects files so that data are not destroyed inadvertently.

Operating systems are organized differently and function differently according to the mode of operation.

1. *Simple executive, single process.* This kind of system is designed to handle the program interrupts and I/O control for one single application for one user.
2. *Single batch.*[78] Batch processing is a technique in which jobs are collected and grouped before processing. A single batch operating system links serially the functions or programs required to perform one entire procedure for one user. For example, to update a serials holding file, a read program, a sort program, a file maintenance program (to update the file), and a print program may have to be used sequentially. The single batch operating system would handle the execution of these programs with little human intervention.
3. *Multiprogram batch.* This operating system allows several jobs or more than one batch stream to be run "at the same time." This seemingly simultaneous processing of several programs is produced by the operating system's ability to transfer control of the CPU between programs. For example, when one program must await the completion of an I/O request, the other program can be executed by the CPU. Additional sophistication is achieved when an operating system has a feature for "quantum" or time-sliced program execution. Time-slicing involves establishing a fixed quantum of time for the execution of each program, and when the quantum expires, the operating system transfers control to a different program already resident in memory.[79]
4. *Online interactive.*[80] The essence of an online environment is that the input data enter the computer directly from the point of origin and/or the output data are transmitted directly to where it is used. Processing occurs in a single transaction as opposed to being deferred and gathered with other transactions to be processed in a batch. This process requires the operating system to perform a different kind of control and sequencing, because there is not always a predetermined sequence of jobs or programs. Rather, the jobs are determined by the user at the time of input or during processing through a conversation or dialog.
5. *Time-sharing.* This process allows multiple users to share system resources in such a way that each thinks he is the sole user. Time-sharing combines the multiprogramming concept with the online interactive mode. The computer handles several jobs in a dynamic state by jumping back and forth between programs as required. A time-sharing operating system is made up of a sophisticated set of control programs "to handle the sharing of system resources, the time-slicing, the storage allocation and program relocation, and the basic servicing of the users, besides the types of operations normally associated with an . . . operating system."[81]

---

[75] *Auerbach on Minicomputers*, p. 60.

[76] Eckhouse, *Minicomputer Systems*, pp. 237-8.

[77] *Auerbach on Minicomputers*, p. 61.

[78] *Introduction to Minicomputer Networks*, (Maynard, Mass.: Digital Equipment Corporation, 1974), p. C-2.

[79] Eckhouse, *Minicomputer Systems*, p. 247.

[80] International Business Machines, *Data Processing Glossary*, 3d ed. (Poughkeepsie, N.Y.: International Business Machines Corp., Programming Systems Publications, 1971), p. 71.

[81] Eckhouse, *Minicomputer Systems*, p. 247.

APPLICATIONS SOFTWARE

To the user of the computer system, the only software of interest is that which allows his data to be processed to accomplish his or her objective. This user-specific, application-oriented software is developed through the use of the systems software and made functional by means of the operating system, but the success of the project depends on the responsiveness, efficacy, and efficiency of the applications software itself.

Applications software is the end of the systems analysis/systems design/systems development process. The flow charts and decision tables become more specific to the application, until a programmer can take the material and write the actual programs. The analyst/designer will have determined the hardware configuration, the system specifications, the files and file structures, the operating mode, and the system software that can be supplied. The programmer then uses any or all of these to write and debug the final module.

Sometimes it is more efficient to base the final module on a general purpose software package that can be purchased from the manufacturer or an independent software vendor. These packages vary in what they cover and to what degree they cover the function. There are total, comprehensive data base management systems such as TOTAL, and there are business application packages that provide 80 to 85 percent of the programming with the remaining 15 to 20 percent to be done by the user to achieve a user-specific module. There are also expanded packages for text-processing, search and retrieval, and data communications. The programmer must decide if and when a general purpose package is useful.

Another option is the use of a complete, turnkey applications package that can be purchased. A special-purpose package requires the user to provide only his own format details and information. The problem is locating a module that meets all of the system specifications for the user's needs.

CONFIGURATIONS

As stated previously, the specific components of a minicomputer system are determined by the application. Four main types of configurations are appropriate for the library environment: a network node (often an intelligent terminal), a data communications node, a data collection station, and a standalone station.

In general, an intelligent terminal as a network node has a small main memory and a CPU with limited power. There is little need for mass storage. The display features are important, as are the communications capabilities: data transfer rate, protocol, and mode (duplex or half-duplex). There may be a requirement for hardcopy output. The systems software can be limited if most of the control is handled by the network.

A data communications node has little need of a man/machine interface but will probably control many I/O devices, both terminals and other computers. The CPU and the I/O controllers must be specially designed for communications control, as must also the systems software. There is little requirement for mass storage.

A data collection station can be an intelligent terminal or a larger unit controlling multiple stations. The unit must have enough main memory, input/output controllers, and CPU power to support multiple stations, and it must have some mass storage. The need to output data in machine-readable form may require special devices or a communications hookup. The input method must be fast and efficient to handle both initial entry and error correction.

A standalone system generally requires a balanced set of components and software to match the complexity or sophistication of the application.

CLASSES OF MINICOMPUTERS

The range of equipment and of features available allows almost unlimited patterns or combinations of components. To provide a frame of reference, five classes of minicomputer systems have been developed. They are designed to provide support to libraries with applications ranging from simple to complex.

Class I minicomputers are little more than intelligent terminals used for data collection. This configuration can do little more than capture the data and write them out in the same form. It has no ability to manipulate data (for example, to sort or update files by resequencing or merging). The CPU is simple, with 8-bit words and an 8K main memory. It can handle only one user and only two input/output devices: one mass storage device, such as paper tape or a tape cassette, and one man/machine interface device, such as an unbuffered CRT/keyboard terminal or a Teletype with a paper tape unit. Little software is provided, only an assembler and a BASIC compiler.

Class II minicomputers can handle more general applications for single users. Manipulation capabilities include tabulating or summarizing data from specific fields in the records of the file; searching and printing data; and outputting in the same sequence as on the file but with the data reformatted if desired. No sorting is possible. The CPU has either 8-bit words and up to 64K main memory or 16-bit words with up to 32K main memory. The I/O control can handle both programmed I/O and DMA. Three I/O devices can be supported: usually one man/machine interface device, such as a buffered CRT or a Teletype and printer, and two mass storage device drives. This class of minis can operate sequential files with paper tape, punch cards, tape cassettes, or magnetic tapes; or it can operate a direct-access file on a floppy disk (with up to two drives). The only communications link possible is a direct line to a host computer. The support software is somewhat more complete than that of Class I; the operating system is either a simple executive, single-process system that must be totally attended, or a single batch system in which one job stream can be computer-controlled. The operating system type depends on the mass storage device, e.g., tape or diskette. The compiler can be for BASIC or FORTRAN.

Class III minicomputers are general-application systems for single users. They can handle single or multiple batch operations. The processing can handle sorting and any other manipulations possible in a batch mode with a quite large file capacity. The CPU has 16-bit words and up to 64K main memory. As many as eight different I/O devices (or devices and drives) can be supported; and concurrent communications are possible with a host computer over a bisynchronous line. Mass storage devices can be tape cassettes, magnetic tape, or floppy disks. The operating system would handle multiprocessing for distributed networking.

Class IV minicomputers introduce interactive processing for multiple online users with a single application. They offer great versatility in manipulations such as sorting and searching. Users can be remotely located because there are four to eight asynchronous lines, and the software necessary for controlling the telecommunications. The CPU has 16-bit words, and the main memory is either 128K real memory or 64K virtual memory. The input/output controller includes PIO, DMA, and a multiplexer. The mass storage devices now can include disks (fixed-head platters, movable head disk cartridges, or disk packs). The operating system will probably be DOS (if the big disk devices are used) or VOS and an online, interactive type. The compiler can be for BASIC, FORTRAN, or COBOL. The system software would be a complete complement.

Class V minicomputers are the largest, most sophisticated systems. They can handle time-sharing for multiple online users. They have large file capacity and are very powerful. The top end of this class overlaps large-scale computer classes. The CPU has 16-bit words and 64K virtual memory. The operating system would be a VOS of a time-sharing type. The system software would probably include a data base management package.

CHAPTER FOUR:

MINICOMPUTERS—LIBRARY APPLICATIONS

SYSTEM SELECTION: THE BRIDGE

With the variety of minicomputer components, designs, and features to choose from, how does one select the proper system? Selection must start with the specific application and its requirements. A bridge must be built connecting the specific requirements of a library operation with the detailed specifications prepared for an RFP or a contract. This bridge must be built each and every time a library chooses a system because each library's requirements differ from those of others. Every bridge is supported by five columns (Table 11): (1) design characteristics, (2) hardware impacts, (3) software impacts, (4) library system specifications, and (5) the specific class of minicomputer required. The following sections describe the process of erecting these columns.

TABLE 11—Design Model: The Bridge

Column 1	Column 2	Column 3	Column 4	Column 5
Design Characteristics	Hardware Impacts	Software Impacts	Library System Specifications	Class of Minicomputer

COLUMN ONE: DESIGN CHARACTERISTICS

In looking at a potential minicomputer application (or even automation in general), the first review must be in conceptual terms that ignore the specific tasks involved.

The design characteristics of a system can be broken down into six categories, which are not necessarily discrete:

- Means of inputting
- Types of output products
- File structure and size
- Transaction/volume
- Applications characteristics
- Interfaces to other systems.

These design characteristics are not definitive. Not all will be of interest in all applications. The categories represent, however, the kinds of considerations that arise in system design. Table 12 lists these considerations in detail.

TABLE 12—Design Characteristics

MEANS OF INPUTTING
  Nature of the data
    alphanumeric
    upper and lower case
    special characteristics
  One-way/conversational input
  Combinations of input devices
  Location of input stations
  Multiple online users
  Outside sources

TYPES OF OUTPUT PRODUCTS
  Hardcopy
    upper and lower case, diacritics
    special forms requirements (cardstock, multipart forms, multiple copies, etc.)
  Display
    character sets
    size of display
    features
    scroll or block
  Combination hardcopy and display
  Machine-readable output
    outgoing communications line
    machine-readable form (offline)

FILE STRUCTURE AND SIZE
  Structure
    sequential
    direct access
  Size
    large number of characters
    number of characters required at one time (online)

TRANSACTION/VOLUME
  Expansion and growth
  Peaks and pressure points of activity
  Response time

TABLE 12—Design Characteristics—Continued

- APPLICATIONS CHARACTERISTICS
  - Sorting/data manipulation
    - nature: numeric or bibliographic
    - amount (number of records)
  - Searching
    - batch
      - Boolean logic
      - free text
      - numerous access points
    - online interactive
      - Boolean logic
      - free text
      - numerous access points
      - multiple users
  - Special input/output hardware
    - light pen, badge reader, OCR scanner, etc.
  - Remote access
  - Response time (online)
- INTERFACE WITH OTHER SYSTEMS
  - Offline interface
  - Online interface
  - Links to multiple systems

Means of Inputting

The input to a system involves getting the data from its source into the computer — from human-readable form into machine-readable form. This is where the human interfaces with the machine. The real impact of any automation effort is felt here first and cannot be eliminated entirely. The best that can be done is to make the means of inputting as efficient as possible.

Input in its larger sense includes the entire data handling process: data capture, transcription, keying, verification, error correcting, and sometimes transmission for processing. It has been estimated that up to 50 percent of a data processing budget can be involved with data handling. This proportion is true for the library also. Cox, Dews, and Dolby wrote that "in a computer system, the most time-consuming and expensive single operation is probably the original preparation of the data in a machine-readable form."[1] Ways to reduce the costs include reducing the amount of manual handling and rehandling of the same data, increasing the ease of both data "keying" and data editing/correction; and increasing the speed of data "keying," "reading," and "transmitting."

Attention must be given to the impact of the inputting on the computer system as well as the impact on the library staff who must perform it. Sometimes the two areas conflict, and compromises must be made.

MEANS OF INPUTTING

- Nature of the data
- One-way or conversational input
- Combinations of input devices
- Location of input stations
- Multiple online users
- Outside sources

Library data are quite complicated in data processing terms. They are mainly alphabetical with requirements for upper and lower case, special characters such as diacritics, and different type fonts such as bold, light, and italics. Hayes and Becker point out that in the library a number of major categories of data can be compiled: management data, circulation data, cataloging/indexing data, selection/acquisitions data, and textual data.[2] Analyzing the nature of the data involved in the new system may dictate the use of some equipment or militate against the use of other devices.

One-way input involves transmitting data to the computer without receiving transmission back at the time of input. One-way input generally is used with batch processing. The initial data conversion document is prepared offline, and the data are gathered into a group or batch and then read or entered into the system as high-volume input. Another form of one-way input is the use of a light pen to read bar code labels in an online mode. Although a light is often used to flash an indication that the data were received, this method is basically one-way as opposed to conversational.

Conversational input is a feature of an interactive processing mode. Input often is in the form of a dialog between the user and the program. Questions are asked and answered, display forms are filled out, problems are noted and corrections prompted. This type of input requires a device that can handle two-way communications — both input and output. This method can ease the editing and error-correction problems of input, but it is slower and in some systems ties up the CPU. If the volume of input is heavy, it can bind up the system if precautions are not taken.

Sometimes applications call for the use of more than one type of input (for example, a light pen plus an interactive terminal for circulation, a card reader for reentry records and a key-to-tape station for a serials system, or a magnetic tape reader for MARC II distribution tapes and an intelligent terminal for cataloging input). Each device is used one at a time, but the fact that there is more than one type must be considered.

---

[1] N. S. M. Cox, J. D. Dews, and J. L. Dolby, *The Computer and the Library, The Role of the Computer in the Organization and Handling of Information in Libraries* (Newcastle upon Tyne, Eng.: University of Newcastle upon Tyne Library, 1966), p. 18.

[2] Robert M. Hayes and Joseph Becker, *Handbook of Data Processing for Libraries*, 2d ed. (Los Angeles: Melville Publishing Co., 1974), pp. 278-9.

The physical locations of the various input stations also must be considered. If the input is handled at a point next to the mainframe, one kind of connection is proper. If it is handled at a remote location, more sophisticated connections may be required.

If more than one user must enter data at the same time, the system must be designed to handle more than one input line. This problem occurs in the online interactive mode and is really a sophisticated requirement. From the user's standpoint, however, it may be the key to a truly useful system.

Because of the time and expense involved in data conversion, it is important to enter information in machine-readable form available from outside sources whenever possible. Input may be in an online, computer-to-computer, or an offline, read-on-request environment.

Types of Output Products

The output products of a computer system make the results of data processing available for use by humans or by another computer system. The form of the output and the amount, format, sequence, frequency, and permanence of the information supplied by the computer all have bearing on the system itself. As with input, the special requirements of library data must be considered in selecting output devices. It is important that the products called for be thought out carefully in terms of the new system, and that they not be mere replications of current manual products. For example, is there any need for a weekly books-on-order list if the on-order file is available through online, direct access at any time?

TYPES OF OUTPUT PRODUCTS

- Hardcopy
- Display
- Combination hardcopy and display
- Machine-readable

The best known type of hardcopy output is the 11-by-14-inch continuous-form printout produced by a computer-driven printer. Other types of hardcopy products are appropriate for some applications: cardstock, stencils, multipart forms, standard 8½-by-11 bond stationery, and 2-, 4-, or 6-up continuous-feed paper. The quality of the type can be critical. Most hardcopy output is used internally by the staff and for limited periods of time, but such products as computer-produced book catalog pages and catalog cards are more permanent and are used by the public; they require greater legibility and better esthetics. If large amounts of hardcopy output are required, the speed of the output devices becomes critical. The system can be output-bound if the devices are too slow.

Display output (a visual image on a screen) is often used in online, interactive systems. It is usually combined with a keyboard device as an interactive terminal. The display device must match the requirements of the nature of the output data and the requirements of the system in expressing or formating that data.

A combination of hardcopy and display output is used when there is a need for a permanent record or a special hardcopy form in an online interactive system that uses a CRT-type keyboard/display terminal for conversation. A printer is combined with the CRT/keyboard unit, sometimes for simultaneous output and sometimes as a print-on-request slave.

Machine-readable output is used to allow two computer systems to "talk" to each other. Output from the system also is used to send to other systems for further processing, such as COM production or photocomposition.

File Structure and Size

The intellectual content of the system, the data that are input, processed, referenced, maintained, created, and output, is made up of data elements handled as subfields or fields. Related fields are combined into a complete, logical unit called a record. Similar records are combined into a file. The order of the records on the file is called the file structure. Common structures are sequential, linked, indexed sequential, direct access, and random access.

The type of file structure selected usually depends on the size of the file (extremely large files almost always are maintained on magnetic tape and therefore in a type of sequential structure) and the nature of the operations conducted on the file (the kind of posting, referencing, and updating). A programmer can define the files as required for the most efficient processing for the application. There are different types of files: master files, transaction files, input files, output files, intermediate files, and reference files. One enormous file can be defined if desired, or a series of files can be defined with no redundancy and linked into an integrated data base. The way the files work against one another during processing also must be considered. In any case the files do affect the computer design.

FILE STRUCTURE AND SIZE
- Structure
    - Sequential
    - Direct Access
- Size
    - Total number of characters
    - Number of characters required at one time

A sequential file can be handled on any type of mass storage device. The device selected depends on such elements as speed, ease of handling, and cost. Sequential files typically are used in the batch mode.

A direct access file generally is used in the online interactive mode and can be handled only on a disk-type mass storage unit: floppy disk, fixed-head disk, or disk pack.

The size of a file can affect the system design. Although a hardware configuration can be put together to support an enormous file online on disks, it becomes inordinately and prohibitively expensive. Instead, slower, but cheaper, magnetic tape is often used. Each mass storage device has a maximum capacity, and the size of the file in characters can be used to compute the number of mass storage units required. For an online system, the number of characters that must be accessible at one time can influence the decision about which mass storage device to use.

Transaction/Volume

The capacity of a system, or its total size, is important of course. However, in many cases a more critical consideration is the peak load at any one time. For example, it is important to know the total circulation transactions per year, for counters and statistics parameters, but more important figures are the largest number of transactions outstanding at any one time (for file capacity) and the greatest number of borrowers at any one time (for input device load).

TRANSACTION/VOLUME
- Expansion and growth
- Peaks and pressure points of activities
- Response time

Although minicomputers need not have the lifespan of large-scale computers and therefore do not have to be installed with a capacity to handle long-range future growth, expansion and growth should be considered. Some hardware can handle only one input/output device; some can handle only one user at a time; some CPUs can be expanded to only a certain memory limit; and some I/O controls can drive only as many as four devices. If growth or expansion is required and the hardware cannot accommodate the changes, an entirely new system (hardware and software) may be needed.

The peaks and pressure points in activities of a system are of vital concern, but they are very difficult to determine correctly in advance as far as CPU cycle time, memory access time, I/O interrupt response time, etc., are concerned. The peaks and pressure points must be considered for the man/machine interface devices also. The number of lines of printing required at one time must be used to compute printing time based on different printer speeds. The number of documents to be keyboarded at any one time must be used with a unit time for entry to determine how many input units are required for one work shift, two work shifts, and so on.

Response time must be considered for all systems. If immediate status information is required, the batch mode may be eliminated and an online interactive system may be needed. Then, with an online system, response time involves the amount of time the user waits at the terminal for an answer from the system. Sometimes heavy loads in a time-sharing system can slow down response. These can be in the I/O control, memory access, or processor areas. For example, some systems are stated to have capacity for eight users, but any more than four online at a time can slow the system down significantly.

Applications Characteristics

What is wanted from the system? What can the system do? The answers to these questions are important in describing the application the system will serve. The type of operations to be performed and the nature of each must be reviewed.

APPLICATIONS CHARACTERISTICS
- Sorting
- Searching
- Special I/O hardware
- Remote access
- Response time
- Reliability

If the files require much manipulation to sort the data, this must be considered by the hardware designer. Also for consideration are the nature of the data to be sorted (alphabetic or numeric), and the number of records to be sorted. The solution to the sorting problem may be to provide a large amount of working space in which to perform the sorting or to provide redundant files, one for each sequence, and to update them, which will increase the amount of mass storage required.

When, how often, and by what elements the files are searched are significant factors in systems design. How a file is to be accessed must be considered in terms of operational mode (batch or online interactive); the search criteria (controlled vocabulary, Boolean logic, or free text); the number of access points in each record, and whether inverted files are used. As the choices are made, careful tradeoffs among system cost and efficiency and user service must be made. For example, file accessibility only through a controlled vocabulary (i.e., thesaurus) is a simple, straightforward systems procedure, but it shifts to the user the heavy responsibility of editing and assigning descriptors at input time. Accessibility of the file by any word (e.g., a free text search method) relieves the user at input of intellectual decisions, but it requires sophisticated hardware and software that are very costly to implement.

Some applications are better served by use of special input/output hardware. In a library, efficiency in the circulation application may require a badge/card reader input station or a light-pen wand for bar labels. A serials system may require computer-punched 80-column cards for use as reentry records. These special requirements must be stated at the beginning.

Access to a system by means of terminals physically removed from the mainframe is called remote access. The unit may be in the next room, the next building, or the next county. Direct line connections can be used for some remote access locations, but over longer distances the laws of physics prohibit the use of direct lines. Remote terminals at these distances must be connected by telecommunications lines, and these have direct impact on both hardware and software.

The response time required varies according to the application. If critical, response time should be a system requirement and considered an application characteristic to be given special attention in the system design.

Reliability is always desirable in both manual and automated systems. Some applications, however, impose stricter constraints on downtime than others; if reliability is critical, it should be stated as an application characteristic. The solution may be an alternative manual backup procedure; a duplicate, totally redundant computer system; or a second system capable of multiprocessing.

Interface With Other Systems

As a system is designed, other systems in the same library or agency and in the library community in general should be considered. The ability to interface with these other systems, whether directly or indirectly, must be established if possible. Sometimes the information flow will be one-way, sometimes two-way. (Implicit in this concept is the exchange of data in machine-readable form.)

INTERFACE WITH OTHER SYSTEMS

- Offline
- Online
- Links to multiple systems

Offline interface merely provides a means of transferring data in machine-readable form by physically transporting the medium (i.e., punch cards, magnetic tape, or floppy disk) to another computer to be read rather than transmitting data directly over a communications line.

Online interface allows direct computer-to-computer communications. The communications exchange can be one-way or two-way, continuous or batch.

"Links-to-multiple systems" is another way of saying distributed networking in which the processors and their operating system software allow tasks and resources to be divided and shared throughout the network. For example, the file used by system A may be a part of, and physically located at, system B. Transfer of data, control of the manipulation, and control of input/output devices are all shared and transferred back and forth as required by the operating systems. These functions are technically complicated but promise greater efficiency and efficacy in the long run.

COLUMNS TWO AND THREE: HARDWARE AND SOFTWARE IMPACTS

When design characteristics have been reviewed, the impact of each is expressed in terms of hardware and software. These columns are interdependent. Often a hardware device dictates a specific software module, and vice versa. Other times a characteristic can be handled by either hardware or software; the choice resides with the system designer and programmer. Table 13 expresses the typical hardware and software impacts of the design characteristics.

TABLE 13—Design Characteristics and Their Impacts

Design Characteristics	Hardware Impacts	Software Impacts
MEANS OF INPUTTING Nature of the Data Alphabetic and/or Numeric, Upper and Lower Case, Special Characteristics,	I/O DEVICE Keypunch machines not suited for special characters, but card readers can handle them. CRTs vary as to what they will handle, but most now handle upper and lower case with a few special characters. Only a few can handle diacritical marks. Tape and disk drives have no limitations.	APPLICATIONS SOFTWARE Must have necessary translation tables to match any character set(s) used.
One-way/Conversational One-way	I/O DEVICE Input only. Ex. keyboard, card reader, paper tape reader, badge reader station, light pen. I/O CONTROLLER. Must match device.	
Conversational	I/O DEVICE Two-way; output or response capability. Ex. Teletype or CRT/keyboard unit I/O CONTROLLER. Must match device.	APPLICATIONS SOFTWARE If a higher level language is used, it must have a compiler that supports conversational programming.
Combinations of Input Devices	I/O DEVICE Ex. card reader, CRT/keyboard unit, magnetic tape reader, light pen, paper tape reader, etc. I/O CONTROLLER Must consider different speeds. May have one for each type.	APPLICATIONS SOFTWARE Scheme or technique for input varies (block or line mode, character or record at a time). SYSTEMS SOFTWARE Must support the I/O functions for the different devices.
Location of Input Station	COMMUNICATIONS EQUIPMENT Modem or data set, with telecommunication lines. OR Hardwire lines (with repeaters).	SYSTEMS SOFTWARE Must support telecommunications functions.
Multiple Online Users	I/O DEVICE Two-way; conversational capability. COMMUNICATIONS EQUIPMENT Modem or data set, with telecommunication lines. OR Hardware lines (with repeaters). COMMUNICATIONS CONTROLLER May require a processor.	APPLICATIONS SOFTWARE Could be affected if a single program is to handle multiple users (as opposed to time-sharing. SYSTEMS SOFTWARE Must partition memory and be able to handle multiple users. Must be able to handle telecommunications functions.
Outside Sources Incoming Communications Line	COMMUNICATIONS EQUIPMENT Modem or data set, with telecommunication lines. OR Hardwire lines (with repeaters). COMMUNICATIONS CONTROLLER May require a processor.	APPLICATIONS SOFTWARE Possibly a requirement to reformat. May require a bridge program. SYSTEMS SOFTWARE Telecommunications functions support. OR Simpler technique to emulate a terminal.

TABLE 13—Design Characteristics and Their Impacts—Continued

Design Characteristics	Hardware Impacts	Software Impacts
MEANS OF INPUTTING—Continued		
Machine-readable Form (Offline)	I/O DEVICE Must be compatible with input (card reader, paper tape reader, magnetic tape reader, floppy disk reader).	APPLICATIONS SOFTWARE Scheme or technique for input varies to match device. SYSTEMS SOFTWARE Must support the I/O functions of the different devices.
TYPES OF OUTPUT PRODUCTS		
Hardcopy		
Upper and Lower Case, Diacritics	I/O DEVICE Printers. Must consider extensiveness of character set or flexibility for interchanging character sets. Line printer is more flexible than serial printer. Teletypewriters are upper case only. I/O CONTROLLER Must match.	
Special forms requirements Ex. cardstock, multipart forms, multiple copies	I/O DEVICE Printers. Must be equipped with a special card platen and card feed. Must accept special forms (e.g., multipart forms with carbon).	
Display	I/O DEVICE CRT unit or Digital Read-Out unit.	APPLICATIONS SOFTWARE Must reflect necessary programming techniques that are device-bound (background/foreground, blinking, reverse video, scroll or block mode).
Character Set	•Must consider the character set available; usually quite broad.	
Size of Display	•Must consider the required number of characters per line and the number of lines per screen for display.	
Features	•Desired features may include foreground/background display, reverse video, blinking/nonblinking.	
Scroll or Block	•Must consider whether display can be shown in an entire unit or block or does it move one line at a time. I/O CONTROLLER Must match device.	
Combination Hardcopy and Display	I/O DEVICE CRT plus printer. May print simultaneously with the display, which will be slow, or may print off the display on demand, allowing fast display and slow printing. I/O CONTROLLER Must match device.	
Machine-Readable Outgoing Communications Line	I/O DEVICE Must be compatible to network or host. COMMUNICATIONS EQUIPMENT Modem or data set, with telecommunications lines. OR Hardwire lines with repeaters. COMMUNICATIONS CONTROLLER May require a processor.	APPLICATIONS SOFTWARE May have to provide the software to move the data (actual transmission). Must be compatible to other end and may require a program as a bridge (reformat, protocols standards, etc.). SYSTEMS SOFTWARE May support a communications control processor function.

TABLE 13—Design Characteristics and Their Impacts—Continued

Design Characteristics	Hardware Impacts	Software Impacts
TYPES OF OUTPUT PRODUCTS—Continued		
Machine-Readable Form (Offline)	I/O DEVICE Ex. Punch cards, paper tape, magnetic tape, floppy disk. Device must be compatible with: •Protocols (density, number of tracks) •Format (standard)	APPLICATIONS SOFTWARE Must be compatible with other end. May require modules to meet format specifications.
FILE STRUCTURE AND SIZE		
Structure		
Sequential	MASS STORAGE DEVICE Punch cards, paper tape, cassette tape, magnetic tape, floppy disk, disk packs. I/O CONTROLLER Must match storage device.	
Direct Access	MASS STORAGE DEVICE Disk (floppy disk, fixed-head disk, movable-head disk) I/O CONTROLLER Must be a DMA direct memory access type.	APPLICATIONS SOFTWARE Method of retrieval required. Some type of indexing scheme required (e.g., index sequential, direct index, random number index). SYSTEMS SOFTWARE Disk operating system (DOS).
Size		
Large Number of Characters	MASS STORAGE DEVICE Eliminates punch cards and paper tape as being too slow and inefficient. Best to use magnetic tape or disk. Usually use direct access instead of sequential, which means disk.	
Number of Characters Required at One Time (Online)	MASS STORAGE DEVICE In online mode, must consider physical capacity per unit available at one time for accessing in terms of number of characters required at one time.	
TRANSACTION/VOLUME		
Expansion and Growth	I/O CONTROLLER Maximum number of user lines equipment can handle. MASS STORAGE DEVICE Maximum number of devices or units controller can handle. CPU Main memory: maximum that can be addressed.	SYSTEMS SOFTWARE Should consider that most have maximum number of remote users.

TABLE 13—Design Characteristics and Their Impacts—Continued

Design Characteristics	Hardware Impacts	Software Impacts
TRANSACTION/VOLUME—Continued		
Peaks and Pressure Points of Activity	CPU Speeds—(cycle time, how cycles are apportioned). COMMUNICATIONS EQUIPMENT Modem speeds. Data transfer rates. I/O DEVICE Rate of speed of operation (unit time versus volume).	
Response Time	CPU Speed (cycle time, how cycles are apportioned).	
Sorting/Data Manipulation Nature: Numeric or Bibliographic Amount (Number of Records)	MASS STORAGE DEVICE Requires large amount of working space and access to several areas at one time. *Tape System*—Would require minimum of three tape drives or large core storage as auxiliary. *Disk*—One would be sufficient but must be large. CPU Large amount of main memory required.	APPLICATIONS SOFTWARE Bibliographic data generally requires variable-length records, which require more sophisticated software development for data manipulation. SYSTEMS SOFTWARE Should require utility routines to provide sorting capabilities.
APPLICATIONS CHARACTERISTICS		
Searching		
Batch	MASS STORAGE DEVICE Any type will work, but should consider the speed of the reader in terms of processing time (card reader, paper tape reader, magnetic tape, disk).	
Boolean Logic	MASS STORAGE DEVICE Any type (cards, paper tape, magnetic tape, disk)	
Free Text	MASS STORAGE DEVICE Any type (cards, paper tape, magnetic tape, disk); speed will be the variable.	
Numerous Access Points	MASS STORAGE DEVICE Any type (cards, paper tape, magnetic tape, disk); speed will be the variable.	
Online	MASS STORAGE DEVICE Disk (must handle direct access).	APPLICATIONS SOFTWARE Generally requires inverted file handling (complex, updated software required. A general-purpose software package might be considered due to the complex nature of search/retrieval programs. SYSTEMS SOFTWARE Requires a time-share operating system (DOS-oriented).

TABLE 13—Design Characteristics and Their Impacts—Continued

Design Characteristics	Hardware Impacts	Software Impacts
APPLICATIONS CHARACTERISTICS—Continued		
Searching, cont.		
Online, cont.		
Boolean Logic	CPU Large amount of main memory required. Sophisticated instruction sets required.	
Free Text	CPU Large amount of main memory required. Sophisticated instruction sets required.	
Numerous Access Points (Multiple Inverted Files)	MASS STORAGE DEVICE Disk; must have direct access online to data base and all inverted files. CPU Sophisticated instruction sets and register structure required.	
Multiple Users	CPU Large amount of main memory required. Sophisticated register structure and instruction sets Must have memory (storage) protection feature. I/O CONTROLLERS Must handle multiple lines. Must be compatible. COMMUNICATIONS CONTROLLER May require a concentrator. May require a processor. COMMUNICATIONS EQUIPMENT Modem or data set, with telecommunications lines. OR Hardwire lines with repeaters.	
Special Input/Output Hardware Light Pen Badge Reader OCR Scanner	I/O CONTROLLER Must match interface with CPU. Must consider data transfer rates.	APPLICATIONS SOFTWARE Scheme or technique for input varies to match device. SYSTEMS SOFTWARE Must support input/output functions for the different devices.
Remote Access	COMMUNICATIONS EQUIPMENT Modem or data set, telecommunications lines. COMMUNICATIONS CONTROLLER May require a processor. I/O CONTROLLER Must match.	SYSTEMS SOFTWARE May require support of telecommunications functions.
Response Time (Online)	CPU Speeds (cycle time), how apportioned.	

TABLE 13—Design Characteristics and Their Impacts—Continued

Design Characteristics	Hardware Impacts	Software Impacts
APPLICATIONS CHARACTERISTICS—Continued		
Reliability	Duplicate part of the hardware. CPU: duplicate. I/O CONTROLLER: duplicate. I/O DEVICE: have one or two spares. COMMUNICATIONS EQUIPMENT: have backup lines. Alternative methods I/O DEVICE Have an offline device to gathered data. Ex. punch cards to replace online keyboard. Must be compatible to the total system configuration. I/O CONTROLLER Must match device. CPU Intelligent terminal may handle data input in a local processing mode and transmit when system is up. Total system configuration must be designed to handle this method.	
INTERFACE WITH OTHER SYSTEMS		
Offline	I/O DEVICE Card reader, paper tape reader, magnetic tape reader, floppy disk reader, etc., must be compatible with input in terms of protocols (density, number of tracks) and format (standard)	APPLICATIONS SOFTWARE Scheme or technique for input varies to match device. May require modules to meet format specifications. SYSTEMS SOFTWARE Must support input/output functions of the different devices.
Online	I/O DEVICE Must be compatible with network or host. COMMUNICATIONS EQUIPMENT Modem or data set, with telecommunications lines. OR Hardwire lines with repeaters COMMUNICATIONS CONTROLLER May require a processor	APPLICATIONS SOFTWARE Must be compatible with other end. May require modules to meet format specifications. SYSTEMS SOFTWARE Telecommunications functions support, or a simpler technique to emulate a terminal.
Link to Multiple Systems (Distributed Network)	CPU Large amount of main memory Sophisticated instruction sets COMMUNICATIONS CONTROLLER May require a processor May require a concentrator COMMUNICATIONS EQUIPMENT Modem or data set with telecommunications lines. I/O DEVICE Must be compatible with the network configuration requirements in terms of protocols (density, number of tracks) and format (standard)	SYSTEMS SOFTWARE Operating system must handle multiprocessing. Telecommunications functions support

Table 14—Classes of Minicomputer Systems

Characteristics	Class I	Class II	Class III	Class IV	Class V
CPU	8-bit word. Memory: up to 8K real.	8- or 16-bit word. Memory: 64K (8-bit) or 32K (16-bit) real.	16-bit word. Memory: up to 16K real.	16-bit word. Memory: up to 128K real or 64K virtual.	16-bit word. Memory: 64K virtual.
Number of I/O Devices	Two.	Three.	Eight.	Eight.	Eight.
Mass Storage Devices	Paper tape or tape cassette.	Punch cards, paper tape, tape cassettes, magnetic tape, or diskette.	Paper tape, punch cards, tape cassette, magnetic tape, diskette.	Paper tape, punch cards, tape cassette, magnetic tape, diskette, disk cartridge, or disk pack.	Paper tape, punch cards, tape cassette magnetic tape, diskette, disk cartridge, or disk pack.
Man/Machine Interface Devices	Teletype, paper tape unit, or unbuffered CRT.	Teletype, paper tape unit, plus printer. Buffered CRT.	Teletype, paper tape unit, plus printer. Buffered CRT.	Teletype, paper tape unit, plus printer. Special devices. Buffered CRT.	Teletype, paper tape unit, plus printer. Special devices. Buffered CRT.
Communications		Direct line to a host.	Direct line to host for concurrent communications.	Direct line to host, 4 to 8 lines to terminals.	Direct line to host, 8 to 32 lines to terminals.
Software	Assembler or BASIC. Little program support.	Assembler, BASIC, or FORTRAN. Limited program support.	Assembler, BASIC, or FORTRAN. Program support software.	Assembler, BASIC, or COBOL. Full program support software.	Assembler, BASIC, FORTRAN, or COBOL. Full program support software. Data base management package.
Operating Mode	Simple executive monitor (attended)	Simple executive or single batch	Single- or multiple-batch processing.	Online interactive or batch processing.	Time-sharing online interactive processing.
General Capabilities	Data collection and list, editing but not manipulation	Summarizing, tabulating, searching, reformating No sorting.	General applications, including sorting.	Online interactive system for multiple users on a single application.	Time-sharing with multiple users online using multiple applications.

COLUMN FIVE: CLASS OF MINICOMPUTER

The final column of the design bridge is the class of computer system appropriate to the application characteristics. (Column four will be discussed later.) Again, it must be emphasized that these categories are merely representative samples. The proper configuration for any system may cut across several of these categories, for the number of combinations is almost infinite. The categories are ordered in a hierarchy from simple to complex or basic to sophisticated (see Table 14).

COLUMN FOUR: LIBRARY SYSTEM SPECIFICATIONS

The fourth column, that of assessing the library application, can now be discussed. This column, however, must be left blank in this book. The answers must come from each individual library. For a library cataloging or circulation system, there are no right or wrong answers to such questions as "How many terminals are required?" "Are remote communications lines necessary?" "Does the output have to be in machine-readable form?" In fact there is often no one answer in a specific library setting. Decisions can be based on tradeoffs or on specific constraints. These decisions are based on the system requirements drawn up during the systems analysis and systems design phases that precede this task.

System Requirements—One End of the Bridge

As stated repeatedly in library literature and in this book, systems analysis and design form the keystone of a successful automated system. At present they often are performed inadequately. The details of systems analysis cannot be covered in this book. The suggested main reference to follow is Markuson et al., *Guidelines for Library Automation*,[3] which was written specifically for Federal libraries.

Markuson presents the major phases of systems analysis and the tasks involved as follows:[4]

1. Perform preliminary planning and preparation
    - Inform library staff of plans
    - Determine objectives and purposes of system study
    - Survey relevant library automation literature
    - Prepare project schedule, allocate funds, and assign staff
    - Establish documentation standards and procedures
    - Prepare project proposal
2. Analyze present operation
    - Perform descriptive analysis
    - Flowcharts
    - Decision tables
    - Questionnaires, case studies, etc.
    - File analysis: files, records, data fields
    - Investigate conditions imposed by outside environment
        - Agency and local agency
        - Library community
        - Others
    - Identify operations requiring human judgment
    - Identify exception situations
    - Identify management and performance data requirements
3. Define system requirements for automated operation
    - Establish input and output requirements
    - Establish operational and human factors
    - Establish staffing, funding, and time requirements
4. Produce project report
    - Document findings in a project report
    - Review findings with appropriate agency management
    - Report on results to library staff.

The first phase is basically a review of the situation and an assessment of what the problems are. It must be pointed out that an automation project should be directed at a problem or need. No one should think of automating simply because it is fashionable or because equipment is available. Automation should be the means, not the end. Some genuine needs for automating can be classified as:[5]

- Increased volume of activity
- Need for improved control over operations
- Need for improved control over collections

---

[3] Barbara Evans Markuson et al., *Guidelines for Library Automation: A Handbook for Federal and Other Libraries* (Santa Monica, Calif.: System Development Corporation, 1972). Another basic library text is Edward A. Chapman, Paul L. St. Pierre, and John Lubans, Jr., *Library Systems Analysis Guidelines* (New York: Wiley-Interscience, John Wiley & Sons, 1970). For a collection of essays on and a bibliography of library systems analysis, see John Lubans, Jr. and Edward A. Chapman, *Reader in Library Systems Analysis* (Englewood, Colo.: Microcard Editions Books, 1975).

[4] Markuson et al., *Guidelines for Library Automation*, p. 26.

[5] Ibid., p. [illegible]

- Need for improved service to users
- Need to provide new services
- Need to prevent duplication of effort
- Need to operate within existing staffing patterns or conditions.

The specific needs must be stated and formal system objectives written. These objectives should be stated broadly, then restated as specific objectives.

The second phase involves an analysis of the present system. Systems analysis is defined as "the study of all of the components, operations, data, information and material flow, work environment, etc., that constitutes the existing system."[6] To facilitate this process, various techniques such as flow charting, decision tables, surveys, questionnaires, interviews, case studies, and cost analyses are used. In general, systems analysis requires systematic attention to the following items.[7]

Data. These are the files, records within the files, and fields within the records.

Operations. These are the tasks that are performed in accomplishing some functional objective. Operations include filing, posting information to a file, searching, labeling, etc.

Materials Flow. This component concerns the physical entities with which the library deals. Materials include books, serials, maps, microform, laboratory notes, etc. The materials flow includes receipt, processing, storage, retrieval, and use.

Information Flow. This component comprises all of the communications that relate to how data and materials are to be handled and how operations are to be performed and, as used, means information about the system and does not refer to content of materials. It includes policies, regulations, oral communication, forms, reports, and statistical data about the system.

The third phase is to identify and define the system requirements for the new system. "System requirements are the set of tasks and characteristics that the new system must perform and have."[8] The systems requirements are based on the *demands* (for information, reports, and action) on the system from all sources.

Three cautions must be stated about the systems requirements. First, while the new system requirements will be similar to the present system description, they should not be a reiteration of the existing system. It is a mistake to automate a manual system *in toto*. The demands and requirements should not even be the same. The problem or need on which the new system is based will trigger some different requirements. The existing system has taken on characteristics that are the results of constraints that have been met or pragmatic adjustments that have been made. The new system requirements should be free of these "systems-bound" requirements and should be an ideal "shopping list."[9] The new requirements should also be drawn up free of existing requirements that are tradition-bound or unnecessary.[10]

The second caution deals with the nature or level of the requirement. The system requirement is still expressed in library terms, "need" terms as opposed to "solution," or automation, terms. For example, the statement "daily access to the order file" would be an appropriate library requirement but the statement "online file access" would not be — it steps over into the "solution," or data processing, area.

The third caution has to do with the view of the library. Libraries rarely exist as independent organizations. They therefore must be viewed in their places within larger structures. The library applications requirements are often the first things considered, but requirements imposed on the system from outside the library must also be considered. For example, Markuson lists these aspects of the outside environment that must be considered:[11]

- Agency budget requirements
- Agency personnel or payroll records (useful for circulation or master user files)
- Agency headquarters library policies
- Agency purchasing requirements
- Library of Congress Card Division requirements for card purchases
- Vendor requirements for purchase orders
- Agency statistical reporting requirements:

Subject to these three cautions, the task of preparing system requirements entails considering the following questions.

- What needs or demands must the system meet?
- What are the impacts (organizational, environmental, and/or other system) and the constraints (physical, organizational, technical, contractual,

---

[6] Ibid., p. 29.

[7] Ibid.

[8] Ibid., p. 40.

[9] Ibid.

[10] Chapman, St. Pierre, and Lubans, *Library Systems Analysis Guidelines*, pp. [illegible]

[11] Markuson et al., *Guidelines for Library Automation*, p. 11

resources) that must be considered by the new system?

- What are the functions that must be performed by the system, in terms of:[12]
  —what triggers each function?
  —what are the restrictions on each function?
  —what are the data elements used in the function?
  —what processing steps must take place in the function?
  —what volume of processing per period of time?
  —what is the frequency of execution of the function?

How these requirements are ordered or expressed is somewhat arbitrary. Markuson suggests a more general, less technical outline.[13]

1. Statement of general requirements
2. Scope of operations to be included
3. Statement of functional requirements, such as:
   Data elements to be included
   Data access points
   Status requirements
4. Statement of human factors requirements
5. Volume to be handled:
   Daily flow
   File size
   File maintenance
6. Statement of management data requirements, such as:
   Analyses needed
   Output products required
7. Statement of desirable features (can include operations, analyses, output products, etc.)

The following is a sample list of requirements for a circulation system.[14]

1. General requirements
   - Improve control over materials in circulation
   - Improve management data reporting
   - Improve control of books and unbound serials
2. Circulation operations to be included
   - Charging and discharging
   - Overdue control
3. Functional requirements
   - Data elements must include brief title, brief author, complete call number (including copy), date due, and borrower identification
   - Data access points to charge file (books in circulation) must include title, call number, and borrower
   - Charges and overdues must be identifiable on a daily basis
   - The system must not depend on conversion of the shelflist
4. Human factors requirements
   - Due to high turnover, the system must be designed so that it can be handled by relatively untrained clerks
   - Equipment must not be noisy and must not require redesign of circulation area
   - Output products should be legible and convenient to scan
5. Volume to be handled
   - Daily: 1,000 charges and discharges, 50 overdues
   - File Size: 40,000 maximum for circulation file; 5,000 maximum users
   - File maintenance: Outside of discharging, corrections occur at a rate of 10 per week
6. Management data requirements
   - Daily tally of overdues and charges
   - Monthly and yearly summary of overdues and charges
   - Desirable but not mandatory: Monthly tally of loan activity for every borrower and average of all borrowers
7. Desirable features
   - Handling of interlibrary loans and pertinent activity summaries
   - Analysis of loans by broad class categories
   - Inclusion of temporary borrowers
   - Lists of items in circulation on a daily basis with weekly cumulation
   - Use of automatic charging device and permanent borrower cards
   - Integration of system with agency payroll account number
   - Handling of reserve requests

Systems Design—The Other End of the Bridge

The system requirements as just described are the basis for the systems design specifications, which describe the nature of the system to be implemented and how it is to operate. These are expressed in a design document that serves as "the blueprint and point of reference for all subsequent implementation work. It provides all the information necessary to solicit bids, if equipment or outside assistance is needed, and to develop work assignments for library or other 'internal' personnel . . . . It must contain all the information necessary for the initiation of computer program design and coding, e.g., flow charts, data formats, file design, and operational procedures."[15] This final document is the result of the systems design phase, which "is concerned with formulating various systems that meet some or all of the requirements developed during the analysis phase. It also includes the critical review of the postulated alternative systems, the recommendation of the 'best' system, and detailed design of the selected system."[16] The following list shows the steps of systems design.[17]

---

[12] U.S., Civil Service Commission, Bureau of Training, ADP Management Training Center, "Management Introduction to Automated Data Bases," Washington, D.C., n.d., p. A-5. (Mimeographed.)

[13] Markuson et al., *Guidelines for Library Automation*, p. 12.

[14] Ibid.

1. Formulate initial systems design
   - Develop flowchart or functional diagrams and tentative system specifications
   - Conceptualize and study alternative system approaches
   - Refine system specifications
   - Obtain approval of initial design
2. Establish hardware specifications
   - Identify processing and offline storage requirements
   - Identify input and output characteristics and requirements
   - Identify any needed modifications for library application
   - Identify essential components at various levels of system implementation
   - Investigate procurement of equipment
   - Estimate lead-time requirement for obtaining equipment
3. Establish software specifications
   - Identify computer programs required
   - Investigate general-purpose programs suitable for system use
   - Develop specifications for program language to be used
   - Develop program documentation requirements
   - Investigate capabilities of programming staff available to library
4. Analyze costs
   - Analyze present operational costs — men, machines, material
   - Analyze projected costs of automated operational system — men, machines, material
   - Compare costs at projected levels of operation
   - Analyze transition costs — acquisition of automated system and costs of maintenance of dual systems during system implementation
   - Identify cost of conversion of essential files
   - Identify costs of training existing personnel, preparation of manuals, code books, etc
5. Assess changes resulting from automation
   - Identify alteration of routines, provision of new services
   - Describe effect on present staff organization
   - Consider increased capability, changes in unit costs, etc
   - Consider potential of phasing contemplated system into proposed agency, local, state, or regional systems
6. Investigate management problems
   - Investigate problems of managing future operation
     Contractor facility
     Use agency computer facility
   - Investigate amount of computer time available to library and schedule for services
7. Report on system investigation results
   - Prepare two reports: general report for library staff, institution management, etc.; detailed report for further implementation activities
   - Prepare displays, charts, etc., covering major points for staff and other briefing sessions
   - Report on investigation to library community as appropriate
   - Develop schedule for further implementation plans
8. Perform project reporting and review
9. Obtain approval for further system implementation

The systems design phase should be the most creative phase of the automation process. It must be firmly based on the system requirements and on experience gained from the description of the existing system, but it can look beyond existing methods, procedures, or operations. The system design is best done by a team consisting of a librarian and data processing experts. Working together, the team can conceptualize and study alternatives from both library and data processing standpoints. The best system will meet both data processing needs and library needs without undue distortion of either.

---

[15] Ibid., p. 56

[16] Ibid., p. 43

[17] Ibid., pp. 44-45

It is during the system design phase that the bridge can be used. In the following section the characteristics of minicomputers will be explored in terms of library applications system requirements.

LIBRARY APPLICATIONS

GENERAL

In the systems design phase a manual system may appear to be the proper solution. Although almost anything can be automated to one degree or another, not everything should be.[18] Manual activities, such as moving materials, shelving books, and opening packages are not suited for automation in a library, as they might be in a warehouse. The intellectual activities of setting policies and making decisions can be supported by automated statistical analyses, simulation, and modeling, but except for those simple decisions that can be reduced to algorithms, the decisions themselves cannot be automated. The same can be said for the personal services of a library. Automated products and operations can aid a reference librarian in the one-to-one interaction with a user, but they cannot completely replace the human element.

The clerical activities of the library are the best candidates for automation. These are typified by recordmaking and keeping, data manipulation (such as sorting or searching), and data control. Even in the clerical areas, not everything should be automated. In many cases, the most efficient system is not one that "does it all," but one that does the routine or standard tasks while leaving the exceptions to be done by human beings.[19]

If automation is appropriate, it can be accomplished by a large-scale computer system, network, time-sharing service bureau, minicomputer system, or combination of these. Some applications, some user needs, and some sizes of libraries typically call for certain types of systems, but the options should be left open.

There is no one type of library or size of library that should or should not automate. It was once axiomatic that no library with a collection of fewer than 100,000 should automate, but Swihart and Hefley suggest that the minimum size for an independent system is 25,000 titles, which brings many more libraries into consideration.[20] Even that number is not inviolate. Sometimes the organizational structure of a library and the structure's complexity can dictate automation, as can the structure of the parent agency in which the library resides. Staffing problems can suggest automation as a solution. For example, either a hiring freeze that results in a lack of clerical support due to attrition or a one-professional shop that requires careful allotting of time can justify an automated system to absorb part of the housekeeping load.

Minicomputers themselves have opened up both more libraries and more library activities to automation. Minis are small, simple, and modular, and they can be used in smaller applications. Minis are ideally suited as single-purpose, dedicated systems. It is conceivable that several mini systems serving a library could be more economical than a single computer system trying to provide the same service. Using a system of several minis allows a library to introduce automation one module at a time. The separate modules could be connected but would remain basically independent. Minis, because they are so flexible and well suited to small-scale applications, can be used in connection with other automation systems as complements or supplements to such systems, or as components. Even libraries that already have automated systems could consider potential mini applications.

In looking at library applications, the major operational areas of the library (cataloging, circulation, serials, interlibrary loans, acquisitions, reference, SDI, and administration) will be used as points of reference. Keep in mind, however, the various approaches to automation reviewed in Chapter Two. Consider also that in any one library the various functions performed can be placed under a number of departments. For example, serials can be part of cataloging, part of acquisitions, or a separate unit. Interlibrary loans can be part of reference, part of circulation, or completely independent. It is for such reasons that the specific descriptions in this book are only illustrative and are not detailed models for imitation.

Markuson and her collaborators reviewed the major library areas and developed a series of lists of functions, involved in the performance of the tasks of these areas.[21] These function lists illustrated that there are "recurring basic functions and tasks: record preparation, record input, filing, file maintenance, searching, materials handling, etc." This approach goes beyond looking at the area on a step-by-step basis as plotted in the flow chart and helps the design to be solution-oriented, rather than merely a replication of the existing system.

---

[18] Ibid., p. 70, for additional discussion.

[19] Cox, Dews, and Dolby, *Computer and the Library*, p. 1.

[20] Stanley J. Swihart and Beryl F. Hefley, *Computer Systems in the Library: A Handbook for Managers and Designers* (Los Angeles: Melville Publishing Co., 1973), p. 10.

ACQUISITIONS

Markuson and her coworkers saw acquisitions systems as centered around two basic groups: (1) fund accounting activities that deal with the "management and control of fund allotments and expenditures and related functions," and (2) order control activities that entail "machine-readable input of order information and the generation of a variety of output products."[22] The Columbia University acquisitions project established "three levels of data flow, namely, (1) process control, or the flow of order data and the processing status of each order through the system; (2) fiscal data flow, or the encumbering of funds, paying of invoices, and reporting of fund status; and (3) bibliographic data flow, or the assembling, verifying, storing, and transferring of bibliographic data needed in the cataloging process . . ."[23] In either case, the following list of acquisitions functions and subfunctions is useful as a summary.[24]

1. Establishment and Surveillance of Policies and Procedures
   Policy development
   Maintenance of procedure manuals
   User feedback analysis
   Performance analysis
   Establishment of procurement sources (vendor files, blanket order agreements, etc.)
   Interlibrary cooperation
2. Fund Control
   Allocation of fund allotments
   Fund encumbering
   Invoice clearing
   Voucher preparation
3. Materials Selection
   Review of, and selection from, notices of potential items
   Preparation of purchase requests
   Approval of purchase requests
   Identification of desiderata materials
4. Order Preparation and Control
   Screening and distribution of purchase requests
   Searching and completion of bibliographic order data
   Vendor and fund assignment
   Order approval
   Order form preparation and file control — monographs, serials, gifts and exchanges, etc.
5. Materials Handling
   Material sorting and distribution
   Routing
   Control of items through processing
6. Receipt processing — monographs, serials, etc.
   Item verification
   Invoice verification
   Claiming
7. File Input and Maintenance
   Record input preparation and revision
   Error correction
   Transaction control: additions and deletions
   Use of data from outside source
8. Output Generation, Dissemination and Reporting
   Preparation of order forms, cancellations, claims
   Printing of lists of items on order
   Output of change or control cards
   Preparation of preliminary catalog copy
   Preparation of accessions lists
   Dissemination of order lists, dealer catalogs, SDI notices, etc.
   Dissemination of products
9. Gifts, Exchanges, Memberships, Vendors, and Other Sources
   Control of gift sources
   Control of exchange partners
   Control of memberships
   Control of vendor agreements
   Maintenance of vendor and other source files
10. Reference and Retrieval
   File searching
   Retrieval of items in process
11. Processing Records from Outside Sources
   Selection of records identified for purchase
   Selection of records for potential interest
   Processing and maintenance of outside data base

---

[21] Markuson et al., *Guidelines for Library Automation*, p. 71.

[22] Ibid., p. 92.

[23] Richard Phillips Palmer, *Case Studies in Library Computer Systems* (New York: R. R. Bowker Co., Xerox, 1973), p. 152.

[24] Markuson et al., *Guidelines for Library Automation*, pp. 97-98.

- Modification of records for local use
- Dissemination of hardcopy records
- Analysis of subject coverage

The major activities of the *establishment and surveillance of policies and procedures* function are intellectual in nature. Automation in this area will generally be limited to support. Data from other automated areas (for example, the management information system) can be provided on which to make decisions; analyses can be performed by computer (for example, statistical procedures as part of the surveillance of the system); and procedures manuals can be updated by a text-editing computer system.

Fund control is a natural area for automation. Bookkeeping has been automated in business and industry for some time because this reduces redundant manual entries and improves accuracy. The information on each order or transaction must be kept current, manipulated many times, and used for a number of purposes. Done manually, this job often requires many files or at least many massages of the file. The capture and control of the order information is well suited to automation.

Materials selection is not even considered acquisitions in some libraries. However, the selection area is where the initial capture of the bibliographic elements occurs, so it does have an impact on acquisitions. This function also uses information from the acquisitions system: Is this book already in the collection? Is it already on order? Is it for sale from a regular source? Is there enough money to cover its cost? So, although the final selection probably will be a human decision, automated processes and files can support this function.

Order preparation and control is a mixture of clerical and intellectual activities; the extent to which it can be automated varies. For example, searching and completion of bibliographic order data could be automated if the *Books in Print* Data Base in machine-readable form were available, or if MARC tapes were available to search, or if a jobber's file, such as BRO-DART's IROS system, were available online. Without these data bases, this function could not be effectively automated. Vendor and fund assignment can be automated if the rules for the decisions can be standardized and reduced to algorithms. (If the decisions are subjective or intuitive, they must be made by human beings.) Actual preparation of orders has been successfully automated, as has control of the files the orders generate.

Materials handling is largely a manual operation, although automation could support such a function as routing. The control of the item through processing is where the automated system can best perform. For example, if a status code is carried in the record for each item and updated as the item travels through the flow, the item always can be located and retrieved. To do this manually requires many files and/or much refiling.

Receipt processing entails physically handling materials and making individual judgments based on the original order information. Although human judgment is required, the data required to make the judgment can be supplied by an automated process. Followup activities connected with claiming also are tied to the order subfunctions. This function also provides information for other subfunctions such as fund control invoice verification.

The remaining functions on the list can be considered as subfunctions of the other areas. They could all be automated as parts of other automated procedures.

The use of minicomputers as part of a new system should be determined by specific problems or needs of the library. For this discussion, however, a number of typical problem areas in acquisitions can be listed.

- Maintenance of the vendor source file
- Control of standing orders
- Control of a depository collection
- Control of items in process
- Preparation of orders and status control
- Bookkeeping and accounting
- An agency accounting system with which to interface
- Control of a processing center serving multiple units
- Integration of purchase request items selected from MARC tapes
- Loss of service from the present large-scale batch system
- Control of preview and purchase of audiovisual materials

All of these areas can be supported by a minicomputer system.

To illustrate the use of the design model (the bridge), one of these problem areas will be developed as a case study and "walked across" the bridge. In lieu of the specific systems analysis documentation that would be used in a specific library, an expanded, detailed description of the acquisitions functions will be used. Table 15 describes the subfunctions, operations, relationships, files, inputs, and outputs.

Table 15—Acquisitions: Detailed Functions and Operations

Function & Description	Operations	Relationships	Files and Description	Inputs	Outputs
ESTABLISHMENT AND SURVEILLANCE OF POLICIES AND PROCEDURES					
Policy Development *Decisions as to selection criteria and procurement policies.*	Selection criteria is an intellectual effort based on professional judgment and organization objectives. Procurement decisions may have to be based on regulations imposed by the parent agency (For example, policy may dictate using only jobbers, no jobbers, competitive-bid contractors; all purchase orders, no purchase orders, blanket purchase orders, standing orders, etc.).	Relates to parent organization. Input from User Feedback. Analysis as potential for policy changes. Input from Performance Analysis considered.	May involve a collection of manuals or regulations for the parent agency.		
Maintenance of Procedure Manuals *Details of how to carry out policy decisions, expressed as step-by-step procedures for staff*	Each procedure must be expressed in writing, with appropriate forms and attached samples	Covers the entire acquisitions flow Will relate to other functions at specific points of interface.	Must be in a written form (duplicate copies desirable). Must be updated as necessary to keep current.		
User Feedback Analysis *Information from users on any phase of collection building or maintenance.*	Consideration of comments on what is purchased, time or speed of providing, availability, etc. Alterations in policy as required.	Can provide input to Policy Development	May be solicited through a survey or may be spontaneous.		
Performance Analysis *Management information used in evaluating performance and forming decisions*	Gathering of data; statistical treatment and analyses of data, interpretation of results for management	Related to Control of Item Through Processing function, Claiming function, Vendor transaction file (or equivalent)	Specific files must be structured to record volume, time spans between events, etc. May be automatically kept or run through specific program (tally) and gathered on a periodic basis. Parameters required must be established in advance.		Variety of hardcopy reports, charts, and graphs.

BEST COPY AVAILABLE

Establishment of Procurement Sources *May involve formal contracts or internal rules for choice of vendor.*	Decision on which, if any, jobbers and for what types of materials. Decision on blanket purchase orders (with whom and how many). Decision on establishing deposit accounts. Decision on approval plans—profiles, etc. Determination of direct sources—publishers, societies, etc.—and special purchase requirements. Decision on standing orders—by title.	May be based on Policy Development decisions. Related to Maintenance of Vendor File function. Related to Vendor Assignment function. Related to Control of Vendor Agreements function.	May have a file of RFPs, bids, contracts, purchase orders, special agreements, etc. Status, length of agreements, dollar amounts may have to be monitored here.		Information to Vendor File Maintenance function.
Interlibrary Cooperation *Formal or informal arrangements may be made between libraries, i.e., exchange programs or consortium memberships.*	A formal consortium agreement or informal agreement on what subject areas of collection building or journal holdings to share may affect selections.	Related to Policy Development and Selection of Materials functions.			
FUND CONTROL					
Allocation of Fund Allotments *Budgeting and allotments by subaccounts.*	Maintenance of balances in each subaccount. Reconciliation of encumbered and paid amounts.	Direct connection to ledger (if a separate file). Accessed for balanced status by Order Approval function.	Budget file may be separate from ledger or combined.	Fiscal year budget amounts. Any budget transfer amounts from ledger (if a separate file).	
Fund Encumbering *Bookkeeping ledger operation to record each order and link to each fund.*	Maintenance of a bookkeeping ledger for accounting of financial status.	Direct connection to budget account (if a separate file). Amounts to encumber, funds involved, and order number/vendor number come from the Order Approval function.	Ledger: Must be accessible by fund account number, order number, vendor number. Contains alphanumeric data, but no bibliographic data. Requires simple computation. Will be added to and changed constantly (three to four times per record).	From Order Approval function.	Hardcopy status reports on a periodic basis.
Invoice Clearing *Record of payment of each invoice against each order, reconciliation of differences.*		Amounts to pay come from the Invoice Verification function.		From Payment Control function and/or Invoice Verification function.	

Table 15—Acquisitions: Detailed Functions and Operations—Continued

Function & Description	Operations	Relationships	Files and Description	Inputs	Outputs
FUND CONTROL—Continued					
Invoice Clearing—Continued			Currency of prime importance (lag can't be more than one day). Status of totals required continuously. Must keep historical data (dead or complete), which also will be accessed.		
Payment Control *Authorization of payment by check or voucher*	Performance of Invoice Clearing function by combining the Invoice Verification and Ledgering functions by order number	Direct connection to Ledger. Related to Invoice Verification function. Must have access to the vendor file.	If a separate file, would contain order number, vendor number, invoice numbers, check or voucher number, date. Must be accessible by order number, vendor/invoice number, check number. Numeric, no bibliographic data. Little or no update or status check required. Must allow for bank statement reconciliation. Additions to file in batch mode acceptable.	Invoice Verification function, input (invoice number and amount tied to order number).	Written check OR Hardcopy voucher OR Machine-readable approval to an agency system.
MATERIALS SELECTION					
Review of, and Selection from, Notices of Potential Items *Involves use of bibliographies, review tools, publishers catalogs, alerting tools (manually printed)*	Available titles are reviewed often by a team of people representing various subject disciplines (can be by users outside the library).	Direct connection to next step.			
Can be an SDI list prepared from a machine-readable file (such as a MARC tape)	Some SDI systems prepare lists from a profile of the user, which is then reviewed by the user			Can be from MARC tapes or BIP tapes. Can be part of BRO-DART IROS system or Baker and Taylor's BATAB system, etc.	
Can be an online scan of a machine-readable data base (like BIP tapes or BRO-DART IROS system)	May be gift or exchange items available to the library				

Preparation of Purchase Requests *An individual record for each title.*	All available bibliographic data identified for each title is captured, along with information as to source of request and priority (when needed).	If from a machine-readable tape or data base, this can be used to produce the record (hard-copy and file).	Manual (paper) file of individual records: • bibliographic data • selection source data • order data if available OR Machine-readable file of the elements available from the source file: • bibliographic data • order data (e.g., price, ISBN)		An individual record for each title requested.
Approval of Purchase Requests *Each title is reviewed to determine whether to purchase or—for gifts and exchanges—to accept.*	A judgment (usually performed by a professional librarian) as to which titles to buy. Decision criteria are available, but algorithms are difficult to prepare.	Can be performed after the File Search function, which determines if title is a duplicate (local library option).	A sort of purchase requests into two files, one of selected titles and one of rejected titles. Rejected titles may be returned to selector for notification.		A group (or file) of records which goes to the Order Preparation function.
Identification of Desiderata Materials *Titles to be held on a "want" list*	Titles that are not in print or available for purchase at request time are reviewed to determine if they should be purchased through a remainder dealer, out-of-print catalog, reprint list, or book exchange list. This is a costly procedure, so only selected titles are put on the "want list." Review of these special tools can be in Selection function or Order function.	This is the next step after Searching and Completion of Bibliographic Order Data for those titles determined not to be available. This also follows the Claiming function for titles reported as "out-of-print" or "not yet published." Goes back to Order function for review of status. Goes back to Order function when source is identified.	A file (list) of titles: • bibliographic data • order status • search trail and history, i.e., catalogs checked and when.	Record of a title from the Order function.	Record back to the Order function when source is identified.

Table 15—Acquisitions: Detailed Functions and Operations—Continued

Function & Description		Operations	Relationships	Files and Description	Inputs
ORDER PREPARATION AND CONTROL					
Screening and Distribution of Purchase Requests *Sort by type (monograph or serial), foreign or domestic, etc., to determine who and how to search*	Some libraries separate orders by type: monograph, serial, document, microform. Some libraries separate by foreign and domestic sources. Intellectual decisions that could possibly be machine-edited (algorithms) but would have to be human-reviewed.	Next step from Approval of Purchase Request function. Connects to next step	Sort by categories (locally determined). No file as such. Determination of whether duplicate requests are present; if so, they are gathered.	Purchase request records (manual or machine-readable).	Purchase request records (sorted).
Searching and Completion of Bibliographic Order Data *Verifies and completes the author, title, edition, imprint information, adds data on ISBN, cost, etc* *Determines if title is currently offered for sale*	Current tools, such as BIP, PBIP, FC; publishers' catalogs; document lists such as MC, GRA, etc., exchange lists, reprint lists, microform catalogs, out-of-print catalogs are used to complete and/or verify the titles desired. Some libraries require a list price, others order on an open price	Related to decisions made in Establishment of Procurement Sources function	If records were previously in a machine-readable file, the entire record is subject to changes and additions. BIP et al. searches are done by author or title; records should be so ordered. Publishers' catalogs are by vendor first, subsorted by author or title. Online files such as BRO-DART's IRCOS are accessed by author/title search key.	Purchase request records (sorted).	All titles not found to be available are sent to the Selection function for desiderata decision.
Vendor and Fund Assignment *Determines which fund or subaccount to charge the purchase to.* *Determines whom to purchase the title from if there is a choice*	Parameters for decisions can be drawn up and algorithms used Choice of jobber or direct purchase Fund accounts can be internal to the library (general collection, reference, periodicals), can be charged to a program (PPBS) or objective (MBO), can be charged to a department or section of the agency or parent organization	Related directly to Establishment of Procurement Sources function Access to vendor file to establish special imprints, conglomerate relationships, and distributorships. Also special ordering restrictions, e.g., pro forma, minimum orders, purchase orders required, membership required	Vendor code and fund account numbers are added to each record. All titles for same vendor should be brought together.		

Order Approval [illegible]	Balance of each fund to be charged against must be checked for sufficiency, then encumbered against when order number is assigned. Order number assigned. Signatures given if required.	Can be performed after the File Search function that determines whether title is a duplicate, if not part of Selection process (local library option). Accesses fund balances and gives encumbrance information to Ledger.	If still in manual form, computer input should occur here. Records must be at the individual title level. Bibliographic data and order data. ●access by: author and/or title, order number, vendor number. ●serves as an on-order file, or is combined with an in-process file with "on-order" status code or is combined with main catalog file.	Balance information for each fund involved.	(Data transfer) Amounts to be encumbered for each fund (includes order number, vendor number). Some require printouts of on-order list. Some use prepunched cards for book on order or journals due in Receipt Processing function. Orders printed ready for mail.
Order Form Preparation and File Control	Preparation of individual orders ready for mailing.	Related to Receipt Processing function. Pre-cataloging may occur (local library option).	Order file and/or vendor file of orders may be required: ●access by order number and vendor number ●must carry details of each title on order ●will be accessed for individual status on a title and order level ●updates on status will occur at any time ●currency is important ●entire fiscal year history must be maintained.		
MATERIALS HANDLING					
Material Sorting and Distribution *Separation of incoming materials from all sources by destination.*	Manual. Sorting of incoming mail, packages, shipments. Routing to specific check-in points. Ex.: monographs are generally checked in separately from periodicals.	Will connect to Receipt Processing function.	No specific file created. Materials can be stamped with arrival date.	Comes from U.S. mail, parcel delivery, truckers.	
Routing *Generally dealing with newspapers and periodicals that are sent directly to users according to a predetermined list by title.*	When a routing list is established in advance by journal title, the list is matched to the copy, affixed, and sent off. Comes after issue has been formally checked in.	Related to and comes from Receipt Processing function.	Profile of users (or departments) that are to receive each title, often in a priority sequence. Must change with change in personnel.	Preprinted lists of names on routing list by title.	

Table 15—Acquisitions: Detailed Functions and Operations—Continued

Function & Description	Operations	Relationships	Files and Description	Inputs	Outputs
MATERIALS HANDLING—Continued					
Control of Item Through Processing *Must be able to locate a specific title, if needed anywhere along the processing flow.*	Each title must carry an active status to show where it is in the processing flow. This will change as it goes through each step. Can be a series of files or a master file with status updated. Includes binder control and routing control for periodicals. Ends when material is on shelf and represented in catalog ready for public use.	Related to Order File, Item Receipt function, Cataloging section, Processing section, and Circulation section. Related to Performance Analysis function. Related to Retrieval of Items in Process function.	File of individual title records with full bibliographic data and some order information. ●must access by author, title, order number ●must search and list by status: all books on order; all books received, not cataloged; all books cataloged, not labeled; all books labeled with no circulation control data ●searches may have parameters applied. May be separate files by function. May be part of main catalog. Will be a large, active single file, often accessed, often updated and changed. Currency is important. Input must be as efficient as possible due to great volume.	May be actual paper cards/work slips to indicate status. Ex.: prepunched card sent forward on arrival of title. May be output of other functions.	Potential hardcopy lists on demand and/or periodically.
Routing continued	Analyses may be required. ●statistical counts ●time flow plots.				
RECEIPT PROCESSING					
Item Verification *Checking and verification of each item received against its order or list. Checking also for damage.*	Materials come from a single source (e.g., vendor) and may be unique to a single order number. Each title is examined against bibliographic data of order to verify correct title, edition, etc. If correct, title is flagged as received. This updates in-process status and eliminates duplicate shipments from vendor. If approval plan is used, a special	Must have access to order, subscription, exchange list, etc., on which item was initiated. Must update order status in Order file (vendor transaction file). Can be performed concurrently with Invoice Verification (next step). May require Claiming function.	If a vendor transaction file, the order number and specific titles must be accessed and updated. Must determine if order is complete or not. If an order file, the specific title must be updated—order may or may not be complete. If an on-order file by item, the title must be accessed and status flagged. May be considered closed or moved to inactive status. If an in-process file	Material comes from Materials Handling function.	

	procedure will be required.		by item, the title must be updated to change status.		
Invoice Verification *Checking of each amount and item on an invoice to lead to approval for payment.*	Must verify receipt of proper item with no outstanding claim. Must verify that charge matches item (both paper, both same number of copies, etc.). Must verify not prepaid. Must verify proper discounts applied if eligible.	Must be closely tied to Item Verification function. Related to Fund Control, Invoice Clearing function.	Establishes exact cost of each item purchased, which must be recorded, as well as miscellaneous charges such as postage, handling, and tax. If o[illegible]le is not updated [illegible]re (by title), it must be updated in Invoice Clearing function (best here because a "by item" approach is more efficient here).	Invoices may come separately by mail; must be held for matching. May require credit memo to match against invoice.	
Claiming *To resolve order problems such as no item received, wrong item received, damaged item received, incorrect invoice, no invoice received, partial receipt of order.*	Requires correspondence (or telephoning) with vendor. May require vendor's permission to return, before shipping. Requires control of pending condition for an item, invoice, and order.	Connected to Item Verification and Control of Item Through Processing or On-Order File function. Must access Vendor File.	Must have complete order, vendor, bibliographic, and status information. Must have a pending/follow-up capability for each claim; access by vendor number, order number, item (author/title), date, category of claim. May not be separate, but rather recorded as a status or condition in another file(s).	Some vendors provide claim forms to use, including blank credit memos for use as needed.	Letters or forms to vendor: ●permission to return ●request for missing invoice ●request for status of a title and/or an order ●request for credit memo ●request for missing copies. Mailing labels.
GIFT, EXCHANGE, MEMBERSHIP, VENDORS, AND OTHER SOURCES					
Control of Gift Sources *Includes philosophy and mechanics of gift procedure.*	Must consult gift policy as to kinds of gifts, dollar amounts, numbers, and conditions or "strings" attached. Must determine if a tax form will be provided.	Related to Materials Selection function. Related to Fund Control function to record dollar value for reports and insurance.	Must maintain name and address of donor; number, type, and dollar amount of gifts accepted or rejected; may require a list by title.		
Control of Exchange Partners *May be a membership in a cooperative or on an individual basis.*	May be a membership in the U.S. Book Exchange or in[illegible]al arrangement. May involve [illegible]. May involve [illegible] of lists or re[illegible] boxes. May involve creation of lists.	Many, but will vary with nature of agreements.	Must maintain: bibliographic information ●on titles offered ●on titles considered ●on titles selected financial data ●on titles sent off ●on titles selected transaction data on each shipment for control.		

Table 15—Acquisitions: Detailed Functions and Operations—Continued

Function & Description	Operations	Relationships	Files and Description	Inputs	Outputs
GIFT, EXCHANGE, MEMBERSHIP, VENDORS, AND OTHER SOURCES—Continued					
Control of Memberships *May be library initiated or agency initiated.*	Some fees are charged to a special fund (either library or agency). Some memberships are required to purchase materials. Some memberships provide free materials automatically.	Related to: Fund Control Vendor Assignment Vendor File Materials Selection Control of Item in Process Retrieval of Items in Process etc.	Must have organization data (name, address, cost of fee, length of membership); must access by name and time of expiration. Must have profile of services: ●purchase arrangements, e.g., discounts ●free titles, automatic or by request.		
Control of Vendor Agreements	Details of each agreement must be kept as to nature, requirements, profiles, dollar levels, length of agreement. Types: Blanket purchase order—one vendor, dollar ceiling. Approval plan—frequency of review, profile, shipments, returns, payments, discounts. Deposit Accounts—one vendor, dollar ceiling. Jobber Contract—level of commitment, length of contract; may have dollar minimum; discount may vary on volume. Standing Orders—by title in a vendor/frequency sequence.	Many, but will vary with nature of agreements. Related to Establishment of Procurement Sources function.	Agreements with dollar levels must be encumbered, paid against, and reconciled with each transaction. This status must be accessed in the Order Approval function. May be part of regular order control flow, with access and reporting available as above.		Reports—periodic or on request—on levels and status.

Maintenance of Vendor and Other Source Files	Control of a master file of information on each vendor, jobber, society, etc. (procurement source) that is an actual or potential source. Reflects the relationship of the book trade industry; profiles of each vendor; specific arrangements with individual library.	Accessed by: Payment Control function Vendor and Fund Assignment function Order Form Preparation function Claiming.	File of Vendors' Information: Control number Name, address, phone number, order-to address, claim-to address, ship-(return) to address Specifications: ●must have purchase order number ●must be prepaid ●must have minimum dollar amount per order ●must have minimum number of items per order ●must have membership ●approval plans available ●standing orders ●deposit accounts. Control over minor presses, imprints, publishing houses, distributorships (cross-references).

Case Study

Problem Area

An efficient means of handling cooperative acquisitions among branch libraries is needed.

Library Environment

A Federal agency has 14 branch or field libraries throughout the United States, as well as a main library at the headquarters in Washington, D.C. A central processing unit provides technical services for all of the libraries. Each library performs its own book selection, and the processing unit orders what is requested. About 4,000 volumes are acquired each year.

Goals

It is desired to coordinate the selection, acquisitions, and cataloging processes of the 15 libraries and the processing unit.[25] The intended benefits will be:

1. Professional expertise used in the book selection process will be shared among the libraries. Clerical efforts will be reduced by elimination of redundant searching, typing, and verification.
2. Collections will be better balanced. Expensive items may not be duplicated within the system and an interlibrary loan will be used if a book available at one branch is needed at another.
3. There is the possibility of earning a volume discount on purchases of multiple copies of a single title.
4. Cataloging will be more efficient if it deals with multiple copies of a single title at one time, rather than performing the added copy routine a number of times.

Proposed Flow

A library will research and select a title to add to the collection through the regular selection workflow (see the Materials Selection function). The bibliographic elements of the purchase request will be keyed, i.e., entered into machine-readable form. On a biweekly schedule, the purchase requests will be sent to the processing unit. The purchase requests from all the libraries will be merged and sorted (see the Order Preparation and Control function). A combined list will be sent to each library. The list will be reviewed by each library and additional copies desired will be indicated. The lists will be returned to the processing unit. The purchase requests will be merged and sorted by vendor, and the orders will be prepared.

System Requirements

1. Input captured in 15 different locations.
2. Processing performed in one central location.
3. Output must be prepared in 15 copies if hardcopy (for the list).
4. Output must be prepared on special forms if hardcopy (for the orders).
5. Record status in the file must be changed several times.
6. Turnaround time must be as short as possible.
7. Data must be sorted and reformated.

Alternative Designs

1. Punch cards are prepared by each library for each purchase request and sent to the processing unit. Cards are read, merged, and sorted. A list of purchase requests is printed with 15 copies, and one copy is sent to each library for review. Branches desiring additional copies mark the list accordingly. The list is returned to the processing unit. Input is keyed and the file is updated. Orders are prepared. (Turnaround time will be extended due to mail delay.)
2. Each library has a key-to-disk station. Purchase requests are entered. The floppy disk is sent to the processing unit. The floppies are read and merged. The complete file is sorted and written out on floppies which are returned to the libraries. The list on the floppy is reviewed, and input to the file (added copies to be purchased) is keyed. The floppies are returned and read, and the main file is updated. Orders are then prepared. (Turnaround time will be extended due to mail delay.)
3. Each library has an online, interactive terminal for entering data on purchase requests. The file is reviewed independently (online at the same terminal) by each library and updated on the spot. After a predetermined cutoff time, the file is sorted and orders are prepared.

The Bridge: Review of Design Characteristics and Their Impacts

The details of the library setting and the elements of alternative designs are compared to the design characteristics. Where a match occurs, the hardware and software impacts are considered. Table 16 is a summary of the three foregoing possibilities in terms of the "bridge" design model. Tables 17, 18, and 19 summarize the relevant design characteristics and the hardware and software impacts of the three.

---

[25] David L. Weisbrod, "Acquisitions Systems: 1973 Applications Status," in *Library Automation: The State of the Art II*, eds. Susan K. Martin and Brett Butler, papers presented at the Preconference Institute on Library Automation sponsored by the Information Science and Automation Division of the American Library Association at Las Vegas, Nevada, 22–23 June 1973 (Chicago: American Library Association, 1975), pp. 96–97.

TABLE 16—Design Characteristics of Each Application

Design Characteristics	General System Requirements and Specific Alternative Systems	Design Characteristics Pertinent to Systems
Inputting Nature of data Oneway/conversational Different types of devices Location of input stations Multiple online users Outside source Output Products Hard copy Special forms Multiple copies Display Combination hardcopy/softcopy Machine-readable Offline Online File Structure and Size Structure Sequential Direct Access Size Transaction/Volume Expansion Peaks Response Time Applications Characteristics Sorting/data manipulation Searching Batch Online Access points Multiple users Special input/output devices Remote access Response time Interface with Other Systems Offline Online Link to multiple systems	General System Requirements:	
	15 libraries perform data capture	
	One processing location	
	4000 volumes per year	
	Purchase request: bibliographic data, selection source data, order data (approximately 300 characters per request)	
	Orders typed on special forms	Output: hardcopy, special forms
	Data must be reformatted	
	Sorted lists required	Applications Characteristics: sorting, alphabetic
	Records must be updated	
	Alternative One:	
	Each library prepares punch cards	Input: one-way
	Center must be able to read cards	File Structure: sequential
	Output list prepared in 15 copies	Output: hardcopy, multiple copies
	Additional data capture at center from 15 lists to update file	
	Alternative Two:	
	Each library performs data capture on key-to-disk units	Input: one-way–offline, machine-readable
	Center must be able to read floppy disks	File Structure: sequential or direct access
	Center must be able to write output on floppy disks (15 copies)	
	Each library must be able to search, read, and update floppies	
	Alternative Three:	
	15 online, interactive terminals	Input: conversational—location of input stations—multiple online users
	Entire current file online	File Structure: direct access
	Online update capability	Application Characteristics: remote access

TABLE 17—Alternative 1

Design Characteristics	Hardware/Software Impacts	Library System Specifications	Minimum Class of Computer
Means of Inputting One-way.	I/O Device: input only. Ex.: keyboard, card reader, paper tape reader.	Keypunch machine at each library. Punch card reader for computer configuration.	
Types of Output Products Hardcopy: Special forms and multiple copies.	I/O Device: printer with special paper capabilities—probably an impact type to handle multiple copies.	Impact printer, probably a line printer for speed in printing 15 copies (i.e., three or four runs).	
File Structure and Size Structure: Sequential.	Mass Storage Device: any type can handle sequential. Ex.: punch cards, paper tape, cassette tape, magnetic tape, diskette, or disk.	Open (Class III dictates no disk).	
Applications Characteristics Sorting: Alphabetic.	Mass Storage Device: large working space required and access to several areas at one time. Tape system will require three drives. Disk system could have one large disk.	Optional (Class III dictates magnetic tape or diskette (if capacity is large enough).	III

TABLE 18—Alternative 2

Design Characteristics	Hardware/Software Impacts	Library System Specifications	Minimum Class of Computer
Means of Inputting One-way. Outside source: Offline machine-readable form.	I/O Device: key-to-disk station or intelligent terminal. System Software: floppy disk operating system.	[illegible] library. [illegible] drive for [illegible]er configuration.	
Type of Output Products Hardcopy: Special forms	I/O Device: printer with special forms capability	Printer, line or character.	
Machine-readable output form (offline).	I/O Device: same medium as input station, i.e., floppy disk. System Software: floppy disk operating system.		
File Structure and Size Structure: Can be either sequential or direct access.	Mass Storage Device: any type can handle. Ex.: cassette tape, magnetic tape, diskette, or disk. Size of file is the controlling variable.	Open (Class III dictates no disk).	
Applications Characteristics Sorting: Alphabetic.	Mass Storage Device: large working space required and access to several areas at one time. Tape system will require three drives. Disk system could have one large disk.	Optional (Class III dictates magnetic tape or diskette if capacity is sufficient).	III

TABLE 19.—Alternative 3

Design Characteristics	Hardware/Software Impacts	Library System Specifications	Minimum Class of Computer
Means of Inputting Conversational.	I/O Device: two-way (output or response capability).		
Location of input stations.	Communications Equipment: hardwire or telecommunication line.	Due to distances of the libraries, telecommunications lines and equipment must be used.	IV
Multiple online users.	I/O Device: two-way (conversational capability). Communications Equipment: telecommunications lines or hardware lines. System Software: must handle telecommunications functions if that method is used.	Terminal must have modem.	
Type of Output Products Hardcopy: Special forms	I/O Device: printer with special forms capability.	Printer, line or character.	
File Structure and Size Structure: Direct access	Mass Storage Device: disk device (diskette, fixed-head disk or movable-head disk). System Software: disk operating system.	Disk system.	
Application Characteristics Sorting: Alphabetic	Mass Storage Device: large working space required and access to several areas at one time. Tape system will require three drives. Disk system could have one large disk.		
Remote Access	Communications Equipment: hardwire lines or telecommunications lines and modem. Communications Controller: may require a processor.	Telecommunications system.	

It is apparent that not every design characteristic pertains to every application and that not every system requirement is translated into a design characteristic. However, in both cases each must be considered, and an individual judgment must be made as to relevance. For example, the size of the file in this case has no impact on the design. The 4,000 volumes would be represented by 1.2 million characters (300 characters per record). Not all would be on the file at one time (only the titles being ordered during the biweekly cycle). There is no file size limitation for punch cards, paper tape, or magnetic tape as mass storage techniques. There is a size limit (capacity) for floppy disks, but a single side holds 250,000 to 300,000 characters, which is large enough for this application.

In determining the lowest class of minicomputer system required to handle the alternative design, there are triggers that eliminate lower classes or dictate a minimum level. For example, Alternative One (Table 17) does not require a Class III system until the design characteristic of sorting is considered. Class III is the lowest level that can handle sorting. Alternative Two (Table 18) follows the same pattern. Alternative Three (Table 19) requires an online interactive system with telecommunications capability. The lowest level that can handle this requirement is Class IV.

The Analysis

The design model makes it apparent that the cruder, simpler Alternative One will not save any money by using a lower class of minicomputer. Both Alternatives One and Two require a Class III system as a minimum. The cost differences will be in the cost of input devices, the cost of the line printer, and the processing time and cost of paper supplies that production of 15 printouts will require. Class III computers can handle communications, so it is feasible to investigate connecting each input station to the computer to transmit data in a batch mode. Turnaround time will decrease.

CATALOGING

Markuson and her collaborators state that there are three distinct types of catalog systems: (1) listing systems, which mainly prepare output products such as catalog cards, book catalogs, and labels; (2) control systems, which support catalog operations in such areas as thesaurus control, shelflist inventory control, and file maintenance; and (3) search systems, which "provide mechanisms for accessing the catalog data by a number of search elements."[26] Markuson's cataloging functions are outlined in the following list.[27]

1. Establishment and Surveillance of Catalog Policies and Procedures
    - Policy development
    - Maintenance of procedure manuals
    - User feedback analysis
    - Performance analysis
    - Interlibrary cooperation
2. Establishment and Maintenance of Local Authority Lists or Adoption of Standard Lists
    - Name authority files
    - Subject authority lists; thesaurus
    - Other authority files
    - Classification schedules
    - Filing rules
    - Descriptive cataloging rules
3. Materials Analysis
    - Descriptive analysis
    - Author entry establishment
    - Subject analysis and indexing
    - Classification and reclassification
    - Abstracting
    - Preparation of initial catalog record
    - Revision and correction of initial record
4. File Input and Maintenance
    - Record input preparation and revision
    - Filing
    - Cross-reference control
    - Error correction
    - Transaction control; additions and deletions
5. Materials Handling
    - Sorting and preliminary control
    - Distribution to catalogers
    - Arrearage control
    - Pasting and labeling
    - Routing
6. Inventory Control
    - Establish shelflist record
    - Added copy control
    - Added volume control
    - Recataloging and reclassification control
    - Inventory statistics maintenance and analysis
7. Reference and Retrieval
    - File searching
    - Retrieval of item in process
    - File inquiry assistance

---

[26] Markuson et al., *Guidelines for Library Automation*, p. 73.

[27] Ibid., pp. 76–77.

8. Output Generation, Dissemination and Reporting
    - Report generation
    - Preparation of printed cards, worksheets, etc.
    - Preparation of printed book catalogs
    - Preparation of printed labels
    - Preparation of printed/punched book cards
    - Preparation of printed lists
    - Preparation of indexes
    - Dissemination of records
    - Union list reporting
9. Processing Catalog Records from Outside Sources
    - Ordering catalog records
    - Organization and dissemination of hardcopy records for cataloging
    - Processing, and maintenance of [illegible] readable records
    - Modification of nonlocal records
    - Selective dissemination of records

Establishment and surveillance of catalog policies and procedures is mainly an intellectual, judgmental function. Information to support decisionmaking can be provided by other automated functions, and statistical analyses and modeling can be performed by computer to support the user feedback and performance analyses activities. In addition, automated text processing can be used to maintain procedures manuals.

Establishment and maintenance of local authority lists or adoption of standard lists is both an intellectual and a clerical area. The maintenance of authority lists is essentially clerical, with file maintenance forming its major part. Automated systems have "maintained" and generated local lists for some time. Editing, updating, and control of the files, however, were done by human beings, who handled the files manually. Computer generation of authority lists has been tried (using a keyword approach), but without much success. More standard lists are now available for purchase in machine-readable form; such a list can be the basis of the local list. Filing rules for computers have been compiled and automated catalog filing has been performed for some time, but the complexity of the rules has varied among systems. Some of the more complicated schemes require human editing and human creation of sort keys, which the computer then manipulates, rather than using computer-generated sort keys based on program parameters. Computer support in this area can produce greater clerical efficiency, but the intellectual basis for authority work must still reside mainly with human beings.

Materials analysis is almost entirely intellectual in nature, and in terms of creation of original cataloging entries no major automation effort has progressed past the experimental or research stage. Initial catalog records have been computer-built from an automated acquisitions file, but the initial identification of the descriptive elements was human-generated in the acquisitions phase. The support of automated authority lists can make materials analysis more efficient and consistent, but it can do little more.

Automation can be used to identify items that already have a cataloging entry established by either the Library of Congress or another library. Searching the MARC data base or a larger network data base is possible through an automated system. If this search is made, then the human can analyze the material (book in hand) in terms of the entry and accept, reject, or modify it as necessary. This procedure has significantly reduced the number of items that require original cataloging and greatly relieved the intellectual burden of the cataloging department of individual libraries. Programs for reclassifying from one scheme to another (e.g., from Dewey to L.C.) can be written, but the results must still undergo heavy human scrutiny and revision.

File input and maintenance is a natural area for automation. The nature and degree of file automation can vary from library to library, however. If the library catalog file is the card catalog, the file input is in the form of catalog cards, which can be formated, printed, overtyped, and sorted for filing by an automated system. The actual filing and maintenance would have to be done by humans. If the file were to be a computer-based file for producing book catalogs or COM catalogs, or for online searching, this file input and maintenance and its automated system would have to be more extensive and elaborate in nature. The catalog entry, once established, would have to be converted to machine-readable form. This conversion entails explicitly identifying separate data elements and their natures, so that the program can manipulate each as required. The MARC format provides a framework for explicitly identifying these elements, but the editing is usually done by people, not machines. The Library of Congress has developed a format recognition program that analyzes a catalog entry and generates MARC tags and fields with an accuracy rate of about 75 percent. Once the editing is performed, the source record must be converted into a machine record and read into the system. Some automated systems generate cross-references by comparing the subject entries for the local catalog to a standard authority list, such as the eighth edition of the Library of Congress subject headings list.

Materials handling is basically a manual activity, but automation can be used to control arrearage or backlogged items, items in process, and routing of items.

Inventory control can be fully controlled, or at least supported, by automation. The degree of control depends partly on how much retrospective conversion of the card catalog to machine-readable form has been undertaken. If automation efforts have been limited to new titles only, there will not be enough basis for complete control. An effort will have to be made to create a file especially for inventory control. It may be possible to convert the shelflist or use a related file, such as an existing circulation system file. Some libraries approach retrospective conversion on a need basis; the added copy or added volume control system triggers the establishment of a machine-readable record in the computer file. The degree of sophistication of the automated statistical data and analyses also depends on how much of the entire catalog is in machine-readable form.

Reference and retrieval can be automated to a certain extent. Once a computer file is created, automated search and retrieval can be performed on it. Human beings, in some circumstances can initiate it, and in other situations program control can. For example, an acquisitions program could search the catalog holdings file automatically to determine whether a purchase request is a duplicate copy.

Output generation, dissemination and reporting (the preparation of cataloging output products — cards, book catalogs, COM catalogs, labels, book pockets, bibliographies, and indexes) is well suited to automation. However, automation depends, of course, on the degree of automation of other cataloging functions, e.g., amount of retrospective cataloging converted, tag structure used, etc. The idea is to key the data once and use it over and over in different forms, formats, and permutations.

Processing catalog records from outside sources becomes more and more practical as the use of catalog records in machine-readable form becomes more widespread. These are available through commercial sources and networks. The MARC tape distribution service has made almost 1 million cataloging records available to [illegible] and vendors, thus reducing the need for much [illegible] effort.

Typical cataloging problems that can be solved wholly or in part by a minicomputer system include:

- Maintenance of a name authority file
- Maintenance of a subject authority file and cross references
- Maintenance of a thesaurus
- Maintenance of a series authority file
- Efficient means to produce overtyped card sets
- Production of book or COM catalogs
- Maintenance and control of a catalog for a special documents collection
- Control of added copy and added volume routines
- Preparation of precataloging card sets and processing kits
- Control of items in process
- Maintenance of a union catalog of several collections
- Preparation of special bibliographies on request
- Preparation of permutated indexes.

SERIALS

Serials, as an operational area of a library, are difficult to pin down. The processes involved are basically acquisitions and cataloging and in some libraries are incorporated in the regular flow for book materials. Even what is defined as a serial can vary: periodicals, newspapers, technical reports, Government documents, annuals, handbooks, and other works issued in frequent editions, sets in progress, services, and monographic series.[28] Markuson, et al., defined two main types of serial systems: (1) listing that provides "access to, and control of, certain facets of the serial operation," and (2) check-in control that provides "mechanisms for the control of the receipt, recording, and routing of incoming serial issues."[29] Her list of serial functions follows.[30]

1. Establishment and Surveillance of Policies and Procedures
   - Policy development
   - Maintenance of procedure manuals
   - User feedback analysis
   - Performance analysis
   - Collection analysis
   - Interlibrary cooperation
2. Subscription Control

---

[28] Ibid., p. 110.

[29] Ibid., p. 109.

[30] Ibid., pp. 113–4.

  - Review of new order requests and renewal requests
  - Determination of procurement procedures
  - Establishment and maintenance of subscription control files
  - Preparation of subscription renewal approval lists
  - Order preparation
  - Fund accounting
  - Vendor and source file maintenance
3. Establishment and Maintenance of Bibliographic File Control
   - Catalog new serials, recatalog old titles
   - Prepare serial record entry
   - Provide cross-reference controls
   - Update serial holdings
   - Transaction control: additions, changes, deletions
4. Recording Incoming Receipts
   - Sorting and assignment of incoming issues
   - Bibliographic identification
   - Posting to control file
   - Marking and routing issues
   - Recording changes in bibliographic or control information for file updating
   - Claiming
5. Materials Handling and Collection Control
   - Sorting and shelving issues
   - Servicing request for serial issues
   - Routing and circulation control
   - Storage of title pages, indexes, etc.
   - Establishment and maintenance of binding control file
   - Missing issues control
   - Preparation of want lists to complete holdings
6. Output Generation, Dissemination, and Reporting
   - Preparation of serial holdings lists
   - Preparation of accessions lists and bibliographies
   - Union list reporting and/or printing
   - Printing order forms and subscription renewal lists
   - Print claim notices
   - Print binding notices
   - Prepare serial check-in forms or arrival cards
7. Reference and Retrieval
   - Processing bibliographic verification inquiries
   - Processing holdings and assigned location inquiries
   - Retrieval of serial issue from processing flow
8. Processing Nonlocal Records
   - Union list maintenance and publication
   - Selection and dissemination uses

Purchasing and bibliographic control

Establishment and surveillance of policies and procedures, like the other areas, involves mainly intellectual effort and can be only supported or aided by automation. The analyses needed in this area can be supported by automation more fully if the other functions are automated and the data can be collected by machine.

Subscription control, like acquisitions, includes both intellectual and clerical activities. Although selection is a human decision, the review of renewal requests can be aided by a computer alert that signals when subscription renewal is imminent. In addition, analysis of previous subscriptions can provide means of projecting subscription prices. Order preparation and fund accounting are clerical in nature and easily automated, as is the vendor and source file.

Establishment and maintenance of bibliographic file control begins with establishing and structuring the bibliographic elements for a serial according to standard rules. This activity is intellectual and must be done by a human being. Each entry then must be coded and converted into machine-readable form for further manipulation. The nature of periodicals is such that once an entry is established, it is subject to many changes, for example, title, frequency, and the corporate name of the issuing agency. The actual holdings of the periodicals are in a constant state of flux, the current status of which must be maintained. These changes indicate the need for many cross-references. All of this maintenance and control can be greatly aided by automation.

Recording incoming receipts can be aided greatly by automation. Although the initial sorting and subsequent routing of incoming issues is a manual process, the remaining activities are clerical and can be automated — or, if not fully, at least the computer can provide support. The means of constantly capturing the data, updating the status of the file, and generating claims when needed are the key elements in an automated serials system.

Materials handling and collection control includes some manual activities, but these can be supported by automated clerical processes, as in control and maintenance of the various files and the generation of different lists.

Output generation, dissemination, and reporting can be fully automated. When the information has been captured in machine-readable form, computer-generated output is an efficient means of producing lists, forms, or notices. With an online computer system, however, the need for some of the printed output may be eliminated.

Reference and retrieval can be eased by automation. The information in serial [illegible] must be accessible by the library staff at various stages of the work flow. Part of the information also is needed by the library public and must be made available in a format designed for public use. Due to the nature of serials, there are many potential access points for any entry, and the historical data for each serial can be significant. The value of automation is its ability to manipulate data over and over, in an efficient and economical way, in hardcopy or softcopy form.

Processing nonlocal records is a function that automation furthers, for example by producing union lists of serials from the collections of several libraries. The holdings and titles on the lists can be kept current with great ease and efficiency, and the production of [illegible] lists can be accomplished quickly and economically. The introduction of serial records [illegible] in MARC format by the Library of Congress and the new cooperative serials project CONSER have made more standard cataloging entries available for libraries and reduced the amount of coding and keying required to build a bibliographic file for serials.

A number of typical tasks that can be done by a minicomputer system are

- Controlling subscriptions: orders and fund control
- Maintaining and controlling a catalog of serial titles
- Preparing and controlling union lists of serials from several collections
- Controlling routing for new issues
- Controlling the finding of materials
- Controlling continuations
- Checking in new issues efficiently
- Efficiently handling claims of both titles and individual issues
- Maintaining and controlling information on where individual serials are indexed and abstracted.

CIRCULATION

Circulation systems vary not so much in type as in degree. Some maintain only transaction information — what is currently charged out — and some maintain files on all potential borrowers, all materials in the collection, and all current transactions. Some systems handle overdue notices, reserve books, and maintain hold files and a bad borrower file.

Markuson and her colleagues compiled the following list of functions of the circulation operation.[31]

1. Establishment and Surveillance of Policies and Procedures
   Policy development
   Maintenance of procedure manuals
   User feedback analysis
   Performance analysis
   Interlibrary cooperation
2. Authorized Borrower Control
   Borrower registration
   Borrower identification
   Special routines for exceptional borrowers
3. Charging Procedures
   Charging materials; books, serials, etc.
   Recording charge transaction
   Book reservation procedures
4. Discharging Procedures
   Discharging materials
   Recording discharge transaction
   Identification of reserved items
5. File Input and Maintenance (all files)
   Transaction record input — charges, discharges, etc.
   Borrower file input
   Transaction control, additions, deletions
   Error correction procedures
   Inventory control records
   Overdue and fine accounting
6. Overdue Control
   Identification of overdue items
   Receipt and control of overdues and fines
7. Interlibrary loan
   Monitoring of interlibrary loan requests — incoming
   Monitoring of interlibrary loan requests — outgoing
8. Output Generation, Dissemination, and Reporting
   Charge records
   Overdue notices
   R[illegible]
   R[illegible]
   P[illegible]ted circulation and discharge
   [illegible]
   Dissemination of records, reports, etc.
9. Reserve or Special Noncirculating Materials

---

[31] Ibid., pp. 130–1.

  - Establishment of control procedures
  - Maintenance of special transaction files for noncirculating and reserve materials

10. Reference Inquiry
    - Identification of items on loan
    - Identification of missing items
11. Materials Handling, Storage, and Maintenance of the Collection
    - Retrieval of requested items
    - Reshelving and maintenance of items
    - Routing of materials
    - Physical preservation of items
    - Inventory of collection
    - Purging of outdated and unwanted items
    - Preparation of materials for shipment

Establishment and surveillance of policies and procedures is mainly an intellectual activity. Automation can be used to support the decisionmaking with statistics gathered and analyses performed.

Authorized borrower control is a candidate for complete automation. Borrower registration can involve merely individual borrowers providing information for the file. However, in many special libraries the borrowers' file can be taken from an existing file of agency employees. In most cases a control number, such as the social security number, is assigned. Often systems use an identifying badge or card with the number in machine-readable form to provide the means of input. If special classes of borrowers exist, these can be handled via algorithms.

Charging procedures entail gathering the necessary information to record an item as in circulation. Information on the book, the borrower, and the details of the loan itself (charge date, due date, branch or location, etc.) are generally needed. The ease with which this information can be gathered is critical. Book reservations procedures can be automated so that a book cannot be checked out except by the person who is next on the "hold" list. A notice to this next person informing him of the availability of the book can also be generated automatically.

Discharging procedures are the reverse of the charging procedures and usually involve gathering statistics on use for future analyses.

File input and maintenance is critical to all files in the system, although they can vary with different system designs. The major factor in this subfunction is the efficiency with which the data can be captured and the records kept current (updated).

Overdue control is eased by automation. A system for automated control of overdue materials, generation of overdue notices, and computation of fines is simple if the original transactions are in machine-readable form. (Interlibrary loans will be discussed as a separate operation.)

Output generation, dissemination, and reporting includes a variety of functions. The output products that can be computer-generated depend on which functions are automated. For online systems many of the printed products are no longer necessary.

Collections of reserve or special noncirculating materials are kept by some libraries under special loan procedures, such as short-term loans for several hours or unlimited loans for indefinite time periods. Once algorithms are defined, the procedures can be automated.

Reference inquiry to circulation must be accommodated. Both library staff and the users need to access the transaction file to ascertain the status of specific items or of particular users' total transactions.

Materials handling, storage, and maintenance of the collection consists mainly of manual activities. Automation is used only in a support or utility role. A minicomputer can be used to support any or all of the phases of a circulation system as discussed and its real value is in making an online system viable.

INTERLIBRARY LOANS

Interlibrary loan processes can be included in the circulation operation or set aside as a separate operation. The process is divided into two main functions, incoming and outgoing interlibrary loan requests.

Hayes and Becker described the interlibrary loan functions as follows.[32]

I. Borrowing
  A. Making request
    1. Assistance to patrons
    2. Determining location, verifying
    3. Completing form, filing
    4. Mailing
  B. Receiving item
    1. Receiving and unwrapping
    2. Checking records
    3. Notifying patron
  C. Payment
    1. Keeping records
    2. Making payment, mailing
  D. Returning item

---

[32] Hayes and Becker, *Handbook of Data Processing for Libraries*, pp. 574–5.

ERIC Full Text Provided by ERIC

    1. Changing records
    2. Wrapping and sending

II. Lending

- A. Receiving request
  1. Receiving and verifying
  2. Checking catalog, locating
  3. Searching shelves, pick-up
  4. Keeping records, check-out
- B. Photocopying
  1. Checking pages requested
  2. Sending to photocopy
  3. Copying
  4. Keeping records
  5. Preparing, logging invoices, filing
  6. Processing payment upon receipt
- C. Sending
  1. Wrapping
  2. Distributing or mailing
- D. Followup
  1. Sending overdue notices
  2. Keeping records
- E. Returned items
  1. Unwrapping and inspecting
  2. Changing records
  3. Discharging
  4. Reshelving

User interface, establishing entries, searching and locating sources or items, and checking materials are all intellectual activities. Searching and locating sources can be automated if there is a data base of holdings accessible to outside sources. For example, an online cataloging or circulation data base that can be accessed from external terminals could be used to search and locate sources. The other intellectual activities do not lend themselves to automation. The manual activities — handling the material, shelving or pulling, wrapping or unwrapping, and photocopying — also cannot be automated. The rest of the functions are mainly clerical or control activities and can be automated.

The interlibrary loan tasks that can be supported by minicomputers include

- Preparation of request forms and control of items in the borrowed file
- Maintenance of a calendar for scheduling material returns
- Maintenance and control of an overdue system
- Maintenance and control of an items-loaned file
- Control of a borrowers' file and/or a source (supplier) file.

REFERENCE AND INFORMATION SERVICES

The reference function represents interaction of the library user with the library representative. This representative can be a human (a member of the library staff) or a tool (the library catalog, an index, or a specific book), that meets the user's need to locate the information desired. Because technical processes provide the materials and means for serving the user, most of the tools used for reference are products of the technical services operations and systems.

Markuson compiled the following list of nine reference and bibliographic functions.[33]

1. Establishment and Surveillance of Policies and Procedures
   - Policy development
   - Maintenance of procedure manuals
   - User feedback analysis
   - Performance analysis
   - Interlibrary cooperation
2. Development and Maintenance of Reference Sources
   - Evaluation and selection of hardcopy reference materials
   - Evaluation and selection of machine-readable reference materials
   - Development of specifications for handling machine-readable reference materials
3. Identification of Relevant Outside Reference Sources
   - Printed material sources
   - Machine-readable bibliographic sources
   - Machine-readable data-base sources
   - Organizational and people sources
   - Establishment of procedures for utilization of outside sources
4. Search and Retrieval
   - Initial screen and/or referral
   - Question negotiation and analysis
   - Conversion of question to appropriate terminology
   - Formulation of search strategy
   - Conduct of search
   - Evaluation of search
   - Maintenance of library and user profiles
5. Preparation of Bibliographies

---

[33] Markuson et al., *Guidelines for Library Automation*, pp. 139-40.

Definition of scope and type of bibliography
Screening of potentially relevant items
Preparation of bibliographic entry including abstract and annotation as required
Preparation of final copy
Duplication and dissemination

6. Requests for Photoduplication
   Screening requests
   Duplication
   Distribution of photocopies
   Accounting control
7. Preparation of Translations
   Identification of language skills available within library or appropriate affiliated group
   Identification of other sources of translation services
   Screening translation requests
   Preparation and presentation of translation
   Accounting control
8. Establishment and Maintenance of Special Materials
   Agency archival files
   Records of local search results
   Clipping and pamphlet files
   Personnel skills inventories
9. Materials Handling
   Maintenance of reference collection
   Handling of items retrieved from other collections
   Physical preparation and filing of special materials

Establishment and surveillance of policies and procedures is an intellectual activity, and automation can only support it, mainly by analyses of performance and user feedback. If formal survey techniques are used to gain user feedback, the necessary statistical analyses can be performed by an automated system.

Development and maintenance of reference sources is an intellectual activity that must be performed by human beings but is involved with automated efforts or products. Actual use of the machine-readable materials may be required for testing purposes.

Although *identification of relevant outside reference sources* is an intellectual activity, implementation of the machine-readable sources requires establishing an automated system or installing an existing system.

Search and retrieval can be performed manually on output products, such as card catalogs, book catalogs, or special bibliographies, or it can be done online at a terminal. In either case, the access points and the way the information will be approached must be determined in advance. The actual queries will depend on the search techniques chosen. (Is free text searching possible? Is Boolean logic searching available? Are predetermined descriptors required? Is a permutation technique appropriate?) Library and user profiles can be used for an automated SDI system if a machine-readable data base is available for searching.

Preparation of bibliographies can be automated. Computer-generated bibliographies can be produced from machine-readable data bases if the records are structured properly. For example, using the fixed-field data in the MARC II format, bibliographies could be compiled based on parameters dealing with language, country of origin, form of contents (yearbook, directory, etc.), types of illustrations present, year published, and so on. Current awareness lists and SDI lists based on profiles also can be considered bibliographies and can be computer-generated.

Requests for photoduplication are handled mainly by manual activity, but automation could be applied in accounting control.

Preparation of translations is an intellectual activity. The clerical functions of accounting control, preparation of orders or requests for translations, and maintenance of a file of sources of translation services can be automated.

Establishment and maintenance of special materials collections (such as photographs, clippings, internal research reports, commercial catalogs, and proposal/contract documents) can be controlled by automation. Some commercial systems provide for capture of source material on microforms and for search and retrieval of the data by minicomputer.

Materials handling is a manual activity that does not lend itself to automation.

Although most applications of minicomputers to reference and information services are approached through problems in the other operational areas, some needs can be mentioned:

- Public (user) access to the serials holdings file
- Public access to the thesaurus used
- Public access to materials in process
- Maintenance and control of special collections
- Maintenance, control, and access to microfiche documents collections
- Maintenance and control of files of human resources and their subject areas
- Maintenance and control of vertical files
- Access to commercially available online data bases of abstracts and indexes
- Maintenance and control of a current awareness and/or SDI system

99

- Production of special bibliographies on demand
- Performance of literature searches on demand.

ADMINISTRATION

Administrative functions vary according to the library's place in the parent organization's structure and the payroll, personnel, and budget control services provided by that organization. The functions of library administration are, in general, those present in any organization, and general business automation systems dealing with administrative activities should apply.[34]

Markuson's list of administrative functions follows.[35]

1. Establishment of Procedures vis-a-vis Parent Agency
   - Legal requirements
   - Funding, staffing, and operational policies
   - Policies for joint sharing of facilities, e.g., computer equipment
   - Reporting requirements and policies
   - Identification of needed library services
   - Establishment of interlibrary and interagency cooperation policies
   - Establishment of contracting requirements
2. Personnel Procedures and Policies
   - Personnel selection
   - Establishment of personnel policies, records, benefits
   - Formal and on-the-job training procedures
   - Personnel and records management
3. Fund and Property Management
   - Budget preparation and review, annual
   - Budget preparation and review, long-range
   - Budget allocation and surveillance
   - Library fund accounting
   - Cost analysis review
   - Property inventory
4. Organization and Administration of Functional Operations
   - Establishment of services to be performed
   - Development and review of organization to perform services
   - Development and review of administrative policies for each organizational unit
   - Assignment of line and staff administrators
   - Establishment of criteria for procedure manuals, forms, etc.
   - Operations analysis review
5. Reporting
   - Establishment of statistical and reporting requirements
   - Preparation and review of annual reports
   - Establishment of staff communication via meetings, newsletters, etc.
6. Coordination and Cooperative Efforts
   - Establishment of local agency coordination policies
   - Establishment of agency coordination policies
   - Establishment of policies for coordination with other libraries and pertinent groups
   - Assignment of staff to specific coordination functions
7. Long-Range Planning
   - Monitor developments and needs in local and central agency
   - Monitor developments in other libraries and relevant organizations
   - Assign staff to planning tasks
   - Prepare and periodically update long-range plans
8. Establishment of Systems Development and Evaluation Procedures
   - Establish mechanisms for operational analysis
   - Establish procedures for utilization of outside skills, e.g., contractors, staff sharing, etc.
   - Establish policies for utilization of equipment, e.g., tele-typewriters, computers, etc.
   - Establish review mechanism for operational changes
   - Establish policies for cooperative planning, development, and implementation

Establishment of procedures vis-a-vis parent agency is an intellectual activity that must be performed by a human.

The major activities of *personnel procedures and policies* are intellectual, but records management can be automated.

Fund and property management consists mainly of intellectual activity, but property inventory control can be automated. The other activities of this function can be supported by automation in the form of statistical analyses, modeling, and simulation.[36]

---

[34] For a discussion of the automation of library administrative activities, see: Hayes and Becker's "Chapter Fourteen, Administrative Data Processing," *Handbook of Data Processing for Libraries*, pp. 383–414.

[35] Markuson et al., *Guidelines for Library Automation*, pp. 147–8.

[36] For a collection of essays on and a bibliography of library operations research, see: Peter Brophy, Michael K. Buckland, and Anthony Hindle, *Reader in Operations Research for Libraries* (Englewood, Colo.: Information Handling Services, Library and Education Division, 1976).

Organization and administration of functional operations is an intellectual area. Computer support is applicable to "operations analysis review."

Reporting can be supported by automation by using data gathered in other automated systems in the library. A word processing system can support preparation of reports.

Coordination and cooperative efforts is entirely intellectual in nature.

Long-range planning can be supported by automation, with either computer-generated data or computer analysis and modeling.

Establishment of systems development and evaluation procedures is an intellectual activity.

SUMMARY

The problems and solutions described in this chapter are merely representative of the tasks to which a minicomputer system could be applied in a Federal library. Individual libraries may have different needs that also can be served by minicomputers in the form of a standalone computer, a part of a network, or a local unit for a host computer. The number of minicomputer applications is limited only by the creativity of systems design teams. The possibilities are endless.

CHAPTER FIVE

SELECTION CRITERIA

As stated in the Introduction, Part II presents guidelines for selecting and implementing a minicomputer system. Part II begins with the assumption that a library has decided to use a minicomputer system. For those readers who are already familiar with minicomputers and who skipped Part I to begin this book at Part II, a suggestion is in order. The following guidelines are based on a design model described in Chapter Four and it may be of value to review this model.

FACTORS INFLUENCING THE DECISION

During systems analysis and design in Part I, various possibilities for solving problems or meeting needs were reviewed. The systems design document reviewed each one, provided the information on which the administrators made the decision, and contained all the details necessary to initiate the computer system.

How did they arrive at that position where they could make their decision? First, the problem or need was expressed in terms of specific goals and objectives. The existing system was analyzed, and system requirements for the new design were compiled. The design model was applied. That is, design characteristics pertinent to the system requirements were determined, and the hardware and software impacts for each were reviewed; the class (or classes) of computers was deduced for each of the choices; the systems document was prepared and the time for decisionmaking was at hand.

What factors were considered in the choice of alternatives? Some of the main factors in the design decisions were considered early in the phase of decisionmaking. The system requirements included the organizational and environmental impacts, as well as those of associated systems, and the constraints (physical, organizational, technical, contractual, and resources) that had to be taken into consideration.

The remaining factors deal with the various possible solutions, which had been shaped by the previous factors.

COMPARISON TO A MANUAL SYSTEM

Undoubtedly, in most problem areas a manual approach could be viable. A manual system and a minicomputer system, it must be stressed, really cannot be compared if the automated system is approached properly. The mini system should do more than the manual system but still allow all the necessary tasks to be performed. With any automated system the staff relief should be in the clerical ranks, as clerical tasks are the ones usually automated. There may be an increase in the duties and responsibilities of the paraprofessional or professional staff. Because information must be expressed explicitly for computer manipulation, the source data must be edited and coded before it is inputted. Higher level staff (if not to edit, at least to revise) is usually required. The entire operational area may feel the impact of automation. An operation, to be automated, must be clearly structured, precisely defined, and tightly organized. Precedents and decisions must be recorded. Rules and authorities must be controlled and consistently applied. The degree of control may be much greater than in the manual system, and this may increase the work in the unit. For applications in which retrospective conversion may be appropriate, the decision whether to convert must be made. The file must be examined to determine how much must be converted (the file's volume) and how "clean" it is (how much editing must be done before converting).

COMPARISON TO A LARGE-SCALE SYSTEM

In comparing a minicomputer system to a large-scale system, the situation or environment must be established.[1] For example, comparison of costs or services must take into account whether the large-scale computer would be the library's own, the agency's (with or without charge-back fees), or that of a commercial service bureau or contractor. The main differences between large-scale computers and minicomputers were discussed in Chapter Two. Additional points should be made.

- An online interactive system on a mini may be cost-competitive with a large-scale batch system.
- A library gains more control over its destiny with its own mini but must take on the duties attendant in operation, maintenance, repair, etc.
- A large-scale computer may be best used along with, not instead of, a mini system. Very large requirements for sorting or printing can be performed on a large-scale computer, and this combination may be an alternative to consider.

SERVICE CONSIDERATIONS

Service considerations affecting alternative designs should be stated explicitly at the end of the system design phase. The choices should be more of style and degree than of services present or not. Impacts on users should be considered first, and then impacts on the staff. Time involved can be a variable, as can ease of operation or use. Undoubtedly tradeoffs will be made.

COST/BENEFIT ANALYSES

Cost analysis is part of systems design as outlined by Markuson, and the costs for each alternative should have been prepared. Markuson suggests that the analyses should include the following.[2]

1. Determine development costs for all aspects of the system including:
   - Manpower costs
     System design manpower
     Programming manpower
     Contractors and consultants
     Secretarial and clerical support
   - Communication costs
     Travel
     Telephone
   - Computer costs
     Computer time for program debugging and testing
     Computer time for data input and file building
   - File conversion
     Input preparation and editing
     Input operators
     Input equipment
   - Training
     Preparation of materials for staff training
     Special courses, etc. for project team
   - Other
     Expendable materials
     Space
2. Develop projections for operational costs:
   - Manpower for system operation and for system modification
   - Manpower for file input and maintenance
   - Equipment costs for input and computer processing
   - Expendable materials—printer forms, paper, labels, etc.
3. Compare operational costs for manual and computer system.
4. Prepare cost report

The costs alone are not enough. A cost/benefit analysis should be performed.[3] This judgment is difficult because so much of the "benefit" would be in

---

[1] Auerbach compares minicomputers with wired logic components, large-scale computers, and commercial time-sharing systems in terms of the five general application areas for minicomputers; *Auerbach on Minicomputers* (New York: Petrocelli Books, 1974), pp. 76-81. Hayes and Becker compare various system configurations for library applications; Robert M. Hayes and Joseph Becker, *Handbook of Data Processing for Libraries*, 2d ed. (Los Angeles: Melville Publishing Co., 1974), pp. 268-74. Davison raises three significant issues that must be considered when deciding between large-scale and minicomputers for libraries and Grosch lists seven questions to be answered; Wayne Davison, "Minicomputers and Library Automation: The Stanford Experience," in *Applications of Minicomputers to Library and Related Problems*, ed. F. Wilfrid Lancaster, papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974 (Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974), pp. 93-95, and Audrey N. Grosch, "Minicomputer — Characteristics, Economics and Selection for an Integrated Library Management System," in *Applications of Minicomputers to Library and Related Problems*, ed. F. Wilfrid Lancaster, papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974 (Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974), pp. 164-5.

[2] Barbara Evans Markuson et al., *Guidelines for Library Automation, A Handbook for Federal and Other Libraries* (Santa Monica, Calif.: System Development Corporation, 1972), p. 58.

[3] Kimber has a brief discussion on the economic aspects of automating, especially unit costs; R[ichard] T. Kimber, *Automation in Libraries*, 2d ed. (Oxford: Pergamon Press, 1974), pp. 23-26. Hayes and Becker have an entire chapter on cost accounting which is used to determine unit costs and other financial data for decisionmaking. They also have a chapter called "System Budgeting and Evaluation." This deals with criteria of evaluation, specifically cost/effectiveness (i.e., cost/benefit); see "Chapter 4, Cost Accounting in Libraries," and "Chapter 7, System Budgeting and Evaluation," *Handbook of Data Processing for Libraries*, pp. 102-21, 178-94. Becker and Hayes discuss the benefit aspect, i.e., defining the criteria of value; Joseph Becker and Robert M. Hayes, *Information Storage and Retrieval: Tools, Elements, Theories* (New York: John Wiley & Sons, 1963), pp. 238-56.

ERIC

service, and library service has been notoriously difficult to quantify and evaluate. Swihart and Hefley call these "intangible costs" and express the problem this way:[4]

> Intangible costs must also be arrived at for every benefit claimed for a system. If service is to be improved, just how will it be improved, and specifically what value can be placed on it. Some library administrators will claim that they cannot estimate the value of faster service, more books, etc., but such judgments can be made with reasonable accuracy and when made, should be recorded in writing.

However it is accomplished, the point is that cost alone should not be the sole consideration. The benefits must be reviewed and considered in the decisionmaking process.

SELECTING THE SYSTEM

Once an alternative has been chosen, the next step is to decide how to proceed. There are three main choices: (1) The library and its agency can develop and implement the system, (2) a commercial contractor can be hired to develop and implement the system, or (3) a turnkey system can be purchased and implemented. The decision may take the form of a process of elimination. If there is no turnkey system available, the next question is whether the agency data processing unit can or will take on such a project. If not, there may be no recourse but to contract out the effort. Where options are available, the decision must be based on ability to meet the system requirements, costs involved, and time to complete.

Whichever method is followed, the steps involved are basically the same. Ollivier wrote that there were four distinct stages.[5]

> Design. Specifications.
> Solicitation. Problem description and specifications are sent as request for proposal (RFP) to a list of qualified vendors.
> Evaluation. Proposals submitted by vendors are reviewed and evaluated according to explicit criteria and rated according to their performance and cost.
> Negotiation. Vendor(s) discussions are undertaken and a contract is agreed on.

These stages will be undertaken on the procurement of a turnkey system which includes hardware, software, and support; on the hiring of a contractor to prepare and install the new system; and on the purchase of hardware by the agency/library design team.

The technical specifications should have been drawn up and completed as part of the system design document. These specifications often take the form of a general description of the components needed to meet the requirements, as opposed to a shopping list of specific brands, model numbers, and quantities. The vendor is thus allowed to suggest the best of what is available to meet the specifications. Markuson and her collaborators provided guidelines for preparing hardware specifications.[6]

1. Develop specifications for each type of equipment required
   - Input equipment
   - Computer equipment
   - Output equipment
2. Input equipment specifications should include
   - Character set and encoding characteristics
   - Operating specifications, e.g., ease of operations
   - Reliability of operation
   - Error detection and correction requirements
   - Throughput rates, i.e., speed
   - Display and/or printing requirements
   - Environmental characteristics, e.g., size, noise of operation, portability, etc.
   - Maintenance and training provisions
3. Computer equipment specifications should include
   - Core storage and secondary storage characteristics
   - Data manipulation capabilities
   - Throughput requirements
   - Input readers required, e.g., paper tape readers, and other special peripheral gear
4. Output specifications should include
   - Printing capabilities needed, e.g., upper and lower case
   - Output form handling capabilities
   - Legibility specifications for printing and/or displays
   - Throughput rates
   - Maintenance provisions

Examples of these kinds of specifications were offered at the 1974 clinic on minicomputer applications in libraries. The University of Chicago example was for a front-end minicomputer to serve as a data concentrator and high-speed interface to a host (IBM 370/168). The specifications included the following.[7]

---

[4] Stanley J. Swihart and Beryl F. Hefley, *Computer Systems in the Library. A Handbook for Managers and Designers* (Los Angeles: Melville Publishing Co., 1973), p. 251.

[5] Robin T. Ollivier, "A Technique for Selecting Small Computers," in *A Practical Guide to Minicomputer Applications*, ed. Fred F. Coury (New York: IEEE Press, 1972), p. 94.

[6] Markuson et al., *Guidelines for Library Automation*, p. 53

1. Processor speed — sufficient to drive the estimated mix of terminals
2. Disk storage — sufficiently large to include software plus certain files
3. Tape drives — two, to log all transactions
4. Console — operator communications and programming tool
5. Communications interfaces — high speed to the computation center, a mix of speeds for the terminals
6. System software — disk oriented, with assemblers or compilers and a communications package
7. Service — to be locally available and reliable

The University of Minnesota Bio-Medical Library specifications for the processor for its minicomputer configuration contained these requirements:[8]

1. CPU must be 8-bit byte-oriented with main memory addressable by byte location and preferred word size a multiple of 8 bits
2. Either explicit character manipulation instructions or some reasonable method of effecting these within the available instruction set
3. Multilevel indirect addressing and indexing or their functional equivalents are required
4. Multilevel indexing is desirable but not required
5. Main memory must be incrementable to at least 64K bytes
6. Direct memory access required
7. Real-time clock required
8. Hardware multiply/divide required
9. Power fail/automatic restart required
10. Memory protection required
11. Operator console keyboard/printer with 30 cps speed required.

One of the best technical specifications ever prepared as part of an RFP was issued by the Library of Congress early in 1977 for the procurement of several minicomputer systems.[9] The system specifications and the weighting for the technical evaluation are presented in Appendix B. The RFP was accompanied by project descriptions for each application area. This one is certainly a well-written model for other libraries to follow.

REVIEW OF HARDWARE/SOFTWARE

Ultimately, somewhere along the flow of events, whatever the method followed, the systems team will have to consider and evaluate specific manufacturer's models, piece by piece. As a background for this procedure, and to provide a common ground for interpretating the proposals, the main components considered in drawing up the five classes of minicomputer systems described in Chapter Three will be reviewed briefly. These descriptions represent a range of what is available in each area. The main characteristics or elements of each component are reviewed. Sample prices or price ranges are given only as a basis for comparison. There is no attempt to specify model names and numbers or precise costs, because the minicomputer industry is in such a state of flux.[10]

A study of minicomputer prices from 1972 to 1975 shows that prices are dropping (Table 20).[11]

TABLE 20—Cost Comparison Between 1972 and 1975

Unit	1972	1975
CPU with 2K memory	$ 2,500	$ 650
4K memory	2,200	550
Chassis	300	100
Power supply	400	375
Control panel	225	275
DMA	300	Included
Power failure restart	350	Included
Teleprinter interface	150	150
Real-time clock	350	Included
Autoload	400	50
16-bit digital I/O	500	500
Paper tape reader/punch	5,800	5,600
Moving head disk	13,000	12,000
Floppy disk (dual)	NA	4,300
Printer	5,500	4,950
	$31,925	$25,200

Technological advances cause changes. The sophistication of a device can increase while the price remains the same or even increases. Prices must be studied carefully for other factors. Some prices are quoted for OEM buyers. These are stripped units with no cabinet, power supply, controller, etc., which are to be incorporated into larger equipment or systems. Prices for end users are higher, because they need complete units ready to use. Some prices are for "packages" or "systems" that include several components with no choice of model or features. Some prices for peripherals do not include the controller, which must be purchased as a separate piece of equipment. Sometimes the controller is part of the larger CPU system, which means that the I/O device would not work on a different manufacturer's CPU. The prices of a major manufacturer can be somewhat higher than those of independent component manufacturers, especially in the area of peripherals.

---

[7] Charles T. Payne, "The University of Chicago Library Data Management System," in *Applications of Minicomputers to Library and Related Problems*, ed. F. Wilfrid Lancaster, papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974 (Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974), p. 116.

[8] Grosch, "Minicomputer — Characteristics, Economics and Selection," p. 166.

[9] U.S., Library of Congress, Procurement and Supply Division, "Mini Computer Systems; Request for Proposal," Washington, D.C., 1977. (Mimeographed.)

[10] Withington predicted advances in computers and included price forecasts to 1985. This included minicomputer components and peripherals — both for state-of-the-art technology and predicted technological developments; see Frederick G. Withington, "Beyond 1974: A Technological Forecast," *Datamation* 21 (January 1975): 54-73.

[11] Marty B. Jarosz, "Minicomputers — Microcomputers — Peripherals; What Are the Real Cost Hang-Ups?" *Mini-Micro Systems* 9 (May 1976): 82.

CPU

The CPU is the most difficult to describe and evaluate. The main elements used to describe the processor are fairly standard, but listing them does not tell the whole story. Note that the manufacturer of each of the following CPUs uses different characteristics to describe it.[12]

Manufacturer 1
Memory technology = core
Memory size (bytes) = 8–32K
Word size (bits) = 16
Cycle time (μs) = 1.2/word
Microprogrammed = Yes

Manufacturer 2
Memory size (byte) = 4–16K
Word size (bits) = 8
Cycle speed (μs) = 1.6
Transmission code = any 7–11 bit code
Maximum I/O devices = 16

Manufacturer 3
Main memory = 10–110 (kc)
Character size = 6 bits
Addressable registers = 3/partition, except common
Cycle time = 3.3 (μs)
No. of I/O channels
  1 slow speed/user partition
  1 high-speed (FAC)
Maximum devices/channels = 10

Manufacturer 4
Memory size = 12–24K
Word size = 24 bits
Maximun I/O channels = 8

Manufacturer 5
Cycle speed (MHz) = 1
Storage technology = MOS
Main memory capacity = 32–80 (Kb)

Manufacturer 6
No. of registers = 20+ implicit register
Memory size (wds) = to 64K (48K available to user)
Bits/word = 8
Parity/Protect = both
Cycle time (μs) = 1.6
Paging = yes
ROM Control memory
Instructions
  Number = 91
  Double precision = yes
  (Hardware) multiply/divide = NA
    floating point = NA
  Stack manipulation = yes
  Priority interrupt = 10 levels

Even with products of a single manufacturer the comparison is difficult. For example, the IBM 370 is known as the latest in large-scale computers. However, 370 is a series number, and a second number identifies the model. At the top of the 370 series, for example, is the 370/168. At the other end of the series, the 370/115 is listed as the minicomputer. The Digital Equipment Company manufactures a minicomputer series called PDP-11. The size and power of this series varies from the PDP11/04 (4K memory) to the PDP11/70 (128K memory). One manual listed the base price of the PDP11/04 as $2,995 and the base price of the PDP11/70 as $54,600.

Sometimes model numbers for a series are based on software alone. For example the General Automation data management DM200 series (220, 230, 240, and 250) all use the same hardware; they differ in their software:[13]

220—Remote job entry
230—Standalone batch system using disk monitor operating system (DMS)
240—Foreground / background (communications / batch) using communications monitor system (CMS)
250—Time-sharing and batch using time-sharing operating system (TSO)

Because of the wide variety of CPUs, it is difficult to state benchmark prices by categories. In fact, price is not always indicative of capability. A General Automation DM 230 standalone batch system was compared to "competitive systems," ranging in price from $47,650 to $82,595 (Table 21).[14] "... It is difficult to be sure that we are comparing apples to apples, since the configurations shown vary in disc storage capacity, basic system components, and card/line printer ratings."

---

[12] *Auerbach Buyers' Guide to Business Minicomputer Systems, Winter 1976–77* (Pennsauken, N.J.: Auerbach Publishers, 1977), pp. 36, 70, 79, 110, 262, 346.
[13] Ibid., pp. 121–2.
[14] Ibid., p. 122.

TABLE 21—Comparison of "Competitive Systems"

Configuration	Main Memory	Mass Storage	I/O Devices	Price Mid 1975
General Automation DM230	32K bytes	20M bytes disk storage	TTY, line printer (600 lpm), card reader (400 cpm)	$47,650
DEC Datasystem 356	32K bytes	40M bytes disk storage	TTY, line printer (300 lpm)	$61,345
Hewlett-Packard M260	32K bytes	23.5 bytes disk storage	Paper tape input, mag tape device, TTY, line printer (200 lpm)	$79,200
Singer System Ten	30K bytes	20M bytes disk storage	Workstation, line printer (450 lpm)	$82,595
Eclipse C/300	96K bytes	10M bytes disk storage	Console, CRT, mag tape device, 60 lpm printer, 4-line async multiplexor	$77,400

The five classes of minicomputers put together in Chapter Three as sample systems can be priced as a basis of comparison (Table 22). The prices that follow are for end-user systems and are on the high end of the price scale. The price for the processor is for the largest end of the main memory range and includes the CPU, main memory, I/O control, communications control, chassis, power source, control panel (or console), support software, and operating system. It does not include a compiler.

TABLE 22—Costs of the Five Classes of Minicomputer Systems

Price	Class	Memory Size	Category
$6000-$7000	I	(4K)	Intelligent terminal (Data Collection)
$15,000	II	(32K)	Simple executive monitor or single batch general application
$30,000	III	(64K)	Single or multiple batch with general applications including sorting
$70,000	IV	(128K)	Interactive single application for multiple online users
$85,000	V	(64K)	Time-sharing interactive system with multiple applications for multiple online users

Taking these as base figures, the cost of the various peripheral and software packages to be used can be added for a total system price.

Peripherals

The peripherals chosen to fill out the hardware configuration deserve special consideration. Not only do they perform the system interface to the user and thereby influence the efficacy of the system, they can involve as much as 90 percent of the total cost (see Table 20) of the hardware. Also, the dramatic price drops found in processors have not occurred in the area of peripherals.

In the early days of the minicomputer, most peripherals used were existing ones designed originally for large-scale computers. The miniperipheral market has now developed, and there are many choices for most devices. There are three sources for peripherals: the minicomputer (processor) manufacturer, the OEM manufacturer who sells the peripheral without interface, and the independent manufacturer specializing in complete plug-compatible peripheral systems.

As *Modern Data* pointed out:[15]

> The safest and easiest way to buy a peripheral is from the mini manufacturer. The minimaker has designed, tested, and fabricated interface logic and diagnostic/ debug software to ensure trouble-free mating of the peripheral with the mini. However, since the mini manufacturer usually sells only to his base, production is limited and prices tend to be higher than the other two alternatives.
>
> Independent miniperipheral manufacturers/suppliers are the second alternative. These manufacturers can either manufacture their own complete peripheral sub-system or they can buy from an OEM manufacturer and provide a specialized interface for certain mini models. Most of these manufacturers have a high degree of flexibility since they are not locked into a specific make or model, and they are generally less expensive than mini manufacturers because of high volumes in certain peripherals.
>
> The third alternative is for the do-it-yourselfer who likes to design his own hardware and software interfaces. The peripheral can be purchased from an OEM manufacturer and the controller from a variety of controller manufacturers for the lowest price of the three alternatives.

The advertisements for peripherals manufactured by independents all emphasize which mini model they are compatible with. Industry magazines prepare survey articles and include charts that show which independent's device fits which minicomputer.

---

[15] "Peripherals Make the Mini; *Modern Data's* Annual Survey of Plug-Compatible Miniperipherals," *Modern Data* 8 (December 1975): 34.

Another facet of the peripheral market is the "IBM-equivalent" or "IBM-standard" product. Some peripherals are designed to replace an IBM device in such a way that the processor and software are "ignorant" of the substitution. For example, disk cartridge systems are presented as equivalent to the IBM 2315 or 5440, and disk packs are described as equivalent to IBM 2314 or 3330 systems.

The point has been made that peripherals are expensive; a high percentage of the cost of the peripheral system is for the interface. Julinssen compiled a table that illustrated the proportion of the cost of the interface to the price of the device (Table 23).[16]

TABLE 23—Comparison of Costs of Peripheral Devices and Their Interfaces

Peripheral Device	Peripheral Price ($K)	Peripheral Interface Price ($K)
TI 3330 Moving Head Disk	20	14
TI Cartridge Moving Head Disk	5.1	3.3
TI Magnetic Tape	6	2.5
DEC Cartridge Moving Head Disk	5.1	5.9
DEC Magnetic Tape	7.5	3.2
AED Floppy Disk	0.75	1.8
Shugard Floppy Disk	0.75	1.6
Sykes Cassette (RS232)	0.6	1.3
Sykes Cassette (Minicomputer Interface)	0.06	1.9

Some independent manufacturers supply interfaces designed specifically for certain device models and miniprocessors. Some firms do custom interfaces on a demand basis.

It should be pointed out that add-on/add-in memory modules are often considered peripherals.[17] They can be offered by the processor manufacturer or an independent vendor. They are used to enhance, augment, or replace existing memory within the limits of the CPU and software.

Mass Storage Devices

There are two basic types of mass storage: sequential and direct access. The sequential devices are usually slower than direct access ones, but they are simpler. They run on smaller, less sophisticated (thus less expensive) hardware and software. Sequential media vary from paper tape and punch cards to magnetic tapes (both cassette type and industry-standard).

The main characteristics of sequential media are speed and ease of use. Capacity does not enter in with cards (except for capacity of card hoppers) but can be a factor with lengths of tape. The data transfer rates determine the price of a device. The rates for reading and punching vary for paper tapes and punch cards, but magnetic tape read and write rates are the same. For all tape devices the speed can vary with the power of the tape drive; fast forward or rewind speeds also can vary and affect system efficiency. Tables 24-28 show the costs of various mass storage devices.

Paper tape and punch card devices.

TABLE 24—Paper Tape (Perforated) Devices (Five-, Six-, or Eight-Channel Tape)

Category	Speed (characters per second)		Price
Reader	Slow:	up to 100	$1,600 plus $1,000 controller
	Medium:	100-300	$1,800 plus $4,000 to $8,000 controller
	High:	greater than 300	
Punch	Slow:	up to 60	$3,000 to $9,900
	Medium:	61-120	$5,500 plus $4,000 to $8,000 controller
Reader/ Punch			$4,200

Note: The prices are only samples randomly chosen. The controller may be part of the mainframe system, and the manufacturer's unit can be used only on its equipment. There may be a separate controller, which must be purchased separately, or the device may have the controller installed in it and included in the price.

TABLE 25—Punch Card Devices (80-Column)

Category	Speed (cards per minute)		Price
Reader	Slow:	up to 300	$4,000 to $6,000 with controller
	Medium:	301-799	Up to $10,000 with controller
	High:	800-1400	$10,000 to $15,000 with controller
Punch	Slow:	up to 70	$1,900 plus controller ($750)
	Medium:	71-149	$9,000 to $12,000
	High:	150-300	$10,000 to $30,000
Reader/Punch	Read 300, punch 60*		$9,500 plus controller ($1,900)
Multipurpose unit (can include sorter)	Read 300, punch 60		$11,400 plus controller ($2,100)

Magnetic tape cassettes and cartridges. The tape cassette is the simplest magnetic tape device. It looks like an audio cassette and was introduced by Philips. The cartridge was developed by IBM; it has wider tape (therefore more tracks) and more capacity. An article in *Modern Data* compared the two forms (Tables 26, 27).[18]

---

[16] J. Egil Juliussen, "The Cost Outlook for Peripheral Controllers," *Mini-Micro Systems* 10 (January 1977): 64.

[17] "Peripherals Make the Mini," p. 36

TABLE 26—Comparison of Magnetic Tape Cassettes and Cartridges

	Cassette	Cartridge
Price of drive with read/write and motor control electronics	$450	$700
Price of a unit (cassette or cartridge)	$ 8	$ 20
Average capacity of media	5.4M bits (2 tracks)	23M bits (4 tracks)
Average transfer rate	8K bps	48K bps

TABLE 27—Tape Cassettes

Read/Write Speed	Rewind Speed	Density	Price
Ranges from 0 to 30 ips	20 to 140 ips	120 to 1000 bpi	$1,200 to $2,700 single drive $2,300 to $3,400 dual drive

Industry-standard magnetic tape. Magnetic tape on open reels of 7½- or 10-inch diameter is the common tape for both large-scale computers and minicomputers. The tapes have seven or nine tracks and are recorded at 800 or 1600 bits per inch (bpi). The read and write speeds range from 10 to about 75 inches per second, or 10-100 thousand bits per second (10-100 KB). Some devices come as clusters composed of multiple drives and controller units. Prices vary according to combinations of variables present (Table 28).

TABLE 28—Industry-Standard Tape Drives

Tracks	Density	Read/Write Speed	No. of Drives	Price
7	800 bpi	12.5 ips	1	$ 7,950
9	800 bpi	12.5 ips	1	7,950
9	800 bpi	12.5 ips	1	8,255
9	800 bpi	10 KB	1	8,600
9	800 bpi	96 KB	1	20,400
9	800 bpi	20 K BPS	1	7,850
9	800 bpi	40 K BPS	1	10,500
9	800 bpi	80 K BPS	1	13,000
9	1600 bpi	20 K BPS	1	17,300
9	1600 bpi	40 K BPS	1	21,900
9	1600 bpi	80 K BPS	1	26,700
9	800 bpi	18 K B	3	26,960
9	800 bpi	18 K B	4	32,160
9	800 bpi	37.5 ips	1	16,000
9	800 bpi	37.5 ips	2nd	10,250

Direct access storage devices. Direct access mass storage uses disks (diskette or floppy disk, fixed-head disk platter, movable-head disk cartridge, movable-head disk pack). Floppies are better compared to tape cassettes than to the large disk devices.[19] The price of the drive is about the same as the tape cartridge drive; the floppy itself costs the same as a tape cassette and its capacity is smaller than that of a cassette or cartridge. The transfer rate is significantly faster for the floppy:

Cassette	8K bps
Cartridge	48K bps
Floppy	250K bps

Some floppies are compatible with IBM equipment, and some are noncompatible (either in the data format or the plug for the drive); there are single; double density floppies.[20]

Each device comes with a drive and a controller. The units come in single, dual, or triple drives:

Single drive and controller	$2,800 to $4,500
Dual drive and controller	$3,750 to $6,000
Triple drive and controller	$6,200 to $8,000

Fixed-head disks. The fixed-head or head-per-track disk is the fastest but has the most expensive per unit capacity. The total capacity available is less than that of a movable-head disk. A unit with a 512K-byte capacity costs $10,880. One movable disk pack has a 20M-byte capacity and costs $9,500.

Disk cartridges. A disk cartridge is a movable-head disk in a unit of one or two platters. Single disk cartridges are removable which means the files can be transferred physically from system to system. Double disk cartridges are often combinations of one fixed disk and one removable disk. A number of characteristics can be used to describe disk cartridges, but only a few are significant:[21]

Drive capacity: storage capacity in megabytes using unformated data

Average access time: speed in positioning to get ready to read or write (expressed in milliseconds).

Transfer rate: read or write speed (expressed in kilobytes per second)

---

[18] Stephen A. Caswell, ed., "Cassette Drives and Systems," [illegible]

[19] Ibid.

[20] Dan M. Bowers, ed., "Floppy Disk Drives and Systems (Part 1: Historical Perspective)," *Mini-Micro Systems* 10 (February [illegible]

[21] Barbara A. Reynolds, ed., "Removable Disk Cartridge Drives," *Modern Data* 9 (January 1976): 19.

TABLE 29—Disk Cartridges

	Capacity	Price	Access Time	Transfer Rate
Low Capacity	2.5M bytes	$5,950 to $11,000	45–90 µs	180–312K bytes/s
Medium Capacity	4.8 to 6.25M bytes	$6,000 to $14,000	42–90 µs	150–312K bytes/s
High Capacity	9.6 to 12.5M bytes	$7,500 to $12,500	38–90 µs	181–312K bytes/s
High-Density (3000–4680 bpi), High Capacity	25 to 26M bytes	$15,000 to $24,500	33 µs	312–937K bytes/s

Bit density: the common density is 2200 bits per inch, but some newer disks have densities of more than 4000 bpi.

Price: prices include the drive and the controller.

Prices are related to the capacity of the cartridges (Table 29). The capacities range from 2.5M bytes to 10M bytes with several exceptionally larger units available.

Disk packs. The disk pack is the largest capacity unit. Within a unit there can be 10 or more platters that are read by a movable head drive. Some disk packs add a second drive. The same elements are important for a disk pack as for a disk cartridge: capacity, access time, and transfer rate. Table 30 compares prices of various disk pack units.

TABLE 30—Disk Packs

Capacity	Number of Drives	Price with Controller
20M bytes to 30M bytes	one	$11,500 to $18,500
	add second drive	$10,000 to $15,300
40M bytes	one	$35,000 to $40,000
80M bytes	one	$28,000 to $30,000
	add second drive	$15,000
95M bytes to 175M bytes	dual	$58,000 to $75,000

Man/Machine Interface Devices

Man/machine interface devices output and input information in human-understandable form. These devices differ significantly in input/output capabilities. Some are input only, others output only, and still others handle both input and output. Some are hardcopy, some display, and some both.

Punch card readers and paper tape readers are input only devices and were discussed under mass storage devices. Printers are the most common output only devices.

Printers. Printers vary according to several characteristics:

Speed — ranges from 10 characters per second to 18,000 lines per minute

Price — varies from approximately $100 to $310,000

Method — impact or nonimpact

Mode — character at a time in a serial format or line at a time in a parallel format

Character images — shaped or dot matrix

Character sets — 64 (upper case), 96 (upper and lower case) 128 (upper and lower case plus special characters)

Type of paper and paper feed — heat-sensitive paper, friction feed roller or sprocket feed roller (these all affect the ability to print special forms and/or multiple copies.

Printer units contain an interface to the computer, a power supply, and control electronics (sometimes including a buffer).

Modern Data prepared a summary of typical printer prices at the end of 1975 (Table 31).[22] It gives an overview of the printer market.

---

[22] Irving L. Wieselman, "Printer Technology and Its Future: A [illegible]," *Modern Data* 8 (November 1975): 34.

TABLE 31—Typical U.S. End User Printer Equipment Prices and Speeds

					Comments
Impact Character **(Shaped Character)**					Multiple copies possible, preprinted forms possible
Speed (characters/second)			10–30	60–120	
Price ($000)			1–4	4–6	
Impact Character **(Dot Matrix Character)**					Multiple copies possible, preprinted forms possible
Speed (characters/second)			30–100	115–660	
Price ($000)			2–8	4–12	
Impact Line **(Shaped Character)**					Multiple copies possible, preprinted forms possible
Speed (lines/minute)	90–250	300–700	800–1800	2000	
Price ($000)	3–17	3–51	35–80	74–112	
Non-Impact Character **(Dot Matrix Character)**					Requires special paper, single copy only
Speed (characters/second)			10–240		
Price ($000)			1–4		
Non-Impact Line **(Dot Matrix Character)**					Requires special paper, single copy only
Speed (lines/minute)		300–600	1000–3000	4000–18,000	
Price ($000)		5–10	7–13	165–310	

Teletypelike Terminals. Teletypelike terminals have keyboards for input; they print on paper for output. Some character printers with keyboards included could qualify as input/output terminals. Many of the same characteristics of the printer apply to these "teleprinters" (impact and nonimpact printing, speeds from 10 to 120 characters per second, upper case and upper/lower case, 80 characters per line or less and up to 132 characters per line). Table 32 gives price ranges for various types of Teletypelike terminals.

TABLE 32—Typical Teletypelike Terminals

Line Width	Print Speed	Price
80 characters per line or less	Up to 30 characters per second	$1,500–$5,000
More than 80 characters per line	Up to 30 characters per second	$2,000–$5,000
Generally 80 or more characters per line	Up to 120 characters per second	$2,500–$5,500

Keyboard/Display Terminals. The most common keyboard/display terminals use a CRT for display. Although these have some characteristics in common with hardcopy printers, other characteristics are peculiar to display units.

Character format (size of the dot matrix used to form the characters) — Common patterns are 5 by 7 or 5 by 9 (the more dots, the more legible)

Display characters (total number displayed on the screen) — Usually 640–2000 characters.

Format (number of characters per line and number of lines per screen; the product of the two equals the total number of display characters) — Common formats are 80 characters by 12–24 lines.

Speed — The speed is equal to the transmission rate because there is no slowdown due to mechanical devices. If the CRT is connected as a TTY replacement, the controller will transmit the data one character at a time, with a speed that varies with the keyboard operator. The printer will work at a rate comparable to that of a Teletype. The more sophisticated controllers allow transmission in blocks at communications speeds, e.g., 9600 BAUD.

Character set — Upper case (64) or upper/lower case (96), a few with special characters capability (128)

Special features — Buffered or unbuffered, page mode, forms fill-in formats; function keys; line and character edit capability, double width characters; controllable cursor, reverse video; blinking, built-in modem, acoustic coupler; portable, and color

Prices vary according to the combinations of features and elements. The bottom level is around $1,000, and the top is more than $12,000. The top end has features that place it in the intelligent terminal class.

Some CRT controllers handle multiple units as a cluster. Some units have the capacity to plug in a printer for hardcopy output.

Software

In many cases, systems software and an operating system are included with the purchase of a minicomputer processor. "Extras," in the form of utilities, compilers, and communications packages, are sold separately. Some manufacturers offer applications software packages designed for special purposes. Many companies will not sell these packages, but only lease them, to protect the proprietary nature of the packages. This arrangement usually allows the user to receive the latest, most up-to-date version of the software.

The following prices are offered from main mini manufacturers' business systems lists. They would not necessarily reflect prices from a systems house.

Systems Software

Utilities	$ 15/month rental
Communications	$700 purchase
Operating system	$ 15/month rental
Compiler	$ 25/month rental
	$ 20 to $ 90/month rental
	$ 11 to $115/month maintenance

Applications Software

Purchase	$ 500 to $3500
	$ 600 to $ 750 plus $12.50/month maintenance
Lease fee	$ 175
	$ 250 to $1450 plus $10/to $65/month
	$2500 to $4600
Monthly rental	$ 75 to $ 420/month

Systems

Many of the major minicomputer manufacturers offer entire systems. Such a system includes a complete hardware configuration, a software package, and sometimes even application packages. The system is offered at a price that is less than that of a piece by piece purchase. This is particularly true for intelligent terminals, which often have all the components in a single cabinet. Some systems are for general applications, but some are configured for special purposes. For example, Raytheon has a system especially designed for distributed processing. Digital Equipment Corp. has a word processing system in the Datasystem 300 series; and Burroughs has an entire series of systems for work-station based data entry/inquiry.

One type of system — the small business computer — has received much attention and has been widely developed by the industry. These systems are "generally characterized by purchase prices between $5,000 and $100,000 and by a strong orientation, in both equipment and software, toward conventional business data processing application."[23] They are developed by the main minicomputer manufacturers or by independent system houses or turnkey vendors. This market was assessed at $1 billion in 1976, and so has received a great deal of attention from the trade press. Articles that survey and review small business systems appear routinely, and looseleaf services such as Auerbach's *Buyer's Guide to Business Minicomputer Systems*[24] offer current information on the market.

One must be very careful in studying systems as described in the press. Not every vendor "bundles" his system the same way and not every reviewer presents the data the same way. For example, the *Datamation*[25] survey described the Datapoint Model 2200 one way, and *Mini-Micro Systems*[26] described it quite another:

Datamation	Mini-Micro Systems
Model Highlights	*CPU*
No. installed/date:	Word length: 8
9000 since April 1972	Memory capacity: 4–16K
Multiprogramming: No	
Communications: 8 lines	*Data Entry Terminal*
Applications: banking,	Video
insurance, government,	
accounting	*Card Reader*
	80 column (300 cpm)
Processor (Datapoint)	
Internal storage	*Disk*
4K to 16K MOS	Capacity: 0.256–40M bytes
(3.2 μsec)	
Word size/add time	*Mag Tape*
8 bits/4.8 μsec	Cartridges (2 units)
	Reel
Mass Storage	
Disks: floppy,	*Printer*
cartridge, pack	Serial 132 columns
Access methods:	30/330 cps
random sequential,	Serial 120 columns
indexed sequential	120/240 cps
Magnetic tapes: reel-	Line 132 columns
to-reel, cassette	300/600 cps
Peripherals	*Communications*
10-key, card, serial	Asynchronous and
printer (120 cps),	synchronous
line printer (to 600 lpm)	

---

[23] Steve A. Bobick, Edmund J. Armon, and Arthur W. Yerkes, "Survey of Small Business Computers," *Datamation* 22 (October 1976): 91.

[24] *Auerbach Buyer's Guide to*. For information on minicomputers for libraries, see *Advanced Technology Libraries*, 1972.

[25] Bobick, Armon, and Yerkes, "Survey of Small Business Computers," p. 96.

[26] Malcolm L. Stiefel, ed., "Small Business Computers," *Mini-Micro Systems* 9, July 1976 [illegible]

Software
RPG BASIC, assembler, DATABUS, SCRIBE, DATA FORM, DATASHARE, accounting package, DBMS
Basic Prices
$8,571 ($216/mo., 2-yr.)
Software and support extra

Programming Languages
DATABUS, BASIC, RPG II, assembler
Modes of Operation
Online batch
Typical Prices

Purchase	$45,000
Monthly rental	$1,200
Maintenance	$250

How does one compare these data? The *Datamation* price is for a basic system (probably the processor only, with the lower end of the memory). The cost of mass storage devices, peripherals, and data communications equipment would be extra, as would the cost of software. The *Mini-Micro Systems* price includes more and is more representative of what this system would actually cost.

An interesting comparison can be made between two entries in the *Datamation* survey:[27]

	Applied Data Communications	Datapoint
Model Highlights	*Model 101*	*1100 Series*
No. installed/date	Introduced in August 1976	6,000 since January 1974
Multiprogramming	No	No
Communications	One line (bisynch 2780, 3780)	One line (2265, 2741, 3780), HASP
Applications	Manufacturing, inventory, text processing	Banking, insurance, government, accounting
Processor	Datapoint 1100	Datapoint
Internal Storage	12K MOS (1.6 μsec)	16K MOS (1.2 μsec)
Word size, add time	8 bits, 16 μsec (5 digits)	8 bits, 48 μsec add time
Mass Storage		
Disks	Floppy	Floppy
Access methods	Random, sequential, indexed sequential	Random, sequential, indexed sequential
Mag tapes	Cassette	Reel (reel)
Peripherals	10-key, serial print (to 165 cps), line print (300 lpm)	10-key, card, serial print (120 cps), line print (to 600 lpm)
Software	BASIC, assembler, Databus II, accounting package	RPG, BASIC, assembler, DATABUS, SCRIBE, DATAFORM, DATASHARE, accounting package, DBMS
Basic Prices	$29,900 ($800/month), sft and support included	$8,571 ($216/month), sft and support extra

The differences here stem from what is included in the basic system. For the minicomputer manufacturer Datapoint the basic price is for the processor and only a few peripherals. No software is included. For the systems house vendor Applied Data Communications the basic system is more inclusive and includes the software. Again, careful comparisons must be made.

EVALUATION

In preparation of an RFP, the criteria to be used in reviewing the proposal are drawn up and often are included in the RFP. The criteria should cover not only the hardware and software, but also the manufacturer and/or vendor's track record, the previous experience of the system (or hardware), and the responsibilities for maintenance of the system. The University of Chicago suggested that the following points be covered as the basis for review.[28]

1. Reputation of manufacturer/supplier — This criterion is the most subjective, but in terms of our requirements, is the most important. The hardware must be widely available and supported over the life of the system, which suggests the importance of dealing with an established vendor. The product must be currently demonstrable and of proven reliability.
2. Maintenance — Maintenance support must be available locally and should be of proven reliability.
3. Communications hardware architecture — The major task of the minicomputer system is that of driving approximately forty-eight lines with various transmission characteristics. The communications hardware must be capable of handling a variety of device types and communications characteristics and of processing the anticipated load, and have sufficient reserve power for expansion. Careful attention must be paid to whether the communications hardware controls transmission on a character-by-character basis (programmed I/O) or on a message basis (direct memory access).
4. Communications software support — In conjunction with the communications hardware, an extremely desirable feature is the availability of communications software.
5. Peripheral devices — Because the front-end computer system provides backup when the main computer is down, a heavy burden is placed on the peripheral devices, most notably on the disk drives which hold several processing files. The mechanical components of the system have the least reliability. Therefore, the disk and tape drives proposed must be devices previously installed and of high reliability.
6. CPU architecture — The state of the art in processor design has reached a point that many fast, reliable minicomputers are on the market. Therefore, CPU to CPU comparisons should be minimized as much as possible.

The points are often weighted as to relative importance. (See L.C. weighting schedule in Appendix B.)

Within each category further evaluation is necessary. Various schemes have been described in the literature. Many of these schemes are based on lists of pertinent factors weighted according to their relative importance to the application. Formulas are applied to compare the results of each proposal's weighted scores.

---

[27] Bobick, Artman, and Yerkes, "Survey of Small Business Computers," pp. [illegible]
[28] Payne, "University of Chicago Library Data Management System," p. [illegible]

At an American Institute of Industrial Engineers (AIIE) conference on minicomputers, John Hughes suggested use of a vendor decision matrix.[29] The criteria he suggested were:[32]

1. Processor Architecture
    Word length
    Memory path width
    Interleaving ability
    Number of channels
    I/O channel data rate
    CPU cycle time
    Memory cycle time
    Number of registers
    Instruction set
    Main memory capacity
    Main memory type
2. Software
    Operating systems
    Data base/file management systems
    High level compiler
    Debugging aids
    Application packages
    Utility programs
    Communications handler
    Multiprogramming
    Editing
    Overlays
3. Maintenance
    Equipment Reliability: MTBF and MTTR
    Onsite FE [field engineer]
    Expense
    Spare parts
    Diagnostics
    Organization
    Preventative maintenance
    Contracts available
4. Marketing Support
    System documentation
    Programming manuals
    Custom programming
    SE support [system engineer]
    Training
5. Company Viability
    Time in business
    Profit picture
    Research and development
    Installed base
    User groups
    Future plans
6. Costs
    Purchase price
    Discount structure
    Lease basis

Once the criteria are established, the weights should be determined. Of the factors listed, some may be of no importance to the application; give them weights of zero. Those factors most important to the application should be given the top weight. The most common scale used is zero to ten. The weighting decisions can be aided by reviewing the hardware and software impacts discerned through use of the Design Model (Chapter Four). Auerbach reviewed the elements of hardware and described how each affects a system.[30] For example, if there is frequent data exchange between the CPU and the external environment (i.e., the peripheral devices), the interrupt efficiency is important and the machine chosen must process interrupts quickly and offer multilevel priorities.

Ollivier did more than subjectively assign weights at review time.[31] He broke down each factor in terms of possible responses and the value each should be assigned. This value is then multiplied by the weight. Table 33 shows an example. Although this appears to be a more scientific approach, a caution must be noted. Manufacturers use their own techniques to achieve the same purpose, and the factors are difficult to quantify precisely. For example, how would Bur-

TABLE 33—Weighting Scheme for Hardware Evaluation

Factor	Weight	Scoring Bases
Word size	10	4: 16 bits or more; 2: 12 bits; 0: 8 bits or less
Cycle time	6	4: 1 μs; 3–1: 1–2 μs; 0: 2 μs
Instruction set	5	4,3: Extensive; 2: Adequate; 1–0: Primitive
Addressing	4	4–0: Score one for each of the following: indirect, relative, indexed, direct to greater than 4096, or by addressing
Interrupts	7	4: 3 or more priority, no identification necessary, 3–1: adequate for 3 devices; 0: none quoted
Physical size	1	4–0: Subtract one point for each 5 inches over 11 inches

---

[29] John Lee Hughes, "Maxi to Mini: A Citibank Case History (Speech)," in *Minicomputers: The Applications Explosion*, ed. David E. Debeau, proceedings of the American Institute of Industrial Engineers Conference held in Washington, D.C., 17–19 November 1975 (Los Angeles: Management Education Corporation, 1975).

[30] *Auerbach on Minicomputers* (New York: Petrocelli Books, 1974), p. 82.

[31] Ollivier, "Technique for Selecting Small Computers," pp. 94–95.

[32] William Barden, Jr., *How to Buy & Use Minicomputers & Microcomputers* (Indianapolis: Howard W. Sams & Co., 1976), pp. 85–87.

rough's variable-word-size devices be compared with machines that use 8-bit, 16-bit, 24-bit, or 64-bit words? Barden suggests that the best way to overcome this problem in the area of speeds is by preparing and running benchmark programs for the various types of minicomputers.[32] The sets of benchmarks can vary according to the application. He illustrated benchmark programs run by the industry: I transfers 100 bytes of data from one block of memory to another; II converts a 6-digit ASCII octal value to a binary value of 16 bits; and III searches an 80-item string of characters for a given character. Table 34 illustrates the use of these benchmarks.

TABLE 34—Instruction Speed Benchmarks

Computer	Benchmark* I	II	III	Average
8008–1 μp chip	69,215	1,767.5	5,542.5	25,762
8080 μp chip	7,854	1,018	3,546	4,139
MC6800 μp chip	3,918	291	807	1,672
F8 μp chip	5,247	439	899	2,195
MCS6502 μp chip	2,604	447	440	1,162
PDP–8/E	1,158	125.8	668.8	651
Nova 800	688	58.8	252.6	334
CAI LSI-3/05 (core)	4,006	339	1,248.75	1,865
Interdata 6/16	290.5	88.3	390.5	256

*Time in microseconds

A similar approach would be to use an existing industrywide comparison. A classic study was done by Butler in 1970;[33] the data are obsolete now, but the approach is still viable. He used data for 45 models as a basis for a price-to-performance ratio by using three equations for calculating: the hardware price/performance; the software price/performance; and the overall price/performance. He showed ratings for the 45 models based on overall price/performance values.

Some evaluation techniques stop when the weighting or weighting times value has been completed. This number can be said to provide a measure of comparison among various manufacturers.[34] The vendor with the most accumulated points is chosen for final evaluation. Ollivier suggests that some factors may be so critical to the system that their absence (a zero value or score) will eliminate the vendor from consideration no matter how high the system's total score was.[35] For example, if delivery is required in 45 days and the proposal offered 90-day delivery, the proposal would be worthless for this application.

Ollivier does not stop with the composite score. He advocates plotting cost against performance, using a minimum performance score and a maximum allowable cost figure as bases of comparison (Fig. 19).[36] In the figure it is obvious that system E is the best value, with system A in second place.

Turnkey Systems

If a packaged or turnkey system is included as one of the proposals (or if all of the proposals are for packaged systems), how is it evaluated? Delanoy, writing on the use of library minicomputer turnkey systems, advised the following.[37]

> In considering a minicomputer. . ., the most important characteristics are that the machine is part of an established and growing product line; the machine as configured is suitable and adequate for your application; the manufacturer-supplied software meets your needs; and the manufacturer's installation and maintenance commitments are clearly specified.

Weisbrod also wrote of library packaged systems. He said four major questions must be answered.[38]

1. Do the functions included satisfy the needs of the prospective user?
2. Can the system be configured to handle the prospective user's processing load?
3. What equipment is required, either for purchase or for rental?
4. What is the system's cost?

He pointed out that these are the same questions that must be answered for a system developed in-house. Therefore, in evaluating a packaged system one uses the same procedures as for any other proposal, but with more emphasis on the software.

"Software, not hardware, is the pivotal element in system selection."[39] That is because the real power and effectiveness of any system rests on its software. For a system purchased for in-house applications program-

---

[33] J. L. Butler, "Comparative Criteria for Minicomputers," in A *Practical Guide to Minicomputer Applications*, ed. Fred F. Coury (New York: IEEE Press, 1972), pp. 77-92. See also Becker and Hayes' parameters of equipment which evaluate the efficiency of single components viewed in the context of defined operations with no other system considerations involved. The three parameters are cost/operating time, and unit of operation used in the formula C/T: N. Becker and Hayes, *Information Storage and Retrieval*, pp. 295–324.

[34] George A. Khtaian, "Cost: Vendor Evaluation System," *Journal of Systems Management* 26 (August 1975): 14.

[35] Ollivier, "Technique for Selecting Small Computers," p. 96.

[36] Ibid., p. 97.

[37] Diana Delanoy, "Technology: Present Status and Trends in Computers," in *Library Automation: The State of the Art II*, eds. Susan K. Martin and Brett Butler, papers presented at the Preconference Institute on Library Automation sponsored by the Information Science and Automation Division of the American Library Association at Las Vegas, Nevada, 22–23 June 1973 (Chicago: American Library Association, 1975), p. 21.

[38] David L. Weisbrod, "Acquisitions Systems: 1973 Applications Status," in *Library Automation: The State of the Art II*, eds. Susan K. Martin and Brett Butler, papers presented at the Preconference Institute on Library Automation sponsored by the Information Science and Automation Divison of the American Library Association at Las Vegas, Nevada, 22–23 June 1973 (Chicago: American Library Association, 1975), p. 91.

[39] Siefel, "Small Business Computers," p. 56.

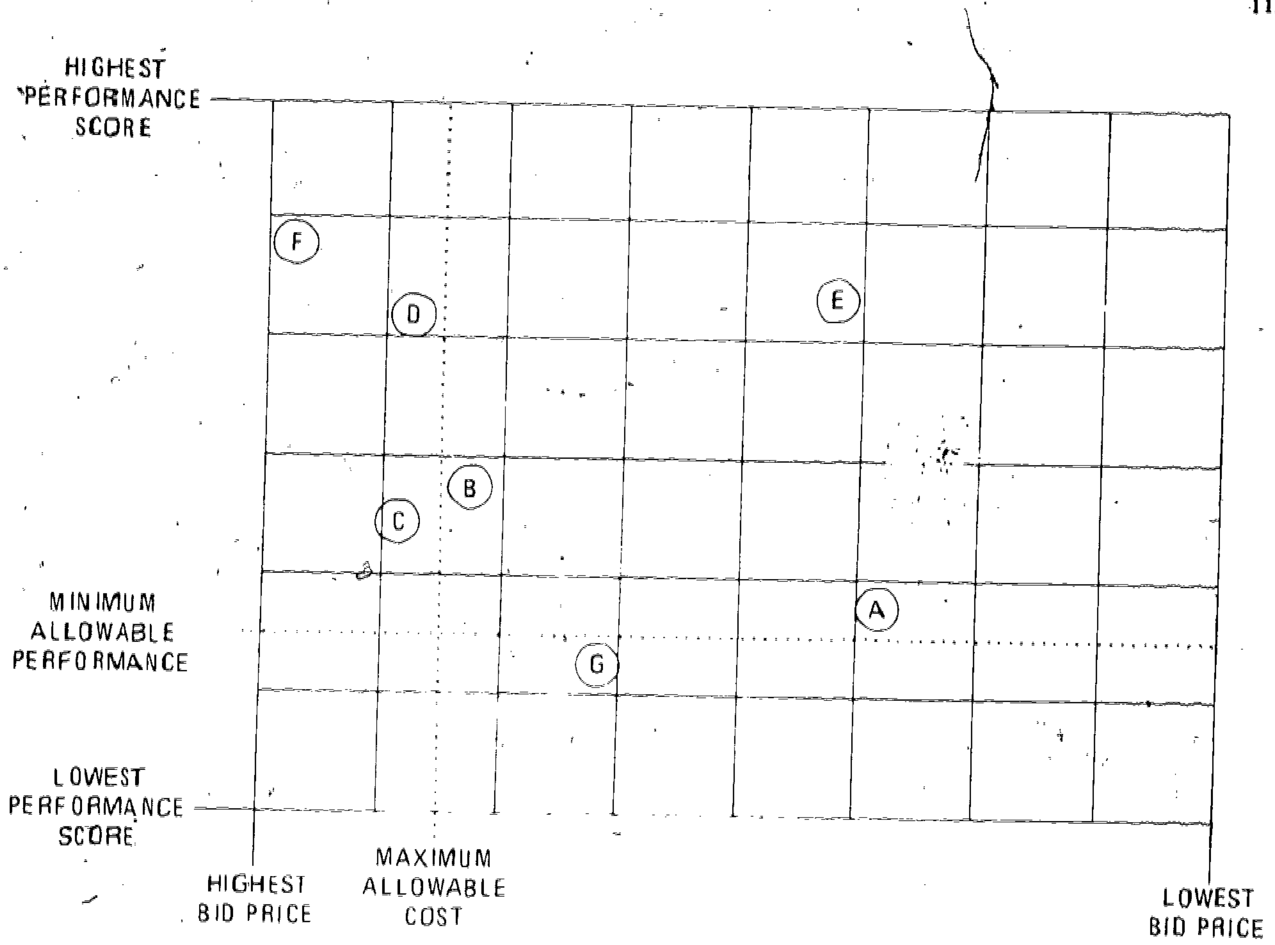


FIGURE 19. Cost versus performance evaluation grid

ming, the systems support software is of critical importance. It affects the development time and the efficiency with which the programmer can work. For a packaged system, the emphasis must be on the applications programs provided.

Stiefel states that two problems are shared by all software packages in the world:[10]

> First, it is highly probable that a given package won't precisely fit 100 percent of the requirements of a given business. After discovering the deficiencies, the user either can correct them (i.e., bring in a consultant) or live with them. The extent to which a user can afford to compromise is a function of the seriousness of the discrepancies.
>
> * * * * *
>
> Ideally, the software should have the flexibility to be altered by the user, perhaps by changing parameter tables to operate in a given environment, without causing a major upset in the using organization. Unfortunately, most packages have limited flexibility, some have none.
>
> The second problem with the precooked software approach is its lack of operating efficiency, measured in terms of usage of system resources (disks, core memory) and running time. Generality is *always* achieved at the expense of efficiency. Running time optimization isn't important in many small business applications where input is entered so slowly and so infrequently that virtually any reasonable response time will be acceptable. But the problem is, if resources are used inefficiently, the user may be forced into acquiring a larger configuration than he really needs, just so he can use a certain package.

Some vendors offer the service of preparing custom modules or customizing existing programs. Others do not which means the library must try to make the changes with their own and/or agency personnel, or they have to hire consultants. It is inadvisable to attempt some changes. Depending on how the programs are designed, some changes can be taken care of without much trouble. But some changes would jeopardize the basic logic stream and continuity of the program. These kinds of changes should be avoided at all costs because the number of changes in a program is inversely proportional to the success (efficiency and effectiveness) of the program.

---

[10] Ibid., pp. 56-57.

REVIEW OF OTHER RESOURCES/REQUIREMENTS

Besides the obvious hardware and software requirements for a system, other types of requirements or resources must be considered. The details of these elements may affect the final selections.

Physical Requirements

Space

The site for the various pieces of equipment must be considered. Minis do not have to have a special room with raised floors and special air-conditioning and humidity control. Because they are small, they can usually be fitted into the existing environment without much trouble. If rearrangement of furniture or remodeling is necessary, for the new equipment or for a new work flow, it should be established early and plans made to make the changes. In this same vein, if the equipment is to be placed in more than one location, provision must be made for laying cables or lines to connect the components (if a hardwire connection is used). Holes in ceilings or walls or channels for the floor may be required.

Although minicomputer equipment can operate under standard air-conditioning, some devices generate more heat than others, and adjustments may have to be made. Specifically, disk packs create quite a bit of heat; if they are used in multiple arrangements, the heat increase may be significant.

Power

Most minis operate on regular 110 volt electrical power. It is often recommended, however, that each device or each system power supply be on a separate line, often with a separate circuit breaker, to lessen the chance of surges. That doesn't mean one line per component, as many devices can be connected to a single power supply unit, but two or three lines are not uncommon. Special electrical rewiring may have to be completed before the system can be installed. Plans for this work should be made as soon as the exact locations and equipment requirements are determined.

Noise

While noise is not a "physical requirement," it is certainly a fact of life and must be considered, not only in the public service area but also in the staff work areas. The mainframe is not very noisy. The most disruptive noise comes from the fans used for cooling the processing unit. In addition, some disk drives whine and some tape drives click as the file is read. This noise should be within the tolerance levels for an office environment. The main noise comes from peripherals such as card punches, printers, and even hardcopy interactive terminals. Some libraries have taken special steps to deal with the problem of noise, as detailed by the editors of *Online* in their discussion of interactive terminals.[41]

> Some libraries go to great lengths to avoid noise. The Congressional Research Service of the Library of Congress, for example, built a special sound-insulated booth for its CRT and high speed impact printer. The U.S. Department of State's library simply put its Centronics printer in a closet and got a long cord to go to the CRT. And the Army Library in the Pentagon put a plastic noise-suppressing cover over its impact printer.

Maintenance

Although maintenance was mentioned in the discussion of hardware and software evaluation, additional comments are in order. In general, minicomputers are very reliable pieces of equipment.[42] Mean time to failure for minis is measured in months instead of hours. Preventive maintenance is reduced to the barest of minimums for the mainframe. The peripherals do not have quite so good a record. For instance, printers, especially impact printers, often require adjustments to provide proper print alignment. Tape drives should be cleaned periodically, but that is often done with a bottle of alcohol and a cotton swab.

The fact that maintenance is not as frequent for minis does not lessen its importance. The history of the mini market has been one of insufficient support after installation. One wit said that minis weren't delivered—they were abandoned. Maintenance is particularly complicated when components from several manufacturers have been assembled and programmed by a systems house. In such cases, maintenance often is contracted out to another firm, which functions as a maintenance company. It is important to know where the maintenance unit is located and what the turnaround time is on the equipment. Because minis are modular, often a board can be pulled and a new one substituted while repairs are being made.

---

[41] [Editor's note] in "The Intelligent Person's Guide to Choosing a Terminal for Online Interactive Use," by Mark S. Radwin, *Online* 1 (January 1977): 16.

[42] Eugene D. Lourey, "Systems Design for a Minicomputer-Based Library Data Management System," in *Applications of Minicomputers to Library and Related Problems*, ed. F. Wilfrid Lancaster, papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974 (Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974), pp. 185-6. Weitzman gave a chart on typical values of MTBF (mean time between failures) for minicomputer components and peripherals. Only line printers (400 to 800 hours), card punches (200 hours) and impact keyboard printer terminals (300 to 1600 hours) have values in hundreds of hours; all of the other items have values in thousands of hours. Cay Weitzman, *Minicomputer Systems: Structure, Implementation, and Application* (Englewood Cliffs, N.J.: Prentice-Hall, 1974), p. 403.

One other area of maintenance must be considered — software maintenance. Programs are rarely ever "finished" and completely "debugged." Arrangements should be made for the software to be maintained as necessary over the entire period of use.

Personnel

Staffing patterns change when an automated system is introduced. Sometimes a special implementation team is required if extensive initial data input is required for building a file or files.

Because the minicomputer is located in the library, a library staff member or members must be assigned the role of computer operator. In many libraries, the input operators can be trained to do the operator's tasks along with their other duties. The duties of the operator vary with the system configuration, the operating mode of the operating system software, and the nature of the application. For the simpler systems with simple executive monitors, the operator must attend the job flow fully. With a single batch operating system, the software does the job control. Cards still must be loaded to be read, tapes must be mounted, and paper in the printer must be aligned. In any system, the equipment must be powered up, the program must be loaded, and the system must be shut down each day. Files of tapes, disks, or cards must be stored and a log must be maintained to keep track of them. The persons responsible for these tasks must be trained thoroughly by the system vendor. There should be comprehensive manuals for the training period and for reference during ongoing use. Complete system documentation should also be provided.

Time Frame

One recurring [illegible] been poor estimation of [illegible]ing a system. It always has taken longer than anticipated. Certain variables affect the time it takes to develop a system. For example, it is quicker to buy a turnkey system than to develop a new system. Also in developing a new system, it is quicker to contract the work to a commercial source than to do it in-house, if for no other reason than that a firm delivery date can be negotiated as part of the contract. As mentioned earlier, the extensiveness of the system software and the presence or absence of a compiler can affect the programming time.

The real key to fast, efficient development is using an experienced minicomputer programmer. Such a programmer will have a "minicomputer philosophy" and will not try to implement large-scale computer systems on the smaller machine. The programmer must shed his "batch" cloak and think "interactive online," for that is the mode to which minicomputers are conducive. If there is no such programmer on the agency data processing staff, it may be worth the time to train a programmer in minicomputers.

THE DECISION

When all the data have been gathered, reviewed, and evaluated, the time for decisionmaking has arrived. When the selection is made, the procurement procedure must get underway. As a Federal agency, the library must comply with Federal procurement regulations and procedures for ADP equipment.

Federal ADP Procurement

The longstanding policy of the Federal Government, as expressed in OMB Circular A-76, is reliance on the private sector for goods and services. This policy as it relates to ADP procurement is expressed in the Brooks Act (Public Law 89-306), "Procurement of ADP Resources by the Federal Government," which emphasizes that ADP requirements should be procured in a competitive manner.

Three agencies have specific responsibilities relating to ADP procurement. The Office of Management and Budget (OMB) is responsible for establishing policy and ensuring that the policies are being followed. OMB sees that agencies rely as much as possible on commercial ADP services and ensures that ADP equipment (when authorized) and services are procured in an efficient and orderly manner. Approval of new or expanded in-house ADP facilities must be obtained in accordance with the "new start" requirements of Circular A-76 before submission to OMB of the agency's budget request that includes the necessary funds.

The General Services Administration (GSA) administers ADP procurement. It provides management guidance and is the authority for approving noncompetitive procurements and delegating procurement authority to user agencies. GSA does not, however, become involved in determining whether an agency requirement is legitimate. GSA publishes lists with prices of equipment and types of services on its supply schedules. In general, any procurement, either on or off schedule, with a one-time or an annual cost of $50,000 or more, must be authorized by GSA.

The National Bureau of Standards is the third agency involved in ADP procurement. It provides necessary hardware and software standards.

Although some minicomputers appear on GSA schedules as office equipment, minicomputers are generally considered ADP equipment. In November 1975, the Commissioner of the Automated Data and Telecommunications Service of the General Services Administration discussed trends in minicomputer procurement. Just as large-scale computers and peripherals were purchased under centralized and volume procurements, "volume procurements of minicomputers must and will occur. The question to be resolved is what is the best method to use for these acquisitions."[43]

> In order to determine the best method for procuring a product or service, we in ADTS have to be aware of the marketplace and the industry . . . . We've gone from the mandatory requirements contracts for the entire government when buying peripherals, to indefinite quantity contracts with voluntary usage when buying minicomputers. We have not created new techniques, but we have been selective in the techniques we've used. The government has been the systems integrator, or industry has provided the systems integration, depending on the environment.
>
> I don't want to leave you with your thinking our evolution is complete. In our future volume procurements for minicomputers, we will specify the maximum amount that can be ordered under each indefinite quantity contract. That number will not exceed 8 to 10 times the guaranteed minimum order. This will allow an agency the flexibility it needs to meet its needs through the procurement, yet is reasonable for industry.[44]

A year later the whole emphasis for all ADP procurement had shifted. Instead of discussing methods of procuring equipment for Government use, the officials were speaking of relying on the private sector for products and services. This shift was largely a result of the Congressional hearings and report on the administration of the Brooks Act during its 10-year life. It was found that the true intent of the Brooks Act was not being carried out and recommendations were made to bring the course of Federal procurement back into line, i.e., use of the private sector and of fully competitive procurement.

In a paper delivered to the AIIE in November 1976 an OMB spokesman expressed the policy on use of the private sector as follows:[45]

> In the case of ADP, we feel that the policy is best served when agencies can satisfy their requirements by contracting for complete ADP services, and that Government acquisition and operation of facilities should be limited to those situations which can be justified as being in the national interest under the guidelines of Circular A-76.
>
> * * * * *
>
> In order to evaluate the feasibility and relative economy of contracting for ADP services, Government needs must be presented in a format suitable for evaluation and for use in a contract statement of work. This requires a fundamental change in the philosophy of requirement definition, since the in-house performance of ADP services has led agencies to define their needs in terms of the hardware, software, and personnel necessary to do the job. Requirements formulated on this basis cannot be properly evaluated to determine if the unique nature of the work necessitates in-house performance, nor can they be used to solicit competitive proposals from industry. The first step, therefore, in achieving greater reliance on private sources for ADP services is for Government agencies to develop and state their requirements in terms of the services performed, rather than the facilities needed to provide those services.
>
> Once the actual work being performed, or proposed to be performed, in a Government ADP facility is identified and described as a service, it can be examined to determine whether there is a compelling reason that it be done in a Federal facility by Federal employees. The fact that work may involve classified data, be part of the agency's basic program, or require privacy safeguards will not necessarily justify Government performance — industrial facilities have been cleared for classified work for many years and commercial data centers can meet privacy requirements as well as Government facilities. There must be a document justification in accordance with one of the exemption criteria of Circular A-76 to justify initiation or continuation of Government activities providing ADP services.
>
> * * * * *
>
> [illegible] the program calls for a significant change in Government practice, it does not reflect a change in policy — rather it is an application of long standing policy to an area where that policy was not always appropriate under past conditions.

At the same conference the trend toward miniaturization [illegible] use of minicomputers [illegible] was discussed by the Director of the Office of ADP and Telecommunications Management of the Department of the Interior.[46]

---

[43] Theodore D. Puckorius, [illegible] ADP Equipment [Speech]," in *Minicomputers: The Applications Explosion*, ed. David E. Debeau, proceedings of the American Institute of Industrial [illegible] Conference held in Washington, D.C., 17-19 November 1975 (Los Angeles: Management Education Corporation, 1976).

[44] Ibid.

[45] William D. Russell, "Applications [illegible] *ADP Practices*, ed. David E. Newman, proceedings of the American Institute of Industrial Engineers Conference held in Washington, D.C., [illegible] November 1976 (Santa Monica, Calif.: Management Education Corporation, 1976), pp. 1[illegible].

> There seems little dispute these days that the future technology will be greatly influenced by the interaction between communications and miniaturization, resulting in an accelerated trend toward interrelated minicomputers forming distributed processing networks. If we believe the communicators and minimanufacturers, this approach will not only bring data processing closer to the user, but at considerably less money.
>
> * * * * *
>
> Let's look at the Government ADP environment from the perspective of the past, rather than the future. . . . My generalizations are that:
>
> a. There is a strong penchant for in-house processing capability.
>
> b. RFPs are geared more to hardware specifications than to functional specifications.
>
> c. There is a very strong reliance on costly benchmarks. The combination of hardware specifications and benchmarks gives us a warm feeling that we are acquiring the best configuration to do our job.
>
> d. We think bigness in terms of hardware. Most of us have grown up in the syndrome created by the maxivendors and we follow the American tradition of wanting the biggest and the best. We may be kidding ourselves, however, to think that bigness and best are synonymous in today's technology.
>
> * * * * *
>
> [Therefore, although] technology is offering potential cost savings through interconnected miniaturization . . . , our RFPs are benchmark dependent. . . . The marketplace has expanded but our traditional methods of procurement do not let the broadened marketplace respond — it does not respond to our traditional benchmark requirements.

The speaker went on to suggest a model procurement that would open up competition. It involved developing an RFP built around users needs, rather than around a predetermined data processing scheme for satisfying those needs. The total marketplace would be asked to respond by submitting a detailed plan, including schedules and estimated pricing scheme, for satisfying the user needs. A team of evaluators would then review the proposals and select the most feasible approaches and best qualified contractors (on a very subjective basis). From this evaluation at least two contractors would be selected for final bid. During this step, a benchmark would be tailored to fit the specific marketplace invited to final bid. "The final contract award then would be based on lowest overall cost and benchmark performance."[47]

This discussion was in terms of numerous minicomputers combining for distributed processing. What of minicomputers used singly as a standalone system? The report of the hearings on the Brooks Act specifically speaks of minicomputers in several sections. One recommendation states that competitive procurement of minicomputers (or service contracts) should be used if possible instead of interim upgrades, add-ons, or replacements for large-scale systems.[48] Minicomputers were included in the category of smaller dollar value procurements (less than $250,000) which made up 56 percent of the 1975 procurement delegations. It was felt that these procurements cost GSA and the user agencies a disproportionate amount of time and resources as they were required to follow the same procedures used for major procurements.[49] The following recommendation was made:[50]

> Federal user agencies should be authorized to procure ADP resources, excluding CPUs, below $250,000 without specific delegation of authority from GSA as long as the agencies document that the procurements are fully competitive. Procurement of CPUs should always require a delegation from GSA.

At present these are just recommendations. Procurement regulations are constantly changing and evolving. The typical procurement of a minicomputer will still involve an alphabet soup of forms and regulations: FPMR 101–32, Public Law 89–306, OMB Circular A 76, F and D, DPA, S.S., M. and M., RFP, IFB, RFQ, nonmandatory schedules, mandatory FSS schedules, mandatory requirements contracts, and so on.

The system suggested in this book will lead the design team to the type of specifications required (functional specifications, technical specifications, or detailed model and make specifications). It should provide the rationale needed for approval requests and justifications and should lead to an efficient, effective system.

---

[46] Harris G. Reiche, "New Directions for ADP Procurement" [Speech] [illegible] *ADP Procurement* [illegible] proceedings of the American Institute of Industrial Engineers Conference held in Washington, D.C., [illegible] November, [illegible] (Santa Monica, Calif.: Management Education Corporation, 1976).

[47] Ibid.

[48] U.S., Congress, House, Committee on Government Operations, *Administration of Public Law 89-306, Procurement of ADP Resources by the Federal Government*, H.R. 1746, 94th Cong., 2d sess. (1976) (Washington, D.C.: U.S. Government Printing Office, 1976), p. [illegible]

[49] Ibid., p. 12.

[50] Ibid., p. 16.

CHAPTER SIX

CONCLUSIONS

This book did not attempt to investigate the potential of the minicomputer in a Federal library. No proof was needed. There was no doubt of the ability of minicomputers to solve many problems and fill many needs of Federal and other libraries. The goal was to demonstrate to the reader this ability and to give guidelines for introducing a minicomputer into a specific library.

What advantages do minicomputers bring to a system? A summary of their attributes gives a good overview.

1. They allow *local control.*
2. The *mini attitude* is a positive, encouraging approach.
3. The mini can be in a *smaller, simpler system.*
4. *Modular system development* is possible.
5. *Custom configurations* are easily accomplished.
6. A mini system is *flexible* and *adaptable to change.*
7. Minis are *less costly.*

Besides these attributes, minicomputers lend themselves to online interactive (as opposed to batch) operations, which provide fuller, more responsive services to the user. Finally, minicomputers "fit in" with any automation environment; they can stand alone, they can perform as parts of networks, they can support networks, they can be connected to hosts as other "terminals," or they can serve in several of these capacities at once.

INITIATING MINICOMPUTER PROJECTS IN FEDERAL LIBRARIES

Okay. The book succeeded you're sold. Now, how do you initiate a minicomputer project in your library?

THE LIBRARY'S ROLE

The library must take the first step. A problem or a need must be identified and the systems analysis/system design phase begun. The library may turn to its agency data processing unit for help at that point, or the help may come from an outside consultant. The system requirements are drawn up. The design model is used. The systems design document is prepared to describe alternative means of meeting system objectives and requirements. The alternatives are reviewed, and a decision is made.

Let us assume that the alternative chosen involves use of a minicomputer. The system specifications are prepared, and the library works closely with the agency procurement unit. RFPs are issued, proposals are reviewed and evaluated, and a selection is made. The contracts are negotiated and signed. The implementation phase follows.

Throughout this period the library administrators have kept their staff informed, sought their comments during the design phase, and performed initial training where needed. This function is critical to the ultimate success of the new system.

THE AGENCY'S ROLE

The parent agency can play an active or passive role. If staff are available, the agency data processing unit can take the main roles in the systems design team. If they do not have the resources to commit to a project of this scope, they should support the efforts of the library to hire an outside contractor or consultant. The data processing unit must contribute to the data gathering phase and give accurate assessments of what, if any, support (in people or equipment) the library can receive both at present and in the future.

The agency procurement unit will play an active role at various stages of the project. If a consultant is hired for the systems analyses, the procurement unit will handle the necessary procedures. After the decision has been made as to systems design, the procurement unit will oversee preparation of the RFP and will issue it. It will gather the proposals submitted and oversee the review, evaluation, and selection process, working closely with the systems design team. Throughout, this unit will perform necessary coordination with the Federal agencies involved (i.e., GSA, OMB, etc.).

FEDERAL GOVERNMENT ROLE

The influence of the Federal Government is strong in the areas of budget, management, and procurement of ADP equipment. The attitudes of the Office of Management and Budget, the General Services Administration, and the National Bureau of Standards were discussed in Chapter Five. There are Federal regulations to follow, forms to complete, and procedures to carry out. The impact of these agencies in a library's purchase of a minicomputer is considerable. As stated above, the link to these agencies is the parent agency's procurement unit.

At the Governmentwide level, the Federal libraries as a group play a role. The Federal Library Committee has recognized this role and has supported model minicomputer projects in individual libraries. It has formed a working group on minicomputers and has sponsored this book. The main needs to be met are establishing communications and ultimately achieving close coordination among libraries. It is important that the knowledge and experience gained by one library be shared with others.

ROLE OF THE PRIVATE SECTOR

The role of private industry in the general area of minicomputers was discussed in Chapter Five. Vendors of library services and products have mounted little in the way of successful, ongoing marketing efforts. The market is there, but many vendors have not bothered to go through the paperwork required to get on a GSA schedule or don't have the patience or marketing budget to invest in the long and complicated bid and negotiation procedures involved in Government procurement. At the other end many libraries have scorned commercial firms as being suspect because of their profit motives. Both vendor and client have much to gain from mutually fair and open relationships.

TRENDS IN FUTURE APPLICATIONS OF MINICOMPUTERS IN FEDERAL LIBRARIES

Almost every book ends with a statement on the future. The most important statement that can be made here is that there is a future for minicomputer applications in Federal libraries. This in itself is the main trend of the present and near future.

We predict that the future will see a trend in Federal libraries toward a distributed network of minicomputers. This network would not be so much a communications network as a network of decentralized data bases and decentralized processing.[1] Bowers envisions a

> ...concept of distributed processing, which holds that data processing is not most efficiently and effectively done in large, centralized computers, but rather should be divided among smaller, loosely-coupled machines. The division might be by geography or by type or task or both, and the coupling might consist of communications lines or a person carrying a reel of tape from one machine to another.[2]

Pezzanite described a similar approach at a workshop on computerized library networks given in Maryland.[3] He proposed a statewide distributed network. The hub of the network was a central processing unit that had a large-scale computer and maintained a large cataloging data base. The data base was to include a generalized data base using MARC tapes, as well as a union state data base. The unit would have "the responsibility of updating and disseminating the distributed data base [contributed by the member libraries] on a regular, cyclic schedule" as well as "the development and distribution of products required by members," such as card sets, book or COM catalogs, and photocomposition. Each member library would have a minicomputer to use for data capture, local processing, and maintenance of its own file. The original cataloging and the holdings would be sent to the central unit to update the master data base. Each library would be responsible for its own file and that file's integrity. Each member would have what Pezzanite described as "functional autonomy."

The viability and feasibility of such a distributed network among Federal libraries are certain. The benefits in efficiency and efficacy would be many. This concept should be uppermost in the minds of Federal library administrators as planning is done over the next few years.

---

[1] For a profile of distributed processing, see the feature presentation in the March 1977 issue of *Mini-Micro Systems*: Stephen A. Kallis, "Networks and Distributed Processing," pp. 32-40, and William G. Moore, Jr., "Going Distributed," pp. 41-48. *Mini-Micro Systems* 10 (March 1977).

[2] Dan M. Bowers, ed., "Small Scale Computing: It's Like Doing Your Laundry, Part I: Computation and Processing," *Modern Data* 8 (May 1975): 46.

[3] Frank A. Pezzanite, "Distributed Library Networking: A New Approach for Maryland," speech given at the Workshop on Computerized Library Networks co-sponsored by the Maryland State Department of Education and the National Library of Medicine, 8-9 April 1976.

IN CLOSING

Walter Curley summarized "what it takes to venture into a library computer program and to make it work."[4] Although his words are not very uplifting or positive, they are realistic and ring true. It is hoped that they will put any new minicomputer plans prompted by this book into perspective, and that they will temper raw enthusiasm with a cautious note of realism.

1. Recognize that hardware is the easiest thing to come by and software the most difficult. The equipment must work; must do what it is intended to do.
2. Know that time is your enemy until your computer program is up and operational.
3. Have the patience of a saint with your staff, your board, your public, your computer experts. Be supportive of staff and understanding about the adjustments which they must make to a new way of doing things.
4. Develop the guts of a cat burglar — you will need them. Once you have made a decision to be innovative with computers, everyone will be from Missouri until you show the promised and hoped-for results.

---

[4] Walter Curley, "Innovative Strategies in Systems and Automation," in *Library Automation: The State of the Art II*, eds. Susan K. Martin and Brett Butler, papers presented at the Preconference Institute on Library Automation sponsored by the Information Science and Automation Division of the American Library Association at Las Vegas, Nevada, 22-23 June 1973 (Chicago: American Library Association, 1975), p. 134.

APPENDIX A

GLOSSARY

ACCUMULATOR

A hardware register that holds the results of arithmetic, logical, and I/O operations.

ACOUSTICAL COUPLER

A device that converts electrical signals into audio signals. It is used with a telephone handset for connecting to the public telephone network for data transmission.

ADDRESS

An identification (name, label, or number) for a register, location in storage, or other source or destination for data.

AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE (ASCII)

A standard code that represents characters by seven-bit-plus-parity codes; for use with various data processing systems, communication systems, and associated equipment.

ANALOG COMPUTER

A computer that operates on analog data by performing physical processes on these data. *Compare*: digital computer

ANALOG DATA

Data in continuous form, usually numerical quantities of physical variables such as voltage, speed, rotation, resistance, etc. *Compare*: digital data

ARITHMETIC UNIT

The part of the central processing unit that carries out computational and logical operations.

ASCII *see* AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE

ASSEMBLER

A computer program that prepares a machine language program by converting symbolic language codes into absolute operation codes and assigning absolute or relocatable addresses for symbolic addresses.

ASYNCHRONOUS

Occurring without regular or predictable time relationship.

ASYNCHRONOUS TRANSMISSION

Data transmission in which control is achieved by "start" and "stop" elements at the beginning and end of each character.

AUTOMATIC COMPUTER

A machine that manipulates symbols according to given rules in a predetermined and self-directed manner.

AUXILIARY MEMORY

(1) Data storage other than main storage on such a device as a magnetic tape unit or a direct access unit.

(2) A storage that supplements another storage. *See also*: mass storage

BASIC

A high-level, algebralike language designed for use in problemsolving by engineers, scientists, and others who are not professional programmers. It is available for interactive, time-sharing direct communications between terminal and host.

BATCH

(1) A type of input in which a number of similar input items or transactions are accumulated and processed together at one time.

(2) The sequential input of computer programs or data.

(3) The technique of executing a set of computer programs such that each is completed before the next program of the set is started, i.e., execution of computer programs serially.

BAUD

An abbreviation for "bit audible." A unit of signaling speed equal to the number of discrete conditions or signal events per second. Note: when the discrete condition is such that one element carries one bit, the baud rate is numerically equal to bits per second (BPS).

BCD *see* BINARY-CODED DECIMAL

BINARY-CODED DECIMAL (BCD)

A type of notation system where each decimal digit is represented by four binary digits (or bits).

BINARY DIGIT (BIT)

(1) One of the two numerals in the binary number system — zero or one. It may be equivalent to any characteristic, property, or condition in which there are only two possible conditions, such as on or off or yes or no.

(2) The kind of number used internally by computers. *See also*: machine language.

BINARY NUMBER SYSTEM

A number representation system using base-two notation, in which the only valid digits are zero and one.

BINARY SYNCHRONOUS TRANSMISSION

Data transmission in which character synchronism is controlled by timing signals generated at the sending and receiving stations. *Also called*: bisynchronous

BISYNCHRONOUS *see* BINARY SYNCHRONOUS TRANSMISSION

BIT *see* BINARY DIGIT

BITS PER INCH (BPI)

A measure of the density in which the number of bits of information are contained or written along an inch of magnetic tape.

BPI *see* **BITS PER INCH**

BUFFER

An area of internal storage or a hardware device used to store information temporarily during data transfers. It is used to compensate for a difference in rate of flow of data, or time of occurrence of events, when transmitting data from one device to another.

BUG

(1) A program defect or error.

(2) A mistake or malfunction.

BUS

A circuit or path over which data or power is transmitted; usually lines that connect locations or a single line that acts as a common connection among a number of locations.

CARTRIDGE TAPE

A type of magnetic tape in a special housing used for mass storage. It is similar to a tape cassette.

CASSETTE TAPE OPERATING SYSTEM

An operating system designed to use a cassette tape as the mass storage device.

CATHODE-RAY TUBE (CRT)

A vacuum tube similar to a television picture tube, used as a storage or a visual display device.

CENTRAL PROCESSING UNIT (CPU)

The unit of a computer that controls the interpretation and execution of instructions such as calculations and logic decisions. It is composed of the arithmetic unit and the control unit and functions directly with main memory. *Also called*: main frame

CENTRALIZED (COMPUTER) NETWORK

A computer network configuration in which a central node provides computing power, control, or other services. *Also called*: star network

CHANNEL

(1) A path along which signals can be sent.

(2) That part of a communications system that connects a message source with a message sink (that is, a terminal installation that receives and processes data). *See also*: information (transfer) channel

(3) A means of one-way transmission. *Compare*: circuit

CHIP *see* **LSI CHIP**

CIRCUIT

In communications, the complete electrical path providing one- or two-way communication between two points comprising associated go and return channels. *Compare*: channel

CLOCK

A device that generates periodic signals used for synchronization.

COBOL

An abbreviation for *C*ommon *B*usiness-*O*riented *L*anguage. A high-level language designed for use in business data processing applications.

CODE

(1) A set of unambiguous rules specifying the way in which data may be represented.

(2) In data communications, a system of rules and conventions according to which the signals representing data can be formed, transmitted, received, and processed.

COM *see* **COMPUTER-OUTPUT MICROFORM**

COMMUNICATIONS LINK

The means of connecting one location to another for the purpose of transmitting and receiving information.

COMPILER

A computer program that prepares a machine language program (object program) from a computer program written in another programming language (source language, usually a high-level language) by using the overall logic structure of the program or by generating more than one machine instruction for each symbolic state, or both, as well as performing the function of an assembler. A compiler usually contains its own library of closed routines.

COMPUTER NETWORK

An interconnection of assemblies of computer systems, terminals, and communications facilities. A complex consisting of two or more interconnected computing units.

COMPUTER-OUTPUT MICROFORM (COM)

A process in which computer information is output onto a microform (e.g., microfilm or microfiche) through a COM printer.

CONCENTRATOR *see* **DATA CONCENTRATOR**

CONFIGURATION

The group of devices that make up a computer or data processing system.

CONSOLE

The unit of a computer containing the control keys and certain special devices used by the operator for direct communication with the computer. It can be used to control the machine manually, correct errors, determine the status of the machine circuits, registers, and counters, determine the contents of storage, and revise manually the contents of storage.

CONTENTION SYSTEM

A system in which one or more terminals and the computer compete for use of the line; involves unregulated bidding for a line by multiple users. *Compare*: polling system

CONTROL UNIT

(1) The part of the central processing unit that directs the sequence of operations, interprets coded instructions, and sends the proper signals instructing other computer circuits to carry out the instructions.

(2) A device that controls the reading, writing, or display of data at one or more input/output devices. *See also*: input/output controller

CONVERSATIONAL MODE

The processes for communication between a terminal and the computer, in which each entry from the terminal requires a response from the computer and vice versa. This mode involves step-by-step interaction between the user and a computer.

CONVERSION

(1) The process of changing from one form of representation to another.

(2) The process of transferring information from one recorded medium to another.

CORE

A configuration of magnetic material used with current-carrying conductors to retain a magnetic polarization in either an "on" or an "off" state, for the purpose of storing data represented as a binary one (on) or zero (off). It is commonly used as main memory for computers; main memory is often called "core memory." *Also called*: magnetic core

COUNTER

A device, such as a register or storage location, used to represent the number of occurrences of an event.

CPU *see* CENTRAL PROCESSING UNIT

CRT *see* CATHODE-RAY TUBE

CTOS *see* CASSETTE TAPE OPERATING SYSTEM

CURSOR

A movable spot of light on a cathode-ray tube of a console or a display unit that indicates where the next character will be entered.

CYCLE

An interval of space or time in which one set of events or phenomena is completed.

CYCLE TIME

(1) The time to read (and restore) a single word in memory.

(2) The minimum time interval, in microseconds, between two successive accesses to a particular storage location.

DATA BASE

(1) The entire collection of information available to a computer system.

(2) A structured collection of information as an entity or collection of related files treated as an entity.

DATA CONCENTRATOR

A communications device that provides communications capability between many low-speed, usually asynchronous channels and one or more high-speed, usually synchronous channels. Usually different speeds, codes, and protocols can be accommodated on the low-speed side. The low-speed channels usually operate in contention, requiring buffering. The concentrator may have the ability to be polled by a computer, and may in turn poll terminals.

DATA ELEMENT

A class or category of data based on natural or assigned relationships.

DATA HIERARCHY

A data structure consisting of sets and subsets such that every subset of a set is of lower rank than the data of the set.

DATA STRUCTURE

The manner in which data are represented and stored in a computer system or program.

DBMS

An abbreviation for *D*ata *B*ase *M*anagement *S*ystem.

DEBUG

To detect, locate, and remove mistakes from a routine or malfunctions from a computer.

DEBUGGER

A systems software program designed to help the programmer determine causes of problems found during the running of his software. It features the ability to stop the executing program and determine the state of the machine at that time, i.e., the content of all memory locations and registers.

DEMODULATOR

A device that receives signals transmitted over a communications link and converts them into electrical pulses, or bits, that can serve as inputs to a data processing machine. *Compare*: modulator

DIAGNOSTICS

A program that facilitates computer maintenance by detecting and isolating malfunctions or mistakes.

DIGITAL COMPUTER

A computer that operates on digital data by performing arithmetic and logical processes on these data. *Compare*: analog computer

DIGITAL DATA

Information represented by a code consisting of a sequence of discrete elements. *Compare*: analog data

DIRECT ACCESS DEVICE

(1) A memory device that allows a particular data address to be accessed independently of the location of that address. The reference is to a location of a volume rather than relative to the previously retrieved or stored data. *Compare*: serial access

(2) A storage device in which the access time is effectively independent of the location of the data.

Also called: random access device

DIRECT MEMORY ACCESS (DMA)

A method of data transfer using a hardware device that sets up a high-speed data path to link memory with peripheral electronics.

DISK OPERATING SYSTEM (DOS)

An operating system designed to use a disk as a mass storage device.

DISK PACK

A removable direct access storage device containing magnetic disks on which data are stored.

DISKETTE *see* **FLOPPY DISK**

DISPLAY

(1) A presentation of data in human-sensible form.

(2) A device for visually presenting data from the computer to a user.

DISTRIBUTED NETWORK

A network in which all node pairs are connected, either directly or through redundant paths through intermediate nodes. *Compare*: fully distributed network. *Also called*: multistar network

DMA *see* **DIRECT MEMORY ACCESS**

DOS *see* **DISK OPERATING SYSTEM**

DUPLEX TRANSMISSION

Simultaneous two-way independent transmission in both directions. *Also called*: full-duplex *Compare*: half-duplex

E-TIME

The execution cycle. One of two basic machine cycles of the control unit. The instruction is performed during the execution cycle. E-time varies according to the length of time required to perform a specific instruction; for example, addition does not take as long as multiplication. *Compare*: I-time

EBCDIC *see* **EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE**

EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE

A system of codes for a set of 256 characters, each represented by a different eight-bit pattern.

FIELD

(1) A set of characters representing logical units or data elements.

(2) In a record, a specified area used for a particular category of data. It can be of fixed or variable length.

FILE

A collection of related records treated as a unit.

FILE LAYOUT

The arrangement and structure of data in a file, including the sequence and size of its components.

FILE MAINTENANCE

Keeping a file up to date by adding, changing, or deleting data.

FIRMWARE

Software instructions that have been more or less permanently burned into a ROM (read-only memory) or PROM (programmed read-only memory) chip.

FIXED-HEAD DISK

A mass storage device which uses a rigid, circular plate with a nonmovable read-write head for each track of the disk.

FIXED-LENGTH FIELD

A data field that has the same predetermined number of characters from record to record. *Compare*: variable-length field

FLAG

A character or other indicator that signals the occurrence of some condition.

FLIP-FLOP

A sequential switching circuit capable of storing one bit of information in one of two stable states. Flip-flops may be grouped to form storage registers, counters, shift registers, or other functional components.

FLOPPY DISK

A mass storage device in which the storage medium is a single flexible plate of Mylar material housed in a paper jacket. *Also called*: diskette

FORMAT

A specific arrangement of data.

FORTRAN

An abbreviation for *For*mula *Tran*slator. A high-level language designed for mathematical computations.

FULL DUPLEX *see* **DUPLEX**

FULLY DISTRIBUTED NETWORK

A network in which each node is directly connected with every other node. *Compare*: distributed network

GATE

A device having one output channel and one or more input channels, such that the output channel state is completely determined by the input channel states except during switching transients. Common logic gates are AND, NO, and NAND (not and).

HALF-DUPLEX

Alternate, one way at a time, independent transmission. *Compare*: duplex

HARDWARE

The physical equipment of a data processing system, as opposed to the computer program or method of use.

HEXADECIMAL NOTATION

A scheme in which hexadecimal numbers are used to represent four-bit patterns as a shorthand means. The hexadecimal number system uses a base of sixteen, with valid digits that range from 0 through F, where F represents the highest units position (15).

HIERARCHICAL (COMPUTER) NETWORK

A computer network in which processing and control functions are performed at several levels by computers specially suited for the functions performed, e.g., in factory or laboratory automation. *Also called*: tree network.

HOST

A computer attached to a network providing mainly services such as computation, data base access, or special programs or programming languages.

I-TIME

The instruction cycle. One of two basic machine cycles of the control unit. During the instruction cycle, the instruction register receives the next instruction in the stored program, the instruction is interpreted in preparation for its execution, and the location counter is adjusted to contain the address to the next instruction. *Compare*: E-time

IDENTIFIER

A symbol whose purpose is to identify, indicate, or name a body of data.

ILL

An abbreviation for *I*nterlibrary *L*oan.

INDEX

(1) In data management, a table in the catalog structure used to locate data sets.

(2) In data management, a table used to locate the records of indexed sequential data sets.

INFORMATION (TRANSFER) CHANNEL

(1) The functional connection between the source and the sink data terminal equipment. It includes the circuit and the associated data communications equipment.

(2) The assembly of data communications and circuits, sometimes including a reverse channel.

INPUT/OUTPUT CONTROLLER

A device that directs the interaction between the processing unit and input/output devices. *See also*: control unit

INQUIRY

A request for information from storage.

INSTRUCTION SET

A set of operations that can be represented in a given operation code. *Also called*: instruction repertoire

INTERACTIVE

Pertaining to exchange of information and control between a user and a computer process. In the interactive mode, direct communication is established and a conversation or dialogue is often undertaken. *See also*: conversational mode

INTERFACE

A shared boundary. An interface might be a hardware component linking two devices, or a portion of storage or registers accessed by two or more computer programs. An interface enables devices to transfer information to and from one device or program and another.

INTERNAL STORAGE

The addressable storage directly controlled by the central processing unit. *Also called*: main memory

INTERPRETER

(1) A program that translates and executes each source language statement before translating and executing the next one.

(2) A device that prints on a punched card the data already punched in the card.

INTERRECORD GAP (IRG)

The area at the end of a block or record on a blank tape or disk. It is created during the start-up and the stop time, during which no data are recorded.

INTERRUPT

To stop a process in such a way that it can be resumed.

INTERRUPTION

A break in the normal sequence of instruction execution which causes an automatic transfer to a preset storage location where appropriate action is taken.

INVERTED FILE

In information retrieval, a method of organizing a cross-index file in which keywords identify records.

INVERTED TABLE LOOK-UP

The process of searching a table whose sequence or order is other than that of the original table.

I/O

An abbreviation for *I*nput/*O*utput.

IRG *see* INTERRECORD GAP

JOB

A specified group of tasks prescribed as a unit of work for a computer. A job usually includes all necessary programs, linkages, files, and instructions to the system.

JOB-ORIENTED TERMINAL
A terminal designed to receive source data in an environment associated with the job to be performed and capable of transmission to and from the system of which it is a part.

KEY
One or more characters in an item of data that are used to identify it or control its use.

KEYWORD
One of the significant and informative words in a title or document, describing the content of that document.

LINKED LIST RELATIONSHIP
A configuration of ordered items, which usually do not occupy contiguous locations in online or offline memory.

LOGICAL ENTRY
All the data input to the data base in one logical grouping.

LOGICAL FILE
A collection of one or more logical records.

LOGICAL RECORD
A collection of items independent of their physical environment. Portions of the same logical record may be located in different physical records.

LOOP NETWORK
A computer network in which each computer is connected to adjacent computers. *Also called*: ring network

LSI CHIP
A small integrated-circuit package containing many logic elements. Large-scale integration (LSI) densities can vary from 500 to 10,000 transistors per chip.

MACHINE LANGUAGE
A binary language or code used directly by a computer.

MACRO
An instruction in a source language, equivalent to a specified sequence of machine instructions.

MAGNETIC CORE *see* CORE

MAIN FRAME *see* CENTRAL PROCESSING UNIT

MAIN MEMORY
(1) The general-purpose storage of the central processing unit that can be accessed directly by the operating registers. It stores both the data on which to be operated and the program dictating the operations to be performed.
(2) All program-addressable storage from which instructions may be executed and from which data can be loaded directly into registers.
Also called: main storage and working storage

MARC
An abbreviation for *Ma*chine-*R*eadable *C*ataloging.

MASS STORAGE
Data storage other than the main memory, usually devices that have large capacities, such as magnetic tape or disks. *Also called*: auxiliary storage

MEAN TIME TO FAILURE (MTTF)
A measurement based on a ratio of the operating time of equipment to the number of observed failures. *Also called*: mean time between failures (MTBF)

MEAN TIME TO REPAIR (MTTR)
A measurement that relates to the normal repair time for a piece of equipment.

MEMORY
A unit of the computer used to store information received through an input unit or developed during the processing of data. The information can be brought out of storage for use without being destroyed. *Also called*: storage or store

MESSAGE SWITCHING
A method of handling messages over communications networks. The entire message is transmitted to an intermediate point (a switching computer), stored for a period of time, and then transmitted towards its destination. The destination of each message is indicated by an address integral to the message.

MICROCOMPUTER
A computer with a microprocessor as its central processing unit.

MICROPROCESSOR
An LSI central processing unit on one or a few chips.

MINICOMPUTER
A physically small, relatively inexpensive, general-purpose computer that can operate in a regular environment with as much peripheral and system support as necessary to meet the requirements of the application.

MODEM
An abbreviation for *Mo*dulator-*Dem*odulator. A device that modulates and demodulates signals transmitted over communication facilities.

MODULATION
The process by which some characteristic of one wave is varied in accordance with another wave or signal. This technique is used in data sets and modems to make business machine signals compatible with communications facilities. It converts digital data for transmission on analog telephone lines and for recovering the digital information at the receiver. *Compare*: demodulator

MOS
An abbreviation for *M*etal *O*xide *S*emiconductor, a kind of material used in constructing chips.

MOVABLE-HEAD DISK

A mass storage device that uses as a medium a rigid, circular plate with a movable read-write head that positions itself over the appropriate track of the disk.

MTBF *see* MEAN TIME TO FAILURE

MTTF *see* MEAN TIME TO FAILURE

MTTR *see* MEAN TIME TO REPAIR

MULTIPLEX MODE

A means of transferring records to or from low-speed input/output devices on the multiplexer channel, by interleaving bytes of data. The multiplexer channel sustains simultaneous input/output operations on several subchannels.

MULTIPLEXER CHANNEL

A channel designed to operate with a number of input/output devices simultaneously.

MULTIPOINT NETWORK

A configuration in which more than two terminal installations are connected.

MULTIPROCESSING

A configuration of two or more central processors that can be independently initiated and have access to a common, jointly-addressable memory. Each processor can operate simultaneously, either on segments of the same job or on entirely different jobs.

MULTIPROGRAMMING

A resource management system in which an executive routine allocates the resources of the computer to many programs concurrently. It is composed of procedures for handling numerous routines or programs seemingly simultaneously by overlapping or interleaving their execution.

MULTISTAR NETWORK *see* DISTRIBUTED NETWORK

MULTITASKING

(1) A program design strategy in which the various logical elements making up a program are written so that they may operate asynchronously with respect to one another.

(2) Procedures in which several separate but interrelated tasks operate under a single program identity and may use common routines, data space, and disk files.

NETWORK

(1) An interconnected or interrelated group of nodes.

(2) In teleprocessing, a number of communication lines connecting a computer with remote terminals.

NODE

(1) An end point of any branch of a network, or a junction common to two or more branches of a network.

(2) Any station, terminal, terminal installation, communications computer, or communications computer installation in a computer network.

OCR *see* OPTICAL CHARACTER RECOGNITION

OFFLINE

Pertaining to equipment or devices not under control of the central processing unit. *Compare*: online

OFFLINE SYSTEM

In teleprocessing, that kind of system in which human operations are required between the original recording functions and the ultimate data processing function. This includes conversion operation as well as the necessary loading and unloading operations incident to the use of point-to-point or data-gathering systems.

ON-DEMAND SYSTEM

A system from which information or service is available at time of request.

ONLINE

Pertaining to equipment or devices under control of the central processing unit. *Compare*: offline

ONLINE DATA PROCESSING

Data processing in which all changes to relevant records and accounts are made at the time that each transaction or event occurs. *Compare*: batch processing

ONLINE SYSTEM

(1) In teleprocessing, a system in which the input data enters the computer directly from the point of origin or in which output data are transmitted directly to where it is used.

(2) A system that eliminates the need for human intervention between source recording and the ultimate processing by a computer.

OPERATING SYSTEM

Software that controls the execution of computer programs and may provide scheduling, debugging, input/output control, accounting, compilation, storage assignment, data management, and related services.

OPTICAL CHARACTER RECOGNITION (OCR)

Machine identification of printed characters through use of light-sensitive devices.

PAGE

A segment of a program or data, usually of fixed length, that has a fixed virtual address but can in fact reside in any region of the computer's working storage.

PARITY

A simple form of error detection in which one nondata bit is added to the data bits in a character so that the total number of "one" bits is either always even or always odd.

PARITY BIT

A check bit appended to an array of binary digits to make the sum of all the binary digits, including the check bit, always odd or always even.

PARITY CHECK
A check that tests whether the number of ones (or zeros) in an array of binary digits is odd or even.

PERIPHERAL
In a data processing system, any unit of equipment, distinct from the central processing unit, which may provide the system with outside communication.

PHYSICAL RECORD
A group of words, characters, or digits held in one section of an input/output medium or store and handled as a unit.

PIO *see* PROGRAMMED INPUT/OUTPUT

POINT-TO-POINT NETWORK
A network configuration in which a connection is established between two, and only two, terminal installations.

POINTER
An identification, as represented by a name, label, or number, for a register, location in storage, or any other data source or destination. Loosely, the address of the next record.

POLLING SYSTEM
A system in which each of the terminals sharing a communications line is periodically interrogated to determine whether it requires servicing.

PRIMARY STORAGE *see* MAIN MEMORY

PROCESS CONTROL
Pertaining to systems whose purpose is to automate continuous operations.

PROCESSOR *see* CENTRAL PROCESSING UNIT

PROGRAM
(1) A series of actions proposed in order to achieve a certain result.
(2) A plan and operating instructions needed to produce results from a computer.

PROGRAMMED INPUT/OUTPUT (PIO)
A method of data transfer that makes use of program instructions rather than hardware devices to control the transfer of information between the central processor and an external device.

PROM
An abbreviation of *P*rogrammable *R*ead *O*nly *M*emory. A kind of chip that is not recorded during its manufacture, but instead requires a physical operation to program it. Some PROMs can be erased and reprogrammed through special physical processes. *Compare*: ROM

PROTOCOL
A formal set of conventions governing the format and relative timing of message exchange between two communicating processes.

RANDOM ACCESS DEVICE *see* DIRECT ACCESS DEVICE

READ-ONLY ACCESS
The sharer may read the data set but he may not change it in any way.

REAL TIME PROCESSING
The processing of information or data rapidly enough that the results of processing are available in time to influence the process being monitored or controlled.

RECORD
(1) A collection of related items of data, treated as a unit.
(2) A collection of fields.

REGISTER
(1) A device capable of storing a specified amount of data, such as one word.
(2) A temporary storage device used for one or more words to facilitate arithmetical, logical, or transferral operations.

REMOTE ACCESS
Pertaining to communication with a data processing facility by one or more stations that are distant from that facility.

REMOTE BATCH PROCESSING
A procedure in which computer programs or data are entered into a remote terminal for transmission to the central processor. This allows various systems to share the resources of a batch-oriented computer.

RFP
An abbreviation for *R*equest *F*or *P*roposal.

RING NETWORK *see* LOOP NETWORK

ROM
An abbreviation of *R*ead *O*nly *M*emory. A kind of chip that has all of its circuits, e.g., logic elements or data, recorded as it is manufactured and can never be erased.

SDI
An abbreviation for *S*elective *D*issemination of *I*nformation.

SEMICONDUCTOR MEMORY
A memory whose storage medium is a semiconductor circuit.

SERIAL ACCESS
(1) Pertaining to the sequential or consecutive transmission of data to or from storage.
(2) Pertaining to the process of obtaining data from, or placing data into, storage, where the time required for such access depends on the location of the data most recently obtained or placed in storage. *Compare*: direct access device

SHARED FILE

A direct access device that may be used by two systems at the same time; a shared file may link two systems.

SINK

(1) The point of use of data in a network.

(2) A data terminal installation that receives and processes data from a connected channel.

SOFTWARE

A set of programs, procedures, and possible associated documentation concerned with the operation of a data processing system.

STACK

(1) A block of successive memory locations that are accessible from one end on a last-in-first-out basis. The stack is coordinated with a stack pointer that keeps track of storage and retrieval of each byte or word of information in the stack. The words "push" (move down) and "pop" (retrieve the most recently stored item) are used to describe its operation.

(2) A hardware device composed of a collection of registers with a counter which serves as a pointer to indicate the most recently loaded register. Registers are unloaded in the reverse of the sequence in which they were loaded.

STAR NETWORK

A computer network with peripheral nodes all connected to one or more computers at a centrally located facility. *See also:* centralized network

STORAGE MEDIUM

The material on which data are stored, e.g., magnetic and paper tapes, disks, and magnetic core.

SUPERVISOR

A control routine or routines through which the use of resources is coordinated and the flow of operations through the central processing unit is maintained.

SYSTEMS ANALYSIS

The study of all of the components, operations, data, information and material flow, work environment, etc., that constitute the existing system.

TABLE

A collection of data in which each item is uniquely identified by a label, by its position relative to the other items, or by some other means.

TAPE OPERATING SYSTEM

An operating system designed to use magnetic tape as the mass storage device.

TASK

A unit of work for the central processing unit, from the standpoint of the control program.

TASK MANAGEMENT

Those functions of the control program that regulate the use by tasks of the central processing unit and other resources.

TELECOMMUNICATIONS

(1) The transmission of signals over long distances, such as by telegraphy, radio, or television.

(2) Data transmission between a computing system and remotely located devices via a unit that performs the necessary format conversion and controls the rate of transmission.

TELEPROCESSING

(1) A form of information handling in which a data processing system uses communication facilities.

(2) The processing of data that is received from or sent to remote locations by way of telecommunications lines.

TERMINAL

(1) A point in a system or communications network at which data can either enter or leave.

(2) Any device capable of sending and receiving information over a communication channel.

TIME-SHARING

(1) A method of using a computing system that allows a number of users to execute programs concurrently and to interact with the programs during execution. Although the computer actually services each user in sequence, the high speed of the computer makes it appear that the users are all handled simultaneously.

(2) Pertaining to the interleaved use of the time of a device.

TIME-SLICE

A designed interval of time during which a job can use a resource without being preempted.

TIME-SLICING

A feature that can be used to prevent a task from monopolizing the central processing unit and thereby delaying the assignment of CPU time to other tasks.

TOS *see* **TAPE OPERATING SYSTEM**

TREE NETWORK *see* **HIERARCHICAL NETWORK**

TTY

An abbreviation for *Teletypewriter* equipment.

TURNKEY SYSTEM

A commercial on-the-shelf package that includes the minicomputer, peripherals, and software necessary for a specific application, complete and ready to use upon installation.

VARIABLE-LENGTH FIELD

A data field that can vary numbers of characters from record to record. *Compare:* fixed-length field

VIRTUAL MEMORY

A method of storage access involving transfer of information one page or more at a time between primary and secondary memory, and allowing the programmer to address total storage without regard to whether primary or secondary storage is actually being addressed.

VIRTUAL OPERATING SYSTEM (VOS)

An operating system designed to use a virtual memory mass storage technique.

VOS *see* VIRTUAL OPERATING SYSTEM

WORD

(1) A character string or a bit string considered as an entity.

(2) A group of characters occupying one storage location in a computer. It is treated by the computer circuits as an entity, by the control unit as an instruction, and by the arithmetic unit as a quantity.

(3) The smallest addressable unit in main memory.

WORKING STORAGE

In programming, storage locations reserved for intermediate results. *Also called*: main memory

ZEBRA LABEL

A label with preprinted lines in coded patterns which are read by a light pen (a photocell device). *Also called*: bar-encoded label

APPENDIX B

LIBRARY OF CONGRESS SYSTEMS SPECIFICATIONS FOR PROCUREMENT OF MINICOMPUTER SYSTEMS

ERIC
Full Text Provided by ERIC

DESIRED/OPTIONAL WEIGHTING

	SECTION E REFERENCE		
A. COMMUNICATIONS			5%
1. Switched network operation	II 2.b.1.c	6	
2. Remote batch concurrent with interactive applications	II 2.b.1.d	6	
3. Higher level synchronous protocols	II 2.c.1.d	7	
4. Loop testing	II 2.c.1.e	7	
5. Software selectable baud rate	II 2.c.2.c	7	
B. PERIPHERALS			5%
1. CRT			
7×9 dot matrix character generation	II 2.d.1.d	9	
character highlight	II 2.d.1.g	9	
alternate upper case only	II 2.d.1.j	10	
serial input port	II 2.d.1.m	10	
58 line × 80 character format	II 2.d.1.o	10	
MARC-II character set	II 2.d.1.k	10	
Hardware data entry format	II 2.d.1.h	9	
Automatic tab	II 2.d.1.f.	9	
2. Separate system disk	II 2.d.2.a.	12	
3. Magnetic tape	II 2.d.3.p	12	
4. Line printer—MARC II character set	II 2.d.4.h	13	
5. Document Printer		14	
alternate upper case only	II 2.d.5.f	14	
word processing capabilities	II 2.d.5.g	14	
MARC II character set	II 2.d.5.h	14	
6. IBM channel interface	II 2.d.6	14	
C. SYSTEM SOFTWARE			15%
1. Generic & specific peripheral reference	II 3.a.2	15	
2. Privileged user access	II 3.a.4	15	
3. Tuning	II 3.a.5	15	
4. User prioritization	II 3.a.10	15	
5. File management—binary data handling	II 3.a.13.a	16	
6. Down line load from host computer	II 3.a.16	17	
7. High level languages—PL/1, ALGOL, BASIC, RPG-II	II 3.b.1	17	
8. Sort/merge performance	II 3.d.1	18	
D. QUERY AND SOURCE DATA ENTRY SOFTWARE	II 3.c	18	15%
E. DBMS	II 3.f	18	10%
F. WORK PROCESSING SUBSYSTEM	II 3.g		10%
G. RELIABILITY			35%
Methodology	II 3.i	27	
Analysis	II 2.a.6	4	
H. SYSTEM HARDWARE EASE OF PROGRAMMING			5%

SECTION F

I. SYSTEM SPECIFICATIONS

This RFP is intended to be a "System Specification" rather than a "Hardware Specification." Overall system requirements and individual configuration requirements have been described so that each prospective vendor will have the capability to propose the most effective approach for his particular product line. It would not be in the best interest of LC to circulate an RFP containing a list of specific model numbers, core sizes, and so on. For a procurement of this kind, it is most useful to rely on the unique problem-solving resources of minicomputer-oriented vendors and OEM's to prescribe a set of hardware and system software which will meet or exceed LC's response and reliability requirements.

A. Global Requirements (common to all systems covered by this procurement).

1. System Architecture

   a. The systems proposed must be multilingual and multiprogramming computer systems capable of supporting interactive program development. There shall be a minimum of two terminals available for development of applications code but not necessarily concurrent with any given application.

   b. To accommodate changes in workload, the systems (hardware and software) *must* allow modular increments in hardware, i.e., able to utilize more or less of any of the individual hardware components (except the cpu) and more or less cpu memory capacity by field modification.

   c. The proposed systems must include an operating system which permits effective control of routine operations as well as unscheduled operations, e.g., recovery.

   d. The systems proposed must communicate with system users, e.g., via operating system commands, a text editor, a language processor and interactive debugger.

   e. The architecture and configuration of each proposed system must enable each system to provide a response time to each user terminal of not more than ten (10) seconds during sustained peak load operation (the functional descriptions of each of the six individual systems provide sufficient information for Bidders to derive "worst-case" and sustained peak load performance requirements for each system).

   f. The system proposed must have software compatibility across all configurations of a single mainframe type or a computer "line," if such configurations are proposed.

2. System Hardware

   a. *Mainframe or CPU*. Each proposal shall include relevant details such as bandwidths, buss structure, multiplicity of data paths, and control paths, instruction repertoire and timings, "intelligent" components, interrupt structure, registers, and performance enhancement strategies.

      (1) The fundamental unit of internal data movement must be a "word" of at least 16 data bits plus any system bits such as parity, etc.

      (2) There must be at least four (4) general purpose registers usable by application programs; sixteen (16) registers are preferred. Describe methodology for using registers as accumulators and for indexing.

      (3) The upper and lower limits of memory must be explicity stated. These limits should be a function of both hardware and system software requirements (i.e.,

for the operating system, compilers etc. specified for this procurement, what is minimum memory required and maximum memory supported).

(4) The basic memory cycle time must be one microsecond or less and the effective memory cycle time due to memory configurations such as interleaving or cache memory paging must be stated in detail with any assumptions used such as hit rate or instruction mixes.

(5) The following features must be included:

- (a) program accessible clock
- (b) automatic program load
- (c) memory protection
- (d) automatic error detection
- (e) automatic power failure detection
- (f) automatic restart from power failure with no loss of memory.

(6) Illustrate using assembler source code statements and in a high-level language such as COBOL, FORTRAN or PL/1 the necessary instructions required to perform the following tasks. For each task, state the time required for execution assuming no interruptions occur:

- (a) Move 1601 bytes from one area of memory to another area of memory.
- (b) Compare a character string of 329 bytes to another string of the same length and branch if not equal.
- (c) Compare one binary value to another binary value and branch if not equal. Both values are initially stored in memory.
- (d) Branch to a location in memory having saved the address of the next instruction (so that a return could be executed).
- (e) Decode and add the following *character strings* together and encode the result back into a character string. Timing should be given for each step in the process (i.e., decode, add, encode.)

  123,932,480.63
  1,420.64
- (f) Specify the time required to access a physical block of data from a disk file considering the operating system overhead and hardware limitations. Assume no file management system.

(7) State whether processor instruction set is hardwired or microcoded.

(8) State whether any of the following desired instruction types are available:

- (a) Immediate data (i.e., data available in instruction word).
- (b) Stack operations (push, pop, etc.).
- (c) Operations on packed, signed, non-integer (i.e., real) data.
- (d) Byte addressing without having to load into register and shift.
- (e) Code translation and string edit.

b. *Communications*

(1) Remote Batch

- (a) The proposed system must be capable of communicating as a remote batch terminal and contain a minimum of two channels.
- (b) As a remote batch terminal, the proposed systems must interface via bisynchronous communication protocol to an IBM 370 Model 158 using 2780 bisync or HASP work station protocol.

(c) The capacity to use a private line service for remote batch communications is mandatory. It is also desirable to have the capacity to use a switched network.

(d) It is desirable that the proposed systems be able to support concurrent operations of remote batch and interactive applications terminals.

(e) Terminal controllers or multiplexors must be capable of accommodating up to at least 60 asynchronous concurrent connections.

(f) Detailed information of the preferred characteristics of the communication channels follows. If a vendor cannot meet all of the preferred specifications he should specify which are not met and why his proposed system will be better or more useful to LC.

c. *Communication Channels*

(1) *Synchronous Channel*

(a) The proposed system shall have a Transmitter/Receiver capable of interfacing with modems whose input characteristics conform to one or more of the following at a transmission rate of at least 9600 bps:

1. EIA RS-232-C, RS-334, RS-269A
2. CCITT Recommendation V. 35
3. WECO 303 (Bell 303) as described in AT&T Publication 41302 "Wideband Data Stations, 303 Type."

When more than one of the above is available the selection of the particular interface shall be done by external cable changes and/or modular replacement of output circuit drivers/receivers at the LC facility.

(b) The synchronous channels shall be capable of handling 2780 bisynchronous protocol or HASP work station protocol. A complete and detailed description of this portion of the system is requested to include hardware interfaces, software supplied or required and any limitations.

(c) The vendor also must supply a cable and signal connection diagram for this section identifying the type connector being proposed and its signals. This will enable LC personnel to adequately determine if the proposed system is suitable.

(d) A desirable option would be the adaptability to several of the new bit oriented, full duplex protocols listed below:

1. FED-STD-1020(A) EIA RS-422 Balanced
2. FED-STD-1030(A) EIA RS-423 Unbalanced
3. FED-STD-1031 (Proposed) EIA XYZ Physical for "Analog" facilities
4. FED-STD-1029 (Proposed) EIA-ABD
5. FED-STD-1040 (Proposed) CCITT X. 21 Public digital networks.

(e) The vendor shall describe any loopback testing facilities available for determining if the computer interface is operating correctly.

(2) *Asynchronous Channels*

(a) The asynchronous channel shall interface with the processor or memory on a character basis. The transmitter shall have a parallel loaded buffer and serial output. An overrun error signal is desirable.

(b) The speed range shall be from a minimum of 110 baud to a maximum of 9600 baud rate.

(c) It is desired that the Transmitter/Receiver be designed to allow software to select the baud rate.

(d) Even, odd, or a no parity scheme is required and each Transmitter/Receiver pair shall generate, check or inhibit parity as determined by software.

(e) The character length shall be selectable to five, six, seven, or eight bits.

(f) The Transmitter/Receiver pair shall select one or two stop bits in addition to one start bit. One and one half stop bits may be utilized if two stop bits are selected for a five bit character.

(g) The modem control signals shall conform to RS-232C for asynchronous communications to be used with Bell 113 A/B, 103 and 202 S/T modems. A detailed description must be part of the proposal identifying connector types and signal pin connections.

(h) The proposal must describe any loopback testing facilities available for determining if the computer interface and multiplexor is operating correctly.

(i) Terminal interfaces must include the standard RS-232C interface.

(j) Terminal ports must accept either hard-wire or service through a dial up or hard wired modem. All these are required.

(k) Acceptable data rates must include 300, 1200, 2400, 4800 and 9600 bits per second.

d. *Peripheral Devices*

(1) CRT Terminals

(a) Keyboard must provide, as a minimum, the standard ASCII 96-character subset. Proposal shall include a table showing the CRT image of each of the 96 characters. Furthermore, the system must be capable of recognizing, transmitting, processing, storing, retrieving, displaying, and printing a full alphanumeric upper and lower case character set.

(b) CRT screen display area must be approximately nine (9) inches wide by seven (7) inches high as a minimum.

(c) CRT screen display area must have a capacity of not less than 1,920 characters displayed at not less than 80 characters per line.

(d) It is desired that CRT display characters consist of, or be equivalent to, a dot matrix of not less than 7 × 9 dots per character.

(e) The CRT display must be fully buffered and the refresh rate must be sufficient to maintain full intensity of each individual character over the entire display area without flicker or other distortion.

(f) The CRT unit must provide a cursor which functions as a visible positional indicator. It must be possible for the terminal operator to change the position of the cursor—left, right, up, down—by means of function keys on the CRT keyboard unit. Also, it is desirable to have the ability to tab or skip the cursor ahead to positions previously determined by the application program.

(g) Under program control, it is desirable to emphasize or highlight individual characters or character positions by means of high-low dual intensity reverse polarity or blinking, and it must be possible by program control to protect selected character positions from being modified.

(h) It is desired that the CRT unit have the ability to accept from the currently executing application program a visually formatted data entry screen display; project that display on the CRT screen in protected mode; then, without further CPU attention, allow the terminal operator to key-enter data into the specified input fields and to selectively re-key into these fields for correction of errors visually detected by the terminal operator; and finally, when initiated by terminal operator action transmit only data currently displayed in the input field locations.

(i) CRT terminal must have an RS-232-C communications interface supporting user-selectable data transfer rates including 300 and 1200 baud.

(j) In addition to the terminal specified in (a) through (i) above, a software compatible alternative terminal not having lower-case capability is desirable. LC may choose to install a lower cost alternate terminal in situations where upper/lower case capability is not required.

(k) It is desirable to support the Library of Congress MARC II character set. This character set utilizes the parity bit to derive characters over and above the standard ASCII characters. It is requested that this capability be addressed.

(l) The CRT system must have a minimum of 10 function keys associated with it to enable selection of user functions. This requires the function keys to be identifiable by the applications code. It is desired that these keys be internal to the CRT keyboard but proposals for a co-located function key box will be entertained.

(m) It is desired that the CRT have an input I/O port which essentially parallels the keyboard. It shall conform to RS-232C serial transmission and be of suitable baud rate to accommodate a bar code wand reader or OCR reader with ASCII output.

(n) The CRT must have an output I/O port which shall be capable of transmission to the document printer specified in paragraph d. (5). This port will be used for printing the screen image.

(o) A desirable option is a terminal such as the one described above but with a 58 line × 80 character display for word processing applications. Refer to section 3.g for details of word processing.

(2) Direct Access Storage Devices

(a) User data/program storage

- System capacity for at least four separate online disk storage devices, each with removable data storage media (disk pack) "physically" compatible with IBM 3330 packs, IBM 2314 or CDC 9876 Trident packs. The average and maximum disk access times shall be stated including seek times and rotational latency. Average access time is defined as seek time across one-half of tracks plus one-half total rotational delay and shall not be greater than 50 millisec.
- System capacity to be not less than 300 million bytes of concurrently online direct access user data storage.
- System (disk subsystem in conjunction with operating system I/O drivers) must be capable of recognizing and bypassing bad tracks. Bidder must describe methodology of this capability.

(b) Systems residence device (optional)

- In view of the relatively large number of concurrent users to be supported, it is anticipated that bidders will find it desirable to include in their proposed system configuration a high-performance system residence device for program overlays, swap space, and so on. Because this device is optional, its detailed specifications are left up to the discretion of the individual bidder.

(3) Magnetic Tape Drives (optional)

(a) System must be capable of supporting up to four (4) tape drives in increments of one, starting with one drive. These drives must have the following characteristics:

(1) 9-track
(2) 800 or 1600 bpi — Preferably 1600 bpi
(3) 75 inches per second read/write speed minimum
(4) 2400 foot reel capacity
(5) ANSII compatible
(6) IBM readable

(b) In order to reduce media inventory and minimize operator intervention, LC is considering *not* including magnetic tape drives in these minicomputer systems. Bidder shall describe the impact (if any) this action would have in regard to each proposed system with particular attention given to aspects of system reliability, operation, and maintenance.

(4) Line Printer

(a) Each system will be equipped with one line printer.

(b) The sustained printer speed must be 200 lines/minute or better. Printer speed is defined as the print rate using a standard ASCII set of 96 characters and printing on all 132 print positions with equal use of all characters.

(c) Six lines/inch capability is required with six or eight lines/inch desirable.

(d) There must be a minimum of 132 print positions.

(e) The printer must be able to print mutiple copies up to six-part forms.

(f) The printer must be equipped with vertical and horizontal alignment controls for special forms, carriage control, and line feed suppression.

(g) The printer must have an adjustable forms width capability to accommodate forms ranging from 11 inches to 14 7/8 inches

(h) In reference to the extended character set it is requested that the line printer interface be such that it would not preclude the use of a full 8 bit code transmission to accommodate the MARC II character set.

(5) Document Printer

(a) There should be a minimum of 80 print positions with a capability of transmission of at least 30 characters per second.

(b) Standard 96 character ASCII (includes upper/lower case).

(c) Multi-part form capability is desirable.

(d) Character formation/impression mechanism must produce clear, "crisp" characters which will reproduce satisfactorily on Xerographic process office copier machines. Bidders shall include in their proposals a sample page printed by the document printer they are proposing.

(e) Printer must be capable of being located and operated remotely via the communications subsystem.

(f) In addition to document printer specified in (a) through (e) above, a software compatible alternative not having lowercase capability must also be operationally available. LC may choose to install lower-cost alternate document printers in situations where upper/lower case capability is not required.

(g) Additional capabilities are desired for the word processing subsystem explained in section 3. g. These are as follows:

(1) Variable pitch (either 10 or 12) and have readily changeable typefonts (not to exceed 5 minutes change time).

(2) In addition to upper/lower case alpha and numeric, the special symbols period, comma, colon, semicolon, plus, section, cent, parenthesis, brackets (squared), quotation, asterisk, question mark, number, exclamation mark, percent, dollars, dash, underline, slash and apostrophe are required.

(h) It is requested that the MARC-II character set discussed in paragraph d. (1)(k) be addressed here also.

(6) IBM Channel Interface

It is highly desirable to have the capability of interfacing the proposed systems to the IBM 370 via a selector and/or multiplexor channel. Response to this requirement should be documented evidence as to both hardware required and software available to accommodate the hardware. Both the minicomputer and IBM 370 system requirements and prerequisites should be identified along with data rate limitations.

3. System Software

General

a. *Operating System*

(1) A complete, fully supported operating system is required that is capable of supporting all peripherals and managing all system resources. This operating system must be capable of supporting a minimum of 60 concurrent interactive terminal users executing a mix of application procedures. If the terminal support is not part of the operating system describe any communication software available to perform this function. Areas of interest include buffer size requirements, amount of memory required for the package and interface to high-level languages.

(2) Both generic and specific program references to the line printer and other devices are desirable.

(3) All peripherals must be accessible through software from terminals.

(4) It is desirable to have a priviliged class of users permitting access and modification of system functions that are not available to non-privileged users.

(5) It is desirable to have a capability to dynamically "tune" the operating system to optimize performance for changing workloads.

(6) It must be possible to deny certain users access to certain system resources, such as disk space and CPU time.

(7) Program development must be possible from a minimum of two terminals.

(8) Capability MUST be provided for disk storage file backup. Offerors in responding should provide a measure of the time required to perform backup. Show all calculations and list the assumptions used for these calculations.

(9) A software library facility is required for frequently used programs.

(10) It is desired that it be possible to assign user priorities and for the system manager to define limits on user activity.

(11) Terminals must be addressable so that the operator can communicate with, and as a privileged user, interrupt individual users.

(12) Spooling to disk for line printer output is mandatory.

(13) There must exist a comprehensive file management system.

(a) Binary data as well as formatted data must be storable on disk and magnetic tape.

(b) File accessing from a program must be device independent.

(c) File structure must be language independent. Files created in one language must be accessible by other(s).

(d) Files created in batch mode must be fully compatible with and accessible in multiple user mode and vice versa.

(e) Files must be sharable. More than one user must be able to access the same file concurrently. Protection *against* concurrent access via user option must be also available.

(f) Record formats supported should include fixed, fixed block, variable, and variable blocked.

(g) File access methods supported should include sequential, index sequential, random, and direct.

(h) Bidders shall describe the file allocation methodology of their software, stating how this allocation methodology functions in a multi-user online interactive environment.

(i) If the requirements (a) through (h) above require a DBMS, bidders shall provide complete functional and operational description of their proposed DBMS (see item "f" below). Particular emphasis shall be placed upon the performance impact of such software.

(j) The file management system *must* have the capability of generic and approximate keyed retrieval. Generic is defined as having one or more of the leading characters of the key and approximate is defined as searching for the closest key such that

$\mathrm{KEY}_R \geq \mathrm{KEY}_F$ or $\mathrm{KEY}_R \leq \mathrm{KEY}_F$

Where KEY_R is the key requested and KEY_F is the key found.

(14) If the proposed software contains any system limitations such as number of files per disk drive, maximum file size, maximum record size, and so on, such specifications must be stated in the proposal.

(15) It is required that the operating system be able to distinguish between system software activity and user application software activity so that the system can protect itself from user errors.

(16) It is highly desirable that the system be capable of accepting a "down-line-load" from a remote host system. Bidders shall provide a comprehensive description of the methodology employed.

(17) Vendor should describe availability and cost of source code for operating system, language processors, and all other vendor-supplied software.

b. *Language Processors*

(1) Describe "high-level" multi-user language capability as applicable to the development of application programs. Specify the extent of multiprogramming, multitasking, and re-entrancy of code generated by available compilers. Highly desirable high-level languages include COBOL, FORTRAN, RPG, BASIC, ALGOL, and PL/1.

(2) Specify and describe the user environment in which application programs would be developed, tested, and implemented. Include a scenario of the coding, integration, and testing process.

(3) Describe and provide examples of accomplishing data storage and retrieval in a high-level language—particularly data with alphanumeric "keys" of at least 25 characters in length.

(4) Describe the available MACRO assembler, if any.

c. *Query and Source Data Entry Package*

(1) A generalized package of this type is not mandatory. If the vendor wishes to describe a supplementary package which would be useful to LC, they may do so.

d. *Sort/Merge and General Utilities*

(1) Provide a "sort/merge" capability and describe typical performance results. State the time required to sort from 1,000 to 20,000 records in 1,000 record increments, listing the system, disk, and core resources required, assuming no other active tasks, a 12-character sort key, and a 200-character record. It is desirable that this sort/merge be callable from application programs in either high-level language or assembly language.

(2) Describe the system's ability to perform general utility functions and the environment in which they are performed (i.e., under language interpreter or operating system control).

(3) Describe the software for performing disk pack backup copies and typical times involved.

e. *Communications*

(1) Describe software for emulating/simulating a 2780/HASP/JES/RES IBM compatible work station for 370 batch communications.

(2) Describe the capability (if any) for a proposed system to communicate with another proposed system.

f. *Data Base Management System* (Optional—Desired)

(1) General

It is desired that the DBMS relieve the application programmer of concern for the overall organization, structure and maintenance of the data base. The DBMS should insulate the application program from equipment differences among storage devices.

(2) Host Languages

The DBMS system should provide a high-level language interface capability to support application programs development. The DBMS should include the capability for such application programs to utilize all of the DBMS functions either through language extensions or supported call statements.

(3) Processing Modes

The DBMS should have the capability to support both batch and interactive modes of processing in a multi-user environment; that is, multiple terminals/programs should be able to update/retrieve from the same data base concurrently.

(4) Data Definition

There should be a facility in the DBMS to support the definition of data to the application program. Included in the definition capability should be features such as the following:

(a) The definition of data bases and keys.

(b) The assignment of symbolic names to data bases, data elements, and groups of data elements.

(c) Characteristics of each data element such as size, number of occurrences, and type, such as alpha or numeric.

(d) A listing of data definitions with diagnostics.

(5) Data Base Recovery

The capability to recover the entire data base in a timely manner is imperative. In this respect, several requirements are specified below.

(a) Ability to copy the entire data base or specified portions onto an auxiliary storage media upon command.

(b) Ability to restore the data base from a previously generated copy along with all associated indices, pointers, and directories.

(6) General Capabilities Desired

The following capabilities are desirable and should be available to users through the query language and directly from a program written in the proposed host languages.

(a) Efficient space allocation and de-allocation.

(b) Variable length records via repeating groups or attached members.

(c) Multiple indexes on a single file with duplicate keys allowed.

(d) Randomized and tree index structures.

(e) Data dictionary.

(f) Minimal record and file size limitations.

(g) Ability to insert, delete and update records in a data base without necessitating its recreation.

(h) Ability for multiple users in all processing modes to concurrently access a common data base.

(i) Specify all utilities required to maintain the data base include vendor supplied or customer required.

(j) The ability to create and retrieve on sub-indexes is desirable.

g. *Word Processing Subsystem* (desired)

(1) General

It is desired that a software system which would facilitate word processing applications be described if available. This system should be oriented around a full page (58 lines of 80 characters or standard 8½ × 11) CRT. The emphasis is placed on ease of use and ease of training of relatively unskilled typists. If the vendor does not have software of this type, LC will entertain recommendations as to where it might be obtained.

(2) Software

The software would be preferred in a high-level language to facilitate easy support. It will be required that the source code for the subsystem be supplied with delivery of the system. Furthermore it is required that a full, detailed configuration and operating system environment description be provided as part of the proposal with a general description as to whether it is a stand alone system or concurrent running with other application programs.

(3) The following word processing functions are desired.

WORD PROCESSING MINIMUM CAPABILITIES

1. Insert character
2. Insert line
3. Delete character
4. Delete line
5. "Edit" or "Break" for variable length input of additional text.
6. Erase (Deletion of character that leaves a space).
7. Move Character
8. Move line
9. Move paragraph
10. Move page
11. Must be able to print and edit or keyboard simultaneously.
12. Local storage of minimum of 30 pages of text (1 page= 200 characters).
13. Keyboard commands and use of system training not to exceed 20 hours for full capability. Employee should be able to be completely productive (maximum productivity) by end of 80 hours (includes training time).
14. Printer must be capable of printing total screen text (58 lines × 120 characters).
15. Should have global search and replace
16. Automatic pagination
17. Ease of handling footnotes
18. Decimal justification
19. Diacritics

(4) File handling capabilities

In addition to the word processing requirements it is required that files or documents be accessed via keyed retrieval methods as described in section 3.a.13. Provisions must be made such that several individuals can be working on the same document concurrently and have the system reassemble the document into proper sequential order.

(5) The system should also be capable of communicating with the LC mainframe CPU, IBM 370/158 with appropriate communication link control as described in section 3.e.

h. *General Requirements*

(1) The equipment design shall provide for maximum useful service life under twenty-four hour/day, seven day/week operation as a design goal, obsolescence not withstanding. Useful service is defined as the continued ability to meet the performance reliability and maintainability specifications herein, assuming that preventive maintenance in accordance with the vendor's peroposal is practiced. Vendor shall state minimum and maximum useful service life and warranty. The equipment will require *no* special facility to be constructed. The operating environment shall be that of normal office conditions.

(2) The equipment shall operate and maintain specified performance from a single phase power source of 115V+- 10%, 60 cycles +- 1 cycle, or a three phase, four wire power source of 115/208 V +- 10%, 60 cycle +- 1%.

(3) The vendor shall insure that all equipment will operate as specified in the probable internal and external electromagnetic and electrostatic environments without loss in performance and without, through the electromagnetic energy it generates, causing other equipment to malfunction. Designed suppression of radiated and constructed interference shall permit the equipment to operate without error from incoming interference sources such as florescent lighting, power lines, general purpose high-speed digital computers and their auxiliary equipment. It shall be the vendor's responsibility to inspect the site and review equipment installation plans with LC.

(4) Electromagnetic or other radiation hazards potentially harmful to operating or maintenance personnel shall be constrained within the limits established by the National Council on Radiation Protection and Measurements.

(5) The equipment shall have UL approval and perform as specified while operating under the following conditions:

- Ambient Temperature Range. 50 to 90 degrees F provides there is no condensation.
- Relative Humidity. 10% to 90% within the ambient temperature range.
- Other Atmospheric Conditions. Any probable combination of smoke, ozone, fine dust, or any other atmospheric conditions which are likely to be encountered in an office building.

(6) The vendor will provide detailed specifications of the proposed system as an entity and define characteristics for system integrity. These shall include, but not be limited to:

- Detailed listing of system components.
- Dimensions of all items of equipment and proposed installation layout.
- Power supply requirements.
- Overload protection.
- Required system operating environment, particularly cooling and electrical.
- Modular growth characteristics and estimated useful life
- Detailed description of the technical characteristics of each component including MTBF (Mean Time Between Failure), MTTR (Mean Time to Repair), and MTBI (Mean Time Between Interrupt — meaning abnormal termination of operating system software.) The vendor must meet or exceed the following figures or explain to the satisfaction of the LC why his figures should be accepted as a reasonable alternative:
  - MTBF: See paragraph i.3. which follows
  - MTTR: See paragraph i.4. which follows
  - MTBI: See paragraph i.5. which follows
- Workload processing capability.
- Operational safeguards.
- Software interfaces.
- Hardware interfaces.
- Previous operational applications.
- Performance expectation (Throughput, etc.).

(7) Specify the expandability or maximum capability of the system for memory, disk storage, CRTs, and printers.

ERIC Full Text Provided by ERIC

(8) Specify the source of all hardware and software being proprosed which has not been directly produced or manufactured by the vendor.

(9) Provide the name, title, address, and telephone number of the vendor's representative who bears sole responsibility for this proposal and ensuing contract. This representative would be contacted only in the event of unusual circumstances; day-to-day situations would be handled at a lower organizational level.

i. *Reliability*

Performance capability of the system will be based upon calculations from demonstrated field experience. Definitions and establishment of the quantitative criteria follow:

(1) A system is the basic configuration of CPU's memory, DISKS, and tapes (if the tape option is exercised) and will *not* include any of the terminals supported by the system, telephone lines, cables, or utility power.

(2) *Failure* : A failure is defined as one or more of the following:

(a) Any component failure in a CPU causing the system to cease functioning.

(b) Failure of the internal system power supplies to deliver the correct voltage

(c) Failure of any component in the memory which disables usage

(d) Failure of any component in the controller (if one is used) controlling the two synchronous channels

(e) Failure of both of the synchronous channels

(f) Failure of any component in the controller (if used) controlling the asynchronous channels.

(g) Failure of more than ten percent of the asynchronous channels.

(h) The system fails to operate within the specified environmental limits.

(3) *MTBF* : The demonstrated mean time between failure for a population of systems shall be 12,000 hours as a goal. The goal should be realized in a one year time period. The LC will evaluate at the end of this time period whether the goal was met and the suitability of the proposed system.

The MTBF shall be calculated as follows:

The system will be utilized over a 12 hour period daily. Failures occurring outside this 12 hour period will not be charged against the system provided they have been repaired before start up time the following day. It is desired that an unattended system be able to report itself "in trouble" to the main 370/158 at the Library of Congress so that service personnel may be dispatched.

Overall system reliability may be accomplished by one or more of the following suggested approaches:

(a) Dynamic reconfiguration which is achieved by online test and diagnostic capability coupled with sufficient hardware sophistication for the system to realize a module is in trouble, configure the module out, report the failure, and not interrupt operation. *This is the desired approach.*

(b) Manual reconfiguration which may be accomplished by a user or by an operator at a central terminal or at the 370/158 site. A short interruption occurs.

(4) *MTTR*: Mean time to repair is desired to be less than 30 minutes (after the field engineer has arrived at the site). The proposal shall identify all evidence leading to this conclusion.

(5) *MTBI:* Mean time between interruptions shall apply to interruptions caused by either the operational system software or the system hardware. It is expected that interruptions will occur more frequently immediately following installation of the system, but will be expected to improve once the burn in period has elapsed. MTBI is intended to cover those interruptions which cannot be traced to either system software or hardware as well as those that can be and is included to encourage the offerer to quote tested stable systems as far as practical. The following MTBI figures are desired:

(a) For the first month—100 hours

(b) For the next two months—500 hours

(c) For the next three months—1,000 hours

(d) For the remainder of the time—during each working day of 12 hours or less the reliability shall be 0.999 equivalent to one tenth or one percent probability of failure, equivalent to a mission MTBI of 12,000 hours.

(6) *Design Life:* The design life of the equipment is to be 10 years at 24 hours per day operation.

j. *Design Approach*

We require that the reader provide us with full details of the design approach bid, including the architecture and logic for achieving the required reliability and/or failure tolerance. We will entertain all sound design concepts that can achieve the desired reliability including n-processor approaches.

BIBLIOGRAPHY

Alper, Bruce H. "Library Automation." In *Annual Review of Information Science and Technology.* Volume 10, pp. 199–236. Edited by Carlos A. Cuadra and Ann W. Luke. Washington, D.C.: American Society for Information Science, 1975.

Anderson, Decima M. *Computer Programming: FORTRAN IV.* New York: Appleton-Century-Crofts, 1966.

Anderson, Walter L. "Minis Are Beautiful!; Keynote Address." In *Minicomputers: The Applications Explosion,* n.p. Edited by David E. Debeau. Proceedings of the American Institute of Industrial Engineers Conference held in Washington, D.C., 17–19 November 1975. Los Angeles: Management Education Corporation, 1975.

Auerbach Buyers' Guide to . . . Business Minicomputer Systems, Winter 1976–77. Pennsauken, N.J.: Auerbach Publishers, 1977.

Auerbach On Minicomputers. New York: Petrocelli Books, 1974.

Awad, Elias M., and Data Processing Management Association. *Automatic Data Processing; Principles and Procedures.* 3d ed. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1973.

Bailey, David C. "Requiem for the Punched Card?" *Modern Data* 8 (October 1975): 66–70.

Barden, William, Jr. *How to Buy & Use Minicomp[illegible] Microcomputers.* Indianapolis: Howard W. Sams & Co., 1976.

Barna, Arpad, and Porat, Dan I. *Introduction to Microcomputers and Microprocessors.* New York: John Wiley & Sons, 1976.

Becker, Joseph. "A Brief History of On-Line Bibliographic Systems." Speech given at the UCLA Conference on Information Systems and Networks, 27–29 March 1974.

Becker, Joseph, and Hayes, Robert M. *Information Storage and Retrieval: Tools, Elements, Theories.* New York: John Wiley & Sons, 1963.

Bobick, Steve A.; Armon, Edmund J.; and Yerkes, Arthur W. "Survey of Small Business Computers." *Datamation* 22 (October 1976): 91–107.

Bohl, Marilyn. *Information Processing.* 2d ed. Chicago: Science Research Associates, Inc., 1976.

Bowers, Dan M., ed. "[Computer Printers] Manufacturers' Data." *Modern Data* 8 (November 1975): 43–45.

——. "Floppy Disk Drives and Systems; Part 1. Historical Perspective." *Mini-Micro Systems* 10 (February 1977): 36–51.

——. "Printers and Teleprinters." *Mini-Micro Systems* 10 (January 1977): 30–53.

——. "Removable Disk Storage; Where It's Come From and Where It's Going." *Modern Data* 9 (January 1976): 36–38.

——. "Small-Scale Computi[illegible] It's Like Doing Your Laundry; Part 1. Computation and Processing." *Modern Data* 8 (May 1975): 43–46.

——. "Small-Scale Computing: It's Time to Get With the Future; Part 2. Data Entry." *Modern Data* 8 (June 1975): 43–49.

Brophy, Peter; Buckland, Michael K.; and Hindle, Anthony. *Reader in Operations Research for Libraries.* Englewood, Colo.: Information Handling Services, Library and Education Division, 1976.

Butler, Brett. "State of the Nation in Networking." *Journal of Library Automation* 8 (September 1975): 215–6.

Butler, J. L. "Comparative Criteria for Minicomputers." In *A Practical Guide to Minicomputer Applications,* pp. 77–92. Edited by Fred F. Coury. New York: IEEE Press, 1972.

Caswell, Stephen A., ed. "Cassette Drives and Systems." *Modern Data* 8 (October 1975): 58–63.

Chapin, Ned. *An Introduction to Automatic Computers.* New York: Van Nostrand, 1957. Cited by Robert M. Hayes and Joseph Becker. *Handbook of Data Processing for Libraries,* p. 237. 2d ed. Los Angeles: Melville Publishing Co., 1974.

Chapman, Edward A.; St. Pierre, Paul L.; and Lubans, John, Jr. *Library Systems Analysis Guidelines.* New York: Wiley-Interscience, John Wiley & Sons, 1970.

Corey, James J. "Configurations and Software: A Tutorial." In *Applications of Minicomputers to Library and Related Problems,* pp. 11–27. Edited by J. Wilfrid Lancaster. Papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974. Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974.

Corliss, William R. *Computers.* Rev. Understanding the Atom Series. Oak Ridge, Tenn.: U. S., Atomic Energy Commission, Division of Technical Information, 1967.

Cox, N. S. M.; Dews, J. D.; and Dolby, J. L. *The Computer and the Library; The Role of the Computer in the Organization and Handling of Information in Libraries.* Newcastle upon Tyne, Eng.: University of Newcastle upon Tyne Library, 1966.

Curley, Walter. "Innovative Strategies in Systems and Automation." In *Library Automation: The State of the Art II*, pp. 127-38. Edited by Susan K. Martin and Brett Butler. Papers presented at the Preconference Institute on Library Automation sponsored by the Information Science and Automation Division of the American Library Association at Las Vegas, Nevada, 22-23 June 1973. Chicago: American Library Association, 1975.

"Data Base for the Mini User." *Mini-Micro Systems* 9 (June 1976): 30.

Davis, Charles H. *Illustrative Computer Programming for Libraries; Selected Examples for Information Specialists*. Westport, Conn.: Greenwood Press, 1974.

Davison, Wayne. "Minicomputers and Library Automation: The Stanford Experience." In *Applications of Minicomputers to Library and Related Problems*, pp. 80-95. Edited by J. Wilfrid Lancaster. Papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974. Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974.

DeGennaro, Richard. "Library Automation: Changing Patterns and New Directions." *Library Journal* 101 (January 1976): 175-83.

Delanoy, Diana. "Technology: Present Status and Trends in Computers." In *Library Automation: The State of the Art II*, pp. 18-37. Edited by Susan K. Martin and Brett Butler. Papers presented at the Preconference Institute on Library Automation sponsored by the Information Science and Automation Division of the American Library Association at Las Vegas, Nevada, 22-23 June 1973. Chicago: American Library Association, 1975.

Demitriades, Paul B. "Mini Update." *Journal of Systems Management* 25 (December 1974):

Dunn, Donald A. "Communications Technology." In *Annual Review of Information Science and Technology*. Volume 10, pp. 165-93. Edited by Carlos A. Cuadra and Ann W. Luke. Washington, D.C.: American Society for Information Science, 1975.

Eckhouse, Richard J., Jr. *Minicomputer Systems; Organization and Programming (PDP-11)*. Englewood Cliffs, N.J.: Prentice-Hall, 1975.

Eyre, John, and Tonks, Peter. *Computer & Systems; An Introduction for Librarians*. London: Clive Bingley, 1971.

Flores, Ivan. *Peripheral Devices*. Englewood Cliffs, N.J.: Prentice-Hall, 1973.

Fussler, Herman H. *Research Libraries and Technology; A Report to the Sloan Foundation*. Chicago: University of Chicago Press, 1973.

Gildenberg, Robert F. *Computer-Output-Microfilm Systems*. Los Angeles: Melville Publishing Co., 1974.

GML. *Minicomputer Review 1975*. Lexington, Mass.: GML Corporation, 1975.

Grosch, Audrey N. "Library Automation." In *Annual Review of Information Science and Technology*. Volume 11, pp. 225-66. Edited by Martha E. Williams. Washington, D.C.: American Society for Information Science, 1976.

____. "Minicomputer—Characteristics, Economics and Selection for an Integrated Library Management System." In *Applications of Minicomputers to Library and Related Problems*, pp. 158-69. Edited by F. Wilfrid Lancaster. Papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974. Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974.

____. "Mini-Computer Systems for Library Management Applications; A New Approach to Bibliographic Processing." In *Computers in Information Data Centers*, pp. 25-33. Edited by Joe Ann Clifton and Duane Helgeson. Montvale, N.J.: AFIPS Press, 1973.

Hayes, Robert J., and Becker, Joseph. *Handbook of Data Processing for Libraries*. 2d ed. Los Angeles: Melville Publishing Co., 1974.

Heiliger, Edward M., and Henderson, Paul B., Jr. *Library Automation: Experience, Methodology, and Technology of the Library as an Information System*. New York: McGraw-Hill Book Co., 1971.

Hoffman, A. A. J.; French, Robert L.; and Lang, Guy M. "Minicomputer Interfaces: Know More, Save More." *IEEE Spectrum* 11 (February 1974): 64-68.

Hollingworth, Dennis. *Minicomputers: A Review of Current Technology, Systems, and Applications*. Santa Monica, Calif.: Rand Corporation, 1973.

Hughes, John Lee. "Maxi to Mini: A Citibank Case History; [Speech]" In *Minicomputers: The Applications Explosion*, n.p. Edited by David E. Debeau. Proceedings of the American Institute of Industrial Engineers Conference held in Washington, D.C., 17-19 November 1975. Los Angeles: Management Education Corporation, 1975.

Huskey, Harry D. "Computer Technology." In *Annual Review of Information Science and Technology*. Volume 5, pp. 73-85. Edited by Carlos A. Cuadra and Ann W. Luke. Chicago: Encyclopaedia Britannica, Inc., 1970.

International Business Machines. *Data Processing Glossary*. 3d ed. Poughkeepsie, N.Y.: International Business Machines Corp., Programming Systems Publications, 1971.

____. *IBM System/360 Operating System Basic Telecommunications Access Method*. 3d ed. IBM Systems Reference Library. White Plains, N.Y.: International Business Machines Corp., Data Processing Division, 1968.

____. *Student Text Introduction to IBM Data Processing Systems*. 2d ed. White Plains, N.Y.: International Business Machines Corp., Technical Publications Department, 1968.

Introduction to Minicomputer Networks. Maynard, Mass.: Digital Equipment Corporation, 1974.

Jarosz, Marty B. "Minicomputers—Microcomputers—Peripherals; What Are the Real Cost Hang-Ups?" *Mini-Micro Systems* 9 (May 1976): 79–88.

Juliussen, J. Egil. "The Cost Outlook for Peripheral Controllers." *Mini-Micro Systems* 10 (January 1977): 64–66.

Kallis, Stephen A. "Networks and Distributed Processing." *Mini-Micro Systems* 10 (March 1977): 32–40.

Kershner, Lois M. "Management Aspects of the Use of the IBM System/7 in Circulation Control." In *Applications of Minicomputers to Library and Related Problems*, pp. 43–54. Edited by F. Wilfrid Lancaster. Papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974. Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974.

———. "User Services: 1973 Applications Status." In *Library Automation: The State of the Art II*, pp. 38–55. Edited by Susan K. Martin and Brett Butler. Papers presented at the Preconference Institute on Library Automation sponsored by the Information Science and Automation Division of the American Library Association at Las Vegas, Nevada, 22–23 June 1973. Chicago: American Library Association, 1975.

Khtaian, George A. "Cost/Vendor Evaluation System." *Journal of Systems Management* 26 (August 1975): 11–19.

Kimber, R[ichard] T. *Automation in Libraries*. 2d ed. Oxford: Pergamon Press, 1974.

Lancaster, F. Wilfrid, ed. *Networking and Other Forms of Cooperation*. Papers presented at the 1973 Clinic on Library Applications of Data Processing, 29 April to 2 May 1973. Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1973.

Long, Philip L. "Computer Technology—An Update." In *Annual Review of Information Science and Technology*. Volume 11, pp. 211-22. Edited by Martha E. Williams. Washington, D.C.: American Society for Information Science, 1976.

Lourey, Eugene D. "Systems Design for a Minicomputer-Based Library Data Management System." In *Applications of Minicomputers to Library and Related Problems*, pp. 181–90. Edited by F. Wilfrid Lancaster. Papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974. Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974.

Lubans, John, Jr., and Chapman, Edward A. *Reader in Library Systems Analysis*. Englewood, Colo.: Microcard Editions Books, 1975.

Lyndes, Horace. "A Wake for Paper Tape?" *Modern Data* 8 (October 1975): 71–72.

MacCafferty, Maxine, comp. *An Annotated Bibliography of Automation in Libraries and Information Systems 1972–1975*. London: Aslib, 1976.

Markuson, Barbara Evans; Wanger, Judith; Schatz, Sharon; and Black, Donald V. *Guidelines for Library Automation; A Handbook for Federal and Other Libraries*. Santa Monica, Calif.: System Development Corporation, 1972.

Marshall, Joseph C. "Distributed Processing on Wall Street." *Datamation* 19 (July 1973): 44–46.

Moore, William G., Jr. "Going Distributed." *Mini-Micro Systems* 10 (March 1977): 41–48.

Neumann, A. J. *A Guide to Networking Terminology*. Washington, D.C.: U.S., Department of Commerce, National Bureau of Standards, 1974.

Nicolaus, John J. "Library Automation—How to Begin: Initiating a Library Automation Program." In *Initiating a Library Automation Program*, pp. 9–23. Papers presented at the 1965–1966 meetings of the Documentation Group, Washington, D.C. Chapter, Special Libraries Association, 1965–66.

Ollivier, Robin T. "A Technique for Selecting Small Computers." In *A Practical Guide to Minicomputer Applications*, pp. 93–97. Edited by Fred F. Coury. New York: IEEE Press, 1972.

Palmer, Richard Phillips. *Case Studies in Library Computer Systems*. New York: R. R. Bowker, Xerox, 1973.

Patrinostro, Frank S., comp. *A Survey of Commonplace Problems in Library Automation*. The LARC Association's World Survey Series Volume 11. Peoria, Ill.: LARC Press Ltd., 1973.

Payne, Charles T. "The University of Chicago Library Data Management System." In *Applications of Minicomputers to Library and Related Problems*, pp. 105–119. Edited by F. Wilfrid Lancaster. Papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974. Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974.

Pearson, Karl M., Jr. "Minicomputers in the Library." In *Annual Review of Information Science and Technology*. Volume 10, pp. 139-63. Edited by Carlos A. Cuadra and Ann W. Luke. Washington, D.C.: American Society for Information Science, 1975.

"Peripherals Make the Mini; *Modern Data's* Annual Survey of Plug-Compatible Miniperipherals." *Modern Data* 8 (December 1975): 34–43.

Pezzanite, Frank A. "Distributed Library Networking: A New Approach for Maryland." Speech given at the Workshop on Computerized Library Networks co-sponsored by the Maryland State Department of Education and the National Library of Medicine, 8-9 April 1976.

Puckorius, Theodore. "Trends in Government Acquisition of High Technology ADP Equipment [Speech]." In *Minicomputers: The Applications Explosion*, n.p. Edited by David E. Debeau. Proceedings of the American Institute of Industrial Engineers Conference held in Washington, D.C., 17-19 November 1975. Los Angeles: Management Education Corporation, 1975.

"Questions and Answers Department." *Library Technology Reports* 13 (September 1976): 465-9.

Radwin, Mark S. "Choosing a Terminal; Part 2." *Online* 1 (April 1977): 61-73.

———. "From Nodes to Modes—Duplex and Half-Duplex." *Online* 1 (January 1977): 13-19.

———. "The Intelligent Person's Guide to Choosing a Terminal for Online Interactive Use; Part 1." *Online* 1 (January 1977): 11.

Reiche, Harris G. "New Directions for ADP Procurement [Speech]." In *Federal ADP Procurement*, n.p. Edited by David T. Newman. Proceedings of the American Institute of Industrial Engineers Conference held in Washington, D.C., 1-3 November 1976. Santa Monica, Calif.: Management Education Corporation, 1975.

Reynolds, Barbara A., ed. "Alphanumeric Display Terminals; Part 1. The CRT Market and Technology—Where They Stand Now." *Modern Data* 9 (February 1976): 44-51.

———. "Alphanumeric Display Terminals; Part 2. Who's Who in CRTs and Where the Market is Going." *Modern Data* 9 (March 1976): 44-51.

———. "Removable Disk Cartridge Drives." *Modern Data* 9 (January 1976): 38-43.

Ritchie, Robert O. "Intelligent Terminals and Distributed Processing." *Computer Decisions* 7 (February 1975): 36-40.

Runyon, Stanley. "Microprocessors in Test Equipment." In *Microprocessors: New Directions for Designers*, pp. 20-25. Edited by Edward A. Torrero. Rochelle Park, N.J.: Hayden Book Co., 1975.

Russell, William D. "Application of OMB Circular A-76 to ADP; [Speech]" In *Federal ADP Procurement*, pp. 129-37. Edited by David T. Newman. Proceedings of the American Institute of Industrial Engineers Conference held in Washington, D.C., 1-3 November 1976. Santa Monica, Calif.: Management Education Corporation, 1975.

Salmon, Stephen R. *Library Automation Systems*. New York: Marcel Dekker, 1975.

Salton, Gerard. *Dynamic Information and Library Processing*. Englewood Cliffs, N.J.: Prentice-Hall, 1975.

Schultheiss, Louis A.; Culbertson, Don S.; and Heiliger, Edward M. *Advanced Data Processing in the University Library*. New York: Scarecrow Press, 1962.

Simms, R. L., and Fuchs, Edward. "Communications Technology." In *Annual Review of Information Science and Technology*. Volume 5, pp. 113-39. Edited by Carlos A. Cuadra and Ann W. Luke. Chicago: Encyclopaedia Britannica, 1970.

Sippl, Charles J., and Kidd, David A. *Microcomputer Dictionary and Guide*. Champaign, Ill.: Matrix Publishers, 1975.

Stein, Philip G., and Shapiro, Howard M. "What Makes OS Racing." *Computer Decisions* 6 (November 1974): 46-47.

Stiefel, Malcolm L., ed. "Small Business Computers." *Mini-Micro Systems* 9 (July 1976): 50-57.

———. "Source Data Automation (Part 2)" *Mini-Micro Systems* 9 (June 1976): 38-43.

Strauss, Lucille J.; Strieby, Irene M.; and Brown, Alberta L. *Scientific and Technical Libraries; Their Organization and Administration*. New York: Interscience Publishers, John Wiley & Sons, 1964.

Swihart, Stanley J., and Hefley, Beryl F. *Computer Systems in the Library: A Handbook for Managers and Designers*. Los Angeles: Melville Publishing Co., 1973.

Tinker, Lynne, comp. *An Annotated Bibliography of Library Automation 1968-1972*. London: Aslib, 1973.

"Tymshare." Brochure prepared by the Tymshare Information Services Division, 1976.

U.S. Civil Service Commission. Bureau of Training. ADP Management Training Center. "Management Introduction to Automated Data Bases." Washington, D.C., n.d. (Mimeographed.)

U.S. Congress. House. Committee on Government Operations. *Administration of Public Law 89-306, Procurement of ADP Resources by the Federal Government*. H. R. 1746, 94th Cong., 2d sess., 1976. Washington, D.C.: U.S. Government Printing Office, 1976.

U.S. Library of Congress. Procurement and Supply Division. "Mini Computer Systems; Request for Proposal." Washington, D.C., 1977. (Mimeographed.)

Veaner, Allen B. "Major Decision Points in Library Automation." *College and Research Libraries* 31 (September 1970). Cited by Richard Phillips Palmer, *Case Studies in Library Computer Systems*, p. 210. New York: R. R. Bowker, Xerox, 1973.

———. "Perspective: Review of 1968-1973 in Library Automation." In *Library Automation: The State of the Art II*, pp. 1-17. Edited by Susan K. Martin and Brett Butler. Papers presented at the Preconference Institute on Library Automation sponsored by the Information Science and Automation Division of the American Library Association at Las Vegas, Nevada, 22-23 June 1973. Chicago: American Library Association, 1975.

Vosatka, G. J. "The Minicomputer—Evolution or Revolution." In *Minicomputer Trends and Applications 1973; Symposium Record*, pp. 1–4. Papers presented at the IEEE Symposium held at National Bureau of Standards, Gaithersburg, Maryland, 4 April 1973. New York: Institute of Electrical and Electronics Engineers, 1973.

Waite, David P. "The Minicomputer: Its Role in a Nationwide Bibliographic and Information Network." In *Applications of Minicomputers to Library and Related Problems*, pp. 136–57. Edited by F. Wilfrid Lancaster. Papers presented at the 1974 Clinic on Library Applications of Data Processing, 28 April to 1 May 1974. Urbana-Champaign, Ill.: University of Illinois, Graduate School of Library Science, 1974.

Weisbrod, David L. "Acquisitions Systems: 1973 Applications Status." In *Library Automation: The State of the Art II*, pp. 87–100. Edited by Susan K. Martin and Brett Butler. Papers presented at the Preconference Institute on Library Automation sponsored by the Information Science and Automation Division of the American Library Association at Las Vegas, Nevada, 22–23 June 1973. Chicago: American Library Association, 1975.

Weitzman, Cay. "Micros, Minis and Midis; [Speech]" In *Minicomputers: The Applications Explosion*, pp. 675–729. Edited by David E. Debeau. Proceedings of the American Institute of Industrial Engineers Conference held in Washington, D.C., 17–19 November 1975. Los Angeles: Management Education Corporation, 1975.

——. *Minicomputer Systems; Structure, Implementation, and Application*. Englewood Cliffs, N.J.: Prentice-Hall, 1974.

Wieselman, Irving L. "Printer Technology and Its Future; A Printer Primer." *Modern Data* 8 (November 1975): 33–42.

Withington, Frederick G. "Beyond 1974: A Technological Forecast." *Datamation* 21 (January 1975): 54–73.

☆ U.S. GOVERNMENT PRINTING OFFICE : 1978 O—259-882