MICROSOFT SOFTWARE CATALOG

Table of Contents

Throughout the evolution of the microcomputer, Microsoft has been recognized as the leader in software. All Microsoft languages and system software products are backed by our experience and reputation. Since 1974, we have consistently provided reliable, proven software that meets computer industry standards.

Microsoft supplies software to manufacturers, retail outlets, and end-users. Whatever your system software requirements, you can rely on Microsoft for highest quality, state-of-the-art technology, and full customer support.

Our name means microcomputer software.

WE SET THE STANDARD.

BASIC

In 1975, Microsoft wrote the first BASIC interpreter for the 8080. Today, Microsoft BASIC, with over 300,000 installations, is recognized as the industry standard. It's the BASIC you'll find on all the most popular microcomputers, and many applications programs have been written in Microsoft BASIC.

The Microsoft family of BASICs includes the following configurations:

1. BASIC-80 Interpreter
   8K, Extended, Disk, and Standalone Disk versions are available.
2. BASIC-86 Interpreter
   Microsoft BASIC for the 8086. Extended and Disk versions.
3. BASIC Compiler
   A compiled BASIC with the language features of BASIC-80.
4. BASIC-68 and BASIC-69
   Microsoft BASIC for the 6800 and 6809 microprocessors.
5. 6502 BASIC
   A 9K BASIC interpreter for the 6502 microprocessor.

BASIC-80

BASIC-80 from Microsoft is the most extensive implementation of BASIC available for 8080 and Z80 microprocessors. In five years of use, it has become the world standard for microcomputer BASICs. That's because BASIC-80 gives users what they want from a BASIC—ease of use plus the features that make a micro perform like a minicomputer or large mainframe.

BASIC-80 is available in three upward compatible versions: 8K BASIC, Extended BASIC and Disk BASIC. Disk BASIC is compatible with the following disk operating systems (specify when ordering): CP/M,® ISIS-II, TEKDOS. A Standalone Disk version is also available.

Features

BASIC-80 meets the requirements for the ANSI subset standard for BASIC, and it supports many unique features rarely found in other BASICs.

- Four variable types: Integer, String, Single Precision Floating Point (7-digits) and Double Precision Floating Point (16-digits)
- Full PRINT USING for formatted output (includes asterisk fill, floating dollar sign, scientific notation, trailing sign, comma insertion)
- Trace facilities (TRON/TROFF)
- Error trapping using the ON ERROR GOTO statement
- Direct access to CPU I/O ports with INP and OUT
- Read or write any memory location using PEEK/POKE
- Extensive program editing facilities via EDIT command and edit mode subcommands
- Automatic line number generation and renumbering
- Call up to 10 assembly language subroutines
- Matrices with up to 255 dimensions
- IF/THEN/ELSE and nested IF/THEN/ELSE
- Boolean operators OR, AND, NOT, XOR, EQV, IMP
- Custom versions of BASIC-80 can be placed on ROM
- Disk BASIC-80 supports random and sequential disk files with a complete set of file manipulation statements: OPEN, CLOSE, GET, PUT, KILL, NAME, MERGE. (Note: TEKDOS does not support random files.)

Release 5.0

Here are some of the new features added to BASIC-80 in its fifth major release:

Long variable names—variable names may have up to 40 characters and may contain embedded reserved words.

Protected files—disk files may be saved in a coded binary format.

Dynamic string space allocation—no need to reserve extra string space.

WHILE/WEND statement—a new conditional statement that gives BASIC a more structured flavor.

CHAIN and COMMON statements—programs may be linked together and share common variables.

RANDOMIZE statement—reseeds the random number generator.

Variable length records—random file record length may be determined by the user.

6502 BASIC	
OEMs	Single Copies
Contact Microsoft for terms and prices	6502 BASIC is available to end users from: Johnson Computer PO Box 523 Medina, OH 44245 216-725-4560

BASIC-80 and BASIC-86

(Disk versions)
Commands–Statements–Functions

Commands

AUTO
LIST
NULL
TROFF
CLEAR
LOAD
RENUM
WIDTH
CONT
MERGE
RUN
DELETE
NAME
SAVE
EDIT
NEW
TRON

Program Statements

CALL
GOSUB
END
GOTO
STOP
WHILE/ WEND
CHAIN
DEF USR
LET
REM
RANDOMIZE
COMMON
DEF FN
ERROR
POKE
RESUME
SWAP
DEFDBL
DEFSTR
DEFSNG
DEFINT
RETURN
WAIT
ON GOSUB
DIM
FOR/NEXT/ STEP
IF/THEN/ ELSE
ON ERROR GOTO
OPTION BASE

Input/Output Statements and Functions

CLOSE
KILL
OUT
RESTORE
READ
TAB
DATA
LINE INPUT
PRINT
WRITE
GET
POS
FIELD
LSET/RSET
PRINT USING
LOC
MKI$
MKS$
MKD$
NAME
PUT
EOF
SPC
INKEY$
INPUT
OPEN
CVI
CVD
CVS

Arithmetic Functions

ABS
INT
SGN
ATN
EXP
SIN
CDBL
CSNG
CINT
SQR
LOG
FIX
COS
RND
TAN

String Functions

ASC
LEN
STRING$
CHR$
LEFT$
STR$
HEX$
OCT$
VAL
INSTR
RIGHT$
MID$
SPACE$

Operators

=
^
–>
>
–
/
*
<=
+
<>
/
>=
XOR
NOT
EQV
MOD
IMP
AND
OR

Special Functions

ERL
USR
ERR
FRE
VARPTR
PEEK

NOTE: Line Printer versions add LPRINT, LLIST, and LPOS.

BASIC-80 Reference Manual, purchased separately $20

BASIC-80

Version	System	Price	Supplied on
Disk	CP/M	$350	8" single density diskette
Disk	ISIS-II	$350	8" single or double density diskette (please specify)
Disk	TEKDOS	$350	8" single density diskette
Extended	MDS	$250	hex paper tape
Extended	SBC 80/10	$250	hex paper tape
Extended	SBC 80/20	$250	hex paper tape
8K	MDS	$150	hex paper tape
8K	SBC 80/10	$150	hex paper tape
8K	SBC 80/20	$150	hex paper tape

Hex paper tapes use standard Intel format.
Above prices include BASIC-80 Reference Manual.

BASIC 80

BASIC COMPILER

Microsoft's BASIC compiler is a powerful new tool for programming BASIC applications or microprocessor system software. The single-pass compiler produces extremely efficient, optimized 8080 machine code that is in Microsoft standard relocatable binary format. Execution speed is typically 3-10 times faster than Microsoft's BASIC-80 interpreter. The minimum amount of memory required to run the compiler with the CP/M® operating system is 40K.

Optimized, Compatible Object Code

The BASIC compiler produces object code that is highly optimized for speed and space, relocatable, and compatible with other Microsoft software products. The loader format is identical to that of Microsoft's COBOL-80 compiler, MACRO-80 assembler and FORTRAN-80 compiler, so programs written in any of these four languages can be loaded and linked together. The compiler also provides a formatted listing of the machine code that is generated.

Compiled programs are fast and compact due to extensive optimizations performed during compilation:

1. Expressions are reordered to minimize temporary storage and (wherever possible) to transform floating point division into multiplication.
2. Constant multiplications are distributed to allow more complete constant folding.
3. Constants are folded wherever possible. The expression reordering finds "hidden" constant operations.
4. Peephole optimizations are performed, including strength reduction.
5. The code generator is template-driven, allowing optimal sequences to be generated for the most commonly used operations.
6. String operations and garbage collection are extremely fast.

Compiled BASIC programs are the ideal end product for BASIC applications programmers. The machine code for any application program may be placed on a diskette, ROM or other media. The program not only runs faster than with the interpreter, but the BASIC source program need not be distributed. Thus the original application program is protected from unauthorized alteration.

Language Features

The Microsoft BASIC Compiler supports all the commercial language features of Microsoft BASIC-80, except those commands that are not usable in the compiler environment (i.e., direct mode commands such as LOAD, AUTO, SAVE, EDIT, etc.). That means you get all the BASIC language features for which Microsoft BASIC is so well known. In addition, the compiler supports double precision transcendental functions (SIN, COS, TAN, ATN, LOG, EXP, SQR).

BASIC Compiler for CP/M,® ISIS-II, and TRSDOS Model II, including MACRO-80 and LINK-80 and all documentation	$395.00
BASIC Compiler, documentation only	$20.00

BASIC COMPILER

BASIC 86
BASIC 68
BASIC 69

BASIC-86

BASIC-86 is an ANSI standard BASIC and contains the features of the latest release of BASIC-80. It is available in Extended and Standalone Disk versions. Both versions are available on single and double density diskettes with the Extended version additionally available on paper tape.

The Extended version is furnished in Intel Hex format and requires the user to have a load capability for this format. The Standalone Disk Version is accompanied by EPROMS for installation on the SBC 86/12 board to provide initialization and bootstrap from diskette.

Hardware requirements for the two versions are:

Extended

- Intel SBC 86/12 printed circuit card
- Backplane and power supply enclosure for the above (such as Intel System 80)
- Operators Console
- Ability to load Intel Hex formatted program

Standalone Disk

- Intel SBC 86/12 printed circuit card
- Additional 32K memory
- Intel single or double density diskette controller and channel printed circuit cards and diskette drives
- Backplane and power supply for the above
- Operators console

BASIC-86 Ext. for SBC 86/12	$350
BASIC-86 Disk for SBC 86/12	$600

BASIC-68
BASIC-69

Microsoft BASIC is also available in disk versions for the 6800 and 6809 microprocessors. Now it's possible for 6800 and 6809 users to take advantage of the large existing library of Microsoft BASIC programs. BASIC-68 and BASIC-69 are compatible with the FLEX operating system, and they support all the language features of Microsoft BASIC-80.

BASIC-68 (for FLEX)	$200
BASIC-69 (for FLEX)	$250

Custom or standalone implementations of BASIC-68 and BASIC-69 are available to OEMs only. Contact Microsoft for prices. Current OEM customers for Microsoft BASIC-68 include:

- Perkin-Elmer
- Ohio Nuclear
- Pertec
- Societe Occitane d'Electronique

COBOL-80

As the business world continues to find new uses for the power and economy of microcomputers, COBOL is more in demand than ever. Inventory, personnel, payroll, order entry, accounting, and forecasting are the applications for which COBOL was specifically designed. It gives programmers the tools they need to meet the rising challenge of information processing: powerful use of disk files, CRT screen handling, easy-to-use syntax, long variable names, and readable programs. Large programs are managed effectively with COBOL, because data and procedures can be hierarchically structured for efficient, clean, top-down design.

Microsoft's COBOL-80 is the most extensive implementation of COBOL for microcomputers. It's an ANSI standard COBOL that supports special features such as:

- Advanced verbs: STRING, UNSTRING, COMPUTE, SEARCH, PERFORM (VARYING/UNTIL)
- Interactive ACCEPT/DISPLAY
- Abbreviated and compound conditions
- Sequential, Relative and Indexed files
- Optional packed and binary data formats
- Runtime assignment of filenames
- Full COPY facility

Adaptable

OEMs: COBOL-80 is easily adapted to new operating environments. We can quickly meet your hardware and operating system requirements with customized versions of COBOL-80. It's the COBOL that's already been chosen by these and other manufacturers:

- Cromemco
- Pertec
- Ohio Scientific
- TEI
- Altos
- ICL
- Billings
- SST

Extensions to the ANSI Standard

In addition to meeting the 1974 ANSI standard for COBOL, Microsoft's COBOL-80 has many extra features that enhance its efficiency and ease-of-use:

- **CHAIN** for smooth transfer of control from one program to another
- **Interactive ACCEPT/DISPLAY** for powerful user/program interaction via the CRT terminal.
- **Line Sequential Files,** compatible with the text files generated by many system editors
- **Trace style debugging** using the READY TRACE and RESET TRACE commands to monitor program flow
- **COMP-3 data format** (optional) packs data two digits to the byte to greatly reduce mass storage requirements

COBOL-80 Runtime System

Compilation of COBOL source programs produces a carefully designed, compact object pseudo-code which is then interpreted by the runtime system. The runtime system is a library of relocatable modules that are loaded selectively, according to the code needed to run a given program. The runtime interpreter (written in 8080 machine code) actually performs the algorithmic functions required for execution of a program.

Memory Requirements

COBOL-80 requires about 40K of user memory, in addition to the operating system's space. COBOL object programs will run on systems as small as 32K.

Speed

COBOL-80 is very fast. The compilation rate is about 300 lines per minute for lines before the Procedure Division and 150 lines per minute in the Procedure Division.

Operating Environment

COBOL-80 runs on 8080/Z80/8085 microprocessors and is available in versions for the following operating systems:

- CP/M® Versions 1 and 2
- ISIS-II
- TRSDOS Model II
- Imsai IMDOS
- Cromemco CDOS*

*Contact Cromemco directly.

Summary of Features

The following table summarizes the features of Microsoft COBOL with respect to the ANSI-74 Standard.

Module	**Features Available in Microsoft COBOL**
Nucleus	All of level 1, plus these features of level 2: CONDITIONS: Level 88 conditions with value series or range Use of logicaı AND/OR/NOT in conditions Use of algebraic relational symbols (<, >, =) Implied subject, or both subject and relation, in relational conditions Sign test Nested IF statements; parentheses in conditions VERBS: Extensions to the functions of ACCEPT and DISPLAY for formatted screen handling ACCEPTance of data from DATE/DAY/TIME STRING and UNSTRING statements COMPUTE with multiple receiving fields PERFORM VARYING IDENITIFIERS: Mnemonic-names for ACCEPT or DISPLAY devices Procedure-names consisting of digits only Qualification of Names (in Procedure Division statements only)
Sequential, Relative and Indexed I/O	All of level 1 plus these features of level 2: RESERVE clause Multiple operands in OPEN & CLOSE, with individual options per file Value of FILE-ID is data-name
Sequential I/O	EXTEND mode for OPEN WRITE ADVANCING data name lines LINAGE phrase and AT END-OF-PAGE clause
Relative and Indexed I/O	DYNAMIC access mode (with READ NEXT) START (with key relationals EQUAL, GREATER, or NOT LESS)
Library	Level 1
Inter-Program Communication	Level 1
Table Handling	All of level 1 Full level 2 formats for SEARCH statement
Debugging	Special extensions to ANSI-74 Standard providing convenient trace-style debugging. Conditional Compilation: lines with "D in column 7" are bypassed unless "WITH DEBUGGING MODE" is given in SOURCE-COMPUTER paragraph.
Segmentation	Level 1

COBOL-80 system including macro assembler, linking loader, and documentation	$750.00
COBOL-80 documentation only	$ 20.00

FORTRAN-80

Microsoft's FORTRAN-80 brings the world's most popular science and engineering programming language to users of 8080/Z80 microcomputers. FORTRAN-80 is comparable to FORTRAN compilers on large mainframes and minicomputers. All of ANSI FORTRAN X3.9-1966 is included except the COMPLEX data type. Therefore, users may take advantage of the many applications programs already written in FORTRAN.

Library Features

FORTRAN-80 is supplied with an extensive library of single and double precision scientific functions.

The standard library routines included are:

ABS
INT
AMAXO
DMAX1
MIN1
SIGN
IDIM
DEXP
DLOG10
DCOS
ATAN
DMOD
OUT
IABS
IDINT
AMAX1
AMINO
DMIN1
ISIGN
SNGL
ALOG
SIN
TANH
DATAN
PEEK
DABS
AMOD
MAXO
AMIN1
FLOAT
DSIGN
DBLE
DLOG
DSIN
SQRT
ATAN2
POKE
AINT
MOD
MAX1
MINO
IFIX
DIM
EXP
ALOG10
COS
DSQRT
DATAN2
INP

The library also contains routines for 32-bit and 64-bit floating point addition, subtraction, multiplication, division, etc. These routines are among the fastest available for performing these functions on the 8080.

Enhancements

The FORTRAN-80 compiler has a number of enhancements to the ANSI Standard:

1. LOGICAL variables which can be used as integer quantities in the range +127 to −128.
2. LOGICAL DO loops for tighter, faster execution of small valued integer loops.
3. Mixed mode arithmetic.
4. Hexadecimal constants.
5. Literals and Holleriths allowed in expressions.
6. Logical operations on integer data. .AND., .OR., .NOT., .XOR. can be used for 16-bit or 8-bit Boolean operations.
7. READ/WRITE End of File or Error Condition transfer. END=n and ERR=n (where n is the statement number) can be included in READ or WRITE statements to transfer control to the specified statement on detection of an error or end of file condition.
8. ENCODE/DECODE for FORMAT operations to memory.
9. IMPLICIT statement changes default variable types.

FORTRAN-80 Compiler Characteristics

The FORTRAN-80 compiler can compile several hundred statements per minute in a single pass and needs less than 25K bytes of memory to compile most programs. Any extra available memory will be used by the compiler for extended optimizations.

In spite of its small size, the FORTRAN-80 compiler optimizes the generated object code in several ways:

1. Common subexpression elimination. Common subexpressions are evaluated once, and the value is substituted in later occurrences of the subexpression.
2. Peephole Optimization. Small sections of code are replaced by more compact, faster code in special cases. Example: I=I+1 uses an INX H instruction instead of a DAD.
3. Constant folding. Integer constant expressions are evaluated at compile time.
4. Branch Optimizations. The number of conditional jumps in arithmetic and logical IFs is minimized.

Long descriptive error messages are another feature of the compiler. For instance:

? Statement unrecognizable

is printed if the compiler scans a statement that is not an assignment or other FORTRAN statement. The last twenty characters scanned before the error is detected are also printed.

The compiler generates a fully symbolic listing of the machine language being generated. At the end of the listing, the compiler produces an error summary and tables showing the addresses assigned to labels, variables and constants.

Custom I/O Drivers

Users may write non-standard I/O drivers for each Logical Unit Number, making the task of interfacing non-standard devices to FORTRAN programs a straightforward one.

FORTRAN-80 is available for the CP/M,® ISIS-II, or TEKDOS operating system.

FORTRAN-80 system including macro assembler, linking loader, and documentation	$500.00
FORTRAN-80 documentation only	$ 20.00

MICRO-SEED DBMS

Many microcomputer applications that manipulate a lot of data are becoming limited by traditional floppy disk file handling. For any program that must access a large database, the efficient management of that data is the key to overall thruput.

A database management system is a general-purpose tool that permits users to work directly with the *content* rather than the *physical representation* of stored information.

- A DBMS makes it practical to develop and implement applications involving many pieces of interrelated data.
- It automatically performs the detailed housekeeping tasks associated with entering, extracting, modifying, and deleting records.
- The user can issue simple requests and directives by specifying contextual information rather than dealing in terms of memory locations, disk pages, array offsets, and similar computer idiosyncracies.

The effect is like the high-level language compiler that substitutes a few application oriented phrases and symbols for many step-by-step machine specific instructions.

Micro-SEED is the first CODASYL-compatible database management system for microcomputer systems. It is cost-effective for almost any microcomputer application and utilizes existing operating system resources. In accordance with CODASYL recommendations, Micro-SEED:

- incorporates a Data Definition Language that is used to specify a structural description of the database.
- allows the user's application program, written in a high level language, to create and update the data through the use of Micro-SEED's Data Manipulation Language statements.
- allows these statements to invoke routines in the database manager that perform the various management tasks.

For program development, Micro-SEED provides relief from the many data manipulation burdens that often slow the development of today's distributed processing applications. For OEMs, Micro-SEED can be packaged with a turnkey application system in which it is transparent to the user.

The initial version of Micro-SEED utilizes Microsoft FORTRAN (purchased separately) as the host language. BASIC and COBOL-80 will be added later as host languages. Micro-SEED also provides utility programs that initialize the database and display information about its content and structure. A 64K CP/M® system is recommended to run Micro-SEED.

Micro-SEED Database Management System including documentation	$900
Documentation only	$ 20

muMATH 79
muLISP 79

muMATH-79

The muMATH-79 symbolic math package represents a major advance for the use of microcomputers in scientific, educational and engineering applications. It provides the facilities to perform a wide variety of symbolic mathematical operations efficiently and accurately that were previously unavailable on a microcomputer. The hierarchical structure of the package, plus its interactive nature, also make it an excellent math teaching device at any level from arithmetic to calculus. The muSIMP language itself is also included in the package, allowing the user to develop additional applications. The muMath/muSIMP package runs under CP/M® and requires 40K of memory.

muMATH

The muMATH package is a set of programs written in muSIMP, an implementation of the SIMP language. The modular approach allows the user to load only the portions of the package needed for an application and so keep the maximum amount of space available for actual computations. muMATH acts as a calculator to perform sophisticated mathematical functions. The system prompts the user with a question mark, after which the user types an expression terminated by a semicolon or a dollar sign. Interactive lessons are provided as part of the muMATH software to make the system easier to learn.

Mathematical capabilities provided in the package include:

- Exact rational arithmetic (611 digits)
- Algebraic simplification of expressions using the operators +, −, *, / and ↑
- Treatment of equations as expressions which can be assigned, added, multiplied, squared, etc.
- Exact solution of an algebraic equation
- Matrix operations on arrays: transpose, multiply, divide, inverse and other integer powers.
- Logarithmic, exponential and trigonometric simplifications and transformations
- Symbolic differentiation: partial derivatives
- Symbolic integration of indefinite and definite integrals

muSIMP

muSIMP (microcomputer Structured IMPlementation language) is an extensible high level programming language highly suitable for symbolic and semi-numerical processing. It is implemented using an efficient and versatile interpreter requiring 7K bytes of machine code. Dynamic, transparent memory management gives muSIMP much of its inherent power. Control structures include LOOP/ENDLOOP, BLOCK/ENDBLOCK, WHEN, and EXIT. SUBROUTINE and FUNCTION are both provided for subprograms.

muMATH/muSIMP Package including muMATH and muSIMP on disk and documentation	$250.00
muMATH/muSIMP documentation only	$ 20.00

muLISP-79

muLISP-79 is the first extensive micro implementation of LISP, a programming language that is highly suitable for artificial intelligence applications. LISP is an applicative, recursive language which makes it ideal for describing complex mathematical. concepts. It provides a highly interactive environment for human-machine communication. The muLISP interpreter runs under CP/M® and resides in only 7K. A minimum system will run in as little as 16K total, including the resident operating system.

muLISP offers all of LISP's unique programming features including:

- 83 LISP functions defined in machine language
- Infinite precision integer arithmetic expressed in any desired radix base from 2 through 36, supported by a complete set of numerical primitives (611 digits)
- A two-pass garbage collector that performs automatic, dynamic memory management on all data spaces, allowing the computer to respond to queries of arbitrary difficulty
- Flexible program control structures including an extended COND, a multiple exit LOOP and a powerful function body evaluation mechanism
- User functions definable as either Call by Value or Call by Name and as either Spread or No-spread.
- Debugging facilitated by a TRACE package, informative diagnostics, and interruptable program execution
- Extremely fast program execution speeds achieved through such techniques as shallow variable binding, address typed data structures and a closed pointer universe.

The muLISP-79 Interpreter is distributed as the executable COMmand file MULISP79.COM. Also included are three muLISP LIBrary files:

UTILITY.LIB	An assortment of utility functions
TRACE.LIB	A function trace debugging package
METAMIND.LIB	A sample game program

muLISP-79 Package Includes muLISP-79 Interpreter, utility functions, debugging package, sample game and documentation	$200.00
muLISP-79 documentation only	$ 15.00

MACRO 80
XMACRO 86

MACRO-80 ASSEMBLER

MACRO-80 is a relocatable macro assembler for 8080 and Z80 microcomputer systems. It incorporates almost all "big computer" assembler features without sacrificing speed or memory space. The MACRO-80 package is comprised of the MACRO-80 assembler, a linking loader, and cross reference facility.

The assembler supports a complete, Intel standard macro facility, including IRP, IRPC, REPEAT, local variables and EXITM. Macro names take precedence over instruction mnemonics and pseudo operations. Nesting of macros is limited only by memory. Code is assembled in relocatable modules that are easily manipulated with the flexible linking loader. Conditional assembly capability is greatly enhanced by an expanded set of conditional pseudo operations that include testing of assembly pass, symbol definition, and parameters to macros.

More MACRO-80 features:

- Comment blocks
- Variable input radix from base 2 to base 16
- Octal or hex listings
- INCLUDE statement assembles an alternate source file into the current program.
- PRINTX statement for printing assembly or diagnostic messages
- PHASE/DEPHASE statements allow code to reside in one area of memory but execute in another
- Ability to accept both 8080 and Z80 opcodes, selectable with a pseudo-op or command switch
- Complete set of listing controls

MACRO-80 resides in approximately 14K of memory. Assembly rate is over 1000 lines per minute. MACRO-80 is compatible with the following operating systems (specify when ordering): CP/M,® ISIS-II, TEKDOS.

Linking Loader and Cross Reference Facility

With LINK-80, any number of programs may be loaded with one command, relocatable modules may be loaded in user-specified locations, and external references between modules are resolved automatically by the loader. The loader also performs library searches for system subroutines and generates a load map of memory showing the locations of the main program and subroutines.

CREF-80, the cross reference facility that is included with MACRO-80, supplies a convenient alphabetic list of all program variable names, along with the line numbers where they are referenced and defined.

MACRO-80 including MACRO-80, LINK-80, CREF-80, and documentation	$200.00
MACRO-80, documentation only	$ 15.00

XMACRO-86 CROSS ASSEMBLER

XMACRO-86 is an easy-to-use cross assembler with an assembly rate of over 1000 lines per minute. It assembles 8086 code on any 8080 or Z80 development system running the CP/M,® ISIS-II, or TEKDOS operating system. XMACRO-86 is a modified version of MACRO-80, Microsoft's 8080/Z80 macro assembler, and consequently contains the same major features, such as relocation, macros, conditional assembly, and listing and loader control. (XMACRO-86 does not assemble 8080 or Z80 instructions.)

Compare Microsoft XMACRO-86 with Intel ASM86

XMACRO-86 adheres to the more traditional form of an assembler that does not use symbol types, as compared to the Intel ASM86 which uses symbol types and generics to reduce the number of mnemonics. For example, ASM86 has only one MOV mnemonic for a class of operations including move byte, move word, and move immediate byte or word. XMACRO-86 uses the mnemonics MOV (for move word), MOVB, MOVI and MOVBI, thus making the distinction between instruction types at a level more familiar to most programmers.

Indexing and addressing methods are essentially the same in XMACRO-86 as in ASM86. However, the more traditional expression evaluation provided by XMACRO-86 lends itself to more straightforward programming.

For example, XMACRO-86 allows:

```
MOV      AX,BETA     ;Contents of word BETA to AX
MOVI     BX,BETA     ;Address of BETA to BX
MOVB     [BX],5      ;Contents of byte location 5
                     ;to location addressed by BX
MOVBI    3[BX],5     ;5 (Immediate) to third byte
                     ;after location addressed by BX
```

ASM86, in contrast, uses the single MOV mnemonic, and other information such as the type of BETA (word or byte) and operators such as BYTE, WORD, DWORD, TYPE, PTR, THIS, SEG and OFFSET to resolve the differences between the types of MOV operations. These extra operators are not needed in XMACRO-86. Otherwise, the operands used in XMACRO-86 are the same as Intel's ASM86.

XMACRO-86 including XMACRO-86 cross assembler, linking loader, cross reference facility and documentation	$300.00
XMACRO-86, documentation only	$ 15.00

EDIT 80

EDIT-80 Text Editor

EDIT-80 is a random access, line oriented text editor similar to those used on large computers like the DEC PDP-10 and IBM 360. It may be used on any 8080 or Z-80 microcomputer system running the CP/M® operating system. While it supports a full range of editing capabilities, EDIT-80 is still fast and easy to use. You will find it versatile enough to meet the most demanding text editing requirements.

Microsoft's MACRO-80 assembler and FORTRAN-80 compiler print listings and error messages with EDIT-80 line numbers, giving the user quick reference to source lines.

In addition to commands that insert, delete, print and replace lines of text, EDIT-80 offers the following features:

- **Alter Mode**
  Alter (or intraline) mode provides a full set of intraline subcommands to edit portions of individual lines. These subcommands give the user more extensive editing capabilities than those provided with the EDIT command in Microsoft BASIC.
- **Numbering**
  Use the Number command to renumber an entire file or just parts of a file. Handy when "making room" for an insertion or just organizing line numbers in a file.
- **Multiple-page Files**
  If desired EDIT-80 files may be divided into sections called "pages." Page divisions mean easy handling of large files (line numbers may be reused on different pages) or convenient markers for the logical subdivisions in a file.
- **Find and Substitute**
  Specified text is efficiently located or replaced with EDIT-80's global Find and Substitute commands.
- **File Parameters**
  EDIT-80 can be used to edit BASIC programs without EDIT-80 line numbers, and files may be output with or without line numbers.

Summary of EDIT-80 Commands

Alter.	Enters Alter mode
Begin.	Moves to the beginning of a page
Delete.	Deletes lines
Exit.	Writes text to disk and exits EDIT-80
Find.	Finds text
Insert.	Inserts lines
Kill.	Deletes page marks
List.	Prints lines at the line printer
Mark.	Inserts a page mark
Number.	Renumbers lines
Print.	Prints lines at the console
Quit.	Exits the editor without writing text to disk
Replace.	Replaces lines
Substitute. . . .	Finds and replaces text
Write.	Writes text to disk
Extend.	Allows extension of lines

FILCOM

The EDIT-80 package includes a file compare utility called FILCOM. FILCOM compares two files and outputs differences between them. Source files or binary files may be compared using FILCOM.

EDIT-80 (includes EDIT-80 Text Editor and FILCOM File Compare Utility supplied on single density, 8" CP/M® diskette, with manual)	$120.00
EDIT-80, manual only	$ 10.00

Prices shown in this catalog are single-copy prices, USA domestic only. Single-copy buyers and dealers must sign a nondisclosure agreement for each Microsoft product purchased. VISA, Mastercharge and COD orders are accepted.

Back-up copies and updates for each product are made available at a nominal cost.

OEMs and Dealers:
Contact Microsoft for terms and prices.

CP/M® is a registered trademark of Digital Research.

MICROSOFT
10800 N.E. Eighth, Suite 819
Bellevue, WA 98004
(206) 455-8080 Telex 328945

VECTOR MICROSOFT
Research Park
B3030 Leuven
Belgium
(016) 20 24 96 Telex 26202 VECTOR

ASCII MICROSOFT
102 Plesarda
3-16-14 Minami Aoyama
Minato-ku
Tokyo 107, Japan
03-403-2120 Telex 2426875 ASCIIJ