Microcomputers

J. J. BRENNAN

Department of Management
The University of Texas at Austin
Austin, Texas 78712

MICHAEL K. MOLLOY

Department of Computer Science
The University of Texas at Austin
Austin, Texas 78712

Before publishing some articles on the application of microcomputers in management science, it seems appropriate to introduce our readers to the different elements that make up a microcomputer and to briefly discuss some of the more popular brands. Jay Brennan and Michael Molloy have written an excellent primer in microcomputers, comparing hardware capabilities and languages. Finally, some examples of management science software are described. A point must be borne in mind: The pace of technological advance is so fast that in two years this article will be completely out of date.

Paul N. Belshaw

The precise number of microcomputers marketed and sold to date is unknown. However, since their advent in the mid 1970's, the combined annual sales by all manufacturers is nearing or exceeds one million units.

Without question, microcomputers have been widely accepted in a startlingly short period of time because of their low unit cost, low for both systems and software. This does not mean that microcomputers are currently a cost-effective way to replace existing machines, although they may be in some specific situations. Rather, their cost constitutes a low threshold, the point at which owning or using a computer becomes viable for an individual. This threshold is important to both the business and personal user and has changed workstation computation from concept to reality.

Among the other factors that have contributed to the acceptance of microcomputers are:

— *Flexibility.* Microcomputers are inexpensive enough to warrant matching computer to application rather than the other way around. Upgrading to new technologies as they appear is also affordable because of the low initial system cost.

— *Competition.* Microcomputers have

0092-2102/83/1301/0028$01.25

INTERFACES **13**: 1 February 1983 (pp. 28-39)

been marketed mainly by smaller firms. Competition has been keen, and technical developments have been fast-paced because users demand real, not promised, benefits. This situation should continue as the entry of the computer giants and overseas manufacturers superheats the steam of competition.

— *Reliability.* Manufacturers cannot profitably service low cost products. Their only alternative is high reliability, which they have, quite convincingly, delivered.

Microcomputers are inexpensive enough to warrant matching computer to application rather than the other way around.

Besides the already wide market acceptance of microcomputers, what other evidence is there that these machines will play an important role in decision-making? Experienced management scientists hold to the canon that managers will use only models they understand and computational methods they trust. For 30 years, management science has been presented to management professionals who, although they have had a computer at work, did not work at the computer. Management skepticism regarding computers in general, and microcomputers in particular, is a hurdle which has not been completely cleared. However, in a climate of slumping productivity, the promise of increasing the quality and quantity of work performed by managers by using low unit-cost computers is a possibility that cannot be long overlooked. Almost unavoidably, some form of workstation computation, most probably a microcomputer, will play a key role in management decision-making in the near future.

Not to be overlooked is the way autonomous computation matches management style. The stand-alone computer frees the user from slavish reliance on the local computer-services personnel and actually forces the decision-maker to shape his analysis using his intuition. Given a proper collection of software tools, the user is free to concentrate his creative energies in ways that suit his tastes and the task at hand, all the while avoiding public acknowledgement of wrong turns and blunders, those inescapable companions of creative work. Successes can, as always, be trumpeted.

Because microcomputers have the potential to engender interest and nurture confidence in scientific decision-making, both managers and management scientists should have some knowledge of their functions and capabilities.

We offer here a compact primer on the microcomputer emphasizing information relevant to using or designing management science methods in this environment. First, we trace the evolution of microcomputers, describe some of their limitations, and briefly compare a selection of systems that are currently available. A discussion of operating systems and high-level languages available on these machines follows, and the scant literature on management science applications is surveyed. Finally we discuss the role of computer networks.

Evolution and Background

Microcomputer systems possess the same components as their more familiar mainframe relatives: a central processing unit (CPU), random access memory (RAM), auxillary storage (tape or disk), and input/output devices (keyboards, terminals, printers, and so forth). Microcomputers are small mainly because of the decrease in size of the CPU and RAM, accomplished by using large-scale and very large-scale integration (LSI and VLSI) manufacturing techniques. (These techniques have allowed the entire CPU to be placed on a single silicon chip less than one square centimeter in area.)

The development of microprocessors was sparked by the high cost of specialized sequential digital circuitry. The manufacturing cost of a circuit board is a function of the number of components (chips) on the board. By replacing a circuit board that has a large number of chips with a single chip microprocessor, the same functions can be performed at less cost. The Intel 4004, 8008, and 8080 microprocessors were originally developed to reduce the cost of computer terminals from $2000-$3000 to their current level of $600-$700. The market quickly expanded with the introduction of process control and communication subsystems built around microprocessors.

The problem of designing systems with microprocessors instead of with digital circuitry is one of writing software instead of laying out wires. The first microcomputer systems were constructed with microprocessors in order to develop software for other applications, for example, process control.

The size of the market remained very small (less than 10,000 units) until two companies, Apple and Tandy, began producing personal, self-contained, and fully-assembled microcomputers. Within a short time, manufacturers had developed the expertise, and users the interest, to change the focus of personal microcomputers from game-playing to decision and information support.

Management skepticism regarding computers in general, and microcomputers in particular, is a hurdle which has not been completely cleared.

Software has become the bottleneck in the fielding of microcomputers so that the state-of-the-art hardware developments are not immediately observable in the marketplace. Microcomputers in use at this moment are already two generations behind the available hardware. Since the original microprocessors were developed with different requirements than general purpose computers, the original architectures were not suited for personal computers. The newer architectures attempt to resolve this quandry.

There are several features of the microprocessor CPU which directly affect the performance and usability of the entire microcomputer system.

Address Space: The maximum number of memory locations that can be accessed, the address space, places a limit on the size of programs that can be run on the microcomputer.

Word Length: The largest data item that

the CPU can manipulate at one time is defined by the word length.

Data Path Width: For every cycle of a microprocessor 100-500 billionths of a second (nanoseconds), only a single transfer from memory can occur. If the data path is narrow (8 bits), less information can be transferred in each clock cycle.

System and User Programs: Since every system must have two programs coresident to be useful (the operating system and the application program), facilities in the microprocessor to speed the switch from executing one program to executing another program (context switching) are important. In multiuser systems, where context switching is frequent (10-100 times a second), the speed of switching has even more impact.

Hardware Multiply and Divide: Without hardwired multiply and divide instructions, normal calculations must be performed in software slowing the operation by as much as 10 times.

The original 8080 from Intel had an address space of 65336 (64K) bytes, a data path of 8 bits (a byte), and limited addressing modes. Compared to the 8080, Zilog extended the addressing modes, the instruction set, and decreased the power requirements in their Z80 microprocessor. The Motorola 6800 was also moderately superior to the 8080. However, these minor changes were not significant enough to drive the earlier 8080 out of the marketplace. In fact, most of the software written for Z80 systems is actually written in 8080 code [Bursky 1978].

Now a series of microprocessors with 1-16 megabytes (millions of bytes) of address space and 16 bit data paths are on the market: Intel has been marketing the 8086, Zilog the Z8000, and Motorola the 68000. These new microprocessors represent a significant improvement in microprocessor architecture. They incorporate such sophisticated techniques as instruction prefetch (8086), segmented addressing (Z8000, 8086), dual stack registers (Z8000, 68000) for fast context switching, and hardware multiply and divide (8086, Z8000, 68000). In many cases, the newer microprocessors are superior in performance to some minicomputers, which have been able to share or expand what was originally the mainframe market.

The microprocessor of the near future will pack additional mainframe capabilities on a single chip. Already announced and soon to be released are the first microprocessors with 32 bit data paths and greater than 16 megabytes of address space such as the Intel 432. These new microprocessors have the features (although not quite the speed) of such machines as the DEC VAX 11/780 and the IBM 370. Significantly, new vendors have developed some of these microprocessors. The Bell System has developed the BELLMAC-32 machine to run their UNIX operating system [Berenbaum 1982]. IBM has announced that the CPU section of a model 370 has been placed on a chip.

Current Systems

Limitations on space preclude presenting anything but a superficial comparison of a limited number of the currently available microcomputer systems. As the tables show, microcomputer systems span a fairly wide price range, and consequently differ widely in computational power. We have compared systems with others of

Make	Model	Retail Price	Data Path	CPU	Memory Size	Disk Storage	MPY DIV	Operating System
LOW End Systems								
IBM	5150	$3045	8	8088	256K	320K	Y	PCDOS (CP/M 86)
Xerox	820	$2995	8	Z80	64K	320K	N	CP/M
Apple	II+	$2375	8	6502	48K	140K	N	Apple DOS
Tandy	III	$2495	8	Z80	48K	320K	N	TRS-80DOS (CP/M)
Zenith	Z89	$2895	8	Z80	48K	100K	N	HDOS (CP/M)
MEDIUM Range Systems								
Dynatec	DB8/1	$3395	8	Z80	64K	180K	N	CP/M
Cromemco	System 2	$3990	8	Z80	64K	350K	N	CDOS (CP/M)
Northstar	Horizon	$4195	8	Z80	64K	360K	Opt	N.S. DOS (CP/M)
California Computer Systems	300-1A	$4849	8	Z80	64K	1.2M	N	CP/M
HIGH End Systems								
Onyx	8002	$14900	16	Z8002	256K	10M	Y	Unix
Apollo		$38000	16	68000	512K	33M	Y	DOMAIN
WICAT	150-WS	$ 8495	16	68000	256K	10M	Y	Unix
Codata	CTS300/6	$20000	16	68000	256K	6.3M	Y	Xenix
3 Rivers	PERQ	$12000	16	68000	256K	12M	Y	PERQ/OS

Table. 1. The features of some microcomputers in the low, medium, and high price ranges. The retail price given is for a typical configuration. MPY/DIV refers to hardware multiply/divide.

comparable cost, so that competing systems have been grouped into three ranges, LOW, MEDIUM, and HIGH, and their features are displayed in Table 1.

The LOW end units maintain their affordable cost (less than $4000) by large volume production and compact design. This forces their designers to allot only minimal space for expansion. Low end units use floppy disks, an 8-bit data path, and have only marginal flexibility. For the most part they can support only standalone applications.

The MEDIUM range units ($4000-$8000) offer a substantial increase in flexibility. Typically, they are based on a standard (although not optimal) bus design, the S-100 bus. (A bus is the parallel connection for multiple components.) This allows the user a wider choice of sources for components. The systems in this category still use an 8-bit data path and floppy disks, but some allow expansion to a multiuser operating system.

The HIGH end systems are noticeably different from the rest in both processing power and price ($9,000-$50,000), and already are becoming the workstation component of computer networks. They are designed with high flexibility and incorporate a 16-bit data path and faster (Winchester) disk drives. They can have as much as 1 megabyte of main storage and 20 megabytes of fast disk storage.

The foregoing descriptions are guidelines — they depict a typical system

within each category based on Summer 1982 data.

System Software

In order to effectively use a microcomputer system various programs must be available, the system software that supports the functions required by a normal user. They are typically supplied by the microcomputer manufacturer. The various programs can be partitioned into two types, operating systems software and programming language support software.

The problem of designing systems with microprocessors instead of with digital circuitry is one of writing software instead of laying out wires.

Operating Systems: The operating system of a computer is designed to free the user from the details of managing disk input and output (I/O) and to perform other system-oriented tasks. The three elementary components of an operating system are the kernel, the I/O drivers, and the support utilities.

Each component of an operating system provides a different function for the user. The kernel provides the interface between the operating system and the user, and it handles requests for I/O and file manipulation made by an applications program. The I/O drivers are designed to mate with the specific physical I/O devices of the system. Both the kernel and I/O drivers are memory resident. Support utilities perform such functions as copying and listing files, reporting system and file status, and supporting program debugging. They are loaded into memory as needed to minimize memory requirements.

Although there are many microcomputer operating systems in existence, they are easily separated into two classes. First is the single user operating system, such as CP/M [Kildall 1981], Northstar DOS, Cromemco CDOS, Zenith HDOS, Apple DOS, and TRS-80 DOS, which can only interact with one user and one program at a time. Second are the multiuser operating systems such as MP/M, Oasis, UNIX, XENIX, and DOMAIN, which can logically handle more than one user and more than one application program at a time. These systems require much more memory (one bank for each user in the case of MP/M and Oasis), faster disk drives (required for swapping programs in UNIX and XENIX), and a real time clock to determine the amount of time to allocate to each user.

One critical point to make about any operating system is the availability of software designed to run with that operating system. Good software written to use a particular operating system is not easily transferred to a different system. Since many successful application programs were originally written for CP/M, now most of the low and medium range systems offer a version of CP/M configured for their system. This has made CP/M the de facto standard for single user systems.

Because the CP/M operating system has many flaws, the new generations of operating systems, typically multiuser systems, have evolved along different lines. The popular UNIX operating system, designed by the Bell Telephone System, shaped a great deal of the new

operating system design. The UNIX operating system itself (and its clones such as XENIX) is available for the 16-bit microcomputers. Now, most new operating systems incorporate such features as lockable, tree-structured file systems; powerful command line interpreters; and flexible I/O routing (called pipes in UNIX).

In the past, the delay between the construction of computer systems and their viable implementation for the end user was significant. The first 8080 microcomputer was available in 1974, yet the wealth of software needed to make that system useful did not appear until 1978-1979. Now, because operating systems such as CP/M (which is written in PL/M) and UNIX (which is written in C) are more portable than assembly language operating systems, the delay from the time a new machine is announced until the operating system is ready is less than one year.

Programming Languages: The programming languages available on a microcomputer are of particular interest to the user. In the near term, programs will either have to be constructed from scratch or existing software running on other machines will have to be modified. In both cases, it is important to know the limits of the languages and operating systems available.

The typical microcomputer in the low to medium range has as intrinsic software, an assembly language (not always available directly to the user) and an interpreted BASIC. The other major high level languages in use, FORTRAN, PL/I, PASCAL, and C, are usually upgrades on the lower range machines, and at least one is usually standard on the higher range machines. Also typical is that the programming software is supported by a third party vendor under contract to the manufacturer.

BASIC is by far the most prevalent high-level language in use on microcomputers today. BASIC is usually implemented as an interpreted language. An interpreter simply translates and executes one line of a program language at a time, never generating a copy of the translation.

Software has become the bottleneck in the fielding of microcomputers . . .

Since an interpreter does not typically need as much memory as a compiler, BASIC was quickly adopted by microcomputer manufacturers.

Although BASIC is standard, it is far from standardized. The trend has been for each manufacturer to adapt the language as originally defined at Dartmouth in the 1960's to utilize the specific features of their graphics software and operating system. BASIC programs, therefore, are far from portable.

Unfortunately, interpreted BASIC is extremely slow, slow enough in fact that one hour of computation time on some microcomputers is roughly equivalent to one second of computation on a large, modern mainframe (using FORTRAN) [Gillbreath 1981]. Interpreted BASIC defines the worst case with respect to execution time on microcomputers, although it must be kept in mind that many non-scientific applications do not require fast

computation. Compiled BASIC executes roughly ten times faster than its interpreted counterpart.

Modular program design, one of the main precepts of good software engineering, is difficult, even dangerous, in BASIC. This is a serious problem to the designer of large programs. In BASIC, variables have global definitions so that the programmer cannot truly isolate program segments. Another disturbing idiosyncracy of the language is that most BASIC implementations identify variables by their first two characters (BASIC-E is an exception). Thus NODE, NOPE, and NOTE would be the same variable.

BASIC is easy to learn but it is not a language built for ambitious program development. The two character name convention, no parameters in subroutine calls, and global variable definitions invite disaster when program modules are independently developed and subsequently concatenated for execution.

FORTRAN: On microcomputers FORTRAN probably ranks third in frequency of use after BASIC and PASCAL. FORTRAN is somewhat low in the language pecking order because most applications to date require more data manipulation than numerical computation, at which FORTRAN is nonpareil. FORTRAN is also plagued by a lack of standardization. Even though the core constructs are standard from 1966, the usual extensions are not. Perhaps the adoption of the 1977 FORTRAN standard, already available on some microcomputers, will help alleviate this problem.

This language has always suffered from a lack of sophistication in input/output and string manipulations. The ability to easily perform both of these operations is important in a primarily "hands-on" environment.

PASCAL: This ALGOL derivative has gained wide currency in academia during the past five to ten years, and for good reason. Its top down approach to programming either instills or enforces good programming habits (depending on one's point of view), and its philosophy of detecting as many programming errors as possible during compilation rather than during execution will be appreciated by anyone using this language. The normal PASCAL implementation is a hybrid, requiring both compilation and interpretation to execute. The source PASCAL program is first compiled into P-code (a pseudo machine code) which is then interpreted during execution. Consequently, it executes somewhat slower than FORTRAN, but only the P-code interpretation is machine dependent, making the compiler as portable as possible. PASCAL is available for the microcomputers in all three ranges.

PASCAL requires that any reference to a variable or procedure at some point in a program be preceded by a complete variable or procedure definition. Each variable is declared by name and by type (implementing another precept of good software engineering), with the types being either predefined or user-defined. The prudent selection of rich and powerful variable types is a great aid to program debugging. PASCAL has many other useful conventions and constructs. For more detailed information see [Grogono 1980]. PASCAL handles both I/O and string manipulation

adequately.

Two recognized deficiencies of PASCAL are: 1) the inability to dynamically dimension arrays, and 2) type conflicts on procedure calls. Unfortunately, there is no way around the former difficulty, and each PASCAL implementation varies in its strictures regarding type conflicts.

Nonetheless, the move to standardize the language with UCSD PASCAL (a version of PASCAL developed at the University of California, San Diego) makes this one of the most portable high level languages available. The lack of portability in high level languages has always been a problem to software designers, and the issue of portability is an important one in the microcomputer environment. USCD PASCAL is also an operating system so its use precludes the concurrent use of operating systems like CP/M.

Application Software

Any discussion of the current availability of microcomputer software for management science applications is necessarily brief. Most of the hundreds of applications programs available for microcomputers fall into one of two categories: support programs and business applications.

Support programs, primarily of interest to the software builder, perform useful tasks not available, or available to only a limited degree, with the system software, for example, line editing. Many such programs were designed by users and are shared on a friendly basis. On the other hand, small-business applications software constitutes the bulk of commercially available software. The phenomenal success of VISICALC, a matrix-based planning package, and the acceptance of microcomputers as small business accounting tools, have re-targeted commercial software to high-volume business applications. This is in spite of the initial belief that microcomputers would live or die in the household market. This emphasis on developing software to support high-volume applications partially explains why only a modest quantity of modelling software has appeared.

There is, however, a variety of statistical and mathematical software available, primarily for LOW and MEDIUM range units. Multiple regression, ANOVA, matrix inversion, and assorted plotting packages are available to support data analysis. The HIGH range machines are so new that, not surprisingly, software for them is quite limited. Software of wider applicability is not far off, however. We know of one manufacturer currently negotiating with the vendor of an advanced statistical package. The result should be a work-station computer with the statistical analysis capabilities normally to be found on a university's primary computer system.

In the area of management science applications, two ambitious efforts have been made, both for the Apple II computer. P. Jensen [1982] has developed software to solve capacitated transshipment problems. In addition, the software has been extended to solve generalized network problems, or network problems with convex quadratic costs. He has also incorporated a dynamic programming solver in the menu of user options, which includes programs to generate, store, and print problem data files. All programs are

written in BASIC.

Richard Duff [1981] has developed a PASCAL implementation to solve both standard capacitated networks and capacitated networks with elastic constraints. (Elastic constraints are those which can be violated by incurring a linear penalty in the amount of the violation). This software has a sophisticated editor and solves networks with nonlinear, convex, and separable costs.

Other traditional applications software is available mainly for either the TRS-80 or Apple II computers. Packages exist or are near completion for PERT/CPM, simple inventory modelling, MRP systems, and decision-tree analysis. More than one linear programming code has already been written and one commercial code is available for the TRS-80. Linear programming run on LOW price range micros is probably a mismatch, however, except for pedagogical purposes. The maximum problem size, even with sparsity, is limited to a few hundred variables, and numerical accuracy is difficult to achieve. What is important to understand is that the HIGH range microcomputers can handle the computations for many, if not most, current management science applications, including "in-core," large-scale linear or mixed integer programming.

The first 8080 microcomputer was available in 1974, yet the wealth of software needed to make that system useful did not appear until 1978-1979.

The use of microcomputers for model solution is gaining momentum, although few successes have been reported to date. One reported application involved the transfer of the MIT Service Planning Model to the Apple computer. This model evaluates (but does not optimize) cost and service performance for rail networks. Originally, the model was run on a VAX 11/780 computer, but was subsequently transferred to an Apple II with PASCAL. The microcomputer version has been used by the Boston and Maine Railroad with success. As Carl Martland and Carl Van Dyke [1981] report, although the solution time on the Apple is 40 times that of the VAX, overnight runs make the effective increase unnoticeable when compared to the former time-sharing environment.

One other computer application deserves mention: an advanced data base management system exists for micros. Although it requires a substantial investment relative to the price of the microcomputer system itself, Micro Data Base System's data base manager rivals, in capability, those available for mainframes.

Networking

One major application of microcomputers is in implementing the workstation concept. In this environment, some computation may be performed on the desktop unit but a communication network provides access to more powerful computers and other users on similar devices.

Computer networks can be divided into two basic types depending upon their geographical size. Long haul networks (or global networks) span distances in excess of two kilometers and usually require the use of established communication equip-

ment such as telephone lines, microwave relays, or satellites. Local area networks (LAN) span distances of less than two kilometers and typically utilize privately-owned mediums such as coaxial cable, private branch exchange (PBX) telephone lines, or direct wire connections.

The long haul networks have been evolving since the ARPA network was installed in the early 1970s. Communication between computers may be through either a physical connection or a message-based logical connection. A physical connection may use a direct wire or be switched like a telephone connection. In message-based systems, permanent lines are only connected between certain message processors to form a type of relay network. In order to communicate, a physical connection to one of the message processors is used and a message, along with its destination address, is given to the processor. The message will be forwarded until it reaches its destination. Such networks are termed store and forward (S/F) networks or packet-switched networks (PS) if messages are broken up into smaller pieces.

Networks which utilize the telephone facilities are used extensively for timesharing access to large host computers. Microcomputer users access such services by running a program that makes the computer act like a simple terminal. Some sophistication can be added to the program to allow copies of the terminal session to be stored on disk or previously stored files to be sent to the host computer. There are already several companies marketing information or computation services for microcomputer users. Such companies as COMPUSERV, MICRONET, The SOURCE, and others are trying to get microcomputer users to access their extensive databases and software [Greenhouse 1981].

Message based systems are being used for electronic mail and time sharing access. Such companies as GTE's TELENET, TYMENET, and GRAPHNET are offering electronic mail services where messages are sent from one computer to another. The basic timesharing services companies are also offering an electronic mail service for their users by storing messages on the system until the receiving user logs in. Although the use of such long haul networks is still increasing, the major share of communications takes place within corporations.

Local area networks are intended for the interconnection of a large number of devices in a relatively small area. A local area network would connect several computers and many workstations within a single building, for example, a corporate headquarters. These networks are just now evolving from the original direct wire connection techniques used to attach many terminals to a single host computer. There are two basic access control schemes proposed for local area networks. First are the contention algorithms, such as carrier sense multiple access with collision detection (CSMA/CD) used in broadcast bus networks such as the Xerox's Ethernet, Zilog's Z-net, and Ungerman-Bass' Net-One. Second are the token passing schemes, which are typically used in ring architectures such as IBM, Appolo's Domain, and Protean Associates' Pronet systems.

The contention networks allow any device to access the network at any time (multiple access) except when some other device is using the network (carrier sense). Unfortunately, when two or more devices start to transmit they are unaware of each other and their transmissions garble each other (collision). These networks have many advantages such as simplicity, expandability, low vulnerability, and low delays in lightly loaded environments. However, the potential collisions waste some of the network's capacity and create additional delays. In heavily loaded environments, this problem becomes critical.

The token passing systems allow access by the user who possesses the token. This technique avoids collisions but requires a user to wait for the token to be passed even when the network is idle. The details of the token and its passing make such networks more complicated to construct. However, since the network access is deterministic, the performance of the network is very good in heavily loaded environments.

Conclusion

We have presented a brief overview of microcomputers along with their development, capabilities, and potential use as a tool of the practicing management scientist. Already microcomputers have proved to be effective in the small-business environment, and an increasing number of large corporations and institutions are looking to the microcomputer to increase managerial productivity.

If there is one point to emphasize, it is this: the advent of newer and more powerful microcomputers vastly increases the size and complexity of models that can be analyzed quickly and cheaply by the workstation user. These microcomputers, and the attendant computer networks, bring the entire range of thirty years of model development closer to the point of impact than ever before. The opportunity to enhance the role of scientific decision-making in management has never seemed brighter.

References

Berenbaum, A., Condry, M. and Lu, P. 1982, "The operating system and language support features of the BELLMAC-32," *Proceedings of the Symposium on Architectural Support for Programming Languages and Operating Systems*, Palo Alto, March 1-3, pp. 30-38.

Bursky, D. 1978, "Microcomputer data manual: appendix," *Electronic Design*, Vol. 26, No. 11 (May), pp. 208-226.

Duff, R. H. 1981, "A microcomputer-based network optimization package," Masters Thesis, Naval Postgraduate School, Monterey.

Elphick, M., Ed. 1978, "Microprocessor selection guide," *Electronic Design*, Vol. 26, No. 21 (October), pp. 53-217.

Gillbreath, J. 1981, "A high level language benchmark," *Byte* (September), pp. 60-64.

Graube, M. 1982, "Local area NETS: a pair of standards," *Spectrum* IEEE (June), pp. 60-64.

Greenhouse, L. R. 1981, "Communications invade the home front," *Data Communications* (February), pp. 56-59.

Grogono, P. 1980, *Programming in PASCAL*, Addison-Wesley, Reading, Mass.

Jensen, Paul 1982, Private communications.

Kildall, G. 1981, "CP/M: A family of 8 and 16-bit operating systems," *Byte* (June), pp. 216.

Martland, C. and Van Dyke C. D. 1981, "Rail operations/service planning with a microcomputer: the Apple version of the M.I.T. service planning model," M.I.T. Department of Civil Engineering Working Paper, Cambridge, Mass.

Raunch-Hindin, W. 1981, "Which technology will rule the automated office?" *Data Communications* (November), pp. 66-79.