Application of Microcomputers in Bridge Design

RONALD A. LOVE, FURMAN W. BARTON, and WALLACE T. McKEEL, Jr.

ABSTRACT

The use of microcomputers in bridge design activities in state transportation departments was evaluated through contacts with 32 state agencies. Although present use of microcomputers was found to be limited, subsequent research showed that the current generation of 16-bit machines offers significant advantages in complementing existing computing facilities in a manner that fully uses the power of both mainframe and microcomputer. The ability of microcomputers to run large bridge design applications in a stand-alone mode was demonstrated by successfully downloading and converting four mainframe programs. Running design and analysis programs in a stand-alone mode frees the mainframe CPU and increases access to software that can be run repetitively without consideration of mainframe costs. When access to larger applications on the mainframe is required, the microcomputer used as an intelligent terminal can process input data locally and send them to the mainframe for processing. Output data, in return, can be downloaded to the microcomputer and reviewed off-line or input into microcomputer applications such as spreadsheets or graphics packages for further processing.

Computer applications in engineering design have had a dramatic effect on the analysis and design process in general. Automating analysis and design procedures has relegated much of the computational burden to machines, allowing the engineer more time to evaluate alternatives and assume a more creative design and decision-making role. Although the role computers play may vary from one organization to another, their effect has been revolutionary.

The manner in which computers are utilized in the design divisions of state departments of transportation is not standardized and varies greatly. Most of the software developed for design calculations within bridge divisions was designed for implementation on large mainframe computers. Bridge designers, in large measure, have access to these programs via terminals, and this has created little demand for other computer configurations such as microcomputers. However, recent developments in microcomputer design have resulted in microcomputers that have stand-alone capabilities that rival those of minicomputers and mainframes and that also possess versatile communications capability.

There still appears to be considerable difference of opinion about the most appropriate role for microcomputers in bridge design applications. Many bridge divisions, which have their own large computer and several terminals, find their present configuration satisfactory and see no reason to incur the additional expense of microcomputers. Other bridge engineers, however, are required to use centralized state computer facilities that are sometimes shared by other state agencies. The inconvenience of gaining access, the high cost of computing and other charges, and excessive turnaround time may not be acceptable. These engineers see the new generation of microcomputers as a cost-effective and preferred alternative for using much of the bridge design software available. In Virginia, as in many other states, many bridge design activities have been decentralized in district offices across the state. The present generation of microcomputers would appear to meet most of the computational needs of these offices. These smaller computers could supplement the mainframe, possibly using downloaded, smaller programs, in a more efficient mode of operation. The many advantages of microcomputers, such as powerful computing capability, stand-alone capability, communications capability, and cost-effectiveness, make them a powerful element in engineering computation.

It is useful and timely to evaluate the manner in which microcomputers are used in other states and to suggest the role that they may play in the future. Such information could assist bridge engineers and administrators in state departments of transportation in making decisions about the use of microcomputers.

The objective of this study was to examine the current and future role of microcomputers in bridge design applications within state departments of transportation. The focus was on the use of microcomputers, as a complement to present computing configurations, to increase productivity and enhance cost-effectiveness.

The manner in which bridge engineers currently use computers for design and analysis was evaluated by contacting a number of state and federal agencies including bridge divisions in several states. These bridge divisions were surveyed to determine their present computer configurations used for bridge design applications and their current and projected uses of microcomputers. The capabilities of the present generation of 16-bit microcomputers were evaluated for bridge design applications, several microcomputers were used to run typical bridge design software, and comparisons of performance were noted. The feasibility of converting current bridge design software from mainframes to microcomputers was evaluated through actual conversions of existing software. After examination and study of the information collected and the tests performed, the potential for increased usage of microcomputers in bridge design activities was evaluated.

R.A. Love and F.W. Barton, Department of Civil Engineering, University of Virginia, Charlottesville, Va. 22903. W.T. McKeel, Jr., Virginia Highway and Transportation Research Council, University of Virginia, Charlottesville, Va. 22903. Current address for R.A. Love: G.W. Beilfuss and Associates, Inc., 700 East Butterfield Road, Suite 220, Lombard, Ill. 60148.

MICROCMPUTER USE IN STATE BRIDGE DIVISIONS

To determine trends in microcomputer use, an informal telephone survey of bridge divisions in various state departments of transportation was conducted, and a total of 32 states were contacted:

Alabama	Massachusetts	Pennsylvania
California	Michigan	South Carolina
Colorado	Minnesota	South Dakota
Connecticut	Mississippi	Tennessee
Delaware	Montana	Texas
Florida	Nebraska	Vermont
Georgia	New Jersey	Virginia
Illinois	New York	Washington
Iowa	North Carolina	West Virginia
Kentucky	Ohio	Wisconsin
Louisiana	Oklahoma	

Initial contacts were based on prior knowledge of microcomputer usage in these states; subsequently, other states involved in microcomputer usage were identified. Additional information on states using microcomputers for bridge design-related activities was obtained from FHWA.

The survey consisted of targeting a knowledgeable person within a state bridge division or computer division and asking the following questions.

1. What kind of computer system is used for bridge design and analysis?
2. Do your engineers and designers have computer access through
   a. Direct access via a terminal?
   b. Submitting data using data entry forms (data actually entered and program run by others)?
3. Do you use microcomputers in bridge design?
4. If not, do you plan to purchase microcomputers in the near future for use in bridge design activities?
5. Do you use your microcomputer as
   a. A stand-alone unit?
   b. An intelligent terminal linked to a larger computer?
6. What kinds of bridge design programs are run on your microcomputer?
7. Can a list of these programs be made available?
8. Are your design programs
   a. Written in-house?
   b. Purchased from outside vendors?
9. What programming languages are used for programs written in-house?
10. Have you converted any programs currently running on a larger computer to run on your microcomputer?
11. If so, how was the program converted?
    a. Method of downloading used.
    b. Type of compiler or interpreter used.
12. Is increased use of microcomputers planned for the future? If so, what are your plans (e.g., upgrade to more powerful machines, microcomputer-aided design systems)?

The questions were designed to determine the current mainframe computing environment, to assess the level of satisfaction with this environment, to identify current utilization of microcomputers in bridge design applications, and to evaluate the attitudes and perceptions of engineers regarding the usefulness of microcomputers in design. Finally, plans for future implementation of microcomputers were discussed. A summary of the responses to the survey is given in Table 1.

As a result of this survey, several conclusions were drawn. First, the large majority of states uses mainframe computers in their bridge design and analysis work. A total of 30 out of 32 states, or 94 percent, use mainframes as their primary computing resource. The two remaining states use minicomputers. However, in almost all instances, the bridge divisions that use mainframes share them with other state agencies under some type of time-sharing arrangement.

Almost all bridge design groups (94 percent) have direct access to the computer through terminals located within the design group. In addition, some states with remote design locations, such as Pennsylvania, have terminal access at the district level. Through terminal access, the engineers are able to run mainframe applications in either interactive or batch mode, review the results, modify input if desired, and rerun the application. Some states, such as Michigan and Delaware, use screen forms packages that simplify data entry at the terminal by creating the actual input form for a given program on the terminal screen. Most states with computer configurations of this type expressed the belief that it served their computing needs well. Eleven of the 30 states with terminal access to a mainframe or minicomputer indicated that it served their computing needs completely and therefore those states showed little or no interest in using microcomputers.

However, the majority of the respondents did see some need for improvement of their computing environment. Reasons cited included slow turnaround time on time-shared systems, a desire for better access to software, and insufficient access to terminals connected to the mainframe. Of the 21 states that indicated a need for improvement in computer access, 9 including Virginia have begun using microcomputers in bridge design activities, although for the most part specific plans have not been developed (see Table 1).

The manner in which microcomputers are currently used for bridge design purposes varies widely from state to state. For example, in Montana, microcomputers are used almost exclusively for bridge design and analysis. Design and analysis programs previously run on IBM minicomputers were converted from their original FORTRAN coding to BASIC and adapted to an IBM-PC. In South Dakota, and as part of this project in Virginia, FORTRAN bridge design and analysis programs were downloaded from a mainframe computer and adapted to run on IBM-PC (or compatible) microcomputers using available microcomputer FORTRAN compilers. Ohio uses an IBM-PC 3270 networked to its mainframe and is in the process of developing some specialized bridge design-related applications. New Jersey, taking a similar approach, has recently purchased several IBM-PCs that will have communications capability with their mainframe via modems. These microcomputers were purchased to satisfy the needs of remote design locations for access to the mainframe and for stand-alone computing capability.

Other states are currently using microcomputers in bridge-related areas but to a lesser extent. New York uses microcomputers for project management functions and for field data collection and review. Future uses may include overload permit and splice design applications. Massachusetts currently uses an IBM-PC for field data collection and expressed intentions to use it for additional bridge design applications in the future. In Vermont an IBM-PC AT, to be delivered in the near future, will be the primary computer used for bridge design applications.

In addition to the states already using or beginning to use microcomputers, nine other states have indicated a desire to begin using them in the near future. Common among the responses from these states was an uncertainty about exactly what the capabili-

TABLE 1 Summary of Responses to Questions 1-4

State	Principal Computer Used for Bridge Design	Access to Mainframe via Terminals	Microcomputer Used for Bridge Design	Plans to Use Micros in Future for Bridge Design Applications
Alabama	MF	Yes	No	Yes
California	MF	Yes	No	Yes[a]
Colorado	MF	Yes	No	No
Connecticut	MF	Yes	No	No
Delaware	MF	Yes	No[b]	No
Florida	MF	Yes	No	Yes[a]
Georgia	MN	Yes	No	Yes[a]
Illinois	MF	Yes	No	Yes[a]
Iowa	MF	Yes	No	Yes[a]
Kentucky	MF	Yes	No[c]	Yes[a]
Louisiana	MF	Yes	No[d]	No
Massachusetts	MF	Yes	Yes[e]	Yes
Michigan	MF	Yes	No	No
Minnesota	MF	Yes	No	Yes[a]
Mississippi	MF	Yes	No	Yes
Montana	MN	Yes	Yes	
Nebraska	MF	Yes	No	No
New Jersey	MF	Yes	Yes	
New York	MF	Yes	Yes[b]	Yes[f]
North Carolina	MF	Yes	No	No
Ohio	MF	Yes	Yes	
Oklahoma	MF	Yes	No	No
Pennsylvania	MF	Yes	No[b]	No
South Carolina	MF	Yes	No	No
South Dakota	MF	Yes	Yes	
Tennessee	MF	Yes	No	No
Texas	MF	Yes	No	No
Vermont	MF	No	Yes	
Virginia	MF	No	Yes	
Washington	MF	Yes	Yes	Yes[a]
West Virginia	MF	Yes	No	No
Wisconsin	MF	Yes	No	Yes[g]

Note: MF = mainframe, MN = minicomputer.

[a]Plans not defined at present.
[b]Microcomputers are used for spread sheets, word processing, data base management, and the like.
[c]Microcomputers are used for planning.
[d]Microcomputers are used in roadway design.
[e]Microcomputers currently used for field data collection.
[f]Microcomputers would be used more for construction management, overload permit, and splice design work; "number crunching" would still be done on mainframe.
[g]Could possibly get involved with microcomputers if they demonstrate the ability to run large-scale programs in an efficient manner.

ties of microcomputers are when used in bridge design and analysis applications. Some engineers expressed doubts about the ability of these machines to handle large programs; doubts also arose about how the integrity of software would be maintained when it was distributed among several users.

Clearly there is a need to better define the role that microcomputers can play in bridge design at the state level. Several instances have been cited of private design firms in which proper implementation of microcomputers as a complement to present computer configurations has served to increase productivity and decrease overall computing costs. This should also be true in bridge design applications.

MICROCOMPUTER HARDWARE AND SOFTWARE

The hardware components of microcomputers, namely central processing unit (CPU), keyboard, and cathode ray tube (CRT), are becoming generally familiar, but specific hardware details and capabilities may not be so familiar. The current generation of 16-bit microcomputers generally uses one of three types of central processor, the Intel 8086, the Intel 8088, or the Motorola 68000 (1). The 8086 is a true 16-bit processor in that it moves data through a 16-bit data bus and processes 16 bits at a time. The 8088 moves data through an 8-bit bus and processes 16 bits at a time. The Motorola 68000 CPU, the most powerful of the three, handles data through a 16-bit data bus but processes 32 bits at a time. Internal memory is classified into two types, read-only memory (ROM) and random access memory (RAM). The ROM is factory installed and is read when the computer is turned on; it is permanent and cannot be altered by the computer operator. RAM is temporary memory and accessible to the user; it gives microcomputers their real power because it determines the size of applications that can be run. Four different 16-bit microcomputers were available for use during this project and are given in Table 2.

In addition to internal memory capabilities, microcomputers also have mass storage capability that enables them to access vast amounts of data outside the CPU. Mass storage memory usually refers to floppy diskettes or hard disks. Storage capacity on 5 1/4-in. floppy diskettes can range from 320 kilobytes (kb) to more than one megabytye. In general, floppy diskettes provide a reliable and portable form of mass storage, though lacking in access speed and overall storage capability. Hard disks provide much greater storage capacities and access data at significantly higher speeds than floppy drives. Hard disk capacities of 20 megabytes and more are common and some allow removal of the disks in a fashion similar to floppy diskettes. Although much more expensive than floppy drives, hard disk drives are becoming more commonplace as user requirements expand. The microcomputers used in this project all had mass storage capacity of 329 kb using double-sided, double-density floppy disk drives.

Another type of mass storage is commonly known as disk emulation or RAM disk. A RAM disk is created by software that in effect partitions unused RAM into what the computer treats as an additional disk

TABLE 2 Microcomputers Used

Attribute	Description
Zenith Z-151 (Marketed by NBI)	
Word length	16-bit
Processor	Intel 8088
Operating system	MS-DOS
Installed RAM	384 kb
Mass storage	Two 360-kb DS/DD disk drives
IBM Personal Computer	
Word length	16-bit
Processor	Intel 8088
Operating system	PC-DOS; CP/M; UCSD P-System
Installed RAM	596 kb
Mass storage	Two 360-kb DS/DD disk drives
COMPAQ Portable Computer	
Word length	16-bit
Processor	Intel 8088
Operating system	MS-DOS; CP/M86; UCSD P-System
Installed RAM	256 kb
Mass storage	Two 360-kb DS/DD disk drives
AT&T Personal Computer 6300	
Word length	16-bit
Processor	Intel 8086
Operating system	MS-DOS
Installed RAM	512 kb
Mass storage	Two 360-kb DS/DD disk drives

drive. This form of mass storage provides the fastest access time because there are no mechanical drive parts involved, such as read-and-write heads. However, RAM disks are limited in capacity to whatever RAM is not used during the application.

In general, it is far more important to consider software than hardware capabilities. The most fundamental piece of software is the operating system that ties the CPU and memory to the display, keyboard, and disks. Some of the different operating systems available for the 16-bit microcomputers are MS-DOS, CP/M-86, and the UCSD P-System for single-tasking operations and Unix from Bell Labs, MP/M (an advanced version of CP/M), Pick, and Oasis for multitasking.

In this study, four different microcomputers were used and MS-DOS version 2.11 was the operating system used on all four machines (Table 2). Two capabilities of MS-DOS, which served well when running the large FORTRAN bridge design programs encountered in this project, were: (a) output files could be spooled to the printer while program execution continued and (b) batch capabilities allowed several program runs without an operator present. Because the execution time of some programs on microcomputers is slow relative to larger machines, the batch capability is a distinct benefit.

The ability of the 16-bit microcomputers to handle a wide variety of programming languages is further indication of their computing power and versatility. Most of these machines come with a BASIC interpreter, but there are also several dozen compilers available for a variety of languages. A fairly complete listing of these compilers and languages, taken from Ruby (2), follows.

PASCAL compilers

1. Turbo PASCAL (Borland International)
2. PASCAL/MT+ (Digital Research)
3. Micro Concurrent PASCAL (Enertec, Inc.)
4. UCSD PASCAL Compiler (IBM)
5. IBM PC PASCAL Compiler 2.0
6. MS PASCAL (Microsoft)
7. PASCAL 86/88 (Real-Time Computer Science Corporation)
8. UCSD PASCAL Compiler (Softech Microsystems)
9. Concurrent PASCAL 8086 (Soft Machines, Inc.)
10. SBB PASCAL (Software Building Blocks)

BASIC compilers

1. CRASIC Compiler 2.0 (Digital Research)
2. BASIC Compiler (IBM)
3. ATV/BASIC (LanTech Systems, Inc.)
4. BASIC Compiler (Microsoft)
5. Business BASIC
6. BASIC Compiler (Quantum Software Systems)
7. BASIC Compiler (Softech Systems)
8. BASIC (Supersoft)
9. Squish (Versaterm Systems, Ltd.)

BASIC interpreters

1. Bl-286 1.4 (Control-C)
2. BASIC Interpreter (Microsoft)

Combined BASIC compilers and interpreters

1. APC BASIC (American Planning Corporation)
2. MegaBASIC
3. HAI*BAS (Holland Automation USA, Inc.)
4. Professional BASIC (Morgan Computing Company, Inc.)
5. Better BASIC (Summit Software Technology, Inc.)

Modula-2 compilers

1. Logitech Modula-2/86 (Logitech, Inc.)
2. Modula-2 for the IBM-PC (Modula Corporation)
3. M2M-PC (Modula Research Institute)
4. Volition Systems Modula-2 (Volition Systems)

APL interpreters

1. IBM-PC APL (IBM)
2. Sharp APL/PC (I.P. Sharp Associates, Ltd.)
3. APL*PLUS/PC (STSC, Inc.)
4. WATCOM APL (WATCOM Products, Inc.)

FORTRAN compilers

1. FORTRAN 77 (Digital Research)
2. FORTRAN 77 Compiler (IBM)
3. FORTRAN Compiler 2.0
4. FORTRAN Compiler (Microsoft)
5. 87 FORTRAN/RTOS (MicroWare, Inc.)
6. FORTRAN 86/88 (Real-Time Computer Science Corporation)
7. FORTRAN 77 (Quantum Software Systems, Inc.)
8. FORTRAN 77 (Softech Microsystems)
9. FORTRAN Compiler (Supersoft)
10. Professional FORTRAN (IBM)
11. R/M FORTRAN (Ryan-McFarland)

FORTH compilers and interpreters

1. HSFORTH 2.01 (Harvard Softworks)
2. PC/FORTH 3.0 (Laboratory Microsystems, Inc.)
3. PC/FORTH+ 3.0
4. MMSFORTH (Miller Microcomputer Services)
5. MVP-FORTH PAD (Mountain View Press)
6. FORTH-32 (Quest Research)

C compilers

1. C Compiler (C-Systems)
2. C Compiler (C Ware)
3. CC 86 (Control-C Software)

4. C86 (Computer Innovations, Inc.)
5. Small-C:PC (Custom Software)
6. Digital Research C3 (Digital Research)
7. Lattice C Compiler (Lifeboat Associates)
8. Aztec C 86 1.06D (Manx Software Systems)
9. MWC-85 (Mark Williams Company)
10. C Compiler (Microsoft)
11. C Compiler (Quantum Software Systems, Inc.)
12. Instant C (Rational Systems)
13. C 86/88 (Real-Time Computer Science Corporation)
14. C Compiler (Supersoft, Inc.)
15. C Compiler (Telecon Systems)
16. C Compiler (Whitesmith's, Ltd.)

COBOL compilers

1. COBOL Compiler (Digital Research)
2. MBP COBOL Compiler (MBP Software Systems Technology)
3. Level II COBOL Compiler 2.6 (Micro Focus, Inc.)
4. Personal COBOL
5. COBOL Compiler (Microsoft)
6. RM/COBOL (Ryan-McFarland)

In this study, FORTRAN was used for all of the applications and the Microsoft FORTRAN compiler was the most convenient to use. The MS-FORTRAN compiler conforms to subset FORTRAN as described in ANSI X3.9-1978 and also contains extensions to the standard. These extensions are listed in the MS-FORTRAN User's Guide, Appendix A (3). Minimizing use of these extensions increased portability and allowed the bridge design programs to be run easily on other microcomputers and the University of Virginia's Cyber mainframe.

APPLICATION SOFTWARE

With the tremendous growth in microcomputer hardware has come a corresponding growth in application software and software vendors. The number of application programs for civil engineering and construction alone has become so large that they are catalogued in Hunt's Directory (4), a good source of software for potential bridge applications. Currently the majority of vendor-supplied programs is analysis packages rather than design applications because design programs typically require more upkeep because of code changes. A review of several software sources determined that, in the area of bridge design, few design applications were available. Design packages that were found included three bridge design systems for small bridges, a pier design program, a pile design program, an influence line generation program, and several coordinate geometry programs. However, almost every conceivable type of structural analysis program was available for all makes of microcomputers. These analysis packages ranged from simple-beam analysis to full-feature integrated finite element analysis packages.

Most states perform in-house software development for their mainframe applications; because the use of microcomputers is not great, similar software development for them is limited. A few state bridge divisions that currently use microcomputers in design, such as Montana, Ohio, and Virginia, develop some software in-house. Such programs are typically written in BASIC, although Montana has converted several bridge design applications from a FORTRAN code running on an IBM 5100 minicomputer to BASIC for use on an IBM-PC. The following list gives typical bridge design applications developed in this manner.

1. Bridge centerline grade (Virginia)
2. Steel beam or girder section properties in negative moment region (Virginia)
3. Steel beam or girder section properties (Virginia)
4. Critical moments and shears (Virginia)
5. Concrete section analysis (Virginia)
6. Live load reactions on pier or abutment (Virginia)
7. Bolted beam/girder splice design and analysis (Virginia)
8. Concentric curve skewed bridge geometry (Virginia)
9. Bearing stiffener design or analysis (Virginia)
10. Transverse stiffener design or analysis (Virginia)
11. Straight roadway skewed bridge geometry and elevations along lines (Virginia)
12. Various programs to determine bridge geometry and elevations (Montana)
13. Various programs to determine bent and girder reactions due to various standard and nonstandard loadings (Montana)
14. Slab analysis by working stress design or ultimate stress design (Montana)
15. Prestressed beam analysis (Montana)
16. Prestressed bulb T-beam analysis (Montana)
17. Welded plate girder analysis (Montana)
18. Two-column bent programs (Montana)
19. Coordinant geometry program (Montana)
20. Beam splice design (Ohio)
21. Crane loading program (Ohio)
22. Analysis of composite rolled beam (Ohio)

Most of these programs are small and designed to perform rather specialized functions. Although a useful first step, they do not fully meet the need of bridge divisions for general application programs to run on microcomputers.

Potentially one of the most attractive schemes for development of microcomputer software for bridge design applications is the downloading and conversion of existing mainframe programs. There are several advantages to having the ability to run large-scale converted mainframe bridge design software on a microcomputer. First, it provides greater flexibility to the engineer. Applications can be run at any time without the need for access to a mainframe. A state bridge division may be only one of several state agencies that must share time on a mainframe; thus, depending on demand, computer access may not always be possible because of low priority. Also, microcomputers can insulate bridge designers from the inconveniences of unscheduled mainframe downtimes. The converted programs will also be familiar to the users. Programs that were converted as part of this study used the same input and output format as those run on the mainframe. In states in which design activities are carried out at remote locations, microcomputers can provide an efficient and relatively inexpensive means of distributing computer power. The high costs of communicating with mainframes over telephone lines can be minimized. There are also other benefits to be realized. Converting mainframe bridge design software to microcomputer use will ease demand on the mainframe and allow more processor time for other, larger agency applications.

As part of this study, several attempts at downloading and converting mainframe programs were made. These conversions provided a means of identifying problems and the level of effort required. With the assistance of the Bridge Division and the Information Systems Division of the Virginia Department of Highways and Transportation, copies of the following bridge design problems were obtained.

1. Prestressed Concrete I-Beam Design and Analysis Program
2. Steel Girder Design and Analysis Program (composite)
3. Deck Slab Design Program
4. Critical Moments and Shears on a Simple Span for Moving Loads
5. Bridge Geometry Program
6. Georgia Continuous Beam Program
7. Georgia Pier Program
8. SIMON (a complete design system for steel bridge girders)

Successful conversions were made on the first four programs, but a number of problems were encountered in attempting to convert the remaining programs. First, most programs currently run on mainframes have been around for a long time and are written in early versions of FORTRAN. Some programs, such as the Bridge Geometry Program, were originally written in assembly language and then converted to FORTRAN. Still others were written such that they required machine-dependent software. These types of problems require extensive changes in coding. Major portions of some of the programs, which were not converted, would have had to be completely rewritten. Another obstacle to program conversions can be the programming technique of the original programmer. An example of this occurred in both the Georgia Continuous Beam and the George Pier programs. These are long programs with few subroutines; this causes problems because large programs usually must be broken into groups of subroutines to be compiled on a microcomputer, and programs without subroutines may require major alterations to existing code. A final obstacle to converting mainframe programs to the microcomputer is program size. Some programs are simply too large to be converted for use on the present generation of 16-bit microcomputers.

The bridge design application programs used in this project consisted of a Prestressed Concrete I-Beam Design and Analysis Program, a Steel Girder Design and Analysis Program, a Deck Slab Design Program, and a Critical Moments and Shears Program. These programs are currently used by the Bridge Division of the Virginia Department of Highways and Transportation on an IBM 3084 mainframe computer. All four of these programs are written in FORTRAN and were converted from mainframe use for use on microcomputers. As an example of the type of bridge design applications that the 16-bit microcomputers are capable of running, two of the larger programs (Prestressed Beam and Steel Girder) were used to run example problems.

Two different example problems were selected for each program and these were run on four different microcomputers and on an additional machine equipped with an 8087 math coprocessor chip. Details of these runs are given in Table 3. Also, Table 3 gives the program source file size and executable run file size for the Prestressed Beam and Steel Girder programs. The Prestressed Beam Program is a fairly long program with approximately 3,000 FORTRAN statements in the source file and an executable run file size of 161,480 bytes. This size program would certainly not run on the earlier 8-bit machines. Theoretically, an IBM-PC with full RAM capacity of 640 kb could run an application program of comparable size. The data in Table 3 indicate not only that programs of significant size do run on the 16-bit microcomputers but also that they execute in a reasonably short time.

Program size is only one of the factors that affect program execution. Another factor that will affect execution time is the type of CPU employed by the microcomputer. The IBM-PC, the Compaq Portable, and the Zenith-151 all use the Intel 8088 CPU. Comparison of test results for the two programs on the IBM and Compaq machines shows virtually identical execution times; however, execution time on the Zenith Z-151 is about 20 percent slower. The probable causes of this are differences in the basic input-output system and elsewhere in the system architecture of the machines (5). The execution times for the test problems using the AT&T PC with the 8086 CPU turned out to be faster than those of the 8088 machines. This is not surprising because the 8086 moves data to and from the CPU through a 16-bit data bus versus an 8-bit bus on the 8088 machines.

Another hardware feature that may have a dramatic effect on program execution time is the 8087 math coprocessor. The test problems in Table 3 show a decrease in execution time of up to 60 percent using an 8087. The extent to which the 8087 math coprocessor will decrease execution time depends largely on the math processing requirements of the program at hand. In general, the more "number crunching" required, the more benefit will be realized from the 8087. All of the bridge design software of this project, and most available commercially, will be able to take advantage of an 8087. There are certain disadvantages, however, to using the 8087. It draws a significant percentage of the power supplied to the system board of a microcomputer and also dissipates a significant amount of heat. Excessive power consumption and heat dissipation can cause erratic operation of the disk drives, memory malfunctions, periodic lockup of the computer, unsafe heat buildup inside the computer cabinet, and possible eventual burnout of the power supply. It has been found that most combinations of the expansion cards with an 8087 will allow safe operation of the microcomputer, but, because of the possible detrimental effects, each individual microcomputer system should be properly evaluated before adding the 8087 coprocessor (5).

It has been noted that a RAM disk may offer increased efficiencies for running certain programs. To illustrate the performance of a RAM disk, the same test problems from Table 3 were run using a RAM disk. The results of the new runs are given in Table 4. The amount of storage in the RAM disk drive varied among machines depending on available RAM. Enough storage was allocated for the RAM disks to allow the executable run files and the input and output files to be stored. This permits direct comparison of the results summarized in Tables 3 and 4. Comparison of the results in Table 4 with those of Table 3 shows

TABLE 3 Bridge Design Program Characteristics Illustrating Memory Capacity and Execution Time

	Execution Time (sec)				
Test Problem	Zenith Z-151	IBM-PC	COMPAQ Portable	AT&T PC	COMPAQ W/8087
Prestressed Concrete I-Beam Design and Analysis Program[a]					
PB1[b]	54	43	43	33	36
PB2[c]	57	46	47	35	38
Steel Girder Design and Analysis Program[d]					
SG1[e]	30	24	24	20	20
SG2[f]	88	74	75	50	36

[a]FORTRAN statements in source file: 2,961 and executable file size: 161,480 bytes.
[b]Design of a standard AASHTO type 5 beam with standard HS-20 loading. (See Appendix B for input form)
[c]Design of a non AASHTO beam for HS-20 loading and additional concentrated dead loads. (See Appendix B)
[d]FORTRAN statements in source file: 896, and executable file size: 90,360 bytes.
[e]Complete analysis of an interior bridge girder of composite construction. (See Appendix B)
[f]Three separate complete designs of an interior composite bridge girder at web depths of 48 in., 51 in., and 54 in. (See Appendix B)

TABLE 4 Comparison of Bridge Design Program Execution Times Using a RAM Disk for Input and Output

	Execution Time (sec)				
Test Problem	Zenith Z-151	IBM-PC	COMPAQ Portable	AT&T PC	COMPAQ W/8087
Prestressed Concrete I-Beam Design and Analysis Program					
PB1	–[a]	24	–[a]	12	–[a]
PB2	–[a]	27	–[a]	14	–[a]
Steel Girder Design and Analysis Program					
SG1	16	12	12	7	9
SG2	83	59	59	28	22

[a]Insufficeint memory exists to simultaneously create an emulated disk drive and run the program.

that disk emulation significantly decreases execution time in all cases. These decreased execution times can be attributed wholly to decreased input-output time and the decreased time required for the programs to be loaded into memory (no mechanical drive components are involved).

Whether application software is purchased, developed in-house, or converted from mainframe programs, considerations such as maintenance, portability, and distribution control cannot be neglected. Many of the mainframe programs used for bridge design applications in Virginia and other states are shared among states. The state that developed a given program usually assumes responsibility for maintaining the program and implementing major changes. If one of these programs has been converted for microcomputer use, subsequent changes must be transferred to the converted version. This may prove difficult if changes are not well documented and if the conversion requires extensive source code modifications.

Changes in computing technology or outgrowing present computing facilities, or both, may necessitate a future changeover to more powerful and sophisticated microcomputers. This can have a drastic effect on currently used software if software portability has not been considered. When software is being planned, the potential for future migration of programs to other computers must be considered. One way to maximize portability is to use standard features of standard programming languages and minimize the use of proprietary languages. In cases in which individual users continue to write programs, portability can be maximized by imposing guidelines for program development. These guidelines should specify the languages and operating systems that can be used. Complete program documentation should also be required.

A major consideration that has become intrinsically associated with microcomputers is control over the distribution of software. Microcomputers have ushered in the age of truly distributed computing power, and associated with this distribution of computing power is the distribution of software. Some form of control is necessary to properly manage this distribution and to maintain the integrity of common software used within an organization. However, excessive control may serve to stifle use of the software and result in reduced efficiency.

Information and examples given thus far make it clear that the current generation of microcomputers possesses sufficient computing power to be seriously considered as an alternative to mainframe computers for bridge design applications and that there is considerable interest in such utilization. As this interest translates into microcomputer implementation, more and more microcomputer bridge design and analysis software will become available. It has already been noted that considerable programming of small design aids has been and is taking place, and, in Virginia and South Dakota, some conversion of mainframe software is taking place. These microcomputer programs should be available for sharing among the state bridge divisions. The following converted mainframe programs are currently available and can be obtained by contacting the bridge division in the appropriate state.

1. Prestressed Concrete I-Beam Design and Analysis (standard AASHTO and nonstandard simple-span bridge girders) (Virginia)
2. Steel Bridge Girder Design and Analysis (Virginia)
3. Deck Slab Design (Virginia)
4. Critical Moments and Shears on a Simple Span (Virginia)
5. Georgia Bent Program (South Dakota)
6. Continuous Span Prestressed Concrete Bridge Girder Design (South Dakota)
7. PCA Reinforced Concrete Column Design (South Dakota)

SCENARIOS FOR MICROCOMPUTER IMPLEMENTATION

How and when a state DOT bridge design unit should start using microcomputers depend on several factors. Basically, microcomputer use should be considered whenever present computing capabilities require enhancement such as additional computing power, distribution of computing power, and addition of communications capabilities.

The basic computing configurations for 16-bit microcomputers are either as stand-alone operation or as intelligent terminals linked to mainframes. In a stand-alone mode the microcomputer can operate independently and provide the engineer with a means of using a significant computing resource without the disadvantages of a time-shared mainframe system. The advantages of using a microcomputer as an intelligent terminal are numerous. Indeed, the ability to use a microcomputer in this mode is an example of how microcomputers can complement existing computer configurations in an efficient and cost-effective manner. The key here is the ability of the microcomputer to communicate with a mainframe computer. Communication enables the engineer both to complement mainframe operations with the microcomputer capabilities and to use mainframe resources to expand microcomputer power. A number of communications packages are available that allow engineers to communicate with virtually any mainframe system. In this mode the microcomputer can be used to run applications that are, at present, too large for microcomputer implementation. Also, off-line preparation of data represents a potential for considerable cost savings.

When both personnel and machine costs are considered, the costs of communicating between terminals and the mainframe becomes a relatively large portion of the total computation cost because the cost of computing is decreasing while those of communication and personnel continue to rise (6). Applications that use microcomputers to assist in the preparation of data and speed communications to the mainframe show great potential. However, there are a number of costs inherent in microcomputer implementation that go beyond the initial purchase price. These costs include service and maintenance costs, training costs, and additional hardware and software costs.

A major cost consideration is that related to training. For example, it may be necessary to form and staff internal user support groups. Other training-associated costs may include the value of the time it takes individual users to learn how to operate the computer, the value of productivity lost while the engineers become computer proficient, the cost of time lost attempting to train persons who never become computer proficient, and even the cost of time lost when skilled users interrupt their own work to assist less skilled users with a problem (7). The bottom line with training costs is that time is much more expensive than hardware or software.

One of the major obstacles to large-scale microcomputer implementation by bridge design groups is divergence from traditional computerization norms. Much computing in typical bridge design groups is done through a mainframe controlled by a computer systems group. The type of support required by microcomputer implementation will require some level of involvement by a computer systems group. The expertise these groups possess in computer hardware systems, and software development and maintenance, will be necessary for proper microcomputer implementation and support. However, for proper microcomputer implementation, changes in traditional attitudes toward computing will be necessary and these groups may, at least initially, be reluctant to accept changes necessitated by the most efficient microcomputer implementation.

Scenarios for microcomputer implementation will vary from state to state because of differences in present computing configurations and the level of satisfaction with these systems. Future computing needs will also play a major role. In states in which mainframe access is good all the way down to the district level and the level of satisfaction is high, microcomputers may play a minor role at best. However, in states in which engineers are hampered in their access to a mainframe or dissatisfied with the service they receive, microcomputers can be a distinct benefit. Their usefulness is bound only by the imagination of the engineers and their ability to modify problem-solving techniques and office procedures to harness the computer's power more effectively (6).

SUMMARY

In this study an effort was made to assess the present overall computer configurations used in state DOT bridge design groups; to determine present utilization of microcomputers in these groups; to illustrate applications of microcomputers in bridge design activities; and, finally, to develop scenarios for the application of microcomputers in bridge design. The feasibility of downloading and converting mainframe programs and the ability of microcomputers to run large bridge design applications efficiently in a stand-alone mode were demonstrated.

The development of microcomputers signals a new era in computer use. The significant computing power they possess, along with their relatively low cost compared with traditional large computers, has assured their success. Their use is being constantly explored in many business and engineering applications. Many state department of transportation bridge design groups are in a position to make full use of microcomputer capabilities, and some states are already beginning to do so. Although there are many serious organizational and financial considerations, a well-planned computing system with microcomputers that complement existing mainframes can significantly improve computing methods and increase efficiency and productivity.

REFERENCES

1. C. Lu. Microcomputers: The Second Wave. High Technology, Sept./Oct. 1982, pp. 36-52.
2. D. Ruby. With Expanding Options, Who's Programming In-House and With What Languages. PC WEEK, Aug. 28, 1984.
3. Microsoft FORTRAN Compiler For the MS-DOS Operating System, Users Guide & Reference Manual. Microsoft Corporation, Bellevue, Wash., 1984.
4. A.J. Hunt, ed. Hunts Directory of Microcomputer Software and Services. Pleasant Hill, Calif., 1984.
5. C. DeVoney. IBM's Personal Computer. Que Corporation, Indianapolis, Ind., 1983.
6. G.W. Davies. Microcomputers in Civil Engineering. In Transportation Research Record 932, TRB, National Research Council, Washington, D.C., 1983, pp. 20-23.
7. Cutting Through The Hidden Costs of Computing. Personal Computing, Oct. 1984, pp. 122-129.

Publication of this paper sponsored by Committee on General Structures.