UNITED STATES
DEPARTMENT OF THE INTERIOR
U.S. GEOLOGICAL SURVEY

U.S. GEOLOGICAL SURVEY MINI IMAGE PROCESSING SYSTEM (MIPS)

By Pat S. Chavez, Jr.

Open File Report 84-[redacted]
880

Reston, Virginia
1984

CONTENTS

ILLUSTRATIONS

U.S. GEOLOGICAL SURVEY MINI IMAGE PROCESSING SYSTEM (MIPS)

By Pat S. Chavez, Jr.
U.S. Geological Survey
Flagstaff, Arizona 86001

ABSTRACT

The U.S. Geological Survey Mini Image Processing System, or MIPS, was designed as an office research and development image processing system with the potential of being used in a field environment. The hardware and the operating system software were selected to take advantage of the approximately 70 to 80 man-years of image processing software developed for the Flagstaff Image Processing System.

The main objective in designing MIPS was to assemble an office system that would be relatively inexpensive, but still powerful enough to be a stand-alone system. The MIPS software package has been designed to allow optimization of the processing speed with a minimum use of assembler code. The program documentation, screen formatting, and user prompting modifications recently made to MIPS have greatly improved the user friendliness of the system. The software base and design make it a very efficient system.

INTRODUCTION

The U.S. Geological Survey (USGS) Mini Image Processing System, or MIPS, was designed as an office research and development image processing system with the potential of being used in a field environment. The PDP 11/23 hardware and the RSX-11M operating system software were selected in order to make MIPS compatible with the major existing Flagstaff Image Processing System (FIPS), which operates on PDP 11/45 and 11/44 computers using the RSX-11M operating system located in the USGS Computer Center in Flagstaff, AZ. The system was designed in this manner to avoid the large expense that can be incurred for software development and conversion, which can easily exceed the cost of the hardware. Thus the MIPS design has available approximately 70 to 80 man-years of image processing software developed for FIPS. Because of this compatibility, new software can be easily interchanged between the two systems and, as an added advantage for a small additional cost, MIPS can be included in an existing FIPS hardware maintenance contract.

Publication authorized by the Director, U.S. Geological Survey, on October 1, 1984. Any use of trade names and trademarks in this publication is for identification purposes only and does not constitute endorsement by the U.S. Geological Survey.

The main objective in designing MIPS was to assemble an office system that would be relatively inexpensive, but still powerful enough to be a stand-alone system where efficient application of the system is the most important parameter. Therefore, once digital data are recorded on magnetic tape, MIPS should be able to reduce, store, edit, enhance, analyze, display, and correlate spatial data without having to go back and forth to a larger mainframe computer.

DISCUSSION

At the first meeting of researchers on the National Mapping Division Spatial Data Processor (SDP) research project team in November 1983, six features were identified as being critical when evaluating spatial processing systems: (1) functionality, (2) software base, (3) long-term support, (4) modularity, (5) software transportability, and (6) expandability. The material that follows discusses the capabilities of MIPS in these six areas. Because some capabilities of the system are applicable to more than one feature, certain comments may be repeated.

Functionality

Functionality is considered to be the most important factor when evaluating a system. The functionality of the system is determined by (a) the number of available options/functions, (b) the efficiency of the processing mode, (c) the maximum size of data set that can be processed, (d) the ease of adding new functions/capabilites, and (e) the ease with which the system can be used (that is, how user-friendly). If the options/functions are not designed correctly, the system will be either too slow or unable to handle a large enough data set to do any real work, and thus its potential will be minimized.

Although the functional capabilities of FIPS and MIPS are identical, the MIPS version has undergone modifications to upgrade the internal program documentation. These modifications have also helped to improve external documentation and to make the software more user-friendly. The FIPS/MIPS software is composed of individual programs and operations that are functionally independent. However, they can be applied in a series of steps to a common data base to generate the desired results. In addition, the design allows easy implementation of new programs that will execute in an efficient manner (discussed further in Software Base).

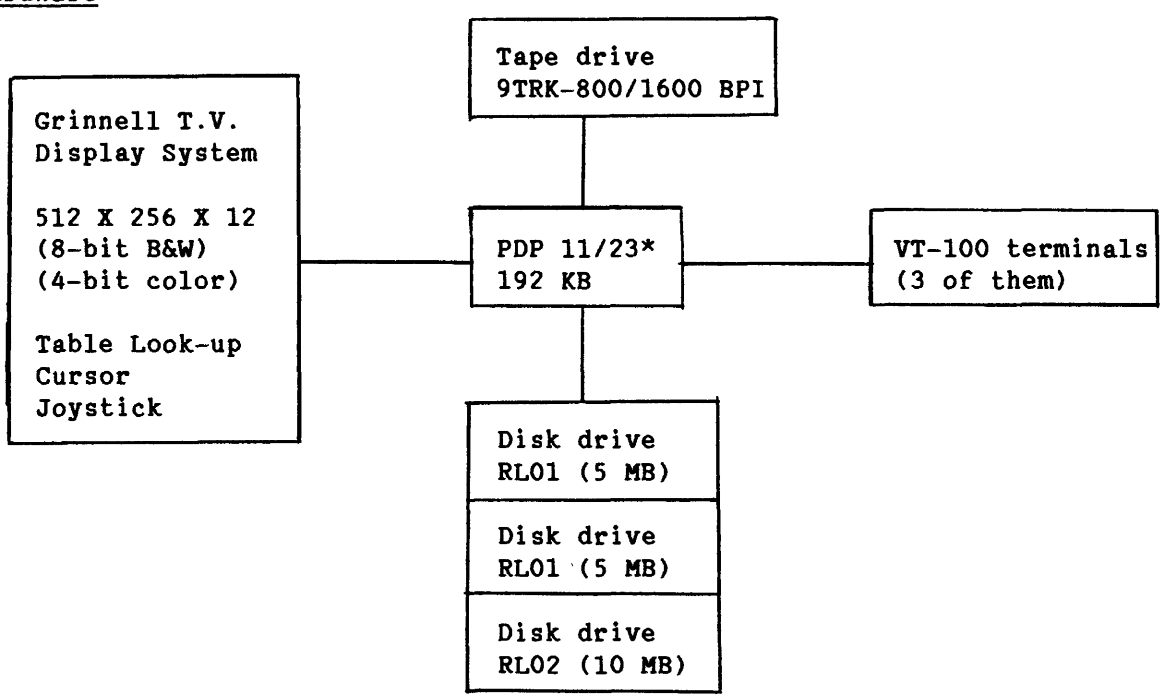
In order to make MIPS or any new spatial data processor a stand-alone system, a magnetic tape drive must be a part of the system. This permits the system to read and write all data formats required and input/output (I/O) the digital data to magnetic tape. Once the software to read the major data formats is written (most of which already exists), the system is able to enter and process most digital data sets. When a data set with a different format is encountered, a program can be written to convert it to the required vector or raster format for subsequent processing using existing tape handling and reformating subroutines.

Data digitized from film and maps are available through non-MIPS spatial data processor hardware (for example, photoscanners and table digitizers). A stand-alone office system does not need these added hardware features, but it should have the necessary software to do editing and processing of the digitized data once the data are on magnetic tape. The same is also true for high-quality hardcopy reproduction hardware. Generation of image products is accomplished using non-MIPS spatial data processor hardware, such as a photowrite unit.

The software design of MIPS allows for processing of large spatial arrays. The maximum array size can usually handle a full Landsat MSS image and often larger arrays of up to 6,000 or 7,000 pixels per line. The limiting size factor is usually the amount of disk storage available on the system and not the amount of memory. As can be seen in figure 1, there are a total of 20 megabytes (Mb) of disk storage available, which can be expanded if needed (figure 2).

The applications software design makes the I/O and execution time highly efficient. However, much of the software is written in FORTRAN and designed in such a way that transportability is possible (discussed further in Software Base and Transportability).

Hardware



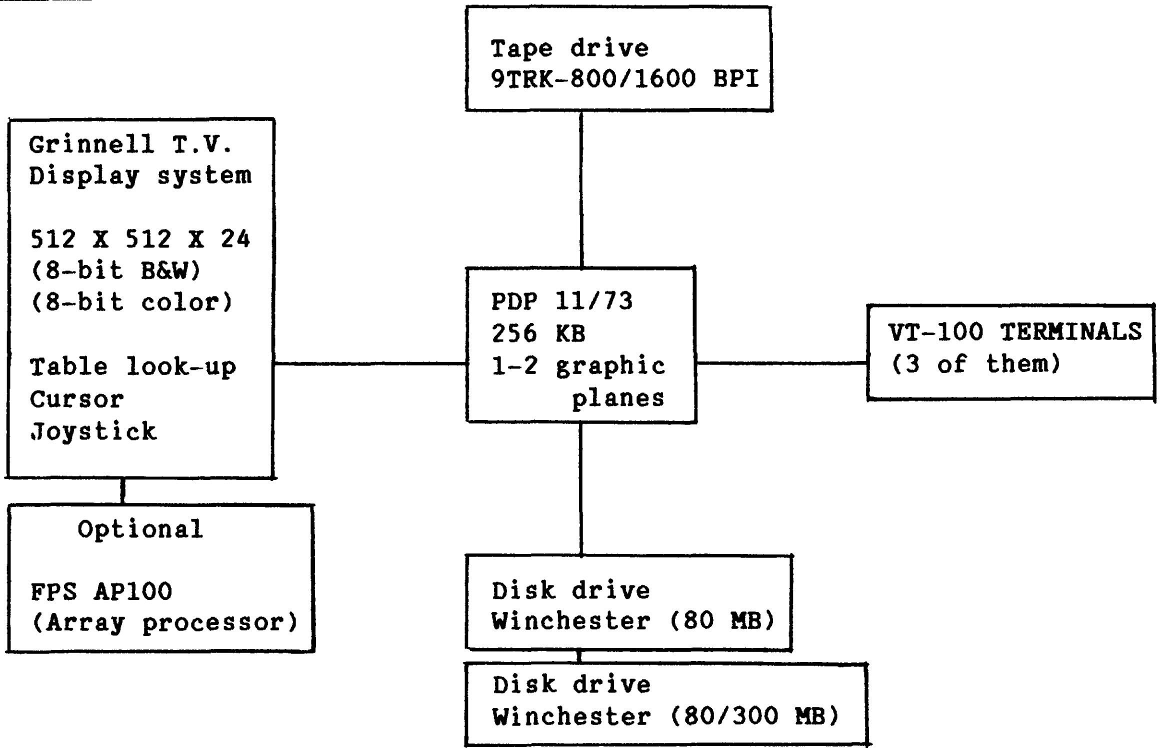
*Currently being upgraded to an 11/73

Software

Operating System	RSX11M
FORTRAN	FORTRAN-4 Plus
Macro	PDP Assembler
Application	All written in Flagstaff by USGS in FORTRAN and Macro

Figure 1.--Hardware and software configuration for current (1984) Mini Image Processing System

Hardware



Software

Operating System	RSX11M (11/73 can run UNIX)
FORTRAN	FORTRAN-4 Plus
Macro	PDP Assembler
Application	All written in Flagstaff by USGS in FORTRAN and Macro

Figure 2.--Hardware and software configuration for possible Mini Image Processing System upgrade

Some examples of execution speed on the current MIPS (figure 1) for a spatial array of 500 x 500 pixels are as follows: shaded relief image of an 8- or 16-bit input file--2 minutes; stereopair generation--3 minutes; spatial filtering using a 3 x 3 to 501 x 501 (or M x N, M = or ≠ N) filter with anywhere from 0 to 100 percent addback of 8- or 16-bit input images--3 and 3.5 minutes, respectively; ratio of two images--2.5 minutes; and nearest neighbor geometric resampling--2.5 minutes. Also, using the 512 x 256 refresh memory, MIPS can accomplish color coding or contrast stretching, with any desired amount of operator control, of an 8-bit image in near realtime.

The MIPS processing functions can be categorized under one of the following headings: (a) geometric corrections, (b) radiometric processing, (c) information extraction and data manipulation, and (d) utility/miscellaneous software. Several representative examples of software capabilities are shown below for each category.

A. Geometric corrections
   1. image to map transformation
   2. image to image transformation
   3. projection i to projection j conversion
   4. convert non-image data to:
      a. image
      b. other non-image data
      c. map
   5. performs geometric corrections with or without ground control points
   6. vector to image/raster conversion
   7. affine scaling and rotation
   8. slant range to ground range corrections for radar and sonar data
   9. perspective view projections using digital elevation data

B. Radiometric processing
   1. noise removal
      a. destriping (MSS data and others)
      b. bit error correction
      c. radar speckle removal
      d. dropped line replacement
      e. general high frequency noise pattern removal (horizontal and vertical)
   2. shading corrections
      a. radar and sonar power-loss corrections
      b. video camera (RBV and digitized photos) radiometric distortions
      c. general low frequency shading interference (often seen in geophysical and geochemical data)
   3. atmospheric scattering (haze) correction
   4. sun elevation normalization
   5. radiometric calibration for multitemporal or multisensor data (correct for gain and offset, as well as haze and sun elevation)

C. Information extraction and data manipulation
   1. contrast stretches (linear, ramp, table, etc.)
   2. band ratios
   3. sums and differences of bands
   4. spatial filtering (3 x 3 to 501 x 501 kernel sizes with any percent addback)
   5. first difference/derivative (three directions)
   6. principal component transformations
   7. clustering
   8. parallelepiped classification
   9. statistical information extraction
      a. average and standard deviation
      b. correlation coefficient
   10. overlay function (two to six bands at a time)
   11. distance function generation
   12. neighborhood operations
   13. reclassify by attribute

14. interactive data analyses
    a. contrast stretch (linear, ramp, thematic--manual or automatic)
    b. color coding (manual, semiautomatic, fully automatic)
    c. stereoviewing
    d. parallelepiped classification
15. digital elevation data processing (stereo, perspective view, shaded relief, etc.)

D. Utility/miscellaneous software
1. mosaicking
    a. tone matching
    b. geometry matching
    c. windowing
2. interpolation/surface generation (contour or random data)
3. format conversions (vector and raster data)
4. tape-to-disk input
    a. full area
    b. subarea
    c. full or subarea at a given subsampling interval
5. frequency histogram generation
6. digital number (DN) listings (at any sampling interval)
7. three-dimensional scattergram generation
8. profiling

One area in which the current MIPS software package needs to be expanded is digital classification of multispectral data. Historically MIPS users have not had applications that required a classification capability. Also, the processing of vector data is currently done in raster format, and it might be desirable to have several vector processing algorithms. However, all of the vector data that have been processed using this software package have been efficiently handled in raster format.

Software Base

A decision that must be made when choosing or writing a software package is whether it should be efficient in executing the desired functions or be compatible or transportable with a large number of hardware systems. System designers must be careful that desires for transportability do not create a design so general that the package becomes very inefficient in executing the desired functions. However, the opposite extreme where software is hardware dependent, making the sharing of software impossible unless the same hardware is used, is not usually a desireable characteristic.

As was stated earlier the FIPS applications software package was developed during the past 11 years by the Geological Survey in Flagstaff and is public domain information with no special restrictions on its use or distribution. The software has been designed so that it runs as efficiently as possible on the given hardware without having to write the entire package in macro/assembler language. In order to do this, assembler subroutines to handle the I/O were developed to greatly improve tape, and more importantly, disk efficiency. Also, many programs have been written using a DO-ALL concept in which all machine-dependent operations such as I/O are isolated from applications routines, making it easy to add new software and transport the existing software to other systems. The DO-ALL concept was developed in 1974 and has evolved into what is currently called the DOIO subroutine. DOIO is designed to accomplish four specific goals in the programming and user environment: (1) reduce the amount of program coding that a programmer needs to write when developing new spatial processing software, (2) minimize the amount of system level details that a programmer needs to consider when developing new spatial processing programs, (3) provide program standardization for the user of spatial processing software, and (4) make it easier to transport the software to other systems. By utilizing the DO-ALL concept, this software package can be used on other systems with a limited investment in software conversion. This subject, along with some new user-friendly features that have been incorporated into MIPS, are discussed further in Software Transportability.

There are four MACRO 11 assembler subroutines which are very important to the MIPS software package for efficient programming, execution, and user interface during execution. The four subroutines are:

DISKIO This is a general-purpose subroutine that allows high-speed access to spatial data stored on disk. It gains its speed by bypassing most of the file system overhead.

TAPEIO This is a general-purpose tape input and output subroutine. It allows the user to read and write data on tape, skip records and files, rewind and backspace records and files, write tape marks, and specify tape density and parity.

IPCSI: This subroutine reads a command line and breaks it up into component file specifications. The subroutine returns the number and names of input and output files specified by a user command line and allows the user to put an upper limit on the number of files to be used. A flag is set if an end of file or an error condition is encountered.

PARAMS: This subroutine reads free format input lines containing keywords/user parameters and returns the input values to the appropriate variables in the calling computer program.

Of the four subroutines, the programmer primarily uses only PARAMS because the data are read onto disk or written onto tape using existing TAPEIN and TAPE-OUT programs and the DOIO routine handles the use of DISKIO and IPCSI. This allows new software to be added easily to the system although in a few cases a particular algorithm might not lend itself to the DO-ALL concept, (such as certain geometry and spatial-filtering techniques). In these cases, the programmer would also have to use DISKIO and IPCSI, which can easily be done with FORTRAN calls.

DOIO is designed to handle all the basic software requirements of initialization, disk I/O handling, and reading input and output file specifications from the user. Use of DOIO in program development is one mechanism to increase the standardization of spatial processing routines.

It is necessary to write two interfacing routines to use DOIO: (1) a short (8-15 statements) main program which defines the dimensions of the input and output buffers, sets the number of input and output buffers needed, and sets a few processing flags, and (2) a user subroutine which manipulates and processes the spatial data passed to it by DOIO, that is, do anything the user wants (hence the name DO-ALL). DOIO will automatically set acceptable defaults on various processing parameters which the programmer does not usually have to be concerned with in terms of how the files will be accessed. The default values setup by DOIO can be changed in the user subroutine.

During the past year, modifications have been made to the MIPS software to improve the internal and external documentation and to make it a user-friendly system. All programs and subroutines have been modified so that they have a standard documentation at the beginning of the source code. A program can then be used to extract any or all of the documentation from each routine to create a programmer's and (or) a user's manual. Also, to make the system more user-friendly, each executable program in MIPS has been modified to prompt the user for processing parameters with a short description of each of the keywords available in the program. Prompting is performed by using a single call to a FORTRAN routine, which formats the terminal screen and defines the required keywords to the user. Because the same routine is used in all the programs, prompting is consistent and easy to interpret. Keyword descriptions are located in a separate disk file so that memory overhead is not introduced in the program, and the descriptions can be modified or changed on disk without having to recompile or task-build the program (programs are not affected unless new keywords are added).

The prompting routine is a separate task, with its own memory partition, and is executed from the calling program. Once the screen is formatted, the processing parameters given, and the error checking completed, the prompting task will exit from its memory partition and the calling program will continue to execute. This technique differs from other user interface routines (such as application executives); instead of having a program continually running and monitoring the terminal for user requests, MIPS interfaces with routines that are executed as separate tasks from the calling programs; these routines disappear once they are completed. Use of this type of interface helps minimize memory overhead and CPU time required.

The operating system used on MIPS is Digital Equipment Corporation's RSX-11M. It is a multiuser system that optimizes computer utilization if multiple users are logged into the system (currently there are three terminals on MIPS; see figure 1). RSX-11M provides quick response time and is one of the faster multiuser operating systems. Some of the features of RSX-11M are: (1) it maintains an error logging subsystem for detecting and logging both hard and soft errors for I/O devices; (2) it provides complete and informative error messages to the user; and (3) it is compatible with VAX/VMS.

Long-Term Support

Most of the MIPS hardware components are off-the-shelf and have good maintenance support and have proven reliable in the field. The PDP 11/23, disk drives, and terminals are covered by the same maintenance contract as

FIPS, and hardware engineers are available within a one-day notice. The operating system (RSX-11M) used by MIPS has been available for several years, is well debugged, and has available software support.

The FIPS/MIPS application software was designed and written by Survey employees in Flagstaff and is well understood; this makes it easy to modify or add software to the system. Also, the software package can be used more efficiently because the interaction between all the different components is well understood. The software changes that have recently been made to improve MIPS user friendliness were also made by Survey employees, and these features will be easy to incorporate into new software.

Modularity

From both a hardware and software point of view, MIPS is a very modular system. The hardware is composed of off-the-shelf components, and MIPS can be reconfigured to accommodate new components. MIPS hardware can be separated into three major parts: (1) a CPU unit plus disk drives and terminals, (2) a TV display system with its own refresh memory, and (3) a tape drive. Figure 1 shows the current configuration of MIPS and figure 2 shows an upgraded version of MIPS. Either one can accommodate more memory, disk storage, or refresh memory.

The MIPS application software is composed of independent programs that can be combined in any sequence to produce the desired results. The DO-ALL concept allows optimization of data handling and minimizes the amount of code and system level instructions needed by the programmer. With DO-ALL, it is easy to write, test, and optimize new application programs by working with only subroutines that apply the desired algorithm; the work is done in a modular form. The new screen formatting and prompting routines are all FORTRAN subroutines, and new software can easily incorporate them as part of their code so that the same level of user friendliness is present and the prompting format remains consistent. Also, the assembler code used to handle the I/O and user interface are separate subroutines (DISKIO, TAPEIO, PARAMS, IPCSI) and can be used in new programs with FORTRAN calls.

Software Transportability

Even though the entire software package cannot be easily transported to a different family of hardware, it could be done with a low to medium amount of programming effort. By using the DO-ALL concept in MIPS, most of the machine dependent code is isolated in a few program modules. These modules include programs that have software written in assembler language to optimize the processing speed, and the I/O handling and user interface subroutines (DISKIO, TAPEIO, PARAMS, IPCSI). Once these are replaced with equivalent routines on the new system, the conversion should be straightforward since the subroutines containing the majority of the algorithms are written in FORTRAN and require only recompilation.

The MIPS package can be transported with a minimum amount of effort if the same family of hardware is used. For example, new programs developed on MIPS (11/23 with RSX-11M) can be transferred to FIPS (11/44 with

RSX-11M) in a matter of minutes. No changes are required to either the FORTRAN or assembler code; all that is needed is to compile and task build the source code, which is transferred by phone, on FIPS.

Expandability

A system often is designed on paper three or more years before it becomes operational and in the interim new hardware options become available by the time the system is in use. With the MIPS design, the hardware can be easily expanded or changed to include new hardware components. For example, figure 2 shows the configuration of an updated MIPS configuration, which incorporates the new PDP 11/73 board and more disk storage. The PDP 11/73 board is advertised as a replacement/upgrade for PDP 11/23 systems and is supposed to have the processing power of a PDP 11/70 or 11/44, running four to seven times faster than the PDP 11/23. Since the PDP 11/44 runs as fast as the VAX-11/750, the PDP 11/73 will give MIPS the speed of a VAX-11/750 for most spatial processing functions. The changeover of MIPS to a PDP 11/73 requires only a board replacement and costs about $2,800. The PDP 11/73 allows up to 4 megabytes of memory to be attached to the system and, if desired, it can run the UNIX operating system.

As shown in figure 2, the TV display system can be expanded for increased resolution. Also, an array processor can be added to the system if very fast processing speed is required. Often an array processor is used with larger mainframes to execute the algorithms requiring faster speeds, and the host serves only as an I/O handling device between the array processor and the disk drive. Using a large mainframe as an entry and output port for an array processor is a waste of computer power if a smaller system can serve as the host for the array processor.

Because RSX-11M can be emulated on a VAX running VMS, the MIPS software package should be able to run on a VAX without a large conversion effort. Therefore, the option exists to use MIPS on a wide range of CPU sizes and configurations, depending on the size of facility it serves (that is, office environment with one to three users or a computer center with many users).

The expansion capability of the MIPS application software is very good. It is easy to add new software by using the DO-ALL concept, and new programs can easily be integrated into the system because most of the tasks are independent of each other. Even with those few algorithms that are unable to make efficient use of the DO-ALL concept, they can be programmed without a major problem because of the existing subroutines available, such as DISKIO, TAPEIO, PARAMS, IPCSI, plus screen formatting and prompting routines.

SUMMARY

An image processing system has been developed that is powerful enough to efficiently execute the same functions as larger systems, but do so as a stand-alone system. The hardware was selected so that it would be compatible with an existing larger image processing system in Flagstaff.

The MIPS software package has been designed to allow optimization of the processing speed with a minimum use of assembler code. The I/O handling tasks are coded in assembler and are easy to use with FORTRAN calls. The use of the DO-ALL concept has minimized the amount of code needed to generate new software and has made the software more transportable. The program documentation, screen formatting, and user-prompting modifications recently made to MIPS have greatly improved the user friendliness of the system. Because of the modular aspects of the system, it is easy to incorporate these capabilites into new software.

MIPS has over 80 man-years of software available to it through FIPS, providing an excellent range of functional capabilites. The software base and design make it a very efficient system. Optimization has been done wherever possible without writing the entire software package in assembler language. Because the hardware consists of off-the-shelf components and the software has been written and is constantly used by the USGS, the system has the basis for good long-term support. The use of the DO-ALL concept and the fact that all the critical assembler or FORTRAN routines needed by the programmer to interface with the system are written as separate subroutines, make the system very modular and expandable. Also, the software package can be transported easily within the same family of hardware and, with a low to medium level of programming effort, to a different family of hardware.

Distributed processing is becoming the current trend because of reduced hardware cost and the increased capabilities of small systems. Several other small image processing systems were developed during the same time frame as MIPS. A comparison of a few of these systems is made by Welch and others (1983). Two other systems, McIDAS and MIDAS, are described in papers by Suomi and others (1983) and Hofman and others (1983), respectively.

ACKNOWLEDGMENTS

During the past 11 years, there have been many people involved in developing FIPS, from which MIPS was derived. There are a few individuals who deserve special credit because of their critical involvement in developing the system. Laurence Soderblom initiated the image processing project in 1971, and his ideas and efforts have helped make FIPS a very good processing system. Dennis McMacken and Alex Acosta are responsible for most of the assembler code, which has made the system so efficient. Eric Eliason wrote the current DOIO subroutine that is used in the DO-ALL concept and, along with Patricia Eliason and Kathleen Edwards, has helped write some of the application software used in the system. Steve Wolfe, Jeff Anderson, and Jody Mack helped bring up MIPS and have worked on the recent modifications to document the software and to make MIPS a much more user-friendly system. The author would also like to thank Lennis Berlin and Eric Eliason for their helpful review of this manuscript.

REFERENCES

Hofman, L. B., Erickson, W. K., and Donovan, W. E., 1983, MIDAS: A microcomputer-based image display and analysis system with full Landsat frame processing capabilities: Proceedings, Seventeenth International Symposium on Remote Sensing of Environment, Ann Arbor, Mich., May 9-13, 1983, 16 p.

Suomi, V. E., Fox, R., Limaye, S. S., and Smith, W. L., 1983, McIDAS III: A modern interactive data access and analysis system: Journal of Applied Meteorology, May 1983, p. 9.

Welch, R. A., Jordan, T. R., and Usery E. L., 1983, Microcomputers in the mapping sciences: Computer Graphics World, v. 6, no. 2, February 1983, 7 p.