NOAA TECHNICAL MEMORANDUM NWS AR-41

CONCURRENT SEISMIC DATA ACQUISITION AND PROCESSING USING A SINGLE IBM PS/2 COMPUTER

W. O'Neill Zitek, A.H. Medberry, and T.J. Sokolowski
Alaska Tsunami Warning Center
Palmer, Alaska

National Weather Service, Regional Headquarters
Anchorage, Alaska
October 1990

noaa NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION / National Weather Service

NOAA TECHNICAL MEMORANDA
National Weather Service, Alaska Region Subseries

The Technical Memoranda series provides an informal medium for the documentation and quick dissemination of results not appropriate, or not yet ready, for formal publication in the standard journals. The series is used to report on work in progress, to describe technical procedures and practices, or to report on investigations devoted primarily to regional and local problems of interest mainly to Alaska Region personnel and, hence, will not be widely distributed.

Papers 1 -3 are in the former series, ESSA Technical Memoranda, Weather Bureau Technical Memoranda (WBTM); and papers 4-38 are part of the series, NOAA Technical Memoranda NWS. All papers are available from the National Weather Service Alaska Region, Scientific Services, Box 23, 222 W. 7th Ave, Anchorage, Alaska 99513.

NWS AR-01 Freeze-Thaw Cycle in the Coastal Artic of Alaska. Harold W. Searby. October, 1968.

NWS AR-02 Climate Along a Pipeline From the Arctic to the Gulf of Alaska. Harold W. Searby. December, 1968.

NWS AR-03 Coastal Weather and Marine Data Summary for Gulf of Alaska, Cape Spencer Westward to Kodiak Island. Harold W. Searby. May, 1969.

NWS AR-04 Climate of the North Slope, Alaska. Harold W. Searby and Marcelle Hunter. February, 1971.

NWS AR-05 Forecasting Ice in Cook Inlet, Alaska. Richard J. Hutcheon. August, 1972.

NWS AR-06 Sea Ice Conditions in the Cook Inlet, Alaska During the 1969-70 Winter. Richard J. Hutcheon. September, 1972.

NWS AR-07 Sea Ice Conditions in the Cook Inlet, Alaska During the 1970-71 Winter. Richard J. Hutcheon. October, 1972.

NWS AR-08 Sea Ice Conditions in the Cook Inlet, Alaska During the 1971-72 Winter. Richard J. Hutcheon. April, 1973.

NWS AR-09 Obtaining the Height of the Freezing Level from the Temperatures Aloft Forecasts (FDAK). Gordon D. Kilday. March 1974.

NWS AR-10 Mean Monthly and Annual Precipitation - Alaska. Gordon D. Kilday. March, 1974.

NOAA TECHNICAL MEMORANDUM NWS AR-41

CONCURRENT SEISMIC DATA ACQUISITION AND PROCESSING USING A SINGLE IBM PS/2 COMPUTER

W. O'Neil, A.H. Medberry, and T.J. Sokolowski
Alaska Tsunami Warning Center
Palmer, Alaska

National Weather Service Alaska Region
Environmental and Scientific Services Division
Anchorage, Alaska

October 1990

UNITED STATES DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary

National Oceanic and Atmospheric Administration
John A. Knauss, Under Secretary

National Weather Service
Elbert W. Friday, Jr., Assistant Administrator

Notice

Contribution No. 003-90 from NOAA/Alaska Tsunami Warning Center

Table of Contents

List of Figures

Concurrent Seismic Data Acquisition And Processing Using a Single IBM PS/2 Computer

W. O'Neill Zitek, A.H. Medbery, and T.J. Sokolowski

ABSTRACT. The Alaska Tsunami Warning Center's (ATWC's) direction in automation is toward a distributed system of microcomputers performing tasks in many different areas, one being seismic data acquisition with concurrent processing. A major limitation of seismic data acquisition and analysis has always been the expense, complexity, and inflexibility of available computer systems. This paper describes a relatively inexpensive microcomputer system that continuously acquires at least 32 channels of short period seismic data at 50 samples per second. The system is capable of concurrent processing during data acquisition. The acquisition programs use an IBM PS/2 computer with the OS/2 operating system. As OS/2 provides for real-time programs and true multi-tasking, the computer retains all the usefulness of a small computer system. One program automatically copies incoming data to a temporary buffer and also currently stores about 24 hours of data on a hard disk. The other program performs data display and invokes many other computational programs. It allows the user to easily select data, modify scales, make P-picks (first arrivals), measure periods and amplitudes, and calculate earthquake epicenters, magnitudes, and other parameters for use at the ATWC. We believe this to be an improvement over current microcomputer technology in acquiring and processing many channels of seismic data.

1. Introduction

The National Weather Service's Alaska Tsunami Warning Center (ATWC) has been improving the tsunami services through automation for more than 10 years. The direction of this automation, using distributed microcomputer systems, is given by Sokolowski (March, 1990) and Sokolowski (October, 1990). The seismic part of the automation efforts have resulted in a highly automatic system which detects, locates, and sizes moderate and major earthquakes on-line and in real-time (Sokolowski et al., 1990). A degraded backup to the real-time computer system is an IBM PC/AT microcomputer which requires manual reading and recording of times of onset of P arrivals (P-picks) which are recorded on helicorders and develocorder(s). The data are interactively entered into the PC/AT for earthquake parameter computations. These manual processes are time consuming, inaccurate, and prone to error. The details of the above procedures and processes using helicorder and develocorder data are given by Sokolowski (1983).

The well-known Teledyne Geotech develocorders were built during the mid 60's and are obsolete. Each develocorder can record a maximum of 20 channels of seismic data which are archived on 16mm film. It has limited usefulness and is costly, and time consuming to repair and maintain. Furthermore, it takes at least 10 minutes, after recording an event, before the data can be viewed or used in other computational programs. The develocorder data, in addition to being used during earthquake-tsunami investigations, are also used to collect routine data that are disseminated to the National Earthquake Information Center in Colorado; USGS-Menlo Park Research Center in California; University of Alaska's Geophysical Institute in Fairbanks; and agencies in Canada. This paper discusses our efforts and accomplishments in successfully implementing a seismic data acquisition system with concurrent processing, using a single IBM PS/2 micro computer.

2. Goals And Efforts

In 1989, funding was made available for: 1. enhancing the degraded IBM PC/AT backup system and, 2. replacing obsolete develocorders. The initial design goals required a relatively inexpensive microcomputer system to:

- Continuously acquire 32 (or more) channels of seismic data at fifty samples per second.
- Store at least the previous 24 hours of data.
- Display any portion of the data.
- Scroll through the data rapidly.
- Be able to make P-picks and compute earthquake locations and magnitudes concurrently during data acquisition.
- Do all of the above in near-real time using a single machine.
- Have growth and enhancement potential.

In a survey of the literature for existing microcomputer based systems (Lee et al., 1988; McNutt, 1986; Kinemetrics Seismological Instruments, 1986), we found that these systems were too expensive; required more than one computer to concurrently acquire and process seismic data; lacked growth potential; and/or required considerable software modifications and hardware additions to meet our immediate and routine processing needs in the Tsunami Warning System (TWS). This necessitated developing a new microcomputer system to meet our needs. An IBM 386 microcomputer system was selected. Since most of our technique developments are done in-house, this system also provided the ATWC with an opportunity to better determine the extent to which these type of micros could be integrated into the ATWC automation plan, and the availability of third party hardware and software components. However, it also provided many challenges concerning a multi-tasking operating system required to handle the demands of real-time data acquisition and processing; hardware and software for acquiring data; and a sufficiently powerful computer for acquiring, displaying, and processing the enormous amount of data that arrives without pause.

After obtaining an IBM PS/2 Model 80 computer, we ran benchmarks to compare it to our Digital Electronics Corporation (DEC) MicroVax II which is currently being used to process real-time seismic data. The IBM performed admirably at CPU and disk intensive tasks, meeting or surpassing the DEC. The next choice was an operating system. MS/DOS, as popular as it is, is incapable of true multiprocessing and does not support more than 640K bytes of memory, unacceptable in a system that is processing 192K bytes per minute.

The choice came down to AIX (IBM's version of UNIX) or Microsoft's OS/2. OS/2 had several advantages. More than ten years younger than UNIX, it has the best features of UNIX and DOS and a number of enhancements. One of these is that a process can be granted time-critical priority. A time-critical process is essentially guaranteed to get CPU time, a requirement of real-time data acquisition systems. This means that a time-critical process, which cannot afford to get behind, will not, no matter what other burdens are placed on the system. The other major feature is OS/2's implementation of threads. Programs running under OS/2 can be multi-threaded where each thread (a little program itself) is self contained in a single program and the subroutine it calls. These "miniprograms" execute concurrently. Under OS/2, separate threads can be assigned to tasks that do not necessarily occur in a known sequence. Since system resources are controlled at the process level, each thread can share the same resources (file handles, semaphores, global variables, etc.) as can any other thread in the same process.

The major drawback to OS/2 is that it is so new that there is little documentation, little support by OEMs, and no shortage of "bugs"; common to new operating systems.

3. Data Acquisition Hardware

We began the project with an IBM Model 70 PS/2 computer until a model 80 was available. It features an Intel 80386 central processor running at 20 MHz, a 115 Mbyte hard disk, and three expansion slots. Upon arrival of an IBM model 80 PS/2 and after the development of the device driver, the project development moved forward. This micro has a 315 Mbyte hard disk, 8 Mbytes RAM memory, a 20MHz CPU, and 7 expansion slots. For the display, we had an 8514 video display terminal. Instead of using the PS/2 default VGA output, with a resolution of 640x480 pixels and sixteen colors, we used the 8514A card, which is faster and has a resolution of 1024x768 pixels and 256 colors. Figure 1 shows a block diagram of the hardware configuration for a PS/2 micro computer system.

To bring in seismic data, we needed an analog/digital (A/D) converter that was compatible with the IBM MicroChannel bus architecture. We selected a product of Data Translation, the DT2901, which is designed to do high-speed A/D conversion (50 KHz throughput) and transfer of the data to RAM of the PS/2. This card can interface sixteen channels of analog input, sixteen channels of analog output, and 16 bits of direct digital input-output. The analog data are connected to isolation amplifiers which perform two functions. They reduce the +-10 volt range of the input signals to +- 5 volts, the maximum range the A/D card can accept. They also serve to isolate the connection to the A/D card to avoid any interference with the other connections to those channels, i.e., our DEC MicroVax II, the Helicorders, and the Develocorder. We also use the digital I/O channels to obtain 16 bits of time code from a satellite clock.

Hardware Overview

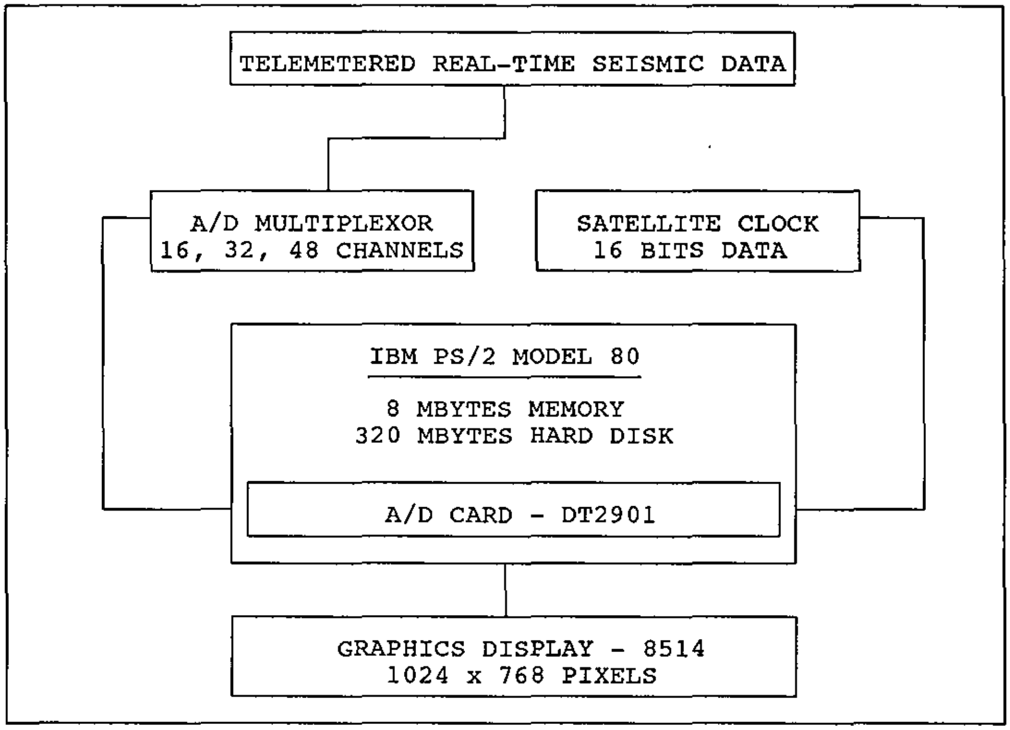


Fig. 1. Block diagram showing the hardware overview. The A/D card (DT2901) receives 16 channelsles of seismic data (by multiplexing the input, multiples of 16 channels can be acquired) and 16 bits of satellite clock time. The IBM PS/2 model 80 computer, which contains the A/D card, has 8 Mbytes of memory and a 315 Mbyte hard disk. Data are displayed on a 16 inch color monitor with a resolution of 1024 by 768 pixels.

After the initial success in acquiring and displaying 16 channels of data, we attempted to acquire 32 which was accomplished by the addition of another 16 channel A/D card. However, an additional card required 2 more direct memory access (DMA) channels (computer has 15 which are used by hard disks, floppies, additional memory, and the other A/D card), and we were still limited to a maximum of 32 channels. This maximum was breached by a creative scheme to modify the DT2901 card to output the "Channel Scan Done" signal used on the card. In a cooperative way, an external multiplexing card uses this signal to switch between sets of 16 channels. This resulted in the capability to acquire 16, 32, 48, or more channels of seismic data.

4. Computer Languages

Since we have been automating seismic data acquisition and analysis for many years, the ATWC has developed numerous techniques and algorithms that are coded into FORTRAN programs (Sokolowski et al., 1990). This project was a new experience in attempting to use hardware for which software had not been written to cope with the intricacies of interacting with hardware (e.g. software driver) and a brand new operating system.

Except for the driver, all the other programs were written in Microsoft C 5.1. The major programs were all multi-threaded. FORTRAN programs cannot be multi-threaded at this time. At the present time, this version limits the number of threads per program to 32. Future versions of C will not have this limitation. Also, operating system calls were easiest to make in C (common in minis, micros, and PCs). All of the programs use dynamic memory allocation, which is also unavailable to FORTRAN. The operating system favors the C language and uses about 2.5 Mbytes of RAM and about 25 Mbytes of disk space. Figure 2 shows an overview of the acquisition, display, and analysis software.

5. Device Driver

The first hurdle was to write a device driver to allow programs to communicate with the A/D card. The device driver was written in Microsoft's Macro Assembly. OS/2 requires that low-level device drivers act as well-behaved subroutines that are treated as part of the operating system. No compiler is presently capable of translating other languages into the appropriate machine code. Perhaps because the operating system is so new, no A/D card manufacturer presently supplies a device driver for OS/2. Under OS/2, the device driver is a type of program that is loaded when the operating system is booted. The driver becomes part of the operating system. For this reason, it must be very well behaved. Drivers are very difficult programs to write, as they are not allowed to use most of the features of more normal programs.

ACQUISITION, DISPLAY, AND ANALYSIS SOFTWARE OVERVIEW

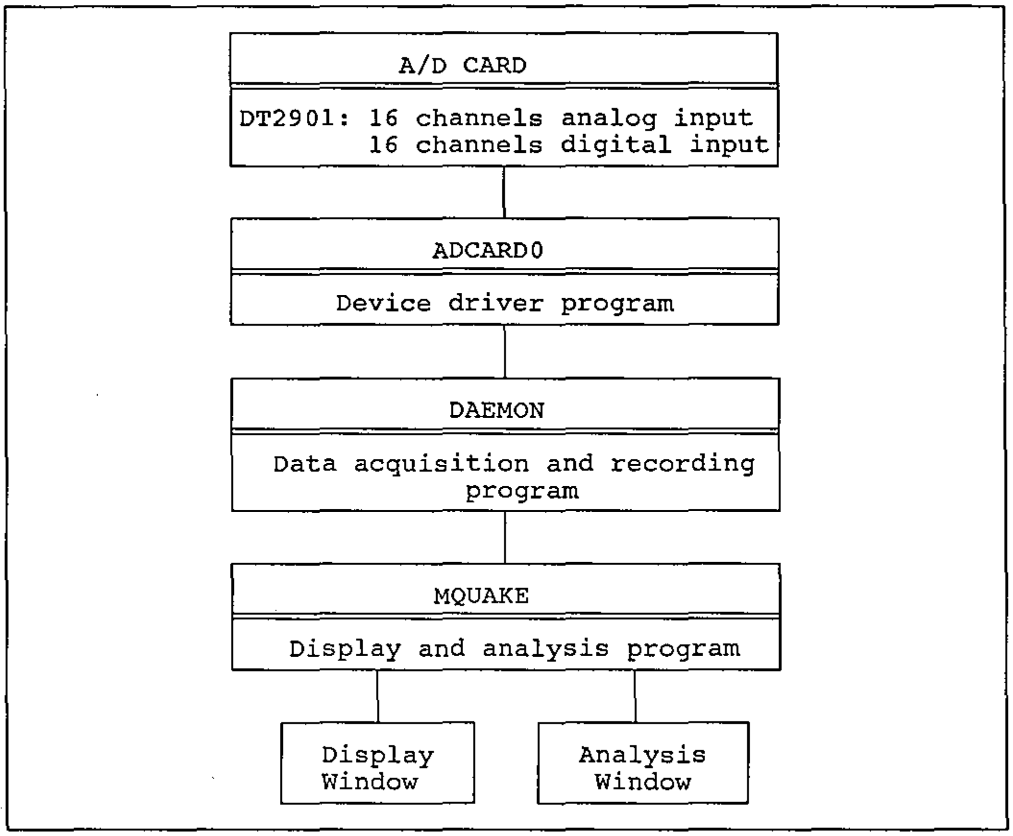


Fig. 2. Block diagram showing the software overview. The device driver (ADCARD0) is loaded when the operating system boots. After the system comes up, the DAEMON is activated. It determines data acquisition parameters, begins the acquisition process, then copies the data to memory and hard disk. MQUAKE, which can be executed or terminated at any time, communicates with the DAEMON to request data for display and analysis.

If the driver fails, the system crashes. They cannot write to the screen or read from the keyboard. They are not allowed access to system functions or libraries. They must be written in Assembly. They are called by the operating system to interpret requests from programs. The driver handles all communications with the device. If the device produces a hardware interrupt, the operating system demands that the driver fix the problem. If it does not, the system crashes.

As Data Translation was unwilling or unable to write a device driver for their card under OS/2, we did. The driver configures the card, telling it how many channels of data to obtain, what conversion rate to use, tells the computer's direct memory access (DMA) controller where to put the data and how much to put there. Since we want to know when data are obtained, the driver assigns two memory buffers, between which the device (and DMA controller) alternates. When one buffer is full, the driver lets the calling program know that data are available. Since the driver does not know exactly when that data will become available, it allows the A/D card to use hardware interrupts to tell it.

6. THE DAEMON

The DAEMON is the program that communicates with the driver. It allocates the small buffers that the device uses, a very large buffer for temporary RAM storage of data, writes the data to hard disk, and provides the data to any program that asks for it. It also talks with the driver to obtain the correct time from the satellite clock, sets the system time, and keeps track of any drift between the two. We call it a DAEMON because it also does not communicate with either the screen or the keyboard. Its job is to guarantee that data from the device is acquired, stored, and available. If the device driver fails its job, the system crashes. If the DAEMON fails, we have lost data. It must be fail-safe. To perform its job, the DAEMON executes six threads, all of which are time-critical. The threads, although part of the DAEMON process, are treated by the operating system as separate entities, in that each runs independently of the others. Each thread gets its own share of CPU time. The threads also inherit file handles, memory, and other resources allocated by the original DAEMON thread 1 which is shown in Fig. 3. Further details of the DAEMON threads are given in Appendix A. To a programmer, a thread looks like a subroutine. They run, however, asynchronously. Since some of the threads must be able to communicate with each other, they use semaphores. A semaphore is a type of flag that has particular meaning to the operating system. It is either set(1) or clear(0). A thread waiting on a set semaphore is not allotted CPU time until that semaphore becomes clear. Semaphores are also used to protect critical resources, such as memory.

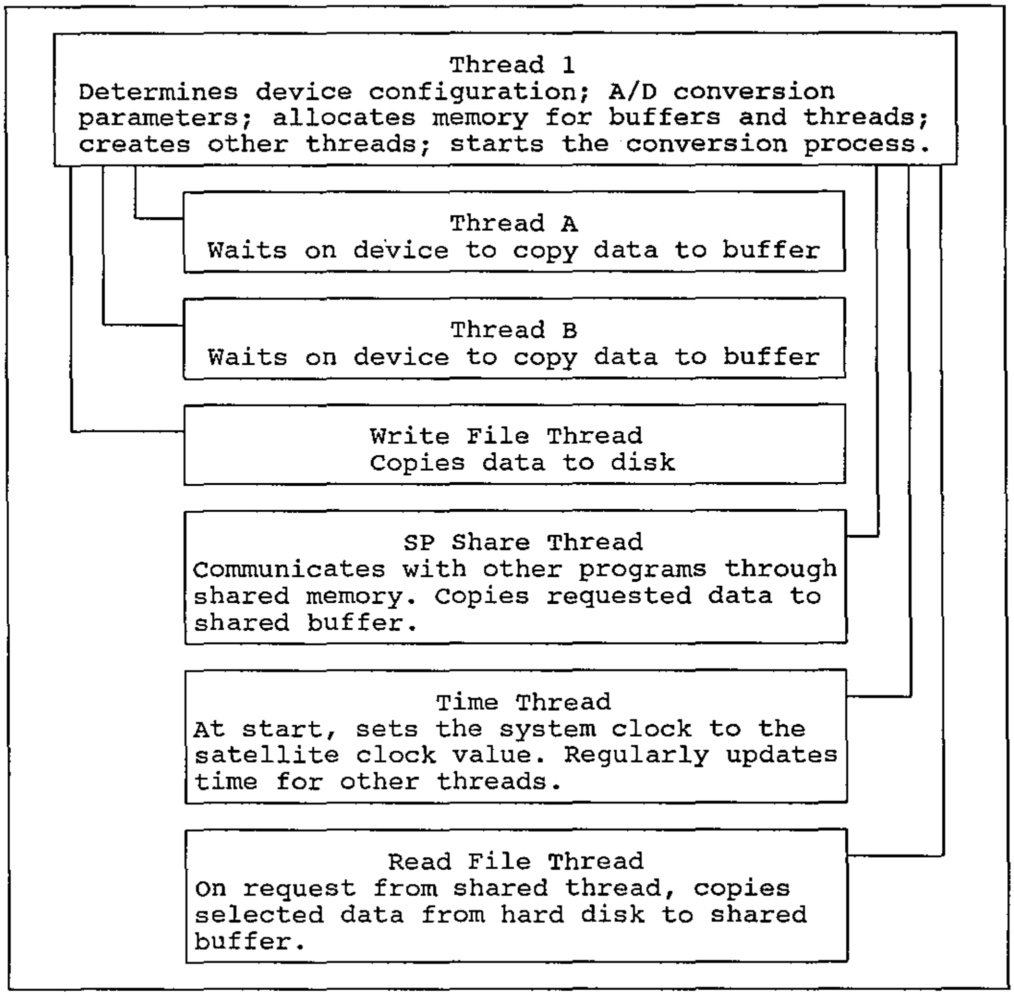


Fig. 3. Block diagram showing the DAEMON and its threads. Although the threads are part of the DAEMON process, they are treated by the operating system as separate entities which execute independently. Each thread gets its share of CPU time and has access to other resources allocated by the DAEMON (Thread 1). Threads communicate with each other via semaphores which are either set(1) or clear(0).

This program is designed to run continuously, acquiring data through the device driver, copying the data to memory and hard disk, and responding to other programs that might request some of the data. The DAEMON keeps track of how much space is left on the hard disk, and erases the oldest data when space is short. The device driver does not know that more than 16 channels are multiplexed to the A/D card. The DAEMON uses information from a startup file to cope with the multiplexing.

7. MQUAKE

The Presentation Manager graphics program displays seismic data and allows a variety of manipulations. It is just one of many programs that a user can interact with merely by using the mouse or the keyboard. As a Presentation Manager program, MQUAKE employs "windows". These are screen displays using bit-mapped characters and high-resolution graphics to provide for pop-up and pull-down menus, dialog screens, and system control panels. MQUAKE itself has two primary screens, the quake display window and the quake solution window. The user toggles between these two separate but equal windows from the keyboard or with a mouse. In the case of the display and analysis program, MQUAKE, many existing ATWC FORTRAN programs (algorithms) were converted to C. These important techniques and algorithms were developed and used at the ATWC over many years.

When the program is started, it communicates with the DAEMON (which executes continuously) to determine the characteristics of the incoming data, e.g., number and name of seismic channels. It also reads a system file that maintains a list of user-set defaults that include normal scaling and trace order. Besides the two primary windows, an additional separate thread is created specifically to request data from the DAEMON. Because of this separate thread, the user can continue to use the menus while data are being copied.

The quake display portion has a number of features:

- The screen displays fifteen or more seismic traces at once.
- Control of vertical and horizontal scaling of the seismic traces via a menu.
- The User can obtain the most recent data (less than 3 seconds old), scroll through the data with scrollbars, or select data from a specific time via a menu.
- The user can scroll vertically to display stations not presently shown.
- The user can turn on or off stations that are clipped or noisy.
- Use a mouse to make P-picks or modifications to P-picks.
- Determine amplitudes and period for magnitude calculations.
- Plot the data on an external printer/plotter.

In retrieval of data, the application does little work. It makes a request, and it is the DAEMON's job to supply the data. This process works very well. The

DAEMON can find the correct data from a day's worth (about 270 Megabytes/day) in about 2 seconds. MQUAKE's job is to demultiplex the data and perform the display. The program is dedicated to the user, making changes in the display format or data as requested.

After picking first arrivals and magnitude (amplitude and frequency) information from the display, the user toggles into the quake solution portion of the program. The quake solution window displays the P-picks, and provides a number of buttons for the user to select. Once a solution is determined, one panel displays the epicentral origin time, latitude, longitude and depth. It also displays the distance from the epicenter to the nearest major cities or towns. A fourth panel uses the magnitude information to compute Ml and mb magnitudes for the event. Throughout the quake solution process, the user has a number of options available from on-screen buttons and the following pull-down menus:

- Search for bad P picks.
- Choose a method for determining initial location.
- Eliminate questionable P-picks temporarily or permanently.
- Add other P-picks interactively.
- Add other magnitude data for MS computations.
- Fix or float an event's depth.
- Add comments to the file that MQUAKE keeps for events.
- Use pP-P time to calculate depth.
- Choose to immediately copy solution data to the printer.

The process of picking and locating an event is so simple and quick that it is common for a user to ping-pong between the display and solution windows. For the program's most common role of determining data to be sent to various recipients, a geophysicist sets a menu option. When a pick is made, the program pops up a dialog box asking what phase was picked. All P-picks and magnitude information are saved so that a seismic information teletypewriter message can be quickly generated. In the case of an earthquake that requires an official ATWC response, the user can immediately transfer to the interactive program and generate appropriate watch-warning-information messages to transmit via teletype or satellite.

Appendices B and C show outputs of screen displays of seismic signatures from both a local and teleseismic earthquake, and their respective solution windows. In addition to the solution windows showing the earthquake's data and computed parameters, they also show the various processing options that are available to a user.

Although we are impressed with the performance of the Model 80, we do not yet know its upper limit in data acquisition, plus additional automatic earthquake analysis. This will become more evident in the future. We have already reached the conclusion that 315 Mbytes of hard disk storage are not enough for a machine that is acquiring 270 Mbytes of data each day. Additional mass storage hardware will correct this problem in the future.

8. Conclusion

This project has been very successful in the replacement of an IBM PC/AT backup system and obsolete develocorders. Although the model 80 became operational in June, 1990, enhancements and modifications will continue in the future. The initial design goal's criteria have been met and exceeded in acquiring, storing, and displaying data, plus we still have the ability to do concurrent processing using a single PS/2 micro computer. This development has demonstrated that this type of micro computer can be used in the ATWC planned distribution of micro computer systems. We believe that this accomplishment is an advancement over present day microcomputer technology and represents state-of-the-art in this area of microcomputer applications.

9. Acknowledgments

The authors thank the National Weather Service Headquarters and the Alaska Regional Headquarters (Mr. Burton D. Goldenberg, Acting Regional Director) for their continued support in our efforts to automate the Alaska Tsunami Warning Center. The authors also thank those many past and present ATWC staff members who have made contributions over the years that are used in part in this project. We also thank Mr. Stuart G. Bigler and Mr. Harry S. Hassel (former Directors of the Alaska Region) for their past support in this project which is now being realized. Lastly, the authors thank Ms. Nancy J. Clarke, Mr. Arthur J. Johnson, and Ms. Susan P. Davis for their help in the preparation of this article for publication.

10. References

Lee, W.H., Tottingham, D.M., and Ellis, J.O., 1988. A PC-based seismic data acquisition and processing system, U.S. Geological Survey, Open-File Report 88-751, PP 31.

McNutt, S., 1986. NEWT system (seismic analysis system), California Geology, P. 51-55.

Kinemetrics Seismological Instruments, 1986. DataSeis II, Digital 16-bit multi-channel acquisition system / Visual event recording system, Kinemetrics Brochure, PP. 10.

Sokolowski, T.J., 1983. Automation highlights of the Pacific and Alaska Tsunami Warning Centers, Tsunami Newsletter, December 1983, V. XVI, No. 2, P. 1-9.

Sokolowski, T.J., March 1990. The Alaska Tsunami Warning Center, NOAA Technical Memorandum NWS AR-38, PP. 27.

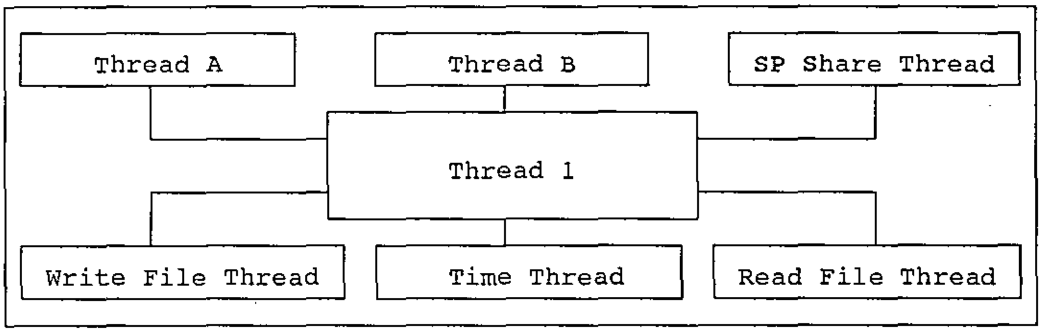
Sokolowski, T.J., Whitmore, P.M., and Jorgensen, W.J., May 1990. The ATWC's automatic and interactive computer processing system, NOAA Technical Memorandum NWS AR-39, PP. 23.

Sokolowski, T.J., Whitmore, P.M., Jorgensen, W.J., Medbery, A.H., October 1990. Expert system in Tsunami Hazard Mitigation. Presented at the International Symposium on Marine Positionin, Miami, Florida. (In publication.)

APPENDIX A

THE DAEMON AND ITS THREADS

THE DAEMON AND ITS THREADS



Thread 1

Thread 1 is the central thread of the DAEMON. Although its work is done in a millisecond, it sets the stage for the other threads. It performs all of the necessary initialization, determines that the A/D device is properly configured, and that sufficient memory is available for the entire program to run.

It begins by reading the DAEMON.INI file to discover what type of data are to be acquired: seismic (data intensive, requiring direct memory access and hardware interrupts) or tide (a more leisurely process that can be satisfied by programmed I/O and a timer); the sample rate; whether to use an external clock; the number of channels of data to acquire; if the input channels are multiplexed (32, 48, or 64 instead of the normal 16); how much system memory and disk space to use; and of course, the channel names.

Next, it "opens" the device (this is actually handled by the device driver). The DAEMON communicates with the device through generic IOCTL commands (input/output control). If seismic data are acquired, it checks with the device driver to ensure that the system has allocated DMA channels. It asks the operating system for memory for a variety of resources:

- The device needs 2 buffers of 64K bytes or less, one for each DMA channel. When acquiring data continuously, the driver will automatically cause the device to alternate between these buffers.
- A huge buffer (the OS/2 convention for memory greater than one 64K byte segment) used by the DAEMON for keeping seismic data in memory.
- Two shared buffers, one for data and one for information about the data. Both can be shared with another program (such as MQUAKE).
- A variety of buffers used by the threads for copying data, informational structures, etc.
- Stack space for the other threads.

Thread 1 passes the data acquisition parameters to the device driver, creates all the threads, and resets the priorities of some threads to time critical. After waiting for the Time Thread to set the system clock, it starts the A/D conversion process, then sleeps forever.

Threads A and B

These identical threads alternate in waiting on the device driver (ADCARD0) until one of the small buffers used by the DT2901 are filled. When reawakened, that thread (either A or B) obtains the current time, then calls the function DataToHugeBuffer. DataToHugeBuffer does the work of copying data from the small temporary buffer to a huge (>64K byte) circular buffer. When this task is complete, the thread makes a call to the device driver, which blocks the thread from running, (i.e., it will be granted no CPU time) until more data are available.

SP Share Thread

The SP Share Thread waits on a shared semaphore for requests from an external program (such as MQUAKE) for data. When a request is received, the thread checks the value of dTimeRequested. If the value is -1.0, the thread knows the request is for the most current data. In this case, the work is done, as copying of the "most current" data is performed by the DataToHugeBuffer procedure (when called by Threads A and B). In other cases, the Share Thread examines the structure that describes the contents of the 64K segments of the huge buffer (remember that the huge buffer is circular). If the time requested is earlier than any of the data in the huge buffer, the thread clears the semaphore for Read File Thread, and lets that thread look for the data in the seismic data files. Otherwise, the thread requests the semaphore for the appropriate segment. If the segment is in use (by DataToHugeBuffer), the thread returns a message to the requesting program that says, in effect, "busy now, try later". If the segment is not in use, the semaphore is set, and SP Share Thread calculates where the appropriate data are, and copies it to the semaphore protecting the segment. It then clears the DataAvailable semaphore that the external program has been waiting on.

Write File Thread

The Write File Thread has two functions. It copies data from the huge buffer to separate files and keeps a structure that describes what data are in which file. The thread is awakened by the DataToHugeBuffer function each time that function has filled a 64K byte segment, either halfway (32K bytes) or completely.

This thread ensures that each file is properly filled (to ~1 Mbyte) and that the number of files on disk do not exceed the MaxNumberOfFiles limit set in the DAEMON.INI file.

When its semaphore is cleared,it checks that a seismic file is open. If not, it opens one. The file names used denote the starting time of the data, e.g., s1852012.s90 (which stands for Seismic day 185, time 2012 minutes, short period, year 90). When a file becomes full, it is closed.

When the number of files is greater than MaxNumberOfFiles, the oldest one is deleted. At the time it also updates the StartOfData variable in the shared information buffer, used by external programs such as MQUAKE.

Time Thread

The Time Thread has two critical functions: it sets the system time once, before the DAEMON tells the driver to initiate A/D conversion, and then it keeps track of the difference between the system time and the correct time. The Time Thread uses the digital I/O portions of the driver and the A/D card to acquire sixteen bits of time information from a satellite clock receiver. Although the computer system keeps track of time to the millisecond, the system time can only be set to the nearest hundredth. When the Time Thread is started, it reads the satellite clock time to the nearest millisecond, then sets the system time. This thread then regularly reads the satellite clock time (once per minute), then adjusts the variable dTimeOffset. The other portions of the DAEMON that need to know the correct time call a special function that uses dTimeOffset to return the adjusted system time.

Read File Thread

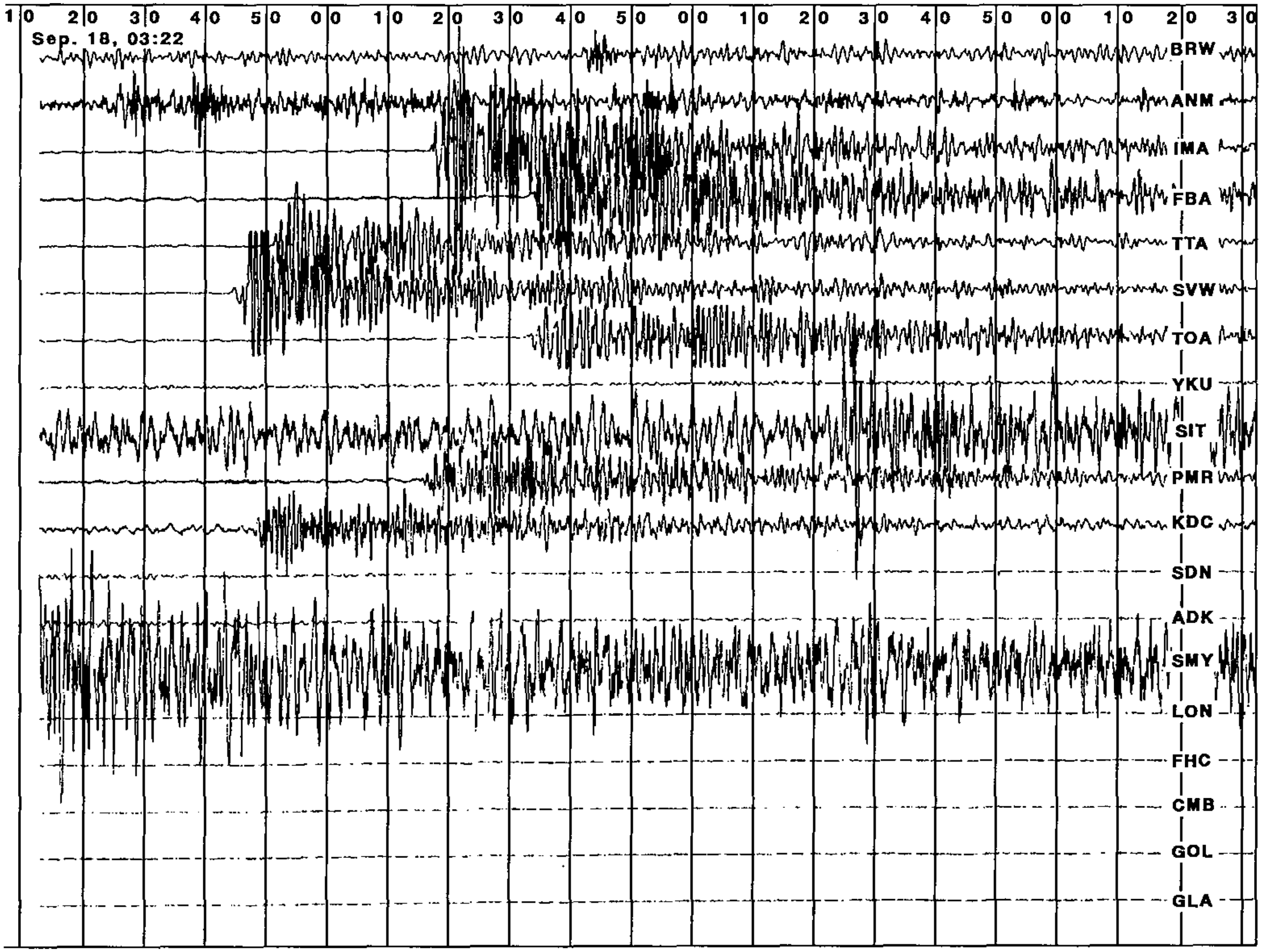
The Read File Thread is awakened by the SP Share Thread when that thread has been unable to find data requested by another program (such as MQUAKE). Since this thread has access to the data structures maintained by the Write File Thread, the thread can determine if the requested data are present in one of the seismic data files. Before actually reading any data, the thread calculates where in the appropriate file the data are (for reasons that are not exactly arbitrary, each file holds a maximum of about 1 Mbyte of data).

How the thread proceeds depends on whether the file is presently held open by the Write File Thread. Read File Thread requests the file's semaphore to lock the resource (this is also necessary since the Write File Thread may choose this occasion to delete or write to the file). If the file is already open, the thread records the position of the file pointer, otherwise the file is opened. Next, the thread moves the file pointer to the calculated position and reads either the total amount of data requested or to the end of file. Then, depending on conditions mentioned above, either the file is closed or the file pointer is returned to its old position. In the latter case, this prevents Write file thread from overwriting data. The thread then copies the data to the shared data buffer, and clears a semaphore to let SP Share Thread know that is finished.

APPENDIX B

OUTPUT: SCREEN DISPLAY OF SEISMIC SIGNATURES FROM A LOCAL EARTHQUAKE AND ITS SOLUTION WINDOW

LOCAL EARTHQUAKE SIGNATURES



LOCAL EARTHQUAKE SOLUTION WINDOW

Print pP-P Depth Comment Magnitude Display

P Times

SMY	031921.2
ADK	031932.1
SDN	032149.3
ANM	032219.3
SVW	032244.2
KDC	032247.7
TTA	032250.8
PMR	032315.2
IMA	032316.7
TOA	032331.7
FBA	032332.2
BRW	032342.2
SIT	032423.9
LON	032601.8
FHC	032620.1
CMB	032650.7
GLA	032741.2
GOL	032751.0

EARTHQUAKE SOLUTION

STA	P-TIME	RES	DIS	AZM	STA	P-TIME	RES	DIS	AZM
SMY	031921.1	-0.0	2.6	297.8	TOA	032331.7	-0.9	22.0	47.5
ADK	031932.1	1.0	3.4	82.6	FBA	032332.2	-1.2	22.1	39.7
SDN	032149.3	-0.7	13.4	65.2	BRW	032342.2	1.3	22.9	20.8
ANM	032219.3	-0.6	15.7	27.4	SIT	032423.9	0.1	27.4	59.9
SVW	032244.2	2.9	17.4	46.4	LON	032601.8	-0.5	38.7	72.6
KDC	032247.7	-2.6	18.1	58.5	FHC	032620.1	2.1	40.6	81.8
TTA	032250.8	2.1	18.0	40.7	CMB	032650.7	0.9	44.5	82.6
PMR	032315.2	-2.2	20.5	48.0	GLA	032741.2	-0.6	51.2	83.7
IMA	032316.7	0.2	20.4	33.8	GOL	032751.0	-1.3	52.6	71.0

H(Z) = 03:18:40 Sep. 18, 1990
H(ADT) = 07:18:40 pm Sep. 17, 1990
LOC = 230 miles W of Adak, Ak.
LOC = 100 miles SE of Buldir Is., Ak.

LOC = 51.6N 177.9E

DEP = 38
AVG. RES. = 1.2
AZIMUTH = 83

Solve

Search For Bad P

Use All Picks

Initial Location
- (•) AK or West Coast
- ( ) Elsewhere
- ( ) Enter Below: [EX. 45N 123W]

Depth Control
- (•) Fix at 40 km
- ( ) Float
- ( ) Fix Depth Below: [EX. 230]

Station Usage
- (•) Don't Use
- ( ) Delete Stations
- ( ) Add Station: [EX. PMR 012345.6]

Real Time Magnitudes

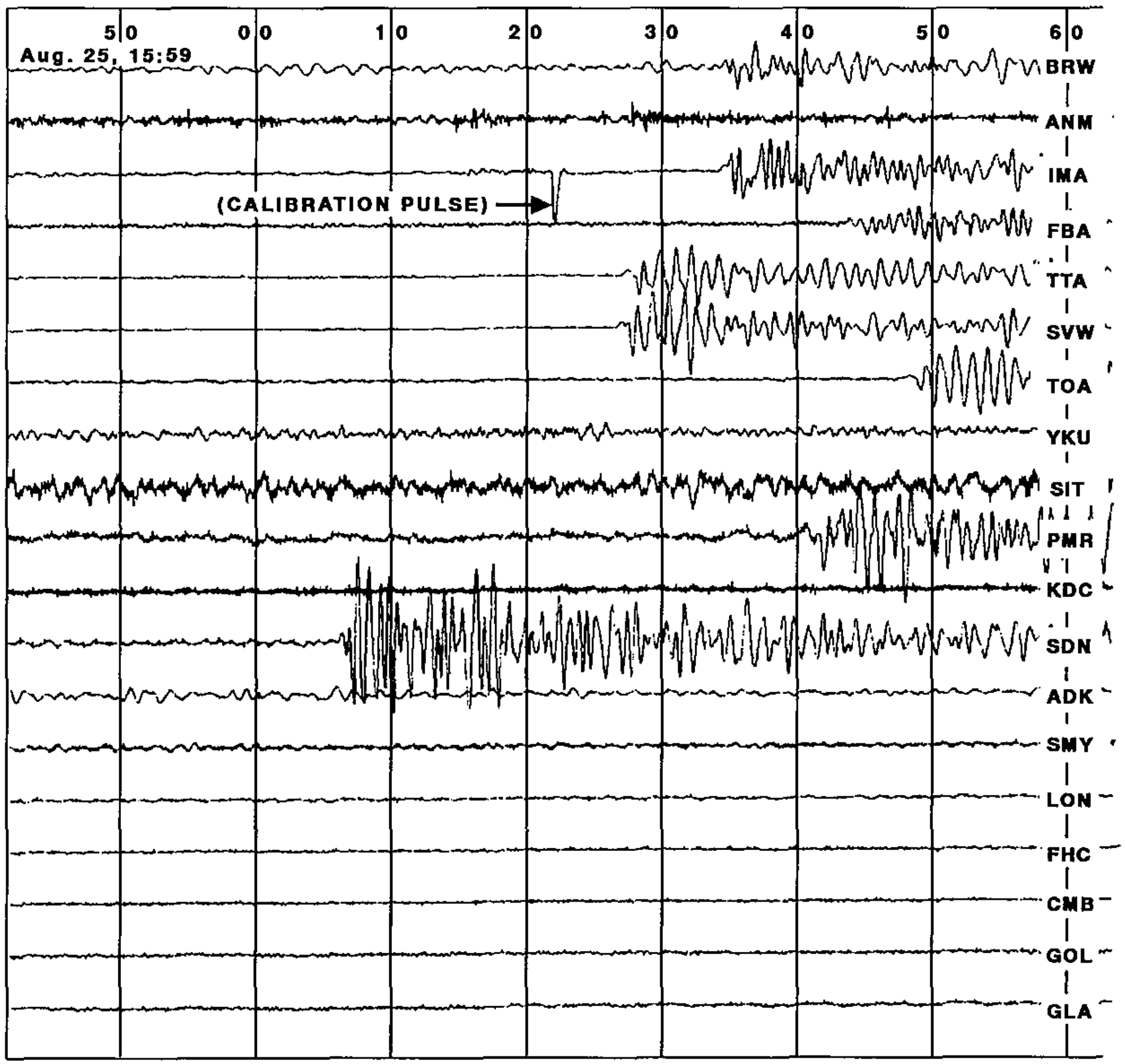
STA	PER	AMP @ 10K	GAIN	DIST	MI	mb	MB	MS
SMY	1.3	277.1	147.8	2.6	5.5			
ADK	1.1	612.2	0.3	3.4	6.0			
KDC	0.7	3.7	816.9	18.1		5.1		
TTA	1.4	13.1	1806.8	18.0		5.9		
PMR	1.0	5.3	958.2	20.5		5.5		
IMA	0.9	5.7	3245.3	20.4		5.5		
TOA	1.1	19.4	1852.9	22.0		6.2		
FBA	1.1	6.3	1754.9	22.1		5.8		

MI= 5.8: 2 Station Average mb= 5.7: 6 Station Average

APPENDIX C

OUTPUT: SCREEN DISPLAY OF SEISMIC SIGNATURES FROM A TELESEISMIC EARTHQUAKE AND ITS SOLUTION WINDOW

TELESEISMIC EARTHQUAKE SIGNATURES



TELESEISMIC EARTHQUAKE SOLUTION WINDOW

Print pP-P Depth Comment Magnitude Display

P Times

SDN	160005.5
SVW	160026.0
TTA	160026.6
IMA	160033.3
BRW	160034.1
PMR	160041.1
FBA	160044.3
TOA	160048.1
SIT	160115.5

EARTHQUAKE SOLUTION

STA	P-TIME	RES	DIS	AZM	STA	P-TIME	RES	DIS	AZM
SDN	160005.5	-0.2	79.0	33.7	PMR	160041.1	0.0	85.9	28.5
SVW	160026.0	0.6	82.7	28.5	FBA	160044.3	-1.0	86.7	25.2
TTA	160026.6	0.3	82.9	26.7	TOA	160048.1	-0.2	87.3	28.0
IMA	160033.3	-0.8	84.4	23.8	SIT	160115.5	0.5	93.1	32.8

H(Z) = 15:48:04 Aug. 25, 1990
H(ADT) = 07:48:04 am Aug. 25, 1990
LOC = General Area of Molucca Sea

LOC = 1.4N 127.0E

DEP = 40
AVG. RES. = 0.5
AZIMUTH = 344

Solve

Search For Bad P

Use All Picks

Initial Location
- (●) AK or West Coast
- ( ) Elsewhere
- ( ) Enter Below: [EX. 45N 123W]

Depth Control
- (●) Fix at 40 km
- ( ) Float
- ( ) Fix Depth Below: [EX. 230]

Station Usage
- (●) Don't Use
- ( ) Delete Stations
- ( ) Add Station: [EX. PMR 012345.6]

Real Time Magnitudes

STA	PER	AMP @ 10K	GAIN	DIST	MI	mb	MB	MS
SDN	0.8	24.6	423.5	79.0		6.7		
SVW	0.7	6.4	1139.3	82.7		6.1		
TTA	1.0	2.8	1662.9	82.9		5.8		
IMA	0.6	1.1	3118.8	84.4		5.4		
PMR	0.8	1.1	947.0	85.9		5.6		
FBA	0.8	0.3	1685.9	86.7		5.1		
TOA	1.0	4.2	1737.8	87.3		6.3		

mb= 6.7: 7 Station Average

NWS AR-11 Winter 73-74 Weather and Carbon Monoxide Air Pollution in Fairbanks, Alaska. John R. Zimmerman and Kenneth W. MacKenzie Jr. February, 1975.

NWS AR-12 Marine Forecast Verification Program for Western and Northern Alaska April 1974 - March 1975. Benjamin C. Hablutzel. April, 1975.

NWS AR-13 Thunderstorm Climatology of Alaska. Gary K. Grice and Albert L. Comiskey. February, 1976.

NWS AR-14 The wintertime Arctic Front and its Effects on Fairbanks. Ronald A. Willis and Gary K. Grice. April, 1976.

NWS AR-15 A Preliminary Analysis of Precipitation in the Chena Basin, Alaska. Henry S. Santeford. October, 1976.

NWS AR-16 Forecasting Avalanches in Juneau, Alaska. Richard Hutcheon and Leif Lie. January, 1977.

NWS AR-17 Modifying Numerical Guidance in Alaska. Theodore F. Fathauer. July, 1977.

NWS AR-18 Sea Ice Conditions in Cook Inlet, Alaska During the 1973-74 Winter. Ruben Schulz. September, 1977.

NWS AR-19 Sea Ice Conditions in Cook Inlet, Alaska During the 1974-75 Winter. Ruben Schulz. December, 1977.

NWS AR-20 Sea Ice Conditions in Cook Inlet, Alaska During the 1975-76 Winter. Ruben Schulz. January, 1978.

NWS AR-21 Forecasting Fire Occurence Using 500 MB Map Correlation. David M. Henry. January, 1978.

NWS AR-22 Monthly Temperature and Precipitation Outlooks for Anchorage, Alaska Using Data from Past Months and Sunspot Numbers. David M. Henry. April, 1978.

NWS AR-23 A Forecast Procedure for Coastal Floods in Alaska. Theodore F. Fathauer. June, 1978.

NWS AR-24 LFM-II Precipitation Guidance in Alaska. Jeffrey P. Walker and Laurence G. Lee. August, 1979.

NWS AR-25 Snowfall at Anchorage, Alaska Associated with Cold Advection. Donald L. Finch and Jeffrey P. Walker. October, 1979.

NWS AR-26 Ice Edge Probabilities for the Eastern Bering Sea. Bruce D. Webster. October, 1979.