AD A105205

DYNEX CONSULTING CO.
LYNN HAVEN, FLORIDA

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

F-200.1473 *DD Form 1473:* Report Documentation Page

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER FR-XX-2	2. GOVT ACCESSION NO. AD-A105 205	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Software and support development for an environmental data buoy system for predicting surf zone characteristics		5. TYPE OF REPORT & PERIOD COVERED Final(5/79-2/81)
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) V. W. Pidgeon and N. A. Pidgeon		8. CONTRACT OR GRANT NUMBER(s) N00014-79-C-0497
9. PERFORMING ORGANIZATION NAME AND ADDRESS Dynex Consulting Company 1001 Kimberly Lane Lynn Haven, Fla 32444		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, Virginia 22217		12. REPORT DATE 2/28/81
		13. NUMBER OF PAGES 126
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) DCASR, Birmingham Birmingham, Alabama 35205		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Unlimited	DISTRIBUTION STATEMENT A Approved for public release; Distribution Unlimited	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Environmental Intelligence (U) Coastal Operations (U) Near-shore Processes (U) UDT/SEAL Reconnaissance (U)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A data-buoy system has been designed, fabricated, tested, and evaluated for measuring offshore wave and currents for a prediction model(s)which predict breaker statistics and longshore currents. Accuracies with errors less than 10% have been achieved with this system. Over 20 software programs have been developed for buoy application. (UNCLASSIFIED).		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DEPARTMENT OF DEFENSE FORMS

F-200.1473 *DD Form 1473:* Report Documentation Page

INSTRUCTIONS FOR PREPARATION OF REPORT DOCUMENTATION PAGE

RESPONSIBILITY. The controlling DoD office will be responsible for completion of the Report Documentation Page, DD Form 1473, in all technical reports prepared by or for DoD organizations

CLASSIFICATION Since this Report Documentation Page, DD Form 1473 is used in preparing announcements, bibliographies and data banks, it should be unclassified if possible If a classification is required, identify the classified items on the page by the appropriate symbol

COMPLETION GUIDE

General Make Blocks 1, 4, 5, 6, 7, 11, 13, 15, and 16 agree with the corresponding information on the report cover. Leave Blocks 2 and 3 blank

Block 1. Report Number Enter the unique alphanumeric report number shown on the cover

Block 2 Government Accession No Leave Blank This space is for use by the Defense Documentation Center

Block 3 Recipient's Catalog Number Leave blank This space is for the use of the report recipient to assist in future retrieval of the document

Block 4. Title and Subtitle Enter the title in all capital letters exactly as it appears on the publication Titles should be unclassified whenever possible Write out the English equivalent for Greek letters and mathematical symbols in the title (see "*Abstracting Scientific and Technical Reports of Defense-sponsored RDT E*," AD-667 000) If the report has a subtitle, this subtitle should follow the main title, be separated by a comma or semicolon if appropriate, and be initially capitalized If a publication has a title in a foreign language, translate the title into English and follow the English translation with the title in the original language Make every effort to simplify the title before publication

Block 5. Type of Report and Period Covered Indicate here whether report is interim, final, etc , and, if applicable, inclusive dates of period covered, such as the life of a contract covered in a final contractor report

Block 6 Performing Organization Report Number Only numbers other than the official report number shown in Block 1, such as series numbers for in-house reports or a contractor/grantee number assigned by him, will be placed in this space If no such numbers are used, leave this space blank

Block 7. Author(s). Include corresponding information from the report cover Give the name(s) of the author(s) in conventional order (*for example, John R Doe or, if author prefers J Robert Doe*) In addition, list the affiliation of an author if it differs from that of the performing organization

Block 8 Contract or Grant Number(s) For a contractor or grantee report, enter the complete contract or grant number(s) under which the work reported was accomplished Leave blank in in-house reports

Block 9 Performing Organization Name and Address For in-house reports enter the name and address, including office symbol, of the performing activity For contractor or grantee reports enter the name and address of the contractor or grantee who prepared the report and identify the appropriate corporate division, school laboratory, etc , of the author List city, state, and ZIP Code

Block 10. Program Element, Project, Task Area, and Work Unit Numbers Enter here the number code from the applicable Department of Defense form, such as the DD Form 1498 "Research and Technology Work Unit Summary" or the DD Form 1634 "Research and Development Planning Summary," which identifies the program element, project, task area, and work unit or equivalent under which the work was authorized

Block 11 Controlling Office Name and Address Enter the full, official name and address including office symbol, of the controlling office (*Equates to funding sponsoring agency For definition see DoD Directive 5200 20 "Distribution Statements on Technical Documents"*)

Block 12 Report Date Enter here the day, month and year or month and year as shown on the cover

Block 13 Number of Pages Enter the total number of pages

Block 14. Monitoring Agency Name and Address (*if different from Controlling Office*) For use when the controlling or funding office does not directly administer a project contract, or grant but delegates the administrative responsibility to another organization

Blocks 15 & 15a Security Classification of the Report: Declassification/Downgrading Schedule of the Report Enter in 15 the highest classification of the report If appropriate, enter in 15a the declassification/downgrading schedule of the report, using the abbreviations for declassification/downgrading schedules listed in paragraph 4-207 of DoD 5200 1-R

Block 16. Distribution Statement of the Report Insert here the applicable distribution statement of the report from DoD Directive 5200 20, "Distribution Statements on Technical Documents "

Block 17. Distribution Statement (*of the abstract entered in Block 20 if different from the distribution statement of the report*) Insert here the applicable distribution statement of the abstract from DoD Directive 5200 20, "Distribution Statements on Technical Documents "

Block 18 Supplementary Notes Enter information not included elsewhere but useful, such as Prepared in cooperation with Translation of (or by) Presented at conference of To be published in

Block 19 Key Words Select terms or short phrases that identify the principal subjects covered in the report, and are sufficiently specific and precise to be used as index entries for cataloging, conforming to standard terminology The DoD "Thesaurus of Engineering and Scientific Terms" (*TEST*) AD-672 000, can be helpful

Block 20 Abstract The abstract should be a brief (*not to exceed 200 words*) factual summary of the most significant information contained in the report If possible, the abstract of a classified report should be unclassified and the abstract to an unclassified report should consist of publicly- releasable information If the report contains a significant bibliography or literature survey, mention it here For information on preparing abstracts see "Abstracting Scientific and Technical Reports of Defense-Sponsored RDT&E," AD-667 000

[The next page is F234.8a]

DYNEX CONSULTING COMPANY

1001 Kimberly Lane
LYNN HAVEN, FLORIDA 32444
904-265-5739

(6) SOFTWARE AND SUPPORT DEVELOPMENT
FOR AN ENVIRONMENTAL DATA BUOY SYSTEM
FOR PREDICTING SURF-ZONE CHARACTERISTICS
~~Final Report~~

(9) Final rept.,

(15) N00014-79-C-0497

(14) Report Number FR-XX-2

(11) 28 February ~~28,~~ 1981

(12) 132

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

(10) V. W./Pidgeon
N. A./Pidgeon

DYNEX CONSULTING COMPANY
1001 Kimberly Lane
LYNN HAVEN, FLORIDA 32444
904-265-5739

ACKNOWLEDGEMENTS

Although Dynex Consulting Company played an important role in the development of the data buoy system to date, this effort was done jointly by three organizations: the Naval Coastal Systems Center, Panama City, Florida; Louisiana State University, and Dynex Consulting Company. The efforts of NCSC and LSU personnel are acknowledged. Specifically the authors wish to acknowledge gratefully the following persons:

W. W. Howard, NCSC Task Leader, who provided the over-all program direction and his expertise both as an environmentalist and as a scientific diver, providing many valuable inputs into the program;

R. A. Brown, formerly NCSC Electronics Technician, who did almost all of the fabrication of the buoy system, who put up with all of the seemingly endless testing and checking-out of the system that the author suggested, for his excellent part in the buoy deployment phases, and for his assistance in the fabrication, testing, and installation of the field data collection system;

J. N. Suhayda, Louisiana State University, for providing the two very good prediction models given in this report and for his time spent with the authors regarding these models and the over-all buoy system;

In addition, the authors wish to acknowledge the efforts by personnel at and for use of the Field Research Facility at Duck, North Carolina, part of the U. S. Army Corps of Engineers Coastal Engineering Research Center. The efforts of William E. Grogg, Electronics Technician, are specifically gratefully acknowledged for his valuable help in setting up, testing, and checking out the field data collection system at that facility and for his assistance in making the data recordings.

Funding for this study was provided by NCSC and the Office of Naval Research under ONR contract number N00014-79-C-0497, and the authors gratefully acknowledge this support. Finally grateful acknowledgement is extended to the UDT/SEAL Hydrographic Reconnaissance Support Program of the Naval Explosive Ordnance Disposal Facility, Indian Head, Maryland, for the over-all funding support of this Data Buoy System program.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF PROGRAMS

Program Listings (Cont)

SUMMARY

An environmental data buoy system is being developed, tested, and evaluated for measuring offshore wave and current information which is to be used to predict the characteristics of the breaker zone. This buoy system, which is bottom-mounted outside of the near-shore surf zone, measures wave height information, tide, and two cartesian components of current. This data is being used as input information into the precition models that predict individual breaker direction, and longshore current information. The breaking wave height is calculated for each wave during a 10 to 30 minute measurement period from which breaker statistics are derived. The maximum longshore current velocity is calculated from the average breaker height and average breaker direction. The prediction models have been tested against actual breaker conditions measured in the surf zone. A method has been developed in this study and was used for making the measurements of the breaker heights.

The buoy system contains a pressure-transducer wave and tide gauge, a two-axis electromagnetic current meter, and several other sensors not used in the surf-zone prediction. The data are digitized and sent in a serial-digital format compatible with most computers with a serial interface (RS-232). On shore, data are input into a 103-type telephone modem, which can be called up from anywhere in the telephone switching network. In this study, the data are processed off-line by a modest processor, but programs have been developed, written, and tested with actual data for a microcomputer system, a similar version of which could be ultimately implemented in the buoy for complete on-board data processing.

Extensive software development was done in this twenty-two month study period. Over twenty software programs were written, tested, and are operational for use either in off-line processing at a remote or processing location or for adaption to on-board buoy processing. These programs cover four basic program types and four programming languages: Assembly Language, BASIC, HPL, and FORTRAN. Each program type is discussed in detail and is flow-charted, and program listings are given for each program type and the appropriate programming language used (most programs were written in more than one

language). In many instances, applications for other Navy Fleet operations are suggested for these program utilizations.

Two surf-zone prediction models were used to describe the nearshore waves and the longshore current velocity in this study. These models are based upon linear theory, which allows shallow and intermediate waves to be treated simply and consistently. In each model, monochromatic waves are assumed and wave processes of refraction, frictional attenuation, shoaling, and breaking arc considered. The initial model is restricted to shallow-water waves which approach the shore nearly perpendicularly. The more-detailed model allows the waves to be measured at greater water depths and at varying approach angles. Both of the models used in this study were developed by personnel at Louisiana State University.

Several series of tests have been made in the Gulf of Mexico near Panama City, Florida, in a six-meter water depth, 300 meters offshore. Data colected from these tests were used for the evaluation of the buoy operation and for testing the prediction models. During some of these tests, measurements of actual breaker conditions were made and have been used for comparing the prediction models with actual measured heights. In addition to the tests at Panama City, tests have been made at the Coastal Engineering Research Center Field Research Facility at Duck, North Carolina. At this location, there are seven surface wave gauges extending from almost 500 meters offshore in eight meters of water to about 70 meters offshore in one meter of water depth. The gauges can measure the wave modification accurately and in detail as the wave progresses shoreward. A field data collection system was developed for this study to collect and record the data digitally in serial format (RS-232) from these gauges and other instruments at this location.

The test results indicate that for the longer-period waves, both models very accurately predict the breaker heights within an accuracy often less than ten percent error when comparing average breaker height, average of the highest one-third breaker height, and average of the highest one-tenth breaker height. These results are for breaker heights up to two meters for the test results using the Panama City data. Further results for the Duck data for the detailed prediction model indicate for breaker heights up to about 2.5 meters, the average prediction error is almost zero, giving extremely accurate results. However, it is cautioned that almost all of the time breaking did not occur at one of the wave gauges; therefore, the actual measurements of the breaker height was not the height of the breaker at the time of breaking. Since these gauges were rather closely spaced in the breaker zone, this difference in measurement is expected to be within the accuracies given above for the Panama City data.

For many applications, the simplified model probably will provide adequate answers, and the additional mathematical complexity would not be justified. This additional complexity not only would require a larger processing system, but requires more time for processing the data. The processing time on a modest processor is about an order of magnitude faster for the simplified version than for the more-detailed version. However, even for the more-detailed processoring, the processing times are short compared with data collection times. With the use of high-speed micro-computer systems and efficient programs, such as those compiled from a FORTRAN source as given in this study, processing times of the order of one minute are reasonable for thirty minutes of wave data.

Complete details of this study are given in the six sections of this report.

INTRODUCTION

The coastal environment plays a key role in many of the Navy's vital operations. Of specific interest to the Navy are such paramaters as near-shore waves, tides, currents, and surf. Other parameters include water temperature and visibility. Normally, for many areas of interest, little or no details exist regarding either generalized or specific information for these and related parameters. For many Navy operations, these parameters are required inputs either directly or indirectly to obtain tactical information leading to decisions regarding those operations. Quite often the effectiveness of an operation may depend upon the reliability of the environmental information associated with the operation that is obtained either from direct measurements or from information in data banks or from numerical models. Such existing data or the ability to obtain for areas of interest to Navy operations the desired data is limited at present times.

Not only is the data limited, but the means and/or instrumentation for obtaining the necessary information are also limited. The tools that underwater teams currently use are the same ones being used for some time: typically unsophisticated equipment and methods such as the measure line and slate board with visual observation of wave height, direction, period, and currents. For some operations, this type of information may be adequate. However, for most operations, it is felt that such results would not provide the basic inputs for strategy and tactics for a successful operation. These, also, would not provide the needed inputs for the development of a point-target intelligence-data base.

For an amphibious assault operation, where the coastal zone must be transversed, the success of the operation might well depend upon knowing when and where the crossing should take place. For such operations, previous information regarding these areas would be significantly helpful in reaching proper decisions. If, however, real-time or near real-time information were available, probabilities of success would be considerably increased. This is particularly the situation for most coastal areas of the world where there can be a large variation in the given environmental parameters.

In addition to being able to obtain the required information, it is necessary to have this information in a form or format that will be readily useful to those needing to make the decisions involved in the operation. In general, this means having an analytical model into which the input environmental parameters are applied, giving prediction results in a simplified format. In the process of reducing down large amounts of data to a concise output, care must be exercised not to lose or average out the important details of the coastal process. What is needed to accomplish these goals is an instrumentation system capable of providing the inputs with the necessary accuracy and reliability for a proven model, that has been tested and evaluated against known ground-truth standards.

The development of such a measurement system with the associated analytical models is the objective of this several-year study. The first phase involves the development of a buoy system with environmental instrumentation for measuring, at an off-shore location outside of the breaker zone, wave and current parameters to be used as model inputs for the prediction of wave-breaker statistics and longshore-current properties. At the same time, models for predicting these effects are being developed. In order to evaluate the effectiveness of the model, it is necessary to also measure the properties being predicted. This is being done during this first phase of the tests. However, the accuracy to which the ground-truth standards can be measured in these first tests may not be much more accurate than much of the differences of the results that the different models predict. For this reason, further evaluation of these models must be done using more accurate measurement techniques and instrumentation. This is to be done in the second phase of this study, using instrumentation dispersed throughout the breaker region.

The end product of this several-year study is expected to be a deployable buoy system that can be readily implanted outside of the breaker zone in an operational area. This buoy system may take on any of several versions which might include either detailed on-board processing, limited on-board processing, or telemetry of raw data to an out-board processor. Incorporated into the processing system would be the appropriate model for the prediction of the breaker and current statistics. This total information from the buoy system would give an operational

Commander not only an over-view of the environmental conditions of the area, but also, hopefully, the necessary information for reaching strategic and/or tactical decisions regarding the operation.

This final report documents the results of a twenty-two month study. Although a detailed interim report (FR-XX-1)* was issued one year ago, detailing the results to date, much of that report has been incorporated into this final report for completeness. However, parts are not included here but are referenced at the appropriate portion of the report. The extensive test data processing results of Appendices A and B of that report are not included in this report, but are referenced in the test results section. Although some of the program development discussions of the interim report are referenced and not detailed in this final report, because software development was a major portion of this study, extensive discussions are included in this report. In this latter respect, this final report is self-contained.

The stated objectives, given in the proposals for this study, are:

1. To develop software for a computer system to process offshore buoy data to obtain wave height, period, and direction, tide, and current speed and direction statistics at the buoy location and to develop for the same computer software for a surf zone model that uses these above processed statistics as inputs for processing surf heights, periods, shore currents and other results.

2. To develop software and generalized overall system design parameters that would lead to the final design and fabrication of an internal processing system integrated within the buoy that would provide the same or similar end products as those given in Objective 1.

3. To design a digital interface system for obtaining serial data in a standard format from wave-gauge and current-meter sensors and to provide technical direction in the check-out and installation of the system at the CERC Field Research Facility, at Duck, N. C.

4. To develop computer software programs for additional models for surf-zone prediction of wave and current parameters and to evaluate these and previous models using programs developed and data collected for a variety of wave and environmental conditions.

5. To develop buoy design specifications for an operational buoy system for Navy Fleet applications incorporating results of this and related previous studies.

---

*PIDGEON, V. W. and PIDGEON, N. A., "An Environmental Data Buoy System for Predicting Surf-Zone Characteristics," Dynex Consulting Company report number FR-XX-1, dated February 29, 1980.

As the reader can determine, after reviewing the extensive software development, the buoy development, testing, and evaluation, and the data-processing and model-evaluation results given in this final report and in the interim report, that all objectives given above have been successfully met by this study.

This report is divided into six sections. In the first section, Data Buoy System and Specifications, is given a detailed description of the buoy system and its instrumentation for making the environmental measurements. Complete details of the interfacing system are also given. This interfacing system, which is the controlling portion of the operational part of the buoy, converts all measured signals to a serial-digital format which is compatible with most computer systems. It also allows for a wide variety of data transmission methods and links, making this buoy system extremely flexible as far as data links are concerned. Also included in this section is a discussion of the accuracies associated with the system, with details of some of the tests given, demonstrating reliability and performance. The second part of this section gives the buoy specification details for making the prototype buoy more adaptable to Navy Fleet operations. Specifically, recommendations and specifications are given for significantly reducing the power consumption of the system as well as a telemetry link, both of which would allow the system far more flexibility and operational time on station without servicing.

Section II, Software Development, gives the very extensive software programs and background information developed in this study. Over thirty software programs were written for various application in this study. These are grouped into four main program types given in this section. Also given in this section are the objectives for the software development in this study. Discussions of each type of program developed are given along with complete functional flow diagrams and program listings for the various processors the software was developed for. This section gives the program development in Assembly Language, HPL, BASIC, and FORTRAN for the appropriate programs.

In the next two sections, Data Processing Using the Simplified Model and Data Processing Using the Detailed Model, the processing techniques for these two models are developed. Although specific details of the model development are not given (refer to LSU reports to be published for this), the basic mathematical algorithms are given which are programmed in detail for the data processing of the wave and current data from the buoy system. Included in these two sections are brief over-views of the models, a detailed description of the programming processes, and program flow diagrams and listings. This is done for two types of processors: a laboratory type and a field type. The latter type of processor has the capabilities of being simplified from a general purpose type of processor to a special purpose processor, which could ultimately be repackaged for the specific application of on-board processing in the buoy package itself.

Section V, A Field Data Collection System, contains the details of the system developed for data collecting and recording at the U. S. Army Coastal Engineering Research Center Field Research Facility at Duck, N. C. This system allows the simplified task of recording data from the seven wave gauges, two components of current, a pressure-transducer wave gauge, and a calibration signal. Data collected by this system was used for continued model testing and evaluation. Complete details are given in this section including the overall system description and operation and functional and block diagrams.

The last section, Field Tests and Processing Results, gives the details of the field operations using the buoy system and the data collected at the CERC facility. Also, during some of the tests, ground truth measurements were made. The methods used are described whereby measurements of the breaker statistics were made at the same time that data were being recorded for the off-line processing. In this section are presented summary results of the data processing from the field tests, with comparisons of actual wave and breaker measurements to the predicted conditions using the models given.

Following Section VI are given the Conclusions and Recommendation of this report in which the major conclusions of the previous sections are summarized and the recommendation for follow-on efforts are given.

SECTION I

DATA BUOY SYSTEM AND SPECIFICATIONS

Part I. Data Buoy System

The design of the data buoy system involved consideration of many factors. One of the major objectives of the design was to develop a measurement system that would operate in shallow water outside of the surf region and would be able to measure the hydrodynamic parameters needed as off-shore inputs into a prediction model for determining the statistics of wave breakers and the longshore currents induced. A secondary consideration was the development of a system that might either lead to or provide significant information for the design of an operational buoy system that could be used in the fleet operations and would provide outputs detailing the near-surf and surf regions of an area under consideration. Although it was realized that this would be a multiple phase program, ultimate end products were considered along the various development stages of the present system. A specific example of this is the work that was done in the processing area. Although considerable program development was done using the NCSC outboard processors (specifically the Hewlett Packard HP-9825 system), much work was also done with processing by microcomputers. The objective of this latter effort is for the proposed capability of the buoy system to perform on-board processing, which would undoubtedly use state-of-the-art microprocessing components.

Much of the development effort for this system was done by personnel at the Naval Coastal Systems Center. Dynex personnel provided significant inputs into the over-all system design, particularly in the area of interfacing the measuring instrumentation to the system output instrumentation. The over-all testing and evaulating of the system was done with technical assistance and direction by Dynex personnel. NCSC personnel did the complete fabrication of the buoy system as well as all deployment and retrieval operations. Data collection and processing was done by Dynex personnel on both the NCSC HP-9825 system and the Dynex COMPAL-80 computer. Actual surf breaker, current, and wave observations were made both by NCSC and Dynex personnel during periods of data collection from the buoy system. The prediction model information was provided by personnel from Louisiana State University.

The buoy system measures waves, tide, two orthogonal components (horizontal) of current, water temperature, internal temperature, and system calibration. The waves and tide are measured by a pressure transducer wave and tide gauge developed by NCSC. A two-axis electro-magnetic induction current meter measures the water motions in the horizontal plane. The temperatures are measured by systems also developed at NCSC. A calibration signal is added to one of the data output channels to monitor system performance. In addition, a leak detector, also developed by NCSC, was added to the system that would detect the presence of water in the bottom of the buoy system. All of the signal outputs from the above sensors are analog voltages. These outputs are fed into a multiplexing, digitizing system for data transmission or hard-wiring to the shore. More details of this system are given in a later portion of this section.

The sensors and all associated electronics are housed in a cylindrical container, sealed to withstand at least 30 meters of water depth. The housing is approximately 60 centimeters long by 20 centimeters in diameter. In water, the system has positive buoyancy and must be ballasted by approximately 15 kilograms of weight, which is attached directly to the bottom of the housing at the time of deployment. The current meter probe is mounted with a bulk-head seal through the top of the housing and protrudes almost 30 centimeters above the housing. The current sensors are located near the top end of the probe in water relatively undisturbed by the housing. The pressure diaphram and the water temperature sensors are also located on the top of the housing.

The buoy system is implanted in a vertical position mounted between two pipes that are jetted into the bottom for support. It can be readily deployed by divers without special equipment except for water jets for the support anchors. If the bottom allows it, screw anchors can be used instead of the jetted pipes. The buoy system, including ballast, weighs less than 25 kilograms in air; therefore, it is not difficult to handle during deployment. In each deployment and retrieval operations were done from a small outboard boat 6 meters in length. It is not expected that the size or configuration of the buoy system for future models will change very much from the present configuration, even with the addition of on-board processing. Therefore, deployment of future versions is expected not to be a difficult task. If necessary to accomodate additional electronics internally, the cylinder can be lengthened by using a longer housing than the present model. It is expected that these future versions still would not be too difficult for one person to handle.

A major consideration in the design of the buoy system was the method of getting the analog signals from the sensors to the processing equipment. Desirably, the data would be sent in digital form in order to obtain the most immunity from noise and other degrading factors. Also, in order to avoid having many signal lines between the buoy and the receiving station, a serial not a parallel system was needed. A very important consideration in the design of this interfacing system was the formatting and makeup of the data. With future on-board processing in mind, a format that would be directly compatible with microcomputers and other similar processors is what is needed. Each of these considerations fit very well into an interfacing system that had been developed by Dynex for another NCSC task. This system, called Information Transmssion System (ITS), takes multiple analog data channels (up to 128), multiplexes them into a 12 bit digitizer (giving one millivolt resolution out of four volts), converts the parallel output data into two eight-bit bytes, which are sequentially applied to a standard UART (Universal Asynchronous Receiver Transmitter), from which the data are sent in a bit-serial, digital format. This serial format is completely compatible with computer standards for serial data transfer, i. e., meets RS-232C standards. Therefore any computer processor with a standard RS-232 serial interface can accept the data from this buoy system without any further modification.

The ITS system was interfaced into the buoy system essentially without modification from the latter design version. The incorporation of this system into the data buoy had several advantages in terms of data recording, transmission, and computer inputting. Two previous reports[1] give specific

1. Pidgeon, V. W. "A Development Study for a Field and Laboratory Data System" Dynex Consulting Company Report, FR-XVI-1, dated May 30, 1979

Pidgeon, V. W. "A Development Study for a Field and Laboratory Data System, Phase II" Dynex Consulting Company Report, FR-XVI-2, dated November 1, 1979 details of the system, operation, and application, including details of the programming with flow diagrams and circuit drawings. However, a brief over-all discussion is included in this report for completeness. The interested reader is referred to the above referenced reports for more details.

An important consideration in the transfer of data from the measuring source to the processing system is the preservation of the data as nearly as measured as practical. Secondly, the ability to input directly into the processor with minimal reformatting means more efficiency, fewer errors, and less processing costs. For on-board processing, both of these considerations are almost mandatory. These factors are the prime ones considered in the development of the ITS system, and they also made it a prime candidate for the data buoy system. The addition of ITS to the buoy system allowed considerable flexibility in how the data could be handled, once it was obtained in the buoy. The data could be transmitted via standard telemetry links with relatively low bandwidths. Or the data could be directly connected with the beach site via cable. This latter version was selected for these preliminary tests since both data signals and buoy power could be cabled between the beach site and the buoy with a modest cable having as few as four conductors -- two for power and two for signal.

After the data were received at the beach site (NCSC Beach Tower #1), there were two primary choices of what to do with the data at that point. The data could be recorded directly on a digital recorder and later taken to the processing site. The method, however, would require either a person to be on the site to do the recording or for the recording to take place at designated intervals under control of a timer. Since it was very desirable to take selected data and not data that would be recorded at set intervals, and since it would be costly to always send a person to the site to do the recording, this method was not used except when personnel were on site for other reasons. The second method, which was selected for most of the data recording, was to input the data signals directly from the buoy cable into a standard Bell 103-type telephone modem that has auto-answer capabilities. Since the data from the buoy is standard RS-232 format and is transmitted at the standard baud rate of 300 baud, the buoy data was completely compatible with this type modem. Thus, when it was desired to take data or to check the buoy operation, all that one had to do was to call up the buoy from NCSC or the Dynex office with either an acoustic coupler (which was used by NCSC) or a modem (which was used by Dynex). When the modem at the beach tower auto-answered, the handshake was completed at the originating end. At this point, data began to stream in and continued until the originating source turned off its lock-up tone or the calling phone was hung-up. The data that was received could then be used in several ways. It could be checked to ascertain proper buoy operation, it could be used to determine the real-time conditions in the Gulf of Mexico at that time, or it could be directly recorded or input into the processor for data processing.

This method of interfacing the data and applying it to a modem allows the data to be called up from anywhere in the world that a telephone can call the number of the modem at the beach tower. The calling person would then be able to receive in real time the data from the Gulf of Mexico. This application added considerable flexibility and power to the data buoy system.

In order to be able to sample the wave data rapidly enough to detail with accuracy the wave profile, the wave data were sampled most often of the data. This was done by putting the wave data on every other channel. Interdispersed between the wave data were the other signals. The specific channel assignment of the data is given in Table I-1. With the ITS system running at 300 buad, the wave data were sampled almost five times per second. This was sufficiently fast enough to well profile the waves. (It is necessary in the data processing to determine the peak and trough of each individual wave as well as its period. Therefore, an accurate reproduction of the wave profile was necessary. More details of the data processing are given in those sections.)

The buoy system was thoroughly checked out and tested prior to each deployment. An extensive testing was done on the ITS system to determine the stability and reliability of the system. This testing was done both in this study and the other related studies under which development of the system was done. A three-month testing period under a large temperature variation showed that this system was stable to one millivolt or less with the system error never exceeding this amount. Also during this time period there were no failures or malfunctions of this system.

Several other tests were performed to determine the reliability of the data received over the transmission lines and for an over-all checking of the system. Known data such as the calibration signal was checked repeatedly and was found to be within one bit during all testing. Extensive testing of transmitted data over the phone lines was done to validate that link and the ITS system. Several hundred thousand readings have been taken, and the address of each reading checked. The results of these tests have indicated the system to be almost entirely error free. All tests to date indicate that this buoy system is a reliable one and an accurate one for making the type of measurements for which it was designed.

TABLE I-1. DATA BUOY CHANNEL ASSIGNMENTS

CHANNEL NO.	SENSOR/INSTRUMENT
1	Wave Gauge
2	Long-shore current
3	Wave Gauge
4	On-shore current
5	Wave Gauge
6	Tide Gauge
7	Wave Gauge
8	Water Temperature
9	Wave Gauge
10	Long-shore current
11	Wave Gauge
12	On-shore current
13	Wave Gauge
14	Voltage calibration/leak detector
15	Wave Gauge
16	Internal buoy temperature

Part II. Data Buoy Specifications

In the first part of this section are given discussions on the buoy system and its instrumentation, test and evaluation of the system, and its application in the NCSC field tests. The results of the first study given by FR-XX-1 demonstrated feasibility of this prototype data buoy system. Now that this study is complete, specifications can be given, based upon the previous usage results, for the overall design of a field data buoy system. The intent of these design specifications is not to give the specific component details, but rather to summarize functionally the prototype system and to suggest areas of modification for making the system more adaptable to a variety of Navy field and fleet operations.

The major areas in the prototype system needing modification are the high-power drain components of the system and the lack of a remote (non-cabled) operational capability. The components needing improvement are discussed along with suggested modifications in the following paragraphs. Also discussed are means of freeing the buoy system from a data (and power) cable to the shore site by use of a suggested telemetry system.

As given in the previous part of this section, this data buoy system consists of a two-component, cartesian coordinate current meter and a wave and tide gauge. These are the instruments for measuring the required parameters for the predictions models developed by LSU personnel for this study. The tidal information is needed to give the average water depth at the time of the wave readings. This information could be supplied by making a depth reading at the time the buoy was implanted and also using a mean tide reading in addition. This, however, would not be as accurate, particularly for deployment in lesser water depth, as using the measured tide from the system. Unless the wave gauge on the submerged buoy has a tide leak, the tidal information is mostly already there either to be filtered off or processed digitally by statistically removing the mean from the wave data. Most pressure-transducer wave gauges that are used in not-too-deep waters would provide this total information concerning waves and tide. The concern is, in addition to the accuracy of the gauge, the power consumption of the instrument. It should be one of low-power drain as the one used in the NCSC data buoy system.

This is usually accomplished by using CMOS or other equivalent low-power circuitry.

In addition to the wave gauge, the other required instrument for providing the needed input information for the surf prediction model is a two-axis current meter. Past experience has demonstrated that rotor and vane current meters are not suitable for this application due to their low reliability of operation, particularly in waters where bio-fouling and other types quickly limit the operation of the system. For these reasons and because the techniques are so well advanced, an electro-magnetic induction current meter, such as the one used in the NCSC system, is strongly recommended. These meters, in addition, have higher response capabilities than the mechanical type meters, making wave following more accurate. An additional requirement, which may be part of the current meter, is a compass, unless the buoy is always either oriented in a certain direction or the orientation of the buoy measured when deployed. This is needed in the data processing for the prediction model.

State-of-the-art current meters and pressure-transducer wave gauges recommended for this system, in general, are well within the accuracy requirements. A wave height measurement in shallow water to an accuracy of one-tenth of a foot resolution (relative) should be well within the capabilities of most off-the-shelf gauges. For a three meter wave height, this corresponds to one part in one hundred. However, where the problem becomes more stringent is in the measurement at the bottom of the shorter period waves, where water-depth attenuation has reduced the signal level considerably. But once again, most state-of-the-art pressure-transducer wave gauges should have this capability to measure, as example, a wave with a period as short as 3 to 4 seconds in water depth of 5 to 20 meters to a relative accuracy of 3 centimeters. (Remember that absolute measurements of wave height is not required, but a relative reading of the peak to trough difference in the readings is used. Absolute reading of the tide (mean water level) is needed, but this can be less accurate by 2 to 4 times and still be satisfactory within the model limitations, except for very-shallow water applications.) In addition, current-meter accuracies of five degrees should be more than sufficient for this application. A study done on the effect of accuracy of the current, wave, and tide measurements has shown that the above given accuracy requirements are suffucient to keep the scatter in the prediction (due only to measurement error) to one or two percent for most typical situations. Repeating for emphasis, this scatter of one to two percent is only due to that produced by error in the measuring instruments and is not the prediction capability of the models. (See Section VI for a discussion of the prediction accuracy of the models themselves.)

In addition to the above instruments, the NCSC buoy system has two temperature gauges, one for water temperature and the other for internal buoy temperature. Two other instruments complete the package: a water leak detector and a calibration source. Each of these instruments add capabilities to the system but are not required for the primary task of predicting surf-zone parameters as given.

A major change recommended for the NCSC buoy system is in the area of power consumption. The present prototype system requires approximately 4 watts of power for all of the on-board functions. Reasonable battery packs for this application range from a modest size of 20 ampere-hours to a large size of 50 ampere-hours. For continuous operation, this would give only about one to two full days of operation of this system using the above range of battery sizes. Even with a twenty-five percent duty cycle of thirty minutes on every two hours, as an example, the system could only operate for four to eight days, depending on the battery-pack size and type. For most operations, this prohibits the collection of very much background information. And, for many operation, frequent changing of the battery pack would either be difficult or prohibitive. Therefore, either larger battery packages are required or a lower-power system is needed. The bulk and expense associated with the larger battery packages very quickly suggest that this is not the solution. Thus, a lower-power buoy system is needed.

The major area of power consumption in the NCSC buoy system is in the multiplexer/analog-to-digital-converter, requiring some two-thirds of the total systems power. The problem can be easily and efficiently corrected at a relatively low cost by modifying this part of the system. It is suggested that the multiplexer and the A-to-D converter be replaced with state-of-the-art CMOS devices instead of the power hungry NMOS devices currently being used. This part of the system was designed for a land-based application where power was not a consideration. In a remote-buoy operation, power is a prime consideration. There are many such CMOS devices that would meet the needs and requirements of this system. Two such suggested devices can be obtained from Analog Devices, Norwood, Massachusetts: for the A-to-D converter: AD7550, a 13-bit A/D, requiring 20 milliwatts total power or less; and for the multiplexer: AD7506, a 16-channel mux, requiring 2 milliwatts total. Most of the other chips in this part of the system are already CMOS. However, those that are not all have a corresponding CMOS version, most of the time pin-for-pin interchangeable. These changes would make the data interfacing and digitizing portion of the system one of the most efficient parts.

In addition, the current meter uses more power than necessary, although its consumption is not excessive. Going to all low-power chips in this meter would help the consumption. However, a large part of the drain is not so much in the amplifier circuitry of the measured signal as is the drive power in the magnet, producing the field in which the current is measured by the voltage induction of a moving conducter (the water) in a magnetic field. Once again, this current meter was designed for generalized applications. By reducing the drive to the magnet, the accuracy of the system is reduced, because the signal-to-noise ratio of the system is also reduced. This can be afforded to be done in this system since small current flow is not being measured; rather the flow associated with the gravity waves is the measured parameter. The amount that the drive power should be reduced is dependent upon at least two factors: the relative power savings compared to the total system and total time of operation; and the effect of the reduced accuracy on the requirements of the specific application of the buoy system.

If the suggested power savings are added to the NCSC data buoy system, total system power can be of the order of one-half to one watt. This would extend operational time for the prototype version a factor of up to almost an order of magnitude. One to two weeks or more of continuous operation would be feasible. Using a 25% duty cycle, one to two months of total operation could be obtained. These lengthened time periods without the need for battery servicing would make this buoy system suitable for most fleet and test operations.

A final proposed modification to the prototype buoy system is that of a telemetry transmitter, allowing the reception of real-time data from the buoy without the present need for a data cable to the shore site. This modification has been ordered by NCSC for their buoy and is expected to be delivered in the near future. The transmitter/receiver system is a model A-4 modified system manufactured by Telecommunications Enterprises, Panama City, Florida. It is to be a self-contained system with its own power source, capable of command interrogation from the receiving site. In this manner, the user can select to turn of the system (including the data buoy system on the bottom) to receive data. This type of operation will considerably extend the total operating time of the system unless the system is more frequently turned on than those intervals given above. Even in the "high-power" operation (100 milliamps) of this transmitter system, over-the-water ranges of 200-250 miles are expected, with the resultant battery life of this part of the system compatible with the time periods of the rest of the data buoy system.

SECTION II
SOFTWARE DEVELOPMENT

During this 22-month study period, over twenty software programs were developed for use in this study and for future applications. This large number of programs developed is the result of two basic factors of major consideration in this study: (1) the development of programs for different types of processors and (2) the development of different versions of similar programs allowing the end user to implement a particular program with minimal modification.

The objectives of the software development portion of the over-all study are:

1. To develop control, formatting, and processing software that could ultimately be used for predicting surf-zone parameters from measured off-shore data in any of the following applications:

   a. complete on-board data buoy operation (within the buoy)
   b. complete remote processing (only the raw data sent from the data buoy to the remote processor)
   c. a hybrid of the above with limited on-board processing and final processing done at the remote processing station

2. To develop display software that would allow an operator to obtain in real time both a quantitative and a visual, dynamic description of the environmental parameters being measured by the data buoy.

3. To develop programs for the test and evaluation of surf-zone prediction models and for testing and evaluating a data-buoy system.

Some of the program development was done for a specific objective or part of it, and some of the development was done for more than one objective. Almost all of the software given in this report was new development or extensive modifications of previously developed software. Some of it, however, had been developed in other similar studies and involved less extensive modifications for the specific application.

Four types of programs were developed in this study. They are:

1. Programs for checking, testing, and validating buoy data

2. Programs for displaying buoy data for operational applications

3. Programs for collecting and storing buoy data

4. Programs for buoy-data processing and predicting surf-zone parameters including model evaluation

Also, four different programming languages were used in this study: FORTRAN, BASIC, Assembly Langauge (Intel 8080), and HPL (language of the Hewlett Packard HP-9825 computer system). Not all programs were written in all four languages. Many were written in only one language -- that of the processor being used for that particular application or the type that is expected to be used. However, some were written in FORTRAN (often with Assembly Language subroutines), BASIC, and HPL.

Since the programs are expected to be used for a variety of applications, often the processor on which the program is being used determines the required programming language. Most of the programs developed in this study are not extensive and time-consumming by standards of large computers. This means that for processing and displaying done remotely, that is on raw data received from the buoy, the computer need not be extensive either in terms of speed, size, or support capabilities. The two Hewlett Packard computers used in this study, HP 9825 and HP 9845, as well as the Dynex COMPAL-80 microcomputer system are examples of this, and their performance in this study well demonstrated the feasibility of their application. Of the langauges given, most similar types of computers use versions compatible with those given. The FORTRAN IV used is ANSI 1966 standard with a few extensions (that most FORTRAN systems have today). Thus, these programs should be directly adaptable with only minor modifications for most systems using FORTRAN compilers. The procedure for use on another system would be to take the source program as given in this report and compile it using the new computer FORTRAN compiler and then to link it with the host library routines and those of the compiled program.

For programs to be used directly within the data buoy itself, there are several methods of application. If the on-board processor is an Intel 8080 or a generation related to it or one that has as a subset of its assembly language the Intel 8080 code (e. g. Zilog Z80, as well as the later Intel processors), the code can be used as it is or with minimal modification. Once again, the source code can be assembled using the assembler of the candidate processor, converting it to machine code, which can be stored in PROM (programmable read only memory) or loaded from tape or other storage media directly into memory. This is also the situation for the FORTRAN programs that are to be used on-board the data buoy. The compiler must produce machine code compatible with the microprocessor that is being used. With a good, efficient compiler and FORTRAN library routines, the machine language code produced should be almost as efficient as the code produced by an assembler, with considerably less effort involved. Most programmers well know the advantages of higher-level programming.

The remainder of this section gives a generalized discussion of the various programs developed in this study. These programs are divided by category following the program types given above. In addition to the program description given are listings and flow diagrams for almost all programs. Most of these follow at the end of the section. The major exception is for the programs for the surf-zone prediction models. Fuller discussions, complete with detailed flow diagrams and listings, are given in the following two sections, Section III for the simplified model and Section IV for the detailed model. Where more than one langauge version of a particular program was written, the separate program listings for each version are given. However, the program discussion is generalized such that the over-all program is discussed, then the specific details of each of the different language are discussed where appropriate and needed. The flow diagrams are functional in nature and are not specific as to individual detail; therefore, separate diagrams for the different versions are not required for understanding each version of the programs.

Part I
Data Validating and Checking

Generalized routine for decoding and latching raw data stream. This routine is being given as a separate program although its use in this study almost always was as a subroutine within another program. However, because of its importance in the other programs it is being given and discussed separately. This program takes the raw data from either the buoy directly or recorder data from a digital tape recorded and does the control input routines for the computer's serial port. The data is sent or recorded serially in RS-232c format at either 300 baud for data from the buoy or 1200 baud for data from the CERC Field Research Facility at Duck. As previously given, the raw data is formatted such that no additional formatting or computer interfacing is required other than a standard RS-232c serial input/output port. The control program sets up the serial port (the UART -- Universal Asynchronous Receiver Transmitter) to receive 8 bits of data, even parity, and 1 stop bit. The status of the input port is checked to ascertain whether a data byte is present or not. When there, the parity flag on the UART is checked to see whether the byte contained even or odd parity. (This allows the determination of whether or not the byte is a high byte (with the data word address contained in it) or whether it is a low byte. Refer to discussions in Section I and in particular to the two Dynex reports referenced in that section for more details of the data format and hardware description.)

The hi byte is located, at which time the word address is stripped off and checked. When the first channel address is found, the data stream is in synchronism, and the data are ready to be read into the computer for use by the processing program (either for checking, displaying, or storing for later processing). In general, the program continues to check the address on each data word to ascertain that lock on the data stream is not lost and that valid data (as far as address is concerned) are being received. Functionally, the various language programs are identical. The various program language versions are given in Program Listing II-1 and the flow chart in Figure II-1.

Routine for checking individual data. This routine has two parts that input the raw data either directly from the data buoy or from the recorded data from the buoy. The first part of the routine is the latching routine described on the previous page. Once the data stream is latched at the selected address (usually the first address channel -- address zero or channel one), the checking portion of the routine begins. There are several sub-parts to this checking routine, and the actual manner in which the data are checked depends on the specific program. There is one for the HP-9825 and one for the COMPAL-80.

In the HP-9825 program the address of each data word is checked, and, if an error is encountered, a counter can be incremented and displayed, allowing the operator to determine whether the data stream is coming in reliably as far as the addresses of the data are concerned. Next the routine can select either a particular channel to display or all channels can be displayed sequentially. Several different versions of this routine have been used, although only one is given in this report. Before displaying, the data word is converted from high-byte/low-byte format into a voltage using the conversion factor associated with the analog-to-digital converter, and then it is displayed in engineering units using the conversion factor associated with the particular instrument making the measurements. This routine is particularly useful in reading and checking the calibration voltages in the system to determine system drift or other problems in the operation.

The COMPAL-80 version of checking the data is considerably more simple in scope, but gives the operator a far more detailed description of the actual data bytes as they are received. The routine displays each data byte in hexadecimal format as a continuous stream across the CRT display. The address all line up vertically, making continuous checking and observation of trends very easy to the trained observer. This routine does not allow the operator to directly check the actual engineering-unit value of the data channels since conversion from hexadecimal bytes to engineering unit words must be done. It is excellent for checking the over-all operation of the system, however. The program listings are given in Program Listing II-2 and the flow charts in Figure II-2.

Data checking and validating using FFT processing. Because it was felt that some of the more important data being used to test the surf-prediction models were not valid, it was decided to use more sophisticated methods of checking and validating the data because of the importance of the usage of the data. On later data received from the CERC facility at Duck, N. C., it was discovered from careful analysis that two pairs of the wave-gauge data were contaminated, probably with cross-talk of one channel leaking into the other. Because this could not be ascertained completely by directly observing the data, more reliable statistical means were employed to check these data. This was done with the use of cross-spectral analysis. (More details of the Duck-data are given in Section VI.)

This routine uses as the core of the processing the Fast Fourier Transform routine, very widely used and described in other literature and will not be discussed in detail in this report. Basically the routine takes in the data by selected channel pairs, each containing 128 data points, which are time-weighted using a cosine-squared weighting technique to minimize edge effects of the records. Next the means and RMS's of the two channels are computed, and the FFT coefficients are also computed. From these coefficients the auto spectra and the co- and quad-spectra are formed. From these spectra the coherence and phase spectra are calculated with the resultant matrices printed out or displayed on the CRT. This program was used to verify that indeed the data suspected were invalid and that the use of them would lead to erronous results in testing the prediction models.

Although the FFT routine given in this program is rather extensive in terms of programming steps required, it more than makes up for this in its efficiency in running. In general, it is a faster routine than others tested against it. However, the main advantage of this routine is that it uses all real data inputs in both the real and imaginary arrays. A subroutine, RLTRAN, converts the results to all real values in the spectra, meaning that a 128-data array of real values will yield 64 spectral lines in the results. Hanning (frequency smoothing) is also part of this program. The listing is given in Program Listing II-3, and a very simplified flow diagram is given in Figure II-3.

Part II
Data Displaying Routines

Routine for printing or displaying data in engineering units. This program uses the routine for latching the data stream that is received either directly from the data buoy or recorded data in the manner previously described. Next all data channels are converted to data words, next to voltages, and finally to engineering-unit data words, using the appropriate conversion factors. The program cycles through eight sets of sixteen channels of data, either putting the results into a data array or storing them in memory to recover in an array when the processing begins.

Once the converted data are put into the designated array, the data are formatted and printed out by function, with water level (waves) given first, tide next, followed by on-shore and long-shore currents, water and internal buoy temperatures are given next, with the last print out for the set being the calibration voltage of the system. There are 64 wave readings, 8 tide readings, 16 readings each of the current components, and 8 readings each for the temperatures and calibration voltage. At the end of the print out or display on the CRT, the program automatically recycles and starts a next series of readings until terminated by the operator.

Such a program was developed originally for checking and validating the buoy data but has many applications. For a remote buoy system, the operator can either call-up the buoy shore site station, as in the case of the NCSC buoy set-up, or receive directly via telemetry the data. Then this program can be executed given the results in real-time for the operator to make immediate observations of the environmental conditions at the buoy site. If the buoy is tied into a shore site with a modem, the buoy can be called from anywhere in the world that the number can be dialed or called, requiring only that the operator have a standard acoustic coupler or modem and reliable telephone communication. This gives a very effective way for real-time observations. A sample print-out of this program with real data is given in Figure II-00, a flow chart in Figure II-4, and the program listings in Program Listing II-4.

Figure II-0. Sample Printout of Data Buoy Real-time Monitoring.

WATER LEVEL (WAVES) IN CENTIMETERS (READ DOWN)

12.95	-1.94	-46.30	-27.54	12.06	8.48	-17.12	-37.07
15.63	-4.91	-42.13	-12.95	15.03	7.59	-23.67	-34.09
12.65	-16.52	-36.77	-11.76	12.95	1.04	-30.82	-28.73
16.23	-23.67	-35.28	-13.55	4.32	1.04	-32.30	-34.39
22.78	-33.20	-37.07	2.53	8.78	-2.53	-33.20	-28.73
18.61	-37.37	-32.60	1.64	13.55	-11.17	-27.84	-25.75
15.03	-40.05	-23.97	2.83	15.03	-16.82	-30.52	-20.99
-.45	-43.32	-19.50	3.13	12.65	-24.86	-34.69	-15.04

TIDE LEVEL IN METERS

5.43	5.44	5.45	5.42	5.38	5.40	5.43	5.48

ON-SHORE CURRENT (Y) IN CM/S (READ DOWN)

-4.84	-14.52	7.81	16.00	5.58	-9.31	-1.86	7.07
-11.54	-1.86	18.98	12.28	-3.35	-8.56	-.37	13.02

LONG-SHORE CURRENT (X) IN CM/S (READ DOWN)

2.60	4.09	4.84	.37	3.35	4.84	.37	-2.61
.37	5.58	.37	.37	7.07	4.09	-1.12	-3.35

WATER TEMPERATURE IN DEG C

18.02	18.02	18.02	18.02	18.02	18.02	18.02	18.02

INTERNAL BUOY TEMPERATURE IN DEG C

22.23	21.50	21.43	21.31	21.43	21.36	21.58	21.28

CALIBTRATION SIGNAL IN VOLTS (REF=0.0)

.001	.001	.001	.001	.001	.001	.001	.001

T

WATER LEVEL (WAVES) IN CENTIMETERS (READ DOWN)

3.42	18.61	-35.28	-53.74	16.52	22.78	-8.49	-43.92
9.38	17.12	-48.68	-36.47	24.86	20.10	-18.01	-41.83
10.87	14.44	-47.49	-26.95	23.37	13.55	-21.59	-37.96
22.18	12.95	-53.44	-15.63	25.46	7.00	-28.73	-43.62
21.88	1.93	-56.12	-16.82	30.22	-1.64	-29.62	-37.07
32.01	-10.27	-54.04	-3.13	26.05	-3.42	-40.94	-30.82
33.49	-16.82	-57.61	3.13	23.07	-8.78	-38.26	-31.11
24.56	-23.07	-51.66	10.57	23.67	-13.55	-38.26	-24.27

TIDE LEVEL IN METERS

5.42	5.40	5.45	5.41	5.38	5.37	5.45	5.44

ON-SHORE CURRENT (Y) IN CM/S (READ DOWN)

10.05	-14.52	.37	22.70	9.30	-6.33	-5.58	7.81
-8.56	-15.26	14.51	15.26	2.60	-8.56	1.11	19.72

LONG-SHORE CURRENT (X) IN CM/S (READ DOWN)

-3.35	4.09	4.84	-2.61	-.37	6.33	1.11	1.86
2.60	4.09	-1.86	12.28	4.09	5.58	3.35	3.35

WATER TEMPERATURE IN DEG C

18.02	18.02	18.02	18.02	18.02	18.02	18.02	18.02

INTERNAL BUOY TEMPERATURE IN DEG C

21.43	21.53	21.33	21.28	21.33	21.36	21.33	21.50

CALIBTRATION SIGNAL IN VOLTS (REF=0.0)

.004	.001	.001	.001	.001	.001	.001	.001

Routines for dynamically displaying wave data. Two programs were written for dynamically displaying in real time wave information either from the data buoy or from the field facility at Duck, N. C. The first program, for the HP-9825, displays only one parameter at a time due to the speed and display limitations of the system. As in the previously described programs, this routines latches onto the data stream, determines the selected channel, and converts that channel to the appropriate units. Then the program wipes the display clean, puts on a wave height axis in the appropriate units, and puts an "x" or any other designated ASCII character at the appropriate height location. As the next data point comes in for that channel, the previous point is blanked, and the new one is displayed. The cycle continues until the operator terminates the program. Actual wave motion (or current movement if selected) can be directly observed in real time in calibrated format. The operator can see directly, in real time, wave motion that is happening at a distant location, which could be as much as several thousand miles away.

The second program was written for the COMPAL-80 which can be used on other similar computers with a FORTRAN system and a serial I/O port. Because of the increased speed and display capabilities of this system, seven channels of data can be displayed in real time simultaneously. This format was selected specifically for the data received from the CERC field site at Duck. Seven wave gauges are being monitored, from the seaward end of the pier all the way in to the breaking region at the shoreward end of the pier. In a similar manner as given above, this routine takes the data from the seven wave channels, converts them to the appropriate units and displays them in real time with the designated character. Each channel has its own separate location on the CRT and is spaced laterally from the others making observation easier and direct. Almost four readings per second per wave gauge are displayed, giving a very dynamic picture of the wave situation and distribution at that site.

The program could be easily modified to display other parameters than waves. Or it could be modified to display the information from several different sites simultaneously. This could be one application for the situation where several data buoys were implanted off-shore of a potential operations or strike area. The observer could display each of the wave channels from each of the data buoys at the same time and in real time if the data were being transmitted in real time. Direct observations could lead to on-the-spot decisions, not based upon the results of one site extrapolated to another, but on the collective field of information. Such dynamic display capabilities can augment and complement other types of processing very well, offering an added dimension to the results.

The flow diagram is given in Figure II-5 and the listings in Program Listings II-5.

Digital strip-chart displaying. This program was developed for use in the data analysis and model prediction testing and evaluating. However, because of its extensive display capabilities and potential applications in this area, it is being included in this part rather in the part on data processing. This routine requires a graphic display with medium to high resolution. State-of-the-art technology in displays and video drivers have made this requirement one that is easily within the reach of most systems today. The HP-9845 has very excellent capabilities in this area and was selected for this part of the study (primarily for the data processing and model testing). Any display with a dot resolution of 400 to 500 dots or so on a side is very adequate.

Because the HP-9845 was not able to keep up with the data recordings from the Duck facility in real time (thus making data reduction times considerably longer than data collect times), the data were read, converted, and stored on disk by the HP-9825. This routine is discussed in a later part on data collection and storage. The data were then read into the computer from the disk storage, the means computed and removed from the individual data points for each wave gauge channel, and the resultant data appropriately converted to height units and formatted. Then the operator selected a subset of wave data for each channel, and the results were plotted across the CRT in a strip-chart manner. The appropriate time grids along the horizontal axis and the height grids along the vertical axis were added to make the display complete. In this manner, one could have in view as much as five minutes or more of wave records for all seven wave gauges. The display gives a very detailed time history of the records for each of the gauges. Trends can be observed and easily followed. Shoaling, breaking, and other similar effects can be either directly or indirectly observed through this process. Its application to this study is discussed in later parts of this report. Its application to other situations and fleet operations follow much of that given above for the other display programs. It can also be used in other research programs in which wave modification studies are being done.

The flow diagram is given in Figure II-6 and the program listing in Program Listing II-6.

Part III
Data Collection and Storage

Routine for collecting and storing data from the buoy. A program was written both for the HP-9825 and the COMPAL-80 for collecting, formatting, converting, and storing the data from the buoy, either taken directly or, as used almost all of the time, from recorded digital tapes. The program uses the latching routine described above to locate the proper address for starting the sequence. The data from each wave-gauge channel (1,3,5, 7,9,11,13,15) is then is put into a long string array in byte format to save space both in memory (most important for the HP-9825) and on tape. In addition, each of the other channels were also put into the same string such that all of the data taken from the buoy could be recorded on a tape that could be used directly in the HP 9825. Approximately ten minutes of continuous data were stored in each set, representing over 3000 data words of wave height and an equal number for the other parameters totalled together. (Note that this data is from the buoy having the format given in Section I and is for only one wave gauge -- not seven as in the CERC data from Duck.) At the end of the ten-minute period, the data are stored on tape for later processing. The cycle can be repeated for all of the data that is to be stored in ten-minute sets.

The FORTRAN program on the microcomputer is functionally the same as the HP-9825 program. The data are read directly into memory in byte format as above. However, instead of going into a string array directly as given, the data fill a memory array prior to being stored. Considerably more data points can be taken continuously by this system due to the large memory size. Although the program is set-up to do a given number in the set, this can very easily be made a variable. At the end of the data collection, the results are stored either on disk or on tape. The flow diagram is given in Figure II-7 and the listings in Program Listing II-7.

Routine to crunch data files. Due to the limited memory size of the HP-9825 that was in existence at the time of the data processing, there was not enough room in memory for the large string array given above and the required processing program. Therefore, a routine was written that took the large file stored by the previous program and crunched or divided it into three smaller files, each containing 1024 data points of wave information. The resultant smaller files were restored for use by the data processing programs in the next part. As in the previous program, there is a complete set of file identification information giving the test date, time, run number, and tape file number. This information was stored in the last string of the array, thus preventing lost of identification of the data or other mix ups. The flow diagram is given in Figure II-8 and the program listing in Program Listing II-8.

Part IV
Data Processing and Model Testing

Data processing program for the simplified model. Section III gives a complete discussion of the simplified model developed for this study by J. N. Suhayda of Louisiana State University and the programming used in testing this model. Refer to that section and reports by LSU for that information and program listings. Also refer to Section VI for results of the data processing and analysis of the model testing.

Data processing for the detailed model. Section IV gives a complete discussion of the detailed model program including a brief background of the equations programmed for the model testing. This moiel was also developed by Suhayda of LSU. Reference is made to his reports for details of the model itself and to Section IV for the programming details and Section VI for the results of the data processing and analysis of the model testing.

Modified processing for the detailed model. This routine was developed for processing the data from the CERC field site at Duck, N. C. Because the wave data were obtained from the readings on the strip-chart digital displays described earlier and in Section VI, the detailed processing program was modified to input from the keyboard the wave height and period read off of the strip charts. The program then used this information to compute predicted breaker information identically as in the program given in the later section of this report, the only difference being the method of inputting the information. Because this program may have other applications than that given here, it is being included in this report. One such application might be a field site where the input information was obtained in a form that could not be read into the computer in a manner as given here, that is, through one of the computer I/O ports. As an example, the data might have been read by an observer directly or from a device not compatible with the computer. Thus the input data could be entered manually through the keyboard, and the prediction results could be displayed or printed out as before. The flow diagram is given in Figure II-9 and the listing in Program Listing II-9.

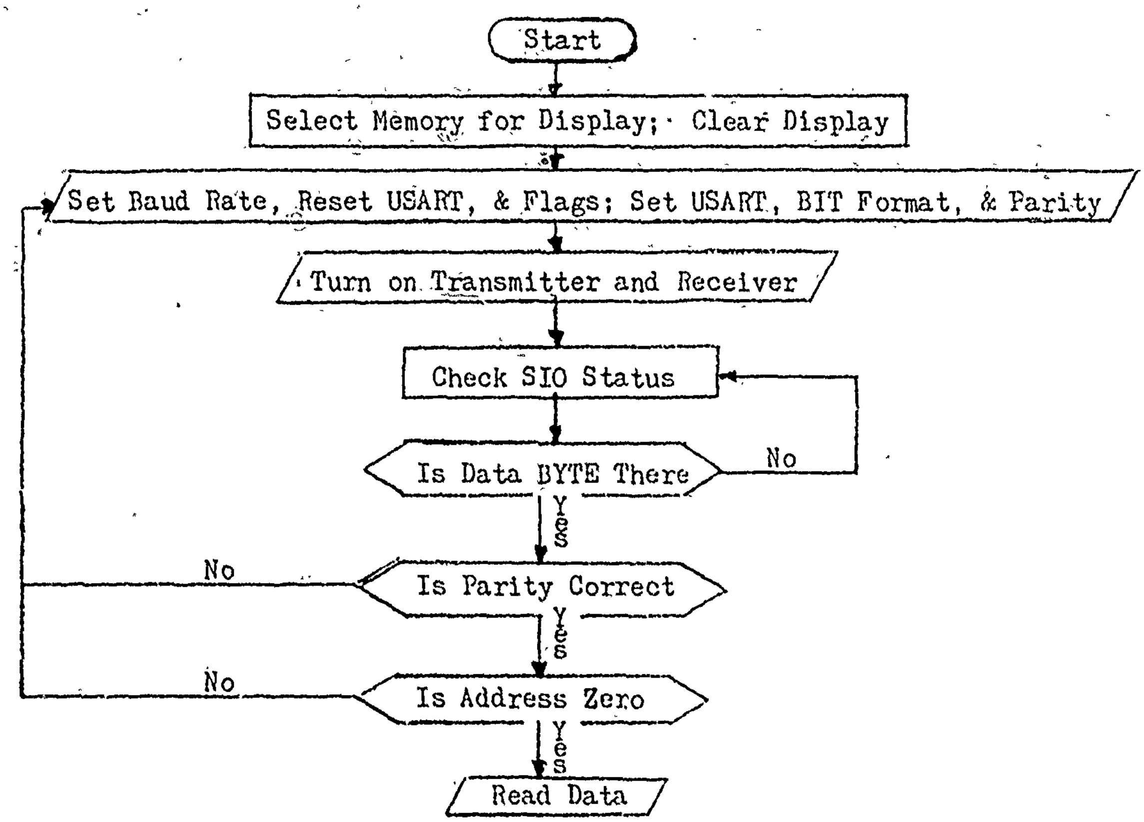
Wave time-of-arrival at successive wave gauges for Duck CERC data. This routine was written to aid the strip-chart reader in following wave trains from one gauge to the next. This was particularly important for the situation where processing was done on data with the number 2 wave gauge not working. This left an interval of almost 300 meters between the seaward wave gauge, which was being used as the model input source, and the next working gauge. This meant that there were many waves between the two making identification of one wave at the next gauge as being the specific one from the previous gauge a dificult task in many instances.

For a large, singular wave, its signature was unique enough such that following it from one gauge to the next was not difficult. However, for the waves that were not that unique, tracing its shoreward motion from one gauge to the next was more difficult. It was determined that if the strip-chart reader knew approximetely where in time to look for a particular wave at the next closer wave gauge, the task would be easier. To this end, this routine was written.

The routine has internal in a depth array the actual depth measurement taken from the field site, and this array is stored on disk such that it is a simple task to input other depths. An auxiliary program was written for storing this data on disk. Using linear wave theory, the routine computes the expected wave lengths for each depth, which were taken at intervals less than 10 meters separation. Then for each depth the routine computes the wave speed. The process is repeated for wave periods from three to ten seconds. Longer wave periods could be used, but for those periods, the propagation times were not enough different for this particular application. It was sufficient enough to be able to locate the wave within its period, not to a particular point on the wave form.

After all of the wave speeds were calculated, the routine next computed, using wave speed and distance between stations, the time predicted to travel from one wave-gauge location to the next. These results were printed out along with other intermediate information in the program. The flow diagram is given in Figure II-10 and the listing in Program Listing II-10.

Figure II-1. Flow Chart of Generalized Routine for Decoding and Latching Raw Data Stream



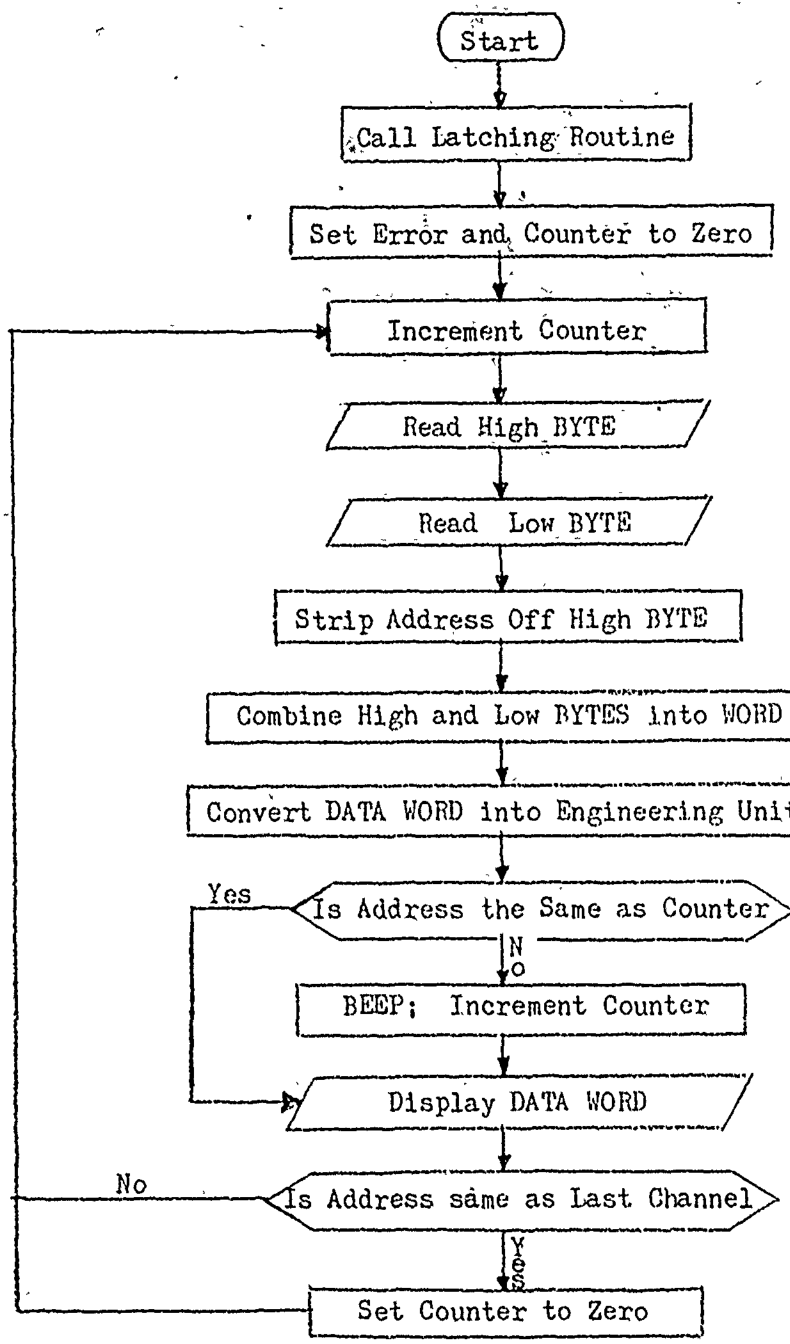
Program Listing II-1A. Generalized Routine for Decoding and Latching Raw Data Stream -- FORTRAN Version

```
A>TYPE LATCH.FOR
        PROGRAM LATCH
COMMENT PROGRAM READS IN SERIAL DATA STREAM THROUGH SIO PORT.
COM USART IS SET FOR EVEN PARITY. AT BEGINNING USART ANDALL FLAGS
COM ARE RESET. BYTE IS READ AND PARITY FLAG (08) IS CHECKED.
COM WHEN PARITY ERROR IS FOUND, THAT IS LOW BYTE; THUS NEXT BYTE
COM IS HIGH BYTE CONTAINING ADDRESS (4 MSB). ADDRESS IS CHECKED
COM UNTIL FIRST CHANNEL (0) IS LOCATED.
COM SERIAL PORT: E0HEX IS DATA; E1HEX IS STATUS; E4 IS BAUD RATE PORT
        DIMENSION IDATA(7)
        LOGICAL IDATA,IHI,LO,NUM,K
COM  FOR CRT MAPPING, GRAPHICS START AT F800HEX
        IREF=Z'F840'
        DO 105 I=1,7
105     IDATA(I)=0
COM  CLEAR THE SCREEN NEXT (127=BLANK)
        DO 106 I=1,1024
        MM=Z'F7FF'+I
106     CALL POKE(MM,127)
COM  SET UP AND RESET THE USART (SIO) NEXT
        IB=Z'E4'
        IS=Z'E1'
        ID=Z'E0'
COM   SET BAUD RATE TO 1200 BAUD
100     CALL OUT(IB,Z'39')
        CALL OUT(IS,Z'AA')
COM  RESET USART AND ALL FLAGS
        CALL OUT(IS,Z'50')
COM  SET USART TO 8 BITS DATA, EVEN PARITY, 1 STOP BIT, X16 CLOCK
        CALL OUT(IS,Z'7E')
        CALL OUT(IS,0)
COM  TURN ON USART TRANSMITTER AND RECEIVER
        CALL OUT(IS,5)
COM  CHECK SIO STATUS; IF BYTE THERE, CHECK PARITY BIT
110     ISTAT=INP(IS)
        IPAR=ISTAT
        ISTAT=(ISTAT .AND. 2)
        IF(ISTAT .EQ. 0)GO TO 110
COM  CHECK BIT 4; IF HIGH (08), PARITY ERROR = LOW BYTE,  LOOK AGAIN
        IPAR=(IPAR .AND. 8)
        IF(IPAR .NE. 0)GO TO 100
        IHI=INP(ID)
        IHI=(IHI .AND. 240)
        IF(IHI .NE. 160)GO TO 100
        CALL STATUS
        LO=INP(ID)
COMMENT HIGH BYTE FOR FIRST CHANNEL IS NEXT; DATA STREAMIS LATCHED.
COMMENT PROGRAM FOR USING DATA WOULD FOLLOW HERE ......
COMMENT MORE PROGRAM HERE .......................
        END
```

Program Listing II-1B. Generalized Routine for Decoding and Latching Raw Data Stream -- HPL Version

```
0: "program to latch data from port.Usart is set for even":
1: "parity.At beginning all flags are reset.Byte is read and parity flag":
2: "is checked.Whe parity error is found(low byte)the next byte is ":
3: "high byte.Address is checked until first channel is located.":
4: dsp "READY FOR DATA -- PUSH CONT";stp
5: wtc 11,1;wtb 11,64;wtb 11,255;wtb 11,20;wtc 11,0
6: rdb(11)→P;wtc 11,1;rdb(11)→P;wtc 11,0;band(P,8)→P;if P#8;jmp 0
7: rdb(11)→A;rdb(11)→B;shf(A,4)→A;dsp A,B;if A#10;jmp 0
8: end
*27164
```

Figure II-2. Flow Chart of Routine for Checking Individual Data



Program Listing II-2A. Routine for Checking Individual Data -- HPL Versions

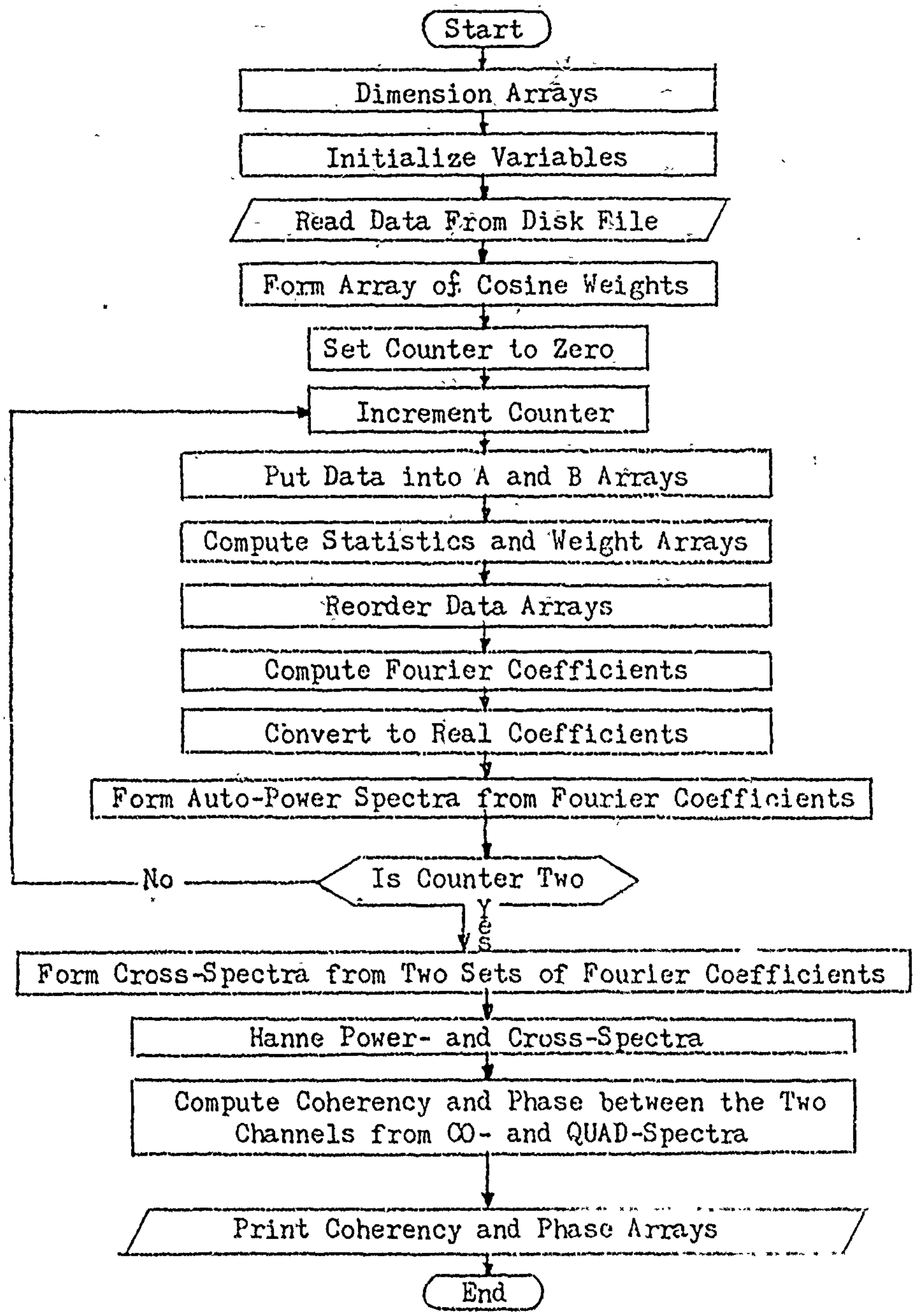
```
0: "program to check data from duck digital tapes":
1: "displays data points as voltages on CRT":fxd 3
2: dsp "READY FOR DATA -- PUSH CONT";stp
3: wtc 11,1;wtb 11,64;wtb 11,255;wtb 11,20;wtc 11,0
4: rdb(11)→P;wtc 11,1;rdb(11)→P;wtc 11,0;band(P,8)→P;if P#8;jmp 0
5: rdb(11)→A;rdb(11)→B;shf(A,4)→A;dsp A,B;if A#10;jmp 0
6: for J=1 to 512;for I=1 to 11
7: rdb(11)→A;rdb(11)→B;if I#8;jmp 3
8: shf(A,4)→C;band(A,15)→A;(256*A+B)*.001221→D;dsp D
9: if C+1#I;beep;E+1→E
10: next I;next J
11: end
*23543
```

```
0: "program to check data from digital tapes of buoy":
1: "displays data points as voltages on CRT":fxd 3
2: dsp "READY FOR DATA -- PUSH CONT";stp
3: wtc 11,1;wtb 11,64;wtb 11,255;wtb 11,20;wtc 11,0
4: rdb(11)→P;wtc 11,1;rdb(11)→P;wtc 11,0;band(P,8)→P;if P#8;jmp 0
5: rdb(11)→A;rdb(11)→B;shf(A,4)→A;dsp A,B;if A#15;jmp 0
6: for J=1 to 512;for I=1 to 16
7: rdb(11)→A;rdb(11)→B;if I#8;jmp 3
8: shf(A,4)→C;band(A,15)→A;(256*A+B)*.001221→D;dsp D
9: if C+1#I;beep;E+1→E
10: next I;next J
11: end
*15915
```

Program Listing II-2B. Routine for Checking Individual Data -- Assembly Language Version

```
ADDR  CODE       LABEL    OP CODE          COMMENTS

C000  3E 0D      CHECK:   MVI   A,'CR'     ;get ASCII code for carriage return
C002  CD 00 C2            CALL  STATOS     ;is CRT ready for display byte?
C005  D3 00               OUT   CRT        ;send it to CRT
C007  3E 0A               MVI   A,'LF'     ;get ASCII code for line feed
C009  CD 00 C2            CALL  STATOS     ;is CRT ready?
C00C  D3 00               OUT   CRT        ;send it to CRT
C00E  0E 20               MVI   C,20Hex    ;set counter to do 32 bytes/line
C010  CD 00 C1   LOOP:    CALL  STATIS     ;is data byte in serial I/O port?
C013  DB E0               IN    SIO        ;yes, get it
C015  CD 00 C2            CALL  STATOS     ;is CRT ready for data byte display?
C018  D3 00               OUT   CRT        ;yes, display it
C01A  0D                  DCR   C          ;decrement counter -- done 32 bytes?
C01B  C2 10 C0            JNZ   LOOP       ;no, keep going
C01E  C3 00 C0            JMP   CHECK      ;yes, go to next display line and do
                                           ;another set till operator resets you


                          ENTRY STATIS
C100  DB E1      STATIS:  IN    E1         ;check serial I/O port status
C102  E6 02               ANI   02         ;is data byte there?
C104  CA 00 C1            JZ    STATIS     ;no, look again
C107  C9                  RET              ;yes, go back to sender


                          ENTRY STATOS
C200  DB 01      STATOS:  IN    01         ;check CRT status
C202  E6 02               ANI   02         ;are you ready?
C204  CA 00 C2            JZ    STATOS     ;no, look again
C207  C9                  RET              ;yes, go back to caller

;comments -- CRT (display) data port is 00, CRT status port is 01,bit 01
;comments -- serial I/O port: data is E0hex and status is E1hex, bit 01
;
;
;program does a carriage/line feed at the beginning and sets up the display
;for displaying 32 bytes per line (each byte is two hexadecimal characters)
;at the end of 32 characters, a carriage return/line feed is done again
;and another set of 32 bytes is done
;program will continue until operator resets computer or no data is entered
;into the serial port (for the latter, program will continue to look for data
;but no more displaying will be done)
```

Figure II-3. Flow Chart of Data Checking and Validating Program using FFT Processing



Program Listing II-3. Data Checking and Validating Program Using FFT Processing -- FORTRAN Version

```
TYPE FFTCP.FOR
          PROGRAM FFTCP
C         DO 2 CHANNELS OF FFT AND COMPUTE COHO AND PHASE
C         DATA READ IN FROM DISK ON LUN 6
          DIMENSION W(64),A(65),B(65),AA(2,25),BB(2,25),DAT(2,128)
          DIMENSION COHO(25),PHA(25),CO(25),QD(25),POW(2,25)
          DIMENSION PW1(25),PW2(25)
10        FORMAT(1X)
20        FORMAT(1X,5F10.3)
88        FORMAT(1X,5E12.3)
          DO 100 I=1,65
          A(I)=0
100       B(I)=0
          NT=1
          NF=5
          DEG=180.0/3.14159
          CNV=1.0
          CALL WEIGHT(W,CNV)
          READ(6)DAT
          ENDFILE 6
          DO 200 J=1,2
          DO 210 I=1,64
          A(I)=DAT(J,I)
210       B(I)=DAT(J,I+64)
          CALL CALC(A,B,W,CNV)
          CALL REORDR(A,B)
          CALL RVFOUR(A,B)
          CALL RLTRAN(A,B)
          DO 220 I=1,25
          AA(J,I)=A(I)
          BB(J,I)=B(I)
220       POW(J,I)=A(I)*A(I)+B(I)*B(I)
200       CONTINUE
          DO 400 I=1,25
          PW1(I)=POW(1,I)
400       PW2(I)=POW(2,I)
          DO 300 I=1,25
          CO(I)=AA(1,I)*AA(2,I)+BB(1,I)*BB(2,I)
300       QD(I)=AA(1,I)*BB(2,I)-BB(1,I)*AA(2,I)
          CALL HANNE(PW1,NF,NT)
          CALL HANNE(PW2,NF,NT)
          CALL HANNE(CO,NF,NT)
          CALL HANNE(QD,NF,NT)
          DO 310 I=1,25
          COHO(I)=(CO(I)*CO(I)+QD(I)*QD(I))/(PW1(I)*PW2(I))
          PHA(I)=ATAN2(QD(I),CO(I))
310       PHA(I)=DEG*PHA(I)
          PAUSE PRINT
```

Program Listing II-3 (cont)

```
        WRITE(5,20)COHO
        WRITE(5,10)
        WRITE(5,20)PHA
        END

A>TYPE WEIGHT.FOR
        SUBROUTINE WEIGHT(W,CNV)
        DIMENSION W(64)
        U=0
        DO 704 I=1,64
        W(I)=1-COS(I*.04908738)
 704    U=U+W(I)*W(I)
        COV=CNV/(22.6274*SQRT(2*U))
        DO 705 I=1,64
 705    W(I)=W(I)*COV
        RETURN
        END

A>TYPE CALC.FOR
        SUBROUTINE CALC(A,B,W,CNV)
        DIMENSION A(65),B(65),W(64)
        S1=0
        S2=0
        DO 601 I=1,64
        S1=S1+A(I)+B(I)
 601    S2=S2+A(I)*A(I)+B(I)*B(I)
        AVG=S1/128
        RMS=SQRT(ABS((S2-S1*S1/128)/(127)))*CNV
 602    FORMAT ('  AVG:',F8.4,'   RMS:',F8.4)
        AVE=AVG*CNV
        WRITE (5,602) AVE,RMS
        DO 605 I=1,64
        A(I)=(A(I)-AVG)*W(I)
        K=65-I
 605    B(I)=(B(I)-AVG)*W(K)
        RETURN
        END

A>TYPE REORDR.FOR
        SUBROUTINE REORDR(A,B)
        DIMENSION A(65),B(65),IC(7),LST(7)
        PI=3.14159265
 200    FORMAT(' ','REORDR')
        WRITE(5,200)
        IC(1)=1
        N=1
```

Program Listing II-3 (Cont)

```
        DO 204 K=2,7
        N=N+N
204     IC(K)=N
        J=6
        KB=1
        DO 205 K=1,63,2
        T=A(K+1)
        A(K+1)=B(K)
205     B(K)=T
        I=0
201     KS=IC(J)+KB
        KU=KS
        JJJ=8-J
        JJ=IC(JJJ)
        KK=KB+JJ
202     K=KK+JJ
203     T=A(KK)
        A(KK)=A(KS)
        A(KS)=T
        T=B(KK)
        B(KK)=B(KS)
        B(KS)=T
        KK=KK+1
        KS=KS+1
        IF(KK .LT. K) GO TO 203
        KK=KK+JJ
        KS=KS+JJ
        IF(KK .LT. KU) GO TO 202
        IF (J .LT. 6) GO TO 206
        J=J-1
        I=I+1
        LST(I+1)=J
        GO TO 201
206     IF (I .EQ. 0) GO TO 207
        J=LST(I+1)
        I=I-1
        KB=KS
        GO TO 201
207     RETURN
        END

A>TYPE RVFOUR.FOR
        SUBROUTINE RVFOUR(A,B)
        DIMENSION A(65),B(65),CC(7),SS(7),JC(7),ID(7)
        PI=3.14159265
300     FORMAT(' ','RVFOUR')
        WRITE(5,300)
        N=1
        JC(1)=1
        DO 301 K=2,7
        N=N+N
301     JC(K)=N
        RAD=PI*2/64
```

Program Listing II-3 (Cont)

```
      JJ=31
      DO 302 K=1,5
302   ID(K)=32-JC(K)
      DO 308 KB1=1,63,2
      KB=63-(KB1-1)
      ISPAN=1
      J=5
      K=JJ
304   IF(K .LT. 16) GO TO 305
      CN=-SIN((K-16)*RAD)
      CC(J)=CN
      SN=SIN((32-K)*RAD)
      SS(J)=SN
      GO TO 303
305   CN=SS(J)
      SN=-CC(J)
303   KKK=KB+ISPAN-1
      DO 306 KK1=KB,KKK
      KK=KKK-(KK1-KB)
      KS=KK+ISPAN
      RE=A(KK)-A(KS)
      A(KK)=A(KK)+A(KS)
      RIM=B(KK)-B(KS)
      B(KK)=B(KK)+B(KS)
      A(KS)=CN*RE-SN*RIM
306   B(KS)=SN*RE+CN*RIM
      IF(JJ .GE. ID(J)) GO TO 307
      JJ=JJ+JC(J)
      J=J-1
      ISPAN=ISPAN+ISPAN
      IF(J .LT. 1) GO TO 309
      K=K+K
      GO TO 304
307   JJ=JJ-JC(J)
308   CONTINUE
309   KS=32
      ISPAN=32
      DO 310 KK=1,32,1
      KS=KS+1
      RE=A(KK)-A(KS)
      A(KK)=A(KK)+A(KS)
      A(KS)=RE
      RIM=B(KK)-B(KS)
      B(KK)=B(KK)+B(KS)
310   B(KS)=RIM
      RETURN
      END

A>+S
```

Program Listing II-3 (Cont)

```
TYPE RLTRAN.FOR
        SUBROUTINE RLTRAN(A,B)
        DIMENSION A(65),B(65)
111     FORMAT(' ','RLTRAN')
        WRITE(5,111)
        PI=3.14159265
        RAD=PI/64
        R=-(2*SIN(.5*RAD))**2
        SD=SIN(RAD)
        CD=-.5*R
        CN=1
        SN=0
        A(65)=A(1)
        B(65)=B(1)
        DO 401 J=1,33,1
        K=66-J
        AA=A(J)+A(K)
        AB=A(J)-A(K)
        BA=B(J)+B(K)
        BB=B(J)-B(K)
        RE=CN*BA+SN*AB
        RIM=SN*BA-CN*AB
        B(K)=RIM-BB
        B(J)=RIM+BB
        A(K)=AA-RE
        A(J)=AA+RE
        CD=R*CN+CD
        CN=CD+CN
        SD=R*SN+SD
401     SN=SD+SN
        RETURN
        END


A>TYPE HANNE.FOR
        SUBROUTINE HANNE(HA,NF,NT)
        DIMENSION HAN(45),HA(25)
        DO 815 I=1,45
815     HAN(I)=0
        IF (NF .EQ. 0) GO TO 803
        DO 800 I=1,25
        K=I+NF
800     HAN(K)=HA(I)
        DO 801 I=1,NF
        K=I+NF
        J=NF-I
        HAN(J+1)=HAN(K+1)
801     HAN(K+25)=HAN(J+25)
        J1=NF+NF
        DO 802 K=1,J1
        J2=25+(2*NF)-K
        DO 802 I=1,J2
802     HAN(I)=HAN(I)+HAN(I+1)
        GO TO 805
```

Program Listing II-3 (Cont)

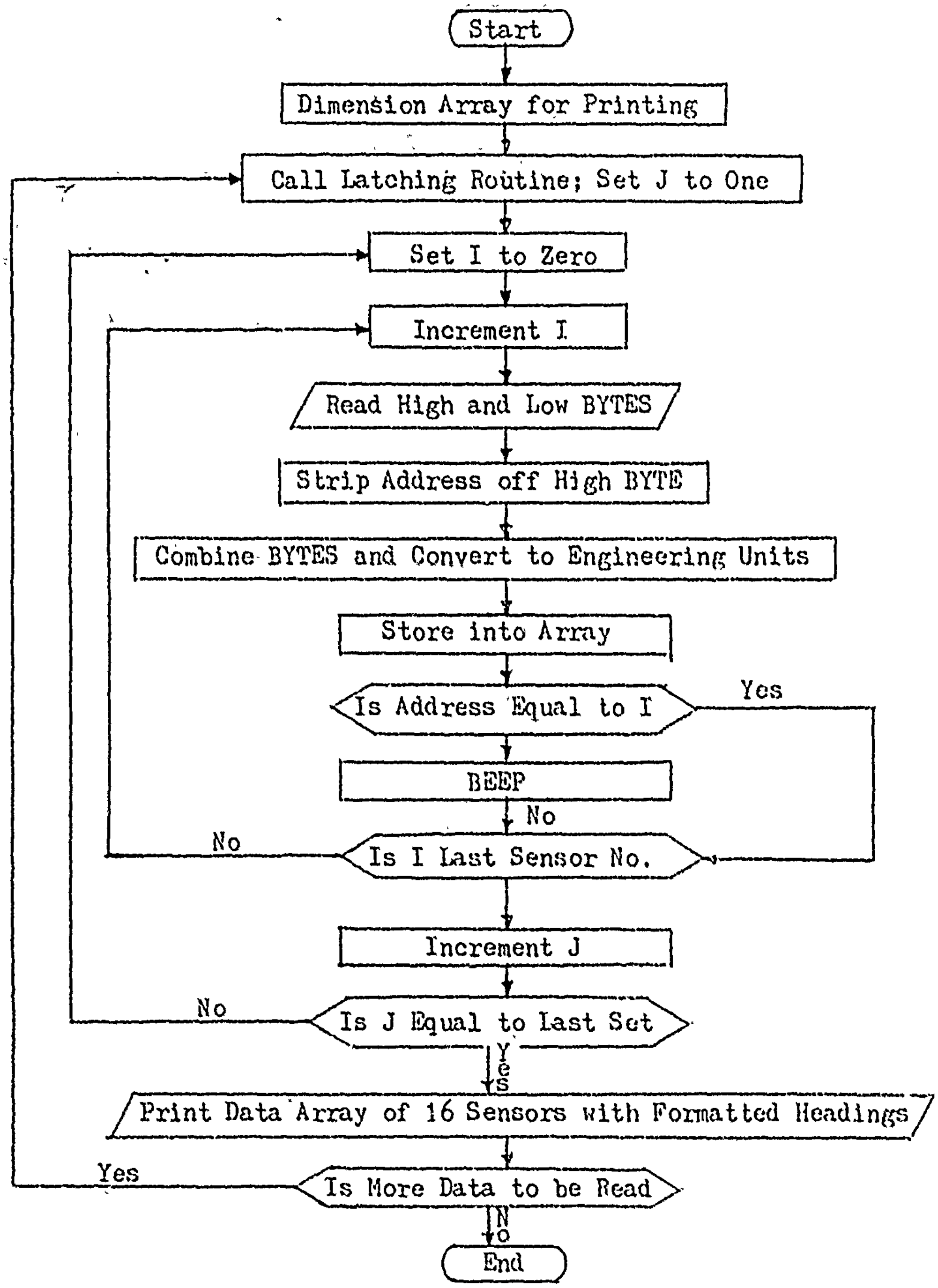
```
803       DO 804 I=1,25
804       HAN(I)=HA(I)
805       F=NT*2**(2*NF)
          F=2/F
          DO 806 I=1,25
806       HA(I)=F*HAN(I)
          RETURN
          END
```

A>

.720	.631	.603	.801	.928
.971	.983	.982	.970	.956
.952	.950	.933	.897	.852
.835	.834	.763	.626	.438
.200	.156	.249	.339	.384

-4.699	-3.772	4.734	12.271	13.901
13.604	12.614	10.830	7.415	1.505
-5.315	-9.819	-11.551	-11.304	-9.023
-3.747	4.328	12.957	21.062	24.806
7.146	-40.209	-63.270	-70.690	-72.588

Figure II-4. Flow Chart of Routine for Printing or Displaying Data in Engineering Units



Program Listing II-4A. Program for monitoring in real time data received from Data Buoy System -- FORTRAN Version

```
      PROGRAM BOPRTM
C     PROGRAM NAME IS BOPRTM READS DATA FROM MEMORY AND CHECKS
C     ADDRESS AND PUTS DATA INTO 1024-DATA ARRAY
C     PRINT OUT RESULTS IN ENG UNITS BY SENSOR FUNCTION
C     PROCEDURE IS TO EXEC BOASSM WHICH LOADS ASSEMBLY
C     LANGUAGE PROG INTO MEMORY AT EC00HEX FOR CALL BY ASSM
C     1024-DATA READ INTO MEMORY LOCATION B000HEX
C     PROGRAM RECYCLES UNTIL TERMINATED AFTER PAUSES
COMMENT
COMMENT  PROGRAM MODIFIED 11/13/79 VWF
COMMENT PROGRAM MODIFIED FOR METRIC SYSTEM 2/20/80 VWF
      DIMENSION IADD(16),CAL(16),DAT(16,8)
10    FORMAT(1X,'DATA ADDRESSES 1 TO 16')
20    FORMAT(1X)
25    FORMAT(7X,'WATER LEVEL (WAVES) IN CENTIMETERS (READ DOWN)')
30    FORMAT(1X,8F7.2)
35    FORMAT(20X,'TIDE LEVEL IN METERS')
40    FORMAT(1X,8F7.3)
45    FORMAT(10X,'ON-SHORE CURRENT (Y) IN CM/S (READ DOWN)')
55    FORMAT(10X,'LONG-SHORE CURRENT (X) IN CM/S (READ DOWN)')
65    FORMAT(20X,'WATER TEMPERATURE IN DEG C')
75    FORMAT(15X,'INTERNAL BUOY TEMPERATURE IN DEG C')
85    FORMAT(15X,'CALIBTRATION SIGNAL IN VOLTS (REF=0.0)')
90    FORMAT(1X,17I3)
COMMENT SET CALIBRATION/CONVERSION FACTORS NEXT
COMMENT CHANNELS 1,3,5,7,9,11,13,15 ARE WAVE DATA
COMMENT CHANNEL 6 IS TIDE DATA
COMMENT CHANNELS 4,12 ARE ON-SHORE CURRENT DATA
COMMENT CHANNELS 2,10 ARE LONG-SHORE CURRENT DATA
COMMENT CHANNELS 8,16 ARE WATER TEMP AND INTERNAL TEMP DATA
COMMENT CHANNEL 14 IS VOLTAGE CALIBRATION DATA
      CM=30.48
      DO 110 I=1,15,2
110   CAL(I)=4.0*CM
      CAL(2)=10.0*CM
      CAL(6)=4.87*CM/100.0
      CAL(4)=10.0*CM
      CAL(8)=10.0
      CAL(10)=10.0*CM
      CAL(12)=10.0*CM
      CAL(16)=10.0
      CAL(14)=1.0
COMMENT INITIALIZE ARRAYS NEXT
334   DO 5 J=1,8
      DO 5 I=1,16
5     DAT(I,J)=0
      DO 7 I=1,16
7     IADD(I)=0
COMMENT INPUT DATA THRU I/O PORT INTO MEMORY STARTING AT B000HEX
      CALL ASSM
      MEM=Z'B000'
```

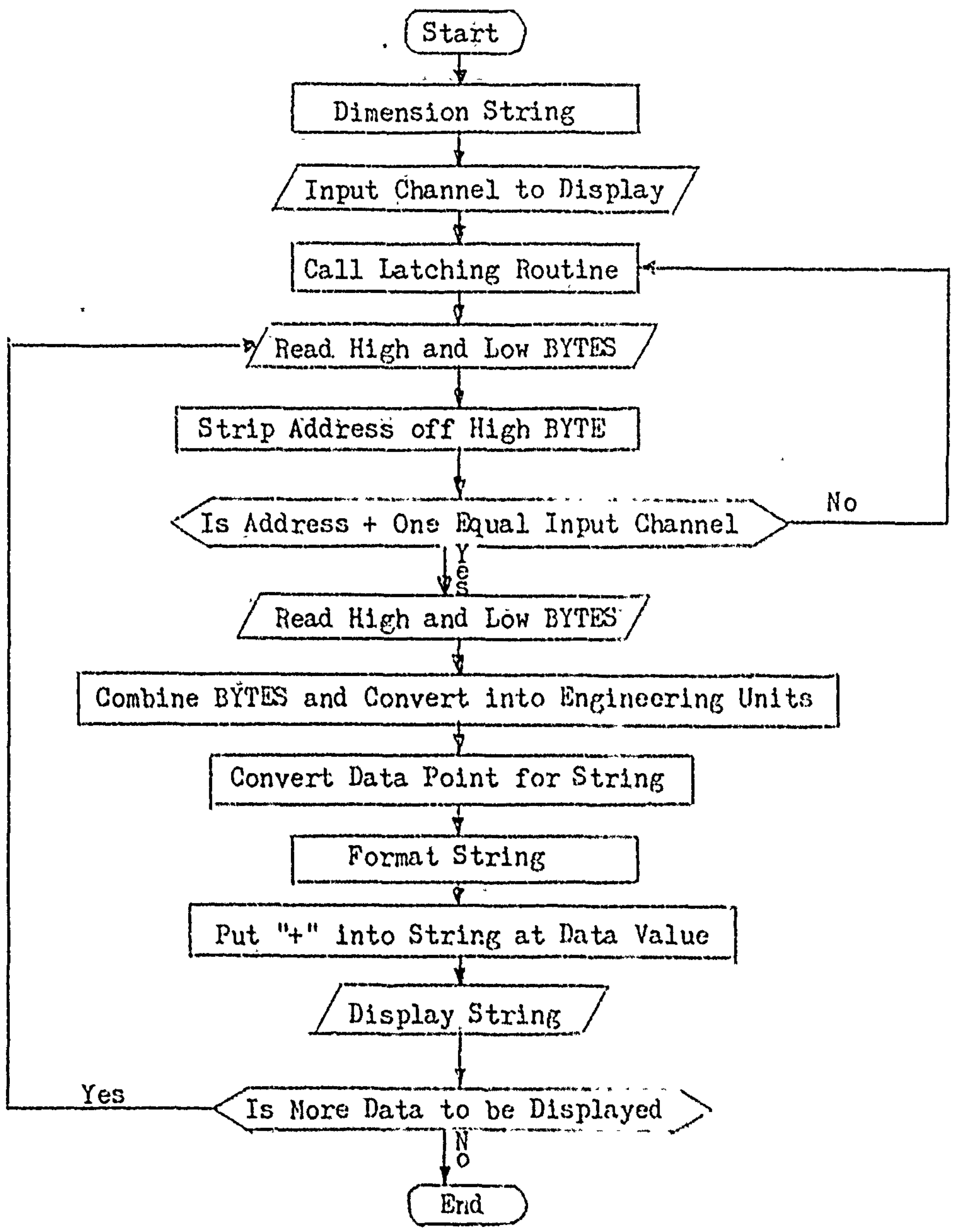
Program Listing II-4A (Cont)

```
COMMENT GET ALL INPUT DATA IN HIGH/LOW BYTE FORMAT,CONVERT IT TO E.U.
        DO 109 J=1,8
        DO 100 I=1,16
        LOC=MEM+(I-1)*2+(J-1)*32
        IHI=PEEK(LOC)
        IHI=(IHI .AND. 255)
        LO=PEEK(LOC+1)
        LO=(LO .AND. 255)
        IADD(I)=IHI/16
        IHI=(15 .AND. IHI)
        DATA=(IHI*256+LO)*.002442-5
        DAT(I,J)=DATA*CAL(I)
100     CONTINUE
COMMENT WRITE OUT DATA ADDRESSES FOR CHECKING VALIDITY
        WRITE(5,90)(IADD(I),I=1,16),J
109     CONTINUE
        PAUSE
        WRITE(5,20)
COMMENT WRITE WAVE DATA NOW
        WRITE(5,25)
        DO 140 I=1,15,2
        WRITE(5,30)(DAT(I,J),J=1,8)
140     CONTINUE
        PAUSE
        WRITE(5,20)
COMMENT WRITE TIDE DATA NEXT
        WRITE(5,35)
        WRITE(5,30)(DAT(6,J),J=1,8)
COMMENT NOW WRITE CURRENT DATA (ON-SHORE, THEN LONG-SHORE)
        WRITE(5,45)
        WRITE(5,30)(DAT(4,J),J=1,8)
        WRITE(5,30)(DAT(12,J),J=1,8)
        WRITE(5,55)
        WRITE(5,30)(DAT(2,J),J=1,8)
        WRITE(5,30)(DAT(10,J),J=1,8)
COMMENT WRITE WATER TEMPERATURE HERE
        WRITE(5,65)
        WRITE(5,30)(DAT(8,J),J=1,8)
COMMENT NOW WRITE INTERNAL BUOY TEMPERATURE
        WRITE(5,75)
        WRITE(5,30)(DAT(16,J),J=1,8)
COMMENT WRITE CALIBRATION VOLTAGE LAST
        WRITE(5,85)
        WRITE(5,40)(DAT(14,J),J=1,8)
        PAUSE
COMMENT TO END PROGRAM PUSH LETTER T, CARRIAGE RETURN.
COMMENT TO CONTINUE PROGRAM FOR ANOTHER PRINT OUT OF DATA,
COMMENT PUSH CARRIAGE RETURN ONLY--PROGRAM AUTO RECYCLES.
        GO TO 334
        END
```

Program Listing II-4B. Routine for Printing or Displaying Data in Engineering Units -- HPL Version

```
0: "PROGRAM READS DATA FROM BUOY,CHECKS ADDRESS,PUTS DATA INTO T$-ARRAY":
1: "RESULTS ARE PRINTED OUT IN ENGINEERING UNITS BY SENSOR FUNCTION":
2: dim X[8,16];fmt 0;fmt 1,z,f10.3;dim T$[14]
3: "START INPUTTING DATA;LOOK FOR LOW BYTE(PARITY ERROR)":
4: "NEXT LOOK FOR HIGH BYTE WITH ZERO ADDRESS (FIRST CHENNEL)":
5: "START":wtc 11,1;wtb 11,64;wtb 11,255;wtb 11,36;wtc 11,0
6: rdb(11)→P;wtc 11,1;rdb(11)→P;wtc 11,0;band(P,8)→P;if P#8;jmp 0
7: rdb(11)→A;rdb(11)→B;shf(A,4)→A;shf(B,4)→B;dsp A,B;if A#15;jmp 0
8: "CONVERT HIGH/BYTE-LOW/BYTE PAIR TO DATA WORD AND PUT IN X-ARRAY":
9: for J=1 to 8;for I=1 to 16;rdb(11)→A→C;band(A,15)→A;rdb(11)→B
10: (256A+B)*.002442-5→X[J,I];shf(C,4)→C;if C#I-1;beep
11: dsp "J=",J,"I=",I;next I;next J
12: "REST OF PROGRAM WRITES OUT TITLES AND DATA FROM SENSORS":
13: fmt 2,20x,"WATER LEVEL (WAVES) IN FEET (READ DOWN)";wrt 15;wrt 15.2
14: fmt 3,z,f10.2;for I=1 to 16 by 2;for J=1 to 8;wrt 15.3,X[J,I]*4
15: next J;wrt 15;next I
16: fmt 4,30x,"TIDE LEVEL IN FEET";wrt 15;wrt 15.4
17: for I=1 to 8;wrt 15.3,X[I,6]*4.87;next I
18: wrt 15;fmt 5,20x,"ON-SHORE CURRENT (Y) IN FT/SEC";wrt 15;wrt 15.5
19: for I=1 to 8;wrt 15.3,10*X[I,4],10*X[I,12];next I
20: wrt 15;fmt 6,20x,"LONG-SHORE CURRENT (X) IN FT/SEC";wrt 15;wrt 15.6
21: for I=1 to 8;wrt 15.3,10*X[I,2],10*X[I,10];next I
22: wrt 15;fmt 7,30x,"WATER TEMPERATURE IN DEG C";wrt 15;wrt 15.7
23: for I=1 to 8;wrt 15.3,10*X[I,8];next I
24: wrt 15;fmt 8,30x,"INTERNAL BUOY TEMPERATURE IN DEG C";wrt 15;wrt 15.8
25: for I=1 to 8;wrt 15.3,10*X[I,16];next I
26: wrt 15;fmt 9,20x,"CALIBRATION SIGNAL IN VOLTS (REF=0.0)";wrt 15;wrt 15.
27: for I=1 to 8;wrt 15.1,X[I,14];next I;wrt 15
28: wrt 9,"R";red 9,T$;wrt 15,T$;wrt 15;wrt 15
29: beep;dsp "PUSH CONTINUE FOR MORE DATA.....";stp
30: dsp "                                        ";gto "START"
31: end
*8992
```

Figure II-5. Flow Chart of Routines for Dynamically Displaying Wave Data



Program Listing II-5A. Routine for Dynamically Displaying Wave Data
FORTRAN Version

```
TYPE DISPLAY7.FOR
        PROGRAM DISPLAY7
COMMENT ROUTINE TO TAKE DATA FROM 7 WAVE GAUGES OUT OF 11 CHANNELS
COM  AND CONVERT THE RESULTS TO E.U. IN WAVE HEIGHT VALUES
COM  THESE WAVE HEIGHTS FROM EACH GAUGE ARE THEN DYNAMICALLY DISPLAYED
COM  IN "REAL TIME" ACROSS 7 ROWS OF THE CRT
COM  PROGRAM CONTINUES OPEN LOOP UNTIL DATA ENDS OR OPERATOR STOPS IT
COM  DATA STREAM IS LATCHED USING SAME ROUTINE AS PROGRAM "LATCH"
COM  INITIALIZE VARIABLES NEXT
DIMENSION IDATA(7)
        LOGICAL IDATA,IHI,LO,NUM,K
        IREF=Z'F840'
        DO 105 I=1,7
105     IDATA(I)=0
COM  CLEAR CRT SCREEN (ADDRESS F800HEX TO FBFFHEX) BY WRITING 127 (DEL)
        DO 106 I=1,1024
        MM=Z'F7FF'+I
106     CALL POKE(MM,127)
COM  BEGIN DATA STREAM LATCHING ROUTINE (SEE PREV PROGRAM COMMENTS)
        IB=Z'E4'
        IS=Z'E1'
        ID=Z'E0'
        MEM=Z'5000'
100     CALL OUT(IB,Z'39')
        CALL OUT(IS,Z'AA')
        CALL OUT(IS,Z'50')
        CALL OUT(IS,Z'7E')
        CALL OUT(IS,0)
        CALL OUT(IS,5)
110     ISTAT=INP(IS)
        IPAR=ISTAT
        ISTAT=(ISTAT .AND. 2)
        IF(ISTAT .EQ. 0)GO TO 110
        IPAR=(IPAR .AND. 8)
        IF(IPAR .NE. 0)GO TO 100
        IHI=INP(ID)
        IHI=(IHI .AND. 240)
        IF(IHI .NE. 160)GO TO 100
        CALL STATUS
        LO=INP(ID)
COM  DATA NOW LATCHED AT CHANNEL 1 (ADDRESS 0).
COM  READY TO BEGIN DISPLAY ROUTINE.  THE VALUE OF EACH WAVE DATA POINT
COM  IS COMPUTED (IN E. U.) WHICH IS ADDED TO THE APPROPRIATE MEMORY
COM  ADDRESS AND IS POKED THERE APPEARING ON THE CRT AT
COM  THAT LOCATION AS AN "*". THE PREVIOUS DISPLAYED "*"IS FIRST ERASED
COM  AFTER THE FIRST 7 CHANNELS ARE READ AND PLOTTED, NEXT FOUR
COM  ARE SKIPPED. PROGRAM THEN JUMPS BACK TO BEGINNING OF LATCHING ROUT

210     DO 200 K=1,7
```

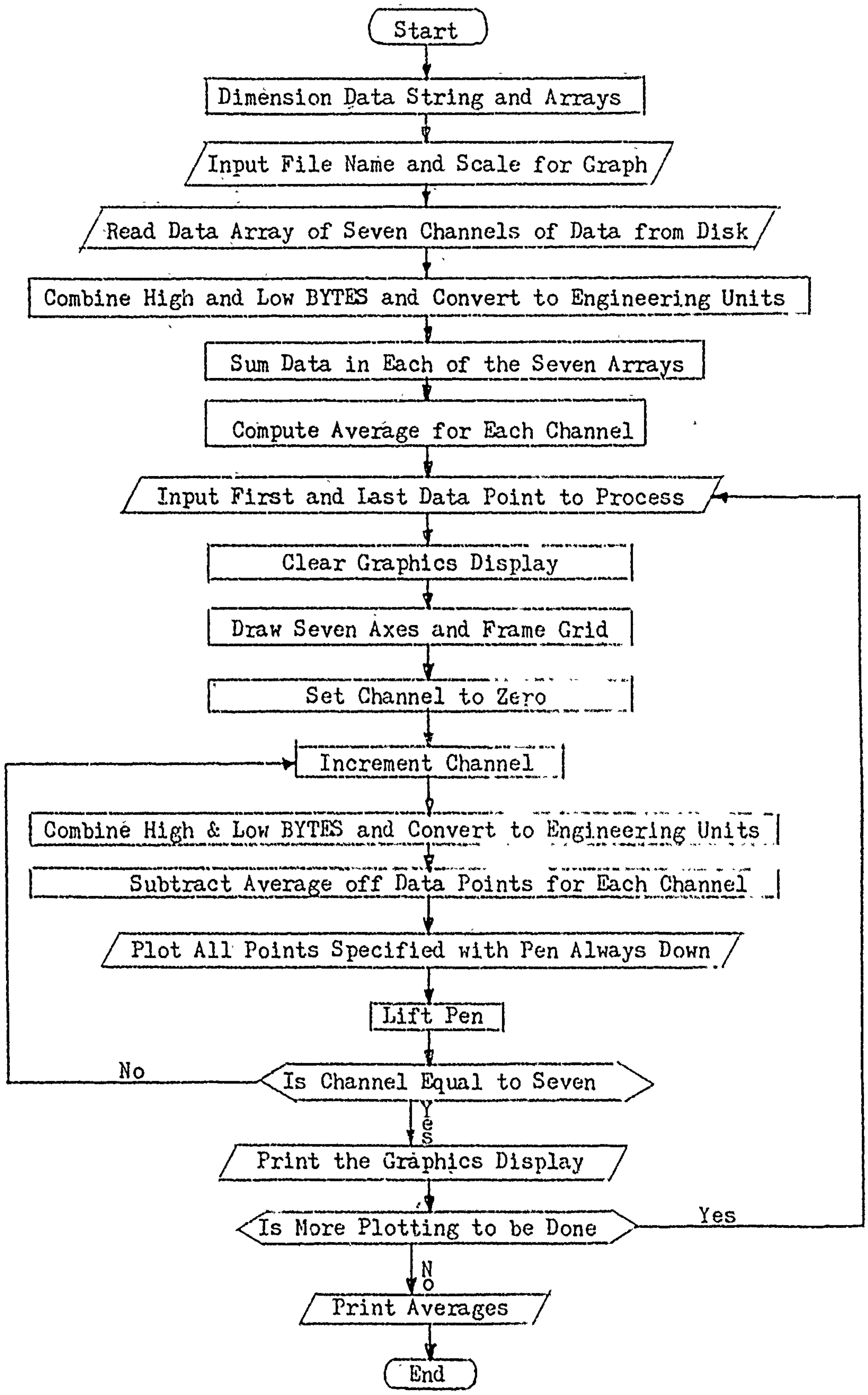
Program Listing II-5A (Cont)

```
        CALL STATUS
        IHI=INP(ID)
        IHI=(IHI .AND. 15)*4
        CALL STATUS
        LO=INP(ID)
        LO=(LO .AND. 192)/64
        NUM=IHI+LO
COM  IROW IS ROW FOR PARTICULAR WAVE GAUGE TO BE DISPLAYED
COM  IN MEMORY ADDRESS: F840, F8C0, F940, F9C0, FA40, FAC0, FB40, FBC
COM  SPECIFIC LOCATION ON THE 7 ROWS DETERMINED NEXT
        IROW=IREF+(K-1)*128
        N=IROW+IDATA(K)
COM  ERASE OLD VALUE "*" BEFORE STORING AND DISPLAYING NEW ONE
        CALL POKE(N,127)
        IDATA(K)=NUM
        N=IROW+IDATA(K)
COM  NOW POKE NEW LOCATION WITH AN "*" ON CRT FOR THAT GAUGE
        CALL POKE(N,Z'AA')
200     CONTINUE
COM  ALL DONE FOR ONE SET OF 7 GAUGES. GO BACK AND GET NEXT SET.
        GO TO 100
COM  THIS NEXT PART NOT USED HERE. COMMENT (C) COULD BE REMOVED
COM  ALONG WITH THE ABOVE GO TO 100 AND THE ROUTINE WOULD READ
COM  THE NEXT FOUR CHANNELS AND THEN START OVER. THE ABOVE ROUTINE
COM  EACH TIME LATCHES ON TO CHANNEL ONE --- A SAFER WAY
COM  SINCE LATCH COULD BE LOST OTHERWISE.
C       DO 250 K=1,4
C       CALL STATUS
C       IHI=INP(ID)
C       CALL STATUS
C250    LO=INP(ID)
C       GO TO 210
        END
```

Program Listing II-5B. Routine for Dynamically Displaying Wave Data--HPL Version

```
0: "PROGRAM TO DISPLAY WAVE LEVEL WITH A '+' CONTINUOUSLY":
1: "PROGRAM SET UP TO DISPLAY CHANNEL 1 ONLY, BUT READS ALL CHANNELS":
2: ent "ENTER CHANNEL NO. TO DISPLAY",N;N-1→N→S
3: ent "ENTER TOTAL NUMBER OF CHANNELS",M
4: if N<0;M-1→N
5: dim D$[32];fxd 3
6: "START":wtc 11,1;wtb 11,64;wtb 11,255;wtb 11,36;wtc 11,0
7: rdb(11)→P;wtc 11,1;rdb(11)→P;wtc 11,0;band(P,8)→P;if P#8;jmp 0
8: rdb(11)→A;rdb(11)→B;shf(A,4)→A;shf(B,4)→B;dsp A,B;if A#N;jmp 0
9: rdb(11)→A→C;band(A,15)→A;rdb(11)→B
10: (256A+B)*.001221→X;shf(C,4)→C;if C#S+1;beep
11: dsp X;for I=2 to M;rdb(11)→A;rdb(11)→A;next I;jmp -2
12: int(X*5)+0→X;""→D$[1,32];"I"→D$[1,1]→D$[11,11]→D$[21,21]→D$[31,31]
13: if X<33 and X>0;"+"→D$[X,X]
14: dsp D$;for I=2 to M;rdb(11)→A;rdb(11)→A;next I
15: jmp -5
16: end
*2152
```

Figure II-6. Flow Chart for Digital Strip-chart Displaying Program



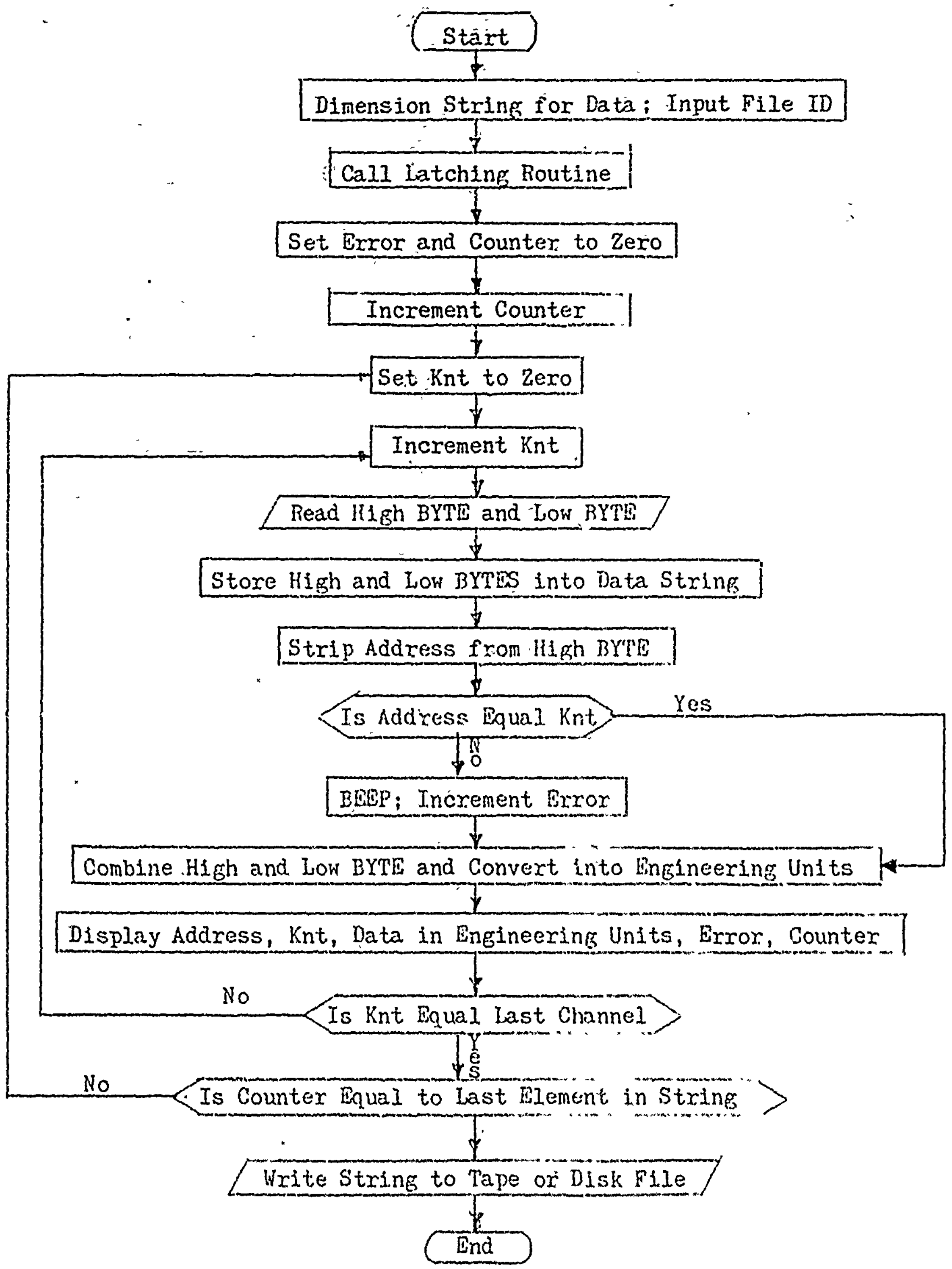
Program Listing II-6. Digital Strip-chart Displaying Program -- BASIC Version

```
10 ! PROGRAM NAME  IS PL1OHR.  READS DATA FROM DISK INITIALIZED BY HP-9825
20 ! PLOTS STRIP CHARTS OF 7 CHANNELS OF DUCK DATA
30    OPTION BASE 1
40    DIM D$(2240)[14],Avg(7),S(7),C(7)
50 !  ENTER PARAMETERS
60    INPUT "DATA FILE NAME ?",F$
70    MASS STORAGE IS ":F8"
80    INPUT "FIRST AND LAST ELEMENT OF PLOT",I1,I2
90    INPUT "SCALING FACTOR FOR RANGE",Scale
100   ASSIGN #1 TO F$
110   READ #1;D$(*)
120   C(1)=C(2)=C(3)=6.2
130   C(4)=C(5)=C(6)=C(7)=5.6
140   MAT S=ZER
150 ! LOOP FOR COMBINING BYTES, CONVERTING TO DIGITAL NUMBERS, & SUMMING
160   FOR J=1 TO 7
170   FOR I=1 TO 2240
180   Hi=NUM(D$(I)[2*J-1,2*J-1])
190   Low=NUM(D$(I)[2*J,2*J])
200   D=256*BINAND(Hi,15)+Low
210   S(J)=S(J)+D
220   NEXT I
230   NEXT J
240 ! COMPUTE AVERAGE FOR EACH OF SEVEN CHANNELS
250   FOR I=1 TO 7
260   Avg(I)=S(I)/2240
270    NEXT I
280   I1=I1-1
290 ! START OF PLOTTING ROUTINE
300   GRAPHICS
310   PLOTTER IS 13,"GRAPHICS"
320   LOCATE 0,123,0,100
330   FRAME
```

Program Listing II-6 (Cont)

```
340   SCALE I1,I2,0,14*Scale
350   FOR J=1 TO 7
360   PENUP
370   PLOT I1,J*(2*Scale)-Scale
380   PLOT I2,J*(2*Scale)-Scale
390   NEXT J
400   AXES (I2-I1)/10,Scale,0,0
410   FRAME
420 ! LOOP FOR PLOTTING OF SEVEN CHANNELS
430   FOR J=1 TO 7
440   PENUP
450 ! INNER LOOP TO COMBINE BYTES, SUBTRACT OFF AVERAGE, CONVERT TO
460 ! ENGINEERING UNITS AND PLOT EACH POINT
470   FOR I=I1+1 TO I2
480   Hi=NUM(D$(I)[2*J-1,2*J-1])
490   Low=NUM(D$(I)[2*J,2*J])
500   D=256*BINAND(Hi,15)+Low-Avg(J)
510   PLOT I,D*.001221*C(J)+2*Scale*J-Scale
520   NEXT I
530   NEXT J
540   DUMP GRAPHICS
550   PRINT PAGE
560   INPUT "DO YOU WANT TO PLOT MORE DATA?";A$
570   IF A$="NO" THEN 600
580   INPUT "FIRST AND LAST ELEMENT OF PLOT?",I1,I2
590   GOTO 280
600   GCLEAR
610   PRINTER IS 0
620   FIXED 3
630   PRINT Avg(1);Avg(2);Avg(3);Avg(4);Avg(5);Avg(6);Avg(7)
640   PRINTER IS 16
650   DISP "END OF PL1ONR"
660   END
```

Figure II-7. Flow Chart of Routine for Collecting and Storing Data from Buoy



Program Listing II-7A. Routine for Collecting and Storing Data -- FORTRAN Version

```
A>TYPE STORED.FOR
        PROGRAM STORED
COMMENT  PROGRAM TO READ DATA FROM SERIAL I/O PORT AND PUT IN ARRAY
COM   TO BE STORED ON DISK (LUN(9))
        DIMENSION IDATA(4096)
COM   LATCHING ROUTINE FOLLOWS NEXT
        IB=Z'E4'
        IS=Z'E1'
        ID=Z'E0'
100     CALL OUT(IB,Z'39' )
        CALL OUT(IS,Z'AA' )
        CALL OUT(IS,Z'50' )
        CALL OUT(IS,Z'7E' )
        CALL OUT(IS,0 )
        CALL OUT(IS,5 )
110     ISTAT=INP(IS )
        IPAR=ISTAT
        ISTAT=(ISTAT .AND. 2 )
        IF(ISTAT .EQ. 0 )GO TO 110
        IPAR=(IPAR .AND. 8 )
        IF(IPAR .NE. 0 )GO TO 100
        IHI=INP(ID )
        IHI=(IHI .AND. 240 )
        IF(IHI .NE. 160 )GO TO 100
        CALL STATUS
        LO=INP(ID )
COM  DATA STREAM IS LATCHED; START READING DATA AT ADDRESS 00
COM  READ 4096 DATA WORDS AND STORE ON DISK (LUN(9))
        DO 200 K=1,4096
        CALL STATUS
        IHI=INP(ID )
COM  STRIP OFF ADDRESS AND THROW AWAY
        IHI=(IHI .AND. 15 )
        CALL STATUS
        LO=INP(ID )
        LO=(LO .AND. 255 )
COM  PUT DATA WORD INTO ARRAY
        IDATA(K )=IHI*256+LO
200     CONTINUE
COM  STORE IT ON DISK NOW; THEN YOU'RE DONE!
        WRITE(9 )IDATA
        END
        SUBROUTINE STATUS
COM  SUBROUTINE TO CHECK STATUS OF SERIAL PORT
COM  ROUTINE KEEPS CHECKING UNTIL BYTE IS THERE; THEN RETURNS
        IS=Z'E1'
10      ISTAT=INP(IS )
        ISTAT=(ISTAT .AND. 2 )
        IF(ISTAT .EQ. 0 )GO TO 10
        RETURN
        END
```

Program Listing II-7B. Routine for Collecting and Storing Data from Duck
BASIC Version

```
10    ! PROGRAM NAME IS DUCDAT
20    ! PROGRAM INPUTS DATA FROM ITS DIGITAL TAPES, LOCATES HIGH BYTE (VIA
30    ! PARITY BIT), LOCATES FIRST CHANNEL (0), THEN READS THE 7 WAVE GAUGES
40    ! INTO A STRING ARRAY D$(7200)[14] HIGH BYTE-LOW BYTE 0 THROUGH 6
50    ! THEN STORES THE ARRAY ON DISK. ERROR ROUTINE KEEPS DATA LOCK.
60    DIM D$(7200)[14]
70    INPUT "DISK FILE NAME?",F$
80    GOSUB Reset
90    GOTO Loop
100 Reset: WAIT WRITE 11,5;1
110   WRITE BIN 11;64
120   WRITE BIN 11;255
130   WRITE BIN 11;20
140   WAIT WRITE 11,5;0
150   P=READBIN(11)
160   WAIT WRITE 11,5;1
170   P=READBIN(11)
180   WAIT WRITE 11,5;0
190   P=BINAND(P,8)
200   IF P<>8 THEN GOTO Reset
210 Add:  A=READBIN(11)
220   B=READBIN(11)
230   A=SHIFT(A,4)
240   DISP A,B
250   IF A<>10 THEN GOTO Add
260   Error=0
270   RETURN
280 Loop:  FOR J=1 TO 7200
290   FOR I=1 TO 11
300   A=READBIN(11)
310   B=READBIN(11)
320   IF I>7 THEN 400
330   D$(J)[2*I-1,2*I-1]=CHR$(A)
340   D$(J)[2*I,2*I]=CHR$(B)
350   A=SHIFT(A,4)
360   IF A+1=1 THEN 400
370   Error=Error+1
380   BEEP
390   GOTO 430
400   NEXT I
420   IF Error=0 THEN 445
430   GOSUB Reset
440   J=J-1
445   D$(7200)[1,5]=VAL$(J)
450   DISP J
460   NEXT J
470   MASS STORAGE IS ":F8"
480   CREATE F$,404
490   ASSIGN #2 TO F$
500   PRINT #2;D$(*)
510   DISP "END OF DATA STORAGE"
520   END
```

Program Listing II-7C. Routine for Collecting and Storing Data from Buoy
Assembly Language Version

```
ADDR  CODE       LABEL    OP CODE        COMMENTS

EC00  CD EF E3   START:   CALL SET UP    ;call monitor routine for init SIO
EC03  36                  DB   36        ;SIO set at 300 baud
EC04  AA                  DB   AA        ;flush SIO
EC05  50                  DB   50        ;reset USART and all flags
EC06  7E                  DB   7E        ;format USART to 8 data bits, even
                                         ;parity, and 1 stopbit
EC07  00                  DB   00        ;end of list
EC08  3E 05               MVI  A,05      ;code for transmit/receive for SIO
EC0A  D3 E1               OUT  SIO       ;turn on SIO port  (USART)
EC0C  DB E1      STATUS:  IN   E1        ;read SIO status
EC0E  4F                  MOV  C,A       ;save it in register C
EC0F  E6 02               ANI  02        ;is data byte there?
EC11  CA 0C EC            JZ   STATUS    ;if not go back & read it again
EC14  79                  MOV  A,C       ;put status word into reg A
EC15  E6 08               ANI  08        ;does it have even parity?
EC17  C2 00 EC            JNZ  START     ;if not reset USART flags and look
                                         ;at next byte
EC1A  DB E0               IN   SIO       ;it's a high byte -- read it
EC1C  4F                  MOV  C,A       ;put it in register C also
EC1D  E6 F0               ANI  0F0H      ;look at the address bits (4 msb)
EC1F  FE 00               CPI  00        ;is it address 00? (first channel)
EC21  C2 00 EC            JNZ  START     ;no get another high byte as above
EC24  79                  MOV  A,C       ;yes it is we're latched and ready
EC25  01 00 80            LXI  B,8000H   ;do 8x4096 data bytes (32768 total)
EC28  21 00 40            LXI  H,4000H   ;starting memory address for the data
EC2B  77                  MOV  M,A       ;put byte in memory
EC2C  DB E1      STATO:   IN   E1        ;read SIO status
EC2E  E6 02               ANI  02        ;is next byte there?
EC30  CA 2C EC            JZ   STATO     ;if not look again
EC33  DB E0               IN   SIO       ;get next byte
EC35  23                  INX  H         ;increment memory address
EC36  77                  MOV  M,A       ;put byte in memory
EC37  0C                  INR  C         ;increment counter
EC38  C2 2C EC            JNZ  STATO     ;all done?
EC3B  04                  INR  B         ;increment outer counter
EC3C  C2 2C EC            JNZ  STATO     ;all done?
EC3F  CD XX XX            CALL TAPE      ;yes, store it on tape
EC42  C3 00 E0            JMP  MONITOR   ;we're through! you can go to monitor
;
;comment -- program that sets up serial I/O port to 300 baud, checks data
;comment -- input for even parity (high byte), then locates 00 address
;comment -- next reads in data into memory starting at 4000hex and does
;comment -- 32768 bytes of data (address is still contained in high byte)
;comment -- after all data are in memory, data are stored on tape (or disk)
;comment -- control then returns to the monitor routine (or anywhere else)
```

Program Listing II-7D. Routine for Collecting and Storing Data from Duck
HPL Version

```
0: "program to store on disk binary data that has been read into a string
1: fmt 1,f2.0,x,f2.0,2x,f6.0,f8.0
2: fmt 2,f2.0,"/",f2.0,"/",f2.0,10x,f2.0,":",f2.0
3: dim D$[2240,14],F$[10]
4: ent "What is file name",F$
5: for I=1 to 2240;"*"→D$[I,1,14];next I
6: dsp "READY FOR DATA -- PUSH CONT";stp
7: dsp "DONE - HERE COMES THE DATA...."
8: gsb "reset"
9: for J=1 to 2240;for I=1 to 11
10: rdb(11)→A;rdb(11)→B;if I>7;jmp 2
11: char(A)→D$[J,2*I-1,2*I-1];char(B)→D$[J,2*I,2*I]
12: shf(A,4)→C;band(A,15)→A
13: if C+1#I;beep;E+1→E;jmp 2
14: wrt .1,C,I-1,J;next I;jmp 2
15: J-1→J;gsb "reset"
16: next J;beep;wait 500;beep
17: open F$,160;asgn F$,2,0,X;sprt 2,D$,"end";dsp "DATA DONE";end
18: open F$,160;asgn F$,2,0,X;sprt 2,D$,"end";dsp "DATA DONE-",F$;end
19: "reset":wtc 11,1;wtb 11,64;wtb 11,255;wtb 11,20;wtc 11,0
20: rdb(11)→P;wtc 11,1;rdb(11)→P;wtc 11,0;band(P,8)→P;if P#8;jmp 0
21: rdb(11)→A;rdb(11)→B;shf(A,4)→A;0→E;if A#10;jmp 0
22: ret
*2597
```

Program Listing II-7. Program to Input, Format, and Store Buoy Data

```
0: "PROGTAM TO STORE ON TAPE BINARY DATA THAT HAS BEEN READ INTO A STRING
1: fmt 1,f2.0,x,f2.0,2x,f4.0,f8.0,f8.0,f5.0;ent "what is run #?",T
2: dim D$[387,32];dim T$[14]
3: "INITIALIZATION OF VARIABLES AND ARRAYS":
4: ent "How many files?",R;ent "What is first file #",F
5: 0→E;for I=1 to 387;"*"→D$[I,1,32];next I
6: dsp "Ready for data -- Push CONT";stp
7: dsp "Done   here comes the data     "
8: for K=1 to R
9: "FINDING LOW BYTE AND FIRST CHANNEL":
10: wtc 11,1;wtb 11,64;wtb 11,255;wtb 11,20;wtc 11,0
11: rdb(11)→P;wtc 11,1;rdb(11)→P;wtc 11,0;band(P,8)→P;if P#8;jmp 0
12: rdb(11)→A;rdb(11)→B;shf(A,4)→A;dsp A,B;if A#15;jmp 0
13: "LOOP FOR PUTTING DATA INTO FILES, STRINGS SIZED 387 BY 32":
14: for L=1 to 3;for J=1 to 128;for I=1 to 16
15: rdb(11)→A;char(A)→D$[J+(L-1)*129,2*I-1,2*I-1]
16: rdb(11)→B;char(B)→D$[J+(L-1)*129,2*I,2*I];shf(A,4)→C;band(A,15)→A
17: "CHECKING ADDRESS IN HIGH BYTE AGAINST PROGRAM COUNTER":
18: if C+1#I;beep;E+1→E
19: wrt .1,C,I-1,(256*A+B)*2.442-5000,E,J,L;next I;next J
20: "PUTTING TIME,RUN #,PART #,FILE # INTO EACH 129TH ROW OF STRING":
21: wrt 9,"R";red 9,T$;T$→D$[129L,1,14]
22: char(T)→D$[129*L,21,21];char(K)→D$[129*L,22,22]
23: char(F)→D$[129*L,23,23]
24: next L;dsp T$;rcf F,D$;F+1→F;next K
25: dsp "End of program taking data";end
*631
```

Program Listing II-8. Program to Crunch Stored Buoy Data to Samller Sets.

```
0: "PROGRAM TO CRUNCH BINARY STRING ARRAY TO STRING ARRAY 1/3RD THE SIZE"
1: fmt 1,f4.0,f4.0,f4.0;dim D$[387,32];dim E$[129,32]
2: ent "HOW MANY FILES OF ORIGINAL DATA?",N
3: ent "NUMBER OF FIRST FILE TO BE READ?",R
4: ent "NUMBER OF FIRST FILE TO BE WRITTEN UPON?",F
5: "LOOP TO READ DATA FROM TRQCK 0 AND STORE SMALLER FILES ON TRACK 1":
6: for K=R to R+N-1;trk 0;ldf K,D$;for L=1 to 3;for J=1 to 129
7: D$[(L-1)*129+J,1,32]→E$[J,1,32];wrt .1,K,L,J,E$[J,1,14];next J
8: trk 1;rcf F,E$;F+1→F;next L;next K;end
*24126
```

Figure II-8. Flow Chart for Routine to Crunch Data Files

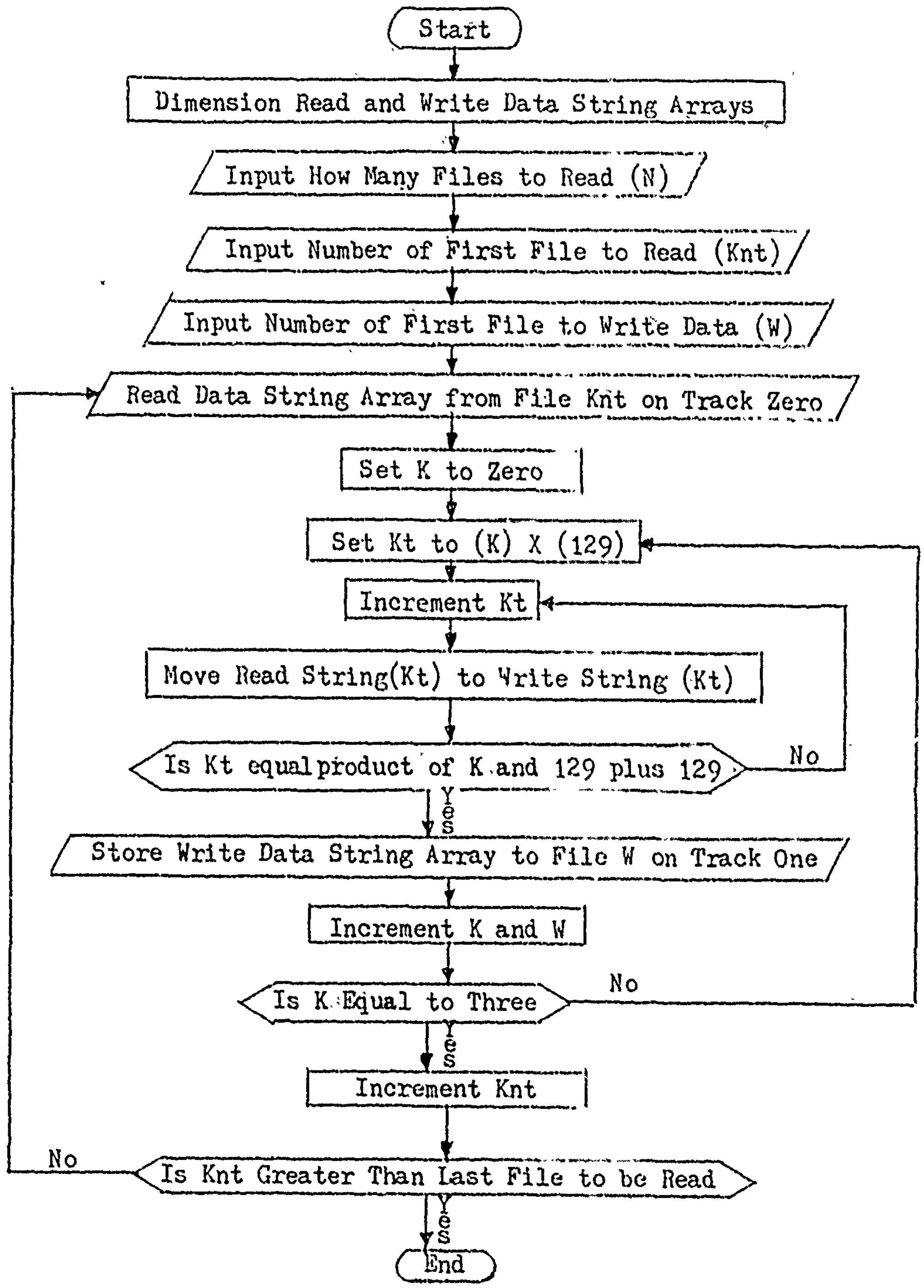
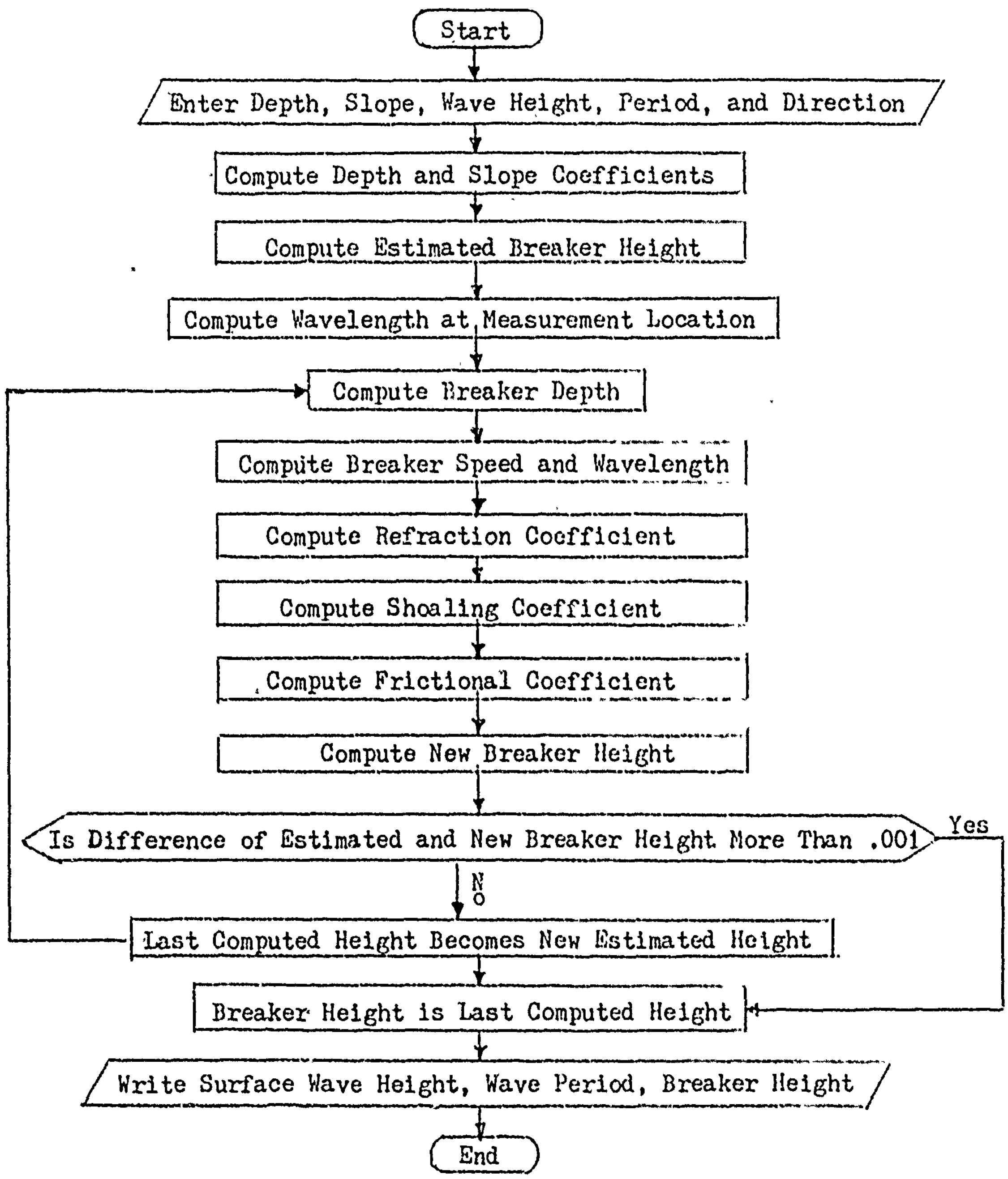


Figure II-9. Flow Chart of Program for Modified Processing for the Detailed Model



Program Listing II-9. Program for Modified Processing for Detailed Model -- FORTRAN Version

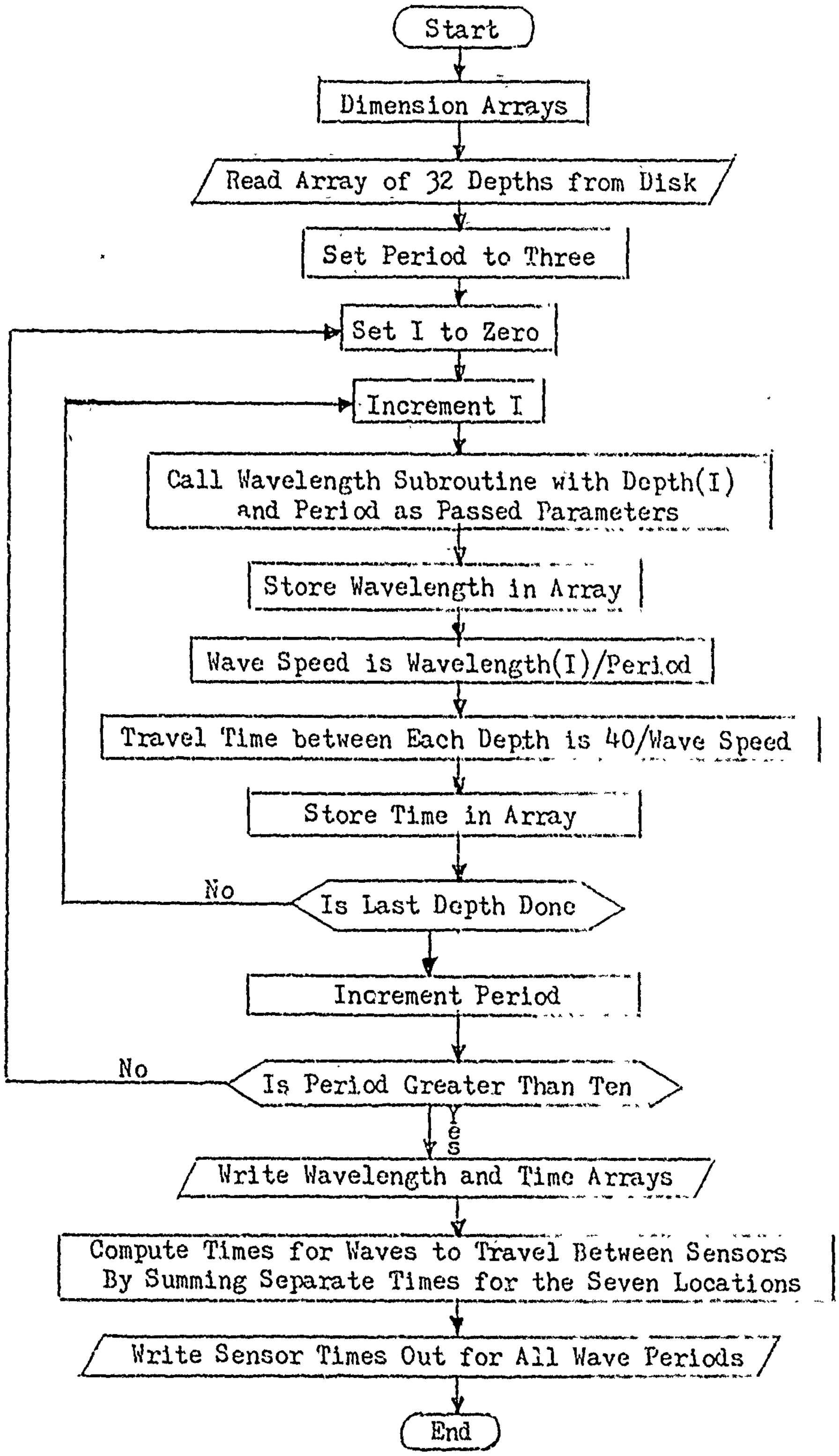
```
TYPE KEYDAT.FOR
          PROGRAM KEYDAT
COMMENT PROGRAM TO ENTER WAVE HEIGHT, PERIOD, DIRECTION, DEPTH, SLOPE
COMMENT FROM THE KEYBOARD AND COMPUTE BREAKER HEIGHT AND DEPTH
COMMENT USING AN ITERATION PROCESS
7         FORMAT(1X,'ENTER WAVE DIRN (E.G. 89.0)')
8         FORMAT(1X,'ENTER DEPTH (E. G. 25.6)')
9         FORMAT(1X,'ENTER SLOPE (E. G. .05)')
10        FORMAT(1X,'WAVE HEIGHT',4X,'WAVE PERIOD',4X,'BREAKER HT',
       1  4X,'BREAKER DEPTH')
11        FORMAT (1X,'ENTER WAVE HEIGHT')
12        FORMAT (1X,'ENTER WAVE PERIOD')
13        FORMAT(1X,F10.2,5X,F10.2,5X,F10.2,5X,F10.2)
99        FORMAT(1X,4F15.5)
COMMENT ENTER PARAMETERS FROM KEYBOARD NEXT
80        WRITE (5,11)
          CALL ENTER (WAVEHT)
          WRITE (5,12)
          CALL ENTER (WAVPER)
          P=WAVPER
          WRITE(5,8)
          CALL ENTER(DEPTH)
          WRITE(5,9)
          CALL ENTER(SLOPE)
          WRITE(5,7)
          CALL ENTER(THETA)
          WRITE(5,99)WAVEHT,P,DEPTH,SLOPE
COMMENT INITIALIZE VARIABLES NEXT
          PI=3.14159
          THETA=ABS(THETA-90.0)
          THETA=THETA*PI/180.0
          H=WAVEHT
COMMENT START ITERATION ROUTINE FOR DETAILED PREDICTION MODEL
          A=1.36*(1-EXP(-19*SLOPE))
          B=1.56/(1+EXP(-19.5*SLOPE))
          H1=(.98**2*((.90*DEPTH)**.25)*H)**.8
18        D=H1/(B-A*H1/(P*P))
          CALL WAVLTH(P,WL,DEPTH)
          C=WL/P
          CB=SQRT(32.2*(H1+D))
          WLB=CB*P
          SINTHB=(CB/C)*SIN(THETA)
          COSTHB=SQRT(1-SINTHB**2)
          CF=.98
          CTHETA=SQRT(ABS((COS(THETA))/COSTHB))
          ARG1=2*PI*DEPTH/WL
          ARG2=2*PI*D/WLB
          TANH1=(1-EXP(-2*ARG1))/(1+EXP(-2*ARG1))
          TANH2=(1-EXP(-2*ARG2))/(1+EXP(-2*ARG2))
          SINH1=(EXP(2*ARG1)-EXP(-2*ARG1))/2
          SINH2=(EXP(2*ARG2)-EXP(-2*ARG2))/2
```

Program Listing II-9 (Cont)

```
        CS=SQRT(TANH1/TANH2*((1+2*ARG1/SINH1)/(1+2*ARG2/SINH2)))
        H2=H*CF*CTHETA*CS
        VAR=ABS(H2-H1)
COMMENT CHECK FOR ITERATION DIFFERENCES LESS THAN .01 ABSOLUTE
        IF (VAR .LT. 0.01)GO TO 19
        H1=H2
        GO TO 18
19      BREKER=H2
COMMENT ALL DONE! WRITE THE RESULTS NOW AND DO SOME MOREIF DESIRED.
        WRITE (5,10)
        WRITE (5,13)WAVEHT,WAVPER,BREKER,D
        PAUSE MORE?
        GO TO 80
        END
        SUBROUTINE WAVLTH(P,WL,DEPTH)
COMMENT ROUTINE TO COMPUTE WAVE LENGTHS USING LINEAR WAVE THEORY
COMMENT ROUTINE ITERATES LENGTHS UNTIL RESULTS ARE WITHIN 0.01%.
        X1=32.2*P*P/(6.28319)
        WL1=X1
15      X=6.28319*DEPTH/WL1
        X2=EXP(2*X)
        WL2=X1*(X2-1)/(X2+1)
        WL3=ABS(WL2-WL1)/WL1
        IF (WL3 .LT. .0001) GO TO 16
        WL1=(WL2+WL1)/2
        GO TO 15
16      WL=WL2
        RETURN
        END
```

```
A>KEYDAT


ENTER WAVE HEIGHT
ENTER WAVE PERIOD
ENTER DEPTH (E. G. 25.6)
ENTER SLOPE (E. G. .05)
ENTER WAVE DIRN (E.G. 89.0)
     5.250      6.470     19.500       .055     98.900
WAVE HEIGHT    WAVE PERIOD    BREAKER HT    BREAKER DEPTH
      5.25           6.47           7.07            6.98 PAUSE MORE?

ENTER WAVE HEIGHT
ENTER WAVE PERIOD
ENTER DEPTH (E. G. 25.6)
ENTER SLOPE (E. G. .05)
ENTER WAVE DIRN (E.G. 89.0)
     7.500      8.000     25.400       .062    100.350
WAVE HEIGHT    WAVE PERIOD    BREAKER HT    BREAKER DEPTH
      7.50           8.00          10.18            9.67 PAUSE MORE? T
```

Figure II-10. Flow Chart of Program for Wave Time-of-arrival at Successive Wave Gauges for Duck CERC Data



Program Listing II-10. Program for Wave Time-of-Arrival at Successive Wave Gauges for Duck CERC Data -- FORTRAN Version

```
TYPE TIME.FOR
       PROGRAM TIME
COMMENT PROGRAM INPUTS FROM DISK (LUN(6)) STORED DATA FOR DEPTHS
COM  THEN COMPUTES SHALLOW-WATER WAVE LENGTH FOR PERIODS3-10 SEC
COM THEN COMPUTES TIME OF TRAVEL BETWEEN DEPTH LOCATIONSUSING
COM VELOCITY (FROM WAVELENGTH AND PERIOD) AND DISTANCE
COM BETWEEN DEPTH LOCATIONS, FINALLY TIME OF TRAVEL BETWEEN
COM WAVE-GAUGE LOCATIONS IS COMPUTED AND PRINTED OUT.
       DIMENSION TIME(10,32),IDEP(32),WAVLEN(10,32),SUM(10,6)
       DIMENSION DEP(32)
200    FORMAT(1X,8F9.2)
205    FORMAT(1X)
       DO 300 I=1,6
       DO 300 J=1,10
300    SUM(J,I)=0.0
COM  INPUT DEPTHS FROM DISK NEXT
       READ(6)IDEP
       ENDFILE 6
       PAUSE READ
COM  COMPUTE WAVE LENGTHS NEXT USING LINEAR THEORY AND ITERATING
       DO 75 I=1,32
75     DEP(I)=(IDEP(I)/10.0)-25.4
       PAUSE DEPTH
       WRITE(5,200)DEP
       DO 80 I=1,32
       DO 80 J=3,10
       X1=(32.2*J*J)/6.2832
       WL1=X1
230    X=6.2832*DEP(I)/WL1
       X2=EXP(2*X)
       WL2=X1*(X2-1)/(X2+1)
       WL3=ABS(WL2-WL1)/WL1
       IF(WL3 .LT. .0001)GO TO 210
220    WL1=(WL2+WL1)/2
       GO TO 230
210    WAVLEN(J,I)=WL2
80     CONTINUE
       PAUSE WAVLEN
       WRITE(5,205)
       DO 85 I=1,32
85     WRITE(5,200)(WAVLEN(J,I),J=3,10)
       DO 70 I=1,32
       DO 70 J=3,10
```

Program Listing II-10 (Cont)

```
COM   COMPUTE WAVE VELOCITY AND TIME OF TRAVEL NEXT.
        VEL=WAVLEN(J,I)/J
70      TIME(J,I)=40.0/VEL
        PAUSE TIME
        WRITE(5,205)
        DO 90 I=1,32
90      WRITE(5,200)(TIME(J,I),J=3,10)
        DO 60 J=3,10
        DO 10 I=1,2
10      SUM(J,1)=TIME(J,I)+SUM(J,1)
        DO 20 I=3,4
20      SUM(J,2)=TIME(J,I)+SUM(J,2)
        DO 30 I=5,7
30      SUM(J,3)=TIME(J,I)+SUM(J,3)
        DO 40 I=8,11
40      SUM(J,4)=TIME(J,I)+SUM(J,4)
        DO 50 I=12,21
50      SUM(J,5)=TIME(J,I)+SUM(J,5)
        DO 60 I=22,32
        SUM(J,6)=TIME(J,I)+SUM(J,6)
60      CONTINUE
        PAUSE SUM
        WRITE(5,205)
        DO 100 I=1,6
100     WRITE(5,200)(SUM(J,I),J=3,10)
        END

A>




A>
```

6.45	5.97	5.78	5.67	5.61	5.58	5.55	5.53
5.55	4.82	4.52	4.38	4.29	4.24	4.20	4.18
7.94	6.51	5.93	5.65	5.49	5.39	5.32	5.28
10.55	8.58	7.78	7.39	7.17	7.04	6.95	6.88
26.25	21.06	18.90	17.86	17.28	16.92	16.68	16.51
28.67	22.17	19.23	17.80	17.02	16.54	16.23	16.01

SECTION III

DATA PROCESSING USING THE SIMPLIFIED MODEL

In this study two prediction models were provided by Louisiana State University. Both of the models use as input information parameters that are measured at the buoy and, from this information, predict characteristics regarding the surf region. The statistics of breaker heights, depths, and longshore currents are predicted based upon linear wave theory and monochromatic waves. Wave processes of refraction, frictional attenuation, shoaling and breaking are treated mathematically in developing these prediction models.

The simplified model does not account for waves approaching the beach at different angles nor does it treat currents associated with individual waves separately. One of the more restrictive limitations of the model is that wave periods are not part of the model. This fact is discussed in latter parts of this section and in the last section of this report.

For the prediction of the breaker heights, the simplified model takes the measured wave height as determined from the pressure transducer output and the measured tide level and computes a breaker height based upon:

$$H_b = \left[.89\left(\frac{\bar{P}}{\rho g} + h_i\right)\left(\frac{\Delta P}{\rho g}\right)^4\right]^{1/5}$$

where H_b is the breaker height, $\bar{P}$ is the mean water pressure, ρ is the density of sea water, g is the acceleration of gravity, h_i is the height of the pressure sensor above the bottom. This computation is made for each individually measured wave at the data buoy.

The prediction model also includes a prediction of the maximum longshore current from the measured average longshore current at the buoy and the predicted breaker depth obtained from the mean breaker height as given by:

$$V_m = 1.7\,\bar{V}_0\left(\frac{h_0}{h_b}\right)^4$$

where V_m is the maximum longshore current, V_0 is the mean longshore current measured at the buoy, h_0 is the mean water depth at the buoy, and h_b is the mean breaker water depth.

The basic procedure for calculating the required parameters and the prediction results is given as follows. After dimensioning arrays and initialization, the buoy data are read into the computer. At this point the data are still in high byte/low byte format, from which they are next converted back to data words and put into a data array. Once again, all data-word addresses are checked for validity. If errors are detected, the operator has the option of manually terminating the run. In addition to the data, file identifications including run number, time and date, are stripped off the data file.

Once the data is converted to data-word arrays, the data processing is started. First, the individual wave heights are computed using a zero-crossing process in which the maximum (or minimum) height is determined between zero crossings. The half wave period is also determined by the number of data points between crossing times the wave sampling frequency. Full wave heights, i. e., peak-to-trough heights, and full wave periods are constructed with the results stored in two arrays -- one for wave heights and one for wave periods. It should be noted at this point that wave period is not necessary for any of the calculations in this model. It was added to the processing so that comparision of the results could be made with extrapolated heights at the surface. In this manner breaker heights predicted could be compared with surface heights at the buoy in addition to bottom-measured heights.

The final part of the program takes the data from the above arrays and performs the breaker prediction routine as given by the above equation. The surface wave heights are calculated using a curve-fitting process based upon results from linear wave theory. The average and RMS wave height for each of the three sets of wave heights are also calculated. The maximum longshore current is then calculated from these results and the measured offshore current velocities. The data are printed out along with the appropriate headings and identification.

The program can either be run for one set of data only or can be run multi-file, i. e., any number of sets can be run by auto-recycling of the program (an input variable). A requirement for recycling is that the data be located on sequentially increasing files since the program increments the file number after each run. Also, for the HP-9825 program, the data must be located on track one of the cassette tape. The data crunching program (given in the previous section) automatically takes the stored data in the large arrays on track zero of the data tape and crunches the large file to three smaller files on track one. If track zero is used, this information must be conveyed in the program.

The program listings for the two programs for the HP-9825 and the COMPAL-80 are given in Program Listings III-1 and III-2, respectively. The programs are quite similar functionally and structurally with basic differences being similar to those already pointed out in the previous section.

Figure III-1 is a sample printout of two data runs using the HP-9825 program and processor. The data is from a time period a little earlier than that given in the sample printout in Section II (Figure II-1).

FigureIII-2 gives the flow diagram for the simplified model program. This diagram is a functional diagram, and, therefore, applies to both processing systems program (HP-9825 and COMPAL-80).

Program Listing III-1. Simplified-model Prediction Program for Computing Breaker Statistics -- HP-9825 Version

```
0: "PROGRAM TO COMPUTE WAVE HEIGHTS & PERIODS FROM DATA OF SEA BUOY":
1: trk 1;fmt 1,f2.0,x,f2.0,x,f8.5,f3.0,f4.0,f3.0
2: "INITIALIZATION OF ARRAYS AND VARIABLES; DATA FILE READ IN":
3: dim D$[129,32];ent "How many files?",R;ent "What is first file #",F
4: dim H[356];dim P[356];dim G$[30];dim W[1024];for K=1 to R;0→L→E;ldf F,D$
5: "WAVES ON EVERY OTHER CHANNEL, STORED IN HI & LO BYTES OF STRING":
6: "WAVE DATA CONVERTED TO VOLTS AND STORED IN ARRAY":
7: for J=1 to 128;for I=1 to 29 by 4;num(D$[J,I,I])→A
8: num(D$[J,I+1,I+1])→B;shf(A,4)→C;band(A,15)→A
9: if C#(I-1)/2;beep;E+1→E
10: L+1→L;.002442*(256*A+B)-5→W[L]
11: wrt .1,C,(I-1)/2,W[L],E,J,K;next I;next J;F+1→F
12: "AVERAGE OF WAVE DATA CALCULATED AND SUBTRACTED FROM EACH DATUM":
13: "HALF WAVE CALCULATED BY TAKING LARGEST ABSOLUTE VOLTAGE BETWEEN ZERO":
14: "CROSSINGS; PERIOD EQUAL TO THE # OF POINTS BETWEEN TIMES DELTATIME":
15: 0→J→S;for I=1 to 1024;S+W[I]→S;next I
16: S/1024→A;for I=1 to 1024;W[I]-A→W[I];next I;-9999→X;9999→M
17: 0→N;if W[1]>0;gto "neg init"
18: "pos init":for L=1 to 1024;if W[L]>0;gto "pos wav"
19: next L
20: "neg init":for L=1 to 1024;if W[L]<0;gto "neg wav"
21: next L
22: "pos wav":for I=L to 1024;if W[I]<0;X→H;gto "array"
23: if W[I]>X;W[I]→X
24: N+1→N;next I;gto "p int"
25: "neg wav":for I=L to 1024;if W[I]>0;M→H;gto "array"
26: if W[I]<M;W[I]→M
27: N+1→N;next I;gto "print"
28: "array":J+1→J;N*.21333→P[J];abs(H)→H[J];0→N;I→L
29: -9999→X;9999→M;if W[I]<0;gto "neg wav"
30: gto "pos wav"
31: "BEGINNING OF PRINT ROUTINES":
32: "print":if Jmod2#0;J-1→J
33: for I=1 to J-1 by 2;H[I]+H[I+1]→H[I];P[I]+P[I+1]→P[I];next I
*23985
```

Program Listing III-1. (Cont)

```
fmt 2,3x,"TEST DATE ",c8,"    TEST TIME ",c8,"    RUN #",f3.0,"-",f2.0
fmt 3,3x,"AVG: ",f6.2," RMS: ",f6.2;num(D$[129,21,21])→G
fmt 4,20x,"PREDICTED SURFACE WAVE HEIGHT IN FEET"
fmt 5,20x,"PEAK-TO-TROUGH WAVE HEIGHT IN FEET"
fmt 6,20x,"PEAK-TO-PEAK WAVE PERIOD IN SECONDS"
fmt 7,20x,"PREDICTED WAVE BREAKER HEIGHT IN FEET"
fmt 8,x,f6.2,z;for L=1 to 20;""→G$[L,L];next L
"CRUNCHING WAVE HEIGHT AND PERIOD ARRAYS TO ELIMINATE THOSE OF LESS":
"THAN THREE SECONDS":
0→L;for I=1 to J by 2;if P[I]<3;jmp 2
L+1→L;H[I]→H[L];P[I]→P[L]
next I;D$[129,1,2]→G$[1,2];"/"→G$[3,3];D$[129,4,5]→G$[4,5]
D$[129,5,6]→G$[10,11];"/79"→G$[6,8];wrt 15.2,G$[1,8],D$[129,7,14],G,K
"PRINT EACH RAW WAVE HEIGHT IN FEET,CALCULATE & PRINT AVG & RMS":
wrt 15;wrt 15.5;0→S→Q;for I=1 to L by 10;for N=I to I+9;if N>L;jmp 2
4*H[N]→H;S+H→S;Q+H*H→Q;wrt 15.8,H;next N;wrt 15;next I
L→P;gsb "calc"
"PRINT EACH WAVE PERIOD IN SECONDS;CALCULATE & PRINT AVG AND RMS":
wrt 15.6;0→S→Q
for I=1 to L by 10;for N=I to I+9;if N>L;jmp 2
S+P[N]→S;Q+P[N]*P[N]→Q;wrt 15.8,P[N];next N;wrt 15;next I
gsb "calc"
"CALCULATE BREAKER HEIGHT AND PRINT EACH HEIGHT,AVERAGE AND RMS":
wrt 15.7;4*.89*(15.5+3)→C;0→S→Q
for I=1 to L by 10;for N=I to I+9;if N>L;jmp 2
((H[N]*4)↑4*C)↑.2→H;S+H→S;Q+H*H→Q;wrt 15.8,H;next N;wrt 15;next I
gsb "calc"
"CALCULATE SURFACE WAVE HEIGHTS AND PRINT EACH, AVERAGE, AND RMS":
wrt 15.4;0→S→Q;for I=1 to L by 10;for N=I to I+9;if N>L;jmp 5
if P[N]<3;999999→H[N];jmp 3
P[N]→A;-.954337632+.509299569*A-.048627407*A*A+.001611285*A↑3→A
S+H[N]*4/A→S;Q+(H[N]*4/A)↑2→Q
wrt 15.8,H[N]*4/A;next N;wrt 15;next I
gsb "calc"
wrt 15;wrt 15;wait 9999;next K;trk 0;dsp "end of  program";end
"COMPUTE AVERAGE AND RMS FROM SUM AND SUMSQ PASSED AND PRINT BOTH":
"calc":S/P→A;rabs((Q-S*S/P)/(P-1))→Q;wrt 15;wrt 15.3,A,Q;ret
*563
```

Program Listing III-2. Simplified-model Prediction Program for Computing Breaker Statistics -- COMPAL-80 Version

```
C       PROGRAM NAME IS BODATA READS DATA FROM MEMORY AND CHECKS
C       ADDRESS AND PUTS DATA INTO 1024-DATA ARRAY
        DIMENSION IDAT(1024),IBAD(10),IADD(8)
        DIMENSION IWH(256),IWP(256),WAVHT(128),WAVPER(128)
C       INITIALIZE VARIABLES AND ARRAYS NEXT
        DEPTH=16.0
        HEIGHT=3.0
        CNV=4.0*.002442
        DELTAT=.21333
        KTL=1
        KNT=1
        MEM=Z'4000'
        KOUNT=0
        SUM=0
        DO 5 I=1,1024
5       IDAT(I)=0
        DO 6 I=1,10
6       IBAD(I)=0
        DO 7 I=1,8
7       IADD(I)=0
150     FORMAT(1X,9I7)
200     FORMAT(1X,16I4)
250     FORMAT(1X)
300     FORMAT(1X,5I10)
31      FORMAT(1X,'    TEST DATE:  ',I2,'/',I2,'/79          TEST',
       1' TIME:  '
       2I2,':',I2,' - ',I2,':',I2,'         RUN NUMBER:  ',I2,'-',I2)
32      FORMAT('0','                               PEAK-TO-TROUGH
       1WAVE HEIGHT IN FEET')
33      FORMAT(10F9.2)
34      FORMAT('0','                               PEAK-TO-PEAK
       1 WAVE PERIOD IN SECONDS')
35      FORMAT('0','                              PREDICTED WAVE
       1BREAKER HEIGHT IN FEET')
36      FORMAT(1X)
C       READ IN DATA STARTING FROM MEMORY LOCATION
C       4000 HEX. DATA IN HIGH BYTE/LO BYTE FORMAT;
C       HIGH BYTE FIRST.
        DO 109 J=1,128
        DO 100 I=1,15,2
        LOC=MEM+(I-1)*2+(J-1)*32
        IHI=PEEK(LOC)
        IHI=(IHI .AND. 255)
        LO=PEEK(LOC+1)
        LO=(LO .AND. 255)
        II=(I+1)/2
        IADD(II)=IHI/16
        IF(IADD(II) .NE. (I-1))GO TO 80
        IHI=(15 .AND. IHI)
        NUM=IHI*256+LO
        IDAT(KNT)=NUM
        KNT=KNT+1
        GO TO 100
80      IBAD(KTL)=LOC
```

Program Listing III-2. (Cont)

```
          KTL=KTL+1
          KNT=KNT+1
100       CONTINUE
          WRITE(5,150)(IADD(I),I=1,8),J
109       CONTINUE
          PAUSE
          WRITE(5,200)(IDAT(I),I=1,1024)
          PAUSE
          WRITE(5,300)(IBAD(I),I=1,10),KTL
COMMENT INITIALIZE TO START DATA PROCESSING FROM ARRAYS
COMMENT AVERAGE REMOVED FROM WAVE ARRAY NEXT
COMMENT NOTE: DATA STILL IN 'BIT' FORM- NOT VOLTS OR
COMMENT E.U. CNV CONVERTS BITS TO VOLTS TO FEET
          DO 10 I=1,1024
10        SUM=SUM+IDAT(I)
          ISUM=SUM/1024
          DO 14 I=1,1024
14        IDAT(I)=IDAT(I)-ISUM
COMMENT START DETERMINING WAVE PEAKS AND TROUGHS
COMMENT AND WAVE PERIODS NEXT
          I4=0
          IF(IDAT(1) .GT. 0)GO TO 12
          DO 11 I3=1,1024
11        IF(IDAT(I3) .GT. 0)GO TO 15
12        DO 13 I3=1,1024
13        IF(IDAT(I3) .LT. 0) GO TO 17
COMMENT POINTS GTREATER THAN ZERO BETWEEN ZERO CROSSINGS
15        MAX=-32767
          DO 16 I1=I3, 1024
          IF(IDAT(I1) .LT. 0) GO TO 28
          IF(IDAT(I1) .GT. MAX) MAX=IDAT(I1)
16        I4=I4+1
          GO TO 25
COMMENT POINTS LESS THAN ZERO BETWEEN ZERO CROSSINGS NEXT
17        MIN=32767
          DO 18 I1=I3, 1024
          IF(IDAT(I1) .GT. 0) GO TO 29
          IF(IDAT(I1) .LT. MIN) MIN=IDAT(I1)
18        I4=I4+1
          GO TO 25
```

Program Listing III-2. (Cont)

```
COMMENT WAVE HEIGHT AND WAVE PERIOD CALCULATED NEXT
28      IHT=MAX
        GO TO 20
29      IHT=MIN
20      KOUNT=KOUNT+1
        IWP(KOUNT)=I4
        IWH(KOUNT)=IHT
        I4=0
        I3=I1
        IF( IDAT(I1) .LT. 0) GO TO 17
        GO TO 15
COMMENT WAVE HEIGHT(FT) AND WAVE PERIOD(SEC)
COMMENT CALCULATED NEXT
25      KNT=0
        KOUNT=KOUNT-1
        DO 27 I=1,KOUNT,2
        KNT=KNT+1
        WAVHT(KNT)=(ABS(IWH(I))+ABS(IWH(I+1)))*CNV
27      WAVPER(KNT)=(IWP(I)+IWP(I+1))*DELTAT
        PAUSE PRINT
        WRITE(5,31)L1,L2,L3,L4,L5,L6,L7,L8
        WRITE(5,32)
        WRITE(5,36)
        WRITE(5,33)(WAVHT(I),I=1,KNT)
        WRITE(5,34)
        WRITE(5,36)
        WRITE(5,33)(WAVPER(I),I=1,KNT)
COMMENT CALCULATE PREDICTED BREAKER WAVE HEIGHT NEXT
        CONST=.89*(DEPTH+HEIGHT)
        DO 24 I=1,KNT
        WAVHT(I)=((WAVHT(I))**4)*CONST
24      WAVHT(I)=(WAVHT(I))**(.2)
        WRITE(5,35)
        WRITE(5,36)
        WRITE(5,33)(WAVHT(I),I=1,KNT)
        END
```

Figure III-1. Sample Data Run for Simplified Prediction Model

TEST DATE 12/06/79   TEST TIME 17:23:31   RUN # 6- 1

PEAK-TO-TROUGH WAVE HEIGHT IN FEET

0.80	0.82	1.58	2.21	2.26	1.95	1.03	0.64	2.90	6.70
2.81	1.80	0.85	0.86	0.85	1.07	1.10	2.00	1.66	1.04
1.50	0.69	0.90	2.07	2.47	2.36	2.22	1.92	1.63	1.23
0.75	0.77								
AVG:	1.67	RMS:	1.14						

PEAK-TO-PEAK WAVE PERIOD IN SECONDS

3.63	3.63	7.47	6.83	8.11	6.40	4.91	4.27	6.40	7.25
7.47	7.25	4.27	7.47	4.05	4.69	6.19	7.04	5.97	7.68
6.61	4.27	4.69	7.68	7.47	6.61	6.83	7.25	6.40	4.91
5.76	3.84								
AVG:	6.04	RMS:	1.41						

PREDICTED WAVE BREAKER HEIGHT IN FEET

1.93	1.97	3.34	4.35	4.43	3.95	2.36	1.63	5.42	10.58
5.29	3.69	2.03	2.05	2.03	2.45	2.50	4.03	3.47	2.38
3.20	1.72	2.12	4.14	4.77	4.60	4.37	3.90	3.42	2.73
1.84	1.88								
AVG:	3.39	RMS:	1.74						

PREDICTED SURFACE WAVE HEIGHT IN FEET

2.43	2.49	1.96	2.87	2.70	2.66	1.82	1.41	3.94	8.41
3.48	2.26	1.85	1.06	2.03	2.02	1.54	2.56	2.39	1.26
2.00	1.51	1.69	2.53	3.06	3.14	2.88	2.42	2.22	2.18
1.12	2.05								
AVG:	2.44	RMS:	1.28						

Figure III-2. Flow Diagram for the Simplified Prediction Model Program.

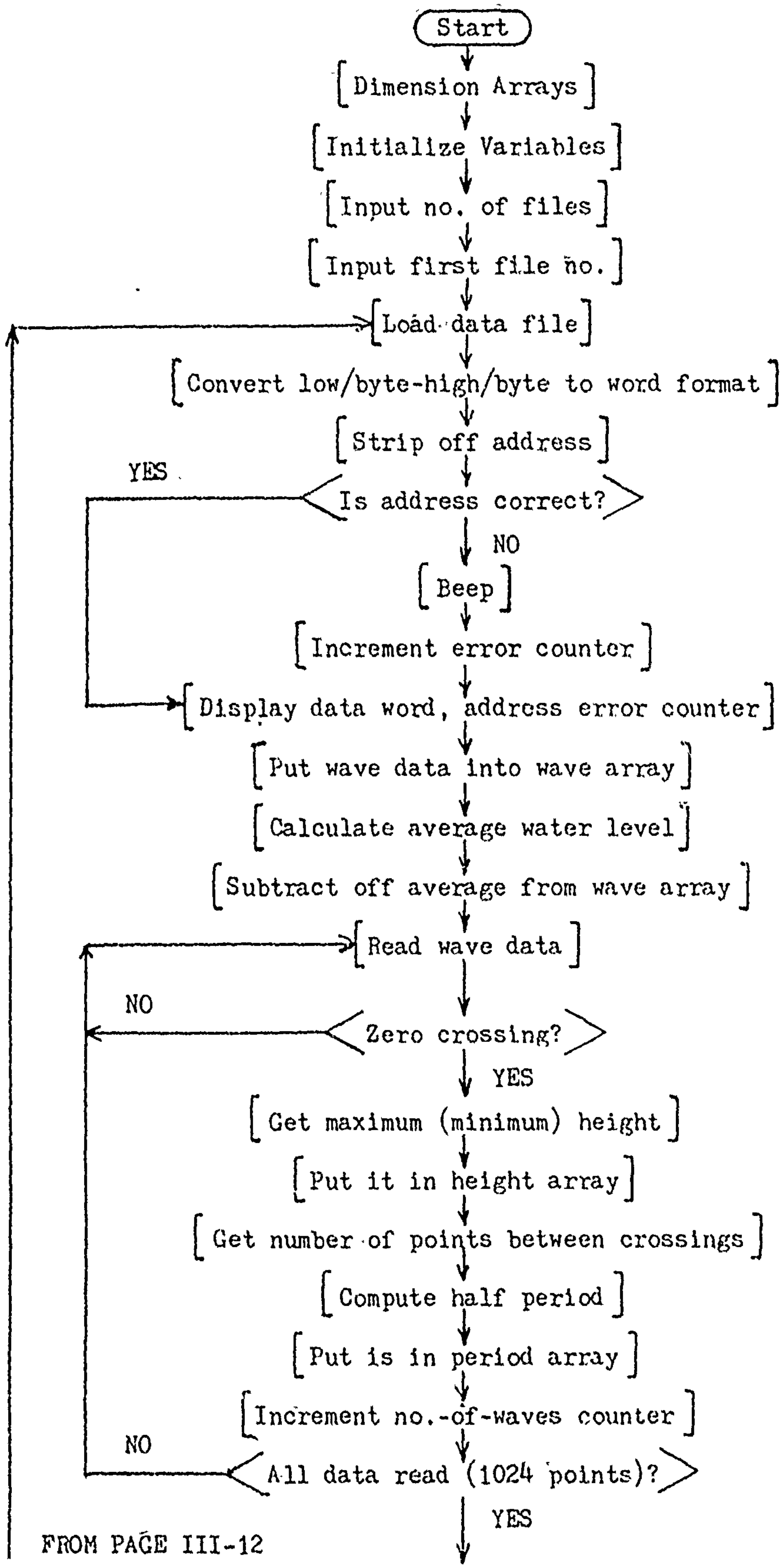


Figure III-2. (Cont)

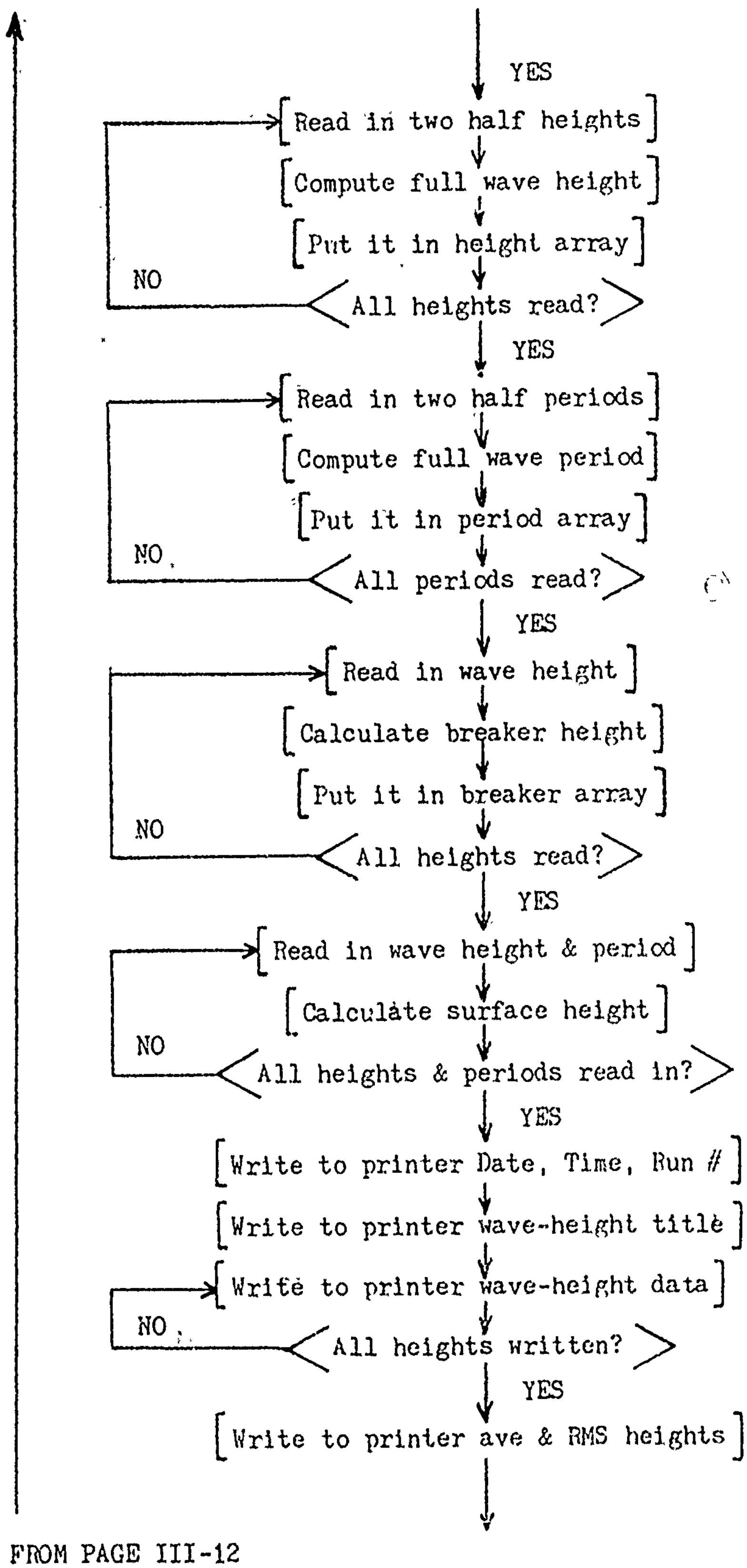
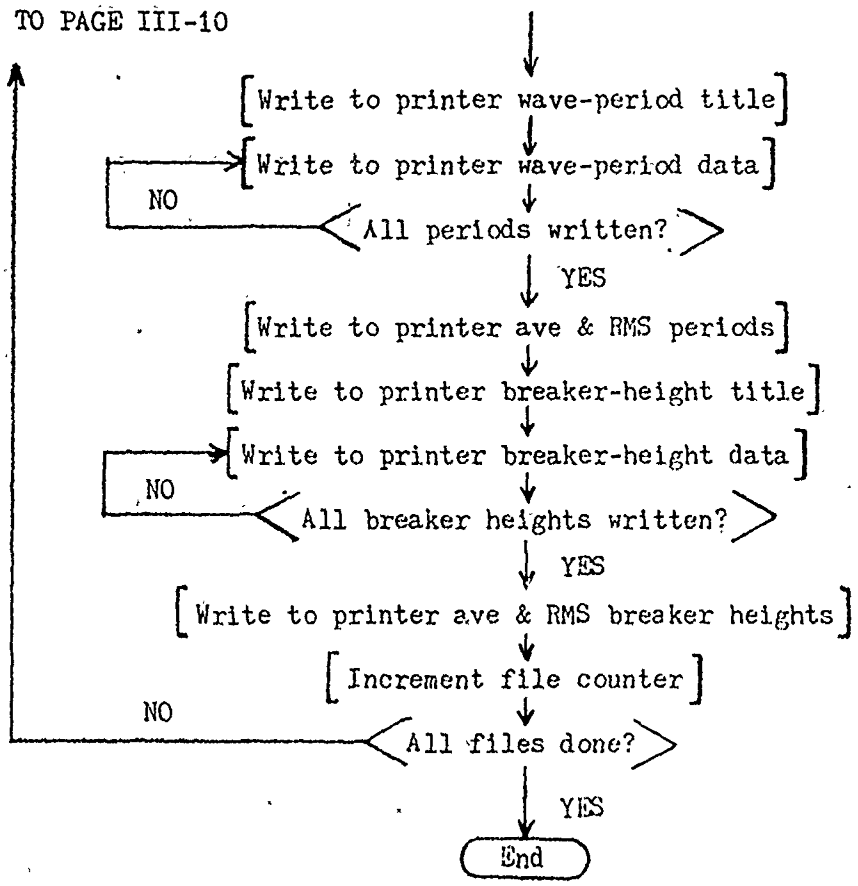


Figure III-2. (Cont)



SECTION IV

DATA PROCESSING USING THE DETAILED MODEL

In this section is given the background information and program development for the detailed prediction model. This model also predicts the individual breaker heights and statistics, the breaker depths, and the longshore currents. Several major differences exist between the two prediction programs. In the detailed model wave period is used directly in the calculation of the breaker height removing the limitation due to the measurement being made at the bottom and hydrodynamic attenuation of the waves not being accounted for. Also, in this model the approach angle of the waves is considered both in terms of the breaker height and the longshore current.

Because of these changes, the prediction portion of this program is considerably more complex than for the simplified model. Not only are the prediction equations more complex, but in several places in the program, the calculations are done by an iterative process, requiring many loops of processing and refining of the calculations. For processing where time and memory are not restricted, these factors are not important. However, for onboard processing, these factors may significantly influence the design. These considerations are not part of the present study, but are to be included in the next phase of this program.

For the prediction of the breaker heights, the detailed model takes the measured wave heights and periods from the pressure-gauge data and computes the extrapolated surface wave heights as given by:

$$H_s = H_m \frac{\cosh(2\pi d/L)}{\cosh(2\pi b/L)}$$

where L is given by:

$$L = (gT^2/2\pi)\tanh(2\pi d/L)$$

where H_s is the surface wave height, H_m is the height measured at the sensor, d is the total water depth at the buoy, b is the height of the pressure sensor above the bottom, L is the wave length, g is acceleration of gravity, and T is the measured wave period.

This computed surface wave height is then used to obtain a preliminary breaker height obtained from:

$$H_b = \left(.98^2 \, (.90h_0)^{1/4} \, H_s \right)^{4/5}$$

where H_b is the predicted breaker height, h_0 is the mean water depth at the buoy, and H_s is the extrapolated surface wave height at the buoy. This initial calculation is then used as the input wave height in an iterative routine to calculate the refined breaker where the refraction, friction, and shoaling coefficients are updated with each new calculation. This procedure is given by:

$$h_b = \frac{H_b}{b - aH_b/gT^2}$$

$$a = 1.36g(1 - e^{-19M})$$

$$b = 1.56/(1 + e^{-19.5M})$$

$$\theta_b = \sin^{-1}\left[\frac{C_b}{C}\sin(\theta)\right]$$

$$C = L/T$$

$$C_b = \left[g(H_b + h_b)\right]^{1/2}$$

$$L_b = C_bT$$

where M is the slope of the beach, θ is the wave angle approach to the beach, θ_b is the breaker angle, C is the wave celerity at the buoy, C_b is the breaker celerity, and L_b is the breaker wave length, with the other parameters as previously given.

From these initial calculations an improved breaker calculation can be made based upon the following:

$$H_b = H_s K_f K_r K_s$$

$$K_f = .98$$

$$K_r = \left[\frac{\cos(\theta)}{\cos(\theta_b)}\right]^{1/2}$$

$$K_s = \left[\frac{\tanh(2\pi h/L)}{\tanh(2\pi h_b/L_b)} \cdot \frac{1 + \frac{4\pi h/L}{\sinh(4\pi h/L)}}{1 + \frac{4\pi h_b/L_b}{\sinh(4\pi h_b/L_b)}}\right]^{1/2}$$

where the K-factors are the breaker coefficients and the other parameters are as given.

The breaker height computed from this second breaker-height equation is then compared with the previous breaker-height calculation. If the difference between the two heights does not exceed a given amount, the process is ended. If the difference is larger than the given amount, the calculations are repeated, using this last breaker height in the breaker depth equation on the previous page and all of the following calculations repeated.

The maximum longshore current can then be computed from the above results using:

$$V_m = 20.7(\tan M)(g\bar{H}_b)^{1/2}\sin(2\theta_b)$$

The program listings for the programs for the two systems are given in Program Listings IV-1 and IV-2. The previous comments in Section III regarding the two programs also apply to the ones given in this section. Figure IV-1 is a sample printout of two data runs from the same time period as Figure III-1 in Section III and was also done on the HP-9825.

Program Listing IV-1. Detailed-model Prediction Program for Computing Breaker Statistics -- HP-9825 Version

```
0: "PROGRAM TO COMPUTE WAVE HEIGHTS AND PERIODS FROM WAVE DATA OF SEA BUOY":
1: "COMPUTES CURRENT DIRECTION FROM X AND Y COMPONENTS;FORMS HISTOGRAM":
2: dim A[18];fmt 1,f2.0,x,f2.0,x,f8.5,f3.0,f4.0,f3.0;dim Z$[38];trk 1
3: dim D$[129,32];ent "How many files?",R;ent "What is first file #",F
4: dim H[290];dim P[290];dim G$[12];dim W[1024];for K=1 to R;0→L→E→Q→S→V→X→Y
5: "DATA FILE READ INTO STRING; WAVES STORED ON EVERY OTHER CHANNEL;":
6: "LONGSHORE CURRENT ON CHANNELS 3 AND 10;ONSHORE ON CHANNELS 4 AND 11":
7: "ALL STORED IN HIGH AND LOW BITES OF STRING;WAVE DATA CONVERTED TO":
8: "VOLTS AND STORED IN ARRAY;CURRENT DIRECTION COMPUTED FROM CURRENT":
9: "COMPONENTS AND STORED AS HISTOGRAM ARRAY;DIRECTIONS SUMMED & SUMSQED":
10: ina A;ldf F,D$;.05→O;for J=1 to 128;for I=1 to 29 by 4;num(D$[J,I,I])→A
11: num(D$[J,I+1,I+1])→B;shf(A,4)→C;band(A,15)→A
12: if C#(I-1)/2;beep;E+1→E
13: L+1→L;.002442*(256*A+B)-5→W[L]
14: wrt .1,C,(I-1)/2,W[L],E,J,K;next I;for I=3 to 19 by 16
15: num(D$[J,I,I])→A;num(D$[J,I+1,I+1])→B;band(A,15)→A
16: .002442*(256*A+B)-5→C;C+X→X;num(D$[J,I+4,I+4])→A
17: num(D$[J,I+5,I+5])→B;band(A,15)→A;.002442*(256*A+B)-5→C
18: C+Y→Y;next I;next J;X/256→r1;Y/256→r2;F+1→F;for J=1 to 128
19: for I=3 to 19 by 16;num(D$[J,I,I])→A;num(D$[J,I+1,I+1])→B;shf(A,4)→C
20: band(A,15)→A;.002442*(256*A+B)-5-r1→X;num(D$[J,I+4,I+4])→A
21: num(D$[J,I+5,I+5])→B;shf(A,4)→C;band(A,15)→A;.002442*(256*A+B)-5-r2→Y
22: atn(Y/X)→A;if A<0;A+180→A
23: V+A→V;Q+A*A→Q;int(A/10)→A;A[A+1]+1→A[A+1];next I;next J;256→B
24: "AVERAGE OF WAVE DATA CALCULATED AND SUBTRACTED FROM EACH DATUM;":
25: "HALF WAVE CALCULATED BY TAKING LARGEST ABSOLUTE VOLTAGE BETWEEN ZERO":
26: "CROSSINGS;PERIOD EQUAL TO # OF PTS. BETWEEN CROSSINGS TIMES DELTATIME":
27: 0→J→S;for I=1 to 1024;S+W[I]→S;next I;3.14159→U
28: S/1024→A;for I=1 to 1024;W[I]-A→W[I];next I;-9999→X;9999→M
29: 0→N;if W[1]>0;gto "negb"
30: "posb":for L=1 to 1024;if W[L]>0;gto "pos"
31: next L
32: "negb":for L=1 to 1024;if W[L]<0;gto "neg"
33: next L
34: "pos":for I=L to 1024;if W[I]<0;X→H;gto "array"
35: if W[I]>X;W[I]→X
36: N+1→N;next I;gto "print"
37: "neg":for I=L to 1024;if W[I]>0;M→H;gto "array"
38: if W[I]<M;W[I]→M
39: N+1→N;next I;gto "print"
40: "array":J+1→J;N*.21333→P[J];abs(H)→H[J];0→N;I→L
41: -9999→X;9999→M;if W[I]<0;gto "neg"
42: gto "pos"
43: "WHOLE WAVE HEIGHTS AND PERIODS FORMED":
44: for I=1 to J-1 by 2;H[I]+H[I+1]→H[I];P[I]+P[I+1]→P[I];next I
45: "BEGINNING OF PRINT ROUTINES":
46: "print":if Jmod2#0;J-1→J
47: fmt 2,3x,"TEST DATE ",c8,"    TEST TIME ",c8,"    RUN #",f3.0,"-",f2.0
48: fmt 3,3x,"AVG: ",f6.2,"  RMS: ",f6.2;num(D$[129,21,21])→G
49: fmt 5,20x,"PEAK-TO-TROUGH WAVE HEIGHT IN FEET"
50: fmt 6,20x,"PEAK-TO-PEAK WAVE PERIOD IN SECONDS"
*29354
```

Program Listing IV-1. (Cont)

```
51: fmt 7,20x,"PREDICTED WAVE BREAKER HEIGHT IN FEET"
52: fmt 8,x,f6.2,z;for L=1 to 12;""→G$[L,L];next L
53: fmt 9,f7.0,z;"   OFFSHORE WAVE DIRECTION HISTOGRAM"→Z$
54: D$[129,1,2]→G$[1,2];"/"→G$[3,3];D$[129,4,5]→G$[4,5];D$[129,5,6]→G$[10,11
55: "/79"→G$[6,8];wrt 15.2,G$[1,8],D$[129,7,14],G,K;wrt 15;wrt 15,Z$
56: "PRINTING OF DIRECTION HISTOGRAM AND CALULATION OF AVERAGE AND RMS":
57: for N=1 to 2;(N-1)*10+1→L
58: for I=L to L+9;if I>18;jmp 2
59: wrt 15.9,I*10;next I
60: wrt 15;for I=L to L+9;if I>18;jmp 2
61: wrt 15.9,A[I];next I
62: wrt 15;next N;B→P;V→S;gsb "calc"
63: "CRUNCHING OF WAVE HEIGHT AND PERIOD ARRAYS TO ELIMINATE THOSE":
64: "OF LESS THAN THREE SECONDS":
65: A→Z;0→L;for I=1 to J by 2;if P[I]<3;jmp 2
66: L+1→L;H[I]→H[L];P[I]→P[L]
67: next I
68: "SURFACE WAVE HEIGHT ARRAY RECALCULATED USING ATTENUATION FACTOR":
69: for I=1 to L;P[I]→A;-.95433763+.509299569A-.048627407A↑2+.001611285A↑3→A
70: 4(H[I]/A)→H[I];next I;1.36(1-exp(-19*O))→C;1.56/(1+exp(-19.5*O))→E
71: "PRINT INDIVIDUAL SURFACE WAVE HEIGHTS, AVERAGE AND RMS":
72: wrt 15.5;0→S→Q;for I=1 to L by 10;for N=I to I+9;if N>L;jmp 2
73: H[N]→H;S+H→S;Q+H*H→Q;wrt 15.8,H;next N;wrt 15;next I
74: wrt 15;L→P;gsb "calc"
75: wrt 15.6;0→S→Q;abs(Z-90)→Z;sin(Z)→r30;abs(cos(Z))→r31
76: "PRINT EACH WAVE PERIOD IN SECONDS;CALCULATE AND PRINT AVERAGE AND RMS":
77: for I=1 to L by 10;for N=I to I+9;if N>L;jmp 2
78: S+P[N]→S;Q+P[N]↑2→Q;wrt 15.8,P[N];next N;wrt 15;next I
79: wrt 15;gsb "calc"
80: "CALCULATE INDIVIDUAL PREDICTED BREAKER HEIGHTS,AVG, AND RMS AND PRINT":
81: 0→S→Q;for I=1 to L;(.98↑2*(.9*19)↑.25*H[I])↑.8→H;P[I]→T
82: "CONVERGENCE LOOP FOR PREDICTED BREAKER HEIGHT":
83: "rout":H/(E-C*H/T↑2)→D;gsb "wavlgth"
84: W/T→r11;r(32.2*(H+D))→r12;r12*T→B
85: r12*r30/r11→r13;r(1-r13↑2)→C
86: 2*U*19/W→r1;2*U*D/B→r2;(1-exp(-2*r1))/(1+exp(-2*r1))→r3
87: (1-exp(-2*r2))/(1+exp(-2*r2))→r4;(exp(2*r1)-exp(-2*r1))/2→r5
88: (exp(2*r2)-exp(-2*r2))/2→r6;r(r3/r4*(1+2*r1/r5)/(1+2*r2/r6))→r1
89: r(r31/C)→r2;H[I]*.98*r1*r2→G;if abs(G-H)>.01;G→H;gto "rout"
90: G→H[I];S+H[I]→S;Q+H[I]*H[I]→Q;next I;wrt 15.7;for I=1 to L by 10
91: for N=I to I+9;if N<L+1;wrt 15.8,H[N];next N;wrt 15;next I
92: wrt 15;L→P;gsb "calc"
93: wrt 15;wrt 15;next K;trk 0;dsp "end";end
94: "COMPUTE AVERAGE AND RMS FROM SUM AND SUMSQ PASSED AND PRINT BOTH":
95: "calc":S/P→A;rabs((Q-S*S/P)/(P-1))→Q;wrt 15.3,A,Q;ret
96: "INDIVIDUAL WAVE LENGTH COMPUTED FROM PERIOD PASSED":
97: "wavlgth":32.2*T*T/(2*U)→r4;r4→r7
98: 2*U*19/r7→r5;exp(2*r5)→r6;r4*(r6-1)/(r6+1)→r8
99: abs(r8-r7)/r7→r9;if r9<.0001;jmp 2
100: (r8+r7)/2→r7;jmp -2
101: r8→W;ret
*31769
```

Program Listing IV-2. Detailed-model Prediction Program for computing Breaker Statistics -- FORTRAN Version

```
      PROGRAM BODATA
COM  PROGRAM NAME IS BODATA READS DATA FROM MEMORY AND CHECKS
COM  ADDRESS AND PUTS DATA INTO 1024-DATA ARRAY
COM  PROGRAM DETERMINES INDIVIDUAL WAVE HEIGHTS AND PERIODS
COM  USING A ZERO CROSSING METHOD. NEXT THE BREAKER HEIGHTS
COM  ARE COMPUTED USING AN ITERATION PROCESS
COM  FINALLY WAVE HEIGHTS, PERIODS, AND BREAKER HEIGHTS ARE PRINTED.
COM  INITIALIZE VARIABLES AND ARRAYS NEXT.
      DIMENSION IDAT(1024),IBAD(10),IADD(8)
      DIMENSION IWH(256),IWP(256),WAVHT(128),WAVPER(128)
      DO 6 I=1,10
6     IBAD(I)=0
      DO 5 I=1,8
5     IADD(I)=0
31     FORMAT(' BREAKER ANGLE:',F6.1)
32    FORMAT('0','                                   PEAK-TO-TROUGH
     1WAVE HEIGHT IN FEET')
33   FORMAT(10F9.2)
34   FORMAT('0','                                    PEAK-TO-PEAK
     1 WAVE PERIOD IN SECONDS')
35   FORMAT('0','                                 PREDICTED WAVE
     1BREAKER HEIGHT IN FEET')
36    FORMAT(1X)
COM  READ IN DATA STARTING FROM MEMORY LOCATION 4000 HEX.
COM  DATA IN HIGH BYTE/LOW BYTE FORMAT, HIGH BYTE FIRST.
      KTL=1
      KNT=1
      MEM=Z'4000'
      DO 109 J=1,128
      DO 100 I=1,15,2
      LOC=MEM+(I-1)*2+(J-1)*32
      IHI=PEEK(LOC)
      IF(IHI .GE. 0)GO TO 40
      IHI=IHI+256
40    LO=PEEK(LOC+1)
      IF(LO .GE. 0)GO TO 50
      LO=LO+256
50    II=(I+1)/2
      IADD(II)=IHI/16
      IF((IADD(II)) .NE. (I-1))GO TO 80
      IHI=(15 .AND. IHI)
      NUM=IHI*256+LO
      IDAT(KNT)=NUM
      KNT=KNT+1
      RNT=RNT+2
800   CONTINUE )=LOC
109   CONTINUE1
```

```
      PAUSE
      PAUSE
COMMENT INITIALIZE TO START DATA PROCESSING FROM ARRAYS
COM   AVERAGE REMOVED FROM WAVE ARRAY NEXT
COM   NOTE! DATA STILL IN 'BIT' FORM -- NOT VOLTS OR E. U.
COM   CNV CONVERTS BITS TO VOLTS TO FEET.
       CNV=4.0*.002442
       DELTAT=.2113
      KOUNT=0
      ISUM=0
      DO 10 I=1,1024
10    ISUM=ISUM+IDAT(I)
      ISUM=ISUM/1024
      DO 14 I=1,1024
14    IDAT(I)=IDAT(I)-ISUM
COM   START DETERMINING WAVE PEAKS, TROUGHS, AND PERIODS NEXT.
      MAX=-32767
      MIN=32767
      I4=0
      IF( IDAT(1) .GT. 0 ) GO TO 12
      DO 11 I3=1,1024
11    IF( IDAT(I3) .GT. 0 ) GO TO 15
12    DO 13 I3=1,1)24
13    IF( IDAT(I3) .LT. 0 ) GO TO 17
COMMENT POINTS GTREATER THAN ZERO BETWEEN ZERO CROSSINGS
15    DO 16 I1=I3, 1024
      IF( IDAT(I1) .LT. 0 ) GO TO 28
      IF( IDAT(I1) .GT. MAX ) MAX=IDAT(I1)
16    I4=I4+1
      GO TO 25
COMMENT POINTS LESS THAN ZERO BETWEEN ZERO CROSSINGS NEXT
17    DO 18 I1=I3, 1024
      IF( IDAT(I1) .GT. 0 ) GO TO 29
      IF( IDAT(I1) .LT. MIN ) MIN=IDAT(I1)
18    I4=I4+1
      GO TO 25
COMMENT WAVE HEIGHT AND WAVE PERIOD CALCULATED NEXT
28    IHT=MAX
      GO TO 20
29    IHT=MIN
20    KOUNT=KOUNT+1
      IWP(KOUNT)=I4
      IWH(KOUNT)=IHT
       MAX=-32766
       MIN=32767
      I4=0
      I3=I1
      IF( IDAT(I1) .LT. 0 ) GO TO 17
      GO TO 15
25    KNT=0
      KOUNT=KOUNT-1
```

```
      DO 27 I=1,KOUNT,2
      KNT=KNT+1
      WAVHT(KNT)=(ABS(IWF(I))+ABS(IWF(I+1)))*CNV
27    WAVPER(KNT)=(IWP(I)+IWP(I+1))*DELTAT*2
      PAUSE PRINT
      WRITE(5,32)
      WRITE(5,36)
      WRITE(5,33)(WAVHT(I),I=1,KNT)
      WRITE(5,34)
      WRITE(5,36)
      WRITE(5,33)(WAVPER(I),I=1,KNT)
COMMENT CALCULATE PREDICTED BREAKER WAVE HEIGHT NEXT
7      FORMAT(1X,'ENTER WAVE DIRN (E.G. 89.0)          ')
8      FORMAT(1X,'ENTER DEPTH (E. G. 25.6)          ')
9      FORMAT(1X,'ENTER SLOPE (E. G. .05)          ')
       WRITE(5,8)
       CALL ENTER(DEPTH)
       WRITE(5,9)
       CALL ENTER(SLOPE)
       WRITE(5,7)
       CALL ENTER(THETA)
COMMENT INITIALIZE VARIABLES NEXT
       PI=3.14159
       THETA=ABS(THETA-90.0)
       THETA=THETA*PI/180.0
       SUMTH=0.0
COMMENT START ITERATION ROUTINE FOR DETAILED PREDICTION MODEL
       A=1.36*(1-EXP(-19*SLOPE))
       B=1.56/(1+EXP(-19.5*SLOPE))
       DO 22 I=1,KNT
       H1=(.98**2*((.90*DEPTH)**.25)*WAVHT(I))**.8
       P=WAVPER(I)
21     D=H1/(B-A*H1/(P*P))
       CALL WAVLTH(P,WL,DEPTH)
       C=WL/P
       CB=SQRT(32.2*(H1+D))
       WLB=CB*P
       SINTHB=(CB/C)*SIN(THETA)
       COSTHB=SQRT(1-SINTHB**2)
       CF=.98
       CTHETA=SQRT(ABS((COS(THETA))/COSTHB))
       ARG1=2*PI*DEPTH/WL
       ARG2=2*PI*D/WLB
       TANH1=(1-EXP(-2*ARG1))/(1+EXP(-2*ARG1))
       TANH2=(1-EXP(-2*ARG2))/(1+EXP(-2*ARG2))
       SINH1=(EXP(2*ARG1)-EXP(-2*ARG1))/2
       SINH2=(EXP(2*ARG2)-EXP(-2*ARG2))/2
       CS=SQRT(TANH1/TANH2*((1+2*ARG)/SINH1)/(1+2*ARG2/SINH2)))
       H2=WAVHT(I)*CF*CTHETA*CS
       VAR=ABS(H2-H1)
COMMENT CHECK FOR ITERATION DIFFERENCES LESS THAN .01 ABSOLUTE
       IF (VAR .LT. 0.01)GO TO 19
       H1=H2
       GO TO 21
19     WAVHT(I)=H2
```

```
COM  BREAKER ANGLE DETERMINED NEXT (THETAB).
        OPP=SQRT(1-COSTHB**2)
        THETAB=ATAN(OPP/COSTHB)
        SUMTH=SUMTH+THETAB
22      CONTINUE
        THETAB=SUMTH/KNT
        WRITE (5,35)
        WRITE(5,36)
        WRITE (5,33)(WAVHT(I),I=1,KNT)
        WRITE(5,36)
        WRITE(5,31)THETAB
        END
        SUBROUTINE WAVLTH(P,WL,DEPTH)
COMMENT ROUTINE TO COMPUTE WAVE LENGTHS USING LINEAR WAVE THEORY
COMMENT ROUTINE ITERATES LENGTHS UNTIL RESULTS ARE WITHIN 0.01%.
        X1=32.2*P*P/(6.28319)
        WL1=X1
23      X=6.28319*DEPTH/WL1
        X2=EXP(2*X)
        WL2=X1*(X2-1)/(X2+1)
        WL3=ABS(WL2-WL1)/WL1
        IF (WL3 .LT. .0001) GO TO 26
        WL1=(WL2+WL1)/2
        GO TO 23
26      WL=WL2
        RETURN
        END
```

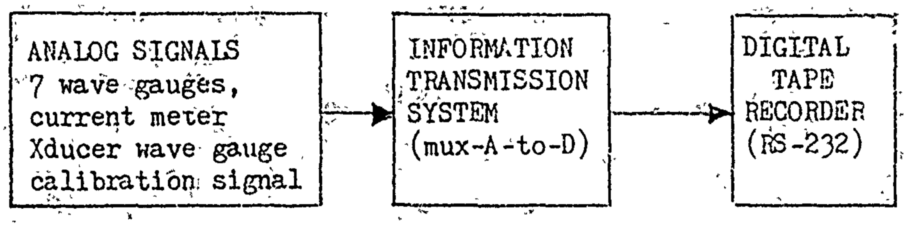
SECTION V

A FIELD DATA COLLECTION SYSTEM

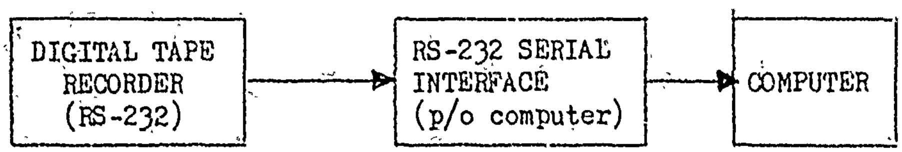
In this section are given the specific details of a self-contained data acquisition and recording system that was used for the collecting and recording of the wave data from the CERC Field Research Facility at Duck, N. C. The design of this system was simplified considerably by making use of extensive design and development studies done previously by Dynex personnel on other related Navy tasks. The heart of this data acquisition and recording system is the Information Transmission System (ITS) developed under other studies and modified in this study for the given application above. As described in Section I, ITS is also used as the central data acquisition system in the data buoy. Although discussed briefly in that section, specific information is given in this section in detail to allow an end user to adapt this system to his specific application. Because of its versatility, ITS should be readily adaptable for many Navy fleet applications. More complete details are given by Dynex reports, FR-XVI-1 and FR-XVI-2.

An overallblock diagram of the data acquisition and recording system is given in Figure V-1, with the data collecting and recording system given in the top part of the figure. The lower part of the figure depicts the manner in which the recorded data are played back into any standard computer with a serial (RS-232) interface port. The distinct advantages of the system are its low cost (around $600.00 for components and the digital tape recorder) and its high reliability (typical tape errors less than one part in 10^5 for tape densities of 1600 bits per inch or less and drift characteristics of about one part in four thousand for long term). It is also highly portable and uses easy-to-obtain digital or high-quality analog cassette tapes.

The specific details of the system are given in the following paragraphs. Up to 16 channels of analog information are input into the 16-channel multiplexer which sends the data sequentially to a 12-bit analog-to-digital converter for digitizing. The digitized output along with the channel address is sent to a UART which converts the parallel data to a bit-serial, RS-232 standard format which is recorded by the digital tape recorder. At power up, the system automatically commences its sequential channel scanning/digitizing operation until system power down. Since channel address is contained in each data word, all data are uniquely identified, making data recovery and processing by the computer an easy and reliable task.



Data Collecting and Recording System



Computer Processing Using Recorded Data

Figure V-1. Diagram of Field Data Collection System and Processing Application.

An internal clock in the ITS system controls all timing of the system. No input information from any processor is required; the system operates in a self-contained manner. A processor is not even required for the output information. The serial output data can be directly recorded on a digital recorder compatible with the system. This recording can be done in the field or at any location the user desires. After the data have been recorded, the digital tape information can be played back into the serial interface port of the processing system as if the data were being obtained live in real time. Since each recorded data word has the identically same format as the real time ITS data, each word contains the address of the analog input channel. In this manner, each word is uniquely identified. The data word is made up of two bytes (8 bits each): a high byte containing the address and the most significant data bits, and a low byte containing the rest of the data bits. The high byte data bits are bits 9, 10, 11, and 12 and are the first four bits of the high byte with data bit 9 being the least significant bit of the high byte. The four address bits form the upper part of the high byte with the most significant bit of the address being the most significant bit of the high byte. In the low byte, which contains data bits 1 through 8, the least significant bit of the data (bit 1) is the least significant bit of the low byte, and data bit 8 is the most significant bit of the low byte. This is shown pictorially in Table V-1.

HIGH BYTE	LOW BYTE
A4 A3 A2 A1 D12 D11 D10 D9	D8 D7 D6 D5 D4 D3 D2 D1

where A4 = MSB of address, A1 = LSB of address, D12 = MSB of data, D1 = LSB of data and the bits are given MSB to LSB, left to right in the byte

Table V-1. Data Word Bit Arrangement in the High Byte and Low Byte

Figures V-2 through V-4 give the sequentially-scanning mode diagrams. Figure V-2 is a functional block diagram; Figure V-3 is a block schematic-wiring diagram of the signal and power of the system; and Figure V-4 is the same type of diagram for the control and timing circuitry of the system. These figures should be referred to in the following discussion of the system.

A 1.8432 megaHertz crystal oscillator derives the master clock signal for the system. This clock signal is input to a baud-rate generator, which in turn derives the desired clock, timing, and control signals for the system operation. The clock signal for the UART (Universal Asychronous Receiver/Transmitter) is 16 times the selected baud rate for the system. Two additional clock signals are developed by the baud-rate generator: a divide by 16 and a divide by 32 of the baud rate selected. The first is used for developing the UART enable signal, and the second is used for the tri-state enable signals (one for the high-byte set and the inverted signal for the low-byte set of tri-states). The second clock signal is also used in the sequentially-scanning mode for setting the parity of the UART with the high byte having even parity and the low byte having odd parity. (Refer to a later discussion in this section on the use of even and odd parity by the processor to determine whether the byte being processed is high byte or a low byte.) In addition, the divide by 32 clock signal that is used to enable the high-byte tri-states is used to strobe (or start the process) of the multiplexer-digitizer (SDM 853).

The UART enable signal is developed by a D-type dual flip-flop, which inconjunction with the associated tri-state produces a signal that is high most of the time, is low for two fixed clock periods of 104 microseconds, and returns high again for the next cycle. The UART is enabled by the low signal, which means that data on the UART input lines (pins 26 through 33 of the CDP 1854) are available to the UART transmitter registers. When the UART enable signal goes high, the data on these lines are latched in the UART. Any changes on the data after that time are not seen by the UART until it is enabled again by a low signal. Also data that is changing while the UART is enabled has no effect; only the data that is there when the enable signal goes high (the rising edge of the UART enable signal) is latched by the UART to be serially sent out. When the data is latched in the UART, it is then serially clocked out at the selected baud rate. The following format was hard-wire selected for the UART: first bit out is the start bit, then the 8 data bits, 1 parity bit, and 2 stop bits.

The output data from the digitizer is connected to four tri-states which form the 8-bit bus system of the ITS. When a set of tri-states are enabled, they allow the data on the input to be passed through onto the bus. When the set is disabled, the tri-states in effect block the input data from appearing on the output bus. In this manner, when the high-byte tri-states are enabled, the data for the address and four most-significant bits appear on the output bus. When the low-byte tri-states are enabled, the data for the data bits 1 through 8 appear on the output bus. It should be noted that the SDM 853 is only strobed once during this cycle such that the data on the output lines are held constant while the high byte and then the low byte are put on the output bus and are transmitted serially high byte then low byte by the UART.

When the SDM 853 is strobed, the multiplexer is stepped to the next address (starting initially at channel 0), the analog data is then presented to the digitizer after a 9 microsecond delay to allow the data to settle before digitizing, and the data is digitized with the 12 data bits and address appearing on the output lines of the SDM 853 24 microseconds later. The entire process takes approximately 33 microseconds and, of course, is independent of the selected baud rate. During the time that the data is being digitized and data on the bus is being switched from one byte (either high or low) to the other byte, the UART is enabled. But the data is not latched at this time. The data is latched 104 microseconds after the start of the process to allow complete settling of all devices before latching to insure valid data in the UART to be transmitted.

The SDM 853 is continually strobed, stepping it through all 16 data channels, with two bytes being transmitted by the UART (a high byte and a low byte) for each data word. The cycle continually repeats itself since it is completely under internal clock control. The ITS does not check to see if the receiving device (a processor or a recorder) is ready. It puts out the data whether or not the receiving device is ready or even there. Therefore, it is necessary that the receiving device be capable of receiving the data bit stream at the rates being transmitted. It should be noted that the 104 microsecond UART enabling period would have to be shortened to prevent over-run of the UART stop bits being sent if the baud rate exceeds 38.4 kiloHertz. This enabling period must be shortened to prevent over-run of the parity and data bits for baud rates higher than 57.6 kiloHertz. This enabling period can be shortened by using a higher frequency than 19.2 kiloHertz out of the baud-rate generator. However, in shortening this enabling period, caution must be exercized not to make the period shorter than the total time period of 33 microseconds for the SDM 853. With care this can be done without exceeding the 200 kiloHertz baud-rate limit of the UART operating at 5 volts.

A major concern in reading the data is to be able to reconstruct the 12-bit data word with the 4-bit address. This can be done by first determining which of the received bytes is a high byte and which is a low byte. When a high byte is located, the address can be stripped off to determine which channel the data came from. Then the high byte and the following data byte, which by definition will be the low byte corresponding to that high byte, can be combined to form the original data word. The parity bit in each byte is checked to determine whether the parity is even (thus a high byte) or odd (thus a low byte). This can be done by the HP 9825 in the following manner: The serial interface is set up for even parity. If the byte read by the interface contains even parity, there will be no parity error. But if the byte read contains odd parity, there will be a parity error and the parity error bit in the status register of the interface will be set, with the flag bit number depending on the specific system. The procedure is then to reset first the status word registers, read a byte, check the status word for a parity error, and when an error is detected, that byte is a low byte because a parity error will occur for a byte with odd parity, which is a low byte. Therefore, the next byte in the stream must be a high byte by definition, since the stream is always alternating high byte, low byte, high byte, low byte, etc. When the high byte is detected, it then can be examined for address by looking at the four most significant bits only. If one wanted to start at address zero (the first channel), then one would follow the following procedure: After a low byte has been detected by the parity-error method given above, read in the next two bytes. These will be a high byte/low byte pair since the high byte is sent first in the set. Examine the address found in the high byte by equating the high byte to a dummy variable, shifting all four bits of that variable right by four places (or anding it with 240). leaving only the address bits in the lowest locations of the variable (or in the highest places of the variable in the case of an anding operation). The dummy variable can then be checked to see if the address is the desired one. If not two more bytes are read in and the address checked following the procedure given above until the correct address is located. Then the data can be read into the computer array for processing. Several specific programs are given in this report that use the above procedures. The program Latch, described in Section II and the FORTRAN listing of that routine given in Program Listing II-1, is a very good example utilizing these procedures.

The final part of this field data collection system is the digital tape recorder. The particular recorder used in this study is a National Multiplex Digital Tape Recorder Model number CC-9, which cost less than $200.00 and uses low-cost digital or good-quality analog cassette tapes. The recorder comes complete with RS-232 interface as well as TTL compatibility. It can operate up to 9600 baud by changing the tape speed to keep the tape density less than 1600 bits per inch (1200 bits per inch is a good, reliable speed for most reasonable cassette tapes). The recorded output from this tape recorder can be directly interfaced with a serial (RS-232) port of any computer, completing the recording/processing sequence.

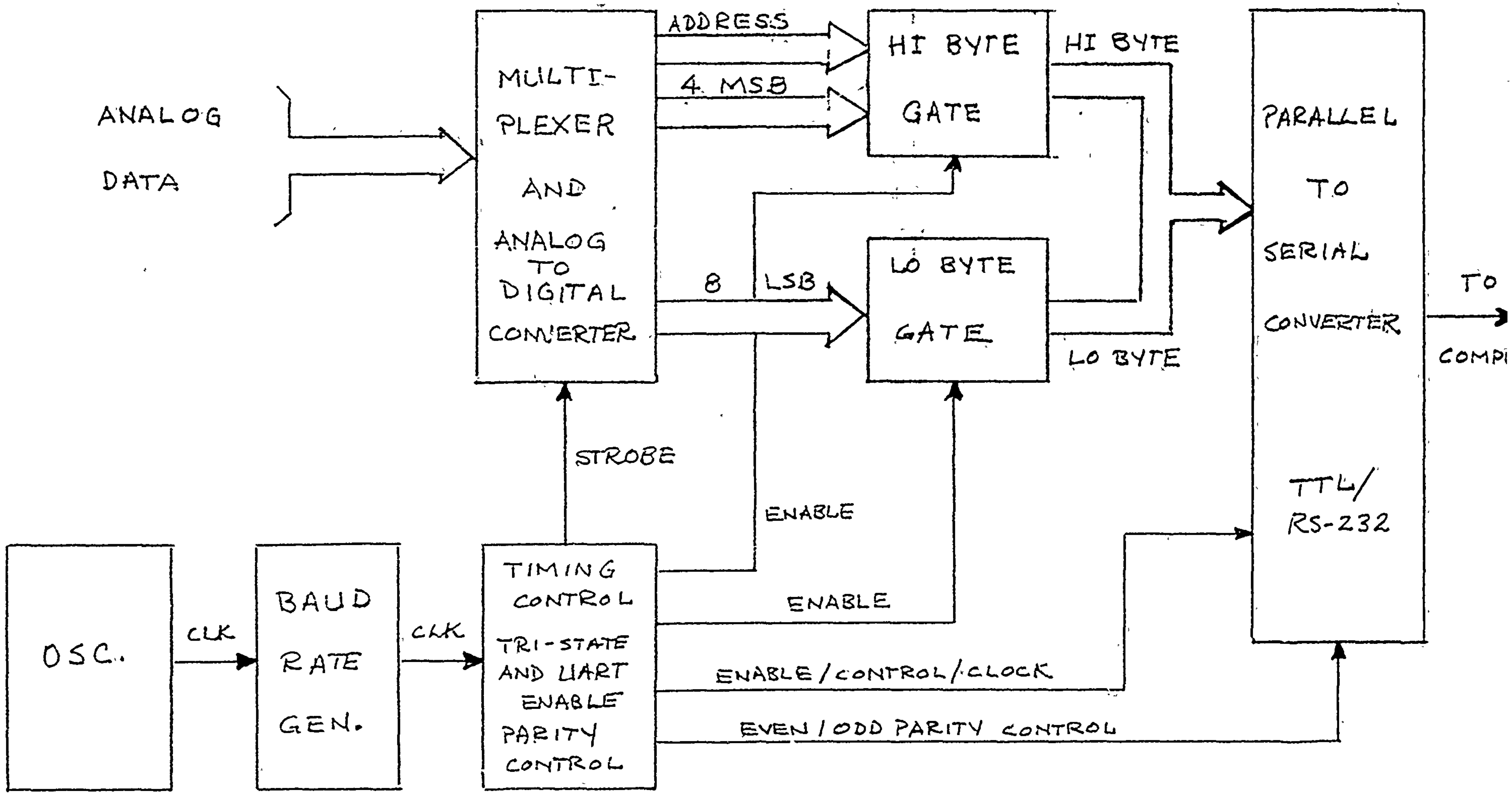


FIGURE V-2. INFORMATION TRANSMISSION SYSTEM, SEQUENTIAL SCANNING MODE -- FUNCTIONAL BLOCK DIAGRAM

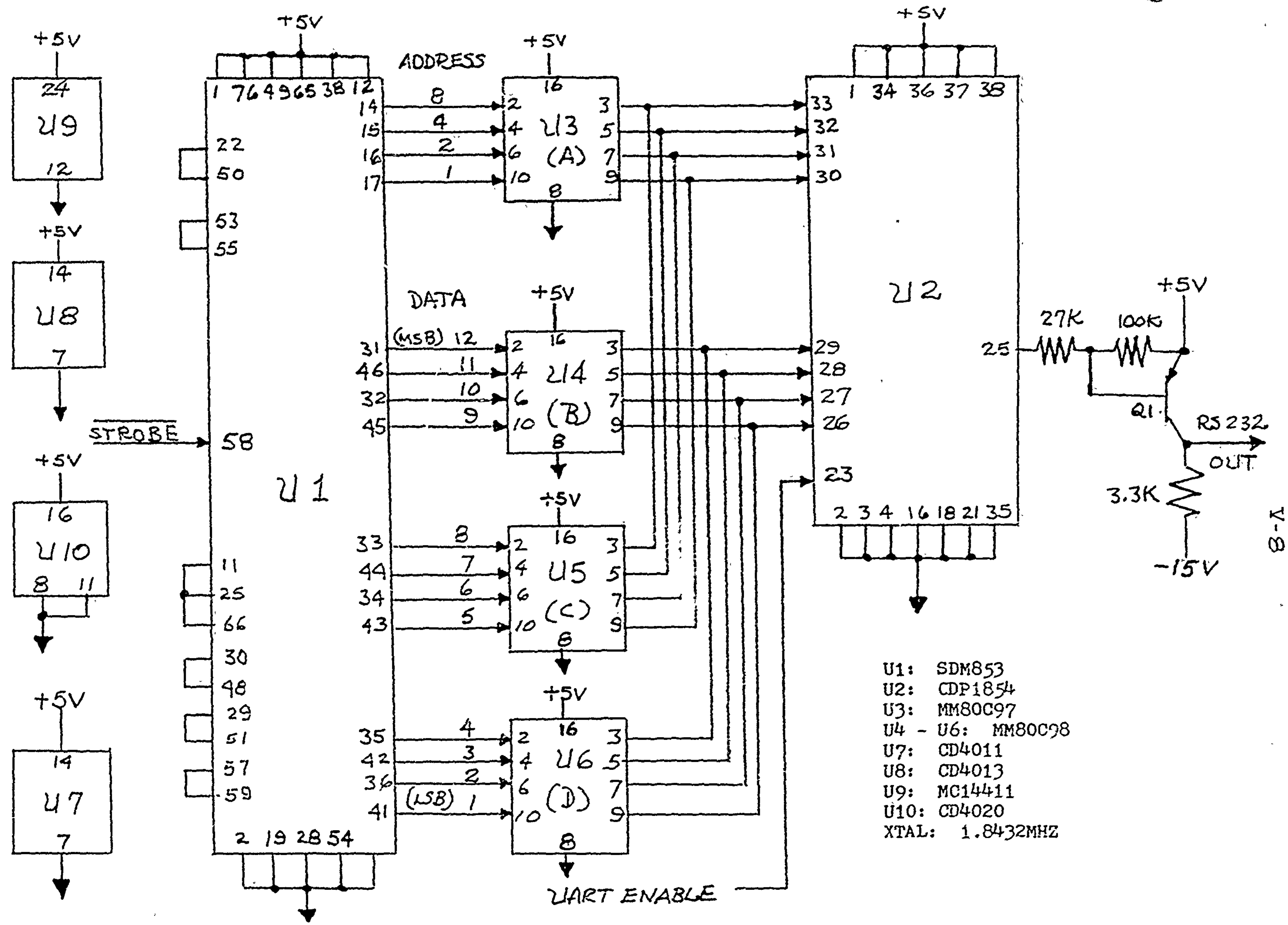


FIGURE V-3. INFORMATION TRANSMISSION SYSTEM - SEQUENTIALLY-SCANNING MODE - SIGNAL AND POWER DIAGRAM

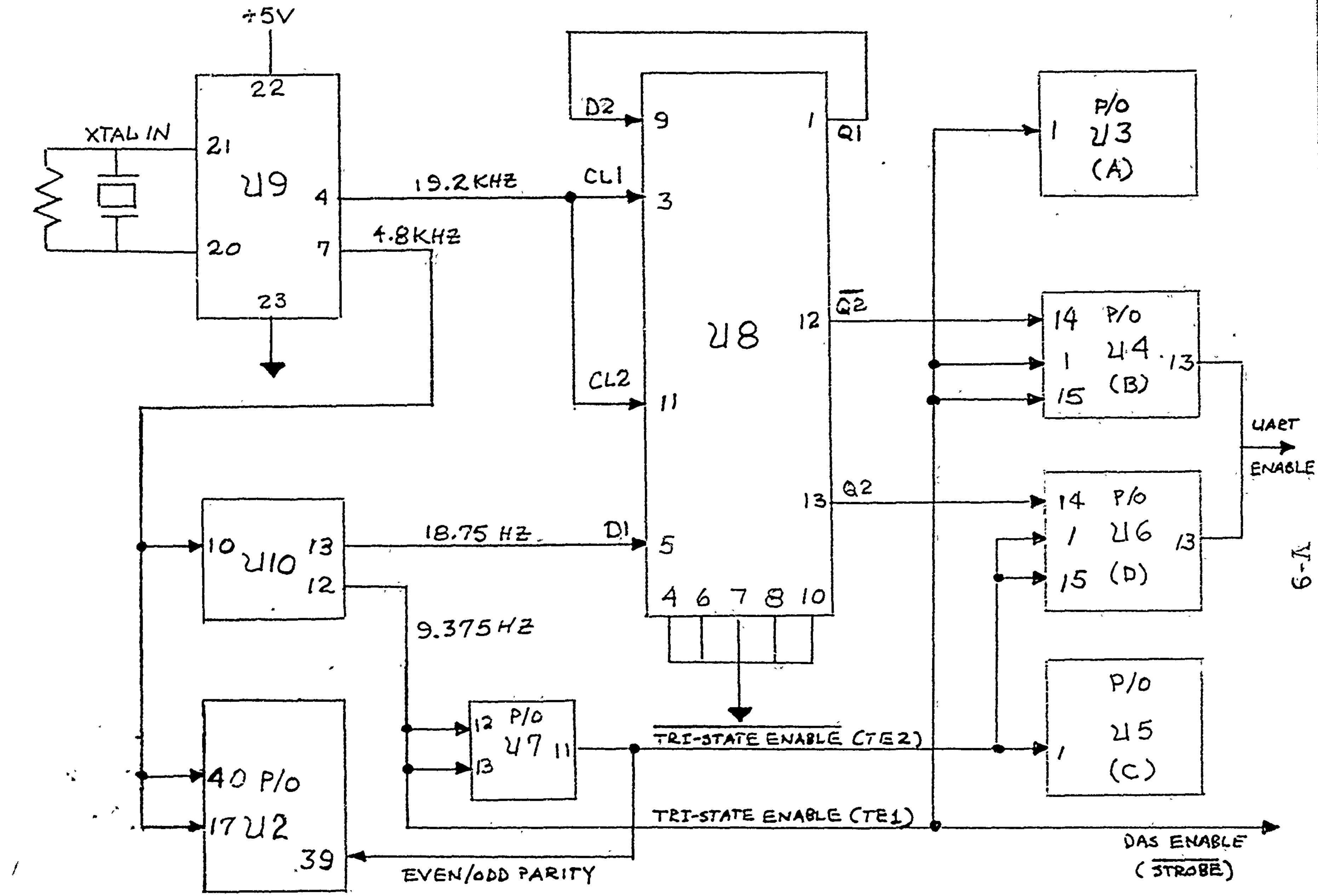


Figure V-4. Information Transmission System - Sequential Scanning Mode - Timing/Control Diagram

SECTION VI
FIELD TESTS AND PROCESSING RESULTS

Part I. NCSC Tests

This first part gives discussions of the deployment, operation, and retrieval of the data buoy in the Gulf of Mexico, the description of the test conditions from which data were collected, and a detailing of the method of conducting data tests. Also included in this section is a discussion of the method of making direct ground-truth measurements to support the tests. The last part of this section presents the results of the data processing, in which the results of the two prediction models are compared with one another and with the ground-truth measurements and the observed conditions by other means during data collection. The specific computer output-line print-outs are given in Appendix A for the simplified model and in Appendix B for the detailed model.* The reader is referred to these two appendices for actual values and specific details of the processing results.* Included in these two appendices are 72 individual test runs, giving over two thousand measurements and calculations of wave heights, periods, direction, predicted breaker heights, and various other associated processing statistics.*

Buoy Operation. Prior to the buoy deployment, the entire system was checked out for proper operation and calibration. The tide gauge was calibrated at NCSC in their test pool down to depths almost as deep as the deployment depth. These results indicated a very linear calibration. Water circulation tests were made for the current meter as well as other in-water testing of the wave sensor. The temperature sensors were compared with laboratory standards. Complete calibration tests were performed on the ITS system. (These tests have been previously discussed.) Prior to the deployment of the buoy, a final check-out of the system was done with the HP-9825 system, testing as a unit all parts of the system. Any results that appeared not satisfactory were corrected, and the system was again retested until all results were satisfactory.

The buoy was deployed by NCSC divers under supervision of the NCSC program task leader and with the technical assistance of Dynex personnel at the beach tower to check and verify proper operation. The buoy was

*Appendices A and B are contained in Interim Report FR-XX-1.

deployed in approximately six meters of water about 300 meters offshore. This location put the data buoy outside of the outer sand bar in open-water conditions. The sensor probes were approximately one meter off of the bottom. The buoy was oriented such that the positive Y-axis of the current meter was directly pointing onshore and the positive X-axis of the meter was pointing directly long-shore to the right as viewed from the shore facing the buoy. The shore at this location is aligned approximately Northwest to Southeast. Thus, the onshore direction is Northeast (positive Y-axis) and the longshore direction is Northwest (positive X-axis). Two vertical pipes jetted into the bottom were used to secure the buoy in place.

The buoy was first deployed on 10/1/79 in the Gulf of Mexico directly off of NCSC Beach Tower Number One. The deployment was made without any undue complications. The data was immediately checked upon deployment and indicated completely satisfactory operation. A data tape was made on site at this time in addition to many other tests and checkouts. After the buoy deployment, the performance of the buoy was checked usually at least once a day and sometimes several times a day. It was realized that the cable that was used for providing power to the buoy and transmitting data back to the beach site was not a long term type of cable. It was available to the program at no cost, and the original objective was to deploy and test the buoy system. This could be done over a short period of time. At that point a more dependable cabling system could be used if further testing was desirable.

On 10/ 7/79 it was determined at the Dynex terminal that there were apparent problems with the buoy system. Although data was being received previously, at this point there was no data on the telephone line. Early the next day a visual inspection of the site indicated that the cable had parted at the splice near the water's edge. At this point, the cable was pulled, with the buoy left in place. After the cable had been repaired, it was reinstalled on 10/12/79. The system was again checked out and found to be totally operational. However, this cable was not able to with-stand the strong waves and currents in the near-shore zone and problems were encountered again -- this time causing a failure within the buoy output electronics. The output transistor attempted to drive a highly increased load when the cable failed. This output circuit was redesigned such that this type of cable failure would not cause a failure within the buoy.

At this point a different cable was obtained from another program area that was interested in the buoy system. This new cable has been demonstrated to be able to with-stand considerable forces in the surf and near-surf zones. On 11/28/79 the buoy system was re-installed with the new cable. At this point the bottom mounting system was changed from screw anchors to the vertical pipes jetted into the bottom as mentioned earlier. As before, a complete checkout of the system was made at deployment, and a test data tape was recorded. The results of this new deployment appear very good; however, as of the time of this report, the present status of the cable is believed to be operational, but not definitely ascertained. On 12/12/79 the buoy was pulled for inspection. It has not been redeployed as of this time.

During the time of deployment, considerable testing and data collection were performed. The results of this testing period indicate that this buoy system is very reliable (excluding cabling problems) and produces in detail all of the required parameters as designed.

Test Conditions. Six separate data runs were made that have been processed to test the prediction models. These data runs are in addition to the other data collected for the specific purpose of testing the buoy. These conditions are summarized in Table V-1. The test runs made during the buoy and/or cable installation all show very calm conditions. This is because it is desirable to install the system during relatively calm conditions. Two of the test dates occurred during moderate to rough conditions. A strong system was propagating large amounts of wave energy from the Gulf into the test area on the sixth and seventh of December. It was on these two days that the best conditions existed and that on these two days the most detailed measurements were made of the breaker heights to be compared with the prediction models. The breakers were well defined, being long-crested and having long periods. During the taping of data from the buoy system, actual measurements were made of the breaker heights using a technique given in the following part. This information was used for the detailed analysis given in the last part of this section.

Table VI-1. Test Conditions Measured and Observed during Buoy Data Collection Periods

DATE	RUN NO.	WIND (ST II)	WAVES(STII)(M)	BREAKER HT (M)	OTHER OBSERVATIONS
10/1/79	100-1,-6	2-9kts/WNW	Not avail.	.3 to .4	local wind: 5-10kts SSW/W; very little current
10/2/79	2-1 to 3-6	14-17kts/West	Not avail.	.8 to 1.3	
10/12/79	4-1 to 4-6	7-12kts/East	.3 to .5	.3 to .6	Waves breaking almost at shore; very little local wind; waves from SW, 5-6 sec
12/6/79	6-1 to 6-6	2-9kts/West	1.1 to 1.3	up to 2	large, well-formed breakers, many plunging beyond inner bar; swell from distant storm
12/7/79	7-1 to 7-6	5-9kts/NNW	.7 to .8	up to 1.3	remains of previous day storm; decaying with time; still well-defined breakers; some plunging, mostly spilling

NOTE: ST II is STAGE II, which is located 1.5 N. Mi. offshore in 18 meters of water

<u>Method of Ground Truth Measurements</u>. One of the major objectives of this study is the evaluation of the prediction models. Therefore, it is necessary to have a method for this evaluation. What is desired is an actual measurement of the breaker height on an individual basis to compare directly with the predicted height as given from the measured offshore parameters and the prediction model. This necessitates being able to time-correlate buoy measurements that are recorded directly with breaker measurements that are observed. The major problem with this is that the waves are still somewhat dispersive even though the water is shallow. For the shorter wave lengths, their propagation is not completely controlled by the depth as much as the longer waves which are almost shallow-water waves. Therefore, the sequence of the waves at the buoy is not necessarily the sequence that they arrive in at the breaker zone. Some may not even make it to this observation region. However, in general, the measurements of a group of waves can be compared with that for time-delayed measurements at the breaker region for a similar group of breakers.

Using the above approach, a technique was developed using a transit with vertical graduations. This vertical scale was calibrated for two known distances that corresponded to two locations in the breaker region where two surface floats had been deployed. The float was selected that was nearest to the wave breaking. The vertical displacement of the float was measured using the calibrated transit scale. Because of the height of the tower (over ten meters at the transit site), the float motion could be followed very accurately, giving an accurate measurement of the wave breaker just before breaking. A time series of measurements were taken during data recording from the data buoy with real time being noted on the list of breaker heights tabulated at periodic intervals.

On 12/6/79 the time series observations were not tabulated in detail, but statistics concerning the observed breaker measurements were which included average breaker heights, typical breaker heights, average of the largest breaker heights, and maximum observed breaker height during the entire interval of data recording from the buoy. On 12/7/79 the entire time series of breaker observations were tabulated on an individual wave-by-wave basis along with time information.

Test Results. The results of the data processing for the two prediction models are given in the two appendices with Appendix A for the Simplified Model and Appendix B for the Detailed Model. Included in the detailed processing was a direction histogramming of the current motion due to the surface waves. This is given at the first of each data run where the number of counts is given for each ten-degree bin. The counts do not correspond to the number of waves but to the number of observations of current. Wave level was sampled eight times each cycle of the ITS system, but current was sampled only two times for each component each cycle, which gave a current sampling for both components approximately once every second. No attempt was made to obtain the average current direction over the individual wave period. This effect is proposed for future study. The current (wave) direction given in the listings has the following orientation: ninety degrees is onshore directly. Zero degrees is parallel to the shore (longshore) and to the right as seen from shore looking towards the buoy. Using the buoy orientation, ninety degrees (onshore) is actually approximately Northeast in actual coordinates and zero degrees is approximately Northwest.

The format of the two sets of listings is similar. For the simplified model, the peak-to-trough wave height given is that measured at the buoy one meter off of the bottom. It is not the surface wave height. The last part given for each run is the extrapolated surface height from the bottom measurement using linear wave theory as mentioned earlier. Also given are the measured wave periods for the run, and the predicted wave breaker heights computed from the model are included. The results for both model processing do not include any data for wave periods less than three seconds. It was found for these shorter wave periods that the linear theory gave not only surface wave heights at the buoy but also breaker heights often very unrealistic. Thus it was decided to eliminate this variation from the results and to concentrate the analysis on the wave periods for which it was felt that reliable data were obtained.

It should be noted for both sets of results given that on the listings the first wave height corresponds to the first wave period which corresponds to the first predicted breaker height and that the second to the second, etc. Also, it should be noted that the first set of numbers for the simplified model corresponds exactly to the first set of numbers for the detailed model on a one-for-one basis. Data for one run for the simplified model can be compared directly with data for the same run number for the detailed model. Note however, that the run number includes two numbers, e. g., Run Number 7-3.

For each set of run numbers the average and RMS values are given. This allows further statistical comparisons to be made. Each set of data runs are the same length in time: three minutes and thirty-six seconds. For each data set there are six runs indicated by the dash numbers one through six, e. g., 7-1 through 7-6. This gives a total data period for that test of 21.6 minutes.

On 10/2/79 two test tapes were recorded -- the first one starting about 0830 and the second one starting about 1130. These two times overlapped the time period when direct observations were being made of the breaker conditions by NCSC personnel. Attempts were made to measure breaker heights using a person in the surf region holding a marker stick. Because of the rough surf, this did not prove feasible. However, some measurements were made. Typical breaker heights were running from about .6 to less than one meter. Largest heights were running about one and a quarter meters. The results of the data processing for runs 2-1 through 3-6 show significant differences for the two models. The average breaker height for the detailed model is approximately .7 meters, whereas for the simplified model the average height is .5 meters. In addition, the largest breaker height predicted from each of the 12 runs on this test date was averaged with the largest from the other runs on this date. For the detailed model this average of the largest is 1.22 meters, whereas for the simplified model the average is 1.0 meters. For the conditions observed, although not in specific detail, the typical height brackets the average height for the detailed model (.7 meters) but exceeds that of the simplified model (.45 meters). The largest heights observed closely match the above averages of the largest wave from each run for the detailed model (1.22 meters), but once again, the observed heights exceed that given by the simplified model (1.0 meters). On a percentage basis the simplified model underpredicts the detailed model for this data about twice as much for the average of all waves as it does for the average of the largest waves.

This indicates for this test condition, the simplified model does better for the larger waves than for the smaller, average waves. However, the detailed model predicts well for all waves according to the limited observations on this date.

On 12/6/79 measurements of the wave breaker heights were made using the transit and the technique discussed in an earlier part of this section. The detailed time series were not taken, but typical measurements gave the following results. Almost all of the breakers were 3/4 of a meter or larger with the typical breaker about one meter. Many breakers were somewhat larger with some between 1 to 1.5 meters. None were observed to exceed 2 meters although an occasional breaker approached 1.8 meters. From the processed data, the detailed model shows that the average of the largest breaker from each set is 1.76 meters with the corresponding height for the simplified model being 1.55 meters. The average of all breaker heights for the detailed model is just over one meter (1.07) which is only slightly larger than that for the simplified model (.99). For these tests, within the accuracies of the observations, both models appear to give reasonable answers, but once agian, the simplified model gives somewhat smaller values than the detailed model. However, in this case, the simplified model may have predicted more accurately than the detailed model. The observation to be made is that for the longer wave periods, the simplified model does better than for the shorter wave periods. The average wave period for this test date is one to one and a half seconds longer than for the previous test date. This same observation can also be made for the following test date results.

On 12/7/79 the best breaker measurements were made for the data that were taken in this study. Although one could still not make one-for-one comparisons between the predicted breaker heights and the measured breaker heights, there were far more statistics for comparison. These results are summarized in Table V-2 where the average, of all breaker heights, the average of the highest one-third breaker heights, and the average of the highest one-tenth breaker heights are given for the two models and for the directly measured with the transit. These values are given for each of the six runs on that date as well as the average for all of the runs. The direct measurements are only given for the first and last three runs averaged together.

TABLE VI-2. Comparison of Measured Versus Predicted Breaker Heights for Simplified and Detailed Models
Test Date: 12/7/79

TIME	AVERAGE BREAKER HEIGHT (M)			AVERAGE HIGHEST 1/3 (M)			AVERAGE HIGHEST 1/10 (M)		
	Simpl. Model	Detail Model	Meas'd	Simpl. Model	Detail Model	Meas'd	Simpl. Model	Detail Model	Meas'd
11:15-11:18	.69	.83		.98	1.19		1.14	1.29	
11:18-11:22	.70	.81		.96	1.06		1.09	1.22	
11:22-11:26	.76	.85		1.01	1.04		1.36	1.37	
AVERAGE 11:15-11:26	.72	.82	.73	.98	1.08	1.07	1.19	1.29	1.13
11:28-11:31	.69	.83		1.02	1.07		1.24	1.25	
11:31-11:35	.79	.86		1.12	1.13		1.23	1.23	
11:35-11:39	.75	.84		1.01	1.02		1.12	1.14	
AVERAGE 11:28-11:39	.74	.84	.76	1.05	1.07	.98	1.20	1.21	1.08
AVERAGE ALL TIMES 11:15-11:39	.73	.83	.75	1.02	1.08	1.03	1.20	1.25	1.10

It can be observed that the detailed model over-predicts the heights in all cases, whereas the simplified model comes closer to the measured conditions, in some cases under-predicting, but in others over-predicting. However, on the average, the simplified model is very close except for the largest of the heights. For this test date the average wave periods are running close to 6 seconds. It should be noted that periods of less than 3 seconds are not included. Refer to the previous discussion regarding this cut-off.

A brief examination of the effect of beach slope on the results given for the detailed model was done. A slope of 1:20 was used for the data processing. At this point, this value is felt to be too large. 1:30 to 1:50 probably fits the conditions more closely. A bottom survey needs to be done to determine this. Using the more gradual slope in the detailed model, the prediction results for the cases tested was less, but not by as much as the results for this test date indicate. The slope, as well as other factors, need to be further investigated for model refinement.

It should be noted at this point that the results of the comparisons of the models with observations are very encouraging. The results indicate that for breaker heights up to two meters, the detailed model predicts breaker heights to an accuracy of the order of ten percent. For the higher energy waves, the simplified model does almost as well, but does less well for the shorter wave periods with lower amplitudes.

Part II. CERC FRF Duck-N. C. Tests

It was decided to perform wave measurements using the Coastal Engineering Research Center (CERC) Field Research Facility (FRF) at Duck, North Carolina, to obtain wave and breaker data for further testing and validating the prediction models. At this facility there are seven surface wave gauges (Baylor gauges) located on a pier that extends out into water depth over eight meters deep. These gauges are located such that they are able to make measurements of the modifications during the shoaling process as the wave progresses into shore and eventually breaks. In addition this facility is located on the Atlantic Ocean coast where, on occasion, large storms produce significant near-shore waves and surf.

The details of the wave gauge locations and water depths are given in Table VI-3.

STATION LOCATION (METERS)	WAVE GAUGE NUMBER	WATER DEPTH* (METERS)
591	End of Pier	7.7
579	#1	8.0
427	#2	6.9
323	#3	5.7
274	#4	5.4
239	#5	4.2
213	#6	2.5
189	#7	1.1
118	Shore	0

*Based on MSL and measurements taken by CERC on 4/2/80

The data acquisition system, described in the previous section, was designed for this operation. It was installed the end of March, 1980 and b came operational the first of April. The first seven channels recorded the seven wave gauges in the order given in the above table. The next three channels are for two components of current and a pressure-transducer wave gauge. The last channel is for calibration voltage (a known standard).

The operation of the collection and recording system is as given in the previous section and is described briefly here. The seven analog signals from the wave gauges on the CERC pier are tied in directly into the first seven channels of the multiplexer. If available, the current meter and pressure transducer wave gauge are tied into the next three channels. (This latter information can be used to evaluate measurements on the bottom as the source information as would be the case of a buoy operation.) On the last channel a mercury battery is tied in with its voltage measured precisely. The shelf life is so long and the drain is so small that stabilities in this voltage of one millivolt or less over periods of one month or more are quite common.

During the collection times, a cassette tape is put into the recorder and is turned on at the selected time. At the end of the recording cycle, the tape recorder automatically shuts off. The tape is then turned over for the next recording time. Power can be left on the system or it can be shut down.

The data tapes were checked by Dynex to ascertain that there was wave data on the tapes that appeared to be reasonable. The calibration voltage was also checked to further verify reliable operation. Since the processing of the data from this location was considerably different than that at NCSC, software routines had to be developed for this. Therefore, it was some time before detailed in-depth processing was done on any of the data. However, all tapes were given a cursory testing to discover obvious malfunctioning of the system. Such was found in some of the data in that only channel address and no data were being received for a period of time. The problem was traced to a switch setting that had been inadvertently changed. Other than this all data appeared normal. Wave fluctuations could be detected on all operating channels, and the amplitudes and periods also appeared normal.

One objective for the data collected from these multiple gauges was to be able to follow a single wave from the seaward gauge all the way in to the last gauge prior to breaking. In this manner, testing of the prediction models could be done on a deterministic basis -- one that had not been done on the previous data. It was done on a statistical basis as given in the previous part of this section. Methods were investigated for doing this with the data processing equipment but without satisfactory results. A major problem was the large spacing between wave gauges, particularly those seaward. In general, there were many wave periods between gauges, and a few seconds error, which was only a small percent of the propagation time for the longer distances, could allow losing track of a particular wave and locking on to an adjacent wave unknowingly.

A method was devised that worked well with the data. A time history of the wave records for all seven channels was displayed on the HP-9845 CRT in a strip-chart manner for the digital data. The data were compressed such that a relatively long time period could be displayed, but were not so compressed that wave heights and periods could not be easily and accurately read. The resultant time-history displays were dumped onto the hard-copy printer for a permanent record. For the most part, a wave train could be readily followed as it progressed from one wave gauge to the next -- particularly for the larger, more prominent waves with longer wave periods. However, there still was the possibility of ambiguity in many cases. In order to minimize this, another program was written that computed the propagation times of waves with various periods from one gauge to the next using bottom data taken by CERC personnel near the time of wave measurements. This utility program, in almost all cases of interest, allowed the data analyst to determine wave progression more readily. For most applications, the utility program was used for verifying that the analyst had selected the correct wave rather than the results being used in the selection. In almost all instances, wave verification was made within a small fraction of a wave period.

As the data were being analyzed from a large storm that occurred in October, 1980, it was discovered that wave progression to the next gauge could not at all be followed -- often the results appeared random rather than orderly, as had been observed in the first set of data and in many other independent observation. The reason for this was not immediately obvious. Upon close examination of the data, it was found that there was a large resemblance, in terms of amplitude and particularly phase (in-phase), between the data on two pairs of the channels (one and two, and four and five). The reason that this was not more obvious was that the gross structure was very similar but the fine detail was enough different as to mask the effect to the eye until close examination was made. It appeared that there was leakage of one channel strongly onto the next for the pairs given above. If this were the case, using this data would or could lead to erroneous results. Since this was some of the best data collected, it was felt that statistically verifying these findings was warranted. To this end, the cross-spectral programs described in Section II were written for this study by modifying routines developed for other programs. The coherence and phase of the two suspected pairs of channels were checked with the suspected results verified. Coherencies approaching unity and phases approaching zero (in phase) resulted for all of the waves up to periods approaching one second. A further test was done on the data felt to be reliable and not having the above problems. Once again, the results were as suspected. As the distances between wave gauges increased, the coherences rapidly decreased, and in no cases did the coherencies approach unity as strongly as the suspected data. Also, the phase spectra displayed as a function of frequency the typical phase progression associated with dispersive waves. Therefore, the suspected data were conclusively shown to be not useable. However, the earlier data taken by Dynex personnel when the system was installed at Duck, N. C., were shown to be reliable, and those data were used in the processing and model verification given in this part of this section.

The data from the April 1, 1980 test were manually read from the strip chart display print-outs as given above. The best seaward gauge was selected for the input information into the prediction model. For each seaward height read, the same wave was read in height at each of the following gauges until it no longer appeared (breaking occurred). For almost all waves used in this part of the study from these data, breaking occurred between gauges #6 and #7. (in a vast majority of the cases the prediction model gave breaker depths that were between these two gauges, further supporting the model.) In addition to the wave height measurements, each wave period was measured. This information was put into the modified manual input prediction program (described in Section II), and the results for each wave input were given and recorded along with the input information. The results of this were then tabulated and examined statistically and are summarized in Table VI-4. The print-outs of the strip-chart display for this test date are given in Figure VI-1.

Table VI-4. Statistics of Measured and Predicted Breaker Heights
CERC Field Research Facility, Duck, N. C., 4/1/80

Range of off-shore wave heights	Range of measured breaker heights	Range of predicted breaker heights	Average of measured minus predicted heights	RMS of measured minus predicted heights	Range of predicted breaker depths	
1.9 - 4.7	2.8 - 7.0	3.3 - 7.4	0.007	0.63	3.1 - 6.9	(Feet)
0.6 - 1.4	0.9 - 2.1	1.0 - 2.2	0.002	0.19	1.0 - 2.1	(Meters)

As the above table indicates, the detailed prediction model very accurately predicts the breaker heights compared statistically with the actual measurements. For the conditions in the above table, the average breaker heights measured was 1.46 meters and the average of the prediction error (measured breaker height minus predicted breaker height) was only 0.002 meters, giving a percent error of only 0.1%. Although the average error is almost zero, the predicted results are probraly not as good as the above would indicate. All of the wave breaking appeared to take place between the last two inshore wave gauges. All of the measured reading were taken from wave gauge #6, the last gauge prior to breaking. However, wave modification undoubtedly continued until the point of breaking. Thus, in reality, the measured values at wave gauge #6 are low compared to the actual breaker height at the location of breaking. This would also make the predicted values low by this difference, since these values predict very closely the heights a short distance prior to breaking. No quantitative numbers are offered in this study as to how large this difference might be. That is not within the scope of this study.

The above discussion points out one of the limitation of this type of verification measurement. For an accurate measurement of the breaker height, the break must break just after passing a wave gauge, i. e., be at the point of breaking when passing the gauge. This will not occur very often. If one requires the wave to break within one or two meters of the gauge, statistically, all else being equal, this will occur between five to ten percent of the time.

As the breaking moves further shoreward of the gauge, the measurement error increases. Thus, it would seem, in order to evaluate the prediction models with more accuracy and reliability than available through the means used in this study, it will be necessary the make the measurements of the wave at the time of breaking by some other means. It may be that the method of measurement used in the NCSC buoy test, using a transit, as described previously, is one of the more accurate methods.

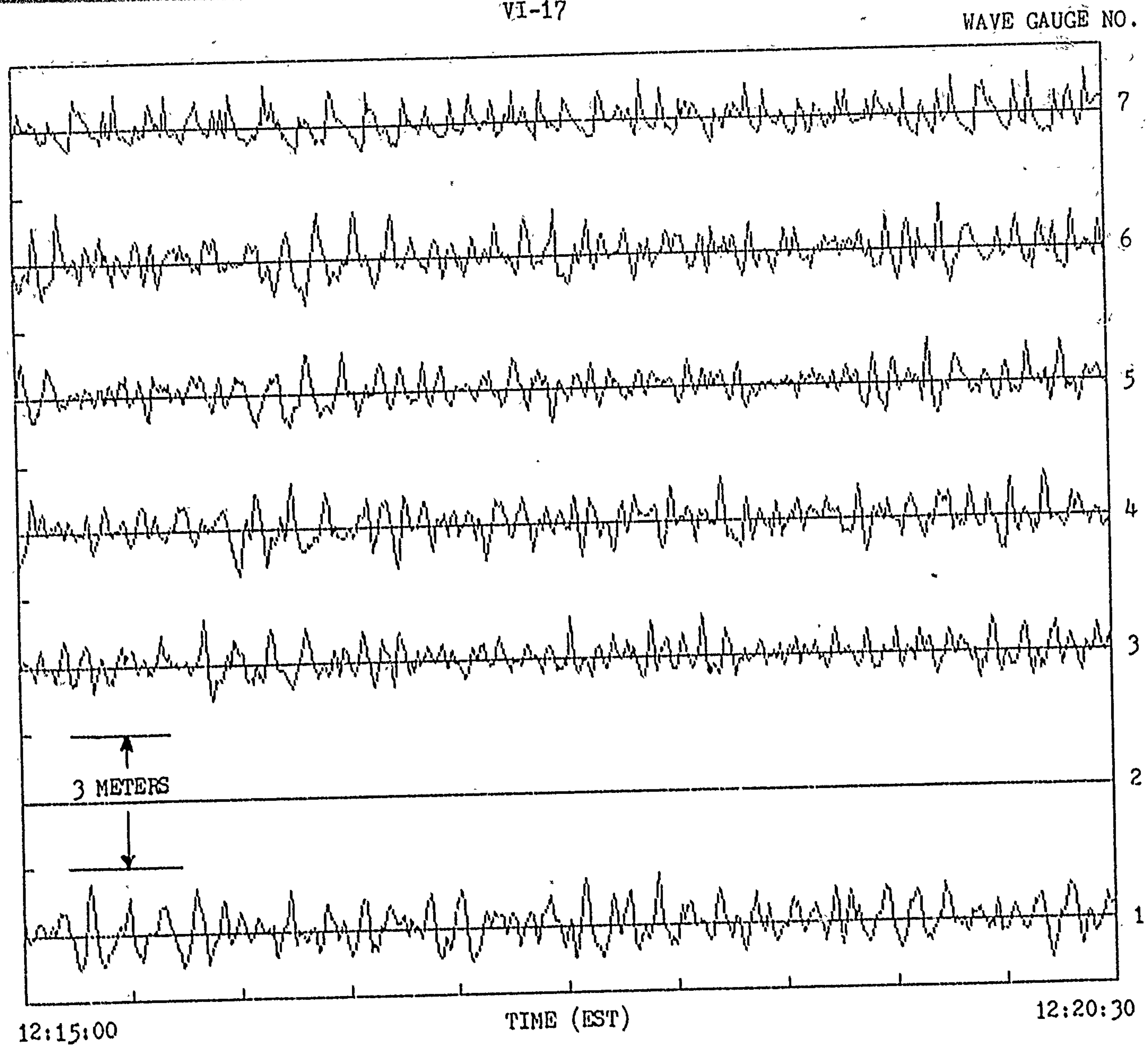


Figure VII-1. Digital Strip Chart Time History of Wave Records from CERC FRF, Duck, North Carolina -- 4/1/80.

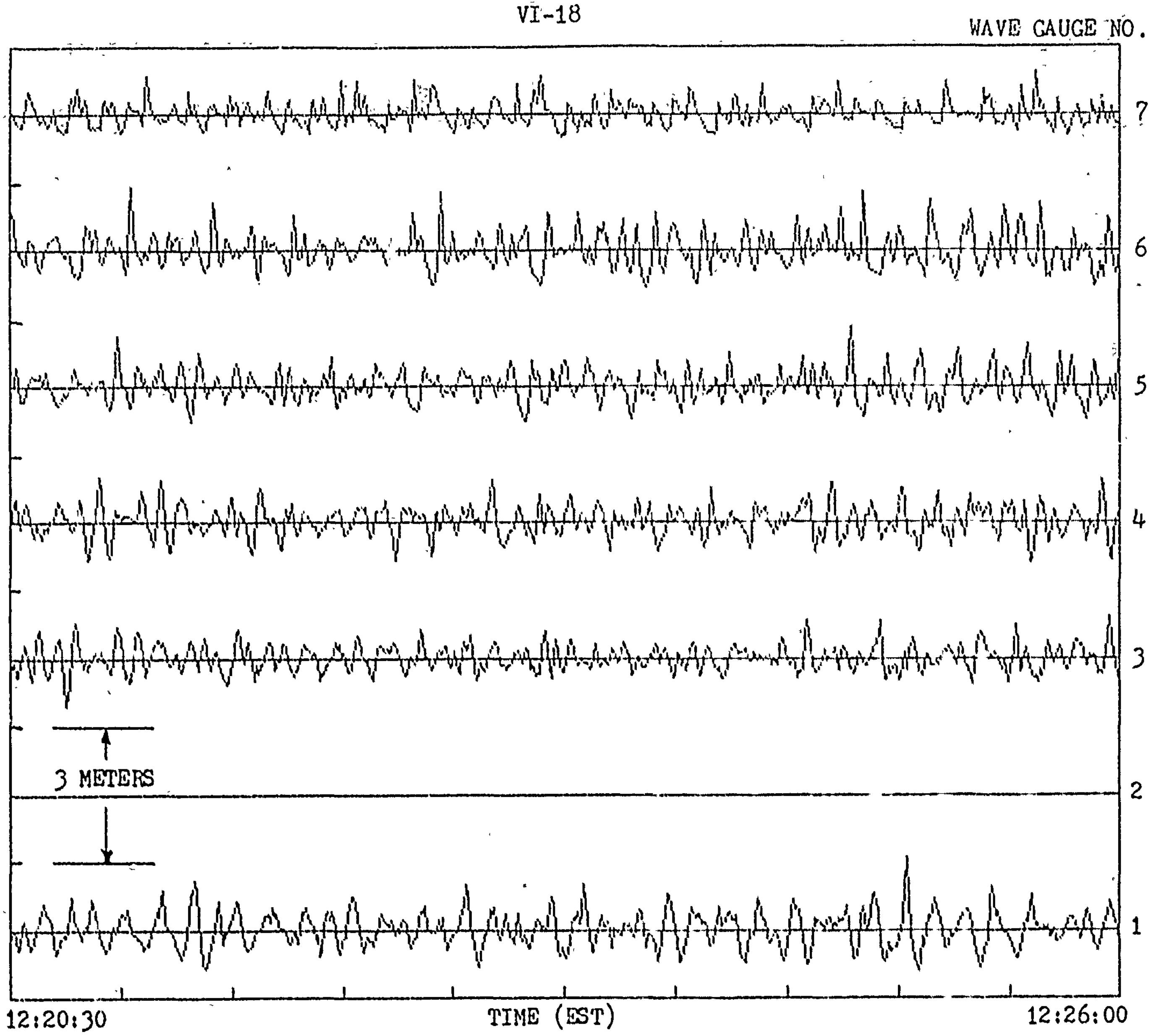


Figure VI-1. Digital Strip Chart Time History of Wave Records from CERC FRF, Duck, North Carolian -- 4/1/80 (Continued)

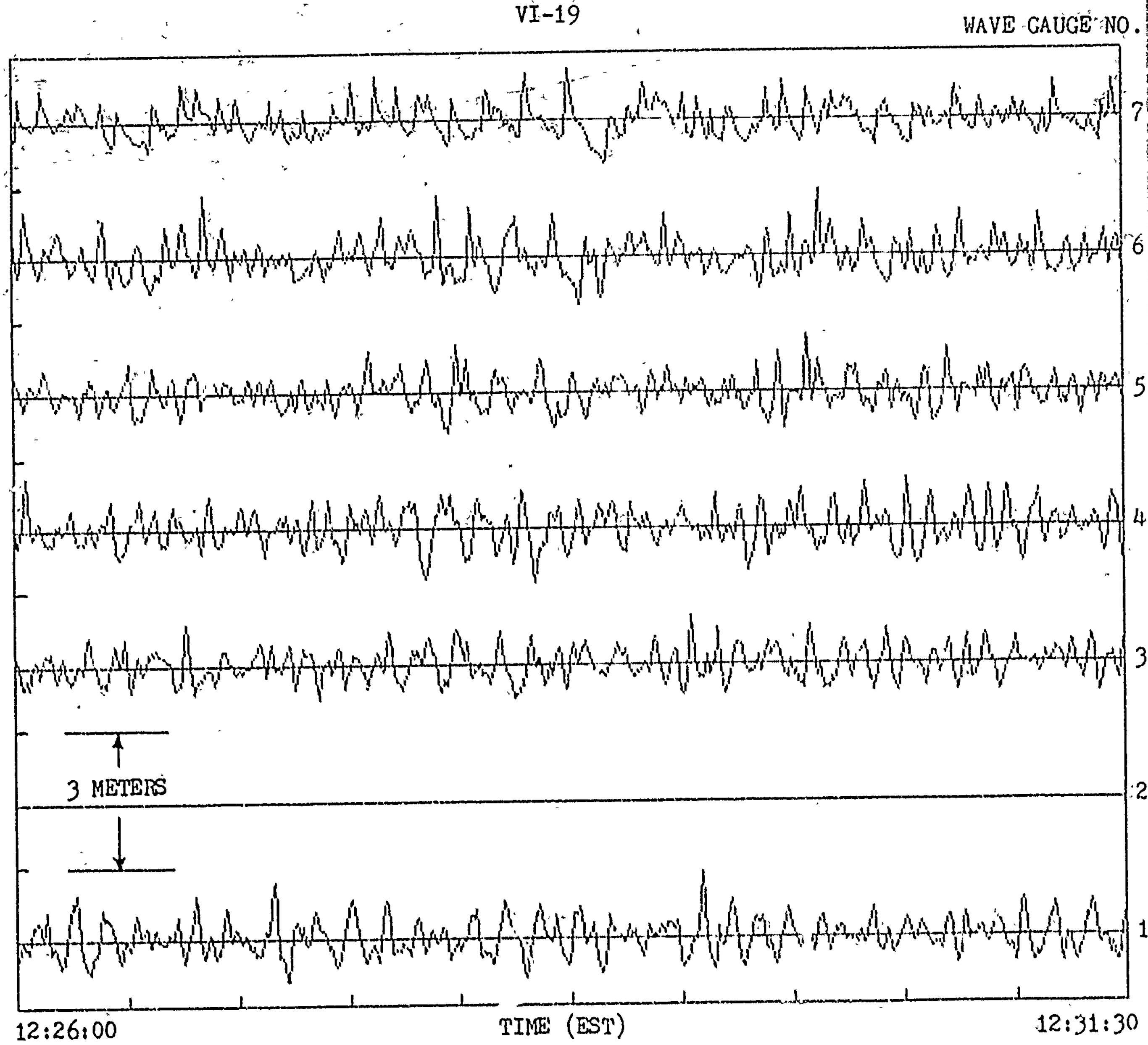


Figure VI-1. Digital Strip Chart Time History of Wave Records from CERC FRF, Duck, North Carolina -- 4/1/80

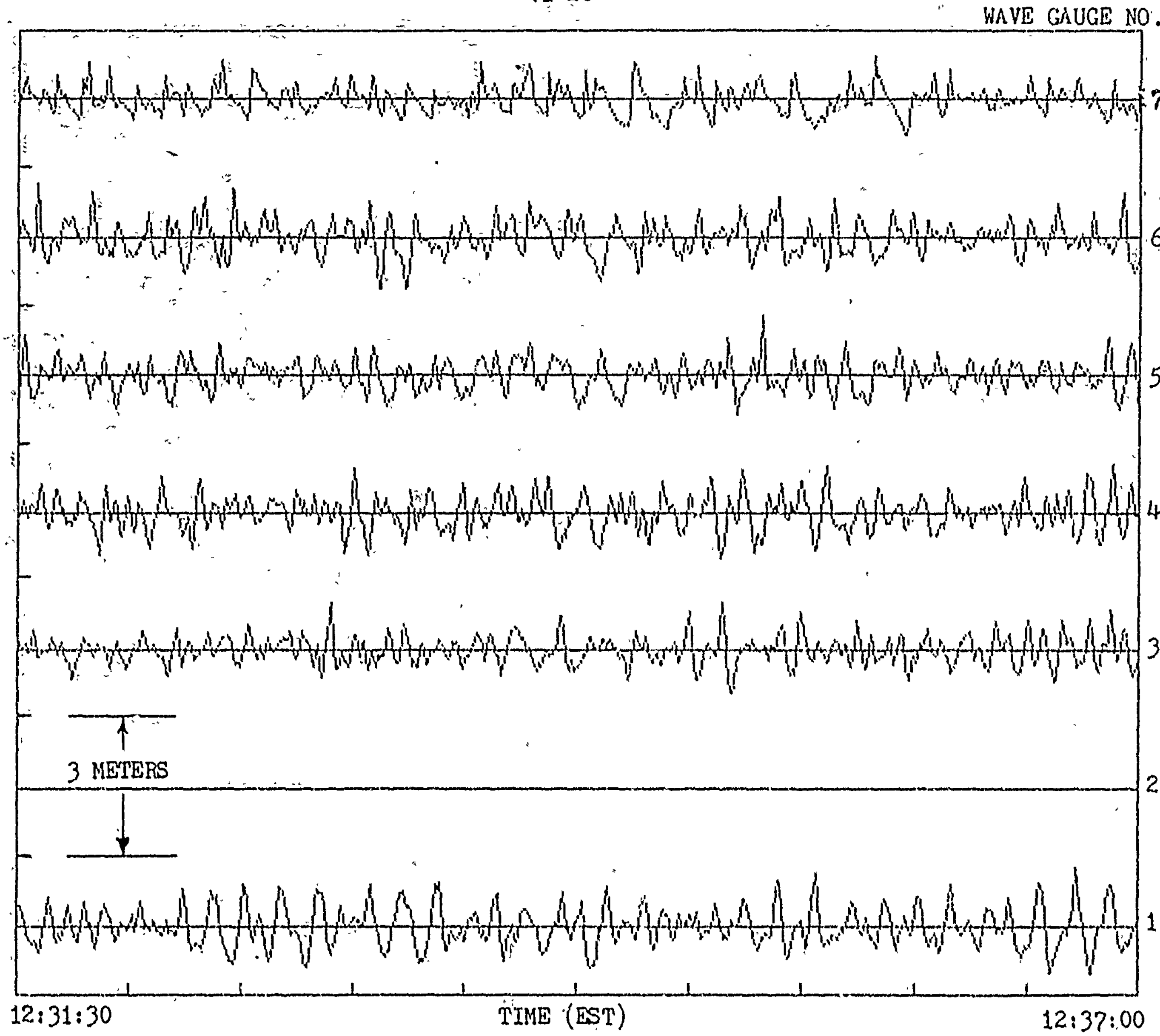


Figure VI-1. Digital Strip Chart Time History of Wave Records from CERC FRF, Duck, North Carolina -- 4/1/80

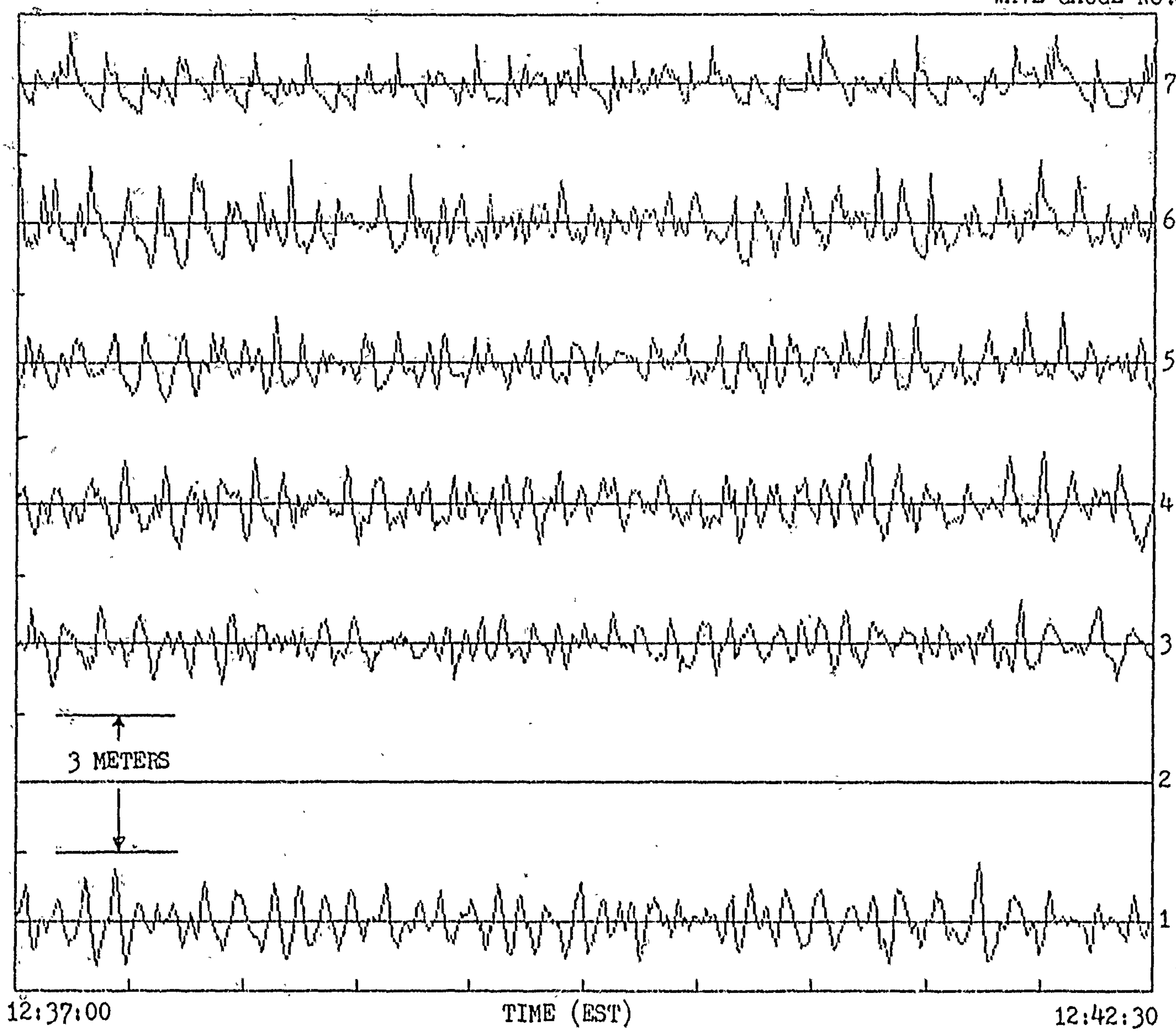


Figure VI-1. Digital Strip Chart Time History of Wave Records from CERC FRF, Duck, North Carolina -- 4/1/80

CONCLUSIONS AND RECOMMENDATIONS

1. Regarding the over-all buoy system and its operation, the results of this study have demonstrated that this system is a very reliable one, has the ability to measure very accurately the required wave, tide and current inputs for data processing, is easily deployable by divers from a small boat with minimum equipment required, and is readily adaptable to a variety of fleet operations and applications. It has demonstrated that it can withstand the hostile environment of the near-surf zone within the limits tested and has the ability to transmit its data via a wide variety of methods to the end user, thus making this buoy system a prime candidate for fleet operations.

2. It is concluded that the software developed for the support operations of the buoy system is efficient in terms of processing capabilities of speed, size, and complexity, can be adapted to on-board buoy processing without considerable modification, is flexible and readily adaptable to a wide range of user applications, and the study results further show the software to be reliable and accurate.

3. It is further concluded that both of the prediction models used in this study give very good results, often with prediction errors of only a few percent, that the detailed model allows for a greater flexibility and range of environmental inputs but does require more complexity in processing, and that the simplified model, although limited, gave accuracies that are well within the requirements of many fleet applications.

4. It is recommended that testing be continued over a larger range of wave and breaker conditions and that a method for more accurately measuring the breaker parameters for ground-truthing the models be explored.

5. Lastly, it is recommended that continued development be done on this buoy system, further investigating the effects of environmental parameters and refinements to the models, and development efforts, leading to the end product of an operational system for the fleet including on-board processing where applicable, be continued.