SACLANTCEN Conference Proceedings No. **25** PART 1

SACLANTCEN

Conference Proceedings No. **25**

PART 1

SESSIONS I to III

SACLANT ASW RESEARCH CENTRE

REAL-TIME GENERAL-PURPOSE, HIGH-SPEED SIGNAL PROCESSING SYSTEMS FOR UNDERWATER RESEARCH

Proceedings of a Working Level Conference held at SACLANTCEN on 18–21 September 1979

Organized by

ROBERT SEYNAEVE

1 DECEMBER 1979

NORTH
ATLANTIC
TREATY
ORGANIZATION

LA SPEZIA, ITALY

Compiled and
Published by

STI
SACLANTCEN

SACLANTCEN
CONFERENCE PROCEEDINGS No. 25

NORTH ATLANTIC TREATY ORGANIZATION

SACLANT ASW Research Centre
Viale San Bartolomeo 400, I-19026 San Bartolomeo (SP), Italy.

tel: national 0187 503540
international + 39 187 503540

telex: 271148 SACENT I

REAL-TIME, GENERAL-PURPOSE, HIGH-SPEED SIGNAL PROCESSING SYSTEMS FOR UNDERWATER RESEARCH

Proceedings of a Working Level Conference held at SACLANTCEN
on 18-21 September 1979

(Part 1: SESSIONS I to III)

Organized by
ROBERT SEYNAEVE

1 December 1979

This document has been prepared from text and illustrations provided by each author. The opinions expressed are those of the authors and are not necessarily those of the SACLANT ASW Research Centre.

LIST OF PARTICIPANTS

CANADA

Mr D.V. Crowe
Defence Research Establishment Atlantic
P.O. Box 1012
Dartmouth
Nova Scotia B2Y 3Z7

DENMARK

Mr B. Damsgaard
Danish Defence Research Establishment
Staunings Plads 2
DK 2100 Copenhagen Ø

Mr. P.B. Ring
Danish Defence Research Establishment
Staunings Plads 2
DK 2100 Copenhagen Ø

FRANCE

Mr J.L. Lambla
GERDSM
D.C.A.N Toulon
83800 Toulon Naval

Mr J.Maigre
GERDSM
D.C.A.N. Toulon
83800 Toulon Naval

GERMANY

Dr J.F. Böhme
FGAN-Forschungsinstitut für Hochfrequenzphysik
Königstrasse 2
D-5307 Wachtberg-Werthhoven

Mr H. Herwig
Forschungsanstalt der Bundeswehr für Wasserschall und Geophysik
Klausdorfer Weg 2-24
2300 Kiel

Ing. Kh. Rosenback
FGAN-Forschungsinstitut für Hochfrequenzphysik
Königstrasse 2
D-5307 Wachtberg-Werthhoven

GERMANY (Cont.)

Mr B. Scholz
Forschungsanstalt der Bundeswehr für Wasserschall und Geophysik
Klausdorfer Weg 2-24
2300 Kiel

Dipl.Ing. H. Urban
Fried. Krupp GMBH Krupp Atlas-Elektronik
Postfach 44 85 45
2800 Bremen 44

Dipl.Ing. W. Wagner
Fried. Krupp GMBH Krupp Atlas-Elektronik
Postfach 44 85 45
2800 Bremen 44

ITALY

Prof V. Cappellini
Istituto di Elettronica
Universita di Firenze
Via di S. Marta 3
50139 Firenze

C.C.(AN) D. Nascetti,IN
Marina Militare
Commissione Permanente per gli Esperimenti del Materiale da Guerra
La Spezia 19026

T.V.(AN) G.C. Testino,IN
Marina Militare
Commissione Permanente per gli Esperimenti del Materiale da Guerra
La Spezia 19026

NETHERLANDS

Mr J.G. Schothorst
Physics Laboratory TNO
Oude Waalsdorperweg 63
Post Box 96864
2509 JG Den Haag

Mr C.H. Vlasblom
Physics Laboratory TNO
Oude Waalsdorperweg 63
Post Box 96864
2509 JG Den Haag

NORWAY

Dr H.J. Alker
Electronics Research Laboratory
O.S. Bragstads Plass 4
N-7034 Trondheim-Nth

Mr K. Hansen
SIMRAD A/S
P.O. Box 111
Horten

Dr. Y.C. Lundh
Norwegian Defence Research Establishment
Postboks 25
N-2007 Kjeller

Dr G.B. Pettersen
Norwegian Defence Research Establishment
Postboks 115
N-3191 Horten

UNITED KINGDOM

Dr J.C. Cook
Admiralty Marine Technology Establishment
Queens Road
Teddington, Middlesex

Dr T.E. Curtis
Admiralty Underwater Weapons Establishment
H.M. Naval Base
Portland, Dorset

Prof J.W.R. Griffiths
Department of Electronic & Electrical Engineering
University of Technology
Loughborough
Leicestershire, LE11 3TU

Mr S.F. Meatcher
Admiralty Underwater Weapons Establishment
H.M. Naval Base
Portland, Dorset

Dr D. Nairn
Admiralty Underwater Weapons Establishment
H.M. Naval Base
Portland, Dorset

Dr J.S. Pyett
Admiralty Underwater Weapons Establishment
H.M. Naval Base
Portland, Dorset

UNITED KINGDOM (Cont.)

Mr C. Richardson
Admiralty Underwater Weapons Establishment
H.M. Naval Base
Portland, Dorset

Mr T. Totten
Royal Aircraft Establishment
Farnborough, Hants

UNITED STATES

Prof A.B. Baggeroer
Massachusetts Institute of Technology
Department of Ocean Engineering
Cambridge
Mass. 02139

Mr J.M. Griffin
Naval Research Laboratory
Code 8160
4555 Overlook Avenue SW
Washington D.C. 20375

Mr E. Hug
Naval Underwater Systems Center
New London, Connecticut 06320

Mr F.W. Molino
Naval Underwater Systems Center
New London, Connecticut 06320

Dr B.L. Pennoyer
Naval Ocean Systems Center
San Diego, California 92152

CAPT C.M. Rigsbee. USN
Assistant Commander for Research & Technology
Naval Air Systems Command
Department of the Navy
Washington D.C. 20360

Mr P.A. Rigsbee
Naval Research Laboratory
Code 7594
Washington D.C. 20375

Mr D. Steiger
Naval Research Laboatory
Code 8003
Washington D.C. 20375

Mr J.E. Vernaglia
Naval Underwater Systems Center
New London, Connecticut 06320

UNITED STATES (Cont.)

Mr S. Weinstein
Naval Research Laboratory
Code 7593
Washington D.C. 20375

Mr Y.S. Wu
Naval Research Laboratory
Code 7590
Washington D.C. 20375

SACLANT HQ

CAPT A. Newing, UKN (Rtd)

SACLANTCEN

Mr P. Boni
Mr E. Cernich
Mr V. Duarte
Mr F. Jensen
Mr.W. Kupermann
Mr C. Maloberti
Mr M. McCann
Mr P. Nesfield
Mr G. Orsini
Mr R. Seynaeve
Mr G.C. Vettori

Other members of SACLANTCEN staff also participated.

TABLE OF CONTENTS

PART 1 - SESSIONS I to III

Table of Contents (Cont'd)

Table of Contents (Cont'd)

SESSION V BEAMFORMING

SESSION VI APPLICATIONS

INTRODUCTION

During the last few years, advances in the computer field have reached a level where it is possible to implement fast signal-processing systems through the use of standard computer equipment operating in association with more specialized devices, such as high-speed array processors and programmable beamformers. As it appears that in underwater acoustics many different signal processing requirements may be met by the use of sets of common hardware and software functional modules, a general-purpose, on-line, signal-processing facility designed to meet the combined needs of a number of underwater research project will often be the most appropriate approach towards making use of present technology. This can provide important savings in capital outlay and manpower, improve operations, and increase response capability. The purpose of the Conference was for specialists in the subject field to present their views and experiences in the designing of such general-purpose systems. The Conference was also open to other computer and signal-processing specialists who have a working interest in such systems.

Robert Seynaeve
Chairman

ACKNOWLEDGEMENTS

The following SACLANTCEN members made particular contributions to the organization of this Conference:

Mrs June Waller	All the administrative work to prepare the Conference and the Proceedings.
Mr P. Nesfield and Mr M. McCann	The reporting of questions and answers, and of panel discussions.
Messrs V. Duarte, P. Boni and C. Maloberti	Various aspects of the Conference logistics.

NOTE: For the record it is pointed out that the discussions and panels printed herein have not been edited or even read by some of the persons being quoted — this is particularly true for the Panel on General Purpose Systems and Future Trends held on the last day of the Conference.

SESSION I SYSTEMS

A GENERAL PURPOSE SOFTWARE SYSTEM FOR APPLICATIONS IN DIFFERENT FIELDS OF PHYSICS

by

Poul B. Ring
Danish Defence Research Establishment
Copenhagen, Denmark

ABSTRACT The paper presents the principle of a software system (FTSS) developed by the Danish Defence Research Establishment (DDRE) for applications in different fields of physics. Software is made up by independent modules exchanging data by standarized disc files. Special attention is paid to the production methods of new modules of the extensible system. Since the basic structure of all modules is the same a schedule for production of new modules can be established. Finally as an example an implementation of a ray-trace model is briefly described.

INTRODUCTION

The paper presents the principle of a software system (FTSS) developed by the Danish Defence Research Establishment (DDRE). The system has been in use since 1973.

The purpose was to make a standarized software system for a number of user groups working in different fields of physics. The intention was that common problems should be solved by a basic package of common software modules, but it should further be possible for the groups to make their own dedicated extensions. Furthermore the system had to be easy to extend for new demands.

These requirements led to the design of a software system with independent modules exchanging data by standardized disc files. Each module performs a specific task and new modules are added to the system depending on new demands. The system is interactive, but batch-facilities are also included.

For the time being (Aug. 1979) the system consists of about 100 modules in fields such as: Timeseries acquisition and analysis, simulation of differential equations, image processing, pattern recognition and a number of simulation models such as sound propagation models, sensor simulations etc.

The system is implemented on a HP-2100 computer using the monitor DOS-III. At present the system is implemented in the environment of the RTE-IV monitor. The software is implemented in Fortran IV.

1. FILE STRUCTURE

Data are stored in about ten types of standarized disc files, e.g. timeseries, spectrum, histogram, digital filter, image and record files. The last type of a file represents different types of tables of complex structures. These files are suited to simulation models.

The overall structure of the files is a division in a descriptor and a data section (See fig. 1). A descriptor contains information, name of module which created the file etc. An example of a descriptor for a timeseries file is shown on fig. 2.

2. STRUCTURE OF PROGRAMS

Software is divided into modules placed on three levels. (See fig. 3). The only module on level one is named MAIN. It is the only legal entrance and exit to the FTSS system modules. The most important task of MAIN is to initialize a system data area containing information about the state of the system, e.g. execution in interactive or batch mode, bookcounting of accessed files etc.

All processing modules are placed on level two. On level three a display module and an editor module (editing of descriptors) for each type of the FTSS-files are found.

When FTSS is entered by MAIN there is random access between the modules. The significance of the leves is: If you return to a module from a higher numbered level the module will be in the state as when it was left. In other cases a module is initialized when called.

All modules use only FTSS disc files when accessing data on the disc. To each type of file a collection of accessing subroutines is available.

3. THE COMMAND SELECTOR CONCEPT

Communication between a user and a module is performed by a command selector, which is a scheme containing a head-line and a number of indexed lines.

The head-line identifies the module. The indexed lines contain values of parameters and processing possibilities. The first step is to select the parameter-lines by typing their index and afterward change or insert the values of the parameters as desired. Parameters are numbers, text strings or file names. Each time a parameter is changed the terminal is erased and the command-selector is displayed with the new value. (Duration one sec.). Processing of data is performed by typing the index of one of the processing-lines, which are marked by a *. Examples of command-selectors are shown on fig. 4.

Each module has one and only one command-selector.

4. DESIGNING NEW MODULES

Several years experience has shown that the standardization of the communication by means of command-selectors is an advantage for the users of the software because the modules seemes very alike. But also for the designer and programmer the architecture gives advantages.
The main reason is that a fixed schedule is used in production of new modules. Following steps are performed.

- Design phase.
  - Seperate problem in modules.
  - For each module write down the command-selector.
  - Design (or copy) processing algorithms.

- Implementation phase.
  - Use the module generator program. This is a program having input as a brief description of a command-selector.
    Output is a Fortran program containing the communication part of the new module.
  - Implement the processing algorithms and merge them (as subroutines) into the generated program.

- Test phase.
  - Test each subroutine and repeat the examples for the whole module. (The classical colution).

In the design phase the command selector concept is a convenient tool in the discussions between designers and users having a demand for new software, because a set of command-selectors gives an informative description of user requirements.

In the implementation phase the use of a module generator reduces the production time because Fortran code-lines describing the communication is automatically generated. Another advantage is that the basic structure of all modules is the same (see fig. 5). Therefore a schedule for the implementation phase (see fig. 6) can be used.

In the maintenance phase the similarity of the modules makes it easier to correct a module, esp. in case the module was implemented by a former colleague.

5. BATCH

If an analysis is to be performed by many datasets it is clumsy to run the system interactively. Therefore FTSS has been designed to store communication sequences in FTSS batch files. These files can be executed with no interference. A sequence may use several modules and an arbitrary number of FTSS batch files can be stored.

6. IMPLEMENTATION OF A RAY-TRACE MODEL

As a dedicated software package an implementation of a ray-trace model is briefly described. (See fig. 7). The model consists of 7 independent modules exchanging data by disc files.

The model is based on analytical formulas by a set of functions fitting a measured sound speed profile and a description of bottom and surface by their reflection coefficients.

The modules perform the following tasks :

- Interactive fitting of Epstein profiles to a measured sound speed profile.
- Calculation of reflection coefficients of bottom and surface for a number of selected frequencies.
- By four modules the ray-pattern can be examined in details along a vertical or horizontal line.
- Calculation of the energy distribution of the sound propagation field along a vertical or horizontal line. In case a 2-D display of the energy field is wanted, the calculated energies are stored in an image-file which can be displayed or further processed by the FTSS image package.

The structure of the ray-trace model is typical of models implemented in FTSS, I.e. to split the model components, and let component data be generated by seperate modules and be stored in individual disc files.

CONCLUSIONS

The principles of a software system with applications in different fields of physics was presented. It is made up by independent modules exchanging data by standarized disc files, which makes the system very easy to extend. All modules have the same basic structure, a standarized communication between user and software and the same schedule is used in production of new modules.

For the user the advantage is a system easy to operate since all modules seems very alike. For the software staff the advantages are a system very excellent to work with in both the production and maintenance phases. After six years experience of working with the system it has proved that structured programming at all levels is very powerfull leading to a fast production of reliable software.

REFERENCES

1. P.B.Ring, J.K.Nielsen, O.S.Madsen :
A software package for timeseries analysis and acquisition.
(In Danish.) DDRE report 1976/29.

2. P.B.Ring, N.O.Jensen :
Digital circuit simulation.
DDRE report 1978/46

3. O.C.Mortensen, P.B.Ring :
A software package for fundamental image processing.
DDRE report 1978/42

4. A.V.Møller, P.B.Ring, O.C.Mortensen :
A database software package for a multichannel sensor system.
DDRE report 1978/47

5. P.B.Ring :
Software packages for description and rendering of geometric objects.
DDRE report 1978/48

6. M.Caspersen, P.B.Ring :
A simulation model for an infrared sensor in a scenario.
DDRE report 1979/27

7. H.Preuss, A.Nielsen, M.Andersen :
Description of geometric objects by their contours.
DDRE report 1979/30

8. T.Strarup, V.Jeppesen :
A ray-trace sound propagation model.
To be published.

9. T.Strarup, A.Pedersen :
A mono pole source array sound propagation model.
To be published.

10. G.Hvedstrup Jensen :
A software package for pattern recognition and image analysis.
To be published.

DISCUSSION

T. Totten Aren't disc files slow?

P.B. Ring It depends very much on the case; in the field of image processing it has been a problem, therefore, we extend the "core" to 256 K.

D. Nairn How many man-years were required in developing the system?

P.B. Ring Up to now we have invested approximately twenty man-years in modelling and programming the system.

R. Seynaeve What will be the future evolution of your system?

P.B. Ring The next step is to switch from the HP-DOS III monitor to the HP-RTE IV monitor and to add an array processor.

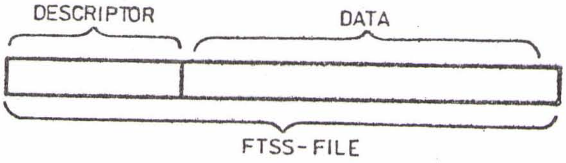
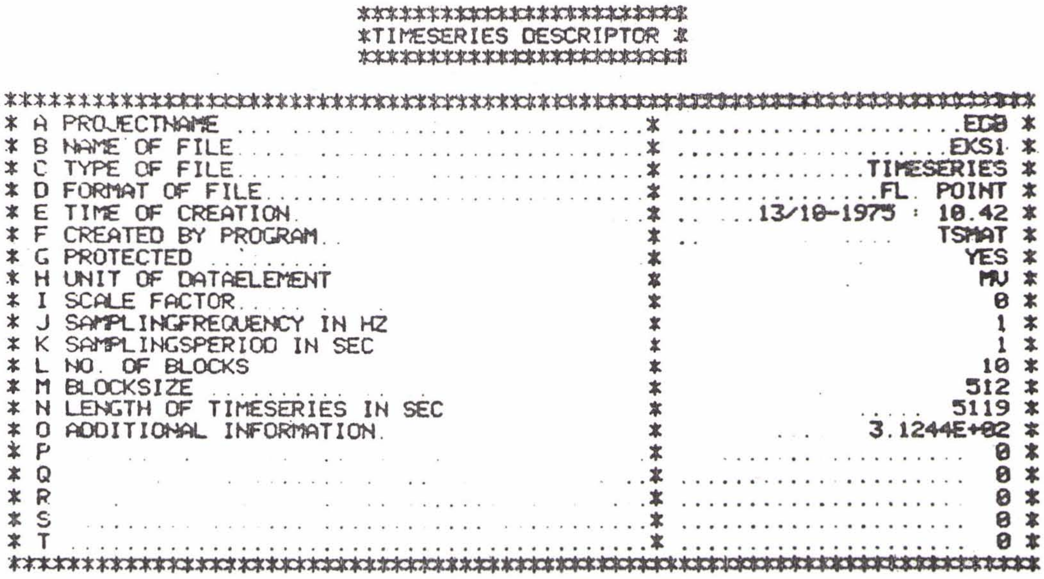


Fig. 1



```
                   **************************
                   *TIMESERIES DESCRIPTOR *
                   **************************

*****************************************************************************
* A PROJECTNAME ..................................* ......................EG0 *
* B NAME OF FILE .................................* .....................EKS1 *
* C TYPE OF FILE .................................* ...............TIMESERIES *
* D FORMAT OF FILE ...............................* ...............FL. POINT *
* E TIME OF CREATION .............................* .....13/10-1975 : 10.42 *
* F CREATED BY PROGRAM ...........................* ..                 TSMAT *
* G PROTECTED ........                            *                      YES *
* H UNIT OF DATAELEMENT                           *                       MV *
* I SCALE FACTOR ......                           *                        0 *
* J SAMPLINGFREQUENCY IN HZ                       *                        1 *
* K SAMPLINGSPERIOD IN SEC                        *                        1 *
* L NO. OF BLOCKS                                 *                       10 *
* M BLOCKSIZE ..........                          *                      512 *
* N LENGTH OF TIMESERIES IN SEC                   *                .... 5119 *
* O ADDITIONAL INFORMATION.                       *          ... 3.1244E+02 *
* P ..............................................* ......................0 *
* Q ..............................................* ......................0 *
* R ..............................................* ......................0 *
* S ..............................................* ......................0 *
* T ..............................................* ......................0 *
*****************************************************************************
```

Fig. 2

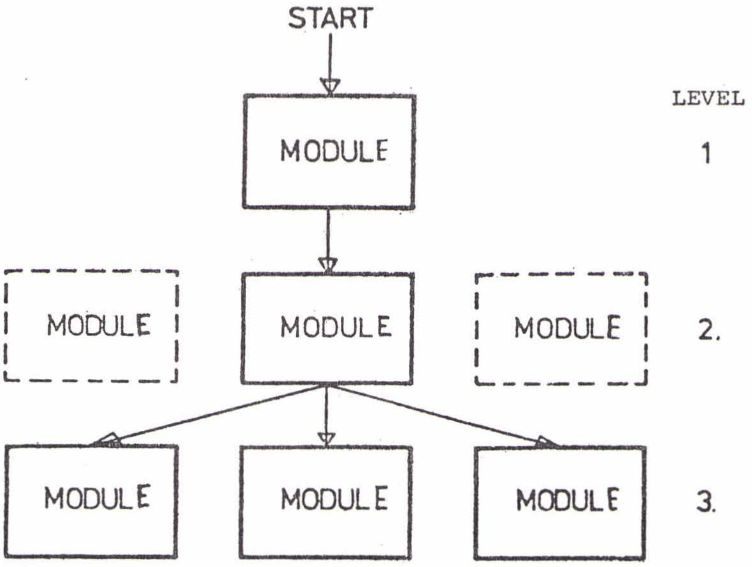
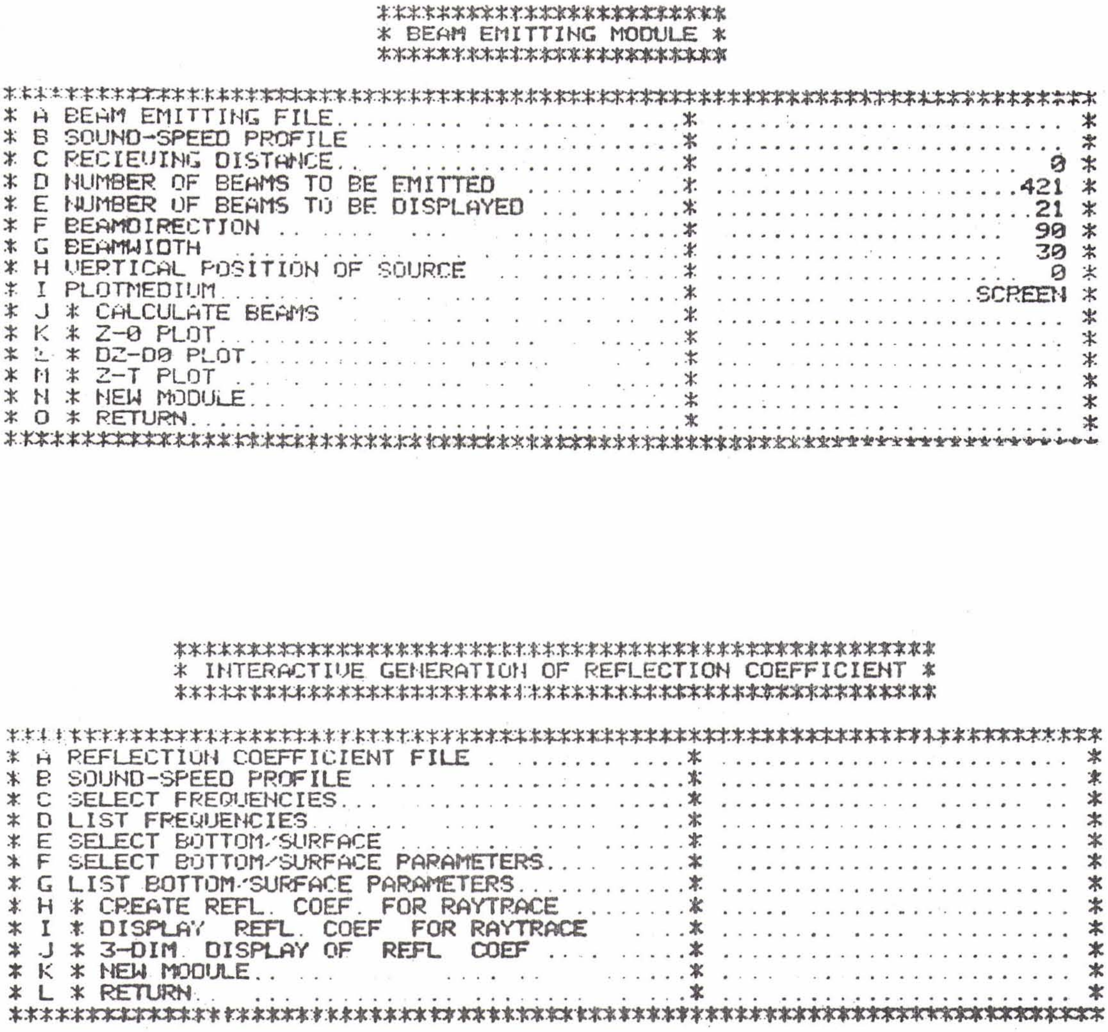


Fig. 3



```
                        ************************
                        * BEAM EMITTING MODULE *
                        ************************

**********************************************************************
* A BEAM EMITTING FILE.....................* .....................  *
* B SOUND-SPEED PROFILE....................* .....................  *
* C RECIEVING DISTANCE.....................* .....................0 *
* D NUMBER OF BEAMS TO BE EMITTED..........* ...................421 *
* E NUMBER OF BEAMS TO BE DISPLAYED........* ....................21 *
* F BEAMDIRECTION..........................* ....................90 *
* G BEAMWIDTH..............................* ....................30 *
* H VERTICAL POSITION OF SOURCE............* .....................0 *
* I PLOTMEDIUM.............................* ................SCREEN *
* J * CALCULATE BEAMS......................* .....................  *
* K * Z-0 PLOT.............................* .....................  *
* L * DZ-D0 PLOT...........................* .....................  *
* M * Z-T PLOT.............................* .....................  *
* N * NEW MODULE...........................* .....................  *
* O * RETURN...............................* .....................  *
**********************************************************************



             ************************************************
             * INTERACTIVE GENERATION OF REFLECTION COEFFICIENT *
             ************************************************

**********************************************************************
* A REFLECTION COEFFICIENT FILE............* ......................*
* B SOUND-SPEED PROFILE....................* ......................*
* C SELECT FREQUENCIES.....................* ......................*
* D LIST FREQUENCIES.......................* ......................*
* E SELECT BOTTOM/SURFACE..................* ......................*
* F SELECT BOTTOM/SURFACE PARAMETERS.......* ......................*
* G LIST BOTTOM/SURFACE PARAMETERS.........* ......................*
* H * CREATE REFL. COEF. FOR RAYTRACE......* ......................*
* I * DISPLAY REFL. COEF FOR RAYTRACE......* ......................*
* J * 3-DIM. DISPLAY OF REFL COEF..........* ......................*
* K * NEW MODULE...........................* ......................*
* L * RETURN...............................* ......................*
**********************************************************************
```

Fig. 4

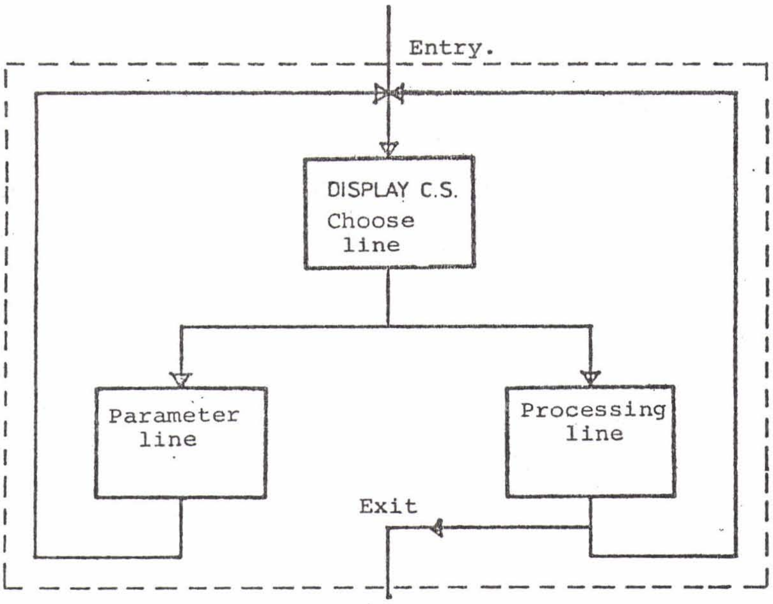


Fig. 5

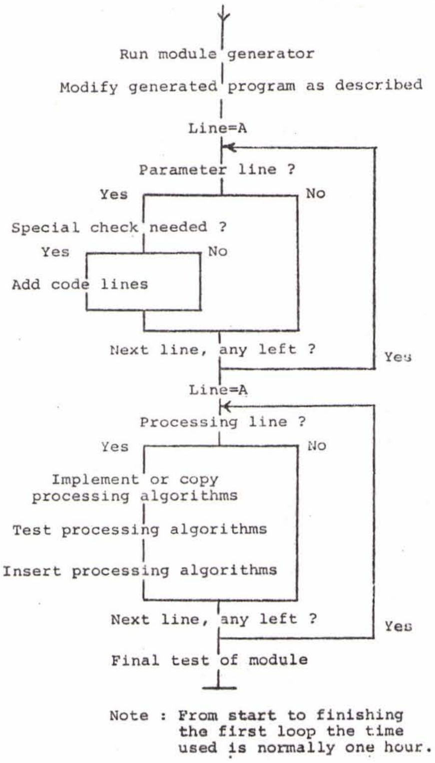

Run module generator
Modify generated program as described
Line=A
Parameter line ?
Yes
No
Special check needed ?
Yes
No
Add code lines
Next line, any left ?
Yes
Line=A
Processing line ?
Yes
No
Implement or copy processing algorithms
Test processing algorithms
Insert processing algorithms
Next line, any left ?
Yes
Final test of module
Note : From start to finishing the first loop the time used is normally one hour.

Fig. 6

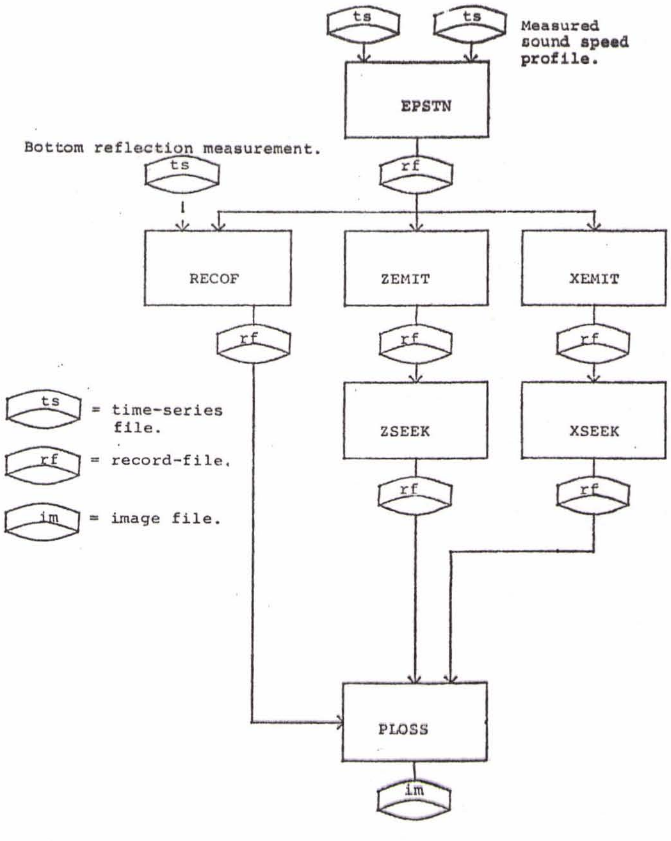

ts
ts
Measured sound speed profile.
EPSTN
Bottom reflection measurement.
ts
rf
RECOF
ZEMIT
XEMIT
rf
rf
rf
ts = time-series file.
rf = record-file.
im = image file.
ZSEEK
XSEEK
rf
rf
PLOSS
im

Fig. 7

A Versatile Signal Processing System for Producing the Angular and Frequency Distribution of Acoustic Energy in Real Time

by

James M. Griffin
NAVAL RESEARCH LABORATORY
WASHINGTON, D. C. 20375 USA

ABSTRACT For a number of years the US Naval Research Laboratory (NRL) has been conducting a research program in underwater sound propagation using large aperture hydrophone arrays. This research requires the spectral and angular distributions of the sound field with a variety of bandwidth and throughput rate requirements. As the size of the aperture, in terms of number of hydrophones, increases, the demand on the signal processing equipment increases correspondingly. In addition to this growth in processing needs, the move to at-sea, real-time systems demands high reliability and very high speeds to prevent loss of data.

A sea-going signal processing system has been developed at NRL to meet these processing requirements. The system is versatile in that it is composed of a number of small array processors operating in parallel which can be redistributed to handle the varying processing loads presented by different research interests. It is designed to handle 256 imput channels at an overall input rate of 256 kHz. The system is designed around a dual bus architecture connected by an interbus window and controlled by a single CPU. All components of the system are commercially available plug in components making replacements simple and system reliability high. This paper describes the system architecture, the system design philosophy and current operating capability.

INTRODUCTION

The desirability of performing data analysis at sea has been recognized for some time. Until recently, however, the speed and size of array processing equipment has not permitted the high data rates necessary to do spectral analysis and beamforming at sea. Systems were either too bulky to be easily taken aboard ship or too slow to handle the required throughput rates. The Naval Research Laboratory has developed a system designed around a number of new compact array processors which is easily transported, can withstand the harsh environment at sea, and is fast enough to handle the real time analysis of multichannel systems.

The use of an on-line system such as the one presented here gives the investigator a distinct advantage over the approach of recording a large number of data channels and performing the analysis back in a laboratory facility. One major advantage is that examination of on-line results can verify the quality of the data being taken and changes in the experimental procedure can be implemented if the situation warrants. A second major advantage to on-board analysis is the significant reduction in lag time between data collection and reporting of results. By performing the bulk of the data reduction at sea rather than competing for valuable processing time at the lab system, the investigator can begin the real challenge of interpreting his results as soon as the experiment is complete.

At the time of this writing the system has been used successfully in two major experiments. The first was performed in February and March 1979 off the east coast of New Zealand and involved the angular and frequency distribution of long range reverberation resulting from large broadband explosive sources. The second was performed in June 1979 off the east coast of the United States and involved the angular and frequency distribution of long range propagation from narrowband cw signals. Reports on both of these experiments have been submitted for publication, which attests to the time savings obtained by on-line data reduction. The difference in signal processing requirements for these two experiments demonstrates the flexibility this type of system offers.

This paper discusses the basic system design philosophy and the architecture of the system. The two configurations used for the two sea experiments performed thus far will be discussed to illustrate the flexibility of the design. Finally, an examination of likely directions for system growth will be presented.

1. BASIC SYSTEM DESIGN PHILOSOPHY

For a system to be practical as a sea-going real-time research tool it must be compact for easy transportability and easily adapted to a variety of processing needs. In addition, the system must be easily repairable at sea or be designed to allow graceful degradation in the event of partial system failure. The approach taken by the Large Aperture Acoustics Branch at NRL is to, wherever possible, use a number of small array processors which fit directly into the backplane of a DEC PDP 11/34. This creates a very compact system with all the processing elements in standard DEC expansion boxes.

The use of these small array processors offer some real advantages over larger processors. Most processing jobs can generally be split into a number of smaller jobs which can be executed in parallel. By assigning a different array processor to each phase of the job a high throughput rate can be achieved. Since each job will differ in the amount of processing power needed at a given phase this system permits the movement of power to the points where it is most needed.

By using several processors of the same type a small number of spares can protect against all but the most catastrophic component failure. Should a processor fail when no backup is available it is possible to eliminate some stages of the analysis scheme and continue running in a partial data reduction mode, thus giving graceful system degradation.

2. BASIC SYSTEM ARCHITECTURE

Early in the development of the system the decision was made to design around the DEC UNIBUS. One reason for selecting the DEC mainframe was to provide software compatibility with the large laboratory facility already in use. A second reason for this selection was the widespread industry support of this bus which gives good flexibility in the choice of components.

Since this system is to be used in the potentially harsh environment of a ship a ruggedized version of the PDP 11/34 built by Plessey was selected as the control element for the system. This unit when used with large TOPAZ isolation transformers provides protection from the high vibration environment and somewhat variable supply voltage.

There are two basic constraints which limit any digital signal processing system, namely the amount of memory the system can support and the total data rate the system can maintain. It was apparent in designing toward a system that could handle 256 channels at a throughput rate of 1 KHz per channel that both these constraints would be a problem with the DEC UNIBUS. By using a two bus system, however, it is possible to support nearly twice the memory and to divide the bus traffic so that the data flow does not exceed the bandwidth of either bus. Since most of the processing load is being carried by array processors installed as system devices the load on the CPU is relatively light and a single CPU can control both busses.

Thus, the basic system architecture (Fig. 1) is designed around a dual bus DEC computer controlled by a single 11/34 CPU. Devices for analog interface, processing, storage, display, and user interface can be moved about as needed for a particular processing job.

3. FEBRUARY 1979 CONFIGURATION

In the February experiment it was necessary to have good spatial resolution, all beam coverage and 100% throughput with broadband sources. The general analysis scheme was to double buffer the input data, take 1024 point transforms to convert to the frequency domain on each channel, and finally to distribute the energy into the angular domain. The results were then stored on digital tape and displayed for on-line monitoring.

3.1 Hardware

With the basic analysis procedure in mind, it is possible to decide on the specific hardware configuration for this application. The first decision to be made is how to split up the processing between the two data busses. Since the A/D operation can take place without loading a system bus the logical split is to perform the time-to-frequency transform on the aux bus while the space-to-angle transform is handled on the main bus.

To accomplish the A/D and double buffering requirements a pair of Computer Design and Application Micro Input Processors (MIPs) were chosen with enough on board memory to handle 32 channels double buffered on one and 8 channels double buffered on the other. These input processors are smart processors which can be programmed to automatically handle the double buffering needs thus freeing the CPU of this chore.

The main signal processing effort was accomplished using the Micro Signal Processors (MSPs) also by CD&A. Since the time-to-frequency transform can be accomplished with a fast Fourier transform a single MSP card was placed on the aux bus for this purpose. The space-to-angle transform was more complex so a pair of MSP cards were placed on the main bus for this phase of the operation.

This leaves the fairly routine decisions of memory requirements, displays, human interfaces, and storage. The memory requirements are easily handled with 64K memory on each bus. The display is a Ramtek model 9153 color display which offers 256 by 256 spatial resolution with different colors being assigned to each of 256 possible intensity levels. Human interface to the machine was provided by two terminals, one Tektronix 4006 CRT to give plot capability and one Texas Instruments Silent 700 to give hard copy. Two types of storage devices were selected for the system; for permanent retention of data a Kennedy 800 BPI 9-track tape drive was used, while for temporary data storage and off-line analysis three 10 M Byte disk drives were selected. The resulting hardware configuration, shown schematically in Figure 2, fits into a two rack system which is easily transported.

3.2 Data Flow

The general flow of data from analog input to digital tape and display is relatively straight forward. The analog input is digitized and auto double buffered by the two MIPs into their onboard memory. The data is made available on the aux bus through an 8K window which is part of the MIPs. This data is transferred one channel at a time to the aux bus MSP where it is transformed to the frequency domain. The transformed results are placed in the 64K aux memory where they are made available to the main bus through the bus window. The lines of interest are taken by the two main bus MSPs and distributed in angle. The angular distributions for each individual frequency are converted to power and then summed across frequency by the CPU. These average distributions across bands of frequencies are then transferred to digital tape for permanent storage and to the color display for on-line monitoring.

3.3 On-Line Results

The results displayed in a real time mode can be a significant advantage in running an experiment. In this case there were three broad bands of acoustic energy being distributed into the time and angle domains plus a pair of cw sources similarly distributed. The on-line display consisted of 64 pixels dedicated to each of the three broadband sources and 32 pixels to each of the narrowband sources, where each pixel corresponds to a given steering direction. This data was sent to the color monitor in a scroll fashion with a new frame being output every second. The result is a display which shows the angular distributions of all bands for the past 256 seconds.

A typical display reproduced in black and white is shown in Figure 3. This shows returns from large topographic features such as seamounts in a time period from about 5 to 9 minutes after the source detonation. As can be seen even in this black and white version, the returns are quite strong and it is clear that data quality is high.

4. JUNE 1979 CONFIGURATION

The second use of the system took place in June 1979 off the east coast of the United States. The type of processing required in this second experiment was quite different from the first. In this case, the interest was in a large number of cw sources and good angular and spectral resolution was the goal, while the 100% throughput requirement could be relaxed.

The high spectral resolution, 1/16 Hz, desired implies the need to take long time samples. At the same time the need to obtain a 500 Hz bandpass requires a sampling frequency of 1024 Hz. The resulting sample size required is 16384 for each of 128 input channels.

4.1 Hardware

Several hardware changes are imposed by the requirements for this system. The two largest changes are caused by the need to take larger transforms and the need to store 128 channels of 16K words each before transforming can begin.

The large storage requirements were satisfied by placing a bulk memory device capable of holding four megabytes of data. This device, manufactured by Monolithic Systems, appears to the system as an RF11 disk system. This device satisfies the storage requirements but does not permit any double buffering of input data. However, since this application does not require a 100% throughput rate this is not a necessary part of the processing.

There are two possible solutions to the requirement of taking 16K transforms. First it is possible to perform the transforms in 4K pieces using the small MSP array processors and then combine the results to acheive the desired 1/16 Hz resolution. Second it is possible to replace the small processor by a large processor which has the capacity to perform the larger transforms. In order to take advantage of previously developed software an SPS-81 normally used in our lab system was selected for use on the sea system.

Two other hardware changes were made for this experiment. The input processors needed to be expanded to handle the required 128 input channels. This consisted of expanding the second MIP to a full 64 channel capability. The final change was dictated by the data output rate. While 100% data throughput was not required it was important not to have large gaps in the data. Hence, no time can be taken for tape changes and additional drives must be added to the system. The final hardware configuration used in this experiment is shown in Figure 4.

4.2 Data Flow

The general flow of data in this configuration begins with the digitization and auto double buffering of the 128 input channels by the two MIPs. The data is moved from the MIP memory to the bulk storage device until 16K points have been saved for each channel. At this point the sampling is suspended and the data on the bulk storage is passed one channel at a time to the SPS-81 and transformed to the frequency domain and selected lines are stored in aux bus memory. The data lines are written to tape at this stage so that different shading coefficients can be applied prior to beamforming in the lab facility.

In order to obtain on-line monitoring the data must still be beamformed in real time. Hence, the data is passed on to the MSPs which perform the angular distributions. At this stage, the data is transferred to the main bus to be converted to power and sent to the disks which provide a 2 1/2 hour display memory. The operator can then select a variety of displays which give a history of either the hydrophone intensities or the angular distribution of energy for any selected frequency bin.

4.3 On-Line Results

A typical display, again reproduced in black and white, is shown in Figure 5. This displays a 2 1/2 hour history of the angular distribution of one of the possible frequency lines. It is possible to determine from these displays the strength, direction, and Doppler shifting of any of the deployed sources.

A second type of display is shown in Figure 6 which gives the average hydrophone intensity at a selected frequency as a function of phone number and time. This display can be used to immediately identify dead or weak channels and permits early correction of such problems.

5. DIRECTIONS OF SYSTEM GROWTH

While the basic system design has worked quite well in two quite different experimental applications some system improvements are called for. The three general areas of system growth are; increased channel capability, increased CPU speed, and greater array processing capability.

The channel capability will soon be expanded to the original design goal of 256 channels with the addition of two fully expanded MIPs. It has become apparent in the last two experiments that software development will be greatly simplified on the expanded system with the addition of a faster CPU. For this purpose a PDP 11/55 will replace the slower 11/34 as the main system CPU.

The final system change will be in the choice of an array processor capable of handling the large transform sizes needed in future experiments. While the SPS-81 is capable of the speed and size requirements it has two disadvantages. First, it is considerably larger than the card size MSPs which presents some transportation problems. Second, by using a single large processor, such as this, it becomes a critical link in the system and the system cannot be gracefully degraded if it should fail. One solution being considered is to use a number of the small MSP processors along with a second CPU as a custom made array processor. By using components identical to the ones already on the system spares are easily available and graceful degradation is possible.

CONCLUSIONS

The system developed by the Large Aperture Acoustics Branch at NRL was designed to be versatile, compact, and reliable. The early implementations of this system, which have been presented here, have been quite successful and have confirmed the validity of the basic design. Future system improvements will make the system capable of performing both frequency and angular distributions at an input rate of 256 KHz.

DISCUSSION

J.S. Pyett With 256 channels at 1 kHz, what is the wordsize and the percentage loading on the auxiliary bus?

J.M. Griffin The wordsize referred to is 12 bits from the A/D converter. Once on the bus, however, we are actually transferring 16-bit words since no packing is done.
I would estimate the total bus load for all DMA's on the auxiliary bus at near 50%.

D. Steiger Is any processing done other than in the array processor?

J.M. Griffin The only processing done by the CPU are the simple tasks such as block floating to true floating conversion and averaging of data after it has been reduced.

D. Steiger What data is recorded?

J.M. Griffin The data recorded varies with the experiment. In the January configuration we were writing beam power in three frequency bands as a function of time. In June the output was a set of frequency estimates for each hydrophone. Although the beam power was computed and displayed, this does not represent a data reduction so no tape is saved. In addition writing the frequency estimates permits application of a variety of weighting coefficients or phase corrections after the fact since phase information is not lost.

D.V. Crowe What operating system do you use?

J.M. Griffin Software development is done under RSX 11M. Once complete the program is booted from disc and runs in a stand-alone mode.

R. Seynaeve How much effort was involved in say the preparation of the June experiment?

J.M. Griffin From a software standpoint, after the development of a general approach, approximately three man-months were used. It should be pointed out, however, that some of the routines, such as those to drive the SPS, had been previously developed.

R. Seynaeve How do you synchronize data?

J.M. Griffin Each device has interrupt capability. Typically the interrupt routine consists of setting a flag. A master scan loop is written to start the devices in proper synchronization.

R. Seynaeve Is data transfer by DMA's?

J.M. Griffin Yes, all the major devices have DMA capability.

R. Seynaeve Do you experiment with pseudocolour and do you find colour useful?

J.M. Griffin We have not made a scientific study of pseudocolour. This particular table was developed by trial and error to be intuitive, not to display maximum information. I believe colour can be of significant advantage in distinguishing intensity levels, although a different colour table would certainly be chosen for this.

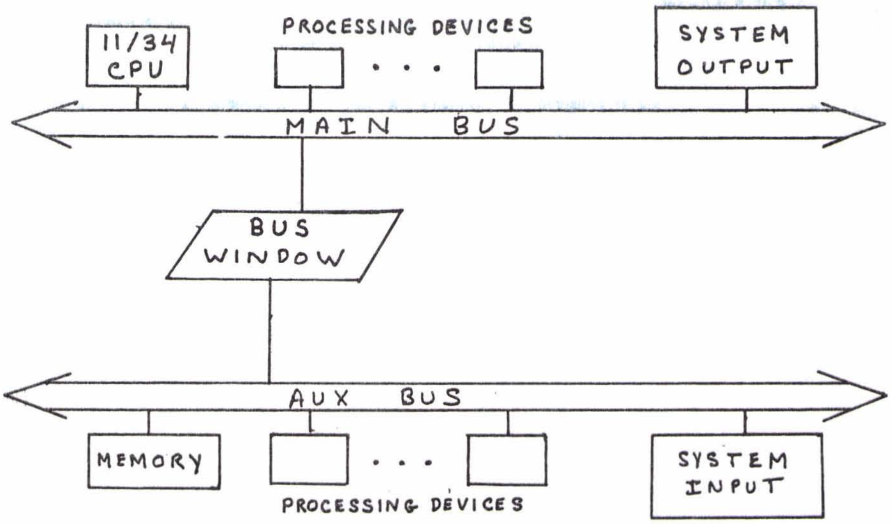


FIG. 1 BASIC SYSTEM ARCHITECTURE

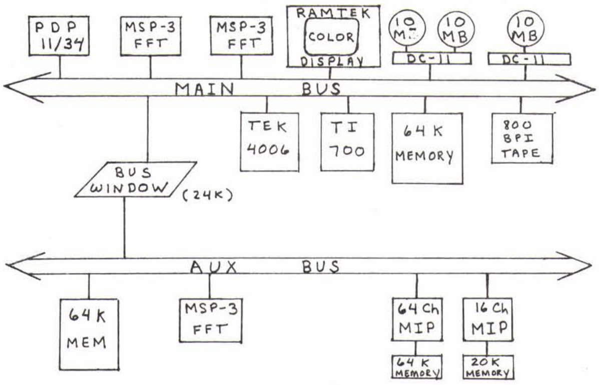


FIG. 2
SYSTEM CONFIGURATION FOR
FEBRUARY 1979

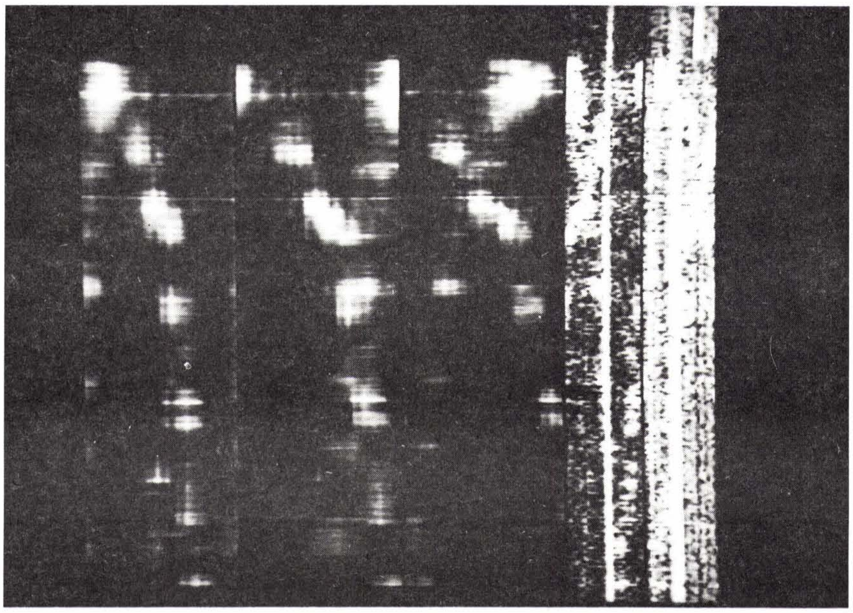


FIG. 3 ON-LINE ANGULAR DISTRIBUTION OF BROADBAND RETURNS
FEBRUARY 1979

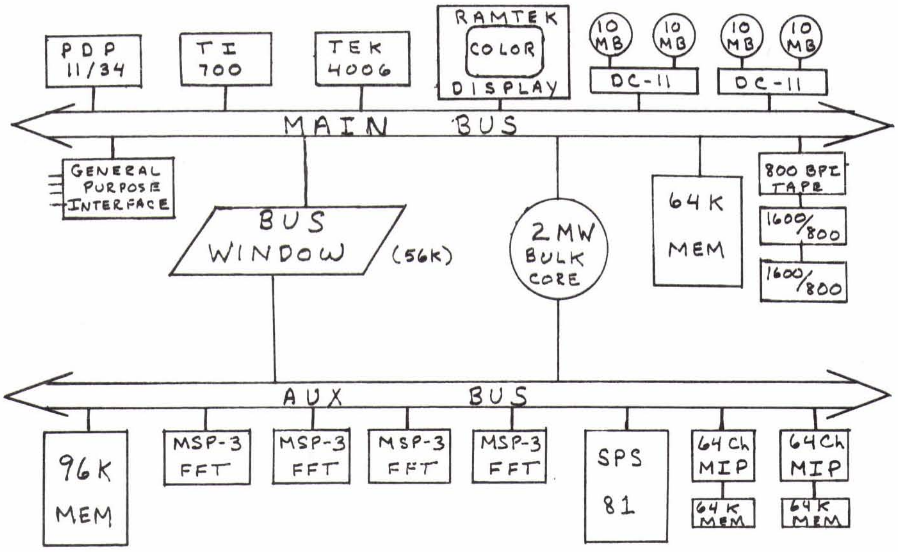


FIG. 4 SYSTEM CONFIGURATION FOR JUNE 1979

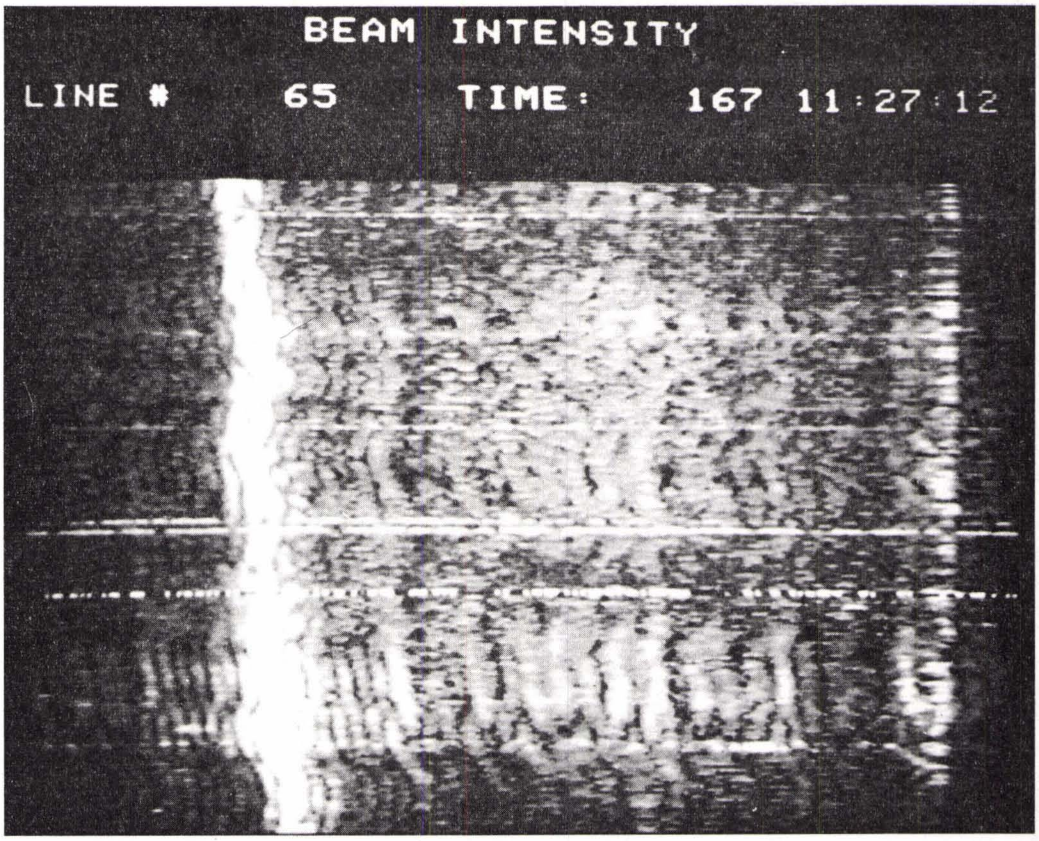


FIG. 5 ON-LINE ANGULAR ,DISTRIBUTION OF NARROWBAND SOURCE JUNE 1979

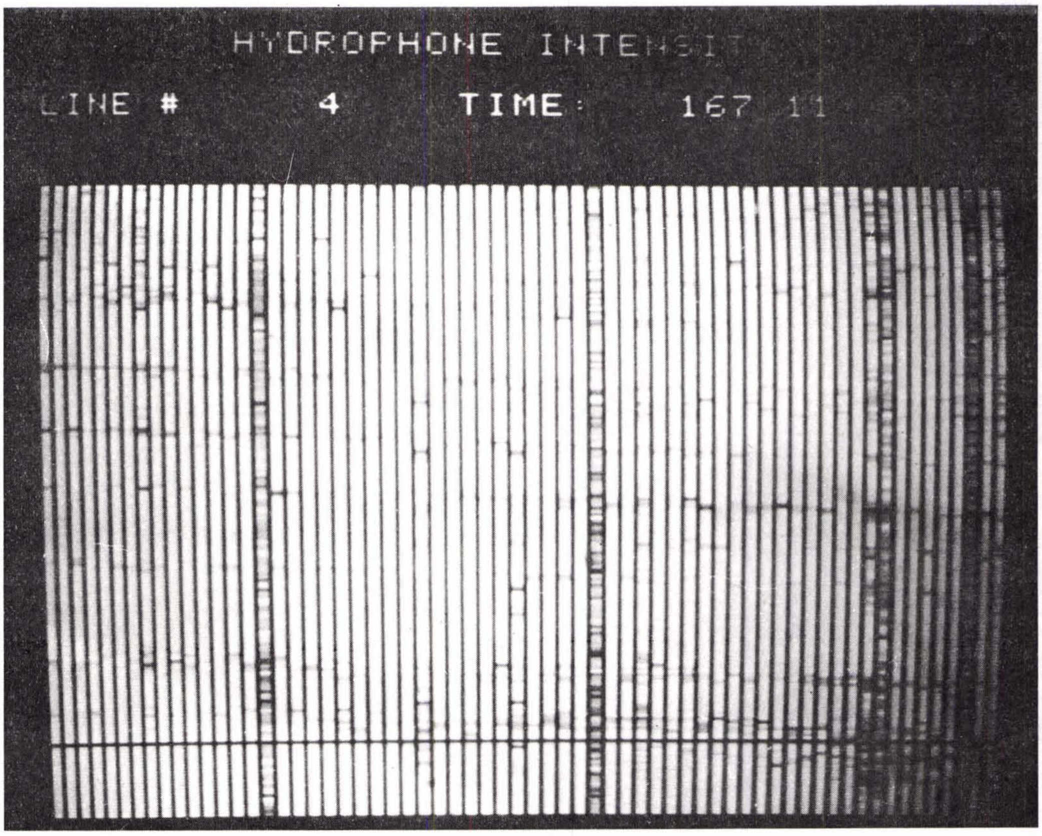


FIG. 6 ON-LINE HYDROPHONE INTENSITY DISPLAY JUNE 1979

SACLANTCEN REAL-TIME SIGNAL PROCESSING SYSTEM

by

R. Seynaeve
and
P.A. Boni, V. Duarte, M. McCann, P. Nesfield
SACLANT ASW Research Centre
La Spezia, Italy

ABSTRACT To meet its increasing requirements in real-time signal processing, SACLANTCEN has developed a general purpose signal processing system called WARP to be used with a fairly wide range of underwater acoustic research projects. To achieve a high throughput and an easy programming, WARP is organized in several subsystems corresponding to the successive signal processing tasks to be performed, each subsystem being relatively independent from the others. Great emphasis has been given to the software and the system interfaces with the user and to the on-line testing and debugging facilities. One of the aims is in fact to make the control and programming as easy as possible, through the use of special purpose languages and interactive methods, thus providing the flexibility and reliability required by the research projects. Because of the wide range of requirements WARP is configurable into larger and smaller systems, all using the same hardware elements, operating system and high-level language.

INTRODUCTION

In the early 1970's the Centre's needs in real-time signal processing were generally met by using mini-computer systems with hardware Fourier processors, together with specialized software for acquistion and signal processing, such as ITSA and SPADA [1,2]. Over the last few years the requirements have increased considerably, and a new family of systems called WARP I has been designed to meet the current and future needs of a fairly wide range of underwater research projects.

1. DESIGN CONSIDERATIONS

Among the most important design goals were the following:

(a) High Throughput : The system was to be used in experiments with multi-sensor arrays and large bandwidths.

(b) Variable Configuration : To cover a fairly wide range of requirements, WARP was to be configurable from a basic set of elements, into specific array processor based systems, all with the same macro languages and operating systems.

(c) Fast Response to User Requirement : The system hardware architecture and software design was to allow easy and reliable programming even for complex tasks.

(d) Directly Programmable by Users without extensive computer experience and, therefore, general use of interactive methods and macro languages.

(e) Special attention to display/graphic end.

(f) Hardware reliability and therefore possibly graceful degradation for the larger systems, and self-tests of subsystems during run time.

(g) Continuity with earlier techniques and systems, wherever possible.

2. SYSTEM DESCRIPTION

2.1 System Overview

Figure 1 shows the functional block diagram of an average WARP configuration, such as used in some towed-array sonar experiments. Figure 2 shows its actual implementation.

(a) The various functions are performed in well defined subsystems: spatial processing in the beamforming subsystem (BF), frequency and other processings in the array processor subsystem (AP), tracking/display/graphics (TD) and control (CO) in the corresponding subsystems. The data synchronization problems between subsystems are eased through the use of the buffer memories I and II.

Buffer II is especially large, therefore, uncoupling the tracking/display end from the on-line processing. As the three subsystems are well decoupled, the programming is easier and more modular, it can more easily be done by separate programmers and the final integration is quick. These factors are of major importance in a multi-project research environment where a quick reaction time to new or changing requirements is a very important point.

(b) The arithmetic can accommodate a large dynamic range: 24-bit arithmetic in the BF with 16-bit block floating output, 32-bit floating point (sometimes 16-bit floating point) in the AP, and 32-bit floating point in TD.

(c) The software reflects the decoupling of the hardware implementation: upon the selection of an operational mode, the control subsystem despatches the appropriate programs and parameter files from the subsystem's libraries to each of the subsystems and starts the processing sequence; many such modes can be prepared individually and stored for an experiment or, alternatively, assembled at sea as the need arises. The subsystem's programming is largely based on the use of interactive methods and special macro languages.

(d) The storage of input and/or output of the data depends on the application and uses peripherals ranging from a 42 Megabit/s very high speed tape recorder or a large disc at the input to a 1600 bpi/75 ips computer tape recorder at the output.

(e) The configuration of Figure 2 can be reduced to smaller ones supported by the same software but where, for example, there is no time domain beamformer, and where the output computations are made in the control processor; alternatively, in some situations, another control processor driving continuously the BF [see para 2.2] and performing other external measurements can be necessary. Such modifications to the system are easily implemented because of the flexibility provided by a modern multiprogramming operating system (RTIV) which includes a good distributed computing software (DS 1000).

The following sections will further describe the various parts and subsystems.

2.2 The Host Computers and Discs

Each of the BF, AP and TD subsystems require support from a host CPU (the FE needs very little). In many applications the BF needs little more than the loading of microcode at the time of setting up a mode of operation and the monitoring of the scaling of the output data. In other applications, where the beamforming is time varying, or recording on disc is required at the level of the beamformer, a CPU dedicated to the BF subsystem can be necessary. The AP and TD require a continuous service, but in the applications where fairly straight forward graphics are required with little operator interaction, one CPU with enough memory can often suffice for both.

In other words, a system can include from one to three host CPU's. The model used is the Hewlett Packard 21MX E and F versions with 7906 20 Megabyte moving head discs. The F version which has a floating point processor is somewhat faster. It is basically a 16-bit mini with a 350 ns extended memory (up to 1 Megaword) accessable through a dynamic mapping system. It can be microprogrammed (arithmetic and I/O) by the user. Typical times for the 21 MX-F are:

	Single Precision	Double Precision
Floating Point Add	.63 μs	.68 μs
Floating Point Multiply	1.78 μs	2.75 μs
Sin	47 μs	
Sqrt	30 μs	
$\log_{10}$	50 μs	

The 21MX's Real-Time IV multiprogramming operating system includes many interesting features among which a file manager, a distributed system software (DS 1000) and the handling of large arrays. The 7906 disc has a 10 Megabyte removable cartridge and its controller can handle several CPU's and several discs, allowing to implement some very interesting configurations.

It is important to note that in a WARP multi CPU configuration, it is usually possible to reconfigure the system with one CPU only, if the need occurs, however, at the cost of an overall slowing down.

2.3 The Analogue Front End

As will be seen in the next paragraph, the type of beamformer used requires that the data be sampled well above the Nyquist rate for accurate beamforming. The remote controlled front end, which was designed by the Centre's Electronic Department, allows for the simultaneous sampling of up to 80 inputs (expendable to 160) at the following rates

Number of Inputs	Max A/D Conversion Rate/Input
1 to 20	100 kHz
21 to 40	75 kHz
41 to 60	50 kHz
61 to 80	25 kHz
81 to 160	12.5 kHz

It should be noted that the simultaneous sampling of the inputs is not necessary for beamforming as the delays between sampling times could be compensated in the beamforming microcode. However, as in several applications the beamformer is bypassed, simultaneous sampling provides often more simplicity in the successive signal processing.

2.4 The Programmable Digital Beamforming System

2.4.1 The Beamformer

In mid 1977 the Centre issued the specifications for a programmable beamformer, the development of which was awarded to PLESSEY, U.K. The equipment was delivered at the end of 1978 and has been in operation since.

This beamformer is of the partial sum type, i.e., as the wavefront travels along the array, the contributions of each hydrophone to each of the output beams are accummulated in ad hoc memory locations ("partial sum memory"), which correspond to the output beam samples. A value in such a location is outputted as a beam sample when all the contributions for that beam sample has been received. No memory is required at the input and thus a high input sampling rate, giving a high accuracy in the beamforming, is easily achievable. This beamformer implementation is specially interesting when a high input time resolution, a large number of hydrophones and a relatively low number of beams are desired.

The implementation is made as a microprogrammable pipeline processor as shown in Fig. 3. The input data is sampled at high rate (typically 16 times Nyquist frequency) and is temporarily stored in the small input buffer. The long words of the control memory each contain the address of an input sample, its shading coefficient, the address of the partial sum location where it must be added, and several control bits, including one which is set when all contributions have been received and the sample can now be output. There is one such control word per hydrophone and per beam. The beamforming performed is thus entirely defined by the control memory content, and therefore beamforming can be done in 1, 2 and 3 dimensions, with any shading and any beam direction, this provided a set of limiting relations is respected [Fig. 4]. There are two separate program memories and the control can be moved from one to the other in only one memory cycle; this feature allows to implement various types of time varying beamforming. The main specifications of this beamformer are as follows:

Maximum number of sensors : NH = 255

Maximum number of beams : NB = 255

Control store : NH · NB = 2048 x 48 bits
(expandable to 4096)

Partial sum store : 4096 x 24 bits
(expandable to 16384)

Input buffer : 8192 x 12 bits

Output buffer : 4096 x 16 bits (expandable to 8192)

Multiply/add rate : NH · NB · $f_o < 3$ MHz

Maximum input data rate : NH · fi < 3 MHz

Maximum fi/fo : 255

Input word : 12 bits

Shading word : 8 bits or 7 bits + sign

Arithmetic/partial sum word : 24 bits

Output word : 16 bits selectable out of the 24 bits
of the partial sum memory.

2.4.2 Software and Development Aids

The beamformer subsystem is supported by BF-M, an interactive program for beamforming microcode generation. The user specifies the geometry of the array, the direction of the beams, the shading coefficients of each sensor for every beam. This results in a table of parameters which is used by the microcode generator; both the microcode generated and the parameter table can be filed away. Means are available to readily edit ASCII parameter tables or microcode.

Program BF-SIM allows to synthesize and send families of wavefronts to the BF and to display the measured beamforming patterns to check the beamformer design and monitor quantizing effects.

2.5 The Array Processor Subsystem

2.5.1 The MAP 300

The array processor is the central element of the WARP system. The model selected is the CSPI MAP 300, a block diagram of which is shown in Fig. 5. Its principal components are:

(a) A dual pipelined arithmetic unit with input and output queues and separate address computation unit [Fig. 6], capable of about 5 million multiply-add per second.

(b) Three memories; one high speed (160 ns) of 8192 words of 32 bits (bus 3), and two medium speed (500 ns) of 40960 words (bus 2) and of 16384 words (bus 1) both of 32 bits.

(c) A control processor (CSPU)

(d) A host interface processor (HIM)

(e) One or more input/output processors (IOS, ADAM, AOM).

The three memories are connected to all the other elements through their multi-port buses; however, the executive and programs run usually on bus 1 memory, while the data are stored on bus 2 and bus 3 memories.

Some of the most interesting features of this processor are its speed, its modularity, its flexibility in input/output, and the real-time features of its operating system.

Most of the AP programming is normally written in SNAP (CSPI's macro-language for signal processing) which is very comprehensive and allows for very good communication with the external world. Some excellent features are:

Logical rather than absolute buffers.

Several data formats (32 bits, 16 bits and 8 bits, fixed and floating).

Ability to handle several I/O processors concurrently.

Its buffer protection in arithmetic and I/O processing.

A more detailed presentation of these features, with their meaning to the programmer, are given in a separate paper [3].

2.5.2 Speed Considerations with SNAP

As SNAP uses an interpreter, the time taken by the MAP to perform an operation consists of the execution time itself and the executive's overhead required to load and start the various MAP modules. The execution time is usually proportional to the block length, while the overhead time is fixed but can be more or less overlapped with the execution time of the previous operation.

With the current software, on the average, at a block size of about 256, the overhead time becomes about equal to the executing time. Using a technique called "binding", where sets of consecutive operations are linked together and stored under a new function name, the overhead time can be reduced to about 250 μs at the expense of more program memory. A further improvement called "stacking" should be available shortly from CSPI. To summarize, the relative speed of the MAP will increase if the data are processed in large blocks (typically 1024 words and above).

2.5.3 Software and Development Aids

At SACLANTCEN over the last nine years, interactive techniques have proven very effective for the design and trouble-shooting of signal processing systems in a quickly changing research environment [1]. Our first experience with the MAP confirmed quickly that here too such techniques would be of great use:

(a) We extended SNAP with AP-M, an interactive monitor which allows to create a signal processing program, step through it visualizing buffers on CRT's, variables, and tables on a terminal, with the possibility of quickly modifying it and observing immediately the effect of the change without recompilation.

(b) We developed a facility called AP-STROBE which allows to observe through a separate I/O processor any buffer, table, variable etc., of the MAP operating at full speed in real-life conditions and at a point in its program externally selectable by the operator. The same can be done repeatedly, storing the "film" of the observations in a file to examine them at a lower rate.

(c) As delivered by CSPI, SNAP appears as a good macro language for signal processing. The I/O is powerful but still requires that the user handles the data traffic and peripheral control in the host if the MAP needs to access them. We are currently preparing a new software that will allow the buffered access to all the host peripherals using very simple calls in SNAP. In this way the user will be able to program almost exclusively in an extended SNAP, the host becoming from the user's point of view essentially a transparent peripheral controller.

2.6 The Tracking/Display Subsystem

In a number of applications further processing between the AP and the display is required: tracking, plotting programs, sorting of data. For this processing the TD subsystem may, according to the application, run in a memory partition of the control CPU or have its independent small computer. The data are passed from the AP to the TD through a 20 Megabyte disc. In this way the output processing is well decoupled and this is specially useful when the experiment monitoring requires operator's interaction at the display level.

WARP's visual output subsystem is presently oriented towards two rather different peripherals: a high resolution video colour console, and a dot-matrix graphic lineprinter. The colour system is aimed at pseudocolour displays of physical data and at sonar console applications, whereas the graphic dot printer is especially useful for multi-curve displays and pseudo grey-tone representations; it is also quite cheap to operate for large productions and allows for one inexpensive device for both graphis and lineprinting.

2.6.1 The Graphic Printer

It presently uses a Printronix 300 line/minute dot matrix lineprinter in the graphic mode, a device somewhat comparable to a Versatec printer/plotter. It has, however, less resolution but is cheaper and uses normal paper instead of the special paper required by the Versatec. The software developed at SACLANTCEN is oriented towards the problem of plotting many curves simultaneously in several different ways for sonar, propagation and sea-floor studies; also the control of the subsystem has been made quite easy to reduce development time and increase flexibility during experiments. As this subsystem is described in detail in another paper [4], no more will be said here about it. An example of a typical output is shown in Fig. 7.

2.6.2 The Colour Station

Colour can be most useful to represent data which are functions of two or more variables. Black and white representation using contour lines with figures are accurate, but can be lengthy to examine, and it is often difficult to associate mentally different areas of equal level, which is easier with colour. Grey tones can be used too, but we usually found it unsatisfactory with representations of scientific data, like spreading functions or propagation loss versus range and frequency, one possible exception being remote sensing images. We had good results with colour in a number of cases, including running spectra, spreading functions and remote sensing.

Initially, WARP used a simple system with three planes of 256 x 256 pixels but has just now received a LEXIDATA 3400 with 10 planes of 640 x 512 pixels and two overlay planes [Fig. 8]. It includes black and white, and colour look-up tables, blinking, hardware zoom and track ball. The 10-bit resolution is required mainly for remote sensing applications. The display is driven from a separate terminal on a slave 21MX CPU or on the 21MX of the control subsystem according to the configuration.

A major problem with colour is how to obtain hard copies. We are presently using small and large (8" x 10") polaroids but we are completing a software to store images on magnetic tape for playback on the APPLICON colour jet-plotter system, on the Centre's UNIVAC 1106. We expect to see by next year a new cheaper model designed to operate directly from a colour raster scan display like the LEXIDATA 3400, and which could be taken on board during sea trials.

2.6.3 Software

Both display subsystems are supported by software (currently still partially under development) which includes:

(a) The retrieval of the data from the buffer disc following defined procedures, or under the operator's control.

(b) Facilities for easier preparation of the plot or image.

2.7 The Control Subsystem

The control subsystem is not a physical entity but corresponds to various program loading, data communication and program control procedures in the overall system. It encompasses:

(a) System Loading and Start-up

Each mode of operation in WARP is defined by a set of programs and parameter files operating in the subsystems. The loading is done using a "transfer file" i.e., a file that contains commands to be executed by the file manager [Fig. 9]. The transfer file is usually given a name related to the mode of operation.

(b) Parameter Files and Program Control

All programs in the AP, BF and TD subsystems run from well defined ASCII parameter files. Figure 10 shows such a file for the MCPL (Multi-Channel Plotting Program). A key feature is that each line starts with a two-letter code identifying the program parameter(s) concerned by the numerical value(s) on the line. Two asteriks indicate a "comment" line. As the programs use default values, only these parameters that need modifications must be present in the file. The parameter files can be edited using a standard editor, or a special one to handle tables more conveniently. This type of parameter file has proven very useful and is now completely adopted in all our systems.

(c) Program-to-Program Communication

The BF, AP and TD subsystems operate with independent programs. Information on the data such as sampling frequency, number of channels, scale factor, etc., is passed between these programs through a "mail-box" system; a program can send a message to "system available memory", where another program can check for its presence and get it. The implementation is conveniently done using RT IV's class I/O. If the programs run on different CPU's the message are sent through the DS1000 links.

(d) Operator Control

On a small system where one CPU only can support the BF, AP and display, one terminal can often support the RT IV system control and the operator's dialogue with specific programs. In certain cases it is more convenient to separate the two functions using two terminals. This is usually so when the TD subsystem requires much operator interaction and includes its own CPU.

3. LOGISTICS AND RELIABILITY OF OPERATION

During the more important sea trials, a full set of spares is available on board, which includes:

One 21MX CPU (minimal memory)
One 7906 disc
One MAP 300 (minimum configuration)
One full set of BF boards (except back plane)

In addition to this, the system has a great capacity for reconfiguration to face the breakdown of a CPU, disc and of minor peripherals.

Back in the laboratory, and often at sea, the spare equipment is usually operated as hot spares in smaller systems. Except for the BF, another complete system can be operated on shore while the first system is at sea. Spares on shore are not available for the AP as there is more time available for repair and as the jobs are usually less critical.

The BF has proven entirely reliable over almost one year. The MAP has had few breakdowns: two power supplies and one ROM chip. The various HP 21MX and 7906 discs gave some problems in the early part of their lives.

CONCLUSIONS

We have described a system designed for high-speed signal processing where the objective has been to simplify drastically the programming without any loss of throughput and of generality within the application field concerned. Experience with the system over several applications has proved the validity of the concepts proposed.

Work is continuing to bring the system to the level where it can be put in less experienced hands, and to complement its basic set of signal processing and display/tracking modules.

REFERENCES

1(a) SEYNAEVE, R. and HOPFORD, P. Interactive signal processing with a mini-computer system, presented at DATAFAIR, U.K., 1973.

1(b) ITSA Manual, Real-Time Systems Department, SACLANTCEN La Spezia, Italy, 1977.

2(a) BARBAGELATA, A., BONI, P., DIESS, B. Operation Manual for SPADA, SACLANTCEN SM-80, La Spezia, Italy, SACLANT ASW Research Centre, 1972.

2(b) SPADA Operator's Manual, Real-Time Systems Department, SACLANTCEN, La Spezia, Italy, 1977.

3. NESFIELD, P., Experience with a MAP 300 array processor, In: "Working Level Conference on Real-Time, General-Purpose, High-Speed Signal Processing Systems for Underwater Research", SACLANTCEN CP-25, La Spezia, Italy, SACLANT ASW Research Centre, 1979.

4. McCANN, M.J., Graphics for real-time signal processing systems, In: "Working Level Conference on Real-Time General-Purpose, High Speed Signal Processing Systems for Underwater Research", SACLANTCEN CP-25, La Spezia, Italy, SACLANT ASW Research Centre, 1979.

DISCUSSION

H. Urban Is the MAP system now working properly, or are there still software and hardware errors in the system?

R. Seynaeve We know in great detail of the problems encountered by CSPI with their first MAPS. Our first unit was delivered in December 1977 after CSPI had moved to the multiwire technology, which seems to have cured all their interference problems. Our first unit has performed very reliably over 1½ years. Due to combined hardware/software reasons the MAP did not run at full speed when delivered and we accepted it with reservations, but CSPI has since corrected this with appropriate upgradings. The software is excellent and has very few errors. More details are given in another paper by P. Nesfield.

D. Nairn What language is the system basically written in?

R. Seynaeve The system is written in HP assembler and Fortran.

D. Nairn Do buffers between the subsystems cause a speed penalty?

R. Seynaeve The buffers between the subsystems do not introduce any speed penalty, but only a slight delay as the system is basically a pipeline.

H.J. Alker Was it necessary to modify the RTE-IV operative system for including the multi-processor concept?

R. Seynaeve No, the system uses a standard RTE-IV with DS100 (the HP distributed system software). We have developed some extra system software: for example, drivers and EMA support routines.

H.J. Alker What has been done for monitoring in real-time hardware and software errors? And, was the system described limited by not having such a monitor system available?

R. Seynaeve We feel we should include run-time automatic tests of the whole system, but this requires time to develop. The hardware, especially the beamformer and the MAP 300 have been very reliable. Also, at this stage, when in operation the systems are usually attended and therefore this problem has not yet been important.

W.G. Wagner Can the array processor be shared by several users concurrently?

R. Seynaeve For normal processing, we do not consider sharing the array processor between several users. Most of the applications where we use array processors are computation intensive and, if anything, would require a faster array processor, so the system at present is basically single-user for processing. For development and testing, the situation is quite different and the array processor is usually mostly idling; we are, therefore working on a scheme where the array processor will be available to several users on a multi-terminal system (eventually as a distributed system under DS1000) but to one user at a given time, and for a short period only (a few seconds or less).

D. Steiger Please comment on the software used for the colour terminal — in particular the graphics and RTE driver.

R. Seynaeve The Lexidata 3400 system includes a number of commands like zooming, scrolling, loading of look-up tables, etc., and this set is currently being complemented to meet our own requirements. The user will access the system through several packages: one for general purpose handling of pseudo-colour representations with contour lines overlayed, the other for sonar console type of applications. We find that the availability of EMA in the 21MX allows the use of large graphic and contouring packages quite conveniently and, in order to standardize with the software used on our UNIVAC 1106 installation, we intend to adapt the Applicon colour plotter software the HP 21MX, unless Applicon can supply it already converted. The Lexidata RTE driver was supplied with the display, it is a DMA driver and works very well.

WARP SYSTEM FUNCTIONAL DIAGRAM

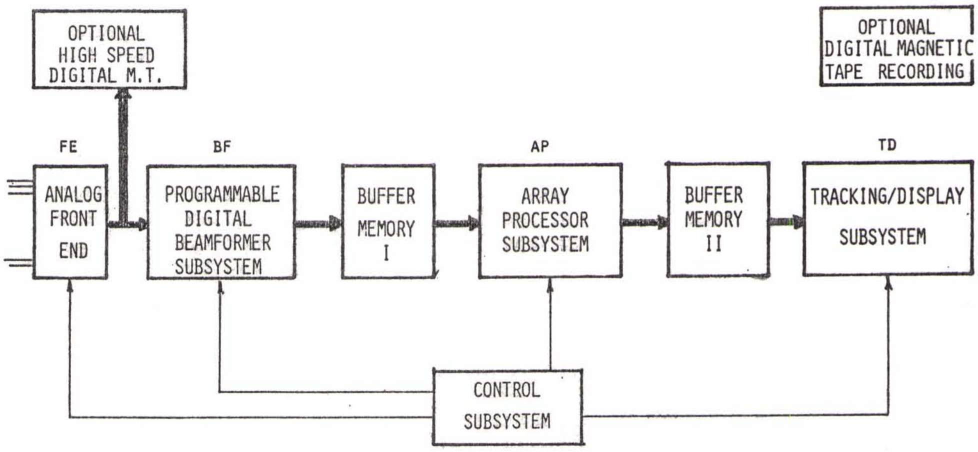

OPTIONAL HIGH SPEED DIGITAL M.T.
OPTIONAL DIGITAL MAGNETIC TAPE RECORDING
FE
BF
AP
TD
ANALOG FRONT END
PROGRAMMABLE DIGITAL BEAMFORMER SUBSYSTEM
BUFFER MEMORY I
ARRAY PROCESSOR SUBSYSTEM
BUFFER MEMORY II
TRACKING/DISPLAY SUBSYSTEM
CONTROL SUBSYSTEM

FIG. 1

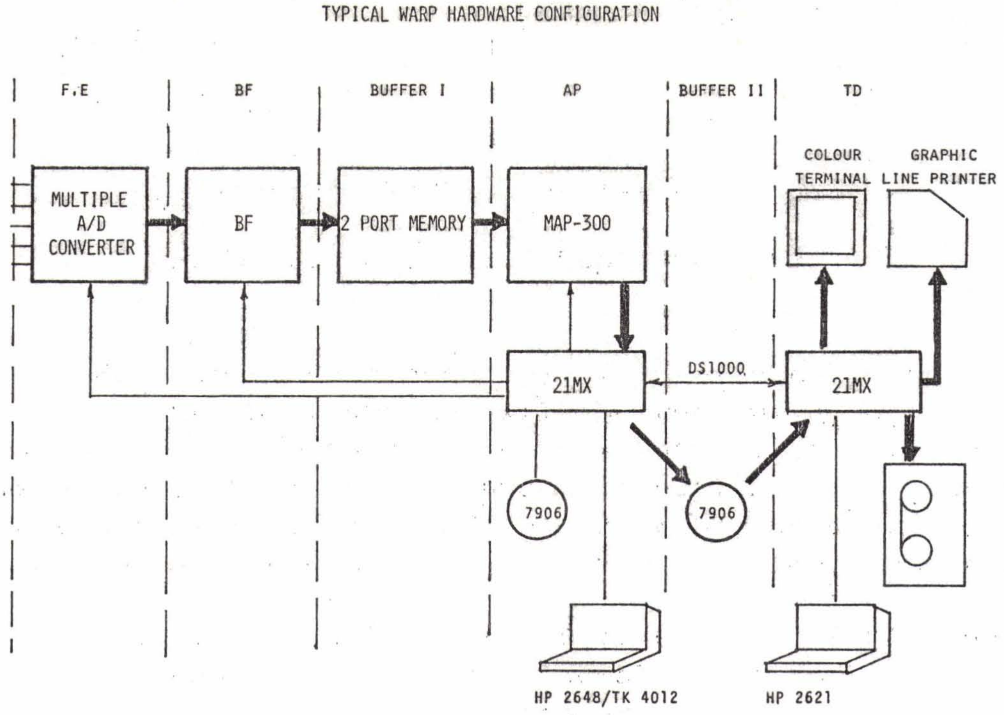

TYPICAL WARP HARDWARE CONFIGURATION
F.E
BF
BUFFER I
AP
BUFFER II
TD
MULTIPLE A/D CONVERTER
BF
2 PORT MEMORY
MAP-300
COLOUR TERMINAL
GRAPHIC LINE PRINTER
21MX
DS1000
21MX
7906
7906
HP 2648/TK 4012
HP 2621

FIG. 2

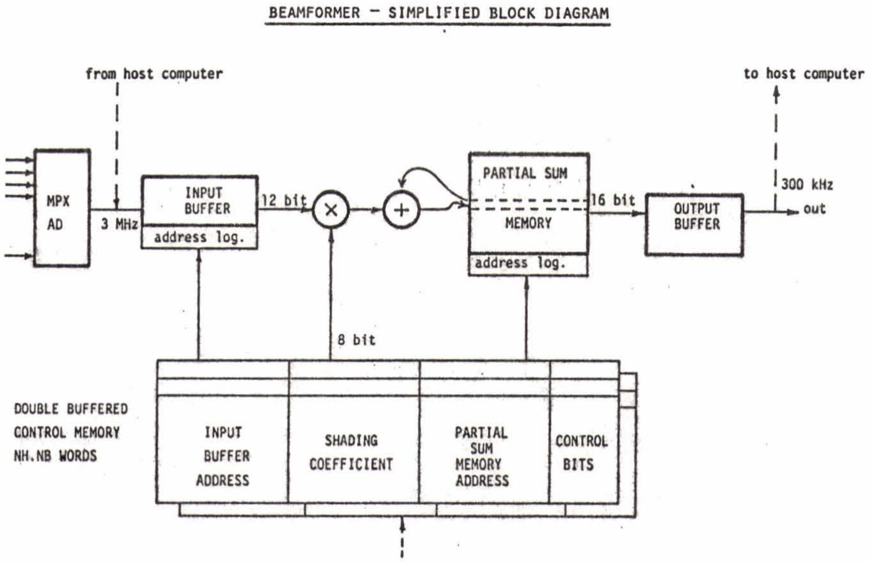

BEAMFORMER – SIMPLIFIED BLOCK DIAGRAM
from host computer
to host computer
MPX AD
3 MHz
INPUT BUFFER
address log.
12 bit
PARTIAL SUM MEMORY
address log.
16 bit
OUTPUT BUFFER
300 kHz
out
8 bit
DOUBLE BUFFERED CONTROL MEMORY NH.NB WORDS
INPUT BUFFER ADDRESS
SHADING COEFFICIENT
PARTIAL SUM MEMORY ADDRESS
CONTROL BITS

FIG. 3

SACLANTCEN BEAMFORMER

CONSTRAINTS ON N_H, N_B, f_i, f_o

ARITHMETIC SPEED	:	$N_H N_B f_o \leq 3$ MHz
INPUT RATE *	:	$N_H f_i \leq 3$ MHz
OUTPUT RATE	:	$N_B f_o \leq 300$ kHz
CONTROL MEMORY SIZE	:	$N_H N_B \leq 2048$
MAXIMUM N_H *	:	$N_H \leq 255$
MAXIMUM N_B	:	$N_B \leq 255$

* N_HMAX, f_iMAX ARE LIMITED BY THE PRESENT A/D CONVERTER.

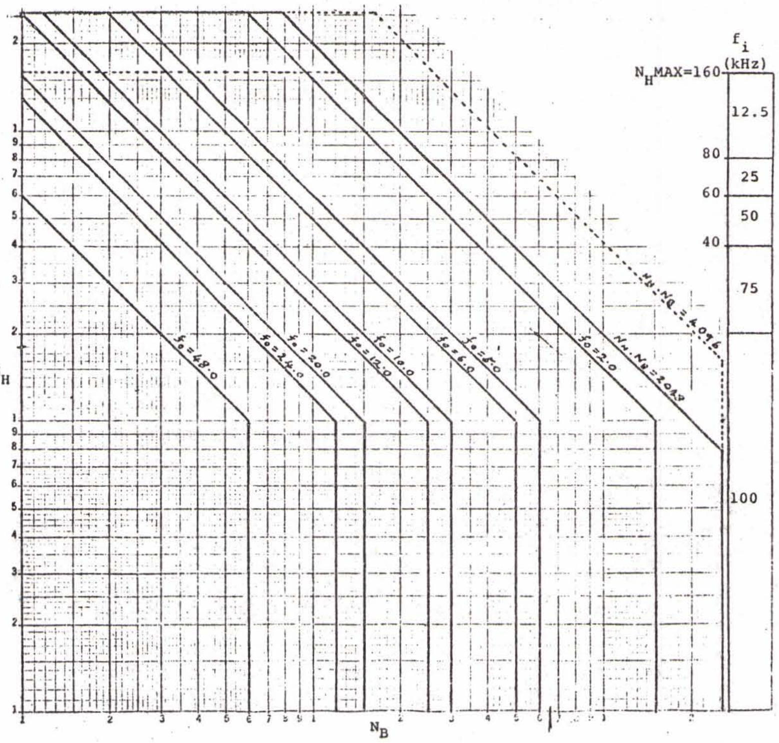


FIG. 4

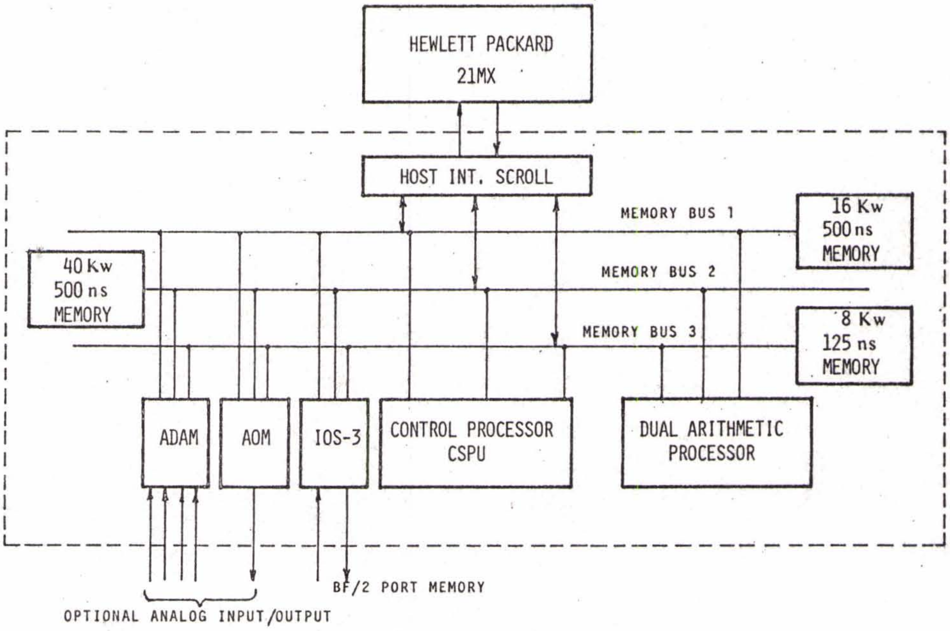


FIG. 5

SIMPLIFIED DIAGRAM OF MAP300 ARITHMETIC UNIT

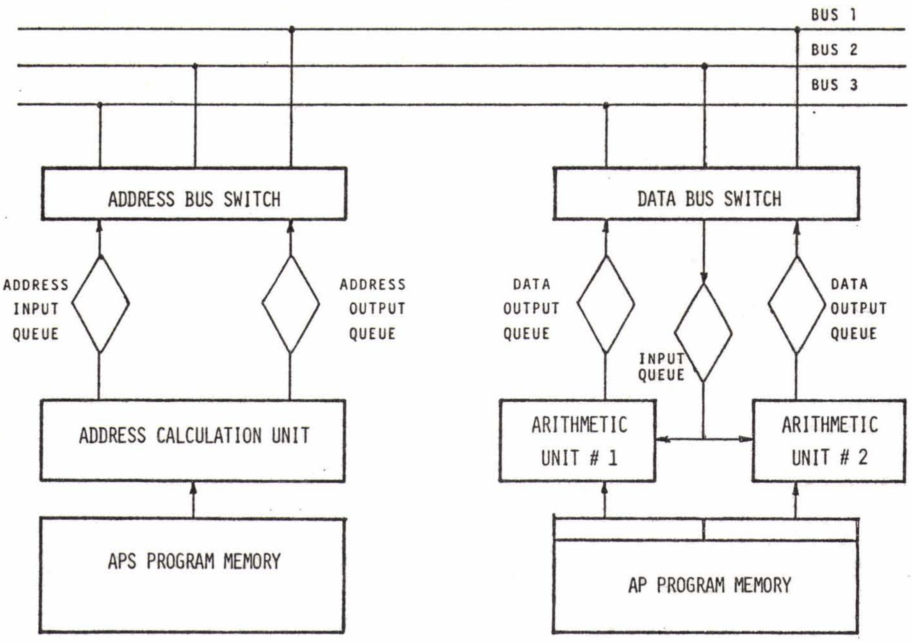


FIG. 6

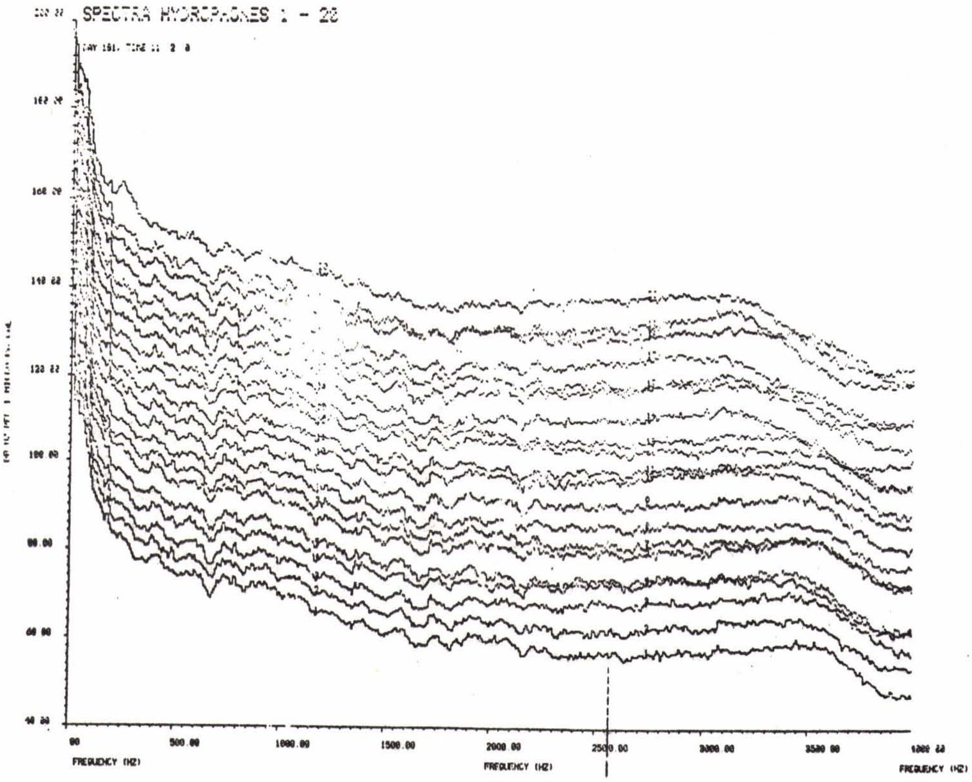


FIG. 7

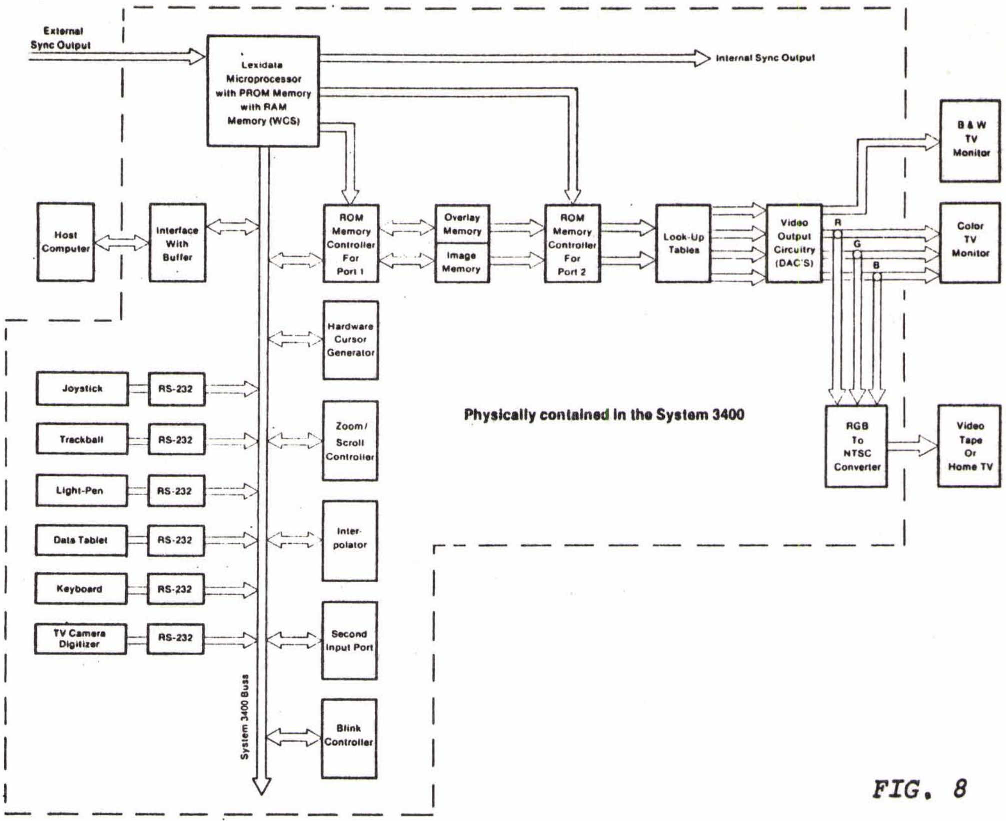

External Sync Output
Lexidata Microprocessor with PROM Memory with RAM Memory (WCS)
Internal Sync Output
B & W TV Monitor
Host Computer
Interface With Buffer
ROM Memory Controller For Port 1
Overlay Memory
Image Memory
ROM Memory Controller For Port 2
Look-Up Tables
Video Output Circuitry (DAC'S)
R
G
B
Color TV Monitor
Hardware Cursor Generator
Joystick
Trackball
Light-Pen
Data Tablet
Keyboard
TV Camera Digitizer
RS-232
RS-232
RS-232
RS-232
RS-232
RS-232
Zoom/ Scroll Controller
Physically contained in the System 3400
RGB To NTSC Converter
Video Tape Or Home TV
Inter-polator
Second Input Port
Blink Controller
System 3400 Buss

FIG. 8

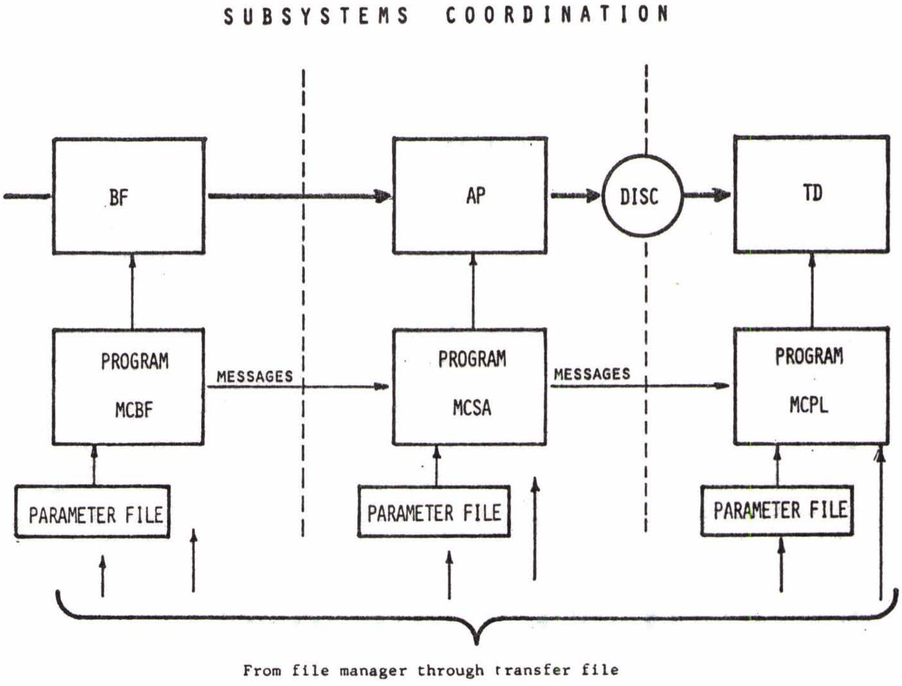
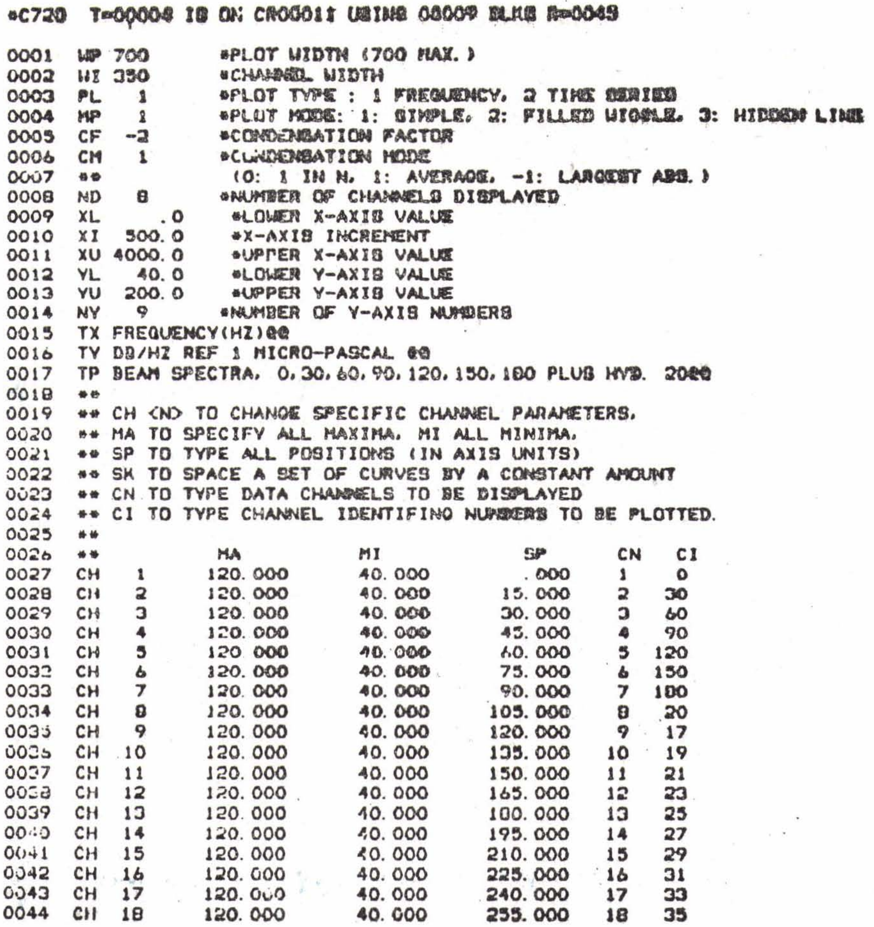

SUBSYSTEMS COORDINATION
BF
AP
DISC
TD
PROGRAM MCBF
MESSAGES
PROGRAM MCSA
MESSAGES
PROGRAM MCPL
PARAMETER FILE
PARAMETER FILE
PARAMETER FILE
From file manager through transfer file

FIG. 9



```
*C720  T=00008 IS ON CR00011 USING 00009 BLKS R=0049

0001  WP 700        *PLOT WIDTH (700 MAX.)
0002  WI 350        *CHANNEL WIDTH
0003  PL   1        *PLOT TYPE : 1 FREQUENCY, 2 TIME SERIES
0004  MP   1        *PLOT MODE: 1: SIMPLE, 2: FILLED WIGGLE, 3: HIDDEN LINE
0005  CF  -2        *CONDENSATION FACTOR
0006  CM   1        *CONDENSATION MODE
0007  **             (0: 1 IN N, 1: AVERAGE, -1: LARGEST ABS.)
0008  ND   8        *NUMBER OF CHANNELS DISPLAYED
0009  XL      .0     *LOWER X-AXIS VALUE
0010  XI  500.0      *X-AXIS INCREMENT
0011  XU 4000.0      *UPPER X-AXIS VALUE
0012  YL   40.0      *LOWER Y-AXIS VALUE
0013  YU  200.0      *UPPER Y-AXIS VALUE
0014  NY   9        *NUMBER OF Y-AXIS NUMBERS
0015  TX FREQUENCY(HZ)@@
0016  TY DB/HZ REF 1 MICRO-PASCAL @@
0017  TP BEAM SPECTRA, 0,30,60,90,120,150,180 PLUS HYD. 20@@
0018  **
0019  ** CH <N> TO CHANGE SPECIFIC CHANNEL PARAMETERS,
0020  ** MA TO SPECIFY ALL MAXIMA, MI ALL MINIMA,
0021  ** SP TO TYPE ALL POSITIONS (IN AXIS UNITS)
0022  ** SK TO SPACE A SET OF CURVES BY A CONSTANT AMOUNT
0023  ** CN TO TYPE DATA CHANNELS TO BE DISPLAYED
0024  ** CI TO TYPE CHANNEL IDENTIFING NUMBERS TO BE PLOTTED.
0025  **
0026  **           MA           MI             SP    CN   CI
0027  CH   1     120.000      40.000          .000    1    0
0028  CH   2     120.000      40.000        15.000    2   30
0029  CH   3     120.000      40.000        30.000    3   60
0030  CH   4     120.000      40.000        45.000    4   90
0031  CH   5     120.000      40.000        60.000    5  120
0032  CH   6     120.000      40.000        75.000    6  150
0033  CH   7     120.000      40.000        90.000    7  180
0034  CH   8     120.000      40.000       105.000    8   20
0035  CH   9     120.000      40.000       120.000    9   17
0036  CH  10     120.000      40.000       135.000   10   19
0037  CH  11     120.000      40.000       150.000   11   21
0038  CH  12     120.000      40.000       165.000   12   23
0039  CH  13     120.000      40.000       180.000   13   25
0040  CH  14     120.000      40.000       195.000   14   27
0041  CH  15     120.000      40.000       210.000   15   29
0042  CH  16     120.000      40.000       225.000   16   31
0043  CH  17     120.000      40.000       240.000   17   33
0044  CH  18     120.000      40.000       255.000   18   35
```

FIG. 10

NOSC SIGNAL PROCESSING EVALUATION LABORATORY

by

Bryson Pennoyer
NAVAL OCEAN SYSTEMS CENTER
San Diego, California, U.S.A.

ABSTRACT A new acoustic Signal Processing Evaluation Laboratory (SPEL) at the Naval Ocean Systems Center (NOSC), San Diego, is described. This laboratory has been designed to support a broad program of research and development of passive and active sonar technology. Full integration of the system allows an analyst to go from multichannel analog tape recorded signals through flexible computer controlled digitization and signal processing such as beamforming, spectral analysis, filtering and detection ending with a high resolution CRT display of the output, all under the control of a central host computer.

The system was designed to provide research analysts an efficient and low cost means to develop and evaluate signal processing concepts, to develop full scale system emulations or simulations and to conduct controlled tests of alternatives through the use of real and simulated data. In addition, the system provides a controlled framework for conducting evaluations of candidate programmable signal processors using real and simulated data. To meet these objectives, the system design capitalizes on the power of new array processor technology and on advanced software concepts to provide the high processing throughput and capability to implement complex processes at a low cost. Combining this technology under the control of a flexible and powerful host computer provides the analyst with an integrated signal processing system.

The principle subsystems of SPEL are the computer controlled 64 channel analog preprocessor, two high-speed signal processors, a PDP-11/7Ø An overall system design philosophy is presented, and the details of host computer with hardcopy and color and/or black and white CRT displays. the system and subsystems which realize the design philosophy are given.

INTRODUCTION

For the last 1Ø years, analysts and researchers in the areas of signal processing and display at NOSC have utilized four separate laboratories to conduct their investigations. These were the Acoustic Signal Data Analysis and Conversion System (ASDACS), the Data Acquisition Playback and Analysis (DAPAAN) System, the Display Research Laboratory and the UNIVAC 11Ø8 Computer Center. Each of these systems provided excellent capabilities for the specific tasks for which they were designed, yet had limitations which were a result of the technology of the time. The major deficiency was the lack of integration of the system's capabilities which greatly reduced its efficiency and increased its costs as a result of multiple programming requirements.

Present day signal processing and display requirements benefit greatly from an integrated system which is efficient and provides a low cost means to develop and evaluate signal processing and display concepts. Also needed are capabilities to develop full scale system emulations or simulations and the ability to conduct controlled tests of alternative processing concepts through the use of real and simulated data. Such a system should provide a controlled framework for conducting evaluations of new or alternate programmable signal processors.

The Signal Processing Evaluation Laboratory(SPEL) was designed to meet these objectives. SPEL utilizes the latest technology in signal processing and display equipment and is designed to be expandable to meet future needs.

1. SPEL System Description

As shown in Figure 1, SPEL consists of a unique combination of computer based subsystems that provide analog preprocessing including analog-to-digital conversion, secure ARPANET communications, high-speed signal processing and display research. These subsystems are coupled with a flexible digital computer under the control of multiple users. Data, signal and display processing are easily performed as a result of system integration, a large software library and a broad set of peripheral devices.

1.1 Secure ARPANET Subsystem

The SPEL host computer is a limited server host on the secure ARPANET. The ARPANET is a resource sharing, host-to-host network linking a wide variety of computers at research centers sponsored by Defense Advanced Research Projects Agency (DARPA) and other activities in the continental United States, Hawaii, Norway and England. The secure ARPANET operates on the regular ARPANET using secure message encryption. This secure network links, among others, the NOSC SPEL facility with the Acoustic Research Center (ARC) at Moffett Field, other NOSC facilities and the Naval Research Laboratory in Washington, D.C.

The SPEL link to the ARC offers users full ARPANET protocols including remote terminal operation (TELNET), file transfer (FTP) and message communications (MAILER). Users, both contractor and Navy, have full access to the ARC PDP-1Ø and hence to the full ARC capabilities via the control routines resident in the PDP-1Ø.

Real data from the ARC data acquisition systems is available on-line to the SPEL facility via transfer of ARC PDP-1Ø data files. This data can be used to validate processing concepts and systems. Joint experiments, which pool the resources of the ARC, SPEL and others on the secure ARPANET can be conducted. SPEL users will be able to conduct experiments at the ARC with the coordination of a test director.

1.2 Analog Preprocessing Subsystem

Preprocessing capabilities include very flexible multi-channel demultiplexing, amplification and filtering of analog signals, as well as programmable analog-to-digital (A/D) conversion of up to 64 channels. The digital data can be immediately processed in the Raytheon Advanced Digital Signal Processor (ADSP) or temporarily stored on a 3ØØ megabyte random access disk for later transfer to 9 track digital tape or the host computer. Analog-to-digital conversion is performed under computer control using IRIG time code data for starting, stopping and searching the analog tape. Initially, the programmable sections of the system will be manually set. Computer control will be provided in the future. A block diagram of the system is shown in Figure 2.

A wide variety of analog recording formats can be accommodated in the system including IRIG Wideband I, II and III, VIDAR, DAPAAN 9 channel/track and DAPAAN 18 channel/track. Once the analog signal is demultiplexed, it is amplified, anti-alias filtered by a six-pole six-zero elliptic filter and then digitized using one independent programmable A/D converter per channel. Each A/D converter is capable of a 1ØØ kHz conversion rate with 8 or 12 bit resolution. However, the system has a maximum throughput of 5ØØ kHz when the digitized output goes to the ADSP or disk and 5Ø kHz using magnetic tape. Variations in sample rate caused by tape speed variations are eliminated by using tape recorded reference signals for speed control and for deriving the A/D sample rate.

On-line data analysis ,quality checks, can be made using spectral analysis, crosscorrelation, signal amplitude, and aural tests of the signals to be digitized and are included in the system to improve accuracy and confidence in the converted data.

Digital-to-analog conversion capabilities for up to 28 channels are provided with filtering, amplification and multiplexing of analog data. Digital data input can come from either the 3ØØ megabyte disk or 9 track digital tape. This capability coupled with the power of the signal processors, provides a means of generating analog tapes with known characteristics.

1.3 Signal Processing Subsystem

This subsystem provides the real processing power of SPEL with the extensive signal processing cabablilities of the Raytheon Advanced Digital Signal Processor (ADSP) and the Signal Processing Systems SPS-81. Each processor is programmable by the user through the host computer. These processors provide the important basic signal processing functions of Fast Fourier Transform, digital filtering, quadrature analysis, beamforming, convolution and correlation plus many others. Functions can be used separately for analysis purposes or combined into a total processing system.

SPEL has been designed to readily accommodate additional programmable signal processors by connecting them to the PDP-11/7Ø. This capability can be used for conducting performance tests on selected signal processors or to provide specialized processing capabilities.

1.3.1 Raytheon ADSP

The ADSP is the primary signal processor in the system. It consists of three major components, as shown in Figure 3; 1) the Bus Access Unit to interface external devices to the processor, 2) the Main Memory for program and data storage and 3) the Signal Processing Unit for performing high-speed "number crunching". The ADSP is designed to provide high-speed signal processing and minimize programming costs by providing the following major features:

1. Greatly simplified programming through the elimination of microprogramming and the use of a unique method of dynamically describing data items. Data are characterized via data descriptor (DD) words which contain pertinent information for each variable. The major consequence of using DDs is the separation of structual details of data, i.e. scalar, matrix, real or complex, from the application program. Thus, modifications in data types can be made by simply altering the data descriptors, leaving the application program untouched.

2. A large, high-level, mathematically-oriented signal processing instruction set is incorporated. Future expansion of the instruction set is available through a writable control store memory. Typical instructions are "FFT", which computes the discrete Fourier Transform and "IIR", which performs recursive filtering on a set of data.

3. High speed processing is provided through the combination of a 75 nanosecond system clock and pipelined implementation of data accessing, data formatting and arithmetic computations. In addition, separate buses are provided between memory and the signal processing unit for commands, instructions, and data. The arithmetic is block-floating point (16 bit) two's complement with automatic data scaling. The ADSP can perform 27M multiplies and 53M adds per second simultaneously. A complex

1024 point FFT takes only 0.84 milliseconds, which includes the application of a data weighting window and bit reversal.

4. A large capacity main memory of 256K 32-bit words, with addressable expansion up to 16 million words, provides the capability of processing relatively large data arrays in memory. Virtual addressing is used to provide easy relocation and context switching.

5. Host computer control is available to allow for execution of the program one step at a time or execution of the total program.

User programs are generated through the use of the signal processing assembler (SPASM) which converts source-level ADSP instructions and data descriptors (DD) to object form. SPASM assigns data to main memory locations and provides linkages from instructions to DDs and from DDs to data.

1.3.2 SPS-81

The SPS-81 is a moderate speed multiprocessor computer designed for signal processing applications. It is composed of an arithmetic processor(AP), an input/output processor (IOP) and a bulk memory of 128K 16-bit words, as shown in Figure 4. The AP handles complex arithmetic and can do 8M multiplies and 12M additions per second simultaneously. The arithmetic is fixed-point (16 bit) two's complement. A complex 1024 point FFT with dynamic scaling of the data typically takes 14 milliseconds.

The SPS-81 is used primarily by means of FORTRAN callable subroutines. Since the machine is entirely microprogrammed, each of these routines may be thought of as causing a load of the appropriate microcode, transmission of data and activation of the SPS-81.

The AP and IOP are separately microprogrammed. Microprogram generation is handled through a cross-assembler called SPASM (not the same as for the ADSP). A utility program called SPUD is available to facilitate program checkout. The resulting microprogram can be initiated by a FORTRAN application routine.

1.4 Host Computer Subsystem

The host computer, a Digital Equipment Corporation PDP-11/70 and peripherals, as shown in Figure 5, is the focal point of SPEL and provides multi-user interface and control of the integrated system. A user may proceed from program development and testing to input of data, signal processing, analysis and display using a common set of software and hardware tools.

Computational speed is attained through the use of the floating point processor and the 640 kilobytes of high speed (500 nanosecond) MOS memory. The floating point processor provides single and double precision (32 or 64 bit) floating point modes to increase computational accuracy. A double precision multiplication is performed in 3 microseconds. However, the real computational power of the system is provided by the SPS-81 and Raytheon ADSP signal processors.

Two 3ØØ megabyte disks provide extensive storage for on-line data, programs and the operating system. Additional on-line data storage is available through the use of a 3ØØ megabyte disk which interfaces with the analog preprocessor subsystem. Long term data and program storage is available using either 9 track 8ØØ/16ØØ bpi or 7 track 556/8ØØbpi magnetic tape.

The operating system for the PDP-11/7Ø is the Programmer's Workbench (PWB)/UNIX system developed by Bell Laboratories. The UNIX time-sharing system is a general-purpose, multi-user, interactive operating system specifically engineered to make the designer's, programmer's, and documenter's computing environment simple, efficient, flexible, and productive. UNIX contains such features as: 1) a hierarchical file system, 2) a flexible, easy-to-use command language, 3) ability to execute sequential, asynchronous and background processes, 4) a powerful context editor, 5) very flexible documentation preparation and text processing system, 6) a high-level programming language called 'C' which is conducive to structured programming, 7) the programming languages BASIC, FORTRAN IV+ and PASCAL, 8) symbolic debugging systems, 9) a variety of system programming tools (e,g. compiler-compilers) and 1Ø) ease of developing device handlers.

The file system consists of a highly-uniform set of directories and files arranged in a tree-like hierarchial structure. Some of the features are: 1) simple and consistent naming conventions, 2)mountable and de-mountable file systems and volumes, 3)file linking across directories, 4)automatic file space allocation and de-allocation that is invisible to the user, 5)a complete set of flexible directory and file protection modes and 6) each physical I/O device, from interactive terminals to main memory, is treated like a file allowing uniform file and device I/O.

1.5 Display Research Subsystem

The display research subsystem is a user oriented laboratory for the analysis and display of digital and video imagery. The laboratory serves as a test site for the development and evaluation of advanced processing and display concepts. In addition, the laboratory, with its controlled environment, is a major resource for human factors research on display formats and perceptual processes.

The laboratory features the high resolution CRT displays and associated processing of the RAMTEK 94ØØ Display System. This system provides two black and white CRT monitors with 1Ø24x1Ø24 point resolution and 256 levels of intensity and a color monitor with 75Øx75Ø point resolution, 4Ø96 colors and 256 levels of intensity. Each monitor can display vectors, conics, bargraphs and raster data in addition to performing pan, zoom, scroll and blink functions. An extensive user library is available for performing image transformations and image processing.

SUMMARY

SPEL is now available at NOSC for the processing of large quantities of analog and digital data. The system can provide real on-line digital data from the Acoustic Research Center by way of the secure ARPANET. Digitized data are also available from an extensive analog library via the analog preprocessor subsystem of SPEL. Signal processing functions are performed in one of two high speed programmable signal processors or in the host computer. Processed data can then be viewed on high resolution color and/or black and white CRT displays, a standard computer line printer or graphic plotter.

The system is intended to serve as a tool for signal and display analysts with special emphasis on ease-of-programming, program development aids, real data access and simulated data generation. It possesses the very high throughput required by the multi- channel, multi-resolution processing problems of future Navy fleet systems.

The system is designed to accommodate new signal processors readily and to provide the means for efficient comparisons of the performance of such devices.

DISCUSSION

F.W. Molino Please give a scenario of how the various signal processors, i.e. TI MVP, SPS-1, will be evaluated? What will you be looking for??

B. Pennoyer The techniques to be used in evaluating the signal processors is the subject of a contract this year. Bench-mark testing using a specific application program and controlled data set will certainly be part of the evaluation.

R. Seynaeve Does the ADSP have provisions to avoid losses of dynamic range after, for example, a multiplication, as the format used is 16-bit block floating point?

B. Pennoyer The multiplication of two 16-bit words results in a 32-bit result with an 8-bit exponent. The 16 most significant bits of the 32 are retained after correction for under or overflow.

R. Seynaeve What is the cost of the ADSP?

B. Pennoyer The cost for future copies of the ADSP has not been firmly established by Raytheon.

Y.S. Wu How long have you worked on this project and how many NOSC people involved?

B. Pennoyer Since 1977; the NOSC project team consists of six in-house people.

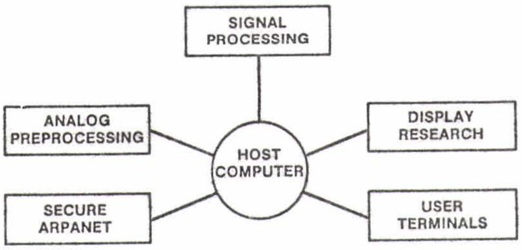


FIG. 1 SPEL CONFIGURATION

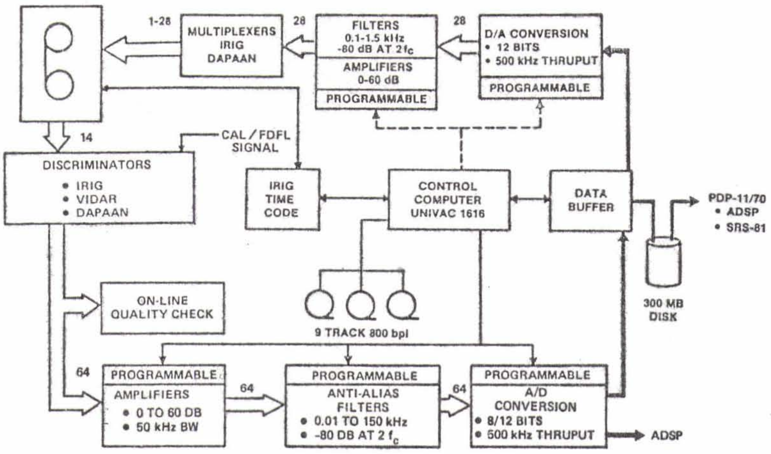


FIG. 2 ANALOG PREPROCESSOR SUBSYSTEM

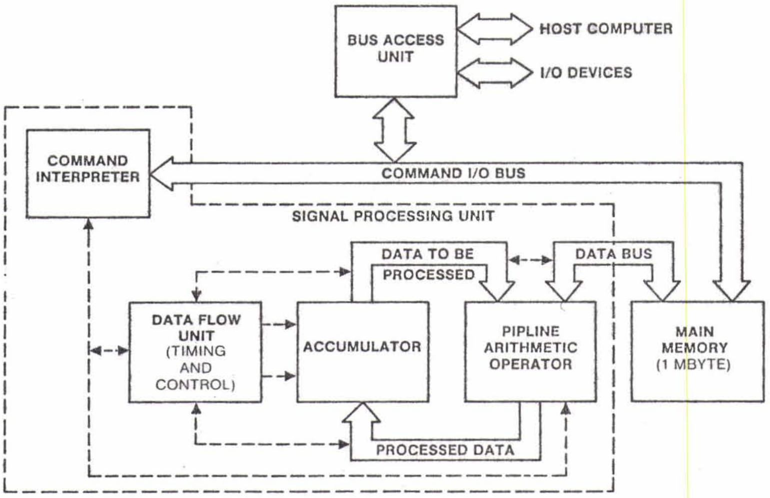


FIG. 3 RAYTHEON ADSP ARCHITECTURE

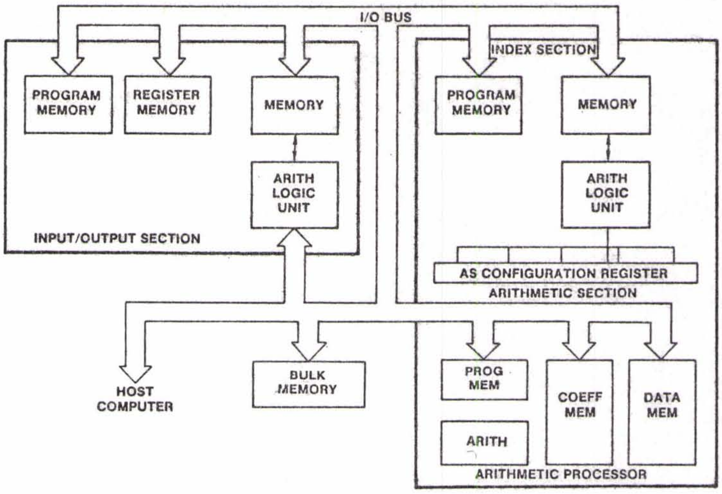


FIG. 4 SPS-81 ARCHITECTURE

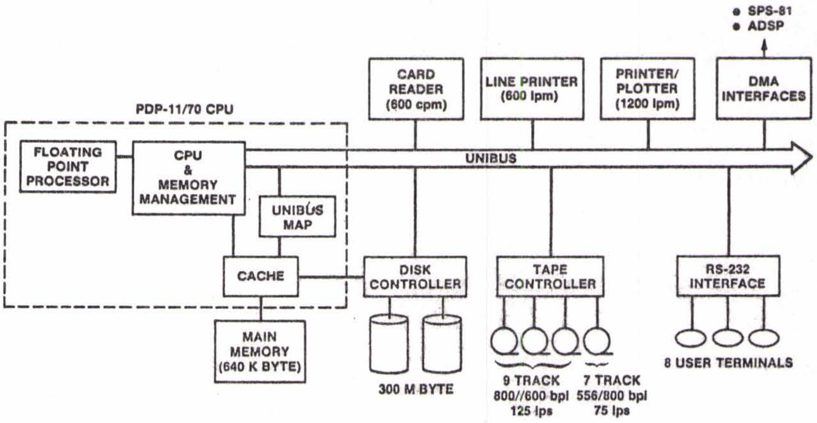


FIG. 5 HOST COMPUTER SUBSYSTEM

USING INTELLIGENT GRAPHICS TERMINALS IN REAL-TIME PROCESSING

by

Daniel Steiger
Naval Research Laboratory
Washington, DC

ABSTRACT At the Naval Research Laboratory intelligent graphics terminals are being utilized for real-time acquisition, storage, processing and display of oceanographic data when mini-computer systems are inappropriate. The intelligent graphics terminal is a well packaged, low cost, reliable and highly portable microcomputer system capable of performing many of the functions required of mini-computer systems. Discussed in the paper are methods of configuring, implementing and utilizing the intelligent graphics terminal for real-time oceanographic processing.

INTRODUCTION

Mini-computer systems have to a large extent successfully fulfilled the oceanographic researchers requirements for data collection, processing and storage in the field and in the laboratory. At NRL mini-computer systems are placed aboard ships, aircraft and land based sites on a routine basis. It is expected that the mini-computer will continue to meet the needs of the researcher for a long time, since the hardware and software are being enhanced by the various manufacturers at a very rapid pace and more applications are continually being computerized.

At NRL there are frequently field experiments conducted where the mini-computer is inappropriate. These experiments are usually aboard aircraft where there is insufficient space, adequate power is not available, the experiment is "piggy backing" another experiment, or the manpower to ship, install and operate the mini-computer system can not be justified. NRL has found a suitable alternative and has developed real-time oceanographic data acquisition and processing systems using intelligent graphics terminals when mini-computers are inappropriate.

NRL considered the use of programmable calculators, commercially available microcomputers and a memory based mini-computer system before selecting the intelligent graphics terminal as the data acquisition and processing device. The terminal was selected because of its ability to be programmed in assembly language and BASIC, its highly portable nature, the built in cassette drives to store the programs and data, its programmable graphics capability, its low cost and when it's not replacing a mini-computer it makes a very desirable computer graphics terminal.

DESCRIPTION OF INTELLIGENT GRAPHICS TERMINAL

A brief definition of an intelligent graphics terminal for the purpose of this paper is, a terminal with a CRT display and keyboard that can be programmed to perform tasks outside its normal terminal function as well as provide graphical plotting functions.

The Hewlett Packard HP2647A intelligent graphics terminal was selected as the best terminal to replace the mini-computer system. The terminal is modularized and can be configured for data acquisition under assembly language programs where maximum speed and flexibility in programming and I/O (Input/Output) can be achieved or it can be configured to be programmed under BASIC using a compatible IEEE 488 standard interface for I/O.

Figure 1 are photographs of the HP264X terminal. The electronic circuit boards shown in the terminal can be easily removed and the terminal reconfigured for any application. The following features listed below make the terminal an excellent device for data acquisition, storage and display.

- 128K bytes of graphic, alphanumeric and programmable memory
- refresh type display
- dual unformatted cassette tape drives
- interface for 5 parallel 16 bit data words at TTL levels
- keyboard with function keys that permit paging under alphanumerics and cursor positioning with ZOOM IN and ZOOM OUT features
- weight of 43 pounds
- power required of 4 amps
- size of approximately 2.5 cubic feet
- can be programmed in an Intel 8080 compatible assembly language or BASIC
- internal clock that can be used when scheduling programs
- ability to utilize operating system software when programming
- supported software with terminal debugger/assembler and cross assembler for HP1000 mini-computers
- writes compatible HP1000 mini-computer cassette tapes
- worldwide service and parts
- can attach peripherals directly to terminal on video interface or IEEE 488 compatible interface

The less desirable features of the terminal are listed below.

- relatively small display for graphic output
- maximum 110K bytes of storage on a cassette
- operates at microcomputer speeds which are significantly slower than mini-computers

DISCUSSION

The Hewlett Packard intelligent terminal can be configured in many ways in order to suit the intended use. Reconfiguring a terminal is done by utilizing certain electronic boards within the terminal itself. There is room for fifteen electronic circuit boards in the terminal. Eight of the boards are standard control boards for keyboard, processor, memory control and cassette drive control. An additional board is required for display memory. Memory can range from 4K to 32K bytes on a single board. This leaves six empty I/O slots in the terminal to be configured based upon the application. Table 1 shows a comparison of the ways the terminal can be configured. For a standard alphanumeric terminal to a computer system an I/O interface card would be required in the terminal and the unit is referred to as an HP2645. For terminal and graphics applications on line to a computer, two additional graphic control boards are required, an additional memory control board and an I/O interface board to the computer. This unit is sold by Hewlett Packard as an HP2648 Graphics Terminal. For standalone computing the terminal is configured with graphics control, and four boards of ROM and RAM memory. This leaves two empty boards that can be used for and IEEE 488 compatible interface, a computer interface for switching from offline to online operations and a video interface board that permits direct copying to hard copy video devices such as the Tektronix 4632. The IEEE 488 interface can be connected to external devices with compatible outputs. Commercially available devices with compatible outputs are printers, plotters, DVM's and other test equipment. This intelligent graphics terminal is called an HP2647. Its primary use is for offline programming in BASIC and it supports an easy to use graphics software package that operates under BASIC. For assembly language terminal applications the HP2649 is offered. This terminal is usually configured with the standard eight control boards and two 32K byte RAM memory boards. A Debugger/Assembler and cross compiler that operates on the HP1000 is available for writing, compiling and debugging programs. With this configuration up to five additional boards may be added to the terminal.

Table 2 shows a comparison between the HP2647 and the HP2649 terminal capabilities. The major tradeoffs between the terminals in a standalone data acquisition and processing mode is the HP2647 is good for ease of programming, data computation, processing, plotting and for I/O using an IEEE 488 device; whereas the HP2649 is significantly faster for real-time data acquisition without the overhead of the BASIC interpreter, more flexible since bits can be easily manipulated and the I/O can support up to five sixteen bit parallel words.

TABLE 1

Comparison of HP264X Intelligent Terminal Configurations

	HP2645	HP2647	HP2648	HP2649
Standalone Data Acquisition	NO	YES	NO	YES
Graphics Hardware	NO	YES	YES	NO
Interface Normally Used	RS232	IEEE 488	RS232	8 Bit Duplex
Programming Language	NONE	BASIC	NONE	ASSEMBLY
Available I/O Slots	6	2	3	5
Major Use	Alphanumeric Computer Terminal	Standalone Computing and Graphics Terminal	Alphanumeric and Graphics Computer Terminal	Standalone Data Acquisition Processing Storage

TABLE 2

Comparison of HP2647 and HP2649 Capabilities

	HP2647	HP2649
Standalone Data Acquisition	Fair	Good
Ease of Programming	Good	Fair
Software Flexibility	Fair	Good
Graphics Hardware	Yes	No
Computational Capability	Good	Fair
Speed	Slow	Fast

DESCRIPTION OF AXBT INTELLIGENT TERMINAL SYSTEM

Acoustic studies are being performed by NRL and NAVOCEANO this year in areas where oceanographic eddy currents exist. NRL aircraft are being used to locate these eddies by dropping AXBT's in a pattern to define the temperature boundaries. These aircraft flights last for one or two days, occur approximately every two weeks and take place aboard different NRL aircraft.

For real-time data acquisition, processing and storage the intelligent graphics terminal was selected. The terminal was selected because of its highly portable nature, it can be installed and removed in minutes without the need of special equipment, it can be transported easily with personnel, small and compact in size and can be operated by the scientist with a small amount of training.

Two intelligent terminals were used for the experiments. One of the terminals was configured as an HP2649 for real-time data acquisition, processing and storage while the other terminal was configured as an HP2647 for computation and plotting. The HP2649 terminal was configured with three 8 bit duplex register boards. A functional diagram of the terminal's real-time AXBT data acquisition system is presented in Figure 2. The output from the AXBT receiver was interfaced to one of the 8 bit duplex interface boards, since the receiver provided an 8 bit binary value at TTL levels. A clock was interfaced to a second 8 bit duplex board and time in seconds recorded when the receiver interrupted with data. A binary panel was used to start, stop and identify the XBT. The panel was interfaced to the third 8 bit duplex board. Data acquired by the terminal was placed on the terminal display and recorded permanently on a cassette cartridge. This real-time acquisition, processing and display provided the scientist with immediate information where the eddy was located and information to be used in the near future and immediately by ships conducting experiments in the area.

The second terminal configured as an HP2647 Intelligent Graphics Terminal was utilized to perform computations on the data, format the data, display and print out the data andplot the results. The processed data and plotted results were obtained in hard copy format by interfacing an HP2631G Printer/Plotter to the terminal using an IEEE 488 compatible interface. Figure 3 is a copy of an AXBT plot produced aboard the aircraft. The terminal was programmed in BASIC and data was transferred between terminals using the cassette cartridges. A result of the additional processing on the HP2647 is a listing and plot of XBT temperatures as a function of depth at one second intervals converted from degrees Fahrenheit to degrees centigrade. Following the experiment the cassette tapes are stored in disk files on the HP1000 computer system. This is easily done since the tape recorded by the HP264X terminals are compatible with the HP1000 computer systems. The HP2647 terminal also serves as a backup to the data acquisition terminal. In just a few minutes with the replacement of several boards the HP2647 becomes an HP2649.

Figure 4 is a photograph of the intelligent terminal AXBT system aboard an NRL aircraft. The two terminals are placed in the same location for lack of space aboard the aircraft and operator convenience. One operator can perform the functions required on both terminals at the same time. The clock and switch panel are mounted in the same short rack. Not shown in the photograph is the HP2631G Printer/Plotter which was strapped to a nearby table.

The installation and checkout takes less than two hours. Dismantling of the equipment is normally performed in less than one-half hour and often while waiting to clear customs. Transportation of the equipment is provided by the scientist in the scientists own vehicle. The terminal systems have proven to be a viable, reliable and a very cost effective approach to real-time data acquisition and processing when microcomputer speeds are acceptable.

FUTURE DEVELOPMENTS

The first intelligent terminal application developed was for the purpose of recording magnetics data and aircraft parameters. Tests have shown that the terminal is well suited for this task when mini-computer systems are inappropriate. Other real-time data acquisition and processing tasks well suited for the terminal aboard oceanographic platforms are listed below.

- Navigation information
- Ships parameters
- Atmospheric sensors
- Oceanographic sensors

SUMMARY

The intelligent graphics terminal is a low cost alternative to oceanographic mini-computer systems for real-time data acquisition and processing when mini-computers are inappropriate and microcomputer speeds are acceptable. The compactness and portability of the units are unsurpassed, they are easy to program, use, operate and maintain and the cost is much lower than the mini-computer system. The data that is recorded on HP264X cassette drives are fully compatible with the Hewlett Packard HP1000 mini-computer systems making data transfers exceptionally easy and fast between terminal and computer. When the terminal is not functioning as a microcomputer system it makes an excellent alphanumeric and graphics terminal to the computer system.

Finally, the intelligent graphics terminal can be thought of as a well packaged, reliable and state of the art microcomputer system.

DISCUSSION

J.S. Pyett On the AXBT traces you showed there were small fluctuations visible — did these correspond to real temperature features in the ocean? What is the temperature resolution of the system?

D. Steiger These fluctuations are due to the resolution of the AXBT receiver. The receiver outputs a binary 8-bit parallel signal with a resolution of ½ degree Farenheit.

D. Nairn With reference to use of non-rugged gear in harsh environments; often the worst treatment is in transport and installation, if this is done with special care it can make a big difference to overall performance and reliability.

D. Steiger The utilization of commercially available terminals have been found to be acceptable in all environments. The most severe environment encountered has been aboard aircraft and the terminals have been found to be very reliable.

S. Meatcher Could the author comment on the reliability of these graphics terminals in the fairly harsh environment of aircraft trials?

D. Steiger From several years of using these terminals in the field they have been found to be very reliable. Also, they are very easy to repair by replacing electronic boards in the console.

M.J. McCann Is it possible to interconnect the two types of terminals to obtain the data acquisition capability, together with the processing/display capability?

D. Steiger It is theoretically possible to connect the two terminals. However, it may not be practical to do this because of the overhead and the fact that the data acquisition terminal (HP 2649) is much faster at acquisition and processing than the HP 2647 which employs a BASIC interpreter. If the data acquisition speeds were slow the HP 2647 could perform all functions.

R. Seynaeve I think there are a number of dedicated real-time acoustic systems where all you would really need is an array processor with I/O (like the MAP) and a terminal like you describe that would take care of the loading of the MAP, the operator's control, and the storage of results on the cassettes.

D. Steiger The intelligent terminal would work well with this application and similar applications.

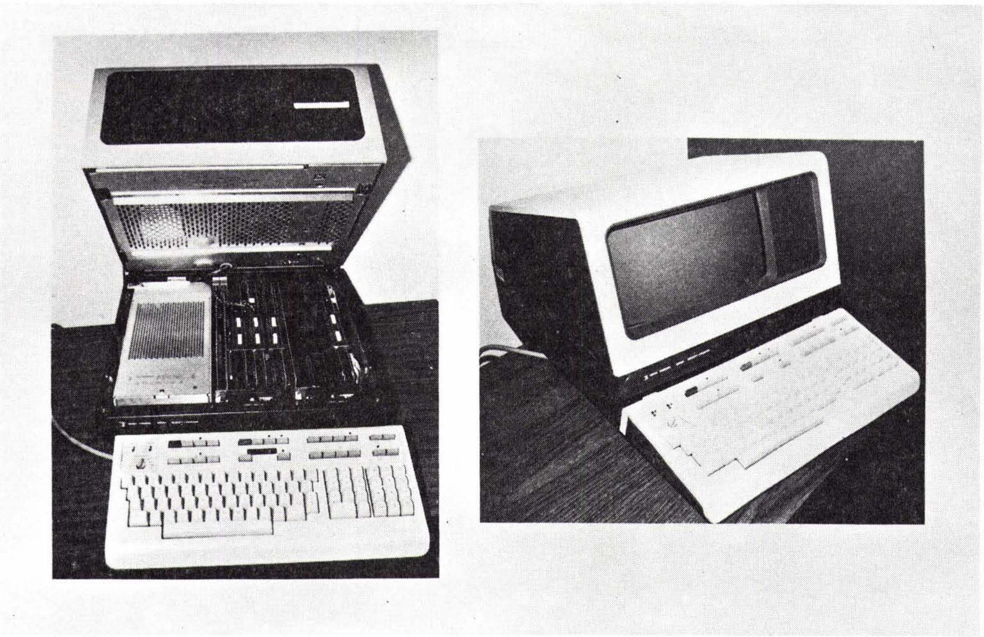


FIG. 1 PHOTOGRAPHS OF HP264X TERMINAL

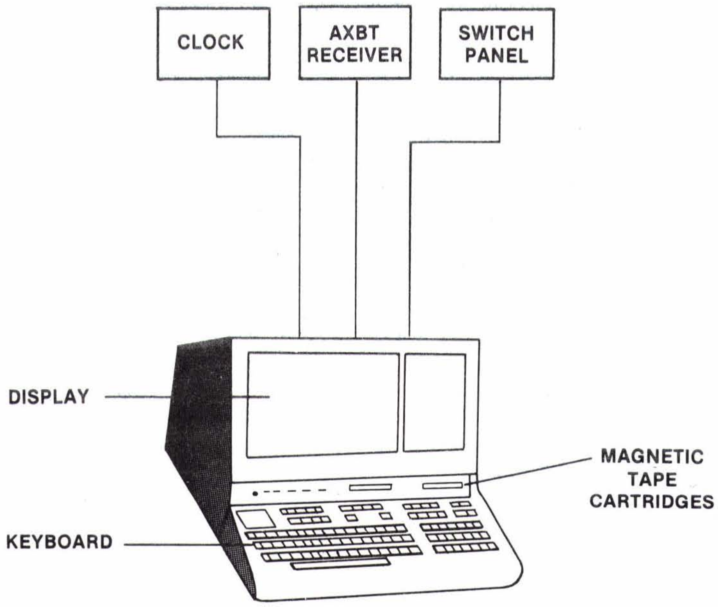


FIG. 2 FUNCTIONAL BLOCK DIAGRAM OF TERMINAL AXBT SYSTEM

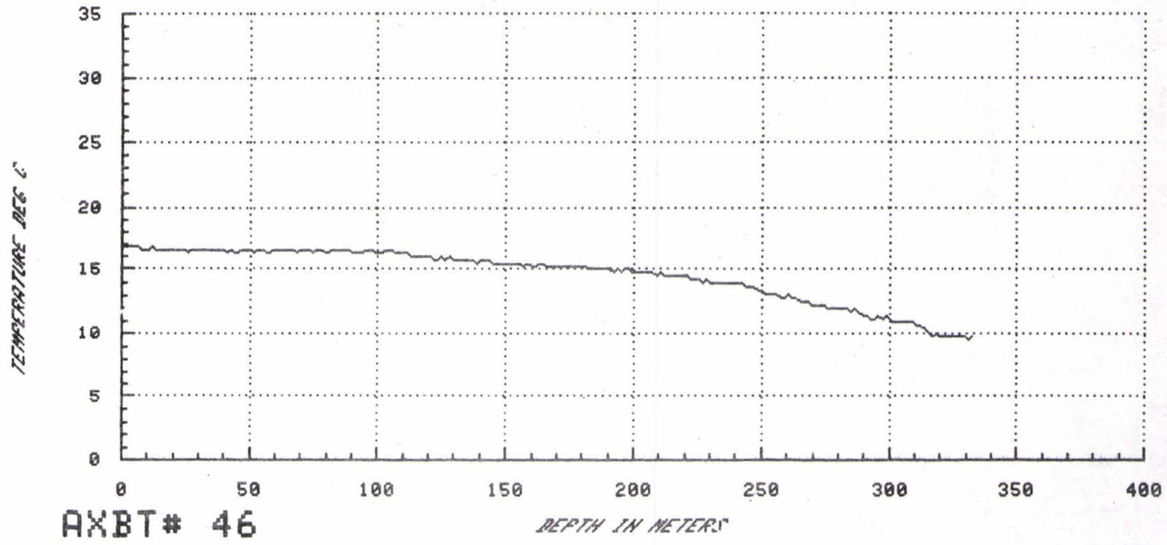


FIG. 3 AXBT PLOT PRODUCED ABOARD NRL AIRCRAFT

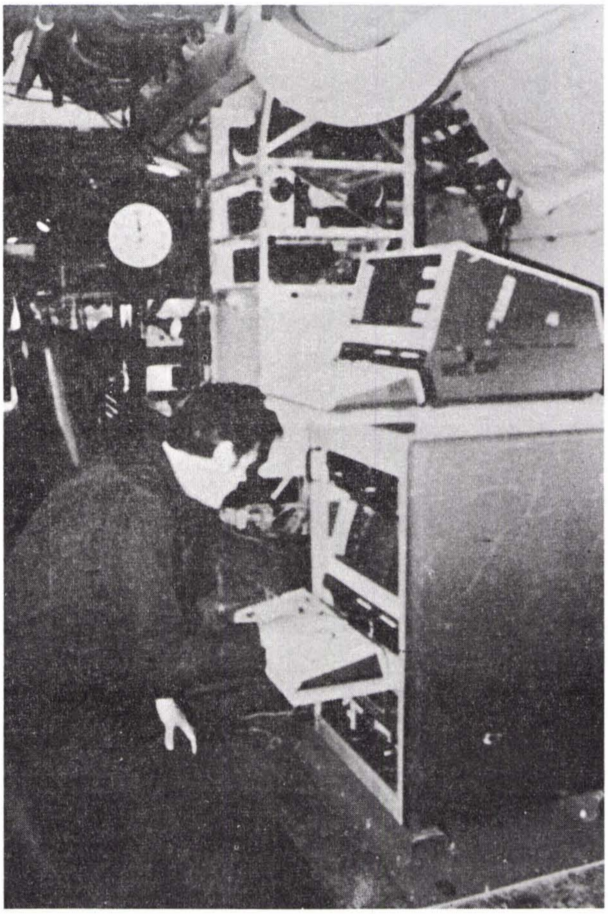


FIG. 4 PHOTOGRAPH OF INTELLIGENT TERMINAL AXBT SYSTEM ABOARD NRL AIRCRAFT

THE APPLICATION OF HIGH-SPEED PROCESSORS TO PROPAGATION EXPERIMENTS USING EXPLOSIVES

by

J. S. Pyett
Admiralty Underwater Weapons Establishment
Portland, Dorset, England

ABSTRACT The possibility of analysing explosive signals directly, without an intermediate recording system, offers large improvements in the quality and rapid availability of results from acoustic propagation experiments. The requirements for a ship-borne system to process a number of broad-band channels simultaneously and present information in both frequency and time domains are examined, together with some practical aspects of implementing such a system using currently available general-purpose array processors.

INTRODUCTION

Explosive charges have been used for many years in measurements of acoustic propagation in the ocean. In a typical experiment charges are dropped at regular intervals from a moving ship or aircraft, and the acoustic signals are received on hydrophones deployed from a stationary ship. Explosive charges are simple to use, and can be arranged to detonate at virtually any depth; the acoustic output is very stable and covers a wide frequency range. Provided the signals at the receiving points are sufficiently above the local background noise they can be analysed to reveal the distribution of energy from each shot in both frequency and time. Finally, absolute transmission loss data can be obtained by comparing the received energy levels with the known source level of the explosion.

MEASUREMENT OF EXPLOSIVE SIGNALS

In early systems for measuring received energy the hydrophone signals were taken through band-pass filters, usually one octave wide, and the output of each filter was connected to some square-law device followed by an integrating circuit. Each discrete arrival of energy from a shot should appear as a step in the output of the integrator. In practice, due to the limited dynamic range of the squaring and integrating circuits, these systems were quite difficult to operate - particularly in varying propagation conditions when the received levels can change appreciably from one shot to the next. Furthermore the measurement of many hydrophone channels in many frequency bands requires an inconveniently large amount of equipment, with increased risk of losing data.

TAPE-RECORDING EXPLOSIVE SIGNALS

It has now become general practice to tape-record the hydrophone signals and make detailed measurements from the tape recording after the experiments at sea have been completed. The equipment on board the receiving ship is designed primarily to acquire good recordings of the signals from all the hydrophone channels. A typical FM instrumentation tape recorder running at 15 inches per second has a nominally flat response from zero to 10 kHz, with a quoted signal-to-noise ratio of about 50 dB. The achievable signal-to-noise performance is however restricted by the following factors :

(a) The recording has to be made somewhat below maximum permitted level, to avoid overloading and wasting the shot.

(b) The explosion provides a very "peaky" waveform, so that if the gain is correctly set for the highest peak most of the recording is well below the maximum level.

(c) The spectral distributions of the signal and the tape system noise are different, so that when the recording is analysed into octave or third-octave bands the signal-to-noise ratio is worse in some bands than others.

Spectral shaping before recording can be applied to compensate for (c) but because the spectrum of the signal varies with the distance from the source any such compensation must be a compromise, and at very long ranges the high-frequency part of the signal is invariably lost. Simple high-frequency pre-emphasis must be used with caution due to the tendency of FM tape systems to produce large noise levels at the bottom as well as the top of their frequency range; in these circumstances the use of high-frequency emphasis to improve the long-range signals can result in the loss of low-frequency data at short ranges.

FM recording systems have been used successfully by AUWE for a bandwidth ratio of 100 (100 Hz to 10 kHz) but this is near the limit for a single channel. For larger bandwidth ratios it would be necessary to divide the range by filtering and to record the sub-bands on separate tape channels.

The explosive signals recorded at sea are analysed in the laboratory into one-third octave bands by replaying through filters and squarer/integrator circuits, or through a digital analyser based on the fast Fourier transform. In the latter case the third-octave levels are obtained by summing the power spectrum between the frequency limits of each band. In this replay process the operator is required to identify the recorded explosion signals and to position the analysis time-window correctly for each hydrophone channel; every shot is replayed as many times as necessary to obtain measurements on all channels.

DIRECT DIGITAL ANALYSIS

Sufficient has been said above to indicate the problems of tape-recording explosive signals for wide-band measurements. General-purpose digital hardware now available offers the possibility of performing continuous on-line frequency analysis of broad-band hydrophone signals. The remainder of this paper discusses how such a processing system can be organized for use on board ship in a propagation experiment using explosive charges. The advantages of such a system are :

(a) The distortions due to tape-recording are avoided.

(b) The signals can be captured and processed in a much greater dynamic range - over 90 dB for 16-bit sampling.

(c) The processed data in the form of one-third octave energy levels can be stored very compactly on digital tape.

The processing system will be made to perform Fourier transforms on successive blocks of each hydrophone signal, convert the spectral data into third-octave band levels for each block, and store this information either continuously throughout the experiment or for a sufficient length of time before and after each shot. The "intelligent" selection of the blocks containing the shot energy will be made by the experimenter; it should be possible to do this on board ship and obtain propagation loss information during the experiment. All the third-octave data will be retained on digital tape for further study if required, and this data will not be affected by any selection made on board ship.

During an explosive run the main task for the operators will be to monitor the hydrophone signals and select gain settings for each shot to give suitable peak levels at the inputs to the processor; this is much the same as setting gains for tape-recording, but with greater latitude in dynamic range, and fast peak-level indicators developed for tape-recording can be used. Signal pre-amplifiers with digitally-controlled gain are available nowadays, but in an explosive run automatic gain setting by program would probably raise more problems than it solves; manual gain control will therefore be retained, with automatic input of the gain settings to the processor.

SYSTEM SPECIFICATION

It is proposed to construct a system using a Digital Equipment Corporation PDP 11-34 minicomputer as host to a Floating Point Systems AP 120 B array processor (Figure 1).

The initial specification provides for 8 signal channels to be analysed into 20 third-octave bands from 100 Hz to 10 kHz, using an A/D sampling rate of about 25 kHz for each channel followed by Fourier transformation in the AP 120 B in 8192-point blocks. The total input rate to the processor is 200,000 16-bit samples per second; it has to perform 25 8192-point transforms per second, plus squaring and summing between band limits. This getting near the limit for one AP 120 B.

The Fourier transform frame size is dictated by the need to get enough spectral lines in the lowest third-octave band; an 8192-point transform will provide a spectral resolution of about 3 Hz, which gives seven lines in the 100 Hz band. Using this block size the processor will produce 20 band levels for each channel three times per second - a total output rate of about 500 numbers per second.

This third-octave data is passed to the PDP 11 and stored on standard digital tape. It will be possible for the operator to control which portions of the data stream are stored, but with the relatively low data rate it is probably simpler and safer to store it all. A single magnetic tape will accommodate continuous data from 8 channels for about 10 hours.

A selection of third-octave data will also be fed to paper and/or CRT outputs to enable the operator to monitor the data from the processor and observe the arrival of each explosion signal. The PDP 11 has two cartridge disk stores to support these activities, as well as the operating system software.

A system of time-coding the stored data blocks is obviously required, and perhaps some means for the operator to key in auxiliary information (for example if a hydrophone channel is faulty). The methods of meeting these needs, and some questions of overall system control, are still under consideration.

The input and output specifications quoted above match those of an existing hybrid system using FM recording and off-line Fourier processing, so that side-by-side testing of the new system will be possible. After that it is intended to improve the specification, increasing the time resolution for the high-frequency bands, and also extending the frequency range downwards. The increased frequency range can be obtained purely by software; decimating the data in the processor and Fourier transforming in longer time-windows will permit third-octave analysis down to 10 Hz - or lower still if required.

DISCUSSION

J.W.R. Griffiths Has the speaker considered using digital filtering to get the third-octave outputs directly, rather than via an FFT?

J.S. Pyett Yes, this has been discussed. I think it would be useful if we only needed one or two filters, but with a bank of 20 to 30 filters it is probably easier by FFT processing, using available software.

R. Seynaeve It appears that you will need 2 x 64 Kwords of buffering space for the input data. Can the AP120B store data in half words to save memory?

J.S. Pyett All the AP120B arithmetic is 38-bit floating-point. We will have at least 128 Kwords of data memory and that memory is 38 bits wide.

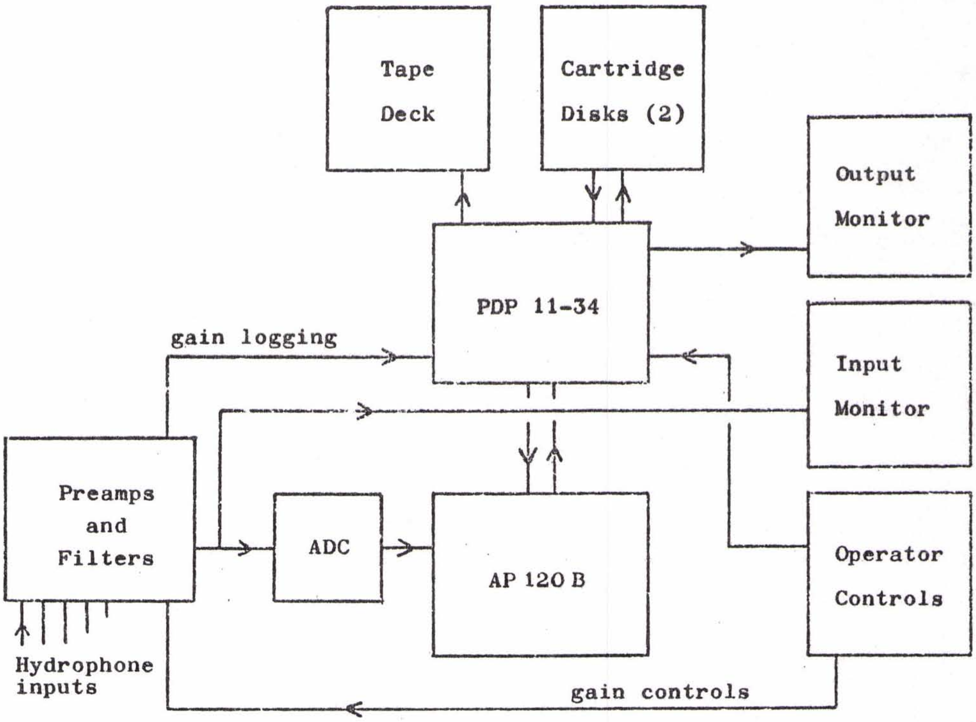


FIG. 1 BLOCK DIAGRAM OF SHIPBORNE PROCESSING SYSTEM

DIGITAL SIGNAL PROCESSING OF HIGH RESOLUTION WITHIN-PULSE SECTOR SCANNING SONARS

by

Dr J C Cook
Dr A Rushworth
Admiralty Marine Technology Establishment
Teddington, Middlesex, England

ABSTRACT The characteristics of three scanning sonars, presently in use at the Admirlaty Marine Technology Establishment, and operating at frequencies of 300kHz, 150kHz and 75kHz are described. The performance of a system for digitally recording the information from these sonars is detailed. A six bit (64 levels) analogue to digital conversion is employed at such a rate as to maintain the basic accuracy of the sonars. This data is recorded onto digital cassettes following ECMA 34 standards. The recorded information can be replayed into a PDP 11/40 computer via a digital highway for subsequent laboratory analysis. The processed data is then presented in various graphical three dimensional formats, examples of which are shown. The use of this analysis and representation is illustrated with reference to the use of the scanning sonar in three different modes of operation, namely echo sounding, forward search and depth scanning.

A system is described that continuously records whole successive frames of sonar information onto an instrument recorder using a multitrack digital recording technique. This system can also capture a frame of data into a random access memory of 512 pixels by 512 pixels of eight bit resolution (256 levels) and display the sonar picture in B scan format on a conventional colour TV monitor. The range and bearing are represented by the vertical and horizontal positions of a point whilst the echo intensity can be represented as one of 127 colour levels using a colour look-up table technique.

INTRODUCTION

High resolution, within pulse, sector scanning sonars were first developed at the Admiralty Marine Technology Establishment (AMTE), Teddington, over 30 years ago. In 1964 the use of this type of equipment was demonstrated to outside civil authorities and found application, particularly in the field of fisheries research (1). In recent years, the impetus given firstly for the requirement for more precise hydrographic surveys (2) and then by the exploration for North Sea oil, has resulted in the present development of a number of commercial sonars based on the same principle of modulation scanning. The full potential of this type of sonar has, however, remained relatively unexploited and the sonar display and analysis techniques have remained substantially the same over the years. The application of digital signal processing and recording systems can now provide the means, for example, of applying pattern recognition techniques, contouring sea bed profiles, and estimating marine biomass and target strength data from basic sonar signals.

The application of computerised signal processing to three scanning sonar systems covering a frequency range of two octaves is described. The prototype digitisation system has now been in operation for a number of years and is at present being updated.

1 THE SONAR SYSTEM

An electronic scanning sonar generally ensonifies a specified volume of the sea (or area of the sea bed) using a separate transmitting transducer. Scanning is normally restricted to the receiver only and by subdividing the array into the appropriate number of elementary receivers suitable phase shifts may be applied to each individual output to steer the direction of maximum sensitivity of the array across the sector. The scanning frequency is matched to the pulse length so that the whole sector is scanned once in the time taken for the acoustic pulse to travel one pulse length in water. Scanning is continuous so that as echoes are received from each range increment they may be uniquely identified with resolution 'cells' defined in both range and azimuth. In the non scanned direction the resolution can only be specified in relationship to the beamwidth in that direction.

A complete 'picture' is built up in this way of the echo structure of the ensonified sector up to a range ultimately defined by the sonar parameters of the equipment. Conventionally the sonar information is presented in the form of a B-scan display in a Bearing (X deflection) - Range (Y deflection) matrix, the sonar signal itself providing the Z modulation, using a long persistence phosphor. Three modulation scanning sonars are now in use, operating at carrier frequencies of approximately 75kHz, 150kHz and 300kHz and these can be operated either singly or in synchronisation.

The 300kHz system employs a 150 λ long receiving array, with some measure of array shading to reduce unwanted side lobes and this results in an overall beamwidth in the scanned direction of 0.4°. The beamwidth in the non scanned plane is approximately 7°. The transmitted pulse length is 100 μs and the scanned sector 30°. This sonar can detect a -45dB target at a range of 100m and a 0dB target to ranges of approximately 300m. The performance of this type of sonar when installed on the fisheries research vessel 'CLIONE' has been reported in reference (3).

The measured resolution of the sonar has been determined to be very close to that defined by theoretical predictions (4) and provides a resolution capability approximately equivalent to that defined by the nominal beamwidth down to ranges of 60m. The basic resolution 'cell' of the sonar, at ranges in excess of 60m may therefore be calculated as:

$$0.007 \times 0.123 \times 0.075 \times R^2 \text{ cubic metres}$$

where R is the operative range in metres.
The effective volume resolution (Figure 1) gives a resolution cell of 1 cubic metre at a useful working range of 125m.

For the 150kHz equipment the basic parameters of the signal processing is very similar to those of the 300kHz sonar, the identical scanning frequency, 30° sector and transmitted pulse length being preserved to give the same range resolution capability of 0.075m. The receiving array has an aperture of 90 λ and a nominal beamwidth of 0.56°. The detection range for a 0dB target is well in excess of 400m but for normal operation the range of detection is generally limited to about 375m defined by a pulse repetition frequency of 2Hz. The beamwidth in the non scanned direction is 10° and with these design parameters the resolution cell at 200m range is approximately 5 cubic metres.

At 75kHz it is not practical to maintain the transmitter pulse length at 100 μs due to bandwidth problems, and at this frequency the pulse length is increased to 0.5 milli-seconds and the appropriate scanning rate is 2kHz. The effective range resolution is therefore 0.375m. A 45λ long receiver array with a beamwidth of about 1.1° in the scanned direction is employed with a beamwidth in the non scanned direction of 22°. For this sonar the sector has been increased to 60° and the range of detection of a OdB target in normal low sea states is well in excess of 800m. A 1Hz pulse repetition rate is normally used, limiting the accepted range gate to 750m. The volume resolution cell at 600m range is now nearly 1,000 cubic metres.

Each sonar is provided with various gain control systems which allow the performance to be optimised for particular operational roles. For general research purposes described in this paper a Time Varied Gain (TVG) system is used which compensates the gain for spherical spreading and absorption, giving the same amplitude of signal for a specific target independent of range.

2 METHODS OF OPERATION

The three sonars provide long range detection with a short range, very high resolution, target delineation capability. The versatility of the arrangement is enhanced by the use of the equipment in any one of three modes of operation.

2.1 Azimuth scanning mode

This is used for forward search applications, where the long axis of the transducer is horizontal and the array is scanned in the horizontal plane, (Figure 2). In this case the target on the sea bed is delineated in both range and azimuth.

2.2 Depth Scanning

When the transducer is rotated so that the major axis of the transducer is vertical, scanning will take place in the vertical plane. The receiver beam is then scanned from the sea surface to the sea bed and the profile of the sea bed contours can be examined. This type of operation can be used for hydrographic survey applications and the determination of the free depth of water above bottomed hazards such as wrecks (5).

2.3 Depth Sounder Mode

A second method of depth contouring can be achieved by tilting the transducer to point straight down and rotating the major axis of the transducer to the athwart ships. In this arrangement the system can be used in the echo sounder mode for across track profiling, and has recently found application in the monitoring of bottom laid oil pipe lines.

In all the applications described above the sonar will operate simultaneously in the passive reception mode. A continuous source of noise in the far field will appear as a target at all ranges on the appropriate bearing. An example of the use of this system is in the monitoring of sea bed noise associated with the movement of sediments and sand waves under the influence of tidal flow (6). The use of small transponding fish tags has enabled fish to be tracked at large ranges in excess of that possible by using the active mode alone.

3 LIMITATIONS OF THE CONVENTIONAL SYSTEMS

The video display is normally the only method available for real time analysis and is adequate for short range use where a higher frame rate can be employed to give a reasonable flicker free display. For the long range sonar with an up-date of 1Hz or less, this type of display becomes practically unusable for locating and tracking small targets. Permanent recording is generally provided by photographic methods, or alternatively by some form of analogue tape recorder. Photographic recording has a severe limitation, however, particularly for research applications, in making quantitative measurements.

The detail of structure that may be obtained of sea bed reverberation, for example, now requires not just a single measurement of backscatter strength to define a given area, but ideally requires a considerable number of simultaneous measurements to be made as the area can now be surveyed in great detail. The sonar originally developed at AMTE was provided with a calibration method by which a signal could be injected across the inputs to the sonar thus establishing a comparison signal to which other target levels could be related. Calibration by this method, however, is time consuming and somewhat inaccurate and when undertaken at sea becomes impossible to use when more than a single target is of interest.

A means by which all these drawbacks may be eliminated is now provided by the implementation of digital signal processing to the analogue output of the basic signal processor, and by subsequent computer based analysis. By this means the amplitude information in each individual resolution cell may be stored, identified and digitally recorded. The sonar data can now, therefore, be examined in much greater detail than has hitherto been possible.

4 DIGITISATION OF THE SONAR DATA

The scanning sonar achieves a very high data rate compared with the conventional sonar, and the bandwidth of the information from the 300kHz sonar is approximately 400kHz. This data rate is the highest associated with all the scanning sonars described and the system requirements will therefore be detailed in relation to this particular equipment.

With a 4Hz information up-date 2,500 scans are made, each scan corresponding to a 0.075m range increment. A sampling frequency of 800kHz is used, which is in accordance with the requirements of basic sampling theory, being approximately twice the highest frequency component appearing in the demodulated output.

When this digitisation system was designed, several years ago, it was not considered to be cost effective, or indeed necessary, to provide for storage of all the information in the complete range gate. For general surveying a much lower storage capacity could be used, either averaging over a number of scan lines, or by selecting every eight or tenth line for storage and display. When it is required to use the full resolution capabilities of the equipment it is generally for the close inspection of a relatively small target for identification purposes. The plotting of sea bed contours in the depth sounder mode (section 6.3) is an example of this, and for this purpose again a much reduced storage capacity can be tolerated. For the prototype equipment, therefore, it was decided to limit the storage capacity to the equivalent of 320 scan lines for the 300kHz sonar in view of the storage components then commercially available.

There are, however, a few occasions where an extended range capability at high resolution is desirable. Amongst these are the detailed examination of underwater hazards such as wrecks, inspection of oil rigs and the monitoring of dykes. The new storage system under development will, therefore, be provided with a full range storage capacity. A shift register system was chosen for the initial development, the new system being based on a Random Access Memory matrix of 512 x 512 storage capacity.

The amount of amplitude information that it is necessary to store determines the number of bits for the amplitude word. To cover the amplitude range necessary to reproduce a reasonably acceptable video picture probably requires no more than a four bit word due to the relatively restricted number of grey levels that can be discerned on the normal TV-type screen. This storage capacity is, however, insufficient for many research applications. For fisheries research, for example, it may be necessary to detect a single fish with a target strength of -37dB (single 0.1m length fish in the broadside aspect) or to estimate the target strength in one resolution cell of a closely packed shoal, which for the same type of fish, in the same orientation, and at maximum packing density could approach a figure of 0dB per cubic metre. The difference in target strength evidenced in a single shoal could therefore be of the order of 40dB. This order of dynamic range is generally more than sufficient for most other applications and the system was therefore based on the use of a six bit word to carry the amplitude information, this is equivalent to a linear amplitude ratio of 64 or a range of 36dB.

The shift register store is arranged in six planes, one plane for each bit of the 6 bit amplitude word. Each shift register chip has the capacity for 1024 bits and so a total of 25 shift registers are required for each bit plane in order to digitise 320 lines, taking 80 samples per line, or for the 75kHz sonar, 512 lines taking 50 samples per line. The digitiser itself consists of two separate units, the digitiser and the cassettee replay unit.

4.1 The Digitiser

This unit samples the signal at a little over 8×10^5 samples per second, to a six bit resolution and stores them in a solid state store (the main shift registers, MSR's). The data is sampled over a time window of 32ms duration (256ms for the 75kHz sonar), delayed in units of 1ms (5ms for the 75kHz equipment), from the transmission initiation signal. The sampling of each scan line is synchronised to the scanning of the sonar. The stored data is converted, for display purposes, to analogue form and can be displayed either at the normal sonar frame rate or at a higher rate of 31Hz to give a flicker free picture on a conventional short persistence oscilloscope tube.

The digitiser transfers the data from the MSRs to the cassette replay unit. Eighteen frames can be recorded on a standard cassette. The format and parity of the data is checked automatically when it is replayed from the cassette tape into the digital store. The digitiser has an averaging facility which will average 2, 4 or 8 successive sonar transmissions.

4.2 The Cassette Replay Unit

The cassette replay unit consists of a cassette tape transport deck, phase encoding and decoding circuits, byte compilation circuits, interblock gap detection circuits and is parallel interfaced for replay into a PDP 11/40 mini computer via a DR 11 C digital highway. Block formatting is used so that on replay into the computer the digital flow can be stopped at the end of each block to allow time for the computer to perform numerical operations and checks.

Within the block the data is organised into bytes of 8 bits of the form ABXXXYYY, where XXXYYY are the six bits from the ADC, A is the parity bit for XXX and B the parity bit for YYY. After 240 bytes of data there follows a block check character (one byte) which is incorporated to give longitudinal parity checks on a byte basis. Odd parity is used throughout.

As the state of digital techniques has advanced it is now more practical to devise a system that will digitally record each successive whole frame of data. Such a system is now being developed and again consists of two units, a recording system and a computer controlled replay system.

4.3 The Mk 11 Recording System

The incoming signal from the signal processing unit is again digitised to a six bit amplitude word, and this information is distributed over 13 tracks of an Ampex PR 2200 instrument recorder operating in the direct wideband 2 mode Each scan line is digitised at a precisely defined sampling rate, the ADC sampling being initiated by the recognition of the line synchronising pulse. Two six bit data values are packed into a 16 bit word along with a line flag, frame flag and a parity bit. A validity flag is incorporated into this 16 bit word which is set positive for valid data and zero for packing data. The 16 bit words are then distributed between the instrument record tracks in serial form at a density of 10K bits per inch. The digitisation and down loading of the data to the instrument recorder is achieved in real time.

4.4 The Mk 11 Replay System

On replay, the signals from the 13 tracks are passed to deserialisers which decode and deserialise the data. The bit parallel words complete with the appropriate flags are then passed via a DR 11 C direct memory access module to the computer. The replay unit buffers the data and checks the validity and parity flags. This data is reorganised into bit format acceptable to the computer. A digital to analogue converter is incorporated into both record and replay units for downstream monitoring. The data, held in the computer may subsequently be reduced and analysed using the display facilities.

4.5 The Display System

The display uses a conventional 625 line colour TV monitor operating at an interlaced frame rate of 50Hz. The data to be displayed is passed from the computer to the display system's random access frame store of 512 by 512 pixels, each of 8 bit amplitude. The top 6 bits of the 8 bit plane store holds the sonar data, the remaining 2 bits are available for annotation and graphics which may be shown simultaneously with the sonar information.

The transfer of data to the frame store is controlled by a fast running software system based on integer arithmetic. The use of software to control the frame store introduces a large amount of flexibility into the system. The reading and writing process for the frame store are independent, can operate simultaneously and can be completely asynchronous. The mini computer has direct access to any picture point in the store so that it can either read information from the frame store, write information to it, or remove data from the picture, modify it and replace it. At all times the contents of the frame store can be displayed on the TV monitor. The store is bit selectable so that any part of the 8 bit word can be read or modified.

The frame store and colour look-up tables are controlled by software to display the sonar data. Two display formats can be used. The whole sonar frame can be displayed in a series of up to six columns each holding 512 scan lines of data or alternatively any range gate of 512 lines can be shown using the full width of the TV monitorby reading each data cell into six successive picture points. The colour look up tables allow the echo intensity to be represented as a particular colour. Any frame of the sonar data may be held indefinitely in the store for detailed examination.

A 'window' facility is available whereby only a limited selected area of incoming data is overwritten. The window area can be designated as either a portion of the picture to be written into the store or as a portion of the store to be protected with the remainder of the picture being entered in memory. A hardware cross wire facility is also included, controlled by the computer. These cross wires are mixed into the video after the store and do not corrupt the store data. They can be used to specify data points or window boundaries and, therefore, form an interactive facility.

5 COMPUTER BASED ANALYSIS

The digital analysis system has now been in use for a number of years and a series of programs have been written to process the data received from various modes of operation.

5.1 Computer Hardware

The computing facility is based on a DEC PDP 11/40 mini computer. The system has a 25K words of memory and a hard wired arithmetic unit. Three discs, each of 1.2M words capacity provide additional storage and operate under the DEC RT 11 system. Additional devices include a laboratory peripheral system, a decwriter and a Versatec 1200 A electrostatic printer plotter. This mini computer is host to a Tektronix 4081 interactive graphics system.

5.2 Computer Software

The transfer of data from the digital tape to the computer is handled by a mixture of hardware and software. The tape replay unit hardware converts the 8 bit word read from tape into 8 bit parallel format for the DR 11 C to transfer. The hardware aligns on word and block boundaries, but not on record boundaries, nor does it do any parity or format checking. The tape driver software institutes and controls tape movements, checks validity of ambles, checks byte and long parities, unscrambles data and converts it into computer internal format. The data is written away for use by other programs and up to 70 sonar transmissions can be entered on a single disc. The raw data from the disc may be operated upon in several ways. Each transmission can be normalised in target strength values by the use of information from standard targets. The data may be operated on by 'picture enhancement' routines, an example of which is for the supression of unwanted sidelobe effects. This takes into account that the sonar directivity response is not ideal but has a sidelobe response of a general sin x/x pattern. A correction can be applied to the raw data to remove first order effects of these sidelobes.

The sonar data may be represented in several forms for further analysis and visual interpretation and these will be considered individually.

5.3 Tabular Listings

This provides possibly the simplest form of output and consists of a tabular listing of the target strength values, in each resolution cell of the range-bearing matrix. The table has 80 columns corresponding to the 80 digitised azimuth cells and 320 rows corresponding to the 320 digitised scan lines, for both the 300kHz and the 150kHz sonars. For the 75kHz sonar there are 50 columns and 512 lines. Only one character can be used to represent the target strength value, due to device constraints, and hence to cover the required dynamic range of say 30dB, using only the digits 0 through 9, each numeric increment represents a 3dB change in target strength. By using the full alpha numerics available a target strength resolution of 1dB can be accommodated. A cut off level is available whereby all target strengths below a selected level are represented by a full stop. This aids readability by the suppression of all but the target of interest in high signal to noise conditions. This form of output (Figure 3) has a major failing in that the aspect ratio is incorrect. The magnitudes of target highlights, however, are very easily obtained from this form of presentation which can be produced very quickly on a line printer.

5.4 Hidden Line Plots

This is a pseudo three dimensional graphical output. A Cartesian axis system is used where the X axis is made proportional to azimuth and the Y axis is made proportional to range. The Z axis represents the target strength for that region of space given by the XY co-ordinate. The effective Z axis is represented by a line parallel to the X axis, whose distance from the X axis represents the magnitude of the Z quantity, in this case the target strength. Such a line is produced for each of the digitised 320 scan lines (512 scan lines for the 75kHz sonar). Each successive scan line is offset along the Y axis to give range information. This form of output is made more visually acceptable by using a hidden line routine.

This representation of the target strength distribution (Figure 4) has an approximately true aspect ratio, which aids the positioning of targets. Unfortunately, true X - Y positioning is not obtained, but is given in terms of range R and sin Θ, where Θ is the target bearing angle. The target strength values are not easily obtained from these plots as it entails measuring the height of the corresponding peak deflection. The hidden line routine also introduces a possible source of error in that a small target echo may be 'lost' behind a larger echo.

5.5 Contour Plots

The target strength data results in an array or matrix of target strength values. This grid can be converted into contour lines and the program shades these to aid readability. The contour plots are converted from R and sin Θ co-ordinates to X - Y co-ordinates before plotting.

The contour plot routine only considers data from a rectangular zone whose position and orientation in the sector can be specified. This was done so that a scaling ratio of 50:1 could be used. With the given scaling ratio in true co-ordinates the location of points on an exact target strength matrix is possible. The direction of ensonification is indicated by a small arrow. The magnitude of the echo is given by the contour level. The number of different contours has been restricted to about 4 or 5 and therefore each contour level usually represents a 4 or 5dB range of target strength values. This type of contour presentation is illustrated in Figure 5.

5.6 A Scan Format

Conventional sonars provide resolution only in time, and for any single range resolution cell only a single echo is recorded. In this case the target strength data is normally presented as a function of range only, in A scan format. To aid the comparison of scanning sonar data with this form of low resolution information it is necessary to degrade the data from the scanning sonar by combining all azimuth data at a given range. The recorded azimuth information can be recombined by the computer over any given number of azimuth cells to simulate a sonar with a transducer beamwidth of any resolution less than that of the scanning sonar up to that defined by the sector width of 30°. This target strength data can be produced in histogram format where each column represents a new scan line or alternatively a non histogram presentation is available where each successive range cell is printed out as a line parallel to the Y axis as shown in Figure 6. The length of the line is proportional to target strength and its position on the X axis represents range.

5.7 Interactive Graphics Terminal

A recent addition to the computer hardware has been a Tektronix 4081 interactive graphics system. This facility provides both refresh display facilities for dynamic pictures, and storage display facilities to allow large amounts of graphics information to be displayed simultaneously. The terminal has internal programming to permit operation in a 'stand alone' mode but it can also be used in a host environment. When the equipment is mated to the basic computer system this permits a more rapid analysis of the digital data to be adopted by the use of superposition and split screen techniques.

6 APPLICATIONS

The use of the digitisation and scanning sonar system can be illustrated by its utilisation for hydrographic survey and fisheries research. The requirement for accurate and detailed hydrographic surveys has increased due to the use of deep draught ships operating in the relatively shallow waters of northern Europe. To produce a typical high density survey using conventional echo sounding sonar involves the survey ship crossing the sea bed in a large number of closely spaced tracks. Any hazards such as isolated rock pinnacles or wrecks can completely escape detection by this tedious method of survey. The use of scanning sonars, coupled with the digital signal processing described, can provide a long range search capability and detailed depth profiling to give an integrated system for high precision surveys.

For fish detection, and in particular for fisheries research, there exists an immediate requirement for fish stock surveys to be undertaken with a reasonable degree of precision. The UK for example has a need for both herring and mackerel stock assessment. The high resolution scanning sonar provides the capability for this type of work.

6.1 Azimuth Scanning

A hidden line plot of a small fish shoal and two standard targets detected with the 300kHz sonar is shown in Figure 4. As this shoal was moving steadily across the field of view, the fish were probably in broadside aspect. If it is assumed that the minimum target strength recorded is that of a single fish and that the target strength in a single resolution cell increases by 3dB for each doubling of fish numbers, it is possible to estimate the shoal biomass.

The lowest fish target strength measured is -27dB indicating the size of the fish to be approximately 0.3m in length. The highest target strength of -11dB suggests a maximum packing density of 50 fish m^{-3} (4), which is about 'normal' for this size of fish. The statistics of target strength occurrence provided by the tabular listing program gives a total fish count of 13,700 fish. For this size of fish this implies that the shoal has a biomass of 6400 Kgs.

A very small shoal of a distinct ring shape is shown in the three representations of Figures 5, 6 and 7. The vertical line (Figure 7) is the noise radiated from a small boat being used to identify the fish. The broadside target strength of these fish is -37dB indicating their size to be only 0.1m.

6.2 Depth Scanning

This mode of operation can be used to examine fish distributions in depth and is also of use in survey work. The example given in Figure 8 obtained with the 300kHz sonar shows the sea surface on the left, the sea bed on the right and a midwater fish shoal. From the sonar calibrations the maximum target strength in this shoal is -20dB and the minimum -41dB. This shoal was moving towards the sonar and the fish were, therefore, in head on aspect. The fish were identified as small whiting some 0.15m to 0.20m in length which agrees with the measured target strength values.

As both the sea surface and sea bed are clearly delineated the water depth can be computed. The sea surface and bed are separated in bearing by 15^{o} at the near range of 84m giving a water depth of approximately 22m. The 75kHz sonar when used in this mode of operation allow sea bed profiling to be achieved at much longer ranges of up to 750m.

6.3 Echo Sounder Mode

The 75kHz sonar's combination of long range detection, high resolution, and a large scanned sector makes it suitable for use as a high precision echo sounder. At 800m range the area of resolution is just over 4000 sq metres. The target strength of a mud sea bed, which is the worst conditions under which the equipment is likely to be used, is no less than -40dB m^{-2} and hence as the detection capability of this sonar is well in excess of a 0dB target at this range all sea beds will be delineated out to the full range of 750m accepted.

Three 'hidden line' plots of bottom profiling in this mode are shown (Figures 9 a, b and c), ranging from the near flat to very steep, in an average water depth of about 100m. In two cases targets, probably small shoals of fish, can be detected near to the sea bed. Figures 10 a, b and c show the respective sea bed profiles computed from this data and presented in true coordinates. At this depth the area of the resolution cell is about 60 sq metres and over a flat sea bed the bottom profile can be sampled in depth over a distance of ± 50m either side of the ships path at an average sampling distance of about 2m.

Figure 3 shows a tabular listing of the target strength data for the maximum slope condition. For a flat sea bed the range of the bottom would increase symmetrically about centre bearing. The back-scattering strength of the sea bed at various angles up to ± 30^{o} can be estimated from this type of data.

CONCLUSIONS

The use of digital signal processing in association with a sector scanning sonar can provide a direct method of measuring the target strength of fish and biomass, although the problems associated with fish orientation and the attenuation of sound through the shoal still remain to be solved. The integrated system is equally well suited to producing detailed and accurate hydrographic surveys.

REFERENCES

1. VOGLIS, G.M. and COOK, J.C. Underwater applications of an Advanced Acoustic Scanning Equipment. Ultrasonics Vol 4 Jan 1966.

2. COOK, J.C. and MACKAY, J.M. A precise sonar for Hydrographic Surveying. National Electronics Review V7 No 1 Jan-Feb 1971.

3. MITSON, R and COOK, J.C. Shipboard Installation and Trials of an Electronics Sector Scanning Sonar. Conference paper at Electronic Engineering in Ocean Technology 21-24 Sept 1970. Inst of Elec & Radio Engineers Conference Proceeding No 19.

4. COOK, J.C. and RUSHWORTH, A. Measuring Fish target strength and shoaling density using a High Resolution Scanning Sonar. Institute of Physics Underwater Acoustics Group. Proceedings of Conference 'Acoustics in Fisheries' 26,27 Sept 1978.

5. COOK, J.C. Some Practical Applications and Limitations of High Definition Depth Scanning Sonars. Proceedings of NORSPEC 70 Conference The North Sea Spectrum 117 - 130.

6. VOGLIS, G.M. and COOK, J.C. A new source of acoustic noise observed in the North Sea. Ultrasonics Vol 8 1970.

DISCUSSION

T. Totten Do you have a feel for the relative performance of pseudo 3-D displays and those using intensity modulation?

J.C. Cook For our purposes the number of display levels available using intensity modulation is not sufficient. One disadvantage this type of display has is, in fact, that if hidden-line techniques are used to make the display more acceptable to the eye there is the possibility of small targets getting lost behind slower larger targets on the same bearing. On the other hand, intensity measurements can be made directly, which is not normally possible using intensity modulations. We have no experience from the point of view of initial detection in noise or reverberation-limited conditions.

J.W.R. Griffiths In the use of the system as an echo-sounder there is obviously a problem if side-lobes on certain types of sea beds. Is it possible to use deconvolution to get around this problem?

J.C. Cook A side-lobe suppression program has been developed and has proved very successful in circumstances where high echo levels from such targets as sea beds at normal incidence cause side-lobe interference across other parts of the sector. (This is mentioned briefly in the published paper Sect. 5.2.)

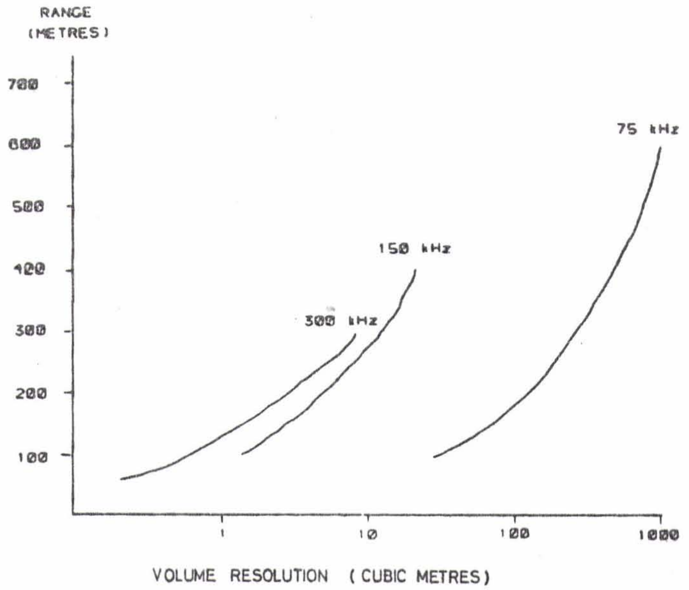


FIG. 1 THE SCANNING SONAR RESOLUTION CELL AS A FUNCTION OF RANGE

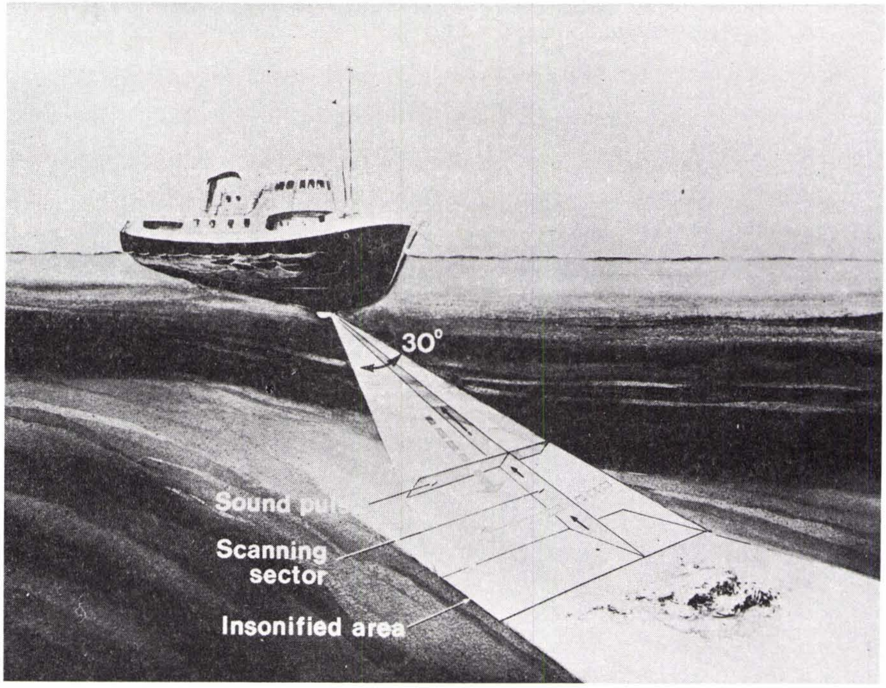
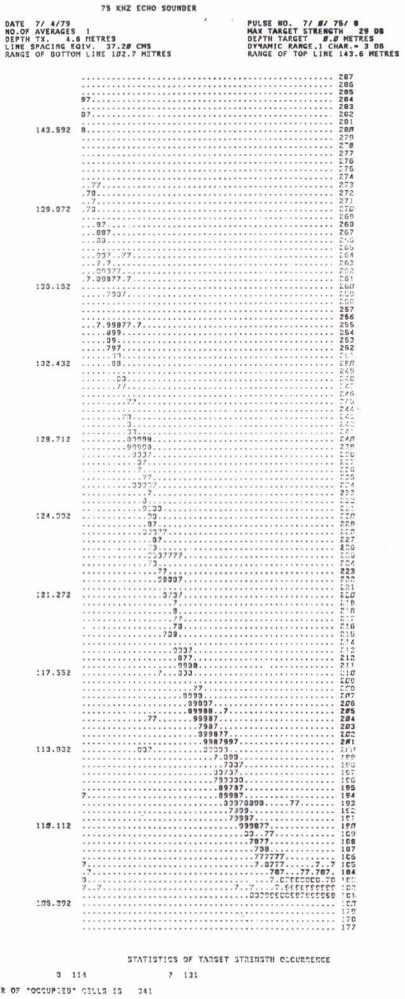


FIG. 2 THE SCANNING SONAR IN AZIMUTH SCANNING MODE



```
                    75 KHZ ECHO SOUNDER

DATE  7/ 4/79                              PULSE NO.  7/ Ø/ 75/ 8
NO.OF AVERAGES  1                          MAX TARGET STRENGTH   29 DB
DEPTH TX.   4.6 METRES                     DEPTH TARGET   Ø.Ø METRES
LINE SPACING EQIV.  37.2Ø CMS              DYNAMIC RANGE.1 CHAR.= 3 DB
RANGE OF BOTTOM LINE 1Ø2.7 METRES          RANGE OF TOP LINE 143.6 METRES
```

```
                              STATISTICS OF TARGET STRENGTH OCCURRENCE
9   95                 3  114                 7  131
TOTAL NUMBER OF "OCCUPIED" CELLS IS   341
```

FIG. 3 AN EXAMPLE OF A TABULAR LISTING OUTPUT OF 75 kHz SONAR OPERATING IN DEPTH SOUNDER MODE

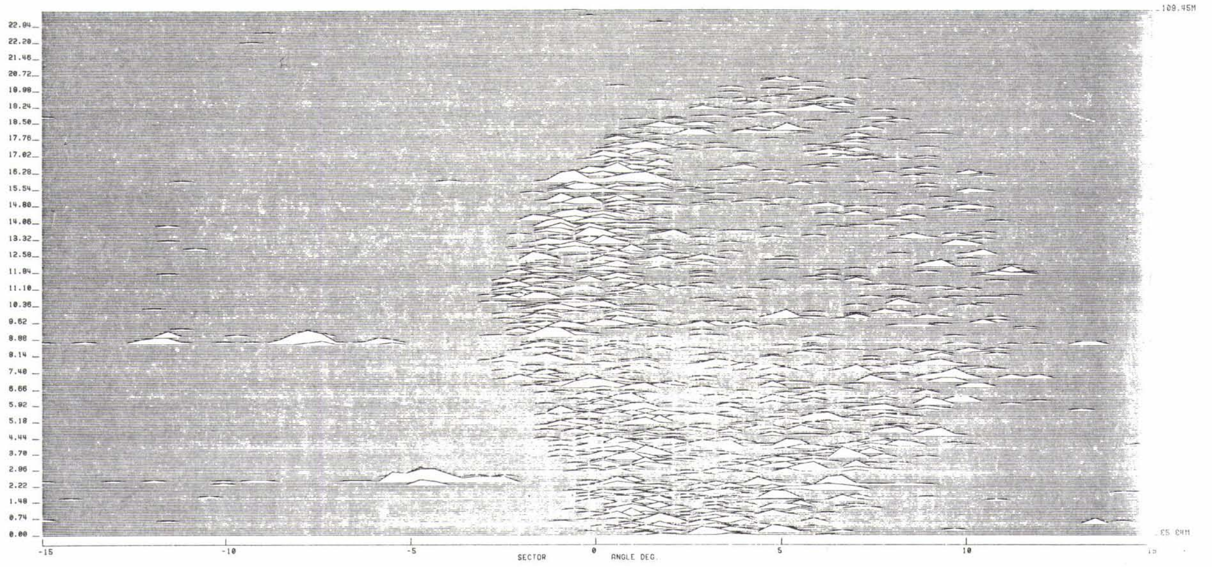


FIG. 4 AN EXAMPLE OF A HIDDEN LINE PLOT SHOWING A FISH SHOAL DETECTED WITH THE 300 kHz SONAR OPERATING IN AZIMUTH SCAN.

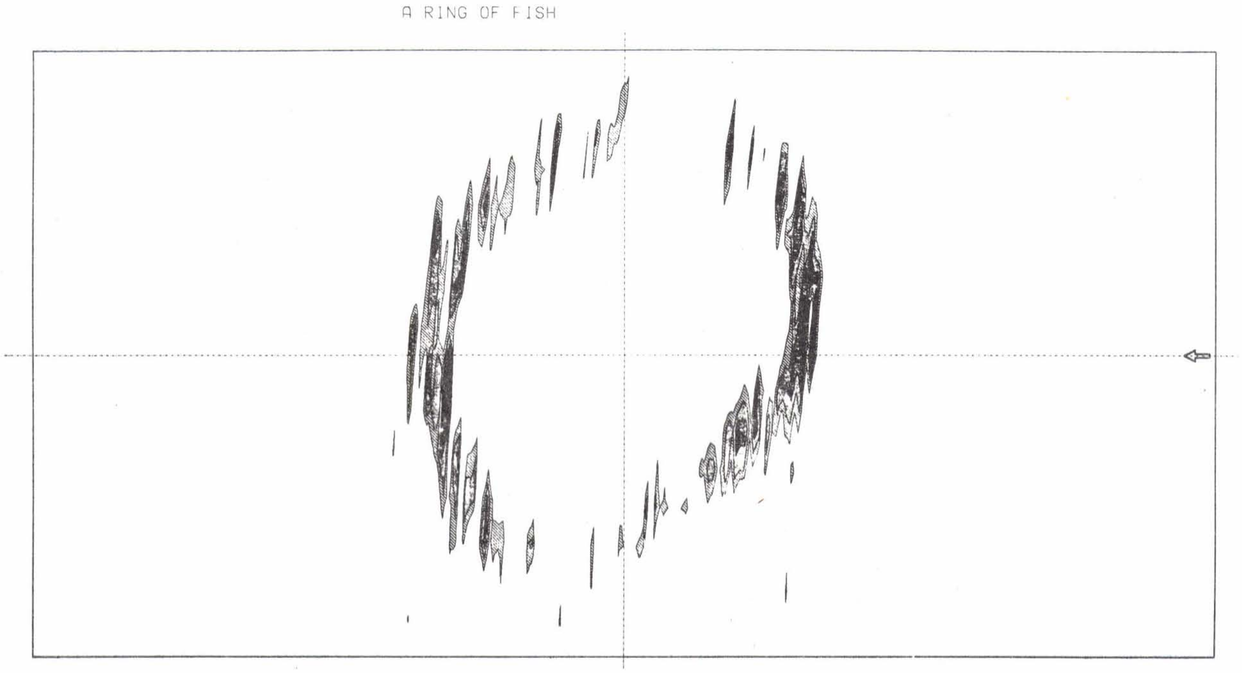


FIG. 5 AN EXAMPLE OF A CONTOUR PLOT SHOWING A VERY DISTINCT RING OF FISH.

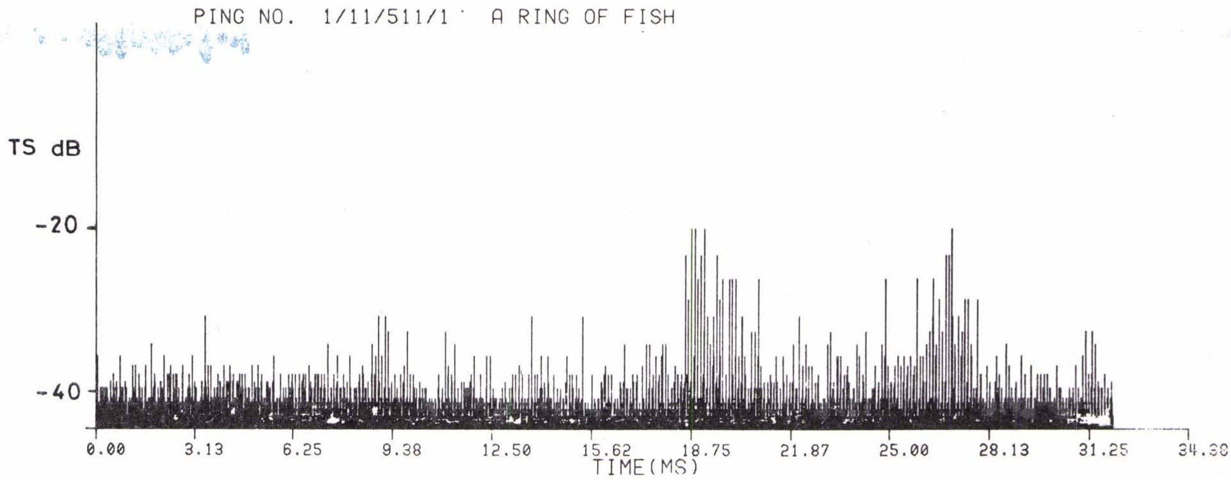


FIG. 6 AN EXAMPLE OF THE A-SCAN FORMAT PRODUCED FROM THE RING OF FISH DATA SHOWN IN FIGURES 5 AND 7.

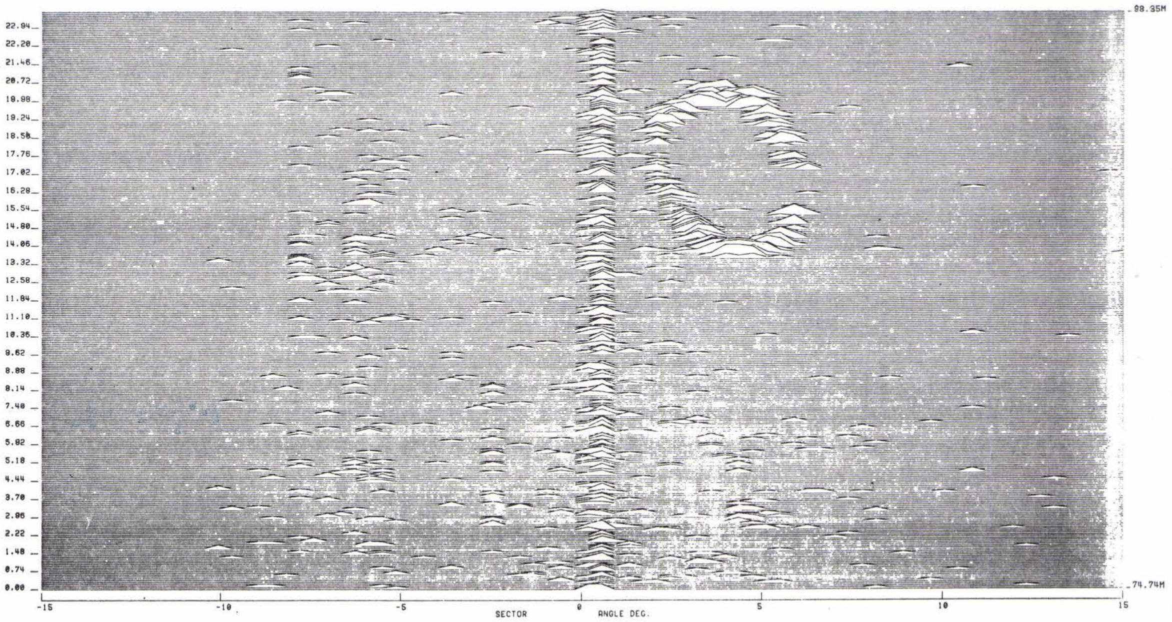


FIG. 7 A HIDDEN LINE PLOT OF THE RING OF FISH.

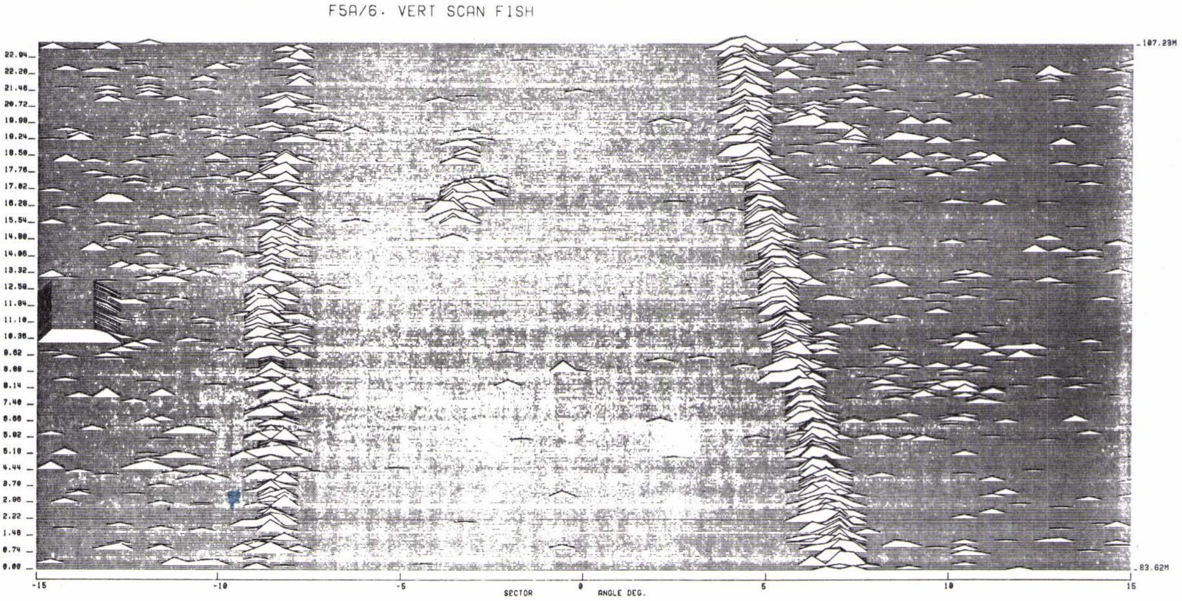


FIG. 8 A HIDDEN LINE PLOT FOR THE 300 kHz SONAR IN VERTICAL SCAN MODE. SHOWS SEA SURFACE TO LEFT AND SEA BED TO RIGHT WITH SOME MID-WATER FISH.

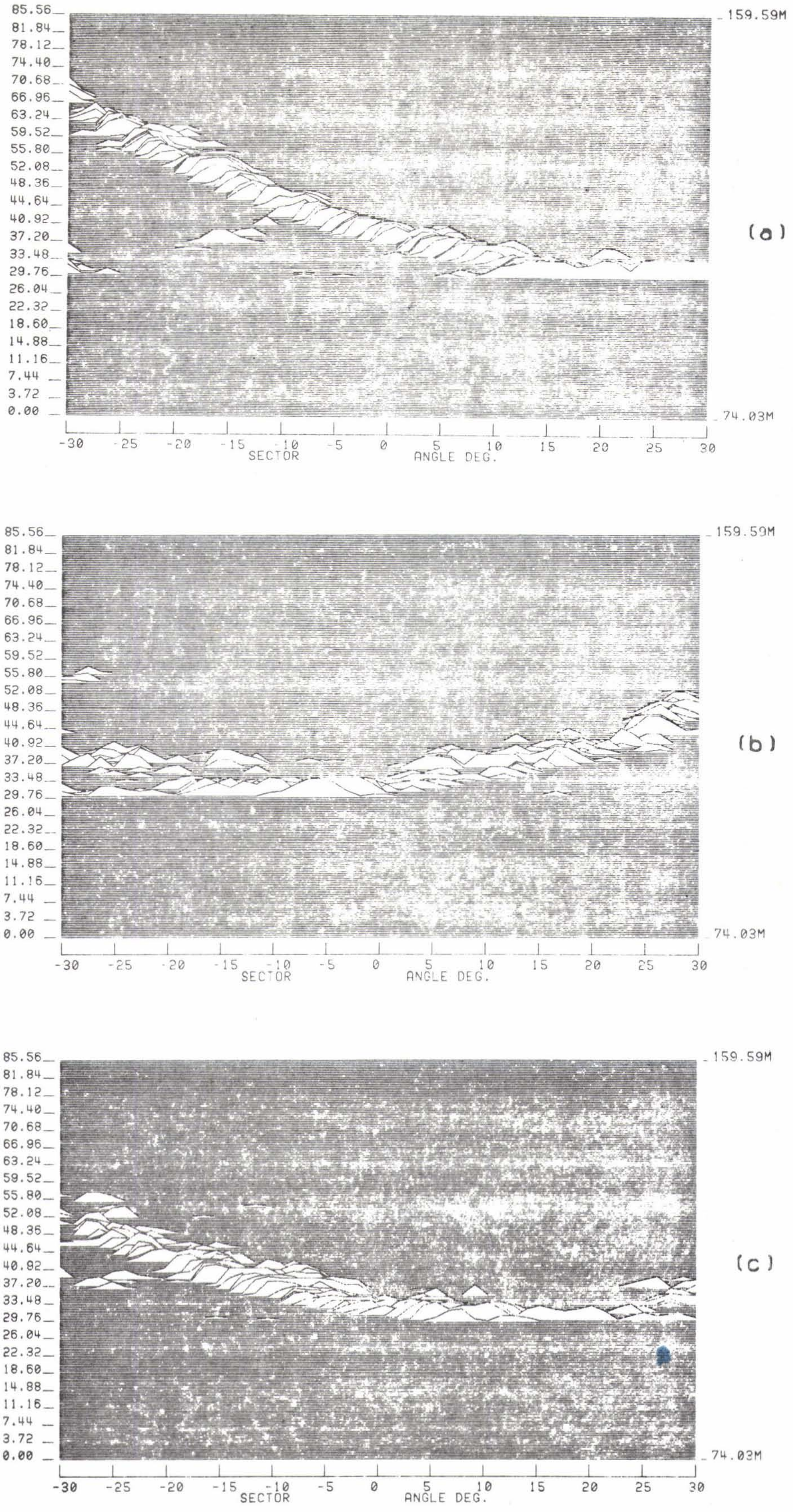


FIG. 9 HIDDEN LINE PLOTS FOR THE 75 kHz SONAR IN DEPTH SOUNDER MODE SHOWING DIFFERENT SLOPED SEA BOTTOMS.

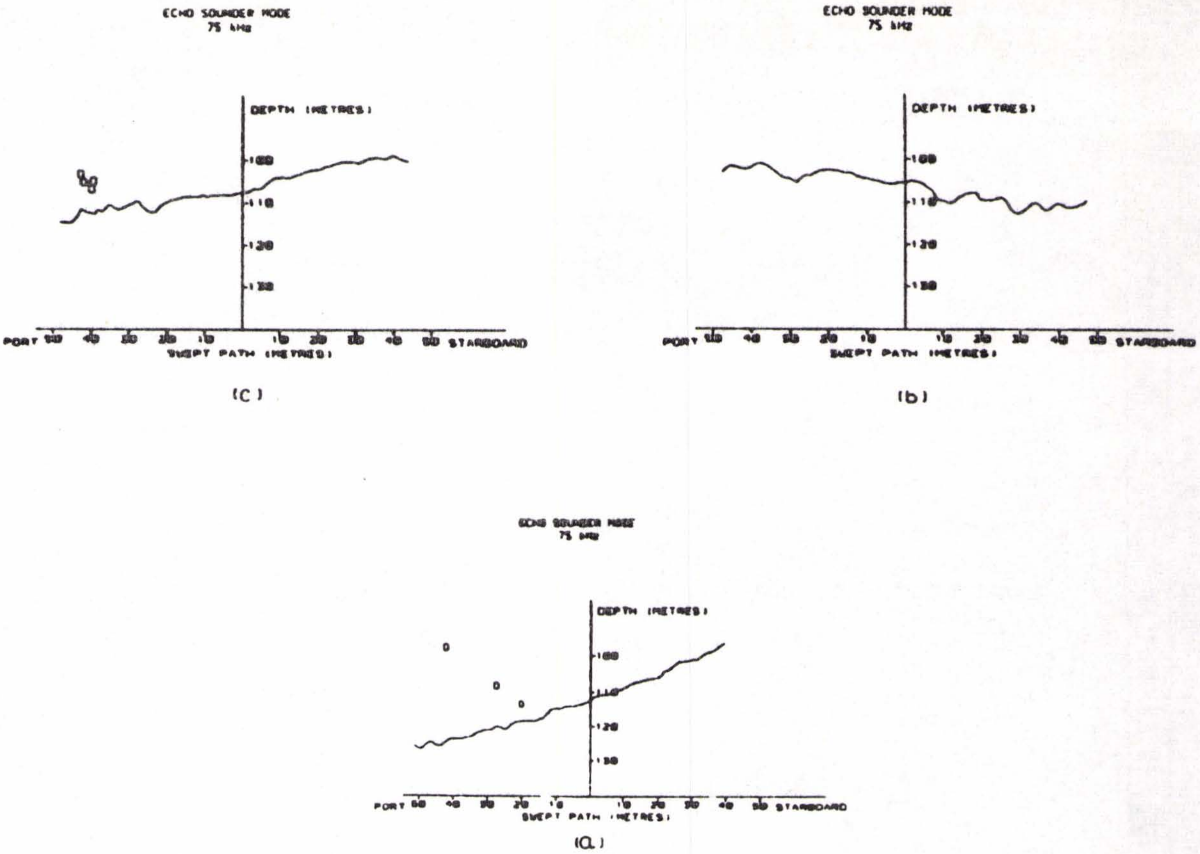


FIG. 10 BOTTOM PROFILES COMPUTED FROM THE DATA GIVEN IN FIGURE 9.

PANEL ON GENERAL PURPOSE SYSTEMS AND FUTURE TRENDS

Chairman· R. Seynaeve

Members : D.V. Crowe
J.M. Griffin
B. Pennoyer

R. Seynaeve We started this Conference with the Session on Systems, which included several papers describing high-speed signal processing systems. It was decided, however, to leave the Panel on this Session until the end of the Conference, after all the various components of the systems had been presented and discussed in their respective sessions. We have now reached this point and we will now go back to the systems and see how they may evolve. All the members of this Panel are personally involved in such systems and have presented papers on the subject. I propose that we each speak in turn about our current problem area, if any, and tell the audience how we see the future.

D.V. Crowe In our small laboratory we have been pleased with our processor in general. We can process and display a lot of data at high speed.

R. Seynaeve At SACLANTCEN the problem we are currently looking at is no longer speed, but how to efficiently set-up general purpose facilities for the display of results. It is not a simple problem if you want a flexible but still reasonably simple system, and it is also to a degree hardware dependent. Would Dr Pennoyer like to comment on this point as he has a display laboratory at NOSC.

B. Pennoyer Yes, true we have done a lot of work on colour and high-resolution black and white, and we are set-up for physiological testing. Perhaps colour doesn't help, but the colour jet plotter is interesting. The hard copy requirements may be underestimated.

R. Seynaeve An interesting point that came from the papers is that we all have used different architectures — NOSC's configuration is a star; the one of NRL is a multi-processor system requiring system programming: ours is a pipeline of subsystems, all programmable in high-level macro language. Is this difference due to different requirements or to different views on the proper system architecture?

J.M. Griffin Our approaches are dictated by use; we have a single-user tendency for sea-going systems, multi-users for lab.

D.V. Crowe I don't think our lab will buy another array processor, but would go to two-board processors. The feeling is that scientists won't use micro-assembler so the need is for high-level language.

R. Seynaeve Which two-board processor?

D.V. Crowe CD&A.

R. Seynaeve This needs system programming — surely scientists are not likely to do this.

J.M. Griffin It can be supported with Fortran calls.

D.V. Crowe We think we will evolve a high-level language — our scientists are quite capable of doing this.

R. Seynaeve Let us ask the scientists present here what their ideal system would be, or what is lacking.

A. Baggeroer More speed is needed.

R. Seynaeve Do we need more push-buttons or more system involvement from the user?

A. Baggeroer Scientists should work on the algorithm problem not display and I/O.

B. Pennoyer Flexible sea-going systems are nice, but if several analysts use one system there is a need to keep changing the configuration, which is a headache. Our approach is to give analysts a large flexible system that remains stable.

T.E. Curtis Isn't reconfiguring hardware no different, in principle, from loading tapes and moving boards?

B. Pennoyer O.K., but analysts like not to have to worry about the configuration they are using.

T.E. Curtis You seem to be going back, toward mainframe, while industry is going toward distributed micros. Analysts like their own little machine for 24 hours a day.

R. Seynaeve Yes, we too find that users like to be allocated a machine, and the less they share the better.

A. Baggeroer The problem is that they all want better and better systems in their corners.

T.E. Curtis Eventually every corner will have a Cray 1 or a Star 100 — rather expensive!

R. Seynaeve Could we establish a time constant for response to users' requirements. I find that 3-4 months is about the maximum a scientist is willing to wait to test his ideas with a proper system.

D. Nairn Our problem is booking a ship two years ahead. We are driven by trial slots not ideas.

A. Baggeroer We have the same problem with ship time.

G.C. Vettori A scientist with an idea must see it tested within a reasonable time, otherwise interest dies. We have experienced sometimes that hardware has had very little use because the scientist who requested it had left the Centre before it was finished. Computers have changed this, ideas can now be played with and tested within a few days. But, could it be that scientists are now becoming removed from the process?

R. Seynaeve Our aim is to give control back to the scientist in a similar way to that in which ITSA works. We are trying to make a language capable of giving full I/O control and display too.

D. Nairn I think that the limiting factor in deploying and changing operational system is rapidly becoming software documentation, and this could be a legitimate research area for SACLANTCEN

T.E. Curtis The American approach is trying to stop people from meddling with it.

E. Hug It could help to build documentation into the machine — push a button and get specs.

R. Seynaeve We already have this in some systems.

D.V. Crowe Industry tends to do this, for instance, DEC ECO documentation, and we will probably have to copy

R. Seynaeve What do you consider the weakest point in these systems? I feel that the weakest point in ours, at present, is the array processor – a more powerful one would be very useful. We also have a problem with sorting and presentation of the data and with the I/O.

J.M. Grifffin I would like to see mass memory improvements – there is need for large, fast memories. Available devices (8 Mbyte) must be addressed in block and we need random access for FFT.

T.E. Curtis The problem is word size for address. 32-bit minis are needed to use the memory.

J.M. Griffin Yes, that is why bulk memories from Monolithic and Dataram are disc emulators.

B. Pennoyer I agree that hardware, e.g. beamformers, will change considerably.

R. Seynaeve Yes our AP's were at first thought of as very fragile, but now they are becoming cheaper, smaller and more rugged.

T.E. Curtis I believe we are becoming ideas limited not technology.

SESSION II PROCESSORS

A MICRO-PROGRAMMABLE CORRELATOR FOR REAL-TIME RADAR PROCESSING

by

Hans-Jørgen Alker
ELECTRONICS RESEARCH LABORATORY (ELAB)
Trondheim, NORWAY

ABSTRACT This paper presents a novel design of a multibit, digital correlator for incoherent scatter radar observations. By utilizing bit-slice microprocessor elements and internal control by microcode instructions, a high-speed arithmetical preprocessor has been developed. Arithmetical and control operations are separated to obtain increased speed performance. The arithmetical part is optimized for complex auto/cross-correlation processing and has an effective rate of $24 \cdot 10^6$ multiplications/sec. Input data has 8-bit accuracy in integer format. The processor includes, at input, a 2-ported buffer memory for storing single/multiple channel inputs. Processing results are temporarily stored in a 4 K word high-speed memory before transferred to a general-purpose computer. Processing speed can be increased by adding up to 3 slave processors thus achieving direct parallel processing. Internal hardware is available for interface with standard CAMAC modules.

INTRODUCTION

During 1976-79 a feasibility study of a digital, multibit correlator for the EISCAT (European Incoherent SCATter) radar system was conducted. The main topics of the study embraced system design and hardware construction of a digital processor which fulfilled the specifications for real-time data processing.

The result of the study is a prototype construction of a high-speed multiprocessor system under interactive control of the EISCAT radar site computer. Support software for microprogram development, simulation and hardware testing are available for execution on a host computer.

The main objectives for constructing a digital processor for the EISCAT system were the ultimate requirements for special purpose data handling algorithms and the real-time processing speed. Remote probing the ionosphere using the incoherent-scatter radar technique has earlier been limited by available computational power. The ionosphere acts as a deep fluctuating target, and modelling is based on estimation of a range-dependent complex autocorrelation function (ACF) derived from the returned range-gated signal. Due to the low signal-to-noise ratios obtainable, real-time data reduction by time integration is required.

In the EISCAT radar the digital data acquisition system generates samples of the complex envelope (inphase and quadrature components in parallel) with typical sampling rates of 250 kHz for ion spectrum estimation. Assuming that typically 25 complex samples are contained in a range-resolution cell, the resulting requirement is $13 \cdot 10^6$ multiplications/sec. for real-time ACF processing. For reducing the time resolution of the radar experiment, the EISCAT radar utilizes multiple frequency transmissions and parallel channel processing in the receivers, increasing the requirement to about $104 \cdot 10^6$ multiplications/sec.

In 1976 a preliminary design of the EISCAT correlator was completed based on a structure using 50 hardware multipliers operating in parallel. The correlator fulfilled the requirements for multiple channel processing but had major limitations in relation to power consumption, system cost and hardware complexity.

This paper describes the architecture of a digital correlator system based on a redesign using an optimized time-serial approach for hardware construction.

1 DESCRIPTION OF CORRELATOR ARCHITECTURE

A block diagram of the correlator system is sketched in Figure 1. The hardware elements can functionally be split into two parts: The dual-channel data processing unit DPU and the control processing unit CPU. Both units are synchronized to a common system clock operating at 6 MHz.

The CPU generates all required control signals within the system. The main control unit is the microprogram sequencer which, at each clock cycle of the system clock, generates the value of the program counter (PC). This is the program memory location counter which maps the PC value into a microcode stored in the referenced location. With minimum hardware for instruction decoding, the microcode controls the function of the DPU on a cycle to cycle basis of the system clock. By feedback of the microcode to operate the sequencer, a predefined execution of a microprogram is possible. Two memory options are selectable for program source. With the RAM option (Random Access Memory), the correlator system is microprogrammable from an external source. Within the available program memory space, the RAM can contain several user-defined programs. With the PROM (Programmable Read Only Memory) option a library of predefined, non-destructable programs may be selected.

Each data channel of the DPU can be operated independently of the others. A data throughput rate of 6 Msamples/sec. is obtained by using a 5 stage "pipeline" in time-sequenced operation. The individual subfunctions of the DPU are given in Figure 2. The DPU is based on integer format arithmetic.

Input samples from buffer memory or external bus are 8-bits. The 4 hardware multipliers are operated in 8x8 bits and the final stage operates on 32 bits. Except for data overflow (hardware detectable), no rounding or truncation effects occur. The accumulators use the complete result memory as an accumulator register file.

The DPU hardware is especially designed for complex arithmetic processing. The buffer-memory is two-word addressable for transfer of values in complex notation.

Both address processors are designed with bit-slice elements enabling an effective base address/increment operation of the memories.

2 PROGRAMMING CONSIDERATIONS

An expanded block diagram of the microcode separation into "function"-fields are shown in Figure 3. The microprogram memory constitutes 2 pages with 64 words a 128 bits in each. A total number of 128 bits are used for all internal operations. The main processing task for the CPU is microcode generation (43 control lines) for controlling the data processing part, where individual operand/pipe-line registers can be strobed.

One 128 bit word of the program memory corresponds to single sample processing through all the stages of the data processing unit. This is achieved by individual delay of subinstructions of the microcode, and gives a more flexible programming of the system.

To save locations in the program memory and increase the execution rate of the microprogram, the system comprises 3 separately controlled hardware counters for program-loop execution (Figure 3). The microprogram sequencer contains 4 registers for storing return addresses when a subroutine call is executed.

Additional "background" operations have to be performed by the CPU. These operations involve control of the interfaces between the correlator and the host computer. The communication can be split into two basic operations: 1. Program load and parameter definitions from host. 2. Transfer of data from the correlator result memory to the host. Another interface is included for starting the correlator from an external trigger signal.

The control part of the correlator has an internal, programmable mode: STOP, which inhibits operation of the internal main clock. When transferring data on the host computer DMA channel, the internal clock is triggered for single instruction execution by the "data-received" signal from the host. This mode compensates for any speed mismatch between the host and the correlator.

The CPU has internal error detecting systems which mainly control numerical overflow of the DPU accumulators and the validity of external control signals.

During testing of the correlator prototype, the host computer programming of the correlator module was based on micro-instruction programming in absolute form (octal coding). In order to ease the programming tasks, a program CORSIM written in FORTRAN IV was developed on the EISCAT site computer (NORD-10). The CORRSIM is a multi-function software package for development and testing of user-defined microprograms. The system structure is shown in Figure 4. An editor is available for construction of programs and the submodule also includes a program source text generator after coding. The ARITEST and PROTEST modules give a simulation of the control and arithmetical part of the simulator at bit-level. The CORRIO module is the software driver for the correlator. Transfer out the correlator is via a standard CAMAC output module. The corresponding input from the correlator result memory uses a standard input module together with the CAMAC CDMA controller. A real-time program DMA-READ takes data from the driver and transfers it to the CORRSIM program. As given in Figure 4 several fixed-file programs are stored within the program. These are hardware test programs and are executed at regular time-intervals.

In the correlator module a PROM module containing a fixed sample set is located at the input of the DPU. This predefined sequence can be transferred to the DPU for processing and then transferred to the computer where a table comparison can be made with the same sequence through the software simulator. This option has been valuable during the final stability tests of the prototype.

3 HARDWARE CONSTRUCTION

The prototype is realized in LSTTL and NMOS technologies. The system is housed in a standard 19" rack section and the power consumption is about 100 Watts at 5 Volts. The number of cards is 8 (double European standard), where the control module requires 5 and the arithmetical part 3.

CONCLUSIONS

A microprogrammable system for real-time radar processing is described. The arithmetical part is optimized for complex processing algorithms.

Considering the correlator operational modes in the EISCAT system, the proposed structure makes it possible to accomodate most of the data produced in dual channel radar experiments. Provisions have been made for hardware expansion. In direct parallel processing only DPU parts need to be added, on advantage given directly by the microcode structure.

DISCUSSION

R. Seynaeve What was the manpower allocated to the project and how long did it take?

H.J. Alker Prototype design and construction has taken two years and involved a small group of two/three engineers.

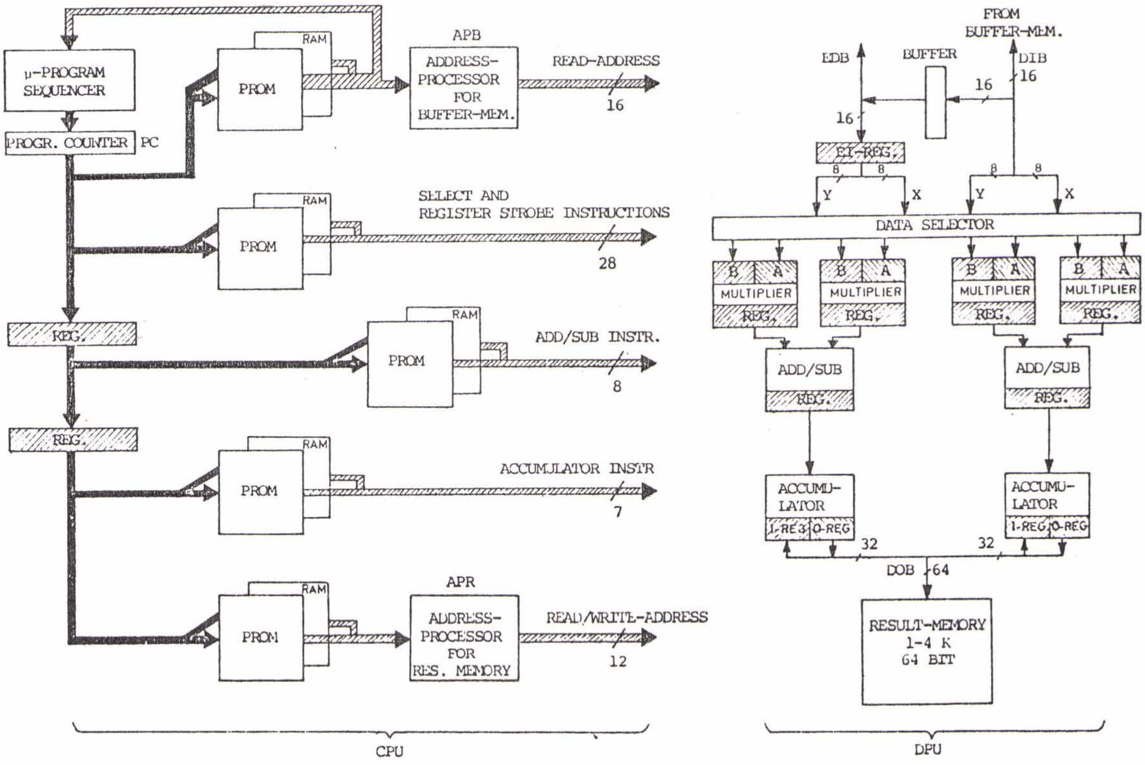


FIG. 1 CORRELATOR ARCHITECTURE

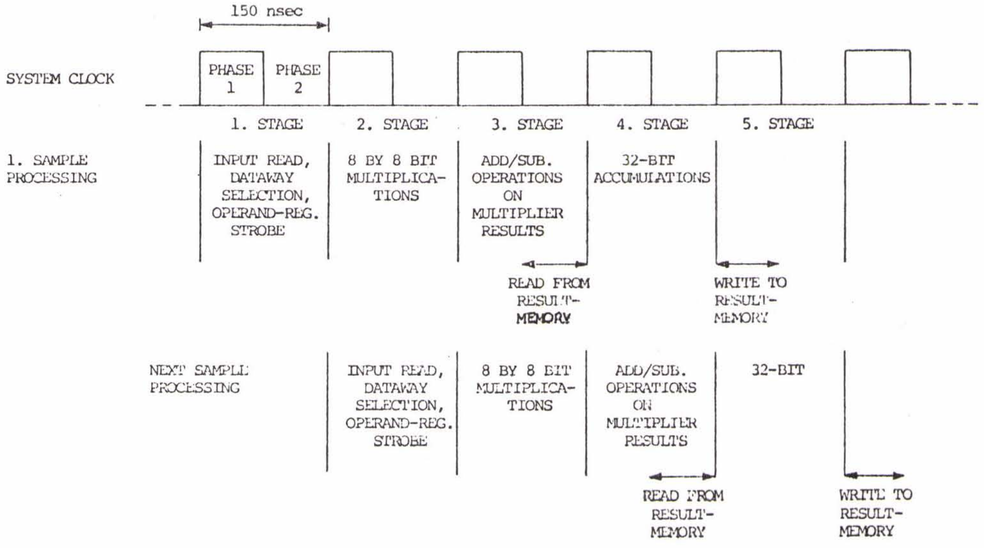

150 nsec
SYSTEM CLOCK
PHASE 1
PHASE 2
1. STAGE
2. STAGE
3. STAGE
4. STAGE
5. STAGE
1. SAMPLE PROCESSING
INPUT READ, DATAWAY SELECTION, OPERAND-REG. STROBE
8 BY 8 BIT MULTIPLICA-TIONS
ADD/SUB. OPERATIONS ON MULTIPLIER RESULTS
32-BIT ACCUMULATIONS
READ FROM RESULT-MEMORY
WRITE TO RESULT-MEMORY
NEXT SAMPLE PROCESSING
INPUT READ, DATAWAY SELECTION, OPERAND-REG. STROBE
8 BY 8 BIT MULTIPLICA-TIONS
ADD/SUB. OPERATIONS ON MULTIPLIER RESULTS
32-BIT
READ FROM RESULT-MEMORY
WRITE TO RESULT-MEMORY

FIG. 2 TIME-SEQUENCED OPERATIONS OF THE DPU

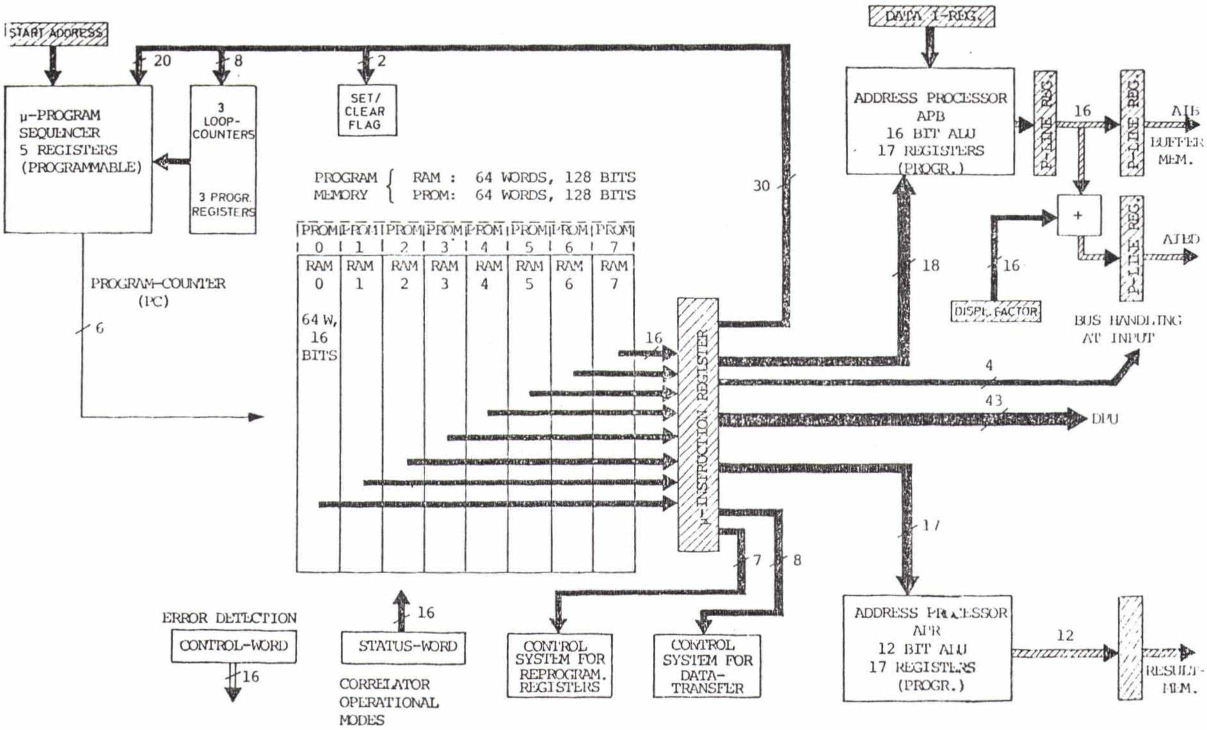


FIG. 3 BLOCK DIAGRAM OF CPU HARDWARE

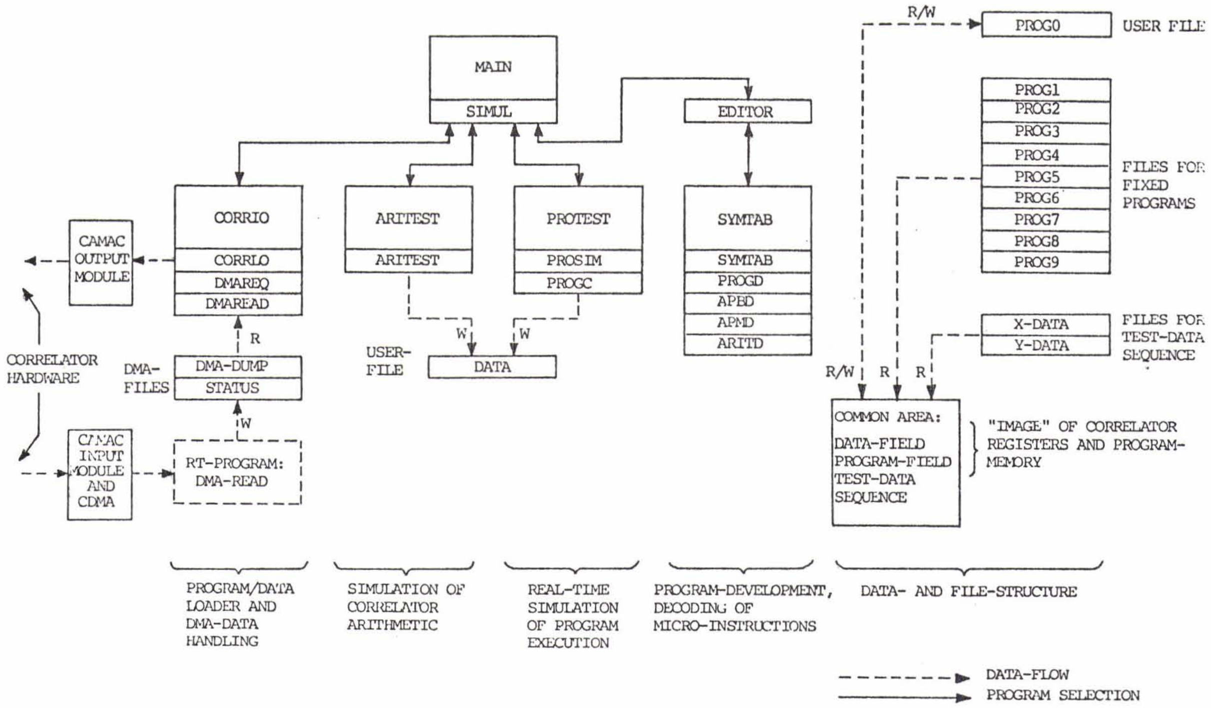


FIG. 4 CORRSIM STRUCTURE

MARTINUS - MULTIPROCESSOR FOR HIGH CAPACITY REAL-TIME PROCESSING

by

Yngvar Lundh
Norwegian Defence Research Establishment
Kjeller, Norway

1 INTRODUCTION

Development of circuit technology in recent years now makes it feasible to build powerful processing machines at quite considerable savings - in comparison to those which have been considered feasible earlier. These savings are such that many real time processes are now practical which were not so before. By utilizing network structures of "computing elements" in which a number of computing elements are connected and so organized that they are able to share the total computing load between them, it is possible to move the limits of the magnitude of processing tasks which can be handled, very far.

Following is a description of a machine, which we shall refer to as a multiprocessor. It is programmable similarly to a "monoprocessor" computer. However, several features of multiprocessor programming are different from, and additional to those of conventional (monoprocessor) programming.

This structure is referred to by the name "Martinus".

2 MULTIPROCESSOR STRUCTURE

2.1 Process partitioning

First of all, the signal process is broken down into parts which are manageable by single computing elements. Two methods are available for this, alone or in some combination: Division of a process into "processing steps", and division of a processing step into "shares". Figure 1 shows these subdivisions.

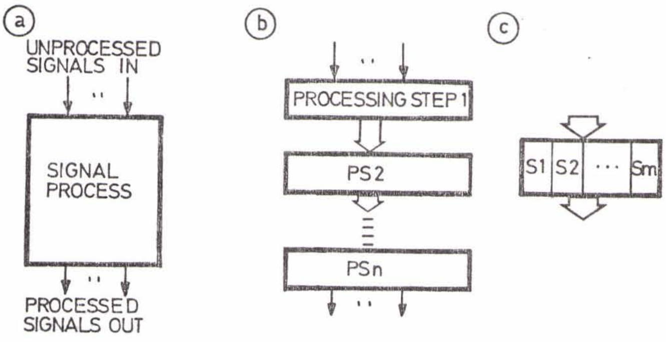


Figure 1 Breaking a process down into manageable parts or steps in a data flow

Figure 1 a) and b) indicate how n processing steps constitute the entire process. Signals are "refined" step by step until they emerge completely processed from the n'th and last step.

A process step may be such that it can be "shared". One computation program or "algorithm" may be executed m times, each time with a different set of input variables, to perform the entire step. This, of course, is much like an ordinary program loop. However, here we are concerned with shares which can be executed independently by separate computing elements if need be, to cope in real time.

2.2 Network components

The multiprocessor is constituted of devices which can communicate with each other. All communication is in the form of "packets" of information being sent over buses, see Figure 2. To each device, the bus effectively means a direct channel to each of the other devices connected to the bus. That means, a channel with enough transmission capacity to carry all traffic without need for each device to wait. In actual fact, the bus is a time shared switch. Each device is connected through a bus interface circuit which comprises a buffer capable of storing one incoming and one outgoing packet. The bus circuits are fast, and comprise "transmission arbitration" such that the transmission capacity normally is limited by the devices themselves.

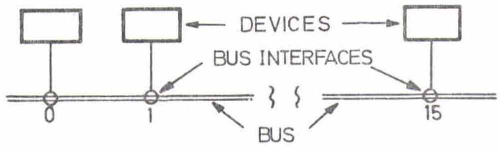


Figure 2 Devices are interconnected by buses

In order to achieve the required bus circuit speed and traffic handling capacity, there is a limit to the number of devices on each bus. The maximum is set at 16 devices.

2.3 Network structures

An arbitrary number of devices may be interconnected in a network. Each device has one or more ports for connection to one or more buses. An example network is shown in Figure 3.

Physically, the same network may appear as the equipment outlined in Figure 4.

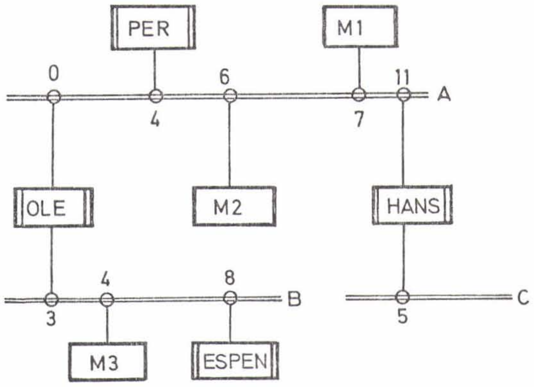


Figure 3 Example network

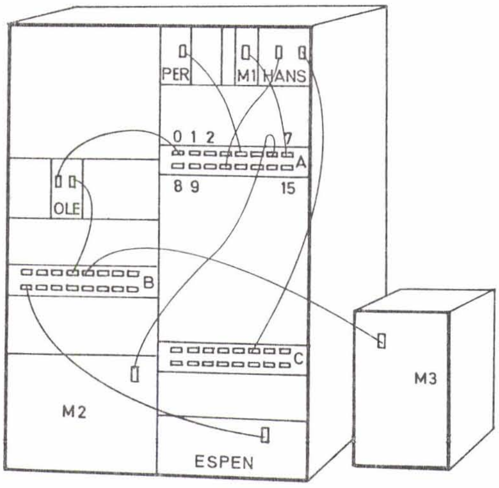


Figure 4 Example physical configuration

A bus appears to the outside as a panel with 16 multipin plugs. A device has a corresponding multipin plug for each "bus port". Connections are "patched" by multiwire cables. Drivers are included in the bus interface and bus port circuits to allow these cables to be up to a few meters in length.

A multiprocessor may be configured to be particularly suitable for a given process, e.g. in accordance with the data flow type architecture indicated in Figure 1. Note however, that we can configure our network any way we want. The structure is limited only by the number (16) of bus interfaces on each bus, and by the number of bus ports on each device.

2.4 Types and functions of devices

The typical device is either a computer called "computing device", processor or computing element, or a memory with controller called "storage device" or external memory. More specialized devices are also possible.

In fact, there is no other constraints on the types of devices which may be connected to a bus than the requirement to comply with the formats, control signals etc for data exchange with the bus interface. This format etc is called the "bus interface protocol".

Each device thus may have a special function - or it may also be a general programmable computer. Different types of devices may be used in any configuration to suit the needs of the process. To the buses all devices look alike in that they comply with the bus interface protocol.

2.5 Network interface protocol

Devices send information to each other in the form of packets. Each packet consists of a variable number, minimum 1, maximum 16 16-bit words plus a 4 bit destination device address.

Higher levels of protocol are used for the actual use of packets and the formatting of information in them.

One 16-bit word is transferred at a time either to or from a device. 16 data wires and 7 control wires are used for connecting each device.

3 MULTIPROCESSOR PROGRAMMING

3.1 Stepwise processing and intermediate storage

Each processing step results in a set of intermediate data. Between steps, data are stored in storage devices.

Computing elements typically have memory also - both for program and data. Storage devices are devices capable of just information storage and retrieval. Figure 5 outlines an example of how a process comprising three process steps PS1, PS2 and PS3 is to be performed by one or more processing elements each, and how intermediate data are stored in separate storage devices.

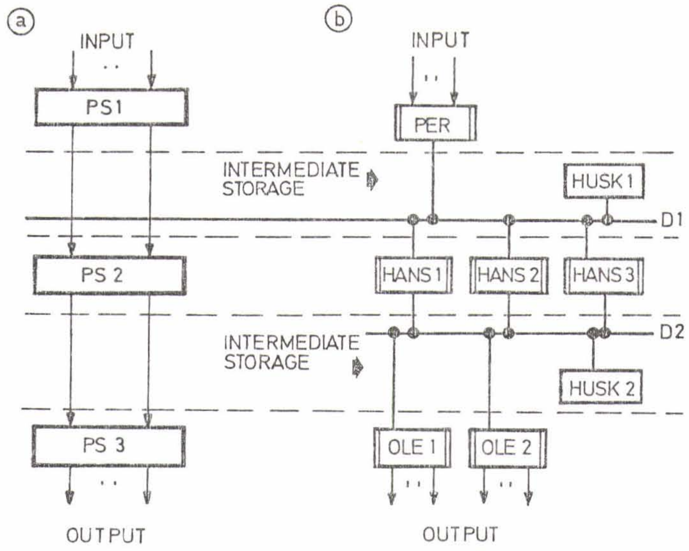


Figure 5 Process steps assigned to devices

In this example, the first step is performed by device PER. The resulting intermediate data from PS1 are stored in storage device HUSK1. The task of performing PS2 is shared by devices HANS1, HANS2 and HANS3, which fetch their input data from HUSK1 and deliver their resulting intermediate data into storage device HUSK2. Devices OLE1 and OLE2 share the task of performing PS3. They fetch their input data from HUSK2.

Note that we have shown computing devices and storage devices by slightly different symbols in the schematic diagram. To the bus they are all equal, i.e. all devices use the same bus interface protocol. The differences in their task, are noted by the programmer in this way.

Nets of many devices may require more than one bus, because of the limitation on the number of devices on each bus. Figure 5 shows how multiple buses can be used.

In addition to limitation of the number of interfaces on a bus, there is also a limit to the amount of traffic which the bus can carry. To avoid traffic congestion as a limiting factor, under-utilization of the bus connectivity may sometimes be required.

3.2 Supervision

Management of the multiprocessor is performed by a "supervisor"-computing device. It normally uses a separate bus for "direct lines" to all the other computing devices. Figure 6 shows how a supervisor and a supervisor bus are added to the net shown in Figure 5. Depending on the number of devices to be connected, more than one "supervisor bus" may be required.

Depending on the size of the processing load which the real time supervisory tasks represent on the supervisor processor, it is also possible for the supervisor to leave some of its task to "lieutenant supervisor processors", thus forming a hierarchy of supervisor processors.

The management tasks performed by the supervisor comprise: Programming, sequencing and self checking.

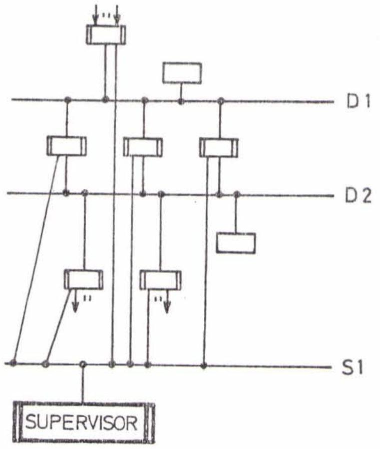


Figure 6 Net comprising supervisor

"Programming" means accepting program statements in symbolic form (source code), converting them into binary code, loading binary programs into the computing devices, and starting execution. Further, it comprises debugging facilities.

The sequencing or synchronization of tasks in computing devices comprises start and stop of program execution on an individual basis as required in continuous real time operation. Part of the sequencing task is to keep track, at all times of the validity of all intermediate data in storage devices.

Self checking is done by a separate set of "passive" programs running (in "background") in the supervisor and all other processors. They are called passive because they are executed in the spare time of the processors, that is the time when active programs, i.e. the signal process itself, can not be run. All processors are arranged to have some spare time in which to run these self checking programs. The purpose of self checking is to detect unexpected or abnormal conditions and report them to the programmers or operators of the signal processing machine.

"Programming" typically takes place prior to the machine starting real time operation, while "sequencing" and self checking are part of the real time operation itself.

The supervisor includes secondary storage (disk memory), and operating system, with multiterminal timesharing, file system etc, and a set of terminals for the programmers and real-time operational personnel. Included with the system programs in the supervisor, are a text editor, cross-translation programs (assemblers and compilers) for each type of computing device, and a "book-keeping system" which facilitates handling of such information as variable names and addresses of intermediate data, timing requirements etc. Also included is a command system which permits operators to select options, change parameter settings, etc during real time operation. Thus, the supervisor is to the multiprocessor what the operating system is to the monoprocessor.

4 COMMUNICATION FEATURES

Intra-communication in the multiprocessor, that is communication between devices, is carried by the buses between devices. The bus traffic comprises data and supervision traffic. The latter, to and from the supervisor, concerns the management functions, including, programming, sequencing and self checking. The former concerns the data representing the actual signals being processed, and which move between computing and storage devices as they are being processed step by step.

All traffic moves in the form of packets as discussed earlier. Inside each packet is information which permits the receiving device to distinguish between different types of incoming packets. First of all there is a need to separate supervision from data traffic. Further there are many types of packets which may occur within each of these types.

Information coding takes place according to a hierarchy of "protocols". The lowest protocol level of interest to the programmer is the bus interface protocol. Inside each packet, data may be identified as types of "data items". Examples of standard data item types or "data formats" are "16 bit word" or "complex number with 32 bit, fixed point precision". Data transfer takes place either as single packet exchanges or as "block transfer" in which a larger amount of data (many packets) is handled more efficiently. For the programmer, there is a standardized set of statements which he can include in symbolic processor programs, and which effect appropriate "packing and unpackaging".

5 IMPLEMENTATION

The general structure of such a multiprocessor, and the ways in which it is programmed have been designed to allow a considerable growth potential.

That means, many types of processing and storage devices can be used together in one multiprocessor. In other words, if as an example, circuit technology permits or a special application demands a new type of processing element, there is a straightforward method in which such a new type of device can be introduced. It does not have to effect the other devices and the way they are programmed and operated. All it means is that new device types may be readily accepted. Similarly, new buses with greater speed may be introduced, when circuit technology becomes available, which look like the old ones to the devices, but which may carry much more traffic - and thus may be commensurate with more powerful devices having greater throughput. None of these changes will require changes in the system programs.

Figures 7-9 show some examples of actual equipment, part of a machine under construction capable of some 120 million instructions per second.

This implementation permits restructuring of the interconnection of devices. This is interesting for large tasks where optimization of the structure, in view of the particular real time signal processing task at hand, is important for economical reasons.

This, of course, does not prevent a fixed structure to be built, which will never be changed. Such a permanent structure could be optimized for a class of problems, and convenient utility software could then be made more efficient and easy to use for such a multi-use type of machine.

6 SUMMARY AND CONCLUSION

An overview has been given of the main principles of a multiprocessor which has been particularly designed for the performance of real-time signal processing algorithms.

The principal objective is to achieve arbitrarily high processing capacity by combining the computing power of many computing and storage devices of limited capacity.

To design and implement programs for such a multiprocessor-machine imply many considerations which are not part of "monoprocessors". Part of the description here concerns methods for handling the additional complexity of a multiprocessor.

Due to the availability of more powerful circuit techniques, the implementation of such multiprocessor systems are now much simpler than before. Such complex structures thereby have become feasible, and make many high capacity signal processing tasks practical to implement.

REFERENCES

1. LUNDH, Y. Multiprocessor system for high-capacity signal processing, Proc. International Conference on Computer Communication, Stockholm (August 1974).

DISCUSSION

D. Nairn What speed is the bus?

Y. Lundh One 16-bit word in ~100 nsec.

Y.S. Wu Have you simulated the supervisor load? What is the capacity?

Y. Lundh No. The capacity of NORD 10 is 1 million instructions per second.

R. Seynaeve Could you compare this structure to CRAY-1 type.

Y. Lundh I am not prepared to discuss or compare other people's machines. It may, however, be of interest to note the fact that this reconfigurable system greatly facilitates the possibility of achieving high efficiency in the utilization of the available processor cycles (per second). The total available number of processor cycles, of course, is determined by the number of processors — a number which can be chosen to meet the needs of a given task.

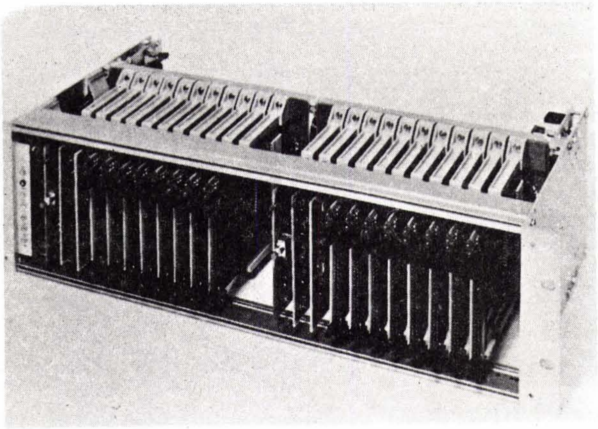


FIG. 7

ONE TRAY HOUSING TWO BUSES, EACH HAVING 8-BUS INTERFACES. LAMPS INDICATING "OUTGOING PACKET" AND "INCOMING PACKET" ASSIST ERROR TRACING.

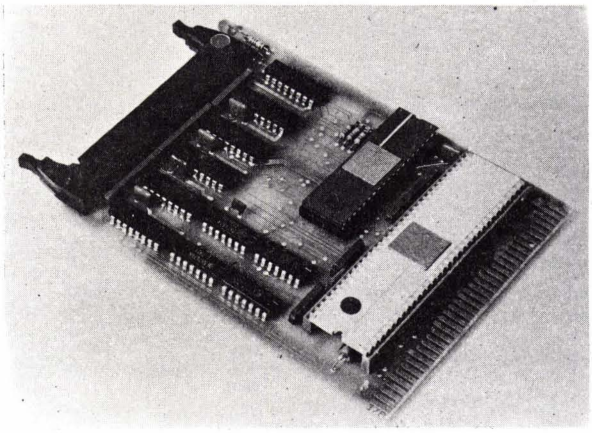


FIG. 8

ONE BUS INTERFACE CARD COMPRISING PACKET BUFFER STORAGE, TRANSMISSION AND CONTROL CIRCUITS.

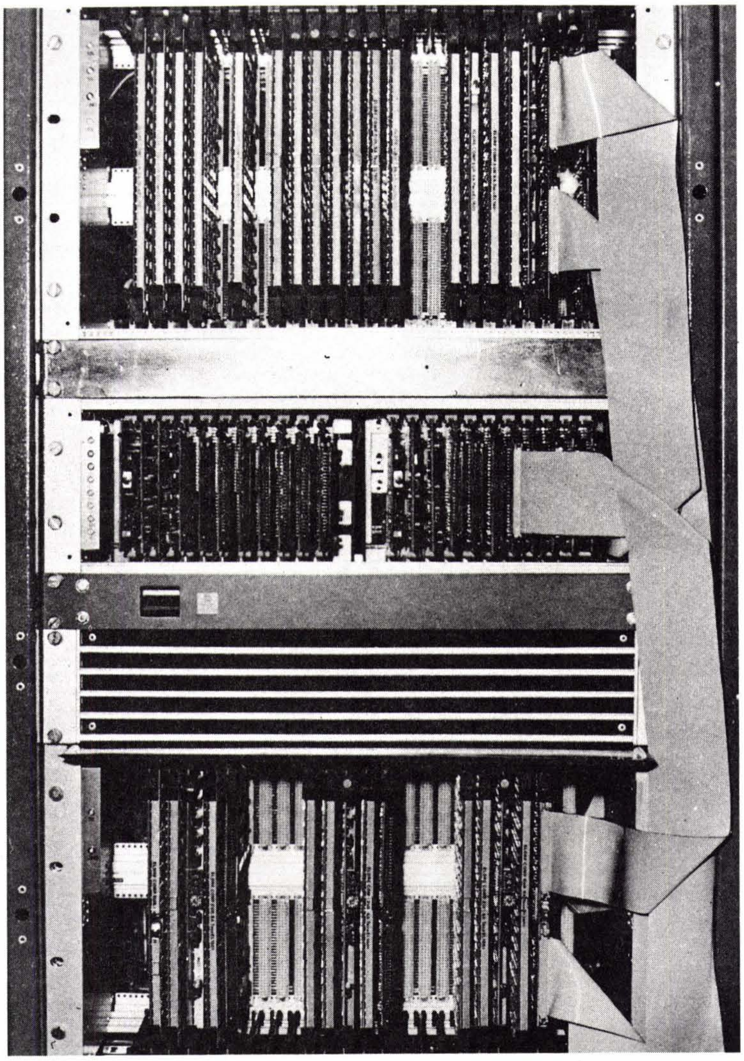


FIG. 9

SECTION OF A MULTIPROCESSOR SHOWING HOW DEVICES AND BUSES ARE INTER-CONNECTED BY MULTIWIRE "FLAT-CABLE" AND PLUGS.

A MODULAR APPROACH TO SIGNAL PROCESSING

by

T. E. Curtis
Admiralty Underwater Weapons Establishment
Portland, Dorset, UK

ABSTRACT This paper outlines work in progress on modular systems for beamforming and signal processing. The modules described were developed using commercially available digital L.S.I. circuits and the emphasis during development has been on ease of use and overall flexibility, so that the modules can be applied effectively to a wide range of different sonar problems. The modules themselves are essentially hardware blocks, but they are under the control of a microprocessor which enables the function of the module to be microprogrammed for particular applications. The power of the system is matched to that required for a particular application by use of a parallel processing architecture, so that the system can be developed cost effectively.

1. BACKGROUND

Development of sonar systems in the past has tended to fall into two distinct classes. Up until the advent of the mini-computer and L.S.I. most systems were technology limited, and it was necessary to custom design specific hardware blocks to perform particular functions. This process was time consuming and costly and, because the system requirements varied from one application to another, it proved difficult to carry over hardware design from one equipment to the next. The mini-computer, and later the microprocessor, tended to change this in that it became possible to build flexibility into systems. This resulted in a number of developments where system design effort was vested in the production of system software. Surprisingly these systems have proved to be as difficult to develop and modify as custom hardware, possibly due to the fact that many system engineers have a basic mistrust of software, whilst most software engineers have limited system experience. The work described here steers a mid-course between two ends of the system spectrum, and exploits the recent developments in technology to perform the required processing functions in L.S.I. circuits that operate under simple software control in an attempt to achieve the sheer processing power of custom hardware with the flexibility of software based systems.

The approach used to define the particular hardware structures developed was initially to define the system requirements in fairly broad terms. Next, various methods of realising such systems were considered, and from these the technique that, at that time, appeared to be the most widely applicable to sonar processing, was selected. Practical hardware systems were then designed to meet these requirements, with the emphasis on flexibility and cost effectiveness.

For convenience, spatial array processing and temporal signal processing were considered separately, although as will become apparent, the above approach leads to hardware structures that could fulfil both requirements.

2. BEAMFORMING

The conventional beamforming problem is defined in Fig. 1, an arbitrary array of elements receiving energy from some source. The natural output from the array, given by the summation of all element outputs, will have the form:-

$$\sum_{N} k_n \cdot f \left\{ t + \left[f\ (n,\ E,\ A) \right] \right\}$$

The beamforming problem is then apparent: the required array output

$$\sum_{N} k_n \cdot f\ (t)$$

is modified by the geometry dependent delay function $f(n,E,A)$.

This delay term must be cancelled in order to steer the array to look preferentially in the source direction. The techniques to achieve this fall broadly into two classes, viz transform beamforming and time delay beamforming. Transform methods include wave number and wave function analysis as well as frequency domain realisation of real time delay systems.

In general, the array may be of arbitrary geometry, often in two or three dimensions, with non-uniform element spacing. A number of individual beams must be available to look in various directions, to provide good angle of cover as well as good angular resolution: each beam must be independently steered, both in order to reconfigure the system and to compensate for array motion, so that beam pointing directions can be stabilised inertially in space. In many applications, adaptive processing must be provided to allow cancellation of interfering sources. Whilst transform processing provides an elegant way of beamforming from arrays of particular geometries, in the general case outlined above it becomes less attractive; this is mainly because the system compaction that can be achieved via F.F.T. processing cannot be applied readily to stabilised beamforming from arbitrary array geometries. In this case, the transform approach then reduces to frequency domain realisation of real time delay systems, and from the system control viewpoint, direct real time delay systems are more attractive.

2(i) Space-Time Beamforming

The conventional method of time delay beamforming is shown in Fig. 2: for small multibeam systems, a number of such delay/summation matrices can be fed in parallel from the array, but this results in a highly redundant system, due to the number of parallel delay channels and the control system becomes confused when large, stabilised systems are considered.

To simplify the system requires a slightly different approach: consider that all the element data from the array feeds a store that then holds a running time history of the array. This store is arranged so that it maps to the array elements directly, and the time history stored is made equal to the acoustic array length. A sample of any required beam can then be formed by generating an address plane across the store and summing the accessed data.

This space-time matrix concept is easily extended to two dimensional arrays: this is shown schematically in Fig. 3, where beams can now be steered in both azimuth and elevation. It can be further generalised to deal with non-planar arrays of arbitrary geometry, eg. conformal arrays, by simply mapping the address plane on to the complex array geometry plane. Array motion can be sensed also, using for example a set of accelerometers, and this data used to correct the beam address plane to compensate and inertially stabilise beam pointing directions.

2(ii) Practical Systems

A system schematic based on the above is in Fig. 4: this shows the block diagram for just one channel of the space/time store. Operation is as follows:- at some time t_o, the store is updated with element data, written into a location defined by the current write address. This write address defines the"TIME NOW" plane in the store, so all read accessing is performed relative to it. Between write updates, samples of the correctly delayed element data are accessed by adding a beam address increment, defining the required element delay, to the write address for each required output beam sequentially. Each output sample is multiplied by its corresponding weighting coefficient and the result added to outputs from adjacent channels, to form a sample of the required beam. Address increments and weighting coefficients are conveniently read from some other store, adddressed by a beam number. This store can be either PROM for a fixed beam system, or RAM preloaded with data resulting from on-line computation.

Using current generation static RAM and L.S.I. multipliers, such systems can be readily operated at clock rates up to 10 MHz, forming beams at the rate of one sample per 100 nSec. For a typical system with say 64 output beams, this allows multichannel beamforming with output bandwidths for each beam in excess of 50 kHz. This is often well in excess of that necessary, and this speed can be traded for system cost.

To minimise cost, it is necessary to multiplex the multiplier over as many channels as possible: this also reduces the amount of address generation logic, and allows the channel summation function to be performed sequentially using L.S.I. multiplier/accumulators. A block schematic of this multiplexed system is in Fig. 5: operation is basically identical to that shown in Fig. 4, except that address and coefficient stores are addressed by a combination of element number and beam number. Further flexibility is achieved by using a RAM control store that can be preloaded to define control and timing functions in a similar manner to the micro-instruction store in a microprocessor. This system can be produced using around 10 DIL packages, with a maximum processing speed of typically 100 nSec per element per beam. Two such systems fit on a Double Euro Card, dissipating around 11 watts: this card will handle all the digital processing to generate 64 half-beams from a 32 element array with an 8 kHz bandwidth.

Larger systems are configured by operating the necessary number of such cards in parallel, together with a control card and microprocessor that loads the individual coefficient, address and control store with data from on or off-line calculations, to set up beam look directions, beam shapes, etc. The beamforming hardware thus essentially acts as a microprocessor peripheral so that hardware functions can be adjusted by software control.

3. SIGNAL PROCESSING

The functions required for sonar processing are numerous; they include filtering, band shifting, correlation, frequency analysis, etc. Consideration of these processes indicates that they can all be achieved using a combination of data storage, cross multiplication of data streams by coefficient streams and integration of multiplier products over defined data fields. Consequently, a general purpose processor can be conceived that uses these basic hardware functions, and that can be configured, via software control and coefficient stream generation, to perform the necessary processes; this leads to a card structure similar to that described in Section 2. This is shown schematically in Fig. 6. Since many channels of identical processors are needed to process data from adjacent beams, coefficient and control streams are common to a number of channels, and separate on-card control and coefficient stores are not required. In-phase and quadrature coefficient multipliers are used, so that signal band-shifted to baseband can be processed without phase dependence.

Practical systems have been based on 12 bit parallel multiplier/accumulators and static RAM, and require around 14 DIL packages. Working at a 5 MHz clock rate, this card can perform 1024 point running block, real time, direct realisation of correlation and DFT with a 2.5 kHz band width, or a fixed block 1024 point FFT in about 10 mSec, allowing real time frequency analysis bandwidths of 50 kHz, and frequency domain realisations of replica correlation with a 25 kHz bandwidth.

For systems use, the cards are organised on a 4 buss architecture, shown schematically in Fig. 7: this enables the required number of cards to operate in parallel under microprocessor control. Input and output data highways are multiplexed in a word parallel/channel serial format, and control and coefficient busses are common to all cards. For filtering, bandshifting and FFT analysis, a number of channels can be handled by a single processor card; for DFT or time domain correlation analysis, one card per beam is used.

4. SUMMARY

Some of the techniques currently in use to process sonar signals digitally have been briefly outlined; they rely heavily on L.S.I. and V.L.S.I. technologies. The scale of integration available in such digital devices now makes possible systems that were inconceivable only a few years ago. For example, the complete processing for a 1000 element array with 128 output beams can now be housed in 6 levels of a standard 19" cabinet — orders of magnitude smaller in size, and hence cost, than systems based on M.S.I. technologies. As a result of this, sonar processing is emerging from the technology limitations that were inherent until recently, and is becoming ideas limited. However, these systems depend on the continued availability of American and Japanese device technology, as many of the components required are not available from U.K. device manufacturers.

DISCUSSION

H.J. Alker I see that you have used the TRW components for signal processing. Could you give a comment on the use of custom-designed components, and what is the situation in the U.K?

T.E. Curtis The single source situation was initially a worry, but a number of U.S. manufacturers are now in the field. We have taken a deliberate decision to use commercially available LSI components and have not pursued custom-designed devices in the U.K. for this application.

V. Cappellini What type of microcomputer is used in the parallel processing architecture?

T.E. Curtis The Texas Instruments TMS 9900 — mainly because the on-chip multiply/divide was useful in many applications, although we have now added an AMD 9522 as a floating point processor for many arithmetic functions.

R. Seynaeve To what level do modules remain programmable?

T.E. Curtis To whatever level you need in a particular application — for example in the beamforming case, delay and shading can be specified for each element in each beam.

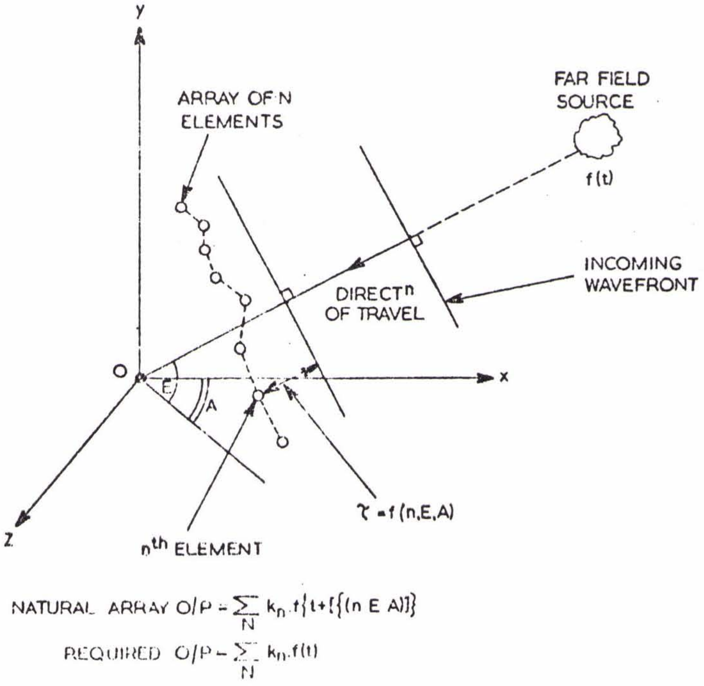


FIG. 1 THE BEAMFORMING PROBLEM

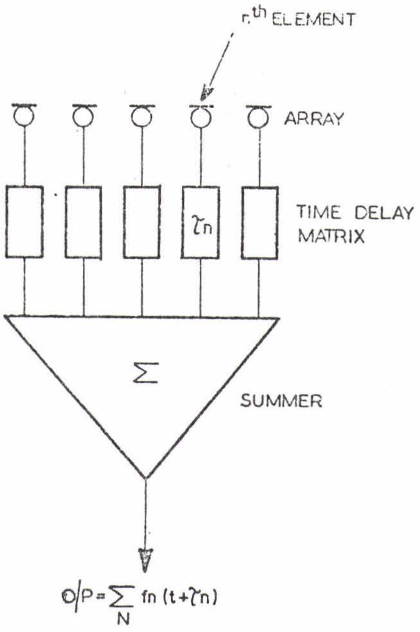


FIG. 2 BLOCK SCHEMATIC OF CONVENTIONAL TIME DELAY BEAMFORMER

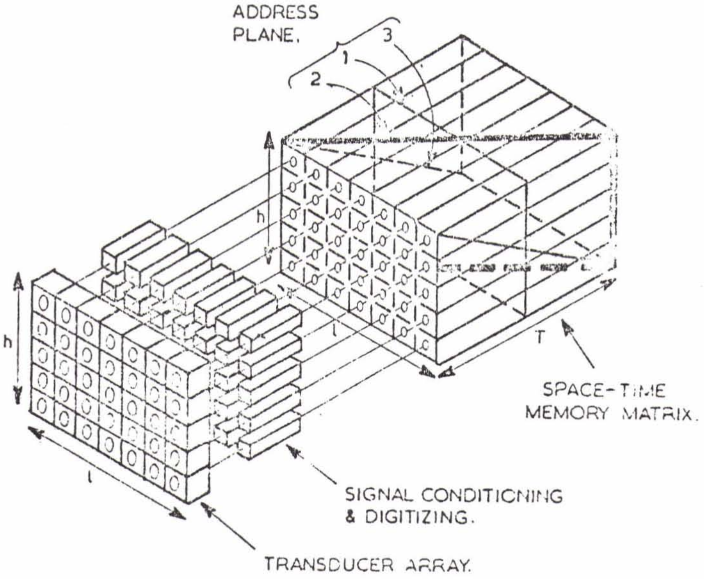


FIG. 3 SPACE TIME MATRIX BEAMFORMER

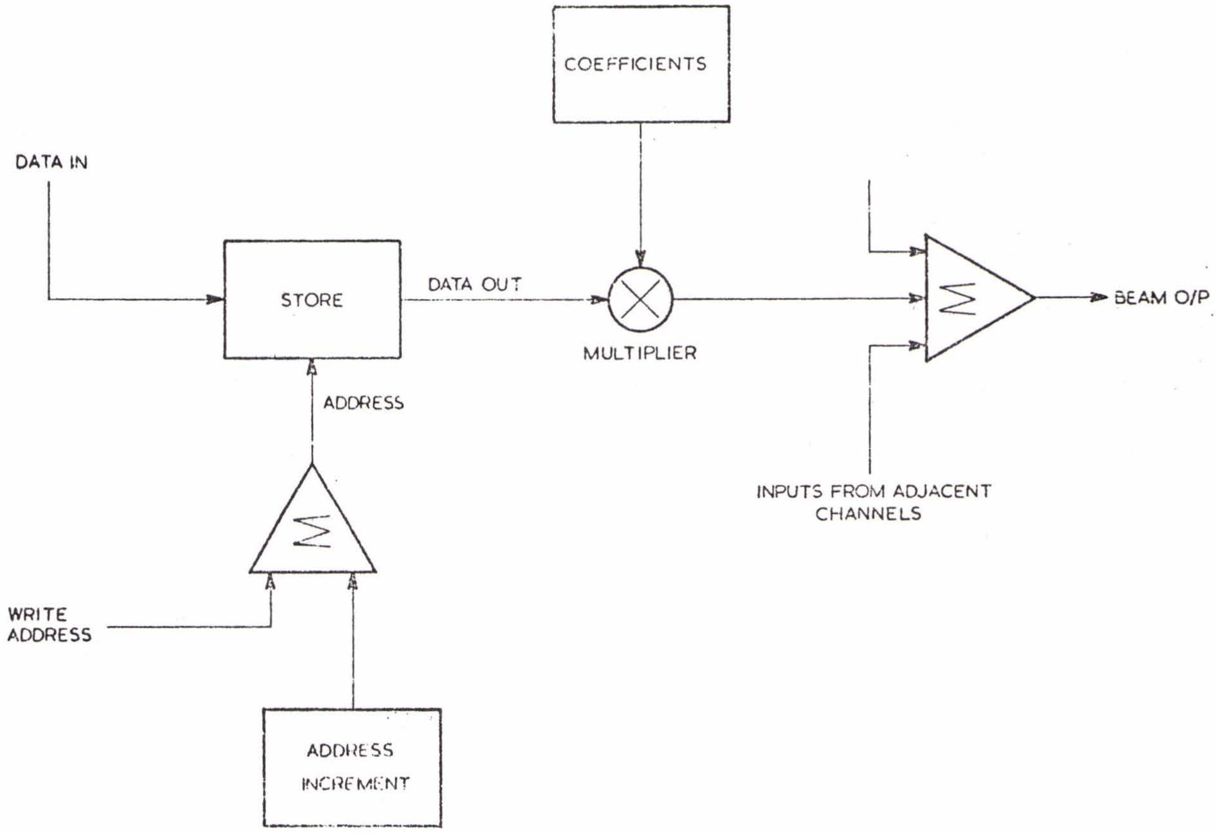


FIG. 4 BEAMFORMER CHANNEL SCHEMATIC

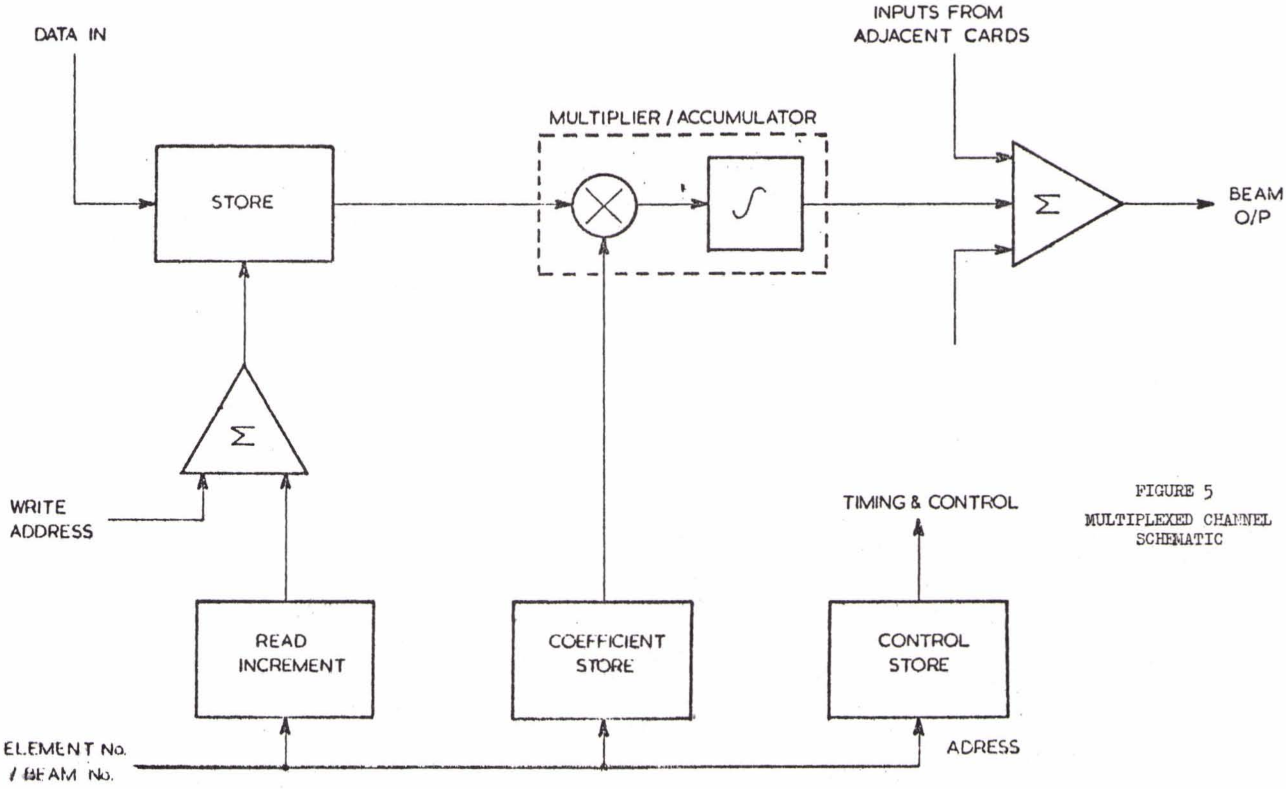


FIG. 5 MULTIPLEXED CHANNEL SCHEMATIC

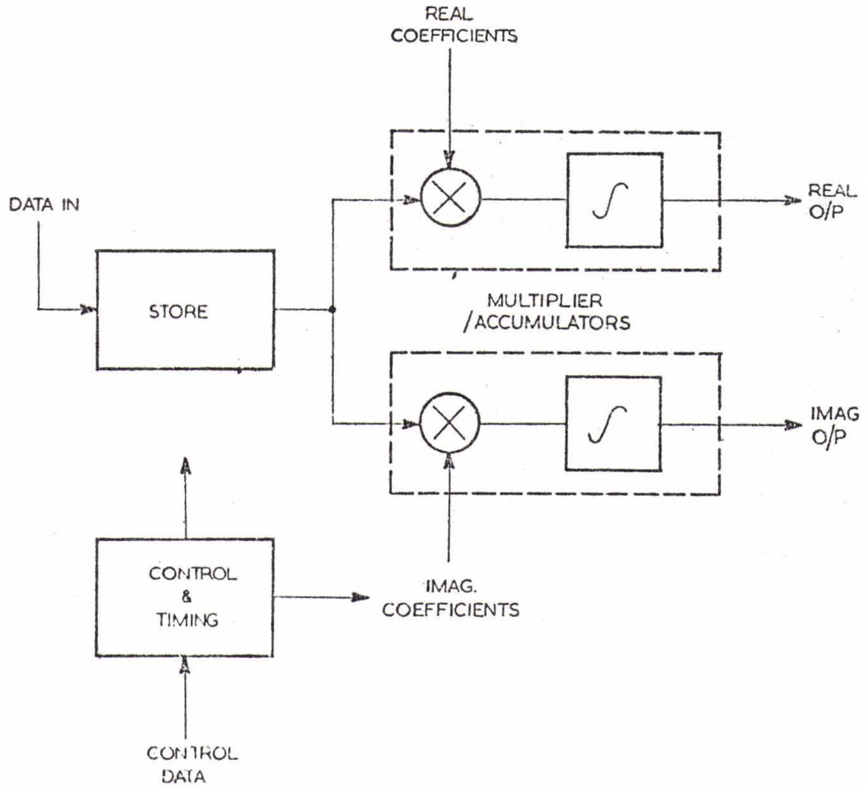


FIG. 6 PROCESSOR SCHEMATIC

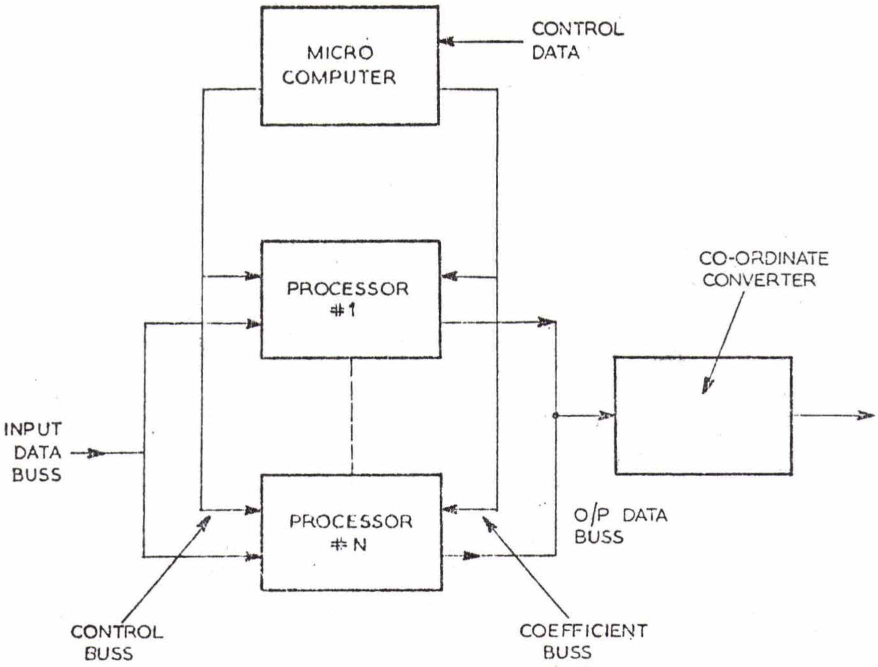


FIG. 7 SYSTEM ARCHITECTURE

DEVELOPMENT OF MILITARY ARGUS COMPUTERS AND MOD BUS FOR CLOSE-COUPLED SIGNAL PROCESSING APPLICATIONS

by

Donald Nairn
Admiralty Underwater Weapons Establishment
Portland, Dorset, UK

SUMMARY The next generation of Naval sonar systems will make extensive use of flexible programable modules to implement signal-processing algorithms. A number of UK Defence Establishments are engaged in a series of coordinated hardware and software developments which will provide basic modules for a wide range of multi-processor applications. These developments include an advanced bus for tightly-coupled computer systems, a general-purpose 2-card computer module (the Military ARGUS), and serial digital links for loosely-coupled processors, together with a new standard military equipment build practice based on the double EUROCARD. Software support includes a CORAL compiler and versions of the MASCOT modular software construction system.

1. INTRODUCTION

For over three years a number of UK Defence Establishments have cooperated in the development of a series of basic hardware and software modules, which will play a major role in the design of the next generation of Naval Systems such as sonars. Many items which have previously been developed separately for each project, are now being developed in a quite general way prior to the start of these projects.

In particular, the next generation of UK Naval Sonar Systems will have the following features:-

a. Both large and small sonar systems will be implemented using programable modules taken from a small number of different types (about 50 card-types in total).

b. They will use the same standard construction practice for all units, based on the commercial double EUROCARD made to military specifications.

c. A 2-card military computer has been developed based on the commercial ARGUS range of machines. This will be used extensively as the main system computer.

d. A new bus has been specially developed (the MODBUS) for the close-coupled operation of many computers etc.

e. Extensive use will be made of serial digital links for sub-system interconnection (eg MIL STD 1553B, or the ASWE multi-drop serial highway).

f. All software will be written where possible in CORAL 66, which is the UK standard language for military applications.

g. A methodology for software production, operation, and testing called MASCOT will be used, which enforces a modular structure on programs.

2. PROGRAMABLE MODULES IMPLEMENT NEXT-GENERATION SONARS

Design studies for the next generation of large sonar systems, are showing that it will be possible to implement them using a relatively small number of basic module-types. Whilst it would obviously be attractive to perform all the functions within general purpose computers, the technology now available leads us to design modules which can be thought of as computers specially tailored to particular application areas. A companion paper by T Curtis shows an example wherein a very wide range of beamformers can be implemented using what amounts to an array processor which produces weighted sums of space-time sample elements. Other examples of specialised computers are in signal processing, display picture-generators, etc. The main point to be made is that a single module can often be designed which can be programmed to perform any task within its area.

Current estimates suggest that perhaps 90% or more of the next generation sonars can be implemented using about 50 double EUROCARD card-types. This is some two orders of magnitude less than the number of card-types employed in current sonars of similar size.

One extremely important aspect which has yet to be demonstrated, is that we are not exchanging a hardware problem for a software dilemma. Whilst it is true that transforming a problem is not the same as solving it, there seems to be every indication that the programming and control aspects of these flexible modules are particularly simple. The general pattern seems to be one of a set-and-forget operation in which a block of control parameters are loaded into the module at relatively infrequent intervals, which correspond to a change in sonar configuration.

3. STANDARD HARDWARE CONSTRUCTION PRACTICE

A later section will describe the adoption of a 2-card version of the ARGUS computer for UK Naval applications, in both surface and submarine fleets. In formulating proposals to use the same embedded computer, it quickly became apparent that this 2-card computer could not be specified without also specifying the equipment construction practice ie one cannot divorce a card specification from its powering, cooling, interconnection, and housing specifications. Furthermore, the widespread use of these computer cards within both surface and submarine systems, implied a common equipment practice being adopted by both branches of the Royal Navy. Whilst the logistic advantages of this proposal hardly need to be stressed, it has proved difficult in the past to ensure that even one contractor uses the same equipment practice on consecutive projects - somehow they always want to begin every project by designing a new power supply!

In the event it has proved possible to set up a joint contract with three major contractors spanning the surface and underwater fields, to have them share the development of a new Military equipment practice which all three will be prepared to use. Since this development is funded by HM Government, it will be made available to other companies, and also to the commercial market, perhaps by 1982.

Many existing equipment practices such as SEMS, ATR, CAMAC, etc were examined. In the end it was decided to base the practice on the 19 inch rack standard, and on the single and double EUROCARDS (100 x 160mm and 233.4 x 160mm) which enjoy wide commercial use. The basic unit is to be one shelf, each shelf containing its own power, cooling, and interconnection arrangements. In effect multiple height "cabinets" can be constructed by stacking the required number of shelves one on the other. Both air-blown and conduction cooled versions are being developed.

4. THE 2-CARD MILITARY ARGUS COMPUTER

A decision was made to develop a 2-card military processor of equivalent power to the (then) standard Naval computer, the FM1600B. This decision to develop a computer-module of equivalent power but greatly reduced size, rather than to exploit the technology advances to produce a much more powerful computer of the same size, reflects the consensus of opinion in favour of distributing intelligence (and software) over the sub-systems rather than centralising same in a small number of powerful computers.

The intention is to use the 2-card processor for all the major computing tasks and, because of its cheapness, to use it also for many mundane hardware-replacement applications.

Furthermore, since hardware development costs are now dwarfed by those of developing software items such as compilers, editors, program preparation systems etc, it was decided to base the military processor on a successful range of commercial computers which already have good software support. With the further decision to limit the choice to a UK 16 bit machine, the Ferranti ARGUS 700 computer range was found to have an architecture best suited to the requirement.

The Military ARGUS 700M processor consists of 2 double EUROCARDS, with an optional 3rd card to provide up to 32K words of private store. It can be thought of as a self-contained processor which uses the MOD BUS (see below essentially as a communication bus - in this respect it differs from the pdp11 UNIBUS which is really the heart of the simpler versions of the pdp11 processor. When working with its private store, the ARGUS does not use the MODBUS, yet it is free to work with store on the latter, perhaps sharing it with other ARGUS processors also connected to the same MODBUS.

The ARGUS instruction set has been designed specially to support the CORAL 66 language, this being its natural mode of use rather than some lower level assembly code. It includes floating point, double-precision, and byte modes of working, with typical integer ADD of $<2\ \mu$sec, and 16 bit multiply/divide of $<10\ \mu$sec, depending on the store option. One feature of particular importance for applications employing numbers of embedded computers,is the extensive range of microcode implemented program - loading and start options available when the power is switched on. Another feature is the processor monitoring facilities also implemented by microcode. Any ARGUS processor can be stopped by plugging a small lead directly into its front-panel and interrogating it directly using a teletype or a VDU.

The ARGUS processor is constructed using 2901 four-bit slice micro-circuit chips, together with many PROM and FPLA devices. An entirely novel development technique was used wherein a hybrid simulation of the entire processor down to the gate level was run on a host computer. The entire design was verified by running the diagnostics of the target (ARGUS) machine on the simulation prior to its construction. Since the paper tapes for programming the PROM and FPLA devices used extensively in the ARGUS design are generated automatically from this simulation, in a wierd way the actual construction of the more complex parts of new machine has therefore been reduced to an interactive software task.

5. THE MODBUS

Since there exist literally dozens of buses (eg the pdp11 UNIBUS etc), it was not without careful consideration that the decision was taken to spend a considerable amount of money and effort to develop an entirely new bus - the MODBUS. It is important to grasp that the MODBUS is primarily a communication bus for processor/processor, processor/peripheral, and peripheral/peripheral applications. Thus for example it is better to adopt a minimum-wire design (28 lines) at the expense of speed, since it is extremely important that a communication bus should not "strangle" a peripheral card by using up too many of its edge-connector pins. This is particularly important in the case of bus-link cards, where two buses have to enter a single card.

Special microcircuit devices have been developed, such that a set of three 40 pin devices per card can perform the entire interfacing, handshaking, control etc to the MODBUS, and provide the user with a wide range of facilities and operating modes. A comparison with the LSI11 Q-BUS interfacing arrangements has shown this 3-device set to occupy less board area than the combination of the Q-bus drivers and the supporting address recognition logic etc, yet the MODBUS set provides many more services for the user. The BUS is fully handshaken, the aim being to achieve as far as possible assynchronous technology-independent operation. BUS integrity is enhanced by each sender chip-set also reading the states of the BUS lines, and comparing same with the states it is asserting, with error flagging as appropriate. The chip-set design is iterative so that wider buses can be constructed by adding one extra 40 pin chip per 8 lines.

The MODBUS can be thought of as "belonging" to the Arbiter, whose job it is to allocate the use of the bus among the competing processors, peripherals, etc on a cycle-by-cycle basis. Each cycle takes about one μ second, with fully overlapped allocation and use operations. Address and data words are multiplexed over the same wires to minimise the number of bus lines (28 lines).

READ AND WRITE CYCLES: Read, Write, and Read - modify - Write cycles are supported, with 256K address range and 16 bit data words. Eight-bit byte addressing is also supported.

PSEUDO ADDRESS SPACE: A Pseudo address space of 256K is provided to enable peripherals to be controlled without encroaching on the data-space. It is also used for Interrupt transactions.

INTERRUPT CYCLES: Simple peripherals can flag the Bus Arbiter to generate processor Interrupts for them. More intelligent peripherals, or processors themselves, can interrupt other processors by requesting bus use and directly sending Vector cycles over the Bus as appropriate.

VECTOR CYCLES: This is a simple cycle consisting of an address (usually in Psuedo address space) and no data ie the address is the message. To allow say 16 different interrupts to a processor, 16 different addresses in Pseudo space are allocated to the processor, with a Vector cycle to any of those addresses producing the desired interrupt. This arrangement results in rather a detailed though immensely valuable design feature to do with using the bus-interface chip sets. In essence, it allows the master and slave operations, to and from the user side of the chip-set, to pass through separate assynchronous ports. all the Master Read and Write operations are confined to the data port of the chip-set, whereas the slave interrupt Vector cycles coming assynchronously into the processor use only the address output port from the chip-set.

BUS CYCLE DE-ALLOCATE: A feature of MODBUS operation is that 2 or more MODBUSES, each operating assynchronously with its own Arbiter, can be linked or chained together. Should say a processor on Bus A generate a Read cycle which addresses a peripheral on Bus B, the A/B linker card will recognise the address and request a cycle on BUS B. Once granted, the A/B cycle will proceed with the required Read data being passed back to the processor through the linker. Similarly, a device on Bus B can address any device on Bus A. Since both buses are assynchronous, an obvious problem exists when simultaneous requests in both directions crash head-on in the A/B Bus linker. The MODBUS includes hardware facilities to resolve this problem in a manner entirely transparent to the users and their software. Clearly either the A or the B bus cycle will have to be aborted. The linker has to be preset as to which BUS has priority (say the A Bus), and it asks the B Bus Arbiter to De-allocate its current cycle and allocate the B Bus to the linker. The B BUS Arbiter signals the device on the B Bus to discontinue its cycle, and the A/B cycle proceeds. The B Bus Arbiter then re-allocates a new cycle to the suspended device and the B/A cycle proceeds. The key point to note is that all of these transactions take place within the device chip-sets, with the user-side being unaware that anything untoward is happening. The above sequence can be further generalised through 3 or more chain-linked buses.

The MODBUS has been developed by Ferranti Computer Systems Ltd under contract to HM Government. All rights to the MODBUS are owned by HM Government, who wish to encourage the widest possible use of this standard in both Military and Commercial applications. A full specification is currently available for the MODBUS[1].

6. SERIAL DIGITAL LINKS

There are two serial digital links currently being adopted as standards for use with the MODBUS. The MIL STD 1553B has the attraction of being a very well defined and widely used standard, with a number of micro-circuit manufacturers producing special interface chip-sets. Currently its drawbacks are that its terminal/system interface is not defined, and that it cannot efficiently work in a high-integrity broadcast mode. Its payload is approximately 2/3 Megabit per second.

Wells and Stainsby of the Admiralty Surface Weapons Establishment, Portsdown, UK[2], have designed a high-integrity broadcast serial link of approximately twice the payload of the 1553B link. Although its system interface is well defined, its hardware aspects are not as well established as the other, and it has not yet been widely adopted as a standard. Interfaces have been produced to a number of different computers and run as a loosely coupled network.

7. MODULAR SOFTWARE USING MASCOT

MASCOT[3] (Modular Approach to Software Construction, Operation and Testing) is a methodology which aims to cover all phases of software development for real-time applications. It enables a System Designer to describe a complex software task by drawing up a network diagram using only 3 entities - ACTIVITIES, CHANNELS and POOLS.

ACTIVITY: A separate software process which operates independently and assynchronously with all the other ACTIVITIES which make up the software system. ACTIVITIES are deemed to be single threads which run in parallel with one another. All interconnections between ACTIVITIES are through very precisely defined ports called CHANNELS or POOLS.

CHANNEL: This is a uni-directional data-transmission buffer which has separate input and output ports. Any number of ACTIVITIES can be attached to the input port for inserting messages into the CHANNEL. Usually only one ACTIVITY is attached to the output port to receive these messages, although in principle more than one ACTIVITY can be so attached. Before using a CHANNEL, an ACTIVITY must claim ownership to say its input port for its exclusive use, via a simple semaphore arrangement. Should this input port already be busy, the claiming ACTIVITY is automatically suspended and placed on a Control Queue associated with the CHANNEL. When eventually the owning ACTIVITY releases the port, the ACTIVITY at the head of its Control Queue (if any) is automatically given ownership of that CHANNEL port, and re-started from where it was suspended.

POOL: This is similar to a CHANNEL, excepting that it stores permanent or semi-permanent information whereas the CHANNEL contains only information in transit.

Once the System Designer has expressed his entire software development as a network of ACTIVITIES interconnected by the appropriate CHANNELS and POOLS, he then goes onto specify each ACTIVITY as a separate task to be written by one of his team of programmers. However, not only does he write all of the CHANNEL and POOL software himself (or extract same from a standard library), but he also supplies to each ACTIVITY - writer the actual code used to read/write messages from/to the CHANNELS and POOLS. Thus all the ACTIVITY interconnections are not only forced to be open and explicit, but the actual code to implement these interconnections is all produced by the one person - the System Designer. Furthermore, since all of the system synchronisation semaphore operations are contained within these centrally produced "Access Procedures", the writers of the individual ACTIVITIES are not concerned with such details - they merely call for input messages, and deliver their output messages using the Access Procedures given to them, and are not aware that their ACTIVITIES are periodically suspended and re-started as required by the overall system operation. In this way it is seen that the MASCOT network is "data-packet" driven rather than say clock-interrupt driven.

MASCOT provides a small supervisor or "Kernel" to manage the scheduling of the current ACTIVITIES on a time-sharing basis, and to manage the semaphore synchronisation etc.

ACKNOWLEDGEMENTS

In pursuing these coordinated developments, the main MOD personnel have been BR Gladman[1], K Jackson[2], R C Makin[3], the writer[3], R S Newton[2], and T D Wells[1]. Ferranti Computer Systems Ltd, Bracknell, UK has been a major contractor.

1. Admiralty Surface Weapons Establishment, Portsmouth, England.

2. Royal Signal and Radar Establishment, Malvern, England.

3. Admiralty Underwater Weapons Establishment, Portland, England.

REFERENCES

1. "Specification for the MODBUS", Copies may be obtained at a nominal charge from Equipment Sales Dept., Ferranti Computer Systems Ltd, Western Road, Bracknell, Berkshire RG12 1RA, England.

2. "ADNET. An Experimental Information Distribution System", Wells T D and Stainsby MG, Proceedings of AVP 36th Technical Meeting on Technology for Data Handling II, 1978.

3. "MASCOT Suppliers Association - Official Definition of MASCOT" Obtainable from K Jackson of RSRE, St Andrews Road, MALVERN, ENGLAND.

DISCUSSION

CAPT C.M. Rigsbee It is refreshing to hear your comments regarding the need for software (S/W) standards and hardware (H/W) standards. We have similar goals resulting from our experience in contractor proposals for many different forms of H/W and S/W. The need for both H/W and S/W standards is fully recognized but difficult thus far to implement. I'll have more to say tomorrow on this.

D. Nairn The U.K. uses GEM but finds them to be too expensive due to the very tight specifications. The GEM approach was designed in the U.S. to enable selection of low-bidders, but is too expensive for U.K. We favour the Euro-card approach to allow standards to be applied but competition to exist.

CAPT C.M. Rigsbee Concur — GEM has too high an "overhead" in weight and cost for many air applications.

J.W.R. Griffiths Would the author like to comment as to why, although CORAL is an accepted language for real-time systems in the U.K. and is supported by RSRE, it has not been used, as far as I am aware, outside U.K.

D. Nairn Main advantages of CORAL is that it is a widely used standard in U.K. MOD applications. It is not held to be specially good and is now rather dated. The general feeling is that CORAL 66 will continue to be adequate until the mid 80's, when the DOD ADA language seems likely to take over.

THE AP 120 B ARRAY PROCESSOR

by

Walter Wagner

KRUPP ATLAS-ELEKTRONIK
2800 Bremen, Germany

ABSTRACT The Floating Point Systems array processor AP 120 B is briefly presented. An overview of the architecture is given. The most important features and the use of the AP software are described.

THE AP 120 B ARRAY PROCESSOR For the past two years, a Floating Point Systems AP 120 B array processor has been used by the KRUPP ATLAS-ELEKTRONIK research group for processing sonar signals. The processor is connected to an HP 2100 minicomputer as a host computer. Normally, analogue data are collected on tape recorders during sea trials. These data have to be processed in the laboratory. Because there is only a small research group, every scientist is obliged to develop his own signal processing program. As a consequence the system must be easy to handle.

The acquisition of the array processor has produced the following main improvements for the signal processing facilities of our sonar research group:

1. A lot of programming time has been saved because a library consisting of more than 200 routines was delivered with the system and could be used easily by everybody.

2. Execution time of signal processing programs was reduced by a factor of about 60 to 80 depending on the program, by the use of simple FORTRAN programming techniques.

3. With some more programming effort, special processing problems could be solved in real time. A good example of this is beamforming in real time. In this way a speed-up factor of several hundred can be achieved.

A short overview of the array processor is given in the following. For detailed information please refer to the FPS manuals.

The AP 120 B is a full floating point array processor, synchronous and pipelined. Once the data have been converted to floating point numbers, all calculations are done in this format and normally no further consideration of overflow - underflow is necessary. For output purposes, conversion is possible to any integer format up to 32 bit.

Fig. 1 shows a block diagram of the processor's hardware with its various memories, address calculators and the two arithmetic units, the floating point adder and the floating point multiplier.

The system has a 6 MHz clock corresponding to 167 ns cycle time. The main data memory is available in two versions

a slow version with 333 ns cycle time

and

a fast version with 167 ns cycle time.

The following table gives an example of the calculation speed for both memories:

	fast memory	slow memory
1024 points real FFT	2.7	4.2 ms
vector multiply	0.8	1.3 us/point
vector add	0.8	1.3 us/point
vector multiply, multiply and add	1.5	2.3 us/point

The fastest vector multiplication between data of two different memories takes 500 ns/point.

The normal programmer does not need to know the structure of the processor in detail. He is not aware of the different types of memory such as the table memory, the program source memory, the main data memory and the scratch pad memories.

The only thing he should not forget is the size of the main data memory.

Fig. 2 shows an example of the mathematical library. Its documentation is self-explanatory.

DISCUSSION

R. Seynaeve Do you frequently use chaining and Assembler?

W.G. Wagner We did not buy the Vector Function chainer, I have only mentioned this feature. I sometimes use chaining myself; this is very easily accomplished by simple assembler coding. We do not use Assembler coding very often, because most of our requirements are met by the given library functions. Assembler coding is used only for special real-time applications.

D.V. Crowe Is anyone using the Floating Point Systems Fortran compiler?

W.G. Wagner No! As far as I know not in Europe, but it is in use in the U.S.

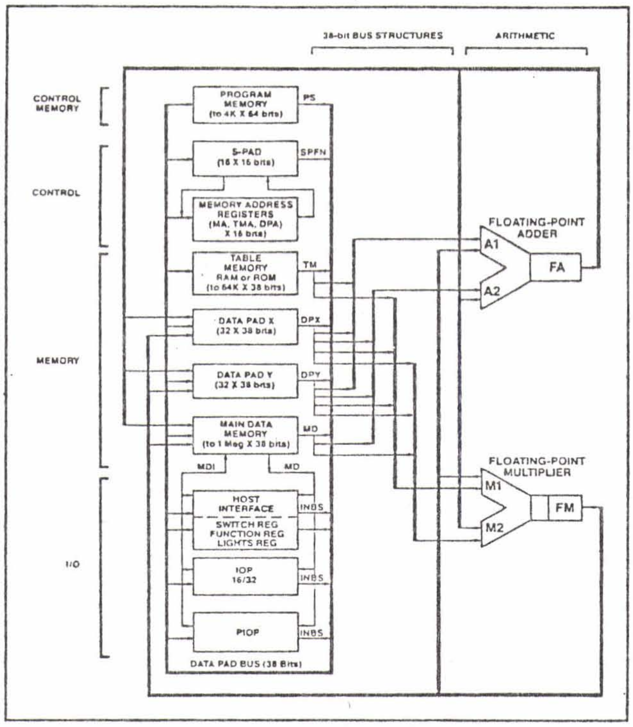
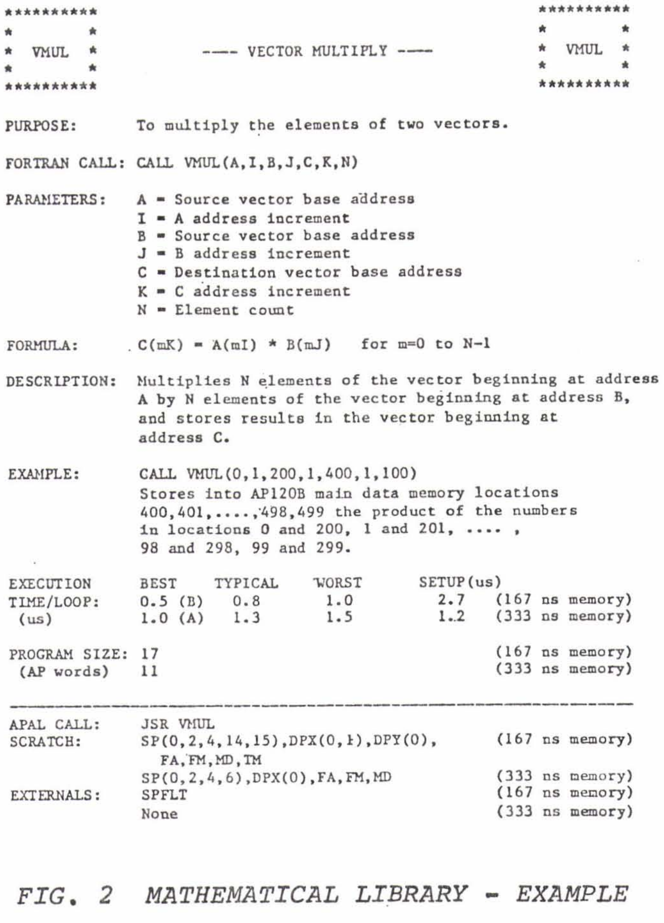

38-bit BUS STRUCTURES
ARITHMETIC
CONTROL MEMORY
CONTROL
MEMORY
I/O
PROGRAM MEMORY (to 4K X 64 bits)
PS
S-PAD (16 X 16 bits)
SPFN
MEMORY ADDRESS REGISTERS (MA, TMA, DPA) X 16 bits)
TABLE MEMORY RAM or ROM (to 64K X 38 bits)
TM
DATA PAD X (32 X 38 bits)
DPX
DATA PAD Y (32 X 38 bits)
DPY
MAIN DATA MEMORY (to 1 Meg X 38 bits)
MD
MDI
MD
HOST INTERFACE
SWITCH REG
FUNCTION REG
LIGHTS REG
INBS
IOP 16/32
INBS
PIOP
INBS
DATA PAD BUS (38 Bits)
FLOATING-POINT ADDER
A1
A2
FA
FLOATING-POINT MULTIPLIER
M1
M2
FM

FIG. 1 FUNCTIONAL BLOCK DIAGRAM



```
**********                                              **********
*        *                                              *        *
*  VMUL  *            ---- VECTOR MULTIPLY ----         *  VMUL  *
*        *                                              *        *
**********                                              **********

PURPOSE:        To multiply the elements of two vectors.

FORTRAN CALL:   CALL VMUL(A,I,B,J,C,K,N)

PARAMETERS:     A = Source vector base address
                I = A address increment
                B = Source vector base address
                J = B address increment
                C = Destination vector base address
                K = C address increment
                N = Element count

FORMULA:        C(mK) = A(mI) * B(mJ)    for m=0 to N-1

DESCRIPTION:    Multiplies N elements of the vector beginning at address
                A by N elements of the vector beginning at address B,
                and stores results in the vector beginning at
                address C.

EXAMPLE:        CALL VMUL(0,1,200,1,400,1,100)
                Stores into AP120B main data memory locations
                400,401,....,498,499 the product of the numbers
                in locations 0 and 200, 1 and 201, .... ,
                98 and 298, 99 and 299.

EXECUTION       BEST      TYPICAL    WORST      SETUP(us)
TIME/LOOP:      0.5 (B)    0.8        1.0         2.7    (167 ns memory)
 (us)           1.0 (A)    1.3        1.5         1.2    (333 ns memory)

PROGRAM SIZE:  17                                        (167 ns memory)
 (AP words)    11                                        (333 ns memory)
-----------------------------------------------------------------------
APAL CALL:      JSR VMUL
SCRATCH:        SP(0,2,4,14,15),DPX(0,1),DPY(0),         (167 ns memory)
                 FA,FM,MD,TM
                SP(0,2,4,6),DPX(0),FA,FM,MD              (333 ns memory)
EXTERNALS:      SPFLT                                    (167 ns memory)
                None                                     (333 ns memory)
```

FIG. 2 MATHEMATICAL LIBRARY - EXAMPLE

THE MSP-3 AND THE SPS-81

by

John S. Padgett *
Naval Research Laboratory
Washington, D.C. 20375 USA

ABSTRACT

The Large Aperture Acoustics Branch of the Naval Research Laboratory has developed a sea-going signal processing system in which two different types of array processors are used - a Signal Processing Systems SPS-81 and several Computer Design and Applications MSP-3 array processors. This paper briefly covers the differences in the two processors and why we have used both types.

INTRODUCTION

For some years, the Large Aperture Acoustics Branch of the Naval Research Laboratory has been involved with recording and subsequent analysis of acoustic signals at sea. In the past year, we have assembled a sea-going signal processing system which will give us the advantages of less time to publication, better results (since analog recording will cause significant data distortion), and improved capability for monitoring the experiment.

When we selected the equipment to be used, we had to pay attention to portability, since the equipment would be used on several different ships, and flexibility of architecture, since the type of data analysis often varies considerably from one experiment to the next. These two constraints, in addition to the traditional values of cost, speed, and reliability, have led us to use multiple MSP-3 processors whenever possible and supplement these with an SPS-81 when necessary. Both types of array processors are interfaced to a DEC PDP-11/34.

This paper will attempt to describe the differences between the design philosophies of the processors and show how each processor fits our requirements. In addition, some general comments will be made based on our experiences (good and bad) with the processors.

* *Presented by J.M. Griffin, Naval Research Laboratory, U.S.*

1. Characteristics of MSP-3 and SPS-81

1.1 MSP-3 (manufactured by Computer Design and Applications, Inc.)

1.1.1 General Description

The MSP-3 is a two card array processor which plugs into two adjacent peripheral slots of a PDP-11 backplane. It is relatively inexpensive (under $14000), consumes 50 watts of power, and is about one third as fast as larger array proccessors (13 msec for 1024 point complex FFT). Programming is in PROM, so loading before running is neither required nor possible. Data is stored in 2K by 48 bit (24 real, 24 imaginary) internal RAM in block floating point format, and many operations (like FFT) auto-scale the data.

1.1.2 Hardware Structure

The MSP-3 has 2K complex RAM for data storage, 2K PROM for trig constants, and 1K of program PROM, in addition to various configuration registers and one 8 bit exponent register. Since programming requires PROM burning, the internal hardware will not be covered any further.

1.1.3 Interface to Host (PDP-11)

Interface to the host computer is through four registers: control and status, opcode, argument, and unibus address. Each operation requires the host to set up at least the first two registers, so the host has to do more work to keep the MSP-3 going than a larger array processor would require.

1.1.4 Software

Programs for the MSP-3 are all in PROM, so altering the internal programming is rather difficult. Library functions can be selected when the MSP-3 is purchased, and include such standard routines as complex FFTs up to 2K.

Since even the simpler functions, such as loading internal MSP-3 registers, typically require the host to load three MSP-3 interface registers, the host has to do a bit of work just to keep the MSP-3 running. This can be alleviated greatly by using the "micro-chainer" which lets the MSP-3 read its instructions from host memory and execute them. We have found this to be a great help when the host processor is heavily loaded.

1.2 SPS-81 (manufactured by Signal Processing Systems, Inc.)

1.2.1 General Description

The SPS-81 is a large 16 bit integer array processor which is essentially a fast mini-computer. It can hold up to 256K by 32 bits of its own bulk memory, and can be interfaced directly to peripherals like disks and A/Ds via the internal bus. Once the host computer starts it, the SPS-81 can process, access its own peripherals, and access host memory and peripherals with no further host intervention. A 1024 point complex FFT takes 4.2 msec, and transform sizes up to 16K complex can be easily handled.

1.2.2 Hardware Structure

Internally the SPS-81 consists of three separate asynchronous processors: IOP (167 nsec) which controls the other two processors, accesses peripherals and host; IS (167 nsec) which sequences and controls the AS; and AS (500 nsec) which does complex integer arithmetic and can do an FFT butterfly in one instruction. The IOP consists of four identical but different priority "channels", where the highest priority channel not in a wait state will run. This provides a good way to have interrupts (for A/D, etc.) with no context switch time.

1.2.3 Interface to Host (PDP-11)

The SPS-81 has four "mode control registers" on the Unibus which enable the host to start, stop, or single step the 81. The registers and program memories are available to the host for loading or reading via a 4K "B-Bus" which appears on the host Unibus.

1.2.4 Software

The initial programming of an array function into SPS assembly language is quite difficult because of the complexity of the machine. Once this is done, future applications become easier because of an SPS software package called the "Executive Interpreter" which interprets and executes commands found in either host memory or in SPS bulk memory, depending on which version of the interpreter is used. The "Interpretive Command Language" (ICL) has 17 different commands available for modifying or reading SPS internal registers, running SPS assembly language routines, conditional GO-TO, etc. Loops and subroutines are easy to use in ICL and SPS registers can be used for counters, pointers, and data storage. This promotes flexibility and ease of programming without adding much overhead, since the ICL interpret-and-execute time is small compared to array operations like FFT.

2. Reasons for Using Each Type Processor

2.1 MSP-3

Since we cannot afford to hold up sea experiments for equipment repairs, we want to opt for the sea-going system that is most reliable, and which allows us a chance to continue partial processing in the event of equipment failure (graceful degradation). The MSP-3 is small, has low power consumption, and is approximately one-fifth the cost of a larger array processor like the SPS-81. If an MSP-3 fails at sea, we can usually continue processing if we process less signal lines or make other reductions. Furthermore, if processing requirements for a particular experiment can be met with just 2 or 3 MSP-3s, then we can take the number required plus 1 for spare and have a physically smaller system which consumes less power.

2.2 SPS-81

The SPS-81 is used when needed because of its speed, independence of host computer, and ability to work with large arrays. Multiple MSP-3s could overcome speed problems, but the host computer would have more work to do to keep them running. In addition, large arrays would be impractical to work with, since the operations would have to be done in steps.

3. Comments

3.1 MSP-3

The MSP-3s have been very reliable (except for some early firmware bugs), with only 1 failure of 5 units in about 8 months. Considering the number of times they have been pulled in and out of backplanes, and the shipping and sea duty they went through, this is perfectly acceptable. Programming is easy, but does load the host computer since the host has to initiate every function. This can be alleviated by the use of the previously mentioned micro-chainer. Program memory size (1K PROM) has been a headache to us, since we always have to compromise on which functions we want in our library.

Block floating point capability has caused a few extra programming steps. For our applications, we require time to frequency transforms for each hydrophone, then space to angle transforms for each selected frequency line. After the first FFT, all lines for one hydrophone will have the same block exponent, but all hydrophones for one line may not. Since the input to the beamformer requires all hydrophones for each line, the numbers have to be shifted to where they all have the same block exponent. This is not a problem with integer or full floating point format.

Power consumption (50 watts) of the MSP-3 is small compared to large array proccessors, but large enough to cause problems with power distribution. The PDP-11 backplanes cannot handle much more than that, so peripheral slots must be sparsely populated when MSP-3s are included.

One fringe benefit of the MSP-3 has been the assistance of the Computer Design and Applications sales representative, Andy Lukas, who has provided many excellent suggestions on using the MSP-3 for our applications.

3.2 SPS-81

Since the SPS-81 is a faster, much more complicated machine, it has been less reliable than the MSP-3, with several failures in the past 2 years. Unfortunately, many of the failures have been of a devious nature, where the machine kept running but the output began looking a little funny. For this reason, we strongly recommend either frequent running of diagnostics or running diagnostics as a background task.

In our applications, the integer nature of the 81 can cause problems. Improper scale settings can result in loss of data either through overflow (if looking at synthesized cw signals) or shifting down too far (if looking at noise).

We have had some problems in writing to the host computer with a Unibus release taking place before the write finishes. This can cause bad data to be written and/or corruption of unrelated areas of host memory. This can be avoided by doing a read after write (often unacceptable) or by waiting a "long" time before doing a Unibus release. Our complaint about this situation is that the hardware should take care of ensuring that Unibus accesses are done in a legal manner.

4. Conclusions

Whenever possible, we prefer to use many small array processors such as the MSP-3 for the advantages of lower power consumption, graceful degradation, and flexibility of architecture. However, we still have to occasionally use a larger processor such as the SPS-81 for its ability to handle large arrays and to run without host processor intervention.

DISCUSSION

H.J. Alker Are there any trace capabilities available for the MSP hardware?

J.M. Griffin The hardware has no trace capability. Microcode must be debugged using a simulator prior to burning the PROMS. The simulator does have a trace capability.

W.G. Wagner Does the SPS 81 have its own I/O interfaces?

J.M. Griffin Yes it does, and we have used it to interface to high-speed A/D systems just as other have with the MAP and FPS processors.

B. Pennoyer Have you increased the SPS 81 software library?

J.M. Griffin No, we have made only minor variations to the library supplied.

R. Seynaeve Can the I/O be hidden on the MSP?

J.M. Griffin Yes, in the sense that data can be transferred on the MSP data bus while arithmetic operations are occurring. However, if you call the MSP Library Program to load the internal memory from the host then this is the only program that can be running so it is not hidden.

R. Seynaeve What is the cost of the MSP?

J.M. Griffin The single quantity price is around $10,000.

EXPERIENCE WITH A MAP 300 ARRAY PROCESSOR

by

P. Nesfield
SACLANT ASW Research Centre
La Spezia, Italy

ABSTRACT The paper is a summary of the experience collected during almost two years of programming a CSPI MAP 300 array processor in signal processing applications. After a short overview of the most interesting features of SNAP 2, the MAP's signal processing language, the paper illustrates their value to the programmer by describing their use in the implementation of a typical signal processing algorithm. The paper also points out the most frequent sources of problems in MAP programming.

INTRODUCTION

This paper is a collection of thoughts concerning the MAP 300 and how it has performed over the last 18 months. First it must be said that any complex digital system must have the confidence of its users, a trust must be built-up such, that when some algorithm fails to perform as expected, or fails completely, the person debugging the code does not suspect everything, starting with the hardware. The problem also occurs when a small change is required, when lacking in trust too often a programmer will not dare touch the code, or breaks-out in a sweat as he imagines the side effects a seemingly harmless addition can have. One early installation of a MAP 300 in a government laboratory had a large logo saying "DANGER MAP 300's can damage your health".

In our opinion the MAP 300 has quickly accummulated that quantum of confidence that is required to pinpoint problems without recourse to a fortuneteller. The hardware is extremely reliable, during the last three sea trials we had more problems with the host CPU and disc (which is an HP and generally reckoned to be very reliable). The software, which is the user's point of contact with the device is extremely ambitious and works very well. Problems mainly concern the executive error handling, which is unfortunate since new users are apt to make simple errors that have a disastrous effect on the system. Error handling has been greatly improved with release 3.5 of the Executive.

We have never had any intermittent problems on our MAPs, every crash is repeatable and the reason entirely logical. I suspect that the government laboratory has now taken down their logo.

1. HARDWARE MAP 300

The array processor [Fig. 1(a) and 1(b)] consists of:

(a) A CSPU, which is a 16-bit fixed-point minicomputer with vector interrupt stack and block move instructions. It can address to the 16-bit (half-word level) on all three memory buses using 18-bit addresses, 512 Kbytes per bus. The CSPU has no input/output facilities as such.

(b) A floating-point arithmetic processor APU (two for a MAP 300) with an internal program memory of 128 words which is loaded by the CSPU through pseudo-memory locations. The APU can access data from its registers and from input/output FIFO's. The cycle time is 70 nsec and an operation may take up to 6 cycles for a multiply. Instructions are started sequentially, and hardware tests for resource availability freezing the APU if the current operation uses a register involved in a prior operation. In this way adds and moves can be hidden behind multiplies.

(c) An APS arithmetic processor sequencer fills and empties the read and write data FIFO's of the APU. It has a 128 x 25 bit program memory which can be loaded by the CSPU. The program of the APS must be written in conjunction with the APU to generate the input and output data sequences required for the particular algorithm. The APS thus removes the burden of main memory access from the APU which can then concentrate on arithmetic. The two processors cycle asynchronously and independently.

(d) MOS 500 and 160 nanosecond 32-bit wide memory modules addressable to the half-word level. The memory buses operate independently and so processors may access memory simultaneously on different buses. The controllers have eleven ports and a "polite priority" access scheme to arbitrate simultaneous accesses on the same bus.

(e) Input/output scrolls are programmable processors to transfer data between MAP memory and external devices. Program memory size and transfer capabilities vary between models, models 2 and 3 which we have at the Centre have a nominal transfer rate of 1.4 mHz, 16/32-bit words, with enough programming power to demultiplex, and make data-dependent decisions. Scrolls can interrupt the CSPU. The host computer is connected to a scroll model 3 called a Host Interface Scroll, which has the added ability to load its program memory from the host computer and convert to and from the different formats of the host and MAP.

The floating point format is IBM hexadecimal signed magnitude. Data is normalized when the most significant hex digit is nonzero.

2. SOFTWARE

CSPI supplies four software modules:-

Host Operating System Driver.

Host Resident Library of SNAP 2 Interfaces.

MAP Executive, Array Functions and I/O Modules.

MAP Cross Assembler and Simulator.

Using these modules the user can now (September 1979) get a MAP system running on a HP 21MX RTE-IV Host very easily and, without lavishing praise on CSPI's resident HP programmers, the documentation is superb. Six test programs are supplied, plus interface diagnostics to check the installation.

Due to the peculiar overlay structure of the HP, the assembler and the simulator cannot be loaded on an HP without some adaptation. This is promised by CSPI in the very near future. However, they may be loaded on Digital Equipment and Data General minicomputers and on any large main frame; at the Centre we use the assembler on the UNIVAC 1106.

2.1 Programming the MAP in Assembler

CSPI publish a document called "It's Simple to Program MAP" which sounds like sarcasm; in fact, the publication should be re-named "Simple MAP Programming" for, in truth, complexity of programming is proportional to the complexity of the program. All processors in the MAP are programmed in Assembler, and as assemblers or instructions sets go, the MAP is much like a PDP-11, with more instructions — 150 CSPU, 40 APU, 15 APS. The multiple programs for multiple processors neatly partition the problem.

2.2 Programming the MAP using SNAP 2

SNAP 2 has some special features:

(a) Buffers or arrays may be defined as non-contiguous in memory or any length with any spacing between samples both positive and negative. Four formats are permissible 8-bit bytes, 16-bit words, integers or 16 and 32-bit real numbers.

(b) Data may be loaded into and out of memory independant of and without interference to the arithmetic functions.

(c) Array functions have a 3 (or more) address format

$$z = f(x,y)$$

In most cases the array can be overlayed.

(d) Function Lists: Groups of functions can be grouped in lists and stored in bus 1 memory. These lists can be executed by one command from the host, and the application programmer's goal is to code all the SNAP 2 functions in lists controlling them only with data availability flags. This is a realistic goal with functions as powerful and flexible as in SNAP 2, and runs the MAP at full speed.

3. TYPICAL APPLICATION

A requirement of Mr Ph. de Heering of the Applied Sonar Group was to generate spreading functions, the operations for which are:

(1) Acquire 2K of data.

(2) Match filter (correlation process).

(3) Select data window (128 points).

(4) Hilbert transform to obtain an estimate of imaginary part of data.

(5) Repeat steps 1 to 4 for 128 arrivals.

(6) Transpose matrix.

(7) Calculate power spectrum.

(8) Display and store result [Figs 2 and 3 show typical spreading function].

The initial version worked in real-time, and with one second between acquisitions the MAP was not stretched. Now the data is read from digital tape (which is a slow 37.5 ips unit) and works at three times real-time.

The specific features of the MAP used for this application are:-

(a) Short floating point format (16 bit) adopted to enable two 16K data area in 64 Kbytes of memory.

(b) Matrix transposition by re-definition of data area; no data is moved.

(c) Independent I/O processing using two 16K data areas; area A being filled with data while area B is having spectra calculated and output.

(d) MAP processing entirely controlled by function lists; no host control required, the host manages I/O only and displays.

3.1 Causes of Problems

The simplest problem causing the most disastrous consequences to user programs written in SNAP 2 is buffer definitions. The confusion is caused by:-

(a) Overlaying two buffers such that the executive resource allocation fails to guard against multiple access.

(b) Incorrect definition of buffer type (using a complex buffer for a real data function, etc.).

(c) Signal processing algorithm design errors, difficult to correct because the final result is the only indication of the problem; the process cannot be stepped.

(d) Buffer synchronization between the host computer and the MAP, the two processors obviously have to co-operate otherwise the system will hang-up. The data traffic is the weak-link in the array processor concept.

(e) Then there is the age-old problem of speed; despite it being one of the fastest commercial array processors on the market, most tasks could use more speed.

To improve on (a), (b), (c) and (d), special software called AP-M and STROBE have been developed at SACLANTCEN.

CONCLUSIONS

The MAP 300 has been a valuable addition to the resources of SACLANTCEN. Sea trials involving the MAP as a signal processor have shown that the MAP is not just a delicate laboratory ornament but a hardy reliable tool.

DISCUSSION

J.M. Griffin Is integer arithmetic available in the MAP.

P. Nesfield Yes, with care but it is not faster and the MAP is a floating point unit.

J.M. Griffin Is there any time advantage to the 16-bit floating mode.

P. Nesfield No advantage.

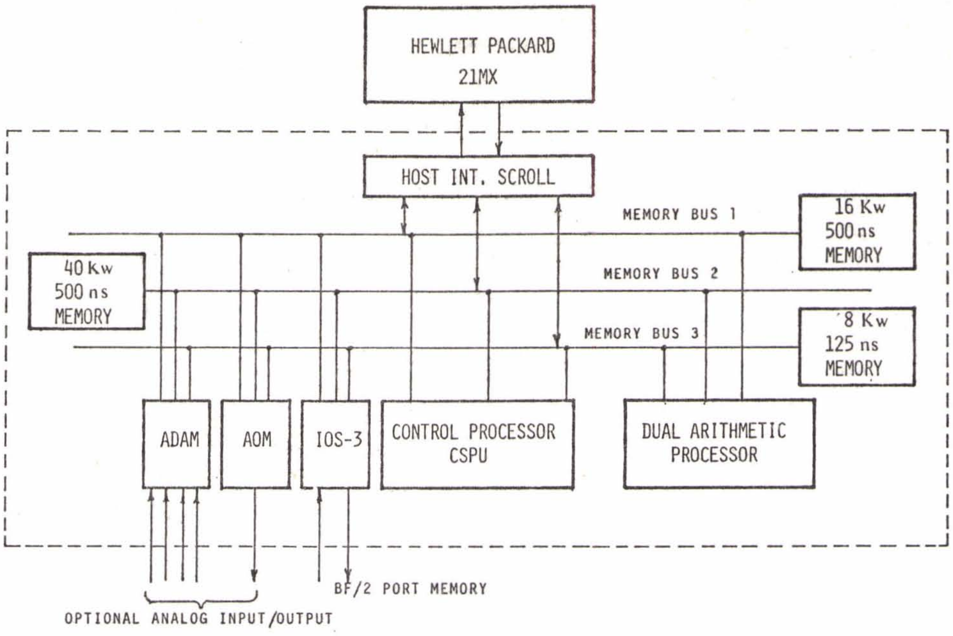


FIG. 1a

SIMPLIFIED DIAGRAM OF MAP300 ARITHMETIC UNIT

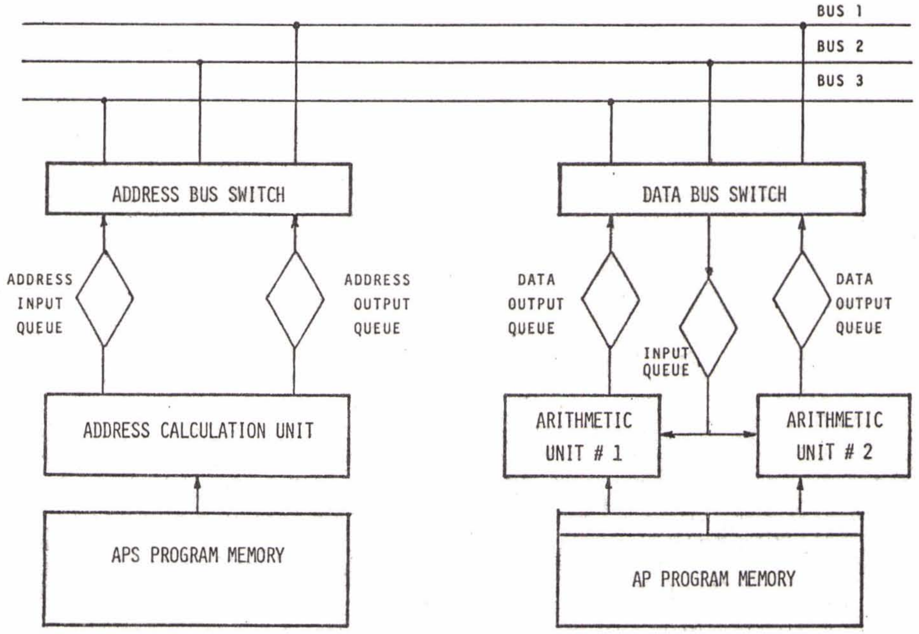


FIG. 1b

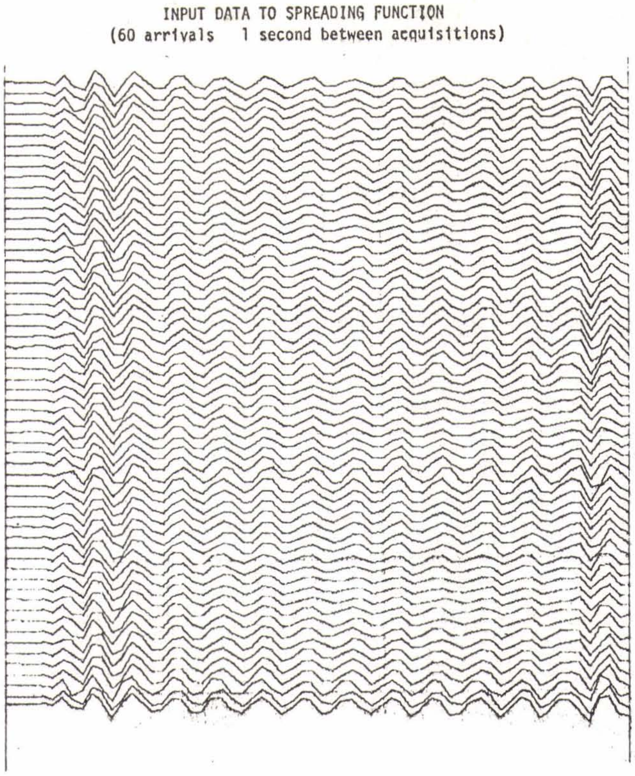

INPUT DATA TO SPREADING FUNCTION
(60 arrivals 1 second between acquisitions)

FIG. 2

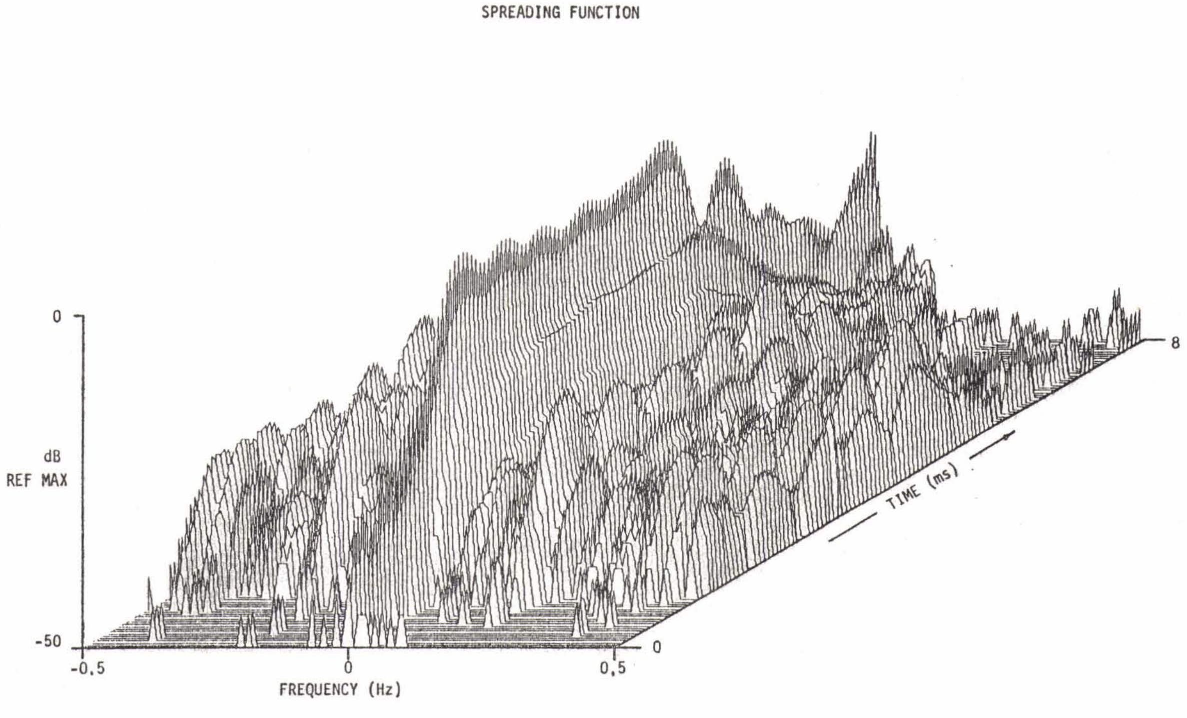

SPREADING FUNCTION
0
dB
REF MAX
-50
-0.5
0
0.5
FREQUENCY (Hz)
0
TIME (ms)
8

FIG. 3

ADVANCED DIGITAL PROCESSOR RESEARCH AT DREA

by

D.V. Crowe
Defense Research Establishment Atlantic
Dartmouth, Nova Scotia, Canada

This informal presentation deals with an array processor built by ESE Ltd of Toronto under contract to the Defense Research Establishment Atlantic, Dartmouth, Nova Scotia, Canada. The processor and software was delivered to DREA in August, 1979 for a brief evaluation period.

The hardware of the processor is configured around arithmetic units (AU) running at 8 MHz. These AU's perform complex floating point arithmetic. The AU's operate with cache memory and can be configured in groups of 2,4,6 or 8. The system that was built has 4 AU's and 128K of 40-bit main data memory. Main data memory has an address space of 8 Megawords and uses inexpensive MOS RAM memory. The entire processor is a synchronous machine and uses pipelining and memory interleaving to achieve 8 MHz throughput.

The control is divided-up among an arithmetic control unit, a data transfer control unit, an I/O control, and a supervisory controller. The instruction sets of both the ACU and TCU are designed to operate on large blocks of data with each instruction. Examples are (1) complex vector multiply with conjugation and add, and (2) an FFT butterfly pass with bit reversal addressing.

The software of the control supervisor schedules the running of the ACU and TCU according to assigned priorities of various tasks. A task might be programmed request from a networked computer, or a predefined task that is initiated by an interrupt (e.g., the arrival of new data). Interleaving of tasks is easily handled as a natural consequence of the design of the processor. Dynamic allocation of main data memory for tasks and intertask communication are possible with this operating system.

Currently software supports the standard vector and signal processing functions. In addition scalar operations and control structures for a demonstration program in the active sonar field has been completed.

The project engineer for this processor is R. Trider at DREA.

DISCUSSION

R. Seynaeve Will the processor you described be used in operational systems?

D.V. Crowe The processor will be included as the hardware part of an operational design program proposal for sonar.

OVERVIEW OF THE ADPS SIGNAL PROCESSOR

by

Bryson Pennoyer
Naval Ocean Systems Center
San Diego, Cal., U.S.A.

An informal presentation, no written text supplied (Ed.)

MAP 300/AP 120B COMPARISON

by

H.J. Alker
Electronics Research Laboratory
University of Trondheim, Norway

An informal presentation, no written text supplied (Ed.)

DISCUSSIONS

J.F. Bohme I should like to comment that I have compared the systems in a similar manner as Dr Alker. With respect to the signal processing problems I was interested in, both array processors were nearly equally powerful. If I had tasks with difficult I/O problems I would prefer the MAP 300. For algorithms with data dependent addressing, e.g., sorting for order statistics or histograms, the AP 120B is better. For the usual signal processing tasks the AP 120B is perhaps more suitable, since it is the simpler system.

H.J. Alker No comment.

PANEL ON PROCESSORS

Chairman: Y.S. Wu

Members : H.J. Alker
Y. Lundh
T.E. Curtis
D. Nairn
R. Seynaeve

Y.S. Wu In reviewing signal processor developments for the past ten years, it may be said that in 1967 we had FFT filter boxes from Bell Labs and IBM, and the standalone system Data/Time 100. Now, in 1979, the hardware in existence includes, among others:

IBM AN/UYS-1
SPERRY SP 900
HUGHES PSP
SPS 81
CSPI MAP
FLOATING POINT

With problem areas in bulk memory, power limitations, software and gross granularity.

In the future, alternatives will include:

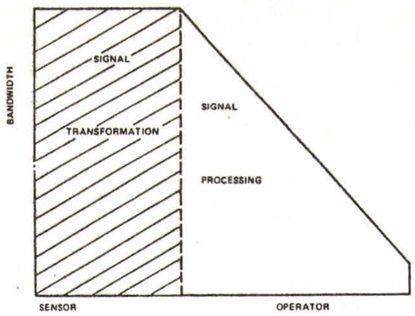
(a) Arraying of distributed signal architecture by channelizing and algorithm decomposition. Problem areas: over-used terminology; lack of definition; software nightmare.

(b) Arraying of analogue sampled data by the implementation of surface wave devices and CCD. Problem areas: limited block size; limited precision; lack of the production base.

Future trends: Deployed hardware physically different from present R & D systems; advances in system generation and reconfiguration software; spatial and function partitioning with special-purpose hardware function libraries.

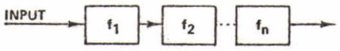
The following pages of Vu-graphs were given in support of the Chairman's introduction (Ed.)

SIGNAL TRANSFORMATION VS PROCESSING

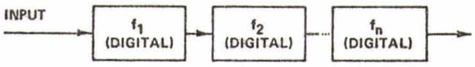


EVOLUTION IN PROGRAMMABILITY

(1) ANALOG

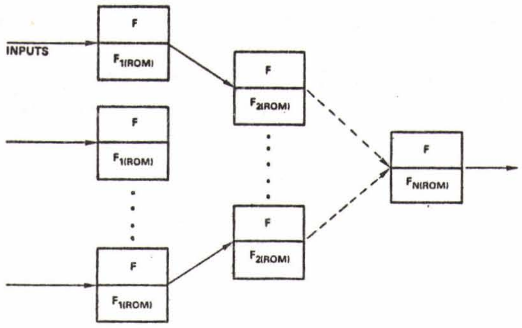


(2) DIRECT DIGITAL MECHANIZATION



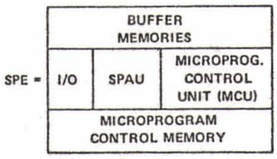
EVOLUTION IN PROGRAMMABILITY

(3) A. ROM CONTROL-DIGITAL

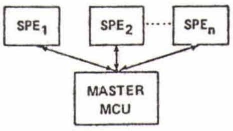


EVOLUTION IN PROGRAMMABILITY

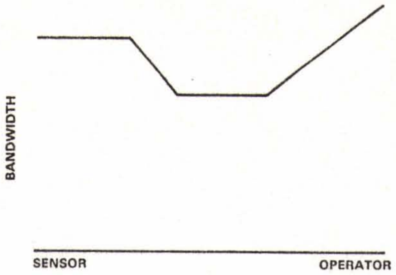
(4) SHARED RESOURCES & CENTRALIZED CONTROL



(5) SPE ARRAY

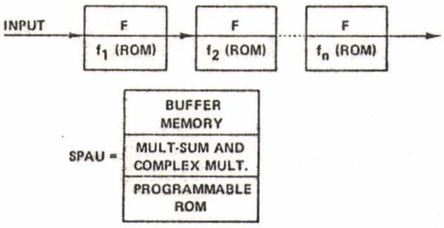


A VERY INTERESTING SIGNAL "PROCESSING" SYSTEM



EVOLUTION IN PROGRAMMABILITY

(3) ROM CONTROL – DIGITAL

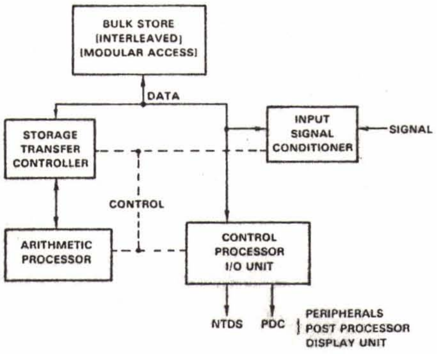


VINTAGE 1967 FFT-FILTER BOXES

COMPUTER ATTACHMENTS
BELL LAB
IBM 2938

STAND-ALONE
TIME-DATA 100

AN/UYS-1 AU ARCHITECTURE



PROBLEM AREAS

BULK MEMORY
POWER LIMITATIONS
PROCESSOR ARRAYS
SOFTWARE
GROSS GRANULARITY

BULK MEMORY

- DETERMINING SYSTEM COST ITEM
- INTEGRATED vs STAND ALONE
- DEDICATED vs MULTI-PORT
- MIL ENVIRONMENT
- MULTI-SOURCE PRODUCTION
- NEW TECHNOLOGY EVOLUTION

1979 INDUSTRY STATUS

HARDWARE IN EXISTENCE

- IBM AN/UYS-1
- SPERRY SP 900
- HUGHES PSP
- SPS SPS 81
- CSPI MAP
- FLOATING POINT AND OTHERS

FUTURE ALTERNATIVES

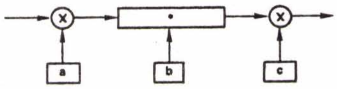
ARRAYING OF DISTRIBUTED DIGITAL ARCHITECTURE

- CHANNELIZING
- ALGORITHM DECOMPOSITION

ARRAYING OF "ANALOG" SAMPLED DATA IMPLEMENTATIONS

- SURFACE WAVE DEVICES
- CCD

LINEAR TRANSFORM PROGRAMMABLE SPAU MODULE



FOR LINEAR TRANSFORMS
a,b,c ARE LOCAL STORES

FOR MATCH FILTERS AND CROSS CORRELATION
$a = 1 \quad b = |b_j| \quad c = 1$
$0 < j < N - 1$

FOR THE DFT VIA THE CHIRP-Z ALGORITHM

$a = e^{-i\pi n^2/N} \quad b = e^{\ i\pi n^2/N} \quad c = e^{-i\pi n^2/N}$

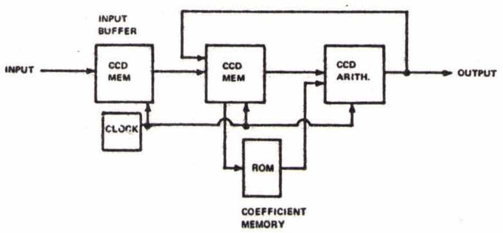
$0 \leq n \leq N - 1 \quad 0 < n \leq 2N - 2 \quad 0 \leq n < N - 1$

• DENOTES CONVOLUTION

PROBLEM AREAS

- LIMITED BLOCK SIZE
- LIMITED PRECISION
- PRODUCTION BASE

256 POINT FFT USING THREE CCD CHIPS
PLUS A READ ONLY MEMORY (ROM) AND CLOCK



COMPUTATION TIME
900 μSEC

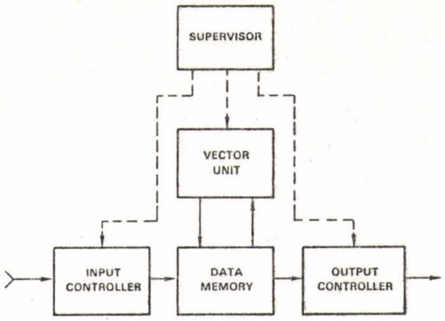
CCD HOLDS PROMISE FOR DIGITAL IMPLEMENTATION

SPAU WITH 512 WORD MEMORY

• EFFECTIVE THRU-PUT	4 MILLION MULT./SEC
• POWER DISSIPATION	< 1 W
• SIZE	4 × 350 MIL CHIPS

BULK MEMORY

MVP BLOCK DIAGRAM



SUPERVISOR

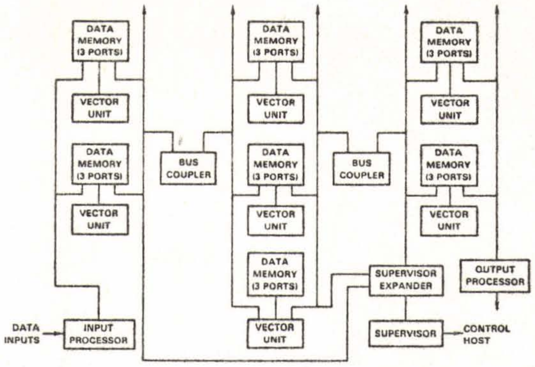
VECTOR UNIT

INPUT CONTROLLER

DATA MEMORY

OUTPUT CONTROLLER

MVP PIPELINE-PARALLEL PROCESSING SYSTEM



DATA MEMORY (3 PORTS)

VECTOR UNIT

BUS COUPLER

SUPERVISOR EXPANDER

OUTPUT PROCESSOR

DATA INPUTS

INPUT PROCESSOR

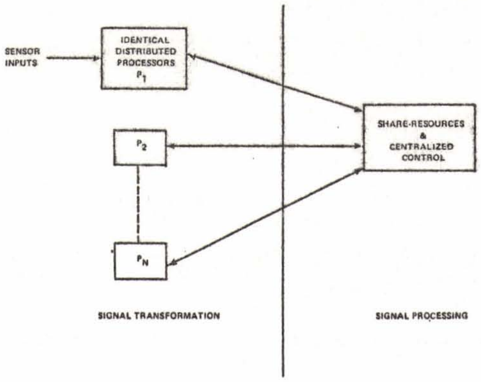
SUPERVISOR

CONTROL HOST

PROBLEMS WITH DISTRIBUTED PROCESSING

- OVER USED TERMINOLOGY
- LACK OF DEFINITION
- SOFTWARE NIGHTMARE

DISTRIBUTED ARCHITECTURE



SENSOR INPUTS

IDENTICAL DISTRIBUTED PROCESSORS P_1

P_2

P_N

SHARE-RESOURCES & CENTRALIZED CONTROL

SIGNAL TRANSFORMATION

SIGNAL PROCESSING

FUTURE TRENDS

- DEPLOYED HARDWARE PHYSICALLY DIFFERENT FROM R&D SYSTEMS
- ADVANCES IN SYSTEM GENERATION AND RECONFIGURATION SOFTWARE
- SPATIAL PARTITION AND FUNCTIONAL PARTITION
- SPECIAL PURPOSE HARDWARE FUNCTION LIBRARY

D. Nairn The implied assumptions behind the use of many current array processors:

- Essentially viewed as a hardware-assist to the software.

- Whole is viewed as a flexible general-purpose computer - configured by software.

- FFT algorithm is KING.

Alternative approaches:-

(a) Be prepared to use rather more specialized array processors, which you might even have to configure mechanically for a given problem (plug together the modules).

(b) Use software more to specify and control the transfer function in a "set the switches" mode - i.e., go for simple parameter-block setting software, rather than treating the whole problem as essentially a software task to be described in a high-level language.

T.E. Curtis We do not want to solve all the world's problems, only a number of operational problems. A complex FFT module in 6 chips takes 6.4 msec and if it is too slow, increase throughput by using, say, 32 of them in parallel. The software is simple - load the parameters and go. My bricks can solve all of my sonar requirements.

R. Seynaeve How flexible do we need to be? In operational systems, perhaps not very much.

Y.S. Wu That is the difference between research and operational.

D. Nairn Is flexibility always a good thing, given the expense of software complexity and processing time?

Y.S. Wu Flexibility is adequate for research at present. We would use flexible systems to generate more specialized deployed systems. Array processors tend to suffer from the *San Gimignano* syndrome* - each new system must be faster and more powerful than its predecessor.

* *San Gimignano is a small Italian town noted for having a large number of towers built by the leading families in attempts to outdo each other.*

H.J. Alker It is interesting to observe what has been happening in the area of the hardware implementation of systems. Before 1975, discrete logic was dominating the market in the field of high-speed, real-time system implementation. Then multi-processor schemes came about, making it easy to design systems, and moving problems to the software/microcode implementation. What has happened now is that more compact signal processing elements in hardware are available and hardware modularity is coming back again.

D. Nairn The pendulum is swinging from software back to hardware. Is it bad to plug boxes together?

R. Seynaeve There is a difference between research and operational environments. Array processors have been good for underwater acoustic research, and I expect array processors to become better integrated with computers.

CAPT C.M. Rigsbee Take care! AN/UYK1 showed that flexibility can cost money, also flexibility causes the user to expect more. Costs are not considered, because software promises much, but costs a fortune. R & D departments tend, often unwittingly, to exaggerate the potential of newly developed systems. This can make the transition from an R & D to an operational environment very difficult. I am impressed by the U.K. approach. We need standardization in computers and software.

Y. Lundh Looking at advances in electronics over the last 15 years, increases in capacity of digital systems are almost solely due to increased complexity and compactness allowing more powerful organization, rather than improvements in the speed of circuits. There are good reasons to expect further improvements in the same direction, and that this will be the main avenue to follow in the search for solutions to even more demanding problems and requirements.

SESSION III DISPLAYS

GRAPHICS FOR REAL-TIME SIGNAL PROCESSING SYSTEMS

by

M.J. McCann
SACLANT ASW Research Centre
La Spezia, Italy

ABSTRACT A graphics software package able to produce hard-copy plots of multi-channel (of the order of 100 channels) data for real-time applications is described. The system is implemented at present using a lineprinter with graphics capability, and is able to plot multiple curves along or across the paper with full annotation. In addition, intensity modulated plots may be made for use in applications such as satellite imaging. Features of the system are multi-channel capability, small core requirements and lack of restriction on plot length. Also described is a control technique for graphics programs whereby plots may be easily modified and plotting mode changed, allowing a high degree of flexibility as is required during on-line experimental work at sea.

INTRODUCTION

This paper deals with the WARP display subsystem developed at SACLANTCEN for use primarily with the lineprinters used by the Real-Time Systems Department. The principles involved, however, are applicable to many "raster-type" plotting devices; that is, devices which generate a plot from a series of dots drawn on parallel lines. The system used to control the plot output is also described.

The system comprises a subroutine package, which is of general utility outside the system as well as within it, and a control program. The subroutine package is described first.

1. THE PLOTTING SUBROUTINE PACKAGE

The package was developed largely in response to the immediate hard-copy plot requirements of a number of projects in underwater acoustics and oceanography. A list of requirements was drawn up as follows:

a. A multi-channel plotting facility able to handle up to 100 data channels plotted side-by-side in real time, or as close to it as possible.

b. An option to suppress hidden lines were curves overlapped.

c. An ability to plot large numbers (several hundred) of signals from successive events, with the option of plotting "filled wiggles" — that is, curves with the area beneath them filled in.

d. A variable density plot able to reproduce photographic images (particularly those from satellites) besides data plots.

e. A fast contouring capability.

f. Full annotation — axes, grids, text and so on.

These were the firm requirements — in addition it was felt that there should be a facility for plots of two independant variables although almost all the data we handle have only one monotonic independant variable such as time or frequency. This, in fact, influenced the final form of the plotting package very greatly.

We examined all the existing graphics packages and systems we could find, but none could meet our requirements. It was decided that a system would have to be produced in-house at SACLANTCEN, which would ultimately be able to be run as a stand-alone interactive graphics facility or added to other systems to provide a flexible graphics output requiring a minimum of programming effort to use.

The hardware in use at the time already included three Printronix lineprinters. These machines have a graphics mode — that is, they may be programmed to print single dots sufficiently close together to form a continuous line horizontally, vertically or diagonally. They are relatively inexpensive (about $5,000), use cheap paper, and have proved reliable in use. As one of them is normally included in the hardware of a system as a lineprinter, we decided to develop the package using these machines.

The basic problem in plotting with this type of device is that a line across the paper is only printed once — it is not possible to return and add additional information. Thus, each line must be absolutely complete, with dots for curves, axes and all text, before being sent to the lineprinter.

The most straight forward way to produce a plot, therefore, is to generate the entire plot as a bit map in an area of core (or some outside storage device), which is then scanned and sent to the printer line-by-line [Fig. 1]. Indeed, for plots with two independant variables this is often the only practical method. The problem is that either a large amount of storage is required or else the plot must be divided into fixed-size pages, presenting a variety of problems.

The alternative to the core image approach is to generate successive single lines of output which are sent to the printer as soon as each is complete. This technique is only suitable for plots in which one variable is monotonic and preferably defined at equally spaced intervals — conditions which apply to at least 90% of experimental data at SACLANTCEN. The advantages are that the plot package as such requires very little storage and there are no inherent limitations on plot length or problems with paging [Fig. 2]. It was decided to make this the basis of the plot package.

2. IMPLEMENTATION

A plotting program using this package normally operates as a loop [Fig. 3], using the number of the printer line currently being assembled as the controlling variable, line 1 being the start of the plot. For each line, the line buffer is first cleared, next a bit is set at the intersection points of each data curve and the current line, then bits for the axes and any characters are set in addition, and finally the line is output. Each routine called has as parameters the current line number and variables specifying at which line its output is to start and stop, together with the parameters peculiar to itself.

The system has four modes of operation. The first is a core image system similar to that shown in Fig. 1. This is primarily used for the output of blocks of text or numbers — in fact, at the time of writing it has been used for nothing else, except test patterns. The core image output routine has been written to allow rotation and enlargement of the core image on output — thus, only a relatively simple routine is required to generate an image in store of alphanumeric output blocks of minimum size (5 x 7 bits per character) and one orientation. The core image area may be of any size, and a routine has been written to generate a vector between any two points on the image. The image output routine is compatible with the rest of the system so that text may be added to any plot.

Secondly, curves may be plotted across the paper, one beneath the other. Each line number is interpreted as a data value relative to the current origin, and the data series scanned to determine whether this level has been crossed by the curve; if so a bit is set in the line buffer. For filled wiggle displays [Fig. 4] bits are set as long as the data level is greater than or equal to the line level. This mode is particularly suited to this type of display — for curves which overlap a "cylindrical buffer" of as many complete data curves as can overlap must be maintained, each curve being plotted on each line [Fig. 5]. Hidden lines may be suppressed by, in effect, a "reverse filled wiggle" clearing bits as long as the curve being plotted is above the current line level.

Note that in order to produce a continuous line it is necessary to set bits as long as the curve is in the range $V \pm I/2$ where

V = current line value

I = Value interval between successive lines [see Fig. 6]

The limitations of this mode are:

a. The number of points plotted on one curve is limited by the line length — 792 for the Printronix, though normally limited to 700 to allow room for annotation.

b. The degree of overlap of curves allowable is limited by the number of data sets which may be held in store at one time.

c. Large-amplitude plots are slow to produce as more lines are required for each curve.

The third plotting mode uses the printer to plot along the paper in a manner similar to a pen or galvanometer recorder. Here, the line number is proportional to the independant variable, and the data value determines the position of the dot along the line. The origin of each curve is normally offset by a constant amount, but the offset may be zero or varied to give grouping of curves if required.

This type of plot is very suitable for multiplexed data, as each line requires one sample from each curve. This is a problem for non-multiplexed data, as the curves must be in effect, re-multiplexed before display. In the Hewlett Packard system this may be conveniently achieved by reading all the curves into a two-dimensional array in core, using the extended memory access (EMA) facility, and reading the data out line by line. EMA is normally required as an array typically occupies around 50K words in our applications. Filled wiggles and hidden lines, as well as simple curves, may be output in this mode; hidden lines are particularly simple to implement by plotting only points to the right of the last plotted point on each line. To obtain continuous curves in this mode it is necessary to retain the positions of the last plotted points for each curve on the previous line, and to print all points between these positions and those for the current line [see Figs. 7 and 8].

Finally there is a grey scale plot mode. The effect is achieved by using pixels of 3 x 3 dots with from 0 to 9 dots printed. This gives 10 grey levels and a plot width of up to 264 points. When designing the pixels care was taken to ensure that areas of similar pixels showed distinctive dot patterns when viewed close-up as an aid to quantitative interpretation of the plot [Figs. 9 and 10]. This plot mode is the only hard copy grey scale output available to us except for photographs of CRT monitors. One problem is that the plot is distorted because the pixels are not square, due to the difference in horizontal and vertical dot spacing (72 dots per inch vs 60 dots per inch). This has not, so far, proved a serious embarassment; if it were, a 6 x 5 pixel would be square and give more grey shades but less resolution. The grey scale plot also gives a useful contouring effect in some applications, and has the advantage of providing a qualitative insight into the shape of the surface as well as a quantitative contouring [Fig. 11].

Annotation of plots with X and Y axes, grids and blocks of text is possible. Axes are plotted as, in effect, special curves, being generated line by line. Text and numerical output, however, is written into buffers as bit patterns, and output as though each pattern were an X-Y plot, as already described. The axis generating routine may also be used to generate X-Y grids.

There are a number of further routines for special displays. For instance there is a routine which will output a core image stored as a series of dot positions, which is more economical on storage than a bit map for "sparse" images such as coastlines. Another routine will plot an annotated grey scale in standard format. A third will plot lines between corresponding points on two separate curves, giving a band of varying width.

3. CONTROL

The control system is similar to that used for the rest of the WARP I system. The information required to specify a plot is held in ASCII format, stored as a file in the standard HP file manager system. This type of file may be read and edited by the operator, using the normal file manager and editing facilities, and also read by the program line by line.

A typical display file listing is shown in Fig. 12. Each line starts with a two-letter code, which is followed by the parameter or parameters to be read. Anything following the parameters is ignored by the program; thus this area is used for descriptive comments. A special code, two asterisks, is used if a whole line of comment is to be included.

When operating, the program runs in a loop which reads each successive line, identifies the parameter, reads it and continues to the next line. The loop is terminated by the /E code at the end of the file. Thus the parameters may be in any order, or repeated, wihout causing any problems.

It was found very useful to write a special editing program for the plot files, so that the operator could, for instance, set all the channel maxima to a given value at once, rather than having to edit line-by-line. This program enables a file to be generated or updated by simply typing the parameter code followed by the value — the comments are added automatically when the file is re-written.

The plot program always read the same file when starting a plot. The file manager is used to copy the relevant file into this standard file before the program is run; in fact, a so-called "transfer file" is used to load all the control files for the WARP I system and run the first program. Files may thus be written and stored in advance of the program run, with descriptive names to assist the operator.

This control system has been found very convenient in use — the average time taken to modify a plot file during a recent trial was around ten minutes. It was also found useful to generate "blank" file printouts — that is, printouts in the standard format, but without the numbers, which could be filled in and then quickly copied by the operator using the special editor already described.

4. FUTURE DEVELOPMENTS

As may be seen in Fig. 12, the plot control file is currently used to specify which data are to be plotted (via the number of channels displayed (ND) and the identifying column (CN) in the channel specification lines). This function will be separated to increase the generality of the plotting program.

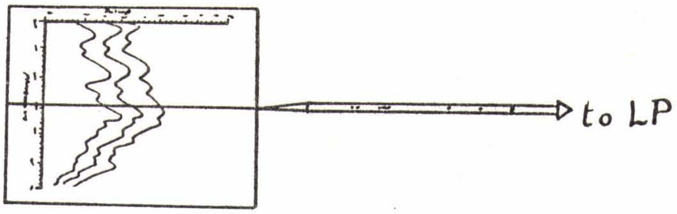
We have found that the lineprinters give good quality output on inexpensive paper, and these will become our standard plotting device. Work will be done on increasing the efficiency of the subroutine package by using assembler and microcoding parts of it — most of the work carried out by the routines consists of logical operations such as bit packing and unpacking, "AND" functions to add bits to lines, shifts and so on which are not handled efficiently in FORTRAN but are easily implemented in assembler and microcode.

The system will shortly be extended to make use of the Lexidata colour display and Applicon colour plotter. We may also add an electrostatic printer/plotter to give better resolution and speed.

CONCLUSIONS

This plot package for multichannel applications is working satisfactorily and provides an efficient and flexible plot facility. The writing did not require a great deal of resources — about two man-months. The principle of control of systems via ASCII files has proved very successful and will be further refined and extended during the coming year.

"Core Image" System

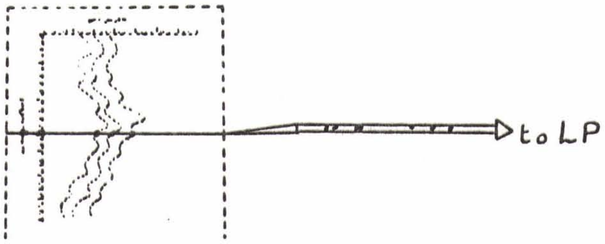


The image or image page is completely specified in core as a bit map, then scanned and output line by line.

- Only practical method for plots of two independent variables
- Require large core area, or a complex paging system

FIG. 1 CORE IMAGE SYSTEM

Alternative System (as used in WARP)



Only one line of the plot at a time is imaged in core, and output when complete.

- Suitable for plots with one independent monotonic variable.
- No paging of plot or limitation of plot length.
- Minimises core requirements.

FIG. 2 ALTERNATIVE SYSTEM (AS USED IN WARP)

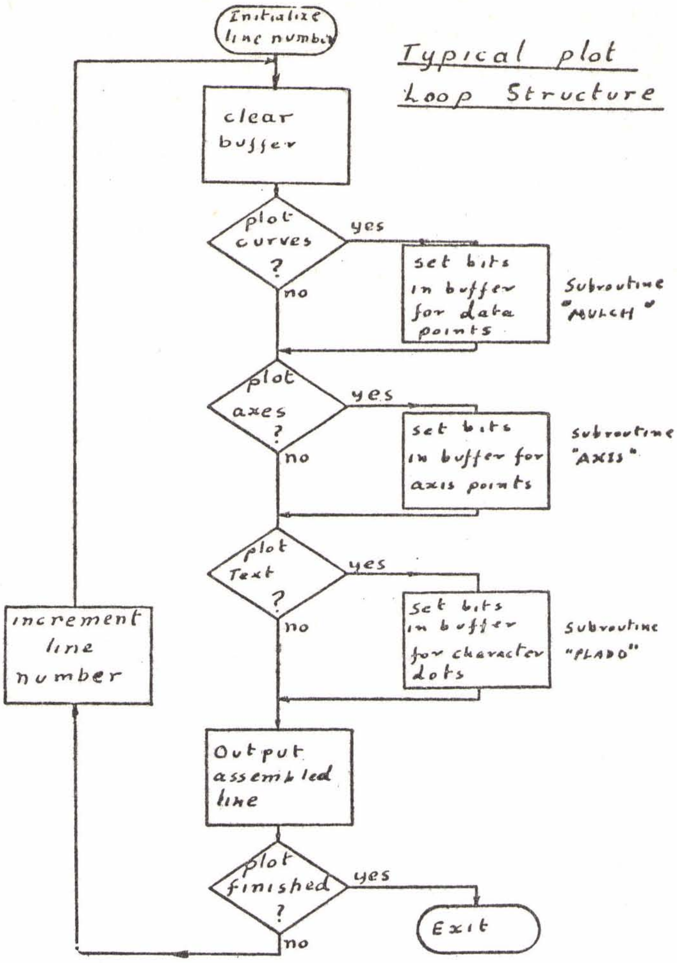


FIG. 3

TYPICAL PLOT LOOP STRUCTURE

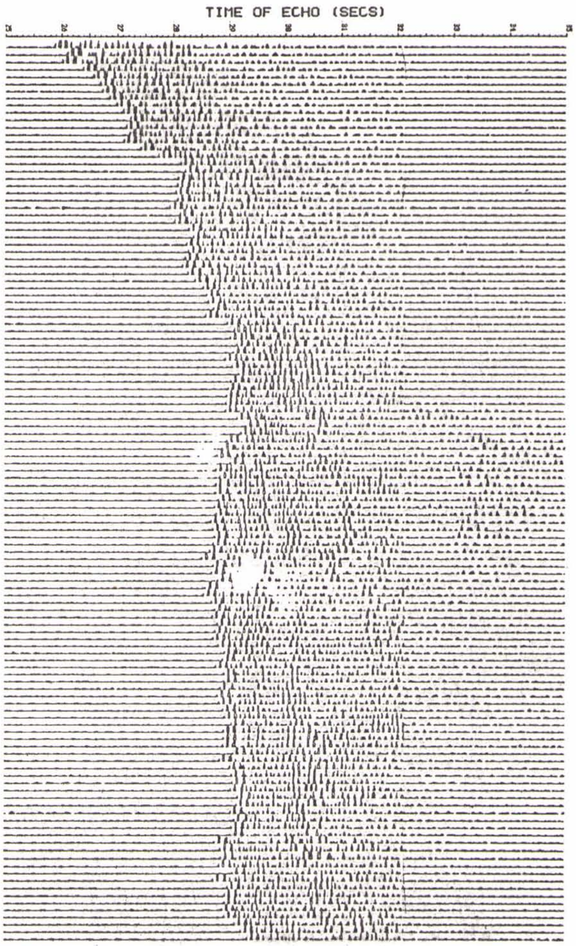
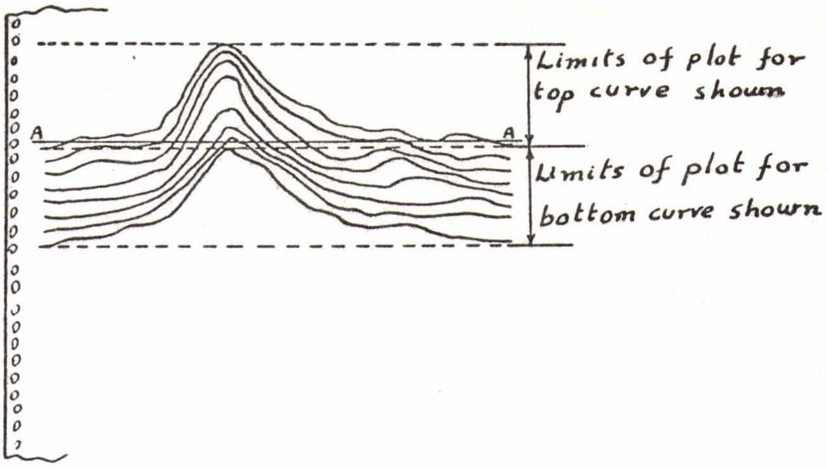


FIG. 4

TYPICAL "FILLED WIGGLE" DISPLAY FROM SUB-BOTTOM PROFILER


Limits of plot for top curve shown
A
A
Limits of plot for bottom curve shown

Here, the curve separation = $\frac{1}{7}$ of the plot height. Thus, line A-A of the plot can interset 7 data curves, requiring seven curves to be in store and examined for each line plotted.

FIG. 5 PRINCIPLE OF ROTATING BUFFER STORAGE WHEN PLOTTING OVERLAPPING CURVES ACROSS PAPER

Each line corresponds to a data value:

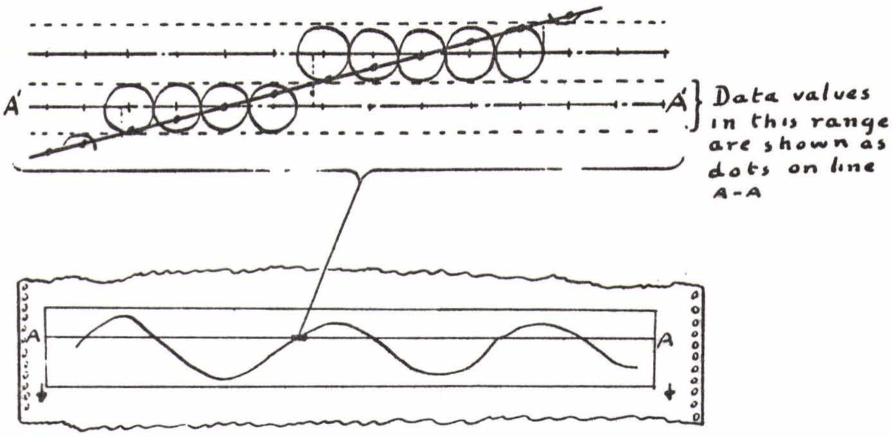
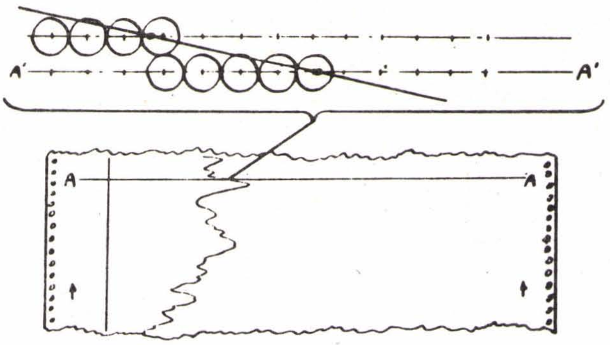


FIG. 6 CURVE PLOTTING ACROSS PAPER



Here, each line corresponds to a data point, and the dot position along the line to the plotted value.

FIG. 7 CURVE PLOTTING ALONG PAPER

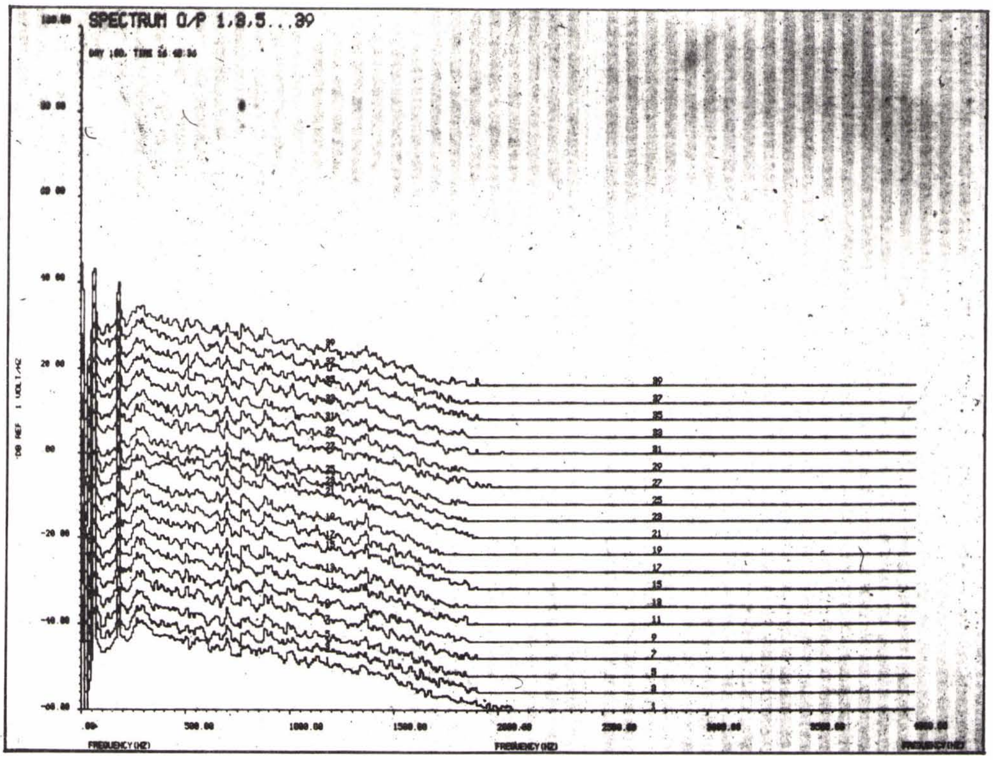


FIG. 8 TYPICAL PLOT OF SPECTRA, PLOTTING ALONG PAPER

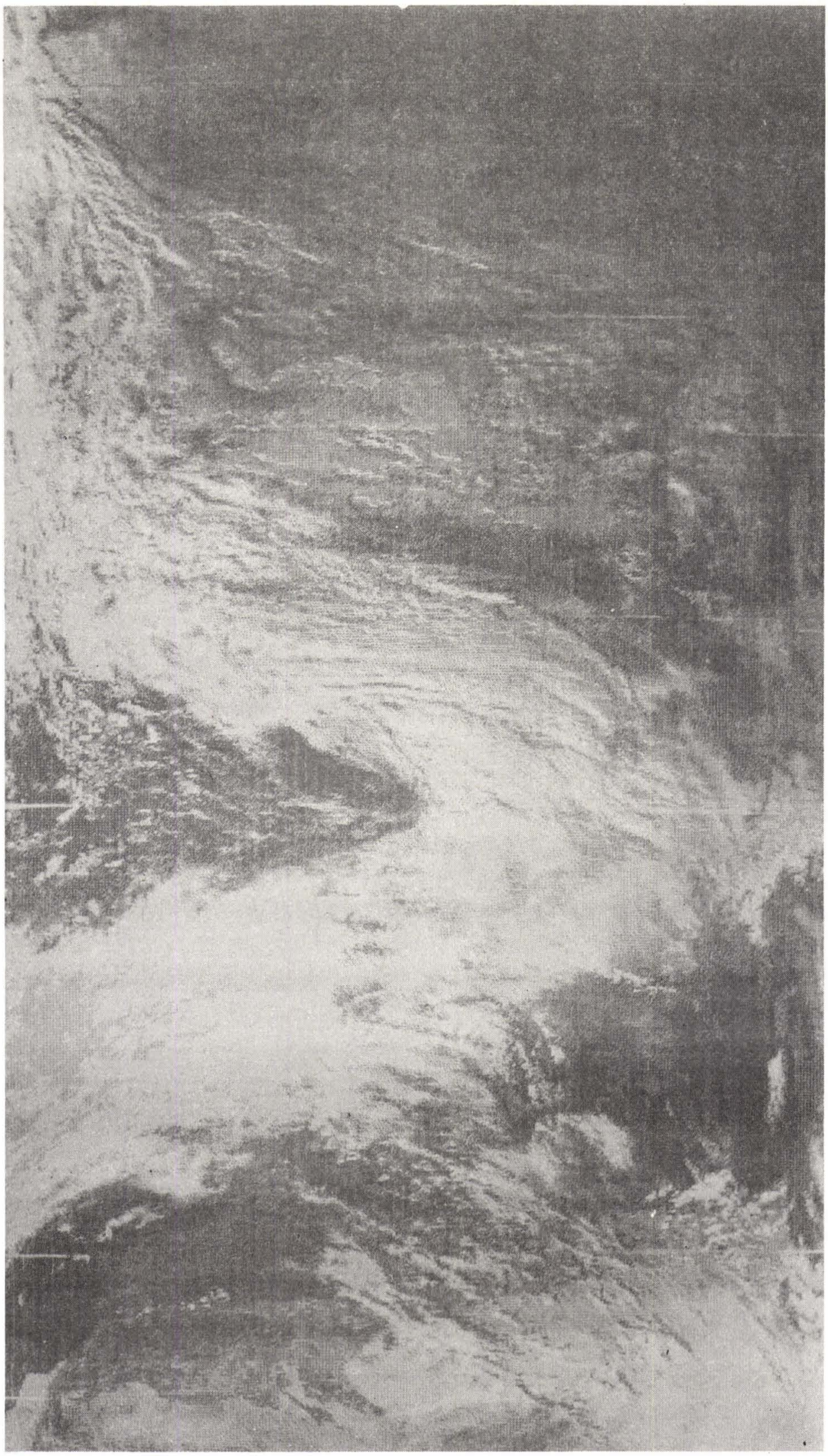


FIG. 9 TIROS-N VISIBLE CHANNEL - N. AFRICA, EUROPE AND E. ATLANTIC OBSCURED BY CLOUD.

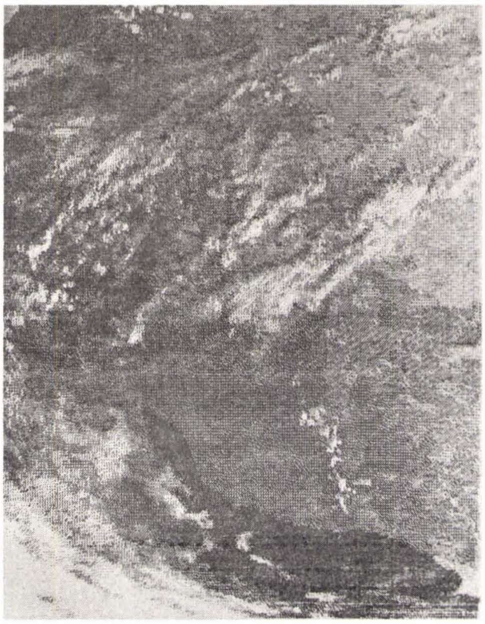


FIG. 10 PORTION OF TIROS-N PHOTOGRAPH DISPLAY (ACTUAL SIZE) SHOWING CLOUDS OVER N.W. AFRICA. ONLY 1 IN 8 POINTS ARE PLOTTED IN THIS CASE

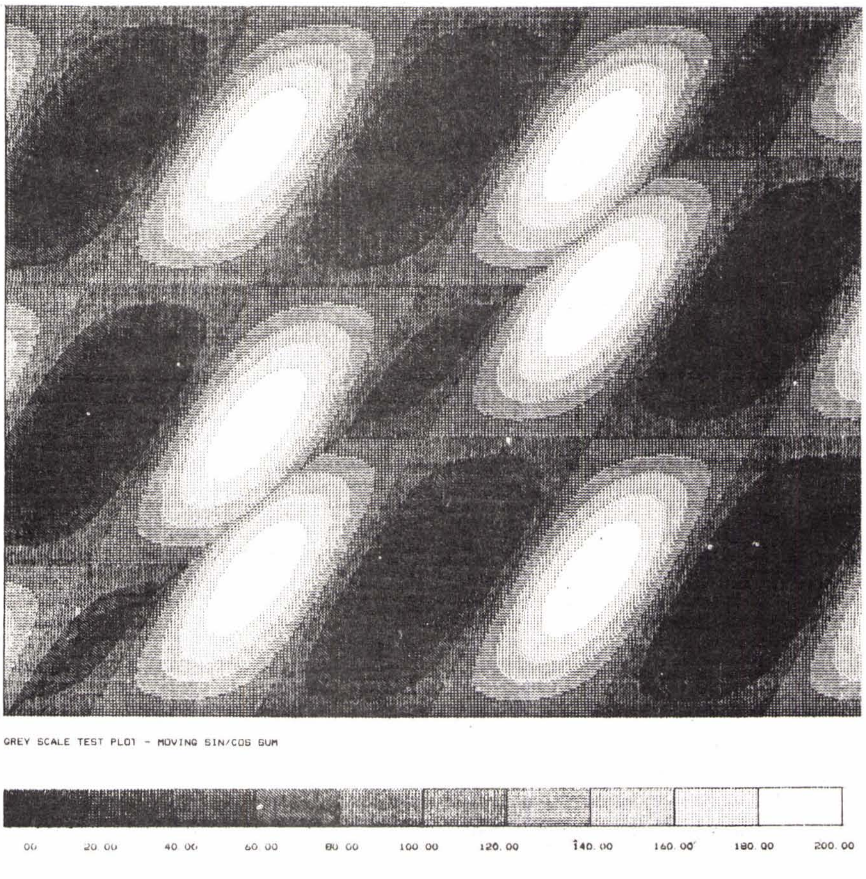
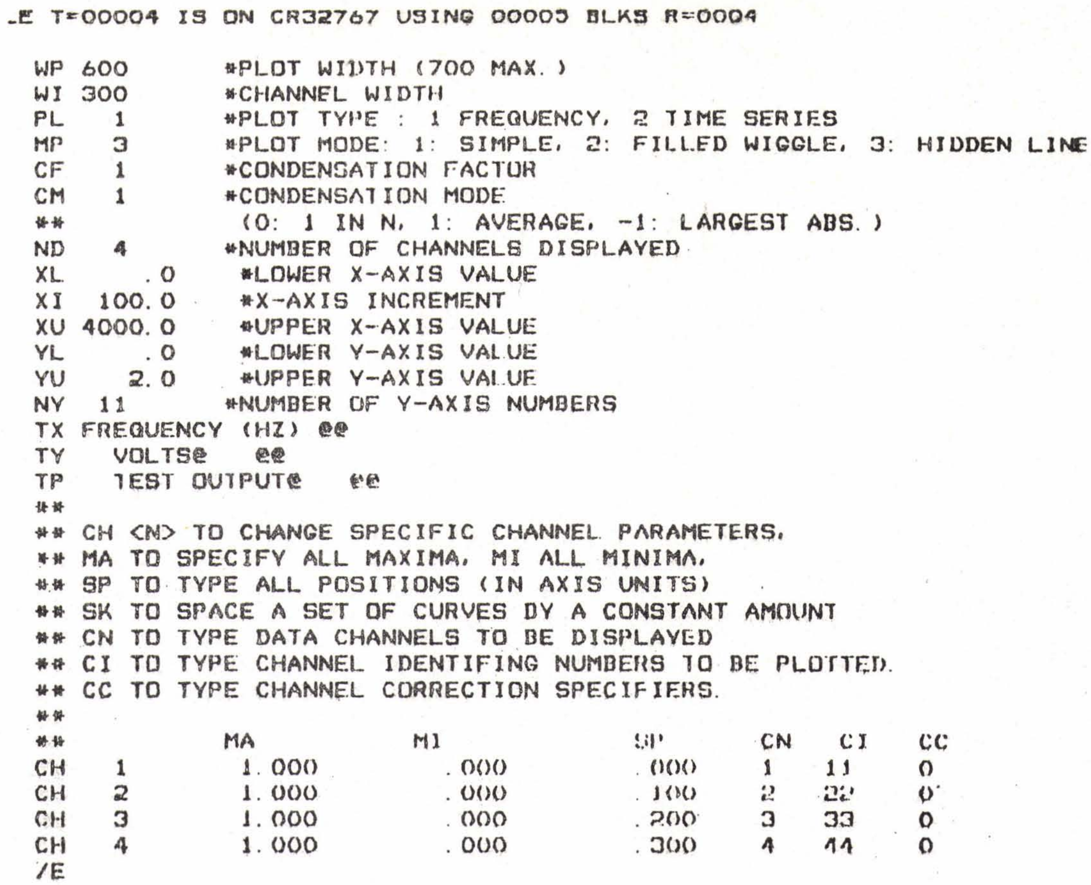


FIG. 11 EXAMPLE OF CREY SCALE TEST PLOT, SHOWING INHERENT CONTOURING.



```
.E T=00004 IS ON CR32767 USING 00005 BLKS R=0004

  WP 600        *PLOT WIDTH (700 MAX. )
  WI 300        *CHANNEL WIDTH
  PL   1        *PLOT TYPE : 1 FREQUENCY, 2 TIME SERIES
  MP   3        *PLOT MODE: 1: SIMPLE, 2: FILLED WIGGLE, 3: HIDDEN LINE
  CF   1        *CONDENSATION FACTOR
  CM   1        *CONDENSATION MODE
  **             (O: 1 IN N, 1: AVERAGE, -1: LARGEST ABS. )
  ND   4        *NUMBER OF CHANNELS DISPLAYED
  XL      .0     *LOWER X-AXIS VALUE
  XI  100.0      *X-AXIS INCREMENT
  XU 4000.0      *UPPER X-AXIS VALUE
  YL      .0     *LOWER Y-AXIS VALUE
  YU     2.0     *UPPER Y-AXIS VALUE
  NY  11        *NUMBER OF Y-AXIS NUMBERS
  TX FREQUENCY (HZ) @@
  TY   VOLTS@   @@
  TP   TEST OUTPUT@   @@
  **
  ** CH <N> TO CHANGE SPECIFIC CHANNEL PARAMETERS,
  ** MA TO SPECIFY ALL MAXIMA, MI ALL MINIMA,
  ** SP TO TYPE ALL POSITIONS (IN AXIS UNITS)
  ** SK TO SPACE A SET OF CURVES DY A CONSTANT AMOUNT
  ** CN TO TYPE DATA CHANNELS TO BE DISPLAYED
  ** CI TO TYPE CHANNEL IDENTIFING NUMBERS TO BE PLOTTED.
  ** CC TO TYPE CHANNEL CORRECTION SPECIFIERS.
  **
  **            MA          MI           SP      CN   CI   CC
  CH   1        1.000        .000        .000    1    11   0
  CH   2        1.000        .000        .100    2    22   0
  CH   3        1.000        .000        .200    3    33   0
  CH   4        1.000        .000        .300    4    44   0
  /E
```

FIG. 12 EXAMPLE OF PLOT CONTROL FILE BASED ON 2-LETTER CODE WITH COMMENTS.

DISPLAY TECHNIQUES

by

S F Meatcher
Admiralty Underwater Weapons Establishment
Portland, Dorset, England

ABSTRACT This paper sets out to review current sonar information displays and to indicate how technological advances over the past decade lead to the concept of a common information display console.

To appreciate the problems involved a brief insight into the nature of the information to be displayed in shipborne systems is given, from which is derived a summary table of the basic display parameter requirements. It is shown that the wide variety of data types required to be displayed may be reduced to a small family of display categories, leading to the conclusion that it should be possible to implement a display system which is sufficiently flexible to be reconfigured dynamically to any category.

An additional requirement for sonar displays is that the operator should be able to interact effectively with the system via the display. Accordingly a discussion is included of current interactive devices together with their advantages and disadvantages.

To examine some of the problems of sonar information display an experimental system was procured in 1975 by AUWE and the salient features of this system are briefly outlined. Much of the work with this system has been on the problems associated with formatting of the sonar data for display and, by way of illustration, some of these formats are presented, together with their potential benefits and problems.

Although much valuable experience has already been gained, a number of questions remain unanswered, particularly those relating to the use and scope of modern display techniques. A new experimental facility being acquired by AUWE is described. This will be used to examine the practicality of the common display console approach.

INTRODUCTION

Shipboard systems utilising a display of some kind to interface between the machine and the operator have, in the past, been designed to match the display to the data output from the system and utilised the most convenient technology to implement the display functions. This had led to a proliferation of display device types and the consequential problems of logistic support, training needs and inflexibility.

Much work has been carried out at AUWE to resolve the problems associated with shipborne displays with the aim of formulating a coherent policy for future display systems.

DATA TYPES

The first step in this process was to look at the various equipments in use and note the type of display format used to present the data to the operator. The summary of data types shown in Fig 1 gives some indication of the wide range of displays in a modern ship.

ACTIVE SONAR	PASSIVE SONAR	AIO/FC	OTHERS
B SCAN PPI A SCAN OVERLAYS LIBRARY COMPUTER-MARKERS TOTE CURSORS	TIME/BEARING TIME/FREQUENCY SPECTRA THREAT LINES LIBRARY COMPUTED TRACKS CURSORS	LPD TOTE BOA F C SOLUTION WEAPON STATUS PREDICTIVE DISPLAYS	BATHYTHERMOGRAPH RAY TRACE ECHO SOUNDER PERISCOPE NAVIGATION M/C STATUS SHIP STATE RADAR INTELLIGENCE ECM

FIG 1 TYPES OF DATA TO BE DISPLAYED

If one extracts the parameters required of a display system to implement these varius formats, the problem of minimising the variety of display types begins to appear a little more tractable.

	REQUIRED CHARACTERISTICS					
SENSOR	MULTI GL	HIGH RESOLUTION	CHARACTERS	MULTI-SCREEN	OVER-LAY	WATER-FALL
ACTIVE SONAR	O	O	O	O	O	
PASSIVE SONAR	O	O	O	O	O	O
AIO/FC		O	O	O		
B/T			O			
RAY TRACE			O			
ECHO SOUNDER	O					O
PERISCOPE	O	O			O	
NAVIGATION	O	O	O		O	
M/C STATUS			O			
SHIP STATE			O			
RADAR		O	O		O	
INTELLIGENCE			O			
ECM	?	?	?	?	?	?

FIG 2 DISPLAY REQUIREMENTS FOR VARIOUS SENSORS

Fig 2 shows, in general terms, the characteristics required in a display and it is apparent that a display system which combines the features of multi-grey level, resolution, and multi-screen with the ability to generate characters, scroll data and overlay data would fulfil virtually all identifiable shipborne display requirements. The design process for such a display system required that we first consider what display techniques were available to us, or likely to become available in the foreseeable future. A review of the wide variety of display techniques was carried out [1] , the results of which are summarised in Fig 3.

TECHNIQUE	ACTIVE SONAR		PASSIVE SONAR		QUALITATIVE	QUANTITATIVE	STATUS
	SURV	CLASSN	SURV	CLASSN			
PAPER	X		X	X			
FILM	X						
LONG PERSISTENCE CRT	X	X					
REFRESHED CRT	X	X	X	X	X	X	X
PLASMA					X	X	X
ELECTRO-LUMINESCENT	O	O	O	O	X	X	X
LED						X	X
LIQUID CRYSTAL	O	O	O	O	O	X	X
LIGHTS							X

(O - Techniques which may become available in the future
X - Techniques currently available

FIG 3 DISPLAY TECHNIQUES

The display requirements are shown as the five distinct categories to which data types may be reduced. viz:

Surveillance - Raw data output from active or passive sonars, probably with some form of overlay.

Classification - Processed data.

Qualitative - Situation display, graphics.

Quantitative - Alphanumeric displays giving factual information such as range, bearing, depth etc.

Status - System condition eg Ready/Not Ready, Open/Closed.

Of the categories shown, surveillance and classification require a grey-level capability whilst the others are normally single level.

It is evident from this survey that the only technique currently available which fulfils all of the requirements is the refreshed CRT. There are, however, two modes in which the refreshed CRT may operate - cursive or raster scan - and it is worth looking at their relative merits. The cursive mode requires three inputs - X, Y and Z - to drive the spot around the screen. Data is time serial and the more complex the picture the longer it takes to write and hence the lower the refresh rate. If overlays in the form of charts, alphanumerics, computer markers etc are required, these have to be time-division multiplexed with the data, reducing the refresh rate still further. The raster mode has a spot scanning the screen repeatedly in a regular fashion. Only Z input is required to brighten the screen at the desired X and Y co-ordinates but Z input must be synchronous with the scanning sequence. If overlays are required they may be readily mixed with the incoming data stream with no modification of the refresh rate, but again the requirement for synchronism exists. The more important differences between cursive and raster are summarised in Fig 4.

CURSIVE	RASTER
REFRESH RATE VARIES WITH DATA	REFRESH RATE CONSTANT
OVERLAYS REDUCE REFRESH RATE	NO EFFECT
BRIGHTNESS VARIES WITH DATA RATE	BRIGHTNESS CONSTANT
REFRESH MEMORY SIZE $\propto$ DATA	REFRESH MEMORY SIZE $\propto$ RESOLUTION
SCREEN RESOLUTION INDEPENDENT OF BANDWIDTH	RESOLUTION $\propto$ BANDWIDTH
DIFFICULT TO MIX INDEPENDENT PICTURES	EASY
INTERACTION STRAIGHTFORWARD	INTERACTION MORE COMPLEX

FIG 4 IMPORTANT DIFFERENCES CURSIVE/RASTER

To achieve the display requirement for a completely flexible workstation all these factors have to be considered and on balance a raster modesystem is favoured. Other factors which influence the choice of raster-mode are:

Ease of transmitting complete pictures around the ship - only a single coaxial cable is required.

Video switching and video mixing are well tried and tested techniques in the commercial world and have considerable potential for shipboard use. For example, displays from any source (eg active sonar) may be routed, on demand, to any other operator (eg passive sonar) as an aid to his decision making process.

Pictures from remote cameras for navigiation or machinery monitoring may be simply called up for display at any workstation as and when required.

FIRST EXPERIMENTAL SYSTEM

For the research stage the decision was taken to adopt, initially, the UK television standard of 625 lines largely to take advantage of commercial developments. Accordingly a specification was drawn up in 1974 for an experimental display facility. This system (Fig 5) basically comprised on twin digital data disc used as a picture memory, together with the associated electronic circuits to format and buffer data, generate timing signals and control data transfers between CPU and discs. The system generates pictures with an interlaced raster of 625 lines at 50 fields/sec refresh rate. Each line comprises 416 cells which, together with 584 visible lines, gives a total screen area of some $\frac{1}{4}$ million pixels. Each picture may be programmed to have an intensity range of 2, 4, 8 or 16 levels. The total system storage capacity is 8 full pictures of 16 grey levels and proportionally more at lower intensity ranges. The disc memory is organised as two separate stores referred to as "Back-up" and "Video". Picture data is written to the back-up disc one TV line at a time and when a picture is complete it may be transferred to the video disc. Since disc read amplifiers tend to saturate when a disc is being written a picture displayed from this disc shows a characteristic white line across the screen at the current write line address. An undisturbed picture is obtained by displaying from the video disc until a picture has been completely assembled on the back-up disc and then transferring a complete picture from back-up to video. Whilst the process of writing the video disc is continuing the display is automatically enabled from the back-up disc.

FORMATTING DATA

Much of the work with the experimental system has been directed towards formatting active sonar data for 625 line raster-scan display, although some work has been carried out on passive and AIO displays. The data base used for the active work came from two active sonar systems. One was a long range, dual ping, multi beam, doppler detection system, the other was an experimental long range, high resolution sonar. A large number of data tapes were recorded, at sea, under a variety of typical operating conditions.

There are a number of problems arising from the use of conventional long persistence CRTs for displaying long range active sonar information. The operator requires to view the display constantly if he is to observe all signals before they fade, and this leads to vigilance problems. In addition there is little scope for re-examining returns within the ping duration. Initially, therefore, efforts were directed to producing a display whereby all the returns from a single ping were stored, the display refreshed at 50 Hz from the stored data, and all the data from that ping presented to the operator with individual intensity levels preserved until overwritten by data from the next ping. The benefits

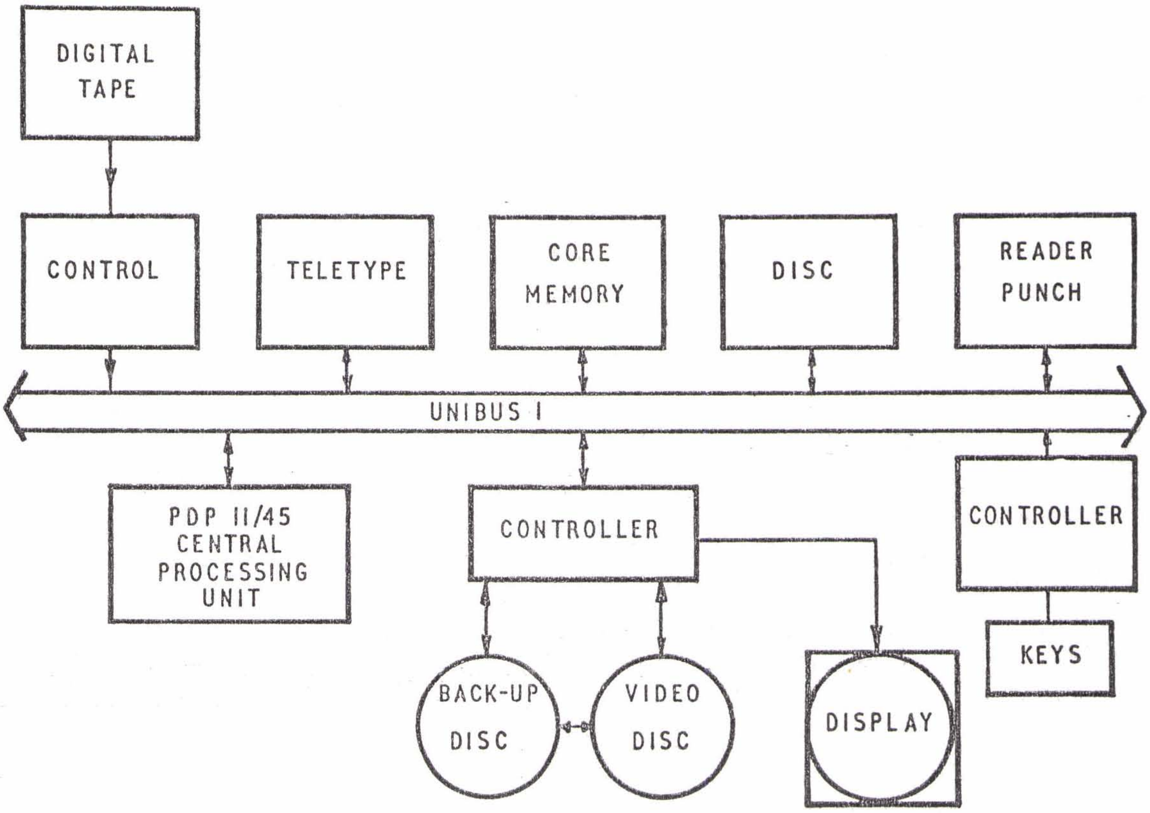


FIG. 5 DISC BASED DISPLAY SYSTEM

from this to the operator are immediate and fairly obvious. He is no longer constrained to following a crawling line to detect targets, he can scan the screen at random, and, because relative intensities are preserved, tracking is much simplified.

An important outcome of these very early efforts was to demonstrate the feasibility of formatting data for refreshed displays in real-time and to get some idea of the amount of computing effort involved. It was shown that there would be no severe requirements in terms of speed, power and size of memory.

A powerful aid to detection for a surveillance display operator is the provision of a historical record of what has happened on previous pings. One of the earlier attempts at this is shown in (Fig 6). Here the output from 8 beams are displayed parallel to each other. Within each beam a history of the last 8 pings is displayed with the oldest ping to the left and the current ping to the right. Using this technique tracking targets stand out very clearly and are easy to detect but bottom reverberation also produces many track-like marks which can lead to an undesirably high false alarm rate. This technique does, in fact, compress the raw data and deprives the operator of many of the clues which are available to him from the simple refreshed display. It appears prudent therefore, when using this format, to display the current ping, in full, adjacent to the history display. Other formats which combine the advantages of history with low false alarm rate have been devised and are currently being evaluated for fleet use.

An interesting example of how dramatic improvements may be made to very cluttered sonar displays is shown in (Figs 7 and 8). (Fig 7) shows the output from a dual ping, ripple transmission, doppler sonar. The axes are, range in Y, beam number in X, doppler channel within each beam, while between each beam is displayed the returns from a second, shorter, ping. The first point to note is that, because the system has a ripple transmission, returns from any target which appear in adjacent beams do not have the same Y co-ordinate on the screen and the returns from the second, short, ping are staggered further. This makes it difficult for the operator to correlate returns from a common target. Once the capability to store all the data is provided it is but a trivial exercise to introduce the appropriate range corrections. However, for this sonar, we wanted to go a stage further and to include the benefits of a history display and, at the same time, to capitalise on the fact that, being a dual ping sonar, there existed two independent samples for each range increment on every transmission. For each transmission, at each range increment, and for each beam, the maximum level from any doppler channel within a beam was extracted, stored, and displayed as range in Y against beam number in X. Displayed adjacent to this and to the right is the short ping data. On successive transmissions the doppler data is displaced one pixel to the left and the short ping data one pixel to the right, with the new data written into the original positions. The result of this is shown in (fig 8). This format is called "Arrowhead" for fairly obvious reasons, closing targets produce an arrowhead pointing down whilst receding

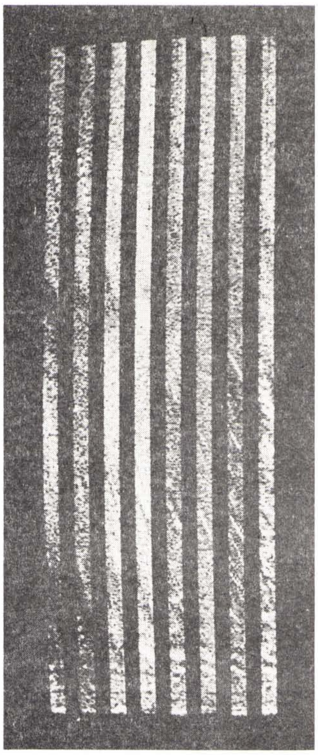


FIG. 6 BEAM HISTORY FORMAT

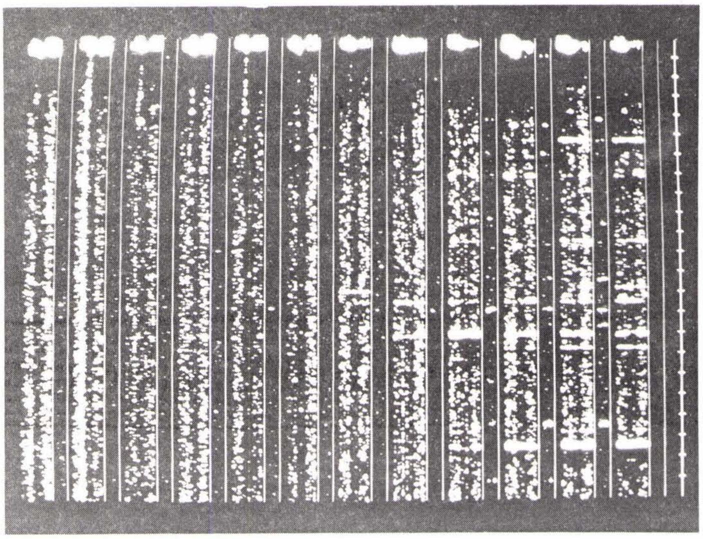


FIG. 7 DUAL PING DOPPLER SONAR

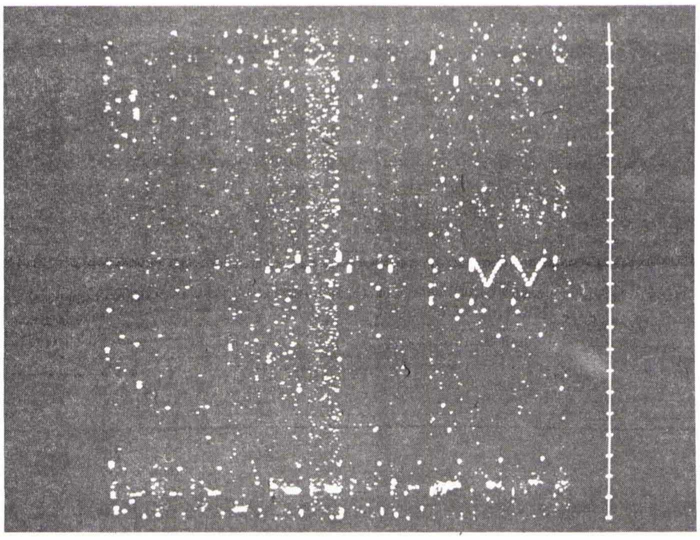


FIG. 8 ARROWHEAD

targets produce an arrowhead pointing up. This technique would also appear to have applications in situations where more than one sonar is in use.

INTERACTION

An operator must be able to communicate with a system to extract or inject information, modify the mode of operation, to initiate tasks or to perform tracking activities. There are many ways in which an operator may communicate with the system and it is incumbent upon the display designer to ensure that the interactive device chosen is optimised for the task in hand. An indication of the range of interactive devices currently available is given in fig 9.

DEVICE	CHARACTERISTICS
DIGITISER	Display mounted. Continuous or discrete steps. Good spatial selection, flexible good feedback correspondence.
ISOMETRIC JOYSTICK	Force operated, robust, suits tracking/positioning tasks.
ISOTONIC JOYSTICK	Displacement operated. Suits tracking/positioning tasks.
KEYBOARD	Fixed format, accurate, fast after training.
LIGHTPEN	Good spatial selection, flexible, good feedback.
PUSHBUTTONS	May be grouped like keyboard or around display to perform like discrete digitiser.
TRACKBALL	Alternative to joystick, ballistic/continuous, fast accurate.
ROTARY CONTROL	Cheap, compact, single variable.
SPEECH INTERPRETER	Hands free, complex, potentially good for data selection tasks.
TABLET	Continous, compatible with pen and paper, spatial selection depends on output device.
SWITCHES	Cheap, simple, good feedback, inflexible.

FIG 9 INTERACTIVE DEVICES

There is, unfortunately, no simple definitive way in which input devices for interactive tasks may be selected, there is inevitably some trade-off. A report by EMI Electronics, England [2] highlighted some of these difficulties. It was shown that, whilst speed and accuracy of various devices could be assessed for simple activities such as selection, position, inject, the speed with which a procedure was performed was heavily dependent upon the design of the procedure itself. Also the results depend to an extent on what activities precede and succeed the procedure, and on the likely extent of errors occuring during the procedure and the time taken to correct them.

If, as seems likely, more than one device is needed to perform all the tasks required of future workstations, then consideration must be given to the way in which devices may, or may not, complement one another. It will not be sufficient to simply optimise a procedure for a given input device or to select what appears to be the most appropriate device for a given task without due consideration of the way in which the choice of device, or design of procedure, influences other procedures. Consider, for example, a procedure which contains largely tracking/ positioning type activities (for which a joystick may be the most suitable device) but which has some limited selection activity (for which a light pen may be more suitable). Analysis may show that the best overall efficiency for this particular procedure is achieved by using the joystick for both tracking and selection activities since no device selection overheads are incurred. If, however, the operators duties also include procedures which are composed largely of selection type activities then, should such a procedure follow the previous procedure, a change of input device may be thought appropriate for best device/ activity match. This could cause problems for the operator, however, in terms of increased stress and momentory disorientation which may outweigh the advantages of choosing what would appear to be the optimum interactive device. The designer must be aware of these problems and ensure that the operators task as a whole is considered. In the absence of any precise design rules this could well involve many iterations of procedural design.

FUTURE TRENDS

Many questions remain to be answered with regard to workstation design of which a large proportion are inter-related. For instance:-

What is the optimum resolution in terms of pixels/display area?

How many screens should be used, what size, how should they be juxtaposed and how should the data for any given task be distributed around the available screens?

Which aspect ratio best matches data display and interactive requirements?

What is the minimum number of types of interactive device required to support foreseen interactive tasks?

It does seem clear, however, that for the display itself there is unlikely to be any serious contender to the CRT in the immediate future and that semiconductor RAM is the most likely technology for picture refresh memory. It was with these thoughts in mind that a new experimental facility for AUWE was proposed. The system [Fig.10], currently being built by Sigma Electronic Systems of Great Britain, has the salient features of:

Intelligence
Graphics Memory
Large pixel address range
Variable organisation of pixel memory in X, Y and Z
Display resolution up to 1024^2 pixels
Interactive processor supporting several interactive devices

The intelligent part of the system responds to coded instructions from the host processor to perform display orientated tasks such as vector drawings, character generation, pixel plane allocation, masking, initialisation and bit packing and unpacking to pixel memory. In addition it will run graphics programs stored in its own graphics memory, as a series of instructions, upon receipt of a single 'GO' command from the host processor.

The pixel store is conceptually 32 planes of 4096 x 4096 pixels with the physical semiconductor RAM, pixel memory of user size (say 512 x 512 or 1024 x 1024) relocatable anywhere within this range. The total capacity of the system will be constrained at present to about 48 Mbits by physical limitations.

The interactive processor enables the user to define and run sophisticated interactive tasks with minimum overheads on the host processor.

This experimental facility will be used intensively over the next two years to enable AUWE to produce a design for a general purpose workstation which will form one of the major building blocks of all future shipborne systems.

REFERENCES

1. MORRIS J C. 'Integrated display concepts for future generation submarines'. Publication 44997 (Restricted) AUWE, Portland, Dorset, England, 1978.

2. RUMENS D J. 'Ergonomics for future submarine systems: Documentation of human performance data on specified input/output devices. MEMO No 451 VOL I. EMI Electronics Ltd, Systems and Weapons Division, Feltham, Middlesex, England, 1978.

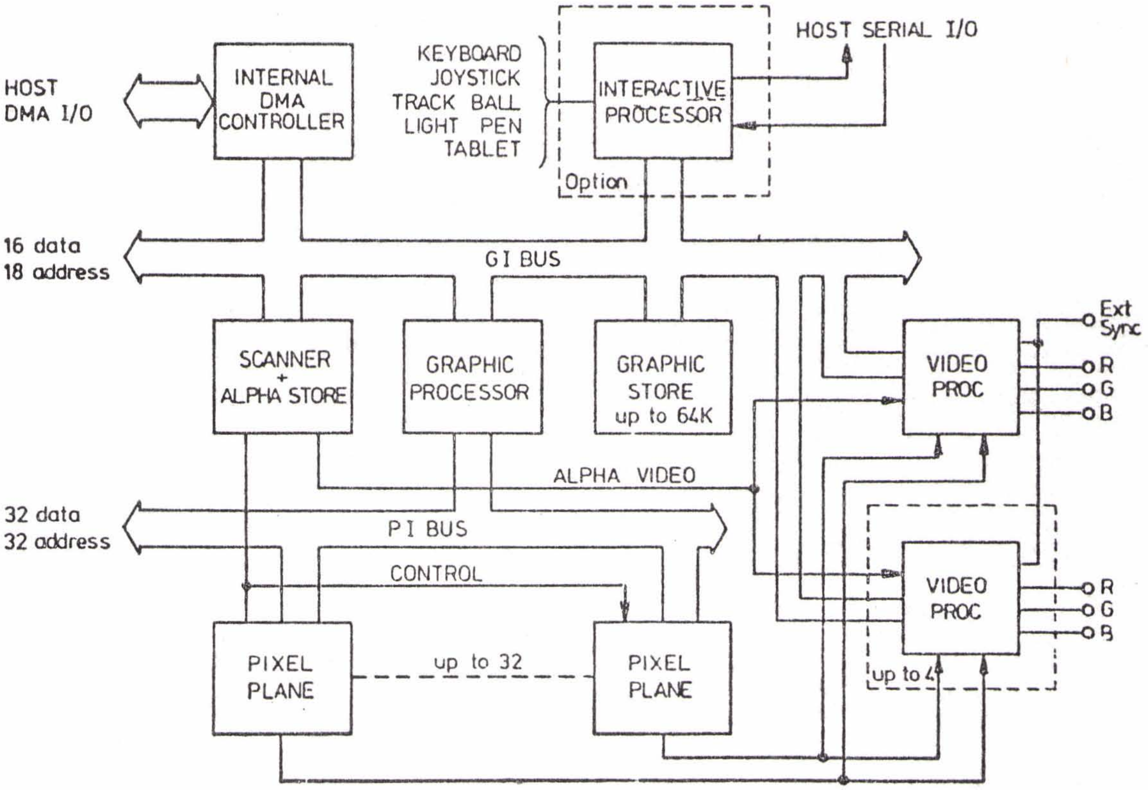


FIG. 10 NEW EXPERIMENTAL FACILITY

DISCUSSION

E. Cernich For a given display system and a number of functions that the operator has to perform on the displayed data, what is the criteria to select: to give the commands by specialized keys or by a keyboard?

S. Meatcher It is not intended that there should be any dedicated keys on the common console. If keysets are used it is intended that their functions should be redefinable under software control, i.e. they will be used as "soft" keyboards.

CAPT A. Newing Is it the intention to restrict the generalized work station to sonar applications or to make it applicable to all sensors and shipboard displays?

S. Meatcher Initially our efforts will be directed to producing a common work station for all sonar and directly-related applications but other areas, such as machine monitoring and ship status, are not being ignored.

B. Pennoyer Please comment on the use of colour since it is missing from your list of parameters.

S. Meatcher At this point in time it has not been shown that the use of colour will confer any significant operational benefits. It is our intention to use monochrome displays until such time we are convinced that the use of colour is cost-effective. It should be pointed out that the two experimental systems shown will in fact support full colour displays.

A FAST DISPLAY PROCESSOR FOR SONAR ECHOES

by

Helge Bodholt
Simrad A/S
Horten, Norway

ABSTRACT This paper presents a display processor for a CRT, which shows a horizontal projection of underwater targets. Sonar echoes are drawn as short lines across a 5° beam, in random scan and with 50 Hz refresh. The high speed requirements is met by digital integration (TTL low power schottky). One of the main advantages of this display processor, relative to alternative principles, such as PPI or raster scan, is brightness. The screen can be viewed in full daylight. This is achieved because the beam spot is positioned only to points with light.

INTRODUCTION

As part of a display system for a fish finding sonar a fast, real time display processor is developed.

The principle and the advantages of this display processor is described here.

The display which is a CRT, shows a horisontal projection of underwater targets.

1 THE PICTURE ON THE CRT SCREEN

The display processor draws sonar echoes as short straight lines perpendicular on the beam direction. These echo vectors fill a 5° beam width on the screen. See figure 1. There are 256 range cells in each beam. An echo vector can have one of four intensity levels, including black.

The picture is drawn in random scan. The beams are plotted one after another. Within each beam the plotting sequence is from the own ship position and out to maximum range.

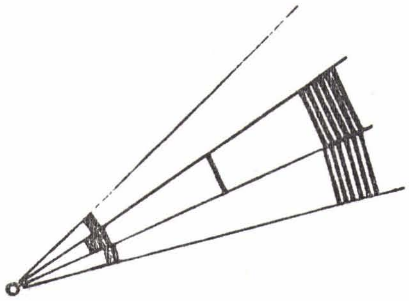


FIGURE 1.

2 THE DISPLAY PROCESSOR JOB IN THE SYSTEM

Figure 2 shows how the display processor is connected to the rest of the system. The display processor fetches the display data from the computer memory where they are placed by the computer program. The display processor has direct memory access with cycle stealing. The computer chosen for this project was ALPHA LSI from Computer Automation. The display is a Cathode Ray Tube with short persistence phosphor. Hewlett Packard 1311 was chosen. It has electrostatic deflection which gives high speed in moving of the spot.

It is the job of the display processor to generate X and Y deflection voltages for the echo vectors. The picture is refreshed with a rate of 50 Hz. This refresh is syncronized with the 50 Hz line frequency. Thereby hum in the deflection amplifiers will give only static errors and not a waving motion in the picture.

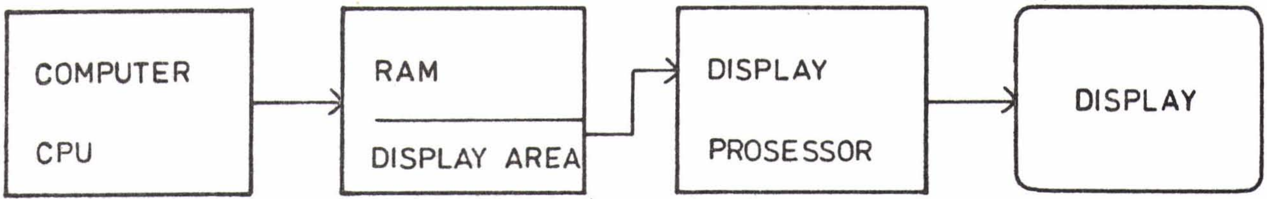


FIGURE 2.

3 FUNCTIONAL DESCRIPTION

The display processor has two identical addition units, one for X deflection and another for Y deflection. Only X deflection will be described here. Figure 3 and 4 illustrate how the echo vectors are drawn. The processor starts by loading the start position X_o into the X register. With the addition loop in the right part of figure 4 increments ΔX are added. ΔX is one range cell. The beam spot now moves out along the beam direction with the spot blanked. This continues until an echo vector with nonzero intensity is found. Then the left add loop takes action and generates the echo vector unblanked. A special circuit, not shown in figure 4, stops the incrementation when the echo vector has reach a length corresponding to the 5° beam width.

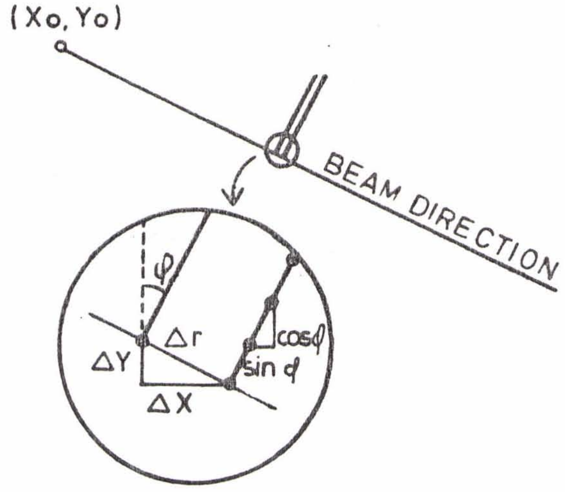
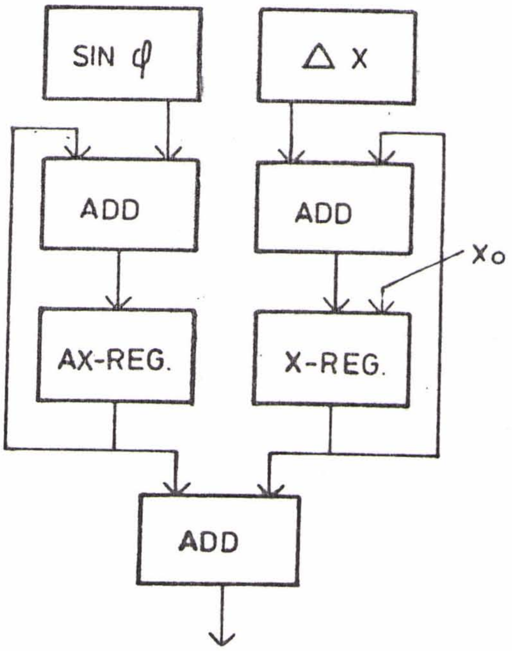


FIGURE 3



X-DEFLECTION

FIGURE 4

When the echo vector has reached this length the AX register is reset to zero, which brings the spot back at the beam direction line ready for a new ΔX increment.

The increment ΔX is set by the computer program without restrictions. This is utilized to control the display scale. A large ΔX gives a magnification of the picture, and can be used to study details.

The beam can be offset by changing X_o.

The increments between points in an echo vector are restricted in size in order to keep a constant point density and thereby a constant brightness. The angle ϕ of the echo vector is defined relative to the Y axis (north), therefore the X-increment comes out as sin ϕ.

The final point resolution on the screen is 1024 x 770. The resolution in the add loops is however much finer, this is in order to prevent accumulated position errors.

When all beams are drawn the display processor rest until the next refresh pulse arrives. The refresh period is 20 ms. If the display processor do not reach to draw all echo vectors in the 20 ms it will continue into next period. This means that no echoes are lost, but the refresh rate is halfed which cause flicker. This serves as a warning to the operator that he should lower the sonar gain or in other ways reduce the number of echoes. In normal use, with the own ship position in the center of the screen and sonar range out to the screen perifery, the display processor can cope with up to 25% nonzero in 27 beams (= 135° search sector).

Figure 5 shows a block diagram of the display processor.

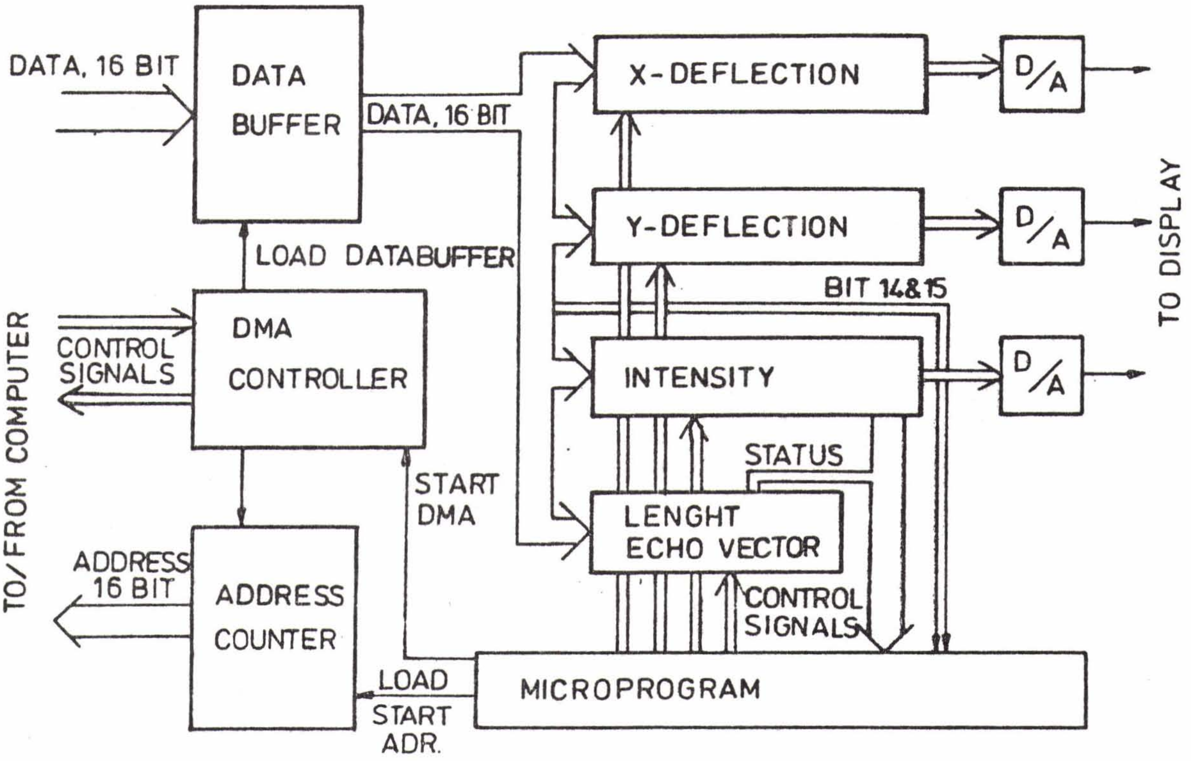


FIGURE 5

4 FORMAT OF DISPLAY DATA

The display data are placed in the memory one beam after another. The data for one beam consists of a data head and 256 intensities. The head has all information specific for this beam. See figure 6. Since the memory word lenght is 16 bit and the intensity is only 2 bit per range cell the intensities are packed, 8 in each word. The size of the display area in the computer memory is (6 words for the head + 32 words for intensities) x number of beams.

There is some redundancy in the data in the head. The reason for this is to free the display processor from timeconsuming trigonometric calculations. They are instead done by the computer program.

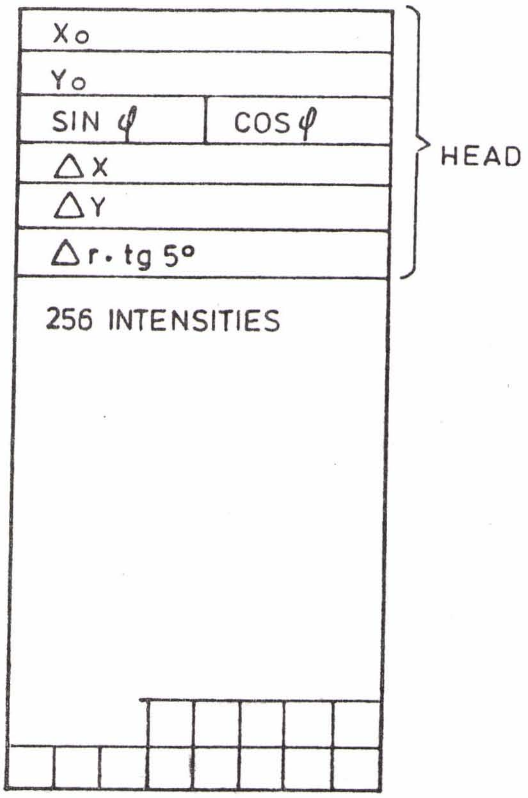


FIGURE 6

5 ADVANTAGES

The goals in the design of this diplay processor were:

1. A bright picture for daylight viewing
2. A picture where large targets, such as fish shools and bottom topography, are shown as bright areas.
3. A flickerfree presentation.
4. Minimum display area in the computer memory

1. A bright picture is achieved by the random scan. Only echo vectors with nonzero intensity are drawn. When the spot moves out along the beam it pass quickly all range cells with zero intensity. A raster scan would give less light output because the spot also devotes time on dark positions. A PPI could give still lesser light if the increasing writing speed in the spiral scan in compensated by lower intensity in the center.

2. In normal use the echo vectors are so close each other that they melt together and form a bright area when the target covers several range cells. The length of the echo vectors correspond to the real sonar resolution, good at short range and poorer at longer ranges.

3. The 50 Hz refresh rate gives flickerfree presentation.

4. The format of display data demands only a modest memory space. The alternative raster scan demands a much larger memory space because every pixel on the screen must have a corresponding memory cell. Such a pixel memory is necessary in a fast raster scan display processor. A random scan alternative with storage of the X-Y coordinates for each nonzero echo was considered. It demands only a small memory space when there are few nonzero echoes, this because all the zeroes do not need to be stored. When the number of nonzero echoes grows, the necessary memory grows.

6 SYMBOLS

In addition to the drawing of sonar echoes the display processor is also able to draw symbols as a vector generator. The computer specify starting point, angle and length. The display processor draw the vector utilizing the same incremental add loop as for echo vectors. Complex symbols can be built up of more separate lines.

7 SWEEP MARKER

It gives a very live impression if the computer program generates a flying sweep marker which moves out the beam just in front of arrived echoes. This echo marker is a single echo vector with maximum intensity. It erase old echoes and leaves the new arrived echoes.

8 PHYSICAL CONSTRUCTION

The circuits are built up of standard TTL low power Schottky and mounted on printed circuit boards which fit into the ALPHA chassis.

The principle of this display processor is patent protected.

Figure 7 is a photo of the display. The echoes in the center are from an artificial target. The echoes farther behind the ship are from the sea bottom.

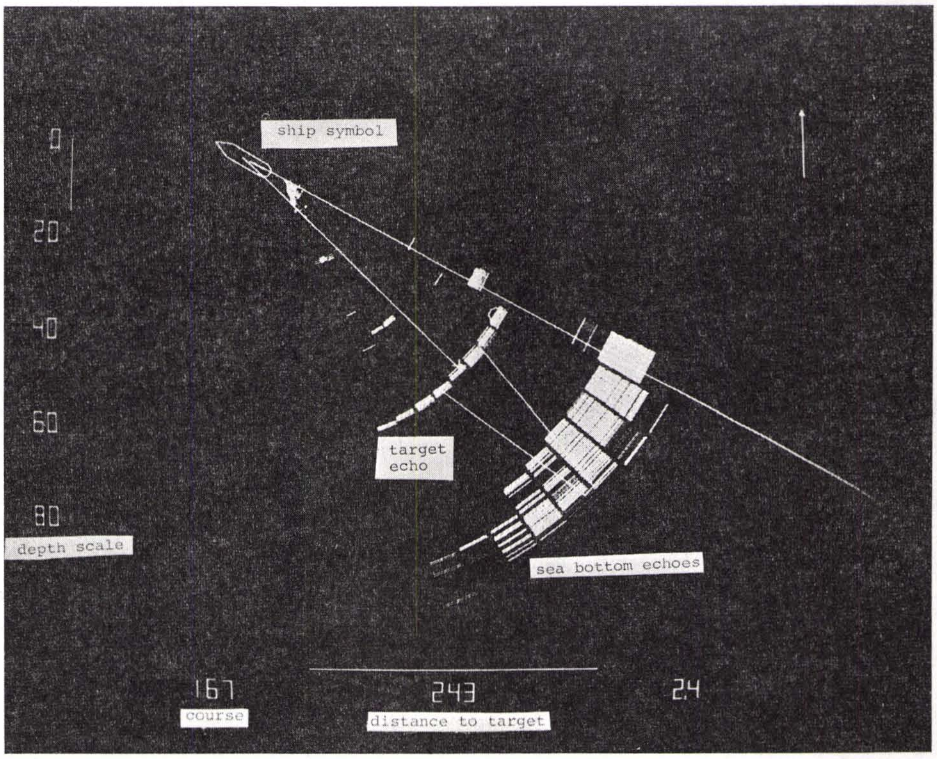


FIG. 7 PHOTOGRAPH OF A CRT SCREEN

PANEL ON DISPLAYS

Chairman: E. Cernich

Members : S.F. Meatcher
M.J. McCann
E. Hug
B. Pennoyer

E. Cernich In view of the interest aroused by the last reply of Mr Meatcher, it would appear that the most appropriate topic for opening the discussion of this Panel should be: "Colour or black-and-white"?

S.F. Meatcher There is lack of evidence showing that colour confers any significant improvement in the operational performance of sonar. Experimental systems will support full colour displays.

Y. Lundh Absence of results due to absence of attempts cannot be taken as proof that colour may not be useful.

CAPT A. Newing Drawing on experience, colour in detection systems makes no difference, but can help in discrimination, particularly for moving targets.

S.F. Meatcher Using synthetic or real data?

CAPT A. Newing Synthetic data.

R. Seynaeve Colour may be used for one or more variables.

S.F. Meatcher How was B/W coded? Were there any flashing or other attention-getters, when the comparisons were made? Colour is an obvious attention-getter.

R. Seynaeve Isn't colour more pleasant for the operator?

T.E. Curtis Mauve and yellow make operators physically sick in some conditions.

J.M. Griffin Colour has use in R & D; fine changes in intensity are easy to display in pseudocolour.

R. Seynaeve And also useful for associating intensity in different areas of the display.

E. Hug The Radiological Clinic of the University of Florence uses colour for medical diagnostic displays and the doctors like it; it helps in recognizing the boundaries of organs and in associating intensities.

J.W.R. Griffiths Colour obviously increases the number of levels that can be distinguised (just noticeable differences). Is this required?

M.J. McCann Sometimes, particularly in remote sensing.

B. Pennoyer Yes, there are indications that colour does not improve detectability. However, we may not be using colour correctly or asking the right questions; there is considerable work in this area at NOSC, particularly in combined brightness and colour coding.

S.F. Meatcher Monochrome is favoured for shipboard operation use; colour is expensive. High resolution and colour to milspec is difficult. The only known high-resolution colour tube is Japanese, and such a single supplier situation is not popular with MOD.

J.M. Griffin Convergence is very difficult too.

D. Nairn I am using 512 x 640. Am I heading towards a dead end?

Y. Lundh No, that is well matched to the resolution of the human eye. For increased resolution use electronic zoom.

R. Seynaeve Let us get back to research. Is colour good for research?

H. Urban Operators like colour. It does not increase detectability, but increases comfort - gives more to research.

D. Nairn Black-and-white photographs are held to be a more exacting test of the resolution of a camera lense than a colour photograph.

R. Seynaeve Physiological averaging by the eye.

E. Goudriaan Experiments have shown that when decision making via grey-tone intensities is required, only 8 to 10 distinct different levels can be used.

J.W.R. Griffiths Detection of tracks does not need very many just noticeable differences on the display because of the visual integration.

Y. Lundh Overly coarse quantization of grey scales can cause false contouring. It is important to realize that the human eye will see such spurious contours in digitally encoded scenes, even for quite small steps. Thus to avoid spurious contouring, quantization must be finer than that required if the eye had to distinguish grey scale values only as absolute tones.

M.J. McCann False contouring can happen with colour displays, also.

J.M. Griffin It can be useful in some applications.

R. Seynaeve Mr Meatcher showed 32 picture planes, why?

S.F. Meatcher This is an experimental system for research. It can drive:

3 x 8-bit D/A converters;
4 Overlay/masking planes;
4 Programmable blinking planes.

Seynaeve Everyone uses colour, except sonar. Why?

. Richardson Sonar rooms are illuminated in red, so perhaps colour printout would not work. However, CRT's should be O.K.

J.S. Pyett Also there are some physiological effects. Some images can go unseen on certain backgrounds; colour perception depends on colour context.

S.F. Meatcher Also some sonar operators may be partially colour-blind, reducing the supply of potential operators.

R. Seynaeve Is there a pseudocolour standard?

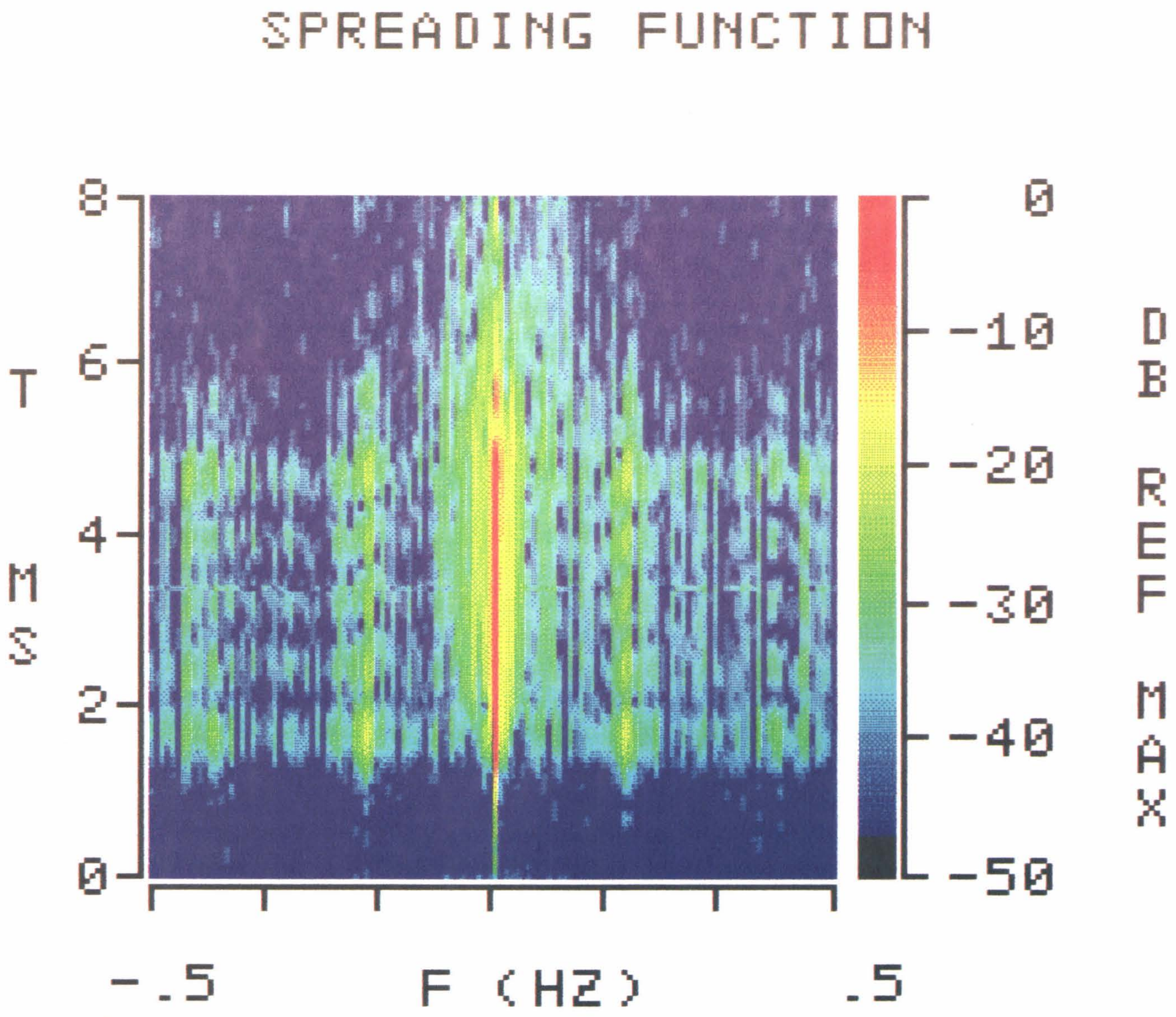
S.F. Meatcher Yes, black to white!

J.M. Griffin Have alternating images been used for detection?

D. Nairn Yes, in the U.K.

S.F. Meatcher We have been displaying the last eight display frames (we call it ripple flashing). This works best for high signal-to-noise ratios. The eye is very confused by noise, and the display is not nice to watch. Increasing the signal-to-noise ratio by using target motion detection, as in radar, is effective. We have found the optimum ripple frequency to be around 10 Hz.

The figure on the next page is an example of a pseudocolour representation of functions of two variables, obtained using the Applicon colour plotter. The data are from an experiment by Ph. de Heering, SACLANTCEN. (Ed.)



SACLANTCEN 1979

CP-25

INITIAL DISTRIBUTION

MINISTRIES OF DEFENCE

	Copies
MOD Belgium	2
DND Canada	10
CHOD Denmark	8
MOD France	8
MOD Germany	15
MOD Greece	11
MOD Italy	10
MOD Netherlands	12
CHOD Norway	10
MOD Portugal	5
MOD Turkey	5
MOD U.K.	16
SECDEF U.S.	61

NATO AUTHORITIES

Defence Planning Committee	3
NAMILCOM	2
SACLANT	10
SACLANTREPEUR	1
CINCWESTLANT/COMOCEANLANT	1
COMIBERLANT	1
CINCEASTLANT	1
COMSUBACLANT	1
COMMAIREASTLANT	1
SACEUR	2
CINCNORTH	1
CINCSOUTH	1
COMNAVSOUTH	1
COMSTRIKFORSOUTH	1
COMEDCENT	1
COMMARAIRMED	1
CINCHAN	1
PAT	1

SCNR FOR SACLANTCEN

	Copies
SCNR Belgium	1
SCNR Canada	1
SCNR Denmark	1
SCNR Germany	1
SCNR Greece	1
SCNR Italy	1
SCNR Netherlands	1
SCNR Norway	1
SCNR Portugal	1
SCNR Turkey	1
SCNR U.K.	1
SCNR U.S.	2
French Delegate SCNR	1
SECGEN Rep. SCNR	1
NAMILCOM Rep. SCNR	1

NATIONAL LIAISON OFFICERS

NLO Canada	1
NLO Denmark	1
NLO Germany	1
NLO Italy	1
NLO U.K.	1
NLO U.S.	1

NLR TO SACLANT

NLR Belgium	1
NLR Canada	1
NLR Germany	1
NLR Greece	1
NLR Italy	1
NLR Norway	1
NLR Portugal	1
NLR Turkey	1

PARTICIPANTS	40
Total initial distribution	274
SACLANTCEN Library	10
Stock	116
Total number of copies	400