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 multi-programming 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 micro-code. 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 : $NH \cdot NB = 2048 \times 48$ 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 : $NH \cdot NB \cdot fo < 3$ MHz

Maximum input data rate : $NH \cdot fi < 3$ 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

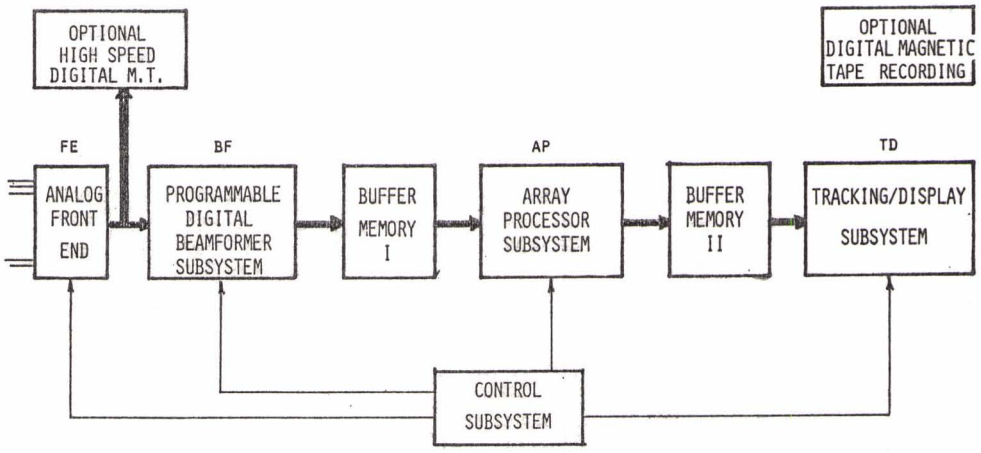

FIG. 1

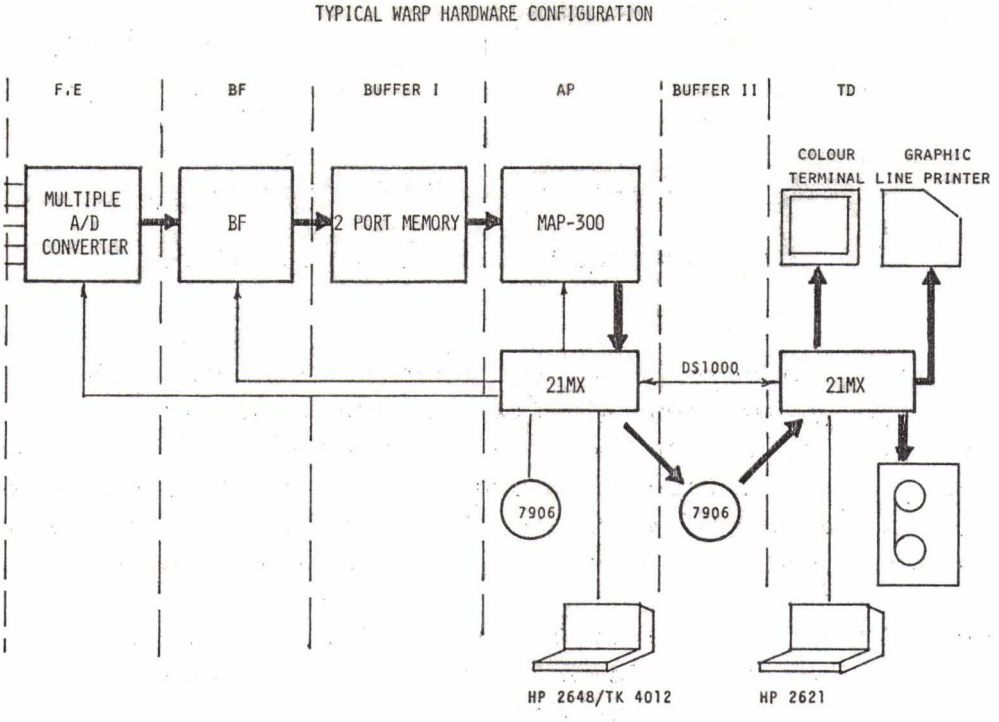
TYPICAL WARP HARDWARE CONFIGURATION
F.E
BF
BUFFER I
AP
BUFFER II
TD
MULTIPLE A/D CONVERTER
BF
2 PORT MEMORY
MAP-300
COLOUR TERMINAL
GRAPHIC LINE PRINTER
21MX
DS1000
21MX
7906
7906
HP 2648/TK 4012
HP 2621

FIG. 2

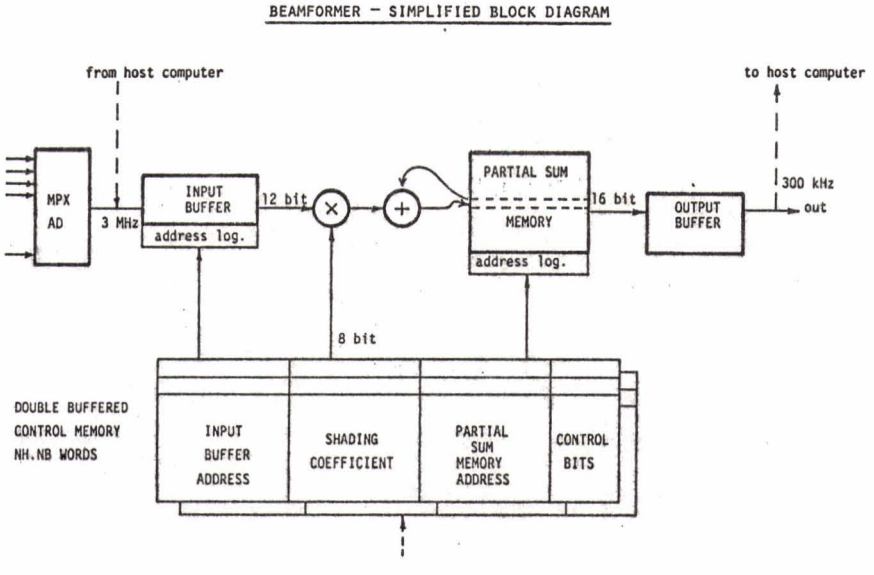
BEAMFORMER – SIMPLIFIED BLOCK DIAGRAM
from host computer
to host computer
MPX AD
3 MHz
INPUT BUFFER
address log.
12 bit
PARTIAL SUM MEMORY
address log.
16 bit
OUTPUT BUFFER
300 kHz
out
8 bit
DOUBLE BUFFERED CONTROL MEMORY NH.NB WORDS
INPUT BUFFER ADDRESS
SHADING COEFFICIENT
PARTIAL SUM MEMORY ADDRESS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

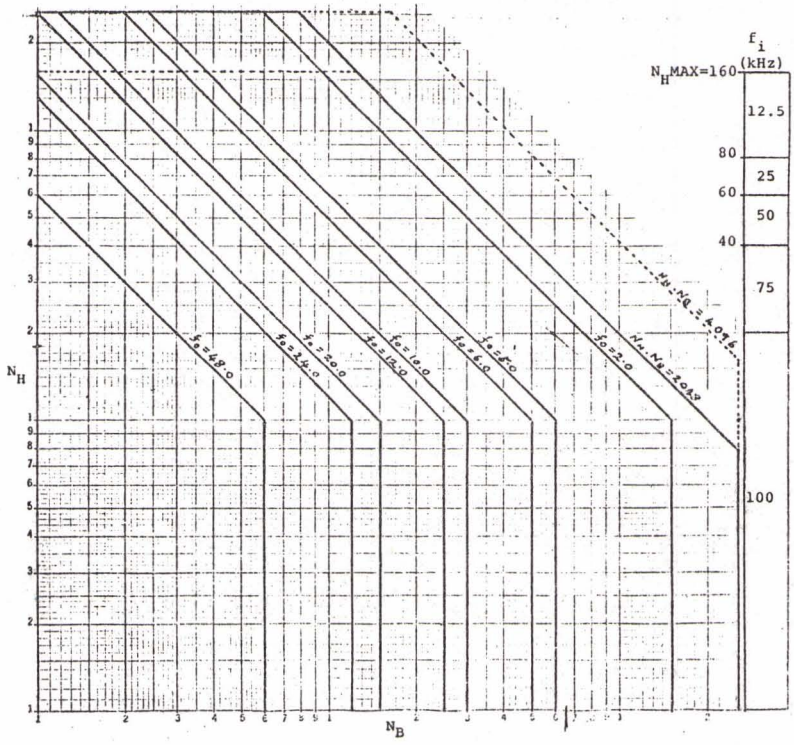

FIG. 4

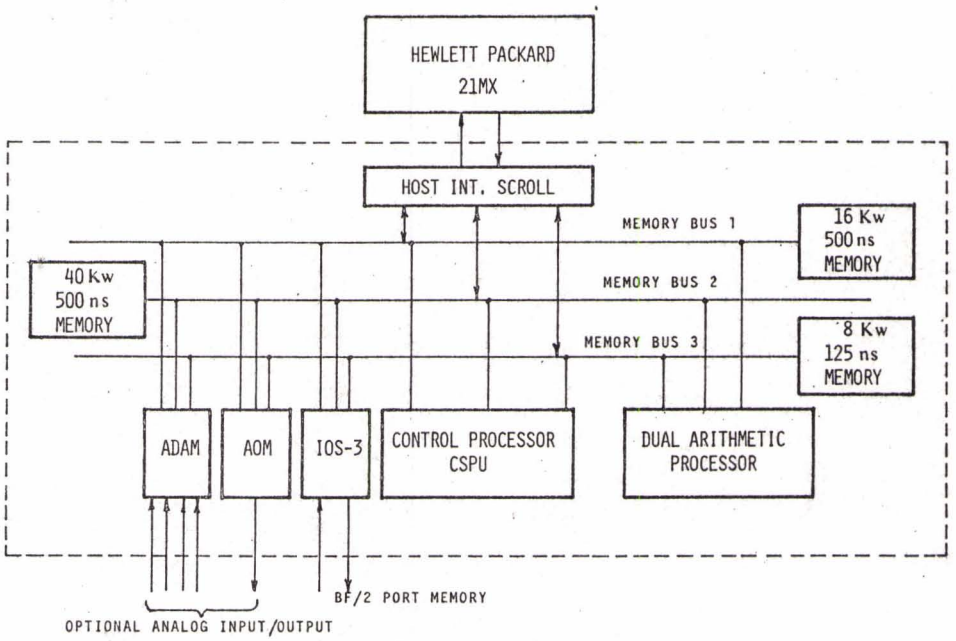

FIG. 5

SIMPLIFIED DIAGRAM OF MAP300 ARITHMETIC UNIT

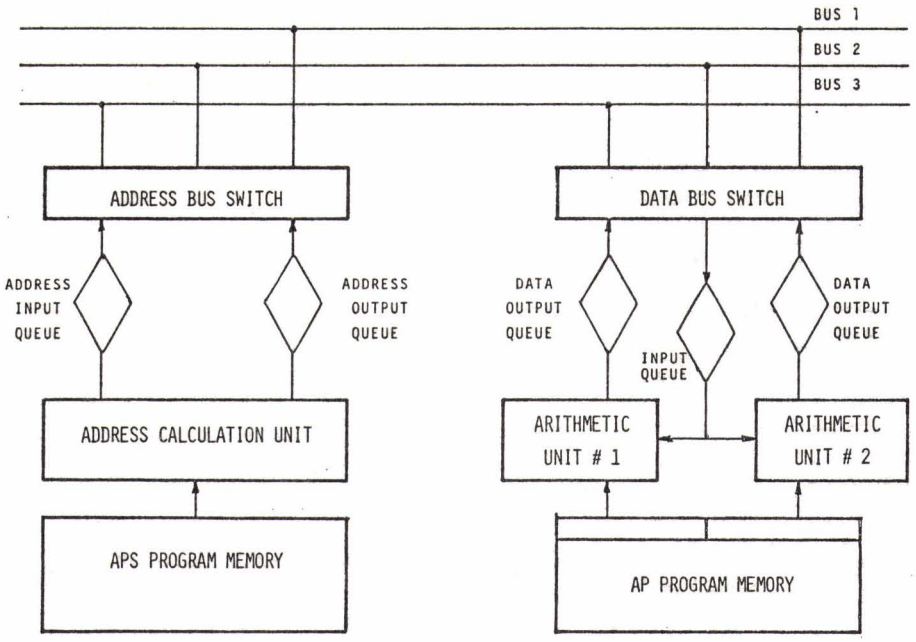

FIG. 6

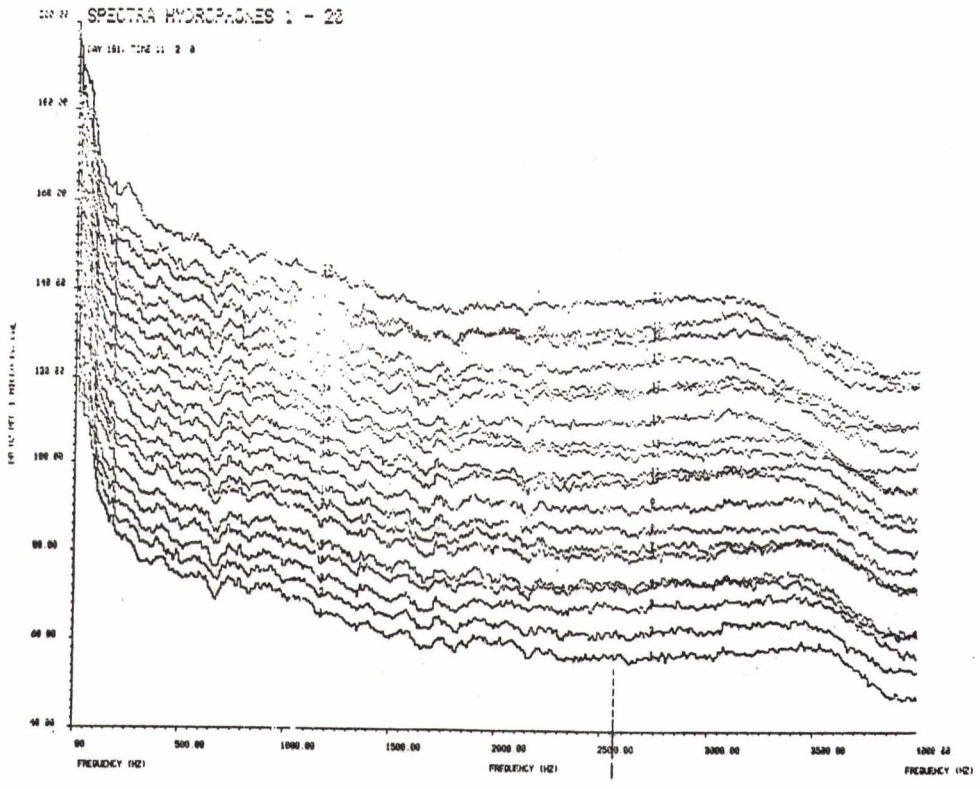

FIG. 7

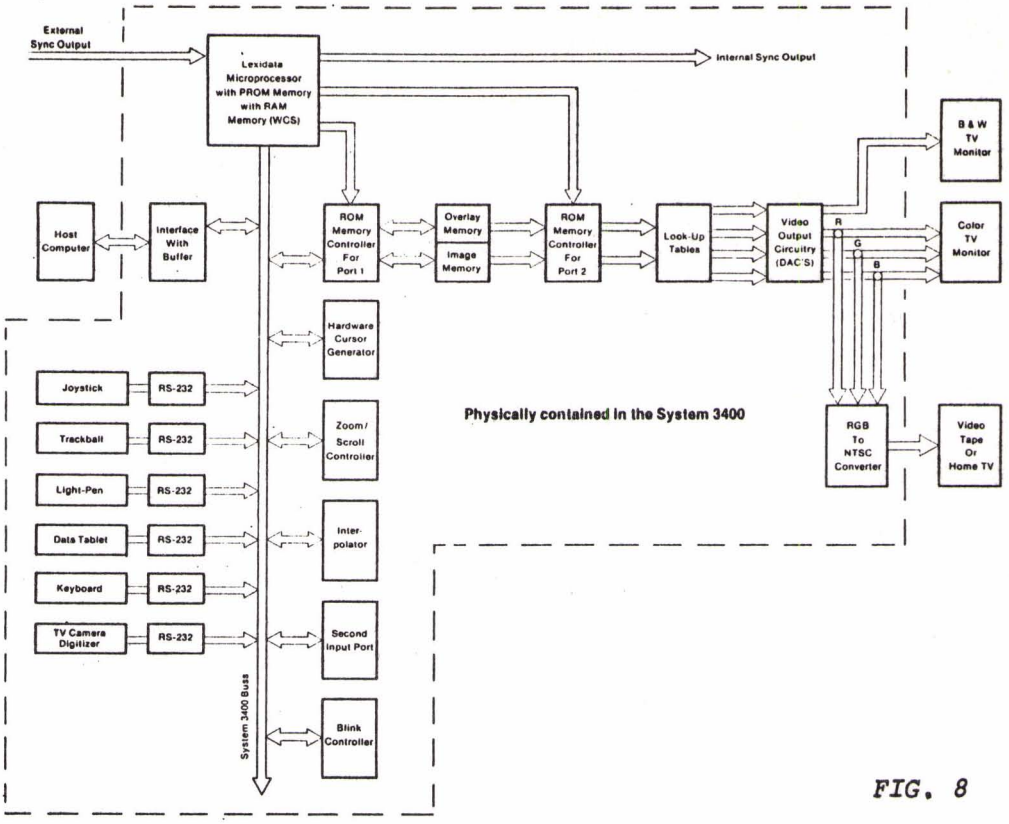
External Sync Output
Lexidata Microprocessor with PROM Memory with RAM Memory (WCS)
Internal Sync Output
B & W TV Monitor
Color TV Monitor
Host Computer
Interface With Buffer
ROM Memory Controller For Port 1
Overlay Memory
Image Memory
ROM Memory Controller For Port 2
Look-Up Tables
Video Output Circuitry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

FIG. 8

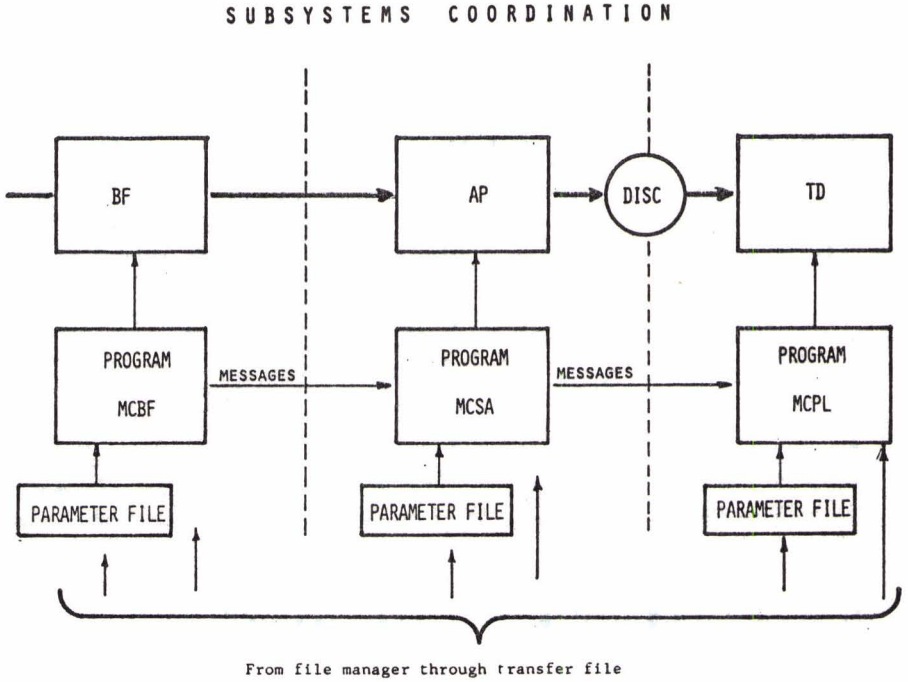
SUBSYSTEMS COORDINATION
BF
AP
DISC
TD
PROGRAM MCBF
MESSAGES
PROGRAM MCSA
MESSAGES
PROGRAM MCPL
PARAMETER FILE
PARAMETER FILE
PARAMETER FILE
From file manager through transfer file

FIG. 9

```
*C720  T=00004 IB ON CROOO11 USING 00009 BLKS R=0049

0001  WP 700       *PLOT WIDTH (700 MAX.)
0002  WI 330       *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. 2000
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