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:-

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

The beamforming problem is then apparent: the required array output

$$\sum_{N} k_n . \; f\,(t)$$

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, addressed 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?

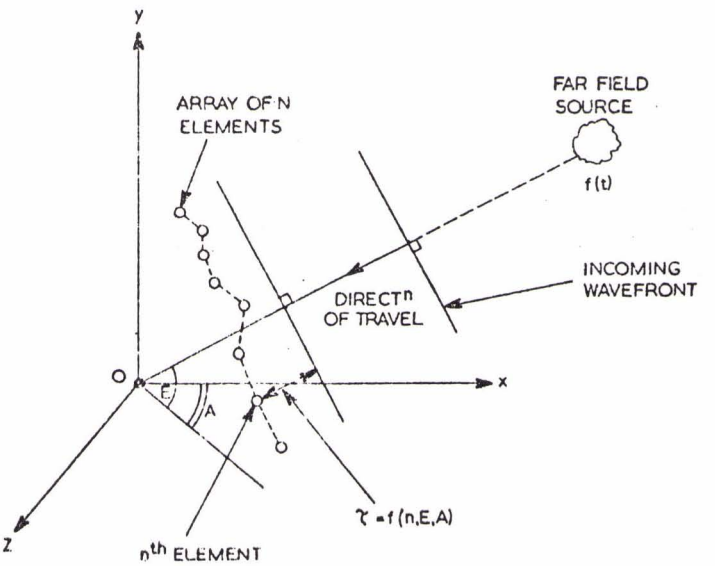
T.E. Curtis The single source situation was initially a worry, but a number of U.S. manufacturers are now in the field. We have taken a deliberate decision to use commercially available LSI components and have not pursued custom-designed devices in the U.K. for this application.

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.



$$\text{NATURAL ARRAY O/P} = \sum_{N} k_n . f\{t + [\{(n\ E\ A)\}\}$$

$$\text{REQUIRED O/P} = \sum_{N} k_n . f(t)$$

FIG. 1 THE BEAMFORMING PROBLEM

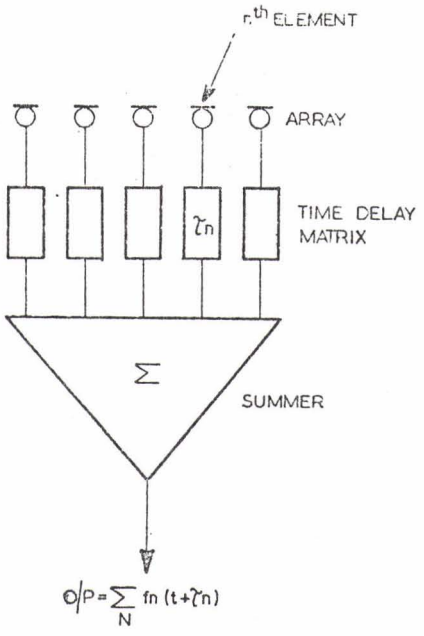


FIG. 2 BLOCK SCHEMATIC OF CONVENTIONAL TIME DELAY BEAMFORMER

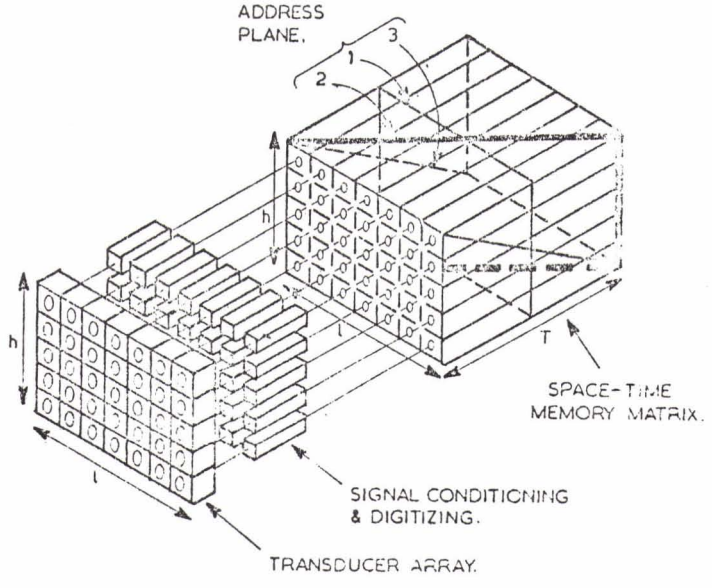


FIG. 3 SPACE TIME MATRIX BEAMFORMER

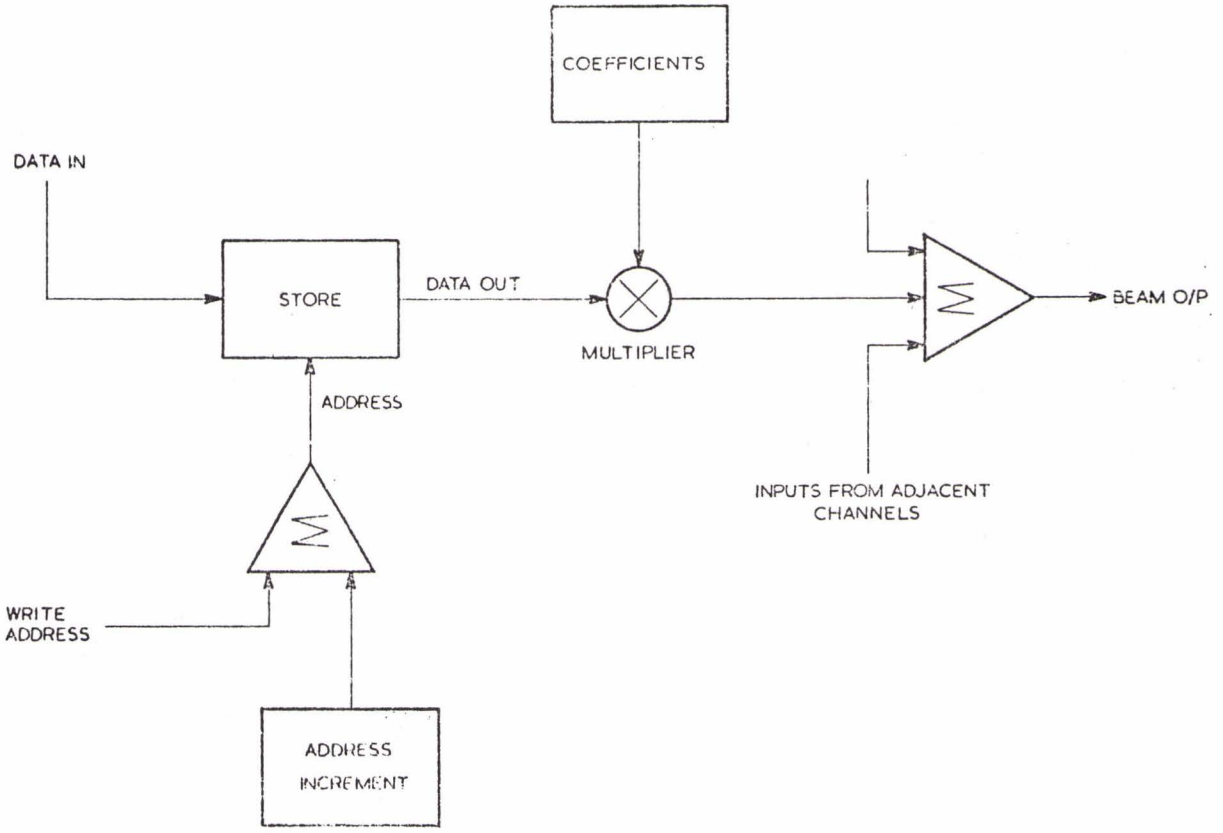


FIG. 4 BEAMFORMER CHANNEL SCHEMATIC

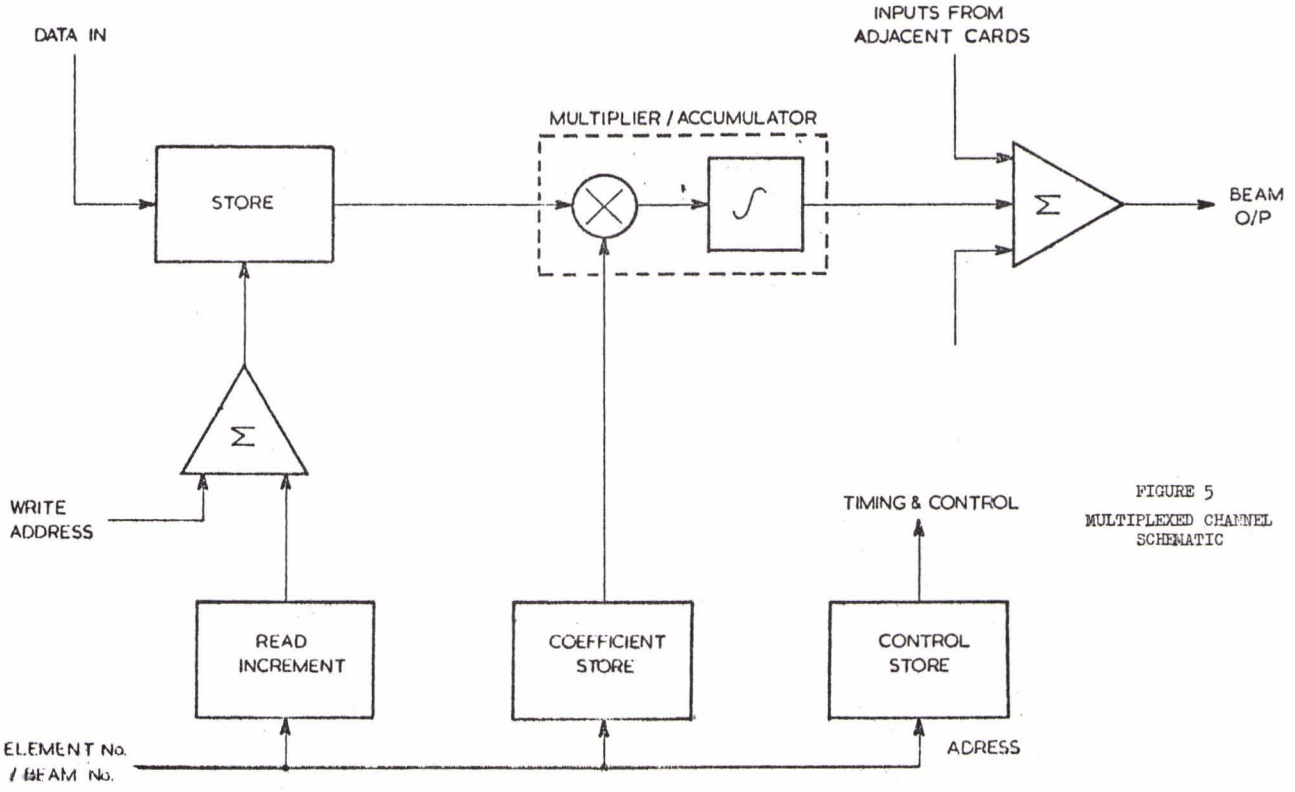


FIGURE 5
MULTIPLEXED CHANNEL SCHEMATIC

FIG. 5 MULTIPLEXED CHANNEL SCHEMATIC

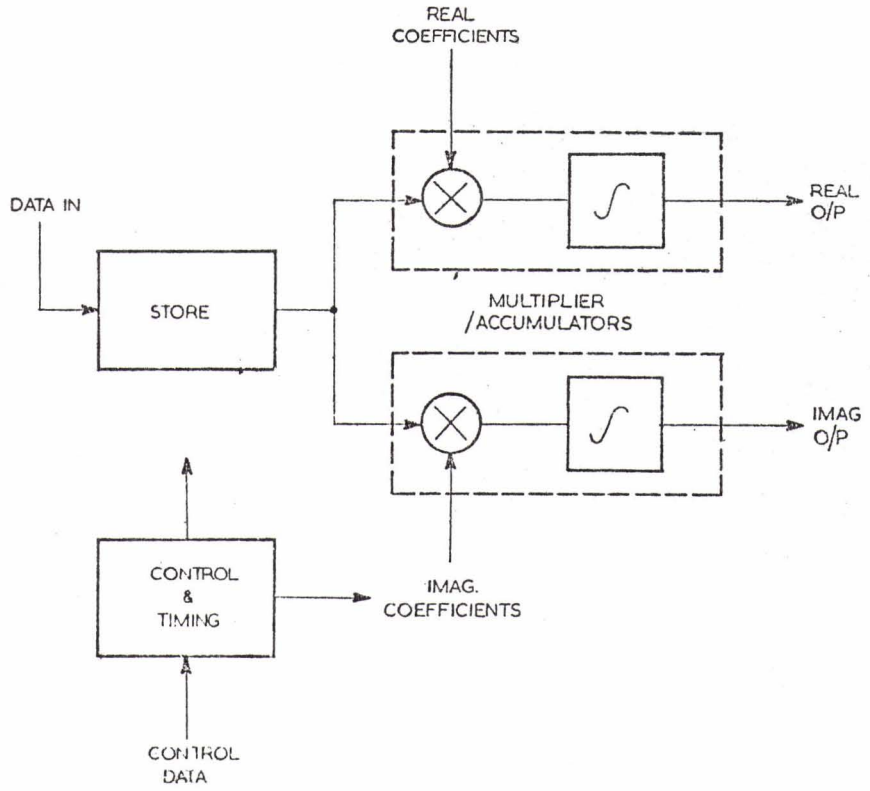


FIG. 6 PROCESSOR SCHEMATIC

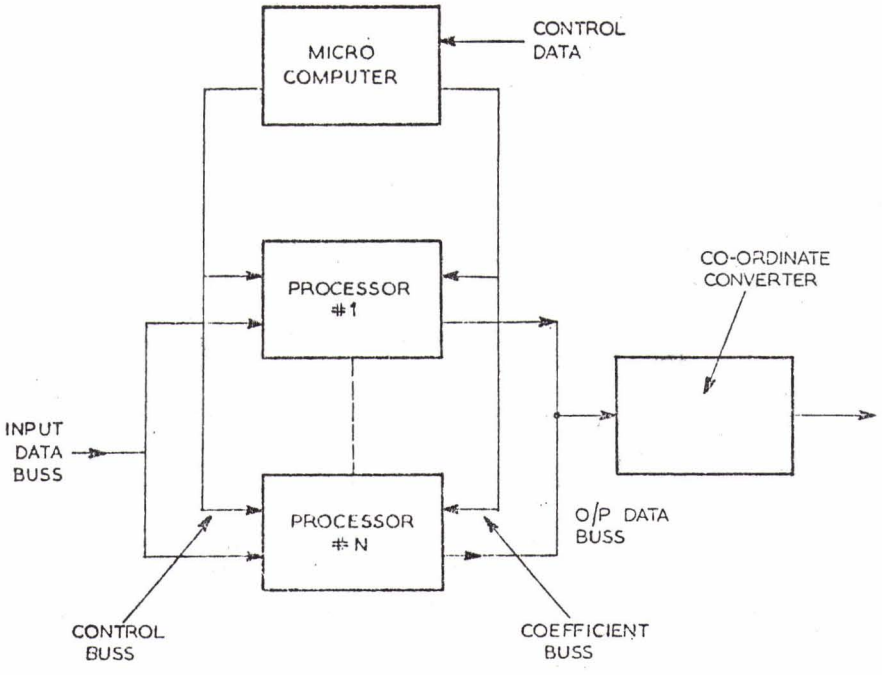


FIG. 7 SYSTEM ARCHITECTURE