A New Architecture for Digital Time Domain Beamforming

WILLIAM ROBERTSON, MEMBER, IEEE, AND DOUGLAS PINCOCK, MEMBER, IEEE

Abstract—This paper introduces a new approach to an architecture for digital time domain beamforming. The approach is unique as it produces sufficient beams to cover the observation space both simultaneously and continuously. Two implementation techniques are discussed; the first is a single chip beamformer based on a multibus architecture; the second is a multichip network-based architecture. The upper constraint on the number of sensors N, handled by a chip or chip set, is imposed by VLSI considerations. In both approaches, an additional chip has been identified which permits a number of systems, each supporting N sensors, to be combined to support larger numbers of sensors. The features of the architecture which permit this extension are the distributed memory and the reconfigurability of the modules.

A two-dimensional FFT beamformer system performs a temporal FFT to separate out the frequencies and then performs a spatial FFT to determine the direction of the source(s) at specific frequencies. The beams produced by the new architecture contain all the signal frequencies which originate along the maximum response axis (MRA) of the beams. Therefore, an FFT may be performed on each beam to determine the frequencies of the signal sources. From this point of view, the new architecture eliminates the need for the temporal FFT required in the two-dimensional FFT beamformer. The processing power replaced by the architecture, assuming a system with 1024 beams, and a maximum signal frequency of 10 kHz, is 820 megafloating point operations per second (MFLOPS).

List of Symbols and Abbreviations

b	The beam factor which determines the beamwidth
B	The number of beams to cover one-half of the observation space
BCM	Beam-distributor clock-period multiplier
BM	Beam-distributor modulus
BS	Basic system
BS-n	Basic system for n sensors, n beams
BU	Beam unit
BUCS	BU control subunit
CSA	Carry save adder
dB	Decibel
DSC	Demultiplexer subclock
DSGC	Demultiplexer segment clock
ET	Even time slot
FB	First beam
FFT	Fast Fourier transform
k_s	The synchronous rate multiplier
LGB	Lower guard band
M	The interpolation factor to obtain the synchronous samples from a signal sampled at the Nyquist rate
N_e	Number of system sensors
N	Number of sensors supported by a BS
OT	Odd time slot
PBSU	Partial beam summer unit
r	Number of data bits in a sample
R	Number of output data lines on a BU
SBU	Single beam unit
SCM	Sensor clock-period multiplier
SDM	Sensor demultiplexer modulus
SE	Switching element
SI	Sensor index
SU	Sensor unit
SUS	SU time slot in a time-multiplexed bus system
UGB	Upper guard band
VLSI	Very large scale integrated circuit

I. An Overview of the System Architecture

A. Introduction

This paper describes a new architecture for the determination of the direction of point sources with respect to a linear array of sensors. This architecture will produce, simultaneously, sufficient beams to cover the observation space. Both active and passive systems [22], [9] require the ability to determine the direction of point sources. The sensors must be sampled with 16-bit resolution for active, and 10-bit resolution for passive, sonar systems. Radar systems [21], [1] with an intermediate frequency (IF) of 0 Hz require samples of 8- to 10-bit resolution.

The rapid advances in technology during the past decade have both consolidated the position of FFT based approaches to the beamforming problem, and have made modern spectral analysis techniques viable. Although the latter have the potential to provide good resolution, the high computational load will lead to complex architectural implementations.

Time domain techniques have, however, been completely overshadowed by the frequency domain techniques for the past decade. Research in the time domain has been restricted to the analysis of the synchronous sampling rate requirement for digital implementation [5]; the processing of complex signals [9], [13], [3], [12]; im-

Manuscript received November 8, 1985; revised March 2, 1987.

W. Robertson is with the School of Computer Science, Technical University of Nova Scotia, Halifax, N.S., Canada B3J 2X4.

D. Pincock is with the Applied Microelectronics Institute, Halifax, N.S., Canada B3J 2X4.

IEEE Log Number 8717264.

plementations for radar [1], [2]; and interpolation to reduce the sampling rate at the sensors [13], [14], [12]. Implementation has consistently dealt with a small number of beams at a time.

Time domain beamforming entails the insertion of time delays into the signal paths of individual sensors in order to align the signals which belong to the same wave front. As indicated in Fig. 1, for the low-pass case, the delay between adjacent sensor signals, for a specific angle of incidence, is easily determined. It follows that a change in the angle of incidence results in a corresponding change in the intersensor delay, and vice versa. The entire observation space may be scanned, by discrete increments in the look angle, if the delays are changed. In a digital system, the sensor sampling rate must be consistent with the highest signal frequency, and must permit the construction of sufficient beams to cover the entire observation space. Having chosen a look angle, the delayed sensor samples are summed as shown in Fig. 2.

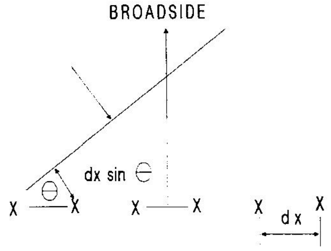


Fig. 1. The line array geometry.

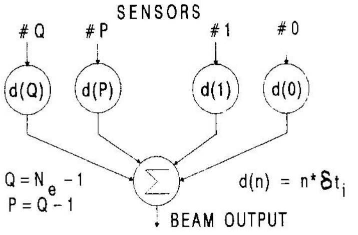


Fig. 2. The delay-sum beamformer.

The basis for the architecture is exploitation of the inherently pipelined nature of the beamforming process; a key feature is the distribution of the storage requirement throughout the architecture. Two viable approaches, each of which uses sensor units (SU's) to control the flow of the sensor data, and beam units (BU's) to form the beam samples, have been identified. A basic system (BS) using SU's and BU's will support a maximum of N sensors and N beams; outputs from each of the system beams are produced at the Nyquist rate.

The discussion which follows is based on a low-pass signal system. The concepts are, however, applicable to bandpass signals, of bandwidth W, centered on a carrier frequency f_o. In the bandpass case, quadrature sampling and interpolation must be assumed; the steering delay, δt, is then governed by a fraction of W, not by f_o. In the simplest implementation, the bandpass signals would be sampled at some δt which is an integer multiple of $1/f_o$ [14].

The array parameters governing the number of beams required to cover the observation space are the number of sensors N, and the attenuation at which adjacent beams intersect each other (the scallop). The new architecture is based on the requirement that no, or very little, physical reconfiguration should be required to change either the scallop, or the number of sensors in the system. An increase in the number of sensors must be achieved by the addition of well-defined subunits. In order to meet these requirements, the system is modular and the modules can be reconfigured by an off-line download of parameters from a host computer. Fig. 3 shows a system configuration to produce B beams for some small number of sensors N. The concept of an extended system to support some large number of sensors, $N_e = U \cdot N$, is shown in Fig. 4. Fig. 4(b) shows the internal structure of the segment unit, of Fig. 4(a), which employs BS's to obtain partial beams which are then combined to form the system beams. A simple example will demonstrate the principle for the low-pass case.

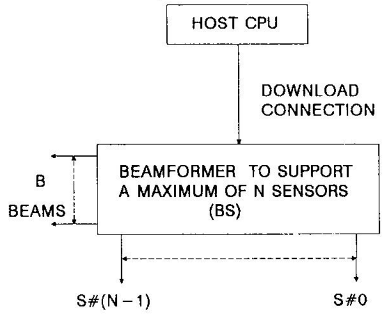
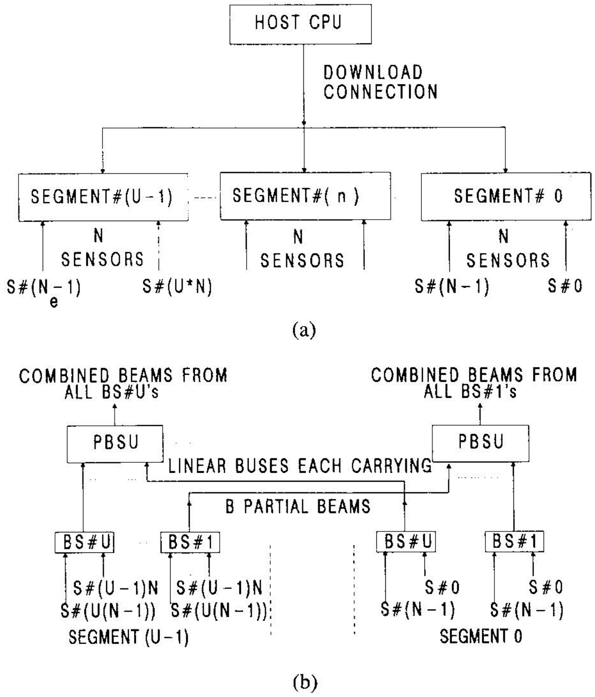


Fig. 3. System configuration to support some small number of sensors N.



(a)

(b)

Fig. 4. Conceptual system configuration for some large number of sensors $N_e = U \cdot N$.

Fig. 5 shows temporal samples from four sensors. The

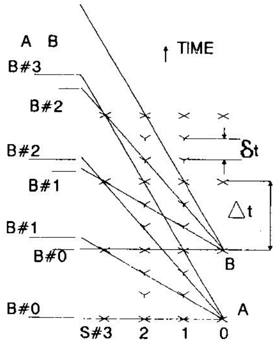


Fig. 5. Alignment of sensor samples into computation lines for a four sensor system.

Nyquist rate samples, at intervals of Δt, are indicated by X's, and other samples at multiples of some smaller interval, δt, are indicated by Y's. Any straight line passing through one sample from each sensor is called a "computation" line and identifies the sensor samples which must be added together to form a beam sample. There are four such lines, starting at sensor zero ($S\#0$), in Fig. 5; the horizontal line represents the broadside beam sample, and each of the others represents off-broadside beam samples for which the intersensor delays are multiples of δt. In Fig. 6, a similar set of lines are shown for a system of eight sensors with identical intersensor spacing and an identical Nyquist interval. In this case, the smallest interval between temporal samples is $\delta t/2$, and there are 7 computation lines starting at $S\#0$ at time zero. On the hypothesis that a basic system (BS-4) capable of producing all the beam samples of Fig. 5 exists, then a number of these systems should be able to cooperate to produce the beam samples of Fig. 6. This is borne out by the following discussion.

It is evident that Fig. 5 is a subset of Fig. 6 as the latter's computation lines 0, 2, 4, and 6 over the portion contributed to by sensors 0, 1, 2, and 3 map directly onto Fig. 5. The BS-4 can, therefore, produce those portions of the 8 sensor system's beam samples directly. If the remaining portions of the computation lines 0, 2, 4, and 6 were to be delayed to coincide with the beginning of a Nyquist period at $S\#4$, they would then map directly on to the computation lines of Fig. 5 and a BS-4 could be used to form those partial beam samples. For Fig. 6's computation lines 1, 3, and 5, it is evident that insertion of delays of $j(\delta t/2)$ at $S\#j$, $j = 0 \cdots 3$, will map the lines onto computation lines 1, 2, and 3 of Fig. 5. The corresponding partial beam samples can then be produced by two BS-4's. The partial beam samples must be aligned prior to summation to produce the beam samples. The new architecture can be configured to implement the required mapping and perform the time alignments [15].

There are two design issues which must be addressed in order to develop an architecture to produce, simultaneously, sufficient beams to cover the observation space. The first of these, addressed in Section I-B, is how to determine, and make provision for, the minimum number of beams required. The second, addressed in Section I-C, is how to establish the flow control required to direct samples from the line array's sensors to the beams.

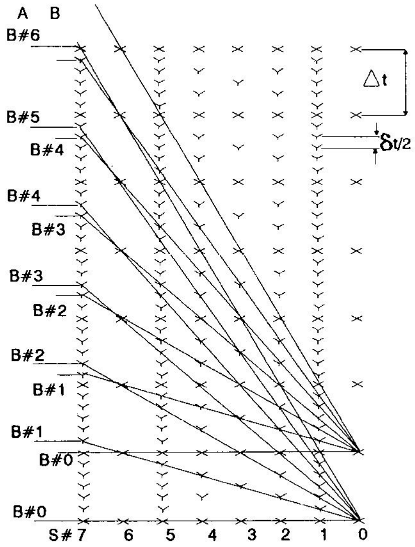


Fig. 6. Alignment of sensor samples into computation lines for an eight sensor system.

B. System Beam and Temporal Sampling Requirement

For a line array of N sensors spaced at intervals of one-half the wavelength of the highest signal frequency, $f_{\max}$, temporal sampling at the Nyquist rate is insufficient to construct the beams required to cover the observation space. This is evident from Fig. 5 as only the Nyquist samples for the broadside beam, and those for the endfire beam, can be connected by a straight line originating at $S\#0$. In order to construct more beams, more sensor samples must be made available either by sampling at a higher frequency or by interpolating between the Nyquist rate samples. The smallest angle from broadside to the first off-broadside beam is governed by the (apparent) temporal sampling rate at the sensors. This is indicated by the samples marked by Y's in Fig. 5 which provide two computation lines in addition to those provided by the Nyquist rate samples. The synchronous sampling rate is the temporal sampling rate which will support the minimum number of beams required to cover the observation space.

For a system with N sensors, the number of beams to cover the observation space may be determined from the beam pattern of the array. It can be shown [18] that

$$k_s \geq 2N/b \quad \text{an integer} \qquad (1.1\text{a})$$

such that

$$B = k_s/2 \quad \text{an integer} \qquad (1.1\text{b})$$

$$M = k_s/2 \quad \text{an integer} \qquad (1.1\text{c})$$

where

k_s is the synchronous frequency multiplier which determines the synchronous sampling frequency;

M is an interpolation factor to obtain samples at intervals of δt_s from sensors sampled at the Nyquist rate;

and

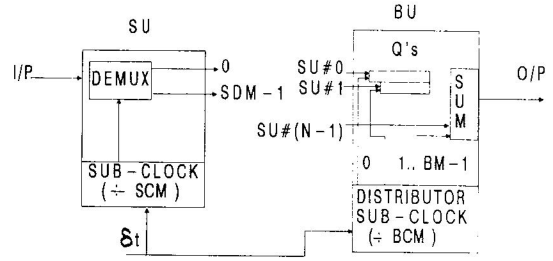
B is the number of (off-broadside) beams to be implemented to cover one-half of the space. The broadside beam will be beam zero ($B\#0$).

Equation (1.1) indicates that the smallest value of k_s should be determined first; k_s should then be increased, if necessary, to obtain an integer number of beams and an integer interpolation factor. The sensor samples presented to the beamforming system are assumed to be at intervals of time $\delta t = \delta t_s$.

C. Data Flow

The required data flow may be envisaged from the computation lines of Fig. 5. At any instant in time, after system startup, the beamforming system must have retained the past history for each beam sample. For example, at the end of the first Nyquist period, the system must have retained sufficient information to construct a sample for the broadside beam ($B\#0$) in addition to retaining those portions of the other computation lines which have already arrived. Therefore, the three sensor samples which lie on the computation line for the first off-broadside beam ($B\#1$), the two which lie on the computation line for the second off-broadside beam ($B\#2$), and the one which lies on the computation line for $B\#3$ must be retained. The volume of information retained by the beamforming system is directly proportional to the interval between successive beam samples; the maximum acceptable interval is the Nyquist interval Δt. In Fig. 5, a second set of computation lines is, correspondingly, initiated at $S\#0$ at the start of the second Nyquist period. By the end of the second Nyquist period, the system must have retained both the contributions to the first set of beam samples, and the contributions to the second set of beam samples. A third set of computation lines will start up at the beginning of the third Nyquist period; by the end of that period, the system must have retained the contributions from three different computation lines for $B\#3$, 2 different ones for $B\#2$, 1 for $B\#1$, and 1 for $B\#0$. Each of these computation lines will have progressed either wholly or partially toward the production of a sample for its own beam. $N - 1$ different sets of computation lines will be established after the $(N - 1)$th Nyquist period.

Consider the case in which there are M temporal samples per Nyquist period. All the samples within a Nyquist period which contribute to the sets of computation lines currently in progress can be stored in $N \cdot M$ memory locations; the individual samples must be mapped to those computation lines to which they contribute. With appropriate addressing techniques and sufficient memory bandwidth, the accumulated value of each computation line could be updated from the memory. The memory requirement grows $O(N^2)$ under such a scheme, and the memory bandwidth, required to access the memory and update all invocations of the computation lines, grows $O(NB)$. Small numbers of sensors and/or a small subset of the required beams could be handled with such a memory based system [12], [11]. The new architecture replaces the need for the single memory unit with its potentially complex addressing requirement by distributing the storage of past history throughout the system modules. This approach is a key factor in limiting the storage in each module to that required by some small number of sensors N, but still permitting larger numbers of sensors to be handled.



(δt REPRESENTS THE SYSTEM CLOCK WITH A PERIOD EQUAL TO THE SYNCHRONOUS PERIOD.)

Fig. 7. Functional schematics of the SU and the BU.

In Fig. 5, the sets of computation lines starting up at intervals of the Nyquist period suggest that the beam samples may be formed by placing demultiplexers in the sensor data paths, and providing appropriate queueing and synchronization mechanisms in the individual beamformers. In the proposed architecture, these functions are performed by the SU's and the BU's. An SU accepts samples at the synchronous rate and directs them, as required, to BU's; there is one BU for each system beam (Fig. 7).

Each SU has a specific interval between useful samples. The interval is determined by a sensor clock-period multiplier (SCM) which is applied to the system clock, δt, to produce the SU subclock. Each output from the subclock steps the demultiplexer to its next output. The number of demultiplexer outputs required at a specific SU is controlled by the sensor demultiplexer modulus (SDM), and each output is associated with a predetermined set of BU's.

A BU receives a sample from each SU in the system and adds it to the tail of a queue; each SU has, therefore, its own queue in each BU. The queues are loaded under control of the BU distributor subclock, and those queues which must be loaded at the same time are connected to the same distributor output. The length of a queue depends upon both the beam and the sensor with which it is associated. In a specific BU, the interval between successive outputs from the distributor is controlled by the beam-distributor clock-period multiplier (BCM) which is applied to the system clock; the number of distinct outputs from the distributor is controlled by the beam-distributor modulus (BM). The heads of the queues are presented to a summer which performs the summation when a sample arrives from the highest numbered SU. The data aligned at the heads of the BU queues are the computation line for the next beam sample [15], [17], [18].

Fig. 8 demonstrates the principle of operation for a fully connected four sensor system with a 2 dB scallop for which the computation lines of Fig. 5 apply. The flow

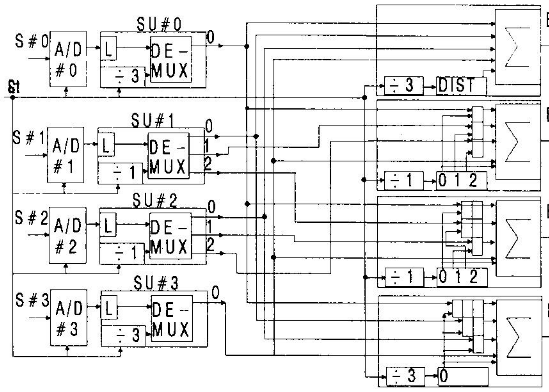


Fig. 8. Fully connected system of four sensors.

TABLE I
(a) SU Flow Control Parameters for a 4 Sensor System

M = 3; B = 3; k_s = 6;				BU'S CONNECTED TO THE DEMULTIPLEXER OUTPUTS.		
SI	SENSORS	SDM	SCM	#0	#1	#2
0	0,3	1	3	0,1,2,3		
1	1	3	1	0,3	1	2
2	2	3	1	0,3	2	1

(b) BU Flow Control Parameters for a 4 Sensor System

	QUEUE-			Q'S CLOCKED BY THE DISTRIBUTOR OUTPUTS		
BU	LENGTHS	BM	BCM	#0	#1	#2
0	0,0,0,0	1	3	0,1,2,3		
1	1,1,1,0	3	1	0,3	1	2
2	2,2,1,0	3	1	0,3	2	1
3	3,2,1,0	1	3	0,1,2,3		

TABLE II
(a) SU Flow Control Parameters for an 8 Sensor System

M = 6; B = 6; k_s = 12;				BU'S CONNECTED TO THE DEMULTIPLEXER OUTPUTS.					
SI	SENSORS	SDM	SCM	#0	#1	#2	#3	#4	#5
0	0,6	1	6	ALL					
1	1,7	6	1	0,6	1	2	3	4	5
	5	6	1	0,6	5	4	3	2	1
2	2	3	2	0,3,6	1,4	2,5			
	4	3	2	0,3,6	2,5	1,4			
3	3	2	3	0,2 4,6	1,3,5				

(b) BU Flow Control Parameters for an 8 Sensor System

	QUEUE-			BU'S CONNECTED TO THE DEMULTIPLEXER OUTPUTS					
BU	LENGTHS	BM	BCM	#0	#1	#2	#3	#4	#5
0	ALL ZERO	1	3	ALL					
1	2,1,1,1, 1,1,1,0	6	1	0,6	1,7	2	3	4	5
2	3,2,2,2, 1,1,1,0	3 3	2 2	0,3,6	1,4,7	2,5			
3	4,3,3,2, 2,1,1,0	2	3	0,2 4,6	1,3, 5,7				
4	5,4,4,3, 2,2,1,0	3	2	0,3,6	2,5	1,4,7			
5	6,5,5,4, 3,2,1,0	6	1	0,6	5	4	3	2	1,7
6	7,6,5,4, 3,2,1,0	1	6	ALL					

control parameters are given in Table I. All SU's with the same control parameters are grouped together and identified by a sensor index (SI) which is equal to the lowest numbered sensor in the group. As there is normally one queue per sensor in a BU, the queue lengths are listed in the order of queues 0 through $N - 1$. The BU's to which a demultiplexer output is connected are listed below the appropriate output; similarly the queues which are clocked simultaneously are listed below the appropriate distributor output.

From the above discussion, it is apparent that the six flow control parameters must be determined after k_s, B, and M have been calculated. It is established in Appendix A that these six parameters are dependent on the required number of beams B and the interpolation factor M. The system parameters which result from the application of (1.1) and (A.1)–(A.5) for the four sensor system with a 2 dB beam scallop, discussed above, are those shown in Table I. Table II shows the flow control parameters determined for the eight sensor system to which the set of computation lines of Fig. 6 applies.

D. Architectural Issues

The fully connected model of Fig. 8 served to demonstrate the flow control principles required. However, a practical system will usually require a large number of sensors and, as each BU must be connected to all SU's, the number of input ports to a BU becomes untenable for a large number of sensors. For example, a 64 sensor system would require 64 ports on each BU; each port would be between 8 and 16 bits wide. In addition, a fully connected system is inflexible due to the hard-wired SU to BU connections, and reconfiguration to support different numbers of sensors and different scallops is difficult.

For large N, there is a high volume of information to be transferred between the SU's and the BU's of a BS during each Nyquist period. In the worst case, there may be N SU's and N BU's in the system requiring N^2 information transfers to take place in the Nyquist interval. This is a major impediment to the provision of a flexible architecture. The proposed implementations, discussed in Section II, ensure that the transfers take place in a timely manner and that reconfiguration for different numbers of sensors and scallops is possible.

II. The Implementation Approaches

A. A Bus Approach

As the information to be transferred is deterministic, in accordance with the relationships of Section I, the present application is rather different from the more general computer system application in which a statistical model is required for the communication channel usage. Therefore, a linear bus system can be used for the communication channel.

1) The Synchronous Bit-Parallel Single Bus Approach: As each BU has its own mechanism for determining when a sample is expected from any specific SU, each SU may be allocated its own time slot on the bus (SUS), and the onus is placed upon the BU to read the bus at the appropriate time. This synchronous system is shown in Fig. 9, where the bus width is $r + 1$ bits. r is

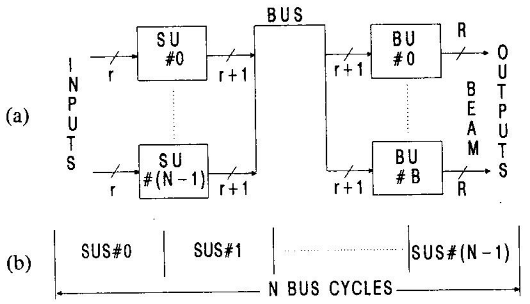


Fig. 9. Synchronous bit parallel single bus system.

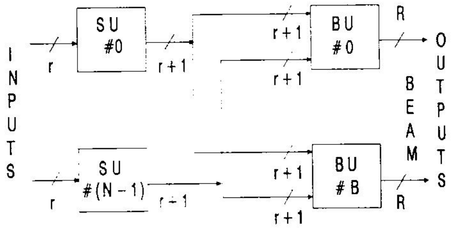


Fig. 10. Schematic of the synchronous multiple bus system.

the sample width, and the additional bit is a valid data bit, placed on the bus by an SU with an active demultiplexer output, to permit the BU to check synchronization. $R = r + \log_2 N$ is the number of bits required for the beam output. If individual SU's and BU's were to be implemented as separate VLSI chips, for which the number of pins is projected to be between 120 [20] and 132 [8], a unit size of 1024 would be easily attainable.

It is evident that each SU can only place information on the bus every N bus cycles, therefore, the minimum synchronous period is N bus cycles. For a fixed bus cycle time, a programmable number of bus time slots will provide a system for which the maximum signal frequency is inversely proportional to the square of the number of sensors. For example, assuming that a bus cycle time of 100 ns were implemented and that $b = 1$ in (1.1a), a system of 1024 sensors could support signal frequencies to 4.768 Hz and a system of 32 sensors signal frequencies to 4.883 kHz.

The single bus system provides a simple interconnection and may be a solution for small numbers of sensors at intermediate frequencies, or for larger numbers of sensors at very low frequencies.

2) The Synchronous Multiple Bus System: An alternative technique (Fig. 10) is to allocate one bus per SU and to provide N bus ports on each BU. This has the potential to provide a δt equal to one bus cycle. The number of signal lines required by a BU is 90 (70, 50) for an $N = 4$ and 159 (123, 87) for an $N = 8$ with an r of 16 (12, 8), assuming serial download of configuration parameters from the host. For low to intermediate frequencies, the buses may be replaced with serial data links, in which case an N of 32 with an r of 16 (12, 8) and a bit rate of 100 ns on the serial links could accommodate signal frequencies to 9.75 (13.00, 19.50) kHz at a scallop of 1 dB.

a) The Single Chip Beamformer: For high-frequency applications with small numbers of sensors, the synchronous multiple bus system may be used as the basis of a single chip system. Each of the beams cannot be allocated its own signal lines as the pin-out would exceed 132. However, as samples for each beam need only be produced at the Nyquist rate, the beam outputs may be time multiplexed onto two sets of output lines. In this manner, it is anticipated that a maximum r of 12 may be accommodated on a single chip beamformer for an N of 8. The number of ticks required to transfer the beam output samples from the chip will be $(B/2) + 1$. For example, a scallop of 2 dB yields $B = M = 6$, from (1.1), for an N of 8; four of these ticks would be required to transfer the six beams and the broadside beam from the chip. At a scallop of 2 dB ($b = 1.446$), signal frequencies in excess of 900 kHz could be accommodated by such a beamformer chip clocked at 100 ns. A system with 64 sensors, capable of handling frequencies to 112 kHz, could be constructed using a number of those single chip beamformers. As the highest signal frequency which can be handled is inversely proportional to the clock period, it is anticipated that a VLSI implementation will perform at higher frequencies.

For any given clock period, larger numbers of sensors require greater numbers of units and result in a proportionally lower frequency capability. Table III shows estimated signal frequency capability, using the equality in (1.1a), for various clock periods, and various numbers of sensors, at a scallop of 2 dB, using a single chip beamformer with an N of 8 as a building block.

B. The Network Approach

A communication network can be used to interconnect a number of processing elements (PE's) to each other, to other PE's, or to shared resources. For example, a 2 × 2 crossbar switch is a network which can connect either of the two PE's on its input ports to either of the resources on its output ports. The nature of the beamforming problem is such that network configurations in which the switching elements (SE's) support packet switching, and internal buffering, are the only ones of immediate interest [10], [19], [20], [23], [24], [4].

Fig. 11 shows the network devised for the beamforming application. The SE's are four-function switches capable of straight-through connection, exchange of upper (lower) input to lower (upper) output, broadcast of lower input to both outputs, and broadcast of upper input to both outputs. The network is based on an adaptation of the shuffle exchange which will be called a progressive shuffle connection. Bank#0 is the leftmost column of SE's in the network and the progressive shuffle proceeds to Bank#1, Bank#2, etc. This topology provides a simple interconnection scheme and permits all input ports to access any output port.

Due to the importance of the chronological order of the samples, the SE's must resolve routing conflicts in such a way as to retain, from one Nyquist period to the next, the chronological order of the samples passing through

TABLE III
SIGNAL FREQUENCY CAPABILITY (kHz)

δt(ns)	50	100	200	400
NUMBER OF SENSORS				
8	1807.5	903.75	451.88	225.94
32	451.88	225.94	112.97	56.48
64	225.94	112.97	56.48	28.24
512	28.24	14.12	7.06	3.53
1024	14.12	7.06	3.53	1.77

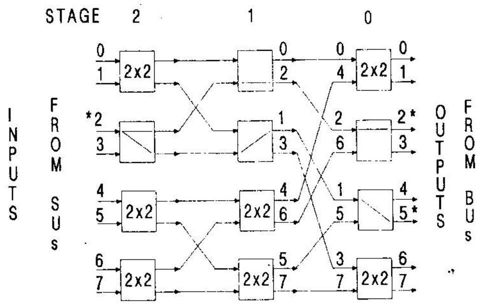


Fig. 11. The progressive shuffle network.

them. In addition, each SE must have sufficient internal storage capacity to ensure that the packets are not backed up. Therefore, an SE port must always be able to accept a packet offered to it by a previous port. If this were not the case, a full buffer in an SE would upset the chronological order of the routing by blocking packets attempting to enter its input port; incorrectly blocked packets, being unable to regain their position, would hold up other packets attempting to pass through the same port.

1) The Beamformer Network Routing Method: The major drawback of network routing schemes [10], [20] is the fact that the Hamming distance determines the number of output ports which can be reached when a distribution tag is used; this relationship between the accessible ports is too restrictive for the beamformer. For example, consider a one-to-one mapping of the SU's of Table II to the input ports of Fig. 11, and a one-to-one mapping of the BU's to the output ports. When demultiplexer output #2 of SU#2 in Table II is activated, the SU sample is destined for both BU#2 and BU#5. Using Siegel's routing technique the standard tag can be either 000_2 or 111_2, the distribution tag is 111_2, the Hamming distance is 3, and the sample will arrive at all of the BU's.

The routing scheme devised for the beamformer uses the destination port number as the standard tag. The distribution tag is built along the paths to the required destinations as the inclusive OR of the Stage#th bit of the destinations required, and accessible, from a specific SE. A standard tag bit of 0 routes a packet to the SE's upper output port, and a standard tag bit of 1 to the SE's lower output port. A distribution tag bit of 1 will override the standard tag and cause the packet to be routed to both output ports. For example, transmission of information from port 2 on the input side of the network to both ports 2 and 5 on the output side of the network will require a standard tag of 010_2 to get to output port 2, a standard tag of 101_2 to get to output port 5, and a distribution tag of 100_2 along the path to port 2 and along the path to port 5. The information travels along the paths indicated in Fig. 11. For the beamformer application, this routing would represent the destinations of SU#2 in a specific time slot in the Nyquist period; the standard and distribution tag bits, for each time slot, must be placed in tag memories in the SE's. The tag memories must be addressed by the source SU number and the time-slot number, therefore, these parameters must be able to be extracted from the packet itself.

In the worst case, an SU connected to an input port of the network may transmit a packet into the network on each tick of the system clock. As each packet must be routed according to both its SU of origin and its time slot of origin, each SE must be able to route up to $N \cdot M_{\max}$ packets according to each packet's individual requirement. In the limit, this requires N^2 locations each for the standard and distribution tags within each SE; a total, for example, of 8192 bits of memory for a 64 sensor system. However, it is apparent from Fig. 11 that only ports 0 to 3 on the input side of the network can access the upper input ports of the last bank. Similarily, only ports 4 to 7 can access the lower input ports of the last bank. It follows that, with knowledge of the input port of origin and appropriate address decoding, only $N^2/2$ locations each for the standard and distribution tags are required. This saving requires transmission of the port of origin, in addition to the SU of origin and the time slot of origin. Therefore, the port of origin and the SU of origin must be combined in the packet to reduce the pin-out requirement of the SE. The simplest method of combining them is by a one-to-one mapping between them. This method is perfectly acceptable if the network is never to be partitioned; it also has the advantage of simplicity of implementation. If, on the other hand, the network is to be useful for a number, x, of powers of two arrays of sensors each of length N_e, such that $x \cdot N_e = N$, then a more complicated mapping is required. The second type of mapping was discussed by Robertson [17]. Fig. 12 shows an example of an $N = 8$ network mapped for two groups of 4 sensors. The packet routes shown are from the top sensor in each group to all beams in the group.

2) Determination of the Routing Tags: The tag memory in each SE must be programmed with the appropriate standard and distribution tags for the number of sensors in the array(s) being processed. This requires that the path(s) through the network be determined in each time slot for each set of destination beams associated with each source sensor.

A recursive algorithm was formulated to determine the tags [18]. The algorithm tracks the accessibility of all SE ports in the network from all possible source ports in Bank#0.

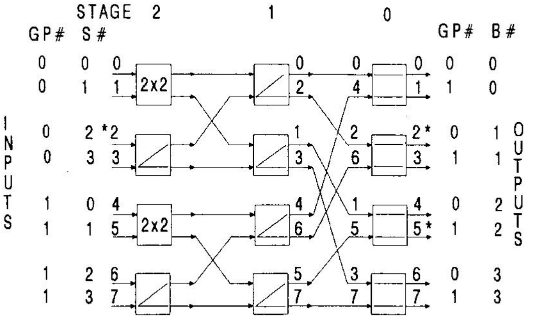


Fig. 12. Network routing for $N = 8$ for two groups.

The standard and distribution tags must be downloaded to the SE's as part of the system configuration procedure.

3) The Switching Element (SE) Bandwidth Requirement: The bandwidth requirement of the network can be determined from a simple analysis of the volume of information which must pass through an SE in one Nyquist period. Samples must first traverse the network before they can be consumed by BU's. In order to retain the chronological order of the outputs from the broadside beam, the samples in a zeroth time slot must reach BU#0 before the samples from the subsequent zeroth time slot. It follows that the transit time of the network is itself irrelevant, except for its effect on the pipeline delay; however, the data for one Nyquist period must arrive at the BU's before the data from the subsequent Nyquist period. The actual order of arrival within the Nyquist period is not critical although precedence must be determined, in part, by the time of origin of a datum. The number of packets entering a BU over a Nyquist period is used to determine the capacity of the SE's ports.

The highest potential for congestion is in those SE's connected to the broadside and end-fire beams. This follows from the fact that each of those beams requires the zeroth time-slot sample from every sensor. As all of those samples enter the network at the same time, they have, given correct precedence, the potential to arrive at the final SE's in the same, or consecutive, ticks. In order to balance the consumption of packets from the output ports with the potentially high rate of arrival of packets at those ports, the following relationships must hold:

$$V_o = C \cdot n \text{ packets}/\delta t \qquad (2.1\text{a})$$

$$V_i = N \text{ packets}/\Delta t \qquad (2.1\text{b})$$

$$V_o = V_i/M \text{ packets}/\delta t \qquad (2.1\text{c})$$

where

V_o is the number of packets leaving the output port to the BU per tick of the system clock,

C is the capacity of a link, in packets per tick, from the port to the BU,

n is the number of links from the port to the BU, and

V_i is the number of packets arriving at the port in an interval equal to the Nyquist period.

As there are M ticks per Nyquist period, it follows that n, the number of links, will be dependent on the smallest value of M. To accommodate scallops such that $1 \leq b \leq 2$ for which the smallest M approaches $N/2$

$$n = 2/C \text{ links.}$$

Therefore, if $C = 1$ packet/δt, each port requires two links to ensure that the information can pass from the SU's to the BU's. As all SE's must be identical, this means that every SE in Fig. 11 must have two links at each port. (A value of $C = 2$ may be considered as an alternative for implementation purposes.)

In order to balance the flow within the SE, such that one-half of the volume of information entering/leaving a port is carried by each of its links, the Bank#0 input ports, which require only one link to be operative, must be assigned in a particular way. Designating the two links as upper and lower, respectively, packets entering the even numbered Bank#0 ports must be connected to an upper link, and packets entering the odd numbered ports must be connected to a lower link. As long as a packet which enters the network on an upper (lower) link traverses the network on an upper (lower) link, the overall capacity will be sufficient. This is easily arranged if the system is restricted to numbers of sensors which are powers of two. Fig. 13 shows a functional schematic of the SE.

It is evident that each SE in the final bank must pass $N/2$ samples on both the upper and lower output links of each port connected to a BU. Therefore, the buffer capacity required in an SE can be determined from the worst case congestion at BU#0. For example, for an $N = 8$ ($M = 6$), the upper links of the even numbered input ports of Fig. 11 will be connected to the even-numbered SU's. By tracing the routing to the final bank's input ports, it is evident that SU#0 and SU#2 will transmit to port 0's upper link, and SU#4 and SU#6 to port 1's upper link. From columns BU1 and BU0 of Table II, it is seen that one packet from SU#0 and two packets from SU#2 will arrive at port 0, and one packet from SU#6 and two packets from SU#4 will arrive at port 1. In each case, one packet must go to both BU#0 and BU#1, and one-half of the remainder to each of BU#0 and BU#1. If all of these packets were to arrive within 3 ticks, this would represent the largest congestion. If, in addition, packets with the same destination were to arrive in the same tick, this will require the largest queuing capacity in the SE. In the cited sample, a queue length of two is necessary. In the general case, the minimum queue capacity required is $N/4$.

4) The Arbitration: An arbiter is required to resolve contentions by passing the packet which has chronological precedence. In the case of all samples originating at time-slot zero, and congregating at the last bank, chronological order alone is clearly insufficient to resolve a conflict. In such cases precedence is given to the packet which originated at the lower numbered SU. This routing decision will permit the arrival of the top SU's packet to indicate that a new computation line is available.

The above routing decisions are adequate when the information in a Nyquist period is considered in isolation; however, at the boundary between successive Nyquist pe-

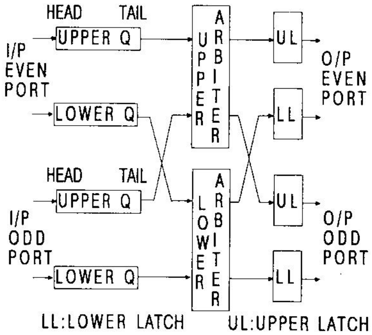


Fig. 13. Functional schematic of a switching element.

riods, precedence must be given to the packets from the earlier period. Preservation of the correct chronological order cannot be guaranteed at the boundary, but, on the other hand, loss of information due to overrun can be avoided. Packets entering the even-numbered and odd-numbered ports contain the time slot in which their data originated and the port of origin in Bank#0; the SU of origin can be obtained from the port of origin. An algorithm, based on time slot and SU of origin, is implemented in the arbiter to promote the packet which is most likely to be chronologically correct under the assumption that both of the packets in contention belong to the same computation line.

As discussed in the preceding sections, the network requires a packet of $r + 1 + 2 \log_2 N$ bits. Each SE requires that each input and output port have a link capacity of two packets per tick of the system clock. Provision of this capacity by duplication of the ports would require in excess of 232 (200, 168) pins for a unit size of 64 and an r of 16 (12, 8). Time multiplexing two packets per port onto a single link would reduce this to 116 (100, 84) pins at the expense of the multiplexing and demultiplexing circuitry. Under the latter scheme each transfer must be completed in one-half the minimum tick period required by the internal SE circuitry. The internal complexity of the arbiter dictates the SE's minimum clock period.

5) Internal Pipelining of the SE: The maximum signal frequency which can be supported in a network based system will depend upon the degree of internal pipelining incorporated in the SE's. The proposed structure is shown in Figs. 14 and 15.

Fig. 14 shows the configuration of the upper link signal path. The lower link signal path must be identical in both form and principle of operation. The arbitration problem is simplified by the adoption of two queues, one for packets with an even-numbered destination port, and one for packets with an odd-numbered destination port. The arbiter requires:

i) the state of the queues to be arbitrated,
ii) the source SU of the packet to be arbitrated, and
iii) the time slot at which the packet entered the network.

The information/control to be supplied by the arbiter is:

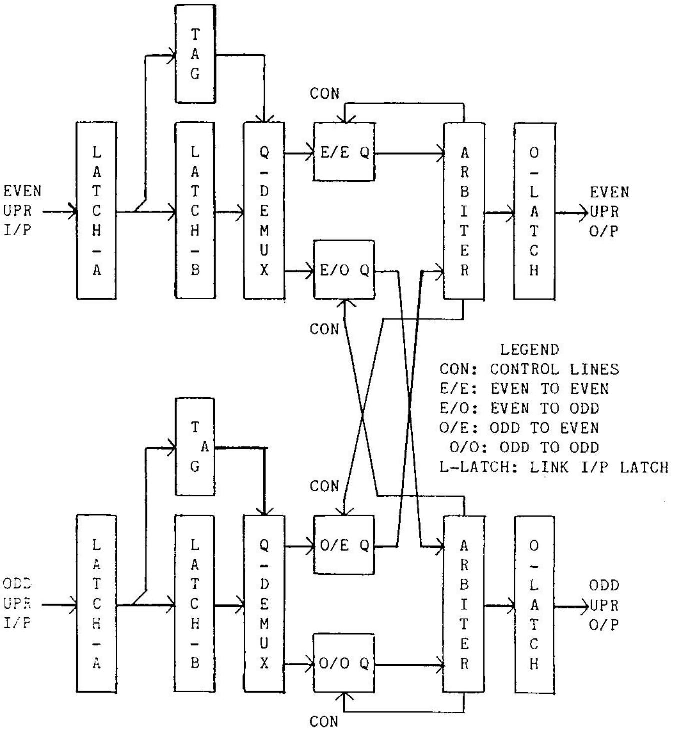


Fig. 14. Pipelined switching element (upper link).

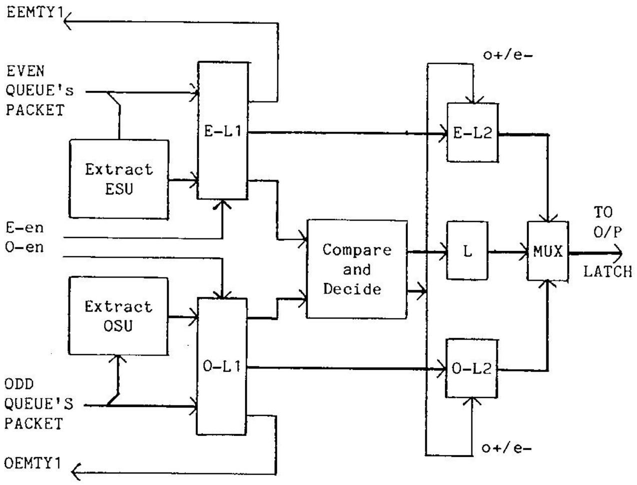


Fig. 15. Internal arbiter pipeline.

i) control signals upon which the queues can determine what action to take, and
ii) the packet routed to the output latch.

The arbiter itself may be pipelined, as shown in Fig. 15, to reduce the effect of the required comparison and decision logic. There are only two conditions under which a queue may update the first latch, $E(O) - L1$.

The first occasion arises when a queue is the winner of the arbitration. In this case, the contents of that queue's first latch must be promoted to its second latch, $E(O) - L2$. The loser of the arbitration must retain the contents of its first latch. $o+/e-$ is asserted high for an odd winner, and low for an even winner.

The second case arises if the input latch is empty, indicated by an asserted even empty (EEMTY) or odd empty (OEMTY); in this case, the queue may promote its head to the first latch. A queue which is awarded an opportunity to promote a packet to the arbiter must always pass along its status, regardless of whether or not it has a packet to pass.

The compare and decide logic is a concurrent implementation of the algorithm to promote the most likely packet to preserve chronological order.

III. Sensor and Beam Unit Configurations

A. *The Sensor Unit (SU)*

An SU (Fig. 16) has three pipe segments. A memory is used to provide the SU subclock counting, to control the demultiplexer modulus, and to hold the routing information. The memory address counter is incremented at intervals of δt, thus acting as the SU subclock by setting up the next state's destination beams, the demultiplexer output active (DEM-ACT) flag, and forcing a recycle to address zero when the demultiplexer modulus is up (MOD). Table II may be used to determine the contents of the memory for each SU in an eight sensor system.

The SU's internal pipeline consists of the demultiplexer input latch ($L1$), the demultiplexer output latch ($L2$), and the communication interface. The BU destinations, from the memory, are combined with the output of $L1$ before latching to $L2$. The DEM-ACT from the memory synchronizes the transfer of information through the pipeline and the active output informs the communication interface that there is a valid sample at its input.

Initialization of the system pipeline is extremely important to ensure that the alignment of the time slots is maintained throughout the pipe. The initial settings of the subclocks and the queue pointers depend upon the number of segments in the pipeline.

B. *The Beam Unit (BU)*

The BU requires different configurations depending upon whether it is intended for a fully connected or synchronous bus system or the network based system. The difference is caused by the necessity, in the network case, to detect the arrival of a packet from the highest numbered system sensor as an indication that a computation line is ready at the heads of the queues. Apart from that detail, the principle of operation of the BU remains identical in both cases. The following discussion of the functional requirements uses the synchronous bus configuration as a model.

The BU has four pipe segments as shown in Fig. 17. In a manner similar to the SU, a memory is used to control the operation of the BU. The memory address counter is incremented at intervals of δt, thus setting up the next state's queue load pulses, the computation line ready (OP-ACT) flag, and tracking the distributor modulus (MOD). The Q-control circuitry in Fig. 17 updates the tail of its queue and checks the synchronization by checking the state of the activity bit for its associated sensor. Each time a queue gets a load pulse there must be a valid sample on the lines from the communication interface. Table II can be used to determine the contents of the control memory for each BU in an eight sensor system. The queue output latches are grouped according to odd- and even-numbered queues to facilitate summation of the computation lines in a CSA summing unit. Fig. 18 shows this interconnection.

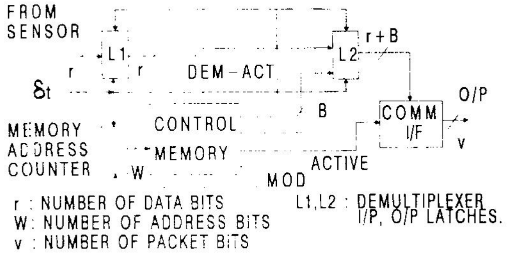


Fig. 16. Functional diagram of an SU.

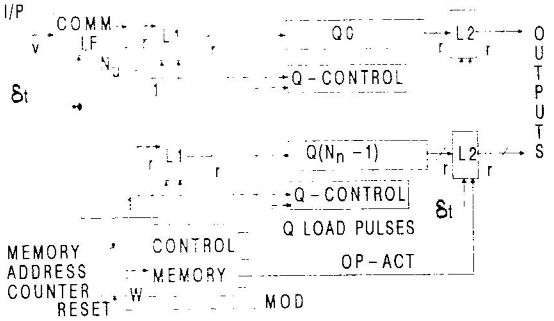


Fig. 17. Functional diagram of a BU.

IV. Accommodating Larger Numbers of Sensors

A. *Summation of the Partial Beam Outputs*

The partial beam outputs produced by each of the units in each of the segments (Fig. 1) must be combined to form the system beams [16], [17]. For the kth beam, the delay between the top samples of consecutive partial beams is $k \cdot \delta t \cdot N$. However, in the network case, the nature of the routing algorithms is such that this relationship is not guaranteed to hold due to the different conflicts which will arise in like-numbered units in the different segments. Therefore, in that case, the alignment delays across segment boundaries must be determined by simulation and the queue initializations downloaded from a host.

1) The Partial Beam Summer Unit (PBSU): For a system with a maximum of $N_e = 1024$ sensors, the number of partial beam samples contributing to each system beam sample would be 128(32) for an N of 8(32). The partial beam samples must be present at the outputs of their respective BS's for a Nyquist period. This may be achieved by the addition of output buffering for each BU in a BS. This buffering must be incorporated in the BU's in a network based system, and in the PBSU's of a single chip beamformer system. A PBSU will then have one Nyquist period to accumulate the system beam samples from the partial beam samples provided by the set of like-numbered BS's with which it is associated.

As the outputs of the different numbered BS's in a segment are destined for different PBSU's, this suggests that

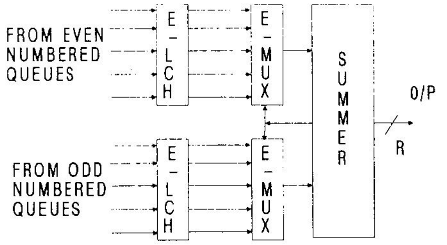


Fig. 18. Functional schematic of the queue to summer interconnection.

TABLE IV
PBSU Pin Requirement

U_{max} = 1024/N; BUS WIDTH=r+LOG$_2$N
TOTAL PINS=(N+1)*BUS WIDTH + LOG$_2$$U_{max}$

r	N	U_{max}	BUS WIDTH	TOTAL PINS
8	64	16	14	914
	32	32	13	432
	8	128	11	106
12	64	16	18	1174
	32	32	17	566
	8	128	15	142
16	64	16	22	1434
	32	32	21	698
	8	128	19	178

a linear bus structure would be adequate to effect the transfer from the BS's to the PBSU's. The Nyquist period consists of U*unit-M ticks, where U is N_e/N and unit-M is the interpolation factor of the BS [16], [17]. Therefore, a PBSU has ample time to time multiplex the U partial beam segments required for each of its system beams. A single chip PBSU would require the bus widths and pin counts shown in Table IV for various values of N, and r, for a maximum sensor requirement of 1024 sensors.

Where the pin counts are prohibitive, the PSBU can be partitioned into single beam units (SBU's) with one SBU for each beam to be formed. The outputs from like-numbered BU's in a set of like-numbered BS's contribute to the same system beam sample and must be time multiplexed on the same bus to the SBU (Fig. 19). The function of the SBU is very similar to that of the BU in that the sample arriving from the BS's must be queued to align the partial computation lines and the aligned samples must be summed to produce the final beam output sample. The maximum number of SBU's in a PBSU is N.

V. Conclusion

Two possible approaches to the simultaneous production of sufficient beams to cover an observation space have been discussed. The two basic units, the sensor unit and the beam unit, are reconfigurable and can be used as the basis for an BS which, in turn, can be used as a building block to produce a beamforming system for some larger number of sensors $N_e \geq N$.

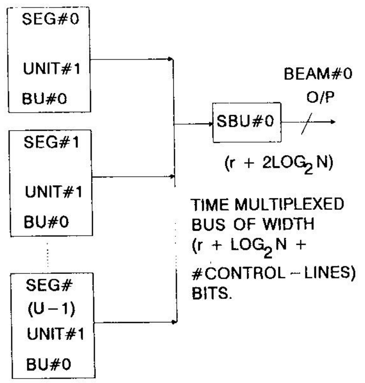


Fig. 19. Block schematic of a single SBU and its associated BU's.

TABLE V
Gate Estimates for the BU

		STATIC QUEUE		DYNAMIC QUEUE	
r	N	TOTAL GATES	Q RAM GATES	TOTAL GATES	Q RAM GATES
8	8	4099	1024	3246	171
	16	8820	4096	5406	682
	32	24387	16384	10741	2728
	64	80182	65536	25558	10912
12	8	5447	1536	4167	256
	16	11992	6144	6871	1023
	32	34289	24576	13805	4092
	64	115802	98304	33866	16368
16	8	6795	2048	5088	341
	16	15164	8192	8336	1364
	32	44181	32768	16869	5456
	64	151422	131072	42174	21824

The BU configuration has the largest number of on-chip functions and has, therefore, the highest device requirement. Therefore, the value of N which can be supported, for both a synchronous bus and a network based system, can be estimated from the BU device requirement. Table V shows the total equivalent NAND gate estimates for a BU for various combinations of r and N. These gate counts indicate that a VLSI implementation for an N of 32 in a network based system should be possible using present technology. The single chip beamformer discussed in Section II-A-2-a would require eight BU's on the single chip yielding values of 43 576 gates for a static queue implementation, and 33 338 gates for dynamic queue implementation, respectively, for an N of 8 and an r of 12. In addition, eight SU's and eight buses would be required on the single chip.

A major consideration is the number of chips required in a system to support N_e sensors, once N has been chosen for the BS. Table VI shows the estimated chip counts to cover one-half of the observation space for various N_e using a single chip beamformer with an N of 8, and a network based beamformer with an N of 32, respectively, as a building block. From this viewpoint, the single chip beamformer has an advantage over the network based beamformer. The authors are of the opinion that the former is worthy of further work aimed at a CMOS VLSI implementation.

Appendix A

For any sensor, $i > 0$, the time increment between the successive computation lines of the set originating at time

TABLE VI
Total Number of Chips in a System

ASSUMPTIONS: U=N /N; U BS'S; 154 CHIPS PER BS COMPRISED OF 32 SU'S, 90 SE'S, AND 32 BU'S; U PBSU'S EACH WITH U SINGLE CHIP SBU'S.

N_e	U	NUMBER BS'S	NUMBER PBSU'S	TOTAL CHIPS
8	1	1	0	154
32	1	1	0	154
64	2	4	2	680
512	16	256	16	39680
1024	32	1024	32	158720

ASSUMPTIONS: 1 CHIP PER BS; U PBSU'S WITH U SINGLE CHIP SBU'S EACH.

N_e	U	NUMBER BS'S	NUMBER PBSU'S	TOTAL CHIPS
8	1	1	0	1
32	4	16	4	32
64	8	64	8	128
512	64	4096	64	8192
1024	128	16384	128	32768

zero is $i \cdot \delta t$ (Fig. 5). It follows that the time at which the computation line for $B\#j$ intersects the samples of $S\#i$ can be expressed as a time slot in the Nyquist period as

$$t_{ij} = (i \cdot j) \text{ MOD } m, \qquad 0 \leq j \leq B. \qquad \text{(A.1)}$$

The number of distinct values which t_{ij} assumes in (A.1) over the range of j is the value of SDM_i. Letting t_{i0} be the time slot for the broadside beam yields the relationship

$$\text{SDM}_i = \text{MIN } [n] \qquad \text{(A.2a)}$$

where n belongs to the set of beams for which

$$t_{\text{in}} = t_{i0}, \qquad 1 \leq n \leq B.$$

It follows that

$$\text{SCM}_i = M/\text{SDM}_i. \qquad \text{(A.2b)}$$

If BU$\#j$ is connected to the output s_{ij} of SU$\#i$, then

$$t_{ij} = s_{ij} \cdot \text{SCM}_i$$

and, therefore, from (A.1),

$$s_{ij} = ((i \cdot j) \text{ MOD } M)/\text{SCM}_i. \qquad \text{(A.3)}$$

It is apparent from (A.3) that a number of sensors in the system may have the same values of SCM and SDM. Such sensors are easily identified and are given a common sensor index (SI).

The queue must have one delay element for each Nyquist period, or part thereof, of delay. Therefore, the queue length is

$$q_{ij} = \begin{cases} f_{ij}/M & f_{ij} \text{ MOD } M = 0 \\ \text{ROUND } [f_{ij}/M] & \text{otherwise} \end{cases} \qquad \text{(A.4)}$$

where

$$f_{ij} = (N - 1 - i) \cdot j$$

is the number of δt periods for which a sample, from $S\#i$, must be delayed to align it with other samples, from other sensors, destined for the same sample of $B\#j$. Equation (A.1) gives the time slot during which a transfer from the ith SU to the jth BU occurs. As the intervals between successive samples on a computation line are necessarily equal, the BCM for BU$\#j$ (BCM_j) is determined from the expression

$$\text{BCM}_j = \text{MIN } [t_{ij}], \qquad 1 \leq i \leq N, \qquad \text{(A.5a)}$$

and the beam-distributor modulus (BM_j) from

$$\text{BM}_j = M/\text{BCM}_j. \qquad \text{(A.5b)}$$

Acknowledgment

The authors are grateful to Dr. M. Shepherd of Dalhousie University, Halifax, and Dr. R. C. Bowen of Carleton University, Ottawa, for their input to the editing of this paper.

References

[1] P. Barton, "Digital beam forming for radar," *IEE Proc. F.*, vol. 127, no. 4, pp. 266–277, 1980.

[2] ——, "Adaptive beam forming for radar," *Electr. Commun.*, vol. 57, no. 1, pp. 62–69, 1982.

[3] D. Borgmann, "An experimental array-antenna with digital beam forming for advanced radar applications," *APS Int. Symp. Dig. Antennas Propagat.*, vol. 2, pp. 431–443, 1982.

[4] D. M. Dias and J. R. Jump, "Analysis and simulation of buffered delta networks," *IEEE Trans. Comput.*, vol. C-30, pp. 1–8, Apr. 1981.

[5] D. E. Dudgeon, "Fundamentals of digital array processing," *Proc. IEEE*, vol. 65, pp. 898–904, 1977.

[6] H. Fan, W. K. Jenkins, and E. I. El-Masry, "Beamforming via apatial signal extrapolation," in *Proc. 27th Midwest Symp. Circuits Syst.*, June 1984, pp. 141–144.

[7] H. Fan, E. I. El-Masry, and W. K. Jenkins, "Resolution enhancement of digital beamformers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 1041–1051, Oct. 1984.

[8] M. A. Fischetti, "Solid state," *IEEE Spectrum*, vol. 12, pp. 58–63, 1984.

[9] W. C. Knight, R. G. Pridham, and S. M. Kay, "Digital signal processing for sonar," *Proc. IEEE*, vol. 69, pp. 1451–1506, 1981.

[10] D. H. Lawrie, "Access and alignment of data in an array processor," *IEEE Trans. Comput.*, vol. C-24, pp. 1145–1155, 1975.

[11] W. Martin, "The microprogrammable beamformer," Raytheon, Tech. Rep., 1974.

[12] R. A. Mucci, "A comparison of efficient beamforming algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 548–558, 1984.

[13] R. G. Pridham and R. A. Mucci, "A novel approach to digital beamforming," *J. Acoust. Soc. Amer.*, vol. 63, no. 2, pp. 425–433, 1978.

[14] ——, "Digital interpolation beamforming for low-pass and bandpass signals," *Proc. IEEE*, vol. 67, pp. 904–919, 1979.

[15] W. Robertson and D. Pincock, "Architecture for time domain digital beamforming," in *Proc. 27th Midwest Symp. Circuits Syst.*, June 11–12, 1984, pp. 145–148.

[16] W. Robertson, D. Pincock, and D. N. Swingler, "A modular architecture for digital time domain beamforming," in *Proc. Int. Assoc. Sci. Technol. Develop. (IASTED) Int. Symp. Appl. Signal Processing and Digital Filtering*, Paris, France, June 1985.

[17] ——, "A reconfigurable architecture for digital time domain beamforming," in *Proc. 14th IEEE Int. Conf. Parallel Processing*, Aug. 1985, pp. 689–696.

[18] W. Robertson, "An architecture for digital time domain beamforming," Ph.D. dissertation, Tech. Univ. Nova Scotia, 1986.

[19] H. J. Siegel and S. D. Smith, "Study of multistage SIMD interconnection networks," in *Proc. 5th Annu. Symp. Comput. Arch.*, 1978, pp. 223–229.

[20] H. J. Siegel and R. J. McMillen, "The multistage cube: A versatile

interconnection network," *Computer*, vol. 14, no. 12, pp. 65-76, 1981.

[21] E. E. Swartzlander, "Signal processor design for digital beamforming (radar systems)," in *IEEE EASCON '80 Rec. Electron. Aerosp. Syst. Conv.*, Oct. 1980, pp. 234-238.

[22] R. J. Urick, *Principles of Underwater Sound for Engineers.* New York: McGraw-Hill, 1975.

[23] C. Wu and T. Feng, "On a class of multistage interconnection networks," *IEEE Trans. Comput.*, vol. C-29, pp. 694-702, 1980.

[24] —, "The universality of the shuffle-exchange network," *IEEE Trans. Comput.*, vol. C-30, pp. 324-332, 1981.

William Robertson (M'75) was born in Scotland on July 8, 1945. He received the B.Sc. Eng. (Hons) degree from the University of Aberdeen, Scotland, in 1967, the M.Sc. degree from the University of Aberdeen in 1969, and the Ph.D. degree in electrical engineering from the Technical University of Nova Scotia, Canada, in 1985.

From 1971 to 1973 he was an Engineer in the Department of Telecomms Planning, Republic of South Africa. He has held teaching positions at the University of Cape Town and the University of Stellenbosch, Republic of South Africa, between 1973 and 1978; engineering positions at Bell Northern Software Research and Taylor Instrument Company from 1978 to 1980. He is currently an Associate Professor of Electrical Engineering at the Technical University of Nova Scotia, Halifax, N.S., Canada.

Douglas Pincock (S'67-M'75) was born in Vancouver, B.C., Canada, on September 29, 1940. He received the B.Sc. degree in electrical engineering from the University of Manitoba, Winnipeg, in 1963, the M.Sc. degree from the University of Manitoba in 1967, and the Ph.D. degree in electrical engineering from the University of New Brunswick, Fredericton, in 1971.

From 1963 to 1966 he was a member of the Royal Canadian Air Force. He has held teaching positions at St. Francis Xavier University, the University of New Brunswick, the University of Paris, and the Technical University of Nova Scotia. Since 1981 he has been Director of Applied Microelectronics Institute, Halifax, N.S., Canada.