Technical Note 1971-22

B. Gold

General Purpose Digital Signal Processing Architectures for Radar

16 April 1971

Prepared for the Department of the Air Force
and the Advanced Research Projects Agency
under Electronic Systems Division Contract F19628-70-C-0230 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Lexington, Massachusetts

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

GENERAL PURPOSE DIGITAL SIGNAL PROCESSING ARCHITECTURES FOR RADAR

B. GOLD

Group 64

TECHNICAL NOTE 1971-22

16 APRIL 1971

LEXINGTON MASSACHUSETTS

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored in part by the Department of the Air Force under Contract F19628-70-C-0230 and in part by the Advanced Research Projects Agency of the Department of Defense under Contract F19628-70-C-0230 (ARPA Order 1559).

ABSTRACT

Digital signal processing techniques expand the capabilities of modern radars. The central algorithm of radar signal processing is the fast Fourier transform (FFT). However, each radar has several modes and in most cases, environmental factors such as noise and clutter background are not completely understood. For this reason, flexibility of the signal processor is desirable. A method of gaining this flexibility is via general purpose (i.e., programmable) digital signal processing computer structures. In this note, a variety of such structures, both programmable, yet suitable for high speed FFT, are expanded.

Accepted for the Air Force
Joseph R. Waterman, Lt. Col., USAF
Chief, Lincoln Laboratory Project Office

GENERAL PURPOSE DIGITAL SIGNAL PROCESSING ARCHITECTURES FOR RADAR

I. Introduction

Modern radars with MTI capabilities are called upon to perform a variety of signal processing tasks, such as ground mapping, beam sharpening, track while search, moving target detection in a heavy ground clutter environment etc. Also, for new radar systems, extensive testing may be desirable before the final configuration is specified. In addition, there are occasions when a radar signal processor is needed to obtain data on the nature of the return signal from different media, such as desert, foliage, sea, clouds, cities, etc. For these reasons, an important component of many radars is the signal processor.

It has been well established that for an important class of radar systems, it is beneficial to use the techniques of digital signal processing. If versatility is also required, we suggest that a general purpose signal processing capability is highly desirable. This capability can be specified more precisely as:

1. Real-time capability to perform spectral analysis.
2. Real-time capability to perform a variety of processing algorithms by programming the signal processing equipment.

This implies that a desirable goal of airborne radar R&D is the eventual construction of a general purpose digital processor, suitable for airborne use, with an architecture which permits rapid spectral analysis and other signal processing functions such as windowing, magnitude taking, etc. It seems certain that spectral analysis in this context is best performed via the fast Fourier transform (FFT).

Most work up to now in airborne radar processors has emphasized the concept of a hard-wired FFT box as the central signal processing element. In contrast, we would like to emphasize the incorporation of FFT algorithms within the framework of a high speed programmable signal processor. That this is a feasible approach is evidenced by work already completed in the development of an experimental ground based radar system. In the next section, we will briefly describe this demonstration radar. We will then discuss an example of a signal processing requirement for a particular airborne radar system which has been described in a separate publication[1]. Finally, we will indicate several promising directions in system architecture and componentry applicable to the development of an airborne radar signal processor.

II. The Demonstration Radar

Fig. 1 shows a block diagram of the gate-selection and signal processing for the demonstration ground radar. In this system, the pre-summing and gate selection is performed by high-speed special-purpose digital hardware. The buffer memory is a large core memory which is controlled by a special-purpose address box which permutes the space and time coordinates of the radar signal, and provides the storage necessary for post-detection integration. The fast digital processor (FDP), a high speed programmable signal processing computer[2] designed and built at the Lincoln Laboratory, has been in operation since October 1970. Its function in the demonstration radar experiment is to perform (in one of its modes) 1000 FFT's per second (64 points per FFT) together with several other operations: e.g. magnitude computations, doppler filtering and post detection integration. As the program develops, we expect that the flexibility of the FDP will allow great experimental freedom.

18-00-7258 (1)

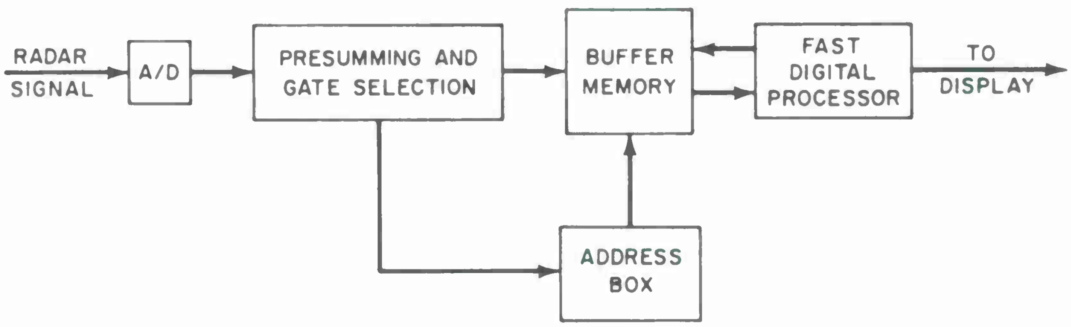


Fig. 1. Gate selection and signal processing for demonstration radar.

The architecture of the FDP allows real-time processing of radar signals at about 100 times the speed that a conventional general-purpose digital computer would allow. This capability can be utilized in the early stages of a program directed towards the development of an airborne radar system, by using recordings made in flight as inputs to the facility.

III. Signal Processing Requirements of a Two-Antenna Airborne Radar

Given MTI processing ability, an airborne radar system is able to create a high resolution ground map. However, because of the large doppler spread of the ground clutter caused by the motion of the airborne platform, moving targets are either difficult or impossible to detect. This clutter may be greatly reduced by means of a multi-antenna technique, leading to an airborne surveillance system which should be able to search for and track moving targets in heavy clutter background even when the platform flies at high speeds such as Mach 1. Aspects of such a system are analyzed in some detail in Reference 1. In this section, we make use of some of those results to estimate signal processing requirements.

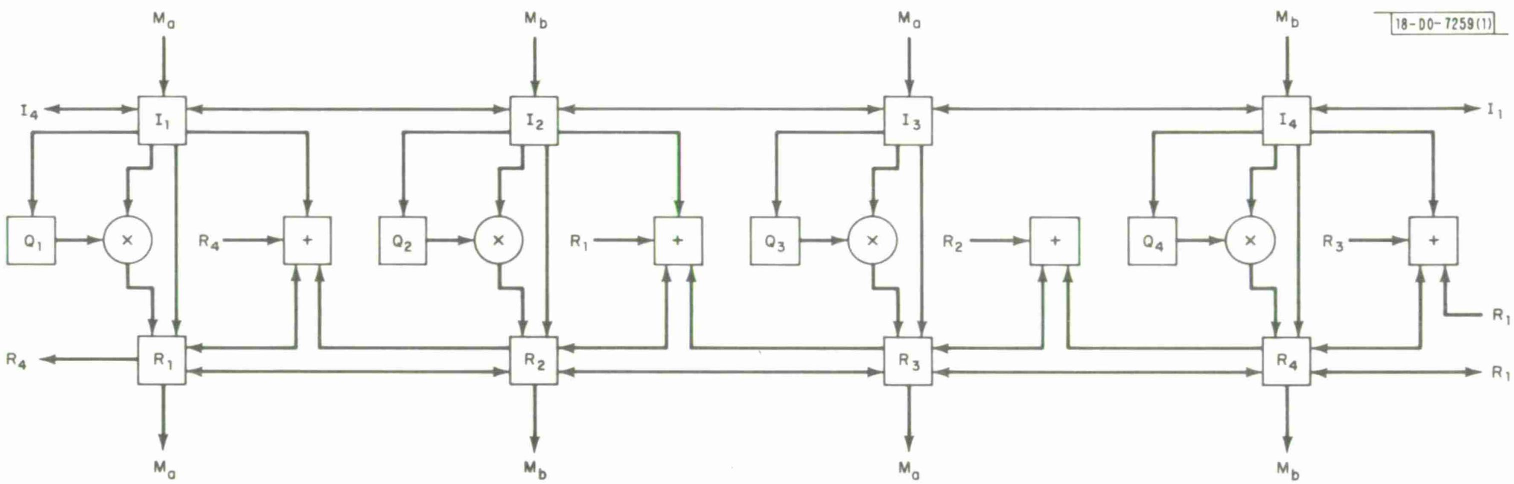
We know that the FDP can perform 1000 64 point FFT's per second plus other algorithms needed to achieve detection of moving targets in a ground radar. This suggests that a two-antenna airborne processor can process two 32 point FFT's per second for 1000 gates. Assuming an antenna beam width of 2°, an angle coverage of 120°, a range coverage from 30 to 100 nautical miles and a range resolution of 200 feet, we find that this total coverage is obtained by processing 126,000 range gates. Thus, if one complete search were made every 126 seconds, a signal processor with FDP capability would be sufficient. Within the constraints of this processing capability, many options exist such as the fineness of spectral measurement, range resolution, angle and range coverage, etc.

IV. Computer Structures with Good FFT Capability

The two starting points for discussion of future signal processing computers are the recent development of the FDP and the LX-1 microprocessor[3]. The FDP was designed with the FFT computation in mind. The LX-1 is not sufficiently fast for radar signal processing but we shall try to show that relatively straightforward modifications can rectify this situation. In the remainder of this section we describe in more detail both the FDP and LX-1 types of architecture and several interesting variations. From these descriptions we will arrive at several conclusions as to the relative merits and shortcomings of these structures as applied to the airborne radar problem. We also include a discussion of some integrated circuit technology and its effect on these structures.

1. FDP

The basic FDP structure is shown in Fig. 2. For details see reference 2. Fig. 2a shows the main signal paths in the arithmetic system. There are four identical arithmetic elements (AE1, AE2, AE3, AE4), each element containing 3 registers, an adder and a buffered multiplier. A single 18 bit instruction permits all 4 AE's to operate in parallel. Figure 2b shows the parallelism between the three main computer functions: arithmetic, memory and control; we see that any 2 of these three can function in parallel. Since memory consists of two separate and independently addressable banks, M_a and M_b, a memory instruction is also a parallel operation. A separate high speed memory contains the program and this



(a) ARITHMETIC SYSTEM

←18→	←18→
MEMORY	CONTROL
ARITHMETIC	CONTROL
MEMORY	ARITHMETIC

(b) INSTRUCTION SYSTEM

Fig. 2. Structure of FDP.

memory operates in parallel with M_a and M_b. Thus, the FDP achieves high speed by combining high speed circuits (the basic cycle time is 150 nanoseconds) with a high level of parallel computation so interleaved, however, that the machine can be programmed sequentially as is a conventional computer. The inner loop (basic computation) of an FFT typically takes 10 instructions or 1.5 microseconds on the FDP.

The FDP is built from emitter-coupled logic (ECL) with a typical gate propagation time of 4 nanoseconds. A large family is commercially available for this line of logic. Using this logic, the 4 FDP array multipliers (450 nanoseconds for a single, 2's complement 18 x 18 bit multiply) require 5 FDP boards (a board is 9" x 16" and holds about 140 integrated circuits packages). The complete arithmetic and control circuitry contain about 8000 integrated circuit packages.

The FDP has proved to be a flexible high speed processor, relatively simple to program and with adequate input-output capabilities for both real time and non-real time applications. Its present realization, however, makes it unsuitable for airborne use.

The primary motivations for the relatively large size of the FDP came from the desire to accelerate the development time by using wire-wrap rather than printed circuit techniques and make the circuits very accessible for troubleshooting. Appreciably more compactness can be attained using the basic FDP architectures. In addition, however, airborne capability would also depend on reducing the package count by means of a higher level of integration or changes in the architecture, or both. An important point to consider is the fact that, for radar applications, 12 bit arithmetic registers are sufficient (compared to the 18 bit word length of the FDP);

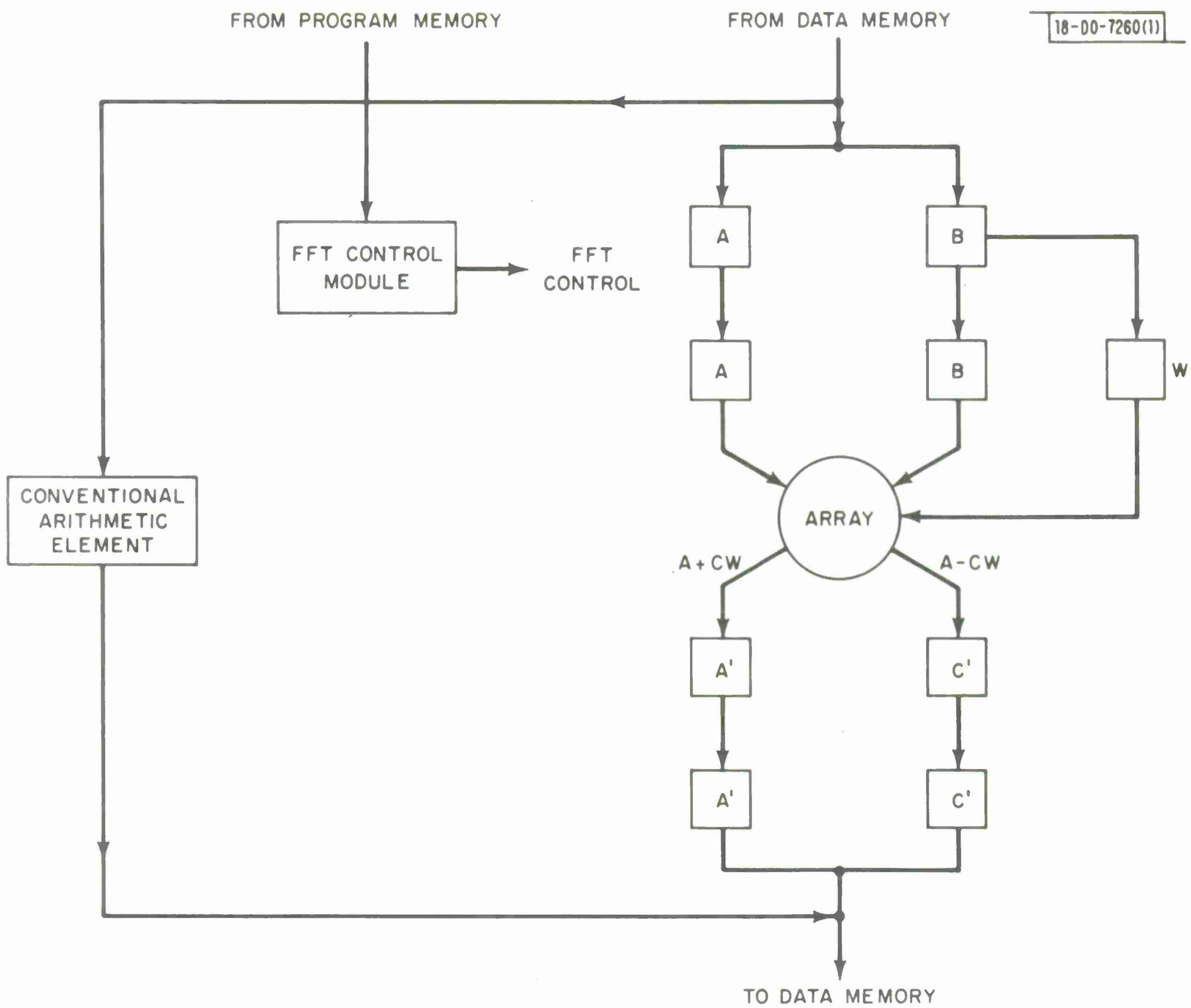


Fig. 3. Special butterfly attachment to general purpose computer.

this fact alone permits a considerable reduction in the FDP package count and also speeds up the multiplier, increasing the throughput.

2. Use of a "Butterfly" Array

The basic FFT algorithm requires the repetition of $\frac{N}{2} \log_2 N$ basic calculations, which are commonly called 'butterflies'. Each butterfly is defined by the equation,

$$A' = A + CW_i$$

$$C' = A - CW_i$$

where the complex numbers A and C may be considered to be the inputs to a butterfly with A' and C' as the corresponding outputs which then became inputs to a subsequent butterfly.

The equation shown here corresponds to a 'decimation in time' algorithm[4]. 'Decimation in frequency' makes use of the equation,

$$A' = A + C$$

$$C' = (A-C)\, W_i$$

In both cases W_i are the complex coefficients, $W_i = \cos\, \theta_i + j \sin\, \theta_i$ where the angle θ_i depends on the specific butterfly being performed.

The instruction repertoire of the FDP is similar to that of conventional computers. There are no special FFT instructions, but the high degree of parallelism is designed to make FFT programming more efficient. An alternate way of achieving both generality and FFT efficiency is by superposing special hardware and special instructions onto an otherwise conventional computer structure.

One such scheme is shown in Fig. 3. Here, memory is assumed to consist of double length registers which contain both the real and imaginary components of the complex numbers comprising the data. The conventional arithmetic element

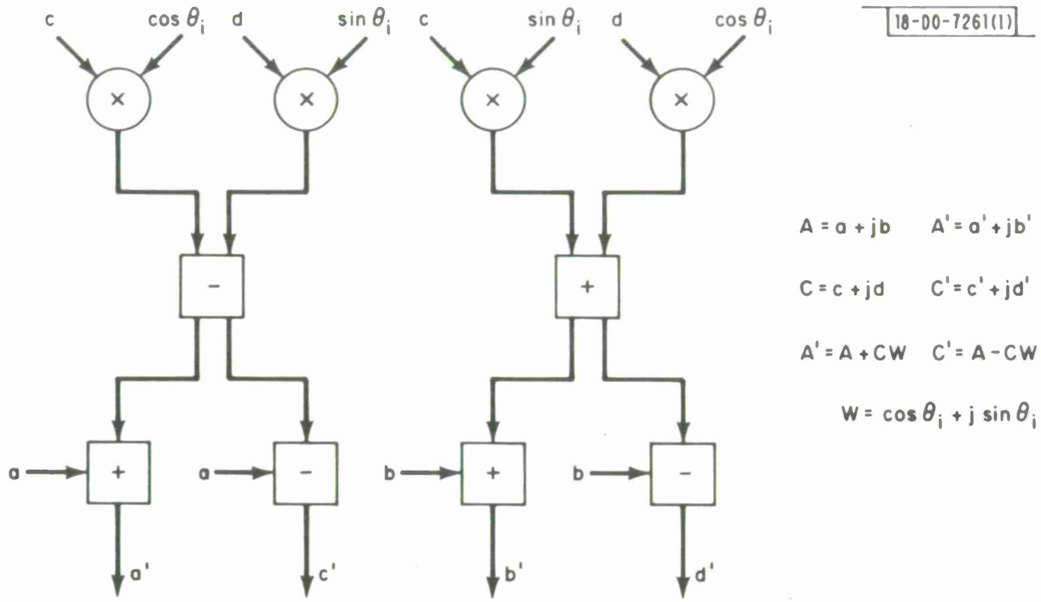


Fig. 4. Combinational butterfly.

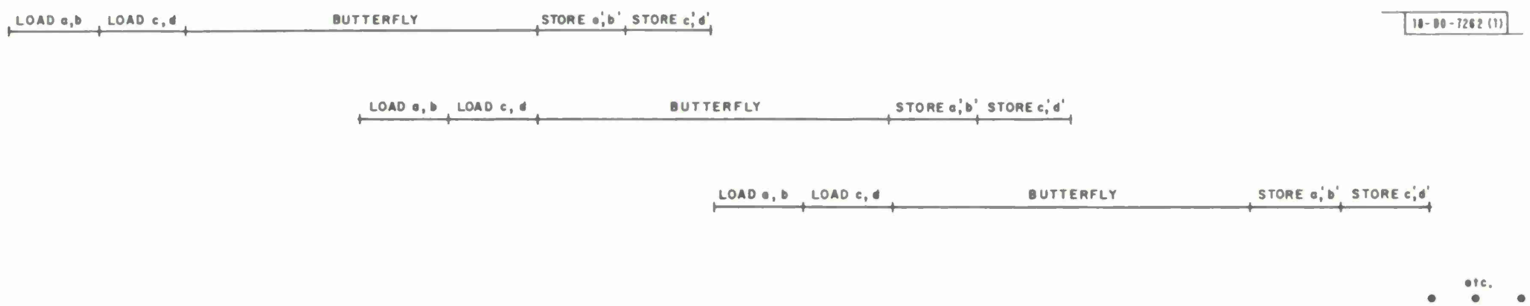


Fig. 5. Timing of a butterfly.

would correspond to a byte-oriented word (reasonable numbers are 24 bit words storing two 12 bit bytes). The 'array' shown and detailed in Fig. 4 is a purely combinational circuit which accepts as inputs the 3 complex numbers A, C and W and produces the two outputs, A' and C'. Nine double length registers are required in order to make this special circuitry perform the FFT at optimum speeds. The special FFT circuit and the conventional arithmetic element share the same memory. The FFT control module receives a special instruction from the computer program memory which actuates the data memory and the FFT arithmetic module. This special instruction (or instructions) could conceivably carry enough control information to be a set of butterflies for a constant W, a complete FFT level ($\frac{N}{2}$ butterflies), or a complete FFT. The optimum design would allow a butterfly to be performed in 4 memory cycles; this is illustrated by the timing diagram of Fig. 5.

The scheme of Fig. 3 is substantially more general than a scheme where the FFT hardware is completely divorced from the remaining signal processing operations. This generality derives first from the sharing of the same memory by the FFT and all other operations. In addition, the order of the FFT and the word length used are completely under computer control. This means that the entire signal processing is centralized, both as to format and control.

3. Compromise Between the FDP and Special Butterfly Array

In the FDP design, great care was taken to maintain the integrity of the general purpose structure. Thus, given the desire for 4 multipliers, it was felt that the programmer ought to be able to utilize these 4 multipliers in routines other than the FFT. For example, it is easy to code a program to multiply data points by a window function,

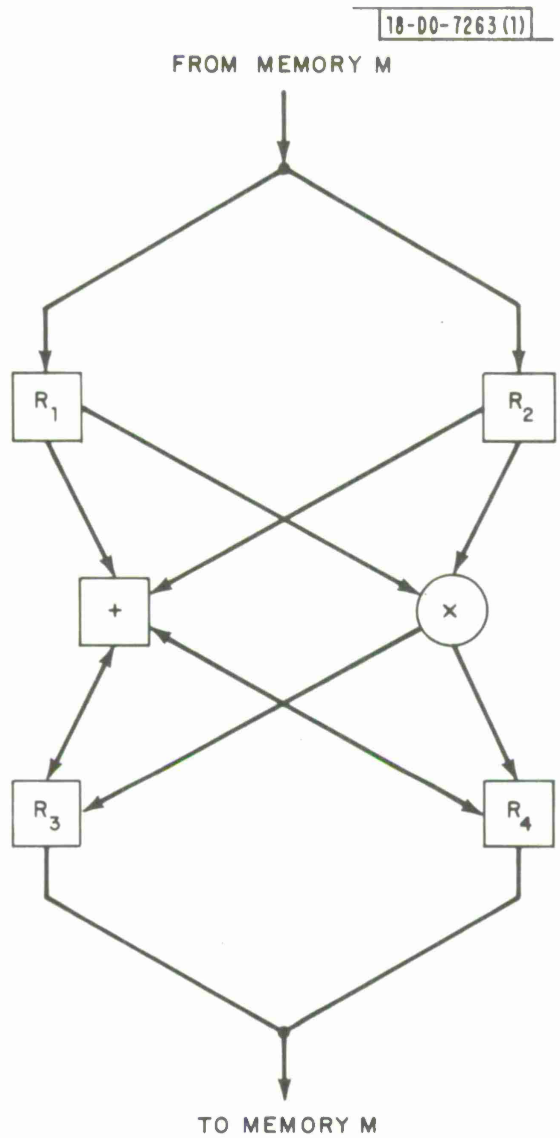


Fig. 6. Complex arithmetic element.

making use of this parallel capability, so that N multiplications can be done in N/4 iterations. On the other hand, in the structure of Fig. 3, no such integration was attempted. Thus, only the FFT was performed with great efficiency; other routines take as many instructions as on conventional computers, with speed being gained solely through the use of fast circuits.

We now describe a third structure which is a compromise between these two extremes. Here we assume that the emphasis is on the manipulation of complex functions. A possible configuration is shown in Fig. 6. R_1, R_2, R_3 and R_4 are double length, or 'complex' registers and the adder and multiplier are also complex. A program for performing a single butterfly $A' = A + CW$, $C' = A - CW$ is shown below: we assume that initially R_2 contains W.

Program	Interpretation
1. $M \rightarrow R_1$	$C \rightarrow R_1$
2. $R_1 \times R_2 \rightarrow R_4$	$CW \rightarrow R_4$
3. $M \rightarrow R_1$	$A \rightarrow R_1$
4. $R_1 + R_4 \rightarrow R_3$	$A' = A + CW \rightarrow R_3$
5. $R_1 - R_4 \rightarrow R_4$	$C' = A - CW \rightarrow R_4$
6. $R_3 \rightarrow M$	Store A'
7. $R_4 \rightarrow M$	Store C'

If this structure were to include the control-memory-arithmetic parallelism of the FDP, then indexing could be buried in this routine and two memory cycles could be saved. Depending on the speed of the multiplier, a substantial saving is possible in butterfly time. Furthermore, it should be easy to construct this computer

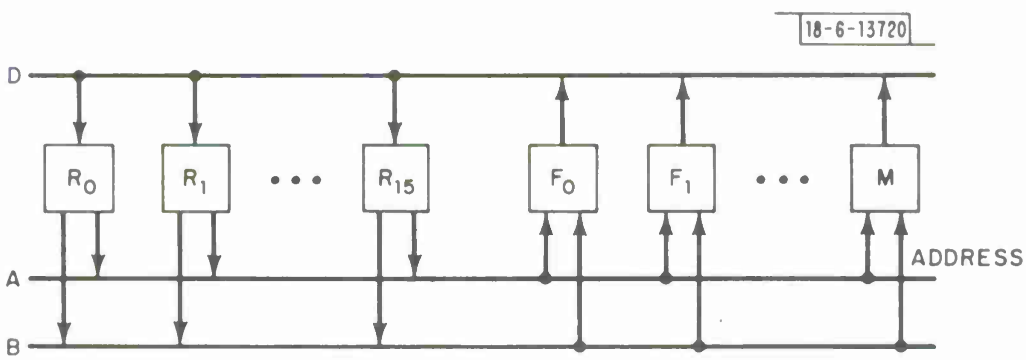


Fig. 7. LX-1 microprocessor.

in a 2 byte-oriented way so that, for example, 2 multiplications or 2 additions could be performed simultaneously.

4. Microprocessors

A microprocessor is a form of general purpose computer designed so that control of the computer resides in a special memory. In one way this is a generalization of the stored program concept since by changing the special memory the control of the computer can be changed. In another way, this technique is restrictive, since, the microprogram is not self-modifiable. Since this restriction does not appear to be harmful in radar application, we were lead to consider some possible microprocessor structures which seemed favorable for signal processing algorithms. We begin with a brief description of LX-1, the microprocessor built at Lincoln Laboratory and then study several variations on this basic structure.

Fig. 7 shows the LX-1 configuration. It consists of 2 output busses A and B, and input bus D, 16 general registers R_0 through R_{15} and an arbitrary number of function generating boxes F_0, F_1, etc. Logically, memory M is treated as a function box, with A as the write input, B as the address and D the read output. The functions are operations such as add, scale, multiplication, etc. Because of the bussing scheme, there is great flexibility in handling the general registers; for example, part of a single instruction may be $R_i + R_j \rightarrow R_k$ where i, j, and k are arbitrary; thus there is one instruction path from any register to any other register. This permits more flexible arithmetic manipulation on the one hand with the associated disadvantage of a lack of parallelism. For example, the FDP can do 4 parallel additions

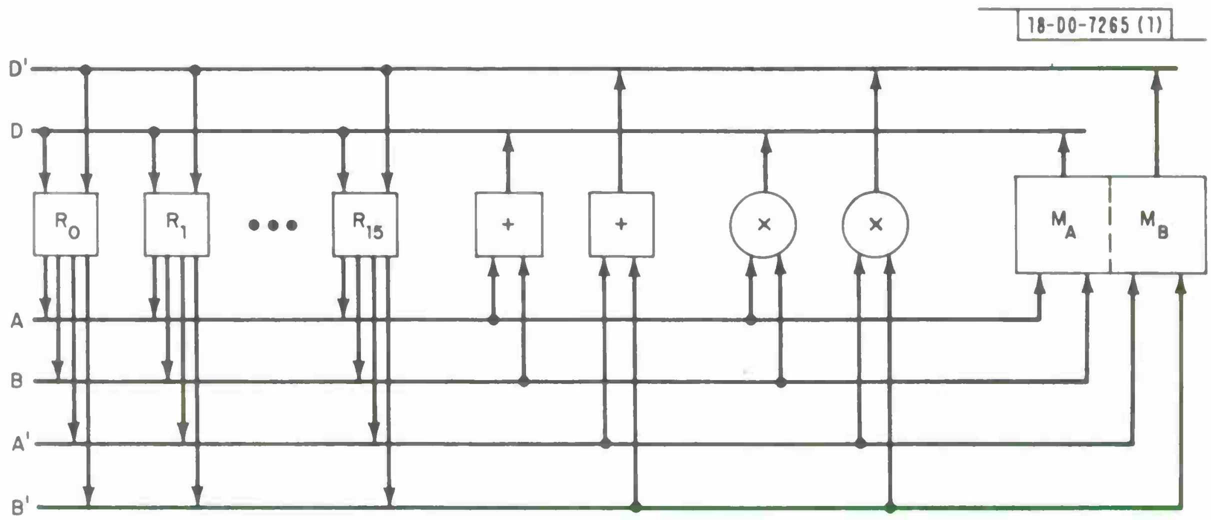


Fig. 8. Microprocessor with double bussing, pairs of function units and double length memory.

but only between certain prescribed registers while the LX-1 can do a single addition among any register combination.

Analysis of the FFT computation time for an LX-1 program yields the following results: The arithmetic portion of the butterfly takes 10 instructions (4 multiplications and 6 additions), while the memory portion takes 8 instructions. Indexing becomes a little difficult because of the limited number of registers; if 32 general registers were used, about 6 indexing instructions would be necessary. Let us guess at 30 instructions. The LX-1 cycle time is 70 nanoseconds and data-program overlap is not perfect; thus 2.5 μsec per butterfly is a reasonable estimate, (about twice the FDP time with about three to four times the number of instruction cycles). Note that in LX-1 (in contrast to the FDP) memory, arithmetic and indexing cannot be performed in parallel. This, of course, means simpler logic but also more instructions per algorithm. Note, also, that the microprocessor and FDP program memories are quite similar, being physically separate from the rest of the system and more or less non-modifiable.

Another interesting aspect of LX-1 is the general register configuration. This appears to require less hardware than the FDP AE configuration and despite its serial nature is still quite powerful arithmetically. A possible compromise is shown in Fig. 8. Three additional busses have been added to the same registers and an extra multiplier and adder have been added. This structure essentially halves the butterfly time compared to LX-1.

We conclude this section with a brief description of a modified version of LX-1 which appears to be a good compromise between versatility, speed and size and cost. This description is at present tentative and incomplete, not including the input-output

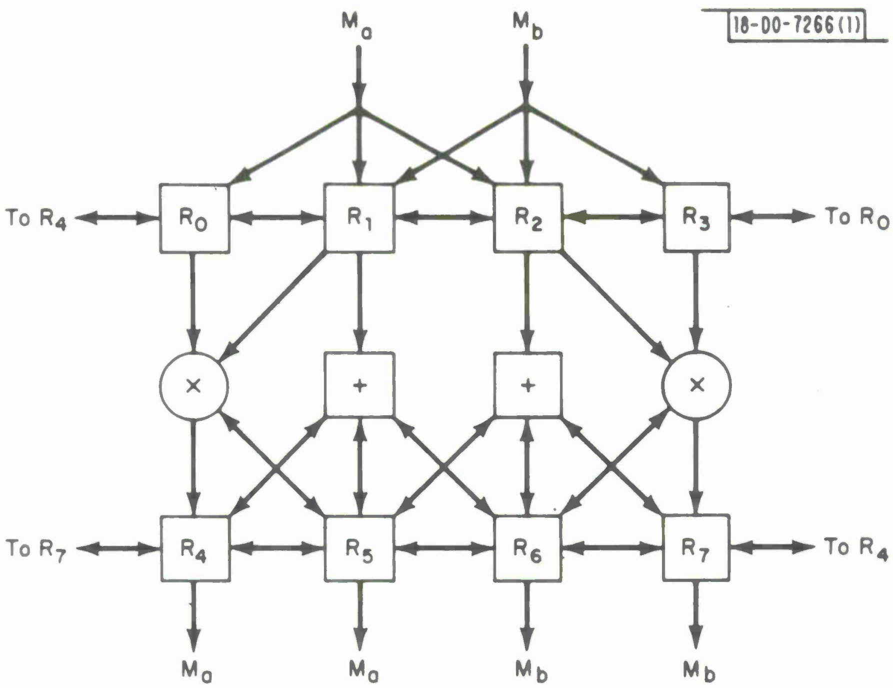


Fig. 9. Eight register arithmetic structure.

structure or branching, but does outline how arithmetic and memory combine to yield enhanced signal processing capability.

The word length of an LX-1 register is 16 bits. In our structure, illustrated in Fig. 9, the word length has been extended to 24 bits arranged as two 12 bit bytes. To allow for flexible manipulation of the bytes, permutation is introduced on the B bus and activity is introduced just prior to entry on the D bus. The multiplier function box consists of two 12 x 12 bit multipliers and both the adder-logic units and shift units come in pairs. Memory addressing is via the 12 bit general register bytes and the use of permutation and activity allows the use of any of the 32 general bytes as addressing registers. With this structure the arithmetic and memory portion of butterflies requires 10 microprocessor instructions. Instruction cycle time is estimated to be between 50 and 100 nanoseconds. If faster FFT's are desired, extra arithmetic hardware can be attached as in-out devices to a 24 bit general register. Additional speed can be obtained by connecting the general registers to the memory via a separate bussing scheme, which would thus allow memory and arithmetic operations to proceed in parallel.

V. Hardware Considerations

In addition to the many possible different possible computer structures, there are a variety of circuit types. The properties of these different circuits have recently been reviewed in a series of three IEEE Spectrum articles[5]. In designing a computer, it is desirable to choose a single logic family; otherwise extra complications result from the need to interface logic with differing voltage levels. Some of the important attributes of logic families are:

1. Availability of a wide variety of package types.
2. Amount of integration per package.
3. Power required.
4. Cost.
5. Speed.
6. Noise immunity.
7. Temperature sensitivity.
8. Reliability.

The FDP was built using ECL logic as mentioned previously; at the present writing, this logic is still the most versatile high-speed family that is commercially available. Somewhat slower but as versatile and more highly integrated is TTL logic. Appreciably slower is the MOS circuit; this appears to be the most highly integratable and is receiving much attention from component manufacturers. Perhaps the fastest commercial circuits available are new ECL circuits with 1-2 nanosecond propagation times. These circuits have the disadvantages of requiring much power, and are more temperature sensitive; they are not highly integrated and very few circuit types are presently available. However, they are of interest to consider as part of a potentially practical future system.

The use of the FDP in the demonstration radar proves that digital processing techniques can result in a greater capability than can be attained feasibly by analog techniques. However, it is still well to keep in mind that the digital hardware required is still quite formidable. Speed and memory requirements increase linearly with either the number of range resolution or velocity resolution cells, so that cost and size tend to rise linearly with these demands. By too casual use of numbers the radar engineer can convince himself that the capabilities of digital processing are almost infinite and that he can have nearly any resolution and coverage he desires. For example, new ECL advertisements and some recent pioneering work on array multipliers at Lincoln Laboratory[6] indicate that 12 x 12 bit multiplication can be performed in about 25 nsec. By placing four of these in parallel, one can then design a butterfly array to work in 30 nsec. Using pipeline FFT techniques[7] only N/2 butterflies are needed to perform a complete FFT. Thus, a 64 point FFT ought to take 30 x 32 = .96 μsec. We stated before that the FDP is capable of 1000 such transforms per second; thus, the 'new' techniques can result in 3 orders of magnitude greater capability. Let us analyze such claims carefully and see if we can discover the degree of their validity.

First of all, the specific array multiplier that has been built is composed of special packages made by a manufacturer who has recently suspended his integrated circuit activities. Its degree of reproducibility and temperature sensitivity have yet to be determined. Second, such an array uses a large amount of power. The pipeline FFT postulated above requires 24 such arrays. Comparable speeds appear to be attainable with new ECL circuits but the power requirements are even greater and the degree of integration less. We doubt that it is at present feasible to employ more than 4 such array multipliers for an airborne processor.

To keep up with the speed of these arrays would require very fast memory and control circuitry, which is not presently available. Also, since the pipeline is a special purpose processor it would have to be augmented by other processing algorithms which would further raise the hardware complexity, or alternatively by a general purpose processor which could not be expected to keep up with the pipeline. In addition to all this, assembling a large signal processing system with super-fast circuits where propagation time on wires is a critical factor is a very difficult and time-consuming engineering job. Finally, faster processing implies a proportional increase in buffer memory size and in the complexity of gate selection hardware.

The point of these comments is to make one aware of the long range potentialities of the digital approach to radar signal processing but also to point out the dangers of assuming that these potentialities are realities. Our feeling is that present digital techniques are very promising for constructing airborne systems with the power of the FDP. We would hope that in several years, 5 to 10 times this power becomes attainable for practical airborne systems. Since, to our knowledge, no existing airborne radar has FDP-like capabilities, these goals seem worth striving for. The remainder of this section is devoted to ideas for attaining such capability in an airborne system as economically as possible. Let us first see which of the structures that we have discussed, coupled with a particular logic line, leads to digital processors comparable to the FDP.

1. 4 Nanosecond ECL Circuits

The FDP was constructed with these circuits. Typical 18 bit add times are 70-100 nanosecond and the 18 x 18 bit array multiplication takes 450 nanoseconds.

In order to realize FDP power with these circuits in an airborne environment would require, in addition to substantial re-packaging, that the number of packages be reduced from 8000 packages to between 2000 and 4000 packages. This saving is not too difficult to attain given the fact that 12 bits is a very reasonable word length for radar applications. Thus, all arithmetic registers, memory and gating are reduced by 33¢. In addition, the 4 multiply arrays are each reduced by more than a factor of two, saving more than 1000 packages. Additional saving can be attained by single rather than double address formation (The FDP memories M_a and M_b are each independently addressable by the instruction word). Finally, the complexity of the FDP AE system can probably be reduced without greatly affecting the FFT speed. An example is the arithmetic system of Fig. 9, where only two multipliers, two adders and 8 registers are used. If this system is incorporated into the FDP control and memory structure and if we assume that the word length reduction and more compact construction result in a 2:1 speed increase in the basic cycle time, the overall speed should be about twice that of tne FDP (for FFT's), and about the same for other routines (the loss of the 4 parallel AE's being compensated for by the 2:1 speed increase).

To increase speed by a factor of 5 to 10 using 4 ns ECL requires a greater degree of parallelism than is embodied in the FDP. With the array concept shown in Figs. 3 and 4 we could expect to achieve a butterfly in about 300 nsec., which is about a 4:1 increase over the FDP tor the FFT but, given conventional arithmetic for other signal processing purposes, there is little likelihood that this gain can be realized in general. If we tried to combine the array with an FDP architecture, compactness would be compromised. Our general feeling is that 4 ns ECL is not a suitable vehicle for substantial speed increases relative to the FDP.

2. 1-2 Nanosecond ECL

1 ns ECL circuits, in conjunction with high speed (10-30 nsecond) memories if used in an FDP like structure should result in 5 to 10 times the speed of the FDP. At the present writing, we have done little work with these circuits and, also, an extensive logic line does not exist. The same comments hold for the 2 ns ECL series, which have been announced and for which versatile logic packages have been promised; it is not at all clear that these will be readily available within the next two or three years. In order to keep an eye on long range possibilities we propose to study and build simple breadboards, such as adders, using these circuits, to develop the engineering techniques needed to eventually put together a complete system.

The use of 1 ns ECL and a fast memory in the LX-1 microprocessor could conceivably make this system faster than the FDP. Since the LX-1 is simpler than the FDP, the major effort in realizing a 1 ns ECL version of LX-1 would be the development of packaging techniques.

3. MOS

MOS logic is appreciably slower than 4 ns ECL, perhaps by an order of magnitude. At present, no practical computer architecture is known which, with MOS circuits, exclusively could compete with the FDP. However, a combination of an ECL microprocessor such as LX-1 in conjunction with a collection of MOS arithmetic units could result in an FFT speed greater than that of the FDP. A possible method of implementing such a structure is to attach an arithmetic element which performs an FFT butterfly as an input-output device to a number of general registers. If, now, the speed of the scratch memory M is much faster than the speed of these elements, the memory could sequentially service them and sequentially retrieve the result. In this way, parallelism in the FFT algorithm could be programmed.

REFERENCES

1. E. Hofstetter, C. Muehe, C. Weinstein, "A Theory of Multiple Antenna AMTI Radar", Technical Note 1971-21, Lincoln Laboratory, M.I.T. (16 April 1971).

2. B. Gold, I. Lebow, P. McHugh, C. Rader, "The Fast Digital Processor - A Programmable High Speed Signal Processing Computer", IEEE Transactions on Computers (January 1971).

3. G. D. Hornbuckle, E. I. Ancona, "The LX-1 Microprocessor and Its Application to Real-Time Signal Processing", IEEE Transactions on Computers (August 1970).

4. B. Gold, C. Rader, "Digital Processing of Signals", McGraw-Hill (1969).

5. L. S. Garrett, "Integrated-Circuit Digital Logic Families", Spectrum (October, November and December 1970).

6. Stylianos D. Pezaris, "A 40 nsec 17-Bit by 17-Bit Array Multiplier", submitted to IEEE Transactions on Computers.

7. G. O'Leary, "A Cascade Fast-Fourier Transform Processor", Transactions Audio and Electroacoustics, (March 1969).

DOCUMENT CONTROL DATA – R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Lincoln Laboratory, M.I.T.	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP None

3. REPORT TITLE

General Purpose Digital Signal Processing Architectures for Radar

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Technical Note

5. AUTHOR(S) *(Last name, first name, initial)*

Gold, Bernard

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
16 April 1971	30	7

8a. CONTRACT OR GRANT NO. F19628-70-C-0230	9a. ORIGINATOR'S REPORT NUMBER(S) Technical Note 1971-22
b. PROJECT NO. 649L c. ARPA Order 1559 d.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* ESD-TR-71-118

10. AVAILABILITY/LIMITATION NOTICES

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
None	Air Force Systems Command, USAF; Advanced Research Projects Agency, Department of Defense

13. ABSTRACT

Digital signal processing techniques expand the capabilities of modern radars. The central algorithm of radar signal processing is the fast Fourier transform (FFT). However, each radar has several modes and in most cases, environmental factors such as noise and clutter background are not completely understood. For this reason, flexibility of the signal processor is desirable. A method of gaining this flexibility is via general purpose (i.e., programmable) digital signal processing computer structures. In this note, a variety of such structures, both programmable, yet suitable for high speed FFT, are expanded.

14. KEY WORDS

fast Fourier transform	airborne radar processors	algorithms
signal processing	butterfly array	

pll-1800