

### 8.3C SIGNAL PROCESSOR ARCHITECTURE FOR BACKSCATTER RADARS

W. E. Swartz and P. Johnston\*

\*School of Electrical Engineering, Cornell University, Ithaca, NY 14843

\*\*NOAA, Poker Flat MST Radar, P. O. Box 80128, College, AK 99708-0128

#### ABSTRACT

Real-time signal processing for backscatter radars requires enormous computational throughput and I/O rates; however, the operations that are usually performed in real time are highly repetitive simple accumulations of samples or of products of samples. Furthermore, since the control logic does not depend on the values of the data, general-purpose computers are not required for the initial high-speed processing. The implications of these facts on the architectures of preprocessors for backscatter radars are explored and applied to the design of the Radar Signal Compendor.

The Radar Signal Compendor is a programmable high-speed pipelined real-time multiprocessor machine intended for coherent and incoherent backscatter radars. Its architecture lends itself to time-critical processing where the operations performed are only the direct accumulations of samples or the accumulations of products of the original samples. The programmability of this machine allows it to be adapted to a wide range of experiments, yet without the difficulty usually found with more general-purpose array processors. The Compendor is composed of several Functional Modules which parallel process multiple data streams, a Master Control Module which provides for timing and communication between the host computer and each of the Functional Modules, and an Analog-to-Digital Conversion Module which feeds samples directly into the input memories of the Functional Modules under the control of external timing logic. Each of the Functional Modules can be individually programmed under the control of the Host Computer and the Master Control Module. Control of each of the data-processing pipelines is nearly transparent to the user, in that, control operands are tagged to the sample address operands and then follow the processing through a control pipeline for use at the proper stage. Input and output memories are fully double buffered for most usual configurations, and all memories are 2 k words deep. The four input memories of each Functional Module are 16 bits wide, while the four output memories are each 32 bits wide and can be configured as two 64-bit wide memories.

Programming the device consists of the loading of the configuration registers and the address control RAMs of each Functional Module using simple directives to the Master Control. The configuration registers establish the data flow paths that are uniquely determined for a given experiment. The address control RAMs consist of BASE plus DISPLACEMENT operands with flexible incrementing and looping control.

A Compendor with 10 Functional Modules and high-speed memories should be capable of a throughput of 100 MHz for multiply-replace-add sequences. The more modest version for the Poker Flat MST radar with 6 Functional Modules and slower memories achieves a 30-MHz throughput.

#### INTRODUCTION

Since the signals received from backscatter radars are noise like, the basic requirement of the processing hardware is to average as many samples as possible in as short a time as possible. For some experiments, the computational limitations restrict only the amount or quality of the real-time displays that can be generated. For others the limitations is a trade-off between what can be done in real-time versus what must be done off-line. Yet

for many experiments, the actual science in terms of height resolution, time resolution, number of heights, bias corrections, or dynamic interaction is limited by insuggicient compute power.

The most popular atmospheric backscatter radar experiments can be split between four major headings, as shown in Table 1. Since the correlation times of the medium being probed under each of the headings differ from one another, various transmitter pulse and receiver sampling schemes are used to optimize a given experiment. However, in every case the initial real-time fast processing is a highly redundant sequence of additions of samples or of products of samples. The bottom two lines give a comparison of the computational requirements in terms of the rate of multiply-replace-add operations. It is obvious that even state-of-art general-purpose array processors with single multipliers and adders cannot keep up for experiments requiring rates of more than just a few Megahertz. Remember too that commercial array processors use floating-point formats, yet integer arithmetic is sufficient provided the data paths are wide enough to avoid truncation of the summations. Floating-point formats can lead to subtle biases and just the conversion from the integer outputs of the analog-to-digital converters can be a bottleneck within the processor. Integer logic is simpler and faster; hence, it should be preferred for the preprocessors used with backscatter radars.

Table 1. Signal processing requirements

	<u>MST</u>	<u>E REGION</u>	<u>F REGION</u>	<u>PROTONOSPHERE</u>
Interpulse period (msec)	0.5-1.0	2-10	10-15	40
Pulse Width (usec)	0.1-4.0	2-4	4-300	1000
Number of Pulses per IFF	1	1-7	1-7	1
Coding	Various	Possibly Barker	Not Usually	No
Number of Bauds	1-256	7-13	13	
Sampling Rate (MHz)	1-20	0.25-0.5	0.05-0.5	0.5
Number of Complex Products per Sample (1)	1	1	1-50	400
Number of Lags		10-20	10-100	30 (60 if ACF is formed at IF)
Number of Heights	200-2000	20-600	20-1000	20 minimum
Rate for Multiply-Replace-Adds (MHz)	100-1000(2) 0.2-100 (2)	4-50 0.04-1.3	100 0.06-3	200 (3) 20 (4)

NOTES: (1) Multiple products are independent only when signal to noise is low. The number of real products is four times the number of complex products given.  
 (2) Rate for additions only -- multiplies not required at this level.  
 (3) Rate for unbuffered case.  
 (4) Rate for double buffered case.

The order (i.e., addressing) of the samples sent to the processor and the ordering of the processed data output to the host computer can be very simple. In fact, there is never any need for the addressing of these two transfers to be anything but sequential. For experiments requiring pulse decoding or multiple lag products, the addresses of samples being supplied to the processing stages are still highly repetitive, but not completely sequential; more will be said about this later.

#### FUNCTIONAL OVERVIEW

With these ideas in mind, one can easily write a block diagram showing the data flow for a simple signal processing example where the samples are simply accumulated before being passed on to the host computer. This has been done in Figure 1. The addressing of the Output Memory at this level can be assumed to be as flexible as required by a given experiment. Since there is no input buffer to temporarily store the samples, each accumulation must be accomplished within the sample interval. If the same sample is used for several accumulations (a very typical situation), then the time needed for multiple fetches and stores to the memory, plus the time for the accumulations, soon exceed the sample interval time, even for the fastest logic available. Many such units could be paralleled together, but one immediately realizes that typical radar applications have a significant amount of time between the end of one sample raster and the start of the next raster. The addition of a buffer memory between the ADC and the accumulator would then allow this extra time to be utilized, at least partially.

With a single memory between the ADCs and the accumulator, the next bottleneck arises when the ADC wants to write a sample to the memory at the same time as the accumulator wants to read some other sample. This would be the situation in general-purpose processors even with double buffering. (All that double buffering alleviates is the problem of guaranteeing the validity of the data before it is over-written with the next sampling sequence, assuming that the processing keeps up.) This bottleneck can only be eliminated by the use of two independent input buffers, where one can be written, while the other is being read. This configuration is illustrated in Figure 2. If the addressing of the two buffers is also independent, then sampling can proceed at the maximum rate allowed by the memory with no need to wait for the multiple memory accesses that may be required for processing.

Finally, Figure 3 illustrates the data paths required for maximum throughput when a multiplier is inserted within the data process stream. Note that this case shows four Data Input Buffers. Four buffers are needed, even for the case where the samples loaded into each memory are the same, but where the multiplications are formed between samples taken at different times (e.g., for a lag product of an autocorrelation function). These four buffers should be considered as two independent double buffers, each supplying one of the multiplicands. In this way, only one memory fetch is needed from each of two memories for each multiplication. Of course, this assumes that the memory fetch time is comparable to the multiply time, which, in practice with current technology, turns out to be true. (If the memories were twice as fast as the multiplier, so that a double fetch could be accomplished in one cycle, then only two memories would be needed.)

The Radar Signal Compender is composed of several Functional Modules (FMs), a Master Controller (MC), an Analog to Digital Conversion module (ADC), and suitable interfacing to a host computer, as shown in Figure 4. Data from the ADC is fed directly to the FMs which perform the data processing. In order to provide flexibility, the host computer can separately program each FM. Programming includes the setting of the Configuration Register (which specifies which data processing paths are to be used, thereby, determining the data word

BLOCK DIAGRAM FOR BASIC SIGNAL PROCESSING

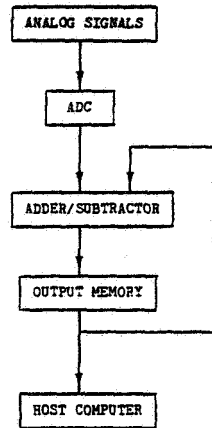


Figure 1.

BLOCK DIAGRAM FOR IMPROVED SIGNAL PROCESSING SPEED

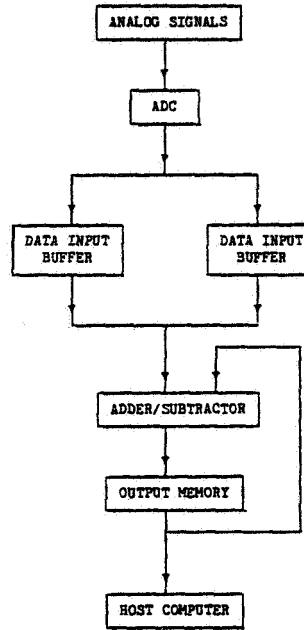


Figure 2.

size and whether or not the multiplier is to be by-passed) and includes the loading of the operands that control the addressing of the Data Input Buffers on the FMs. The latter is described, in detail, in a later section.

Data flow within one of the FMs is generally as illustrated in Figure 2 or 3 where each block may represent several stages in the pipeline. The processing pipeline is actually 9 or 7 stages long and uses either 23 or 18 cycles of the master clock, depending on whether the multiplier is used or by-passed, respectively. New data can be stuffed into the pipeline every 5 cycles of the master clock. The data paths can be up to 64 bits wide, or split up into as many as four 16-bit-wide paths for multiple independent parallel processing within each Functional Module. This feature is particularly useful in MST work where the extra guard bits are not needed. The Poker Flat MST radar will use the dual 32-bit-wide path configuration, while the incoherent-scatter applications use either the 48-bit or 64-bit configurations. For each configuration, the carry bits are appropriately propagated and any overflow conditions flagged.

The Functional Modules have been wired on 11" x 16" boards using a semi-automatic wire-wrapping service. Most of the 200-plus ICs on each FM either carry data or are part of one of the address busses. Since little space was left, much of the combinational logic required to control the FMs was placed in various PAL (Programmable Array Logic) circuits that must be specially programmed for the RSC.

An additional feature that had high priority in the design was the provision for automatic test features. Each of the memories (including the Data Input Buffers, the Base and Displacement Operand Memories, and the Output Data

BLOCK DIAGRAM FOR MORE GENERALIZED SIGNAL PROCESSING

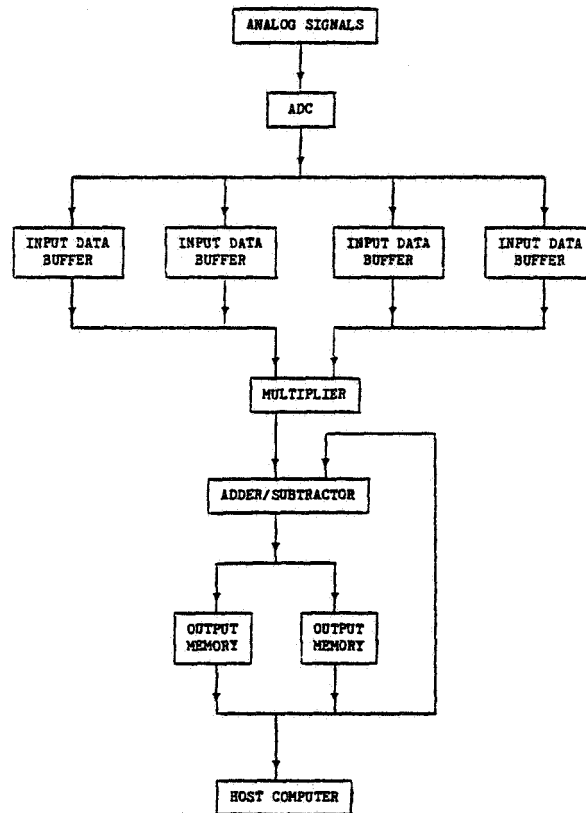


Figure 3.

Memories) can be loaded with test data from the host or MC and then read back out again to check memory and data buss integrity. Also, the multiphase clock can be single stepped to allow probing each stage of the pipeline.

The Master Control modules are somewhat dependent on the host to be used with the system. Differences arise from different I/O buss widths, handshaking, and the number formats (particularly in the integer to floating-point converters that are included). Control functions are generated and controlled by an on-board Z80 microprocessor.

#### ADDRESSING OF THE PROCESSOR INPUT BUFFERS

Although sequential addressing of the Data Input Memories is possible during raw data input from the ADCs, a random addressing scheme must be provided for reading the data back out for sample processing. For both pulse decoding and lag produce calculations (which are the most complicated cases) the addresses can be formed as the sum of two operands -- one based on a given sample referenced to a specific range, and the other determined as a relative displacement to the other samples that contribute to the calculation of the desired quantity for that range. This is simply a nested loop structure where the outer loop indexes the range and where the inner loop indexes the terms that

BLOCK DIAGRAM FOR A SYSTEM USING THE RADAR  
SIGNAL COMPENDER ANALOG SIGNALS

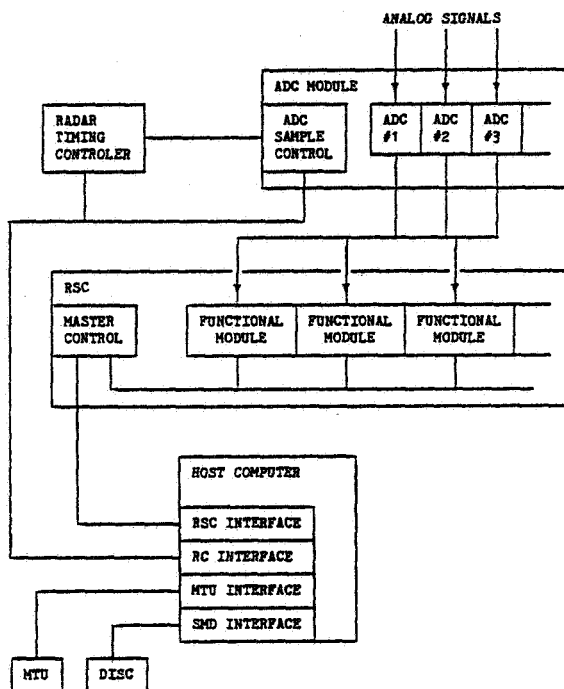


Figure 4.

contribute to that range. The Radar Signal Compendor obtains such Base and Displacement operands for the Data Input Buffers from sequential locations in three operand memories. Figure 5 diagrams how this addressing scheme is accomplished.

Each stage in the address computations is also pipelined to maximize the speed. (Other more general address schemes were not fast enough.) Address generation using the Base and Displacement operands is applied to one buffer of each Data Input Buffer pair for data processing while a separate counter provides sequential addresses to the remaining buffers for data input from the ADCs. Since the I/O busses, the Data Input Buffers and their addresses are all independent, no memory cycles are lost from the processing for the I/O transfers. Selection of the opposite buffer requires only a change in the state of a control line, a change that takes only a fraction of one microsecond to accomplish. Hence, the entire time is available for processing the data. This is a tremendous advantage over the situation in general-purpose processors which must give up memory cycles even for double buffered I/O. Separate Displacement operands are provided for the left and right Data Input Memories so that samples taken at different times can be selected for the multiplier to create the lag products of an autocorrelation function (ACF). Only the lower 11 bits of the Base and the two Displacement RAMs (which are 2 k words deep) are used for address generation; the remaining 5-bits are used for process and address counter control. Note that the Base Address Computer generates the address for the Base Operand Memory, while the Displacement Address Counter generates a common address for both Displacement Operand Memories.

#### CONCLUSIONS

The basic architecture of the Radar Signal Compendor has been illustrated with respect to the very specific high-speed real-time signal processing

BLOCK DIAGRAM FOR ADDRESSING INPUT DATA  
FOR SIGNAL PROCESSING

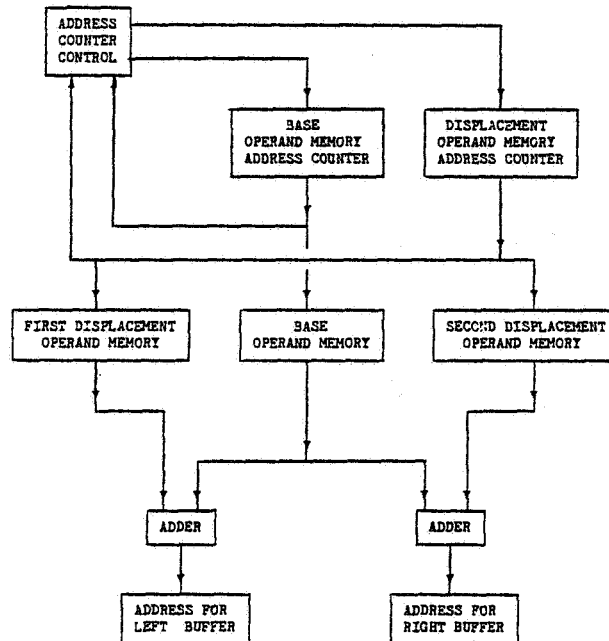


Figure 5.

requirements of backscatter radars. A full technical description will be available in the RSC users manual. The major features of the RSC are listed below.

- (1) Multiple Functional Modules provide many parallel data-processing streams, each of which is fully pipelined and programmable for maximum throughput and flexibility.
- (2) Multiple Independent Data Input Buffers allow processing to be completely independent of I/O.
- (3) Addressing is sequential for I/O with the RSC, but is flexible for processing within the RSC.
- (4) Integer processing is used with user selectable data path widths. Sufficient guard bits can be chosen to avoid overflows for even very long integrations; even so, error checking for overflows is provided.
- (5) Full multibit multiplications reduce biases and simplify the computation of weighting factors for off-line analysis.

Other uses of the RSC are envisioned. For example, since the Input Data Memories can be loaded directly from the host computer as well as from the ADCs, the device can also be used as an integer array processor for off-line analysis of much of our work that begins with Fourier transforms of large amounts of raw data. (This direct data load feature was originally developed for automatic testing of the RSC.) Other possible configurations have been considered where the output of one RSC was fed into another RSC for two-stage processing of the data. Eventually it may be desirable to substitute floating-point arithmetic units for the integer units where greater dynamic range is necessary for array manipulations. Note, however, that there is no reason to go to floating-point arithmetic for just the initial real-time processing of backscatter radar data.