A MICRO-PROGRAMMABLE CORRELATOR FOR REAL-TIME RADAR PROCESSING

by

Hans-Jørgen Alker
ELECTRONICS RESEARCH LABORATORY (ELAB)
Trondheim, NORWAY

ABSTRACT This paper presents a novel design of a multibit, digital correlator for incoherent scatter radar observations. By utilizing bit-slice microprocessor elements and internal control by microcode instructions, a high-speed arithmetical preprocessor has been developed. Arithmetical and control operations are separated to obtain increased speed performance. The arithmetical part is optimized for complex auto/cross-correlation processing and has an effective rate of $24 \cdot 10^6$ multiplications/sec. Input data has 8-bit accuracy in integer format. The processor includes, at input, a 2-ported buffer memory for storing single/multiple channel inputs. Processing results are temporarily stored in a 4 K word high-speed memory before transferred to a general-purpose computer. Processing speed can be increased by adding up to 3 slave processors thus achieving direct parallel processing. Internal hardware is available for interface with standard CAMAC modules.

INTRODUCTION

During 1976-79 a feasibility study of a digital, multibit correlator for the EISCAT (European Incoherent SCATter) radar system was conducted. The main topics of the study embraced system design and hardware construction of a digital processor which fulfilled the specifications for real-time data processing.

The result of the study is a prototype construction of a high-speed multiprocessor system under interactive control of the EISCAT radar site computer. Support software for microprogram development, simulation and hardware testing are available for execution on a host computer.

The main objectives for constructing a digital processor for the EISCAT system were the ultimate requirements for special purpose data handling algorithms and the real-time processing speed. Remote probing the ionosphere using the incoherent-scatter radar technique has earlier been limited by available computational power. The ionosphere acts as a deep fluctuating target, and modelling is based on estimation of a range-dependent complex autocorrelation function (ACF) derived from the

returned range-gated signal. Due to the low signal-to-noise ratios obtainable, real-time data reduction by time integration is required.

In the EISCAT radar the digital data acquisition system generates samples of the complex envelope (inphase and quadrature components in parallel) with typical sampling rates of 250 kHz for ion spectrum estimation. Assuming that typically 25 complex samples are contained in a range-resolution cell, the resulting requirement is $13 \cdot 10^6$ multiplications/sec. for real-time ACF processing. For reducing the time resolution of the radar experiment, the EISCAT radar utilizes multiple frequency transmissions and parallel channel processing in the receivers, increasing the requirement to about $104 \cdot 10^6$ multiplications/sec.

In 1976 a preliminary design of the EISCAT correlator was completed based on a structure using 50 hardware multipliers operating in parallel. The correlator fulfilled the requirements for multiple channel processing but had major limitations in relation to power consumption, system cost and hardware complexity.

This paper describes the architecture of a digital correlator system based on a redesign using an optimized time-serial approach for hardware construction.

1 DESCRIPTION OF CORRELATOR ARCHITECTURE

A block diagram of the correlator system is sketched in Figure 1. The hardware elements can functionally be split into two parts: The dual-channel data processing unit DPU and the control processing unit CPU. Both units are synchronized to a common system clock operating at 6 MHz.

The CPU generates all required control signals within the system. The main control unit is the microprogram sequencer which, at each clock cycle of the system clock, generates the value of the program counter (PC). This is the program memory location counter which maps the PC value into a microcode stored in the referenced location. With minimum hardware for instruction decoding, the microcode controls the function of the DPU on a cycle to cycle basis of the system clock. By feedback of the microcode to operate the sequencer, a predefined execution of a microprogram is possible. Two memory options are selectable for program source. With the RAM option (Random Access Memory), the correlator system is microprogrammable from an external source. Within the available program memory space, the RAM can contain several user-defined programs. With the PROM (Programmable Read Only Memory) option a library of predefined, non-destructable programs may be selected.

Each data channel of the DPU can be operated independently of the others. A data throughput rate of 6 Msamples/sec. is obtained by using a 5 stage "pipeline" in time-sequenced operation. The individual subfunctions of the DPU are given in Figure 2. The DPU is based on integer format arithmetic.

Input samples from buffer memory or external bus are 8-bits. The 4 hardware multipliers are operated in 8x8 bits and the final stage operates on 32 bits. Except for data overflow (hardware detectable), no rounding or truncation effects occur. The accumulators use the complete result memory as an accumulator register file.

The DPU hardware is especially designed for complex arithmetic processing. The buffer-memory is two-word addressable for transfer of values in complex notation.

Both address processors are designed with bit-slice elements enabling an effective base address/increment operation of the memories.

2 PROGRAMMING CONSIDERATIONS

An expanded block diagram of the microcode separation into "function"-fields are shown in Figure 3. The microprogram memory constitutes 2 pages with 64 words a 128 bits in each. A total number of 128 bits are used for all internal operations. The main processing task for the CPU is microcode generation (43 control lines) for controlling the data processing part, where individual operand/pipe-line registers can be strobed.

One 128 bit word of the program memory corresponds to single sample processing through all the stages of the data processing unit. This is achieved by individual delay of subinstructions of the microcode, and gives a more flexible programming of the system.

To save locations in the program memory and increase the execution rate of the microprogram, the system comprises 3 separately controlled hardware counters for program-loop execution (Figure 3). The microprogram sequencer contains 4 registers for storing return addresses when a subroutine call is executed.

Additional "background" operations have to be performed by the CPU. These operations involve control of the interfaces between the correlator and the host computer. The communication can be split into two basic operations: 1. Program load and parameter definitions from host. 2. Transfer of data from the correlator result memory to the host. Another interface is included for starting the correlator from an external trigger signal.

The control part of the correlator has an internal, programmable mode: STOP, which inhibits operation of the internal main clock. When transferring data on the host computer DMA channel, the internal clock is triggered for single instruction execution by the "data-received" signal from the host. This mode compensates for any speed mismatch between the host and the correlator.

The CPU has internal error detecting systems which mainly control numerical overflow of the DPU accumulators and the validity of external control signals.

During testing of the correlator prototype, the host computer programming of the correlator module was based on micro-instruction programming in absolute form (octal coding). In order to ease the programming tasks, a program CORSIM written in FORTRAN IV was developed on the EISCAT site computer (NORD-10). The CORRSIM is a multi-function software package for development and testing of user-defined microprograms. The system structure is shown in Figure 4. An editor is available for construction of programs and the submodule also includes a program source text generator after coding. The ARITEST and PROTEST modules give a simulation of the control and arithmetical part of the simulator at bit-level. The CORRIO module is the software driver for the correlator. Transfer out the correlator is via a standard CAMAC output module. The corresponding input from the correlator result memory uses a standard input module together with the CAMAC CDMA controller. A real-time program DMA-READ takes data from the driver and transfers it to the CORRSIM program. As given in Figure 4 several fixed-file programs are stored within the program. These are hardware test programs and are executed at regular time-intervals.

In the correlator module a PROM module containing a fixed sample set is located at the input of the DPU. This predefined sequence can be transferred to the DPU for processing and then transferred to the computer where a table comparison can be made with the same sequence through the software simulator. This option has been valuable during the final stability tests of the prototype.

3 HARDWARE CONSTRUCTION

The prototype is realized in LSTTL and NMOS technologies. The system is housed in a standard 19" rack section and the power consumption is about 100 Watts at 5 Volts. The number of cards is 8 (double European standard), where the control module requires 5 and the arithmetical part 3.

CONCLUSIONS

A microprogrammable system for real-time radar processing is described. The arithmetical part is optimized for complex processing algorithms.

Considering the correlator operational modes in the EISCAT system, the proposed structure makes it possible to accomodate most of the data produced in dual channel radar experiments. Provisions have been made for hardware expansion. In direct parallel processing only DPU parts need to be added, on advantage given directly by the microcode structure.

DISCUSSION

R. Seynaeve What was the manpower allocated to the project and how long did it take?

H.J. Alker Prototype design and construction has taken two years and involved a small group of two/three engineers.

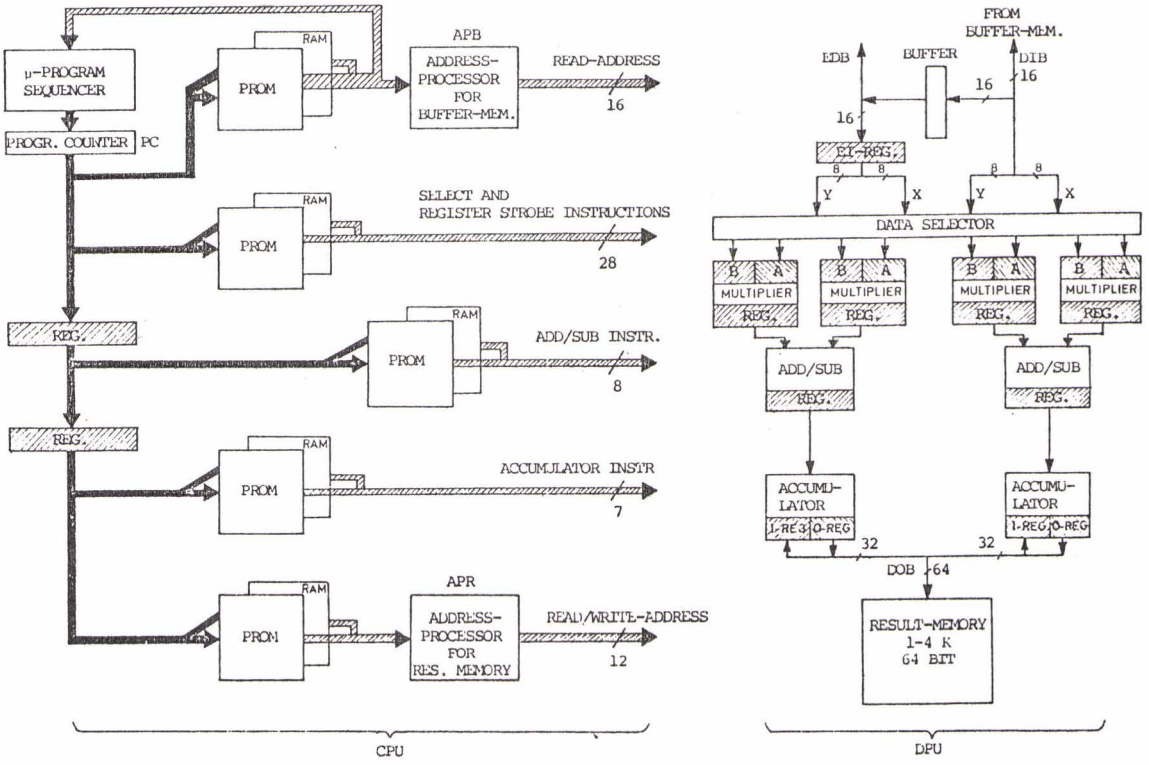

FIG. 1 CORRELATOR ARCHITECTURE

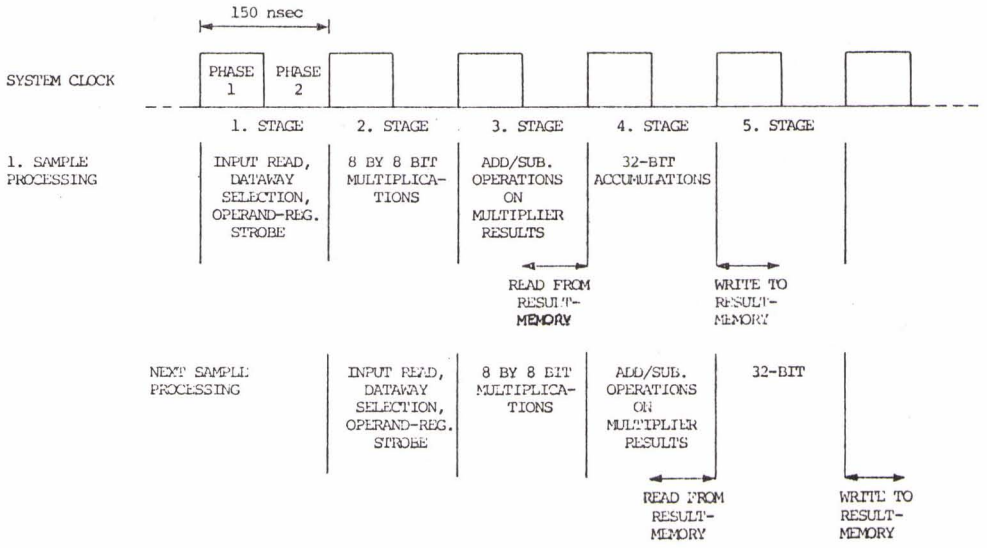

FIG. 2 TIME-SEQUENCED OPERATIONS OF THE DPU

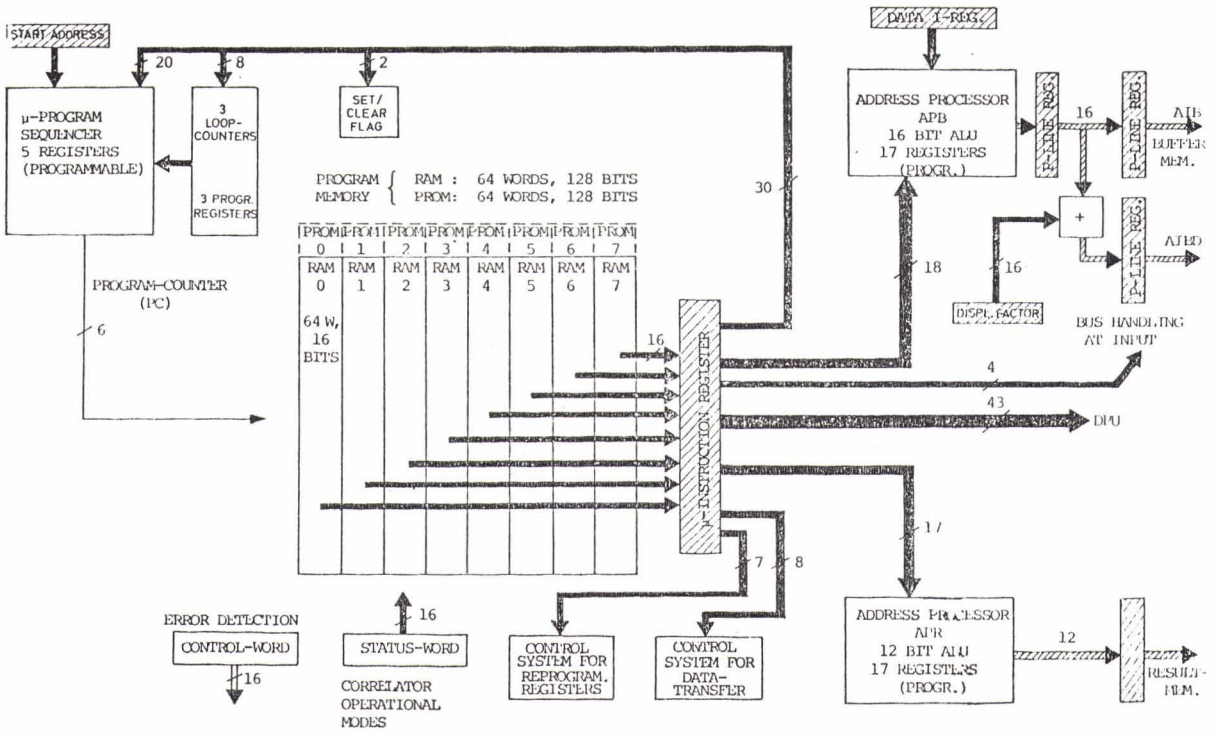

FIG. 3 BLOCK DIAGRAM OF CPU HARDWARE

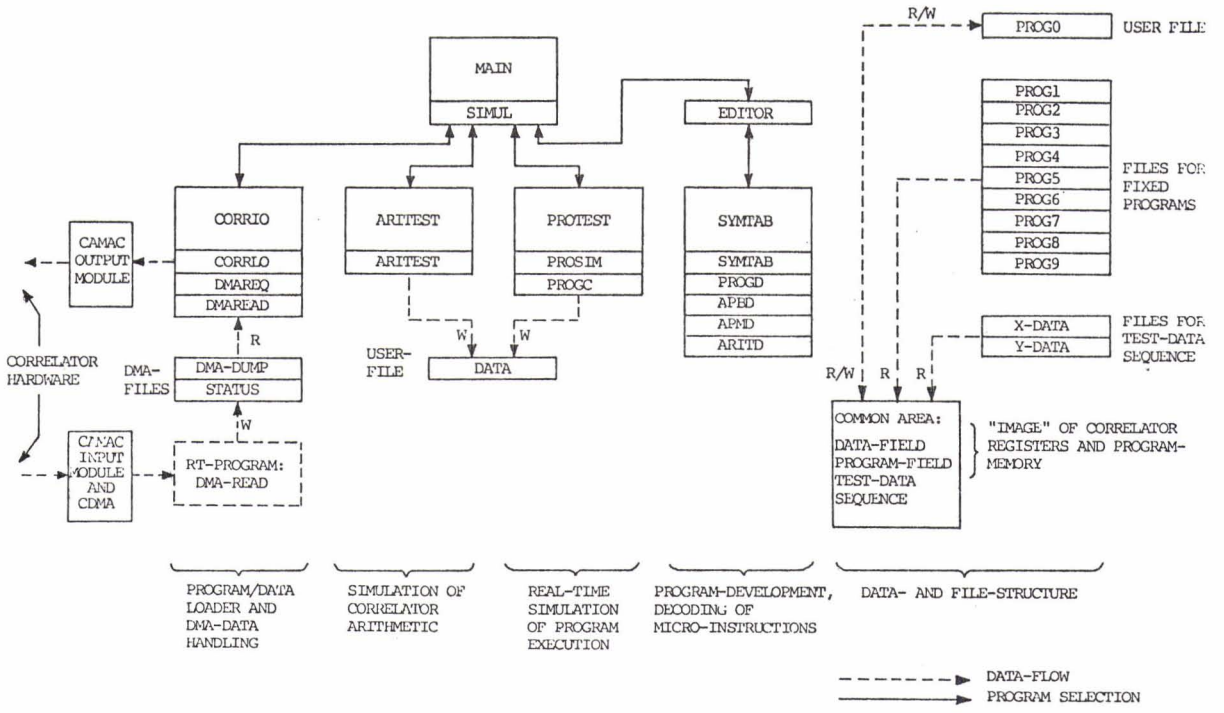

FIG. 4 CORRSIM STRUCTURE