# TRENDS IN AEROSPACE COMPUTERS

Paper No. 77-1379

R. L. Carberry
IBM Corporation
Federal Systems Division
Owego, New York

## Abstract

Within the militarized computer disciplines, several significant trends have emerged within the recent past which have substantially affected both the development and application of computers to the aerospace environment. These include governmental standardization in most major areas of computer design including architecture, I/O, programming methodology, and packaging. Additionally, the application of computers into systems have come under the affects of governmental standardization through the use of data buses. Programming languages are beginning to experience significantly increased use despite some initial concerns about efficiency. From a technology perspective, the broad use of custom LSI has not yet impacted computer development; however, the use of devices such as microprocessors and Programmable Logic Arrays (PLAs) have received an increasing amount of use and will probably be the interim vehicle between current medium scale integration (MSI) and the eventual use of custom LSI. In this paper the trends in these miltary computers will be examined by briefly considering the direction of each of the areas indicated above. Additionally, because of their association with the general purpose militarized computer, signal processors will also be briefly examined. Signal processors are a class of general purpose processor which have been tailored to a specific application in order to achieve substantially higher performance over a limited class of applications. As such, the signal processor is usually controlled by a general purpose (GP) processor or is organized as an "attached processor" and is architecturally viewed as simply another CPU engine in a larger computing system.

## High Order Language (HOLs)

In the recent past, computer requirements have been tending to include the need for high order languages (HOLs). There is a continuing debate as to the benefits of the use of HOLs versus their claimed inefficiencies. The author does not intend to resolve such a debate within this paper but rather to indicate some considerations which one must be sensitive to when using HOLs and to try and indicate the forward trend.

The principal attribute of HOLs over assembly language is that the HOL presents an interface to the programmer which is much like English text and is at a much higher semantic level than assembly language. Such an advantage may be apparent when developing code in that fewer lines of code need be written for a given function (this implies greater programmer productivity). Additionally, from a software maintenance perspective, maintaining an HOL-based software system is claimed to cost less because of the English text-like format of HOLs, and consequently, their tendency to be self-documenting. From a training viewpoint, once a programmer knows a given HOL, they can write programs in that language independent of the eventual host machine architecture. On the negative side, opponents to the use of HOLs claim that even though a program is written in an HOL, the debug of that program will ultimately be done at the assembly language level. Consequently, the gains for use of an HOL are limited to coding, which is a relatively small percentage of the total development cycle. Additionally, a key consideration in the use of an HOL is the efficiency of the HOL compiler. An inefficient compiler can have an extreme negative effect on the amount of memory storage required, and consequently, have a negative effect on machine performance and reliability.

While the HOL controversy is not resolved at this point in time, it is clear that the trend will be toward increased use of HOLs. Because of their relatively recent widespread application, there is no universal standard HOL. However, each of the major user groups or services are standardizing within their areas of influence. The Navy is standardizing on CMS-2; the Air Force on JOVIAL, and NASA on HAL. Eventually there may be a broad-based standard such as DOD-1. Additionally, in each of the computer development-related areas of test and support software host languages there may also be standard HOLs.

## General Purpose Processors

For the purposes of this paper, militarized general purpose processor will be defined as a moderate performance (e.g. 500 KOPS), nonvolatile memory (e.g. 64K x 18b of core memory), single CPU engine packaged in a volume consistent with ATR specifications (e.g. toll ATR or one-half ATR). The need for this type of definition is to restrict the scope of discussion and to eliminate, for the moment, processing systems made up of collections of microprocessors or very large multiprocessing systems. These areas will be addressed later in this paper.

The GP machine of this class is currently receiving a considerable amount of attention from a governmental standardization viewpoint. While the focus is on standardization, the exact definition of the level of standardization is not yet agreed to. There is a level of standardization which holds that the standards should be extended to the major subassembly or group of subassemblies level. This implies that a set of standards for a GP machine would consist of CPU modules, memory modules, I/O, power supply, and structure. This implies a level of maintenance and second-source procurement at this major subassembly level.

An alternate approach to the standardization level would be to standardize at the box level. This would imply a standard box form factor such

Copyright © American Institute of Aeronautics and Astronautics, Inc., 1977. All rights reserved.

as full ATR, standard I/O interface at the box level such as a 1553 data bus, and a standard software or architectural interface. Such a standardization approach would potentially involve box level maintenance such as is used with a Reliability Improvement Warranty (RIU).

Within either approach to the level of standardization, a key question is: "What should be the software standard interface?" In the case of the Navy AYK-14 computer, the selected interface is the UYK-20 assembly language. The Air Force DAIS program has selected a HOL interface in the form of JOVIAL. The joint Army-Navy Military Computer Family (MCF) originally focused on the PDP-11 architecture, but is currently considering an approach involving multiple assembly language interfaces. In general, it appears that when an assembly language interface is selected as a standard, it is because there is an existing assembly language with a substantial support and application software base which must be supported. With time, as the existing software systems are updated via HOLs, then the HOL interface may become the standard.

## Data Bus Systems

The government standard which is having an affect on both the CPU and the entire system integration philosophy is the 1553 Data Bus. Data bus systems have been flown and proven on several systems such as B-1, F-15, F-16, and will be on most newly developed aircraft. The attribute of 1553 which makes its affect significant, is that 1553 is becoming an operational standard. The 1553 definition extends beyond the bus and bus protocol. 1553 includes several fundamental system architecture concepts and implementation approaches. Included in the 1553 architecture is the concept of master-slave relationship between the principal processor (or Bus Control Interface Unit - BCIU) and the controlled subsystem (or Remote Terminal Units - RIUs). Concepts for bus and system redundancy are included although not specified in detail. The standard bus concept will continue to be fundamental to avionic systems in the future.

1553 does, however, tend to be technology dependent in that most implementations are valid for 1 MHz serial, wire bus structures. The bus technology may be trending toward fiber optic implementations where many of the 1553 definitions do not apply. Consequently, the 1553 definitions must continue to be detailed out and extended for new technology. There are other bus structures such as MIL-STD-488C and RS-232 which are also becoming popularly used by the government in areas such as test equipment interfaces and in applications such as shipboard systems.

## Microprocessor

Perhaps the area which has seen the greatest dynamics in the past one to two year period is that of "microprocessors." The definition of this GP processor is not at all precise. The literature includes in this category 8-bit devices such as the INTEL 8080, 16-bit devices like the TI 9900, and bit slices such as the AMD 2900. While one can collectively speak of all these types of devices as microprocessors, the application areas to which the devices are applied cause the user to differentiate carefully between the various functions and performance ranges of "microprocessors."

There has been an extremely broad-based application of microprocessors in areas such as I/O controllers of CPUs, intelligent terminals, display processors, and generally, in the area of imbedded processors. Usually, a microprocessor is used as a replacement for a collection of random logic performing a dedicated function. The advantages of a microprocessor over random logic are nearly self-evident: ability to change or increase function with minimal hardware impact; high circuit and functional density in a small area; multiple applications of the same part number devices over a wide range of applications.

## Signal Processors

Signal processors are a form of general purpose processor which has been optimized to support signal processing functions such as FFT and digital filtering. Because signal processors are programmable, all the issues relating to GP processing are equally relevant to signal processors. There are three key elements effecting the design of a signal processor: (1) the algorithms supported, (2) the architecture of the processor, and (3) the programming systems hosted by the processor. Each of these is briefly discussed in terms of the current state-of-the-art and trends.

The transform is basic to all of signal processing. The Fourier transform is the principal time to frequency domain transform. Once in the frequency domain functions such digital filtering and power spectrum analysis can be applied to extract signature information for example. Computationally, to date, the most efficient transform has been the Fast Fourier Transform (FFT). Recently, new results using an integer ring transform approach which, for the class of applications to which it applies, has a more efficient computational characteristic than the FFT. These results are stated in terms of complex multiplies and adds and are summarized in Table 1.

Architecturally, the trend in signal processors is toward a control function with one or more arithmetic unit. The arithmetic units are structured typically in some form of "pipeline." Pipelined architectures are characterized by having several functional sections where each section performs a different subfunction and the total function is achieved sequentially passing data from subfunction to subfunction. This type of architecture is illustrated in Figure 1.

Programming signal processors is following many of the trends previously established in the solution of GP problems. Previously, the bulk of signal processor programming was written in various forms of assembly languages. The rationale for use of the assembly language was twofold: (1) a strong concern for programming efficiency, and (2) no standardization or wide acceptability of a single high order language (HOL). The current trend is toward the increased use of HOLs. Specifically, within Navy application, the use of SPL-1 signal processing language is increasing in application.

TABLE 1

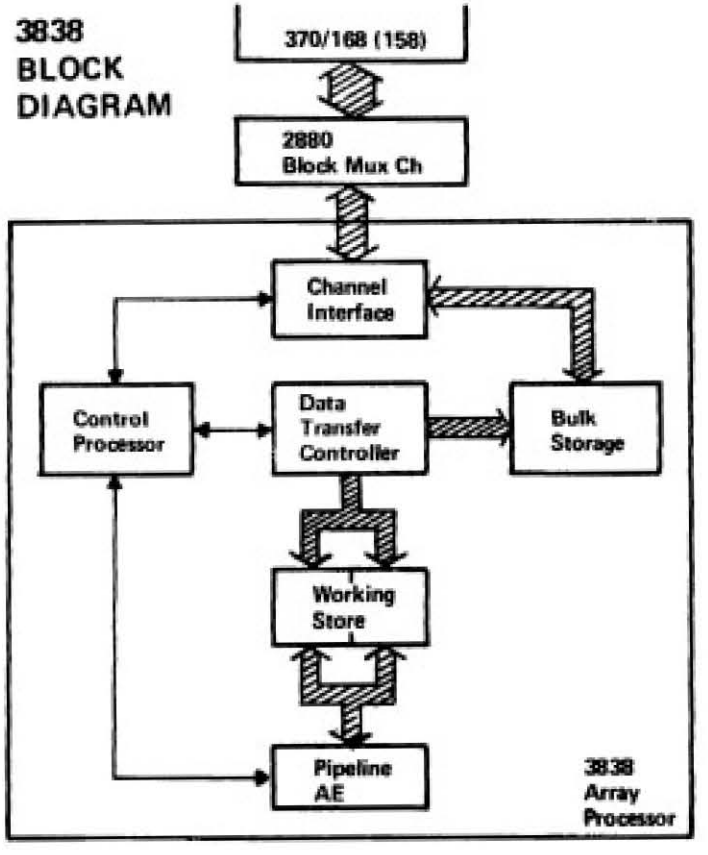
NEW INTEGER RING SPECTRUM METHODS
HAVE SIGNIFICANTLY REDUCED COMPUTATION
FOR CONVOLUTION AND CORRELATION

| Convolution Method | Number of Complex Multiplies | |
|---|---|---|
| | Generally | N = 1,024 |
| Direct Time Domain | $N^2$ | 1,048,576 |
| Fast Fourier Transform | $\frac{3}{2}N \log_2 N + N$ | 16,384 |
| Integer Ring Spectrum | $<N$ | $<1,024$ |

IBM

GAG3-21A

FIGURE 1




IBM

6A63-26