CR ~~86~~ 86001

Honeywell Document 12017-FR2

PHASE II FINAL REPORT

SPACEBORNE MEMORY ORGANIZATION

AN ASSOCIATIVE DATA ACQUISITION SYSTEM

By Dale C. Gunderson, Charles Wm. Hastings, and H. Randolph Holt

December 1966

Prepared under Contract No. NAS 12-38 by

Honeywell Inc.
Systems and Research Center
Minneapolis, Minnesota

Electronics Research Center
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

FOREWORD

This is the final report on Phase II of Contract NAS 12-38, Spaceborne Memory Organization. This work was performed for the National Aeronautics and Space Administration, Electronics Research Center, Cambridge, Massachusetts, during the period April 1966 to December 1966. The NASA/ERC technical monitor for this project was Mr. Neil Patt.

The authors wish to acknowledge assistance from coworkers Gregory J. Prom on the analog to digital conversion problem and on the design of the strand logic, and Ralph Livingston and William Eckart of the Computer Sciences Department of the Aeronautical Division on the simulation investigations. We also wish to thank Neil Patt for the technical direction he provided and for providing the data used in the simulation effort.

TABLE OF CONTENTS

ILLUSTRATIONS

TABLES

SECTION I
INTRODUCTION

The data acquisition task on a space vehicle is that of collecting data from a large number of data sources and of performing the necessary conversion and processing operations to allow efficient transmission to earth or to some other space vehicle. This task is taking on increased importance on space exploration missions as scientific payloads become more elaborate and as the communications problems become more severe due to the increased distances involved. The objective of this project was to investigate a new approach to the organization of a spaceborne data acquisition system

The first portion of the project was aimed at defining the specific requirements of a data acquisition system. While the details of this effort are found in Section II, a few brief comments are in order. Some characteristics of the problem that increase the difficulties in obtaining a simple solution are the relatively large numbers of data sources involved (in the hundreds); and that these data sources are producing outputs in various forms that must be sampled at various rates. For example, as a vehicle moves from one phase of a mission to another or from one situation to another, varying subsets of the data sources, particularly the scientific sensors, must be enabled. Another important fact is that considerably more data is being collected than can be transmitted. To summarize, the primary functions of such a system are multiplexing to provide flexible selection of data sources, analog-to-digital conversion to change analog inputs to digital form suitable for processing and transmission, and data compression to eliminate the redundant data points prior to transmission.

A system based on the use of an associative memory is proposed for the data acquisition task. The associative memory has the capability of performing processing (arithmetic) operations, as well as the search operations normally included in an associative memory, and has the capability of performing any one of these operations simultaneously over all stored words. Since in this application the processing capabilities of the associative memory are emphasized, it will be called an associative computer throughout the remainder of this report.

An associative computer is particularly well suited to handle the functions of data acquisition because the shape of the computer is like the shape of the problem; that is, a computer consisting of many partially independent units operating in parallel is well suited to monitor and process many independently originating streams of data. The approach taken is to connect one data source to each "strand", the name given to the basic building block of the associative computer. Each strand has the capability of storing a single word and of performing certain, read, write, search, and arithmetic operations. Thus each strand of associative computer serves as a processing unit for the connected data source.

The search capabilities of the associative computer are used effectively to perform the required multiplexing function. Searches over special tag fields of each word can identify or select subsets of data sources for each "data acquisition cycle". Such an approach to multiplexing is highly flexible since the contents of tag fields can readily be changed.

The processing capabilities of each strand of the associative computer are used to execute data compression algorithms simultaneously for all data sources. The third major function, the analog-to-digital conversion, is also built into the associative computer so that all analog data sources are simultaneously converted.

While the associative computer performs all three major functions of the data acquisition problem, a general-purpose (GP) computer is included in the system to serve as a master controller and to provide program memory for the associative computer.

The details of the system design summarized above are contained in Section III. The operating procedures, as well as a number of somewhat special applications, are included in Section V.

Another major effort on this project was to evaluate the use of data compression algorithms modified to better fit the special capabilities of the associative computer. The advantage of the modified approach versus the conventional approach is that a hard limit can be placed on the number of data points acquired during each acquisition cycle. This then allows the data acquiring rate and the transmission rate to be matched, thus eliminating the need for buffering except in special situations. The results of this evaluation, which was done by simulation using real data, are given in Section IV.

SECTION II
THE DATA ACQUISITION PROBLEM

The data acquisition problem on a space vehicle is essentially that of collecting data from a large number of sources, which are producing outputs of various types and at various rates, and preparing this data for transmission to earth or to other space vehicles. To obtain specific requirements of the spaceborne data acquisition system, a study of space mission requirements and of the characteristics of typical space vehicle instrumentation systems was performed. Of particular interest are the properties that space vehicle instruments exhibit with respect to computer tie-in. While very little material dealing with this specific problem is available, References 1 through 5 were found to be helpful in performing this investigation.

Before getting into the details of the data processing requirements for specific instruments, it is well to consider the types of space vehicles and missions necessary to carry out the space exploration task in the 1975 to 1985 time period. There are many possible types of space missions. Space vehicles may reasonably be subdivided into the following categories: space probes, flyby probes, orbiters, impacters, landers, lander-rovers, rovers, and shelters. In planetary exploration, one may distinguish the following more or less spherically concentric separate zones of exploration for our own planet, and most of them would apply equally to any other planet: endosphere (core); biosphere (life forms); atmosphere; electro-magnetosphere (belts of charged particles); gravisphere (region of significant gravitational influence); and solar system (everything else, for the time being, until the first interstellar probe goes up).

Vehicles of the same type may still be used for widely varying purposes. An orbiter, for instance, may be intended to map the planet surface, to predict weather by observing the cloud cover, to investigate gravitational anomalies, or to investigate the density and nature of charged particles at various altitudes.

The order in which the various types of space vehicles are listed is approximately that in which they would be used to study a previously unvisited planet. Also, the longer-range vehicles are more apt to be multiple-purpose; it is now easy enough to put up earth orbiters that some of them are single-purpose, but it would not be reasonable in the near future to plan on using a Mars orbiter for just one experiment or class of experiments.

The data rates attainable between a space vehicle and an earthside tracking station fall off roughly according to the inverse-square law. Hence, there are many decimal orders of magnitude of difference between the data rates of earth orbiters (10^6 or more bits per second in some cases) and those of long-range probes (sometimes as small as 1 to 5 bits per second). This has a direct effect on the spaceborne data processing system requirements.

1. CLASSIFICATION OF INSTRUMENTS

There is no particular unifying principle which one can state about all space vehicle instruments, any more than there is about all instruments in general; space vehicle instruments are a very heterogeneous lot. They exhibit a profuse variety of purposes, operating modes, output formats, and idiosyncrasies.

The following information comprises roughly what a computer design engineer would like to know about the complement of instruments aboard a space vehicle:

- The form of the output reading (digital, analog, random pulse, synchronous pulse, etc.).

- The equivalent binary precision (how many bits are needed to adequately represent the reading).

- The minimum sampling frequency (the inverse of how long the instrument may be left unread).

- The mode of operation (continuous, continuous at certain times, intermittent, occasional, one-shot).

- The delay (if any) between the receipt by the instrument of a command to supply a reading and the availability of that reading.

- Any relevant special characteristics (such as "monotonicity" - certain devices present their readings as counts, and these counts usually always increase monotonically and never decrease).

Also, a space systems engineer would like to know one more thing;

- How much digital or analog electronics of a basically computational nature has been included in the "instrument" package, and could be eliminated at a saving in size, weight, and power consumption now that the instrument is being connected to a full-scale associative computer or GP computer.

The precision and sampling frequencies could if necessary be given in terms of ranges without too much loss of information. The computer design engineer does not necessarily even need to know what the instruments are, but only needs to know their characteristics when they are viewed as data sources, and can as well treat them collectively as one-by-one. Thus it may suffice to state, for instance, that there are X analog instruments whose outputs must be converted to 10-bit precision and which must be sampled at least every 20 milliseconds, and so forth. One useful approach, taken in Reference 1, is to treat instruments for far-future missions in terms of the basic properties of the measurements to be taken rather than of the instruments now in existence, since whatever instruments are invented to take those measurements will have to tailor their precision, sampling frequency, and so forth to whatever is appropriate for the measurement.

Gathering this sort of information about space vehicle instruments proved to be a difficult task. The difficulties which stand in the way include the following:

(1) The development of suitable space-vehicle instruments sometimes lags the formulation of mission requirements by several years, and radical departures from contemporary instrumentation are often called for. Hence, the characteristics of many space-vehicle instruments are not predictable with much confidence over the time period 1975 - 1985, particularly where there are competing instruments with radically different data formats (the spectrograph and the chromatograph for atmospheric composition analysis, for instance).

(2) The capabilities and needs of space-vehicle data processing systems have not discernibly influenced the design of space-vehicle instruments yet, and such influence would be desirable. When an instrument is to be sampled by a computer capable of data compression, the most convenient type of instrument is one which can be sampled without a significant delay at an arbitrary time (i.e., "continuously"), and which normally produces an output consisting of one or two pieces of numeric data of the precision (i.e., "word length") taken as standard for the computer. It makes no sense in this context to speak of how many "bits per second" the instrument produces, since that depends on how often it is sampled; yet this figure is the only data-processing-type parameter given in many reports for most instruments, and the inference is that a sampling rate has been

chosen in advance (and possibly even in some way built into the instrument) to provide for a minimum loading of the space-to-ground communications system consistent with adequate earth-side reconstruction of the time history of the variable read by the instrument. If now it is assumed that a computer capable of data compression is available aboard the space vehicle, such an advance choice becomes not only unnecessary but undesirable; the choice should instead be made adaptively in real time by the computer. In some cases, these considerations may mean some simplification of the instrument; in others, they would favor one type of instrument (such as chromatograph) over another (such as a spectrograph) because the first adheres to a standard data format and the second does not.

(3) The data rates for long-range space exploration vehicles are often assumed to be very low (8 - 10 bits per second). On the other hand, the data rates for satellites in orbit about the earth are often quite high (many kilobits per second). There is a strong motivation for using data compression in each case: primarily to conserve the power needed for data transmission in the long-range case, and primarily to avoid overwhelming the earth-side computer center with masses of redundant data in the earth-orbiter case. Also, one might reasonably expect that the assumed data rates for the exploratory vehicles assigned to any given celestial body will increase with time, as the nature of the exploration of that body changes from an initial contact to a detailed mapping and examination. Hence, it is not at all reasonable to assume a "typical" space vehicle data rate, since the variation is too great. It may be reasonable, however, to choose as "representative" some rate characteristic of a vehicle doing a detailed study of, say, the Martian surface.

The following classification scheme for space vehicle instruments may seem excessively general, but it definitely appears feasible to apply, and it still is of value to the computer designer:

(A) High-speed or variable-speed continuously readable data sources

(B) Low-speed continuously readable data sources

(C) Data sources which can be read only after a certain delay following their receipt of a command

(D) Data sources corresponding to experiments which are only done once or a few times during the mission

(E) Anomalous data sources which present a non-numeric interface

Instruments of classes A and B would be suitable for direct connection to an associative computer. However, all of them might not be connected to the same associative computer, because of the vast difference in sampling rates between, for example, a television camera and a thermocouple which indicates the temperature of the computer memory container.

Instruments of classes (C) and (D), referred to in Reference 8 as "on-demand" instruments, require a fundamentally different sort of servicing by the data processing system than do those of classes (A) and (B). These instruments need to be activated and read at certain well-defined and isolated times, and hence for them the associative computer functions as an "alarm clock" featuring multiple future alarm settings as described in Reference 8.

Instruments of class (E) could be connected to the associative computer, but it would not be particularly profitable to do so. A reading from a class (E) instrument would probably just have to be transmitted intact to the GP computer, or else processed in a mode unique to that instrument while most of the associative computer strands did nothing, and hence it is just as well to connect such instruments directly to the GP computer to begin with.

Certain instruments can reasonably be connected to the associative computer, but their output readings require special treatment because they consist of multiple pieces of data. For instance, a "scintillator" produces a reading consisting of a pair of data ΔE and $E - \Delta E$; the first of these is usually plotted versus the second, rather than plotting either of them versus time.

Also, sometimes the output of several instruments can be processed to yield an estimate of the output of some other instrument for calibration and verification purposes. Such verification computations would fall to the GP computer to perform, rather than to the associative computer.

2. FUNCTIONS OF THE DATA ACQUISITION SYSTEM

The study of space missions and instruments has shown that a spaceborne data acquisition system must perform three major functions: multiplexing, conversion and compression. A number of conclusions regarding specific requirements for each of these functions are given below.

a. Multiplexing

Multiplexing is the function of providing a path or connection for purposes of collecting data from all data sources. The multiplexing function includes the requirement for selecting various subsets of data sources and for accepting data from each source according to the sampling rate requirement of that source. Whereas the number of data sources aboard a space vehicle may range from a couple of dozen to several hundred, it appears that 200 - 300 is a generous amount for most missions. Thus for purposes of sizing the associative computer, 256 data sources are assumed. This number is not critical, however, due to the modularity of the associative computer. Appreciable changes in the number of data sources will not make the system design described here inappropriate.

b. Data Conversion

The main reason for the need of data conversion is the fact that data sources are producing outputs of various kinds. The outputs of some are in analog form, others produce random pulses, and still others produce digital numbers coded in various ways. All data points must be converted to a digital form suitable for arithmetic computation and for transmission. The data acquisition system must be capable of performing all these conversions as well as other signal conditioning type functions.

c. Data Compression

The terms "data compression" and "data editing" properly refer to any systematic technique for extracting, from a large number of sensor readings, the "most significant" information. The objective is to reduce the total amount of information which must be retained, or transmitted, to as close as possible to the minimum which will represent the time history of the sensor variables to a desired accuracy.

There are a number of reasons why data compression techniques are becoming attractive. The amount of rocket fuel required to put a satellite or space vehicle outside the earth's gravitational field is very great, and its cost is obviously minimized by keeping the weight of the vehicle as low as possible. In the past, this consideration has led to aerospace computers being designed as extremely rudimentary devices, lacking many of the programming conveniences taken for granted even in low-cost ground-based computers. The advent of integrated circuits, however, changed the economics of aerospace computer design to a degree where such an extreme approach is no longer necessary; the most recently designed aerospace computers now have many of the worthwhile programming features long familiar to users of ground-based scientific and real-time computers. Integrated circuits have a very high physical packing density, consume very little power, and improve the reliability picture over what it was with discrete semiconductors. Substantial, although less spectacular, progress has also been made in the field of aerospace digital memories. All in all, it is no longer prohibitively expensive to put an appreciable amount of computing power aboard an aerospace vehicle.

At the same time that the economic penalty is decreasing for the inclusion of very powerful computing devices on board an aerospace vehicle, the penalty is increasing with respect to the amount of data which can be transmitted from the vehicle back to ground during a mission. One reason is that the power required to transmit data from a vehicle back to earth is considerable in terms of the power supplies available. Moreover, as the vehicle ventures further from the earth, still more power is required to produce a given level of signal-to-noise ratio at the receiver on earth, and this problem will be vastly aggravated on interplanetary and interstellar missions. The second major reason is that the number of sensors included aboard vehicles is subject to rapid increase as more and more experiments are thought up and more and more previously ignored factors are taken into consideration. The sheer amount of data being transmitted is already such as to stagger not only the transmission capabilities of the satellites, but also the capabilities of the computation centers on earth (at, for instance, NASA-Goddard and JPL) to record, "decommutate"*, and analyze the data. NASA-Goddard, for instance, has to analyze about a trillion such pieces of data per year, or roughly 30,000 per second.

*Sensor data is "commutated" into a complicated sequential format for transmission. The process of unscrambling this format so that the data for each given sensor are all together, with a post-reckoned approximate time of sampling for each datum, is referred to as "decommutation".

This data traffic jam would be more easily tolerated if all of the data were truly necessary. However, much of it is "redundant" in the sense that the time history of the variables of interest can be known adequately without it. A given sensor may exhibit very little change for a period of days or months; although the fact that the sensor is quiescent is necessary information, sending a great number of unchanged sample readings is an inefficient way of communicating this fact. Also, there may be reasons why the outputs of some sensors are not felt to be important during certain phases of a mission.

To sum up, the time is approaching when it will be feasible to put very powerful computing systems aboard relatively small space vehicles, but it will not be feasible for these space vehicles to transmit back to earth more than a minute portion of the data gathered by the vehicle's sensors, and it will not even be economical to require that the receiving stations and data processing centers on earth cope with the full amount of data which could be gathered by the vehicle's sensors. Hence, both the capability which can be put aboard a vehicle to do data compression and the need for this capability will be subject to very rapid increase as integrated circuits get smaller and smaller and space missions get longer and longer.

There are a large number of techniques available to perform data compression. The only ones considered here are those which involve a relatively simple decision-making process, based on the data themselves or on simple quantities readily calculated from the data. The specific data compression algorithms used are described in Section IV. It is also necessary to employ certain filtering, averaging, or simple tests to eliminate wild points from the data prior to use of the data compression algorithms. These techniques are also described in Section IV.

SECTION III
SYSTEM DESIGN

1. DESIGN PHILOSOPHY

A system design based on the use of an associative computer is proposed to handle the functions of data acquisition. An associative computer with its repetitive structure is well suited to perform the required functions simultaneously for all data sources. A data source is permanently connected to each strand, the basic building block of the associative computer. The search capabilities of the associative computer can be used effectively to perform the multiplexing function and the processing capabilities are used to execute data compression algorithms. The analog-to-digital conversion capability is also provided in each strand serving an analog data source.

A typical acquisition cycle includes selection of strands to take part in the current cycle, readin and conversion of data from data sources connected to selected strands, compression of the data to eliminate redundant data points, and output of significant data points. The outputs do not go directly to the transmitter but instead go to a general-purpose computer which also serves as a master control for the system.

In performing the design of this system a number of guidelines were followed. These were increased reliability, simplicity and flexibility of programming, reduced power consumption, and applicability to advanced component technology.

The cellular or repetitive structure of the associative computer helps achieve reliability gains and also is amenable to batch fabrication techniques. Where possible, centralized hardware has been minimized, at the cost in some cases of more complex per-strand logic. This is believed to be a desirable tradeoff, particularly in view of the capabilities of large-scale integration (LSI). The number of connections between strands, as well as between other units of the system, has also been minimized, again at the cost in some cases of increased complexity within the strand.

In addition to the integrated circuit technology, the system design will allow the use of advanced NDRO memory elements in the strand memory. This will not only contribute to system reliability but also will reduce the power consumption.

The aim for simplicity and flexibility of programming is achieved by using a micro-programming approach, the details of which are described in Subsection III-3.

2. SYSTEM ORGANIZATION

The Associative Data Acquisition System (Figure 1) consists of two major parts--an associative computer and a general-purpose (GP) computer. The associative computer provides the capability of monitoring and processing the inputs from many data sources, and the GP computer performs storage and control functions.

The specific functions of the GP computer include: program storage for both associative computer and GP computer instructions, storage for the acquired data, and over-all system control. To perform these functions, data paths are provided from various points in the associative computer to two GP computer registers--the data register and the memory address register--as shown in Figure 1.

Access to the memory address register of the GP computer is required to allow the associative computer to obtain instructions from the random-access memory and also for the interchange of data between the two computers. The associative computer has its own sequence counter which shares access to the memory address register with the sequence counter of the GP computer. Thus the associative computer can sequence through its program by stealing memory cycles from the random-access memory.

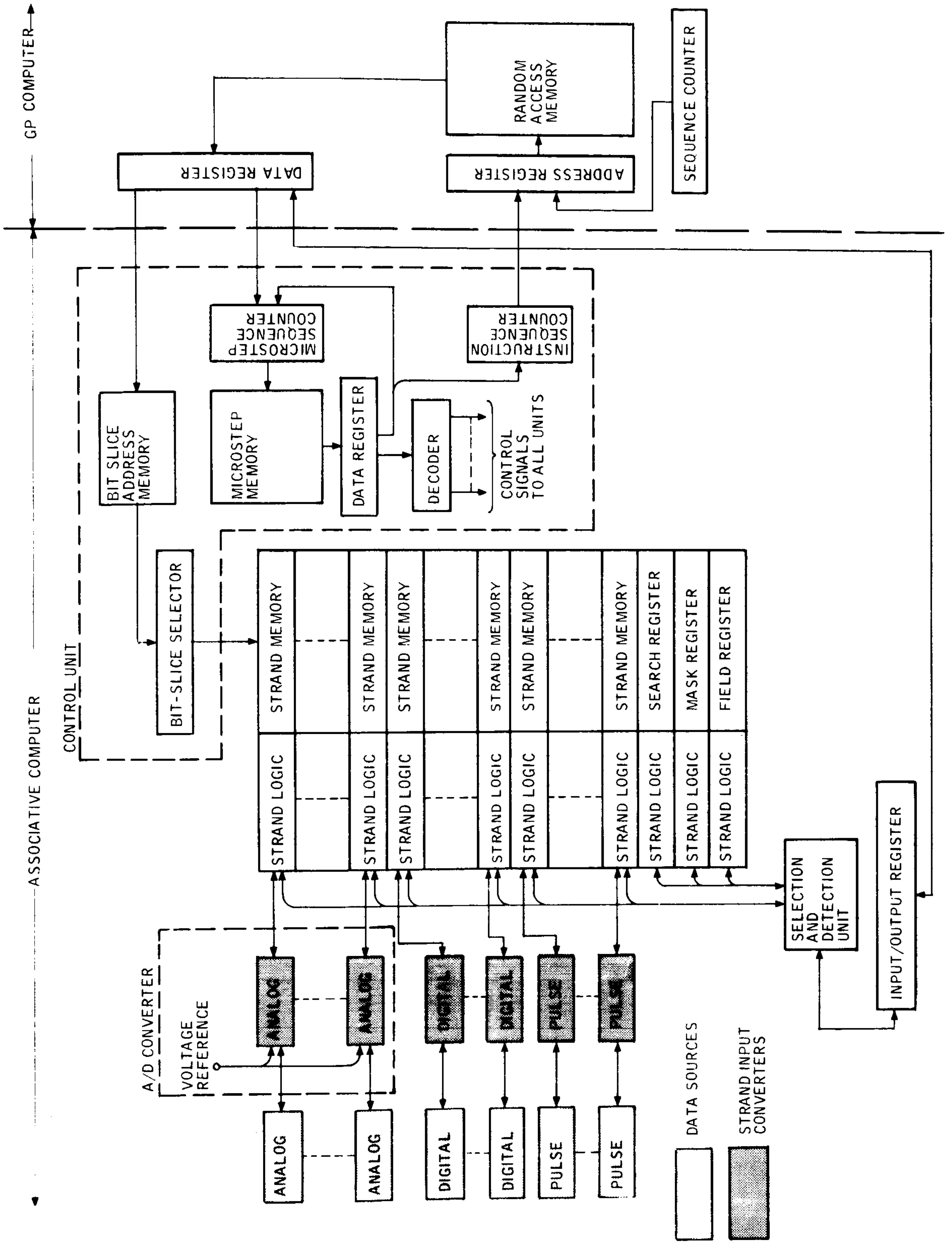


Figure 1. Data Acquisition System

The data register provides the path for instructions from the GP computer and for all data transfers. Instructions are sent directly to the associative computer control unit where they are used to initiate the proper sequence of microsteps. Data transfers, however, are directly between the data register and the associative computer I/O register.

The heart of the associative computer is an array of processing units called strands, one of which is provided for each data source. Each strand is made up of one word of memory (normally several data fields), a block of logic which is capable of serial arithmetic as well as simple comparisons, and an input converter. These parts of the strand will be called strand memory, strand logic, and strand input converter, respectively.

To describe the reading and writing capabilities of the array of strands, the term bit slice will be used; a bit-slice operation involves the same bit of all strands, and it is the only mode (bit-serial) used in the associative computer. The bit-slice selector selects the appropriate bit slice for all operations.

Three special strands--labeled the search register, the mask register, and the field register in Figure 1--are required to allow the array of strands to perform the search operations of an associative memory. All or a portion of the contents of the search strand can be compared simultaneously to the contents of the other strands for equality, inequality, or any other search criterion. The mask strand determines the portion of the search strand to be used in the search operations. Flexibility in field length is provided with the field strand; a unit in a bit position signifies the beginning of a field and zeros are used as spacers. Arithmetic operations of two general types can be performed by each strand; these are called central and field operations. In a central operation each active strand furnishes one of the operands from its own strand memory, while the second operand is obtained from the search strand. In field operations each active strand obtains both operands from its own strand memory.

The inputs from data sources to the array of strands are in one of three forms: analog, digital, or pulse. A strand input converter is provided for each of these types of inputs. The digital strand input converter will at most contain a buffer register from which measurements can be transferred bit-serially into specified fields of the strand memory. Similarly, the strand input converter for pulse inputs may include a counter in which random pulses from the data source are accumulated prior to their transfer to strand memory. The strand input converter for analog inputs performs a number of functions which allow simultaneous conversion of all analog inputs. This is described in detail in Subsection III-5.

The selection and detection unit performs several functions: It serves to detect any solutions in the strands in the course of a search operation, and the circuitry also allows readout of the selected strand into the I/O register. It also provides for distribution of the search, mask, and field words to the strands for all central operations, and for distribution of the contents of the I/O register to the strands for writing purposes. It also provides for selection of strands one at a time for readout in a situation where more than one strand has been identified by a search operation.

The control unit of the associative computer is based on the use of microprogramming. The reasons for this approach are discussed in Subsection III-3. The microstep memory stores the various microstep (microinstruction) sequences necessary in the performance of all associative instructions. For each instruction received from the GP computer, the control unit sequences through an appropriate set of microsteps. When a given associative instruction is completed, the sequence counter is incremented and the next associative instruction is read out from the random-access memory. The one or more addresses included in the associative instruction are stored in the bit-slice address memory. During the execution of an instruction they are used by the bit-slice selector to select bit slices of the strand array. Each time an address is used it is incremented and returned to the bit-slice address memory.

3. ASSOCIATIVE COMPUTER CONTROL UNIT

A microprogramming concept is employed to control the associative computer. This approach allows great flexibility in structuring the various instruction sequences, facilitates modification of the existing sequences, and allows expansion of the instruction repertoire if necessary.

a. Commands

The instruction set for an associative data acquisition system is illustrated in Table 1. The repertoire was determined after examining various data compression algorithms and developing instructions to allow execution of these procedures. If there is any necessity to add to this list, the microprogrammed structure allows this with little effort. The instruction set is divided into five subsets. The first subset contains all arithmetic operations performed by the associative device; it is subdivided into two parts, one in which the operations are performed between fields in the strands, and the other in which the contents of the search strand (or the I/O register) and the contents of one field are the operands. The second subset of instructions implements all reading and writing operations in the associative computer; these operations are performed between two fields, between the I/O register and one field in designated strands, or between a single "control" bit in the strands and the contents of a flipflop in strand logic. All search instructions are contained in the third grouping, and the subdivision is like that of the first subset. All special instructions, such as the command to sample data sources, are to be found in the fifth subset. All of the above subsets contain "conditional" instructions - i.e., the enable flipflop in strand logic determines whether that strand is to be operative or inoperative during the execution of such an instruction. The fourth subset includes all instructions which manipulate the contents of the strand logic flipflops; only those commands prefixed with an "E" are unconditional; all others depend on the contents of the enable flipflop for activation in a strand.

Table 1. Instruction Set

Group	Op Code	Name	Effect	Timing
I Arithmetic Operations	FAD	Field Add	$(A) + (B) \rightarrow C$	$3nR + nW$
	FSU	Field Subtract	$(A) - (B) \rightarrow C$	$3nR + nW$
	FMU	Field Multiply	$(A) \cdot (B) \rightarrow CC'$	$(n^2+n)R + \left(\frac{n^2+n}{2}\right) W$
	FDV	Field Divide	$(AA') \div (B) \rightarrow C$	$2n^2R + n^2W$
	FAO	Field Add One	$(A) + 1 \rightarrow A$	$nR + nW$
	FSO	Field Subtract One	$(A) - 1 \rightarrow A$	$nR + nW$
	FAV	Field Average	$1/2[(A) + (B)] \rightarrow C$	$3nR + nW$
	CAD	Central Add	$(A) + (S) \rightarrow B$	$2nR + nW$
	CSU	Central Subtract	$(A) - (S) \rightarrow B$	$2nR + nW$
	CIS	Central Inverse Subtract	$(S) - (A) \rightarrow B$	$2nR + nW$
	CMU	Central Multiply	$(A) \cdot (S) \rightarrow B$	$\left(\frac{n^2+n}{2}\right)R + \left(\frac{n^2+n}{2}\right) W$
	CDV	Central Divide	$(A) \div (S) \rightarrow B$	$n^2R + n^2W$
II Read/Write Operations	FCL	Field Clear	$O \rightarrow A$	nW
	FMV	Field Move	$(A) \rightarrow B$	$nR + nW$
	FRE	Field Read	$(A) \rightarrow IO$	nR
	CWR	Central Write	$(IO) \rightarrow A$	nW
	ELD	Enable Load	$(V) \rightarrow E$	R
	EST	Enable Store	$(E) \rightarrow V$	W
	TLD	Tag Load	$(V) \rightarrow T$	R
	TST	Tag Store	$(T) \rightarrow V$	R
	RCB	Read Control Bit	$(V) \rightarrow X$	R
III Search Operations	FGT	Field Greater Than	$1 \rightarrow T$ if $(A) > (B)$	$2nR$
	FGE	Field Greater Than or Equal	$1 \rightarrow T$ if $(A) \geq (B)$	$2nR$
	FLT	Field Less Than	$1 \rightarrow T$ if $(A) < (B)$	$2nR$
	FLE	Field Less Than or Equal	$1 \rightarrow T$ if $(A) \leq (B)$	$2nR$
	FZE	Field Zero	$1 \rightarrow T$ if $(A) = 0$	$(n-1)R$
	FNZ	Field Non-zero	$1 \rightarrow T$ if $(A) \neq 0$	$(n-1)R$
	FPS	Field Positive	$1 \rightarrow T$ if $(A) \geq 0$	R
	FNG	Field Negative	$1 \rightarrow T$ if $(A) \leq 0$	R
	FEQ	Field Equality	$1 \rightarrow T$ if $(A) = (B)$	$2nR$
	FIQ	Field Inequality	$1 \rightarrow T$ if $(A) \neq (B)$	$2nR$
	CGT	Central Greater Than	$1 \rightarrow T$ if $(A) > (S)$	nR
	CGE	Central Greater Than or Equal	$1 \rightarrow T$ if $(A) \geq (S)$	nR
	CLT	Central Less Than	$1 \rightarrow T$ if $(A) < (S)$	nR
	CLE	Central Less Than or Equal	$1 \rightarrow T$ if $(A) \leq (S)$	nR
	CEQ	Central Equality	$1 \rightarrow T$ if $(A) = (S)$	nR
	CLQ	Central Inequality	$1 \rightarrow T$ if $(A) \neq (S)$	nR
	CMA	Central Maximum	$1 \rightarrow T$ if (A) Max.	nR
	CMI	Central Minimum	$1 \rightarrow T$ if (A) Min.	nR
IV Strand Logic Operations	ESS	Enable Set	$1 \rightarrow E$	-
	ERR	Enable Reset	$0 \rightarrow E$	-
	ECM	Enable Complement	$E \rightarrow E$	-
	TSS	Tag Set	$1 \rightarrow T$	-
	TRR	Tag Reset	$0 \rightarrow T$	-
	TCM	Tag Complement	$E \rightarrow T$	-
	GTE	Gate Tag to Enable	$(T) \rightarrow E$	-
	GET	Gate Enable to Tag	$(E) \rightarrow E$	-
	OXT	"OR" Exchange and Tag	$(X) + (T) \rightarrow T$	-
V Miscellaneous Operations	SDS	Sample Data Sources	-	-
	FRD	Field Round	$(A) + 1 \rightarrow A$ if $A'_{n-1} = 1$	$(n+1) R + nW$
	GBC	Gray-to-Binary Conversion	-	-
	BRU	Unconditional Branch	-	-
	BRC	Conditional Branch	-	-
	SEL	Selection	-	-

Under the "Effect" heading in Table 1, the letters A, A', B, C, and C' denote fields in strand memory and S represents the search strand; parentheses around a letter designate the contents of that field. A' is the next field after A, and likewise for C' and C. The arrow in each operation statement represents the expression "and the result is stored in". "E" denotes the value of the enable flipflop in the strand logic, and "V" is used to designate control bits in the search memory. "T" stands for the tag flipflop strand logic. "IO" in the "Effect" column refers to the input/output register of the associative computer.

In the "Timing" column, "n" signifies the length of a single field participating in the operation; "R" and "W" respectively denote the time required for a bit-slice read operation and a bit-slice write operation in the memory.

The format for all associative computer instructions is as follows:

Operation Code	Bit-Slice Address (Operand 1)	Bit-Slice Address (Operand 2)	Bit-Slice Address (Operand 3)
8 bits	8 bits	8 bits	8 bits

The operation code defines the entry address to the particular instruction "micro-routine". The three bit-slice addresses locate the fields to be considered in the instruction execution; where only one or two fields are necessary, a "no operation" code is entered in the remaining field(s). This bit-slice address could be either the least significant or the most significant of the field, or the only bit-slice under consideration; the microprogramming of the various instructions determines the orientation of this bit-slice address. As in the ordinary usage of subroutines in GP computers, the programmer must be aware of the appropriate parameters to submit to the associative computer for proper execution of the instruction.

Two of the data compression algorithms expected to be used in the data acquisition application are programmed using this instruction list. These programs, one for ZOI (zero-order-interpolator) and one for FOIDIS (first-order-interpolator, disjoined line segment), are found in Appendix C.

b. Organization

The main parts of the control unit are the "microstep memory", the "bit-slice-address memory", and the "bit-slice selector" as shown in Figure 2.

The microstep memory provides for storage of the sequences of microsteps, or "micro-routines", for each instruction executed by the associative computer. The operation code of a given instruction is transferred to the "microstep sequence counter" (MSSC), and the contents of that memory location are then loaded from microstep memory into the "microstep data register" (MSDR). Each "microstep" (memory word) is composed of two fields of 8 bits per field, which, when decoded, present "micrands" (micro-commands) to the associative computer. One micrand is generated per field, and both decoding units are identical except for the handling of "branch" micrands which transfer the contents of the second field into MSSC when the first field is decoded as such a command. If a branch micrand is not being performed, the address in MSSC is incremented by one and the next microstep is obtained.

As shown in Figure1, the associative computer control unit provides control to the strand logic, the bit-slice selector, the selection and detection logic, and the strand input converters. A list of the necessary micrands is shown in Table 2, and Appendix B illustrates the implementation of representative instructions with this micrand repertoire.

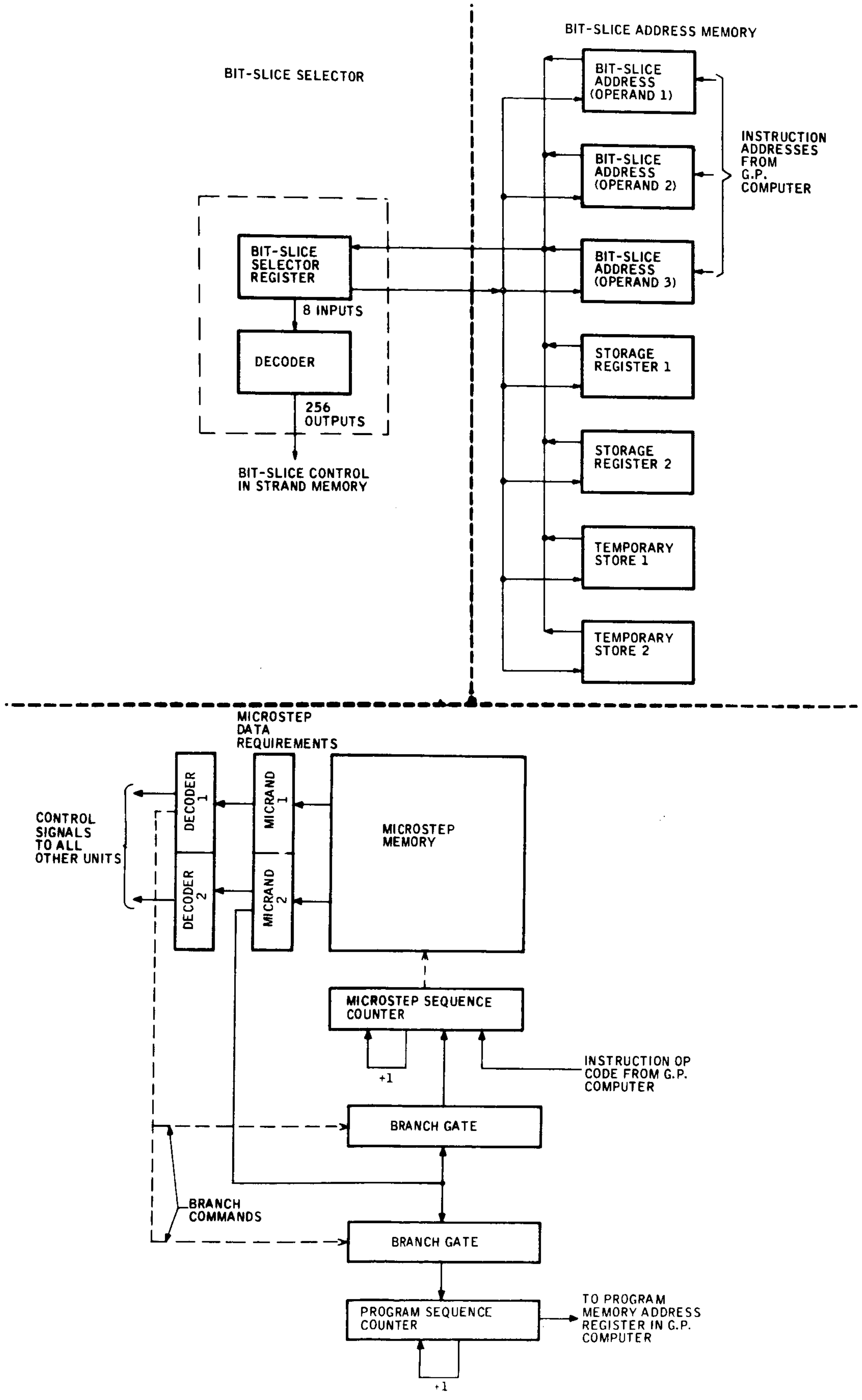

BIT-SLICE SELECTOR
BIT-SLICE ADDRESS MEMORY
BIT-SLICE ADDRESS (OPERAND 1)
BIT-SLICE ADDRESS (OPERAND 2)
BIT-SLICE ADDRESS (OPERAND 3)
INSTRUCTION ADDRESSES FROM G.P. COMPUTER
BIT-SLICE SELECTOR REGISTER
8 INPUTS
DECODER
256 OUTPUTS
BIT-SLICE CONTROL IN STRAND MEMORY
STORAGE REGISTER 1
STORAGE REGISTER 2
TEMPORARY STORE 1
TEMPORARY STORE 2
MICROSTEP DATA REQUIREMENTS
CONTROL SIGNALS TO ALL OTHER UNITS
DECODER 1
DECODER 2
MICRAND 1
MICRAND 2
MICROSTEP MEMORY
MICROSTEP SEQUENCE COUNTER
+1
INSTRUCTION OP CODE FROM G.P. COMPUTER
BRANCH GATE
BRANCH COMMANDS
BRANCH GATE
PROGRAM SEQUENCE COUNTER
+1
TO PROGRAM MEMORY ADDRESS REGISTER IN G.P. COMPUTER

Figure 2. Associative Computer Control Unit

Table 2. Micrand List (page 1 of 2)

GLOSSARY

(XXX)	=	"the contents of XXX"	XFF	=	exchange flipflop (strand logic
EFF	=	enable flipflop (strand logic)	GP. DR	=	GP computer data register
TFF	=	tag flipflop (strand logic)	IOR	=	input/output register
BSS	=	bit-slice selector	CFF	=	carry flipflop (strand logic)
MSSC	=	microstep sequence counter	MS	=	microstep
DETFF	=	detection flipflop (selection and detection)	M_i	=	i^{th} bit of mask word
S_i	=	i^{th} bit of search word	IO_o	=	least significant bit of input/output register

Micrand	Definition
MAD1	First operand addition
MAD2	Second operand addition
MAET	"AND" (EFF) with (TFF), store in TFF
MBO1	Transfer (BSS) to operand 1 register
MBO2	Transfer (BSS) to operand 2 register
MBO3	Transfer (BSS) to operand 3 register
MBRC	Transfer branch address to MSSC if DETFF = 1
MBRD	Shift buffer registers "down"
MBRU	Transfer branch address to MSSC unconditionally
MBSB	Turn on bit-slice specified by (BSS), gate into XFF
MBWD	If (BR_0) is 1, gate WDP on; otherwise gate WDN on
MCBβ	Turn on bit-slice β in control field (β = 0, 1, 2, . . . n)
MCEF	Complement (EFF)
MCTF	Complement (TFF)
MCXF	Complement (XFF)
MDBS	Decrement (BSS)
MDET	Detect presence of at least one "1", set DETFF
MDIR	Transfer data from GP. DR to IOR request
MGBX	Gate memory bit into XFF
MGET	Gate (EFF) into TFF
MGIS	Gate I/O bit or search bit into XFF
MGIX	Gate I/O bit "on"
MGSX	Gate Search bit "on"
MGTE	Gate (TFF) into EFF
MGXC	Gate (XFF) into CFF
MGXE	Gate (XFF) into EFF
MGXT	Gate (XFF) into TFF
MIBS	Increment (BSS)
MIEQ	Inequality ($<$, $>$, $\leq$, $\geq$) operation; if the current bits of the two operands are unequal, set TFF if the bit of the second operand is "0", reset TFF if that bit is "1"
MIOD	Shift IO register "down"

Table 2. Micrand List (page 2 of 2)

Micrand	Definition
MISC	Increment instruction sequential counter
MOXT	"OR" (XFF) with (TFF), store in TFF
MPRP	Propagate
MRCF	Reset CFF
MRDF	Reset DETFF
MRDS	Read Data source bit into XFF
MREF	Reset EFF
MRTF	Reset TFF
MSAR	Transfer (MS seq. ctr) to GP address register request
MSBS	Set DETFF if (BSS) is zero
MSCF	Set CFF
MSDF	Set DETFF if the field bit is "1"
MSDS	Sample digital and pulse sources
MSEF	Set EFF
MSKC	Skip next instruction if DETFF = 1
MSOP	Gate (S_i) and (M_i) to enabled strands
MSTF	Set TFF
MTBP	Transfer (BSS) to temporary store II
MTBQ	Transfer (BSS) to temporary store I
MTBR	Transfer (BSS) to storage register II
MTBS	Transfer (BSS) to storage register I
MTDI	Transfer (DETFF) to MSB of IO register
MTIS	Transfer instruction address to MS seq. ctr.
MTO1	Transfer operand 1 address to BSS
MTO2	Transfer operand 2 address to BSS
MTO3	Transfer operand 3 address to BSS
MTPB	Transfer (temporary store II) to BSS
MTQB	Transfer (temporary store I) to BSS
MTRB	Transfer (storage register II) to BSS
MTSB	Transfer (storage register I) to BSS
MWEB	Write (EFF) into memory slice
MWIO	Gate (IO_o) to strands
MWTB	Write (TFF) into memory slice

The bit-slice-address memory provides storage for the bit-slice addresses contained in associative computer instructions. These addresses specify bit slices within the strand memory which define the fields being utilized in the instruction; as many as three of these addresses are required for some instructions, so three registers are provided. Four more registers are needed to provide temporary storage for specific bit-slice addresses in the multiplication and division microprograms. All of the above registers communicate with the bit-slice selector, and the incrementing or decrementing of these bit-slice addresses is performed in the selector.

Selection of specific bit slices for reading or writing as well as the incrementing or decrementing of the bit-slice addresses are all performed in the bit-slice selector. A bit-slice address is transferred to the selector register, and the decoded address enables the bit slice corresponding to that address. Incrementing and decrementing capabilities are included in this selector register, the choice of action dependent upon the type of instruction being executed. The double-ranked feature, which is necessary to provide the incrementing and decrementing capability, can be used to provide a transfer rate advantage as well; that is, one bit-slice address can be transferred to the selector register while another address is sent back into bit-slice address memory.

4. ORGANIZATION OF STRANDS

Each strand of the associative computer is divided into three parts: (1) strand memory, (2) strand logic, and (3) strand input converter.

a. Strand Memory

The memory portion of each strand consists of 192 memory elements, one for each bit slice of the array. The number of memory elements per strand is due primarily to the requirements of the data comparison algorithm described in Section IV. All memory elements in the strand memory must be randomly accessible for reading and writing operations. A data transfer path exists from each element of a strand to the strand logic. It is assumed that no element-to-element data transfers will be performed within a strand, so no interconnection hardware will be included between strand elements. If reliability arguments warrant transfers between strands, these data movements would be effected in strand logic and not strand memory; however, no such transfer criteria is assumed in the present design.

Considered collectively the strand memories of a 256-strand associative computer consist of a 192 x 256 array of memory elements. Random access to a bit slice (one of 192) is necessary for data transfers between a single memory element of each strand and its logic, simultaneously for all strands. The enable or interrogate signal to the selected bit slice is furnished by the bit-slice selector (Figure 1).

The bit-at-a-time operating mode of the associative computer places certain requirements on the memory element; it must have fast read and write times compared with conventional memory elements, it must consume relatively little power, and it should be non-volatile. Magnetic elements such as plated wire appear suitable.

b. Strand Logic

The strand logic contains the circuits necessary to perform the required read, write, search, and arithmetic operations. The main information flow paths in this part of a strand are illustrated in Figure 3.

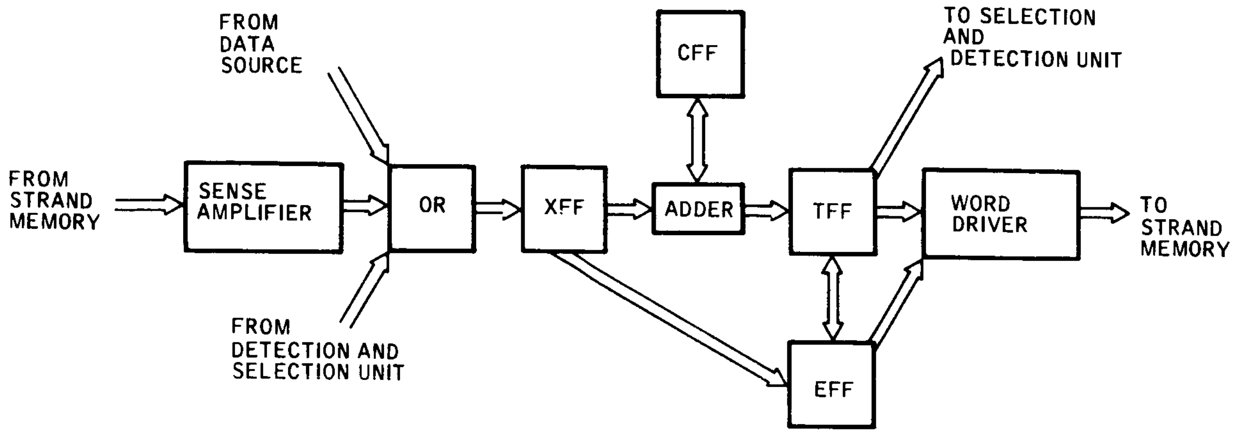


Figure 3. Information Flow Diagram for Strand Logic

The associative computer operates in a "bit-slice" mode, so that the information flow begins at the readout of a selected bit from strand memory, at the readout of a bit from the search strand, or when a bit is obtained from the strand input converter; in any case the bit is first stored in the exchange flipflop (XFF). From this location the bit value can be used in an addition operation; it can also be transferred to the tag flipflop (TFF), to the enable flipflop (EFF), or the carry flipflop (CFF). The carry flipflop (CFF) stores the partial carry during the "half add" operation and also the carry into the next most significant bit position at the conclusion of each bit operation. The TFF is also loaded with intermediate and final results of microstep sequences, so that the content of the TFF must sometimes be written into strand memory and in other cases must be tested during associative computer instructions. The state of each enable flipflop (EFF) determines whether that strand is to be active during the instruction or inactive. During instruction execution, EFF is loaded from XFF, TFF, or controlled by special micrands; the contents of EFF can also be written into a bit-slice of strand memory.

The strand logic for the search, mask, and field strands does not require arithmetic capability; however, to avoid having to provide a special design, the strand logic for these strands will be identical to that of all other strands, the only differences being in the control connections. The only parts of the strand logic necessary in these special strands are a sense amplifier, XFF, TFF, EFF, and a word driver. The contents of a bit location is stored in XFF for subsequent use by the selection and detection logic. The TFF allows write-in of data through its connection to the word driver, the write-in conditional upon the value of EFF.

The logic equations for strand logic are displayed in Table 3, and the logic diagram is found in Figure 4. All combinational logic is performed with "NAND" logic blocks, and all storage devices are of the double-ranked, clocked J-K flipflop variety. The inputs to all blocks are flipflop outputs, micrand signals, or outputs from strand memory or the selection and detection unit.

c. Strand Input Converter

There are three types of data inputs to the associative computer: analog, digital, and pulse. Both the analog and pulse inputs require "conversion" before they can be submitted to strand memory; the analog pulse and digital inputs also require a "sample-and-hold" capability to inhibit any change in values while the data is being gated into the elements of strand memory. The circuitry which effects the necessary conversion and holding of the data value is called the strand input converter.

Table 3. Strand Logic Equations

$$\begin{aligned}
\mathrm{XFF}^{t+1} &= [(\overline{\mathrm{XFF}}_K)(\mathrm{XFF}) + (\mathrm{XFF}_J)(\overline{\mathrm{XFF}})]^t \\
\mathrm{XFF}_J &= (\mathrm{MRDS})(\mathrm{SIC}_i) + (\mathrm{MGBX})(\mathrm{B}_i) + (\mathrm{MCXF}) \\
&\quad + [(\mathrm{MGSX})(\mathrm{S}_i) + (\mathrm{MGIX})(\mathrm{IO}_i)]\ (\mathrm{MGIS}) \\
\mathrm{XFF}_K &= (\mathrm{MRDS})(\overline{\mathrm{SIC}}_i)\ (\ (\mathrm{MGBX})(\overline{\mathrm{B}}_i) + (\mathrm{MCXF}) \\
&\quad + [\overline{(\mathrm{MGSX})(\mathrm{S}_i) + (\mathrm{MGIX})(\mathrm{IO}_i)}]\ (\mathrm{MGIS})
\end{aligned}$$

$$\begin{aligned}
\mathrm{CFF}^{t+1} &= [(\overline{\mathrm{CFF}}_K)(\mathrm{CFF}) + (\mathrm{CFF}_J)(\overline{\mathrm{CFF}})]^t \\
\mathrm{CFF}_J &= (\mathrm{MGXC})(\ \mathrm{XFF}) + (\mathrm{MAD1})(\mathrm{CFF})(\mathrm{XFF}) \\
&\quad + (\mathrm{MAD2})(\overline{\mathrm{CFF}})(\mathrm{TFF})(\mathrm{XFF}) + \mathrm{MSCF} \\
\mathrm{CFF}_K &= (\mathrm{MGXC})(\overline{\mathrm{XFF}}) + (\mathrm{MAD1})\ \overline{(\mathrm{CFF})(\mathrm{XFF})} \\
&\quad + (\mathrm{MAD2})(\overline{\mathrm{CFF}})\ \overline{(\mathrm{TFF})(\mathrm{XFF})} + \mathrm{MRCF}
\end{aligned}$$

$$\begin{aligned}
\mathrm{EFF}^{t+1} &= [(\overline{\mathrm{EFF}}_K)(\mathrm{EFF}) + (\mathrm{EFF}_J)(\overline{\mathrm{EFF}})]^t \\
\mathrm{EFF}_J &= (\mathrm{MGXE})(\mathrm{XFF}) + \mathrm{MSEF} + \mathrm{MCEF} \\
\mathrm{EFF}_K &= (\mathrm{MGXE})(\overline{\mathrm{XFF}}) + \mathrm{MREF} + \mathrm{MCEF} + (\mathrm{MGTE})(\overline{\mathrm{TFF}})
\end{aligned}$$

$$\begin{aligned}
\mathrm{TFF}^{t+1} &= [(\overline{\mathrm{TFF}}_K)(\mathrm{TFF}) + (\mathrm{TFF}_J)(\overline{\mathrm{TFF}})]^t \\
\mathrm{TFF}_J &= (\mathrm{MOXT})(\mathrm{XFF} + \mathrm{TFF}) + (\mathrm{MGET})(\mathrm{EFF}) \\
&\quad + \Big\{[(\mathrm{MIEQ})(\overline{\mathrm{XFF}}) + (\mathrm{MAD1})]\ [(\mathrm{CFF})(\overline{\mathrm{XFF}}) + (\overline{\mathrm{CFF}})(\mathrm{XFF})] \\
&\quad + (\mathrm{MAD2})\ [(\mathrm{TFF})(\overline{\mathrm{XFF}}) + (\mathrm{TFF})(\overline{\mathrm{XFF}})] \\
&\quad + (\mathrm{MGXT})(\mathrm{XFF}) + \mathrm{MSTF} + \mathrm{MCTF}\Big\} \\
\mathrm{TFF}_K &= (\mathrm{MOXT})(\overline{\mathrm{XFF} + \mathrm{TFF}}) + \mathrm{MGET}\ (\overline{\mathrm{EFF}}) + (\mathrm{MAET})(\overline{\mathrm{EFF}}) \\
&\quad\ [(\mathrm{MIEQ})(\mathrm{XFF}) + (\mathrm{MSOP})(\mathrm{M}_i)]\ [(\mathrm{CFF})(\overline{\mathrm{XFF}}) + (\overline{\mathrm{CFF}})(\mathrm{XFF})] \\
&\quad + \mathrm{MGXT}\ (\overline{\mathrm{XFF}}) + \mathrm{MRTF} + (\mathrm{MAD1})\ [(\mathrm{CFF})(\overline{\mathrm{XFF}}) + (\overline{\mathrm{CFF}})(\mathrm{XFF})] \\
&\quad + (\mathrm{MAD2})\ [(\mathrm{TFF})(\overline{\mathrm{XFF}}) + (\overline{\mathrm{TFF}})(\mathrm{XFF})] + (\mathrm{MCTF})(\mathrm{EFF})
\end{aligned}$$

$$\mathrm{WDP} = (\mathrm{EFF})\ [(\mathrm{TFF})(\mathrm{MWTB}) + \mathrm{MWEB}]$$

$$\mathrm{WDN} = (\mathrm{EFF})(\overline{\mathrm{TFF}})(\mathrm{MWTB}) + (\mathrm{MWEB})(\overline{\mathrm{EFF}})$$

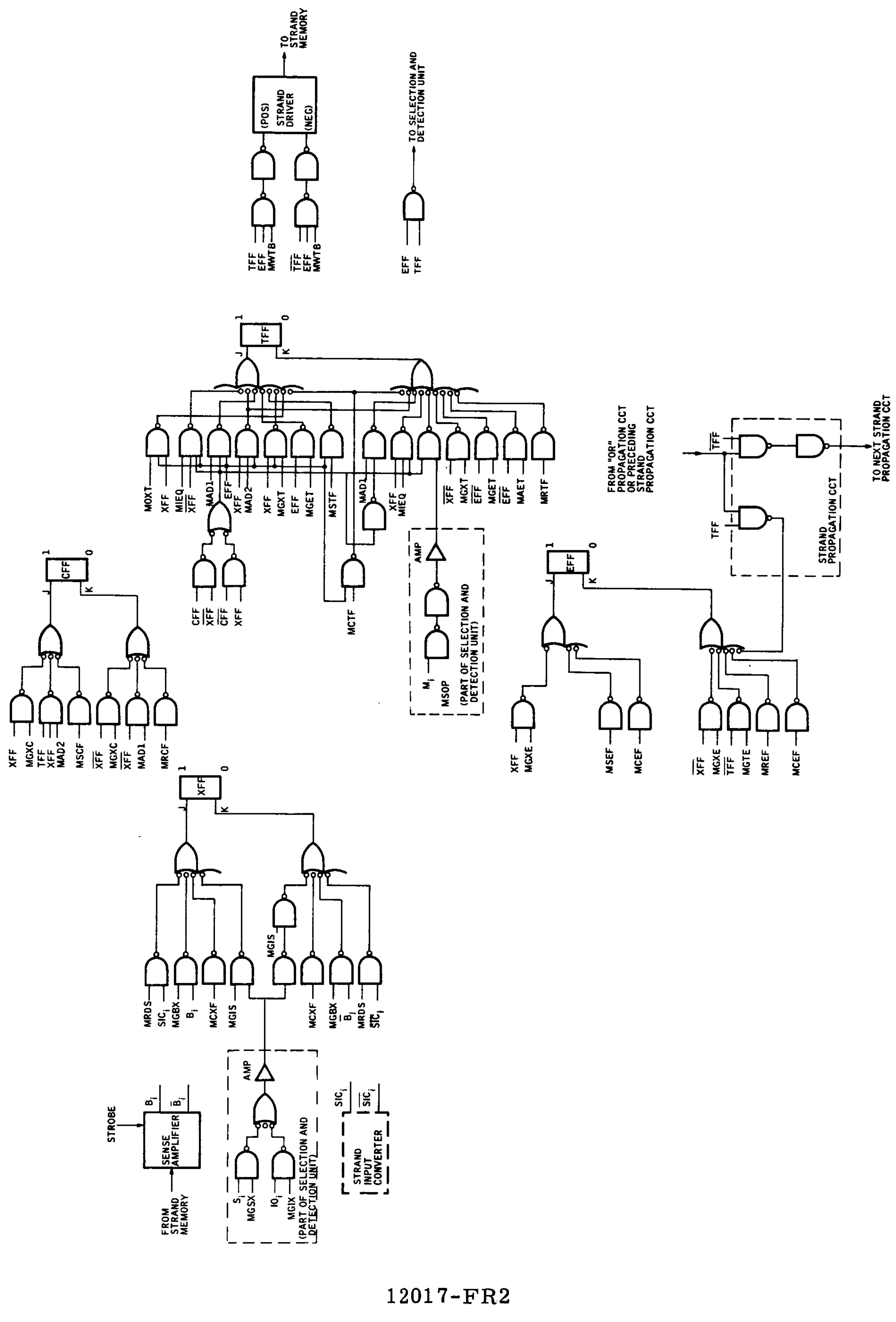

TO STRAND MEMORY
(POS) STRAND DRIVER (NEG)
TFF EFF MWTB
TFF EFF MWTB
TO SELECTION AND DETECTION UNIT
EFF TFF
TFF
CFF
XFF
EFF
MOXT XFF MIEQ XFF MAD1 EFF XFF MAD2 XFF MGXT EFF MGET MSTF MAD1 XFF MIEQ XFF MGXT EFF MGET EFF MAET MRTF
CFF XFF CFF XFF
MCTF
AMP
M_i MSOP (PART OF SELECTION AND DETECTION UNIT)
XFF MGXC TFF XFF MAD2 MSCF XFF MGXC XFF MAD1 MRCF
FROM "OR" PROPAGATION CCT OR PRECEDING STRAND PROPAGATION CCT
TFF
TFF
STRAND PROPAGATION CCT
TO NEXT STRAND PROPAGATION CCT
XFF MGXE MSEF MCEF XFF MGXE TFF MGTE MREF MCEF
MRDS SIC_i MGBX B_i MCXF MGIS MGIS MCXF MGBX B_i MRDS SIC_i
STROBE
SENSE AMPLIFIER
B_i B_i
FROM STRAND MEMORY
S_i MGSX IO_i MGIX (PART OF SELECTION AND DETECTION UNIT)
SIC_i SIC_i
STRAND INPUT CONVERTER

Figure 4. Strand Logic

The strand input converter for analog inputs will consist of the necessary normalizing, sampling, holding, and comparator circuits to accommodate the analog-to-digital conversion scheme. The details of this circuitry are found in Subsection III-5.

The strand input converter for digital inputs will consist of a buffer register which will accept a digital number from the data source in a parallel mode and will read it out bit-serially to strand memory. When the command is given to sample inputs, the "holding" circuitry must first make sure that the input from the data source is stable before the transfer to the buffer register is effected. The digital strand input converter will not perform any conversion on the data value; Gray-to-binary conversion, if necessary, will be accomplished with a special associative computer instruction.

Since the burst rate expected in many of the pulse inputs will exceed the sampling rate of the associative computer, the pulse strand input converters must function as "frequency dividers". This frequency division is accomplished by providing a binary counter to count the number of pulses from the pulse input between data acquisition cycles. The size of this binary counter is such as to contain the maximum number of pulses that can be generated between successive acquisition cycles. The sampling command transfers the contents of the binary counter into a buffer register after the absence of an input to the counter is detected (so as to lose no counts). The information is then read bit-serially into strand memory. The transfer of information to the buffer register and the resetting of the binary counter must be performed within one period of the maximum pulse rate.

5. ANALOG-TO-DIGITAL CONVERSION

The method of performing analog-to-digital conversion in the associative data acquisition system is called the sequential approximation method (Ref. 13). It might also be described as a "sectioned counter" approach with one section for each output bit. It appears to have significant advantages over the other approaches investigated. (See Reference 8 for the description of a simple counter approach, a sectioned counter approach, and a pulse-type approach.) In addition to a speed advantage, the sequential approximation approach also has the advantage that it minimizes the amount the central hardware and it requires no word-slice write operations. The central hardware is reduced to a single constant-voltage reference versus a counter and D/A converter in other methods. The elimination of the need for a word-slice write is due to the fact that a single bit of the digital representation is determined for all analog inputs during each step of the conversion operation. Thus the bit-slice write operation can be used to transfer one bit-at-a-time of the digital number to the strand memory.

The sequential approximation method is described with reference to the block diagram of Figure 5. The blocks and associated components in this diagram are intended only to represent the function being performed and not to specify the actual circuit, since there are several ways of performing many of these functions. The analog input is normalized, sampled, and stored in a holding circuit. It is then compared with the reference voltage (E_R) which is set at one-half the magnitude of the largest voltage to be converted. For all strands in which the input voltage is greater than or equal to E_R, the output bit is made a "1", the reference voltage E_R is subtracted from the input voltage, and this difference voltage is then multiplied by two. For all strands in which the input voltage is less than E_R, the output bit is made a "0" and the input voltage is multiplied by two. The output of the "doubling" amplifier is then stored in the input holding circuit in place of the input from the data source and this procedure is repeated until a sufficient number of bits (the required precision) is obtained.

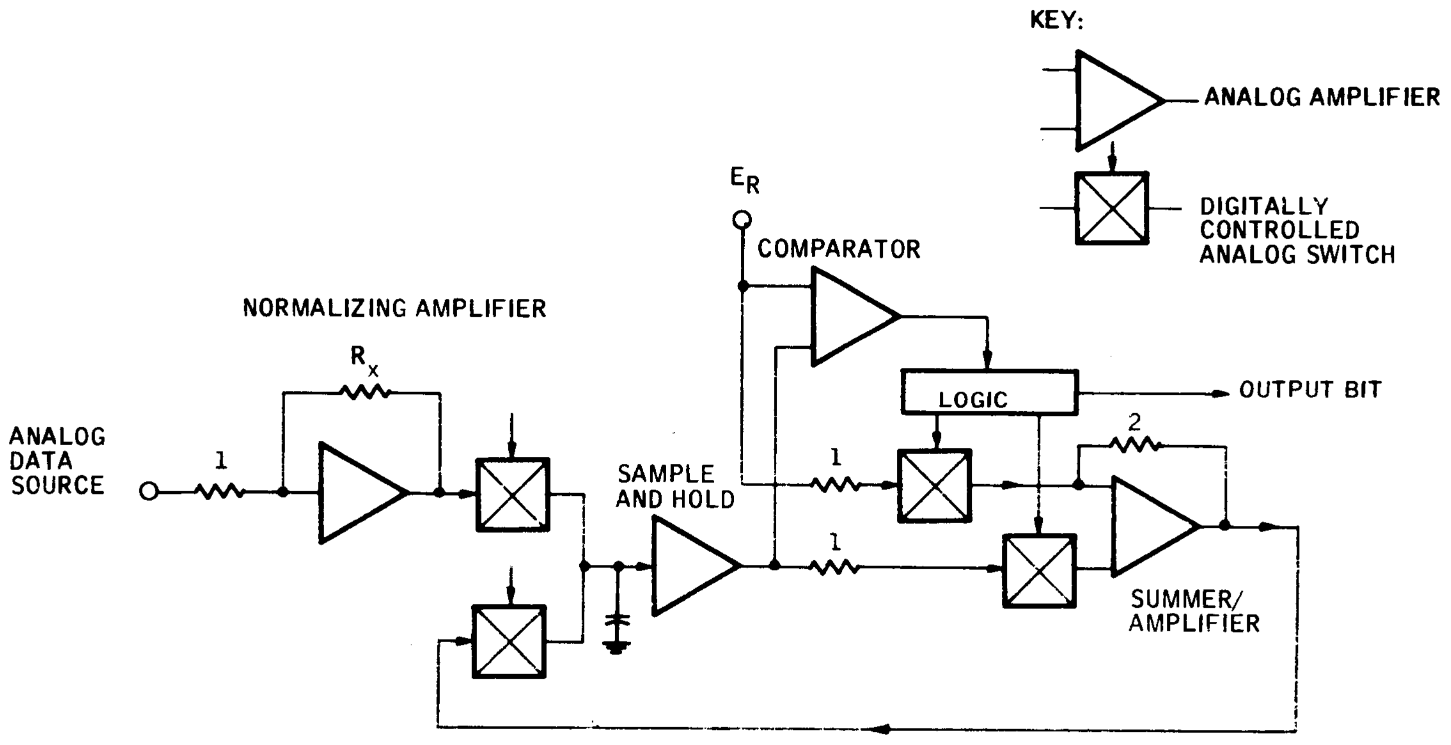


Figure 5. Strand Input Converter Employing Sequential Approximation A/D Conversion Method

A numerical example of this procedure may make its operation clearer. If the input voltage from a particular data source is 11 volts and the maximum possible input voltage is 16 volts, E_R would be eight volts and the following steps would be required to obtain a 4-bit result:

$$\begin{array}{llrrc}
\text{Is } 11 \geq 8? & & & 11.0 & 1 \\
 & & - & \underline{8.0} & \\
 & & & 3.0 & \\
 & \times 2 & = & 6.0 & \\
\hline
\text{Is } 6 \geq 8? & & & 6.0 & 0 \\
 & \times 2 & = & 12.0 & \\
\hline
\text{Is } 12 \geq 8? & & & 12.0 & 1 \\
 & & - & \underline{8.0} & \\
 & & & 4.0 & \\
 & \times 2 & = & 8.0 & \\
\hline
\text{Is } 8 \geq 8? & & & 8.0 & 1 \\
 & & - & \underline{8.0} & \\
 & & & 0.0 &
\end{array}$$

Thus the input voltage is $\frac{(1011)_2}{(8)_{10}}$ times the reference voltage or 11 volts. If the reference voltage is not a power of 2, the binary output must be multiplied by a scale factor to obtain the binary representation of the input voltage. This, of course, is true regardless of the type of converter employed.

This A/D converter can be constructed from presently available integrated circuit analog amplifiers and switches. However external circuit elements such as compensating networks constructed from discrete devices will still be required. Eight to ten bits precision (0.25 to 0.1 percent) and conversion times of approximately 50 to 100 microseconds should be achievable.* For potential applications in which these performance figures are not satisfactory, it may be necessary to take a fresh look at the problem, considering the entire converter rather than a single portion of it. As an example, in situations which require extremely high speeds, sample and hold circuits constructed

*Thus, if there are n strands connected to analog sensors, the conversion time <u>per reading</u> should be considered as 50/n or 100/n microseconds when comparing this A/D converter to a conventional one-channel device.

from presently available integrated circuits are relatively slow. In these situations, it may be necessary to apply techniques similar to those employed in high-speed sampling oscilloscopes to this problem. Another likely bottleneck as far as speed is concerned is the summer/amplifier using integrated circuit operational amplifiers. Other methods of implementing this function should be investigated.

6. SELECTION AND DETECTION UNIT

The selection and detection unit performs a number of control and input/output type functions for the array of strands. They are:

- Detection of search results
- Sequential selection of multiple matches
- Distribution of the search, mask, and input words to the strands
- Collection of outputs from the strands

To perform these functions, the unit is connected not only to the strands but also to the control unit and to the GP computer as shown in Figure 4.

The major part of the logic required to mechanize these functions is shown in Figure 6. It is basically an "OR" tree along with some special propagation logic. A portion of the tree is actually distributed among the strands as indicated in the figures.

a. Detection

The main detection function required is that of being able to determine when one or more strands have satisfied a search operation. This is accomplished by "ORing" the contents of the tag flipflop's (TFF's) in all enabled strands

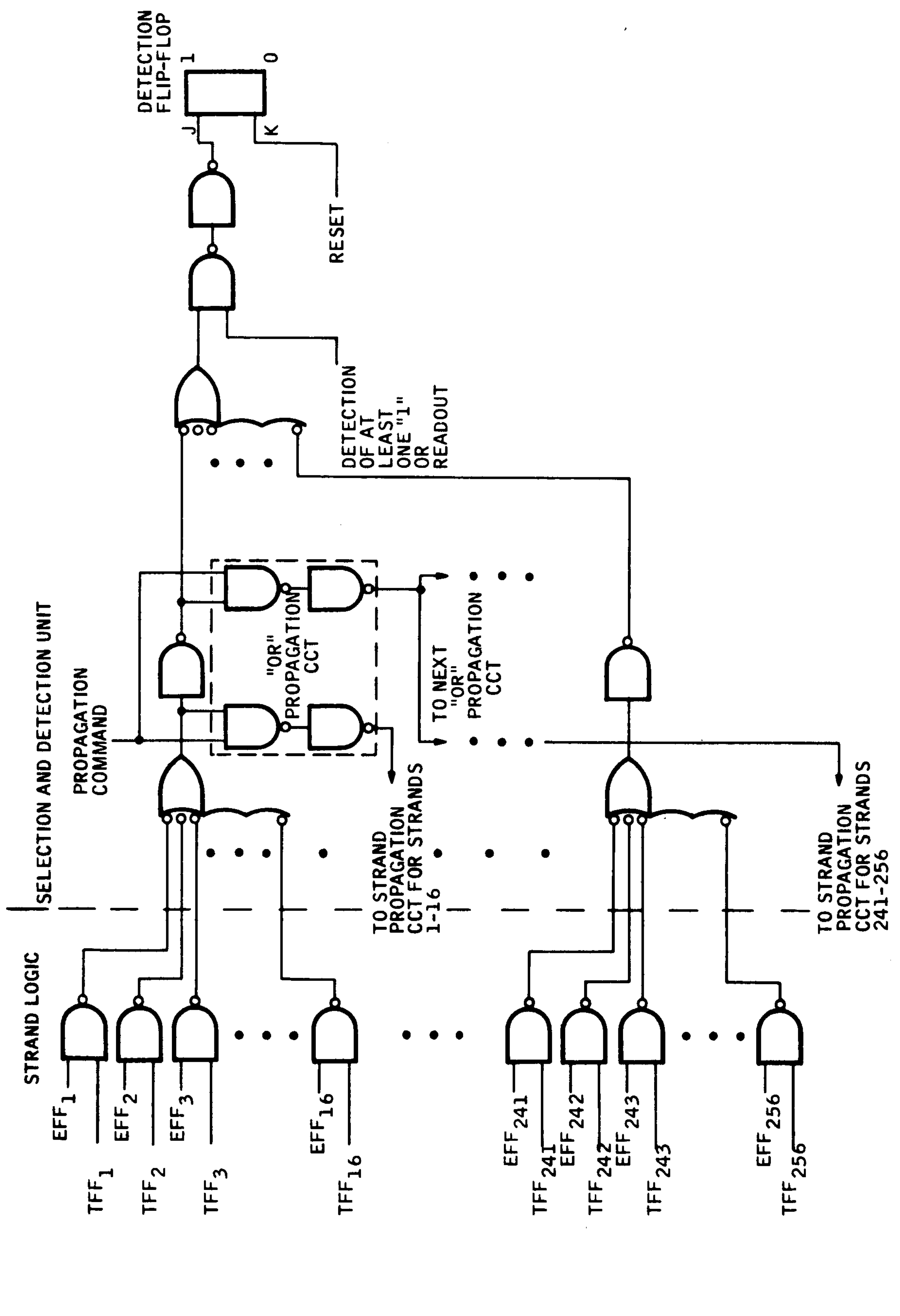


Figure 6. Selection and Detection "OR" Tree

(i.e., where EFF = 1). The detection flipflop will be set if at least one TFF is set. This result is in turn transferred to the control unit where it has an effect on the microstep sequence for certain instructions. The maximum search is an example of one operation where this information is required to select the proper microstep sequence.

Another detection function required is concerned with the use of the field strand, the purpose of which is to designate end-of-field for multi-bit operations. An example of the type of information stored in the field strand is as follows:

Field 1	Field 2	
100---0	100---0	-----

The contents of the field strand are read out to the detection flipflop (see Figure 6) in the selection and detection unit in synchronism with the readout of corresponding bits of the strands over which operations are being performed. The state of the detection flipflop is then used to effect conditional microstep branches in the control unit. For example, a "1" in the detection flipflop will be interpreted as an end of field and will cause termination of the operation.

b. Selection

The most difficult to mechanize of the functions of the selection and detection unit is that of sequential selection of multiple matches. A common requirement for this capability occurs when a search is performed and all words satisfying the search are to be read out one-at-a-time. The method chosen for mechanizing this function is called the modified propagation approach. It has a number of advantages over three other approaches (described in Reference 8) that were also investigated. In the modified propagation approach, the "OR" tree of Figure 6 divides the 256 strands into 16 subsets of 16 strands each. Selection of the first TFF containing a "1" is accomplished by propagating a control signal through circuitry which examines the logical "OR" of the TFF outputs of enabled strands in the first subset; if no "1" is found in that subset, the propagating signal proceeds to the next "OR" propagation circuit representing the next subset of 16 strands; if one or more TFF's are true in the subset, the propagating signal is directed into the subset where it examines the TFF outputs, propagating until a "1" is detected (see Figure 4). Prior to the selection sequence, the EFF is set in all strands. When the TFF value is true, the EFF in that strand is reset; EFF's are then complemented in all strands. Since only one EFF will be set at this time, a readout operation will result in the output of one word. When readout is completed, the EFF's are recomplemented and "ANDed" with TFF. The result is stored in TFF. This serves to clear TFF corresponding to the word read out. This propagation sequence repeats until all or as many as desired of the words have been read out.

c. Distribution

The third major function of the selection and detection unit is distribution of the search and mask words to the strands in "central" operations and distribution of the contents of the input/output register to the strands in write operations. In search operations, the search and mask words are obtained by bit-serial readout of the search and mask strands; in write operations, the input word is shifted bit serially from the input/output register. The selection and detection unit simply amplifies the received bit and passes it on to all strands as shown in Figure 4.

d. Collection

The fourth function of the selection and detection unit is to collect data from strands for readout purposes. The "OR" tree previously described for accomplishing the detection function also performs this function. Assuming that only one strand is enabled (i.e., only one EFF is set), the "OR" tree serves to transfer the contents of TFF in the enabled strand to the detection flipflop in the selection and detection unit. Readout of a selected word is accomplished by serial transfer from the strand memory to TFF, from TFF through the "OR" tree to the detection flipflop, and from there to the input/output register.

SECTION IV
DATA COMPRESSION TECHNIQUES

Probably the most obvious reason for choosing the associative computer approach to the data acquisition problem is that "the shape of the computer is like the shape of the problem" -- that is, a computer consisting of many partially independent units operating in parallel is well suited to monitor and process many independently originating streams of data. However, an associative computer has one other characteristic which is as fundamental as its inherent parallelism; it still retains the properties of an associative memory, and is able to perform search operations over the same data field position in all strands. Because of this charactersitic, the associative computer can conveniently examine certain aspects of all of the different data streams at once, considering them all simultaneously by means of search operations rather than one-at-a-time as a general-purpose computer would need to. It is only necessary that the pertinent data stream behavior be expressed as some sort of numerical figure-of-merit, over which one of the standard associative search operations can be performed for all strands at once.

The question then is, "How may these search capabilities best be applied to the data compression function to enhance the performance of an associative computer?" Our answer to this question, which has been presented at greater length in previous documents (Refs. 7 and 8), is summarized in this section. Basically, search operations over a figure-of-merit are used to decide how important the instantaneous value of each data source is, relative to the values for all other sources, and hence to select a fixed number of values for transmission on each sampling cycle in accordance with "interpolator" type data compression algorithms.

Assuming that a data compression procedure using search algorithms has been defined, it is then fair to ask, "What assurance is there that this mode of operation really has any practical benefits ?" To answer this question, we have simulated the operation of a data acquisition system on a large-scale digital computer, using real data in such a way that the conventional approach could be compared with the approach using search operations. The methodology and results of the simulation investigations are also included in this section.

1. THEORY

a. Choice of Data Compression Algorithms

Since each strand of an associative computer functions in effect as a small-scale stored-program computer with externally supplied instructions, a very wide variety of data compression algorithms could in principle be used. It was decided, however, to restrict attention to a small class of algorithms for the purposes of this study. The class chosen was peak-error polynomial interpolators, which approach the data compression problem from the "redundancy reduction" viewpoint; that is, these algorithms operate on the assumption that the time history of a data stream can be adequately represented by a considerably smaller number of points than were originally sampled, if a rational selection is made. The basis for this choice was as follows:

(1) These algorithms can all be readily programmed for a computer with a limited instruction repertoire.

(2) A very large proportion of naturally occurring data can be processed using algorithms of this class: any data stream which originates as a periodically sampled analog signal, any periodically incremented digital count, etc.

(3) Algorithms of this class have been found to be relatively effective in several investigations, as compared to other redundancy-reduction-type algorithms (Refs. 9, 10, 11, 12).

A detailed description of the operation of several peak-error polynomial interpolation algorithms is given in Appendix A of Reference 7. Those which are of interest here are either "zero-order" or "first-order" algorithms, which respectively fit a horizontal or a sloping straight line to a sequence of data points. Moreover, only those algorithms are useful in associative computers which require a fixed number of data storage fields regardless of the number of data points in a sequence to which a single line segment is fitted; this criterion excludes algorithms which must have all preceding points of the sequence available in storage in order to perform the computations necessary when the next point is received. The three algorithms which were incorporated in the simulator were ZOI (zero-order interpolator), FOIJON (first-order interpolator, joined line segment), and FOIDIS (first-order interpolator, disjoined line segment); the last of these was given the most attention, since a previous study (Ref. 9) found it to be essentially the best one all around.

b. Conventional Operation of the Algorithm

The conventional operation of each of these three algorithms can be qualitatively pictured in terms of an anchor point, which is the beginning point in a sequence, and a corridor, which is the locus of all admissible line segments which could still fit the sequence of points to the required tolerance ϵ up to and including the point most recently received.

In the zero-order case, the corridor initially lies from ϵ below to ϵ above the anchor point. In the first-order case, the boundaries of the corridor are lines both of which must pass through the anchor point, and are initially constructed to pass through points ϵ below and ϵ above the next point received, to see if it is still possible to fit a line segment to all points of the sequence including that one. If so, the corridor is narrowed if necessary to make all admissible line segments be within ϵ of the new point, and the sequence is continued; if not, the sequence is terminated.

When termination occurs, a computed data point is selected for transmission. This point corresponds to the sampling time before termination, and is taken as the average of the upper and lower corridor boundaries. In ZOI and FOIDIS, the point which caused the sequence to be terminated becomes the new anchor point; in FOIJON, the computed point becomes the new anchor point. In ZOI and FOIJON, only computed data points are transmitted; in FOIDIS, those real data points which are used as anchor points are transmitted as well as the computed data points, and the two types alternate in the series of transmitted points.

c. Modified Operation of the Algorithms

In the conventional or "each-source-independently" approach which has just been described, the data compression algorithms are applied in such a way that the time history of each individual data source is considered independently of those of all the rest. In the modified or "all-sources-jointly" approach, in contrast, the results obtained by applying the algorithms to all data sources are considered relative to one another or jointly.

The mathematical difference between these two approaches, for the case of interpolation algorithms, is in the criterion for deciding when to transmit the reading of a data source. For the conventional case, transmission occurs when the "corridor width" decreases to zero. For the modified case, transmission occurs when the normalized corridor width (the corridor width divided by 2ϵ) for that data source is smaller than all but K normalized corridor widths from the other data sources, where K is the number of readings which can be transmitted on that sampling cycle according to an a priori decision. The normalized corridor width for each data source is a "figure of merit" suitable as the object of a search operation (here a succession of K "minimum searches") by the associative computer.

The primary advantage of the modified approach is that the rate at which data is gathered is effectively adjusted to the rate at which it can be transmitted, instead of being allowed to fluctuate widely as in the conventional case. Since a sensor reading stands a good chance of being transmitted any time that the corridor width is small, the modified form of a data compression algorithm can in this sense anticipate to some degree when transmission is about to be absolutely required - for transmission must occur when the corridor has gone to zero - and perform the transmission at a time when not too many other values are also requiring transmission.

If a great many data sources are active, the corridor widths at the time of transmission will tend to run much smaller than if only a few are active. Thus it may be seen that the modified approach should tend to smooth out the "peaks and valleys" in the number of data sources whose readings are due for transmission on a given sampling cycle.

d. Use of a Threshold for Figures-of-Merit

More generally, the conventional mode of operation and the "pure" modified mode of operation which have been described are the endpoints in a continuum of possible modes. A threshold value may be introduced such that a reading, which is eligible for transmission according to the minimum-normalized-corridor-width criterion,is selected for transmission only if its normalized corridor width is under the threshold, which is the same for all data sources. If the threshold is set to zero, the mode of operation reduces to the conventional one. If it is set to infinity, the "pure" modified mode of operation results.

Actually, the normalized corridor width can never exceed 1.0 for the Z0I algorithm; and there is also such a limit for the first-order algorithms, but it is more complicated. Thus, as the threshold is reduced the operation comes to more nearly resemble conventional operation, and as the threshold is increased the operation similarly tends toward "pure" modified operation.

The use of such a threshold may be desirable to keep from sending a fairly large number of points for inactive data sources. Depending on the circumstances, it may be desired only to hard-limit the number of points selected for transmission on each sampling cycle, or on the other hand it may be desired to curtail the total number of points transmitted as sharply as possible; in the former case no threshold is necessary, but in the latter case it is necessary and should be chosen carefully. It is evident that at least as many points will always be transmitted in the modified mode as in the conventional mode for the same choices of tolerances for all data sources, since in the conventional mode only points corresponding to zero corridor widths are eligible for transmission whereas other points are also eligible in the modified mode. However, the additional points transmitted in the modified mode result in a better fit to the raw data, and hence if the tolerances are relaxed somewhat for the modified mode the fit should remain as good as for the conventional mode with the original tolerances. The threshold can then be adjusted so that the number of points sent in the modified mode is approximately the same as the number sent in the original conventional mode. The benefits of the modified procedure are then that, for approximately the same degree of fit and the same number of transmitted points, at most K points will be selected for transmission on each sampling cycle. Simulation investigations have now established the feasibility of such operation, as is discussed in the next section.

e. Wild-Point Rejection

Because of electrical transients and other noise aboard space vehicles, real data sequences may be afflicted with "wild points" - points which are obviously far removed from the main trend of the data. If such points are numerous, either the conventional or the modified mode of operation will function quite inefficiently, since each isolated wild point inevitably terminates a sequence of points, and the next non-wild point also terminates a sequence. To eliminate this problem, a technique of screening the raw data as it is received in order to detect wild points is necessary, so that when a wild point is discovered it can simply be ignored in further calculations.

The following definition of a wild point proved satisfactory for our purposes: a point is considered wild if it is more than 3ϵ removed from (above or below) the point before it, and more than 3ϵ removed from the point after it, if and only if the point after it is less than 5ϵ removed from the point before it, where ϵ is the tolerance for that data source. In symbols, the condition for $y(t_k)$ to be a wild point is then:

$$|y(t_k) - y(t_{k-1})| > 3\epsilon \text{ and } |y(t_{k+1}) - y(t_k)| > 3\epsilon \text{ and } |y(t_{k+1}) - y(t_{k-1})| < 5\epsilon$$

It should be noticed that a given point may be wild or not according to the ϵ assigned to the data source. For ϵ very large, it will not be wild because it is not removed by 3ϵ from the points before and after it. Also, for ϵ very small, it will still not be wild because the points before and after it will be more than 5ϵ apart. For intermediate values of ϵ, it will be wild.

2. SIMULATION INVESTIGATIONS

a. Objectives and Approach

The purpose of the simulationwas to compare the behavior of the conventional and modified approaches on real data, to determine the answers to the following questions:

(1) Is it really practical to "hard-limit" the number of data points selected for transmission on each sampling cycle to some predetermined value? That is, can the modified algorithm anticipate the need to send readings corresponding to active sources well enough that there will rarely, if ever, be too many readings to send on one sampling cycle because several corridors widths have gone to zero?

(2) Will the modified approach provide as good a "fit" to the raw data, in terms of the reconstructed time history of the values for each data source as deduced from the compressed data, as the conventional approach for a given number of points transmitted?

(3) Can raw data characteristics be abstracted and interpreted in such a way that the tolerances assigned to individual data sources can be computed automatically by the associative data acquisition system, and changed from time to time to keep the data acquisition process behaving in as optimum a manner as possible?

All three of these questions were answered affirmatively, as is discussed below; and thus the simulation investigations basically achieved what they were intended to. However, certain investigations we had originally hoped to include had to be shelved for lack of time, since one particular one - on wild-point rejection - turned out to be crucial, and to require a great deal more time than was originally planned. Although our automatic tolerance-changing were not tested, our simulation results clearly indicate the sort of procedures required.

A Scientific Data Systems 9300 computer, installed at the Honeywell Aeronautical Division in Minneapolis, was used for this simulation. The 9300 is a large-scale floating-point scientific and real-time computer, essentially comparable in gross arithmetic speed to an IBM 7094 or Univac 1107. However, Honeywell's installation is only medium-scale for a 9300; the main-frame memory comprises 16,384 24-bit words, and the peripheral equipment includes a card reader and card punch, just three magnetic tape units, a medium-scale line printer, a large-scale display console, and an analog interface to connect the 9300 in a closed-loop arrangement to a very large analog computing installation in the same room. The simulation program was written in SDS FORTRAN IV. Both the line printer and the display console were used extensively for output of results.

The raw data used for the simulation was obtained, courtesy of NASA/GSFC, in the form of an "experimenter tape" of satellite "housekeeping data" (signals monitoring the proper operation of the satellite itself). The characteristics of this data turned out to be quite fortunate for our purposes; there were enough wild points to make us aware very early of the problem of wild-point rejection, there were about the right number of sensors for this type of study (16), about half the sensors (nine in all) were relatively inactive, and the remainder exhibited several separate types of behavior - slow variation (one sensor) damped oscillations (four sensors), an almost perfect sawtooth wave (one sensor), and an irregular noise pattern (one sensor).

b. Results

The simulation has been carried out at the level of system operation, and not at the level of tracing the operation of the hardware. That is, the computations assigned to each strand are gone through, but the order of steps and manner of performing search operations have been chosen to be that most natural for a FORTRAN program rather than that most natural for the operation of a real associative computer. It is convenient to consider 16 data sources, since that is the number occurring on the NASA/GSFC tape with which we were provided. An early decision was made to pick K = 2; thus, in modified operation, the two data sources having the smallest normalized corridor widths are selected for transmission on each sampling time. Because all results must be kept in the SDS 9300 main memory in order to be displayed, only 100 sampling times can be simulated in one run, although continuing the run without reinitializing is possible so that a larger number of sampling times can be simulated in groups of 100. Each sampling time corresponds, of course, to 16 data values; and, for each data value, the raw data and the output (if any) for transmission in the conventional mode and in the modified mode are retained. Since all computations are performed in floating-point, and two SDS 9300 memory words are required for a floating-point number, 100 x 16 x 3 x 2 = 9600 memory words in all are tied up for display storage.

The program provides various options controllable from the computer sense switches and console typewriter: suppression of the printout of either the transmission record, or of the numerical results, or of both; display of the raw data, of the raw data plus the conventional mode output as a line-segment fit, of the raw data plus the similar fit for the modified mode, or of the raw data with both fits superimposed; performance of the data compression using the ZOI, FOIJON, or FOIDIS algorithms; or change of tolerances, between runs, for one or more data sources.

The printout format consists of two main parts: a transmission record (Table 4 is a typical transmission record) consisting of an array of printout codes (0, 1, 2, or 3) indicating what occurred for each data source on each sampling time, and a numeric printout (Table 5 is a typical numeric record) of three columns of data for each data source - raw data, the representation of the raw data "transmitted" after processing by the conventional algorithm, and that "transmitted" after processing by the modified algorithm. The tolerances assigned to the data sources for the run are all printed out across the top of the page for the transmission record; "EPS 1" means the tolerance ϵ assigned to sensor 1 (the first data source), etc. The data compression algorithm used (ZOI, FOIJON, or FOIDIS), the date, the value of f_t (labeled "FIG OF MERIT"), and a run number are printed out at the top of the transmission record. The leftmost column gives the index number of the sampling time, for both the transmission record and the numeric printout; in the case of the FOIDIS algorithm, there can be no output until time 3 for mathematical reasons. The numbers 01, 05, 09, 13 at the top of columns in the transmission record indicate sensors; the first block of output on the left is for sensors 1-4 in the conventional mode, the next for sensors 5-8, etc., and the similarly designated blocks on the right are for the modified mode. A single column of numbers to the right of the 13-16 block for each mode gives the number of "forced" transmissions (transmissions which occurred because the corridor width went to zero); for the modified mode only, a second column of numbers gives the total number of transmissions which occurred as a result of a search for minimum values of the normalized corridor width (figure of merit). Totals for these three columns of numbers are printed out at the bottom on the second page. Within the transmission record array, the printout codes are as follows:

0	-	no transmission for that sensor on that sampling time
1	-	transmission for that sensor on that sampling time, as a result of a search
2	-	forced transmission for that sensor on that sampling time
3	-	wild point for that sensor on that sampling time, ignored and not transmitted

Eight cases can occur on a given sampling time, as follows (where K is the desired number of data points selected):

A	-	exactly K 1 codes, all other codes 0 or 3
B	-	exactly K transmission codes including both 1 and 2 codes, all other codes 0 or 3
C	-	exactly K 2 codes, all other codes 0 or 3
D	-	more than K 2 codes; no 1 codes can then occur, for mathematical reasons

E - all codes 0 or 3

F - K-1 or fewer 1 codes, all other codes 0 or 3

G - K-1 or fewer 2 codes, all other codes 0 or 3

H - K-1 or fewer transmission codes including both 1 and 2 codes, all other codes 0 or 3

For K = 2, these cases reduce to the following:

A - two 1 codes, others 0 or 3

B - a 1 code and a 2 code, others 0 or 3

C - two 2 codes, others 0 or 3

D - three or more 2 codes, others 0 or 3

E - all codes 0 or 3

F - one 1 code, others 0 or 3

G - one 2 code, others 0 or 3

H - mathematically impossible for K = 2

A column at the right of the modified printout block gives the case for each sampling time. There is no such column for the conventional printout block, but the codes can be read from the set of runs taken for $f_t = 0$ since that condition reduces the modified mode to the conventional mode. The total number of cases of each type is listed at the bottom of the transmission record — "11 B" means 11 sampling times for which case B occurred, etc. The rightmost column gives a W code for each sampling time when one or more 3 codes occurs; the wild-point rejection scheme used (described in the preceding section) does not depend on the slope of the fitted lines, and hence the wild points detected are the same in both the conventional and the modified modes. The total number of W codes is given at the bottom of this column.

The numeric printout has 12 columns of six-place (3 integer places) decimal data per page, three columns for each of four sensors. For each sensor, RAW data, CON (conventional-mode compressed) data, and MOD (modified-mode compressed) data are given in adjoining columns. The sensor number, the tolerance ("EPS = 8.00," for instance), and the assigned priority ("P = 1.00") are given at the top of the three columns for a sensor. All runs made thus far have used P = 1.00 on all sensors; the provision made for varying P was in anticipation of more sophisticated future studies, in which another means besides the tolerance might be employed to raise or lower the frequency of transmission on a given sensor. The CON and MOD columns are delayed by one sampling time with respect to the RAW columns, for unavoidable mathematical reasons having nothing to do with the FORTRAN program. In the FOIDIS data compression algorithm runs, the first, third, and in general 2n-1th output in a CON or MOD column is a real data point which begins a line segment; and the second, fourth, and in general 2nth output number in one of these columns is a computed data point which terminates a line segment.

The transmission record for one particularly good run is reprinted here as Table 4, the numeric printout for the same run as Table 5, and the transmission record for a different run chosen to make the conventional mode produce about as many transmissions altogether as the modified mode produced on the first run (87 as compared to 88, as it turned out) as Table 6.

The final set of runs comprised a total of 105 standard runs, plus special runs. One subset of 15 runs was obtained for each of seven values of f_t, the figure of merit threshold, respectively: 0 (which reduces the modified mode to the conventional mode), 0.2, 0.4, 0.45 (which proved to be about the best choice), 0.5, 0.8 and ∞ (which simply means that no f_t is considered in the decision of whether or not to transmit a reading for a given sensor on a given

Table 4. Transmission Record-Run 35

```
                 FRICIS  11/21/66  FIG OF MERIT = .45   RUN  35

     EPS 1  EPS 2  EPS 3  EPS 4  EPS 5  EPS 6  EPS 7  EPS 8  EPS 9  EPS10  EPS11  EPS12  EPS13  EPS14  EPS15  EPS16
      8.00   8.00   8.00   8.00   8.00   8.00   8.00   8.00 100.00  75.00  75.00 125.00  75.00   8.00  30.00   8.00

01         05         09         13                01         05         09         13
 3 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
 4 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
 5 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 1 0 0  0 0 0 0   0  1  F
 6 0 0 0 0  0 0 0 0  0 1 0 0  0 0 0 0   1   0 0 0 0  0 0 0 0  0 0 1 0  0 0 1 0   0  2  A
 7 0 0 0 0  0 0 0 0  1 0 1 1  0 0 1 0   4   0 0 0 0  0 0 0 0  2 0 0 2  0 0 0 0   2  0  C
 8 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 2 0   1  0  G
 9 0 0 0 0  0 0 0 0  0 0 0 0  1 0 1 0   2   0 0 0 0  0 0 0 0  0 1 0 0  2 0 0 0   1  1  B
10 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
11 0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   1   0 0 0 0  0 0 0 0  0 0 0 1  0 0 2 0   1  1  B
12 0 0 0 0  0 0 0 0  0 1 0 1  0 0 0 0   2   0 0 0 0  0 0 0 0  1 0 1 0  0 0 0 0   0  2  A
13 0 0 0 0  0 0 0 0  1 0 1 0  0 0 1 1   4   0 0 0 0  0 0 0 0  0 0 0 0  0 0 2 2   2  0  C
14 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 2 0 1  0 0 0 0   1  1  B
15 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 1   1   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 2   1  0  G
16 0 0 0 0  0 0 0 0  0 1 0 0  0 0 0 0   1   0 0 0 0  0 0 0 0  0 1 1 0  0 0 0 0   0  2  A
17 0 0 0 0  0 0 0 0  1 0 1 1  0 0 0 0   3   0 0 0 0  0 0 0 0  2 0 0 0  0 0 0 0   1  0  G
18 0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   1   0 0 0 0  0 0 0 0  0 0 0 0  0 0 2 0   1  0  G
19 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 1  0 0 0 0   0  1  F
20 0 0 0 0  0 0 0 0  0 1 0 0  3 3 3 3   1   0 0 0 0  0 0 0 0  0 2 1 0  3 3 3 3   1  1  B  W
21 3 3 0 3  3 3 3 0  1 0 1 0  0 0 1 0   3   3 3 0 3  3 3 3 0  2 0 0 0  0 0 2 0   2  0  C  W
22 0 0 0 0  0 0 0 0  3 0 0 3  0 0 0 0   0   0 0 0 0  0 0 0 0  3 0 0 3  0 0 0 0   0  0  E  W
23 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 1 2  0 0 0 0   1  1  B
24 0 0 0 0  0 0 0 0  0 1 1 1  0 0 0 0   3   0 0 0 0  0 0 0 0  1 2 0 0  0 0 0 0   1  1  B
25 0 0 0 0  0 0 0 0  1 0 0 0  1 0 0 0   2   0 0 0 0  0 0 0 0  0 0 0 0  2 0 0 0   1  0  G
26 0 0 0 0  0 0 0 0  0 1 0 0  0 0 0 0   1   0 0 0 0  0 0 0 0  0 2 1 0  0 0 0 0   1  1  B
27 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  1 0 0 2  0 0 0 0   1  1  B
28 0 0 0 0  0 0 0 0  0 0 1 0  0 0 0 0   1   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
29 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
30 0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   1   0 0 0 0  1 0 0 0  0 0 0 0  0 0 2 0   1  1  B
31 0 0 0 0  0 0 0 0  0 0 0 1  0 0 0 0   1   0 0 0 0  0 0 0 0  0 1 1 0  0 0 0 0   0  2  A
32 0 0 0 0  0 0 0 0  0 1 1 0  0 0 0 0   2   0 0 0 0  0 0 0 0  0 0 0 1  0 0 1 0   0  2  A
33 0 0 0 0  0 0 0 0  0 0 0 1  0 0 0 0   1   0 0 0 0  0 0 0 0  1 1 0 0  0 0 0 0   0  2  A
34 0 0 0 0  0 0 0 0  1 1 0 0  0 0 1 0   3   0 0 0 0  0 0 0 0  0 0 2 0  0 0 1 0   1  1  B
35 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 2 0 2  0 0 0 0   2  0  C
36 0 0 0 0  1 0 0 0  0 0 1 0  0 0 0 0   2   0 0 0 0  0 0 0 0  1 0 0 0  0 0 0 0   0  1  F
37 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
38 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
39 0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   1   0 0 0 0  0 0 0 0  0 0 0 0  0 0 2 0   1  0  G
40 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
41 0 0 0 0  0 0 0 0  0 0 0 0  1 0 0 3   1   0 0 0 0  0 0 0 0  0 0 0 0  2 0 1 3   1  1  B  W
42 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
43 0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   1   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
44 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 2 0   1  0  G
45 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
46 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
47 0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   1   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
48 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
49 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
50 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 3   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 2 3   1  0  G  W
51 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
52 0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   1   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
53 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
54 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   0  1  F
55 0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   1   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
56 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
57 0 0 0 0  0 0 0 0  0 0 0 0  1 0 0 0   1   0 0 0 0  0 0 0 0  0 0 0 0  2 0 0 0   1  0  G
58 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   0  1  F
59 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
60 0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   1   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
61 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 2 0   1  0  G
62 0 0 0 0  0 1 0 0  0 0 0 0  0 0 1 3   2   0 0 0 0  0 2 0 0  0 0 0 0  0 0 0 3   1  0  G  W
63 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   0  1  F
64 0 0 0 0  0 1 0 0  0 0 0 0  0 0 0 0   1   0 0 0 0  0 2 0 0  0 0 0 0  0 0 0 0   1  0  G
65 0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   1   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
66 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
67 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
68 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 2 0   1  0  G
69 0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   1   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
70 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 2 0   1  0  G
71 0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   1   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
72 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
73 0 0 0 0  0 0 0 0  0 0 0 0  1 0 0 0   1   0 0 0 0  0 0 0 0  0 0 0 0  2 0 1 0   1  1  B
74 0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   1   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
75 0 0 0 0  0 3 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 3 0 0  0 0 0 0  0 0 0 0   0  0  E  W
76 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
77 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
78 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
79 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   0  1  F
80 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
81 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  1 0 1 0  0 0 0 0   0  2  A
82 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
83 0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   1   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
84 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   0  1  F
85 0 0 0 0  0 0 0 0  1 0 0 0  0 0 0 0   1   0 0 0 0  0 0 0 0  1 0 0 0  0 0 0 0   0  1  F
86 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
87 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  1 0 0 0  0 0 0 0   0  1  F
88 0 0 0 0  0 0 0 0  0 1 0 0  0 0 0 0   1   0 0 0 0  0 0 0 0  0 2 0 1  0 0 0 0   1  1  B
89 0 0 0 0  0 0 0 0  0 0 1 1  1 0 0 0   3   0 0 0 0  0 0 0 0  0 0 2 0  2 0 0 0   2  0  C
90 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 1 0 0  0 0 0 0   0  1  F
91 0 0 0 0  0 0 0 0  1 1 0 0  0 0 0 0   2   0 0 0 0  0 0 0 0  2 0 0 2  0 0 0 0   2  0  C
92 0 0 0 0  0 0 0 0  0 0 1 1  0 0 0 0   2   0 0 0 0  0 0 0 0  0 1 2 0  0 0 0 0   1  1  B
93 0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0   1   0 0 0 0  0 0 0 0  0 0 0 1  0 0 2 0   1  1  B
94 0 0 0 0  1 0 0 0  0 1 0 0  0 0 0 0   2   0 0 0 0  1 0 0 0  1 0 0 0  0 0 0 0   0  2  A
95 0 0 0 0  0 0 0 0  1 0 0 0  0 0 0 0   1   0 0 0 0  0 0 0 0  0 2 0 0  0 0 0 0   1  0  G
96 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
97 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0   0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0   0  0  E
98 0 0 0 0  0 0 0 0  0 1 0 0  0 0 0 0   1   0 0 0 0  0 0 0 0  0 2 1 0  0 0 0 0   1  1  B
99 0 0 0 0  0 0 0 0  1 0 1 1  0 0 0 0   3   0 0 0 0  0 0 0 0  2 0 0 2  0 0 0 0   2  0  C

                                       75                                         45 43     7
                  40  E                11  F                  15  G
                  16  B                 7  C                   8  A
                   C  D
```

Table 5. Numeric Printout-Run 35 (page 1 of 4)

```
                   FOIDIS  11/21/66  FIG OF MERIT = .45   RUN  35

      SENSOR  1                  SENSOR  2                  SENSOR  3                  SENSOR  4
        EPS =       8.00           EPS =       8.00           EPS =       8.00           EPS =       8.00
              P =   1.00                 P =   1.00                 P =   1.00                 P =   1.00
      RAW      CON      MOD      RAW      CON      MOD      RAW      CON      MOD      RAW      CON      MOD
  1  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  634.000     .000     .000
  2  176.000  177.000  177.000  856.000  856.000  856.000  397.000  397.000  397.000  634.000  634.000  634.000
  3  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  634.000     .000     .000
  4  176.000     .000     .000  856.000     .000     .000  397.000     .000     .000  634.000     .000     .000
  5  176.000     .000     .000  856.000     .000     .000  396.000     .000     .000  634.000     .000     .000
  6  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  634.000     .000     .000
  7  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  635.000     .000     .000
  8  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  635.000     .000     .000
  9  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 10  176.000     .000     .000  856.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 11  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 12  176.000     .000     .000  856.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 13  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  634.000     .000     .000
 14  175.000     .000     .000  856.000     .000     .000  397.000     .000     .000  634.000     .000     .000
 15  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  634.000     .000     .000
 16  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 17  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 18  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 19  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 20  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 21  998.000     .000     .000  998.000     .000     .000  396.000     .000     .000  999.000     .000     .000
 22  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  634.000     .000     .000
 23  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  634.000     .000     .000
 24  176.000     .000     .000  856.000     .000     .000  397.000     .000     .000  634.000     .000     .000
 25  176.000     .000     .000  856.000     .000     .000  397.000     .000     .000  634.000     .000     .000
 26  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 27  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 28  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 29  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 30  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 31  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 32  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 33  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 34  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 35  176.000     .000     .000  856.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 36  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  634.000     .000     .000
 37  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 38  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 39  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 40  176.000     .000     .000  857.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 41  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 42  176.000     .000     .000  856.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 43  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  634.000     .000     .000
 44  176.000     .000     .000  856.000     .000     .000  397.000     .000     .000  634.000     .000     .000
 45  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  634.000     .000     .000
 46  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  634.000     .000     .000
 47  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  634.000     .000     .000
 48  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 49  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 50  176.000     .000     .000  857.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 51  178.000     .000     .000  856.000     .000     .000  395.000     .000     .000  634.000     .000     .000
 52  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 53  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  634.000     .000     .000
 54  176.000     .000     .000  856.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 55  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  634.000     .000     .000
 56  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  634.000     .000     .000
 57  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 58  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 59  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 60  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 61  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 62  177.000     .000     .000  857.000     .000     .000  395.000     .000     .000  635.000     .000     .000
 63  179.000     .000     .000  857.000     .000     .000  395.000     .000     .000  634.000     .000     .000
 64  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 65  176.000     .000     .000  856.000     .000     .000  397.000     .000     .000  634.000     .000     .000
 66  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  634.000     .000     .000
 67  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  635.000     .000     .000
 68  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 69  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 70  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  636.000     .000     .000
 71  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 72  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 73  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  635.000     .000     .000
 74  176.000     .000     .000  856.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 75  176.000     .000     .000  856.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 76  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 77  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 78  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  636.000     .000     .000
 79  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 80  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 81  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  636.000     .000     .000
 82  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 83  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 84  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 85  176.000     .000     .000  856.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 86  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  636.000     .000     .000
 87  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  637.000     .000     .000
 88  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  637.000     .000     .000
 89  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  637.000     .000     .000
 90  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  637.000     .000     .000
 91  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  637.000     .000     .000
 92  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  636.000     .000     .000
 93  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  637.000     .000     .000
 94  177.000     .000     .000  857.000     .000     .000  396.000     .000     .000  637.000     .000     .000
 95  176.000     .000     .000  856.000     .000     .000  397.000     .000     .000  637.000     .000     .000
 96  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  637.000     .000     .000
 97  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  637.000     .000     .000
 98  177.000     .000     .000  856.000     .000     .000  397.000     .000     .000  636.000     .000     .000
 99  177.000     .000     .000  856.000     .000     .000  396.000     .000     .000  637.000     .000     .000
100  177.000     .000     .000  857.000     .000     .000  397.000     .000     .000  637.000     .000     .000
```

Table 5. Numeric Printout-Run 35 (page 2 of 4)

FOIDI8 11/21/66 FIG OF MERIT = .45 RUN 35

	SENSOR 5			SENSOR 6			SENSOR 7			SENSOR 8		
	EPS = 8.00	P = 1.00		EPS = 8.00	P = 1.00		EPS = 8.00	P = 1.00		EPS = 8.00	P = 1.00	
	RAW	CON	MOD	RAW	CON	MOD	RAW	CON	MOD	RAW	CON	MOD
1	616.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
2	618.000	616.000	616.000	219.000	218.000	218.000	160.000	159.000	159.000	189.000	188.000	188.000
3	621.000	.000	.000	220.000	.000	.000	161.000	.000	.000	189.000	.000	.000
4	623.000	.000	.000	219.000	.000	.000	161.000	.000	.000	190.000	.000	.000
5	623.000	.000	.000	220.000	.000	.000	161.000	.000	.000	190.000	.000	.000
6	625.000	.000	.000	218.000	.000	.000	160.000	.000	.000	189.000	.000	.000
7	625.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
8	624.000	.000	.000	217.000	.000	.000	159.000	.000	.000	188.000	.000	.000
9	625.000	.000	.000	217.000	.000	.000	158.000	.000	.000	187.000	.000	.000
10	625.000	.000	.000	217.000	.000	.000	158.000	.000	.000	187.000	.000	.000
11	627.000	.000	.000	218.000	.000	.000	158.000	.000	.000	187.000	.000	.000
12	629.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
13	631.000	.000	.000	219.000	.000	.000	160.000	.000	.000	189.000	.000	.000
14	633.000	.000	.000	220.000	.000	.000	161.000	.000	.000	190.000	.000	.000
15	634.000	.000	.000	220.000	.000	.000	161.000	.000	.000	190.000	.000	.000
16	635.000	.000	.000	219.000	.000	.000	160.000	.000	.000	189.000	.000	.000
17	635.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
18	634.000	.000	.000	217.000	.000	.000	159.000	.000	.000	188.000	.000	.000
19	633.000	.000	.000	217.000	.000	.000	158.000	.000	.000	187.000	.000	.000
20	632.000	.000	.000	216.000	.000	.000	158.000	.000	.000	187.000	.000	.000
21	999.000	.000	.000	999.000	.000	.000	999.000	.000	.000	187.000	.000	.000
22	633.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
23	634.000	.000	.000	218.000	.000	.000	159.000	.000	.000	189.000	.000	.000
24	636.000	.000	.000	219.000	.000	.000	160.000	.000	.000	189.000	.000	.000
25	638.000	.000	.000	219.000	.000	.000	160.000	.000	.000	189.000	.000	.000
26	638.000	.000	.000	218.000	.000	.000	160.000	.000	.000	189.000	.000	.000
27	637.000	.000	.000	218.000	.000	.000	159.000	.000	.000	189.000	.000	.000
28	636.000	.000	.000	218.000	.000	.000	158.000	.000	.000	188.000	.000	.000
29	636.000	.000	.000	217.000	.000	.000	159.000	.000	.000	188.000	.000	.000
30	635.000	.000	640.267	217.000	.000	.000	158.000	.000	.000	187.000	.000	.000
31	634.000	.000	635.000	216.000	.000	.000	158.000	.000	.000	187.000	.000	.000
32	633.000	.000	.000	217.000	.000	.000	158.000	.000	.000	188.000	.000	.000
33	633.000	.000	.000	217.000	.000	.000	159.000	.000	.000	188.000	.000	.000
34	634.000	.000	.000	218.000	.000	.000	160.000	.000	.000	189.000	.000	.000
35	633.000	.000	.000	218.000	.000	.000	160.000	.000	.000	188.000	.000	.000
36	631.000	640.967	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
37	631.000	631.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
38	630.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
39	628.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
40	626.000	.000	.000	218.000	.000	.000	158.000	.000	.000	187.000	.000	.000
41	626.000	.000	.000	217.000	.000	.000	159.000	.000	.000	188.000	.000	.000
42	625.000	.000	.000	217.000	.000	.000	159.000	.000	.000	188.000	.000	.000
43	623.000	.000	.000	218.000	.000	.000	159.000	.000	.000	189.000	.000	.000
44	621.000	.000	.000	217.000	.000	.000	159.000	.000	.000	189.000	.000	.000
45	620.000	.000	.000	218.000	.000	.000	160.000	.000	.000	188.000	.000	.000
46	618.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
47	616.000	.000	.000	217.000	.000	.000	158.000	.000	.000	187.000	.000	.000
48	615.000	.000	.000	217.000	.000	.000	158.000	.000	.000	188.000	.000	.000
49	613.000	.000	.000	217.000	.000	.000	159.000	.000	.000	188.000	.000	.000
50	612.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
51	610.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
52	609.000	.000	.000	218.000	.000	.000	160.000	.000	.000	188.000	.000	.000
53	607.000	.000	.000	218.000	.000	.000	160.000	.000	.000	189.000	.000	.000
54	606.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
55	604.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
56	603.000	.000	.000	218.000	.000	.000	159.000	.000	.000	187.000	.000	.000
57	601.000	.000	.000	217.000	.000	.000	158.000	.000	.000	188.000	.000	.000
58	600.000	.000	.000	217.000	.000	.000	158.000	.000	.000	188.000	.000	.000
59	597.000	.000	.000	217.000	.000	.000	158.000	.000	.000	188.000	.000	.000
60	596.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
61	594.000	.000	.000	218.000	.000	.000	160.000	.000	.000	188.000	.000	.000
62	592.000	.000	.000	416.000	217.621	217.621	159.000	.000	.000	188.000	.000	.000
63	590.000	.000	.000	434.000	416.000	416.000	159.000	.000	.000	189.000	.000	.000
64	589.000	.000	.000	219.000	434.000	434.000	159.000	.000	.000	189.000	.000	.000
65	588.000	.000	.000	219.000	219.000	219.000	159.000	.000	.000	188.000	.000	.000
66	586.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
67	586.000	.000	.000	218.000	.000	.000	158.000	.000	.000	187.000	.000	.000
68	584.000	.000	.000	216.000	.000	.000	158.000	.000	.000	188.000	.000	.000
69	582.000	.000	.000	217.000	.000	.000	158.000	.000	.000	187.000	.000	.000
70	580.000	.000	.000	218.000	.000	.000	158.000	.000	.000	187.000	.000	.000
71	579.000	.000	.000	218.000	.000	.000	159.000	.000	.000	189.000	.000	.000
72	577.000	.000	.000	218.000	.000	.000	160.000	.000	.000	189.000	.000	.000
73	575.000	.000	.000	220.000	.000	.000	160.000	.000	.000	189.000	.000	.000
74	573.000	.000	.000	219.000	.000	.000	160.000	.000	.000	189.000	.000	.000
75	572.000	.000	.000	459.000	.000	.000	159.000	.000	.000	189.000	.000	.000
76	571.000	.000	.000	219.000	.000	.000	159.000	.000	.000	188.000	.000	.000
77	570.000	.000	.000	217.000	.000	.000	158.000	.000	.000	188.000	.000	.000
78	569.000	.000	.000	217.000	.000	.000	159.000	.000	.000	187.000	.000	.000
79	568.000	.000	.000	217.000	.000	.000	159.000	.000	.000	187.000	.000	.000
80	567.000	.000	.000	217.000	.000	.000	159.000	.000	.000	188.000	.000	.000
81	566.000	.000	.000	218.000	.000	.000	160.000	.000	.000	189.000	.000	.000
82	565.000	.000	.000	219.000	.000	.000	160.000	.000	.000	189.000	.000	.000
83	564.000	.000	.000	219.000	.000	.000	160.000	.000	.000	190.000	.000	.000
84	563.000	.000	.000	220.000	.000	.000	160.000	.000	.000	190.000	.000	.000
85	561.000	.000	.000	219.000	.000	.000	160.000	.000	.000	189.000	.000	.000
86	560.000	.000	.000	219.000	.000	.000	160.000	.000	.000	189.000	.000	.000
87	559.000	.000	.000	218.000	.000	.000	160.000	.000	.000	189.000	.000	.000
88	559.000	.000	.000	218.000	.000	.000	158.000	.000	.000	188.000	.000	.000
89	558.000	.000	.000	218.000	.000	.000	159.000	.000	.000	187.000	.000	.000
90	558.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
91	560.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
92	563.000	.000	.000	218.000	.000	.000	159.000	.000	.000	189.000	.000	.000
93	564.000	.000	.000	219.000	.000	.000	160.000	.000	.000	190.000	.000	.000
94	563.000	556.000	556.841	219.000	.000	.000	160.000	.000	.000	190.000	.000	.000
95	563.000	563.000	563.000	219.000	.000	.000	161.000	.000	.000	190.000	.000	.000
96	562.000	.000	.000	219.000	.000	.000	160.000	.000	.000	190.000	.000	.000
97	562.000	.000	.000	219.000	.000	.000	160.000	.000	.000	189.000	.000	.000
98	562.000	.000	.000	218.000	.000	.000	160.000	.000	.000	188.000	.000	.000
99	562.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000
100	562.000	.000	.000	218.000	.000	.000	159.000	.000	.000	188.000	.000	.000

Table 5. Numeric Printout-Run 35 (page 3 of 4)

FØICIS 11/21/66 FIG ØF MERIT = .45 RUN 35

	SENSØR 9			SENSØR 10			SENSØR 11			SENSØR 12		
	FPS = 100.00			EPS = 75.00			FPS = 75.00			EPS = 125.00		
	P = 1.00			P = 1.00			P = 1.00			P = 1.00		
	RAW	CØN	MØD	RAW	CØN	MØD	RAW	CØN	MØD	RAW	CØN	MØD
1	436.000	.000	.000	321.000	.000	.000	495.000	.000	.000	320.000	.000	.000
2	477.000	436.000	436.000	365.000	321.000	321.000	544.000	495.000	495.000	368.000	320.000	320.000
3	555.000	.000	.000	465.000	.000	.000	615.000	.000	.000	477.000	.000	.000
4	651.000	.000	.000	604.000	.000	.000	693.000	.000	.000	617.000	.000	.000
5	824.000	.000	.000	814.000	.000	589.250	829.000	.000	.000	788.000	.000	.000
6	822.000	.000	.000	752.000	749.000	814.000	776.000	.000	806.500	784.000	.000	.000
7	739.000	859.000	859.000	698.000	752.000	.000	738.000	834.875	776.000	694.000	828.875	828.875
8	638.000	739.000	739.000	535.000	.000	.000	638.000	738.000	.000	501.000	694.000	694.000
9	505.000	.000	.000	364.000	.000	568.750	527.000	.000	.000	349.000	.000	.000
10	407.000	.000	.000	293.000	.000	364.000	440.000	.000	.000	284.000	.000	.000
11	375.000	.000	.000	272.000	.000	.000	404.000	.000	.000	267.000	.000	261.500
12	384.000	.000	352.333	287.000	213.667	.000	430.000	.000	389.375	282.000	198.000	267.000
13	458.000	318.167	384.000	369.000	287.000	.000	533.000	360.667	430.000	376.000	282.000	.000
14	644.000	458.000	.000	605.000	.000	312.000	700.000	533.000	.000	643.000	.000	376.000
15	821.000	.000	.000	814.000	.000	605.000	830.000	.000	.000	787.000	.000	643.000
16	819.000	.000	.000	809.000	746.500	814.000	824.000	.000	830.000	786.000	.000	.000
17	810.000	885.750	876.167	797.000	809.000	809.000	815.000	882.500	824.000	784.000	849.833	.000
18	800.000	810.000	810.000	785.000	.000	.000	807.000	815.000	.000	779.000	784.000	.000
19	770.000	.000	.000	718.000	.000	.000	751.000	.000	.000	696.000	.000	811.500
20	562.000	.000	.000	416.000	726.750	726.750	568.000	.000	756.000	389.000	.000	696.000
21	419.000	631.000	631.000	302.000	416.000	416.000	453.000	624.750	568.000	291.000	.000	.000
22	781.000	419.000	419.000	285.000	.000	.000	428.000	453.000	.000	762.000	.000	.000
23	439.000	.000	.000	341.000	.000	.000	508.000	.000	420.500	346.000	.000	25.500
24	640.000	.000	439.000	599.000	282.500	282.500	700.000	493.000	508.000	646.000	226.500	346.000
25	826.000	569.500	640.000	820.000	599.000	599.000	833.000	700.000	.000	788.000	646.000	.000
26	819.000	826.000	.000	809.000	820.000	820.000	823.000	.000	833.000	786.000	.000	.000
27	800.000	.000	865.500	781.000	809.000	809.000	802.000	.000	823.000	774.000	.000	891.000
28	776.000	.000	800.000	752.000	.000	.000	783.000	875.500	.000	754.000	.000	774.000
29	774.000	.000	.000	753.000	.000	.000	785.000	783.000	.000	758.000	.000	.000
30	790.000	.000	.000	774.000	.000	.000	802.000	.000	.000	774.000	.000	.000
31	700.000	.000	.000	631.000	.000	766.667	690.000	.000	799.667	604.000	823.500	.000
32	575.000	.000	.000	458.000	688.750	631.000	594.000	732.000	690.000	441.000	604.000	668.167
33	650.000	.000	645.833	468.000	458.000	458.000	615.000	594.000	.000	748.000	441.000	441.000
34	829.000	630.914	650.000	825.000	468.000	468.000	836.000	.000	594.000	788.000	748.000	.000
35	828.000	829.000	.000	823.000	825.000	825.000	834.000	.000	836.000	788.000	.000	859.000
36	822.000	.000	868.000	813.000	.000	823.000	827.000	863.250	.000	766.000	.000	788.000
37	809.000	.000	822.000	794.000	.000	.000	812.000	827.000	.000	781.000	.000	.000
38	790.000	.000	.000	768.000	.000	.000	795.000	.000	.000	766.000	.000	.000
39	781.000	.000	.000	758.000	.000	.000	788.000	.000	.000	760.000	.000	.000
40	788.000	.000	.000	770.000	.000	.000	797.000	.000	.000	771.000	.000	.000
41	804.000	.000	.000	792.000	.000	.000	813.000	.000	.000	782.000	.000	.000
42	817.000	.000	.000	810.000	.000	.000	826.000	.000	.000	785.000	.000	.000
43	824.000	.000	.000	809.000	.000	.000	831.000	.000	.000	787.000	.000	.000
44	827.000	.000	.000	822.000	.000	.000	834.000	.000	.000	788.000	.000	.000
45	827.000	.000	.000	821.000	.000	.000	833.000	.000	.000	788.000	.000	.000
46	824.000	.000	.000	816.000	.000	.000	829.000	.000	.000	787.000	.000	.000
47	817.000	.000	.000	806.000	.000	.000	822.000	.000	.000	785.000	.000	.000
48	810.000	.000	.000	795.000	.000	.000	814.000	.000	.000	782.000	.000	.000
49	803.000	.000	.000	788.000	.000	.000	809.000	.000	.000	780.000	.000	.000
50	804.000	.000	.000	789.000	.000	.000	809.000	.000	.000	780.000	.000	.000
51	804.000	.000	.000	792.000	.000	.000	812.000	.000	.000	781.000	.000	.000
52	810.000	.000	.000	799.000	.000	.000	818.000	.000	.000	784.000	.000	.000
53	816.000	.000	.000	806.000	.000	.000	824.000	.000	.000	786.000	.000	.000
54	820.000	.000	.000	812.000	.000	.000	827.000	.000	.000	787.000	.000	.000
55	822.000	.000	.000	814.000	.000	.000	828.000	.000	.000	788.000	.000	.000
56	822.000	.000	.000	814.000	.000	.000	829.000	.000	.000	787.000	.000	.000
57	822.000	.000	.000	814.000	.000	.000	828.000	.000	.000	786.000	.000	.000
58	821.000	.000	.000	811.000	.000	.000	826.000	.000	.000	786.000	.000	.000
59	820.000	.000	.000	810.000	.000	.000	825.000	.000	.000	786.000	.000	.000
60	819.000	.000	.000	809.000	.000	.000	824.000	.000	.000	787.000	.000	.000
61	819.000	.000	.000	809.000	.000	.000	824.000	.000	.000	786.000	.000	.000
62	814.000	.000	.000	803.000	.000	.000	820.000	.000	.000	786.000	.000	.000
63	811.000	.000	.000	800.000	.000	.000	818.000	.000	.000	784.000	.000	.000
64	811.000	.000	.000	800.000	.000	.000	818.000	.000	.000	784.000	.000	.000
65	814.000	.000	.000	803.000	.000	.000	820.000	.000	.000	785.000	.000	.000
66	817.000	.000	.000	808.000	.000	.000	825.000	.000	.000	786.000	.000	.000
67	822.000	.000	.000	814.000	.000	.000	829.000	.000	.000	787.000	.000	.000
68	825.000	.000	.000	819.000	.000	.000	832.000	.000	.000	788.000	.000	.000
69	827.000	.000	.000	822.000	.000	.000	834.000	.000	.000	787.000	.000	.000
70	828.000	.000	.000	824.000	.000	.000	835.000	.000	.000	788.000	.000	.000
71	827.000	.000	.000	822.000	.000	.000	833.000	.000	.000	788.000	.000	.000
72	824.000	.000	.000	817.000	.000	.000	829.000	.000	.000	787.000	.000	.000
73	817.000	.000	.000	806.000	.000	.000	821.000	.000	.000	785.000	.000	.000
74	806.000	.000	.000	793.000	.000	.000	812.000	.000	.000	781.000	.000	.000
75	802.000	.000	.000	788.000	.000	.000	810.000	.000	.000	780.000	.000	.000
76	808.000	.000	.000	794.000	.000	.000	815.000	.000	.000	783.000	.000	.000
77	818.000	.000	.000	809.000	.000	.000	825.000	.000	.000	785.000	.000	.000
78	803.000	.000	.000	809.000	.000	.000	823.000	.000	.000	766.000	.000	.000
79	715.000	.000	.000	774.000	.000	.000	753.000	.000	.000	695.000	.000	.000
80	740.000	.000	.000	755.000	.000	.000	778.000	.000	.000	778.000	.000	.000
81	828.000	.000	762.614	822.000	.000	.000	836.000	.000	785.911	788.000	.000	.000
82	828.000	.000	828.000	824.000	.000	.000	836.000	.000	836.000	788.000	.000	.000
83	825.000	.000	.000	817.000	.000	.000	830.000	.000	.000	787.000	.000	.000
84	812.000	.000	.000	799.000	.000	.000	816.000	.000	.000	784.000	.000	.000
85	998.000	768.618	812.000	780.000	.000	.000	803.000	.000	.000	775.000	.000	.000
86	795.000	998.000	998.000	778.000	.000	.000	803.000	.000	.000	776.000	.000	.000
87	807.000	.000	795.000	794.000	.000	.000	816.000	.000	.000	783.000	.000	.000
88	697.000	.000	807.000	636.000	785.922	786.108	702.000	.000	.000	584.000	.000	741.909
89	501.000	.000	.000	367.000	636.000	636.000	546.000	764.696	751.083	351.000	682.667	584.000
90	440.000	.000	.000	313.000	.000	367.000	504.000	546.000	546.000	312.000	351.000	.000
91	514.000	411.500	419.000	387.000	243.000	313.000	580.000	.000	.000	432.000	.000	277.500
92	719.000	514.000	514.000	586.000	387.000	387.000	764.000	664.500	558.500	787.000	415.000	432.000
93	828.000	.000	.000	823.000	.000	686.000	835.000	764.000	764.000	788.000	787.000	787.000
94	820.000	.000	828.000	811.000	866.500	.000	825.000	.000	.000	787.000	.000	788.000
95	806.000	877.500	820.000	791.000	811.000	848.000	811.000	.000	.000	781.000	.000	.000
96	797.000	800.000	.000	781.000	.000	791.000	804.000	.000	.000	776.000	.000	.000
97	800.000	.000	.000	786.000	.000	.000	809.000	.000	.000	777.000	.000	.000
98	661.000	.000	.000	588.000	786.000	786.000	669.000	.000	814.000	536.000	.000	.000
99	453.000	704.000	710.500	332.000	588.000	588.000	498.000	762.000	669.000	304.000	643.000	639.500
100	365.000	453.000	453.000	256.000	.000	.000	414.000	498.000	.000	244.000	304.000	304.000

Table 5. Numeric Printout-Run 35 (page 4 of 4)

FRIEIS 11/21/66 FIG OF MERIT = .45 RUN 35

	SENSOR 13			SENSOR 14			SENSOR 15			SENSOR 16		
	EPS = 75.00			EPS = 8.00			EPS = 30.00			EPS = 8.00		
	P = 1.00			P = 1.00			P = 1.00			P = 1.00		
	RAW	CON	MOD	RAW	CON	MOD	RAW	CON	MOD	RAW	CON	MOD
1	499.000	.000	.000	408.000	.000	.000	643.000	.000	.000	544.000	.000	.000
2	458.000	499.000	499.000	408.000	408.000	408.000	652.000	643.000	643.000	543.000	544.000	544.000
3	416.000	.000	.000	408.000	.000	.000	624.000	.000	.000	544.000	.000	.000
4	373.000	.000	.000	407.000	.000	.000	613.000	.000	.000	544.000	.000	.000
5	330.000	.000	.000	407.000	.000	.000	572.000	.000	.000	544.000	.000	.000
6	284.000	.000	.000	407.000	.000	.000	538.000	.000	582.500	543.000	.000	.000
7	237.000	.000	.000	407.000	.000	.000	607.000	555.500	538.000	544.000	.000	.000
8	191.000	.000	.000	407.000	.000	.000	535.000	607.000	607.000	543.000	.000	.000
9	822.000	191.000	191.000	407.000	.000	.000	599.000	535.000	535.000	542.000	.000	.000
10	782.000	822.000	822.000	408.000	.000	.000	676.000	599.000	.000	544.000	.000	.000
11	743.000	.000	.000	408.000	.000	.000	636.000	676.000	676.000	544.000	.000	.000
12	703.000	.000	.000	408.000	.000	.000	713.000	636.000	636.000	539.000	.000	.000
13	662.000	.000	.000	410.000	.000	.000	635.000	713.000	713.000	516.000	541.100	541.100
14	624.000	.000	.000	407.000	.000	.000	602.000	635.000	635.000	544.000	516.000	516.000
15	582.000	.000	.000	407.000	.000	.000	606.000	.000	.000	542.000	544.000	544.000
16	541.000	.000	.000	407.000	.000	.000	554.000	.000	.000	544.000	542.000	542.000
17	499.000	.000	.000	408.000	.000	.000	552.000	.000	.000	544.000	.000	.000
18	458.000	.000	.000	407.000	.000	.000	584.000	544.500	544.500	544.000	.000	.000
19	414.000	.000	.000	407.000	.000	.000	526.000	584.000	584.000	544.000	.000	.000
20	998.000	.000	.000	998.000	.000	.000	998.000	.000	.000	998.000	.000	.000
21	328.000	.000	.000	407.000	.000	.000	595.000	468.000	468.000	544.000	.000	.000
22	283.000	.000	.000	407.000	.000	.000	614.000	595.000	595.000	544.000	.000	.000
23	237.000	.000	.000	407.000	.000	.000	605.000	.000	.000	543.000	.000	.000
24	194.000	.000	.000	407.000	.000	.000	601.000	.000	.000	543.000	.000	.000
25	823.000	194.000	194.000	407.000	.000	.000	610.000	.000	.000	544.000	.000	.000
26	784.000	823.000	823.000	408.000	.000	.000	639.000	.000	.000	544.000	.000	.000
27	744.000	.000	.000	407.000	.000	.000	642.000	.000	.000	544.000	.000	.000
28	703.000	.000	.000	408.000	.000	.000	629.000	.000	.000	544.000	.000	.000
29	662.000	.000	.000	408.000	.000	.000	625.000	.000	.000	544.000	.000	.000
30	622.000	.000	.000	407.000	.000	.000	549.000	636.333	636.333	544.000	.000	.000
31	581.000	.000	.000	408.000	.000	.000	517.000	549.000	549.000	543.000	.000	.000
32	540.000	.000	.000	407.000	.000	.000	564.000	.000	517.000	543.000	.000	.000
33	499.000	.000	.000	407.000	.000	.000	543.000	.000	564.000	544.000	.000	.000
34	458.000	.000	.000	406.000	.000	.000	606.000	534.750	543.000	544.000	.000	.000
35	415.000	.000	.000	407.000	.000	.000	602.000	606.000	606.000	544.000	.000	.000
36	372.000	.000	.000	407.000	.000	.000	644.000	.000	.000	544.000	.000	.000
37	329.000	.000	.000	407.000	.000	.000	626.000	.000	.000	544.000	.000	.000
38	282.000	.000	.000	407.000	.000	.000	634.000	.000	.000	543.000	.000	.000
39	237.000	.000	.000	407.000	.000	.000	709.000	643.000	643.000	543.000	.000	.000
40	192.000	.000	.000	408.000	.000	.000	625.000	709.000	709.000	544.000	.000	.000
41	822.000	192.000	192.000	412.000	.000	.000	609.000	.000	625.000	501.000	.000	.000
42	783.000	822.000	822.000	407.000	.000	.000	539.000	.000	609.000	544.000	.000	.000
43	743.000	.000	.000	407.000	.000	.000	528.000	530.500	.000	544.000	.000	.000
44	703.000	.000	.000	406.000	.000	.000	589.000	528.000	513.500	543.000	.000	.000
45	662.000	.000	.000	407.000	.000	.000	601.000	.000	589.000	544.000	.000	.000
46	622.000	.000	.000	408.000	.000	.000	619.000	.000	.000	543.000	.000	.000
47	581.000	.000	.000	408.000	.000	.000	593.000	635.000	.000	543.000	.000	.000
48	541.000	.000	.000	407.000	.000	.000	615.000	593.000	.000	544.000	.000	.000
49	499.000	.000	.000	406.000	.000	.000	641.000	.000	.000	544.000	.000	.000
50	458.000	.000	.000	407.000	.000	.000	711.000	.000	628.333	514.000	.000	.000
51	415.000	.000	.000	407.000	.000	.000	686.000	.000	711.000	543.000	.000	.000
52	372.000	.000	.000	407.000	.000	.000	620.000	713.167	.000	544.000	.000	.000
53	328.000	.000	.000	407.000	.000	.000	563.000	620.000	.000	543.000	.000	.000
54	283.000	.000	.000	407.000	.000	.000	583.000	.000	569.500	543.000	.000	.000
55	237.000	.000	.000	408.000	.000	.000	611.000	559.500	583.000	544.000	.000	.000
56	191.000	.000	.000	407.000	.000	.000	608.000	611.000	.000	543.000	.000	.000
57	822.000	191.000	191.000	408.000	.000	.000	581.000	.000	.000	544.000	.000	.000
58	782.000	822.000	822.000	408.000	.000	.000	547.000	.000	594.000	544.000	.000	.000
59	743.000	.000	.000	408.000	.000	.000	541.000	.000	547.000	544.000	.000	.000
60	703.000	.000	.000	406.000	.000	.000	594.000	538.333	.000	543.000	.000	.000
61	662.000	.000	.000	407.000	.000	.000	704.000	594.000	579.500	542.000	.000	.000
62	623.000	.000	.000	405.000	.000	.000	710.000	704.000	704.000	440.000	.000	.000
63	582.000	.000	.000	407.000	.000	.000	639.000	710.000	710.000	543.000	.000	.000
64	541.000	.000	.000	407.000	.000	.000	639.000	.000	639.000	544.000	.000	.000
65	499.000	.000	.000	407.000	.000	.000	635.000	618.500	.000	543.000	.000	.000
66	458.000	.000	.000	407.000	.000	.000	603.000	635.000	.000	543.000	.000	.000
67	414.000	.000	.000	407.000	.000	.000	603.000	.000	.000	544.000	.000	.000
68	371.000	.000	.000	408.000	.000	.000	522.000	.000	602.000	544.000	.000	.000
69	329.000	.000	.000	407.000	.000	.000	594.000	547.000	522.000	544.000	.000	.000
70	283.000	.000	.000	408.000	.000	.000	533.000	594.000	594.000	544.000	.000	.000
71	238.000	.000	.000	407.000	.000	.000	577.000	533.000	533.000	543.000	.000	.000
72	193.000	.000	.000	407.000	.000	.000	659.000	577.000	.000	543.000	.000	.000
73	822.000	193.000	193.000	407.000	.000	.000	659.000	.000	655.000	543.000	.000	.000
74	783.000	822.000	822.000	408.000	.000	.000	660.000	685.000	659.000	544.000	.000	.000
75	743.000	.000	.000	410.000	.000	.000	656.000	660.000	.000	543.000	.000	.000
76	703.000	.000	.000	408.000	.000	.000	609.000	.000	.000	543.000	.000	.000
77	661.000	.000	.000	408.000	.000	.000	613.000	.000	.000	544.000	.000	.000
78	623.000	.000	.000	407.000	.000	.000	584.000	.000	.000	544.000	.000	.000
79	581.000	.000	.000	407.000	.000	.000	556.000	.000	595.250	544.000	.000	.000
80	541.000	.000	.000	408.000	.000	.000	575.000	.000	556.000	544.000	.000	.000
81	500.000	.000	.000	407.000	.000	.000	543.000	.000	.000	543.000	.000	.000
82	459.000	.000	.000	408.000	.000	.000	556.000	.000	.000	543.000	.000	.000
83	415.000	.000	.000	408.000	.000	.000	588.000	533.800	.000	543.000	.000	.000
84	373.000	.000	.000	408.000	.000	.000	627.000	588.000	574.000	544.000	.000	.000
85	329.000	.000	.000	407.000	.000	.000	610.000	.000	627.000	544.000	.000	.000
86	283.000	.000	.000	407.000	.000	.000	615.000	.000	.000	542.000	.000	.000
87	237.000	.000	.000	407.000	.000	.000	613.000	.000	.000	543.000	.000	.000
88	192.000	.000	.000	408.000	.000	.000	627.000	.000	.000	544.000	.000	.000
89	822.000	192.000	192.000	407.000	.000	.000	653.000	.000	.000	543.000	.000	.000
90	783.000	822.000	822.000	407.000	.000	.000	647.000	.000	.000	543.000	.000	.000
91	743.000	.000	.000	407.000	.000	.000	636.000	.000	.000	543.000	.000	.000
92	704.000	.000	.000	407.000	.000	.000	642.000	.000	.000	543.000	.000	.000
93	663.000	.000	.000	408.000	.000	.000	521.000	670.500	645.173	544.000	.000	.000
94	623.000	.000	.000	408.000	.000	.000	562.000	521.000	521.000	543.000	.000	.000
95	583.000	.000	.000	408.000	.000	.000	557.000	.000	.000	544.000	.000	.000
96	542.000	.000	.000	408.000	.000	.000	603.000	.000	.000	543.000	.000	.000
97	499.000	.000	.000	408.000	.000	.000	619.000	.000	.000	544.000	.000	.000
98	458.000	.000	.000	408.000	.000	.000	640.000	.000	.000	544.000	.000	.000
99	415.000	.000	.000	407.000	.000	.000	655.000	.000	.000	543.000	.000	.000
100	372.000	.000	.000	407.000	.000	.000	638.000	.000	.000	544.000	.000	.000

Table 6. Transmission Record - Run 36

FRICIS 11/21/66 FIG OF MERIT = .45 RUN 36

EPS 1 8.00 | EPS 2 8.00 | EPS 3 8.00 | EPS 4 8.00 | EPS 5 8.00 | EPS 6 8.00 | EPS 7 8.00 | EPS 8 8.00 | EPS 9 65.00 | EPS10 45.00 | EPS11 45.00 | EPS12 45.00 | EPS13 45.00 | EPS14 8.00 | EPS15 20.00 | EPS16 8.00

	01	05	09	13		01	05	09	13				
3	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
4	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 1 0 0	0 0 0 0	0	1	F	
5	0 0 0 0	0 0 0 0	0 1 0 0	0 0 1 0	2	0 0 0 0	0 0 0 0	1 0 0 0	0 0 2 0	1	1	B	
6	0 0 0 0	0 0 0 0	0 0 1 0	0 0 0 0	1	0 0 0 0	0 0 0 0	0 2 2 0	0 0 0 0	2	0	C	
7	0 0 0 0	0 0 0 0	1 0 0 1	0 0 3 0	2	0 0 0 0	0 0 0 0	0 0 0 2	0 0 3 0	1	0	G	W
8	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 1 0 0	0 0 1 0	0	2	A	
9	0 0 0 0	0 0 0 0	0 1 0 0	1 0 1 0	3	0 0 0 0	0 0 0 0	1 0 0 0	2 0 0 0	1	1	B	
10	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 1 0 0	0 0 0 0	0	1	F	
11	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 0	2	0 0 0 0	0 0 0 0	0 0 0 2	0 0 2 0	2	0	C	
12	0 0 0 0	0 0 0 0	1 0 1 0	0 0 3 0	2	0 0 0 0	0 0 0 0	0 0 2 0	0 0 3 0	1	0	G	W
13	0 0 0 0	0 0 0 0	0 1 0 0	0 0 0 1	2	0 0 0 0	0 0 0 0	2 0 0 0	0 0 0 2	2	0	C	
14	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 2 0 1	0 0 0 0	1	1	B	
15	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 1	2	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 2	1	0	G	
16	0 0 0 0	0 0 0 0	0 1 1 0	0 0 1 0	3	0 0 0 0	0 0 0 0	2 2 2 0	0 0 2 0	4	0	D	
17	0 0 0 0	0 0 0 0	1 0 0 0	0 0 0 0	1	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0	0	1	F	
18	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
19	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 2 0	1	0	G	
20	0 0 0 0	0 0 0 0	1 1 1 1	3 3 3 3	4	0 0 0 0	0 0 0 0	2 2 2 2	3 3 3 3	4	0	D	W
21	3 3 0 3	3 3 3 0	0 0 0 0	0 0 0 0	0	3 3 0 3	3 3 3 0	0 0 0 0	0 0 0 0	0	0	E	W
22	0 0 0 0	0 0 0 0	3 0 0 3	0 0 0 0	0	0 0 0 0	0 0 0 0	3 1 0 3	0 0 0 0	0	1	F	W
23	0 0 0 0	0 0 0 0	1 1 1 0	0 0 0 0	3	0 0 0 0	0 0 0 0	2 0 2 0	0 0 0 0	2	0	C	
24	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 0	2	0 0 0 0	0 0 0 0	0 2 0 2	0 0 2 0	3	0	D	
25	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0	1	0 0 0 0	0 0 0 0	0 0 0 0	2 0 0 0	1	0	G	
26	0 0 0 0	0 0 0 0	1 1 1 0	0 0 0 0	3	0 0 0 0	0 0 0 0	2 2 2 0	0 0 0 0	3	0	D	
27	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0	0	1	F	
28	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0	1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	0	1	F	
29	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
30	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	1 0 0 0	0 0 0 0	0 0 2 0	1	1	B	
31	0 0 0 0	0 0 0 0	0 1 1 0	0 0 0 0	2	0 0 0 0	0 0 0 0	0 2 2 0	0 0 0 0	2	0	C	
32	0 0 0 0	0 0 0 0	1 0 0 1	0 0 1 0	3	0 0 0 0	0 0 0 0	2 0 0 2	0 0 2 0	3	0	D	
33	0 0 0 0	0 0 0 0	0 1 0 0	0 0 0 0	1	0 0 0 0	0 0 0 0	0 2 1 0	0 0 0 0	1	1	B	
34	0 0 0 0	0 0 0 0	0 0 1 1	0 0 1 0	3	0 0 0 0	0 0 0 0	0 0 0 2	0 0 2 0	2	0	C	
35	0 0 0 0	0 0 0 0	0 1 0 0	0 0 0 0	1	0 0 0 0	0 0 0 0	0 2 2 0	0 0 0 0	2	0	C	
36	0 0 0 0	1 0 0 0	1 0 0 0	0 0 0 0	2	0 0 0 0	0 0 0 0	2 0 0 0	0 0 1 0	1	1	B	
37	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
38	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
39	0 0 0 0	0 0 0 0	0 0 0 0	0 0 3 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 3 0	0	0	E	W
40	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
41	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 3	1	0 0 0 0	0 0 0 0	0 0 0 0	2 0 0 3	1	0	G	W
42	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 1 0 0	0 0 2 0	1	1	B	
43	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
44	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 2 0	1	0	G	
45	0 0 0 0	0 0 0 0	0 1 0 0	0 0 0 0	1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
46	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
47	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	0	1	F	
48	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
49	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
50	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 3	1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 3	0	1	F	W
51	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
52	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
53	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	0	1	F	
54	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
55	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
56	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
57	0 0 0 0	0 0 0 0	0 0 0 0	1 0 1 0	2	0 0 0 0	0 0 0 0	0 0 0 0	2 0 2 0	2	0	C	
58	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
59	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
60	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 2 0	1	0	G	
61	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
62	0 0 0 0	0 1 0 0	0 0 0 0	0 0 1 3	2	0 0 0 0	0 2 0 0	0 0 0 0	0 0 2 3	2	0	C	W
63	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
64	0 0 0 0	0 1 0 0	0 0 0 0	0 0 1 0	2	0 0 0 0	0 2 0 0	0 0 0 0	0 0 2 0	2	0	C	
65	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
66	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
67	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
68	0 0 0 0	0 0 0 0	0 0 0 0	0 0 3 0	0	0 0 0 0	0 0 0 0	0 0 1 0	0 0 3 0	0	1	F	W
69	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
70	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 2 0	1	0	G	
71	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
72	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
73	0 0 0 0	0 0 0 0	0 0 0 0	1 0 1 0	2	0 0 0 0	0 0 0 0	0 0 0 0	2 0 2 0	2	0	C	
74	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
75	0 0 0 0	0 3 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 3 0 0	0 0 0 0	0 0 0 0	0	0	E	W
76	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	0	1	F	
77	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
78	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
79	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	1	0 0 0 0	0 0 0 0	1 0 1 0	0 0 0 0	0	2	A	
80	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 1 0 0	0 0 1 0	0	2	A	
81	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
82	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 0	0	2	A	
83	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
84	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	1	0 0 0 0	0 0 0 0	0 1 1 0	0 0 0 0	0	2	A	
85	0 0 0 0	0 0 0 0	1 0 0 0	0 0 0 0	1	0 0 0 0	0 0 0 0	2 0 0 0	0 0 2 0	2	0	C	
86	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
87	0 0 0 0	0 0 0 0	1 0 0 0	0 0 0 0	1	0 0 0 0	0 0 0 0	2 0 0 0	0 0 0 0	1	0	G	
88	0 0 0 0	0 0 0 0	0 1 1 1	0 0 0 0	3	0 0 0 0	0 0 0 0	0 2 2 2	0 0 0 0	3	0	D	
89	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0	1	0 0 0 0	0 0 0 0	0 0 0 0	2 0 0 0	1	0	G	
90	0 0 0 0	0 0 0 0	0 1 0 0	0 0 0 0	1	0 0 0 0	0 0 0 0	0 2 1 0	0 0 0 0	1	1	B	
91	0 0 0 0	0 0 0 0	1 0 1 1	0 0 0 0	3	0 0 0 0	0 0 0 0	2 0 0 2	0 0 0 0	2	0	C	
92	0 0 0 0	0 0 0 0	0 1 0 0	0 0 0 0	1	0 0 0 0	1 0 0 0	0 2 0 0	0 0 0 0	1	1	B	
93	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 0	2	0 0 0 0	0 0 0 0	0 0 0 2	0 0 2 0	2	0	C	
94	0 0 0 0	1 0 0 0	1 1 1 0	0 0 0 0	4	0 0 0 0	0 0 0 0	2 2 2 0	0 0 0 0	3	0	D	
95	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	0	1	F	
96	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
97	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	E	
98	0 0 0 0	0 0 0 0	0 1 1 1	0 0 0 0	3	0 0 0 0	0 0 0 0	0 2 2 2	0 0 0 0	3	0	D	
99	0 0 0 0	0 0 0 0	1 0 0 0	0 0 0 0	1	0 0 0 0	0 0 0 0	2 0 0 0	0 0 1 0	1	1	B	
					87					75	32		11

37 E
10 B
8 D

12 F
14 C

11 G
5 A

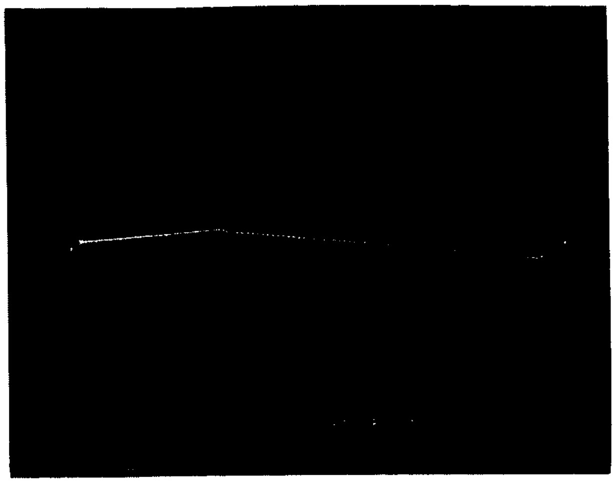
sampling time). The nine "inactive sensors (sensors 1-4, 6-8, 14, and 16) and the 1 "slowly varying" sensor (sensor 5), are all given tolerances of 8.00 (Option I), 4.00 (Option II), or 2.00 (Option III). The six "active" sensors (9-13 and 15) are given tolerances from one of the following standard options:

Option	Sensor					
	9	10	11	12	13	15
Option 1	200.00	150.00	150.00	250.00	150.00	65.00
Option 1A	200.00	150.00	150.00	25.00	150.00	65.00
Option 2	100.00	75.00	75.00	125.00	75.00	30.00
Option 3	50.00	35.00	35.00	65.00	35.00	15.00
Option 4	25.00	15.00	15.00	30.00	15.00	8.00

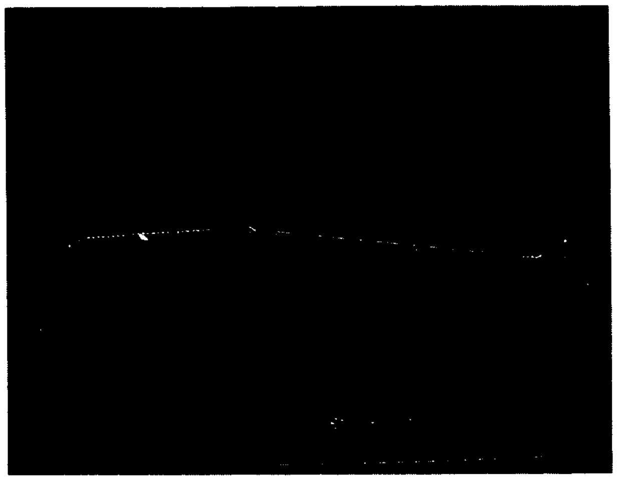
A combination of an option for the inactive sensors and one for the active sensors gives a tolerance set (I-1, III-2, etc). All 15 possible tolerance sets were run for each of the seven values of f_t, thus yielding 105 standard runs. The particular special run whose transmission record is printed out as Table 6 is denoted hereafter as that for "tolerance set I-2S," and uses the following tolerances: 8.00 for sensors 1-8, 14, and 16; 65.00 for 9; 45.00 for 10, 11, and 13; 85.00 for 12; and 20.00 for 15.

Figure 7 through 13 give the results for runs I-2 and I-2S with $f_t = 0.45$, for the interesting sensors. For sensors 5, 9-12, and 15, four pictures are given to compare the I-2 modified, I-2 conventional, and I-2S conventional, with one more showing the first two of these superimposed. Sensor 13 put out an almost perfect sawtooth wave, and all of these four plots coincided perfectly for it; hence only one picture is given.

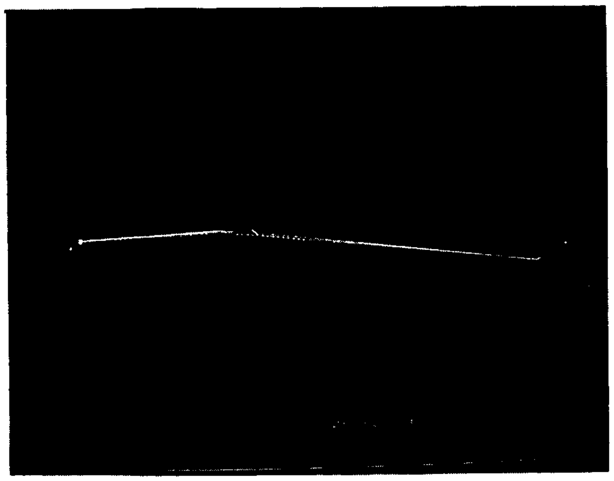
Table 7 gives the results in short form for each of the 105 standard runs. The tolerance sets are indicated across the top of the table, and the value of f_t is indicated for each block of data. The number of sampling times for



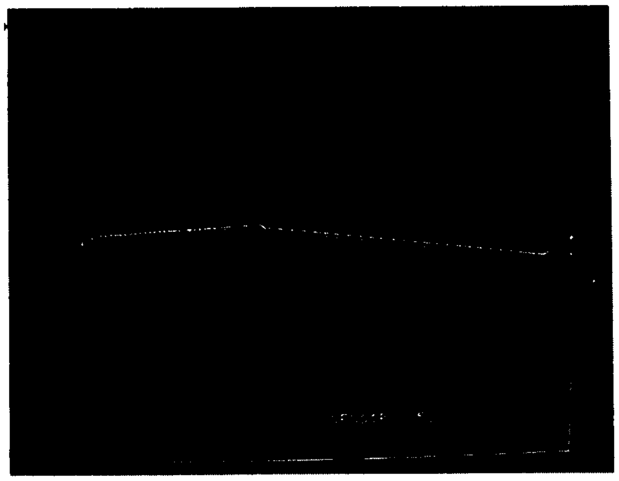
(a) Modified, f_t = .45, I-2



(b) Conventional, I-2

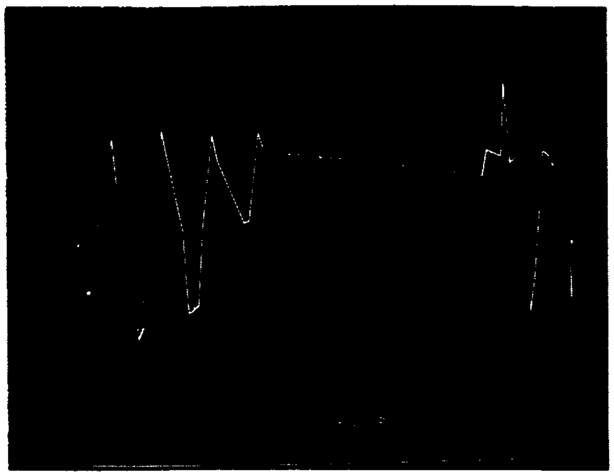


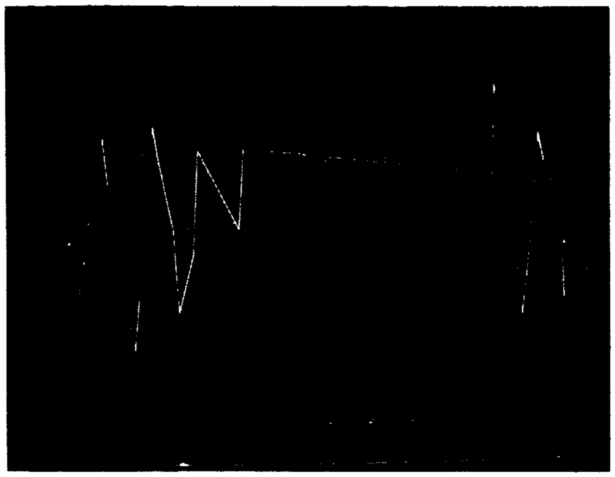
(c) Modified f_t = .45, I-2, with Conventional, I-2, Superimposed

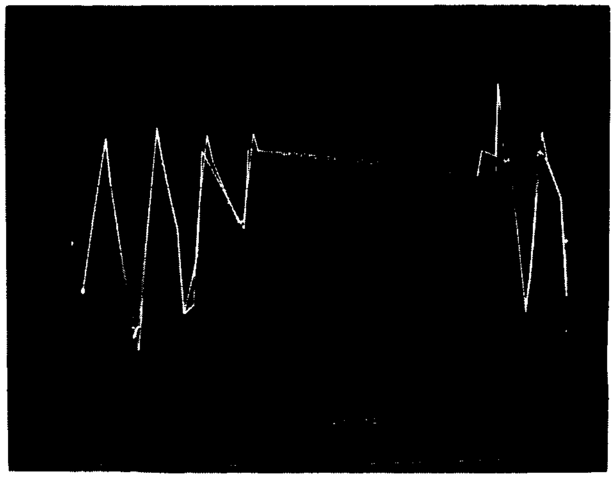


(d) Conventional, I-2S

Figure 7. Selected Runs for Sensor 5


(a) Modified, f_t = .45, I-2


(b) Conventional, I-2


(c) Modified f_t = .45, I-2, with Conventional, I-2, Superimposed

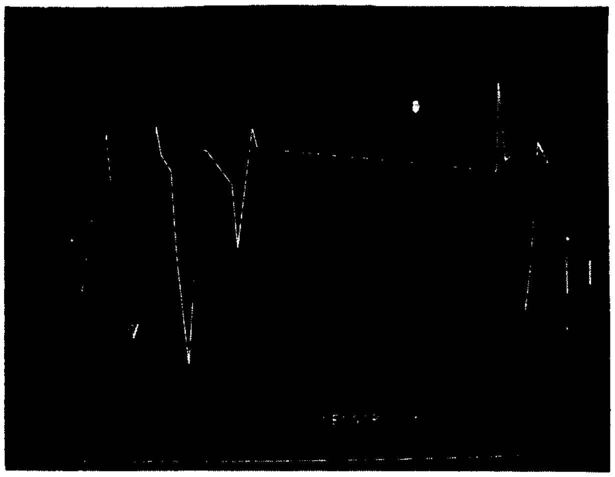
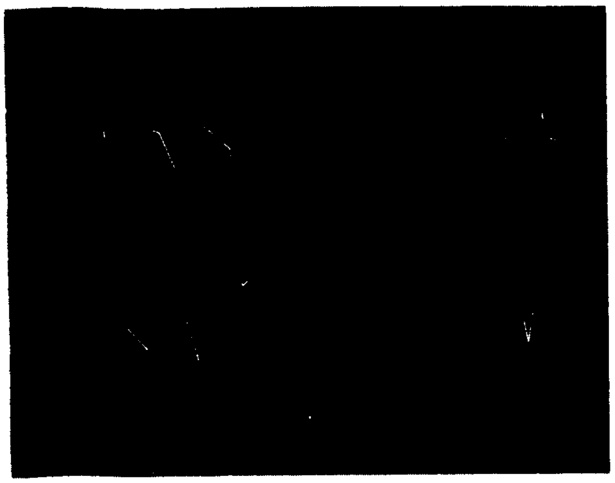
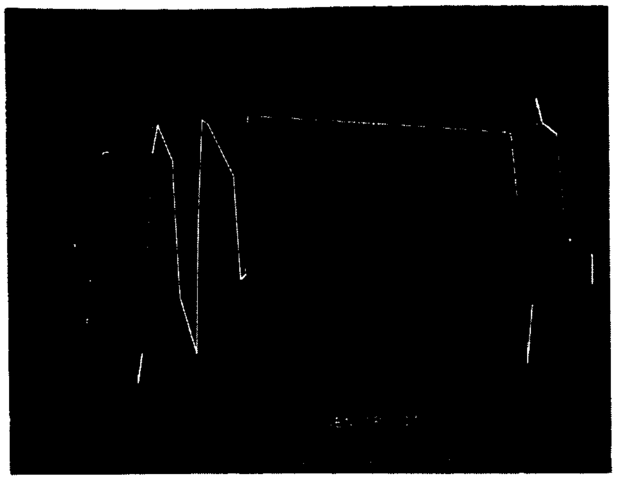

(d) Conventional, I-2S

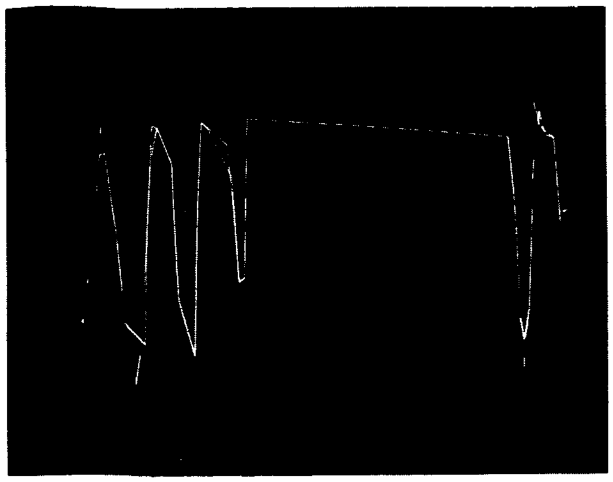
Figure 8. Selected Runs for Sensor 9



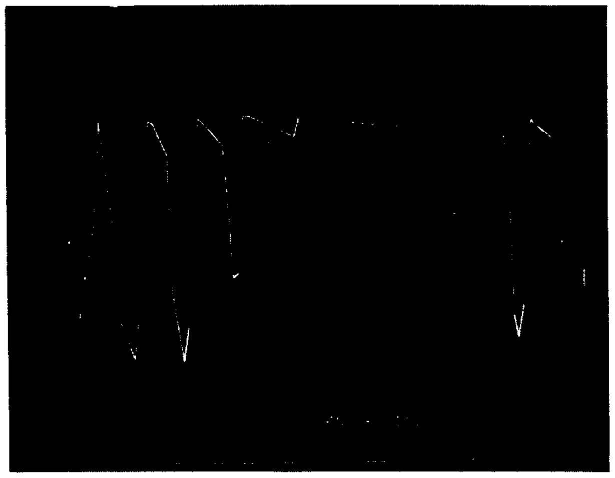
(a) Modified, f_t = .45, I-2



(b) Conventional, I-2

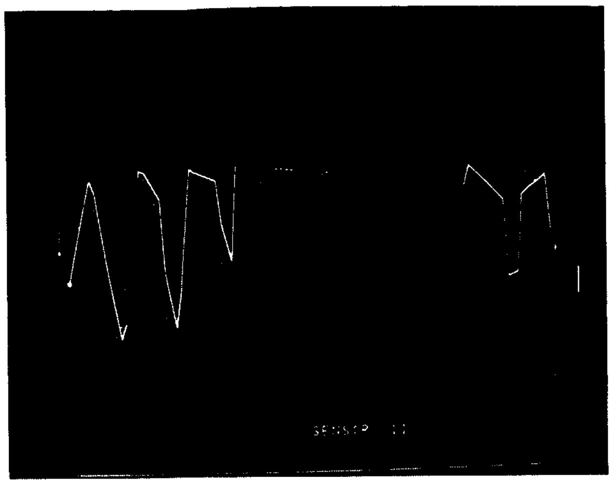


(c) Modified f_t = .45, I-2, with Conventional, I-2, Superimposed

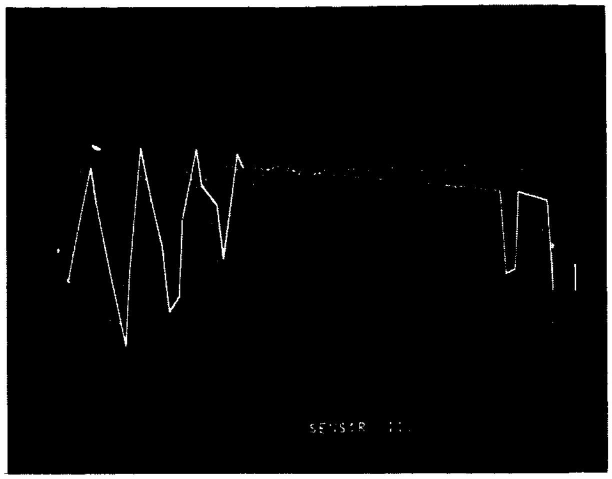


(d) Conventional, I-2S

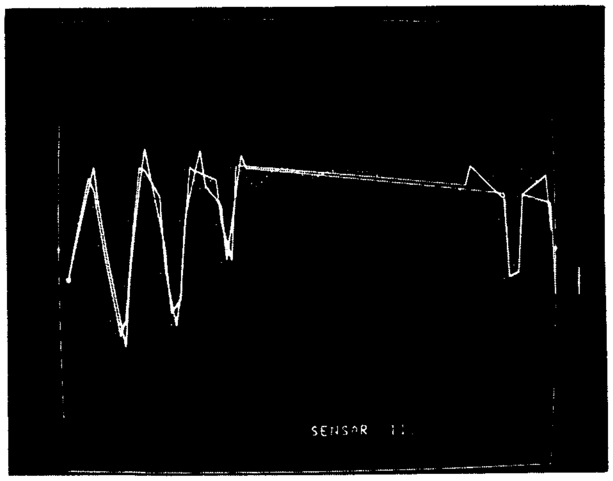
Figure 9. Selected Runs for Sensor 10



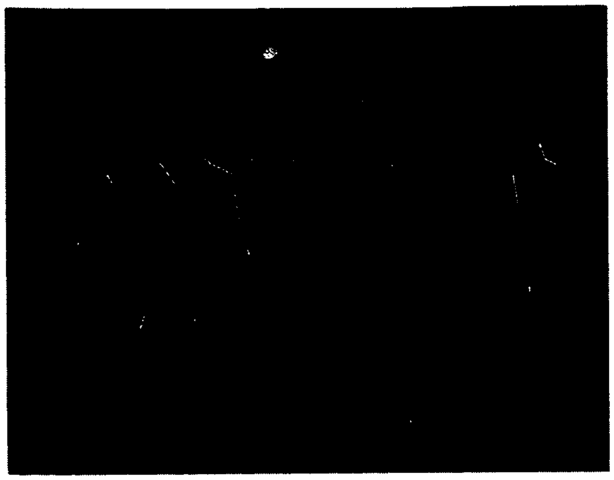
(a) Modified, f_t = .45, I-2



(b) Conventional, I-2

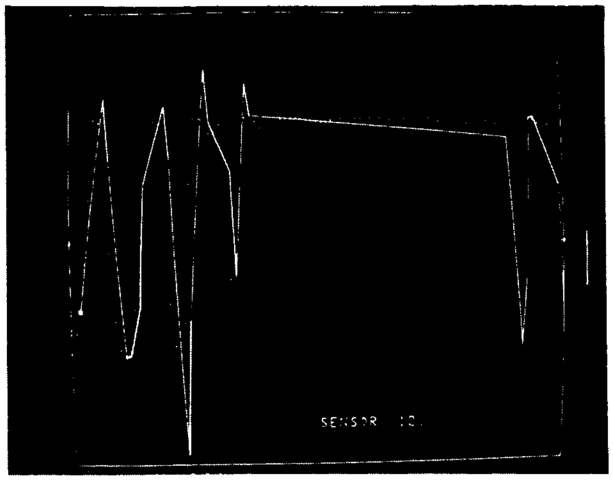


(c) Modified, f_t = .45, I-2, with Conventional, I-2, Superimposed

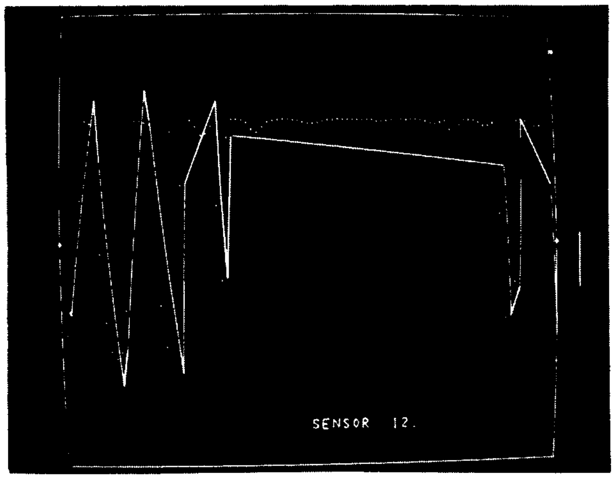


(d) Conventional, I-2S

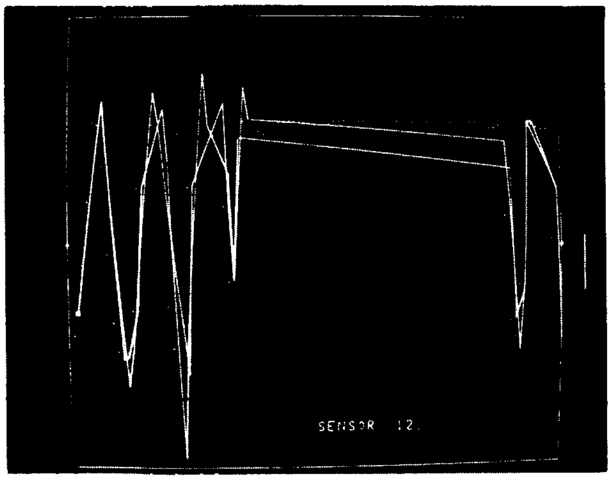
Figure 10. Selected Runs for Sensor 11



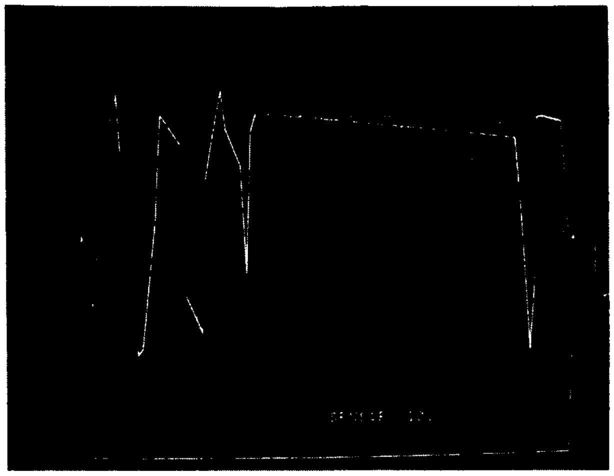
(a) Modified, f_t = .45, I-2



(b) Conventional, I-2

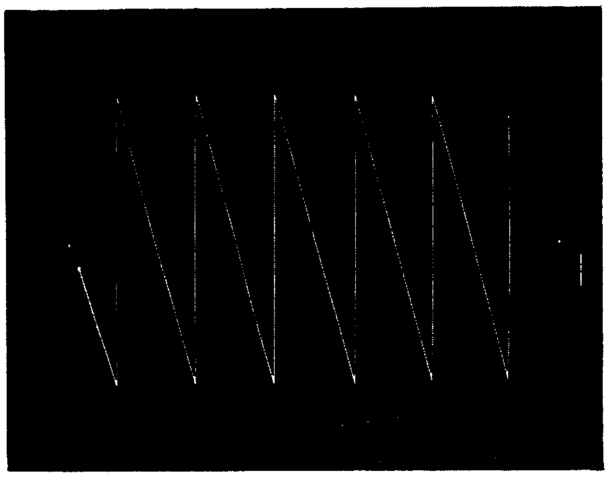


(c) Modified, f_t = .45, I-2, with Conventional, I-2, Superimposed



(d) Conventional, I-2S

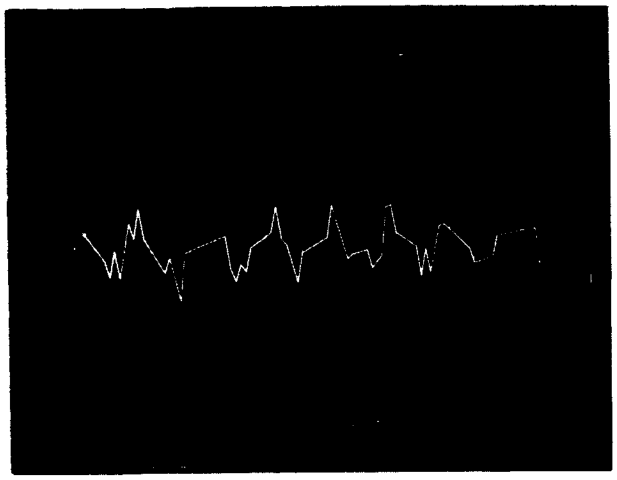
Figure 11. Selected Runs for Sensor 12

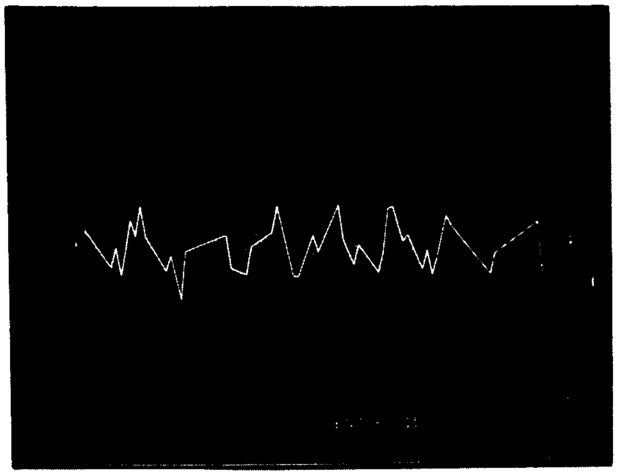


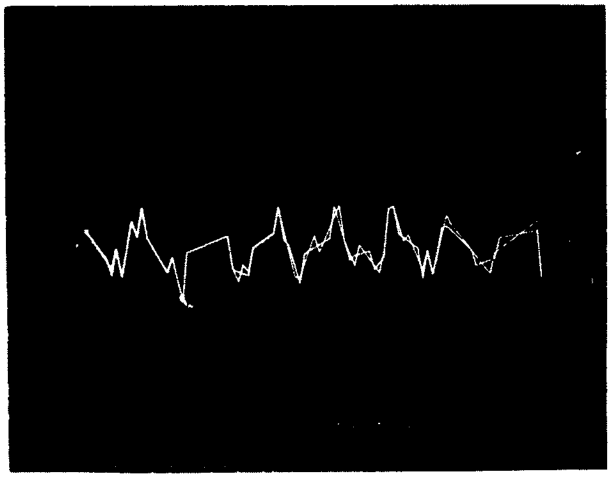
(c) Modified, f_t = .45, I-2 with Conventional, I-2, Superimposed

(a), (b), and (d) similar)

Figure 12. Selected Run for Sensor 13


(a) Modified, f_t = .45, I-2


(b) Conventional, I-2


(c) Modified, f_t = .45, I-2, with Conventional, I-2, Superimposed

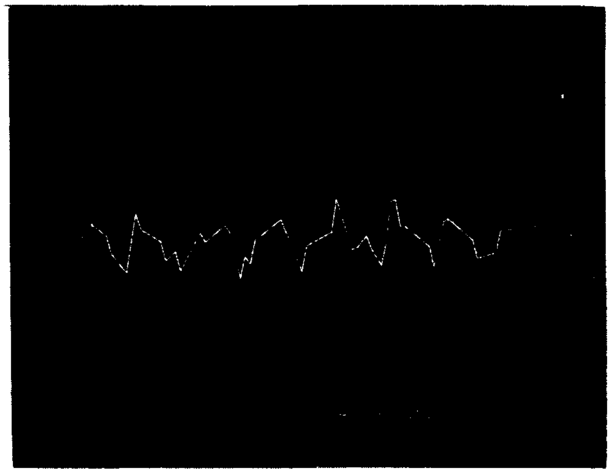

(d) Conventional, I-2S

Figure 13. Selected Runs for Sensor 15

Table 7. Results from 105 Standard Runs (page 1 of 4)

$f_t = 0$	1			1A			2			3			4		
	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	6	6	9	12	10	14	9	7	10	21	17	21	12	13	15
D	2	2	3	3	3	4	8	9	10	12	13	15	21	20	21
E	62	62	56	54	52	48	49	48	43	42	40	38	30	28	27
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	27	27	29	28	32	31	31	33	34	22	27	23	34	36	34
Max M	3	3	3	3	3	4	4	4	4	5	5	5	4	4	5
Max C	3	3	3	3	3	4	4	4	4	5	5	5	4	4	5
W	6	7	7	7	8	8	7	8	8	12	13	13	19	20	20
Tot 2M	45	45	56	61	61	72	75	75	86	104	104	115	130	130	141
Tot 1M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tot M	45	45	56	61	61	72	75	75	86	104	104	115	130	130	141
Tot C	45	45	56	61	61	72	75	75	86	104	104	115	130	130	141

Table 7. Results from 105 Standard Runs (page 2 of 4)

	1			1A			2			3			4		
	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III
$f_t = .2$															
A	1	3	6	1	2	4	0	1	1	1	2	4	1	1	1
B	6	7	6	3	6	4	7	7	10	4	6	7	6	7	11
C	4	2	4	8	8	9	7	8	7	15	13	15	8	9	10
D	1	1	2	2	1	3	6	5	8	14	14	15	21	20	21
E	59	58	47	49	49	39	46	44	39	40	39	34	30	29	22
F	13	13	16	12	11	17	8	11	13	5	6	8	4	5	6
G	13	13	16	22	20	21	23	21	19	18	17	14	27	26	26
Max M	3	3	3	3	3	3	4	4	4	5	5	5	5	4	4
Max C	3	3	3	3	3	4	4	4	4	5	5	5	4	4	5
W	6	7	7	7	8	8	7	8	8	12	13	13	19	20	20
Tot 2M	30	27	36	47	45	52	63	60	68	98	95	102	119	118	127
Tot 1M	21	26	34	17	21	29	15	20	25	11	16	23	12	14	19
Tot M	51	53	70	64	66	81	78	80	93	109	111	125	131	132	146
Tot C	45	45	56	61	61	72	75	75	86	104	104	115	130	130	141
$f_t = .4$															
A	6	9	15	7	6	12	3	6	7	4	4	9	2	3	3
B	9	5	9	5	13	10	15	14	20	10	11	11	8	11	17
C	1	1	5	5	3	7	8	9	11	13	13	18	10	11	10
D	0	0	0	2	0	2	2	1	4	12	11	13	20	19	20
E	54	49	40	48	44	33	44	42	33	36	34	23	27	27	16
F	20	27	23	15	19	22	10	9	16	7	10	14	8	7	16
G	7	6	5	15	12	11	15	16	6	15	14	9	22	19	15
Max M	2	2	2	3	2	3	3	3	3	4	4	4	4	4	5
Max C	3	3	3	3	3	4	4	4	4	5	5	5	4	4	5
W	6	7	7	7	8	8	7	8	8	12	13	13	19	20	20
Tot 2M	18	13	24	36	31	41	52	51	60	89	86	97	115	113	120
Tot 1M	41	50	62	34	44	56	31	35	50	25	29	43	20	24	39
Tot M	59	63	86	70	75	97	83	86	110	114	115	140	135	137	159
Tot C	45	45	56	61	61	72	75	75	86	104	104	115	130	130	141

Table 7. Results from 105 Standard Runs (page 3 of 4)

	1			1A			2			3			4		
	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III
$f_t = .45$															
A	8	9	11	6	4	10	8	8	12	3	3	11	2	4	5
B	8	6	12	7	13	15	16	17	22	11	12	10	8	10	17
C	0	1	3	3	3	6	7	6	9	14	15	19	10	11	10
D	0	0	0	2	0	1	0	0	3	11	9	12	20	19	20
E	52	48	34	46	41	30	40	38	33	34	31	21	27	26	15
F	21	27	32	18	24	26	11	12	11	10	14	14	8	7	14
G	8	6	5	15	12	9	15	16	7	14	13	10	22	20	16
Max M	1	2	2	3	2	3	2	2	3	4	4	4	4	4	5
Max C	3	3	3	3	3	4	4	4	4	5	5	5	4	4	5
W	6	7	7	7	8	8	7	8	8	12	13	13	19	20	20
Tot 2M	16	14	23	34	31	39	45	45	56	88	84	96	115	113	120
Tot 1M	45	51	66	37	45	61	43	45	57	27	32	46	20	25	41
Tot M	61	65	89	71	76	100	88	90	113	115	116	142	135	138	161
Tot C	45	45	56	61	61	72	75	75	86	104	104	115	130	130	141
$f_t = .5$															
A	8	14	23	6	11	15	9	11	15	2	4	14	3	4	8
B	8	6	12	7	12	15	15	17	24	12	12	16	7	9	21
C	1	0	3	3	2	7	7	6	8	13	11	16	10	11	10
D	0	0	0	2	0	0	0	0	4	9	10	11	20	19	20
E	53	48	31	46	42	27	39	36	21	28	27	14	27	25	11
F	21	23	26	19	18	25	10	10	21	17	17	17	7	8	16
G	6	6	2	14	12	8	17	17	4	16	16	9	23	21	11
Max M	2	1	2	3	2	2	2	2	3	4	4	4	4	4	5
Max C	3	3	3	3	3	4	4	4	4	5	5	5	4	4	5
W	6	7	7	7	8	8	7	8	8	12	13	13	19	20	20
Tot 2M	16	12	20	33	28	37	46	46	56	82	81	92	115	113	120
Tot 1M	45	57	84	38	52	70	43	49	75	33	37	61	20	25	53
Tot M	61	71	104	71	80	107	89	95	131	115	118	153	135	138	173
Tot C	45	45	56	61	61	72	75	75	86	104	104	115	130	130	141

Table 7. Results from 105 Standard Runs (page 4 of 4)

	1			1A			2			3			4		
f_t = .8	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III
A	17	28	63	10	19	49	16	26	51	10	19	41	7	14	31
B	10	8	11	16	22	25	16	19	21	13	18	25	15	21	31
C	0	1	2	4	1	3	8	9	11	10	14	13	12	13	17
D	0	0	0	0	0	1	1	0	0	8	6	10	19	18	16
E	42	26	6	36	22	6	34	19	5	23	12	1	18	10	1
F	24	32	15	25	30	13	18	22	9	16	19	7	12	13	1
G	4	2	0	6	3	0	4	2	0	17	9	0	14	8	0
Max M	1	2	2	2	2	3	3	2	2	4	4	4	4	4	5
Max C	3	3	3	3	3	4	4	4	4	5	5	5	4	4	5
W	6	7	7	7	8	8	7	8	8	12	13	13	19	20	20
Tot 2M	14	12	15	30	27	34	39	39	43	75	74	83	114	113	117
Tot 1M	68	96	152	61	90	136	66	93	132	49	75	114	41	62	94
Tot M	82	108	167	91	117	170	105	132	175	124	149	197	155	175	211
Tot C	45	45	56	61	61	72	75	75	86	104	104	115	130	130	141
f_t = ∞															
A	85	85	83	73	74	72	69	69	65	49	50	48	37	37	33
B	12	12	12	24	22	20	25	22	21	33	31	26	29	28	31
C	0	0	2	0	1	4	3	6	11	7	9	13	15	16	16
D	0	0	0	0	0	1	0	0	0	8	7	10	16	16	17
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Max M	1	1	2	1	2	3	2	2	2	4	4	4	4	4	5
Max C	3	3	3	3	3	4	4	4	4	5	5	5	4	4	5
W	6	7	7	7	8	8	7	8	8	12	13	13	19	20	20
Tot 2M	12	12	16	24	24	31	31	34	43	73	71	84	111	112	118
Tot 1M	182	182	178	170	170	164	163	160	151	131	131	122	103	102	97
Tot M	194	194	194	194	194	195	194	194	194	204	202	206	214	214	215
Tot C	45	45	56	61	61	72	75	75	86	104	104	115	130	130	141

which each case occurred is given, for cases A-G; the largest number of transmissions on a sampling time is given for the conventional ("Max C") and modified ("Max M") modes; the number of wild points "W" is given for the run of 100 sampling times; and totals are given over all sensors for the number of forced transmission in modified mode ("Tot 2M"), the number of transmissions as a result of a search in modified mode ("Tot 1M"), the sum of these two preceding totals ("Tot M") which is the total number of transmissions in modified mode, and the total number of transmissions in conventional mode ("Tot C").

Certain trends are evident from Table 7. The number of points transmitted in conventional mode shows an expected increase as the tolerances are made smaller, for either the inactive or the active group of sensors or for both; and similarly for the modified mode - except for the runs for $f_t = \infty$, where, until an appreciable number of D cases begin to show up, the number of transmissions is exactly K = 2 per sample time and hence is 2x97 = 194. (The first two sample times in a run do not produce transmissions, for mathematical reasons; and the last time in the run does not do so either, for programming reasons). In the FOIDIS algorithm, of course, a "transmission" represents the sending of a computed data point on the given sampling time and a real data point on the next sampling time; whereas, in the ZOI and FOIJON algorithms a "transmission" represents the sending of only one data point, which is a computed one.

The number of D cases is large for active-sensor options 3 and 4, and small or zero for options 1, 1A, and 2. Because of the mathematical definitions for these cases, there are no A, B, or F cases for $f_t = 0$ and no E, F, or G cases for $f_t = \infty$; also, Tot 1M = 0 for $f_t = 0$, and also for $f_t = 0$ Max M = Max C and Tot 2M = Tot M = Tot C. The values for Max C, W, and Tot C are independent of f_t but are repeated in each block of data for ready reference.

The effect of making one tolerance disproportionately small (option 1A for the active sensors) is similar to that of making all tolerances a bit smaller, with the exception that a few more D cases seem to result. Options I and II for the inactive sensors do not give significantly different results, but Option III does - presumably because of noise fluctuations in the last binary digit of the data, since the minimum binary resolution for this data as we scaled it is 1.00.

It is evident from these results that the behavior of the system can truly be varied from conventional ($f_t = 0$) to "pure" modified ($f_t = \infty$) simply by adjusting f_t; our criterion for what comprised an "optimum" setting of f_t was that Tot M should be made as small as possible without permitting D cases to occur.

For the higher values of f_t, Max M is almost always smaller than Max C by 1 or 2 unless there are many D cases, indicating that the "hard-limiting" is generally effective unless the system is rather heavily loaded.

For tolerance sets I-1, I-1A, I-2, II-1, II-1A, and II-2, conditions are pretty well optimized. For the other tolerance sets, either there are too many D cases or else too much last digit noise is being transmitted, so that the conventional and modified modes lead to much the same sort of results.

c. Conclusions

The following conclusions have been reached as an outcome of the simulation investigations:

(1) The modified approach is capable of providing a good fit to the raw data, at little if any penalty in terms of extra points needing to be transmitted, if the data source tolerances and f_t are properly chosen.

(2) The most satisfactory operation using the modified approach was obtained for $f_t = 0.45$. It is believed that the value of this threshold for optimum operation would not change much even if data characteristics were to change rather substantially. However, it may sometimes be desirable to adjust f_t up or down slightly in accordance with criteria deduced from Tot M and the number of D cases.

(3) Inactive data sources should be given a tolerance of at least four times the "minimum binary resolution" (the smallest increment possible between successive readings). Otherwise, fluctuations in the last binary position or two can cause a great many data points of dubious worth to be transmitted. This conclusion is similar to that reached in another study (Ref. 9), wherein a value of eight times the minimum binary resolution was suggested. A factor of eight might actually be as good as one of four in many cases on the basis of our results also; the factor of eight is preferable where the tolerances on the active sensors are also relatively large (e.e., Option 1), as otherwise the mathematical behavior of the modified approach algorithms will in this case tend to favor the inactive sensors somewhat over the active ones.

(4) Active data sources should be given a tolerance (ε) sufficiently large that the desired compression ratio R for that data source can be achieved. Initially, ε may be chosen as $\varepsilon = \max |y(t_i)-y(t_j)|$ where $k \le i, j \le k + R-1$ (the "maximum spread" over the R points beginning k), where the ZOI algorithm is being used. This same choice is also possible where one of the first-order algorithms is being used, but is less satisfactory since a straight-line data point sequence with an appreciable slope could yield an inappropriately large ε. The rule can still be applied if a straight line is fit to the first R data points by any plausible method which can be used without a preassigned tolerance, such as least-squares, and the points $y(t_k)$

are then replaced by new points $y(t_k) - y_f(t_k)$ where $y_f(t_k)$ is the value of the ordinate of the straight-line fit.

(5) After initial choices have been made for all tolerances, they can be changed individually according to whether the average number N of points in a sequence is greater than or less than R, perhaps simply by replacing ϵ by $\left(\frac{R}{N}\right)\epsilon$ every so often.

(6) To regulate the total number of points selected for transmission without changing the tolerances individually, a formula of the following type can be used after some fairly large number Q of sampling times has elapsed, to compute an instantaneous tolerance ϵ_{inst} based on the assigned tolerance ϵ:

$$\epsilon_{inst} = \epsilon \left(\frac{Q^{a+b} + 1}{(Q-Q_2 - Q_D)^a \, (Q-Q_D)^b} \right)^c = \epsilon \left(\frac{Q^{a+b} + 1}{Q_1{}^a (Q_1 + Q_2)^b} \right)^c$$

where

Q = total number of sampling times

$Q_1 = Q_A + Q_E + Q_F$

$Q_2 = Q_B + Q_C + Q_G$

Q_J = number of sampling times when case J occurred, where J = A, B, C, D, E, F, G as defined for the printout code.

Likely combinations for the exponents a, b, and c are:

	a	b	c
Formula 1	1	1	1
Formula 2	1	2	1
Formula 3	1	2	2

A formula of this type will not "blow up", and can increase the tolerances as rapidly as desired to keep down the number of D cases and even the number of forced transmissions. A somewhat similar type of formula (see Reference 12) has been suggested to modify the tolerances, depending on the length of a queue of data awaiting transmission in a buffer.

SECTION V
SYSTEM OPERATION

1. STEPS OF ACQUISITION CYCLE

The sequence of functions performed by the system in accomplishing the data acquisition task is called an acquisition cycle. The five steps of the cycle are described in the following paragraphs.

The time required to execute a complete acquisition cycle might be expected to vary from 100 to 1000 μsec, depending primarily on the speed of the strand memory, the choice of data compression algorithm, the number of significant bits, and the speed of the analog strand input converter.

a. Data Source Selection

The first part of each acquisition cycle is the multiplexing operation, the purpose of which is to select those data sources to be included in the cycle. This function is necessary because not all data sources are of interest at any given time and because those that are of interest will not in most cases have the same sampling frequency requirement.

The search capabilities of the associative computer can be used to accomplish a part of this function by simply performing equality searches over all or part of the "identity tag" field to locate predesignated subsets of data sources that are "active" during the current phase. Of the data sources identified in this manner, they may or may not be enabled during the current acquisition cycle depending on the sampling frequency requirements.

This requirement for enabling data sources according to their sampling frequency may be handled in several ways. First, it may actually be avoided due to the use of data compression techniques. It may be reasonable to simply enable all active data sources each cycle regardless of their required sampling rate. The redundant data points obtained from data sources that are being sampled at an unnecessarily high rate will be eliminated by the data compression algorithm.

If this approach proves undesirable for one reason or another, more elaborate procedures for enabling data sources may be used. For example one possible procedure requires the following information in the strand memory associated with each active data source:

Enable Count	Frequency of Measurement Divider	----------------------	Strand Memory

The "Frequency of Measurement Divider" is a fixed quantity that specifies the rate at which the data source is to be sampled and the "Enable Count" is used to determine if the data source is to be enabled on the current cycle.

Given that the data acquisition system performs acquisition cycles at some fixed repetition rate, the "frequency of measurement" of a given sensor is determined by dividing the repetition rate of the system by the "Frequency of Measurement Divider". For example, if the repetition rate of the system is 1000 cps and if a certain data source is to be measured 100 times per second, its "Frequency of Measurement Divider" should be the value 10. The actual enabling is done by the following steps:

(1) Initially the "Enable Count" field should have the same value as the "Frequency of Measurement Divider".

(2) Prior to each acquisition cycle, the "Enable Counts" of all strands are decremented.

(3) An equality search is then performed over all "Enable Count" fields using 00---0 as the search word.

(4) The enable F/F is set for all strands satisfying the above search.

(5) The "Frequency of Measurement Divider" field is copied into the "Enable Count" field in all of the above strands.

Thus it is seen that all data sources will be measured at a repetition rate stored within the corresponding strand memory. Extreme flexibility is achieved by this method since any multiple of the basic repetition rate of the system can be chosen. Also the measurement frequency for a given sensor can be changed by the general-purpose computer as desired by simply changing the contents of the "Frequency of Measurement Divider".

b. Measurement

When all data sources to be included in the current acquisition cycle have been enabled, the measurement step is initiated. This is simply a matter of transferring the present reading from the data source to the strand input converter. In the case of analog data sources, it requires energizing the sample-and-hold amplifier in the strand input converter; in the case of digital inputs it requires a transfer from the data source to the buffer register in the strand input converter; and in the case of pulse inputs it may require the disabling of the pulse input momentarily* so that the counter in the pulse input converter will not be changing during the input step. It is assumed that, in all cases, the data is available from the data source without delay so that the measurement step is very short compared to other steps of the acquisition cycle.

c. Input and Conversion

With the current readings stored in all strand input converters, the next step is to transfer these readings to a prespecified field of the strand memory. For digital and pulse inputs this is nothing more than a bit-by-bit data move

*If necessary, buffering can be provided such that no pulses will actually be lost by the system.

operation. However, for the analog inputs, analog-to-digital conversion is required. A detailed description of the method of performing analog-to-digital conversion simultaneously for all analog data sources was described in Section III. It has been estimated that conversion of 10-bit numbers will last up to 100 μseconds, with all analog sensor readings being converted simultaneously.

In addition to the analog-to-digital conversion it may be necessary to convert some of the digital readings. For example, Gray-to-binary conversion may be necessary on data received from angle-encoder-type data sources. A given conversion can be performed simultaneously over all such data sources, but different types of conversion must be performed sequentially.

d. Compression

When the new data point has been read in for each data source, one of a number of possible data compression algorithms is used to eliminate redundant data points. The algorithms are described in detail in Section IV, and sample programs of two of the algorithms are included as Appendix C. The data compression function includes comparison of each new data point with the past history of that data source,and the computation of figures of merit which indicate the significance of each new data point. These figures of merit are then used in the output step of the acquisition cycle to determine which data points are to be transmitted.

While the time required to perform the data compression function will vary with the specific algorithm used and the number of significant bits involved, a value of up to 400 μsec is a reasonable estimate.

e. Output

The final step of the acquisition cycle is to select data points on the basis of the figures of merit calculated in the compression step, and to transfer these data points to the general-purpose computer. The method of selecting data points is based on the examination of all figures of merit simultaneously to assure sending the most significant data points. This is done by interleaving minimum searches over the figure-of-merit field with readout operations to transfer the identified data point to the general-purpose computer until a specific number of data points have been transferred, or until the figure of merit located in this manner is larger than some prespecified value.

Each data point selected for output must be accompanied by an identification tag, which is stored in one field in each strand memory. A time tag must also be sent, but a single value will suffice for the entire set of data points selected each cycle.

2. ACQUIRED DATA HANDLING PROCEDURES

While one of the significant features of this data acquisition system is that it automatically matches the amount of data acquired to the current transmission capability, it is assumed that situations may still occur where some data must be stored for transmission at some future time. For instance, the transmitter may fail temporarily because of radiofrequency interference, or because the vehicle is passing through a planet shadow and its solar cells are not producing enough power to enable transmission. Therefore, a "mass memory" device such as a magnetic tape unit is assumed to be in use.

The three possible modes of operation that are required are described below:

(1) Acquiring Rate Equals Transmission Rate -- The most common situation is where the data-acquiring rate of the associative computer and the transmission rate are equal.

(2) Acquiring Rate Exceeds Transmission Rate -- In this case, more data is being gathered than can be handled by the transmitter. A complete communications blackout, for instance, will require that all gathered data be stored for later transmission. It is also possible that even though the transmitter is operable, the control procedures implemented in the data acquisition system still determine that some storing of data in the mass memory is preferable to attempting to "hard-limit" the amount of data acquired. For instance, if matching the acquiring rate to the transmission rate requires corridor widths in the compression algorithms that exceed the "recommended" corridor width to the extent that excessive "filtering" is being applied to the data, then the system may elect to automatically shift to the "acquiring rate exceeds the transmission rate" mode and must then store some of the acquired data for later transmission. The most "significant" data values acquired during each sampling cycle will still be transmitted; the others will be accumulated in a buffer area of the random-access memory of the GP computer. When the buffer area is filled, it will in turn be emptied into the mass memory.

(3) Transmission Rate Exceeds Acquiring Rate -- In this case, some of the data previously stored in the mass memory can share the transmission channel with the data currently being acquired. The data acquisition system will now operate as closely as possible to the "corridor width limits" mentioned previously, the buffer area will be used to read from the mass memory, and the GP computer will include as many data points from the buffer as is possible for each transmission after a sampling cycle. Each time the buffer becomes empty, a new block is transferred from the mass memory to the buffer area, until all stored data points have been transmitted. The system then reverts to mode (1), either by reducing the corridor width tolerances or by reducing the transmission rate.

3. SPECIAL APPLICATIONS

a. Real-Time Compression of TV Pictures

Raster-type instruments such as television cameras, certain types of radars, and scanning radiometers can be tied in to the associative computer to a single strand as with all other instruments, but if real-time operation is required, a speed problem exists. For example, the rate at which points are produced by a TV camera is in the neighborhood of a million or more points per second, but the maximum repetition rate of an associative data acquisition system is approximately 10,000 acquisition cycles per second. The solution proposed is to provide a number of strands equal to the number of data points in each row of the picture as illustrated in Figure 14.

Assuming a 200-point by 200-point picture, the first row of 200 points produced by the camera is distributed to the 200 strands, one point per strand. This same method of distribution is used for all rows of points produced, so that strand 1 receives all data points in column 1 of the picture, strand 2 receives all points in column 2, and strand 200 receives all points in the 200th column. Each strand performs analog-to-digital conversion and data compression on the data points it receives just as if it were connected to an independent source.

In order that the array of strands can perform A/D conversion and data compression on the data points of the n^{th} row of the picture and at the same time be receiving data points of the $(n+1)^{th}$ row, a second sample and hold amplifier must be provided in each strand input converter. Also, an improvement in the speed of operation of sample and hold amplifiers will be necessary to meet the 1-megacycle requirement that would exist for a TV camera producing 200- by 200-point pictures at a rate of 25 frames per second.

It appears that an associative computer can readily handle a TV camera producing 200- by 200-point pictures at the rate of 25 frames per second. The

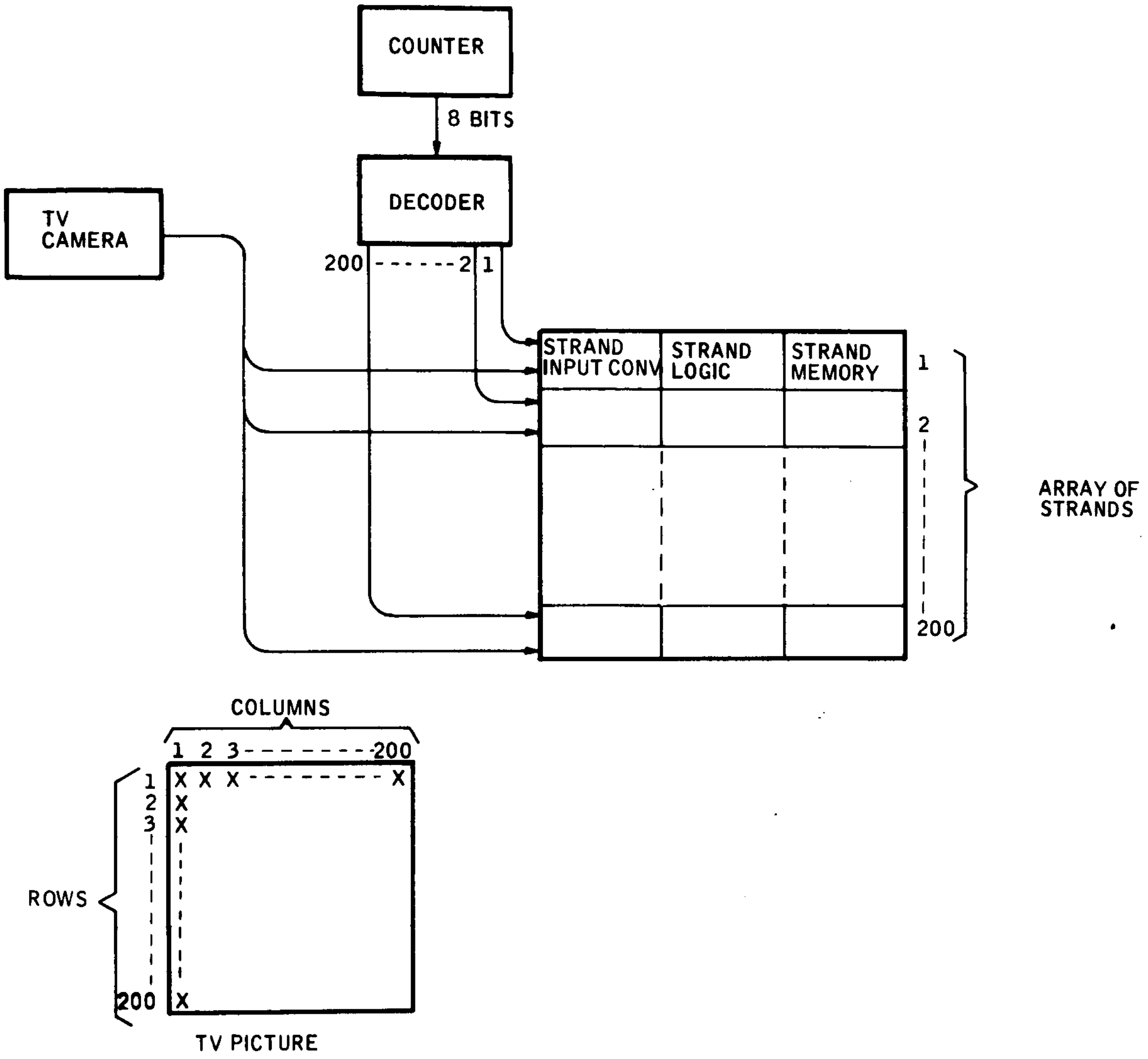


Figure 14. Real-Time Compression of TV Pictures

significant figures resulting from these assumptions are that points are being generated at a 1-megacycle rate and that an entire row of points is being generated every 200 μsec. This means that an acquisition cycle must be less than 200 μsec and that the sample and hold amplifiers in the strand input converters must be able to store a sample in less than 1 μsec. These rates appear to be achievable; however, the speeds necessary to handle 500-point-by-500-point pictures at 30 frames per second, which is more typical of television systems, could be achieved only with a significant improvement in the capabilities of sample-and-hold amplifiers.

The significant feature of the associative computer in this application is its capability of performing operations simultaneously on many data points - in this case on all the points along a row. This is true of both the analog-to-digital conversion and the data compression functions. It appears to be the only way such speed requirements can be met.

The algorithms described in Section IV of this report have been evaluated for use on video data (Ref. 12). A bandwidth compression ratio of up to "6 to 1" might be expected.

While the only computational function considered here was that of data compression, the general-purpose computational capabilities present in each strand can be used for various types of onboard processing of pictures as well.

b. System Status Monitoring

While in most sections of this report the scientific-sensor-type of data source has been assumed, the data source measuring system status is also of importance on any space mission. Actually there is no hard and fast line between what is status information and what is sensor information; in general, it may be assumed for present purposes that the status information is whichever data is not of interest in connection with the scientific investigation the vehicle is carrying out but is needed in order that the vehicle and its subsystems can be kept operating properly.

The mode of operation to be used by the data acquisition system for status-type data sources will, first of all, depend on whether the status condition is to be used onboard the vehicle or at the earth-based control point. For those data sources where corrective action is to be taken onboard as the result of a fault condition, an "out-of-tolerance" type comparison must be made during each acquisition cycle or at some slower rate. While any number of such conditions could occur during any one cycle, this will not overload the transmitter since these results are for onboard use only.

For transmission of system status information to earth, all data sources, including those given the "out-of-tolerance" test, will be treated in exactly the same manner as the sensor-type data sources. The earth-based control station will thus receive the entire history of each such data source and can perform its own "out-of-tolerance" comparison to determine when fault conditions exist. This allows the "all-sources-jointly" approach to data compression to include the system status sources as well as the scientific sensor sources.

c. Control of Complex Combination of Sensors

Future unmanned space vehicles will carry an elaborate set of scientific sensors and will be expected to perform many relatively complex experiments. In these cases, a certain amount of simple onboard decision making will be required to provide proper control over the instruments involved. For example, the value of the output obtained from one sensor might be used to enable other sensors or to adjust other sensors. Also, it might be necessary to accurately measure time intervals for many sensors simultaneously for purposes of controlling the time at which each takes a measurement.

The associative computer has capabilities that make it well suited to the performance of such functions. It can readily perform associations between combinations of sensors, since all sensors are monitored simultaneously and since the search capabilities allow examination of all results jointly. Also, since each strand of the associative computer has the capability of counting, each can measure a time interval for use in enabling a group of sensors relative to one another. This would involve the use of a special field in each strand memory that could be preset to any value and decremented each cycle as described in the first part of this section. The completion of the "countdown" can be used to turn on a sensor, initiate a measurement, or effect any operation that might be desired.

SECTION VI
SUMMARY OF RESULTS AND RECOMMENDATIONS

1. SUMMARY OF RESULTS

An associative data acquisition system has been described which is suitable for use on advanced space vehicles. The system is based on the use of an associative computer in which each data source is connected to a single word (strand). The structure and capabilities of the associative computer make it a natural fit to the problem. All three of the major data acquisition functions of multiplexing, data conversion, and data compression are performed within a single associative computer.

The system design of the associative computer portion of the proposed system was performed. The design effort was directed at the array of strands, where each of the 256 strands consists of the following:

(1) A strand memory with 192 memory elements which are randomly accessible for data transfers with the strand logic. The memory element should be fast, consume little power, and be non-volatile. Magnetic elements such as plated wire appear suitable.

(2) Strand logic which includes four flipflops for storage of operands and results of operations, approximately 45 gates to provide the necessary arithmetic, comparison, and transfer capabilities, a sense amplifier to allow reading from the strand memory, and a write driver to allow writing into the strand memory.

(3) A strand input converter which, in the case of analog inputs, contains the analog circuits needed to perform analog-to-digital conversion by the sequential approximation method.

Another significant part of the system design effort was concerned with the control unit. A microprogram approach was taken to provide control sequence flexibility and logic design simplicity.

The method of handling the data compression function in the associative data acquisition system is slightly different than in a conventional system. The search and processing capabilities of the associative computer allow what is called the modified approach to execution of the algorithms. This modified approach allows the placement of a limit on the number of data points selected for transmission each cycle by performing search operations over the results of the data compression algorithm. A simulation evaluation of this approach using real data showed that the modified approach represented the data as well as the conventional use of the algorithms for the same number of data points sent. The modified approach reduces buffering problems since it allows a match between the data acquiring rate and the data sending rate of the system.

Probably the most significant single feature of the associative data acquisition system is its cellular or repetitive structure. This gives it significant speed advantages over conventional systems since the operations are performed simultaneously over all data sources. This advantage makes it reasonable to consider on-line compression of TV pictures. The cellular structure also makes the system flexible since varying numbers of strands can be included to fit the specific requirements of any mission. Another important advantage of the cellular structure is that it makes mechanization amenable to batch fabrication techniques. This in turn provides potential advantages in reliability, cost, size, and weight.

2. RECOMMENDATIONS FOR FURTHER WORK

There are a number of areas where future work should be directed to realize the full potential of the associative data acquisition system. The following are described briefly:

(1) Detailed prefabrication design

(2) Application in biomedical instrumentation systems

(3) Construction and checkout of a complete system

(4) Design studies of using associative computers for data compression in real-time remote TV systems

(5) Hardware development for high-speed distributed A/D conversion

(6) Control of complex sensors

(1) Detailed prefabrication design -- The type of design and simulation being done during Phase II did not result in a complete design which could be immediately fabricated. The logic design being performed is mostly at the system level and only partly at the logic-element level; and the simulation being performed is of the over-all functions performed by the associative computer and the execution of algorithms, rather than the flow of control and information signals through the logic.

Hence it is proposed that the present design efforts be carried further to the obvious next step of performing a detailed logic design and a bit-time-by-bit-time logic simulation. This simulation would not be at the level of executing algorithms, but strictly at the level of executing associative computer microsteps; its purpose would be that of ensuring the correctness of the logic equations defining the design. Concurrent with this effort, microprograms for all common associative computer instructions would be coded; these would also be checked out by simulation. Both the bit-time-by-bit-time logic simulation and the checkout of microprograms would be carried out using a digital computer.

This task is prerequisite to performing task (3).

(2) Applications in biomedical instrumentation systems -- The instruments which have been investigated during Phase II of this contract have all been spaceborne instruments, such as might be used on an exploratory

orbiter, lander, or deep space probe, with some minor attention being given to surface exploration vehicles. Another area where the associative computer approach looks as if it might be fruitful is that of biomedical instrumentation systems.

Hence, the survey could reasonably be extended to the instruments customarily used in biomedical work at such facilities as the Honeywell Systems and Research Division Manned Systems Sciences Section, the University of Minnesota Medical School, the Mayo Clinic, and the NASA Manned Space Flight Center in Houston. This study would be greatly aided by the ready availability of consulting help from Honeywell biomedical researchers and experimental psychologists of the Manned Systems Sciences Section of the Research Department.

Task (2) might also, under the circumstances indicated below, be a prerequisite to task (3).

(3) Construction and checkout of a complete system -- Given the performance of task (1), and possibly that of task (2) if the intended application were in a biomedical instrumentation system, our technology has reached the point that construction and checkout of a realistic and operationally useful associative computer can then follow. This associative computer would be developed complete with the required analog-to-digital conversion apparatus; and Honeywell would, if desired, furnish a computer suitable for tie-in as the cooperating general-purpose computer.

To avoid the necessity for confronting the special problems of severe-environment component choice and packaging at this stage, the solution of which would considerably increase the cost of the prototype, it is recommended that the application for this prototype system be chosen from the biomedical field, since such a choice would usually permit the system to be used on the ground in a relatively favorable environment.

On the other hand, if it were desired, a space-vehicle data processing system could instead be chosen as the prototype application and a proper aerospace packaging job could be performed.

In the latter event, task (3) would only depend on the prior completion of task (1) and not on that of task (2).

(4) Design studies of using associative computers for data compression in real-time remote TV systems -- A very important problem now on the horizon is the transmission of real-time TV information back to earth from space vehicles exploring other celestial bodies. If performed in the most straightforward way without the use of data compression, this type of video transmission would involve tremendous data rates, possibly as high as a billion bits a second. Such a rate is tolerable for the foreseeable future, and obviously some form of data compression will have to be employed to make such transmission feasible.

One possible technical approach is to use an associative computer in the manner described in Section V, and in effect assign one "column" of a TV picture composed of raster lines in the "row" direction to each individual strand. For instance, if there were 200 raster lines in the Y direction, and an equal resolution were desired in the X direction, the associative computer would have 200 strands. This approach can provide compression of still pictures or individual video frames.

A number of variations of the above approach can be used to obtain compression between one picture or frame and the next. The data compression algorithms in this case can be applied to corresponding data points on successive pictures rather than along columns of points within a picture. This approach, while more difficult in terms of hardware requirements, would be particularly effective in the situation where the TV camera is in a fixed position, since large portions of the picture could be expected to remain constant for relatively long periods. For instance, a sequence of frames might show a fixed landscape with an astronaut walking across it, and only the portion of the picture showing the astronaut would have to be sent for each successive frame. Since it would be unreasonable to provide an associative computer strand for each point of the picture, some method of time-sharing of strands would have to be used. In situations where the camera is moving, these same techniques might still be used, by noting that the picture has been translated horizontally or vertically by a certain number of resolution-matrix points between one frame and the next, so that the repeated portions of the earlier frame can be repositioned and reused.

Various combinations of within-a-frame and frame-to-frame compression techniques could certainly provide an enormous degree of video data compression. It is recommended that these techniques be studied, and the form of the required associative computer be investigated.

(5) <u>Hardware development for high-speed distributed A/D conversion</u> -- If task (3) were felt at this stage to represent too large a program, the analog-to-digital portion of it could reasonable be broken out as a separate smaller study project. In the sequential approximation approach described in Section III, speed and accuracy are two potential problem areas that should be investigated.

(6) <u>Control of Complex Sensors</u> -- The associative computer has capabilities that make it well suited to the performance of relatively tight control over sensors. It can simultaneously monitor all sensors, perform various types of searches over results of computations, and adjust the parameters of each sensor to fit the specific situation and the specific experiment being performed.

A study of complex instruments should be performed to determine the specific types of control required and the ways in which this control can be used to maximize the use of the scientific payload. Methods of providing this control by taking advantage of the unique capabilities of the associative computer should then be investigated.

REFERENCES

1. P. R. Swan, et al., Manned Mars Surface Operations, AVCO/RAD, Wilmington, Massachusetts; AVCO/RAD-TR-65-26, Contract No. NAS 8-11353, 30 September 1965.

2. D. R. Williams, T. Samaras, M. L. Richter, Jr., Space Measurements Survey for DASA, Electro-Optical Systems, Inc., Pasadena, California, Volume I, No. 1, July 1962; Volume II, 28 February 1965. EOS Inc., Report No. 1890, DASA Report No. 1277, WEB No. 07.013, Contract Number DA-49-146-XZ-100. Volumes I and II have since been issued, along with additional material intended to comprise a Volume III, as a NASA Special Publication, Instruments and Spacecraft, edited by H. L. Richter, Jr., NASA SP-3028, NASA Scientific and Technical Information Division, Washington, D.C., 1966.

3. C. T. Benfield, NASA PATTERN Relevance Guide - Volume III, Honeywell Aeronautical Division, St. Petersburg, Florida; Honeywell Document No. 8-20207-RG, Contract No. NAS 8-20207, 15 October 1965.

4. G. W. Longanecker, D. J. Williams, R. O. Wales, Small Standard Satellite (S^3) Feasibility Study, NASA Goddard Space Flight Center, Greenbelt, Maryland; Goddard Document No. X-724-66-120, March 1966.

5. A. Egger, A Study to Determine an Efficient Data Format and Data System for a Lightweight Deep Space Probe, Research Report No. 1, TRW Systems, TRW, Inc., Redondo Beach, California, Contract No. NAS 2-3254, 18 February 1966.

6. D. C. Gunderson, C. W. Hastings, G. J. Prom, Interim Report-Spaceborne Memory Organization, Contract Number NAS 12-38, Honeywell Systems and Research Division, December 1965.

7. D. C. Gunderson, C. W. Hastings, G. J. Prom, Phase I Final Report-Spaceborne Memory Organization, Contract Number NAS 12-38, Honeywell Systems and Research Division, April 1966.

8. D. C. Gunderson, C W Hastings, H. R. Holt, Spaceborne Memory Organization - An Associative Data Acquisition System, Phase II, Interim Report, Contract Number NAS 12-38, Honeywell Systems and Research Division, September 1966.

9. W. R. Bechtold, J. E. Medlin, D. R. Weber, Final Report - PCM Telemetry Data Compression Study, Phase I, Contract Number NAS 5-9729, Lockheed Missiles and Space Company, October 1965.

10. D. R. Weber, "A Synopsis on Data Compression", Proceedings of the 1965 National Telemetering Conference, April 1965, pages 9-16.

11. R. S. Simpson, C. A. Blackwell, W. O. Frost, "Compendium of Redundancy Removal Processes", IEEE Transactions on Aerospace and Electronic Systems, July 1966, pages 471-474.

12. L. C. Bunyan, W. P. Hogan, D. R. Weber, Final Report - TIROS Video Data Compression Study, Contract Number NAS 5-9569, Lockheed Missiles and Space Company, December 1965.

13. Analog-Digital Conversion Handbook, Digital Equipment Corporation, Publication No. E-5100, 1964.

APPENDIX A
DESCRIPTION OF SIMULATION PROGRAM

Ralph S. Livingston

APPENDIX A
DESCRIPTION OF SIMULATION PROGRAM

1. INTRODUCTION

The data compression program is a simulation of two data compression algorithms. The algorithms are referred to as the conventional and modified version of the FOIDIS (First-Order Interpolator, DISjoined line segment) algorithm.

The program uses actual experimental data as input data to the algorithms. The output from the program are those points, as determined by the algorithms, which best represent the data.

The program has the capability of rejecting raw data points which are wild points. These points do not affect or influence the algorithm.

A display capability is included in the program as a visual aid to the analysis in evaluating and comparing the algorithms.

2. PARAMETER LIST

a. Arrays

BASE (I, J) - Base Points
EPS (I) - Epsilons
EPOS (I, J) - Top Slope
ENEG (I, J) - Bottom Slope
FM (I) - Figure of Merit
IAFL (I, J) - Flags (Indicate if Last Point was Transmitted)
IM (I) - Modified } Output Indicator
IN (I) - Conventional } Array

IW (I) - Wild Point Indicator

OUT (K, I, J) - Output Array (Contains All Transmitted Values)

SU (I) - Average Slope (EPOS, ENEG) for Last Time Interval

SI (K, I) - Raw Data

TIME (I, J) - Time Elapsed (Number of Time Intervals) Since Last Transmit

b. Singles

SIK - Present Sensor Value Being Processed

TOP - Value of Point on EPOS at Time K

BOT - Value of Point on ENEG at Time K

FML - Value of Maximum Figure of Merit which will be sent in Modified Algorithm

IMC - Count of Required Transmits in Modified Algorithm ($FM \leq O$)

IMCC - Total of IMC's for Run

INC - Count of Required Transmits in Conventional Algorithm

INCC - Total of INC's for Run

IMD - Count of Transmits Due to Small FM's (Modified)

IRUN - Run Number

K - Present Time Interval Being Processed

KM = K-1

KP = K+1

3. OPERATIONAL INFORMATION

a. Peripheral Equipment

3 - Raw Data Tape (Binary)

4 - Processed Data Tape (Binary)

5 - Card Reader

7 - Typewriter

9 - Line Printer

b. Card Input

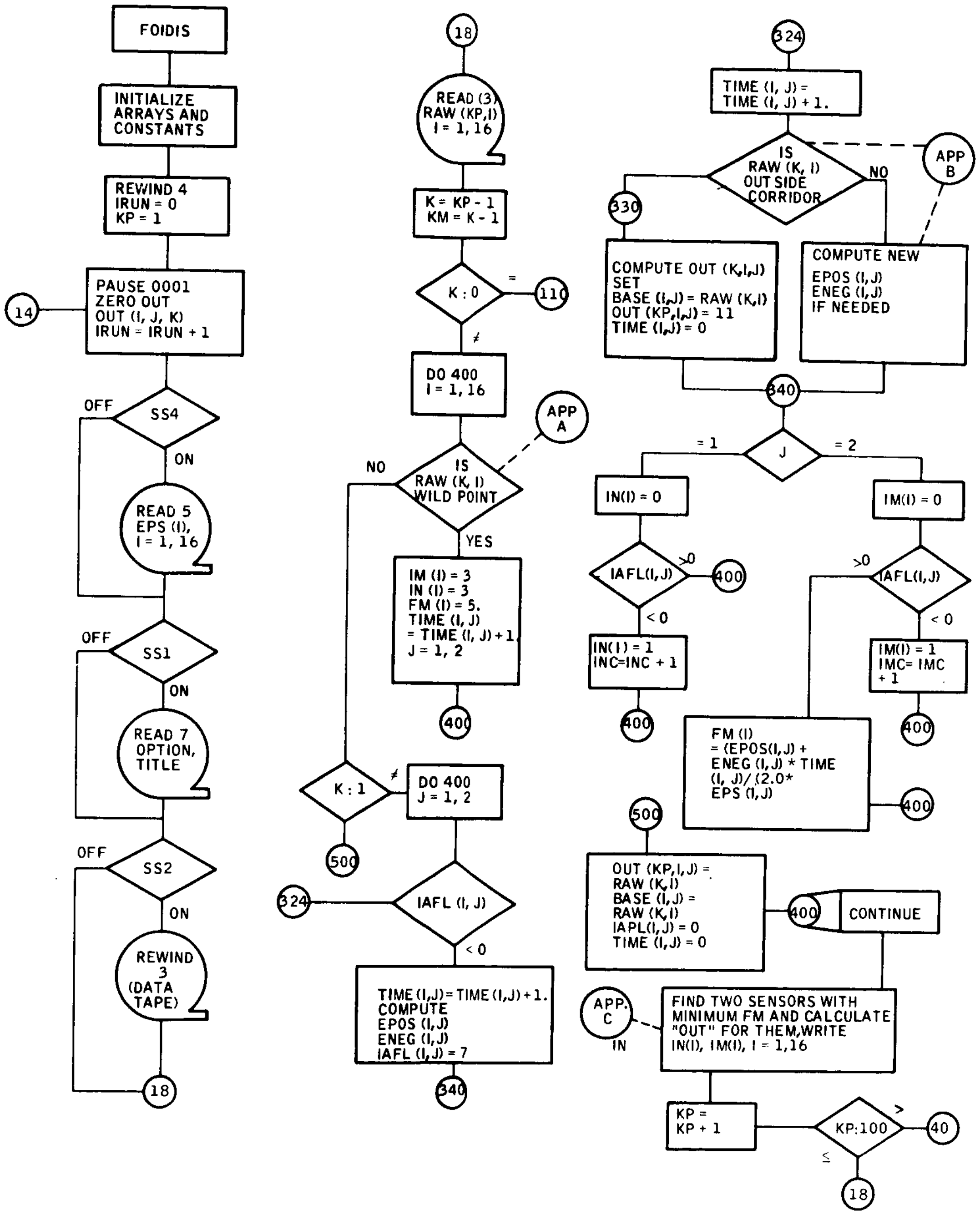
Card 1 - N	Information Printed on Typewriter
Card N + 1	/in Column 1
Card N + 2	/in Column 1
Card N + 3	EPS (I), I = 1, 8
Card N + 4	EPS (I), I = 9, 16

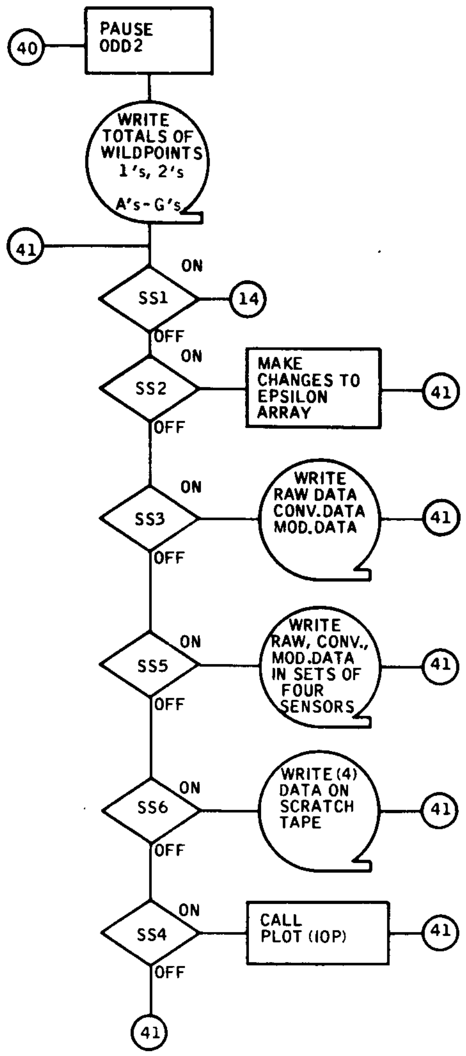
c. Sense Switch Options on Conditions

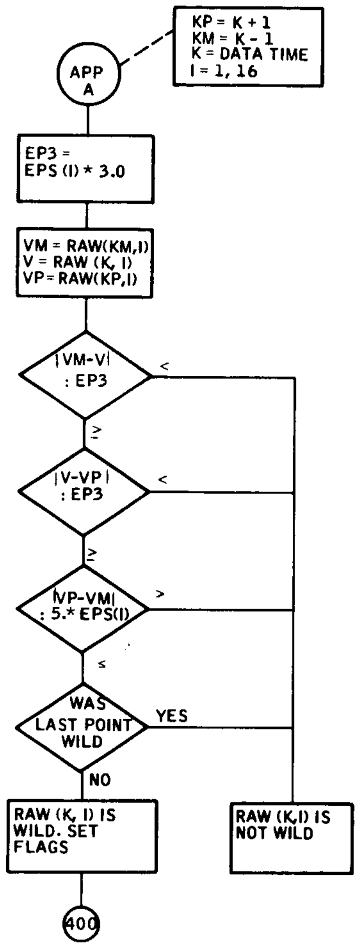
	At Pause 1	At Pause 2	In Subroutine Plot
Sense Switch 1	Do Not Read New Title or Option	Return to Pause 1	
Sense Switch 2	Do Not Rewind the Raw Data Tape	Make Changes to the Epsilon Array via the Typewriter	Display Conventional Algorithm
Sense Switch 3	P Array Option	Old Style Block Output	Display Modified Algorithm
Sense Switch 4	Do Not Read A New Set of Epsilons	Call Plot (Scope Routine)	Stay in this Routine (Do Not Return)
Sense Switch 5	Read in a New FM (Figure of Merit) Output Limit	Write Out Data on Printer	Do Not Need a New Sensor (Redisplay as per SS2 - SS3)
Sense Switch 6		Write Data on a Binary Work Tape for Later Use	Display the Plot on the Scope on the Line Printer

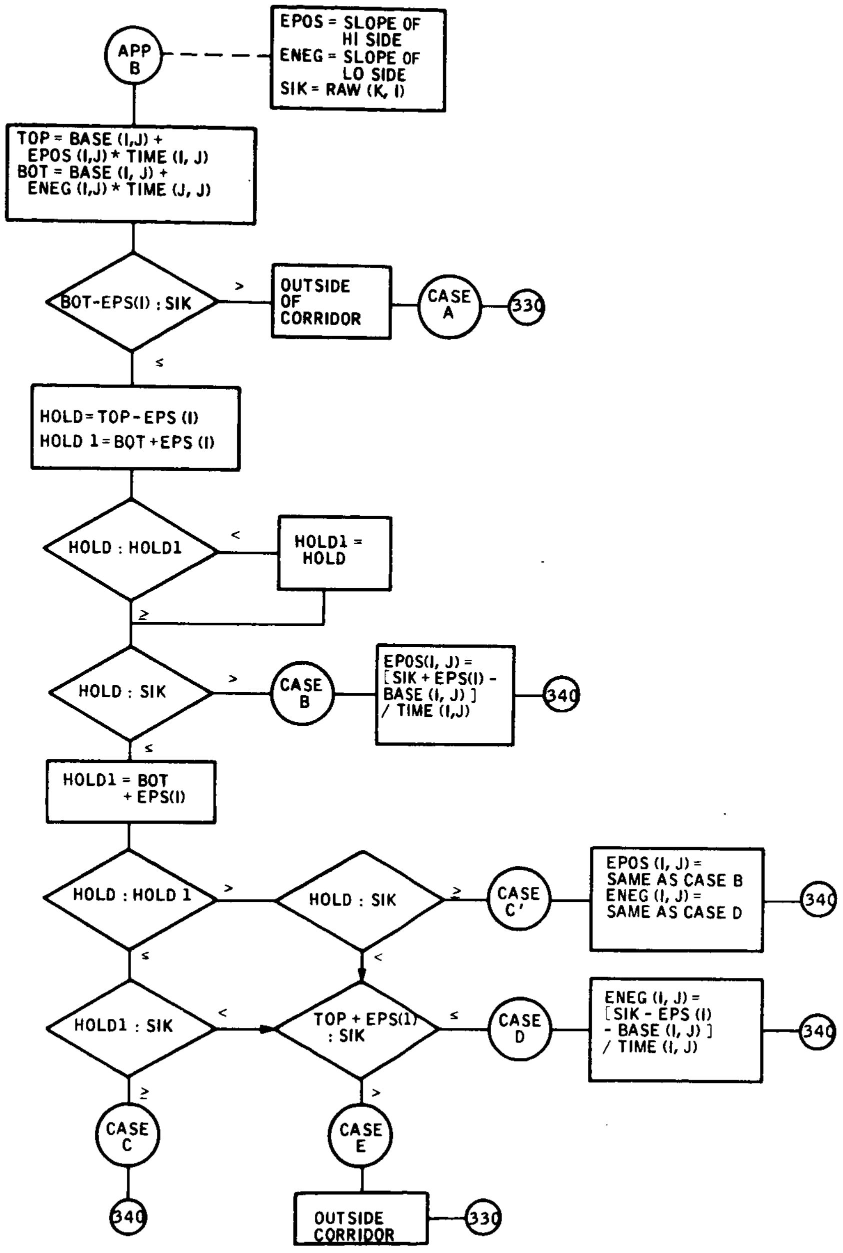
4. FLOW CHART AND PRINTOUT

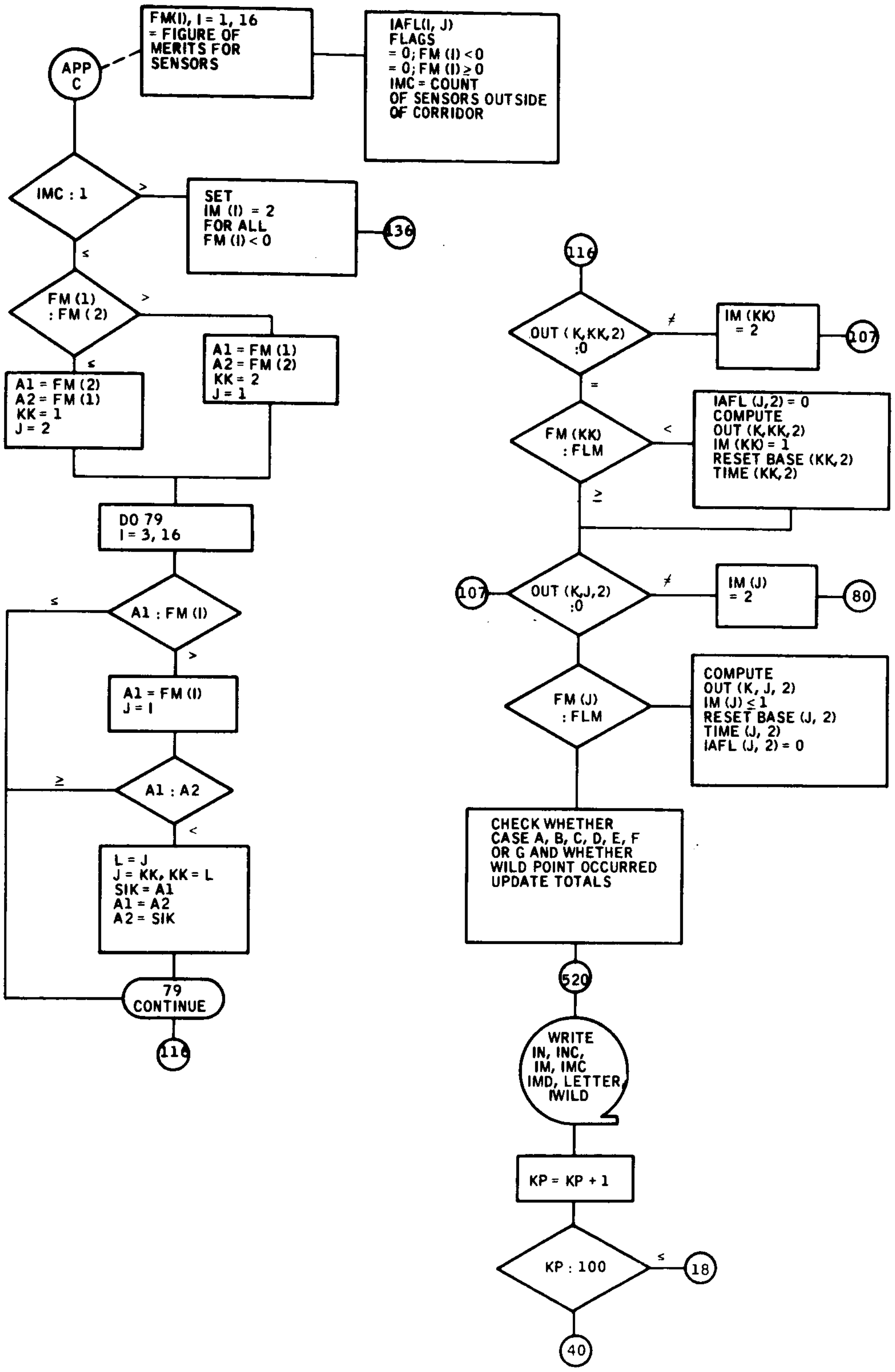
The flow chart and printout for the simulation program are shown on the following pages.


FOIDIS
INITIALIZE ARRAYS AND CONSTANTS
REWIND 4 IRUN = 0 KP = 1
14
PAUSE 0001 ZERO OUT OUT (I, J, K) IRUN = IRUN + 1
OFF
SS4
ON
READ 5 EPS (I), I = 1, 16
OFF
SS1
ON
READ 7 OPTION, TITLE
OFF
SS2
ON
REWIND 3 (DATA TAPE)
18
18
READ (3) RAW (KP,I) I = 1, 16
K = KP - 1 KM = K - 1
K : 0
=
110
≠
DO 400 I = 1, 16
APP A
NO
IS RAW (K, I) WILD POINT
YES
IM (I) = 3 IN (I) = 3 FM (I) = 5. TIME (I, J) = TIME (I, J) + 1 J = 1, 2
400
K : 1
≠
DO 400 J = 1, 2
500
324
IAFL (I, J)
< 0
TIME (I,J) = TIME (I,J) + 1. COMPUTE EPOS (I,J) ENEG (I,J) IAFL (I,J) = 7
340
324
TIME (I, J) = TIME (I, J) + 1.
IS RAW (K, I) OUT SIDE CORRIDOR
NO
APP B
330
COMPUTE OUT (K,I,J) SET BASE (I,J) = RAW (K,I) OUT (KP,I,J) = 11 TIME (I,J) = 0
COMPUTE NEW EPOS (I,J) ENEG (I,J) IF NEEDED
340
= 1
J
= 2
IN(I) = 0
IM(I) = 0
IAFL(I,J)
>0
400
< 0
IN(I) = 1 INC=INC + 1
400
>0
IAFL(I,J)
< 0
IM(I) = 1 IMC= IMC + 1
400
FM (I) = (EPOS(I,J) + ENEG (I,J) * TIME (I, J)/(2.0* EPS (I,J)
400
500
OUT (KP,I,J) = RAW (K,I) BASE (I,J) = RAW (K,I) IAPL(I,J) = 0 TIME (I,J) = 0
400
CONTINUE
APP. C
IN
FIND TWO SENSORS WITH MINIMUM FM AND CALCULATE "OUT" FOR THEM, WRITE IN(I), IM(I), I = 1,16
KP = KP + 1
KP:100
>
40
≤
18


40
PAUSE ODD2
WRITE TOTALS OF WILDPOINTS 1's, 2's A's-G's
41
SS1
ON
14
OFF
SS2
ON
MAKE CHANGES TO EPSILON ARRAY
41
OFF
SS3
ON
WRITE RAW DATA CONV.DATA MOD.DATA
41
OFF
SS5
ON
WRITE RAW, CONV., MOD.DATA IN SETS OF FOUR SENSORS
41
OFF
SS6
ON
WRITE (4) DATA ON SCRATCH TAPE
41
OFF
SS4
ON
CALL PLOT (IOP)
41
OFF
41


APP A
KP = K + 1
KM = K - 1
K = DATA TIME
I = 1, 16
EP3 = EPS (I) * 3.0
VM = RAW(KM,I)
V = RAW (K, I)
VP = RAW(KP,I)
|VM-V| : EP3
<
≥
|V-VP| : EP3
<
≥
|VP-VM| : 5.* EPS(I)
>
≤
WAS LAST POINT WILD
YES
NO
RAW (K, I) IS WILD. SET FLAGS
RAW (K,I) IS NOT WILD
400


APP B
EPOS = SLOPE OF HI SIDE
ENEG = SLOPE OF LO SIDE
SIK = RAW (K, I)
TOP = BASE (I,J) + EPOS (I,J) * TIME (I, J)
BOT = BASE (I, J) + ENEG (I,J) * TIME (J, J)
BOT-EPS(I) : SIK
>
OUTSIDE OF CORRIDOR
CASE A
330
≤
HOLD = TOP - EPS (I)
HOLD 1 = BOT + EPS (I)
HOLD : HOLD1
<
HOLD1 = HOLD
≥
HOLD : SIK
>
CASE B
EPOS(I, J) = [SIK + EPS(I) - BASE (I, J)] / TIME (I,J)
340
≤
HOLD1 = BOT + EPS(I)
HOLD : HOLD 1
>
HOLD : SIK
≥
CASE C'
EPOS (I, J) = SAME AS CASE B
ENEG (I, J) = SAME AS CASE D
340
≤
<
HOLD1 : SIK
<
TOP + EPS(1) : SIK
≤
CASE D
ENEG (I, J) = [SIK - EPS (I) - BASE (I, J)] / TIME (I, J)
340
≥
CASE C
340
>
CASE E
OUTSIDE CORRIDOR
330


APP C
FM(I), I = 1, 16 = FIGURE OF MERITS FOR SENSORS
IAFL(I, J) FLAGS = 0; FM (I) < 0 = 0; FM (I) ≥ 0 IMC = COUNT OF SENSORS OUTSIDE OF CORRIDOR
IMC : 1
>
SET IM (I) = 2 FOR ALL FM (I) < 0
136
≤
FM (1) : FM (2)
>
≤
A1 = FM (2) A2 = FM (1) KK = 1 J = 2
A1 = FM (1) A2 = FM (2) KK = 2 J = 1
DO 79 I = 3, 16
≤
A1 : FM (I)
>
A1 = FM (I) J = I
≥
A1 : A2
<
L = J J = KK, KK = L SIK = A1 A1 = A2 A2 = SIK
79 CONTINUE
116
116
OUT (K,KK,2) :0
≠
IM (KK) = 2
107
=
FM (KK) : FLM
<
IAFL (J,2) = 0 COMPUTE OUT (K,KK,2) IM (KK) = 1 RESET BASE (KK,2) TIME (KK,2)
≥
107
OUT (K,J,2) :0
≠
IM (J) = 2
80
FM (J) : FLM
COMPUTE OUT (K, J, 2) IM (J) ≤ 1 RESET BASE (J, 2) TIME (J, 2) IAFL (J, 2) = 0
CHECK WHETHER CASE A, B, C, D, E, F OR G AND WHETHER WILD POINT OCCURRED UPDATE TOTALS
520
WRITE IN, INC, IM, IMC IMD, LETTER IWILD
KP = KP + 1
KP : 100
≤
18
40

```
ΔJOB COMPRES,303480
ΔASSIGN 5=CR1A,6=CP1A,9=LP1A,7=TY1A,3=MT3A,4=MT2A
ΔFORTRAN LS,GO.
      COMMON IN(16), IW(16)       ,FM(16),IMC ,EPOS(16,2),ENEG(16,2),
     1BASE(16,2),OUT(100,16,2),SI(100,16),IAFL(16,2),EPS(16),AA,IM(16)
     2,TIME(16,2),ITLE(10),IP(16),P(16),SV(16)
      DIMENSION ITOTC(7),ILET(7)
      IMC = 0
      INC = 0
      FML = 10.
      IRUN = 0
      IBLANK = 4H
      IWILD = 4HW
      ILET(1) = 4HE
      ILET(2) = 4HF
      ILET(3) = 4HG
      ILET(4) = 4HB
      ILET(5) = 4HC
      ILET(6) = 4HA
      ILET(7) = 4HD
      REWIND 4
   12 READ (5,9) IM,(IN(I),I=1,5)
    9 FORMAT (A1,A3,19A4)
      IF (IM(1).EQ.1H/) GO TO 14
      WRITE (7,9) IM,(IN(I),I=1,5)
      GO TO 12
   14 PAUSE 0001
      IMDC = 0
      DO 522 I = 1,7
  522 ITOTC(I) = 0
      IMCC= 0
      ILDW = 0
      INCC = 0
      IF (SENSE SWITCH 5) 140,142
  140 WRITE (7,144)
  144 FORMAT (16HFIGURE OF MERIT     )
      READ (7,146) FML
  146 FORMAT (F10.4)
  142 DO 52 J = 1,16
      DO 52 I = 1,100
      DO 52 K = 1,2
   52 OUT (I,J,K) = 0.
      KP= 1
      IRUN = IRUN + 1
      IF (SENSE SWITCH 4) 91,90
   90 READ (5,9) IWILL
      IF (IWILL.NE.1H/) PAUSE 707070
      READ (5,1) EPS
      DO 27 J = 1,16
   27 P(J) = 1.0
    1 FORMAT (8F10.3)
   91 IF ( SENSE SWITCH 1) 99,92
   92 WRITE (7,7)
   7 FORMAT(56HOPTION PLEASE  1=ZOI,  2=FOIJON,   3=FOIDIS
    1    /3HIII ,35X,5HTITLE  )
     READ (7,13) IOP,ITLE
  13 FORMAT (I3,10A4)
  99 IF (SENSE SWITCH 2) 130,131
 131 REWIND (3)
 130 WRITE (9,20) ITLE,IRUN,(I,I=1,16),(EPS(I),I=1,16)
  20 FORMAT (1H1,20X,10A4,4H RUN,I4//6X,16(1X,3HEPS,I2,1X)/6X,16F7.2//
    13X,2HO1,9X,2HO5,9X,2HO9,9X,2H13,19X,2HO1,9X,2HO5,9X,2HO9,9X,2H13)
  18 IF (SENSE SWITCH 2) 530,532
 530 READ(3)(SI(KP,I),I=1,16)
 532 K = KP - 1
     KM = K - 1
     IF (K.EQ.0) GO TO 110
     DO 400 I = 1,16
     SIK = SI(K,I)
     IF (K.EQ.1) L=1 $ GO TO 500
 404 EP3 = EPS(I   ) * 3.0
     IF (ABS(SI(KM ,I)     -SI(K  ,I)).LT.EP3) GO TO398
     IF (ABS(SI(K  ,I)     -SI(KP ,I)).LT.EP3) GO TO 398
     IF (ABS(SI(KM ,I)        -SI(KP,I)).GT.5.0*EPS(I   )) GO TO  398
     IF (IW(I).EQ.1) GO TO 398
     IW(I) = 1
     IM(I) = 3
     IN(I) = 3
     FM(I) = 5.
     TIME(I,1) = TIME(I,1) + 1.0
     TIME(I,2) = TIME(I,2) + 1.0
     GO TO 400
 398 IW(I)=0
     DO 400 J=1,2
     IF (IAFL(I,J))320,322,324
 320 BASE(I,J) = SIK
     TIME(I,J) = 0.
     OUT(K,I,J) = SIK
     IAFL(I,J) = 0
     TOP = 4.0*EPS(I)
     BOT = 0.
     GO TO340
 322 TIME(I,J) =TIME(I,J) + 1.0
     EPOS(I,J) =(SIK+EPS(I   ) - BASE(I,J))/TIME(I,J)
     ENEG(I,J) =(SIK-EPS(I   ) - BASE(I,J))/TIME(I,J)
     IAFL(I,J) = 7
     TOP = 4.0 * EPS(I)
     BOT = 0.
     GO TO340
 324 TIME(I,J) = TIME(I,J)+1.0
     SV(I) = (EPOS(I,2)+ENEG(I,2))*0.5
     TOP = EPOS(I,J) * TIME(I,J) + BASE(I,J)
     BOT = ENEG(I,J) * TIME(I,J) + BASE(I,J)
     IF ((BOT-EPS(I)).GT.SIK) GO TO330
     HOLD = TOP - EPS(I)
     HOLD1 = BOT + EPS(I)
     IF (HOLD - HOLD1) 326,328,328
```

```
326 HOLD1 = HOLD
328 IF (HOLD1.GT.SIK) GO TO334
    HOLD1 = BOT+ EPS(I)
    HOLD = TOP - EPS(I)
    IF (HOLD .GT. HOLD1) GO TO327
    IF (HOLD1.GE.SIK) GO TO340
    GO TO331
327 IF (HOLD.GE.SIK) GO TO336
331 IF ((TOP+EPS(I)).GE.SIK) GO TO333
330 IAFL(I,J) =  0
    OUT(K,I,J) = BASE(I,J)+(EPOS(I,J)+ENEG(I,J))*(TIME(I,J)-1.)*0.5
    BASE(I,J) = SI(K,I)
    OUT(KP,I,J) = SI(K,I)
    TIME(I,J) = 0.
    GO TO340
333 ENEG(I,J) =(SIK - EPS(I   )- BASE(I,J)) / TIME(I,J)
    GO TO340
334 EPOS(I,J) = (SIK + EPS(I   ) - BASE(I,J))/ TIME(I,J)
    GO TO340
336 ENEG(I,J) = (SIK - EPS(I   ) - BASE(I,J))/ TIME(I,J)
    EPOS(I,J) = (SIK + EPS(I   ) - BASE(I,J))/ TIME(I,J)
340 GO TO (342,346),J
342 IN(I) = 0
    IF (IAFL(I,J)) 344,344,350
344 IN(I) = 1
    INC = INC + 1
    GO TO350
346 IM(I) = 0
    IF (IAFL(I,J)) 348,348,349
348 FM(I) = -1.
    IMC = IMC + 1
    IM(I) = 1
    GO TO350
349 FM(I) = (EPOS(I,J)-ENEG(I,J))*TIME(I,J)/(2.0*EPS(I   ))*P(I)
350 CONTINUE
    GO TO 400
500 OUT(KP,I,L) = SI(K,I)
    BASE(I,L) = SI(K,I)
    IAFL(I,L) = 0
    TIME(I,L) = 0.
    IF (L.EQ.1) L=2 $ GO TO 500
400 CONTINUE
    IF (K.LE.2) GO TO 110
    IF(IMC-1) 70,70,82
 82 DO 81 L=1,16
 81 IF (IM(L).EQ.1) IM(L) = 2
    GO TO 80
 70 IF (FM(1)-FM(2)) 72,72,74
 72 A1 = FM(2)
    A2 = FM(1)
    KK = 1
    J = 2
    GO TO 75
 74 A1 = FM(1)
```

```
      A2 = FM(2)
      J = 1
      KK= 2
   75 DO 79 I = 3,16
      IF (A1-FM(I)) 79,79,76
   76 A1 = FM(I)
      J = I
      IF (A1-A2)78,79,79
   78 L= J
      J= KK
      KK= L
      SIK = A1
      A1 = A2
      A2 = SIK
   79 CONTINUE
  116 IF (OUT(K ,KK,2).NE.0.) IM(KK) = 2 $ GO TO 107
      IF (FM(KK).GE.FML) GO TO 107
      OUT(K ,KK,2) = BASE(KK,2)+SV(KK)    *(TIME(KK,2)  -1.0)
      IM(KK)= 1
      BASE(KK,2) = SI(K,KK)
      IAFL(KK,2) = 0
      TIME(KK,2) = 0.
      OUT(KP,KK,2) = SI(K,KK)
  107 IF (OUT(K ,J,2) .NE.0.) IM(J) = 2 $ GO TO 80
      IF (FM(J).GE.FML) GO TO 80
      OUT(K ,J ,2) = BASE(J ,2)+SV(J )    *(TIME(J ,2)  -1.0)
      IM(J) = 1
      BASE(J ,2) = SI(K,J )
      IAFL(J,2) = 0
      TIME(J ,2) = 0.
      OUT(KP,J ,2) = SI(K,J)
   80 IF (SENSE SWITCH 3) 137,136
  137 DO 138 I = 1,16
      IP(I) = IP(I) + 1
      IF (IM(I).EQ.0) GO TO 138
      IF (IP(I).EQ.1) GO TO 138
      SIK = IP(I)
      P(I) = P(I) + SIK/EPS(17)
      IP(I) = 0
  138 CONTINUE
  136 IMD = 0
      ITOT = 0
      ITWO = 0
      IWLD = 1
      DO 512 I = 1,16
      IF (IM(I) ) 512,512,502
  502 IF (IM(I).EQ.3) GO TO 511
      IF (IM(I).EQ.2) ITWO = 1
      IF (IM(I) .EQ.1) IMD = IMD + 1
      ITOT = ITOT + IM(I)
      GO TO 512
  511 IWLD = 2
  512 CONTINUE
      IF (ITOT.NE.2) GO TO 514
      IF (ITWO.EQ.C) ITOT = 5
  514 IF (ITOT.GT.6) ITOT = 6
      ITOT = ITOT + 1
      ITOTC(ITOT) = ITOTC(ITOT) + 1
      LETTER = ILET(ITOT)
      GO TO (516,518),IWLD
  516 IPRW = IBLANK
      GO TO 520
  518 IPRW = IWILD
      ILDW = ILDW + 1
  520 WRITE (9,4) K ,IN,INC,IM,IMC,IMD,LETTER,IPRW
    4 FORMAT (I4,2(4I2,3X,4I2,3X),I3,7X,2(4I2,3X,4I2,3X),I3,I4,2X,A4,A4)
      IMCC = IMCC + IMC
      INCC = INCC + INC
      IMDC = IMDC + IMD
      INC = 0
  103 IMC = 0
  11C KP= KP+ 1
      IF (KP- 10C)18,18,40
   40 PAUSE 0002
      WRITE (9,598) INCC, IMCC ,IMDC,ILDW,(ITOTC(I),ILET(I),I = 1,7)
  598 FORMAT (/48X,I3,51X,I3,I4,I7,3(/20X,I3,2X,A2,20X,I3,2X,A2,20X,I3,2
     1 X,A2) )
   41 IF (SENSE SWITCH 1)14,42
   42 IF (SENSE SWITCH 2)36,44
   44, IF(SENSE SWITCH 3)50,96
   96 IF (SENSE SWITCH 5) 98,22
   22 IF (SENSE SWITCH 6) 26,132
  132 IF (SENSE SWITCH 4) 133,41
  133 CONTINUE
      CALL PLOT (IOP)
      GO TO 41
   26 DO 28 I = 1,16
   28 WRITE (4) IRUN,I,EPS(I), (SI(K,I),OUT(K,I,1),OUT(K,I,2),K = 1,100)
      GO TO 41
   36 WRITE (7,2)
    2 FORMAT (37H PLEASE MAKE CHANGES TO EPSILON ARRAY /22HIIXXXXX.XXXXY
     1YYYY.YYYY )
   38 READ (7,3) I,SIK,A1
    3 FORMAT (I2,2F1C.4)
      IF ( I ) 41,41,39
   39 IF (SIK .GT.0.0) EPS(I) = SIK
      IF (A1.GT.0.0) P(I) = A1
      GO TO 38
   98 I = 1
      J = 4
   93 WRITE (9,24) ITLE,IRUN
   24 FORMAT (1H1,20X,10A4,4H RUN,I4/)
      WRITE (9,95) (K,K=I,J),(EPS(K),K=I,J),(P(K),K=I,J)
   95 FORMAT (     4( 7X,6HSENSOR,I4,10X)/4X,4(5X,5HEPS =,F10.2, 7X)/4X,
     14(10X,3HP =,F7.2,7X)/4X,4(3X,3HRAW,6X,3HCON,6X,3HMOD, 3X))
      WRITE (9,97) (K,(SI(K,L),OUT(K,L,1),OUT(K,L,2),L=I,J),K=1,100)
   97 FORMAT (I4,12F9.3)
      I = I + 4
      J = J + 4
      IF (I-16) 93,93,41
   50 K = 1
      WRITE (9,24) ITLE,IRUN
      WRITE (9,5) ((SI(I,J),J=1,16),I=1,100)
      WRITE (9,24) ITLE,IRUN
      WRITE (9,5) ((OUT(I,J,K),J=1,16),I=1,100)
      K = 2
      WRITE (9,24) ITLE,IRUN
      WRITE (9,5) ((OUT(I,J,K),J=1,16),I=1,100)
    5 FORMAT (16F7.2)
      GO TO 41
      END
       SUBROUTINE PLOT  (IOP)
      COMMON IN(16), IW(16)      ,FM(16),IMC ,EPOS(16,2),ENEG(16,2),
     1BASE(16,2),OUT(100,16,2),SI(100,16),IAFL(16,2),EPS(16),AA,IM(16)
     2,TIME(16,2),ITLE(10),IP(16),P(16),SV(16)
      DIMENSION SENSOR(40),ITAB1(100),ITAB2(100),ITABLE(600),ISENSOR(40)
       NAME LIST ISEN
   40 WRITE(7,155)
  155 FORMAT( 23HTYPE SENSOR NUMBER ISEN)
       INPUT(7)
       XMAX=0.
      XMIN=1000.
       DO 60 K=1,100
      II=ISEN
       IF(SI(K,II).GT.XMIN) GO TO 55
       XMIN=SI(K,II)
   55 CONTINUE
       IF(SI(K,II).LT.XMAX) GO TO 60
       IF(SI(K,II).GT. 900) GO TO 60
       XMAX=SI(K,II)
   60 CONTINUE
      XMAX=(XMAX+XMIN)*.5
       CALL DSCALE(-10.,118.,-10.,118.)
       CALL DSETUP(ITABLE,600,0,0)
       CALL DCHAR(ITABLE,0.,0.,73S,1H )
   73 CALL DLINE(ITABLE,100.,0.,74S)
   74 CALL DLINE(ITABLE,100.,100.,75S)
   75 CALL DLINE(ITABLE,0.,100.,76S)
   76 CALL DLINE(ITABLE,0.,0.,77S)
   77 CALL DCHAR(ITABLE,0.,50.,78S,1H+)
   78 CALL DCHAR(ITABLE,100.,50.,79S,1H+)
   79 CALL DCONTROL(ITABLE,1)
      ENCODE(160,90S,SENSOR)II
  90S FORMAT(6HSENSOR,F5.0 ,1H*)
      CALL DSETUP(ISENSOR,40,1,0)
      CALL DTEXT(ISENSOR,50., 10.,601S,SENSOR)
  601 CALL DCONTROL(ISENSOR,1)
      CALL DISP(1)
      XA=0.
      DO 70 K=1,100
      XA=XA+1
      CALL DCHAR(ITABLE,XA,    SI(K,II)*50./XMAX,70S,1H.)
```

```
 70 CONTINUE
    CALL DCHAR (ITABLE,105.,40.,610S,1H )
610 CALL DLINE (ITABLE,105.,40.+EPS(II)*50./XMAX,611S)
611 CONTINUE
    CALL DSETUP(ITAB1,100,0,0)
    CALL DSETUP(ITAB2,100,0,0)
    XA=1.
    IF (IOP.EQ.1) GO TO 110
    XA = XA + 1.
    K=2
    CALL DCHAR(ITAB1 ,XA,   +SI(K,II)*50./XMAX,71S,1HC)
 71 DO 72 K=3,100
    XA=XA+1.
    IF(OUT(K,II,1).LT.1) GO TO 72
    CALL DLINE(ITAB1 ,XA, OUT(K,II,1)*50./XMAX,72S)
 72 CONTINUE
    XA=2.
    K=2
    CALL DCHAR(ITAB2 ,XA,    SI(K,II)*50./XMAX,83S,1HM)
 83 DO 84 K=3,100
    XA=XA+1
    IF(OUT(K,II,2).LT.1) GO TO 84
    CALL DLINE( ITAB2 ,XA, OUT(K,II,2)*50./XMAX,84S)
 84 CONTINUE
    PAUSE 3
    GO TO 104
110 XAS = 1
111 DO 112 K = 2,100
    XA = XA + 1.
    IF (OUT(K,II,1) .LT.1.) GO TO 112
    OUU = OUT(K,II,1)*50./XMAX
    CALL DCHAR (ITAB1,XAS,OUU,111S,1H.)
    CALL DLINE(ITAB1,XA,OUU,111S)
    XAS = XA
112 CONTINUE
    XAS = 1.
    XA = 1.
113 DO 114 K = 2,100
    XA = XA + 1.
    IF (OUT(K,II,2).LT.1.) GO TO 114
    OUU = OUT(K,II,2) *50./XMAX
    CALL DCHAR (ITAB2,XAS,OUU,113S,1H.)
    CALL DLINE (ITAB2,XA,OUU,113S)
    XAS = XA
114 CONTINUE
104 IF (SENSE SWITCH 2) 100,101
100 CALL DCONTROL (ITAB1,1)
101 IF (SENSE SWITCH 3) 102,103
102 CALL DCONTROL (ITAB2,1)
103 PAUSE 4
    IF(SENSE SWITCH 6) 200,210
200 WRITE (9,1)
  1 FORMAT (1H1)
    CALL DPLOT (120,50)
 210  CONTINUE
      CALL DCONTROL(ITAB1,0)
      CALL DCONTROL(ITAB2,0)
      IF (SENSE SWITCH 5) 104,105
  105 CALL DISP(0)
       IF(SENSE SWITCH 4) 40,80
   80 RETURN
       END
ΔLOAD XM,FLIB,MAP.
ΔDATA
/
SS1    DO NOT READTITLE
SS1    DO NOT READ TITLE                                RETURN PAUSE 1
SS2    DO NOT REWIND                                    MALE EPSILON CHANGES
SS3    P ARRAY OP.                                      BLOCK OUTPUT
SS4    DO NOT READ NEW EPSILON                          CALL PLOT
SS5    READ IN NEW FM.S                                 WRITE OUT DATA
SS6                                                     WRITE OUT DATA ON TAPE
/
   8.        8.         8.       8.         8.       8.        8.        8.
 200.       150.       150.        250.   150.       8.       65.         8.
/
   8.        8.         8.       8.         8.       8.        8.        8.
  200.        150.     150.      25.        150.     8.        65.        8.
/
   8.        8.         8.       8.         8.       8.        8.        8.
 100.       75.        75.     125.       75.       8.         30.        8.
/
   8.        8.         8.       8.         8.       8.        8.        8.
  50.        35.        35.        65.       35.     8.        15.         8.
/
   8.        8.         8.       8.         8.       8.        8.        8.
  25.       15.        15.      30.         15.       8.        8.        8.
/
   4.        4.         4.       4.         4.       4.        4.        4.
 200.       150.       150.        250.   150.       4.       65.         4.
/
   4.        4.         4.       4.         4.       4.        4.        4.
  200.        150.     150.      25.        150.     4.        65.        4.
/
   4.        4.         4.       4.         4.       4.        4.        4.
 100.       75.        75.     125.       75.       4.         30.        4.
/
   4.        4.         4.       4.         4.       4.        4.        4.
  50.        35.        35.        65.       35.     4.        15.         4.
/
   4.        4.         4.       4.         4.       4.        4.        4.
  25.       15.        15.      30.         15.       4.        8.        4.
/
   2.        2.         2.       2.         2.       2.        2.        2.
 200.       150.       150.        250.   150.       2.       65.        2.
/
   2.        2.         2.       2.         2.       2.        2.        2.
  200.        150.     150.      25.        150.     2.        65.       2.
   2.        2.         2.       2.         2.       2.        2.        2.
 100.       75.        75.     125.       75.       2.         30.       2.
/
   2.        2.         2.       2.         2.       2.        2.        2.
  50.        35.        35.        65.       35.     2.        15.         2.
/
   2.        2.         2.       2.         2.       2.        2.        2.
  25.       15.        15.      30.         15.       2.        8.        2.
```

APPENDIX B
IMPLEMENTATION OF COMMANDS

APPENDIX B
IMPLEMENTATION OF COMMANDS

A representative group of instructions has been implemented with the micrand list (Table 2) developed for the associative computer. The choice of instructions to be microprogrammed was made to illustrate the various program structures necessary in the associative data acquisition system.

All microstep routines use a base address α for the beginning of the sequence: of course, in the loading of microstep memory, these base addresses will all differ, and these beginning addresses (or a part of them) will define the operation code for the instruction. When a given instruction is completed, the micropro-gram will branch to an exit routine ("end") to increment the associative computer sequence counter and request a new instruction. If a specific condition in the sequence is not satisfied, the microprogram will jump to a micro-routine which notifies the GP computer of this situation, possibly through setting certain status indicators.

If a branch address is not a part of the given microstep sequence, it is noted by an "ℓ" followed by a parenthesized expression "XXX" and read as "the location of the micro-routine XXX". Each instruction name in the following sequences is followed by a format which explains the use of the three address fields in that associative instruction. The term "l.b.(X)" is read "the least significant bit address of field X." "No-op" represents the insertion of a "no operation" code in that address field. All of the processing operations allow the results to be written into one of the operand fields except for the multiplication and division instructions; the operands in these operations (multiplier, multiplicand, divisor, and dividend) must be retained for all bit-slice results. Two's comple-ment notation is assumed for all data in the strands. Micrands not filled in are assumed to be "no-ops" which do not cause any action to be taken.

1. FIELD LESS THAN (O1) < (O2) ?

Op Code	l. b. (O1)	l. b. (O2)	no-op

Address	Micrand 1	Micrand 2
α	MSTF	
$\alpha + 1$	MTO1	
$\alpha + 2$	MBSB	
$\alpha + 3$	MGXC	MIBS
$\alpha + 4$	MBSB*	
$\alpha + 5$	MSDF*	
$\alpha + 6$	MBRC*	$\alpha + 12$
$\alpha + 7$	MBO1	MTO2
$\alpha + 8$	MBSB	
$\alpha + 9$	MIEQ	MIBS
$\alpha + 10$	MBO2	
$\alpha + 11$	MBRU	$\alpha + 1$
$\alpha + 12$	MCTF	
$\alpha + 13$	MBRU	ℓ("end")

Initially set TFF and proceed from the least significant bit of each operand. If the corresponding set of bits differs, the exclusive-or gate goes true and the TFF of that strand is set or reset depending on the value of the second operand (Y) bit and the presence of the MIEQ control signal. A "zero" in any TFF designates that O1 < O2, so it must be complemented to represent truth in this algorithm.

2. FIELD GREATER THAN OR EQUAL (O1) $\geq$ (O2) ?

The procedure is the same as for FIELD LESS THAN, except that a "one" in TFF now specifies the strands satisfying the O1 $\geq$ O2 criterion; no complementation is thus necessary for proper representation, and the MCTF step at address $\alpha + 12$ may be deleted.

* The addition of one new micrand to the Table 2 list can save 2 microsteps here, and also in most of the other instructions. This micrand is MBFO (Transfer branch address to MSAR if and only if the current field bit is "1"), and requires an address.

3. FIELD GREATER THAN (O1) < (O2) ?

Op Code	l. b. (O1)	l. b. (O2)	no-op

Address	Micrand 1	Micrand 2
$\alpha - 1$	MRTF	
$\alpha + 1$	MT	
$\alpha + 2$	MBSB	
$\alpha + 3$	MGXC	MIBS
$\alpha + 4$	MBSB	
$\alpha + 5$	MSDF	
$\alpha + 6$	MBRC	ℓ("end")
$\alpha + 7$	MB	MTO2
$\alpha + 8$	MBSB	
$\alpha + 9$	MIEQ	MIBS
$\alpha + 10$	MB	
$\alpha + 11$	MBRU	$\alpha + 1$

Initially reset TFF and proceed from the least significant bit of each operand. If the corresponding set of bits is different, the exclusive -or gate goes true and the TFF is set or reset depending on the value of the O2 bit and the presence of the MIEQ control signal. A "one" in any TFF designates that O1 > O2, and a zero is found for all other cases, i. e., O1 ≤ O2.

4. FIELD LESS THAN OR EQUAL (O1) ≤ (O2) ?

Same as FIELD GREATER THAN, except that the "zeroes" now specify the strands satisfying this criteria. Hence before one proceeds to the "END" routine, the -TFF must be complemented.

5. <u>CENTRAL EQUALITY SEARCH</u> (O1) = (S)?

OP Code	l. b. (O1)	no-op	no-op

Address	Micrand 1	Micrand 2
α	MT O1	
$\alpha + 1$	MBSB	
$\alpha + 2$	MSDF	MIBS
$\alpha + 3$	MBRC	ℓ("end")
$\alpha + 4$	MGXC	MGSX
$\alpha + 5$	MSOP	
$\alpha + 6$	MBRU	α

6. <u>FIELD EQUALITY SEARCH</u> (O1) = (O2)?

OP Code	l. b. (O1)	l. b. (O2)	no-op

Address	Micrand 1	Micrand 2
α	MT O1	
$\alpha + 1$	MBSB	
$\alpha + 2$	MGXC	MIBS
$\alpha + 3$	MB O1	MTO2
$\alpha + 4$	MBSB	
$\alpha + 5$	MIBS	MSDF
$\alpha + 6$	MBRC	ℓ("end")
$\alpha + 7$	MSOP	
$\alpha + 8$	MBRU	α

7. <u>FIELD MOVE</u> (O1) $\rightarrow$ O2

OP Code	l. b. (O1)	l. b. (O2)	no-op

Address	Micrand 1	Micrand 2
α	MTO1	
$\alpha + 1$	MBSB	
$\alpha + 2$	MGXT	MIBS
$\alpha + 3$	MB O1	MTO2
$\alpha + 4$	MBSB	MWTB

$\alpha + 5$	MIBS	MWTB
$\alpha + 6$	MSDF	
$\alpha + 7$	MBRC	ℓ("end")
$\alpha + 8$	MBRU	α

8. MAXIMUM(MINIMUM) SEARCH (O1) Max (min)?

OP Code	s. b. (O1)	no-op	no-op

	Address	Micrand 1	Micrand 2
	α	MTO1	
	$\alpha + 1$	MBSB	
Maximum	$\alpha + 2$	MSCF	
(Minimum	$\alpha + 2$	MRCF)	
	$\alpha + 3$	MSOP	
	$\alpha + 4$	MDBS	MDET
	$\alpha + 5$	MSKC	
	$\alpha + 6$	MGET	
	$\alpha + 7$	MSDF	
	$\alpha + 8$	MBRC	ℓ("end")
	$\alpha + 9$	MBRU	α

9. FIELD CLEAR $0 \rightarrow$ O1

OP Code	1. b (O1)	no-op	no-op

Address	Micrand 1	Micrand 2
α	MTO1	MRTF
$\alpha + 1$	MBSB	MWTB
$\alpha + 2$	MIBS	MWTB
$\alpha + 3$	MSDF	
$\alpha + 4$	MBRC	ℓ("end")
$\alpha + 5$	MBRU	$\alpha + 1$

10. SELECTION

OP Code	no-op	no-op	no-op

Address	Micrand 1	Micrand 2
α	MDET	
$\alpha + 1$	MSKC	
$\alpha + 2$	MBRU	ℓ("no 1's")
$\alpha + 3$	MPRP	
$\alpha + 4$	MPRP	
$\alpha + 5$	MPRP	
$\alpha + 6$	MPRP	
$\alpha + 7$	MPRP	
$\alpha + 8$	MPRP	
$\alpha + 9$	MPRP	
$\alpha + 10$	MCEF	
$\alpha + 11$	MBRU	ℓ("end")

11. FIELD ADD (O1) + (O2) $\rightarrow$ O3

OP Code	l. b. (O1)	l. b. (O2)	l. b. (O3)

Address	Micrand 1	Micrand 2
α	MRCF	
$\alpha + 1$	MTO1	
$\alpha + 2$	MBSB	
$\alpha + 3$	MAD1	MIBS
$\alpha + 4$	MBO1	MTO2
$\alpha + 5$	MBSB	
$\alpha + 6$	MAD2	MIBS
$\alpha + 7$	MBO2	MTO3
$\alpha + 8$	MBSB	MWTB
$\alpha + 9$	MIBS	MWTB
$\alpha + 10$	MBO3	MSDF
$\alpha + 11$	MBRC	ℓ("end")
$\alpha + 12$	MBRU	$\alpha + 1$

12. FIELD SUBTRACT (O1) - (O2) → O3

OP Code	l. b(O1)	l. b. (O2)	l. b. (O3)

Address	Micrand 1	Micrand 2
α	MRCF	
$\alpha + 1$	MTO1	
$\alpha + 2$	MBSB	
$\alpha + 3$	MAD1	MIBS
$\alpha + 4$	MBO1	MTO2
$\alpha + 5$	MBSB	
$\alpha + 6$	MCSF	
$\alpha + 7$	MAD2	MIBS
$\alpha + 8$	MBO2	MTO3
$\alpha + 9$	MBSB	MWTB
$\alpha + 10$	MIBS	MWTB
$\alpha + 11$	MBO3	MSDF
$\alpha + 12$	MBRC	ℓ("end")
$\alpha + 13$	MBRU	$\alpha + 1$

13. FIELD MULTIPLY (O1) x (O2) → O3, O3 + 1

OP Code	l. b (O2)	l. b. (O2)	l. b(O3)

Address	Micrand 1	Micrand 2
α	MTO1	
$\alpha + 1$	MTBS	MTO3
$\alpha + 2$	MTBR	
$\alpha + 3$	MTSB	
$\alpha + 4$	MBO1	MTO2
$\alpha + 5$	MBSB	
$\alpha + 6$	MGXE	MSDF
$\alpha + 7$	MBRC	$\alpha + 40$
$\alpha + 8$	MIBS	
$\alpha + 9$	MBO2	
$\alpha + 10$	MRCF	
$\alpha + 11$	MTO1	
$\alpha + 12$	MBSB	

Address	Micrand 1	Micrand 2
$\alpha + 13$	MAD1	MSDF
$\alpha + 14$	MBRC	$\alpha + 27$
$\alpha + 15$	MIBS	
$\alpha \pm 16$	MBO1	MTO3
$\alpha + 17$	MBSB	
$\alpha + 18$	MAD2	
$\alpha + 19$	MWTB	MBSB
$\alpha + 20$	MWTB	MIBS
$\alpha + 21$	MBSB	
$\alpha + 22$	MSDF	MBO3
$\alpha + 23$	MSKC	
$\alpha + 24$	MBRU	$\alpha + 11$
$\alpha + 25$	MIBS	
$\alpha + 26$	MBRU	$\alpha + 11$
$\alpha + 27$	MTO3	
$\alpha + 28$	MBSB	
$\alpha + 29$	MAD2	
$\alpha + 30$	MWTB	MBSB
$\alpha + 31$	MWTB	MIBS
$\alpha + 32$	MBO3	MBSB
$\alpha + 33$	MSDF	
$\alpha + 34$	MSKC	
$\alpha + 35$	MBRU	$\alpha + 12$
$\alpha + 36$	MTRB	
$\alpha + 37$	MIBS	
$\alpha + 38$	MTBR	MBO3
$\alpha + 39$	MBRU	$\alpha + 3$
$\alpha + 40$	MSCF	
$\alpha + 41$	MTO3	
$\alpha + 42$	MBSB	
$\alpha + 43$	MAD1	
$\alpha + 44$	MBO3	MTO1

Address	Micrand 1	Micrand 2
$\alpha + 45$	MBSB	
$\alpha + 46$	MCSF	MIBS
$\alpha + 47$	MAD2	MBO1
$\alpha + 48$	MTO3	
$\alpha + 49$	MBSB	MWTB
$\alpha + 50$	MIBS	MWTB
$\alpha + 51$	MSDR	MBO3
$\alpha + 52$	MSKC	
$\alpha + 53$	MBRU	$\alpha + 41$
$\alpha + 54$	MBRU	ℓ("end")

14. <u>FIELD DIVIDE</u> (O1), (O2 + ±) ÷ (O2) → O3

OP Code	s. b. (O1)	s. b. (O2)	s. b. (O3)

Address	Micrand 1	Micrand 2
α	MTO1	
$\alpha + 1$	MDBS	
$\alpha + 2$	MBSB	
$\alpha + 3$	MSDF	
$\alpha + 4$	MSKC	
$\alpha + 5$	MBRU	$\alpha + 1$
$\alpha + 6$	MTBS	MTO2
$\alpha + 7$	MDBS	
$\alpha + 8$	MBSB	
$\alpha + 9$	MSDF	
$\alpha + 10$	MSKC	
$\alpha + 11$	MBRU	$\alpha + 7$
$\alpha + 13$	MTO1	
$\alpha + 14$	MBSB	
$\alpha + 15$	MGXC	MDBS
$\alpha + 16$	MBO1	MTO2
$\alpha + 17$	MBSB	
$\alpha + 18$	MSOP	MDBS

Address	Micrand 1	Micrand 2
α + 19	MTO3	MBO2
α + 20	MBSB	MWTB
α + 21	MDBS	MWTB
α + 22	MBO3	
α + 23	MWEB	MCB1
α + 24	MWEB	
α + 25	MGTE	
α + 26	MSCF	MTSB
α + 27	MTRB	MTBQ
α + 28	MBSB	MSDF
α + 29	MCXF	
α + 30	MAD1	MIBS
α + 31	MTBP	MTQB
α + 32	MBSB	
α + 33	MAD2	
α + 34	MBSB	MWTB
α + 35	MIBS	MWTB
α + 36	MBRC	α + 46
α + 37	MTBQ	MTPB
α + 38	MBRU	α + 28
α + 39	MCB1	
α + 40	MGXT	MCEF
α + 41	MAET	
α + 42	MRCF	MTSB
α + 43	MTRB	MTBQ
α + 44	MBSB	MSDF
α + 45	MBRU	α + 30
α + 46	MTO1	
α + 47	MDBS	
α + 48	MBO1	MTO3
α + 49	MDBS	
α + 50	MBSB	

Address	Micrand 1	Micrand 2
$\alpha + 51$	MSDF	
$\alpha + 52$	MSKC	
$\alpha + 53$	MBRU	$\alpha + 59$
$\alpha + 54$	MIBS	
$\alpha + 55$	MSTF	
$\alpha + 56$	MWTB	MBSB
$\alpha + 57$	MWTB	
$\alpha + 58$	MBRU	ℓ("end")
$\alpha + 59$	MBO3	MTSB
$\alpha + 60$	MDBS	
$\alpha + 61$	MTBS	
$\alpha + 62$	MBRU	$\alpha + 13$

APPENDIX C
DATA COMPRESSION PROGRAMS

APPENDIX C
DATA COMPRESSION PROGRAMS

The instruction list for the associative data acquisition system has been organized to allow use of many different data compression algorithms in this system. In the course of this instruction list development, several interpolator algorithms were programmed to check the completeness of the repertoire. Two examples of programmed algorithms are thus presented in this appendix to demonstrate how the associative data acquisition cycle might proceed in compressing the data sent from the system to ground.

The two programmed algorithms, ZOI and FOIDIS, are examined in detail in Reference 7. These programs utilize the instructions developed for the associative data acquisition system. Appendix B describes the various instruction formats, but a few explanations are necessary for notations used in the following programs. The operands are designated in terms of the fields they represent in strand memory; thus "l.b.(F10)" means "the least-significant bit of field 10." Control bit-slices are indicated as "$C\beta$" meaning "control bit-slice β." The strand memory format for each algorithm precedes the program, and the notation conforms to that in Appendix B.

These algorithms do not include any wild-point rejection scheme, but this rejection program would be prefixed to the existing programs; the strand is disabled through the compression cycle if a wild point has been found.

ZERO-ORDER INTERPOLATOR (ZOI)

Strand Memory Format

F0	$\sim$ Steering Tag (Strand Number)
F1	$\sim 2\epsilon$
F2	$\sim y(t_{k+1})$
F3	$\sim l(t_k) - \epsilon$
F4	$\sim l(t_k) + \epsilon$
F5	$\sim u(t_k) - \epsilon$
F6	$\sim u(t_k) + \epsilon$
F7	$\sim$ Transmitted result
F8	$\sim$ Figure of merit

$$\left.\begin{array}{l} C0 \sim \\ C1 \sim \end{array}\right\} \text{Control bit-slices}$$

ZOI Instruction List

ADDRESS	OP. CODE	ADDRESS 1	ADDRESS 2	ADDRESS 3
β	ESS	no-op	no-op	no-op
$\beta + 1$	SDS	l.b.(F2)	no-op	no-op
$\beta + 2$	FSU	l.b.(F2)	l.b.(F1)	l.b.(F3)
$\beta + 3$	FAD	l.b.(F1)	l.b.(F2)	l.b.(F6)
$\beta + 4$	FMV	l.b.(F2)	l.b.(F4)	no-op
$\beta + 5$	FMV	l.b.(F2)	l.b.(F5)	no-op

$\beta + 6$	ESS	no-op	no-op	no-op
$\beta + 7$	TSS	no-op	no-op	no-op
$\beta + 8$	SDS	l.b.(F2)	no-op	no-op
$\beta + 9$	FLT	l.b.(F2)	l.b.(F3)	no-op
$\beta + 10$	TST	l.(C0)	no-op	no-op
$\beta + 11$	TCM	no-op	no-op	no-op
$\beta + 12$	FLT	l.b.(F6)	l.b.(F2)	no-op
$\beta + 13$	GTE	no-op	no-op	no-op
$\beta + 14$	TLD	l.(C0)	no-op	no-op
$\beta + 15$	GTE	no-op	no-op	no-op
$\beta + 16$	GET	no-op	no-op	no-op
$\beta + 17$	TST	l.(C0)	no-op	no-op
$\beta + 18$	TCM	no-op	no-op	no-op
$\beta + 19$	TST	l.(C1)	no-op	no-op
$\beta + 20$	GTE	no-op	no-op	no-op
$\beta + 21$	FGT	l.b.(F5)	l.b.(F2)	no-op
$\beta + 22$	GTE	no-op	no-op	no-op
$\beta + 23$	FMV	l.b.(F2)	l.b.(F5)	no-op
$\beta + 24$	FAD	l.b.(F1)	l.b.(F2)	l.b.(F6)
$\beta + 25$	ECM	no-op	no-op	no-op
$\beta + 26$	TLD	l.(C1)	no-op	no-op
$\beta + 27$	GTE	no-op	no-op	no-op
$\beta + 28$	FLT	l.b.(F4)	l.b.(F2)	no-op
$\beta + 29$	TST	l.(C1)	no-op	no-op
$\beta + 30$	FLT	l.b.(F6)	l.b.(F2)	no-op
$\beta + 31$	TCM	no-op	no-op	no-op
$\beta + 32$	GTE	no-op	no-op	no-op
$\beta + 33$	TLD	l.(C1)	no-op	no-op
$\beta + 34$	GTE	no-op	no-op	no-op
$\beta + 35$	FSU	l.b.(F2)	l.b.(F1)	l.b.(F3)
$\beta + 36$	FMV	l.b.(F2)	l.b.(F4)	no-op
$\beta + 37$	ELD	l.(C0)	no-op	no-op
$\beta + 38$	FAV	l.b.(F4)	l.b.(F5)	l.b.(F7)

$\beta + 39$	GET	no-op	no-op	no-op
$\beta + 40$	SEL	no-op	no-op	no-op
$\beta + 41$	BRC	$\beta + 43$	no-op	no-op
$\beta + 42$	BRU	$\beta + 45$	no-op	no-op
$\beta + 43$	FRE	l.b.(F0)	no-op	no-op
$\beta + 44$	FRE	l.b.(F7)	no-op	no-op
$\beta + 45$	ECM	no-op	no-op	no-op
$\beta + 46$	FAV	l.b.(F4)	l.b.(F5)	l.b.(F7)
$\beta + 47$	FSU	l.b.(F6)	l.b.(F4)	l.b.(F3)
$\beta + 48$	FCL	l.b.(F4)	no-op	no-op
$\beta + 49$	FDV	s.b.(F3)	s.b.(F1)	s.b.(F8)
$\beta + 50$	BRC	$\beta + 57$	no-op	no-op
$\beta + 51$	CMI	s.b.(F8)	no-op	no-op
$\beta + 52$	FRE	l.b.(F0)	no-op	no-op
$\beta + 53$	FRE	l.b.(F7)	no-op	no-op
$\beta + 54$	TCM	no-op	no-op	no-op
$\beta + 55$	GTE	no-op	no-op	no-op
$\beta + 56$	BRU	$\beta + 50$	no-op	no-op
$\beta + 57$	RCB	l.(C0)	no-op	no-op
$\beta + 58$	GET	no-op	no-op	no-op
$\beta + 59$	OXT	no-op	no-op	no-op
$\beta + 60$	ESS	no-op	no-op	no-op
$\beta + 61$	GTE	no-op	no-op	no-op
$\beta + 62$	BRU	$\beta + 2$	no-op	no-op

FIRST-ORDER INTERPOLATOR DISJOINED LINE SEGMENT (FOIDIS)

Strand Memory Format

F0 ~ Steering Tag

F1 ~ $\alpha(t_k)$

F2 ~ $\beta(t_k)$

F3 ~ ϵ

F4 ~ $y^*(to)$ ~ Anchor Point

F5 ~ $y(t_{k+1})$ ~ Sample

F6 ~ $y_1(t_{k+1}) - \epsilon$

F7 ~ $y_1(t_{k+1}) + \epsilon$

F8 ~ $y_u(t_{k+1}) - \epsilon$

F9 ~ $y_u(t_{k+1}) + \epsilon$

F10 ~ i Counter

F11 ~ Intermediate Results

F12 ~ Intermediate Results

$$\left.\begin{array}{l} C0 \sim \\ C1 \sim \\ C2 \sim \\ C3 \sim \\ C4 \sim \\ C5 \sim \\ C6 \sim \end{array}\right\} \text{Control bit-slices}$$

FOIDIS Instruction List

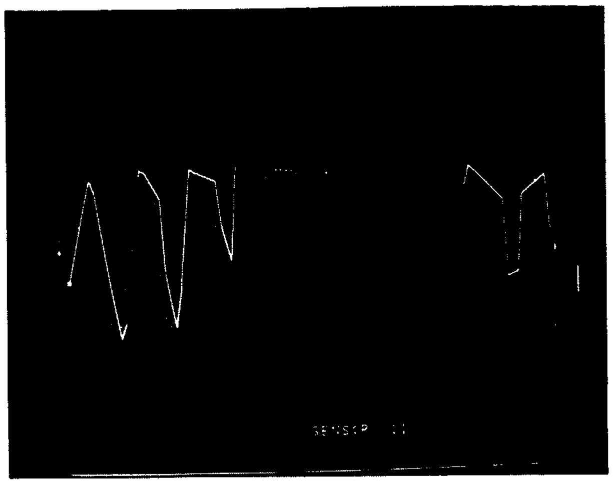
ADDRESS	OP. CODE	ADDRESS 1	ADDRESS 2	ADDRESS 3
β	ESS	no-op	no-op	no-op
$\beta + 1$	FCL	l.b.(F10)	no-op	no-op
$\beta + 2$	SDS	l.b.(F5)	no-op	no-op
$\beta + 3$	FAO	l.b.(F10)	no-op	no-op

$\beta + 4$	FMV	l.b.(F5)	l.b.(F4)	no-op
$\beta + 5$	SDS	l.b.(F5)	no-op	no-op
$\beta + 6$	FAD	l.b.(F3)	l.b.(F5)	l.b.(F1)
$\beta + 7$	FSU	l.b.(F1)	l.b.(F4)	l.b.(F11)
$\beta + 8$	FCL	l.b.(F12)	no-op	no-op
$\beta + 9$	FDV	s.b.(F11)	s.b.(F10)	s.b.(F1)
$\beta + 10$	FSU	l.b.(F11)	l.b.(F3)	l.b.(F11)
$\beta + 11$	FSU	l.b.(F11)	l.b.(F3)	l.b.(F11)
$\beta + 12$	FCL	l.b.(F12)	no-op	no-op
$\beta + 13$	FDV	s.b.(F11)	s.b.(F10)	s.b.(F2)
$\beta + 14$	ESS	no-op	no-op	no-op
$\beta + 15$	FAO	l.b.(F10)	no-op	no-op
$\beta + 16$	SDS	l.b.(F5)	no-op	no-op
$\beta + 17$	FMU	l.b.(F2)	l.b.(F10)	l.b.(F12)
$\beta + 18$	FAD	l.b.(F4)	l.b.(F9)	l.b.(F6)
$\beta + 19$	FSU	l.b.(F6)	l.b.(F3)	l.b.(F6)
$\beta + 20$	FAD	l.b.(F6)	l.b.(F3)	l.b.(F7)
$\beta + 21$	FAD	l.b.(F6)	l.b.(F3)	l.b.(F7)
$\beta + 22$	FMU	l.b.(F1)	l.b.(F10)	l.b.(F12)
$\beta + 23$	FAD	l.b.(F4)	l.b.(F11)	l.b.(F8)
$\beta + 24$	FSU	l.b.(F8)	l.b.(F3)	l.b.(F8)
$\beta + 25$	FAD	l.b.(F8)	l.b.(F3)	l.b.(F9)
$\beta + 26$	FAD	l.b.(F9)	l.b.(F3)	l.b.(F9)
$\beta + 27$	FGT	l.b.(F6)	l.b.(F5)	no-op
$\beta + 28$	TST	l.(C0)	no-op	no-op
$\beta + 29$	ESS	no-op	no-op	no-op
$\beta + 30$	TSS	no-op	no-op	no-op
$\beta + 31$	FLT	l.b.(F9)	l.b.(F5)	no-op
$\beta + 32$	RCB	l.(C0)	no-op	no-op
$\beta + 33$	OXT	no-op	no-op	no-op
$\beta + 34$	TST	l.(C0)	no-op	no-op
$\beta + 35$	GTE	no-op	no-op	no-op
$\beta + 36$	ECM	no-op	no-op	no-op

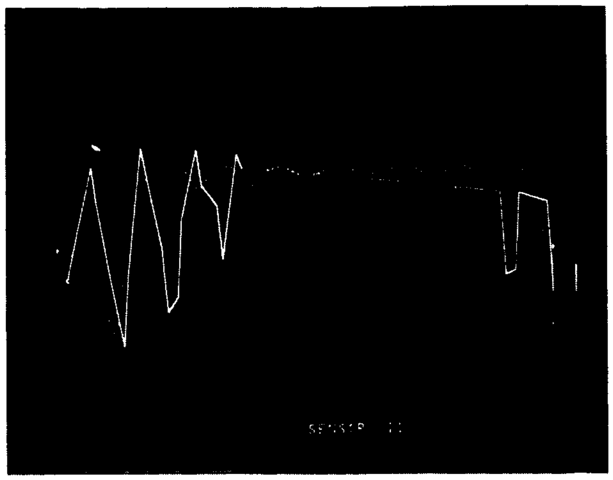
β + 37	GET	no-op	no-op	no-op
β + 38	FIQ	l.b.(F7)	l.b.(F8)	no-op
β + 39	TST	l.(C1)	no-op	no-op
β + 40	GTE	no-op	no-op	no-op
β + 41	FLT	l.b.(F8)	l.b.(F7)	no-op
β + 42	TST	l.(C2)	no-op	no-op
β + 43	TLD	l.(C1)	no-op	no-op
β + 44	GTE	no-op	no-op	no-op
β + 45	FLT	l.b.(F7)	l.b.(F8)	no-op
β + 46	TST	l.(C3)	no-op	no-op
β + 47	GTE	no-op	no-op	no-op
β + 48	FLT	l.b.(F5)	l.b.(F7)	no-op
β + 49	TST	l.(C4)	no-op	no-op
β + 50	ELD	l.(C2)	no-op	no-op
β + 51	GET	no-op	no-op	no-op
β + 52	FLT	l.b.(F5)	l.b.(F8)	no-op
β + 53	RCB	l.(C4)	no-op	no-op
β + 54	ESS	no-op	no-op	no-op
β + 55	OXT	no-op	no-op	no-op
β + 56	TST	l.(C4)	no-op	no-op
β + 57	GTE	no-op	no-op	no-op
β + 58	ECM	no-op	no-op	no-op
β + 59	TLD	l.(C1)	no-op	no-op
β + 60	GTE	no-op	no-op	no-op
β + 61	GET	no-op	no-op	no-op
β + 62	TST	l.(C1)	no-op	no-op
β + 63	TLD	l.(C3)	no-op	no-op
β + 64	GTE	no-op	no-op	no-op
β + 65	FGE	l.b.(F8)	l.b.(F5)	no-op
β + 66	TST	l.(C5)	no-op	no-op
β + 67	GTE	no-op	no-op	no-op
β + 68	ECM	no-op	no-op	no-op
β + 69	TLD	l.(C1)	no-op	no-op
β + 70	GTE	no-op	no-op	no-op

β + 71	GET	no-op	no-op	no-op
β + 72	TST	l.(C1)	no-op	no-op
β + 73	TLD	l.(C2)	no-op	no-op
β + 74	GTE	no-op	no-op	no-op
β + 75	TSS	no-op	no-op	no-op
β + 76	FLT	l.b.(F7)	l.b.(F5)	no-op
β + 77	TST	l.(C6)	no-op	no-op
β + 78	ELD	l.(C1)	no-op	no-op
β + 79	TLD	l.(C3)	no-op	no-op
β + 80	GTE	no-op	no-op	no-op
β + 81	FLT	l.b.(F8)	l.b.(F5)	no-op
β + 82	ESS	no-op	no-op	no-op
β + 83	RCB	l.(C6)	no-op	no-op
β + 84	OXT	no-op	no-op	no-op
β + 85	FGE	l.b.(F9)	l.b.(F5)	no-op
β + 86	RCB	l.(C5)	no-op	no-op
β + 87	OXT	no-op	no-op	no-op
β + 88	GTE	no-op	no-op	no-op
β + 89	FSU	l.b.(F5)	l.b.(F3)	l.b.(F2)
β + 90	FSU	l.b.(F2)	l.b.(F4)	l.b.(F11)
β + 91	FCL	l.b.(F12)	no-op	no-op
β + 92	FDV	s.b.(F11)	s.b.(F8)	s.b.(F2)
β + 93	ESS	no-op	no-op	no-op
β + 94	TLD	l.(C4)	no-op	no-op
β + 95	RCB	l.(C5)	no-op	no-op
β + 96	OXT	no-op	no-op	no-op
β + 97	GTE	no-op	no-op	no-op
β + 98	FAD	l.b.(F3)	l.b.(F5)	l.b.(F1)
β + 99	FSU	l.b.(F1)	l.b.(F4)	l.b.(F11)
β + 100	FCL	l.b.(F12)	no-op	no-op
β + 101	FDV	s.b.(F11)	s.b.(F10)	s.b.(F1)
β + 102	ELD	l.(C0)	no-op	no-op
β + 103	FAV	l.b.(F1)	l.b.(F2)	l.b.(F11)

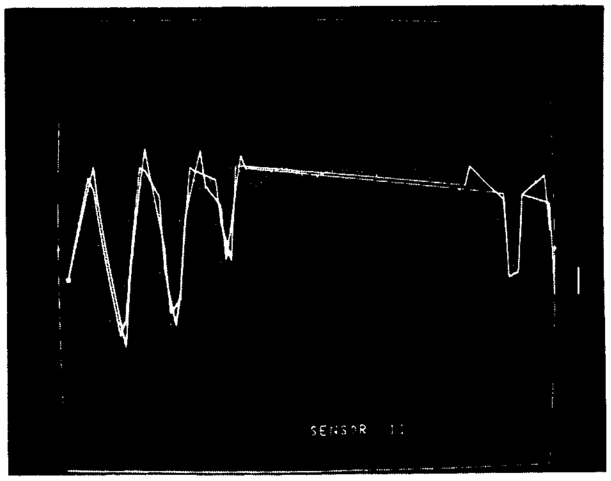
β + 104	GET	no-op	no-op	no-op
β + 105	SEL	no-op	no-op	no-op
β + 106	BRC	β + 107	no-op	no-op
β + 107	BRU	β + 111	no-op	no-op
β + 108	FRE	l.b.(F0)	no-op	no-op
β + 109	FRE	l.b.(F11)	no-op	no-op
β + 110	FRE	l.b.(F10)	no-op	no-op
β + 111	ECM	no-op	no-op	no-op
β + 112	FSU	l.b.(F8)	l.b.(F7)	l.b.(F11)
β + 113	FCL	l.b.(F12)	no-op	no-op
β + 114	FDV	s.b.(F11)	s.b.(F3)	s.b.(F6)
β + 115	FMV	l.b.(F6)	l.b.(F12)	no-op
β + 116	FAV	l.b.(F1)	l.b.(F2)	l.b.(F11)
β + 117	BRC	β + 127	no-op	no-op
β + 118	CMI	s.b.(F12)	no-op	no-op
β + 119	SEL	no-op	no-op	no-op
β + 120	BRC	β + 117	no-op	no-op
β + 121	FRE	l.b.(F0)	no-op	no-op
β + 122	FRE	l.b.(F11)	no-op	no-op
β + 123	FRE	l.b.(F10)	no-op	no-op
β + 124	TCM	no-op	no-op	no-op
β + 125	GTE	no-op	no-op	no-op
β + 126	BRU	β + 119	no-op	no-op
β + 127	RCB	l.(C0)	no-op	no-op
β + 128	ECM	no-op	no-op	no-op
β + 129	GET	no-op	no-op	no-op
β + 130	OXT	no-op	no-op	no-op
β + 131	ESS	no-op	no-op	no-op
β + 132	GTE	no-op	no-op	no-op
β + 133	BRU	β + 5	no-op	no-op



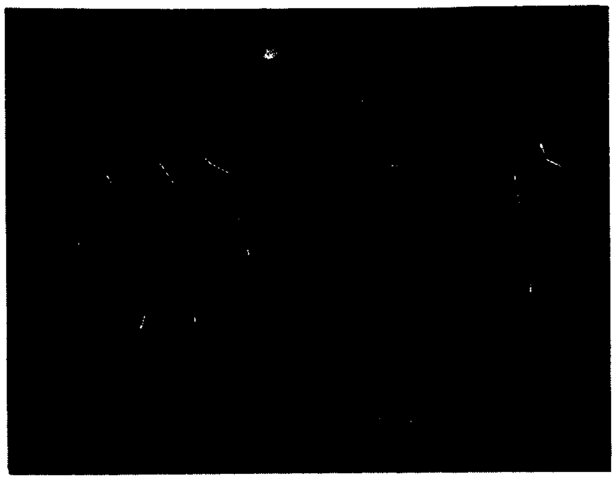
(a) Modified, f_t = .45, I-2



(b) Conventional, I-2



(c) Modified, f_t = .45, I-2, with Conventional, I-2, Superimposed



(d) Conventional, I-2S

Figure 10. Selected Runs for Sensor 11