NASA Technical Memorandum 83510

A Graphics Subsystem Retrofit Design for the Bladed-Disk Data Aquisition System

Ronald R. Carney
Lewis Research Center
Cleveland, Ohio

January 1984

NASA

CHAPTER 1

INTRODUCTION

The National Aeronautics and Space Administration at the Lewis Research Center (LeRC) has developed a data acquisition system that is capable of recording the detailed motion of vibrating blades on bladed-disk assemblies. [1] This data acquisition system, called BDDAS (Bladed-Disk Data Acquisition System) in this thesis, represents a new and unique capability in data collection for gas turbine engines. In the past, data collection on bladed-disk assemblies was accomplished by the use of strain gauges. [2] In order to obtain detailed motion, at least one and if not more, strain gauges were required on every blade of the bladed-disk assembly. This technique presented many problems. Strain gauges are expensive and provide poor output in adverse conditions. Also, the large number of strain gauges needed presently exceeds the capacity of available slip ring assemblies. [3] To overcome these problems the BDDAS was developed. The BDDAS takes blade data from optical probes around the circumference of a casing that surrounds the bladed-disk assembly. Each optical probe acquires its data by bouncing a light beam off the end of a blade. Thus, blade data is acquired without the use of strain gauges, although they can be used to provide additional data it desired.

The BDDAS can record large amounts of data in a very short period of time, approximately 400,000 data points in 70 milliseconds. [3] Since

there is no real-time viewing mode showing this data being collected by the BDDAS, the operator has to guess when the best time would be to start the detailed data collection runs. Therefore, the author was requested to design a graphics subsystem to be added to the BDDAS. The subsystem will provide the operator with real-time viewing of gross blade motions of the bladed-disk assembly.

Along with a real-time mode, LeRC recognized the need for a post-processing visual mode that allows repeated viewing of detailed blade motion on a playback basis, as a supplement to the FFT (Fast Fourier Transform) analyzer frequency output provided by the BDDAS. Therefore, LeRC requested that a post-processing visual mode be provided in addition to the real-time visual mode.

This addition of viewing modes was to be accomplished without substantial modification of the existing BDDAS. And, to keep the cost low, equipment and parts in stock at LeRC were to be used, as far as possible. Thus, the graphics subsystem addition is a retrofit design.

The purpose of this thesis is to present the Graphics Subsystem (GS) and to show that it meets the above LeRC requirements.

CHAPTER 2

INTRODUCTION TO THE BLADED-DISK DATA ACQUISITION SYSTEM

In order to present the GS retrofit design, the reader must be acquainted with the BDDAS. This chapter will be devoted to the BDDAS. The next chapter will present an introduction to the GS retrofit design.

2.1 BDDAS Description

The BDDAS consists of four system components the Spin Rig, the Microcomputer Rack, the Host Computer and the FFT Analyzer. Figure 1 shows how these system components are interfaced. Each system component will be described, along with its interface, in the following sections.

2.2 Spin Rig

The Spin Rig is a test chamber that mounts a bladed-disk assembly in an atmosphere free environment. Figure 2 shows a bladed-disk assembly mounted in the Spin Rig. On top of the spin rig electrodynamic shakers are connected to the bladed-disk assembly in order that impulses may be applied. The applied impulses permit the study of the bladed-disk assemblies impulse response, in addition to any transients that may exist.

In order to get quantitative data about the transient and impulse responses optical probes have been placed in the shroud that surrounds the bladed-disk assembly. The optical probes, one of which is shown in Figure 3, contain an optical transmitter, receiver, and pulse shaping circuitry. The optical probe window, which holds the transmitter and receiver, is placed approximately one-tenth of an inch from the end of a blade. The transmitter transmits a light beam to the end of a blade and the beam is then reflected back to the receiver. The pulse shaping circuitry takes the receiver output and prepares it for transmission to the Microcomputer Rack.

The optical probes in the shroud are configured in groups of three, called a port. The ports are equally spaced about the circumference of the shroud, which is shown in Figure 2 (see optical ports). The present system contains sixteen of these ports and can be expanded to thirty-two. In addition, the ports are oriented such that each port measures three points on a blade as it passes. Probe A of a port measures the leading edge of the blade while Probe B and C measure the middle and bottom (trailing) edges respectively. This is illustrated in Figure 4.

The ninety-six cables from the Spin Rig, when configured with 32 ports, form a parallel interface from which a large amount of data is to be collected. A subsystem must be provided which can do limited processing and storage of the data in real-time. This subsystem is the Microcomputer Rack.

2.3 Microcomputer Rack

The Microcomputer Rack contains microcomputer boards and two additional circuit boards, the TTL and ECL/TTL control boards. It also houses power supplies and two frequency synthesizers.

The microcomputers are arranged in three groups with sixteen boards in each group. The three groups correspond to the three positions in a port. The top group is the A group and records leading edges of the blades while groups B and C record the middle and bottom portions respectively. Each group can be extended to thirty-two microcomputers by placing an identical rack beside the existing one. Each microcomputer in the system is connected directly to an optical probe through the ninety-six cable parallel port of the Spin Rig. This allows each microcomputer to monitor its probe independently of any other microcomputer in the rack.

A major problem of the retrofit design was accommodating the bus architecture of the Microcomputer Rack. Figure 5 shows the details of the bus architecture and how each component of the Microcomputer Rack is interconnected. Each microcomputer is connected to one of three parallel data busses Label A, B, and C.

Although data from the optical probe is collected in parallel, data transmission on the group bus is serial with respect to the individual ports. This is a limiting factor to a real-time data processing mode. [4] In addition to the data busses, each group has a daisy chain bus so that a one byte message can propagate in a round-robin fashion, in

one direction only. [5] The command bus, in which microcomputers receive their instructions, is an 8-bit bus that connects all microcomputers. It is this bus in which the user interfaces the BDDAS, through the Host Computer.

Another bus, that connects all microcomputers, is the ECL bus. The ECL bus provides an 8-bit angle count. This count is in effect a shaft encoder and used to determine blade position. When a blade is detected by an optical probe a register on the microcomputer board is instructed to record the 8-bit blade position off the ECL count bus. The 8-bit word that the microcomputer records represents a localized blade position. That is, a position within the vibration range of the blade. Therefore, in order for this count to have meaning a reference count must be stored. The generation of this count will be referred to later, as the expected blade value.

The frequency synthesizers located in the bottom of the rack are used to provide high resolution shaft encoding to precisely locate blades. Encoding is accomplished by having the Host Computer monitor a once per revolution pulse (1/REV). When a 1/REV is received the host computer determines the rotation speed of the bladed-disk assembly and programs one frequency synthesizer for a frequency which will maintain 78,000 pulses per revolution. The pulses trigger a 16-bit counter on the ECL/TTL board. The lower 8-bits of the counter are the ones sent across the ECL bus, as the blade position mentioned above. When the next 1/REV is received the synthesizer just programmed is put on line while the other is removed for programming. Therefore, only one of the

two synthesizers is on line at a time and they are alternated every other 1/REV. If only one synthesizer were used, counts would be lost during the programming by the Host Computer.

The organization and control of the busses is provided by the TTL and ECL/TTL control boards. The TTL control board interfaces the host computer to the command bus by providing command decoding and signal buffering. In addition, the frequency synthesizers are programmed through the decoding provided by the TTL control board. The ECL/TTL control board has two main functions. First, the data collected by the host computer is collected throught a 16-bit port which is multiplexed to the three data busses A, B, and C. This provides the means for collecting data from each microcomputer. Secondly, the ECL/TTL control board provides the counter for the ECL count bus. The pulses from the synthesizers are sent to the ECL/TTL board and the created count distributed to all microcomputers.

2.4 Host Computer

The Host Computer is an HP-1000 Model 40 minicomputer. It is a general purpose disk-based system used to control the BDDAS.

The HP-1000 has been programmed using Fortran and assembler. The Fortran programs are BLADE and EXER. BLADE is the main program and provides a menu that will lead the operator through a series of steps necessary for a data collection run. This menu is shown in Figure 6. The menu also has prompts for initiating the normalization run, which creates the expected value table for the blades. This table will be explained

further in Chapter 6. The other Fortran program, EXER, is used to troubleshoot the BDDAS. It allows the operator to enter commands directly to the microcomputers through the operator's console or directly from a disk file. The assembler programs are used to program the frequency synthesizers by performing an interrupt service routine that is initiated by the 1/REV from the Spin Rig. The reason for writing this, using an assembler language, is because this routine must execute within one revolution of the bladed disk assembly.

Once a data collection run has finished, the resulting data is stored on a disk. This data is saved for later analysis by the FFT Analyzer.

2.5 FFT Analyzer

The Host Computer up-loads its data to the FFT Analyzer through an RS-232-C serial link, as shown in Figure 1. The FFT Analyzer will convert the data to the frequency domain and provide plots of the transformed data, in addition to a video display.

2.6 Data Acquisition Modules

The microcomputers in the Microcomputer Rack are called DAM's (Data Acquisition Modules). Each DAM is connected to an optical probe in the Spin Rig. The function of the DAM is to convert the optical probe pulse into an 8-bit number representing the position of a blade and transmit this number to the Host Computer. To accomplish this task the DAM has been designed with special functions to ensure reliability with-

out compromising flexibility. Figure 7 shows a block diagram of the DAM and will be discussed below.

The DAM was designed around the Z-80A microprocessor from the Zilog Corporation. This processor was chosen since it could execute the critical timing loop in the shortest period of time, compared to the other processors tested. [6] By using the microcomputer instead of random logic, the DAM could be reprogrammed by simply replacing firmware in a 1K byte EPROM. This means that the critical timing loop could be changed, if needed, to meet the timing of other applications such as new and novel rotor designs.

Figure 7, the DAM block diagram, shows the three types of memory used on the DAM: DRAM (Dynamic Randon Access Memory), SRAM (Static Random Access Memory), and EPROM (Erasable Read Only Memory). The DRAM is used to store data and consists of 4K bytes of memory. The SRAM is used for parameter storage, such as the expected value table, maximum blade deflections, etc., and consists of 256 bytes. The EPROM memory is divided into two blocks. One block consists of 1K bytes on a 2708 EPROM and contains the DAM operational program. The other block consists of 2K bytes on a 2716 EPROM. The 2716 EPROM is not used during operational runs and is free for any addition functions. The 2716 EPROM was intended for diagnostic routines that run a test fixture separate from the Microcomputer Rack.

In order for the DAM to communicate to other devices, three 8225A PPI's (Programmable Parallel Interfaces) have been provided. Each 8225A has two 8-bit ports plus handshaking giving a total of six ports. Port

one permits the Host Computer to collect data stored in DRAM and is called the data port. Port two is used for DAM identification. On the back plane the port is hardwired to a 7-bit identification code which is unique for each DAM. This code is used to identify which DAM is sending the data. Ports three and four, on the second 8255A, are used for the daisy chain bus. Port three connects DAM(X) to DAM(X+1), while port four connects DAM(X) to DAM(X-1).

The third 8255A interfaces the command bus and the ECL count bus. The command bus uses port five plus two handshaking lines as status indicators. The indicators are 1RDY (one ready) and ALRDY (all ready). When a command is sent to a DAM it responds by pulling 1RDY low and ALRDY high. ALRDY indicates that a global command has been accepted by all DAM's. On the other hand, 1RDY indicates that a local command has been accepted by an individual DAM. The other half of the third 8255A is connected to a first-in first-out queue, or FIFO [7], that buffers the ECL count bus. The handshaking for this FIFO is handled automatically by the 8255A.

The FIFO is forty 8-bit words long and receives its input from the ECL count bus through an ECL to TTL translator. The FIFO will only record data when commanded by the Hold Logic. The Hold Logic requires two signals, DAV and RCNT, before the command will be issued. The DAV (Data Valid) signal indicates that the count has stabilized. The RCNT (Record Count) signal indicates that a blade event has occurred. The blade event, being the actual blade pulse or an artificial pulse inserted by the system, is used to maintain data integrity.

The FIFO is an important part to the successful operation of both the BDDAS and the GS retrofit design. Its primary function is to prevent loss of data due to blade excursions. Figure 8 shows a system with 12 blades and four ports. Blades one and two are shown with maximum excursions. It can be seen from this figure that as a blade passes a port, the position in which it is recorded depends on the blade excursion. Excursions of individual blades can be different and therefore, the timing between the recording of blade(X) and blade(X+1) for a port can vary considerably. Since the processor requires a finite amount of time to process data, a second blade could pass the port before the processor could return to collect it. Therefore, the FIFO was added to hold the data until the processor could return to process it. It is expected that the time needed to process the data will be less than the average time between blade detections and will prevent the FIFO from overflowing. [6]

Another mechanism for the prevention of data loss is called data windowing. [3] This is done by four programmable counters that can be directly programmed from the Z-80A. The Arm Delay Counter, Counter D in Figure 7, is used to prevent data collection until the Arm Delay Counter times out. The first blade detected after the counter times out will be defined as the number one blade. This ensures that all DAM's start recording data on the same blade. Otherwise, the first data point recorded would be an unknown, since each blade could be vibrating and its position may or may not be in front of a port at the instant recording is started. By using the Arming Delay Counter, the first blade recorded is known and each succeeding blade recorded is offset by one, unless the

processor is instructed by the data collection program to ignore some subset of blades. In this case the succeeding blades are pulled off the FIFO and discarded.

Once the Arm Delay Counter times out the data windowing counters take over, counters A, B, and C. The objective of these three counters is to ensure that data organization in memory is not lost if for some reason a blade is missed or noise is detected due to debris in the optical path of the probes. The details of how this is accomplished is beyond the scope of this text, since its operation is transparent to the GS retrofit design. The interested reader can find a full explanation in References 3 and 6. It is only important to realize that this mechanism ensures that the data in memory is organized starting with blade one and increments in a predetermined order. Any missed data will have an artificial data point inserted to ensure data integrity.

This chapter described the parts of the BDDAS considered in the GS retrofit design.

CHAPTER 3

INTRODUCTION TO THE GRAPHICS SUBSYSTEM RETROFIT DESIGN

Now that the reader has been introduced to the BDDAS, it is possible to define in general terms the Graphics Subsystem (GS) retrofit design. Figure 9 shows a diagram of the BDDAS interfaced to the Graphics Subsystem. The interfacing links between the two systems are four of the five busses in the Microcomputer Rack. These are the A, B, and C data busses and the command bus. All four busses are extended and connected to the Graphics Pre-processor Computer (GPC) in the GS.

The GS itself contains three major components the GPC, the Graphics Translator, and the Oscilloscope Display. The GPC receives commands through the command bus and responds as if it were a DAM. In addition to commands, background screen data and expected blade position data are sent over the command bus by the Host Computer. The GPC must, also, convert 8-bit raw displacement data, from the A, B, and C data busses, into appropriate 16-bit screen vector words for the Graphics Translator. In the two visual modes, the blade data is sent to the GPC, by the DAM's, over the three data busses. The GPC must coordinate this data, since each bus represents a portion of a blade.

The Graphics Translator is a Hewlett Packard Model 1350A. It has a plug-in module which allows it to communicate to a several different

interfaces by simply exchanging modules. It was purchased with an IEEE-488 (GPIB) module, which was replaced by a 16-bit parallel interface module. The reasons for this change will be discussed further in Chapter 4.

The function of the Graphics Translator is to take digital data representing points on an X-Y plane and draw vectors between the points. This is done by converting discrete data (points) on the 16-bit port to analog data (lines) on three analog ports. The analog ports are called X, Y, and Z which interface to the Oscilloscope Display. The Z port controls the intensity of the display.

The Oscilloscope Display takes the analog data and transforms it to a visual image. Along with blade traces, this image contains background information such as blade numbers and blocks in which the blades can be clearly defined. Figure 10 is a typical display. The lines within each box in this figure are the blade images for a bladed-disk assembly having the maximum number of blades, 64. The blades are shown in a vibrating mode. A normal display will have each blade moving on the screen.

The GS has been designed to run in two modes. The first mode is a Real-Time mode. In this mode the blade images may move or remain stationary and appear sharp or define an envelope of motion, depending on the type of vibrations in the bladed-disk assembly. The other mode, the post-processing mode, will have the blade images move in a continuous fashion. This mode will allow the viewer to clearly observe individual blade motion and relative motions between blades.

The GS retrofit design will not require changes to the operational program in the DAM's 2708 EPROM. This program has a command that permits a jump to the diagnostic EPROM, which is not used when the DAM is in the Microcomputer Rack. By placing the graphics firmware in the diagnostic EPROM and having the Host Computer issue a "jump to diagnostic" command, the graphics firmware can be executed. This approach meets the requirement of minimum hardware modification to the BDDAS. Details of the graphics firmware which includes Real-Time and Post-Processing modes will be described in Chapter 6.

Also, software changes to the Fortran program BLADE will be required. These changes will not be included in this thesis, but will be forthcoming in a separate report after the GS has been implemented.

Other software required, but not provided in this thesis, is the actual microcode for the GPC. The flowchart for this microcode will be discussed and presented in Chapter 6.

To conclude, this chapter presented the overall GS retrofit design. The details of this design will be provided in the remaining chapters of this thesis.

CHAPTER 4

RETROFIT DESIGN SPECIFICATIONS AND EQUIPMENT EVALUTATION

The object of the GS is to provide an interface to the operator through a display on an oscilloscope type device. This interface ultimately determined the specifications of the design, in terms of data throughput rate and equipment configuration. LeRC supplied equipment, which was to be used if possible, was evaluated to determine if the requirements could be met. Two experiments were performed on the equipment, as outlined below.

The experiments were designed to determine the quality of the display devices and to observe visual effects at two different data throughput rates. A low data throughput rate was used to test the phosphor persistence on the display device. The high data throughput rate verified the translation ability of the Graphics Translator by creating a display having a maximum number of moving components. The high data rate display was used also to determine the effect of screen size on the quality of the display, since two different screen sizes were available in the LeRC supplied equipment. The experiments were conducted by using an HP-9845B desktop computer which simulated, at a slower speed, the operation of the GPC.

As as result of the experiments, a data throughput rate was established and recommendations were made to upgrade to provide a

workable interface to the Graphic Pre-processor Computer and to the operator.

4.1 Specifications

The requirements specified by LeRC were that there be no screen flicker and that continuous blade motion be represented on the screen. Screen flicker is related to the "frequency of illumination" of the entire display. It is well known in cinematography [8], that if the screen is illuminated at rates less than 48 times per second, the eye will see the display "flicker". This flicker is very annoying and can even cause nausea (from the author's own experience). The flicker problem is handled automatically by the Graphics Translator and is transparent to the GS. The second requirement, continuous motion, is created by controlling the number of frames (complete display) used per second to update the screen images. To created continuous motion with drawn images a minimum of 18 fps are required. [8] Otherwise, the motion will appear "jerky". The motion picture industry uses 24 fps, because of the sound track, to update the screen and the television industry has chosen 30 fps, which is easily synchronized to the power lines. [9] Therefore, 30 fps, in accordance with the television industry, has been chosen as the screen update rate for the GS.

Taking the 30 fps as a reference the minimum data throughput rates can be determined by considering the maximum number of blades, 64, that are to be displayed. A typical display is shown in Figure 10, of Chapter 3. Each blade in the display requires three vectors to draw the blade. Therefore 192 vectors per frame or a total of 5760 vectors per

second are needed to update the display. This is the minimum data throughput rate in terms of vectors, which will be translated to word transfers, between the GPC and the Graphics Translator discussed in the next section.

Continuous motion can be created at 30 frames per second providing the motion from frame to frame changes in minor increments. [10] The human brain does the rest of the work by interpolating, or filling in, the motion between the discrete values drawn. [11] At the higher framing rates the extra data is wasted because the eye has already done the work. A lower limit is set for the rates at which a post-processing playback can occur. In other words, if each blade is updated less than 30 fps "jerky" motion will occur and interpolation between data points will be required to eliminate the jerky motions.

4.2 Equipment Evaluation

The vector throughput rate has to be converted to a digital throughput rate. This is done by considering the amount of digital data the Graphics Translator requires to produce one analog vector.

The Graphics Translator is an HP-1350A which is supplied with an IEEE Std 488-1978 digital interface. [12] Using this interface, a minimum of 48 ASCII characters (bytes) are required to draw one vector, not including bus protocol characters. This requires a staggering transfer rate of 265,200 characters per second. The interface can pass only 250,000 characters per second which is too slow for 30 fps display update rate.

To avoid this bottleneck the author recommended that the IEEE Std 488-1978 digital interface be replaced by HP's parallel interface which uses 16-bit words. This would reduce the number of transfers (or words) required to update the display from 265,000 8-bit words down to 11,520 16-bit words per second. The reason for the big reduction is because the 16-bit interface requires only two 16-bit words to draw a vector.

Another advantage of the 16-bit parallel interface is the reduction in code for the GPC since it does not need to convert raw binary data, from the BDDAS, to a decimal ASCII format. Also, by leaving the data in binary form the buffer memory in the GPC will be reduced, because the long ASCII vector format will be replaced by the shorter binary format (two 16-bit words).

In addition to the rate at which vectors are drawn on the display, one other aspect of the display must be considered. This is the phosphor persistence. If the phosphor persistence is too long and high screen velocity is encountered, streaking will be caused on the screen. Streaking is due to one image being drawn and persisting long after the driving voltage has been removed. This causes a series of images to be seen at the same time giving the effect of a streak. To reduce the undesirable streaking the phosphor persistence must be reduced. How the HP 1350A phosphor would behave under the framing rates considered was unknown and could only be determined by experimenting in a qualitative fashion on the LeRC equipment supplied.

4.3 Throughput Experiment

The throughput experiment was designed to exercise both the Graphics Translator and the 16-bit parallel interface. Since design parameters required verification before the 16-bit interface would arrive the IEEE Std 488 interface was used.

Figure 11 shows the equipment setup used in the experiment and the interfaces between them. The 9845B desktop computer is connected to the 1350A Graphics Translator through the IEEE Std 488 interface (GPIB). The Graphics Translator takes the digital data off the interface and converts it to analog data representing vectors. The analog data is sent across three channels Label X, Y, and Z to the Oscilloscope Display. The X channel controls the horizontal position of the CRT beam; the Y channel controls the vertical position of the beam. The Z channel modulates the beam, turning it on and off, so that vectors can be drawn without the interconnecting trace being seen.

The display shown in Figure 10, of Chapter 3, contains the maximum number of blades to be displayed at one time. Therefore, creating this display exercises the Graphics Translator at a high data throughput rate. The display consists of background information shown as 64 squares isolating each blade for easy viewing. Within each square two vectors are drawn connecting three points. The third vector used for each blade is a blanked vector and connects the last point in a square to the first point in the next square.

The experiment began by calculating frames picturing a sinusoidal motion for each blade plus a phase angle offset that increased as the blade number increased. Each additional frame increased the phase angle. This process continued until all of the available memory of the 9845B was filled. At that time a high speed DMA data transfer was performed and repeated until interrupted by the operator. Thus causing the illusion of sinusoidal motion.

Motion created on the screen moved slowly due to small increments in phase angle and the slow data transfer rate of the GPIB. In addition, some jerkiness was present and initially it was thought to be flicker. Further analysis showed that the phenomenon was due to the delay needed to repeat the DMA transfer and, thus, was a software problem which could not be avoided. Otherwise, the display behaved as expected.

The experiment verified several things. First, it proved that the Graphics Translator could handle data transfer rates at close to 250K words per second, although this was across the GPIB. The 16-bit interface still transfers at the same rate, but requires less transfers per vector. What remains is the testing of the 16-bit interface to ensure that it works. This will be done in the very near future. Second, it showed that continuous motion can be created provided data transfer rates meet the 30 fps display update rate.

4.4 Display Experiment

The purpose of the display experiment was to evaluate two LeRC supplied oscilloscope type devices for quality of display image and phos-

phor persistence. The two devices are an HP-1338A tri-color display and an HP-1310A oscilloscope display.

The tri-color display uses a beam penetration tube which can provide three colors depending on how far the election beam penetrates the phosphor. This device can provide background information in one color and moving information in another, which provides easy eye references. The disadvantages of the display is size. The viewing area is 9.6 cm by 11.9 cm. This allows only 1.8 cm^2 per blade, using a 64 bladed display, which is not sufficient for detailed viewing. This was verified by performing a test using the screen in Figure 10.

The HP 1310A oscilloscope display, on the other hand, has a much larger display area, 48.3 cm by 27.9 cm. This area will permit 21 cm^2 per blade for a 64 bladed display (much more than the tri-color display). By using the display in Figure 10 it was apparent that detailed motion would be seen easily.

Another test performed in the display experiment was the phosphor persistence test. In order to test the phosphor a display was created, shown in Figure 12. The lattice that disappears into the focal point is used as a reference for the eye. This background information was setup at the beginning of the test and is stored in the Graphics Translator memory. Once the test is started the cube will fall into the focal point. The motion is created by overlaying frames in the Graphics Translator's memory. This screen was chosen because it requires a low data throughput rate per frame. To draw one cube required only nine vectors.

The test was set up to change the number of frames between the cube starting point and the focal point. Since the 9845B performs DMA for data transfer, the only way of controlling the number of frames was to control the handshaking of the GPIB bus. Therefore, a circuit was built to perform this task. Figure 13 shows the equipment layout with the circuit attached to the GPIB cable.

By changing the number of frames, different rates of motion for the cube could be implemented. It was found that at 30 fps and higher, causing high rates of motion, the persistence of the phosphor for both displays lasted long enough to cause streaking of the image. To correct this problem a phosphor with a lower persistence should be used, such as, the phosphor on a measurement oscilloscope.

A recommendation arising from this experiment is to use the HP 1310A oscilloscope display only if another suitable device cannot be found to replace it. The HP-1338A tri-color display is rejected on both counts of display size and phosphor persistence.

The rejection of these devices, based on the phosphor persistence, was of a qualitative nature, that is, by observation. A measurement oscilloscope was tried as a display replacement. Streaking did not occur, but screen size was still a problem.

This chapter discussed two basic experiments that exercised LeRC supplied equipment, according to specifications that were derived beforehand.

CHAPTER 5

MODIFIED DAM PROCESSING FOR GS MODES

A major element in the design of the two data processing modes of the GS was the interface with the existing bus architecture of the BDDAS. All the data to the GS mus be sent over three data busses, each of which connects to one third of the DAM's. A bus protocol must be setup to eliminate bus contention between DAM's [13], allowing only one processor to access the bus at any one time. The bandwidth of the data busses is such that it is impossible to transmit all recorded data across them. This requires that algorithms select data for transmission to reduce it to no more than the bus bandwidth or the maximum throughput rate of the GS.

In the three sections that follow, the bus protocol and the two data processing algorithms will be explained. The algorithms are the Real-Time and Post-Processing algorithms, which are used exclusively for data transfer to the GPC.

5.1 Bus Protocol

Four techniques were considered for the bus protocol. [5, 7, 13] These are as follows:

1. Random access

2. Microcomputer interrupts

3. Program phasing

4. Daisy chain bus handshaking

The first method considered was a random access to the data bus. When a DAM has acquired a data point it would take control of the bus and transmit the data along with its address and which blade the data was for. This method must also involve a protocol that assures exclusive control of the bus that effectively "locks out" all other data transfer requests from other DAMs.

A primary disadvantage of the random access method is the amount of information that is required for each data transfer. For each data point transmitted, additional information was required to identify the data. This slows the bus considerably. A second disadvantage is that if the DAMs are randomly contending for the bus, there is no centralized control over blade sampling. In addition, the problem of handling "locked out" transfer requests remains. Thus, this method was dropped for a better approach.

The second method considered was to poll each DAM by using hardware interrupts. By using this technique data points could be collected at any time and in any order. This appears to be a big advantage. But, when the DAM was investigated it was found that these interrupts were not available without DAM Modification. Since one of LeRC requirements was to minimize modification to the BDDAS this technique was not pursued any further.

The third method, program phasing, was considered seriouly. It consisted of having each DAM start the execution of its program, which is identical on all DAM's, by a predetermined delay. The amount of delay would be such that each DAM would come to its output routine just after the previous DAM, transmitted its data. In this manner, the maximum bus bandwidth could be utilized. Using this protocol demands that the data processing algorithm have no loops with undetermined lengths. When the DAM reads the FIFO the data had better be there, otherwise data order would not remain organized with known values. The dynamics of the blades was such that data in the FIFO could not be ensured. Therefore, program phasing was abandoned.

The last method, Daisy Chain Bus handshaking, is the method that was adopted. When the DAM finished using the bus it flags the following DAM by using one of the control lines in the Daisy Chain bus. At this point the following DAM takes control of the bus and transmits its data. This process continues in a round robin fashion. The fixed order of DAM access to the bus is utilized in GS processing.

The only problem foreseen is the designation of a number one DAM, the one that transmits first. This can be easily handled in the firmware algorithms by setting a software switch, before a data run, indicating the number one DAM.

Based on the daisy chain data bus protocol, DAM algorithms for data selection to meet the bus bandwidth were developed. Two separate algorithms will be needed, for the Real-Time mode and for the

Post-Processing mode. The algorithm for the Real-Time mode must select a "viewing" subset of data to forward over the three data busses. This subset of data will be selected so that gross blade characteristics can be recognized, such as, maximum deflections and standing waves. [14, 15, 16] The operator will use this to determine when to record detailed data.

Once the detailed data has been recorded and sorted, by the Host Computer, a Post-Processing mode will be used to analyze the detailed data. This will be done by reloading the sorted detailed data back into the DAM's. The DAM's will use the Post-Processing algorithm to output, in an orderly fashion, blade data to the Graphics Pre-processor Computer.

The following two sections will describe the algorithms used by the DAMs to systematically transmit data to the GPC. The first algorithm described will be for the Real-Time mode and the second one for the Post-Processing mode.

5.2 Real-Time Algorithm

Given a bladed-disk assembly, with 64 or less blades, many possible "viewing modes" can be defined, where a viewing mode is determined by the subset of blades being displayed on the screen. LeRC has requested that the operator be able to define a viewing mode by inputting parameters into the Host Computer's terminal. The Host Computer will use these parameters to create a parameter array that will be down loaded to the DAMs for use in the Real-Time algorithm. Once the operator has

determined a viewing mode, the Real-Time algorithm will use the parameter array to select a data subset that will update the screen.

The DAM's collect all incoming data. All of this data could not possibly be sent to the GPC, due to the bus limitations discussed in Chapter 2. Therefore, the Real-Time algorithm will instruct the DAM's to discard all undesired data to create a data subset. The amount of data to be discarded was determined by the experiments of Chapter 4, which concluded that 30 fps would be a sufficient update rate to provide continuous motion and prevent flicker.

A problem that remains is how a data subset is to be selected for Real-Time viewing. There are several possible ways data could be used: from only one DAM, from a subset of DAM's, or from all the DAM's. If only one DAM is used to create a data subset, fundamental sampling requirements would not be met to detect the high frequency components of blade motion. The resulting displayed image would appear as if it were strobed once per revolution giving a highly aliased image. Effects of such aliasing problems were explored in optical blade flutter experiments done at LeRC. [17] Using the BDDAS all blades could be tracked instead of one (as in the optical blade flutter experiments). If a subset of DAM's were always used, it would be hard to detect synchronous [18] blade vibrations. That is, every time a blade passes a DAM it would be positioned at the same place and no vibrations would be detected. In order to avoid this, as many DAM's as possible should be used to sample the blade motions. Therefore, the Real-Time algorithm must permit all DAM's to participate in the sampling process.

One thing that will not be detected is a transient response that only lasts for several revolutions of the bladed-disk assembly. But, this can be recorded and analyzed in the Post-Processing mode.

There is no unique way of using DAMs to create a data subset, in fact many possible schemes could be devised using the parameter arrays sent by the Host Computer. For algorithm simplification, the technique devised permits only one DAM to transmit data during a revolution. Each DAM implements the algorithm using one of several parameter arrays that tell the DAM which blade data to discard and which to save. After all DAM's have had their turn transmitting data to the GPC, the process will repeat until reset by the Host Computer. LeRC concurs that this is a suitable approach for beginning studies. If deficiencies are identified, the algorithm can be changed by simply reprogramming DAM EPROMS.

When a DAM has transmit control, i.e., permission to transmit, on a revolution it will be instructed by the algorithm to discard all blade data except for a predetermined subset of blade values, which are transmitted to the GPC. The blade values to be transmitted are not stored in RAM, but are transmitted on the fly. In other words, data must be processed before new data is received, which gives the algorithm its real-time nature.

Once a DAM's "transmit" revolution has been completed, the DAM passes bus control to the next DAM on the Daisy Chain bus. This DAM will transmit a different subset of blades. Again, control will be passed

and different subsets sent until all blades of the bladed-disk assembly have been updated. At this point, the DAM's will repeat the subsets in the same order. As an example, consider a "two" blade bladed-disk assembly. DAM one will transmit blade one only. DAM two will transmit blade two only and DAM three will be back to blade one again. This process will continue as the DAM's are encountered within a group. At the other end of the bus, the GPC will see blades one, two, one, two, one, etc.. Therefore, the GPC only needs to store the sequence one, two, to be able to interpret the data.

The above method allows the blade data to be sent in any order by simply changing DAM subsets. The method requires enough memory in the GPC to store the received sequence of blades, which can be as small as the number of blades on the bladed-disk assembly. Absolutely any sequence can be sent.

Blade sampling is pre-programmed by the Host Computer by loading into the DAM the number of blades to be sampled (NB) and an array (SKIP) of length NB+1 giving the number of blades to be skipped before and after each sample is taken. Skipping involves reading, and thus unloading the FIFO.

A flow chart that implements the data selection for the Real-Time mode algorithm is shown in Figure 14. It starts out by initializing one flag and two counters. The flag, which is called PTS (Persmission to Send), is used to designate the number one DAM and to indicate which DAM has transmit control. The two counters are SC (Skip Counter) and SP

(Skip Pointer). SC is used to keep track of how many blades have been ingored between valid blades. SP, on the other hand, keeps track of how many valid data points are to be transmitted and is used as an index to the SKIP array.

The Skip Pointer (SP) indicates when the last valid data point has been transmitted during the current revolution. If SP is equal to zero, PTS is tested to see if the DAM has transmit control of the bus. If so, transmit control is passed to the next DAM. The next step in the loop tests to see if the Host Computer is signalling a data collection run. If the Host Computer did not signal a data collection run SP is reset to the number of blades minus one. SP will never be set to the highest value again. This is because the highest entry in the skip array (SKIP(NB)) is used as a starting skip value. The lowest value in the array (SKIP(0)) holds a value equal to the number of blades to be skipped between the last valid blade sample in the revolution to the first value blade sample in the next revolution. SKIP(NB) represents the number of skips between blade 1 and the first blade to be sampled. Therefore, there is no need for SKIP(NB) except at the beginning of the run, when the arm delay counter starts every DAM at blade 1. Next, PTS is set (F) indicating a non-transmitting revolution and the main loop is re-entered. If SP was not zero, SP will be decremented so that it points to the next group, number of blades, to be skipped.

Next, SC is tested to see if all unwanted blades on this revolution have been skipped, if not a loop is entered where data is pulled off the FIFO and ignored until the desired blade has entered the FIFO.

It is possible that the outer loop, which indicates testing for permission to send, transferring transmit control, testing the number of blades, and transmission of blade data, may exceed the 58 msec. average minimum time between blades entering the FIFO. The innermost loop, however, requires only 15 msec. and is active most of the time, keeping the FIFO essentially clear of data.

If SC is equal to zero the rest of the main loop is entered by testing the FIFO for valid blade data. When the valid blade data arrives PTS is again tested to see if the DAM has permission to transmit. If so, the data is sent to the GPC, otherwise, it is ignored. SC is then loaded with the next group of blades to skip and the main loop re-entered starting the process over.

It is possible that only a few of the DAM's will be required to send data in the selected viewing mode. In nonparticipating DAM's an alternative program is executed, shown in Figure 15, which causes the DAM to simply watch for the data collection signal and for a PTS signal, which it echos to the next processor. This program in effect removes the DAM from the data collection process.

5.3 Post-Processing Algorithm

In order to view short lived phenomenon, such as the transient response from the application of an impulse, the rate at which data is displayed must be slowed down considerably from real-time. This will be accomplished in the Post-Processing mode. This mode will process data,

stored in the Host Computer, to match operator requested playback rates. If the playback rate is so slow that there is not enough data to create 30 frames per second, the Host Computer will create extra frames of data by performing a linear interpolation between data points. On the other hand, if the playback rate requested exceeds the bus bandwidth the Host Computer will have to reduce the amount of data. Thus, during a given playback session the data playback rate can be varied between the lower limit of 30 fps and the upper limit which is the data bus bandwidth or the GS throughout rate (see Chapters 2 and 4).

When the Host Computer finishes sorting the data for a specific data collection run, the data will be in a frame format consisting of a minimum of 2048 frames. Each frame will contain update information for all the blades currently being displayed. The data will be loaded into the DAM's RAM memory by the Host Computer, with frames it has created. DAM number one of each group will be loaded with the first set of frames, DAM number two with the next set and so on down the line. The Host Computer will then instruct the DAM's to output the frames to the GS, in a serial fashion, until all frames have been transmitted. This will in effect create a movie lasting from a few seconds to several minutes depending on the framing rate and the number of blades for which data was collected.

The collected data is uploaded to the Host Computer for processing because of limited ROM memory space on the DAM's. Each DAM has only 2K of ROM space available for both the Real-Time algorithm and the Post-Processing algorithm. This is not much memory space to do post-processing.

Unforeseen changes may be required to the Real-Time and Post-Processing modes. Therefore, to use as little memory space on the DAM's as possible it was decided to upload the data to the Host Computer. So, the Post-Processing algorithm program that resides in the ROM memory of the DAMs executes a minimal algorithm that controls the rate at which frames are sent to the GS.

The Post-Processing algorithm is shown in Figure 16. The algorithm starts by initializing the parameters NF (Number of Frames) and PTS (Permission to Send). If the DAM is the number one DAM its PTS flag is set true (T), otherwise, it is set false (F). If PTS is (F) then a loop is entered until PTS is set (T) by the preceeding DAM. Once PTS is (T) the DAM is then instructed to monitor the command bus for a strobe. This strobe will control the rate at which data is sent to the GPC. If no strobe is present the system will continue to loop until a strobe is detected or the system is reset. If a command bus strobe is detected, data is transmitted and a test is performed to see if all data has been sent. If more data still exists the cycle is repeated. Otherwise, PTS is set (F) to prevent further transmission and the next DAM is then instructed to output its data. The whole process continues until reset by the Host Computer.

This chapter described the bus protocol used to access the data busses by the DAMs. In addition, it described the algorithms for the Real-Time and Post-Processing modes of the GS.

CHAPTER 6

THE GS GRAPHICS PRE-PROCESSOR COMPUTER

A gap must be bridged between the BDDAS and the Graphics Translator (see Chapter 2 through 5). The device that bridges the gap will be called a Graphics Pre-Processing Computer (GPC). This computer must decode the order of data sent by the BDDAS, normalize this data, and add the normalized data to screen vectors for transmission to the Graphics Translator.

Two arrays are sent to the GPC to help it perform the three tasks mentioned above. The arrays are the expected value table and the screen vector table. The expected value table contains an expected blade position for every blade on the bladed-disk assembly that is being displayed. This expected blade position is determined from a normalization run done by the BDDAS. The normalization run is a data collection run with no stimulus applied to the bladed-disk assembly. When the data collection run has finished the DAM memory will be filled with 64 or more blade samples for each blade on the bladed-disk assembly. The values are averaged, for each blade, and an expected value (blade position without stimulus) is created and stored in the expected value table. The screen vector table has six coordinate values for each blade being updated on the screen of the Oscilloscope Display device. These values are the X and Y coordinates of the three points representing the blade.

The input data and the expected values are 8-bit words. The screen vector values are 16-bit words. The screen vector coordinates require the ten least significant bits of the 16-bit word. The other six bits are used for control information, such as vector blanking. Therefore, the GPC must be a 16-bit machine and must support the addition of 8-bit words to 16-bit words.

When the DAMs send data to the GPC they do not send data identification with the data. But, the order in which data is sent is predefined and is cyclic. Therefore, the GPC can decode the data order by having the expected value table and the screen vector table ordered the same as the incoming data.

Figure 17 shows a block diagram of the GPC with its inputs and output. The expected value and vector screen tables are down loaded to the GPC over the command bus. The processor loads the tables in memory for later use. Also sent over the command bus is any background information for the display screen. The processor takes this information and transmits it directly to the Graphics Translator through the output port. The Graphics Translator will hold this data in its memory allowing the GPC to be dedicated to data processing.

Data processing takes place when the Host Computer instructs the GPC to monitor the data input ports. The processor starts by monitoring the A-bus for leading edge blade data. When data is received the processor retrieves the corresponding expected blade value from the expected table. It subtracts the data from the expected blade value giving an offset from the expected or normalized blade position. The processor

then retrieves the cooresponding X-component, adds the offset to it, and sends the result to the Graphics Translator. The Y-component is retrieved next and sent directly to the Graphics Translator, thus defining the first blade point. The processor then goes on to the B-bus and C-bus, transmitting coordinates defining visible vectors representing the blade. The cycle is started over with the A-bus and will continue until reset by the Host Computer.

6.1 Microprocessor versus Bit-Slice Processor

One of the design decisions that had to be made was how to implement the processor of the GPC. Two techniques were considered for the implementation. An acceptable technique would have been to use a one chip 16-bit processor, such as the Motorola 68000. Compared to a special purpose digital system, this approach has an advantage of greatly reducing hardware requirements. The major disadvantage was that no 16-bit development system was available at LeRC to aid the design of program code.

It was decided to implement the GPC as a microprogrammed bit-slice computer. The programs for the processor will be short requiring two I/O operations, two arithmetic operations, and a lot of memory access for operands. With microprogramming the instruction fetch operation can be eliminated, allowing one-third more of the processor's time, on the average, to be devoted to data operations. By using the bit-slice architecture the programming can be imbedded in microcode and, thus, eliminating machine level instruction altogether. Also, the bit-slice architecture allows complete control of processor components so that

operations can take place in parallel gaining even more speed. [19] Therefore, using bit-slice architecture, a very fast processor can be designed that permits extreme flexibility. This flexibility could allow LeRC to do data processing to extract data about characteristics of the bladed-disk assembly that are, as yet, unknown.

LeRC has both bit-slice components and a bit-slice development system in-house. This development system (The Advanced Micro Devices System 29) has the ability to incircuit emulate the microcode ROM, allowing changes of microcode to be made easily during operation of the GPC. [20]

6.2 GPC Design

The GPC will be a modified form of Advanced Micro Devices Super Sixteen computer. [21] The Super Sixteen was selected for modification for several reasons. The main reason is that the Super Sixteen is based on the AMD 2900 family of bit-slice components. Secondly, the design time for the GPC is considerably reduced by making changes to an architecture which is already proven. Last of all, the Super Sixteen being a bit-slice architecture is fast and flexible due to pipelining and microcoding respectively.

Showing in Figure 18 is a block diagram of the Super Sixteen which breaks out the six major sections in which the computer is divided, and shows their architecture. These sections are as follows:

1. The ALU (Arithmetic/Logic Unit)
2. The PCU (Program Control Unit)
3. The Data Path
4. The Microcode Memory
5. The Clock and Memory Control
6. The I/O Interface and Control Bus

The GPC was designed by making major modifications to only two of the sections mentioned above. These sections are the Data Path and the I/O Interface and Control Bus. The other sections will have only minor changes and will be discussed as needed in connection with the major changes used to create the GPC. The blocks that are to be changed are shown in Figure 18 as shaded blocks.

The changes to the Data Path section consists of removing the instruction register (I-REG) and its associated instruction decoder (DECODER). These components were removed since no machine level instruction will be used in the GPC.

The GPC architecture is shown in Figure 19. In order to do more parallel operations, the Data Bus of the Super Sixteen was divided in two. An A-data bus will be interfaced to the DA bus of the ALU through, a new register, the A-Z REG, and the existing Z0 REG. A B-data bus will be interfaced to the DB bus of the ALU through, a new register, the B-Z REG, and the existing Z1 REG. This change permits data to flow from the external data busses, A and B, to the internal data busses of the ALU. In order to provide data flow in the opposite direction, an extra D REG

has been provided so that both data busses can be given access to the ALU Y-bus.

The Super Sixteen has two AM 9551's which are serial interfaces for a console and a peripheral. The GPC has no need for either of these, since all communication with the external world will be in parallel, for speed purposes. Therefore, the AM 9551's will be replaced with two AM 9555's.

The AM 9555 is a high speed version of the popular PPI (Programmable Parallel Interface) from Intel, called the 8255A. The PPI has 25 I/O pins that can be configured in several ways. [22] In the GPC, the PPI will be configured with two 8-bit ports and two handshaking lines for each port. It will operate in a mode in which handshaking will be accomplished automatically. Once data is accepted in the port a condition line will be provided to indicate that data has been accepted. The condition line is connected to a test tree that can flag the microcode indicating data has arrived. When the data is read from the PPI handshaking lines are reset and the process starts over.

With two AM 9555's, four 8-bit ports are provided to interface the BDDAS. Three of the ports are connected to FIFO's (not the DAM FIFO's) that interface to the three data busses of the BDDAS. The AM 9555 ports are shown at the bottom of the GPC detailed block diagram, in Figure 19. The other AM 9555 port is connected directly to the command bus of the BDDAS. The FIFO's which are 40 8-bit words deep, provide buffering for the GPC so that handshaking microcode is not required. The buffering allows data transfer on the GS data busses to be independent of what the

GPC is doing. The command bus port will accept commands from the Host Computer, as well as tables and screen background data as mentioned earlier. The AM 9555's, with minor modifications in interfacing hardware, makes the GPC appear as just three more DAMs connected to the BDDAS.

The Graphics Translator interface is provided by latching the output of the ALU Y-bus directly to the 16-bit input port of the Graphics Translator. The latch, designated the V-REG in Figure 19, replaces the unneeded AM 9519 (secondary interrupt controller) of the Super Sixteen. When the final vector component, X or Y, is formed in the ALU it will be latched into the V-REG and strobed into the Graphics Translator using a one shot multivibrator. This eliminates handshaking and its associated microcode. It should be noted here that if the input data rate exceeds the 500K transfer rate of the Graphics Translator interface, some of the input data can be lost. Care must be exercised not to exceed this rate.

The remaining, unmodified hardware for the Super Sixteen Computer is explained in AMD's "Built A Microcomputer" series, Chapter IX. [21] The author highly recommends the "Build A Microcomputer" tutorial series by Advanced Micro Devices, Inc..

6.3 Microcode Flow Chart

The microcode starts initializing the I/O devices and immediately enters a loop which polls the command bus, similar to the action of the DAM. While in this loop, the GPC will respond to three commands, load

tables in memory, initialize Graphics Translator, and take data. Figure 20 shows the microcode program flow chart for the GPC.

The next three decision blocks test the received command to determine which command it is. If it is the "load memory" command, the following data on the command will be loaded directly into memory. The amount of data to be taken off the command bus is given as part of the command. If the command is the "Graphics Translator" command the following data will be transmitted directly to the Graphics Translator.

The only other command that will be accepted by the GPC is the "take data" command. When this command is received the Z-REG's are loaded with appropriate data from memory and enters an endless loop which polls the ports for data, processes the data, and transmits it to the Graphics Translator. The GPC exits the loop when the GS is reset.

6.4 Microcode Word Format

The microcode word consists of 96-bits and is broken down into 11 fields. The bits are used to control the GPC, and bits that control related functions of the GPC are grouped into a field.

Figure 21 breaks out the microcode word giving the mnemonics assigned to each bit. In the figure asterisks are placed beside the bits that have been changed due to hardware modifications made on the Super Sixteen. The field that experienced the most changes is the Data Path field. Five bits have been changed; these are the ENCTR, INC, LDD, Z1-I, and BRIEN. These bits controlled the flow of data through the Z-REG, the D-REG, and associated latches of the Super Sixteen. The bits in the

microcode MEMORY CONTROL field, REQB and HREQ, will not be used. The bits were designated for bus control of a multiprocessor architecture. Only one processor is used in the GS and these bits can be reassigned to the Data Path field for register control.

The only other bit to be changed is a multiplexed bit, of the CONTROL STROBES field. This bit will be used to latch the output of the ALU into the V-REG of the GPC.

The AM 9555's will be controlled by the same I/O controls as the AM 9551's. When the IOEN bit is pulled low an I/O read or write will be initiated. The read or write strobe and the I/O address will be encoded in the CONTROL BITS field and sent when the IOEN goes low.

This chapter described the final part of the GS retrofit design, the GPC. The GPC is a modified Super Sixteen computer. The major changes of the Super Sixteen were made to the Data Path and I/O Interface sections.

CHAPTER 7

SUMMARY

A Graphics Subsystem has been designed as a retrofit to the Bladed-Disk Data Acquisition System (introduced in Chapter 1 through 3). This design provides two "viewing modes" in which the motion of individual blades, on the bladed-disk assembly, can be seen on an Oscilloscope Display. One of the viewing modes is a Real-Time mode that permits viewing of gross blade characteristics, such as standing waves and blade displacements. The second mode is a Post-Processing mode and allows the operator to view detailed blade motion on a play-back basis. The operator has the ability to control the viewing modes by entering parameters related to the bladed-disk assembly and to the number of blades to be viewed.

Specifications outlined in Chapter 4 for the Graphics Subsystem were derived in terms of data throughput rates and type of screens to be displayed. From these specifications LeRC supplied equipment was exercised to ensure that they will perform at the pre-derived specifications.

Two results from the specifications and experiments indicated further action. First, the IEEE-488 interface, on the Graphics Translator, will be upgraded to a 16-bit parallel interface in order to meet the data throughput of the Graphics Subsystem. Secondly, the experiments indicated that streaking will occur at high screen velocities, on the display

devices, due to long phosphor persistence. To avoid this potential problem it is recommended that a lower phosphor display device be used, such as a measurement oscilloscope, if available in-house. If a suitable device is not available in-house further investigation giving quantitative data on phosphor persistence should be performed.

Two DAM algorithms were developed in Chapter 5 for the viewing modes of the Graphics Subsystem. A Real-Time algorithm is used to select data from the optical probes and transmit the data directly to the Graphics Pre-processor Computer. A second algorithm, the Post-Processing algorithm, transmits data stored in DAM memory, in a frame format, to the Graphics Pre-processor Computer, also. Although, the Post-Processing algorithm is much simpler in concept.

Finally, Chapter 6 describes the design of the Graphics Pre-processor Computer (GPC). This computer takes data from the BDDAS, reduces it to vector coordinates, and transmits it to the Graphics Translator. The GPC was derived from Advanced Micro Devices, Inc., Super Sixteen bit-slice computer.

This design has met NASA requirements as listed below:

1. Low cost
2. Use in-house equipment, if possible
3. Little to no modification
4. No flicker and continuous motion (in Post-Processing mode)
5. Selectable screen formats

REFERENCES

1. _____, BDDAS System Manual, Ballston Lake, N.Y.: Shaker Research Corp.

2. G. V. Brown, "Engine Structural Dynamics Program; Lewis Research Center Spin Rig," NASA Lewis Workshop on Structural Dynamics, Lewis Research Center, Cleveland, Ohio, June 15, 1982.

3. L. J. Kiraly, "Digital System for Dynamic Turbine Engine Blade Displacement Measurements," NASA TM-81382, 1980.

4. M. Fry, "Real-Time Graphics in Command/Control Situation Displays," Computer, Vol. 15, No. 4, Apr. 1982, pp. 9-17.

5. L. A. Leventhal, Introduction to Microprocessors: Software, Hardware, Programming, Englewood Cliffs, N. J.: Prentice-Hall, 1978.

6. J. L. Frarey, N. J. Petersen, and D. A. Hess, "Turbojet Blade Vibration Data Acquisition Design and Feasibility Testing," SRC-TR-78-36, Shaker Research Corp., Ballston Lake, N. Y., NASA CR-159505, 1978.

7. B. A. Artwick, Microcomputer Interfacing, N. J.: Edgewood Cliffs, Prentice-Hall, 1980.

8. A. L. Burton, Cinematographic Techniques in Biology and Medicine, New York: Academic Press, 1971.

9. H. E. Ennes, Television Broadcasting: Equipment, Systems, and Operating Fundamentals, Indianapolis: Howard W. Sams, 1971.

10. E. L. Levitan, Electronic Imaging Techniques: A Handbook of Conventional and Computer-Controlled Animation, Optical, and Editing Processes, New York: Van Nostrand Reinhold Company, 1977.

11. L. Levi, Applied Optics, vol. 2, New York: John Wiley and Sons, 1980.

12. _____, "IEEE Standard Digital Interface for Programmable Instrumentation," ANSI/IEEE, Std. 488-1978, Nov. 1978.

13. I. H. Witten, Communicating with Microcomputers: An Introduction to the Technoloby of Man-Computer Communication, New York, Academic Press, 1980.

14. J. Dugundji and D. Bundas, "Standing Wave Flutter Analysis of Tuned and Mistuned Blades," NASA/Lewis Workshop on Structural Dynamics, NASA Lewis Research Center, Cleveland, Ohio, June 15, 1982.

15. J. C. Macbain, "Vibrations of Twisted Plates-Numerical Models Vs. Experiment," NASA/Lewis Workshop on Structural Dynamics, NASA Lewis Research Center, Cleveland, Ohio, June 15, 1982.

16. J. H. Griffin and A. Sinha, "The Simulation of Aerodynamic Damping/Mistuning/Friction Interaction in Bladed Disk Vibration," NASA/Lewis Workshop on Structural Dynamics, NASA Lewis Research Center, Cleveland, Ohio, June 15, 1982.

17. W. C. Nieberding and J. L. Pollack, "Optical Detection of Blade Flutter," NASA TM X-73573, 1977.

18. W. T. Thomson, Theory of Vibration with Applications, Englewood Cliffs, N. J.: Prentice-Hall, 1972.

19. G. J. Myers, Digital System Design with LSI Bit-Slice Logic, New York: John Wiley and Sons, 1980.

20. _____, Advanced Micro Computers, Advanced Microprogramming Development System, Santa Clara, Ca., Advanced Micro Computers, 1978.

21. _____, "Build a Microcomputer, Ch. 9: Super Sixteen, Sunnyvale, Ca.: Advanced Micro Devices, 1979.

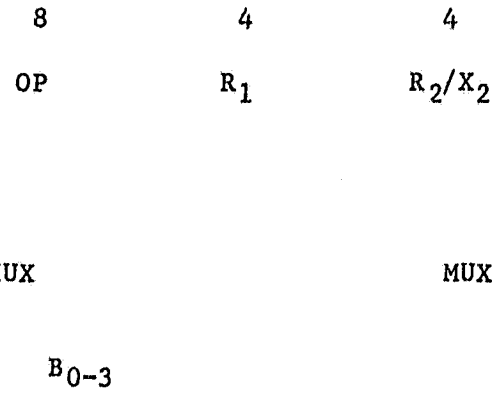
22. S. Libes and M. Garetz, Interfacing to S-100/IEEE 696 Microcomputers, Berkeley, Ca.: Osborne/McGraw-Hill, 1981.

APPENDIX - SUPER SIXTEEN MICROCODE DEFINITIONS

		Definition
95	RTB	Routes second register field to B-RAM of Am2903.
92	Z ß Z_1	Loads the value in the Z Register into the Z_1 Register at the beginning of the microcycle.
91	CCEN	Enables the CC input of the Am2910
ALU		
90	WORD	These bits control the four Am2903's. The function of Ea, OEY, OEB, and I_{8-0} to operate on words (16-bits). When disabled (HIGH) the ALU operates on bytes (the least significant byte). This bit disabled blocks WE to the upper two Am2903's and turns off their Y outputs.
89	EA	
88	OEY	
87	OEB	
86	I_8	
85	I_7	Zeros should be forced to the upper 8 bits of the Y bus via the PCU to allow the zero status to operate correctly when the WORD bit is disabled. Also, when disabled the status (C, OVR, S) sent to the Am2904 is taken from the second Am2903 (numbering 0-3 least significant to most significant slice) instead of the most significant Am2903.
84	I_6	
83	I_5	
82	I_4	
81	I_3	
80	I_2	
79	I_1	
78	I_0	
77	ENTREG	Enable Transfer Register - enables the Transfer Register onto the DA input bus of the Am2901A's and Am2903's.
76	LDTREG	Load Transfer Register - loads the Transfer Register from the Y bus.
75	ENCTR	Enable I Register Counter - enables the I Register Counter (I_{7-14}) to count. This value is used to address the general registers during stack intrauctions and by incrementing or decrementing this value the microprogram can read or write successive registers.
74	INC	I Register INC/DEC - the value in I_{7-14} can be either57 incremented (if this bit is HIGH) or decremented.
73	PCUCD	PCU Transceiver Disable - when HIGH this bit disables the PCU Transceivers from receiving of transmitting data.
72	PCU ß Y	PCU Transceiver Control - when HIGH this bit allows the PCU Transceivers to pass data from PCU to the Y bus. [WORD high (microbit 90) disables the least significant 8 bits of these transceivers.] When LOW data passes from the Y bus to the MAR.
71	LDMAR	Load Memory Adress Register (MAR) - this bit loads the Memory Address Register.

		Definition
70	LDD	Load D Register - this bit loads the D Register with data from the Y bus.
69	Z_1 ß I	Load Z_1 into I Register - this bit loads data from Z_1 into the I Register. The I Register holds only the upper 16 bits of the instruction.
68	ENZ_0	Enable Z_0 DA - this bit LOW enables the Z_0 Register onto the ALU DA.
67	PSW	Enable PSW - this bit LOW enables the PSW onto the ALU DB
66	SHIFT EN	Shift Count to Am2910 - this bit LOW enables the least significant four bits of the instruction (I_{0-3}) onto the Am2910 sequencer. This allows the value to be entered into the Am2910 internal counter to be used during shift instructions.
65	BRIEN	Branch Instruction Enable - this bit LOW enables I_{4-7} of the Instruction Register onto the Am2904 I_{0-3} input. The I_{0-3} inputs control the tests of the status register.
PCU		
64	$PCUI_7$	
63	$PCUI_3$	
62	$PCUI_2$	These bits control the PCU which is designed around four Am2901's. The $PCUI_7$, $PCUI_3$, $PCUI_2$, $PCUI_1$, and $PCUI_0$ bits connect directly to the Am2901 I_7, I_3, I_2, I_1 and I_0 respectively. The $PCUA_2$-$PCUA_0$, $PCUB_2$-$PCUB_0$, connect to the A and B Address inputs of the Am2901. I_4, I_5, I_8, A_3 and B_3 are tied to ground. I_6 is tied to I_7.
61	$PCUI_1$	
60	$PCUI_0$	
59	$PCUA_2$	
58	$PCUA_1$	
57	$PCUA_0$	
56	$PCUB_2$	
55	$PCUB_1$	
54	$PCUB_0$	
53	REQB	Request Bus - this bit requests use of the system bus. This request is made the microcycle preceding a Memory Request or use of the bus for an I/O transfer. If the request is not honored, the processing of the next micro-instruction is halted until the acknowledge is issued.
52	MREQ	Memory Request - this bit requests the memory to do a read or write operation.
51	HREQ	Hold Request - this bit LOW blocks the bus controller from releasing the system bus to another device. Normally a Bus Request is cleared as soon as the Bus Acknowledge is issued. HREQ holds Bus Request and prevents any other device from using the bus.

		Definition
50	WRITE	Memory Write/READ - this bit indicates to the memory the MREQ is for a write operation (if HIGH) and a read operation (if LOW).
49	MWORD	Memory Word/BYTE - the Memory Word/BYTE microbit specifies whether the memory operation will be a word operation or abyte operation. If the operation specified is a byte operation the least significant address bit determines which byte of the two byte pair in memory is affected. If the LSBit is a zero, the most significant byte is read or written, and the LSBit is a one, the least significant byte is read or written.
48	IMMD	EN Immediate DA Bus - this bit LOW enables the 16-bit immediate value (least significant 16 bits of the microinstruction) to the ALU DA bus.
47	ROM/I	ROM/I REG Enable - this bit enables either the ROM bits 42-35 or the I Register bits I_{0-7} onto the A/B address inputs of the ALU according to the following:



46	IOEN	I/O Control Register Enable - this bit loads the I/O Control Register with microbits 42-35.
45	INTDIS	Am2914 Interrupt Disable - this bit disables the Am2914 Interrupt Controller from recognzing interrupt requests.
44	INTRIEN	Am2914 ENI_0-ENI_3 - this bit is the instruction enable for the Am2914. The instruction inputs I_{0-3} are connected to microbits 35-38 respectively.
43	SHFTEN	Am2904 Shift Enable - this bit is connected to the shift enable of the Am2904. The shift controls I_{6-10} are connected to microbits 35-39 respectively.

		Definition	
42	$CNTLB_7$		
41	$CNTLB_6$		
40	$CNTLB_5$	This control field is used to provide several different	
39	$CNTLB_4$	functions as defined by the previously described control	
38	$CNTLB_3$	strobes (microbits 47-43).	
37	$CNTLB_2$		
36	$CNTLB_1$		
35	$CNTLB_0$		
34	OECT	OUT EN CONDITIONAL TEST	
33	EZ	ENZERO	
32	EC	EN CARRY	These bits are used to control
31	ES	EN SIGN	the Am2901. Their functions are
30	EOVR	EN OVERFLOW	defined in Figure 21. OECT is
29	CEM	EN MACRO STATUS	used to enable the test output
28	CE	EN MICRO STATUS	of the Am2904 to the CC input of
27	I_{12}	CARRY OUT CONTROL	the Am 2910.
26	I_{11}	CARRY OUT CONTROL	
25	$TEST_5$		
24	$TEST_4$	These bits determine which test is to be performed for	
23	$TEST_3$	the conditional branch and stack functions. The various	
22	$TEST_2$	tests are listed in Figure 25. The testing is done both	
21	$TEST_1$	in the Am2904 and an 8 to 1 multiplexer.	
20	$TEST_0$		
19	NAC_3	291013 These bits are connected to the I_{3-0} inputs of	
18	NAC_2	291012 the Am2910 to control the sequencing of the	
17	NAC_1	291011 microprogram. Their definitions are listed in	
16	NAC_0	291010 Figure 26.	
15	M_{15}		
14	M_{14}		
13	M_{13}		
12	M_{12}		
11	M_{11}		
10	M_{10}		
9	M_9	These bits provide the branch address for the Am2910	
8	M_8	and the 16-bit immediate field.	
7	M_7		
6	M_6		
5	M_5		
4	M_4		
3	M_3		
2	M_2		
1	M_1		
0	M_0		

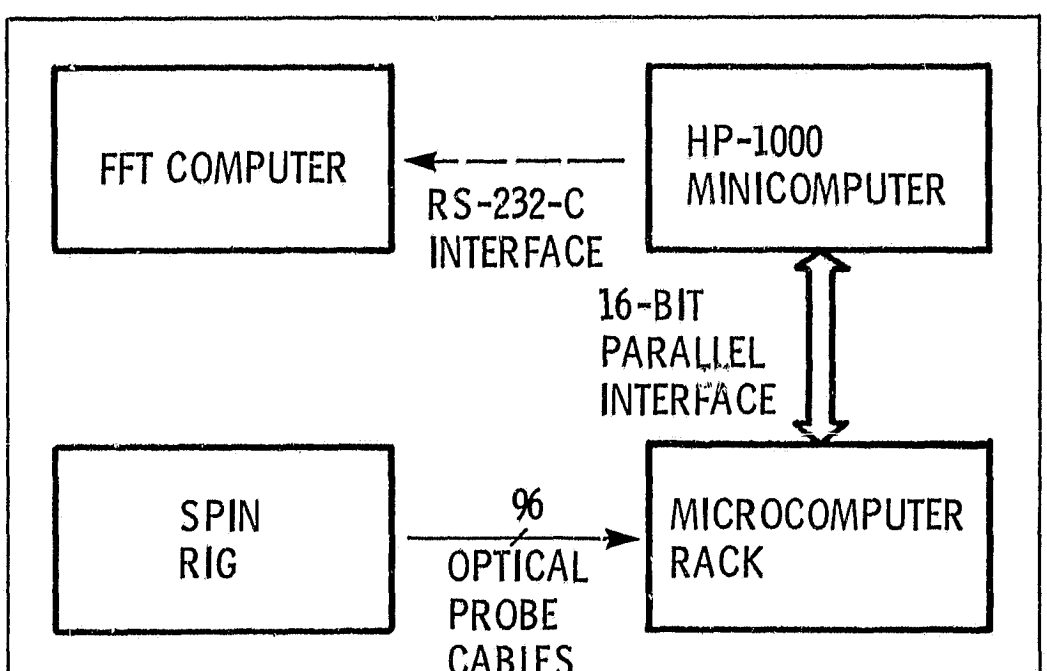


Figure 1. - BDDAS block diagram.

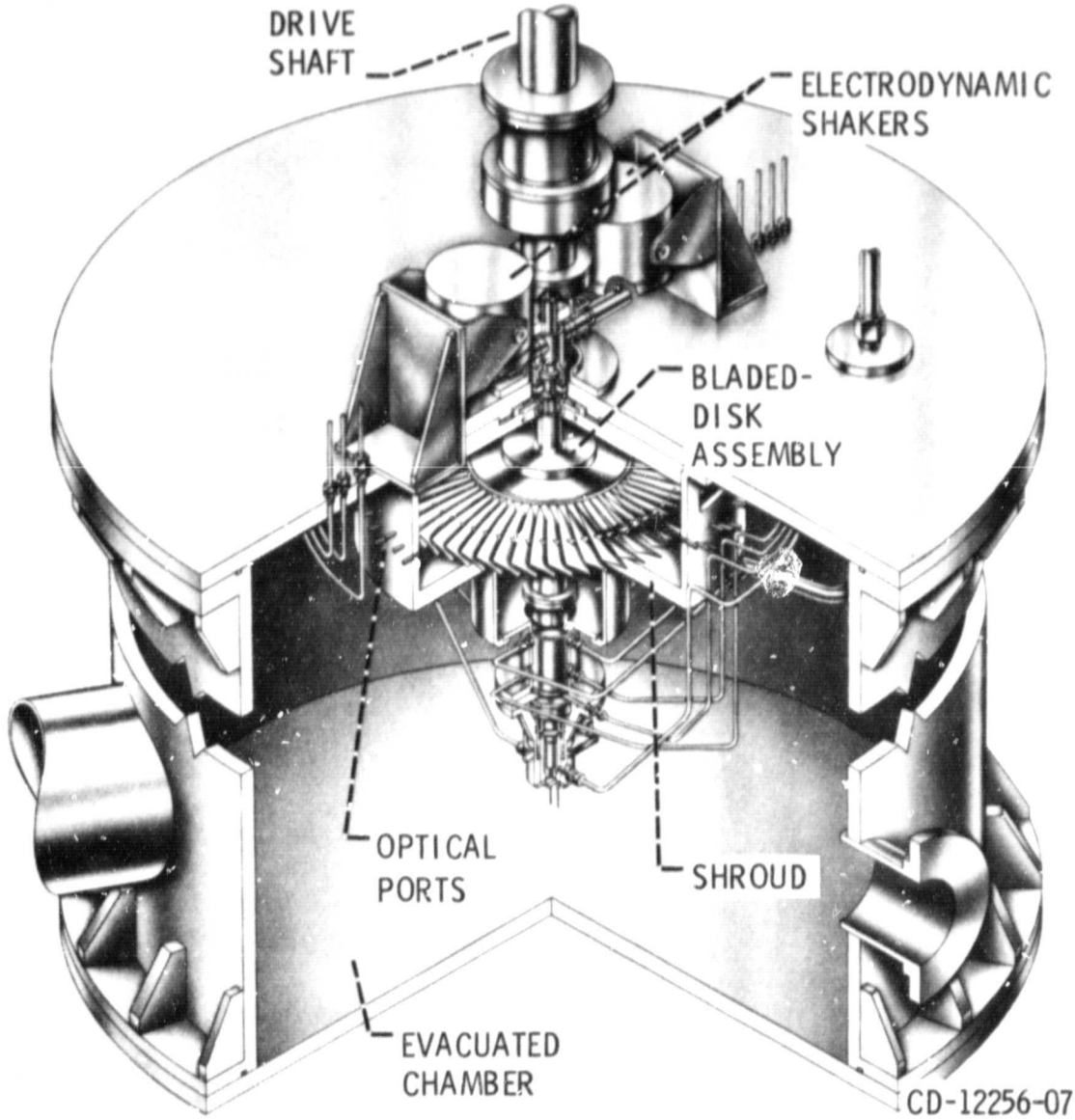
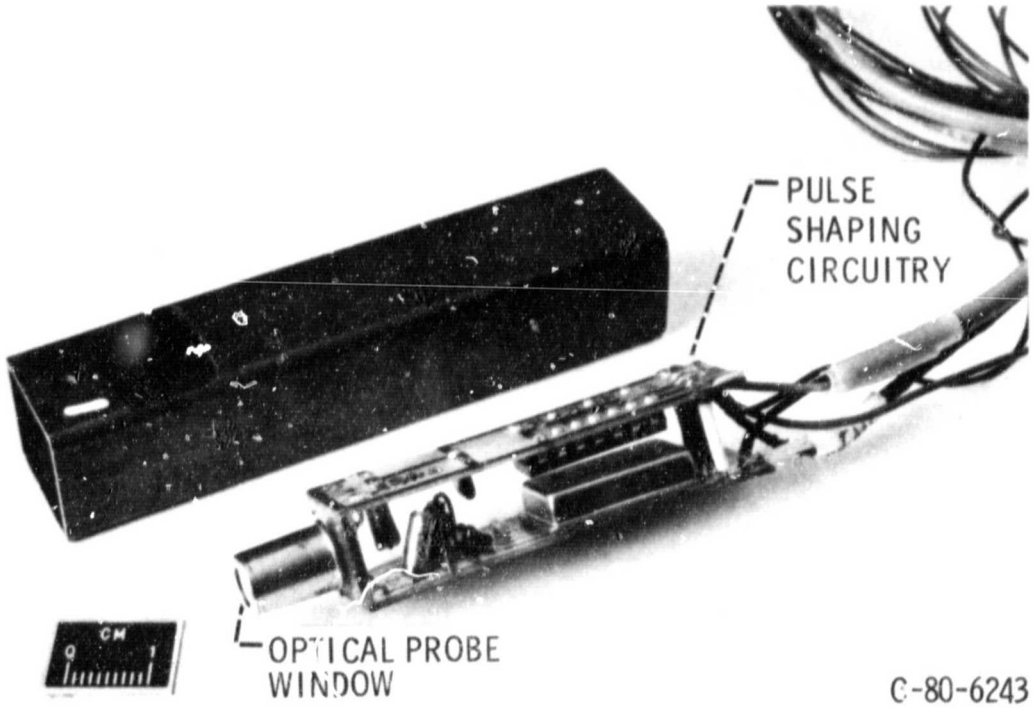


Figure 2. - Spin rig with Bladed-Disk assembly mounted.



C-80-6243

Figure 3. - Optical probe.

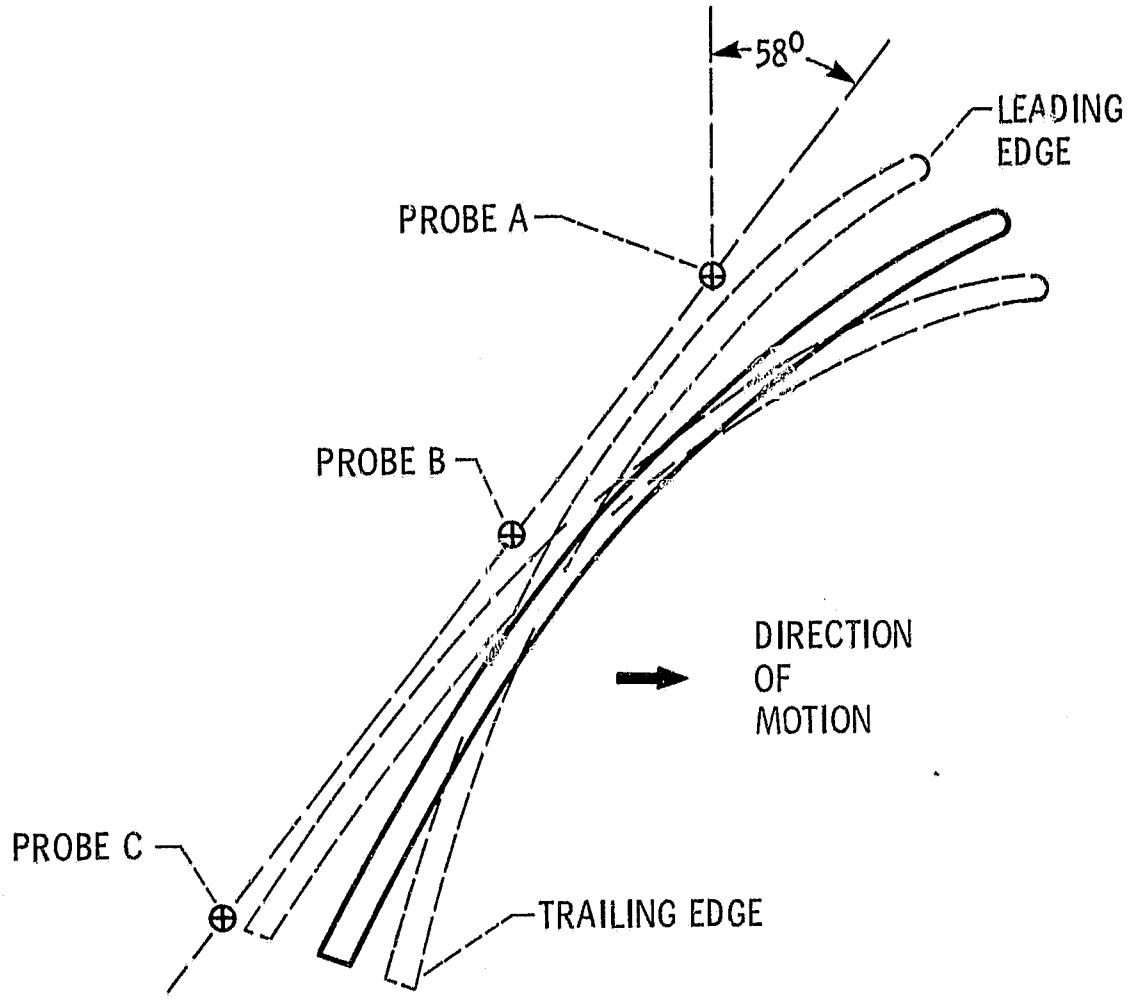


Figure 4. - Optical probe port measuring a vibrating blade.

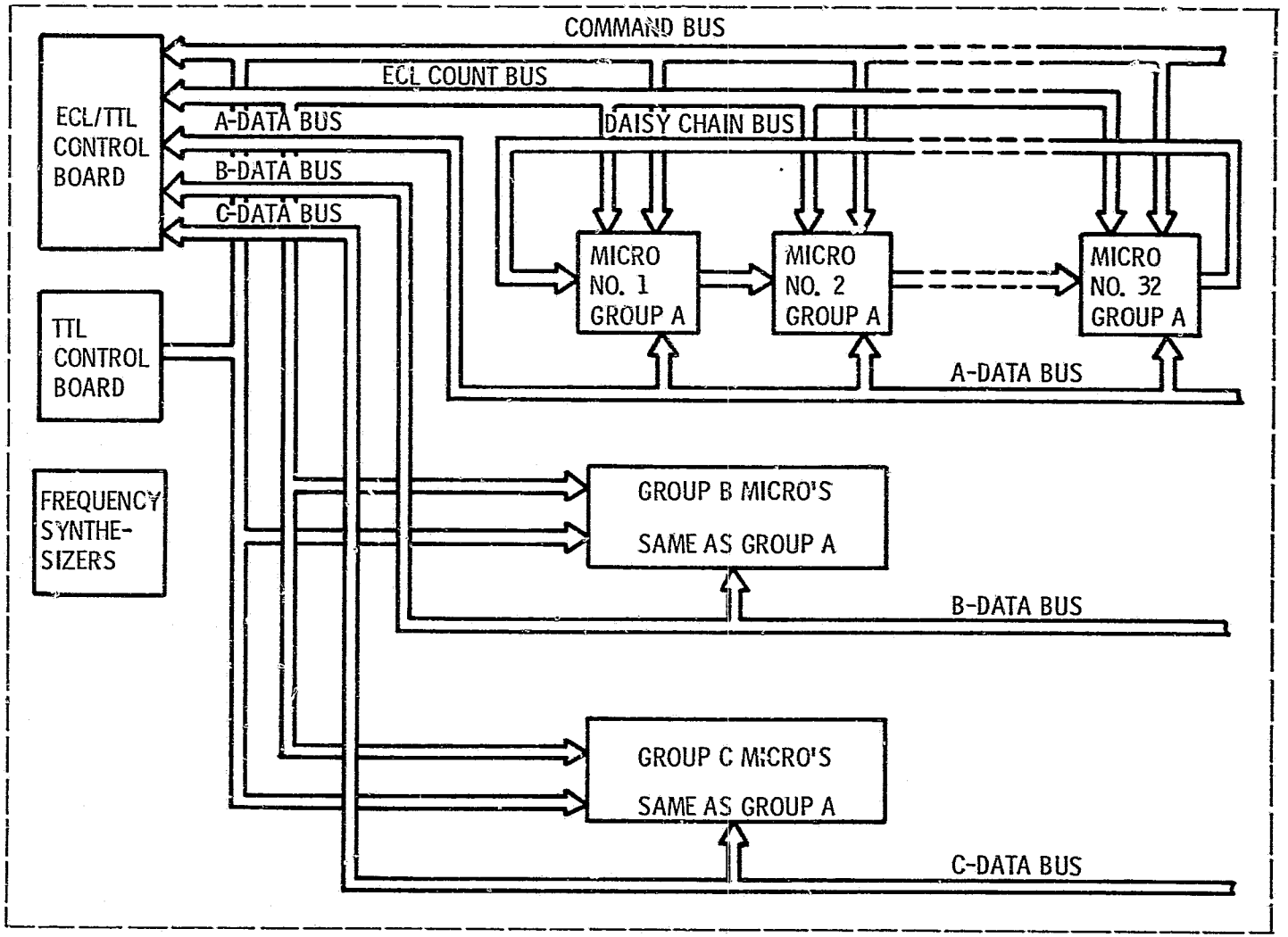


Figure 5. - Microcomputer Rack bus architecture.

FAN BLADE VIBRATION EXPERIMENT

DATA ENTRY 9:33 AM MON., 22 MAR. 1982

NUMBER OF BLADES = 64
NUMBER OF ACTIVE PORTS = 16
NUMBER OF BLADES TO SKIP = 0
DATA POINTS/BLADE = 1024
DEFLECTION: FULL SCALE 0.00000 in.
RESOLUTION 0.00000 in.
MAX ERROR: OCNTS 0.00000 in

RPM = 0 NOT STABILIZED
ANGLE CLOCK FREQ. = 0.000 KHz
ARM DELAYS AT PORT 1:
A = 0.00000 deg
B = 0.00000 deg
C = 0.00000 deg
EXPECTED VALUES FROM: NONE . . . . . .

COMMAND MENU

1. COLD START PROCEDURE
2. SPECIFY NUMBER OF BLADES
3. SPECIFY NUMBER OF SAMPLE PORTS
4. SPECIFY NUMBER OF BLADES TO SKIP
5. SPECIFY NUMBER DATA PTS./BLADE
6. SPECIFY DEFLECTION RESOLUTION
7. READ EXPECTED VALUES FROM FILE
8. START NORMALIZATION RUN
9. START PRESAMPLE MODE
10. START DATA COLLECTION RUN
11. STORE DATA IN DISK FILE
12. MOVE DATA TO FIXED FORMAT DISK
13. SYNCHRONIZE ANGLE CLOCK
99. EXIT TO RTE IV FILE MANAGER

ENTER COMMAND (OR <CR> FOR MENU):

Figure 6. - Menu for program BLADE.

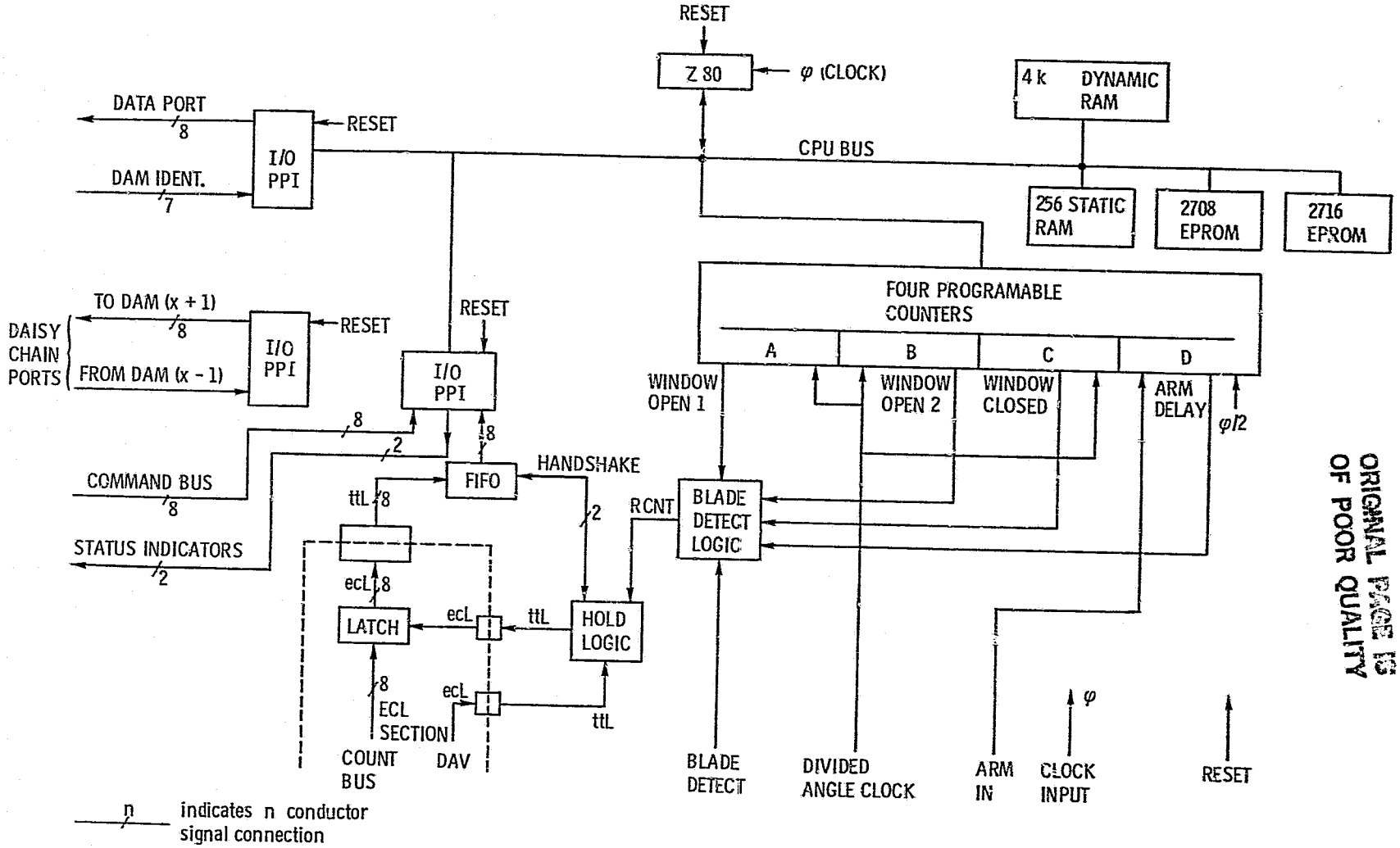


Figure 7. – DAM block diagram.

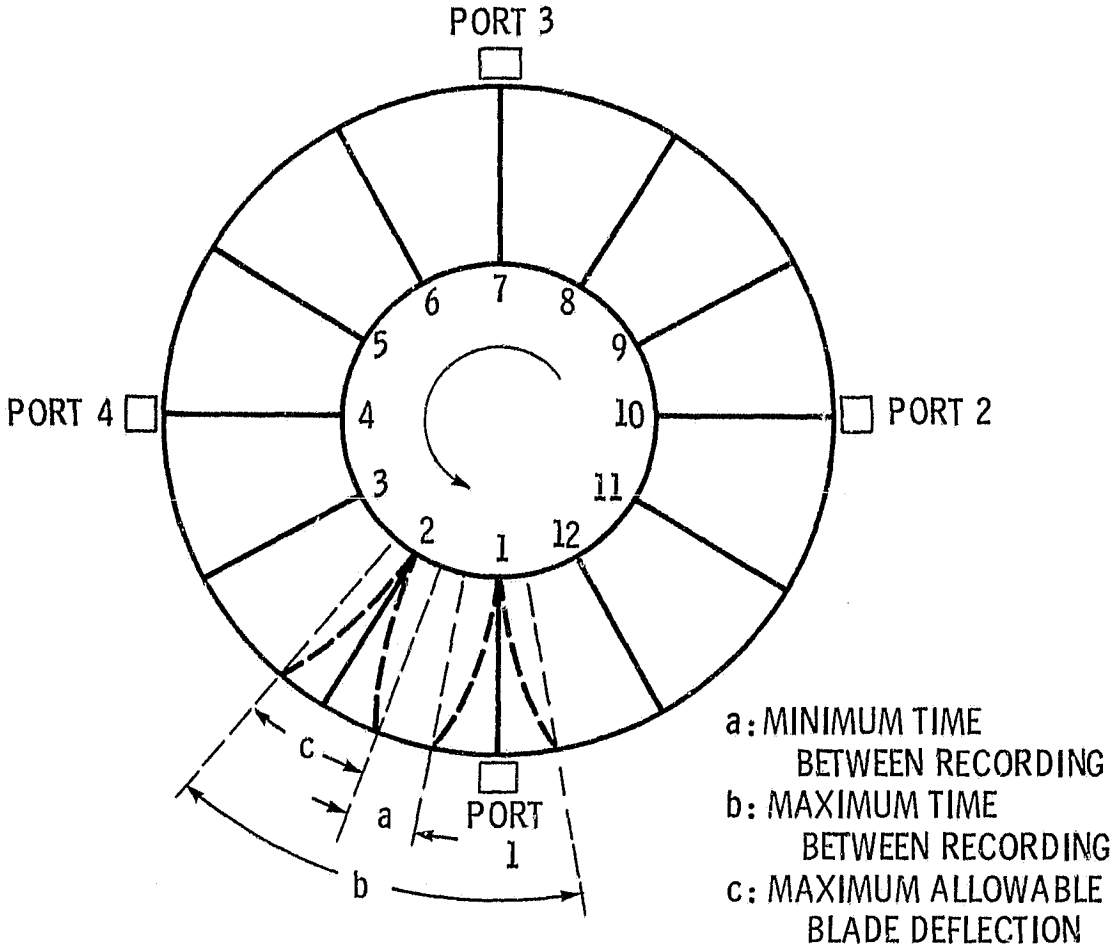


Figure 8. - Diagram showing blade excursions and timing between data samples.

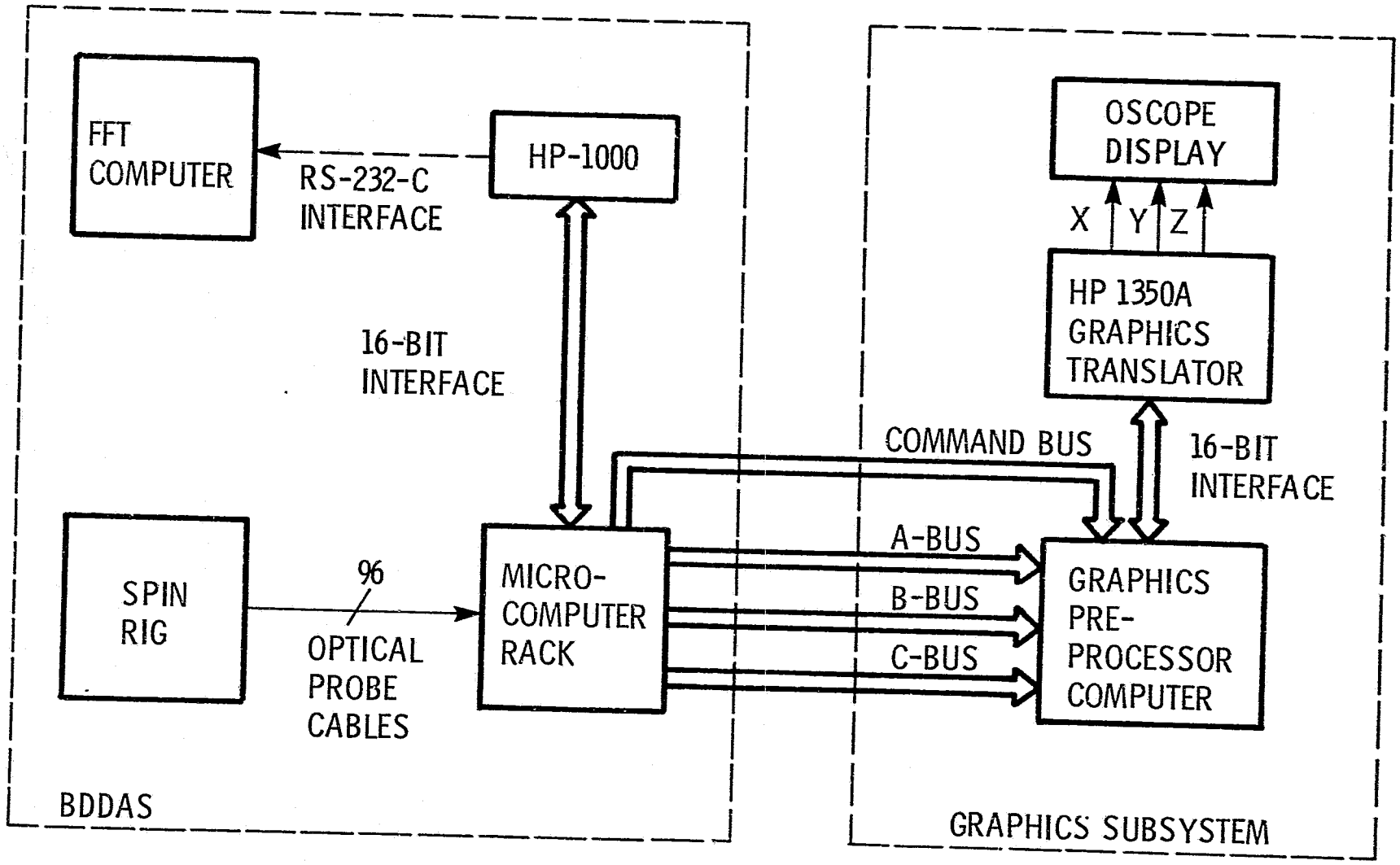
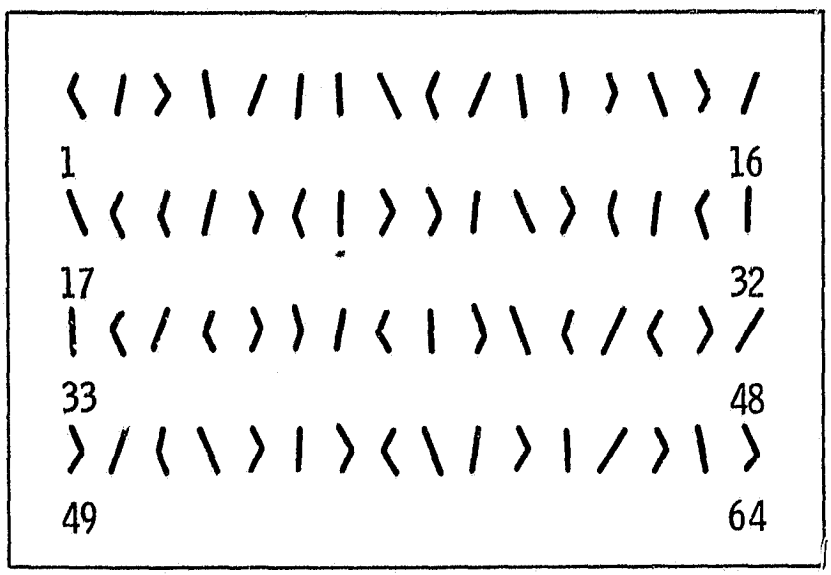
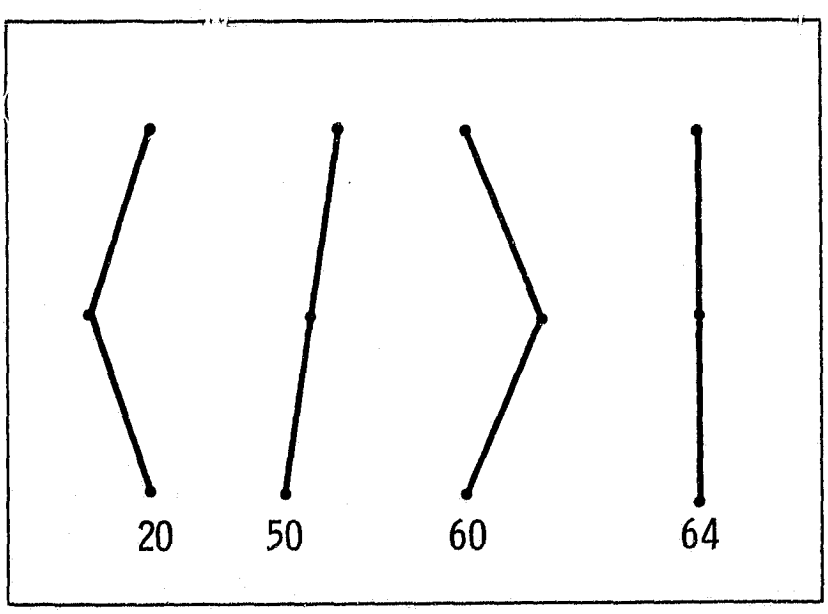


Figure 9. - BDDAS to GS interface.



COMPLETE DISPLAY



SELECTED BLADE DISPLAY

Figure 10. - Typical screen for the oscilloscope display.

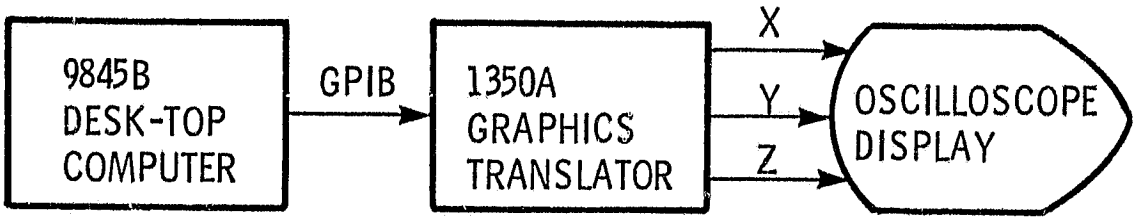


Figure 11. - Equipment setup for throughput experiment.

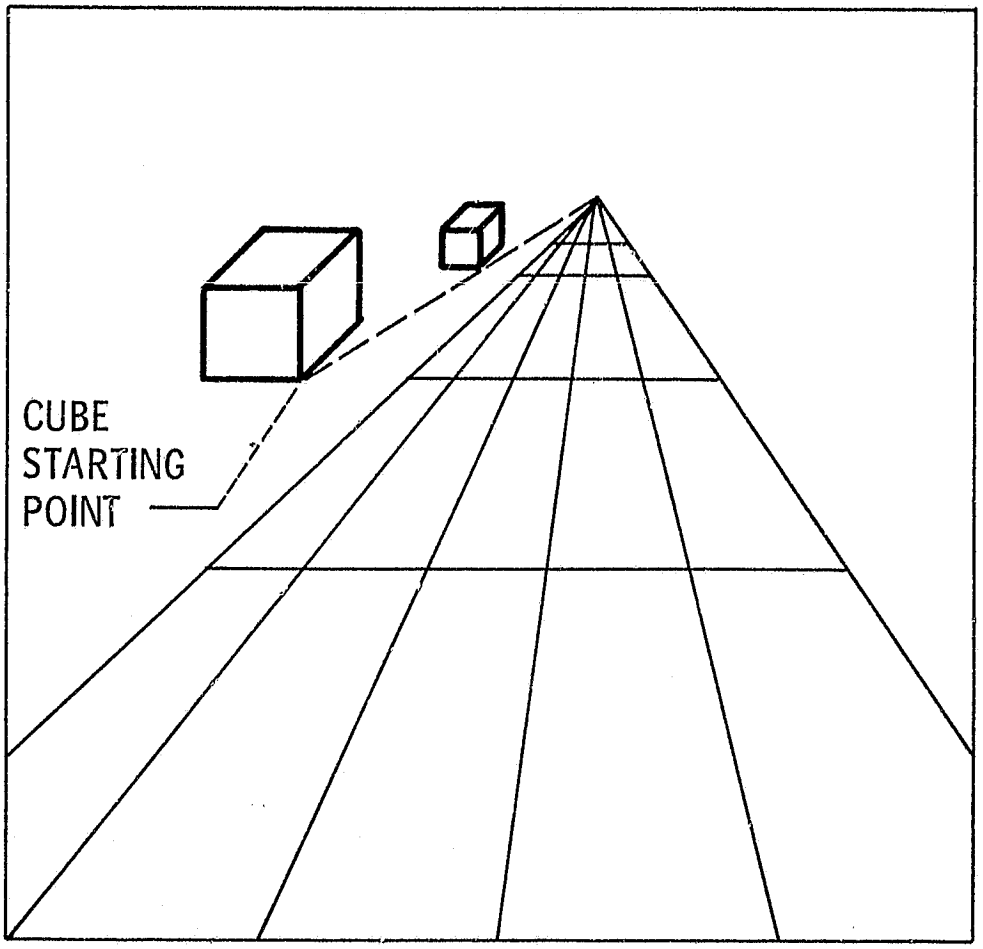


Figure 12. - Phosphor test screen for the oscilloscope display.

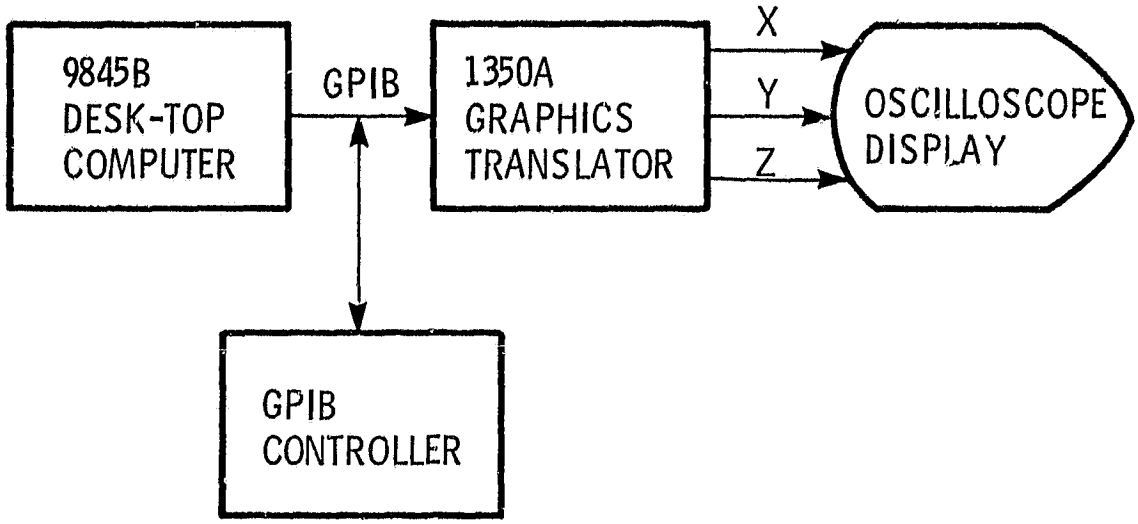


Figure 13. - Equipment setup for the framing experiment.

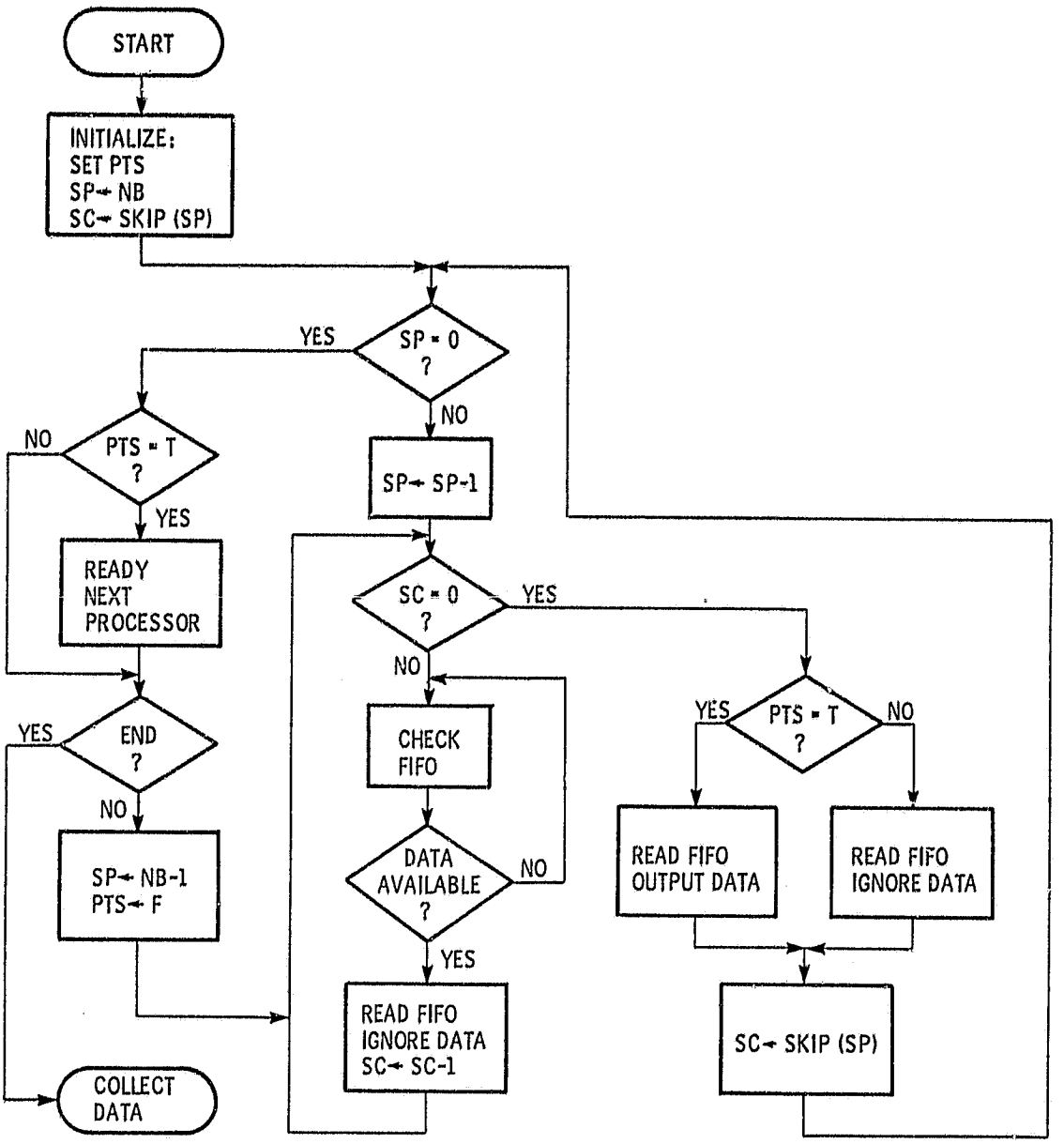
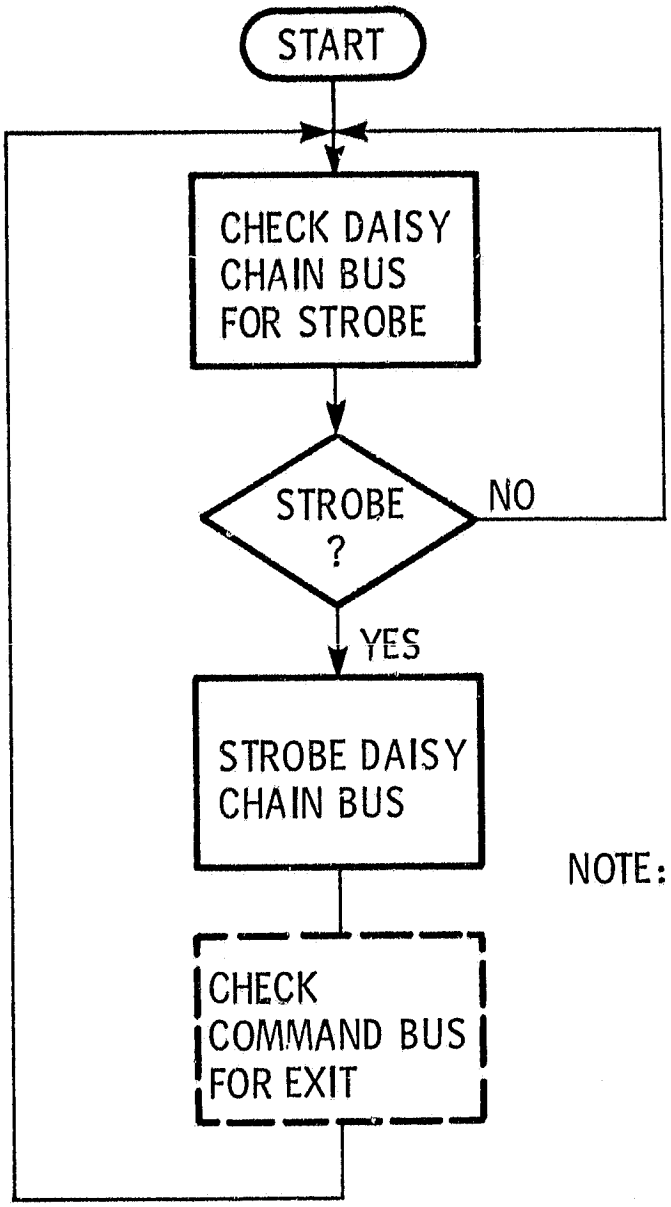


Figure 14. - Flow chart for the real-time agorithm.



NOTE: EXIT LOOP WHEN RESET BY HOST COMPUTER

Figure 15. - Non-participating DAM program flow chart.

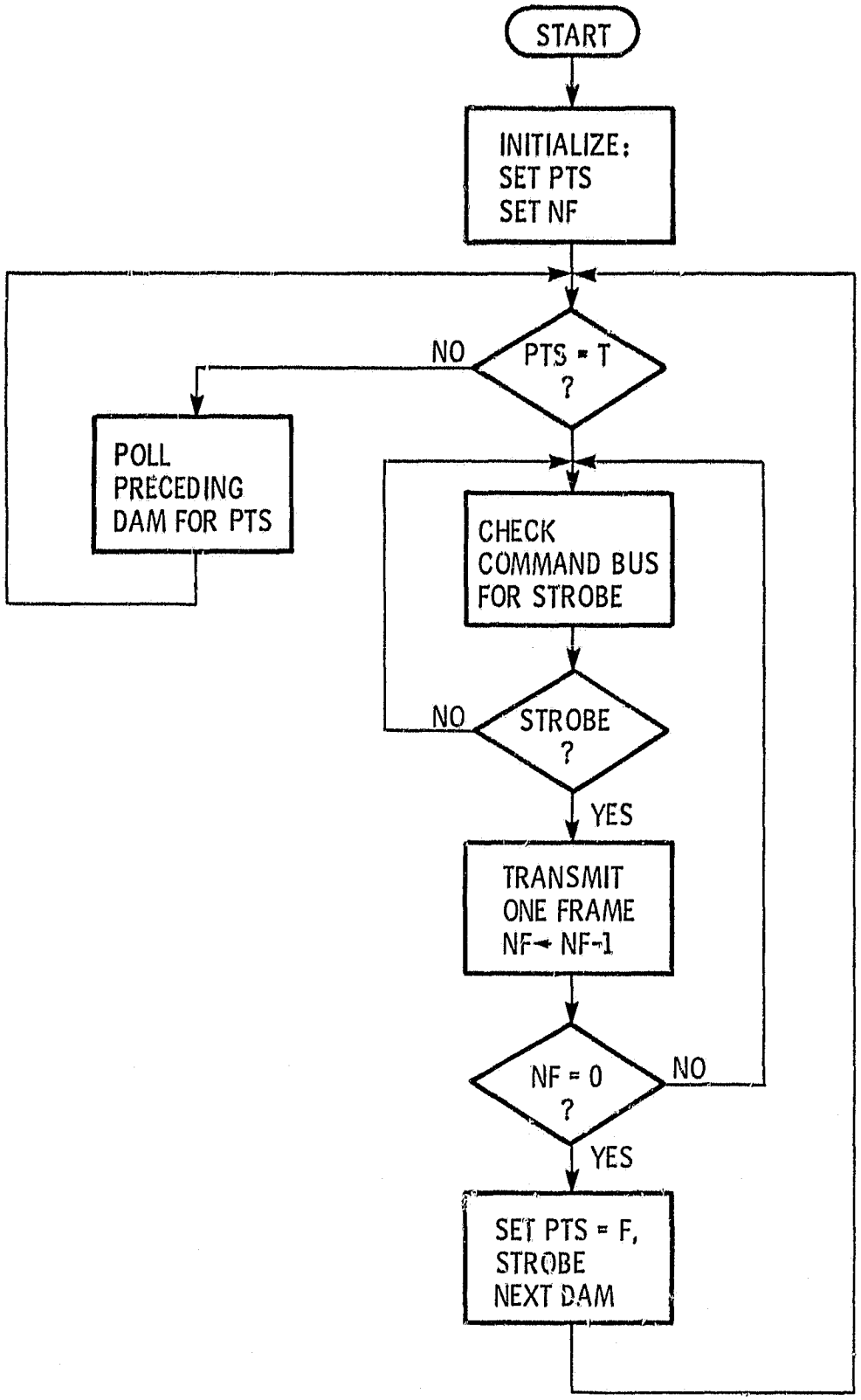


Figure 16. - Flow chart for the post-processing algorithm.

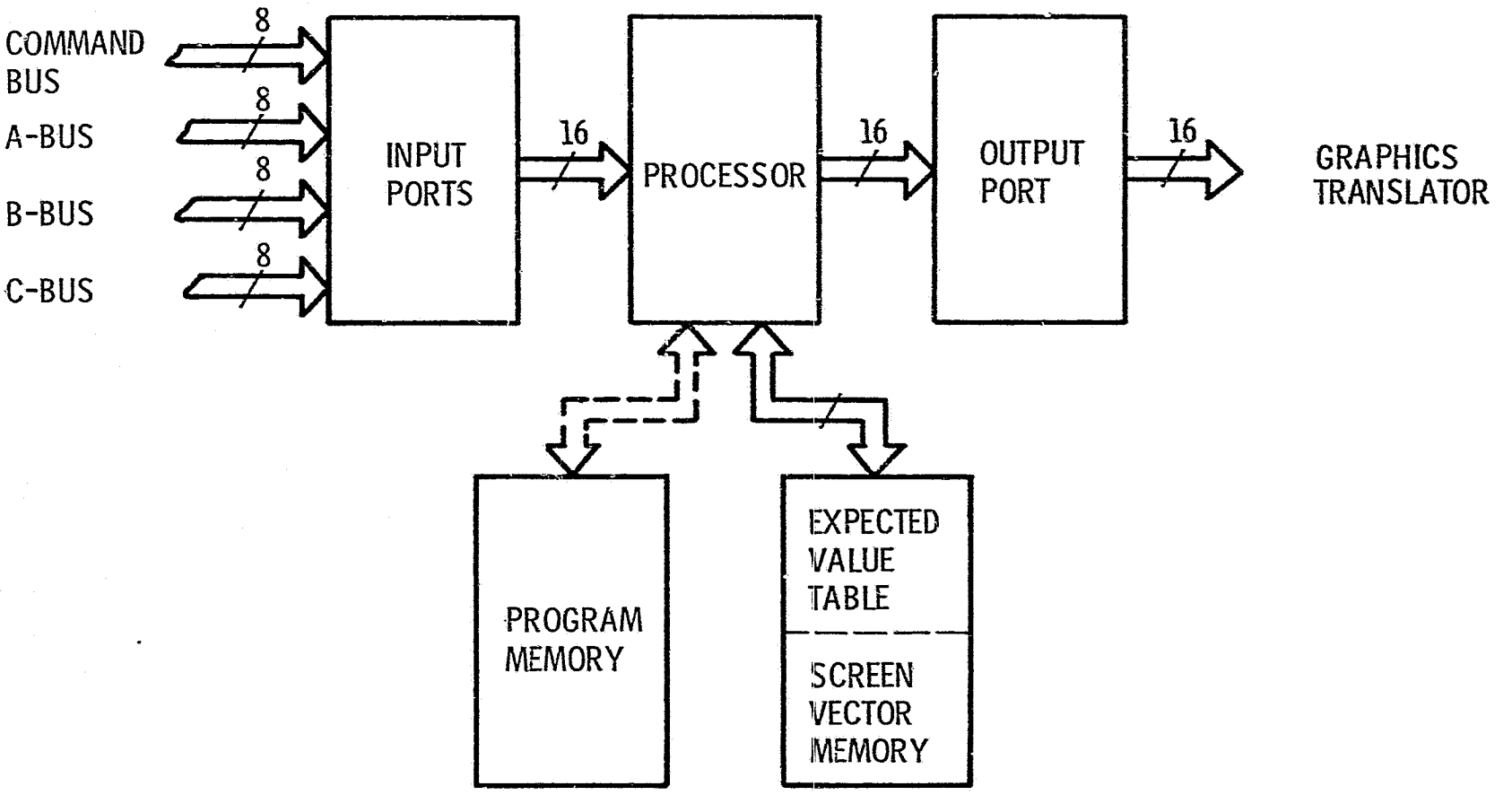


Figure 17. - Block diagram of the GPC.

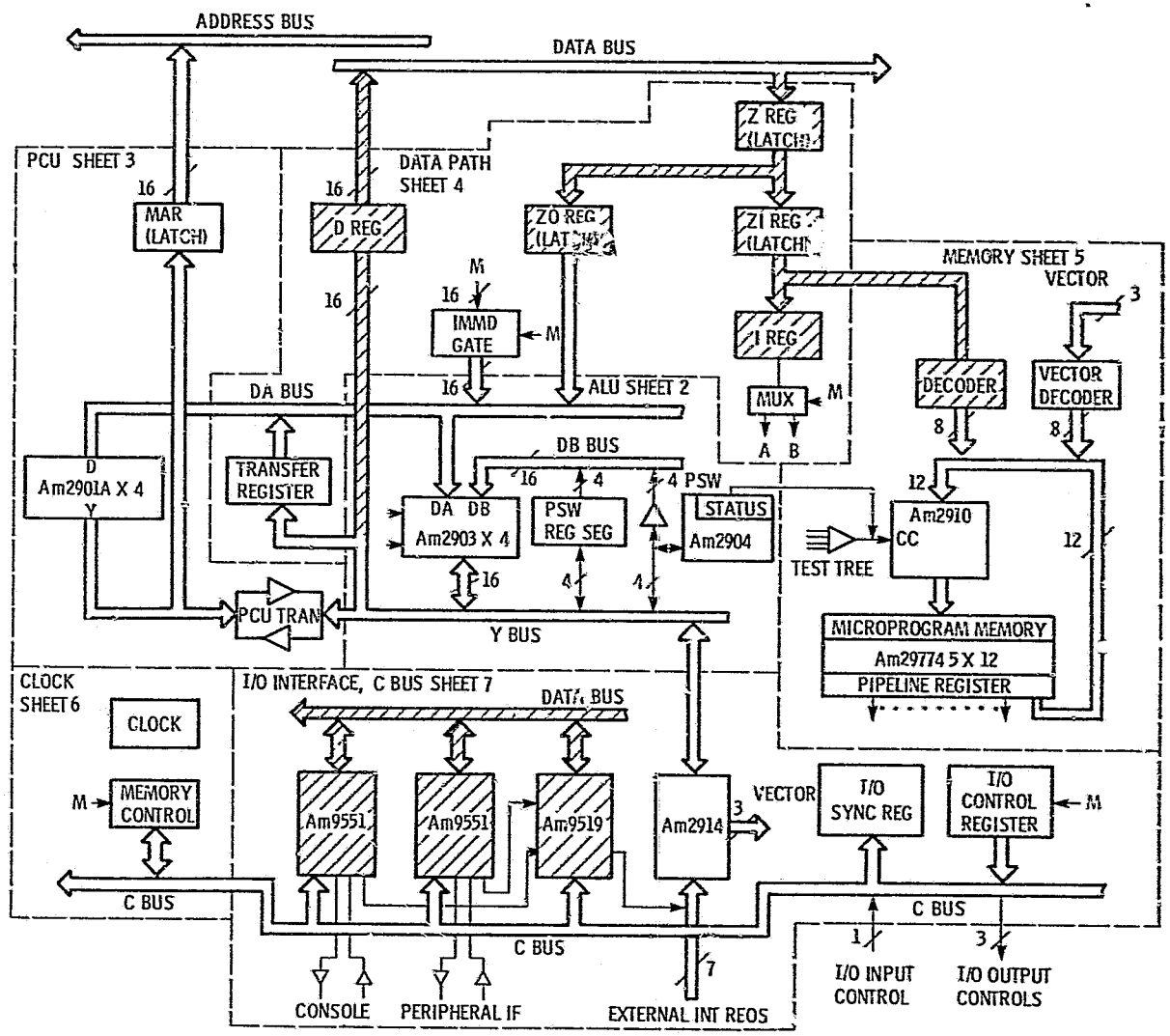


Figure 18. - Super sixteen architecture with components to be modified shaded.

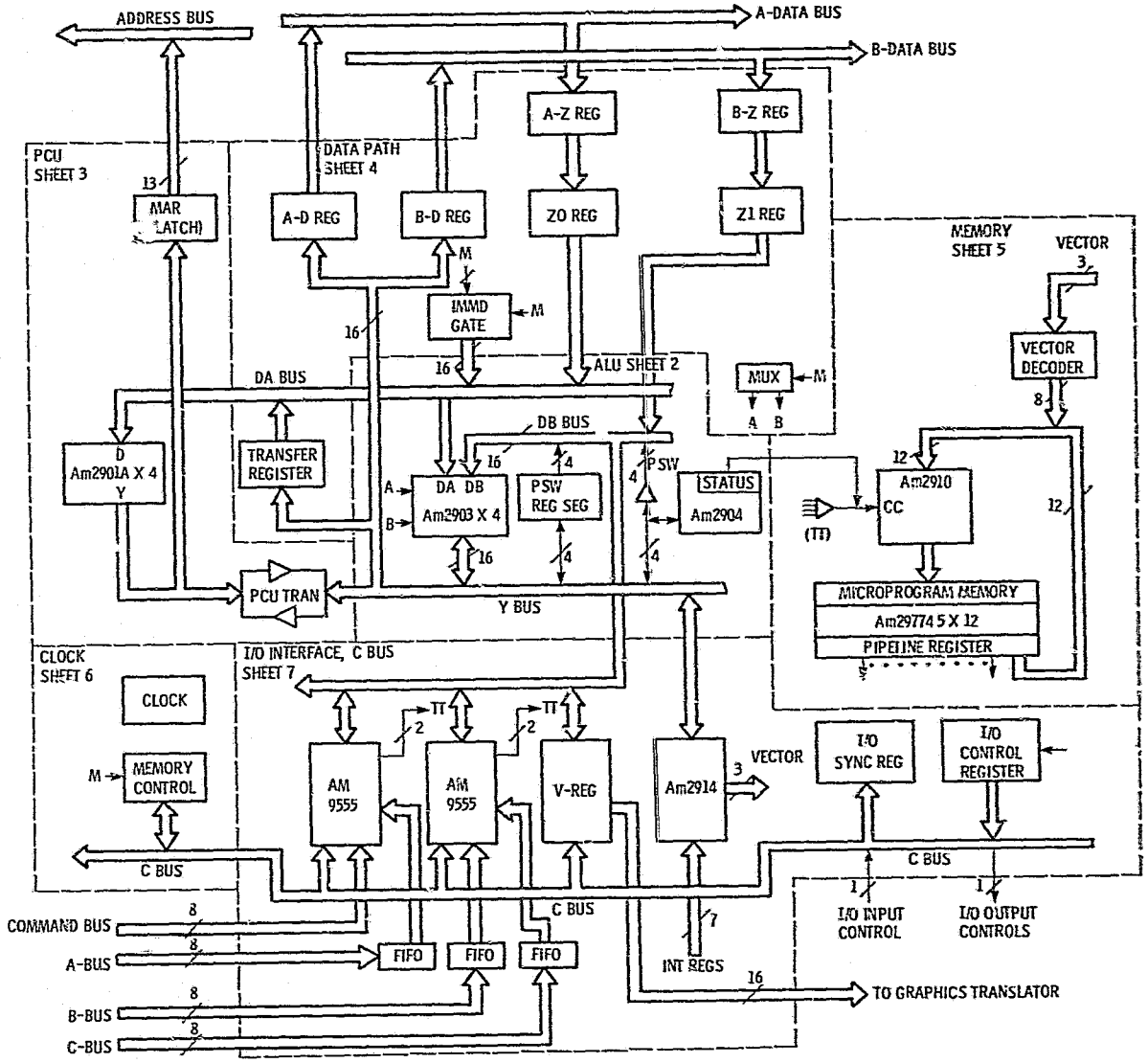


Figure 19. – GPC detailed block diagram.

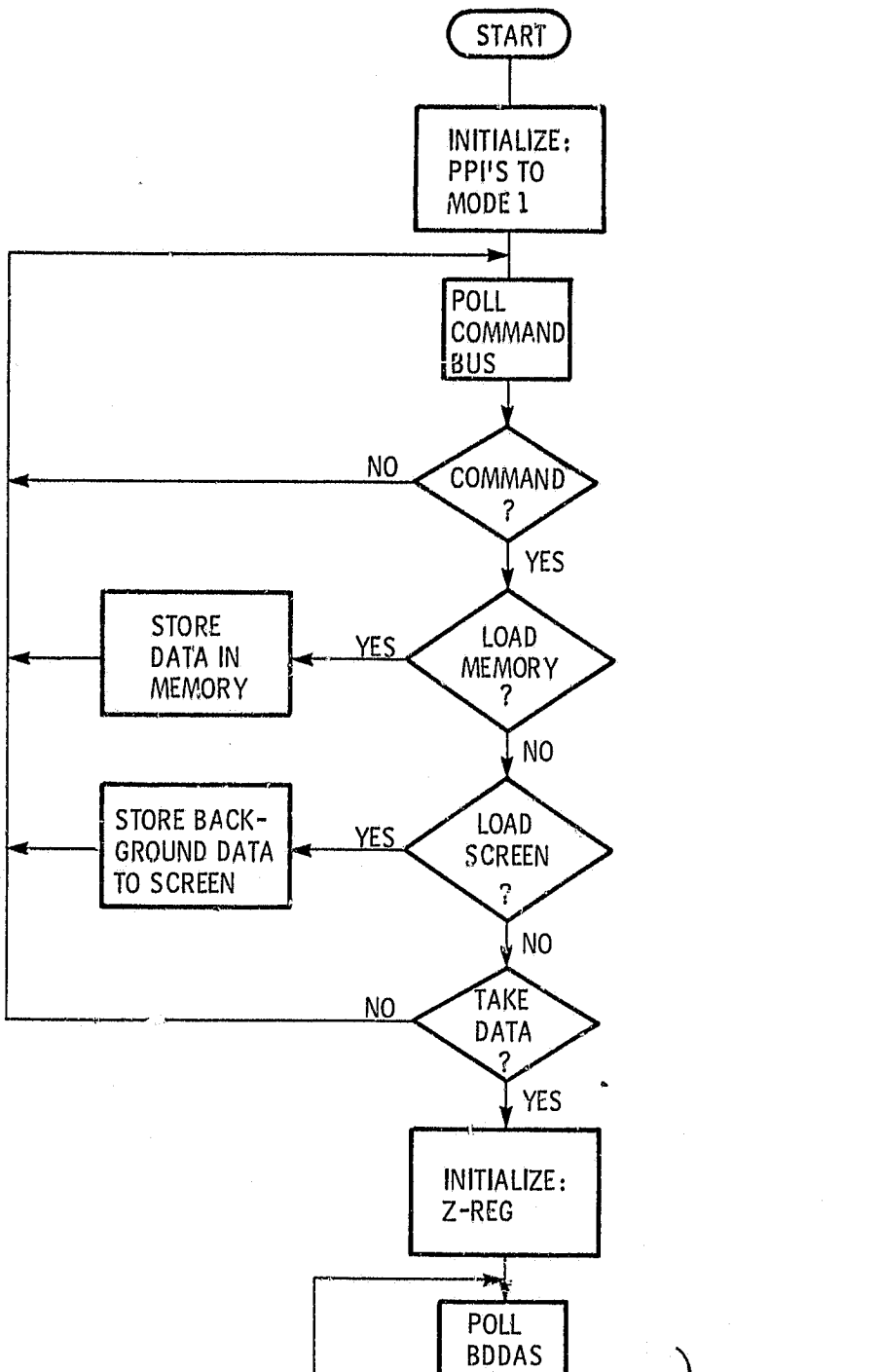


Figure 20. - Microcode program flow chart for the GPC.

16-BIT COMPUTER					
MICRO CONTROL WORD BIT DEFINITIONS					
MISC		ALU (13)	Data Path (13)	Program Control (11)	Memory Control (5)
95	9291	90898887868584838281807978	77767574737271706968676665	64636261595857565554	5352515049
$\overline{RTB}$	(BP) Z → ZI	$\overline{WORD}$	$\overline{ENTREG}$	$PCUI_7$	REQB
	$\overline{CCEN}$	$\overline{EA}$	$\overline{LDTREG}$	$PCUI_6$	$\overline{MREQ}$
		$\overline{OEY}$	$\overline{ENCTR}$	$PCUI_2$	$\overline{HREQ}$
		$\overline{OEB}$	INC	$PCUI_1$	WRITE
		I_8	PCUCD	$PCUI_0$	MWORD
		I_7	PCU → Y	$PCUA_2$	
		I_6	LDMAR	$PCUA_1$	
		I_5	LDD	$PCUA_0$	
		I_4	ZI → I	$PCUB_2$	
		I_3	ENZ0	$PCUB_1$	
		I_2	$\overline{PSW}$	$PCUB_0$	
		I_1	$\overline{SHTCNTEN}$		
		I_0	$\overline{BRIEN}$		

16-BIT COMPUTER					
MICRO CONTROL WORD BIT DEFINITIONS					
Control Strobes (6)	Control Bits (8)	Status (9)	Test (6)	Sequences CNTL (4)	Next Micro Addr & Immed (16)
X X 484746454443	4241403938373635	X X 343332313029282726	252423222120	19181716	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
$\overline{IMMD}$	$CNTLB_7$	$\overline{OECT}$	$TEST_5$	NAC_3	M_{15}
$\overline{ROMI}$	$CNTLB_6$	$\overline{EZ}$	$TEST_4$	NAC_2	M_{14}
$\overline{IOEN}$	$CNTLB_5$	$\overline{EC}$	$TEST_3$	NAC_1	M_{13}
$\overline{INTDIS}$	$CNTLB_4$	$\overline{ES}$	$TEST_2$	NAC_0	M_{12}
$\overline{INTRIEN}$	$CNTLB_3$	$\overline{EOVR}$	$TEST_1$		M_{11}
$\overline{SHFTEN}$	$CNTLB_2$	$\overline{CEM}$	$TEST_0$		M_{10}
	$CNTLB_1$	$\overline{CE\mu}$			M_9
	$CNTLB_0$	I_{12}			M_8
		I_{11}			M_7
					M_6
					M_5
					M_4
					M_3
					M_2
					M_1
					M_0

Figure 21. - Changes to the super sixteen microword.