To the Graduate Council:

I am submitting herewith a thesis written by Timothy W. Bratcher entitled "The Development of an Image Recording and Processing System for an Autoranging Intensified Silicon Array Camera." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Bruce W. Bomar, Major Professor

We have read this thesis
and recommend its acceptance:

Accepted for the Council:

Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at the University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Timothy W. Bratcher

Date July 18, 1990

The Development of an Image Recording and Processing System for an Autoranging Intensified Silicon Array Camera

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Approved for public release; distribution is unlimited.

Timothy W. Bratcher

August 1990

ACKNOWLEDGMENTS

The author wishes to thank Dr. Bruce W. Bomar for his advice and assistance in the preparation of this thesis.

The author also wishes to thank the United States Air Force, the Arnold Engineering Development Center (AEDC), and the management of Sverdrup Technology, Inc., AEDC Group, for the opportunity to attend UTSI and for providing the printing services for this thesis.

A special thanks is extended to Mr. Fred G. Sherrell, Mr. J. Hal Bridges, and Mr. Ron K. Wood of Sverdrup Technology, Inc., for their assistance in the development, assembly, and checkout of this system.

References to named commercial products in this thesis are not to be considered in any sense as an endorsement of the product by the United States Air Force or the Government.

The research reported herein was performed by the AEDC, Air Force Systems Command. Work and analysis for this research were done by personnel of Sverdrup Technology, Inc., AEDC Group, operating contractor for the AEDC propulsion test facilities. Further reproduction is authorized to satisfy the needs of the U.S. Government.

ABSTRACT

An image recording and processing system has been developed for use with an autoranging camera controller and an intensified silicon array camera. The system consists of an intensified silicon target camera, autoranging camera controller, time code translator, tape interface, computer, and software. The image processing system acquires the system gain and time along with standard video, making possible quantitative radiometric data reduction and temporal correlation of data taken simultaneously from multiple systems. This system can be used in any application where it is necessary to quantify data taken over a wide dynamic range.

Using an existing autoranging camera controller and a time code translator to provide the system gain and frame times as digital data, a tape interface has been designed and implemented to modulate this data as an audio signal. The modulated signal is recorded on the audio channels of a video cassette recorder while the video is being recorded on the video channel. This digital data tape interface also provides the capability of demodulating the audio signal to recover the original digital data, which is then downloaded to a computer. The data recorded by the tape interface is not restricted to the system gain and time at each individual frame, but there is a limit of twenty bits of data that can be recorded per frame of video.

Software was written to acquire the system gain and time reference data on the computer, and to integrate this data with the video obtained using a frame grabber. As a result of this work, each frame can now be reduced to absolute radiometric units and displayed as either iso-radiance contours or gray scale plots. Digitization of video and acquisition of the gain and time data is all under computer control, allowing automation of the data reduction process and thus diminishing data reduction time while at the same time adding capabilities that were not possible with previous processing techniques.

TABLE OF CONTENTS

CHAPTER		PAGE
1.	INTRODUCTION	1
2.	DATA RECORDING SYSTEM	6
	Time Code Translator	7
	Video Recorder in Record Mode	9
	Intensified Silicon Array Camera	12
	Autoranging Camera Controller	13
	Digital Data Tape Interface in Record Mode	16
3.	DATA ACQUISITION SYSTEM	18
	Video Recorder in Playback Mode	18
	Digital Data Tape Interface in Playback Mode	20
	VCR Controller	21
	Frame Grabber	22
	Computer System	23
4.	DIGITAL DATA TAPE INTERFACE RECORD MODE	26
	Original Design	27
	Design Modifications	34
	Data Encoding and Modulation Format	36
	Record Mode Programmable Logic Devices	39

CHAPTER PAGE

4. (Continued)

Record Mode Microsequencers 40

5. DIGITAL DATA TAPE INTERFACE PLAYBACK MODE . 50

Data Demodulation and Decoding 50

Playback Mode Programmable Logic Devices 53

Playback Mode Microsequencers 56

6. DATA ACQUISITION SOFTWARE 59

DDTI Serial Data Transfer 59

Frame Grabbing 64

7. DATA REDUCTION SOFTWARE 67

Inverse Responsivity Calculation 67

Data Plotting 71

8. CALIBRATION 73

9. RESULTS AND DISCUSSION 77

10. SUMMARY AND RECOMMENDATIONS 80

Summary of Results 80

Recommendations 81

LIST OF REFERENCES 82

APPENDIX 86

VITA 88

LIST OF FIGURES

FIGURE		PAGE
1.1.	Image Processing System Overview	4
2.1.	Data Recording System	6
2.2.	Time Code Translator	7
2.3.	Time Code Unit Timing	9
2.4.	Video Recorder in Record Mode	11
2.5.	Intensified Silicon Array Camera	13
2.6.	Autoranging Camera Controller	14
2.7.	Digital Data Tape Interface in Record Mode	16
3.1.	Data Acquisition System	19
3.2.	Video Recorder in Playback Mode	19
3.3	VCR Controller	22
3.4.	Frame Grabber	23
4.1.	Intensified Camera Image Processing System	28
4.2.	DDTI Record Mode Block Diagram	37
4.3.	EPLD #1 Equivalent Circuit	41
4.4.	SAM #1 Flowchart	43
4.5.	SAM #2 Record Mode Flowchart	46
5.1.	DDTI Playback Mode Block Diagram	51

FIGURE PAGE

5.2. EPLD #2 Equivalent Circuit 54

5.3. EPLD #2 Timing Diagram 56

5.4. SAM #2 Playback Mode Flowchart 57

6.1. Serial Data Transfer Program Flowchart 63

6.2. Frame Grab Program Flowchart 65

7.1. Inverse Responsivity Calculation Program Flowchart 68

7.2. Data Plotting Program Flowchart 72

8.1. System Setup for Calibration 74

A.1. Sample Gray Scale Plot 87

LIST OF TABLES

TABLE		PAGE
2.1.	Time Code Unit Addressing	10
2.2.	MCP Gain State Encoding	15
4.1.	Transmitted Data for each Field in Original Design	32
4.2.	DDTI Data Transmission Order	39
6.1.	PC-AT Serial Interface	60
7.1.	Sample Peak Video Decoding	70
7.2.	Sample Duty Cycle Decoding	70

CHAPTER 1

INTRODUCTION

The application areas of intensified silicon array cameras require that the camera systems be capable of obtaining data over a wide dynamic range. Low light level and ultraviolet (UV) sensitive cameras utilizing an intensified silicon array are available to convert normally invisible (to the human eye) radiation into standard television images; and autoranging camera systems are available which provide the information required to measure absolute radiance by generating an instantaneous system gain indication along with the video signal. Whenever the radiation level of the object on which the camera is focused changes very rapidly from a low magnitude to a high magnitude, or vice versa, some mechanism is necessary to keep the video level output by the camera at some nominal output level. This is accomplished by the autoranging controller [1]*. Whenever low levels of radiation are being observed, the controller causes the camera to operate in a high gain state. That is, the sensitivity of the camera is increased. On the other hand, if high levels of radiation are being observed, the camera is operated in a low gain state in which the sensitivity of the camera is decreased.

---

*Numbers in brackets refer to similarly numbered references in the List of References.

Previous data acquisition systems for autoranging intensified cameras recorded the video signal on video tape and an analog indication of the system gain on FM tape. Consequently, quantitative radiometric data reduction included a slow and tedious manual process for determining the system gain. Since the system gain could never be determined for a precise frame, but only under steady state conditions where the gain remained stable, transient data were lost. Although normalized contour plots could be produced, there was no means of processing the data on a pixel-by-pixel basis. Also, previous methods provided no automatic scheme for temporally correlating data taken simultaneously from multiple radiometric systems, but instead relied on time insertion into the video which had resolution only to the nearest second.

The objective of this work was to eliminate these problems by replacing the laborious process of determining the system gain for an individual frame with an automated data processing system under microcomputer control. Data reduction times are decreased and the analysis of transient data is made possible by precise frame positioning and gain determination. The capabilities of the autoranging controller are fully realized by recording the camera gain at the time of each frame of video, thus enabling the eventual reduction of each frame of data to absolute radiometric units. Also, correlation of the video in time with data taken from other systems is made possible by

recording of the actual time of each frame. This thesis describes a method of recording the system gain and the time for each frame, then recovering it when played back from tape. The thesis also describes the system and software developed for digitizing the camera video data frame-by-frame on a pixel-by-pixel basis and reducing the digitized data to engineering units.

The functions of the image processing system can be divided into four areas: generation of the original data, recording of the data to tape, data acquisition by the computer, and data reduction. Much effort was employed to utilize existing hardware, either commercially available or AEDC built, to minimize design time and reduce costs. The work presented in this thesis consists of the integration of the pre-existing units and the development of a digital data tape interface (DDTI) to modulate the digital gain and time data as an audio signal, which is recorded along with the camera video on a video recorder. An overview of the system is presented in Figure 1.1. A description of the overall data generation and recording system, consisting of a time code translator, video recorder, intensified silicon array camera, autoranging camera controller, and digital data tape interface, is presented in Chapter 2. In Chapter 3, the video recorder, digital data tape interface, and computer interfaces are identified with regard to their data acquisition functions. The approach used by the digital data tape interface (DDTI) to record the system gain and time data on tape along with the video is described in detail in

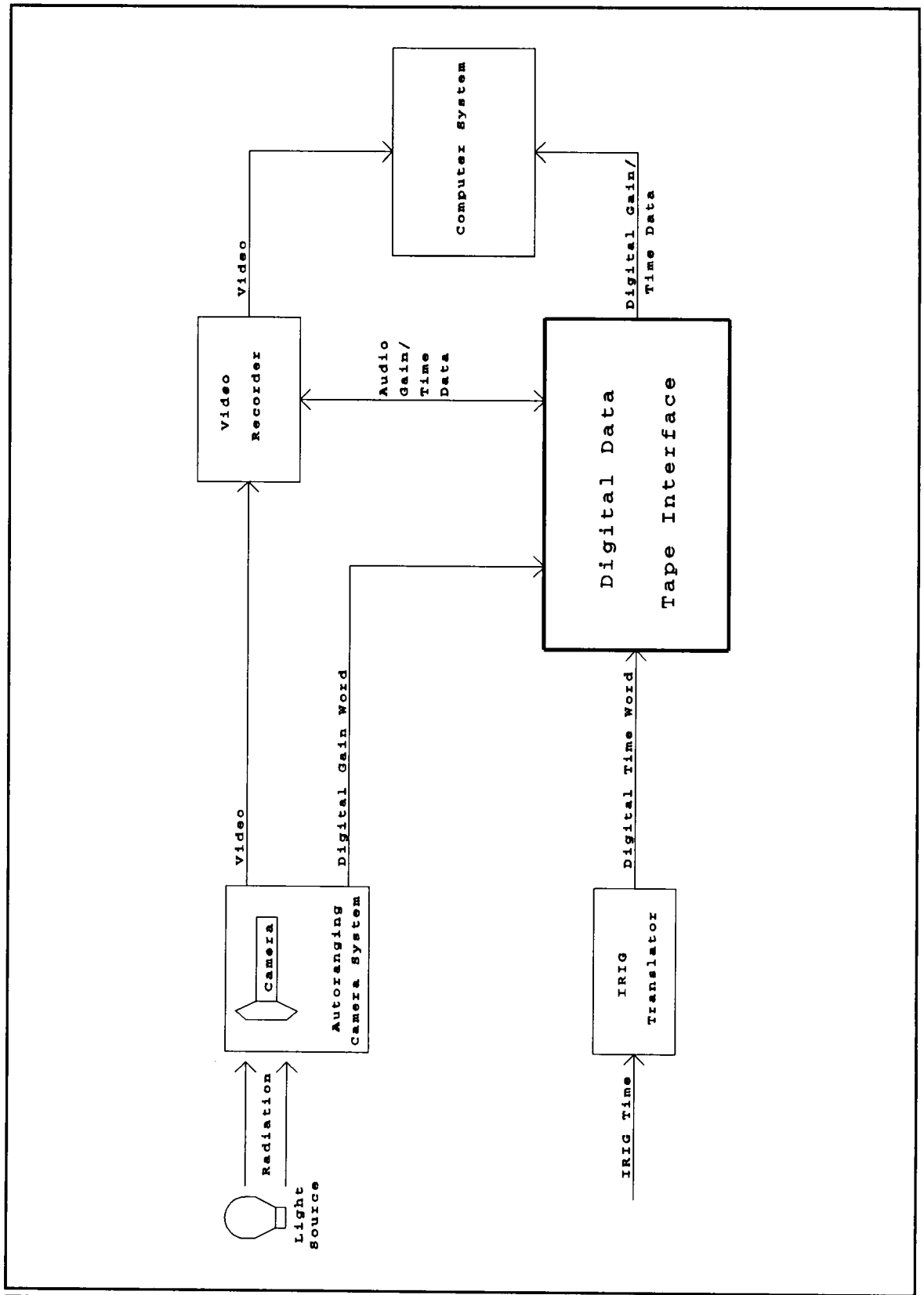


Figure 1.1. Image Processing System Overview

Chapter 4. Chapter 5 discusses in detail demodulation and decoding of the data when it is played back from tape. Chapters 6 and 7 contain a description of the data acquisition and reduction software, respectively; Chapter 8 gives a description of the calibration procedure; Chapter 9 delineates the results of this development; and, finally, Chapter 10 supplies a summary of results along with recommendations.

CHAPTER 2

DATA RECORDING SYSTEM

Shown in Figure 2.1 is a block diagram of the components used to record the system gain and time for each frame. An IRIG translator converts an analog time code signal into digital data. The autoranging camera system performs autoranging of the camera gain to insure an on-scale video signal; the camera gain is output in a digital format. The digital data tape interface (DDTI) modulates the digital gain and time data as an audio signal, which is recorded along with the camera video on a video recorder. Each of the individual components is described in greater detail in subsequent sections.

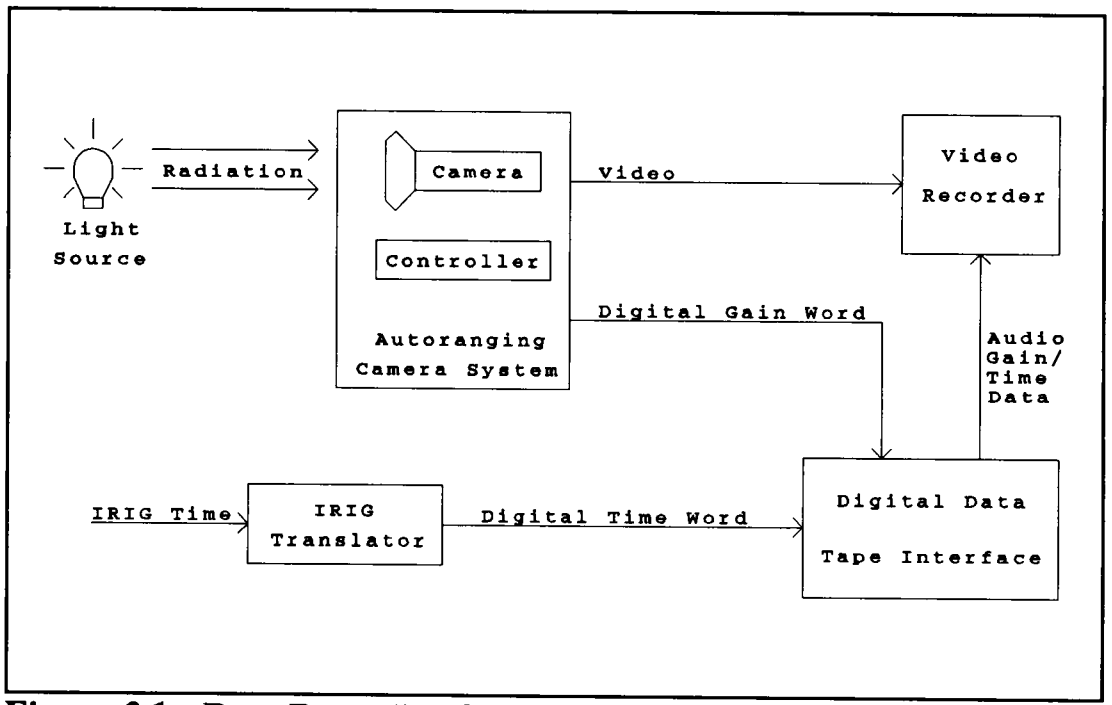


Figure 2.1. Data Recording System

Time Code Translator

A timing control unit from Kode, Inc., is used to decode an IRIG standard serial time code [2]. This interface provides IRIG time in a digital format consisting of eight eight-bit bytes on a parallel output interface. A diagram of this unit is shown in Figure 2.2. The unit can be operated in either translate mode, where a standard analog time code is input to the unit and translated to a digital format, or in generate mode, where in addition to supplying time in a digital format on the parallel outputs, a standard time code analog output is generated. When in generate mode, the time can be set manually from the front panel of the unit. For the purposes of the data recording system presented here, it is irrelevant which mode the time unit is operated in, since the parallel interface remains functionally the same in both modes.

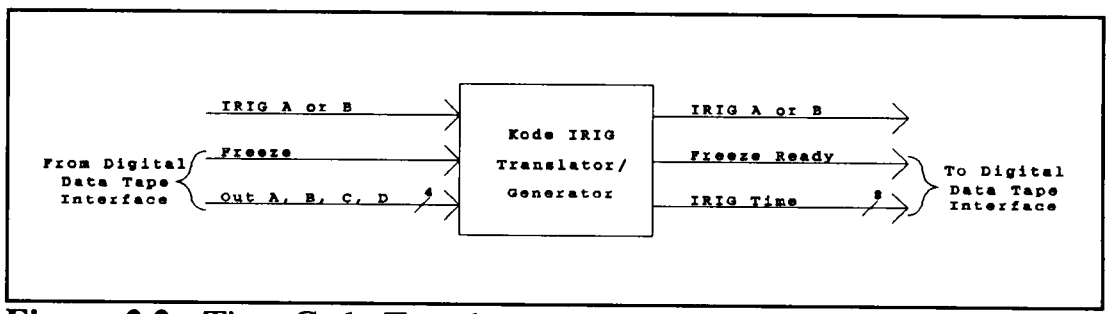


Figure 2.2. Time Code Translator

A timing diagram showing the time unit control logic is shown in Figure 2.3. The Data Ready line will be low whenever the time registers are being updated. The time between updates, t_1, is once per carrier cycle if the unit is in translate mode, or once per 100 microseconds if the unit is in generate mode. The carrier cycle varies depending on the input time code format used, in this case either IRIG A or IRIG B. The maximum time to perform an update, t_2, is eight microseconds. The digital time data is valid and can be read whenever Data Ready is high.

The update of the digital time registers can be disabled by taking the Freeze line high. The Freeze line high time, t_3, must be at least 0.5 microseconds. The time unit will acknowledge that the freeze has been performed by taking the Freeze Ready line high within eight microseconds (t_4) of the rising edge of the Freeze line. The digital outputs will not be updated while the Freeze Ready line is high, but after the Freeze Ready line goes back low (within 0.25 microseconds, t_5, of the falling edge of the Freeze line) the time data will resume being updated.

The OutA, OutB, OutC, and OutD inputs are used to address the digital time data. Although it can be addressed in four-bit units, the DDTI described later always keeps OutA at a logic zero and uses the other three interface lines to address the time data in eight-bit binary-coded decimal (BCD) bytes consisting of two BCD digits each. The addressing scheme is

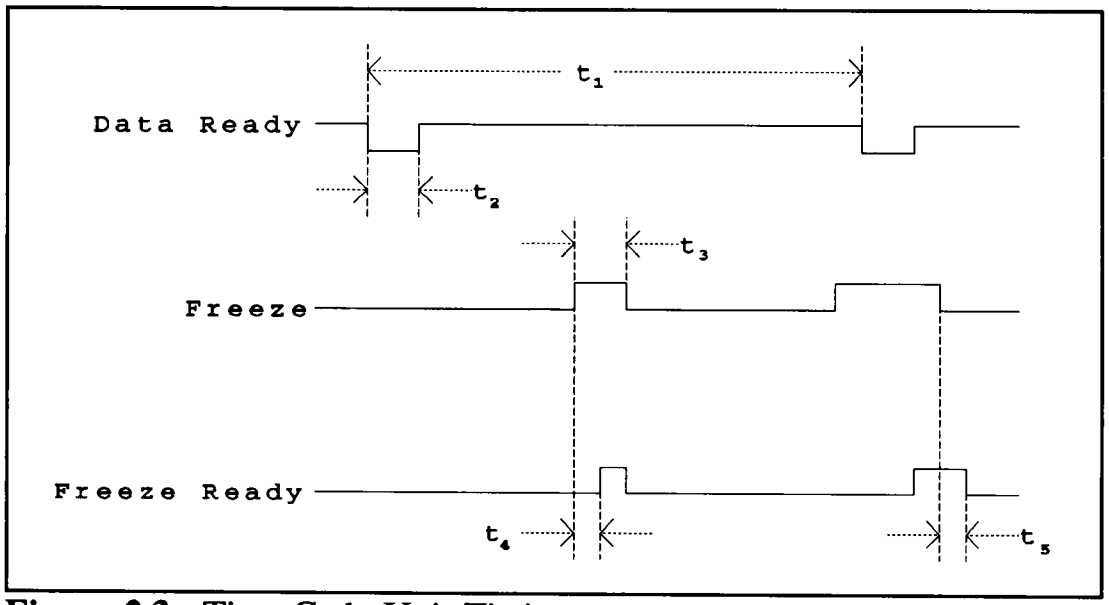


Figure 2.3. Time Code Unit Timing

shown in Table 2.1, where MS represents milliseconds, S represents seconds, M represents minutes, H represents hours, D represents days, and the subscript is the power of ten which the binary digit represents. The bits marked with a dash (--) are not used. The timing and addressing relationship with the DDTI is described in more detail in Chapter 4.

Video Recorder in Record Mode

An extensive product review was conducted prior to the selection of a video cassette recorder (VCR) for the image processing system. In order to remain compatible with tape reproduction capabilities readily available at the AEDC, and to ensure the highest possible quality at the most reasonable cost, it was determined that a 3/4 inch U format umatic machine should be used.

Table 2.1. Time Code Unit Addressing

OutA	0	0	0	0	0	0	0	0
OutB	0	1	0	1	0	1	0	1
OutC	0	0	1	1	0	0	1	1
OutD	0	0	0	0	1	1	1	1
Bit 0	MS_1	MS_{100}	S_1	M_1	H_1	--	D_1	D_{100}
Bit 1	MS_2	MS_{200}	S_2	M_2	H_2	--	D_2	D_{200}
Bit 2	MS_4	MS_{400}	S_4	M_4	H_4	--	D_4	0
Bit 3	MS_8	MS_{800}	S_8	M_8	H_8	--	D_8	0
Bit 4	MS_{10}	--	S_{10}	M_{10}	H_{10}	--	D_{10}	--
Bit 5	MS_{20}	--	S_{20}	M_{20}	H_{20}	--	D_{20}	--
Bit 6	MS_{40}	--	S_{40}	M_{40}	0	--	D_{40}	--
Bit 7	MS_{80}	--	0	0	0	--	D_{80}	--

Another requirement was an automatic gain control (AGC) disable capability. Most VCRs have built in video AGC circuitry to dynamically adjust the gain applied to the video signal before being recorded. Since this is performed directly by the autoranging controller, it is necessary to disable the AGC in order to keep track of true system gain and eventually quantify the data. Another requirement, imposed by the amount of data to be recorded and the modulating scheme used (see Chapter 4), was that the recorder have two or more audio channels. The final requirement was that the recorder have a Society of Motion Picture and Television Engineers (SMPTE) standard forty-five pin parallel control interface. This constraint was imposed by the VCR interface available for use in the computer, described in Chapter 3.

After evaluating procurable umatic recorders against the performance stipulations just given, the JVC CR-600U [3,4] was chosen for use with this imaging system. A block diagram showing the VCR inputs used for data recording is shown in Figure 2.4. The RS-170 video signal from the autoranging controller is a buffered version of the video signal output by the camera. The gain and time data are the modulated digital data provided by the DDTI along with the three control lines. While the parallel remote control interface on the VCR provides access to every function available on the front panel of the machine, the only functions needed by the imaging system to record the data are Record, Play, and Stop.

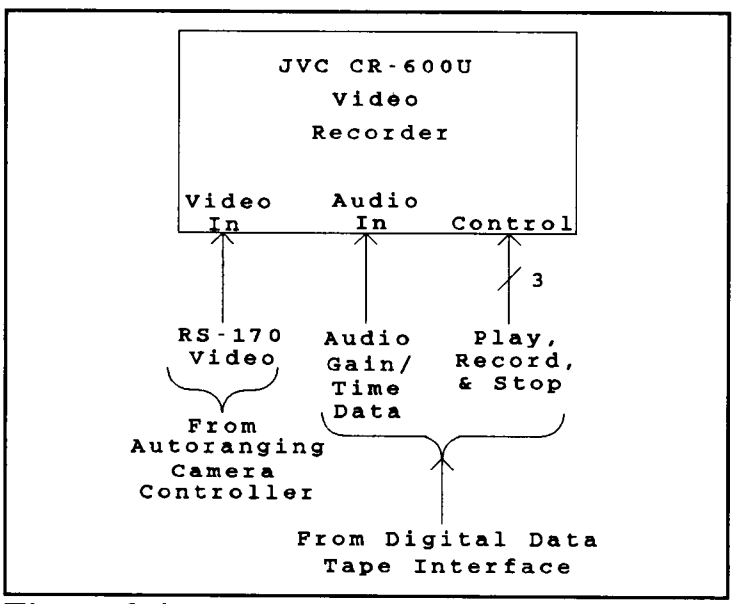


Figure 2.4. Video Recorder in Record Mode

Intensified Silicon Array Camera

The image intensifier and charge coupled device (CCD) array camera used during the development of this imaging system were acquired from ITT [5]. The CCD camera is a CID Technologies Inc. model TN2507A [6]; the type F4111 image intensifier was manufactured by ITT and integrated with the CCD camera to produce an intensified CCD camera system. Input radiation incident on the photocathode is converted into electrons which are multiplied by a microchannel plate (MCP) [7] before being accelerated by a strong electrostatic field and striking a phosphor screen. The resulting visible image is transmitted via a fiber optic bundle to the CCD array. This array consists of a 488 x 388 array of discrete detectors which are electronically scanned to produce the RS-170 video output signal.

The camera gain is adjusted by two different methods. The first is a coarse adjustment obtained by varying the high voltage applied to the MCP. The high voltage level is controlled by a low voltage control signal. This control voltage, which can be adjusted between zero and ten volts, controls the MCP electron gain before the electrons strike the phosphor screen. The second means of gain control is implemented by adjusting the photocathode-on time. This adjustment is realized by controlling the duty cycle of the photocathode gate. This provides a fine adjustment of the phosphor screen

image intensity, and is implemented with a five volt positive logic signal whose duty cycle determines the on time of the photocathode. A block diagram showing the camera control lines is shown in Figure 2.5. As indicated, horizontal and vertical sync outputs are provided in addition to the composite video.

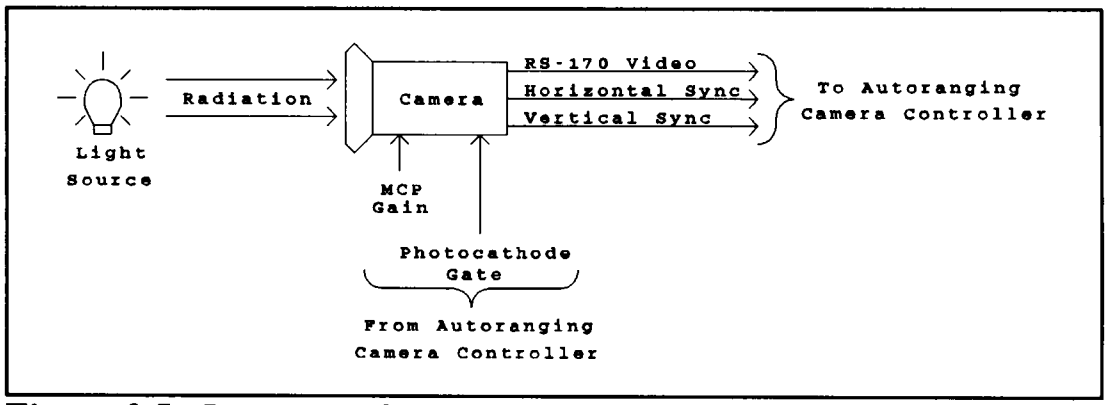


Figure 2.5. Intensified Silicon Array Camera

Autoranging Camera Controller

The autoranging camera controller adjusts the MCP gain voltage and photocathode-on time inputs to the intensified silicon array camera in response to the signal strength of the video signal from the camera. A block diagram of the autoranging controller is shown in Figure 2.6. The MCP gain control voltage and photocathode gate outputs are changed during the vertical retrace interval to ensure that the upcoming field of data will have constant gain parameters for each horizontal scan line. In attempting to maintain a

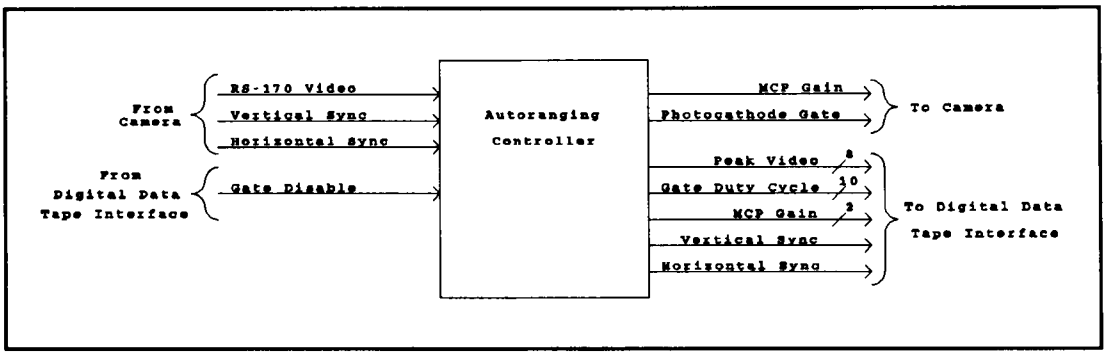


Figure 2.6. Autoranging Camera Controller

one volt peak-to-peak video level, the controller first varies the photocathode-on time. This on time, which is indicated by the photocathode gate output, is increased if the video level falls below one volt and decreased if the video level rises above one volt.

Upper and lower thresholds have been set on the photocathode gate duty cycle variation to ensure linear operation of the camera. Once the gate duty cycle is increased to the upper threshold, the duty cycle is reset to the lower threshold and the MCP gain control voltage increased. Likewise, after the duty cycle has been lowered to the lower threshold, it is reset to the upper threshold and the MCP gain control voltage decreased.

The MCP gain control voltage provides three coarse levels of adjustment. These three settings, corresponding to a different voltage level for high, medium, and low gain ranges, are chosen in a manner such that continuous system gain control is realized as the photocathode duty cycle varies between the lower and upper thresholds. The MCP gain state is also

provided as a two-bit digital output to the DDTI. The encoding for this digital word is given in Table 2.2.

Table 2.2. MCP Gain State Encoding

Bit 1	Bit 0	Gain State
0	0	Medium
0	1	High
1	0	Not Defined
1	1	Low

The autoranging camera controller buffers the horizontal and vertical sync signals and provides them as outputs. Additionally, the peak video is provided as an eight-bit offset binary output; and the gate duty cycle is provided as ten-bit offset binary output. It is the ten-bit gate duty cycle and two-bit MCP gain that are collectively referred to as the "gain word."

The Photocathode Gate line generated by the autoranging controller is wire ANDed with the Gate Disable input from the DDTI (both lines are driven by open collector outputs) to form the Photocathode Gate output to the camera. The Gate Disable line is used for photocathode on/off control by the DDTI. The purpose of this function is described in Chapter 4.

The autoranging controller also provides a manual mode of operation in which the autoranging functions are disabled. This mode is useful for calibration (described in Chapter 8) since it allows the MCP gain control

voltage and photocathode gate duty cycle to be set independently of the magnitude of the camera video.

Digital Data Tape Interface in Record Mode

When operated in the record mode, the DDTI modulates gain data from the autoranging camera controller and time data from the time code translator and sends it in the form of an audio signal to the VCR. A block diagram of this interface is shown in Figure 2.7.

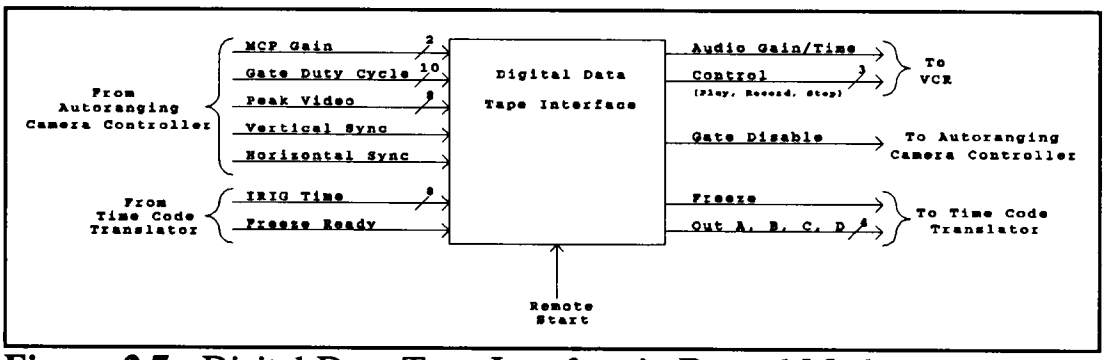


Figure 2.7. Digital Data Tape Interface in Record Mode

The Freeze line is used to halt the time code update just long enough to read it. Addressing of the eight bytes of time data is performed with OutA, OutB, OutC, and OutD. The MCP gain, gate duty cycle, and peak video data are read just after the rising edge of the vertical sync at the completion of the vertical retrace interval. The MCP gain and gate duty cycle will be the gain parameters for the upcoming field; and the peak video will correspond to the video peak detected during the previous field. The three control lines used

by the DDTI are Play, Record, and Stop. The Remote Start line is used to remotely activate the data recording process. For a detailed description of the DDTI in record mode, see Chapter 4.

CHAPTER 3

DATA ACQUISITION SYSTEM

A diagram of the intensified camera image processing system in data acquisition mode is shown in Figure 3.1. This is the operational mode in effect after experimental radiometric data has been recorded on the video tape. The digital data tape interface (DDTI) demodulates the gain and time data as it is played back from the audio channels of the VCR, forming a serial bit stream which is downloaded to the computer via an RS-232C interface, then stored on a hard drive. The RS-170 video signal from the video channel on the VCR is digitized with a frame grabber and also stored on the hard drive. Image processing software then combines the digitized video image with the gain data and reduces the adjusted video to absolute radiometric units. The reduced data can then be plotted on a video monitor or hard copy device as iso-radiance contours or gray scale plots.

Video Recorder in Playback Mode

The VCR outputs and control inputs for the playback mode of operation of the image processing system are diagramed in Figure 3.2. There are three steps involved in the playback operation. First, the gain and time

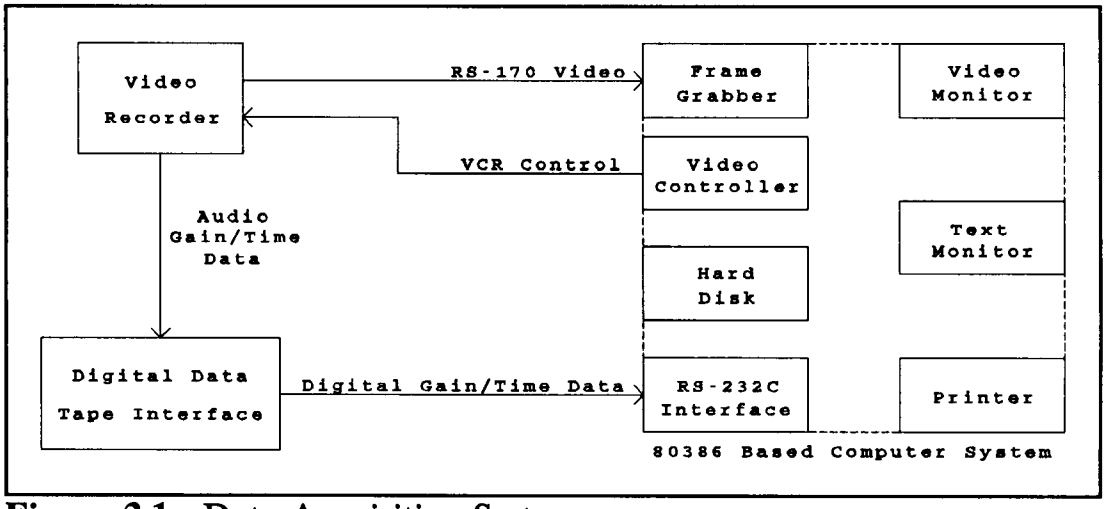


Figure 3.1. Data Acquisition System

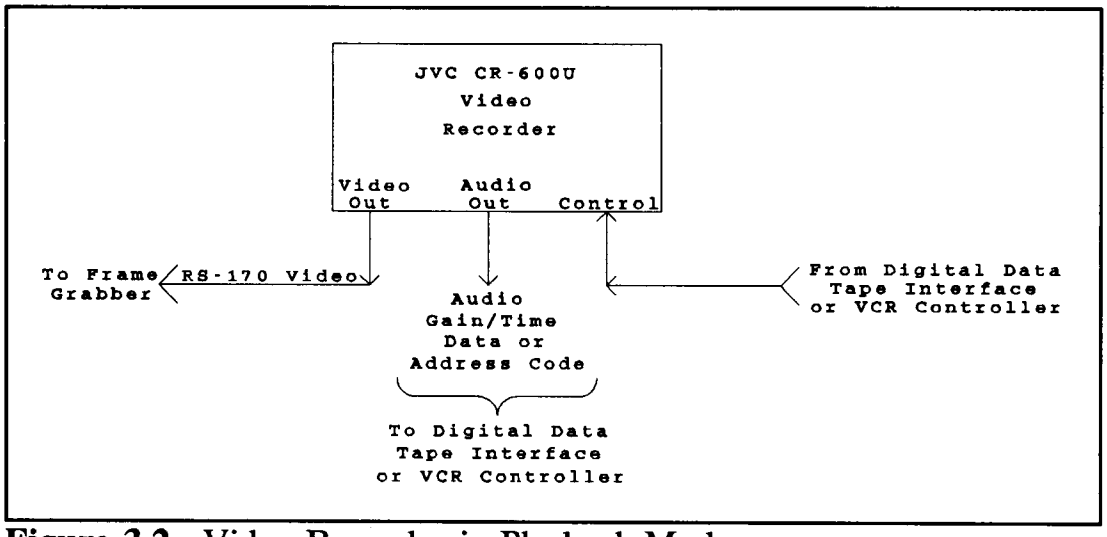


Figure 3.2. Video Recorder in Playback Mode

data are demodulated by the DDTI and downloaded to the computer. The DDTI is not used for the remainder of the playback process. Next, the audio channel two and VCR control cables are rerouted to the VCR controller, and tape address code is written to audio channel two using the VCR controller. Finally, the tape is replayed and selected video frames are digitized with the frame grabber. A total of four tape passes are thus required to process the data--one pass to obtain the gain data on audio channel one, a second pass to secure the gain data on audio channel two, a third pass to write the tape address code, and finally a pass to get the desired video data. The VCR controller and frame grabber are described in more detail later.

Digital Data Tape Interface in Playback Mode

When operated in playback mode, the DDTI receives the modulated gain and time data from the audio channels of the VCR, demodulates it to obtain the original digital data, converts the digital data to a serial bit stream and inserts start and stop bits, and then downloads it to the computer. For a detailed account of the operation of this interface in playback mode, see Chapter 5.

VCR Controller

In order to control the VCR from the computer via software, a computer to VCR interface was needed. This interface has been provided by the BCD-4000 Professional Video Controller [8] purchased from BCD Associates, Inc., which consists of a CPU board with an eight-bit PC bus interface and a breakout box. This interface provides precision control of the VCR, so that the frame grabber can be triggered at precise video frames.

A block diagram of the VCR controller is shown in Figure 3.3. The VCR control cable from the breakout box connects to the forty-five pin remote control connector on the VCR. Control code for all the VCR control functions (Stop, Play, Record, Fast Forward, etc.) is stored in a programmable read only memory (PROM) integrated circuit (IC) on the BCD-4000. Exact positioning of the video tape is made possible by BCD's Absolute Addressing frame number code and tape identification number, which are recorded on audio channel two of the VCR at every frame. This code, consisting of a twenty-four bit frame count and a sixteen bit tape ID, numbers each video frame on the tape, enabling random access of individual frames. Since audio channel two is also used by the DDTI, the gain and time data must be downloaded to the computer prior to writing address code to the tape.

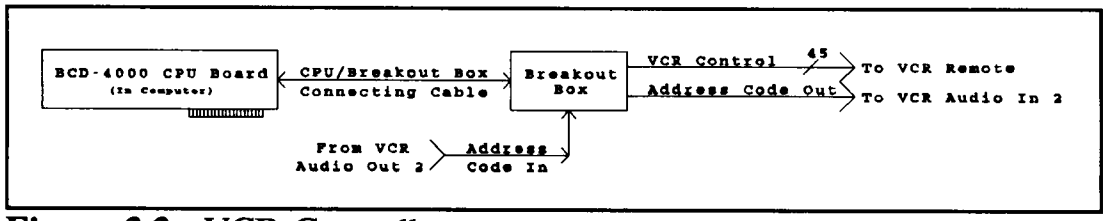


Figure 3.3. VCR Controller

The switch selectable base address of the BCD-4000 was set to 0x282.* Four of the I/O addresses in the PC are occupied (0x282, 0x283, 0x284, and 0x285), but only the upper two are used. These are the bidirectional data port, PD, and the status port, PS. All commands to and data from the BCD-4000 are sent one byte at a time through PD, with PS providing handshaking functions. The commands, consisting of four ASCII characters each, are integrated into the data acquisition software described in Chapter 6.

Frame Grabber

The frame grabber chosen for this application was the MVP-AT real time image processor board set [9,10] from Matrox Electronic Systems Limited, shown in Figure 3.4. These three boards occupy two sixteen-bit slots in the computer, and are configured so that they are addressed at I/O locations 0x300 through 0x31F. The IMAGER-AT software package [11], also

*Numbers preceded by 0x represent hexadecimal values.

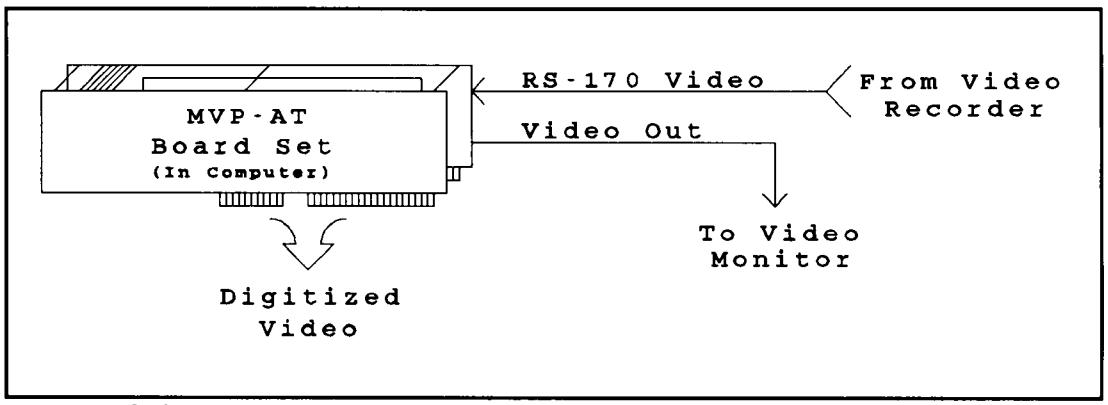


Figure 3.4. Frame Grabber

from Matrox Electronic Systems, provides a library of C callable imaging, graphics, and I/O commands that are used by the data acquisition software.

An interlaced frame of the input RS-170 video signal is digitized by an eight-bit flash analog-to-digital converter on board the MVP-AT, providing 256 discrete intensity levels per digitized pixel in real time with a resolution of 512 pixels by 480 lines. This digitized video image, which is displayed on the video monitor, is read by the acquisition software and stored on a hard drive in the computer. Real time averaging of up to 256 frames is provided by an arithmetic logic unit positioned between the A/D and the MVP-AT video buffers.

Computer System

The computer system used for this image processing system is the 80386 microprocessor based PC-AT compatible System 310 from Dell Computer

Corporation. This system was chosen because of the wide variety of readily available add in cards and software utilities. Even though the bandwidth requirement imposed by the serial data from the DDTI is only 9600 bits per second, the processing time required by the image processing software mandated that a fast thirty-two bit system with high capacity hard drive be used. See Figure 3.1 on page 19 for a diagram of the computer system.

The machine acquired for this imaging system was configured with an 80387 numeric coprocessor, four megabytes of RAM, one 1.2 megabyte 5¼ inch floppy disk drive, a 150 megabyte hard drive, a 150 megabyte tape backup unit, and a VGA color graphics adapter. A parallel port provides the interface to one of two hardcopy devices: an HP LaserJet laser printer for high resolution monochrome output, and a Xerox 4020 color ink jet printer for color plots. Two serial ports provide a connection to the DDTI. The add-in boards are mapped to I/O ports, with the frame grabber occupying addresses 0x300 through 0x31F, the VCR controller 0x282 through 0x285, and the tape backup interface assigned a base address of 0x288. The system runs under version 4.01 of the MS-DOS operating system from Microsoft Corporation.

The choice of compilers for the data acquisition and image processing software was dictated by the software libraries available for use with the hardware. The IMAGER-AT library for the frame grabber mandates a C

interface, hence the Microsoft QuickC compiler [12] was used to develop the data acquisition software. To make use of available graphics libraries and to maintain compatibility with existing AEDC code, the image processing software was developed using the F77L-EM/32 thirty-two bit extended memory FORTRAN compiler [13] from Lahey Computer Systems, Inc., in conjunction with the Graphics Development Toolkit [14] from Graphic Software Systems, Inc. The Pizzaz Plus utility [15] from Application Techniques, Inc., provides hard copy output of video images generated by the image processing software.

CHAPTER 4

DIGITAL DATA TAPE INTERFACE RECORD MODE

As explained in Chapter 1, to eventually reduce the video image from the intensified silicon array camera to absolute radiometric units, the computer processing software must have access to the instantaneous camera gain for each frame of video. Several approaches for providing this data to the computer were considered, including merging the digital data with the video signal, either by inserting it on a horizontal line of video or actually inserting it into the video image itself. This approach was ruled out because it would destroy possibly useful video information. Another approach that was considered was to go straight to the computer with the gain and time data via a digital computer interface, skipping the tape altogether. This approach was ruled out because it would require locating the computer with the camera and the autoranging camera controller whenever video was being recorded. It is preferable to minimize the amount of equipment located at the data collection site, and then download to the computer later. The solution implemented here modulates the digital data as an analog signal and records it on the audio channels of the VCR, so the video image itself is unaltered and the computer is not necessary for the initial data recording process.

Shown in Figure 4.1 is a block diagram of the completed intensified silicon array camera image processing system. At the heart of the system is the digital data tape interface (DDTI), which provides modulation and demodulation of the digital gain information generated by the autoranging camera system, as well as VCR control and serial data transmission to the computer. The digital gain word consists of the two-bit instantaneous MCP gain factor and the ten-bit gate duty cycle. This chapter contains a detailed description of the DDTI in Record mode. The demodulation function for playback of the data from tape is described in Chapter 5. The operating mode, either playback or record, is controlled by a switch on the front panel of the unit.

Original Design

When powered up or reset with the mode switch in the record position, the DDTI first waits on a switch closure on the remote start input to activate the recording process. After the remote start is received the VCR is placed in Record mode, then a five second delay is implemented to give the tape time to get up to speed. After this delay, the camera video is momentarily blanked by turning the intensifier off (using the Gate Disable line) for two

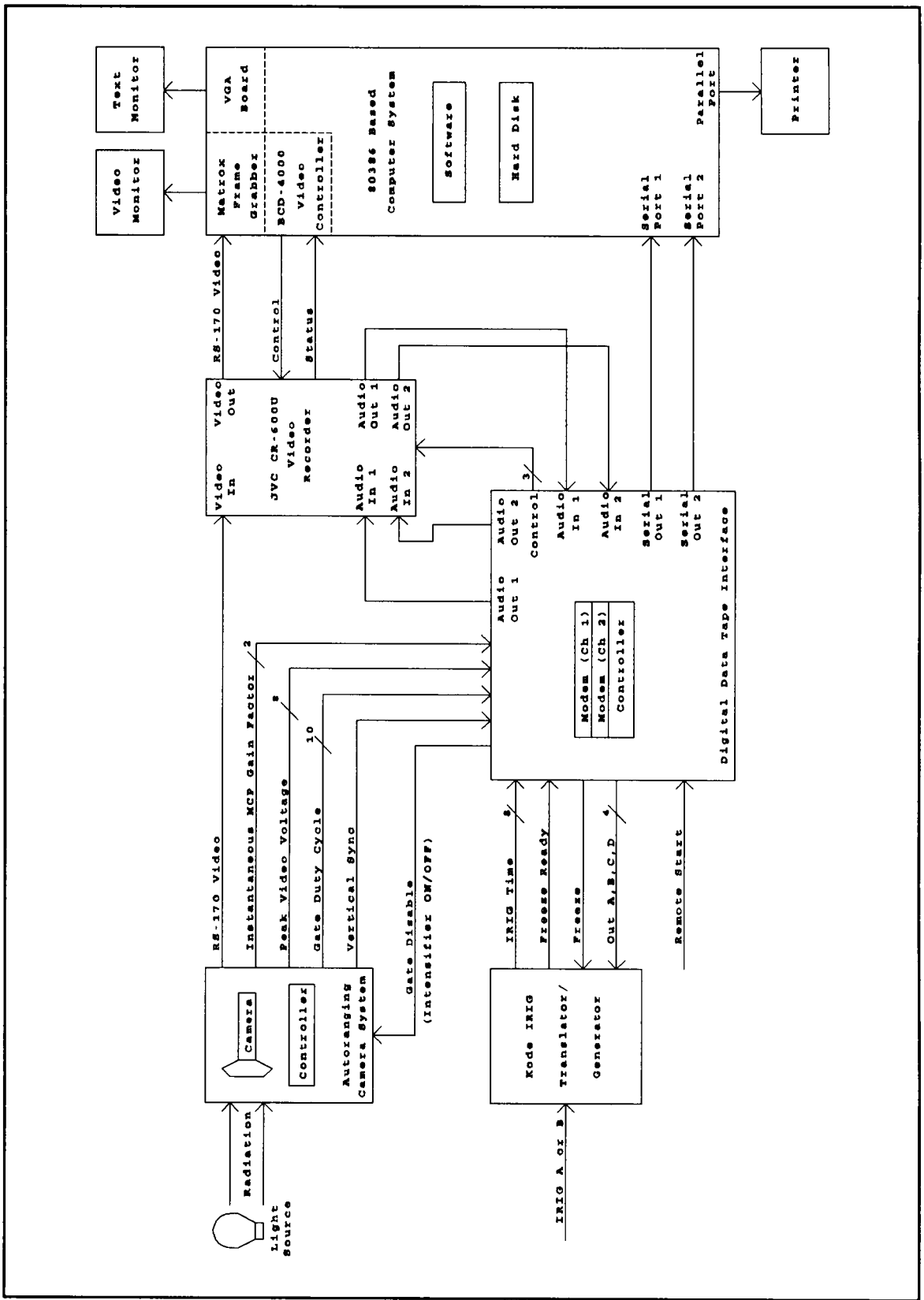


Figure 4.1. Intensified Camera Image Processing System

seconds, then back on. The timing is determined by counting rising edges of the vertical sync input, which occur sixty times a second.

The video blanking is performed so that the audio channel data can be correlated with the corresponding frame of video when the data is played back from tape. During the first seven seconds of recording, digital zeros are modulated and sent to the audio channels of the VCR. Next, two sync bytes are written, then the IRIG translator is polled for the IRIG time, which is modulated and recorded on the audio channels of the VCR. Following the two sync bytes and the eight bytes of IRIG, the MCP gain factor, peak video voltage, and gate duty cycle (for the first frame after the intensifier is turned back on) are read, modulated, and sent to the VCR. From here on, just after every rising edge of the vertical sync, the IRIG time, MCP gain factor, peak video voltage, and gate duty cycle are read, modulated, and recorded. This is repeated for every field of video until an open circuit appears on the remote start input.

The data modulation and demodulation functions are performed by a single-chip modem IC from Yamaha Corporation, the YM7109 [16]. Operating at speeds up to 9600 bits per second, this forty pin dual in-line package (DIP) meets the Consultative Committee for International Telegraph and Telephone (CCITT) standards V.29, V.27ter, V.23 backward channel, V.21, V.21 channel 2, and Bell 103 [17]. It provides modulation and

demodulation, a transmission filter to reject out of band frequencies, adaptive equalization, carrier detection, automatic gain control, a scrambler and descrambler, built-in programmable tone detection and double-tone generation, and long and short training modes. The training modes are a handshaking protocol that determine echo characteristics of the transmission line so that taps in the echo canceler's digital filters can be set. Both the transmission level and reception dynamic range are programmable. Implemented in a 1.2 μm process, this IC typically consumes about 200 mW of power, and requires only a single five volt power supply.

Control of the modem IC is performed through a set of internal registers that set the operating mode, flags, and parameters. These sixteen registers, numbered 0x00 through 0x0F, are set via a parallel interface. Transmit and receive data is transferred via either the parallel interface or a serial interface. The parallel interface consists of sixteen terminals: a bidirectional data bus (D0 through D7), four register select inputs (RS0, RS1, RS2, and RS3), a read strobe input (/RD), a write strobe input (/WR), a chip select input (/CS), and an interrupt request output (/IRQ). An interrupt request can be generated under three different conditions: to request a data transfer during the parallel transfer of transmitter data, to indicate the reception of data during parallel transfer of receiver data, and after the completion of an internal RAM access. Corresponding to each of these cases

is an interrupt enable control bit in the interface registers. Other control bits select the interface to be used (parallel or serial) and the operating mode. Selection of the interface registers is performed with the register select inputs. When a low pulse is sent to /CS and /WR, the contents of the data bus are written to the specified register. When a low pulse is input to /CS and /RD, the specified register contents are read to the data bus. When sending Transmitter data to the modem over the parallel bus, data is written into Register 4 (TxDP) in eight bit units.

Two analog filters are installed external to the modem IC. A third order filter is used to eliminate the higher order harmonics present after the D/A conversion of the audio output signal. Another filter prevents loopback noise caused by the sampling of the A/D during digitization of the audio input signal.

The initial modulation scheme used was sixteen point QAM, or quadrature amplitude modulation, with a 1700 Hz carrier. This modulation method, a combination of amplitude and phase shift techniques, allowed data to be transmitted at 9600 bits per second, or 1200 bytes per second. Since there are sixty fields of video per second, this scheme provided twenty bytes of digital data for each field of video after transmission of the two sync bytes. A description of what each of these bytes represented is shown in Table 4.1. The first two bytes were not used, but reserved for future expansion with a

Table 4.1. Transmitted Data for each Field in Original Design

Byte Number	Description
1	0
2	0
3	Count Byte 0
4	Count Byte 1
5	Count Byte 2
6	Count Byte 3
7	Filter Position
8	IRIG Byte 0
9	IRIG Byte 1
10	IRIG Byte 2
11	IRIG Byte 3
12	IRIG Byte 4
13	IRIG Byte 5
14	IRIG Byte 6
15	IRIG Byte 7
16	Peak Video (Low Byte)
17	Peak Video (High Byte)
18	MCP Gain Factor
19	Gate Duty Cycle (Low Byte)
20	Gate Duty Cycle (High Byte)

zero transmitted for each. The next four bytes formed a thirty-two bit absolute field number that was to be used to sequentially number each field of video. The next byte was reserved for transmitting a filter position on an optical bandpass filter wheel, but since a filter wheel was not implemented in the original version, 0xFF was transmitted here. The next eight bytes transmitted were the eight bytes of IRIG time read just after the rising edge of the vertical sync for this field. Following the IRIG data, two bytes of peak video were transmitted. Since this system provides only eight bits of peak

video, the second byte was provided for future expansion capabilities, and a zero was always transmitted. Next, the two bit MCP gain factor was transmitted, with the high six bits always set to zero. Finally, the gate duty cycle was transmitted, first the low byte, then the two most significant bits. The high six bits of this second duty cycle byte were always zero.

The 9600 bit per second data rate obtained in this mode is strictly 9600 synchronously transmitted data bits with no start or stop bits inserted, so initially there is no way of knowing where one byte stops and another starts. This is the reason for the two sync bytes transmitted prior to the gain and time data at the very beginning of the data transmission process. A sync detector watches the incoming serial bit stream when the data is played back from tape and demodulated, and when the eight bit sync byte is detected, it then knows where the byte boundary is. The original sync byte was 0xFF, or 11111111 binary. This seemed like a logical choice at first, since 0's were being sent to tape a full seven seconds before any real data, but it did not always work. A problem occurred when the tape was rewound and the VCR placed in play mode--the modem would think it saw a string of 1's, sometimes a long enough string to pass as 0xFF. A solution to this problem is discussed in the next section.

Another, much more serious problem, was uncovered in the checkout of the original system. The modem IC did not properly demodulate the data

as it was played back from tape! Three other operating modes were tried (7200 bps eight point QAM, 4800 bps eight point PSK, and 2400 bps four point PSK), but none worked. Finally, out of desperation, the data was recorded on ½ inch tape using a high quality FM tape recorder, with a bandwidth from DC to 40000 Hz. Magically, when it was played back, it was demodulated correctly! This seemed mysterious, since the VCR audio channels had a bandwidth of about twenty kHz and the QAM signal required only about three kHz. Several days of intense troubleshooting followed, and it was eventually determined that the phase modulated modes will not work because of phase changes introduced at the audio record/play heads by the mechanical servo mechanism on the VCR. The reproduced signal undergoes a phase change of about 180 degrees at a frequency of about two Hz (100 false phase jumps in fifty-one seconds). Because of these phase errors, neither the QAM mode nor the PSK mode will operate correctly if a standard VCR is used to record the data. A solution to this problem is discussed in the next section.

Design Modifications

Once it was concluded that the original QAM scheme would not work with the VCR, several alternative approaches were investigated. One idea was that of using the spare audio channel on the VCR to directly record the

IRIG time data. Upon trying this approach, it was found that the IRIG modulation format was not compatible with the VCR, thus prohibiting correct demodulation when the data was played back. The idea of going straight to the computer was investigated again, but was ruled out for the same reasons given previously. The approach ultimately realized implements two modems, splitting the data in half and recording on both audio channels of the VCR using a frequency shift keying (FSK) mode that can be demodulated correctly. Also, since the field rate of the camera is crystal controlled to generate exactly sixty fields per second, only the IRIG time of the first frame must be recorded, and each subsequent frame will be separated by 1/30 second. Another saving grace is that the frame grabber grabs an interlaced frame anyway, so the gain data need only be recorded for each frame instead of for each field as was originally implemented. This by itself cuts the required data throughput rate in half. Every other sync edge is used since each frame of RS-170 video is composed of two fields, each with its own vertical sync pulse.

As for the problem of determining a unique sync byte, 0x69 was chosen. This byte, which is a binary 01101001, has nibbles which are logical complements of each other. It tests the transmission of a single 0 with a 1 on each side, a single 1 with a 0 on each side, two consecutive 0's, and two consecutive 1's. A sync byte with mixed 0's and 1's was chosen because the false data generated when the tape is stopped and started consists of strings

of several 0's or 1's in a row. In the hundreds of cases run with this sync byte, a false sync has never been generated. A block diagram of the completed DDTI in record mode is shown in Figure 4.2.

Data Encoding and Modulation Format

The fastest modulation method implemented by the modem IC that did not depend on a phase shift technique was the 300 bit per second FSK mode. In FSK, spaces (logic 0) and marks (logic 1) are represented by sinusoids of different frequencies. The mode chosen was CCITT V.21 channel 2, with a space frequency of 1850 Hz and a mark frequency of 1650 Hz. Since both audio channels on the VCR are now being used, a total of seventy-five bytes per second of data can be recorded, or 37.5 bytes per second per audio channel. Since there are thirty frames of video per second, this works out to 1.25 bytes (ten bits) per frame per audio channel. For each frame, the ten bit gate duty cycle is recorded on audio channel one, and the eight bit peak video and two bit MCP gain are recorded on audio channel 2.

A different problem arises now since the frame rate of thirty frames per second is not an even multiple of the data transmission rate to each modem of 37.5 bytes per second. This was not a problem with the 9600 bits per second (1200 bytes per second) QAM mode, since sixty fields per second

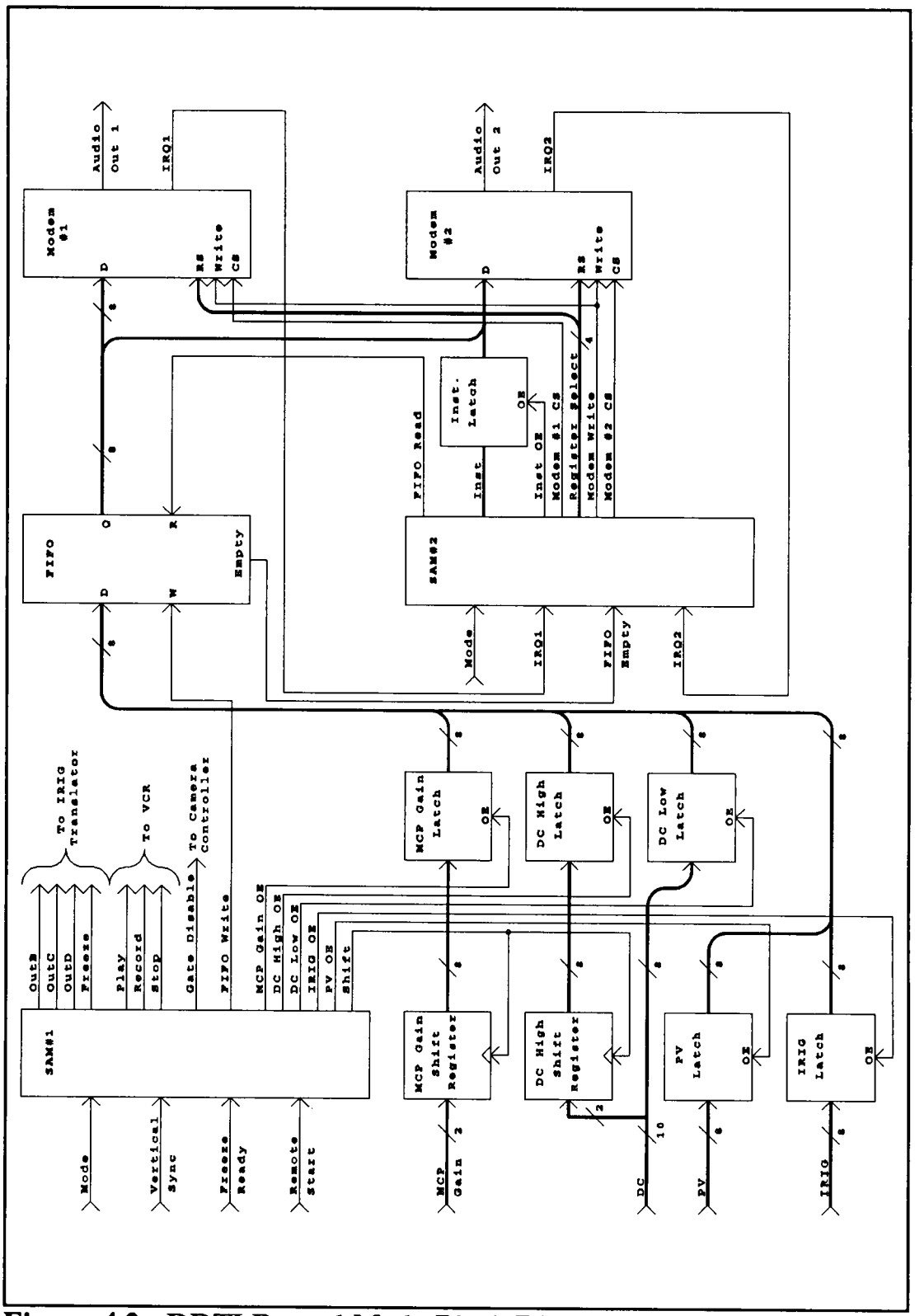


Figure 4.2. DDTI Record Mode Block Diagram

divided evenly to give twenty bytes per field. The problem here can be solved by noticing that an even five bytes is transmitted per audio channel for every four frames of video, so four eight bit bytes can be transmitted at each of four consecutive frames, with the leftover two bits for each frame packed into a fifth byte transmitted every fourth frame. An IDT7202 2048 word first-in first-out (FIFO) [18] dual port memory is inserted between the incoming data and the modem ICs to match the data rates between the camera controller and the modems. Simultaneous reads from and writes to the FIFO can be performed by pulsing the corresponding read or write input low.

The average throughput of data entering and leaving the FIFO is 600 bits per second. The data comes in two bytes per frame (thirty frames per second) with the exception of every fourth frame, when there are four bytes; and it leaves the FIFO at seventy-five bytes per second (one byte every 13.33 milliseconds). The order of the data transmission to tape is shown in Table 4.2. Note that the time data is the IRIG time for the first frame after the image intensifier is turned back on after the two second shutdown, and is transmitted only for this first frame. It is transmitted on both channels so that it can be used as a checkpoint for proper demodulation when the data is played back from tape. The eight bytes of IRIG (read twice for a total of sixteen bytes) plus the duty cycle low byte and peak video byte for the first frame make a total of eighteen bytes clocked into the FIFO during the first

Table 4.2. DDTI Data Transmission Order

Byte Number	Channel 1 Data	Channel 2 Data
0	Sync (0x69)	Sync (0x69)
1	Sync (0x69)	Sync (0x69)
2	IRIG Byte 0	IRIG Byte 0
3	IRIG Byte 1	IRIG Byte 1
4	IRIG Byte 2	IRIG Byte 2
5	IRIG Byte 3	IRIG Byte 3
6	IRIG Byte 4	IRIG Byte 4
7	IRIG Byte 5	IRIG Byte 5
8	IRIG Byte 6	IRIG Byte 6
9	IRIG Byte 7	IRIG Byte 7
10	Frame 1 Duty Cycle Bits 0..7	Frame 1 Peak Video
11	Frame 2 Duty Cycle Bits 0..7	Frame 2 Peak Video
12	Frame 3 Duty Cycle Bits 0..7	Frame 3 Peak Video
13	Frame 4 Duty Cycle Bits 0..7	Frame 4 Peak Video
14	Frames 1-4 DC Bits 8 & 9	Frames 1-4 MCP Gain
15	Frame 5 Duty Cycle Bits 0..7	Frame 5 Peak Video
.	.	.
.	.	.
.	.	.

frame time. For succeeding frames only two bytes are transmitted to the FIFO, except for every fourth frame when four bytes are transmitted.

Record Mode Programmable Logic Devices

To squeeze the two MCP gain bits and the upper two gate duty cycle bits for each group of four frames into two packed bytes, an EP600 erasable programmable logic device (EPLD) [19] from Altera Corporation was used. This device implements over 600 equivalent gates of logic functions, using sum of products (SOP) logic with a programmable AND and fixed OR. Both

combinatorial and sequential logic functions are supported, with the ability to program D, T, SR, or JK flip-flops on each of sixteen macrocells. A high level part definition file is created in the Altera Design File (ADF) format using a text editor, then compiled into an industry standard Joint Electronic Device Engineering Council (JEDEC) fuse map file using the A+ Plus software package [20,21]. The JEDEC file is used by LogicMap software and programming hardware to program the device [22]. The EP600 used by the record mode circuitry (EPLD #1) provides four four-bit shift registers for creating the two eight-bit packed bytes. An equivalent circuit depicting the logic implemented by this device is shown in Figure 4.3.

Record Mode Microsequencers

Control flow and sequencing operations are performed by EPS448 Stand-Alone Microsequencers (SAMs) [19] from Altera Corporation, clocked at 16 MHz. These UV erasable, programmable controllers have 448 words of microcode memory, an eight bit wide fifteen deep stack, a loop counter, eight branch control inputs with multi-way branching, and sixteen control outputs. The SAM+ Plus software package supports assembly language design entry of the microcode [23]. A text file is processed into a JEDEC file, which is used by the LogicMap package to program the device. SAM #1 from Figure

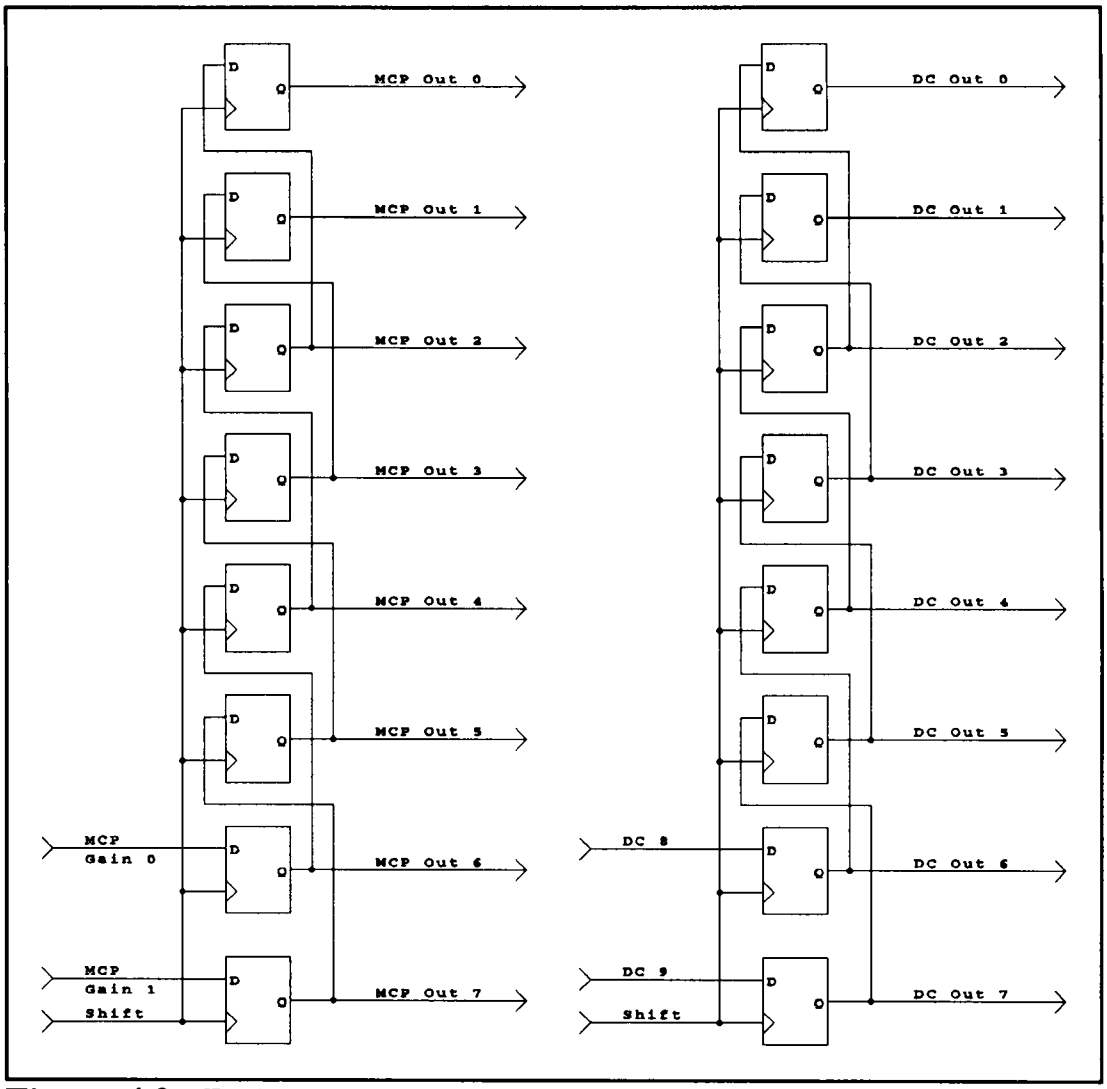


Figure 4.3. EPLD #1 Equivalent Circuit

4.2 controls the VCR, time code translator, EP600 shift registers, and data flow into the FIFO. A flowchart showing the operation of this microsequencer is shown in Figure 4.4.

This sequencer does nothing if the mode switch is set to playback, but if the record mode is set, it enters a loop waiting for a low logic level (switch closure) to appear on the remote start input. A microcoded noise filtering operation is implemented after the apparent reception of a remote start to filter out unwanted noise spikes on the remote start input. The algorithm used implements a sixteen microsecond delay after the first indication of a remote start switch closure and then reads it again. If the switch closure is no longer present, a noise spike has occurred so control is passed back to the beginning of the microprogram. If the switch closure is still present, another sixteen microsecond delay is implemented and the procedure is repeated for a total of three sixteen microsecond delays. The sixteen microsecond delay is implemented by counting 256 clock cycles using the counter on board the SAM. Since the part is being clocked at sixteen MHz, one clock period is 62.5 nanoseconds, so a 256 count loop will expend exactly sixteen microseconds.

After verifying that the remote start has indeed been received, the VCR is placed in record mode by simultaneously taking the Play and Record lines low. There is a five second delay, then the image intensifier is turned off for two seconds, just as in the original design. The five second delay is

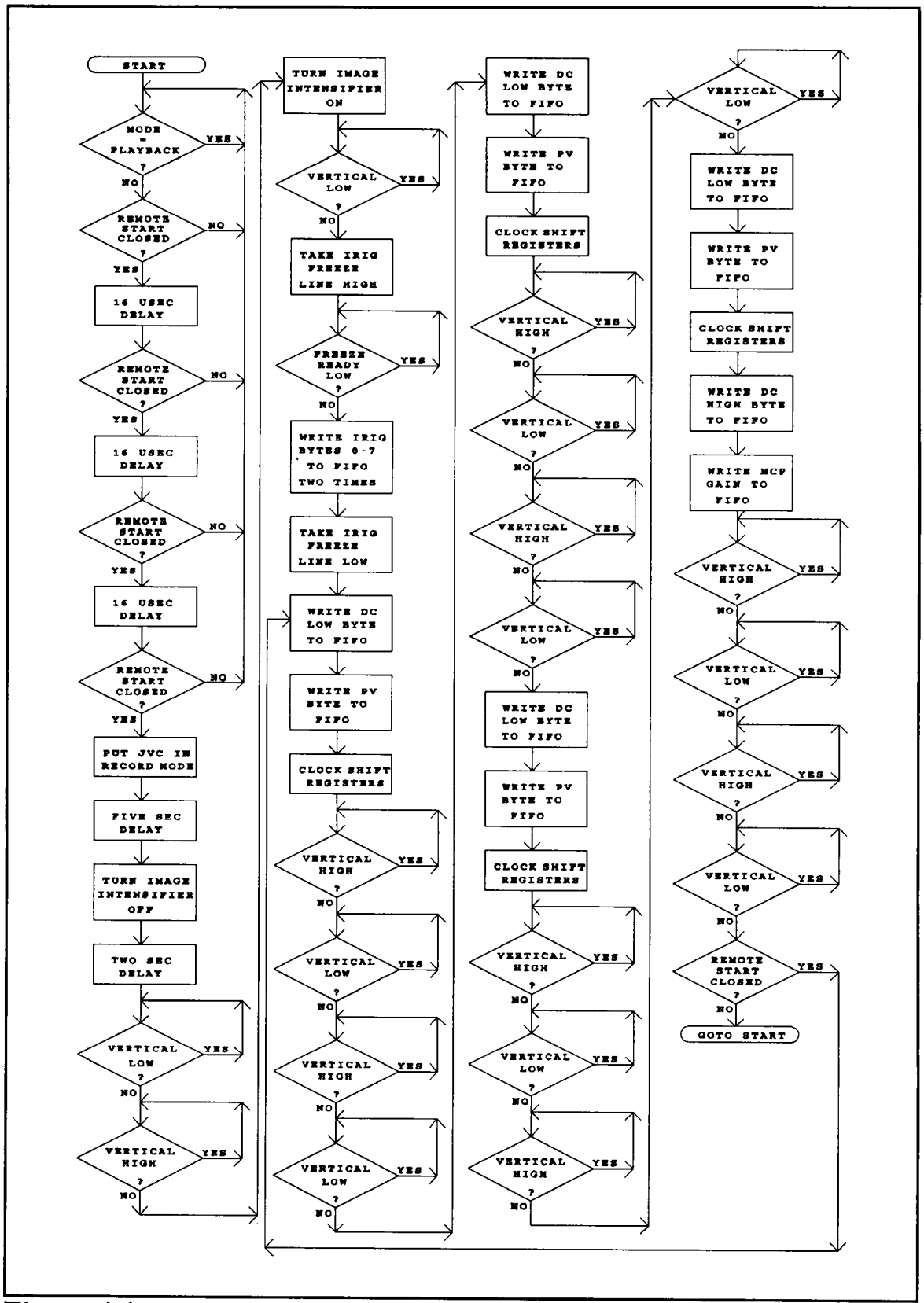


Figure 4.4. SAM #1 Flowchart

implemented by counting vertical sync pulses, which occur exactly sixty times each second. These pulses are counted by watching for a low to high transition, which will always be seen since the sequencer clock period of 62.5 nanoseconds is much smaller than the vertical sync period. Following the two second intensifier off period, and just after the rising edge of the vertical sync, the Freeze input to the IRIG translator is taken high. Since the time freeze is guaranteed to occur within eight microseconds, the frozen time will be accurate to at least the nearest millisecond, which is all the resolution the unit provides. There is a pause by the sequencer until the freeze is acknowledged by the Freeze Ready line going high, then each byte of IRIG is read and written twice to the FIFO by making the data available on the inputs to the FIFO and taking the FIFO Write line low. Since the width of this pulse must be at least eighty nanoseconds, it is held low for two sequencer clock periods to give an actual pulse width of 125 nanoseconds. The Freeze line is kept high until the last byte of IRIG has been read, then it is taken back low, and usage of the time code translator is now complete for the duration of the recording process. See Chapter 2 for a time code unit timing diagram and data addressing table.

After the time code data has been written into the FIFO, the low byte of the gate duty cycle is written, followed by the peak video byte. Next, the EP600 shift registers are clocked. The microprogram now waits on two rising

edges on the vertical sync input, corresponding to two fields or one frame, then it writes the gate duty cycle low byte and peak video data into the FIFO again and clocks the shift registers. This procedure is repeated two more times. After the fourth cycle of writing the gate duty cycle and peak video data to the FIFO and clocking the shift registers, the packed high byte of gate duty cycle data is written to the FIFO followed by the packed MCP gain byte. After two more rising edges on the vertical sync input, the remote start line is checked again. If a switch closure is still present, control loops back and the procedure to write the duty cycle low byte and the peak video byte to the FIFO and clock the shift registers is repeated. If the switch closure is no longer present, control is passed back to the beginning of the microprogram.

The second sequencer depicted in Figure 4.2 on page 37, SAM #2, is actually two SAMs connected in parallel to provide twice as many outputs. This SAM controls the modem ICs and data flow from the FIFO to the modems while in record mode. A flowchart of the operation of this microsequencer in record mode is shown in Figure 4.5. Unlike SAM#1, SAM#2 also functions in the playback mode. The playback mode operation of this sequencer is described in Chapter 5.

Regardless of the operating mode, the sequencer first puts the modems in 300 bit per second FSK mode. Next, assuming the mode switch is in the playback position, the transmitters of each modem are set to parallel data

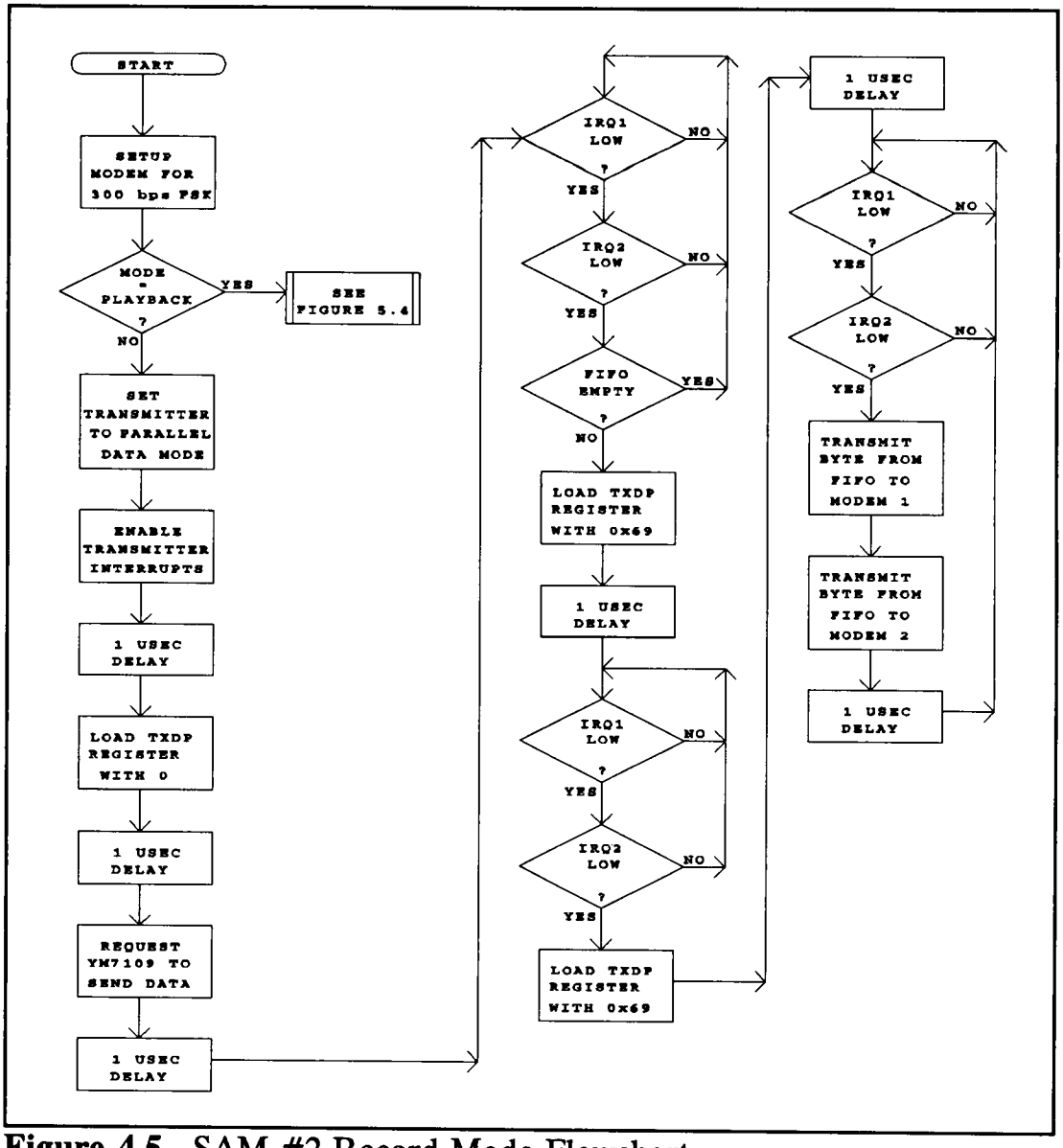


Figure 4.5. SAM #2 Record Mode Flowchart

mode and transmitter interrupts are enabled. To meet the modem IC timing specifications, a one microsecond delay is executed after this and all other accesses to one of the modem registers. This delay is implemented as a subroutine that executes in sixteen clock cycles: one clock to load the loop counter with thirteen, fourteen clocks to extinguish the loop counter, and one clock to return from the subroutine.

Next, a zero is loaded into the TxDP register of both modems, followed by a one microsecond delay, then the modems are sent the command to begin sending parallel data, followed by another one microsecond delay. The interrupt request outputs from both modems will go low as soon as they are ready for another parallel data transfer, which will occur 37.5 times a second if they are fed new data as soon as the IRQ goes low. If new data is not transferred, the modem ICs will continue transferring the old data until a new byte is written to their TxDP registers.

The sequencer allows a continuous stream of 0's to be sent out until both IRQs are low AND the FIFO is no longer empty. The FIFO not empty condition will occur when SAM #1 begins to write the time data into the FIFO. As soon as this happens, SAM #2 writes the first sync byte into both TxDP registers and waits one microsecond. As soon as both modems again request more data, the second sync byte is written, followed by another one microsecond delay. The sequencer again waits on both modems to request

data, then transmits the top byte in the FIFO to modem one. A new top byte will be present in the FIFO, and it is transmitted to modem two, followed by a one microsecond delay. The sequencer now enters an endless loop, waiting for both modems to request new data, writing a byte from the FIFO to modem one, then writing the next byte from the FIFO to modem two. In this way, every other byte is sent to modem one, while the other half of the data is sent to modem two. This is why each byte of the IRIG data was written into the FIFO twice--so it will be sent to both audio channels of the VCR. Also, since SAM #1 writes the duty cycle data immediately following the time data, and then writes the peak video data, channel one will receive the duty cycle data, and channel two will receive the peak video and MCP gain data.

It could be argued that writes to the modem registers not immediately followed by another register write need not be followed by the one microsecond delay. Indeed, a delay was not originally implemented in these cases. However, what resulted was that the sequencer would sense both IRQs going low, write data to TxDP, and be finished and ready to check the IRQs again before the IRQs returned to a high state in response to the initial TxDP write. The sequencer could write several bytes to TxDP before the IRQs ever went back high. Close observation of the IRQ lines on an oscilloscope revealed that if a byte were written to TxDP immediately after the falling edge of the IRQ, the IRQ was still low for about 800 nanoseconds. The

sequencer could execute about twelve instructions during this period. One obvious solution was to implement a delay long enough for the IRQs to go back high before checking their status, and since a one microsecond (1000 nanosecond) delay subroutine had already been implemented, it was used for this purpose.

CHAPTER 5

DIGITAL DATA TAPE INTERFACE PLAYBACK MODE

This chapter contains a detailed description of the digital data tape interface (DDTI) in playback mode. In this mode, a modem IC demodulates the incoming audio data and sends it in the form of a serial bit stream to a sync detector, which monitors the incoming data and waits for the sync bytes (0x69) to be received. The very first set of data after the sync bytes corresponds to the time and gain data for the video frame just after the intensifier was turned back on during the recording process described in Chapter 4. Start and stop bits are inserted by an 8251 Programmable Communication Interface (PCI) [24], enabling the data to be downloaded via a serial link to the computer. A block diagram of the entire intensified camera image processing system can be found in Figure 4.1 on page 28. A block diagram of the playback mode circuitry is shown in Figure 5.1.

Data Demodulation and Decoding

The YM7109 modem ICs demodulate the audio data and produce a serial bit stream which is transmitted to the sync detectors for each channel. The RT output of the modem ICs is the receiver/transmitter data clock, which,

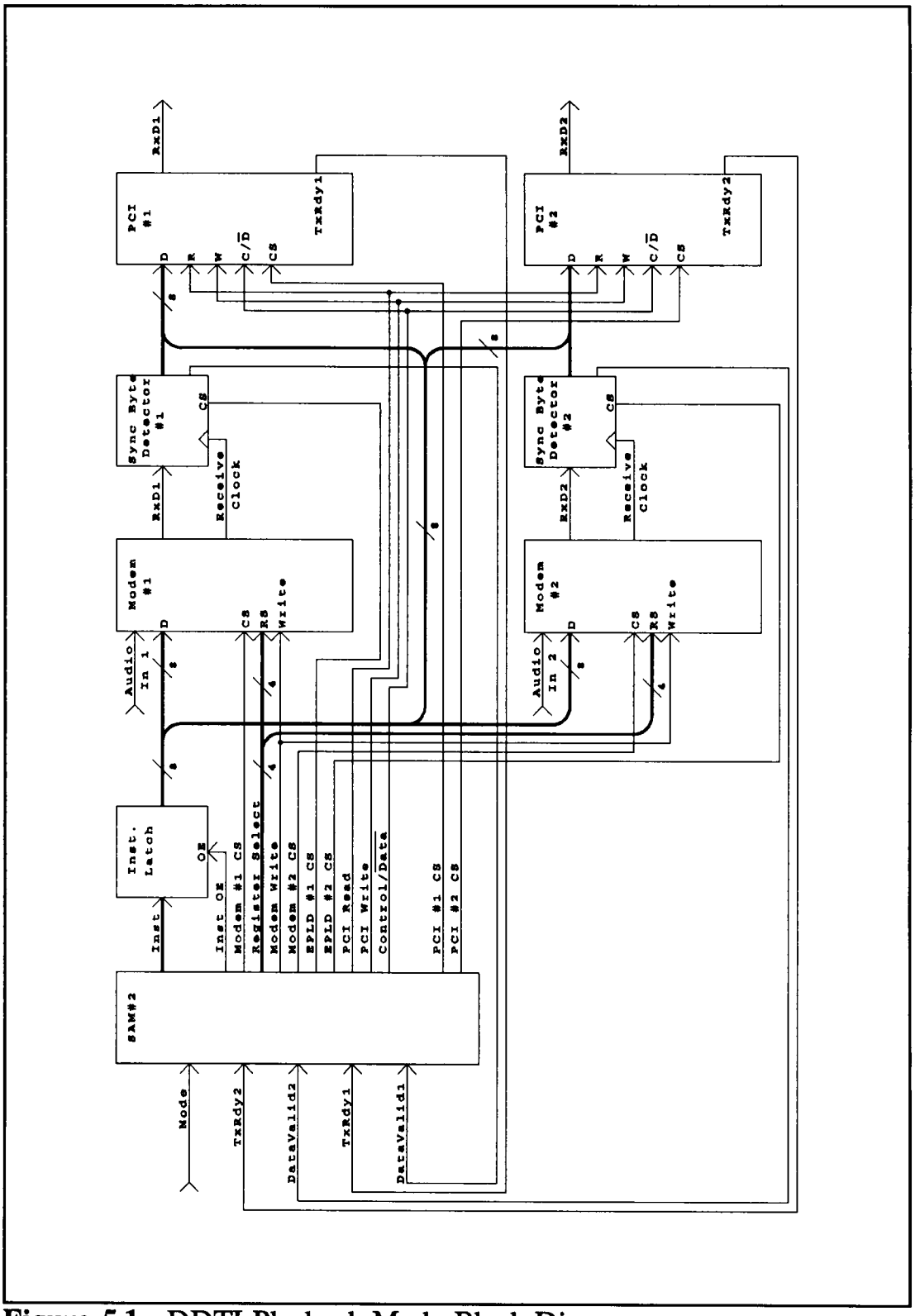


Figure 5.1. DDTI Playback Mode Block Diagram

when operated at 300 bits per second, is a 300 Hz square wave. The demodulated data output on RxD is changed in sync with the falling edge of RT, giving a bit width of 3333 microseconds.

After detection of the sync bytes, the data is transmitted eight bits at a time to the 8251 PCI, where start and stop bits are inserted and it is transmitted out at 9600 bits per second to a computer serial port (COM1 or COM2). As used by the DDTI, the PCI performs parallel-to-serial conversion on eight bit data characters. Communication with this device is enabled only when the chip select line (/CS) is low. The first two words written to the PCI following a reset are the mode and command instructions. Status information can be read when the read line (/RD) is low; and control words or data can be written when the write line (/WR) is low. The control/data line (C/D) is used to indicate if the read/write operation is on data, status information, or a control word. The transmitter ready output (TxRDY) goes high whenever the transmitter is ready to accept a new data character, and is automatically reset whenever a character is written to the device.

Since the computer transfers no data to the DDTI, the serial link is uni-directional and needs only transmit and ground wires. Data is transmitted asynchronously one bit at a time, using one start bit, eight data bits, and one stop bit. There is no parity bit. To translate to RS-232C signal levels, where a space (logic 0) is represented by +12 volts and a mark (logic 1) is

represented by -12 volts, the MAX235 RS-232 driver [25] from Maxim, Inc., is used. The data as received by the computer will consists of the two sync bytes, eight bytes of IRIG indicating the time of the first frame, four regular bytes of data, and a packed byte. The pattern of four regular data bytes followed by a packed byte is then repeated for the remainder of the playback session.

Playback Mode Programmable Logic Devices

Two EP600 EPLDs used by the playback circuitry have each been programmed to provide an eight bit shift register with tri-statable outputs for the demodulated data from the modem IC, a sync byte detection circuit, a DataValid bit, and a two bit divide-by-four counter. One device is used for data from audio channel one, and the other for data from audio channel two. An equivalent circuit of these devices is shown in Figure 5.2.

As each new bit becomes available from the modem, it is clocked into the eight bit shift register. Although the parallel output of the shift register is tri-stated until the channel select line is taken low, the D flip-flop outputs are available for use internal to the EP600. An eight input AND gate monitors these eight bits, watching for the 1-0-0-1-0-1-1-0 bit pattern. This is a backwards 0x69, since the least significant bit will be received first and the

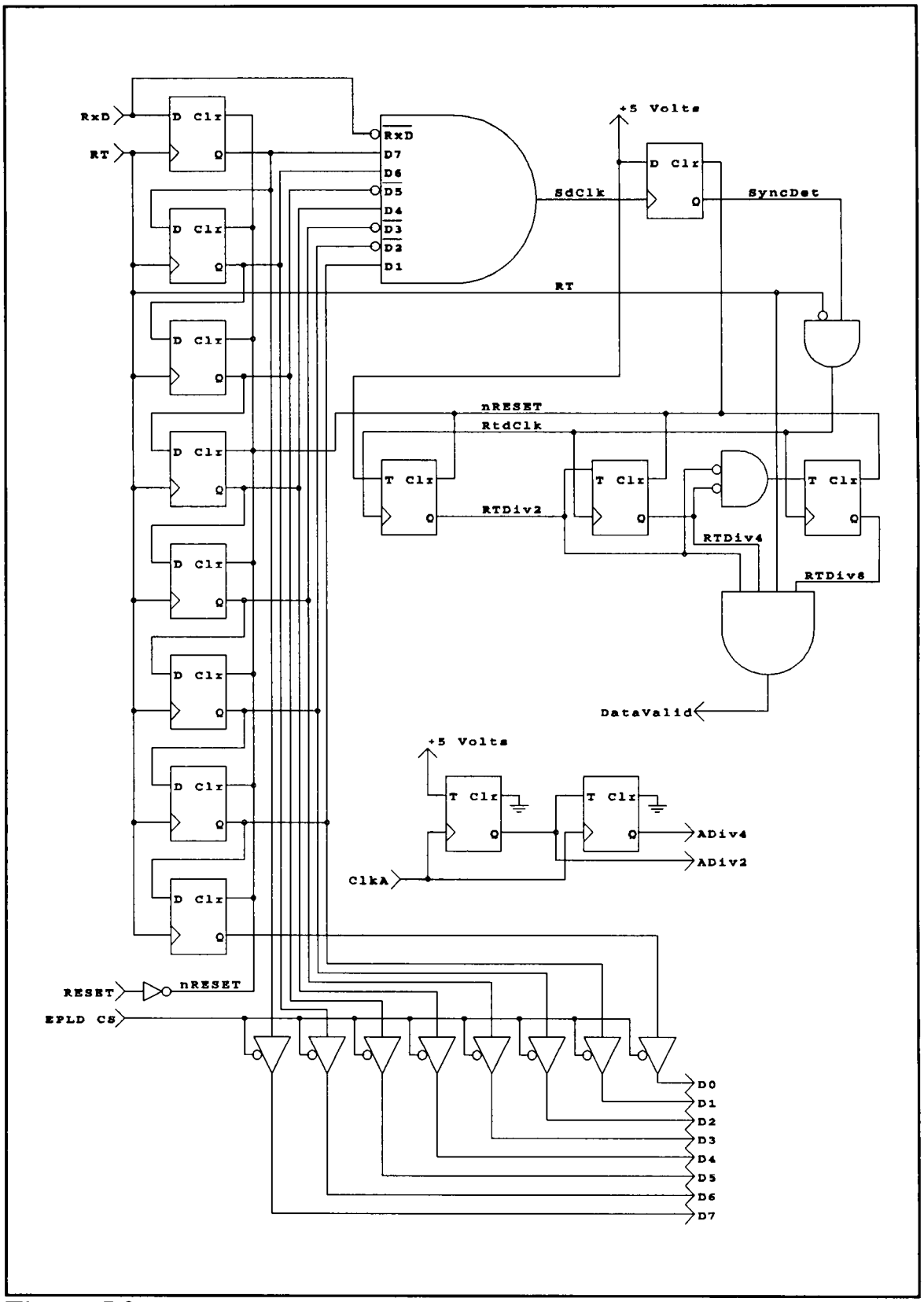


Figure 5.2. EPLD #2 Equivalent Circuit

most significant bit last. The output of the eight input AND will be low until the sync byte is detected, and then it will go high, which will clock a one into a D flip-flop. The output of this D flip-flop in turn is ANDed with the inverted RT clock to form a line called RTDCLK, which is used as the clock input to a bank of three T flip-flops which divide the RT clock by two, four, and eight. These lines, called RTDIV2, RTDIV4, and RTDIV8, respectively, are then ANDed with RT. Since there will be no activity on the divided clocks until the sync byte is detected, this AND will form a DataValid output which goes high for half a bit at the reception of the sync byte, and every eight bits thereafter. The eight bits of demodulated data are valid and can be read whenever the DataValid bit is high. The waveform relationships among the EPLD #2 signals are shown in Figure 5.3.

Finally, the divide-by-four circuit is used for reducing the 9.8304 MHz clock used by the modem ICs to frequencies acceptable to the 8251 PCI. The transmitter clock (/TxC) on the PCI is programmable to be either one, sixteen, or sixty-four times the baud rate. The 9.8304 MHz clock is divided by four in EPLD #3 to give 2.4576 MHz, which is in turn divided by four in EPLD #2 to give 614.4 kHz, which is sixty-four times the baud rate of 9600.

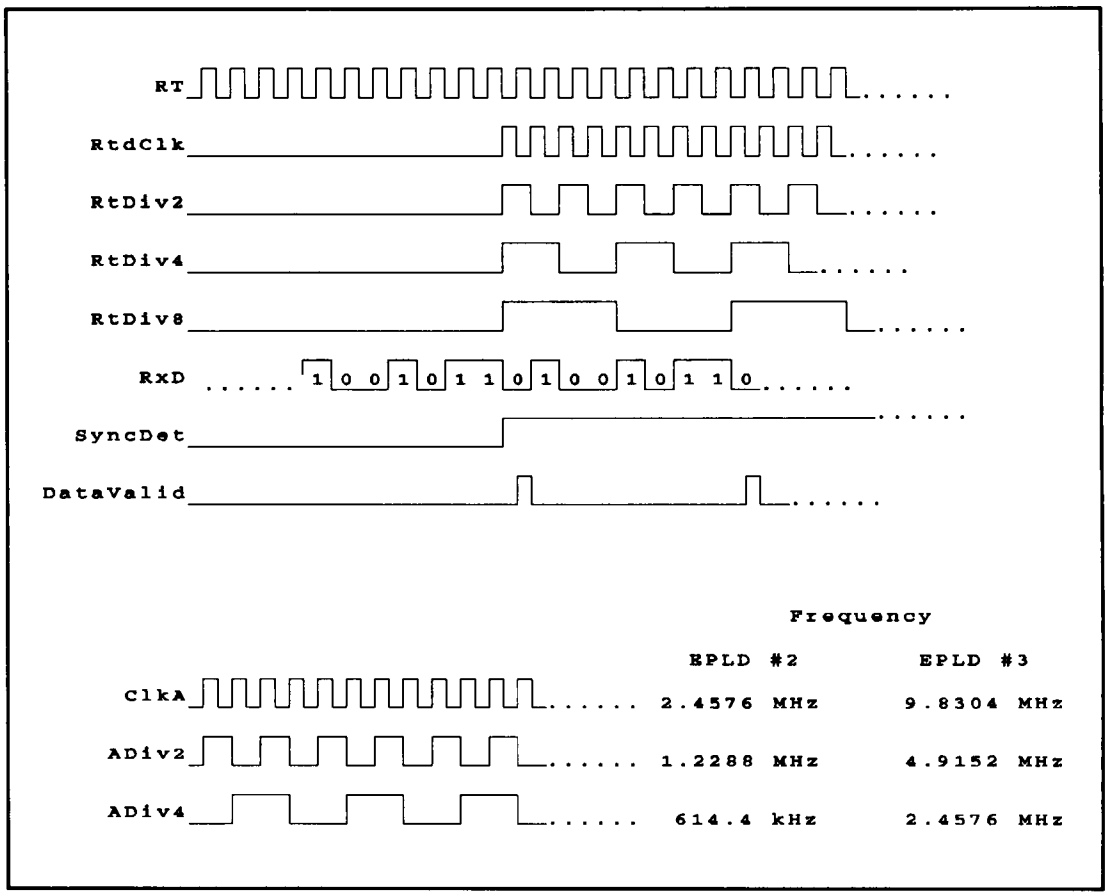


Figure 5.3. EPLD #2 Timing Diagram

Playback Mode Microsequencers

When in playback mode, SAM #2 from Figure 5.1 controls data flow from VCR audio channel two, through the channel two modem IC and PCI, and eventually to the computer. An operational flowchart is shown in Figure 5.4. The original implementation attempted to demodulate both audio channels as they were played back from tape and download both channels simultaneously to the computer. For this to be possible, both channels would

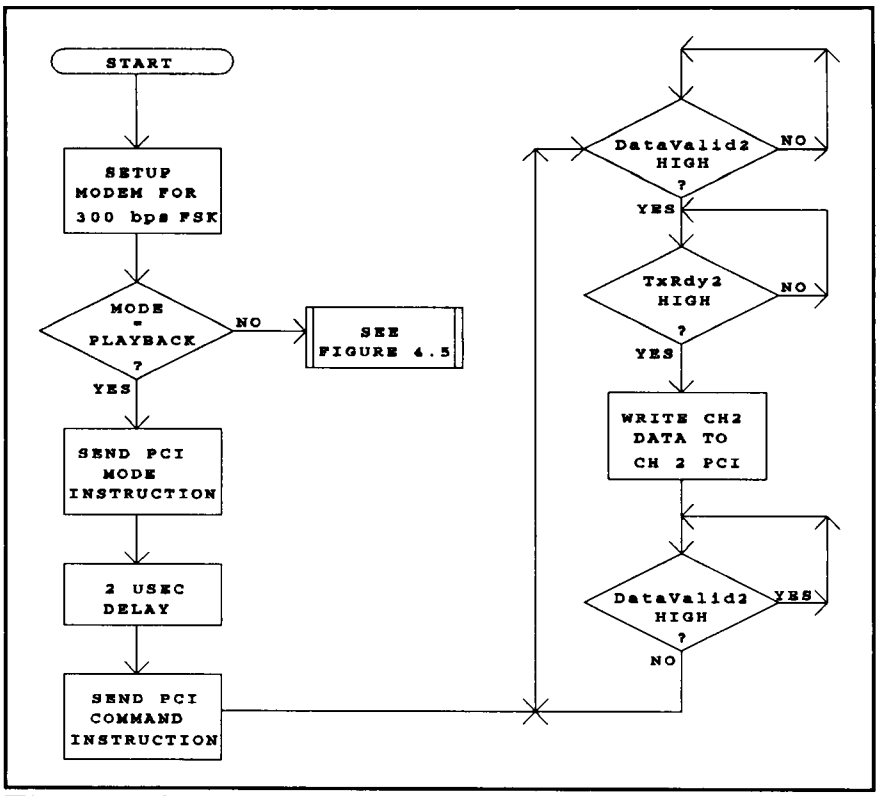


Figure 5.4. SAM #2 Playback Mode Flowchart

have to be receiving data and downloading it in sync with each other. In practice, however, the receive data clocks from the two modems were out of phase, causing DataValid1 and DataValid2 to be out of sync. The final working version of the DDTI demodulates only audio data input on channel two. The tape must be played back twice to get all of the data downloaded to the computer.

After putting the modem in 300 bit per second FSK mode, the mode instruction is sent to the 8251 PCI. Since a two microsecond delay is required between PCI writes, a delay is implemented next, then the command instruction is transmitted. The mode instruction, 0x6F, tells the PCI to

transmit data with one stop bit, no parity bit, eight bits per character, and that the input transmitter clock (/TxC) is sixty-four times the baud rate. The command instruction, 0x01, simply instructs the PCI to commence normal mode transmission.

After transmitting the command word, the sequencer waits for the channel two data to become valid and the PCI to request data. When this happens, the data is transferred from the EP600 shift register to the PCI. Since this transfer is completed before the DataValid2 line goes back low, the sequencer waits for the falling edge on DataValid2 before continuing, then enters an endless loop waiting for new data to become valid and the PCI to request new data, after which it sends a new byte and repeats the process described above. Although data is transmitted from the PCI to the computer at 9600 bits per second, there are still only 37.5 bytes of data per second. The transfer of data out of the PCI must be faster than the 300 bit per second rate out of the modem in order to make room for start and stop bits. In the original design, when the data was demodulated by the modem at 9600 bits per second, it was transmitted to the computer at 19200 bits per second.

CHAPTER 6

DATA ACQUISITION SOFTWARE

This chapter describes the software used to acquire the imaging system data on the computer. Transfer of the serial data to the computer is accomplished through the program GETG, written in C. Another C program, CAMDAC, acquires the video data through the frame grabber after the addressing code has been written to the tape using the BCD-4000 VCR controller.

DDTI Serial Data Transfer

The standard PC-AT comes with two serial ports, COM1 and COM2. Each of these ports has eight registers, as defined in Table 6.1. If the system is configured properly, whenever one of the ports receives a byte, a hardware interrupt is generated and CPU control is vectored to an interrupt service routine (ISR) based on an address stored in a vector table in low memory. Since multiple interrupts could possibly occur at the same time, two 8259 programmable interrupt controllers (PICs) in the PC are used to manage the hardware interrupts on a priority basis. The 8259 interrupt mask register (IMR) at port 0x21 can be used to mask out unwanted interrupts by controlling whether a particular interrupt can be processed by the PIC.

Table 6.1. PC-AT Serial Interface

Register	Function
Base	Data
Base+1	Interrupt Enable
Base+2	Interrupt Identification
Base+3	Data Format
Base+4	Serial Control
Base+5	Serial Status
Base+6	Input Status

After prioritizing the hardware interrupts, the PIC generates a software interrupt. Software interrupts are enabled with the STI instruction. COM1, with a base address of 0x3F8, is assigned hardware interrupt level 0x04 and software interrupt level 0x0C; and COM2, with a base address of 0x2F8, is assigned hardware interrupt level 0x03 and software interrupt level 0x0B. For an ISR to be called, one must enable interrupts using the interrupt-enable register, the serial control register, and the IMR.

The GETG main program sets up two data buffers for the incoming data, one for COM1 and another for COM2. In the original design, COM1 was to be used for data from audio channel one, and COM2 was to be used for data from audio channel two; but since two tape passes are used in the final working system, one of the serial ports is unused. The two buffers are divided into halves, an upper half and a lower half, and a separate buffer pointer for each buffer is initialized to point to the first byte in the lower half.

The ISR for each channel will read the incoming byte into the current position in the buffer, and increment the buffer pointer. If the buffer is exactly half full, a buffer half full flag is set; if the buffer is full, a buffer full flag is set and the pointer is reset to point to the first byte in the buffer. The buffer full and half full flags, as well as the buffers themselves, can be accessed by the main program. After reading the data and processing the flags, the ISR will send an end of interrupt notice to the PIC in the PC and then execute a return from interrupt instruction.

After setting up the input data buffers, the main program opens output files for data from each of the two serial ports. Next, the contents of the IMR are saved so that GETG can leave the system in its original state after gathering the data. Interrupt levels three and four are now disabled by masking them in the IMR. This is done to insure that no interrupts will occur while the new ISR addresses are being installed. Now, the old ISR addresses are saved and the new addresses installed. Next, the serial ports are set up for data transmission (one start bit, eight data bits, one stop bit) at 9600 bits per second, then level three and four interrupts are enabled, followed by setting up the serial port universal asynchronous receiver-transmitter (UART) internal to the PC to enable interrupts and enabling receive interrupts on both ports. Finally, the interrupt enable flag is set to enable processor interrupts.

The main program now enters a loop, waiting for a key to be pressed or one of the buffers to become either half full or full. If the buffer half full condition is detected, the lower half of the buffer is written to disk and that flag cleared. If a buffer full is detected, the upper half of the buffer is written to disk and that flag is cleared. In this way, a continuous stream of data can come in on both channels simultaneously and be written to disk without ever a byte being lost. While one half of the buffer is being filled, the other half is being written to disk, and vice versa. The buffer size is made sufficiently large that a disk write of half a buffer finishes well before the other half of the buffer is filled. After a key is pressed on the keyboard, interrupt levels three and four are again disabled and the old serial port interrupt vectors are restored along with the old interrupt mask register. Also, any partially filled buffers are written to disk and the output files are closed. A flowchart of the GETG main program and ISR can be found in Figure 6.1. Note that the ISR for COM1 is identical to the ISR for COM2 except for the interrupt level and buffer address.

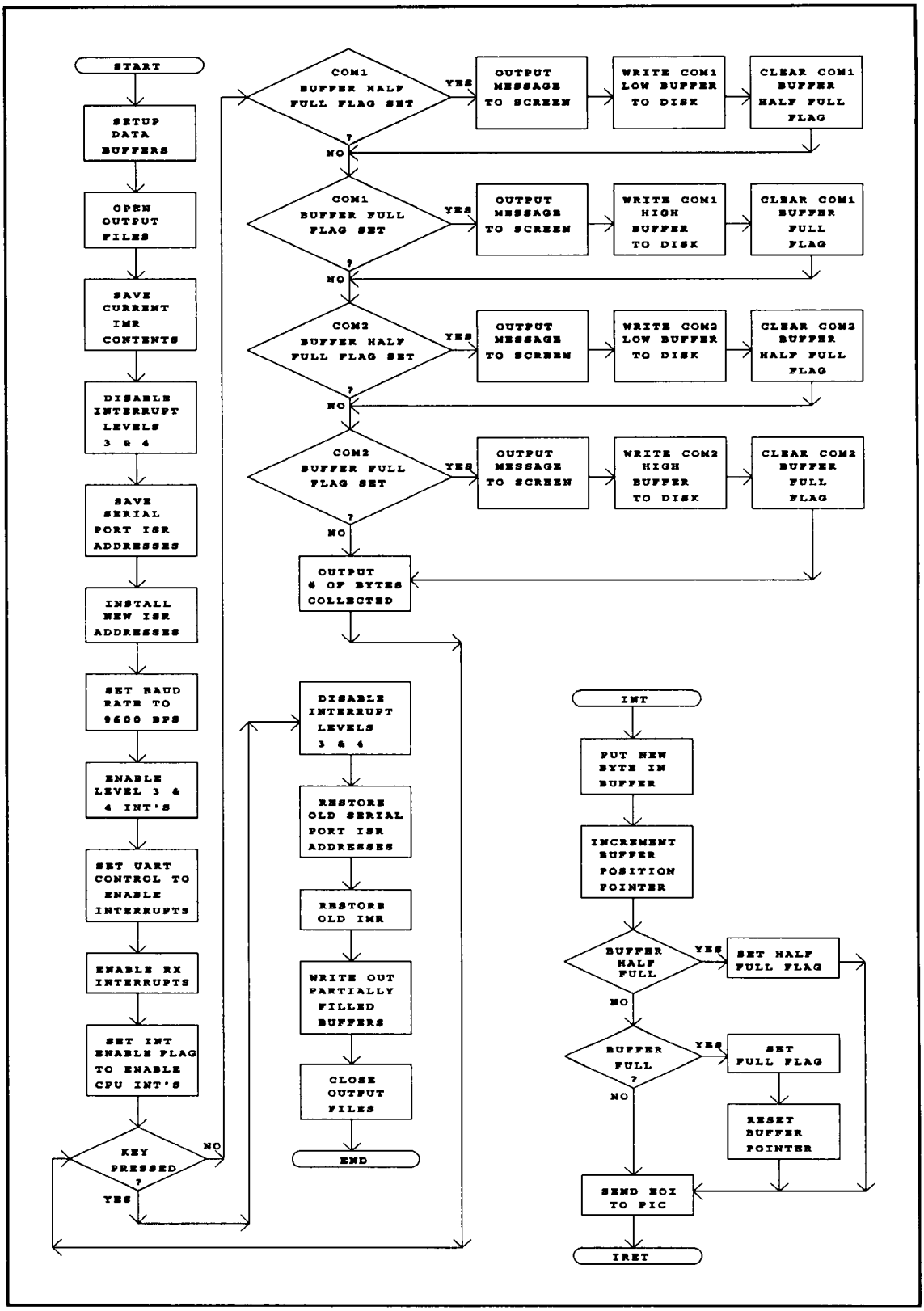


Figure 6.1. Serial Data Transfer Program Flowchart

Frame Grabbing

This software integrates the Matrox frame grabber, the JVC video recorder, and the BCD-4000 video control system, while making possible the digitization of a video frame falling at any given IRIG time. One of the commands provided by the BCD board is STIL X, which causes the VCR to display a still frame at frame X. The first approach at digitizing video played back from the VCR was to use the BCD board to position the tape at the desired frame using STIL X, then to call the Matrox frame grabber and digitize the currently displayed frame. While this approach did provide digitization of exact video frames, it was not acceptable due to jitter present in the stilled image.

Experimentation with the system showed that the jitter problem was not present in frames digitized while the VCR was in the Play mode. The jitter problem was corrected by writing software to place the VCR in Play mode and monitor the current frame number. It was found that if the frame grab subroutine was called exactly two frames before the frame that should be digitized, it would actually digitize the desired frame. A flowchart of this program, CAMDAC.C, is shown in Figure 6.2.

After reading the desired frame number and the output file name, CAMDAC resets the BCD-4000 and initializes it for control of the JVC CR-

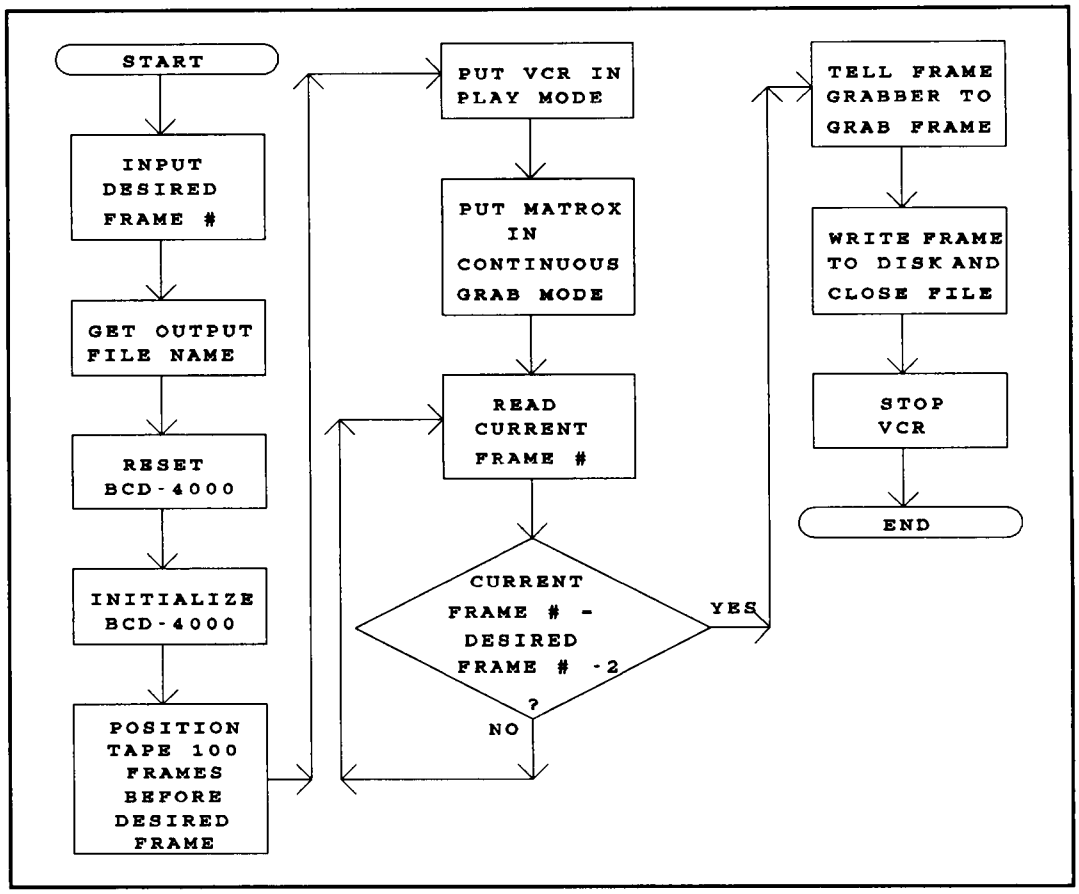


Figure 6.2. Frame Grab Program Flowchart

600U. Next, using the VCR controller, the tape is positioned 100 frames before the desired frame, then the VCR is placed in PLAY mode, all under software control. At the same time, the frame grabber is placed in continuous grab mode, so it can become synced to the incoming video. After this, the program begins monitoring the current frame number, and when the tape gets to within two frames of the desired frame, the frame grab subroutine is called and the grabbed frame is written to disk.

A frame number differential will be present between CAMDAC and EXTRACT (described in Chapter 7) since the audio data does not begin until

the frame just after the image intensifier is turned back on, but the addressing code written by the VCR controller begins at the beginning of the tape, usually about seven seconds earlier. This is easily corrected by stepping the video a frame at a time until the intensifier has turned off and then come back on, and recording the frame number. This number decremented by one should be subtracted from the frame number used by CAMDAC and used as the input to EXTRACT as described in Chapter 7.

CHAPTER 7

DATA REDUCTION SOFTWARE

This chapter contains a description of the software used for data reduction. A C program, EXTRACT, is used to calculate the inverse responsivity for each frame. Input to this program is a frame number and two data files containing the audio channel data; output is the IRIG time and inverse responsivity for the indicated frame, which is calculated based upon the gain range and duty cycle for that frame. The relationship of the gain data to the inverse responsivity is described in Chapter 8. The inverse responsivity is used by the FORTRAN program CAMPLOT to produce iso-radiance contours and gray scale plots of individual frames digitized with CAMDAC.

Inverse Responsivity Calculation

A flowchart of EXTRACT.C can be found in Figure 7.1. This program gets the time and gain data for a particular frame from the audio channel input files, and writes it into a header to the data file along with the maximum and minimum pixel values, the location of the maximum pixel value, the frame number, and the inverse responsivity. After setup of header and image

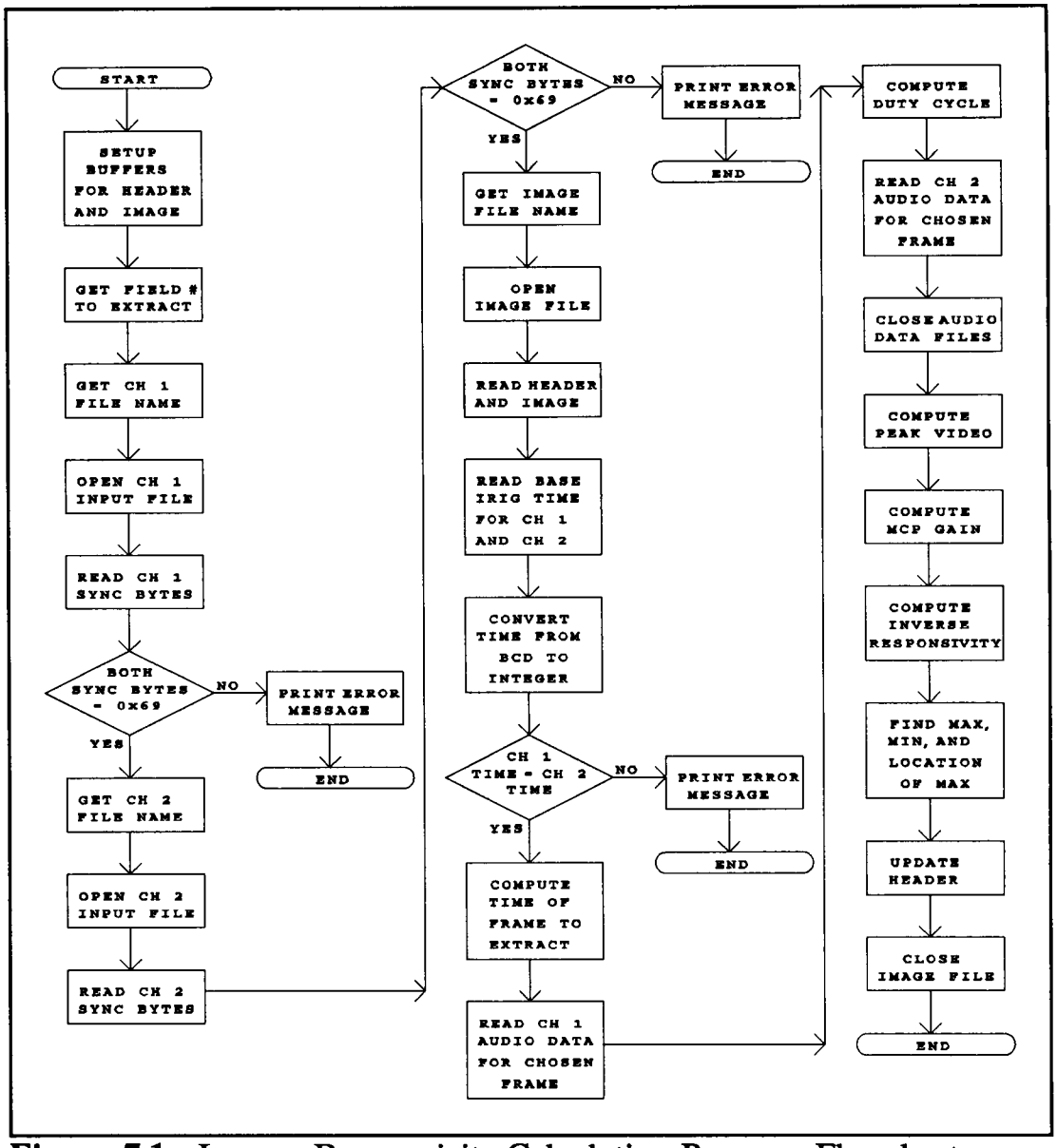


Figure 7.1. Inverse Responsivity Calculation Program Flowchart

buffers, the frame number to extract is read. Next, the audio channel file names are read, and the files are opened. These files were created with GETG. The sync bytes from both channels are read and verified. If they are not 0x69, an error message is printed and the program is terminated.

If both sync bytes in both data files are correct, the image file name is input and the 512 x 480 image is read. This is followed by reading the IRIG time data from channel one, and decoding it from the packed BCD format to separate data variables for days, hours, minutes, seconds, and milliseconds. This is repeated for channel two. If the two sets of time data are not equivalent, an error message is printed and the program is terminated. If the time data from both audio channels matches, the time of the frame to be extracted is computed by adding the product of the frame number multiplied by 1/30 to the base time for the first frame.

After computing the frame time, channel one audio data for the chosen frame is read, and the duty cycle is computed; then channel two data is read and the peak video and MCP gain are computed. Sample peak video decoding is given in Table 7.1, and sample duty cycle decoding is given in Table 7.2. The two bit MCP gain state encoding format is given in Table 2.2 on page 15. To decode the two packed bytes containing the MCP gain and the high two bits of duty cycle for each of four frames, the remainder of the desired frame number divided by four is found. If the most significant bit of

Table 7.1. Sample Peak Video Decoding

Peak Video (volts)	A/D Counts (hex)	A/D Counts (decimal)
0.00	0x80	128
0.25	0xA0	160
0.50	0xC0	192
0.75	0xE0	224
1.00	0x00	0
1.25	0x1F	31
1.50	0x3F	63
1.75	0x5F	95
2.00	0x7F	127

Table 7.2. Sample Duty Cycle Decoding

Duty Cycle (%)	Counts (hex)	Counts (decimal)
0	0x200	512
25	0x300	768
50	0x000	0
75	0x100	256
100	0x1FF	511

the packed byte is bit seven and the least significant bit is bit zero, then a remainder of zero means that the data for that frame is in bits six and seven of the packed byte. Bits four and five correspond to frame numbers giving a remainder of three when divided by four, bits two and three for a remainder of two, and bits zero and one for a remainder of one. Finally, the duty cycle and MCP gain data is used to calculate the inverse responsivity, the maximum and minimum pixel values are found, and the header is updated and the image file closed.

Data Plotting

A flowchart of CAMPLOT.FOR can be found in Figure 7.2. After getting the input image file name, this program puts the video board in graphics mode and draws a contour plot of the image, then a plot title is displayed based on information in the header. The fifteen contours, corresponding to fifteen different on-screen colors, are automatically set so that they are spaced evenly between the maximum and minimum pixel values, which are also in the header.

After drawing the iso-radiance contour plot, the program enters a loop waiting on a key to be pressed. If the escape key is pressed, text mode is entered and the program is terminated. If the F2 key is pressed, the display mode is toggled between iso-radiance contours and gray scales. If the F1 key

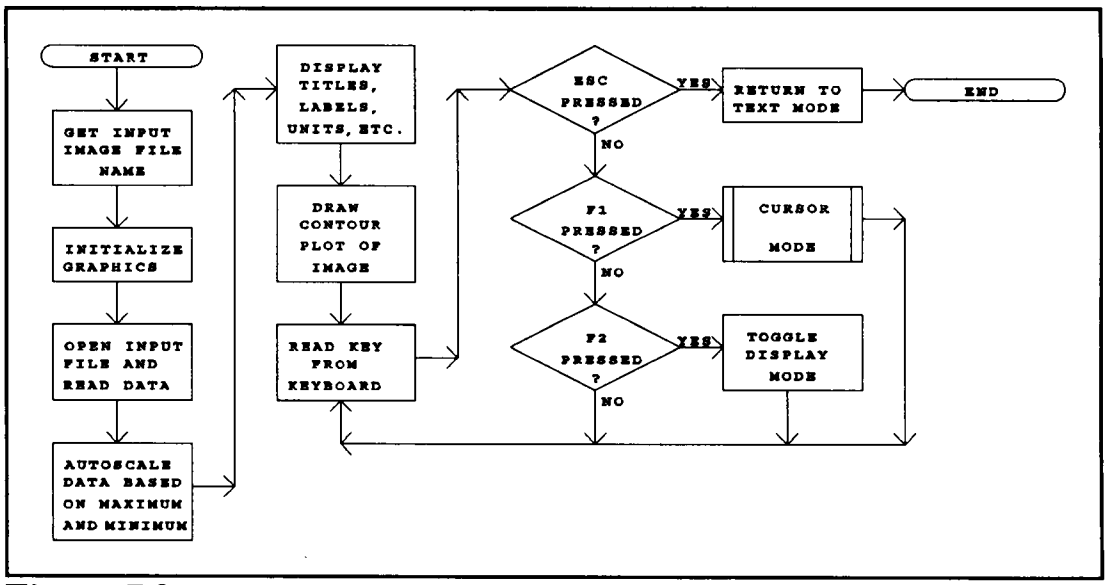


Figure 7.2. Data Plotting Program Flowchart

is pressed, a cursor mode is entered in which a cursor can be moved around the screen to display the value of individual pixels. Cursor mode is exited when the escape key is pressed. A constant (the inverse responsivity factor computed by EXTRACT) in the header is used to convert each pixel to radiance units. A sample gray scale plot generated using the Xerox color ink jet printer is shown in Figure A.1 on page 88 in the Appendix. The color bar along the right side of the plot shows the range of values of pixels of a given color.

CHAPTER 8

CALIBRATION

The first step in calibrating the intensified camera image processing system is to determine the relationship of the peak video level to the gate duty cycle. This is determined for the high gain range by putting the autoranging camera controller in manual mode and manually varying the duty cycle. An expression is determined for the peak video (PV) in terms of the duty cycle (DC) by expanding linearly about a point corresponding to one volt. Next, the peak video relationship to the MCP gain range is determined by leaving the duty cycle fixed and manually changing the gain range. It was found that a gain of 4.1 is present between the low range and the medium range, and that a gain of 3.7 is present between the medium and high ranges.

The imaging system is normally operated with the camera fitted with a filter that transmits only a portion of the total detectable wavelength range. The camera is sensitive to in-band source radiance in units of power/unit source area • unit solid angle. Once the peak video relationship with the gate duty cycle and MCP gain range is determined, a radiance calibration is made using a standard radiance source. However, the standard source is a spectral irradiance standard calibrated in units of power/unit wavelength • unit area at some specified distance from the lamp. A standard radiance source is

obtained by reflecting the irradiance standard from a Lambertian surface (diffuse reflector) at the specified distance. A sketch of the calibration setup is shown in Figure 8.1. The shield was used to block the arc lamp from the field of view of the camera, and the camera was placed ten feet from the source because the lens was focused at this distance. The lamp used here was a 1000 watt quartz iodine lamp, but for calibrating the camera with filters that pass wavelengths below about 250 nanometers, a deuterium arc lamp calibrated at thirty centimeters is used.

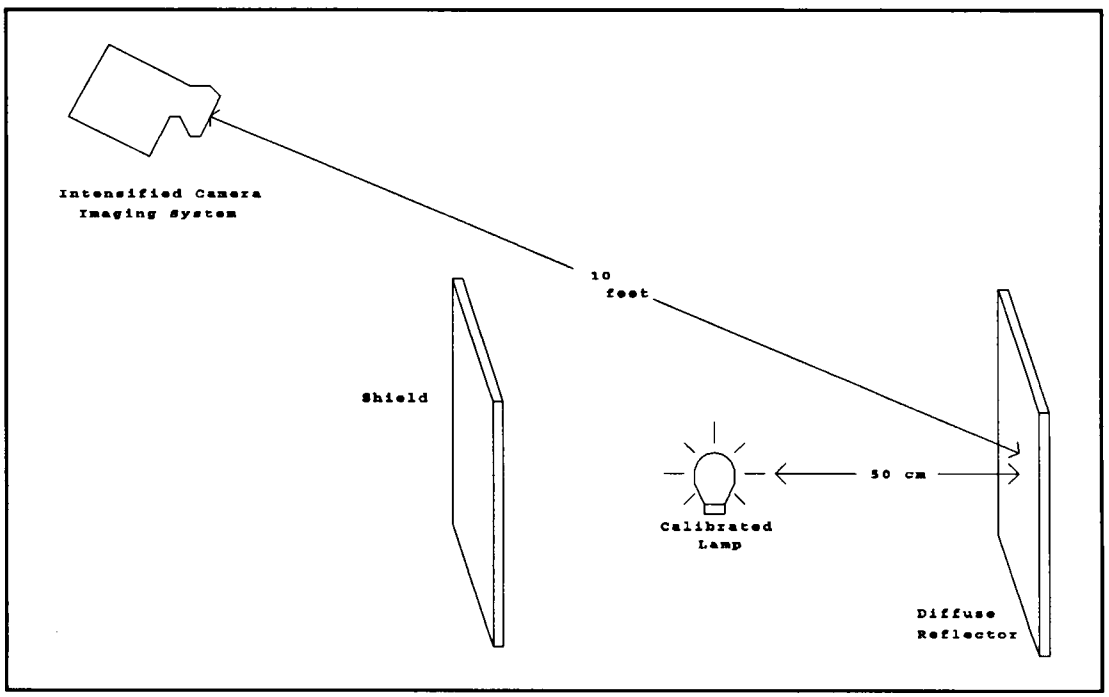


Figure 8.1. System Setup for Calibration

The spectral radiance of the source is the lamp spectral irradiance times the diffuser reflectance divided by pi. Both the lamp spectral irradiance and

the diffuser reflectance can be found from data sheets supplied by the manufacturers. The actual in-band source radiance can be found by integrating the product of the normalized system response and the source radiance data. The system response is the product of the filter transmission, the photocathode sensitivity, the lens transmission, and the reflectivity of the diffuse reflector, which are all a function of wavelength. Using the auto-ranging camera controller and the digital data tape interface (DDTI), a recording was made on the VCR, then a single frame and its corresponding gain data were grabbed on the PC. Since the MCP gain and gate duty cycle for each frame are known, and the in-band radiance from the calibration source is known, an inverse responsivity factor (R^{-1}) can be computed for each frame in units of power/unit source area • unit solid angle • count, where a count is the unit assigned to the eight bit value obtained for each pixel when the analog video signal is digitized. The inverse responsivity is just the in-band source radiance divided by the instantaneous peak video (in counts) of the calibration source, where the instantaneous peak video is a function of the duty cycle as determined above.

A radiance value for each pixel can now be computed from the equation

$N = C \times G \times R^{-1}$ [power/unit source area x unit solid angle]

where N is the radiance at a given pixel, C is the A/D counts for that pixel, G is the gain factor (1.0 for high, 3.7 for medium, 15.2 for low), and R^{-1} is the inverse responsivity.

CHAPTER 9

RESULTS AND DISCUSSION

The product of this work permitted an integrated system for the recording and processing of data from an autoranging intensified camera system to be completed. Using an intensified silicon target camera and autoranging camera controller from a previous system, the new system adds an IRIG time code translator, a video recorder, a computer system, and a digital data tape interface (DDTI). The DDTI was developed as part of this study to allow recording of the system gain and time on the audio channels of the recorder, which makes possible the reduction of a video signal and an instantaneous camera gain indication to absolute radiometric units, and provides temporal correlation of data taken simultaneously from multiple radiometric systems. Unlike previous data processing methods for intensified cameras, the addition of a computer to the system developed here provides automated data processing and dedicated image analysis capabilities with a well defined data calibration, acquisition, and reduction procedure.

The new image processing system provides significant improvements over the previous approach to reducing intensified camera data. Where the old system could only produce normalized contour plots with a manual calculation of the peak video level, the new system can reduce data pixel-by-

pixel automatically using a microcomputer and provides either gray scale or contour plots. Storage of the data is also consolidated, with both video and gain data going on the VCR tape instead of the video on VCR tape and the gain data on FM tape. Whereas the time correlation method was previously based on time insertion into the video and could only be determined to the nearest second, it can now be resolved to the nearest millisecond for each frame of video. Using the computer based VCR controller to provide precise frame positioning enables digitization, frame time determination, and quantitative radiometric data reduction of up to thirty frames of video per second. This allows the reduction of transient data which was unusable before; and it speeds up the reduction process, resulting in a complete image processing system that can reduce data in a matter of hours as opposed to days, without sacrificing image quality.

The development of the DDTI itself provided insight into the intricacies involved with audio encoding of digital data. Whereas the original intent was to modulate the data using quadrature amplitude modulation at 9600 bits per second, anomalies inherent in the audio channel of a tape recorder forced the use of frequency shift keying at 300 bits per second. This compromise necessitated the use of both audio channels of the tape for the ensuing development, but the original intention to develop a system that could record the system gain and time for each frame was still met. The resultant system

is actually more flexible in that it can record additional data, up to a total of twenty bits per frame, using the customized DDTI circuitry.

CHAPTER 10

SUMMARY AND RECOMMENDATIONS

Summary of Results

An intensified silicon array camera image recording and processing system has been designed and built that will reduce a video signal and an instantaneous camera gain indication to absolute radiometric units in addition to providing temporal correlation of data taken simultaneously from multiple radiometric systems. The new automated system has resulted in reduced data reduction times and has been utilized with four different cameras during the writing of this thesis--three UV sensitive intensified silicon array type cameras and one low light level silicon target type visible camera. The technique has proven so successful that another digital data tape interface (DDTI) has been built for use exclusively with the visible camera. The systems have been employed to reduce optical data covering various spectral regions to absolute radiometric units for tests conducted at the AEDC.

Recommendations

Some areas for improvement do remain in the image processing system. One observed problem is that the gain changes (or can change) with every video field, but the frame grabber grabs an interlaced frame consisting of two fields. Under transient conditions when the video level is changing rapidly, every other line of a frame will have different gain parameters. This can be remedied by modifying the autoranging camera controller so that it only changes the MCP gain and gate duty cycle parameters every other field. Another area for possible improvement is to enhance the software. The functions of CAMDAC and EXTRACT could be combined, so that the gain data is automatically extracted whenever a frame of video is digitized. Another enhancement would place the VCR under computer control during the operation of the GETG program, instead of having to operate it manually. Still another enhancement would be the use of a video laser disk in place of the VCR. This would reduce data reduction times since it would allow virtually instantaneous random access to individual frames of video, as opposed to the slow sequential access of the tape; and it would make possible the reduction of every frame of data to engineering units, instead of selected frames as is done with the VCR.

LIST OF REFERENCES

LIST OF REFERENCES

1. Bridges, J.H., III. "The Development of a Video-Level Controller for an Ultra-Violet Intensified CCD Camera." Master's Thesis, University of Tennessee Space Institute, March 1988.

2. "Model 175-321 Timing Control Unit." Kode Inc., Anaheim, California.

3. "JVC Instructions: Color Video Cassette Recorder CR-600U." Victor Company of Japan, Ltd., 1987.

4. "JVC Service Manual: Color Video Cassette Recorder CR-600U." Victor Company of Japan, Ltd., March 1987.

5. "Operation Manual for a Gated Solid State Video Camera." ITT Project 14851, ITT Electro-Optical Products Division, Fort Wayne, Indiana, October 1988.

6. "Solid State CID Camera User's Manual." CID Technologies, Inc., Liverpool, New York.

7. Eberhardt, E.H. "Gated Microchannel Plate Detectors." Applications Note E23, ITT Electro-Optical Division, 1984.

8. "BCD-4000 Professional Video Controller." BCD Associates, Inc., Oklahoma City, Oklahoma, July 15, 1988.

9. "MVP-AT Installation Manual." Manual Number 272-MY-00, Revision 10. Matrox Electronic Systems Limited, Dorval, Quebec, January 31, 1989.

10. "Introduction to the MVP-AT: A Machine Vision Processor." Manual Number 272-MP-00, Revision 4. Matrox Electronic Systems Limited, Dorval, Quebec, February 1, 1989.

11. "Imager-AT: MS-DOS Software Library for the MVP-AT Video Digitizer, Reference Manual." Manual Number 272-MF-00. Matrox Electronic Systems Limited, Dorval, Quebec, January 31, 1989.

12. "Microsoft QuickC: C for Yourself." Microsoft Corporation, Redmond, Washington, 1988.

13. "F77L-EM/32 32-Bit FORTRAN 77 Language System for 80386-Based Personal Computers: Reference Manual." Lahey Computer Systems, Inc., Incline Village, Nevada, June 1989.

14. "Graphics Development Toolkit." Volumes 1 and 2. Graphic Software Systems, Inc., Beaverton, Oregon, August 1988.

15. "Pizazz Plus: Software to Improve Your Image." Application Techniques, Inc., Pepperell, Massachusetts, 1989.

16. "YM7109C 9600 bps Fax Modem Application Manual." 1989.12, Yamaha Corporation of America, Systems Technology Division, Buena Park, California.

17. Folts, Harold C., Ed. "Data Communications Standards." McGraw-Hill Publications Company, New York, 1982

18. "CMOS Parallel First-In/First-Out FIFO: 2048 x 9-Bit and 4096 x 9-Bit." Integrated Device Technology, Inc., Santa Clara, California.

19. "Altera Data Book." Altera Corporation, Santa Clara, California, 1987.

20. "A+ Plus Reference Guide." Version 5.1, Altera Corporation, Santa Clara, California, September 1987.

21. "A+ Plus User Guide." Version 5.1, Altera Corporation, Santa Clara, California, June 1988.

22. "LogicMap II." Version 5.0, Altera Corporation, Santa Clara, California, September 1987.

23. "SAM+ Plus." Version 1.0, Altera Corporation, Santa Clara, California, July 1987.

24. "Component Data Catalog." Intel Corporation, Santa Clara, California, January 1981.

25. "Maxim +5V Powered RS-232 Drivers/Receivers." Maxim Integrated Products, Sunnyvale, California, 1988.

APPENDIX

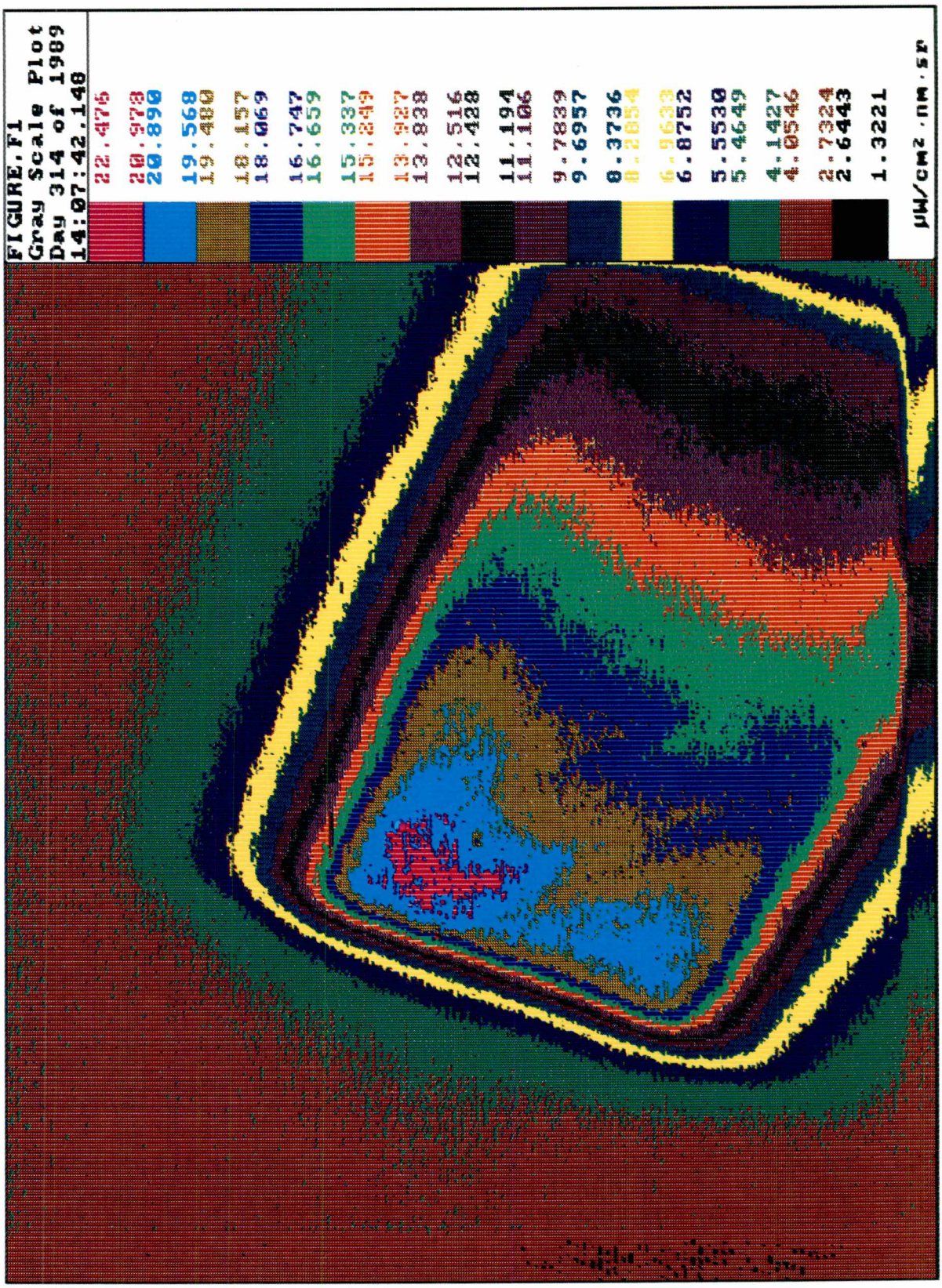


Figure A.1. Sample Gray Scale Plot

VITA

Timothy Wayne Bratcher, son of Mr. and Mrs. Melvin Bratcher, was born in Smithville, Tennessee on October 2, 1965. He attended elementary schools in Warren County and was graduated from Warren County Senior High School in May 1983. The following September he entered Rose-Hulman Institute of Technology, and in May 1987 he received a Bachelor of Science degree in Electrical Engineering and Computer Science. In June of 1987 he accepted a position with Sverdrup Technology, Inc., at the Arnold Engineering and Development Center, where he is currently employed as an instrumentation engineer in propulsion diagnostics research. Also in June 1987 he enrolled at the University of Tennessee Space Institute and began study toward a Master's degree in Electrical Engineering, which was awarded in August of 1990.