The Pennsylvania State University

The Graduate School

Department of Electrical Engineering

DESIGN OF A DATA ACQUISITION AND CONTROL SYSTEM FOR A VOLUME SCANNING MULTI-WAVELENGTH POLARIZATION LIDAR

A Thesis in

Electrical Engineering

by

Michael D. O'Brien

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 1994

We approve the thesis of Michael D. O'Brien.

Date of Signature

_____	_____
Charles R. Philbrick Professor of Electrical Engineering Thesis Advisor	
_____	_____
Timothy J. Kane Assistant Professor of Electrical Engineering	
_____	_____
Larry C. Burton Professor of Electrical Engineering Head of the Electrical Engineering Department	

Abstract

A system architecture for a data acquisition and control system capable of handling the demanding data throughput requirements of a volume scanning, multi-wavelength polarization lidar will be presented. The volume scanning lidar is one component of a new research tool called WAVE-LARS (Water Aerosol Vapor Experiment - Lidar And Radar Sounder). This unique instrument employs a variety of remote sensing techniques to collect information on several atmospheric properties important in solving problems in areas of research such as the Earth's radiation budget and cloud physics. The scanning lidar measures the depolarization ratio at three wavelengths to determine the particle size and shape distributions in a three dimensional volume of the atmosphere.

The design of the data acquisition and control system for the volume scanning lidar begins with a consideration of the expected input signals, the desired output signals, and the hardware constraints. This information is then used to formulate design goals and system specifications. A system architecture capable of attaining those goals is then presented, followed by a description of the hardware and software used to implement the proposed system. Test results of critical functions of the constructed system are then presented validating the systems performance. The first atmospheric data collected by the system is also presented. Finally, the system design is evaluated and suggestions are made for improvements and future research.

Table of Contents

List Of Figures

List of Tables

Acknowledgments

First of all I would like to thank my thesis advisor, C. R. Philbrick, for all his valuable advice and support. I would also like to thank T. Kane for his many useful suggestions in the preparation of this thesis.

Additionally I would like to thank the many people who aided me in conducting the research described in this thesis, D. Lysak, R. Smith, G. Pancoast, T. Petach, G. Evanisko, B. Mathason, T. Stevens, J. Yurack, J. Anuskewicz, T. Manning, P. Haris, S. Rajan, D. Machuga, S. Mckinley, S. Sprague, and S. Maruvada.

The WAVE-LARS system was funded by the Department of Energy, The Pennsylvania State University, and the National Science Foundation.

Chapter 1

Introduction

The investigations into the composition of the Earth's atmosphere and the propagation of optical radiation through the atmosphere are interrelated fields of study pursued by a variety of scientists and engineers. The scattering and absorption of solar and terrestrial radiation by atmospheric aerosol particles plays an important role in determining the Earth's climate by affecting the rate that energy enters and leaves the Earth's atmosphere. Atmospheric aerosols are also involved in the formation and sustenance of clouds, which have a large influence on the thermodynamic processes of the atmosphere. Additionally, scattering from atmospheric particles affects the operation and calibration of optical sensors used in such applications as meteorological research, pollution monitoring, and optical communications.

For scientists and engineers to successfully solve the problems described above, it is necessary to obtain experimental data on properties of the atmosphere with improved spatial and temporal resolution. This data is important for researchers to formulate and verify the theoretical models allowing them to make reliable predictions on several important issues. Some of these include the radiative processes that could lead to global warming, the processes of cloud physics which are important for understanding the Earth's hydrological cycle, and the properties of atmospheric optical propagation governing the performance of optical systems. The most effective method to achieve this goal is to utilize the synergism generated by employing a variety of measurement devices and techniques. These devices should include in situ sensors, active remote sensors, and passive remote sensors.

At Penn State, an interdisciplinary group of researchers is developing a new research tool called WAVE-LARS. This unique instrument employs several different lidar and radar remote sensing techniques to gather a substantial amount of data about a variety of atmospheric parameters. This information can be combined with information from other instruments to address the scientific and engineering questions previously discussed. This thesis describes the design of a system architecture for a data acquisition and control system capable of handling the data throughput requirements of a multi-wavelength, volume scanning, polarization lidar. This lidar plays an important role in allowing WAVE-LARS to achieve a better understanding of the structure and dynamics of our atmosphere.

1.1 WAVE-LARS

The data acquisition and control system described in this thesis is a component of WAVE-LARS, or Water, Aerosol, Vapor Experiment - Lidar And Radar Sounder. This instrument was developed as a Department Of Energy University Research Instrument (URI) initiative by Penn State's Communications and Space Sciences Laboratory (CSSL) in the Department of Electrical Engineering and the Department of Meteorology. The components of this instrument include a new volume scanning, multi-wavelength, polarization lidar, a new volume scanning, 94 GHz radar, and a fixed position Rayleigh/Raman lidar. These instruments are housed together in standard shipping containers that serve as easily transportable field laboratories.

The volume scanning lidar is an instrument that collects the backscattered radiation from laser pulses transmitted into the atmosphere while scanning a three-dimensional region

of the atmosphere. This optical instrument is designed to collect information on the size distribution of atmospheric aerosol particles and on the phase of ice and water particles in clouds. It accomplishes this by transmitting light pulses in the ultraviolet, visible, and infrared wavelength regions, and measuring the backscattered returns parallel and perpendicular to the transmitted laser polarization.

The 94 GHz radar will complement the lidar measurements by penetrating thicker clouds and measuring cloud structure and liquid water content within the cloud. This radar has a polarization measurement capability, enabling it to detect and locate ice crystals in clouds. This radar was designed based on a 94 GHz radar currently operating at Penn State. The new radar will be mounted on a volume scanning platform and will simultaneously take measurements in the same region as the volume scanning lidar.

Another lidar system will supplement the information gathered from the volume scanning lidar and volume scanning radar with vertical profiles of water vapor content and temperature. This instrument is the LAMP lidar system which is a Rayleigh/Raman lidar constructed by CSSL and Penn State's Applied Research Laboratory.

The simultaneous measurement of depolarization ratios of backscattered returns from transmitted electromagnetic pulses in the ultraviolet, visible, infrared, and microwave portions of the electromagnetic spectrum, together with lidar water vapor and temperature measurements, will provide a unique and powerful tool for studying the structure and dynamics of the atmosphere. Some scientific goals which the WAVE-LARS instruments will attempt to address are:

1. Remote sensing of cirrus cloud properties which have been identified as important

in modeling global radiation transfer.

2. Studies of cloud formation using a combination of aerosol sensing below the cloud with the volume scanning lidar, water vapor and temperature profiles below the cloud using the Raman lidar, and remote sensing of interior cloud structures with the volume scanning radar.

3. Studies of optical scattering properties of aerosol particles to provide useful technical information for designers of instruments requiring the propagation of optical radiation through the atmosphere.

4. Locating regions of multiple scattering.

Integrating the data taken by these instruments makes it possible to achieve the scientific goals listed above more successfully than has been possible with previous independent measurements.

1.2 Lidar Principles

Only one of the WAVE-LARS instruments will be discussed in this thesis, the volume scanning lidar. The LAMP system and the radar system are described in detail elsewhere [Stevens, 1992] [Haris, 1993] [Ackerman, 1993]. This section will discuss the basic operating principles of a lidar system. Lidar is an acronym which stands for LIght Detection And Ranging. Lidar determines numerous properties about a scattering object by transmitting an electromagnetic pulse and monitoring the energy which is backscattered. Usually the source of the transmitted radiation is a laser, which is a highly monochromatic source of high intensity optical pulses. The pulsed nature of the laser allows the range or position of the

scattering volume to be determined.

The lidar technique evolved from the searchlight techniques used by Elterman [Elterman, 1953]. However, it was the invention of the laser in the 1960's that provided the essential technology to make lidar a feasible measurement technique. The earliest lidar measurements were taken by Fiocco and Smullin [Fiocco and Smullin,1963]. Other important early investigations were conducted by Schotland [Schotland, 1971], Cooney [Cooney, 1971], and Melfi [Melfi, 1972]. These early efforts proved that lidar could be a useful research tool and several techniques emerged during the following years. These lidar techniques included elastic backscatter Rayleigh lidars, Raman scattering lidars, polarization lidars, Differential Absorption Lidars (DIAL), and coherent Doppler velocity detection lidars. The first three of these techniques are used in WAVE-LARS and are described below.

The elastic backscattering lidar, commonly referred to as Rayleigh/Mie scattering lidar, is the simplest lidar system. The term elastic refers to the fact that the transmitted and scattered radiation are at the same frequency. The laser pulses that are transmitted into the atmosphere are scattered by atmospheric molecules and aerosol particles. This technique is used in both the LAMP system and the new volume scanning lidar.

Raman scattering lidars make use of the Raman shifted backscattered radiation and are described in the next section. This type of lidar uses vibrational Raman scattering to measure the concentrations of atmospheric constituents and to determine true optical extinction. The detection of rotational Raman lines can be used to measure the temperature profile of the atmosphere.

The third lidar technique used in WAVE-LARS measures the elastic scattered returns

at the polarizations parallel and perpendicular to the transmitted beam. This technique is an enhancement of the elastic backscatter lidar technique. The polarization information is useful in determining the size and shape of the scattering object. This information can then be applied to characterization of the particle type. For instance, it will determine the difference between ice crystals and water drops. In addition this technique makes it possible to distinguish regions of single scattering and regions of multiple scattering [Sassen, 1991].

1.3 Optical Scattering Mechanisms

An essential element in all lidar systems is the optical scattering from a volume element. There are several optical scattering mechanisms and the three of these which are applicable to WAVE-LARS, are described in this section. The three types are molecular scattering, aerosol scattering, and Raman scattering. Molecular scattering consists of the central unshifted Cabannes line plus the adjacent Stokes and anti-Stokes rotational lines [Young, 1982]. This type of scattering is often referred to as Rayleigh scattering. The molecular Rayleigh scattering can not be easily distinguished from scattering by particles which are small when compared to wavelength. The backscattering cross section for Rayleigh scattering is given by the equation,

$$\sigma_{\pi} = \frac{\pi^2 (n^2 - 1)^2}{N^2 \lambda^4} \quad , \tag{1.1}$$

where n is the complex refractive index of the medium and N is the number density of the scatterers. An important feature in molecular scattering is the λ^{-4} dependance. Another important feature of molecular scattering important to WAVE-LARS is molecular scattering

induces only a small depolarization component, typically less than 4%, due to the anisotropy of the electric dipole field of the molecule [Measures, 1984].

The second scattering mechanism used by the WAVE-LARS instrument is aerosol scattering. This type of optical scattering occurs when the transmitted light is incident upon a scatterer whose dimensions are comparable to the wavelength of the incident light. In the wavelength region comparable to the size of the scatterer, or resonance region, the scattering cross section is a very sensitive function of particle size. Figure 1.1 shows the behavior of scattering cross section in regions where the size of the particle is smaller, comparable, and larger than the wavelength of the radiation. The figure also shows that when the particle size is small compared to the wavelength, the graph approximates the λ^{-4} Rayleigh scattering behavior, and as the particle size becomes larger than the wavelength of the incident light, the cross section approaches the physical optics approximation. It is important to note that Figure 1.1 was generated from the Mie scattering theory, which applies only to spherical particles. The scattering functions from irregularly shaped particles, such as cirrus ice particles, will be more complicated in the resonance region. Also, irregular shaped particles will not preserve the incident polarization as in the case of dipole scattering. This depolarization effect can be used to learn important information about the scattering particles.

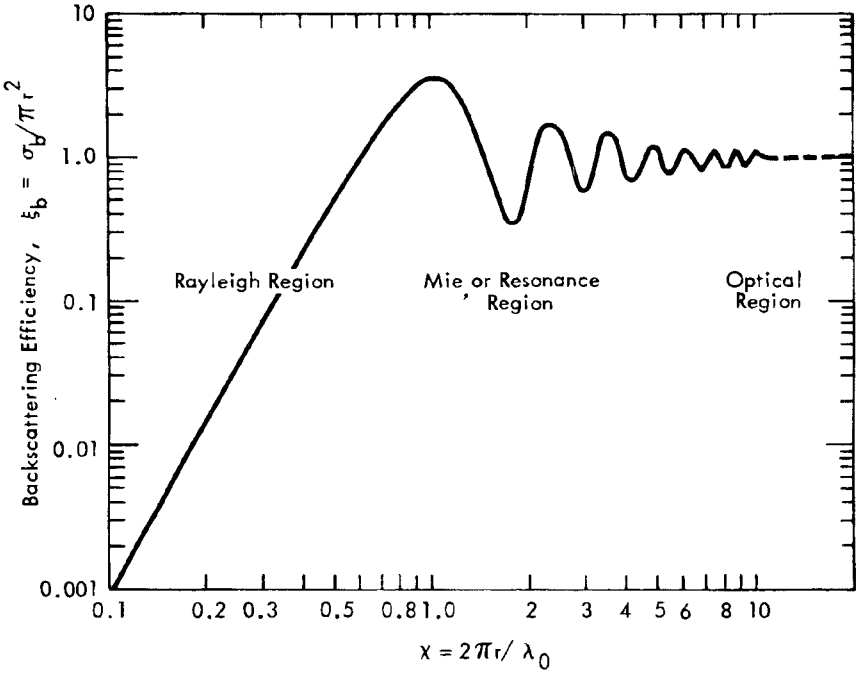

Figure 1.1. Backscattering efficiency verses size parameter [Ulaby, 1981].

The volume scanning lidar can address research areas such as aerosols and cloud physics by using the depolarization information described above at multiple wavelengths. The polarization lidar technique was one of the earliest lidar techniques, employed as early as 1970 [Schotland, 1971]. The volume scanning lidar will measure the depolarization ratio at 355 nm, 532 nm, and 1064 nm. It will then use this information to experimentally determine aerosol size distributions and the phase of water in clouds, providing important experimental data for studies in these regions.

The linear depolarization ratio is defined by,

$$\delta_p = \frac{I^{\perp}}{I^{\parallel}} \quad . \tag{1.2}$$

This depolarization can be caused by irregularly shaped aerosol particles, by internal reflections in ice crystals, and by multiple scattering. Molecular scattering has a small depolarization component associated with it which is caused by anisotropy of the polarizability of molecules. The depolarization due to air molecules has been estimated to be 0.035 [Zuev, 1976]. However, the depolarization due to irregular particles and ice crystals can vary as high as 0.7 [Sassen, 1991]. The ratio thus provides a convenient way of determining the water phase in clouds. Additionally, the volume scanning results in a three-dimensional volume map of the water phase state in the cloud. These measurements will be useful in the study of cloud formation, cloud maintenance, and cloud physics.

Other researchers have had success in using the backscattered lidar signals at three different wavelengths to determine the particle size distribution. However, this method is based on spherical particles which can be described by the Mie scattering theory. If the

particles are not spherical, results suggest that there can be a significant error in the particle size distribution. However, if depolarization detection is employed, nonspherical particles can be detected, and the algorithm can be successfully corrected [Flescia, 1991].

The third scattering mechanism employed by the WAVE-LARS instrument is Raman scattering. Raman scattering describes an inelastic optical scattering process where the scattered light is frequency shifted from the incident light. This frequency shift is caused by the different energy levels of the vibrational and rotational states of a molecule. These effects can be used to determine the concentrations of various atmospheric constituents and to determine temperature profiles of the atmosphere. The difficulty with the technique is that the effective scattering cross section for this process is typically three orders of magnitude less than the unshifted Cabannes line.

The vibrational Raman scattering effect allows the unique identification of the scattering molecule. When a photon is incident upon a molecule's electron cloud, it excites an electron to a higher virtual energy level. The electron can only exist at this level for a short time before it will return to a lower level, usually the ground state, and emit a scattered photon. If this electron decays to an energy level different from the original level, such as one of the several vibrational states of the molecule, the scattered photon will have a frequency different from the incident photon. The size of this frequency shift is unique to each molecule. This process is represented in Figure 1.2.

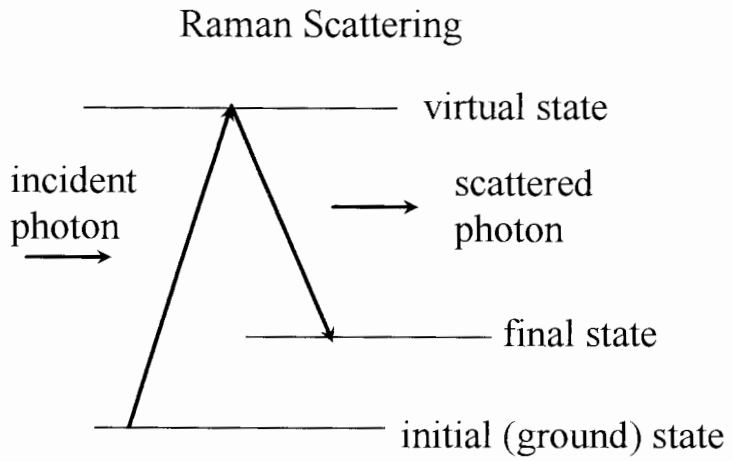

Figure 1.2. Schematic representation of Raman vibrational Stokes Scattering.

By measuring the amount of light scattered at the frequency shifted wavelength corresponding to a particular molecule, a lidar can determine the concentration of that molecule in the scattering volume. Usually the signal for a species, such as water vapor, is measured simultaneously with the signal for molecular nitrogen. The concentration of nitrogen is uniform throughout the lowest 90 km of the Earth's atmosphere, so the ratio of the signal corresponding to the unknown species to that of the nitrogen signal provides a direct measure of the unknown species' concentration.

1.4 The Lidar Equation

A lidar systems performance can be described by the lidar equation,

$$P(\lambda,R)=P_L \cdot \frac{A_0}{R^2} \cdot \xi(R) \cdot \xi(\lambda) \cdot \beta(\lambda_L,\lambda,R)\frac{c\tau}{2} \cdot \exp\left(-\int_0^R \kappa(r)dr\right), \qquad (1.3)$$

(1) (2) (3) (4) (5) (6) (7)

where the terms are described below:

(1) Power of scattered light

(2) Power of a transmitted laser pulse

(3) Effective reception area of the receiver

(4) Geometric form factor of the receiving telescope

(5) Efficiency of the optical detection system

(6) Number of scattered photons in the scattering volume

(7) The transmission through the atmosphere

This equation is used in predicting the expected returns from a proposed lidar system and is useful in calculating expected dynamic ranges and signal levels.

1.5 Description of WAVE-LARS Components

The WAVE-LARS system consists of three separate remote sensing systems, a volume scanning lidar, a volume scanning radar, and a vertically pointing lidar. The volume scanning units will be housed in one standard 8 ft x 20 ft x 8 ft shipping container that functions as a rugged transportable field laboratory. The fixed vertically pointing lidar is the

LAMP system housed in a second shipping container. The volume scanning units will provide particle size and shape distributions, while the LAMP system can provide atmospheric water vapor and temperature profiles.

The LAMP lidar determines atmospheric water vapor profiles by monitoring the vibrational Raman shifted wavelength for water vapor molecules and nitrogen molecules from the transmitted laser wavelengths. The water vapor profile is determined by taking the ratio of the water vapor return signal to the nitrogen return signal. The amount of nitrogen is a fixed percentage in the troposphere so this ratio gives the relative percentage of water vapor in the atmosphere. The LAMP system can be used to measure water vapor profiles in the atmosphere below cloud layers, thus providing important information for better understanding cloud formation and sustenance. The LAMP system can also provide temperature profiles by using the rotational Raman technique described in the previous section.

The volume scanning radar is a new 94 GHz cloud radar constructed by the Penn State Meteorology Department following a previous design which is currently in operation. This three mm wavelength radar can sense liquid cloud droplets and determine the extent and structure of a cloud. The new volume scanning radar will also measure the depolarization of the return caused by ice crystals in the cloud. This information should allow the determination of the ice mass content in the cloud when crystals are present.

In addition, these instruments will be supplemented with data from other meteorological sensors such as radiosondes. All of this information, in addition to the information collected by the volume scanning lidar, can be collected through a local area

computer network and integrated on a workstation computer. The emphasis of this thesis is on the control and data acquisition subsystem of the volume scanning lidar, and a detailed description of the volume scanning lidar is given below.

A lidar consists of five principle components, the transmitter, receiver, detector, data system, and safety system. The transmitter, receiver, and safety radar of this system are mounted on a volume scanning platform which is shown in Figure 1.4. The optical table and its supporting hardware is referred to as the volume scanning mechanism (VSM).

The transmitter section of the volume scanning lidar consists of a Continuum Surelight II-20 model ND:YAG laser. This laser is Q-switched at 20 Hz and emits 500 mJ of energy in a 4-7 ns pulse of linearly polarized light at the fundamental wavelength of 1064 nm. Non-linear crystals are used to produce the second and third harmonics of this fundamental at 532 nm and 355 nm. This laser was chosen for its high powered output, its small size, and its ruggedness. An off axis parabolic beam expander expands the laser beam five times to reduce the power density and to decrease the beam divergence.

The receiver system consists of a 16.5 in. (0.4 m) Richey-Chretien telescope and a receiver box that houses a calcite prism to split the received light into its two polarizations. The receiver box also houses the optics necessary to focus the received light into two optical fibers. This type of telescope was chosen because of its compact size which reduces the loads on the scanning mechanism and because of its polarization maintaining characteristics.

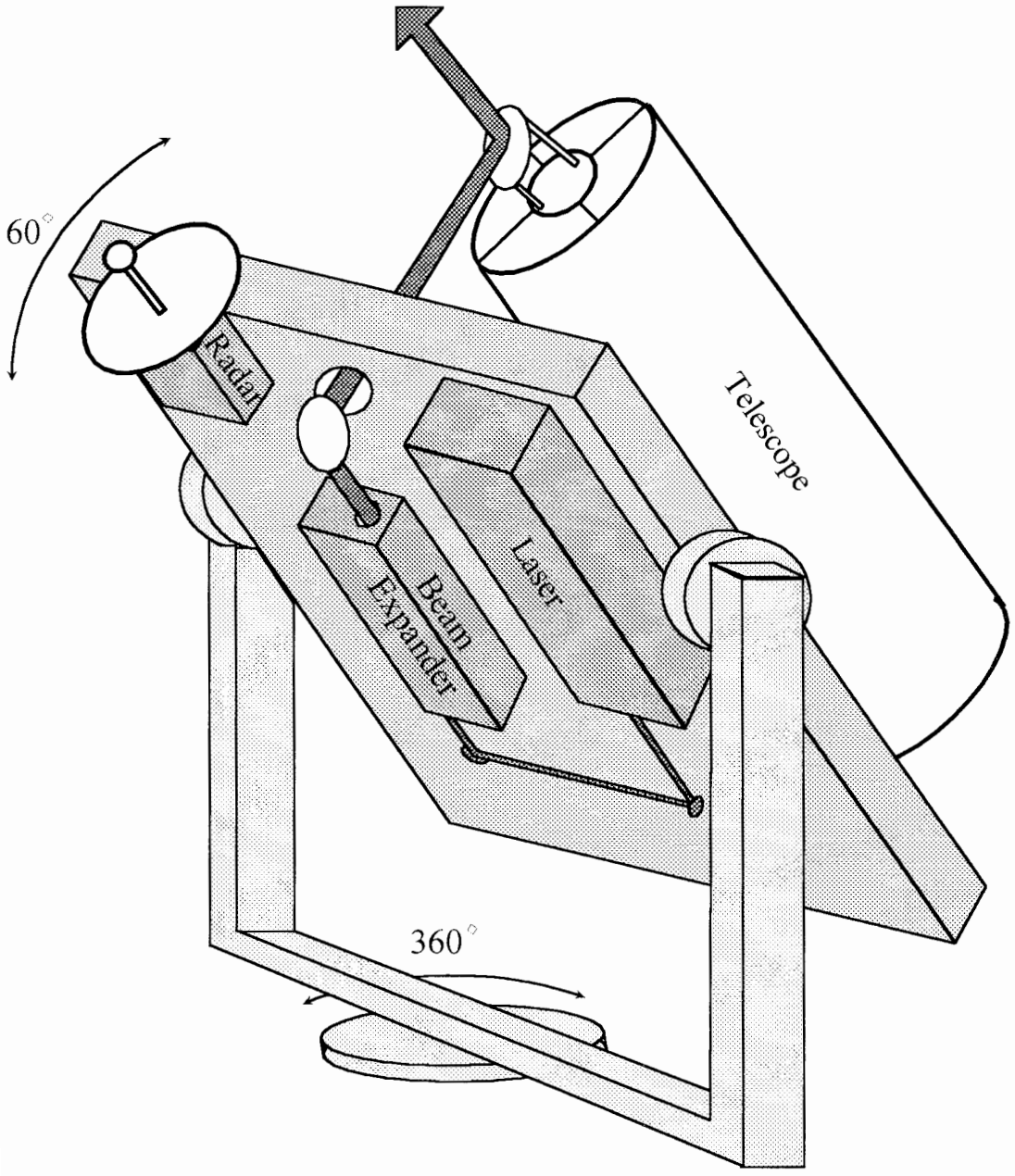

Figure 1.4. A diagram of the Volume Scanning Lidar [prepared by G. R. Evanisko].

Two 1 mm diameter optical fibers guide the received light signal from the receiver to a detection system housed in an instrument rack [Mathason, 1994]. This detector system uses beam splitters and narrow bandwidth interference filters to separate the returned signal by wavelength. The 355 nm and 532 nm signals of both polarizations are detected by photomultiplier tubes (PMT's). The 607 nm Raman shifted nitrogen channel is monitored using a PMT optimized for photon counting because of the weaker signal expected from a Raman channel. The 1064 nm infrared channels are detected using PIN photodiodes.

The data acquisition system has six analog detection channels with 12 bit resolution sampling at a rate of 20 MHz. This sampling rate results in a vertical range resolution of 7.5 m. In addition, the 607 nm channel data is acquired in the photon counting mode with a 300 MHz count rate, with a 100 ns dwell time resulting in a range resolution of 15 m. The data is collected and stored by a rack mounted personal computer. This computer is interfaced with a custom timing and control unit which coordinates the functions of the transmitter, the data acquisition system, and the scanning mechanism.

The safety system for the scanning lidar consists of interlocks, kill switches, and an airplane detection radar. This airplane detection radar is constructed from a modified marine radar and will automatically suspend laser operation if an airplane is in danger of intercepting the transmitted laser beam. This will prevent the possibility of an observer on an aircraft or on the ground from directly viewing the laser beam or a specular reflection of the laser beam, both of which would pose a serious optical hazard.

The VSM consists of two identical custom designed scanning platforms and control systems to allow simultaneous scanning of the polarization lidar and radar. The scanning

platforms were designed to support either the lidar or radar at scanning rates up to 10 degrees per second with a 1 second reversal time. The design of this structure has been detailed elsewhere [Manning, 1993]. The control system for the VSM consists of a computer interfaced controller for each axis. The controllers pulse the stepper motors at each axis to achieve the desired scan profile. This profile can be changed through the software interface to the motor control computer. This computer is also responsible for saving the position information of both scanning platforms as a function of time. This is essential so that a three dimensional map of the collected data from the scanning lidar and radar can be constructed. The design of the motor control system has been previously described [Anuskiewicz, 1993].

1.6 Outline of Thesis

This thesis will detail the process of specifying, designing, and testing a state of the art data acquisition and control subsystem for a volume scanning lidar. The thesis begins with a description of basic lidar principles and the scattering mechanisms involved in lidar measurements. It continues with a discussion of the types of signals expected to be acquired, and the parameters which describe these signals. Next, calculations of signal levels and data rates are presented along with an evaluation of the applicable technologies to achieve the scientific goals. The design of the overall system is then presented with an evaluation of its actual performance and some sample data. The thesis will conclude with a discussion of how successfully the resulting system met the design goals and recommendations for improving the system and suggestions for future research.

Chapter 2

Data Acquisition Fundamentals

Before designing a system to collect and analyze data from an instrument it is imperative to carefully consider what types of signals need to be acquired and the parameters that describe these signals. Some of these parameters include the source of the signal, the strength of the signal, and the noise present in the signal. It is also important to define how accurately and precisely these signals are to be measured. This chapter discusses the types of signals detected by the volume scanning lidar and the methods of acquiring them. This is followed by a discussion of the strengths and limitation of each method. Finally, some of the signal processing performed on the data is discussed.

2.1 Types of lidar signals

The purpose of the system described in this thesis is to sample the electrical signals generated by the volume scanning lidar's optical detectors in such a way as to obtain the strength of the lidar's return signal as a function of altitude. The electrical signal that is sampled by the data acquisition system, S_s is actually the superposition of the desired signal S_a, and noise terms contributed by the background solar radiation, N_b , the optical detector, N_d, and electrical noise, N_e, coupled into the system and generated by the data acquisition system's components. The sampled signal is given by the equation,

$$S_s = S_a + N_b + N_d + N_e \quad . \tag{2.1}$$

An important feature of a lidar signal is that the returned signal is sampled during short time intervals. This is commonly referred to as range gating. By knowing the time when each sample is taken relative to the time when the laser pulse was transmitted, one can easily determine at what range the backscattered radiation originated. The range, R, can be calculated using the equation,

$$R = \frac{1}{2} c T_s \quad , \tag{2.2}$$

where c is the speed of light and T_s is the sampling interval. Repeating this process results in a series of range gates, each representing the return signal at some distance from the transmitter. Notice the factor of 1/2 in Equation 2.1 which accounts for the transit time out and back for the laser pulse. When employing analog to digital conversion methods the sampling interval is the inverse of the sampling frequency. When employing photon counting methods the sampling interval is the dwell time, or the period of time during which the PMT pulses are counted.

There are many types of lidar signals that may need to be acquired depending on the particular lidar technique employed. In this application, there will be return signals generated by molecular scattering, aerosol scattering, and Raman scattering. Each of these scattering mechanisms result in return signals with different characteristics. The backscattered returns from molecular and aerosol scattering can cover a dynamic range of five orders of magnitude.

The large scattering cross section of molecules and aerosols at optical wavelengths leads to relatively large signal voltages from the detector system. On the other hand, the scattering cross sections for Raman scattering processes are much smaller. During each detection interval, or range gate, a Raman channel will generate several short electrical pulses which correspond to detection of individual photons. These pulses must then be counted to determine the strength of the returned signal.

2.2 Methods of Data Acquisition

Each of these types of signals lends itself to a particular mode of data taking. Analog to digital conversion (A/D) and photon counting (PC) are the two methods that we use to acquire the backscattered signal. The A/D method is particularly suited to strong return signals requiring large dynamic ranges. This method is used to measure the combined molecular and aerosol backscatter signals collected with the volume scanning lidar. The large backscattered signal from these particles will result in a large number of current pulses from the optical detectors. There will be so many pulses that it will be impossible to resolve individual pulses and count them. Therefore, the individual pulses are integrated using a special filter to form a voltage signal that represents the amplitude of the returned signal. This signal is then sampled using a high resolution A/D converter with a very high sampling rate. This method can not be used with weaker signals because it is not capable of detecting individual photons.

The photon counting method is preferred when sensitivity for weak signals is needed. A weak signal will result in a train of current pulses from the detector corresponding to the

detection of individual photons. A device called a discriminator is used to detect the current pulses and convert them to a standard logic level pulse which can be digitally counted. The discriminator has a threshold level that determines the current level which constitutes the detection of a photon. The logic pulses from the discriminator arriving during each range interval are counted and the resultant sum is proportional to the amplitude of the return signal.

2.3 Quantifying Performance

Both the A/D and PC data acquisition techniques need to be quantified in terms of their performance. There are several methods for quantifying the performance of an A/D system designed to digitize transient signals at a very high sampling rate. The dynamic performance of such a system is critically important. Two important measures of the dynamic performance are the signal to noise ratio, SNR, and the total harmonic distortion, THD. The SNR indicates the dynamic range of a system, and the THD indicates the level of nonlinearity present in the system. The SNR_{rms} of a data acquisition system is the ratio of the rms signal amplitude and the rms noise amplitude. The noise is contributed by the quatization noise generated by the sampling process, noise sources within the optical detectors, noise sources within other components of the system, such as amplifiers and filters, system nonlinearities, and noise from outside sources coupled to the system either radiatively or directly. The SNR_{rms} is intended to be a measure of the data acquisition system performance and does not include noise associated with solar, lunar, or stellar background radiation.

An effective approach to measuring the SNR and the THD is to input into the system

a sinusoidal signal in the frequency range of the actual lidar signals. The sine wave should cover the full input scale of the A/D converter. The data acquisition system can sample the sinusoidal signal and then perform an FFT of the signal. The SNR can be determined from the FFT spectrum using,

$$SNR = 20\log_{10}\left[\frac{RMS\ SIGNAL\ LEVEL}{RMS\ NOISE\ LEVEL}\right]. \tag{2.3}$$

The THD of the system can also be determined from the FFT spectrum. Any nonlinearities in the system will result in unwanted harmonics of the fundamental sinusoidal input. The THD is given by,

$$THD = 20\log_{10}\left[\frac{RMS\ SIGNAL\ LEVEL}{RMS\ HARMONIC\ LEVEL}\right], \tag{2.4}$$

with the RMS harmonic level determined by the RMS value of the first five harmonics of the sinusoidal test signal.

There is an important theoretical relationship between the number of bits in the A/D converter output code and the converter's SNR. The process of quantizing an analog signal and converting it to a digital representation has an inherent uncertainty or noise level associated with it. The SNR of an A/D converter, when considering only quatization noise, is given by,

$$SNR = 6.02N + 1.8dB\ . \tag{2.5}$$

This equation shows that using a converter with more bits will directly increase the SNR and thus increase the usable dynamic range. This relationship is useful in selecting the necessary A/D output code length for a data acquisition system based on the expected dynamic range [Sheingold, 1986].

It is also important to quantify the performance of a photon counting system. Some of the factors that affect the performance of such systems are the random arrival of photon pulses and the random distribution of their pulse heights. This randomness reduces the number of pulses that can be counted during a dwell time and can lead to non linearities in the signal. This occurs because as the incoming photon rate increases, the resultant current pulses begin to overlap and are counted as one pulse. This decreases the observed count rate from the true rate. In the worst case, the photon arrival rate will be so high that all the current pulses will overlap to form only one pulse which can be counted. For a photon counting system with uniform pulses and a photon counting dead time, τ_d, the true count rate, N, can be related to the observed count rate, S, with,

$$N = S \cdot \exp(-S\tau_d) \ . \tag{2.6}$$

If the discriminator level is not properly set, then the distortion due to this effect can be even worse [Donovan, et. al., 1993]. However, the discriminator settings also plays a role in the SNR of a photon counting system. The discriminator thresholds in a photon counting system are typically set just above the noise level so only signals corresponding to a photon arrival are detected and counted. If the threshold is set too high, some photons will not be counted and the dynamic range of the photon counting system is diminished. However, this process

can compete with the non linear effect so great care must be used in setting the discriminator level to optimize overall system performance.

2.4 Signal Averaging

The SNR of a system which takes repetitive measurements of a stationary signal can be significantly improved by averaging several samples of the same signal. This improvement is proportional to the square root of the number of signals averaged. This relation is given as,

$$SNR_{averaged} = SNR\sqrt{N} \quad . \tag{2.7}$$

The increase in SNR given by (2.7) is an important result for remote sensing applications. If the atmospheric conditions being studied vary little during a data acquisition interval, the data accumulated during this interval can be averaged resulting in an increase in the effective SNR. This enables the remote sensing system to observe signal details too weak to be observed with one signal return. This technique will be used in the data acquisition system for the volume scanning lidar and will be important in determining the speacifications for the final system design. The following chapter details the final system speacifications and discusses all of the factors involved in their determination.

Chapter 3

System Specifications

The design of the volume scanning lidar's data acquisition system began by carefully determining the system specifications. This chapter discusses the process of selecting these specifications. Most electrical or electronic systems can be described by a combination of functions that operate on a set of input signals to generate the desired output. In this case, the input signals are the backscattered lidar returns from the volume scanning lidar's detectors, and the desired output is a digital representation of the size and shape of aerosols and cloud particles in the three dimensional region of space being scanned. The specifications are determined by considering the characteristics of the expected input signals and the technology available to implement the system functions necessary to achieve the desired outputs.

This chapter begins with a discussion of the expected input signals. Data from other lidar systems such as the LAMP lidar and theoretical simulations of system performance provided the required information to calculate the expected properties of system inputs. Exact specifications for the output data are given by considering the scientific goals of the volume scanning lidar. In addition, the chapter considers some technological constraints imposed on the system by rapidly collecting a continuous stream of data. The chapter concludes with a listing of the design goals that were established for the completed system.

3.1 Expected Input Signals

The volume scanning lidar has seven data channels, including three pairs of channels to measure the depolarization ratio at the transmitted wavelengths of 1064 nm, 532 nm, and 355 nm. Each pair consists of one channel to monitor the backscattered light polarized parallel to the polarization of the incident laser light and one channel to monitor the backscattered light polarized perpendicular to the polarization of the incident laser light. In addition, there is one channel used to measure the molecular nitrogen concentration from a vibrational Raman scattering signal. The dynamic range and signal strength of the backscattered signal will be different for all seven channels. The signal characteristics will also change as the atmospheric conditions vary. For example, a low altitude cloud can cause a huge near field return and expand the dynamic range that must be covered, so that weaker far field returns can still be detected.

A system intended to conduct lidar measurements of aerosol scattering from 100 m to 20 km, requires a dynamic range of approximately five orders of magnitude [Measures, 1984]. For each pair of the depolarization channels, the parallel channel will receive signals with a dynamic range of this magnitude. The perpendicular channel will only have large return signals from altitude regions containing polarizing particles such as irregularly shaped ice or aerosol particles in clouds, or due to the depolarization of the scattered signal from multiple scattering from cloud droplets. However, clouds can cause a return signal that varies two or more orders of magnitude and clouds can have a depolarization ratio as high as 0.7. Therefore, the signals on the perpendicular channel can also cover a dynamic range of two or more orders of magnitude. For these channels, the signal strength will be too

strong to consider using the photon counting technique. This fact, along with the large expected dynamic range, requires the selection of the A/D acquisition method. This conclusion will be verified below.

The nitrogen channel of the volume scanning lidar is intended to provide information which can be used in optical extinction correction algorithms [Rau, 1994]. This channel will also have a dynamic range which can cover several orders of magnitude in measurements taken from the ground to 20 km. However, the scattering cross section of Raman scattering is approximately one thousand times less than the molecular scattering cross section at the same wavelength. Therefore the return signal for the Raman scattering channel will be too small to effectively digitize the signal under most measurement conditions. Thus the photon counting method has been selected for the nitrogen channel, with signal averaging employed to increase the signal to noise ratio and the dynamic range.

An effort was made to verify the assumptions made in the previous section on signal strength and dynamic range under a variety of atmospheric conditions. This was accomplished by examining sample data from the LAMP system and by conducting computer simulations.

The results from the computer simulations are presented in Figure 3.1, and clearly illustrate the dynamic range and signal level expected for each of the volume scanning lidar's channels on a clear night. The parameters that were included in the simulation are the optical efficiencies of the lidar transmitter and receiver, the quantum efficiency of the detectors, and the form factor of the telescope. Figure 3.1 shows that the return signals from 355 nm, 532 nm, and 1064 nm cover five orders of magnitude, even without the presence of aerosol

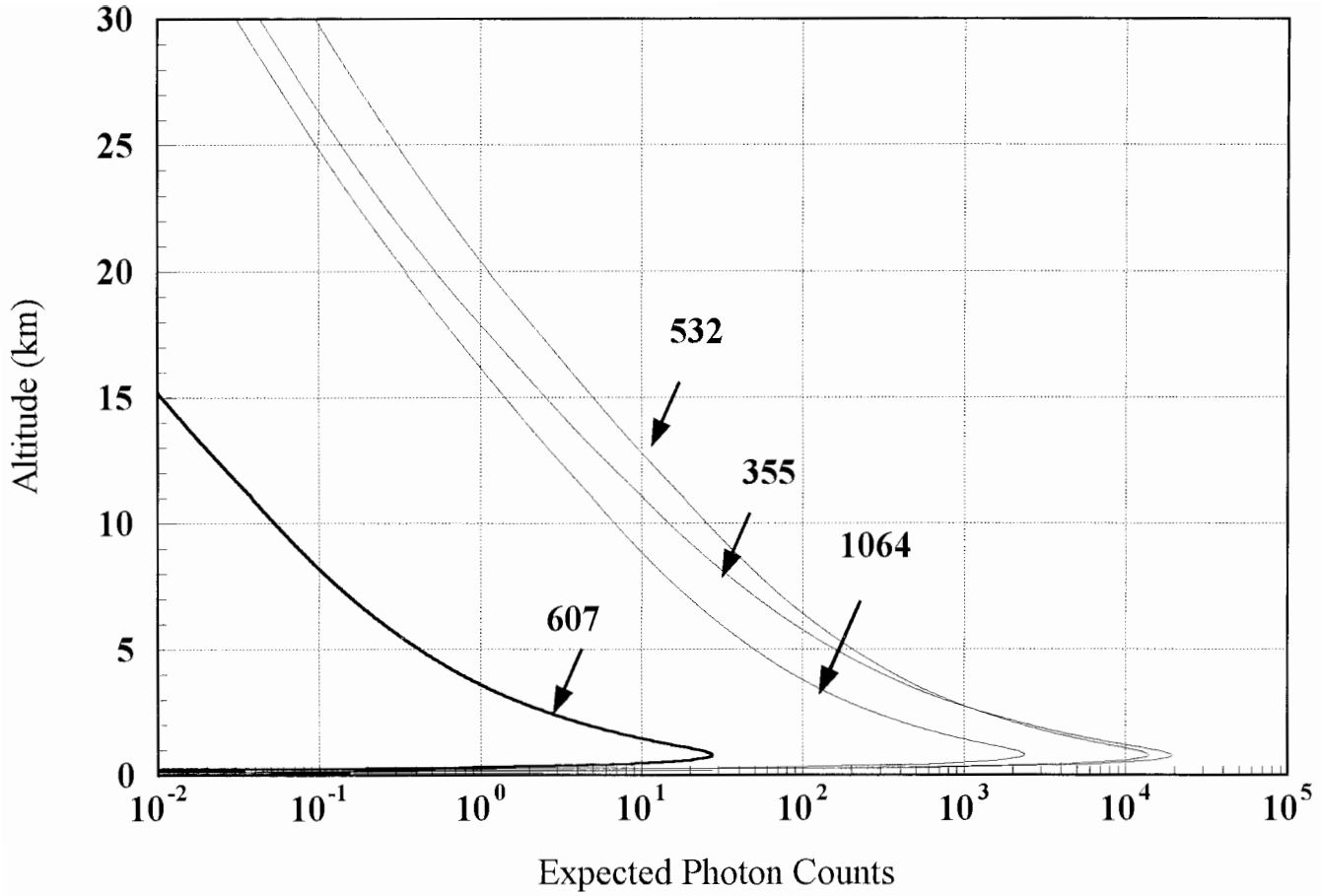

Figure 3.1. Expected photon counts for the Raman nitogen channel and the parallel polarized channels of the volume scanning lidar [Mathason, 1994].

particles. Also the simulation indicates that there will be approximately 20 counts per bin maximum for the nitrogen channel. This is an acceptable signal level if one chooses to use the photon counting method. Similarly Figure 3.1 indicates that with a maximum count rate of 10,000 counts per bin, the A/D method is the preferable acquisition method for the 355 nm, 532 nm, and 1064 nm channel pairs.

The computer simulation was conducted only for a clear atmosphere and does not predict what will occur if an aerosol layer or cloud is present. Figure 3.2 shows some data that was obtained with the LAMP lidar system. This figure shows that a cloud layer can increase the signal return by two orders of magnitude. This is important because it indicates that the perpendicular polarized channels at 355 nm, 532 nm, and 1064 nm could have a signal range of this order if the cloud has a large ice crystal content.

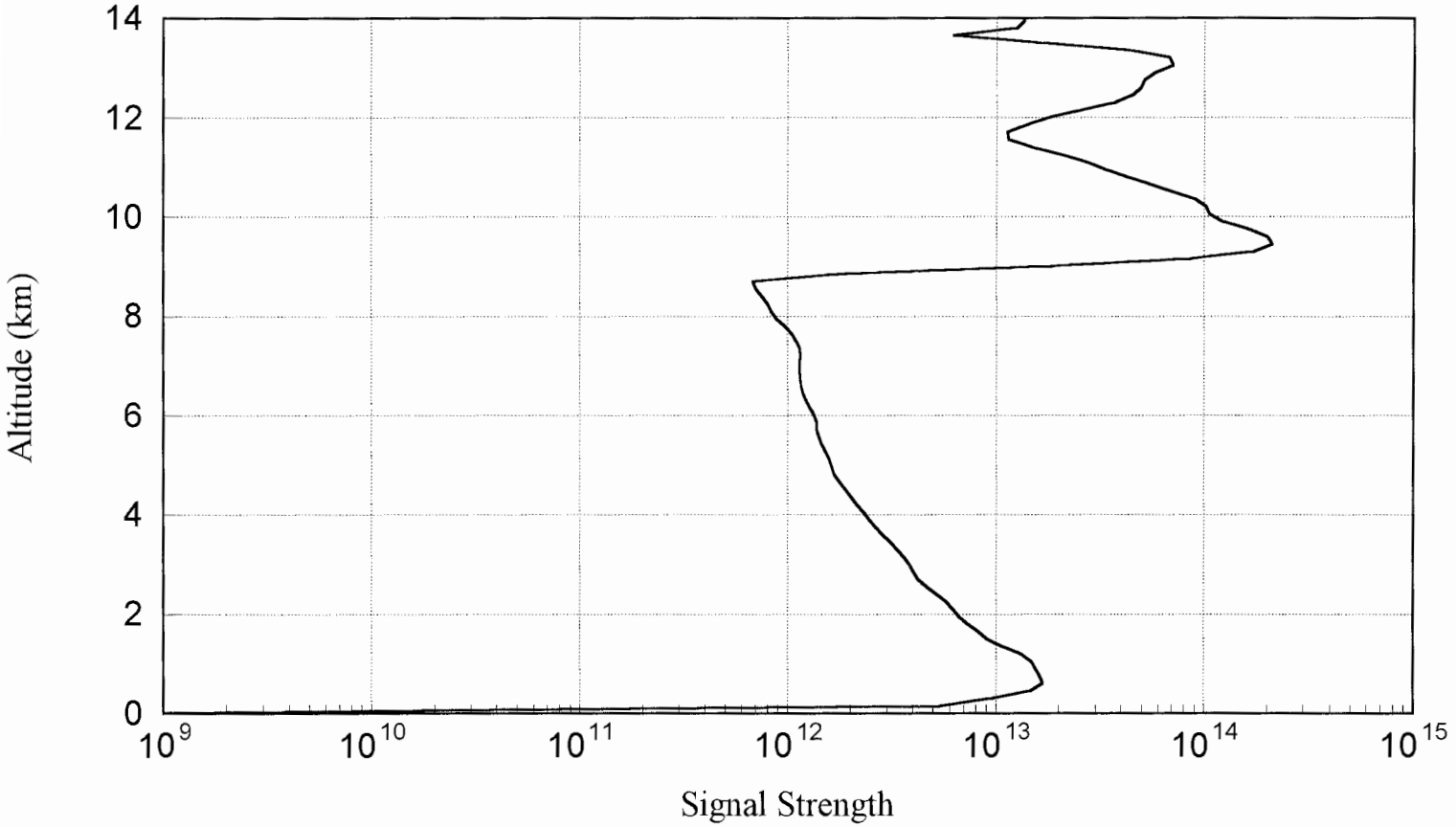

Figure 3.2. Signal return from the 532 nm channel of the LAMP lidar. The profile, showing two clouds, was obtained as a 30 minute average 10-3-91, during the Ladimas Campaign.

3.2 Desired System Outputs

The desired output from the volume scanning lidar is a numerical representation of the size and shape distributions of aerosol and cloud particles in a region of the atmosphere. The VSL is capable of scanning the entire surrounding atmospheric region from the zenith down to 30° above the horizon. With a maximum 20 km line of sight range, the system will be able to map the size and shape distributions in a conical volume with a minimum height of 10 km and a radial distance of about 17 km. This is required if the system is to have the capability to map the majority of the troposphere. The size and shape distributions for the aerosol and cloud particles in this region will be inferred from the backscatter and depolarization data collected at the three transmitted wavelengths using the methods described in section 1.3.

A channel that measures the nitrogen density using the Raman scattering lidar technique is also included. The nitrogen channel is used to aid in system calibration using the extinction correction techniques mentioned above.

A data acquisition session with the volume scanning lidar commences when the instrument begins scanning through a predetermined region of the atmosphere. Throughout the scan, data is collected simultaneously on all seven channels. The data acquisition system converts the returned signal strength at the various ranges along the line of sight to a digital value and saves the information along with the time it was taken. The raw data from each channel is saved as a function of time in a standard ASCII output. Each file contains the returned signal strength as a function of range and also includes a header containing the time that particular profile was taken and other pertinent system information such as laser power.

By coordinating the activities of the VSM and the data acquisition system, the position of the returned signal can be determined.

The data collected from the seven input channels is transferred to a processing workstation via a computer network and combined with the data from the volume scanning radar and LAMP. The data can then be processed and integrated to provide an accurate representation of the composition and dynamics of the atmospheric region scanned. This representation will hopefully provide the insight needed to achieve the system's scientific goals.

3.3 System Constraints

There are several hardware constraints which will limit the performance of some key parameters of the data acquisition system. These parameters include the dynamic range and the spatial resolution of each channel. First, the dynamic range of the system is limited by the SNR of the data acquisition modules. The SNR of the A/D mode is determined by the number of bits in the converter's output code, with typical code lengths of 8 to 16 bits. The SNR of the photon counting mode is determined by the maximum count rate of the module, with typical count rates of 100 MHz to 300 MHz.

A second important parameter to consider is the spatial resolution of the system. There are several factors which limit the resolution that can be achieved, including the sampling rate of the data acquisition modules and the laser pulse repetition frequency. The transmitter of the volume scanning lidar has a constant pulse repetition frequency of 20 Hz. If 20 laser shots are averaged to improve the SNR while the volume scanning platform is in

motion, at its maximum scan rate of 10 degrees/second, the resultant data will have a higher SNR, but will correspond to a 10 degree region of the scan volume. To increase the spatial resolution the scan rate would have to be decreased. This is sometimes not desirable if atmospheric conditions are rapidly varying. Some degree of flexibility in the system parameters is therefore required to tailor system performance to specific atmospheric conditions. The spatial resolution of the system is also determined by the minimum sampling interval of the data acquisition modules. However, there is usually a trade-off the between the minimum sampling interval, the dynamic range, and the cost of the module.

3.4 Formulation of Design Goals

The hardware constraints of the previous section will now be used, along with the expected system inputs and required system outputs, to formulate the design goals for the volume scanning lidar's data acquisition and control system.

First, there will be three pairs of channels to measure the depolarization ratio at the three transmitted laser frequencies. As mentioned in section 3.1, these channels will have a maximum dynamic range of three to five orders of magnitude. Under most atmospheric conditions a dynamic range of 80 dB, or four orders of magnitude, will be sufficient. The polarization channels are well suited for the A/D technique of data acquisition because of this large dynamic range. The first parameter that must be determined for these channels is the number of bits in the A/D converter output code. Using (2.3) and (2.5) the resulting SNR for three converter sizes and different number of averaged laser shots can be calculated. The results of these calculations are shown in Table 3.1.

Table 3.1. Signal to Noise Ratios for different A/D converter sizes and number of shots averaged

Converter Size	1 shot	20 shots	1200 shots
8-bit	49.96 dB	62.97 dB	80.75 dB
12-bit	74.04 dB	87.0 dB	104 dB
16-bit	98.12 dB	111 dB	128 dB

All three converter sizes are ultimately capable of achieving the required 80 dB SNR. However, 16 bit converters are not available at the sampling rates necessary to achieve adequate spatial resolution. The 8 bit converters require an averaging time of one minute before they can achieve an 80 dB SNR. Thus, the 12 bit A/D converter, with a minimum of a one second averaging time, is ideal for this application.

The discussion in section 3.1. indicated that the Raman nitrogen channel of the volume scanning lidar will require the photon counting method of data acquisition. Spatial resolution is of greatest interest for this channel. Therefore, the photon counting system with the shortest dwell time will be selected for the nitrogen channel.

Another important goal for the data acquisition system is to utilize every laser shot that is transmitted. In order to meet this goal, the data system architecture must be configured in a manner that the data collected by the system can transfer through its various components in the time between laser shots. For a 20 Hz laser this time is 50 ms. Otherwise, the data will begin to pile up, and the system could possibly crash. This is the most critical design goal, and the most difficult to achieve.

In addition, the system must be rugged enough to withstand the variable and harsh

conditions that are usually encountered during field campaigns. Also, programmability of system parameters should be designed into the system so unexpected operating conditions can be handled. Finally, the system must be constructed with a total hardware cost under $50,000. These design goals are summarized in the following list.

1. 6 A/D channels - 80 dB dynamic range with 7.5 to 15 m range resolution
   - 10 to 20 MHz 12 bit converters
2. 1 photon counting channel - 15 to 75 m range resolution
   - 100 to 500 ns dwell times
3. Minimum averaging time of 1 second
4. Capability to capture data from every laser shot
5. Rugged enough for field campaigns
6. Computer control of system parameters for flexibility
7. Total hardware cost of less than $50,000

These specifications will be used in the next chapter in designing the data acquisition and control system for the volume scanning lidar.

Chapter 4

Design of Data Acquisition and Control System

This chapter describes the design of the data acquisition and control system for the volume scanning lidar that will achieve the design goals listed in Chapter 3. First, the overall system architecture is described. This is followed with a description and evaluation of the major components of the system. The next two sections detail the system hardware and software.

4.1 System Architecture

Figure 4.1 shows a block diagram of the WAVE-LARS system architecture. The diagram includes the data acquisition and control systems for the volume scanning radar, the volume scanning lidar, and LAMP. It also shows the scan control system for the volume scanning platform. Each system can operate independently of the others. However, each system has the capability to communicate to the other systems so that coordinated measurements are possible. The intersystem communication is provided by local area network, and direct signal connections to the scanning platform motor control system. The local area network provides for the transfer of information to a central data processing workstation. Here the information from the scanning lidar, scanning radar, and LAMP can be integrated and analyzed. In addition, data from other instruments can be added to the WAVE-LARS data through the network. This is illustrated in Figure 4.1 with an auxiliary instrument block. The direct signal connections to the motor control unit of the scanning

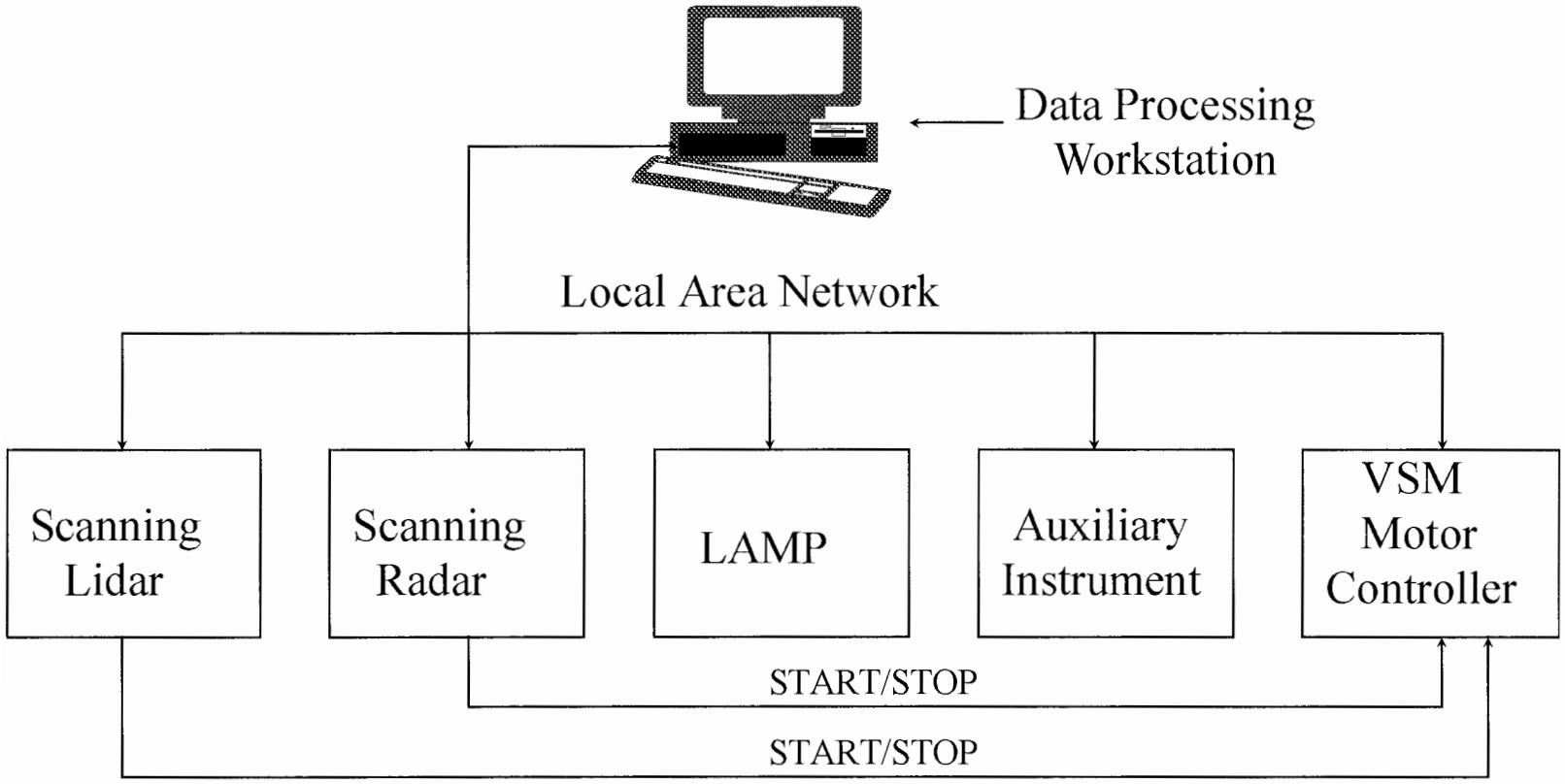

Figure 4.1. WAVE-LARS block diagram.

platform are provided so that the platform can be started and stopped with precision timing signals originating from the individual control systems of the scanning lidar and scanning radar. The individual volume scanning mechanisms for the scanning lidar and the scanning radar can be moved simultaneously in an identical scan pattern, or they can be moved independently.

The block diagram of the volume scanning lidar's data acquisition and control system is shown in Figure 4.2. This figure shows that the signals detected with the photodetectors are sent to data acquisition subsystems located in a rugged rack mounted personal computer designated for the data acquisition system. The timing and control signals for the data acquisition units are generated by a custom designed timing and control unit (TCU). The timing and control unit also controls the laser and the motor controls for the volume scanning lidar platform. The timing and control unit is interfaced to the data acquisition computer so the system operator can control the entire system from the computer terminal.

Some important terms must be defined before describing the individual components of the data acquisition system.

Photon Counting Unit:

Acquires data by counting the number of photon pulses occurring during a dwell period. Also known as a digital channel or a multi channel scaler.

Dwell Period:

Period of time that represents one range bin in the photon counting units.

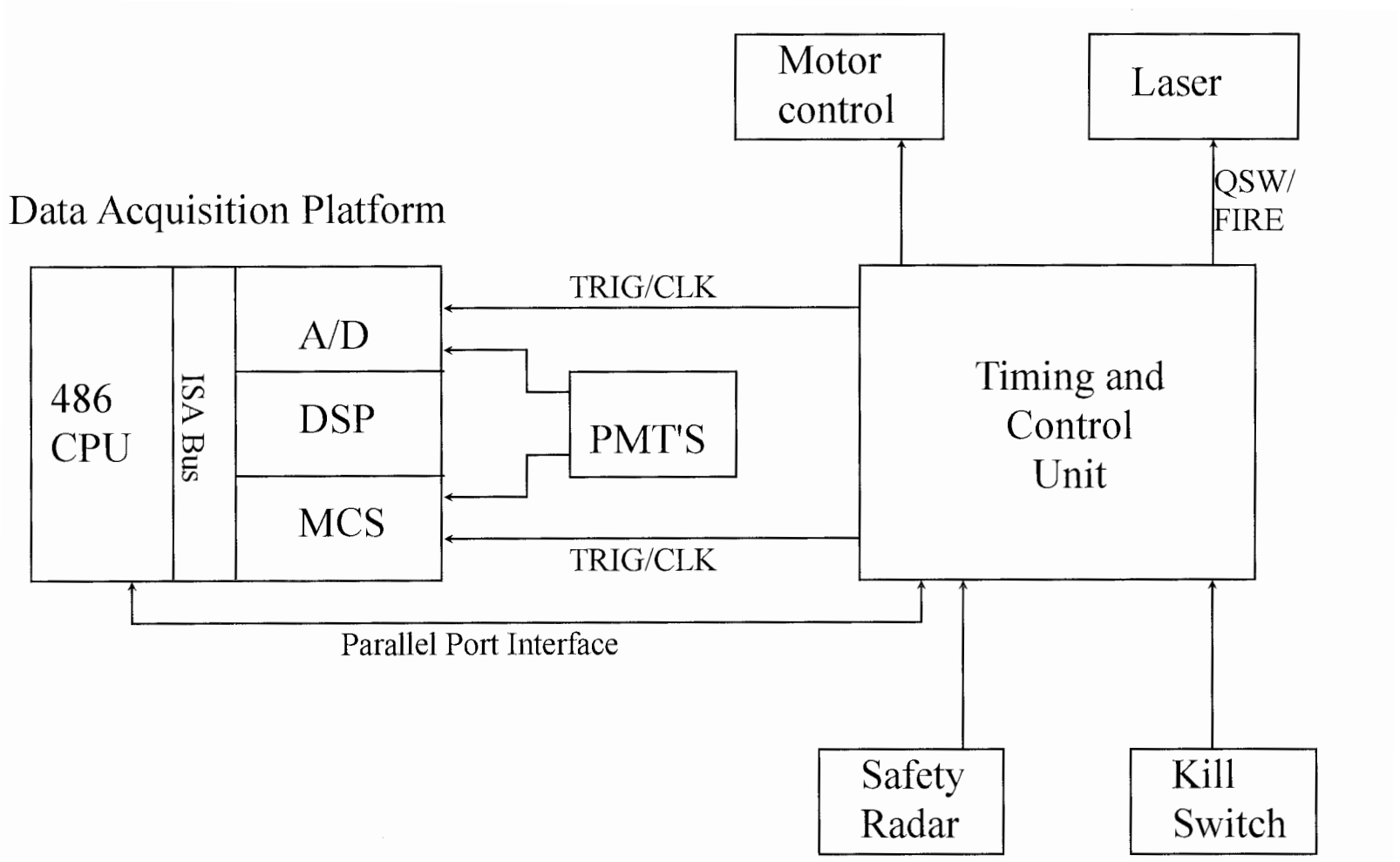

Figure 4.2. Block diagram of the volume scanning lidar data acquisition and control system.

Digitizing channel:

Data is acquired by converting the voltage signal generated by one of the aerosol backscatter channels into a digital value. Sometimes referred to as an analog channel or a transient recorder.

Single shot of data:

Data profile taken of one laser shot along the volume scanning lidar's current line of sight.

Single segment of data:

Collection of single shot data averaged by the data acquisition units. Typically consists of 20 to 100 shots taken during a 1 to 5 second time interval. The position of the segment is determined by the position of the scanning platform at the time the data was acquired.

Single Scan of Data:

Collection of 100 to 300 segments combined with the scanning platform position data saved by the motor control computer which describes the atmospheric volume scanned during the scan period.

4.2 Evaluation of Major Components

The major components of the data acquisition system include the data acquisition platform, the data acquisition modules, the timing and control unit, and the data acquisition computer. A data acquisition platform provides power, communication, and an enclosure for the individual modules of a data acquisition system. Provision is usually made for some type of computer interfacing to the acquisition modules for the purpose of data collection, data archiving, and computerized control of the acquisition modules. The platform can also support other types of auxiliary modules such as signal conditioners and specialized controllers. The data acquisition platform was specified first so that compatible digitizing and photon counting modules could be selected. Some of the determining factors in this decision were cost, data transfer speed, and availability of the necessary data acquisition modules. The three platform types considered for this project were a CAMAC, a VXI, and an ISA data bus.

The CAMAC (Computer Automated Measurement And Control) bus is defined by ANSI/IEEE Standard 583. This platform will house and interface up to 24 instrumentation modules in a special housing which is called a crate. The modules can communicate with each other and to a host control computer. Data is transferred from the modules through a 24 bit data bus clocked at 1 MHz, which is known as the dataway.

Individual modules are available for this platform that implement a local bus to adjacent modules, for very high data throughput applications. This is particularly well suited for lidar applications where bursts of data are acquired and must then be transferred to a data buffer, where signal averaging is performed. This data transfer must be completed before the next burst of data occurs. In this application the data burst will occur once every 50 ms,

which corresponds to the interval between transmitted laser pulses. The CAMAC dataway is not capable of transferring the data generated by the seven data channels of this system in this time period making the local bus a necessity. However, for permanent data storage, the data must still be transferred over the much slower dataway. Another disadvantage with this platform is the high cost of the modules. The modules are costly because they are highly specialized pieces of equipment generally used in nuclear physics experiments.

The VXI bus is a similar arrangement that provides a standardized platform and interface for independent instrumentation modules. This specification is an extension of the VME (Versa Module Europa) bus standard and the instrument modules are again housed in a standardized instrument crate. As a more recent standard it incorporates more features such as 32 bit data transfers at 20 Mbytes per second. It also includes local bus capability for adjacent modules in the platform. The drawbacks to this platform option are the large cost of the modules and the lack of a commercially available photon counting modules.

The third data acquisition platform option is a standard ISA (Industry Standard Association) backplane personal computer. This option eliminates the expense of purchasing an external instrumentation crate and computer interface. Also there exist several inexpensive modules for both digitizing and photon counting for the ISA platform. The type of ISA platform considered for the data system was a computer intended for use in harsh industrial environments. This type of computer is housed in a rugged enclosure that contains an ISA backplane with 10 to 20 slots for computer cards. Several manufactures offer single board IBM compatible computers that can be plugged into the passive backplane. In addition, the enclosure can house any of the other standard peripherals such as floppy disks drives, hard

drives, and network cards. A table which summarizes the features of the three platforms is given in Table 4.1.

Table 4.1. Features of possible data acquisition platforms

Platform	CAMAC	VXI	ISA
Bus Width	24 bits	32	8
Bus Data Transfer Rate	1 MHz	20 MHz	8 MHz
Local Bus yes/no	yes	yes	yes
A/D and PC Modules Available yes/no	yes	no	yes
Cost	$6,000	$10,000	$4,000

Before making the final selection of the data acquisition platform, it was necessary to evaluate the data acquisition modules that were available. Only commercially available modules were considered for this project. Table 4.2 summarizes the units that were evaluated for each platform. As you can see the CAMAC and VXI options were very expensive. Also there is no known photon counting solution for the VXI platform. For the ISA platform the Bittware Research module was the only A/D unit that could give the required specification of 12 bit resolution at 20 MHz. For photon counting , the MCS-100 was the best option both in terms of price and performance for the ISA platform. It was also superior to the CAMAC option for photon counting. The optimal configuration considering price and performance is a ISA based system, with a Bittware system for digitizing and a Santa Fe system for photon

counting.

Table 4.2. Manufacture specifications for data acquisition modules

Platform	Manufacturer	Part #	Type	Specifications	Cost Per Channel
CAMAC	DSP Tech.	2112	A/D	20 MHz 12 bits	$12,000
CAMAC	DSP Tech.	2190	MCS	100 MHz 2 μs	$7,000
VXI	Hewlett Packard	E1429	A/D	20 MHz 12 bits	$10,000
VXI	Racal-Dana	6651	A/D	20 MHz 12 bits	$10,000
ISA	Bittware Research	Custom	A/D	20 Mhz 12 bits	$4,000
ISA	Signatec	DSP100A	A/D	100 MHz 8 bits	$6,000
ISA	EG&G	MCS-plus	MCS	100 MHz 2 μs	$5,000
ISA	Santa Fe Energy Research	MCS-100	MCS	300 Mhz 100 ns	$5,000

Some type of computer system was needed for the data system. This computer performs several functions including data storage and system control. It also provides the human interface so that the operator can monitor system operation and observe the data visually as it is collected. The ISA platform chosen for this system is also commonly used in

many desktop computers and in industrial control computers. Therefore, there is already a large amount of hardware and software available for this system. In addition, since many IBM compatible computers are designed to work with this bus, the data can be transferred directly from the acquisition module over the ISA bus to computer memory or a permanent storage media such as a hard disk or optical drive. This eliminates the need to interface the computer to an external platform such as a CAMAC or VXI crate with a much slower communication protocol such as the IEEE-488.

It was mentioned in the previous paragraph that the ISA platform is used in many industrial control applications. The harsh operating conditions often found in factories has resulted in a large number of commercially available ruggedized ISA based computer systems. This type of computer system is also ideally suited for remote sensing applications where the instruments are often shipped to remote sites to participate in field campaigns in almost any operating environment. The computer chosen for the data acquisition system is an industrial rackmounted unit which houses the monitor, keyboard, CPU, peripherals, and ISA bus. The single board computer which inserts into one of the 15 slots on the ISA backplane is a 50 MHz 486DX based computer. The enclosure has a 15 slot passive ISA backplane for the CPU board, data acquisition modules, and the peripherals. It includes a 300W power supply and several cooling fans. The enclosure provides additional mounting hardware to properly secure any board inserted into the ISA backplane and prevent problems from vibration.

The central and most demanding function of this computer is to save the large amounts of data taken by the data acquisition units. Therefore, enough monetary resources were allocated so that premium quality data storage components could be purchased. These

included a 500 MB SCSI-II hard drive and a 20 MB floptical drive. The computer was equipped with standard VGA graphics, because the graphics functions of this computer are primarily system control and monitoring. The acquired data will be transferred via a network connection to a data processing workstation. This workstation will have the graphical resources required to visually represent the data in three dimensions.

The last major system component evaluated was the timing and control unit. This custom designed and constructed unit is responsible for coordination and control of the data acquisition modules, the transmitter, and the VSM motor control system. The timing functions were accomplished by using programmable timers synchronized with a single stable clock source. The timers and the control functions are implemented with TTL chips. This technology was selected because the chips are inexpensive, easy to replace, and consume moderate amounts of power. The timing and control unit is interfaced with the data acquisition computer through its printer port. This enables the timing and control functions to be software programmable from the data system computer. The timing and control unit is housed in a rack mounted enclosure. The specifications for the major components selected for the data acquisition system are summarized in table 4.3.

Table 4.3. Final System Configuration

System Component	Manufacturer	Component Specifications	Cost
6 channel A/D system	Bittware Research	6 channels, 12 bit, 20 MHz A/D 25 MHz DSP processor	$25,000
photon counting system	Santa Fe Research	1 channel photon counter 100 ns dwell time 300 MHz count rate	$5,000
ISA bus computer system	Apro	Intel 486DX CPU ISA system bus SCSI-II hard drive floptical drive	$5,000
Timing and Control Unit	custom	Custom built electronics parallel port interface	$5,000
		Total Cost =	$40,000

4.3 System Hardware

There are several hardware components making up this system, including signal conditioners, data acquisition modules, the system platform and computer, and the timing and control unit. Figure 4.3 shows how the input signals travel through the system components. First, the backscattered lidar signal is converted from an optical signal to an electrical signal by the photodetector. In the A/D channel the signal, which is comprised of many short duration current pulses, is converted to a voltage signal and filtered to reduce the noise level. In the photon counting channels, the incoming current pulses above a threshold level, are converted into uniform logic pulses by a discriminator. Each of the six polarization channels is connected to the Bittware data acquisition module, which is represented as a block diagram in Figure 4.4. The Bittware module consists of two separate boards which are inserted into

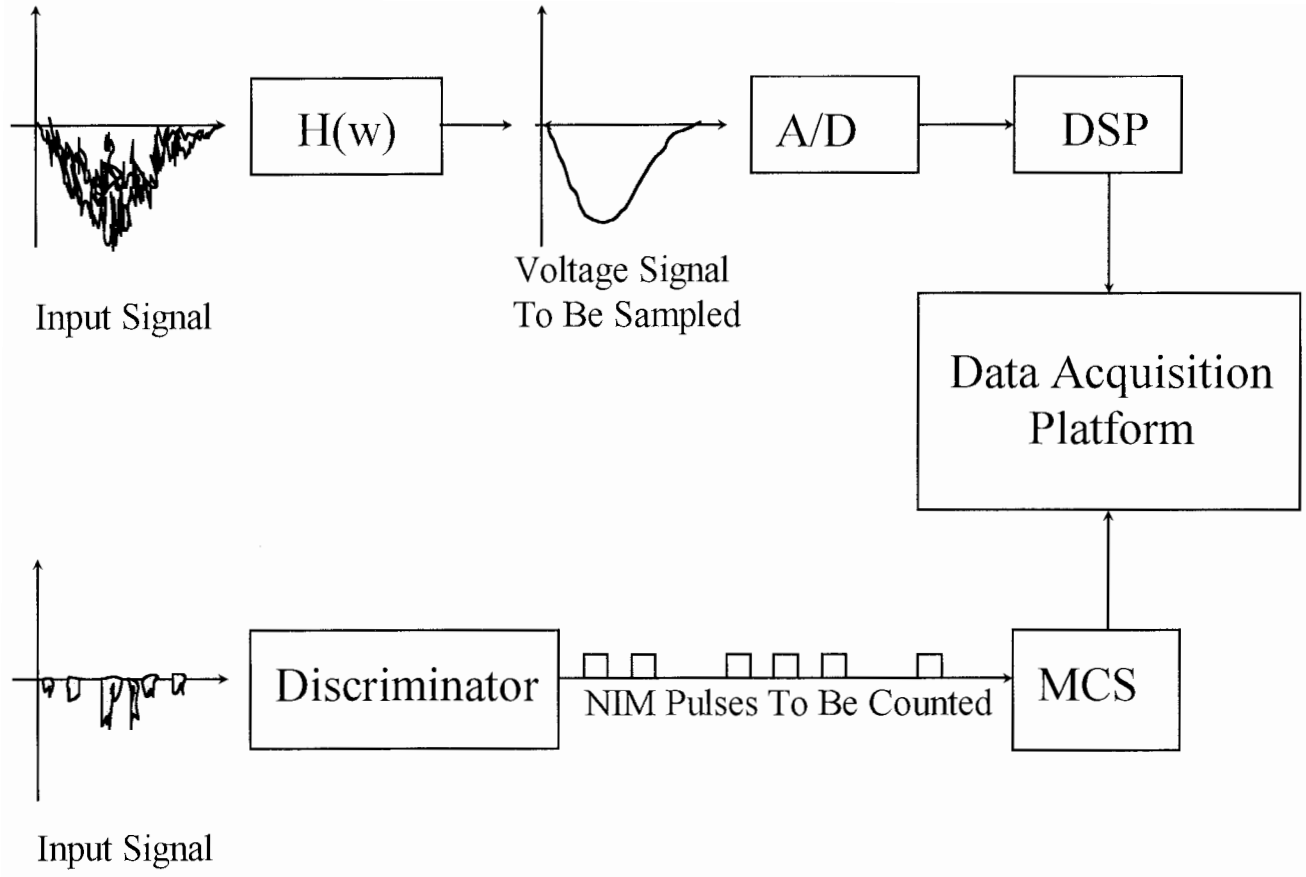

Figure 4.3. Signal flow diagram.

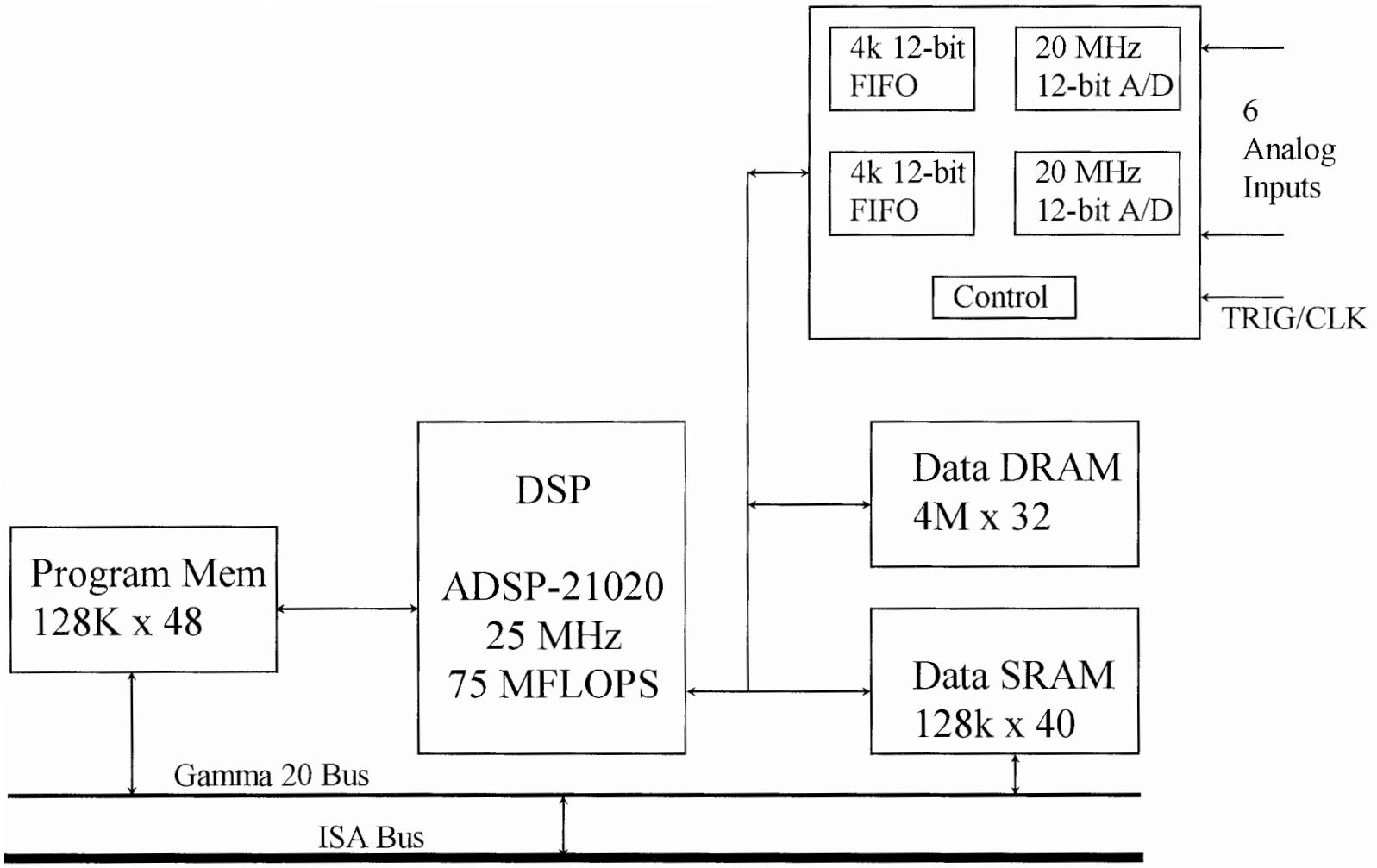

Figure 4.4. Block diagram of Bittware module.

the ISA backplane of the data acquisition platform. The first board is a custom designed analog interface board which consists of six A/D converters, a buffer memory, and timing and control circuitry. This board is connected via a ribbon cable which implements a proprietary high speed local bus with an adjacent board. The adjacent board is a high performance digital signal processing board consisting of an Analog Devices 25 MHz DSP processor and onboard memory. The filtered signals from each of the A/D channels of the volume scanning lidar are sent to one of the 6 A/D channels on the analog interface board and are sampled. The sampled data from each burst is stored in a temporary FIFO data buffer. The data is then transferred from the FIFOs over the local bus to Bittware's Gamma-20 DSP board for processing. The DSP board has 4 MW of 32 bit DRAM and 128KW of 48 bit SRAM onboard memory for storing the data. Once a segment of data is processed, it is transferred to the memory of the data acquisition computer. While the board is acquiring and averaging the next segment of data, the previous segment's data is saved to permanent storage media.

The Raman nitrogen channel of the VSL is connected to the MCS-100 photon counting module. This module consists of a single board which is also inserted into the backplane of the ISA platform. The module counts the number of incoming pulses from the discriminator during one dwell period which corresponds to one range bin. The individual shots of data are accumulated on the board and averaged to form a single segment of data. The segment of data is then transferred to the host computer, where it is saved to disk while the photon counting module is acquiring the next segment of data.

For the volume scanning lidar's data acquisition system to operate properly, all of the individual components of the system must be coordinated. In addition the actions of the

transmitter and the motor control system must be coordinated with the data system. The timing and control unit generates the timing and control signals necessary to accomplish this task, and is the heart of the entire system.

The timing signals that are generated by the timing portion of the timing and control unit are shown in Figure 4.5. All of the timing signals are derived from the MCLK signal generated by a stable 20 MHz crystal oscillator. The data acquisition modules each require a trigger signal and an acquisition clock. The Bittware module requires a 20 MHz sampling clock, ADCLK, and the Santa Fe MCS-100 module requires a 10 MHz dwell clock, DCLK. Both modules receive the same trigger signal, TRIG. The trigger signal occurs before the laser is fired so the initial range bins of data are actually measurements of background noise. These background measurements are used later in the analysis of the lidar data.

The firing of the laser is synchronized with the data system by external controlling the discharge of the laser flashlamps and the opening of the Q-switch. The laser transmitter operates by exciting a gain medium in a cavity with the flashlamp discharge. This excitation is allowed to build for an optimal period of time before a laser pulse is transmitted by opening the Q-switch in the laser cavity. For the scanning lidar's transmitter, this optimal period of time is 201μs. The flashlamps are fired with a logic low 20 μs TTL pulse, FIRE, and the Q-switch is controlled also with a logic low 20 μs pulse, QSW.

The control logic for the scanning lidar is shown in Figure 4.6. There are three functions that the control portion of the timing and control unit must perform. These include enabling the firing of the flashlamps, starting and stopping the laser from firing, and beginning a scan. The flashlamps are enabled with a front panel toggle switch. This allows the laser to

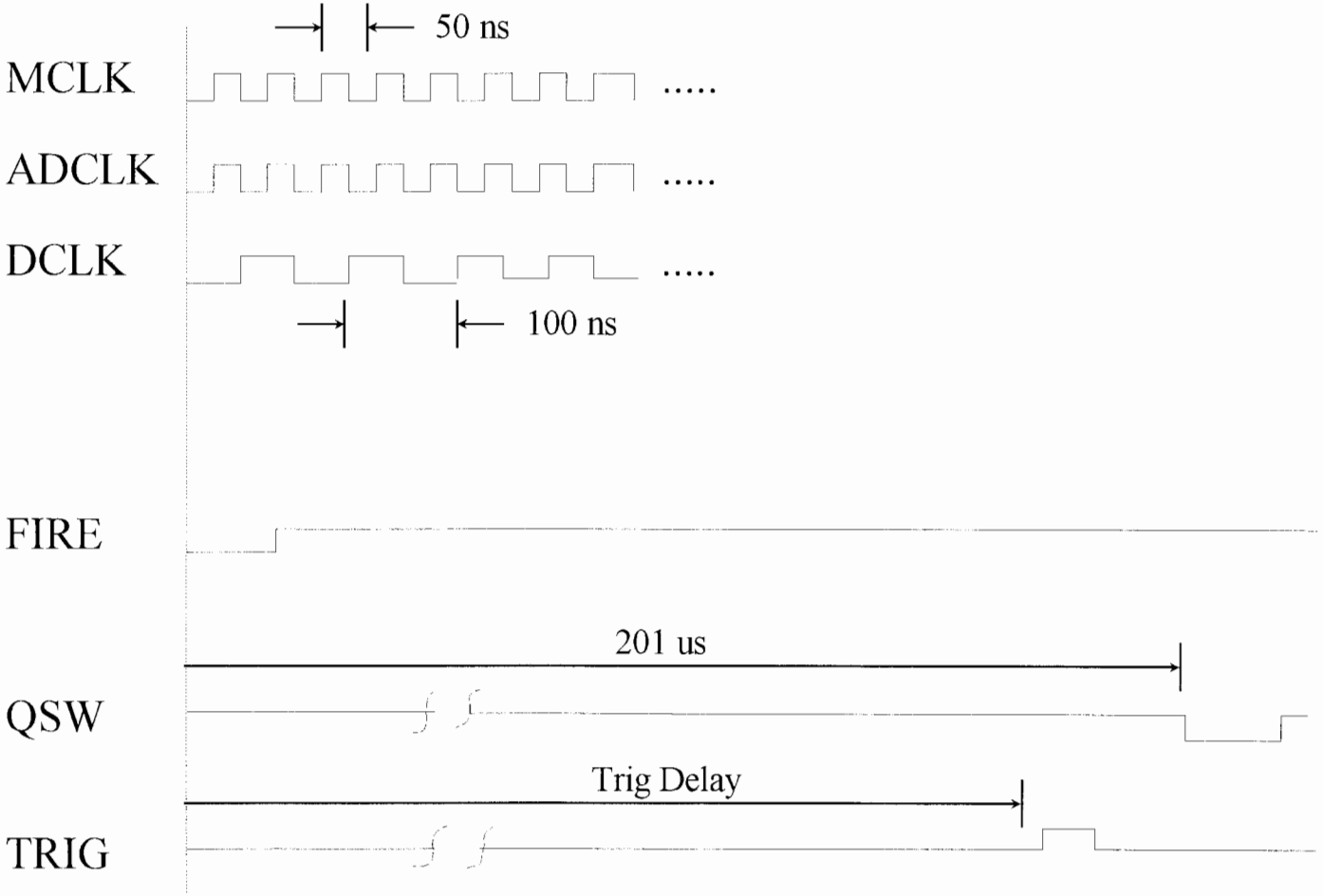

Figure 4.5. Timing diagram

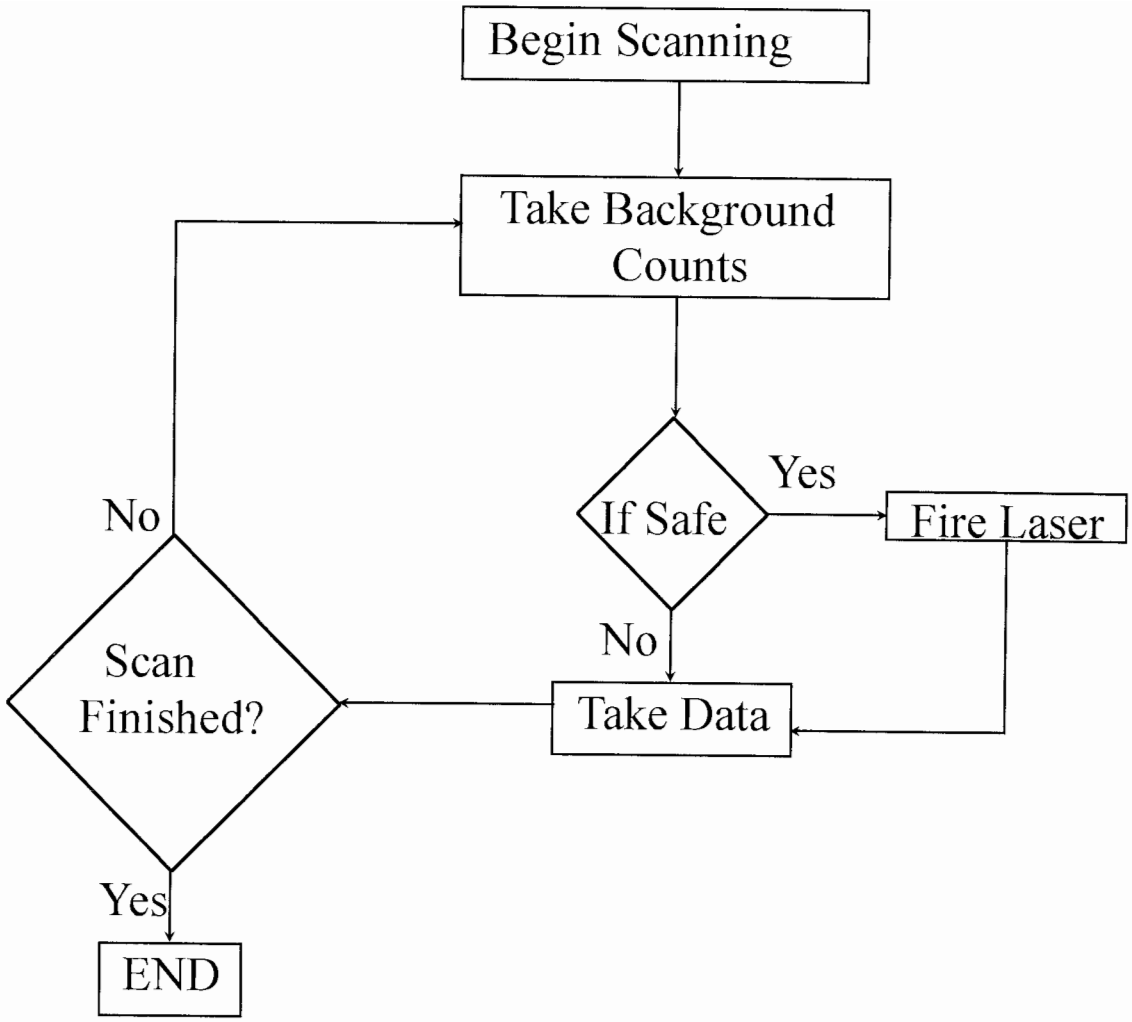

Figure 4.6. Control logic implemented by the timing and control unit.

be warmed up while it is in external control mode. The firing of the laser can be accomplished with either front panel buttons, or through the computer. The control portion of the timing and control unit also monitors the safety interlocks of the system. If the airplane warning radar detects an airplane, the laser is inhibited from firing and the scanning platform stops. Also a panic switch, for an auxiliary observer, can stop the laser for any reason. The third control function that is implemented is starting a scan. This can only be done through the system software running on the data system computer. When a scan is initiated, the trigger signal to the data acquisition modules is enabled and the motor control system initiates the scan program for the scanning platform.

The timing and control functions discussed above are implemented with the timing and control unit. A complete block diagram of the TCU electronics is shown in Figure 4.7. The actual schematics for this unit are included in Appendix A.

4.4 System Software

An essential component of the data acquisition system is the software. This software performs the functions of configuring the system parameters, system control, displaying the system status, plotting each data segment in raw form, and saving the data. The main software program is called LARSDAQ and was written in the C language. There is also a program which runs on the embedded DSP processor in the Bittware module that performs the signal averaging. Listings of the programs are included in Appendix B.

The system configuration file, LARS.CFG is another important feature of the system software. This file contains the current hardware configuration of the system. This file can

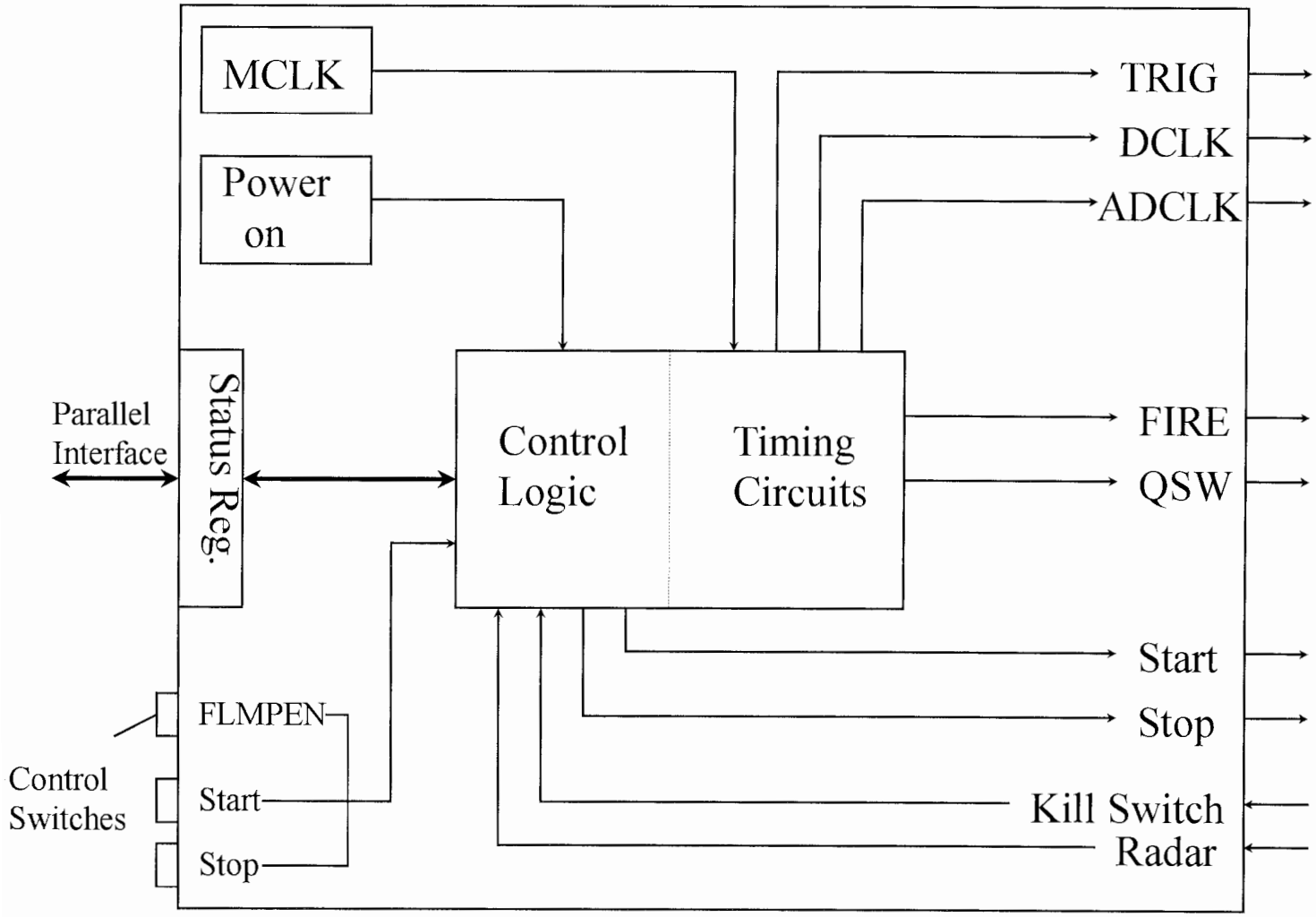

Figure 4.7. Block diagram of Timing and Control Unit (TCU).

be viewed and edited using the LARSDAQ program or with a text editor to change such parameters as the trigger delay time.

The LARSDAQ program performs system control functions such as starting and stopping the laser and displaying the system status. Once a data profile of a single segment is taken, the data from each of the channels for this segment can be displayed for diagnostic purposes.

The system software also coordinates the transfer of data from the data acquisition modules, to the data acquisition computer, and finally some form of permanent data storage. The real time nature of this system results in three data paths within the system that have critical timing requirements. The first critical path occurs between the data buffer on the Bittware A/D board and the memory on the Bittware DSP board. All of the data must be transferred from the A/D buffer before another signal return can be digitized. If this transfer is not completed in the 50 ms before the next laser pulse is transmitted, the system will not be able to sample the return from this pulse. Because this transfer occurs after every laser pulse is digitized, missing one laser shot for every one acquired will result in data being collected from only 50% of the laser pulses transmitted. The Bittware system was selected in part because its high speed local bus connecting the A/D buffers to the data DRAM should prevent this situation from occurring.

The second critical data path is the ISA bus which connects the data acquisition modules and the data acquisition computer. Data flows through this path once a complete segment of data has been collected. It is desired that this transfer also occur in less than 50 ms so that no laser shots are missed. However, this requirement is not as stringent as with

the first path, because this transfer only happens after at least 20 shots are acquired. If 1 laser shot is missed during a data transfer through this path, 95% of all the laser shots will still be utilized.

The third critical path is from the computer to a permanent storage medium, such as a hard disk. Because the data acquisition modules function independently from the computer, they can be collecting the next segment of data while the previous segment is saved to disk. With each segment taking at least 1 second to acquire and average, there is plenty of time to transfer data through the third path. However, the faster the transfer occurs, the more CPU time is available to perform other functions such as data plotting.

The data from each segment of the entire scan is stored in the American Standard Code for Information Interchange (ASCII) format. This is a generic format which enables the information to be understood by a variety of computer systems and programs for subsequent data analysis. Each data file also has a header file containing important information about the time the data was taken, which channels were used, what the spatial resolution was, and the laser power. This will enable the data processing workstation to combine with the position data from the VSM motion controllers to form three dimensional maps of the region being scanned.

The LARSDAQ program is also used to setup and control three separate pieces of hardware, the timing and control unit, the two Bittware units, and the Santa Fe energy unit. The electronics unit is controlled through the computers printer port. The value for the trigger delay register is downloaded through this connection. This register controls the length of time data is acquired before the laser is fired. As mentioned previously this data

corresponds to background measurements. Also, commands can be sent from the LARSDAQ program to the timing and control unit to start and stop the laser and to begin an atmospheric scan.

This completes the description of the data system design. The next chapter presents test results to verify the performance of the critical data transfers of the system, along with data from the initial measurements taken with the system. This is followed by conclusions and suggestions for future research.

Chapter 5

Results and Conclusions

The system described in this thesis has been constructed and is currently being tested. This chapter presents results from tests verifying the performance of the critical data paths and the initial measurements from the system. Next, the conclusions reached by this research are presented along with a discussion of suggested improvements and possible areas for future research.

5.1 Results

A measurement of the critical data transfer rates was the first test of the data system's performance. The measured values are tabulated in Table 5.1.

Table 5.1. Summary of measured data transfer rates

Critical Path	Data Amount	Transfer Time (ms)	Transfer Rate (MB/s)
A/D Board to DSP Board	55.2 KBytes	12	4.5
DSP Board to Data Acquisition Computer	55.2 KBytes	120	0.5
Computer to Hard Disk	55.52 KBytes	14	4

The most critical data path occurs from the A/D data buffer to the memory on the DSP board. This transfer must occur in less than 50 ms. The measured time of 12 ms was

less than this limit, so the system met the design specification. The second critical path occurs from the DSP board's memory to the data acquisition computer's memory. The transfer time for this path was measured to be 120 ms. This time will result in three laser shots being missed by the system between each segment of data. Currently, the data from each segment of the scan is averaged and temporarily stored in the DRAM portion of the Bittware modules the segment is scanned. Once the scan of that segment is complete, the data is transferred from the DRAM to the data acquisition computer. Changing the temporary storage area to the faster SRAM section could improve the transfer time through the second critical path.

The third critical path occurs from the data acquisition computer's memory to the SCSI-II hard disk where the data is permanently stored. This transfer must be completed before the next segment of data is acquired. The system has no trouble satisfying this requirement, with a measured transfer time of 14 ms.

Collection of actual atmospheric data in both the A/D mode and the PC mode, was the next test conducted to verify proper system performance. The first of these tests was an evaluation of the 607 nm photon counting channel that measures atmospheric nitrogen. The data acquisition system performed well in this test and a plot of the measured return for a 5 minute average is given in Figure 5.1. However, due to difficulties experienced by the vendor in manufacturing a photon counting board with a 100 ns dwell time, a slower photon counting board was employed for initial testing, resulting in a vertical range resolution of 480 m. This is greater than the 75 m or less given by the specifications in Chapter 3, but was unavoidable. The measured results compare well with the results predicted in Figure 3.1. Both the predicted and measured signals for the 607 channel peaked at approximately 1 km.

For 480 m range bins, the predicted peak count was 23,040 counts and the measured peak count was 16,000 counts.

Once the Bittware A/D system was operational, it was also tested by performing several measurements of the atmosphere under a variety of conditions. This data, taken on two separate evenings, is presented in Figure 5.2 to Figure 5.5. Figure 5.2, showing the data for a 1 second average, indicates that 1 second averages will result in data of a limited range. Part of this is attributed to misalignment of the laser beam and the field of view of the receiving telescope. Also, this figure and the others show a strong low altitude spike occurring near an altitude of 500 m. The cause of this spike is most likely the nearfield scattering of light from the optical surfaces and nearby structures. Figures 5.3 and 5.4 show that longer averaging times will increase the altitude range and also show a cloud layer which moved in during the data taking session. Figure 5.4 also shows a rapid change in signal strength occurring at 2600 m. The apparent cause of this is the laser leaving the field of view of the telescope. The system was tested on a subsequent night and also functioned well. This data, displayed in Figure 5.5, showed multiple cloud layers in the atmosphere, with the highest occurring at 6.5 km.

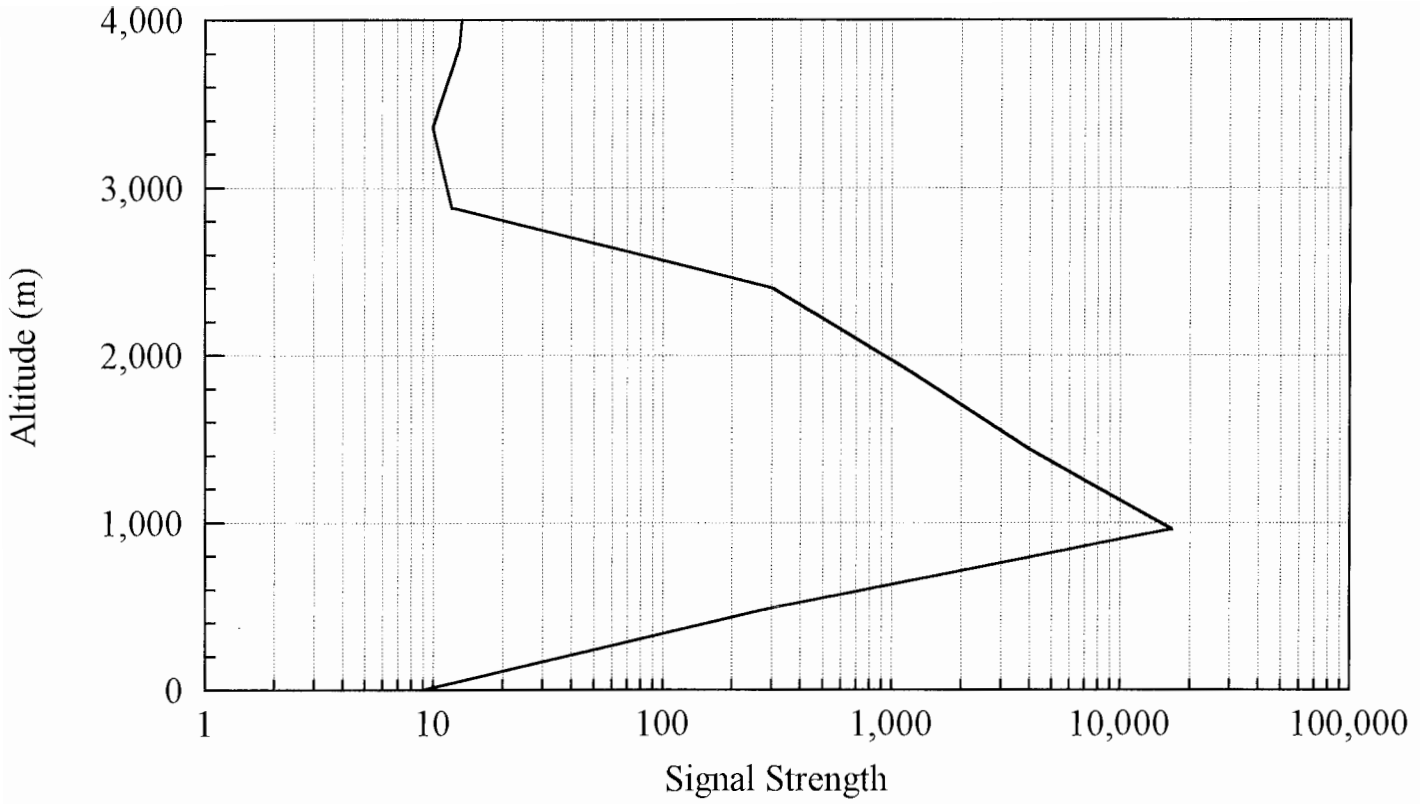

Figure 5.1. Signal return from the 607 nm Raman channel of the volume scanning lidar. Profile obtained on May 3, 1994, averaging for 5 minutes.

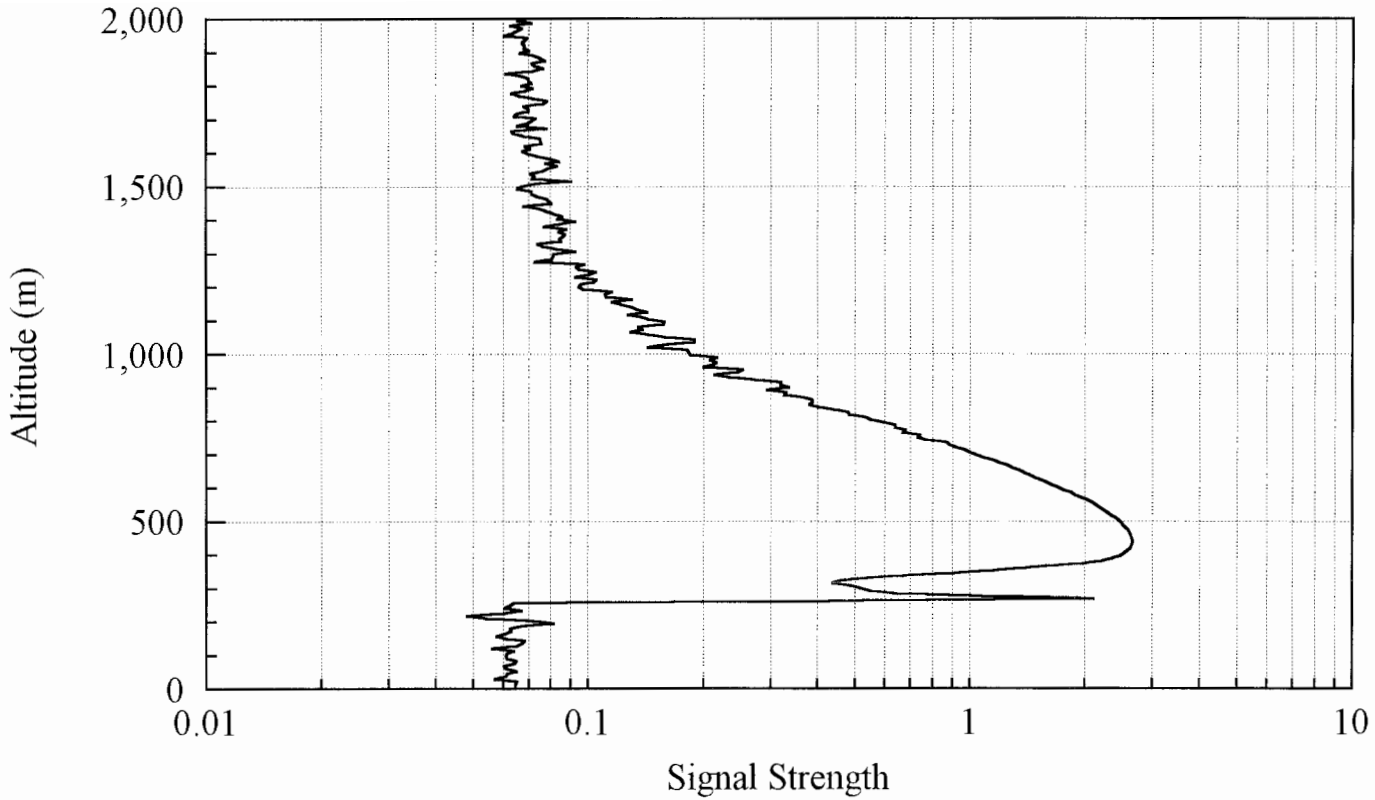

Figure 5.2. Signal return from the parallel polarized 532 nm channel of the volume scanning lidar. The profile was obtained on October 30, 1994, averaging for 1 second.

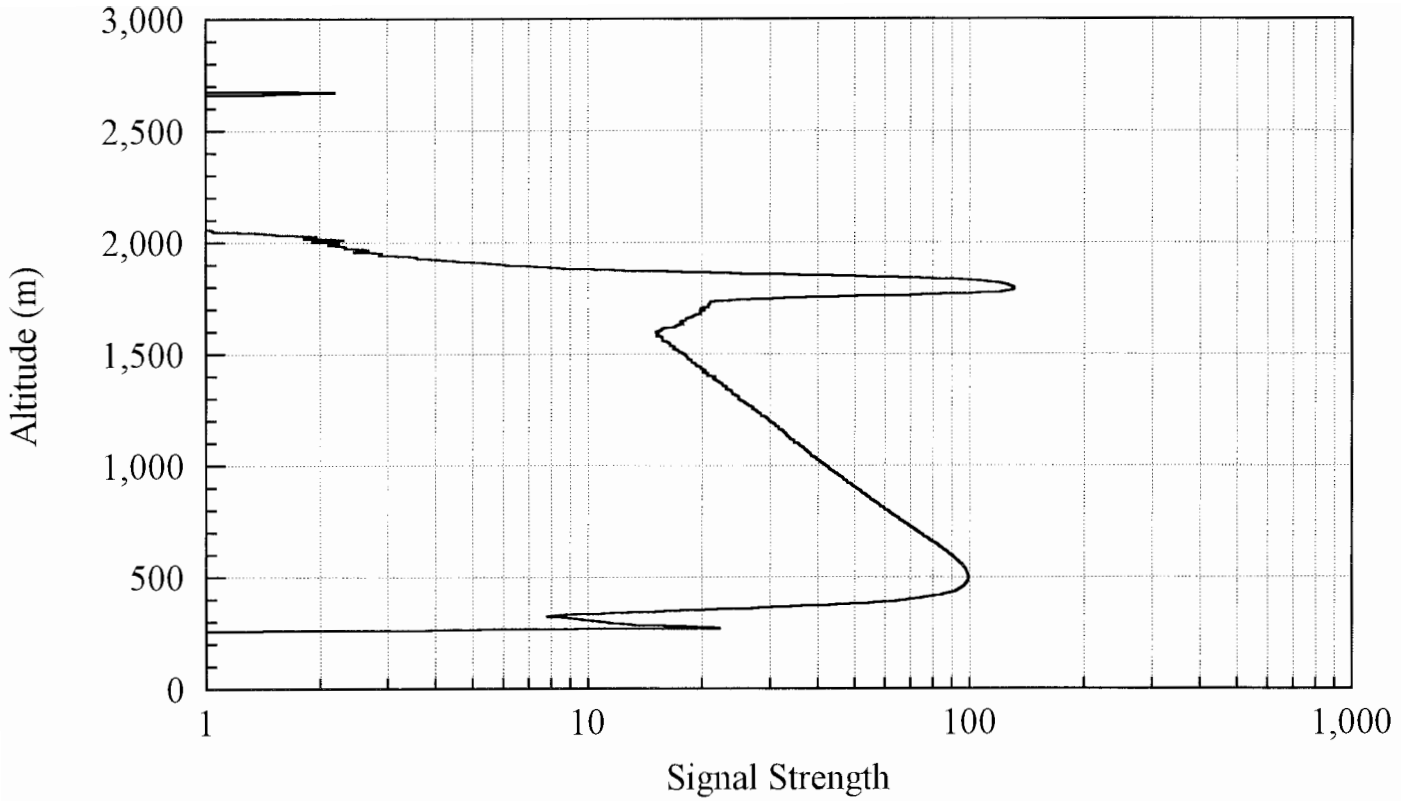

Figure 5.3. Signal Return from the parallel polarized 532 nm channel of the volume scanning lidar. The profile was obtained on October 30, 1994, averaging for 10 seconds.

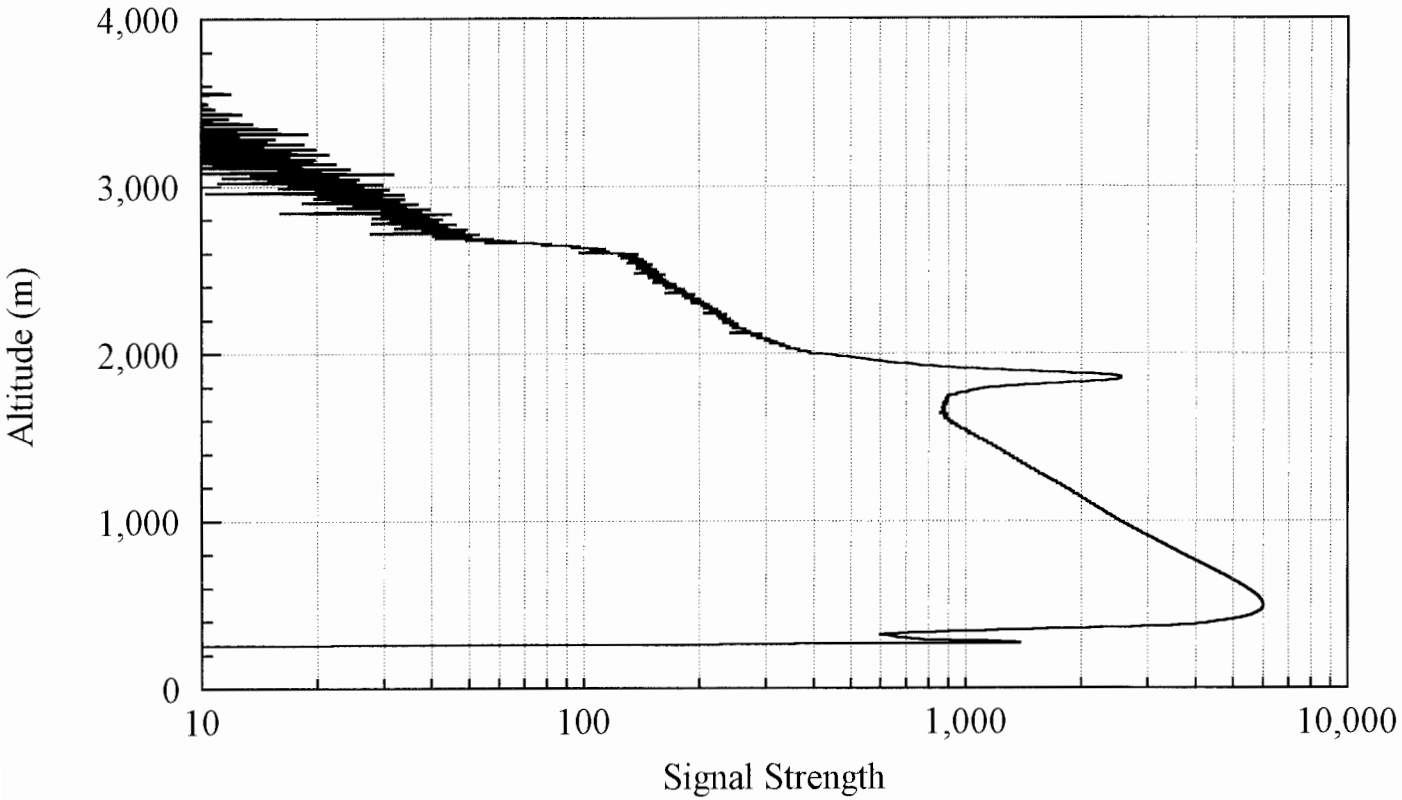

Figure 5.4. Signal return from the parallel polarized 532nm channel of the volume scanning lidar. The profile was obtained on October 30, 1994, averaging for 10 minutes.

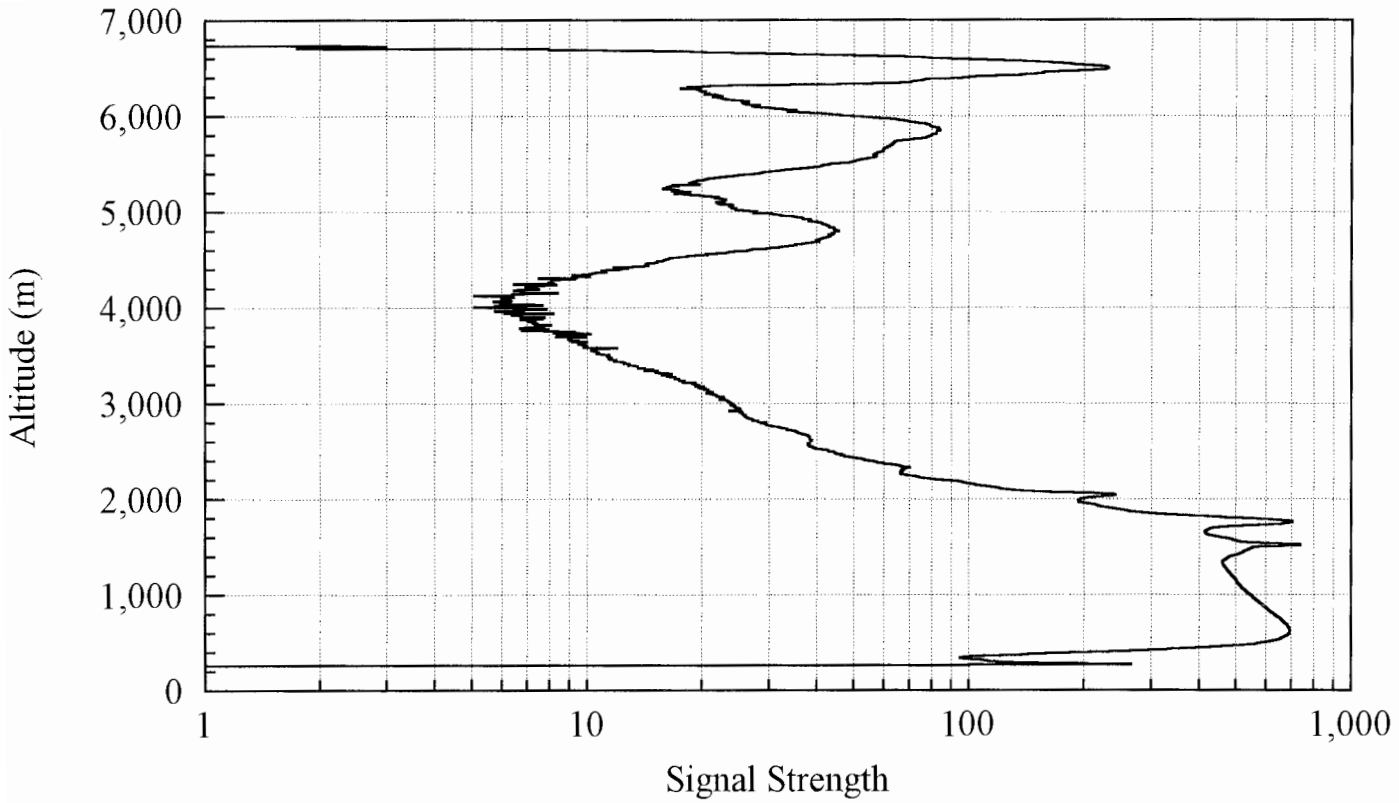

Figure 5.5. Signal return from the parallel polarized 532 nm channel of the volume scanning lidar. The profile was obtained on November 3, 1994, averaging for 10 minutes.

5.2 Conclusion

This thesis has discussed the design of a data acquisition system capable of handling the large data throughput requirements for a volume scanning, multi-wavelength, polarization lidar. This lidar is a component of the WAVE-LARS remote sensing system, designed to answer several important questions in atmospheric research. The system is currently capable of conducting vertically pointing measurements. Presently the main system components, the A/D system, the photon counting system, the timing and control unit, and system software, are operational and functioning properly. Only single channel measurements have been conducting at this point because the detector channels are not all fully functional. In addition, volume scanning measurements were not conducted because the scanning platform still needed to be balanced properly. The data system is ready for these capabilities and they are expected to be completed soon. The completed system met all of the design goals except the range resolution for the photon counting system and the duration of the data transfer time from the data acquisition modules to the data acquisition computer. The first problem will be addressed by selecting an alternative photon counting system, as additional funds become available. The second problem can possibly be addressed by altering the system software, as explained earlier.

5.3 Future Research

The design and initial demonstrations of the data acquisition and control system described in this thesis has been successfully completed. As mentioned above, some work is still needed on other components of the volume scanning lidar before the full capabilities of

this system can be utilized. Also, some improvement in system performance can be achieved by software modifications and replacement of one of the data acquisition modules. Once this is completed, the system will be able to perform the measurements necessary to answer the scientific questions posed in the beginning of this thesis. The focus for future research with this system, is validating system performance against other sensors, and conducting the multiple sensor measurements necessary to fully utilize this system's capabilities.

References

Ackerman, T. P., Albrecht, M. A., Miller, M. A., Clothiaux, E., Peters, R. M., Syrett, w., "Remeote Sensing of Cloud Properties Using a 94 GHz Radar," COMEAS Proceedings, 211-214, 1993.

Anuskiewicz, J. A., "A Volume Scanning System for a Lidar and Radar Sounder," MS Thesis, The Pennsylvania State University, 1993.

Beller, J., "A High-Performance Signal Processing System for the HP 8146A Optical Time-Domain Reflectometer," Hewlett-Packard Journal,63-68, 1993.

Cooney, J. A., "Comparisons of Water Vapor Profiles Obtained by Radiosonde and Laser Backscatter," J. of Applied Meteorology, Vol. 10, 301-308, 1971.

Donovan, D.P., Whiteway, J. A., Carswell, A. I., "Correction for Nonlinear Photon-counting Effects in Lidar Systems," Applied Optics, vol. 32, no. 33, 6742-6753, 1993.

Elterman, L. B., "The measurements of stratospheric density distribution with the search light technique," J. Geopys. Res., 58, 519-530, 1953.

Fiocco, G., Smullin, L. D., "Detection of Scattering Layers in the Upper Atmosphere (60-140 km) by Optical Radar," Nature (London), vol. 199, no. 4900, 1275-1276, 1963.

Haris, P. A. T., "Performance Analysis of the LAMP Rayleigh/Raman Lidar System," MS Thesis, The Pennsylvania State University, 1992.

Manning, T. E., "Design, Manufacture, and Testing of Volume Scanning Mechanism for a Lidar and Radar Sounder," MS Thesis, The Pennsylvania State University, 1993.

Mathason, B. K., "Performance Efficiency of the WAVE-LARS Scanning Lidar System," MS Thesis, The Pennsylvania State University, 1994.

Measures, R. M., *Laser Remote Sensing*. Florida: Krieger Publishing Co,. 1992.

Melfi, S. H., "Remote measurements of the atmosphere using Raman Scattering," Appl. Opt., Vol. 11, 1605-1610, 1972.

Sassen, K., "The Polarization Lidar Technique for Cloud Research: A Review and Current Assessment," Bull. of the Amer. Meteo. Soc. vol. 72, no. 12, 1848-1866, 1991.

Scheingold, D. H., *Analog-Digitial Conversion Handbook*, Englewood Cliffs, NJ, Prentice-Hall, 1986.

Schotland, R. M., Sassen, K., Stone, R. J., "Observations by Lidar of linear depolarization ratios bt hydrometers," J. Appl. Meteor., vol. 10, 1011-1017, 1971.

Stevens, T. D., "An Optical Detection System for a Rayleigh/Raman Lidar," MS Thesis, The Pennsylvania State University, 1992.

Ulaby, F. T., *Microwave Remote Sensing, Volume I: Microwave Remote Sensing - Fundamentals and Radiometry*. Norwood, MA: Artech House, 1981.

Young, A. T., "Rayleigh Scattering," Physics Today, 2-8 January 1982.

Zuev, V. E., "Laser-Light Transmission through the Atmosphere," *Laser Monitoring of the Atmosphere*, (E. D. Hinckley, ed.), Springer-Verlog, 1976.

Appendix A

The circuit schematics in this appendix are the actual circuitry used to implement the timing and control unit for the volume scanning lidar data acquisition system.

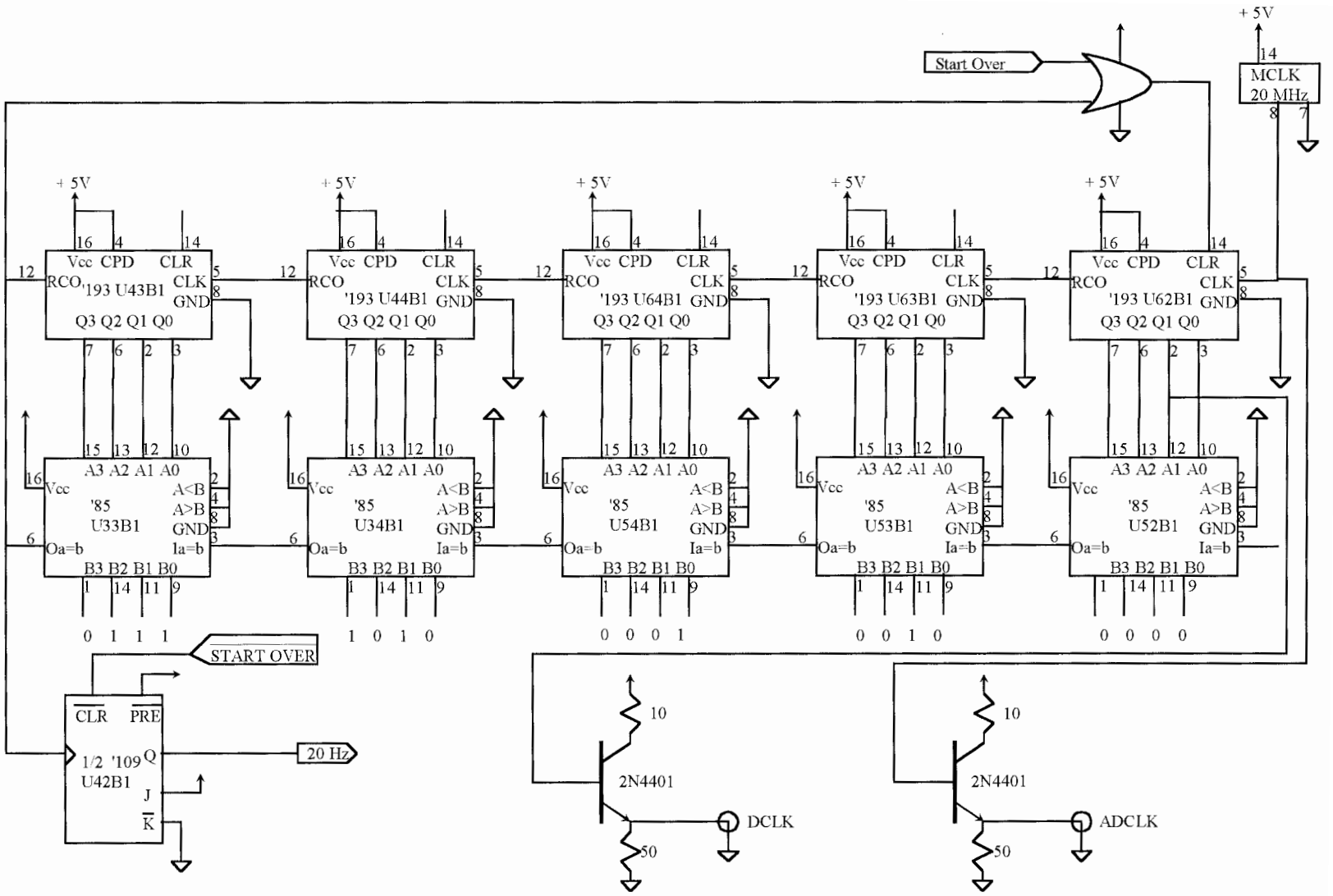

Figure A.1. Clock Generator Circuit

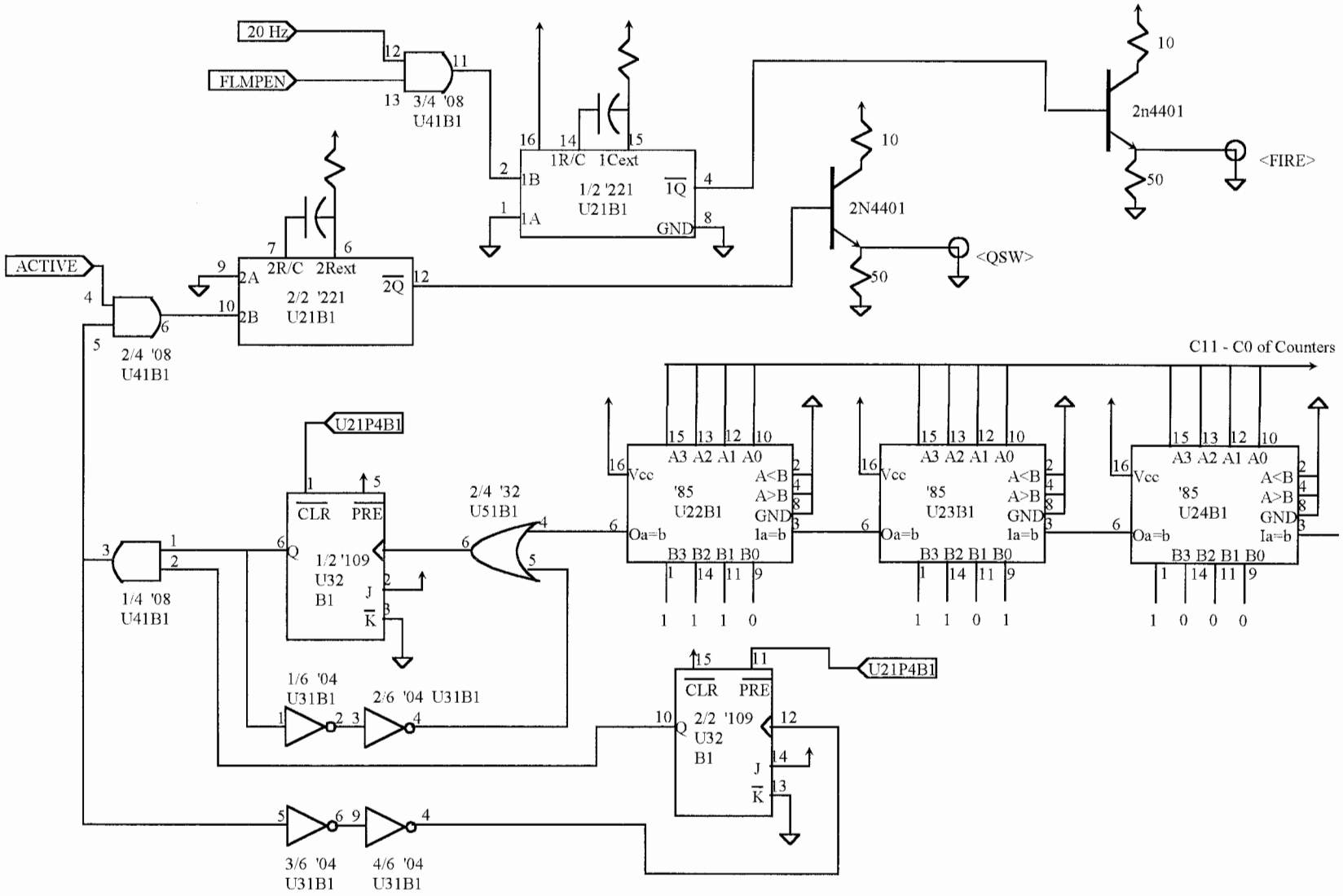

Figure A.2. Laser Timing Circuit

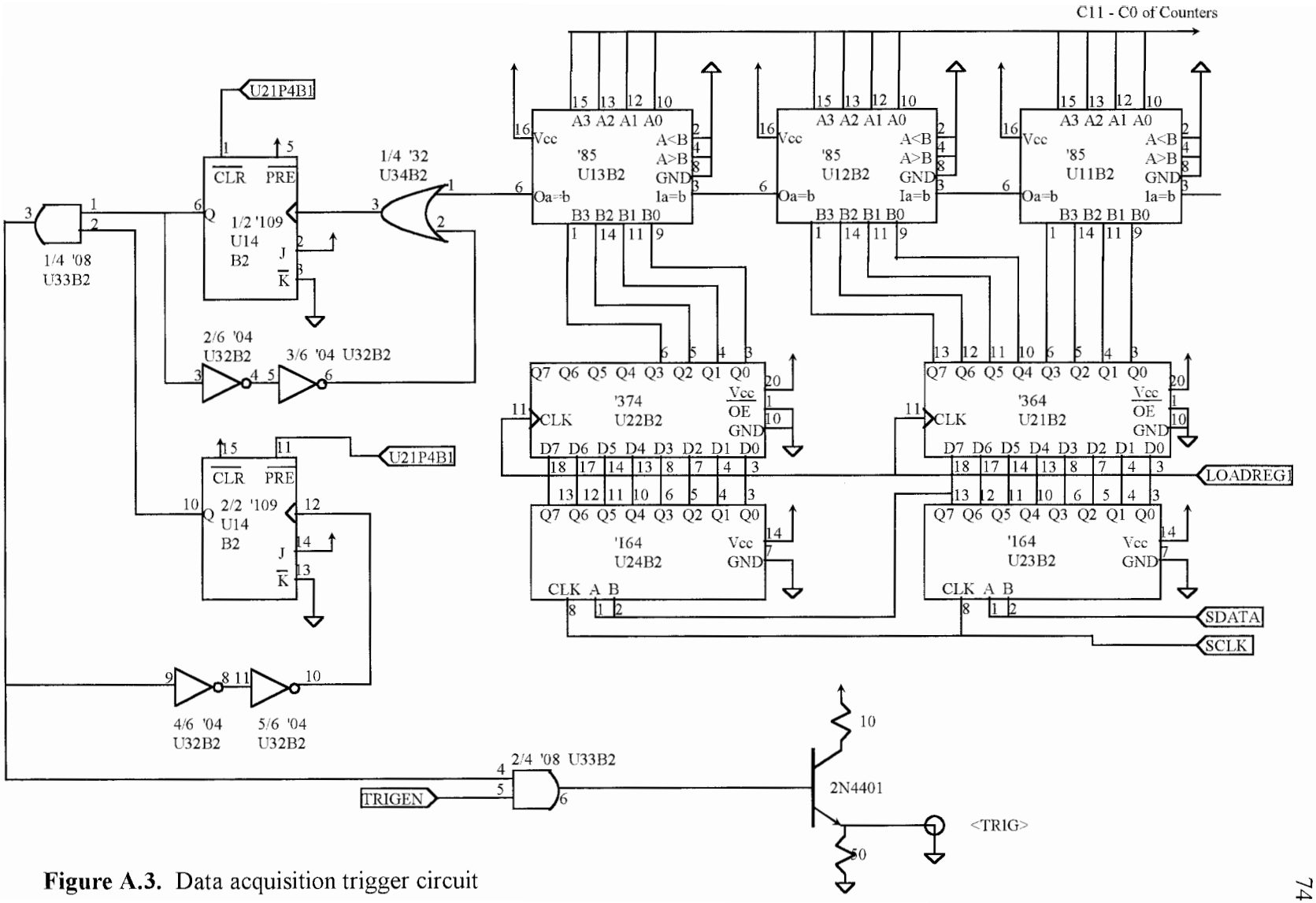

Figure A.3. Data acquisition trigger circuit

Appendix B

The software modules for the data acquisition system are listed in this appendix. They include LARSDAQ.C and LARS.C.

```
/*****************************************************************
**
 *
 * larsdaq.c : This program is the data acquisition and control prog. for
 *          the LARS volume scanning lidar.  The program sets system
 *          timing and communicates with the data acquisition units.
 *          The data units are a 6 channel 20 MHz 12 bit A/D system,
 *          and two photon counting channels.  The A/D system is made
 *          by Bittware Research and the photon counters (multi-channel-
 *          scalers) are made by Santa Fe Energy Research. Communication
 *          with the timing and control electronics is through the
 *          printer port.  The data units are controled by various
 *          vender supplied software routines. The following lists
 *          the files that need to be linked to this program.
 *
 * Files to be linked!!
 *
 *    larsdaq.obj   compiled using Microsoft C 6.00
 *
 *    ioport.obj    needed for MCS-II
 *    ioportb.obj   needed for MCS-II
 *    mcslib2.obj   needed for MCS-II
 *
 *
 * Required Linker settings
 *    stack size 4096
 *    No Extended Dictionary
 *    Additional libraries:   llibfor7.lib (needed for MCS-II)
 *                            graphics.lib
 *                            c:\dsp21k\msclib\dsp21kl.lib (large lib
 *                               needed by bittware)
 *
 * Required Build Settings
 *    default options
 *
 * Required Compiler Options
 *    LARGE memory model:     needed because mcs and bittware use lib's
 *                            based on large memory model
 *
 * NOTE!!!:  mcs communication routines use the FORTRAN calling convention
 *       So all parameters passed by reference!!!
 *
 * Additional files needed in current directory at run time:
```

```
*     lars.cfg--> contains configuration info for larsdaq
*     lars.21k--> contains the DSP program that runs on the AD21020
*        DSP processor on the bittware system. Program acquires
*        samples and averages them
*
*
*
*
* Revision History:
*
* Ver. 1.0
*
* 2/94    M. D. O'Brien
*     Initial coding.  Timing and control elelctronics routines
*     developed.
*
* 3/23/94 MDO
*     Communication to MCS-II units implemented.
*
* 4/19/94 MDO
*     scr.c was merged with larsdaq.c and revised.  This implemented
*     the user interface and system configuration routines.
*
* 6/12/94 MDO
*     Bittware communication routines implemented.
*
*
**********************************************************************
**/


#include <stdio.h>
#include <graph.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
#include <malloc.h>
#include <time.h>
#include "dsp21k.h"     /* needed for bittware communication routines */
#include "mcs.h"        /* function prototypes for MCS-II routines */
```

```
#define FALSE 0
#define TRUE !FALSE


#define data_reg 0x03BC


#define initial_value 0x18
#define cstop 0x10
#define cstart 0x8
#define creset 0x38
#define taking_data 0x48


#define BOARD_NUMBER 0      /* dsp board number */
#define PROGNAME "lars3.21k" /* dsp program name */
#define NUM_CHANS 6         /* num of A/D channels */



void open_screen(void),
    draw_screen(void),
    config_system(void),
    system_reset(void),
    motor_setup(void),
    take_data(void),
    process_data(void),
    trig_reg(int),
    start_trigger(void);


void init_bittware(void);
void go_bittware(int seconds, int num_samps);
void end_bittware(int num_samps);
void init_mcs(void);
void go_mcs(int shots);
void end_mcs(long *pc_out_buf, long num_bins);
void save_data(long *pc_out_buf, int seg_number,
               int num_ad_samps, int num_pc_samps);



DSP21K *board;      /* temporary global variable */
```

```
static float ad_out_buf[6][2300];



main()
{

char choice;
int quit;
short oldfgd;
long oldbgd;
struct rccoord oldpos;
struct videoconfig vc;

   oldfgd=_gettextcolor();
   oldbgd=_getbkcolor();
   oldpos=_gettextposition();
   _getvideoconfig(&vc);

   open_screen();
   draw_screen();
   _settextcolor(7);

   quit=FALSE;
   while (!quit){
         choice=getch();
         switch(choice){
            case '1':
                  config_system();
                  break;
            case '2':
                  system_reset();
                  break;
            case '3':
                  motor_setup();
                  break;
            case '4':
                  take_data();
                  break;
            case '5':
                  process_data();
```

```
                break;
            case '6':
                quit=TRUE;
                _outtext("Terminating LARSDAQ.\n");
                _settextcolor(oldfgd);
                _setbkcolor(oldbgd);
                _setvideomode(vc.mode);
                _clearscreen(_GCLEARSCREEN);
                _settextposition(oldpos.row,oldpos.col);
                break;
            default:
                _outtext("Enter 1-6 only.\n");
                break;
        } /* end of switch */
    }   /* end of while */
}       /* end of main */


/*************************************************************************
**
 *
 *  function: void open_screen(void)
 *
 *  purpose:  This function clears the screen and prints the opening
 *            screen.  Waits for the user to hit enter.
 *

*************************************************************************
*/

void open_screen(void)
{

char ch;

   _clearscreen(_GCLEARSCREEN);
   printf("\n\n\n\t\t\t          LARSDAQ\n");
   printf("\t\t\t Data acquisition program for\n");
   printf("\t\t\t LARS volume scanning lidar\n");
   printf("\n\n\n\t\t\t Ver. 1.0\n");
   printf("\n\n\n\t\t\t Press ENTER to continue\n");

   while ((ch=getch()) != 13);      /* wait for return */
```

```
}



/*************************************************************************
**
 *
 *  function: void draw_screen(void)
 *
 *  purpose:  This function draws the user screen.
 *
*************************************************************************
*/

void draw_screen(void)
{

struct videoconfig vc;
char buffer[80];

   _clearscreen(_GCLEARSCREEN);
   _setvideomode(_VRES16COLOR);
   _getvideoconfig(&vc);

   _setbkcolor(0);
   _setcolor(14);

   _moveto(vc.numxpixels/2, 0);
   _lineto(vc.numxpixels/2, vc.numypixels-1);
   _moveto(0,0);
   _lineto(vc.numxpixels-1, 0);
   _lineto(vc.numxpixels-1, vc.numypixels-1);
   _lineto(0, vc.numypixels-1);
   _lineto(0,0);

   _settextcolor(2);
   _settextposition(2,2);
   sprintf(buffer,"1. Config. System.");
   _outtext( buffer );
   _settextposition(2,25);
```

```
    sprintf(buffer, "2. System reset.");
    _outtext(buffer);
    _settextposition(4,2);
    sprintf(buffer,"3. Scan Setup.");
    _outtext(buffer);
    _settextposition(4,25);
    sprintf(buffer,"4. Take Data.");
    _outtext(buffer);
    _settextposition(6,2);
    sprintf(buffer, "5. Process Data.");
    _outtext(buffer);
    _settextposition(6,25);
    sprintf(buffer, "6. Quit.");
    _outtext(buffer);

    _moveto(0,vc.numypixels/4);
    _lineto(vc.numxpixels/2,vc.numypixels/4);
    _moveto(0,(vc.numypixels*4)/5);
    _lineto(vc.numxpixels/2,(vc.numypixels*4)/5);

    _settextposition(26,14);
    _settextcolor(4);
    sprintf(buffer,"System Status");
    _outtext(buffer);

}

void config_system(void)
{
char buffer[80], output[13][80], input[80], tmp;
char choice;
char *result;
int i,quit,channel,row,column;
FILE *fp;
struct rccoord pos;

    _settextcolor(7);
    if ((fp=fopen("lars.cfg","r+t"))==NULL){

          _settextwindow(9,2,22,40);
          sprintf(buffer,"lars.cfg not found.\n");
          _outtext(buffer);
          sprintf(buffer,"Loading default config.\n");
```

```
        _outtext(buffer);
        sprintf(buffer,"Hit any key to continue.\n");
        _outtext(buffer);
        getch();
        _clearscreen(_GWINDOW);

        sprintf(output[0],"a) Backgrd samples/shot(0-1000): 1000\n");
        sprintf(output[1],"b) A/D clk (MHz): 20.0\n");
        sprintf(output[2],"c) Dwell clk (MHz): 0.322580\n");
        sprintf(output[3],"d) CHAN#  TYPE   NAME\n");
        sprintf(output[4],"    1      N.A.\n");
        sprintf(output[5],"    2      N.A.\n");
        sprintf(output[6],"    3      N.A.\n");
        sprintf(output[7],"    4      N.A.\n");
        sprintf(output[8],"    5      N.A.\n");
        sprintf(output[9],"    6      N.A.\n");
        sprintf(output[10],"    7      N.A.\n");
        sprintf(output[11],"    8      N.A.\n");
        sprintf(output[12],"\n");
        for(i=0;i<13;i++){
           sprintf(buffer,output[i]);
           _outtext(buffer);
           fputs(buffer,fp);
        }
}
else{
        for(i=0;i<13;i++){
           fgets(buffer,80,fp);
           strcpy(output[i],buffer);
        }
}
fclose(fp);

_settextwindow(2,42,29,79);
for(i=0;i<13;i++)
        _outtext(output[i]);

_settextwindow(9,2,24,40);
sprintf(buffer,"Enter letter of any field to change.\n Or x to quit.\n");
_outtext(buffer);

quit=FALSE;
while(!quit){
```

```
choice=getch();
switch(choice){
   case'a':
         sprintf(buffer,"Enter background samples/shot.\n");
         _outtext(buffer);
         sprintf(input,"\n");
         scanf("%s", &input);
         sprintf(buffer,"\n");
         _outtext(buffer);
         sprintf(output[0],"a) Backgnd samples/shot(0-1000): ");
         strcat(input,"\n");
         strcat(output[0], input);
         _outtext(output[0]);
         break;

   case 'b':
         sprintf(buffer,"Can only change through hardware.\n");
         _outtext(buffer);
         break;

   case 'c':
         sprintf(buffer,"Can only change through hardware.\n");
         _outtext(buffer);
         break;

   case 'd':
         sprintf(buffer,"Enter channel # to change.\n");
         _outtext(buffer);
         scanf("%d",&channel);
         _outtext("\n");
         _outtext("Enter 1. A/D  2. PC  3. NA\n");
         choice=getch();

         _outtext("Enter name of channel eg. 607nm etc.\n");
         _outtext("\n");
         i=0;
         pos=_gettextposition();
         while(i<20){
            tmp=getch();
            input[i]=tmp;
            sprintf(buffer," \n");
            buffer[0]=tmp;
            _outtext(buffer);
```

```
    pos.col=pos.col+1;
    _settextposition(pos.row,pos.col);
    if (input[i]==13)
          i=20;
    else
          i=i+1;

}
_outtext("\n");

switch(choice){
    case '1':
          sprintf(output[channel+3],"\n");
          sprintf(output[channel+3],"    ");
          sprintf(buffer,"\n");
          itoa(channel,buffer,10);
          strcat(output[channel+3],buffer);
          strcat(output[channel+3],"      A/D   ");
          strcat(output[channel+3],input);
          strcat(output[channel+3],"\n");
          _outtext(output[channel+3]);
          _outtext("\n");
          break;
    case '2':
          sprintf(output[channel+3],"\n");
          sprintf(output[channel+3],"    ");
          sprintf(buffer,"\n");
          itoa(channel,buffer,10);
          strcat(output[channel+3],buffer);
          strcat(output[channel+3],"      PC    ");
          strcat(output[channel+3],input);
          strcat(output[channel+3],"\n");
          _outtext(output[channel+3]);
          _outtext("\n");
          break;
    case '3':
          sprintf(output[channel+3],"\n");
          sprintf(output[channel+3],"    ");
          sprintf(buffer,"\n");
          itoa(channel,buffer,10);
          strcat(output[channel+3],buffer);
          strcat(output[channel+3],"      N.A.  ");
          strcat(output[channel+3],input);
```

```
                        strcat(output[channel+3],"\n");
                        _outtext(output[channel+3]);
                        _outtext("\n");
                        break;
                }
                break;


            case'x':
                quit=TRUE;
                break;
            default:
                sprintf(buffer, "Invalid entry. Try again.\n");
                _outtext(buffer);
                break;
        }
    }
    if((fp=fopen("lars.cfg","w+t"))==NULL)
        _outtext ("error opening lars.cfg");
    else {
        for(i=0;i<13;i++){
            sprintf(buffer,output[i]);
            _outtext(buffer);
            fputs(buffer,fp);
        }
        fclose(fp);
    }

}


void system_reset(void)
{
char ch, buffer[80], counts[20];
int i,done,bkcounts;
FILE *fp;


    _settextwindow(2,42,29,79);
    _clearscreen(_GWINDOW);
    _settextwindow(9,2,22,40);
    _clearscreen(_GWINDOW);
```

```
    sprintf(buffer,"\n");
    sprintf(counts,"\n");
    _outtext ("Setting trigger delay.\n");
    if ((fp=fopen("lars.cfg","r+t"))==NULL)
            _outtext("The filelars.cfg not found.\nNeed to configure system first.\n");
    else{
            fgets(buffer,80,fp);
            i=33;
            done=FALSE;
            while(!done){
               ch=buffer[i];
               if (ch==13)
                     done=1;
               else{
                     counts[i-33]=ch;
                     i=i+1;
               }
            }
    }

    bkcounts=atoi(counts);
    trig_reg(bkcounts);
}
```

```
void trig_reg(int bkcounts)
{
int i, x;
int sdata;

        sdata = (201E-6 - bkcounts * 50E-9) / 50E-9;

        for (i=12; i>=2; i--)
                {
                        x = sdata;
                        x = x >> (i-2);
                        x = x & 0x2;
                        outp(data_reg, x);
                        x = x | 0x1;
                        outp(data_reg, x);
                        x = x & 0x2;
                        outp(data_reg, x);
                }

        x = sdata;
        x = x << 1;
        x = x & 0x2;
        outp(data_reg, x);
        x = x | 0x1;
        outp(data_reg, x);
        x = x & 0x2;
        outp(data_reg, x);

        outp(data_reg, 0x4);
        outp(data_reg, 0x0);

}


void motor_setup(void)
{
}
```

```
/*************************************************************************
**
 *
 *  function prototype: void take_data(void)
 *
 *
 *
 *  This funtion initializes the MCS and Bittware data acquisition
 *       cards and collects the lidar data. First the function prompts
 *       the user for the number data segments and the number of seconds
 *       per segment.  Then the data acq. hardware is initialized and the
 *       data trigger signal from the timing and control unit (TCU) is
 *       initialized.  Data acquisition is started with the go_mcs and
 *       go_bittware functions.          The data is saved to disk by calling the
 *       save data function.  The data is first loadded to memory.  Storage
 *       for the A/D data is allocated with the static global array
 *       ad_out_buf.  The photon counting data is saved to dynamically
 *       allocated memory.
 *
 *
*************************************************************************
*/

void take_data(void){

int loop1, shots;
int seconds, segments;
long  *pc_out_buf;
int num_ad_samps=2300;          /* 1000 backgnd+1300 data --> 10 km */
int num_pc_samps=40;       /* 16 backgnd + 24 data --> 10 km */
int num_ad_chans=6;
int num_pc_chans=2;

   _settextwindow(9,2,22,40);
   _clearscreen(_GWINDOW);
   _settextposition(2,2);
   _outtext("Enter # of seconds/segment of data.\n");
   scanf("%d", &seconds);
   _outtext("\n");
   _outtext("Enter # of segments of data.\n");
   scanf("%d", &segments);
   _outtext("\n");
```

```
   shots=seconds*20;   /* # of laser shots = seconds x 20 Hz */

   if ((pc_out_buf=(long *)malloc(sizeof(long) * num_pc_samps *
num_pc_chans))==NULL){
         _outtext("PC memory allocation error.\n Data acquisition failed.\n");
         exit(1);
   }


   /* add check to see if system has been reset */

   init_bittware();
   init_mcs();
   start_trigger();

   for(loop1=1; loop1<=segments; loop1++){
         go_mcs(shots);
         go_bittware(seconds,num_ad_samps);
         if (loop1!=1){
            save_data(pc_out_buf, loop1-1, num_ad_samps, num_pc_samps);
            /*plot_data();*/
         }
         end_bittware(num_ad_samps);
         end_mcs(pc_out_buf, num_pc_samps);
   }

   save_data(pc_out_buf, loop1-1, num_ad_samps, num_pc_samps);
   free(pc_out_buf);
   dsp21k_close(board);
   outp(data_reg,initial_value);         /* stops data trigger */
                                         /* does not affect laser status */

   _outtext("Finished taking data\n");


}                          /* End of take_data */


/*****************************************************************
**
 *
 *  function init_bittware
 *
```

```
 *  This function creates a pointer (global var board) to a structure that
 *       contains the information from the board environment var set in
 *       the DOS AUTOEXEC.BAT file.  Then the DSP program is downloaded to
 *       the DSP program memory.
 *
*************************************************************************
*/

void init_bittware(void)
{

   board=dsp21k_open(BOARD_NUMBER);
   dsp21k_dl_exe(board,PROGNAME);

}

/*************************************************************************
**
 *
 *  function go_bittware
 *
 *  This function starts the acquisition program loaded into the DSP.
 *       It then downloads the acquisition parameters to the board.
 *       Then it sets the _acquire flag high to enable data acuisition on
 *       the next data trigger signal from the TCU.
 *

*************************************************************************
*/


void go_bittware(int seconds, int num_samps)
{

float samprate=20e6;
int pulserate=20;
int num_pulses=seconds*pulserate;

   dsp21k_start(board);

   dsp21k_dl_flt(board,dsp21k_get_addr(board,"_samprate"), samprate);
   dsp21k_dl_int(board,dsp21k_get_addr(board,"_pulserate"),20);
```

```
                /* if external trigger download 0 */
                /* else download pulse_rate */

   dsp21k_dl_int(board,dsp21k_get_addr(board,"_num_pulses"),num_pulses);
   dsp21k_dl_int(board,dsp21k_get_addr(board,"_num_samps"),num_samps);

   dsp21k_dl_int(board,dsp21k_get_addr(board,"_acquire_flag"),!0);


}

/*************************************************************************
**
 *
 *  function: end_bittware
 *
 *  This function waits for the DSP _acquire flag to go low.  This means
 *       that the specified number of laser shots have been sampled and
 *       averaged.  The data is uploaded to the ad_out_buf array of floats.
 *

*************************************************************************
*/


void end_bittware(int num_samps)
{

clock_t start, end;
register int loop1;
int loop2;
long adrs;
char buffer[80];

   while(dsp21k_ul_int(board, dsp21k_get_addr(board, "_acquire_flag")));

   start=clock();
   for(loop1=0; loop1<NUM_CHANS; loop1++){
        adrs = dsp21k_get_addr(board,"_obuf") + (loop1*num_samps);
        dsp21k_ul_flts(board,adrs,num_samps,&ad_out_buf[loop1]);
   }
   end=clock();
   sprintf(buffer,"\n clock %4.2f
```

```
seconds\n",((float)end-(float)start)/CLOCKS_PER_SEC);
   _outtext(buffer);

   dsp21k_reset(board);

}



void init_mcs(void)
{

clock_t start, stop;
long int port1, port2;
long int control_synch, control_tri, control_external, units, mcs_data;
long int memory_size, memory_partition;
long int preset[3];
float elap;
char scale;


   port1=528;
   port2=288;
   control_synch=1;
   control_tri=0;
   control_external=1;
   units=2;
   scale='e';
   preset[0]=0;
   preset[1]=0;
   preset[2]=0;
   memory_size=8192;
   memory_partition=0;

   MCS_START(&port1,&control_synch,&control_tri,&control_external,&units,
            &scale,&mcs_data,&memory_size,&memory_partition,preset);
   MCS_STOP(&port1);
   MCS_ZERO_DATA(&port1);

   MCS_START(&port2,&control_synch,&control_tri,&control_external,&units,
            &scale,&mcs_data,&memory_size,&memory_partition,preset);
   MCS_STOP(&port2);
```

```
    MCS_ZERO_DATA(&port2);

    start = clock();
    elap = 0;
    while(elap < 0.05){
        stop = clock();
        elap = ((float)stop-(float)start)/CLOCKS_PER_SEC;
    }

}



void go_mcs(int shots)
{

long int port1, port2;
long int control_synch, control_tri, control_external, units, mcs_data;
long int memory_size, memory_partition;

long int preset[3];
char scale;


    port1=528;
    port2=288;
    control_synch=1;
    control_tri=0;
    control_external=1;
    units=2;
    scale='e';
    preset[0]=shots;
    preset[1]=0;
    preset[2]=0;
    memory_size=8192;
    memory_partition=0;


    MCS_START(&port1,&control_synch,&control_tri,&control_external,&units,
            &scale,&mcs_data,&memory_size,&memory_partition,preset);
    MCS_START(&port2,&control_synch,&control_tri,&control_external,&units,
            &scale,&mcs_data,&memory_size,&memory_partition,preset);
```

```
}



void end_mcs(long *pc_out_buf, long num_bins)
{

long int port1=528;
long int port2=288;


   MCS_STOP(&port1);
   MCS_STOP(&port2);

   MCS_READ_DATA(&port1, pc_out_buf, &num_bins);
   MCS_READ_DATA(&port2, pc_out_buf + num_bins, &num_bins);

}



void start_trigger(void)
{

   outp(data_reg,initial_value);
   outp(data_reg,taking_data);

}



void process_data(void)
{
}

/*************************************************************************
**
 *
 *  function save_data
 *
 *  This function writes the data from a single data segment to a disk
```

```
 *       file in binary format.    The data file is later converted to ASCII
 *       by convert.exe.
 *

*************************************************************************
*/

void save_data(long *pc_out_buf, int seg_number,
               int num_ad_samps, int num_pc_samps)
{

clock_t start,stop;
char buffer[10], ext[3], file_name[25];
FILE *fp;
register int loop1;
float *mike;

   mike=malloc(sizeof(float)*1000);

   itoa(seg_number,buffer,10);
   sprintf(ext,buffer);
   sprintf(file_name,"c:\\lars\\data\\ad.%s",ext);

   if((fp=fopen(file_name,"w+b"))==NULL)
         _outtext("Error opening data file\n");
   else{
         start=clock();
         /*for(loop1=0; loop1<NUM_CHANS; loop1++){
            fwrite(&ad_out_buf[loop1], sizeof(float), num_ad_samps, fp);
         }
         for(loop1=1; loop1<=2; loop1++){
            fwrite(pc_out_buf, sizeof(long), num_pc_samps, fp);
            pc_out_buf=pc_out_buf+num_pc_samps;
         }*/

         for(loop1=1; loop1<100; loop1++)
           fwrite(mike, sizeof(float), 1000, fp);

         fclose(fp);
         stop=clock();
         sprintf(buffer,"\nclock save
%4.3f\n",((float)stop-(float)start)/CLOCKS_PER_SEC);
         _outtext(buffer);
```

```
    }
}
```

```
/*****************************************************************
**
 *
 *  LARS.C: This dsp program is to be run from the host program LARSDAQ.
 *      The program waits to have acquistion enabled, acquires data,
 *      and then clears the acquire_flag variable.  Next the program
 *      accumulates this data to any previous data until the specified
 *      number of laser shots are sampled.
 *
*****************************************************************
*/

/* include files */

#include "acqdat.h"    /* for aquire_data function */
#include <stdio.h>


/* constants */

#define DRAM_BASE       (int *) 0x42000000
#define NUM_CHANS       6
#define BUFSIZE         2300

/* global variables acessed by pc */

static int acquire_flag = 0;    /* set by pc to acquire, cleared by dsp */

static float samprate=20e6;          /* sample rate */
static int pulserate=0;          /* number pulses per second */
static int num_pulses=20;          /* number of pulses to acquire */
static int num_samps=2300;          /* # samps to keep after num_skip */
static int valid_acquire;

float obuf[NUM_CHANS][BUFSIZE];         /* output buffer */

void main(void)
{

int tcount;
int loop1;
int loop2;
```

```
int loop3;
float *data_ptr0;
float *data_ptr1;
int *source_ptr;
int data;


  acquire_flag=1;
  printf("size of float=%d.\n",sizeof(float));

  while(!acquire_flag);

  /* number of clock ticks per trigger -- if 0, external trigger */
  tcount = (pulserate != 0) ? (25e6 / pulserate) : 0;

  for(loop1=1;loop1<=num_pulses; loop1++){
     valid_acquire = acquire_data(1,          /* # of pulses */
                                  num_samps,  /* number of keep samples */
                                  DRAM_BASE,  /* store in DRAM */
                                  tcount);    /* tcount */
     for(loop2=0; loop2<NUM_CHANS; loop2 +=2){
          data_ptr0=obuf[loop2];
          data_ptr1=obuf[loop2+1];
          source_ptr=DRAM_BASE + (loop2/2)*BUFSIZE;
          if(loop1==1){
             for(loop3=0; loop3<BUFSIZE; loop3++){
                  data=*source_ptr++;
                  *data_ptr0++ = (float)(((data>>0) & 0x0000fff0)-32768)/32768.0;
                  *data_ptr1++ = (float)(((data>>16) & 0x0000fff0)-32768)/32768.0;
             } /* end of loop3 */
          }
          else{
             for(loop3=0; loop3<BUFSIZE; loop3++){
                  data=*source_ptr++;
                  *data_ptr0++ += (float)(((data>>0) & 0x0000fff0)-32768)/32768.0;
                  *data_ptr1++ += (float)(((data>>16) & 0x0000fff0)-32768)/32768.0;
             } /*end of loop3*/
          }

     } /* end of loop2 */

  } /* end of loop1 */
```

```
    acquire_flag=0;
}
```