PROJECT REPORT

"REMOTE MEASUREMENT OF PRESSURE, TEMPERATURE AND RELATIVE HUMIDITY"

submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION

BY

ROOPA . P
SRIVIDYA .B.V
UMA MAHESHWARI .R

Undertaken at

RAMAN RESEARCH INSTITUTE

BANGALORE

Under the guidance of

Mr. R. GANESAN
Scientist ,
Raman Research Institute,
Bangalore.

Mrs. B. RAJESHWARI
Lecturer, Dept. of E&C,
P.E.S.I.T,
Bangalore.

Department of Electronics and Communication
P.E.S. INSTITUTE OF TECHNOLOGY ,BANGALORE

1999-2000

RAMAN RESEARCH INSTITUTE

C.V. Raman Avenue, Sadashivanagar, Bangalore-560 080 India

July 6, 2000

Certificate

This is to certify that the project work titled

"Remote Measurement of Pressure, Temperature & Relative-Humidity"

has been successfully completed by

Roopa P

Srividya B.V

Uma Maheshwari R

Students of the final year,
Bachelor of Engineering (Electronics and Communication)
P.E.S. Institute of Technology, Bangalore
under the guidance of *Mr. R. Ganesan.*

This project is in partial fulfillment of the requirement for the award of Bachelor's Degree in Electronics & Communication Engineering during the academic year 1999-2000.

Dr. D.K. Ravindra
Head, Radio Astronomy Lab.

R. Ganesan
Scientist

P.E.S.INSTITUTE OF TECHNOLOGY
BANGALORE - 560 085
DEPARTMENT OF ELECTRONICS & COMMUNICATION

CERTIFICATE

This is to certify that the project work entitled

"REMOTE MEASUREMENT OF PRESSURE, TEMPERATURE AND RELATIVE HUMIDITY "

has been successfully completed by

ROOPA .P
SRIVIDYA .B.V
UMA MAHESHWARI .R

In partial fulfillment for the award of degree in Bachelor of Engineering in Electronics & Communication of Bangalore University , during the academic year 1999-2000.

Prof. S.Ravishankar
Head of Dept. of E&C

Professor & Head
Dept. of Electronics & Communications
PES Institute of Technology
100 Feet Ring Road, B S.K. 3rd Stage
Bangalore-560 085

Prof. M. V. Satyanarayana
Principal, P.E.S.I.T.

Mrs. B. Rajeshwari
Project Guide

Acknowledgement

It is overwhelming for us to bring out our project report titled **“Remote Measurements of Pressure, Temperature and Relative Humidity”** at the end of our B.E. course.

We are grateful to **Prof. N. Kumar**, Director and **Dr. D.K. Ravindra,** Radio Astronomy Lab of **Raman Research Institute** for having given us an opportunity to undertake our project work in this reputed institute.

We express our gratitude to **Prof. M.V. Satyanarayana** ,Principal of P.E.S.I.T and **Prof. S. Ravishankar,** Head of E & C Dept for the encouragement and help they have given us.

Our sincere thanks to **Mr. R. Ganesan** , for his concern and co-operation given to us to complete the project successfully. Special mention has to be made of **Mr. K. Gurukiran** for his skillful steering and time to time assistance to us in shaping our project.

We would like to extend our thanks to **Mr. K. Ramesh** and **Mrs. Swarna** of RRI and **Mrs. B. Rajeshwari** ,Lecturer, P.E.S.I.T for providing continuous assistance and guidance while developing our project .

We wish to acknowledge the co-operation provided to us by all the staff members of the **Electronics lab** and **DSP lab** during the course of our project. Finally we thank all of them who have directly or indirectly helped us for the successful completion of our project.

CONTENTS

Chapter 1.

Chapter 2.

Chapter 3.

Appendices

INTRODUCTION

1. INTRODUCTION TO RADIO ASTRONOMY

1.1 Radio Astronomy : An Insight

Our knowledge of the universe is based on the observations of the various celestial bodies-stars, pulsars, quasars , supernova to name a few. With the passage of time, human resources have varied from the naked eye to the optical telescope in this heavenly pursuit of knowledge and the field called astronomy has slowly begun to take shape. As research in this field increased, it became clear that the aperture of the telescope has to be very large compared to the wavelength of light in order to make fine observations.

Groundbreaking work was done in this field by a radio engineer at the Bell Telephone Laboratory called Karl Guthe Jansky. In 1932, while studying the direction of thunderstorm static, Jansky was able to detect the origin of a steady Hiss type static, which was previously unaccounted for. He observed that the Hiss is due to the radio waves of extra-terrestrial origin, thus laying a foundation for Radio Astronomy. The importance of these results took a while to be recognized but since then Radio Astronomy has rapidly become a major branch of Astronomy.

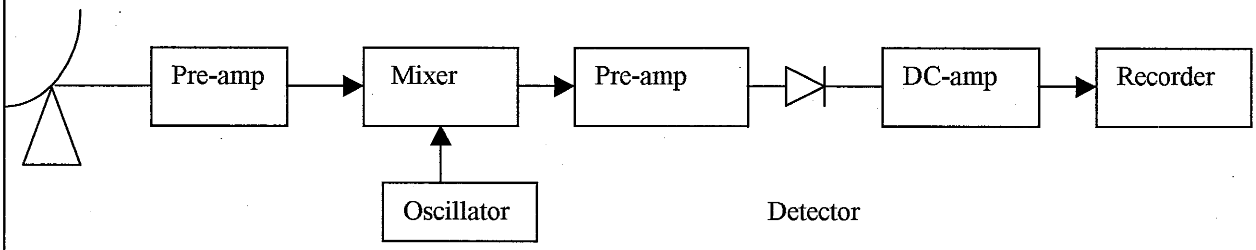
Radio wavelengths are typically about a million times longer than those in optical range. Regions of space that are opaque to light waves because of interstellar dust, generally transparent to radio waves. As Radio Astronomy progressed, the need for Radio Telescopes with better resolution came up. This led to several innovations both in Electronics and Antenna Engineering complementing with each other.

Many pioneering high tech development are due to research done in this field. Spin-off from this area find applications in various fields including satellite communication, space research, image processing and biomedical sciences.

1.2 Principle of Radio Telescope

Radio Frequency waves that can penetrate Earth's atmosphere range from wavelengths of a few millimetres to nearly 10 metres. Although these wavelengths have no discernable effect on human eye or photographic plates, they do induce a very weak electric current in a conductor such as antenna. Most Radio Telescope antennas are parabolic(dish-shaped) reflectors that can be pointed toward any part of the sky. They gather up the radiation and reflect it to a central focus, where the radiation is concentrated. The weak current at the focus can then be amplified by a radio receiver so that it is strong enough to measure and record. Electronic filters in the receiver can be tuned to amplify one range of frequencies at a time or, using sophisticated data processing techniques, thousands of separate narrow bands can be detected. Thus, we can find out what frequencies are present in RF radiation and what are their relative strength. The intensity of RF energy reaching the earth is small compared with the radiation received in the visible range. Thus a radio telescope must have a large 'collecting area', or antenna in order to be useful.

1.3 Block Diagram of a Simple Radio Telescope



ANTENNA

The antenna gathers the minute amount of radio frequency energy from the sky and transforms it to a tiny electrical current. The most widely used antenna is the parabolic dish.

PRE AMPLIFIER

The radio signals are generally very weak. To measure them we have to amplify them by factors of millions of times. The electronic components in the radio telescope produce random electrical noise, which also is amplified by this huge amount. It is easy to see that the noise from the early parts of the chain of amplifiers is multiplied by more than the later stages. If we are not careful, this noise can totally hide the weak noise we are trying to measure from the radio source.

The role of the pre amplifier is to boost this incoming signal from the antenna many times while adding as little noise as possible. The pre amplifier is often called an LNA or low noise amplifier.

MIXER AND LOCAL OSCILLATOR

The job of the mixer is to lower the frequency of the signal from the pre amplifier. We do this for a couple of reasons. First, it is hard to build good amplifiers, filters and other components for higher frequencies. Secondly, if we do all of the amplifying at the frequency which we are receiving, there is a good chance that some of the amplified signal will escape back into the antenna and produce feedback.

The local oscillator produces a signal which is injected into the mixer along with the signal from the antenna in order to effectively change the antenna signal to a frequency which can be handled by the IF amplifier.

IF AMPLIFIER

The intermediate frequency (IF) amplifier is a radio frequency amplifier which processes the output of the mixer. In addition to amplifying the signal, the IF amplifier usually has some of bandpass filtering so that only a selected range of frequencies is allowed through.

DETECTOR

The radio frequency energy exiting from the earlier portions of the receiver alternates in polarity around some central voltage. If we could just hook up a DC (direct current) meter to this signal it would read zero volts because the positive and negative swings of the voltage would cancel each other out. In order to measure the intensity of this signal we must throw half of it away. This can be achieved with the help of a semiconductor diode. If we pass just the right range of the current through these diodes, the voltage we measure coming out of them will be the square of the input, and thus will be proportional to the power which is fed to them from the receiver. It is the power received by the radio telescope antenna that we want to measure and we owe to detector the ability to measure it.

DC AMPLIFIER

Once the radio frequency energy has been converted to a DC signal by the detector, we need to digitize to make it easier to record.

1.4 **The Radio Telescope at Raman Research Institute**

The Raman Research Institute, Bangalore can stake its claim to being one of the finest research institute in the whole country. It boasts of its research facility in Astronomy and Astrophysics including a fully equipped radio astronomy laboratory complete with a millimetre wave radio telescope on campus and two other radio telescopes in Gauribidanur and in Mauritius.

A radio telescope operates in the radio band of the electromagnetic spectrum thus making it suitable for observing celestial bodies.

The main components are :

* an antenna with its feed that selectively collects the radio power from a narrow solid angle

* a low noise receiver that amplifies the received signal over a restricted frequency band, detects, correlates and integrates information and stores the output in digital form. Following are a few important specifications of the above mentioned telescope :

Antenna type	: Parabolic Reflector
Diameter	: 10.4 metres
Beam Width	: 80 arc seconds at 86 GHz
Normal frequency range of operations	: 1.4 GHz , 6.7 GHz & 85 - 115 GHz

REMOTE MEASUREMENT

OF P,T &RH

2. REQUIREMENT FOR REMOTE MEASUREMENT

The 10.4metre millimetre wave Telescope at Raman Research Institute is being used for Astronomical Observations from 1.4 GHz to 115 GHz. The beam size varies from ~1.5 degree at the lowest frequency to ~1 arcminute at the highest frequency. The positional accuracy required at the highest frequency is ~ 5 arcseconds, to achieve such an accuracy the pointing of the telescope has to take into account the refraction through the atmosphere and the elevation angle has to be appropriately corrected for it.

The atmosphere consists of several layers of different densities. As we travel towards the earth the layers are more denser. We know that, refraction takes place when a ray of light travels from a rarer to a denser medium. Thus when observing a star or any celestial body, the radiation from that particular body, as it passes through these layers gets refracted. Thus due to this refraction the star appears at a position which it is not at. The correction between its actual position and its refracted position is important, for the pointing of the telescope. This correction is referred to as refraction correction.

Any error in the refraction correction will seriously affect the pointing, especially at low elevations for millimetrewave observations. The refraction correction calculation using an atmospheric model needs accurate values of atmospheric pressure, temperature and relative humidity existing at the time of observation.

In the present setup, the pressure is read from the barometer, temperature from the Observer's desk display and relative humidity is measured using hygrometer. All these values are then manually fed into the observing program. Apart from inherent errors in calibration, etc any lapse in periodic update during observations can often lead to pointing errors and associated drop in efficiency.

Our project involved implementing an automated setup in which these values will be obtained from remotely mounted pressure, temperature and relative humidity sensors through their 4 – 20 mA current output. In the control room the control PC (CPC) will read these values through a ADC card. With these acquired values the control PC calculates the refraction correction and thus controls the alignment of the telescope.

Why 4 – 20 mA current output?

In many process control applications information is transmitted in the form of current with a span of 16mA full scale and an offset range of 4 – 20mA. The transmission of current provides a degree of noise immunity since the received information is unaffected by voltage drops in the line, stray thermocouples, contact voltage or resistance and induced voltage and noise. At the same time the offset provides a distinction between zero (represented by 4mA) and no information due to an open circuit (zero current flow).

An additional benefit of this form of transmission is that for some application power can be furnished remotely via the 4mA of current that is not needed for information transfer. Thus power is transmitted in one direction – signal in the return direction.

No local source of power is needed at such a transducer and only two wires are needed for transmission.

Finally, information in the form of current permits several loads at different locations to be connected in series upto a specified maximum voltage. For example, the output of the transducer could drive a chart recorder and a meter and provide an input to a controller.

2.1 OVERVIEW

The digital computers in current times provide computation, data processing and decision capability but also does the real time signal processing. We know that most of the real world signals are analog in nature and are distinguished by their continuous levels. Their levels cannot be directly read by the computers, which requires a need for translation from analog to digital.

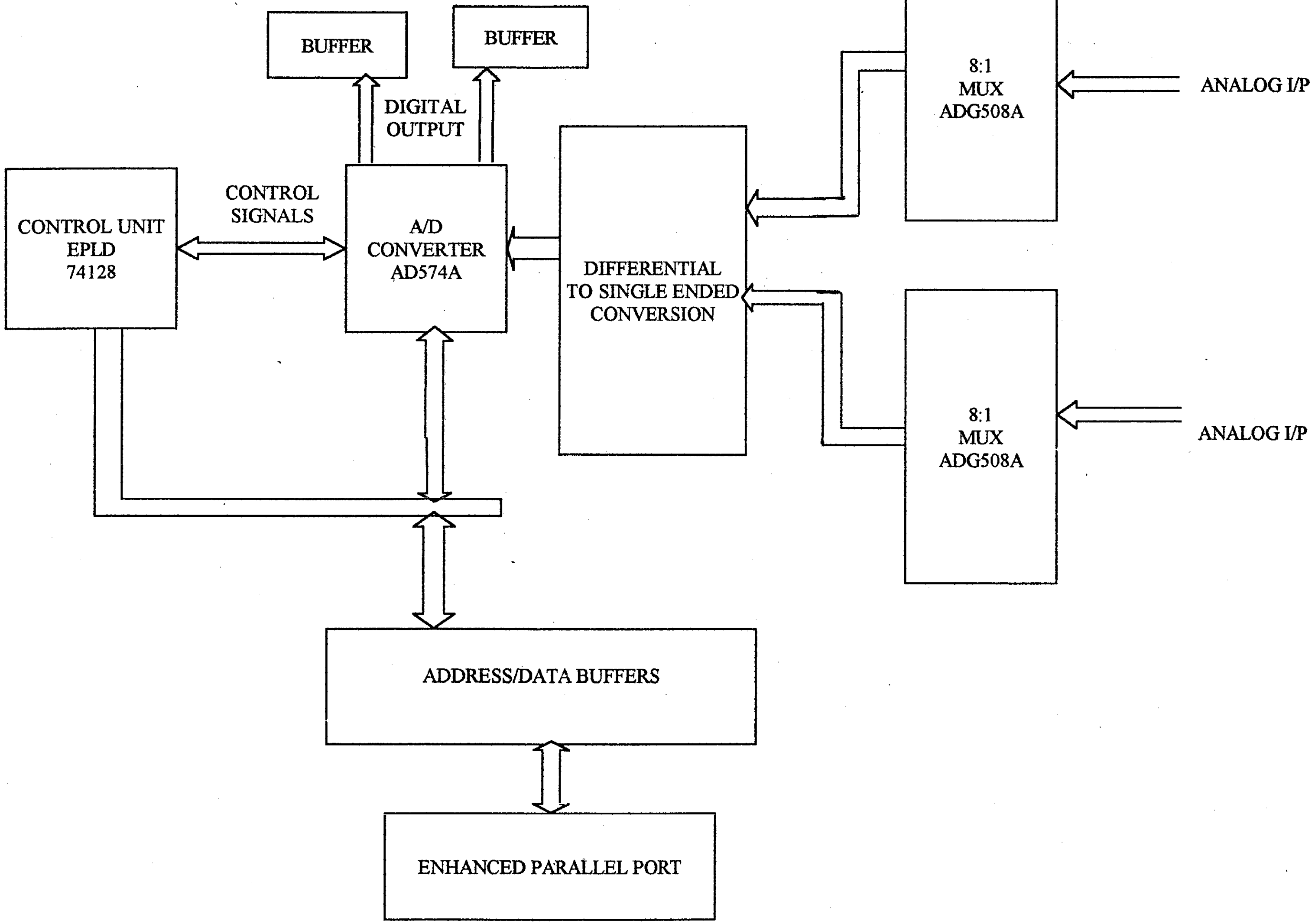
Thus, data acquisition system interfaces between the real world of physical parameters and the artificial world of digital computation and control. Digital systems are widely used because of its accuracy and reduces the circuit complexity. Data acquisition systems are being used in large number of fields in a variety of industrial and scientific areas, including aerospace, biomedical and telemetry industries.

In the data acquisition system developed, the workstation reads the accurate values of the Pressure, Temperature and Relative Humidity from a prototype card interfaced to the Personal Computer through the Enhanced Parallel Port(EPP). The output of the sensors are fed into two 8:1 analog multiplexers. The hardware can be configured for either single or differential ended outputs. The selected channel's output is fed into the Analog to Digital Converter(ADC). The start of conversion signal for the ADC is generated by the EPP. The control signals necessary for the selection of the channels and conversion are generated by the Control Unit which is implemented using a Erasable Programmable Logic Device(EPLD). The read cycle initiated by the EPP reads the values from the card and is displayed on the PC. With these acquired values the Control PC calculates the refraction correction and thus controls the alignment of the telescope.

A brief introduction to the sensors, EPLD and EPP are dealt in the forthcoming sections.

HARDWARE SECTION

PROJECT SETUP



3. HARDWARE

3.1 HARDWARE DESCRIPTION

3.1.1 SENSORS

What is a sensor?

A sensor is a transducer that provides a usable electrical output in response to a specific physical quantity. For example, a device used to sense temperature is a temperature sensor and to sense pressure it is pressure sensor. A sensor in its simplest form may be regarded as a system with an input x(t) and output y(t). Fig 3.1 shows a system representation of (a) self exciting sensor, has its output energy supplied entirely by the input signal x(t).

(b) modulating sensor, has its output energy is dependent on both the external drive x(d) and input signal x(t).

a) Self exciting sensor

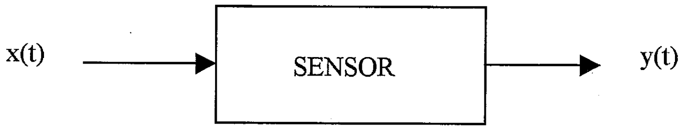
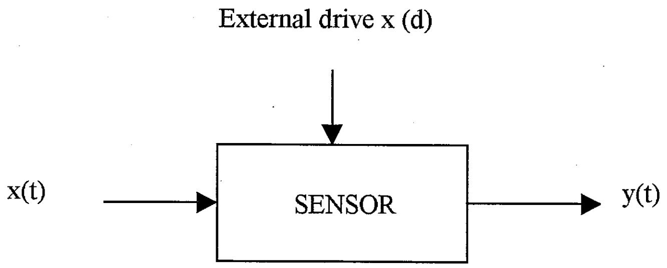


Fig 3.1(a)

b) Modulating sensor



Some of the important characteristics of a transducer are as follows:

Accuracy:

Accuracy is defined as the ratio of error to full scale output usually expressed in the term "within ± ______%FSO (Full Scale Output)", sometimes in terms of units of measurand or in terms of the error / output ratio.

Hysterisis:

Hysterisis is the maximum difference in output at any measurand value within the specified range, when the value is approached first with increasing and then with decreasing measurand. Many types of transducers exhibit hysterisis which is typically caused by a lag in the action of the sensing element. Hysterisis is expressed as % FSO.

Repeatability:

Repeatability is the ability of a transducer to reproduce output readings when the same measurand value is applied to it consecutively under the same condition and in the same direction. It is expressed as the maximum difference between output readings as determined by two calibration cycles unless otherwise specified and is usually expressed as "within ±_____% FSO". If the sampling is increased by increasing the number of calibration cycles, a better statistical measure of repeatability can be obtained.

Linearity:

Linearity is the closeness of a transducer's calibration curve to a specified straight line. It is expressed as "within ±_____FSO".

Resolution:

Resolution is descriptive of “smallest increment”. When the measurand is continuously varied over the range of output of certain transducers will not be perfectly smooth but instead will change in small steps. The magnitude of the output step changes as the measurand is continuously varied over the range is the resolution of the transducer. It is expressed in %FSO.

Sensitivity:

Sensitivity is simply the ratio of the change in the output to the change in the value of the measurand.

Temperature Sensors

Temperature is a measure of the kinetic energy of the molecules of a substance due to heat agitation. Absolute temperature is temperature measured on the thermodynamic scale – temperature measured from absolute zero(0 k , -273.15° C).

The most widely used semiconductor temperature sensing elements are thermistors.

Thermistors are characterized by the small size, high negative temperature coefficient and fast time constant. When used for temperature measurement, the current flowing through the thermistor must be kept very low to assure near zero power dissipation and zero self heating. Thermistors are designed to produce a resistance change, or voltage output that varies linearly with temperature over selected temperature ranges when used with a regulated voltage source.

Relative Humidity Sensors

Relative Humidity is the ratio of the water vapour pressure actually present to the water vapour pressure required for saturation at a given temperature. This ratio is expressed in percent and is temperature independent. Relative Humidity is expressed in percent(%RH). Relative Humidity is usually measured with capacitance hygrometer. The principle is that capacitance of capacitor, where an element has hygroscopic characteristics, will vary with amount of water retained by the electrode. The change in capacitance is determined by a bridge circuit and indicated in terms of relative humidity by a directly calibrated meter. Good accuracy of measurement is possible over a wide range of humidity and temperatures. Thin film capacitive sensors are widely used as humidity sensors. The absorption element is a very thin dielectric polymer (typically 1 micron thick) applied to the lower electrode. The upper electrode is a metal film vacuum encapsulated on to the polymer surface through a mechanical mask. The thickness of the upper layer is chosen as compromise between response time and ohmic loss.

Pressure Sensors

Pressure is defined as the force acting on unit area. It is measured as force per unit area exerted at a given point; expressed as Pascal. Absolute pressure is measured relative to zero pressure. Differential pressure is the pressure difference between two points of measurement, measured relative to a reference pressure or a range of reference pressures.

Piezo resistive pressure transducers are widely used for pressure measurements. It consists essentially of a conductor /semiconductor of small cross-section, which is mounted on to the measured surface, so that it undergoes the small elongations/contractions due to the tension/ compression stresses respectively in that surface. As a result, the sensor undergoes a corresponding change in resistance due to stress and this is called piezo resistive effect. The resistance change is usually converted into voltage, by connecting it as arms of a wheatstone's bridge.

3.1.2 ERASABLE PROGRAMMABLE LOGIC DEVICES

Programmable Logic Overview

Programmable logic devices combine the logistical advantages of standard, fixed integrated circuits with architectural flexibility of custom devices. These devices allow engineers to electrically program standard, off-the-shelf logic elements to meet the specific needs of their applications.

Proprietary logic functions can be designed and fabricated in-house, eliminating the long engineering lead times, high tooling costs, complex procurement logistics, and dedicated inventory problems associated with custom Application Specific Integrated Circuit (ASIC) devices, such as gate arrays and standard cells. The key to this 'off-the-shelf ASIC ' ability is reprogrammable CMOS technology, which is used to create EPLDs. Altera has taken advantage of speed and density advances in CMOS memory products to create sophisticated EPLDs that solve many logic design problems. Altera provides the broadest line of CMOS EPLDs in the industry, with products ranging in density from 300 to 40000 maximum gates, offered in a variety of packages with 20 to 288 pins. These EPLDs , together with Altera development software, enable system manufacturers to create custom logic functions for a wide variety of applications. EPLDs can be used to integrate complete PCBs of TTL, PAL and FPGA devices into a single package.

CLASSIFICATION

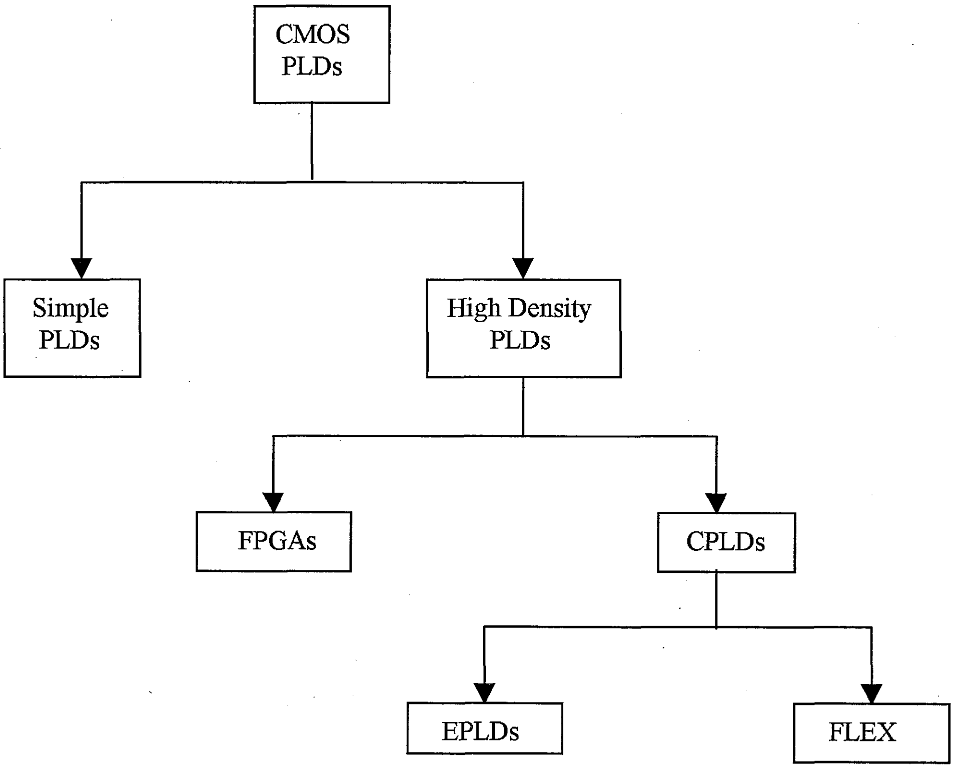
The CMOS programmable logic devices are classified into :

* Simple PLDs.

* High density PLDs.

Their classification is as shown in the block diagram.

BLOCK DIAGRAM



A few examples of the EPLDs are :

* MAX 9000 * MAX 7000

* MAX 5000 * FLASH LOGIC * CLASSIC

CPLDs and FPGAs have different interconnect structures. The segmented interconnect structures of FPGAs uses multiple metal lines of varying lengths connected by pass transistors or anti-fuses to connect logic cells. In contrast, the continuous structure of CPLDs uses continuous metal lines to provide logic cell to logic cell connectivity. The continuous interconnect structure eliminates the unpredictable timing associated with the segmented interconnected structure and provides fast, fixed delay paths between logic cells. The continuous interconnect structure makes it easier to implement or design and thus shortens the development cycle.

MAX 7000 FAMILY

Altera offers seven families of general-purpose PLDs : FLEX 10K , FLEX 8000 , MAX 9000, MAX 7000, FLASH Logic, MAX 5000 and Classic devices. The Flexible Logic Element Matrix (FLEX) architecture uses look-up tables (LUTs) to implement logic functions , whereas the Multiple Array Matrix (MAX), FLASH logic and Classic architectures user programmable-AND / fixed - OR product term architecture. Each device family offers unique features as well as distinct speed and utilization advantages for implementing particular applications.

The MAX 7000 family of high density, high performance programmable logic devices is based on Altera's second generation MAX architecture.

Fabricated with advanced CMOS technology , the EEPROM-based MAX 7000 family provides 600 –5000 usable gates , pin to pin delays as fast as 5ns , and counter speeds of upto 179MHz. The higher density members of the MAX 7000 family–called MAX 7000E devices–include the EPM7128E , EPM7160E , EPM7192E and EPM7256E devices. These devices have several enhanced features : additional global clocking, additional output enable controls, enhanced interconnect resources , fast input registers and a programmable slew rate.

The MAX 7000 architecture supports 100% TTL emulation and high density integration of SSI , MSI and LSI logic functions. MAX 7000 devices use CMOS EEPROM cells to implement logic functions. The user configurable MAX 7000 Architecture accommodates a variety of independent combinatorial and sequential logic functions. The devices can be reprogrammed for quick and efficient iterations during design development and debug cycles, and can be programmed and erased over a 100 times. MAX 7000 devices contain from 32 to 256 macrocells that are combined into groups of 16 macrocells, called Logic Array Blocks (LABs).

Each macrocell has a programmable-AND / fixed-OR array and a configurable register with independently programmable clock , clock enable , clear and preset functions. To build complex logic functions, each macrocell can be supplemented with both shareable expander product terms and high speed parallel expander product terms to provide upto 32 product terms per macrocell.

The MAX 7000 architecture includes the following elements :

* Logic Array Blocks

* Macrocells

* Expander product terms(shareable and parallel)

* Programmable Interconnect Array

* I/O Control blocks

The MAX 7000 architecture includes four dedicated inputs that can be used as general purpose inputs or as high speed global control signals (clock, clear and two output enable signals) for each macrocell and I/O pin.

LOGIC ARRAY BLOCKS

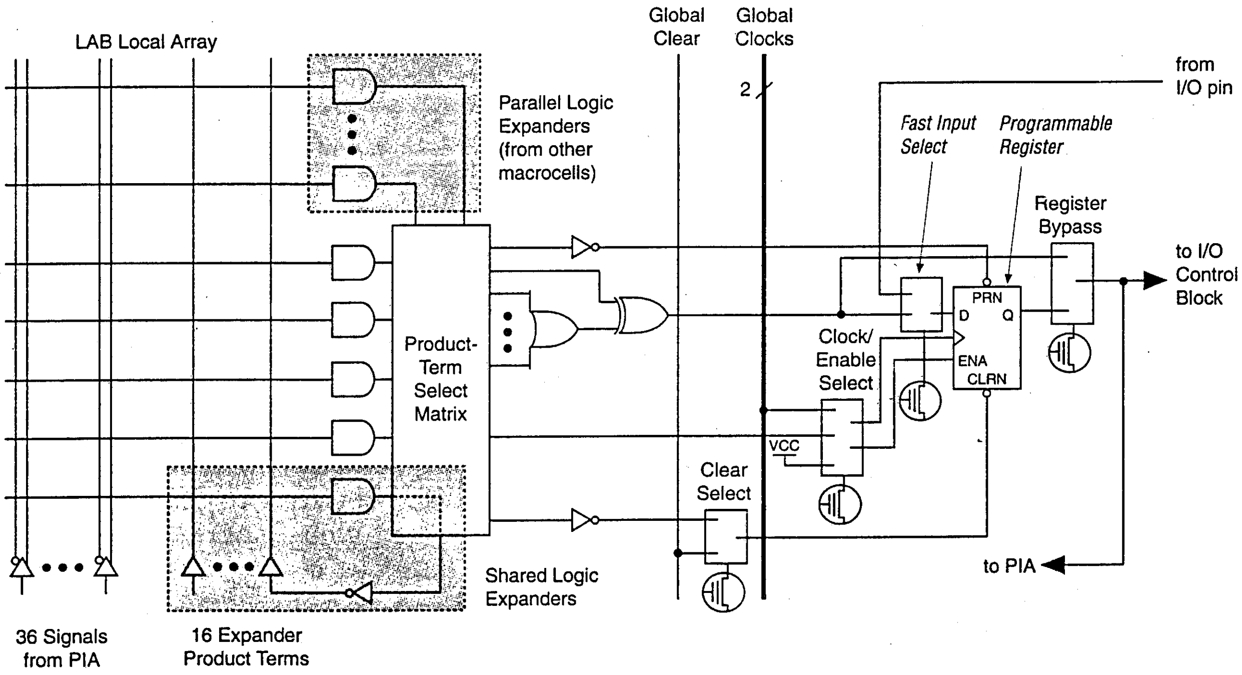
The MAX 7000 device architecture is based on the linking of high performance, flexible, logic array modules called logic array blocks (LABs). LABs consist of 16 macrocell arrays. Multiple LABs are linked together via the programmable interconnect array (PIA), a global bus that is fed by all dedicated inputs I/O pins and macrocells. Each LAB is fed by the following signals :

* 36 signals from the PIA that are used for general logic inputs .

* Global controls that are used for secondary register functions .

* Direct input paths from I/O pins to the registers that are used for fast setup times.

Figure 4. MAX 7000E & MAX 7000S Device Macrocell



The MAX 7000 macrocell can be individually configured for both sequential and combinatorial logic operation. The macrocell consists of three functional blocks : the logic array, the product-term select matrix and the programmable register. Combinatorial logic is implemented in the logic array which provides five product terms per macrocell. The product term select matrix allocates these product terms for use as either primary logic inputs (to the OR and EXOR gates) to implement combinatorial functions, or as secondary inputs to the macrocell's register clear, preset, clock and clock enable control functions. Two kinds of expander product terms are available to supplement macrocell logic resources :

* Shareable expanders, which are inverted product terms that are fed back into the logic array.

* Parallel expanders, which are product terms borrowed from adjacent macrocells.

For registered functions, each macrocell flip-flop can be individually programmed to implement D , T , JK or SR operation with programmable control . The flip-flop can be bypassed for combinatorial operation. During design entry, the designer specifies the desired flip-flop type ; MAX +PLUS II then selects the most efficient flip-flop operation for each registered function to optimize resource utilization.

Each programmable register can be clocked in three different modes :

* By a global clock signal. This mode achieves the fastest clock to output performance.

* By a global clock signal and enabled by an active high clock enable. This mode provides an enable on each flip-flop while still achieving the fast clock-output performance of the global clock.

* By an array clock implemented with a product term. In this mode, the flip-flop can be clocked by signals from buried macrocells or I/O pins.

Each register also supports asynchronous preset and clear functions. The product term select matrix allocates product terms to control these operations. Although the product term driven preset and clear of the register are active high, active low control an be obtained by inverting the signal within the logic array. In addition, each register clear function can be individually driven by the active low dedicated global clear pin (GCLRn) . All MAX 7000E I/O pins have a fast input path to a macrocell register. This dedicated path allows a signal to bypass the PIA and combinatorial logic and be clocked to an input D flip-flop with an extremely fast (3ns) input setup time.

PROGRAMMABLE INTERCONNECT ARRAY (PIA)

Logic is routed between LABs on the programmable interconnect array. This global bus is a programmable path that connects any signal source to any destination in the device. All MAX 7000 dedicated inputs, I/O pins and macrocell outputs feed the PIA, which makes the signals available throughout the entire device. Only the signals required by each LAB are actually routed from the PIA into the LAB. An EEPROM cell controls one input to a two input AND gate, which selects a PIA signal to drive into the LAB. The MAX 7000 PIA has a fixed delay. The PIA thus eliminates skew between signals and makes timing performance easy to predict.

I/O CONTROL BLOCKS

The I/O control block allows each I/O pin to be individually configured for input, output or bi-directional operation. All I/O pins have a tri-state buffer that is individually controlled by one of the global output enable signals or directly connected to Gnd or Vcc. The I/O control block of MAX 7000E devices has 6 global output enable signals that are driven by the true or complement of two output enable signals, a subset of the I/O pins , or a subset of the I/O macrocells.

MAX +PLUS II

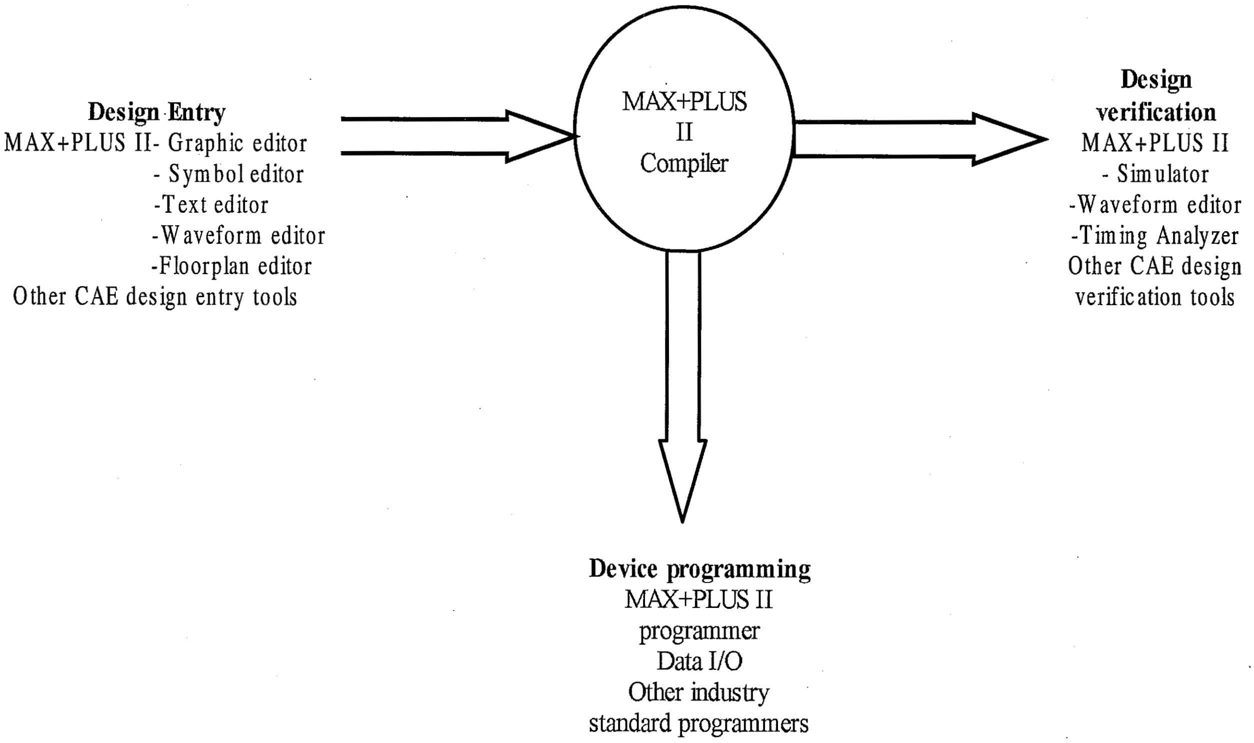
MAX+PLUS II development software is a fully integrated package for designing Altera programmable logic devices-including the MAX 5000, MAX 7000, FLEX 8000 and FLEX 10K families of devices.

The Altera Multiple Array MatriX Programmable Logic User System (MAX+PLUS II) provides a multi platform, architecture independent design environment that easily adapts to our specific needs. MAX+PLUS II offers easy design entry, quick processing and straight forward device programming. MAX+PLUS II offers a full spectrum of logic design capabilities: a variety of design entry methods for hierarchical designs, powerful logic synthesis, timing driven compilation, partitioning, functional and timing simulation and device programming and verification. It also offers a rich graphical user interface complemented with an illustrated, easy to use online help system.

The MAX+PLUS II compiler lies at the heart of the MAX+PLUS II system, providing powerful project processing that we can customize to achieve the best possible silicon implementation of the project. Automatic error location and extensive documentation on error and warning messages make design modifications quick and easy.

Many features and commands—such as opening files; entering device, pin and logic cell assignments; and compiling the current project—are shared by many or all MAX+PLUS II applications.

MAX+PLUS II Design Environment



The Design Flow

1. Create a design file with MAX+PLUS II graphic, text and waveform editors.
2. Assign a device family for the project or allow the compiler to select a device.
3. Compile the project.
4. Perform timing and simulation analysis.
5. Insert the device into a programming adapter on the master programming unit(MPU) and choose the **Program** button to program.

3.1.3 PARALLEL PORTS

The parallel port is the most commonly used port for interfacing. This port will allow an input of upto 9 bits and an output of upto 12 bits at any given time. The port is composed of 4 control lines, 5 status lines and 8 data lines. The control lines are used as interface control and handshaking signals from PC to the peripheral. The status signal are used as handshaking signals and as status indicators for such things as paper empty ,busy indication , and interface or peripheral layers. The data lines are used to provide data from the PC to the printer and also for data to be driven from the peripheral to the PC.

The parallel port is included in the IEEE 1284 standard. The use of the various 1284 data transfer modes provide the capability to create a forward and reverse channel connection between the host computer and attached peripheral. Since there is only one set of data lines the connection is half duplex, data is transferred in one direction at a time.

The common modes in which the parallel port is used are:

1. Compatibility Mode(Standard parallel Port).
2. Enhanced Parallel Port (EPP) Mode.
3. Extended Capabilities Mode (ECP).

COMPATIBILITY MODE:

Compatibility mode or Centronics mode as it is commonly known can only send data in the forward direction at a typical speed of 50KBps but can be as high as 150KBps. Centronics is an early standard for transferring data from the host to peripheral.

To output a byte of data the software must:

1.Write the byte to the data port.

2.Check to see if the peripheral is busy.

3.If the peripheral is busy it will not accept any data, thus any data written will be lost.

4.Take the strobe(pin 1) low. This tells the peripheral that there is correct data on the data lines.

5.Put the strobe high again after waiting for approximately 5μs after putting the strobe low. This limits the speed at which the port can run at.

The EPP and ECP gets around this by letting the hardware check to see whether the peripheral is busy and generate the appropriate signals for handshaking.

This means only one I/O instruction needs to be performed, thus increasing the speed.

EPP – Enhanced Parallel Port

The Enhanced Parallel Port(EPP) was designed in a joint venture between Intel, Xircom and Zenith Data Systems. EPP ports were first specified in the EPP 1.7 standard, and then later include in the IEEE 1284 Standard released in 1994. EPP has a typical transfer rate in the order of 500KB/S to 2MB/S. this is achieved by the allowing the hardware contained in the port to generate handshaking, strobing etc, rather that have the software do it, which was the case with Centronics.

EPP is most commonly used than ECP. EPP differs from ECP by the fact that the EPP generates and controls all the transfers to and from the peripherals. ECP on the other hand requires the peripheral to negotiate a reverse channel and control the handshaking .This is harder to achieve with common glue logic, thus really requires a dedicated controller or ECP peripheral chip.

The EPP Handshake

The EPP connector is of 25 pins. The various pin assignments and the details of the EPP Data Read, EPP Data Write, EPP Address Read , EPP Address Write cycles are given in the appendix attached at the end. In order to perform a valid exchange of data using EPP we must follow the EPP handshake. As the hardware does all the work, this handshake only requires to be used for the hardware and not for the software as the case with Centronics. To initiate an EPP cycle the software needs to perform only one I/O operation to the relevant

EPP Register. The basic difference between the two EPP standards EPP 1.7 and EPP 1.9 is in the handshake signals.

If implementing EPP 1.7 Handshake (Pre IEEE 1284) the Data and Address Strobes can be asserted to start a cycle regardless of the Wait State. EPP 1.9 will only start a cycle once wait is low. Both EPP 1.7 and EPP 1.9 require the wait to be high to finish a cycle.

The EPP's Software Registers

The EPP port has a set of seven registers. However 3 of them have been inherited from the Standard Parallel Port. Below is a table showing the registers.

Address	Port Name	Read/Write
Base + 0	Data Port (SPP)	Write
Base + 1	Status Port (SPP)	Read
Base + 2	Control Port (SPP)	Write
Base + 4	Address Port (EPP)	Read/Write
Base + 5	Data Port (EPP)	Read/Write
Base + 6	Undefined(16/32 bit transfer)	--
Base + 6	Undefined(32 bit transfer)	--
Base + 7	Undefined(32 bit transfer)	--

As we can see, the first three addresses are exactly the same than the Standard Parallel Port register and behave in exactly the same way. Therefore if we used a EPP, we can output data to Base + 0 in exactly same fashion as SPP i.e. then we have to check to see if the port is busy and then assert and de-assert the strobe using Control and Status Port , then wait for the acknowledgement.

If we wish to communicate with a EPP compatible device the all we have to do, is place any data we wish to send in the EPP Data register at Base + 4 and then the card will generate all the necessary handshaking required. Likewise if we wish to send an address to the device, then we use the EPP Address register at offset + 3.Both the EPP Address register and the EPP Data register are read/write, thus to read data from the device we can use the same registers. However the EPP card has to initiate a read cycle as both the Data Strobe and Address Strobe are outputs.

Bit 0 of Status Port , which was reserved in the SPP register set, now becomes the EPP Time-out Bit. This bit will be set when an EPP time-out occurs. This happens when the nWait line is not deasserted within approximately 10µs (depending upon the port) of the IOW or IOR line being asserted. The IOW or IOR are the I/O Read and Write lines present on the ISA bus.

The EPP mode is very dependent on the ISA bus timing. When a Read cycle is performed , the port must undertake the appropriate Read/write handshake and return the data in that ISA cycle. Of course this does not occur within one ISA cycle , thus the port uses the IOCHRDY(I/O channel ready) on the ISA bus to introduce wait states, until the cycle completes. If a EPP read or write is started with no peripheral connected , the port never gets an acknowledgement (Wait), thus keeps sending requests for wait states , and the computer locks up. Therefore the EPP implements a type of watchdog, which times out after approximately 10μs.

The three registers, Base + 5, Base + 6, Base + 7 can be used for 16 and 32 bit read/write operations if the port supports it. This can further reduce the I/O operations. The parallel port can only transport 8 bits at a time , thus any 32 or 16 bit word written to the parallel port will be split into byte size blocks and sent via the parallel port's 8 data lines.

EPP Programming Considerations

Before we can start any EPP cycles by reading and writing to the EPP data and address ports, the port must be configured correctly. In the idle state, an EPP port should have it's Address Strobe, Data Strobe, Write and Reset lines inactive, high. Some ports require to set this before starting any EPP cycle.

Therefore, our first task is to manually initialize these lines by writing XXXX0100 to the control port. The details of the different bits of the control port is provided in the appendix.

3.2 HARDWARE DESIGN AND IMPLEMENTATION

The various components of the data acquisition system are mounted on a prototype card . The hardware designed uses 8 bit data bus as the card is interfaces through Enhanced Parallel Port. The necessary control signals are generated by the control unit.

3.2.1. SELECTION OF SENSORS

The specifications laid out for the sensors were as follows:

SENSOR	RANGE	ACCURACY
TEMPERATURE	0-60°C	±0.5°C
PRESSURE	700-1000 mbar	10 m bar
RELATIVE HUMIDITY	10-99 %	1%

All the sensors were required to have a current output in the range of 4 –20 mA along with a digital panel indicator. We found that the sensors and indicators supplied by Instrument Research Associates would meet our requirements.. The specifications of these are given below:

DIGITAL PRESSURE INDICATOR

PRM – 300M

General Description:

PRM – 300M is designed to measure gauge (relative) and differential Pressure with good repeatability , accuracy & stability using semiconductor Piezo resistive property. The Specifications are :

SENSOR:

RANGE	:	0 –1 Bar (abs)
ACCURACY	:	± 0.3 % F.S

INDICATOR:

DISPLAY	:	3½ digit, 0.5" LED Display
WARM UP TIME	:	Instantaneous
SAMPLING RATE	:	3 samples /sec.
THERMAL ERROR OF SENSOR	:	± 0.1 % F.S. per °C

The indicator is calibrated for a resistance of 10Ω

TEMPERATURE AND RELATIVE HUMIDITY INDICATOR

HUMITHERM–842MR

General Description:

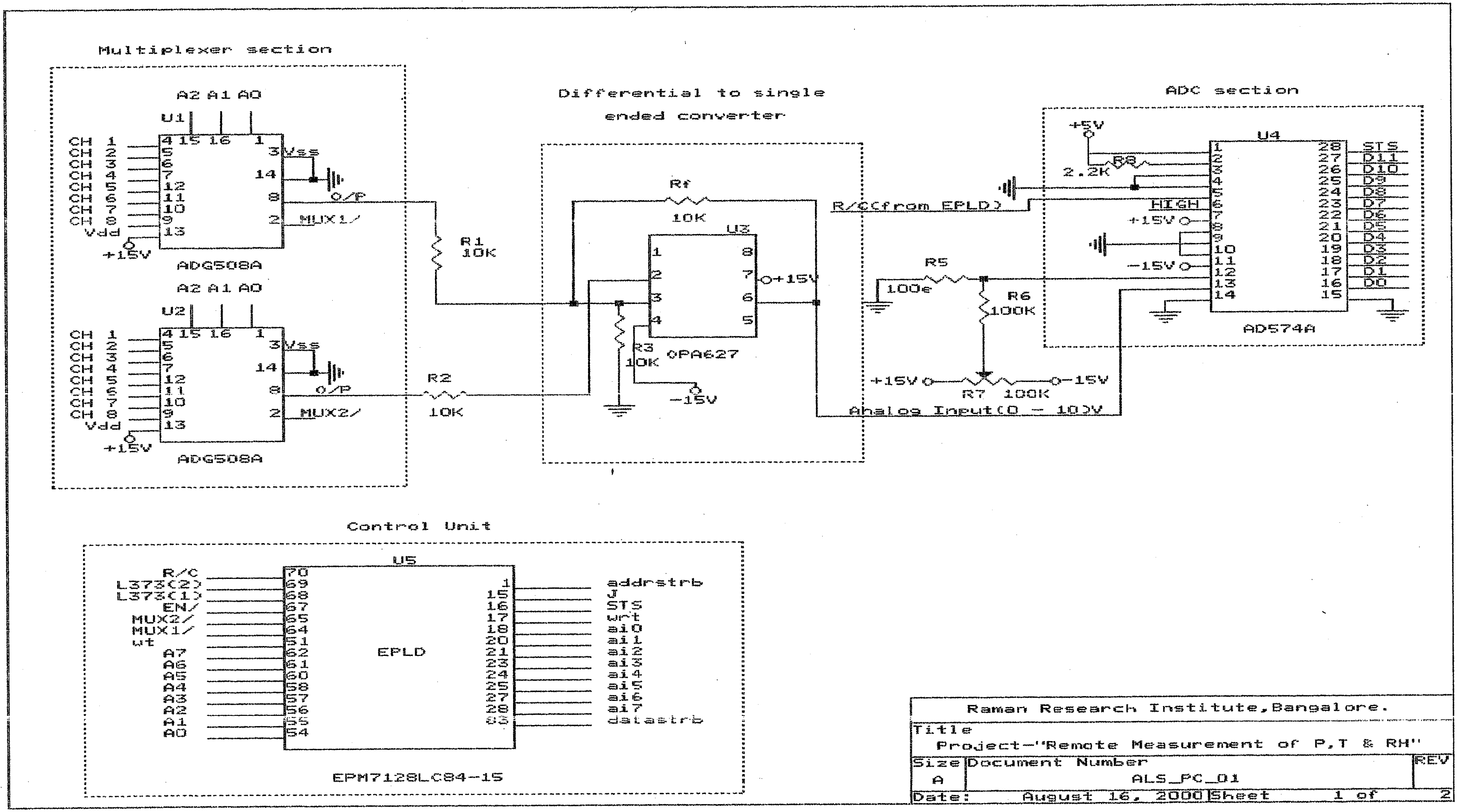
HUMITHERM –842MR is an accurate and reliable Indicator to measure Relative Humidity and Dry Bulb Temperature. It works in conjunction with a transmitting type combined RH and Temperature sensor, HUMITHERM-842TC. It employs a solid state macromolecular RH sensor for sensing humidity and a solid state temperature sensor for sensing the dry bulb temperature. The specifications are as follows:

PARAMETER	RELATIVE HUMIDITY	TEMPERATURE
SENSOR	CAPACITIVE	SOLID STATE
RANGE	0% TO 99% RH	0 TO 50 deg C.
ACCURACY	±2% RH at 25 deg C	± 0.5 deg C
OUTPUT	4-20 mA	4-20 mA
HYSTERISIS	± 4% RH for cycling of 10 to 90% and 90% to 10% RH.	-

INDICATOR:

DISPLAY	:	3 digit, 0.5" LED Display
WARM UP TIME	:	Instantaneous
SAMPLING RATE	:	3 samples /sec.

Both the indicators are calibrated for a resistance of 100Ω

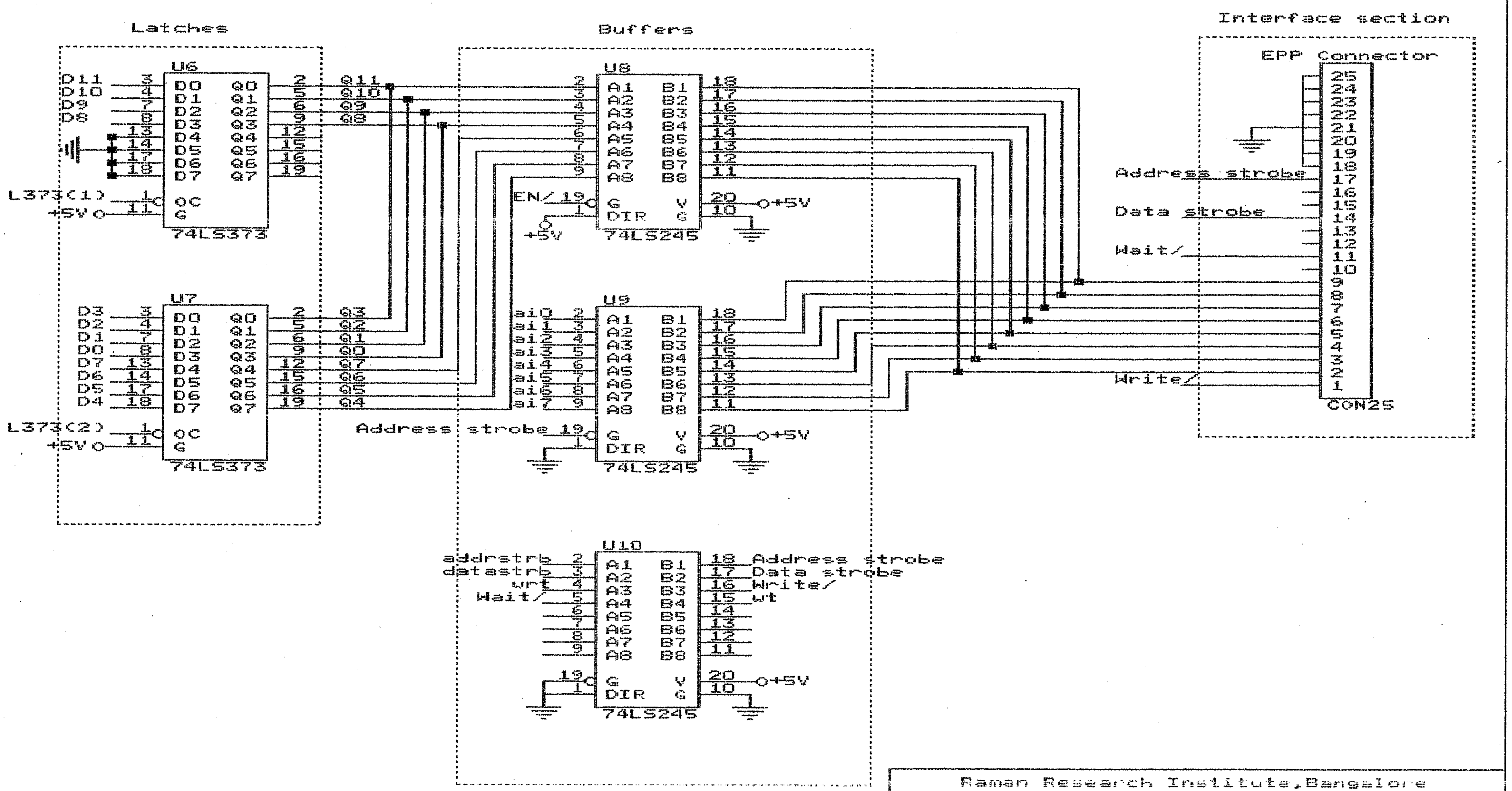


Raman Research Institute,Bangalore.

Title
Project-"Remote Measurement of P,T & RH"

Size A | Document Number ALS_PC_01 | REV

Date: August 16, 2000 | Sheet 1 of 2



Raman Research Institute,Bangalore

Title

Project-''Remote Measurement of P,T & RH''

Size	Document Number	REV
A	ALS_PC_01	
Date: August 16, 2000	Sheet 2 of 2	

3.2.2 Multiplexer section:

The requirement of the hardware was that, it should be capable of handling 8 differential or 16 single ended inputs.

In many analog to digital applications it becomes too expensive to dedicate one ADC to each of the channel therefore, it is always feasible to have one ADC which can be shared by all the analog signals through a common path. This process is known as multiplexing. Since many signals share a common transmission path to the receiving device. The multiplexer can be visualized as a simple rotary switch having multiple taps which can be switched from one position to another manually or by electronic means. In each case, the communication path is dedicated to one channel for a short period of time before switching to next channel. At any point of time, only one channel uses the transmission path.

The requirement of 8 differential or 16 single ended inputs are met by selecting two 8:1 analog mux. The MUX used are ADG508A. Even though our project involved the usage of only 3 channels, the remaining channels will be used by the institute to monitor other slow varying signals. The outputs of the sensors are given as input to the mux through the connector.

The hardware can be configured for single ended or differential ended output depending on the jumper setting and the bit A3 of the address given by the EPP. In the single ended mode only one of the mux is selected. In the differential ended mode both the multiplexers are selected. The multiplexers are enabled by the control signal MUX1/ and MUX2/ generated by the EPLD.

A3	Jumper Setting	Mux Enabled	Operation
0	1 & 2 shorted	MUX 1	Single ended
0	2 & 3 shorted	MUX 1 & MUX 2	Differential ended
1	1 &2 shorted	MUX 2	Single ended
1	2 & 3 shorted	MUX 1 & MUX 2	Differential ended

Table 1

The channel selection of the MUX is dependent on the bits A0, A1, A2 (select lines) of the 8 bit address written on to the data lines of the EPP. The channel selection of the single ended and differential mode of operation is given in the following tables:

Channel Selection of Multiplexer:

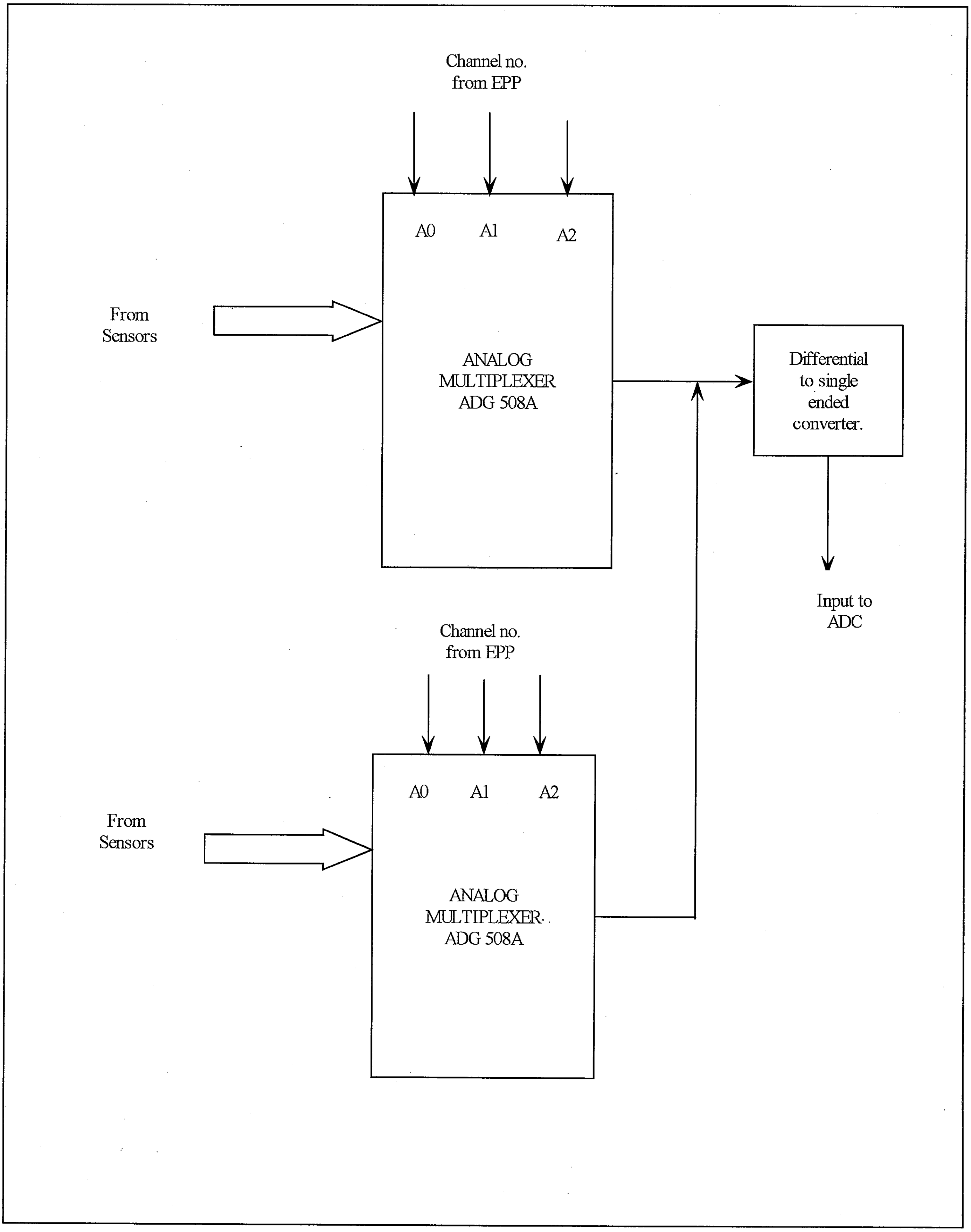
(a) Single ended mode:

MUX1/	MUX2/	A2	A1	A0	Channel Selected
0	1	0	0	0	Channel 1
0	1	0	0	1	Channel 2
0	1	0	1	0	Channel 3
0	1	0	1	1	Channel 4
0	1	1	0	0	Channel 5
0	1	1	0	1	Channel 6
0	1	1	1	0	Channel 7
0	1	1	1	1	Channel 8
1	0	0	0	0	Channel 9
1	0	0	0	1	Channel 10
1	0	0	1	0	Channel 11
1	0	0	1	1	Channel 12
1	0	1	0	0	Channel 13
1	0	1	0	1	Channel 14
1	0	1	1	0	Channel 15
1	0	1	1	1	Channel 16

b) Differential mode

MUX1/	MUX2/	A2	A1	A0	Channel Selected
0	0	0	0	0	Channel 1
0	0	0	0	1	Channel 2
0	0	0	1	0	Channel 3
0	0	0	1	1	Channel 4
0	0	1	0	0	Channel 5
0	0	1	0	1	Channel 6
0	0	1	1	0	Channel 7
0	0	1	1	1	Channel 8

Table 2


Channel no.
from EPP
A0
A1
A2
From
Sensors
ANALOG
MULTIPLEXER
ADG 508A
Differential
to single
ended
converter.
Input to
ADC
Channel no.
from EPP
A0
A1
A2
From
Sensors
ANALOG
MULTIPLEXER
ADG 508A

3.2.3 ANALOG TO DIGITAL CONVERTER SECTION

The heart of the DAS is the A/D converter. Basically an A/D converter converts an analog voltage to digital output that best represents the input. The ADC used in our circuit works on the principle of successive approximation method AD574 which is a 12 bit ADC.

The choice for a 12 bit ADC is based on the resolution required for the parameters that are being measured i.e. temp. , RH, pressure. For example, the humidity sensor used has a range of 0% to 99% RH with a resolution of 0.1% RH.

99% corresponds to a current of 20 mA

i.e. 0.1% would correspond to a current of (0.1 * 20mA) / 99 = 0.02mA

this should correspond to ± 1 LSB variation of the ADC.

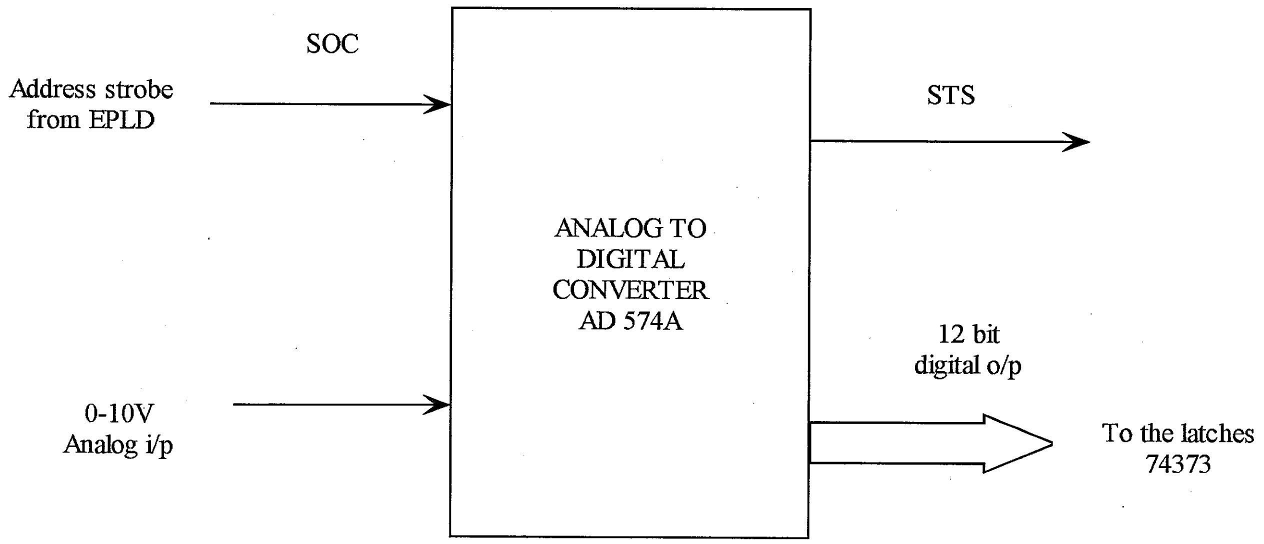
The resolution provided by an 8 bit ADC = 20mA / 2^8 = 0.078 mA and

that provided by a 10 bit ADC is 20mA / 2^12 = 0.019mA which is sufficient for the resolution required. Since commercially 10 bit ADCs are not available, a 12 bit ADC which provides better resolution is chosen.

AD574A has a maximum conversion speed of 35μs.

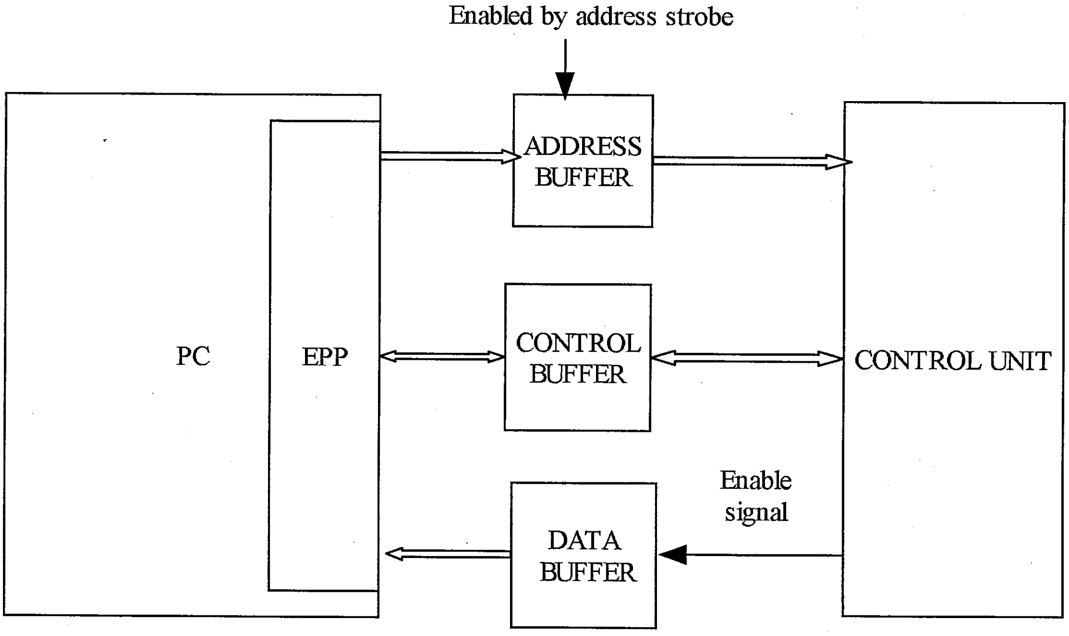
The ADC is configured for a 0 –10 V, unipolar input, 12 bit conversion.

The start of conversion signal to the ADC is given by the control unit. The output signal STS (pin no 28) indicates the status of the converter. STS goes high at the beginning of a conversion and returns back to low state when the conversion cycle is completed (i.e. End of Conversion). When the STS goes low, the 12 bit output is latched on to two 8 bit latches [74LS373].



3.2.4 INTERFACE SECTION TO THE EPP

The base address of the card is initialized to 0X378H. All the control signals to and from the EPP(data strobe, address strobe, write, wait) are through a buffer. When the acquisition command is given by the software, Address strobe is generated by the EPP. This acts as the start-of-conversion for the ADC. The Wait which indicates the end of a read / write cycle is generated by the control unit. The 12 bit digitized outputs are read through two data read cycles since the EPP has only an 8 bit data bus. The read cycles are initiated by the Data strobe. The first cycle reads the 8 LSBs (D0 – D7) and the second cycle reads the 4 MSBs (D8 – D11). The necessary handshaking signals required for the interface are provided by the EPLD.



3.2.5 CONTROL UNIT

The control unit is implemented on an EPLD. The device used is EPM7128ELC 84 –15 which is a higher density member of MAX 7000 family consisting of nearly 2500 usable gates. It is an 84 pin IC with 128 macrocells.

The control unit generates signals that are used for enabling various components on the card.

The two 8:1 multiplexers ADG508A can be configured for either single ended or differential ended which corresponds to 16 channels or 8 channels respectively. The logic to generate the enable signals to these multiplexers is implemented in the control unit. This configuration is dependent on jumper settings and the fourth address bit A3. When A3 is at logic 1 both the multiplexers are enabled. When A3 is at logic 0, depending on the jumper settings either mux1(J is grounded) or mux2 (J to Vcc) is enabled. The signals for enabling the multiplexer1 (MUX1/) and multiplexer2 (MUX2/) are generated by the control unit.

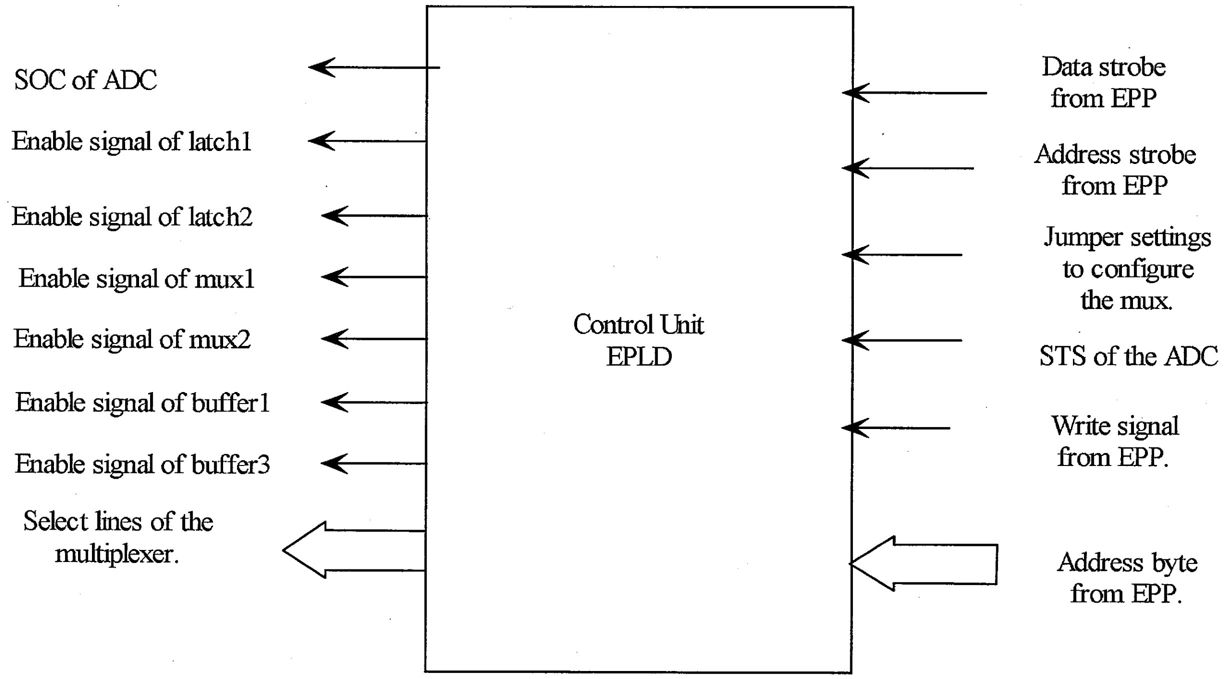
The start of conversion (r/c) for the ADC is initiated by the address strobe and is given by the EPLD. Whenever an address strobe is given by the EPP, it indicates to the analog to digital converter to start a 12-bit conversion.

In implementing a handshake, the data strobe and address strobe can be asserted to start a cycle only when wait is low. It also requires the wait to be high to complete a cycle.

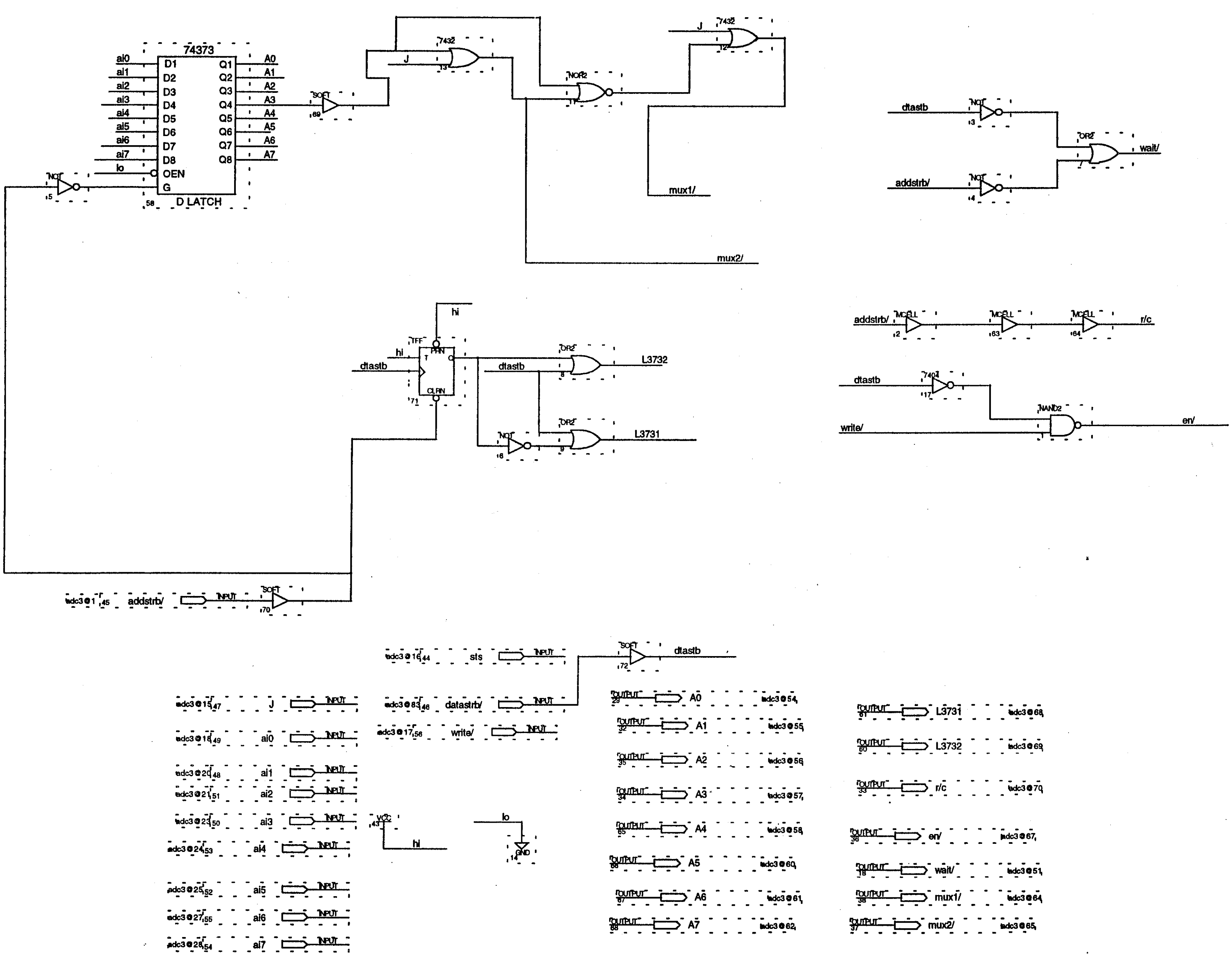
The hardware necessary to generate the wait signal is implemented in the control unit. Wait signal goes high when either Data strobe and Address strobe signals is asserted and the address write and data read cycles are completed.

As already described , two data read cycles are needed to read the 12 bit digitized output from the ADC. During the first data read cycle, the 8 LSBs (D7 – D0) are read and the remaining 4 MSBs (D11 – D8) are read in the next data read cycle. The outputs of the two 8 bit latches are multiplexed and given to the buffer. The availability of these outputs at the latches are dependent on the 1 bit counter triggered by the Data strobe. With appropriate preset and clear facilities, the output of the counter is Ored with the Data strobe to enable the latches. The latched outputs are finally available on the PC.

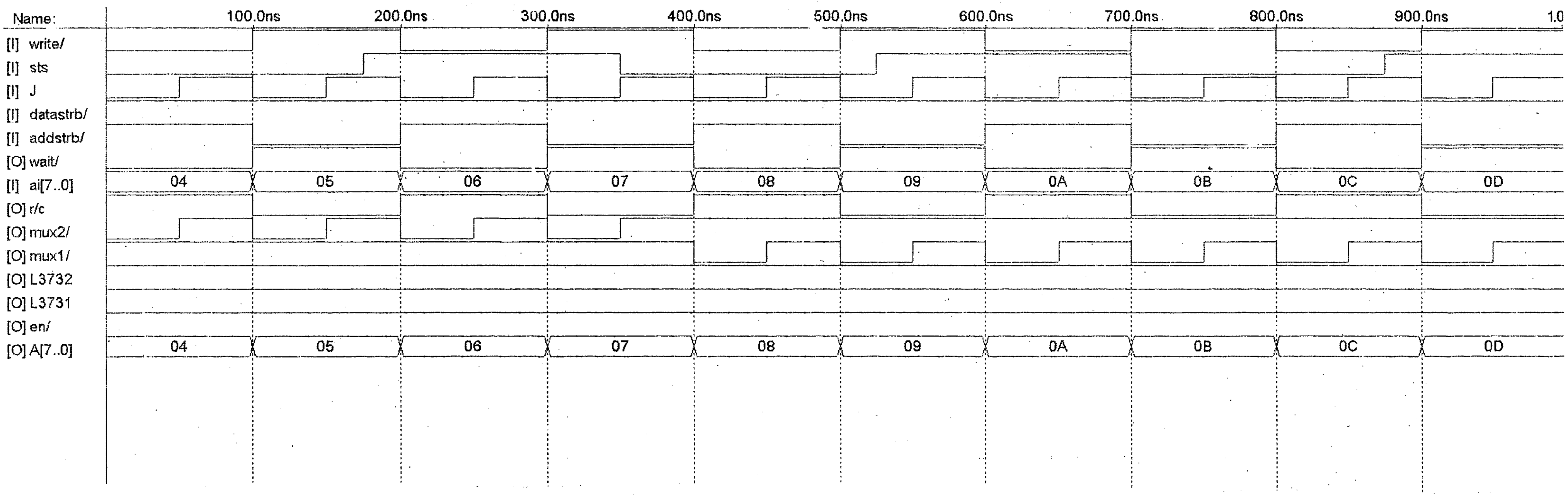
Therefore, these control signals are prominent in the successful working of the card.



The logic circuit implemented in the EPLD and the simulated waveform of the control signals generated are as attached.


74373
D1
D2
D3
D4
D5
D6
D7
D8
OEN
G
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
D LATCH
58
ai0
ai1
ai2
ai3
ai4
ai5
ai6
ai7
lo
A0
A1
A2
A3
A4
A5
A6
A7
NOT
5
SOFT
69
7432
13
J
NOR2
11
7432
12
J
mux1/
mux2/
dtastb
NOT
3
addstrb/
NOT
4
OR2
7
wait/
addstrb/
MCELL
2
MCELL
63
MCELL
64
r/c
dtastb
7404
17
write/
NAND2
en/
hi
TFF
PRN
T
CLRN
71
hi
dtastb
OR2
8
dtastb
L3732
NOT
6
OR2
9
L3731
adc3@1 45
addstrb/
INPUT
SOFT
70
adc3@16 44
sts
INPUT
SOFT
72
dtastb
adc3@83 46
datastrb/
INPUT
adc3@17 56
write/
INPUT
adc3@15 47
J
INPUT
adc3@18 49
ai0
INPUT
adc3@20 48
ai1
INPUT
adc3@21 51
ai2
INPUT
adc3@23 50
ai3
INPUT
adc3@24 53
ai4
INPUT
adc3@25 52
ai5
INPUT
adc3@27 55
ai6
INPUT
adc3@28 54
ai7
INPUT
VCC
43
hi
lo
GND
14
OUTPUT 29
A0
adc3@54
OUTPUT 32
A1
adc3@55
OUTPUT 35
A2
adc3@56
OUTPUT 34
A3
adc3@57
OUTPUT 65
A4
adc3@58
OUTPUT 66
A5
adc3@60
OUTPUT 67
A6
adc3@61
OUTPUT 68
A7
adc3@62
OUTPUT 61
L3731
adc3@68
OUTPUT 60
L3732
adc3@69
OUTPUT 33
r/c
adc3@70
OUTPUT 36
en/
adc3@67
OUTPUT 18
wait/
adc3@51
OUTPUT 38
mux1/
adc3@64
OUTPUT 37
mux2/
adc3@65

MAX+plus II 9.3   File: K:\GURU\PROJ\ADC_PROJ\ADC3.SCF   Date: 08/05/2000 15:59:20   Page: 1



3.3 HARDWARE TESTING

The data acquisition system was tested module by module initially and then after integration it was tested as a whole.

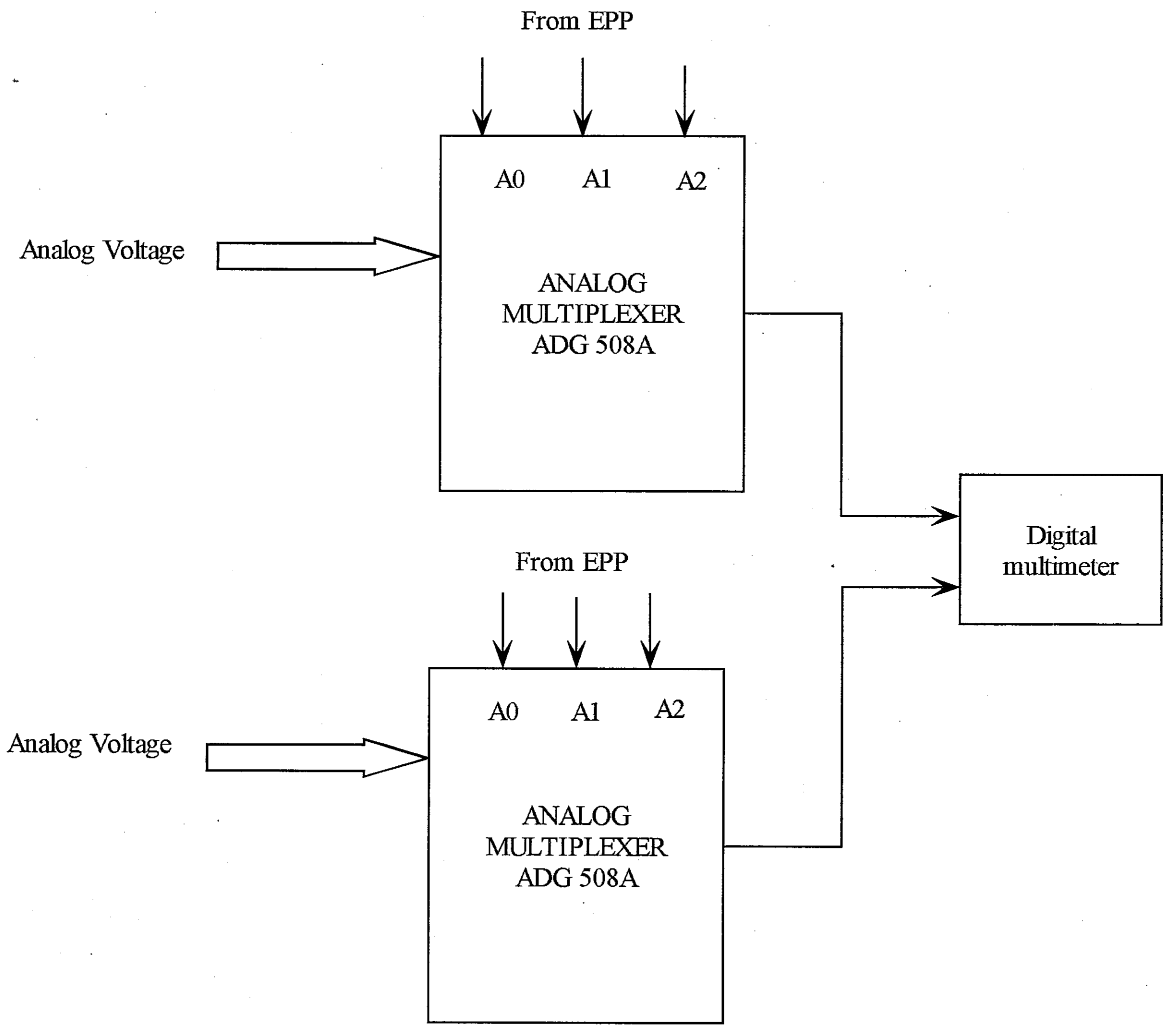
MUX SECTION

The multiplexer section was tested in various steps. Firstly, verification of the channel selection was carried out. The select lines of the MUX were tied to the logic 1/0 and some arbitrary analog input were given to the 8 different channels. The MUX was enabled. It was verified for all possible values of the select lines whether the corresponding channels were selected.

Then the operation of the MUX in the single ended and the differential mode was tested with the required jumper settings and the 4th LSB of the base address of the card(A3). Hence the bit(A3) (input to the EPLD ai3) was tied to 1/0 and in the jumper the pins 1&2/2&3 were shorted. The MUX enable signals was generated by the EPLD as the control unit. It was seen that for the single ended mode, any one of the 2 multiplexers were selected but for the differential ended mode both the multiplexers were selected and the Tables 1 and 2 (Section 3.2.1)were verified.

In the later stages, the select lines of the MUX and the bit A3 were given by the EPP. The select lines of the MUX along with A3 are generated by the EPP and the jumper was set either for single ended or differential ended. Some arbitrary input was given to the 16 channels through the connector. Thus the operation of the MUX in the single ended and the differential mode and the channel selection was verified.

The MUX section was thus thoroughly tested.



ADC SECTION

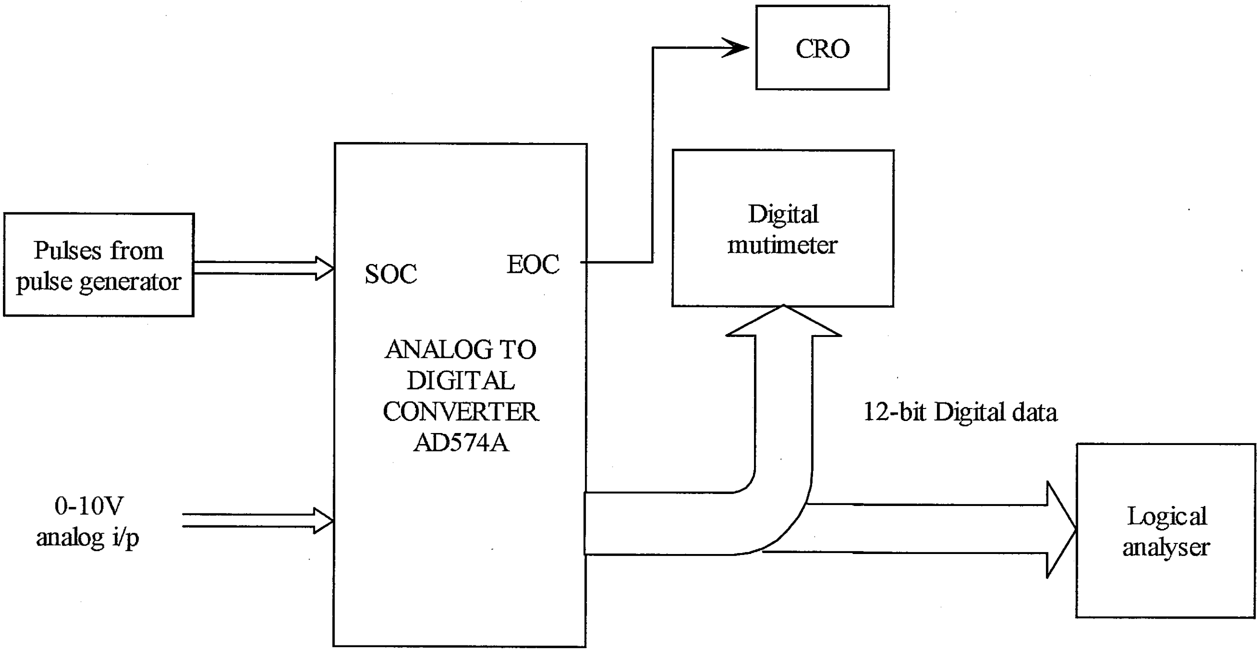
The ADC section was tested step by step. First, both the SOC and the analog input was directly given to AD574A . For the SOC we used a single pulse from the pulse generator and analog input was from a 1.5V battery. The STS line which indicates the status of the ADC was continuously checked. When STS went low, the 12 bit o/p was available at the output of the ADC. These bits were checked using a digital multimeter.

For a 1.5V input the 12 bit digital equivalent must be 0010 0110 0110 i.e. 266H. The output obtained was 263H. The LSB variation of 3 was due to the grounding considerations of the card.

Next , instead of a single pulse, a train of pulses was given as Start Of Conversion(SOC) and it was verified that the digitized output remained nearly the same for a 1.5V battery input. The digital output for various analog voltages was observed on the CRO. The ADC has a maximum conversion time of 35µs. the duration for which the STS is high i.e. the conversion time, was observed to be in the range of (28 – 35)µs.

Next, the inputs to the ADC was from the MUX. For this some known inputs were given to the connector. The select lines and the jumper settings were given through the buffer. First the SOC was a single pulse. The digital output was verified. Next the SOC was a train of pulses. The STS line and the digital outputs were monitored on a CRO. For the train of pulses the digital output nearly remained the same.

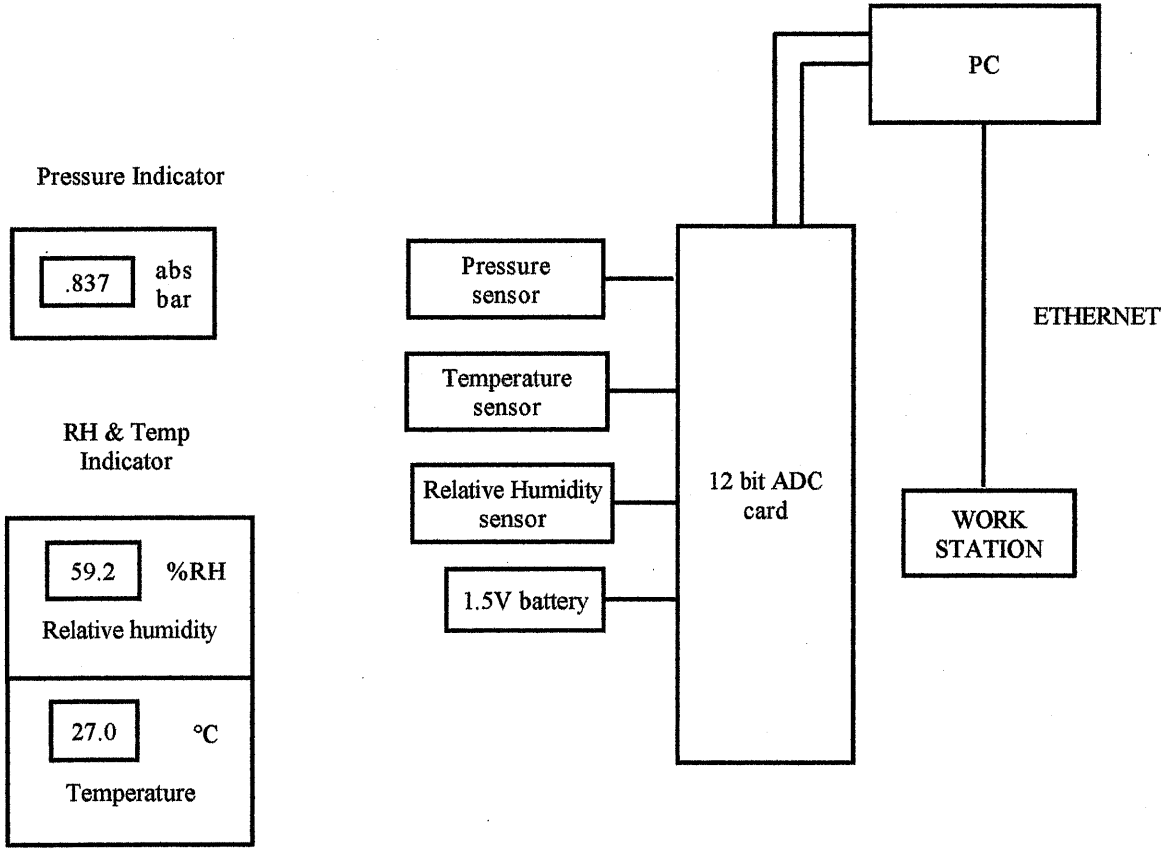
In the next step, the same was tested on a logic analyzer. The logical analyzer was triggered by a train of pulses. Each conversion is initiated by a pulse. The outputs were then verified .



INTERFACE SECTION

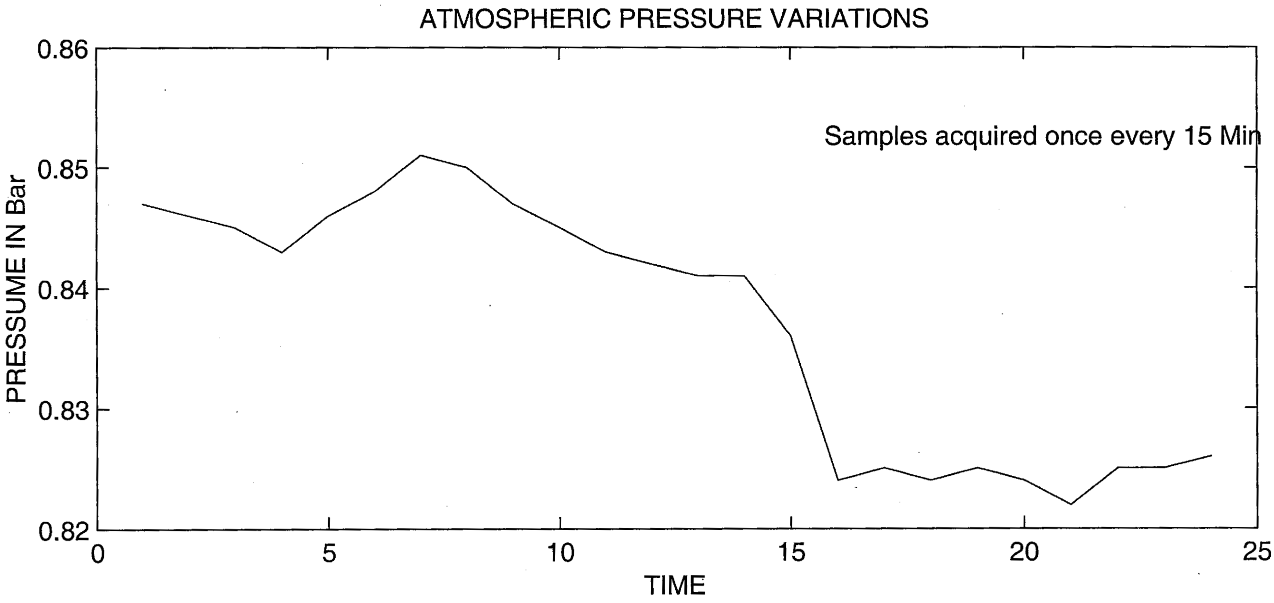
The major part of testing was the operation of the EPP in its bi-directional mode. For this, a 12 bit counter and the necessary hardware signals were implemented in an EPLD. The Data strobe from the EPP triggered the counter. The counter was preset by a logic high and cleared by the Address strobe. For each Address strobe, the counter was incremented and the incremented value was read in the data read cycle.

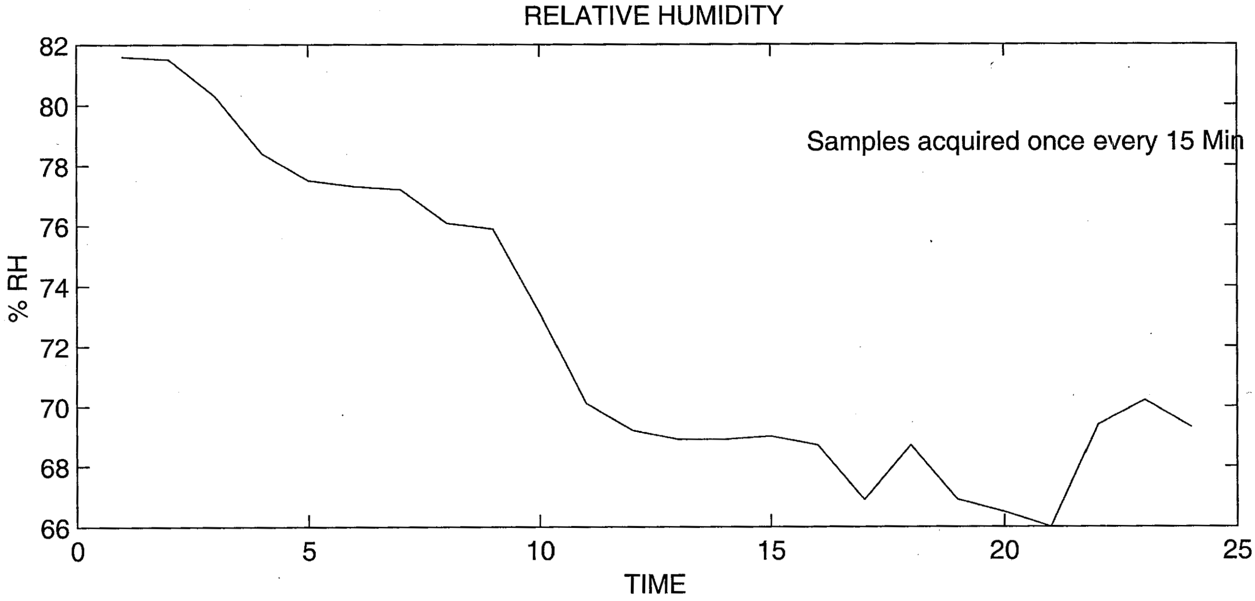
Then, the EPP was interfaced to the ADC card and the values of Pressure, Temperature and Relative Humidity were acquired successfully.

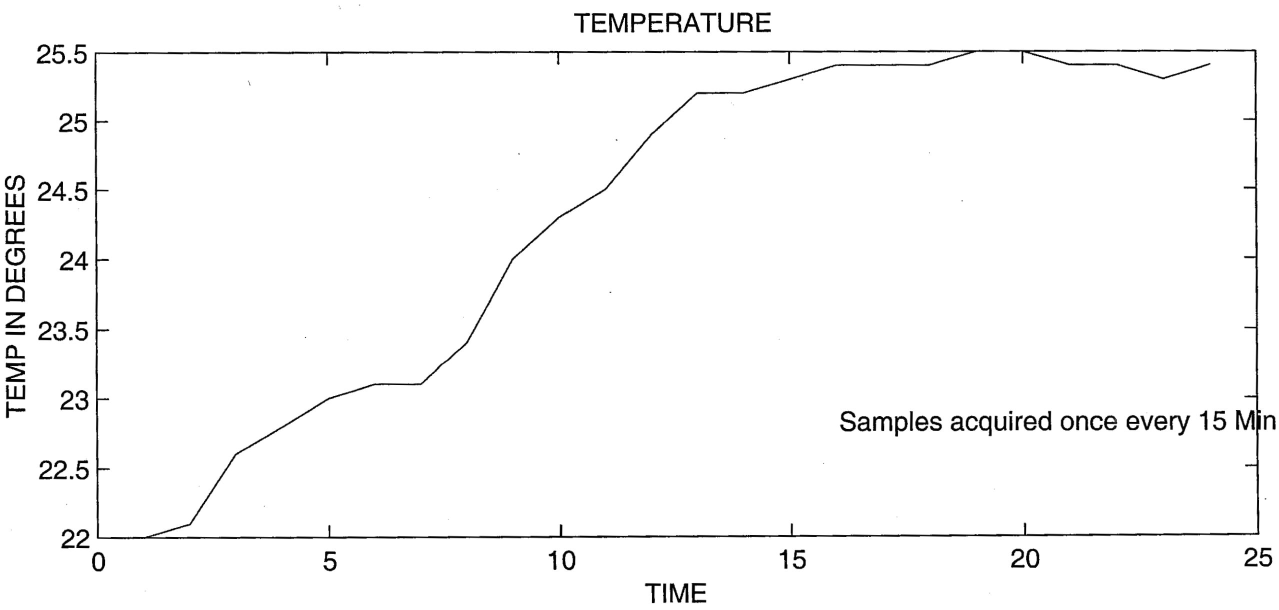

Pressure Indicator
.837
abs bar
RH & Temp Indicator
59.2
%RH
Relative humidity
27.0
°C
Temperature
Pressure sensor
Temperature sensor
Relative Humidity sensor
1.5V battery
12 bit ADC card
PC
ETHERNET
WORK STATION

In the setup shown, the hardware is configured for the differential mode of operation. The channels 1,2,3 represents pressure, temperature and relative humidity respectively. The input to the channel 4 is a 1.5V battery. The battery is used to cross check the stability of the ADC. The output of the sensors are in the range of 4 – 20mA. This is converted to its voltage equivalent by tapping the output across a 100Ω resistor. The output voltage is then converted into the respective parameter units by multiplying with a suitable scaling factor.

A continuous acquisition was carried out and the results are as plotted.


ATMOSPHERIC PRESSURE VARIATIONS
PRESSUME IN Bar
0.86
0.85
0.84
0.83
0.82
0
5
10
15
20
25
TIME
Samples acquired once every 15 Min


RELATIVE HUMIDITY
% RH
82
80
78
76
74
72
70
68
66
0
5
10
15
20
25
TIME
Samples acquired once every 15 Min


TEMPERATURE
TEMP IN DEGREES
25.5
25
24.5
24
23.5
23
22.5
22
0
5
10
15
20
25
TIME
Samples acquired once every 15 Min

SOFTWARE SECTION

4. SOFTWARE ENVIRONMENT AND TOOLS.

4.1 TRANSMISSION CONTROL PROTOCOL AND INTERNET PROTOCOL

4.1.1 Overview of TCP/IP

TCP/IP is one of the software packages that currently dominate UNIX data communications. It is the leading communication software for Unix local area networks. The name TCP/IP refers to an entire set of data communication protocols. The suite gets its name from two of the protocol that belongs to it: Transmission control Protocol.

TCP/IP and the INTERNET:

In 1969, the defence advanced research project agency (DARPA) founded a research and development project to create an experimental packet switching network called ARPANET.

The experimental arpanet was so successful that many of the organizations attached to it began to use it for daily data communications. The basic TCP/IP protocols were developed after ARPANET was operational. The network protocol in ARPANET is IP (Internet protocol) which is connection less and was designed to handle inter connection of vast number of WAN and LAN networks comprising the ARPA Internet. The ARPANET transport protocol is a connection oriented protocol called TCP (Transmission control PROTOCOL) and is used in Berkeley Unix.

The Internet is the worldwide collections of interconnected networks, which grew out of the original arpanet that uses the Internet protocol to link various physical networks into a single logical network. As TCP/IP is required for Internet connection, the large number of diverse new organizations recently added to the Internet have spurred interest in TCP/IP. As more organizations became familiar with TCP/IP, they see that its power can be applied in other network applications. It was common for a site to use TCP/IP for communication over a local Ethernet, which UUCP for communication with remote computer sites.

TCP/IP Features:

The popularity of the TCP/IP protocols on the internet grew rapidly as they met an important need (world wide data communication) at the right time and they had several important features that allowed them to meet this need. These are:

* Open protocol standards, freely available and developed independently from any specific computer hardware or operating system. As it is so widely supported, TCP/IP is ideal for uniting different hardware and software even if there is no communication over the Internet.

* In dependence from specific physical networks hardware. This allows TCP/IP to integrate many different kinds of networks. TCP/IP can run over an Ethernet, token ring, a

dialup line, an X.25 net and virtually any other kind of physical transmission media. A common addressing scheme that allows any TCP/IP device to uniquely address any other device in the entire network, even if the network is as large as the world wide network.

* Standardized high level protocols for consistent, widely available user services.

PROTOCOL STANDARDS:

The open nature of TCP/IP requires publicly available standard documents. The information about TCP/IP protocols is published as Request for comments (RFC). RFC contain the latest versions of specifications of all standard TCP/IP protocol. RFC's contain a wide range of interesting and useful information and are not limited to a formal specification of data communication protocols.

4.1.2 DATA COMMUNICATION MODEL

An architectural model developed by the International standard organization (ISO) is frequently used to describe the structure and function of data communication protocol. This architectural model is called open systems interconnection (OSI) reference model.

Although the OSI model is useful, the TCP/IP protocol doesn't match its structure exactly. The layered mode of TCP/IP is generally viewed as being composed of fewer layers than the OSI model. The 4-layer model of TCP/IP is as shown below.

4. APPLICATION LAYER: Consists of application and Processes that use the network.
3. HOST TO HOST TRANSPORT LAYER: Provides End-to-End data delivery services.
2. INTERNET LAYER: Defines the datagram and handles the routing of data.
1. NETWORK ACCESS LAYER: Consists of routines for accessing Physical networks.

The four layered structure is seen in the way data is handled as it passes down the protocol stack from application layer to the underlying physical network. Each layer adds its control information called the header (to the data received from the previous layer) to ensure proper delivery. This addition of delivery information at each layer is called encapsulation.

When data is received, opposite happens wherein each layer strips off its header before passing the data to the layer above. Each layer has its own independent data structure and its own terminology to describe that structure.

The functionality of each layer is as follows:

NETWORK ACCESS LAYER:

This is the lowest layer of the TCP/IP protocol hierarchy. The functions performed by the protocols in this level include encapsulation of IP datagram into frames transmitted by the network, in the mapping of the IP addresses to physical addresses used by the network. One of the TCP/IP strengths is its addressing scheme that uniquely identifies every host on the Internet. This IP address must be converted into whatever address is appropriate for the physical network over which the data gram is transmitted.

INTERNET LAYER:

The layer above the network access layer in the protocol hierarchy is the Internet layer. This layer provides the ' VIRTUAL NETWORK ' image of Internet (i.e. this layer shields the higher levels from the typical network architecture below it). The Internet protocol is the heart of the TCP/IP and most important protocol in Internet layer. The other useful protocols of this layer are ARP, RARP, etc.

TRANSPORT LAYER:

This layer is just above the Internet layer. This layer is also called HOST-TO-HOST transport layer. The two important protocols in this layer are Transmission CONTROL PROTOCOL and USER DATAGRAM PROTOCOL. TCP is a connection-oriented protocol. It establishes end to end connection between the two communicating hosts. Control information called a handshake is exchanged between the two end points to establish a dialogue, before data is transmitted.

APPLICATION LAYER:

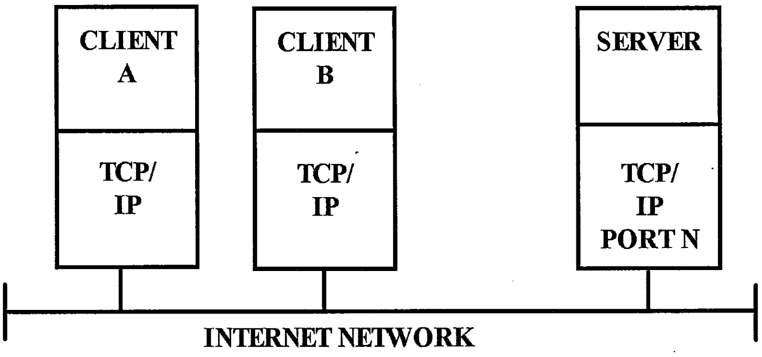
At the top of the TCP/IP, protocol architecture is the application layer. This layer includes all processes that use the transport layer protocols to deliver data. There are many applications protocols, they communicate with applications and other Internet hosts and is user visible interface to the TCP/IP protocol suite. Most of the Application protocol use client server model of interaction.

4.2 CLIENT SERVER MODEL

TCP is a peer to peer, connection oriented protocol. There are on master slave relations. The applications however use a client server model for communication.

A server is an application that offers a service to Internet users: A client is a 'REQUESTER' of a service. An application consists of both the server and the client part that can run on the same or on different systems. Users usually invoke the client part of the application, which builds a request for a particular service and sends it to the server part of the application using the TCP/IP as the transport vehicle.

The server is a program that receives request performs the required service and sends back the results in a reply. A server can usually deal with multiple requests (multiple clients) at the same time.



THE CLIENT SERVER MODEL CONFIGURATION

Some servers wait for request at known port so that the clients know to which IP socket they must direct their request. The client uses an arbitrary port for its communications. Clients that wish to communicate with a server that does not use a well-known port must have another mechanism for learning to which port they must address their requests. This mechanism might employ a registration service such as port map, which uses a well-known port.

From the viewpoint of an application, TCP/IP like most other computer communication protocols merely provides basic mechanisms used to transfer data. In particular, TCP/IP allows the programs to establish communication between two application programs and to pass data back and forth. Thus, we say that TCP/IP provides PEER-TO-PEER communication.

Although TCP/IP specifies details of how data passes between a pair of communicating applications, it does not dictate when or why peer application interact, nor does it specify how programmers should organize such application programs in a distributed environment. In practice, one organizational method dominates the use of TCP/IP to such an extent that all applications use it. This method is known as CLIENT SERVER PARADIGM. The client server paradigm divides communicating applications into two broad categories depending on whether the application waits for communication or initiates it.

4.2.1 CLIENT AND SERVERS:

The Client-server Paradigm uses the direction of initiation to categorize whether a program is a client or server.

In general, an application that initiates peer-to-peer communication is called CLIENT. End users usually invoke client software when they use a network service. Most client software consists of conventional application programs. Each time a client application executes, it contacts a server, sends a request and awaits a response. When the response arrives, the client continues processing. Clients are often easier to build than servers are and usually require no special system privileges to operate.

A typical scenario that takes place for a connection oriented transfer :

First the server opens a communication channel and informs the local host of its willingness to accept client requests on some well-known address and waits for a client request to arrive at the well known address.

The client process opens a communication channel and connects to a specific server. It sends service request messages to the server and receives the responses.

When the server opens a communication channel and waits for a client request it is called passive open. The client, on the other hand, executes what is called an active open since it expects the server to be waiting.

4.3 PORTS AND SOCKETS

Each process that wants to communicate with another process identifies itself to the TCP/IP protocol suit by one or more ports. The port is a 16 bit number, used by the HOST-TO-HOST protocol to identify to which higher level protocol or application program (process) it must deliver incoming messages.

As some higher level programs are themselves protocols, Standardized in the TCP IP protocol suite, such as telnet and ftp, they use the same port number in all TCP/IP implementations. Those ‘ assigned ’ port numbers are called WELL KNOWN PORTS and the standard applications are called well-known services.

A socket is a special type of file handle, which is used by a process to request network services from the operating system.

A socket address is a triple consisting of:

[Protocol, Local Address, Local process]

The socket interface is one of the API's to the communication protocols.

socket system call:

To do network I/O, the first thing a process must do is call the socket system call,specifying the type of communication protocol desired.

int socket(int *family*, int *type*, int *protocol*);

The family is one of

AF_UNIX	Unix internal protocols
AF_INET	Internet protocol
AF_NS	Xerox NS protocol

AF prefix stands for 'address family'.

The socket type is one of the following:

SOCK_STREAM	Stream socket
SOCK_DGRAM	Datagram sockets
SOCK_SEQPACKET	Sequenced packet socket

The SOCK_STREAM and SOCK_DATAGRAM can be used with all family types given earlier while the SOCK_SEQPACKET can be used only with AF_NS.

The *protocol* argument to the socket system call is typically set to zero for most user applications.

The socket system call returns a small integer value, similar to file descriptor. This is called a socket descriptor or *sockfd.* To obtain this socket descriptor all we have to do is specify the address family and the socket type.

bind system call:

The bind( ) system call assigns a name to an unnamed socket.

int bind(int *sockfd*, struct sock addr * *my addr*, int *addrlen*)

The first argument is the socket descriptor. The second argument is a pointer to protocol specific address and the third argument is the size of the address structure. By using the bind ( ) system call the servers register their well-known address with the system. It tells the system "this is my address and any messages received for these address are to be given to me". A client can also register a specific address for itself.

Connect system call:

A client process connects a socket descriptor following the socket system call to establish a connection with the server.

int connect(int *sockfd*, struct sockaddr **serv addr*, int *addrlen*)

The sockfd is a socket descriptor that was returned by a socket system call. The second and third arguments are pointer to a socket address, and its size.

listen system call:

This system call is used by a connection oriented server to indicate that it is willing to receive connections.

int listen (int *sockfd*, int *backlog*);

It is usually executed after both the socket and bind system calls and immediately before the accept system call. The *backlog* argument specifies how many connection requests can be queued by the system while it waits for the server to execute the accept system call.

accept system call:

After a connection oriented server executes the listen system call, an actual connection from the client process is waited for by having the server execute the accept system call.

int accept(int *sockfd*, struct sockaddr **peer*, int **addrlen*);

The peer and addrlen arguments are used to return the address of the connected peer process(the client).

read system call:

Data is read from an open socket using

int read(int *sockfd* ,char **buffer* ,unsigned int *nbytes*);

If the read is successful the number of bytes read is returned- this can be less than the n bytes that was requested.

write system call:

Data is written to an open socket using

int write(int *sockfd*,char **buffer*, unsigned int *nbytes*);

4.4 *PC/TCP* System Calls

The PC/TCP system calls provide the lowest –level application programming interface(API) to the PC/TCP kernel. The PC/TCP kernel implements the TCP/IP protocols for personal computers.

For the remote mode of data acquisition, the server program is a DOS application on a PC and hence PC/TCP system calls are used for establishing a connection with the client . In native mode programming communication between hosts is based on a network descriptor. A network descriptor is similar to a socket in UNIX operating system. A network descriptor is a integer value handle that identifies a specific connection or the endpoints of a communication and protocol type to use for the communication. Network descriptor is also used by system calls to affect or report on the attributes or status of the connection or communication.

The protocol layer for PC/TCP system calls share a common programming interface based on the manipulation of a network descriptor. Essentially we use a network descriptor (nd) to represent either a client process or a server process. A network descriptor is initialized so that it activates a connection or associates end point information with a particular protocol type.

Before initiating the network descriptor, we have to specifically request a global network descriptor with net_getglobdesc( ) which allocates a descriptor.

```
int net_getglobdesc( ) ;
```

The net_getglobdesc( ) system call allocates a global network descriptor which can be accessed from within any process.

The n/w connection or the association of endpoint information is not established until we initialize the network descriptor to a connection using net_listen( ) for a TCP server process. net_listen( ) initializes a network descriptor with protocol type and end point information.

int net_listen(int *nd*, int *type*, struct addr **addr*)

The net_listen( ) system call passively opens (listens) or initiates an association of type(DGRAM or STREAM) on network descriptor and returns network descriptor. Upon successful completion, the remote host internet address and the socket number or port to use on remote host fields is updated.

Thereafter we can manipulate the network descriptor to carry out network I/O routines, to reset option, to show status or to perform various other tasks.

net_read system call:

Reads into a buffer data received on a network descriptor.

int net_read (int *nd*, char **buf*, unsigned *len*, struct addr **from*, unsigned *flags*);

where

buf is a pointer to the buffer containing incoming data,

len is the number of bytes to read into *buf*,

from is a pointer to an *addr* structure containing address information,

flags is the Boolean options for the receive operation.

net_write system call:

Transmits data over a network descriptor.

int net_write(int *nd*, char **buf*, unsigned *len*, unsigned *flags*).

When the network descriptor has no further function to carry out, we shall release it using net_release( ) to free all PC/TCP kernel resources.

4.5 FEATURES PROVIDED IN THE SOFTWARE

- Dos based GUI using C language.
- Software integration possible.
- Client server model.

The software can operate in two modes:

1.Local mode.

2.Remote mode.

4.5.1 DESCRIPTION

LOCAL MODE :

In the local mode the user can enter the command from the keyboard and perform the required operation. The commands used for different operations are:

* config 1 -- configure for single ended mode.

* config 2 -- configure for differential mode.

* acq N fname.log -- acquires N samples for each channel and saves it onto fname.log file.

* disp – displays the acquired values.

When the command for acquisition is issued, the EPP is initialized for its bi-directional mode of operation. The channel number whose values have to be acquired is then written onto the EPP. The start of conversion to the ADC is initiated by the Address strobe of the EPP. It requires two data read cycles to read the 12 bit output. The data read cycle is initiated by the Data strobe.

The acquired values are in the range of (0.6 – 3) volts. These values are then converted to the respective units of Pressure, Temperature and Relative Humidity by multiplying with a suitable factor. The software then displays these values.

REMOTE MODE:

In the remote mode, the PC can be hooked on to a remote host. Since the server program is a DOS application we need to use PC/TCP system calls to establish a connection with the remote client. After creating a network descriptor, the server listens to the remote client. The client (workstation) then establishes a connection with the server using internet domain socket with the system call connect( ) . After the client has communicated with the PC, net_listen( ) updates the value of the remote host address and also the socket no. of the remote host to which the data has to be read from or written to. Thus, a connection is established between the server and the client.

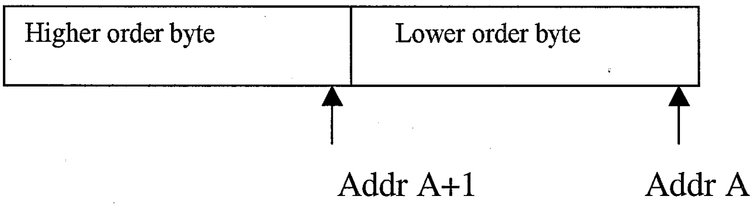
The workstation sends a command to the PC for acquiring N samples for each channel .

The command for single ended mode is SIN ACQ N.

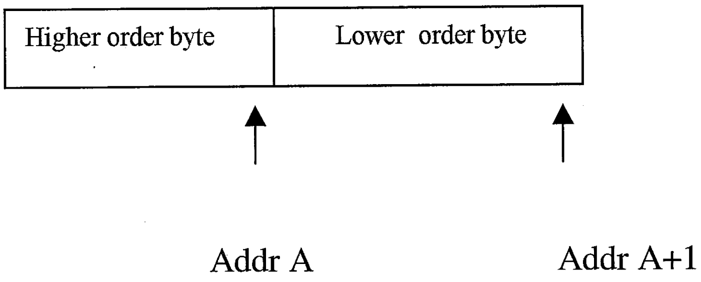
The command for differential ended mode is DIFF ACQ N.

The server then reads N samples for each channel successively, calculates the average of these N samples for each channel and is sent to the workstation.

The server program is on an Intel machine which store floating point numbers in an little endian format. Here the lower byte is stored first as shown below.



The acquired values are sent to the workstation that stores number in big endian format. In big endian format the higher order byte is stored first and the lower byte as shown



And hence when the transmitted bytes are received, the order of the bytes should be swapped so that they are converted into big endian format. This ensures that the values stored at the workstation is the same floating point values sent from the server.

4.6 DEVELOPMENT CYCLE

A program for acquiring the data from the 12 bit ADC card which is interfaced to the Enhanced Parallel Port was developed. This program initializes the port for bi-directional mode and initiates the address and data read cycles to read the values from the card.

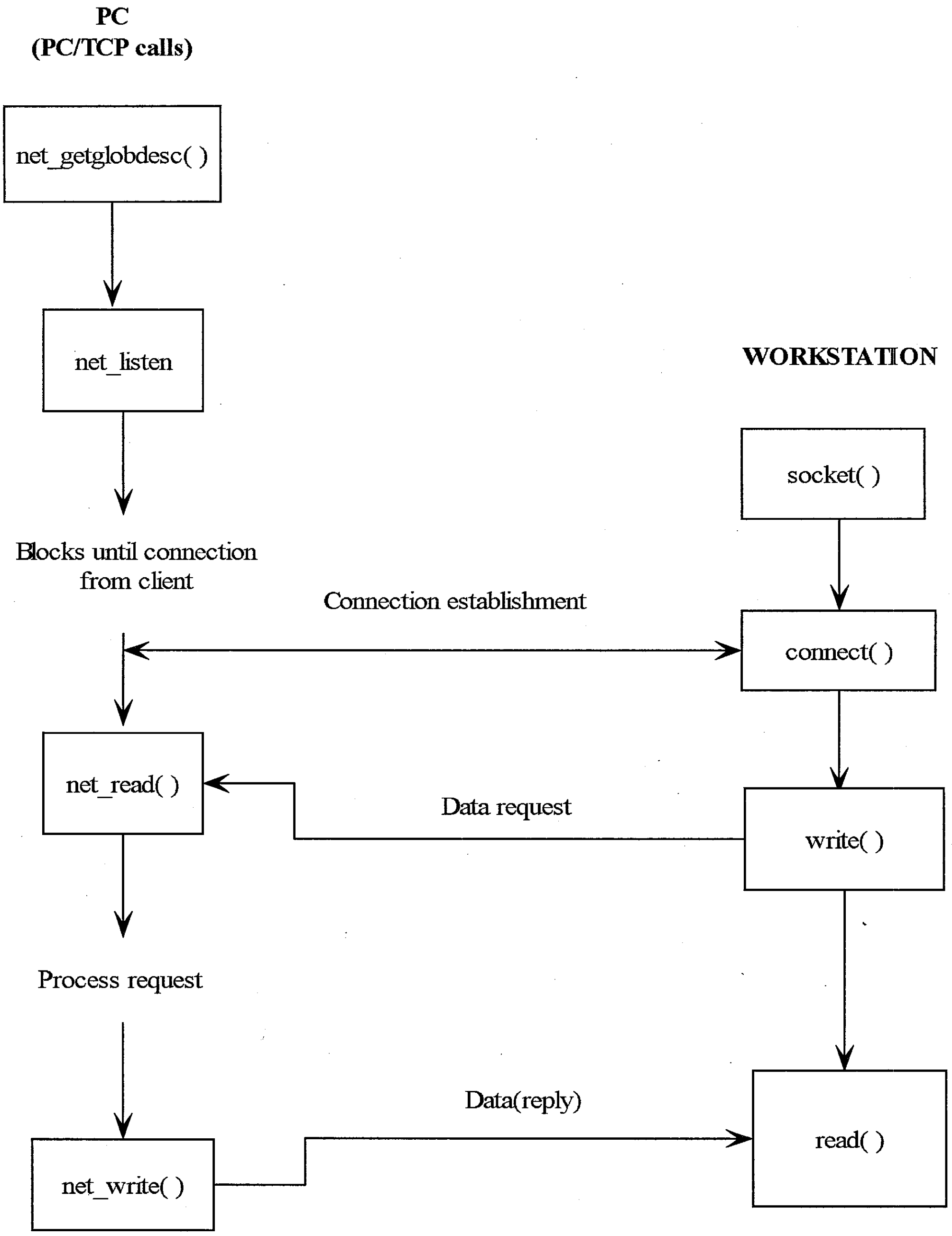
* A user interface for the local mode of operation was developed. Here commands for configuration of the card for single or differential ended modes, to acquire N number of samples for each channel and display the acquired values was developed.

* A client-server model had to be implemented for remote measurement, to communicate with the PC to which the data acquisition card is interfaced. Initially, both client and server programs were implemented on different workstations. Connection was established between them using Internet Domain sockets so that they could communicate with each other.

* Since the acquisition software was a DOS application, the server program had to be developed using PC/TCP system calls or native mode calls. The server uses PC/TCP system calls to establish a connection with the client, while the client communicates using Internet Domain sockets.

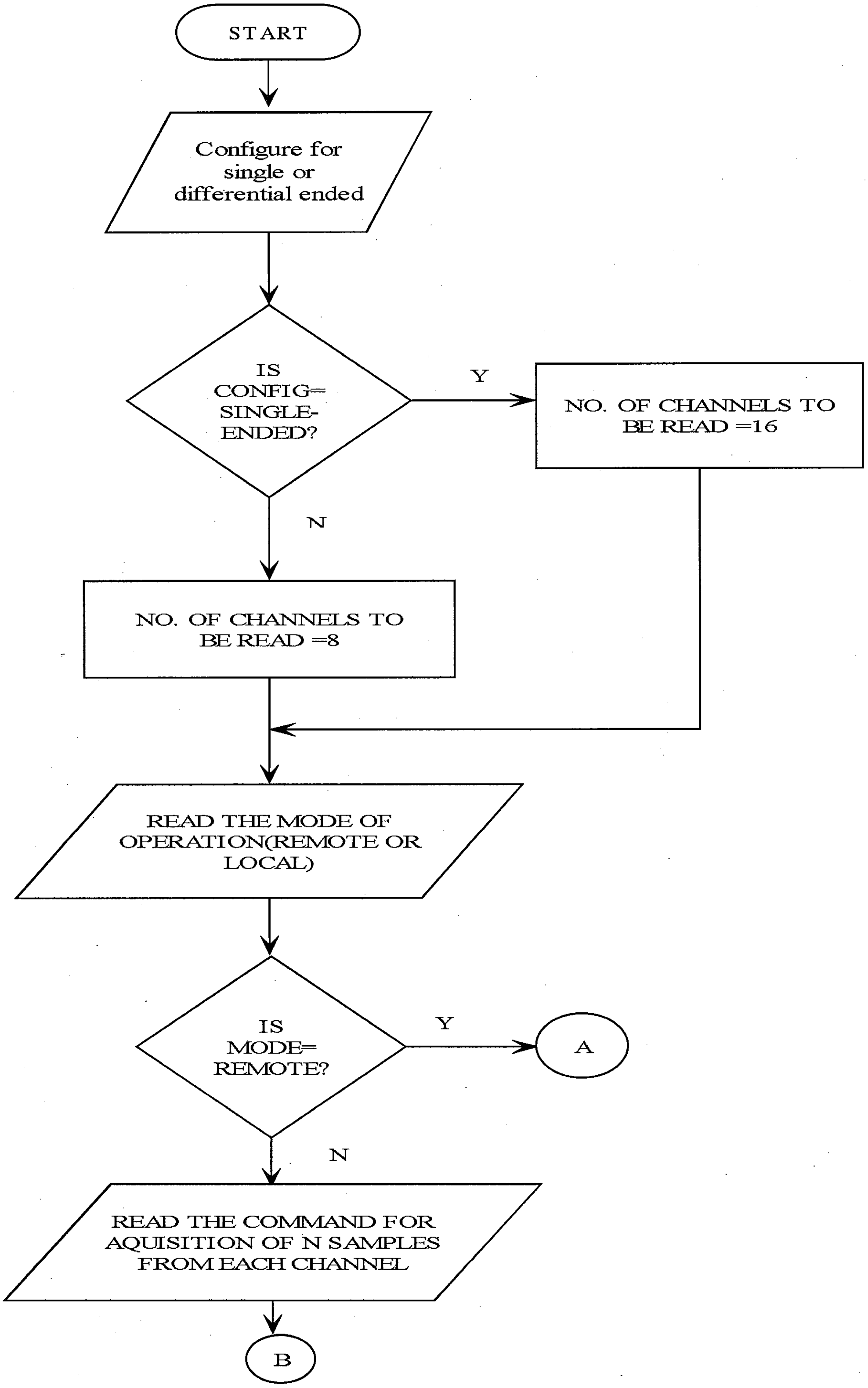
Thus, the program which could operate in both local and remote mode along with the necessary user interface was developed and tested successfully.

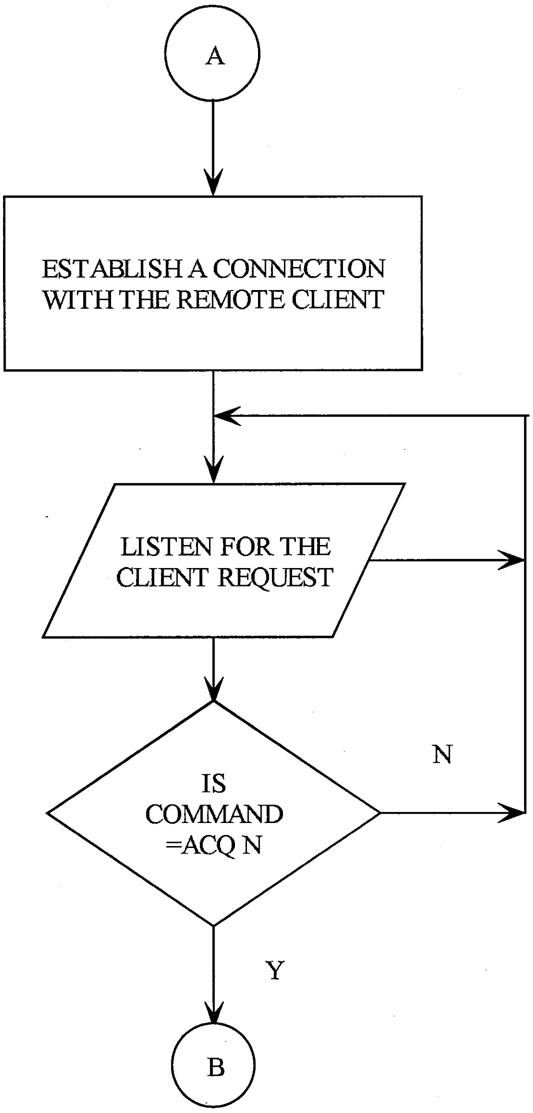
Communication of the Workstation with the PC

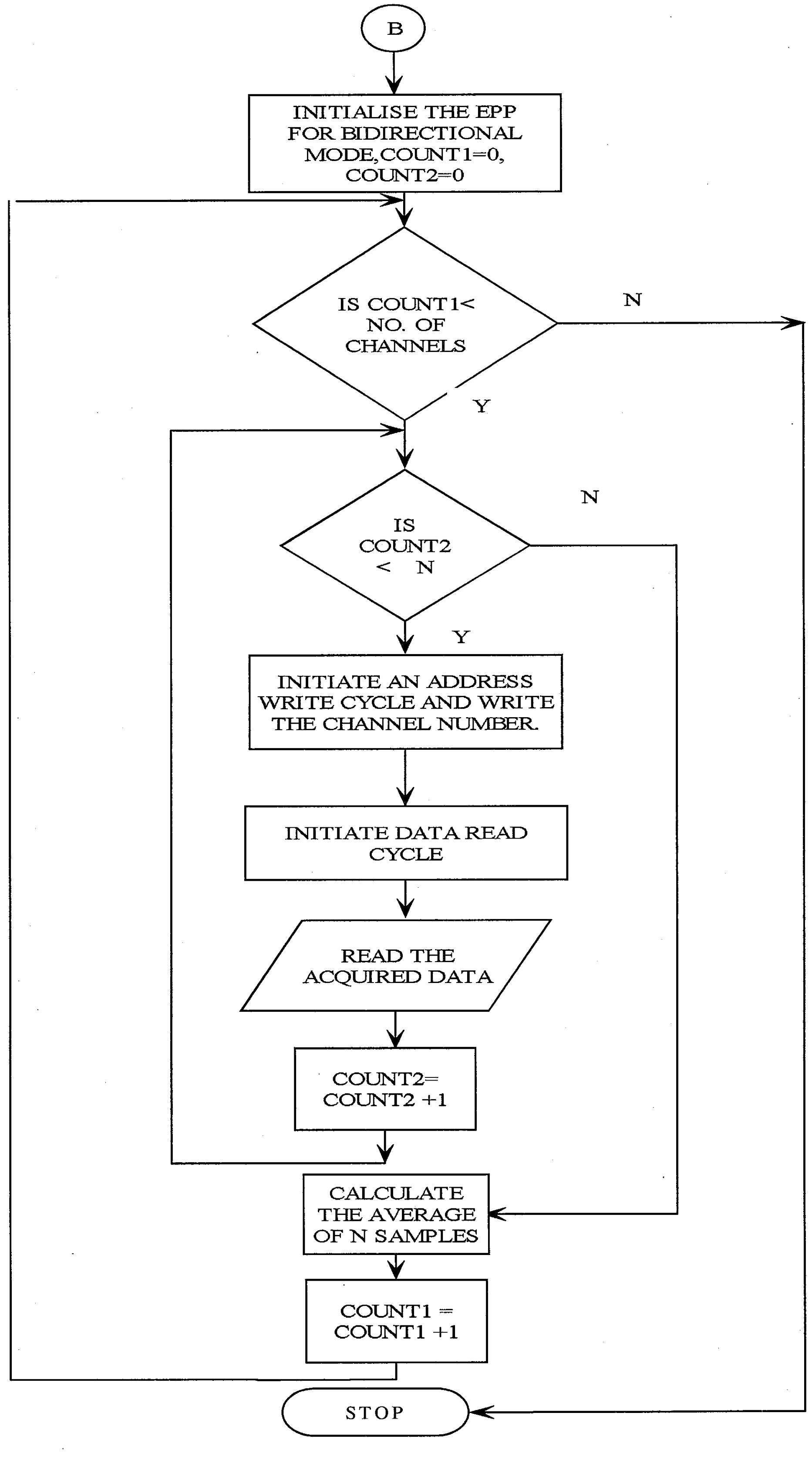


FLOW CHART

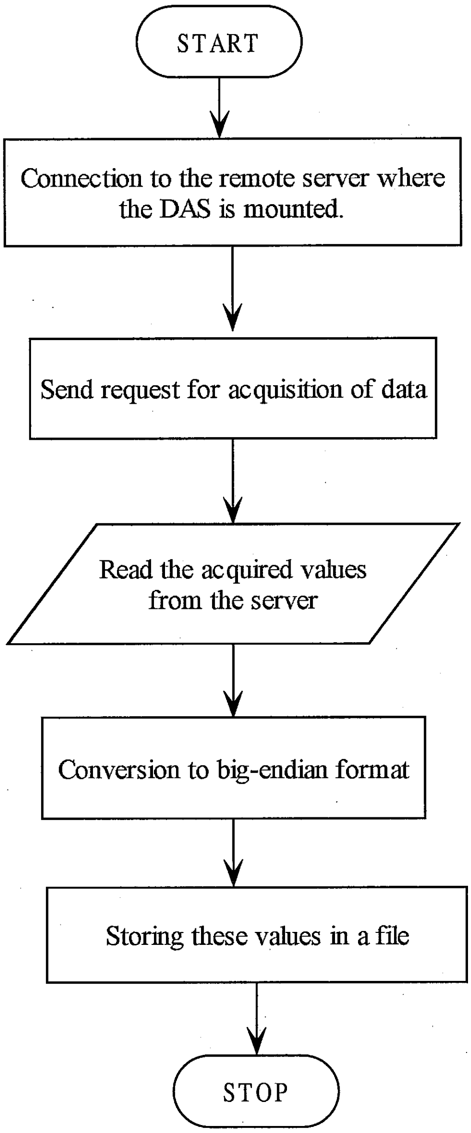
Server program


START
Configure for single or differential ended
IS CONFIG= SINGLE-ENDED?
Y
NO. OF CHANNELS TO BE READ =16
N
NO. OF CHANNELS TO BE READ =8
READ THE MODE OF OPERATION(REMOTE OR LOCAL)
IS MODE= REMOTE?
Y
A
N
READ THE COMMAND FOR AQUISITION OF N SAMPLES FROM EACH CHANNEL
B


A
ESTABLISH A CONNECTION WITH THE REMOTE CLIENT
LISTEN FOR THE CLIENT REQUEST
IS COMMAND =ACQ N
N
Y
B

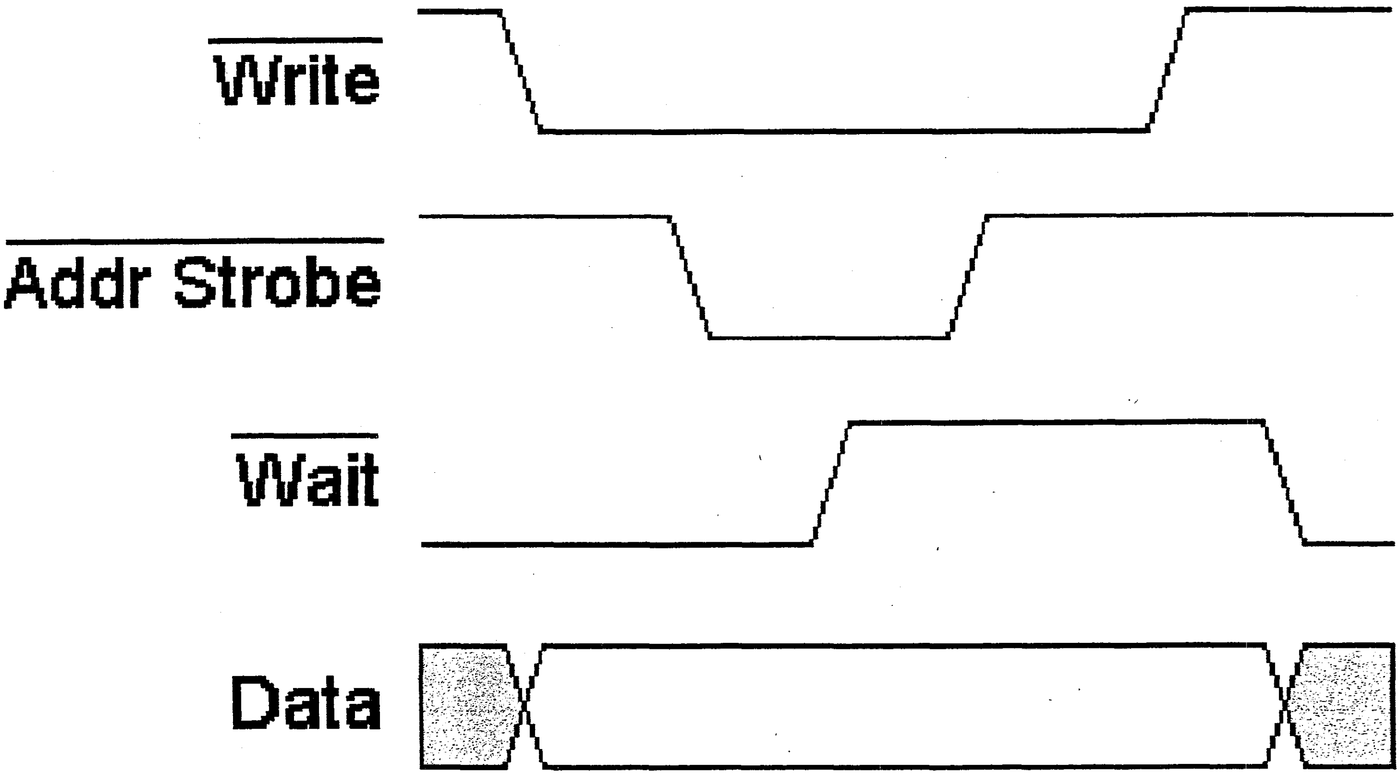

B
INITIALISE THE EPP FOR BIDIRECTIONAL MODE,COUNT1=0, COUNT2=0
IS COUNT1< NO. OF CHANNELS
N
Y
IS COUNT2 < N
N
Y
INITIATE AN ADDRESS WRITE CYCLE AND WRITE THE CHANNEL NUMBER.
INITIATE DATA READ CYCLE
READ THE ACQUIRED DATA
COUNT2= COUNT2 +1
CALCULATE THE AVERAGE OF N SAMPLES
COUNT1 = COUNT1 +1
STOP

CLIENT PROGRAM

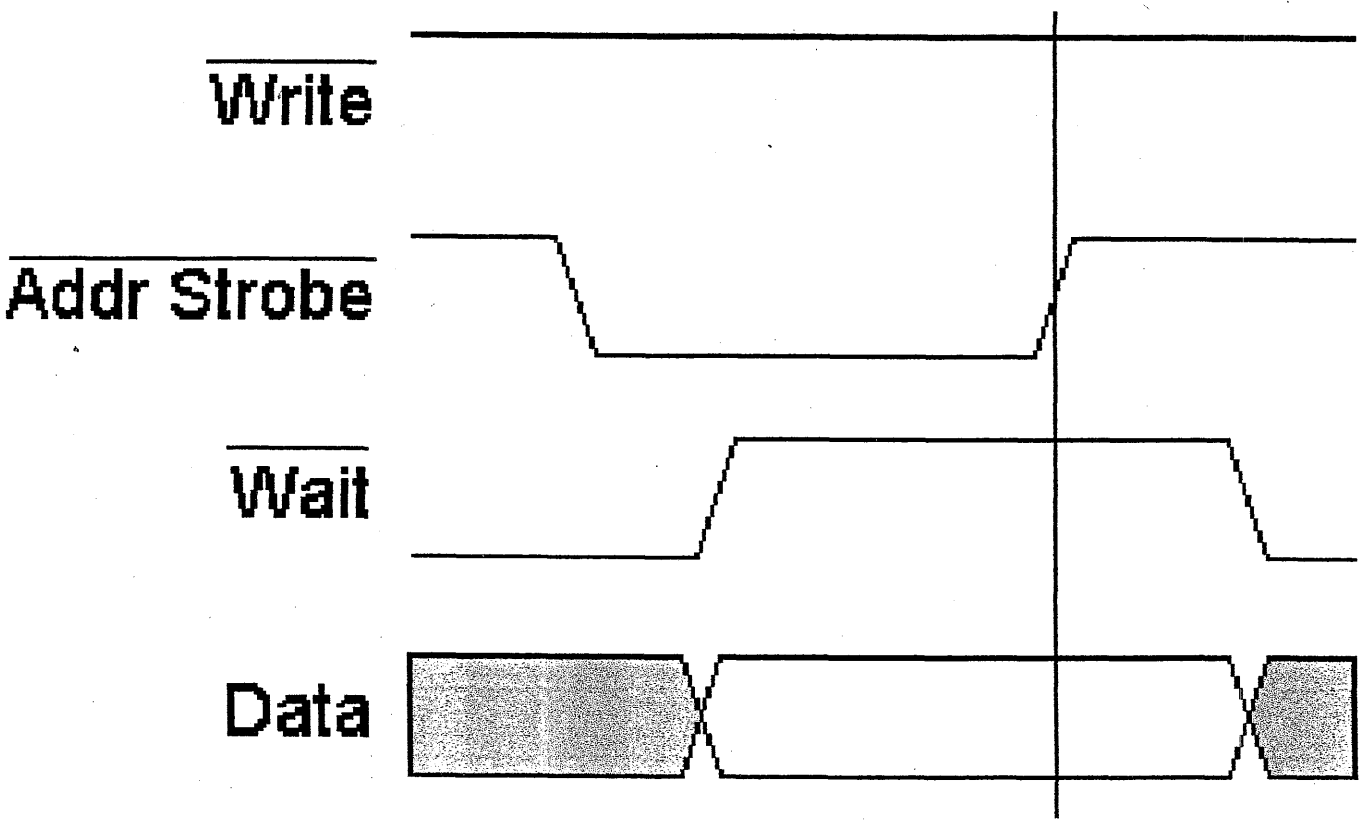

START
Connection to the remote server where the DAS is mounted.
Send request for acquisition of data
Read the acquired values from the server
Conversion to big-endian format
Storing these values in a file
STOP

APPENDIX A

EPP Address Write

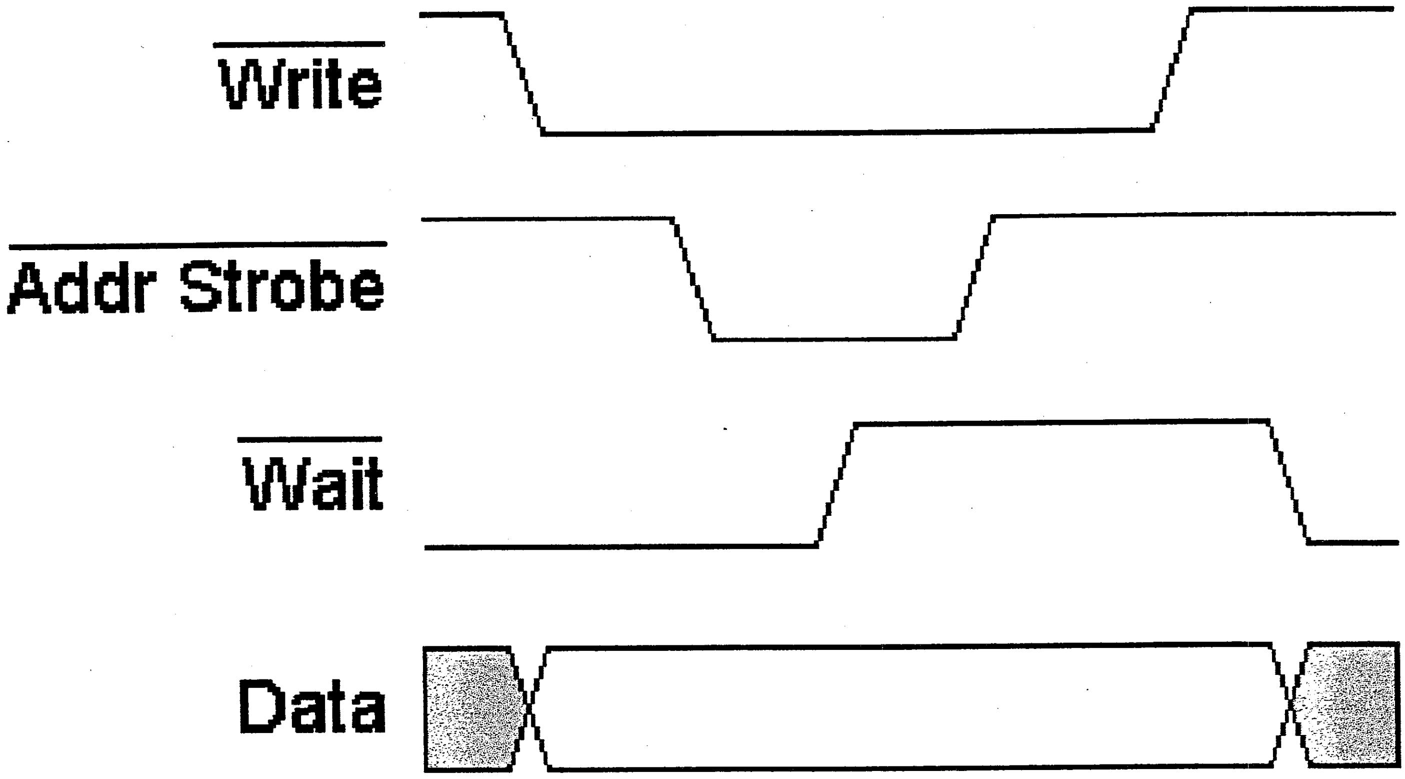


EPP Address Read

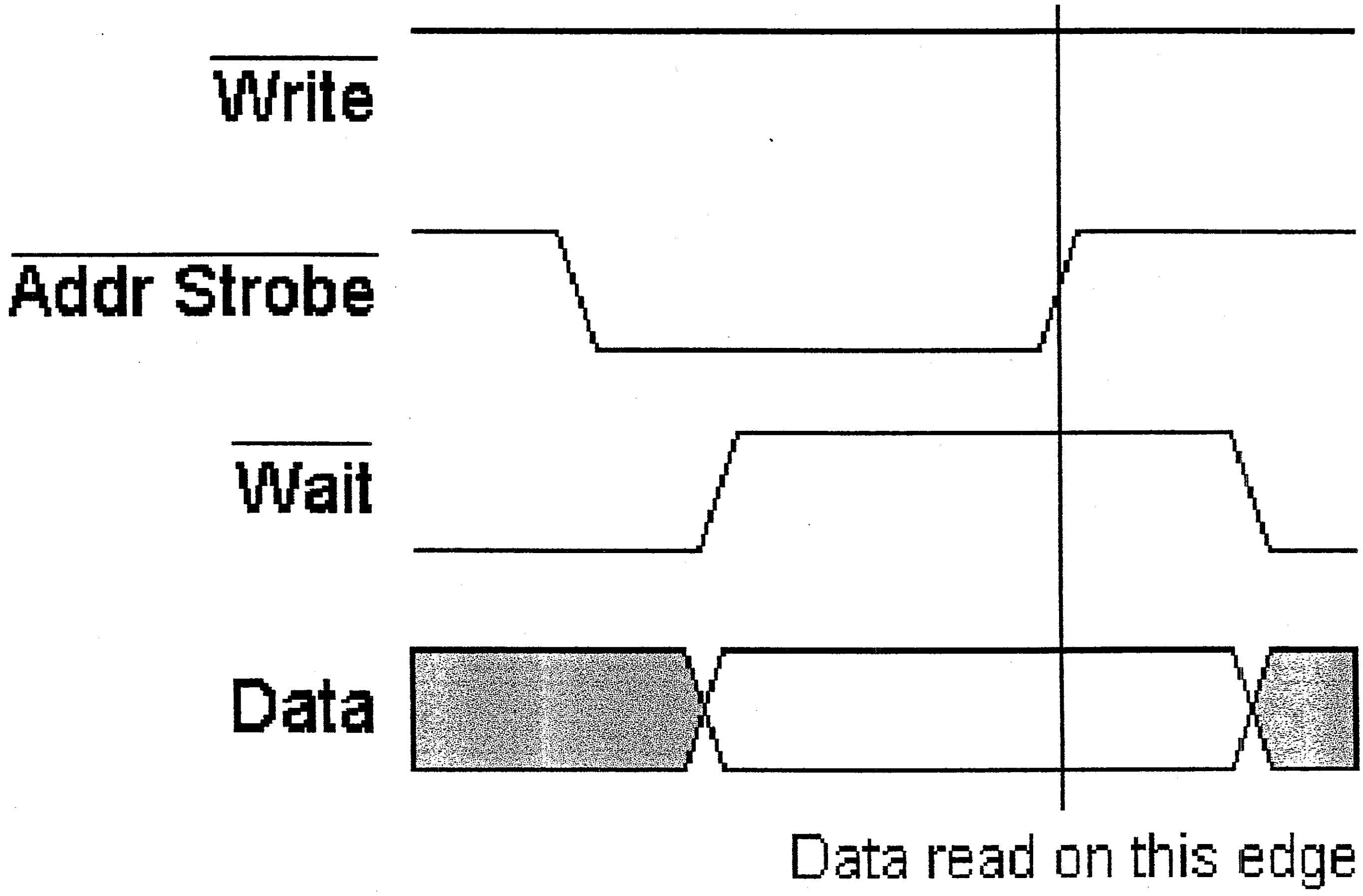


Data read on this edge

EPP Address Write



EPP Address Read



APPENDIX B

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<dos.h>
#include<time.h>
#include "types.h"
#include "pctcp.h"
#include "options.h"
#include "error.h"
#include "interfac.h"

#define PORTNO 8005  /* PORT NUMBER OF THE REMOTE CLIENT THROUGH
                        THE COMMUNICATION TAKES PLACE*/
         /*ADDRESS CONFIGURATION OF ENHANCED PARALLEL PORT*/
#define BASE_ADD 0x378
#define ADD_PORT BASE_ADD+3       /*address of the address port of EPP*/
#define DATA_PORT BASE_ADD+4      /*address of the data port of EPP*/
#define CONTROL_PORT BASE_ADD+2   /*address of the control port*/
#define LO 0x00
#define HI 0x80
int k,n,a,m,c,i,j,d,b,b2,temp,ch_no,h,count1=0,i1,j1,z=0,t,count=0;
char *p[4];
char input[30],fname[10];
float sum,acquire1[8],sam[200];
FILE *fp1;
char ch[7],ch1[200];
int net,netd,readflag,wrflag;
struct addr address;
unsigned int lo_byte,hi_byte,fi_byte,final_byte;
void border();
void acq1(int ch_no);
void paint();
int main(void)
{
   paint();
   gotoxy(5,6);
   printf("Enter the mode of operation:\n");
   gotoxy(34,8);
   printf("LOCAL\n");
   gotoxy(34,10);
   printf("REMOTE\n");
   window(2,21,78,24);
   textbackground(LIGHTGRAY);
   textcolor(BLACK);
   clrscr();
   gotoxy(2,15);
   printf("$");
   gotoxy(3,15);
   p[0] = gets(input);
   if((strcmpi(p[0],"local")) == 0)
   menu();
   else
   {
   if((strcmpi(p[0],"remote")) == 0)
   {
    printf("remote mode");
    clrscr();
    paint();
    gotoxy(4,15);

    /* ADDRESS OF THE REMOTE HOST(WORKSTATION)     */
    address.fhost = 0L;
    address.fsocket = 0;
    address.lsocket= PORTNO;
    address.protocol = STREAM;
    if((netd = net_getglobdesc())<0)
    printf("error in descriptor");

    /*  WAITING FOR THE REQUEST TO ACQUIRE PARAMETERS FROM THE CLIENT */

    printf("waiting for request from client........");
    net = net_listen(netd,STREAM,&address);
    if(net <0)
     printf("error in connecting");
    else
     {

      paint();

      readflag = net_read(net,&ch,7,&address,0);/* After conecting to the client
                                                  accepts the command from it*/

      if(readflag>0)
      {

      if((strcmp(ch,"ACQ 100")) == 0)/* Check if the command sent is for
                                     acquiring the values from different channel
s*/

      {
        for(b=0;b<8;b++)
        {
          acq1(b);
        }
        memcpy(ch1,acquire1,sizeof(acquire1));

        /* Sends the values acquired from data acquisition card to the client*/

        wrflag = net_write(net,ch1,sizeof(ch1),0);
        if(wrflag<0)
        printf("error in writing....");
        gotoxy(4,15);
        printf("acquired values sent successfully\n");
        gotoxy(4,18);
        printf("press any key to close screen");
        getch();
      }
    }
     net_release(netd);
     exfn();
     }
   }
   else
   {
      printf("Invalid Mode\n");
      getch();
      exfn();
   }
  }
 }

 chprn(int c)
 {
     window(1,4,80,21);
     textbackground(BLUE);
     textcolor(WHITE);
     clrscr();
     chnl();
    if((c==1)¦¦(c==2))
    {
    for(i=0,j=4;i<8;i++)
    {
      gotoxy(13,j);
      printf("CHANNEL %d =%6.4f",i,acquire1[i]);
      j=j+2;
    }
   }
   if(c==1)
   {
     for(i=8,j=4;i<16;i++)
     {
       gotoxy(50,j);
       printf("CHANNEL %d = %6.4f",i,acquire1[i]);
       j=j+2;
     }
   }
   getch();
   return;
}

/* This function accepts the commands which the user keys in for local mode
   of operation*/
 menu()
 {
   count1++;
   paint();
   window(2,21,78,24);
   textbackground(LIGHTGRAY);
   textcolor(BLACK);
   clrscr();
   gotoxy(3,1);
   printf("$");
   p[0] = strtok(gets(input)," ");

   if((strcmpi(p[0],"config"))==0)
    {
     p[1]= strtok(NULL," ");
      if(p[1]==NULL)
      {
       cprintf("too less parameter,should specify 1 or 2");
       getch();
       menu();
      }
     n=atoi(p[1]);
     if(n==1)
     c=16;
     if(n==2)
     c=8;
     menu();
    }
   if((strcmpi(p[0],"acq"))==0)
   {
     if((count1==1))
     {
      cprintf("first config for single or diff ended before acquiring");
      getch();
      menu();
     }
     p[1] =strtok(NULL," ");
     if(p[1]==NULL)
     {
      cprintf("too less parameter,should specify the no of samples to be acquire
d&also file name");
      getch();
      menu();
     }
    p[2]=strtok(NULL," ");
    if(p[2]==NULL)
    {
     cprintf("should specify file name to which acq data is to be written");
     getch();
     menu();
    }
      strcpy(fname,p[2]);
      strcat(fname,".log");
      window(2,20,79,24);
      textbackground(LIGHTGRAY);
      textcolor(BLACK);
      clrscr();
      gotoxy(2,4);
      printf("acquiring and saving data........");
      if((fp1=fopen(fname,"w"))==NULL)
      {
       gotoxy(2,3);
       printf("                                                          ");
       gotoxy(2,3);
       printf("Error in opening file %s",p[2]);
       exfn();
      }
      else
       {
         b2 = atoi(p[1]);
         fprintf(fp1,"count\tacquired values\n");

        for(b=0;b<c;b++)
          {
            acq1(c);
            fprintf(fp1,"\n %d ",++count);

             for(d=0;d<c;d++)
             {
               fprintf(fp1,"%4.2f ",acquire1[d]);
             }
           }
          }
      fclose(fp1);
      clrscr();
  }

  else
  {
   if((strcmpi(p[0],"disp"))==0)
     {
       p[1]=strtok(NULL," ");
        if(p[1]!=NULL)
        {
        cprintf("only one parameter sufficient for displaying the values");
        getch();
        exfn();
        }
        chprn(c);
        clrscr();
      }
    else
   {
   if((strcmpi(p[0],"exit"))==0)
          {
           exfn();
           clrscr();
   }       }
}
  window(1,1,80,25);
  textbackground(BLUE);
```

```
   clrscr();
   menu();
   return;

}
 exfn()
 {
       textbackground(BLACK);
       textcolor(LIGHTGRAY);
       window(1,1,80,25);
       clrscr();
       exit(0);
 }

  /* THIS FUNCTION ACQUIRES THE VALUES OF ALL THE CHANNELS SUCCESIVELY
     BY ACCESSING THE DATA AQUISITION CARD THROUGH THE ENHANCED PARALLEL PORT*/
  void  acq1(int ch_no)
   {
     sum = 0;
     for(b1=0;b1<20;b1++)/* Acquire n samples for each channel*/
     {

     outportb(CONTROL_PORT,0x04);/* Configure the EPP for bidirectional mode*/

     outportb(ADD_PORT,ch_no);/* Write the channel number whose value
                                  is to be acquired*/

     hi_byte=inportb(DATA_PORT);/* Read the 8 LSBs out of the 12 bit
                                   output from the card*/

     lo_byte=(inportb(DATA_PORT) & 0x0f);/* Read the 4 MSBs*/

     fi_byte=(hi_byte<<4 + lo_byte);/* Convert the data read
                                       to a 12 bit format*/

     sam[b1] = ((float)fi_byte * 10.0/4095.0);
     sum = sum + sam[b1];
     }
     acquire1[ch_no] = sum/20.0;
     return;
   }
 void border()
 {
   for(j1=2;j1<79;j1++)
   {
     gotoxy(j1,1);
     cprintf("%c",205);
     gotoxy(j1,25);
     cprintf("%c",205);
   }
   for(j1=2;j1<25;j1++)
   {
     gotoxy(1,j1);
     cprintf("%c",186);
     gotoxy(79,j1);
     cprintf("%c",186);
   }
   gotoxy(1,1);
   cprintf("%c",201);
   gotoxy(79,1);
   cprintf("%c",187);
   gotoxy(1,25);
   cprintf("%c",200);
   gotoxy(79,25);
   cprintf("%c",188);
}
 void paint()
```

```
   {
    window(1,1,80,25);
    textbackground(BLUE);
    textcolor(WHITE);
    clrscr();
    border();
    gotoxy(6,1);
    highvideo();
    textbackground(CYAN);
    textcolor(WHITE);
    cprintf("   REMOTE MEASUREMENT OF PRESSURE,TEMPERATURE  & RELATIVE-HUMIDITY \
n");
    textbackground(BLUE);
    textcolor(WHITE);
    lowvideo();
   }
```

```
/********************************CHARCLIENT.C********************************
             This program acts as a client and establishes connection
    with the server(PC): using internet domain sockets and requests the
    server to acquire values from the DAS card connected
    to the server.
 ****************************************************************************/


#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#define SERV_TCP_PORT 8005
#define SERV_HOST_ADDR "192.168.3.4"   /*IP address of the server*/

unsigned int *test;
char *pname,
     command[7];
int  charflag,
     rdflag,
     c,i,j,d,h,b,num=0;
float ch[8],
      rev[8],
      acq[16];

char str1[7]="ACQ 100";
char str[200],str2[200];

main(argc, argv)
int argc;
char *argv[];
{
        int sockfd,error;
        struct sockaddr_in servaddr;
        pname=argv[0];
        bzero((char *)&servaddr,sizeof(servaddr));

        /*Intialising the socket address specific for the internet socket*/

        servaddr.sin_family=AF_INET; /* internet family address */
        servaddr.sin_addr.s_addr=inet_addr(SERV_HOST_ADDR); /*32-bit netid*/
        servaddr.sin_port=htons(SERV_TCP_PORT); /*16 -bit port number */

        /*        Opens a stream socket using internet protocols    */
        if((sockfd=socket(AF_INET,SOCK_STREAM,0))<0)
               printf("error opening the socket");

         /*    Establishes a connection with the server             */

        if(connect(sockfd,(struct sockaddr *)&servaddr,sizeof(servaddr))<0)
                printf("error connecting the socket...\n");

         else
         {

                 /*  sends a command for acquisition to the server*/

                  charflag=write(sockfd,str1,sizeof(str1));
                  printf("%d\n%d",charflag,sizeof(ch));
                  if(charflag<0)
                  printf("error in writing\n");
                  printf("\n waiting to receive values from server..... ");

                  /*   Reads the values of the 8 channnels  and prints it*/

                  rdflag=read(sockfd,str,sizeof(ch));
                 printf("\nthe number of bytes read = %d\n",rdflag);

                 for(i=rdflag-1;i>=0;i--)
                 {
                  str2[rdflag -1-i]=str[i];
                 }
                 memcpy(ch,str2,rdflag);

                /* Swap the bytes to convert from little endian to
                    big endian format */

                for(h=0,d=7;h<7,d>=0;h++,d--)
                {
                   rev[h]=ch[d];
                   printf("%f\n",rev[h]);
                }
         }
        close(sockfd);

}
```

APPENDIX C

ANALOG DEVICES

Complete 12-Bit A/D Converter

AD574A*

FEATURES

Complete 12-Bit A/D Converter with Reference and Clock
8- and 16-Bit Microprocessor Bus Interface
Guaranteed Linearity Over Temperature
0 to +70°C – AD574AJ, K, L
−55°C to +125°C – AD574AS, T, U
No Missing Codes Over Temperature
35µs Maximum Conversion Time
Buried Zener Reference for Long-Term Stability and Low Gain T.C. 10ppm/°C max AD574AL
12.5ppm/°C max AD574AU
Ceramic DIP, Plastic DIP or PLCC Package

PRODUCT DESCRIPTION

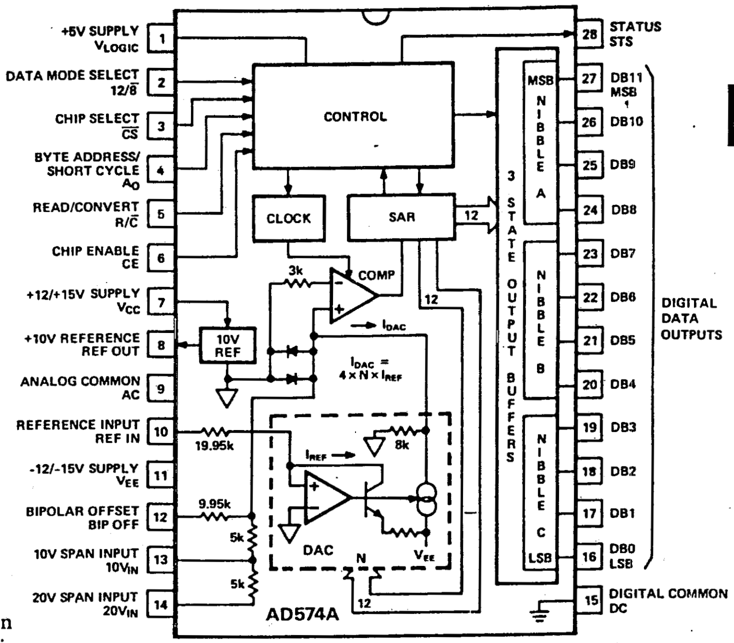
The AD574A is a complete 12-bit successive-approximation analog-to-digital converter with 3-state output buffer circuitry for direct interface to an 8- or 16-bit microprocessor bus. A high-precision voltage reference and clock are included on-chip, and the circuit guarantees full-rated performance without external circuitry or clock signals.

The AD574A design is implemented using Analog Devices' Bipolar/I^{2}L process, and integrates all analog and digital functions on one chip. Offset, linearity and scaling errors are minimized by active laser-trimming of thin-film resistors at the wafer stage. The voltage reference uses an implanted buried Zener for low noise and low drift. On the digital side, I^{2}L logic is used for the successive-approximation register, control circuitry and 3-state output buffers.

The AD574A is available in six different grades. The AD574AJ, K, and L grades are specified for operation over the 0 to +70°C temperature range. The AD574AS, T, and U are specified for the −55°C to +125°C range. All grades are available in a 28-pin hermetically-sealed ceramic DIP. The J, K, and L grades are also available in a 28-pin plastic DIP and PLCC.

The S, T, and U grades are available with optional processing to MIL-STD-883C Class B. The Analog Devices' Military Products Databook should be consulted for details on /883B testing of the AD574A.

AD574A BLOCK DIAGRAM AND PIN CONFIGURATION



PRODUCT HIGHLIGHTS

1. The AD574A interfaces to most 8- or 16-bit microprocessors. Multiple-mode three-state output buffers connect directly to the data bus while the read and convert commands are taken from the control bus. The 12 bits of output data can be read either as one 12-bit word or as two 8-bit bytes (one with 8 data bits, the other with 4 data bits and 4 trailing zeros).
2. The precision, laser-trimmed scaling and bipolar offset resistors provide four calibrated ranges: 0 to +10 and 0 to +20 volts unipolar, −5 to +5 and −10 to +10 volts bipolar. Typical bipolar offset and full-scale calibration errors of ±0.1% can be trimmed to zero with one external component each.
3. The internal buried Zener reference is trimmed to 10.00 volts with 0.2% maximum error and 15ppm/°C typical T.C. The reference is available externally and can drive up to 1.5mA beyond the requirements of the reference and bipolar offset resistors.

***Protected by U.S. Patent Nos. 3,803,590; 4,213,806; 4,511,413; RE 28,633.**

CIRCUIT OPERATION

The AD574A is a complete 12-bit A/D converter which requires no external components to provide the complete successive-approximation analog-to-digital conversion function. A block diagram of the AD574A is shown in Figure 1.

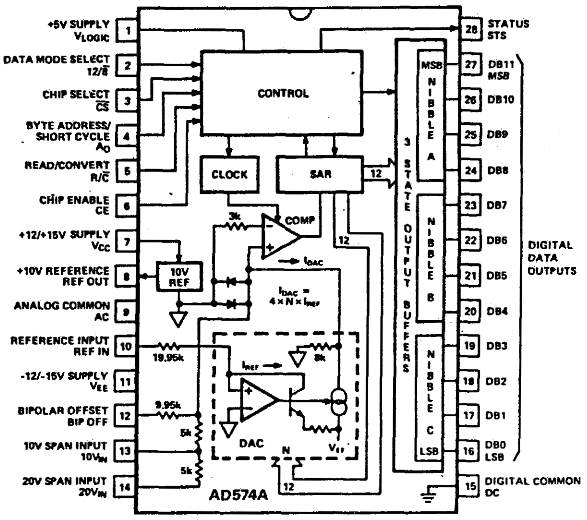


Figure 1. Block Diagram of AD574A 12-Bit A-to-D Converter

When the control section is commanded to initiate a conversion (as described later), it enables the clock and resets the successive-approximation register (SAR) to all zeros. Once a conversion cycle has begun, it cannot be stopped or re-started and data is not available from the output buffers. The SAR, timed by the clock, will sequence through the conversion cycle and return an end-of-convert flag to the control section. The control section will then disable the clock, bring the output status flag low, and enable control functions to allow data read functions by external command.

During the conversion cycle, the internal 12-bit current output DAC is sequenced by the SAR from the most significant bit (MSB) to least significant bit (LSB) to provide an output current which accurately balances the input signal current through the 5kΩ (or 10kΩ) input resistor. The comparator determines whether the addition of each successively-weighted bit current causes the DAC current sum to be greater or less than the input current; if the sum is less, the bit is left on; if more, the bit is turned off. After testing all the bits, the SAR contains a 12-bit binary code which accurately represents the input signal to within ± 1/2LSB.

The temperature-compensated buried zener reference provides the primary voltage reference to the DAC and guarantees excellent stability with both time and temperature. The reference is trimmed to 10.00 volts ±0.2%; it can supply up to 1.5mA to an external load in addition to the requirements of the reference input resistor (0.5mA) and bipolar offset resistor (1mA) when the AD574A is powered from ±15V supplies. If the AD574A is used with ±12V supplies, or if external current must be supplied over the full temperature range, an external buffer amplifier is recommended. Any external load on the AD574A reference must remain constant during conversion. The thin-film application resistors are trimmed to match the full-scale output current of the DAC. There are two 5kΩ input scaling resistors to allow either a 10 volt or 20 volt span. The 10kΩ bipolar offset resistor is grounded for unipolar operation and connected to the 10 volt reference for bipolar operation.

DRIVING THE AD574 ANALOG INPUT

The internal circuitry of the AD574 dictates that its analog input be driven by a low source impedance. Voltage changes at the current summing node of the internal comparator result in abrupt modulations of the current at the analog input. For accurate 12-bit conversions the driving source must be capable of holding a constant output voltage under these dynamically changing load conditions.

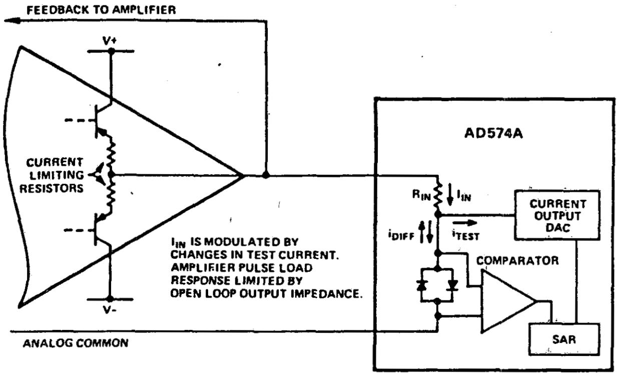


Figure 2. Op Amp – AD574A Interface

The output impedance of an op amp has an open-loop value which, in a closed loop, is divided by the loop gain available at the frequency of interest. The amplifier should have acceptable loop gain at 500kHz for use with the AD574A. To check whether the output properties of a signal source are suitable, monitor the AD574's input with an oscilloscope while a conversion is in progress. Each of the 12 disturbances should subside in 1μs or less.

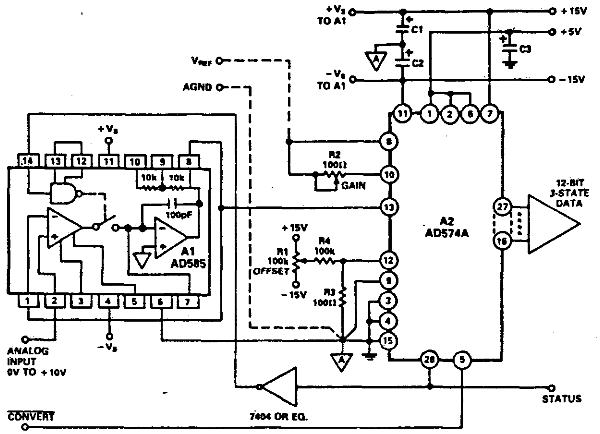
For applications involving the use of a sample-and-hold amplifier, the AD585 is recommended. The AD711 or AD544 op amps are recommended for dc applications.

SAMPLE-AND-HOLD AMPLIFIERS

Although the conversion time of the AD574A is a maximum of 35μs, to achieve accurate 12-bit conversions of frequencies greater than a few Hz requires the use of a sample-and-hold amplifier (SHA). If the voltage of the analog input signal driving the AD574A changes by more than 1/2LSB over the time interval needed to make a conversion, then the input requires a SHA.

The AD585 is a high-linearity SHA capable of directly driving the analog input of the AD574A. The AD585's fast acquisition time, low aperture and low aperture jitter are ideally suited for high-speed data acquisition systems. Consider the AD574A converter with a 35μs conversion time and an input signal of 10V p-p: the maximum frequency which may be applied to achieve rated accuracy is 1.5Hz. However, with the addition of an AD585, as shown in Figure 3, the maximum frequency increases to 26kHz.

The AD585's low output impedance, fast-loop response, and low droop maintain 12-bits of accuracy under the changing load conditions that occur during a conversion, making it suitable for use in high-accuracy conversion systems. Many other SHAs cannot achieve 12-bits of accuracy and can thus compromise a system. The AD585 is recommended for AD574A applications requiring a sample and hold.



NOTE
1. C1, C2, C3 ARE 47μF TANTALUM, BYPASSED BY 0.1μF CERAMIC. LOCATE AT ASSOCIATED A2 PINS.

Figure 3. AD574A with AD585 Sample and Hold

SUPPLY DECOUPLING AND LAYOUT CONSIDERATIONS

It is critically important that the AD574A power supplies be filtered, well regulated, and free from high-frequency noise. Use of noisy supplies will cause unstable output codes. Switching power supplies are not recommended for circuits attempting to achieve 12-bit accuracy unless great care is used in filtering any switching spikes present in the output. Remember that a few millivolts of noise represents several counts of error in a 12-bit ADC.

Decoupling capacitors should be used on all power supply pins; the +5V supply decoupling capacitor should be connected directly from pin 1 to pin 15 (digital common) and the $+V_{CC}$ and $-V_{EE}$ pins should be decoupled directly to analog common (pin 9). A suitable decoupling capacitor is a 4.7μF tantalum type in parallel with a 0.1μF disc ceramic type.

Circuit layout should attempt to locate the AD574A, associated analog input circuitry, and interconnections as far as possible from logic circuitry. For this reason, the use of wire-wrap circuit construction is not recommended. Careful printed-circuit construction is preferred.

GROUNDING CONSIDERATIONS

The analog common at pin 9 is the ground reference point for the internal reference and is thus the "high quality" ground for the AD574A; it should be connected directly to the analog reference point of the system. In order to achieve all of the high-accuracy performance available from the AD574A in an environment of high digital noise content, the analog and digital commons should be connected together at the package. In some situations, the digital common at pin 15 can be connected to the most convenient ground reference point; analog power return is preferred.

UNIPOLAR RANGE CONNECTIONS FOR THE AD574A

The AD574A contains all the active components required to perform a complete 12-bit A/D conversion. Thus, for most situations, all that is necessary is connection of the power supplies (+5, +12/+15 and −12/−15 volts), the analog input, and the conversion initiation command, as discussed on the next page. Analog input connections and calibration are easily accomplished; the unipolar operating mode is shown in Figure 4.

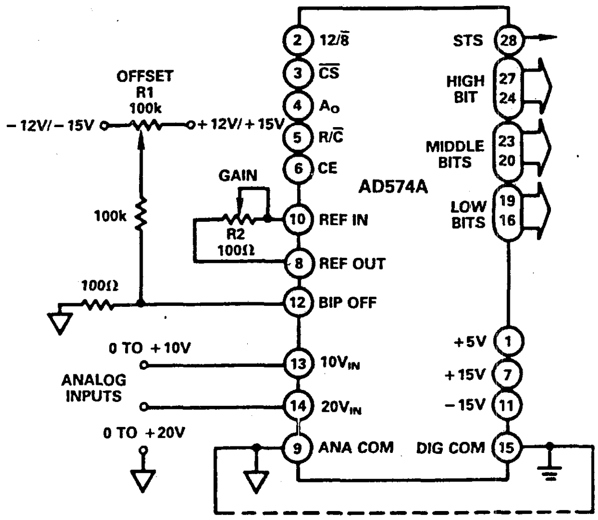


Figure 4. Unipolar Input Connections

All of the thin-film application resistors of the AD574A are trimmed for absolute calibration. Therefore, in many applications, no calibration trimming will be required. The absolute accuracy for each grade is given in the specification tables. For example, if no trims are used, the AD574AK guarantees ±1LSB max zero offset error and ±0.25% (10LSB) max full-scale error. (Typical full-scale error is ±2LSB.) If the offset trim is not required, pin 12 can be connected directly to pin 9; the two resistors and trimmer for pin 12 are then not needed. If the full-scale trim is not needed, a 50Ω ±1% metal film resistor should be connected between pin 8 and pin 10.

The analog input is connected between pin 13 and pin 9 for a 0 to +10V input range, between 14 and pin 9 for a 0 to +20V input range. The AD574A easily accommodates an input signal beyond the supplies. For the 10 volt span input, the LSB has a nominal value of 2.44mV; for the 20 volt span, 4.88mV. If a 10.24V range is desired (nominal 2.5mV/bit), the gain trimmer (R2) should be replaced by a 50Ω resistor, and a 200Ω trimmer inserted in series with the analog input to pin 13 for a full-scale range of 20.48V (5mV/bit), use a 500Ω trimmer into pin 14. The gain trim described below is now done with these trimmers. The nominal input impedance into pin 13 is 5kΩ, and 10kΩ into pin 14.

UNIPOLAR CALIBRATION

The AD574A is intended to have a nominal 1/2LSB offset so that the exact analog input for a given code will be in the middle of that code (halfway between the transitions to the codes above and below it). Thus, the first transition (from 0000 0000 0000 to 0000 0000 0001) will occur for an input level of + 1/2LSB (1.22mV for 10V range).

If pin 12 is connected to pin 9, the unit will behave in this manner, within specifications. If the offset trim (R1) is used, it should be trimmed as above, although a different offset can be set for a particular system requirement. This circuit will give approximately ± 15mV of offset trim range.

The full-scale trim is done by applying a signal 1 1/2LSB below the nominal full scale (9.9963 for a 10V range). Trim R2 to give the last transition (1111 1111 1110 to 1111 1111 1111).

BIPOLAR OPERATION

The connections for bipolar ranges are shown in Figure 5. Again, as for the unipolar ranges, if the offset and gain specifications are sufficient, one or both of the trimmers shown can be replaced by a 50Ω ± 1% fixed resistor. Bipolar calibration is similar to unipolar calibration. First, a signal ½LSB above negative full scale (−4.9988V for the ± 5V range) is applied and R1 is trimmed to give the first transition (0000 0000 0000 to 0000 0000 0001). Then a signal 1½LSB below positive full scale (+4.9963V for the ± 5V range) is applied and R2 trimmed to give the last transition (1111 1111 1110 to 1111 1111 1111).

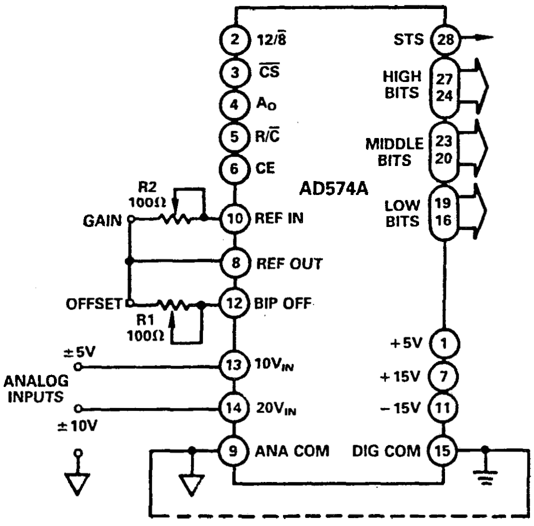
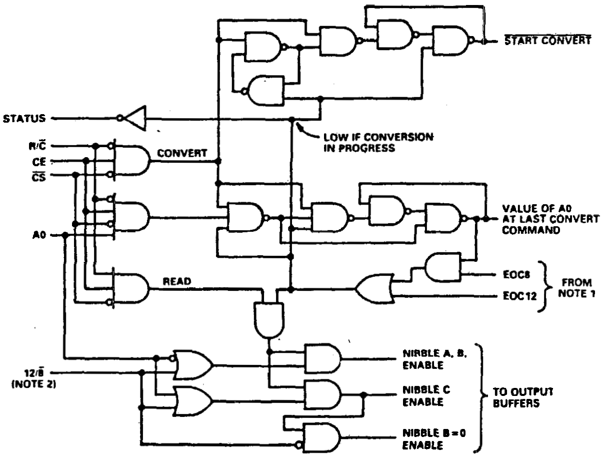


Figure 5. Bipolar Input Connections

CONTROL LOGIC

The AD574A contains on-chip logic to provide conversion initiation and data read operations from signals commonly available in microprocessor systems. Figure 6 shows the internal logic circuitry of the AD574A.

The control signals CE, $\overline{CS}$, and R/$\overline{C}$ control the operation of the converter. The state of R/$\overline{C}$ when CE and $\overline{CS}$ are both asserted determines whether a data read (R/$\overline{C}$ = 1) or a convert (R/$\overline{C}$ = 0) is in progress. The register control inputs A_O and 12/$\overline{8}$ control conversion length and data format. The A_O line is usually tied to the least significant bit of the address bus. If a conversion is started with A_O low, a full 12-bit conversion cycle is initiated. If A_O is high during a convert start, a shorter 8-bit conversion cycle results. During data read operations, A_O determines whether the three-state buffers containing the 8 MSBs of the conversion result (A_O = 0) or the 4 LSBs (A_O = 1) are enabled. The 12/$\overline{8}$ pin determines whether the output data is to be organized as two 8-bit words (12/$\overline{8}$ tied to DIGITAL COMMON) or a single 12-bit word (12/$\overline{8}$ tied to V_{LOGIC}). The 12/$\overline{8}$ pin is not TTL-compatible and must be hard-wired to either V_{LOGIC} or DIGITAL COMMON. In the 8-bit mode, the byte addressed when A_O is high contains the 4 LSBs from the conversion followed by four trailing zeroes. This organization allows the data lines to be overlapped for direct interface to 8-bit buses without the need for external three-state buffers.



NOTE 1: WHEN $\overline{START}$ $\overline{CONVERT}$ GOES LOW, THE EOC (END OF CONVERSION) SIGNALS GO LOW. EOC8 RETURNS HIGH AFTER AN 8-BIT CONVERSION CYCLE IS COMPLETE, AND EOC12 RETURNS HIGH WHEN ALL 12 BITS HAVE BEEN CONVERTED. THE EOC SIGNALS PREVENT DATA FROM BEING READ DURING CONVERSIONS.

NOTE 2: 12/$\overline{8}$ IS NOT A TTL-COMPATIBLE INPUT AND SHOULD ALWAYS BE WIRED DIRECTLY TO V_{LOGIC} OR DIGITAL COMMON.

Figure 6. AD574A Control Logic

It is not recommended that A_O change state during a data read operation. Asymmetrical enable and disable times of the three-state buffers could cause internal bus contention resulting in potential damage to the AD574A.

An output signal, STS, indicates the status of the converter. STS goes high at the beginning of a conversion and returns low when the conversion cycle is complete.

CE	$\overline{CS}$	R/$\overline{C}$	12/$\overline{8}$	A_O	Operation
0	X	X	X	X	None
X	1	X	X	X	None
1	0	0	X	0	Initiate 12-Bit Conversion
1	0	0	X	1	Initiate 8-Bit Conversion
1	0	1	Pin 1	X	Enable 12-Bit Parallel Output
1	0	1	Pin 15	0	Enable 8 Most Significant Bits
1	0	1	Pin 15	1	Enable 4LSBs + 4 Trailing Zeroes

Table I. AD574A Truth Table

TIMING

The AD574A is easily interfaced to a wide variety of microprocessors and other digital systems. The following discussion of the timing requirements of the AD574A control signals should provide the system designer with useful insight into the operation of the device.

CONVERT START TIMING – FULL CONTROL MODE

Symbol	Parameter	Min	Typ	Max	Units
t_{DSC}	STS Delay from CE			400	ns
t_{HEC}	CE Pulse Width	300			ns
t_{SSC}	$\overline{CS}$ to CE Setup	300			ns
t_{HSC}	$\overline{CS}$ Low During CE High	200			ns
t_{SRC}	R/$\overline{C}$ to CE Setup	250			ns
t_{HRC}	R/$\overline{C}$ Low During CE High	200			ns
t_{SAC}	A_O to CE Setup	0			ns
t_{HAC}	A_O Valid During CE High	300			ns
t_C	Conversion Time				
	8-Bit Cycle	10		24	µs
	12-Bit Cycle	15		35	µs

Figure 7 shows a complete timing diagram for the AD574A convert start operation. R/$\overline{C}$ should be low before both CE and $\overline{CS}$ are asserted; if R/$\overline{C}$ is high, a read operation will momentarily occur, possibly resulting in system bus contention. Either CE or $\overline{CS}$ may be used to initiate a conversion; however, use of CE is recommended since it includes one less propagation delay than $\overline{CS}$ and is the faster input. In Figure 7, CE is used to initiate the conversion.

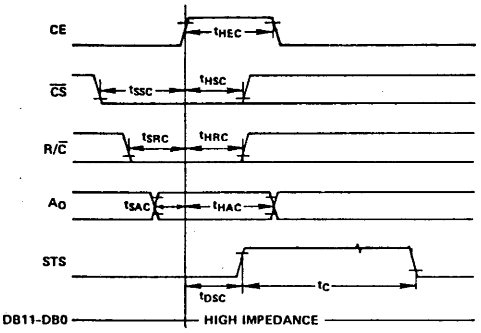


Figure 7. Convert Start Timing

Once a conversion is started and the STS line goes high, convert start commands will be ignored until the conversion cycle is complete. The output data buffers cannot be enabled during conversion.

Figure 8 shows the timing for data read operations. During data read operations, access time is measured from the point where CE and R/$\overline{C}$ both are high (assuming $\overline{CS}$ is already low). If $\overline{CS}$ is used to enable the device, access time is extended by 100ns.

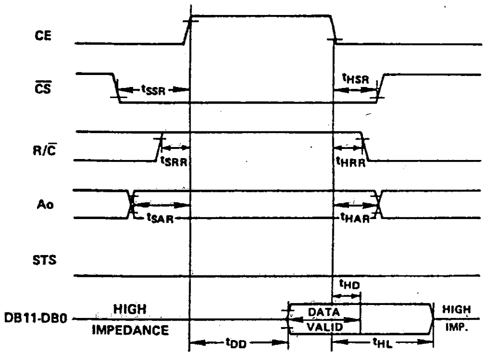


Figure 8. Read Cycle Timing

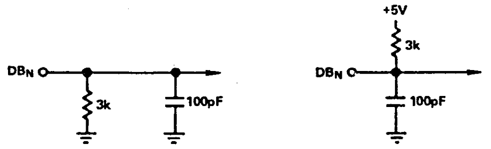
In the 8-bit bus interface mode (12/$\overline{8}$ input wired to DIGITAL COMMON), the address bit, A_O, must be stable at least 150ns prior to $\overline{CE}$ going high and must remain stable during the entire read cycle. If A_O is allowed to change, damage to the AD574A output buffers may result.

READ TIMING – FULL CONTROL MODE

Symbol	Parameter	Min	Typ	Max	Units
t_{DD}[1]	Access Time (from CE)			200	ns
t_{HD}	Data Valid after CE Low	25			ns
t_{HL}[2]	Output Float Delay			100	ns
t_{SSR}	$\overline{CS}$ to CE Setup	150			ns
t_{SRR}	R/$\overline{C}$ to CE Setup	0			ns
t_{SAR}	A_O to CE Setup	150			ns
t_{HSR}	$\overline{CS}$ Valid After CE Low	50			ns
t_{HRR}	R/$\overline{C}$ High After CE Low	0			ns
t_{HAR}	A_O Valid After CE low	50			ns

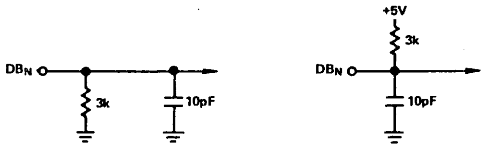
[1] t_{DD} is measured with the load circuit of Figure 8 and defined as the time required for an output to cross 0.4V or 2.4V.

[2] t_{HL} is defined as the time required for the data lines to change 0.5V when loaded with the circuit of Figure 9.



a. High-Z to Logic 1 *b. High-Z to Logic 0*

Figure 9. Load Circuit for Access Time Test



a. Logic 1 to High-Z *b. Logic 0 to High-Z*

Figure 10. Load Circuit for Output Float Delay Test

"STAND-ALONE" OPERATION

The AD574A can be used in a "stand-alone" mode, which is useful in systems with dedicated input ports available and thus not requiring full bus interface capability.

In this mode, CE and 12/$\overline{8}$ are wired high, $\overline{CS}$ and A_O are wired low, and conversion is controlled by R/$\overline{C}$. The three-state buffers are enabled when R/$\overline{C}$ is high and a conversion starts when R/$\overline{C}$ goes low. This allows two possible control signals – a high pulse or a low pulse. Operation with a low pulse is shown in Figure 11. In this case, the outputs are forced into the high-impedance state in response to the falling edge of R/$\overline{C}$ and return

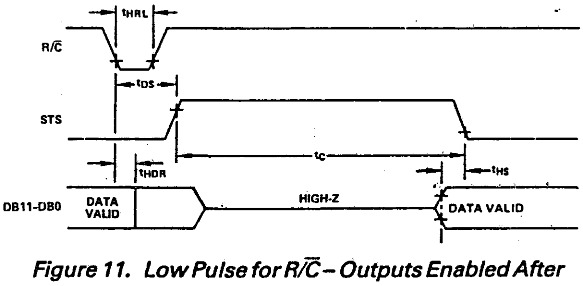


Figure 11. Low Pulse for R/$\overline{C}$ – Outputs Enabled After Conversion

TYPES SN54LS373, SN54LS374, SN54S373, SN54S374, SN74LS373, SN74LS374, SN74S373, SN74S374 OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

OCTOBER 1975—REVISED APRIL 1985

- Choice of 8 Latches or 8 D-Type Flip-Flops In a Single Package
- 3-State Bus-Driving Outputs
- Full Parallel-Access for Loading
- Buffered Control Inputs
- Clock/Enable Input Has Hysteresis to Improve Noise Rejection ('S373 and 'S374)
- P-N-P Inputs Reduce D-C Loading on Data Lines ('S373 and 'S374)

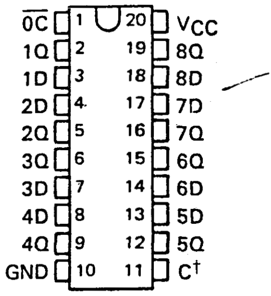
'LS373, 'S373
FUNCTION TABLE

OUTPUT ENABLE	ENABLE LATCH	D	OUTPUT
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

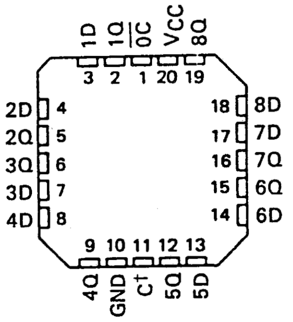
'LS374, 'S374
FUNCTION TABLE

OUTPUT ENABLE	CLOCK	D	OUTPUT
L	↑	H	H
L	↑	L	L
L	L	X	Q_0
H	X	X	Z

SN54LS373, SN54LS374, SN54S373,
SN54S374 . . . J PACKAGE
SN74LS373, SN74LS374, SN74S373,
SN74S374 . . . DW, J OR N PACKAGE
(TOP VIEW)



SN54LS373, SN54LS374, SN54S373,
SN54S374 . . . FK PACKAGE
SN74LS373, SN74LS374, SN74S373,
SN74S374 . . . FN PACKAGE
(TOP VIEW)



† C for 'LS373 and 'S373; CLK for 'LS374 and 'S374.

description

These 8-bit registers feature three-state outputs designed specifically for driving highly-capacitive or relatively low-impedance loads. The high-impedance third state and increased high-logic-level drive provide these registers with the capability of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pull-up components. They are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers.

The eight latches of the 'LS373 and 'S373 are transparent D-type latches meaning that while the enable (C) is high the Q outputs will follow the data (D) inputs. When the enable is taken low the output will be latched at the level of the data that was set up.

PRODUCTION DATA
This document contains information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TYPES SN54LS373, SN54LS374, SN54S373, SN54S374, SN74LS373, SN74LS374, SN74S373, SN74S374 OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

description (continued)

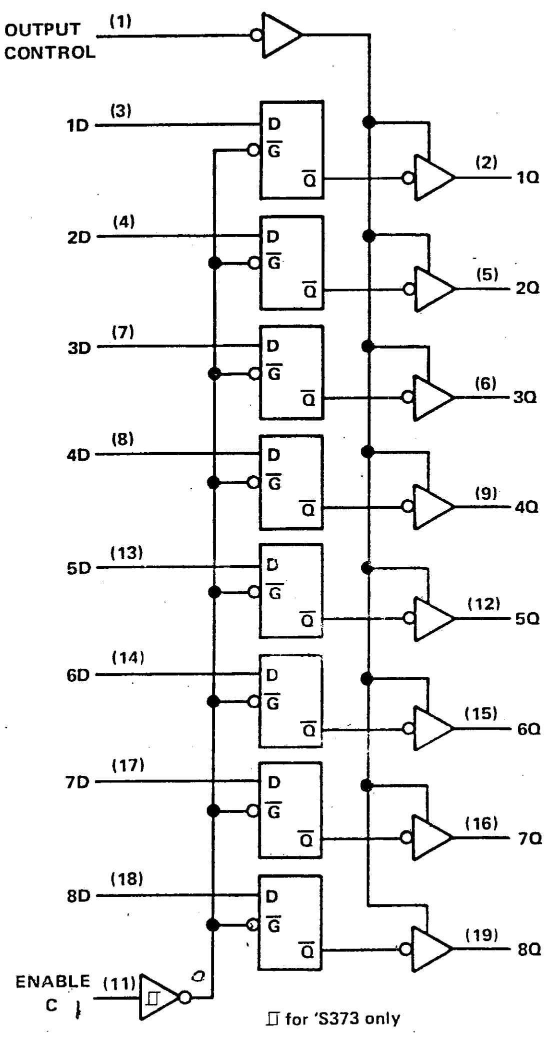
The eight flip-flops of the 'LS374 and 'S374 are edge-triggered D-type flip-flops. On the positive transition of the clock, the Q outputs will be set to the logic states that were setup at the D inputs.

Schmitt-trigger buffered inputs at the enable/clock lines of the 'S373 and 'S374 devices, simplify system design as ac and dc noise rejection is improved by typically 400 mV due to the input hysteresis. A buffered output control input can be used to place the eight outputs in either a normal logic state (high or low logic levels) or a high-impedance state. In the high-impedance state the outputs neither load nor drive the bus lines significantly.

The output control does not affect the internal operation of the latches or flip-flops. That is, the old data can be retained or new data can be entered even while the outputs are off.

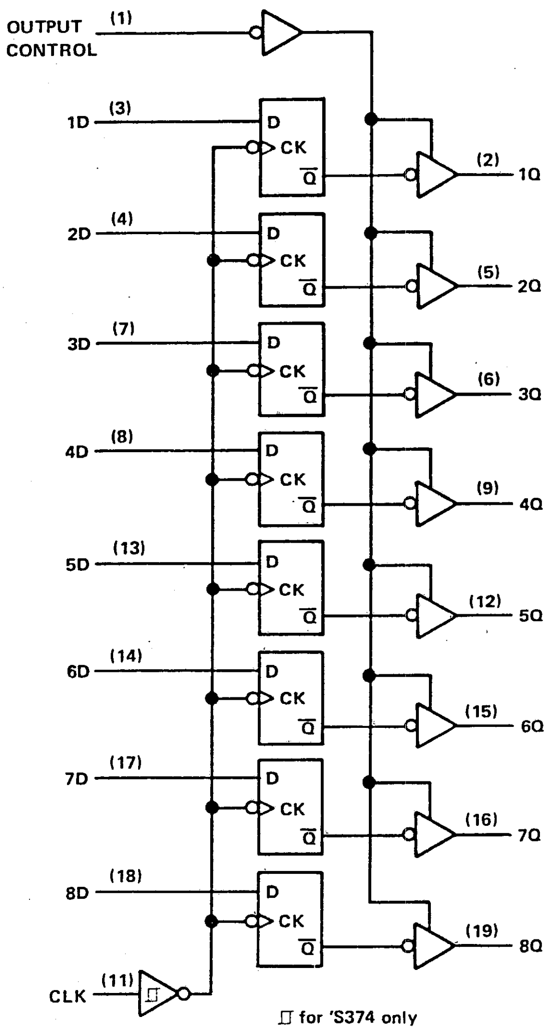
logic diagrams

'LS373, 'S373
TRANSPARENT LATCHES



⎍ for 'S373 only

'LS374, 'S374
POSITIVE-EDGE-TRIGGERED FLIP-FLOPS



⎍ for 'S374 only

Pin numbers shown on logic notation are for DW, J or N packages.

ANALOG DEVICES

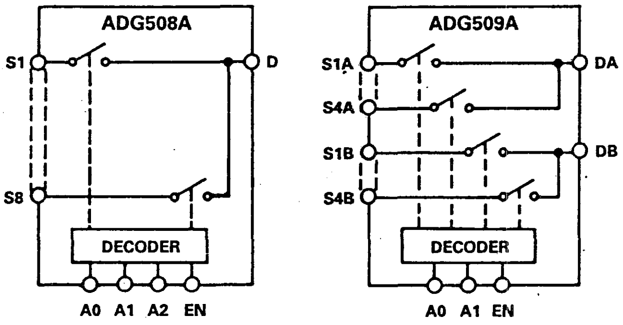
CMOS 4/8 Channel Analog Multiplexers

ADG508A/ADG509A

FEATURES

44V Supply Maximum Rating
V_{SS} to V_{DD} Analog Signal Range
Single/Dual Supply Specifications
Wide Supply Ranges (10.8V to 16.5V)
Extended Plastic Temperature Range (−40°C to +85°C)
Low Power Dissipation (28mW max)
Low Leakage (20pA typ)
Superior Alternative to:
DG508A, HI-508
DG509A, HI-509

ADG508A/ADG509A FUNCTIONAL BLOCK DIAGRAMS



GENERAL DESCRIPTION

The ADG508A and ADG509A are CMOS monolithic analog multiplexers with 8 channels and dual 4 channels respectively. The ADG508A switches one of 8 inputs to a common output depending on the state of three binary addresses and an enable input. The ADG509A switches one of 4 differential inputs to a common differential output depending on the state of two binary addresses and an enable input. Both devices have TTL and 5V CMOS logic compatible digital inputs.

The ADG508A and ADG509A are designed on an enhanced LC^{2}MOS process which gives an increased signal capability of V_{SS} to V_{DD} and enables operation over a wide range of supply voltages. The devices can comfortably operate anywhere in the 10.8V to 16.5V single or dual supply range. These multiplexers also feature high switching speeds and low R_{ON}.

PRODUCT HIGHLIGHTS

1. Single/Dual Supply Specifications with a Wide Tolerance:
The devices are specified in the 10.8V to 16.5V range for both single and dual supplies.
2. Extended Signal Range:
The enhanced LC^{2}MOS processing results in a high breakdown and an increased analog signal range of V_{SS} to V_{DD}.
3. Break-Before-Make Switching:
Switches are guaranteed break-before-make so that input signals are protected against momentary shorting.
4. Low Leakage:
Leakage currents in the range of 20pA make these multiplexers suitable for high precision circuits.

ORDERING INFORMATION[1]

Temperature Range and Package Options[2]

−40°C to +85°C	−40°C to +85°C	−55°C to +125°C
Plastic DIP (N-16) ADG508AKN ADG509AKN	**Hermetic (Q-16)** ADG508ABQ ADG509ABQ	**Hermetic (Q-16)** ADG508ATQ ADG509ATQ
PLCC[3] (P-20A) ADG508AKP ADG509AKP		**LCCC[4] (E-20A)** ADG508ATE ADG509ATE

NOTES
[1] To order MIL-STD-883, Class B processed parts, add /883B to part number. Contact your local sales office for military data sheet.
[2] See Section 13 for package outline information.
[3] PLCC: Plastic Leaded Chip Carrier.
[4] LCCC: Leadless Ceramic Chip Carrier.

SPECIFICATIONS

Dual Supply (V_{DD} = +10.8V to +16.5V, V_{SS} = −10.8V to −16.5V unless otherwise noted)

	ADG508A ADG509A K Version		ADG508A ADG509A B Version		ADG508A ADG509A T Version			
		−40°C to		−40°C to		−55°C to		
Parameter	+25°C	+85°C	25°C	+85°C	+25°C	+125°C	Units	Comments
ANALOG SWITCH								
Analog Signal Range	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V min	
	V_{DD}	V_{DD}	V_{DD}	V_{DD}	V_{DD}	V_{DD}	V max	
R_{ON}	280		280		280		Ω typ	$-10V \leq V_S \leq +10V, I_{DS} = 1mA$
	450	600	450	600	450	600	Ω max	
	300	400	300	400			Ω max	$V_{DD} = 15V(\pm 10\%), V_{SS} = -15V(\pm 10\%)$
					300	400	Ω max	$V_{DD} = 15V(\pm 5\%), V_{SS} = -15V(\pm 5\%)$
R_{ON} Drift	0.6		0.6		0.6		%/°C typ	$V_S = 0, I_{DS} = 1mA$
R_{ON} Match	5		5		5		% typ	$-10V \leq V_S \leq +10V, I_{DS} = 1mA$
I_S (OFF), Off Input Leakage	0.02		0.02		0.02		nA typ	$V_{S1} = \pm 10V, V_D = V_{S2}$ to $V_{SN} = \mp 10V$
	1	50	1	50	1	50	nA max	
I_D (OFF), Off Output Leakage	0.04		0.04		0.04		nA typ	V_{S1} to $V_{SN} = \pm 10V, V_D = \mp 10V$
ADG508A	1	100	1	100	1	100	nA max	
ADG509A	1	50	1	50	1	50	nA max	
I_D (ON), On Channel Leakage	0.04		0.04		0.04		nA typ	V_{S2} to $V_{SN} = \pm 10V, V_D = V_{S1} = \mp 10V$
ADG508A	1	100	1	100	1	100	nA max	
ADG509A	1	50	1	50	1	50	nA max	
I_{DIFF}, Differential Off Output Leakage (ADG509A only)		25		25		25	nA max	$V_{S1A/B}$ to $V_{S4A/B} = \pm 10V, V_{DA} = V_{DB} = \mp 10V$
DIGITAL CONTROL								
V_{INH}, Input High Voltage		2.4		2.4		2.4	V min	
V_{INL}, Input Low Voltage		0.8		0.8		0.8	V max	
I_{INL} or I_{INH}		1		1		1	µA max	$V_{IN} = 0$ to V_{DD}
C_{IN} Digital Input Capacitance	8		8		8		pF max	
DYNAMIC CHARACTERISTICS								
$t_{TRANSITION}$[1]	200		200		200		ns typ	$R_L = 1M\Omega, C_L = 35pF$
	300	400	300	400	300	400	ns max	
t_{OPEN}[1]	50		50		50		ns typ	$R_L = 1k\Omega, C_L = 35pF$
	25	10	25	10	25	10	ns min	
t_{ON} (EN)[1]	200		200		200		ns typ	$R_L = 1k\Omega, C_L = 35pF$
	300	400	300	400	300	400	ns max	
t_{OFF} (EN)[1]	200		200		200		ns typ	$R_L = 1k\Omega, C_L = 35pF$
	300	400	300	400	300	400	ns max	
OFF Isolation	68		68		68		dB typ	$V_{EN} = 0.8V, R_L = 1k\Omega, C_L = 15pF,$
	50		50		50		dB min	$V_S = 7V$ rms, f = 100kHz
C_S (OFF)	5		5		5		pF typ	$V_{EN} = 0.8V$
C_D (OFF)								
ADG508A	22		22		22		pF typ	$V_{EN} = 0.8V$
ADG509A	11		11		11		pF typ	
Q_{INJ}, Charge Injection	4		4		4		pC typ	$R_S = 0\Omega, C_L = 1000pF, V_S = 0V$
POWER SUPPLY								
I_{DD}	0.6		0.6		0.6		mA typ	$V_{IN} = V_{INL}$ or V_{INH}
		1.5		1.5		1.5	mA max	
I_{SS}	20		20		20		µA typ	$V_{IN} = V_{INL}$ or V_{INH}
		0.2		0.2		0.2	mA max	
Power Dissipation	10		10		10		mW typ	
		28		28		28	mW max	

NOTE

[1] Sample tested at 25°C to ensure compliance.

Specifications subject to change without notice.

Single Supply (V_{DD} = +10.8V to +16.5V, V_{SS} = GND = 0V unless otherwise noted)

Parameter	ADG508A ADG509A K Version +25°C	 −40°C to +85°C	ADG508A ADG509A B Version +25°C	 −40°C to +85°C	ADG508A ADG509A T Version +25°C	 −55°C to +125°C	Units	Comments
ANALOG SWITCH								
Analog Signal Range	GND	GND	GND	GND	GND	GND	V min	
	V_{DD}	V_{DD}	V_{DD}	V_{DD}	V_{DD}	V_{DD}	V max	
R_{ON}	500		500		500		Ω typ	GND≤V_S≤+10V, I_{DS}=0.5mA
	700	1000	700	1000	700	1000	Ω max	
R_{ON} Drift	0.6		0.6		0.6		%/°C typ	V_S=0, I_{DS}=0.5mA
R_{ON} Match	5		5		5		% typ	GND≤V_S≤+10V, I_{DS}=0.5mA
I_S (OFF), Off Input Leakage	0.02		0.02		0.02		nA typ	V_{S1} = +10V/GND, V_D = V_{S2} to V_{SN} = GND/+10V
	1	50	1	50	1	50	nA max	
I_D (OFF), Off Output Leakage	0.04		0.04		0.04		nA typ	V_{S1} to V_{SN} = +10V/GND, V_D = GND/+10V
ADG508A	1	100	1	100	1	100	nA max	
ADG509A	1	50	1	50	1	50	nA max	
I_D (ON), On Channel Leakage	0.04		0.04		0.04		nA typ	V_{S2} to V_{SN} = +10V/GND, V_D = V_{S1} = GND/+10V
ADG508A	1	100	1	100	1	100	nA max	
ADG509A	1	50	1	50	1	50	nA max	
I_{DIFF}, Differential Off Output Leakage (ADG509A only)		25		25		25	nA max	$V_{S1A/B}$ to $V_{S4A/B}$ = +10V/GND, V_{DA} = V_{DB} = GND/+10V
DIGITAL CONTROL								
V_{INH}, Input High Voltage		2.4		2.4		2.4	V min	
V_{INL}, Input Low Voltage		0.8		0.8		0.8	V max	
I_{INL} or I_{INH}		1		1		1	μA max	V_{IN} = 0 to V_{DD}
C_{IN} Digital Input Capacitance	8		8		8		pF max	
DYNAMIC CHARACTERISTICS								
$t_{TRANSITION}$[1]	300		300		300		ns typ	R_L = 1MΩ, C_L = 35pF
	450	600	450	600	450	600	ns max	
t_{OPEN}[1]	50		50		50		ns typ	R_L = 1kΩ, C_L = 35pF
	25	10	25	10	25	10	ns min	
t_{ON} (EN)[1]	250		250		250		ns typ	R_L = 1kΩ, C_L = 35pF
	450	600	450	600	450	600	ns max	
t_{OFF} (EN)[1]	250		250		250		ns typ	R_L = 1kΩ, C_L = 35pF
	450	600	450	600	450	600	ns max	
OFF Isolation	68		68		68		dB typ	V_{EN} = 0.8V, R_L = 1kΩ, C_L = 15pF,
	50		50		50		dB min	V_S = 3.5V rms, f = 100kHz
C_S (OFF)	5		5		5		pF typ	V_{EN} = 0.8V
C_D (OFF)								
ADG508A	22		22		22		pF typ	V_{EN} = 0.8V
ADG509A	11		11		11		pF typ	
Q_{INJ}, Charge Injection	4		4		4		pC typ	R_S = 0Ω, C_L = 1000pF, V_S = 0V
POWER SUPPLY								
I_{DD}	0.6		0.6		0.6		mA typ	V_{IN} = V_{INL} or V_{INH}
		1.5		1.5		1.5	mA max	
Power Dissipation	10		10		10		mW typ	
		25		25		25	mW max	

NOTE
[1] Sample tested at 25°C to ensure compliance.
Specifications subject to change without notice.

TRUTH TABLES

A2	A1	A0	EN	ON SWITCH
X	X	X	0	NONE
0	0	0	1	1
0	0	1	1	2
0	1	0	1	3
0	1	1	1	4
1	0	0	1	5
1	0	1	1	6
1	1	0	1	7
1	1	1	1	8

X = Don't Care

ADG508A

A1	A0	EN	ON SWITCH PAIR
X	X	0	NONE
0	0	1	1
0	1	1	2
1	0	1	3
1	1	1	4

X = Don't Care

ADG509A

ABSOLUTE MAXIMUM RATINGS*

(T_A = 25°C unless otherwise noted)

V_{DD} to V_{SS}	44V
V_{DD} to GND	25V
V_{SS} to GND	−25V
Analog Inputs[1]	
Voltage at S, D	V_{SS} −2V to V_{DD} +2V or 20mA, Whichever Occurs First
Continuous Current, S or D	20mA
Pulsed Current S or D	
1ms Duration, 10% Duty Cycle	40mA

NOTE

[1] Overvoltage at A, EN, S or D will be clamped by diodes. Current should be limited to the Maximum Rating above.

Digital Inputs[1]	
Voltage at A, EN	V_{SS} −4V tc V_{DD} +4V oı 20mA, Whichever Occurs Firs
Power Dissipation (Any Package)	
Up to +75°C	470mW
Derates above +75°C by	6mW/°C
Operating Temperature	
Commercial (K Version)	−40°C to +85°C
Industrial (B Version)	−40°C to +85°C
Extended (T Version)	−55°C to +125°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature Range (Soldering, 10sec)	+300°C

*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

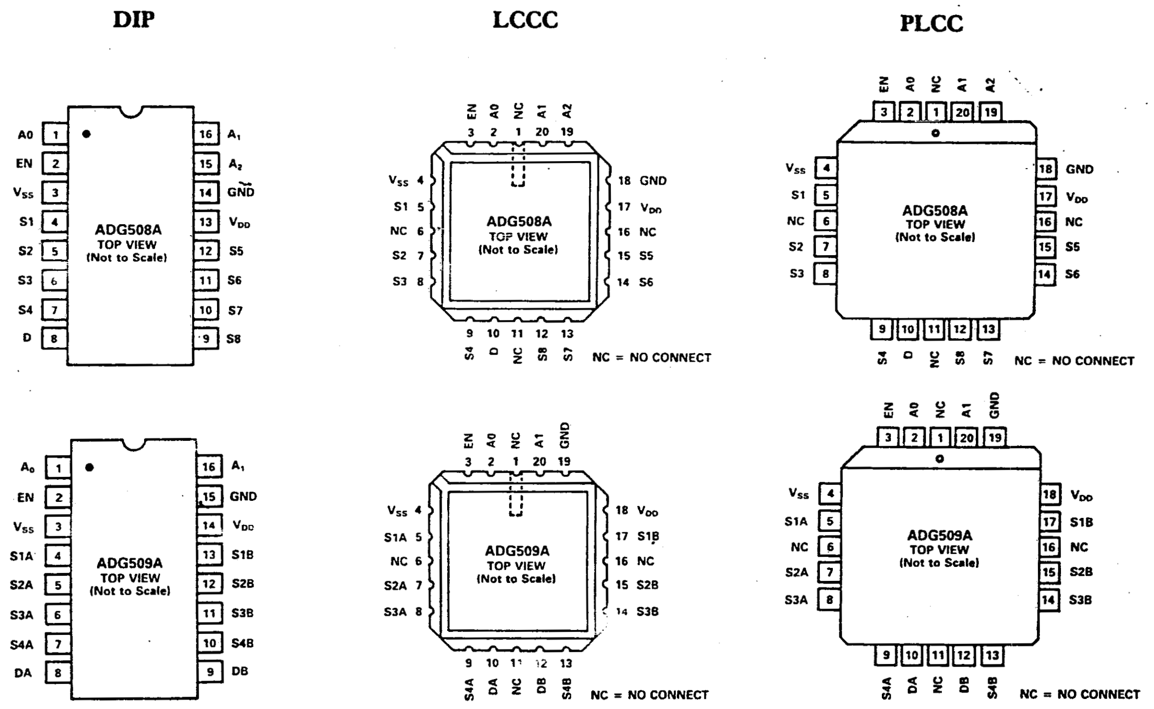
CAUTION

ESD (Electro-Static-Discharge) sensitive device. The digital control inputs are diode protected; however, permanent damage may occur on unconnected devices subject to high energy electrostatic fields. Unused devices must be stored in conductive foam or shunts. The protective foam should be discharged to the destination socket before devices are removed.

WARNING!

ESD SENSITIVE DEVICE

PIN CONFIGURATIONS



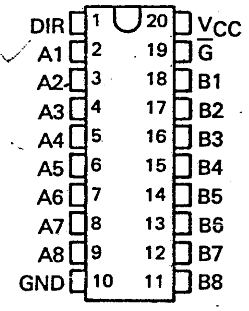
TYPES SN54LS245, SN74LS245
OCTAL BUS TRANSCEIVERS WITH 3-STATE OUTPUTS

OCTOBER 1976—REVISED APRIL 1985

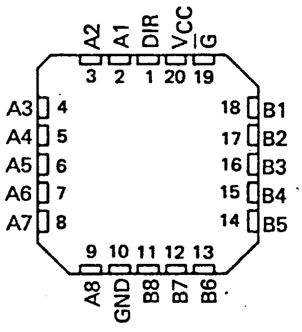
- Bi-directional Bus Transceiver in a High-Density 20-Pin Package
- 3-State Outputs Drive Bus Lines Directly
- PNP Inputs Reduce D-C Loading on Bus Lines
- Hysteresis at Bus Inputs Improve Noise Margins
- Typical Propagation Delay Times, Port-to-Port . . . 8 ns

TYPE	I_{OL} (SINK CURRENT)	I_{OH} (SOURCE CURRENT)
SN54LS245	12 mA	-12 mA
SN74LS245	24 mA	-15 mA

SN54LS245 . . . J PACKAGE
SN74LS245 . . . DW, J OR N PACKAGE
(TOP VIEW)



SN54LS245 . . . FK PACKAGE
SN74LS245 . . . FN PACKAGE
(TOP VIEW)



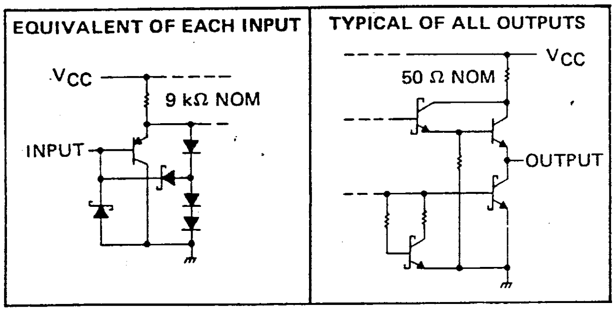
description

These octal bus transceivers are designed for asynchronous two-way communication between data buses. The control function implementation minimizes external timing requirements.

The devices allow data transmission from the A bus to the B bus or from the B bus to the A bus depending upon the logic level at the direction control (DIR) input. The enable input (G̅) can be used to disable the device so that the buses are effectively isolated.

The SN54LS245 is characterized for operation over the full military temperature range of −55°C to 125°C. The SN74LS245 is characterized for operation from 0°C to 70°C.

schematics of inputs and outputs



FUNCTION TABLE

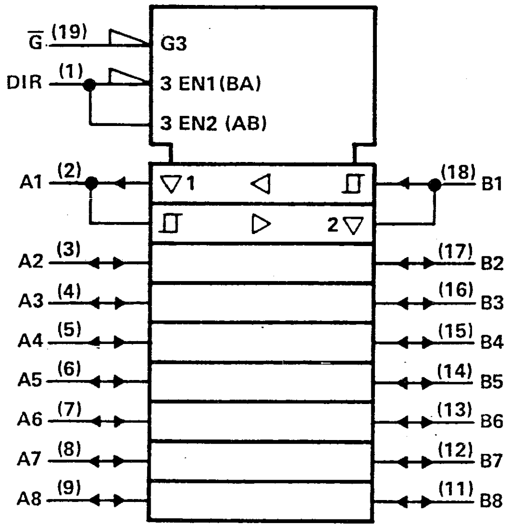
ENABLE G̅	DIRECTION CONTROL DIR	OPERATION
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

H = high level, L = low level, X = irrelevant

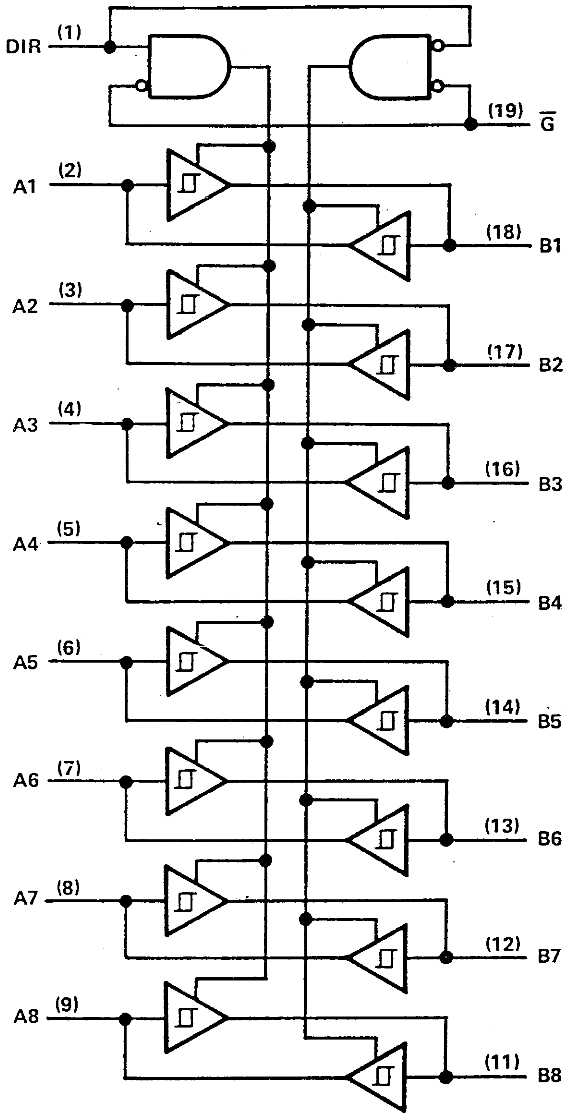
PRODUCTION DATA
This document contains information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS INSTRUMENTS
POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

logic symbol



logic diagram (positive logic)



Pin numbers shown on logic notation are for DW, J or N packages.

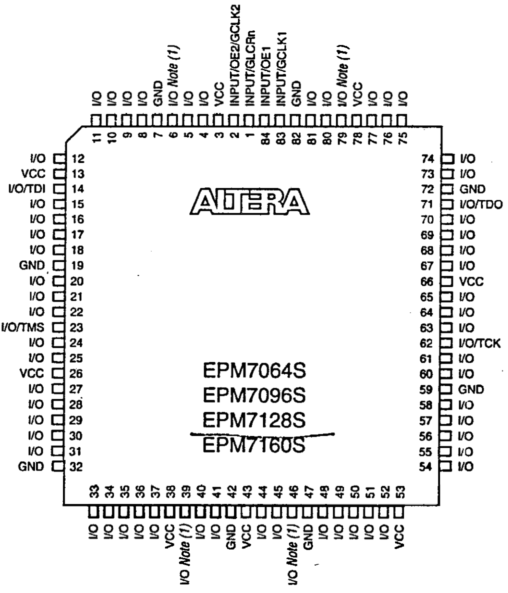
absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)		7 V
Input voltage		7 V
Off-state output voltage		5.5 V
Operating free-air temperature range:	SN54LS'	−55°C to 125°C
	SN74LS'	0°C to 70°C
Storage temperature range		−65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

Figure 19. 84-Pin Package Pin-Out Diagram

Package outline not drawn to scale. Pin functions in parentheses are for MAX 7000S devices only.



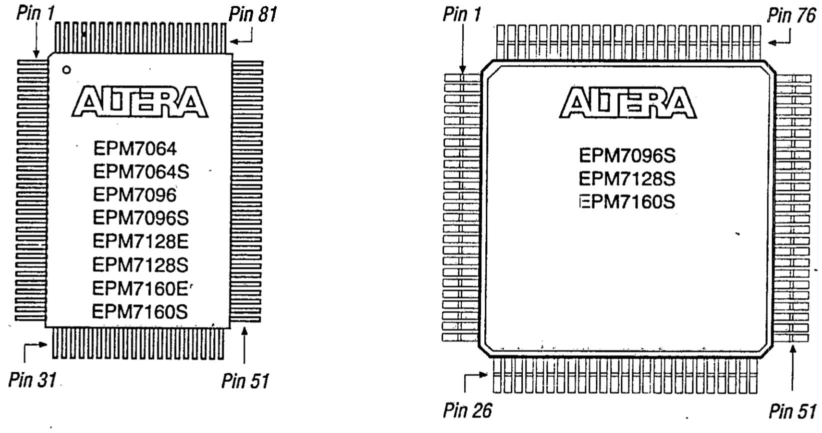
84-Pin J-Lead

Note:

(1) Pins 6, 39, 46, and 79 are no-connect (N.C.) pins on EPM7096, EPM7096S, EPM7160E, and EPM7160S devices.

Figure 20. 100-Pin Package Pin-Out Diagram

Package outline not drawn to scale.



100-Pin PQFP

100-Pin TQFP

```
                                                  d
              R  R  R  R     R  R  R        a     a     R  R  R     R  R  R
              E  E  E  E     E  E  E        d     t     E  E  E     E  E  E
              S  S  S  S     S  S  S  V     d     a     S  S  S     S  S  S
              E  E  E  E     E  E  E  C     s     s     E  E  E  V  E  E  E
              R  R  R  R     R  R  R  C     t     t     R  R  R  C  R  R  R
              V  V  V  V  G  V  V  V  I  G  r  G  r  G  V  V  V  C  V  V  V
              E  E  E  E  N  E  E  E  N  N  b  N  b  N  E  E  E  I  E  E  E
              D  D  D  D  D  D  D  D  T  D  /  D  /  D  D  D  D  O  D  D  D
            ------------------------------------------------------------------
          /  11 10  9  8  7  6  5  4  3  2  1 84 83 82 81 80 79 78 77 76 75    |
RESERVED | 12                                                               74 | RESERVED
   VCCIO | 13                                                               73 | RESERVED
RESERVED | 14                                                               72 | GND
       J | 15                                                               71 | RESERVED
     sts | 16                                                               70 | r/c
  write/ | 17                                                               69 | L3732
     ai0 | 18                                                               68 | L3731
     GND | 19                                                               67 | en/
     ai1 | 20                                                               66 | VCCIO
     ai2 | 21                                                               65 | mux2/
RESERVED | 22                       EPM7128ELC84-15                         64 | mux1/
     ai3 | 23                                                               63 | RESERVED
     ai4 | 24                                                               62 | A7
     ai5 | 25                                                               61 | A6
   VCCIO | 26                                                               60 | A5
     ai6 | 27                                                               59 | GND
     ai7 | 28                                                               58 | A4
RESERVED | 29                                                               57 | A3
RESERVED | 30                                                               56 | A2
RESERVED | 31                                                               55 | A1
     GND | 32                                                               54 | A0
         |_  33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53   _|
            ------------------------------------------------------------------
              R  R  R  R  R  V  R  R  R  G  V  R  R  R  G  R  R  R  w  R  V
              E  E  E  E  E  C  E  E  E  N  C  E  E  E  N  E  E  E  a  E  C
              S  S  S  S  S  C  S  S  S  D  C  S  S  S  D  S  S  S  i  S  C
              E  E  E  E  E  I  E  E  E     I  E  E  E     E  E  E  t  E  I
```