Microcontroller Based P. I. D. Controller

PROJECT WORK

Submitted By

Tom .P .A
Riju .K .P
Murali .V
Dilip Nair .R

Under the Guidance of

Mrs. G. Shanthi, M.E.,

IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF
BACHELOR OF ENGINEERING
IN ELECTRONICS AND COMMUNICATION ENGINEERING

1995

Department of Electronics and Communication Engineering
Kumaraguru College of Technology
Coimbatore - 641 006

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Kumaraguru College of Technology

COIMBATORE - 641 006.

CERTIFICATE

This is to certify that the Project Report entitled

MICROCONTROLLER BASED PID CONTROLLER

has been submitted by

Mr. TOM·P·A ; RIJU·K·P ; MURALI·V ; DILIP NAIR·R

in partial fulfilment for the award of the degree of

Bachelor of Engineering
in Electronics and Communication Engineering

Branch of the Bharathiar University, Coimbatore
during the academic year 1994-95

G Shanthi
Guide

Head of the Department

Certified that the candidate with University Register No. 9127D0 191/179/176/167

was examined in project work Viva-Voce by us on April 6, 1995

Internal Examiner

External Examiner

Dedicated

to our

Beloved Parents

who made

everything possible

Acknowledgement...

ACKNOWLEDGEMENT

We are solemnly indebted to the Department of Electronics And Communication Engineering. We express our heartfelt gratitude to our Principal **Dr. S. SUBRAMANIAM** M.Sc.,(ENGG.),Ph.D.,S.M.I.E.E.E.,for the ample facilities provided to us for successfully carrying out our project work.

It is indeed a matter of immense pleasure to thank our beloved head of the Department, **Prof. MUTHURAMAN RAMASAMY** M.E.,M.I.E.E.E.,M.I.S.T.E.,M.I.E,C.(Engg) who extended us kind cooperation and continued support during the course of our project work.

A special thanks goes to our guide **Mrs. G.SHANTHI M.E.** for helping us by providing constructive suggestions.

We would like to express our lasting gratitude to our advisor and to all the staff members and lab technicians of our department who helped us directly or indirectly in the completion of our project.

(Authors)

Synopsis...

SYNOPSIS

Our project which is the Microcontroller based PID controller is purely an industrial oriented project. The heart of the system is the microcontroller our project aims at controlling the output of a heater which proceeds the necessary temperature for a particular process. We have designed a closed loop system for this purpose.

The temperature of the heater is measured using a sensor and converted into voltage using a transducer. The output of the transducer is fed to the kernel of the system which is the microcontroller. Before being fed, It is converted into its digital equivalent using an A/D converter. The required value of the temperature which is the set value is also fed into the mcu using keyboard. The mcu compares the output of the ADC and the set value and provides an error signal. Using PID routine, the mcu calculates the control voltage. This is them converted to its digital equivalent using a DAC routine. This signal is converted by a transducer and the output of the transducer controls the heater output. The A/D conversion and D/A conversion are also controlled by the mcu.

A seven segment display has been interfaced with the system to display the set value. The controller chip 8279 has been used for both keyboard and display interface.

Contents...

CONTENTS

Introduction...

CHAPTER 1

INTRODUCTION

1.1 General

During the past twenty years,the automated process control industry has matured significantly due to the introduction of microprocessors, as an element of control system. Though the use of microprocessors was limited to a supervisory status at the beginning, today systems use off the shelf digital hardware and software to perform all the control applications.

Our project aims at controlling various process parameters using a micro controller based on the PID algorithm with greater accuracy ,reliability and flexibility.

1.2 P.I.D. Controller

In industries many process parameters have to be controlled either manually or automatically. Manual control will not be accurate and the quality of end product will be less and thus automatic control is preferred.Open loop control system is not stable and the acccuracy of the output cannot be guaranteed. But since the error signal is used in closed loop system the accuracy and the stability is high. Among the various feedback or closed loop control system the ON OFF control, the three mode system i.e. PID control system is the most advantageous .

In the feedback control system using the PID algorithm i.e.Proportional, Integral and Derivative control algorithm the difference between the set point and the measured system variable is found.The output signal will then be a function of the error signal.

For the proportional control the controller output is given by the equation

$$V_p = K * e$$

where V_p is the output signal

K is the controller gain

The proportional control action will take place only if an appreciable error is present

The integral control system is usually used along with the proportional control system. The controller output is given by the equation

$V_i = 1/K_1 \int e \, dt$

where V_i is the integral control signal and K_1 is the integral time constant.

The derivative control action is never used alone. It is always used in conjunction with proporitonal and integral action. The derivative output control signal is given by the equation.

$V_d = K_d \, de/dt$

where V_d is derivative output control signal and K_d is derivative time constant.

For the PID control action all the individual control actions are combined. That is the total control signal is given by the sum of the three control signals.

1.3 Microcontroller Based PID Controller

In our project PID control is done by means of software using a single chip microcontroller-80C31. Hence our project has all the inherent advantages of microcontrollers.

The data from the actual process is taken to the processor and compared with desired value and the compared output is used to drive the control voltage using the PID algorithm. This is then converted back to analog and given to the process control element.

Advantages :-

* This system is very flexible if the desired values or any gain constants are to be changed then it can be done in the software itself.There is no need to change the design unlike the one using electronic circuits.

* This system can be used to control any process parameter without changing the hardware. A large number of parameters can be controlled by using multiplexers.

* The hardware is simple, unlike the controller using various electronic circuits for implementing the PID algorithm.

* Since it is microcontroller based, the system will be very fast, accurate and reliable.

* The size of the product will be compact unlike the one using various electronic circuits.

* Future expansion is easy.

Microcontroller...

CHAPTER 2

Microcontroller

2.1 General

As the heart of the system revolves around the microcontroller and the PID algorithm is done using software,a general understanding of the architecture,their instruction sets and comparison betweem some microcontrollers are imperative, before system description. All these features are dealt with in this chapter.

2.2 Single Chip Microcomputer Architecture

A single chip microcomputer, of which microcontrollers are a subset, is a single integrated circuit that contains the five essential elements of a computer, i.e. input,output,memory,ALU,and control unit. The primary difference among the various models is the type of on chip program storage, the two options being no program ROM or 4K bytes EPROM (87C51). The amount of RAM is limited to 128 bytes,but this is adequate for a number of applications.

The on chip I/O is implemented as 32 programmable lines consisting of four ports of eight bits each.

A major architectural difference among microcontrollers and computers in general is the method of accessing I/O ports.The different approaches are memory mapped I/O, separate I/O and I/O processors. Memory mapped I/O means that all I/O is performed as if the I/O devices were memory addresses. Separate I/O has, in addition to a read/write control line an I/O memory control line. This method can use identical address decoders for both I/O and memory with the final determination which of these is accessed and determined by the status of the I/O /memory control line.

Another important feature is that most microcontrollers have versatile timers that can be configured in real time by writing approximate codes to control registers. These timers can be reprogrammed as to the duration that they will measure,the type of interrupt,if any that they will generate and whether they will reload and continue to time the next period automatically. They can also usually be reconfigured again in real time,to act as event counters or to count the duration between transitions on the inputs.

Some microcontrollers make no provision for external memory and are designed to operate autonomously. Others allow for the possibility of adding external memory. A disadvantage of this approach is that the I/O ports are lost and this may not be acceptable. Another need for external memory is during the development phase, in which it is inconvenient to continually erase and reprogram on chip EPROM program memory.

2.3 Comparison Of Various Microcontrollers

An illustration of a variety of microcontrollers available from two major manufacturers - MOTOROLA and INTEL are given below in the next page.

TABLE 1.0 Capabilities of Representative-8bit microcotrollers from Two Families.

Chip	RAM	ROM/EPROM EAROM/EEPROM	CLOCKS	I/O PORTS	A/D	TIMERS
M68HC11A0	256	---	0.476	4x8 1x6	4/8	9
M68HC11A1	256	512 EEPORM	0.476	4x8 1x6	4/8	9
M68HC11A2	---	2048 EEPORM	0.476	4x8 1x6	4/9	9
M68HC11A8	256	8K ROM,512 EEPROM	0.476	4x8 1x6	4/8	9
M68HC11E0	512	---	0.476	4x8 1x6	4/8	9
M68HC11E1	512	512EEPORM	0.476	4x8 1x6	4/8	9
M68HC11E1	256	2048 (EE)	0.476	4x8 1x6	4/8	9
M68HC11E9	512	12K (ROM), 512 (EE)	0.476	4x8 1x6	4/8	9

Chip	RAM	ROM/EPROM EAROM/EEPROM	CLOCKS	I/O PORTS	A/D	TIMERS
M68HC11D3	192	4096 (ROM)	0.476	4x8 1x6	8	9
M68HC11F1	1024	512(EEPORM)	0.476	4x8 1x6	8	9
Intel8021	64	1024 (ROM)	2.5	2x8 1x4	–	2
Intel 8022	64	2048 (ROM)	2.5	3x8	–	2
Intel 8035	64	----	2.5	3x8	–	2
Intel 8039	128	------	1.4	3x8	–	2
Intel 8041	64	1024 (ROM)	2.5	3x8	–	2
Intel 8048	64	1024 (ROM)	2.5	3x8	–	2
Intel 87498	64	2048 (ROM)	1.4	3x8	–	2
Intel 8748	64	1024 (EPROM)	2.5	3x8	–	2
Intel 8Y031	128	-----	1	4x8	–	2
Intel 8051	128	4096 (ROM)	1	4x8	–	2
Intel 8751	128	4096 (EPROM)	1	4x8	–	2

2.4 Instruction Set

The actual selection for a microcontroller is a complex interaction of the five fundamental issues.

i) Type of operation :-

The number and complexity of the operations selected

ii) Types of data :-

Which type of numeral,character, or logical data types are desired as operands.

iii) Instruction formats :-

The structure of the instruction including the number of opcodes,the number of addresses and whether it is a fixed or variable size.

iv) Registers :-

The amount of on chip directly addressable storage.

v) Addressing :

The mode of addressing that is allowed.

This chapter has given an overview on the microcontrollers and hence the choice of microcontrollers instead of opting for microprocessors which requires the use of additional peripheral devices like 8255 PPI etc is justified.

PID controller implementation...

CHAPTER 3

PID Controller Implementation.

3.1 GENERAL

In industries it may be required to control most of the manufacturing processes.For example in spinning mills the humidity and temperature have to be controlled. In cement industry the flow rate has to be controlled and in some industries the pressure has to be controlled. For all these control actions various feedback control systems are available. In this chapter a detailed understanding of the PID CONTROLLER which is a classical controller used for controlling all these parameters, is done.

3.2 Typical PID characteristics and related terminology.

The PID controller transfer function that is often sought in the construction of most commercial controllers is given by,

$$G_c(S)=K_c(1+1/T_iS+T_ds)$$

where K_c is controller gain

T_i is integral time constant and

T_d is derivative time constant.

The proportional action associated with K_c is called proportional band (PB) and is inversely related to the concept of gain (K_c).Doubling gain K_c requires that the proportional band be halved.

Controller output is usually fed to the actuator.With proportional action ,the controller output varies proportionally to the deviation from the set point.When the input to the controller gradually increases above the set value,the actuator goes on closing and vice versa.If for a 50% increase,the actuator completely closes and for a 50% decrease it completely opens,the PB is said to be 100%.Similarly if the actuator closes by 75% for a 50% increase and remains 25% closed for a 50% decrease,then the PB is 200% and so on.This is shown schematically in figure 3.1.If the PB becomes infinity,control action ceases.

Depending on the process to be controlled, simpler combinations of the ideal three actions are often used.

P+I control:

When a demand from the process is fully met by P-action alone, the I-action is unnecessary, but if a higher demand is made from the process , it is unlikely that the simple P-action will be able to meet the demand . When the P-action alone is used the output will not be large enough to meet this demand and a settlement occurs at any intermediate value leaving an offset error . If an I action is provided, then it supplies an ever increasing output,allowing the actuator to open more.The demand by the process can be met such that the deviation and the proportional output start falling towards the set point.The procedure will actually bring the offset to zero.With the proper choice of the P action and I action parameters it is posible to check the rise in the I unit output and steady the P unit output when the demand is fully met.Then the outputs from the P unit and I unit together give the required output.The integral time constant,is the interval to cause the same amount of output as that produced by proportional action due to a steady deviation from set point.

P+D control:-

Another mode of control is the P+D control.This mode is complementary to the I action.When a process runs with small disturbance wide PB and low I action are usually preferrred.Because of these settings,the action of the controller is too slow.If now,suddenly,large disturbances occur at wide intervals,this P-I action alone will not be sufficient and a D action may be introduced to challenge such immediate deviations and ensure immediate corrective action.As soon as the deviation becomes zero, the P-action will stop.Thus the derivative system detects a change in the increase or decrease of the deviation and provides an outpput in proportion to the speed of deviation.

Each of these system has in own advantages and disadvantages which will be dealt next.The choice of the controller depends on the transient response of the process.The ideal three mode controller is one whose output is the sum of all the three control actions.

In the PID controller K_c, T_i, T_d are all adjustable. From the equation it is clear that integral time (T_i)and derivative time (T_d) respectively attect integral and derivative parts only while proportional gain (K_c) affects all the parts of the control action.

When the process characteristics are known approximately the values at which $K_c, T_i$and T_d are to be adjusted are determinable.Final setting of these parameters will always be made by making a compromise between the steady state and dynamic performances.The adjustment of the parameters can be done in many ways.In one method the proportional gain is increased till the system begins to have stable oscillations putting off the other two control actions.If the oscillator period is T_o and system gain is K_{co}then the parameter values for the PID control action can be derived from the equation

$K_c = 0.6\ K_{co}$

$T_c = 0.5\ T_o$

$T_d = 0.125\ T_o$ as given by Ziegler and Nichols.

3.3 NEED FOR PID:

A proportional feed back system has a fast response but they are normally not used since corrections can not be made until an appreciable error has been detected.There are other disadvantages also. When the gain is too large oscillations are introduced. In the proportional feed back scheme the system is unable to maintain the control element at some point when the error signal is not there. Thus in the proportional feed back scheme there is always an offset error.

An integral feed back system will not produce an offset error or steady state error. The elimination of the offset is an important control objective. Thus the integral control term is widely used in conjunction with proportional control element.But above all this the integral control scheme has a very slow response. A combination of proportional and integral control schemes would make possible to achieve the faster response of the proportional controller with resetting capabilty of integral controller to give a system with no offset error.

An additional problem to the proportional integral controller is to avoid problem with severe overshooting or oscillation when we want a high gain system. Thus gain figure is also limited and thus in turn limits the rate at which the system responds to input change. Thus we are introducing the derivative feedback scheme also with the existing proportional integral scheme.

The derivative term included in the P-I scheme will provide an output which is a function of the rate of change in the error signal. It anticipates the further behaviour of the system and improves the dynamic response to the controlled variable by decreasing the process response time. The derivative control scheme is not used alone in the control system because if there is a constant error the output will be zero. The main disadvantage of a derivative control is that it tends to amplify noise which is picked up during the measurement itself.

Thus by combining the three proportional integral and differential control schemes we can get a fast response system, with no offset errors and overshoots .

3.4 SYSTEM DESCRIPTION:

A general idea of the entire system can be obtained from the block diagram. It mainly consists of

(i) Sensor and signal conditioning

(ii) A/D and D/A converters

(iii) Keyboard and Display interface

A general description about this system is given below.

Sensor and Signal conditioning :-

Sensor or a transducer is a device by which a physical or mechanical or optical quantity to be measured is directly transformed into an electrical voltage or current proportional to the input measured value. Different types of sensors are available for different types of measurements. For example various capacitor senses are used for displacement, velocity, pressure and force measurements. LVDT is used for pressure measurement etc.

For temperature measurement the various sensors used are the thermistors, resistance temperature detectors, thermocouples, integrated circuits etc. While RTD's give linear output and offer temperature range of -220°C to 55°C, thermocouples are used particularly for higher temperatures. Integrated circuits like AD590 are available with a linear current output for temperature measurement. It can sense a range of temperatures from -55°C to 150°C. Since this IC gives a linear current output proportional to the absolute temperature there is no necessity for special linearising or signal conditioning circuitry. However if the other sensors are used the output must be linearised with respect to the input temperature.

A/D and D/A converters :-

The input signal from the transducer which is an analog signal is digitized using the A/D converter. The ADC0809 with eight channel multiplexer is used for this purpose. It uses the successive approximation technique for conversion having a resolution of eight bits. The A/D output range is from 0 to

FF for an input of 0 to 5Volts. Any one of the channels can be selected by the use of three address lines. The conversion time is 100 μS. Thus the A/D converter which is highly accurate, a high speed device is well suited for control and measurement applications.

The digitized signal from the A/D is given to the processor for calculating the control voltage using the software. Once the control voltage is calculated, the digital output from the processor is fed to the D/A converter . This is also a highly accurate device with fast setting time and low power consumption. Thus the digital signal is again conveted back to the analog for supplying to the driver and actuator circuitry.

Driver/Actuator circuit :-

The ouput from the digital to analog converter is fed to the process actuator. So this analog signal has to be amplified before being fed to the controlling element. Depending on the process various types of amplifiers are used. Usually instrumentation amplifier is used for this purpose. Depending on the different measuring parameters different types of controlling circuits are used.

Keyboard and Display Interface :-

The desired values of the processes are input through the 7 x 3 standard matrix keyboard. It is interfaced to the microcontroller using 8279 keyboard interface. Keyboard entries are stored in the internal FIFO (first in first out) memory while an interrupt singal is generated with the keyboard entry.

The display section consists of nine seven segment displays interfaced to the microcontroller using the 8279 keyboard/display interface. The display segment has 16 x 8 R/W memory (RAM) which can be used for read/write information for display purposes. The display can be setup either in the right entry or left entry.

The 8279 keyboard and display interface has got four sections.

(i) Keyboard Section

(ii) Scan Section

(iii) Display Section

(iv) MPU Interface Section

(i) Keyboard Section :-

This section has 8 lines (RL0-RL7) that can be connected to 8 columns of a keyboard plus two additional lines: SHIFT and CNTL/STB (Control/strobe). The status of SHIFT key and the CONTROL key can be stored along with a key closure. The keyboard also includes 8x8 FIFO (First in First out)RAM. the FIFO RAM consists of 8 registers that can store 8 keyboard entries. The status logic keeps track of the number of entries and provides an IRQ(interrupt request) signal when the FIFO is not empty.

(ii) Scan Section :-

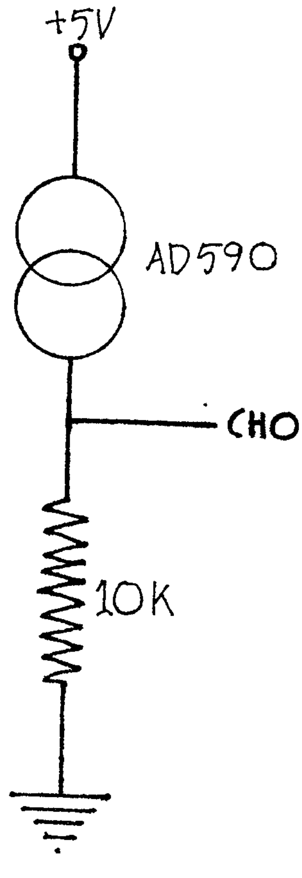
The scan section has a scan counter and 4 scan lines(SL0-SL3). These four scan lines can be decoded using 4 to 16 decoder to generate 16 lines for scanning. These lines can be connected to the rows of a matrix keyboard and the digit drives of a multiplexed display.

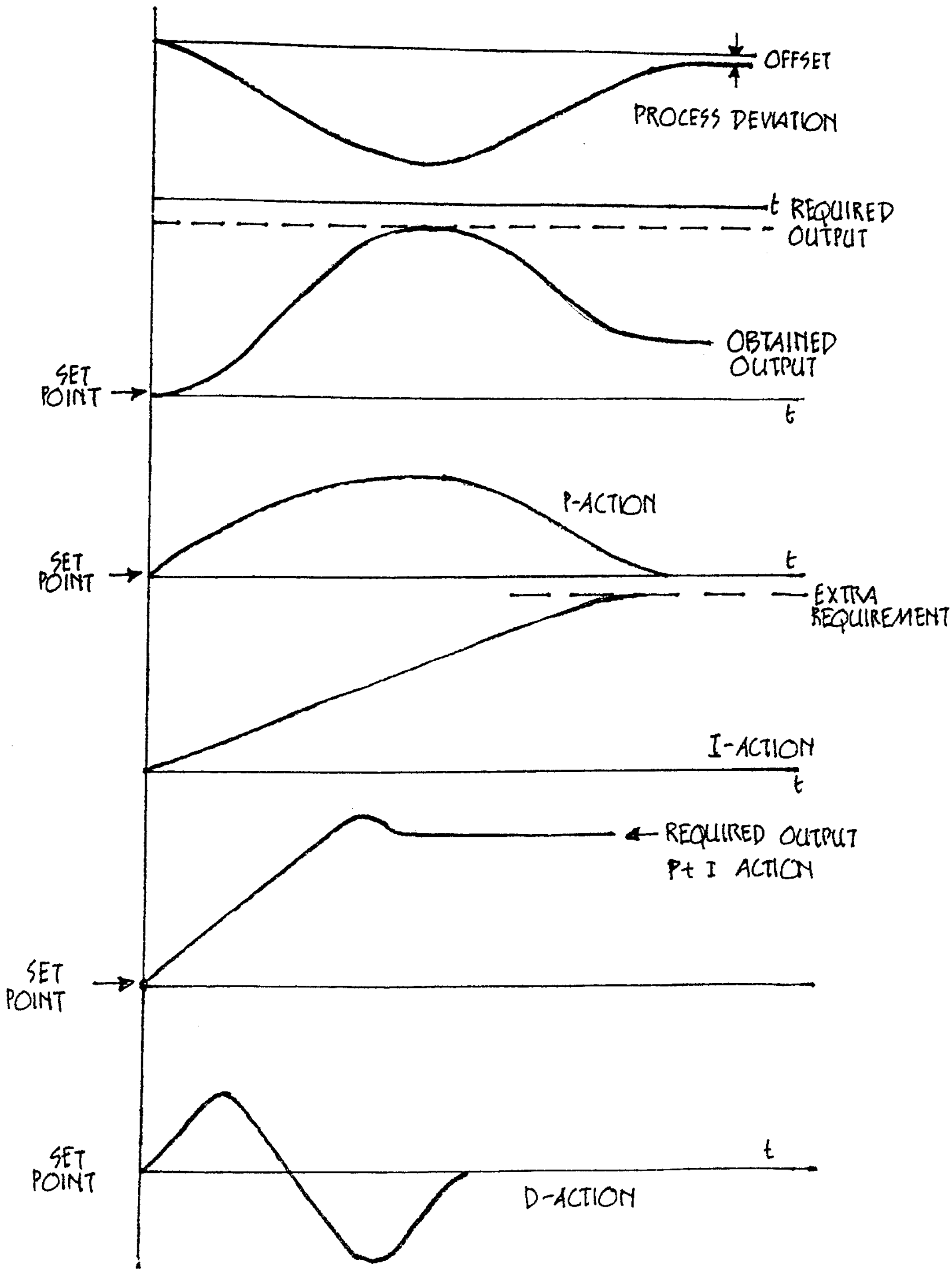
(iii) Display Section:

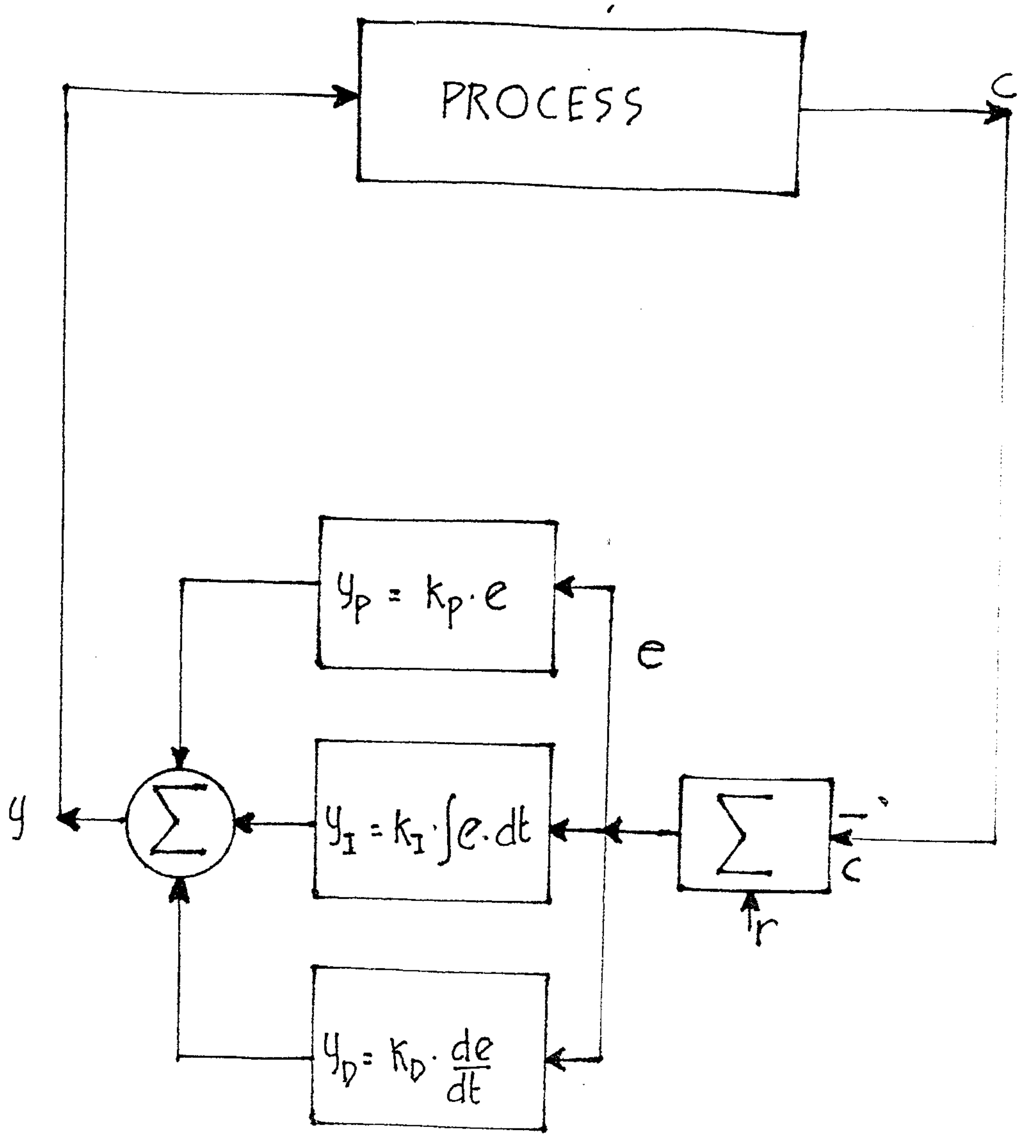
The display section has 8 output lines divided into two groups - A0 to A3 and B0 to B3. These lines can be used either as a group of 8 lines or as two groups of four in conjuction with the scan lines for a multiplexed display. The display can be blanked by using the BD line. This section includes 16X8 display RAM. The MPU can read from or write into any of these registers.

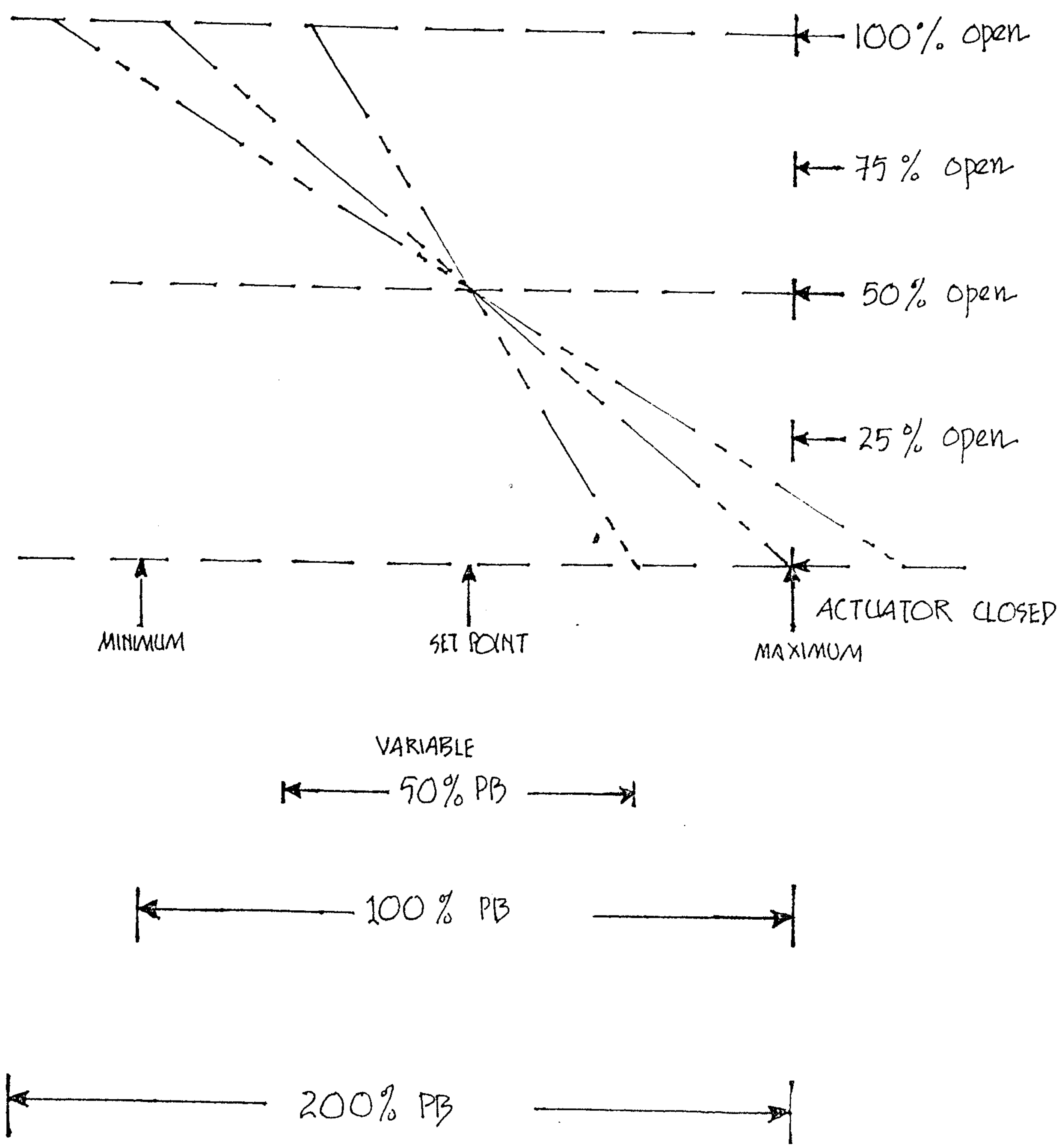
(iv) MPU Interface Section :-

This section includes 8 bi-directional data lines(DB0-DB7),one interrupt request line(IRQ) and 6 lines for interfacing,including the buffer address line(A0). When A11 is high,the signals are interpreted as control words are status. When A0 is low, the signals are interpreted as data.The IRQ line goes high whenever data entries are stored in the FIFO. This signal is used to interrupt the MPU to indicate the availability of data.


+5V
AD590
CHO
10K


OFFSET
PROCESS DEVIATION
t
REQUIRED OUTPUT
OBTAINED OUTPUT
SET POINT
t
P-ACTION
SET POINT
t
EXTRA REQUIREMENT
I-ACTION
t
REQUIRED OUTPUT
P+ I ACTION
SET POINT
SET POINT
t
D-ACTION


PROCESS
c
e
$y_P = K_P \cdot e$
$y_I = K_I \cdot \int e \cdot dt$
$y_D = K_D \cdot \frac{de}{dt}$
y
r
c



100% open
75 % open
50% open
25 % open
ACTUATOR CLOSED
MINIMUM
SET POINT
MAXIMUM
VARIABLE
50% PB
100 % PB
200% PB

Microcontroller 80C31...

CHAPTER 4

µC 80C31

4.1 Introduction :-

An overview about the microcontrollers was given in the chapter 2. In this chapter a more detailed understanding of 80C31 is done.

The 80C31 microcontroller is one of the members of the MCS-51 family of 8-bit microcontrollers all of which are based on the same architecture and uses the same powerful instructions. The 80C31 is fabricated on the CHMOS technology and is a ROM less version of the 80C51 and 87C51 microcontrollers.

The salient features of 80C31 microcontroller are

* 8 bit CPU optimized for control applications
* 128 Byte data RAM
* 64K external program memory space and
* 64K external data memory space
* 32 Programmable I/O lines

* Two 16 bit timer/Counters
* 5 interrupt sources
* Draws less current and can be operated in two reduced power modes.

Because of these salient features the highly efficient 80C31 microcontroller was selected as the heart of the system.

4.2 Architecture :-

The architecture of the 80C31 microcontroller is same as that of the MCS-51 family of microcontrollers. Please refer to the appendix for the block diagram.

4.2.1 Memory Organisation :-

The 80C31 has separate address spaces for program memory and data memory. The logical separation of program and data memory allows the data memory to be accessed by 8-bit addresses,which can be more quickly stored and manipulated by an 8-bit CPU. Also 16-bit data memory addresses can be generated through the DPTR register.

Program Memory :-

Program memory can only be read,not written to. There can be up to 64Kb of program memory in the ROM and EPROM versions; the lowest order bytes are provided on the chip. In the ROM less 80C31 version all program memory is external.The selection whether the lowest 4K (or 8K or 16K) bytes of program memory be the onchip ROM or in an external ROM is made by strapping the EA pin to either V_{CC} or V_{SS}. In the case of 4K ROM devices if EA pin is strapped to V_{CC} then program fetches to addresses 0000H through 0FFFH are directed to the internal ROM and addresses from 1000H through FFFFH are directed to external ROM. If EA pin is strapped to V_{SS} then all fetches are directed to the external ROM. In 80C31 this pin is externally strapped to V_{SS} to enable it to execute properly.

For accessing the external program memory two of the microcontroller ports,port0 and port2 are used.Port0 carries the lower order address multiplexed with data and port2 carries the higher order address. As soon as the address is on the bus the PSEN is enabled and code byte is read in to the microcontroller.

Moreover each interuppt of 80C31 is given a location in the program memory.As soon as the interuppt is made the CPU jumps to that particular location and commences execution of the service subroutine.

Data Memory :-

In the case of 80C31 microcontroller chip upto 64KB of external RAM can be addressed in the external data memory space. The CPU generates read and write signals,RD and WR as needed during external data memory addresses. The exteranal data memory are often used in conjuncttion with one or more other I/O lines to page the RAM. The lower 128 bytes of RAM are present in all MCS -51 devices.

4.2.2 Registers :-

The lowest 32 bytes of the internal RAM are grouped into four banks of 8 registers. The program instructions call out these registers as R_0 through R_7. Two bits in the program status word are used to select the register banks.

The next sixteen bytes above the register banks form a block of bit addressable memory space. The 128 bits in this area can be directly addressed by instructions. The bit addresses in this area are 00H through 7FH.

Apart from the register banks there are many special function registers (SFR).The SFR's includes port latches,timer,peripheral controls etc.These registers can only be addressed by direct addressing. Accumulator register,B register,Program status word register,Stack point register,Data pointer register are examples of special functions register.The B register is used during multiply and divide operations.For other instructions it can be treated as another scratch pad register.The PSW register contains program status informations as shown below.

MSB LSB

CY AC FO RS1 RS0 OV-P

SYMBOL	POSTION	NAME AND SIGNIFICANCE
CY	PSW.7	Carry flag
AC	PSW.6	Auxiliary carry flag (for BCD operations)
FO	PSW.5	Flag 0
RS1	PSW.4	Register bank select control bits 0 set
RS0	PSW.3	cleared by software to determine working register bank.
OV	PSW.2	overflow flag
-	PSW.1	user definable flag
P	PSW.0	parity flag

The stack pointer register which is 8-bit wide may reside anywhere in on-chip RAM.It is incremented before data is stored during PUSH and CALL execution.The stack pointer is initialised to 07H after a reset. The data pointer consists of a high byte (DPH) and a low byte (DPL).Its intended function is to hold a 16-bit address. It can also be manipulated as two independent 8-bit registers.

Apart from these registers,there are four port latches and other timer,capture and control registers under the special function registers.

4.2.3 Input/Output

There are totally four input/output bidirectional ports present in 80C31 microcontroller. All the ports have 8 I/O lines which make a total of 32 programmable I/O lines. Each port consist of a latch,an output driver and an input buffer. The output drivers of port0 and port2 and the input buffers of port0 are used to access external memory. The port0 outputs the low byte of the external memory address,time multiplexed with the byte being written or read.The port2 carries the higher order address of the external memory.

All the port 3 pins and two port1 pins are having functions other than as port pins. They are as listed below:

PORT PIN	ALTERNATE FUNCTION
P3.0	RXD (serial input)
P3.1	TXD(serial output port)

P3.2	INTO (external interrupt)
P3.3	INT1 (external interrupt)
P3.4	TO (timer/counter 0)
P3.5	T1 (timer/counter 2)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

The alternate functions can only be activated if the corressponding bit latch in the port SFR contains a 1.

4.2.4. Interrupts :

The 80C31 provides 5 interrupts sources two external interrupts , two timer interrupts & the serial port interrupt. What follows is an over view fo the interrupt structure of 80C31.

Each of the interrupts can be enabled or disabled by setting or clearing a bit in the special register named IE register. This register also contains a bit which can be cleared to disable all interrupts at once. Each interrupt can also be individually programmed to one of two priority levels by setting or clearing a bit in the SFR named interrupt priority. Thus a low priority interrupt can be interrupted by a high priority interrupt but not the other way. If interrupt requests of the same priority levels are received then an internal polling sequence determines which request is serviced.

The external sources can be programmed to be level activated or transition activated by setting or clearing the bit IT1 or IT0 in register TCON. Since external interrupts pins are sampled once each machine cycle, an input high or low should hold for atleast 12 oscillator periods to ensure sampling. If the external interrupt is transition activated the external source has to hold the request pin high for atleast one cycle and then hold it low for atleast one cycle to ensure that the transition is seen so that interrupt request flag IE_X will be set. IE_X is automatically cleared by CPU when the service routine is called. If the external is level activated, the external source has to hold the request until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, otherwise another interrupt will generated.

4.2.5 Addressing Modes :

The various modes used in 80C31 are discussed below.

Direct Addressing:

In direct addressing the operand is specified by an 8 bit address field in the instruction. Only internal data RAM and SFR's can be directy addressed.

Indirect Addressing:

In indirect addressing the instruction specifies the register which contains the address of the operand.Both internal and external RAM can be indirectly addressed.

The address register can be R_0 or R_1 of the selected register bank or the stack pointer for 8-bit addresses.Four 16-bit addresses the address register can only be the 16bit data pointer register DPTR.

Register Instructions :

One of the eight registers from a register bank can be selected using a three bit register specification within the opcode of the instructions. This eliminates the need for an address byte. The register banks is selected by means of two bank select bits in the program status word.

Register Specific Instructions :

These are instructions that are specific to certain registers like accumulator or data pointer and thus no address byte is needed to point it.

Immediate Addressing :

In this type the value of the constant can be written along with the opcode in program memory.

Index Addressing :

Only program memory can be accessed with indexed address and it can only be read. This type of addressing is intended for reading look up tables in program memory. A 16 bit base register acts as the pointer and the accumulator is used for loading the table entry number. The address of the table entry in program memory is formed by adding the accumulator data to the base point.

4.2.6 Timer/Counter :

In the timer functions, the register is incremented in every machine cycle. Thus one can think of it as counting machine cycles. Since a machine cycle consists of 12 oscilator periods,the count rate is 1/12 of the oscillator frequency.

In the counter function a register is incremented in response to a 1 or 0 transition at the corresponding external input pin t0 or t1. The timer 0 and timer 1 have four operating modes to select.

4.2.7: Timer/Counter mode control (TMOD) register :

MSB							LSB
GATE	C/T	M1	MO	GATE	C/T	M1	M0

GATE - Gating control when set. Timer/Counter 'x' is enabled only while TNTX pin is high and 'IRX' control pin is set. When cleared Timer is enabled whenever 'TRX' control bit is set.

C/T - Timer/Counter selector clear for tmer operation (input from internal system clock).Set for counter operation.

M1	M0	Operation Mode
0	0	8031 Timer TLX serves as 5 bit pre-scaler.
0	1	16 - bit timer/counter. THX and TLX are cascaded. There is no pre-scaler.
1	0	8 bit auto-reload timer/ counter THX holds a value which is to be reloaded in to TLX each time it overflows.
1	1	(Timer 0) TLO is an 8 bit timer/counter controlled by

		standard timer 0 control bits. THO is an 8 bit timer only, controlled by timer 1 control bits.
1	1	(Timer 1) Timer/Counter 1 stopped.

MODE 0 :

Putting either timer into mode 0 makes it look like a timer, which is an 8 bit counter with a divide by 32 pre-scaler.

In this mode the timer register is configured as a 32 bit register. As the counter rolls over, from all 1's to all 0's it sets the timer interrupt flag TF1. The counter input is enabled to the timer TR1 = 1 and either GATE = 0 or INT1 = 1. Setting GATE = 1 allows the lower to be controlled by external input INT1 to facilitate pulse width measurements. TR1 is a control bit in the special function register TCON GATE is in TMOD.

The 13 bit register consists of all 8 bits of TH1 as the lower 5 bits TL1. The upper 3 bits of TL1 are interminate and should be ignored. Setting the run flag TR1 does not clear the register.

Mode 0 operation is the same for timer 0 as timer 1. Substitute TRO , TFO and INTO for the corresponding timer 1 signals in figure. There are two different gate bits , one for timer 1(TMOD.7) and the other for timer 0(TMOD.3).

MODE 1:

Mode 1 is the same as mode 0 except that the timer register is being run with all 16 bits.

MODE 2:

The mode 2 configuration , the timer register is an 8 bit (TF1) with automatic reload.

MODE 3:

Timer 1 in mode 1 simply holds its count.The effect is the same as setting TR1 = 0.

4.2.8 STANDARD SERIAL INTERFACE:

MSB							LSB
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

SYMBOL	POSITION	NAME AND SIGNIFICANCE
TF1	TCON.7	Timer 1 overflow flag
TR1	TCON.6	Timer 1 run control bit set/cleared by software to turn Timer/Counter ON/OFF.
TF0	TCON.5	Timer 0 overflow flag set by hardware on timer/counter overflow.
TR0	TCON.4	Timer 0 run control bit .
IR1	TCON.3	Interrupt 1 edge flag set by hardware when external interrupt edge detected cleared when interrupt processed.
IT1	TCON.2	Interrupt 1 type control bit/cleared by software to specify falling edge/low

		level triggered external interrupts.
IE0	TCON.1	Interrupt 0 edge flag set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT0	TCON.0	Interrupt 0 type control bit set/cleared by software to specify falling edge/low level triggered external interrupts.

4.2.9 Instruction set:

The instruction set of MCS-51 family includes 111 instructions ,49 of which are single bytes,45 two bytes and 17 three bytes instruction.These instructions are divided into four functional groups .

* Data transfer
* Arithmetic
* Logic
* Control transfer

Data transfer:

These operations are further divided into three classes.

* General purpose
* Accumulator specific
* Address object

None of these operations affect the PSW flag settings except POP or MOV directly to the PSW

The general purpose transfers include instructions like MOV , PUSH and POP which transfer a byte from the source operand to the destination operand.The accumulator specific transfers include instructions like XCH,XCHD,MOVX,MOVC,etc. While the first two instructions perform exchange operations between the accumulator and source operand the last two performs byte movement between memory and accumulator. The address object transfer instruction like MOV DPTR #data loads 16 bits of immediate data into a pair of destination registers DPH and DPL.

Arithmetic instructions:

Basically four mathematical operations can be done using 80C31 for 8 bit unsigned operands. Addition and subtraction can be done on both unsigned and signed binary integers using the overflow flag.

The addition instructions include increment (INC) source operand, ADD which adds accumulator to the source operand, ADD C (add with carry) and DA (decimal-add-adjust for BCD addition).The subtraction operation include SUBB (subtract with borrow) and DEC (decrement from source operand).80C31 instruction can also perform the multiplication of A register by B register with lower byte of the result in A and higher byte in B.

The division instruction DIV performs the unsigned division of the A register by the B register and returns the integer quotient to A register and returns fractional remainder to B register.

Logic instructions:

The 80C31 performs basic logic operations on both bit and byte operands. The single operand operations include CLR which sets A to 0 , SETB which sets any directly addressible bit to 1 and CPL which is used to complement the contents of the A register without affecting flags . Apart from these instructions ther are five rotate instructions that can be performed on A. Two operand operations include ANL which performs a bitwise AND operations , ORL which performs a bitwise logical OR and XRL, which performs a bitwise logical XOR of two source operands.

Control transfer instructions:

There are three classes of control transfer operations, unconditional calls ,returns and jumps ,conditional jumps and interrupts . All these operations cause the program execution to continue at a non sequential location in program memory .

The unconditional calls , returns and jumps,transfer control from the current value of the program counter to the target address. ACALL and LCALL push the address of the next instruction into the stack and then transfers control to the target address. LCALL is a three byte instruction and ACALL is a two byte instruction . RET transfers control to the return address and decrements stack by two locations . AJMP, LJMP and SJMP transfers control to the target operand.SJMP is for transfering control with in 256 bytes centered about the starting address of the next instruction. The instructions are JZ, JNZ, JC, JNC, JB, JNB,etc. which transfers the control to the address specified when the condition is satisfied . The RETI transfer control as does RET but additionally enables interrupts of the current priorty level.

Hardware Details...

CHAPTER 5

HARDWARE DETAILS

The 8031 microcontroller controls the entire process of PID control.We have used an ADC chip 0809 for converting the measured value from the transducer into its corresponding digital value.A DAC chip 0800 is used to convert the digital signal into analog signal which is then converted by a transducer and is used to control the process.

The transducer used is IC AD590 which will give 1 micro ampere per kelvin linear output. Over the temperature range of this transducer,the generated current will vary from 218 micro ampere to 423 micro amperes.These numbers can be obtained easily by noticing that -55°C to +150°C corresponds to 218 kelvin to 423 kelvin.

A 10Kohm resistance will force the V_{IN} of the ADC to vary between approximately 2.18 volts to 4.23 volts.This corresponds to approximately half the full scale range of the ADC which is 5 volts.If the $V_{REF}(-)$ input of the ADC is set to 2.18 volts then the ADC will output all zeroes when the input temperature is approxmately -55°C.

Port 0(P0) of microcontroller 8031 is used as data port and Port0 and Port2 together are used for addressing the external chips.The higher order byte of the address comes out of Port2 while the lower order byte is given out on Port0 pins. Port1 pins are bit addressable and are used for control of the ADC and DAC chips.Three pins of Port1,P1.3,P1.5,P1.6 are connected to the address lines of the A/D converter.Addresses are generated using software.The clock input to the ADC is given from a monostable multivibrator 4047.P1.4 is provided to the ALE and SOC pins of ADC chips.Using software a pulse can be generated at the output of this pin by setting and clearing this pin alternatively.

The controller is put in loop to check for end of conversion signal.Once it recognizes this signal,it gives out a RD signal.The RD signal along with the output of the decoder provides the output enable pulse to the ADC and the digital data will be available at the output.

The digital output which is the digital equivalent of the control voltage V_c is provided by the microcontroller. The 74373 latch is used for latching the output data.The clock pulse of 74373 is generated using the address for DAC and the WR pulse.The output DA0 to DA7 of the latch is coupled to the inputs D01 to D08 of DAC using optocouplers. The advantage of using optocouplers is that it provides isolation between digital ground and analog ground.So perfect noise isolation will be present.

The analog output is available at pin number 4 of DAC. Reference voltage is generated at pin 14 using voltage supply and zener diode.The analog output V_o is applied to a voltage follower circuit which is using IC-741 and this acts as a buffer.The output of the buffer is provide to transducer whose output controlls the process whose parameter has to be controlled.

All the programs are stored in EPROM.The address for EPROM comes out on P0 and P2 lines.The lower order byte is latched using a latch(74373).The clock for the chip is connected to the ALE of the microcontroller 8031.Program Store Enable signal (PSEN) should be held low when EPROM is accessed.The output data bus P00 to P07 are connected to the data bus of microcontroller 8031.

The keyboard/display controller chip 8279 is used.The data bus of 80C31 is connected to the data lines of 8279. 8279 has got 16 bytes of display RAM. The data to be displayed is brought to the display RAM when the WR signal is enabled.The clock input to the 8279 is given from the ALE and the address buffer signal is given from the P20 of 80C31.Whenever the chip select is given the scan lines will generate a code which will select one display and the code for the data in the RAM is output to the output port.From the output port,it is given to the 7 segments of the display.The return lines and the 4 scan lines are used to scan the 7*3 matrix keyboard.If any key is pressed the address of the switch in the matrix plus the status of shift and control are transferred to the FIFO.Then the FIFO status can be read by an RD with CS low and A0 high.The status logic also provides an IRQ signal when the FIFO is not empty when the key is prcoed 8279 will give an interrupt request which is high all time . This is inverted using a monostable multivibrator and given to INT1 of the 80C31.Then RD signal is given by the CPU the data is read by the CPU.This can lines are given to a 4x16 Demultiplexer chip 4515. This will select one of the displays using transistor as switches. The output of the 4515 are 16 mutually exclusive low outputs. The transistor are used for driving the displays.

Software Details...

```
; SOFTWARE FOR PID CONTROLLER
;---------------------------

; INITIALISATION

CTRL_8279        EQU     2100H

DATA_8279        EQU     2000H

ADC_0808         EQU     6000H

DAC_08           EQU     4000H

TEMP_SET         EQU     30H

TEMP_READ        EQU     TEMP_SET+1
KEYB             EQU     40H
FL_KEYPRESSED    EQU     20H


VP               EQU     KEYB+1
E                EQU     VP+1
Vn               EQU     E+1
V[n-1]           EQU     Vn+1
K1               EQU     V[n-1]+1
K                EQU     K1+1
TS               EQU     K+1
T1               EQU     TS+1
VD               EQU     T1+1
VI               EQU     VD+1
VTOT             EQU     VI+1
E[n-1]           EQU     VTOT+1
X                EQU     E[n-1] +1
Y                EQU     X+1

          ORG     0000
          LJMP    START
          ORG     0003H
          RETI
          ORG     000BH
          RETI
          ORG     0013H

          JMP     ISR_8279
          ORG     001BH
          RETI
          ORG     23H
          RETI
```

```
ISR_8279:
        PUSH    DPL
        PUSH    DPH
        PUSH    ACC
        MOV     DPTR,#CTRL_8279
        MOVX    A,@DPTR
        ANL     A,#07H
        JZ      END_ISR_8279
        MOV     DPTR,#DATA_8279
        MOVX    A,@DPTR

        CALL    TRANS_KEY_CODE
        MOV     KEYB,A
        SETB    FL_KEYPRESSED

END_ISR_8279:
        RETI
```

Main program for PID algorithm

```
START:  CLR     EA      ;       Disable all the interrupts.
        CALL    INT_DEVICES

DEE1:
        JNB     FL_KEYPRESSED,DEE1


        CALL    CONVERSION

        CALL    FND_ERR

        CALL    CAL_PROP

        CALL    CAL_DERV

        CALL    CAL_INTEG

        CALL    FND_TOT

        CALL    OUT_DAC

        CLR     FL_KEYPRESSED

        JMP     DEE1
```

Data accusition from ADC

```
CONVERSION:
        MOV     DPTR,#ADC_0808
        NOP
        NOP
        MOV     A,@DPTR
        MOV     V1,A

        CALL    BIN2BCD
        CALL    DISP

        RET
```

Calculation of error voltage

```
FND_ERR:
        MOV     A,KEYB
        MOV     B,V1
        SUBB    A,B
        MOV     E,A
        RET
```

Calculation of proportional voltage

```
CAL_PROP:
        MOV     A,E
        MOV     B,K
        MUL     AB
        MOV     VP,A

        RET
```

Calculation of integral voltage

```
CAL_INTEG:
        MOV     A,TS
        MOV     B,K
        MUL     AB
        MOV     B,TI
        DIV     AB
        MOV     B,E
        MUL     AB
        MOV     B,V[n-1]
        ADD     A,B
        MOV     VI,A
        RET
```

Calculation of derivative voltage

```
CAL_DERV:
        MOV     A,TD
        MOV     B,K
        MUL     AB
        MOV     B,TS
        DIV     AB
        MOV     KD,A
        MOV     A,En
        MOV     B,E[n-1]
        SUBB    A,B
        MOV     B,KD
        MUL     AB
        MOV     VD,A
        RET
```

Calculation of total control voltage

```
CALL TOT:
        MOV     A,VP
        MOV     B,VD
        ADDC    A,B
        MOV     B,VI
        ADDC    A,B
        MOV     VTOT,A
        RET


INIT_DEVICES:
        MOV     P3,#0FFH
        MOV     P1,#00H
        SETB    EX1
        SETB    IT1
        MOV     DPTR,#CTRL_8279
        MOV     A,#00
        MOVX    @DPTRA,A
        MOV     A,#90H
        MOVX     @DPTR,A
        MOV     A,#50H
        MOVX    @DPTR,A
        MOV     A,#60H
        MOVX    @DPTR,A
        MOV     A,OOFH
        MOVX    @DPTR,A
        MOV     DPTR,# DAC_08
        MOV     A,#0FFH
        MOVX    @DPTR,A
        MOV     P1,#00H
        CLR     A
        SETB    P1.4
        CLR     P1.4
        RET
```

```
DISP_TBL:
        DB      0FCH
        DB      060H
        DB      0DAH
        DB      0F2H
        DB      066H
        DB      0B6H
        DB      0BEH
        DB      0FEH
        DB      0E6H

KEY_TBL:
        DB      0D1H,0D2H,0CAH,0C2H,0D3H,0CBH,0C3H ;
        DB      0D0H,0CBH,0C0H,0C1H,0C9H,0E2H,0E3H ;
        DB      0DAH,0DBH,0D8H,0D9H;

Output control voltage to DAC

OUT_DAC:
        MOV     DPTR,#DAC_08
        MOV     A,VTOT
        MOVX    @DPTR,A
        RET

TRANS_KEY_CODE:
        MOV     B,A
        MOV     DPTR,#KEY_TBL
        MOV     R7,#NO_OF_KEYS
        MOV     R0,#00H
        CLR     A
        MOVC    A,@A+DPTR
        XRL     A,B
        JZ      CODE_FOUND
        INC     R0
        INC     DPTR
        MOV     A,#0FFH
        SJMP    END_TKE
CODE_FOUND:
        MOV     A,R0
END_TKE:
        RET
```

```
; BIN2BCD32 converts a four byte binary number into a eight
digit packed BCD
; string.
;

BIN2BCD:
        PUSH    7
        MOV     Y,#0
        MOV     Y+1,#0
        MOV     Y+2,#0
        MOV     Y+3,#0
        MOV     R7,#32
BIN2BCD20:
        CLR     C
        MOV     A,X+3
        RLC     A
        MOV     X+3,A
        MOV     A,X+2
        RLC     A
        MOV     X+2,A
        MOV     A,X+1
        RLC     A
        MOV     X+1,A
        MOV     A,X
        RLC     A
        MOV     X,A
        MOV     A,Y+3
        ADDC    A,Y+3
        DA      A
        MOV     Y+3,A
        MOV     A,Y+2
        ADDC    A,Y+2
        DA      A
        MOV     Y+2,A
        MOV     A,Y+1
        ADDC    A,Y+1
        DA      A
        MOV     Y+1,A
        MOV     A,Y
        ADDC    A,Y
        DA      A
        MOV     Y,A
        DBNZ    R7,BIN2BCD20
        MOV     X,Y
        MOV     X+1,Y+1
        MOV     X+2,Y+2
        MOV     X+3,Y+3
        POP     7
        RET
```

```
DISP        :
        PUSH    DPL
        PUSH    DPH
        PUSH    0
        PUSH    6
        PUSH    7
        PUSH    B
        PUSH    A
        PUSH    PSW
        MOV     DPTR, #CTRL_8279          ;Initialize Display RAM
        MOV     A,#90H
        MOVX    @DPTR,A
        MOV     DPTR,#DATA _8279
        MOV      R0,#DISP
        MOV     R6,#01H
        MOV     R7,#06H
DPO:
        MOV     A,@R0
        MOV     B,A
        MOV     A,R6
        CJNE    A,DP_POSITION,DP1
         MOV    A,B
        CLR     ACC.7
        MOV     B,A
DP1:
        INC     R6
        INC     RO
        MOV     A,B
        MOVX    @DPTR,A
        DJNZ    R7,DPO
        POP     PSW
        POP     ACC
        POP     B
        POP     7
        POP     6
        POP     0
        POP     DPH
        POP     DPL
        RET
```

Conclusion...

CONCLUSION

We have successfully designed a system which controls the temperature of a process. Our project has been designed on a single printed circuit board which makes our system compact. Our project has been found to work satisfactorily. All the I/O devices like ADC, DAC and controller chip 8279 has been interfaced successfully to the microcontroller. The 7 segment LEDS displays the set value continuously. The system has been checked for different set value and each time it was found to work satisfactorily. The printed circuit board along with the keyboard has been enclosed in a cabinet.

SUGGESTIONS:

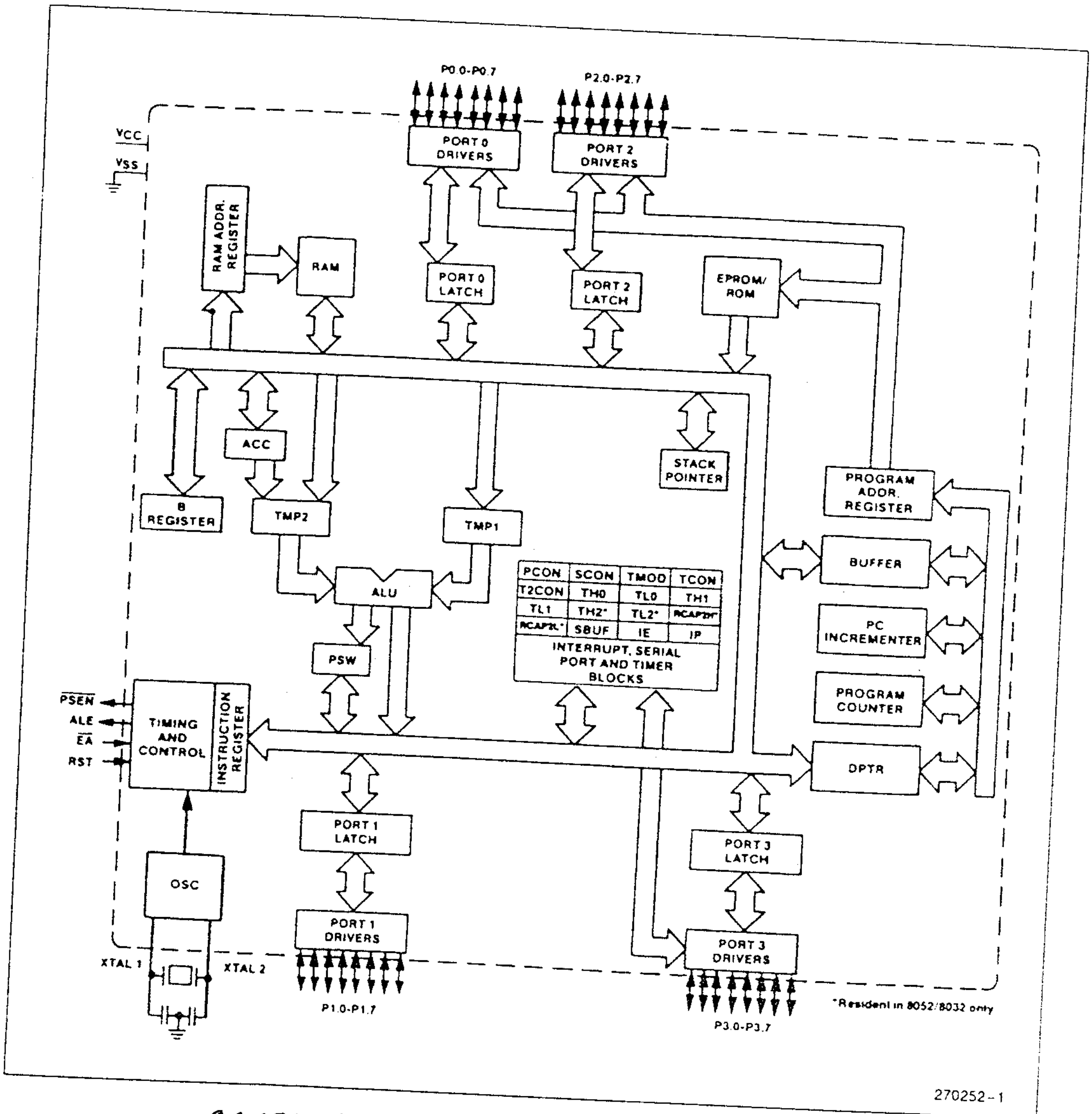
Our project can be extended such that different parameters other than temperature can also be controlled. Totally 8 parameters can be controlled through 8 channels. For this purpose the 8 channel ADC chip can be used. This will result in software complesity since mcu has to control all the parameters. Open loop control of the parameters can also be done by suign relays which switches the heater ON or OFF depending upon the decrease or increase of the measured value with respect to the set value.

Bibliography...

BIBLIOGRAPHY

1. PATRANABIS - PRINCIPLES OF PROCESS CONTROL
2. PETER HARRIOT - PROCESS CONTROL
3. 8 - Bit Embedded Microcontroller Handbook, Intel. Corp.
4. Douglass V. Hall, "Microprocessors and Interfacing - programming and Hardware - Tata McGraw Hill, 1991.
5. Rafiquzzman, "Microprocessors and Microcomputer based system Design" - Universal book stall, 1992.
6. K.P. SAWHENY - Electronic Measurements and Instru mentation.
7. T.T.L. Data book - Intel. Corp.
8. Linear IC manual - Intel corp.

Appendix...



80C31 BLOCK DIAGRAM

Hex Code	Number of Bytes	Mnemonic	Operands
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr

Hex Code	Number of Bytes	Mnemonic	Operands
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

80C31 INSTRUCTION SET

Mnemonic		Description	Byte	Oscillator Period
DATA TRANSFER (Continued)				
MOV	@Ri,direct	Move direct byte to indirect RAM	2	24
MOV	@Ri,#data	Move immediate data to indirect RAM	2	12
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	24
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to Acc	1	24
MOVC	A,@A+PC	Move Code byte relative to PC to Acc	1	24
MOVX	A,@Ri	Move External RAM (8-bit addr) to Acc	1	24
MOVX	A,@DPTR	Move External RAM (16-bit addr) to Acc	1	24
MOVX	@Ri,A	Move Acc to External RAM (8-bit addr)	1	24
MOVX	@DPTR,A	Move Acc to External RAM (16-bit addr)	1	24
PUSH	direct	Push direct byte onto stack	2	24
POP	direct	Pop direct byte from stack	2	24
XCH	A,Rn	Exchange register with Accumulator	1	12
XCH	A,direct	Exchange direct byte with Accumulator	2	12
XCH	A,@Ri	Exchange indirect RAM with Accumulator	1	12
XCHD	A,@Ri	Exchange low-order Digit indirect RAM with Acc	1	12

Mnemonic		Description	Byte	Oscillator Period
BOOLEAN VARIABLE MANIPULATION				
CLR	C	Clear Carry	1	12
CLR	bit	Clear direct bit	2	12
SETB	C	Set Carry	1	12
SETB	bit	Set direct bit	2	12
CPL	C	Complement Carry	1	12
CPL	bit	Complement direct bit	2	12
ANL	C,bit	AND direct bit to CARRY	2	24
ANL	C,/bit	AND complement of direct bit to Carry	2	24
ORL	C,bit	OR direct bit to Carry	2	24
ORL	C,/bit	OR complement of direct bit to Carry	2	24
MOV	C,bit	Move direct bit to Carry	2	12
MOV	bit,C	Move Carry to direct bit	2	24
JC	rel	Jump if Carry is set	2	24
JNC	rel	Jump if Carry not set	2	24
JB	bit,rel	Jump if direct Bit is set	3	24
JNB	bit,rel	Jump if direct Bit is Not set	3	24
JBC	bit,rel	Jump if direct Bit is set & clear bit	3	24
PROGRAM BRANCHING				
ACALL	addr11	Absolute Subroutine Call	2	24
LCALL	addr16	Long Subroutine Call	3	24
RET		Return from Subroutine	1	24
RETI		Return from interrupt	1	24
AJMP	addr11	Absolute Jump	2	24
LJMP	addr16	Long Jump	3	24
SJMP	rel	Short Jump (relative addr)	2	24

80C31 INSTRUCTION SET

Hex Code	Number of Bytes	Mnemonic	Operands
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr, code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr, code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A, # data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr, code addr
31	2	ACALL	code addr
32	1	RETI	

Hex Code	Number of Bytes	Mnemonic	Operands
33	1	RLC	A
34	2	ADDC	A, # data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr, # data
44	2	ORL	A, # data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr, # data
54	2	ANL	A, # data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr,A
63	3	XRL	data addr, # data
64	2	XRL	A, # data
65	2	XRL	A,data addr

80C31 INSTRUCTION SET

Hex Code	Number of Bytes	Mnemonic	Operands
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C,bit addr
73	1	JMP	@A+DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A+PC
84	1	DIV	AB
85	3	MOV	data addr, data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A+DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0

Hex Code	Number of Bytes	Mnemonic	Operands
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,/bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,/bit addr
B1	2	ACALL	code addr
B2	2	CPL	bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3

80 C31 INSTRUCTION SET

Mnemonic		Description	Byte	Oscillator Period
PROGRAM BRANCHING (Continued)				
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	24
JZ	rel	Jump if Accumulator is Zero	2	24
JNZ	rel	Jump if Accumulator is Not Zero	2	24
CJNE	A,direct,rel	Compare direct byte to Acc and Jump if Not Equal	3	24
CJNE	A,#data,rel	Compare immediate to Acc and Jump if Not Equal	3	24

Mnemonic		Description	Byte	Oscillator Period
PROGRAM BRANCHING (Continued)				
CJNE	Rn,#data,rel	Compare immediate to register and Jump if Not Equal	3	24
CJNE	@Ri,#data,rel	Compare immediate to indirect and Jump if Not Equal	3	24
DJNZ	Rn,rel	Decrement register and Jump if Not Zero	2	24
DJNZ	direct,rel	Decrement direct byte and Jump if Not Zero	3	24
NOP		No Operation	1	12

80C31 INSTRUCTION SET

Interrupt Response Time: Refer to Hardware Description Chapter.

Instructions that Affect Flag Settings(1)

Instruction	Flag C	Flag OV	Flag AC	Instruction	Flag C	Flag OV	Flag AC
ADD	X	X	X	CLR C	O		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C,bit	X		
MUL	O	X		ANL C,/bit	X		
DIV	O	X		ORL C,bit	X		
DA	X			ORL C,bit	X		
RRC	X			MOV C,bit	X		
RLC	X			CJNE	X		
SETB C	1						

(1)Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

Note on instruction set and addressing modes:

Rn — Register R7-R0 of the currently selected Register Bank.

direct — 8-bit internal data location's address. This could be an Internal Data RAM location (0-127) or a SFR [i.e., I/O port, control register, status register, etc. (128-255)].

@Ri — 8-bit internal data RAM location (0-255) addressed indirectly through register R1 or R0.

#data — 8-bit constant included in instruction.

#data 16 — 16-bit constant included in instruction.

addr 16 — 16-bit destination address. Used by LCALL & LJMP. A branch can be anywhere within the 64K-byte Program Memory address space.

addr 11 — 11-bit destination address. Used by ACALL & AJMP. The branch will be within the same 2K-byte page of program memory as the first byte of the following instruction.

rel — Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is −128 to +127 bytes relative to first byte of the following instruction.

bit — Direct Addressed bit in Internal Data RAM or Special Function Register.

Mnemonic		Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS				
ADD	A,Rn	Add register to Accumulator	1	12
ADD	A,direct	Add direct byte to Accumulator	2	12
ADD	A,@Ri	Add indirect RAM to Accumulator	1	12
ADD	A,#data	Add immediate data to Accumulator	2	12
ADDC	A,Rn	Add register to Accumulator with Carry	1	12
ADDC	A,direct	Add direct byte to Accumulator with Carry	2	12
ADDC	A,@Ri	Add indirect RAM to Accumulator with Carry	1	12
ADDC	A,#data	Add immediate data to Acc with Carry	2	12
SUBB	A,Rn	Subtract Register from Acc with borrow	1	12
SUBB	A,direct	Subtract direct byte from Acc with borrow	2	12
SUBB	A,@Ri	Subtract indirect RAM from ACC with borrow	1	12
SUBB	A,#data	Subtract immediate data from Acc with borrow	2	12
INC	A	Increment Accumulator	1	12
INC	Rn	Increment register	1	12
INC	direct	Increment direct byte	2	12
INC	@Ri	Increment direct RAM	1	12
DEC	A	Decrement Accumulator	1	12
DEC	Rn	Decrement Register	1	12
DEC	direct	Decrement direct byte	2	12
DEC	@Ri	Decrement indirect RAM	1	12

80C31 INSTRUCTION SET

Mnemonic		Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS (Continued)				
INC	DPTR	Increment Data Pointer	1	24
MUL	AB	Multiply A & B	1	48
DIV	AB	Divide A by B	1	48
DA	A	Decimal Adjust Accumulator	1	12
LOGICAL OPERATIONS				
ANL	A,Rn	AND Register to Accumulator	1	12
ANL	A,direct	AND direct byte to Accumulator	2	12
ANL	A,@Ri	AND indirect RAM to Accumulator	1	12
ANL	A,#data	AND immediate data to Accumulator	2	12
ANL	direct,A	AND Accumulator to direct byte	2	12
ANL	direct,#data	AND immediate data to direct byte	3	24
ORL	A,Rn	OR register to Accumulator	1	12
ORL	A,direct	OR direct byte to Accumulator	2	12
ORL	A,@Ri	OR indirect RAM to Accumulator	1	12
ORL	A,#data	OR immediate data to Accumulator	2	12
ORL	direct,A	OR Accumulator to direct byte	2	12
ORL	direct,#data	OR immediate data to direct byte	3	24
XRL	A,Rn	Exclusive-OR register to Accumulator	1	12
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	12
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	12
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	12
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	12
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	24
CLR	A	Clear Accumulator	1	12
CPL	A	Complement Accumulator	1	12

Mnemonic		Description	Byte	Oscillator Period
LOGICAL OPERATIONS (Continued)				
RL	A	Rotate Accumulator Left	1	12
RLC	A	Rotate Accumulator Left through the Carry	1	12
RR	A	Rotate Accumulator Right	1	12
RRC	A	Rotate Accumulator Right through the Carry	1	12
SWAP	A	Swap nibbles within the Accumulator	1	12
DATA TRANSFER				
MOV	A,Rn	Move register to Accumulator	1	12
MOV	A,direct	Move direct byte to Accumulator	2	12
MOV	A,@Ri	Move indirect RAM to Accumulator	1	12
MOV	A,#data	Move immediate data to Accumulator	2	12
MOV	Rn,A	Move Accumulator to register	1	12
MOV	Rn,direct	Move direct byte to register	2	24
MOV	Rn,#data	Move immediate data to register	2	12
MOV	direct,A	Move Accumulator to direct byte	2	12
MOV	direct,Rn	Move register to direct byte	2	24
MOV	direct,direct	Move direct byte to direct	3	24
MOV	direct,@Ri	Move indirect RAM to direct byte	2	24
MOV	direct,#data	Move immediate data to direct byte	3	24
MOV	@Ri,A	Move Accumulator to indirect RAM	1	12

80C51 INSTRUCTION SET

National Semiconductor

ADC0808, ADC0809 8-Bit μP Compatible A/D Converters With 8-Channel Multiplexer

General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals.

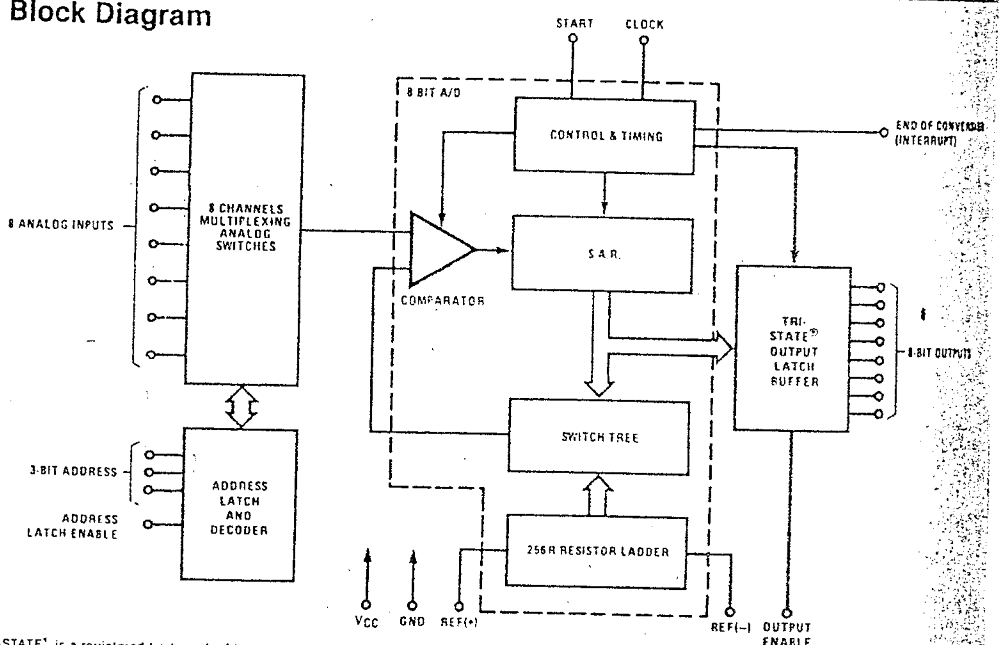
The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE® outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.

Features

- Resolution — 8-bits
- Total unadjusted error — ± 1/2 LSB and ± 1 LSB
- No missing codes
- Conversion time — 100 μs
- Single supply — 5 V_{DC}
- Operates ratiometrically or with 5 V_{DC} or analog span adjusted voltage reference
- 8-channel multiplexer with latched control logic
- Easy interface to all microprocessors, or operates "stand alone"
- Outputs meet T^2L voltage level specifications
- 0V to 5V analog input voltage range with single 5V supply
- No zero or full-scale adjust required
- Standard hermetic or molded 28-pin DIP package
- Temperature range −40°C to +85°C or −55°C to +125°C
- Low power consumption — 15 mW
- Latched TRI-STATE® output

Block Diagram



...te Maximum Ratings (Notes 1 and 2)

...(V_{CC}) (Note 3)	6.5V
...y Pin	$-0.3V$ to $(V_{CC}+0.3V)$
...ntrol Inputs	
...ntrol Inputs	$-0.3V$ to $+15V$
...(CLOCK, ALE, ADD A, ADD B, ADD C)	
...perature Range	$-65°C$ to $+150°C$
...ipation at $T_A = 25°C$	875 mW
...(Soldering, 10 seconds)	300°C

Operating Ratings (Notes 1 and 2)

Temperature Range (Note 1)	$T_{MIN} \le T_A \le T_{MAX}$
ADC0808CJ	$-55°C \le T_A \le +125°C$
ADC0808CCJ, ADC0808CCN, ADC0809CCN	$-40°C \le T_A \le +85°C$
Range of V_{CC} (Note 1)	$4.5 V_{DC}$ to $6.0 V_{DC}$

...ical Characteristics

...r Specifications: $V_{CC} = 5 V_{DC} = V_{REF(+)}$, $V_{REF(-)} = GND$, $T_{MIN} \le T_A \le T_{MAX}$ and $f_{CLK} = 640$ kHz ...erwise stated.

Parameter	Conditions	Min	Typ	Max	Units
ADC0808					
Total Unadjusted Error (Note 5)	25°C			± 1/2	LSB
	T_{MIN} to T_{MAX}			± 3/4	LSB
ADC0809					
Total Unadjusted Error (Note 5)	0°C to 70°C			± 1	LSB
	T_{MIN} to T_{MAX}			± 1 1/4	LSB
Input Resistance	From Ref(+) to Ref(−)	1.0	2.5		kΩ
Analog Input Voltage Range	(Note 4) V(+) or V(−)	GND−0.10		V_{CC}+0.10	V_{DC}
Voltage, Top of Ladder	Measured at Ref(+)		V_{CC}	V_{CC}+0.1	V
Voltage, Center of Ladder		V_{CC}/2−0.1	V_{CC}/2	V_{CC}/2+0.1	V
Voltage, Bottom of Ladder	Measured at Ref(−)	−0.1	0		V
Comparator Input Current	$f_c = 640$ kHz, (Note 6)	−2	±0.5	2	μA

...ctrical Characteristics

...l Levels and DC Specifications: ADC0808CJ $4.5V \le V_{CC} \le 5.5V$, $-55°C \le T_A \le +125°C$ unless otherwise noted ...0808CCJ, ADC0808CCN, and ADC0809CCN $4.75 \le V_{CC} \le 5.25V$, $-40°C \le T_A \le +85°C$ unless otherwise noted

Parameter	Conditions	Min	Typ	Max	Units
...ALOG MULTIPLEXER					
OFF Channel Leakage Current	$V_{CC} = 5V$, $V_{IN} = 5V$,				
	$T_A = 25°C$		10	200	nA
	T_{MIN} to T_{MAX}			1.0	μA
OFF Channel Leakage Current	$V_{CC} = 5V$, $V_{IN} = 0$,				
	$T_A = 25°C$	−200	−10		nA
	T_{MIN} to T_{MAX}	−1.0			μA
...NTROL INPUTS					
Logical "1" Input Voltage		V_{CC}−1.5			V
Logical "0" Input Voltage				1.5	V
Logical "1" Input Current (The Control Inputs)	$V_{IN} = 15V$			1.0	μA
Logical "0" Input Current (The Control Inputs)	$V_{IN} = 0$	−1.0			μA
Supply Current	$f_{CLK} = 640$ kHz		0.3	3.0	mA

Functional Description

Multiplexer: The device contains an 8-channel single-ended analog signal multiplexer. A particular input channel is selected by using the address decoder. Table I shows the input states for the address lines to select any channel. The address is latched into the decoder on the low-to-high transition of the address latch enable signal.

TABLE I

SELECTED ANALOG CHANNEL	ADDRESS LINE C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

CONVERTER CHARACTERISTICS

The Converter

The heart of this single chip data acquisition system is its 8-bit analog-to-digital converter. The converter is designed to give fast, accurate, and repeatable conversions over a wide range of temperatures. The converter is partitioned into 3 major sections: the 256R ladder network, the successive approximation register, and the comparator. The converter's digital outputs are positive true.

The 256R ladder network approach *(Figure 1)* was chosen over the conventional R/2R ladder because of its inherent monotonicity, which guarantees no missing digital codes. Monotonicity is particularly important in closed loop feedback control systems. A non-monotonic relationship can cause oscillations that will be catastrophic for the system. Additionally, the 256R network does not cause load variations on the reference voltage.

The bottom resistor and the top resistor of the ladder network in *Figure 1* are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occurs when the analog signal has reached +1/2 LSB and succeeding output transitions occur every 1 LSB later up to full-scale.

The successive approximation register (SAR) performs 8 iterations to approximate the input voltage. For any SAR type converter, n-iterations are required for an n-bit converter. *Figure 2* shows a typical example of a 3-bit converter. In the ADC0808, ADC0809, the approximation technique is extended to 8 bits using the 256R network.

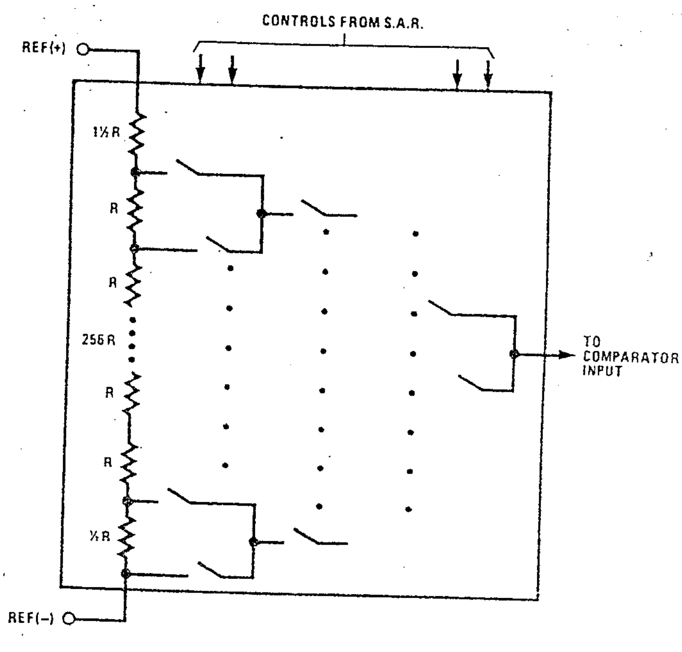


FIGURE 1. Resistor Ladder and Switch Tree

Electrical Characteristics (Continued)

Digital Levels and DC Specifications: ADC0808CJ $4.5V \leq V_{CC} \leq 5.5V$, $-55°C \leq T_A \leq +125°C$ unless otherwise noted
ADC0808CCJ, ADC0808CCN, and ADC0809CCN $4.75 \leq V_{CC} \leq 5.25V$, $-40°C \leq T_A \leq +85°C$ unless otherwise noted

Parameter		Conditions	Min	Typ	Max	Units
DATA OUTPUTS AND EOC (INTERRUPT)						
$V_{OUT(1)}$	Logical "1" Output Voltage	$I_O = -360\ \mu A$	V_{CC}-0.4			V
$V_{OUT(0)}$	Logical "0" Output Voltage	$I_O = 1.6$ mA			0.45	V
$V_{OUT(0)}$	Logical "0" Output Voltage EOC	$I_O = 1.2$ mA			0.45	V
I_{OUT}	TRI-STATE® Output Current	$V_O = 5V$			3	μA
		$V_O = 0$	−3			μA

Electrical Characteristics

Timing Specifications: $V_{CC} = V_{REF(+)} = 5V$, $V_{REF(-)} = GND$, $t_r = t_f = 20$ ns and $T_A = 25°C$ unless otherwise noted.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{WS}	Minimum Start Pulse Width	*(Figure 5)*		100	200	ns
t_{WALE}	Minimum ALE Pulse Width	*(Figure 5)*		100	200	ns
t_s	Minimum Address Set-Up Time	*(Figure 5)*		25	50	ns
t_H	Minimum Address Hold Time	*(Figure 5)*		25	50	ns
t_D	Analog MUX Delay Time From ALE	$R_S = 0\Omega$ *(Figure 5)*		1	2.5	μs
t_{H1}, t_{H0}	OE Control to Q Logic State	$C_L = 50$ pF, $R_L = 10k$ *(Figure 8)*		125	250	ns
t_{1H}, t_{0H}	OE Control to Hi-Z	$C_L = 10$ pF, $R_L = 10k$ *(Figure 8)*		125	250	ns
t_c	Conversion Time	$f_c = 640$ kHz, *(Figure 5)* (Note 7)	90	100	116	μs
f_c	Clock Frequency		10	640	1280	kHz
t_{EOC}	EOC Delay Time	*(Figure 5)*	0		8 + 2 μs	Clock Periods
C_{IN}	Input Capacitance	At Control Inputs		10	15	pF
C_{OUT}	TRI-STATE® Output Capacitance	At TRI-STATE® Outputs, (Note 12)		10	15	pF

Note 1: Absolute maximum ratings are those values beyond which the life of the device may be impaired.

Note 2: All voltages are measured with respect to GND, unless otherwise specified.

Note 3: A zener diode exists, internally, from V_{CC} to GND and has a typical breakdown voltage of 7 V_{DC}.

Note 4: Two on-chip diodes are tied to each analog input which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CC} supply. The spec allows 100 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 100 mV, the output code will be correct. To achieve an absolute 0 V_{DC} to 5 V_{DC} input voltage range will therefore require a minimum supply voltage of 4.900 V_{DC} over temperature variations, initial tolerance and loading.

Note 5: Total unadjusted error includes offset, full-scale, linearity, and multiplexer errors. See *Figure 3*. None of these A/Ds requires a zero or full-scale adjust. However, if an all zero code is desired for an analog input other than 0.0V, or if a narrow full-scale span exists (for example: 0.5V to 4.5V full-scale) the reference voltages can be adjusted to achieve this. See *Figure 13*.

Note 6: Comparator input current is a bias current into or out of the chopper stabilized comparator. The bias current varies directly with clock frequency and has little temperature dependence *(Figure 6)*. See paragraph 4.0.

Note 7: The outputs of the data register are updated one clock cycle before the rising edge of EOC.

National Semiconductor

DAC0808, DAC0807, DAC0806 8-Bit D/A Converters

General Description

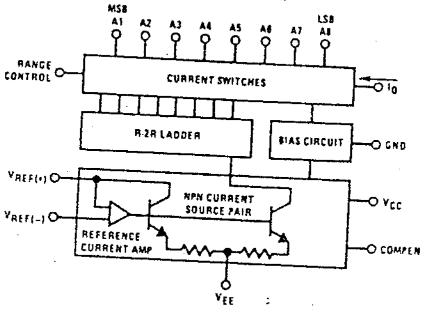
The DAC0808 series is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 150 ns while dissipating only 33 mW with ±5V supplies. No reference current (I_{REF}) trimming is required for most applications since the full scale output current is typically ±1 LSB of 255 I_{REF}/256. Relative accuracies of better than ±0.19% assure 8-bit monotonicity and linearity while zero level output current of less than 4 μA provides 8-bit zero accuracy for $I_{REF} \geq 2$ mA. The power supply currents of the DAC0808 series are independent of bit codes, and exhibits essentially constant device characteristics over the entire supply voltage range.

The DAC0808 will interface directly with popular TTL, DTL or CMOS logic levels, and is a direct replacement for the MC1508/MC1408. For higher speed applications, see DAC0800 data sheet.

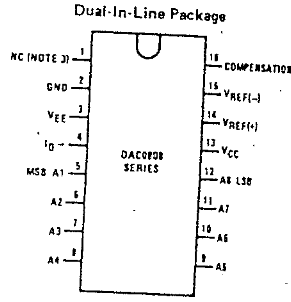
Features

- Relative accuracy: ±0.19% error maximum (DAC0808)
- Full scale current match: ±1 LSB typ
- 7 and 6-bit accuracy available (DAC0807, DAC0806)
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 8 mA/μs
- Power supply voltage range: ±4.5V to ±18V
- Low power consumption: 33 mW @ ±5V

Block and Connection Diagrams

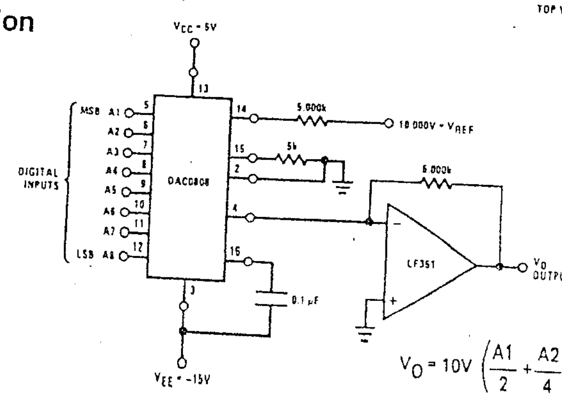


Dual-In-Line Package



TOP VIEW

Typical Application



$$V_O = 10V \left(\frac{A1}{2} + \frac{A2}{4} + \ldots \frac{A8}{256}\right)$$

FIGURE 1. +10V Output Digital to Analog Converter

Ordering Information

ACCURACY	OPERATING TEMPERATURE RANGE	ORDER NUMBERS* D PACKAGE (D16C)		J PACKAGE (J16A)		N PACKAGE (N16A)	
8-bit	$-55°C \leq T_A \leq +125°C$	DAC0808LD	MC1508L8				
8-bit	$0°C \leq T_A \leq +75°C$			DAC0808LCJ	MC1408L8	DAC0808LCN	MC1408P8
7-bit	$0°C \leq T_A \leq +75°C$			DAC0807LCJ	MC1408L7	DAC0807LCN	MC1408P7
6-bit	$0°C \leq T_A \leq +75°C$			DAC0806LCJ	MC1408L6	DAC0806LCN	MC1408P6

*Note. Devices may be ordered by using either order number.

Functional Description (Continued)

The A/D converter's successive approximation register (SAR) is reset on the positive edge of the start conversion (SC) pulse. The conversion is begun on the falling edge of the start conversion pulse. A conversion in process will be interrupted by receipt of a new start conversion pulse. Continuous conversion may be accomplished by tying the end-of-conversion (EOC) output to the SC input. If used in this mode, an external start conversion pulse should be applied after power up. End-of-conversion will go low between 0 and 8 clock pulses after the rising edge of start conversion.

The most important section of the A/D converter is the comparator. It is this section which is responsible for the ultimate accuracy of the entire converter. It is also the comparator drift which has the greatest influence on the repeatability of the device. A chopper-stabilized comparator provides the most effective method of satisfying all the converter requirements.

The chopper-stabilized comparator converts the DC input signal into an AC signal. This signal is then fed through a high gain AC amplifier and has the DC level restored. This technique limits the drift component of the amplifier since the drift is a DC component which is not passed by the AC amplifier. This makes the entire A/D converter extremely insensitive to temperature, long term drift and input offset errors.

Figure 4 shows a typical error curve for the ADC0808 as measured using the procedures outlined in AN-179.

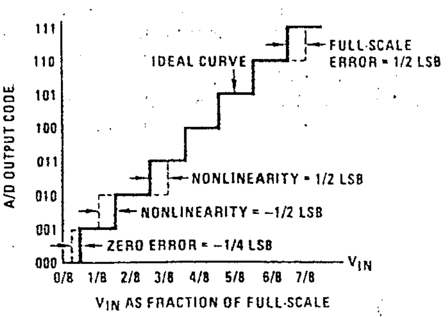


FIGURE 2. 3-Bit A/D Transfer Curve

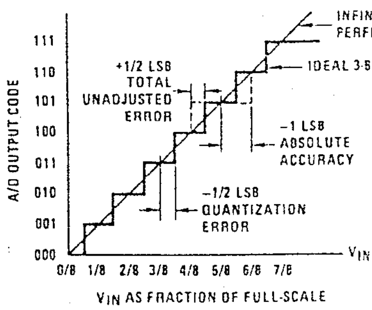


FIGURE 3. 3-Bit A/D Absolute Accuracy Curve

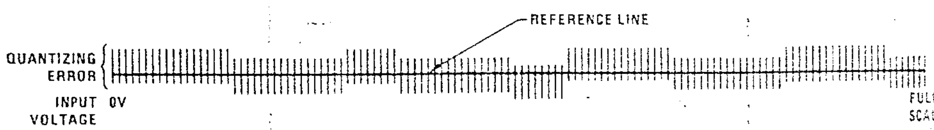
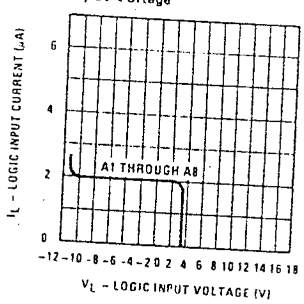


FIGURE 4. Typical Error Curve

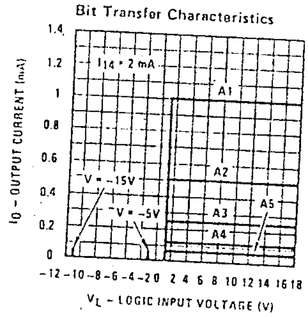
Typical Performance Characteristics

V_{CC} = 5V, V_{EE} = −15V, T_A = 25°C, unless otherwise noted

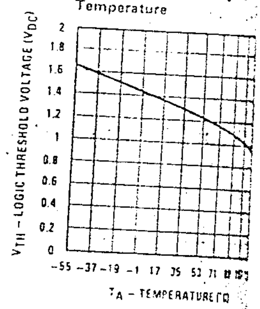
Logic Input Current vs Input Voltage



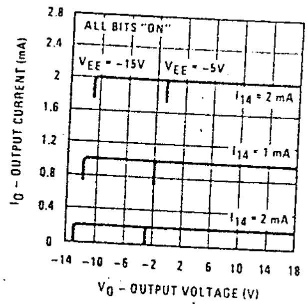
Bit Transfer Characteristics



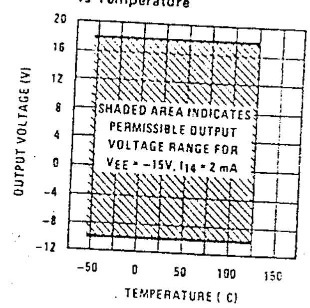
Logic Threshold Voltage vs Temperature



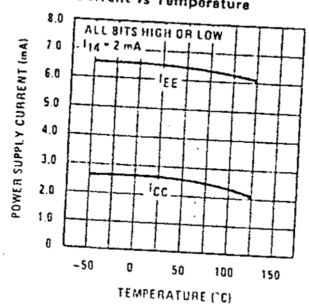
Output Current vs Output Voltage (Output Voltage Compliance)



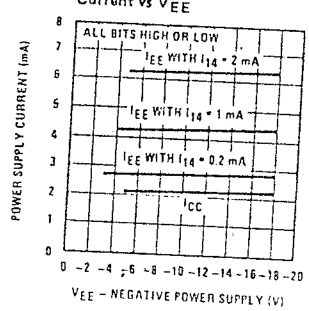
Output Voltage Compliance vs Temperature



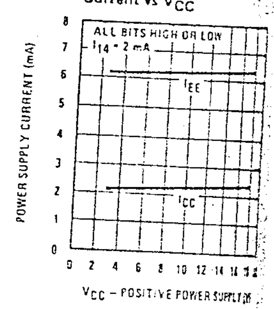
Typical Power Supply Current vs Temperature



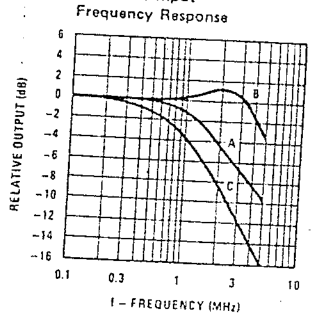
Typical Power Supply Current vs V_{EE}



Typical Power Supply Current vs V_{CC}



Reference Input Frequency Response



Unless otherwise specified: R14 = R15 = 1 kΩ, C = 15 pF, pin 16 to V_{EE}; R_L = 50Ω, pin 4 to ground.

Curve A: Large Signal Bandwidth Method of *Figure 7*, V_{REF} = 2 Vp-p offset 1 V above ground

Curve B: Small Signal Bandwidth Method of *Figure 7*, R_L = 250Ω, V_{REF} = 50 mVp-p offset 200 mV above ground.

Curve C: Large and Small Signal Bandwidth Method of *Figure 9* (no op amp, R_L = 50Ω), R_S = 50Ω, V_{REF} = 2V, V_S = 100 mVp-p centered at 0V.

Absolute Maximum Ratings

Power Supply Voltage	
V_{CC}	+18 V_{DC}
V_{EE}	−18 V_{DC}
Digital Input Voltage, V5–V12	−10 V_{DC} to +18 V_{DC}
Applied Output Voltage, V_O	−11 V_{DC} to +18 V_{DC}
Reference Current, I_{14}	5 mA
Reference Amplifier Inputs, V14, V15	V_{CC}, V_{EE}
Power Dissipation (Package Limitation)	1000 mW
Derate above T_A = 25°C	6.7 mW/°C
Operating Temperature Range	
DAC0808L	−55°C ≤ T_A ≤ +125°C
DAC0808LC Series	0 ≤ T_A ≤ +75°C
Storage Temperature Range	−65°C to +150°C

Electrical Characteristics

(V_{CC} = 5V, V_{EE} = −15 V_{DC}, V_{REF}/R14 = 2 mA, DAC0808: T_A = −55°C to +125°C, DAC0808C, DAC0807C, DAC0806C, T_A = 0°C to +75°C, and all digital inputs at high logic level unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Relative Accuracy (Error Relative to Full Scale I_O)	*(Figure 4)*				%
DAC0808L (LM1508-8), DAC0808LC (LM1408-8)				±0.19	%
DAC0807LC (LM1408-7), (Note 1)				±0.39	%
DAC0806LC (LM1408-6), (Note 1)				±0.78	%
Settling Time to Within 1/2 LSB (Includes t_{PLH})	T_A = 25°C (Note 2), *(Figure 5)*		150		ns
Propagation Delay Time	T_A = 25°C, *(Figure 5)*		30	100	ns
Output Full Scale Current Drift			±20		ppm/°C
Digital Input Logic Levels	*(Figure 3)*				
High Level, Logic "1"		2			V_{DC}
Low Level, Logic "0"				0.8	V_{DC}
Digital Input Current	*(Figure 3)*				
High Level	V_{IH} = 5V		0	0.040	mA
Low Level	V_{IL} = 0.8V		−0.003	−0.8	mA
Reference Input Bias Current	*(Figure 3)*		−1	−3	μA
Output Current Range	*(Figure 3)*				
	V_{EE} = −5V	0	2.0	2.1	mA
	V_{EE} = −15V, T_A = 25°C	0	2.0	4.2	mA
Output Current	V_{REF} = 2.000V, R14 = 1000Ω, *(Figure 3)*	1.9	1.99	2.1	mA
Output Current, All Bits Low	*(Figure 3)*		0	4	μA
Output Voltage Compliance	E_r ≤ 0.19%, T_A = 25°C				
Pin 1 Grounded,				−0.55, +0.4	V_{DC}
V_{EE} Below −10V				−5.0, +0.4	V_{DC}
SR I_{REF} Reference Current Slew Rate	*(Figure 6)*	4	8		mA/μs
Output Current Power Supply Sensitivity	−5V ≤ V_{EE} ≤ −16.5V		0.05	2.7	μA/V
Power Supply Current (All Bits Low)	*(Figure 3)*				
I_{CC}			2.3	22	mA
I_{EE}			−4.3	−13	mA
Power Supply Voltage Range	T_A = 25°C, *(Figure 3)*				
V_{CC}		4.5	5.0	5.5	V_{DC}
V_{EE}		−4.5	−15	−16.5	V_{DC}
Power Dissipation					
All Bits Low	V_{CC} = 5V, V_{EE} = −5V		33	170	mW
	V_{CC} = 5V, V_{EE} = −15V		106	305	mW
All Bits High	V_{CC} = 15V, V_{EE} = −5V		90		mW
	V_{CC} = 15V, V_{EE} = −15V		160		mW

Note 1: All current switches are tested to guarantee at least 50% of rated current.
Note 2: All bits switched.
Note 3: Range control is not required.

SN54122, SN54123, SN54130, SN54L122, SN54L123, SN54LS122, SN54LS123, SN74122, SN74123, SN74130, SN74LS122, SN74LS123 RETRIGGERABLE MONOSTABLE MULTIVIBRATORS

REVISED DECEMBER 1983

-)-C Triggered from Active-High or ctive -Low Gated Logic Inputs
- etriggerable for Very Long Output Pulses, p to 100% Duty Cycle
- verriding Clear Terminates Output Pulse
- 122, 'L122, 'LS122 Have Internal Timing esistors

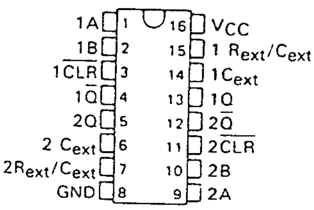
ption

hese d-c triggered multivibrators feature output pulse idth control by three methods. The basic pulse time is rogrammed by selection of external resistance and apacitance values (see typical application data). The 122, 'L122, and 'LS122 have internal timing resistors hat allow the circuits to be used with only an external apacitor, if so desired. Once triggered, the basic pulse idth may be extended by retriggering the gated low-vel-active (A) or high-level-active (B) inputs, or be educed by use of the overriding clear. Figure 1 il-ustrates pulse control by retriggering and early clear.

he 'LS122 and 'LS123 are provided enough Schmitt ysteresis to ensure jitter-free triggering from the B in-ut with transition rates as slow as 0.1 millivolt per anosecond.

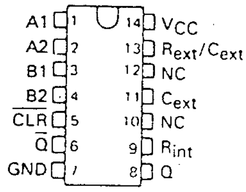
he R_{int} is nominally 10 k ohms for '122, 'LS122, and s nominally 20 k ohms for 'L122.

SN54122, SN54LS122 . . . J OR W PACKAGE
SN54L122 . . . J PACKAGE
SN74122 . . . J OR N PACKAGE
SN74LS122 . . . D, J OR N PACKAGE
(TOP VIEW) (SEE NOTES 1 THRU 4)

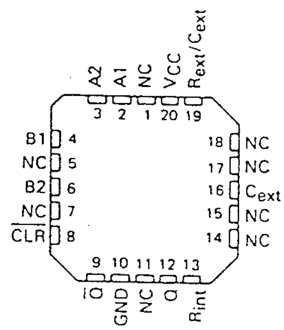


1. An external timing capacitor may be connected between C_{ext} and R_{ext}/C_{ext} (positive).
2. To use the internal timing resistor of '122, 'L122, or 'LS122, connect R_{int} to V_{CC}.
3. For improved pulse width accuracy and repeatability, connect an external resistor between R_{ext}/C_{ext} and V_{CC} with R_{int} open-circuited.
4. To obtain variable pulse widths, connect an external variable resistance between R_{int} or R_{ext}/C_{ext} and V_{CC}.

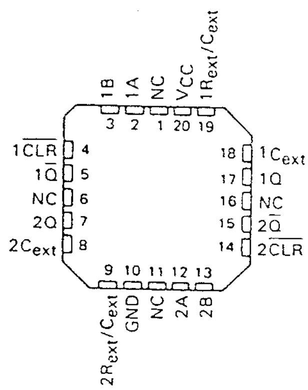
SN54123, SN54130, SN54LS123 . . . J OR W PACKAGE
SN54L123 . . . J PACKAGE
SN74123, SN74130 . . . J OR N PACKAGE
SN74LS123 . . . D, J OR N PACKAGE
(TOP VIEW) (SEE NOTES 1 THRU 4)



SN54LS122 . . . FK PACKAGE
SN74LS122 . . . FN PACKAGE
(TOP VIEW) (SEE NOTES 1 THRU 4)



SN54LS123 . . . FK PACKAGE
SN74LS123 . . . FN PACKAGE
(TOP VIEW) (SEE NOTES 1 THRU 4)



NC - No internal connection

3

TTL DEVICES

TYPES SN54LS138, SN54S138, SN74LS138, SN74S138
3-LINE TO 8-LINE DECODERS/DEMULTIPLEXERS

DECEMBER 1972—REVISED APRIL 1985

- **Designed Specifically for High-Speed:**
  Memory Decoders
  Data Transmission Systems
- **3 Enable Inputs to Simplify Cascading and/or Data Reception**
- **Schottky-Clamped for High Performance**

scription

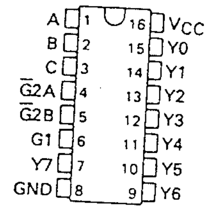
These Schottky-clamped TTL MSI circuits are designed to be used in high-performance memory decoding or data-routing applications requiring very short propagation delay times. In high-performance memory systems these decoders can be used to minimize the effects of system decoding. When employed with high-speed memories utilizing a fast enable circuit the delay times of these decoders and the enable time of the memory are usually less than the typical access time of the memory. This means that the effective system delay introduced by the Schottky-clamped system decoder is negligible.

The 'LS138 and 'S138 decode one of eight lines dependent on the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented without external inverters and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

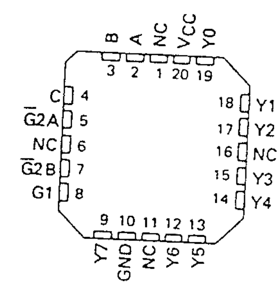
All of these decoder/demultiplexers feature fully buffered inputs, each of which represents only one normalized load to its driving circuit. All inputs are clamped with high-performance Schottky diodes to suppress line-ringing and to simplify system design.

The SN54LS138 and SN54S138 are characterized for operation over the full military temperature range of −55°C to 125°C. The SN74LS138 and SN74S138 are characterized for operation from 0°C to 70°C.

SN54LS138, SN54S138 . . . J OR W PACKAGE
SN74LS138, SN74S138 . . . D, J OR N PACKAGE
(TOP VIEW)

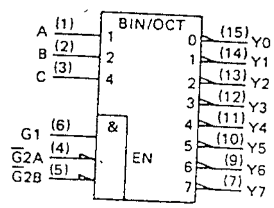


SN54LS138, SN54S138 . . . FK PACKAGE
SN74LS138, SN74S138 . . . FN PACKAGE
(TOP VIEW)

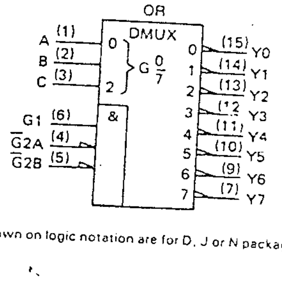


NC – No internal connection

logic symbols



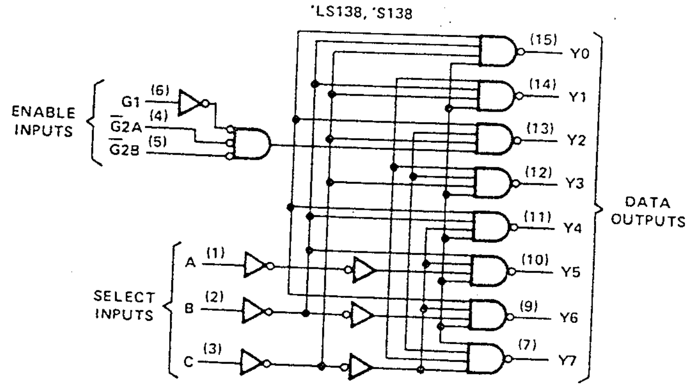
OR



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

logic diagram and function table

'LS138, 'S138



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

'LS138, 'S138
FUNCTION TABLE

INPUTS					OUTPUTS							
ENABLE		SELECT										
G1	$\bar{G}2$*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

*$\bar{G}2 = \bar{G}2A + \bar{G}2B$

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

12

DUAL D-TYPE FLIP-FLOP

The HEF4013B is a dual D-type flip-flop which features independent set direct (S_D), clear direct (C_D), clock inputs (CP) and outputs (O,$\bar{O}$). Data is accepted when CP is LOW and transferred to the output on the positive-going edge of the clock. The active HIGH asynchronous clear-direct (C_D) and set-direct (S_D) are independent and override the D or CP inputs. The outputs are buffered for best system performance. Schmitt-trigger action in the clock input makes the circuit highly tolerant to slower clock rise and fall times.

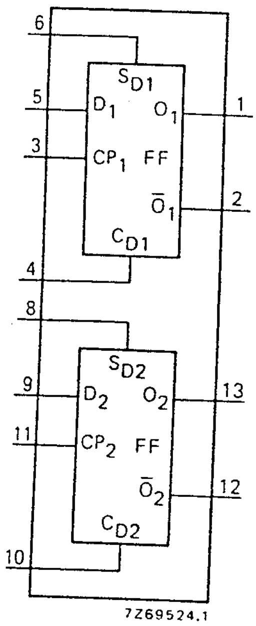


Fig. 1 Functional diagram.

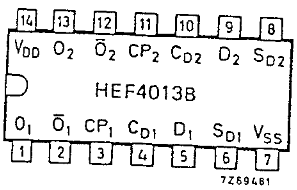


Fig. 2 Pinning diagram.

FUNCTION TABLES

inputs				outputs	
S_D	C_D	CP	D	O	$\bar{O}$
H	L	X	X	H	L
L	H	X	X	L	H
H	H	X	X	H	H

inputs				outputs	
S_D	C_D	CP	D	O_{n+1}	$\bar{O}_{n+1}$
L	L	∫	L	L	H
L	L	∫	H	H	L

H = HIGH state (the more positive voltage)
L = LOW state (the less positive voltage)
X = state is immaterial
∫ = positive-going transition
O_{n+1} = state after clock positive transition

PINNING

D data inputs
CP clock input (L to H edge-triggered)
S_D asynchronous set-direct input (active HIGH)
C_D asynchronous clear-direct input (active HIGH)
O true output
$\bar{O}$ complement output

HEF4013BP : 14-lead DIL; plastic (SOT-27K, M, T).
HEF4013BD: 14-lead DIL; ceramic (cerdip) (SOT-73).
HEF4013BT : 14-lead mini-pack; plastic (SO-14; SOT-108A).

FAMILY DATA

I_{DD} LIMITS category FLIP-FLOPS } see Family Specifications

1-OF-16 DECODER/DEMULTIPLEXER WITH INPUT LATCHES

The HEF4515B is a 1-of-16 decoder/demultiplexer, having four binary weighted address inputs (A_0 to A_3), a latch enable input (EL), and an active LOW enable input ($\overline{E}$). The 16 outputs ($\overline{O}_0$ to $\overline{O}_{15}$) are mutually exclusive active LOW. When EL is HIGH, the selected output is determined by the data on A_n. When EL goes LOW, the last data present at A_n are stored in the latches and the outputs remain stable. When $\overline{E}$ is LOW, the selected output, determined by the contents of the latch, is LOW. At $\overline{E}$ HIGH, all outputs are HIGH. The enable input ($\overline{E}$) does not affect the state of the latch. When the HEF4515B is used as a demultiplexer, $\overline{E}$ is the data input and A_0 to A_3 are the address inputs.

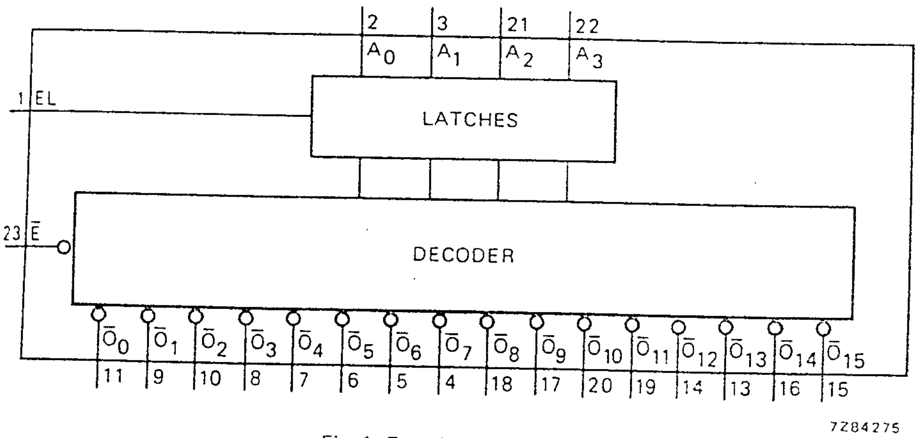


Fig. 1 Functional diagram.

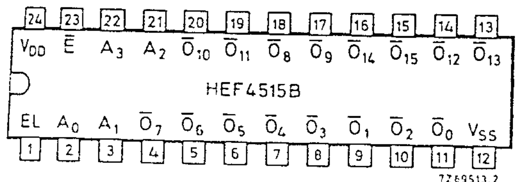


Fig. 2 Pinning diagram.

PINNING

A_0 to A_3	address inputs
$\overline{E}$	enable input (active LOW)
EL	latch enable input
$\overline{O}_0$ to $\overline{O}_{15}$	outputs (active LOW)

HEF4515BP : 24-lead DIL; plastic (SOT-101A).
HEF4515BD: 24-lead DIL; ceramic (cerdip) (SOT-94).
HEF4515BT : 24-lead mini-pack; plastic (SO-24; SOT-137A).

APPLICATION INFORMATION

Some examples of applications for the HEF4515B are:

- Digital multiplexing.
- Address decoding.
- Hexadecimal/BCD decoding.

FAMILY DATA	see Family Specifications
I_{DD} LIMITS category MSI	