ELECTRONIC DASH BOARD SPEEDO METER

PROJECT WORK

SUBMITTED BY

HEMA. J
SRIVIDHYA. N
SENTHIL KUMAR. L
SATHISH SARAVANAN. S

INTERNAL GUIDE

Mr. P. Govindaraju
Assistant Professor, ECE Department

EXTERNAL GUIDE

Mr. S. Sivakumar
Miss. P. Sudha
PE [Electronics] PRICOL

1997-98

Submitted in partial fulfilment of the requirements for the award of the degree of BACHELOR OF ENGINEERING IN ELECTRONICS AND COMMUNICATION ENGINEERING Branch of the Bharathiar University

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Kumaraguru College of Technology

COIMBATORE - 641 006.

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641006.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

Name.. *Reg. No* ....

certified that this is the project work entitled

"ELECTRONIC DASH BOARD - SPEEDO METER"

done by

Mr/Miss ....

In partial fulfilment for the degree of

Bachelor of Engineering in

Electronics and Communication Engineering,

Branch of the Bharathiar University,

during the academic year 1997-98

....

Head of the Department. *Guide*

Date :

Submitted for the University Examination held on....................

----------------------------- -----------------------------

Internal Examiner *External Examiner.*

MIER INSTRUMENTS & CONTROLS LIMITED

TO WHOMSOEVER IT MAY CONCERN

P-1340

This is to certify that the following Final Year B.E.(Electronics and Communication Engineering) students of Kumaraguru College Of Technology, Coimbatore have worked on project work entitled "Electronic Dash Board - Speedometer" in Product Engg (Electronics) department in our Organisation from July 1997 to March 1998.

1. HEMA J
2. SRIVIDHYA N
3. SATHISH SARAVANAN S
4. SENTHIL KUMAR L

During this period their attendance and conduct were found to be good.

We wish them the very best in the future endeavours.

R. [illegible]

ACKNOWLEDGMENT

We express our sincere thanks to our Principal **Dr.Subramanian**, B.E., M.Sc., (Engg.), Ph.D., for providing us the necessary facilities.

We are much grateful to our Head of Department **Prof.M.Ramaswamy**, M.E., MIEEE., MISTE., for his encouragement and guidance.

There are no words to thank our Internal guide, **Mr.Govindaraj**.M.E., Assistant Professor ECE Department for his encouragement, inspiration, guidance and suggestions in preparation and completion of this project.

We are extremely thankful to **M/s.PRICOL**, Coimbatore for sponsoring our project and availing their lab facilities. We are especially thankful to our external guides **Mr.S.Siva Kumar** and **Miss.P.Sudha** for extending their maximum support for without whose help our project wouldn't have been very successful.

We sincerely thank **Ms.P.Vijayalakshmi** and **Mr.D.Suresh** who has always been with us and providing the timely help for the completion of the project.

Last but not the least, we extend our heartfelt gratitude to **Mr.R.Krishnamurthy**, Deputy General Manager, PRICOL under whom me worked. We are thankful to him for his consistent encouragement in the completion of the project.

We thank our beloved parents and friends for their consistent and unlimited support.

SYNOPSIS

A Dash Board used in automobiles consists of speedometer,odometer,temperature and pressure gauges,fuel guages. The measured parameters are displayed in analog form, and measurement is made by mechanical means.

An "ELECTRONIC DASH BOARD" is a recent development incorporated in automobiles. The complexity of mechanical measurement is reduced by the use of electronic devices.

This project particularly deals with the electronic measurement of speed and distance , and is displayed using analog meters.

A Reed Switch sensor which is connected to the gear box of the vehicle generates number of pulses depending on the Gear Ratio. These generated pulses are given as inputs to the Zilog Z86E3O micro controller port lines. The microcontroller generates a PWM output on the respective output port lines for the movement of the pointer in the meter according to the angle corresponding to the speed. The speed is displayed in analog form.

The Odometer consists of an impulse counter which increments by 0.1km for the specified number of pulses.

The usage of electronic components in the measurement of parameters of an automobile has many advantages over the existing mechanical meters such as low cost,high reliability, increase in quality increase in efficiency and reduction in usage area.

Contents

CONTENTS

1. INTRODUCTION

Today we are stepping into 21st century. So, our science and technology should be abreast with the fast growing world. Present automobiles on Indian roads uses analog devices for indication of fuel level, speed, distance, temperature etc. Recently many known automobile companies have introduced sophisticated electronic devices.

Electronic manufacturing is the fastest growing segment of the manufacturing world. Electromechanical and mechanical devices are being replaced by electronic devices at a fast rate. The emphasis on quality is matched with the expectations that electronic goods will be manufactured with ever increasing capablility at even lower cost.

The digital circuits have many advantages over their analog counterparts. They are more precise, more accurate, more reliable and more flexible system. They can meet the applications with the specified accuracy and dynamic range requirements.

This project uses the Zilog microcontroller Z86E30 and a reed switch sensor. Depending on the speed of the vehicle, the sensor senses the number of pulses and sends it to the microcontroller for the generation of PWM. Depending on the output from the microcontroller the needle connected to the coil deflects and indicates the speed of the vehicle.

2. MEASUREMENT OF SPEED

2.1 GEAR FUNDAMENTALS

Gears are modified simple machines such as lever, pulley, wheel- and-axle, and inclined plane. They all serve to multiply force, change speed, or direction of motion and serve as connecting devices between driving units and driven mechanisms.

The mechanical advantyage of the gear is found by ratio of the resistance to the effort. The mechanical advantage is found by dividing the number of teeth on the driven gear by the number of teeth on the driving gear. This ratio is generally known as GEAR RATIO.

2.2 MECHANICAL MEASUREMENT

From the gear box a cable is connected to the speedometer unit in the dashboard. When ignition is ON and when the vehicle starts moving, the axle connected to the gear box rotates according to the speed of the vi\ehicle. The cable rotates along with the axle which moves the needle in the speedometer. The spring present in the speedo meter moves the pointer back & forth according to the speed. Once the ignition is OFF, the needle is brought to zero position by the spring connected to the pointer.

Depending upon the revolutions of the wheel the counter is incremented by one which is equvalent to incrementing the distance by 0.1 Km.

2.3. ELECTRONIC MEASUREMENT:

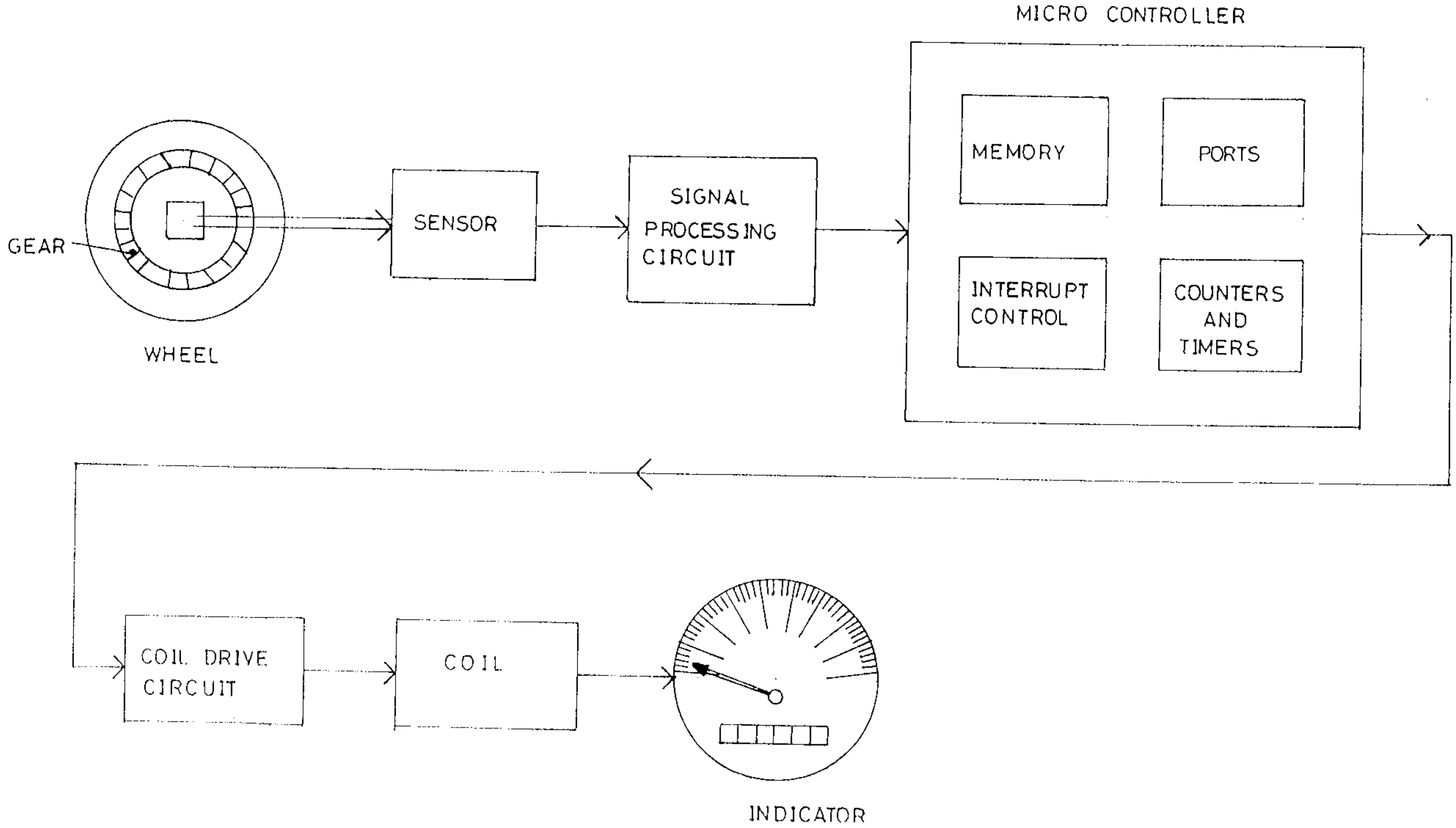
The cable is removed and a reed switch sensor assembly is connected to the projecting end of the gear box. From the other end of the sensor assembly two wires, signal and ground are taken out and connected to the corresponding wires coming out from the speedometer unit. The speedometer is connected to the battery terminals. When the ignition is ON the axle in the gear box starts rotating and depending on the polarities of the magnets the sensor switches ON and OFF. Thus producing a pulse output. The ON and OFF time of the puleses vary according.to the speed. The frequency of the pulses of the pulses produced is calculated using the revolutions per minute of the axle and the number of pulses produced per revolution.

Frequency = rpm x no. of pulses / 60

Thus the frequency is calculated using the formulae,

Number of pulses per second varies according to the gear ratio as specified by the manufacture. Once the frequency is calculated the corresponding angle movement of the pointer in the speedometer is calculated. Based on the angle of deflection, the speedometer dial is calibirated in kms/hour.

From the frequency calculated the odometer reading is incremented by 0.1 km.



BLOCK DIAGRAM

4. SENSOR DESCRIPTION

Reed switch sensor is used for sensing operations and is connected to the gear box to sense the speed of the vehicle.

The advantages of reed switch sensor are:

-Speedy response time

-Hermetically sealed contacts

-Compact size

-Long mechanical life

Read switch characteristics are as follows:

(1) Reed switches are hermetically sealed within a glass tube with inert gas and do not receive any influence from the external atmospheric environment.

(2) Quick response because of small mass of moving parts.

(3) The structure comprises the operating parts and electrical circuits arranged coaxially. Reed switches are suited to applications in radio frequency operation.

(4) Reed switches are compact and light weight.

(5) Superior corrosion resistance and wear resistance of the contacts assures stable switching operation and a long life.

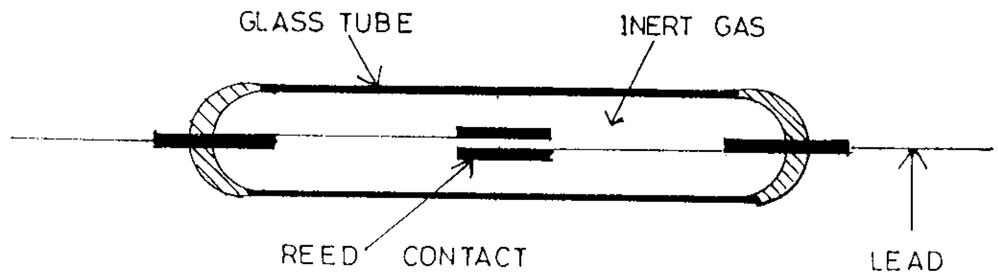
(6) With a permanent magnet installed, reed switches economically and easily become proximity switches.

STRUCTURE:

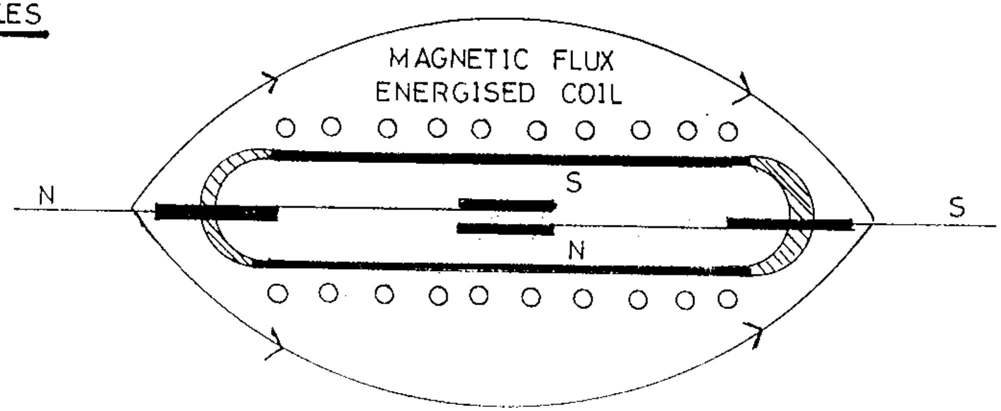
The reed switches comprise two ferromagnetic reeds placed with a gap in between and hermatically sealed in a glass tube. The glass tube is filled withe inert gas to prevent the activation of the contacts. The surfaces of the reed contacts are plated with Rhodium.

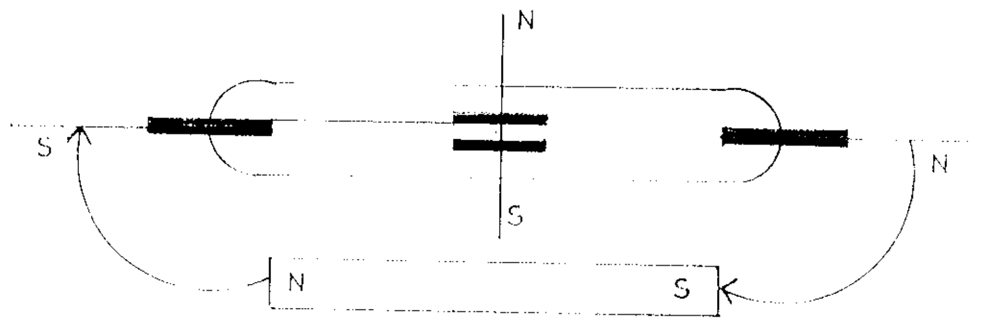
OPERATING PRINCIPLES:

The reed switch is operated by the magnetic field of an energised coil or a permanent magnet which induces north and south poles on the reeds. The reed contacts are closed by this magnetic attractive force. When the magnetic field is removed the reed elasticity causes the contacts to open the circuit.



OPERATING PRINCIPLES





PERMANENT MAGNET

APPLICATIONS OF REED SWITCH:

Reed switches play important roles in the recent marked progress in the development of electronics of and mechatronics equipment. Important applications of reed switches cover a wide variey of fields such as those in:

-Communications equipment

-Office automation equipment

-Control equipment

Consumer electronics equipment.

5. SIGNAL PROCESSING CIRCUIT

The pulses sensed by the sensor which is connected to the gear box cannot be directly sent to the microcontroller for the generation of PWM since the signal is associated with noise. Normally any signal with noise when given to the mocrocontroller disturbs its normal operation. Therefore the signal should be processed , conditioned and converted to perfect pulse. The circuit performing the above operation is called signal conditioning circuit.

The input signal is given to an RC network which acts as a low pass filter. The transorb(ICTE12) is used as a noise protective device which eliminates the high spiked noise.

A +12V given to the circuit through a 10K pull up resistor is dropped to around 6V across the zener diode. The zener diode acts as a coltage regulator. When the load cukrrent and the supply voltage are fixed suitably, the zener diode operates in the breakdown region and the voltage across the diode remains unchanged. The 6V output of the zener is then redukced to about 3.78V using a potential divider circuit.

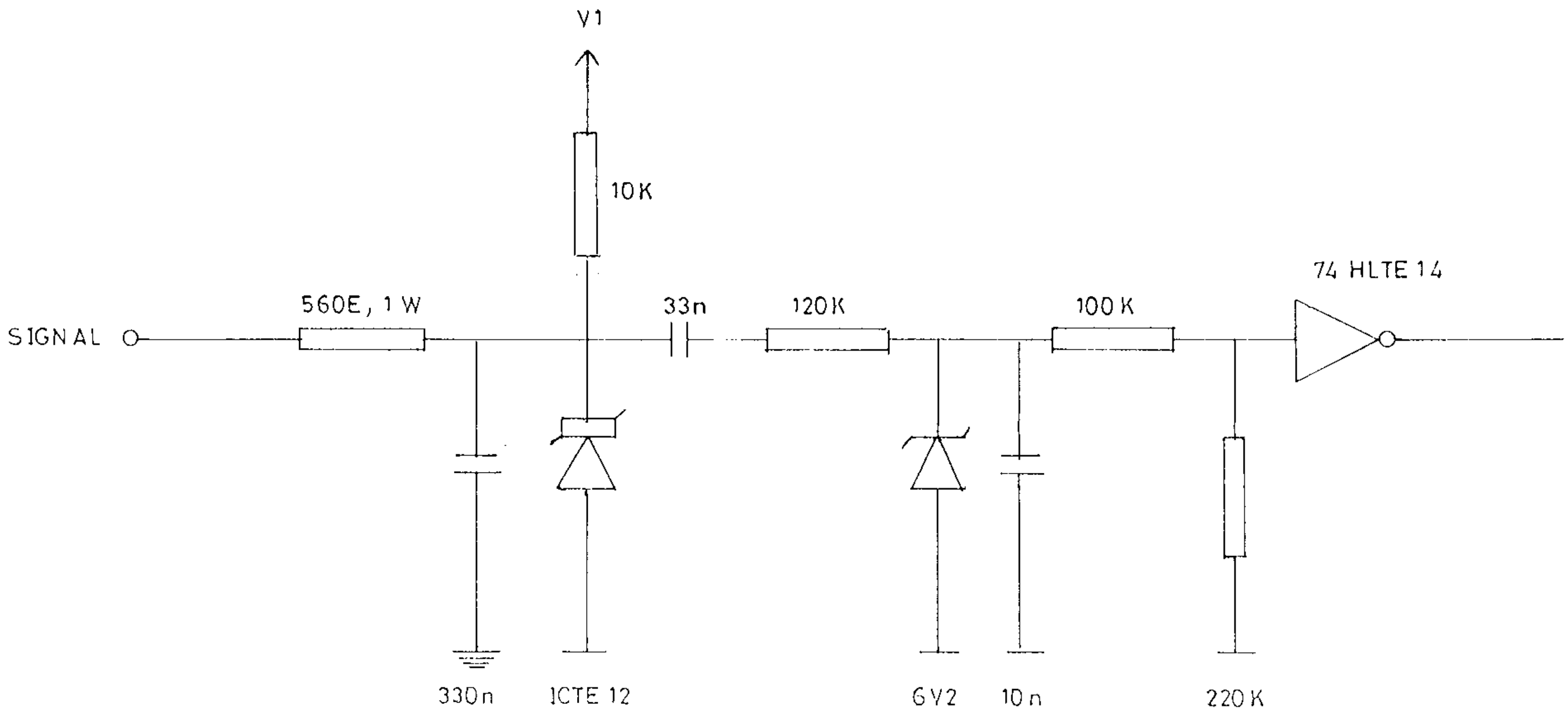
The potential divider divides a voltage into a number of equal parts so that its output voltage is some fraction of the input voltage.

Input voltage to the invertor = (5.5 x 220 K) / (100 K+220 K)

= 3.78 Volts

This 3.78V is given to an invertor which is a schmitt trigger.

A schmitt trigger is basically a square wave generator. Therfore the output of the invertor is a perfect square wave or pulse which is given to the microcontroller.



SIGNAL PROCESSING CIRCUIT

6. MICROCONTROLLER

Z86E30 AND Z86E31 CMOS Z8 OTP CCP MICROCONTROLLER

6.1 FEATURES:

The Z86E30 and Z86E31 have the following general characteristics:

PART	EPROM	RAM	SPEED
Z86E30	4K	236	12MHZ
Z86E31	2K	124	8MHZ

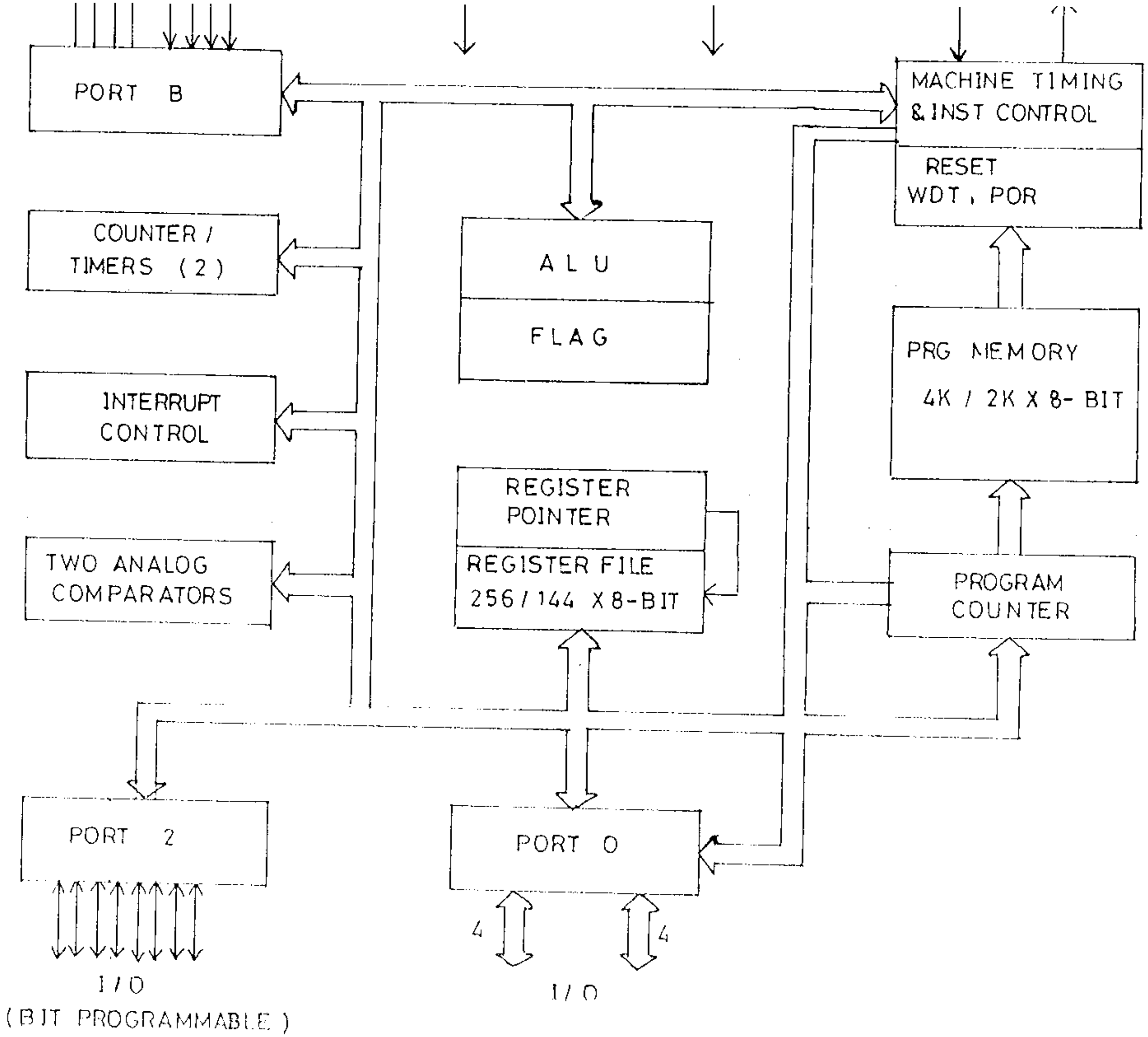
* 8-bit CMOS microcontroller
* 28-pin DIP package
* Low cost
* 4.5V to 5.5V operating range
* Software programmable low EMI mode
* Pull-up active/open drain programmable on ports 0 and 2
* EPROM protect option
* RAM protect programmable
* RC oscillator programmable
* Low power consumption- 60mW
* 2 standby modes (STOP and HALT)

* 24 I/O pins(Three with comparator inputs)
* 17 Digital inputs with CMOS levels,schmitt triggered
* Three digital inputs with CMOS levels only
* Three expanded register file control registers
* Two programmable 8- bit counter/timers each with a 6- bit programmable prescaler
* Six vectored, priority interrupts from six different sources
* Clock speeds upto 8MHZ (E31) and 12MHZ(E3C)
* Watch dog timer
* Auto power- on - reset
* Two comparators with programmable interrupt polarity
* On-chip oscillator that accepts a crystal, ceramic resonator, LC, RC or external clock drive.

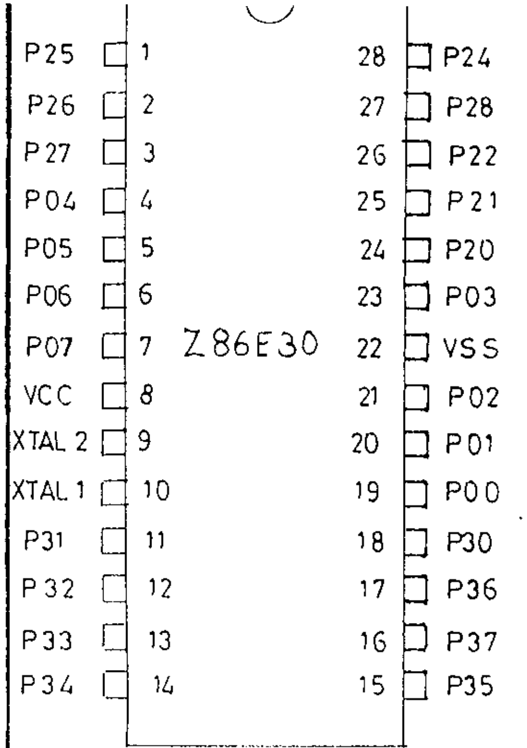
6.2 GENERAL DESCRIPTION:

The Z86E30/31 CCP(consumer controller processor) is a member of the Z8 single chip microcontroller family with 4K/2K bytes of EPROM and 236/124 bytes of RAM.

The Z86E30/31 architecture is characterised by zilog's 8-bit microcontroller core with an expanded register file to allow easy access register mapped peripheral and I/O circuits. The CCP offers a flexible I/O scheme, an efficient register and address space structure, and a number of ancillary features that are useful in many consumer, automotive, peripheral and industrial applications.



FUNCTIONAL BLOCK DIAGRAM



4 - 7	P04 - P07	PORT 0, PINS 4, 5, 6, 7	IN / OUTPUT
8	VCC	POWER SUPPLY	
9	XTAL 2	CRYSTAL. OSCILLATOR	OUT PUT
10	X TAL 1	CRYSTAL OSCILLATOR	INPUT
11 13	P31 P33	PORT 3 , PINS 1, 2, 3	INPUT
14 15	P34 P35	PORT 3 , PINS 4 , 5	OUTPUT
16	P37	PORT 3 , PIN 7	OUTPUT
17	P36	PORT 3 , PIN 6	OUTPUT
18	P30	PORT 3 , PIN 0	INPUT
19 - 21	P00 - P02	PORT 0 , PINS 0 , 1 , 2	IN / OUTPUT
22	VSS	GROUND	
23	P03	PORT 0 , PIN 3	IN / OUTPUT
24 - 28	P20 - P 24	PORT 2 , PINS 0, 1, 2, 3 , 4	IN / OUTPUT

6.3 PORTS:

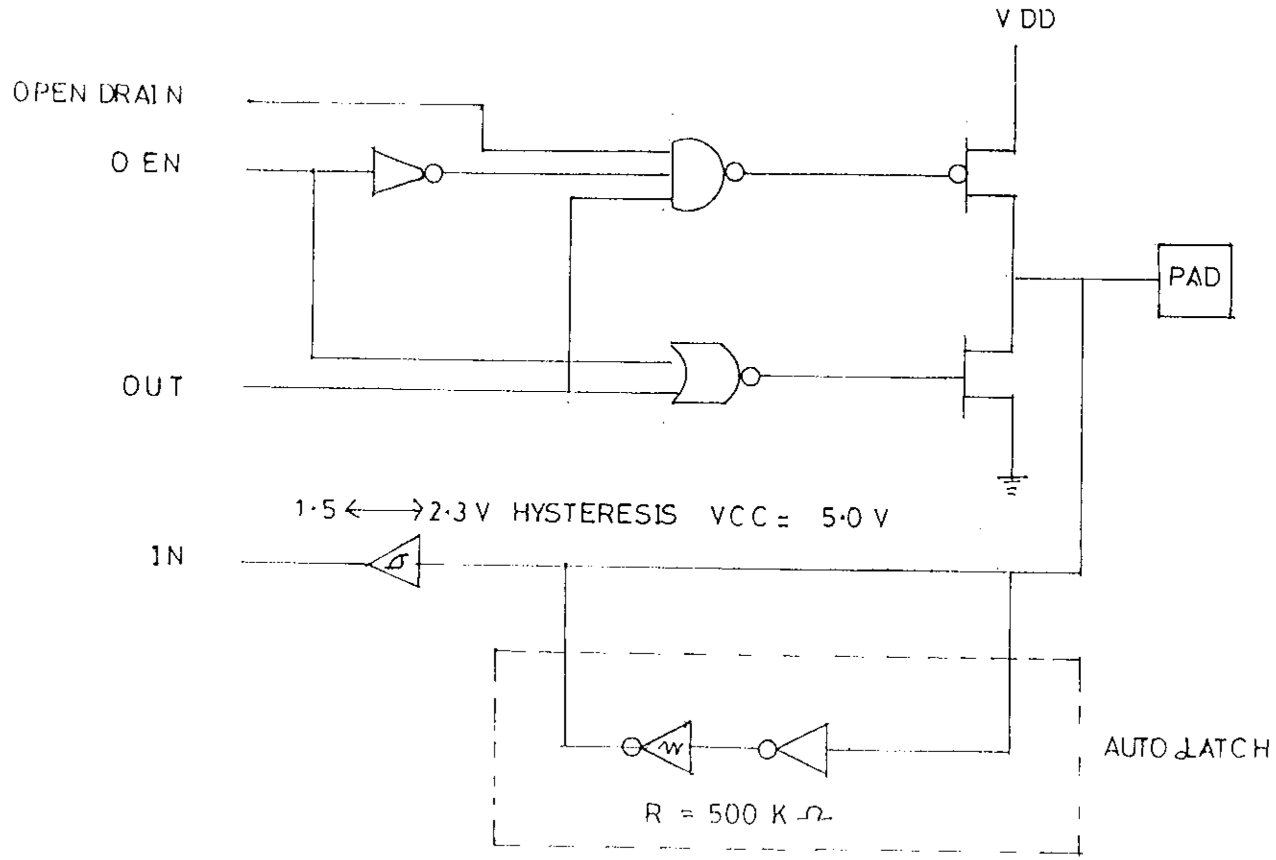
PORT 0(P07-P00)

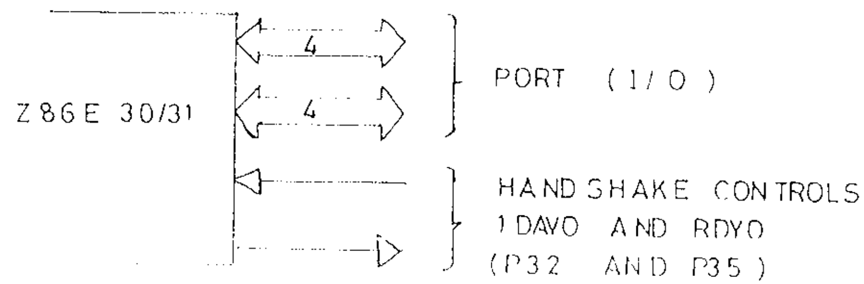
Port 0 is an 8- bit, bi-directional, CMOS compatible I/O port. These eight I/O lines can be nibble programmed as P03-P00 I/O and P07-P04 I/O separately. The input buffers are schmitt-triggered and nibbles programmed as outputs can be globally programmed as eitheir push - pul or open drain. Low EMI output buffers can be globally programmed by the software. Port 0 can be also used as a handshake I/O port.

In Handshake mode, port 3 lines P32 and P35 are used as handshake control lines. The handshake direction is determined by the configuration (input or output) assigned to Port 0's upper nibble. The lower nibble must have the same direction as the upper nibble.

PORT-2(P27-P20)

Port 2 is an 8- bit, bi-directional, CMOS compatible I/O port. These eight I/O lines can be configured under software control as an input or output independently. All the input buffers are schmitt-triggered. Bits programmed as outputs can be globally programmed as either push-pull or open drain. Low EMI output buffers can be globally programmed by the software. When used as an input or output port, port2 can be placed under handshake control In handshake mode, port3 lines P31 and P36 are used as handshake control lines. The handshake direction is determined by the configuration assigned to bit 7 of port2.



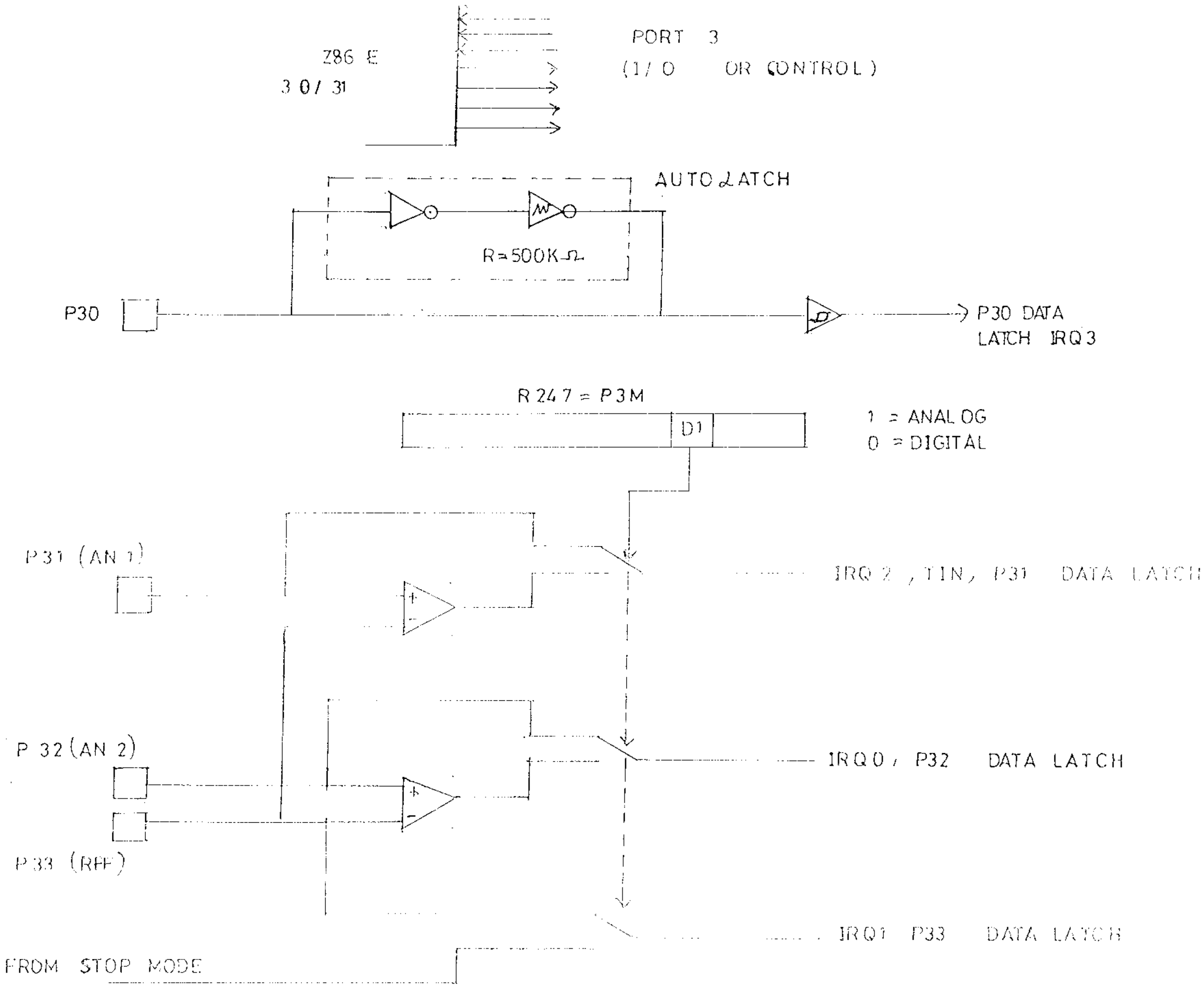


PORT O CONFIGURATION

PORT 3 (P37-P30)

Port 3 is an 8-bit CMOS compatible port with four fixed inputs and four fixed outputs. Port 3 consists of 4 fixed inputs (P33-P30) and 4 fixed outputs (P37-P34) and can be configured under software for interrupt and handshake control functions. Port3, pin 0 is schmitt triggered. Pins P31,P32 and P33 are standard CMOS inputs (no auto latches) and pins P34,P35,P36and P37 are pushpull output lines. Low EMI output buffers can be globally programmed by software. Two on - board comparators can process analog signals on P31 and P32 with reference to the voltage on P33. The comparator output can be outputed from P34 and P37, respectively, by setting the PCON register bit D0 to 1.

The analog function is enabled by setting the D1 of port 3.



PORT 3 CONFIGURATION

MODE REGISTER (P3M)

For the interrupt function, P30 and P33 are falling edge triggered interrupt inputs. P31 and P32 can be programmed as falling, rising or both edge triggered interrupt inputs. Access to counter/timer 1 is made through P31 (Tin) and P36 (Tout). Handshake lines for ports 0 and 2 are also available on port 3. Tin modes are enabled by setting R243 PRE1 bit D1 to 0.

COMPARATOR INPUTS:

Port 3, pins P31 and P32 each have a comparator front end. The comparator reference voltage (pin P33)is common to both comparators. In analog mode, P31 and P32 are the positive inputs of the comparators and P33 is the reference voltage of the comparators.

AUTO LATCH:

The auto latch puts valid CMOS levels on all CMOS inputs (except P33 -P31) that are not externally driven. Whether this is zero or one, cannot be determined. A valid CMOS level, rather than a floating node, reduces excessive supply current flow in the input buffer.

LOW EMI EMISSION:

The Z86E30/31 can be programmed to operate in a low EMI emission mode in the PCON register. The oscillator and all I/O ports can be programmed as low EMI emission mode independently. Use of this feature results in :

* The pre-drivers slew rate reduced to 10 ns typical
* Low EMI output drivers have resistance of 200 ohms (typical)
* Low EMI oscillator
* Internal SCLK/TCLK = XTAL operation limited to a maximum of 4MHZ-250 ns cycle time.

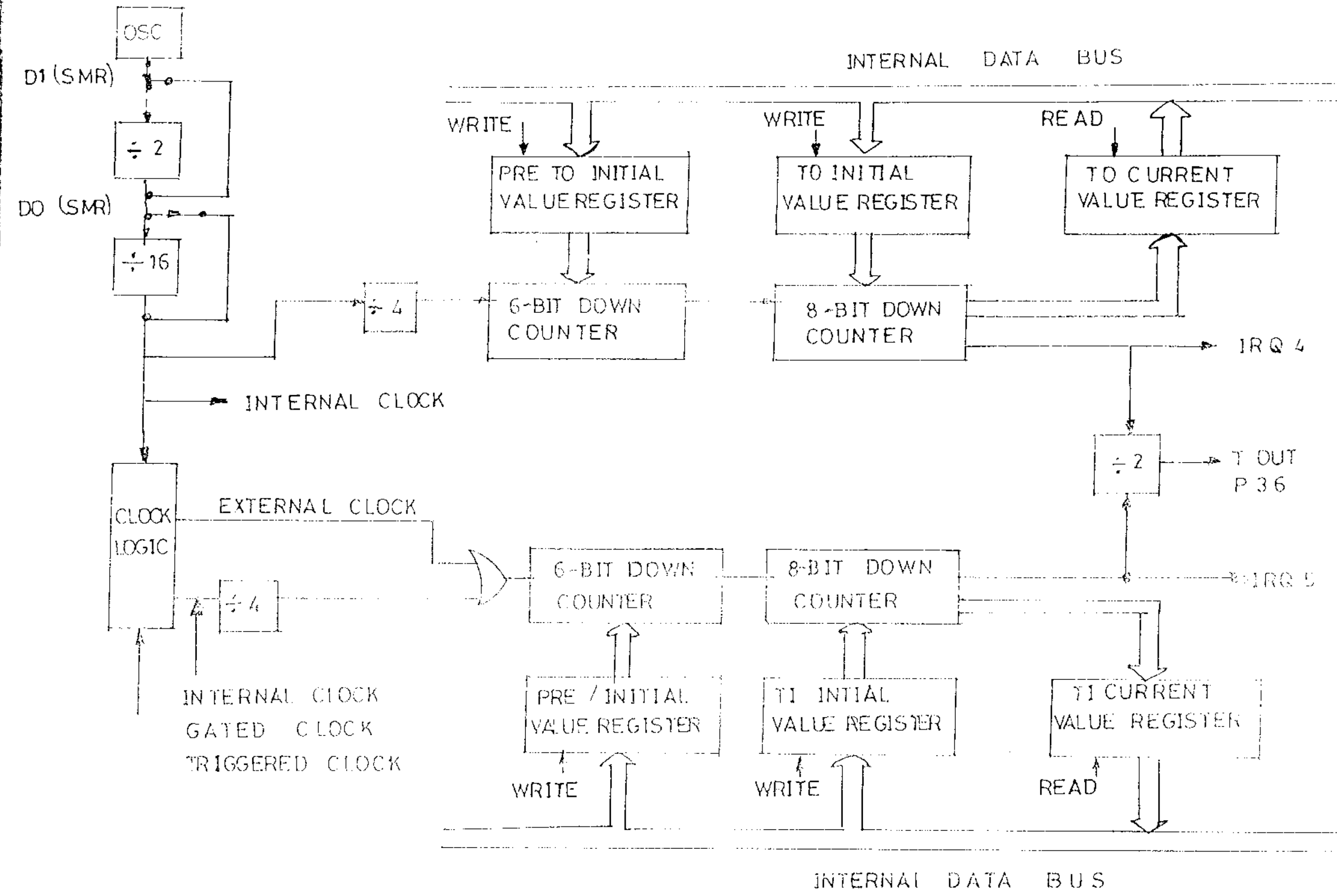
6.4 COUNTER/TIMERS

There are two 8-bit programmable counter/timers (To and Ti) each driven by it's own 6-bit programmable prescaler. The T1 prescaler can be driven by interval or external clock sources; however the T0 prescaler is driven by the internal clock only.

The 6-bit prescalers can divide the input frequency of the clock source by any integer number from 1 to 64. Each prescaler drives its counter, which decrements the value (1 to 256), that has been loaded into the counter. When the counter reaches the end of count, a timer interrupt request IRQ4(T0) or IRQ5(T1) is generated.

The counters can be programmed to start, stop, restart to continue, or restart from the initial value. The counters can also be programmed to stop upon reaching zero (single pass mode) or to automatically reload the initial value and continue counting (modulo-n-continuous mode).

The counters, but not the prescalers can be read at any time without disturbing their value or count mode. The clock source for Ti is user - definable and can be either the internal microprocessor clock divided-by-four, or an external signal input through port3. The Timer mode register configures the external timer input (P30) as an external clock, a trigger input that can be retriggerable are not-retriggerable, or as a gain input for the internal clock. port3 line P36 serves as a timer output (Tout) through which T0,T1 or the internal clock is output. The counter/timers are cascaded by connecting the T0 output to the input to T1.



COUNTER TIMER BLOCK DIAGRAM

6.5 MEMORY ORGANISATION

PROGRAM MEMORY:

The Z86E30/31 can address upto 4K/2K bytes of internal program memory. The first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts. Address 12 (000CH) to address 4095(0FFFH)/2047(07FFH) are reserved for the user program. After reset, the program counter points at the address 000CH which is the starting address of the user program.

EPROM PROTECT:

The 4K/2K bytes program memory is a one time PROM, An EPROM protect feature provents "dumping" of the ROM contents by inhibiting execution of LDC and LDCI instructions to program memory in all modes. In EPROM protect mode, the instructions of LDC and LDCI are disabled globally.

EXPANDED REGISTER FILE:

The register file has been expanded to allow for additional system control registers, mapping of additional peripheral devices, and I/O ports into the register address area. The Z8 register address space R0 through R15 is implemented as 16 groups of 16 registers per group. These register groups are known as the ERF(Expanded Regester File). The low nibble (D3-D0) of the register pointer (RP) selects the active ERF group, and the high nibble (D7-D4) of register RP selects the working register group.

Three system configuration registers reside in the ERF at bank

FH: PCON,SMR, and WDTMR. The rest of the expanded register is not physically implemented and is reserved for future expansion.

REGISTER FILE:

The register file consists of three I/O port registers, 236/124 general - purpose registers,15 control registers and status registers and 3 system configuration registers in the expanded register group. The instructions can access registers directly or indirectly through an 8-bit address field. This allows a short 4- bit register address using the RP. In the 4- bit mode, the register file is divided into 16 working register groups, each occupying 16 continuous locations. The RP addresses the starting location of the working - register groups.

GPR:

The register R254 is a general purpose register.The six sources are divided as follows:

The four sources are claimed by port 3 lines, P33_P30

and two in counter/timers.

RAM PROTECT (Z86E30 ONLY)

The upper portion of the RAM'S address spaces %7F to %EF (excluding the control registers) can be protected from reading and writing. The RAM protect bit option is EPROM-programmable. After the EPROM option is selected, the user can activate from the internal ROM code to turn OFF/ON the RAM protect by loading a bit D6 in the IMR register to either a 0 or 1, respectively. A ,1 in D6 indicates RAM protect enabled.

STACK:

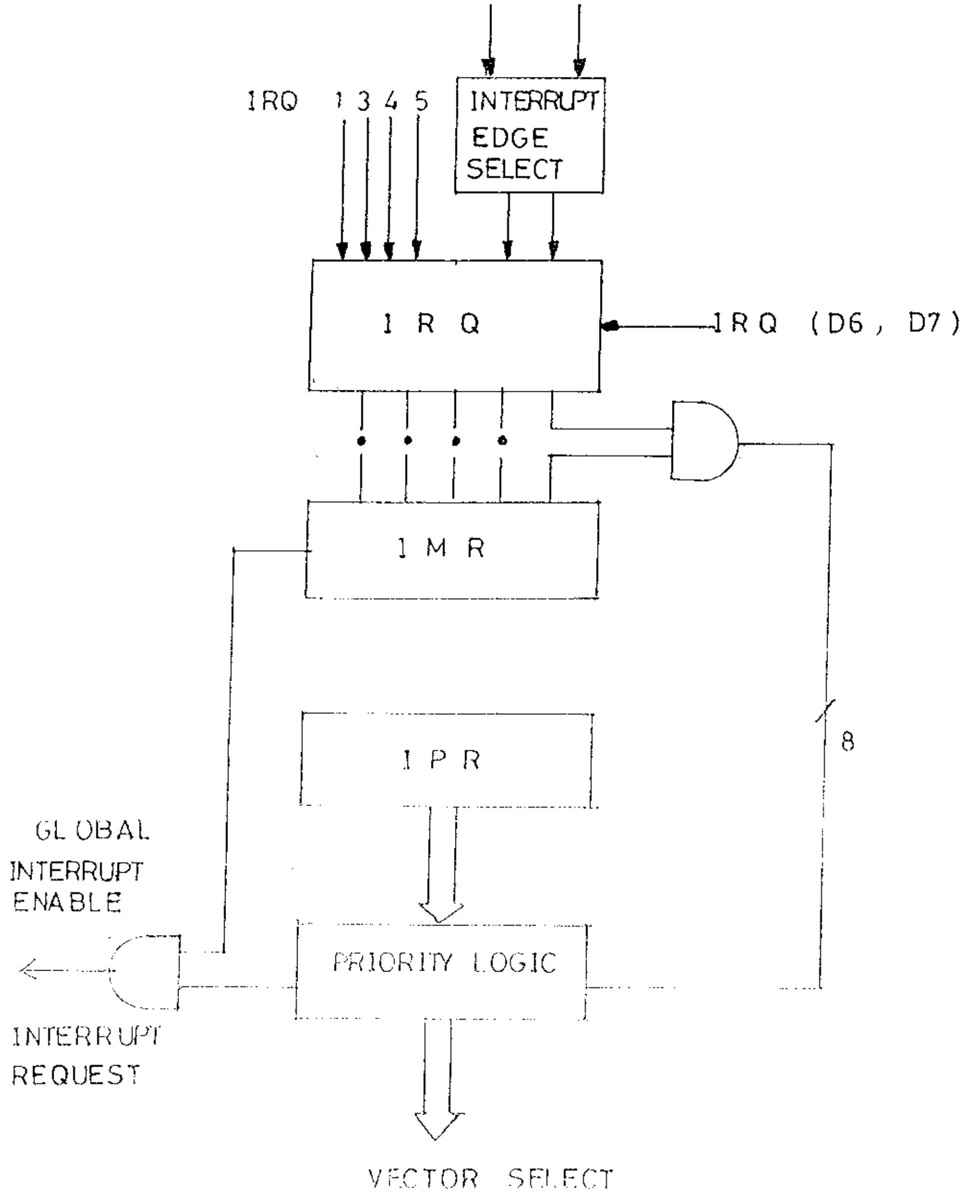
The Z86E30/31 has 236/124 general - purpose registers. An 8 - bit stack pointer (R255) is used for the internal stack that resides within the general - purpose registers.

6.6 INTERRUPTS

The Z86E30/31 has six different interrupts from six different sources. The interrupts are maskable and prioritized. The interrupt mask register globally or individually enables or disables the six interrupt requests.

When more than one interrupt is pending, priorities are resolved by a programmable priority encode that is controlled by the interrupts priority register. An interrupt machine cycle is activated when an interrupt request is granted. Thus, disabling all subsequent interrupts, saves the program counter and status flags, and then branches to the program memory vector location reserved for that interrupt. All Z86E30/31 interrupts are vectored through locations in the program memory. This memory location and the next byte contain the 16- bit starting address of the interrupt service routine for that particular interrupt request.

To accomodate polled interrupt systems, interrupt inputs are masked and the interrupt request register is polled to determine which of the interrupt requests need service.



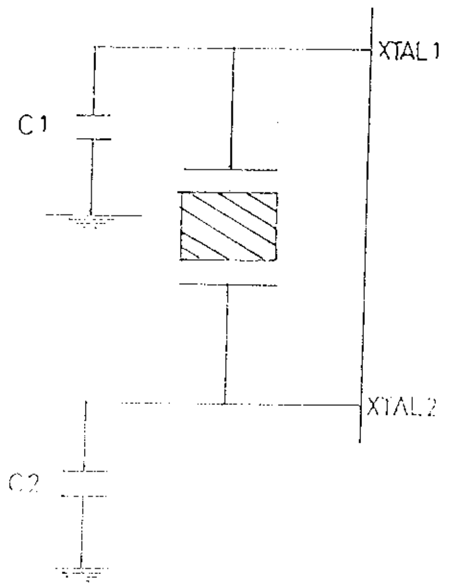
INTERRUPT BLOCK DIAGRAM

6.7 OSCILLATOR CHARACTERISTICS

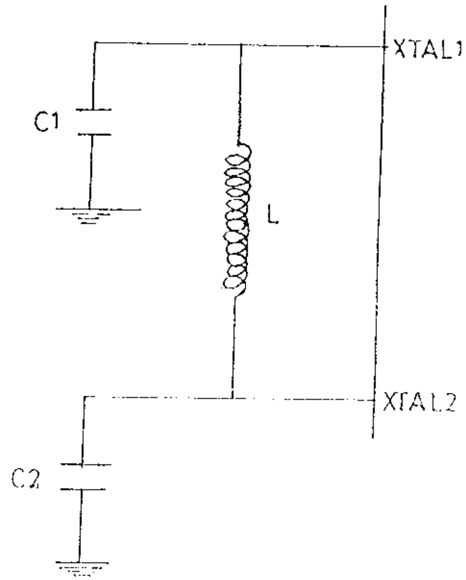
The Z86E30/31 on - chip oscillator has a high - gain, parallel - resonant amplifier for connection to a crystal, RC, ceramic resonator, or any suitable external clock source

(XTAL1 = Input, XTAL2 = Output). The crystal should be at cut, 10 KHz to 12 KHz max., with a series resistance(RS) less than or equal to 100 ohms.

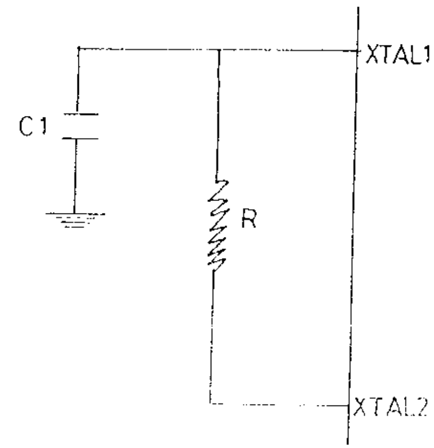
The crystal should be connected across XTAL1 and XTAL2 using the vendor's recommended capacitors from each pin directly to device pin 22. This is to reduce injection of system ground noise. The RC oscillator configuration must be an external resistor connected from XTAL1 to XTAL2, with a frequency - setting capacitor from XTAL1 to ground.



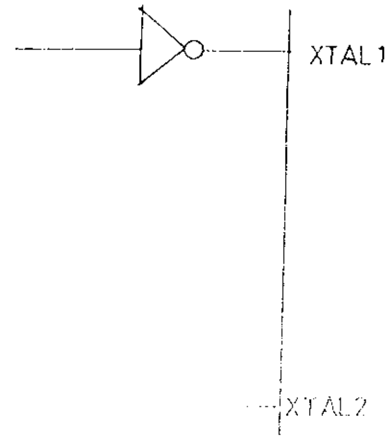
CERAMIC RESONATOR(OR)
CRYSTAL
C1, C2 = 47 PF TYP
F = 8 MHZ



L C
C1, C2 = 22 PF
L = 130μ H
F = 3 MHZ



R C
@ 5V VCC (TYP)
C1 = 33 PF
R = 1 K
F = 6.6 MHZ



EXTERNAL CLOCK

OSCILLATOR CONFIGURATION

7. POWER SUPPLY

The power supply circuit consists of :

1) a dc voltage of 12 - 17V from the battery

2) a diode which acts as a rectifier

3) a regulator which regulates the high voltage to a constant 5V

1) Battery voltage:

The dc sullply from the battery is about 12-17V.Since the microcontroller requires only 5V for its operation, the voltage is regulated to 5V.

2)Rectifier:

The most common single phase rectifiers are

i) Half wave

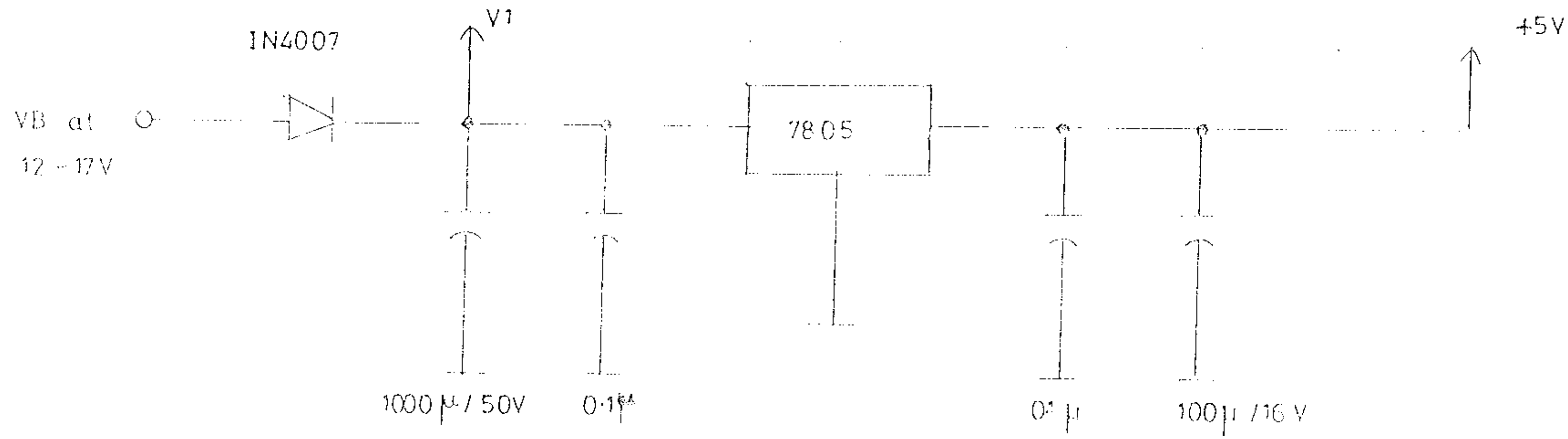
ii) Full wave centre tapped

iii) Full wave bridge

The type of rectifier circuit used in this power supply is a half wave rectifier which consists of a single diode. The advantage of the half wave rectifier is its simplicity.

3) Regulator:

The function of a voltage regulator is to provide a stable DC voltage independent of the load current, temperature and ac voltage variations. Itis in this stage that the battery voltage is reduced to a stable DC voltage of 5



POWER SUPPLY

8. METER COIL

8.1 DRIVE CIRCUITARY

A Drive circuit is used to drive the coil since the pulses cannot be directly given to the coil from the microcontroller.The drive circuit employs an op-amp which is used as a buffer in between the microcontroller and the coil. The Op-amp used is a quadrapule Op-amp_LM324. It consists of four Op-amps in-built. The buffer is also called a voltage follower whose output follows the input.

8.2 COILTHEORY

The processed pulses are given to the microcontroller which generates the PWM according to the speed variations. The PWM output is fed to the coil in terms of sine and cosine waves from the port lines of the microcontroller.

OPERATION:

In a solenoid, the strength of the magnetic field produced is given by

$$H=nI$$

Where, n = number of turns

I = current through the coil

When two coils are placed in right angles with the currents through the coils being Ic and Is, the resultant magnetic fields are

Hc = nIc

Hs = nIs

The magnitude of the resultant magnetic field, H is given by the vector sum of Hc and Hs.

$$[H] = \sqrt{Hc^2 + Hs^2}$$

The angle of the magnetic field is given by H = arc tan (Hs/Hc)

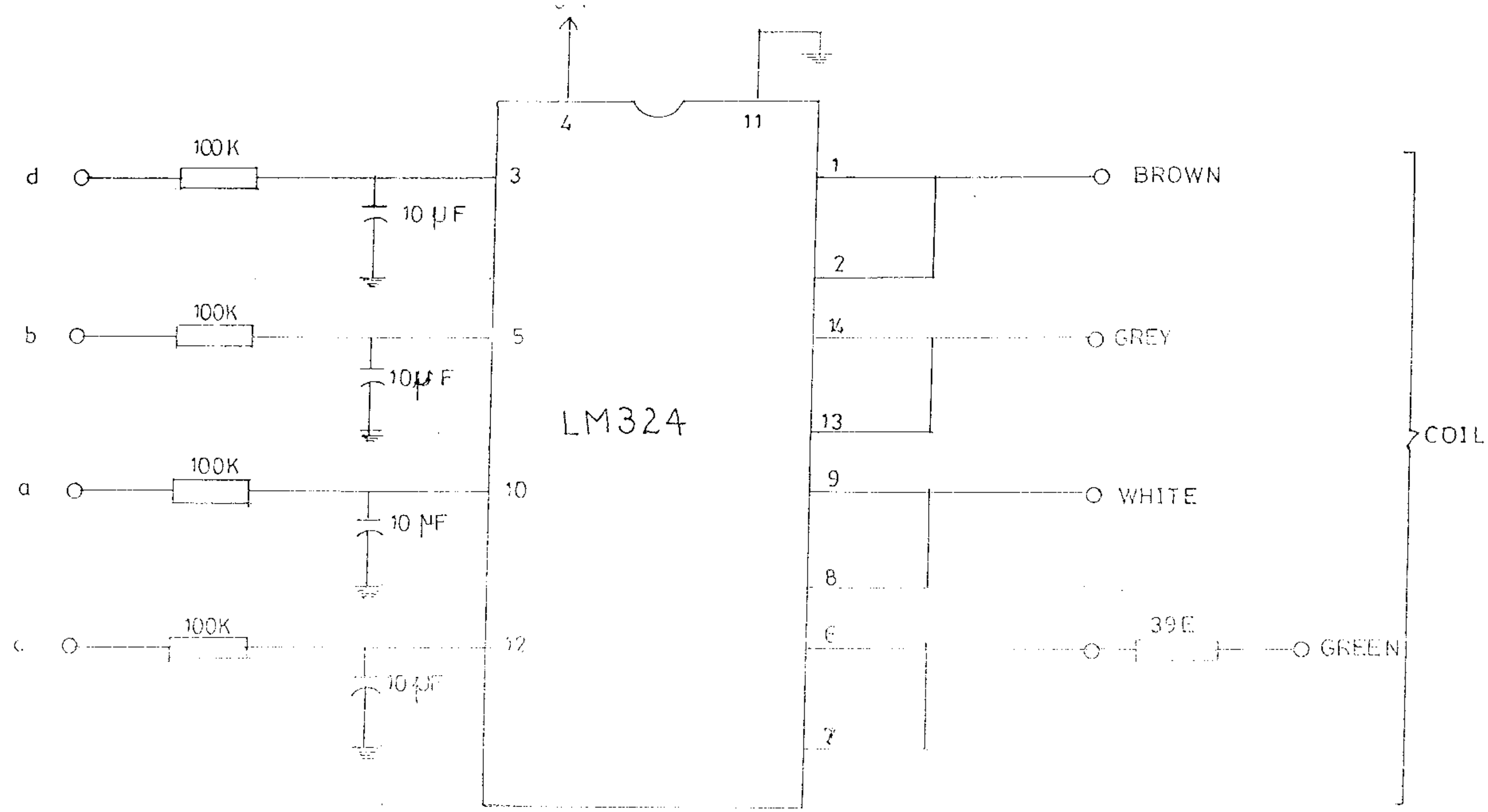
The currents Ic and Is are obtained from the relationship as follows:

$$I_C = I\cos\left(\frac{360}{1024}code\right)$$
$$I_S = I\sin\left(\frac{360}{1024}code\right)$$

2 pow 10 bits are used for resolution of 360 degrees and the code is the input to the coil from the microcontroller.

$$[H] = nI$$
$$\angle H = \tan^{-1}\left(\frac{\sin\theta}{\cos\theta}\right) = \theta, \quad where\,\theta = \frac{360}{1024}code$$

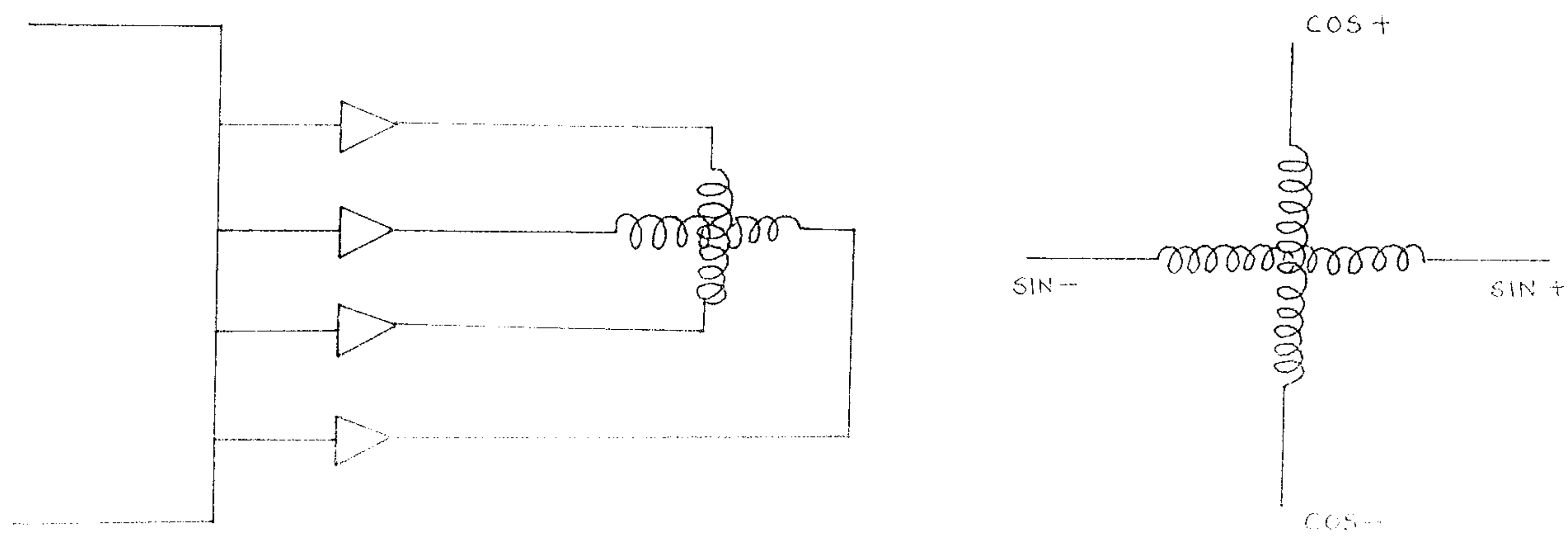
It is clear that the torque which is dependent on modH is a constant if I is a constant and theta is linearly dependent upon the input code.



DRIVE CIRCUIT

INTERFACING COIL TO THE PORT LINES

Coil is interfaced to the microcontroller through buffers. When maximum current flows through the sine coil, zero current is passed through the cos coil and the pointer of the meter shows zero deflection. As the current through the sine coil is decreased and the current through the cos coil is increased, the pointer deflects through an angle. For angles of deflection above 45 degrees maximum current is passed through the cos coil and minimum current through the sine coil. For deflections above 90 degrees, the currents through the port lines are reversed and the same process is repeated.



INTERFACING COIL TO THE PORT LINES

9. ODOMETER

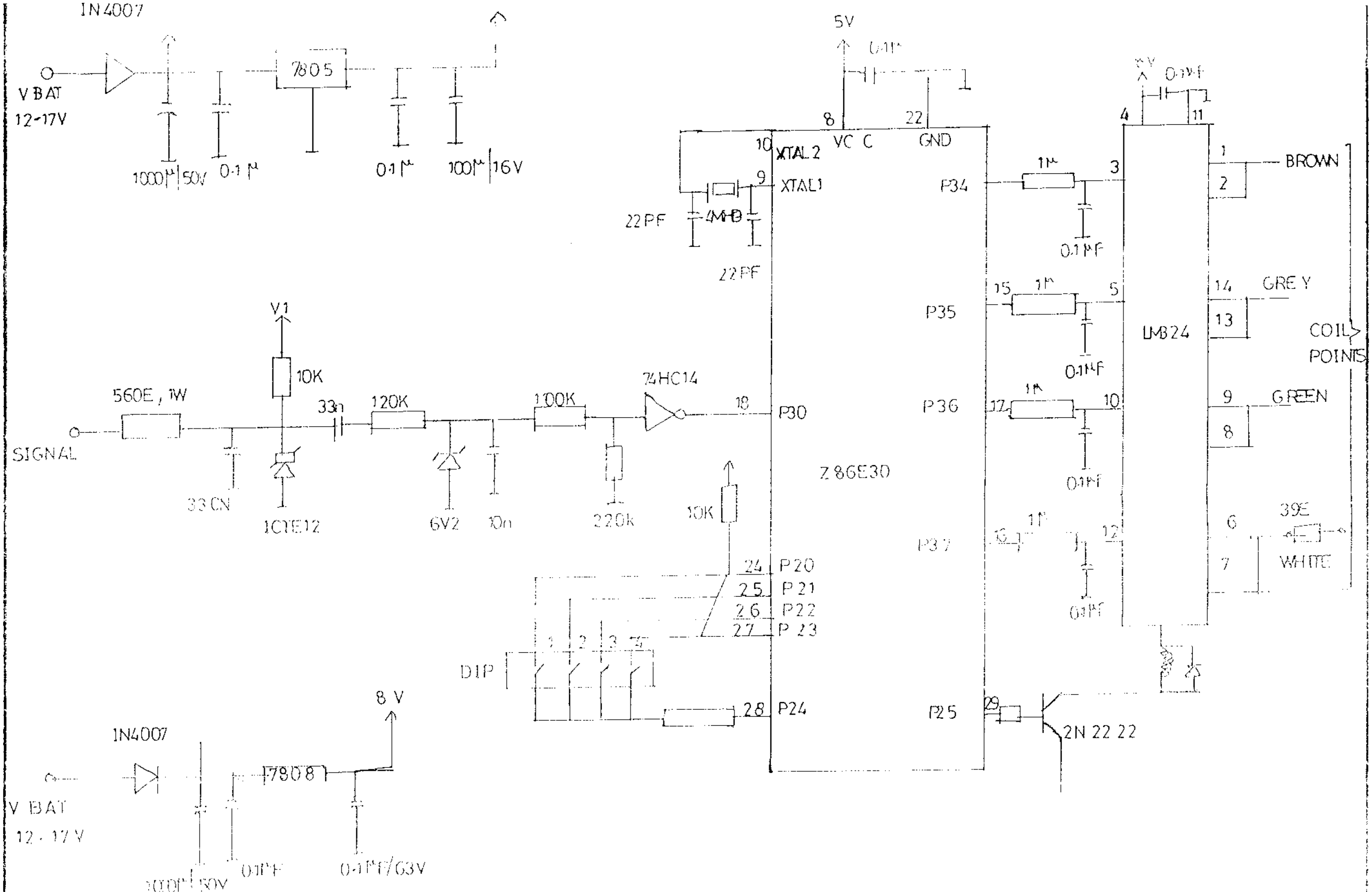
Odometer displays the distance travelled in kilometres. To update the odometer reading and display a counter is designed using electronic components.

PRINCIPLE OF ODOMETER:

It is based on the number of rotations of the tyre. For each rotation of the tyre the reading increases by one. The use of the fraction part is to increase the accuracy of the system.

OPERATION:

The circuit consists of a differentiator diode and a transistor. When there is a pulse from the output of the microcontroller port pin, it is differentiated and is shifted by 0.7V by the diode. This shifted spike is fed to the base of the transistor. The collector of the transistor which is connected to one end of the odo coil is low. Since the other end of the odo coil is connected to +8V, the reading of the odometer is incremented by 0.1 Km.



HARDWARE DIAGRAM

10. SPEEDOMETER

10.2. ALGORITHM

Main routine:

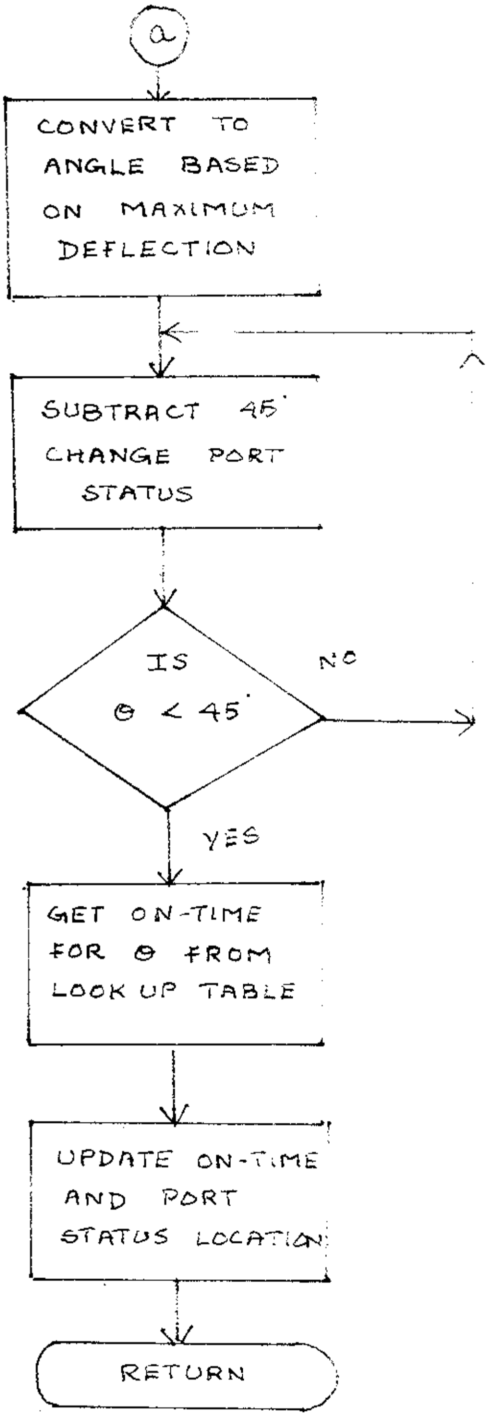
1. Check the pulse available.
2. Get the frequency (call frequency.sub)
3. Is frequency changed from previous frequency
4. If frequency change, call frequency sub.
5. No frequency change, return.

Frequency subroutine:

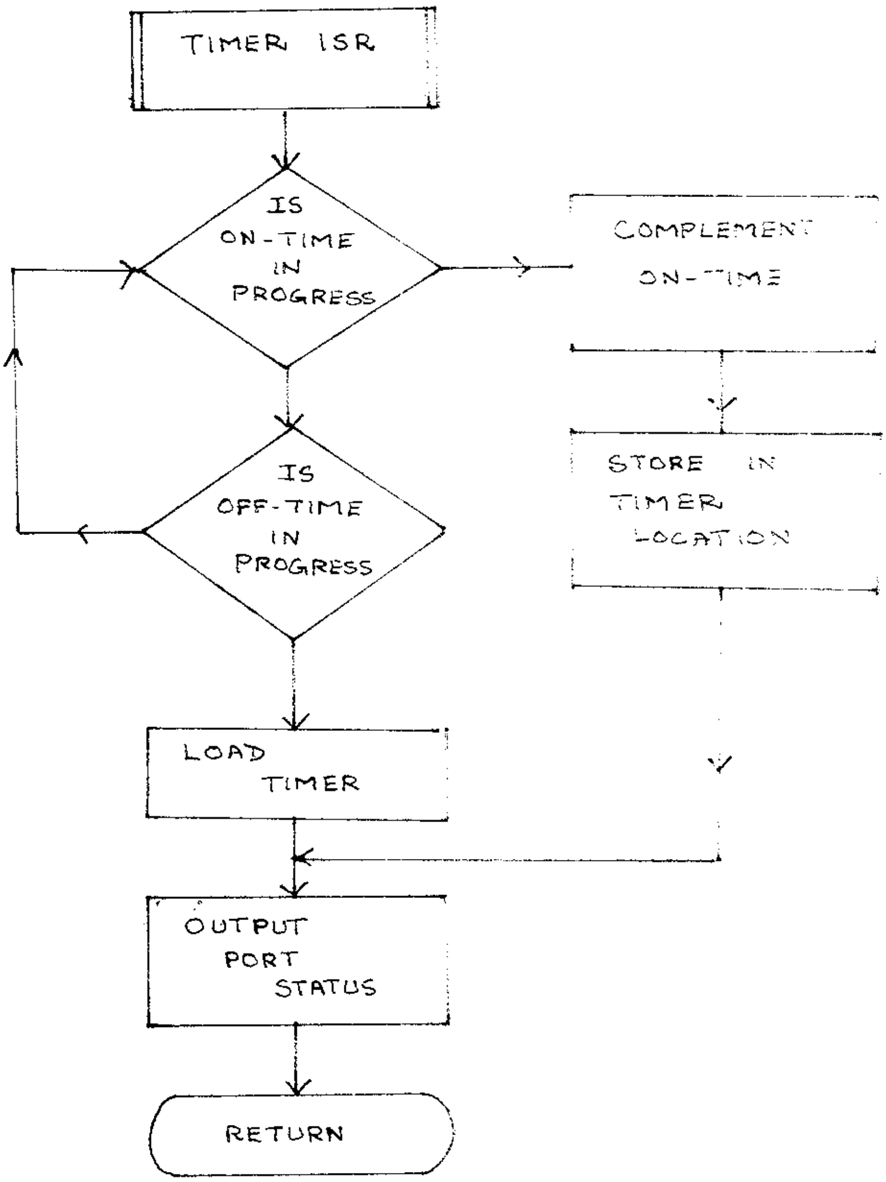
1. Get frequency
2. Get kmph using look up table
3. Scale kmph for 0 - 255 range
4. Convert to angle based on max. deflection used
5. Subtract 45 degrees and change port status
6. Is angle < 45
7. If theta >45 degrees go to 5.
8. If theta <45 degrees , get on time for theta
9. Update ontime location and port status location
10. Return

Timer ISR:

1. Check whether on time or off time is in progress
2. Get the ontime, if it is in progress and complement and store in timer location
3. If offtime is in progress, load timer.
4. Output port status
5. Return.



a
CONVERT TO ANGLE BASED ON MAXIMUM DEFLECTION
SUBTRACT 45°
CHANGE PORT STATUS
IS θ < 45°
NO
YES
GET ON-TIME FOR θ FROM LOOK UP TABLE
UPDATE ON-TIME AND PORT STATUS LOCATION
RETURN


TIMER ISR
IS ON-TIME IN PROGRESS
COMPLEMENT ON-TIME
STORE IN TIMER LOCATION
IS OFF-TIME IN PROGRESS
LOAD TIMER
OUTPUT PORT STATUS
RETURN

10.4 SOFTWARE LISTING FOR SPEEDOMETER

```
start:
          nop
          nop
          nop

initialise:                                        ;initialize funct.reg.

        ld      spl ,#7fh
        ld      p3m ,#00000001b
        ld      p2m ,#0
        and     p2,#0
        ld      tmr,#2ch
        ld      pre1,00010111b
        ld      ipr,00001100b
        ld      imr,#10100000b
        ld      irq  ,#00000000b
        srp     #0


main_loop:
        call    freq_calc
        call    freq_angle
        call    pnt_mov
        call    odo_on
        jr      main_loop


freq_calc:
        tm      bit_ctr , #8
        jr      z , freq_calc_ret
        ld      divide_ctr+1 , #0d0h
        ld      divide_ctr , #7
aa:
        sub     divide_ctr+1 , puls_ctr+1
        sbc     divide_ctr , puls_ctr
        jr      c , freq_calc_ret
        inc     new_freq
        cp      new_freq , old_freq
        jr      z , bb
        ld      old_freq , new_freq
        or      bit_ctr , #1
bb:
        and     bit_ctr , #0f7h

freq_calc_ret:
        ret


freq_angle:
        tm      bit_ctr , #1
        jr      z , freq_angle_ret
        and     bit_ctr , #0feh
        ldr     table_reg+1,#low angle_tbl
        ldr     table_reg,#high  angle_tbl
        addr    table_reg+1,rfreq_ctr
        adc     table_reg, #0
        ld      bang+1 , table_reg+1
        ld      bang,#0
        or      bit_ctr , #4
freq_angle_ret:
        ret

odo_on:
        tm      bit_ctr , #2
```

```
        jr      z , odo_on_ret
        and     bit_ctr , #0fdh
        or      p0 , #1
        nop
        nop
        nop
        nop
        and     p0 , #0feh
odo_on_ret:
        ret

pont_mov:
        tm      bit_ctr , #4
        jr      z , pont_mov_ret

ld_angle:
        ld      bang+1 , #50h
        ld      bang,#0
        and     bit,#0feh
        ld      temp_ang+1,bang+1
        ld      temp_ang,bang
        sub     temp_ang+1,#46
        sbc     temp_ang,#0
        jr      nc,aa
        ld      offset+1,bang+1
        ld      offset,bang
        jr      chk_angl
aa:
        add     angle+1,#45
        adc     angle,#0
        ld      temp_ang+1,angle+1
        ld      temp_ang,angle
        sub     temp_ang+1,bang+1
        sbc     temp_ang,bang
        jr      nc,bb
        jr      cc
bb:
        ld      temp_ang+1,angle+1
        ld      temp_ang,angle
        sub     temp_ang+1,bang+1
        sbc     temp_ang,bang
        sub     angle+1,temp_ang+1
        sbc     angle,temp_ang
        or      bit,#1
cc:
        ld      offset+1,angle+1
        ld      offset,angle

chk_angl:
        ld      temp_ang+1,angle+1
        ld      temp_ang,angle
        sub     temp_ang+1,#50
        sbc     temp_ang,#0
        jr      nc,rt
        and     p2,#0fh
        or      p2,#40h
        ld      port_ctr,#20h
        jp      pwm

rt:
        ld      temp_ang+1,angle+1
        ld      temp_ang,angle
        sub     temp_ang+1,#90
        sbc     temp_ang,#0
        jr      nc,rt1
        and     p2,#0fh
```

```
        or      p2,#60h
        ld      port_ctr,#40h
        jp      pwm

rt1:
        ld      temp_ang+1,angle+1
        ld      temp_ang,angle
        sub     temp_ang+1,#135
        sbc     temp_ang,#0
        jr      nc,rt2
        and     p2,#0fh
        or      p2,#0a0h
        ld      port_ctr,#80h
        jr      pwm

rt2:
        ld      temp_ang+1,angle+1
        ld      temp_ang,angle
        sub     temp_ang+1,#180
        sbc     temp_ang,#0
        jr      nc,rt3
        and     p2,#0fh
        or      p2,#80h
        ld      port_ctr,#20h
        jr      pwm

rt3:
        ld      temp_ang+1,angle+1
        ld      temp_ang,angle
        sub     temp_ang+1,#230
        sbc     temp_ang,#0
        jr      nc,rt4
        and     p2,#0fh
        or      p2,#90h
        ld      port_ctr,#10h
        jr      pwm

rt4:
        ld      temp_ang+1,angle+1
        ld      temp_ang,angle
        sub     temp_ang+1,#0fh
        sbc     temp_ang,#1
        jr      c,rt5
        ld      offset+1,#0eh
        ld      offset,#1

rt5:
        and     p2,#0fh
        or      p2,#90h
        ld      port_ctr,#80h

pwm:
        ldr     table_reg+1,#low ontime_tbl
        ldr     table_reg,#high ontime_tbl
        addr    table_reg+1,roffset+1
        adc     table_reg,offset
        ldc     rpwm_timer,@rrtable_reg
        ld      t1,pwm_timer
        and     bit_ctr , #0fbh
        ei
pont_mov_ret:
        ret


timer_int0:
        incw    puls ctr
```

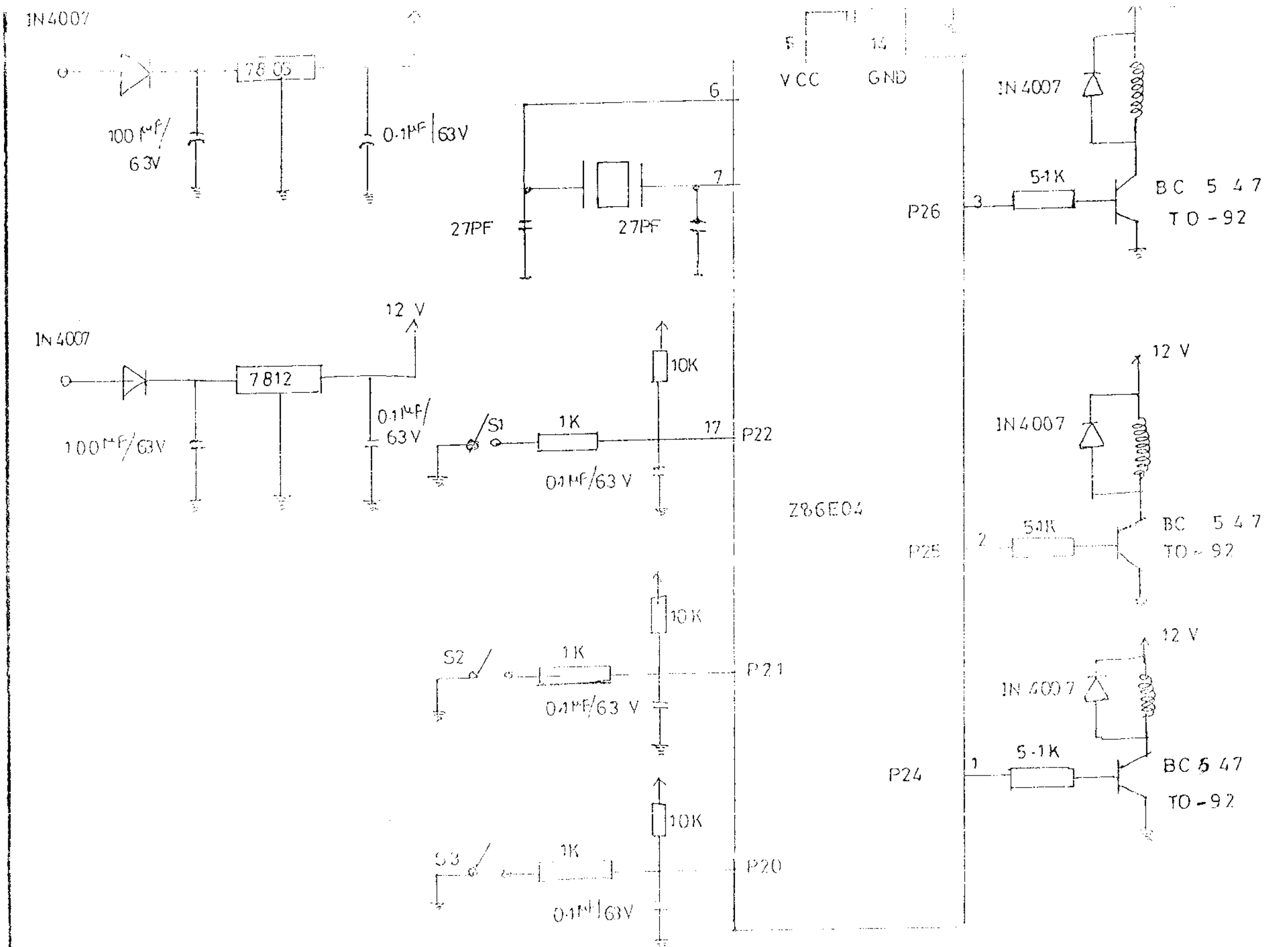
```
timer_ret0:
        iret


timer_int1:
        xor     p2,port_ctr
        com     pwm_timer
        ld      t1,pwm_timer
timer_ret1:
        iret

ext_int:
        di
        and     imr , #0feh
        ld      temp_puls+1 , puls_ctr+1
        ld      temp_puls , puls_ctr
        clr     puls_ctr+1
        clr     puls_ctr
        or      bit_ctr , #8
        inc     odo_ctr
        cp      odo_ctr , #127
        jr      nz , ext_int_ret
        clr     odo_ctr
        or      bit_ctr , #2
ext_int_ret:
        or      imr , #1
        iret


ontime_tbl:

dfb     004h , 007h , 010h , 015h , 019h , 01fh , 024h , 027h , 02bh , 030
dfb     035h , 038h , 03ah , 03fh , 044h , 048h , 04bh , 050h , 053h , 057
dfb     05ch , 05fh , 062h , 066h , 069h , 06ch , 070h , 075h , 078h , 07b
dfb     07fh , 083h , 086h , 08bh , 08fh , 093h , 097h , 09ah , 09dh , 0a2
dfb     0a6h , 0a9h , 0ach , 0b0h , 0b3h , 0b6h , 0bah , 0bdh , 0bfh , 0c2
dfb     0c5h , 0c7h , 0c8h , 0cah , 0cch , 0cfh , 0d2h , 0d5h , 0d7h , 0d9
dfb     0dbh , 0dfh , 0e2h , 0e5h , 0e7h , 0e9h , 0ebh , 0edh , 0f0h , 0f3
dfb     0f5h , 0f7h , 0f9h , 0fbh , 0ffh


angle_tbl:

dfb     001h,   002h,   003h,   004h,   005h,   006h,   007h,   008h,   00
dfb     00Ah,   00Bh,   00Ch,   00Dh,   00Eh,   00Fh,   010h,   011h,   01
dfb     013h,   014h,   015h,   016h,   017h,   018h,   019h,   01Ah,   01
dfb     01Ch,   01Dh,   01Eh,   01Fh,   020h,   021h,   022h,   023h,   02
dfb     025h,   026h,   027h,   028h,   029h,   02Ah,   02Bh,   02Ch,   02
dfb     02Eh,   02Fh,   030h,   031h,   032h,   033h,   034h,   035h,   03
dfb     037h,   038h,   039h,   03Ah,   03Bh,   03Ch,   03Dh,   03Eh,   03
dfb     040h,   041h,   042h,   043h,   044h,   045h,   046h,   047h,   04
dfb     049h,   04Ah,   04Bh,   04Ch,   04Dh,   04Eh,   04Fh,   050h,   05
dfb     052h,   053h,   054h,   055h,   056h,   057h,   058h,   059h,   05
```

11. TEST JIG

Test jig is a test procedure to know how the meter would perform in practical running conditions subjected to mechanical stress.

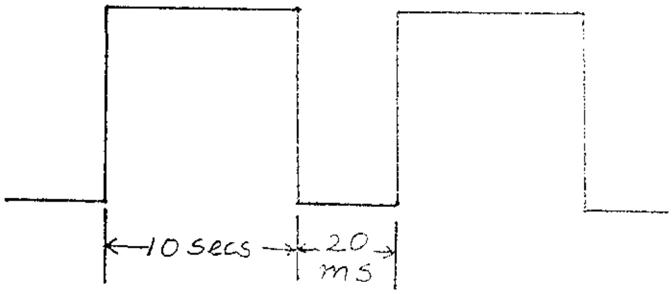
OBJECTIVES OF TEST JIG:

1) It should have an accurate method for testing the performance of meters.
2) Test the performance of meters for varying input signals that would prevail in practical conditions.

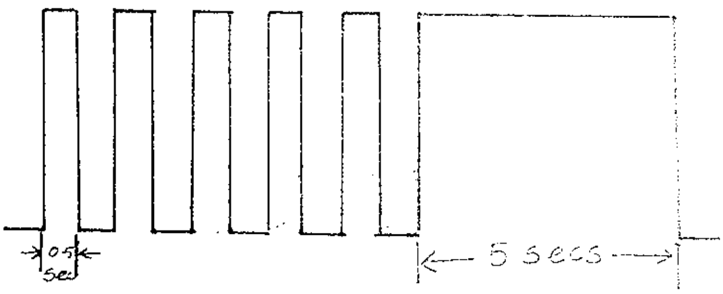


CIRCUIT DIAGRAM OF TESTING JIG

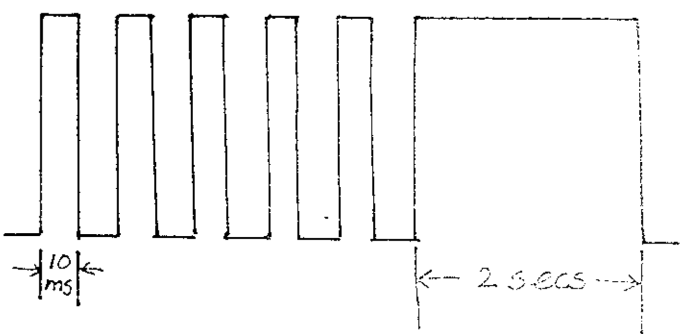
1) POWER SUPPLY WITH INSTANTANEOUS CUT-OFF :-



2) IGNITION KEY SWITCH INTERMITTENT TEST :-



3) SUPPLY VOLTAGE INTERMITTENT TEST :-



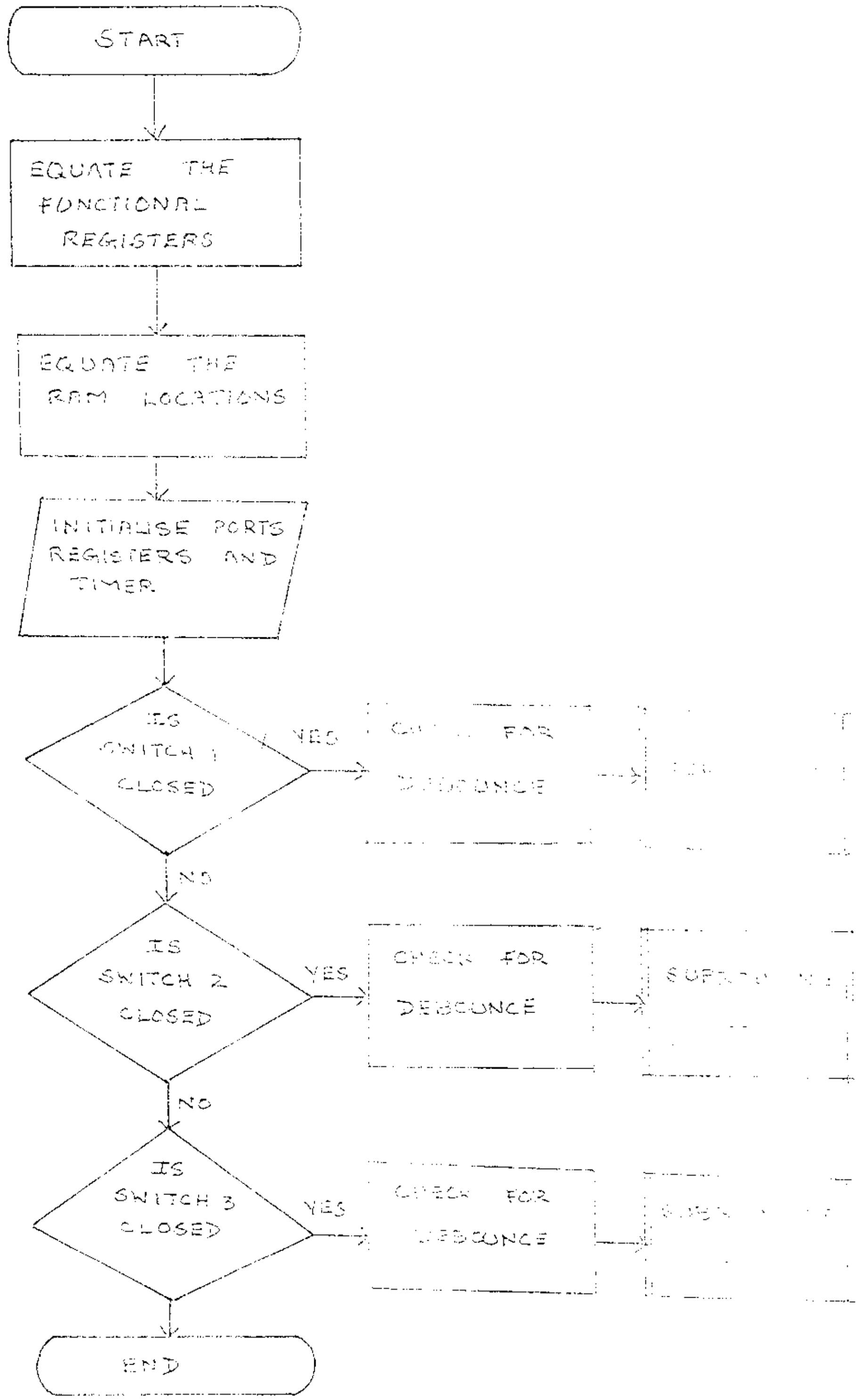
11.2. ALGORITHM

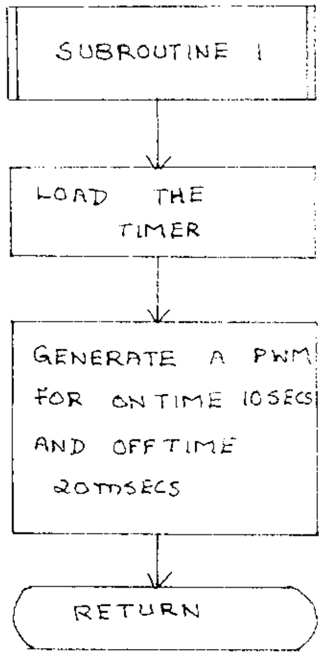
* Initialize the ports, timer and interrupt registers
* Test if any of the switches are closed and check for debounce
* If any of the switches are closed then go to their respective subroutine

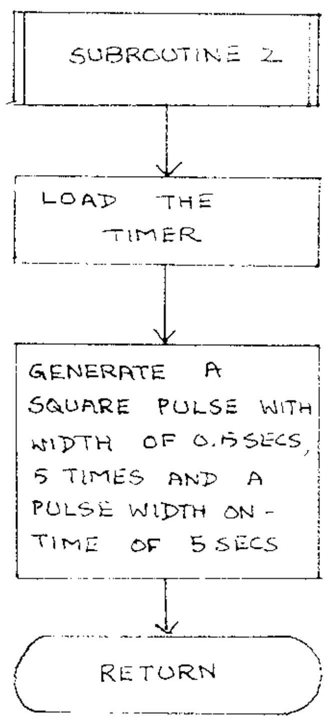
SUBROUTINE:

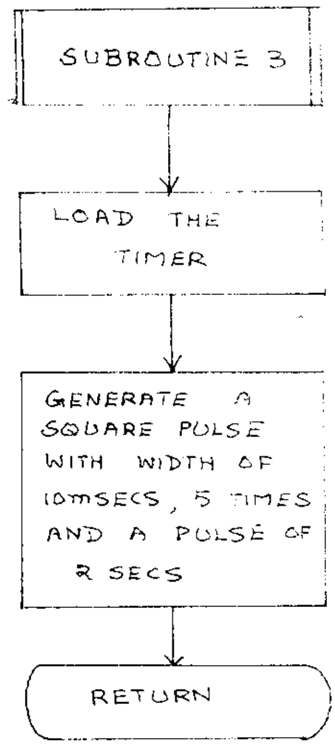
* The prescaler and the tilmer is loaded with a particular value corresponding to the width of the pulse for each test.
* When the down counter of both the prescaler and timer reduces to zero interrupt is generated.

FLOWCHART


START
EQUATE THE FUNCTIONAL REGISTERS
EQUATE THE RAM LOCATIONS
INITIALISE PORTS REGISTERS AND TIMER
IS SWITCH 1 CLOSED
YES
NO
IS SWITCH 2 CLOSED
YES
CHECK FOR DEBOUNCE
NO
IS SWITCH 3 CLOSED
YES
CHECK FOR DEBOUNCE
END


SUBROUTINE 1
LOAD THE TIMER
GENERATE A PWM FOR ON TIME 10SECS AND OFF TIME 20mSECS
RETURN


SUBROUTINE 2
LOAD THE TIMER
GENERATE A SQUARE PULSE WITH WIDTH OF 0.5SECS, 5 TIMES AND A PULSE WIDTH ON-TIME OF 5 SECS
RETURN


SUBROUTINE 3
LOAD THE TIMER
GENERATE A SQUARE PULSE WITH WIDTH OF 10mSECS, 5 TIMES AND A PULSE OF 2 SECS
RETURN

```
initialise:
                ld SPL,#90H
                ld P3M,#01H
                ld P2M,#0FH
                ld TMR,#3CH
                ld P01M,#00H
                ld IPR,#0AH
                ld IRQ,#00H
                ld IMR,#0A0H
                ld RP,#10H
                ld PRE1,#50          ;for every 10msecs
                ld T1,#200
START1:         IRET
main:
                TM P2 ,#01H
                jr nz,debounce0
                TM P2 ,#02H
                jr nz,debounce1
                TM P2 ,#04H
                jr nz,debounce2
                jr main
debounce0:
                clr deb_reg
                ld ctr ,#00H
routine:
                dec deb_reg
                jr nz,routine
                inc ctr
                cp ctr,#10H
                jr nz,routine
                TM P2,#01H
                jr nz,subroutine0
                jr main
debounce1:
                clr deb_reg
                ld ctr ,#00H
rout:
                dec deb_reg
                jr nz,rout
                inc ctr
                cp ctr,#10H
                jr nz,rout
                TM P2,#02H
                jr nz,subroutine1
                jr main
debounce2:
                clr deb_reg
                ld ctr ,#00H
rt:
                dec deb_reg
                jr nz,rt
                inc ctr
                cp ctr,#10H
                jr nz,rt
                TM P2,#04H
                jr nz,subroutine2
                jr main
subroutine0:
                ld on_ctr,#100
                ld off_ctr,#02
                or p2,#01
                and bit,#0f9h
subroutine1:
```

```
                ld short_on_ctr ,#100
                ld short_off_ctr,#100
                ld long_on_ctr,#250
                or p2,#02
                and bit,#0fah
subroutine2:
                ld s_on_ctr ,#1
                ld s_off_ctr,#1
                ld l_on_ctr,#200
                or p2,#04
                and bit,#0fch
timer_int:
                tm bit,#01
                jr nz, loop1
                tm bit,#02
                jr nz, loop2
                tm bit,#04
                jp nz, loop3

loop1:          inc ctr2
                cp ctr2,#5
                jp nz,timer_ret
                tm bit,#01
                jr nz,off_loop
                inc temp_on_ctr
                cp temp_on_ctr,on_ctr
                jp nz,timer_ret
                clr temp_on_ctr
                and p2,#0feh
                or bit,#01

off_loop:       inc temp_off_ctr
                cp temp_off_ctr,off_ctr
                jp nz,timer_ret
                clr temp_off_ctr
                or p2,#01
                and bit,#0feh
                jp timer_ret

loop21:         inc ctr3
                cp ctr3,#02
                ld ctr0,#10h
                ld sh_ctr,#05h

loop2:          tm bit,#02h
                jr nz,off_loop2
                inc temp_short_on_ctr
                cp temp_short_on_ctr,short_on_ctr
                jp nz,timer_ret
                clr temp_short_on_ctr
                and p2,#0fdh
                or bit,#02

off_loop2:      inc temp_short_off_ctr
                cp temp_short_off_ctr,short_off_ctr
                jp nz,timer_ret
                clr temp_short_off_ctr
                or p2,#02
                and bit,#0fdh
                dec sh_ctr
                jr nz, loop2
                tm bit,#02
                inc temp_long_on_ctr
                cp temp_long_on_ctr,long_on_ctr
                jp nz,timer_ret
                clr temp_long_on_ctr
```

```
                and bit,#0fdh
                or p2,#02
                dec ctr
                jr nz,loop2
                jp timer_ret

loop31:         ld ctrl,#10h
                ld sh_ctrl,#05h

loop3:          tm bit,#04h
                jr nz,off_loop3
                inc temp_s_on_ctr
                cp temp_s_on_ctr,s_on_ctr
                jp nz,timer_ret
                clr temp_s_on_ctr
                and p2,#0fbh
                or bit,#04h

off_loop3:      inc temp_s_off_ctr
                cp temp_s_off_ctr,s_off_ctr
                jp nz,timer_ret
                clr temp_s_off_ctr
                or p2,#04h
                and bit,#0fbh
                dec sh_ctrl
                jp nz,loop3
                tm bit,#04h
                inc temp_l_on_ctr
                cp temp_l_on_ctr,l_on_ctr
                jp nz,timer_ret
                clr temp_l_on_ctr
                and bit,#0fbh
                or p2,#04h
                dec ctrl
                jp nz,loop3
                jp timer_ret

timer_ret:
                iret
```

12. EMULATOR

Specification:

Maximum emulation speed 8MHz

Minimum emulation speed 1 MHz

Power requirements +8V Vdc at 0.5 A

Dimensions

Width 7.0 in

Length 9.0 in

Height 0.9 in

Serial inter face - RS232C at 19200 baud

KIT CONTENTS:

Z8 CCP emulator

CMOS Z86C9320VSC

RS232C interface

Reset switch

20 MHz CMOS Z86C5020FSE ICE chip

8K x 8 static RAM (for code memory)

18-pin DIP ZIF Programming socket

18-pin target connector cable

Holes available for 28/40 pin ZIF sockets

Sockets available for 18/28/40-Pin target cables

Software (IBM PC platform)

ZASM cross-assembler and MOBJ object file Util

Z8 GUI Emulator s/w

Production Languages Corporation COMPASS/Z8

SYSTEM REQUIREMENTS:

386 or 486 IBM compatible PC

VGA Video adapter

20 MHz, minimum

4 Mbytes RAM

Microsoft windows 3.0 or 3.1

Hard disk drive (1Mbyte free space)

High density(HD) floppy disk drive(3.5 inch)

RS-232 COM PORT

13. SPECIFICATIONS

SPEEDOMETER

Ratio	1 : 1
Range	0-140 Kmph
Type of Indication Analog Type	
Maximum pointer deflection	- 260 degrees
Type of mechanical system	- Cross Coil NRZ Type
Sensor	- Reed Switch
Amplitude of signal	- 12 v
Accuracy	- ± 2 Kmph

ODO METER

Display Type	- Impulse Counter
Maximum distance	- 999999.9 Max
Resolution	- 0.1 Km
Accuracy	- ± 0.5%.

14. ADVANTAGES

ADVANTAGES OF USING ELECTRONICS:

- Low cost
- High reliability
- Improvement in efficiency

ADVANTAGES OF USING CROSS COILS IN METERS:

- There are no springs
- Accuracy does not degrade with age
- Not prone to damage due to vibrations or handling
- Built in damping.

15. CONCLUSION

Our project 'ELECTRONIC DASH BOARD - SPEEDOMETER' is designed for use in all vehicles. The Electronic Dash Board is a low cost device suitable for Indian conditions. The values of the inputs indicated are quite accurate and the error in indication does not exceed ± 3%. The Electronic Dash Board is designed in such a way that facilities available at present can be expanded.

Improvements that can be made for future use:

- LCD display can be used to display the distance travelled.
- The speedometer indication can be made using LCD because of which the needle and coil is replaced by Electronic Display.
- Vacuum Fluorescent Tube (VFT) can be used for Digital and Illumination purposes.

BIBLIOGRAPHY

1. ZILOG MICROCONTROLLER DATA BOOK

2. OKI '96 DATA BOOK FOR REED SWITCH / REED RELAY.

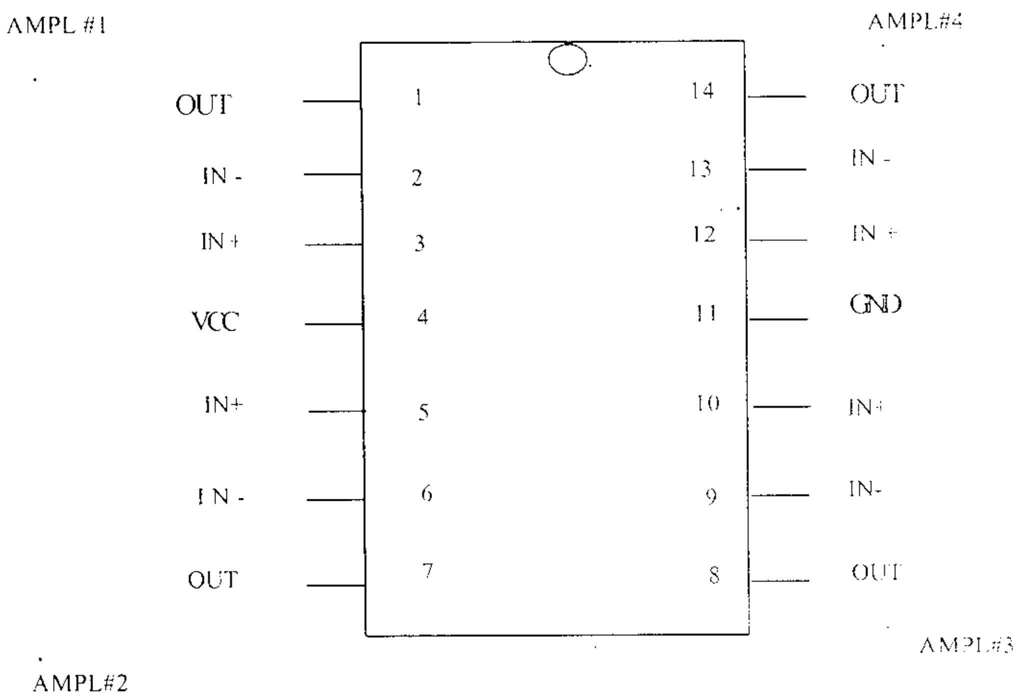
3. LINEAR IC DATA BOOK - NATIONAL SEMICONDUCTOR

4. AUTOMOTIVE MECHANICS - JOSEPH HEITNER.

5. MICROPROCESSOR AND APPLICATIONS - GAONKAR.

LM 324 QUADRUPLE OPERATIONAL AMPLIFIER

- Wide range of supply voltages: Single supply 3V to 30V.
- Low supply current drain independent of supply voltage 0.8mA.



* Common _mode input voltage range includes ground allowing direct sensing near ground.

* Low input bias and offset parameters:

input offset voltage 3mv typ

A versions	2mv typ
input offset current	2nA typ
input bias current	20nA typ
A versions	15nA typ

* Differential input voltage range equal to maximum rated supply voltage 32V
* open loop differential voltage amplification 100mV typ
* internal frequency compensation.

DESCRIPTION:

These devices consists of 4 independent, high - frequency compensated op-amps that were designed. Specifically to operate from a single supply over a wide range of voltages. Operation from split supplies is also possible so long as the difference between the two supplies is 3V to 30V,and Vccis atleast 1.5V more positive than the input common mode voltage.The supply current drain is independent of the magnitude of supply voltage.

Applications include transducer amplifiers, d-c amplification blocks and all the conventional amlifier circuits that now can be more easily implemented in single supply voltage systems.

FUNCTIONAL DESCRIPTION (Continued)

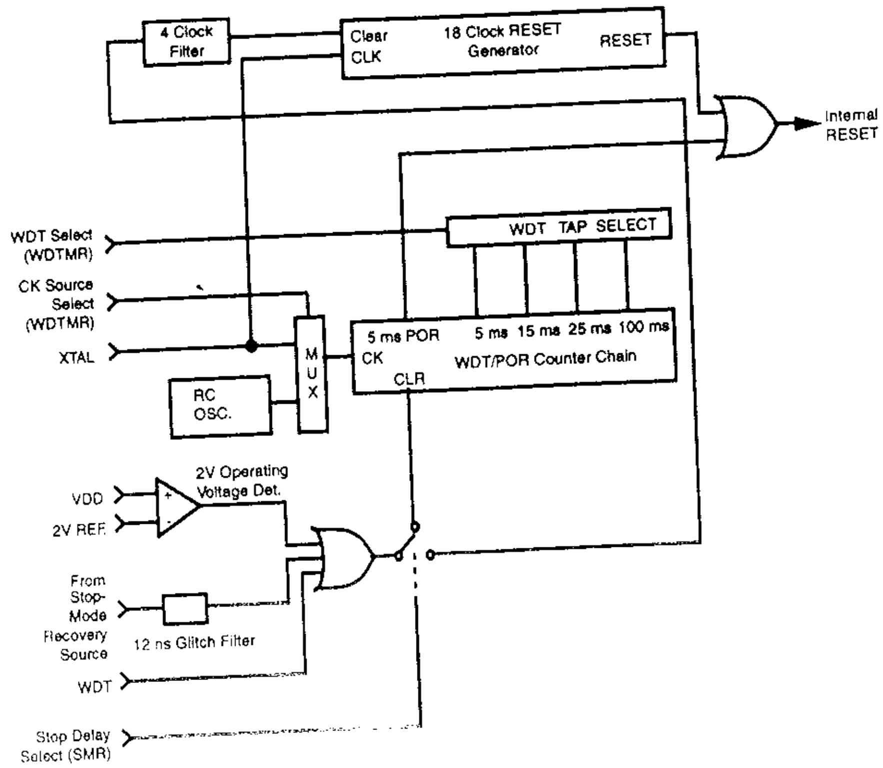


Figure 19. Resets and WDT

Auto Reset Voltage. An on-board Voltage Comparator checks that V_{CC} is at the required level to ensure correct operation of the device. Reset is globally driven if V_{CC} is below V_{RST} (Auto Reset Voltage - Figure 20).

If the V_{CC} drops below 4.5V while the device is in operation, the device must be powered down and then re-powered up again.

Note: V_{CC} must be in the allowed operating range (4.5V to 5.5V) prior to the minimum Power-On Reset time-out (T_{POR}).

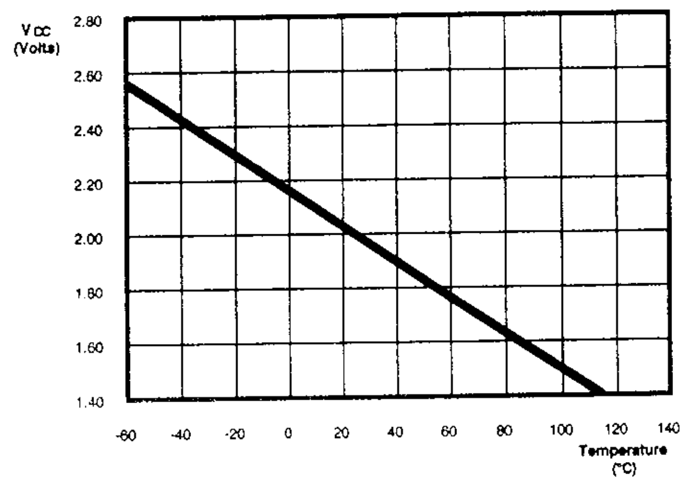


Figure 20. Typical Z86E30/31 VRST Voltage vs Temperature

FUNCTIONAL DESCRIPTION (Continued)

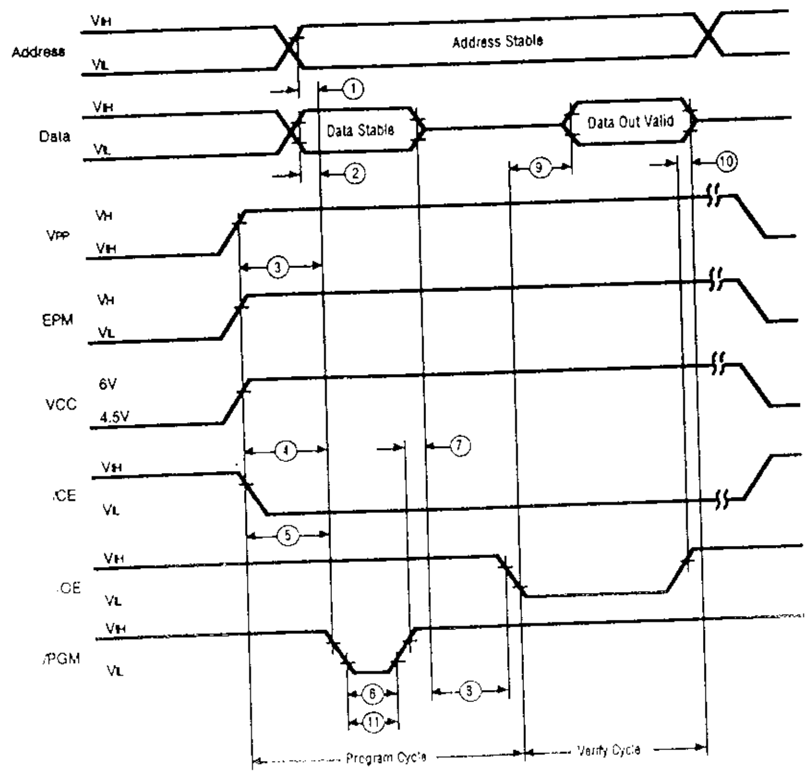


Figure 22. Timing Diagram of EPROM Program and Verify Modes

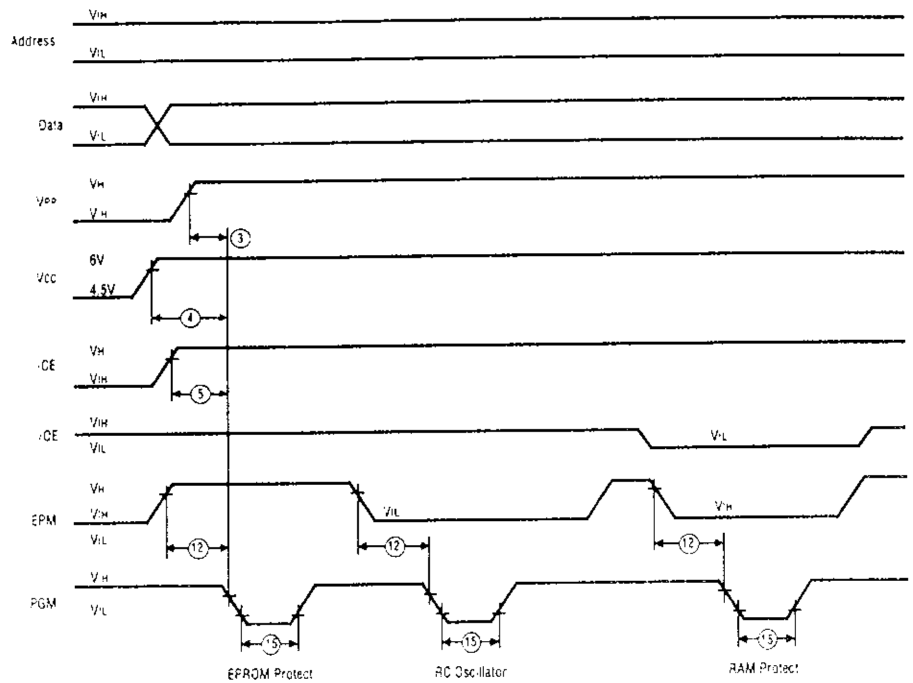


Figure 23. Timing Diagram of EPROM Protect, RAM Protect, and RC OSC Modes

DC ELECTRICAL CHARACTERISTICS (Continued)

Symbol	Parameter	V_{CC} Note[3]	$T_A = 0°C$ to $+70°C$ Min	Max	Typical at 25°C	Units	Conditions	Notes
I_{CC1}	Standby Current (Standard Mode)	5.0V		6.0	3.5	mA	HALT mode V_{IN} = 0V, V_{CC}; V_{CC} @ 8 MHz	[4,5]
		5.0V		3.0	1.50	mA	Clock Divide-by-16 @ 8 MHz	[4,5]
I_{CC}	Supply Current (SCLK/TCLK = XTAL)	5.0V		7.5	5.0	mA	@ 2 MHz	[4,5,10]
		5.0V		12.0	8.0	mA	@ 4 MHz	[4,5,10]
I_{CC1}	Standby Current (SCLK/TCLK = XTAL)	5.0V		2.0	1.0	mA	@ 2 MHz	[4,5,10]
		5.0V		3.0	1.5	mA	@ 4 MHz	[4,5,10]
I_{CC1}	(Standard Mode)	5.0V		2.0	0.75	mA	Clock Divide-by-16 @ 2 MHz	[4,5]
		5.0V		2.0	1.0	mA	Clock Divide-by-16 @ 4 MHz	[4,5]
I_{CC2}	Standby Current	5.0V		10	2	μA	STOP mode V_{IN} = 0V, V_{CC} WDT is not Running	[6]
		5.0V		800	450	μA	STOP mode V_{IN} = 0V, V_{CC} WDT is Running	[6]
I_{ALL}	Auto Latch Low Current	5.0V		−10	−5	μA	$0V < V_{IN} < V_{CC}$	
I_{ALH}	Auto Latch High Current	5.0V		20	10	μA	$0V < V_{IN} < V_{CC}$	
T_{POR}	Power-On Reset	5.0V	2.5		4.5	ms		[3]
V_{RST}	Auto Reset Voltage			3.0	2.6	V		

Notes:

[1] I_{CC}

	Typ	Max	Unit	Freq
Clock-driven XTAL	3.3 mA	6.0	mA	8 MHz
Crystal or resonator	3.5 mA	6.0	mA	8 MHz

[2] V_{SS} = 0V = GND

[3] V_{CC} must be in the allowed operating range (4.5V to 5.5V) prior to the minimum T_{POR} timeout. V_{CC} specified at 4.5V to 5.5V

[4] All outputs unloaded, I/O pins floating, inputs at rail

[5] CL1 = CL2 = 100 pF

[6] Same as note [4] except inputs at V_{CC}

[7] Except clock pins and Port 3 input pins in EPROM mode

[8] Port Low EMI mode

[9] Port STD mode

[10] SMR Reg Bit D1 = 1

[11] Z86E30 only

AC ELECTRICAL CHARACTERISTICS

Additional Timing Diagram (Standard Mode for SCLK/TCLK ÷ XTAL/2)

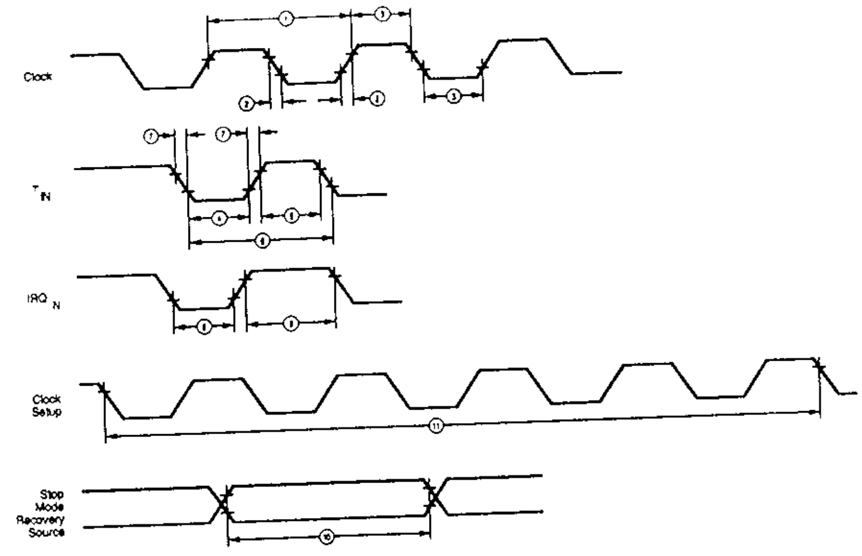


Figure 27. Additional Timing

AC ELECTRICAL CHARACTERISTICS

Additional Timing Table (Standard Mode)

No	Symbol	Parameter	V_{CC} Note[6]	T_A = 0°C to +70°C 8 MHz[11] Min	Max	12 MHz[11] Min	Max	Units	Notes
1	TpC	Input Clock Period	5.0V	125	DC	83.3	DC	ns	[1]
2	TrC,TfC	Clock Input Rise & Fall Times	5.0V		25		15	ns	[1]
3	TwC	Input Clock Width	5.0V	62.5		41.6		ns	[1]
4	TwTinL	Timer Input Low Width	5.0V	70		70		ns	[1]
5	TwTinH	Timer Input High Width	5.0V	5TpC		5TpC			[1]

4.7 IMMEDIATE DATA ADDRESSING (IM)

Immediate data is considered an "addressing mode" for the purposes of this discussion. It is the only addressing mode that does not indicate a register or memory address as the source operand; the operand value used by the instruction is the value supplied in the operand field itself. Because an immediate operand is part of the instruction, it is always located in the program memory address space.

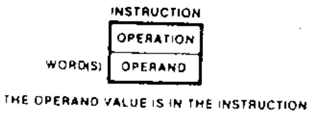


Figure 4-8. Immediate Data Addressing

ZiLOG

Chapter 5 Instruction Set

5.1 FUNCTIONAL SUMMARY

Z8 instructions can be divided functionally into the following eight groups:

- Load
- Arithmetic
- Logical
- Program Control
- Bit Manipulation
- Block Transfer
- Rotate and Shift
- CPU Control

The following summary shows the instructions belonging to each group and the number of operands required for each. The source operand is "src", "dst" is the destination operand, and "cc" is a condition code.

Load Instructions

Mnemonic	Operands	Instruction
CLR	dst	Clear
LD	dst,src	Load
LDC	dst,src	Load Constant
LDE	dst,src	Load External
POP	dst	Pop
PUSH	src	Push

Arithmetic Instructions

Mnemonic	Operands	Instruction
ADC	dst,src	Add With Carry
ADD	dst,src	Add
CP	dst,src	Compare
DA	dst	Decimal Adjust
DEC	dst	Decrement
DECW	dst	Decrement Word
INC	dst	Increment
INCW	dst	Increment Word
SBC	dst,src	Subtract With Carry
SUB	dst,src	Subtract

Logical Instructions

Mnemonic	Operands	Instruction
AND	dst,src	Logical And
COM	dst	Complement
OR	dst,src	Logical Or
XOR	dst,src	Logical Exclusive Or

Program-Control Instructions

Mnemonic	Operands	Instruction
CALL	dst	Call Procedure
DJNZ	r,dst	Decrement and Jump NonO
IRET		Interrupt Return
JP	cc,dst	Jump
JR	cc,dst	Jump Relative
RET		Return

Bit-Manipulation Instructions

Mnemonic	Operands	Instruction
TCM	dst,src	Test Complement Under Mask
TM	dst,src	Test Under Mask
AND	dst,src	Bit Clear
OR	dst,src	Bit Set
XOR	dst,src	Bit Complement

Block-Transfer Instructions

Mnemonic	Operands	Instruction
LDCI	dst,src	Load Constant Auto-increment
LDEI	dst,src	Load External Auto-increment

Rotate and Shift Instructions

Mnemonic	Operands	Instruction
RL	dst	Rotate Left
RLC	dst	Rotate Left Through Carry
RR	dst	Rotate Right
RRC	dst	Rotate Right Through Carry
SRA	dst	Shift Right Arithmetic
SWAP	dst	Swap Nibbles

Z8® CONTROL REGISTER DIAGRAMS

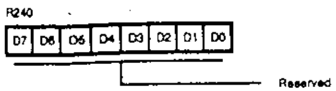


Figure 33. Reserved

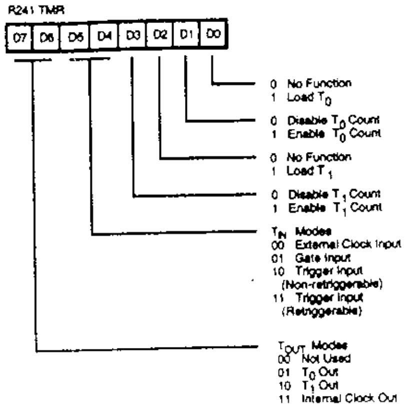


Figure 34. Timer Mode Register (F1H: Read/Write)

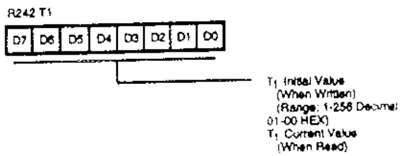


Figure 35. Counter Timer 1 Register (F2H: Read/Write)

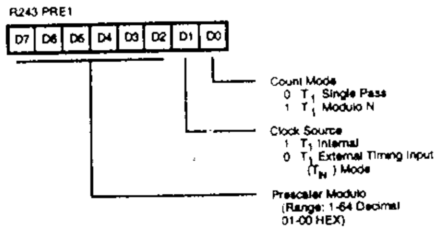


Figure 36. Prescaler 1 Register (F3H: Write Only)

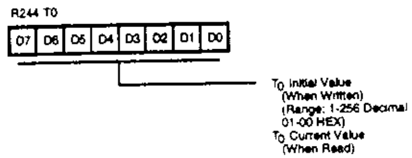


Figure 37. Counter/Timer 0 Register (F4H: Read/Write)

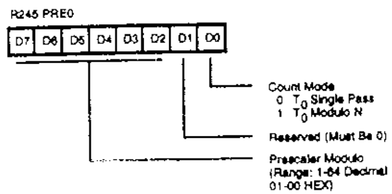


Figure 38. Prescaler 0 Register (F5H: Write Only)

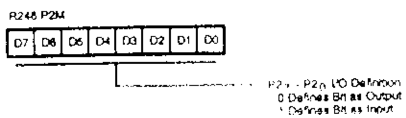


Figure 39. Port 2 Mode Register (F6H: Write Only)

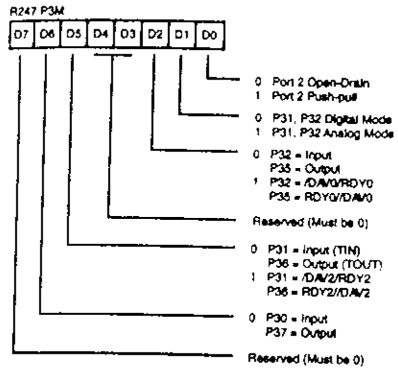


Figure 40. Port 3 Mode Register (F7H: Write Only)

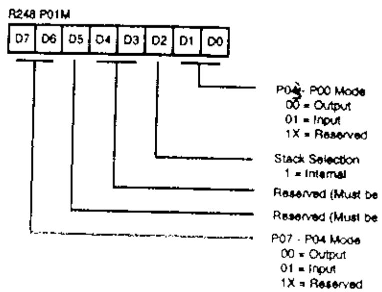


Figure 41. Port 0 and 1 Mode Register (F8H: Write Only)

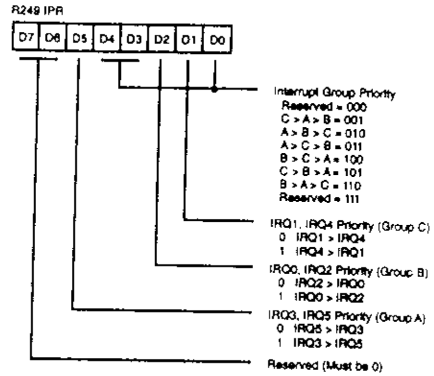


Figure 42. Interrupt Priority Register (F9H: Write Only)

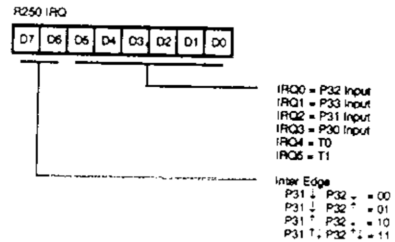


Figure 43. Interrupt Request Register (FAH: Read/Write)

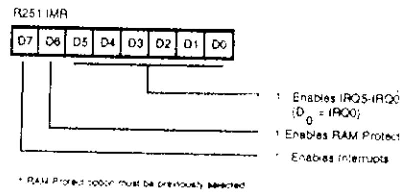


Figure 44. Interrupt Mask Register (FBH: Read/Write)

X	Indexed	reg (Rn): where reg represent a number in the range 0-127, 240-255 and n = 0-15
DA	Direct Address	addrs: where addrs represents a number in the range 0-65,535
RA	Relative Address	addrs: where addrs represents a number in the range +127, -128 which is an offset relative to the address of the next instruction
IM	Immediate	#data: where data is a number between 0 and 255

Additional symbols used are:

Symbol	Meaning
dst	Destination operand
src	Source operand
@	Indirect address prefix
SP	Stack Pointer
PC	Program Counter
FLAGS	Flag register (R252)
RP	Register Pointer (R253)
IMR	Interrupt mask register (251)
#	Immediate operand prefix
%	Hexadecimal number prefix
OPC	Opcode

Assignment of a value is indicated by the symbol "<-". For example,

dst <- dst + src

indicates that the source data is added to the destination data and the result is stored in the destination location. The notation "addr(n)" is used to refer to bit "n" of a given location. For example,

dst (7)

refers to bit 7 of the destination operand.

For proper instruction execution, Z8 PLZ/ASM assembly language syntax requires that "dst, src" be specified, in that order. The following instruction descriptions show the format of the object code produced by the assembler. This binary format should be followed by users who prefer manual program coding or who intend to implement their own assembler.

Example: If the contents of registers %43 and %08 are added and the result stored in %43, the assembly syntax and resulting object code are:

```
ASM:  ADD  %43, %08    (ADD dst, src)
OBJ:  04   08   43     (OPC src, dst)
```

In general, whenever an instruction format requires an 8-bit register address, that address can specify any register location in the range 0-127, 240-255 or a working register R0-R15. If, in the above example, register %08 is a working register, the assembly syntax and resulting object code would be:

```
ASM:  ADD  %43, R8     (ADD dst src)
OBJ:  04   E8   43     (OPC src dst)
```

For a more complete description of assembler syntax refer to the Z8 PLZ/ASM Assembly Language Manual (publication no. 03-3023-03) and ZSCAN 8 User's Tutorial (publication no. 03-8200-01).

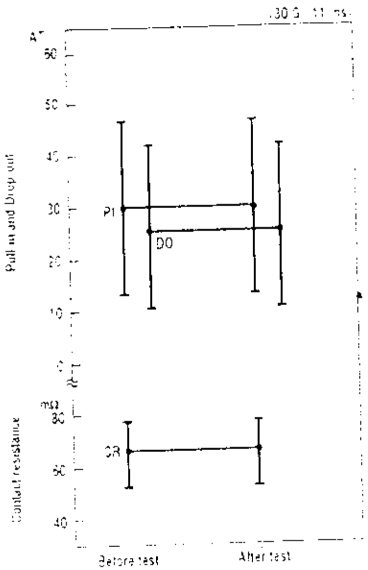
5.4.2 Condition Codes and Flag Settings

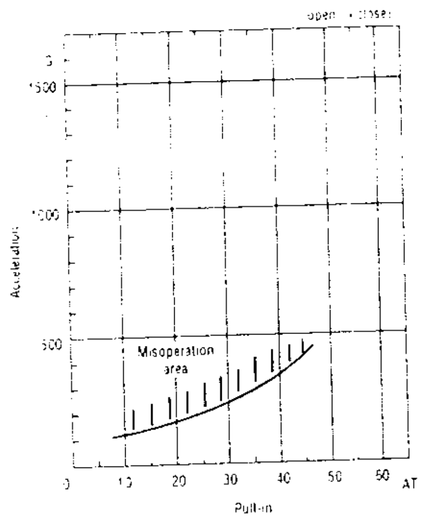
The condition codes and flag settings are summarized in the following tables. Notation for the flags and how they are affected are as follows:

C	Carry flag	0	Cleared to 0
Z	Zero flag	1	Set to 1
S	Sign flag	*	Set or cleared according to operation
V	Overflow flag	-	Unaffected
D	Decimal-adjust flag	X	Undefined
H	Half-carry flag		

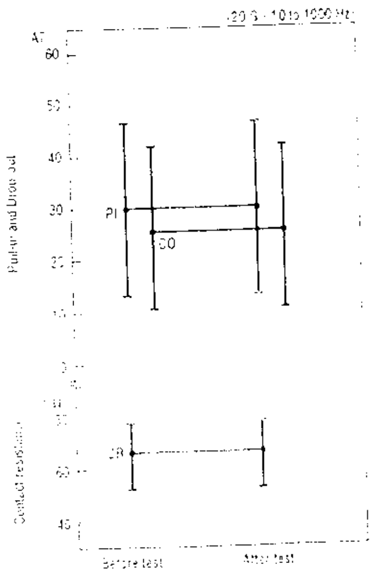
Binary	Mnemonic	Meaning	Flags Settings
0000	F	Always false	-
1000	(blank)	Always true	-
0111	C	Carry	C = 1
1111	NC	No carry	C = 0
0110	Z	Zero	Z = 1
1110	NZ	Not 0	Z = 0
1101	PL	Plus	S = 0
0101	MI	Minus	S = 1
0100	OV	Overflow	V = 1
1100	NOV	No overflow	V = 0
0110	EQ	Equal	Z = 1
1110	NE	Not equal	Z = 0
1001	GE	Greater than or equal	(S XOR V) = 0
0001	LT	Less than	(S XOR V) = 1
1010	GT	Greater Than	(Z OR (S XOR V))=0
0010	LE	Less than or equal	(Z OR (S XOR V))=1
1111	UGE	Unsigned greater than or equal	C = 0
0111	ULT	Unsigned less than	C = 1
1011	UGT	Unsigned greater than	(C=0 AND Z=0) = 1
0011	ULE	Unsigned less than or equal	(C OR Z) = 1

(6) Shock test





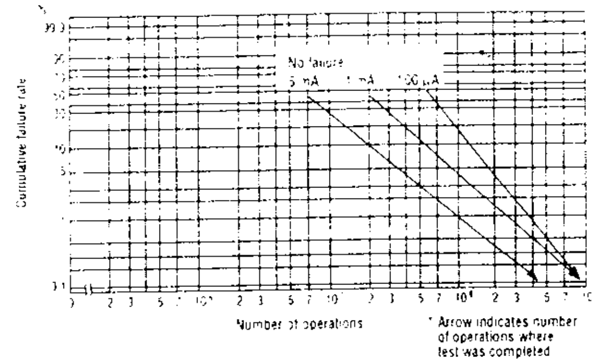
(7) Vibration test



LIFE EXPECTANCY DATA: ORD2212

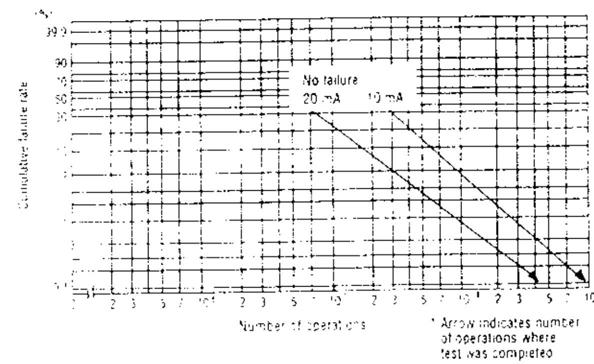
Load conditions

Voltage : 5 VDC
Current : 100 μA, 1 mA, 5 mA
Load : Resistive load



Load conditions

Voltage : 6 VDC
Current : 10 mA, 20 mA
Load : Resistive load



Load conditions

Voltage : 15 VDC
Current : 5 mA, 10 mA
Load : Resistive load