MICRO CONTROLLER BASED REMOTE CONTROL OF ELECTRICAL APPLIANCES THROUGH TELEPHONE LINES

PROJECT WORK

Submitted by

D. ANAND PRABAKAR
N. JAYANTHI
K. SENTHIL NAYAGI
R. SHALINI
D. THARANI

Under the Guidance of

Prof. K. RAM PRAKASH, M.E., M.I.S.T.E.,

1997-98

in partial fulfilment of the requirements
for the award of the Degree of
BACHELOR OF ENGINEERING IN
ELECTRONICS AND COMMUNICATION ENGINEERING
of the Bharathiar University, Coimbatore

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.

(Affiliated to Bharathiyar University)

CERTIFICATE

This is to certify that the project report entitled "MICRO CONTROLLER BASED REMOTE CONTROL OF ELECTRICAL APPLIANCES THROUGH TELEPHONE LINES" has been submitted by

____

in partial fulfilment for the award of Bachelor of Engineering in **ELECTRONICS AND COMMUNICATION ENGINEERING** *branch of the Bharathiyar University, Coimbatore - 641 006 during the academic year 1997-98*

Faculty Guide

Head of the Department

Certified that the candidate was examined by us in the project work viva-voce Examination held on ____________________ and the university register number was ____________________

Internal Examiner

External Examiner

107 Dr. NANJAPPA ROAD,
P.O. BOX 3990,
COIMBATORE - 641 018,
INDIA.

PHONE:
+91 - 422 - 380561
PHONE/FAX:
+91 - 422 - 213849

AL SYSTEMS

TO WHOM SO EVER IT MAY CONCERN

This is to certify that the following final year B.E. (Electronics & Communication Engineering) students of Kumaraguru College of Technology, Coimbatore have worked on their project entitled "MICRO CONTROLLER BASED REMOTE CONTROL OF ELECTRICAL APPLIANCES THROUGH TELEPHONE LINES" in our Organisation.

The students are: D. Anand Prabakar
N. Jayanthi
K. Senthil Nayagi
R. Shalini
D. Tharani

During this period their attendance and conduct were found to be good. We wish them the very best in their endeavor.

For AL SYSTEMS
J. Narayan Kumar

Partner

ACKNOWLEDGEMENT

Our project report would be duly complete only if we thank our beloved principal **Dr.S.SUBRAMANIAN, B.E, M.Sc(Engg)**, Ph.D for all his well wishes and inspiring thoughts.

Our Head of the department, **PROF. M. RAMASAMY B.E., M.E**, has always guided us with his ever valuable suggestions. Words fail us to thank him.

We would like to express our sincere thanks to our project guide **MR. K. RAMPRAKASH, B.E, M.E, MISTE**, who had lent us his fullest support during the progress of our project. He had always been ready to clear our skeptics. Without his inspiring guidance. we would not have completed the project.

We would also like to thank our class advisor **MR. N. SANJEEVI RAMANATHAN B.E, M.E, MISTE** and all the staff members of ECE department for their constant support in our endeavour.

Finally, our thanks to Mr. **J. NARAYANKUMAR B.E.** Managing Partner, AL System, for providing the necessary infrastructure and guidance for our project.

SYNOPSIS

Our project titled **"Micro controller based Remote control of electrical appliances through telephone lines"** aims at switching on or off the various electrical appliances through telephone lines.The system hardware and software are designed based on the telephone standards. The circuit which is designed here is capable of controlling upto four mains powered loads with the aid of commands (numbers) received via telephone. A Dual Tone Multi Frequency telephone set is used to send commands to the circuit and remotely control a wide range of mains appliances in and around the home (or) office with the help of system access code, channel information and system feedback.The hardware is centred around PIC Microcontroller and the software is written in the assembly language of the PIC microcontroller.

CONTENTS

INTRODUCTION

In this modern era of evergrowing electronics and telecommunication, telephone line communication is one of the most reliable forms of communication.Many a times,need may arise to switch on or off a light, a fan.or a coffee maker at home from office or any other place. The cost incurred is as per standard telephone rates. The project has been designed based on telephone standards.

1.2 BUILDING BLOCK OF A TELEPHONE

A Telephone consists of seven main components.

1. Receiver
2. Transmitter
3. Speech network
4. Hook switch
5. Ringer
6. Dialer
7. Bridge rectifier

The block diagram in Fig.1.1 illustrates the interconnection of the seven main components within a subscriber set.The transmitter and receiver are normally located in the handset section of a subscriber set. The transmitter converts user voice signals into electrical signals that are transmitted to the local switching center. The receiver converts electrical signals into sound. The signal at the receiver consists of the voice band signals from the switching centre and attenuated feedback from the transmitter. The feedback or “sidetone” function is performed by the speech network. The speech network also provides for separation of the transmit and receive signals at the subscriber set. Thus all signals between the switching centre and subscriber set may be carried over a single wire pair.

The hook switch may be in either of two positions, on hook or offhook. These conditions correspond to idle and busy circuits, respectively, with the offhook condition normally activated by lifting the handset. When the handset is lifted, a current sensing device at the switching centre defects the offhook state. The switching centre's logic circuitry will then turn off any ring signal and prepare to send and receive voice communication. If the subscriber is placing the call, the switching center will prepare to accept dial signals. The hook switch connects the telephone line to the ringer in an on hook position and to the speech network in an off hook position. In the off hook position, the subscriber set circuitry receives a DC bias from the power supply at the switching center. In the on hook position, a ring signal may be initiated by a caller. An electrical signal of about 75V and 20 30 Hz is typically generated at the switching center to activate the ringer at a subscriber set.

The two methods commonly used to transmit dialling information to the switching center are pulse generation and tone generation. Rotary-type dialers generate pulses on the line, and these pulses are sensed and counted by the switching centre Electronic pulse dialers simulate the mechanical action of a rotary dialer. Tonedialers generate tone combinations of various frequencies. When electronic dialers are used in a subscriber set, a bridge rectifier is used to prevent damage to the dialer due to line reversal. The bridge provides the dialer with the proper polarity of the DC line bias.

The simplest type of dialer in use today is the pulse dialer, which uses a series of pulses to transmit dial signals to the central office. The dial signals shown in Fig 1.2, has make, break and Inter digit intervals that constitute the digits. There is one another dialling method which is called Dual Tone Multi Frequency (DTMF) shown in Fig1.3. The DTMF address signaling is used by the telephone industry to signal over the voice transmission path of a telephone system. DTMF address signalling is used by the telephone industry to signal over the voice transmission path of a telephone system. DTMF signal has various advantages over pulse signalling, such as faster dialling speeds and the ability to signal over any voice grade transmission path.

1.3 PARAMETERS OF TELEPHONE

i. DC VOLTAGE

When the handset is on hook the voltage across the telephone line should be between 25 and 48 V.When the handset is off hook the voltage across the telephone line should be between 6 and 10 V.

ii. LINE CURRENT

When the telephone is off hook the current drawn by the telephone is about 50 - 60 m A.

iii. RINGER

The Ringer potential is about 75 V rms.

iv. DTMF DIALLING

This method of signalling uses 16 distinct voicebased frequencies each consisting of two sinusoidal signals, one from a "low group" and one form a "high group" of frequencies. The characters that represent these DTMF signals are shown in Table 1.1.

v. PULSE DIALLING

Pulse dialers must have the following characteristics

(a) The dial pulse signal should consist of sequence of momentary breaks in the telephone loop current corresponding to the numerical value of each digit, except digit '0'. which should be represented by 10 break intervals.

(b) For an automatic dialer, the make time should be between 32 and 35 ms, break time between 65 and 68 ms, which interdigit period (IDP) should be between 720 and 880 ms.

(vi) SPECIAL KEYS

(a) PAUSE:

This key is used to introduce additional delays between two digits. The delay should be within 2 to 2.4 sec.

(b) FLASH:

This key produce a loop break, the duration of which is between 280 and 320 ms.

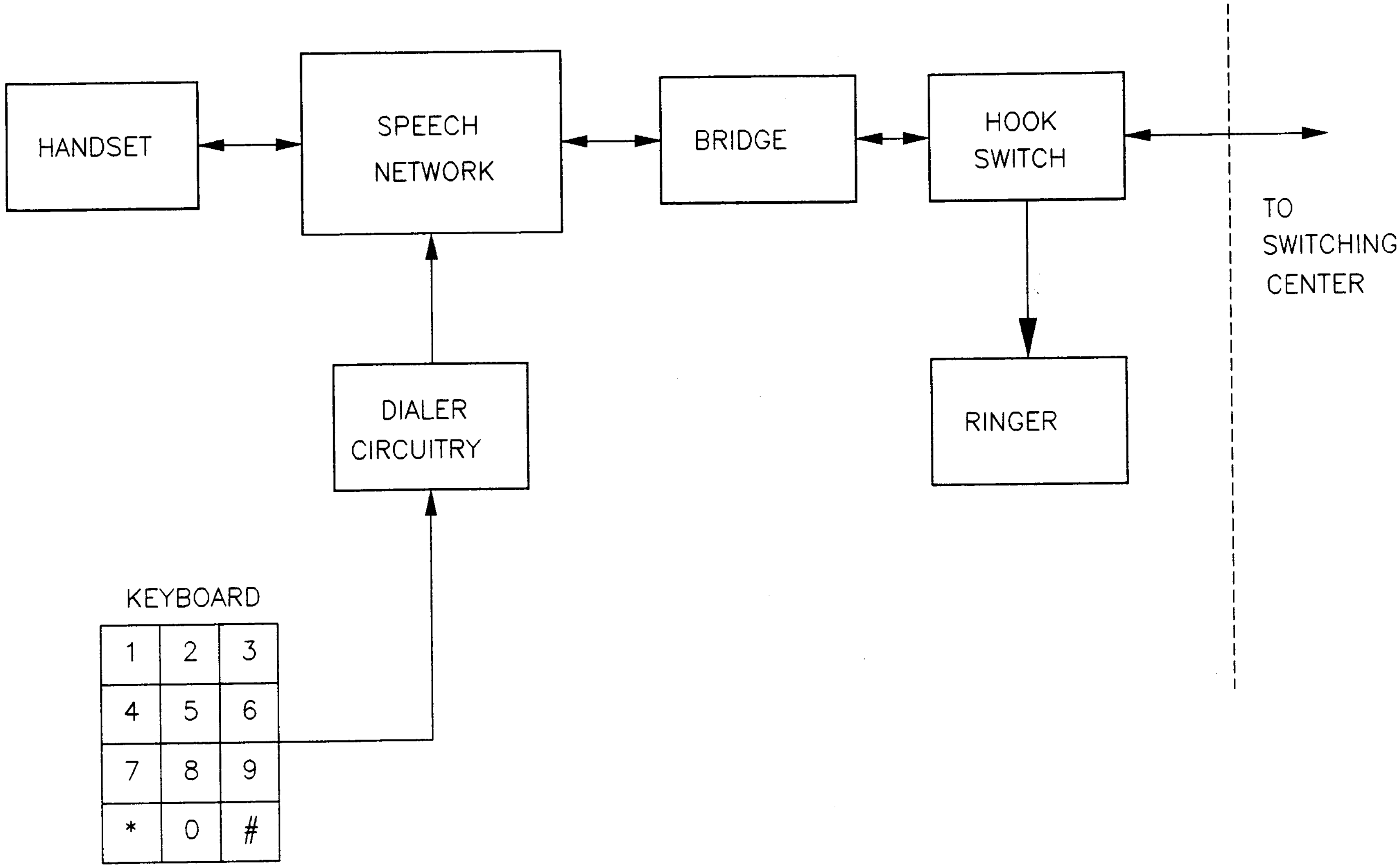


FIG 1.1 BLOCK DIAGRAM OF A TELEPHONE

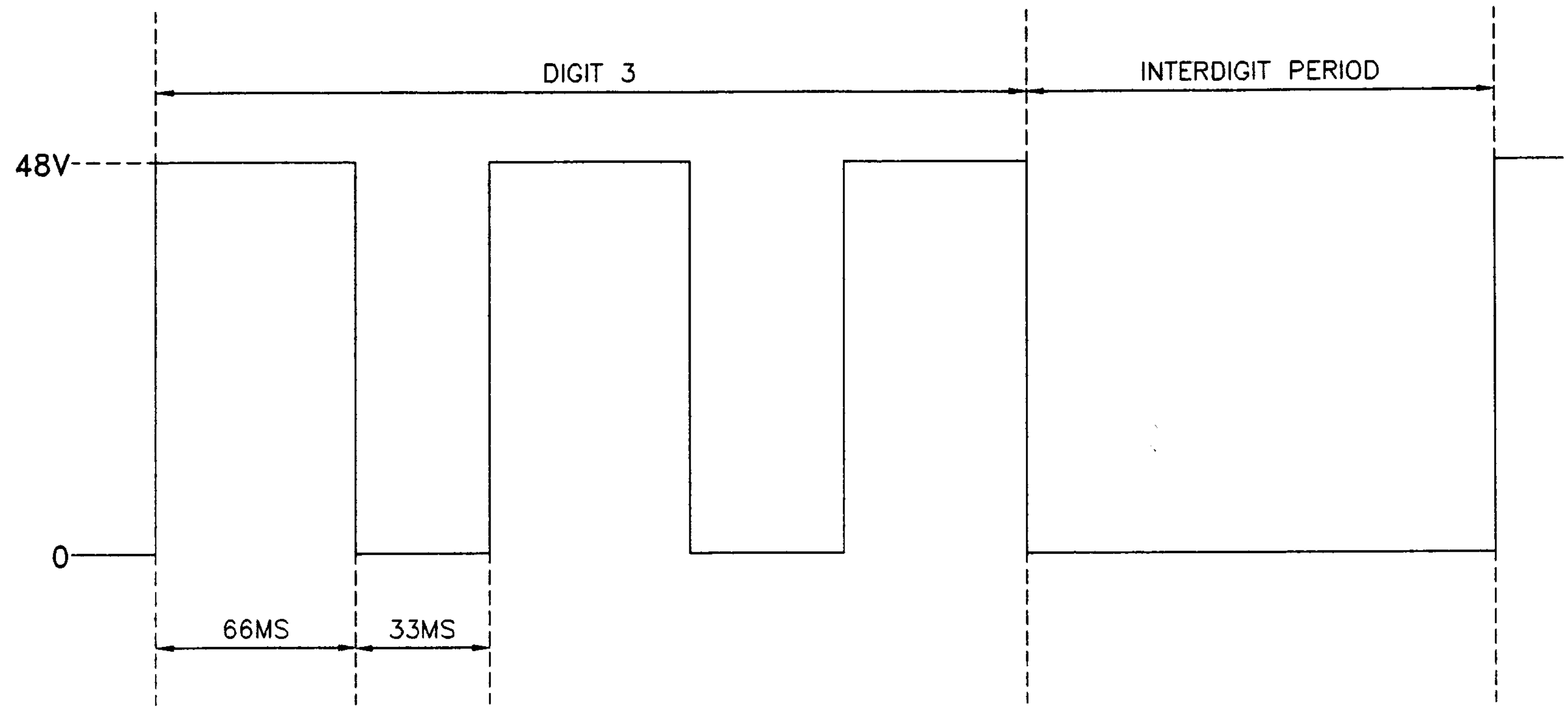


FIG 1.2 A DIAL PULSE TRAIN

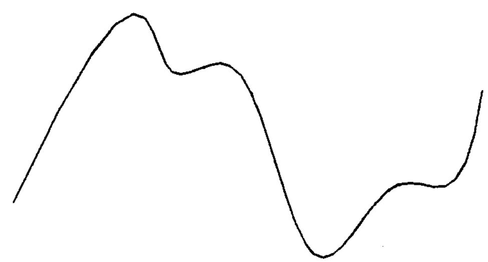


FIG 1.3 DTMF SIGNAL

NOMINAL LOW GRPUP FREQUENCIES (Hz)	NOMINAL HIGH GROUP FREQUENCIES (Hz)			
	1209	1336	1447	1663
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

TABLE 1.1 DTMF FREQUENCIES

CHAPTER 2

BASIC BLOCK DIAGRAM

OPERATION

1. The number of the telephone unit, to which the microcontroller is connected is dialled.

2. After one ring, the handset of the called terminal is electrically lifted.

3. The Microcontroller based switching unit waits for a preprogrammed system access code which the caller must transmit with the help of the DTMF keypad in his telephone set.

4. Reception of the correct code is acknowledged by a tone.

5. Preprogrammed channel information, to turn on or off a particular device is sent.

6. Depending on the codes sent, the particular electrical appliance is switched on or off.

DETAILED BLOCK DIAGRAM

This consists of 4 blocks

(i) Ring Detector

(ii) DTMF Receiver

(iii) PIC Microcontroller

(iv) Relay Control

OPERATION

When the ring arises, the 75V AC ring voltage appears at the input of the ring detector. The function of the ring detector is to convert bidirectional AC into undirectional DC. The 5V DC output of the ring detector is applied as input to the PIC Microcontroller which in turn energises the hookswitch relay.

The DTMF receiver gets connected to the telephone line through the hook switch relay. This connection is equivalent to electrically lifting the receiver of the telephone set. The relay remains energised for a certain period, within which the system access codes and the channel information have to be sent. The DTMF Receiver detects and decodes the digital frequency pairs corresponding to the pressed codes. The codes are then applied as input to the PIC Microcontroller. Corresponding to the codes, the PIC energies the output relay through which the particular appliance of interest is turned on or off.

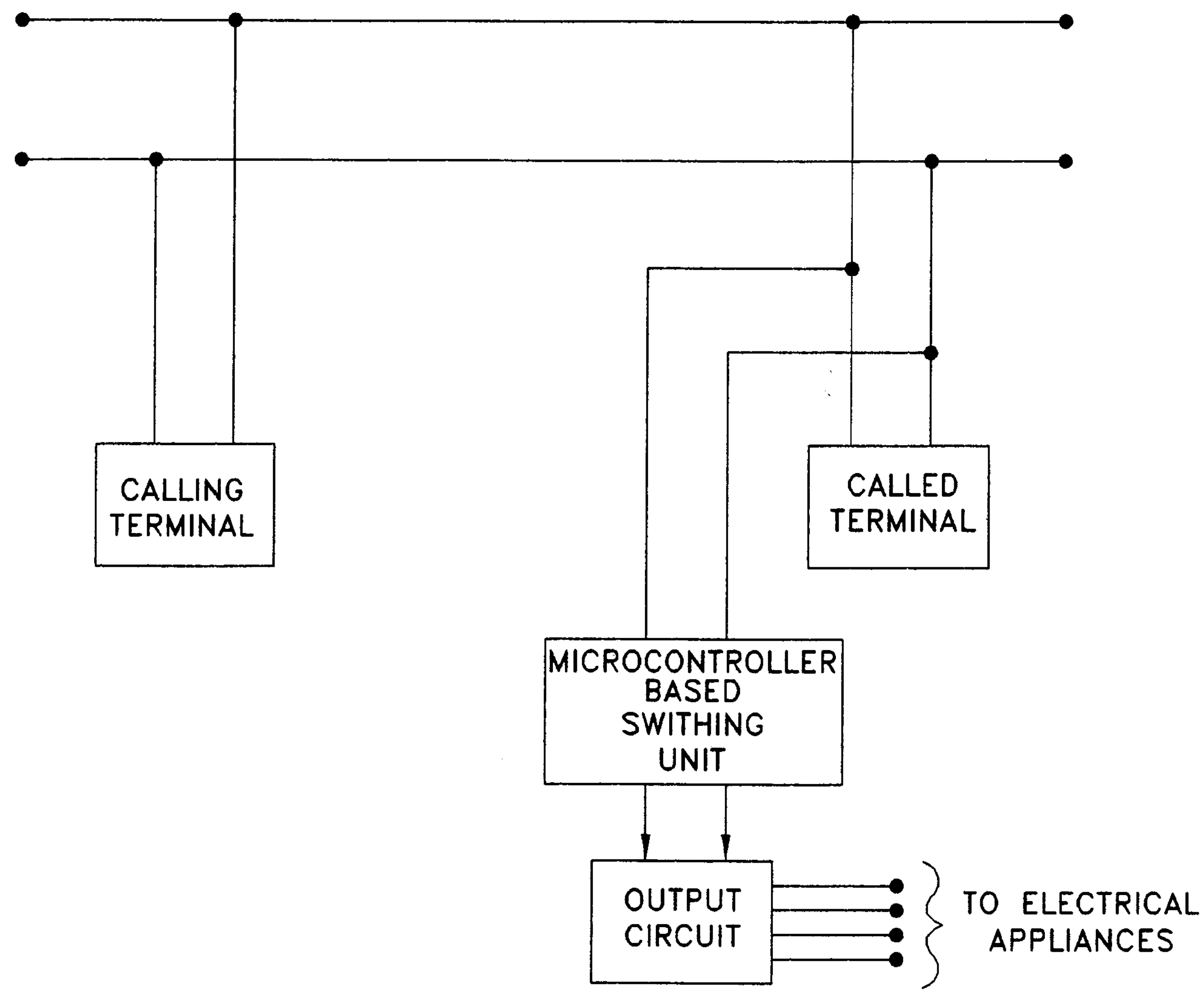


FIG 2.1 <u>BASIC BLOCK DIAGRAM</u>

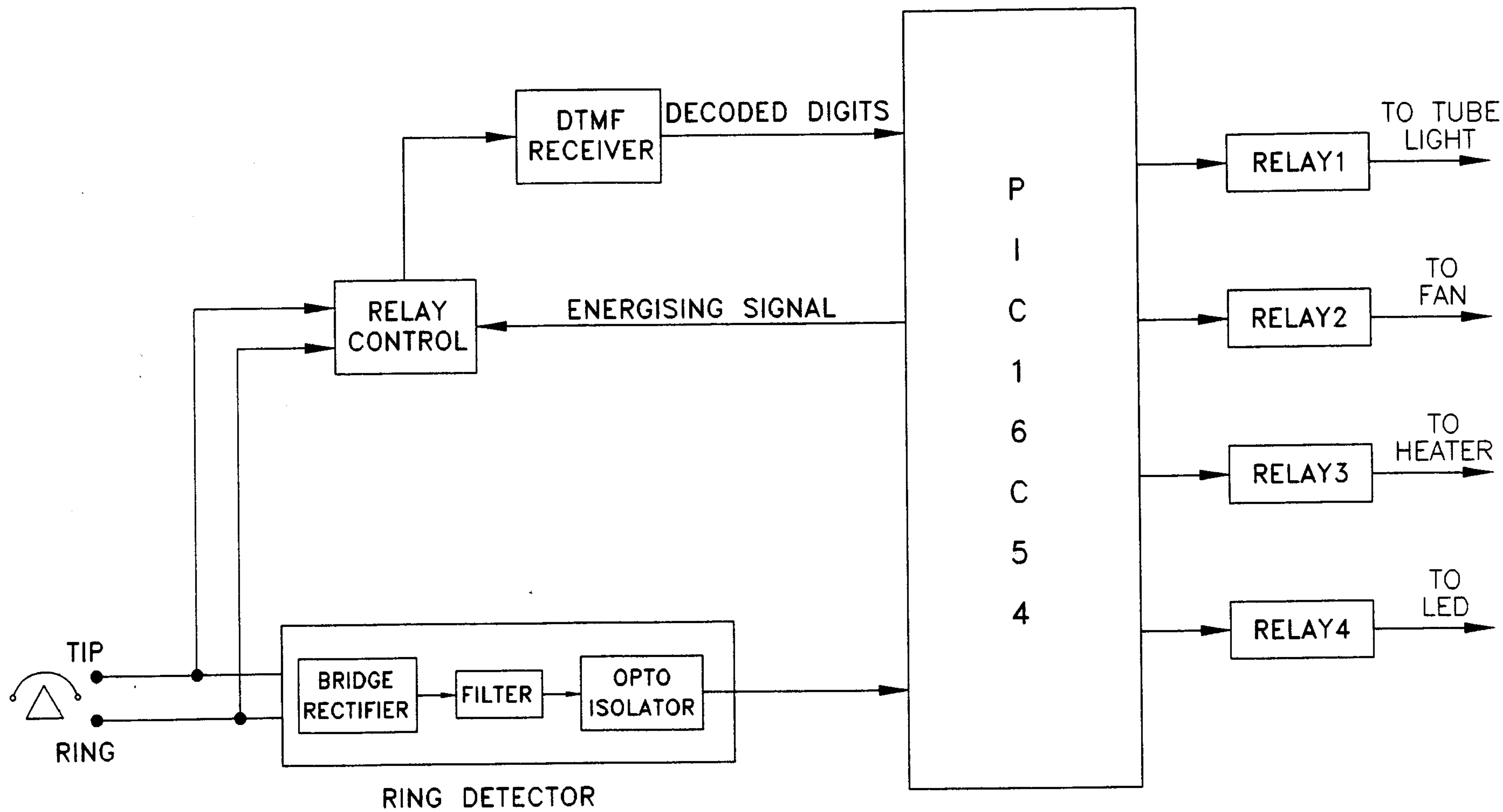


FIG 2.2 <u>DETAILED BLOCK DIAGRAM</u>

CHAPTER 3

PIC MICROCONTROLLER FEATURES

- High Performance RISC like CPU
- Only 33 single word instructions to learn All Single Cycle instructions (200 ns) except for program branches which are two cycle
- Operating speed DC-20 MHz clock input
- Dc_20Ns instruction cycle
- 12 Bit wide instructions
- 8 Bit wide data path
- 512-2K x 12 on-chip EPROM program memory
- 25_72 x 8 general purpose registers(SRAM)
- Seven special function hardware registers
- Two level deep hardware stack
- Direct, indirect and relative addressing modes for data and instructions

PERIPHERAL FEATURES

- 12-20 I/O pins with individual direction control
- 8_bit real time clock/counter (RTCC) with 8_bit programmable prescaler
- Power On reset
- Oscillator Start up Timer
- Watchdog Timer(WDT) with its own on_chip RC oscillator for reliable operation.

- Security EPROM fuse for code_protection
- Power saving SLEEP mode
- EPROM fuse selectable oscillator options:
  1. Low cost RC oscillator:XT
  2. Standard crystal/resonator:XT
  3. High speed crystal/reasonator:HS
  4. Power saving, low frequency crystal:LP

CMOS Technology :

- Low_power, high speed CMOS EPROM technology
- Fully static design
- Wide_operating voltage range :
  1. Commercial : 2.5V to 6.25V
  2. Industrial:2.5V to 6.25V
  3. Automotive 2.5V to 6.0V

Low Power consumption :

1. <2mA typical @ 5V, 4MHz
2. 5 A typical @ 3v,32 KHz
3. <3 typical standby current @ 3V. 0 c to 70 c

GENERAL DESCRIPTION

The PIC16C5X from Microchip Technology is a family low_cost. high_performance, 8 _bit, fully static, EPROM based COMOS microcontrollers. It employs a RISC_like architecture with only 33 single word/single cycle instructions to learn. All instructions are single cycle (200ns) except for program branches which take two cycles. The PIC 16C54 delivers performance an order of magnitude higher than its competitors in similar price category. The 12_bit wide instructions are highly symmetrical resulting in 2:1 code compression over other 8 bit microcontrollers in its class. The easy to use and easy to remember instruction set reduces development time significantly.

The PIC16C5X products are equipped with special microcontroller like features that reduce system cost and power requirements. The Power_on Reset and oscillator start_up timer eliminate the need for external reset circuitry. There are four oscillator configurations to choose from, including the power_saving LP (lw Power) oscillator and cost_saving RC oscillator. Power saving SLEEP mode. watchdog timer and code protection features improves system cost, power and reliability.The UV erasable cerdip_packaged versions are ideal for code development, while the cost_effective one Time Programmable(OTP) versions are suitable for production in any volume. The customer can take full advantage of Microchip's price leadership in OTP microcoontroller while benefiting from the OTP FLEXIBILITY.The PIC165X products are supported by an assembler, a software simulator. an in_circuit emulator and a production quality programmer. All the tools are supported by IBM PC and compatible machines.

APPLICATIONS

The PIC16C54 fits perfectly in applications ranging from high_speed automotive and appliance motor control to low_power remote transmitters/receivers, pointing devices and telecom processors. The EPROOM technology makes customization of application programs (transmitter codes, motor speeds, receiver frequencies. etc) extremely fast and convenient. The small footprint packages for through hole or surface mounting make this microcontroller perfect for all applications with space limitations. Low-cost, low-power, high performance, ease of use and I/O flexibility make the PIC16C54 very versatile even in areas where no microcontroller use has been considered before (e.g. timer functions, replacement of 'glue' logic in larger systems, co_processor applications).

ARCHITECTURAL DESCRIPTION :

HAVARD ARCHITECTURE :

The PIC16C54 single_chip microcomputer is low power. high speed, full static CMOS device containing EPROM. RAM. I/O and central processing unit on a single chip.The architecture is based on a register file concept with separate bus and memories for data and instructions(Harvard Architecture). The data bus and memory(RAM) are 8-bits wide, while the program bus and program memory(EPROM) have a width of 12-bits. This concept allows a simple yet powerful instruction set designed to emphasize bit, byte and register perations under high speed with overlapping instruction fetch and execution cycles. That means that, while one instruction is executed. the following instruction is already being read from the program memory. A block diagram of the PIC16C54 is given in Figure 3.1.

CLOCKING SCHEME/INSTRUCTION CYCLE

The clock input (from pin OSC1) is internally divided by four to generate four non overlapping quadrature clocks namely q1,q2,q3 and q4. Internally, PC is incremented every q1, instruction is fetched from program memory and latched into instruction register in q4. It is decoded and executed during the following q1 through q4.

DATA REGISTER FILE

The 8_bit data bus connects two basic functional elements together, the Register File composed of up to 80 addressable 8-bit registers including the I/O Ports, and AN 8-BIT wide Arithmetic Logic Unit. The 32 bytes of RAM are directly addressable while a "banking" scheme, with banks of 16 bytes each, is employed to address larger data memories.

Data can be addressed direct, or indirect using the file selectregister. Immediate data addressing is supported by special "literal" instructions which load data from program memory into the W register.The register file is divided into two functional groups:

Operational registers and general purpose registers. The operational registers include the Real Time Clock Counter (RTCC) register, the program Counter (PC), the status Register, the I/O registers(PORTs) and the File Select Register. The general purpose registers are used for data and control information under command of the instructions.In addition, special purpose registers are used to control the I/O port configuration and the prescaler options.

ARITHMETIC/LOGIC UNIT (ALU)

The 8_bit wide ALU contains one temporary working register (W Register). It performs arithmetic and Boolean functions between data held in the W Register and any file register. It also does single operand operations on either the W register or any file register.2.5

PROGRAM MEMORY

Up to 512 words of 12_bit wide on_chip program memory(Eprom) can be directly addressed. Larger program memories can be addressed by selecting one of up to four available pages with 512 words each. Sequencing of microinstructions is controlled via the Program Counter(PC) which automatically increments to execute in_line programs. Program control operation, supporting direct. Indirect, relative addressing modes, can be performed by Bit Test and Skip instructions. Call instructions, Jump instructions or by loading computed addresses into the PC. In addition, an on_chip two level stack is employed to provide easy to use subroutine nesting.

A wide variety of EPROM and RAM sizes, number of I/O pins, oscillator types, frequency ranges and packaging options are available. Depending on application and production requirements the proper device option can be selected using the information and tables in this section. When placing orders, please use the PIC16C5X product Identification System on the back page of this data sheet to specify the correct part number.

UV ERASABLE DEVICES

Four different device versions, are available to accommodate the different EPROM, RAM, and I/O configuration. These devices are optimal for prototype development and pilot series. The desired oscillator configuration is EPROM programmable as "RC","XT","HS" OR "LP". An erased device is configured as "RC", by default. Depending on the selected oscillator type and frequency, the operating supply voltage must be within the same range as a OTP/QTP part would be specified for.

ONE TIME PROGRAMMABLE (OTP) DEVICES

The availability of OTP device is especially useful for customers expecting frequent code changes and updates. OTP devices have the oscillator type pre_configured by the factor, and they are tested only for this special configuration (including voltage and frequency ranges, current consumption).The program EPROM is erased, allowing the use to write the application code into it. In addition, the watch dog timer can be disabled, and/or the code protection logic can be activated by programming special EPROM fuses. The 16 special EPROM bits for ID code storage are also user programmable.

QUICK TIMEAROUND PRODUCTION(QTP) DEVICES

Microchip offers a QTP Programming Service for factory production. orders. This service is made available for users who chose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical the OTP devices but with all EPROM locations and fuse options already programmed by the factory. Certain code and prototype verification procedures do apply before production shipments are available.

OPERATIONAL REGISTER FILES

INDIRECT DATA ADDRESSING(INDF)

This is not a physically implemented register. Addressing INDF calls for the contents of the File Select Register to be used to select a file register. INDF is useful as an indirect address pointer. For example, in the instruction ADDWF INDF, W will add the contents of the register pointed to by the FSR to the content of the W Register and place the result in W.If INDF itself is read through Indirect addressing (i.e. FSR = oh), then OOh is read. If INDF is written to via indirect addressing. the result will be a NOP.

REAL TIME CLOCK/COUNTER REGISTER (RTCC)

This register can be loaded and read by the program as any other register. In addition, its contents can be incremented by an external signal edge applied to the RTCC pin, or by the internal instruction cycle clock(CLKOUT=fosc/4). An 8-bit presclaer can be assigned to the

RTCC by writing the proper values to the PSA bit and the PS bits in the OPTION register. OPTION register is a special register (not mapped in data memory) addressable using the 'OPTION' instruction. If the prescaler is assigned to the RTCC, instructions writing to RTCC(e.g.CLRF RTCC, or BSF RTCC,5,...etc) clear the prescaler.The bit "RTS" (RTCC signal Source) in the OPTIOn register determines if RTCC is incremented internally or externally.

PROGRAM COUNTER

The program counter generates the addresses for upto 2048 x 112 on chip EPROM cells containing the program instruction words. Depending on the device type, the program counter and its associated two level hardware stack is 9 -11 bits wide.

PROGRAM COUNTER STACK WIDTH

The program counter is set to all "1" is upon a RESET condition. During program execution it is auto incremented with each instruction unless the result of that instruction changes the PC itself:

a) "GOTO" instructions allow the direct loading of the lower nine program counter bits(PC<8:0>).
   GOTO allows jump to any location on any page.

b) "CALL" instructions load the lower 8_bits of the PC directly, while the ninth bit is cleared to "0". The PC value, incremented by one, will be pushed into the stack.

c) "RETLW" instructions load the program counter with the top stack contents.

d) If PC is the destination in nay instruction (e.g.MOVWF PC, ADDWF PC, or BSF PC,5) then the computed 8_bit result will be loaded into the lower 8_bits of program counter. The ninth bit of PC will be cleared.

It should noted that because bit 8 (ninth bit) of PC is cleared in CALL instruction or any instruction which writes to the PC (e.g.MOVWF PC) all subroutine calls or computed jumps are limited to the first 256 locations of any program memory page.

I/O REGISTERS (PORTS)

The I/O registers can be written and read under program control like any other register of the register file. How ever. "read" instructions (e.g. MOVFPORTTB,W) always read the I/O pins, regardless if a pin is defined as "input" or "output". Upon a RESET condition. all I/O ports are defined as "input"(=high impedance mode) as the I/O control registers (TRISA,TRISB,TRISC) ARE ALL SET TO "ones".The execution of a "TRISF" instruction with corresponding "zeros" in the W_register is necessary to define any of the I/O pins as output.

PORT- A

4 - bit I/O register. Low order 4_bits only are used (RAO_RA3). Bits 4_7 are unimplemented and read as "zeros".

PORTB

8 - bit I/O register.

overwriting the previous content. As long as the pin stays in the input moode, no problem occurs. However, if bit 0 is switched into output mode later on, the content of the data latch may now be unknown.A pin actively outputting a "o" or "1" should not be driven from external devices at the same time in order to change the level on this pin("wired_or,"wired_and"). The resulting high output currents may damage the chip.

SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle. whereas for reading, the data must be valid at the beginning of the instruction cycle. Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before the next instruction which causes that file to be read into the CPU is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a NOP or an other instruction not accessing this I/O port.

GENERAL PURPOSE REGISTERS

PIC16C54 : f08h-f1Fh are general purpose register files.

SPECIAL PURPOSE REGISTERS

WORKING REGISTER

Holds second operand in two operand instructions and/or supports the internal data transfer.

TRISA I/O CONTROL REGISTER FOR PORT - A

Only bits 0-3 are available . The corresponding I/O port (f5) is only 4-bit wide.

TRISB I/O CONTROL REGISTER FOR PORT - B

TRISC I/O CONTROL REGISTER FOR PORT - C

The I/O control registers will be loaded with the content of the W register by executing of the TRIS f instruction.A "1" in the I/O control register puts the corresponding I/O pin into a high impedance mode. A "O" puts the contents of file register PORTA,PORTB or PORTC, respectively, out onthe selected I/O pins.These registers are "write_only" and are set to all "ones" upon a RESET condition.

OPTION PRESCALER/RTCC OPTION REGISTER

Defines prescaler assignment (RTCC or WDT), PRESCALER VALUE, signal source and signal edge of the RTCC. The OPTION register is "write-only" and is 6-bit wide. By executing the "option register. Upon a RESET condition, the option register is too all "ones"

RESET CONDITION

A reset condition can be caused by applying power to the chip (power_up). pulling the MCLR input "low", or by a watchdog Timer timeout. The device will stay in RESET as long as the oscillator startup timer (OST) is active or the MCLR input is "low".The oscillator startup timer is activated as soon as MCLR input is sensed to be high. This implies that in case of Power On Reset with MCLR TIED TO Vdd the

OST starts from power_up. In case of WDT time_out, it will start at the end of the time_out (since MCLR is high). In case of MCLR reset, the OST will start when MCLR goes high. The nominal OST time out period is 18ms. During a RESET condition the state of the PIC16C54 is defined as : The oscillator is running, or will be starte (power-up or wake up from SLEEP).All I/O port pins (RAO-RA3,RBO-RB7,RCO,RC7) are put into the high impedance state by setting the "TRIS" registers to all "ones"(=input mode). The option register is set to all "ones".The watchdog Timer and its prescaler are cleared.The upper-three bits(page select bits) in the STATUS Register are cleared to "zero"."RC" devices only: The "CLKOUT" signal on the OSC2 pin is held at a "low" level.

PRESCALER

An 8-bit counter is available as prescaler for the RTCC , or as a post_scaler for the Watchdog Timer, respectively. For simplicity, this counter is being referred to as "prescaler" throughout this data sheet. Note that there is only one prescaler available which is mutually exclusively shared between the RTCC and the Watchdog Timer. Thus, a prescaler assignment for the RTCC means that there is no prescaler for the Watchdog Timer, and vice versa.The PSA and PSO-PS2 bits in the OPTION REGISTER determine the prescaler assignment and pre scaler ratio. When assigned to the RTCC, all instructions writing to the RTCC (e.g.CLRF RTCC,MOVWF RTCC, BSF RTCC.X...etc) will clear the prescaler. When assigned to WDT , a CLRWDT instruction will clear the prescaler along with the watchdog Timer.

SWITCHING PRESCALER ASSIGNMENT

CHANGING PRESCALER FROM RTCC TO WDTThe prescaler assignment is fully under software control, i.e. it can be changed " on the fly" during program execution. To avoid an unintended device RESET, the following instruction sequence must be executed when changing the prescaler assignment from RTCC to WDT:

1. movlw b'xx0xx0xxx'	:	Select internal clock and select new
2. OPTION	:	Prescaler value. if new prescale value is ='000' OR '001', then select any other prescale value temporarily.
3. CLRF 1	:	Clear RTCC and prescaler
4. MOVLW B'xxxx1xxx'	:	Select WDT, Do not change prescale value
5.OPTION	:	
6. CLRWDT	:	clears wdt and prescaler
7. MOVLW'B'XXXX1XXXX	:	Select new prescale value
8. OPTION	:	

Steps 1 and 2 are only required if an external RTCC source is used. Steps 7 and 8 are necessary only if the desired prescale value is '000' or '001'.

CHANGING PRESCALER FROM WDT TO RTCC

To change prescaler from WDT to RTCC use the following sequence:

1. CLRWDT : Clear WDT and prescaler

2. MOVLWB'XXXX0XXXX' : Select RTCC, new prescale value and clock source.

3.2 INSTRUCTION SET OF PIC 16C54

ARITHMETIC INSTRUCTIONS

1. ADD fr, # literal
2. ADD B fr, bit
3. ADD fra, frb
4. ADD fr, w
5. ADD w, fr
6. SUB fv, # literal
7. SNC
8. SNZ
9. SUB fr, w
10. SUB B fr, bvt
11. SUB fr1, fr2
12. SNB bit
13. SB bit
14. STC
15. STZ
16. SC
17. SZ
18. INC fr

19. IJN2 fr. add r9

20. DEC fr

DATA TRANSFER INSTRUCTIONS

1. MOV B bit 1, / bit 2
2. MOV SZ w, ++ fr
3. MOV w. << fr
4. MOV w. / fr
5. MOV w. fr-w
6. MOV ! port - fr, fr
7. MOV option, # literal
8. MOV option, fr
9. MOV fr. # literal
10. MOV SZ w, - - fr
11. NEG fr
12. MOV w. <> fr
13. MOV B bit 1. bit 2
14. MOV w. ++ fr
15. MOV w. - - fr
16. MOV w. # literal
17. MOV w. fr
18. MOV ! port - fr, w

LOGICAL INSTRUCTIONS

1. AND fr. # literal
2. AND w. # literal
3. OR w. # bit
4. OR w. fr
5. OR fr1. fr2
6. OR fr. # literal
7. XOR w. fr
8. XOR fr. # literal
9. XOR fr1, fr2
10. XOR fr. w
11. XOR w, # literal
12. CJA fr1, fr2, addr 9
13. CJAE fr, # literal, addr 9
14. CJB fr1. fr2, addr 9
15. CJBE fr, # literal, addr 9
16. CJE fr1, fr2, addr 9
17. CJNE fr, # literal, addr 9
18. CLR fr.
19. CLR w
13\. CLC
14\. CSBE fr1, fr2
15\. CSNE fr1, fr2
16\. CSE fr. # literal

EXPLANATION OF INSTRUCTION

ADD fr. # literal (AD literal in to fr)

words : 2 cycles : 2 Affects : w. C, DC, Z

Operation : Literal added into fr via w c will be set if an overflow occurs : otherwisse. C will be cleared. DC will be set or cleared depending on whether or not an overflow occurs in the lower nibble. Z will be set if the result is zero: otherwise, z will be cleared. W is left holding the literal value.

coding : 1100 KKKK KKKK MOV w, # bit (MOV LW bit)

0001 111F FFFF (ADDWF fr,1)

eg. sample holds 90H

add sample, # 5

Sample now holds 95H (90+5). Both C & DC are cleared since no overflow occured in the byte on in lower nibble. Z is cleared since the result was not 0. W is left holding literal 5

XOR w, fr (XOR fr into w)

W : 1 C : 1 A:Z

Operation : Fr is XOR' d into w. Z will be set to 1 ie., result was 0. otherwise Z will be cleared to 0.

Coding : 0001 100f ffff XORwf fr,0

MOV w, fr (more fr into w)

W :1 C:1 A:Z : 0010 000f ffff MOVf fr, 0

Operation : Fr is moved into w. Z will be set to 1 is the value moved was 0, otherwise z will be cleared to 0.

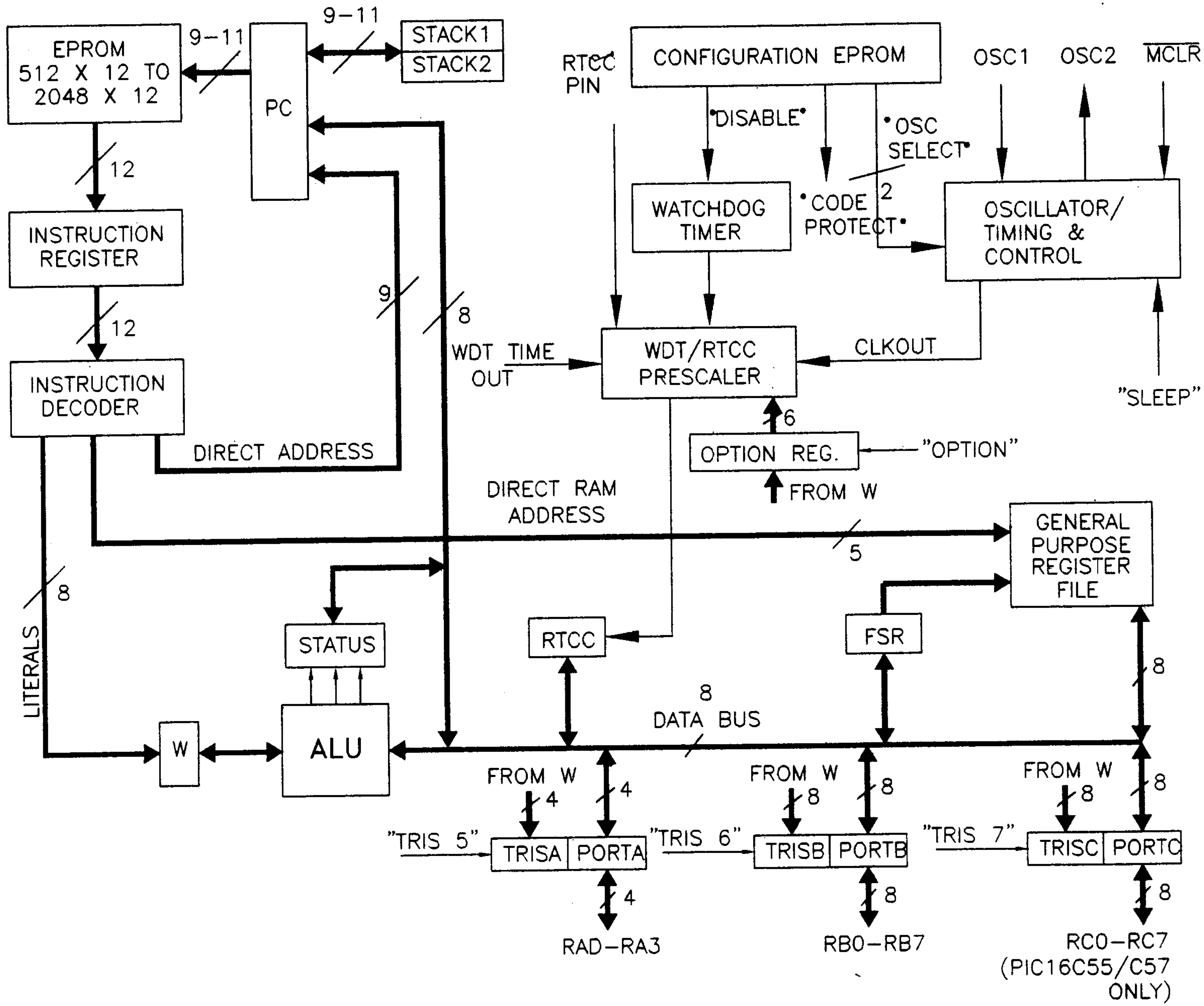


Fig : 3.1 PIC16C54 BLOCK DIAGRAM

PDIP, SOIC.CERDIP Window SSOP

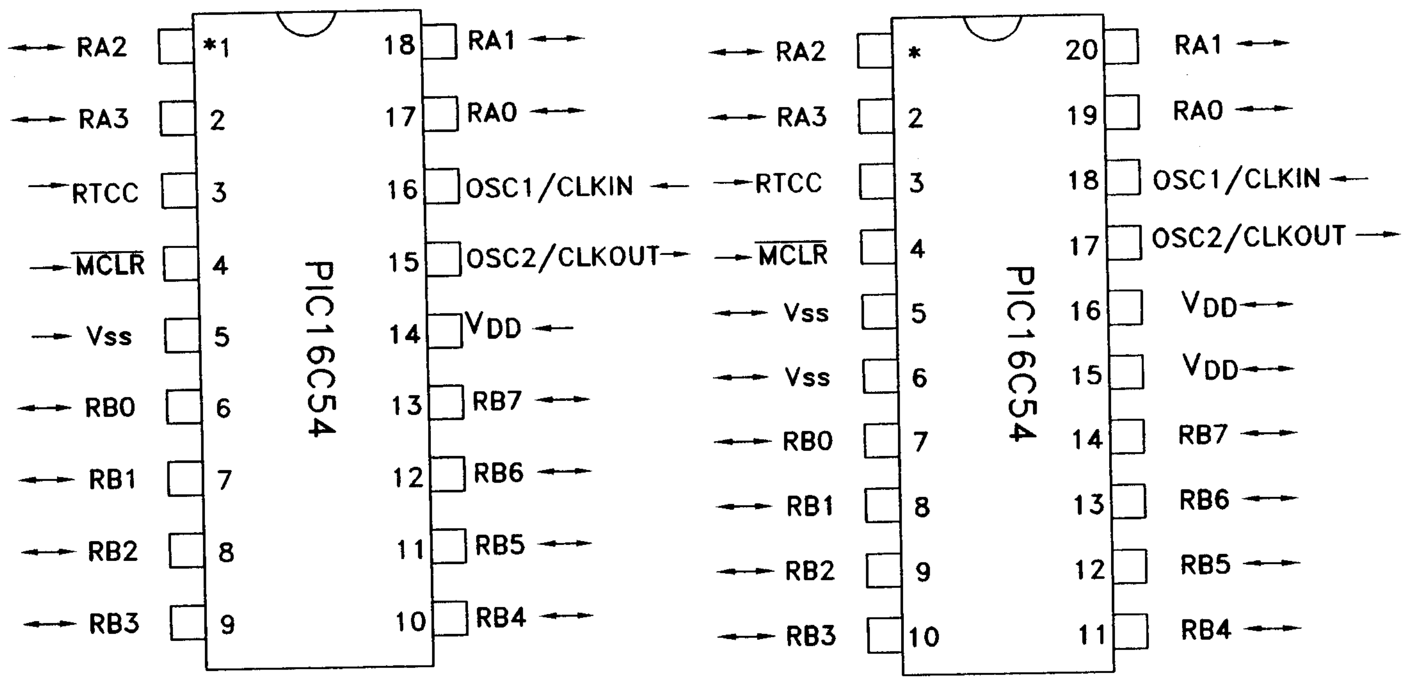
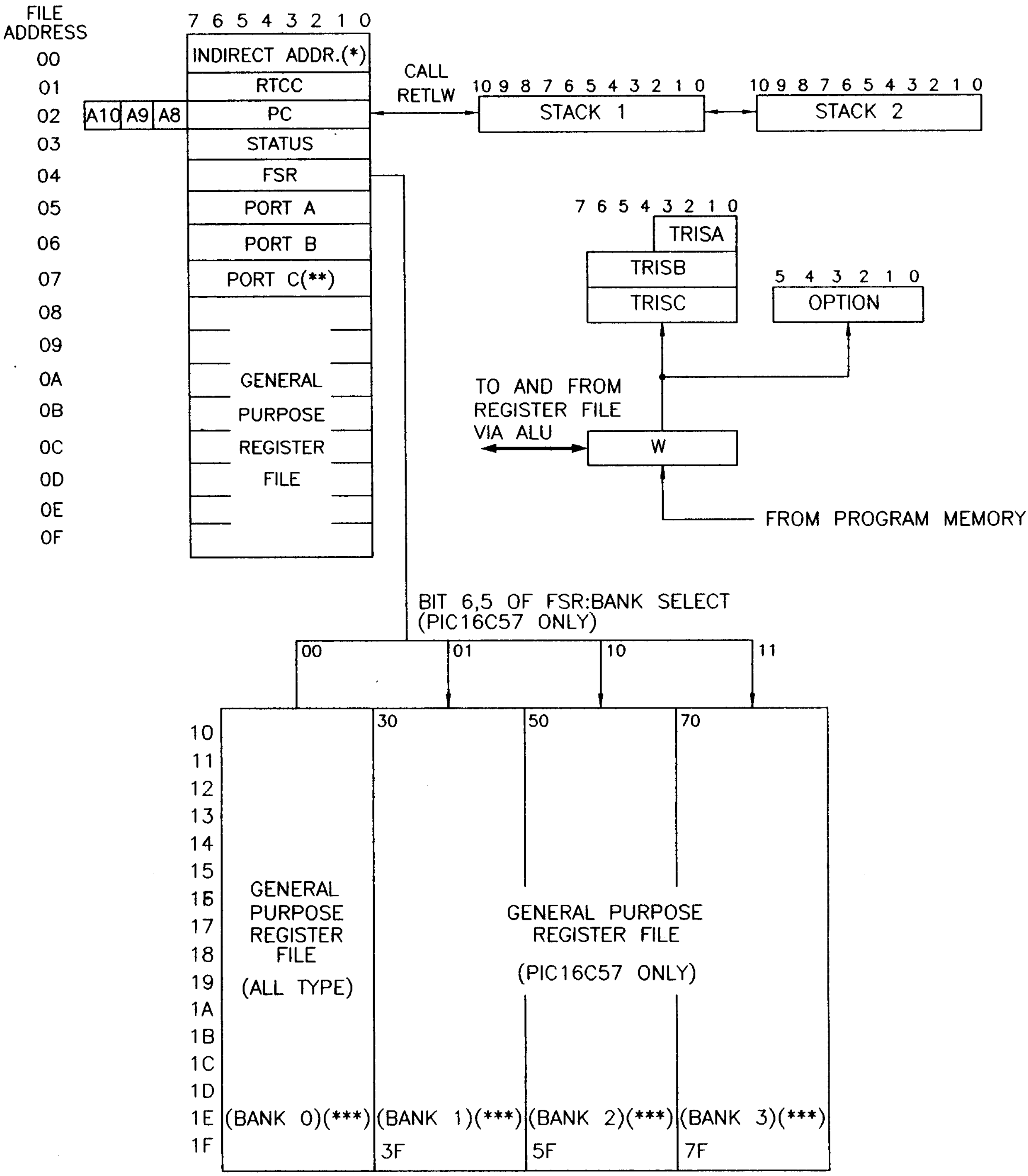


FIG 3.2 PIN CONFIGURATION

Name	Functions
RA0–RA3	I/O PORTA
RB0–RB7	I/O PORTB
RC0–RC7	I/O PORTC (C55/57 only)
RTCC	Real Time Clock/counter
$\overline{MCLR}$	Master Clear
OSC1/CLKIN	Oscillator (input)
OSC2/CLKOUT	Oscillator (output)
V_{DD}	Power supply
Vss	Ground
N/C	No (internal) Connection

Table : 3.1 PIN FUNCTIONS



(*) NOT A PHYSICALLY IMPLEMENTED REGISTER. SEE SECTION 4.0 FOR DETAILS.

(**) FILE ADDRESS 7h IS A GENERAL PURPOSE REGISTER ON THE PIC16C54/C56

(***) BANK 0 IS AVAILABLE ON ALL MICROCONTROLLERS WHILE BANK 1 TO 3 ARE ONLY AVAILABLE ON THE PIC16C57. (SEE SECTION 4.6 FOR DETAILS)

FIG 3.3 PIC16C5X DATA MEMORY MAP

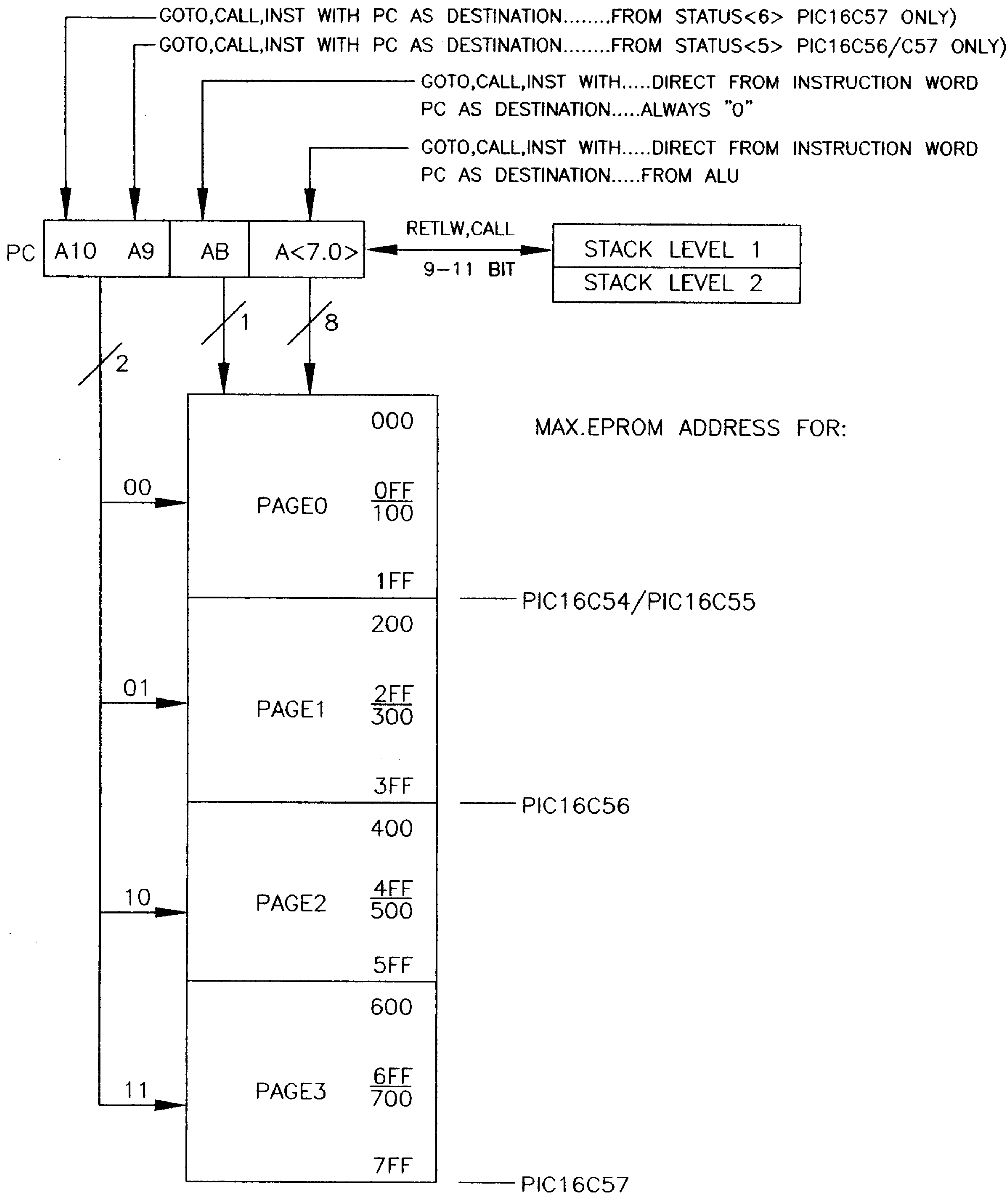
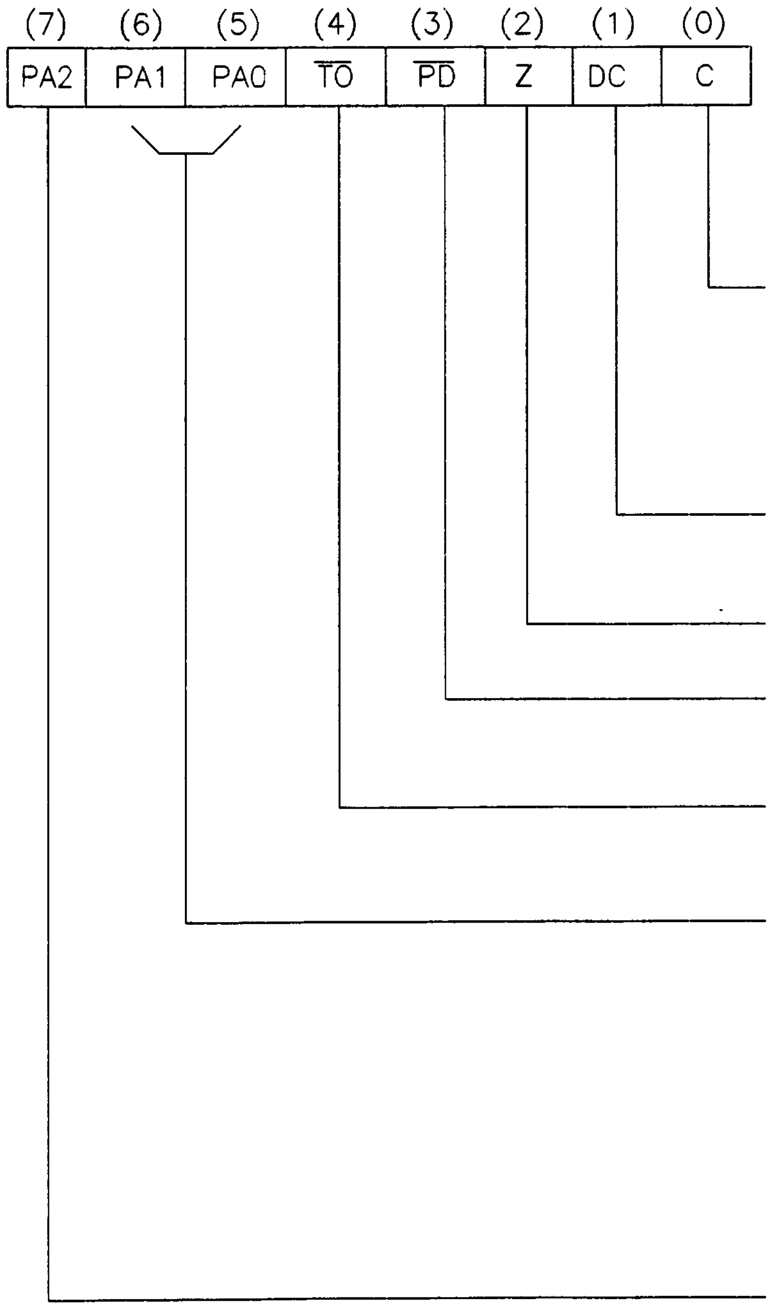


FIG : 3.4 PROGRAM MEMORY ORGANIZATION



RESET CONDITION:
PA2,PA1,PA0 cleared to '0'
TO,PD are set or reset as shown in Table 4.521
Z,DC,C are unknown on power on reset and unchanged in any other reset.

CARRY/$\overline{BORROW}$ BIT:
For ADDWF and SUBWF instructions , this bit is set if there is a carry out from the most significant bit of the resultant.
Note that a subtraction is executed by adding the two complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low bit of the source register.

DIGIT CARRY/$\overline{BORROW}$ BIT:
For ADDWF and SUBWF instructions, this bit is set if there is a carry out from the 4th low order bit of the resultant.

ZERO BIT:
Set if the result of an arithmetic or logic operation is zero.

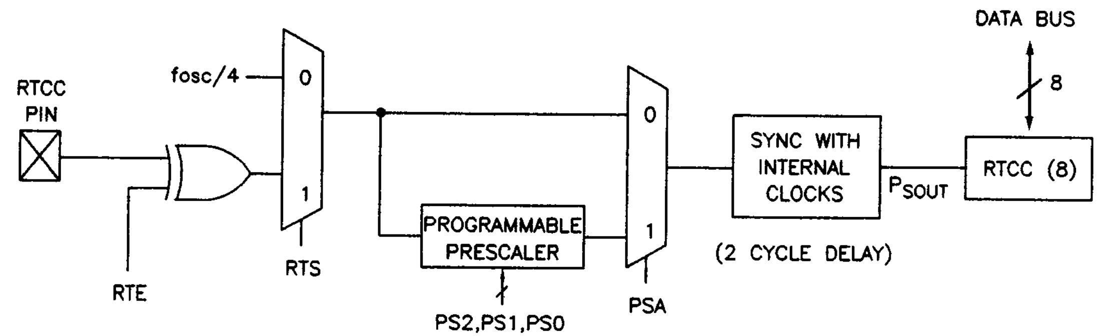
POWER DOWN BIT:
Set to "1" during power up or by a CLRWDT command. This bit is reset to "0" by a SLEEP instruction.

TIME-OUT BIT:
Set to "1" during power up and bt the CLRWDT and SLEEP command. This bit is reset to "0" by a watchdog timer time out.

PIC16C54/55 : Two geeral purpose read/write bits.
PIC16C56 : BIT 5....Page preselect bit
0=Page 0 (000-1FF)
1=Page 1 (200-3FF)
BIT 6....General purpose read/write bit
PIC16C57 : Two page preselect bits
00=Page 0(000-1FF)
01=Page 1(200-3FF)
10=Page 2(400-5FF)
11=Page 3(600-7FF)

BIT 7: General purpose read/write bit (reserved for future use)

FIG : 3.5 STATUS WORD REGISTER



NOTE:

1. Bits,RTE,RTS,PS2,PS1,PS0 are located in option register.
2. The prescaler is shared with Watchdog Timer.

FIG 3.16 RTCC BLOCK DIAGRAM(SIMPLIFIED)

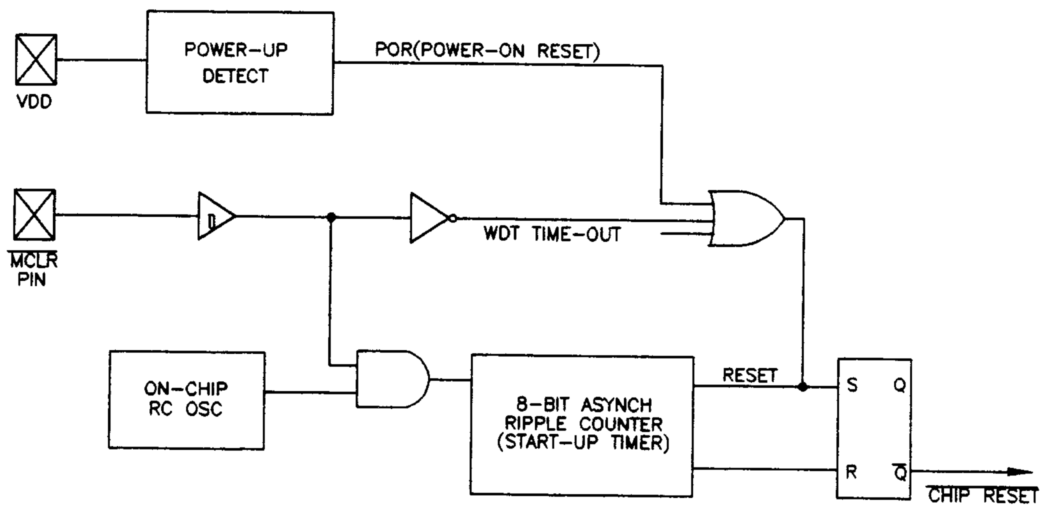


FIG 3.15 SIMPLIFIED POWER ON RESET BLOCK DIAGRAM

CHAPTER 4

M-8870 DTMF RECEIVER

The teletone M-8870 is a full DTMF receiver that integrates both bandsplit filter and decoder functions into a single 18_pin DIP or DIP or SOIC package. Manufactured using CMOS process technology, the M_8870 offers low power consumption (35mW max) and precise data handling. Its filter section uses switched capacitor technology for both the high and low group filters and for dial tone rejection. its decoder uses digital counting techniques to detect and decode all 16 DTMF tone pairs into a 4 bit code. External component count is minimized by provision of an on chip diffenential input amplifier, clock generator, and latched tristate interface bus. Minimal external components required include a low_cost 3.579545 Mhz color burst crystal, a timing resistor, and a timing capacitor.

The M-8870-02 and M-8870-03 provide a "power_down" option which, when enabled, drops consumption to less than 0.5 mW. The -02 and -03 versions can also inhibit the decoding of fourth column digits. The -03 version features increased input sensitivity.

FEATURES

- Low power consumption
- Adjustible acquisition and release times
- Central office quality and performance
- Power down and inhibit modes (-02 and -03 ersions)

- Inexpensive 3.58 Mhz time basesingle 5 volt power supply
- Dial tone suppression

APPLICATIONS

- Telephone switch equipment
- Mobile radio
- Remote control
- Remote data entry
- Paging System
- Personal ComputersT
- Telephone Answering Machines
- Credit Card Systems

FUNCTIONAL DESCRIPTION

M-8870 operating functions include a bandsplit filter that separates the high and low tones of the received pair. and a difital decoder that verifies both the frequency and duration of the received tones before passing the resulting 4 bit code to the output bus.

FILTER

The low and high group tones are separated by applying the dual tone signal to the inputs of two 6th order switched capacitor bandpass filters with bandwidths that correspond to the bands enclosing the low and high group tones. the filter also incorporates notches at 350 and 440 Hz, providing excellent dial tone rejection. Each filter output is followed by a single-order switched capacitor section that smoothes the signals prior to liiting. Signal limiting is performed by high-gain comparators provided with hysteresis to prevent detectin of unwanted low¬level signals and noise. The comparator outputs provide full rail logic swings at the frequences of the incoming tones.

DECODER

The M_8870 decoder uses a digital counting technique to deterine the frequencencies of the limited tones and to verify that theycorrespond to standard DTMF freqencies. A complex a averaging algorithm is used to protect against tone simulation by extraneous signals (such as voice) while tloerating small frequency variations. The algorithm ensures an optimum combination of immunity to talkoff and tolerance to interfering signals (third tones) and noise. When the detector recognizes the simultaneous presence of two valid tones (known as "signals condition"), it raises the Early Steering flag (ESt). Any subsequent loss of signal condition will cause ESt to fall.

STEERING CIRCUIT

Before a decoded tone pair is registered. the reciver checks for a valid signal duration (referred to as "character recognition¬condition"). This check is performed by an external RC time constant driven by ESt. A logic high on EST causes VC to rise as the capacitor discharges. Provided that signal condition is maintained (ESt remains high) for the validation period(tGTF), Vc reaches the threshold (VTST)) OF THE STEERING LOGIC TO REGISTER THE TONE PAIR. THUS LATCHING IS corresponding 4-bit code into the output latch. At this point, the GT output is activated and drives Vc to Vdd. GT continues to drive high as long as ESt remains high. Finally, after a short delay to allow the outpt latch to settle , the "delayed steering" output flag (StD) goes high, signaling that a received tone pair has been registered. The contents of the output latch are made available on the 4-bit output bus by raising the tree state control input(OE) to a logic high. The steering circuit works in reverse to validate the interdigit pause between signals. Thus, as well as rejecting signals too short to be considered valid, the receiver will tolerate signal interruptions (drop outs) too short to be considered a valid pause. This capability, together with the ability to select the steering time constants externally, allows the designer to tailor performance to meet a wide variety of system requirements.

GUARD TIME ADJUSTMENT

Where independent selection of signal duration and inter digit puase are not required , the simple steering circuit of is applicable. Componenent values are chosen according to the formula :tREC = tDP + tGTPtgtp = 0.67 RCThe value of tDP is a parameter of the device and tREC is the minimum signal duration to be recognised by the receiver. A value for C of 0.1 uF is recommended for most applications, leaving R to be selected by the designer. For example, a suitable value of R for a tREC of 40 ms would be 300 K ohm. The timing for most telecommunication applications are satisfied with this circuit. Different steering arrangements may be used to select independently the guard times for tone-present (tGTP) and tone-absent(tGTA). This may be necessary to meet system specifications that place both accept and reject limits on both tone duration and interdigit pause.Guard time adjustment also allows the designer to tailor system parameters such as talkoff and noise immunity. increasing tREC improves talkf off performance. Since it reduces the probability that tones simulated by speech will maintain signal condition long enough to be registered. On the other hand, a relatively short tREC with a long tDO would be appropriate for extremely noisy environments where fast acquisition time and immunity to dropouts would be required. A logic high applied to pin 6 (PD) will place the deviece into standby mode to minimize power consumption. It stops the oscillator and the functioning of the filetes. On _01 models, this pin is tied to ground (logic low).

TONE DECODING

Inhibit mode is enabled by a logic high input to pin 5 (INH). It inhibits the detection of 1633 Hz. The output code will remain the same as the previous detected code. On-01 models, this pin is tied to ground (logic low).Input ConfigurationThe input arrangement of the M_8870 provides a differential input operational amplifier as well as a bias source (VREF) to bias the inputs at mid-rail. Provisions is made for connection of feedback resistor to the op-amp output (GS) for gain adjustment.

DTMF CLOCK CIRCUIT.

The internal clock circuit is completed with the addition of a standard 3.579545 MHz television color burst crystal. The crystal can be connected to a single M-8870s or to a series of M-8870s. A single crystal can be used to connect a series of of M-8870s by coupling the oscillator output of each M-8870 through a 30 pF capacitor to the oscillator input of the next M-8870.

OPTOISOLATOR

The optoisolator used here is MCT2E. It is 6 pin DIP. It consists of a gallium arsenide infrared emitting diode optically coupled to a monolithic silicon phototransistor detector.

APPLICATIONS

- General purpose switching circuits
- Interfacing and coupling systems of different potentials and impedances
- I/O Interfacing
- Solid state relays
- Monitor and detection circuits

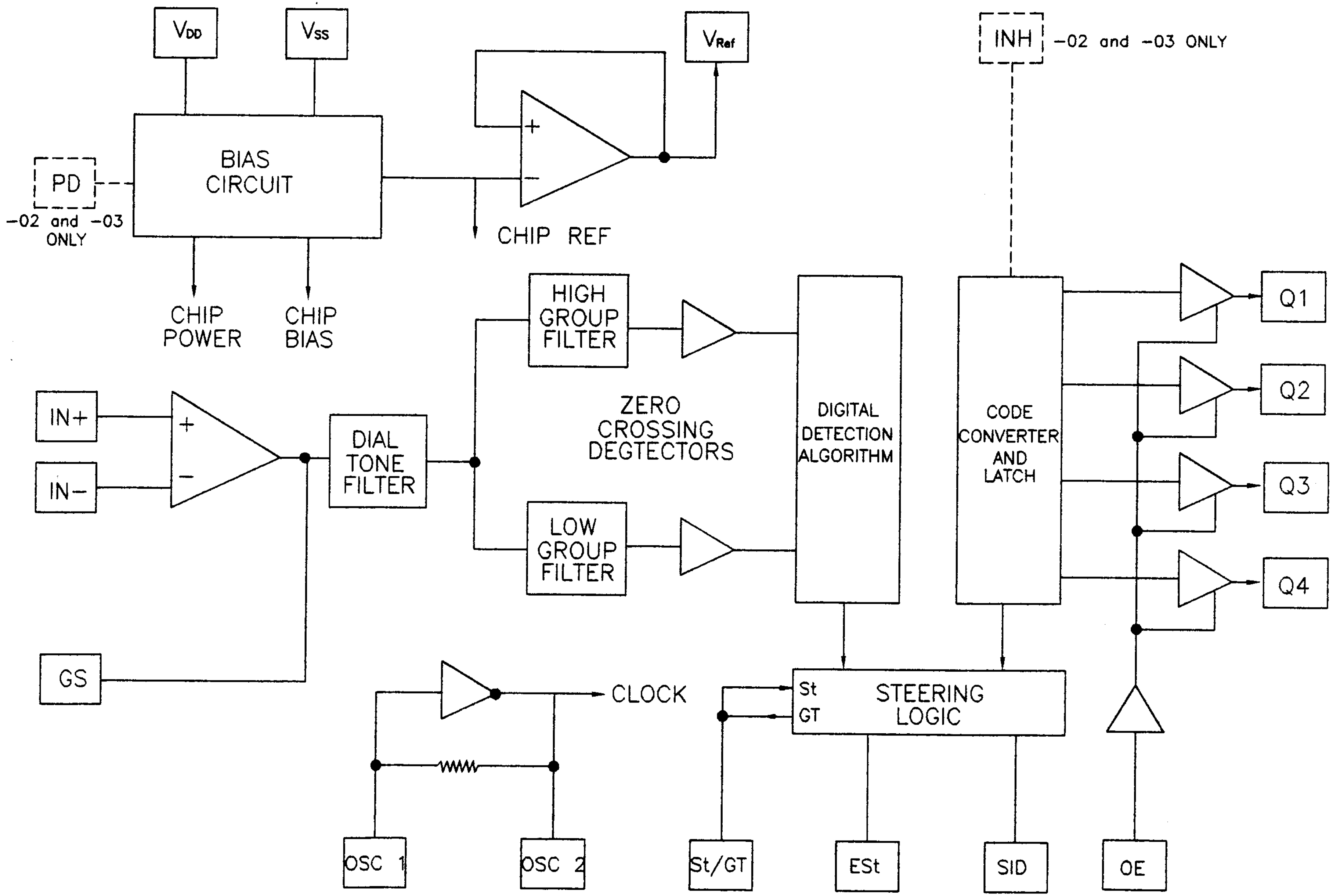


FIG 4.3 BLOCK DIAGRAM

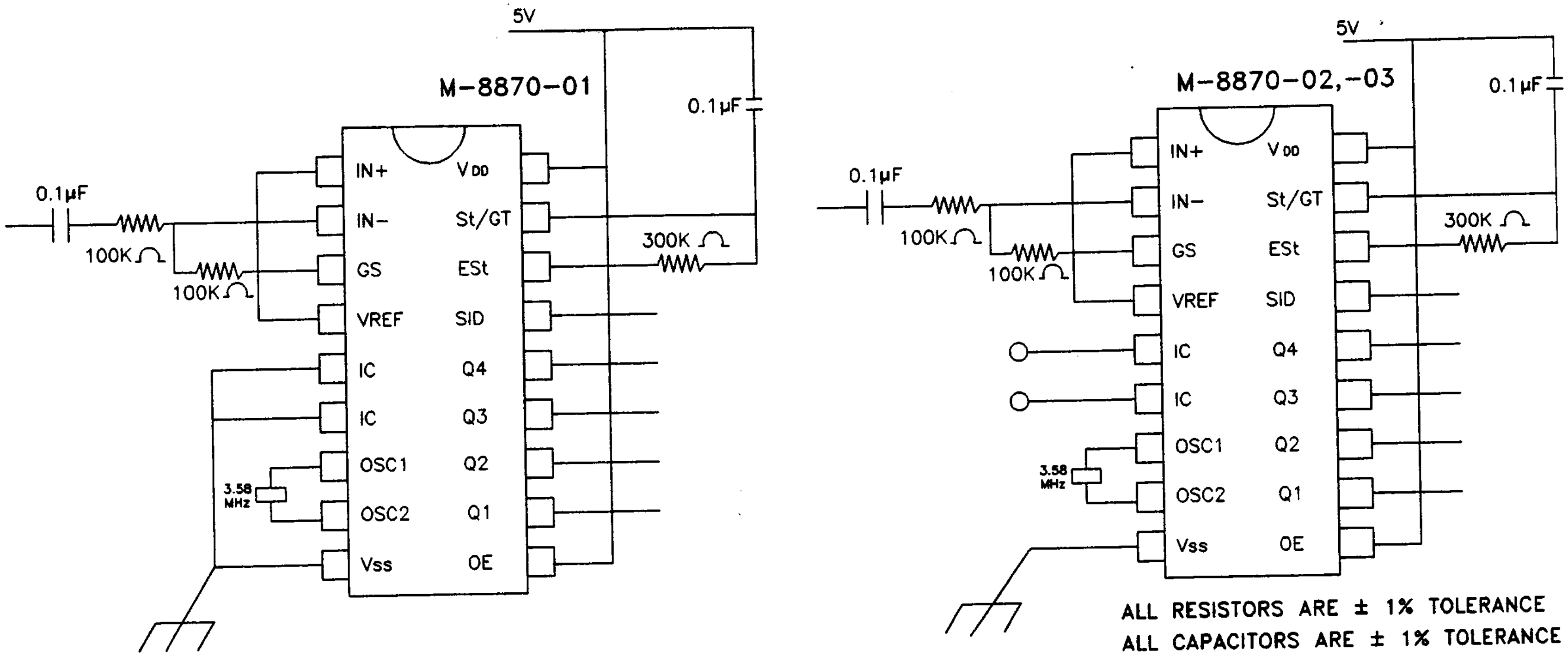
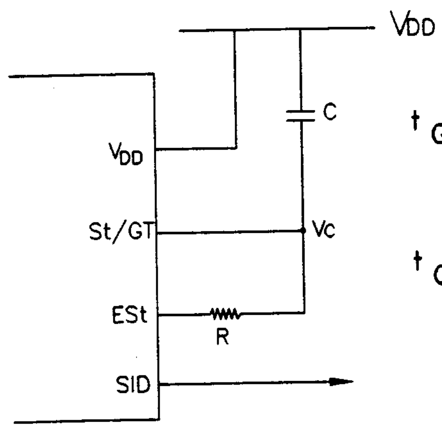


FIG 4.3 SINGLE-ENDED INPUT CONFIGURATION



$$t_{GTA} = (RC)\mathrm{Ln}\left(\frac{V_{DD}}{V_{TSt}}\right)$$

$$t_{GTP} = (RC)\mathrm{Ln}\left(\frac{V_{DD}}{V_{DD} - V_{TSt}}\right)$$

Fig : 4.2 BASIC STEERING CIRCUIT

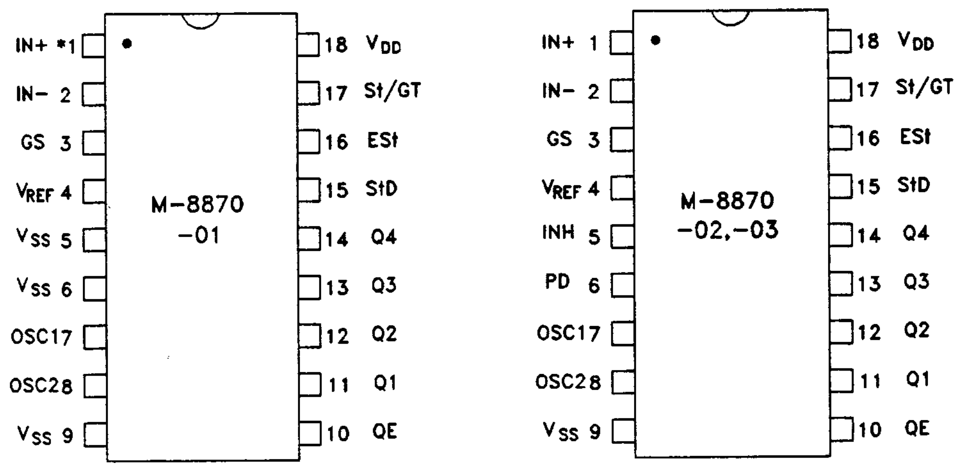
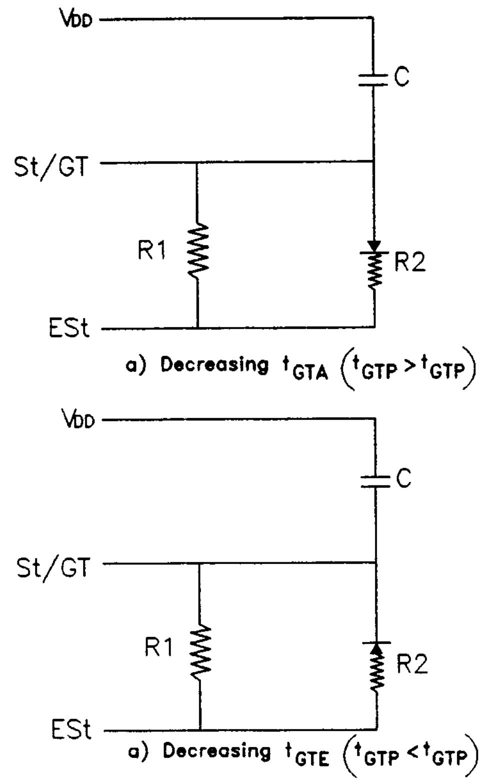


FIG 4.2 PIN CONNECTIONS



$$t_{GTP} = (R_1 C) \ln\left(\frac{V_{DD}}{V_{DD} - V_{TSt}}\right)$$

$$t_{GTA} = (R_P C) \ln\left(\frac{V_{DD}}{V_{TSt}}\right)$$

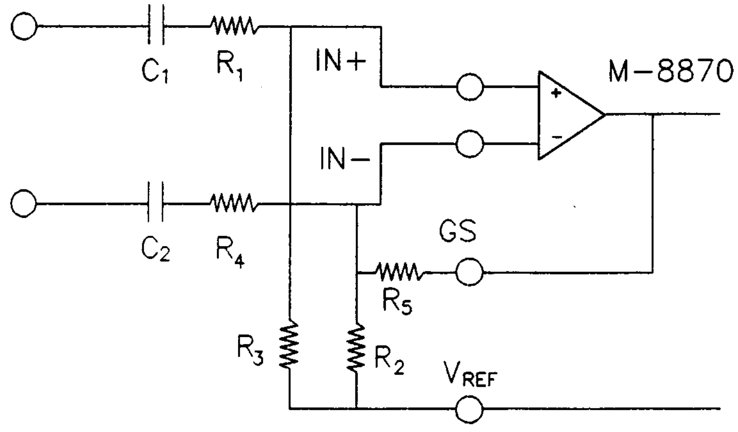
$$R_p = \frac{R_1 R_2}{R_1 + R_2}$$

$$t_{GTP} = (R_p C) \ln\left(\frac{V_{DD}}{V_{DD} - V_{TSt}}\right)$$

$$t_{GTA} = (R_1 C) \ln\left(\frac{V_{DD}}{V_{TSt}}\right)$$

$$R_p = \frac{R_1 R_2}{R_1 + R_2}$$

FIG : 4.4 GUARD TIME ADJUSTMENT



DIFFERENTIAL INPUT AMPLIFIER

$C_1 = C_2 = 10nF$

$R_1 = R_4 = R_5 = 100K\Omega$

$R_2 = 60K\Omega \quad R_3 = 37.5K\Omega$

$$R_3 = \frac{R_2 R_5}{R_2 + R_5}$$

$$\text{VOLTAGE GAIN (Av diff)} = \frac{R_5}{R_1}$$

ALL RESISTORS ARE ± 1% TOLERANCE
ALL CAPACITORS ARE ± 1% TOLERANCE

INPUT IMPEDANCE

$$\left(Z_{INDIFF}\right) = 2\sqrt{R_1^{\,2} + \left(\frac{1}{\omega C}\right)^2}$$

FIG : 4.5 DIFFERENTIAL INPUT CONFIGURATION

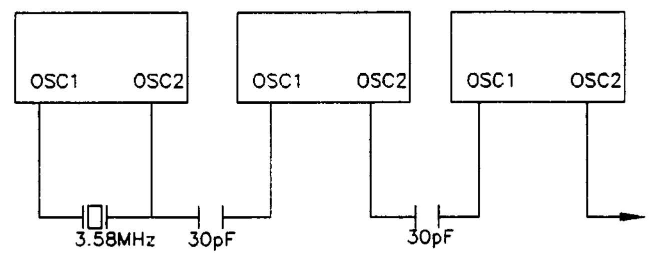


FIG : 4.6 COMMON CRYSTAL CONNECTION

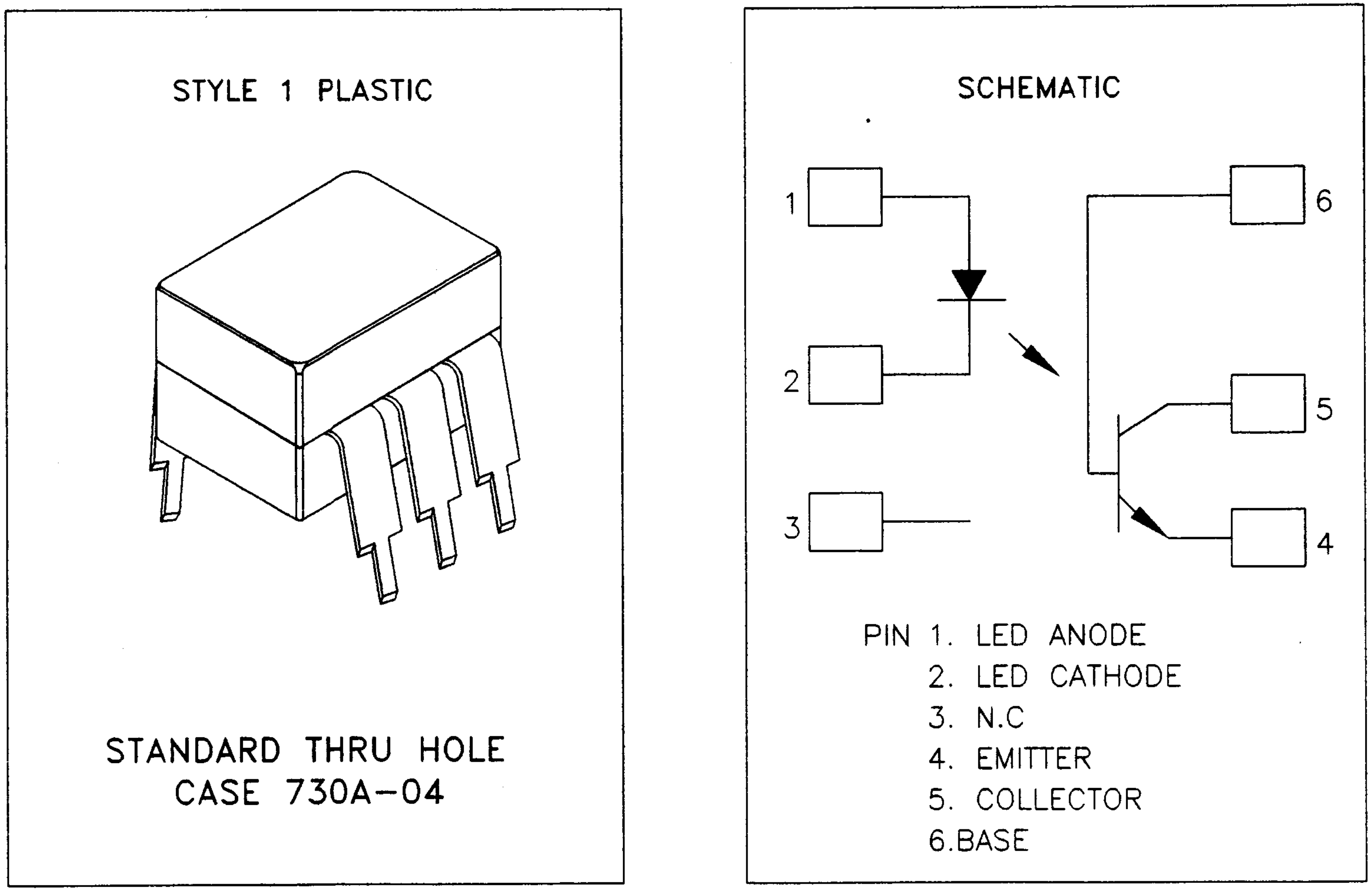


FIG 4.8 <u>OPTO ISOLATOR (MCT2E)</u>

CHAPTER - 5

CIRCUIT DIAGRAM EXPLANATION:

The heart of the circuit diagram is formed by the PIC microcontroller (16C54). All integrated DTMF decoder type M8870 decodes the dialling codes received via the telephone line.The telephone line interface consists of two parts. One to detect the ringing signals that enables the unit to answer the call at the right moment and another to receive and transmit tones via the telephone line.

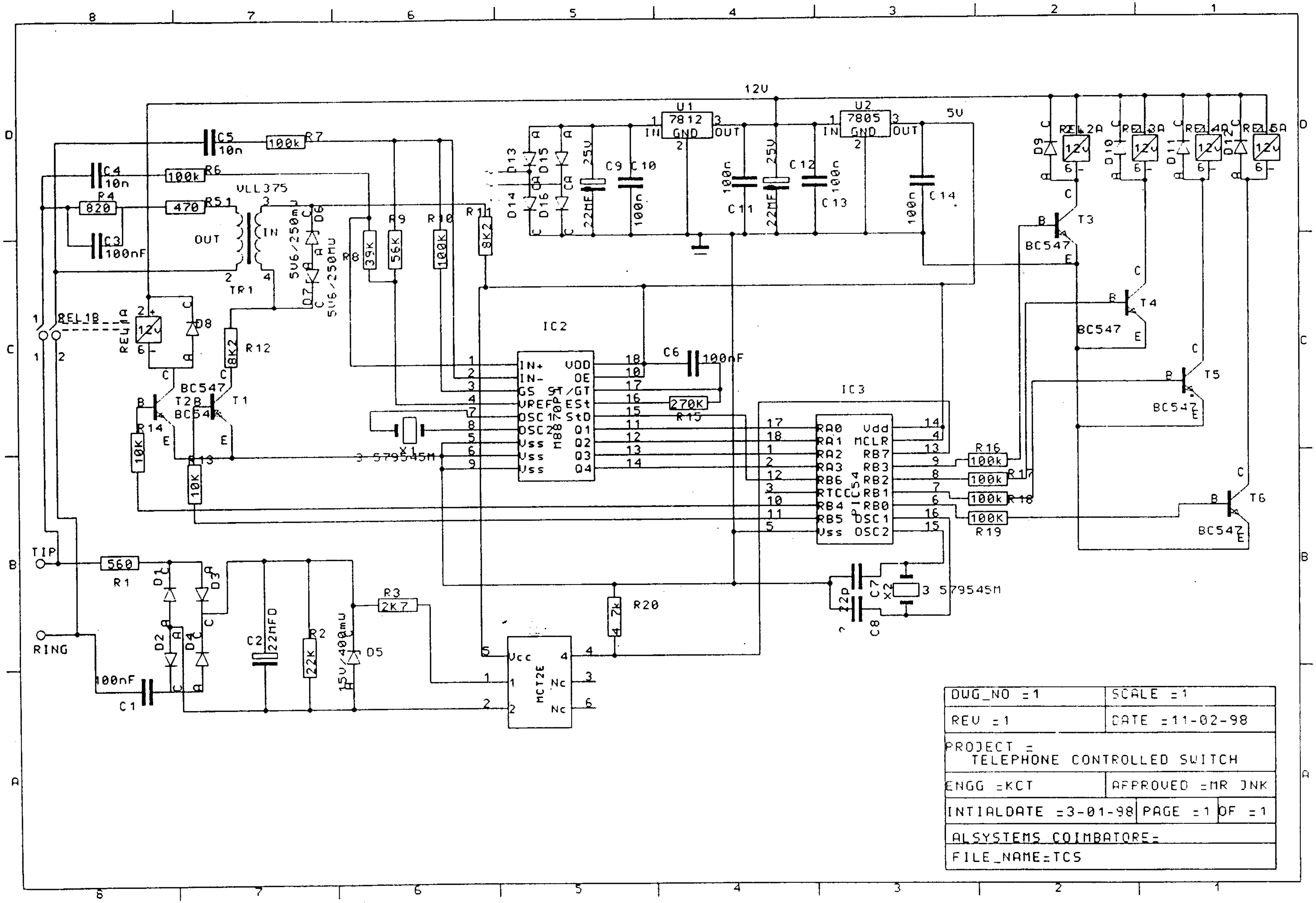
The ringing signal detector is relatively simple. The ring detector comprises the bridge rectfier, filter and opto isolater. The operation of the bridge recrfier is as follows. During the positive half cycle of the AC ring voltage diodes D1 and D2 conduct. During the other half cycle diodes D! & D2 conduct. The current throught the output of the bridge rectifer however flows in the same direction in both the halves of the AC ring voltage. Hence the output is esentially a pulsating DC.

The principal feature of the bridge rectifier which makes it superior to other rectifier is that since two diodes are present in series in each conduction path the peak inverse voltage is shared eqully by the two diode. Hence the bridge rectifier is eminently suited for high voltage applications.

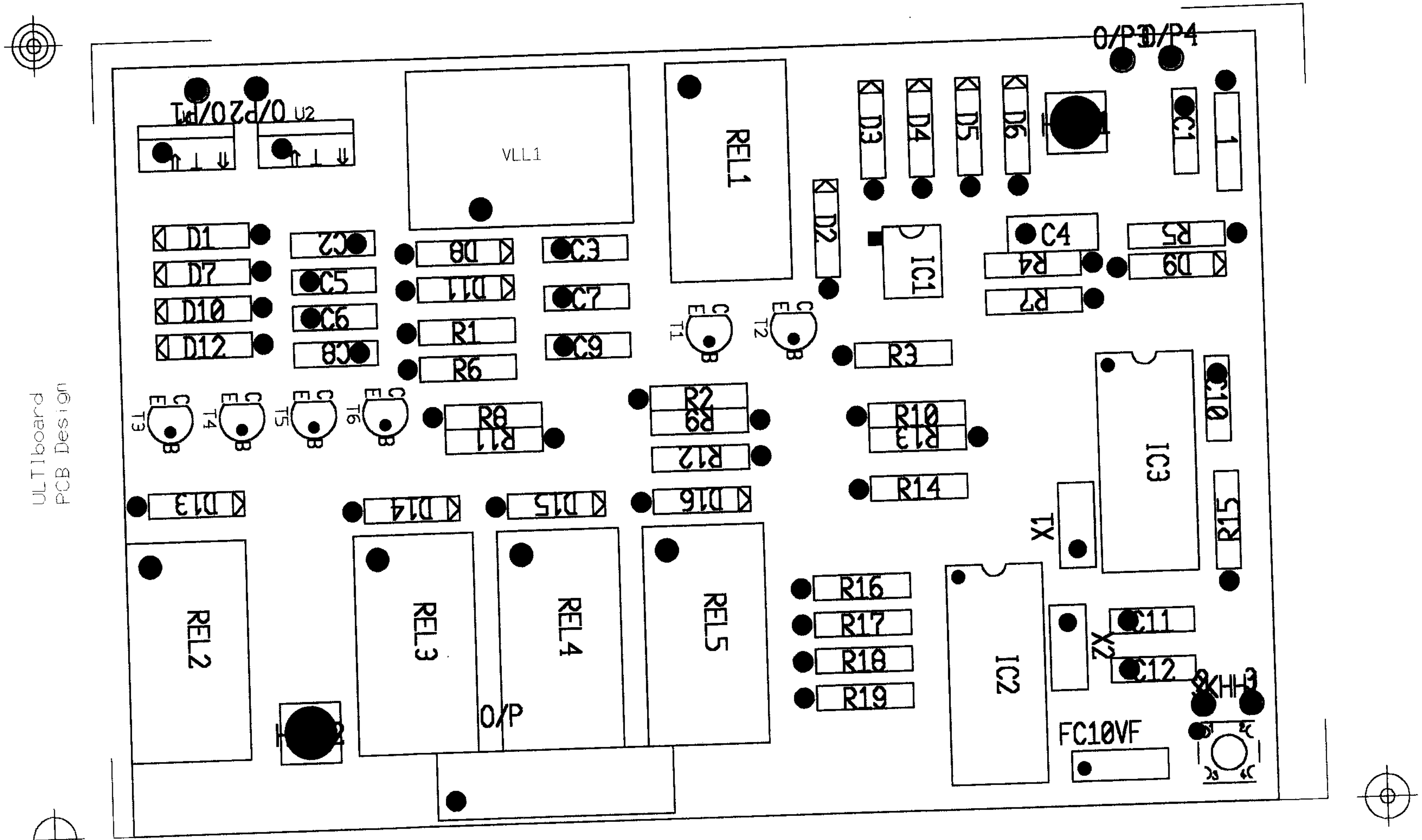
The next section in the ring detector is the filter section. It's a RC filter. It's essentially used to convert the pulsating DC output of the bridge rectifier to unidirectional DC. The zenes diode has been used to provide a constand output voltatge. The following section is the opto isolator. it is a six pin IC (MCT2E). The major function of the opto osolator is to isolate the telephone line ground from the digital ground. The resistors R1 and R3 are used as current limitng resistors. The 5v output of the optisolator is applied as an input to the RTCC pin of the PIC microcontroller. After the microcontroller counts the programmed number of ringing pulses, it responds by making PIN RB4 logic high which causes REla (RELIA) to be energised (via) R14. This means that the receiver is lifted (ie) the call has be answered consequently, the series network R4-R5 and the primary winding of the telephone line transformer TR, is connected to the A and B lines Via the relay contacts. The current flowing through this network is sufficiently large to maintain the connection. One end of the transformer secondary winding is connected to the positive supply voltage (via) R11. This means that rectangular voltages fenerated by the controller are coupled directly on to the telephone network lines. The two zener diodes D6 and d7 limit the voltage across the secondary winding to safe levels.

The received DTMF signals are capactively coupled to the decoder, Ic2. The external components that enable M_8870 DTMF decoder to operate reliably are limited to face resistors, a capacitor and a quarter crystal. The frequency of the quartz crystal is 3.579545 Mhz. The four decoder outputs q1,q2,q3,q4 supply a bit pattern that correspoonds to the received DTMF number. The four bit DTMF code is applied to the microcontroller via pins rao ra3.

The two user programmable parameters, the number of ringing signals and the acces code are convieniently stored in the memory of the PIC microcontroller. The bits on Pins Rbo to Rb3 are used to control the relays (Relay 2A,3A,4A,5A). The flyback diodes D9 to D12 protect the transistors (T3,T4,T5,T6) against back ef surges produced by the relay coils when these are switched off. The power supply is conventional and based on fixed voltage regulators.(7812, 7805). The 12v and 5v supply voltages used for the relay sections and the digital sections respectively are derived from a single mains transformer.

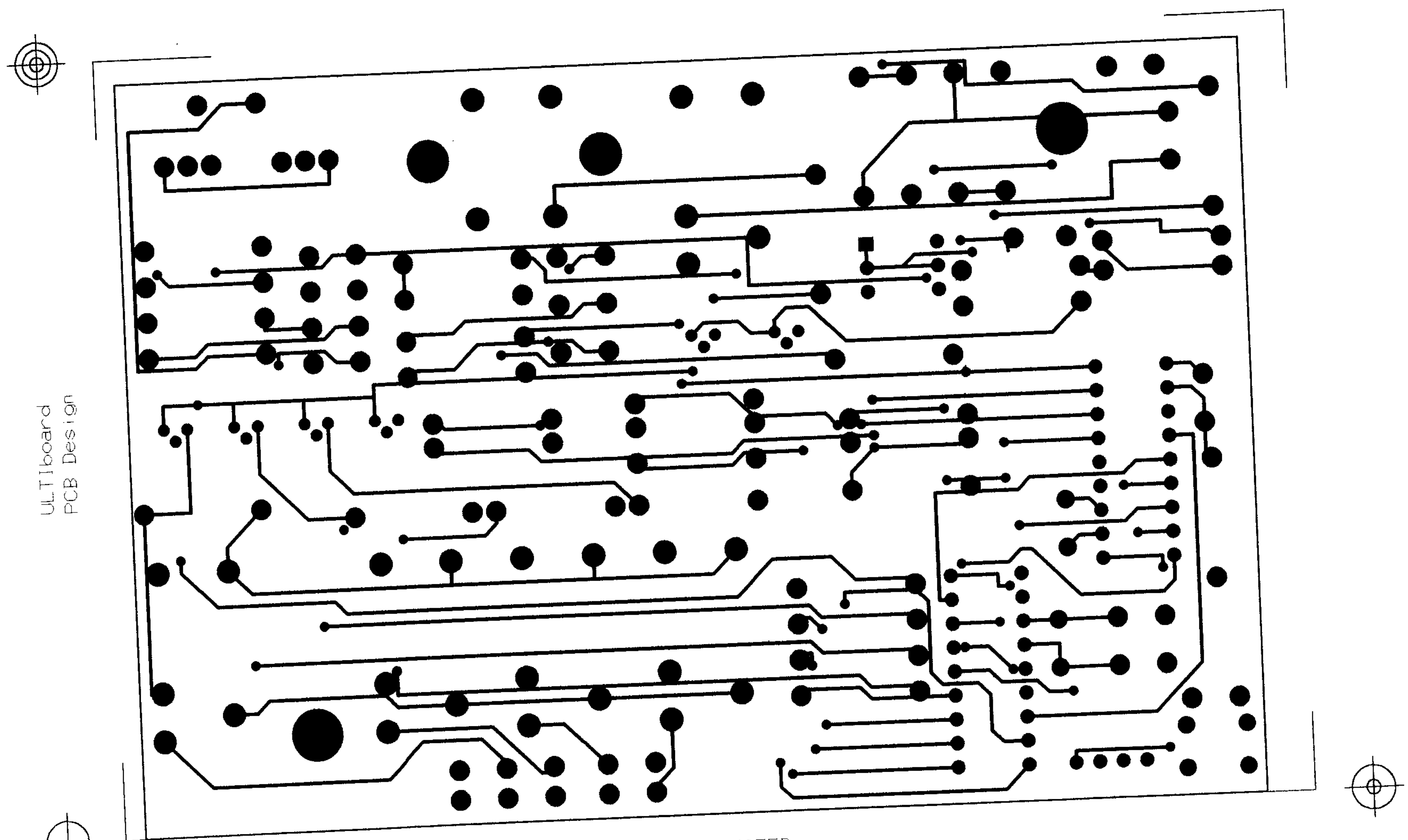


12V
5V
U1 7812
U2 7805
IN GND OUT
IC2
M8870P
IN+ IN- GS VREF OSC1 OSC2 Vss Vss Vss
VDD OE ST/GT ESt StD Q1 Q2 Q3 Q4
IC3
PIC54
RA0 RA1 RA2 RA3 RB6 RTCC RB4 RB5 Vss
Vdd MCLR RB7 RB3 RB2 RB1 RB0 OSC1 OSC2
MCT2E
Vcc Nc Nc
X1 3 579545M
X2 3 579545M
VLL375
TR1
OUT
IN
5V6/250mW
15V/400mW
BC547
T1 T2B T3 T4 T5 T6
REL1B REL2A REL3A REL4A REL5A
12V
TIP
RING
R1 560
R2 22K
R3 2K7
R4 820
R51 470
R6 100k
R7 100k
R8 39K
R9 56K
R10 100K
R11 8K2
R12 8K2
R13 10K
R14 10K
R15 270K
R16 100k
R17 100k
R18 100k
R19 100K
R20 4 7k
C1 100nF
C2 22MFD
C3 100nF
C4 10n
C5 10n
C6 100nF
C7 C8 22p
C9 22MF 25V
C10 100n
C11 100n
C12 22MF 25V
C13 100n
C14 100n
DWG_NO =1
SCALE =1
REV =1
DATE =11-02-98
PROJECT =
TELEPHONE CONTROLLED SWITCH
ENGG =KCT
APPROVED =MR JNK
INTIALDATE =3-01-98
PAGE =1
OF =1
ALSYSTEMS COIMBATORE-
FILE_NAME=TCS

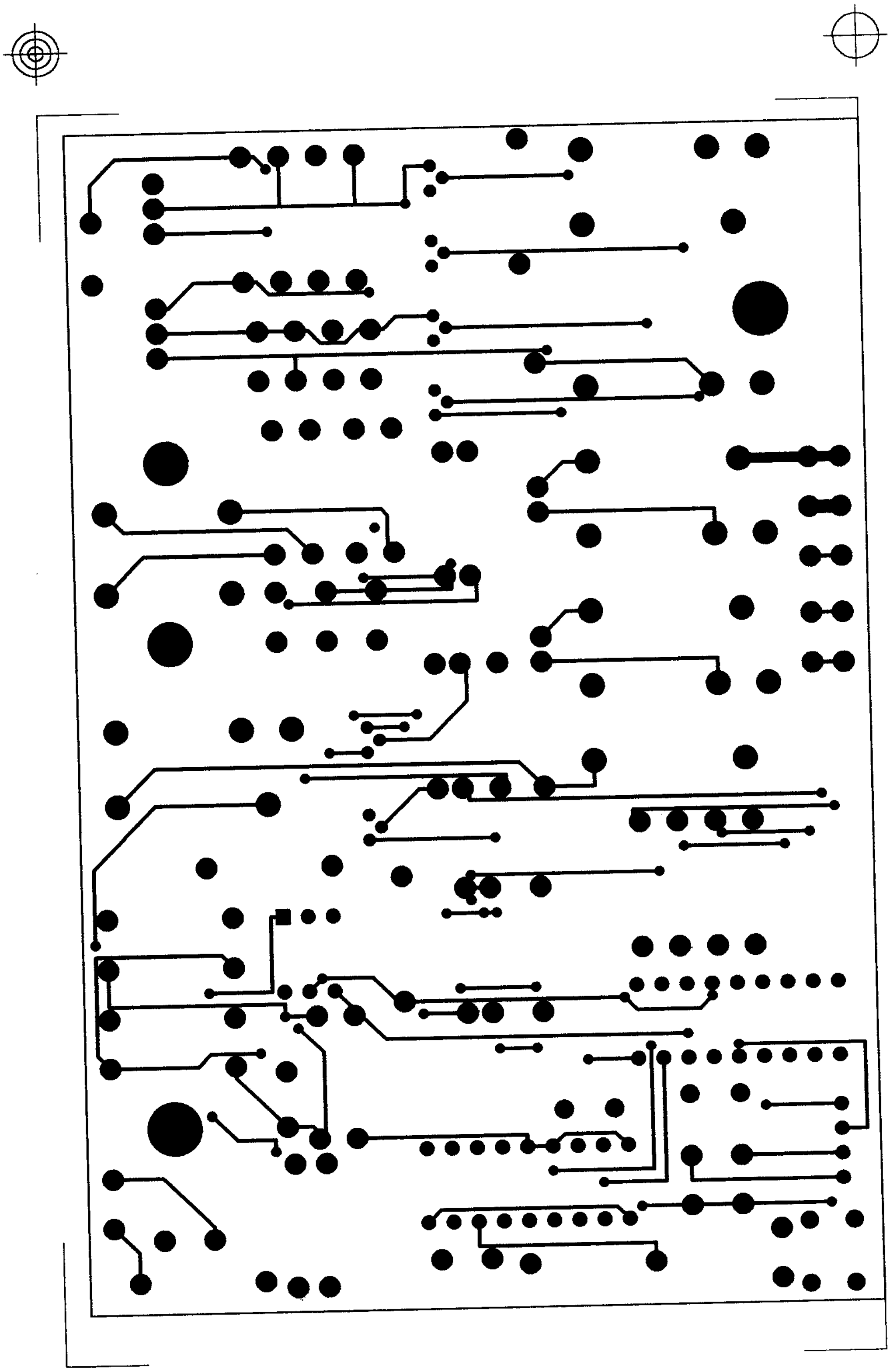


Tcs. Silkscreen (Apr 1, 1998) (04:21) (PCB) SCALE: 150% ROTATED
Drill Ref Pnt: 2.095, 0.000 (mm)

ULTIboard
PCB Design



Tcs, Top (Apr. 1, 1998) (04:21) (PCB) SCALE: 150% ROTATED
Drill Ref Pnt: 2.095, 0.000 (mm)



Tcs, Bottom (Apr 1, 1998) (04:21) (PCB) SCALE: 150% ROTATED REFLECTED
Drill Ref Pnt: 125.751, 0.000 (mm)

ULTIboard
PCB Design

6.1 ALGORITHM

1) Set the directions of ports A & B

2) Clear the contents of ports A & B

3) Wait for the ring

4) If the ring occurs, energise the hook switch relay which is equivalent to closing the subscriber loop.

5) Wait for the system access code (05h in this case)

6) If System Access Code received is incorrect, release hook switch relay
   If System Access Code received is correct, provide a long tone

7) Wait for device code

8) Valid device codes are 01h or 02h or 03h or 04h
   If any other code is sent, release the hook switch relay & go back to start

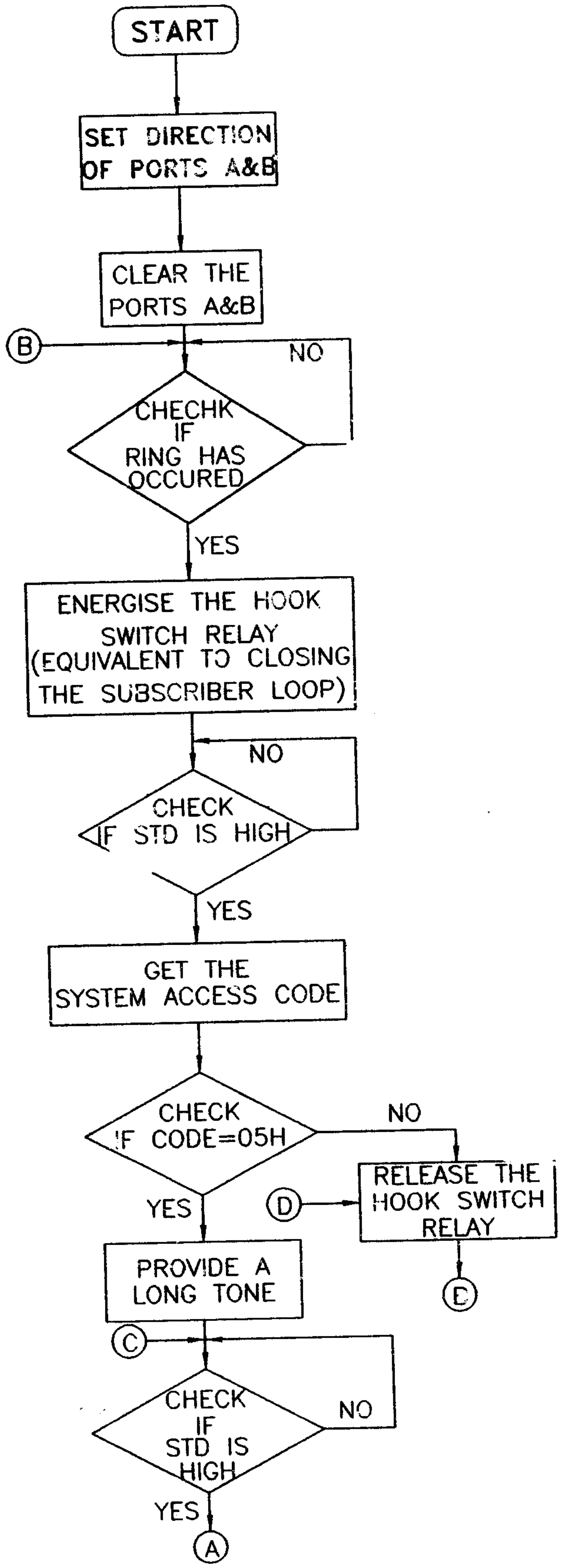
9) Wait for channel code

If the code is 00h,	switch off the device depending on device code
If the code is 01h,	switch on the device depending on device code
if the code is 02h,	provide long tone if the corresponding device is or provide short tone if the corresponding device is off.

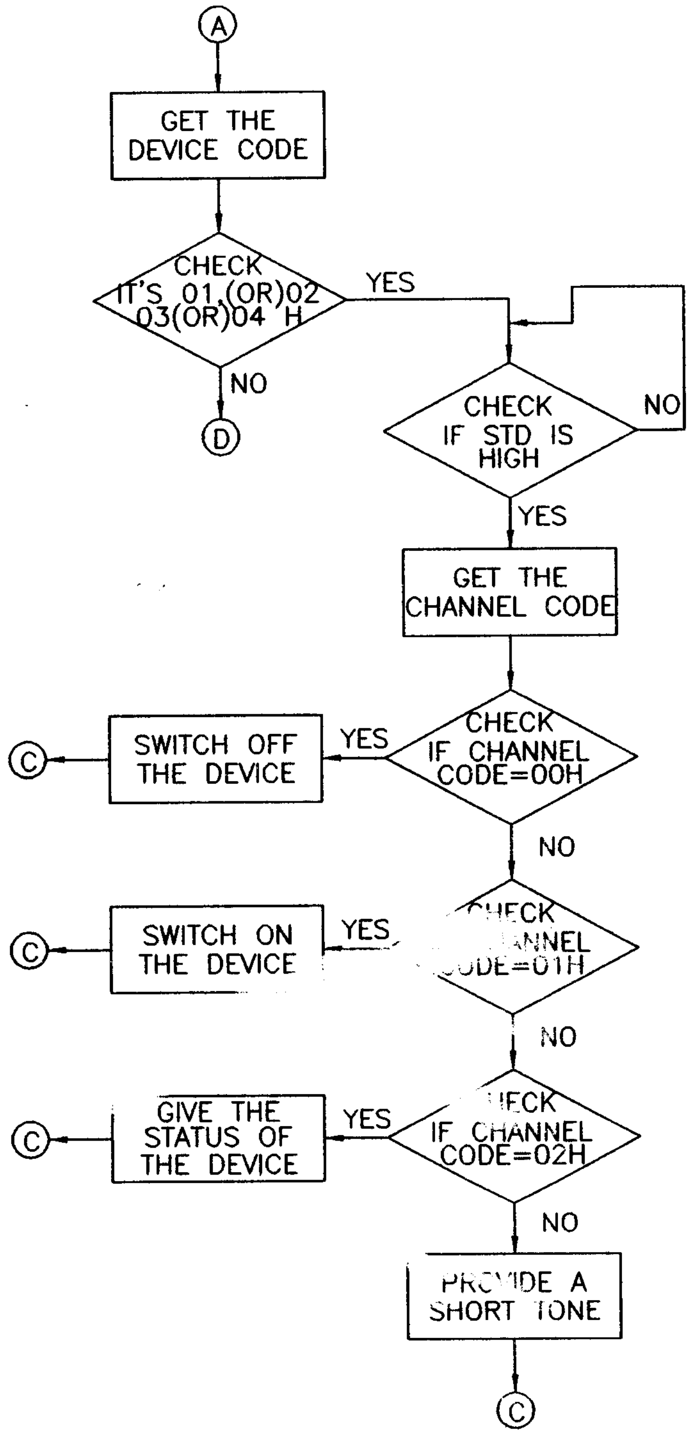
If any other code is sent provide a short tone and proceed the same steps from 7.

10) If there is no code sent for more than 10 seconds release hook switch relay and go to start.

FLOW CHART



START
SET DIRECTION OF PORTS A&B
CLEAR THE PORTS A&B
B
NO
CHECHK IF RING HAS OCCURED
YES
ENERGISE THE HOOK SWITCH RELAY (EQUIVALENT TO CLOSING THE SUBSCRIBER LOOP)
NO
CHECK IF STD IS HIGH
YES
GET THE SYSTEM ACCESS CODE
CHECK IF CODE=05H
NO
RELEASE THE HOOK SWITCH RELAY
D
E
YES
PROVIDE A LONG TONE
C
CHECK IF STD IS HIGH
NO
YES
A


A
GET THE DEVICE CODE
CHECK IT'S 01,(OR)02 03(OR)04 H
YES
NO
D
CHECK IF STD IS HIGH
NO
YES
GET THE CHANNEL CODE
CHECK IF CHANNEL CODE=00H
YES
SWITCH OFF THE DEVICE
C
NO
YES
SWITCH ON THE DEVICE
C
NO
CHECK IF CHANNEL CODE=02H
YES
GIVE THE STATUS OF THE DEVICE
C
NO
SHORT TONE
C

7.1 FUTURE SCOPE AND DEVELOPMENTS

This project work which has been designed for the remote control of the appliance can be extended and improved to control many more devices. The system access code used here is an one digit number. This code length can be increased in future.Another development which can be made in this projects is that the system access code may be changed using the # symbol on the telephone key pad.Thus this project will certainly prove to be an useful tool as an easy and accurate remote control in the years to come.

7.2 CONCLUSION

This remote control unit has been succesfully constructed and tested. The three electrical appliances, viz; light,fan, coffee maker have been interfaced via relays.This project" Microcontroller based remote control of electrical appliances through telephone lines has been successfully completed.This project can be used for switching on or off the electrical appliance in the absence of any person by making a call with the help of system access code.The report consists of the pic details the hardware and circuit description and the software details.

BIBLIOGRAPHY

1) Telecommunication switching principles - M.T. HILLS

2) Microchip data hand book.

3) PIC APPLICATION NOTE HAND BOOK.

4) G.K. MITHALElectronic devices and circuits.
Khana Publishes (Applied Electronics Vol I)

5) PIC tools manual 3.0 Parallax, Inc.,

PIC16Cxx Simulator

The PIC simulator is a piece of software that simulates the execution of PIC programs on the PC. Using a listing file from the assembler, the simulator "runs" PIC code on the screen. A single screen shows all of the PIC's registers, as well as the source code given in the listing file. Execution can be single-stepped, breakpoints can be set, and registers can be modified.

Many customers find the simulator very useful for preliminary testing and debugging.

```
                                  PIC 16C54 Simulator  v2.07
           HEX      BINARY        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
STACK 1    000   00000000000   0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
STACK 2    000   00000000000   1 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
OPTION      FF      11111111   3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
W          000   00000000000   5 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
RTCC       000   00000000000   7 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PC         1FF   00111111111          LATCH              PIN               TRI-STATE
STATUS      18      00011000   PORT A  0F  00001111   00  00000000     0F  00001111
FSR         E0      11100000   PORT B  FF  11111111   00  00000000     FF  11111111

 MCLR RTCC PA2 PA1 PA0 TO PD  Z DC  C   WD       CYCLES      TIME       XTAL
   1    0   0   0   0   1  1  0  0  0  0.0183  00000000  0.0000000   8.00MHz
005                           retw    $                        ;current origin va
006                           djnz    file_register,$          ;use $ for delay 1
007
008                           jb      bit_address,$
009
00A                           setb    file_register.7          ;bit address funct
00B                           mov     w,#expression&7Fh        ;AND function
00C                           or      w,#expression|40h        ;OR function
00D                           sub     file_register,#'X'^80h   ;XOR function
00E
00F                           inc     file_register+1          ;ADD function
F1-HELP F2-BRKPT F3-CLEAR F4-HERE F5-TIME F6-GO F7-STEP F8-NEXT F9-RUN F10-RST
```

Although not discussed in this chapter, the simulator can also be used with our I/O simulator and in-circuit emulator hardware products. In both cases, the simulator can exercise a real circuit designed to accept a PIC chip. For more information on these products, please refer to the appropriate chapters in this manual.

Running the Simulator

To run the simulator, type the following command at the DOS prompt:

PSIM *filename*	Runs the simulator and loads *filename.lst*.

The simulator uses the listing file that is optionally produced by the PIC assemblers. The extension ".lst" can be omitted, since it's assumed by default. If, however, an extension is given, it will be used by the simulator.

The simulator loads the listing file and displays the source code in the lower half of the screen. The PIC's registers are displayed in the upper half of the screen. The bottom line of the screen lists the function key commands.

Scrolling Through Source Code

You can scroll through the source code displayed in the lower half of the screen by using the up and down arrow keys, as well as PGUP, PGDN, HOME and END. The HOME key brings you to the beginning of the address space, while the END key brings you to the end.

Two highlighted lines are always present in the source code display. The "current line" indicates the current line being executed, while the "marker line" is used for special functions, such as setting breakpoints.

The current line is indicated by blue text on a grey background. The current line moves from line to line as your program runs. Any change to the Program Counter (PC register) will move the current line.

The marker line is indicated by yellow text on a black background. The marker line does not move; instead, it remains in the middle of the source code display. By scrolling the source code up and down, you can position a particular line of code in the marker line. When a line is in the marker line, special functions can be performed on the line, such as setting a breakpoint.

Device Type

The simulator determines the device type being simulated by a variety of methods. The recommended method is to include the DEVICE directive in your source code. The simulator locates the directive in the listing file and sets the device type accordingly. If the directive is not present in the file, the simulator determines the device type by the type of listing file and the amount of program space used by the source code. If there are no instructions above address 1FFh hex, the PIC16C54 is assumed. If there are instructions higher than 1FFh, but not higher than 3FFh, the PIC16C56 is assumed. If there are instructions above 3FFh, then the PIC16C57 is assumed.

Another method of selecting the device type is to use the command-line option "/d=". Examples are as follows (they are all equivalent):

PSIM *filename* /d=PIC16C54
PSIM *filename* /d=PIC54
PSIM *filename* /d=54

The final method of selecting the device type is to use the Alt-D command from within the simulator. Pressing Alt-D will pop up a menu where you can use the up and down arrow keys to select the desired device. Press RETURN when the desired device is highlighted.

When loading files into the simulator, keep in mind that the PIC16C5X and new PICs ('71, '84,...) do not have compatible object code. If the device type is set to PIC16C71 while simulating PIC16C5X code, unexpexted results will occur.

Crystal Frequency

The simulator uses a simulated crystal value to calculate the instruction cycle time. The crystal value can be entered by one of two methods. The first method is to enter it in the command-line after the filename using the "/x=" option. Possible suffixes are "Mhz", "khz", and "hz". If no suffix is entered, the frequency is assumed to be in hertz (cycles/second). Some examples are as follows:

PSIM *filename* /x=4Mhz

PSIM *filename* /x=32.768khz
PSIM *filename* /x=32768hz
PSIM *filename* /x=32768

The second method of setting the crystal value is to press Alt-C from within the simulator and then type in a new value. See the Alt-C later in this chapter for further details. If no crystal frequency is entered, the simulator will default to 8 Mhz upon startup.

Setting Breakpoints

As mentioned at the beginning of this chapter, the highlighted line in the middle of the source code display is the marker line. To set a breakpoint, scroll through the code until the desired line is highlighted by the marker, and then press F2. The line will be highlighted in red, which indicates a breakpoint. To clear the breakpoint, press F2 again.

There is no limit to the number of breakpoints you can set.

Modifying Registers

To modify a register during simulation, you may use one of two methods. If your computer has a mouse, you can move the mouse cursor until the desired register is highlighted, then press the left mouse button to increment the register contents or the right mouse button to decrement it. The upper and lower nibbles of a register can be incremented or decremented separately for registers that are displayed in hex. For registers that are displayed in binary, you can change each bit separately. Some of the registers cannot be changed. For instance, you can't change the indirect address register (00h). The indirect address register is not physically implemented in the PIC, so altering it would have no effect. Also, if the selected device is a PIC16C54 or '55, you can't change the upper two bits of the program counter, since these devices do not implement these bits. For the '56, you can't alter the upper bit of the program counter.

If you do not have a mouse, you can select a register by pressing and holding the CTRL key while using the arrow keys to move the cursor. The cursor will move from register to register as you press the arrow keys. When the cursor is on the desired register, release the CTRL key and type the desired value for the register. If you are modifying a register that is displayed in hex, you can type any valid hex character. If you are modifying a register that is displayed in binary, you can only type in 0's and 1's. You must type in all the numbers that are highlighted before the new value will be accepted. The cursor will disappear when you resume simulation.

Watchdog Timer

The watchdog timer may be enabled or disabled using one of two methods. The first is to include the WDT_ON or WDT_OFF option in the DEVICE directive in your source code. The second method is to press Alt-W from within the simulator. A menu will pop up, allowing you to select "enable" or "disable" with the up and down arrow keys. Press RETURN when the proper selection is highlighted.

Exiting the Simulator

To exit the simulator, press escape ESC or Alt-X. You will be asked to verify your decision. Press "Y" to quit or "N" to return to the simulator.

Function Keys

The use of each function key is described in the following text:

F1 **Help menu.** This displays the help menu. Use the PGDN key to display the second page of the help menu.

F2 **Toggle breakpoint.** Toggle breakpoint at the marker line. To insert a breakpoint at a specific line, use the cursor keys to scroll the program display up and down until the desired line is highlighted by the marker, and then press F2. The line will then turn red to indicate that a breakpoint is set. Pressing F2 again on the same line will clear the breakpoint.

F3 **Clear all breakpoints.**

F4 **Execute to marker.** Execute code until the marker line is reached. Pressing F4 will simulate the program line-by-line until the marker is reached. "Running..." will appear at the bottom of the screen. When the line is reached, the screen will be updated and the registers that were altered will be highlighted. To stop the simulation while it's running, press any key.

F5 **Reset time.** Reset the real-time display to zero. This is useful for timing code execution.

F6 Execute code (with update). Pressing F6 causes the simulator to start executing code. It will not stop until a key is pressed or a breakpoint is reached. *The screen is updated and changes are highlighted after each line is executed.*

F7 **Step line-by-line.** This causes one line to be executed. Changes in the registers are highlighted after the line is executed. Pressing the space bar also executes one line.

SPACE Step line-by-line. Same as F7.

F8 **Execute to next line.** Pressing F8 causes the simulator to execute code until the next line is reached or a breakpoint is reached. This is useful for executing through subroutine calls. If F8 is pressed when the cursor is on a subroutine call, the simulator will execute the code until the line after the call is reached. The user can stop execution at any time by pressing any key. If the F8 key is pressed when the highlighted line is at the end of the address space (1FFh for the PIC16C54, etc.), the simulator will never reach the next line, since it is out of code space. In this case, the simulator will continue running until a key is pressed.

F9 **Execute code (without update).** Pressing F9 causes the simulator to start executing code. It will not stop until a key is pressed or a breakpoint is reached. *The screen is not updated until execution is stopped.*

F10 Reset PIC. Pressing F10 simulates a hardware reset.

Alt-C **Set crystal frequency.** The crystal frequency defaults to 8 Mhz, which is used to calculate execution time. To enter a new frequency, press Alt-C. A window will pop up, allowing you to enter the desired value. You can enter the value in hertz (hz), kilohertz (khz), or megahertz (Mhz) by typing the value and the appropriate suffix. If no suffix is entered, the frequency is assumed to be in hertz. The suffix can be upper or lower case. Some examples are: 32768, 32768hz, 32.768khz, 1Mhz, 3.57Mhz.

Alt-D **Select device type.** To select a different device type, press Alt-D. A menu will appear, allowing you to select the desired device. Press ENTER when the desired PIC is highlighted.

You must be careful when using this method. Any code above the address limit of the newly selected device will be lost. Also, keep in mind that the PIC16C5x and new PICs ('71, '84,...) are not object code compatible. If the device is set to PIC16C71 while simulating PIC16C5x code, unexpexted results will occur.

To exit from the device menu without changing the device type, press ESC.

Alt-E **Display EEPROM data.** If the simulated PIC has EEPROM (currently, the '84), you can view the contents of the EEPROM data memory by pressing Alt-E. This is not available when any other PIC is selected.

Alt-F **Load file.** To load a different listing file, press Alt-F. A file menu will appear, allowing you to select a file to load. Listing files are displayed in yellow and directories are in red (only files with a '.lst' extension are shown). To load a file, use the cursor keys to highlight the desired file and then press ENTER. If you need to load a file from a different directory, move the cursor to the desired directory and then press ENTER. The display will list the files in the selected directory for you to choose from. To quit the file menu without loading in a new file, press ESC.

Alt-S **Display stack.** The first two stack locations are displayed on the screen at all times. If the PIC being simulated is a new PIC ('71, '84,...), you can display all stack locations in a pop up window by pressing Alt-S.

Alt-W **Watchdog timer.** To enable/disable the watchdog timer, press Alt-W. A menu will pop up, allowing you to select "enable" or "disable" with the up and down arrow keys. Press RETURN when the proper selection is highlighted.

Alt-F4 **Jump to marker.** Pressing Alt-F4 causes the simulator to jump to the marker line, without executing any instructions. This is useful for jumping directly to code segments that you wish to debug, without having to run through the normal code execution to get there. It's also useful for jumping over long delay loops in your code.

Alt-F7 **Back step.** Pressing Alt-F7 steps backward one line in your code. The entire state of the PIC is stored in a 100-step history buffer. By recalling the history buffer contents, you can step back through your code (up to 100 steps).

Alt-X **Exit simulator.** Pressing Alt-X exits the simulator and returns to DOS. A pop-up prompt will allow you to confirm the exit command.

Advanced Features

Journal Files

All keystrokes entered during simulation are automatically stored in a journal file called PSIM.JRN. This file can be "played back" to re-create a simulator session.

The journal file is automatically created during each simulator session (unless disabled with the "/j-" option when starting the simulator). When the simulator is started, any old journal file will be overwritten. If you want to retain a journal file for later use, rename it or copy it to another directory.

To re-run a journal file, start the simulator by typing the following:

PSIM *filename* /j=*journalfile*

where *journalfile* is the name of the journal file you want to use. This will load *filename* and then execute the keystrokes read from the journal file.

The journal file feature is useful for configuring your simulation session upon startup. For instance, you can initialize an I/O port to a certain value or set breakpoints in your code. This setup can then be executed upon startup of later simulation sessions.

Note that the journal file only records keystrokes – mouse actions are not stored in the file.

Input Stimulus Files

The stimulus file feature allows you to schedule bit changes in I/O port pins at specified times during simulation. This scheduling is controlled via a text file called a *stimulus file*. The "/i=filename" command-line option is used to read the stimulus file.

The stimulus file is an ASCII text file that contains instructions for applying stimuli to the I/O pins during simulation. All commands must be comma separated. White spaces and tabs are ignored. Any line beginning with a semicolon is a comment line.

There are five schedule commands that tell the simulator when a given stimulus should be applied. The five commands are:

t = real	Provide stimulus when execution time reaches real.
c = int	Provide stimulus when cycle counter reaches int.
dt = real	Provide stimulus when delta time reaches real.
dc = int	Provide stimulus when delta cycles reaches int.
pc = int	Provide stimulus when program counter reaches int.

After the schedule command comes the stimulus command. Stimulus commands can be applied to an individual I/O pin or to a full port. They can also be applied to the RTCC pin or to any analog inputs (when simulating a PIC with A/D).

Up to eight bit changes or three port changes can be present on a single line in the stimulus file. The following are valid stimulus commands:

;This is a comment line. It has no effect on the simulator.

pc=8, ra=2Dh, rb=FEh	When the program counter reaches 8, port A will be set to 2Dh and port B will be set to FEh.
t=0.00002, ra0=1, ra3=0	When execution time reaches 0.00002 seconds, bit 0 of port A will be set and bit 3 will be cleared.
c=50, rb0=1	When the cycle counter reaches 50, bit 0 of port B will be set.
dt=0.00001, rb7=1	This sets bit 7 of port B when dt=0.00001 seconds. In this example, it will occur 0.00001 seconds after the last stimulus.
dc=12, rtcc=1	Twelve cycles after the last stimulus, this sets the RTCC pin.
dc=5, ain2=4.80v	Five cycles after the last stimulus, this sets analog input #2 to 4.8 volts.

Command-line Options

The simulator has various options that can be invoked when it's started. These options are shown below:

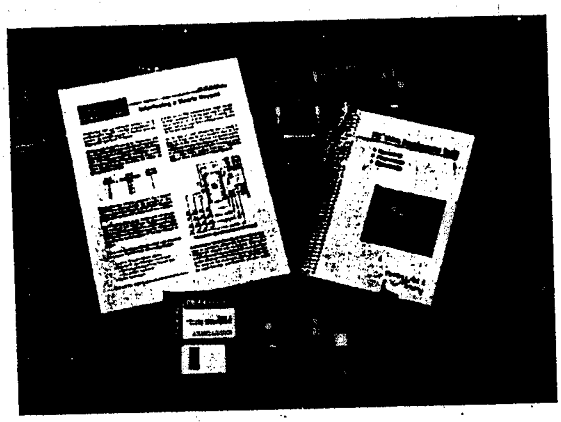
PSIM *filename*	Runs the simulator and loads *filename.lst*.
PSIM *filename.xxx*	Runs the simulator and loads *filename.xxx*.
PSIM *filename* /d=*device*	Runs the simulator and loads *filename.lst*. Device type is set to *device*, where *device* is "16C54", "16C55", "16C71", etc.
PSIM *filename* /e-	Runs the simulator and loads *filename.lst*. Disables serial port checking for an emulator, such as ClearView 5x.
PSIM *filename* /e=*n*	Runs the simulator and loads *filename.lst*. Causes the simulator to only look for an emulator on serial port #n.
PSIM *filename* /f-	Runs the simulator and loads *filename.lst*. Disables serial port checking for an I/O simulator, such as Reflection 5x.
PSIM *filename* /f=*n*	Runs the simulator and loads *filename.lst*. Causes the simulator to only look for an I/O simulator on serial port #n.
PSIM *filename* /i=*stimfile*	Runs the simulator and loads *filename.lst*. Then loads and runs the stimulus file called *stimfile*.

PSIM *filename* /j-	Runs the simulator and loads *filename.lst*. Disables saving of keystrokes in journal file.
PSIM *filename* /j=*journalfile*	Runs the simulator and loads *filename.lst*. Then executes keystrokes from *journalfile*.
PSIM *filename* /r	Runs the simulator and loads *filename.lst*. Causes the simulator to start with registers in a random state, which more accurately simulates a real PIC.
PSIM *filename* /s	Runs the simulator and loads *filename.lst*. Disables normal stack overflow and underflow error detection.
PSIM *filename* /x=xtal	Runs the simulator and loads *filename.lst*. Crystal freq. is set to *xtal*, where *xtal* is "4mhz", "3.528khz", "1000hz", etc.
PSIM *filename* /?	Display help menu which shows valid command-line options.

PIC16C5x Downloader

The PIC16C5x Downloader functions as a ROM emulator for PIC16C5x devices. It plugs in place of a '5x PIC in your target system and runs your code at 8 MHz. A simple 2-second download replaces the hassle of programming, erasing, and removing parts.

The downloader does not have debugging features, such as stepping and breakpoints. However, many customers find it useful when testing code in-circuit, especially if the code requires multiple changes to get it just right (such as adjusting timing values, etc.).



System Requirements

To use the PIC16C5x Downloader, you will need the following items:

- IBM PC or compatible computer
- 3.5-inch disk drive
- Parallel port
- 128K of RAM
- MS-DOS 2.0 or greater

If you plan to write your own PIC programs, you will also need the following software:

- Text editor or word processor capable of saving ASCII files

Packing List

The downloader package should contain the following items. If any are missing, please let us know.

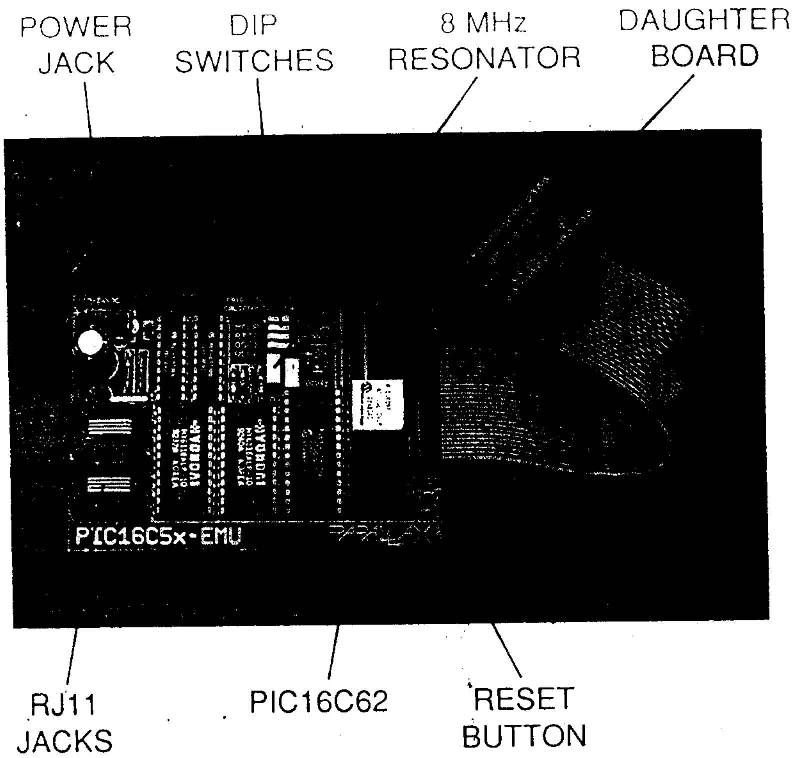
- Downloader PC boards (main board and daughter board)
- Power supply*
- DB25-to-RJ11 adapter
- 26-conductor ribbon cable
- 4-conductor telephone cable (7 feet)
- Male-to-male pin strips (two 5-pin strips and two 9-pin strips)
- PIC Tools diskette

* Power supplies are only shipped with orders to the United States and Canada. If your order was shipped to another country, you will need to obtain power supplies with the proper output voltage and current:

12 VAC or 16 VDC, 250 mA

PIC16C5x Downloader

Hardware Features



Power Jack: Accepts power from external power supply. A "wall pack" power supply is included with orders shipped to the United States and Canada.

RJ11 Jacks: Connect to PC parallel port via 4-conductor telephone cable.

PIC16C62: PIC emulator chip. Emulates PIC16C5x microcontrollers, up to 8 MHz.

8 MHz Resonator: If enabled by DIP switches, the resonator provides the clock source for the PIC16C62.

DIP Switches: Enable/disable on-board clock, MCLR pull-up resistor, and RTCC pull-up resistor.

Reset Button: Resets the downloader.

Daughter Board: Connects to downloader via ribbon cable. Plugs into the socket normally occupied by a PIC in your target system.

Downloader Installation

To install your downloader, follow these steps:

1) Plug one end of the 7-foot telephone cable into the DB25-to-RJ11 adapter.
2) Plug the DB25-to-RJ11 adapter into an available PARALLEL port on your PC.
3) Plug the other end of the telephone cable into one of the downloader's RJ11 jacks.
4) Plug the power supply into an AC outlet.
5) Plug the power supply cord into the downloader's power jack.
6) Plug one end of the 26-conductor ribbon cable onto the downloader's 26-position dual-row header. The cable should extend to the right of the downloader.

   You may want to make sure that the cable's colored conductor is near the copyright notice on the main board; in doing so, the colored conductor will indicate pin 1.
7) Plug the remaining end of the 26-conductor cable onto the daughter board. The daughter board should be oriented the same way as the main board, with no flips in the ribbon cable.
8) Plug the appropriate pin strips into the underside of the daughter board. Use two 9-pin strips to make an 18-pin PIC or all four strips to make a 28-pin PIC. The exposed ends of the strips will plug into the socket normally occupied by a PIC in your target system.

Oscillator Information

The downloader has 4-MHz and 8-MHz resonators. The resonators are small, 3-pin, blue devices that contain a ceramic resonator and two capacitors (each connected to ground). The center pin is ground, and the two outer pins are OSC1 and OSC2.

The two resonators are located below the DIP switches. The resonator on the right is soldered into the PC board and runs at 4-MHz; the resonator on the left is socketed and runs at 8-MHz.

The 4-MHz resonator runs the downloader control circuitry, and has nothing to do with the execution speed of your code.

The 8-MHz resonator, however, determines execution speed. For different execution speeds, the resonator can be replaced with other resonators. Typically speaking, resonators can be found in frequencies from 1-MHz to 8-MHz. Most 3-pin resonators will work, and Digi-Key usually has a good supply (in the U.S., call Digi-Key at 1-800-344-4539).

In addition to resonators, crystals and TTL signals can be used to clock the downloader. To use a crystal, plug it into the two outer contacts of the resonator socket. Please note, however, that a crystal will require two 15 pF capacitors (one from each crystal pin to ground). As for a TTL clock signal, one can be used if it is connected to the downloader's OSC1 input (the top contact on the resonator socket).

In many low-cost designs, an RC (resistor-capacitor) oscillator is used with a PIC. Unfortunately, the downloader will not work with RC oscillators. This is because the 48-pin emulation chip (the heart of the downloader) does not support RC oscillators.

Everything that has been mentioned so far also applies to an oscillator on your target system. An oscillator on your target system will be connected to the downloader via the ribbon cable. Therefore, if you use the target's oscillator, the downloader's on-board oscillator should be disabled using SW3 and SW4. Conversely, if you use the on-board oscillator, any target system oscillator should be disabled.

DIP Switch Settings

The downloader has four DIP switches that control an MCLR pull-up resistor, RTCC pull-up resistor, and on-board oscillator. The function of each switch is shown below:

SW1 **MCLR pull-up:** enables/disables MCLR pull-up resistor. Many PIC designs call for a pull-up resistor on MCLR to ensure a stable reset line. Turning the switch ON enables the pull-up resistor.

SW2 **RTCC pull-up:** enables/disables RTCC pull-up resistor. The pull-up resistor can be used to stabilize the RTCC input. Turning the switch ON enables the pull-up resistor.

SW3 **OSC1:** enables/disables OSC1 connection to on-board oscillator. For convenience, you can turn this switch ON to use the downloader's on-board oscillator. However, if you want to use the oscillator on your target system, this switch should be turned OFF.

SW4 **OSC2:** same as SW3, but applies to OSC2 connection.

Both SW3 and SW4 must be turned ON to use the on-board oscillator. Either switch can be turned OFF to disable the oscillator.

Power Supply Considerations

The downloader provides 5 volts on the target system ribbon cable. This can be used to power a small target system, but the current draw of the target should not exceed 100 mA.

If the target system has its own power supply, care should be taken to avoid having the downloader and target power supplies on at the same time.

If both systems are connected and providing power, damage to the downloader and/or target system may result.

Running the Software

To run the downloader software, type

PEP

from the DOS prompt. After several seconds, you'll see the screen shown below:

```
       DEVICE: 54 55 56 57 58    PIC16C5x-PGM/EMU
   OSCILLATOR: RC HS XT LP       (C)1994 Parallax
     WATCHDOG: ON OFF
 CODE PROTECT: OFF ON      ID: FFFF CHECKSUM

 000- FFF FFF FFF FFF FFF FFF FFF FFF  ........
 008- FFF FFF FFF FFF FFF FFF FFF FFF  ........
 010- FFF FFF FFF FFF FFF FFF FFF FFF  ........
 018- FFF FFF FFF FFF FFF FFF FFF FFF  ........
 020- FFF FFF FFF FFF FFF FFF FFF FFF  ........
 028- FFF FFF FFF FFF FFF FFF FFF FFF  ........
 030- FFF FFF FFF FFF FFF FFF FFF FFF  ........
 038- FFF FFF FFF FFF FFF FFF FFF FFF  ........

   HEX ENTRY       ASCII ENTRY       FILL ( ↵ )

 PROGRAM             EMULATE              ESC EXIT
 VERIFY
 READ                LOAD:
 BLANK CHECK         SAVE:
```

The software will automatically adjust to the type of display you are using. However, if you are using a laptop computer that has a monochrome display, you may have to tell the software to use monochrome attributes. To do this, type PEP /m from the DOS prompt.

The following pages describe the functions available from this screen. To exit the software, press ESC.

Device-Specific Options

There are several options at the top of the screen that are specific to the device you're using. These options are listed on below:

Device:	Pressing 'D' selects the type of device you're using. Possible settings are 54, 55, 56, and 57.
Oscillator:	Pressing 'O' selects the type of oscillator you intend to use with the device selected. Possible settings are RC, HS, XT, and LP.
Watchdog:	Pressing 'W' toggles the PIC's watchdog timer. Possible settings are ON and OFF.
Code Protect:	Pressing 'C' toggles the device's code protect bit. Possible settings are ON and OFF. If code protect is on, you will not be able to read the device after programming. *Code protect has no effect on the downloader.*
ID:	Pressing 'I' toggles the device ID between a checksum and a 2-byte hex value.

Sending to the Downloader

To send the current screen data to the downloader, press 'E' (for emulate). The program data, device type, oscillator type, watchdog timer status, and ID data will be sent to the downloader. During the process, the software will display an incrementing value next to the word "emulate". However, since the process only takes a second or two, you may only see a blur as the value appears and then disappears.

When the downloader has received all of the data, it will immediately start executing its new program at whatever clock speed you have selected (usually 8 MHz).

Loading a File from Disk

To load an object file from disk, press 'L'. Blinking arrows will appear to the right of the word "Load". These arrows indicate that you may enter a filename:

```
       DEVICE: 54 55 56 57 58   PIC16C5x-PGM/EMU
   OSCILLATOR: RC HS XT LP      (C)1994 Parallax
     WATCHDOG: ON OFF
 CODE PROTECT: OFF ON      ID: FFFF CHECKSUM

 000- FFF FFF FFF FFF FFF FFF FFF FFF  ........
 008- FFF FFF FFF FFF FFF FFF FFF FFF  ........
 010- FFF FFF FFF FFF FFF FFF FFF FFF  ........
 018- FFF FFF FFF FFF FFF FFF FFF FFF  ........
 020- FFF FFF FFF FFF FFF FFF FFF FFF  ........
 028- FFF FFF FFF FFF FFF FFF FFF FFF  ........
 030- FFF FFF FFF FFF FFF FFF FFF FFF  ........
 038- FFF FFF FFF FFF FFF FFF FFF FFF  ........

   HEX ENTRY        ASCII ENTRY        FILL ( ↵ )

 PROGRAM            EMULATE               ESC EXIT
 VERIFY
 READ               LOAD:>>FILE.OBJ             <<
 BLANK CHECK        SAVE:
```

When you have entered the filename, or if the current filename is correct, press RETURN. The software will then attempt to load the specified file. If an error occurs while loading the file, the appropriate message will be displayed.

Saving a File on Disk

To save the buffer data and device options on disk, press 'S'. Blinking arrows will appear to the right of the word "Save". These arrows indicate that you may enter a filename.

When you have entered the filename, or if the current filename is correct, press RETURN. The software will then attempt to save the file. If the file already exists, it will be overwritten. If an error occurs while saving the file, the appropriate message will be displayed.

Saved files contain the buffer data, as well as device type, oscillator type, watchdog status, power-up timer status (if PEPX), code protect status, and device ID.

Moving Around in the Buffer

To move the buffer's cursor, the normal editing keys are used. The effect of each key is shown below:

Left arrow	Moves the cursor 1 location to the left.
Right arrow	Moves the cursor 1 location to the right.
Up arrow	Moves the cursor 1 line up (8 locations).
Down arrow	Moves the cursor 1 line down (8 locations).
PGUP	Moves the cursor 8 lines up (64 locations).
PGDN	Moves the cursor 8 lines down (64 locations).
HOME	Moves the cursor to the first location in the buffer.
END	Moves the cursor to the last location in the buffer.

Editing the Buffer

There are three methods available for editing the buffer. These methods are described on the following page:

Hex entry: Pressing 'H' invokes the hex entry mode. In this mode, any valid hex values that you type are entered into the buffer at the current cursor location. To exit hex entry mode, press RETURN or ESC.

ASCII entry: Pressing 'A' selects the ASCII entry mode. In this mode, any keys that you type are entered into the buffer as 3-digit values. These 3-digit values are comprised of the number '8' and the ASCII value for the key typed. For instance, if you type 'X', the value '878' will be entered into the buffer (78 is the ASCII value for the letter 'X'). The number '8' is the opcode for the "retw" instruction, which loads the subsequent value (78, for instance) into the W register and then returns to the calling routine. This is useful for creating lookup tables.

Fill: Pressing 'F' invokes the fill mode. In this mode, you can fill a portion of the buffer with a specific value.

To fill an area, follow these steps:

1) Place the cursor at the first location within the area to be filled.
2) Press 'F' to invoke the fill mode.
3) Move the cursor to the last location within the area to be filled.
4) Press RETURN. The area from the first location to the last will be filled with the value in the first location.

Command-Line Options

The downloader software has several useful options that can be specified from the DOS command-line. The use of these options is shown below:

PEP	Runs downloader software.
PEP *filename*	Runs software and loads *filename*.
PEP *filename* /e	Runs software, loads *filename*, and sends all file data to the downloader. These operations take place entirely from the command-line (the normal display is never seen). This is useful if you want to include the downloading process as part of a batch file.
PEP *filename* /m	Runs software in monochrome mode and loads *filename*.

Error Messages

The following list shows the errors that may occur while using the downloader software:

PIC16Cxx-EMU not found:	The software could not find the downloader on any parallel port.

FIGURE 18.0.14 - MAXIMUM IDD vs FREQ (EXT CLOCK, -55° to +125°C)

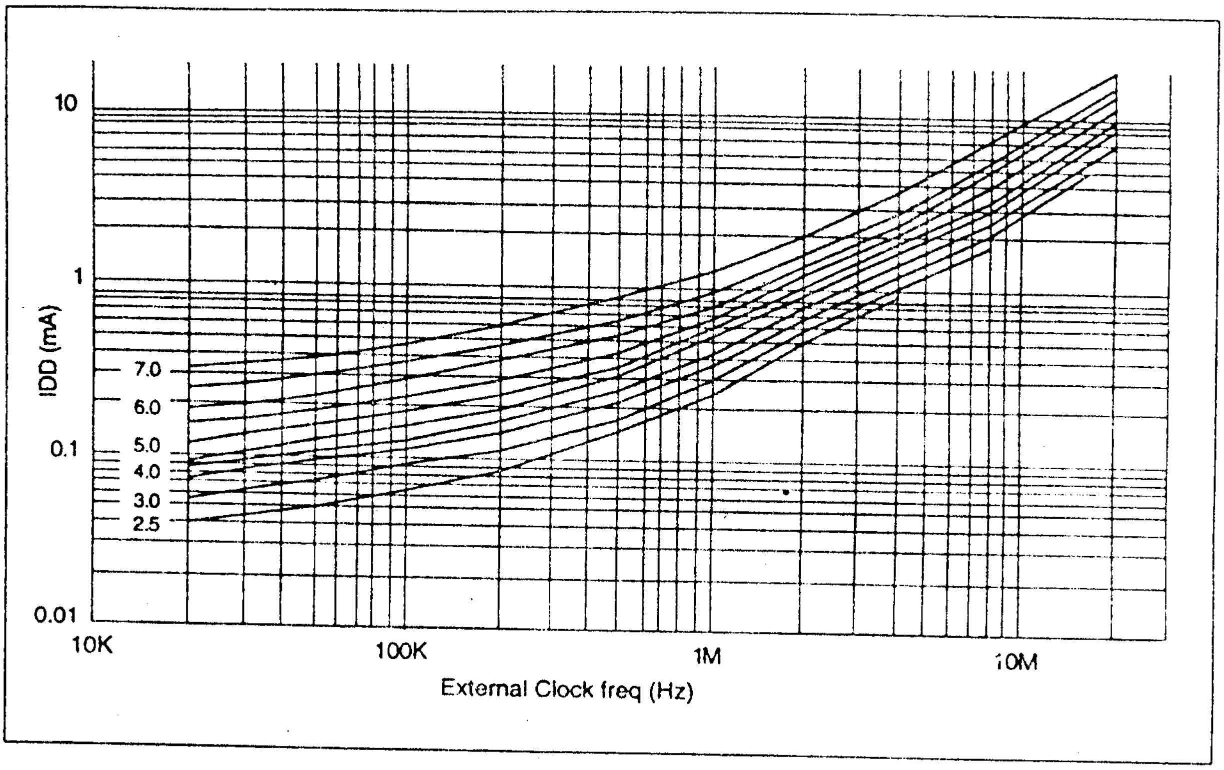


FIGURE 18.0.15 - WDT Timer Time-out Period vs VDD

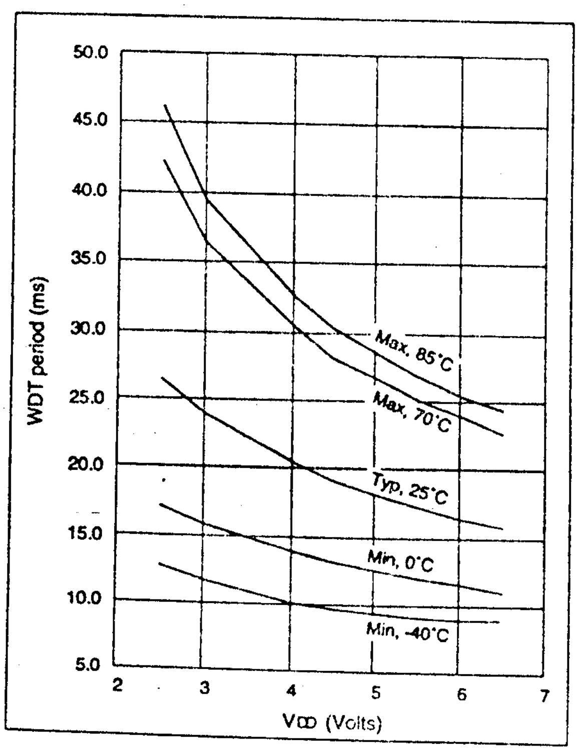
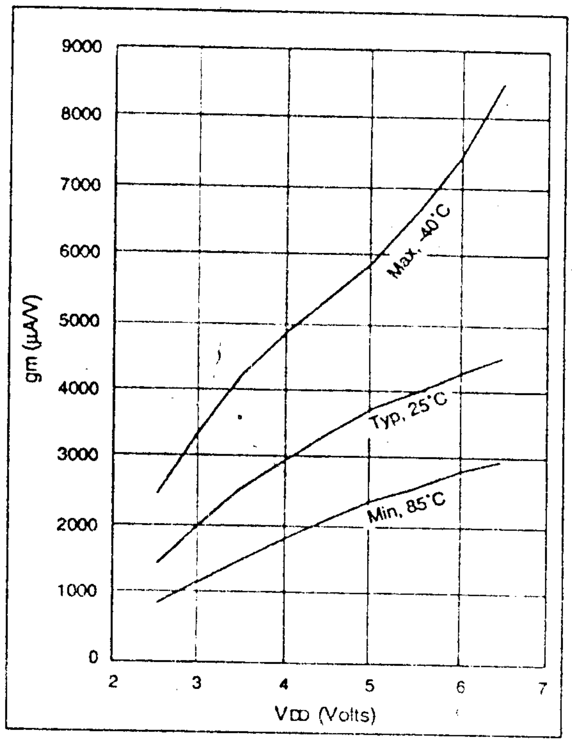


FIGURE 18.0.16 - Transconductance (gm) of HS Oscillator vs VDD



3.2 AC Program and Test Mode Characteristics

TABLE 5 - AC CHARACTERISTICS (TA = +10°C to +40°C, VDD = 5.0 V)

Characteristics	Symbol	Min	Typ	Max	Units	Conditions
$\overline{MCLR}$ Rise Time	T_R	0.15	1.0	5	µs	Note
$\overline{MCLR}$ Fall Time	T_F	0.5	2.0	5	µs	Note
Program Mode Setup Time	T_{PS}	1.0			µs	
Data Access Time	T_{ACC}			250	ns	
Data Setup Time	T_{DS}	1.0			µs	
Data Hold Time	T_{DH}	1.0			µs	
Output Enable Time	T_{OE}	0		100	ns	
Output Disable Time	T_{OZ}	0		100	ns	
Programming Pulse Width	T_{PW}	10.0	100	1000	µs	Standard and TEST EPROM
Programming Pulse Width	T_{PWF}		10000		µs	Configuration Fuses Only
Recovery Time	T_{RC}	10.0			µs	
Frequency on OSC1	F_{OSC}	DC		5	MHz	For Incrementing of the PC

3.3 Timing Diagrams

FIGURE 3.3.1 - PROGRAMMING AND VERIFY TIMING WAVEFORM

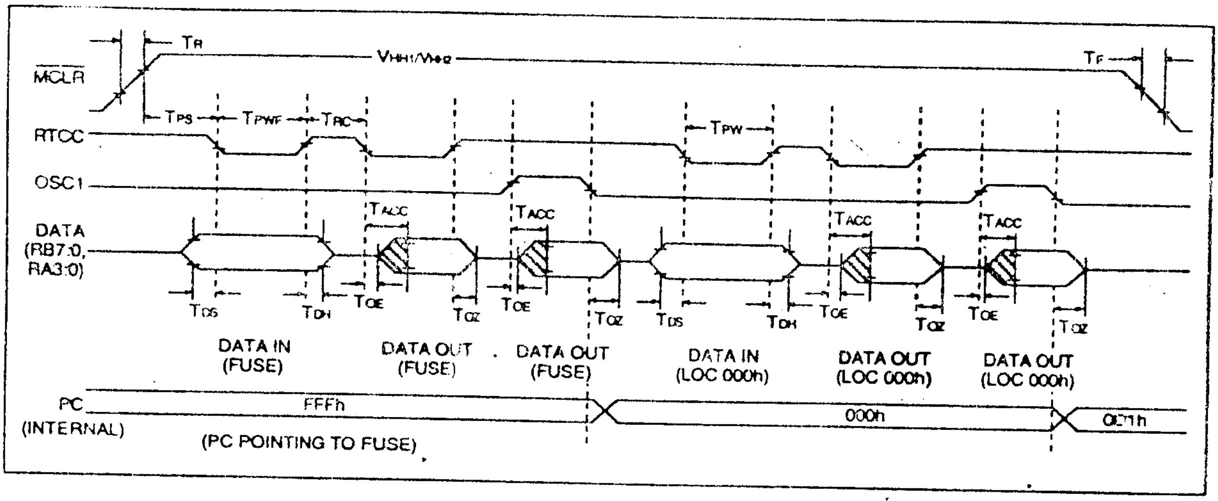
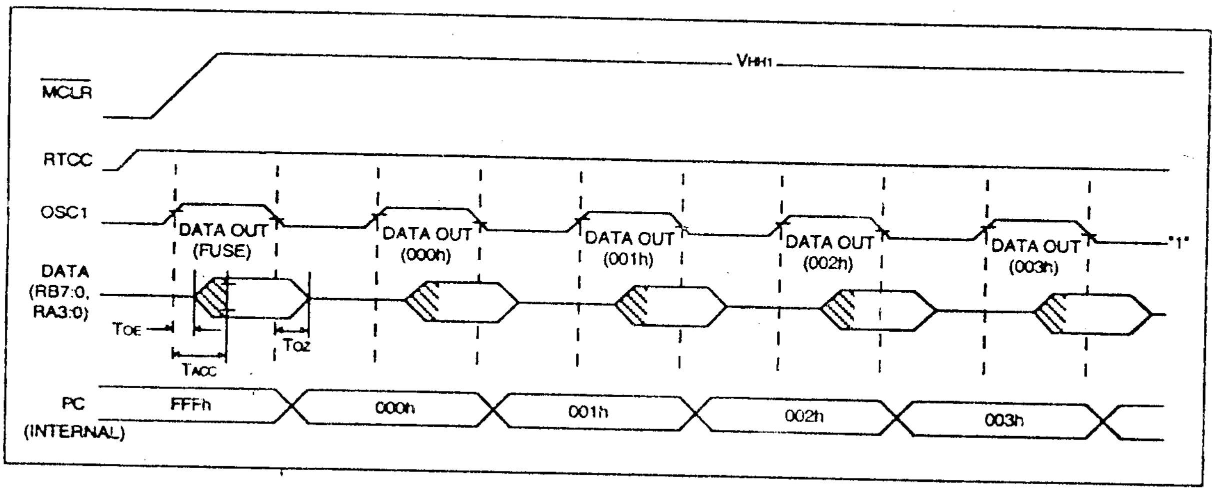


FIGURE 3.3.2 - SPEED VERIFY TIMING WAVEFORM



16.7 DC CHARACTERISTICS: PIC16C5X-RC, XT, HS, LP (Automotive)

DC CHARACTERISTICS, ALL PINS EXCEPT POWER SUPPLY			Standard Operating Conditions (unless otherwise stated) Operating temperature -40 < T_A < +125°C Operating voltage V_{DD} range as described in DC spec tables 16.3 and 16.4			
Characteristic	**Sym**	**Min**	**Typ (Note 1)**	**Max**	**Units**	**Conditions**
Input Low Voltage						
I/O ports	V_{IL}	V_{SS}		0.15 V_{DD}	V	Pin at high-impedance
$\overline{MCLR}$ (Schmitt trigger)		V_{SS}		0.15 V_{DD}	V	
RTCC (Schmitt trigger)		V_{SS}		0.15 V_{DD}	V	
OSC1 (Schmitt trigger)		V_{SS}		0.15 V_{DD}	V	PIC16C5XRC only (Note 5)
OSC1		V_{SS}		0.3 V_{DD}	V	PIC16C5X-XT, HS, LP
Input High Voltage						
I/O ports	V_{IH}	0.45 V_{DD}		V_{DD}	V	For all V_{DD} (Note 6)
		2.0		V_{DD}	V	4.0 V < V_{DD} ≤ 5.5 V (Note 6)
		0.36 V_{DD}		V_{DD}	V	V_{DD} > 5.5 V
$\overline{MCLR}$ (Schmitt trigger)		0.85 V_{DD}		V_{DD}	V	
RTCC (Schmitt trigger)		0.85 V_{DD}		V_{DD}	V	
OSC1 (Schmitt trigger)		0.85 V_{DD}		V_{DD}	V	PIC16C5X-RC only (Note 5)
OSC1		0.7 V_{DD}		V_{DD}	V	PIC16C5X-XT, HS, LP
Input Leakage Current (Notes 3, 4)						**For V_{DD} ≤ 5.5V**
I/O ports	I_{IL}	-1	0.5	+1	µA	V_{SS} ≤ V_{PIN} ≤ V_{DD}, Pin at hi-impedance
$\overline{MCLR}$		-5			µA	V_{PIN} = V_{SS} + 0.25V
$\overline{MCLR}$			0.5	+5	µA	V_{PIN} = V_{DD}
RTCC		-3	0.5	+3	µA	V_{SS} ≤ V_{PIN} ≤ V_{DD}
OSC1		-3	0.5	+3	µA	V_{SS} ≤ V_{PIN} ≤ V_{DD}, PIC16C5X-XT, HS, LP
Output Low Voltage						
I/O Ports	V_{OL}			0.6	V	I_{OL} = 8.7 mA, V_{DD} = 4.5V
OSC2/CLKOUT (PIC16C5X-RC)				0.6	V	I_{OL} = 1.6 mA, V_{DD} = 4.5V
Output High Voltage						
I/O Ports (Note 4)	V_{OH}	V_{DD}-0.7			V	I_{OH} = -5.4 mA, V_{DD} = 4.5V
OSC2/CLKOUT (PIC16C5X-RC)		V_{DD}-0.7			V	I_{OH} = -1.0 mA, V_{DD} = 4.5V

Note 1: Data in the column labeled "Typical" is based on characterization results at 25 °C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

Note 2 : Total power dissipation as stated under absolute maximum ratings must not be exceeded.

Note 3 : The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

Note 4 : Negative current is defined as coming out of the pin.

Note 5 : For PIC16C5XRC devices, the OSC1 pin is a Schmitt trigger input. It is not recommended that the PIC16C5X ven with external clock in RC mode.

Note 6: user may use better of the two specifications.

16.8 AC CHARACTERISTICS: PIC16C5X-RC, XT, HS, LP (Commercial) PIC16C5XI-RC, XT, HS, LP (Industrial) PIC16C5XI-RC, XT, HS, LP (Automotive)

AC CHARACTERISTICS	Standard Operating Conditions (unless otherwise stated) Operating temperature T_A = -40°C to +85°C (industrial), T_A = -40°C to +125°C (automotive) and 0°C ≤ T_A ≤ +70°C (commercial) Operating voltage VDD range as described in DC spec tables 16.3 and 16.4

Characteristic	Sym	Min	Typ (Note 1)	Max	Units	Conditions
External CLOCKIN Frequency (Note 2)	FOSC	DC		4	MHz	RC mode
		DC		4	MHz	XT mode
		DC		20	MHz	HS mode (Com/Ind)
		DC		16	MHz	HS mode (Automotive)
		DC		40	KHz	LP mode
Oscillator Frequency (Note 2)	FOSC	DC		4	MHz	RC mode
		0.1		4	MHz	XT mode
		4		20	MHz	HS mode (Com/Ind)
		4		16	MHz	HS mode (Automotive)
		DC		40	KHz	LP mode
Instruction Cycle Time (Note 2)	TCY	1.0	4/FOSC	DC	µs	RC mode
		1.0		DC	µs	XT mode
		0.2		DC	µs	HS mode
		100		DC	µs	LP mode
External Clock in Timing (Note 4)						
Clock in (OSC1) High or Low Time						
XT oscillator type	TCKHLXT	50*			ns	
LP oscillator type	TCKHLLP	2*			µs	
HS oscillator type	TCKHLHS	20*			ns	
Clock in (OSC1) Rise or Fall Time						
XT oscillator type	TCKRFXT	25*			ns	
LP oscillator type	TCKRFLP	50*			ns	
HS oscillator type	TCKRFHS	25*			ns	
RESET Timing						
$\overline{MCLR}$ Pulse Width (low)	TMCL	100*			ns	
RTCC Input Timing, No Prescaler						
RTCC High Pulse Width	TRTH	0.5 TCY+ 20*			ns	Note 3
RTCC Low Pulse Width	TRTL	0.5 TCY+ 20*			ns	Note 3
RTCC Input Timing, With Prescaler						
RTCC High Pulse Width	TRTH	10*			ns	Note 3
RTCC Low Pulse Width	TRTL	10*			ns	Note 3
RTCC Period	TRTP	$\frac{TCY + 40}{N}$ *			ns	Note 3. Where N = prescale value (2,4, ..., 256)
Watchdog Timer Timeout Period (No Prescaler)	TWDT	9*	18*	30*	ms	VDD = 5.0V
Oscillation Start-up Timer Period	TOST	9*	18*	30*	ms	VDD = 5.0V
I/O Timing						
I/O Pin Input Valid Before CLKOUT↑ (RC Mode)	TDS	0.25 TCY+ 30*			ns	
I/O Pin Input Hold After CLKOUT↑ (RC Mode)	TDH	0*			ns	
I/O Pin Output Valid After CLKOUT↓ (RC Mode)	TPD			40*	ns	

* Guaranteed by characterization, but not tested.

(Notes on next page)

16.5 DC CHARACTERISTICS: PIC16C5XE-RC, XT, HS, LP (Automotive)

DC CHARACTERISTICS, POWER SUPPLY PINS	Standard Operating Conditions Operating temperature $-40 \leq T_A \leq +125$°C, unless otherwise stated Operating voltage VDD = 3.5V to 5.5V unless otherwise stated					
Characteristic	**Sym**	**Min**	**Typ (Note 1)**	**Max**	**Units**	**Conditions**
Supply Voltage						
PIC16C5X-XT	VDD	3.25		6.0	V	Fosc = DC to 4 MHz
PIC16C5X-RC		3.25		6.0	V	Fosc = DC to 4 MHz
PIC16C5X-HS		4.5		5.5	V	Fosc = DC to 20 MHz
PIC16C5X-LP		2.5		6.0	V	Fosc = DC to 40 KHz
RAM Data Retention Voltage (Note 3)	VDR		1.5		V	Device in SLEEP mode
VDD start voltage to guarantee power on reset	VPOR		Vss		V	See section 13.1 for details on power on reset
VDD rise rate to guarantee power on reset	SVDD	0.05*			V/ms	See section 13.1 for details on power on reset
Supply Current (Note 2)	IDD					
PIC16C5X-XT			1.8	3.3	mA	Fosc = 4 MHz, VDD = 5.5V
PIC16C5X-RC (Note 5)			1.8	3.3	mA	Fosc = 4 MHz, VDD = 5.5V
PIC16C5X-HS			4.8	10.0	mA	Fosc = 10 MHz, VDD = 5.5V
			9.0	20.0	mA	Fosc = 16 MHz, VDD = 5.5V
PIC16C5X-LP			25	55	µA	Fosc = 32 KHz, VDD = 3.25V, WDT disabled
Power Down Current (Note 4)	IPD					
PIC16C5X			5	22	µA	VDD = 3.25V, WDT enabled
			0.8	18	µA	VDD = 3.25V, WDT disabled

* These parameters are based on characterization and are not tested.

Note 1: Data in the column labeled "Typical" is based on characterization results at 25°C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

Note 2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.
a) The test conditions for all IDD measurements in active operation mode are: OSC1= external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, RT = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.
b) For stand-by current measurements, the conditions are the same, except that the device is in SLEEP mode.

Note 3: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

Note 4: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedence state and tied to VDD and Vss.

Note 5: Does not include current through Rext. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

16.6 DC CHARACTERISTICS: PIC16C5X-RC, XT, HS, LP (Commercial) PIC16C5XI-RC, XT, HS, LP (Industrial)

DC CHARACTERISTICS, ALL PINS EXCEPT POWER SUPPLY	Standard Operating Conditions (unless otherwise stated) Operating temperature -40 < TA < +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD range as described in DC spec tables 16.3 and 16.4					
Characteristic	**Sym**	**Min**	**Typ (Note 1)**	**Max**	**Units**	**Conditions**
Input Low Voltage						
I/O ports	VIL	Vss		0.2 VDD	V	Pin at hi-impedance
$\overline{MCLR}$ (Schmitt trigger)		Vss		0.15 VDD	V	
RTCC (Schmitt trigger)		Vss		0.15 VDD	V	
OSC1 (Schmitt trigger)		Vss		0.15 VDD	V	PIC16C5XRC only (Note 5)
OSC1		Vss		0.3 VDD	V	PIC16C5X-XT, HS, LP
Input High Voltage						
I/O ports	VIH	0.45 VDD		VDD	V	For all VDD (Note 6)
		2.0		VDD	V	4.0 V < VDD ≤ 5.5 V (Note 6)
		0.36 VDD		VDD	V	VDD > 5.5 V
$\overline{MCLR}$ (Schmitt trigger)		0.85 VDD		VDD	V	
RTCC (Schmitt trigger)		0.85 VDD		VDD	V	
OSC1 (Schmitt trigger)		0.85 VDD		VDD	V	PIC16C5X-RC only (Note 5)
OSC1		0.7 VDD		VDD	V	PIC16C5X-XT, HS, LP
Input Leakage Current (Notes 3, 4)						For VDD ≤ 5.5V
I/O ports	IIL	-1	0.5	+1	µA	VSS ≤ VPIN ≤ VDD, Pin at hi-impedance
$\overline{MCLR}$		-5			µA	VPIN = VSS + 0.25V
$\overline{MCLR}$			0.5	+5	µA	VPIN = VDD
RTCC		-3	0.5	+3	µA	VSS ≤ VPIN ≤ VDD
OSC1		-3	0.5	+3	µA	VSS ≤ VPIN ≤ VDD, PIC16C5X-XT, HS, LP
Output Low Voltage						
I/O Ports	VOL			0.6	V	IOL = 8.7 mA, VDD = 4.5V
OSC2/CLKOUT (PIC16C5X-RC)				0.6	V	IOL = 1.6 mA, VDD = 4.5V
Output High Voltage						
I/O Ports (Note 4)	VOH	VDD-0.7			V	IOH = -5.4 mA, VDD = 4.5V
OSC2/CLKOUT (PIC16C5X-RC)		VDD-0.7			V	IOH = -1.0 mA, VDD = 4.5V

Note 1: Data in the column labeled "Typical" is based on characterization results at 25 °C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

Note 2 : Total power dissipation as stated under absolute maximum ratings must not be exceeded.

Note 3 : The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

Note 4 : Negative current is defined as coming out of the pin.

Note 5 : For PIC16C5XRC devices, the OSC1 pin is a Schmitt trigger input. It is not recommended that the PIC16C5X be driven with external clock in RC mode.

Note 6: The user may use better of the two specifications.

16.3 DC CHARACTERISTICS: PIC16C5X-RC, XT, HS, LP (Commercial)

DC CHARACTERISTICS, POWER SUPPLY PINS	Standard Operating Conditions Operating temperature $0 \le T_A \le +70°C$, unless otherwise stated Operating voltage VDD = 3.0V to 5.5V unless otherwise stated					
Characteristic	**Sym**	**Min**	**Typ (Note 1)**	**Max**	**Units**	**Conditions**
Supply Voltage						
PIC16C5X-XT	VDD	3.0		6.25	V	FOSC = DC to 4 MHz
PIC16C5X-RC		3.0		6.25	V	FOSC = DC to 4 MHz
PIC16C5X-HS		4.5		5.5	V	FOSC = DC to 20 MHz
PIC16C5X-LP		2.5		6.25	V	FOSC = DC to 40 KHz
RAM Data Retention Voltage (Note 3)	VDR		1.5		V	Device in SLEEP mode
VDD start voltage to guarantee power on reset	VPOR		Vss		V	See Section 13.1 for details on power on reset
VDD rise rate to guarantee power on reset	SVDD	0.05*			V/ms	See Section 13.1 for details on power on
Supply Current (Note 2)						
PIC16C5X-XT	IDD		1.8	3.3	mA	FOSC = 4 MHz, VDD = 5.5V
PIC16C5X-RC (Note 5)			1.8	3.3	mA	FOSC = 4 MHz , VDD = 5.5V
PIC16C5X-HS			4.8	10	mA	FOSC =10 MHz, VDD = 5.5V
			9.0	20	mA	FOSC = 20 MHz, VDD = 5.5V
PIC16C5X-LP			15	32	µA	FOSC = 32 KHz, VDD=3.0V, WDT disabled
Power Down Current (Note 4)						
PIC16C5X	IPD		4	12	µA	VDD = 3.0V, WDT enabled
			0.6	9	µA	VDD = 3.0V, WDT disabled

* These parameters are based on characterization and are not tested.

Note 1: Data in the column labeled "Typical" is based on characterization results at 25°C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

Note 2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, RT = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

b) For stand-by current measurements, the conditions are the same, except that the device is in SLEEP mode.

Note 3: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

Note 4: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedence state and tied to VDD and Vss.

Note 5: Does not include current through Rext. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

16.4 DC CHARACTERISTICS: PIC16C5XI-RC, XT, HS, LP (Industrial)

DC CHARACTERISTICS, POWER SUPPLY PINS	Standard Operating Conditions Operating temperature -40 ≤ TA ≤ +85°C, unless otherwise stated Operating voltage VDD = 3.5V to 5.5V unless otherwise stated					
Characteristic	**Sym**	**Min**	**Typ (Note 1)**	**Max**	**Units**	**Conditions**
Supply Voltage						
PIC16C5X-XT	VDD	3.0		6.25	V	Fosc = DC to 4 MHz
PIC16C5X-RC		3.0		6.25	V	Fosc = DC to 4 MHz
PIC16C5X-HS		4.5		5.5	V	Fosc = DC to 20 MHz
PIC16C5X-LP		2.5		6.25	V	Fosc = DC to 40 KHz
RAM Data Retention Voltage (Note 3)	VDR		1.5		V	Device in SLEEP mode
VDD start voltage to guarantee power on reset	VPOR		Vss		V	See section 13.1 for details on power on reset
VDD rise rate to guarantee power on reset	SVDD	0.05*			V/ms	See section 13.1 for details on power on reset
Supply Current (Note 2)						
PIC16C5X-XT	IDD		1.8	3.3	mA	Fosc = 4 MHz, VDD = 5.5V
PIC16C5X-RC (Note 5)			1.8	3.3	mA	Fosc = 4 MHz, VDD = 5.5V
PIC16C5X-HS			4.8	10.0	mA	Fosc = 10 MHz, VDD = 5.5V
			9.0	20.0	mA	Fosc = 20 MHz, VDD = 5.5V
PIC16C5X-LP			19	40	µA	Fosc = 32 KHz, VDD = 3.0V, WDT disabled
Power Down Current (Note 4)						
PIC16C5X	IPD		5	14	µA	VDD = 3.0V, WDT enabled
			0.8	12	µA	VDD = 3.0V, WDT disabled

* These parameters are based on characterization and are not tested.

Note 1: Data in the column labeled "Typical" is based on characterization results at 25°C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

Note 2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.
a) The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, RT = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.
b) For stand-by current measurements, the conditions are the same, except that the device is in SLEEP mode.

Note 3: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

Note 4: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedence state and tied to VDD and VSS.

Note 5: Does not include current through Rext. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

FIGURE 18.0.21 - IOL vs VOL, VDD = 3V

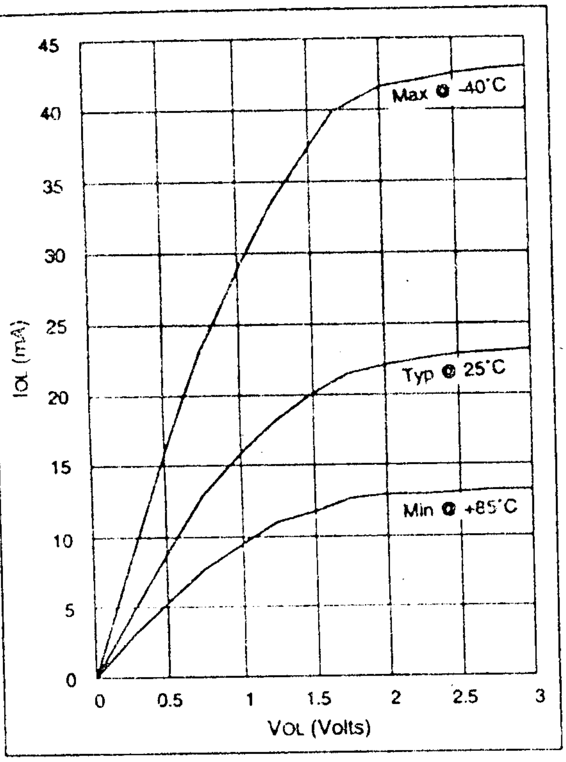


FIGURE 18.0.22 - IOL vs VOL, VDD = 5V

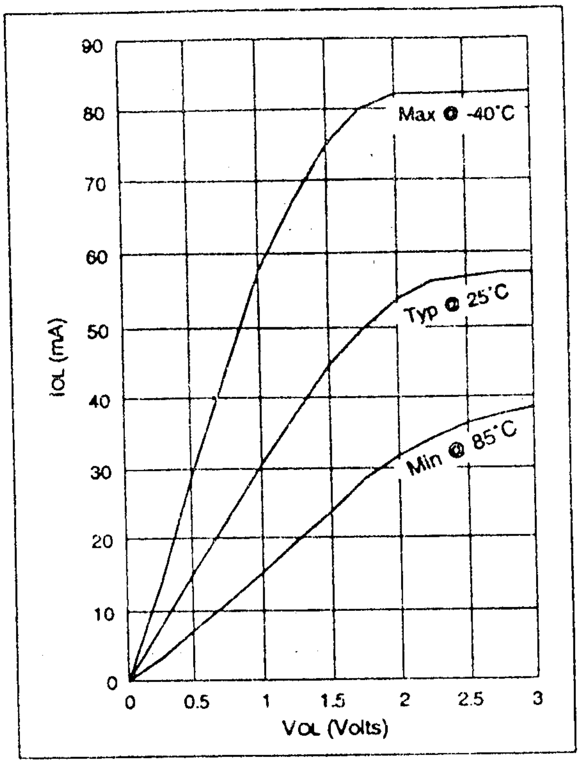


TABLE 18.0.2 - INPUT CAPACITANCE FOR PIC16C54/56 *

Pin Name	Typical Capacitance (pF)	
	18L PDIP	18L SOIC
RA port	5.0	4.3
RB port	5.0	4.3
MCLR	17.0	17.0
OSC1	4.0	3.5
OSC2/CLKOUT	4.3	3.5
RTCC	3.2	2.8

* All capacitance values are typical at 25°C and measured at 1 MHz. A part to part variation of ±25% (three standard deviations) should be taken into account.

TABLE 18.0.3 - INPUT CAPACITANCE FOR PIC16C55/57 *

Pin Name	Typical Capacitance (pF)	
	28L PDIP (600 mil)	28L SOIC
RA port	5.2	4.8
RB port	5.6	4.7
RC port	5.0	4.1
MCLR	17.0	17.0
OSC1	6.6	3.5
OSC2/CLKOUT	4.6	3.5
RTCC	4.5	3.5

FIGURE 18.0.17 - Transconductance (gm) of LP Oscillator vs VDD

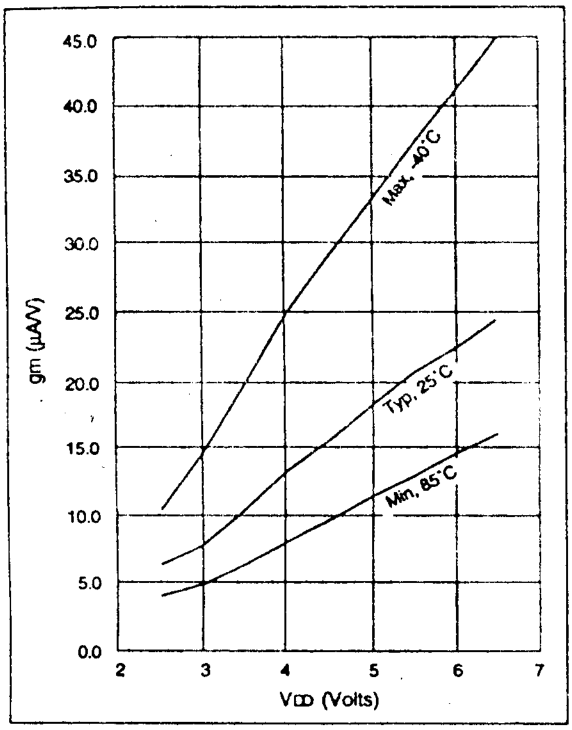


FIGURE 18.0.18 - Transconductance (gm) of XT Oscillator vs VDD

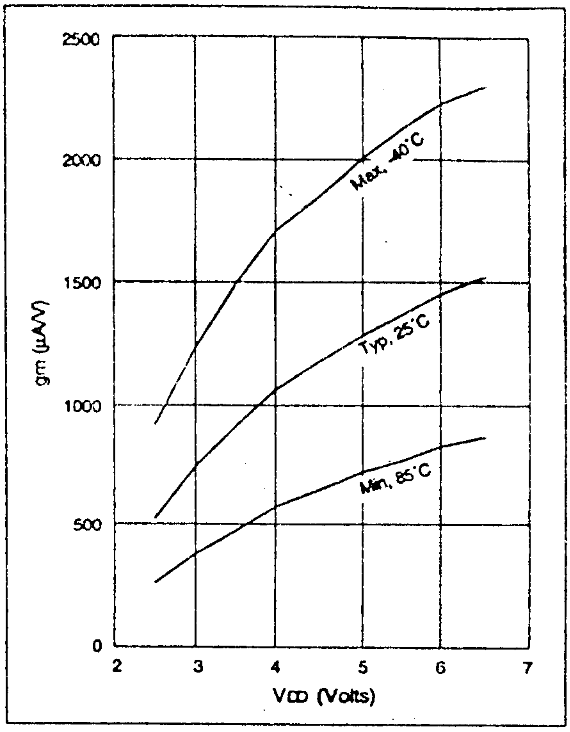


FIGURE 18.0.19 - IOH vs VOH, VDD = 3V

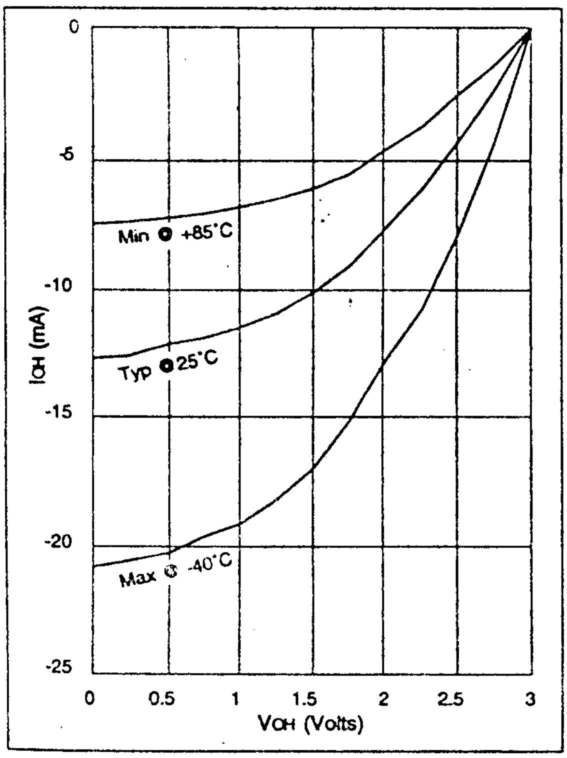


FIGURE 18.0.20 - IOH vs VOH, VDD = 5V

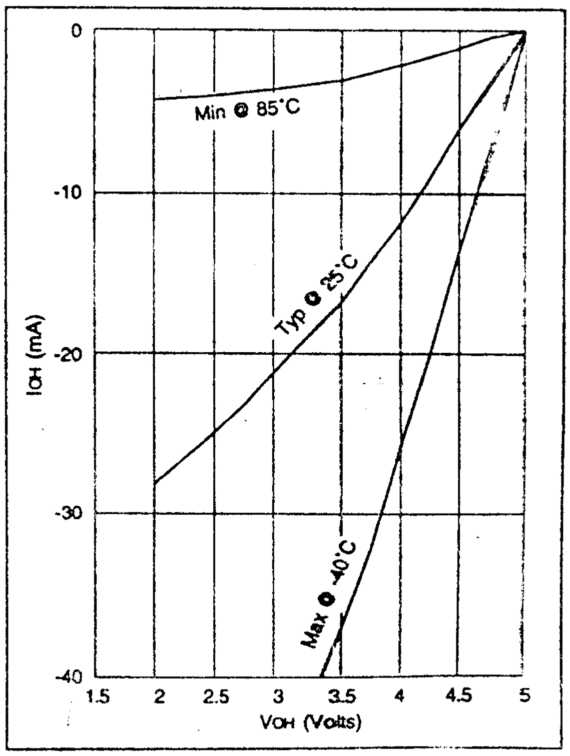
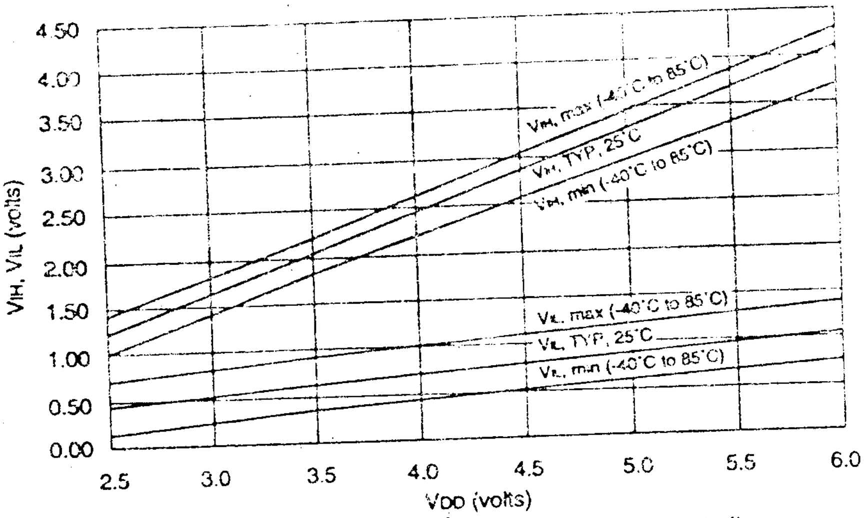


FIGURE 18.0.10 - VIH, VIL OF MCLR, RTCC AND OSC1 (IN RC MODE) vs VDD



Note: These input pins have Schmitt trigger input buffer.

FIGURE 18.0.11 - VTH (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN XT, HS, AND LP MODES) vs VDD

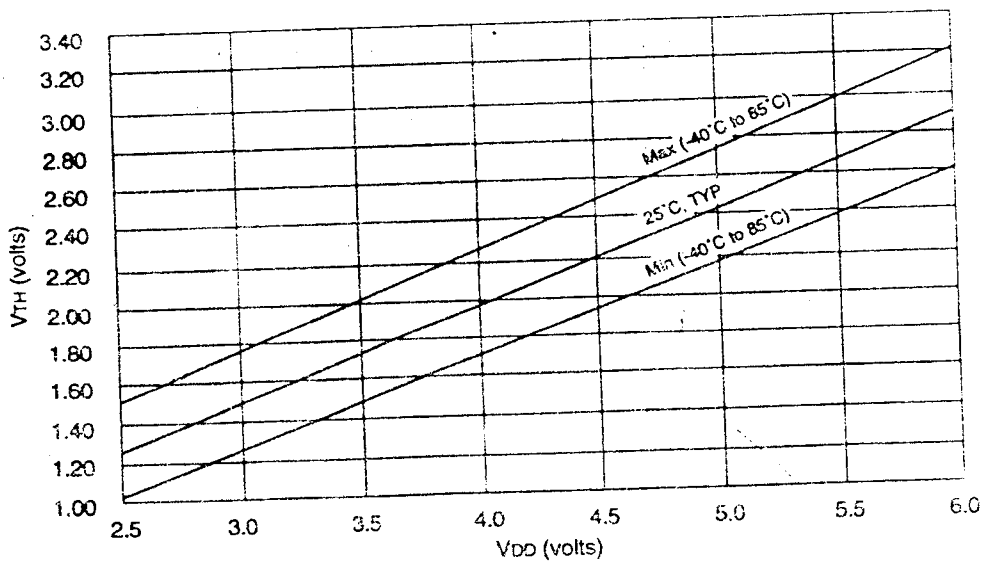


FIGURE 18.0.12 - TYPICAL IDD vs FREQ (EXT CLOCK, 25°C)

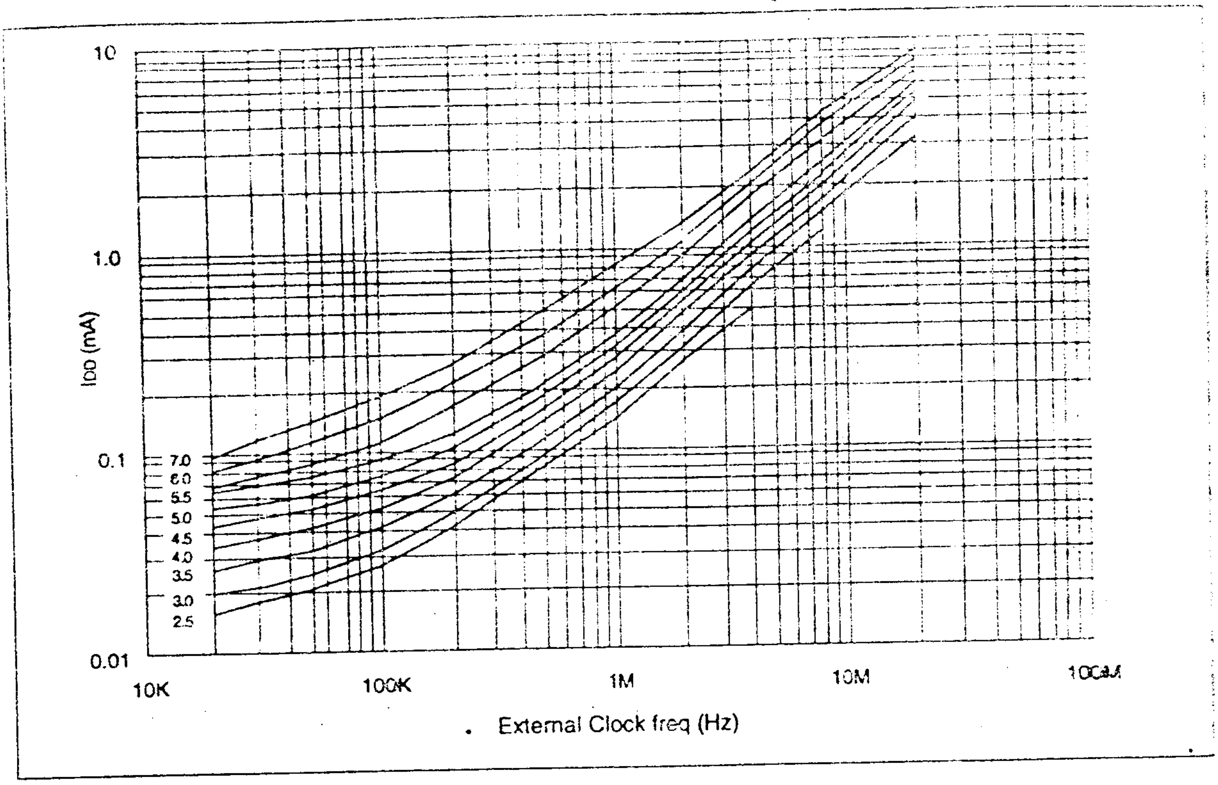


FIGURE 18.0.13 - MAXIMUM IDD vs FREQ (EXT CLOCK, -40° to +85°C)

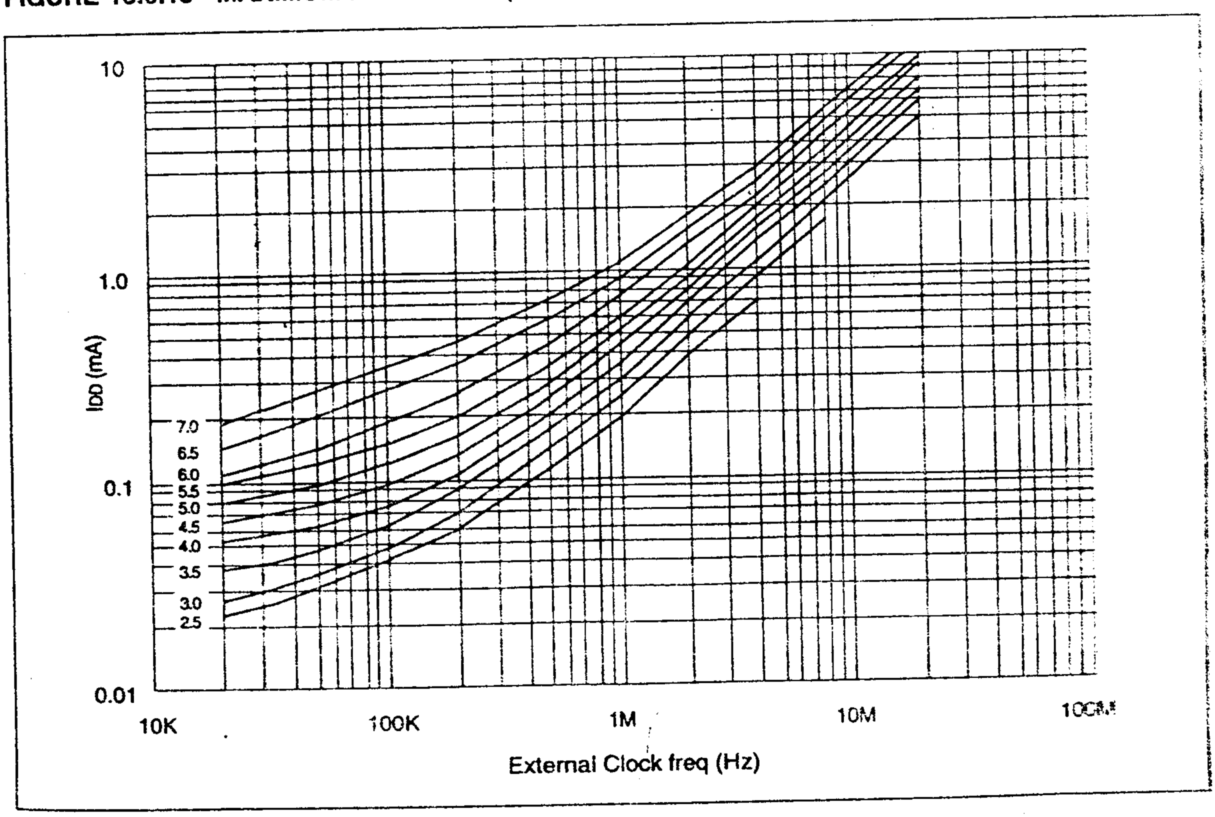
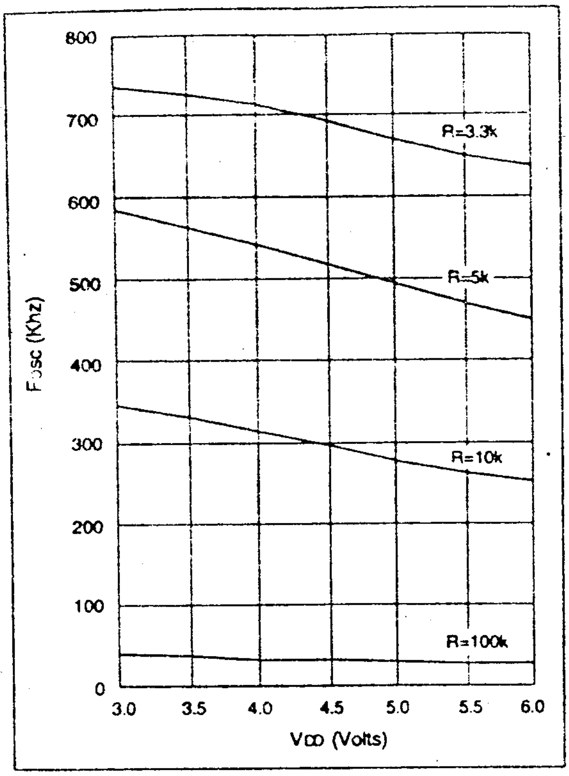


FIGURE 18.0.4 - TYPICAL RC OSCILLATOR FREQUENCY vs VDD*



* Measured on DIP packages.

TABLE 18.0.1 - RC OSCILLATOR FREQUENCIES*

Cext	Rext	Average Fosc @ 5V, 25°C	
20pF	3.3k	4.973 MHz	± 27%
	5k	3.82 MHz	± 21%
	10k	2.22 MHz	± 21%
	100k	262.15 KHz	± 31%
100pF	3.3k	1.63 MHz	± 13%
	5k	1.19 MHz	± 13%
	10k	648.64 KHz	± 18%
	100k	71.56 KHz	± 25%
300pF	3.3k	660.0 KHz	± 10%
	5k	484.1 KHz	± 14%
	10k	267.63 KHz	± 15%
	100k	29.44 KHz	± 19%

* Measured on DIP packages.

The percentage variation indicated here is part to part variation due to normal process distribution. The variation indicated is ±3 standard deviation from average value for VDD = 5V.

FIGURE 18.0.5 - TYPICAL Ipd vs VDD WATCHDOG DISABLED 25°C

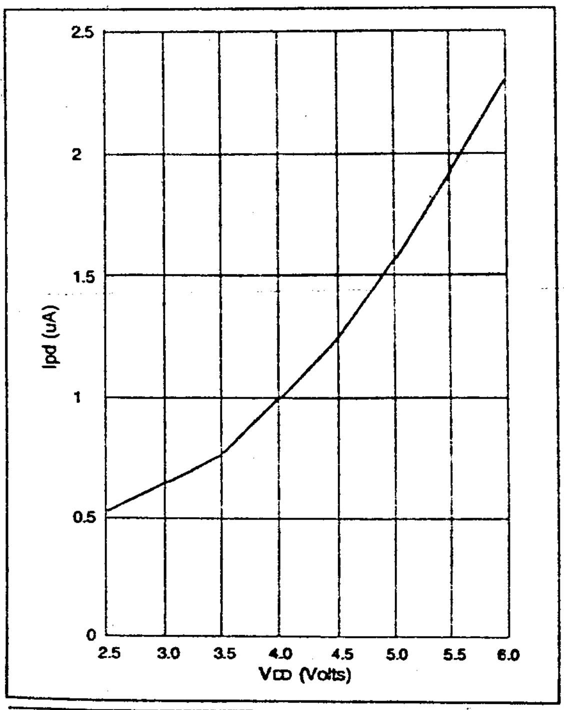


FIGURE 18.0.6 - TYPICAL Ipd vs VDD WATCHDOG ENABLED 25°C

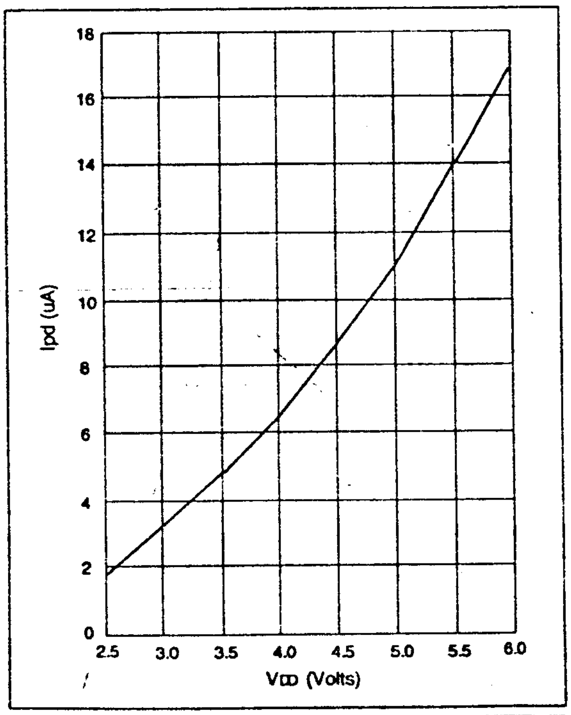


FIGURE 18.0.7 - MAXIMUM Ipd vs VDD WATCHDOG DISABLED

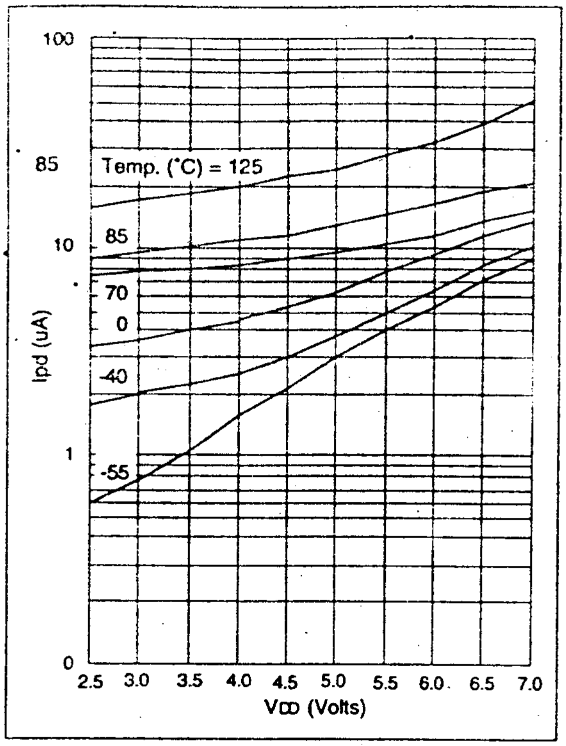
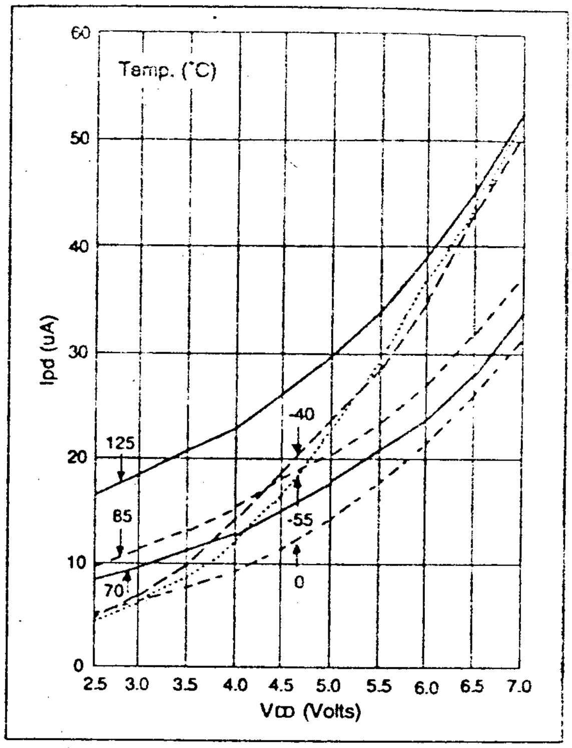
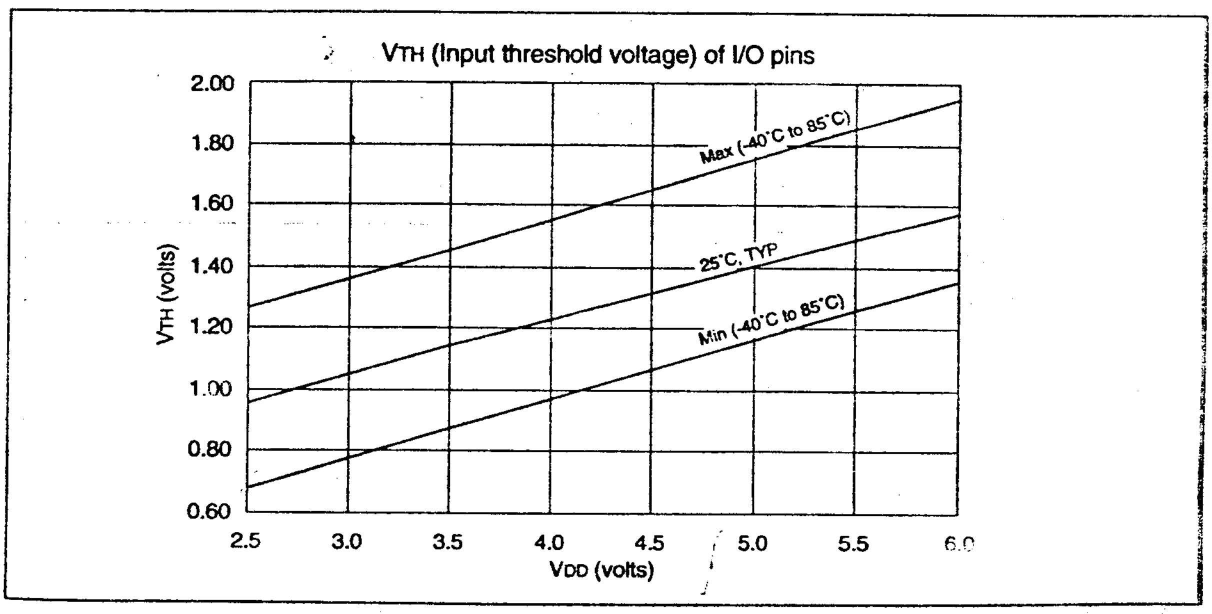


FIGURE 18.0.8 - MAXIMUM Ipd vs VDD WATCHDOG ENABLED*

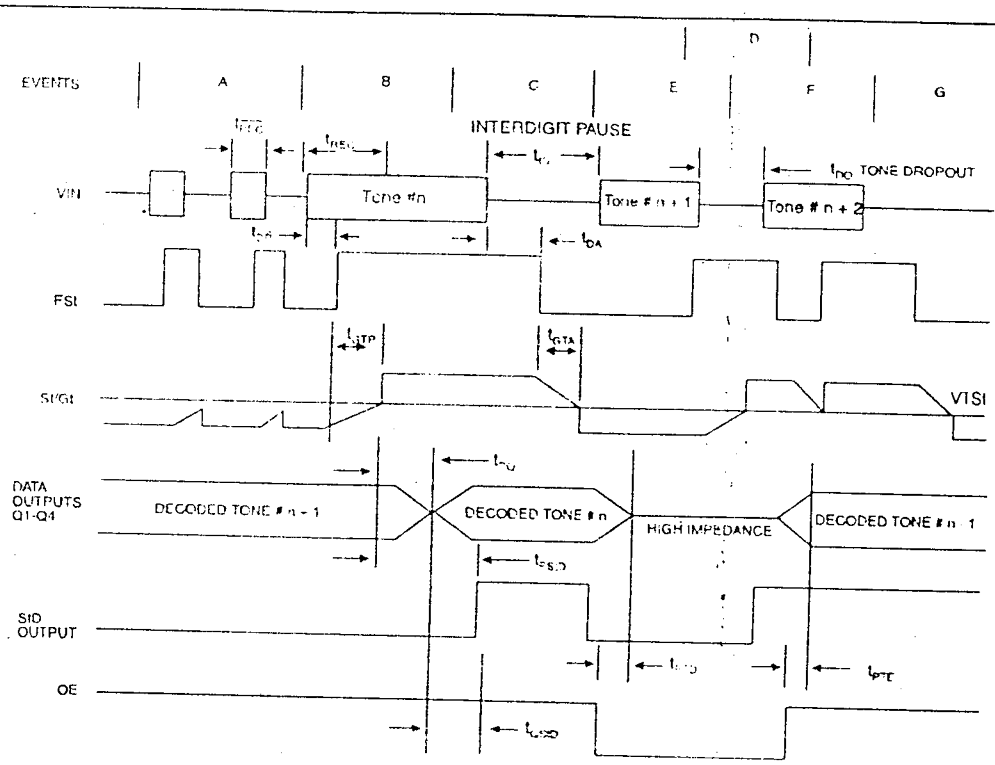


* IPD, with watchdog timer enabled, has two components: The leakage current which increases with higher temperature and the operating current of the watchdog timer logic which increases with lower temperature. At -40°C, the latter dominates explaining the apparently anomalous behavior.

FIGURE 18.0.9 - VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS vs VDD



Appendix C



Explanation of Events

(A) Tone bursts detected, tone duration invalid, outputs not updated.
(B) Tone #n detected, tone duration valid, tone decoded and latched in outputs.
(C) End of tone #n detected, tone absent duration valid, outputs remain latched until next valid tone.
(D) Outputs switched to high impedance state.
(E) Tone #n + 1 detected, tone duration valid, tone decoded and latched in outputs (currently high impedance).
(F) Acceptable dropout of tone #n + 1, tone absent duration invalid, outputs remain latched.
(G) End of tone #n + 1 detected, tone absent duration valid, outputs remain latched until next valid tone.

Explanation of Symbols

VIN	DTMF composite input signal.
ESt	Early steering output. Indicates detection of valid tone frequencies.
St/GT	Steering input/guard time output. Drives external RC timing circuit.
Q1 - Q4	4-bit decoded tone output.
StD	Delayed steering output. Indicates that valid frequencies have been present/absent for the required guardtime, thus constituting a valid signal.
$\overline{OE}$	Output enable (input). A low level shifts Q1 - Q4 to its high impedance state.
t_{REC}	Maximum DTMF signal duration not detected as valid.
t_{REC}	Minimum DTMF signal duration required for valid recognition.
t_{ID}	Minimum time between valid DTMF signals.
t_{DO}	Maximum allowable dropout during valid DTMF signal.
t_{DP}	Time to detect the presence of valid DTMF signals.
t_{DA}	Time to detect the absence of valid DTMF signals.
t_{GTP}	Guard time, tone present.
t_{GTA}	Guard time, tone absent.

Figure 7 Timing Diagram

Table 1 Pin Functions

PIN	NAME	DESCRIPTION	
1	IN+	Non-inverting input	Connections to the front-end differential amplifier
2	IN-	Inverting input	
3	GS	Gain select. Gives access to output of front-end amplifier for connection of feedback resistor.	
4	V_{REF}	Reference voltage output (nominally $V_{DD}/2$). May be used to bias the inputs at mid-rail.	
5	INH*	Inhibits detection of tones representing keys A, B, C, and D.	
6	PD*	Power down. Logic high powers down the device and inhibits the oscillator. Internal pulldown.	
7	OSC1	Clock input	3.579545 MHz crystal connected between these pins completes the internal oscillator.
8	OSC2	Clock output	
9	V_{SS}	Negative power supply (normally connected to 0 V).	
10	OE	Three-state output enable (input). Logic high enables the outputs O1 - Q4. Internal pullup.	
11 - 14	Q1, Q2, Q3, Q4	Three-state data outputs. When enabled by OE, provides the code corresponding to the last valid tone pair received (see Table 5)	
15	StD	Delayed steering output. Presents a logic high when a received tone pair has been registered and the output latch is updated. Returns to logic low when the voltage on St/GT falls below V_{TSt}.	
16	ESt	Early steering output. Presents a logic high immediately when the digital algorithm detects a recognizable tone pair (signal condition). Any momentary loss of signal condition will cause ESt to return to a logic low.	
17	St/GT	Steering input/guard time output (bidirectional). A voltage greater than V_{TSt} detected at St causes the device to register the detected tone pair and update the output latch. A voltage less than V_{TSt} frees the device to accept a new tone pair. The GT output acts to reset the external steering time constant, and its state is a function of ESt and the voltage on St. (See Figure 7).	
18	V_{DD}	Positive power supply. (Normally connected to +5V.)	

* -02 and -03 only. Connect to V_{SS} for -01 version

Table 5 Tone Decoding

F_{LOW}	F_{HIGH}	KEY (ref.)	OE	Q4	Q3	Q2	Q1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	*	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
ANY	ANY	ANY	L	Z	Z	Z	Z

L = logic low, H = logic high, Z = high impedance

Table 2 Absolute Maximum Ratings

PARAMETER	SYMBOL	VALUE
Power supply voltage (V_{DD} - V_{SS})	V_{DD}	6.0 V max
Voltage on any pin	V_{dc}	V_{SS} -0.3, V_{DD} +0.3
Current on any pin	I_{DD}	10 mA max
Operating temperature	T_A	-40° C to + 85 °C
Storage temperature	T_S	-65° C to + 150° C

Table 3 DC Characteristics

PARAMETER	SYMBOL	MIN	‡TYP	MAX	UNITS	TEST CONDITIONS
Operating supply voltage	V_{DD}	4.75		5.25	V	
Operating supply current	I_{DD}		3.0	7.0	mA	
Standby supply current (see Note 3)	I_{DDQ}			100	µA	PD=V_{DD}
Power consumption	P_O		15	35	mW	f = 3.579 MHz, V_{DD} = 5.0 V
Low level input voltage	V_{IL}			1.5	V	
High level input voltage	V_{IH}	3 5			V	
Input leakage current	I_{IH}/I_{IL}		0.1		µA	V_{IN} = V_{SS} or V_{DD} (see Note 2)
Pull up (source) current on OE	I_{SO}		6.5	15.0	µA	OE = 0 V
Input impedance, signal inputs 1, 2	R_{IN}	8	10		M Ω	@ 1 kHz
Steering threshold voltage	V_{TSt}	2.2		2.5	V	
Low level output voltage	V_{OL}			0.03	V	No load
High level output voltage	V_{OH}	4.97			V	No load
Output low (sink) current	I_{OL}	1.0	2.5		mA	V_{OUT} = 0.4 V
Output high (source) current	I_{OH}	0.4	0.8		mA	V_{OUT} = 4.6 V
Output voltage V_{REF}	V_{REF}	2.4		2.7	V	No load
Output resistance V_{REF}	R_{OR}		10		kΩ	

‡ Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

Notes: 1. All voltages referenced to Vss unless otherwise noted. For typical values, V_{DD} = 5.0V, V_{SS} = 0V, TA = 25°C.

2. Input pins defined as IN+, IN-, or OE.

3. -02 and -03, only.

Table 4 Operating Characteristics - Gain Setting Amplifier

PARAMETER	SYMBOL	MIN	‡ TYP	MAX	UNITS	TEXT CONDITIONS
Input leakage current	I_{IN}		±100		nA	V_{SS} < V_{IN} < V_{DD}
Input resistance	R_{IN}	4			MΩ	
Input offset voltage	V_{OS}		±25		mV	
Power supply rejection	PSRR	50			dB	1 kHz
Common mode rejection	CMRR	55			dB	-3.0V < V_{IN} < 3.0V
DC open loop voltage gain	A_{VOL}	60			dB	
Open loop unity gain bandwidth	f_C	1.2	1.5		MHz	
Output voltage swing	V_O	3.5			V_{P-P}	RL ≥ 100 kΩ to V_{SS}
Tolerable capacitive load (GS)	C_L			100	pF	
Tolerable resistive load (GS)	R_L			50	kΩ	
Common mode range	V_{CM}	2.5			V_{P-P}	No load

All voltages referenced to Vss unless otherwise noted. For typical values V_{DD} = 5.0 V, V_{SS} = 0 V, TA = 25° C.

‡ Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

Table 6 AC Specifications

PARAMETER	SYMBOL	MIN	‡TYP	MAX	UNITS	NOTES
Valid input signal levels (each tone of composite signal) (-01 and -02 only)		-29		+1	dBm	1,2,3,4,5,8
		27.5		869	mVRMS	
Valid input signal levels (each tone of composite signal) (-03 only)		-40		+1	dBm	1,2,3,4,5,8
		7.75		869	mVRMS	
Positive twist accept				10	dB	2,3,4,8
Negative twist accept				10	dB	
Frequency deviation accept limit				± 1.5% + 2 Hz	Nom.	2,3,5,8,10
Frequency deviation reject limit		±3.5%			Nom.	2,3,5
Third tone tolerance		-25	-16		dB	2,3,4,5,8,9,13,14
Noise tolerance			-12		dB	2,3,4,5,6,8,9
Dial tone tolerance		+18	+22		dB	2,3,4,5,7,8,9
Tone present detection time	t_{DP}	5	8	14	ms	See Figure 7
Tone absent detection time	t_{DA}	0.5	3	8.5	ms	
Minimum tone duration accept	t_{REC}			40	ms	User adjustable (see Figures 2 and 4)
Maximum tone duration reject	$t_{\overline{REC}}$	20			ms	
Minimum interdigit pause accept	t_{ID}			40	ms	
Maximum interdigit pause reject	t_{DO}	20			ms	
Propagation delay (St to Q)	t_{PQ}		6	11	µs	OE = V_{DD}
Propagation delay (St to StD)	t_{PStD}		9	16	µs	
Output data setup (Q to StD)	t_{QStD}		4.0		µs	
Propagation delay (OE to Q), enable	t_{PTE}		50	60	ns	$R_L = 10k\Omega$, $C_L = 50$ pF
Propagation delay (OE to Q), disable	t_{PTD}		300		ns	
Crystal clock frequency	f_{CLK}	3.5759	3.5795	3.5831	MHz	
Clock output (OSC2), capacitive load	C_{LO}			30	pF	

All voltages referenced to V_{SS} unless otherwise noted. For typical values V_{DD} = 5.0 V, V_{SS} = 0 V, T_A = 25 °C, fCLK = 3.579545 MHz.
‡Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

Notes:
1. dBm = decibels above or below a reference power of 1 mW into a 600 ohm load.
2. Digit sequence consists of all 16 DTMF tones.
3. Tone duration = 40 ms. Tone pause = 40 ms.
4. Nominal DTMF frequencies are used, measured at GS.
5. Both tones in the composite signal have an equal amplitude.
6. Bandwidth limited (0 to 3 kHz) Gaussian noise.
7. The precise dial tone frequencies are (350 and 440 Hz) ± 2%.
8. For an error rate of better than 1 in 10,000.
9. Referenced to lowest level frequency component in DTMF signal.
10. Minimum signal acceptance level is measured with specified maximum frequency deviation.
11. Input pins defined as IN+, IN-, and OE.
12. External voltage source used to bias V_{REF}.
13. This parameter also applies to a third tone injected onto the power supply.
14. Referenced to Figure 3. Input DTMF tone level at -28 dBm.

MOTOROLA
SEMICONDUCTOR TECHNICAL DATA

GlobalOptoisolator™

VDE	UL	CSA	SETI	SEMKO	DEMKO	NEMKO	BABT

6-Pin DIP Optoisolators Transistor Output

The MCT and MCT2E devices consist of a gallium arsenide infrared emitting diode optically coupled to a monolithic silicon phototransistor detector.

Applications

- General Purpose Switching Circuits
- Interfacing and coupling systems of different potentials and impedances
- I/O Interfacing
- Solid State Relays
- Monitor and Detection Circuits
- ***To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a test option.***

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
INPUT LED			
Reverse Voltage	V_R	3	Volts
Forward Current — Continuous	I_F	60	mA
LED Power Dissipation @ T_A = 25°C with Negligible Power in Output Detector Derate above 25°C	P_D	120 1.41	mW mW/°C
OUTPUT TRANSISTOR			
Collector–Emitter Voltage	V_{CEO}	30	Volts
Emitter–Collector Voltage	V_{ECO}	7	Volts
Collector–Base Voltage	V_{CBO}	70	Volts
Collector Current — Continuous	I_C	150	mA
Detector Power Dissipation @ T_A = 25°C with Negligible Power in Input LED Derate above 25°C	P_D	150 1.76	mW mW/°C
TOTAL DEVICE			
Isolation Surge Voltage(1) (Peak ac Voltage, 60 Hz, 1 sec Duration)	V_{ISO}	7500	Vac(pk)
Total Device Power Dissipation @ T_A = 25°C Derate above 25°C	P_D	250 2.94	mW mW/°C
Ambient Operating Temperature Range(2)	T_A	−55 to +100	°C
Storage Temperature Range(2)	T_{stg}	−55 to +150	°C
Soldering Temperature (10 sec, 1/16″ from case)	T_L	260	°C

1. Isolation surge voltage is an internal device dielectric breakdown rating. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.
2. Refer to Quality and Reliability Section in Opto Data Book for information on test conditions.