Intelligent Telephone

PROJECT REPORT

ISO 9001:2000
Certified

Submitted In Partial Fulfillment Of The Requirements For The Award Of The Degree Of Bachelor Of Engineering In Information Technology Of Bharathiar University, Coimbatore

Submitted By

Nandhini K. *(2KIT29)*

Sindhuja K.N. *(2KIT46)*

Under the guidance of

Mr. Mohammed Azad A.K.,
Executive Director,
Reach Information
& Communication Systems,
Coimbatore

Mr. M. Nageswara Guptha
Lecturer
Kumaraguru College Of Technology
Coimbatore

Department of Information Technology
KUMARAGURU COLLEGE OF TECHNOLOGY
Coimbatore – 641006
March 2004

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE - 641 006

CERTIFICATE

This is to certify that the project entitled

"Intelligent Telephone"

is the bonafide work of

Nandhini K.	*(0027S0089)*
Sindhuja K.N.	*(0027S0106)*

and submitted in partial fulfillment of the requirement for the award of Bachelor of Engineering Degree in Information Technology of Bharathiar university, Coimbatore

24/3/04

Head of the department
(Dr. S. Thangasamy)

Guide
(Mr. M. Nageswara Guptha)

Submitted for University Examination held on 27-03-04

27/3

Internal Examiner

27/3/04

External Examiner

BHARAT SANCHAR NIGAM LIMITED

Connecting India

Office of The Principal General Manager
Telecommunications
Telecom Building, Bharathi Park Road 2, Coimbatore – 641043.

Date : 25.03.2004.

TO WHOM SOEVER IT MAY CONCERN

This is to certify that Ms **Nandhini.K (0027S0089)** and Ms **Sindhuja.K.N (0027S0106)**, students of Kumaraguru College of Technology, Coimbatore underwent a project work in our organisation titled "**Intelligent Telephone**" from October 2003 to March 2004.

25/3/04

(SEKHAR N.S.C.)

सहायक महा प्रबंधक (विपणन)
ASSISTANT GENERAL MANAGER (MKTG)
भा.स.नि.लि. / BSNL
कोयम्बत्तूर / Coimbatore-641 043

Declaration

DECLARATION

We,

Nandhini K. (0027S0089),
Sindhuja K. N. (0027S0106)

declare that, this project titled "Intelligent Telephone" was done by us and to the best of our knowledge, so far, a similar work has not been submitted to the Bharathiar University, Coimbatore, or any other institution as an requirement for the fulfillment of any course of study.

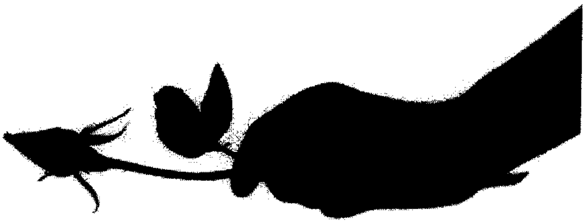
This report is submitted by us in partial fulfillment of the requirement for the award of Bachelor of Engineering Degree in Information Technology of Bharathiar University, Coimbatore.

Place: Coimbatore
Date:

Nandhini | K. N. Sindhuja

Nandhini K. | **Sindhuja K. N.**

Dedicated To Our Beloved Parents



Acknowledgements

ACKNOWLEDGEMENTS

This project report is an outcome of an enjoyed project, coordinated with the cooperation of various persons, views and ideas shared, warmth and affection showed by family and friends.

The credit goes to the faculty members, students and staff of the college especially our Principal, **Dr. K.K. Padmanaban, PhD** who directed us with sincerity and dedication.

We express our sincere thanks to our internal project guide Lecturer **Mr. M. Nageswara Guptha, B.E**, for his thought provoking suggestions and encouragement.

We would like to express our sincere thanks to **Mr. Gurusamy G. G.M (TT)** and to **Mr. Sekhar N.S.C., A.G.M. (Marketing), BSNL**, Coimbatore for giving us the opportunity to carry out this project work and their moral support that helped us in the successful completion of the project.

We are greatly indebted to our guide and mentor, **Mr. Mohammed Azad A.K.**, Managing Director, Reach Information & Communication Systems, Coimbatore and **Mr. Neelakandan A.**, Product Engineer, Reach Information & Communication Systems, Coimbatore for their relentless support, valuable suggestions they gave throughout the project.

We cherish the love and moral support of our **parents**, which has helped in the successful completion of this project.

We thank our **friends** who deserve special mention for their magnanimous care and concern.

Above all, we owe our gratitude to **God Almighty** for showering abundant blessings on us.

Contents

TABLE OF CONTENTS

Synopsis

SYNOPSIS

The mobile phone industry has stolen much of the thunder in past few years, with spectacular growth rates. In certain pockets, the number of mobile connections has exceeded the number of fixed telephone connections.

Hence, we extend most of the cell phone services to basic fixed-line telephones and provide few additional services too. This is a project done for BSNL and it would help BSNL in attracting new customers. This project " **THE INTELLIGENT TELEPHONE** " aims at designing a telephone with the following features

- ❑ SMS (Short Messaging Service)
- ❑ Remote Control of Devices
- ❑ Call Register
- ❑ Address Book
- ❑ Caller ID

Introduction

1. INTRODUCTION

If you have started treating your landline phone as a poor cousin of your cellular instrument, think again! Some exciting times are ahead with your wired friend as it helps send short messages and enjoy many other incredible features.

The mobile phone industry has stolen much of the thunder in past few years, with some spectacular growth rates. In certain pockets, the number of mobile connections exceeded the number of fixed-line connections. This growth in mobile telephony may continue.

But, fixed-lines are creating some unique propositions of their own. We extend most of the cell phone services to basic fixed-line telephones.

This project is proposed for **BSNL** and it would help BSNL in attracting new customers.

1.1 EXISTING SYSTEM AND ITS LIMITATIONS

The existing landline telephone network is the public switched telephone network (PSTN) and it's the oldest and largest telecommunications network in existence.

It offers many services to customers. The most important teleservices provided by landline telephone subscribers are

- Fixed telephony;
- Cordless telephony, or "fixed telephony with limited terminal mobility";
- Fax (via a built-in modem); and
- Data communication (via modem).

Certain features offered by the cellphones such as SMS, ring tones, address book, call register, caller ID are not provided with the landline telephones. So many landline customers prefer cell phones.

1.2 PROPOSED SYSTEM AND ITS ADVANTAGES

The proposed system offers most of the features of the cell phones such as SMS, address book, call register, caller ID to the landline telephones and certain extra features such as remote controlling of devices. This helps the landline service providers to attract more customers.

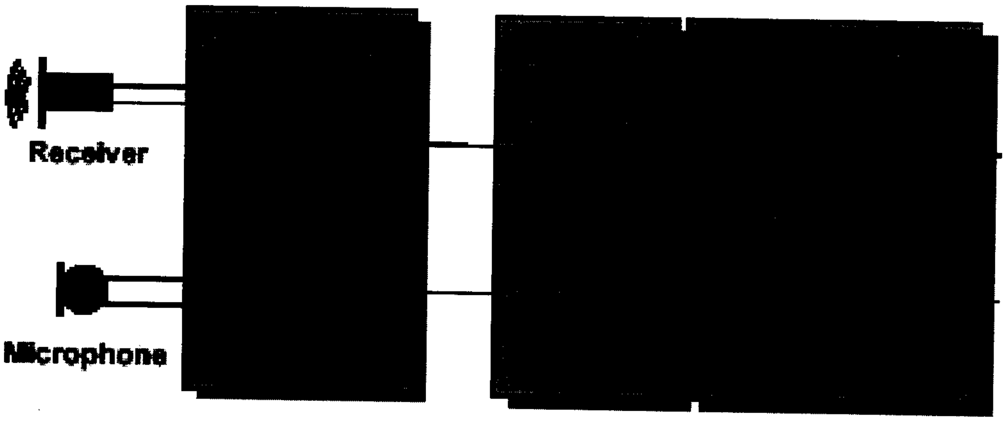
The user has to attach this gadget in parallel to the telephone line, and give power supply and the gadget is ready for operation. No extra installation is needed and no extra costs incurred.

Literature Survey

2. LITERATURE SURVEY

The Telephone

The telephones and telephone company practices may vary from one locality to another, but the basic principles underlying the way they work remain unchanged.

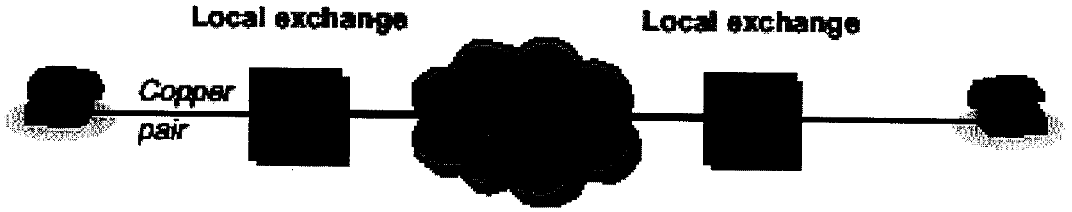


Every telephone consists of three separate subassemblies, each capable of independent operation. These assemblies are the speech network, the dialing mechanism, and the ringer or bell. Together, these parts - as well as any additional devices such as modems, dialers, and answering machines - are attached to the phone line.

The Phone line

A telephone is usually connected to the telephone exchange by about three miles (4.83 km) of a twisted pair of No.22 (AWG) or 0.5 mm copper wires, known by the phone company as "the loop". The resistance of No.22 AWG wire is 16.46 Ohms per thousand feet at 77 degrees F (25 degrees C). Wire resistance is usually expressed as Ohms per kilometer.

The Phone Line



The telephone apparatus is generally considered to be current driven, hence all phone measurements refer to current consumption, not voltage. The length of the wire connecting the subscriber to the telephone exchange affects the total amount of current that can be drawn by anything attached at the subscriber's end of the line.

The voltage applied to the line to drive the telephone is 50 VDC. Telephones are peculiar in that the signal line is also the power supply line. The voltage is supplied by lead acid cells, thus assuring a hum-free supply and complete independence from the electric company, which may be especially useful during power outages.

The phone line has impedance composed of distributed resistance, capacitance, and inductance. The impedance will vary according to the length of the loop, the type of insulation of the wire, and whether the wire is aerial cable, buried cable, or bare parallel wires strung on telephone poles.

The Speech Network

The speech network - also known as the "hybrid" or the "two wire/four wire network" - takes the incoming signal and feeds it to the earpiece and takes the microphone output and feeds it down the line. The standard network used all over the world is an LC device with a carbon microphone; some newer phones use discrete transistors or ICs. One of the advantages of an LC network is that it has no semiconductors, is not voltage sensitive, and will work continuously as the voltage across the line is reduced. Many transistorized phones stop working as the voltage approaches 3 to 4 Volts.

When a telephone is taken off the hook, the line voltage drops from 48 Volts to between 9 and 3 Volts, depending on the length of the loop. If another telephone in parallel is taken off the hook, the current consumption of the line will remain the same and the voltage across the terminals of both telephones will drop.

While low levels of audio may be difficult to hear, overly loud audio can be painful. Consequently, a well designed telephone will automatically adjust its transmit and receive levels to allow for the attenuation - or lack of it - caused by the length of the loop. This adjustment is called "loop compensation."

Because a telephone is a duplex device, both transmitting and receiving on the same pair of wires, the speech network must ensure that not too much of the caller's voice is fed back into his or her receiver. This function, called "side tone," is achieved by phasing the signal so that some cancellation occurs in the speech network before the signal is fed to the receiver.

The Dial

There are two types of dials in use around the world. The most common one is called pulse, loop disconnect, or rotary; the oldest form of dialing, it's been with us since the 1920's. The other dialing method, more modern and much loved by Radio Amateurs is called Touch-tone, Dual Tone Multi-Frequency (DTMF) or Multi-Frequency (MF) in Europe.

Pulse dialing

Pulse dialing is traditionally accomplished with a rotary dial, which is a speed-governed wheel with a cam that opens and closes a switch in series with your phone and the line. It works by actually disconnecting or "hanging up" the telephone at specific intervals. The standard is one disconnect per digit, so if we dial a "1," your telephone is "disconnected" once. Dial a seven and one we will be "disconnected" seven times; dial a zero, and it'll "hang up " ten times. Some countries invert the system so "1" causes ten "disconnects" and 0, one disconnect. Some add a digit so that dialing a 5 would cause six disconnects and 0, eleven disconnects. There are even some systems in which dialing 0 results in one disconnect, and all other digits are plus one, making a 5 cause six disconnects and 9, ten disconnects.

Although most exchanges use rates of 6 to 15 Pulses Per Second (PPS), the phone company accepted standard is 8 to 10 PPS. Some modern digital exchanges, free of the mechanical inertia problems of older systems, will accept a PPS rate as high as 20.

Besides the PPS rate, the dialing pulses have a make/break ratio, usually described as a percentage, but sometimes as a straight ratio. This is the pulse measured at the telephone, not at the exchange, where it's somewhat different, having traveled through the phone line with its distributed resistance, capacitance, and inductance. In practice, the make/break ratio does not seem to affect the performance of the dial when attached to a normal loop. Each pulse is a switch connect and disconnect across complex impedance, so the switching transient often reaches 300 Volts.

Most pulse dialing phones produced today use a CMOS IC and a keyboard. Instead of pushing our finger round in circles, then removing our finger and waiting for the dial to return before dialing the next digit, we punch the button as fast as we want. The IC stores the number and pulses it out at the correct rate with the correct make/break ratio and the switching is done with a high-voltage switching transistor. Because the IC has already stored the dialed number in order to pulse it out at the correct rate, it's a simple matter for telephone designers to keep the memory "alive" and allow the telephone to store, recall, and redial the Last Number Dialed (LND). This feature enables us to redial by picking up the handset and pushing just one button.

Tone dialing

Touch-tone, the most modern form of dialing is fast and less prone to error than pulse dialing. Compared to pulse, its major advantage is that its audio band signals can travel down phone lines further than pulse, which can travel only as far as the local exchange Touch-tone can therefore send signals around the world via the telephone lines, and can be used to control phone answering machines and computers.

DTMF

Bell Labs developed DTMF in order to have a dialing system that could travel across microwave links and work rapidly with computer-controlled exchanges. Each transmitted digit consists of two separate audio tones that are mixed together. The four vertical columns on the keypad are known as the high group and the four horizontal rows as the low group; the digit 8 is composed of 1336 Hz and 852 Hz. The level of each tone is within 3 dB of the other, (the telephone company calls this "Twist"). A complete touch-tone pad has 16 digits, as opposed to ten on a pulse dial. Besides the numerals 0 to 9, a DTMF "dial" has *, #, A, B, C, and D. Although the letters are not normally found on consumer telephones, the IC in the phone is capable of generating them.

The * sign is usually called "star" or "asterisk." The # sign, often referred to as the "pound sign." is actually called an octothorpe. They are used for control purposes; phone answering machines, bringing up remote bases, electronic banking, and repeater control.

When DTMF dials first came out they had complicated cams and switches for selecting the digits and used a transistor oscillator with an LC tuning network to generate the tones. Modern dials use a matrix switch and a CMOS IC that synthesizes the tones from a 3.57MHz (TV color burst) crystal. This oscillator runs only during dialing, so it doesn't normally produce QRM.

Standard DTMF dials will produce a tone as long as a key is depressed. No matter how long you press, the tone will be decoded as the appropriate digit. The shortest duration in which a digit can be sent and decoded is about 100 milliseconds (ms). It's pretty difficult to dial by hand at such a speed, but automatic dialers can do it. An automatic dialer can dial a twelve-digit long distance number in a little more than a second - about as long as it takes a pulse dial to send a single 0 digit.

The Ringer

This is a device that alerts us to an incoming call. It may be a bell, light, or warbling tone. The telephone company sends a ringing signal, which is an AC waveform. Frequencies between 20 and 40 Hz are used. The voltage at the subscribers end depends upon loop length and number of ringers attached to the line; it could be between 40 and 150 Volts. The telephone company may or may not remove the 48 VDC during ringing.

The ringing cadence - the timing of ringing to pause - varies from company to company. The most common ringing device is the gong ringer, a solenoid coil with a clapper that strikes either a single or double bell. A gong ringer is the loudest signaling device that is solely phone-line powered.

Modern telephones tend to use warbling ringers, which are usually ICs powered by the rectified ringing signal. The audio transducer is either a piezoceramic disk or a small loudspeaker via a transformer.

Ringers are isolated from the DC of the phone line by a capacitor. Telephone companies in other parts of the world use capacitors between 0.2 and 2.0 μF. The paper capacitors of the past have been replaced almost exclusively with capacitors made of Mylar film. Their voltage rating is always 250 Volts. The capacitor and ringer coil, or Zeners in a warbling ringer, constitute a resonant circuit.

System Requirements

Software Requirements Specification

3. SYSTEM REQUIREMENTS

3.1 HARDWARE AND SOFTWARE REQUIREMENTS

The hardware requirements are

- ❑ 16F877 PIC microcontroller
- ❑ M8880 DTMF Transceiver
- ❑ Ring Detector
- ❑ Latch
- ❑ LCD

The software requirements are

Assembly language with the instruction set of the PIC microcontroller PIC16F877 is used to generate the source code, which is converted into hex code to be embedded in the microcontroller.

MPLAB IDE is needed to write the code, compile and build and debug.

Software Requirements Specification

4. SOFTWARE REQUREMENTS SPECIFICATION

4.1 INTRODUCTION

4.1.1 PURPOSE

The purpose of the project 'Intelligent phone' is to extend few of the cell phone services to basic fixed-line telephones. This project is developed for Bharat Sanchar Nigam Limited (BSNL), Coimbatore.

4.1.2 SCOPE

The project is developed to bring in additional features to the current landline connection to help retain the service providers like BSNL, the largest telecom company in India and the 7th largest in the world to attract new customers.

4.1.3 OVERVIEW

This SRS document lists the product requirements, its functions, hardware requirements and constraints.

4.2 GENERAL DESCRIPTION

4.2.1 PRODUCT PERSPECTIVE

This project provides a powerful vehicle for service differentiation for landline providers. If the market allows for it, it can also represent an additional source of revenue for the service provider.

4.2.2 PRODUCT FUNCTIONS

The product offers various functionalities such as

- ❑ SMS (Short Messaging Service)
- ❑ Remote Control of Devices
- ❑ Call Register
- ❑ Address Book
- ❑ Caller ID

4.2.3 USER CHARACTERISTICS

The user has to know the codes required for the controlling operations over telephone. The user has to be made familiar with the keypad mapping techniques to compose message and store names in address book.

4.2.4 GENERAL CONSTRAINTS

The telephone instrument must support Touch tone dialing or DTMF.

4.3 SPECIFIC REQUIREMENTS

4.3.1 FUNCTIONAL REQUIREMENTS

4.3.1.1 INTRODUCTION

SMS:

SMS provides a mechanism for transmitting short text messages to and from wireless devices. Here the user dialed number along with the control messages are received, the latter control messages are interpreted and if it's a SMS code the following text is sent.

REMOTE CONTROL OF DEVICES:

The user-dialed code is interpreted to be for remote control of devices and the following control signals are used by the micro controller to control devices.

CALL REGISTER:

A list of Dialed calls is maintained.

4.3.1.2 LIST OF INPUT

SMS:

- □ Code for SMS
- □ SMS
- □ Text

REMOTE CONTROL OF DEVICES:

- ❑ Code for Remote Controlling
- ❑ Device Code
- ❑ 1 or 0 to switch device on or off

CALL REGISTER:

- ❑ Incoming Call Number

4.3.1.3 INFORMATION PROCESSING REQUIREMENTS

SMS:

The control code identifies that the forthcoming data is SMS text. The data is typed using Keypad Mapping Technique, it is stored and the recipient number is got and it is dialed, a connection is established and the tones corresponding to the SMS text is sent to the receiving side after an acknowledgement is received from the other side on establishment of connection and "a message sent" notification is displayed after the tones are sent else on failure of reception of acknowledgement from recipient side "a host link failure" message is notified.

REMOTE CONTROL OF DEVICES:

After the call is established, based on the device number, the micro controller identifies the device and sends control signals to it appropriately to switch it on or off.

CALL REGISTER:

The incoming numbers are stored by default in the call register.

CALLER ID:

When an incoming call comes the number is displayed on the LCD as the phone rings. If the caller is identified on match with an entry in the address book, the name and number is also displayed during ring.

4.3.2 PERFORMANCE REQUIREMENTS

4.3.2.1 AVAILABILITY

Any landline user who attaches the gadget has the facility installed automatically.

4.3.2.2 CAPACITY

- For a single call only one facility can be availed.
- For SMS, any number of characters can be sent.
- For remote controlling of devices, the number of devices that can be controlled depends on the micro controller used.

4.3.3 DESIGN CONSTRAINTS

4.3.3.1 HARDWARE LIMITATIONS

Minimal requirements are:

A gadget consisting of

- LCD
- Microcontroller
- DTMF Transceiver
- Ring Detector
- Latch

4.3.4 GLOSSARY

SMS

Short Messaging Service

DTMF

Dual Tone Multi Frequency

PSTN

Public Switched Telephone Network

PIC

Programmable Interface Controller

LCD

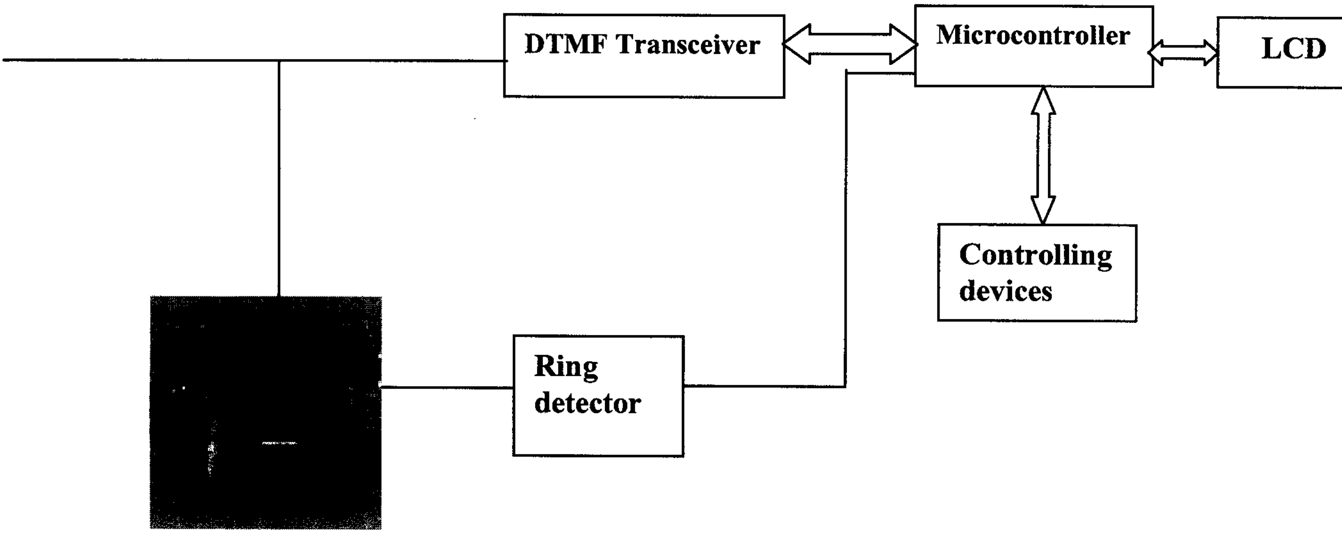
Liquid Crystal Display

System Design

5. SYSTEM DESIGN

5.1 STRUCTURE DIAGRAM

The product structure is explained in detail in the structure diagram. The working of the product is described here.



DTMF TRANSCEIVER

In telephone keypad each key is considered to be an intersection of row and column, and a tone is generated for each row and column, these 2 tones are decoded by the DTMF transceiver and sent to the microcontroller.

MICROCONTROLLER

The microcontroller receives the tones and checks for control codes and handles appropriately. The programmed chip sends the SMS data and dialed numbers to the LCD. It also controls the devices it is connected, based on controlling codes.

RING DETECTOR

The Ring Detector detects the voltage when the phone rings, and sends a signal to indicate a call to microcontroller that handles it by latching the call after a predefined number of rings.

5.2 MODULE SPECIFICATION

SMS

Initially the user composes a message prefixing and post fixing the control codes. The user is asked to enter the recipient number. The microcontroller dials the number and sends the SMS data after it receives an acknowledgment from the called number.

REMOTE CONTROL OF DEVICES

The user dials the recipient's number and after the connection is established, dials the code for controlling devices attached on the other end.

CALL REGISTER

A list of Received calls is maintained automatically as the calls are received.

ADDRESS BOOK

A list of phone numbers and names can be entered by user and stored.

CALLER ID

The caller number is displayed during ringing of telephone. If the caller name and number is stored in address book, the caller name is displayed.

User Manual

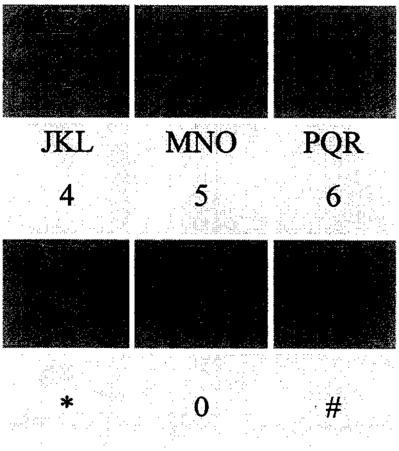
6. USER MANUAL

KEYPAD MAPPING TECHNIQUE

Typing any alphabets (say 'K') or character can be done as under.

1. Pressing the key where the alphabet appears (for 'K' press '4')
2. A string of 2 or 3 alphabets is mapped for each key. Press the key that corresponds to the position of the alphabets in the string. ('K' corresponds to 2nd alphabet in the string on the key 5. So press '2')
3. So when you press the sequence 4 2, it will understand the alphabet as K.
4. For digits press the key followed by 0 (for '2' press '2 0')
5. For space press '0 1'

Key Pad Mapping



SMS

To compose a SMS

- Type #* to begin composing SMS
- Using keypad mapping technique type the message
- As the message is typed, it is displayed on LCD to facilitate the user to check his composed message
- The user can reset using the button provided on the kit in case of an error in composing procedure
- Type ## to end composing SMS
- The recipient number is entered by the user who can then keep the handset down
- The call is made from the caller phone, on getting the acknowledgement from the called number, the tones corresponding to the sms data is sent
- If the acknowledgement is not received within the predefined time the message "host link failed"

Example:

To send a message "HELLO"

Type

#* 32 22 43 43 53 ##

Recipient number?

2312312

To remote control a device

- Dial the recipient number
- On connection establishment, the user should type *, the code for remote controlling devices
- The code is followed by the device number that is defined by the pin of microcontroller it is attached to.
- The user has to finally dial in the code 1 or 0 for switching on or off the device

Example:

Dial Recipient number

***11 (to switch 1st device on)**

To view stored messages

- □ Type #1 for the viewing stored messages followed by message number

Example:

#11 (to view first message)

To view call register

- □ Type #2 for the viewing stored incoming numbers followed by the position.

Example:

#21 (to view last called numbers)

To view address book

- □ Type #3 for the viewing name and number stored in address book followed by the position.

Example:

#31 (to view fist entry in address book)

To add an entry in the address book

- □ Type #4, then the position in address book
- □ Enter the number and name to store in address book

Example:

Type #4 1

Number?

2534342

Name?

Sfadfsd

Product Testing

7. PRODUCT TESTING

FUNCTIONAL TEST

The **MPLAB SIM** is used to perform the functional testing of the product.

MPLAB SIM is a discrete-event simulator for the PIC microcontroller (MCU) families. It is integrated into MPLAB IDE integrated development environment. The MPLAB SIM debugging tool is designed to model operation of Microchip Technology's PIC microcontrollers to assist users in debugging software for these devices.

MPLAB SIM allows to:

❑ Modify object code and immediately re-execute it
❑ Inject external stimuli to the simulated processor
❑ Set pin and register values at pre specified intervals
❑ Trace the execution of the object code

The product was tested and found to be working properly under all testing conditions.

Project Legacy

8. PROJECT LEGACY

8.1 PROJECT DESCRIPTION

The product is an electronic gadget, which is attached to the existing landline telephone. After it is attached it is used to send SMS, control signals to other landline telephones with the similar gadget attached. It is also used to maintain address book, call register, view caller id.

8.2 INITIAL EXPECTATIONS

The main purpose of the project is to bring the key feature of the SMS to the landline telephone and few other features of the cell phone and extra features like remote controlling of household appliances.

8.3 CURRENT STATUS OF THE PROJECT

The product has been made to meet its initial expectations and is made flexible to accept future changes and updations.

8.4 TECHNICAL LESSONS LEARNT

Many technical lessons were learnt while working on the project. They are programming PIC microcontrollers in embedded assembly language, architectures of PIC microcontrollers, DTMF transceivers, and how to interconnect the hardware components in a multi purpose board.

8.5 MANAGERIAL LESSONS LEARNT

During the course of the project, apart from the technical lessons, certain Managerial aspects were also learnt. The systematic approach to a program, effective communication with the guide, and above all the teamwork helped in successful completion of the project. The time slots were set for each and every module and the time taken were noted after their completion and hence effective management of time was learnt. Maintenance of the project note helped in achieving the above.

8.6 RECOMMENDATIONS FOR FUTURE

The product can be enhanced with additional features like redirect calls, assign tones, call duration limiter, voice mail, email, games, MMS.

Conclusion

Bibliography

10. BIBLIOGRAPHY

a) Books

i) Regis J. (Bud) Bates, Donald W. Gregory, "Voice and Data Communication"

ii) Ted Van Sickle, "Programming Microcontrollers in C"

b) Websites

i) www.microchip.com

The specifications of the microcontroller PIC16F877 to be used along with architecture and instruction set were referred.

MPLAB software and its manual were present in this site.

ii) http://www.mobilecomms-technology.com

The current SMS technology details were referred.

iii) www.motorola.com

The specifications of the DTMF transceiver MT8880 to be used along with architecture and instruction set were referred.

Appendix

11 APPENDIX

11.1 DATASHEETS

PIC 16F877

MICROCONTROLLER CORE FEATURES

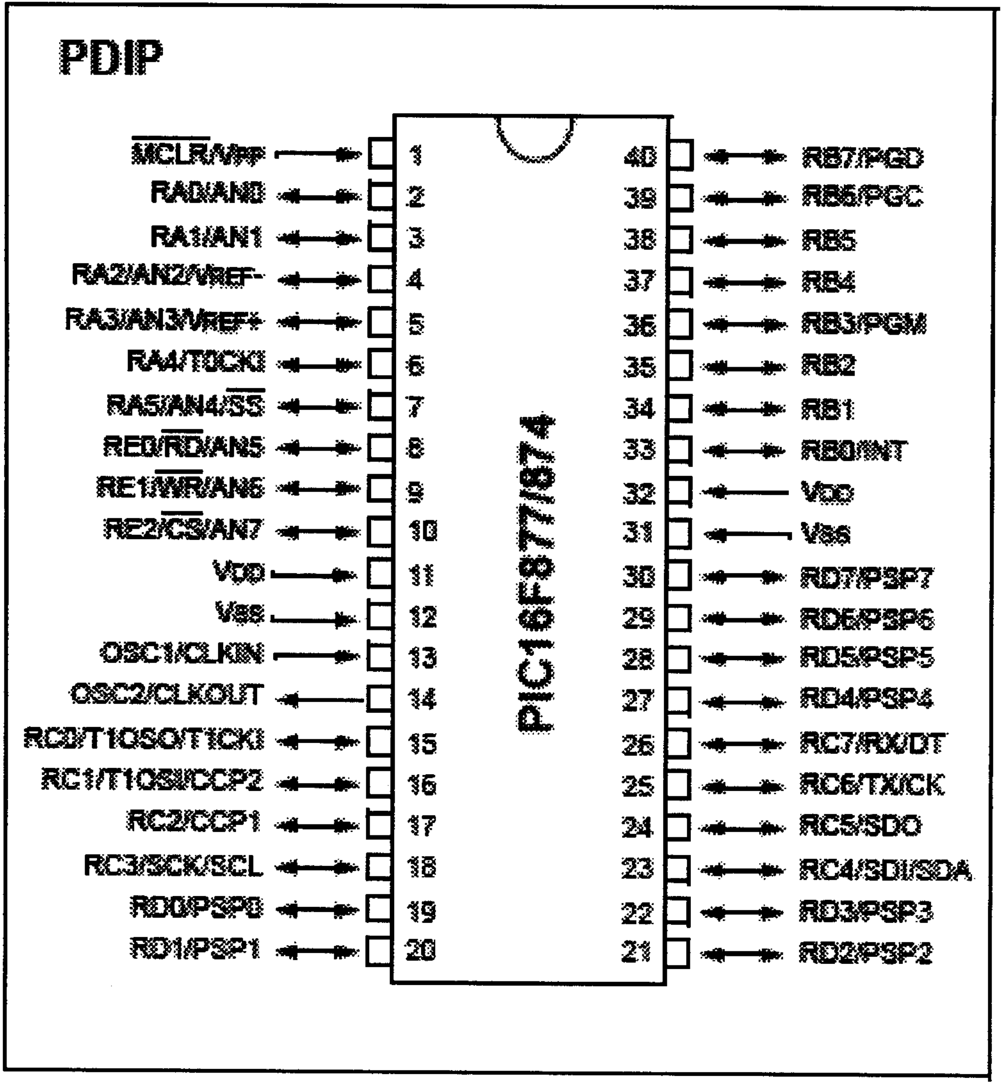
- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
- Up to 368 x 8 bytes of Data Memory (RAM)
- Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming□ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA

- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
  < 0.6 mA typical @ 3V, 4 MHz
  20 μA typical @ 3V, 32 kHz
  < 1 μA typical standby current

PERIPHERAL FEATURES

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
- Capture is 16-bit, max. resolution is 12.5 ns
- Compare is 16-bit, max. resolution is 200 ns
- PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI□ (Master mode) and I2C□ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

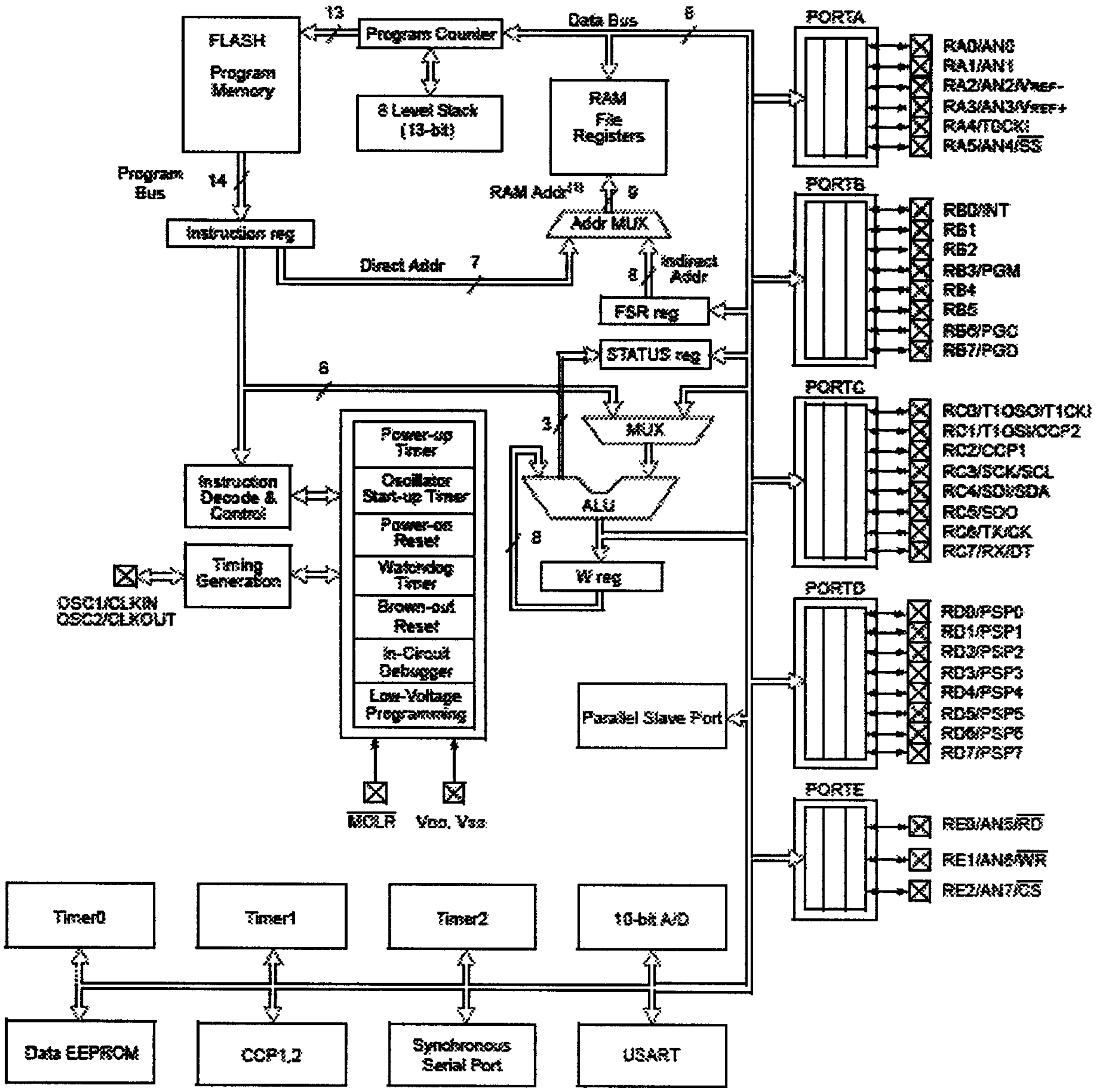
Pin Diagram


PDIP
PIC16F877/874
MCLR/VPP 1
RA0/AN0 2
RA1/AN1 3
RA2/AN2/VREF- 4
RA3/AN3/VREF+ 5
RA4/T0CKI 6
RA5/AN4/SS 7
RE0/RD/AN5 8
RE1/WR/AN6 9
RE2/CS/AN7 10
VDD 11
VSS 12
OSC1/CLKIN 13
OSC2/CLKOUT 14
RC0/T1OSO/T1CKI 15
RC1/T1OSI/CCP2 16
RC2/CCP1 17
RC3/SCK/SCL 18
RD0/PSP0 19
RD1/PSP1 20
40 RB7/PGD
39 RB6/PGC
38 RB5
37 RB4
36 RB3/PGM
35 RB2
34 RB1
33 RB0/INT
32 VDD
31 VSS
30 RD7/PSP7
29 RD6/PSP6
28 RD5/PSP5
27 RD4/PSP4
26 RC7/RX/DT
25 RC6/TX/CK
24 RC5/SDO
23 RC4/SDI/SDA
22 RD3/PSP3
21 RD2/PSP2

Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS[4]	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{\text{MCLR}}$/VPP	1	2	18	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
						PORTA is a bi-directional I/O port.
RA0/AN0	2	3	19	I/O	TTL	RA0 can also be analog input0.
RA1/AN1	3	4	20	I/O	TTL	RA1 can also be analog input1.
RA2/AN2/VREF-	4	5	21	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage.
RA3/AN3/VREF+	5	6	22	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage.
RA4/T0CKI	6	7	23	I/O	ST	RA4 can also be the clock input to the Timer0 timer/counter. Output is open drain type.
RA5/$\overline{\text{SS}}$/AN4	7	8	24	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
						PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	33	36	8	I/O	TTL/ST[1]	RB0 can also be the external interrupt pin.
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3/PGM	36	39	11	I/O	TTL	RB3 can also be the low voltage programming input.
RB4	37	41	14	I/O	TTL	Interrupt-on-change pin.
RB5	38	42	15	I/O	TTL	Interrupt-on-change pin.
RB6/PGC	39	43	16	I/O	TTL/ST[2]	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	40	44	17	I/O	TTL/ST[2]	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.

2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.

3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).

4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION (CONTINUED)

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
						PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	RC0 can also be the Timer1 oscillator output or a Timer1 clock input.
RC1/T1OSI/CCP2	16	18	35	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	17	19	36	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	18	20	37	I/O	ST	RC3 can also be the synchronous serial clock input/ output for both SPI and I^2C modes.
RC4/SDI/SDA	23	25	42	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	24	26	43	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	25	27	44	I/O	ST	RC6 can also be the USART Asynchronous Transmi or Synchronous Clock.
RC7/RX/DT	26	29	1	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.
						PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.
RD0/PSP0	19	21	38	I/O	ST/TTL(3)	
RD1/PSP1	20	22	39	I/O	ST/TTL(3)	
RD2/PSP2	21	23	40	I/O	ST/TTL(3)	
RD3/PSP3	22	24	41	I/O	ST/TTL(3)	
RD4/PSP4	27	30	2	I/O	ST/TTL(3)	
RD5/PSP5	28	31	3	I/O	ST/TTL(3)	
RD6/PSP6	29	32	4	I/O	ST/TTL(3)	
RD7/PSP7	30	33	5	I/O	ST/TTL(3)	
						PORTE is a bi-directional I/O port.
RE0/$\overline{RD}$/AN5	8	9	25	I/O	ST/TTL(3)	RE0 can also be read control for the parallel slave port, or analog input5.
RE1/$\overline{WR}$/AN6	9	10	26	I/O	ST/TTL(3)	RE1 can also be write control for the parallel slave port, or analog input6.
RE2/$\overline{CS}$/AN7	10	11	27	I/O	ST/TTL(3)	RE2 can also be select control for the parallel slave port, or analog input7.
VSS	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
VDD	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28,40	12,13,33,34		—	These pins are not internally connected. These pins should be left unconnected.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

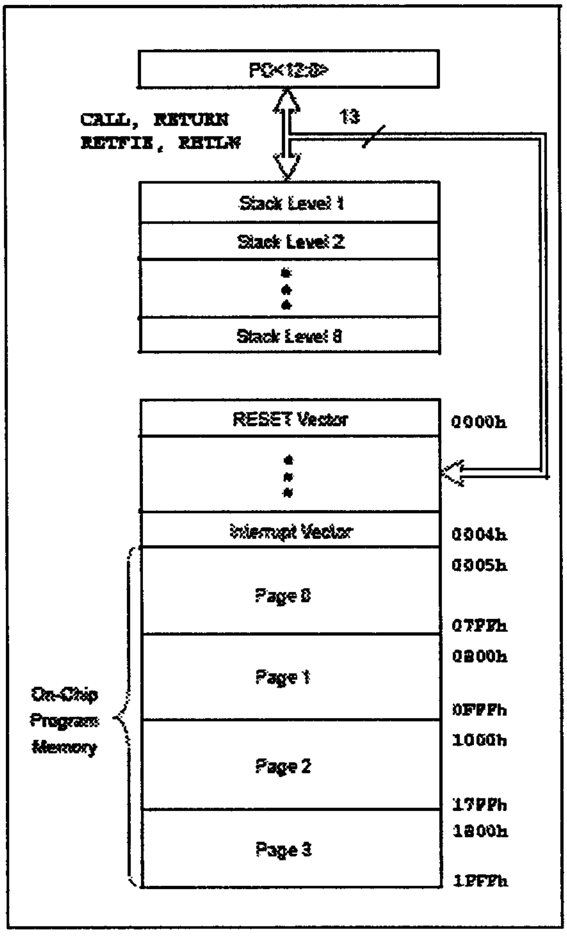
MEMORY ORGANIZATION

There are three memory blocks in each of the PIC16F87X MCUs. The Program Memory and Data Memory have separate buses so that concurrent access can occur.

Program Memory Organization

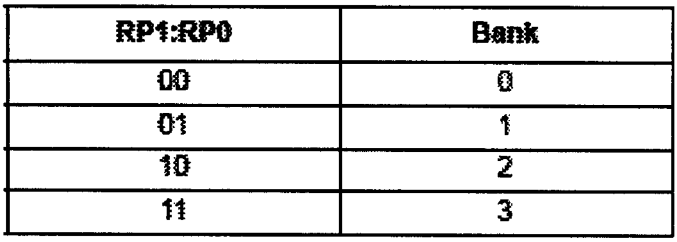
The PIC16F87X devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F877/876 devices have 8K x 14 words of FLASH program memory. Accessing a location above the physically implemented address will cause a wraparound. The RESET vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-1: PIC16F877/876 PROGRAM MEMORY MAP AND STACK



DATA MEMORY ORGANIZATION

The data memory is partitioned into multiple banks, which contain the General Purpose Registers and the Special Function Registers. Bits RP1 (STATUS<6>) and RP0 (STATUS<5>) are the bank select bits. Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some frequently used Special Function Registers from one bank may be mirrored in another bank for code reduction and quicker access.

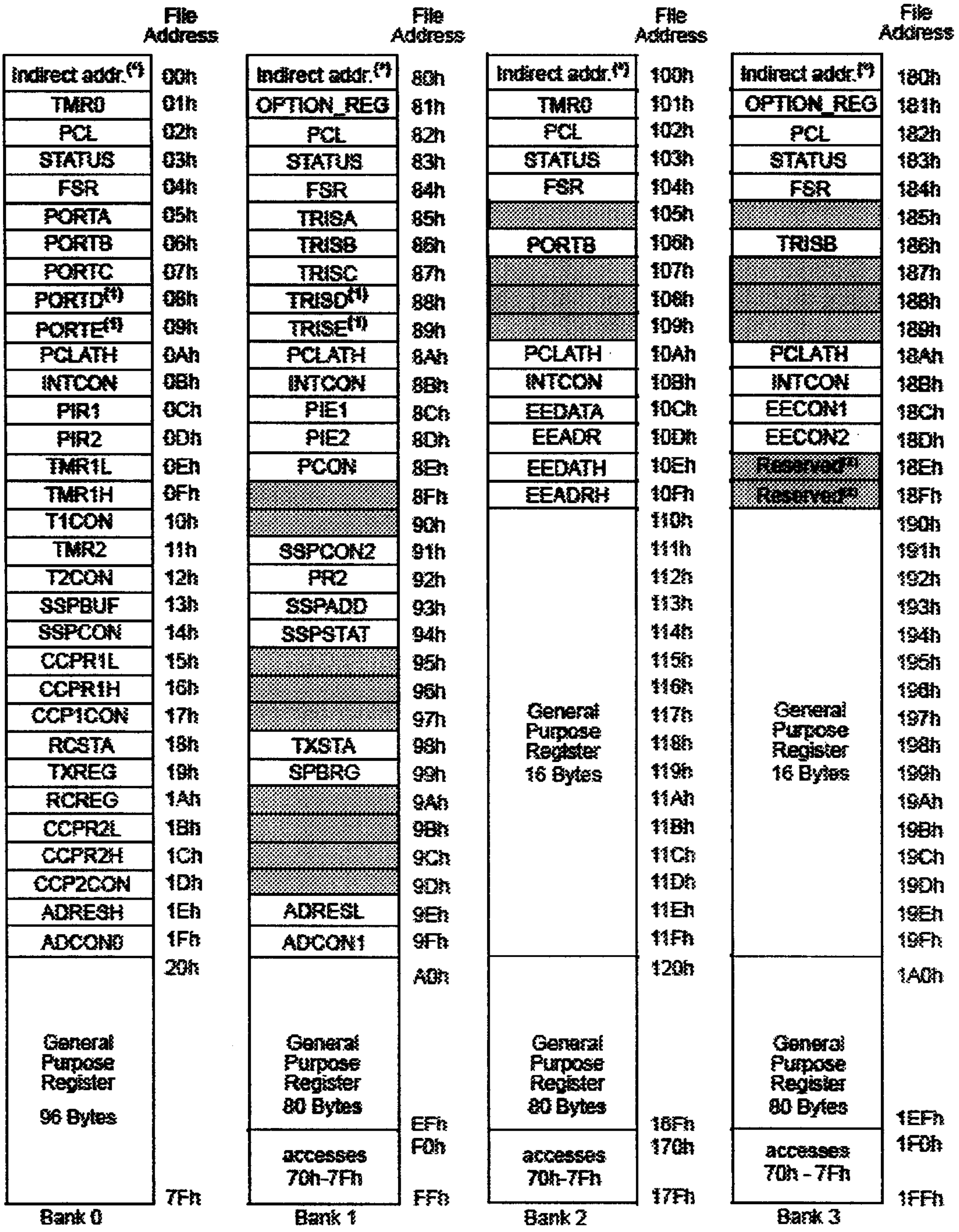


RP1:RP0	Bank
00	0
01	1
10	2
11	3

GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly, or indirectly through the File Select Register (FSR).

FIGURE 2-3: PIC16F877/876 REGISTER FILE MAP



	File Address		File Address		File Address		File Address
Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h	▨	105h	▨	185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h	▨	107h	▨	187h
PORTD(1)	08h	TRISD(1)	88h	▨	108h	▨	188h
PORTE(1)	09h	TRISE(1)	89h	▨	109h	▨	189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved(2)	18Eh
TMR1H	0Fh	▨	8Fh	EEADRH	10Fh	Reserved(2)	18Fh
T1CON	10h	▨	90h	General Purpose Register 16 Bytes	110h	General Purpose Register 16 Bytes	190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h	▨	95h		115h		195h
CCPR1H	16h	▨	96h		116h		196h
CCP1CON	17h	▨	97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah	▨	9Ah		11Ah		19Ah
CCPR2L	1Bh	▨	9Bh		11Bh		19Bh
CCPR2H	1Ch	▨	9Ch		11Ch		19Ch
CCP2CON	1Dh	▨	9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Register 96 Bytes	20h	General Purpose Register 80 Bytes	A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h
			EFh		16Fh		1EFh
		accesses 70h-7Fh	F0h	accesses 70h-7Fh	170h	accesses 70h - 7Fh	1F0h
	7Fh		FFh		17Fh		1FFh
Bank 0		Bank 1		Bank 2		Bank 3	

▨ Unimplemented data memory locations, read as '0'.

* Not a physical register.

Note 1: These registers are not implemented on the PIC16F876.

2: These registers are reserved, maintain these registers clear.

SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 2-1. The Special Function Registers can be classified into two sets: core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in the peripheral features section.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:
Bank 0											
00h[3]	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
01h	TMR0	Timer0 Module Register								xxxx xxxx	47
02h[3]	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26
03h[3]	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	18
04h[3]	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	29
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	31
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	33
08h[4]	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	35
09h[4]	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	36
0Ah[1,3]	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
0Bh[3]	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
0Ch	PIR1	PSPIF[2]	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	22
0Dh	PIR2	—	(5)	—	EEIF	BCLIF	—	—	CCP2IF	-r-0 0--0	24
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	52
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	52
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	51
11h	TMR2	Timer2 Module Register								0000 0000	55
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	55
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	70, 73
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	67
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	57
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	57
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	58
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	96
19h	TXREG	USART Transmit Data Register								0000 0000	99
1Ah	RCREG	USART Receive Data Register								0000 0000	101
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	57
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	57
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	58
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	116
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	—	ADON	0000 00-0	111

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

3: These registers can be addressed from any bank.

4: PORTD, PORTE, TRISD, and TRISE are not physically implemented on PIC16F873/876 devices; read as '0'.

5: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:
Bank 1											
80h[3]	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
81h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	19
82h[3]	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26
83h[3]	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	18
84h[3]	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	29
86h	TRISB	PORTB Data Direction Register								1111 1111	31
87h	TRISC	PORTC Data Direction Register								1111 1111	33
88h[4]	TRISD	PORTD Data Direction Register								1111 1111	35
89h[4]	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	37
8Ah[1,3]	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
8Bh[3]	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
8Ch	PIE1	PSPIE[2]	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	21
8Dh	PIE2	—	(5)	—	EEIE	BCLIE	—	—	CCP2IE	-r-0 0--0	23
8Eh	PCON	—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	---- --qq	25
8Fh	—	Unimplemented								—	—
90h	—	Unimplemented								—	—
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	68
92h	PR2	Timer2 Period Register								1111 1111	55
93h	SSPADD	Synchronous Serial Port (I^2C mode) Address Register								0000 0000	73, 74
94h	SSPSTAT	SMP	CKE	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	UA	BF	0000 0000	66
95h	—	Unimplemented								—	—
96h	—	Unimplemented								—	—
97h	—	Unimplemented								—	—
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	95
99h	SPBRG	Baud Rate Generator Register								0000 0000	97
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	—	Unimplemented								—	—
9Dh	—	Unimplemented								—	—
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	116
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000	112

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

3: These registers can be addressed from any bank.

4: PORTD, PORTE, TRISD, and TRISE are not physically implemented on PIC16F873/876 devices; read as '0'.

5: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Det o pa
Bank 2											
100h[3]	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	2
101h	TMR0	Timer0 Module Register								xxxx xxxx	4
102h[3]	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	2
103h[3]	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	1
104h[3]	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	2
105h	—	Unimplemented								—	—
106h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	3
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah[1,3]	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	2
10Bh[3]	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	2
10Ch	EEDATA	EEPROM Data Register Low Byte								xxxx xxxx	4
10Dh	EEADR	EEPROM Address Register Low Byte								xxxx xxxx	4
10Eh	EEDATH	—	—	EEPROM Data Register High Byte						xxxx xxxx	4
10Fh	EEADRH	—	—	—	EEPROM Address Register High Byte					xxxx xxxx	4
Bank 3											
180h[3]	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	2
181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1
182h[3]	PCL	Program Counter (PC) Least Significant Byte								0000 0000	2
183h[3]	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	1
184h[3]	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	2
185h	—	Unimplemented								—	—
186h	TRISB	PORTB Data Direction Register								1111 1111	3
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah[1,3]	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	2
18Bh[3]	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	2
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	41,
18Dh	EECON2	EEPROM Control Register2 (not a physical register)								---- ----	4
18Eh	—	Reserved maintain clear								0000 0000	—
18Fh	—	Reserved maintain clear								0000 0000	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

3: These registers can be addressed from any bank.

4: PORTD, PORTE, TRISD, and TRISE are not physically implemented on PIC16F873/876 devices; read as '0'.

5: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

STATUS REGISTER

The STATUS register contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory. The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the TO and PD bits are not writable, therefore, the result of an instruction with the STATUS register as destination may be different than intended. For example, CLRF STATUS will clear the upper three bits and set the Z bit. This leaves the STATUS register as 000u u1uu (where u = unchanged). It is recommended, therefore, that only BCF, BSF, SWAPF and MOVWF instructions are used to alter the STATUS register, because these instructions do not affect the Z, C or DC bits from the STATUS register. For other instructions not affecting any status bits, see the “Instruction Set Summary."
Note: The C and DC bits operate as a borrow and digit borrow bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

REGISTER 2-1: STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

bit 7 IRP: Register Bank Select bit (used for indirect addressing)

1 = Bank 2, 3 (100h - 1FFh)

0 = Bank 0, 1 (00h - FFh)

bit 6-5 RP1:RP0: Register Bank Select bits (used for direct addressing)

11 = Bank 3 (180h - 1FFh)

10 = Bank 2 (100h - 17Fh)

01 = Bank 1 (80h - FFh)

00 = Bank 0 (00h - 7Fh)

Each bank is 128 bytes

bit 4 TO: Time-out bit

1 = After power-up, CLRWDT instruction, or SLEEP instruction

0 = A WDT time-out occurred

bit 3 PD: Power-down bit

1 = After power-up or by the CLRWDT instruction

0 = By execution of the SLEEP instruction

bit 2 Z: Zero bit

1 = The result of an arithmetic or logic operation is zero

0 = The result of an arithmetic or logic operation is not zero

bit 1 DC: Digit carry/borrow bit (ADDWF, ADDLW,SUBLW,SUBWF instructions)

(for borrow, the polarity is reversed)

1 = A carry-out from the 4th low order bit of the result occurred

0 = No carry-out from the 4th low order bit of the result

bit 0 C: Carry/borrow bit (ADDWF, ADDLW,SUBLW,SUBWF instructions)

1 = A carry-out from the Most Significant bit of the result occurred

0 = No carry-out from the Most Significant bit of the result occurred

Note: For borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high, or low order bit of the source register.

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'

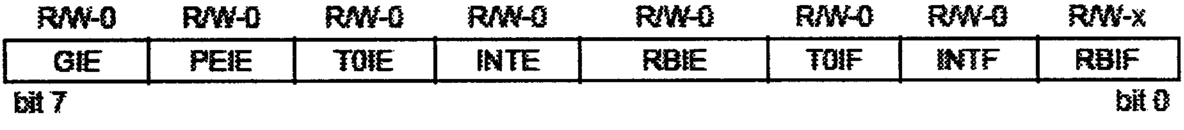
- n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

INTCON Register

The INTCON Register is a readable and writable register, which contains various enable and flag bits for the TMR0 register overflow, RB Port change and External RB0/INT pin interrupts. Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software

should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 2-3: INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)



R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit 7							bit 0

bit 7 GIE: Global Interrupt Enable bit

1 = Enables all unmasked interrupts

0 = Disables all interrupts

bit 6 PEIE: Peripheral Interrupt Enable bit

1 = Enables all unmasked peripheral interrupts

0 = Disables all peripheral interrupts

bit 5 T0IE: TMR0 Overflow Interrupt Enable bit

1 = Enables the TMR0 interrupt

0 = Disables the TMR0 interrupt

bit 4 INTE: RB0/INT External Interrupt Enable bit

1 = Enables the RB0/INT external interrupt

0 = Disables the RB0/INT external interrupt

bit 3 RBIE: RB Port Change Interrupt Enable bit

1 = Enables the RB port change interrupt

0 = Disables the RB port change interrupt

bit 2 T0IF: TMR0 Overflow Interrupt Flag bit

1 = TMR0 register has overflowed (must be cleared in software)

0 = TMR0 register did not overflow

bit 1 INTF: RB0/INT External Interrupt Flag bit

1 = The RB0/INT external interrupt occurred (must be cleared in software)

0 = The RB0/INT external interrupt did not occur

bit 0 RBIF: RB Port Change Interrupt Flag bit

1 = At least one of the RB7:RB4 pins changed state; a mismatch condition will continue to set

the bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared (must be cleared in software).

0 = None of the RB7:RB4 pins have changed state

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'

- n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

Indirect Addressing, INDF and FSR Registers

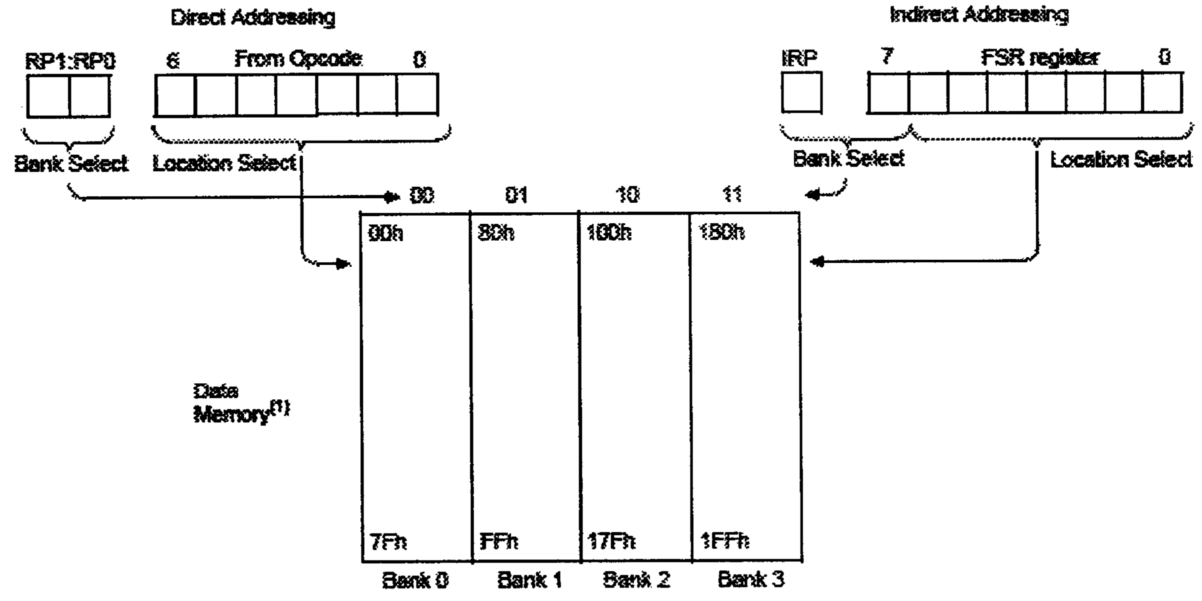
The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing. Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF register itself, indirectly (FSR = ’0’) will read 00h. Writing to the INDF register indirectly results in a no operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 2-6.

A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 2-2.

EXAMPLE 2-2: INDIRECT ADDRESSING

```
MOVLW 0x20 ;initialize pointer
MOVWF FSR ;to RAM
NEXT CLRF INDF ;clear INDF register
INCF FSR,F ;inc pointer
BTFSS FSR,4 ;all done?
GOTO NEXT ;no clear next
CONTINUE: ;yes continue
```

FIGURE 2-6: DIRECT/INDIRECT ADDRESSING



Note 1: For register file map detail, see Figure 2-3.

DATA EEPROM AND FLASH PROGRAM MEMORY

The Data EEPROM and FLASH Program Memory are readable and writable during normal operation over the entire VDD range. These operations take place on a single byte for Data EEPROM memory and a single word for Program memory. A write operation causes an erase-then-write operation to take place on the specified byte or word. A bulk erase operation may not be issued from user code (which includes removing code protection).

Access to program memory allows for checksum calculation. The values written to program memory do not need to be valid instructions. Therefore, up to 14-bit numbers can be stored in memory for use as calibration

parameters, serial numbers, packed 7-bit ASCII, etc. Executing a program memory location containing data that form an invalid instruction, results in the execution of a NOP instruction.

The EEPROM Data memory is rated for high erase/ write cycles (specification D120). The FLASH program memory is rated much lower (specification D130), because EEPROM data memory can be used to store

frequently updated values. An on-chip timer controls the write time and it will vary with voltage and temperature, as well as from chip to chip. Please refer to the specifications for exact limits (specifications D122 and D133).

A byte or word write automatically erases the location and writes the new value (erase before write). Writing to EEPROM data memory does not impact the operation of the device. Writing to program memory will

cease the execution of instructions until the write is complete. The program memory cannot be accessed during the write. During the write operation, the oscillator continues to run, the peripherals continue to function

and interrupt events will be detected and essentially "queued" until the write is complete. When the write completes, the next instruction in the pipeline is executed and the branch to the interrupt vector will take place, if the interrupt is enabled and occurred during the write. Read and write access to both memories take place indirectly through a set of Special Function Registers (SFR).

The six SFRs used are:

- EEDATA
- EEDATH
- EEADR
- EEADRH
- EECON1
- EECON2

The EEPROM data memory allows byte read and write operations without interfering with the normal operation of the microcontroller. When interfacing to EEPROM data memory, the EEADR register holds the address to be accessed. Depending on the operation, the EEDATA register holds the data to be written, or the data read, at the address in EEADR. The PIC16F873/874 devices have 128 bytes of EEPROM data memory and therefore, require that the MSb of EEADR remain clear. The EEPROM data memory on these devices do not wrap around to 0, i.e., 0x80 in the EEADR does not map to 0x00. The PIC16F876/877 devices have 256 bytes of EEPROM data memory and therefore, uses all 8-bits of the EEADR.

The FLASH program memory allows non-intrusive read access, but write operations cause the device to stop executing instructions, until the write completes. When interfacing to the program memory, the EEADRH:EEADR registers form a two-byte word, which holds the 13-bit address of the memory location being accessed. The register combination of EEDATH:EEDATA holds the 14-bit data for writes, or reflects the value of program memory after a read operation. Just as in EEPROM data memory accesses, the value of the EEADRH:EEADR registers must be within the valid range of program memory, depending on the device: 0000h to 1FFFh for the PIC16F873/874, or 0000h to 3FFFh for the PIC16F876/877. Addresses outside of this range do not wrap around to 0000h (i.e., 4000h does not map to 0000h on the PIC16F877).

EECON1 and EECON2 Registers

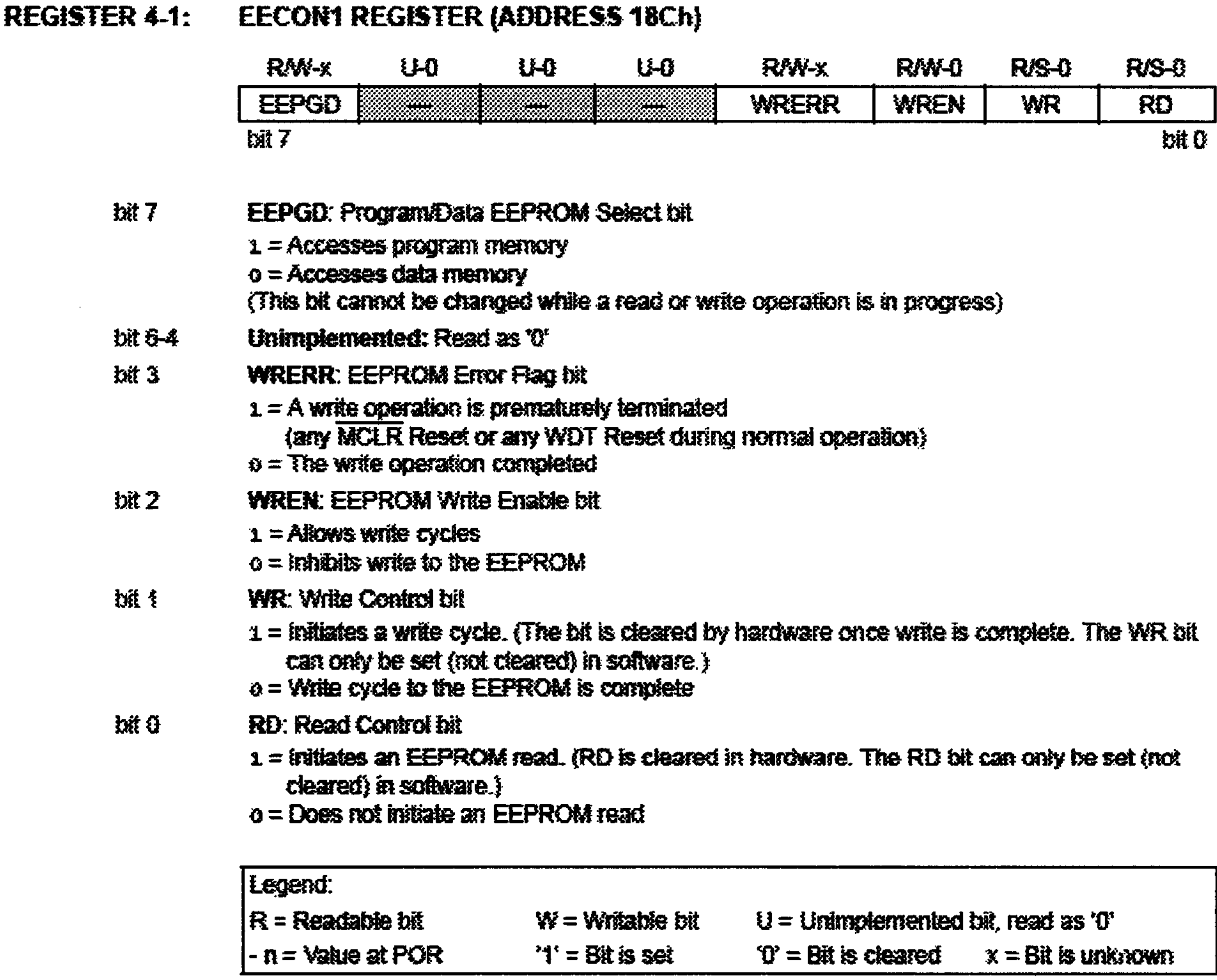
The EECON1 register is the control register for configuring and initiating the access. The EECON2 register is not a physically implemented register, but is used exclusively in the memory write sequence to prevent inadvertent writes. There are many bits used to control the read and write operations to EEPROM data and FLASH program memory. The EEPGD bit determines if the access will be a program or data memory access. When clear, any subsequent operations will work on the EEPROM data memory. When set, all subsequent operations will operate in the program memory. Read operations only use one additional bit, RD, which initiates the read operation from the desired memory location. Once this bit is set, the value of the desired memory location will be available in the data registers. This bit cannot be cleared by firmware. It is automatically cleared at the end of the read operation. For EEPROM data memory reads, the data will be available in the EEDATA register in the very next instruction cycle after the RD bit is set. For program memory reads, the data will be loaded into the EEDATH:EEDATA registers, following the second instruction after the RD bit is set.

Write operations have two control bits, WR and WREN, and two status bits, WRERR and EEIF. The WREN bit is used to enable or disable the write operation. When WREN is clear, the write operation will be disabled. Therefore, the WREN bit must be set before executing a write operation. The WR bit is used to initiate the write operation. It also is automatically cleared at the end of the write operation. The interrupt flag EEIF is used to determine when the memory write completes. This flag must be cleared in software before setting the WR bit.

For EEPROM data memory, once the WREN bit and the WR bit have been set, the desired memory address in EEADR will be erased, followed by a write of the data in EEDATA. This operation takes place in parallel with the microcontroller continuing to execute normally. When the write is complete, the EEIF flag bit will be set. For program memory, once the WREN bit and the WR bit have been set, the microcontroller will cease to execute instructions. The desired memory location pointed to by EEADRH:EEADR will be erased. Then, the data value in

EDATH:EEDATA will be programmed. When complete, the EEIF flag bit will be set and the microcontroller will continue to execute code.

The WRERR bit is used to indicate when the PIC16F87X device has been reset during a write operation. WRERR should be cleared after Power-on Reset. Thereafter, it should be checked on any other RESET. The WRERR bit is set when a write operation is interrupted by a MCLR Reset, or a WDT Time-out Reset, during normal operation. In these situations, following a RESET, the user should check the WRERR bit and rewrite the memory location, if set. The contents of the data registers, address registers and EEPGD bit are not affected by either MCLR Reset, or WDT Timeout Reset, during normal operation.



REGISTER 4-1: EECON1 REGISTER (ADDRESS 18Ch)

R/W-x	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	—	—	—	WRERR	WREN	WR	RD
bit 7							bit 0

bit 7 **EEPGD**: Program/Data EEPROM Select bit
1 = Accesses program memory
0 = Accesses data memory
(This bit cannot be changed while a read or write operation is in progress)

bit 6-4 **Unimplemented:** Read as '0'

bit 3 **WRERR**: EEPROM Error Flag bit
1 = A write operation is prematurely terminated
(any $\overline{\text{MCLR}}$ Reset or any WDT Reset during normal operation)
0 = The write operation completed

bit 2 **WREN**: EEPROM Write Enable bit
1 = Allows write cycles
0 = Inhibits write to the EEPROM

bit 1 **WR**: Write Control bit
1 = Initiates a write cycle. (The bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.)
0 = Write cycle to the EEPROM is complete

bit 0 **RD**: Read Control bit
1 = Initiates an EEPROM read. (RD is cleared in hardware. The RD bit can only be set (not cleared) in software.)
0 = Does not initiate an EEPROM read

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

READING THE EEPROM DATA MEMORY

Reading EEPROM data memory only requires that the desired address to access be written to the EEADR register and clear the EEPGD bit. After the RD bit is set, data will be available in the EEDATA register on the very next instruction cycle. EEDATA will hold this value until another read operation is initiated or until it is written by firmware.

The steps to reading the EEPROM data memory are:

1. Write the address to EEDATA. Make sure that the address is not larger than the memory size of the PIC16F87X device.
2. Clear the EEPGD bit to point to EEPROM data memory.
3. Set the RD bit to start the read operation.
4. Read the data from the EEDATA register.

EXAMPLE 4-1: EEPROM DATA READ

```
BSF STATUS, RP1 ;
BCF STATUS, RP0 ;Bank 2
MOVF ADDR, W ;Write address
MOVWF EEADR ;to read from
BSF STATUS, RP0 ;Bank 3
BCF EECON1, EEPGD ;Point to Data memory
BSF EECON1, RD ;Start read operation
BCF STATUS, RP0 ;Bank 2
MOVF EEDATA, W ;W = EEDATA
```

WRITING TO THE EEPROM DATA MEMORY

There are many steps in writing to the EEPROM data memory. Both address and data values must be written to the SFRs. The EEPGD bit must be cleared, and the WREN bit must be set, to enable writes. The WREN bit should be kept clear at all times, except when writing to the EEPROM data. The WR bit can only be set if the WREN bit was set in a previous operation, i.e., they both cannot be set in the same operation. The WREN bit should then be cleared by firmware after the write. Clearing the WREN bit before the write actually completes will not terminate the write in progress. Writes to EEPROM data memory must also be prefaced with a special sequence of instructions, that prevent inadvertent write operations. This is a sequence of five instructions that must be executed without interruptions. The firmware should verify that a write is not in progress, before starting another cycle. The steps to write to EEPROM data memory are:

1. If step 10 is not implemented, check the WR bit to see if a write is in progress.
2. Write the address to EEADR. Make sure that the address is not larger than the memory size of the PIC16F87X device.
3. Write the 8-bit data value to be programmed in the EEDATA register.
4. Clear the EEPGD bit to point to EEPROM data memory.
5. Set the WREN bit to enable program operations.
6. Disable interrupts (if enabled).
7. Execute the special five instruction sequence:
    - Write 55h to EECON2 in two steps (first to W, then to EECON2)
    - Write AAh to EECON2 in two steps (first to W, then to EECON2)
    - Set the WR bit
8. Enable interrupts (if using interrupts).
9. Clear the WREN bit to disable program operations.
10. At the completion of the write cycle, the WR bit is cleared and the EEIF interrupt flag bit is set. (EEIF must be cleared by firmware.) If step 1 is not implemented, then firmware should check for EEIF to be set, or WR to clear, to indicate the end of the program cycle.

EXAMPLE 4-2: EEPROM DATA WRITE

```
BSF STATUS, RP1 ;
BSF STATUS, RP0 ;Bank 3
BTFSC EECON1, WR ;Wait for
GOTO $-1 ;write to finish
BCF STATUS, RP0 ;Bank 2
```

```
MOVF ADDR, W ;Address to
MOVWF EEADR ;write to
MOVF VALUE, W ;Data to
MOVWF EEDATA ;write
BSF STATUS, RP0 ;Bank 3
BCF EECON1, EEPGD ;Point to Data memory
BSF EECON1, WREN ;Enable writes
;Only disable interrupts
BCF INTCON, GIE ;if already enabled,
;otherwise discard
MOVLW 0x55 ;Write 55h to
MOVWF EECON2 ;EECON2
MOVLW 0xAA ;Write AAh to
MOVWF EECON2 ;EECON2
BSF EECON1, WR ;Start write operation
;Only enable interrupts
BSF INTCON, GIE ;if using interrupts,
;otherwise discard
BCF EECON1, WREN ;Disable writes
```

TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Timer mode is selected by clearing bit TOCS (OPTION_REG<5>). In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If the TMR0 register is written, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register. Counter mode is selected by setting bit T0CS (OPTION_REG<5>). In Counter mode, Timer0 will increment either on every rising, or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit, T0SE (OPTION_REG<4>). Clearing bit T0SE selects the rising edge.

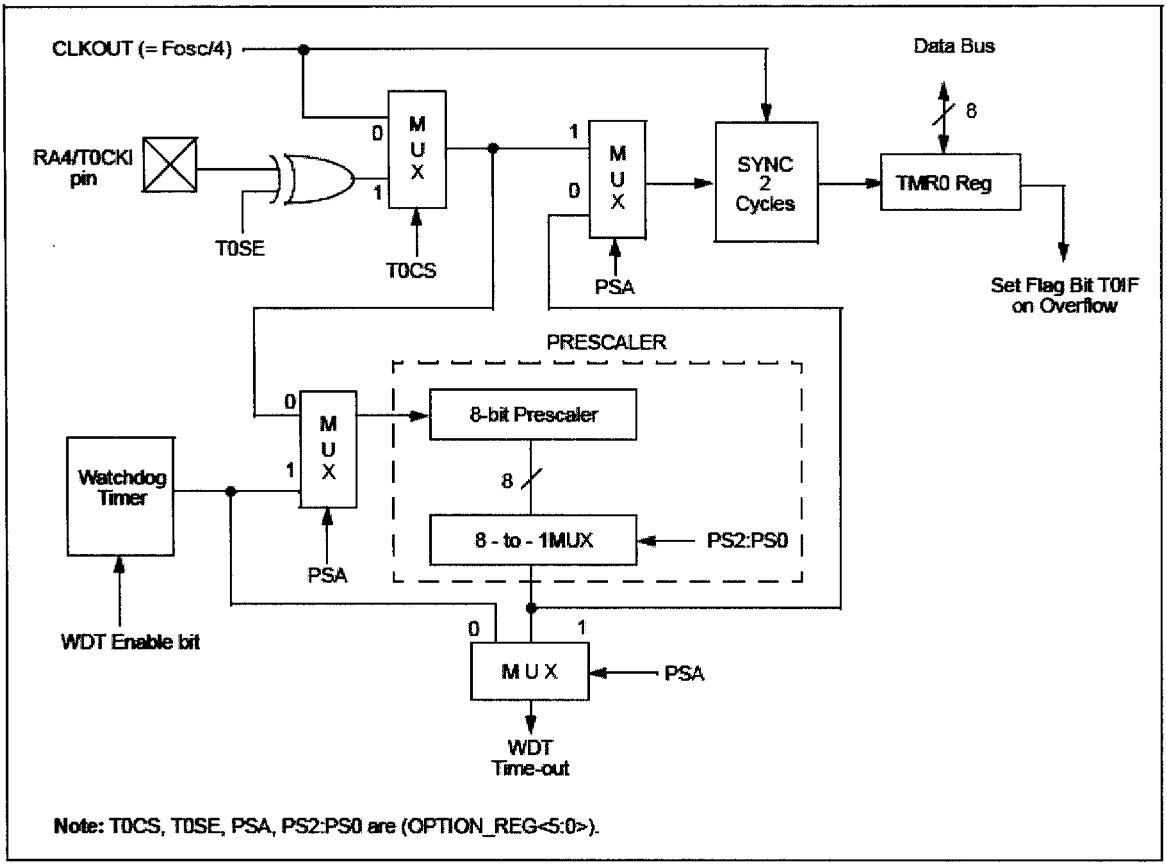
The prescaler is mutually exclusively shared between the Timer0 module and the Watchdog Timer. The prescaler is not readable or writable.

Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets bit T0IF (INTCON<2>). The interrupt can be masked by clearing bit T0IE (INTCON<5>). Bit T0IF must be cleared in software by the Timer0 module Interrupt Service Routine before re-enabling this interrupt. The TMR0 interrupt cannot awaken the processor from

SLEEP, since the timer is shut-off during SLEEP.

FIGURE 5-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER



Using Timer0 with an External Clock

When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and

Q4 cycles of the internal phase clocks. Therefore, it is necessary for T0CKI to be high for at least 2Tosc (and a small RC delay of 20 ns) and low for at least 2Tosc (and a small RC delay of 20 ns). Refer to the electrical

specification of the desired device.

PRESCALER

There is only one prescaler available, which is mutually exclusively shared between the Timer0 module and the Watchdog Timer. A prescaler assignment for the Timer0 module means that there is no prescaler for the

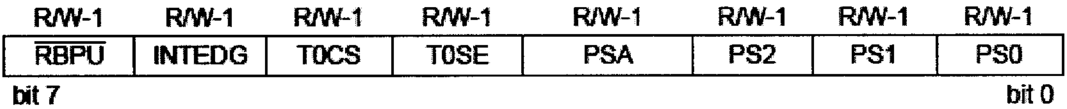
Watchdog Timer, and vice-versa. This prescaler is not readable or writable (see Figure 5-1). The PSA and PS2:PS0 bits (OPTION_REG<3:0>)

determine the prescaler assignment and prescale ratio. When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g. CLRF 1, MOVWF 1,BSF 1,x....etc.) will clear the prescaler. When assigned

to WDT, a CLRWDT instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable or writable.

Note: Writing to TMR0, when the prescaler is assigned to Timer0, will clear the prescaler count, but will not change the prescaler assignment.

REGISTER 5-1: OPTION_REG REGISTER



R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

bit 7 bit 0

bit 7 RBPU

bit 6 INTEDG

bit 5 T0CS: TMR0 Clock Source Select bit

1 = Transition on T0CKI pin

0 = Internal instruction cycle clock (CLKOUT)

bit 4 T0SE: TMR0 Source Edge Select bit

1 = Increment on high-to-low transition on T0CKI pin

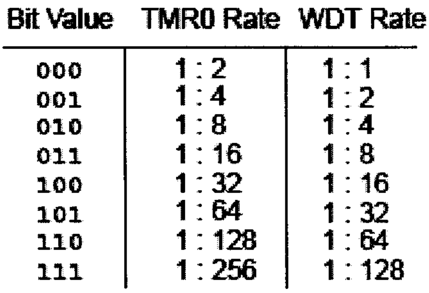
0 = Increment on low-to-high transition on T0CKI pin

bit 3 PSA: Prescaler Assignment bit

1 = Prescaler is assigned to the WDT

0 = Prescaler is assigned to the Timer0 module

bit 2-0 PS2:PS0: Prescaler Rate Select bits



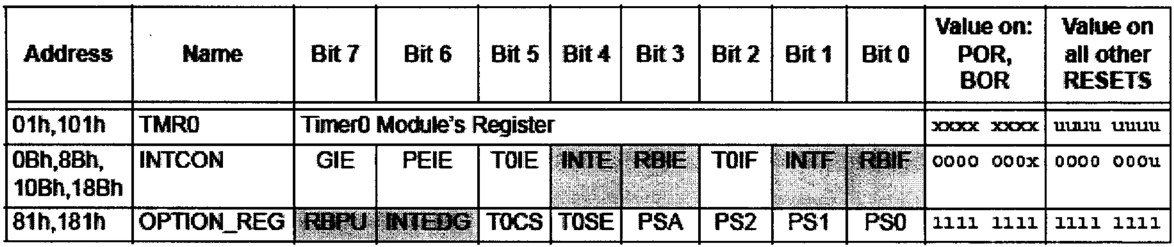
Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as ‘0’

- n = Value at POR ’1’ = Bit is set ’0’ = Bit is cleared x = Bit is unknown

TABLE 5-1: REGISTERS ASSOCIATED WITH TIMER0



Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
01h,101h	TMR0	Timer0 Module’s Register								xxxx xxxx	uuuu uuuu
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
81h,181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'.
Shaded cells are not used by Timer0.

INT INTERRUPT

External interrupt on the RB0/INT pin is edge triggered, either rising, if bit INTEDG (OPTION_REG<6>) is set, or falling, if the INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, flag bit INTF (INTCON<1>) is set. This interrupt can be disabled by clearing enable bit INTE (INTCON<4>). Flag bit INTF must be cleared in software in the Interrupt Service Routine before re-enabling this interrupt. The INT interrupt

can wake-up the processor from SLEEP, if bit INTE was set prior to going into SLEEP. The status of global interrupt enable bit, GIE, decides whether or not the processor branches to the interrupt vector following wake-up.

TMR0 INTERRUPT

An overflow (FFh 00h) in the TMR0 register will set flag bit T0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit T0IE (INTCON<5.

PORTB INTCON CHANGE

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit RBIE (INTCON<4>).

Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt, (i.e., W register and STATUS register). This will have to be implemented in software. For the PIC16F873/874 devices, the register W_TEMP must be defined in both banks 0 and 1 and must be defined at the same offset from the bank base address (i.e., If W_TEMP is defined at 0x20 in bank 0, it must also be defined at 0xA0 in bank 1). The registers, PCLATH_TEMP and STATUS_TEMP, are only defined in bank 0. Since the upper 16 bytes of each bank are common in the PIC16F876/877 devices, temporary holding registers W_TEMP, STATUS_TEMP, and PCLATH_TEMP should be placed in here. These 16 locations don't require banking and therefore, make it easier for context save and restore.

EXAMPLE 12-1: SAVING STATUS, W, AND PCLATH REGISTERS IN RAM

```
MOVWF W_TEMP ;Copy W to TEMP register
SWAPF STATUS,W ;Swap status to be saved into W
CLRF STATUS ;bank 0, regardless of current bank, Clears IRP,RP1,RP0
MOVWF STATUS_TEMP ;Save status to bank zero STATUS_TEMP register
MOVF PCLATH, W ;Only required if using pages 1, 2 and/or 3
MOVWF PCLATH_TEMP ;Save PCLATH into W
CLRF PCLATH ;Page zero, regardless of current page
:
:(ISR) ;(Insert user code here)
:
MOVF PCLATH_TEMP, W ;Restore PCLATH
```

```
MOVWF PCLATH ;Move W into PCLATH
SWAPF STATUS_TEMP,W ;Swap STATUS_TEMP register into W
;(sets bank to original state)
MOVWF STATUS ;Move W into STATUS register
SWAPF W_TEMP,F ;Swap W_TEMP
SWAPF W_TEMP,W ;Swap W_TEMP into W
```

INSTRUCTION SET SUMMARY

Each PIC16F87X instruction is a 14-bit word, divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16F87X instruction set summary in Table 13-2 lists byte-oriented, bit-oriented, and literal and control operations. Table 13-1 shows the opcode field descriptions.

For byte-oriented instructions, ’f’ represents a file register designator and ’d’ represents a destination designator. The file register designator specifies which file register is to be used by the instruction.
The destination designator specifies where the result of the operation is to be placed. If ’d’ is zero, the result is placed in the W register. If ’d’ is one, the result is placed in the file register specified in the instruction.

For bit-oriented instructions, ’b’ represents a bit field designator which selects the number of the bit affected by the operation, while ’f’ represents the address of the file in which the bit is located.

For literal and control operations, ’k’ represents an eight or eleven bit constant or literal value.

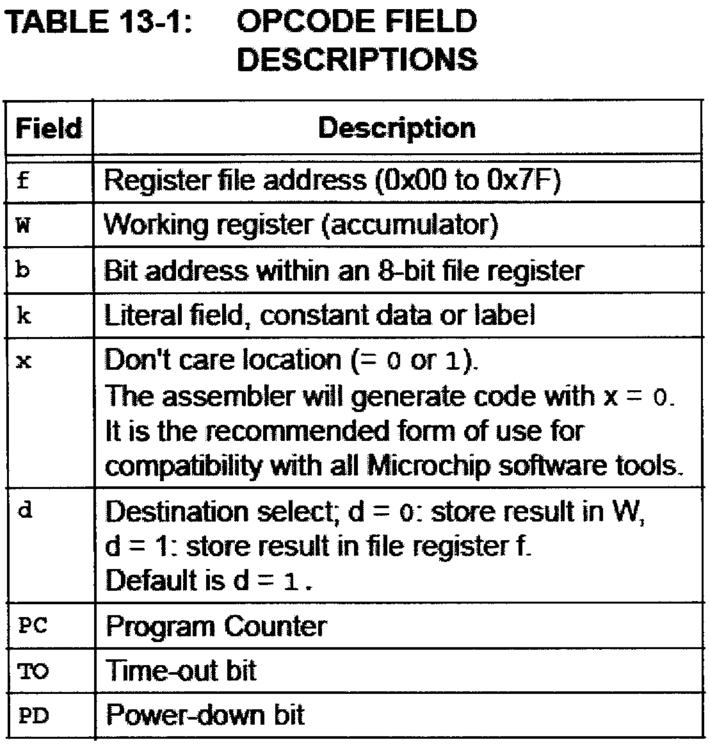


TABLE 13-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
TO	Time-out bit
PD	Power-down bit

The instruction set is grouped into three basic categories:

- Byte-oriented operations
- Bit-oriented operations
- Literal and control operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 s. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 s. All examples use the following format to represent a hexadecimal number:0xhh where h signifies a hexadecimal digit. Note: To maintain upward compatibility with future PIC16F87X products, do not use the OPTION and TRIS instructions.

FIGURE 13-1: GENERAL FORMAT FOR INSTRUCTIONS

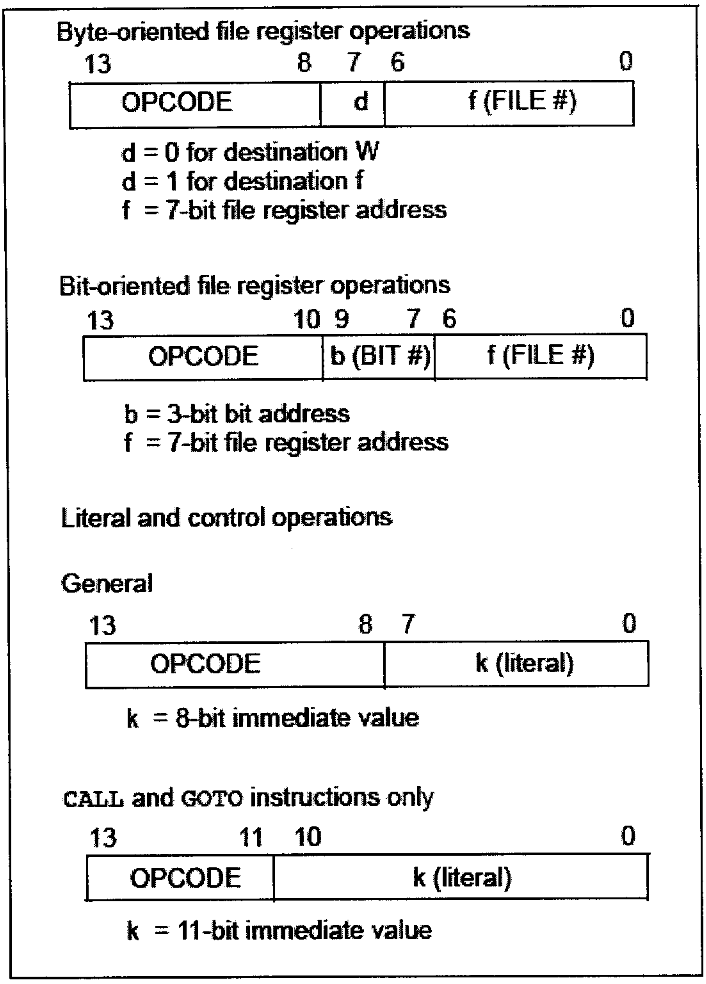

Byte-oriented file register operations
13
8
7
6
0
OPCODE
d
f (FILE #)
d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address
Bit-oriented file register operations
13
10
9
7
6
0
OPCODE
b (BIT #)
f (FILE #)
b = 3-bit bit address
f = 7-bit file register address
Literal and control operations
General
13
8
7
0
OPCODE
k (literal)
k = 8-bit immediate value
CALL and GOTO instructions only
13
11
10
0
OPCODE
k (literal)
k = 11-bit immediate value

TABLE 13-2: PIC16F87X INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb			LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{\text{TO}}$,$\overline{\text{PD}}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{\text{TO}}$,$\overline{\text{PD}}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.

3: If Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

Note: Additional information on the mid-range instruction set is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

DEVELOPMENT SUPPORT

The PICmicro® microcontrollers are supported with a full range of hardware and software development tools:

• Integrated Development Environment

- MPLAB® IDE Software

• Assemblers/Compilers/Linkers

- MPASMTM Assembler

- MPLAB C17 and MPLAB C18 C Compilers

- MPLINKTM Object Linker/ MPLIBTM Object Librarian

• Simulators

- MPLAB SIM Software Simulator

MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8-bit microcontroller market. The MPLAB IDE is a Windows®-based application that contains:

• An interface to debugging tools

- simulator

- programmer (sold separately)

- emulator (sold separately)

- in-circuit debugger (sold separately)

• A full-featured editor

• A project manager

• Customizable toolbar and key mapping

• A status bar

• On-line help

The MPLAB IDE allows you to:

• Edit your source files (either assembly or 'C')

• One touch assemble (or compile) and download to PICmicro emulator and simulator tools (automatically updates all project information)

• Debug using:

- source files

- absolute listing file
- machine code

The ability to use MPLAB IDE with multiple debugging tools allows users to easily switch from the cost effective simulator to a full-featured emulator with minimal retraining.

MPASM ASSEMBLER

The MPASM assembler is a full-featured universal macro assembler for all PICmicro MCU's. The MPASM assembler has a command line interface and a Windows shell. It can be used as a stand-alone application on a Windows 3.x or greater system, or it can be used through MPLAB IDE. The MPASM assembler generates relocatable object files for the MPLINK
object linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, an absolute LST file that contains source lines and generated machine code, and a COD file for debugging.

The MPASM assembler features include:

- Integration into MPLAB IDE projects.
- User-defined macros to streamline assembly code.
- Conditional assembly for multi-purpose source files.
- Directives that allow complete control over the assembly process.

MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC-hosted environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be
examined or modified and stimuli can be applied from a file, or user-defined key press, to any of the pins. The execution can be performed in single step, execute until break, or trace mode.

The MPLAB SIM simulator fully supports symbolic debugging
using the MPLAB C17 and the MPLAB C18 C compilers and the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent multiproject software development tool.

MT8880 DTMF Transceiver

The MT8880C is a monolithic DTMF transceiver with call progress filter. It is fabricated in ZarlinkZarlinkZarlink's ISO 2 -CMOS technology, which provides low power dissipation and high reliability. The DTMF receiver is based upon the industry standard MT8870 monolithic DTMF receiver; the transmitter utilizes a switched capacitor D/A converter for low distortion, high accuracy DTMF signalling. Internal counters provide a burst mode such that tone bursts can be transmitted with precise timing. A call progress filter can be selected allowing a microprocessor to analyze call progress tones. A standard microprocessor bus is provided and is
directly compatible with 6800 series microprocessors.

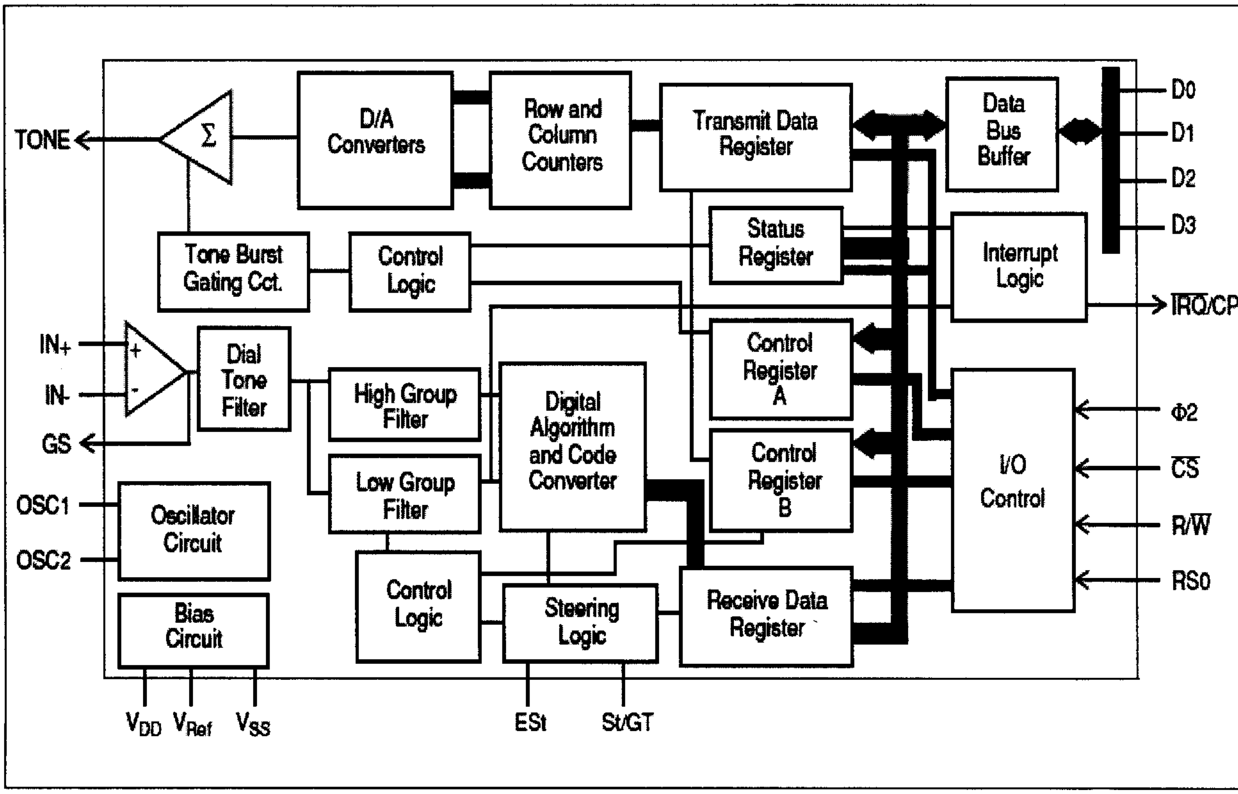


Figure 1 - Functional Block Diagram

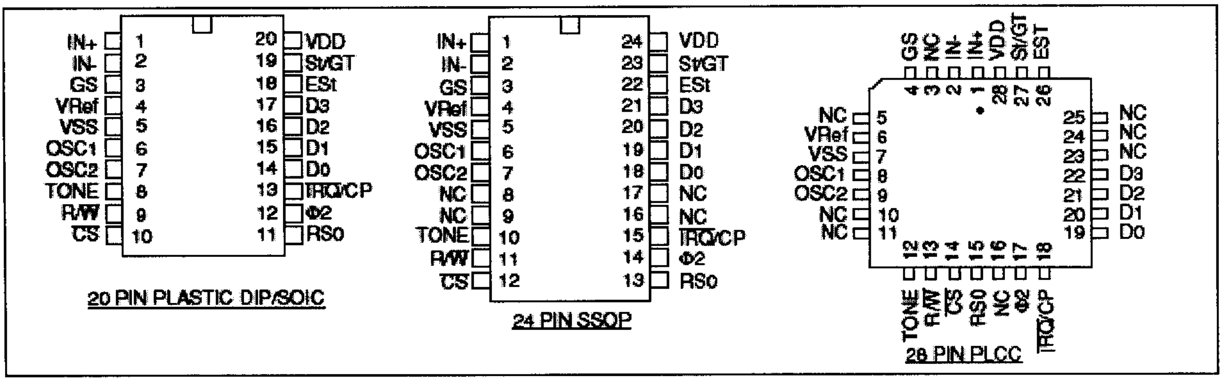


Figure 2 - Pin Connections

Pin Description

Pin #			Name	Description
20	24	28		
1	1	1	IN+	Non-inverting op-amp input.
2	2	2	IN-	Inverting op-amp input.
3	3	4	GS	**Gain Select.** Gives access to output of front end differential amplifier for connection of feedback resistor.
4	4	6	V_{Ref}	**Reference Voltage** output, nominally $V_{DD}/2$ is used to bias inputs at mid-rail (see Fig. 13).
5	5	7	V_{SS}	Ground input (0V).
6	6	8	OSC1	DTMF clock/oscillator input. Connect a 4.7MΩ resistor to VSS if crystal oscillator is used.
7	7	9	OSC2	Clock output. A 3.579545 MHz crystal connected between OSC1 and OSC2 completes the internal oscillator circuit. Leave open circuit when OSC1 is clock input.
8	10	12	TONE	Tone output (DTMF or single tone).
9	11	13	R/$\overline{W}$	**Read/Write** input. Controls the direction of data transfer to and from the MPU and the transceiver registers. TTL compatible.
10	12	14	$\overline{CS}$	**Chip Select**, TTL input ($\overline{CS}$=0 to select the chip).
11	13	15	RS0	**Register Select** input. See register decode table. TTL compatible.
12	14	17	Φ2	**System Clock** input. TTL compatible. **N.B. Φ2 clock input need not be active when the device is not being accessed.**
13	15	18	$\overline{IRQ}$/ CP	**Interrupt Request to MPU** (open drain output). Also, when call progress (CP) mode has been selected and interrupt enabled the $\overline{IRQ}$/CP pin will output a rectangular wave signal representative of the input signal applied at the input op-amp. The input signal must be within the bandwidth limits of the call progress filter. See Figure 8.
14-17	18-21	19-22	D0-D3	Microprocessor Data Bus (TTL compatible). High impedance when $\overline{CS}$ = 1 or Φ2 is low.
18	22	26	ESt	**Early Steering** output. Presents a logic high once the digital algorithm has detected a valid tone pair (signal condition). Any momentary loss of signal condition will cause ESt to return to a logic low.
19	23	27	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; its state is a function of ESt and the voltage on St.
20	24	28	V_{DD}	**Positive power supply** input (+5V typical).
	8, 9, 16,17	3,5,10, 11, 16, 23-25	NC	No Connection.

FUNCTIONAL DESCRIPTION

The MT8880C Integrated DTMF Transceiver architecture consists of a high performance DTMF receiver with internal gain setting amplifier and a DTMF generator which employs a burst counter such that precise tone bursts and pauses can be synthesized. A call progress mode can be selected such that frequencies within the specified passband can be detected. A standard microprocessor interface allows access to an internal status register, two control registers and two data registers.

INPUT CONFIGURATION

The input arrangement of the MT8880C provides a differential-input operational amplifier as well as a bias source (V_{Ref}) which is used to bias the inputs at $V_{DD}/2$. Provision is made for connection of a feedback resistor to the op-amp output (GS) for adjustment of gain. In a single-ended configuration, the input pins are connected as shown in Figure 3.

Figure 4 shows the necessary connections for a differential input configuration.

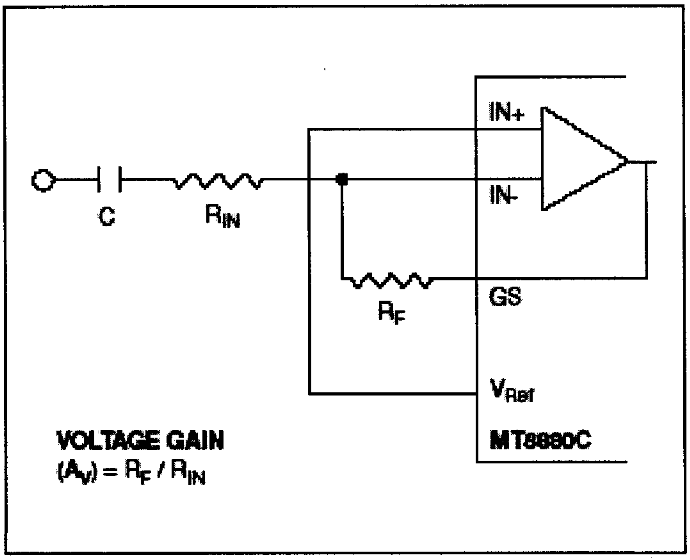
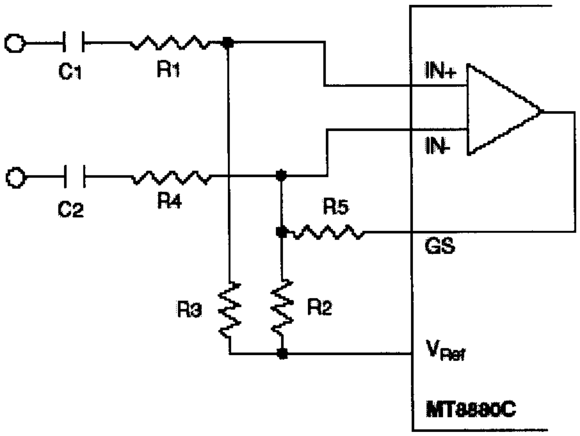


Figure 3 - Single-Ended Input Configuration



DIFFERENTIAL INPUT AMPLIFIER
C1 = C2 = 10 nF
R1 = R4 = R5 = 100 kΩ
R2 = 60kΩ, R3 = 37.5 kΩ
R3 = (R2R5)/(R2 + R5)

VOLTAGE GAIN
$(A_V \text{ diff}) = R5/R1$

INPUT IMPEDANCE
$(Z_{IN}\text{diff}) = 2\sqrt{R1^2 + (1/\omega C)^2}$

Figure 4 - Differential Input Configuration

RECEIVER SECTION

Separation of the low and high group tones is achieved by applying the DTMF signal to the inputs of two sixth-order switched capacitor bandpass filters, the bandwidths of which correspond to the low and high group frequencies (see Fig. 7). These filters also incorporate notches at 350 Hz and 440 Hz for exceptional dial tone rejection. Each filter output is

followed by a single order switched capacitor filter section which smooths the signals prior to limiting. Limiting is performed by high-gain comparators which are provided with hysteresis to prevent detection of unwanted low-level signals. The outputs of the comparators provide full rail logic swings at the frequencies of the incoming DTMF signals.

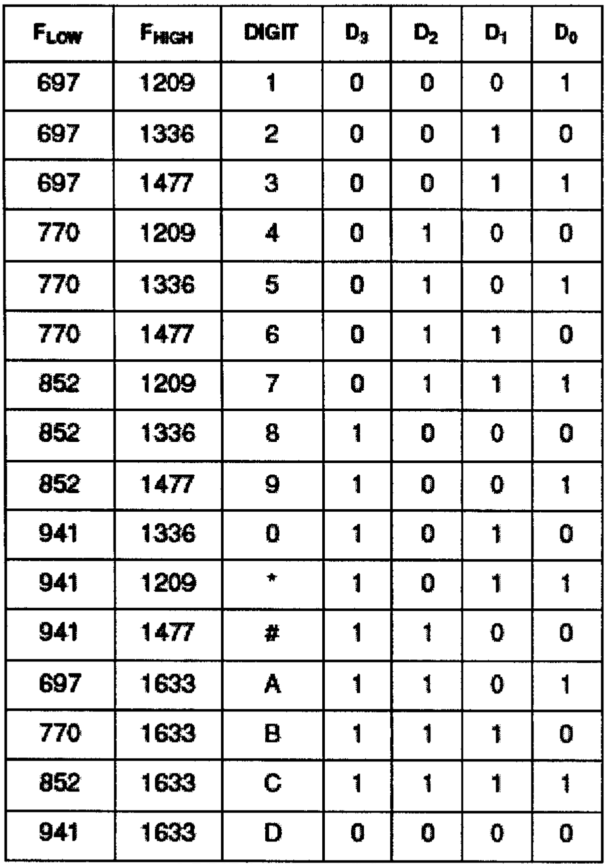
Following the filter section is a decoder employing digital counting techniques to determine the frequencies of the incoming tones and to verify that they correspond to standard DTMF frequencies.

A complex averaging algorithm protects against tone simulation by extraneous signals such as voice while providing tolerance to small frequency deviations and variations. This averaging algorithm has been

developed to ensure an optimum combination of immunity to talk-off and tolerance to the presence of interfering frequencies (third tones) and noise. When the detector recognizes the presence of two valid tones (this is referred to as the "signal condition" in some industry specifications) the "Early Steering" (ESt) output will go to an active state. Any subsequent loss of signal condition will cause ESt to assume an inactive state.

DTMF GENERATOR

The DTMF transmitter employed in the MT8880C is capable of generating all sixteen standard DTMF tone pairs with low distortion and high accuracy. All frequencies are derived from an external 3.579545 MHz crystal. The sinusoidal waveforms for the individual tones are digitally synthesized using row and column programmable dividers and switched capacitor D/A converters. The row and column tones are mixed and filtered providing a DTMF signal with low total harmonic distortion and high accuracy. To specify a DTMF signal, data conforming to the encoding format shown in Figure 7 must be written to the transmit Data Register. Note that this is the same as the receiver output code. The individual tones which are generated (f_{LOW} and f_{HIGH}) are referred to as Low Group and High Group tones. As seen from the table, the low group frequencies are 697, 770, 852 and 941 Hz. The high group frequencies are 1209, 1336, 1477 and 1633 Hz. Typically, the high group to low group amplitude ratio (pre-emphasis) is 2dB to compensate for high group attenuation on long loops.



F_{LOW}	F_{HIGH}	DIGIT	D_3	D_2	D_1	D_0
697	1209	1	0	0	0	1
697	1336	2	0	0	1	0
697	1477	3	0	0	1	1
770	1209	4	0	1	0	0
770	1336	5	0	1	0	1
770	1477	6	0	1	1	0
852	1209	7	0	1	1	1
852	1336	8	1	0	0	0
852	1477	9	1	0	0	1
941	1336	0	1	0	1	0
941	1209	*	1	0	1	1
941	1477	#	1	1	0	0
697	1633	A	1	1	0	1
770	1633	B	1	1	1	0
852	1633	C	1	1	1	1
941	1633	D	0	0	0	0

0= LOGIC LOW, 1= LOGIC HIGH

Figure 7 - Functional Encode/Decode Table

BURST MODE

In certain telephony applications it is required that DTMF signals being generated are of a specific duration determined either by the particular application or by any one of the exchange transmitter specifications currently existing. Standard DTMF signal timing can be accomplished by making use of the Burst Mode. The transmitter is capable of issuing symmetric bursts/pauses of predetermined duration. This burst/pause duration is 51 ms±1 ms which is a standard interval for autodialer and central office applications. After the burst/pause has been issued, the appropriate bit is set in the Status Register indicating that the transmitter is ready for more data. The timing described above is available when DTMF mode has been selected. However, when CP mode (Call Progress mode) is selected, a second burst/ pause time of 102 ms ±2 ms is available. This extended interval is useful when precise tone bursts of longer than 51 ms duration and 51 ms pause are desired. Note that when CP mode and Burst mode have been selected, DTMF tones may be transmitted only and *not* received. In applications where a non-standard burst/pause duration is required, burst mode must be disabled and the transmitter gated on and off by an external hardware or software timer.

RS0	R/$\overline{\text{W}}$	FUNCTION
0	0	Write to Transmit Data Register
0	1	Read from Receive Data Register
1	0	Write to Control Register
1	1	Read from Status Register

Table 2. Internal Register Functions

b3	b2	b1	b0
RSEL	IRQ	CP/$\overline{\text{DTMF}}$	TOUT

Table 3. CRA Bit Positions

b3	b2	b1	b0
C/R	S/$\overline{\text{D}}$	TEST	BURST

Table 4. CRB Bit Positions

BIT	NAME	FUNCTION	DESCRIPTION
b0	TOUT	TONE OUTPUT	A logic '1' enables the tone output. This function can be implemented in either the burst mode or non-burst mode.
b1	CP/$\overline{\text{DTMF}}$	MODE CONTROL	In DTMF mode (logic '0') the device is capable of generating and receiving Dual Tone Multi-Frequency signals. When the CP (Call Progress) mode is selected (logic '1') a 6th order bandpass filter is enabled to allow call progress tones to be detected. Call progress tones which are within the specified bandwidth will be presented at the $\overline{\text{IRQ}}$/CP pin in rectangular wave format if the IRQ bit has been enabled (b2=1). Also, when the CP mode and BURST mode have both been selected, the transmitter will issue DTMF signals with a burst and pause of 102 ms (typ) duration. This signal duration is twice that obtained from the DTMF transmitter if DTMF mode had been selected. Note that DTMF signals cannot be decoded when the CP mode of operation has been selected.
b2	IRQ	INTERRUPT ENABLE	A logic '1' enables the INTERRUPT mode. When this mode is active and the DTMF mode has been selected (b1=0) the $\overline{\text{IRQ}}$/ CP pin will pull to a logic '0' condition when either 1) a valid DTMF signal has been received and has been present for the guard time duration or 2) the transmitter is ready for more data (BURST mode only).
b3	RSEL	REGISTER SELECT	A logic '1' selects Control Register B on the next Write cycle to the Control Register address. Subsequent Write cycles to the Control Register are directed back to Control Register A.

Table 5. Control Register A Description

BIT	NAME	FUNCTION	DESCRIPTION
b0	BURST	BURST MODE	A logic '0' enables the burst mode. When this mode is selected, data corresponding to the desired DTMF tone pair can be written to the Transmit Register resulting in a tone burst of a specific duration (see AC Characteristics). Subsequently, a pause of the same duration is induced. Immediately following the pause, the Status Register is updated indicating that the Transmit Register is ready for further instructions and an interrupt will be generated if the interrupt mode has been enabled. Additionally, if call progress (CP) mode has been enabled, the burst and pause duration is increased by a factor of two. When the burst mode is not selected (logic '1') tone bursts of any desired duration may be generated.
b1	TEST	TEST MODE	By enabling the test mode (logic '1'), the $\overline{\text{IRQ}}$/CP pin will present the delayed steering (inverted) signal from the DTMF receiver. Refer to Figure 9 (b3 waveform) for details concerning the output waveform. DTMF mode must be selected (CRA b1=0) before test mode can be implemented.
b2	S/$\overline{\text{D}}$	SINGLE /DUAL TONE GENERATION	A logic '0' will allow Dual Tone Multi-Frequency signals to be produced. If single tone generation is enabled (logic '1'), either row or column tones (low group or high group) can be generated depending on the state of b3 in Control Register B.
b3	C/$\overline{\text{R}}$	COLUMN/ROW TONES	When used in conjunction with b2 (above) the transmitter can be made to generate single row or single column frequencies. A logic '0' will select row frequencies and a logic '1' will select column frequencies.

Table 6. Control Register B Description

BIT	NAME	STATUS FLAG SET	STATUS FLAG CLEARED
b0	IRQ	Interrupt has occurred. Bit one (b1) or bit two (b2) is set.	Interrupt is inactive. Cleared after Status Register is read.
b1	TRANSMIT DATA REGISTER EMPTY (BURST MODE ONLY)	Pause duration has terminated and transmitter is ready for new data.	Cleared after Status Register is read or when in non-burst mode.
b2	RECEIVE DATA REGISTER FULL	Valid data is in the Receive Data Register.	Cleared after Status Register is read.
b3	$\overline{\text{DELAYED STEERING}}$	Set upon the valid detection of the absence of a DTMF signal.	Cleared upon the detection of a valid DTMF signal.

Table 7. Status Register Description

Instruction Table

Instruction	Code										Description	Execution time(max) (when fcp or fosc is 250 kHz)
	RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0		
Celar Display	0	0	0	0	0	0	0	0	0	1	Clear entire display and returns cursor to home position (Address 0)	1.64 ms
Return Home	0	0	0	0	0	0	0	0	1	*	Returns the Curspr to the home position (Address 0). Also returns display being shifted to original position DD RAM contents remains unchanged.	1.64 ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Select cursor move direction and specifies shift of display. These operations are performed during data write and read.	40 µS
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Sets ON/OFF of entire display (D) cursor ON/OFF (C) and blink of cursor position character (B)	40 µS
Cursor or Display shift	0	0	0	0	0	1	S/C	R/L	*	*	Mover cursor and shift display without changing DD RAM contents.	40 µS
unction Set	0	0	0	0	1	DL	N	F	*	*	Set interface data length (DL), number of display lines (N) and character font (F)	40 µS
t CG RAM Address	0	0	0	1	ACG						Sets CG RAM address. CG RAM data is sent and received after this setting.	40 µS
T DD RAM Address	0	0	1	ADD							Sets DDRAM address. DD RAM data is sent and received after this setting.	40 µS
ad Busy Flag & ddress	0	1	BF	AC							Read Busy flag (BF) indicating internal operations are being performed and reads address counter contents	0 µS
e Data to 5 or DD RAM	1	0	Write Data								Write data into DD RAM or CG RAM	40 µS
d Data n CG or RAM	1	1	Read Data								Read data from DD RAM or CG RAM	40 µS

I/D = 1 : Increment
I/D = 0 : Decrement
S = 1 : Accompanies display shift
S/C= 1 : Display shift
S/C= 0 : Cursor move
R/L = 1 : Shift to the right
R/L = 0 : Shift to the left
DL = 1 : 8 Bits, DL = 0 : 4 Bits
N = 1 : 2 lines, N = 0 : 1 line
F = 1 : 5X10 Dots, F = 0 : 5X7 Dots
BF = 1 : internally operating
BF = 0 : Can accept instruction

DD RAM : Display data RAM
CG RAM : Character generator RAM
ACG : CG RAM Address
ADD : DD RAM Address corresponds to cursor address
AC : Address counter used for both DD and CG RAM Address

Execution time changes when frequency changes eg: when fcp or fosc is 270 KHZ :

40 µS x $\frac{250}{270}$ = 37S

Initializing By Instructions

The following sequence should be adopted for intialization

Power on

Wait for more than 15ms for V_{DD} to rise to 4.5v

RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0
0	0	0	0	1	1	*	*	*	*

Function Set

Wait for more than 4.1 ms

RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0
0	0	0	0	1	1	*	*	*	*

Function Set

Wait for more than 100 S

BF cannot be checked before this instruction

RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0
0	0	0	0	1	1	*	*	*	*

BF can be checed after this instruction. When BF is not checked, the waiting time between instruction is longer than the esxecutio time.(see instruction table)

RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0	
0	0	0	0	1	1	N	F	*	*	Function Set (See foot notes)
0	0	0	0	0	1	S/C	R/L	*	*	Cursor/Display Shift
0	0	0	0	0	0	1	D	C	B	Dispay ON/OFF Control
0	0	0	0	0	0	0	1	I/D	S/H	Entry Mode Set
0	0	0	0	0	0	0	0	0	1	Clear Display

Initialization Ends

Important Notes :

1 Specify the number of lines and character font (The number of lines and character font cannot be changed afterwards.)

2 Above example is valid for 8 bit interface.

3 . Read busy flag (BF) and ensure it is Ø before sending any instrtuction to ODM.

4 If busy flag check not possible, provide sufficient delay (atleast 5 times of execution period) before sending instruction to ODM.

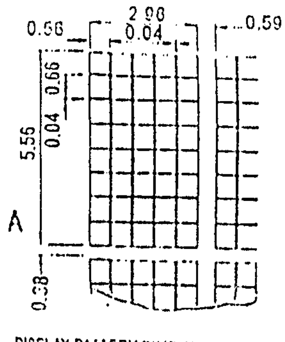
phanumeric LCD Module

Oriol
ELECTRONICS PVT

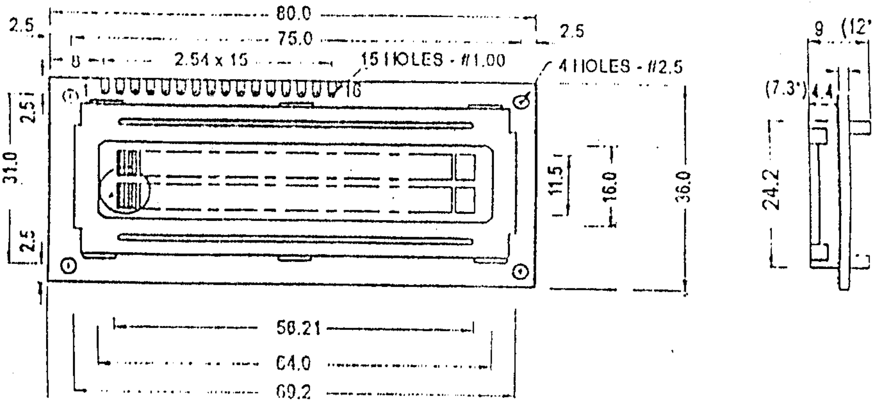
ODM-16216-9

16 Character 2 line

EXTERNAL DIMENSIONS



DISPLAY PATTERN DIMENSIONS



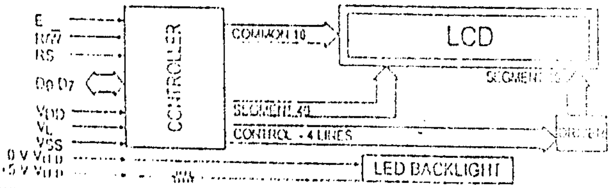
Specification marked * pertains to Backlight Version

GENERAL TOLERANCE : +/- 0.5

UNIT : m

BLOCK DIAGRAM

PIN NO.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
SYMBOL	VSS	VDD	VL	RS	R/W	E	D0	D1	D2	D3	D4	D5	D6	D7	+VLED	-VL



Definition of Terminals

SYMBOL	LEVEL	FUNCTION
V_{SS}	-	0v Power Supply
V_{DD}	-	+5V Power Supply
V_L	-	Contrast Voltage
RS	H/L	H - Data Register Select L - Command/Status Register Select
R/W	H/L	H - Read L - Write
E	⎍	Enable signal
D0-D7	H/L	Data Bus Lines
$+V_{LED}$	-	+5V Backlight Supply Voltage
$-V_{LED}$	-	0V Backlight Supply Voltage

ODM 16216 - 9SLW