DATA INTERFACE TO PRINTER

USING PIC MICROCONTROLLER

PROJECT REPORT

SUBMITTED BY

LAKSHMI R.

NAGARAJAN K.

NAGARAJAN M.

RAJALAKSHMI E.

GUIDED BY

Mr. G.C. THIYAGARAJAN M.E.,

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING IN

ELECTRONICS AND COMMUNICATION ENGINEERING

OF THE BHARATHIAR UNIVERSITY , COIMBATORE.

Department of Electronics & Communication Engineering

Kumaraguru College of Technology

Coimbatore – 641 006

Kumaraguru College of Technology

Coimbatore – 641006.

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project entitled

DATA INTERFACE TO PRINTER

USING PIC MICROCONTROLLER

has been submitted by

P- 1391

LAKSHMI R.

NAGARAJAN K.

NAGARAJAN M.

RAJALAKSHMI E.

in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering in **Electronics & Communication Engineering**

branch of **BHARATHIAR UNIVERSITY, Coimbatore**

during the academic year 2002-2003.

18/8/03

(Internal guide)

(Head of the Department)

Certified that the candidate was examined by us in the project work

Viva – Voce Examination held on 18-03-2003

University Register Number

18/3/03

(Internal Examiner)

(External Examiner)

[redacted]

CERTIFICATE

This is to certify that the following B.E. [Branch : Electronics & Communication Engineering] students of **KUMARAGURU COLLEGE OF TECHNOLOGY**, Coimbatore, had undertaken their project **"Data Interface using PIC Microcontroller"**, from May 2002 to March 2003 at our Industry and have successfully completed it.

Ms.E. Rajalakshmi
Ms.R. Lakshmi
Mr.K. Nagarajan
Mr.M. Nagarajan

Their performance during that period was found to be good. We wish them all success.

Place : Coimbatore
Date : March 17, 2003

Industry Seal

[redacted]

K K VENKATARAMAN
Vice President - R&D

premier polytronics pvt. ltd.

Acknowledgements

We are greatly indebted to our beloved Principal Dr.K.K. Padmanabhan B. Sc. (Engg.), M. Tech., Ph. D., **who has been the backbone of all our deeds.**

We earnestly express our gratitude and sincere thanks to our guide Mr.G.C.Thiyagarajan M.E., Department of Electronics & Communication Engineering **for his consummate technical guidance, with constant encouragement and suggestions in carrying out this project successfully.**

We would also like to thank Prof. M.Muthuraman Ramaswamy M.E.,FIE,.FIETE.,MIEEE(USA).,MISTE.,MBMESI.,C.ENG.(I)., Head of the Department of Electronics & Communication Engineering **for lending a helping hand in this project.**

We also express our sincere thanks to Mr. K.K.Venkatraman , Vice President , Premier Polytronics Limited , for giving us an opportunity to do this project in their organization. We express our deepest gratitude to our external guide Mrs.Tahsina Jammal ,Deputy Engineer ,Premier Polytronics Limited and Mr. S. Pradeep Kumar of Premier Polytronics Limited, Coimbatore **for guiding us throughout the project work.**

We also express our thanks to all the faculty members of the Department of Electronics and Communication Engineering **for their support.**

We owe much to our parents and friends **for their moral support and valuable help rendered to us.**

SYNOPSIS

This project deals with interfacing yarn clearing system M2010 to a printer. The output of any device cannot directly be given to a printer. So we design an interface such that it accepts the output of the system to the printer.

It is required that the printer should print data only when needed by the user.This interface is specifically designed for parallel printing. The interface requires sufficient memory capacity so that it holds data output from the system continuously.

The interface should efficiently handle data without loss or redundancy. It should be able to interact effectively with the system and the printer ,handle interrupts , timers , serial transmission and reception, speed compatibility and low power consumption.

According to the above requirements the best microcontroller for this application will be ***PIC16F876.***

CONTENTS

CHAPTER – 5

Algorithm

5.1 USART
5.2 EU Address generation
5.3 Get data
5.4 Retrieve & store data
5.5 Printer interfacing
5.6 Print out format

CHAPTER – 6

Flow chart

CHAPTER – 7

Program coding

Chapter – 8

Conclusion

Bibliography

Appendix

Chapter -1

INTRODUCTION

INTRODUCTION

The project "PRINTER INTERFACE USING PIC MICROCONTROLLER" is for tapping 8-bit data from the yarn clearing machine M2010 and sending to the printer.There are five types of yarn :

- Neps
- Short Thick
- Long Thick
- Long Thin
- Count

An interface using PIC16F876 is designed to collect data regarding the type of yarn and store in an external memory.The data collection is done for a day's work.The controller is reset after 24 hours so that fresh data for the next day can be stored.printout of the stored data can be taken as and when required.

The data source is the yarn clearing system from where data is tapped serially using RS-232 and then stored in memory through latches by accessing the port pins of the microcontroller.Handshaking signals are provided for parallel printing from memory to printer.

Chapter - 2

WHY CHOOSE PIC ?

WHY CHOOSE PIC ?

PIC16f876 is a 28 – pin IC with three bi-directional ports, supporting USART , with three TIMER modules, Flash memory and various special features and yet very compatible. PIC16F876 device has a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection.

The following are the core features of PIC which increases its utility:

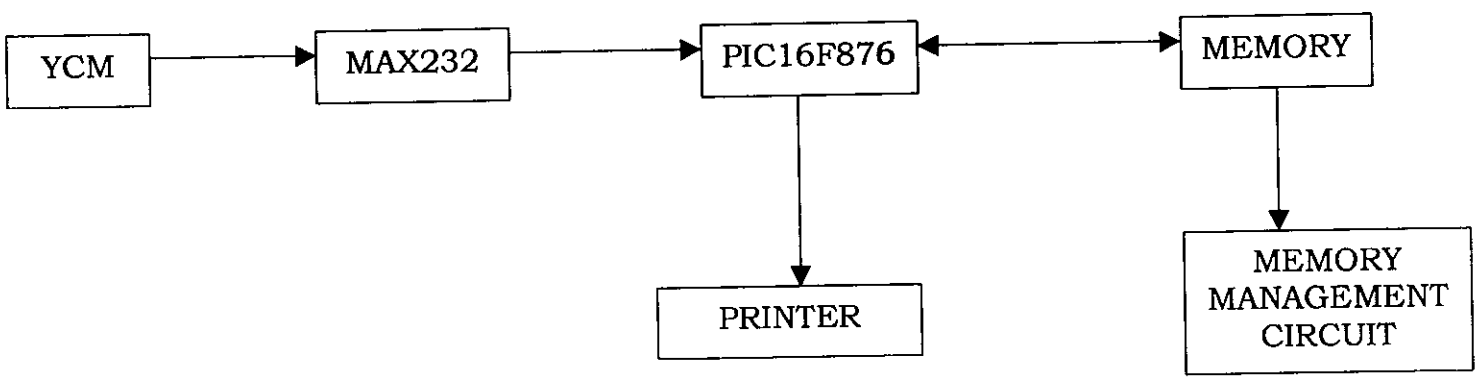
- It consists of high performance RISC CPU.
- 35 single cycle instructions except for program branches which are two cycle instructions.
- It employs 20MHz clock input with operating speed of 200ns instruction per cycle.
- It is pinout compatible to the PIC16C73B/74B/76/77
- It provides the options of Power–On Reset, Power-up Timer(PWRT) and Oscillator Start Timer(OST).
- It has a Watch Dog Timer(WDT) with its own on-chip RC oscillator for reliable operation.
- It provides the options for programmable code protection and power saving SLEEP mode.

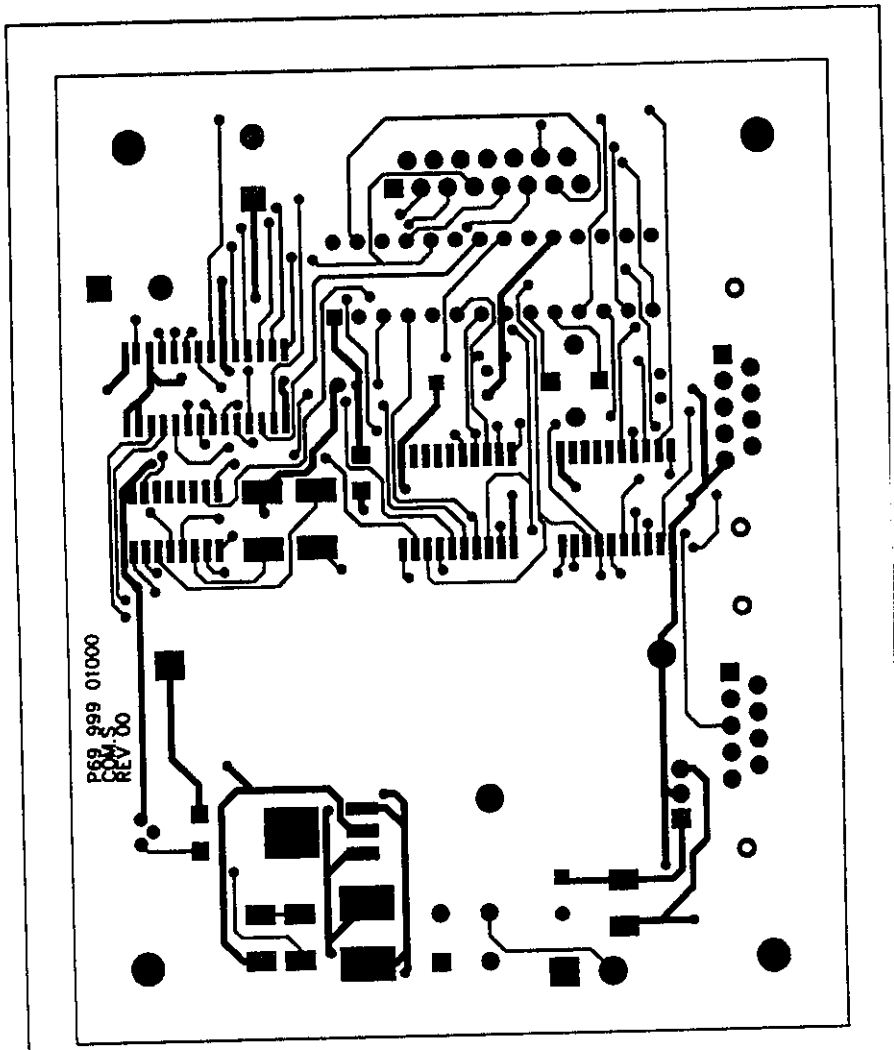
- It supplies the advantage of low power, high speed CMOS FLASH/EEPROM technology.
- It provides a fully static design and selectable oscillator options.
- It has a wide commercial, Industrial and Extended temperature ranges.
- It offers a wide operating voltage range : 2.0V to 5.5V.
- It provides in-circuit Debugging via two pins, in-circuit serial programming via two pins.
- It offers Processor read/write access to program memory.
- It presents the merit of low power consumption:

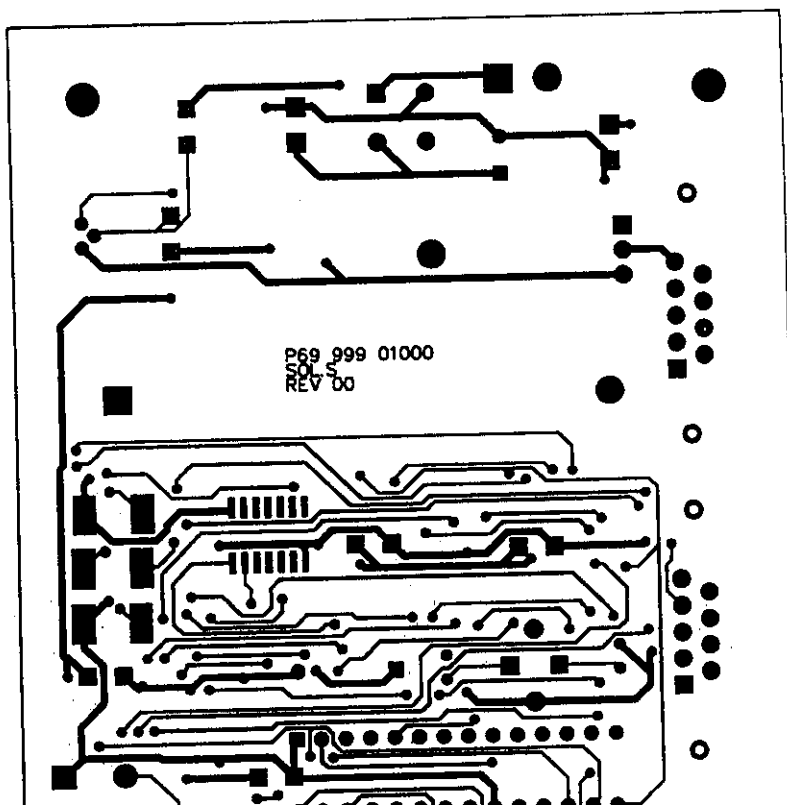
  - < 0.6 mA typical @ 3V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

These merits offered by PIC is the best choice for the compatible, high speed operations of this Project of data interface to printer.

BLOCK DIAGRAM




P69 999 01000
COM S
REV 00


P69 999 01000
SOL S
REV 00

PIC16F876

SPECIAL FEATURES OF THE CONTROLLER

PIC16F876 has a host of features intended to maximize system reliability, minimize Cost through elimination of external components, provide power saving operating modes and offer code protection.

Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits is used to select various options.

DATA EEPROM AND FLASH PROGRAM MEMORY:

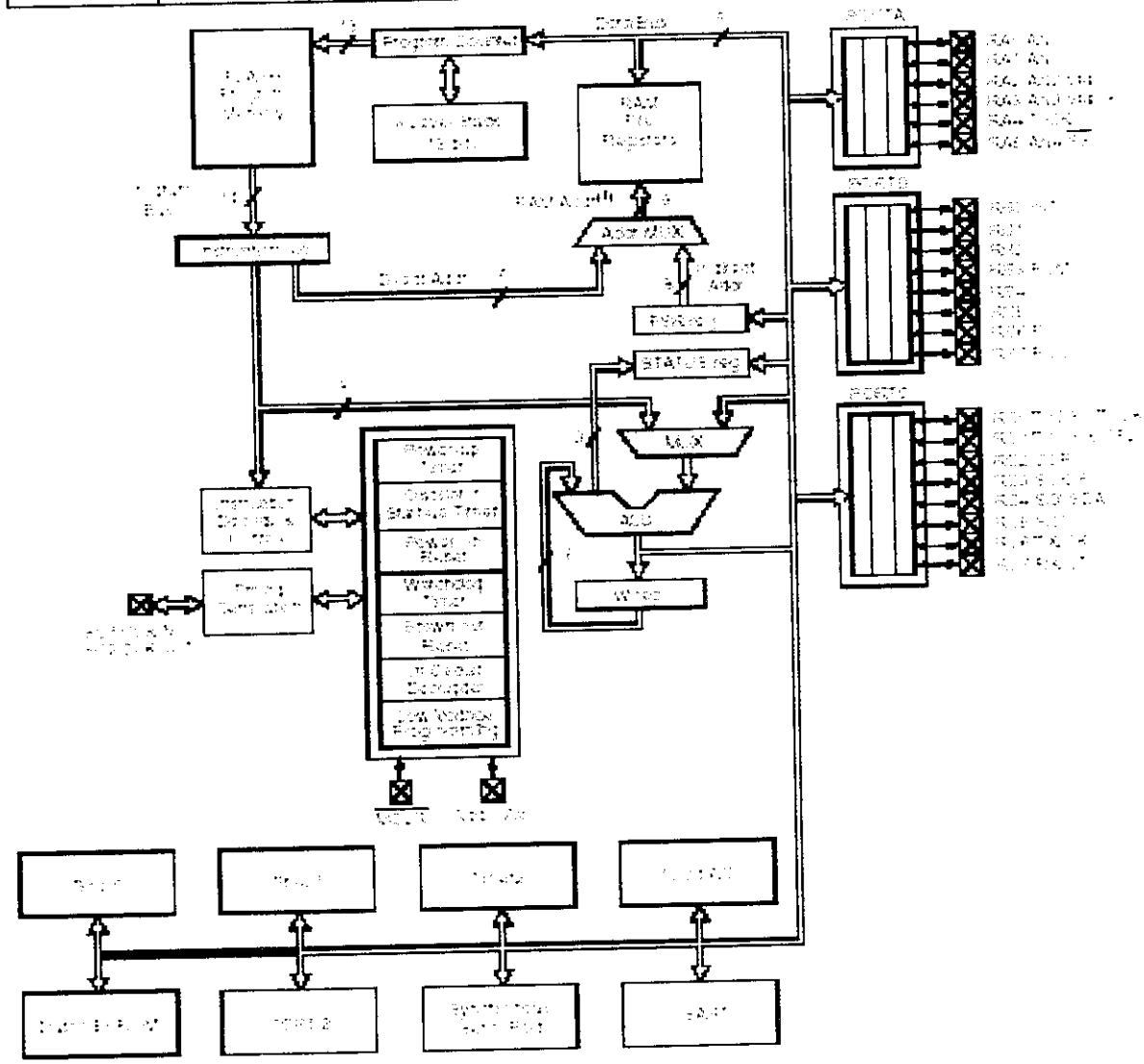
The Data EEPROM and FLASH Program Memory are readable and writable during normal operation over the entire VDD range. EEPROM data memory can be used to store frequently updated values.

PCL and PCLATH:

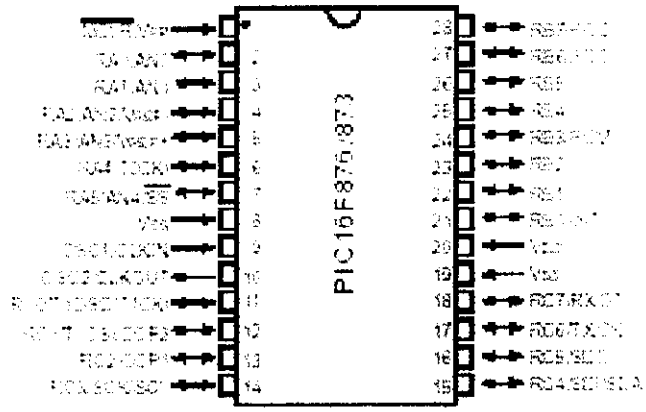
The program counter (PC) is 13-bits wide. The low byte is from the PCL register, which is a readable & writable register.The upper bits (PC<12:8>) are not readable, but are indirectly writable through the PCLATH register. On any RESET, the upper bits of the PC will be cleared.

ARCHITECTURE OF PIC16F876

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F873	4K	192 Bytes	128 Bytes
PIC16F876	8K	368 Bytes	256 Bytes



The following is the pin configuration of PIC16F876:



I/O PORTS:

PORTA & TRISA:

PORTA is a 6-bit wide, bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin). Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations.Therefore, a write to a port implies that the port pins are read, the value is modified and then written to the port data latch. The TRISA

register controls the direction of the RA pins, even when they are being used as analog inputs.The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

PORTB & TRISB:

PORTB is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a i-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., put the contents of the output latch on the selected pin).

PORTC & TRISC:

PORTC is an 8-bit wide, bi-directional port. The Corresponding data direction register is TRISC. Setting a TRISC bit (=1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a hi-Impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., put the contents of the output latch on the selected pin).

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

TIMER1 MODULE:

The Timer1 module is a 16-bit timer/counter consisting of two 8-bit registers (TMR1H and TMR1L), which are readable and writable. The TMR1 Register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The TMR1 Interrupt, if enabled is generated on overflow, which is latched in interrupt flag bit TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing TMR1 interrupt enable bit TMR1IE (PIE1<0>).

Timer1 can operate in one of two modes:

- As a timer
- As a counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>).In Timer mode, Timer1 increments every instructio

cycle. In Counter mode, it increments on every rising edge of the external clock input. Timer1 can be enabled/disabled by setting/clearing control bit TMR1ON (T1CON<0>).

TIMER2 MODULE:

Timer2 is an 8-bit timer with a prescaler and a postscaler. The Timer2 module has an 8-bit period register, PR2.Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register.The PR2 register is initialized to FFh upon RESET.

ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART):

The USART module is one of the two seriall/O modules. (USART is also known as a Serial Com-munications Interface or SCI.) The USART can be con-figured as a full duplex asynchronous system that can communicate with peripheral devices such as CRT ter-minalsnd personal computers, or it can be configured as a half duplex synchronous system that can commu-nicate with peripheral devices such as A/D or D/A inte-grated circuits, serial EEPROMs etc. The USART can be configured in the following modes:

- Asynchronous (full duplex)

- Synchronous - Master (half duplex)
- Synchronous - Slave (half duplex)

CAPTURE/COMPARE/PWM MODULES:

Each Capture/Compare/PWM (CCP) module contains a 16-bit register which can operate as a:

- 16-bit Capture register
- 16-bit Compare register
- PWM Master/Slave Duty Cycle register

Both the CCP1 and CCP2 modules are identical in operation, with the exception being the operation of the special event trigger.

CCP1 Module:

Capture/Compare/PWM Register1 (CCPR1) is comprised of two bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP1CON register controls the operation of CCP1. The special event trigger is generated by a compare match and will reset Timer1.

CCP2 Module:

Capture/Compare/PWM Register2 (CCPR2) is comprised of two 8-bit registers: CCPR2L (low byte) and CCPR2H (high byte). The CP2CON register controls the operation of CCP2. The special event trigge is generated by a compare match and will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

INSTRUCTION SET :

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 includes **byte-oriented**, **bit-oriented**, and **literal and control** operations.

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. 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 μs. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs.

SPECIAL FEATURES OF PIC:

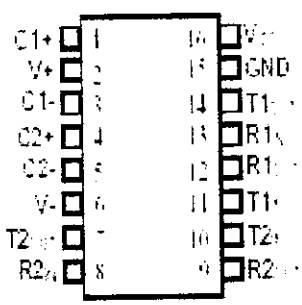
PIC16F87X devices have a Watchdog Timer, which can be shut-off only through configuration bits. It runs off its own RC oscillator for added reliability.SLEEP mode is designed to offer a very low current Power-down mode. The user can wake-up from SLEEP through external RESET, Watchdog Timer Wake-up, or through an interrupt.

MAX232

FEATURES:

- High data rate - 250 kbits/sec under load
- 16-pin DIP or SOIC package
- Operate from single +5V power
- Meets all EIA-232E and V0.28 specifications
- Uses small capacitors: 10 µF
- Optional industrial temperature range available (-40...C to +85...C)

The MAX232A is a dual RS-232 driver/receiver pair that generates RS-232 voltage levels from a single +5 volt power supply. Additional ±12-volt supplies are not needed since the MAX232A uses on-board charge pumps to convert the +5-volt supply to ±10 volts. The following is the pin configuration of MAX 232 :



The MAX232A contains two drivers and two receivers.

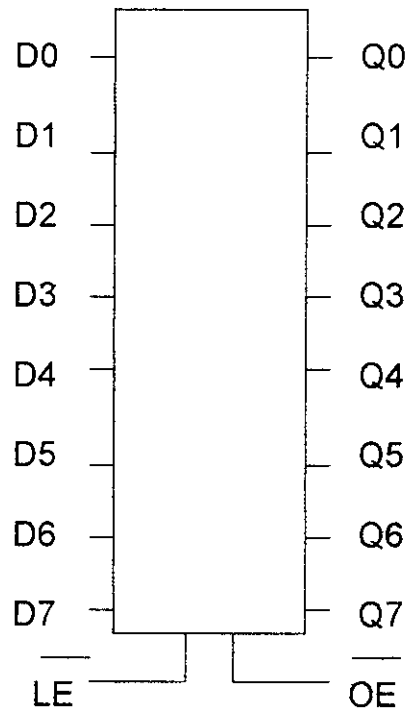
RS-232 DRIVERS:

The two RS-232 drivers are powered by the internal ±10 volt supplies generated by the on-board charge pump. The driver inputs are both TTL and CMOS compatible.. As required by EIA-232E and V.28 specifications, the slew rate at the output is limited to less than 30 volts/μs. Driver outputs maintain high impedance when power is off.

RS-232 RECEIVERS:

The two receivers conform fully to the RS-232E specifications. The input impedance is typically 5k and can withstand up to ±25 volts with or without VCC applied. The TTL/CMOS compatible outputs of the receivers will be low whenever the RS-232 input is greater than 2.4 volts. The receiver output will be high when the input is floating or driven between +0.8 volts and -25 volts.

LATCH 74LS373:



SALIENT FEATURES OF OCTAL TRANSPARENT LATCH

- Common Latch Enable Control
- Common 3 State Output Enable Control
- Buffered Inputs
- 3 State Outputs
- Bus Line Driving Capacity
- Typical Propagation Delay = 12ns
- Low Power Consumption

TRUTH TABLE:

$\overline{OE}$	$\overline{LE}$	DATA	OUTPUT
L	H	H	H
L	H	L	L
L	L	l	L
L	L	h	H
H	X	X	Z

L = Low voltage level
H = High voltage level
X = Don't care
Z = High impedance state
l = Low voltage level one set up prior to the high to low latch enable
h = high voltage level one set up prior to the high to low latch enable

They possess the low power consumption of standard CMOS integrated circuits as well as the ability to drive 15 LSTTL devices. The outputs are transparent to the inputs when the latch enable is high. When the latch enable becomes low , the data is latched. The output enable controls the 3 – state outputs. When the output enable is high the outputs are in high impedance state. The operation of latch is independent to the state of the output enable.

POWER SUPPLY CIRCUIT

Power supplies are extensively used in all industrial applications.They are required to meet all or most of the following specifications.

- Isolation between source and the load
- High power density for reduction of size and weight
- Controlled direction of power flow
- High conversion efficiency
- Provides regulated ripple free input and output wave forms with low total harmonic distortion
- Controlled power factor if source is an AC voltage

DESCRIPTION:

The power supply section consists of step down transformer, a bridge rectifier and a voltage regulator.The AC input is stepped down using transformer and given to the bridge rectifier which converts AC input to pulsating DC.This pulsating DC is passed through ripple factor circuit for smooth and pure DC.

The output voltage is fed to voltage regulators like 78XX IC in which XX is replaced by fixed positive / negative output voltage ratings.This is done to stabilize the DC voltage for sensitive circuits.We use 7805 regulators to get +5V DC supply to PIC and other IC's.

The capacitors are used before and after regulating IC in

order to reduce the oscillations and ripples.The free wheeling diode IN4007 is employed to provide a flyback path for discharging excess capacitance available at the output terminals of the regulating IC.Thus the output of the power supply designed is a constant regulated DC voltage of +5V.

LM 7805- SERIES VOLTAGE REGULATOR

The LM 78XX series of 3 terminal regulators is available with several fixed output voltages making them useful in a wide range of applications.One of these is local on card regulation,eliminating the distribution problems associated with single point regulation.

The voltages available allow these regulators to be used in logic systems , instrumentation and other solid state electronic equipments.Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltage currents.

The LM 78XX series is available in an aluminium TO-3 package which willl allow 1.0 A load current if adequate heat sinking is

provided.Current Limiting is included to limit the weak output current to a safe value.Safe area protection for the output transistor is provided to limit

for heat sinking provided the thermal shutdown circuit takkes over preventing the IC from overheating

FEATURES:

- Output current in excess of 1 A
- Internal Thermal overload protection
- No external components required.
- Output transistor safe area protection.
- Internal short circuit current limit.
- Available in Aluminium TO - 3 package.

ELECTRICAL CHARACTERISTICS:

- Input voltage = 10 V
- Output current = 500 mA
- Cin = 0.33 microFarad
- Cout = 0.1 microFarad
- Output voltage = 5 V
- Short circuit current = 750 mA
- Peak output current = 2.2 A

RS-232

RS–232 SPECIFICATIONS:

RS–232 is a "complete" standard. This means that the standard sets out to ensure compatibility between the host and peripheral systems by specifying

1) common voltage and signal levels
2) common pin wiring configurations
3) a minimal amount of control information between the host and peripheral systems.

ELECTRICAL CHARACTERISTICS:

The electrical characteristics section of the RS–232 standard includes specifications on voltage levels, rate of change of signal levels, and line impedance. A high level for the driver output is defined as being +5 to +15 volts and a low level for the driver output is defined as being between –5 and –15 volts. The receiver logic levels were defined to provide a 2 volt noise margin.It is necessary to note that, for RS–232 communication, a low level (–3 to –15 volts) s defined as a logic 1 and is historically referred to as "marking". Likewise a high level (+3 to +15 volts) is defined as a logic 0 and is referred to as "spacing". The RS–232 standard also limits the maximum slew rate at the driver output. This limitation was included to help reduce the likelihood of cross–talk between

adjacent signals. The slower the rise and fall time, the smaller the chance of cross talk. With this in mind, the maximum slew rate allowed is 30 V/μs.

FUNCTIONAL CHARACTERISTICS:

Since RS–232 is a "complete" standard, it includes more than just specifications on electrical characteristics. The second aspect of operation that is covered by the standard concerns the functional characteristics of the interface. This essentially means that RS–232 has defined the function of the different signals that are used in the interface. These signals are divided into four different categories:

common, data, control, and timing.

The standard provides an abundance of control signals and supports a primary and secondary communications channel.

MECHANICAL INTERFACE CHARACTERISTICS:

The third area covered by RS–232 concerns the mechanical interface. In particular, RS–232 specifies a 25–pin connector. This is the minimum connector size that can accommodate all of the signals defined in the functional portion of the standard. The connector for DCE equipment is male for the connector housing and female for the connection pins. The DTE connector is a female housing with male connection pins.

AS7C3256 – MEMORY IC

Features:

- • AS7C3256 (3.3V version)
- • Industrial and commercial temperature
- • Organization: 32,768 words × 8 bits
- • High speed
  - 12/15/20 ns address access time
  - 6, 7, 8 ns output enable access time
- • Very low power consumption: ACTIVE
  - 660mW (AS7C256) / max @ 12 ns
  - 216mW (AS7C3256) / max @ 12 ns
- • Very low power consumption: STANDBY
  - 22 mW (AS7C256) / max CMOS I/O
  - 7.2 mW (AS7C3256) / max CMOS I/O
- • Easy memory expansion with CE and OE inputs
- • TTL-compatible, three-state I/O
- • ESD protection ≥2000 volts
- • Latch-up current ≥200 Ma

The AS7C3256 is a 3.3V high-performance CMOS 262,144-bit Static Random-Access Memory (SRAM) device organized as 32,768 words × 8 bits. It is designed for memory applications requiring fast data access at low voltage, including Pentium TM , PowerPC TM , and

portable computing. Alliance's advanced circuit design and process techniques permit 3.3V operation without sacrificing performance or operating margins.

The device enters *standby mode* when CE is high. CMOS standby mode consumes ≤3.6 mW. Normal operation offers 75% power reduction after initial access, resulting in significant power savings during CPU idle, suspend, and stretch mode. The chip enable (CE) input permits easy memory expansion with multiple-bank memory organizations.

A write cycle is accomplished by asserting chip enable (CE) and write enable (WE) LOW. Data on the input pins I/O0-I/O7 is written on the rising edge of WE (write cycle 1) or CE (write cycle 2). To avoid bus contention, external devices should drive I/O pins only after outputs have been disabled with output enable (OE) or write enable (WE).A read cycle is accomplished by asserting chip enable (CE) and output enable (OE) LOW, with write enable (WE) high. The chip drives I/O pins with the data word referenced by the input address. When chip enable or output enable is high, or write enable is low, output drivers stay in high-impedance mode. During V CC power-up, a pull-up resistor to V CC on CE is required to meet I SB specification. This parameter is sampled, but not 100% tested. WE is High for read cycle. CE and OE are Low for read cycle.

MEMORY RETENTION CIRCUIT

SRAM:

SRAM is a high speed reprogrammable memory technology which is limited by its volatility and relatively low density.As a volatile memory technology SRAM requires constant power to retain its contents. Built in battery backup is therefore required when main power source is turned off.

The memory retention circuit is used to retain data stored in memory even when power is switched off.This is done by supplying power using battery back – up. The 3.6 V battery is a re-chargeable one. The maintenance circuit charges the battery when power is on. The battery supplies power to the memory when power is off.

When power is on , Vcc = 5 V and the transistor is , and the diodes are reverse biased and so the voltage goes to the battery. When power is off , the diodes are forward biased and voltage drop occurs at the diodes for the battery to act as supply.

REGULATOR – REG1117

This is the regulator employed in the memory retention circuit. This regulator provides a regulated output of 3.3 V. This is well within the data retention rate of memory IC AS7C3256. The data retention rate of memory IC is 2 v.

The following are the special features of regulator 1117:

- 2.85v ,3v , 3.3v and adjustable versions.
- Output current = 800 mA maximum
- Output tolerance = (+/-) 1% maximum
- Drop – out voltage = 1.2 v maximum at Io = 800 mA.
- Thermal overload protection
- Internal current limit.

APPLICATIONS:

The following are the applications of 1117 regulator:

- SCS1 – 2 active termination.
- Hand – held data collection devices.
- High efficiency linear regulators.
- 5V linear regulators.
- Battery powered instrumentation.
- Battery management circuits .

Chapter - 4

SOFTWARE DESCRIPTIO

The data is transmitted from M2010 every 700 ms.

The data transmission protocol is as follows:

1. STX1 : ee

2. STX2 : ee

3. TAG ADDR : 0X81 h

4. EU ADDR :

5. Cut data 1

6. Cut data 2

7. Cut data 3

8. Cut data 4

.

.

.

14. Cut data 10

15. MVH 1

16. MVL 1

.

.

.

23. MVH 10

24. MVL 10

25. LENGTH L1

26. LENGTH H1

.

33. LENGTH L10

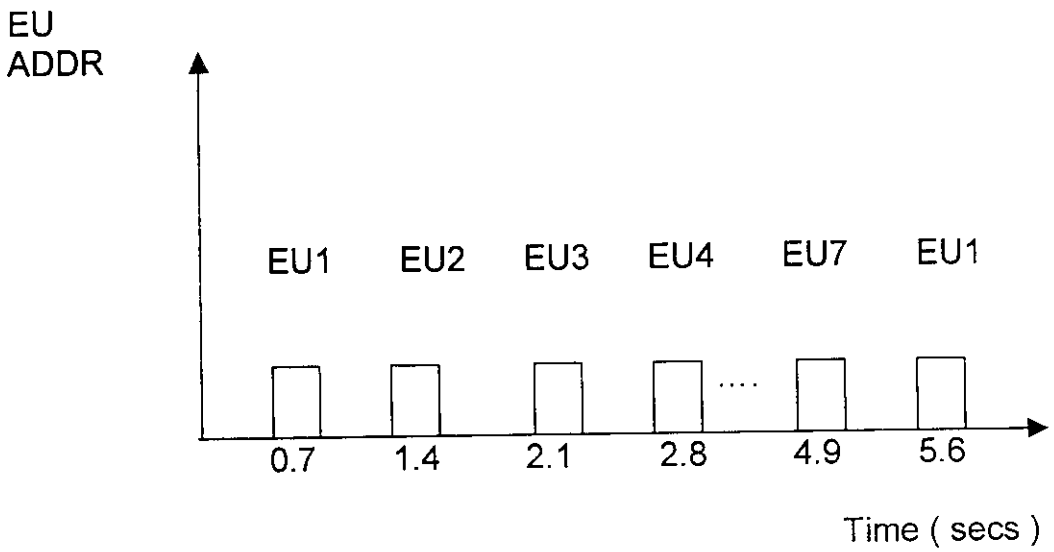
34. LENGTH H10

35. ERROR STATUS 1

36. ERROR STATUS 2

37. CHECK SUM

The first two bytes indicate the start of text.The third byte of the protocol is the TAG address which when equals 0x81 h specifies that the following bytes are valid data bytes.The next consecutive byte is the EU address



There are 8 EU in total with 10 drums per EU.The cut data of each drum is given in the next consecutive 10 bytes.

The cut data format is as follows:

STATUS	X	COUNT	X	LONG THIN	LONG THICK	SHORT THICK	NEPS

Each bit when set indicates the corresponding type of yarn.The MSB status is to check the running condition of the yarn machine.

STATUS = 1 ; yarn running
= 0 ; not running

LOGIC DESCRIPTION

The overall logic logic description of the printer interface project is as follows:

The data from the yarn clearing machine is passed on to the PCB through RS232 cable . The MAX232 IC transfers the data received at RS 232 9 pin connector to the USART receive pin of PIC16F876.

The PIC accepts data and checks for tag address validity and EU address. The port A is configured for address transfer and port B

for data transfer. The six port A pins are used to transfer 12 bit address to the memory through latch1 and latch2. The first 6 bits of address are passed on to the latch1 by configuring port A as output and enabling the latch by setting the latch enable pin. The next 6 bits of address are passed to latch2 similarly. By enabling memory write pin and latch output enable pin , the address is fully transferred to the memory. To read from memory a similar process is followed . Address is passed through latches, data is fetched from memory by enabling the memory read and output enable pins. Address is passed in port A and data in port B.

In memory, the corresponding ASCII code for every data is stored instead of in the binary format.Thus each type of entry in memory requires 3 bytes allotted to it. This is to make the printing option easier. Thus for every drum 21 bytes are allotted . The print out format is also stored in the memory in separate address location.

When data is received at the PIC, the EU address is identified. The address is sent to the memory via both latches by configuring port A as output. The memory location corresponding to address is accessed and the data is moved on to port B of PIC. Prior to this port B is configured as input. Thus the data obtained in the work register form port B is incremented and put back in the same locations in the memory by enabling the memory write pin and the latches.When writing to

memory the port A and B are configured as output. The address is passed through port A and data through port B as before. The port C pins are used for enabling different operations like latch enable, memory write, memory write, output enable, etc..

The printer interfacing part uses two handshaking signals namely STROBE and BUSY. The STROBE is an active low signal, activated by clearing RC3 pin. This handshaking signal is sent by the controller to the printer. The BUSY signal is activated by the setting of RC2 pin by the printer. If the printer is busy the pin is low . If the printer is not busy the status of the pin is active high.

When printer is ON the BUSY status is checked. The STROBE signal is sent to the printer. If the printer is free the contents of memory is transferred to the printer, byte by byte, by configuring the port A,B and C pins respectively. Before of transfer of data, the print out format is moved to the printer from memory first , and then the data. Thus the contents of memory are obtained in printed format as designed to fullfil the requirements.

Chapter - 5

ALGORITHM

ALGORITHM :

USART:

1. The baud rate generator register SPBRG is initialized for 9.6 K bauds.
2. The TXSTA register is initialized for enabling transmission at low speed , and asynchronous communication.
3. The RCSTA register is initialized for enabling the serial port , continuous reception and error control.
4. Check for RCIF in PIR1 register.
5. If set, the frame and overrun errors are checked and cleared.
6. Data received in RCREG is moved to the working register and a temporary register "datatemp".
7. Count register is initialized to 0xFF h to start incrementing count from 0x00
8. Count register MCOUNT is incremented twice to discard the first two bytes of received data i.e, the start of text bytes.
9. The third byte tag address is checked.
10. If the tag address is equal to 0x81h the following data bytes are stored,else discarded.
11. After all the 37 bytes are received polling pin 0 of port c checks for the print option.

EU ADDRESS GENERATION:

1. The last three bits of EU are checked and address is assigned respectively.

2. The LSB of addres is moved to latch1 by enabling RC1

3. The MSB is moved to latch2 by enabling RC4 and thus output of both latches are enabled address is passed.

GET DATA:

1. The status of the machine is known by checking the bit7 of the received cut data.

2. If "1" then the run and monitor registers are incremented,else only the monitor register is incremented.

3. The type of yarn is checked for and the corresponding subroutines are called.

4. Latch1 is incremented till all bits are set.If set both latches are incremented.

5. Total is incremented for every data received.

RETRIEVE AND STORE DATA:

1. Data is obtained from the memory location pointed(Initially pointed to units, then decremented to point to tenths, then decremented to point to hundredths)

2. Port B is configured as input to get data from memory.

3. The received data is incremented and stored back in memory by configuring Port B as output and the corresponding Port C pins.

4. The latches are incremented using the same procedure as before for address.

5. After storing all data the pointer is incremented by 1 for getting the next data of the protocol.

PRINTER INTERFACING:

1. The printer is checked for ON / OFF condition.
2. If ON the STB (active low) signal is sent.
3. The BUSY (active high) is checked.
4. If NOT BUSY data is sent as per requirement from memory.

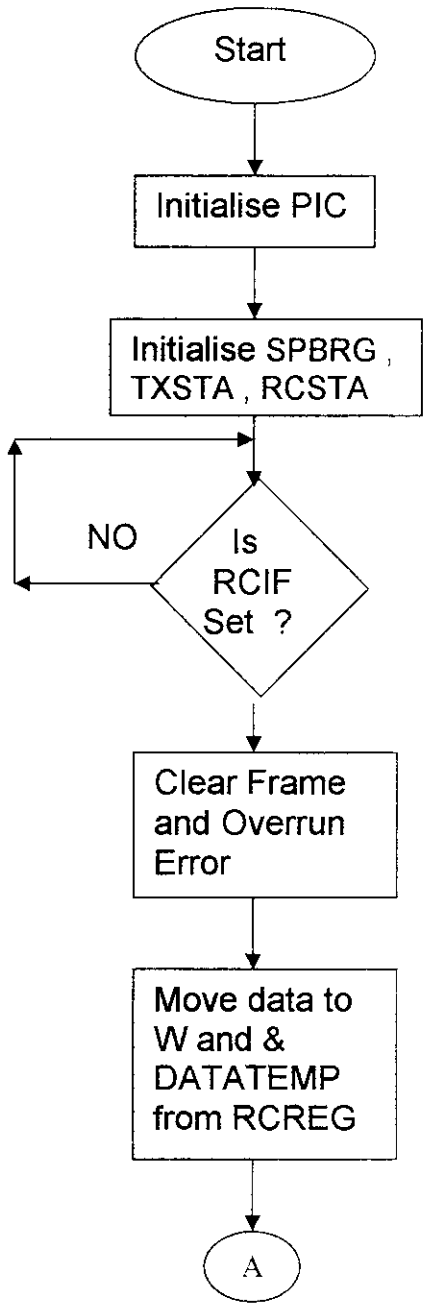
PRINT OUT FORMAT:

The print out format is as follows:

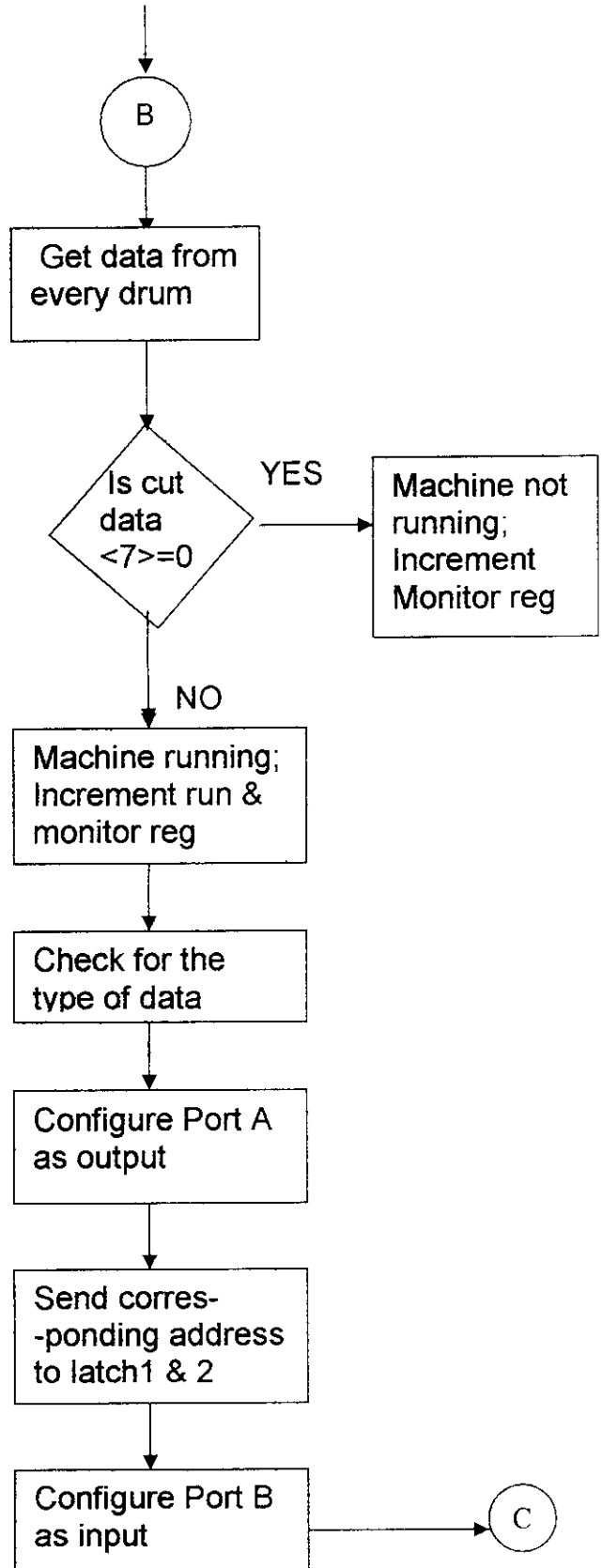
1. The row count is initialized as 0x06 for printing the types of data and the total.
2. The column count is initialized to 0x0A for 10 drums.
3. The ASCII values of each heading are printed.
4. For each type of yarn, the stored data in each drum is printed.
5. Row count is decremented and checked if zero.
6. If zero , MAIN is executed ,else the next type of data for all the drums are stored.

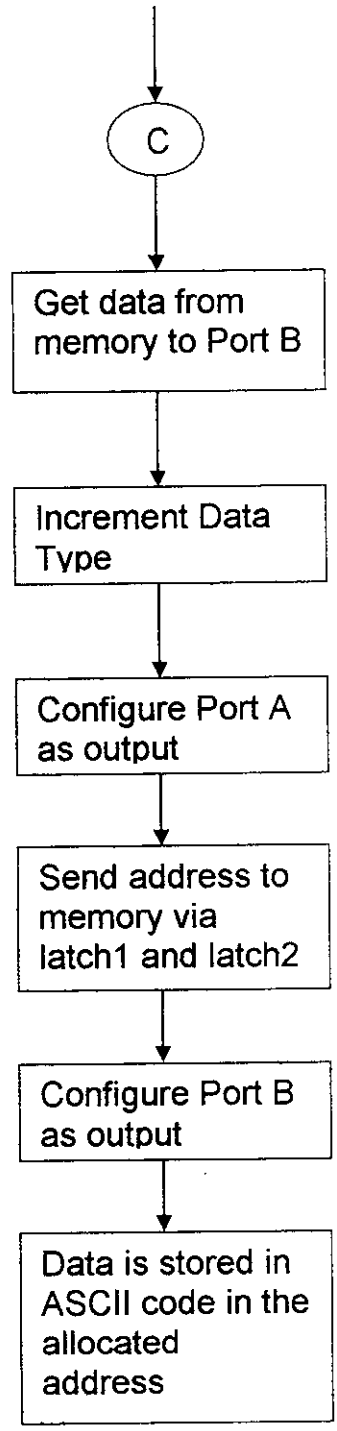
FLOWCHART:

USART:

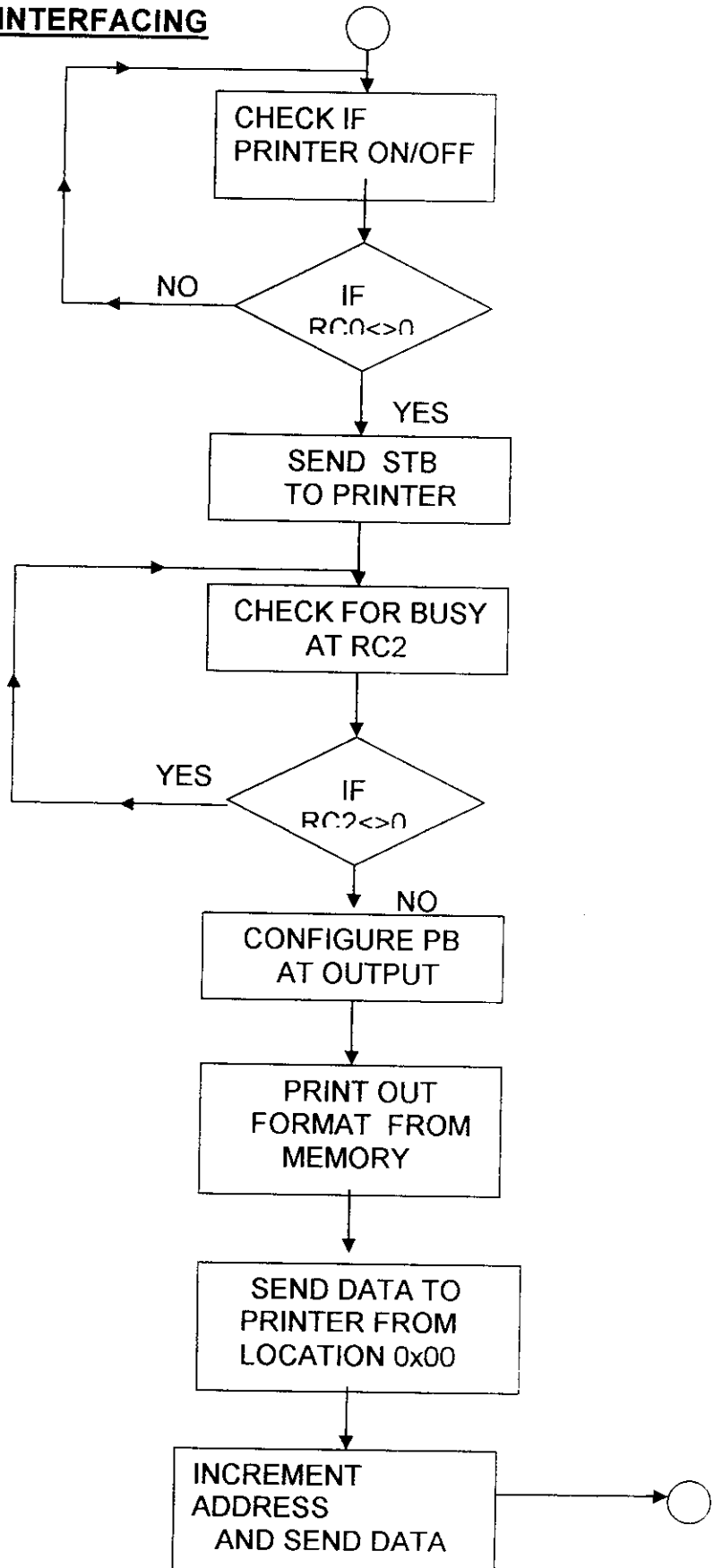


GET & STORE DATA:




C
Get data from memory to Port B
Increment Data Type
Configure Port A as output
Send address to memory via latch1 and latch2
Configure Port B as output
Data is stored in ASCII code in the allocated address

PRINTER INTERFACING



Chapter -

PROGRAM CODING

```
LIST P=PIC16F876
INCLUDE "P16F876.INC"

COUNT           EQU     0X20
MCOUNT          EQU     0X21
DTEMP           EQU     0X22
LATCH1          EQU     0X23
LATCH2          EQU     0X24
TOTLATCH1       EQU     0X25
TOTLATCH2       EQU     0X26
TOTCOUNT        EQU     0X27
EUADDR          EQU     0X28
TEMP1           EQU     0X29
NEPS            EQU     0X2A
STHICK          EQU     0X2B
LTHICK          EQU     0X2C
LTHIN           EQU     0X2D
DATACOUNT       EQU     0X2E
RUNREG          EQU     0X2F
MONITOREG       EQU     0X30
LDATA           EQU     0X31
DATA10COUNT     EQU     0X32
PRINTERON       EQU     0X33
ADDRCOUNT       EQU     0X34
RETRIEVCOUNT    EQU     0X35
NUMBER          EQU     0X36
COCOUNT         EQU     0X37
ROCOUNT         EQU     0X38
TEMLATCH1       EQU     0X39
TEMLATCH2       EQU     0X3A
SETCOUNT        EQU     0X3B
SPACECOUNT3     EQU     0X3C
PLATCH1         EQU     0X3D
PLATCH2         EQU     0X3E
BIT4            EQU     0X3F
DELAYCOUNT      EQU     0X40
T1              EQU     0X41
T2              EQU     0X42
MEMINIT         EQU     0X43
RP1             EQU     H'0006'
RP0             EQU     H'0005'
PORTA           EQU     0X05
```

```
PORTB           EQU     0X06
PORTC           EQU     0X07
SPBRG           EQU     0X99
TXSTA           EQU     0X98
RCSTA           EQU     0X18

                ORG 0x00
                GOTO MAIN


MAIN
;_____________________________________________________________________
; Memory initialisation ie., all locations by 0x30
; Initialising the ports[A,B,C],MCOUNT.
;_____________________________________________________________________

    BSF         STATUS,RP0
    CLRF        PORTB
    BCF         STATUS,RP0
    MOVLW       #0X03
    MOVWF       T2
    MOVLW       #0X01
    MOVWF       DATACOUNT
ML
    MOVLW       #0XFF
    MOVWF       T1
ML1
    DECFSZ      T1,1
    GOTO        STORE
    DECFSZ      T2,1
    GOTO        ML



MAIN1
    BSF         STATUS,RP0
    BCF         STATUS,RP1
    MOVLW       0X06
    MOVWF       ADCON0
```

```
MOVLW   0X00
MOVWF   PORTA
MOVLW   0X00
MOVWF   PORTB
MOVLW   0X97
MOVWF   PORTC
BCF     STATUS,RP0
MOVLW   0XFF
MOVWF   MCOUNT
BSF     STATUS,RP0
MOVLW   0X06
MOVWF   TXREG
BCF     STATUS,RP0
MOVLW   0X00
BSF     STATUS,RP0
BCF     STATUS,RP1
MOVWF   RCSTA
BCF     STATUS,RP0
BCF     STATUS,RP1
MOVLW   0X90
MOVWF   RCSTA


SUB1

    BTFSS   PIR1,5
    GOTO    SUB1
    BTFSC   RCSTA,1
    CALL    SUB2
    BTFSC   RCSTA,2
    BCF     RCSTA,2
    MOVLW   0X81
    MOVWF   RCREG
    MOVF    RCREG,0
    INCF    MCOUNT,1
    BTFS S  MCOUNT,1
    GOTO    SUB1
    MOVWF   DTEMP
    BTFSS   DTEMP,0
    GOTO    SUB
```

```
;_____________________________________
;GETTING EU ADDRESS
;_____________________________________


MAINSUB

    BTFSS     PIR1,5
    GOTO      MAINSUB
    BTFSC     RCSTA,1
    CALL      SUB2
    BTFSC     RCSTA,2
    BCF       RCSTA,2
    INCF      MCOUNT, 1
    MOVLW     0X00
    MOVWF     RCREG
    MOVF      RCREG,0
    MOVWF     EUADDR


;-------------------------------------------------------------------------------
; Checking the EU address
;
; REGISTERS USED:EUADDR, DATA10COUNT=0X0A,LATCH1,
LATCH2,TOTOLATCH1,TOTLATCH2
;-------------------------------------------------------------------------------


EUCHECK

    BCF       STATUS,RP0
    MOVLW     0X03
    MOVWF     EUADDR
    MOVLW     0X0A
    MOVWF     DATA10COUNT
    BTFSC     EUADDR,2
    GOTO      EU5678
    BTFSC     EUADDR,1
    GOTO      EU34
    BTFSC     EUADDR,0
    GOTO      EU2
```

```
;-------------------------------------
;EUADDRESS #1
;-------------------------------------

    BCF     STATUS,RP0
    CLRF    PORTA
    MOVLW   0X00
    MOVWF   LATCH1
    MOVWF   LATCH2
    MOVWF   TOTLATCH1
    MOVWF   TOTLATCH2
    BSF     STATUS,RP0
    BCF     PORTC,1
    BCF     STATUS,RP0
    MOVWF   PORTA
    BSF     STATUS,RP0
    BSF     PORTC,1
    BCF     PORTC,4
    BCF     STATUS,RP0
    MOVWF   PORTA
    BSF     STATUS,RP0
    GOTO    GETDATA

;-------------------------------------
;EUADDRESS #2
;-------------------------------------

EU2

    BSF     STATUS,RP0
    BCF     PORTC,1
    BCF     STATUS,RP0
    MOVLW   0X12
    MOVWF   LATCH1
    MOVWF   TOTLATCH1
    MOVWF   PORTA
    BSF     STATUS,RP0
    BSF     PORTC,1
    BCF     STATUS,RP0
```

```
    MOVLW   0X03
    MOVWF   LATCH2
    MOVWF   TOTLATCH2
    BSF     STATUS,RP0
    BCF     PORTC,4
    BCF     STATUS,RP0
    MOVWF   PORTA
    BSF     STATUS,RP0
    BSF     PORTC,4
    GOTO    GETDATA

;-------------------------------------------------------------
;CHECKING FOR EUADDR 3 OR 4
;-------------------------------------------------------------

EU34

    BCF     STATUS,RP0
    BTFSC   EUADDR,0
    GOTO    EU4

;-----------------------------------------
;EUADDRESS #3
;-----------------------------------------


    BSF     STATUS,RP0
    BCF     PORTC,1
    BCF     STATUS,RP0
    MOVLW   0X24
    MOVWF   LATCH1
    MOVWF   TOTLATCH1
    MOVWF   PORTA
    BSF     STATUS,RP0
    BSF     PORTC,1
    BCF     PORTC,4
    BCF     STATUS,RP0
    MOVLW   0X06
    MOVWF   LATCH2
    MOVWF   TOTLATCH2
    MOVWF   PORTA
```

```
    BSF         STATUS,RP0
    BSF         PORTC,4
    GOTO        GETDATA

;-----------------------------------------
;EUADRESS #4
;-----------------------------------------


EU4

    BSF         STATUS,RP0
    BCF         PORTC,1
    BCF         STATUS,RP0
    MOVLW       0X36
    MOVWF       LATCH1
    MOVWF       TOTLATCH1
    MOVWF       PORTA
    BSF         STATUS,RP0
    BSF         PORTC,1
    BCF         PORTC,4
    BCF         STATUS,RP0
    MOVLW       0X09
    MOVWF       LATCH2
    MOVWF       TOTLATCH2
    MOVWF       PORTA
    BSF         STATUS,RP0
    BSF         PORTC,4
    GOTO        GETDATA


;-------------------------------------------------------------
; CHECKING THE EUADDRESS 5,6,7,8.
;-------------------------------------------------------------

EU5678

    BCF         STATUS,RP0
    BTFSC       EUADDR,1
    GOTO        EU78
    BTFSC       EUADDR,0
    GOTO        EU6
```

```
;-----------------------------------------
;EUADDRESS #5
;-----------------------------------------

    BSF         STATUS,RP0
    BCF         PORTC,1
    BCF         STATUS,RP0
    MOVLW       0X08
    MOVWF       PORTA
    MOVWF       LATCH1
    MOVWF       TOTLATCH1
    BSF         STATUS,RP0
    BSF         PORTC,1
    BCF         PORTC,4
    BCF         STATUS,RP0
    MOVLW       0X0D
    MOVWF       PORTA
    MOVWF       LATCH2
    MOVWF       TOTLATCH2
    BSF         STATUS,RP0
    BSF         PORTC,4
    GOTO        GETDATA

;-----------------------------------------
;EUADDRESS #6
;-----------------------------------------

EU6

    BSF         STATUS,RP0
    BCF         PORTC,1
    BCF         STATUS,RP0
    MOVLW       0X1A
    MOVWF       PORTA
    MOVWF       LATCH1
    MOVWF       TOTLATCH1
    BSF         STATUS,RP0
    BSF         PORTC,1
    BCF         PORTC,4
    BCF         STATUS,RP0
    MOVLW       0X10
```

```
    MOVWF   PORTA
    MOVWF   LATCH2
    MOVWF   TOTLATCH2
    BSF     STATUS,RP0
    BSF     PORTC,4
    GOTO    GETDATA

;-----------------------------------------------------------------
; CHECKING THE EUADDRESS 7,8.
;-----------------------------------------------------------------

EU78

    BTFSC   EUADDR,0
    GOTO    EU8

;---------------------------------------
;EUADDRESS #7
;---------------------------------------


    BSF     STATUS,RP0
    BCF     PORTC,1
    BCF     STATUS,RP0
    MOVLW   0X2C
    MOVWF   LATCH1
    MOVWF   TOTLATCH1
    MOVWF   PORTA
    BSF     STATUS,RP0
    BSF     PORTC,1
    BCF     PORTC,4
    BCF     STATUS,RP0
    MOVLW   0X13
    MOVWF   LATCH2
    MOVWF   TOTLATCH2
    MOVWF   PORTA
    BSF     STATUS,RP0
    BSF     PORTC,4
    GOTO    GETDATA
```

```
;----------------------------------------
;EUADDRESS #8
;----------------------------------------

EU8

    BSF         STATUS,RP0
    BCF         PORTC,1
    BCF         STATUS,RP0
    MOVLW       0X3E
    MOVWF       LATCH1
    MOVWF       TOTLATCH1
    MOVWF       PORTA
    BSF         STATUS,RP0
    BSF         PORTC,1
    BCF         PORTC,4
    BCF         STATUS,RP0
    MOVLW       0X16
    MOVWF       LATCH2
    MOVWF       TOTLATCH2
    MOVWF       PORTA
    BSF         STATUS,RP0
    BSF         PORTC,4
    GOTO        GETDATA

;-----------------------------------------------------------------------------
; GETTING THE DATA
; REGISTERS USED:
   MONITOREG,RUNREG,MCOUNT,ADDRCOUNT=0X02
;-----------------------------------------------------------------------------

GETDATA

    BCF         STATUS,RP0
    BSF         PIR1,5
    BTFSS       PIR1,5
    GOTO        GETDATA
    BTFSC       RCSTA,1
    CALL        SUB2
    BTFSC       RCSTA,2
    BCF         RCSTA,2
    MOVLW       0X01
```

```
   MOVWF    RCREG
   MOVF     RCREG,0
   MOVWF    TEMP1
   INCF     MCOUNT,1
   BTFSC    TEMP1,7
   INCF     RUNREG,1
   INCF     MONITOREG,1
   MOVLW    0X02
   MOVWF    ADDRCOUNT
   BTFSC    TEMP1,0
   GOTO     DNEPS
   BTFSC    TEMP1,1
   GOTO     DSTHICK
   BTFSC    TEMP1,2
   GOTO     DLTHICK
   BTFSC    TEMP1,3
   GOTO     DLTHIN
   MOVLW    0X0E
   MOVWF    DATACOUNT
   CALL     LATCHCHECK

START
   CALL     RETRIEVEDATA

;-------------------------------------------------------------------------------
; MOVING THE ADDRESS TO THE LOCATION OF THE DATA -TOTAL
; REGISTERS USED:
    ADDRCOUNT,TOTLATCH1,TOTLATCH2,LATCH1,LATCH2,DATACO
UNT
;-------------------------------------------------------------------------------

    BCF      STATUS,RP0
    MOVF     TOTLATCH1,0
    MOVWF    LATCH1
    MOVF     TOTLATCH2,0
    MOVWF    LATCH2
    MOVLW    0X05
    MOVWF    ADDRCOUNT
    MOVLW    0X14
    MOVWF    DATACOUNT
    CALL     LATCHCHECK
```

```
MOVF    LATCH1,0
MOVWF   TOTLATCH1
MOVF    LATCH2,0
MOVWF   TOTLATCH2
CALL    RETRIEVEDATA
```

;----------

; SETTING THE ADDRESS OF THE NEXT DRUM [hundreth digit's location of neps]
; REGISTERS USED:LATCH1, LATCH2, TOTLATCH1,TOTLATCH2,DATACOUNT,DATA10COUNT.
; GETTING 10 DATAS

;----------

```
BCF     STATUS,RP0
MOVF    TOTLATCH1,0
MOVWF   LATCH1
MOVF    TOTLATCH2,0
MOVWF   LATCH2
MOVLW   0X01
MOVWF   DATACOUNT
CALL    LATCHCHECK
MOVF    LATCH1,0
MOVWF   TOTLATCH1
MOVF    LATCH2,0
MOVWF   TOTLATCH2
DECFSZ  DATA10COUNT,1
GOTO    GETDATA
```

;----------
;DISCARDING THE UNWANTED DATA
;----------

SUB

```
BSF     PIR1,5
BTFSS   PIR1,5
GOTO    SUB
BTFSC   RCSTA,1
```

```
CALL    SUB2
BTFSC   RCSTA,2
BCF     RCSTA,2
MOVLW   0X81
MOVWF   RCREG
MOVF    RCREG,0
INCF    MCOUNT,1
BTFSS   MCOUNT,5
GOTO    SUB
BTFSS   MCOUNT,2
GOTO    SUB
BTFSS   MCOUNT,0
GOTO    SUB
MOVF    PORTC,0
MOVWF   PRINTERON
BTFSS   PRINTERON,0
GOTO    MAIN
GOTO    PRINTER

;-------------------------------------------------------------------------------

;MOVE DATA FROM MEMORY TO PIC & INCREMENT THE DATA

;-------------------------------------------------------------------------------

RETRIEVEDATA

    BSF     STATUS,RP0
    MOVLW   0XFF
    MOVWF   PORTB
    BSF     PORTC,5
    BCF     STATUS,RP0
    MOVF    PORTB,0
    BSF     STATUS,RP0
    BCF     PORTC,5
    BCF     STATUS,RP0
    MOVWF   LDATA
    BTFSS   LDATA,3
    GOTO    DATAINCR
    BTFSC   LDATA,1
```

```
    GOTO        ADDDECR

DATAINCR

    BCF         STATUS,RP0
    INCF        LDATA,1
    GOTO        DATASTORE


; Initialising LDATA to 0 and decrementing the address

ADDDECR
;*********************************
;
; writing 0x30 to first location
;PORTB=O/P & then as I/P
;*********************************
;
    BSF         STATUS,RP0
    CLRF        PORTB
    BCF         STATUS,RP0
    MOVLW       0X30
    MOVWF       LDATA
    BSF         STATUS,RP0
    BCF         PORTC,5
    BCF         STATUS,RP0
    MOVWF       PORTB


;#################################
;decrementing the address
;decr LATCH1 & LATCH2
;#################################

    DECF        LATCH1,1
    BTFSS       LATCH1,5
    GOTO        READM
    BTFSS       LATCH1,4
    GOTO        READM
    BTFSS       LATCH1,3
    GOTO        READM
    BTFSS       LATCH1,2
```

```
        GOTO    READM
        BTFSS   LATCH1,1
        GOTO    READM
        BTFSC   LATCH1,0
        DECF    LATCH2,1

READM

        MOVF    LATCH1,0
        BSF     STATUS,RP0
        BCF     PORTC,1
        BCF     STATUS,RP0
        MOVWF   PORTA
        BSF     STATUS,RP0
        BSF     PORTC,1
        BCF     STATUS,RP0
        MOVF    LATCH2,0
        BSF     STATUS,RP0
        BCF     PORTC,4
        BCF     STATUS,RP0
        MOVWF   PORTA
        BSF     STATUS,RP0
        BSF     PORTC,4
        BCF     STATUS,RP0

        DECFSZ  ADDRCOUNT,1
        GOTO    RETRIEVEDATA
        BSF     STATUS,RP0
        MOVLW   0X00
        MOVWF   PORTB
        BCF     STATUS,RP0
        MOVLW   0X30
        MOVWF   PORTB

DATASTORE

        BCF     STATUS,RP0
        MOVF    LATCH1,0
        BSF     STATUS,RP0
        BCF     PORTC,1
        BCF     STATUS,RP0
```

```
MOVWF   PORTA
BSF     STATUS,RP0
BSF     PORTC,1
BCF     STATUS,RP0
MOVF    LATCH2,0
BSF     STATUS,RP0
BCF     PORTC,4
BCF     STATUS,RP0
MOVWF   PORTA
BSF     STATUS,RP0
BSF     PORTC,4
BSF     STATUS,RP0
CLRF    PORTB
BSF     STATUS,RP0
BCF     PORTC,5
BCF     STATUS,RP0
MOVF    LDATA,0
MOVWF   PORTB
BSF     STATUS,RP0
BSF     PORTC,5
BCF     STATUS,RP0
RETURN


;.....
;MOVING DATACOUNT VALUES TO THE DATAS
;.....


STORE

   CALL     LATCHCHECK
   MOVLW    #0X30
   MOVWF    MEMINIT
   MOVWF    PORTB
   GOTO     ML1

DNEPS

   BCF      STATUS,RP0
```

```
    MOVLW   0X02
    MOVWF   DATACOUNT
    CALL    LATCHCHECK
    GOTO    START

DSTHICK

    BCF     STATUS,RP0
    MOVLW   0X05
    MOVWF   DATACOUNT
    CALL    LATCHCHECK
    GOTO    START


DLTHICK

    BCF     STATUS,RP0
    MOVLW   0X03
    MOVWF   DATACOUNT
    CALL    LATCHCHECK
    GOTO    START

DLTHIN

    BCF     STATUS,RP0
    MOVLW   0X0B
    MOVWF   DATACOUNT
    CALL    LATCHCHECK
    GOTO    START


;*******************************************************************************
;
;***********
;SUBROUTINE FOR INCREMENTING THE ADRESSES AS PER THE
DATACOUNT VALUE
;*******************************************************************************
;
;***********

LATCHCHECK

    BCF     STATUS,RP0
```

```
    BTFSS   LATCH1,5
    GOTO    INC
    BTFSS   LATCH1,4
    GOTO    INC
    BTFSS   LATCH1,3
    GOTO    INC
    BTFSS   LATCH1,2
    GOTO    INC
    BTFSS   LATCH1,1
    GOTO    INC
    BTFSC   LATCH1,0
    GOTO    L2INC

INC

    BCF     STATUS,RP0
    INCF    LATCH1,0
    MOVWF   LATCH1
    BSF     STATUS,RP0
    BCF     PORTC,1
    BCF     STATUS,RP0
    MOVWF   PORTA
    BSF     STATUS,RP0
    BSF     PORTC,1
    BCF     PORTC,4
    BCF     STATUS,RP0
    MOVF    LATCH2,0
    MOVWF   PORTA
    BSF     STATUS,RP0
    BSF     PORTC,4
    BCF     STATUS,RP0
    DECFSZ  DATACOUNT,1
    GOTO    LATCHCHECK
    RETURN


L2INC

    BCF     STATUS,RP0
    CLRF    LATCH1
    MOVF    LATCH1,0
```

```
        BSF     STATUS,RP0
        BCF     PORTC,1
        BCF     STATUS,RP0
        MOVWF   PORTA
        BSF     STATUS,RP0
        BSF     PORTC,1
        BCF     STATUS,RP0
        INCF    LATCH2,0
        MOVWF   LATCH2
        BSF     STATUS,RP0
        BCF     PORTC,4
        BCF     STATUS,RP0
        MOVWF   PORTA
        BSF     STATUS,RP0
        BSF     PORTC,4
        BCF     STATUS,RP0
        DECFSZ  DATACOUNT,1
        GOTO    LATCHCHECK
        RETURN


SUB2

        BCF     STATUS,RP0
        BCF     RCSTA,4
        BSF     RCSTA,4
        RETURN


;*****************************************************************************
;
;************
; PRINTING THE DATA STORED
; SENDING THE STROBE & CHECKING BUSY
;*****************************************************************************
;
;************

PRINTER

        BSF     STATUS,RP0
        MOVLW   0X00
```

```
        MOVWF   PORTB
        BCF     STATUS,RP0
        MOVLW   0X7F
        MOVWF   DELAYCOUNT
        BSF     STATUS,RP0
        BCF     PORTC,3
        BCF     STATUS,RP0
DELAYDECFSZ     DELAYCOUNT,1
        GOTO    DELAY
        BSF     STATUS,RP0
        BSF     PORTC,3
        BCF     STATUS,RP0
        CALL    TESTBUSY

        MOVLW   0X0A
        MOVWF   COCOUNT
        MOVLW   0X30
        MOVWF   BIT4
        MOVLW   0X00
        MOVWF   TEMLATCH1
        MOVWF   TEMLATCH2
        MOVWF   PLATCH1
        MOVWF   PLATCH2

;_____________________________________
;PRINTING DRUM NUMBER & DR.NO:
;_____________________________________

DRUMNO

        MOVLW   0X30
        MOVWF   NUMBER
        MOVLW   0X02
        CALL    TESTBUSY
        MOVWF   PORTB
        CALL    CARRIAGE
        CALL    LINEFEED
        CALL    LINEFEED
        CALL    SPACE3
        CALL    SPACE3
        CALL    SPACE3
        MOVLW   0X44
```

```
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X52
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X3E
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X4E
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X4F
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X3A
    CALL    TESTBUSY
    MOVWF   PORTB
    CALL    SPACE3

NO

    CALL    SPACE3
    INCF    NUMBER,1
    BTFSS   NUMBER,3
    GOTO    NOCHANGE

;_____________________________________
;Check for 0111010[0A]
;_____________________________________

    BTFSS   NUMBER,1
    GOTO    NOCHANGE
    INCF    BIT4,1
    MOVF    BIT4,0
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X30
    MOVWF   NUMBER
    CALL    TESTBUSY
    MOVWF   PORTB
    CALL    SPACE3
    DECFSZ  COCOUNT,1
```

```
;_____________________________________
;PRINTING NEPS
;_____________________________________

PNEPS

    CALL    CARRIAGE
    MOVLW   0X4E
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X45
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X50
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X53
    CALL    TESTBUSY
    MOVWF   PORTB
    CALL    SPACE3
    CALL    SPACE1
    CALL    SPACE1

;_____________________________________
;PRINTING THE DATA FROM MEMORY
;_____________________________________
;

    MOVLW   0X00
    MOVWF   TEMLATCH1
    MOVWF   TEMLATCH2
    MOVWF   PLATCH1
    MOVWF   PLATCH2
    MOVLW   0X06
    MOVWF   ROCOUNT
    MOVLW   0X08
    MOVWF   SETCOUNT
```

```
INAD

    MOVLW     0X0A
    MOVWF     COCOUNT

INAD0

    MOVF      PLATCH1,0
    MOVWF     LATCH1
    MOVF      PLATCH2,0
    MOVWF     LATCH2
    MOVLW     0X03
    MOVWF     COUNT
;_____________________________________
; CHANGING THE ADDRESS
; Sending the address to memory
;_____________________________________

LOCATION

    BSF       STATUS,RP0
    BCF       PORTC,1
    BCF       STATUS,RP0
    MOVF      LATCH1,0
    MOVWF     PORTA
    BSF       STATUS,RP0
    BSF       PORTC,1
    BCF       PORTC,4
    BCF       STATUS,RP0
    MOVF      LATCH2,0
    MOVWF     PORTA
    BSF       STATUS,RP0
    BSF       PORTC,1

;_____________________________________

; DATA READ FROM MEMORY
; Hundredth,tenth,unit digit of 10 drums
; PORTB Input
;_____________________________________
```

```
    MOVLW   0XFF
    MOVWF   PORTB
    BCF     STATUS,RP0
    CALL    TESTBUSY
    MOVF    PORTB,1
    DECFSZ  COUNT,1
    GOTO    AD1INCR
    CALL    SPACE1
    CALL    SPACE1
    DECFSZ  COCOUNT,1
    GOTO    AD21INCR
    MOVLW   0X0A
    MOVWF   COCOUNT
    CALL    P2LATINC1
;_____________________________
;Portb as output
;_____________________________

    BSF     STATUS,RP0
    CLRF    PORTB
    BCF     STATUS,RP0
    DECF    ROCOUNT,1   ; NEXT DATA OF DIFF TYPE
    BTFSS   ROCOUNT,2
    GOTO    CHECK1
    BTFSC   ROCOUNT,0
    GOTO    PSTK

;_____________________________
; PRINTING LTK[long thick]
;_____________________________

PLTK

    CALL    CARRIAGE
    MOVLW   0X4C
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X54
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X4B
```

```
        CALL    TESTBUSY
        MOVWF   PORTB
        CALL    SPACE3
        CALL    SPACE3
        GOTO    INAD0

AD1INCR

        MOVLW   0X01
        MOVWF   DATACOUNT
        CALL    LATCHCHECK
        GOTO    LOCATION

AD21INCR

        MOVF    PLATCH1,0
        MOVWF   LATCH1
        MOVF    PLATCH2,0
        MOVWF   LATCH2
        MOVLW   0X15
        MOVWF   DATACOUNT
        CALL    LATCHCHECK
        MOVF    LATCH1,0
        MOVWF   PLATCH1
        MOVF    LATCH2,0
        MOVWF   PLATCH2
        DECF    ROCOUNT,0
        BTFSC   STATUS,2
        GOTO    INAD0
        CALL    SPACE3
        GOTO    INAD0


NOCHANGE:

        CALL    SPACE1
        MOVF    NUMBER,0
        CALL    TESTBUSY
        MOVWF   PORTB
        CALL    SPACE3
        CALL    SPACE1
```

```
    CALL        SPACE1
    DECFSZ      COCOUNT,1
    GOTO        NO
    GOTO        PNEPS

P2LATINC1

    MOVF        TEMLATCH1,0
    MOVWF       LATCH1
    MOVF        TEMLATCH2,0
    MOVWF       LATCH2
    MOVLW       0X03
    MOVWF       DATACOUNT
    CALL        LATCHCHECK
    MOVF        LATCH1,0
    MOVWF       PLATCH1
    MOVWF       TEMLATCH1
    MOVF        LATCH2,0
    MOVWF       PLATCH2
    MOVWF       TEMLATCH2
    RETURN

CHECK1

    BTFSS       ROCOUNT,1
    GOTO        CHECK2
    BTFSS       ROCOUNT,0
    GOTO        PCOUNT

;_____________________________________
;PTINTING LTN[long thin]
;_____________________________________

PLTN

    CALL        CARRIAGE
    MOVLW       0X4C
    CALL        TESTBUSY
    MOVWF       PORTB
    MOVLW       0X54
    CALL        TESTBUSY
```

```
    MOVWF   PORTB
    MOVLW   0X4E
    CALL    TESTBUSY
    MOVWF   PORTB
    CALL    SPACE3
    CALL    SPACE3
    GOTO    INAD0

;_____________________________________
;PTINTING STK[short thick]
;_____________________________________

PSTK

    CALL    CARRIAGE
    MOVLW   0X4C
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X54
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X4B
    CALL    TESTBUSY
    MOVWF   PORTB
    CALL    SPACE3
    CALL    SPACE3
    GOTO    INAD0

;_____________________________________
;PTINTING COUNT
;_____________________________________

PCOUNT

    CALL    CARRIAGE
    MOVLW   0X4C
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X54
    CALL    TESTBUSY
```

```
    MOVWF   PORTB
    MOVLW   0X4B
    CALL    TESTBUSY
    MOVWF   PORTB
    CALL    SPACE3
    CALL    SPACE3
    GOTO    INAD0
;_____________________________________
;PTINTING TOTAL
;_____________________________________

PTOTAL

    CALL    CARRIAGE
    MOVLW   0X54
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X4F
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X41
    CALL    TESTBUSY
    MOVWF   PORTB
    MOVLW   0X4C
    CALL    TESTBUSY
    MOVWF   PORTB
    CALL    SPACE3
    CALL    SPACE1
    GOTO    INAD0

    CHECK2

    BTFSC   ROCOUNT,0
    GOTO     PTOTAL

PDRUM

    CALL    CARRIAGE
    CALL    LINEFEED
    DECFSZ  SETCOUNT,1
    GOTO    NEXTSET
```

```
    MOVLW   0X03
    CALL    TESTBUSY
    MOVWF   PORTB
    GOTO    MAIN1


NEXTSET

    MOVF    PLATCH1,0
    MOVWF   LATCH1
    MOVF    PLATCH2,0
    MOVWF   LATCH2
    MOVLW   0X01
    MOVWF   DATACOUNT
    CALL    LATCHCHECK
    MOVF    LATCH1,0
    MOVWF   TEMLATCH1
    MOVWF   PLATCH1
    MOVF    LATCH2,0
    MOVWF   PLATCH2
    MOVWF   TEMLATCH2
    GOTO    DRUMNO
;_____________________________________
; TESTING BUSY LINE
;_____________________________________

TESTBUSY

    BTFSC   PORTC,2
    GOTO    TESTBUSY
    RETURN

;_____________________________________
; LINEFEED TO THE PRINTER
;_____________________________________

LINEFEED

    MOVLW   0X0A
    CALL    TESTBUSY
```

```
    MOVWF   PORTB
    RETURN

;_____________________________________
; CARRIAGE RETURN TO THE PRINTER
;_____________________________________

CARRIAGE

    MOVLW   0X0D
    CALL    TESTBUSY
    MOVWF   PORTB
    CALL    SPACE3
    CALL    SPACE3
    CALL    SPACE3
    RETURN
;_____________________________________
; LEAVING 3 BLANKSPACES
;_____________________________________

SPACE3

    MOVLW   0X03
    MOVWF   SPACECOUNT3
SP
    MOVLW   0X20
    CALL    TESTBUSY
    MOVWF   PORTB
    DECFSZ  SPACECOUNT3,1
    GOTO    SP
    RETURN
;_____________________________________
; LEAVING SINGLE SPACE
;_____________________________________

SPACE1

    MOVLW   0X20
    CALL    TESTBUSY
    MOVWF   PORTB
    RETURN
    GOTO $
```

Chapter -

CONCLUSION

CONCLUSION

The data interfacing to printer is done using PIC Microcontroller .The data from the yarn clearing machine is tapped serially and received by PIC via RS-232 cable.

Each type of yarn obtained as input every time is incremented accordingly and is stored in an external RAM.Thus of 6 types of yarn present,each type of yarn obtained at every input is known by the incremented value in memory.When the user requires a hard copy of the data obtained, a printer is connected. After necessary handshaking signals, printout of data in memory is obtained.

This project provides necessary data in memory for calculating the production efficiency of each day.The type of yarn obtaind for all the three shifts ,the production efficiency etc is calculated using the data stored.A hard copy in required format as in the control unit of M2010 is also obtained.

Presently the project is designed only for parallel p printing.For further developments an option may be provided for either serial or parallel printing.

BIBLIOGRAPH

BIBLIOGRAPHY:

- Morris Mano, M. (2000) DIGITAL DESIGN, 2nd Edition, Prentice Hall, USA.
- Ramesh Gaonkar, S. (2000) MICROPROCESSOR ARCHITECTURE, PROGRAMMING, AND APPLICATIONS WITH THE 8085, 4th Edition, Pena ram International, India.
- Douglas Hall ,V. (1991) MICROPROCESSORS AND INTERFACING , Tata Mc Graw Hill , USA
- John Peatman ,(2001) PROGRAMMING WITH PIC MICROCONTROLLERS , Pearson Education, Singapore
- Power Supply Circuits, Texas Instruments Data Book,1996
- Flash Memory , Volume 1, Intel Manual,1997
- Linear Products, Burr Brown IC Data Book,1995
- Printer Manual - Epson LX 300

WEBSITES:

www.epson.com
www.microchip.com
www.google.com
www.printeronnet.com
www.made-it.com

APPENDIX

1.0 DEVICE OVERVIEW

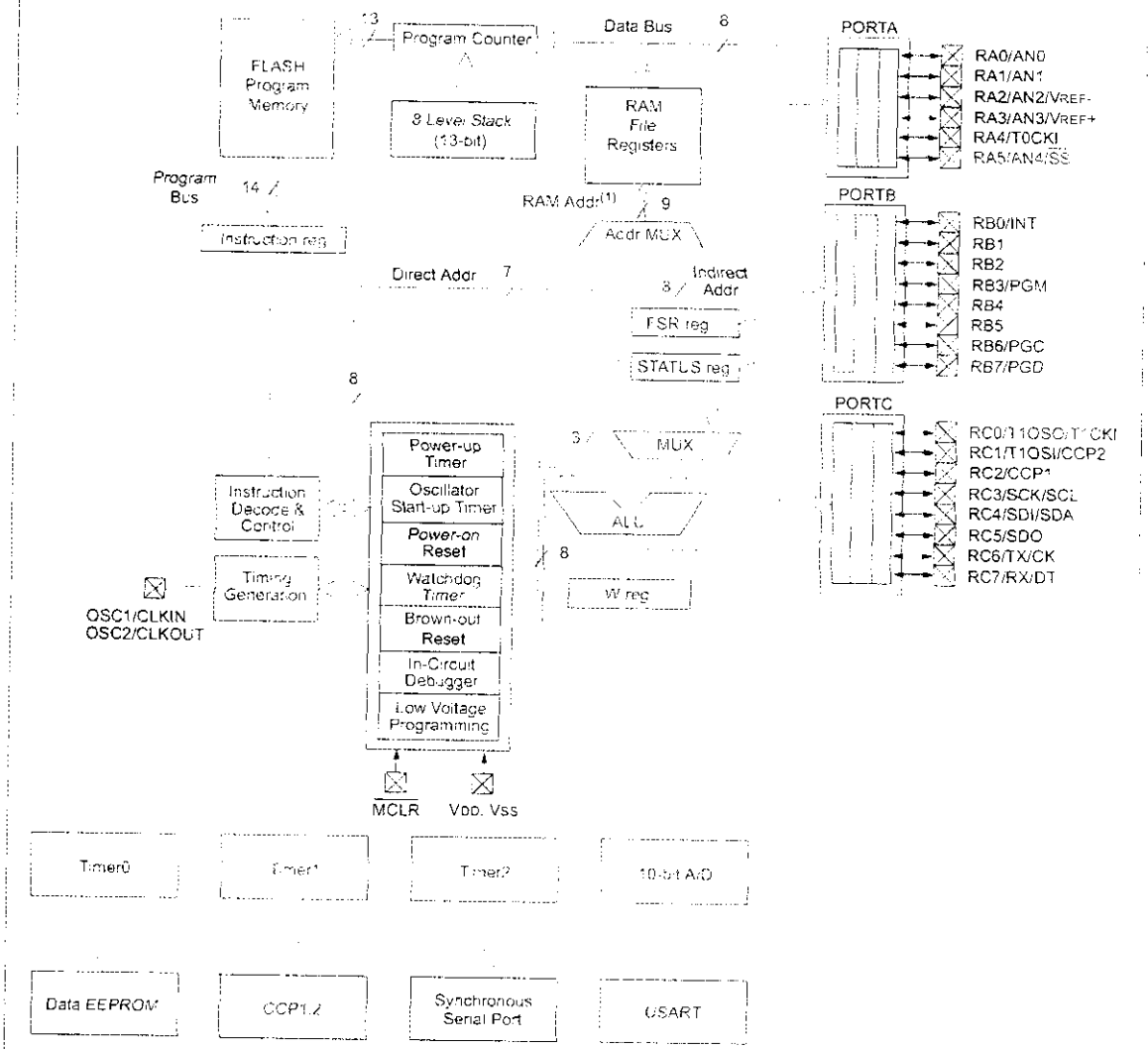
This document contains device specific information. Additional information may be found in the PICmicro™ Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended *reading for a better understanding of the device* architecture and operation of the peripheral modules.

There are four devices (PIC16F873, PIC16F874, PIC16F876 and PIC16F877) covered by this data sheet. The PIC16F876/873 devices come in 28-pin packages and the PIC16F877/874 devices come in 40-pin packages. The Parallel Slave Port is not implemented on the 28-pin devices.

The following device block diagrams are sorted by pin number; 28-pin for Figure 1-1 and 40-pin for Figure 1-2. The 28-pin and 40-pin pinouts are listed in Table 1-1 and Table 1-2, respectively.

FIGURE 1-1: PIC16F873 AND PIC16F876 BLOCK DIAGRAM

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F873	4K	192 Bytes	128 Bytes
PIC16F876	8K	368 Bytes	256 Bytes



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

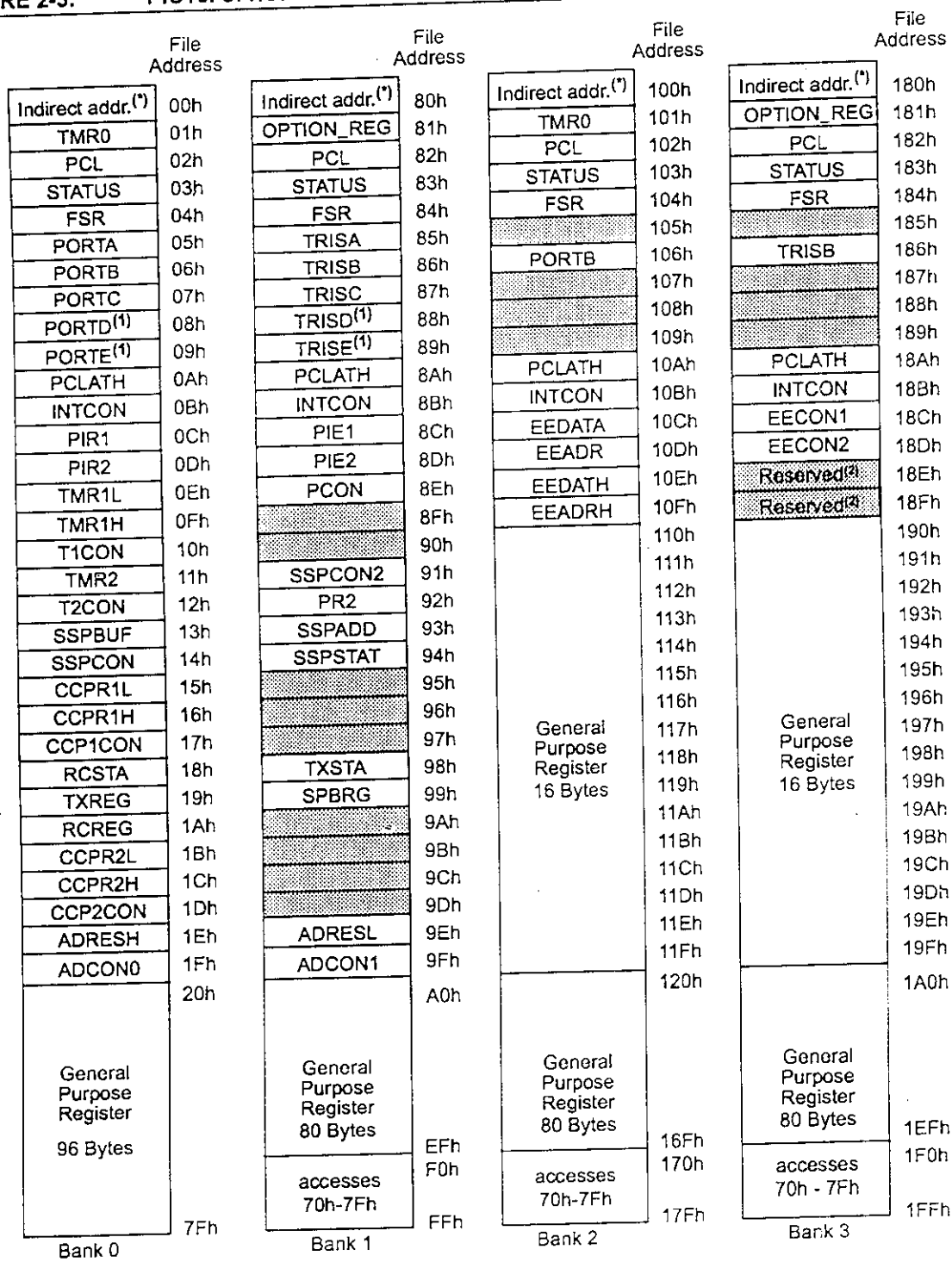
TABLE 1-1: PIC16F873 AND PIC16F876 PINOUT DESCRIPTION

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS(3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP	1	1	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	2	I/O	TTL	RA0 can also be analog input0.
RA1/AN1	3	3	I/O	TTL	RA1 can also be analog input1.
RA2/AN2/VREF-	4	4	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage.
RA3/AN3/VREF+	5	5	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage.
RA4/T0CKI	6	6	I/O	ST	*RA4 can also be the clock input to the Timer0* module. Output is open drain type.
RA5/$\overline{SS}$/AN4	7	7	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	21	21	I/O	TTL/ST(1)	RB0 can also be the external interrupt pin.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL	RB3 can also be the low voltage programming input
RB4	25	25	I/O	TTL	Interrupt-on-change pin.
RB5	26	26	I/O	TTL	Interrupt-on-change pin.
RB6/PGC	27	27	I/O	TTL/ST(2)	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	28	28	I/O	TTL/ST(2)	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.
					PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	11	11	I/O	ST	RC0 can also be the Timer1 oscillator output or Timer1 clock input.
RC1/T1OSI/CCP2	12	12	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	13	13	I/O	ST	*RC2 can also be the Capture1 input/Compare1 output/* PWM1 output.
RC3/SCK/SCL	14	14	I/O	ST	*RC3 can also be the synchronous serial clock input/output* for both SPI and I^2C modes
RC4/SDI/SDA	15	15	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	16	16	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	17	17	I/O	ST	RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	18	18	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.
VSS	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

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 the 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 in RC oscillator mode and a CMOS input otherwise.

FIGURE 2-3: PIC16F877/876 REGISTER FILE MAP



▒ 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.

2.2 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.

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

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.

Note: EEPROM Data Memory description can be found in Section 4.0 of this data sheet.

2.2.1 GENERAL PURPOSE REGISTER FILE

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

2.0 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 and is detailed in this section. The EEPROM data memory block is detailed in Section 4.0.

Additional information on device memory may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

2.1 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, and the PIC16F873/874 devices have 4K x 14. 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

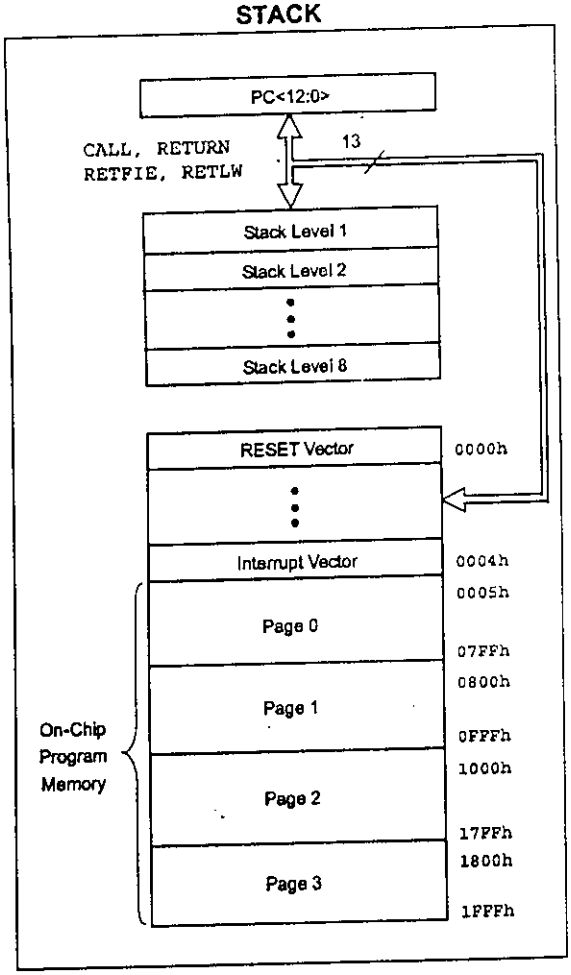
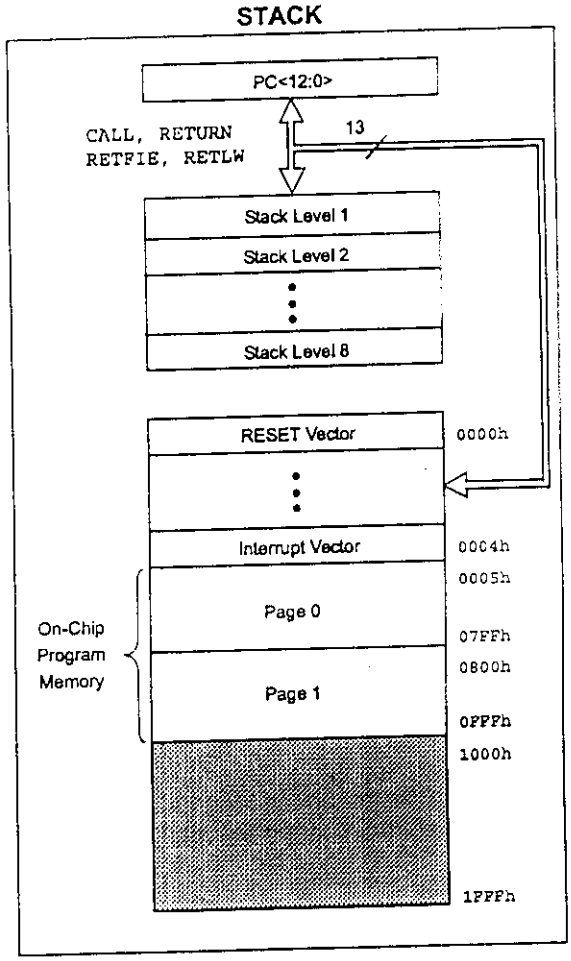


FIGURE 2-2: PIC16F874/873 PROGRAM MEMORY MAP AND STACK



2.2.2 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(3)	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	$\overline{RBPU}$	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	$\overline{TO}$	$\overline{PD}$	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	---	---	---	---	---	---	$\overline{POR}$	$\overline{BOR}$	---- --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/$\overline{A}$	P	S	R/$\overline{W}$	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	Details on page:
Bank 2											
100h(3)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
101h	TMR0	Timer0 Module Register								xxxx xxxx	47
102h(3)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	26
103h(3)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	18
104h(3)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
105h	---	Unimplemented								---	---
106h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	31
107h	---	Unimplemented								---	---
108h	---	Unimplemented								---	---
109h	---	Unimplemented								---	---
10Ah(1,3)	PCLATH	---	---	---	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
10Bh(3)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
10Ch	EEDATA	EEPROM Data Register Low Byte								xxxx xxxx	41
10Dh	EEADR	EEPROM Address Register Low Byte								xxxx xxxx	41
10Eh	EEDATH	---	---	EEPROM Data Register High Byte						xxxx xxxx	41
10Fh	EEADRH	---	---	---	EEPROM Address Register High Byte					xxxx xxxx	41
Bank 3											
180h(3)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	19
182h(3)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26
183h(3)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	18
184h(3)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
185h	---	Unimplemented								---	---
186h	TRISB	PORTB Data Direction Register								1111 1111	31
187h	---	Unimplemented								---	---
188h	---	Unimplemented								---	---
189h	---	Unimplemented								---	---
18Ah(1,3)	PCLATH	---	---	---	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
18Bh(3)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
18Ch	EECON1	EEPGD	---	---	---	WRERR	WREN	WR	RD	x--- x000	41, 42
18Dh	EECON2	EEPROM Control Register2 (not a physical register)								---- ----	41
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.

2.2.2.1 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 $\overline{TO}$ and $\overline{PD}$ 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 uluu` (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 **$\overline{TO}$:** Time-out bit
1 = After power-up, `CLRWDT` instruction, or `SLEEP` instruction
0 = A WDT time-out occurred

bit 3 **$\overline{PD}$:** 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/$\overline{\text{borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
(for $\overline{\text{borrow}}$, 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/$\overline{\text{borrow}}$ 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 $\overline{\text{borrow}}$, 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

2.2.2.2 OPTION_REG Register

The OPTION_REG Register is a readable and writable register, which contains various control bits to configure the TMR0 prescaler/WDT postscaler (single assignable register known also as the prescaler), the External INT Interrupt, TMR0 and the weak pull-ups on PORTB.

Note: To achieve a 1:1 prescaler assignment for the TMR0 register, assign the prescaler to the Watchdog Timer.

REGISTER 2-2: OPTION_REG REGISTER (ADDRESS 81h, 181h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

bit 7 **$\overline{\text{RBPU}}$**: PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG**: Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

bit 5 **T0CS**: TMR0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE**: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/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

Note: When using low voltage ICSP programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper operation of the device

2.2.2.3 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

2.2.2.4 PIE1 Register

The PIE1 register contains the individual enable bits for the peripheral interrupts.

Note: Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.

REGISTER 2-4: PIE1 REGISTER (ADDRESS 8Ch)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

bit 7 **PSPIE(1):** Parallel Slave Port Read/Write Interrupt Enable bit
1 = Enables the PSP read/write interrupt
0 = Disables the PSP read/write interrupt

bit 6 **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables the A/D converter interrupt
0 = Disables the A/D converter interrupt

bit 5 **RCIE:** USART Receive Interrupt Enable bit
1 = Enables the USART receive interrupt
0 = Disables the USART receive interrupt

bit 4 **TXIE:** USART Transmit Interrupt Enable bit
1 = Enables the USART transmit interrupt
0 = Disables the USART transmit interrupt

bit 3 **SSPIE:** Synchronous Serial Port Interrupt Enable bit
1 = Enables the SSP interrupt
0 = Disables the SSP interrupt

bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt

bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

Note 1: PSPIE is reserved on PIC16F873/876 devices; always maintain this bit clear.

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

2.2.2.5 PIR1 Register

The PIR1 register contains the individual flag bits for the peripheral 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 bits are clear prior to enabling an interrupt.

REGISTER 2-5: PIR1 REGISTER (ADDRESS 0Ch)

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

bit 7 **PSPIF(1):** Parallel Slave Port Read/Write Interrupt Flag bit
1 = A read or a write operation has taken place (must be cleared in software)
0 = No read or write has occurred

bit 6 **ADIF:** A/D Converter Interrupt Flag bit
1 = An A/D conversion completed
0 = The A/D conversion is not complete

bit 5 **RCIF:** USART Receive Interrupt Flag bit
1 = The USART receive buffer is full
0 = The USART receive buffer is empty

bit 4 **TXIF:** USART Transmit Interrupt Flag bit
1 = The USART transmit buffer is empty
0 = The USART transmit buffer is full

bit 3 **SSPIF:** Synchronous Serial Port (SSP) Interrupt Flag
1 = The SSP interrupt condition has occurred, and must be cleared in software before returning from the Interrupt Service Routine. The conditions that will set this bit are:
- SPI
  - A transmission/reception has taken place.
- I^2C Slave
  - A transmission/reception has taken place.
- I^2C Master
  - A transmission/reception has taken place.
  - The initiated START condition was completed by the SSP module.
  - The initiated STOP condition was completed by the SSP module.
  - The initiated Restart condition was completed by the SSP module.
  - The initiated Acknowledge condition was completed by the SSP module.
  - A START condition occurred while the SSP module was idle (Multi-Master system).
  - A STOP condition occurred while the SSP module was idle (Multi-Master system).

0 = No SSP interrupt condition has occurred.

bit 2 **CCP1IF:** CCP1 Interrupt Flag bit
Capture mode:
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare mode:
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM mode:
Unused in this mode

bit 1 **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit
1 = TMR1 register overflowed (must be cleared in software)
0 = TMR1 register did not overflow

Note 1: PSPIF is reserved on PIC16F873/876 devices; always maintain this bit clear.

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

2.2.2.6 PIE2 Register

The PIE2 register contains the individual enable bits for the CCP2 peripheral interrupt, the SSP bus collision interrupt, and the EEPROM write operation interrupt.

REGISTER 2-6: PIE2 REGISTER (ADDRESS 8Dh)

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
—	Reserved	—	EEIE	BCLIE	—	—	CCP2IE
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6 **Reserved:** Always maintain this bit clear

bit 5 **Unimplemented:** Read as '0'

bit 4 **EEIE:** EEPROM Write Operation Interrupt Enable
1 = Enable EE Write Interrupt
0 = Disable EE Write Interrupt

bit 3 **BCLIE:** Bus Collision Interrupt Enable
1 = Enable Bus Collision Interrupt
0 = Disable Bus Collision Interrupt

bit 2-1 **Unimplemented:** Read as '0'

bit 0 **CCP2IE:** CCP2 Interrupt Enable bit
1 = Enables the CCP2 interrupt
0 = Disables the CCP2 interrupt

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

2.2.2.7 PIR2 Register

The PIR2 register contains the flag bits for the CCP2 interrupt, the SSP bus collision interrupt and the EEPROM write operation interrupt.

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 2-7: PIR2 REGISTER (ADDRESS 0Dh)

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
—	Reserved	—	EEIF	BCLIF	—	—	CCP2IF
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6 **Reserved**: Always maintain this bit clear

bit 5 **Unimplemented:** Read as '0'

bit 4 **EEIF:** EEPROM Write Operation Interrupt Flag bit
1 = The write operation completed (must be cleared in software)
0 = The write operation is not complete or has not been started

bit 3 **BCLIF:** Bus Collision Interrupt Flag bit
1 = A bus collision has occurred in the SSP, when configured for I2C Master mode
0 = No bus collision has occurred

bit 2-1 **Unimplemented:** Read as '0'

bit 0 **CCP2IF:** CCP2 Interrupt Flag bit
Capture mode:
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare mode:
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM mode:
Unused

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

2.5 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

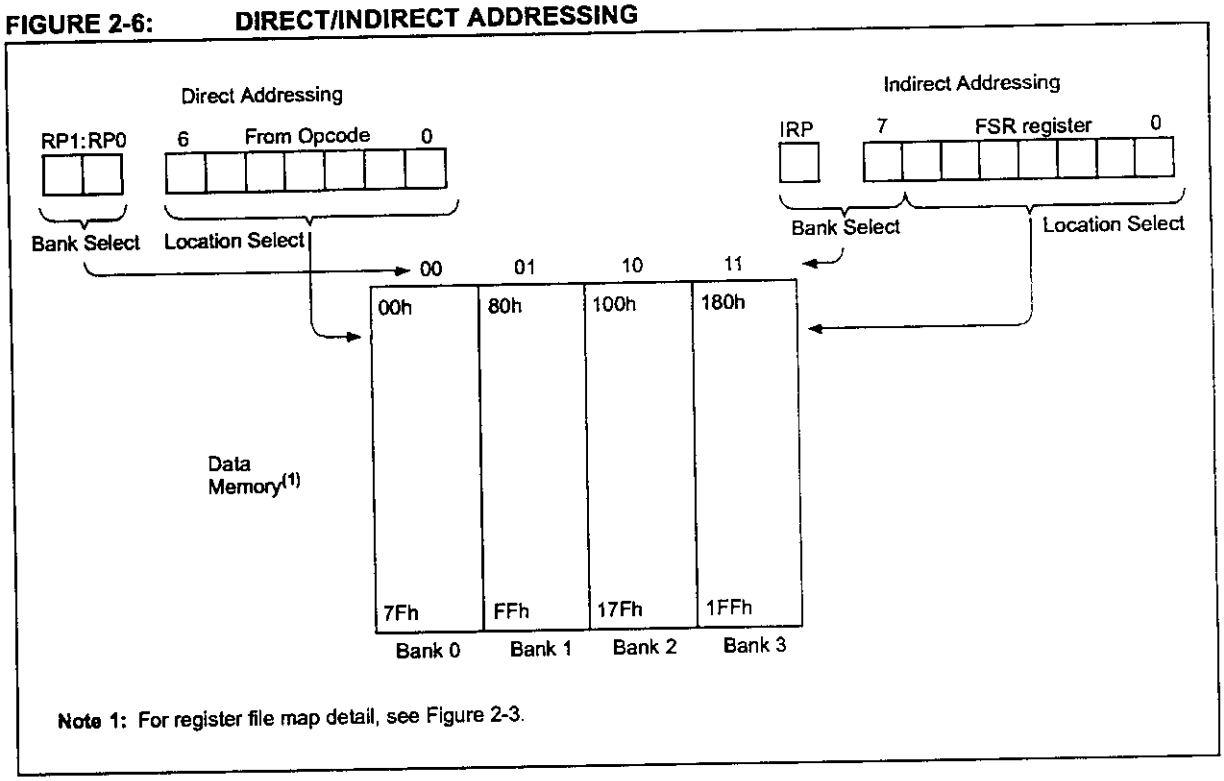
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF register itself, indirectly (FSR = '0') will read 00h. Writing to the INDF register indirectly results in a 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.

3.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Additional information on I/O ports may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

3.1 PORTA and the TRISA Register

PORTA is a 6-bit wide, bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, the value is modified and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other PORTA pins have TTL input levels and full CMOS output drivers.

Other PORTA pins are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

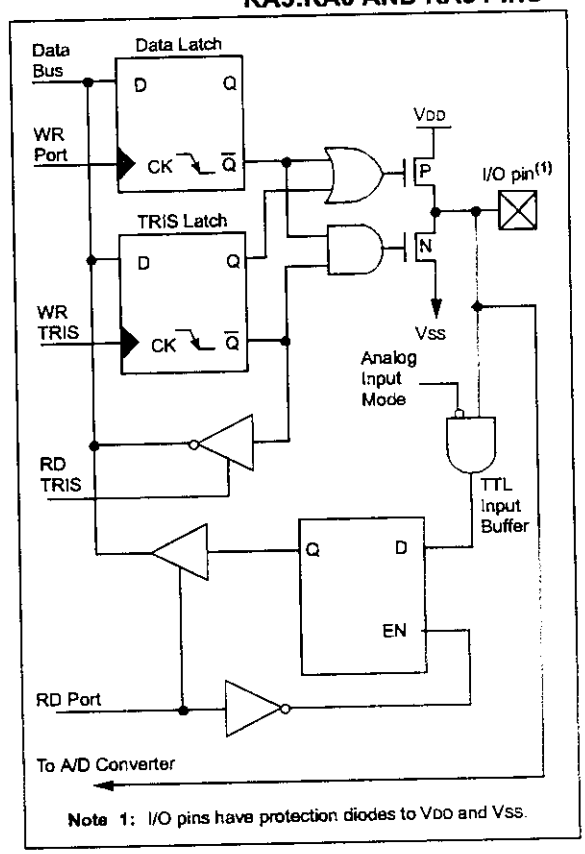
Note: On a Power-on Reset, these pins are configured as analog inputs and read as '0'.

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

EXAMPLE 3-1: INITIALIZING PORTA

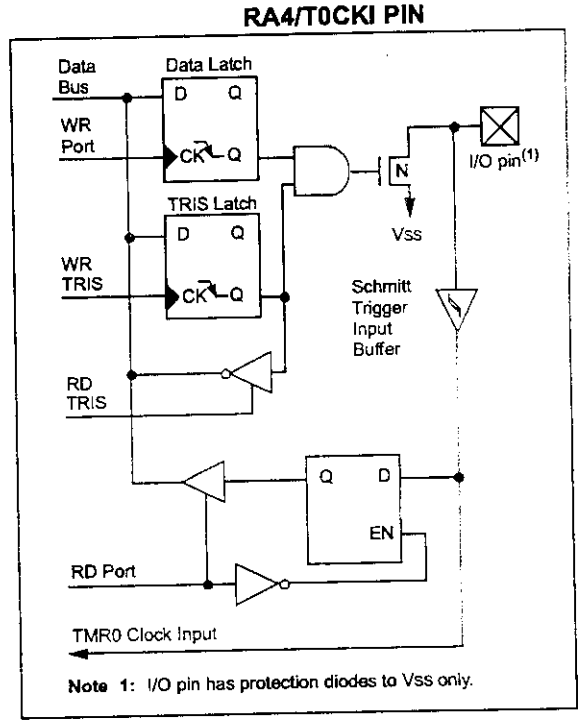
```
BCF     STATUS, RP0   ;
BCF     STATUS, RP1   ; Bank0
CLRF    PORTA         ; Initialize PORTA by
                      ; clearing output
                      ; data latches
BSF     STATUS, RP0   ; Select Bank 1
MOVLW   0x06          ; Configure all pins
MOVWF   ADCON1        ; as digital inputs
MOVLW   0xCF          ; Value used to
                      ; initialize data
                      ; direction
MOVWF   TRISA         ; Set RA<3:0> as inputs
                      ; RA<5:4> as outputs
                      ; TRISA<7:6>are always
                      ; read as '0'.
```

FIGURE 3-1: BLOCK DIAGRAM OF RA3:RA0 AND RA5 PINS



Note 1: I/O pins have protection diodes to VDD and Vss.

FIGURE 3-2: BLOCK DIAGRAM OF RA4/T0CKI PIN



Note 1: I/O pin has protection diodes to Vss only.

TABLE 3-1: PORTA FUNCTIONS

Name	Bit#	Buffer	Function
RA0/AN0	bit0	TTL	Input/output or analog input.
RA1/AN1	bit1	TTL	Input/output or analog input.
RA2/AN2	bit2	TTL	Input/output or analog input.
RA3/AN3/VREF	bit3	TTL	Input/output or analog input or VREF.
RA4/T0CKI	bit4	ST	Input/output or external clock input for Timer0. Output is open drain type.
RA5/$\overline{SS}$/AN4	bit5	TTL	Input/output or slave select input for synchronous serial port or analog input.

Legend: TTL = TTL input, ST = Schmitt Trigger input

TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

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
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000

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

Note: When using the SSP module in SPI Slave mode and $\overline{SS}$ enabled, the A/D converter must be set to one of the following modes, where PCFG3:PCFG0 = 0100, 0101, 011x, 1101, 1110, 1111.

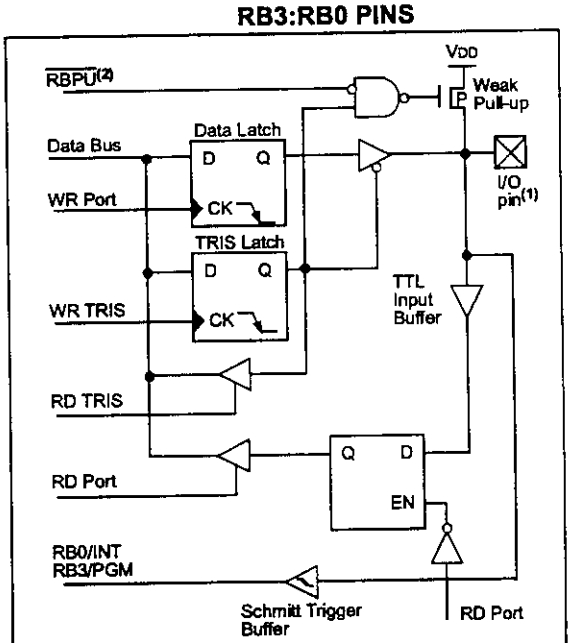
3.2 PORTB and the TRISB Register

PORTB is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., put the contents of the output latch on the selected pin).

Three pins of PORTB are multiplexed with the Low Voltage Programming function: RB3/PGM, RB6/PGC and RB7/PGD. The alternate functions of these pins are described in the Special Features Section.

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit $\overline{\text{RBPU}}$ (OPTION_REG<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

FIGURE 3-3: BLOCK DIAGRAM OF RB3:RB0 PINS



Note 1: I/O pins have diode protection to VDD and VSS.

2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit (OPTION_REG<7>).

Four of the PORTB pins, RB7:RB4, have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt-on-change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RB Port Change Interrupt with flag bit RBIF (INTCON<0>).

This interrupt can wake the device from SLEEP. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

a) Any read or write of PORTB. This will end the mismatch condition.

b) Clear flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.

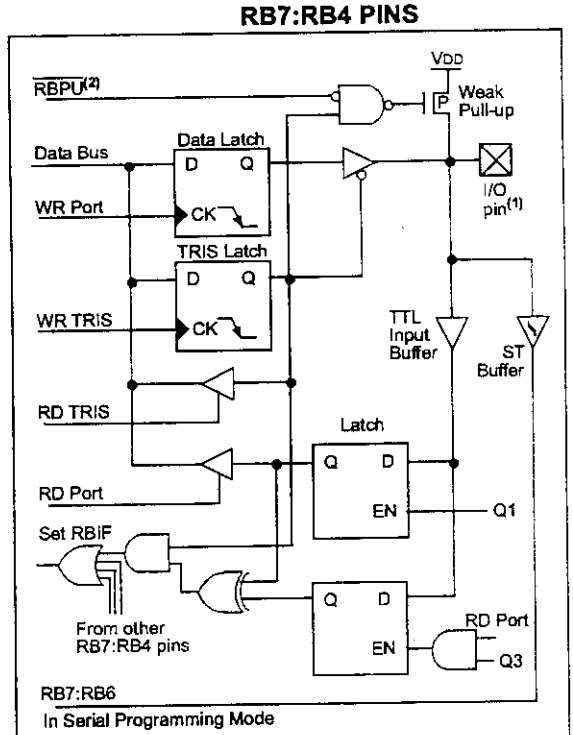
The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

This interrupt-on-mismatch feature, together with software configureable pull-ups on these four pins, allow easy interface to a keypad and make it possible for wake-up on key depression. Refer to the Embedded Control Handbook, *"Implementing Wake-up on Key Strokes"* (AN552).

RB0/INT is an external interrupt input pin and is configured using the INTEDG bit (OPTION_REG<6>).

RB0/INT is discussed in detail in Section 12.10.1.

FIGURE 3-4: BLOCK DIAGRAM OF RB7:RB4 PINS



Note 1: I/O pins have diode protection to VDD and VSS.

2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit (OPTION_REG<7>).

TABLE 3-3: PORTB FUNCTIONS

Name	Bit#	Buffer	Function
RB0/INT	bit0	TTL/ST(1)	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3/PGM(3)	bit3	TTL	Input/output pin or programming pin in LVP mode. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB6/PGC	bit6	TTL/ST(2)	Input/output pin (with interrupt-on-change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming clock.
RB7/PGD	bit7	TTL/ST(2)	Input/output pin (with interrupt-on-change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL input, ST = Schmitt Trigger input

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

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

3: Low Voltage ICSP Programming (LVP) is enabled by default, which disables the RB3 I/O function. LVP must be disabled to enable RB3 as an I/O pin and allow maximum compatibility to the other 28-pin and 40-pin mid-range devices.

TABLE 3-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

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
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h, 186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
81h, 181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

3.3 PORTC and the TRISC Register

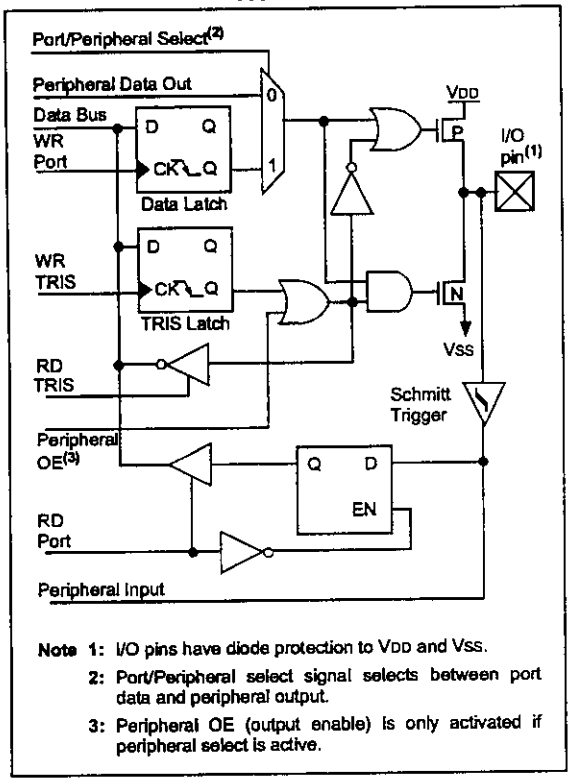
PORTC is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISC. Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., put the contents of the output latch on the selected pin).

PORTC is multiplexed with several peripheral functions (Table 3-5). PORTC pins have Schmitt Trigger input buffers.

When the I^2C module is enabled, the PORTC<4:3> pins can be configured with normal I^2C levels, or with SMBus levels by using the CKE bit (SSPSTAT<6>).

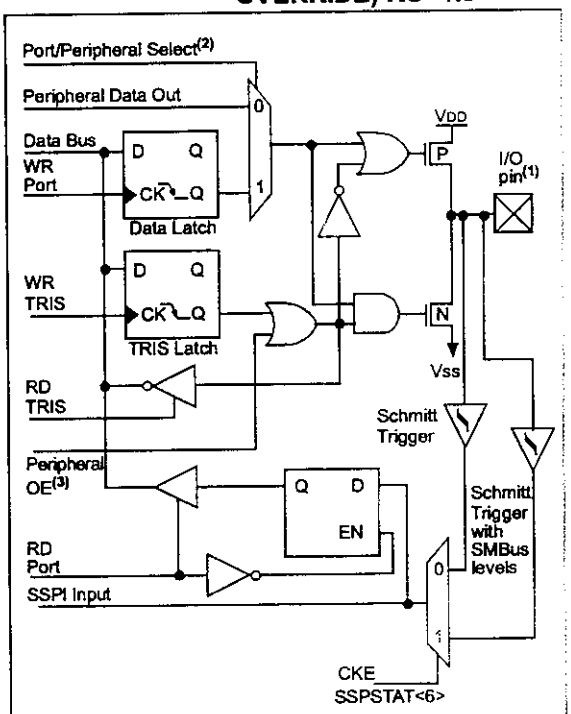
When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. Since the TRIS bit override is in effect while the peripheral is enabled, read-modify-write instructions (`BSF, BCF, XORWF`) with TRISC as destination, should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

FIGURE 3-5: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<2:0>, RC<7:5>



Note 1: I/O pins have diode protection to VDD and Vss.
2: Port/Peripheral select signal selects between port data and peripheral output.
3: Peripheral OE (output enable) is only activated if peripheral select is active.

FIGURE 3-6: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<4:3>



Note 1: I/O pins have diode protection to VDD and Vss.
2: Port/Peripheral select signal selects between port data and peripheral output.
3: Peripheral OE (output enable) is only activated if peripheral select is active.

TABLE 3-5: PORTC FUNCTIONS

Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI/CCP2	bit1	ST	Input/output port pin or Timer1 oscillator input or Capture2 input/ Compare2 output/PWM2 output.
RC2/CCP1	bit2	ST	Input/output port pin or Capture1 input/Compare1 output/ PWM1 output.
RC3/SCK/SCL	bit3	ST	RC3 can also be the synchronous serial clock for both SPI and I^2C modes.
RC4/SDI/SDA	bit4	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	bit5	ST	Input/output port pin or Synchronous Serial Port data output.
RC6/TX/CK	bit6	ST	Input/output port pin or USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	bit7	ST	Input/output port pin or USART Asynchronous Receive or Synchronous Data.

Legend: ST = Schmitt Trigger input

TABLE 3-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

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
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged

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.

5.2 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.

5.3 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 **$\overline{\text{RBPU}}$**

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

Note: To avoid an unintended device RESET, the instruction sequence shown in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023) must be executed when changing the prescaler assignment from Timer0 to the WDT. This sequence must be followed even if the WDT is disabled.

5.0 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

Figure 5-1 is a block diagram of the Timer0 module and the prescaler shared with the WDT.

Additional information on the Timer0 module is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

Timer mode is selected by clearing bit T0CS (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.

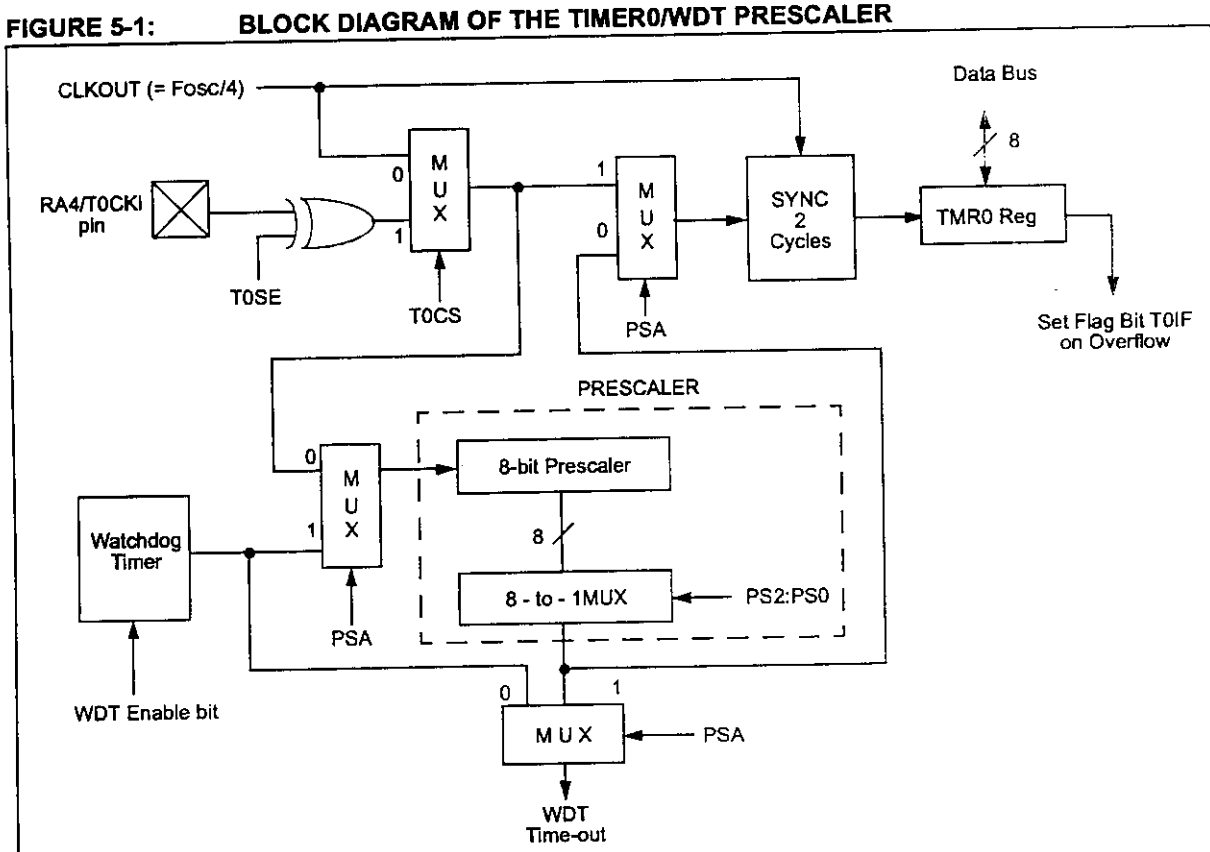
Counter mode is selected by setting bit T0CS (OPTION_REG<5>). In Counter mode, Timer0 will increment either on every rising, or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit, T0SE (OPTION_REG<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 5.2.

The prescaler is mutually exclusively shared between the Timer0 module and the Watchdog Timer. The prescaler is not readable or writable. Section 5.3 details the operation of the prescaler.

5.1 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



Note: T0CS, T0SE, PSA, PS2:PS0 are (OPTION_REG<5:0>).

10.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module is one of the two serial I/O modules. (USART is also known as a Serial Communications Interface or SCI.) The USART can be configured as a full duplex asynchronous system that can communicate with peripheral devices such as CRT terminals and personal computers, or it can be configured as a half duplex synchronous system that can communicate with peripheral devices such as A/D or D/A integrated circuits, serial EEPROMs etc.

The USART can be configured in the following modes:

- Asynchronous (full duplex)
- Synchronous - Master (half duplex)
- Synchronous - Slave (half duplex)

Bit SPEN (RCSTA<7>) and bits TRISC<7:6> have to be set in order to configure pins RC6/TX/CK and RC7/RX/DT as the Universal Synchronous Asynchronous Receiver Transmitter.

The USART module also has a multi-processor communication capability using 9-bit address detection.

REGISTER 10-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

bit 7 **CSRC:** Clock Source Select bit
Asynchronous mode:
Don't care
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled

Note: SREN/CREN overrides TXEN in SYNC mode.

bit 4 **SYNC:** USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **Unimplemented:** Read as '0'

bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode

bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** 9th bit of Transmit Data, can be parity bit

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

REGISTER 10-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

bit 7 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (configures RC7/RX/DT and RC6/TX/CK pins as serial port pins)
0 = Serial port disabled

bit 6 **RX9:** 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
Don't care
Synchronous mode - master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode - slave:
Don't care

bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
1 = Enables continuous receive
0 = Disables continuous receive
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enables interrupt and load of the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received, and ninth bit can be used as parity bit

bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D:** 9th bit of Received Data (can be parity bit, but must be calculated by user firmware)

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

10.1 USART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In Asynchronous mode, bit BRGH (TXSTA<2>) also controls the baud rate. In Synchronous mode, bit BRGH is ignored. Table 10-1 shows the formula for computation of the baud rate for different USART modes which only apply in Master mode (internal clock).

Given the desired baud rate and Fosc, the nearest integer value for the SPBRG register can be calculated using the formula in Table 10-1. From this, the error in baud rate can be determined.

It may be advantageous to use the high baud rate (BRGH = 1), even for slower baud clocks. This is because the Fosc/(16(X + 1)) equation can reduce the baud rate error in some cases.

Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

10.1.1 SAMPLING

The data on the RC7/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

TABLE 10-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = Fosc/(64(X+1))	Baud Rate = Fosc/(16(X+1))
1	(Synchronous) Baud Rate = Fosc/(4(X+1))	N/A

X = value in SPBRG (0 to 255)

TABLE 10-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

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
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used by the BRG.

TABLE 10-3: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	1.221	1.75	255	1.202	0.17	207	1.202	0.17	129
2.4	2.404	0.17	129	2.404	0.17	103	2.404	0.17	64
9.6	9.766	1.73	31	9.615	0.16	25	9.766	1.73	15
19.2	19.531	1.72	15	19.231	0.16	12	19.531	1.72	7
28.8	31.250	8.51	9	27.778	3.55	8	31.250	8.51	4
33.6	34.722	3.34	8	35.714	6.29	6	31.250	6.99	4
57.6	62.500	8.51	4	62.500	8.51	3	52.083	9.58	2
HIGH	1.221	-	255	0.977	-	255	0.610	-	255
LOW	312.500	-	0	250.000	-	0	156.250	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207	0.3	0	191
1.2	1.202	0.17	51	1.2	0	47
2.4	2.404	0.17	25	2.4	0	23
9.6	8.929	6.99	6	9.6	0	5
19.2	20.833	8.51	2	19.2	0	2
28.8	31.250	8.51	1	28.8	0	1
33.6	-	-	-	-	-	-
57.6	62.500	8.51	0	57.6	0	0
HIGH	0.244	-	255	0.225	-	255
LOW	62.500	-	0	57.6	-	0

TABLE 10-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	-	-	-
2.4	-	-	-	-	-	-	2.441	1.71	255
9.6	9.615	0.16	129	9.615	0.16	103	9.615	0.16	64
19.2	19.231	0.16	64	19.231	0.16	51	19.531	1.72	31
28.8	29.070	0.94	42	29.412	2.13	33	28.409	1.36	21
33.6	33.784	0.55	36	33.333	0.79	29	32.895	2.10	18
57.6	59.524	3.34	20	58.824	2.13	16	56.818	1.36	10
HIGH	4.883	-	255	3.906	-	255	2.441	-	255
LOW	1250.000	-	0	1000.000		0	625.000	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-
1.2	1.202	0.17	207	1.2	0	191
2.4	2.404	0.17	103	2.4	0	95
9.6	9.615	0.16	25	9.6	0	23
19.2	19.231	0.16	12	19.2	0	11
28.8	27.798	3.55	8	28.8	0	7
33.6	35.714	6.29	6	32.9	2.04	6
57.6	62.500	8.51	3	57.6	0	3
HIGH	0.977	-	255	0.9	-	255
LOW	250.000	-	0	230.4	-	0

10.2 USART Asynchronous Mode

In this mode, the USART uses standard non-return-to-zero (NRZ) format (one START bit, eight or nine data bits, and one STOP bit). The most common data format is 8-bits. An on-chip, dedicated, 8-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator. The USART transmits and receives the LSb first. The transmitter and receiver are functionally independent, but use the same data format and baud rate. The baud rate generator produces a clock, either x16 or x64 of the bit shift rate, depending on bit BRGH (TXSTA<2>). Parity is not supported by the hardware, but can be implemented in software (and stored as the ninth data bit). Asynchronous mode is stopped during SLEEP.

Asynchronous mode is selected by clearing bit SYNC (TXSTA<4>).

The USART Asynchronous module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver

10.2.1 USART ASYNCHRONOUS TRANSMITTER

The USART transmitter block diagram is shown in Figure 10-1. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the STOP bit has been transmitted from the previous load. As soon as the STOP bit is transmitted, the TSR is loaded with new data from the TXREG register (if available). Once the TXREG register transfers the data to the TSR register (occurs in one TCY), the TXREG register is empty and flag bit TXIF (PIR1<4>) is set. This interrupt can be enabled/disabled by setting/clearing enable bit TXIE (PIE1<4>). Flag bit TXIF will be set, regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicates the status of the TXREG register, another bit TRMT (TXSTA<1>) shows the status of the TSR register. Status bit TRMT is a read only bit, which is set when the TSR register is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty.

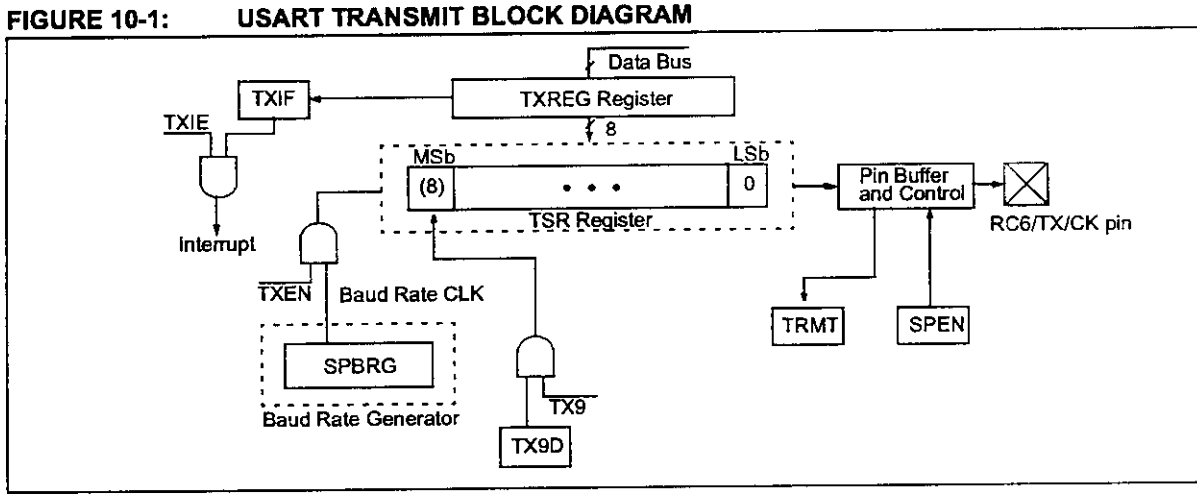
Note 1: The TSR register is not mapped in data memory, so it is not available to the user.

2: Flag bit TXIF is set when enable bit TXEN is set. TXIF is cleared by loading TXREG.

Transmission is enabled by setting enable bit TXEN (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data and the baud rate generator (BRG) has produced a shift clock (Figure 10-2). The transmission can also be started by first loading the TXREG register and then setting enable bit TXEN. Normally, when transmission is first started, the TSR register is empty. At that point, transfer to the TXREG register will result in an immediate transfer to TSR, resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 10-3). Clearing enable bit TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. As a result, the RC6/TX/CK pin will revert to hi-impedance.

In order to select 9-bit transmission, transmit bit TX9 (TXSTA<6>) should be set and the ninth bit should be written to TX9D (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG register. This is because a data write to the TXREG register can result in an immediate transfer of the data to the TSR register (if the TSR is empty). In such a case, an incorrect ninth data bit may be loaded in the TSR register.

FIGURE 10-1: USART TRANSMIT BLOCK DIAGRAM



When setting up an Asynchronous Transmission, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH (Section 10.1).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set transmit bit TX9.
5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).
8. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

FIGURE 10-2: ASYNCHRONOUS MASTER TRANSMISSION

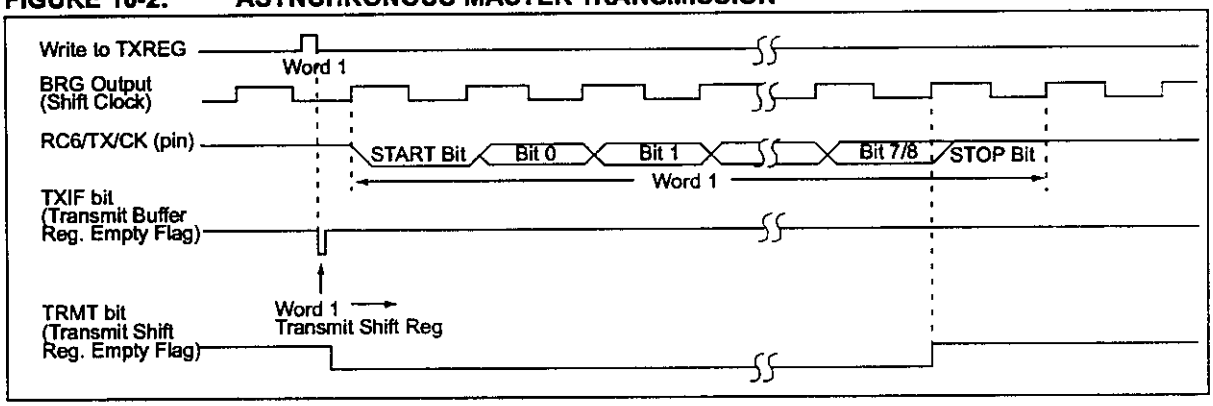
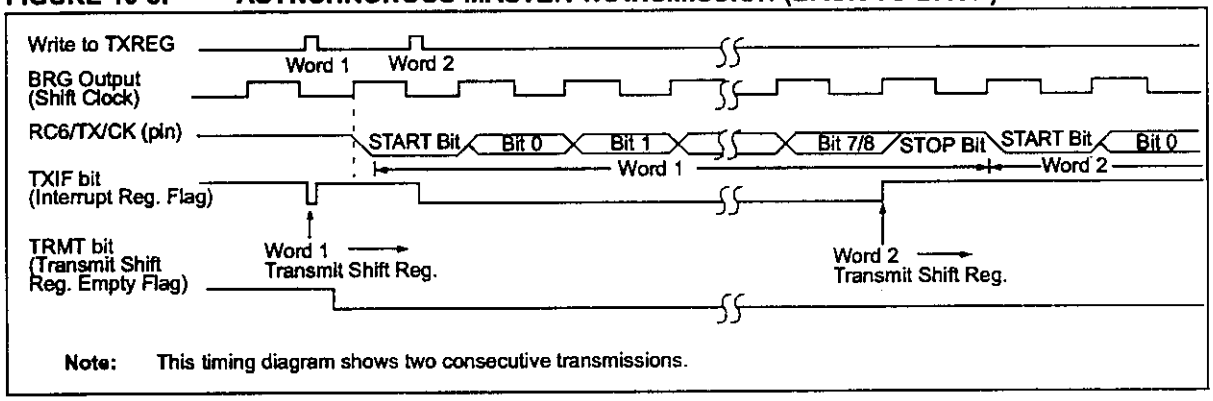


FIGURE 10-3: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 10-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

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
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F873/876; always maintain these bits clear.

10.2.2 USART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 10-4. The data is received on the RC7/RX/DT pin and drives the data recovery block. The data recovery block is actually a high speed shifter, operating at x16 times the baud rate; whereas, the main receive serial shifter operates at the bit rate or at Fosc.

Once Asynchronous mode is selected, reception is enabled by setting bit CREN (RCSTA<4>).

The heart of the receiver is the receive (serial) shift register (RSR). After sampling the STOP bit, the received data in the RSR is transferred to the RCREG register (if it is empty). If the transfer is complete, flag bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit RCIE (PIE1<5>). Flag bit RCIF is a read only bit, which is cleared by the hardware. It is cleared when the RCREG register has been read and is empty. The RCREG is a double buffered register (i.e., it is a two deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting to the RSR register. On the detection of the STOP bit of the third byte, if the RCREG register is still full, the overrun error bit OERR (RCSTA<1>) will be set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Overrun bit OERR has to be cleared in software. This is done by resetting the receive logic (CREN is cleared and then set). If bit OERR is set, transfers from the RSR register to the RCREG register are inhibited, and no further data will be received. It is therefore, essential to clear error bit OERR if it is set. Framing error bit FERR (RCSTA<2>) is set if a STOP bit is detected as clear. Bit FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG will load bits RX9D and FERR with new values, therefore, it is essential for the user to read the RCSTA register before reading the RCREG register in order not to lose the old FERR and RX9D information.

FIGURE 10-4: USART RECEIVE BLOCK DIAGRAM

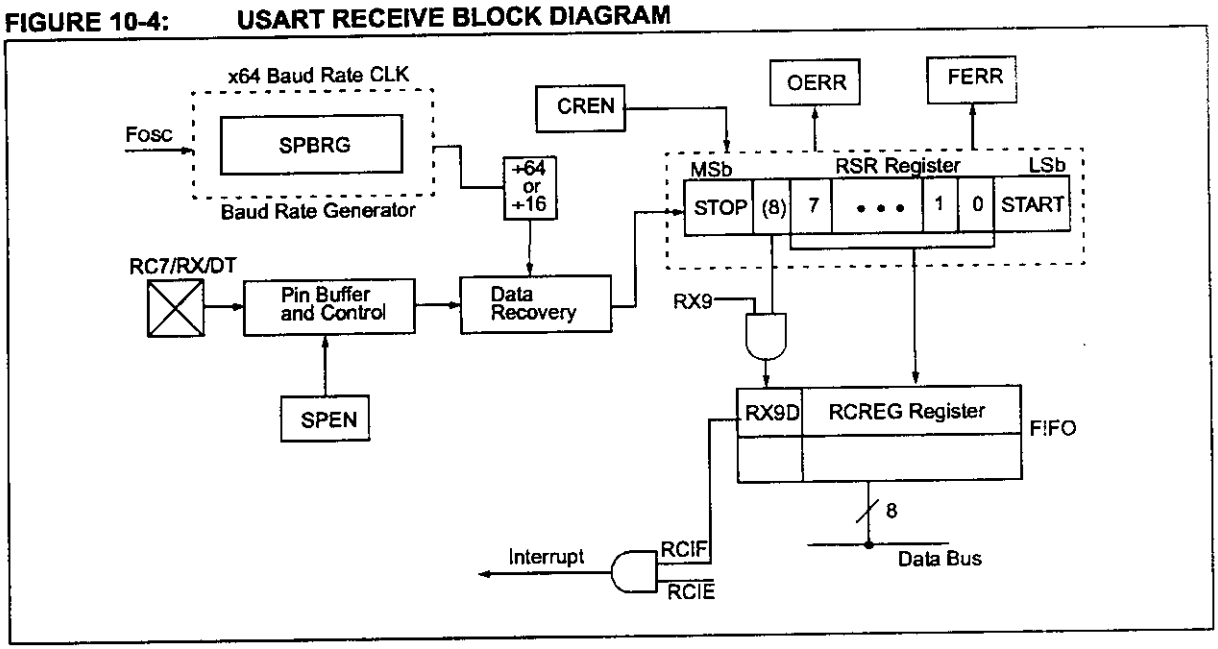
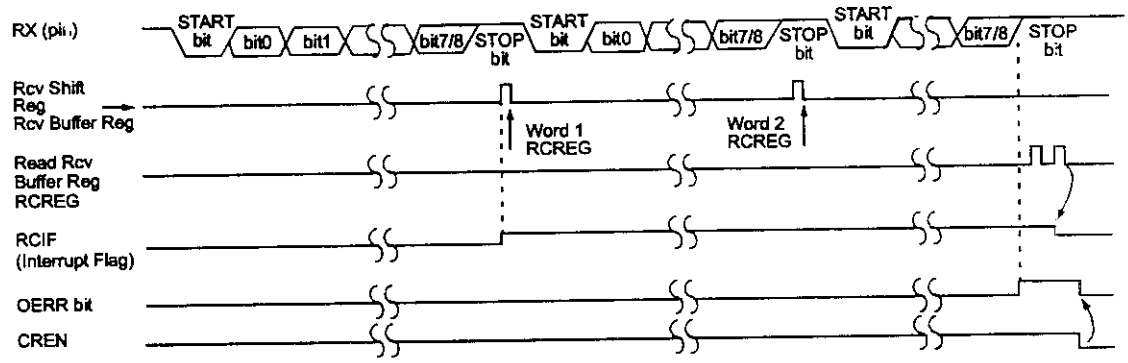


FIGURE 10-5: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

When setting up an Asynchronous Reception, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH (Section 10.1).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit RCIE.
4. If 9-bit reception is desired, then set bit RX9.
5. Enable the reception by setting bit CREN.
6. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE is set.
7. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing enable bit CREN.
10. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

TABLE 10-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

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
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.
Note 1: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

12.0 SPECIAL FEATURES OF THE CPU

All PIC16F87X devices have a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

- Oscillator Selection
- RESET
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- SLEEP
- Code Protection
- ID Locations
- In-Circuit Serial Programming
- Low Voltage In-Circuit Serial Programming
- In-Circuit Debugger

PIC16F87X devices have a Watchdog Timer, which can be shut-off only through configuration bits. It runs off its own RC oscillator for added reliability.

There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in RESET until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only. It is designed to keep the part in RESET while the power supply stabilizes. With these two timers on-chip, most applications need no external RESET circuitry.

SLEEP mode is designed to offer a very low current Power-down mode. The user can wake-up from SLEEP through external RESET, Watchdog Timer Wake-up, or through an interrupt.

Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits is used to select various options.

Additional information on special features is available in the PICmicro™ Mid-Range Reference Manual, (DS33023).

12.1 Configuration Bits

The configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1'), to select various device configurations. The erased, or unprogrammed value of the configuration word is 3FFFh. These bits are mapped in program memory location 2007h.

It is important to note that address 2007h is beyond the user program memory space, which can be accessed only during programming.

13.0 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 highly orthogonal and 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 µs. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 µs.

Table 13-2 lists the instructions recognized by the MPASM™ assembler.

Figure 13-1 shows the general formats that the instructions can have.

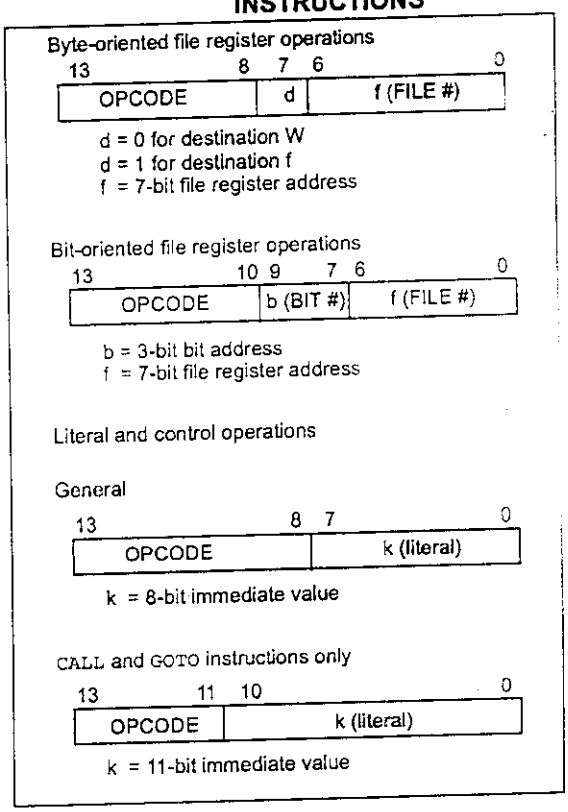
Note: To maintain upward compatibility with future PIC16F87X products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

FIGURE 13-1: GENERAL FORMAT FOR INSTRUCTIONS



A description of each instruction is available in the PICmicro™ Mid-Range Reference Manual, (DS33023).

TABLE 13-2: PIC16F87X INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
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	$\overline{TO}$,$\overline{PD}$	
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	$\overline{TO}$,$\overline{PD}$	
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).

13.1 Instruction Descriptions

ADDLW	Add Literal and W
Syntax:	[*label*] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the eight bit literal 'k' and the result is placed in the W register.

ADDWF	Add W and f
Syntax:	[*label*] ADDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

ANDLW	AND Literal with W
Syntax:	[*label*] ANDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.

ANDWF	AND W with f
Syntax:	[*label*] ANDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .AND. (f) → (destination)
Status Affected:	Z
Description:	AND the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

BCF	Bit Clear f
Syntax:	[*label*] BCF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	0 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

BSF	Bit Set f
Syntax:	[*label*] BSF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	1 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b < 7
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2TCY instruction.

BTFSC	Bit Test, Skip if Clear
Syntax:	[*label*] BTFSC f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making

CALL	Call Subroutine
Syntax:	[*label*] CALL k
Operands:	0 ≤ k ≤ 2047
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

CLRF	Clear f
Syntax:	[*label*] CLRF f
Operands:	0 ≤ f ≤ 127
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

CLRW	Clear W
Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

CLRWDT	Clear Watchdog Timer
Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

COMF	Complement f
Syntax:	[*label*] COMF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	($\bar{f}$) → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.

DECF	Decrement f
Syntax:	[*label*] DECF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

DECFSZ **Decrement f, Skip if 0**

Syntax: [*label*] DECFSZ f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) - 1 → (destination);
skip if result = 0

Status Affected: None

Description: The contents of register 'f' are decremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.
If the result is 1, the next instruction is executed. If the result is 0, then a NOP is executed instead making it a 2TCY instruction.

GOTO **Unconditional Branch**

Syntax: [*label*] GOTO k

Operands: $0 \le k \le 2047$

Operation: k → PC<10:0>
PCLATH<4:3> → PC<12:11>

Status Affected: None

Description: GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

INCF **Increment f**

Syntax: [*label*] INCF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) + 1 → (destination)

Status Affected: Z

Description: The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.

INCFSZ **Increment f, Skip if 0**

Syntax: [*label*] INCFSZ f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) + 1 → (destination),
skip if result = 0

Status Affected: None

Description: The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.
If the result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead, making it a 2TCY instruction.

IORLW **Inclusive OR Literal with W**

Syntax: [*label*] IORLW k

Operands: $0 \le k \le 255$

Operation: (W) .OR. k → (W)

Status Affected: Z

Description: The contents of the W register are OR'ed with the eight bit literal 'k'. The result is placed in the W register.

IORWF **Inclusive OR W with f**

Syntax: [*label*] IORWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .OR. (f) → (destination)

Status Affected: Z

Description: Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

RLF **Rotate Left f through Carry**

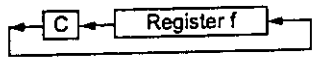
Syntax: [*label*] RLF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.



RETURN **Return from Subroutine**

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC

Status Affected: None

Description: Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

RRF **Rotate Right f through Carry**

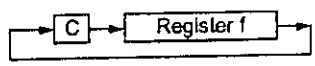
Syntax: [*label*] RRF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$.

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.



SLEEP

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Description: The power-down status bit, $\overline{PD}$ is cleared. Time-out status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared.
The processor is put into SLEEP mode with the oscillator stopped.

SUBLW **Subtract W from Literal**

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Description: The W register is subtracted (2's complement method) from the eight-bit literal 'k'. The result is placed in the W register.

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) - (W) → (destination)

Status Affected: C, DC, Z

Description: Subtract (2's complement method) W register from register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

SWAPF	Swap Nibbles in f
Syntax:	[*label*] SWAPF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f<3:0>) → (destination<7:4>), (f<7:4>) → (destination<3:0>)
Status Affected:	None
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed in register 'f'.

XORLW	Exclusive OR Literal with W
Syntax:	[*label*] XORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .XOR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register.

XORWF	Exclusive OR W with f
Syntax:	[*label*] XORWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .XOR. (f) → (destination)
Status Affected:	Z
Description:	Exclusive OR the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

PIE project

, CD54/74HCT373
, CD54/74HCT573

File Number 1679

MOS Logic

Octal Transparent Latch, 3-State Output

Type Features:

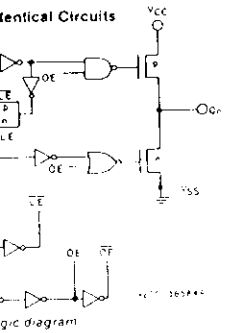
- Common latch enable control
- Common 3-state output enable control
- Buffered inputs
- 3-State outputs
- Bus line driving capacity
- Typical propagation delay = 12 ns @ V_{CC} = 5V, C_L = 15 pF, T_A = 25° C (Data to Output for HC373)

573 and CD54/74HCT373/573
sparent Latches manufactured
hnology. They possess the low
dard CMOS integrated circuits.
drive 15 LSTTL devices. The
functionally as well as pin
rd 54/74LS373 and 573.

nt to the inputs when the latch
e latch enable ($\overline{LE}$) goes low the
enable ($\overline{OE}$) controls the 3-state
enable ($\overline{OE}$) is high the outputs
e state. The latch operation is
the output enable. The 373 and
n and differ only in their pinout

are supplied in 20 lead ceramic
ffix). The CD74HC/HCT373/573
astic dual-in-line plastic package
urface mount plastic packages
e also available in chip form

Identical Circuits



gic diagram

Family Features:

- Fanout (Over Temperature Range):
  Standard Outputs - 10 LSTTL Loads
  Bus Driver Outputs - 15 LSTTL Loads
- Wide Operating Temperature Range:
  CD74HC/HCT: -40 to +85° C
- Balanced Propagation Delay and Transition Times
- Significant Power Reduction Compared to LSTTL Logic ICs
- Alternate Source is Philips/Signetics
- CD54HC/CD74HC Types:
  2 to 6 V Operation
  High Noise Immunity: N_{IL} = 30%, N_{IH} = 30% of V_{CC} @ V_{CC} = 5 V
- CD54HCT/CD74HCT Types:
  4.5 to 5.5 V Operation
  Direct LSTTL Input Logic Compatibility
  V_{IL} = 0.8 V Max., V_{IH} = 2 V Min.
  CMOS Input Compatibility
  $I_I \leq 1\ \mu A$ @ V_{OL}, V_{OH}

TRUTH TABLE

$\overline{\text{Output Enable}}$	$\overline{\text{Latch Enable}}$	Data	Output
L	H	H	H
L	H	L	L
L	L	l	L
L	L	h	H
H	X	X	Z

Note:
L = Low voltage level
H = High voltage level
X = Don't Care
Z = High Impedance State
l = Low voltage level one set-up time prior to the high to low latch enable transition
h = High voltage level one set-up time prior to the high to low latch enable transition

CD54/74HC373, CD54/74HCT373
CD54/74HC573, CD54/74HCT573

MAXIMUM RATINGS, *Absolute-Maximum Values:*

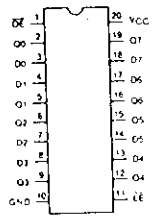
DC SUPPLY-VOLTAGE, (V_{CC}): ..... -0.5 to +7 V
(Voltages referenced to ground) ..... ±20 mA
DC INPUT DIODE CURRENT, I_{IK} (FOR $V_I < -0.5$ V OR $V_I > V_{CC}$ +0.5 V) ..... ±20 mA
DC OUTPUT DIODE CURRENT, I_{OK} (FOR $V_O < -0.5$ V OR $V_O > V_{CC}$ +0.5 V) ..... ±35 mA
DC DRAIN CURRENT, PER OUTPUT (I_O) (FOR -0.5 V < V_O < V_{CC} +0.5 V) ..... ±70 mA
DC V_{CC} OR GROUND CURRENT, (I_{CC}) ..... 500 mW
POWER DISSIPATION PER PACKAGE (P_D): ..... Derate Linearly at 8 mW/°C to 300 mW
For T_A = -40 to +60°C (PACKAGE TYPE E) ..... 500 mW
For T_A = +60 to +85°C (PACKAGE TYPE E) ..... Derate Linearly at 8 mW/°C to 300 mW
For T_A = -55 to +100°C (PACKAGE TYPE F, H) ..... 400 mW
For T_A = +100 to +125°C (PACKAGE TYPE F, H) ..... Derate Linearly at 6 mW/°C to 70 mW
For T_A = -40 to +70°C (PACKAGE TYPE M) .....
For T_A = +70 to +125°C (PACKAGE TYPE M) .....
OPERATING-TEMPERATURE RANGE (T_A) ..... -55 to +125°C
PACKAGE TYPE F, H ..... -40 to +85°C
PACKAGE TYPE E, M ..... -65 to +150°C
STORAGE TEMPERATURE (T_{stg}) ..... +265°C
LEAD TEMPERATURE (DURING SOLDERING): ..... +300°C
At distance 1/16 ± 1/32 in. (1.59 ± 0.79 mm) from case for 10 s max.
Unit inserted into a PC Board (min. thickness 1/16 in., 1.59 mm) with solder contacting lead tips only

RECOMMENDED OPERATING CONDITIONS:

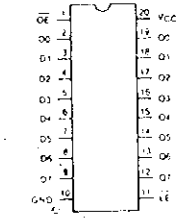
For maximum reliability, nominal operating conditions should be selected so that operation is always within the following ranges:

CHARACTERISTIC	LIMITS MIN.	LIMITS MAX.	UNITS
Supply-Voltage Range (For T_A = Full Package-Temperature Range) V_{CC}*			
CD54/74HC Types	2	6	V
CD54/74HCT Types	4.5	5.5	V
DC Input or Output Voltage V_I, V_O	0	V_{CC}	V
Operating Temperature T_A:			
CD74 Types	-40	+85	°C
CD54 Types	-55	+125	°C
Input Rise and Fall Times, t_r, t_f			
at 2 V	0	1000	ns
at 4.5 V	0	500	ns
at 6 V	0	400	ns

*Unless otherwise specified, all voltages are referenced to Ground.



CD54/74HC373, CD54/74HCT373
TERMINAL ASSIGNMENT



CD54/74HC573, CD54/74HCT573
TERMINAL ASSIGNMENT

Centronics

General

One of the most used printer interfaces, since every Intel/DOS/Windows based Personal Computer today has a Sub-D25 female connector which is called LPT1.

Pinning 36 pin Amphenol

Male	Female

Pin	Signal	Abbr.	Source
1	Data Strobe (low)	STROBE	Computer
2	Data Bit 1 (LSB)	D1	Computer
3	Data Bit 2	D2	Computer
4	Data Bit 3	D3	Computer
5	Data Bit 4	D4	Computer
6	Data Bit 5	D5	Computer
7	Data Bit 6	D6	Computer
8	Data Bit 7	D7	Computer
9	Data Bit 8 (LSB)	D8	Computer
10	Acknowledge (low)	ACK	Printer
11	Busy (high)	BUSY	Printer
12	Paper End (high)	PE	Printer
13	Select (high)	SEL	Printer
14	Supply Ground		-
15	Oscillator Transmit		Printer
16	Logical Ground		-
17	Chassis Ground		-
18	+5 Vdc	+V	Printer
19	Return Data Strobe		-
20	Return Data Bit 1		-
21	Return Data Bit 2		-

11	Busy (high)	BUSY	Printer
12	Paper End (high)	PE	Printer
13	Select (high)	SEL	Printer
14	Auto Line Feed (low)	LF	Computer
15	Error (low)	ERROR	Printer
16	Initialize Printer (prime-low)	PRIME	Computer
17	Select Input (low)	SEL	Computer
18	Return/ground	GND	-
19	Return/ground	GND	-
20	Return/ground	GND	-
21	Return/ground	GND	-
22	Return/ground	GND	-
23	Return/ground	GND	-
24	Return/ground	GND	-
25	Return/ground	GND	-

Functional Description

STROBE: Active low pulse used to transfer data into the printer. Pulse width must be between 0.5 and 500 microseconds for most printers.

Dn: Data lines, high is a one.

ACK: Active low pulse indicates that data has been received and the printer is ready to accept more.

BUSY: A high signal indicates that the printer cannot receive data.

PE: A high signal indicates that the printer is out of paper (Paper End)

SELECT A high signal indicates that the

OUT:	printer is on-line	
AUTO FEED:	A low signal indicates to the printer that a line feed is required after each Carriage return.	This signal is usea as a ground line by some manufacturers.
OSCXT:	A 100-200 KHz signal used by true Centronics printers only.	
+5V:	+5Vdc	Not provided by all manufacturers
PRIME:	A low signal resets the printer to its power-up state and the printer buffer is cleared	
FAULT:	A low signal indicates that the printer is in an off-line or error state	
LINE COUNT:	Used by true Centronics printers only.	Most of the time not used
SELECT IN:	A high signal indicates to the printer that a DC1/ DC3 code is valid.	This signal is used by a few manufacturers

Sub-D25 to Centronics cable

Sub-D25	**Centronics 36**
1	--- 1
2	--- 2
3	--- 3
4	--- 4
5	--- 5
6	--- 6
7	--- 7
8	--- 8
9	--- 9

10 --- 10
11 --- 11
12 --- 12
13 --- 13
14 --- 14
15 --- 32
16 --- 31
17 --- 36
18 --- 33
19 --- 19
20 --- 21
21 --- 23
22 --- 25
23 --- 27
24 --- 29
25 --- 30