DATA ACQUISITION SYSTEM
WITH

PRINTER INTERFACE

Dissertation submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING IN ELECTRONICS AND COMMUNICATION SYSTEMS
Of Bharathiar University

Submitted by
G.DEVIKA
PL.RATHNA
N.MANOPRIYA

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE –641 006
TAMIL NADU
MARCH 2004

CERTIFICATES

KUMARAGURU COLLEGE OF TECHNOLOGY
(Affiliated to Bharathiar University, Coimbatore)

CERTIFICATE

This is to certify that the project entitled

DATA ACQUISITION SYSTEM WITH PRINTER INTERFACE

done by

G. DEVIKA	**0027D0187**
PL.RATHNA	**0027D0220**
N.MANOPRIYA	**0027D0204**

And submitted in partial fulfillment of the Requirement for the award of the degree of

Bachelor of Electronics and Communication Engineering of Bharathiar University, Coimbatore.

____

Professor & Head of the department
(Prof. Muthuraman Ramasamy)

____

Project Guide
(Prof.Muthuraman Ramasamy)

Certified that the candidates were examined by us in the project work Viva voce examination held on ________________.

____

Internal Examiner

____

External Examiner

BEC
Bright Electro Control Equipments (P) Ltd
Manufacturers of Electronic & Electrical Equipments
GRASS HILL ROAD, SF No.162, KAMADENU NAGAR,
AVARAMPALAYAM, COIMBATORE 641 006. INDIA
TEL : 0422 - 2560116, 2567249 FAX : 0422 - 2562973

ISO 9001 TUV REGISTERED COMPANY

PROJECT CERTIFICATE

This is to certify that

1. N.MANOPRIYA (2KECE 25)
2. P.L.RATHNA (2KECE 41)
3. G.DEVIKA (2KECE 08)

Final year engineering students of KUMARAGURU COLLEGE OF TECHNOLOGY, CHINNAVEDAMPATTI ,COIMBATORE have finished their project "DATA ACQUISITION SYSTEM WITH PRINTER INTERFACE" in our company.

During this period their conduct and behavior was good.

Their interest towards work and sincerity is appreciated.

We wish them all success in the future.

For BRIGHT ELETRO CONTROL EQUIPMENTS (P) LTD

N. [signature]

Authorised Signatory.

TABLE OF CONTENTS

TABLE OF CONTENTS

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

We would like to begin by thanking the Almighty for showering his divine grace on us.

We express our sincere and profound sense of gratitude to our principal **Dr.K.K.Padmanabhan** B.sc (Engg),M.Tech..,Ph.D., for his kind partronage.

We are highly indebted and grateful to our guide **Prof. Muthuraman Ramasamy**,ME.,FIE.,FIETE,MIEEE(USA).,MISTE.,MBMESI.,C.ENG.(I), Head of the Department, Electronics and communication Engineering for his constant support and guidance offered to us during the course of this project work.His encouragement has given us the urge to do our best and we would like to express our sincere gratitude to him.

We express our sincere thanks to our external guide Mr.N.Shanmugasundaram, Managing director, Bright Electro Control Equipment (p) Ltd, Coimbatore, for his valuable and timely suggestions

We would be failing in our duty if we don't express our indebtedness to all our teachers in the Department of Electronics and Communication for their suggestions and constant encouragement.

SYNOPSIS

SYNOPSIS

Most of the data acquisition systems are time varying.But certain applications demands constant values of inputs so as to function properly.In order to maintain a constant value , a printer could be interfaced with the system and the instantaneous values of the inputs could be printed at particular intervals of time.

The functioning of a computer system interfaced with the microcontroller is inefficient in the sense that, it is more power consuming and the data could be changed any number of times by just altering the software.

These factors provides room for the drawback that an optimum use of the computer, in a specific period , is not made and there is a waste of resources and insecurity of data. Solution to overcome this drawback is the **printer device driver** through which the printer is interfaced with the data acquisition system.

The printer device driver is the interfacing unit, which controls all the actions of the printer apart from interfacing it with the micro controller unit . The printer device driver is programmed to accept the digital values from the micro controller and prints the same at particular instants of time.

In our project entitled 'DATA ACQUISITION SYSTEM WITH PRINTER INTERFACE' is designed and implemented with hardware and software programs in assembly language. The system consists of a PIC programmable microcontroller PIC16F876, which s programmed for Analog to Digital conversion and printer interfacing.

The system has been working satisfactorily and the parameters of the acquisition system are printed at particular instants of time through the printer interface.

INTRODUCTION

INTRODUCTION

FEATURES OF THE SYSTEM:

The project "DATA ACQUISITION SYSTEM WITH PRINTER INTERFACE" is for tapping a 8-bit data from the data acquisition system and sending to the printer. The three analog inputs are

- Voltage
- Temperature
- Current

An interface using PIC16F876 is designed to collect data regarding the parameters of acquisition system and get the instantaneous values of those parameters printed through a printer interfaced with the system.Printout of the stored data could be taken as and when required.

The data source is the acquisition system from where the data is tapped in parallel using the printer device driver 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.

ABOUT EMBEDDED SYSTEMS:

In an embedded system, software is fused into a hardware device. An embedded system is a component within some larger system, contrast with general-purpose computer. The software in the embedded system is called firmware. Embedded systems are everywhere and affect virtually every aspect of our lives. We use embedded systems in our homes as washers, dryers, microwave oven, stereos, climatic control systems, alarm system devices, hand-held thermometers, greeting cards and smart toys.

CHARACTERISTICS OF EMBEDDED SYSTEMS:

- Inbuilt intelligence
- Immediate control of hardware
- Uses dedicated software
- Performs a specific function
- Their work is subjected to deadlines
- Respond to external events

SELECTION OF THE ORGANISATION

BRIGHT ELECTRO CONTROL EQUIPMENTS LTD

Bright Electro Control Equipments Ltd., is one of the pioneer in power electronics. Established in 1987. It is an ISO 9001 registered company, located in the Premier Industrial city of Coimbatore, in Southern India. Today the company has extended its operations in Chennai. They are concentrating in process control and power electronics.

Their major products are

UPS ON line/OFF line

UPS for Computer &Industrial Applications

SERVO, CVT and Automatic Stabilizers

The mission of Bright Electro Control Equipments Ltd is to be the leader of corporate with quality and excellence in all endeavours. They strive to provide quality to their customers to deliver value added services. They are frames for their high performance and low price products.

PROBLEM FORMULATION

MAIN OBJECTIVE

The main motive of our project is to interface a printer with a data acquisition system and the get the instantaneous values of the inputs printed

SPECIFIC OBJECTIVE

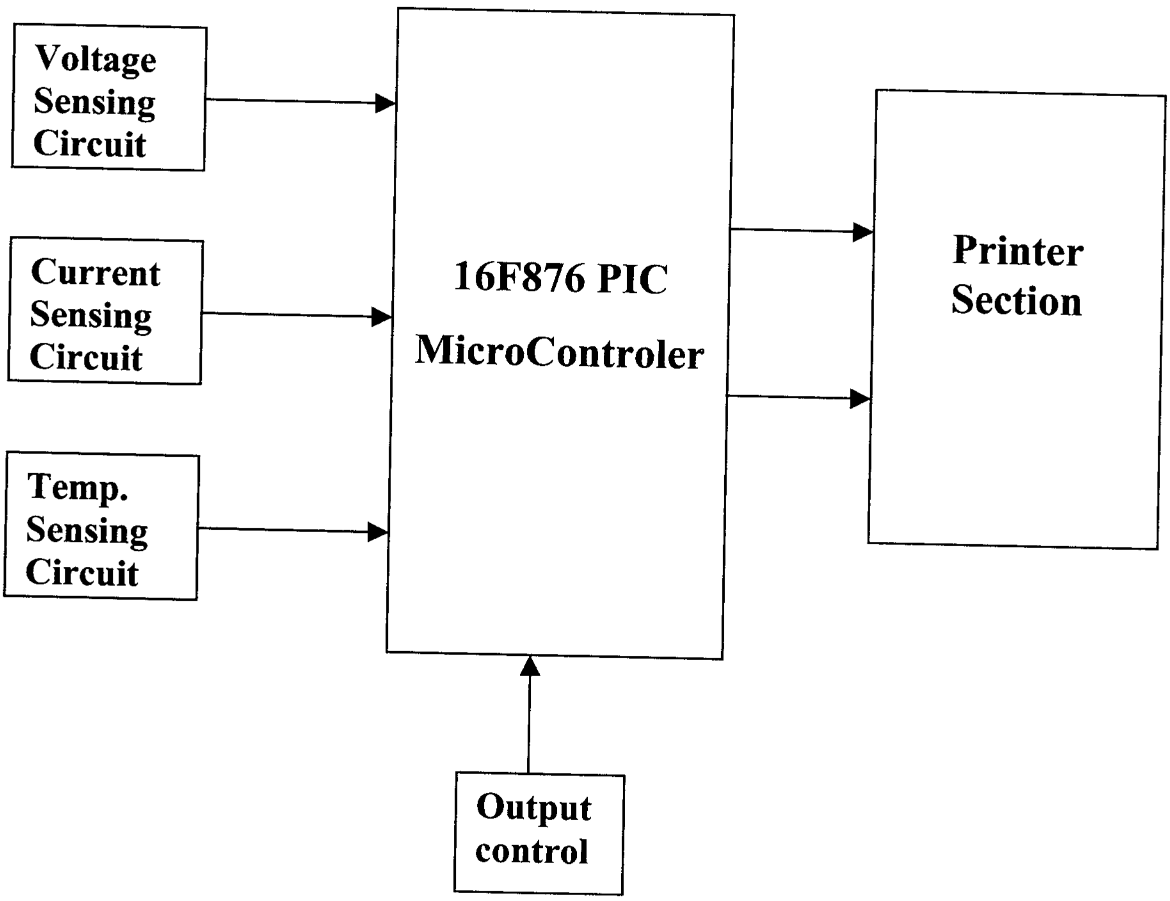
Following are the specific objectives of this system

- PIC programmed for A/D conversion
- Printer device driver – to interface the system with the printer
- Cost effective and high reliability

METHODOLOGY

Methodology is a project management activity, which is a consistent set of standards that are understood by all stakeholders and users.These standards should embrace both user and corporate needs.

BLOCK DIAGRAM



NEED FOR MICROCONTROLLERS

NEED FOR MICROCONTROLLER:

Now we are in the embedded information processing revolution. This revolution is truly hidden inside the products we are using everyday. Think of a washing machine that will adjust the water height inside it. Think about a car security system that immobilizes a car when an unauthorized entry happen or automatically unlocks the door when an authorized person approaches the car. All these examples show the information processing or intelligence inside a product.

Micro controllers drive the revolution in embedded intelligence. The tiny micro controllers are the most ubiquitous of all semiconductors. We can find at least 12 to 14 micro controllers used by everyday, and by everyone. For ex: Cellular phone, pager, watch and battery charger. The microcontroller plays an important key role in the project.

HARDWARE DESCRIPTION:

MICROCONTROLLER:

Microcontroller is a general-purpose device, which has inbuilt CPU, memory and peripherals to make it as a minicomputer.

MICROCONTROLLER Vs MICROPROCESSOR:

- Most microprocessors have many operational codes for moving data from external memory to the CPU but microcontroller have one or two.
- Microprocessors have one or two bit handling instructions. But microcontroller has many.
- Microprocessor don't have inbuilt peripherals but microcontrollers have.
- Microcontrollers work faster than microprocessor because of rapid movement of bit within the chip.
- Microcontroller can function as a computer with the addition of no external parts.
- In microcontroller all instructions are single word.

NEED FOR PIC

PIC

What is PIC?

PIC stands for Peripheral Interface Controller as coined by Microchip Technology Inc., USA.

Why PIC?

- PIC is a very popular microcontroller worldwide. Microchip is the first manufacturer of 8 pin RISC MCU.
- Microchip is the world's second largest chip manufacturer.
- Focus on high performance cost-effective, field programmable embedded control solutions.
- Variety of end-user Application Specific Standard Products (ASSP) and Application-Specific Integrated Circuits (ASIC).
- Global network of manufacturing and customer support facilities.

PIC micro devices have the following architectural features to attain the high Performance.

- Harvard architecture.
- Long word instruction
- Single word instruction
- Single cycle instruction
- Instruction pipelining

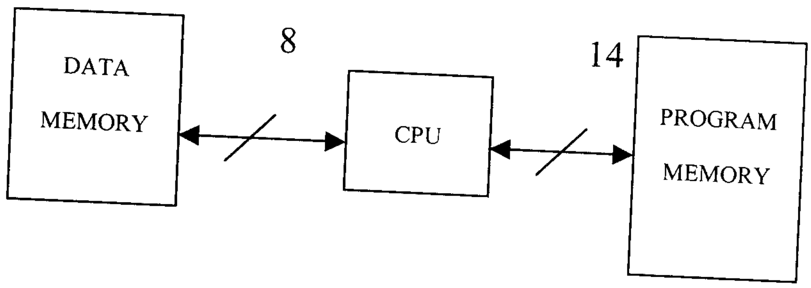
- Reduced instruction set
- Register file architecture
- Orthogonal (symmetric) instruction

HARVARD ARCHITECTURE:

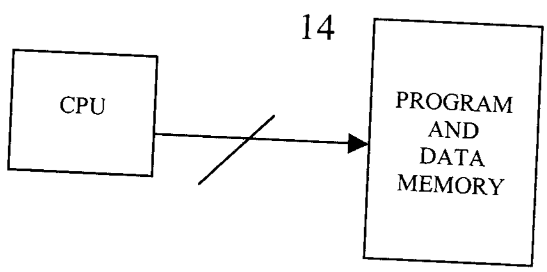
Harvard architecture has the program memory and data memory as separate memories and they are accessed from separate buses. This improves bandwidth over traditional Von-Newman architecture in which program and data are fetched from the same memory using the same bus. To execute an instruction, a Von-Newman machine must make one or more access across the 8-bit bus to fetch the instruction. Then data may need to be fetched, operated on and possibly written.

While with Harvard architecture, the instruction is fetched on a single instruction cycle. While the program memory is being accessed, the data memory is on an independent bus and can be read and written. These separate buses allow one instruction to execute while the next instruction is fetched. A comparison of both these architecture are shown below.

HARVARD :



VON-NEUMANN:



LONG WORD INSTRUCTION:

Long word instruction have a wider instruction bus the 8-bit data memory bus. This is possible because the two buses are separate. This further allows instructions to be seized differently than the 8-bit wide data word, which allows a more efficient use of the program memory, since the program memory width is optimized to the architectural requirements.

SINGLE WORD INSTRUCTION:

Single word instruction opcodes are 14-bits wide making it possible to have all single word instructions. A 14-bit wide program memory access bus fetches all single word instruction in a single cycle. With single word instruction, the number of words of program memory locations equals the number of instructions for the device. This means that all locations are valid instructions. Typically in the Von-Neumann architecture, most instructions are multi-byte. In general, a device with 4KB of program memory would allow approximately 2K of instructions. This 2:1 ratio is generalized and dependent on application code. Since each instruction may take multiple bytes, there is no assurance that each location is a valid instruction.

INSTRUCTION PIPELINE:

The instruction pipeline is a two-stage pipeline, which overlaps the fetch and execution of instructions. The fetch of the instruction takes one TCY, while the execution takes another TCY. However, due to the overlap of the fetch of the current instruction and execution of previous instruction, an instruction is fetched and another instruction is executed every single TCY.

SINGLE CYCLE INSTRUCTION:

With the program memory bus being 14-bits wide, the entire instruction is fetched in a single machine cycle(TCY). The instruction contains all the information required and is executed in a single cycle. There may be a one cycle delay in execution if the result of the instruction modified the contents of the program counter. This requires the pipeline to be flushed and a new instruction to be fetched.

REDUCED INSTRUCTION SET:

When an instruction set is well designed and highly orthogonal (symmetric)fewer instructions are required to perform all needed tasks. With fewer instructions, the whole set can be more rapidly learned.

REGISTER FILE ARCHITECTURE:

The register files or data memory can be directly or indirectly addressed. All the special function registers, including the program counter are mapped in the data memory.

ORTHOGONAL (SYMMETRIC) INSTRUCTION:

Orthogonal instructions make it possible to carry out any operation on any special register using any addressing mode. This symmetrical nature and lack of “special instructions” make programming simple yet efficient. In addition, the learning curve is reduced significantly. The mid-range instruction set uses only two non-register oriented instructions, which are used for the core features. One is the SLEEP instruction, which places the device into the lowest power use mode.The other is the CLRWDT instruction, which verifies the chip is operating properly by preventing the on-chip Watchdog Timer (WDT) from overflowing and resetting the device.

INSTRUCTION FLOW/PIPELINING:

An "instruction cycle" consists of four Q cycles (Q1,Q2,Q3,Q4). Fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change, then an extra cycle is required to complete the instruction.

- The instruction FETCH begins with the program counter incrementing in Q1.
- In the EXECUTON cycle, the fetched instruction is latched into the "Instruction Register (IR)" in cycle Q1.This instruction is then decoded and executed during the Q2,Q3 and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

CENTRAL PROCESSING UNIT (CPU):

The CPU can be thought of as the "brains" of the device. The Central Processing Unit (CPU) is responsible for the information in the program memory (instruction) to control the operation of the devices. Many of these instructions operate on data memory. It is responsible for fetching the correct instruction for execution, decoding that instruction, and then executing that instruction. The CPU sometimes works in conjunction with the ALU to complete the execution of the instruction (in arithmetic and logical operations). The CPU controls the program memory address bus, the data memory address bus, the data memory address bus, and accesses to the stack.

INSTRUCTION CLOCK:

Each instruction cycle (TCY) is comprised of four Q cycles (Q1-Q4). The Q cycle time is the same as the device oscillator cycle time (Tosc). The Q cycles provide the timing/designation for the decode, Read, Process data, Write, etc., of each instruction cycle. The following diagram shows the relationship of the Qcycles to the instruction cycle.

The four cycles that make up an instruction cycle (TCY) can be generalized as:

- Q1: Instruction Decode cycle or forced No operation
- Q2: Instruction Read Data Cycle or No operation

- Q3: Process the data
- Q4: Instruction Write Data Cycle or No operation

OSCILLATOR:

The internal oscillator circuit is used to generate the device clock. The device clock is required for the device to execute instruction and for the peripherals to function. Four device clock periods generate one internal instruction clock (TCY) cycle. There are up to eight different modes, which the oscillator may have. There are two modes, which allows the selection of the internal RC oscillator clock out (CLKOUT) to be driven on an I/O pin, or allow that I/O pin to be used for a general purpose function. The device configuration bits select the oscillator mode. The device configuration bits are non-volatile memory locations and the operating mode is determined by the value written during device programming. The RC oscillator option saves system cost while the LP crystal option saves power. Configuration bits are used to select the various options.

OSCILLATOR CONFIGURATIONS:

Mid-range devices can have up to eight different oscillator modes. The user can program up to three device configuration bits to select one of these eight modes:

- Low Frequency Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC External Resistor/Capacitor
- EXTRC External Resistor/Capacitor
- EXTRC External Resistor/Capacitor with CLKOUT
- INTRC Internal 4MHz Resistor/Capacitor
- INTRC Internal 4MHz Resistor/Capacitor with CLKOUT

The main difference between the LP, XT and HS modes is the gain of the internal inverter of the oscillator circuit, which allows the different frequency ranges.

MPLAB – AN INTEGRATED DEVELOPMENT ENVIRONMENT

MPLAB

MPLAB is Microchip Programming Laboratory, which is a Windows-based Integrated Development Environment (IDE) for the Microchip Technology Incorporated PIC micro controller families.

MPLAB allows us to write, debug and optimize PIC micro applications for firmware product designs. MPLAB includes a text editor, simulator and project manager.

MPLAB also supports the MPLAB-ICE and PICMASTER emulators, PICSTART Plus and PRO MATE II programmers, and other Microchip third party development system tools.

A project in MPLAB is the group of files needed to build an applications, along with their associations to various build tools. In this MPLAB Project, a c source file (main c) is associated with the MPLAB-CXX compiler. MPLAB will use this information to generate an object file (main C) for input into the linker (MPLINK). An assembly source file (prog.asm) is shown also with its associated assembler, MPASM. MPLAB will use this information to generate an object file (prog.o) for input into MPLINK.

The main output file generated by MPLINK is the Hex file (prog.hex) used by simulators (MPLAB- SIM), emulators (MPLAB-ICE and PICMASTER) and programmers(PRO MATE II and PICSTART plus.

How MPLAB helps:

The organization of MPLAB tools by function helps to make pull-down menus and customizable quick keys that are easy to find and use.

MPLAB- An Integrated Development Environment (IDE):

MPLAB is an easy-to-learn and use Integrated Development Environment (IDE). The IDE provides firmware development engineers the flexibility t develop and debug firmware for Microchip's PIC micro controller families. The MPLAB IDE runs under Microsoft Windows 3.1x, Windows 95,98,NT and 2000. Not all hardware components that function under the MPLAB IDE (such as emulators and device programmers) function under all operating systems.

Hi-Tech – C:

The HI –TECH C Compiler is a set of software, which translates programs written in the C language to executable machine code programs. Versions are available, which compile program for operation under the host operating system, or produce programs for execution in embedded system without an operating system.

Features of Hi_tech C:

Some of HI-TECH C's features are

- A single batch file or command file will compile, assemble and link entire program.
- The compiler performs strong type checking and issues warnings about various constructs which may represent programming errors.
- The generated code is extremely small and fast in execution.
- A full time run-time library is provided implementing all standard C input/output and other functions.
- The source code for all run-time routines is provided.
- A powerful general-purpose macro assembler is included.
- Programs may be generated to execute under the host operating system, or customized for installation in ROM.

ANALOG TO DIGITAL CONVERSION

ANALOG TO DIGITAL CONVERSION

In analog-to-digital converter (ADC) accepts an analog input a voltage or a current and converts it to a digital value that can be read by a microprocessor. This hypothetical part has two inputs: a reference and the signal to be measured. It has one output, an 8-bit digital word that represents the input value.

The reference voltage is the maximum value that the ADC can convert. Our example 8-bit ADC can convert values from 0V to the reference voltage. This voltage range is divided into 256 values, or steps. The size of the step is given by: $V_{ref}/256$ where V_{ref} is the reference voltage. The step size of the converter defines the converter's resolution. For a 5V reference, the step size is: 5V/256 = 0.0195V or 19.5Mv.

Our 8-bit converter represents the analog input as a digital word. The most significant bit of this word indicates whether the input voltage is greater than half the reference (2.5V, with a 5V reference). Each succeeding bit represents half the range of the previous bit.

Table 1 Example conversion, on an 8-bit ADC

Bit:	7	6	5	4	3	2	1	
Volts:	2.5	1.25	0.625	0.312	0.156	0.078	0.039	0 0 9
Output Value:	0	0	1	0	1	1	0	0

The resolution of an ADC is determined by the reference input and by the word width. The resolution defines the smallest voltage change that can be measured by the ADC. The resolution is the same as the smallest step size, and can be calculated by dividing the reference voltage by the number of possible conversion values. Resolution can be improved by reducing the reference input. The only way to increase resolution without reducing the range is to use an ADC with more bits. A 10-bit ADC has 2^{10}, or 1,024 possible output codes. So the resolution is 5V/1,024, or 4.88mV; a 12-bit ADC has a 1.22mV resolution for this same reference.

Types of ADCs are flash, successive approximation, and sigma-delta.

CHARACTERISTICS OF ADC:

a) RESOLUTION:

The resolution of the converter indicates the number of discrete values it can produce. It is usually expressed in bits. For example, an ADC that encodes an analog input to one of 256 discrete values has a resolution of eight bits, since

$$2^8 = 256$$

b) RESPONSE TIME:

Most ADCs are linear, which means that they are designed to produce an output value that is a linear function of, i.e. proportional to, the input. Another common type is the logarithmic ADC, which is used in telecommunications systems where the amplitude of the input signal varies over a wide range. The logarithmic ADC compresses the input signal into a smaller number of bits than a linear ADC with the same input range and resolution.

c) ACCURACY:

Accuracy depends on the error in the conversion. If the ADC is not broken, this error has two components: quantization error and (assuming the ADC is intended to be linear) non-linearity. These errors are measured in a unit called the LSB, which is an abbreviation for least significant bit. In the above example of an eight-bit ADC, an error of one LSB is 1/256 of the full signal range, or about 0.4%. Quantization error is due to the finite resolution of the ADC, and is an unavoidable imperfection in all types of ADC. The magnitude of the quantization error at the sampling instant is between zero and half of one LSB.

d) SAMPLING RATE:

Commonly, the analog signal is continuous in time and it is necessary to convert this to a flow of digital values. It is therefore required to define the rate at which new digital values are sampled from the analog signal. The rate of new values is called sampling rate of the converter.

The key idea here is that a continuously varying band limited signal can be sampled(ie. the signal values at intervals of time T, the sampling time, are measured and stored.) and then the original signal can be exactly reproduced from the discrete-time values by an interpolation formula. The accuracy is however limited by quantization error. However this faithful reproduction is only possible if the sampling rate is higher than twice the highest frequency component present in the signal. This is essentially what is called Shannon's sampling theorem.

e) ALIASING:

All ADCs work by sampling their input at discrete intervals of time. Their output is therefore an incomplete picture of the behaviour of the input. There is no way of knowing, by looking at the output, what the input was doing between one sampling instant and the next. If the input is known to be changing slowly compared to the sampling rate, then it can be assumed that the value of the signal between two sample instants was somewhere between the two sampled values. If, however, the input signal is changing fast compared to the sample rate, then this assumption is not valid. If the digital values produced by the ADC are, at some later stage in the system, converted back to analog values by a ADC or a DAC, it is desirable that the output of the DAC is a faithful representation of the original signal. If the input signal is changing much faster than the sample rate, then this will not be the case, and spurious signals called aliases will be produced at the output of the DAC. This problem is called aliasing.

To avoid aliasing, the input to an ADC is often filtered to prevent it from changing faster than the sample rate. This filter is called an anti-aliasing filter.

SUCCESSIVE APPROXIMATION CONVERTER:

A successive approximation converter uses a comparator and counting logic to perform a conversion. The first step in the conversion is to see if the input is greater than half the reference voltage. If it is, the most significant bit (MSB) of the output is set. This value is then subtracted from the input, and the result is checked for one quarter of the reference voltage. This process continues until all the output bits have been set or reset. A successive approximation ADC takes as many clock cycles as there are output bits to perform a conversion. These convert very fast, and have good resolutions and quite wide ranges.

UNDERSTANDING SAR ADC'S:

Successive-approximation-register (SAR) analog-to-digital converters (ADCs) represent the majority of the ADC market for medium to high resolution ADCs. SAR ADCs provide up to 5Msps sampling rates with resolutions from 8 to 18 bits. The SAR architecture allows for high performance, low power ADCs to be packaged in small form factors for today's demanding applications.

Successive-approximation-register (SAR) analog-to-digital converters (ADCs) are frequently the architecture of choice for medium-to-high-resolution applications with sample rates under 5 megasamples per second (Msps). SAR ADCs most commonly range in resolution from 8 to 16 bits and provide low power consumption as well as a small form factor. This combination makes them ideal for a wide variety of applications, such as portable/battery-powered instruments, pen digitizers, industrial controls, and Data/signal acquisition.As the name implies, the SAR ADC basically implements a binary search algorithm. Therefore, while the internal circuitry may be running at several megahertz (MHz), the ADC sample rate is a fraction of that number due to the successive-approximation algorithm.

SAR ADC ARCHITECTURE:

Although there are many variations in the implementation of a SAR ADC, the basic architecture is quite simple (see Figure 1). The analog input voltage (V_{IN}) is held on a track/hold. To implement the binary search algorithm, the N-bit register is first set to midscale (that is, 100... .00, where the MSB is set to '1'). This forces the DAC output

(V_{DAC}) to be $V_{REF}/2$, where V_{REF} is the reference voltage provided to the ADC.

A comparison is then performed to determine if V_{IN} is less than or greater than V_{DAC}. If V_{IN} is greater than V_{DAC}, the comparator output is a logic high or '1' and the MSB of the N-bit register remains at '1'. Conversely, if V_{IN} is less than V_{DAC}, the comparator output is a logic low and the MSB of the register is cleared to logic '0'. The SAR control logic then moves to the next bit down, forces that bit high, and does another comparison. The sequence continues all the way down to the LSB. Once this is done, the conversion is complete, and the N-bit digital word is available in the register.

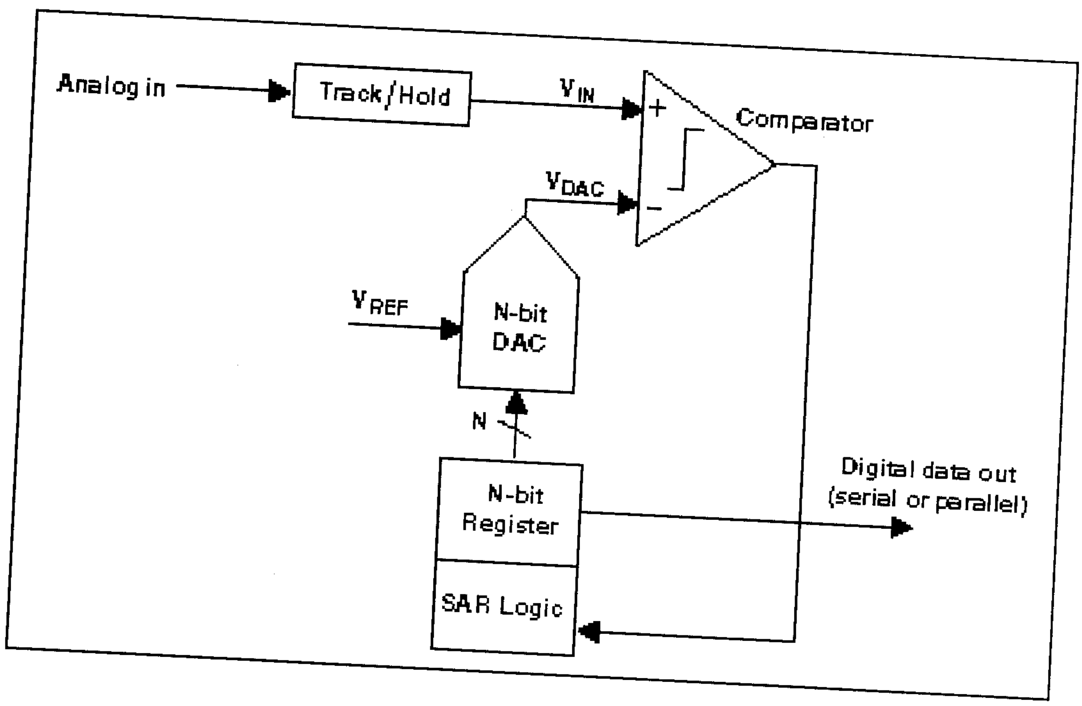


Figure 1. Simplified N-bit SAR ADC architecture

Figure 2 shows an example of a 4-bit conversion. The y-axis (and the bold line in the figure) represents the DAC output voltage. In the example, the first comparison how that $V_{IN} < V_{DAC}$. Thus, bit 3 is set to '0'. The DAC is then set to 0100_2 and the second comparison is performed. As $V_{IN} > V_{DAC}$, bit 2 remains at '1'. The DAC is then set to 0110_2, and the third comparison is performed. Bit 1 is set to '0', and the DAC is then set to 0101_2 for the final comparison. Finally, bit 0 remains at '1' because $V_{IN} > V_{DAC}$.

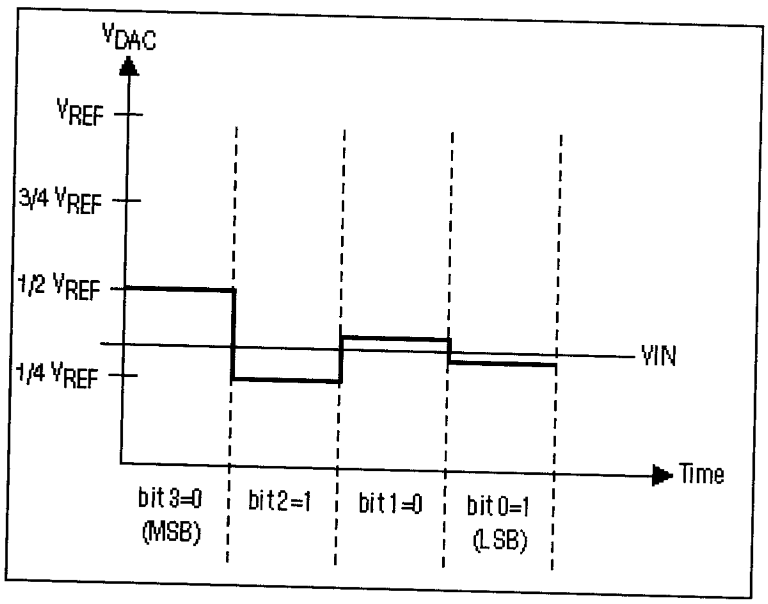
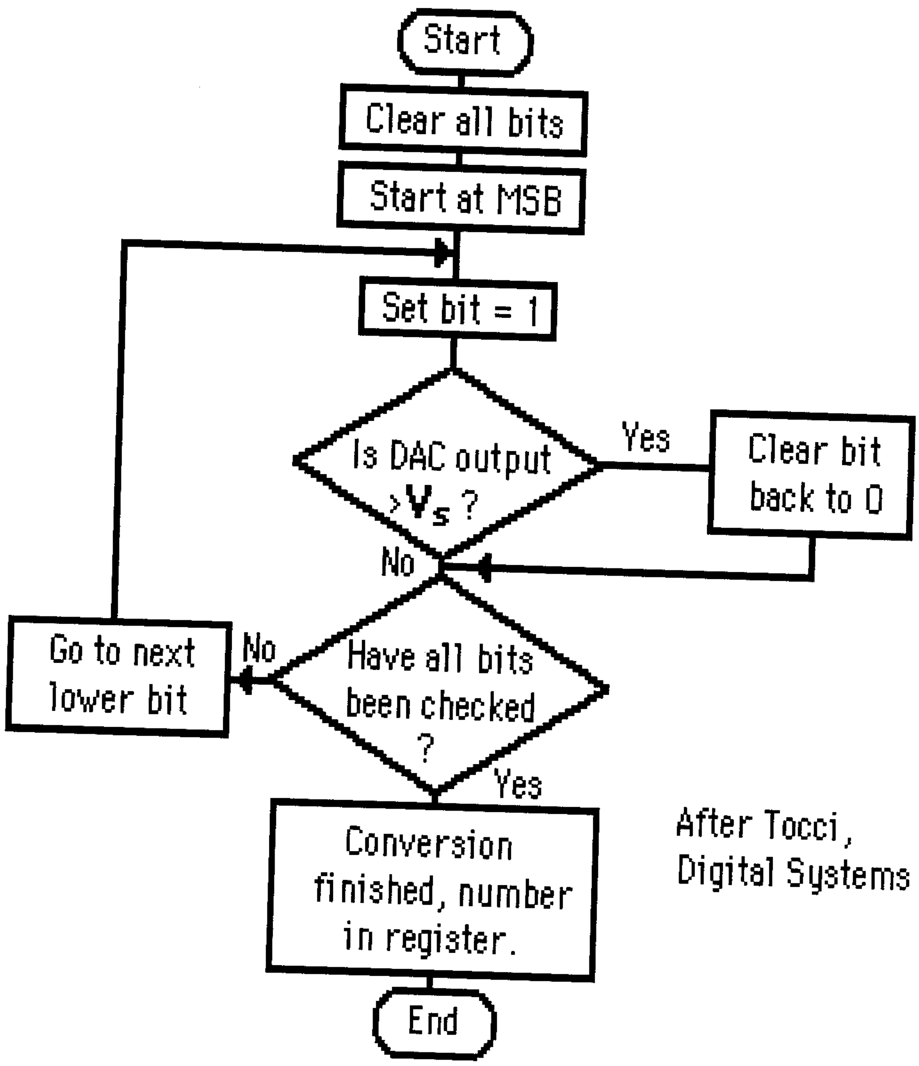


Figure 2. SAR operation (4-bit ADC example)

Notice that four comparison periods are required for a 4-bit ADC. Generally speaking, an N-bit SAR ADC will require N comparison periods and will not be ready for the next conversion until the current one is complete. This explains why these types of ADCs are power- and space-efficient, yet are rarely seen in speed-and-resolution combinations beyond a few Msps at 14 to 16 bits. One other feature of SAR ADCs is that power dissipation scales with the sample rate, unlike flash or pipelined ADCs, which usually have constant power dissipation versus sample rate. This is especially useful in low-power applications or applications where the data acquisition is not (continuous for example, PDA digitizers).

FLOWCAHRT FOR SAR ADC:



ADVANTAGES OF SAR ADC:

In summary, the primary advantages of SAR ADCs are low power consumption, high resolution and accuracy, and a small form factor. Because of these benefits, SAR ADCs can often be integrated with other larger functions. The main limitations of the SAR architecture are the lower sampling rates and the requirements for the building blocks (such as the DAC and the comparator) to be as accurate as the overall system.

CONTROL WORD:

ADCS1	ADCS0	CHS2	CHS1	CHS0	ADGO	-	ADON
BIT:7	BIT:6	BIT:5	BIT:4	BIT:3	BIT:2	BIT:1	BIT:0

BIT 7:6 ADCS1:ADCS0:A/D conversion clock select bits

00 = Fosc/2

01 = Fosc/8

10 = Fosc/32

11 = FRC(clock derived from the internal A/D RC oscillator)

BIT 5:3 CHS2:CHS0:Analog channel select bits

000 = channel 0,(AN0) (RA0)

001 = channel 1,(AN1) (RA1)

010 = channel 2,(AN2) (RA2)

011 = channel 3,(AN3) (RA3)

100 = channel 4,(AN4) (RA4)

101 = channel 5,(AN5) (RA5)

110 = channel 6,(AN6) (RA6)

111 = channel 7,(AN7) (RA7)

BIT 2 GO/DONE:A/D conversion status bit

1 = A/D conversion in progress

0 = A/D conversion not in progress

BIT 1 Reserved. Always maintain this bit cleared.

BIT 0 ADON:A/D on bit.

1 = A/D converter module is operating

0 = A/D converter module is shut off and consumes no operating current.

ADCON1 REGISTER:

ADFM	-	-	-	PCFG3	PCFG2	PCFG 1	PCFG0
BIT:7	BIT:6	BIT:5	BIT:4	BIT:3	BIT:2	BIT:1	BIT:0

BIT 7:6 Unimplemented : Read as ‘0’

BIT 5 ADFM:A/D result format select

1 = Right justified. 6 most significant bits of ADRESH are read as ‘0’.

0 = Left justified. 6 least significant bits of ADRESL are read as ‘0’.

BIT 4 Unimplemented: Read as ‘0’.

BIT 3:0 PCFG3:PCFG0:A/D port configuration control bits.

LOGIC DESCRIPTION

LOGIC DESCRIPTION

The overall logic description of the data acquisition system with printer interface is as follows:

The data from the acquisition system (Temperature, Voltage and Current) is first sampled and these analog values are first converted to Digital using the A/D conversion module programmed in the PIC.These sampled values are then stored temporarily in the buffer which is interfaced with the Epson dotmatrix printer through the printer device driver.The printer device driver controls all the actions of the printer apart from interfacing.

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 setting of RC2 pin by the printer activates the BUSY signal. If the printer is busy, the pin is low. If the printer is not busy the status of the pin is active high.

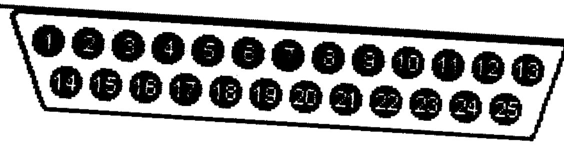
When the printer is ON the busy status is checked. The STROBE signal is sent to the printer. If the printer is free the contents of memory is transferred to the printer, byte by byte , by configuring the port A,B and C pins respectively.Before the transfer of data,the printout format is moved to the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

PRINTER INTERFACE

PRINTER INTERFACE

25 pin Parallel Port Pinout Details

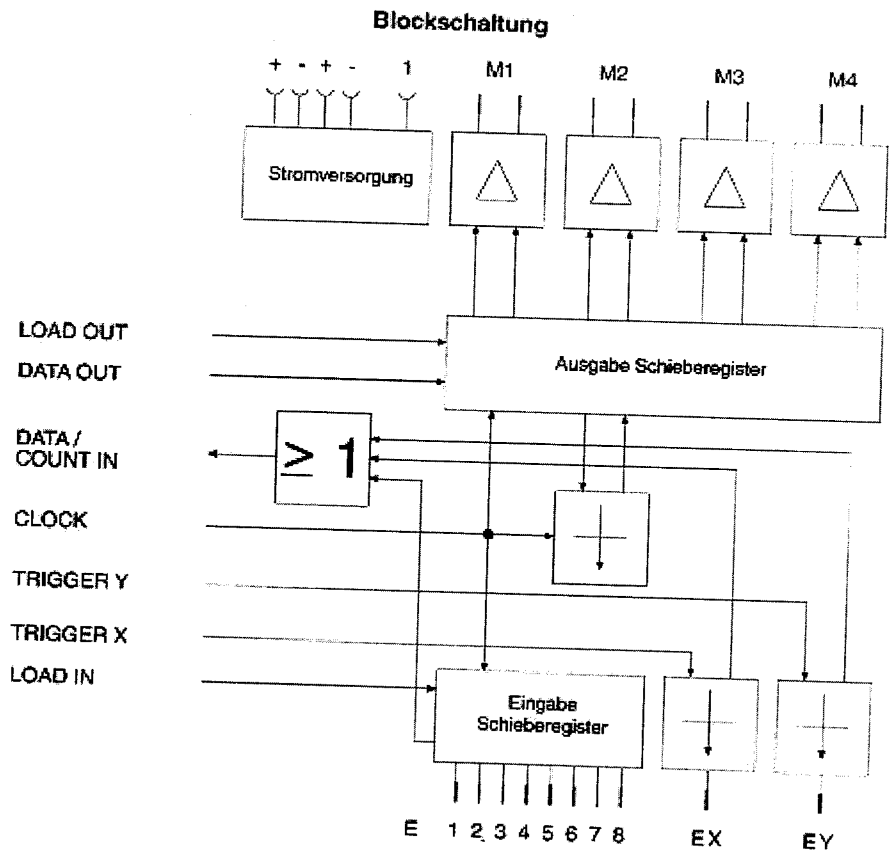
25 pin Parallel Port Connector 25 pin "D" connector



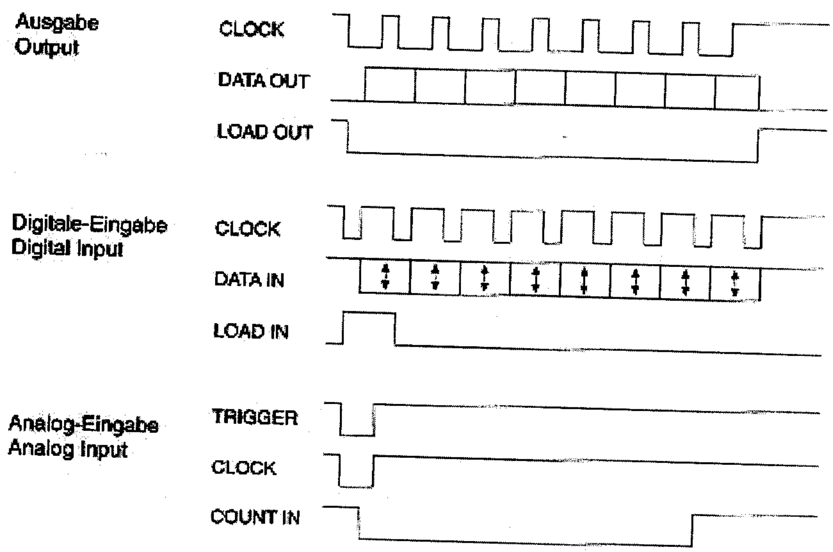
Connector may be reversed depending on which side is viewed. All pins are numbered.

Pin No.	Function	Pin No.	Function
1	Strobe	14	Auto Feed
2	Data 0	15	Error
3	Data 1	16	Init
4	Data 2	17	Select In
5	Data 3	18	Ground
6	Data 4	19	Ground
7	Data 5	20	Ground
8	Data 6	21	Ground
9	Data 7	22	Ground
10	Acknowledge	23	Ground
11	Busy	24	Ground
12	Paper Empty	25	Ground
13	Select		

Internal construction of the parallel interfaces



block diagram



Impulsdiagramm des fischertechnik Interface

pulse diagram

Parallel Port Specifications

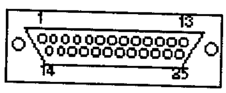
The PC parallel port usually consists of 25 pins in a DB-25 connector. These pins can interface to the TTL logic of an external device, either as inputs or outputs. Some pins can be used as inputs only, while some can be switched in software, on-the-fly, between input mode and output mode. Note, however, that it can be dangerous for two devices to assert an output value on the same line at the same time. Devices that are using bidirectional pins must agree (somehow) on who is supposed to control the line at any given time.

From the point of view of Parapin, the pinout of the parallel port is as follows: Pin	Direction
1	In/Out
2-9	In/Out (see note)
10	Input, Interrupt Generator
11	Input
12	Input
13	Input
14	In/Out
15	Input
16	In/Out
17	In/Out
18-25	Ground

Pins 2-9--called the parallel port's ``Data Pins"--are *ganged*. That is, their directions are not individually controllable; they must be either all inputs or all outputs. (The actual *values* of the pins--on or off--*are* individually controllable.) Also, some of the oldest parallel ports do not support switching between inputs and outputs on pins 2-9 at all, so pins 2-9 are always outputs. Many PC motherboards allow the user to select the personality of the parallel port in the BIOS. If you need to use pins 2-9 as bidirectional or input pins, make sure your port is configured as a ``PS/2" port or one of the other advanced port types; ports in SPP (Standard Parallel Port) mode may not support direction switching on pins 2-9.

Pin 10 is special because it can generate interrupts. Interrupt handling is discussed in more detail in Section 9.

Output pins can assert either a TTL high value (between +2.4v and +5.0v), or a TTL low (between 0v and +0.8v). The port can not source much current. Specs for different implementations vary somewhat, but a safe assumption (staying within spec) is that the voltage will be at least 2.5v when drawing up to 2.5mA. In reality you can sometimes get away with using an output pin to power a component that uses 3v or even 5v logic, but the supplied voltage may sag if more than 2.5mA is drawn. Input pins are typically spec'ed to be able to sink up to about 20mA. For a more detailed treatment of the parallel port electronics, see the references above. These two figures illustrate the pin assignments on the 25 pin connector and the bit assignments on the three ports.



View is looking at Connector side of DB-25 Male Connector.

Pin	Description	
1	Strobe	PC Output
2	Data 0	PC Output
3	Data 1	PC Output
4	Data 2	PC Output
5	Data 3	PC Output
6	Data 4	PC Output
7	Data 5	PC Output
8	Data 6	PC Output
9	Data 7	PC Output
10	ACK	PC Input
11	Busy	PC Input
12	Paper Empty	PC Input
13	Select	PC Input
14	Auto Feed	PC Output
15	Error	PC Input
16	Initialize Printer	PC Output
17	Select Input	PC Output

Pin Assignments

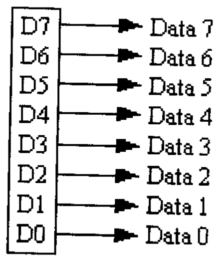
Note: 8 Data Outputs
4 Misc Other Outputs

5 Data Inputs

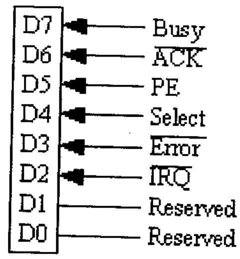
Note: Pins 18-25 are Ground

Fig 1. Pin Assignments

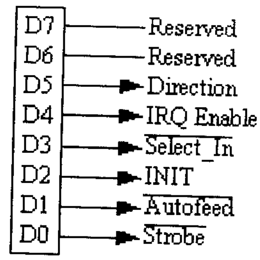
Data Port



Status Port



Control Port



CONCLUSION

CONCLUSION

The data interfacing to printer is done using PIC Microcontroller. The data from the acquisition system are transmitted to the printer through the parallel ports and is printed at particular instants of time.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.

PIC is programmed for A/D conversion which converts the analog samples to digital values, which are then printed using the printer interface. It is required that the printer should print data only when needed by the user.The interface is specially designed for parallel printing.

This interface could efficiently handle data without loss or redundancy.It can interact effectively with the system and the printer, handle interrupts, timers, speed compatibility and low power consumption.

FUTURE ENHANCEMENTS

SCOPE FOR FUTURE ENHANCEMENTS

The proposed system uses an Epson Dotmatrix printer that is interfaced with the system to get the instantaneous values of the inputs. The PIC could also be programmed to interface any type of printer with the acquisition system and moreover serial type of transmission between the printer and the system could be made possible

BIBILIOGRAPHY

BIBILIOGRAPHY

BOOKS

- Embedded systems Design – An introduction to process, tools and techniques – Arnold S. Berger, CMP books
- Programming Embedded systems in C and C++ - Michael barr, O'Reilly, SPD
- Design with PIC micro controller – John B.Peatman, Pearson education, Asia

JOURNALS

Sumeet Kumar, " Hardware fundamentals for Embedded software", Information Technology

B.Kainka, " PC peripheral design", Elecktroindia

WEBSITES

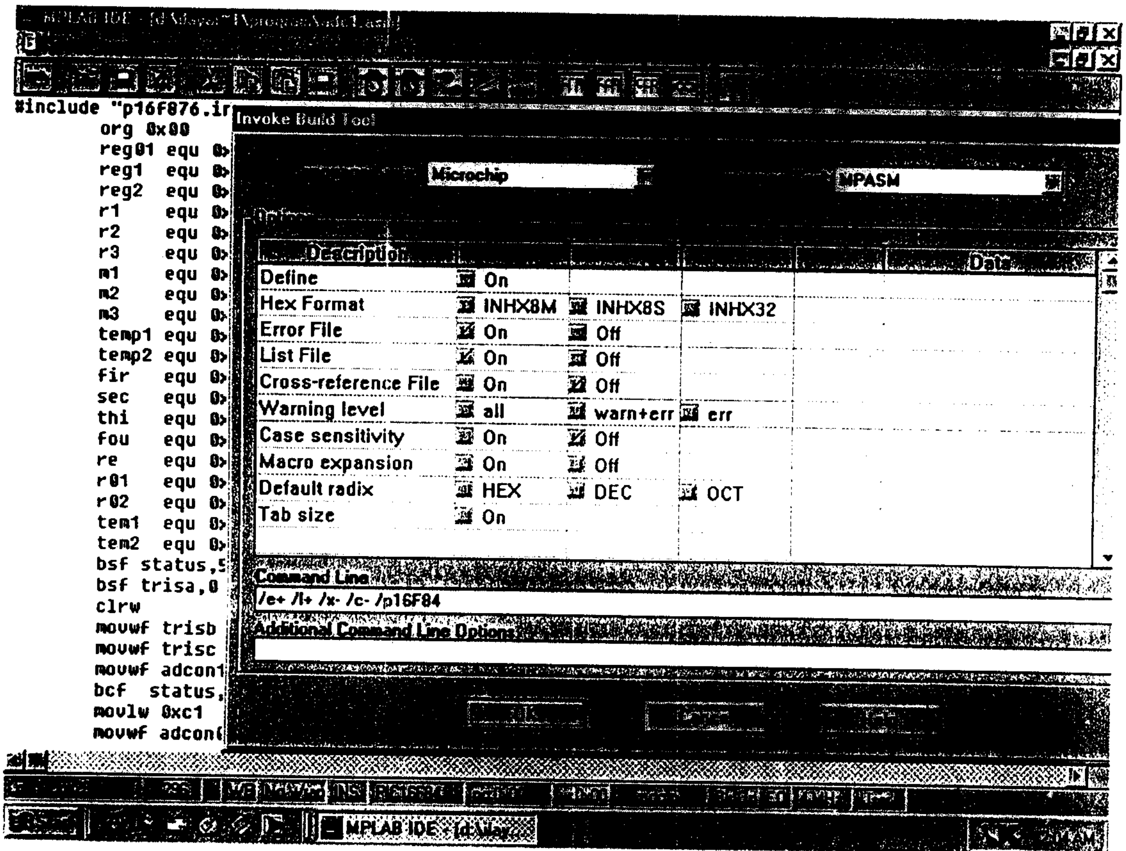
- www.embedded-zone.com
- www.embedded.com
- www.microchip.com
- www.elecappliances.com

APPENDICES

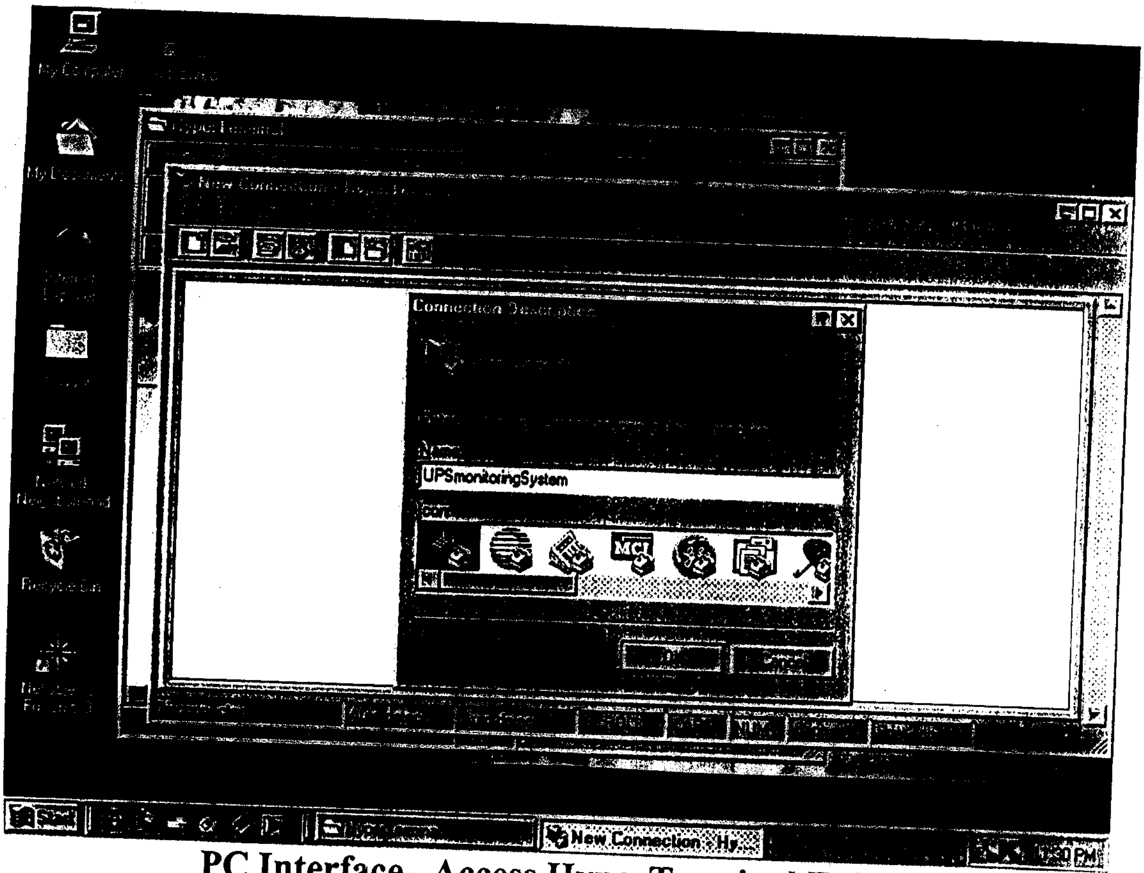
SCREEN SHOTS

APPENDIX D

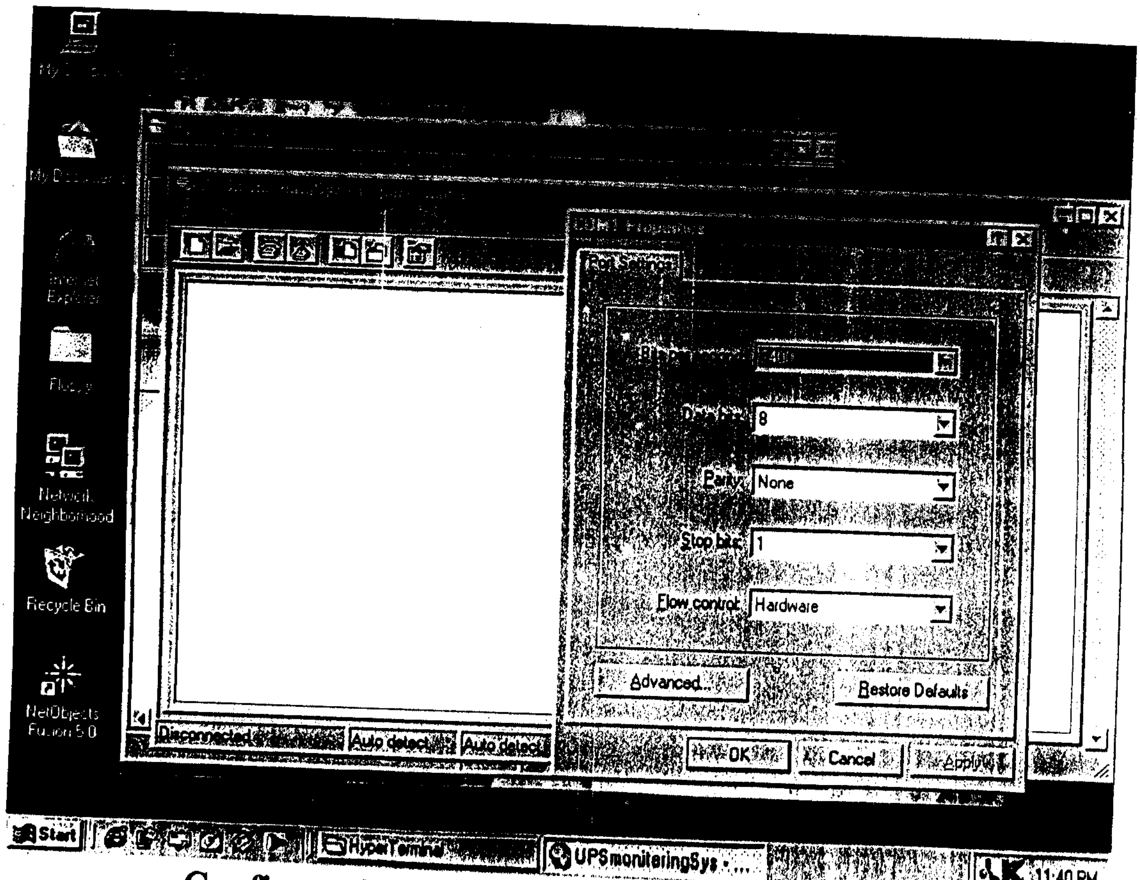
Program Simulation using MPLAP IDE



Program error checking



PC Interface –Access HyperTerminal Editor



Configuration setting for hyper terminal editor

DATA SHEETS

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

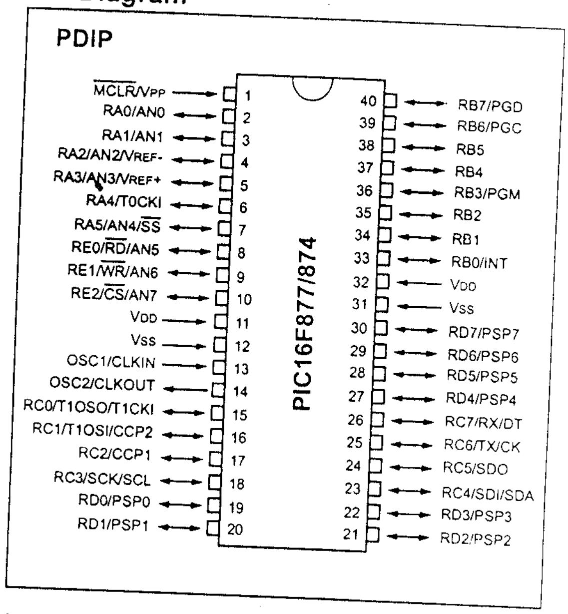
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
  Up to 368 x 8 bytes of Data Memory (RAM)
  Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
  Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
  - < 0.6 mA typical @ 3V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I^{2}C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

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

1.0 DEVICE OVERVIEW

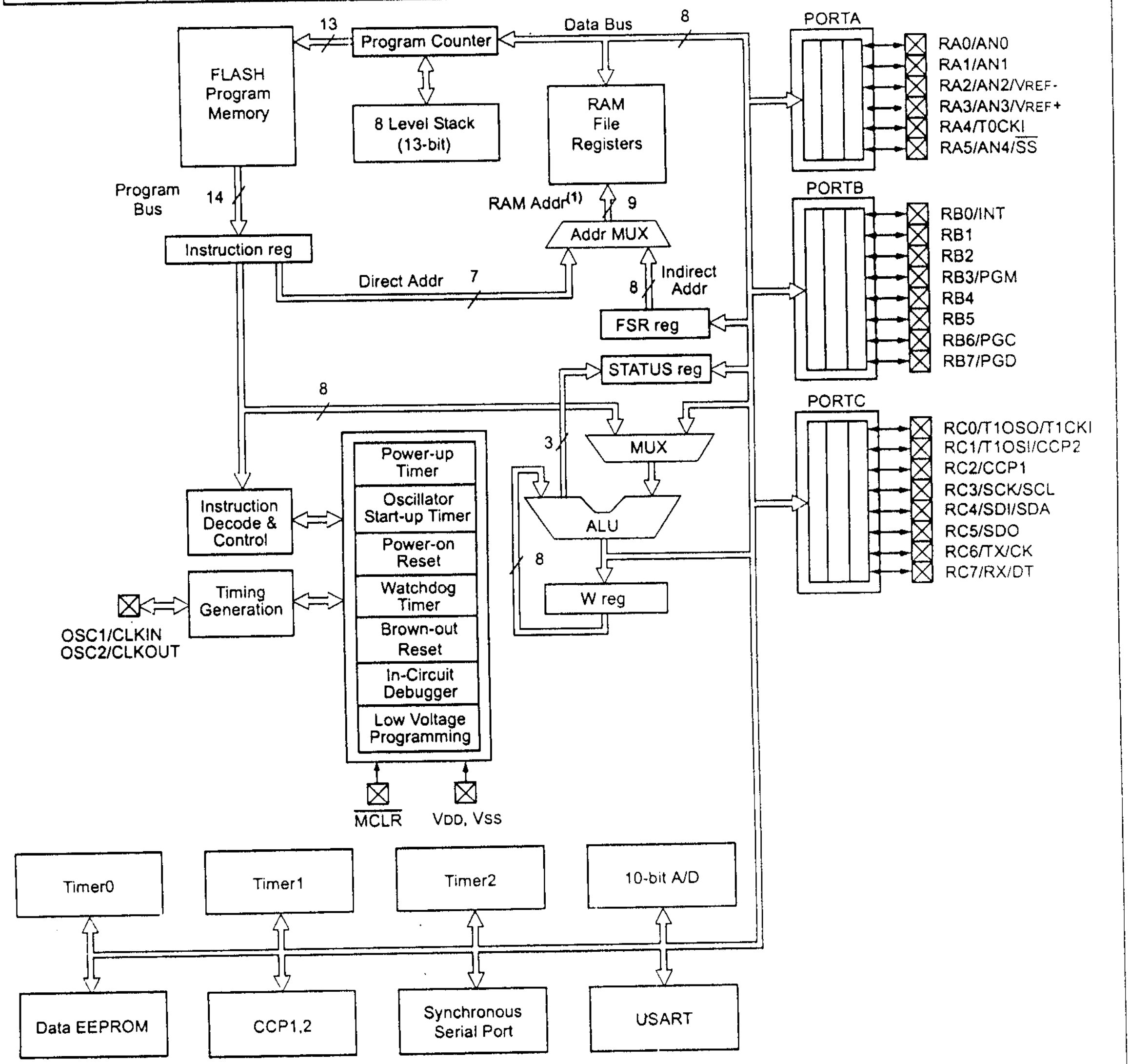
This document contains device specific information. Additional information may be found in the PICmicro™ Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. The Reference Manual should be considered a complementary 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

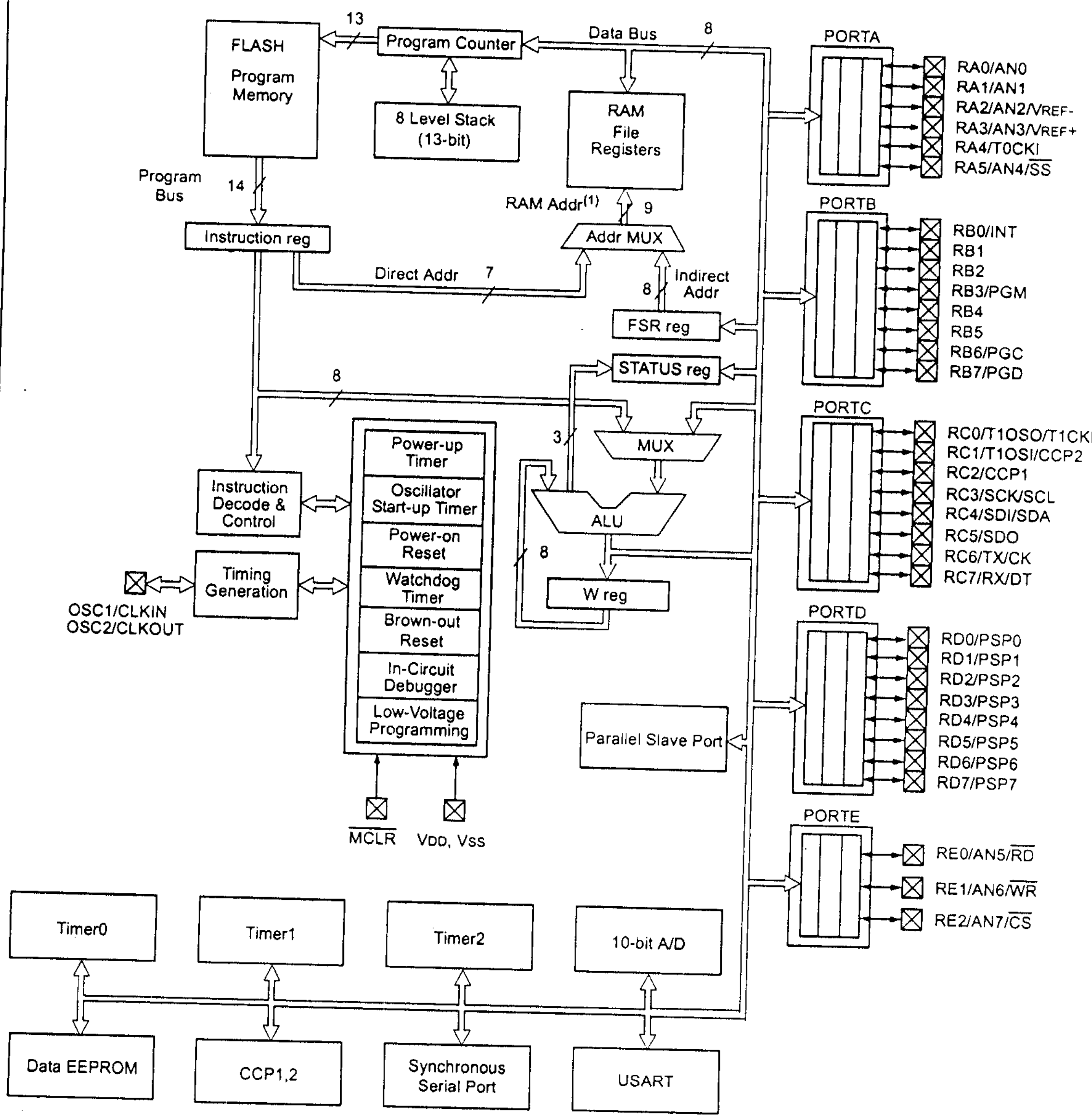
Device	Program FLASH	Data Memory	Data EEPROM
PIC16F873	4K	192 Bytes	128 Bytes
PIC16F876	8K	368 Bytes	256 Bytes



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

FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM

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



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

TABLE 1-1: 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/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.

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION

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

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

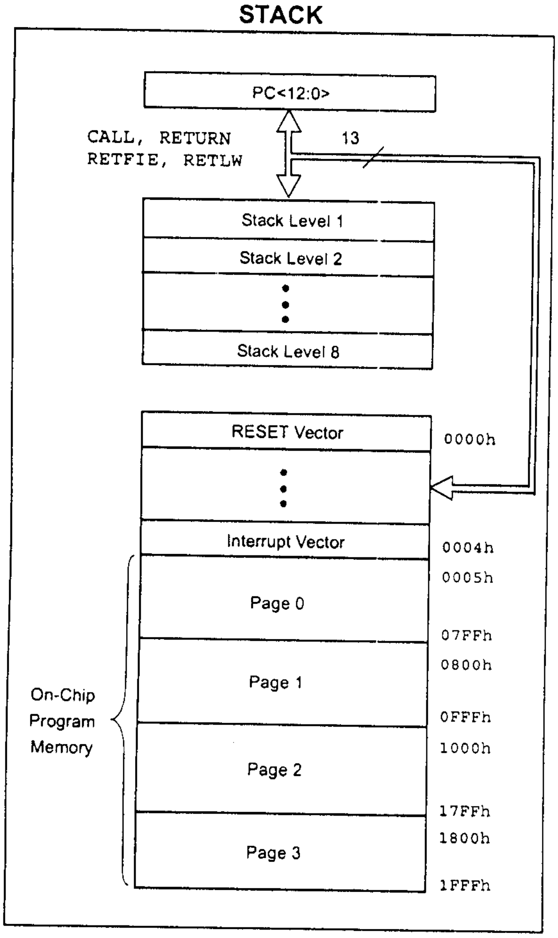
Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port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

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).

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



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

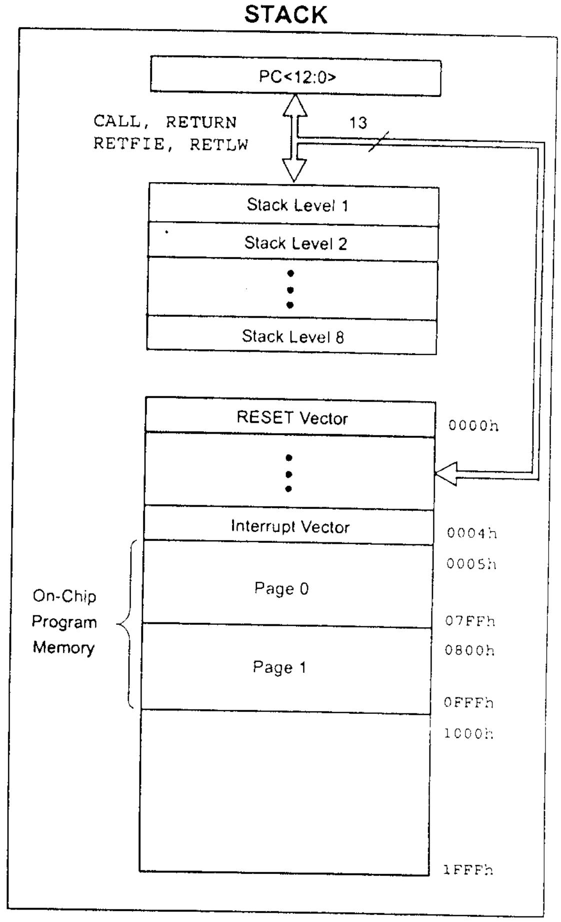


FIGURE 2-3: PIC16F877/876 REGISTER FILE MAP

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

▒ Unimplemented data memory locations, read as '0'.
* Not a physical register.

Note 1: These registers are not implemented on the PIC16F876.
2: These registers are reserved, maintain these registers clear.

FIGURE 2-4: PIC16F874/873 REGISTER FILE MAP

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

Bank 0 | Bank 1 | Bank 2 | Bank 3

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

* Not a physical register.

Note 1: These registers are not implemented on the PIC16F873.

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

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).

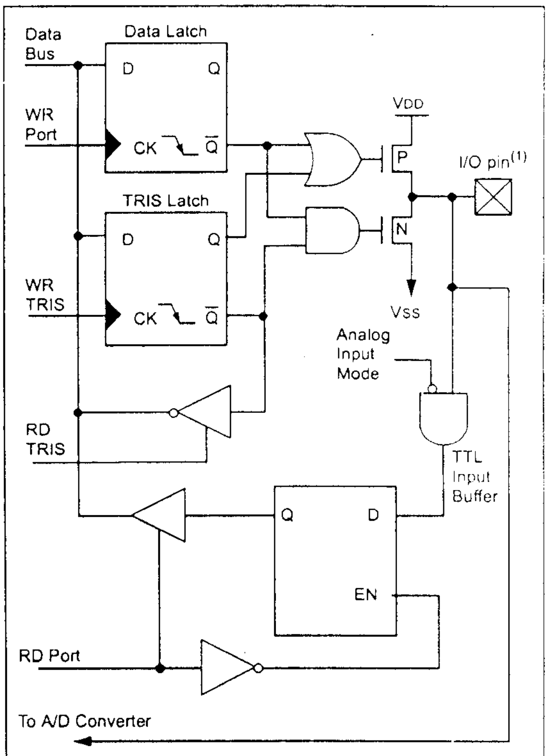
Note:	On a Power-on Reset, these pins are configured as analog inputs and read as '0'.

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

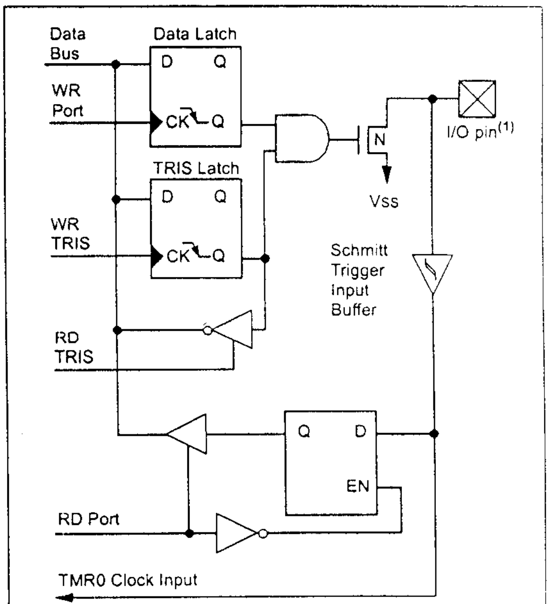
```
BCF     STATUS, RP0   ;
BCF     STATUS, RP1   ; Bank0
CLRF    PORTA         ; Initialize PORTA by
                      ; clearing output
                      ; data latches
BSF     STATUS, RP0   ; Select Bank 1
MOVLW   0x06          ; Configure all pins
MOVWF   ADCON1        ; as digital inputs
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.

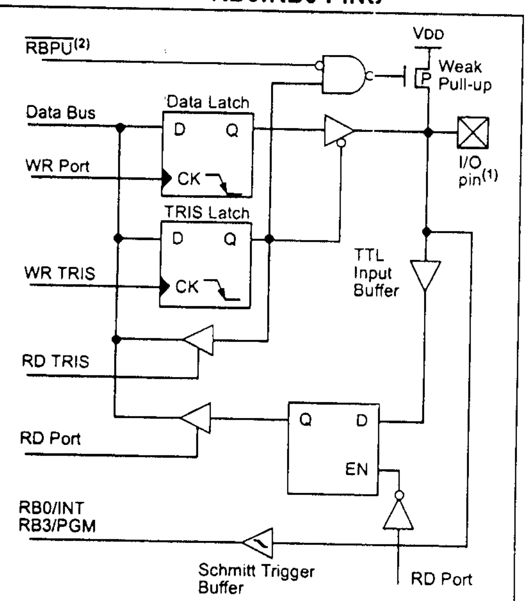
3.2 PORTB and the TRISB Register

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.

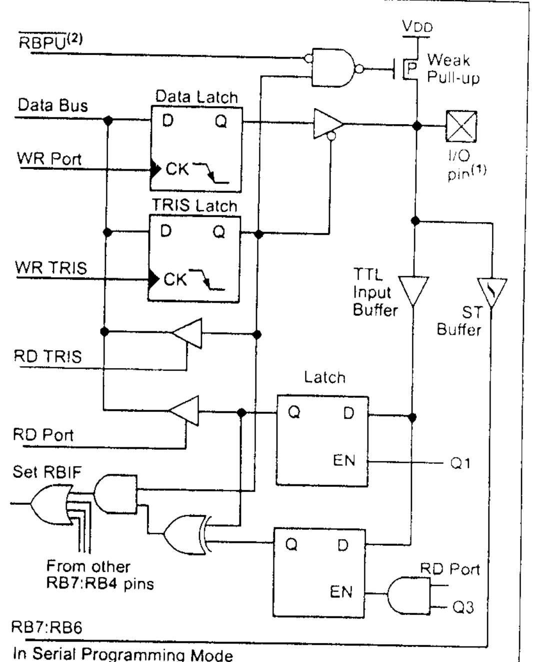
The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

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>).

3.3 PORTC and the TRISC Register

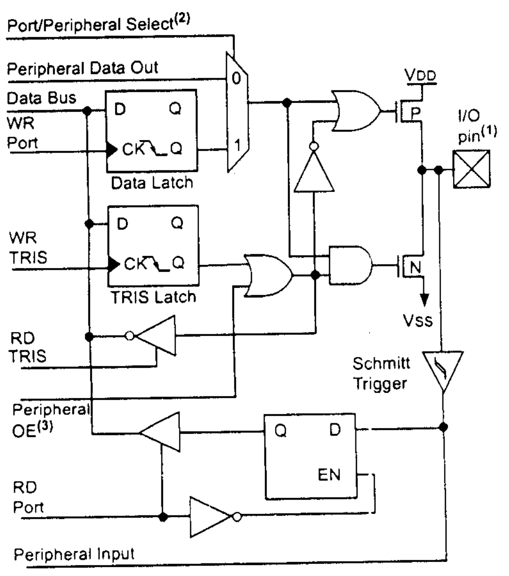
PORTC is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISC. Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin 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 I²C module is enabled, the PORTC<4:3> pins can be configured with normal I²C levels, or with SMBus levels by using the CKE bit (SSPSTAT<6>).

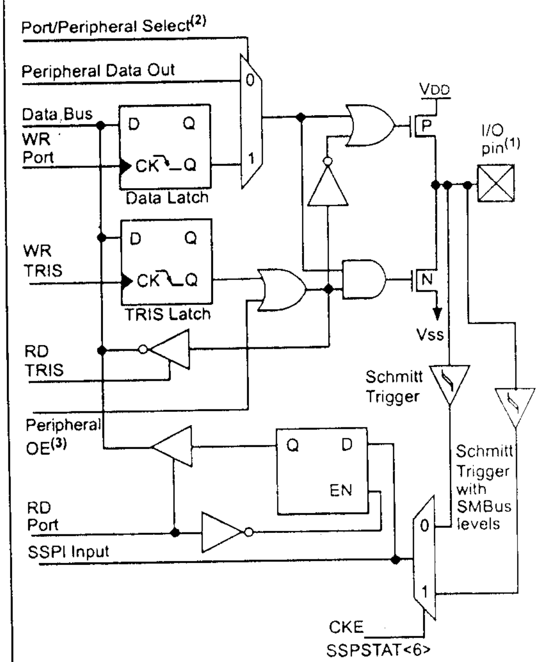
When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. Since the TRIS bit override is in effect while the peripheral is enabled, 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.