Introduction to Input Output

Input / Output Applications

Human Interface

- Lights and Switches
- Character and Pixel dot Matrix displays
- Keyboards

Communication with other Machines

- Printers
- Modems
- Other Instruments

Communication with the Environment

- Robotics
- Industrial Control

Overview of I/O methods

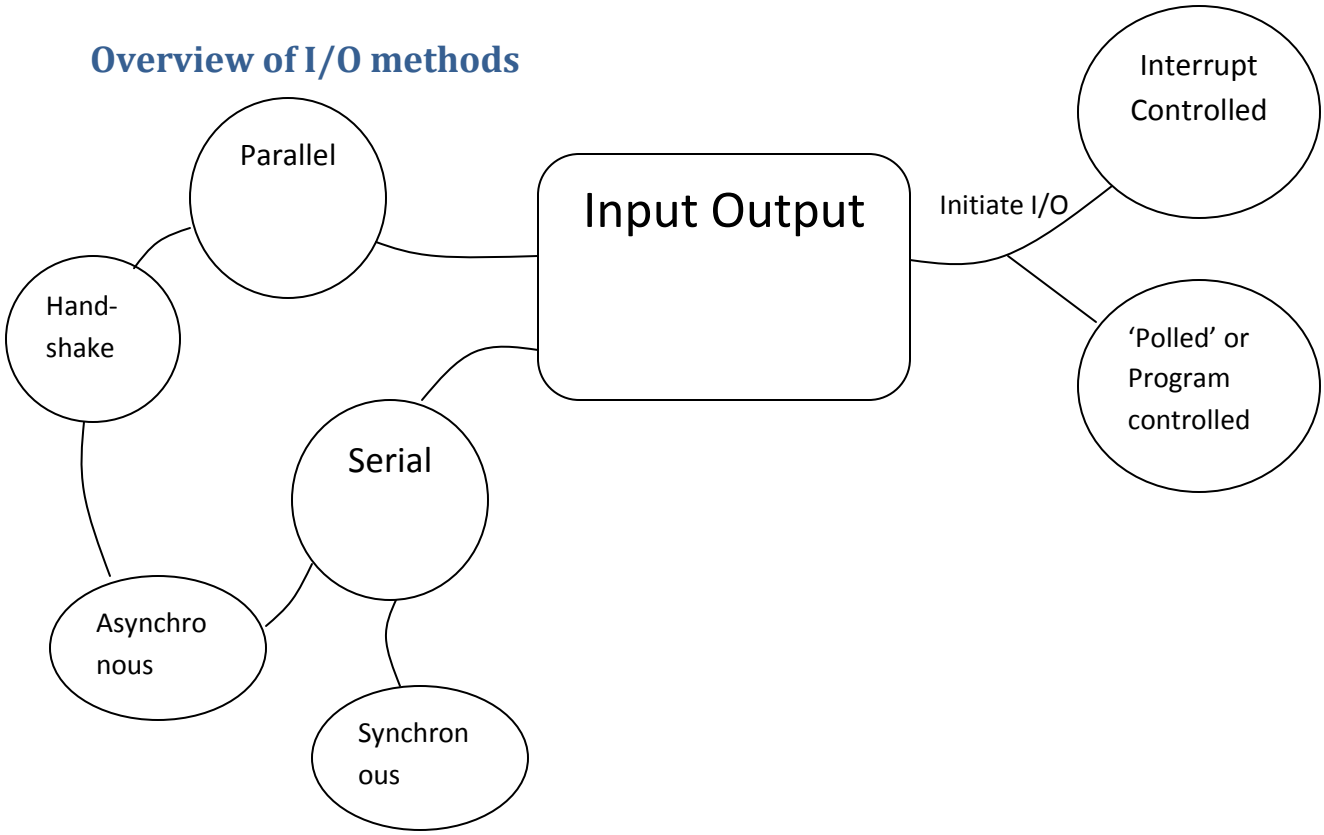


Figure 1 Classification of Input Output Methods

Parallel and Serial I/O

Parallel I/O

Data is transferred a word (typically a byte) at a time on a set of parallel connected digital interface wires. The data to be transferred is stored in a register at the microcomputer side . On the peripheral side the data may also be stored in a register, but may also be used to directly control the peripheral electronics. In parallel data ports the data is often synchronized using a separate control line or signal transmitted through another port. This approach is called HANDSHAKE.

Data is sent to a register on one side of the interface

A control line sends out a signal to the peripheral to indicate data is ready and to latch data out to the peripheral.

A status line indicates the readiness of the peripheral to received or transmit data or if the data has been received or transmitted

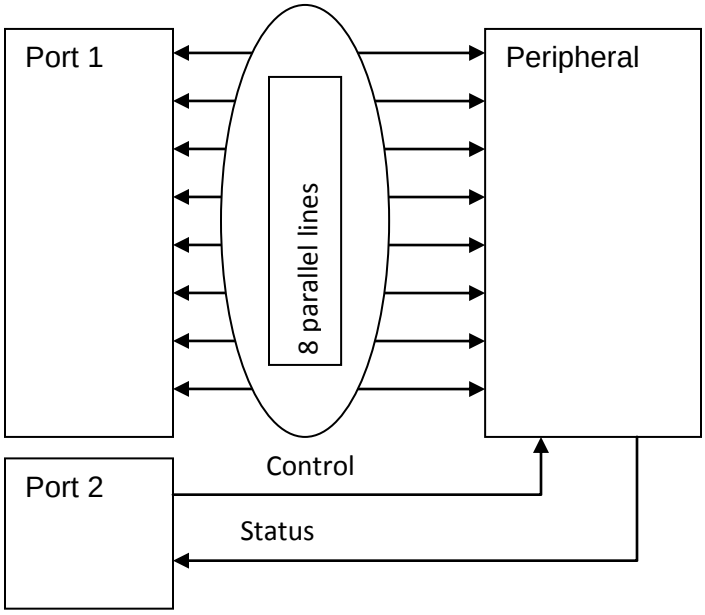


Figure 2 Handshake control for parallel I/O

Serial I/O

Data is transferred on a single wire (one wire for each direction), sequentially or serially one BIT at a time with words (or bytes) made up of a sequence of transmitted BITS. A s indicated above for the parallel data control lines may be used to synchronize the data transfer both in terms of status and control.

Additionally start and stop bits are used within the serial data to synchronize the data transfer.

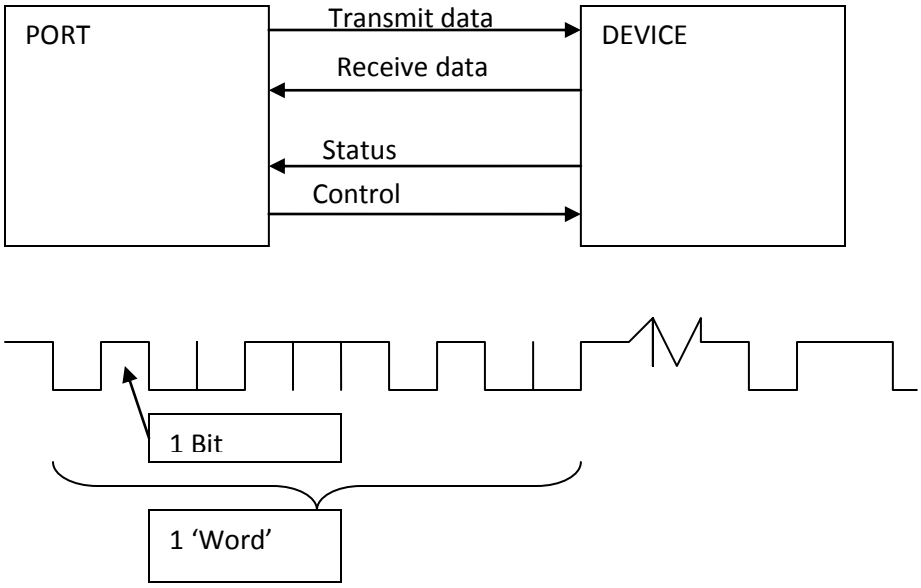


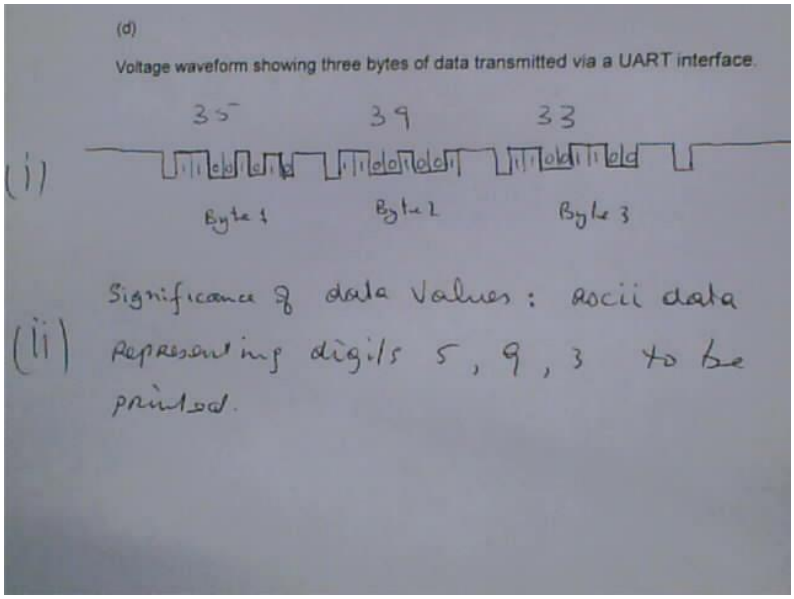
Figure 3 Asynchronous Serial with Handshake control

Lower hardware cost and therefore easier to implement multible peripheral busses. This leads to more protocol based interfaces. More standards available for serial interfaces

Asynchronous I/O

Serial data transfer where both the transmitter and the receiver have their own timing clock signals. Both sides have to be initially set up to the same data rate. Along with the serial word there is also sent a start bit and a stop bit to allow the receiver to synchroniseitself and know when to start reading the incoming data.

Voltage waveform showing three bytes of data transmitted via a UART interface.



Synchronous I/O

One of the sides of the interface controls the transmission by transmitting a clock or synchronising signal. The 'MASTER' transmits the timing signal on the separate timing connection. The 'SLAVE' reads this signal and uses it to synchronise its reading of the data

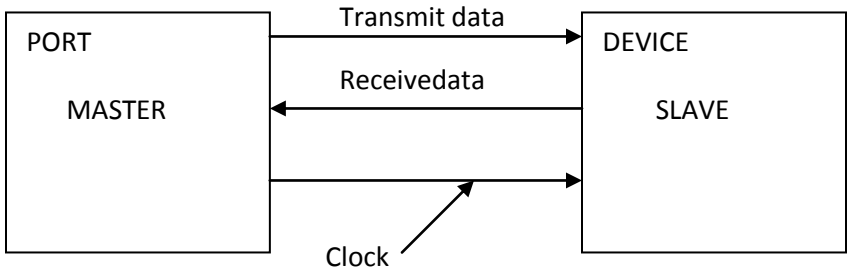


Figure 5 Synchronous Serial transfer

Input/Output Control

An I/O is initiated either by the computer or the peripheral. If the computer initiates then the I/O is program controlled.

If the peripheral initiates the either of two methods are used to start the transfer:

- Polled I/O
- Interrupt 'driven' I/O

Polled I/O.

Also called program controlled I/O, the program

1. Starts the I/O to a peripheral if required, eg updates/refreshes display
2. Checks each peripheral device in turn(handshake control line) to see if an I/O is required and acts accordingly.(Checks to see if Keyboard button has been pressed.

Interrupt Controlled

The peripheral Device initiates the I/O. When the device requires to send or receive data it activates an 'interrupt request' line/wire, connected to the processor. This change sets a flag bit in the processor which causes the computer to interrupt its 'normal' program and respond to the INTERRUPT REQUEST.

Summary of the Interrupt Process.

1. On power on or reset maskable interrupts are disabled. Must be enabled by instructions: setting bits in registers

2. Interrupt request – electrical signal from:
   a. External – keyboard, button, trip switch
   b. Internal – timer, ADC, Serial port.
3. MPU checks *flags* at end of each instruction.
4. MPU does the following:
   a. Completes current instruction
   b. Places [PC] on stack
   c. Disables interrupts
   d. (May) places other register contents on stack
   e. Gets new PC value from a specific memory location (*interrupt vector* pointing to address of *Interrupt Service Routine* ISR) and continues execution.
5. Executes ISR → *Return from Interrupt* (RTI) instruction.
6. RTI Instruction:
   a. Retrieves return address from stack → PC
   b. Retrieves other registers if necessary
   c. Enables Interrupts

Pic18F Interrupts.

External Sources

PORTB pins:

- RB0/INT0, RB1/INT1, RB2/INT2 -- Edge triggered.
- RB4 – RB7 -- Change in logic level.

Internal Sources

Timers, ADCs, Serial I/O. and more.

Interrupt Control

To set upthe Interrupt Response Process in the PIC18 the following Special Function Registers (SFRs') are used:

- RCON: Register Control
- INTCON: Interrupt Control
- INTCON1: Interrupt Control 1
- INTCON2: Interrupt Control 2
- PIR1 and PIR2: Peripheral Interrupt Registers 1 & 2
- PIE1 & PIE 2: Peripheral Interrupt Enable 1 & 2
- IPR1 & IPR2: Interrupt Priority Registers 1 & 2

RCON: Sets up the global priority for all interrupts.

IPEN			R	R	R	R	R

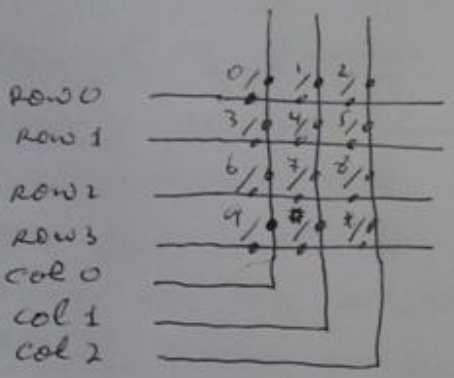
INTCON registers deal with External Interrupts.

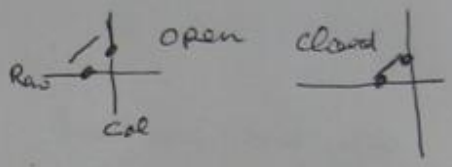
PIR, PIE and IPR handle Internal Interrupts.

Some Peripheral Devices

Keyboard

(a)

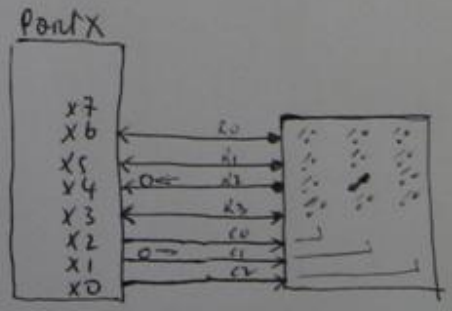




when a button is pressed one col wire is connected to one ROW wire

For 12 keys arranged in a matrix as shown each key is uniquely identified if it can be established which ROW is connected to which column.

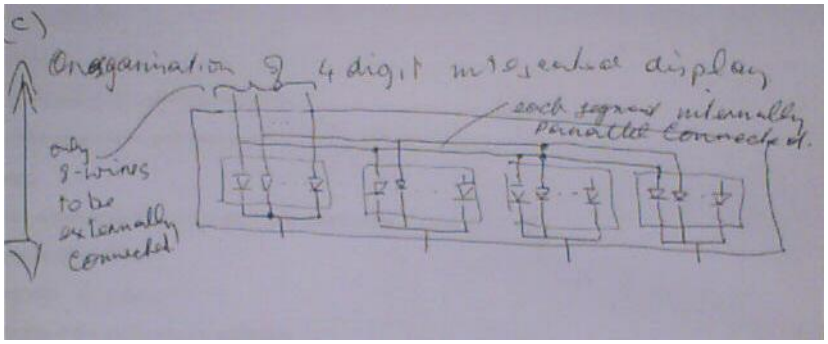
(b)

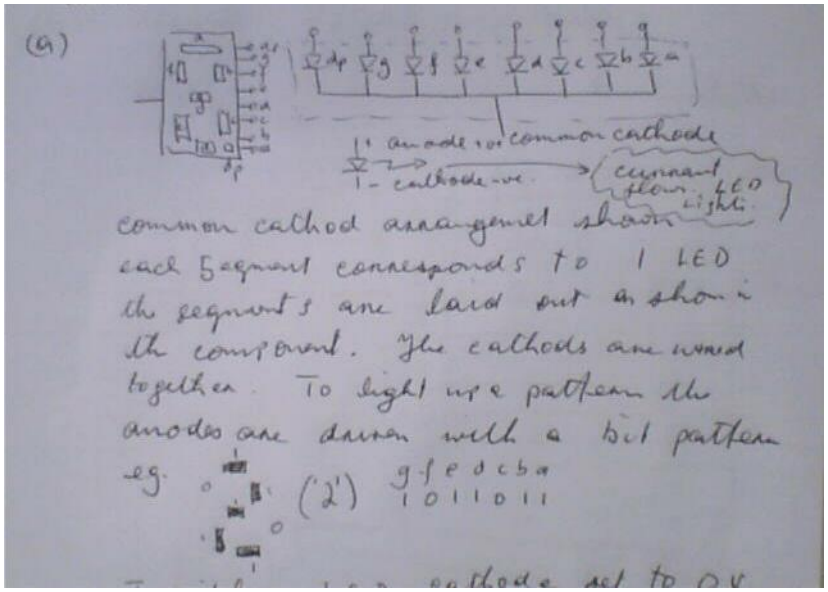


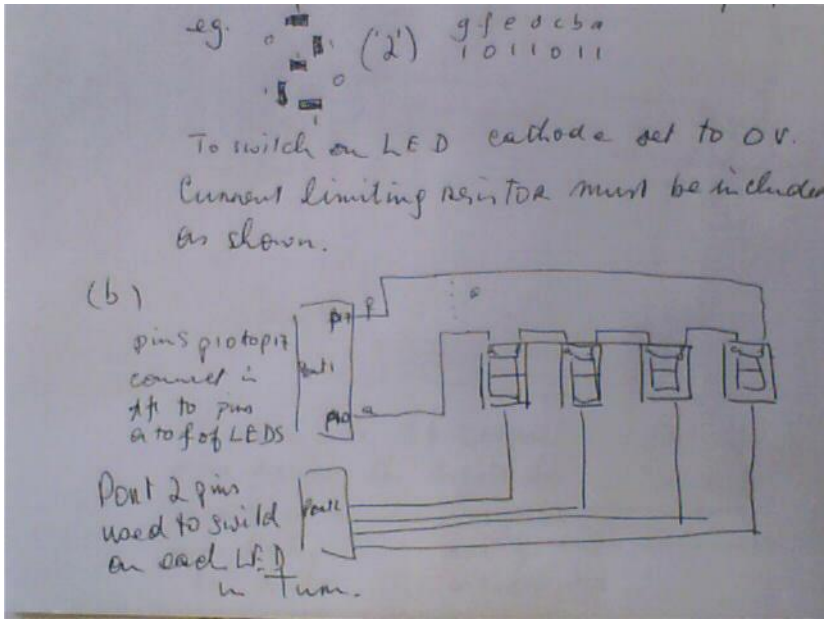
programmed as inputs

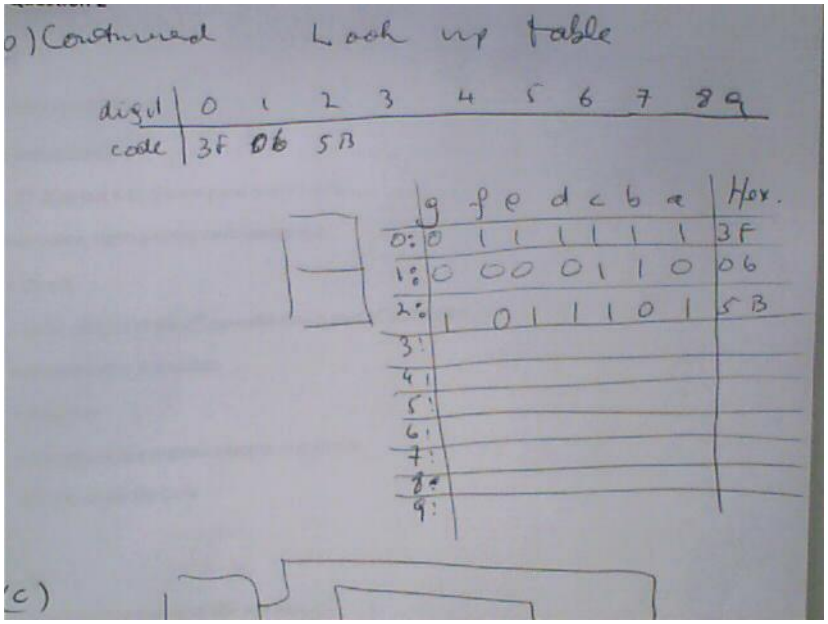
Four ROWs connected to pins X3 to X6, Internal pull up resistors hold these pins at +5 V. (logic 1).

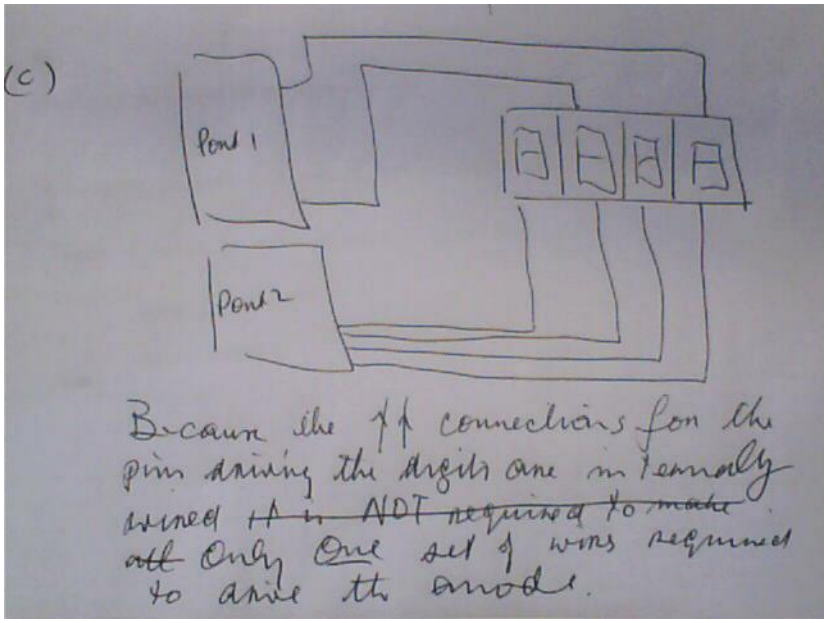
7-Segment LED Display

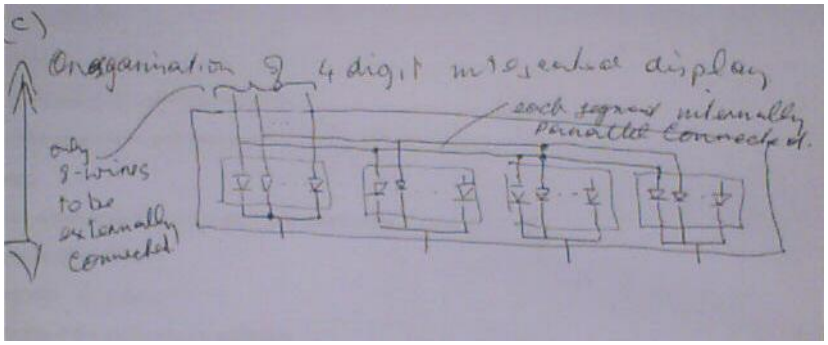







b) Continued Look up table
digit 0 1 2 3 4 5 6 7 8 9
code 3F 06 5B
g f e d c b a Hex.
0: 0 1 1 1 1 1 1 3F
1: 0 0 0 0 1 1 0 06
2: 1 0 1 1 0 1 1 5B
3:
4:
5:
6:
7:
8:
9:
(c)


(c)
Port 1
Port 2
Because the 4 connections for the pins driving the digits are internally ~~wired it is NOT required to make~~ all Only One set of wires required to drive the anode.


(c)
Organization of 4 digit multiplexed display
each segment internally parallel connected.
only 8-wires to be externally connected