IO & Storage System

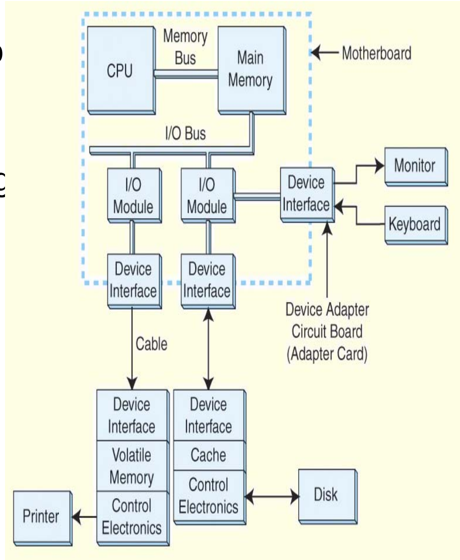
Lecture 19

I/O Architectures

- input/output components are seen as a subsystem of components that moves data between external devices CPU and main memory.
- I/O subsystems include:
  – Blocks of main memory that are devoted to I/O functions.
  – Buses that move data into and out of the system.
  – Control modules in the host and in peripheral devices
  – Interfaces to external components such as keyboards and disks.
  – Cabling or communications links between the host system and its peripherals.

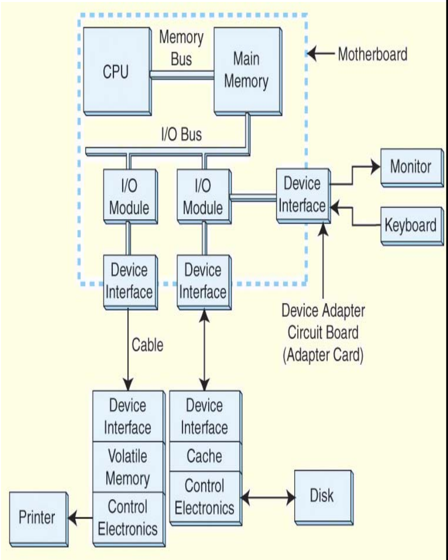
I/O Architectures

- Figure shows how all components can fit together to form an integrated I/O subsystem
- I/o module take care of moving data between main memory and a device interface
- Interfaces are designed to ensure communication between devices
  - Interface ensures that devices and host are ready to exchange data



I/O Architectures

- The exact form and meaning of the signals exchanged between sender and receiver is called "Protocol" and comprises
  - Command signals
  - Status signals
  - Data processing signals
- The handshaking is part of protocol

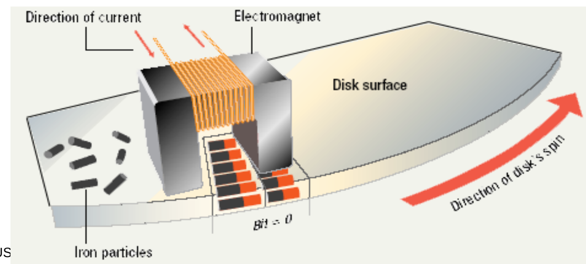


Storage Devices

- Purpose is to store data even when computer is off
- Storage of data involve two processes
  - Writing data Reading data
  - Transferring the data in computer memory
- Storage media is used to store the data on storage device
  - Media is the material storing data, Storage devices manage the media
    - Diskette is storage media while diskette drive is storage device
  - Magnetic storage devices uses a magnet
  - Optical devices use lasers
  - Solid-state devices have physical switches

Magnetic Storage Devices

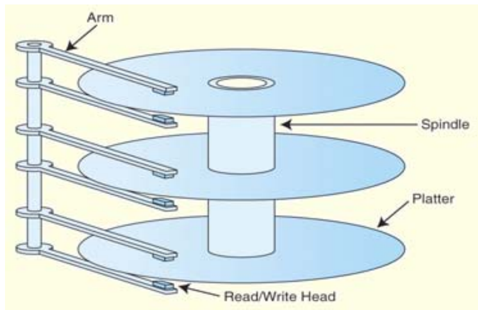
- Most common form of storage
  - Hard drives, floppy drives, tape
- All magnetic drives uses the same media and similar techniques to read and write
  - The surface of the diskette, hard disk are coated with magnetic sensitive material "iron oxide" that react to the magnetic field
- Diskette store data on both the sides
  - Each side has its own read/ write head
- Hard disk contains multiple disks called platter



Magnetic Disk Technology

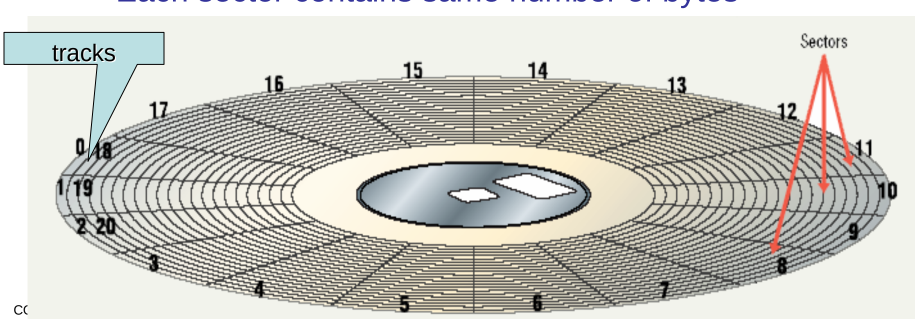
- Hard disk platters are mounted on spindles.
- Read/write heads are mounted on a comb that swings radially to read the disk.
  - Heads never touch the platter, they maintain a safe distance
    - If head touches the surface of the disk, the disk will become unusable, condition is known as head crash
  - Once the power is off, heads retreat to safe place known as parking of heads

- The rotating disk forms a logical cylinder beneath the read/write heads.
- Data blocks are addressed by their cylinder, surface, and sector.



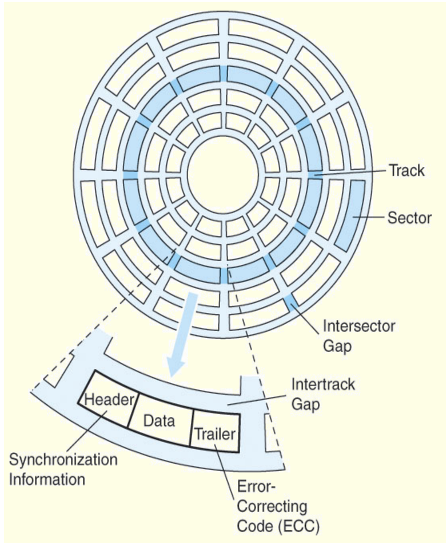
Magnetic Disk Technology

- Each unit of storage, Sector, has a unique address that can be accessed independently
  - blocks of data can be accessed according to their location on the disk.
- Sectors are division of concentric rings- tracks,
  - Every track contains exactly same sectors
  - Each sector contains same number of bytes



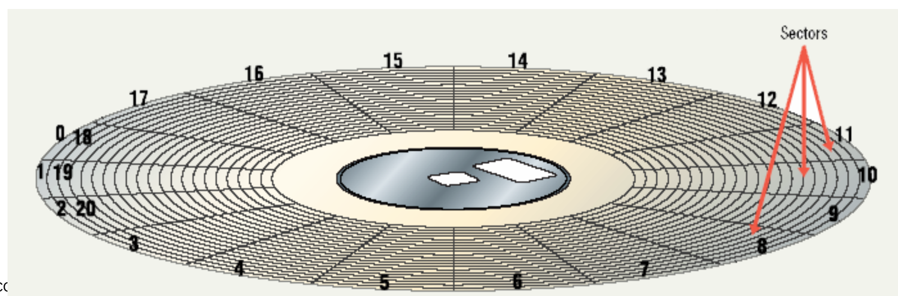
Magnetic Disk Technology

- Disk tracks are numbered,
  - starting with track zero at the outside edge.
- Sectors may not be in consecutive order
  - Sector has a unique address that can be accessed independently
- Each track has exactly the same number of sectors
- Each sector contains exactly the same bytes of data



Magnetic Storage Devices

- Data organization
  - Before use, the disk surface must be magnetically mapped - Formating
    - So as the head may move to any place on the disk
  - Formatting creates set of rings on the disk - tracks
  - Tracks are divided into pie shaped sectors
    - Data is physically stored on the sectors
    - Each sector is a smallest unit and numbered and store 512 byte



Magnetic Storage Devices

- Operating System can find data on disk
  - Each track and sector is labeled during formatting
  - Different Operating System format the disk in different way
    - File system – a logical method for managing storage
  - Common logical format of Windows are:
    - File Allocation Table (FAT)
      - FAT 16 – used by MS-DOS as bases of early Windows
      - FAT32 – introduced by Win-95, 2000 and XP
      - NTFS- introduced by Windows NT, better security
  - Data is organized in group of sectors “clusters”
    - OS sees the group of sectors as one single unit
    - Size of data the OS handles

File Allocation Table

- A table that an operating system maintains on Hard disk:
  - Provides a map of the clusters that a file has been stored in
  - When you write a new file to a hard disk,
    - The file is stored in one or more clusters that are not necessarily next to each other;
    - They may be rather widely scattered over the disk
    - A typical cluster size is 2,048 bytes, 4,096 bytes, or 8,192 bytes
  - The operating system creates a FAT entry for the new file that records where each cluster is located and their sequential order
    - When you read a file, the operating system reassembles the file from clusters and places it as an entire file where you want to read it

File Allocation Table

- File Allocation Table
  - **FAT12:**
    - The oldest type of FAT uses a 12-bit binary number to hold the cluster number
      - A volume formatted using FAT12 can hold a maximum of 4,086 clusters, which is 2^12 minus a few values (to allow for reserved values to be used in the FAT)
    - FAT12 is therefore most suitable for very small volumes, and is used on floppy disks and hard disk partitions smaller than about 16 MB

File Allocation Table

- File Allocation Table
  - **FAT32:**
    - The newest FAT type, FAT32 is supported by newer versions of Windows.
      - FAT32 uses a 28-bit binary cluster number--*not* 32, because 4 of the 32 bits are "reserved".
    - 28 bits is still enough to permit ridiculously huge volumes
    - FAT32 can theoretically handle volumes with over 268 million clusters, and will support (theoretically) drives up to 2 TB in size.
      - However to do this the size of the FAT grows very large

Magnetic Disk Technology

- There are a number of electromechanical properties of hard disk drives that determine how fast its data can be accessed.
  - *Seek Time: (Locate the* track)
    - *The amount of time required for the read/write heads to move between tracks over the surfaces of the platters.*
      - The time that it takes for a disk arm to move into position over the desired cylinder.
      - Normally expressed in milliseconds
    - It does not includes the time that it takes for the head to read the disk directory, that maps the logical file information

Magnetic Disk Technology

- *Rotational delay*: (*Locate the Sector*)
  - The time it takes for the desired sector to move into position beneath the read/write head.
- Access Time
  - The sum of the rotational delay and the seek time is known as access time
    - Seek time + rotational delay = *access time*.
- Transfer Time (Fetch Block)
  - If the seek time is added with the time that it takes to actually read the data from the disk, the transfer time is get.
- Transfer time (Fetch Block) varies depending on how much data is read
  - *Transfer rate* gives us the rate at which data can be read from the disk

Magnetic Disk Technology

- *Average latency* is a function of the rotational speed:
  - It is a measure of the amount of time it takes for the desired sector to move beneath the R/W head, after the disk arm has positioned itself over the desired track

$$\frac{\frac{60\ \text{seconds}}{\text{disk rotation speed}} \times \frac{1000\ \text{ms}}{\text{second}}}{2}$$

- *The disk directory has to be read before every data read and write operation*
  - *The placement of head is important*
    - *It is placed to any sector on track zero*
      - *Head has to position first to the outermost track and to desired track and sector – introduce overhead*
    - *It may be placed in the middle track*
      - *Head will take less time to rea and write the data*

Magnetic Disk Technology

- On a standard 1.44MB floppy, the FAT is limited to nine 512-byte sectors.
  - There are two copies of the FAT
  - Each FAT is nine sector long.
- There are 18 sectors per track and 80 tracks on each surface of a floppy, for a total of 2880 sectors on the disk.
  - So each FAT entry needs at least 12 bits (2^{11}= 2048 < 2880 < 2^{12} = 4096) to point a cluster.
    - Thus, FAT entries for disks smaller than 10MB are 12 bits, and the organization is called FAT12.
    - In fact each FAT entry on a disk is 16 bit wide, so the organization is known as FAT 16

Magnetic Disk Technology

- The disk directory associates logical file names with physical disk locations.
  - Directories contain a file name and the file's first FAT entry.
- If the file spans more than one sector (or cluster), the FAT contains a pointer to the next cluster (and FAT entry) for the file.
  - The FAT is read like a linked list until the <EOF> entry is found.

FAT Index →	120	121	122	123	124	125	126	127
FAT Contents	97	124	<EOF>	1258	126	<BAD>	122	577

If a file we want to read starts at sector 121 in the FAT fragment

Example

- Suppose a disk drive has the following characteristics:
  - 4 surfaces
  - 1024 tracks per surface
  - 128 sectors per track
  - 512 bytes/sector
  - Track-to-track seek time of 5 milliseconds
  - Rotational speed of 5000 RPM.
    - a. What is the capacity of the drive?
    - b. What is the access time?

$$\frac{\dfrac{60 \text{ seconds}}{\text{disk rotation speed}} \times \dfrac{1000 \text{ ms}}{\text{second}}}{2}$$

Ans.

a. 4 surfaces x 1,024 tracks per surface x 128 sectors per track x 512 bytes/sector = 4x1024x128x512 = 256MB.

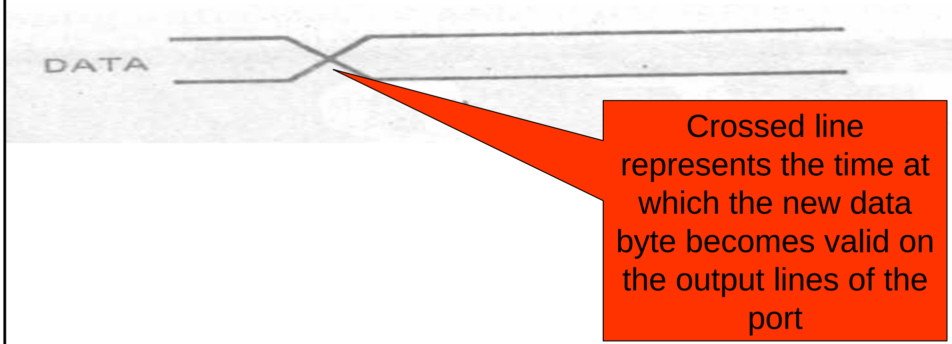
b. Rotational delay = [(60 seconds/5000 rpm) x (1000ms/second)]/2 = 6 ms + 5ms seek time = 11 ms.

Types of I/O

- The I/O system allow peripherals to provide data or receive results of processing the data
  – I/O ports are used
  – x86 can employ two different types of I/O
    - Isolated and memory mapped I/O
  – Some microprocessor system employ both types of I/O
- Information on I/O port is organized as bytes of data
  – I/O address space contains 64K consecutive bytes address in the range 0000h to FFFFh
    - Parts of address space 0000h to 00FFh referred as ‘page0’
    - Port 0 and port 1 may be considered as word wide port 0

Parallel Data Transfer - Methods

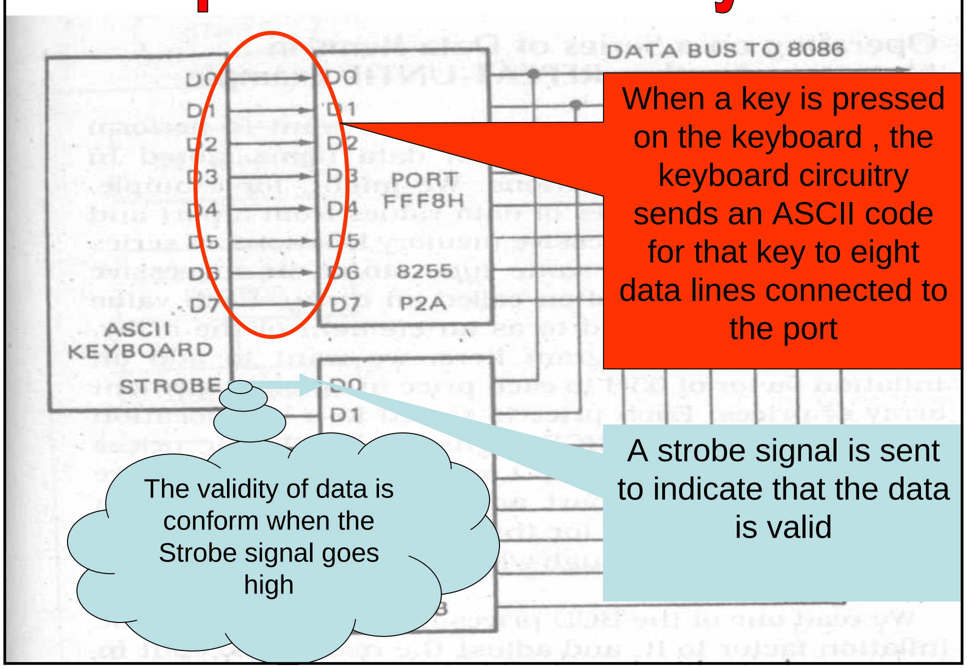
- Simple Input and Output
  – To get the digital data from a simple switch into microprocessor
    - Connect the switch to an input port line and read the port
  – To output data to a simple display device (LED)
    - Connect the LED on an output port pin and output the required logic level to turn on the light

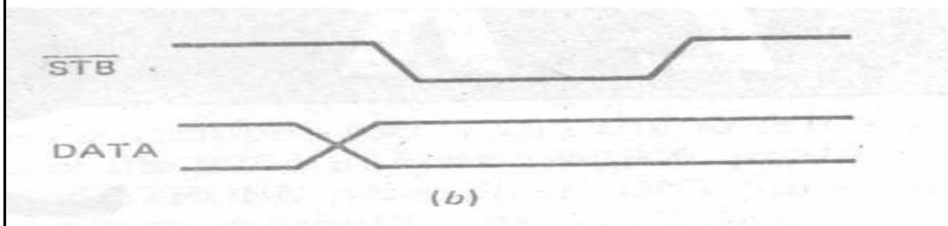


Simple Strobe I/O

- In applications when valid data is present on an external device only at a certain time
  - The data must be read in at that time
- System interface with the microcomputer output data on parallel signal lines and then output a separate signal to indicate the presence of valid data on the parallel line, that data ready signal is often called a "Strobe"

Example of Strobe data system



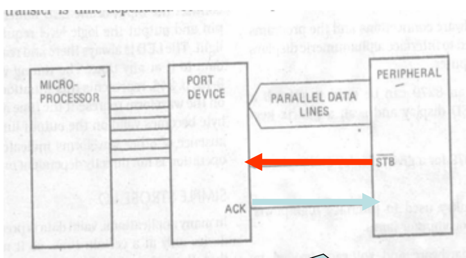


The sending device (keyboard) output parallel data on the data line and then sends out the $\overline{STB}$ signal to confirm the validity of the data

Handshake I/O Scheme

- A simple Strobe transfer works well for low rate of data transfer
- Not well for high data transfer
  - There is no signal which tells the sending device, when it is safe to send the next data byte
  - The sending device may send the data byte faster then the receiving system to read it
- To prevent such problems a handshake data transfer scheme is used
  - Single handshake I/O
  - Double handshake I/O

Single Handshake I/O

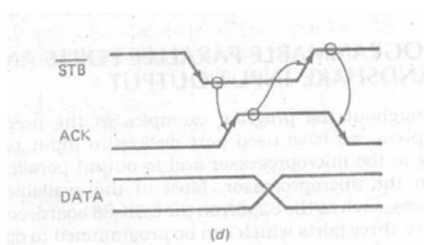


The peripheral sends an parallel data and sends an STB signal while microprocessor send the ACK signal to peripheral

The operation is refer to as a handshake or strobed I/O

Double Handshake I/O

•When more coordination is required between sending and receiving system



The sending device asserts STB signal low to check that if the receiving device is ready

The receiving device will rises its ACK signal high to accept data

The sending device asserts STB signal high to send data

The receiving device asserts its ACK sig low to accept data

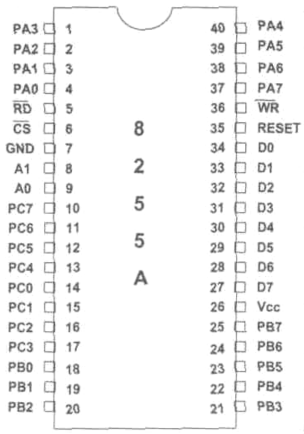
8255 Programmable Peripheral Interfacing

- one of the most widely used I/O chips.
  - It can be used for both memory-mapped and peripheral I/O
  - Provide a flexible parallel interface including, single bit, 4-bit and byte-wide input and output port
    - All these features configuration through software
- The 8255 is a 40-pin DIP chip
  - It has three separately accessible ports
    - Port A, Port B and Port C
    - The source and destination registers within 8255A is selected by a 2 bit register select code applied by MP on A0 and A1 lines of 8255A
      - Port A → A1A0 → 00
      - Port B → A1A0 → 01
      - Port C → A1A0 → 10

8255 Programmable Peripheral Interfacing

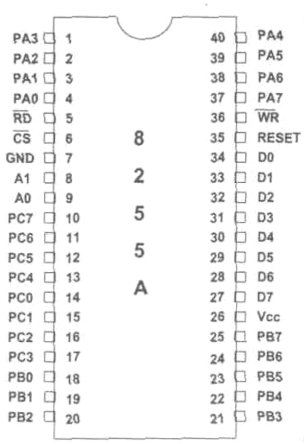
- More important,
  - one can program the individual ports of the 8255 to be input or output, and change them dynamically
    - Contains a 8 bit internal control register for programming the ports
  - In addition, the 8255 can be used to interface the CPU to devices using handshake signals

8255A, Pin layout



- **PA0 - PA7**
  - **This 8-bit port A can be programmed**
    - **all as input or all as output or all bits as bidirectional input/output.**
- **PB0-PB7**
  - **This 8-bit port B can be programmed**
    - **all as input or all as output. Port B cannot be used as a bidirectional port**
- **PC0-PC7**
  - **This 8-bit port C can be all input or all output.**
    - **It can also be split into two parts,**
      - CU (upper bits PC4 - PC7) and
      - CL (lower bits PCO - PC3)

8255A, Pin layout

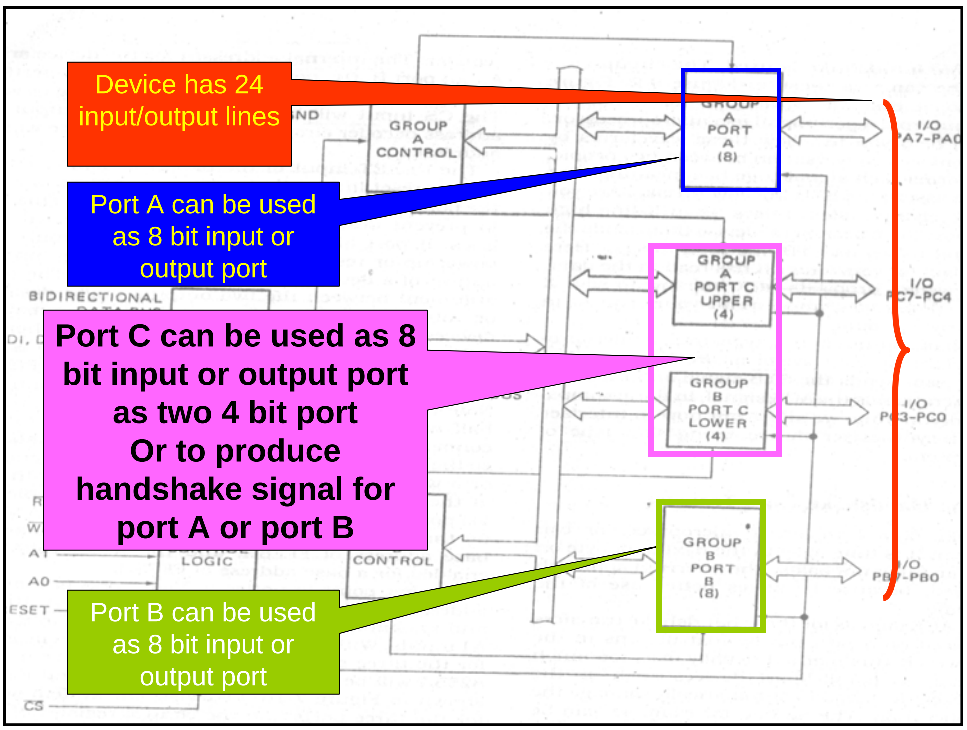


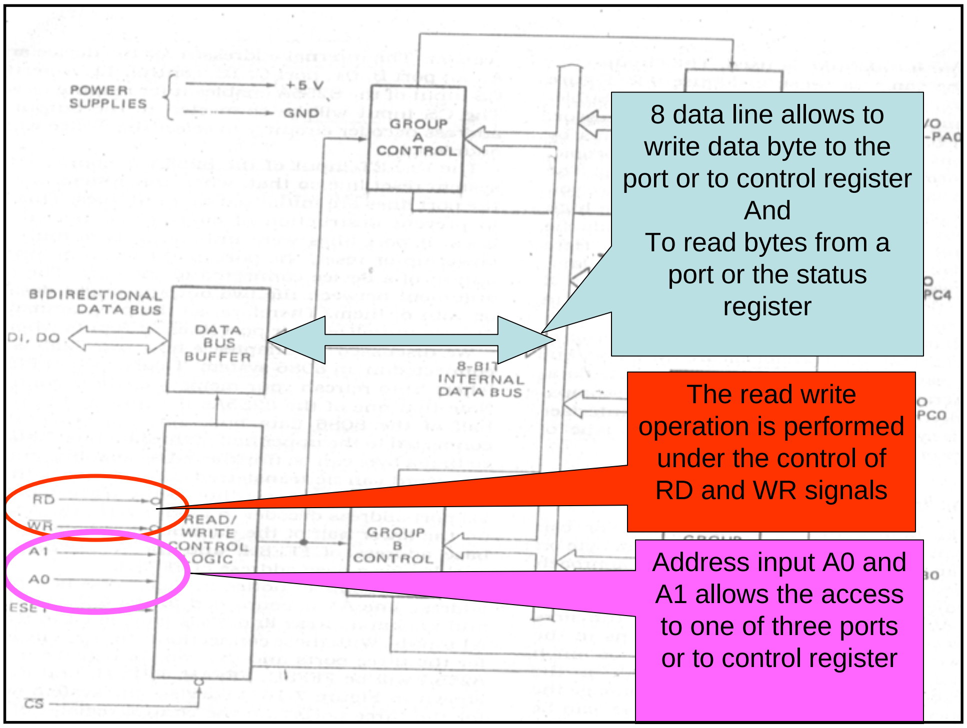
- **Each internal port can be used for input or output. In addition, any of PC0 to PC7 can be programmed**
  - **RESET and CS pins**
    - **CS' must be logic 0 during read and write operations**
    - **RESET is used to initialize the device**
- **RD and WR**
  - **These two active-low control signals are inputs to the 8255.**
  - **If the 8255 is using peripheral I/O design, IOR and IOW of the system bus are connected to these two pins. If the port uses memory-mapped I/O, MEMR and MEMW activate them**

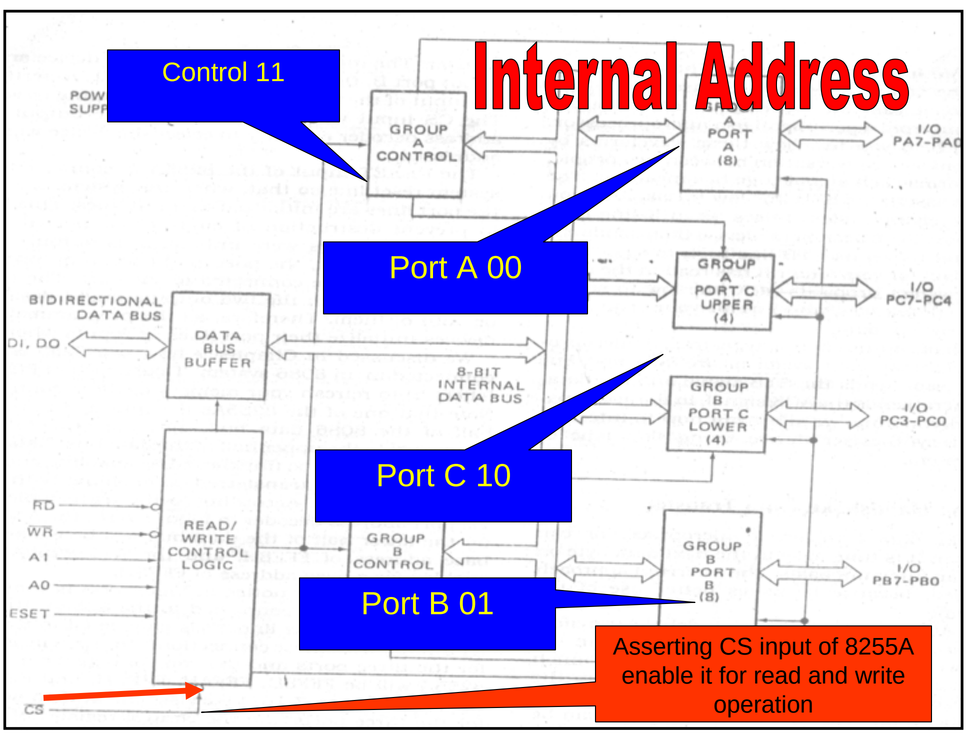
8255A pins

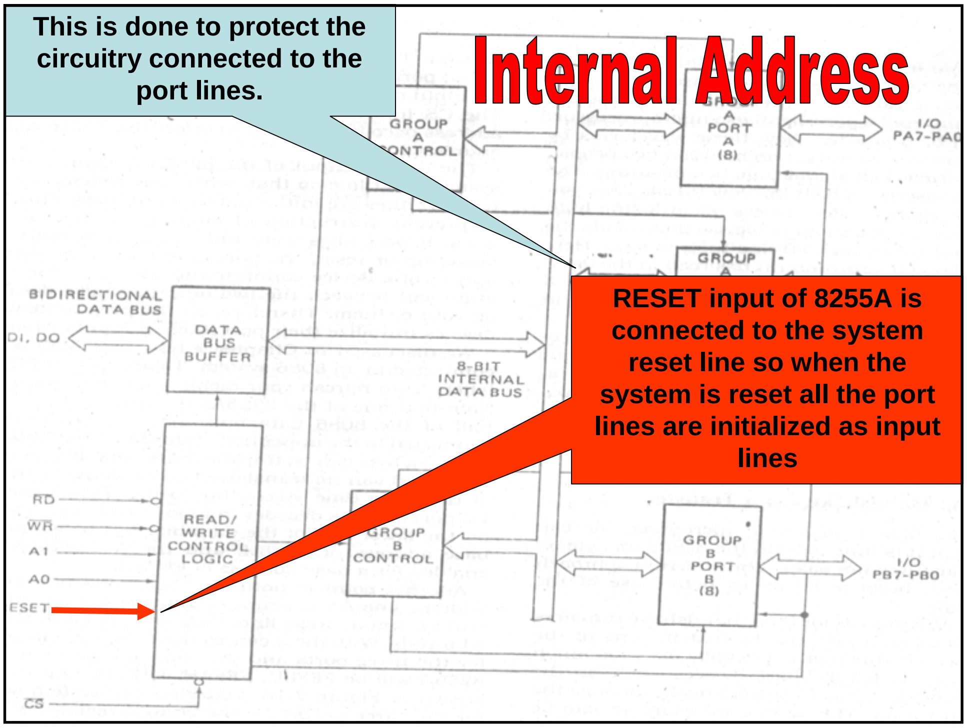
- **A0,A1,and CS'**
  - **While CS selects the entire chip, it is A0 and A 1 that select the specific port. These three pins are used to access ports A, B, C, or the control register according to the following Table**

CS'	A1	A0	Selects
0	0	0	Port A
0	0	1	Port B
0	1	0	Port C
0	1	I	Control register
1	X	X	8255 is not selected !




8 data line allows to write data byte to the port or to control register And To read bytes from a port or the status register
The read write operation is performed under the control of RD and WR signals
Address input A0 and A1 allows the access to one of three ports or to control register
POWER SUPPLIES
+5 V
GND
GROUP A CONTROL
BIDIRECTIONAL DATA BUS
DI, DO
DATA BUS BUFFER
8-BIT INTERNAL DATA BUS
RD
WR
A1
A0
READ/ WRITE CONTROL LOGIC
GROUP B CONTROL
CS


Internal Address
Control 11
Port A 00
Port C 10
Port B 01
Asserting CS input of 8255A enable it for read and write operation
GROUP A CONTROL
GROUP A PORT A (8)
I/O PA7-PA0
GROUP A PORT C UPPER (4)
I/O PC7-PC4
BIDIRECTIONAL DATA BUS
DI, DO
DATA BUS BUFFER
8-BIT INTERNAL DATA BUS
GROUP B PORT C LOWER (4)
I/O PC3-PC0
RD
WR
A1
A0
RESET
READ/ WRITE CONTROL LOGIC
GROUP B CONTROL
GROUP B PORT B (8)
I/O PB7-PB0
CS



Internal Address
This is done to protect the circuitry connected to the port lines.
RESET input of 8255A is connected to the system reset line so when the system is reset all the port lines are initialized as input lines
BIDIRECTIONAL DATA BUS
DI, DO
DATA BUS BUFFER
8-BIT INTERNAL DATA BUS
READ/ WRITE CONTROL LOGIC
RD
WR
A1
A0
ESET
CS
GROUP A CONTROL
GROUP A PORT A (8)
I/O PA7-PA0
GROUP A
GROUP B CONTROL
GROUP B PORT B (8)
I/O PB7-PB0