Input-Output and Communications

Instructor: Saraju P. Mohanty

- Computer I/O
- Sample Peripherals
- I/O Interfaces
- Serial Communication
- Modes of Transfer
- Priority Interrupts
- Direct Memory Access
- I/O Processors

Sources

- Logic and Computer Design Fundamentals by M. M. Mano and C. R. Kime.
- Dr. Valavanis lectures

Computer I/O

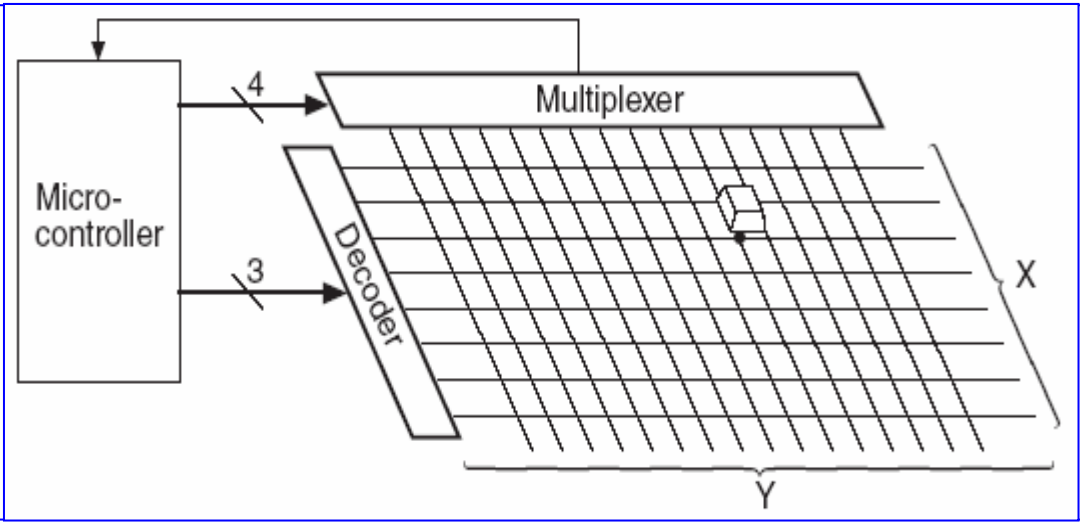
- Computer I/O provides an efficient mode of communication between the CPU and the outside environment.
- Common I/O devices:
  - Keyboards
  - Monitors
  - Printers
  - Magnetic disks
  - Compact disk read-only memory (CD-ROM) drivers
  - Modems and other communication interfaces
  - Magnetic tape drivers
  - Scanners
  - Sound cards with speakers and microphones

Sample Peripherals

- Three peripherals are considered in this chapter:
  - –Keyboard
  - –Hard disk
  - –Graphics display
- On-line devices – devices that the CPU controls directly are said to be connected *on-line*.
  - –On-line devices communicate directly with the CPU or transfer binary information into or out of the memory upon command from the CPU.
- Peripherals – Input or output devices attached to the computer on-line are called peripherals.

Sample Peripherals: Keyboard

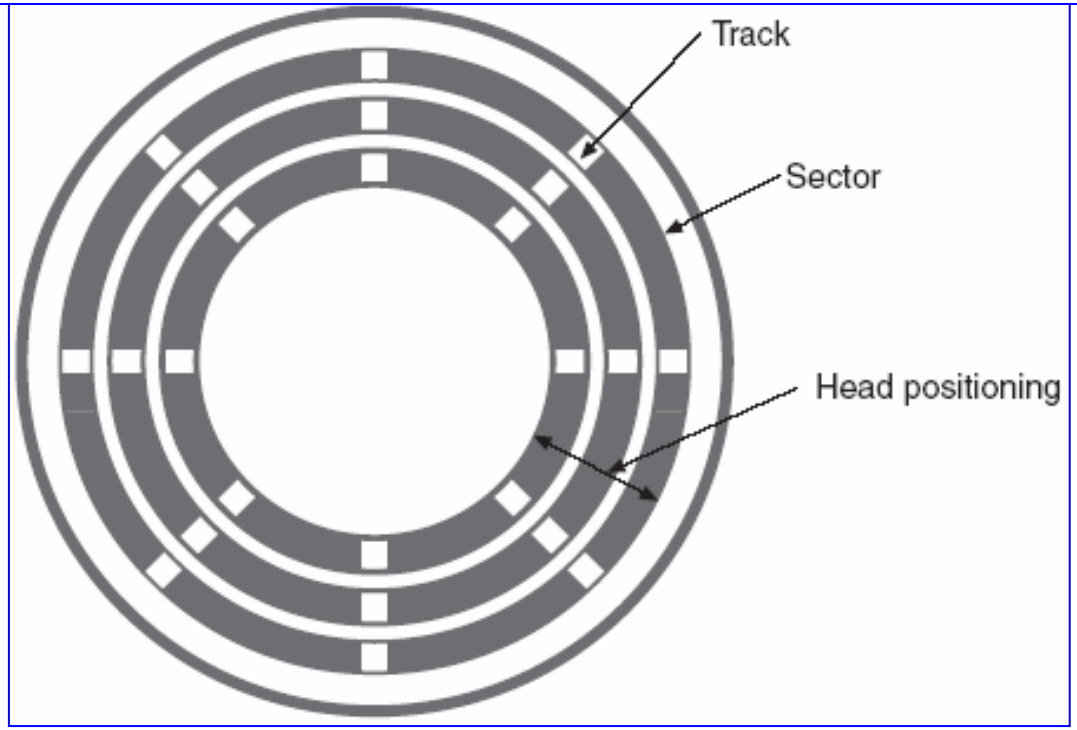
- Keyboard – one of the simplest electromagnetic devices attached to a computer.
- Has on of the slowest data rates of any peripheral, as it is manually controlled.
- The keyboard consists of a collection of keys that can be depressed by the user.



- The scan-matrix that lies beneath the keys is used to detect which of the keys have been depressed.
- The matrix is 8 X 16, giving 128 intersections; can handle up to 128 keys (2^7)
- Decoder drives X lines of matrix, and multiplexer is attached to the Y lines.
- Decoder and multiplexer are controlled by the microcontroller, a tiny computer that contains RAM, ROM, a timer and simple I/O interfaces.

Sample Peripherals: Hard Disk

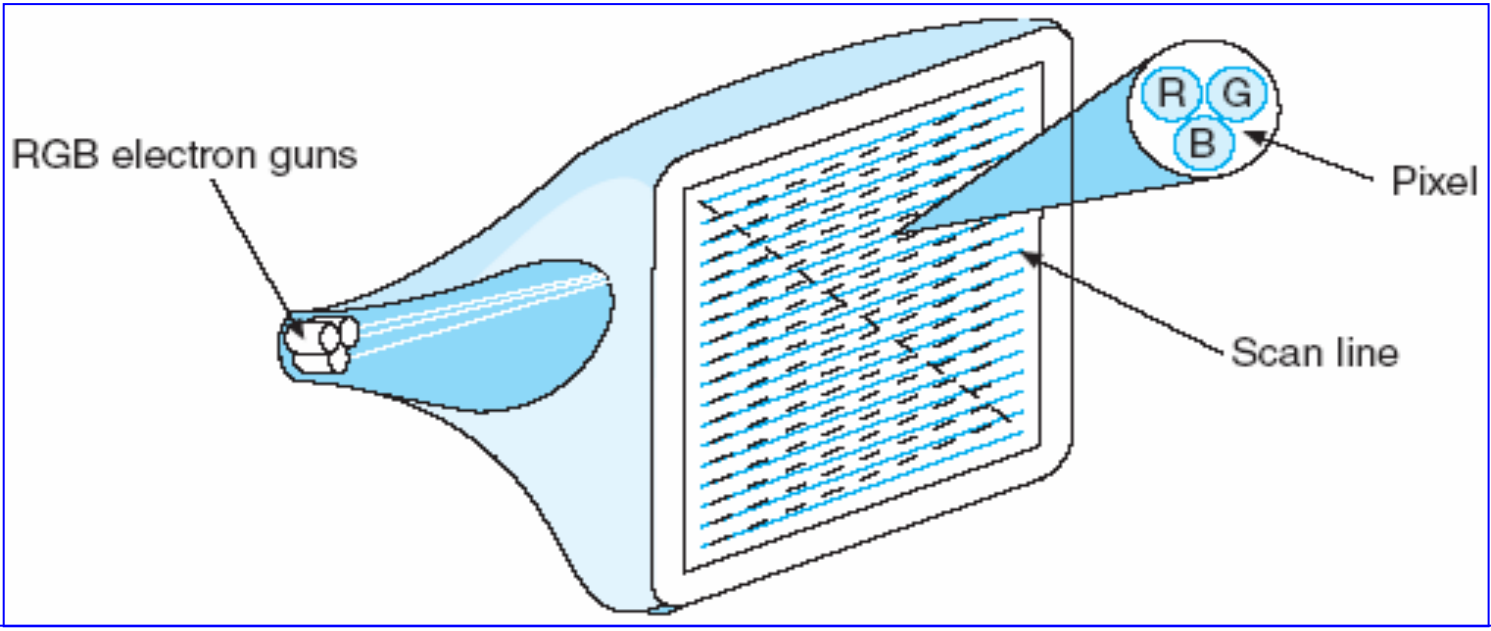
- Hard disk is a intermediate-speed, nonvolatile, writable storage medium.
- Typical hard drive stores information on a non-removable disk with a few many platters. Each platter is magnetizable on one or both surfaces.
- Tracks: Each platter is divided into concentric tracks
- Sectors: Each track is divided into containing a fixed number of bytes
- Cylinder: Set of tracks that are at same distance from the center of disk



Sample Peripherals: Hard Disk

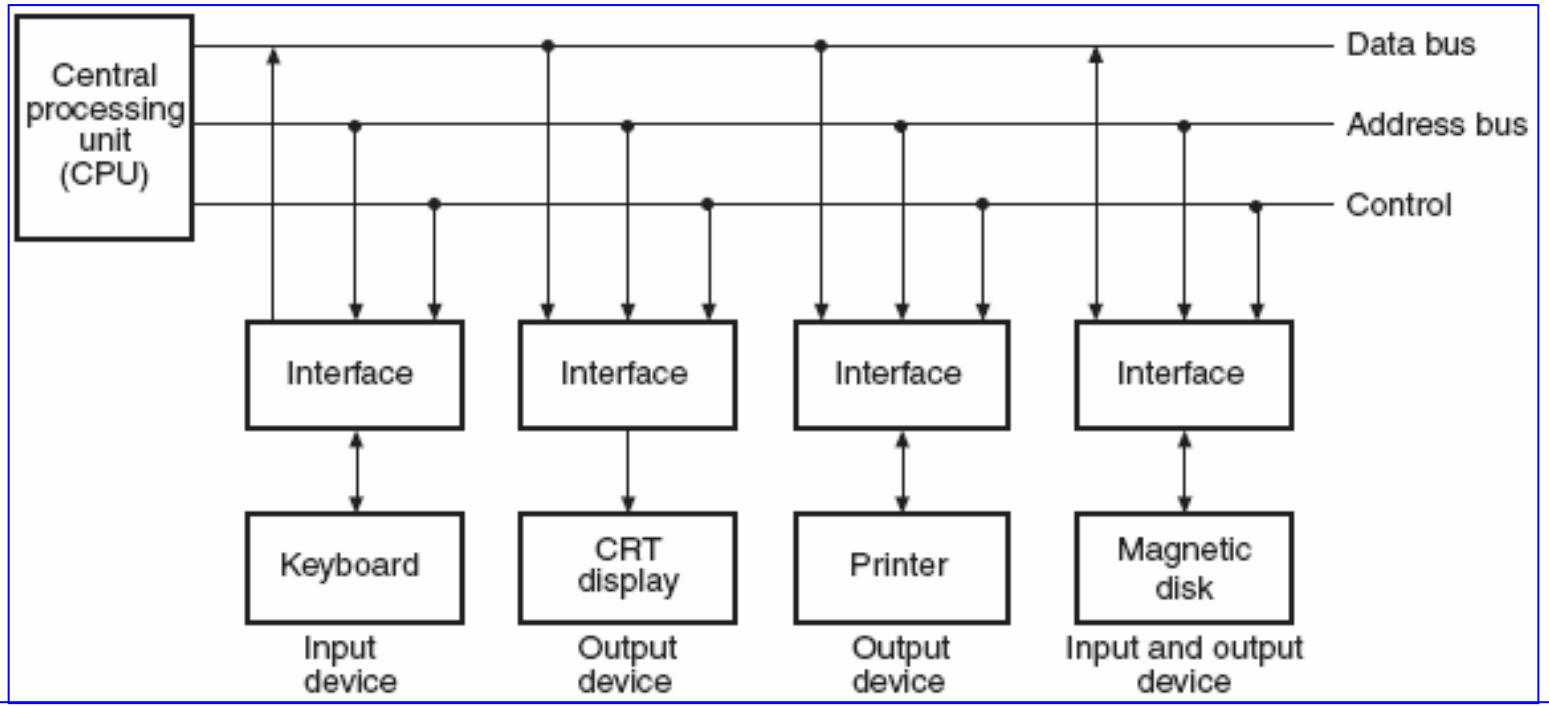
- Byte address = (cylinder number, head number, sector number, word offset within the sector).
- Seek Time: the time required to move the heads from the current cylinder to the desired cylinder.
- Rotational Delay: the time required to rotate the disk from its current position to that having the desired sector under the heads.
- Controller Time: the time required by the disk controller to access and output information.
- Disk Access Time: the time required to locate a word on the disk **= controller time + seek time + rotational delay.**
- Disk Transfer Rate (MB/s) – the number of bytes in the sector divided by the length of time taken to read a sector from the disk.
- The length of time required to read a sector is equal to the proportion of the cylinder occupied by the sector divided by the rotational speed of the disk.

Sample Peripherals: Graphics Display



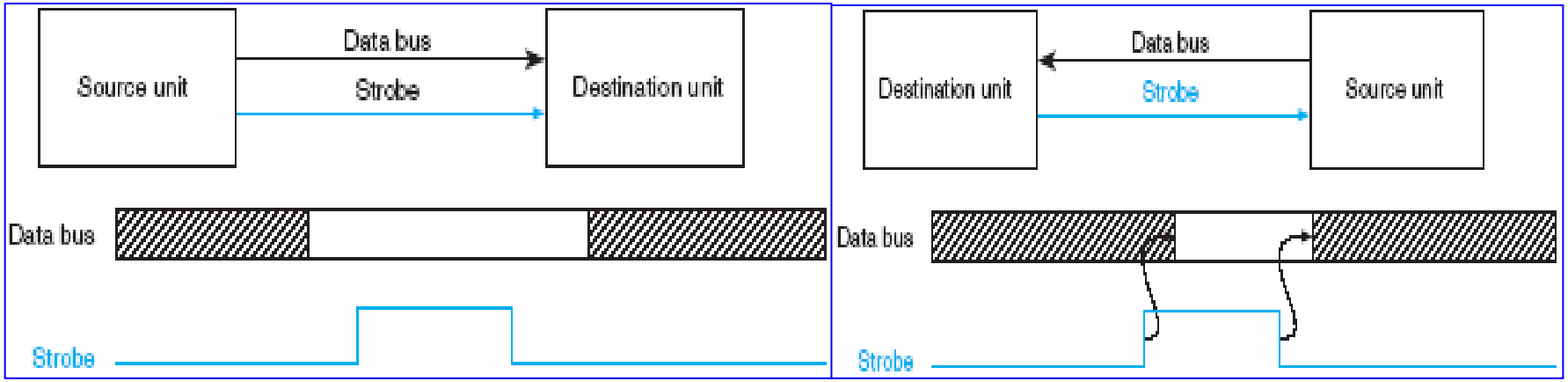
- Primary output device for the interactive use of computers.
- Display is defined in terms of picture elements called pixels.
- Color display has 3 locations (corresponding to primary colors red, green and blue) associated with each pixel on screen.
- Phosphor at each location is exited by the RGB electron guns. The color that results for a given pixel is predetermined by the intensity of the electron beam striking the phosphorous within the pixel.

I/O Interface: I/O Bus and Interface Unit



- Each peripheral has an interface unit associated with it.
- To communicate with a particular device, the CPU places its address on the address bus, and a function code on the control bus.
- Each interface monitors the address – the interface with specified address activates the path between the bus lines and its device, others are disabled.
- The selected interface responds to the function code and proceeds to execute it.

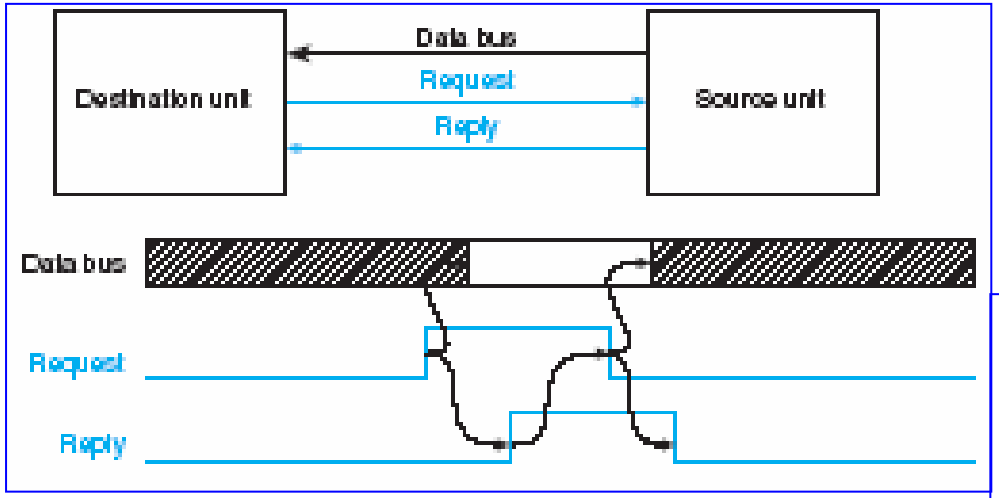
I/O Interface: Strobing



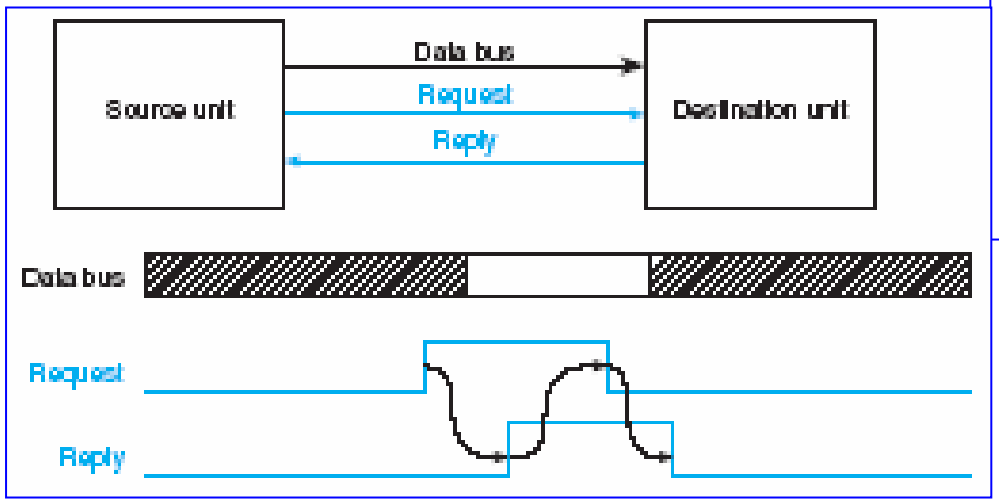
Destination-initiated transfer Source-initiated transfer

- CPU, interface, and I/O device have different clocks and are asynchronous with respect to each other. Strobing or handshaking mechanism used is such cases.
- The data bus is made bidirectional by the use of three-state buffers.
- Destination-initiated transfer – destination changes strobe from 0 to 1. source reads strobe to be 1 and places data on data bus. Destination unit captures data in a register and changes Strobe to 0.
- Source-initiated transfer – source places data on data bus, changes strobe from 0 to 1 after data settles, destination reads strobe and sets up transfer to its register, source then changes strobe to 0 which triggers the transfer.

I/O Interface: Handshaking



Destination-initiated transfer



Source-initiated transfer

Uses 2 control signals to deal with the timing of transfers: one control line (Request) is used to request a response from other unit, second control line (Reply) is used to reply to the initiating unit that the response is occurring.

Serial Communication: Asynchronous

- Data transfer may be either parallel or serial.
- In serial transfer one bit is transmitted at a time and require one or two lines, whereas parallel transmission require more lines and is faster.
- Modems (modulator-demodulators) are useful for signal conversion for communication between computer and other remote devices.
- Serial transmission may be synchronous or asynchronous.

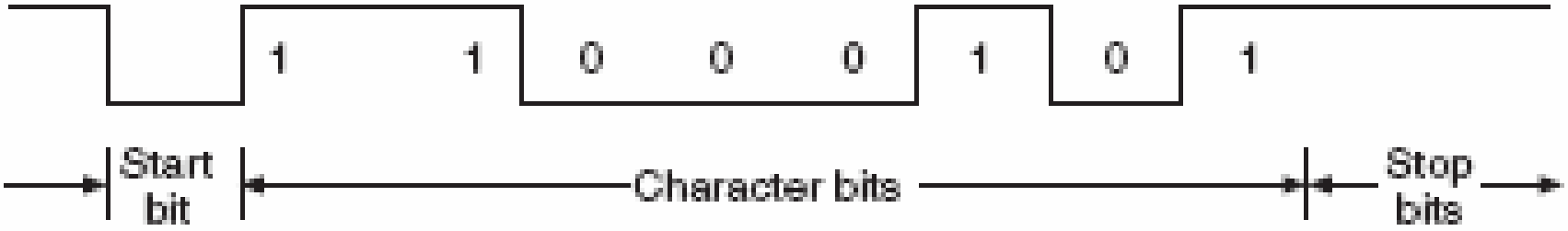


Fig. 11-8 Format of Asynchronous Serial Transfer of Data

Each character consists of three parts:

- Start bit: is always a 0 and is used to indicate the beginning of a character.
- Character bits: 8 bits following the start bit that carry the actual character.
- Stop bits: series of 1s after the character bits, to specify the end of character.

A transmission character is detected by applying transmission rules.

Serial Communication: Synchronous

- Synchronous transmission does not use start or stop bits to frame characters.
- Modems employed in synchronous transmission have internal clocks that are set to the frequency at which bits are being transmitted.
- For proper operation, it is required that the clocks of the transmitter and receiver modems remain synchronized at all times.
- Contrary to asynchronous transmission, a continuous message is sent not character by character.
- The message consists of a group of bits that form a block data. The entire block transmitted with special control bits a the beginning and end.

Serial Communication: A Packet-Based Serial I/O Bus

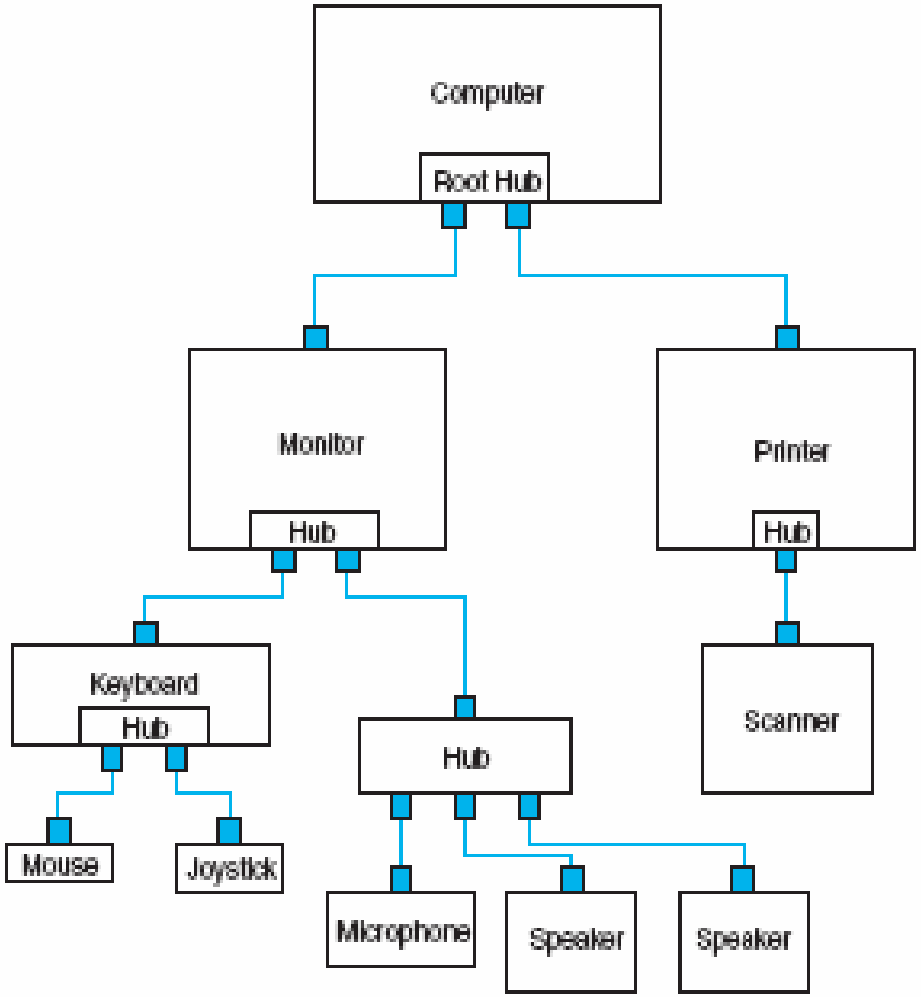


Fig. 11-10 I/O Device Connection Using the Universal Serial Bus (USB)

- The computers and attached devices can be classified as hubs, devices, or compound devices.
- A hub provides attachment points for USB devices and other hubs.
- A hub contains a USB interface for control and status handling and a repeater for transferring information through the hub.
- High speed USB products have a design data rate of 480 Mb/s. Full speed USB devices signal at 12Mb/s, while low speed devices use a 1.5Mb/s subchannel. Serial port: 115kbits/s (.115Mbits/s)

Modes of Transfer: Program-Controlled

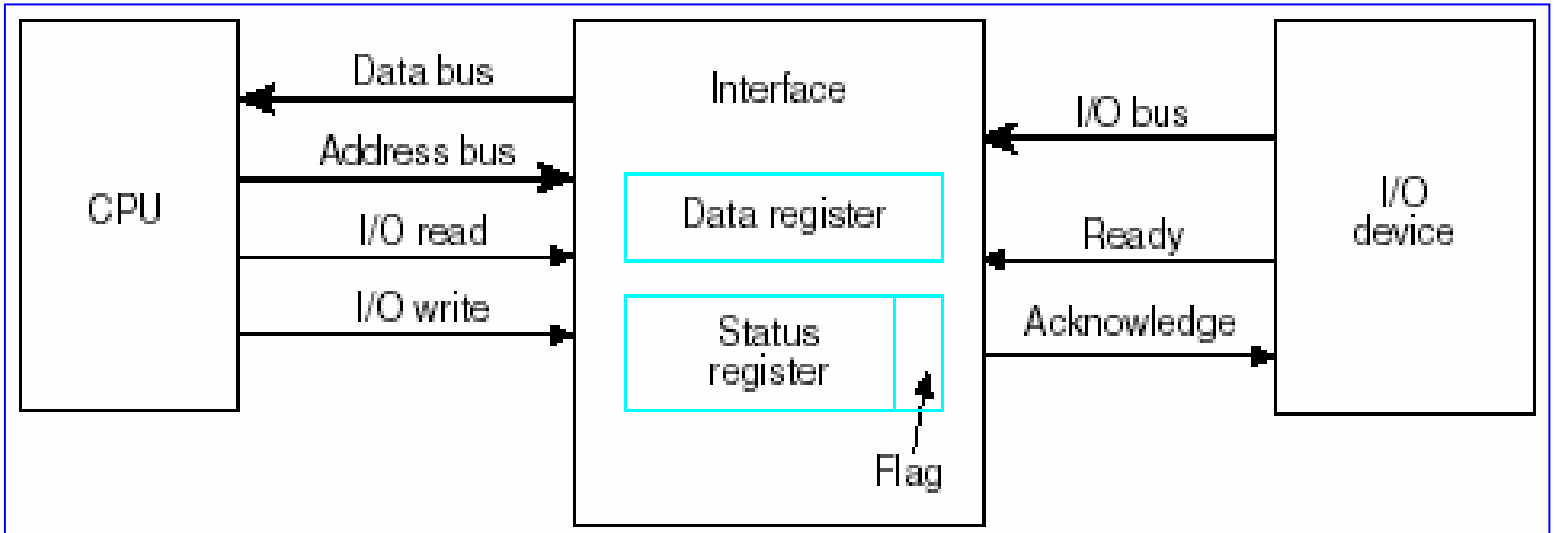


Fig. 11-13 Data Transfer from I/O Device to CPU

- When a byte is available, the device places it on the I/O bus and enables Ready.
- The interface accepts the byte into its data register and enables Acknowledge.
- The interface sets a bit (*flag*) in the status register.
- The device can now disable Ready, but, it will not transfer another byte until Acknowledge is disabled by the interface.

Priority Interrupt: Daisy Chain Priority

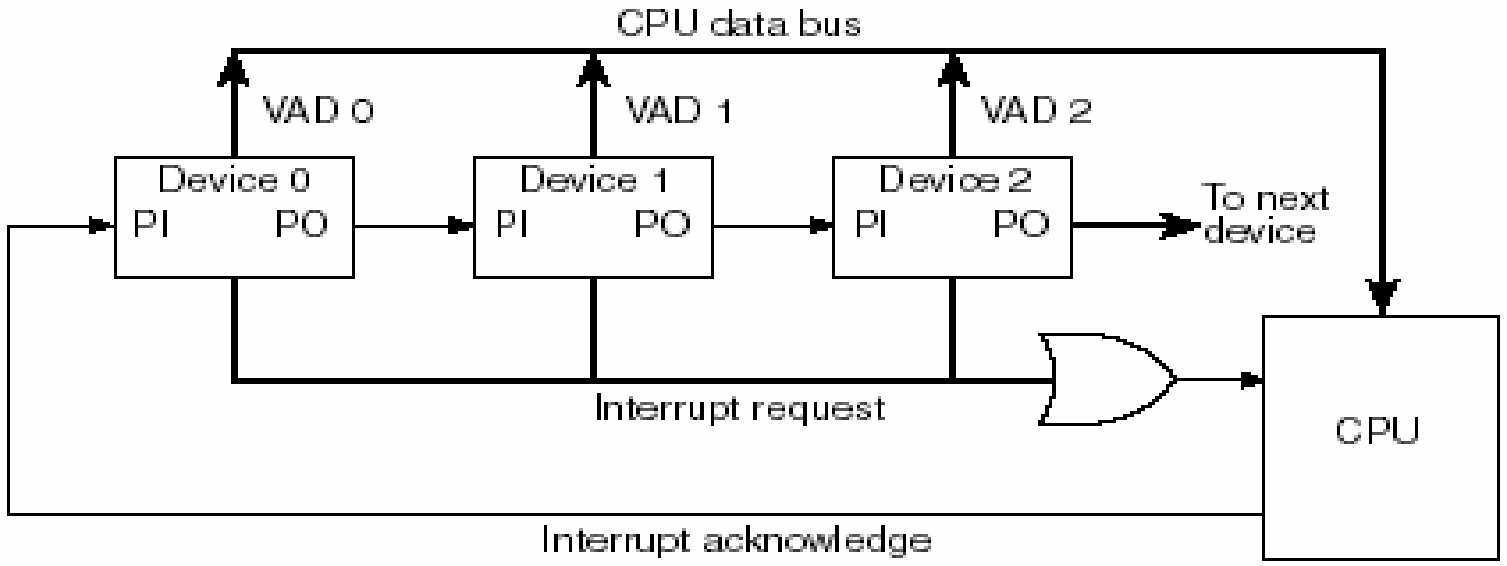
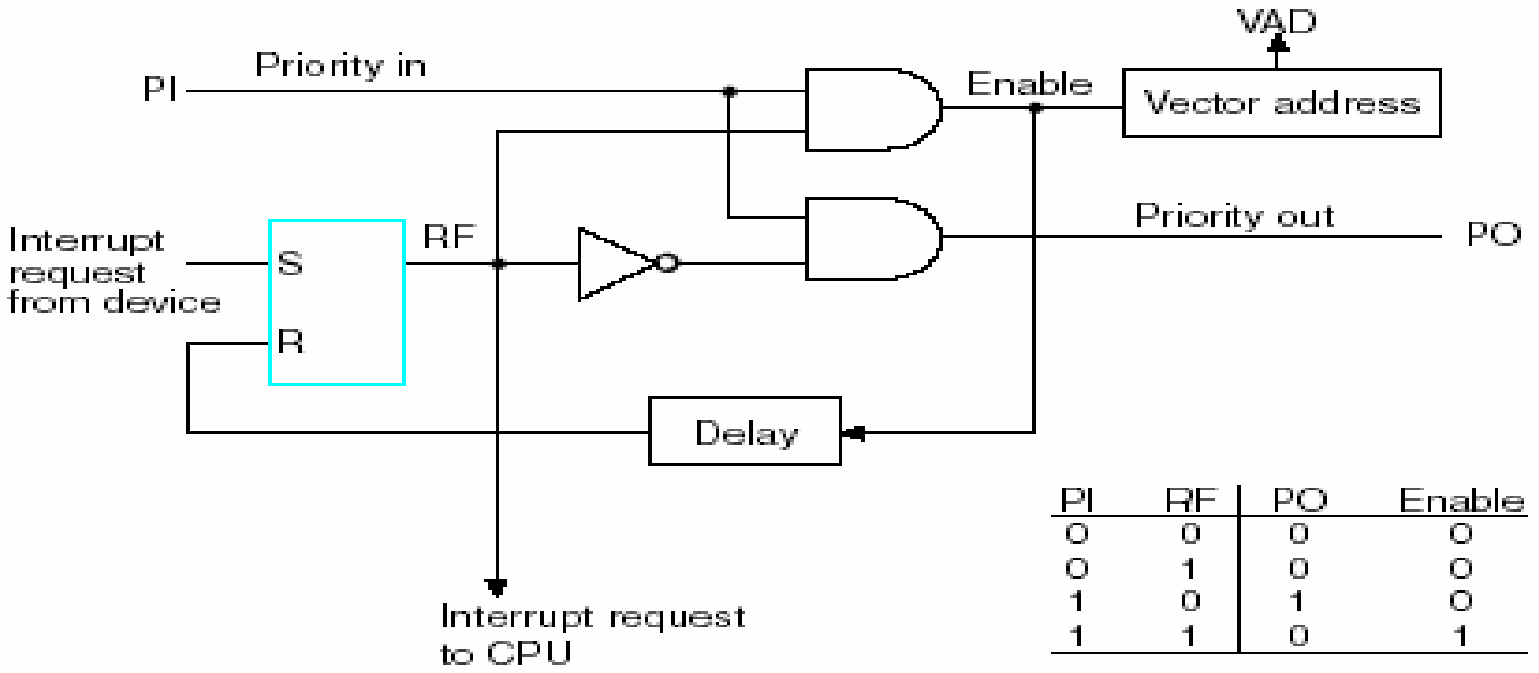


Fig. 11-15 Daisy Chain Priority Interrupt

- *Daisy chain* consists of a serial connection of all devices that request an interrupt.
- Devices are placed in the chain in the descending order of their priority.
- Interrupt request lines from all the devices are ORed to form the interrupt line to the CPU.
- CPU responds by enabling Interrupt Acknowledge.
- The signal is received by device 0 at PI input and then passes on to the next device through PO output only if device 0 is not requesting an interrupt.

Priority Interrupt: Daisy Chain Priority



PI	RF	PO	Enable
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1

Fig. 11-16 One Stage of the Daisy Chain Priority Arrangement

- The device sets its RF latch when its about to interrupt the CPU.
- The output of the latch functionally enters the OR that drives interrupt line.
- If PI = 0, both PO and enable line to VAD are equal to 0, irrespective of the value of RF.
- If PI = 1 and RF = 0, then PO = 1, the vector address is disabled and acknowledge signal passes to the next device through PO.

Priority Interrupt: Parallel Priority Hardware

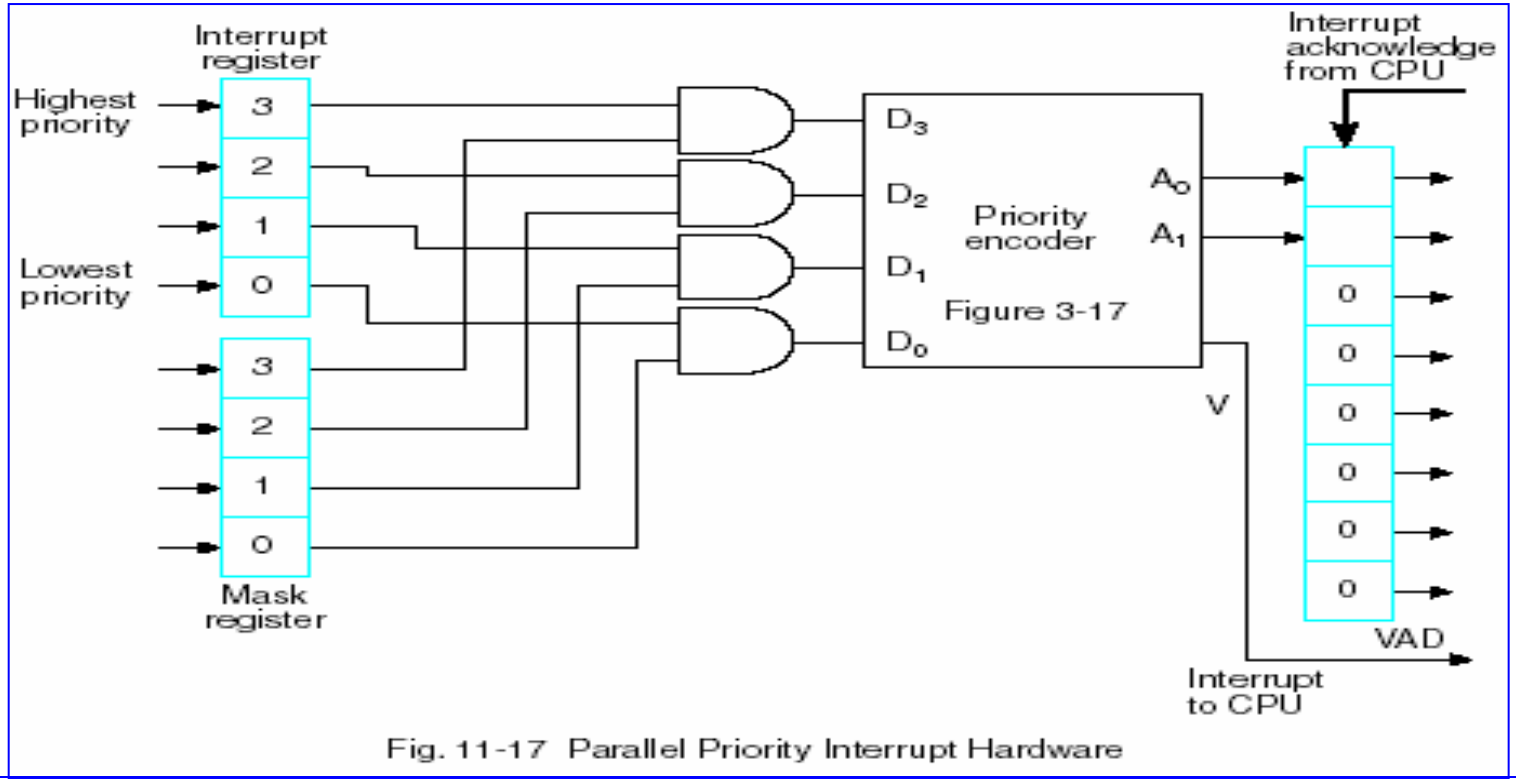


Fig. 11-17 Parallel Priority Interrupt Hardware

- Uses a register with bits set separately by the interrupt signal from each device. Priority is established according to the position of bits in register.
- The individual bits of interrupt register are set by external conditions and cleared by program instructions (input 3 has highest priority and 0 lowest).
- The mask register has the same number of bits as interrupt register.

Direct Memory Access

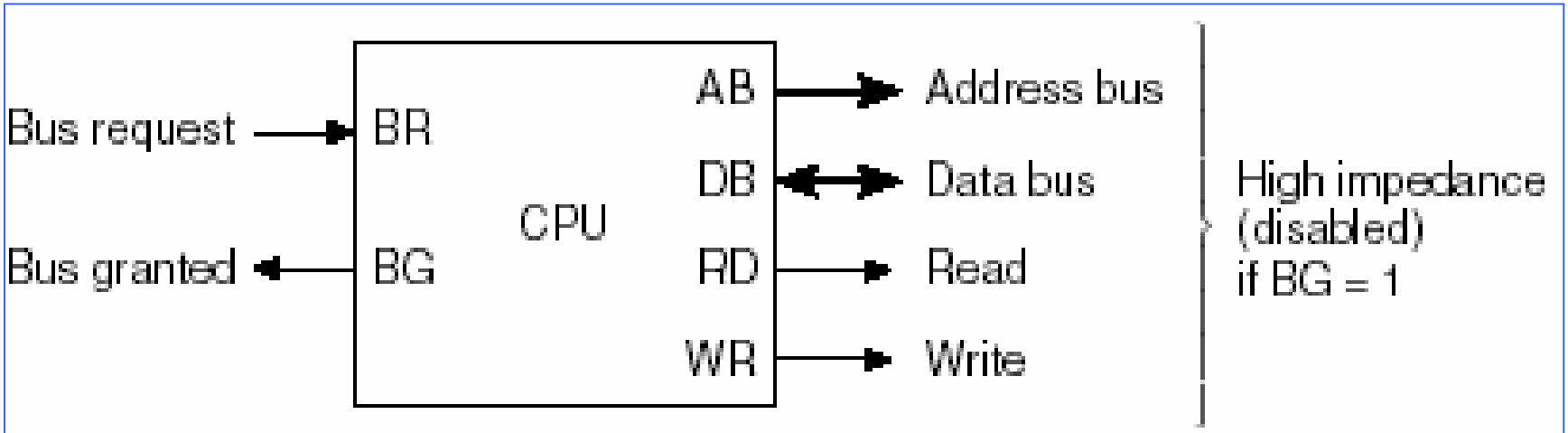


Fig. 11-18 CPU Bus Control Signals

- DMA – A transfer technique in which data is transferred directly between the I/O device and memory.
- The Bus Request (BR) input is used by DMA controller to request the CPU to relinquish control of the buses.
- When BR is active, CPU places the address bus, data bus and the read & write lines into a high-impedance state.
- The CPU then activates the Bus Granted (BG) output to inform the external DMA that it can take control of the buses.

Direct Memory Access: DMA Controller

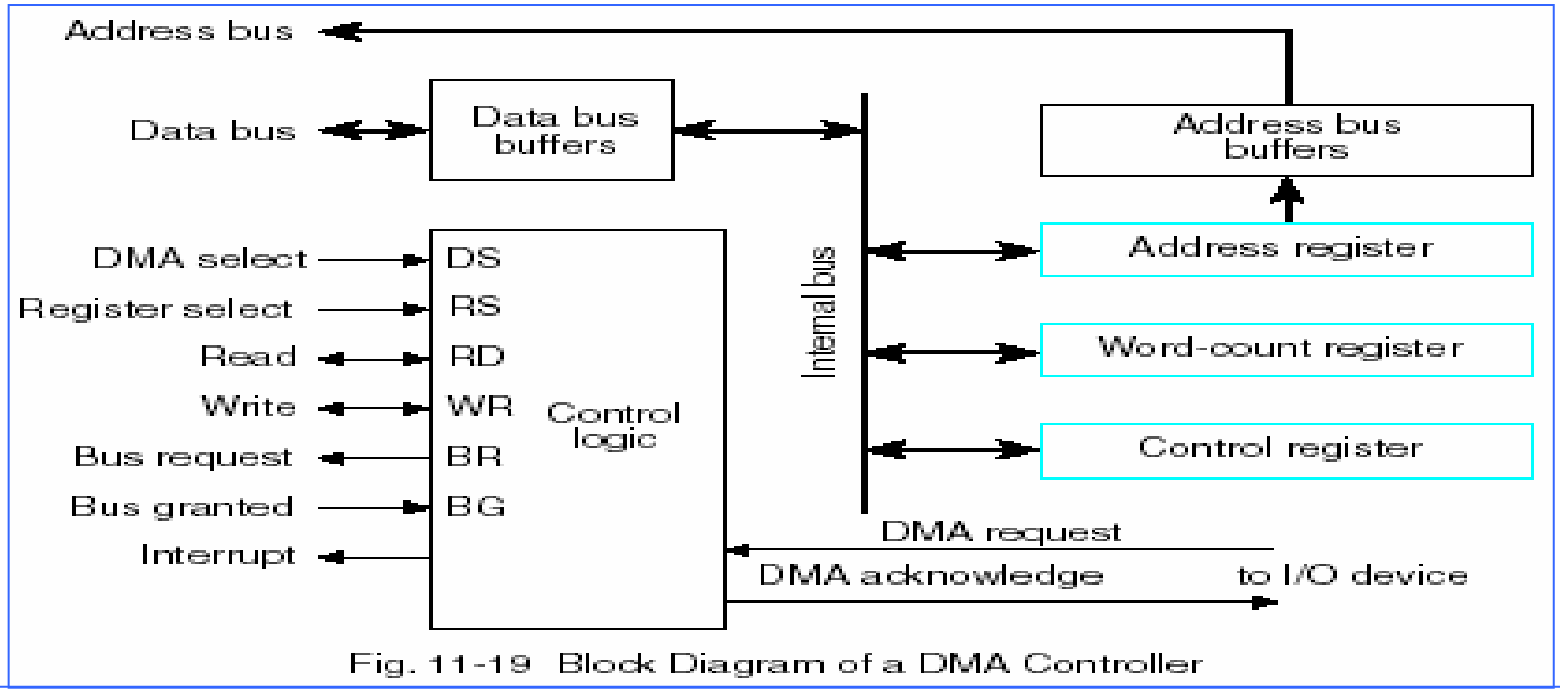


Fig. 11-19 Block Diagram of a DMA Controller

- Registers in the DMA are selected by the CPU through address bus by enabling the DS and RS inputs.
- RD and WR inputs are bidirectional.
- When BG = 0, CPU can read from or write to the DMA registers.
- When BG = 1, CPU has relinquished the buses, and the DMA can communicate directly with memory by specifying an address on address bus and activating the RD or WR control.

Direct Memory Access: DMA Transfer

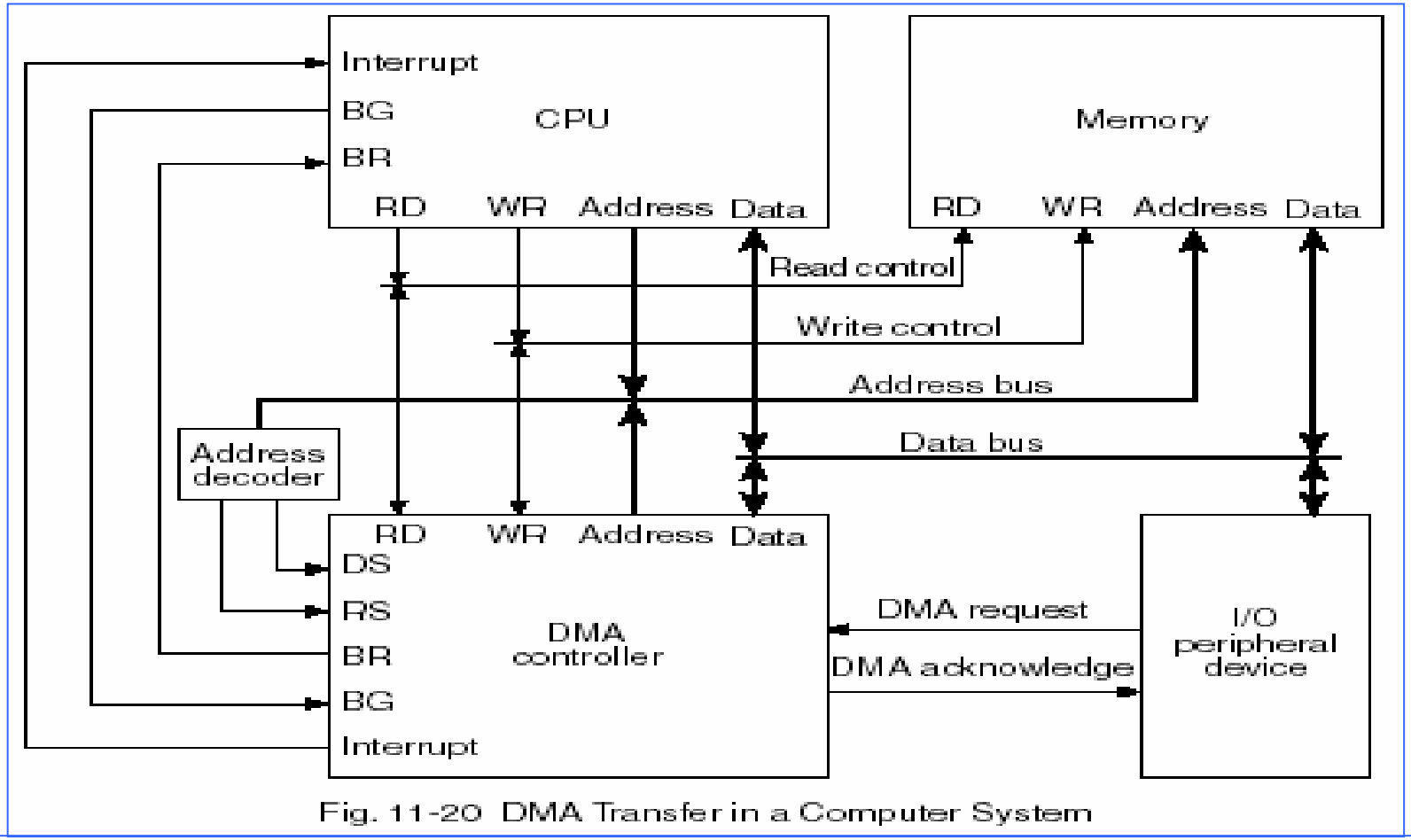


Fig. 11-20 DMA Transfer in a Computer System

- CPU communicates with DMA through address and data buses.
- DMA has its own addresses, which activates the DS and RS lines.

I/O Processors

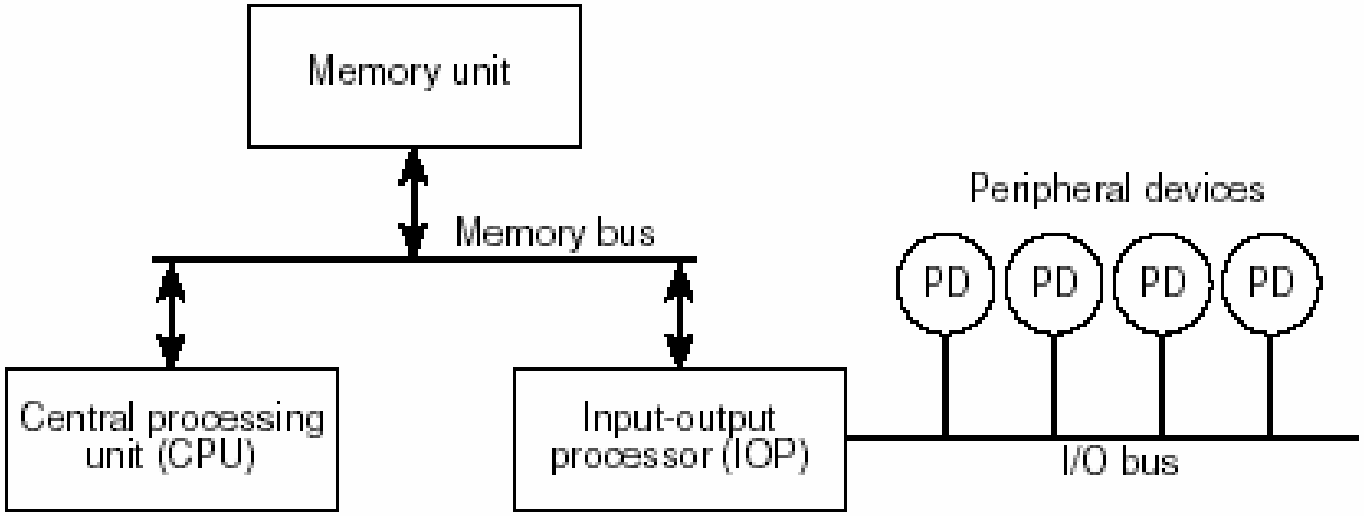


Fig. 11-21 Block Diagram of a Computer with I/O Processor

- I/O Processor (IOP) is a processor with direct memory access capability that communicates with I/O devices.
- The memory occupies a central position and can communicate with each processor by means of DMA.
- The CPU is responsible for processing the data needed in the solution of computational tasks.
- The IOP provides a path for the transfer of data between various peripheral devices and the memory.