Input and output organization

Chapter 7

Peripheral Devices

- A peripheral device is any device attached to a computer in order to expand its functionality.
- Basically input and output devices together are known as peripherals.
- Input devices are keyboard, optical input devices like bar code reader, screen input devices like touch screen and light pen.
- Output devices may be monitor, printer , speakers, etc

Input Output Interface

- Input output interface provides a method for transferring information between internal storage and external I/O devices. Peripherals connected to a computer need communication links for interfacing them with CPU. The purpose of communication link is to resolve the differences that exist between the computer and each peripheral. The major differences are:

1. Peripherals are electromechanical and electromagnetic devices and their manner of operation is different from the operation of CPU and memory, which are electronic devices.

2. Data transfer rate of peripherals is slower than that of CPU so some synchronization mechanism may be needed

IO interface

3. Data codes and formats in peripherals differ from that of the word format in CPU and memory.

4. Operating modes of peripherals are different from each other and each must be controlled so as not to disturb others.

To resolve these differences, computer system usually include special hardware unit between CPU and peripherals to supervise and synchronize I/O transfers, which are called **interface units** since they interface processor bus and peripherals.

A typical communication link between the processor and several peripherals is shown in the figure

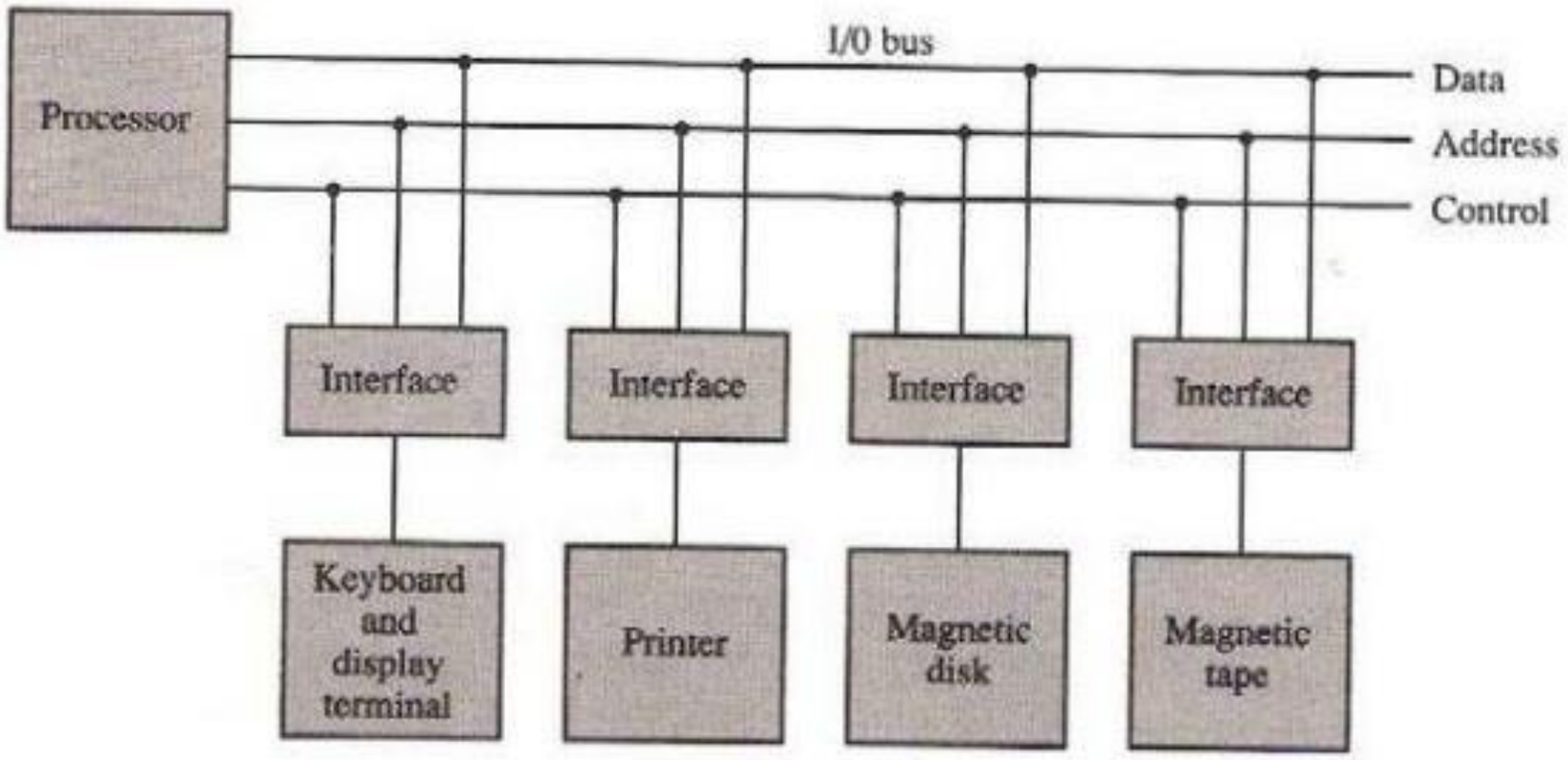


Fig: Connection of I/O bus to I/O devices

IO interface

- The I/O bus consists of data lines, address lines and control lines. Different peripherals are connected to it. Each peripheral has an interface module associated with it.
  Functions of interface are:
  - Decodes the device address.
  - Decodes the I/O command in control lines.
  - Provides signal for the peripheral controller.
  - Synchronizes the data flow
  - Supervises the transfer rate between peripheral and CPU or memory.

Modes of Data IO data transfer

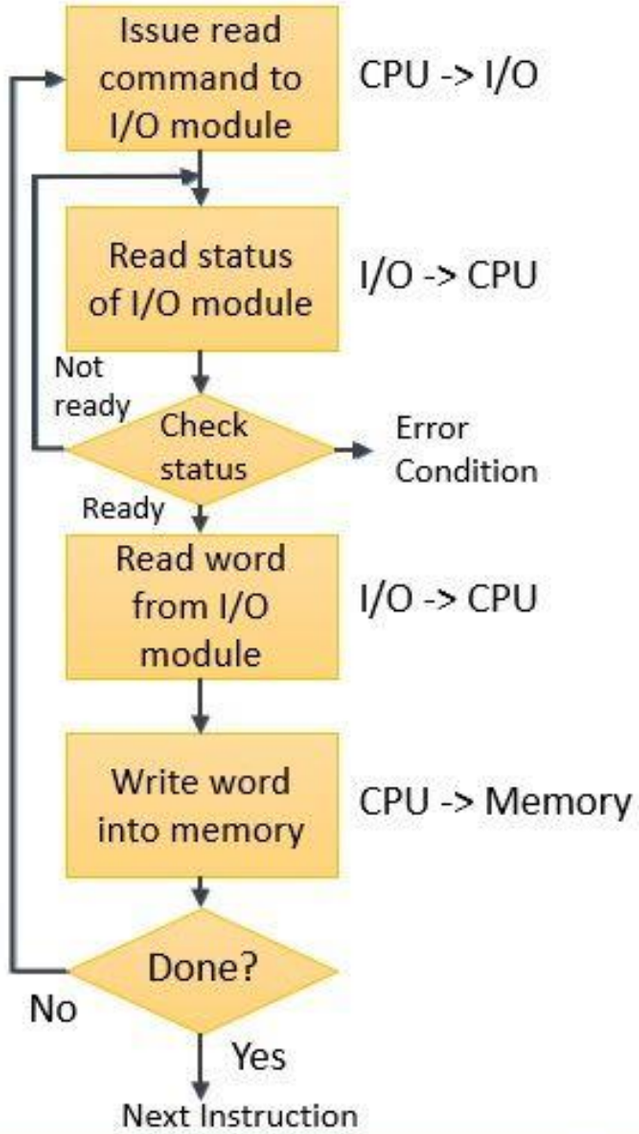
- Programmed IO
- Interrupt
- DMA

Programmed I/0

In this mode the data transfer is initiated by the instructions written in a computer program. An input instruction is required to store the data from the device to the CPU and a store instruction is required to transfer the data from the CPU to the device. Data transfer through this mode requires constant monitoring of the peripheral device by the CPU and also monitor the possibility of new transfer once the transfer has been initiated.

Thus CPU stays in a loop until the I/O device indicates that it is ready for data transfer. Thus programmed I/O is a time consuming process that keeps the processor busy needlessly and leads to wastage of the CPU cycles.

This can be overcome by the use of an interrupt facility. This forms the basis for the Interrupt Initiated I/O.



Programmed I/O to transfer data from I/O module to memory

Polling vs Interrupt

- Polling means the CPU keeps checking a flag to indicate if something happens.
- An interrupt driven device driver is one where the hardware device being controlled will cause a hardware interrupt to occur whenever it needs to be serviced.
- With interrupt, CPU is free to do other things, and when something happens, an interrupt is generated to notify the CPU. So it means the CPU does not need to check the flag.
- Polling is like picking up your phone every few seconds to see if you have a call. Interrupts are like waiting for the phone to ring.
- Search for the advantages of Interrupt over polling???
- What is Interrupt service routine?

Interrupt

- Interrupt is signals send by an external device to the processor, to request the processor to perform a particular task or work.
- Mainly in the microprocessor based system the interrupts are used for data transfer between the peripheral and the microprocessor.
- The processor will check the interrupts always at the 2nd T-state of last machine cycle.
- If there is any interrupt it accept the interrupt and send the INTA (active low) signal to the peripheral.
- The vectored address of particular interrupt is stored in program counter.
- The processor executes an interrupt service routine (ISR) addressed in program counter.
- It returned to main program by RET instruction

Interrupt structures:

- A processor is usually provided with one or more interrupt pins on the chip.
- Therefore a special mechanism is necessary to handle interrupts from several devices that share one of these interrupt lines.
- There are mainly two ways of servicing multiple interrupts which are

1. Polled interrupts and
2. Daisy chain (vectored) interrupts.

1. Polled interrupts

- In this method, all interrupts are serviced by branching to the same service program.
- This program then checks with each device if it is the one generating the interrupt.
- The order of checking is determined by the priority that has to be set. The device having the highest priority is checked first and then devices are checked in descending order of priority.
- If the device is checked to be generating the interrupt, another service program is called which works specifically for that particular device
- Here several devices are connected to a single interrupt line (INTR) of the microprocessor.
- The major disadvantage of this method is that it is quite slow. To overcome this, we can use hardware solution, one of which involves connecting the devices in series. This is called Daisy-chaining method.

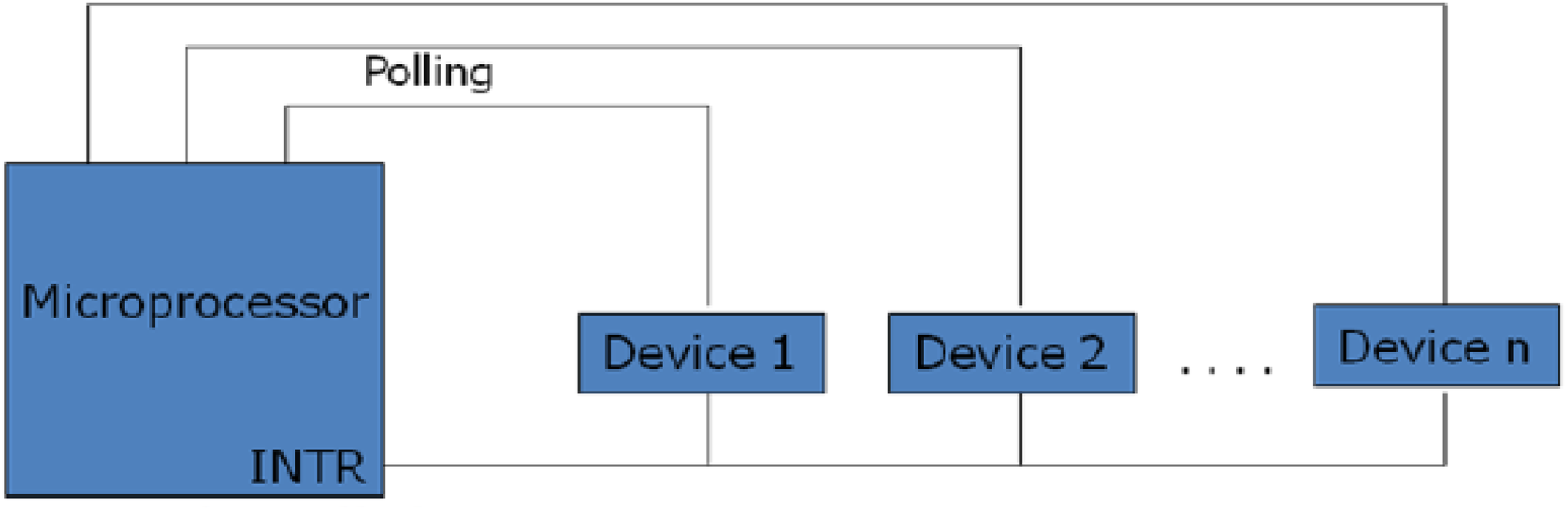


Fig: Polled Interrupt

2. Daisy chain (vectored) interrupt

- In polled interrupt, the time required to poll each device may exceed the time to service the device through software.
- To improve this, the faster mechanism called vectored or daisy chain interrupt is used. Here the devices are connected in chain fashion.
- When INTR pin goes up, the processor saves its current status and then generates INTA signal to the highest priority device. If this device has generated the interrupt, it will accept the INTA; otherwise it will push to the INTA next priority device until it is accepted by the interrupting device.
- When INTA is accepted, the device provides a means to the processor for findings the interrupt address vector using external hardware.
- The accepted device responds by placing a word on the data lines which becomes the vector address with the help of any hardware through which the processor points to appropriate device service routine

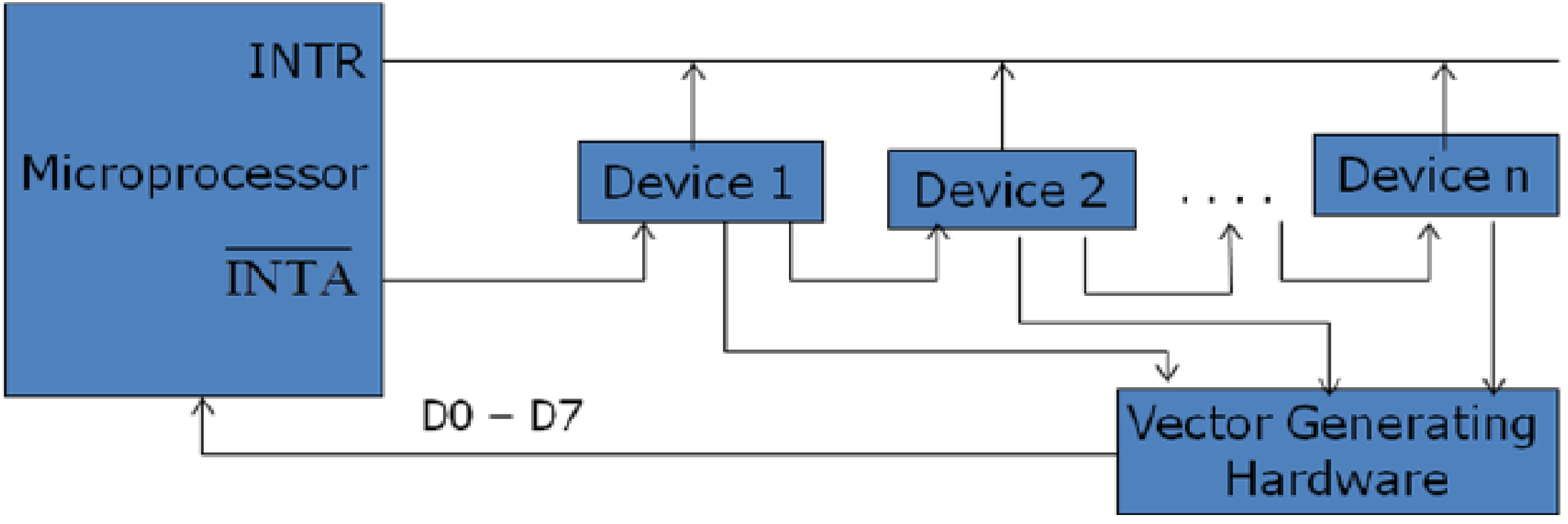
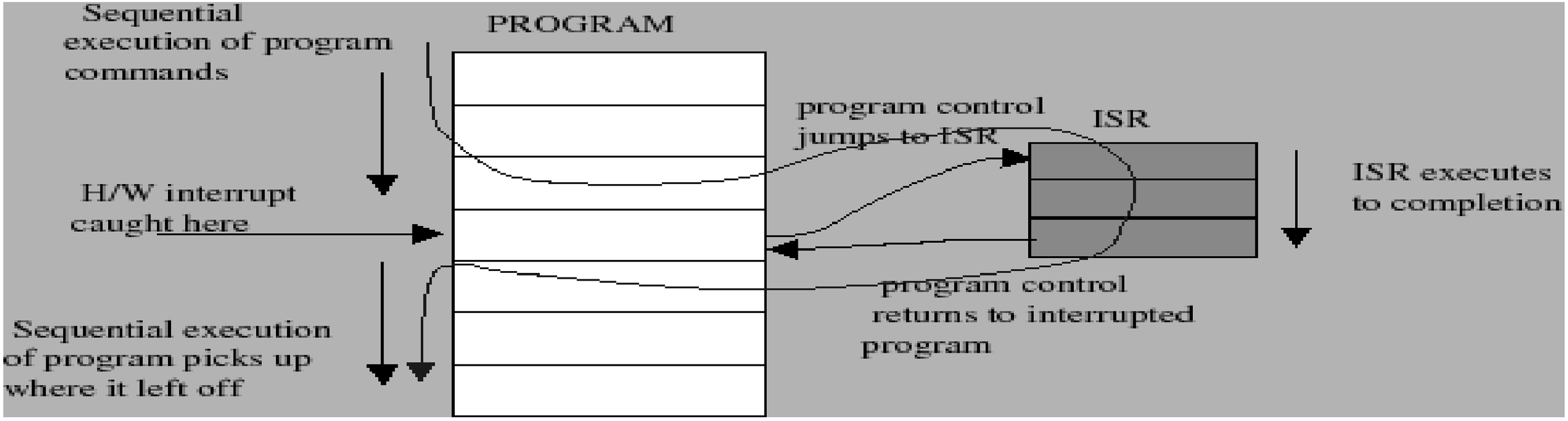


Fig: Vectored (Daisy Chain) Interrupt

Interrupt Processing Sequence

- The occurrence of interrupt triggers a number of events, both in processor hardware and in software. The interrupt driven I/O operation takes the following steps.
- The I/O unit issues an interrupt signal to the processor for exchange of data between them.
- The processor finishes execution of the current instruction before responding to the interrupt.
- The processor sends an acknowledgement signal to the device that it issued the interrupt.
- The processor transfers its control to the requested routine called “Interrupt Service Routine (ISR)” by saving the contents of program status word (PSW) and program counter (PC).
- The processor now loads the PC with the location of interrupt service routine and the fetches the instructions. The result is transferred to the interrupt handler program.
- When interrupt processing is completed, the saved register’s value are retrieved from the stack and restored to the register.
- Finally it restores the PSW and PC values from the stack.

Control flow in the presence of a hardware interrupt



Interrupt Service Routine

- An interrupt service routine (ISR) is a software routine that hardware invokes in response to an interrupt.
- ISRs examine an interrupt and determine how to handle it.
- ISRs handle the interrupt, and then return a logical interrupt value.
- Its central purpose is to process the interrupt and then return control to the main program.
- An ISR must perform very fast to avoid slowing down the operation of the device and the operation of all lower priority ISRs.
- As in procedures, the last instruction in an ISR should be iret

ISR is responsible for doing the following things:

1. Saving the processor context

Because the ISR and main program use the same processor registers, it is the responsibility of the ISR to save the processor's registers before beginning any processing of the interrupt. The processor context consists of the instruction pointer, registers, and any flags. Some processors perform this step automatically.

2. Acknowledging the interrupt

The ISR must clear the existing interrupt, which is done either in the peripheral that generated the interrupt, in the interrupt controller, or both.

3. Restoring the processor context

After interrupt processing, in order to resume the main program, the values that were saved prior to the ISR execution must be restored. Some processors perform this step automatically

DMA -Introduction

- DMA is acronym to Direct Memory Access.
- Direct Memory Access (DMA) is a method of allowing data to be moved from one location to another in a computer without intervention from the central processor (CPU).
- It is also a fast way of transferring data within (and sometimes between) computer.
- The DMA I/O technique provides direct access to the memory while the microprocessor is temporarily disabled.
- The DMA controller temporarily borrows the address bus, data bus and control bus from the microprocessor and transfers the data directly from the external devices to a series of memory locations (and vice versa).

Cont…

- I/O devices are connected to system bus via a special interference circuit known as DMA CONTROLLER.
- Both CPU and DMA have access to main memory via a shared system bus having data address and control line
- The controller manages data transfer between memory and peripheral directly by borrowing the address bus, control bus and data bus from microprocessor and bypassing it.

DMA operation

- The direct memory access (DMA) technique provides direct access to the memory while the microprocessor is temporarily disabled.
- A DMA controller temporarily borrows the address bus, data bus, and control bus from the microprocessor and transfers the data bytes directly between an I/O port and a series of memory locations.
- The DMA transfer is also used to do high-speed memory-to memory transfers.
- Two control signals are used to request and acknowledge a DMA transfer in the microprocessor-based system.
- The HOLD signal is a bus request signal which asks the microprocessor to release control of the buses after the current bus cycle.
- The HLDA signal is a bus grant signal which indicates that the microprocessor has indeed released control of its buses by placing the buses at their high-impedance states. The HOLD input has a higher priority than the INTR or NMI interrupt inputs.

Cont..

- HOLD input has higher priority than INTR or NMI.
- The only microprocessor pin that has a higher priority than a HOLD is the RESET pin.
- HLDA becomes active to indicate that the processor has placed its buses at high impedance state.
- A **DMA read** transfers data from the memory to the I/O device.
- A **DMA write** transfers data from an I/O device to memory.
- The system contains separate memory and I/O control signals.

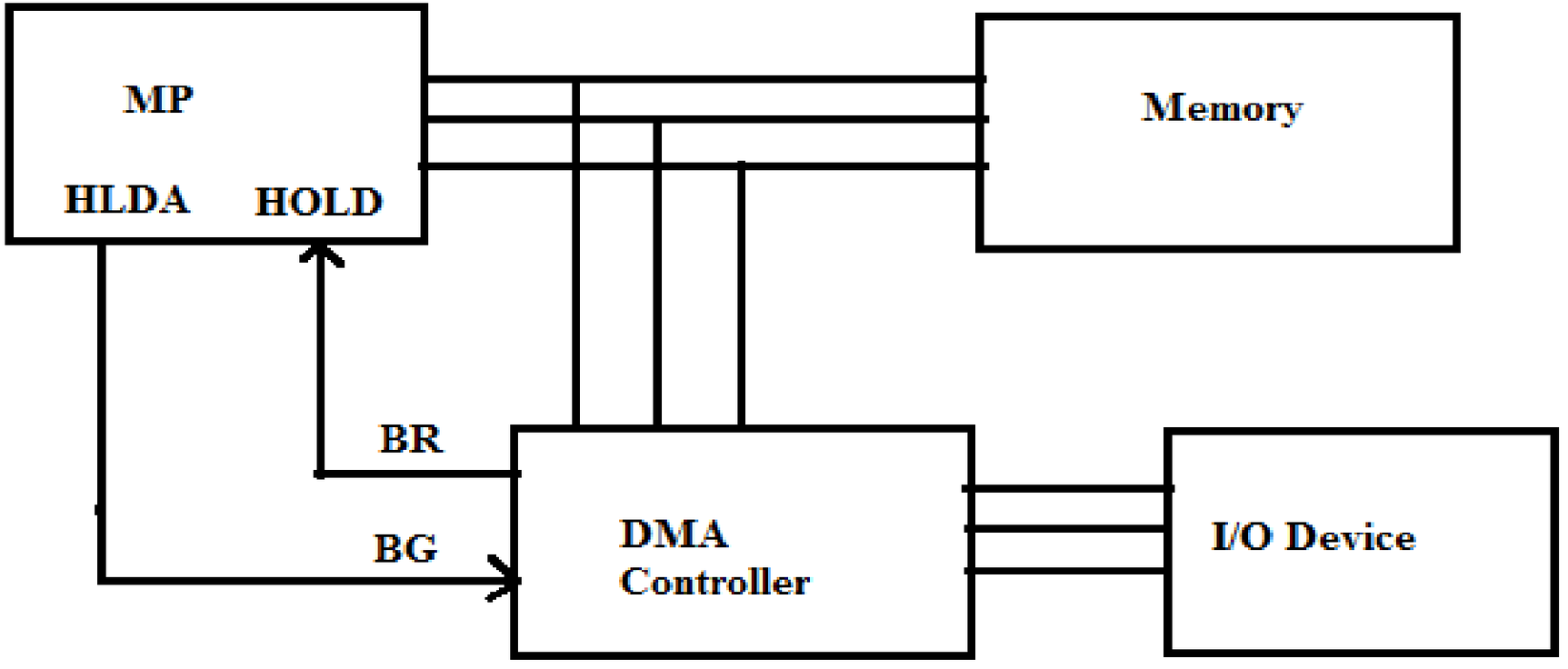
- Taking control of the bus for a bus cycle is called cycle stealing. Just like the bus control logic, a master must be capable of placing addresses on the address bus and directing the bus activity during a bus cycle.
- The components capable of becoming masters are processors (and their bus control logic) and DMA controllers.
- Sometimes a DMA controller is associated with a single interface, but they are often designed to accommodate more than one interface.

DMA Data Transfer scheme

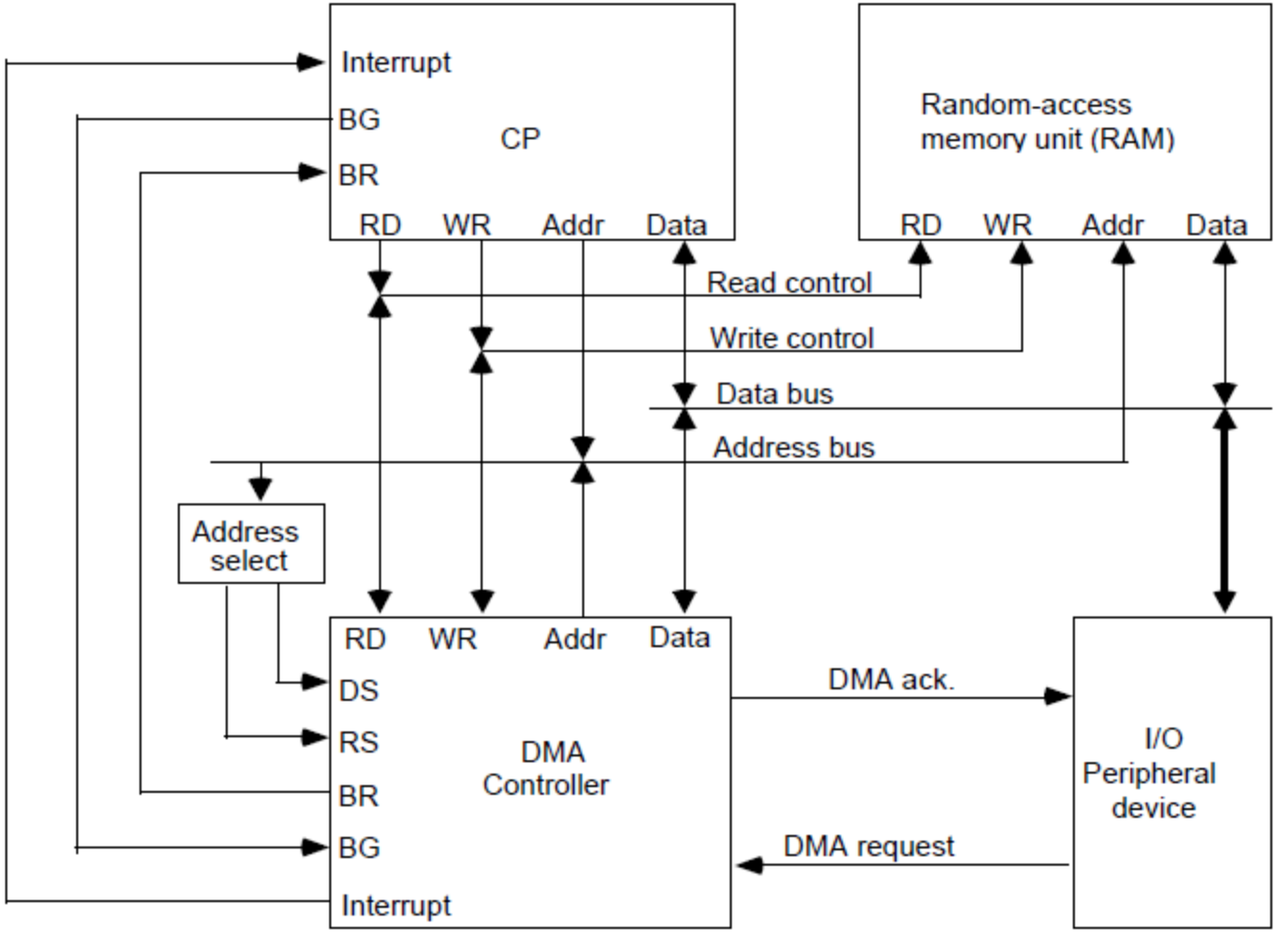
- Data transfer from I/O device to memory or vice-versa is controlled by a DMA controller.
- This scheme is employed when large amount of data is to be transferred.
- The DMA requests the control of buses through the HOLD signal and the MPU acknowledges the request through HLDA signal and releases the control of buses to DMA.
- It's a faster scheme and hence used for high speed printers

Block (Burst) mode of data transfer
In this scheme the I/O device withdraws the DMA request only after all the data bytes have been transferred.
Cycle stealing technique
In this scheme the bytes are divided into several parts and after transferring every part the control of buses is given back to MPU and later stolen back when MPU does not need it

DMA controller



DMA Data transfer



IO processor

- I/O processor is a processor with DMA capability that communicates with I/O devices. In this configuration, the computer system can be divided into a memory unit, and a number of processors comprised of CPU and one or more IOPs.
- IOP is similar to CPU expect that it is designed to handle the details of I/O processing. Unlike DMA controller which is set up by the CPU, IOP can fetch and execute its own instructions. IOP instructions are designed specifically to facilitate I/O transfers. In addition, IOP can perform other processing tasks such as arithmetic, logic, branching and code translation.

Block diagram of IO procesor

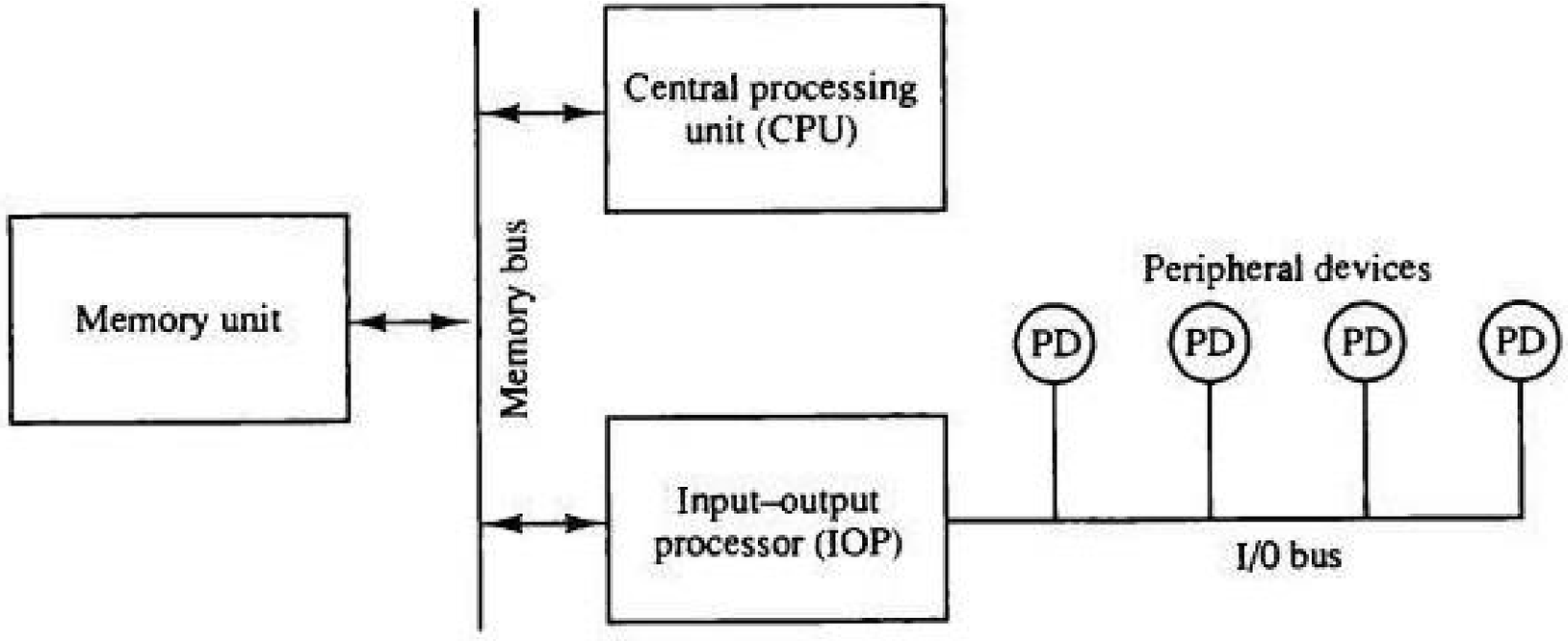
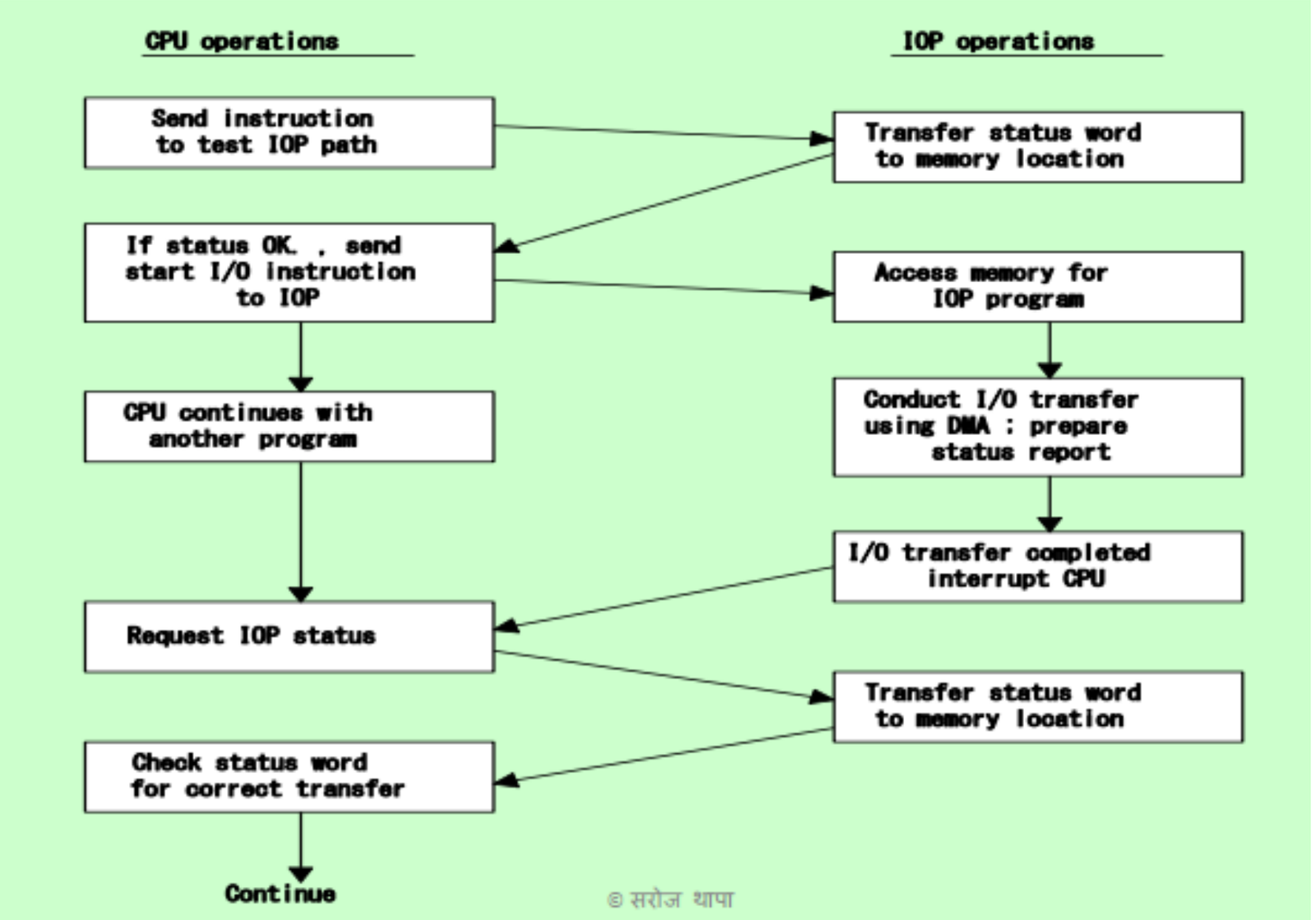


Fig: Block Diagram of computer with I/O Processor

Explanation of Block diagram

- The block diagram of a computer with two processor is shown in figure. The memory unit occupies a central position and can communicate with each processor by means of direct memory access. The CPU is responsible for processing data needed in the solution of the computational tasks. The IOP provides a path for transfer of data between various peripheral devices and the memory unit.
- The CPU is usually assigned the task of initiating the I/O program. From then, the IOP operates independent of the CPU and continues to transfer data from external devices and memory.
  The data formats of peripheral devices differ from memory and CPU data formats. Communication between IOP and devices attached to it is similar to program control method of transfer. Communication with the memory is similar to direct memory access method.

IO communication



End of UNIT 7