Modes of I/O Data Transfer

- Data transfer between the central computer and I/O devices may be handled in a variety of modes .
- Some modes use the CPU as an intermediate path others transfer the data directly to and from the memory unit.
- Data transfer between the central unit and I/O devices can be handled in generally three types of modes which are given below:

1. Programmed I/O
2. Interrupt Initiated I/O
3. Direct Memory Access

Programmed I/O

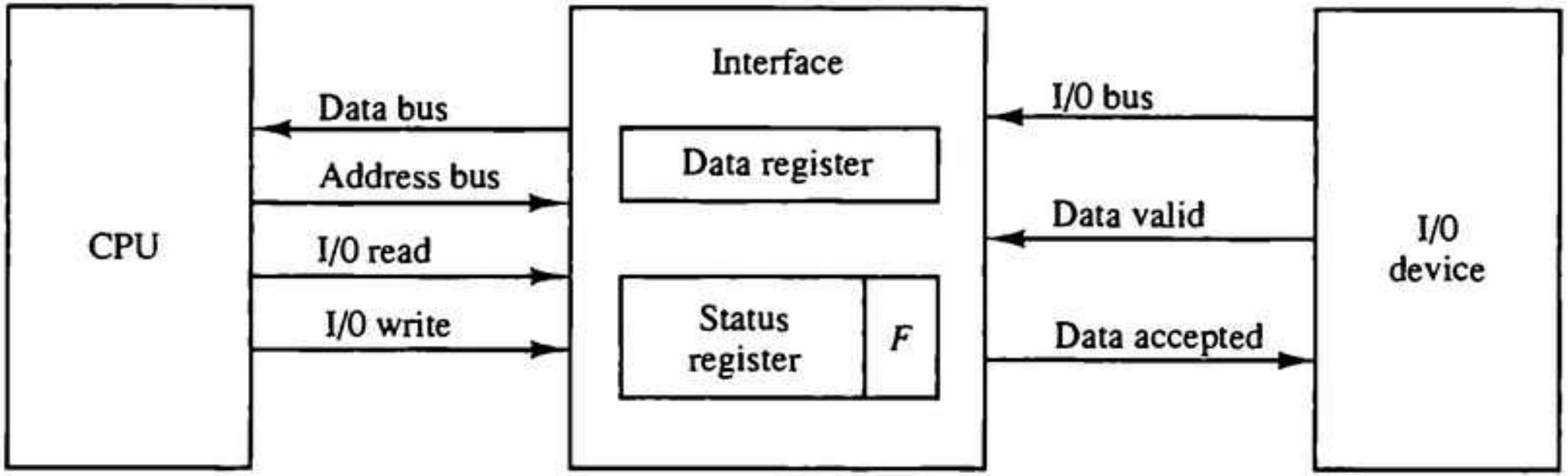
- Programmed input/output (PIO) is a method of transferring data between the CPU and a peripheral, such as a network adapter.
- Programmed I/O instructions are the result of I/O instructions written in computer program. Each data item transfer is initiated by the instruction in the program.
- Usually the program controls data transfer to and from CPU and peripheral.
- Transferring data under programmed I/O requires constant monitoring of the peripherals by the CPU.
- Once the data is initiated the CPU starts monitoring the interface to see when next transfer can made. The instructions of the program keep close tabs on everything that takes place in the interface unit and the I/O devices.

Programmed I/O:

- CPU requests I/O operation
- I/O module performs operations.
- I/O module sets status bits
- CPU checks status bits periodically
- I/O module does not inform CPU directly
- I/O module does not interrupt CPU
- CPU may wait or come back later

Programmed I/O

Figure 11-10 Data transfer from I/O device to CPU.



F = Flag bit

Dis-advantage of Programmed I/O:

- The problem with programmed I/O is that the processor has to wait a long time for the I/O module of concern to be ready for either reception or transmission of data.
- The processor, while waiting, must repeatedly interrogate the status of the I/O module.
- Performance of the entire system is severely degraded.

Interrupt Initiated I/O

- In the programmed I/O method the CPU stays in the program loop until the I/O unit indicates that it is ready for data transfer. This is time consuming process because it keeps the processor busy needlessly.
- This problem can be overcome by using **interrupt initiated I/O**.
- In this when the interface determines that the peripheral is ready for data transfer, it generates an interrupt.
- After receiving the interrupt signal, the CPU stops the task which it is processing and service the I/O transfer and then returns back to its previous processing task.

Interrupt Initiated I/O

- the interface meanwhile keeps monitoring the device. Whenever it is determined that the device is ready for data transfer it initiates an interrupt request signal to the computer.
- Upon detection of an external interrupt signal the CPU stops momentarily the task that it was already performing, branches to the service program to process the I/O transfer, and then return to the task it was originally performing.

Below are the basic operations of Interrupt:

- CPU issues read command
- I/O module gets data from peripheral whilst CPU does other work
- I/O module interrupts CPU
- CPU requests data
- I/O module transfers data

Direct Memory Access (DMA)

- The transfer of data between a fast storage device such as magnetic disk and memory is often limited by the speed of the CPU. Removing the CPU from the path and letting the peripheral device manage the memory buses directly would improve the speed of transfer. This transfer technique is called **Direct Memory Access (DMA).**
- During the DMA transfer, the CPU is idle and has no control of the memory buses. A DMA Controller takes over the buses to manage the transfer directly between the I/O device and memory.

- DMA module controls exchange of data between main memory and the I/O device.
- CPU is only involved at the beginning and end of the transfer and interrupted only after entire block has been transferred.
- Direct Memory Access needs a special hardware called DMA controller (DMAC) that manages the data transfers and arbitrates access to the system bus.

- During DMA transfer, the CPU is idle and has no control of the memory buses. The buses can be disabled by using two special control signals.
  - Bus Request (BR)
  - Bus Grant(BG)

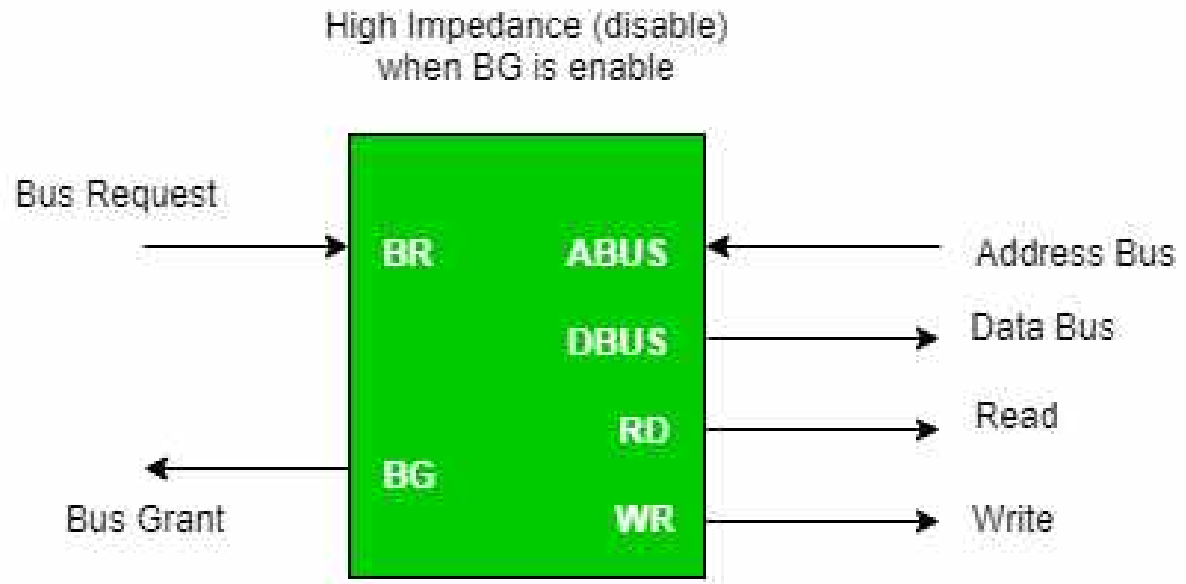


Figure - CPU Bus Signals for DMA Transfer

- **Bus Request** : It is used by the DMA controller to request the CPU to relinquish the control of the buses.

- **Bus Grant :** It is activated by the CPU to Inform the external DMA controller that the buses are in high impedance state and the requesting DMA can take control of the buses. Once the DMA has taken the control of the buses it transfers the data. This transfer can take place in many ways.

DMA Controller

- During the DMA transfer, the CPU is idle and has no control of the memory buses. A DMA controller takes over the buses to manage the transfer directly between the I/O device(s) and main memory. The structure of DMA controller is described below.

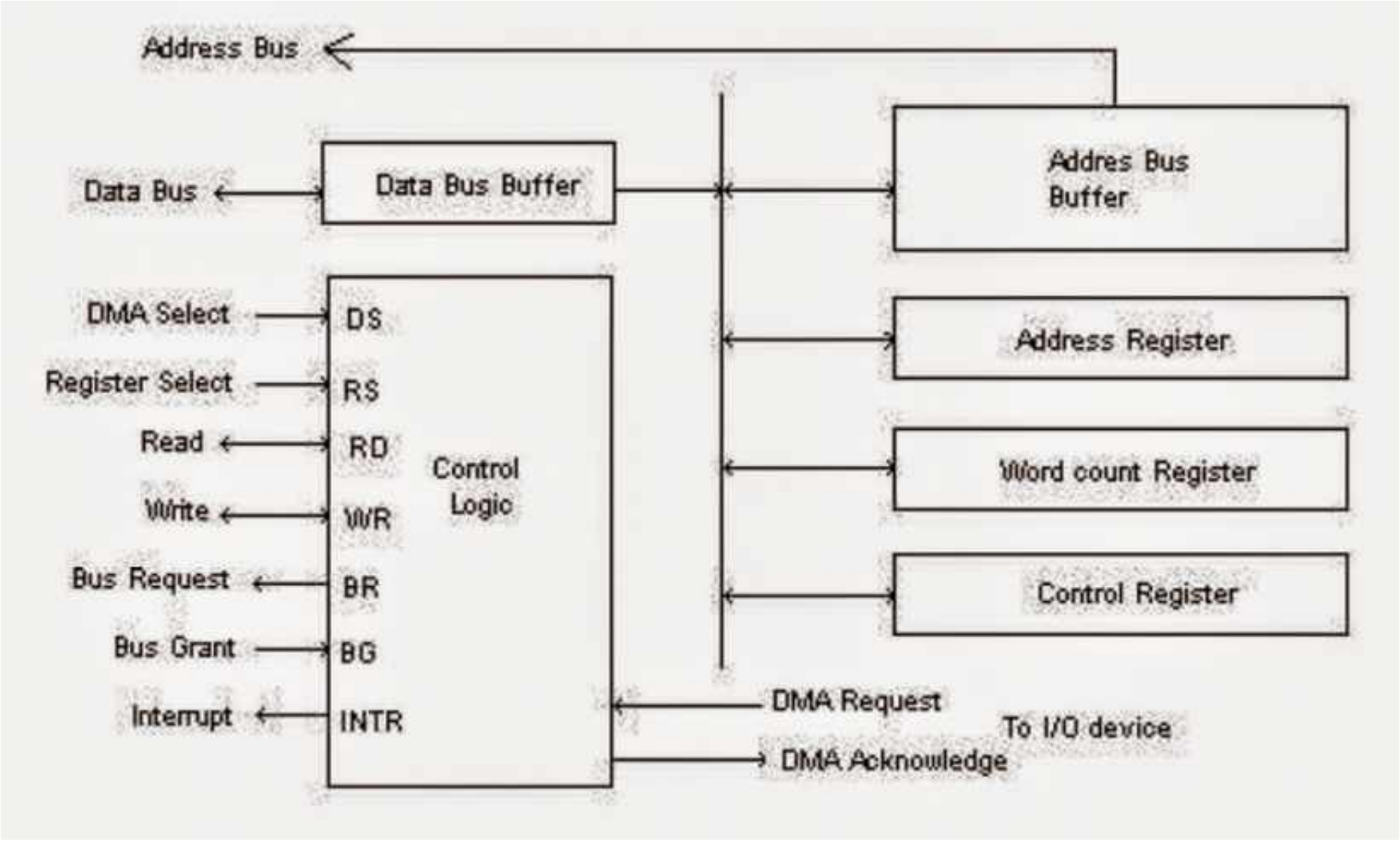


Figure: Block Diagram of DMA Controller

- The control unit communicates the CPU via data bus and control lines.
- The DMA controls the system bus using BR (Bus Request) and BG (Bus Grant) signals.
- DMA operates read and write operations via RD (Read) and WR (Write) signals.
- DMA sends request and acknowledge to I/O devices via DMA request and DMA acknowledge signals.
- The registers in DMA are selected by CPU through the address bus by enabling DS (DMA Select) and RS (Register Select) inputs.
- All registers in the DMA appear to the CPU as I/O interface registers.
    - The address register contains an address to specify the desired location in memory.
    - The word count register holds the number of words to be transferred.
    - The control register specifies the mode of transfer.

Types of DMA transfer using DMA controller:

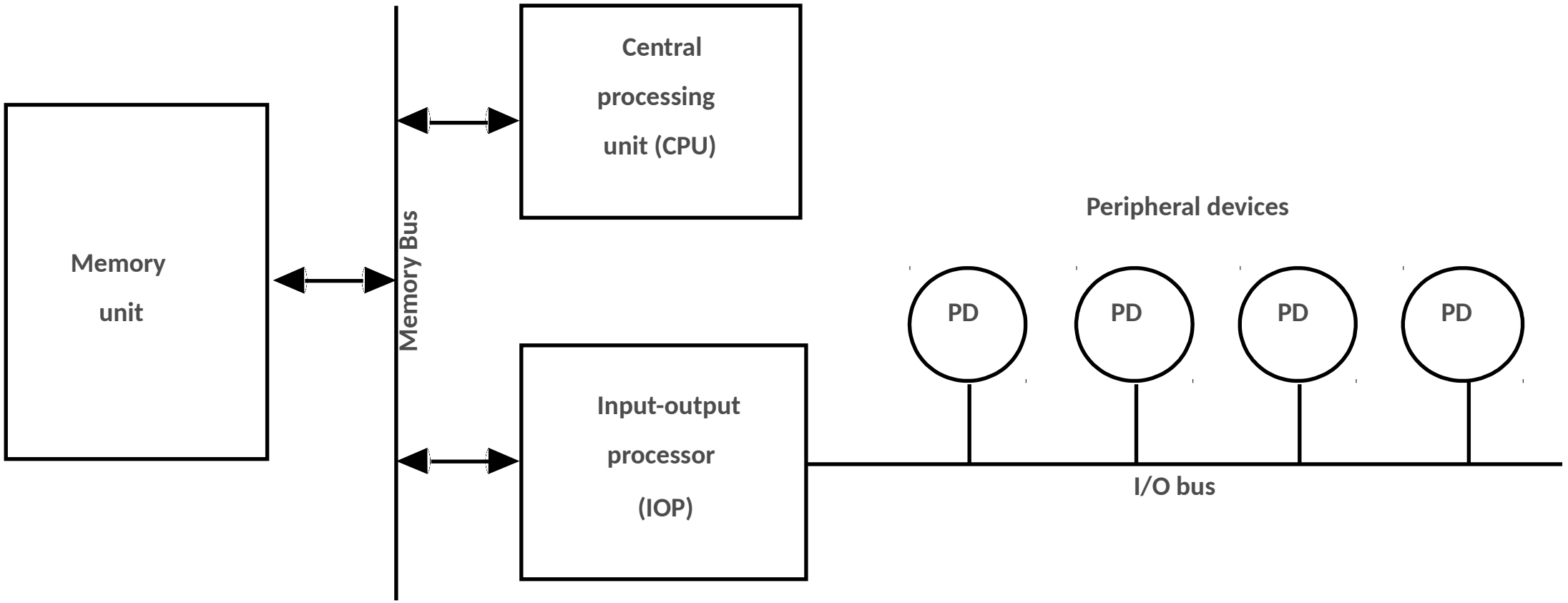
- **Burst or block transfer DMA**
  - It is the fastest DMA mode.
  - In this mode, two or more data bytes are transferred continuously.
- **Cycle Steal or Single Byte Transfer DMA**
  - In cycle steal transfer only one byte of data is transferred at a time.
  - This type of DMA is slower than burst DMA.

I/O Processor

- I/O Processors also known as:
  - I/O Controllers
  - Channel Controllers
  - Peripheral Processing units (PPU)
  - Data Channel

I/O Processor

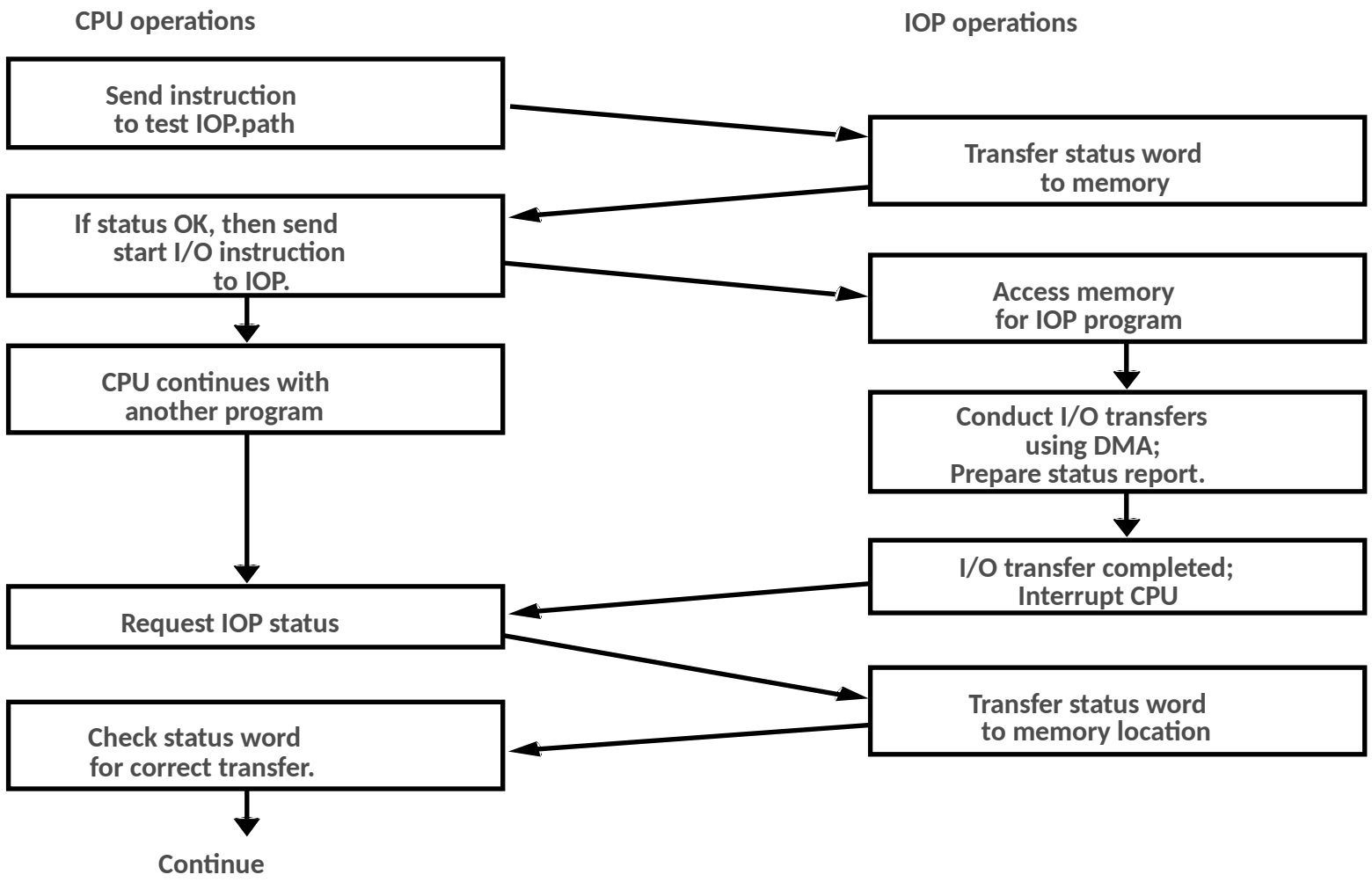
- IOP is similar to a CPU except that is designed to handle the details of I/O processing
- The IOP can fetch and execute its own instruction. IOP instruction are specifically designed to facilitate I/O transfer.
- IOP may be classified as a processor with direct memory access capability that communicate with I/O devices
- Each IOP takes care of input and output task , relieving the CPU from the house keeping chores involved in I/O transfers.


Memory
unit
Memory Bus
Central
processing
unit (CPU)
Input-output
processor
(IOP)
Peripheral devices
PD
PD
PD
PD
I/O bus

Working

- 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 computational tasks.
- The IOP provides a path for transfer of data between various peripheral devices and the memory unit.
- The CPU usually assigned the task of initiating the I/O program.
- From then on the IOP operates independent of the CPU and continious to transfer data from external device and memory.

CPU-IOP Communication



Explaination

- The CPU sends an instruction to test the IOP path.
- The IOP responds by inserting a status word in memory for the CPU to check.
- The staus word indicates the condition of the IOP and device. such as : IOP overload condition , device busy with another transfer , or device ready for I/O transfer.
- The CPU refers to the status word in memory to decide what to do next.
- If all is in order , the CPU sends the instruction to start I/O transfer.
- The memory address recieved with the instruction tells the IOP where to find its program.

- The CPU can now continue with another program while the IOP is busy with the I/O program.
- IOP conduct I/O transfer using DMA.
- When IOP complete the I/O transfer , it send an interrupt request to the CPU and prepare a status report and place it into a specified memory location.
- The status word indicates whether the transfer has been completed or if any errors occured during the transfer.