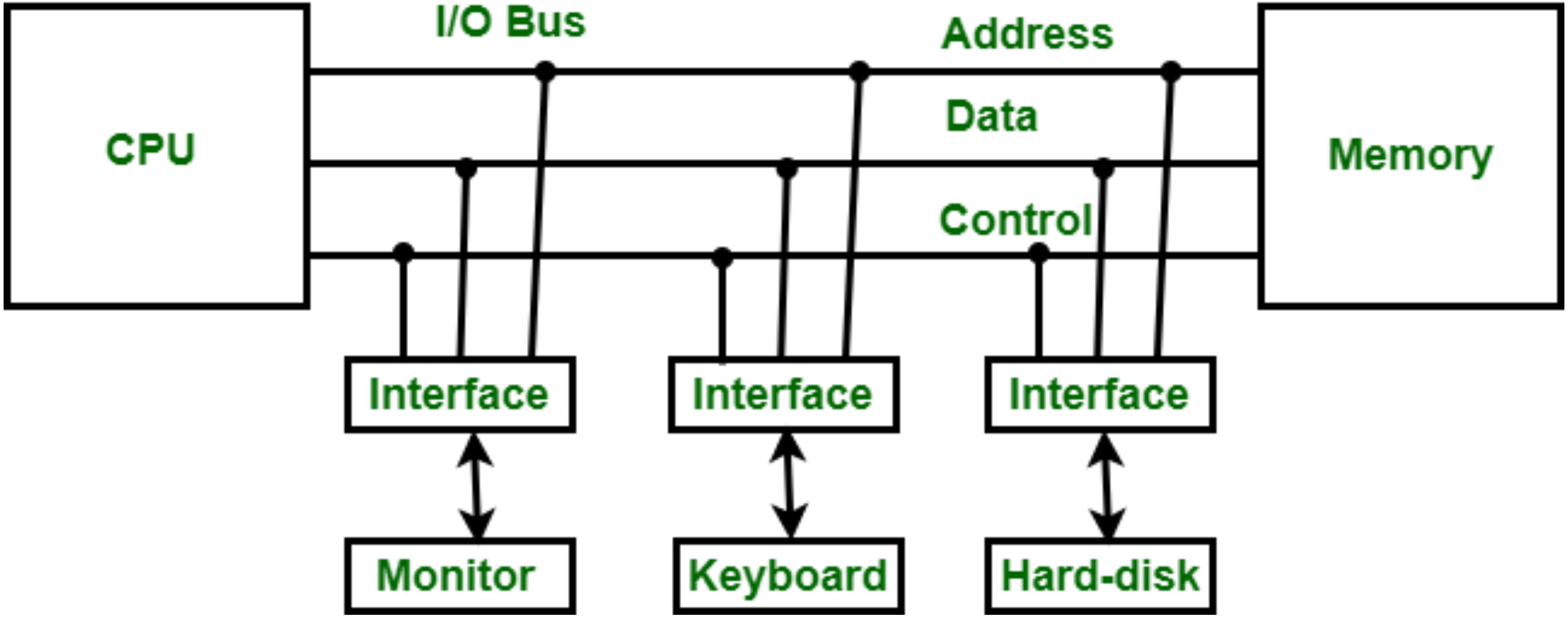
I/O Interface

I/O Interface

- **Input-Output Interface** is used as an method which helps in transferring of information between the internal storage devices i.e. memory and the external peripheral device .
- A **peripheral device** is that which provide input and output for the computer, it is also called Input-Output devices.
- A keyboard and mouse provide Input to the computer are called input devices while a monitor and printer that provide output to the computer are called output devices.
- Just like the external hard-drives, there is also availability of some peripheral devices which are able to provide both input and output.


I/O Bus
Address
Data
Control
CPU
Memory
Interface
Interface
Interface
Monitor
Keyboard
Hard-disk

Why Interface between I/O and CPU?

- There is a special need of the additional hardware to resolve the differences between CPU and peripheral devices to supervise and synchronize all input and output devices.
- There exists special hardware components between CPU and peripherals to supervise and synchronize all the input and output transfers that are called **interface units**.
    - There is a lot of difference in the mode of operation of both peripheral devices and CPU.
    - There is also a synchronization mechanism because the data transfer rate of peripheral devices are slow than CPU.

- In peripheral devices, data code and formats are different from the format in the CPU and memory.
- The operating mode of peripheral devices are different and each may be controlled so as not to disturb the operation of other peripheral devices connected to CPU.

Functions of Input-Output Interface:

- It is used to synchronize the operating speed of CPU with respect to input-output devices.
- It selects the input-output device which is appropriate for the interpretation of the input-output device.
- It is capable of providing signals like control and timing signals.

- There are various error detectors.
- It converts serial data into parallel data and vice-versa.
- It also convert digital data into analog signal and vice-versa.

Mode of transfer

- Data transfer to and from the peripherals may be done in any of the three possible ways
    - Programmed I/O.
    - Interrupt- initiated I/O.
    - Direct memory access(DMA).

Programmed I/O

- 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**.
- In programmed I/O, the CPU stays in the program loop until the I/O unit indicates that it is ready for data transfer. This is a time consuming process since it needlessly keeps the **CPU busy**. This situation can be avoided by using an interrupt facility.

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.
- 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.

- Both the methods programmed I/O and Interrupt-driven I/O require the active intervention of the processor to transfer data between memory and the I/O module, and any data transfer must transverse a path through the processor. Thus both these forms of I/O suffer from two inherent drawbacks.

    - The I/O transfer rate is limited by the speed with which the processor can test and service a device.
    - The processor is tied up in managing an I/O transfer; a number of instructions must be executed for each I/O transfer.

Direct Memory Access (DMA)

- Removing the CPU from the path and letting the peripheral device manage the memory buses directly would improve the speed of transfer. This technique is known as **DMA**.
- In this, the interface transfer data to and from the memory through memory bus. A DMA controller manages to transfer data between peripherals and memory unit.
- Many hardware systems use DMA such as disk drive controllers, graphic cards, network cards and sound cards etc. It is also used for intra chip data transfer in multicore processors. In DMA, CPU would initiate the transfer, do other operations while the transfer is in progress and receive an interrupt from the DMA controller when the transfer has been completed.

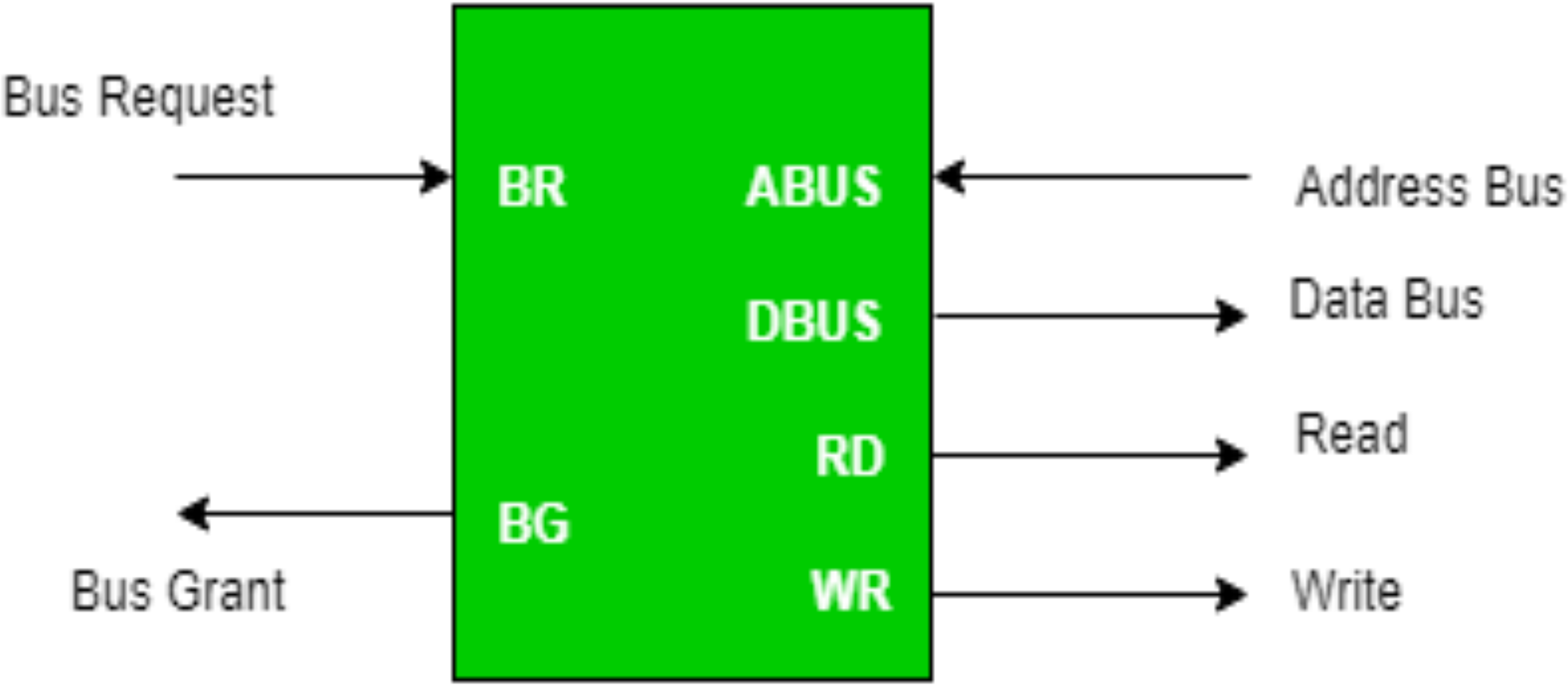


Figure - CPU Bus Signals for DMA Transfer

- **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.

Types of DMA transfer using DMA controller:

- **Burst Transfer :**
  DMA returns the bus after complete data transfer. A register is used as a byte count, being decremented for each byte transfer, and upon the byte count reaching zero, the DMAC will release the bus. When the DMAC operates in burst mode, the CPU is halted for the duration of the data transfer.
- **Cyclic Stealing :**
  An alternative method in which DMA controller transfers one word at a time after which it must return the control of the buses to the CPU. The CPU delays its operation only for one memory cycle to allow the direct memory I/O transfer to “steal” one memory cycle.

Parallel and Serial Interface

Sr. No.	Parameters	Serial Interface	Parallel Interface
1.	**Main Concept**	One bit at a time.	Multiple bits at a time.
2.	**Components**	Serial interface consists of an I2C bus, SPI bus, or synchronous serial control and data lines.	Parallel interface consists of 8 data pins and 3 control lines.
3.	**Cost**	It requires fewer pins and is easy to set up hence the cost is comparatively less.	It requires more pins and is not easy to set up hence the cost is comparatively high.

4.	**Purpose**	Serial interface used for connecting serial lines to facilitate serial communication.	Parallel is an external interface used for connecting computer peripherals such as printers, or any other device that requires relatively high bandwidth, to a PC.
5.	**Transmission**	In serial interface bits of data are sent one at a time in a single stream of 1s and 0s along a single wire.	Parallel interface can move a set of 8 bits (1s and 0s) of data in parallel at the same time along 8 separate wires meaning multiple bits of data are sent simultaneously.
6.	**Speed**	Transmission speed is Low.	Transmission speed is High.

7.	**Number of Wires**	Less number of wires are used in serial interface.	More wires are used in parallel interface.
8.	**Data Sending Mechanism**	A serial interface sends data bit by bit after sending a bit at a time.	A parallel interface sends data by sending multiple bits in a parallel fashion.
9.	**Capability**	A serial interface can transmit a single stream of data at a time.	A parallel interface can transmit multiple data streams at a time.

10.	**Port Type**	A serial interface uses Male ports.	A parallel interface uses Female ports.
11.	**Redundancy**	Bottom-Up model is better suited as it ensures minimum data redundancy and focus is on re-usability.	Top-down model has a high ratio of redundancy as the size of the project increases.
12.	**Applications**	Modems, security cameras, device controllers, GPS Receivers, Telescopes, Power Inverters, and Flat-Screen Monitors use serial ports.	Printers, Hard Drives, CD drives, Scanners, External CD-ROM, and Optical Drivers use parallel ports.