

Input Output Organization



➤ IO Interface



IO Interface



- IO interface provides a method for transferring information between internal system (CPU) and external I/O devices.
- I/O devices can not directly communicate with the CPU due to various differences between them.

Significance of I/O Interface

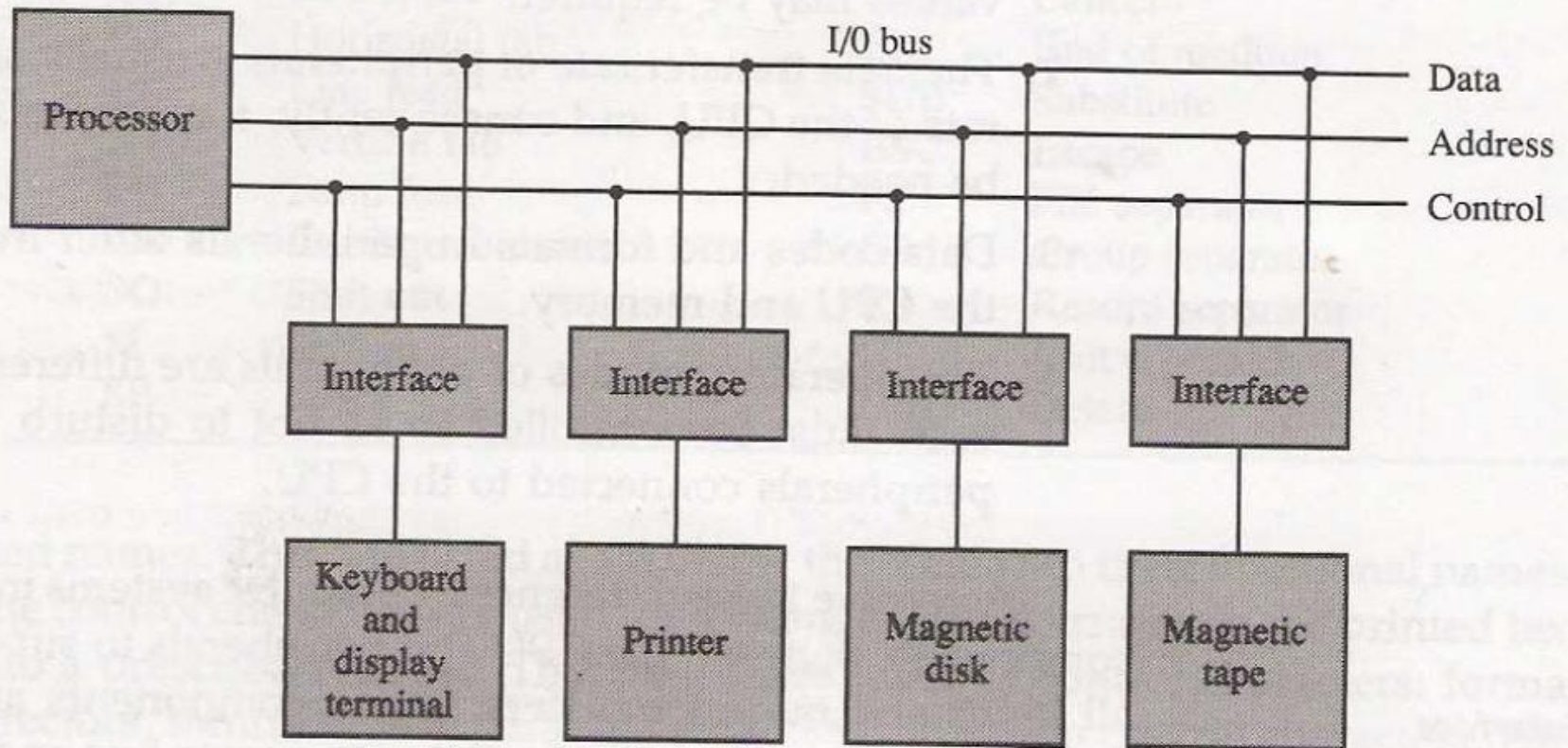


1. Peripherals are electromechanical and electromagnetic devices and their manner of operation is different from the operation of the CPU and memory, which are electronic devices. Therefore, a conversion of signal values may be required.
2. The data transfer rate of peripherals is usually slower than the transfer rate of the CPU, and consequently, a synchronization mechanism may be needed.
3. Data codes and formats in peripherals differ from the word format in the CPU and memory.
4. The operating modes of peripherals are different from each other and each must be controlled so as not to disturb the operation of other peripherals connected to the CPU.

I/O Bus and Interface Modules



Figure 1 Connection of I/O bus to input-output devices.





- I/O bus is attached to all peripherals.
- Interface monitors the address on address bus.
- If it find its own address, it activates path between bus line and device.
- All the other devices are disabled by their interfaces.
- Data to read or write is made available in the data bus.
- IO command is made available in the control bus.



- **There are four types of IO command:**
 - Control Command
 - Status Command
 - Output Data Command
 - Input Data Command

IO versus Memory bus



- Like IO bus Memory bus also contain data, address and control lines.
- There are two ways in which computer bus communicates with memory and IO.
 1. Use two separate buses, one for memory and other for I/O (Isolated I/O)
- Use one common bus for memory and I/O common control lines (Memory Mapped I/O)



- In the first method, CPU has independent set of buses for both memory and IO. It is done in computers that has separate IOP and CPU.
- In second method computers use common bus to transfer data between IO or memory and CPU.



- This configuration isolates all I/O interface address with the memory address and is referred to as **Isolated IO** method.
- In memory mapped method, same address space is used for both memory and IO interface. They have only one set of read and write signals.
- Computer treats interface as being part of the memory system.



- It assign address for each interface which cannot be used for memory words which reduces the memory address range available.
- This configuration is referred to as **memory-mapped IO**.

Modes of transfer



- Data transfer between computer and IO device can be handle in the following ways:
 - Programmed IO
 - Interrupt-Initiated IO
 - Direct Memory Access(DMA)

Programmed IO

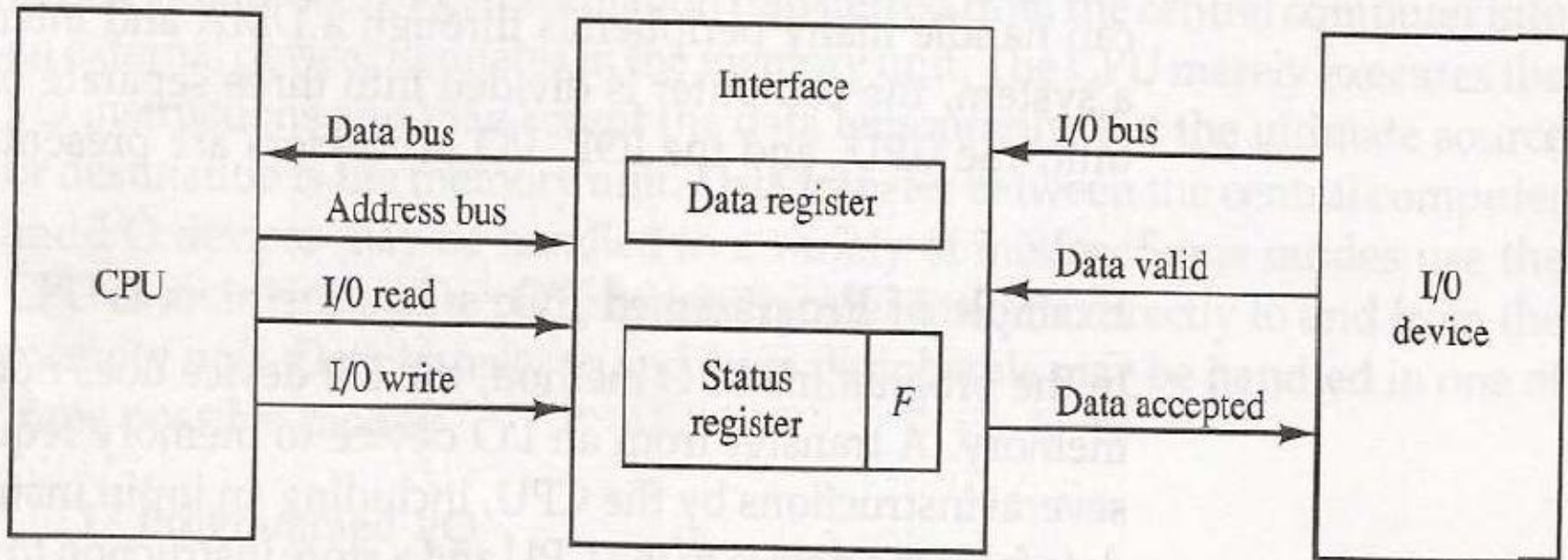


- Each data transfer is initiated by an instruction in the program.
- CPU stays in the program loop until IO unit indicates it is ready for the data transfer.
- It is time consuming process since it keeps processor busy needlessly.

Example of programmed IO



Figure Data transfer from I/O device to CPU.



F = Flag bit



- The transfer of data requires three instructions:

1. Read the status register.
2. Check the status of the flag bit and branch to step 1 if not set or to step 3 if set.
3. Read the data register.

Flow chart of programmed I/O from e-book



Interrupt Initiated IO



- In this type of IO, computer does not check the flag. It continue to perform its task.
- Whenever any device wants the attention, it sends the interrupt signal to the CPU.
- CPU then deviates from what it was doing, store the return address from PC and branch to the address of the subroutine.



- There are two ways of choosing the branch address:
 - Vectored Interrupt
 - Non-vectored Interrupt
- In vectored interrupt the source that interrupt the CPU provides the branch information. This information is called interrupt vectored.
- In non-vectored interrupt, the branch address is assigned to the fixed address in the memory.

Direct Memory Access(DMA) (11.6)



- In DMA there is direct communication between memory and the peripheral devices.
- **Direct memory access (DMA)** is a feature of modern computer systems that allows certain hardware subsystems to read/write data to/from memory without microprocessor intervention, allowing the processor to do other work.
- Typically, the CPU initiates DMA transfer, does other operations while the transfer is in progress, and receives an interrupt from the DMA controller once the operation is complete.

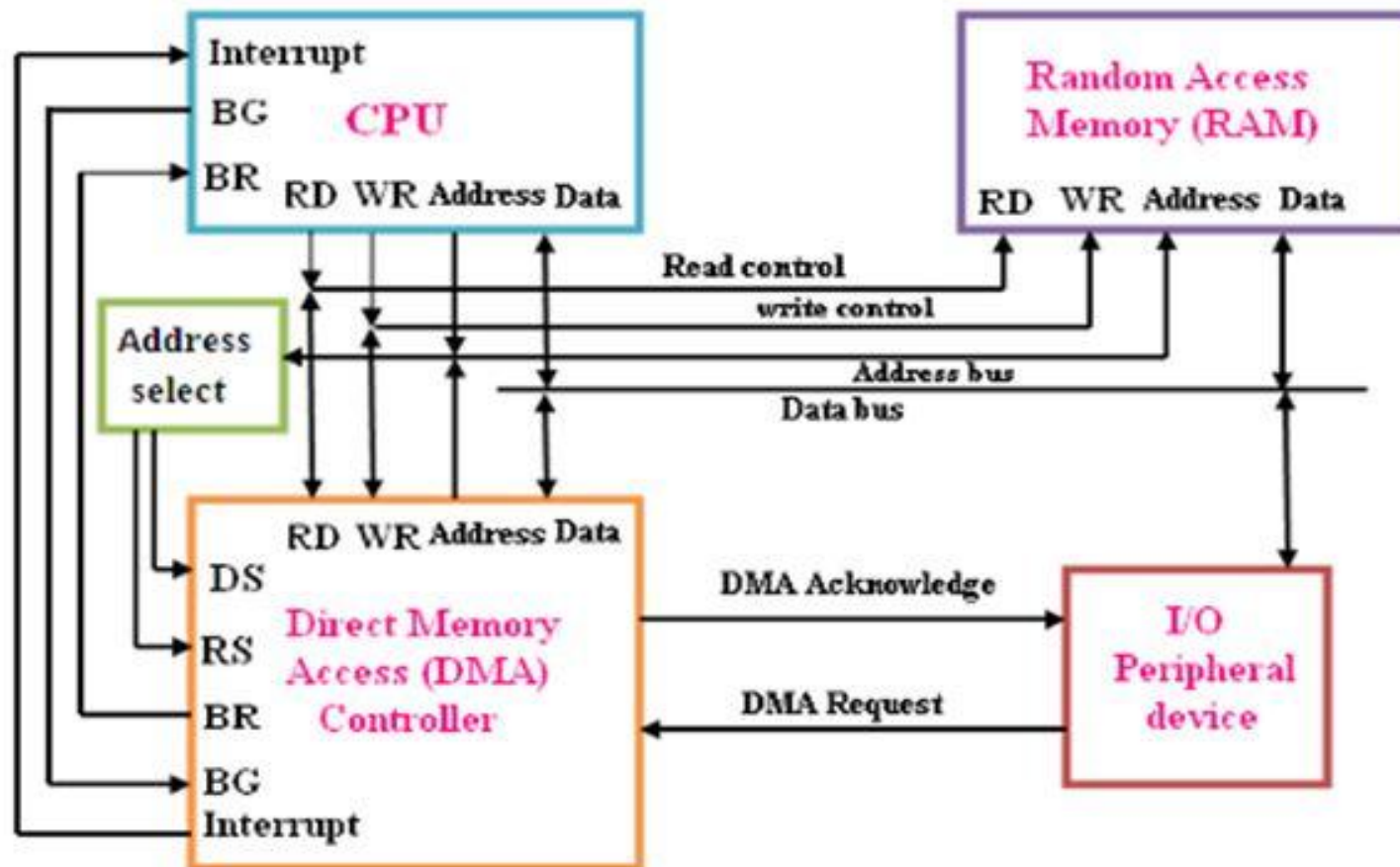


Fig: DMA Transfer



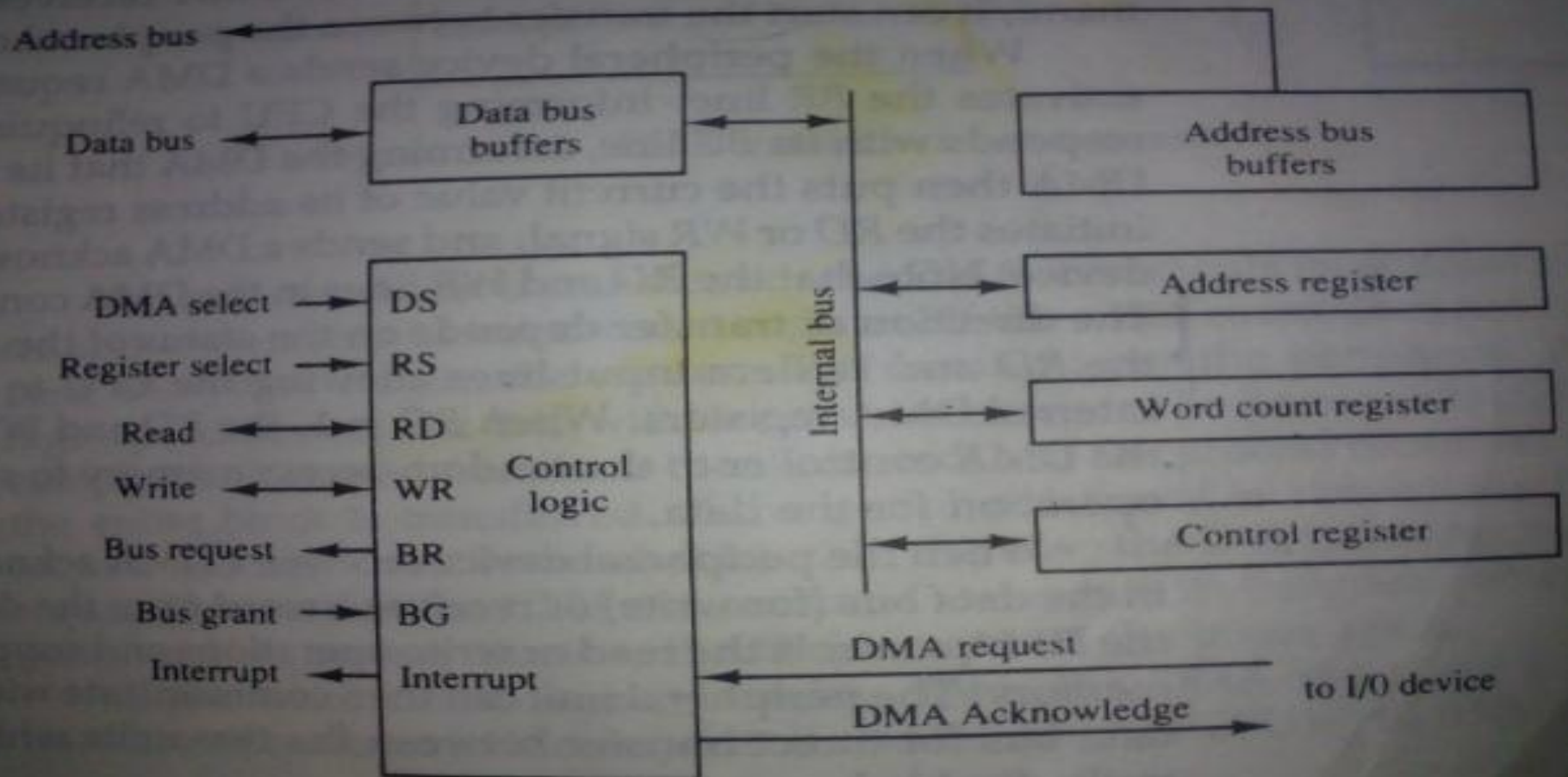
- BR(Bus Request) signal is used by DMA controller to request CPU for the buses.
- CPU then places the bus on the high impedance state which behave like open circuit.
- CPU then activates BG(Bus Grant) signal to acknowledge BR signal.
- DMA now has full control over the buses and perform the transfer.



- Transfer can be performed in two ways:
 - Burst Transfer
 - Cycle Stealing
- In burst transfer, a number of memory words is transferred in a continuous burst. It is done while communicating with fast devices and can't be stopped or slowed down.
- In cycle stealing, one data word is transferred at a time.
- CPU delays its operation for one cycle.

DMA Controller

Figure 11-17 Block diagram of DMA controller.



DMA Controller



- It communicates with CPU through data bus and control lines.
- Registers in DMA are selected by CPU by enabling DS and RS.
- RD and WR signals are for read and write operation.
- When $BG=0$ CPU can communicate with DMA register for read or write operation.
- When $BG=1$ DMA communicate directly with the memory.

DMA Controller



- It has three registers: address, word count and control register.
- Address register contains address which specifies the location of memory to read or write.
- It is incremented after each word is transferred into memory.
- Word count register holds the number of words to be transferred.
- It is decremented by one after each word is transferred into memory and regularly checked for zero.



- Control Register specify the mode of transfer.
- DMA is first initialized by CPU. After that DMA continue to transfer data.
- CPU initialize the DMA by sending the following information through the data bus:
 - Starting address of memory block for read or write.
 - The word count or number of words to read or write.
 - Control to specify mode such as read or write.
 - Control to start DMA



- Once DMA is initialized, CPU stop communicating with DMA unless it receive interrupt signal or if it wants to check how many words has been transferred.