Input - Output Interface

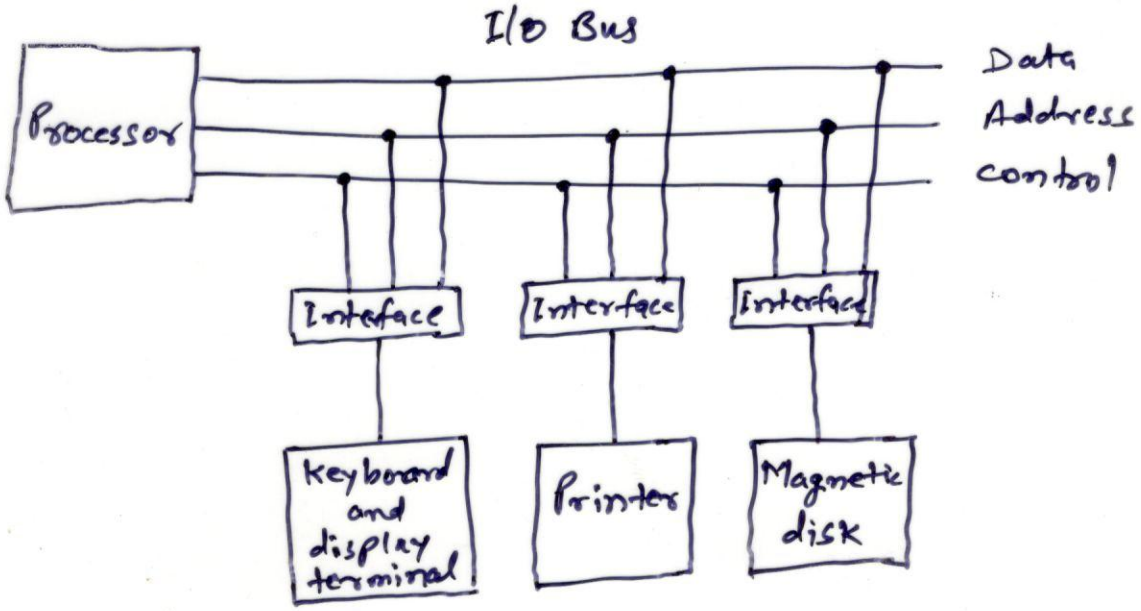
- Input-output interface provides a method for transferring information between internal storage and external I/O devices.

- Peripherals ~~connected~~ need communication links for interfacing ~~that~~ that resolve the differences.

Major differences:

1. ~~*~~ convesion of signal
   (digital to analog and vice-versa)
2. ~~*~~ synchronization : The data transfer rate of peripherals is usually slower than the transfer rate of the CPU.
3. Data codes and ~~formats~~ in peripherals differ from the word format in the CPU and memory.
4. The operation modes of peripherals are different from each other.

I/O Bus and Interface Modules



- The I/o bus consists of data lines, address lines, and control lines.
- Each peripheral device has associated with it an interface unit.
- Each peripheral has its own controller that operates the particular electromechanical device.
- Each interface decodes the address and control lines.
- And synchronize the data flow and supervises the transfer between peripheral and processor.

I/O versus Memory Bus

In addition to communicating with I/O, the processor must communicate with the memory unit. Like the I/O bus, the memory bus contains data, address, and read/write control lines.

There are three ways that computer buses can be used to communicate with memory and I/O.:

1. Use two separate buses, one for memory and the other for I/O.
2. Use one common bus for both memory and I/O but have separate control lines for each.
3. Use one common bus for memory and I/O with common control lines.

Asynchronous Data Transfer

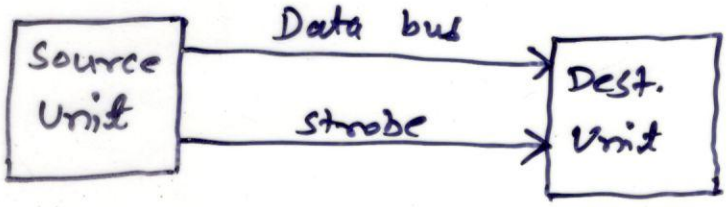
- The internal operation in a digital system are synchronized by means of clock pulses supplied by a common pulse generator.
- Clock pulses are applied to all registers within a unit and all data transfer among internal registers occur simultaneously.
- If the registers in the interface share a common clock with the CPU registers, the transfer between the two units is said to be synchronous.
- If the internal timing in each unit is independent mean each unit has its own private clock for internal registers is said to be asynchronous.

Control Method of asynchronous data transfer :

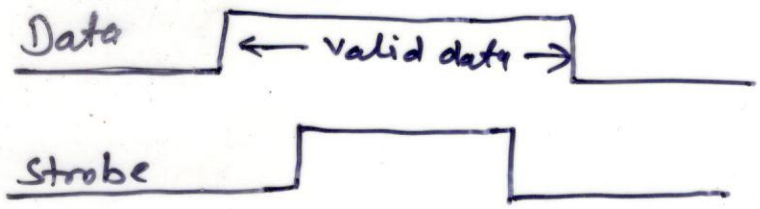
1. Strobe
2. Handshaking

1. Strobe Control

The strobe control method of asynchronous data transfer employs a single control line to time each transfer. The ~~strobe~~ strobe may be activated by either the source or the destination unit.



(a) Block diagram

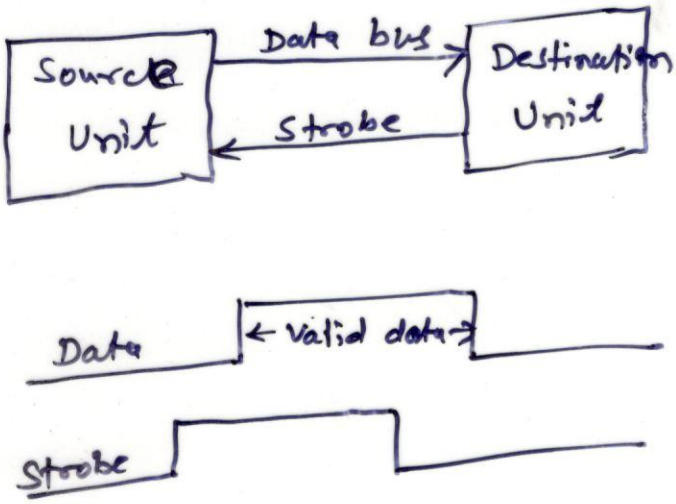


(b) Timing diagram

(Source-initiated strobe for data transfer)

- The data bus carrying the binary information
- The strobe is a ~~signle~~ single line that informs the destination unit when a valid data word is available in the bus.

Destination - initiated strobe for data transfer:

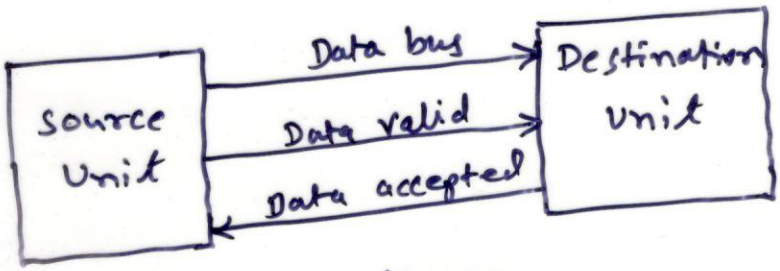


- In this case the destination Unit activates the strobe pulse, informing the source to provide the data.
- The source Unit responds by placing the requested binary information on the data bus.
- The data must be valid and remain in the bus long enough for the destination unit to accept it.

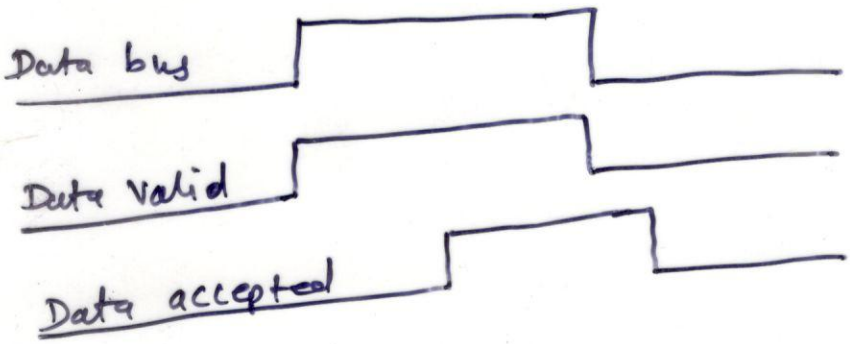
2. Handshaking

The handshake method introduce a second control signal that provides a reply to the unit that initiates the transfer.

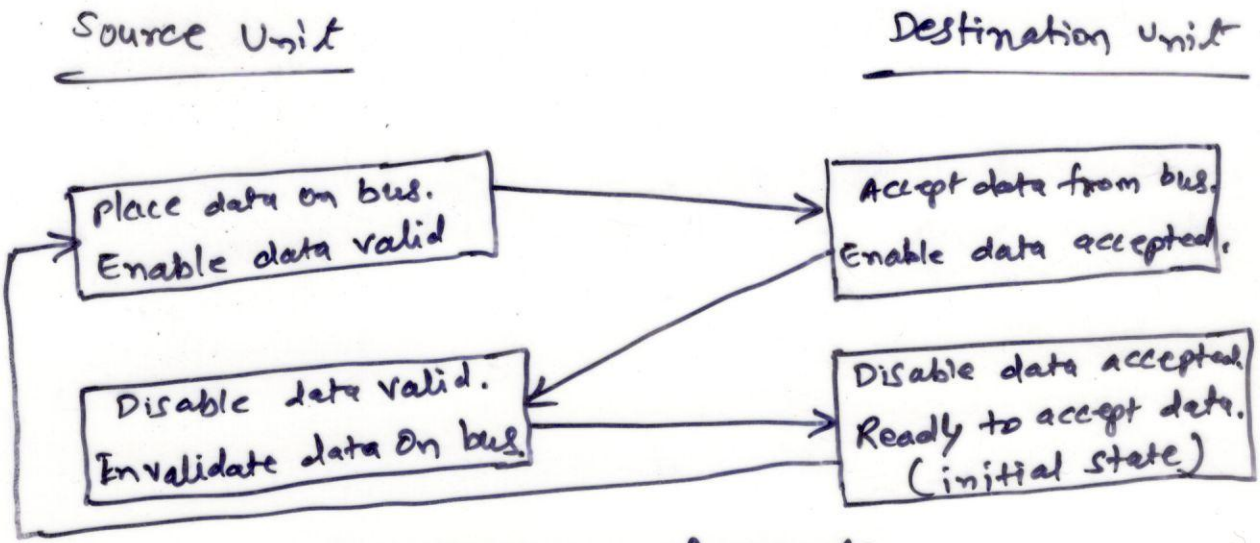
Source-initiated transfer



(a) Block diagram

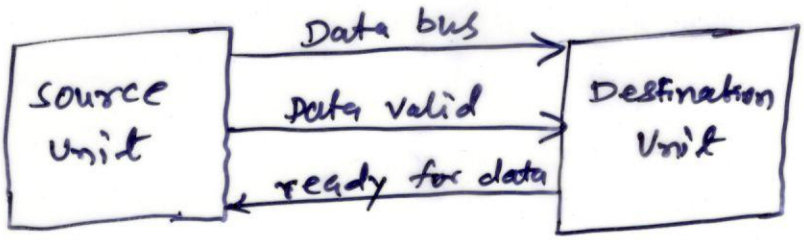


(b) Timing diagram

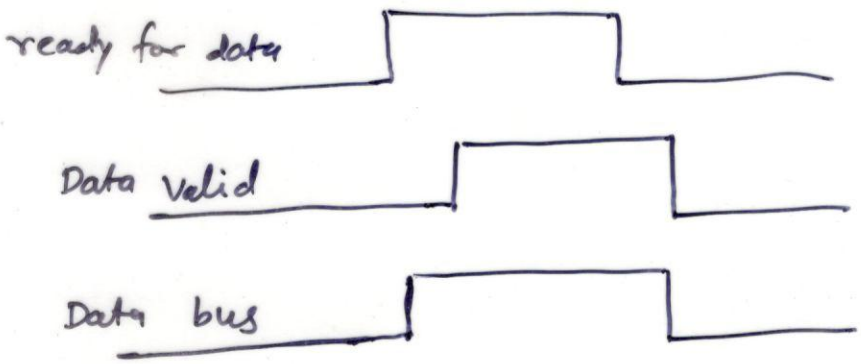


(c) Sequence of events

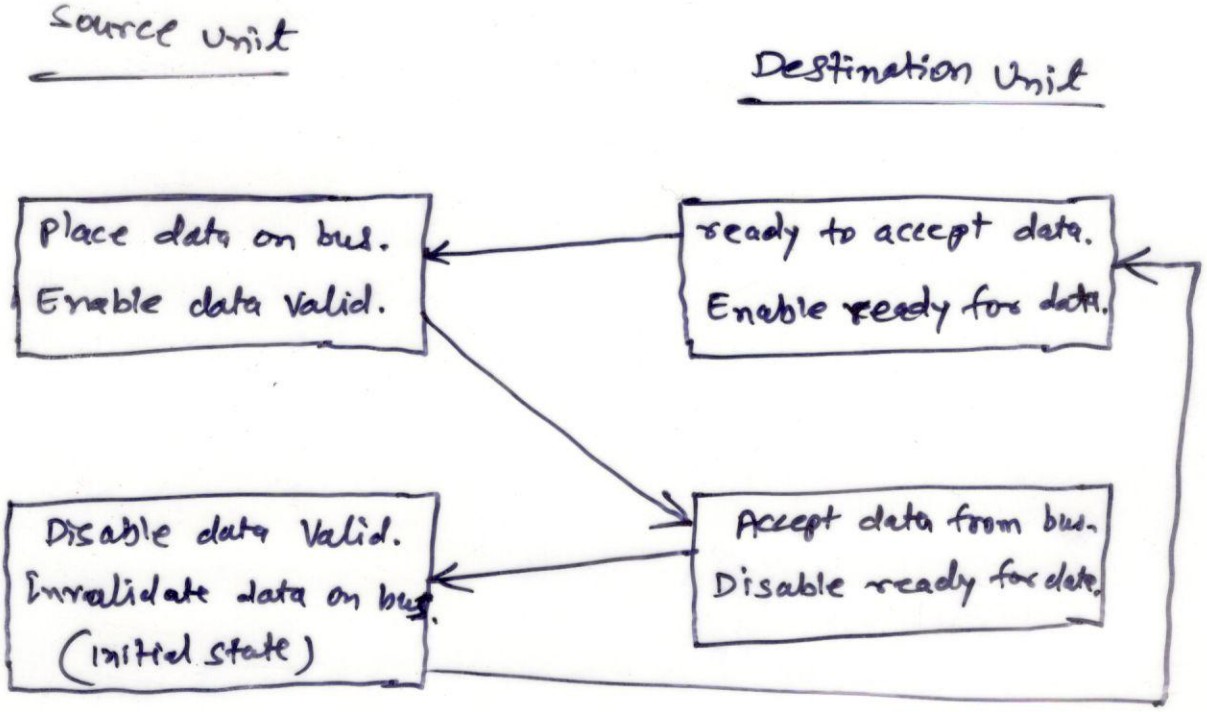
Destination-initiated transfer:



(a) Block diagram



(b) Timing diagram



(c) Sequence of events

Modes of Transfer

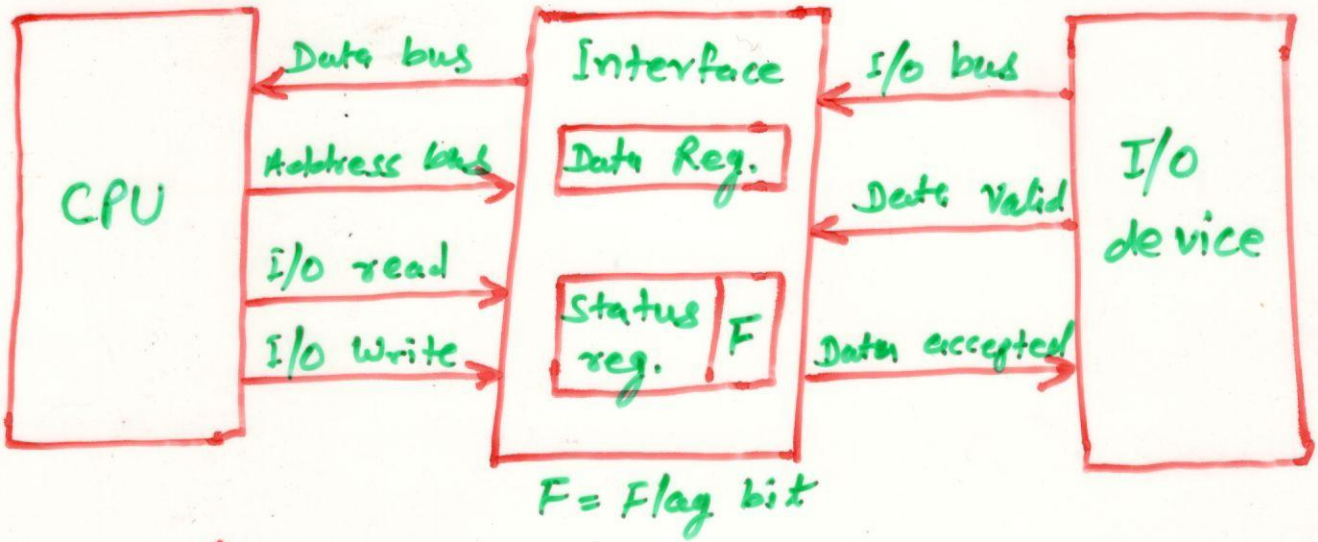
Data transfer between the central computer and I/O devices may be handled in a variety of modes.

1. Programmed I/O
2. Interrupt-initiated I/O
3. Direct memory access (DMA)

Programmed I/O

In the programmed I/O method, the I/O device does not have direct access to memory. A transfer from an I/O device to memory requires the execution of several instructions by the CPU.

1. An input instruction to transfer the data from the device to the CPU.
2. A store instruction to transfer the data from the CPU to memory.



F = Flag bit

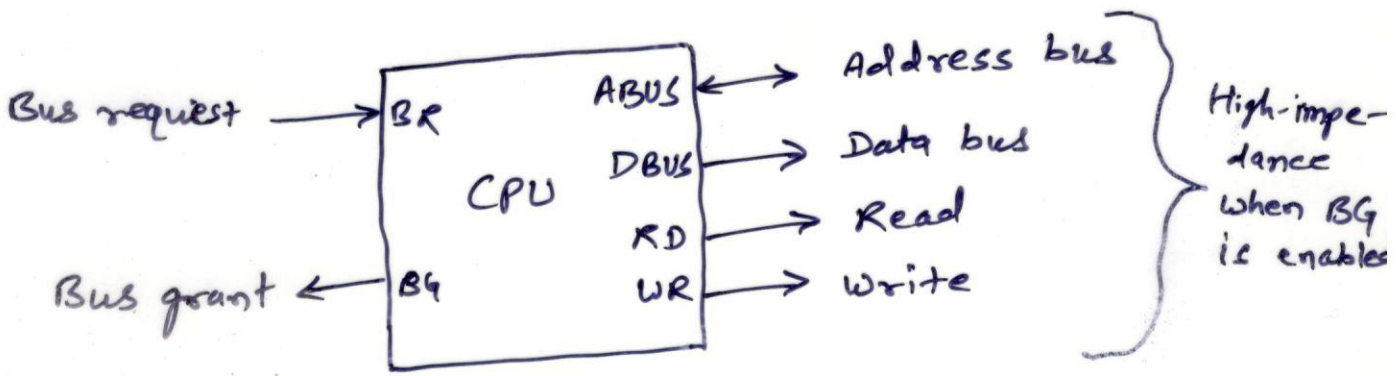
(Data transfer from I/O device to CPU)

Direct Memory Access (DMA)

The transfer of data between memory and peripheral device directly without using of CPU.

- The peripheral device manage the memory buses directly would improve the speed of transfer.
- During 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.

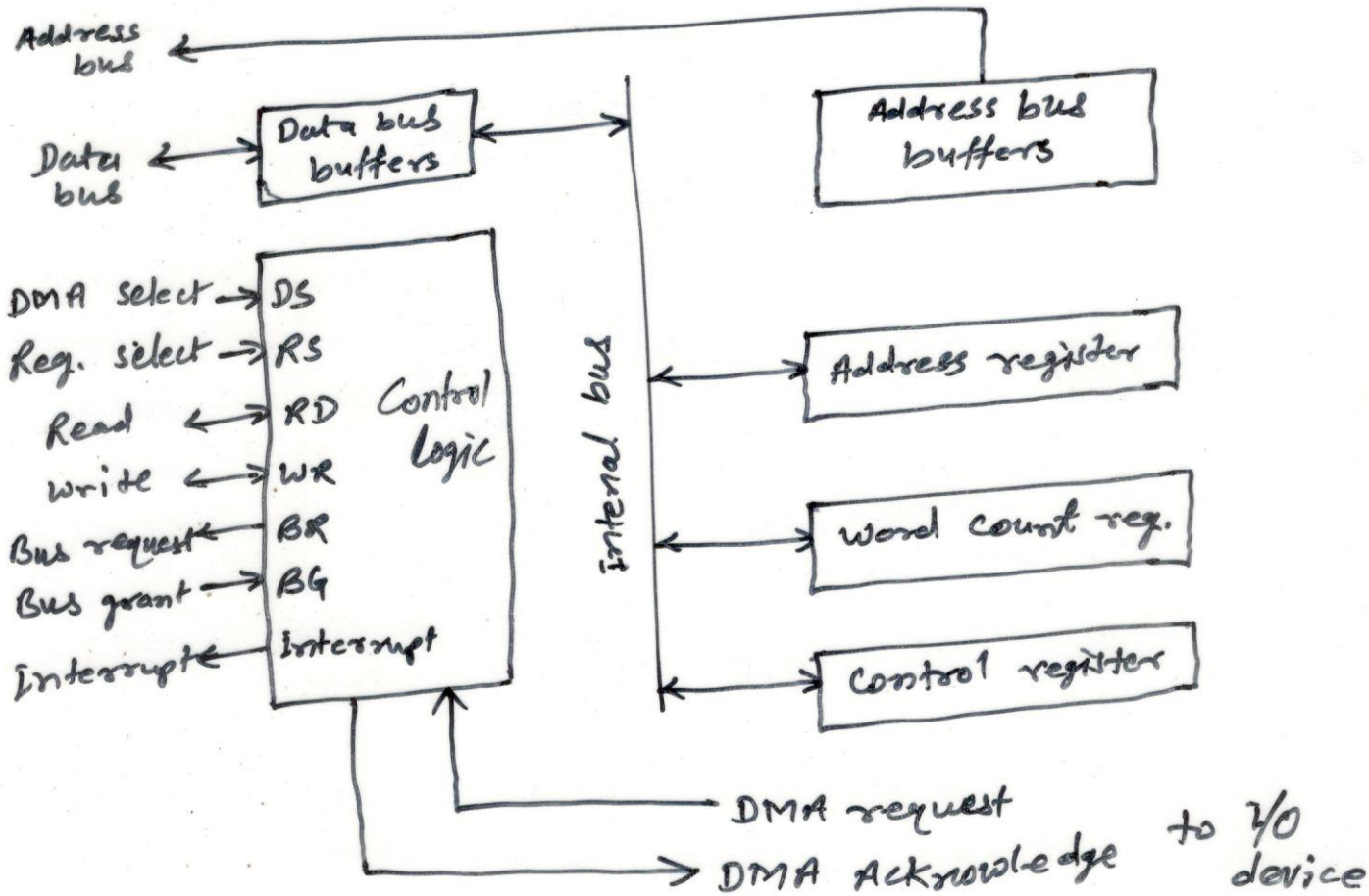
CPU bus signals for DMA transfer



- The bus request (BR) input is used by the DMA controller to request the CPU for control of the buses.
- When BR became active, the CPU terminates the execution of the current instruction and places the address bus, the data bus, and the read and write lines into a high-impedance state.
- The CPU activates the bus grant (BG) output then the DMA controller take control of the buses to conduct memory transfers without processor intervention.
- When the DMA terminates the transfer, it disables the bus request line. The CPU disables the bus grant, takes control of the buses, and returns to its normal operation.

DMA Controller

The DMA Controller needs the usual circuits of an interface to communicate with the CPU and I/O device. In addition, it needs an address register, a word count register, and a set of address lines.



(Block diagram of DMA controller)

- The address ~~and~~ register and address lines are used for direct communication with the memory.
- The word count register specifies the number of words that must be transferred.
- The data transfer may be done directly between the device and memory under control of the DMA.

The DMA controller has three registers:

(1) Address register:- Contains an address to specify the desired location in memory.

(2) Word count register:- Holds the number of words to be transferred.

(3) Control register:- Specifies the mode of transfer.

The DMA is first initialized by the CPU. After that, the DMA starts and continues to transfer data between memory and peripheral unit until an entire block is transferred.
The CPU initializes the DMA by sending the following information through the data bus:

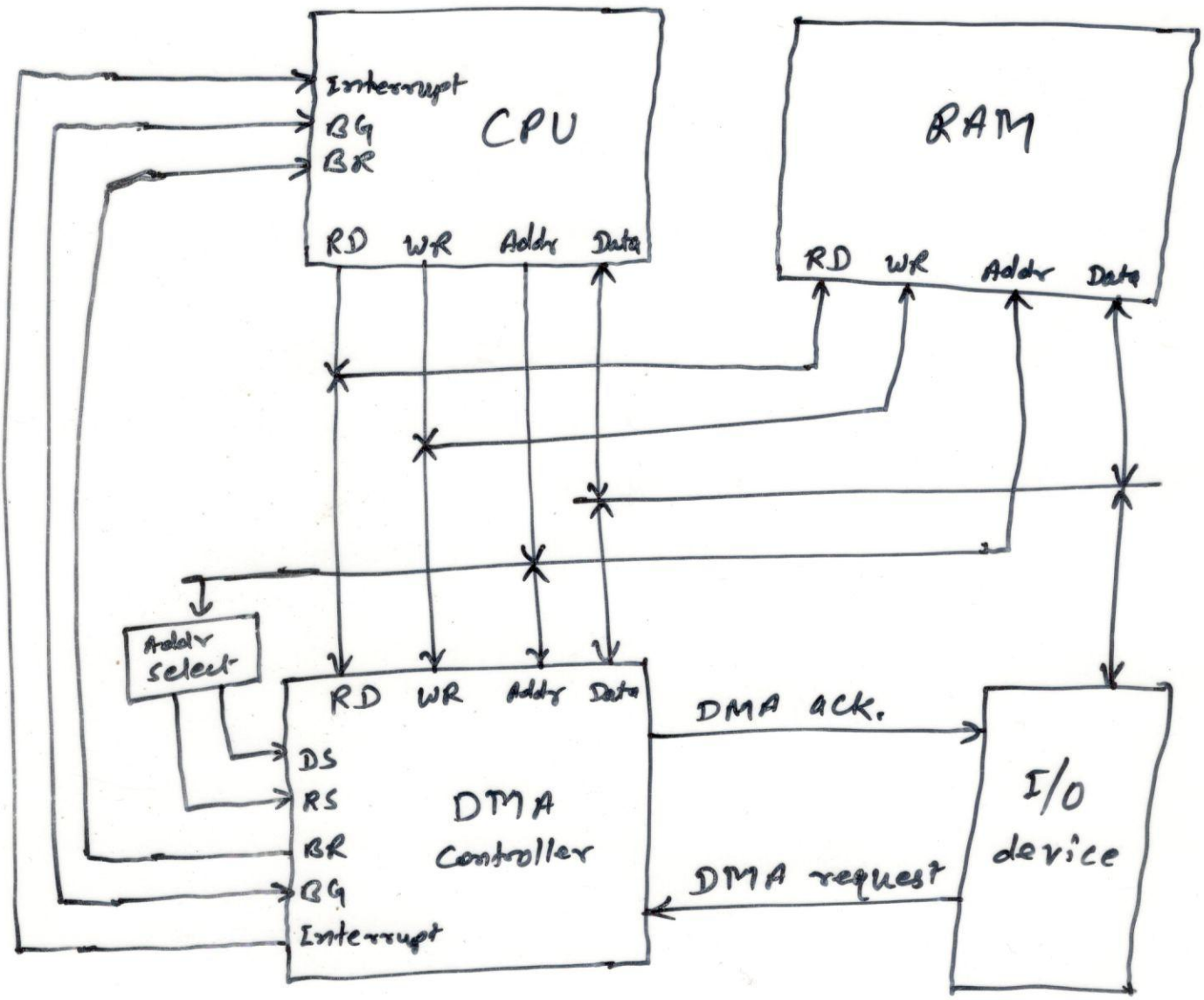
(1) The starting address of the memory block where data are available (to be stored).

(2) The word count, which is the number of words in the memory block.

(3) Control to specify the mode of transfer such as read or write.

(4) A control to start the DMA transfer.

DMA Transfer



The direction of transfer depends on the status of BG line. when BG=0, the RD and WR are input line allowing the CPU to comm. with the internall DMA registers. When BG=1, the RD and WR are output lines from DMA controller to RAM to specify the read or write operation for the data.