Chapter 11

COA

Peripheral

- I/O devices attached to the computer are called peripheral.
- Common peripheral are:

  keyboards, display unit, printers.
- That provides auxiliary storage are:

  Magnetic disk and tapes
- They are electromechanical and electromagnetic device of some complexity.

Input-Output Interface

- Method for transferring information between internal storage and external I/O device.
- For doing communication between CPU and peripheral communication link is required to resolve the difference between central computer and peripherals.
- Major difference are:

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

- To resolve the above issue interface unit is used.
- I/O bus consist of data line, address line and control line.
- Interface unit synchronize the data flow and supervise the transfer between processor and peripheral.
- Each peripheral have their own controller.
- I/O bus connect processor to each interface.
- Four types of command interface may received:

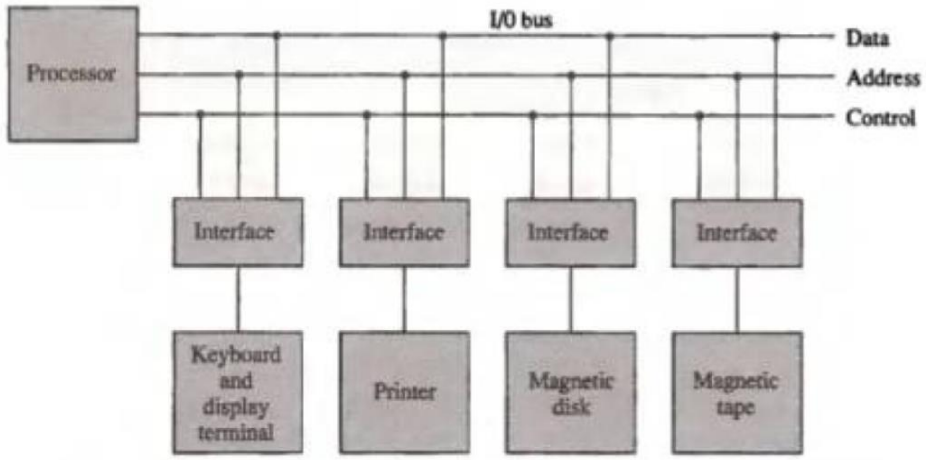
1.Control

2. Status,

3. Data Input and

4. Data Output

Figure 11-1 Connection of I/O bus to input-output devices.



I/O vs Memory Bus

- Processor communicate with memory unit
- Memory bus contains: data, address, read/write control line.
- Three ways to use the computer bus 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.

Isolated versus Memory mapped I/o

- Isolated I/O method: Configuration used to isolate all I/O interface addresses from the addresses assigned to memory.

  Separate set of read and write signal is used

- Memory- mapped: Used one set of read and write signals and do not distinguish between memory and I/O addresses.

  single set of read and write signal is used.

Asynchronous Data Transfer

- If CPU register and Interface register share a common clock then transfer between the two unit is said to be synchronous.
- If each unit use its own private clock for internal register and it is independent to each other then it is known as asynchronous data transfer.
- Data transfer between two unit is done with the help of control signal. Following are the ways of doing asynchronous Data transfer.
- Strobe : Strobe pulse supply by one unit is used to indicate other unit when the transfer has to occur.
- Handshaking: Data transfer accompany by control signal indicating presence of data in the bus. After receiving data unit send another control signal as an acknowledgement of received signal.

Strobe signal:

1. Source unit place the data on bus.
2. After setting down of data, activate the strobe signal.
3. Strobe signal inform the availability of data to destination unit.
4. After transmission, source remove the data from the bus.

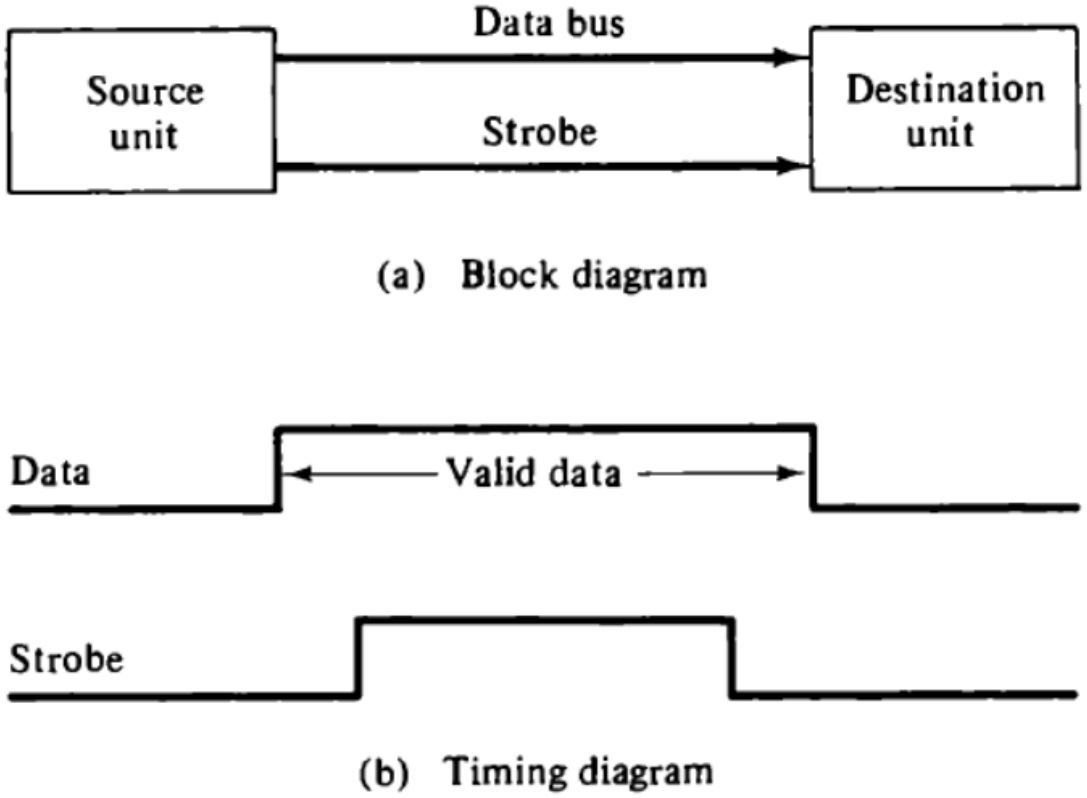


Figure 11-3 Source-initiated strobe for data transfer.

1. Destination unit activate the strobe pulse.
2. DU inform the source unit to transfer the data.
3. Source unit response by placing the data on bus
4. After transmission, DU disable the strobe pulse and SU remove the data from the bus.

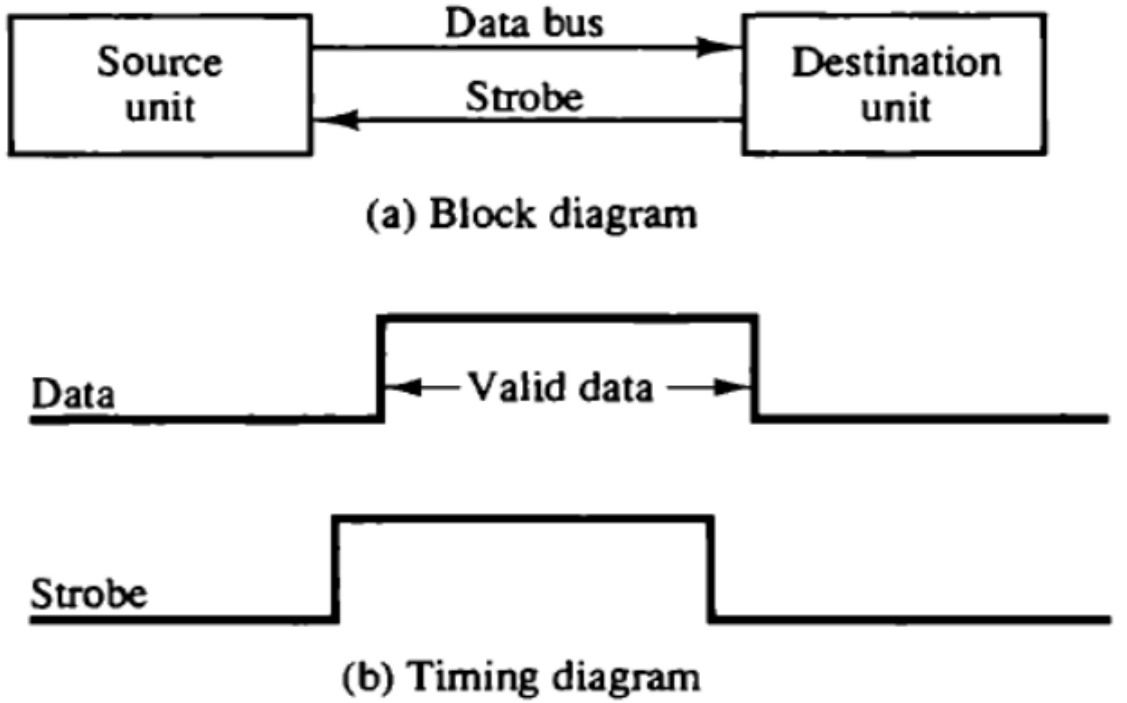


Figure 11-4 Destination-initiated strobe for data transfer.

Hand shaking

- Solve the problem of strobe signal.
- Use two wire control.
- One control line in the direction of source unit used to inform destination unit the availability of validate data in the bus.
- Second control unit in opposite direction use to inform source unit whether destination unit accept data.

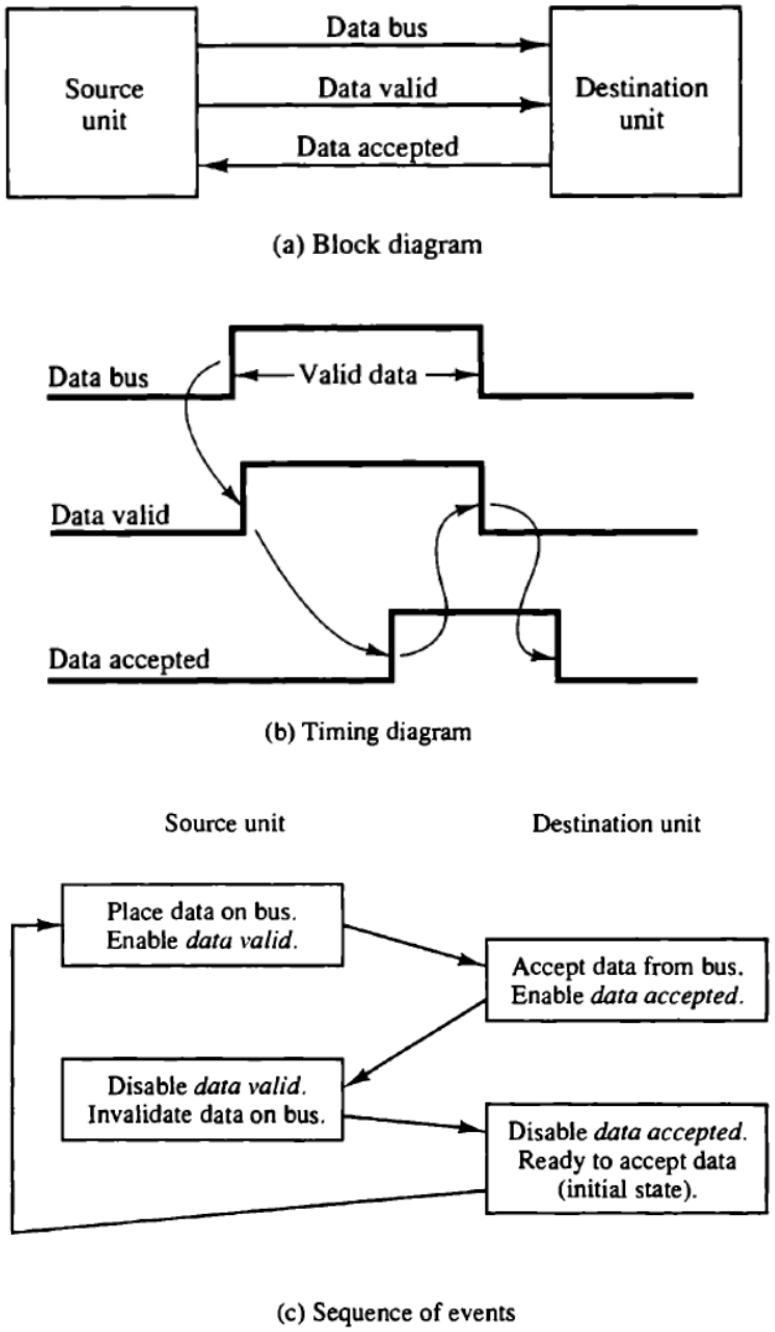


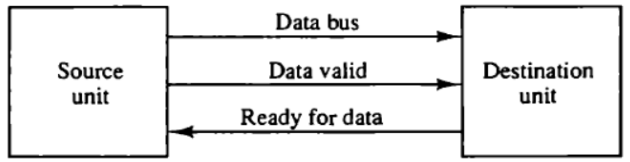
Figure 11-5 Source-initiated transfer using handshaking.

Hand shaking

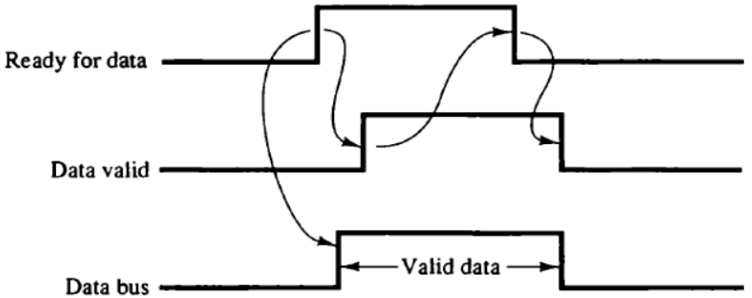
- Source unit placed data on bus only after receiving ‘ready for data’ signal from destination unit.
- After receiving the signal , source unit place the data on bus and also enable data valid control signal.
- After acceptance of data , DU disable the ready for data signal and SU disable Data valid signal.

- Timeout: Use for detecting error.

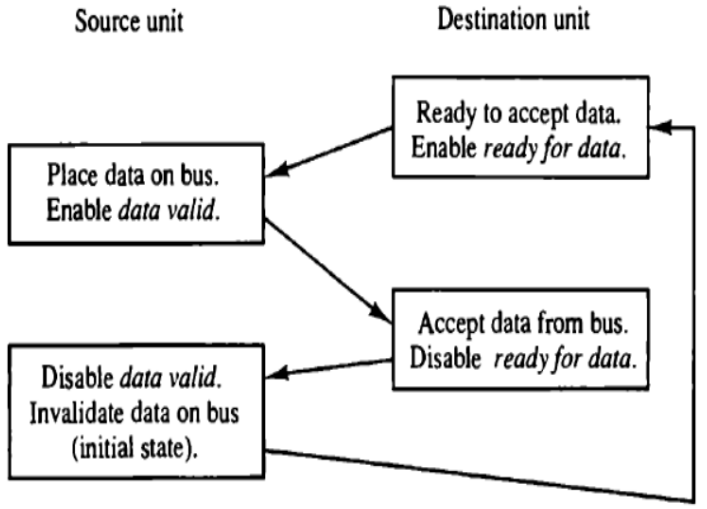
Figure 11-6 Destination-initiated transfer using handshaking.



(a) Block diagram



(b) Timing diagram



(c) Sequence of events

Mode of Transfer

- Data transfer to and from peripherals are handled in one of the 3 possible mode:

1. Programmed I/O: These operation are the result of I/O instruction written in the computer program. Transfer of data is initiated by an instruction in the program.
2. Interrupt initiated I/O: Transfer of data is done after checking the status of interrupt request signal.
3. Direct memory access (DMA): Interface transfers data into and out of the memory unit through memory bus.

Priority Interrupt

- Used to establish priority over various sources to determine which condition is to be serviced first when more then one request arrive simultaneously.
- Establishing the Priority of simultaneous interrupts can be done by software or hardware.
- Polling: Used to identify the highest priority source by mean of software.

Daisy Chaining priority

- The hardware priority function can be established by either a serial or a parallel connection of interrupt line.
- Daisy chaining method of establishing priority consist of serial connection of all device that request an interrupt.
- Devices with the highest priority is placed in the first position followed by lower priorities.

Daisy Chaining Priority

- Interrupt request line is common to all device forms a wired logic connection.
- Interrupt line enable the interrupt input when goes to low level state and remain at high level when no interrupts are pending. It is equivalent to negative logic OR operation.
- CPU response to interrupt request by enabling the interrupt acknowledge line.
- Device 1 received the signal at its PI (priority input) and passed this acknowledge signal to next device through PO(priority out) only if device 1 not requesting interrupt.
- PO=0 if device 1 has a pending interrupt and insert its own interrupt vector address (VAD) into the data bus for CPU to use during the interrupt cycle.

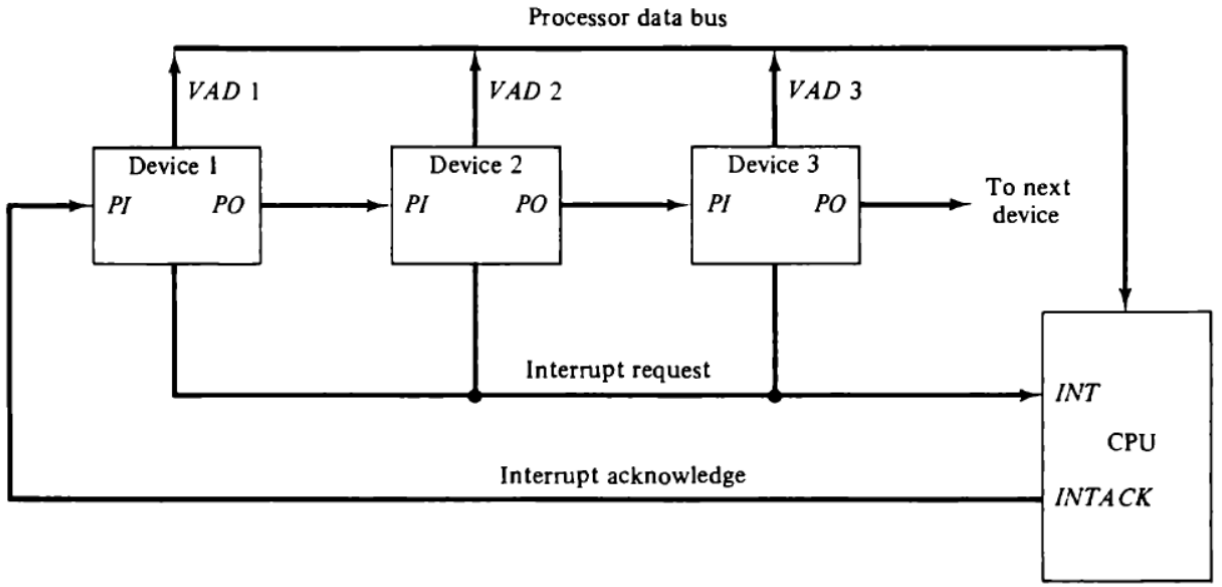


Figure 11-12 Daisy-chain priority interrupt.

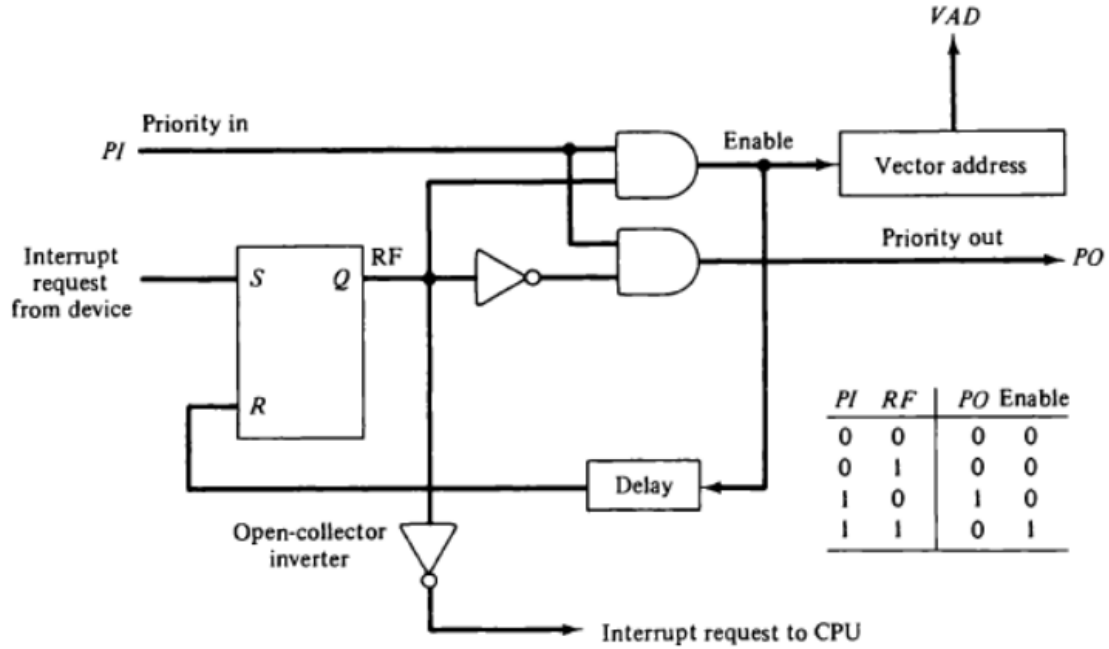


Figure 11-13 One stage of the daisy-chain priority arrangement.

Parallel Priority Interrupt

- Use a register whose bits are set separately by interrupt signal from each device.
- Priority establish according to the bit in register.
- Mask register use to control the status of the interrupt signal.

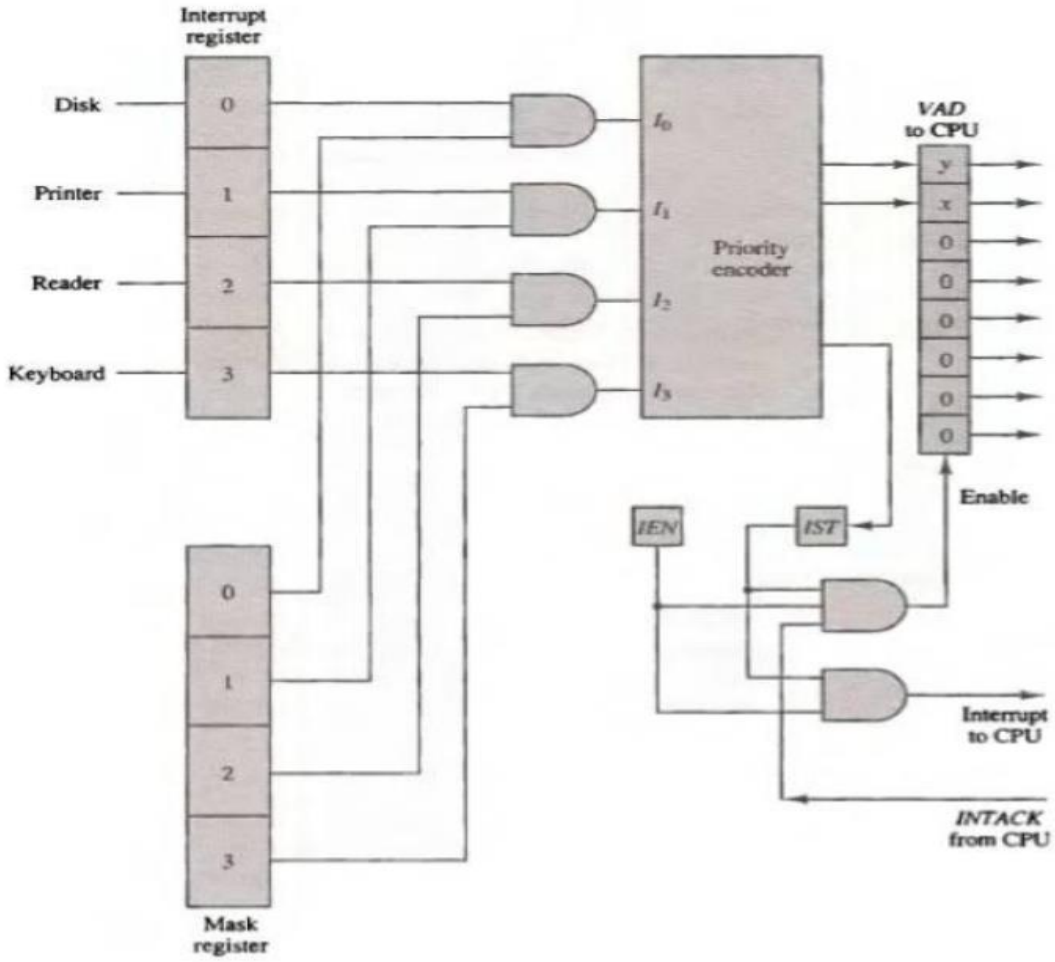


Figure 11-14 Priority interrupt hardware.

Direct Memory Access

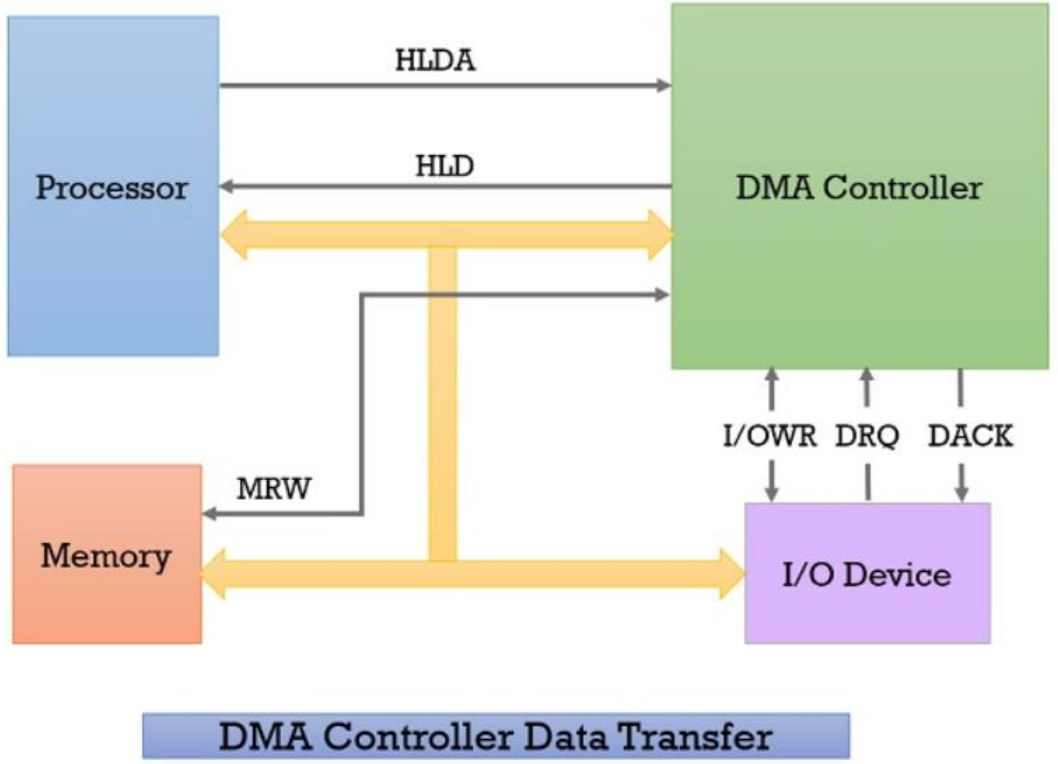
- The term DMA stands for direct memory access. The hardware device used for direct memory access is called the DMA controller. DMA controller is a control unit, part of I/O device's interface circuit, which can transfer blocks of data between I/O devices and main memory with minimal intervention from the processor.

The DMA transfers the data in three modes which include the following.

- a) **Burst Mode**: In this mode DMA handover the buses to CPU only after completion of whole data transfer. Meanwhile, if the CPU requires the bus it has to stay ideal and wait for data transfer.
- b) **Cycle Stealing Mode**: In this mode, DMA gives control of buses to CPU after transfer of every byte. It continuously issues a request for bus control, makes the transfer of one byte and returns the bus. By this CPU doesn't have to wait for a long time if it needs a bus for higher priority task.
- c) **Transparent Mode:** Here, DMA transfers data only when CPU is executing the instruction which does not require the use of buses.

Working Of DMA

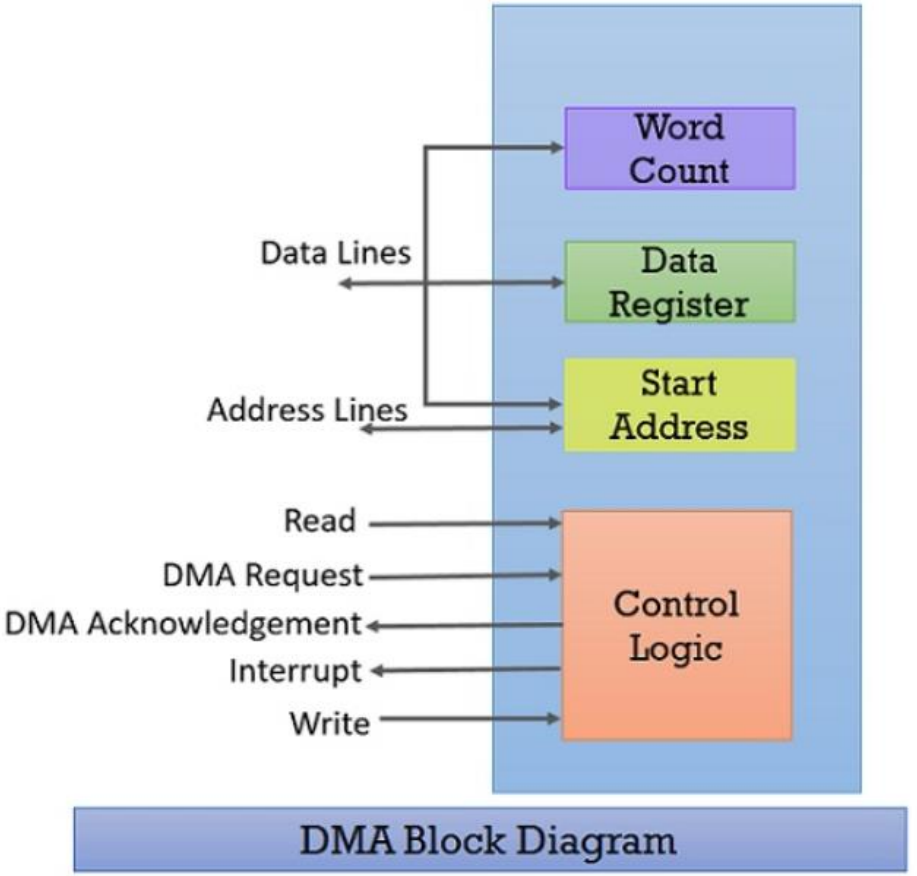
1. Whenever an I/O device wants to transfer the data to or from memory, it sends the DMA request (**DRQ**) to the DMA controller. DMA controller accepts this DRQ and asks the CPU to hold for a few clock cycles by sending it the Hold request (**HLD**).
2. CPU receives the Hold request (HLD) from DMA controller and relinquishes the bus and sends the Hold acknowledgement (**HLDA**) to DMA controller.
3. After receiving the Hold acknowledgement (HLDA), DMA controller acknowledges I/O device **(DACK)** that the data transfer can be performed and DMA controller takes the charge of the system bus and transfers the data to or from memory.
4. When the data transfer is accomplished, the DMA raise an **interrupt** to let know the processor that the task of data transfer is finished and the processor can take control over the bus again and start processing where it has left.



DMA

Whenever a processor is requested to read or write a block of data, i.e. transfer a block of data, it instructs the DMA controller by sending the following information.

1. The first information is whether the data has to be read from memory or the data has to be written to the memory. It passes this information via **read or write control lines** that is between the processor and DMA controllers **control logic unit**.
2. The processor also provides the **starting address** of/ for the data block in the memory, from where the data block in memory has to be read or where the data block has to be written in memory. DMA controller stores this in its **address register**. It is also called the **starting address register**.
3. The processor also sends the **word count**, i.e. how many words are to be read or written. It stores this information in the **data count** or the **word count** register.
4. The most important is the **address of I/O device** that wants to read or write data. This information is stored in the **data register.**



Advantage and Disadvantage of DMA

- Advantages:

1. Transferring the data without the involvement of the processor will **speed up** the read-write task.
2. DMA **reduces the clock cycle** requires to read or write a block of data.
3. Implementing DMA also **reduces the overhead** of the processor.

- Disadvantages

1. As it is a hardware unit, it would **cost** to implement a DMA controller in the system.
2. Cache **coherence** problem can occur while using DMA controller.