3) DMA BASED I/O

DMA means **transferring data directly between memory and I/O.**
DMA transfers are **very fast** as compared to Processor based transfers due to two reasons.
1. They are **hardware based** so no time is wasted in fetching and decoding instructions.
2. Transfers are **directly between memory and I/O** without data going via the Processor.

To Perform a DMA transfer we need a **DMA Controller like 8237/ 8257.**
It is capable of taking control of the buses from the Processor.
The process is performed as follows.

1) By Default **Processor is the bus master**.
2) The DMA transfer parameters first initialized by the processor.
3) **Processor programs two registers inside the DMAC called CAR and CWCR** giving the starting address and the number of bytes to be transferred.
4) DMAC now ensures that the I/O device is ready for the transfer by checking the DREQ signal.
5) **If DREQ=1, then DMAC gives HOLD signal** to the Processor requesting control of the system bus.
6) **Processor releases control of the bus** after finishing the current machine (bus) cycle.
7) Processor **gives HLDA** informing DMAC that it is now the bus master.
8) **DMAC issues DACK#** (by default active low, but can be changed) to I/O device indicating that the transfer is about to begin.
9) Now DMAC **transfers one byte in one cycle**.
10) After every byte is transferred the **Address register and Count register are decremented by 1**.
11) This repeats till Count reaches **"0"** also called **Terminal Count**.
12) Now the **transfer is complete**.
13) DMAC **returns the system bus to Processor by making HOLD = 0**.
14) Processor once again **becomes bus master**.

Advantage of DMA
DMA transfers are very fast.

Drawback of DMA
DMAC becomes the bus master. Hence during DMA cycles, the processor cannot perform any operations as the bus is already being used for DMA. The processor remains in HOLD state.

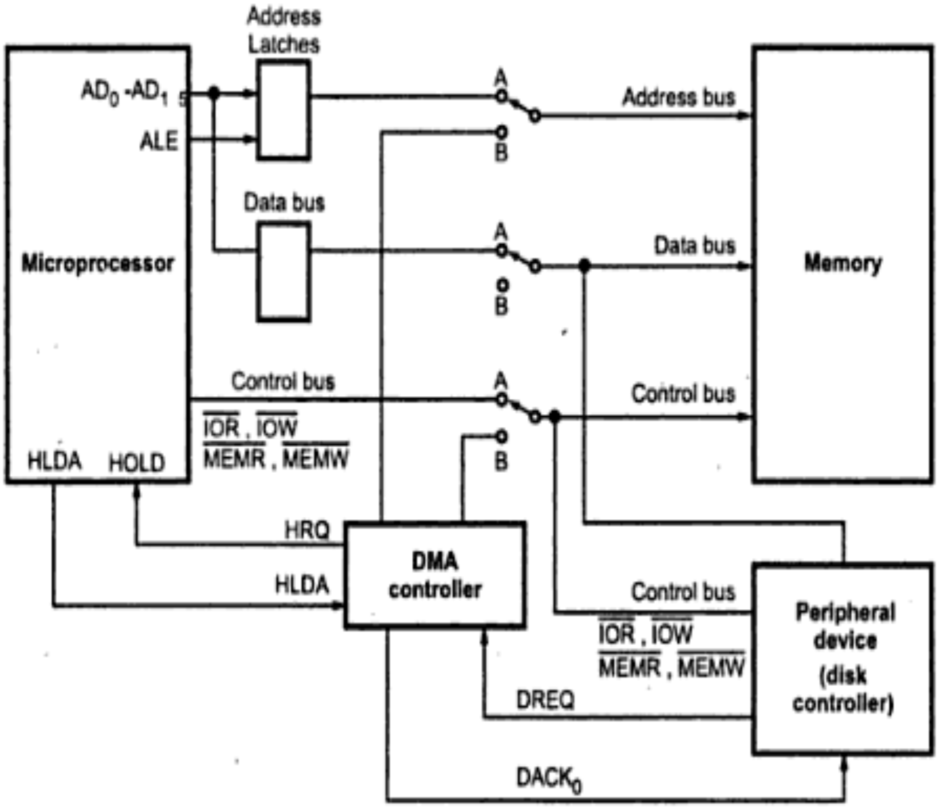
Difference between Interrupt Request and DMA request

When an interrupt occurs, the processor has to suspend the current program, execute the ISR and then **return to the next instruction** of the main program. Hence it is necessary that the **processor completes the current instruction** before servicing an interrupt request.

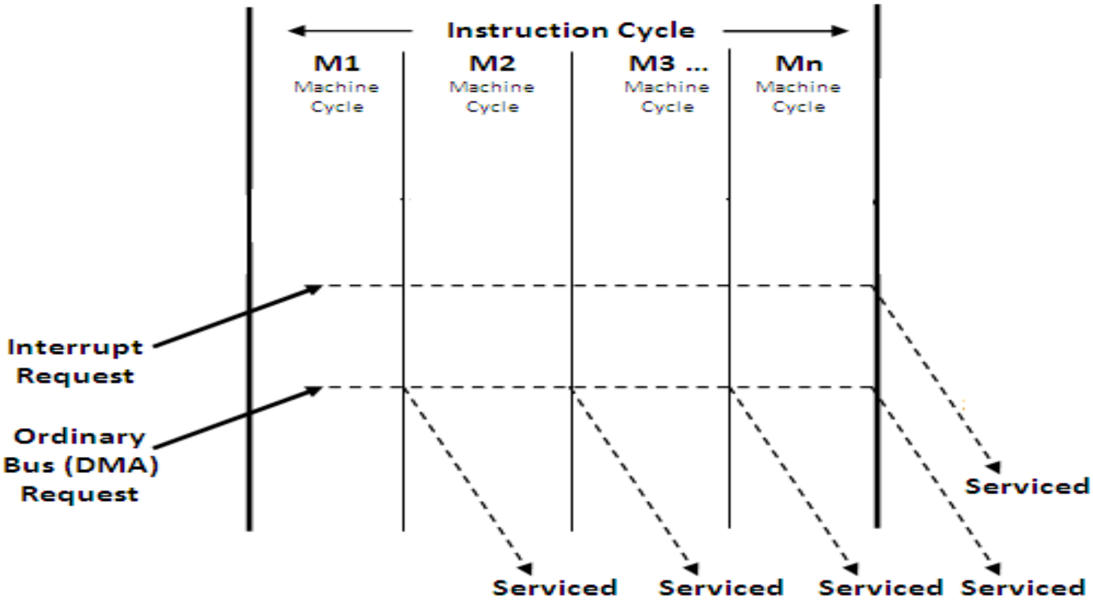
When a DMA request occurs, the processor has to simply relinquish (give away) control of the system bus and enter hold state. When ever it gets back the bus it can **resume from where it had left**. Hence the processor **need not finish the current instruction** before servicing a DMA request.
It simply has to **finish the current machine cycle**.

Hence Instruction cycles are Interrupt Breakpoints and Machine cycles are DMA breakpoints.

DMA Operation



Difference between Interrupt Request and DMA Request



TYPES / METHODS / TECHNIQUES OF DMA TRANSFERS

8237 has **four modes of data transfer**:

1) BLOCK TRANSFER MODE / BURST MODE.

In this mode, the DMAC is programmed to **transfer ALL THE BYTES** in one complete DMA operation.
After a byte is transferred, the **CAR and CWCR are adjusted** accordingly.
The **system bus** is **returned** to the **processor, ONLY after all the bytes are transferred.**
It is the **fastest** form of DMA but keeps the **processor inactive** for a long time.

2) SINGLE BYTE TRANSFER MODE/ CYCLE STEALING.

Once the DMAC becomes the bus master, it will transfer only **ONE BYTE** and return the bus to the processor. As soon as the processor performs one bus cycle, DMAC will once again take the bus back from the processor.
Hence both **DMAC and processor** are **constantly stealing bus cycles** from each other.
It is the **most popular** method of DMA, because it keeps the **processor active in the background**.
After a byte is transferred, the **CAR** and **CWCR** are **adjusted** accordingly.

3) DEMAND TRANSFER MODE.

It is very **similar to Block Transfer**, **except** that the **DREQ must remain active throughout the DMA operation**.
If during the operation **DREQ goes low**, the **DMA operation is stopped** and the **busses** are **returned** to the **processor.** *#Please refer Bharat Sir's Lecture Notes for this ...*
In the meantime, the **processor** can **continue** with its own operations. **Once DREQ goes high again**, the **DMA operation continues** from where it had stopped.
This means, the transfer happens on demand from the I/O device, hence the name.

4) HIDDEN MODE / TRANSPARENT MODE.

In this mode, **once the processor programs all parameters inside the DMAC, the DMAC** does not request the processor for the control of the bus. Instead it observes the processor. **It waits for the processor to enter idle state**. Once the processor is idle, the DMAC will take control of the bus and perform the Transfer. So the Transfer is **totally transparent to the processor** or hidden from the processor. Hence the name.