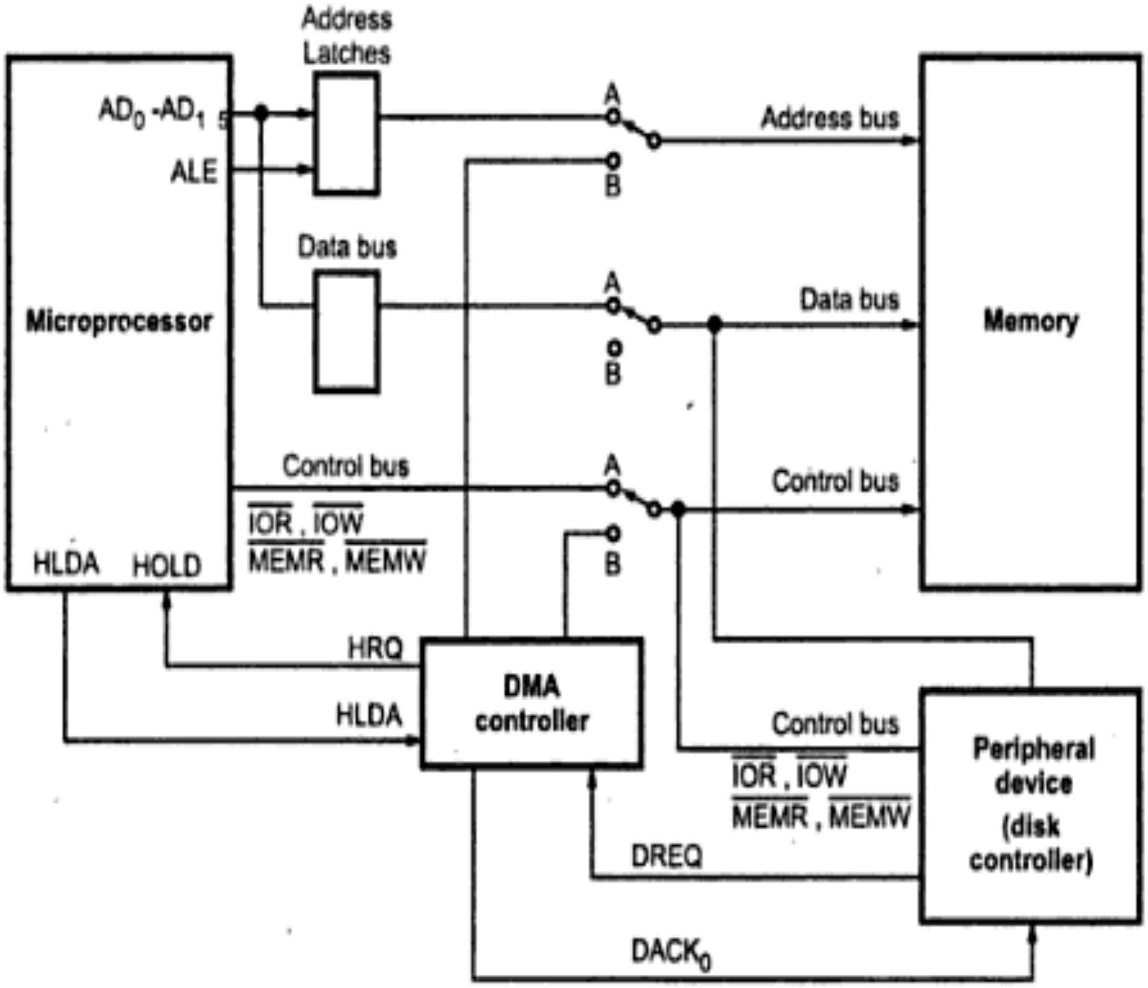
CONCEPT OF DMA



DMA Transfer is a **hardware controlled I/O** Transfer technique.
It is mainly used for **high-speed data transfer between I/O and Memory** where the speed of the peripheral is generally faster than the µP. **For doubts contact Bharat Sir on 98204 08217**
In Program Controlled I/O, Status or interrupt driven I/O the speed of transfer is **slow** mainly **because instructions** need to be **decoded** and **then executed** for the transfer.
DMA transfer is **software independent** and **hence much faster**.
A device known as the DMA Controller (DMAC) is responsible for the DMA transfer.

The **sequence of DMA transfer** is as follows:

1) µP initializes the DMAC by giving the starting address and the number of bytes to be transferred.
2) An **I/O device requests** the **DMAC**, to perform DMA transfer, **through the DREQ line**.
3) The **DMAC** in turn sends a **request signal to** the **µP**, through the **HOLD line**.
4) The **µP finishes the current machine cycle** and **releases** the **system bus** (gets disconnected from it).
   It also **acknowledges** receiving the HOLD signal through the **HLDA line**.
5) The **DMAC acquires control of the system bus.**
   The **DMAC sends** the **DACK signal** to the I/O peripheral and the **DMA transfer begins**.
6) After every byte is transferred, the Address Reg is incremented (or decremented) and the Count Reg is decremented.
7) This continues till the Count reaches zero (Terminal Count). Now the DMA transfer is completed.
8) **At the end** of the transfer, **the system bus is released by the DMAC** by making HOLD = 0.
   Thus **µP takes control of the system bus** and continues its operation.

The DMA Controller (DMAC) does DMA transfer through its channels.
The minimum requirements of each channel are:

i. **Address Register** (to store the starting address for the transfer).
ii. **Count Register** (to store the number of bytes to be transferred).
iii. **A DREQ signal** from the IO device**.**
iv. **A DACK signal** to the IO device.

Operation Cycles of DMAC

There are mainly two operation cycles of a DMAC.

i.Idle Cycle

After Reset, the DMAC is in idle state (idle cycle).
During idle state, **no DMA operation is taking place**.
No DMA requests are **active**.
The **initialization** of the DMAC takes place in the idle mode.

ii. Active Cycle

Once **DMA operation begins,** the **DMAC** is said to be **in active mode**.
Now the **DMAC controls** the **system bus**.
There are **three types of ACTIVE DMA Cycles** while performing DMA transfer:

1) DMA Read

The DMAC **reads** data **from** the **memory** and **writes into** to the **I/O device**.
Thus, **MEMR** and **IOW** signals are used.

2) DMA Write

The DMAC **reads** data **from** the **I/O device** and **writes into** to the **memory**.
Thus, **IOR** and **MEMW** signals are used.

3) DMA Verify

In this cycle, 8237 does not generate any control signals.
Hence, no data transfer takes place.
During this time, the peripheral and the DMAC verify the correctness of the data transferred, using some error detection method.

Transfer Modes of 8237

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

1)Single Byte Transfer Mode/ Cycle Stealing.

Once the DMAC becomes the bus master, it will transfer only **ONE BYTE** and return the bus back to the microprocessor. As soon as the microprocessor performs one bus cycle, DMAC will once again take the bus back from the microprocessor.
Hence both **DMAC and microprocessor** are **constantly stealing bus cycles** from each other.
It is the **most popular** method of DMA, because it keeps the **microprocessor active in the background**.
After a byte is transferred, the **CAR** and **CWCR** are **adjusted** accordingly.
The **system bus is returned** to the **µP.**
For further bytes to be transferred, **the DREQ line must go active again**, and then the entire operation is repeated.

2) Block Transfer Mode.

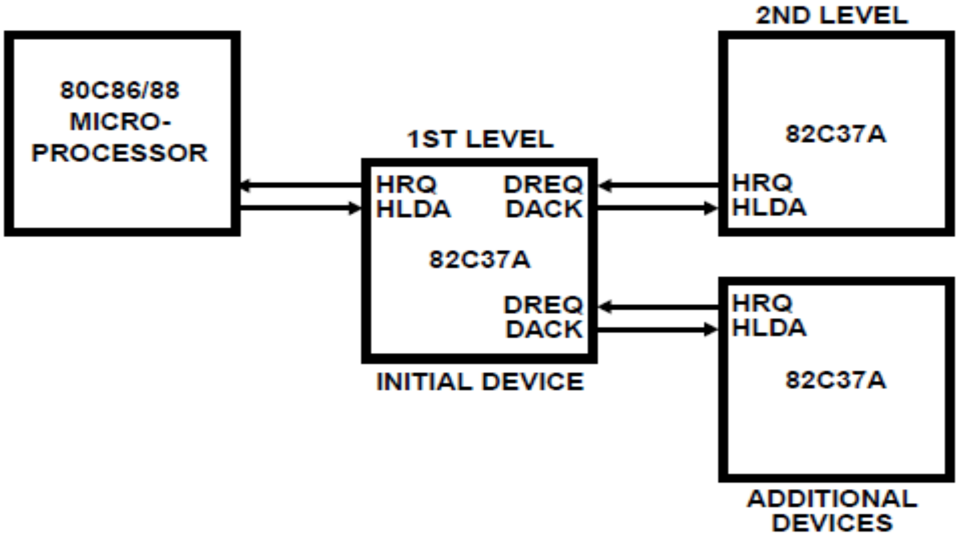
In this mode, the DMAC is programmed to **transfer ALL THE BYTES** in one complete DMA operation.
After a byte is transferred, the **CAR and CWCR are adjusted** accordingly.
The **system bus** is **returned** to the **µP, ONLY after all the bytes are transferred.** I.e. **TC** is **reached or EOP** signal is issued.
It is the **fastest** form of DMA but keeps the **microprocessor inactive** for a long time.
The **DREQ** signal **needs to be active only in the beginning** for requesting the DMA service initially.
Thereafter **DREQ can become low during the transfer**.

3) Demand Transfer Mode.

It is very **similar to Block Transfer**, **except** that the **DREQ must active throughout the DMA operation**.
If during the operation **DREQ goes low**, the **DMA operation is stopped** and the **busses** are **returned** to the **µP.** *#Please refer Bharat Sir's Lecture Notes for this ...*
In the meantime, the **µP** can **continue** with its own operations. **Once DREQ goes high again**, the **DMA operation continues** from where it had stopped.

4) Cascade Transfer Mode. **Specific for 8237**

In this mode, **more than one DMACs** are cascaded together.
It is used to **increase** the **number of devices interfaced** to the **µP**.
Here we have **one Master DMAC**, to which **one or more Slave DMACs** are connected. The **Slave gives HRQ** to the Master **on the DREQ of the Master**, and the **Master gives HRQ** to the **µP on the HOLD of the µP.**


80C86/88
MICRO-
PROCESSOR
1ST LEVEL
HRQ
HLDA
DREQ
DACK
82C37A
DREQ
DACK
INITIAL DEVICE
2ND LEVEL
82C37A
HRQ
HLDA
HRQ
HLDA
82C37A
ADDITIONAL
DEVICES