What is DMA Controller?

1. DMA stands for Direct Memory Access. It is designed by Intel to transfer data at the fastest rate.

2. It allows the device to transfer the data directly to/from memory without any interference of the CPU.

3. DMA can directly access memory and used to transfer data from one memory location to another or from an I/O device to memory and vice-versa.

❖ Why DMA Controller is needed?

1. Usually, processors control all the process of transferring data, right from initiating the transfer to the storage of data at the destination.

2. To speed up the transfer of data between I/O devices and memory, DMA controller acts as station master.

3. DMA controller transfers data with minimal intervention of the processor.

How DMA Controller works?

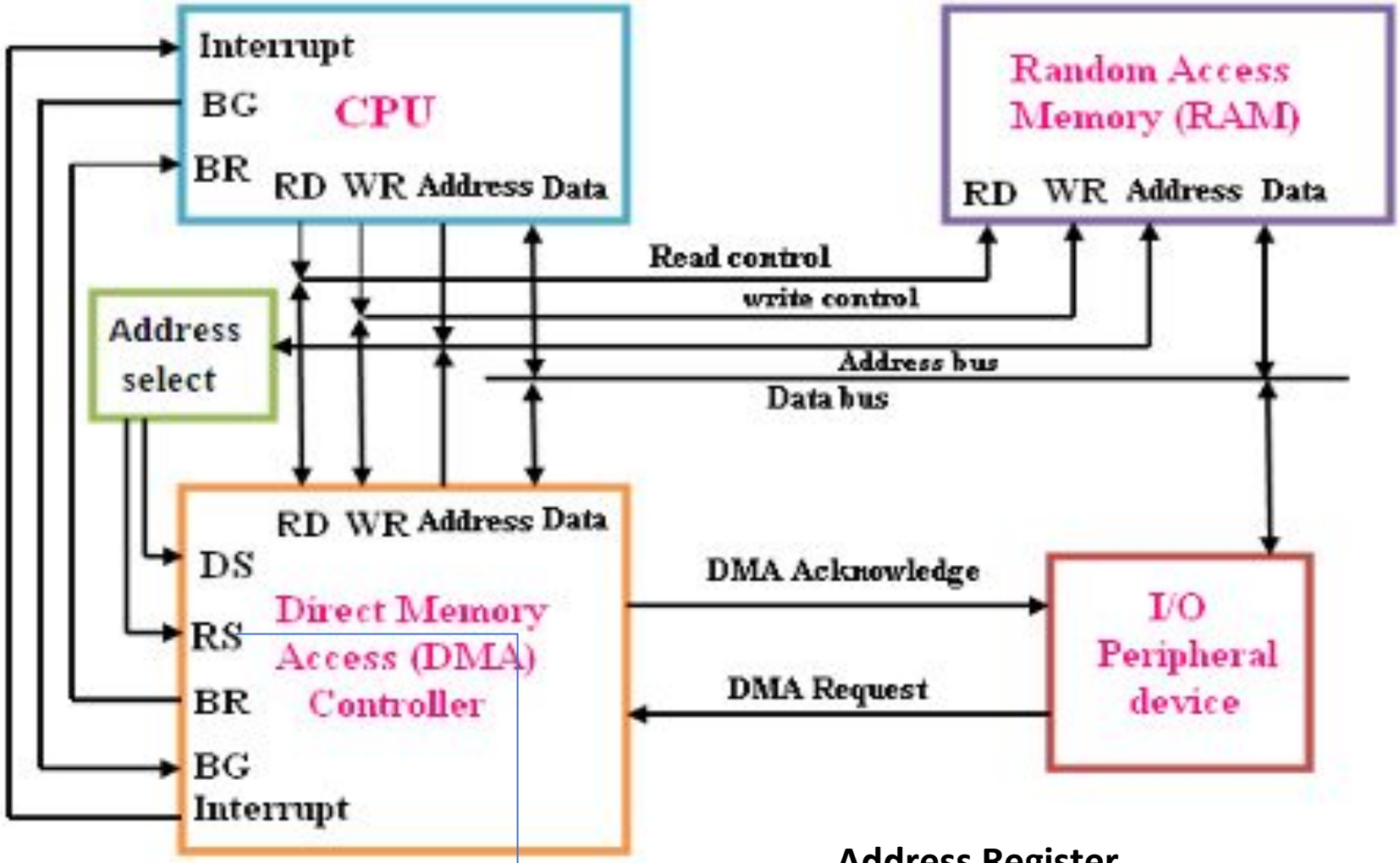
1. Initially, when any device has to send data between the device and the memory, the device has to send DMA request (DRQ) to DMA controller.

2. The DMA controller sends Hold request (HRQ) to the CPU and waits for the CPU to assert the HLDA.

3. Then the microprocessor tri-states all the data bus, address bus, and control bus. The CPU leaves the control over bus and acknowledges the HOLD request through HLDA signal.

4. Now the CPU is in HOLD state and the DMA controller has to manage the operations over buses between the CPU, memory, and I/O devices.

How Data are transferred in using DMA?



Address Register
Word Count Register
Control Register

How Data are transferred in DMA?

- Although it transfers data without intervention of processor, it is controlled by the processor.
- If the DMA controller is free, it requests the control of bus from the processor by raising the bus request signal.
- Processor grants the bus to the controller by raising the bus grant signal, now DMA controller is the bus master.
- The processor initiates the DMA controller by sending the memory addresses, number of blocks of data to be transferred and direction of data transfer.
- After assigning the data transfer task to the DMA controller, instead of waiting ideally till completion of data transfer, the processor resumes the execution of the program after retrieving instructions from the stack.
- The processor initiates the DMA controller by sending the starting address, Number of words in the data block and direction of transfer of data .i.e. from I/O devices to the memory or from main memory to I/O devices.
- It also contains the control unit and data count for keeping counts of the number of blocks transferred and indicating the direction of transfer of data.
- When the transfer is completed, DMA informs the processor by raising an interrupt.

Data transfer mode of DMA Controller

❖ The DMA transfers the data in three modes which includes the following:

1. **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.

2. **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.

3. **Transparent Mode:** Here, DMA transfers data only when CPU is executing the instruction which does not require the use of buses.

Advantages & Disadvantages of DMA Controller

- **Advantages**

1. DMA speedups the memory operations by bypassing the involvement of the CPU.

2. The work overload on the CPU decreases.

3. Large number of data transferred is the real advantage of using DMA controller.

- **Disadvantages**

1. Cache coherence problem can be seen when DMA is used for data transfer.

2. Increases the price of the system.

8257 DMA Controller

❖ **Features of 8257 DMA Controller:**

1. It has four channels which can be used over four I/O devices.
2. Each channel has 16-bit address and 14-bit counter.
3. Each channel can transfer data up to 64kb.
4. Each channel can be programmed independently.
5. Each channel can perform read transfer, write transfer and verify transfer operations.
6. It generates MARK signal to the peripheral device that 128 bytes have been transferred.
7. Its frequency ranges from 250Hz to 3MHz.
8. It operates in 2 modes, i.e., Master mode (Control of the buses are given to DMA Controller) and Slave mode (Processor keeps the control of the buses and send data to the DMA controller to complete the data transfer operation).

8257 DMA Controller

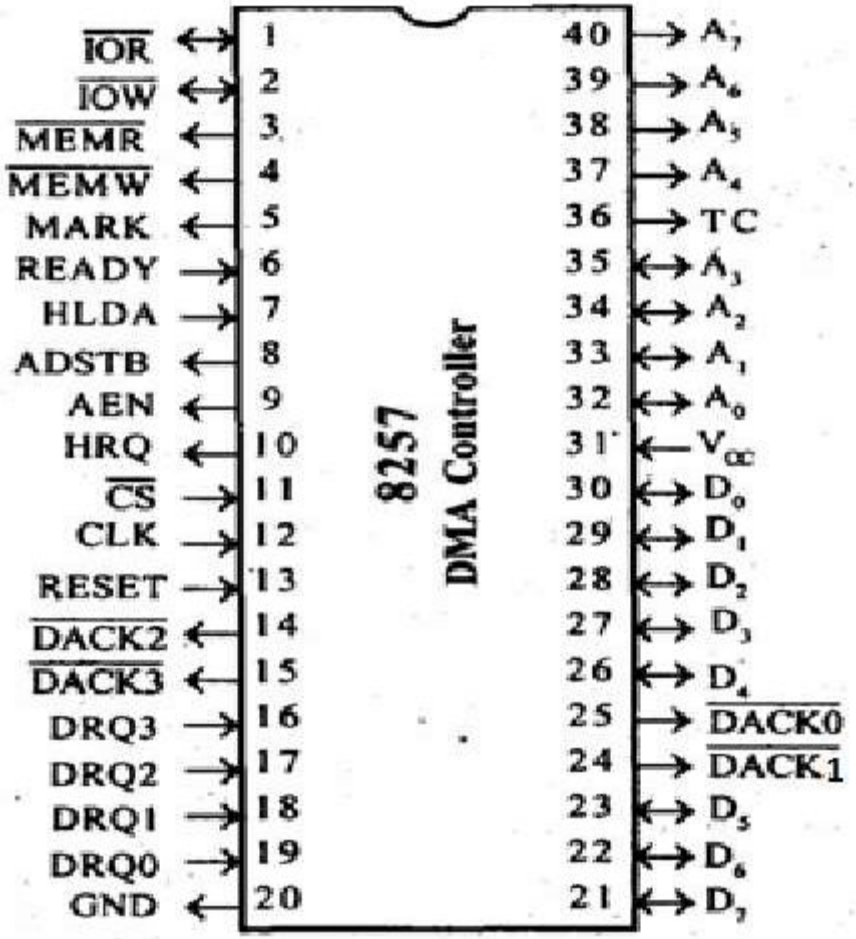
- **Pin Description of 8257 DMA Controller:**

- **DRQ_0–DRQ3**

These are the four individual channel DMA request inputs, which are used by the peripheral devices for using DMA services. When the fixed priority mode is selected, then DRQ_0 has the highest priority and DRQ_3 has the lowest priority among them.

- **$DACK_o$ – $DACK_3$**

These are the active-low DMA acknowledge lines, which updates the requesting peripheral about the status of their request by the CPU. These lines can also act as strobe lines for the requesting devices.



8257 DMA Controller

- **IOR**

It is an active-low bidirectional tri-state input line, which is used by the CPU to read internal registers of 8257 in the Slave mode. In the master mode, it is used to read data from the peripheral devices during a memory write cycle.

- **D_o – D_7**

These are bidirectional, data lines which are used to interface the system bus with the internal data bus of DMA controller. In the Slave mode, it carries command words to 8257 and status word from 8257. In the master mode, these lines are used to send higher byte of the generated address to the latch. This address is further latched using ADSTB signal.

8257 DMA Controller

- **IOW**

It is an active low bi-direction tri-state line, which is used to load the contents of the data bus to the 8-bit mode register or upper/lower byte of a 16-bit DMA address register or terminal count register. In the master mode, it is used to load the data to the peripheral devices during DMA memory read cycle.

- **CLK**

It is used to generate timing signals which controls the operations of 8257.

- **RESET**

This signal is used to RESET the DMA controller by disabling all the DMA channels. Used to clear mode set registers and status registers, command register and all requests.

- **A_o - A_3**

These are the four least significant address lines. In the slave mode, they act as an input, which selects one of the registers to be read or written. In the master mode, they are the four least significant memory address output lines generated by 8257.

8257 DMA Controller

- **CS**

It is an active-low chip select line. In the Slave mode, it enables the read/write operations to/from 8257. In the master mode, it disables the read/write operations to/from 8257.

- **A_4 - A_7**

These are the higher nibble of the lower byte address generated by DMA in the master mode.

- **READY**

It is an active-high asynchronous input signal, which makes DMA ready.

- **HRQ**

This signal is used to receive the hold request signal from the output device. In the slave mode, it is connected with a DRQ input line 8257. In Master mode, it is connected with HOLD input of the CPU.

8257 DMA Controller

- **HLDA**

It is the hold acknowledgement signal which indicates the DMA controller that the bus has been granted to the requesting peripheral by the CPU when it is set to 1.

- **MEMR**

It is the low memory read signal, which is used to read the data from the addressed memory locations during DMA read cycles.

- **MEMW**

It is the active-low three state signal which is used to write the data to the addressed memory location during DMA write operation.

- **ADST**

This signal is used to convert the higher byte of the memory address generated by the DMA controller into the latches.

8257 DMA Controller

- **AEN**

This signal is used to disable the address bus/data bus.

- **TC**

It stands for ‘Terminal Count’ and active high signal, which indicates that the count value has reached to zero.

- **MARK**

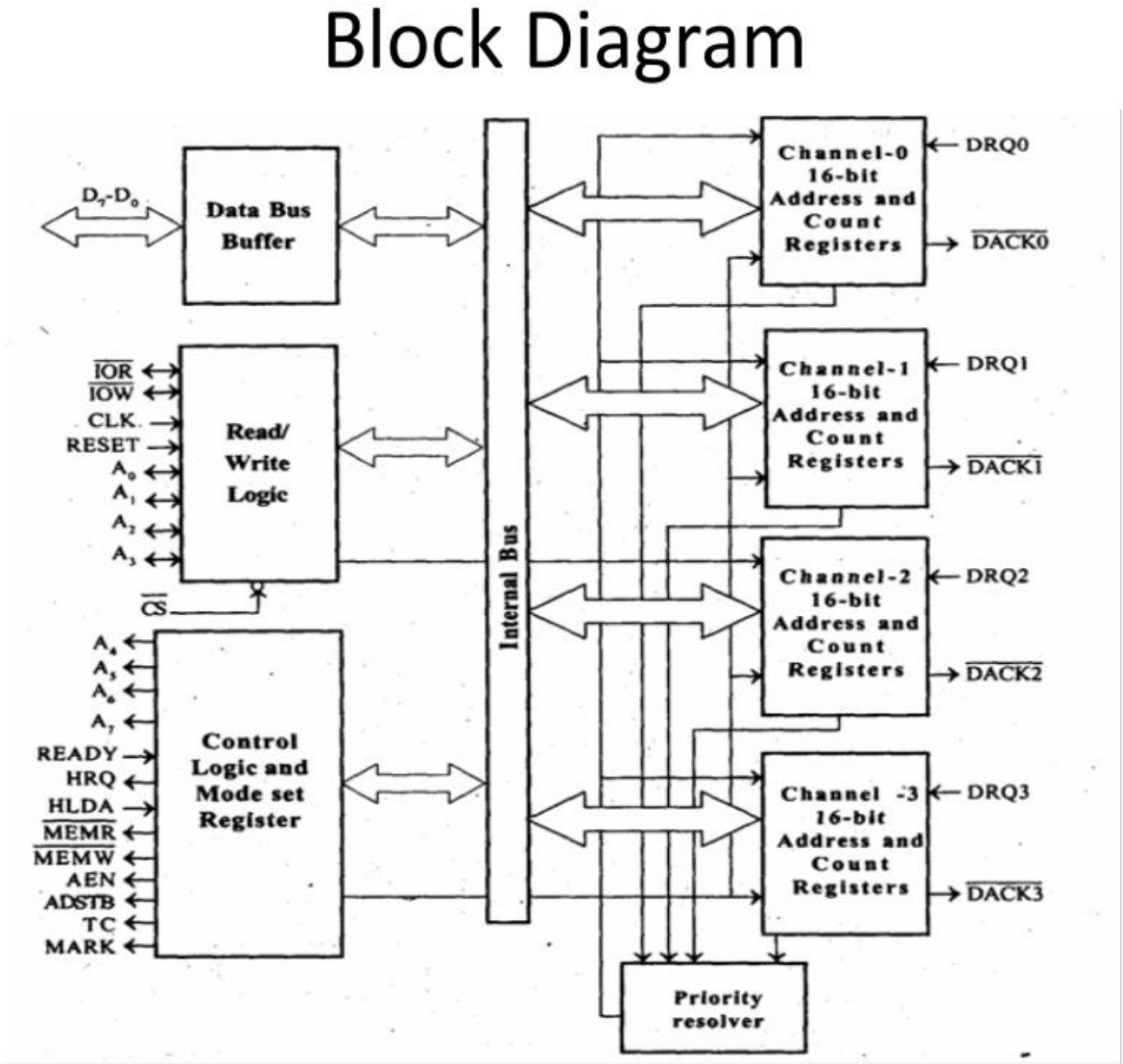
It will be activated when the 128 bytes of data has been transferred by the DMA. It also indicates that current DMA cycle is the 128^{th} cycle since the previous MARK cycle signal to the selected peripheral device.

- **V_{cc}**

It is the power signal which is required for the operation of the circuit.

Internal Architecture of 8257 DMA Controller

1. DMA channels
2. Data bus buffer
3. Read/Write logic
4. Control logic
5. Mode set Register
6. Status Register



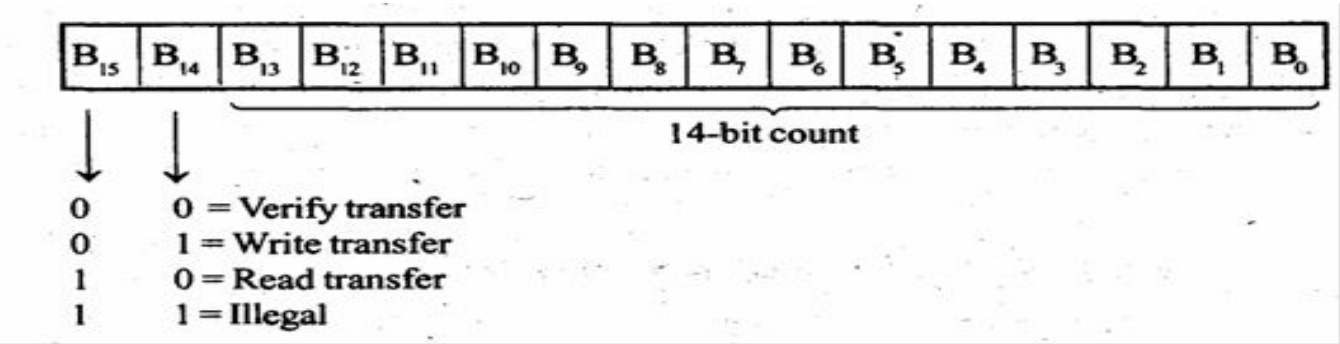
Operational Modes of 8257 DMA Controller

- 8257 DMA Controller has **four channels. This are channel 0, channel 1, channel 2, channel 3**.

- Each channel has **two programmable 16-bit registers.**

1. **Address Register:** It holds the addresses of the memory location and the I/O devices which want to communicate with it. It holds the starting address of the memory location for data transfer. The address in the address register is automatically incremented after each read/write/verify transfer operation.

2. **Counter Register:** Used to count the number of bytes or words are transferred by DMA controller. The format of Count register is,

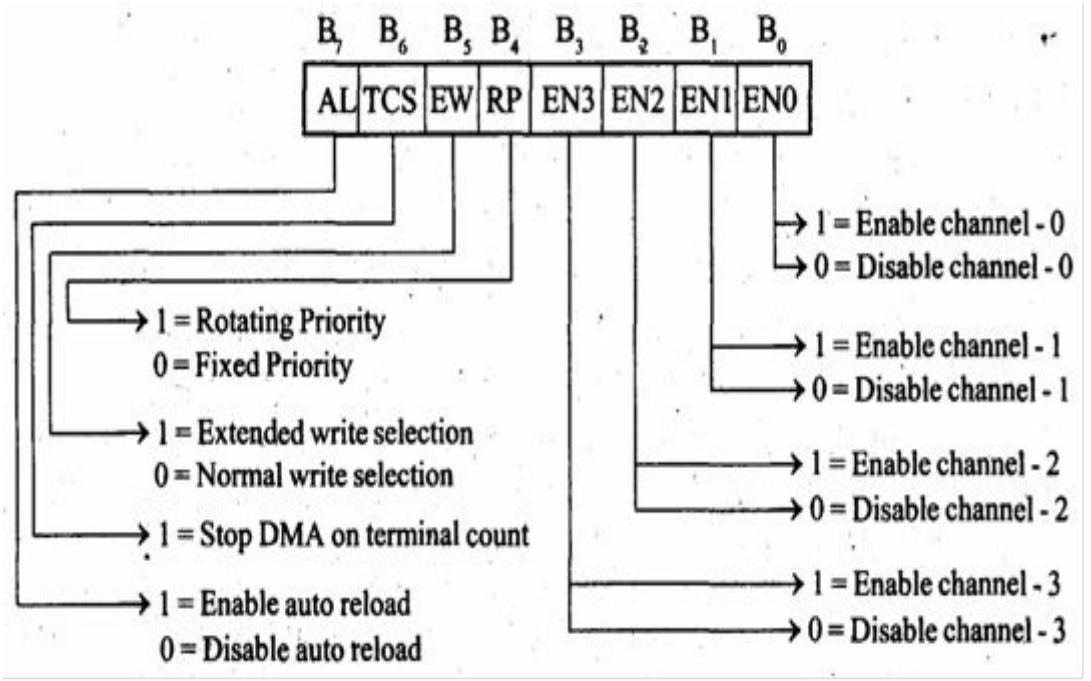


Operational Modes of 8257 DMA Controller

- 14-bits B0-B13 is used to count value and a 2-bits is used for indicate the type of DMA transfer (Read/Write/Verify transfer).
- In read transfer the data is transferred from memory to I/O device.
- In write transfer the data is transferred from I/O device to memory.
- Verification operation is used to verify the transferred data by the DMA controller.

Operational Modes of 8257 DMA Controller

- The 8257 has two 8-bit registers called mode set register and status register. The format of mode set register is,



- **Uses of mode set register:**

1. Enable/disable a channel.
2. Fixed/rotating priority
3. Stop DMA on terminal count.
4. Extended/normal write time.
5. Auto reloading of channel-2.

Operational Modes of 8257 DMA Controller

- The bits B0, B1, B2, and B3 of mode set register are used to enable/disable channel -0, 1, 2 and 3 respectively. A one in this bit position will enable a particular channel and a zero will disable it.
- If the bit B4 is set to one, then the channels will have rotating priority and if it zero then the channels wilt have fixed priority. In rotating priority after servicing a channel its priority is made as lowest. In fixed priority the channel-0 has highest priority and channel-2 has lowest priority.
- If the bit B5 is set to one, then the timing of low write signals (MEMW and IOW) will be extended.
- If the bit B6 is set to one then the DMA operation is stopped at the terminal count. Stop the operation of DMA when terminal count is reached (means count value is zero).
- The bit B7 is used to select the auto load feature for DMA channel-2.
- When bit B7 is set to one, then the content of channel-3 count and address registers are loaded in channel-2 count and address registers respectively whenever the channel-2 reaches terminal count. When this mode is activated the number of channels available for DMA reduces from four to three.

Operational Modes of 8257 DMA Controller

The format of status register of 8257 is,

- The bit B0, B1, B2, and B3 of status register indicates the terminal count status of channel-0, 1,2 and 3 respectively. A one in these bit positions indicates that the particular channel has reached terminal count.
- These status bits are cleared after a read operation by microprocessor.
- The bit B4 of status register is called update flag and a one in this bit position indicates that the channel-2 register has been reloaded from channel-3 registers in the auto load mode of operation.