

DMA and DMA Controller

11.1 Introduction

In microprocessor based systems data transfer can be controlled by either software or hardware. Upto this point we have used program instructions to transfer data from I/O device to memory or from memory to I/O device. To transfer data by this method microprocessor has to do following tasks :

1. To fetch the instruction
2. To decode the instruction and
3. To execute the instruction.

To carryout these tasks microprocessor requires considerable time, so this method of data transfer is not suitable for large data transfers such as data transfer from magnetic disk or optical disk to memory. In such situations hardware controlled data transfer technique is used.

11.1.1 Software Controlled Data Transfer

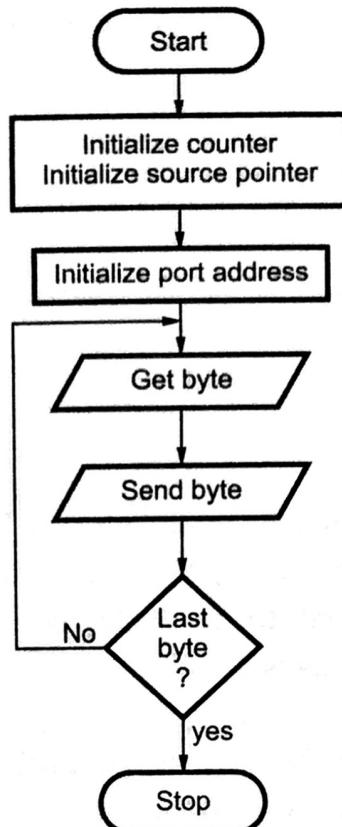
In this method programmer executes a series of instructions to carry out data transfer. The sample flow chart and program required to transfer data from memory to I/O device is shown in Fig. 11.1

Program :

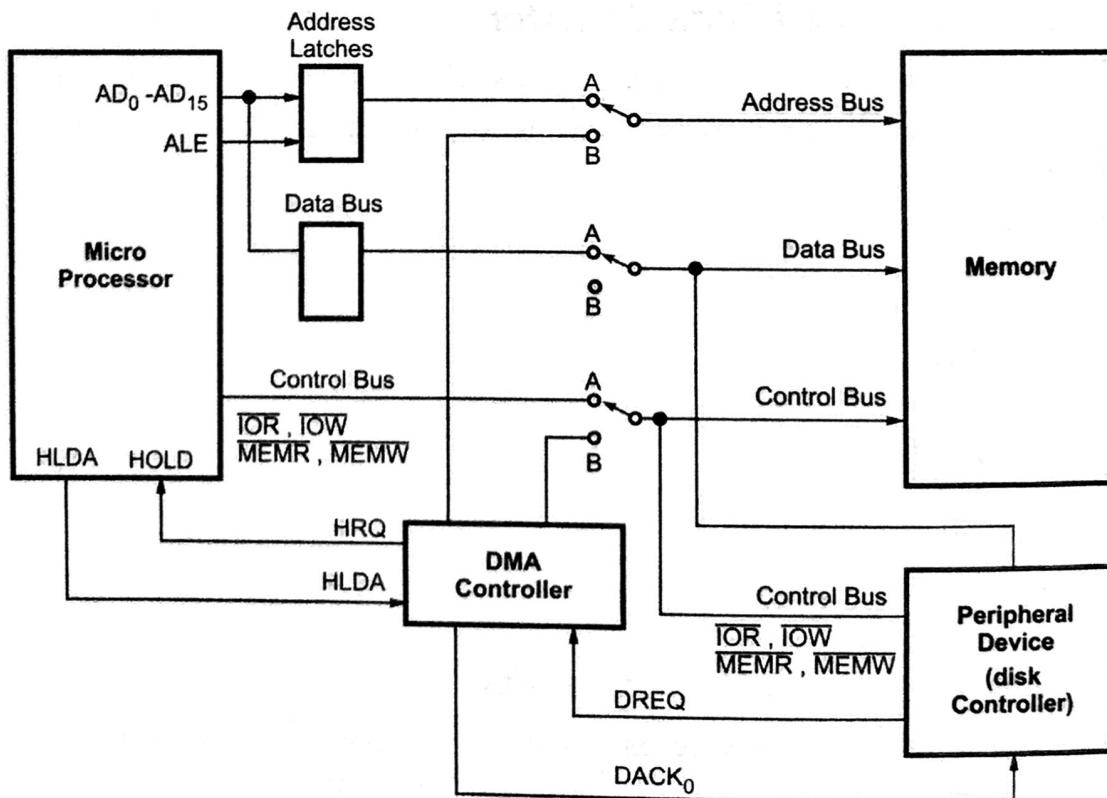
```
Transfer Subroutine
LXI H,6000H
BACK : MOV A,M
        OUT PA
        MOV A,L
        CPI 20H
        JNZ BACK
        RET
```

11.1.2 Hardware Controlled Data Transfer

In this technique external device is used to control data transfer. External device generates address and control signals required to control data transfer and allows peripheral device to directly access the memory. Hence this technique is referred to as **Direct Memory Access (DMA)** and external device which controls the data transfer is

**Fig. 11.1 Flowchart**

referred to as **DMA controller**. Fig. 11.2 shows that how DMA controller operates in a microprocessor system.

**Fig. 11.2 DMA controller operating in a microprocessor system**

11.1.2.1 DMA Idle Cycle

When the system is turned on, the switches are in the A position, so the buses are connected from the microprocessor to the system memory and peripherals. Microprocessor then executes the program until it needs to read a block of data from the disk. To read a block of data from the disk microprocessor sends a series of commands to the disk controller device telling it to search and read the desired block of data from the disk. When disk controller is ready to transfer first byte of data from disk, it sends DMA request DRQ signal to the DMA controller. Then DMA controller sends a hold request HRQ signal to the microprocessor HOLD input. The microprocessor responds this HOLD signal by floating its buses and sending out a hold acknowledge signal HLDA, to the DMA controller. When the DMA controller receives the HLDA signal, it sends a control signal to change switch position from A to B. This disconnects the microprocessor from the buses and connects DMA controller to the buses.

11.1.2.2 DMA Active Cycle

When DMA controller gets control of the buses, it sends the memory address where the first byte of data from the disk is to be written. It also sends a DMA acknowledge, DACK signal to the disk controller device telling it to get ready for data transfer. Finally (in case of DMA write operation), it asserts both the IOR and MEMW signals on the control bus. Asserting the IOR signal enables the disk controller to output the byte of data from the disk on the data bus and asserting the MEMW signal enables the addressed memory to accept data from the data bus. In this technique data is transferred directly from the disk controller to the memory location without passing through the CPU or the DMA controller.

When the data transfer is complete, the DMA controller unasserts the HOLD request signal to the microprocessor and releases the bus by changing switch position from B to A. After getting the control of all buses the microprocessor executes the remaining program.

11.2 Data Transfer Modes of DMA

The three data transfer modes of DMA controller are :

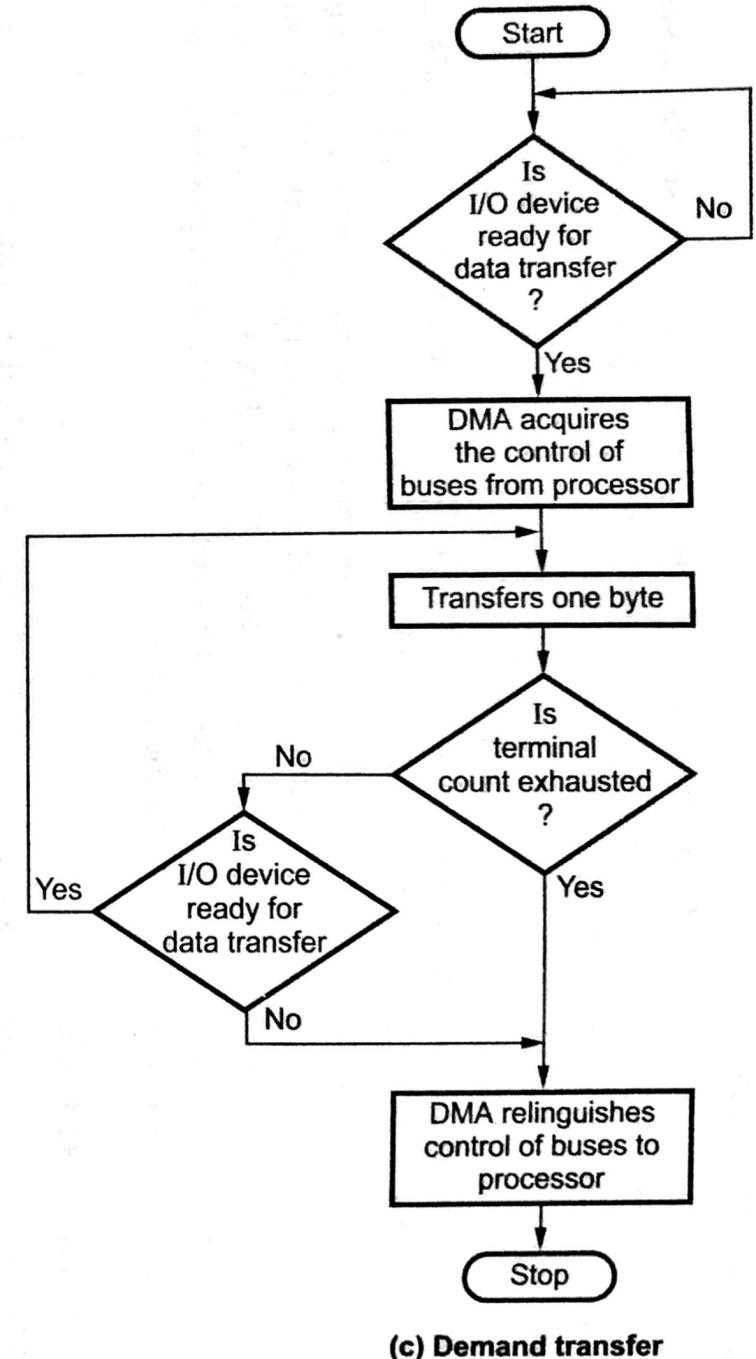
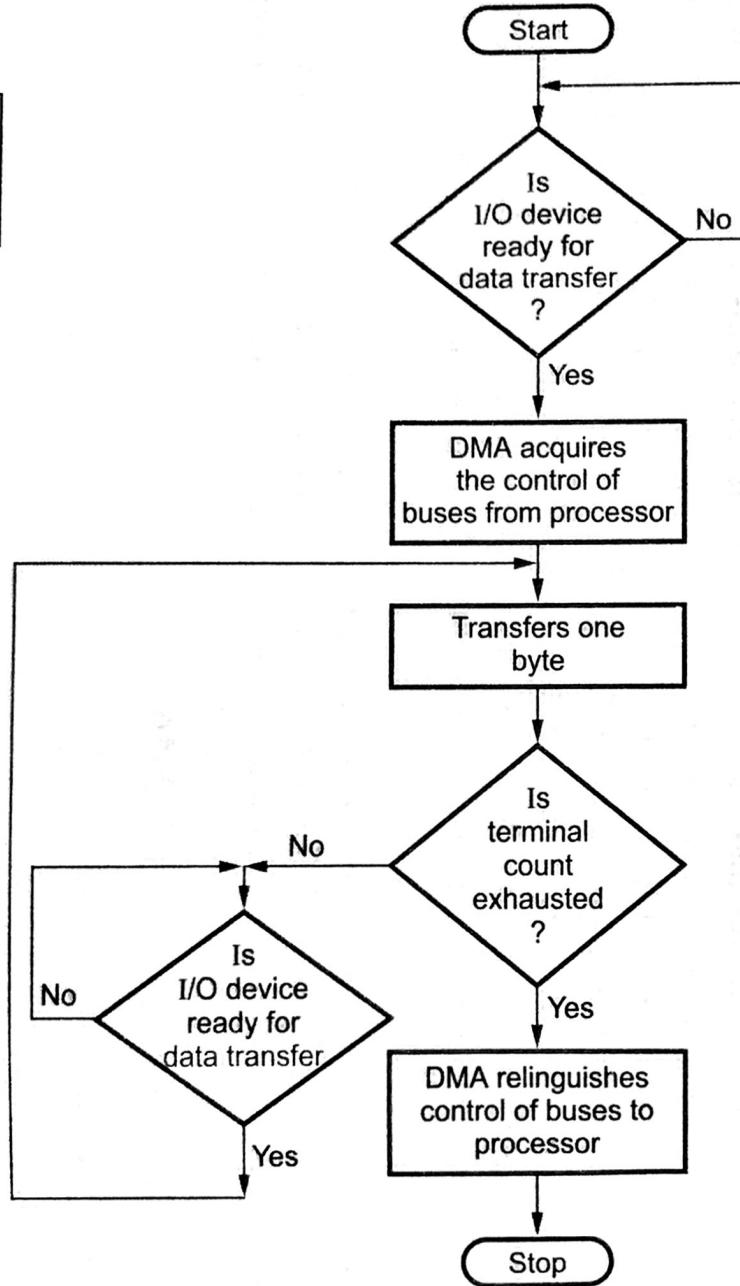
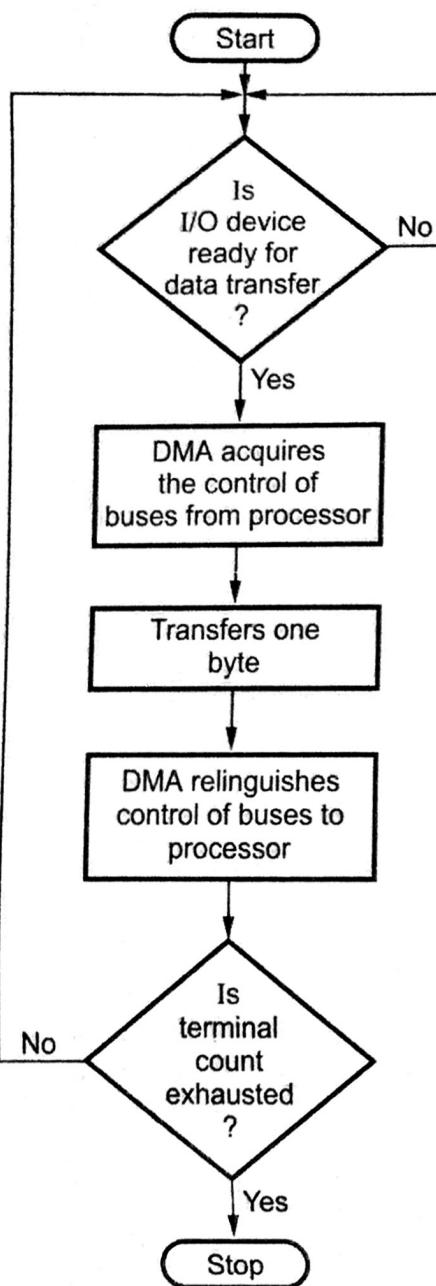
- Single Transfer Mode
- Block Transfer Mode
- Demand or Burst Transfer Mode

Single Transfer Mode

In this mode device can make only one transfer (byte or word). After each transfer DMAC gives the control of all buses to the processor. Due to this processor can have access to the buses on a regular basis.

It allows the DMAC to time share the buses with the processor, hence this mode is most commonly used.

Fig. 11.3 Three data transfer modes of DMA



The operation of the DMA in a single transfer mode is as given below :

1. I/O device asserts DRQ line when it is ready to transfer data.
2. The DMAC asserts HLDA line to request use of the buses from the processor.
3. The processor asserts HLDA, granting the control of buses to the DMAC.
4. The DMAC asserts DACK to the requesting I/O device and executes DMA bus cycle, resulting data transfer.
5. I/O device deasserts its DRQ after data transfer of one byte or word.
6. DMA deasserts DACK line.
7. The word/byte transfer count is decremented and the memory address is incremented.
8. The HOLD line is deasserted to give control of all buses back to the processor.
9. HOLD signal is reasserted to request the use of buses when I/O device is ready to transfer another byte or word. The same process is then repeated until the last transfer.
10. When the transfer count is exhausted, terminal count is generated to indicate the end of the transfer.

Block Transfer Mode

In this mode device can make number of transfers as programmed in the word count register. After each transfer word count is decremented by 1 and the address is decremented or incremented by 1. The DMA transfer is continued until the word count "rolls over" from zero to FFFFH, a Terminal Count (TC) or an external END of Process (EOP) is encountered. Block transfer mode is used when the DMAC needs to transfer a block of data.

The operation of DMA in block transfer mode is as given below :

1. I/O device asserts DRQ line when it is ready to transfer data.
2. The DMAC asserts HLDA line to request use of the buses from the microprocessor.
3. The microprocessor asserts HLDA, granting the control of buses to the DMAC.
4. The DMAC asserts DACK to the requesting I/O device and executes DMA bus cycle, resulting data transfer.
5. I/O device deasserts its DRQ after data transfer of one byte or word.
6. DMA deasserts DACK line.
7. The word/byte transfer count is decremented and the memory address is incremented.
8. When the transfer count is exhausted, the data transfer is not complete and the DMAC waits for another DMA request from the I/O device, indicating that it has

another byte or word to transfer. When DMAC receives DMA request steps through are repeated.

9. If the transfer count is not exhausted, the data transfer is complete then DMAC deasserts the HOLD to tell the microprocessor that it no longer needs the buses.
10. Microprocessor then deasserts the HLDA signal to tell the DMAC that it has resumed control of the buses.

Demand or Burst Transfer Mode

In this mode the device is programmed to continue making transfers until a TC or external EOP is encountered or until DREQ goes inactive.

The operation of DMA in demand transfer mode is as given below :

1. I/O device asserts DRQ line when it is ready to transfer data.
2. The DMAC asserts HLDA line to request use of the buses from the microprocessor.
3. The microprocessor asserts HLDA, granting the control of buses to the DMAC.
4. The DMAC asserts DACK to the requesting I/O device and executes DMA bus cycle , resulting data transfer.
5. I/O device deasserts its DRQ after data transfer of one byte or word.
6. DMA deasserts DACK line.
7. The word/byte transfer count is decremented and the memory address is incremented.
8. The DMAC continues to execute transfer cycles until the I/O device deasserts DRQ indicating its inability to continue delivering data. The DMAC deasserts HOLD signal, giving the buses back to microprocessor. It also deasserts DACK.
9. I/O device can re-initiate demand transfer by reasserting DRQ signal.
10. Transfer continues in this way until the transfer count has been exhausted.

The flowcharts in the Fig. 11.3 summarized the three data transfer modes of DMA (See Fig. 11.3 on previous page).

11.3 DMA Controller 8237

11.3.1 Features of 8237A

1. It has 4 identical channel allow external devices to directly transfer information to/from the system memory.
2. It also allows memory-to-memory data transfer.
3. The 8237A is designed to be used in conjunction with an external 8-bit address register such as the 8282. It contains four independent channels and can be expanded to any number of channels by cascading additional controller chips.

4. Each channel has a full 64K address and word count capability.
5. DMA controllers 8237A, 8237A-4, 8237A-5 operate with 3 MHz, 4 MHz, and 5 MHz respectively. The DMA controller 8237A-5 can transfer up to 1.6 Mbytes/second.

11.3.2 Pin Diagram of 8237A

Fig. 11.4 shows the pin diagram of 8237A.

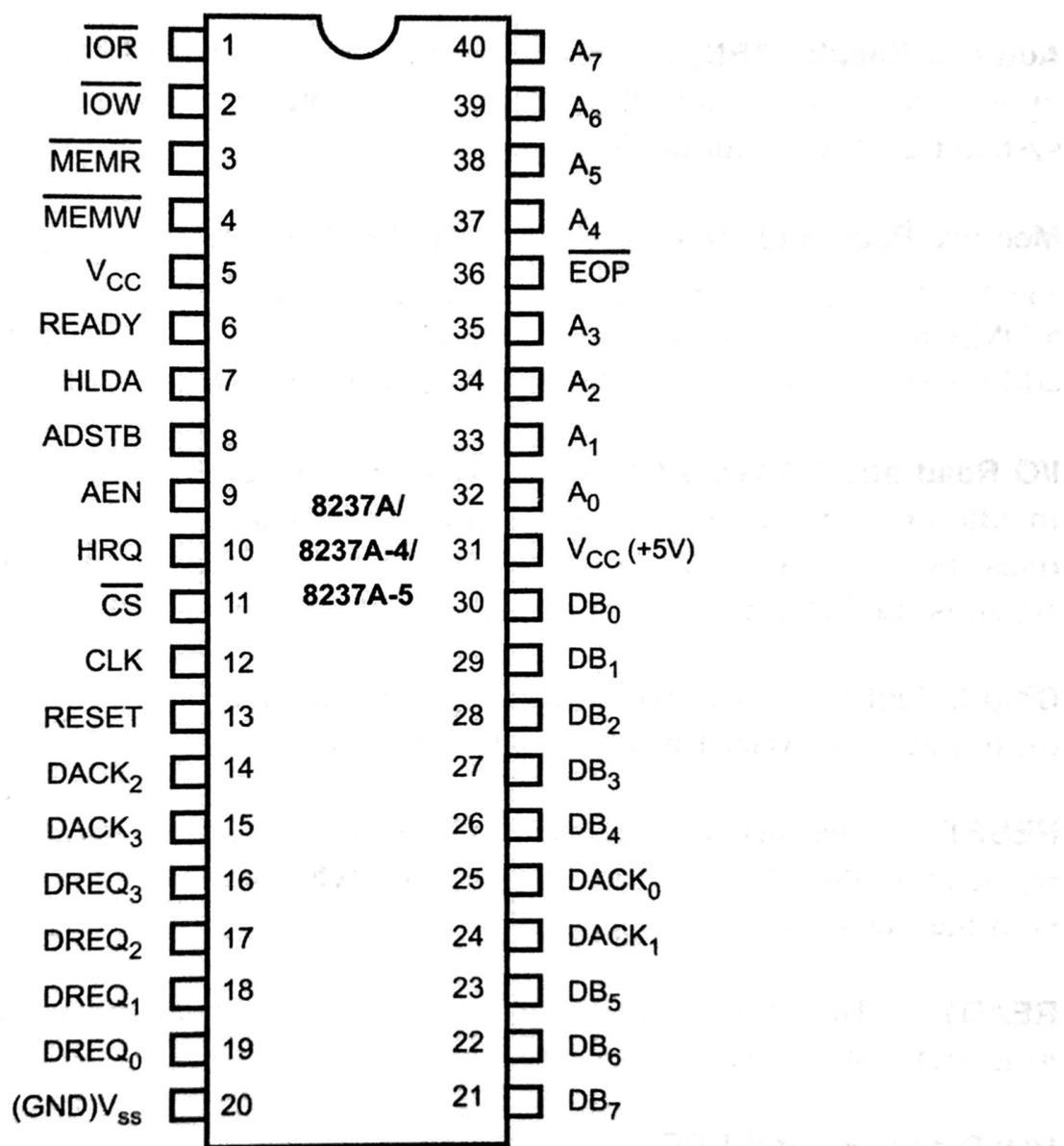


Fig. 11.4 Pin diagram of 8237A

Data Bus (DB₀-DB₇) :

These are bi-directional tri-state signals connected to the system data bus. When CPU is having control of system bus it can read contents of address register, status register, temporary register or a word count register and it can also program control registers of DMA controller, through the data bus. During DMA cycles these lines are used to send the most significant bytes of the address.

Address Bus (A_0 - A_3 and A_4 - A_7) : The four least significant lines A_0 - A_3 are bi-directional tri-state signals. In the idle cycle they are inputs and are used by the CPU to address the register to be loaded or read. In the active cycle they output the lower 4 bits of the address for DMA operation. A_4 - A_7 are unidirectional lines, provide 4-bits of address during DMA service.

Address Strobe (ADSTB) : This signal is used to demultiplex higher byte address and data using external latch.

Address Enable (AEN) : This active high signal enables the 8-bit latch containing the upper 8-address bits onto the system address bus. AEN can also be used to disable other system bus drivers during DMA transfers.

Memory Read and Memory Write ($\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$) : These are active low tri-state signals. The $\overline{\text{MEMR}}$ signal used to access data from the addressed memory location during a DMA read or memory-to-memory transfer and $\overline{\text{MEMW}}$ signal is used to write data to the addressed memory location during DMA write or memory to memory transfer.

I/O Read and I/O Write ($\overline{\text{IOR}}$ and $\overline{\text{IOW}}$) : These are active low bi-directional signals. In idle cycle, these are an input control signals used by CPU to read/write the control registers. In the active cycle $\overline{\text{IOR}}$ signal is used to access data from a peripheral and $\overline{\text{IOW}}$ signal is used to load data to the peripheral.

Chip Select : This is an active low input, used to select the 8237A as an I/O device during the idle cycle. This allows CPU to communicate with 8237A.

RESET : This active high signal clears the command, status, request and temporary registers. It also clears the first flip-flop and sets the Master Register. After reset the device is in the idle cycle.

READY : This input is used to extend the memory read and write pulses from the 8237A to interface slow memories or I/O devices.

HOLD request and HRQ : Any valid DREQ causes 8237A to issue the HRQ. It is used for requesting CPU to get the control of system bus.

HOLD Acknowledge (HLDA) : The active high Hold Acknowledge from the CPU indicates that it has relinquished control of the system bus.

DREQ₀-DREQ₃ : These are DMA request lines, which are activated to obtain DMA service, until the corresponding DACK signal goes active.

DACK₀-DACK₃ : These are used to indicate peripheral devices that the DMA request is granted.

End Of Process (EOP) : This is active low bi-directional signal concern with the completion of DMA service. The EOP output signal is activated at the end of DMA service. The 8237A allows an external signal to terminate an active DMA service by polling the EOP input low.

11.3.3 Block Diagram of 8237A

Fig. 11.5 shows the internal block diagram of 8237A. It contains blocks of control logic and internal registers.

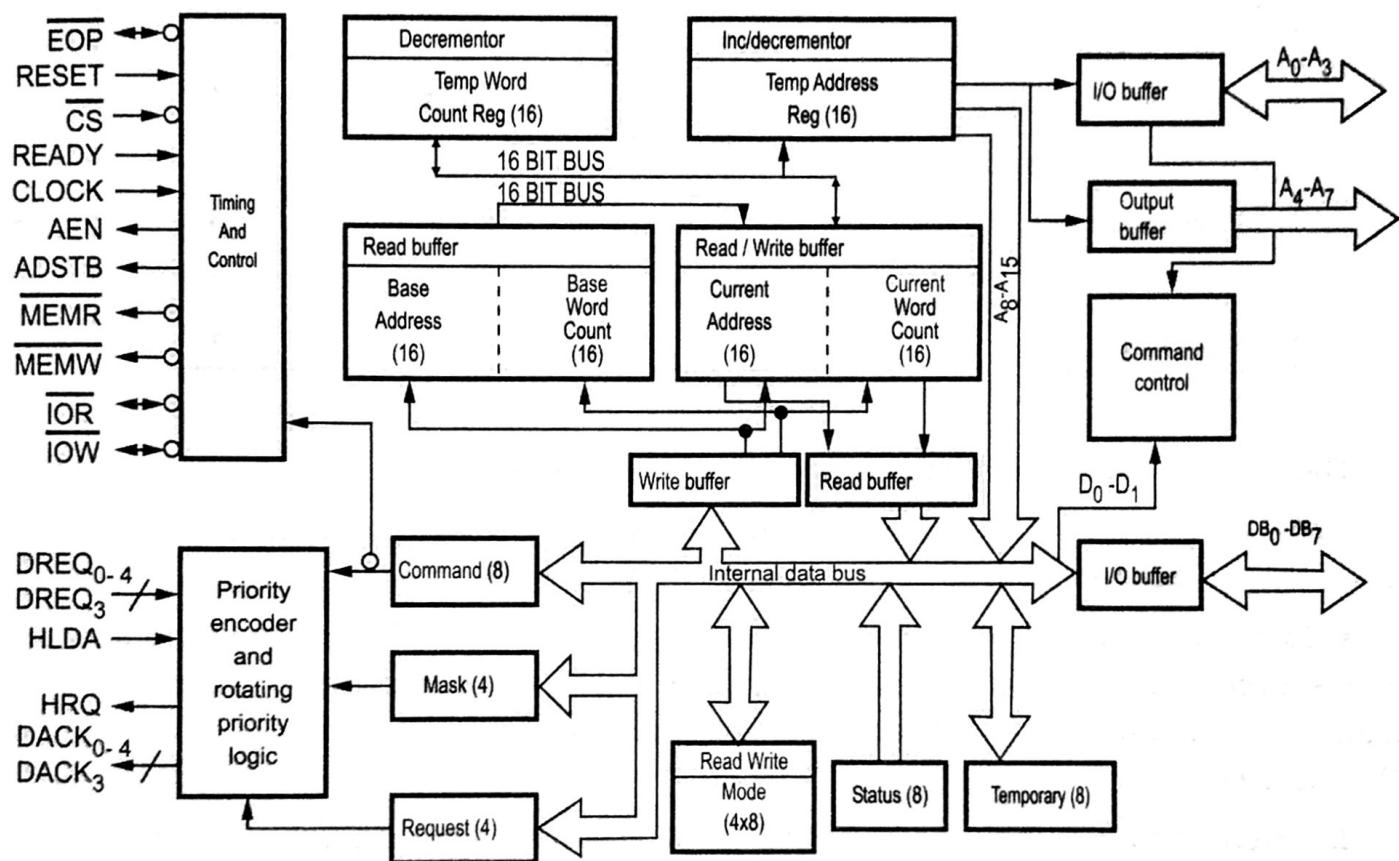


Fig. 11.5 Internal block diagram of 8237A

Control Logic : The 8273A contains three basic blocks of control logic.

1. Timing Control Block : It generates internal timing and external control signals for the 8237A.

2. Program Command Control Block : It decodes various commands given to the 8237A by the microprocessor before servicing a DMA request. It also decodes the Mode Control Word, which is used to select the type of DMA during the servicing.

3. Priority Encoder Block : It resolves the priority between DMA channels requesting service simultaneously.

Internal Registers : The 8237A contains 344 bits internal memory in the form of registers. Table 11.1 gives the name, size and number of each register.

Name	Size	Number
Base Address Registers	16 bits	4
Base Word Count Registers	16 bits	4
Current Address Registers	16 bits	4
Current Word Count Registers	16 bits	4
Temporary Address Registers	16 bits	1
Temporary Word Count Registers	16 bits	1
Status Registers	8 bits	1
Command Registers	8 bits	1
Temporary Registers	8 bits	1
Mode Registers	6 bits	4
Mask Registers	4 bits	1
Request Registers	4 bits	1

Table 11.1

The detailed description and bit pattern of each register is given in the next section.

DMA Operation : As discussed earlier DMA, 8237A can operate in two major cycles, idle cycle and active cycle.

1. Idle cycle : When no channel is requesting service, the 8237A enters the idle cycle. In this cycle the 8237A polls the DREQ lines every CLOCK cycle to determine if any channel is requesting a DMA service. Microprocessor configures DMA channels in the DMA idle cycle.

2. Active cycle : In this mode DMAC transfers data in one of the four modes.

- Single Transfer Mode
- Block Transfer Mode
- Demand Transfer Mode
- Cascade Mode