CHAPTER 17

8237 DMA Controller

17.1 Introduction

Q. Explain about any DMA Controller.

In I/O data transfer, data is transferred, by using microprocessor. The microprocessor will read data from I/O device and then will write data to memory. In this case, there are two operations for single data transfer. If the ***data*** is less, then microprocessor will not waste its time; transferring data from I/O to memory or back. But suppose, ***data*** is huge, then the transfer rate from I/O to memory or back will slow down because of microprocessor intervention. In such case, to speed up the process of transferring the data, we can think, ***Can I/O have direct access to memory ?***; and the answer is, yes. It can have ***Direct memory access (DMA)***, but under ***Supervision***. The device which supervises, data transfer is named as ***DMA controller***. Now let's have diagrammatic representation of the scheme, which depicts microprocessor, DMA controller, memory and I/O device.

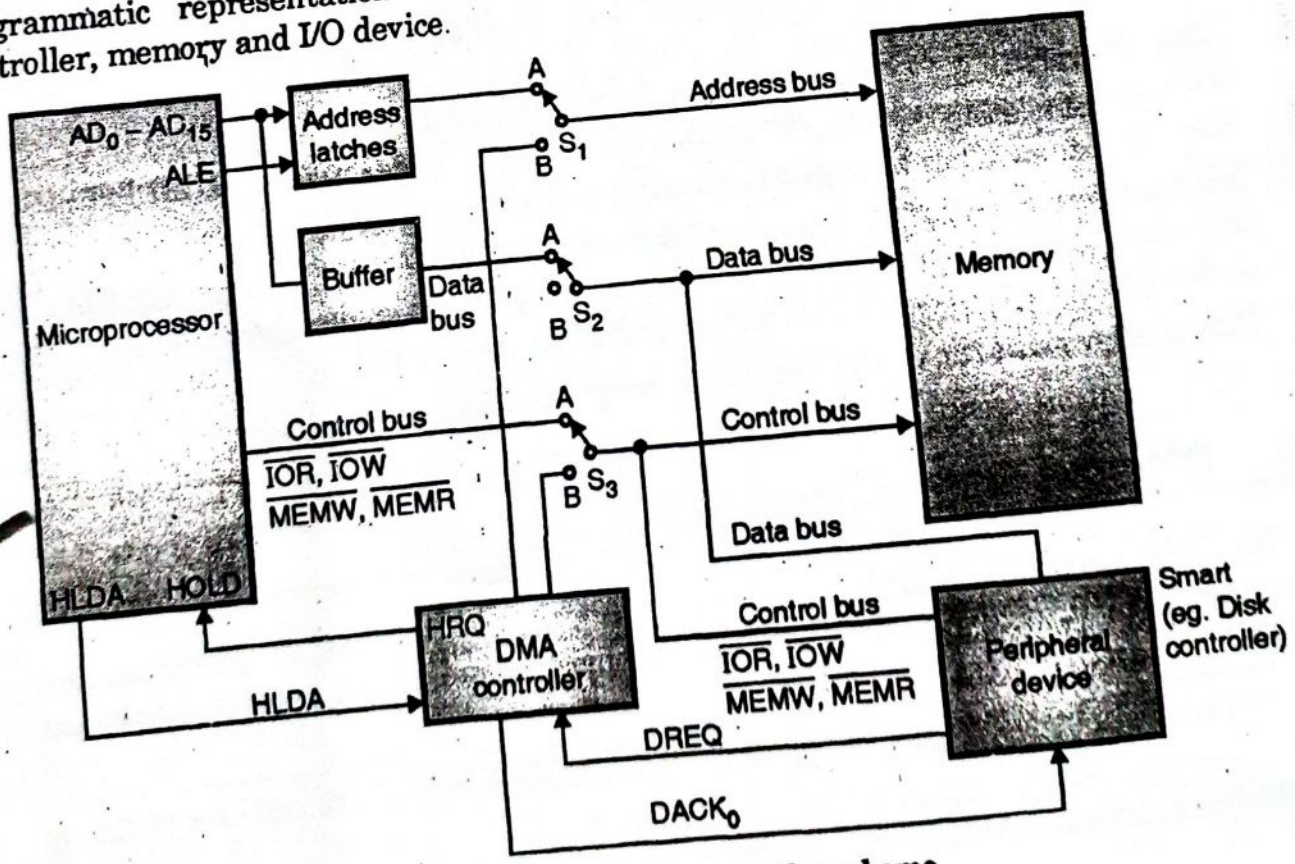

Fig. 17.1.1 : DMA controller scheme

Let's understand the concept clearly.

1. Initially, switches S_1, S_2 and S_3 are at position A.
2. No direct access to memory by I/O.
3. Microprocessor is MASTER of all three buses; address, data and control.
4. Microprocessor treats **DMA controller**, as I/O device ONLY.
5. Using IN/OUT instruction, you can program **DMA controller** chip, for various modes.
6. Whenever, peripheral device is ready to transfer data, DIRECTLY to memory, it will generate REQUEST (DRQ → DMA REQUEST), to DMA controller, asking for direct access.
7. In response to DRQ, DMA controller will activate, HRQ (HOLD request); connected to HOLD pin of microprocessor. By activating HOLD line, DMA controller request microprocessor, to HOLD for sometime and allow him to become a master of all three buses.
8. Moment HOLD pin is HIGH, microprocessor will complete the present job and also activate HLDA (HOLD acknowledge) signal; informing, DMA controller to become master.
9. Microprocessor tristates, all its buses, so total cutoff from memory and I/O device. Thus microprocessor relinquishes the buses and provides control to DMA controller.
10. Now DMA controller is master. It will position all three switches to position B.
11. DMA controller will also generate $\overline{\text{DACK}}$ (DMA acknowledge) signal to peripheral device; informing that, direct access is allowed.
12. Now it will generate address and control signal. Data will flow from memory to I/O or vice versa.
13. After completing data transfer, DMA controller will deactivate HOLD line. It also positions, switches back to position A.
14. Now microprocessor will regain the control over the three buses.
15. Microprocessor will start executing instructions from main program. Till DMA is inactive or not master of the bus, is referred as **DMA IDLE Cycle**. When DMA controller gains the control, it is referred as **DMA Active Cycle**.

This is about basics of DMA. Now let's study the different methods of transferring data.

17.2 DMA Controller : Data Transfer Modes

The DMA controller functions as a bus master and bus slave. It performs data transfer operations. DMA controlled input/output is further divided into the following categories :

- Burst or Block transfer DMA.
- Cycle steal or Single byte transfer DMA.
- Transparent or Hidden DMA.

17.2.1 Burst or Block Transfer DMA

- It is the fastest DMA mode.
- In this mode, two or more data bytes are transferred continuously.
- The microprocessor is disconnected from the system bus during DMA transfer i.e. the microprocessor cannot executes its own program during this transfer.
- N number of DMA cycles are added into the machine cycles of the microprocessor where N is number of bytes to be transferred.
- In this mode, the DMA controller sends ***'HOLD'*** signal to the microprocessor and waits for HLDA signal.
- After receiving HLDA signal, the DMA controller gains control of the system bus and executes a DMA cycle to transfer one byte.
- After transferring one byte, it increments memory address, decrements counter and transfers next byte.
- In this way, it transfers all data bytes between memory and I/O devices. After transferring all data bytes, the DMA controller disables ***'HOLD'*** signal and enters into slave mode.

17.2.2 Cycle Steal or Single Byte Transfer DMA

- In cycle steal transfer only one byte of data is transferred at a time.
- This type of DMA is slower than burst DMA.
- In this mode, only one DMA cycle is added between two machine cycles of the microprocessor, hence the instruction execution speed of the microprocessor is reduced slightly.
- In this mode the DMA controller sends ***'HOLD'*** signal to the microprocessor and waits for HLDA signal.
- After receiving HLDA signal, the DMA controller gains control of the system bus and executes only one DMA cycle.
- After transferring one byte, it disables ***'HOLD'*** signal and enters into slave mode.
- The microprocessor then gains control of the system bus and executes next machine cycle. If the count is not zero and next data is available then the DMA controller sends ***'HOLD'*** signal to the microprocessor and transfers next byte of data block.

17.2.3 Transparent or Hidden DMA Transfer

- The microprocessor executes some states during which it floats the address and data buses.
- During these states, the microprocessor is isolated from the system bus.
- The DMA controller transfers data between memory and I/O devices during these states. This operation is transparent to microprocessor.

- This is the slowest DMA transfer, In this mode, the instruction execution speed of microprocessor is not reduced. But, the transparent DMA requires logic to detect the states when the microprocessor is floating the buses.

Now, we will study DMA controller chip 8237.

17.3 8237 High Performance Programmable DMA Controller

17.3.1 Features

- It provides various modes of DMA.
- It provides on chip four independent DMA channels. The number of channels can be increased by cascading DMA controller chips.
- Each channel can be used in auto initialization mode.
- It can transfer data between two memory blocks in DMA mode i.e. memory to memory transfer.
- In memory to memory transfer a single word can be written into all location of memory block.
- The address of memory is either incremented or decremented after each DMA cycle depending upon the mode.
- The clock frequency is 3 MHz (8237) or 5 MHz (8237-2).
- The data transfer rate is very high e.g. 1.6 M bytes/second for 8237-2 at 5 MHz.
- Directly expandable to any number of channels. It doesn't require any additional chip for cascading. There is no limitations on cascading.
- It provides EOP line that is used to terminate DMA operation. This signal can be generated by external hardware.
- The DMA can be requested by setting an appropriate bit of request register.
- Independent control for DREQ and DACK signal. DREQ and DACK signals can be initialized either for active high or active low.
- It provides compressed timings to improve throughput of the system. It can compress the transfer time to two cycles (2S).

17.4 Pin Configuration of 8237

Q. 1 Draw and explain pin configuration of 8237.

Q. 2 Explain the significance of AEN, ADSTB, $\overline{EOP}$, READY pins of 8237.

The Fig. 17.4.1 shows pin configuration of 8237.

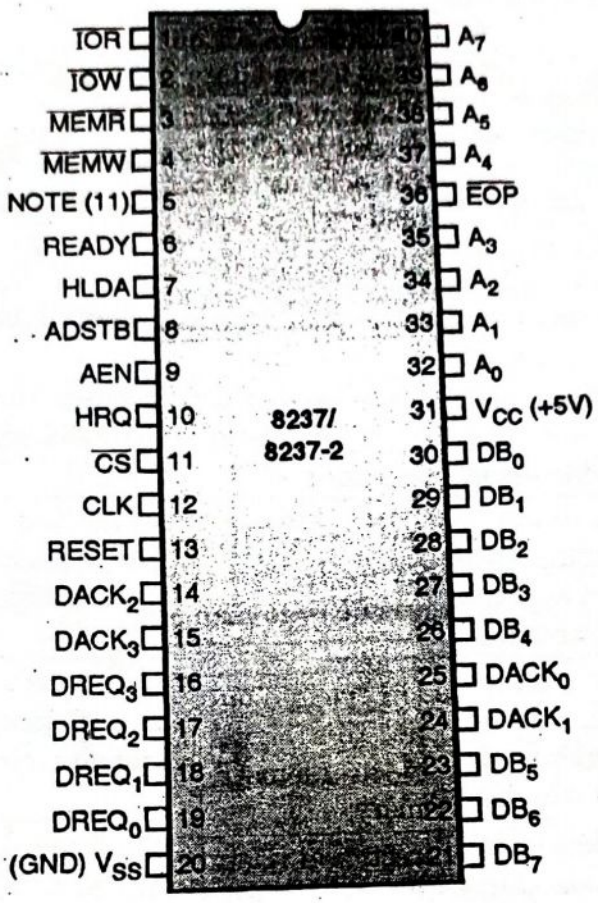

Fig. 17.4.1 : Pin configuration of 8237

Symbol	Description
CLK	This is a clock input line ignored in slave mode. In master mode, this signal controls all internal and external DMA operations. The data transfer rate depends upon the frequency of this signal.
$\overline{CS}$	In slave mode, this signal is generated by address decoder to select 8237 chip for communication between CPU and 8237. In master mode, this signal is ignored.
Reset	It is an asynchronous input line. This signal clears the command, status, request and temporary register and forces 8237 into slave mode.
READY	In master mode, this signal is used to add wait states into a DMA cycle.
HRQ	It is a hold request output line. It is connected to hold input of the CPU. It is used to request control of the system bus.
HLDA	It is a hold acknowledge input line. This signal is generated by CPU. In response to this signal, the 8237 gains control of the system bus and enters into master mode.
$\overline{IOR}$	It is an active low bi-directional tristate line. In slave mode, it acts as an input line and used to read contents of 8237 registers. In

Symbol	Description
	master mode, it acts as an output line. This signal is generated during DMA cycle to read data from I/O device.
$\overline{IOW}$	It is an active low bi-directional line. In slave mode, it acts as an input line and used by CPU to write contents to 8237 registers. In master mode, it acts as an output line. This signal is generated during DMA read cycle to write data into I/O device.
A_0 - A_3	These are bi-directional, address lines. In slave mode, these lines act as input lines, used to select one of the registers of 8237. In master mode, the 8237 provides lower bits of memory address on these lines.
A_4 - A_7	These are tristate address output lines. These lines are tristated in slave mode. In master mode, the 8237 transfers bits of memory address on these lines.
$\overline{MEMR}$	It is an active low tristate control output line. It is tristated in slave mode. In master mode, this signal is generated during DMA read cycle or during memory to memory transfer cycle to read contents of source memory.
$\overline{MEMW}$	It is an active low tristate output line. It is tristated in slave mode. In master mode, this signal is activated during DMA write or during memory to memory transfer cycle to write data into destination memory.
DB_0 - DB_7	These are bi-directional tristate buffered data lines. In slave mode, these lines are used to transfer data between CPU and 8237's registers. In master mode, these lines act as address output lines. The 8237 places higher byte of address on these lines during DMA cycles.
AEN	This active high output enables the 8 bit latch that drives the upper 8 bit address bus. The AEN pin is used to disable other bus drivers during DMA transfers.
ADSTB (Address Strobe)	This output line is used to strobe the upper address byte generated by 8237 in master mode into an external latch.
$DREQ_0$ -$DREQ_3$	These are asynchronous DMA channel request lines used by peripheral. The polarity of each signal is programmable i.e. these lines can be used as either active high or active low input. DREQ must be maintained until the corresponding DACK is activated.
$DACK_0$ - $DACK_3$	These are DMA acknowledge output lines. The polarity of each line is programmable. This signal indicates that the requesting peripheral has been granted for DMA cycle.
$\overline{EOP}$	End of Process : It is an active low bi-directional signal. This line is also used to terminate DMA cycle. The DMA cycle can be terminated by pulling $\overline{EOP}$ input low. The 8237 also generates EOP pulse, when the terminal count for any channel is reached.

17.5 Block Diagram of 8237

Q. Draw and explain the block diagram of 8237 DMA.

The Fig. 17.5.1 shows an internal block diagram of 8237. It consists of logic blocks and internal register.

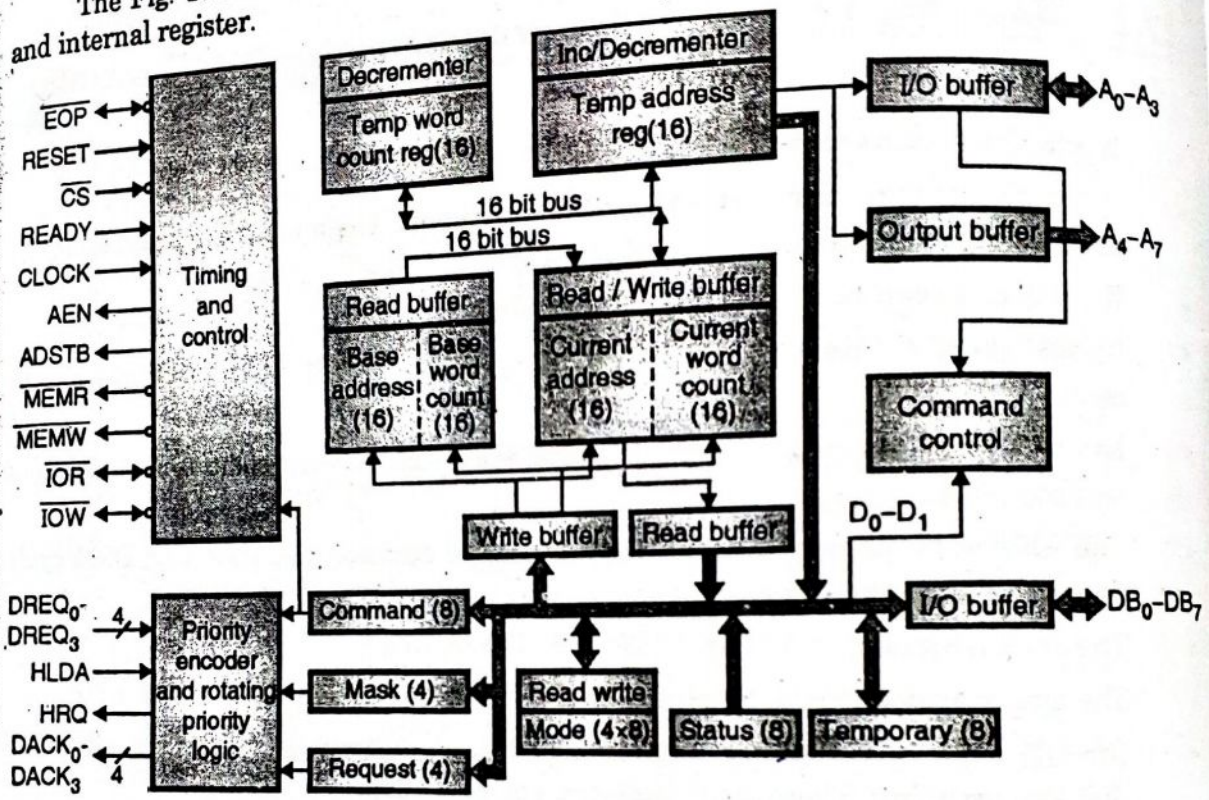

Fig. 17.5.1 : Internal block diagram of 8237

8237 Registers

It contains 344 bits of internal memory in the form of registers. It contains 12 types of registers.

- Current address register
- Base address register
- Command register
- Request register
- Status register
- Current word register
- Base count register
- Mode register
- Mask register
- Temporary register

(1) Current address register

Each channel has a 16 bit current address register.

This register holds the address of memory location to be accessed during current DMA cycle. The address stored in this register is automatically incremented or decremented after each transfer.

- It is a read and write register.
- It is divided into two parts : lower byte and higher byte.

- In autoinitialization mode, it is initialized automatically with original address after $\overline{EOP}$ signal. The current address register format is as shown in Fig. 17.5.2.

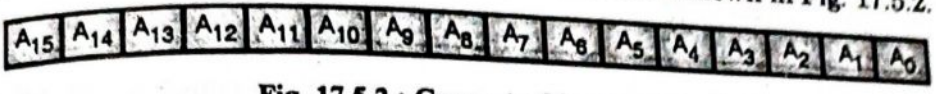

Fig. 17.5.2 : Current address register

(2) Current word register

- Each channel has a 16 bit current word count register.
- The original value stored in this register indicates the number of bytes to be transferred.
- The word count is decremented after each transfer. The current count indicates the numbers of pending transfers.
- When the count value in the register goes to zero, a TC will be generated. It is a read and write register. It is also divided into two parts : lower byte and higher byte.
- In autoinitialization mode, this register is initialized automatically with original count value after $\overline{EOP}$ signal. The current word register format is as shown in Fig. 17.5.3.

Fig. 17.5.3 : Current word register

(3) Base address register

- Each channel has a 16 bit base address register.
- It is a write only register.
- It holds original value of address during all DMA transfers i.e. the contents of this register is not updated during DMA transfers.
- When $\overline{EOP}$ is activated, the 8237 transfers contents of base address registers into current address register in autoinitialization mode.
- This register is written along with current address register during initialization. Format is same as current address register.

(4) Base count register

- Each channel has a 16 bit base word count register.
- It is write only register.
- It holds original count value during all DMA cycle i.e. the contents of this register are not updated during DMA transfers.
- When $\overline{EOP}$ is activated, the 8237 transfers contents of this register into current word register in autoinitialization mode.
- This register is written along with current address register during initialization. Format is same as current word register.

99

(5) Command register

- It is an 8 bit control register. It is a write only register.
- This register is cleared by reset signal. It is used to initialize operation modes of 8237.
- The format of command register is as shown in Fig. 17.5.4.

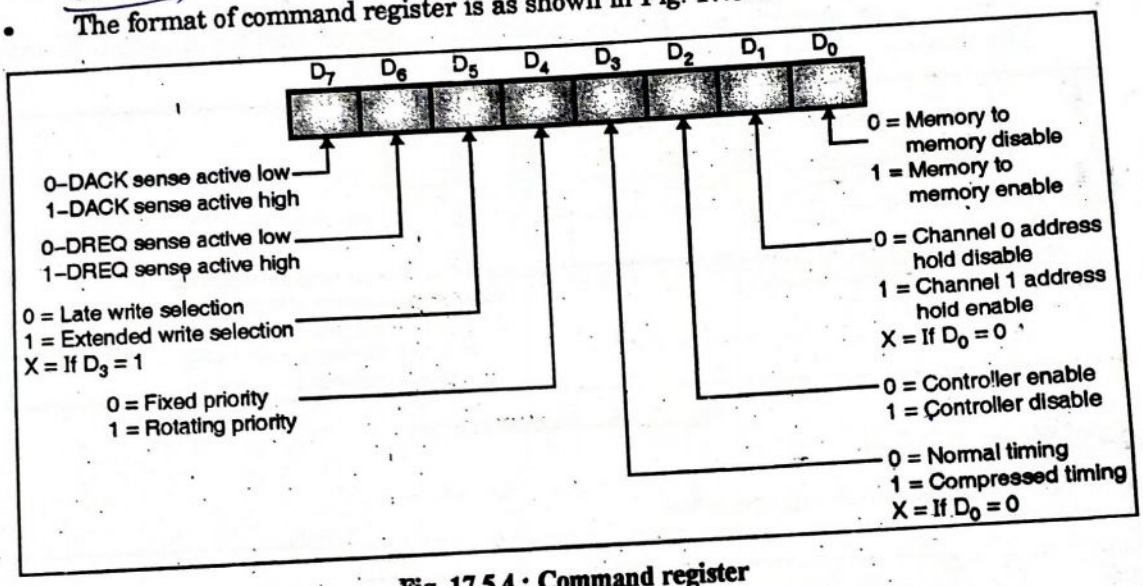

Fig. 17.5.4 : Command register

(6) Mode register

- It is an 8 bit write only register.
- It is used to set operating modes of 8237.
- Each channel has a 6 bit mode register.
- All registers are cleared by reset signal. The format of mode register is as shown in Fig. 17.5.5.

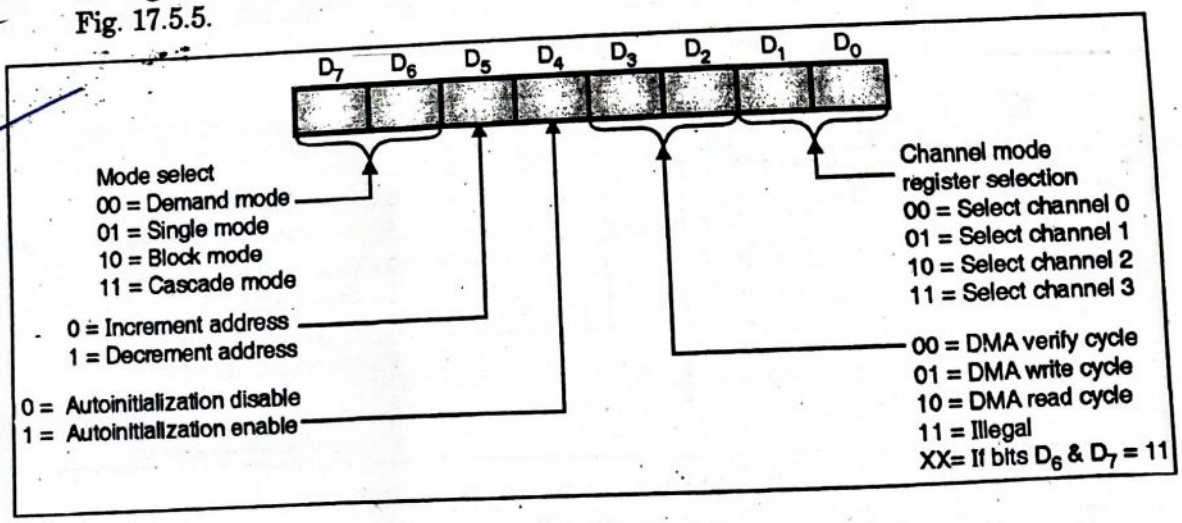

Fig. 17.5.5 : Mode register

(7) Request register

- It is an 8 bit write only register.

- It is used to request DMA through software. Each channel has a request bit associated with it in the 4 bit request register.
- Each register bit is set or reset separately under software control. The request is automatically cleared after TC. It is also cleared by external $\overline{EOC}$ signal.
- This register is cleared by reset signal.
- Software request will be serviced only if the channel is in block mode.
- In memory to memory transfer, the software request for channel 0 should be set.

The format of request register is as shown in Fig. 17.5.6.

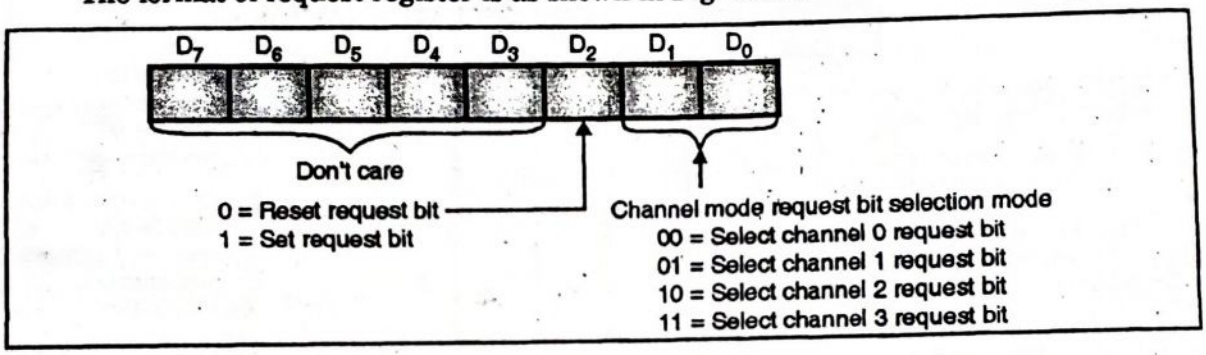

Fig. 17.5.6 : Request register

(8) Mask register

- It is an 8 bit write only register.

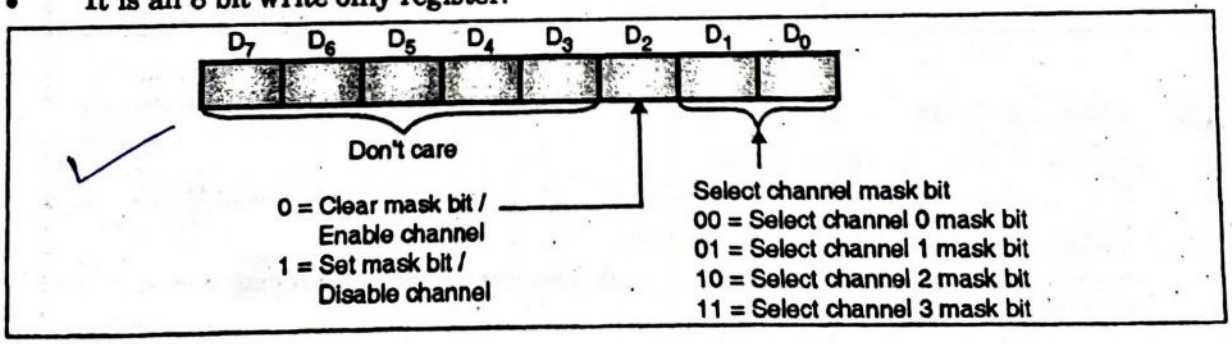

Fig. 17.5.7 : Byte written mask register

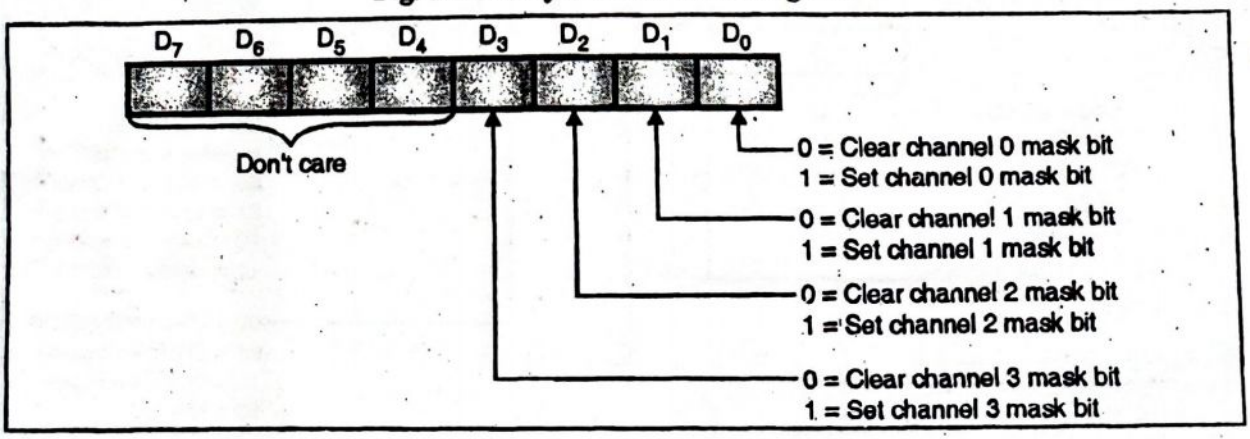

Fig. 17.5.8 : Bit written mask register

- It is used to maskout the channel DMA request.

- In normal mode, mask bit is set automatically after TC. It is not affected in autoinitialization mode.
- The reset signal sets mask bit of all channels i.e. it disables all channels.
- The channel mask bit is selected by D_1 and D_0 bits of the mask format as shown in Fig. 17.5.7. Each bit of the 4 bit mask register can be set or reset separately under software control.
- All 4 bits of the mask register may also be written with a single command as shown in Fig. 17.5.8.

(9) Status register

- It is an 8 bit read only register.
- It indicates which channels have reached a terminal count and which channels have pending DMA requests.
- Bits 0-3 are set every time a TC is reached by that channel or external EOP signal is applied.
- These bits are cleared automatically on reading the status register and upon reset signal.
- Bits 4-7 are set whenever their corresponding channel is requesting service. The format of status register is as shown in Fig. 17.5.9.

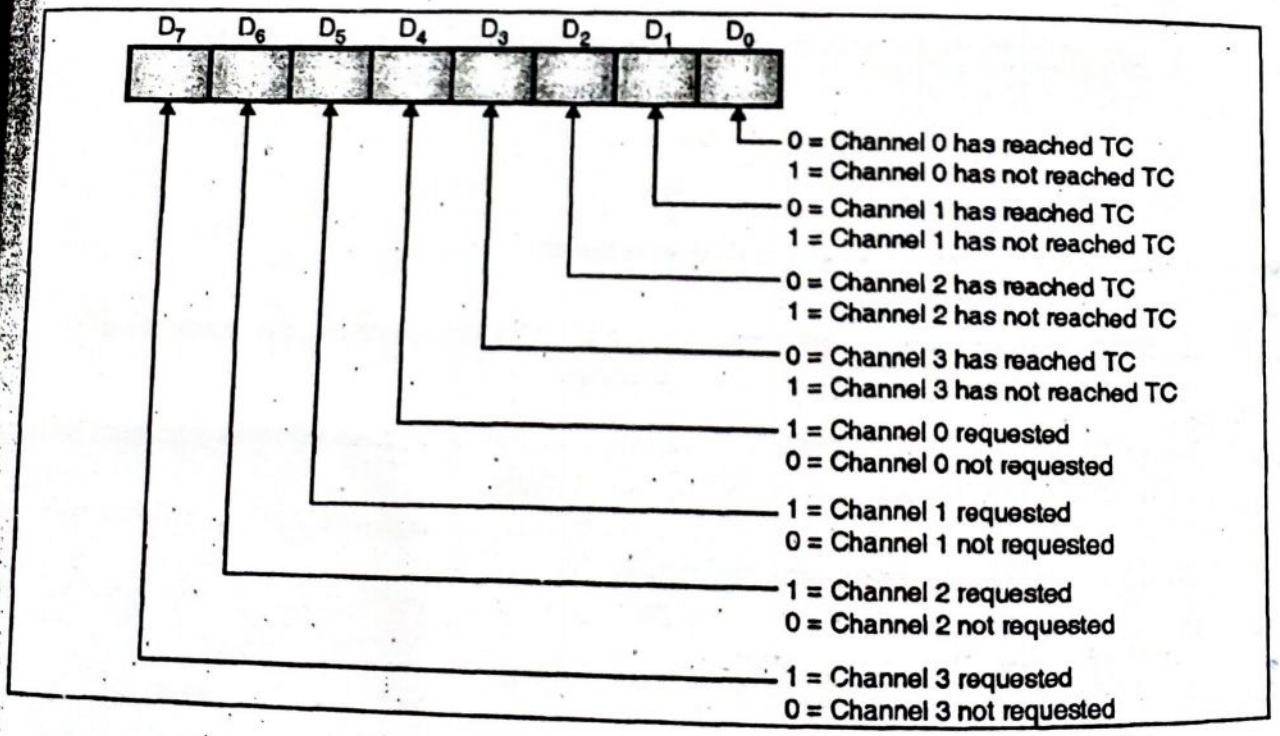

Fig. 17.5.9 : Status register

(10) Temporary register

- This register is used to hold data during memory to memory transfer.
- It is an 8 bit read only register.
- The microprocessor can read last byte of memory to memory transfer.
- It is cleared by reset signal.

(11) Software commands

- These are additional special write only software commands.
- These commands do not require a specific data pattern. i.e. writing only data into corresponding address enables such commands.

(12) Clear first / last flip-flop

- This command is used to reset internal first / last flip-flop.
- After issuing this command the microprocessor can access lower byte of any 16 bit register. The F/L flip-flop is toggled after each read or write 16 bit register.

(13) Master clear

- It is a software reset command.
- It clear command, status, request, temporary registers and F/L flip-flop.
- It sets mask register and forces 8237 in slave mode.
- The Tables 17.5.1 and 17.5.2 shows addresses of all registers.

Table 17.5.1 : Software command codes

Signals						Operation
A_3	A_2	A_1	A_0	$\overline{IOR}$	$\overline{IOW}$	
1	0	0	0	0	1	Read Status Register
1	0	0	0	1	0	Write Command Register
1	0	0	1	0	1	Illegal
1	0	0	1	1	0	Write Request Register
1	0	1	0	0	1	Illegal
1	0	1	0	1	0	Write Single Mask Register Bit
1	0	1	1	0	1	Illegal
1	0	1	1	1	0	Write Mode Register
1	1	0	0	0	1	Illegal
1	1	0	0	1	0	Clear Byte Pointer Flip-Flop
1	1	0	1	0	1	Read Temporary Register
1	1	0	1	1	0	Master Clear
1	1	1	0	0	1	Illegal
1	1	1	0	1	0	Clear Mask Register
1	1	1	1	0	1	Illegal
1	1	1	1	1	0	Write All Mask Register Bits

Table 17.5.2 : Word count and address register command codes

Channel	Register	Operation	Signals							Internal Flip-Flop	Data Bus $DB_0 - DB_7$
			$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A_3	A_2	A_1	A_0		
0	Base and Current Address	Write	0	1	0	0	0	0	0	0	$A_0 - A_7$
			0	1	0	0	0	0	0	1	$A_8 - A_{15}$
	Current Address	Read	0	0	1	0	0	0	0	0	$A_0 - A_7$
			0	0	1	0	0	0	0	1	$A_8 - A_{15}$