8257 DMA CONTROLLER

1) Why DMA ?

Ans When there is a bulk data transfer between memory and and peripheral devices through CPU, it is a time consuming process for such cases DMA (Direct memory Access) method is preferred.

2) What is DMA ?

Ans. It is a technique of transferring data from I/O to memory and from memory to I/O without the intervention of CPU. DMA needs a hardware called DMAC [Direct memory Access Controller] to manage data transfer.

8257 DMAC => It is a 4 channel DMA controller that manages data transfer for microprocessor. The I/O devices request to 8257 for DMA, the 8257 temporarily uses address bus, data bus and control bus from microprocessor and transfers the data directly between I/O and memory.

The DMA controller transfers the data in three modes :->

1) Burst mode :- Once DMA has control of System Bus, it releases the System Bus only after completion of data transfer, meanwhile the CPU has to wait for the system Bus.

2) Cycle stealing mode :- In this mode, DMA forcefully make the CPU to stop its operation and gains control over the bus for short period time to DMAC. after successful data transfer, the DMAC releases the bus and then again request for system bus, like this the DMAC steals the clock cycle for

Page No.
Date

Transferring

3) Transparent Mode :- In this mode, DMAC takes the charge of system Bus only if the CPU / processor does not require the system Bus.

[The DMA Data Transfer Is Initiated Only After Receiving HLDA signal From The CPU]

DMAC Operational Sequence

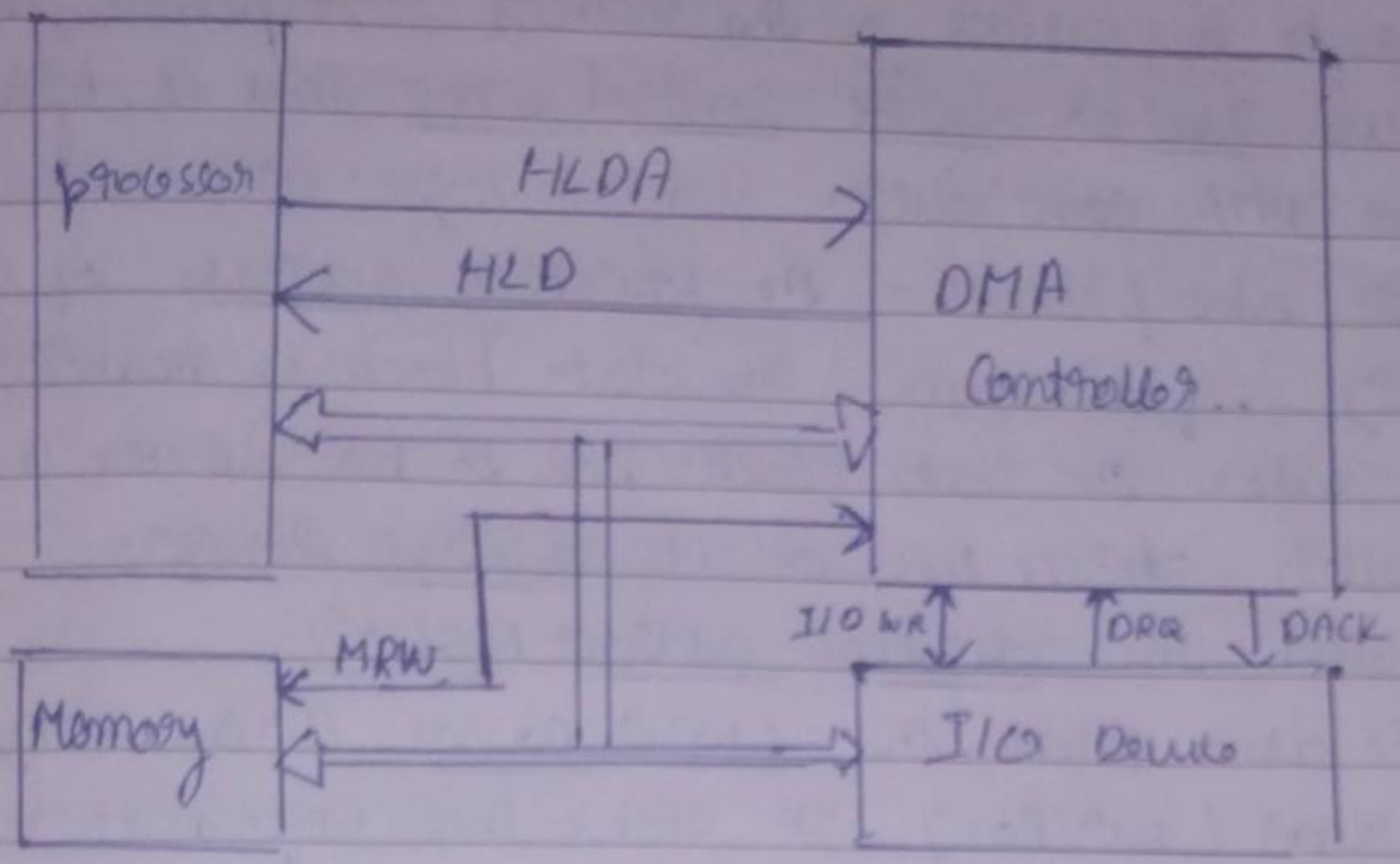
1) Whenever an I/O device wants to transfer the data to or from memory, it sends the DMA request (DRQ) to DMAC. DMAC accepts this and asks the CPU to hold for few clock cycles by sending it the 'Hold Request' [HLD]

2) CPU receives the 'Hold Request' (HLD) from DMAC and relinquishes the bus and sends the 'HOLD Acknowledgement' [HLDA] to DMAC

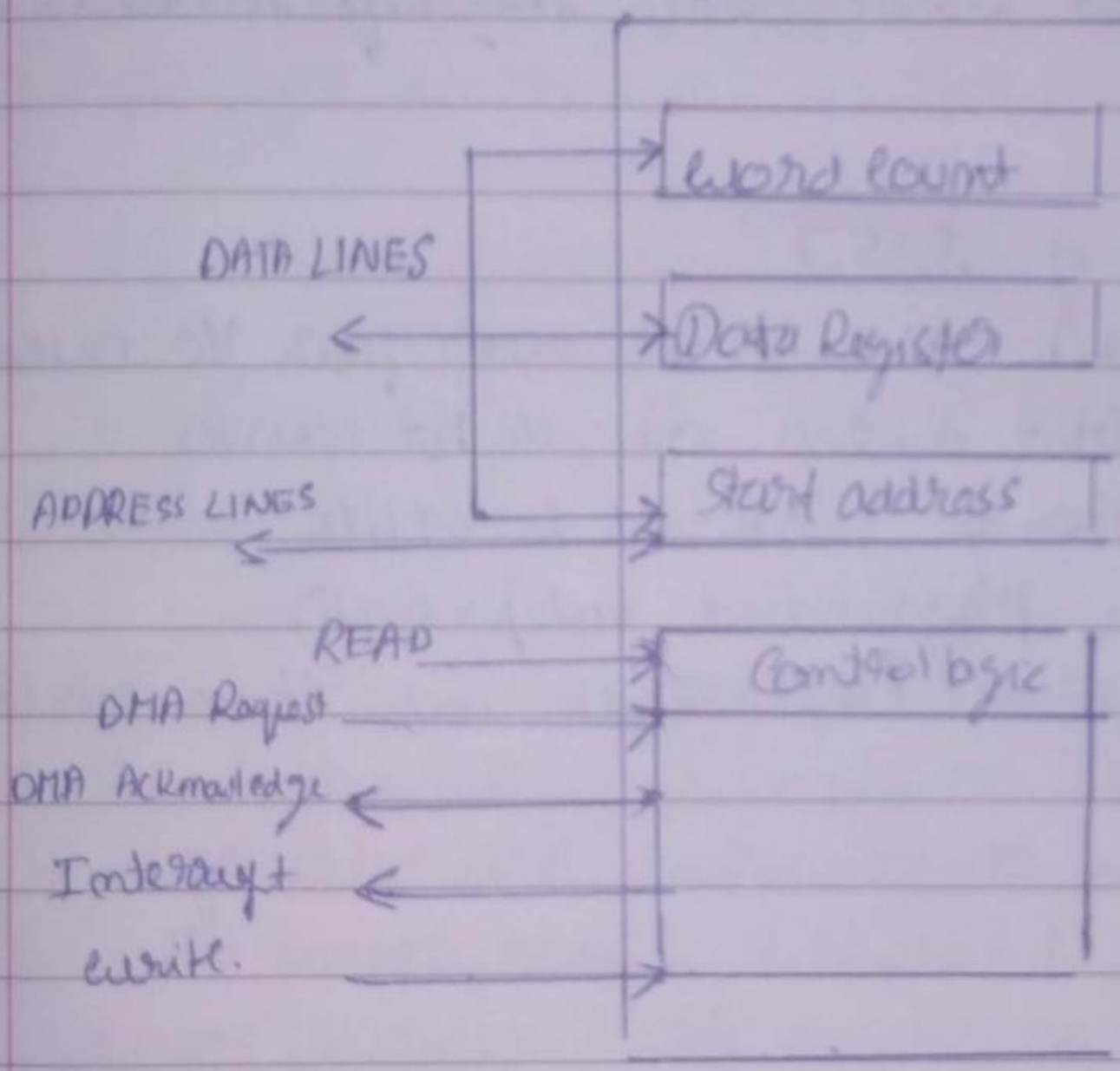
3) After Receiving the "Hold Acknowledgement' (HLDA), DMAC acknowledges I/O device (DACK) that the data transfer can be performed and DMA controller takes the charge of the system bus and transfers the data to or from memory.

4) When the data transfer is accomplished, the DMA raise an 'Interrupt' to let know the processor that the task of data transfer is finished and the processor can take control over the bus again and start processing where it has left.

DMAC



DMA Block Diagram [DMAC]



Whenever a processor is requested to read or write a block of data, it instructs the DMA controller by sending the following information.

1). The 1st information is whether the data has to be read from memory or the data has to be written to the memory. It passes this information like read or write control lines that is between the processor and DMA controllers control logic unit.

2) The processor also provides the starting address of the data block in the memory, from where the data block has to be read or where the data block has to be written in memory. DMA controller stores this in its address Registers. It is also called the starting address Register.

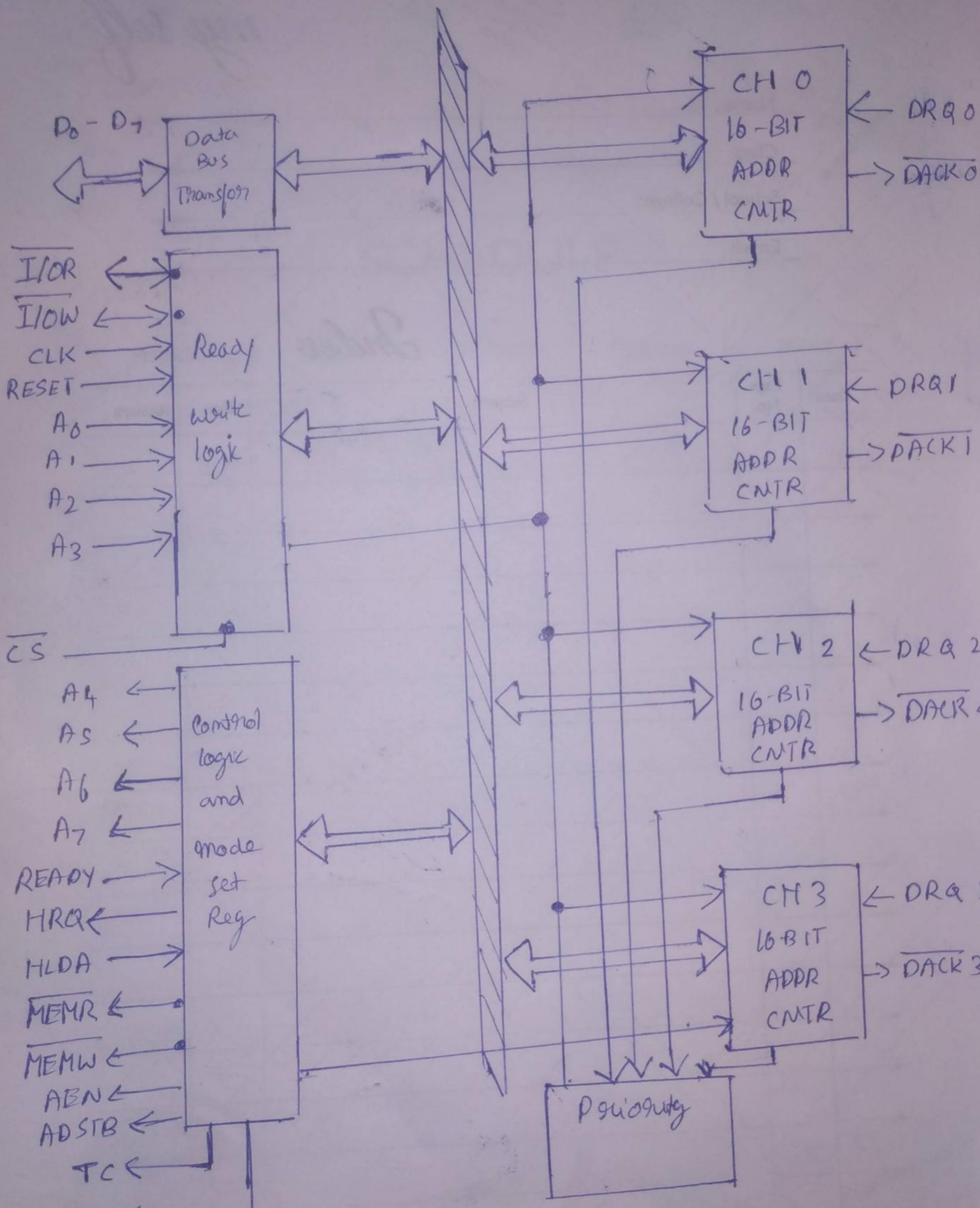
3) The processor also sends the word count [how many words are to be read / written]. It stores this information in Data Count or the word Count register.

4) The most important is the address of I/O device that wants to read or write data. This information is stored in data Register.

Features of 8257

→ it has 4 channel which can be used over four I/O devices.
→ each channel has 16bit address and 14 bit counter
→ each channel can transfer data up to 64 Kb.
→ each channel can be programmed independently
→ It operates in 2 mode :→ Master mode and Slave mode

8257 ARCHIETECTURE



1) DRQ 0 - DRQ 3 : There are 4 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 lowest priority

2) DACK 0 - DACK 3 : These are the active low DMA acknowledge lines, which updates the requesting peripheral about the status of their request by CPU. These lines can also act as strobe lines for requesting devices.

3) D0 - D7 :- Bidirectional 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 Master mode, these lines are used to send higher byte of generated address to latch.

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

5) IOW :- It is active ~~high~~ low bidirection tri-state line, which is used to load content of data bus to 8bit, mode register / upper/lower byte of 16bit DMA address register or terminal count register. In the master mode, it is used to load the data to peripheral devices during DMA memory read cycle.

6) CLK :- It is a clock freqeny signal which is required for internal operation of 8257.

7) RESET :- This signal is used to RESET the DMAC by disabling all DMA channels.

8) $\underline{A_0 - A_3}$:- These are four lowest significant address lines. In the slave mode, they act as an input, which selects one of the registers to be read or written. In master mode, they are the four least significant memory address output lines generated by 8257

9) $\underline{CS}$:- 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.

10) $\underline{A_4 - A_7}$:- These are the higher nibble of lower byte address generated by DMA in the master mode.

11) $\underline{READY}$:- it is an Active high asynchronous input signal, which makes DMA ready by inserting wait states

12) $\underline{HRQ}$ - This signal is used to receive the hold request signal from output device. In slave mode, it is connected with a DRQ input line 8257. In master mode, it is connected with HOLD input of CPU.

13) $\underline{HLDA}$ - 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.

14) $\underline{MEMR}$ - It is the low memory read signal, which is used to read the data from addressed memory location during DMA read cycle.

15) 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.

16) AOST :- This signal is used to connect the higher byte of the memory address generated by the DMA controller into latches

17) AEN :- This signal is used to disable the address bus / data bus

18) TC :-> "Terminal Count" - which indicates the present DMA cycle to the peripheral device.

19) MARK :- The mark will be activated after each 128 cycles from the beginning. It indicates the current DMA cycle is the 128th cycle since previous MARK output to the selected peripheral device.

20) V_{cc} :- It is the power signal which is required for the operation of circuits