DMA CONTROLLER 8257

In microprocessor based system, **data transfer can be controlled by either software or hardware. To transfer data microprocessor has to do the following tasks:**

Fetch the instruction
Decode the instruction
Execution of the instruction

- Microprocessor needs some **amount of time** to complete these tasks. But this data transfer is not suitable for large data transfer such as data transfer from **magnetic disk or optical disk to memory.** In this situation **hardware controlled data transfer technique is required.**
- The Direct Memory Access or DMA mode of data transfer is the **fastest amongst all the modes of data transfer.**
- In this mode, the device may transfer **data directly to/from memory** without any interference from the CPU.
- Intel's 8257 is a **four channel DMA controller** designed to be interfaced with their family of microprocessors. Each channel can be independently programmable to transfer up to 64kb of data by DMA. Each channel can be independently perform read transfer, write transfer and verify transfer.
- In maximum mode of the microprocessor RQ/GT pin is used as bus request input.
- On receiving the **HLDA** signal (in minimum mode) or **RQ/GT** signal (in maximum mode) from the CPU, the requesting devices **gets the access of the bus**, and it completes the required number of DMA cycles for the data transfer and then hands over the control of the **bus back to the CPU**.

Direct Memory Access (DMA)

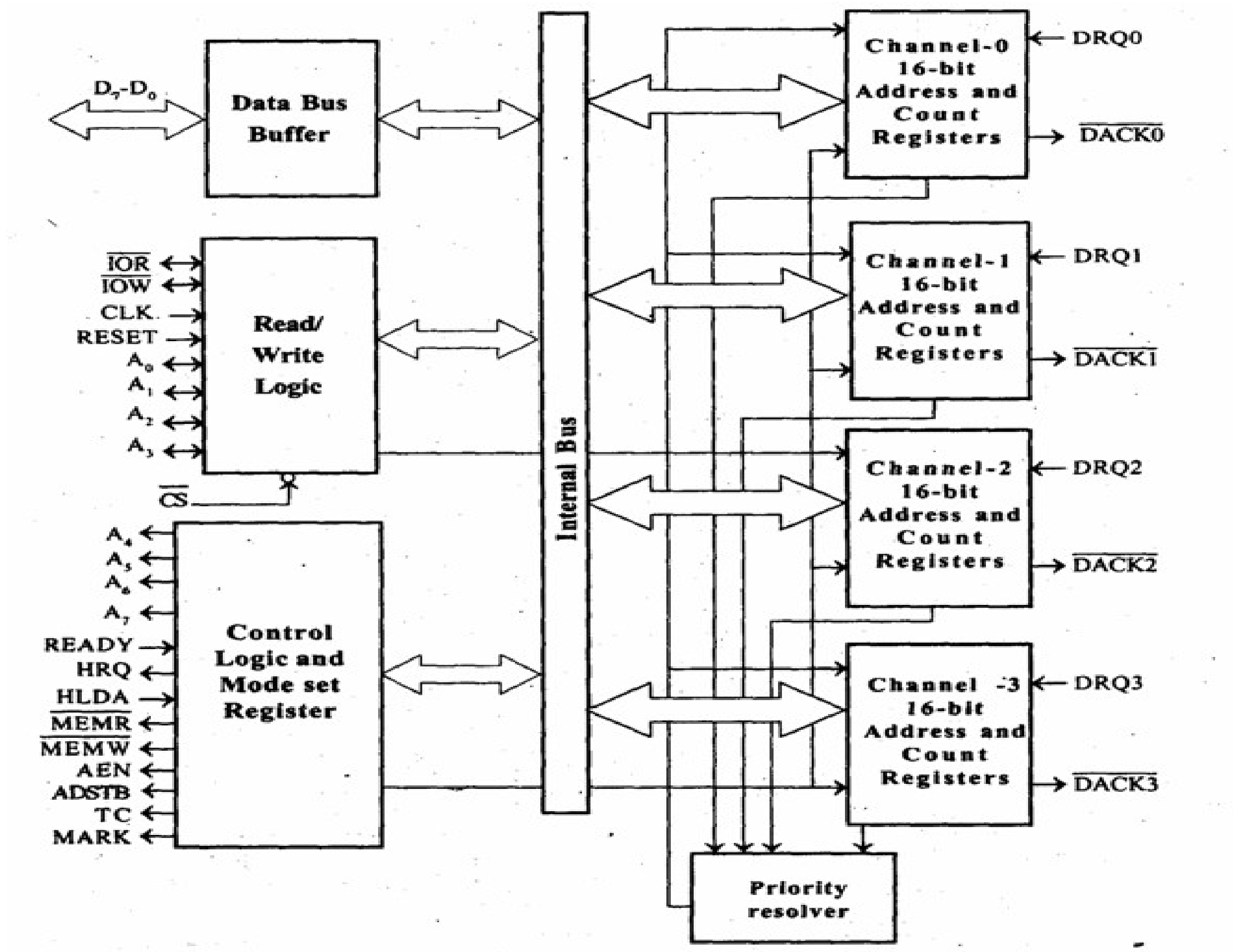
- During **any given bus cycle, one of the system components connected to the system bus is given control of the bus.** This component is said to be the **master** during that cycle and the component it is communicating with is said to be the **slave.**

- The **CPU with its bus control logic is normally the master,** but other specially designed components can gain control of the bus by sending a bus request to the CPU.

- After the current bus cycle is completed the CPU will return a bus grant signal and the component sending the request will become the master.
- **Taking control of the bus for a bus cycle is called cycle stealing.**
- The DMA data transfer is initiated only after receiving **HLDA signal from the CPU.**

- The 8257, on behalf of the devices, requests the CPU for bus access using local bus request input i.e. **HOLD in minimum mode.**

Internal Architecture of 8257


D7-D0
Data Bus Buffer
IOR
IOW
CLK
RESET
A0
A1
A2
A3
Read/ Write Logic
CS
A4
A5
A6
A7
READY
HRQ
HLDA
MEMR
MEMW
AEN
ADSTB
TC
MARK
Control Logic and Mode set Register
Internal Bus
Channel-0 16-bit Address and Count Registers
DRQ0
DACK0
Channel-1 16-bit Address and Count Registers
DRQ1
DACK1
Channel-2 16-bit Address and Count Registers
DRQ2
DACK2
Channel -3 16-bit Address and Count Registers
DRQ3
DACK3
Priority resolver

How DMA Operations are Performed?

Following is the sequence of operations performed by a DMA .

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

- 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.

- 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.

- The chip support four DMA channels, i.e. four peripheral devices can independently request for DMA data transfer through these channels at a time.

The DMA controller has

- 8- bit internal data buffer,
- a read/write unit,
- a control unit,
- a priority resolving unit along with a set of registers.

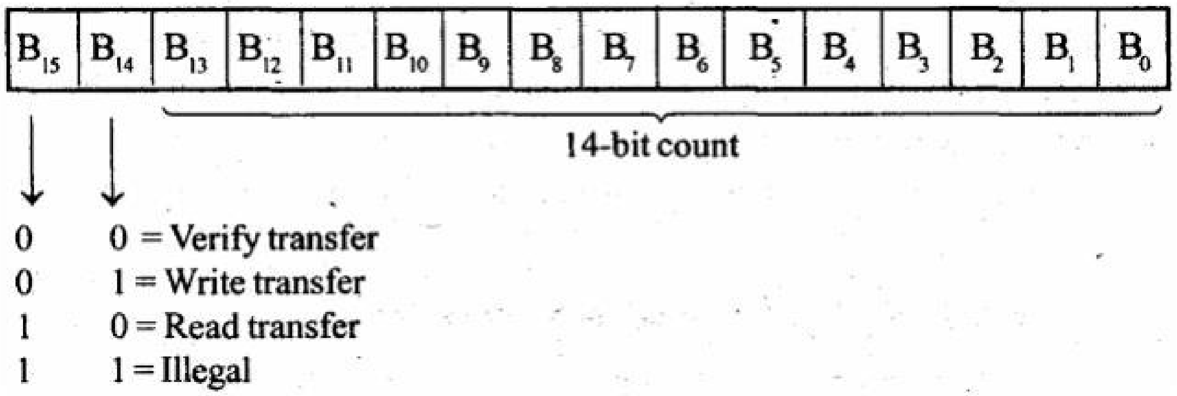
The 8257 performs the DMA operation over **four independent DMA channels.**

- Each of four channels of 8257 has a pair of **two 16- bit registers,** viz. DMA **address register and terminal count register.**

- There are two common registers for all the channels, namely, **mode set register and status register**. Thus there are a total of ten registers.

The CPU selects one of these ***ten registers using address lines Ao- A3.*** *Table shows how the Ao- A3 bits may be used for selecting one of these registers.*

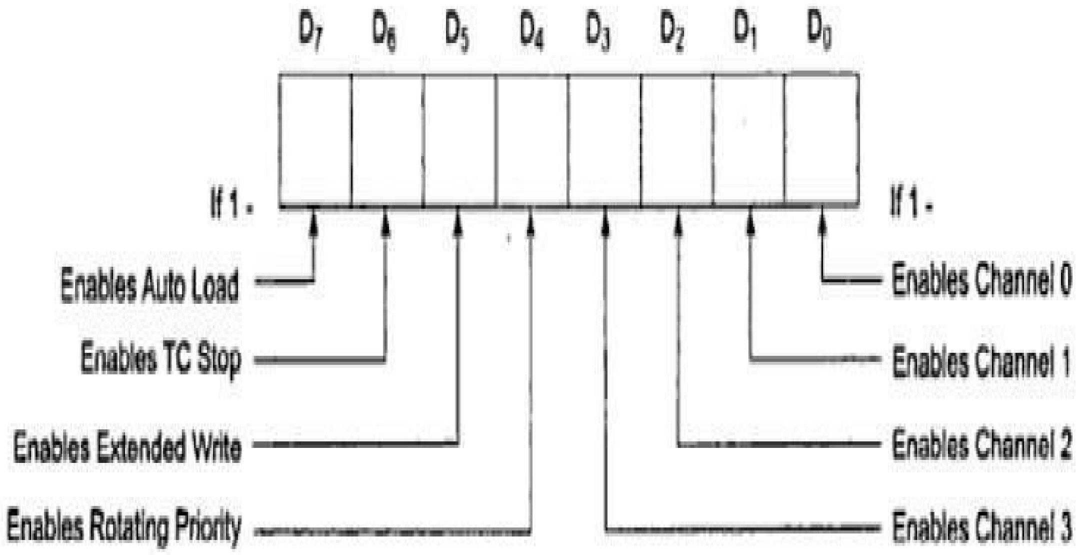
Register	Address			
	A_3	A_2	A_1	A_0
Channel-0 DMA address register	0	0	0	0
Channel-0 Count register	0	0	0	1
Channel-1 DMA address register	0	0	1	0
Channel-1 Count register	0	0	1	1
Channel-2 DMA address register	0	1	0	0
Channel-2 Count register	0	1	0	1
Channel-3 DMA address register	0	1	1	0
Channel-3 Count register	0	1	1	1
Mode set register (Write only)	1	0	0	0
Status register (Read only)	1	0	0	0



B_{15} B_{14} B_{13} B_{12} B_{11} B_{10} B_9 B_8 B_7 B_6 B_5 B_4 B_3 B_2 B_1 B_0
14-bit count
0 0 = Verify transfer
0 1 = Write transfer
1 0 = Read transfer
1 1 = Illegal

Mode Set Register

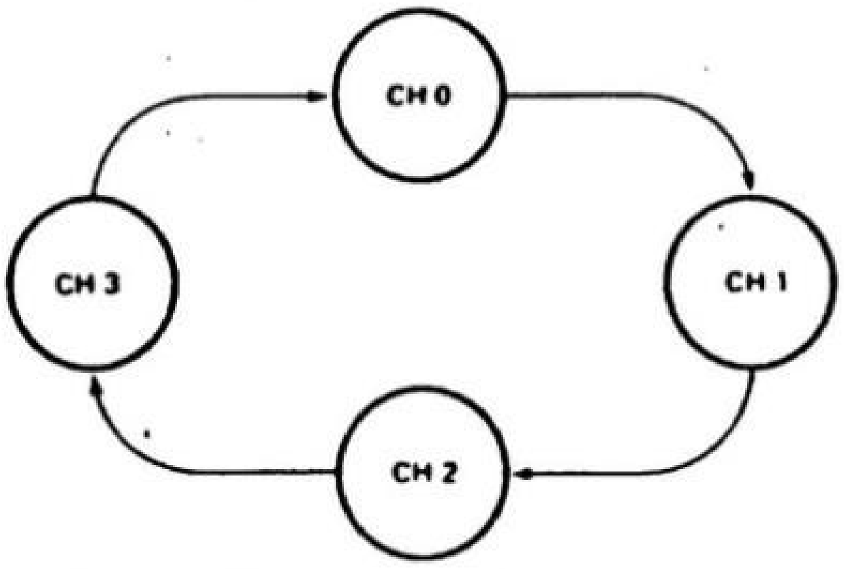
- The mode set register is used for programming the 8257 as per the requirements of the system.
- The function of the mode set register is to enable the **DMA channels individually and also to set the various modes**

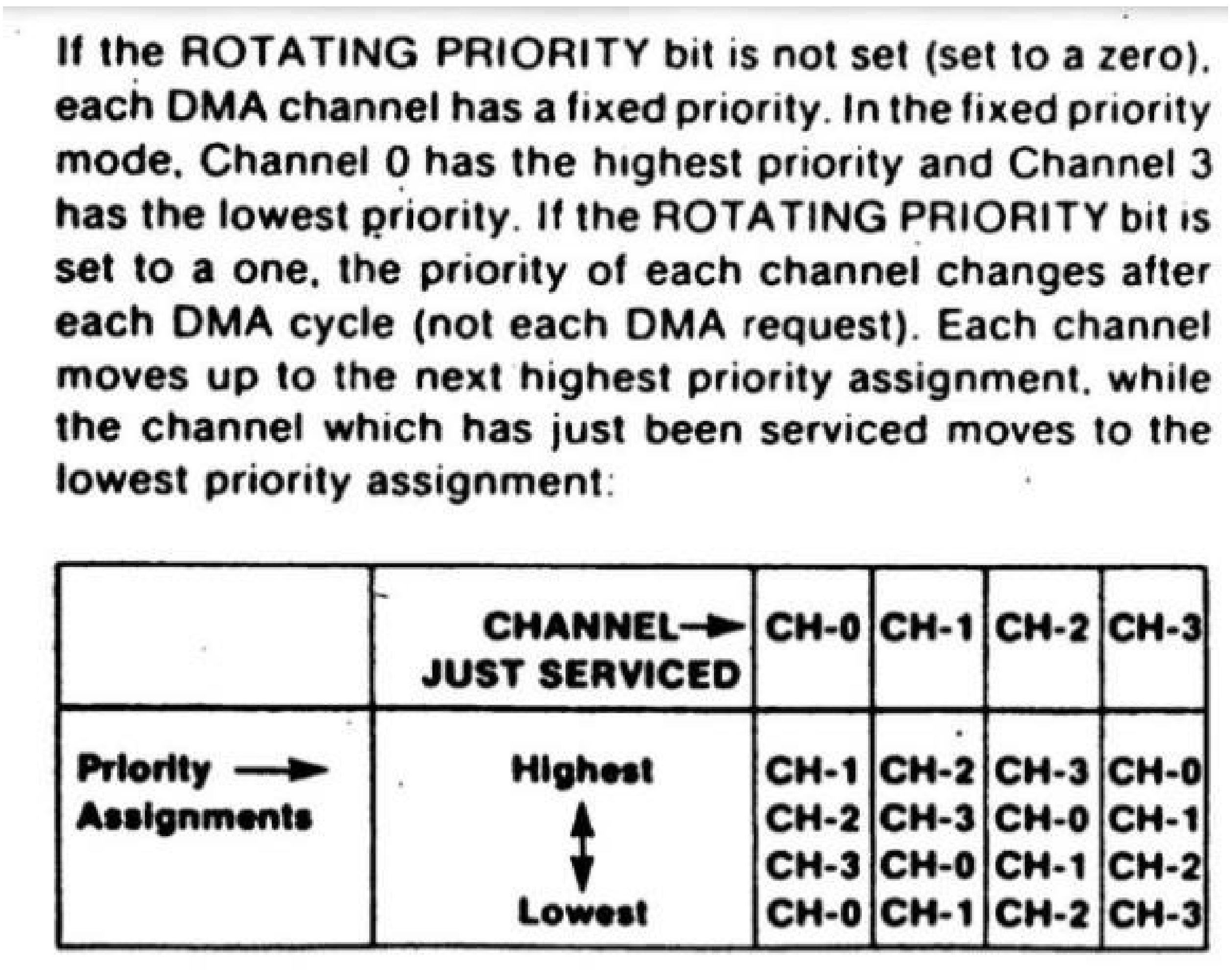


of operation .

Rotating Priority Bit 4

In the Rotating Priority Mode, the priority of the channels has a circular sequence. After each DMA cycle, the priority of each channel changes. The channel which had just been serviced will have the lowest priority.





If the ROTATING PRIORITY bit is not set (set to a zero), each DMA channel has a fixed priority. In the fixed priority mode, Channel 0 has the highest priority and Channel 3 has the lowest priority. If the ROTATING PRIORITY bit is set to a one, the priority of each channel changes after each DMA cycle (not each DMA request). Each channel moves up to the next highest priority assignment, while the channel which has just been serviced moves to the lowest priority assignment:

	CHANNEL→ JUST SERVICED	CH-0	CH-1	CH-2	CH-3
Priority → Assignments	Highest ↕ Lowest	CH-1 CH-2 CH-3 CH-0	CH-2 CH-3 CH-0 CH-1	CH-3 CH-0 CH-1 CH-2	CH-0 CH-1 CH-2 CH-3

- The DMA channel should not be enabled till the DMA address register and the terminal count register **contain valid information**; otherwise, an unwanted DMA request may initiate a DMA cycle, probably destroying the valid memory data.
- **The bits Do - D3 enable one of the four DMA channels of 8257. For example, if Do is = '1', channel 0 is enabled**. If bit 4 is set, rotating priority is enabled, otherwise, the normal, i.e. fixed priority is enabled.

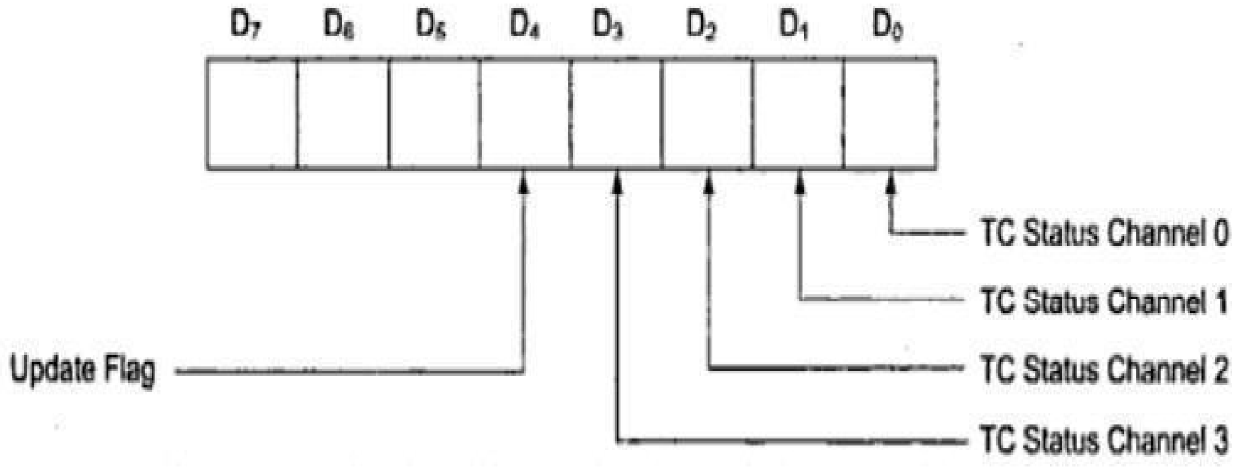
- The DMA channel should not be enabled till the DMA address register and the terminal count register **contain valid information**; otherwise, an unwanted DMA request may initiate a DMA cycle, probably destroying the valid memory data.
- **The bits Do - D3 enable one of the four DMA channels of 8257. For example, if Do is**

= ‘1‘, channel 0 is enabled. If bit 4 is set, rotating priority is enabled, otherwise, the normal, i.e. fixed priority is enabled.

- The auto load bit, if set, **enables channel 2** for the repeat block chaining operations, without immediate software intervention between the two successive blocks.
- The channel 2 registers are used as usual, while the channel 3 registers are used to store the block initialisation parameters, i.e. the DMA starting address and terminal count.

Status Register

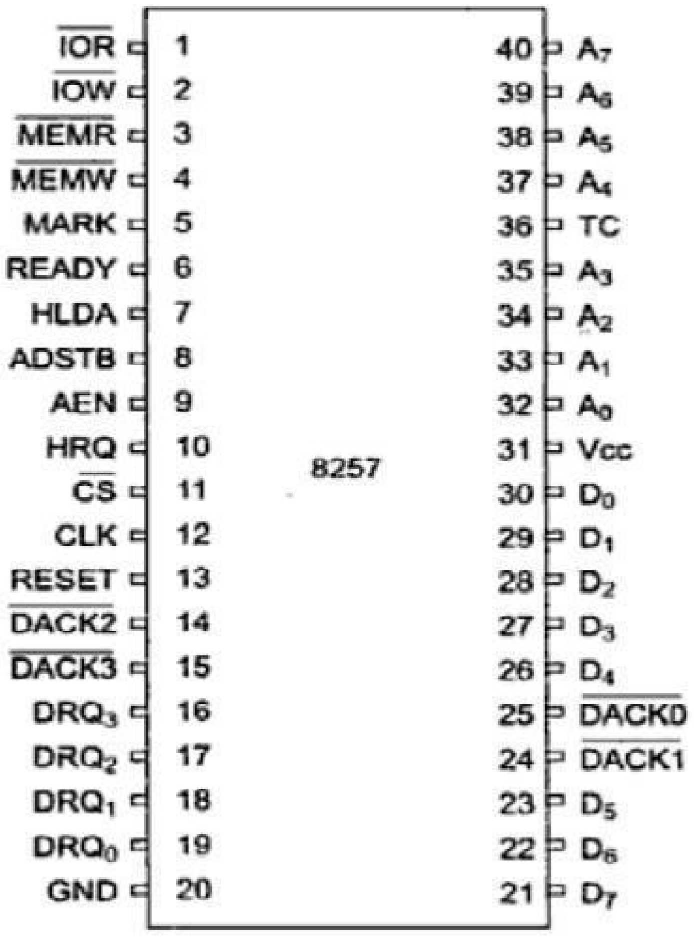
- The status register of 8257 is shown in figure. The lower order 4- bits of this register contain the terminal count status for the four individual channels.
- If any of these **bits is set**, it indicates that the specific **channel has reached the terminal count condition.** These bits remain set till either the status is read by the CPU or the 8257 is reset.



If the update flag is set, the contents of the channel 3 registers are reloaded to the corresponding registers of channel 2 whenever the channel 2 reaches a terminal count condition, after transferring one block and the next block is to be transferred using the auto load feature of 8257.

The update flag is set every time; the channel 2 registers are loaded with contents of the channel 3 registers. It is cleared by the completion of the first DMA cycle of the new block. This register can only read.

Signals/PIN diagram



Signal Description of

8257 DRQo- DRQ3:

These are the four individual channel DMA request inputs, used by the peripheral devices for requesting the DMA services. The DRQo has the highest priority while DRQ**3** has the lowest one, if the fixed priority mode is selected.

DACKo- DACK3:

These are the active- low DMA acknowledge output lines which inform the requesting peripheral that the request has been honoured and the bus is relinquished by the CPU. These lines may act as strobe lines for the requesting devices.

Do- D7:

- These are bidirectional, data lines used to interface the system bus with the internal data bus of 8257. These lines carry command words to

8257 and status word from 8257, in

slave mode, i.e. under the control of CPU. The data over these lines may be transferred in both the directions. When the 8257 is the bus master (master mode, i.e. not under CPU control), it uses Do- D7 lines to send higher byte of the generated address to the latch. This address is further latched using ADSTB signal. the address is transferred over Do- D7 during the first clock cycle of the DMA cycle. During the rest of the period, data is available on the data bus.

IOR:

- This is an active- low bidirectional tristate input line that acts as an input in the slave mode. In slave mode, this input signal is used by the CPU to read internal registers of 8257.This line acts output in master mode. In master mode, this signal is used to read data from a peripheral during a memory write cycle.

IOW:

- This is an active low bidirectional tristate line that acts as input in slave mode 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 a control output that loads the data to a peripheral during DMA memory read cycle (write to peripheral).

CLK:

- This is a clock frequency input required to derive basic system timings for the internal operation of 8257.

RESET:

- This active- high asynchronous input is disables all the DMA channels by clearing the mode register and tri states all the control lines.

Ao- A3:

These are the four least significant address lines. In slave mode, they act as 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.

CS:

This is an active-low chip select line that enables the read/write operations from/to 8257, in slave mode. In the master mode, it is automatically disabled to prevent the chip from getting selected (by CPU) while performing the DMA operation.

A4- A7:

This is the higher nibble of the lower byte address generated by 8257 during the master mode of DMA operation.

READY:

This is an active- high asynchronous input used to stretch memory read and write cycles of 8257 by inserting wait states. This is used while interfacing slower peripherals..

HRQ:

The hold request output requests the access of the system bus.

HLDA:

The CPU drives this input to the DMA controller high, while granting the bus to the device.

This pin is connected to the HLDA output of the CPU. This input, if high, indicates to the DMA controller that the bus has been granted to the requesting peripheral by the CPU.

MEMR

This active –low memory read output is used to read data from the addressed memory locations during DMA read cycles.

MEMW:

This active- low three state output is used to write data to the addressed memory location during DMA write operation.

ADST:

This output from 8257 strobes the higher byte of the memory address generated by the DMA controller into the latches.

AEN:

This output is used to disable the system data bus and the control the bus driven by the CPU, this may be used to disable the system address and data bus by using the enable input of the bus drivers to inhibit the non- DMA devices from responding during DMA operations. If the 8257 is I/O mapped, this should be used to disable the other I/O devices, when the DMA controller addresses is on the address bus.

TC:

If the TC STOP bit in the mode set register is set, the selected channel will be disabled at the end of the DMA cycle. The TC pin is activated when the 14- bit content of the terminal count register of the selected channel becomes equal to zero. The lower order 14 bits of the terminal count register are to be programmed with a 14- bit equivalent of (n- 1), if n is the desired number of DMA cycles.

Vcc:

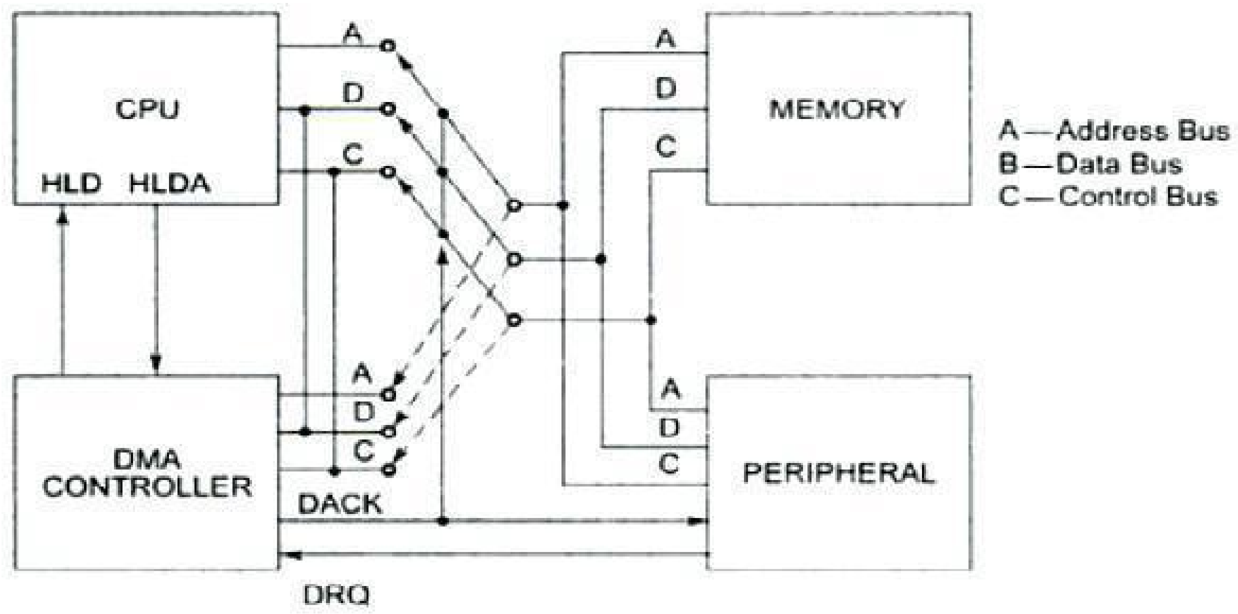
This is a +5v supply pin required for operation of the circuit.

GND:

This is a return line for the supply (ground pin of the IC).

Interfacing 8257 with 8086

- Once a DMA controller is initialized by a CPU property, it is ready to take control of the system bus on a DMA request, either from a peripheral or itself (in case of memory- to memory transfer).
- The DMA controller **sends a HOLD** request to the CPU and waits for the CPU to assert the HLDA signal. The CPU relinquishes the control of the bus before asserting the HLDA signal



Once the HLDA signal goes high, the DMA controller activates the DACK signal to the requesting peripheral and gains the control of the system bus

The DMA controller is the sole master of the bus, till the DMA operation is over.

The CPU remains in the HOLD status (all of its signals are tristate except HOLD and HLDA), till the DMA controller is the master of the bus.

In other words, the DMA controller interfacing circuit implements a **switching arrangement for the address, data and control busses of the memory and peripheral subsystem from/to the CPU to/from the DMA controller.**

Direct Memory Access Controller (DMAC) options for data transfer

The DMA Controller has several options available for the transfer of data. They are:

1) Cycle Steal:

- A read or write signal is generated by the DMAC, and the I/O device either generates or latches the data. The DMAC effectively **steals cycles from the processor in order to transfer the byte**, so **single byte transfer is also known as cycle stealing.**

2) Burst Transfer:

- To achieve block transfers, some **DMAC's incorporate an automatic sequencing of the value presented on the address bus.** A register is used as a byte count, being decremented for each byte transfer, and upon the byte count reaching zero, the DMAC will release the bus. When the DMAC operates in burst mode, the CPU is halted for the duration of the data transfer.

3) Hidden DMA:

- It is possible to perform hidden DMA, which is transparent to the normal operation of the CPU. In other words, the bus is **grabbed by the DMAC when the processor is not using it**. The DMAC monitors the execution of the processor, and when it recognises the processor executing an instruction which has sufficient **empty clock cycles to perform a byte transfer**; it waits till the processor is decoding the op code, then grabs the bus during this time.