Direct Memory Access

- ***Definition**: A direct memory access (DMA) is an operation in which data is copied (transported) from one resource to another resource in a computer system without the involvement of the CPU.*
- The task of a DMA-controller (DMAC) is to execute the copy operation of data from one resource location to another. The copy of data can be performed from:
- I/O-device to memory
- memory to I/O-device
- memory to memory
- I/O-device to I/O-device

A DMA is an independent (from CPU) resource of a computer system added for the concurrent execution of DMA-operations. The first two operation modes are ’read from’ and ’write to’ transfers of an I/O-device to the main memory, which are the common operation of a DMA-controller. The other two operations are slightly more difficult to implement, and most DMA-controllers do not implement device to device transfers.

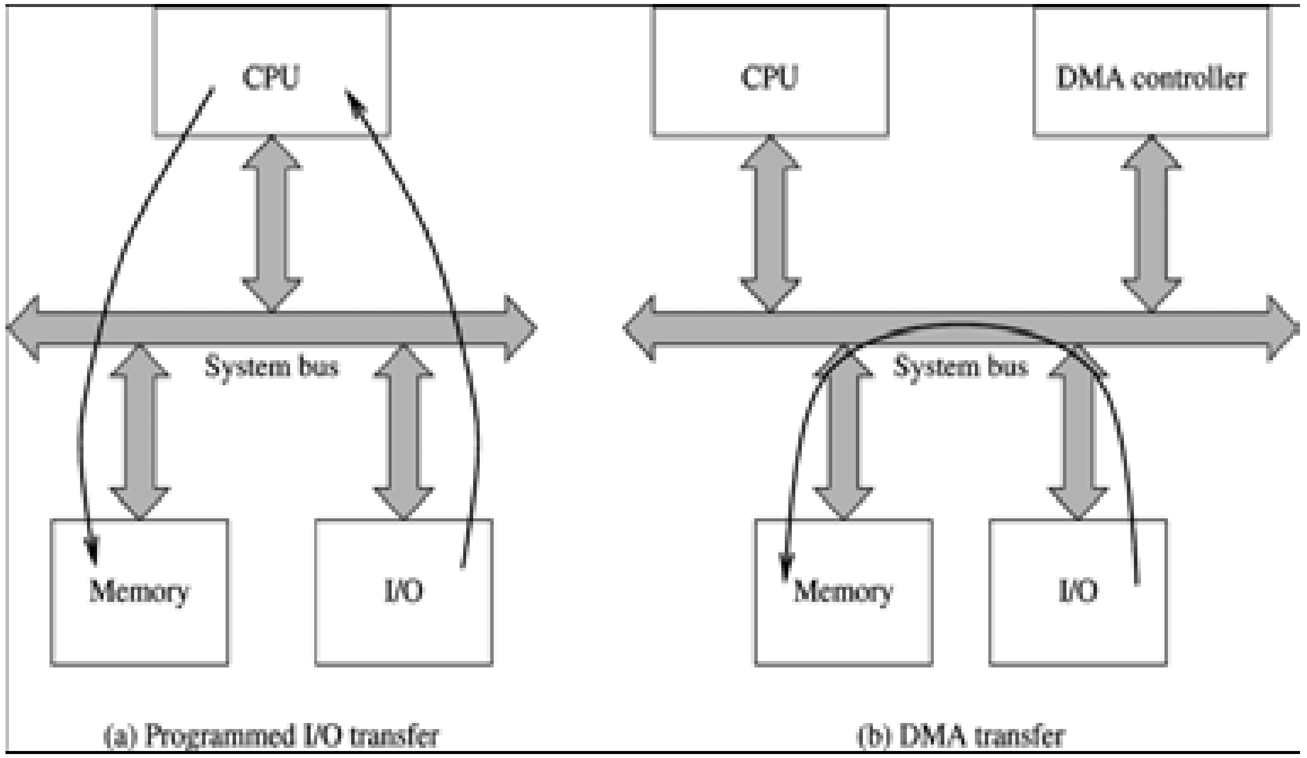


Figure (1)
Computer system with DMA

BASIC DMA OPERATION:

Two control signals are used to request and acknowledge a direct memory access (DMA) transfer in the microprocessor-based system.

- The HOLD pin is an input used to request a DMA action.
- The HLDA pin is an output that acknowledges the DMA action.
- Figure 2 shows the timing that is typically found on these two DMA control pins.

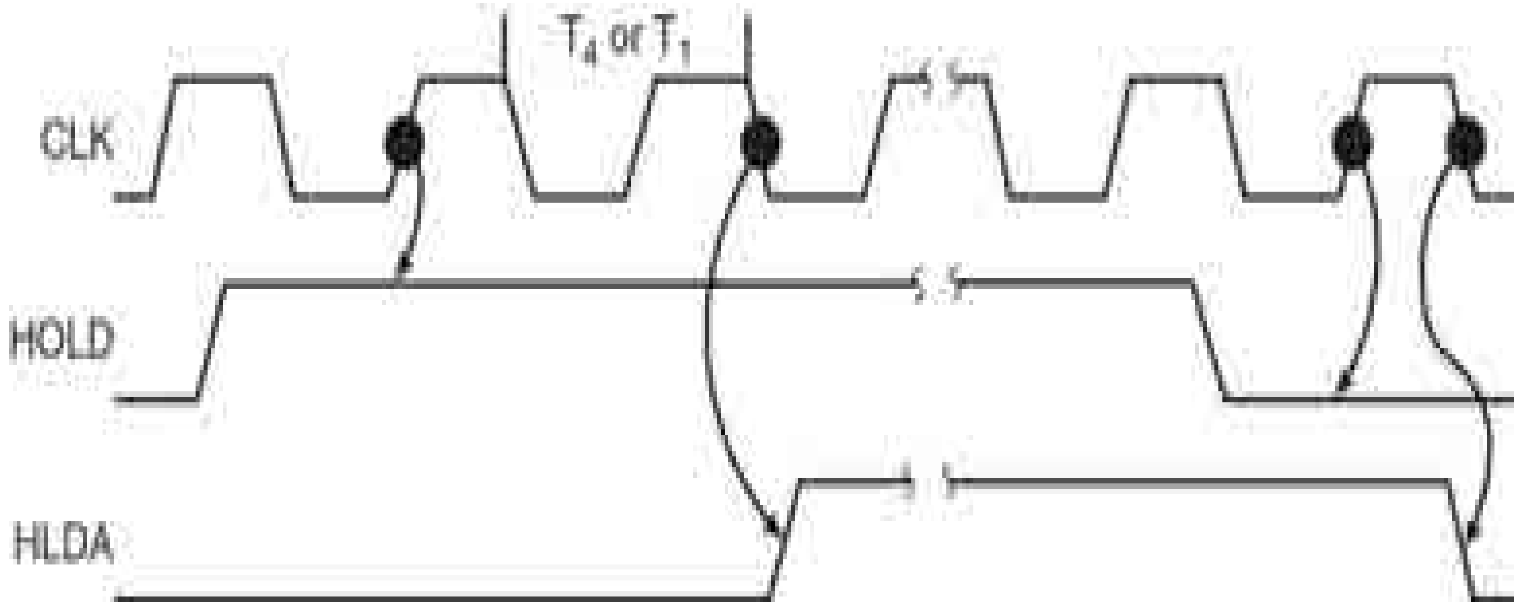


Figure (2)
Timing Diagram of DMA

- HOLD is sampled in any clocking cycle
- when the processor recognizes the hold, it stops executing software and enters hold cycles
- HOLD input has higher priority than INTR or NMI
- The only microprocessor pin that has a higher priority than a HOLD is the RESET pin
- HLDA becomes active to indicate the processor has placed its buses at high-impedance state.
- As can be seen in the timing diagram, there are a few clock cycles between the time that HOLD changes and until HLDA changes

DMA Controller

- A DMA read causes $\overline{MRDC}$ and $\overline{IOWC}$ signals to activate simultaneously.
- A DMA write causes $\overline{MWTC}$ and $\overline{IORC}$ signals to both activate.
- 8086 require a controller or circuit such as shown in Figure 3 for control bus signal generation.
- The DMA controller provides memory with its address, and controller signal ($\overline{DACK}$) selects the I/O device during the transfer.

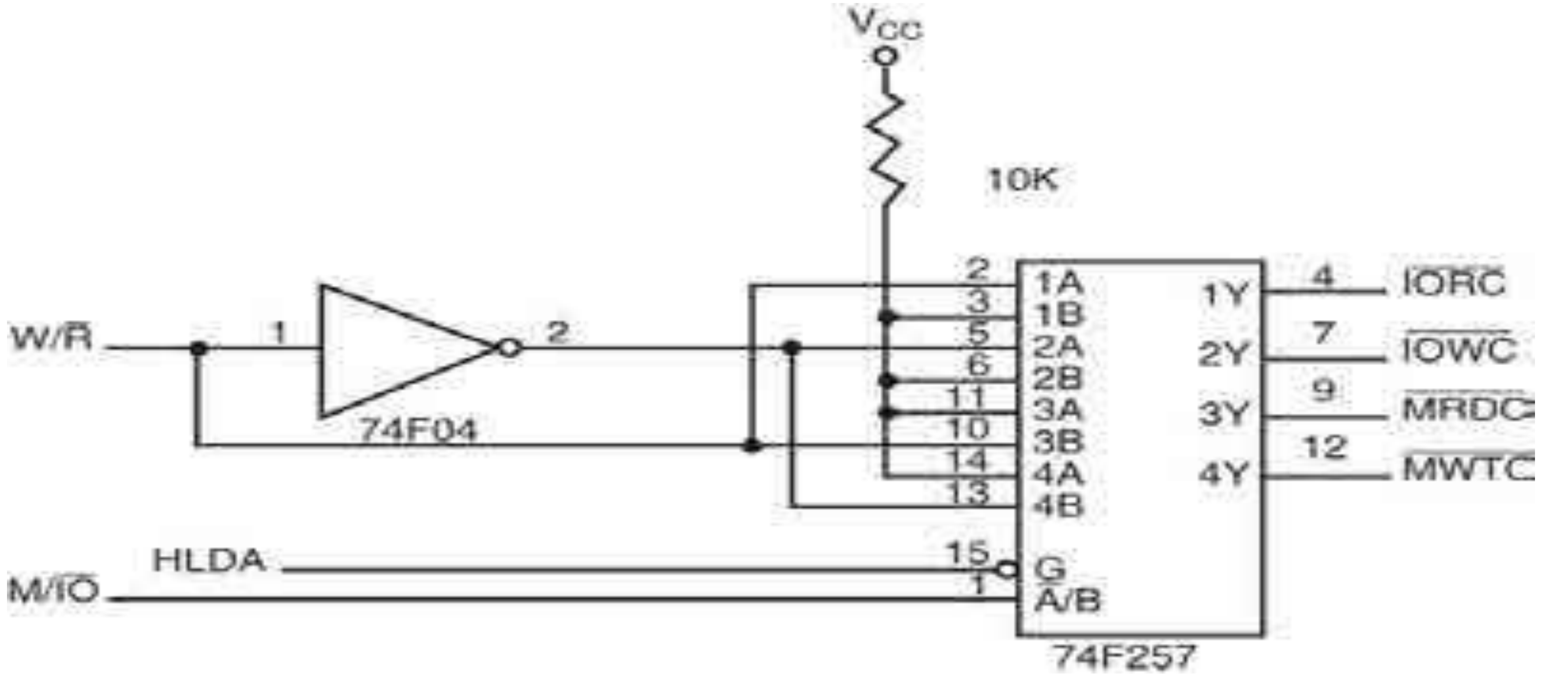


Figure (3)
A circuit that generates system control signals in a DMA environment

Data transfer speed is determined by speed of the memory device or a DMA controller.

- if memory speed is 50 ns, DMA transfers occur at rates up to 1/50 ns or 20 M bytes per second
- if the DMA controller functions at a maximum rate of 15 MHz with 50 ns memory, maximum transfer rate is 15 MHz because the DMA controller is slower than the memory
- In many cases, the DMA controller *slows* the speed of the system when transfers occur.

DMA controller details

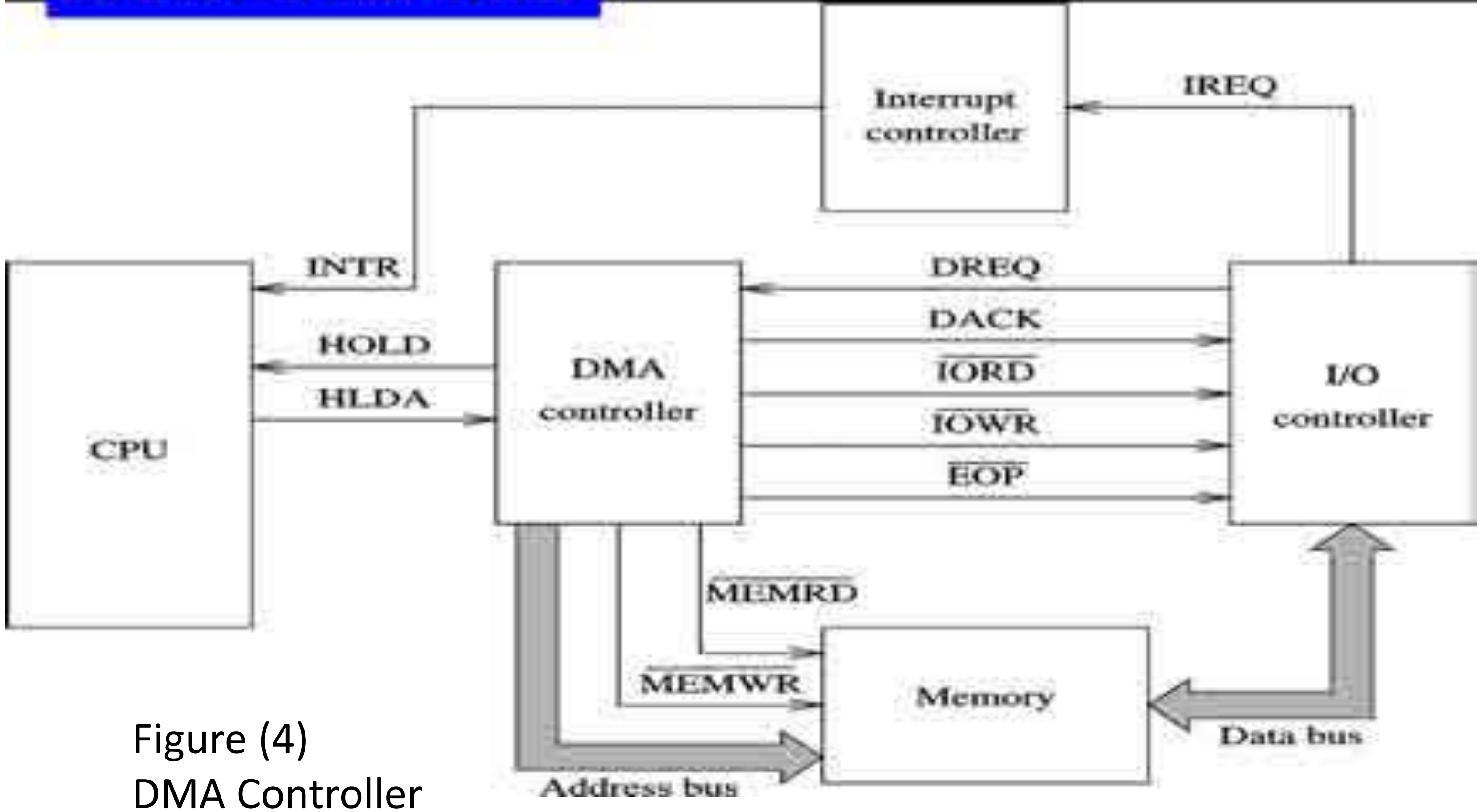


Figure (4)
DMA Controller

Internal Configuration of DMA

The DMA controller includes several registers:-

- The DMA **Address Register** contains the memory address to be used in the data transfer. The CPU treats this signal as one or more output ports.
- The DMA **Count Register**, also called Word Count Register, contains the no. of bytes of data to be transferred. Like the DMA address register, it too is treated as an O/P port (with a diff. Address) by the CPU.
- The DMA **Control Register** accepts commands from the CPU. It is also treated as an O/P port by the CPU.
- Although not shown in figure (5), most DMA controllers also have a Status Register. This register supplies information to the CPU, which accesses it as an I/P port.

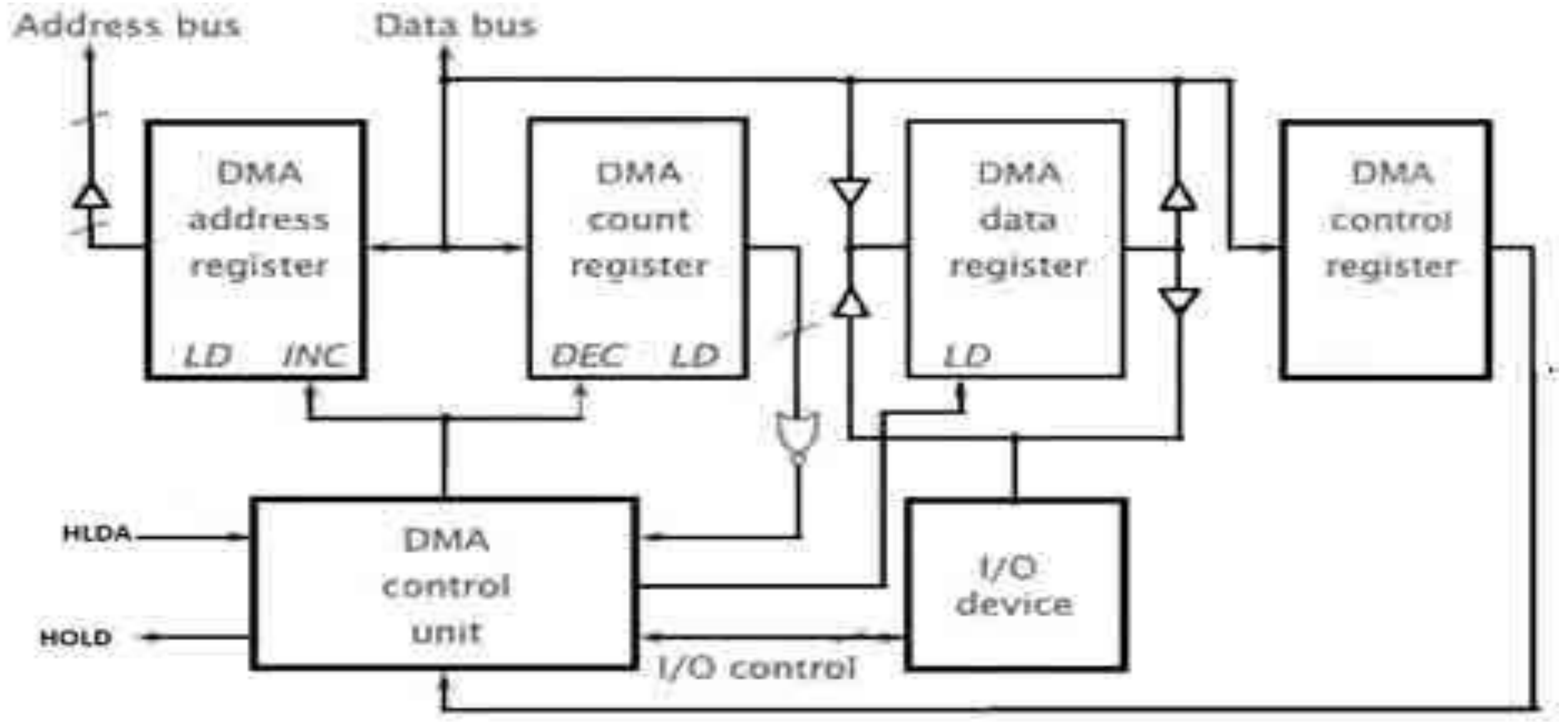


Figure (5)
Internal Configuration of DMA Controller

Process of DMA Transfer

To initiate a DMA transfer, the CPU loads the address of the first memory location of the memory block (to be read or written from) into the DMA address register. It does his via an I/O output instruction, such as the OTPT instruction for the relatively simple CPU.

- It then writes the no. of bytes to be transferred into the DMA count register in the same manner.
- Finally, it writes one or more commands to the DMA control register
- These commands may specify transfer options such as the DMA transfer mode, but should always specify the direction of the transfer, either from I/O to memory or from memory to I/O.
- The last command causes the DMA controller to initiate the transfer. The controller then sets HOLD to 1 and, once HLDA becomes 1, seizes control of the system buses

DMA Transfer Modes

1. BURST mode

■ Sometimes called **Block Transfer** Mode.

■ An entire block of data is transferred in one contiguous sequence. Once the DMA controller is granted access to the system buses by the CPU, it transfers all bytes of data in the data block before releasing control of the system buses back to the CPU.

2. CYCLE STEALING Mode

■ Viable alternative for systems in which the CPU should not be disabled for the length of time needed for Burst transfer modes.

3. TRANSPARENT Mode

- This requires the most time to transfer a block of data, yet it is also the most efficient in terms of overall system performance

Advantages of DMA

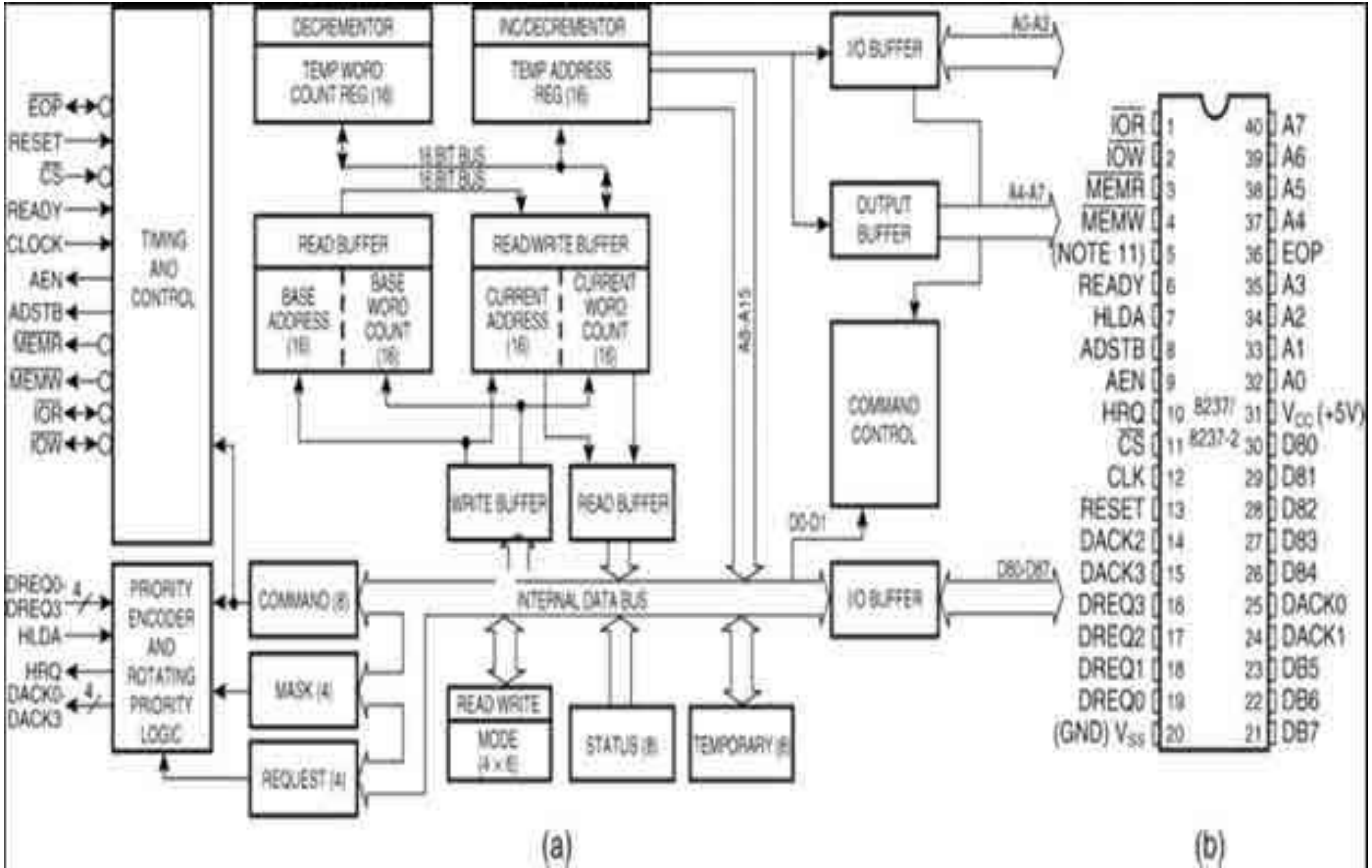
- Computer system performance is improved by direct transfer of data between memory and I/O devices, bypassing the CPU.
- CPU is free to perform operations that do not use system buses.

Disadvantages of DMA

- In case of Burst Mode data transfer, the CPU is rendered inactive for relatively long periods of time.

THE 8237 DMA CONTROLLER

- The 8237 supplies memory & I/O with control signals and memory address information during the DMA transfer.
- Actually a special-purpose microprocessor whose job is high-speed data transfer between memory and I/O
- Figure 6 shows the pin-out and block diagram of the 8237 programmable DMA controller.


DECREMENTOR
TEMP WORD COUNT REG (16)
INC/DECREMENTOR
TEMP ADDRESS REG (16)
I/O BUFFER
A0-A3
OUTPUT BUFFER
A4-A7
16 BIT BUS
16 BIT BUS
READ BUFFER
BASE ADDRESS (16)
BASE WORD COUNT (16)
READ/WRITE BUFFER
CURRENT ADDRESS (16)
CURRENT WORD COUNT (16)
A8-A15
COMMAND CONTROL
EOP
RESET
CS
READY
CLOCK
AEN
ADSTB
MEMR
MEMW
IOR
IOW
TIMING AND CONTROL
WRITE BUFFER
READ BUFFER
D0-D1
DREQ0-DREQ3
4
HLDA
HRQ
DACK0-DACK3
4
PRIORITY ENCODER AND ROTATING PRIORITY LOGIC
COMMAND (8)
MASK (4)
REQUEST (4)
INTERNAL DATA BUS
I/O BUFFER
DB0-DB7
READ WRITE
MODE (4 × 6)
STATUS (8)
TEMPORARY (8)
IOR 1
IOW 2
MEMR 3
MEMW 4
(NOTE 11) 5
READY 6
HLDA 7
ADSTB 8
AEN 9
HRQ 10
CS 11
CLK 12
RESET 13
DACK2 14
DACK3 15
DREQ3 16
DREQ2 17
DREQ1 18
DREQ0 19
(GND) VSS 20
40 A7
39 A6
38 A5
37 A4
36 EOP
35 A3
34 A2
33 A1
32 A0
31 VCC (+5V)
30 DB0
29 DB1
28 DB2
27 DB3
26 DB4
25 DACK0
24 DACK1
23 DB5
22 DB6
21 DB7
8237/8237-2
(a)
(b)