

UNIT-3Architecture of 8086 and Interfacing:-▷ Pin diagram of 8086 MP :-

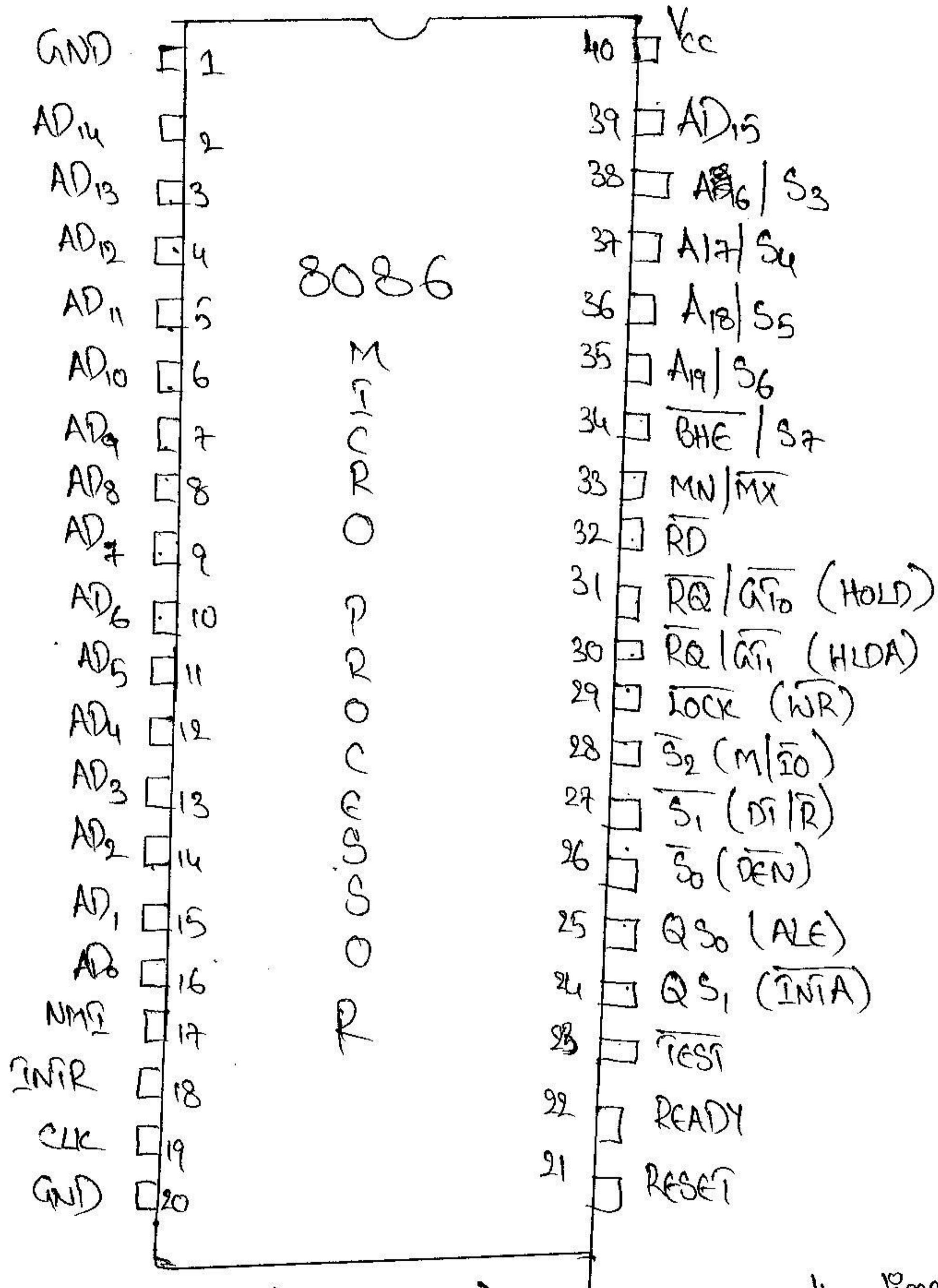
- ① The 8086 MP is a 16-bit CPU available in three clock rates 5, 8, and 10 MHz.
- ② The 8086 MP is available in 40-Pin DIP (Dual In-line Package) where 8086 MP can operate in single Processor (or) Multiprocessors configuration to achieve high performance.
- ③ Single Processor Mode is called Minimum mode, which has only one Bus master with few peripherals such as latches, transceivers, memory and I/O devices.
- ④ The Multiprocessor mode is called Maximum mode which has more than one microprocessor (or) processor in the system. Any one of them gain control over a system and acts as Bus Master.

The 8086 MP signals can be categorized in 3-graph:-

- ① Signals having common functions in minimum as well as maximum mode.
- ② Signals having special functions for minimum mode.
- ③ Signals " " " Maximum mode.

① Signals having common functions in minimum as well as maximum mode:-

- a) AD<sub>0</sub> - AD<sub>15</sub> (Address/Data bus) : These are the time multiplexed memory I/O address and data lines. Address remains on the line during  $T_1$  clock cycle of bus cycle, while data is available on data bus during  $T_2$ ,  $T_3$ ,  $T_W$ , and  $T_4$ . There are bidirectional lines.



b) A<sub>16</sub> | S<sub>3</sub> - A<sub>9</sub> | S<sub>6</sub> (Address | Status lines):- There are the time multiplexed address and status lines. During T<sub>1</sub>, there are most significant memory operations. During T<sub>0</sub> operations, these lines are low. During memory (or) I/O operations, status information is available on those lines for T<sub>2</sub>, T<sub>3</sub>, T<sub>W</sub> and T<sub>4</sub>. The status lines S<sub>6</sub> is always low. The status line S<sub>5</sub> indicates status of interrupt enable flag bit and it is updated at beginning of each clock cycle. The S<sub>4</sub> and S<sub>3</sub> combinedly indicate which segment register is presently being used for memory access.

S<sub>4</sub>    S<sub>3</sub>    Segment Register Employed

0    0    ES

0    1    SS

1    0    CS (or) none

1    1    DS

c) BHE (Bus High Enable) :-

It is a time multiplexed line. It remains active high whether a byte (or) word will be transferred from/to memory location, which is BHE in conjunction with 'A<sub>0</sub>'.

<u>BHE</u>	<u>A<sub>0</sub></u>	<u>Indication</u>
0	0	whole word
0	1	upper byte from odd address
1	0	lower byte from/to even address
1	1	None.

d) RD (Read Output) :- It is a read signal which is active low signal. It indicates that the Processor is performing read operation which memory | IO :-

e) READY (Input) :- It is active high signal. When it is high it indicates Peripheral device Ready to transfer data. It is an acknowledgement received from the address memory | IO.

f) RESET (Input) :- It is a active high signal. This signal causes the processor to terminate the current activity and start execution from FFFF0H.

g) CLK (Clock Input) :- It provides basic timing for Processor operation and bus control activity. It is asymmetric square wave with 33% duty cycle. The range of frequency for different 8086 versions is 5MHz, 8MHz & 11MHz.

4

- b) TEST (Input):- This I/O is tested by WAIT instruction. The 8086 will enter a wait state after execution of wait " and enter to execution.

It is used to interface with Peripheral devices

$\hookrightarrow$  OV  $\rightarrow$  interfacing is done correctly.

$\hookrightarrow$  SV  $\rightarrow$  " " not done . "

- c) INTR (Interrupt Request) Input:- This is level triggered I/O. This is sampled during the last clock cycle of each instruction to determine the availability of the request.

If an interrupt request is pending, the processor enters the interrupt acknowledge cycle. This can be initially masked by enable interrupt flag (IF).

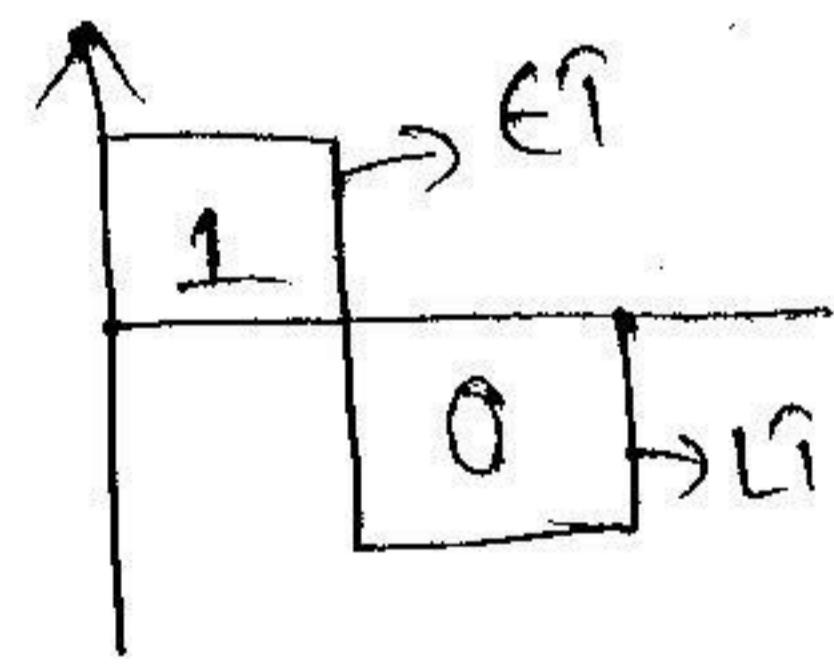
$\rightarrow$  To stop execution of any program we give SV to I/O.

- d) NMI (Non-Maskable Interrupt) I/O:- This is an edge triggered I/O which causes a type 2 interrupt.

It is used to interrupt for only one functionality when one is interfaced with peripheral device.

Edge trigger (EI)  $\rightarrow$  during execution (0-1) (1-0)

Level trigger (LI)  $\rightarrow$  doing " 1st Executed only at level



- e) V<sub>cc</sub>:- +5V Power supply for the operation of the internal circuit

f) GND:- Ground for internal circuit

- g) MN/MX (I/O):- used to select minimum (or) maximum mode.

When SV is applied we can select max mode.  
 " OV " " " " min mode.

## ② Signals having special functions for minimum mode :-

- i) INIA (Interrupt Acknowledge) o/p :- This signal is used to send strobe for interrupt acknowledge cycle. When it goes low, it means that the Processor has accepted the interrupt. It is active low during  $t_2, t_3$ , &  $t_6$  of each interrupt acknowledge cycle.
- ii) ALE (Address Latch Enable) o/p :- This o/p signal indicates the availability of valid address on Address/data lines, and is connected to latch enable o/p of latches. It is used to demultiplex the address and data lines.
- iii) DEN (Data Enable) o/p :- It indicates availability of valid data over the address/data lines. It is used to enable transceiver (bidirectional buffers) to separate the data from the multiplexed address/data signals. It is active from middle  $t_2$  until the middle of  $t_4$ .
- iv). DR/R (Data Transmit/Receive) o/p :- The o/p is used to decide the direction of data flow through the transceiver (bidirectional buffers) when the Processor sends out data, this signal is high and is receiving data, the signal is low.
- v) M/I<sub>O</sub> - memory/I<sub>O</sub> (o/p) :- It is used by Processor to distinguish a memory access from an I/O access. When this signal is high memory is accessed. When this " " low and I/O device is accessed.
- vi) WR (write) o/p :- When this signal is low, the Processor performs memory write (or) I/O write operation depending on states of M/I<sub>O</sub> signal.

(Vii) HOLD (Op) :- It is active high signal, which is op signal to the Processor from other bus master to request the control of the bus. ⑥

(Viii) HOLDA (Hold Acknowledge) op :- It is an active high signal on receiving HOLD signal, the Processor is use a HOLD acknowledge signal through this pin, to the master requesting the control of bus.

③ Signals having special functions for maximum modes :-

(i) QS<sub>1</sub>, QS<sub>0</sub> (Queue Status) op :- These two op signals reflect the status of instruction queue.

<u>QS<sub>1</sub></u>	<u>QS<sub>0</sub></u>	<u>Indication</u>
0	0	No operation.
0	1	1 <sup>st</sup> byte of op code from the queue
1	0	Empty queue
1	1	Subsequent byte from queue.

These signals provide status to allow external tracking of the instruction queue.

(ii) S<sub>2</sub>, S<sub>1</sub>, S<sub>0</sub> (Status lines) op :- These are used by 8288 bus controller to generate all memory and I/O access control signals.

Machine Cycle.

<u>S<sub>2</sub></u>	<u>S<sub>1</sub></u>	<u>S<sub>0</sub></u>	
0	0	0	Interrupt acknowledge
0	0	1	I/O READ
0	1	0	I/O WRITE
0	1	1	HALT
1	0	0	OPCODE FETCH
1	0	1	MEMORY READ
1	1	0	MEMORY write
1	1	1	PASSIVE (d) INACTIVE,

iii) Lock (Op) :- A Prefix instruction activates Lock Signal. (7)  
It is an active low signal. It remains active until the completion of next instruction. When it goes low all interrupted are masked and HOLD request is not granted. Consequently other devices do not get control over the system bus while Lock is low.

iv) RQ | GR<sub>IO</sub>, RQ | GR<sub>I</sub> (Request Grant) :- These pins are used by other local bus masters, in maximum mode.  
Each of this pins is bidirectional with RQ | GR<sub>IO</sub> having higher priority than RQ | GR<sub>I</sub>. After receiving hold request, the processor sends acknowledge signal through these signals.

→ Minimum Mode \* 8086 System and Pinings :-

The 8086 is operated in minimum mode by stopping its MN/MX pin to logic 1.  
In this mode all the control signals are given out by microprocessor chip itself. There are single processors in minimum mode system.

The remaining components in system are latches, clock generators, transceivers, memory and I/O devices.

\*) The latches are generally buffered D-type flipflops like 74LS373 (or) 8282. They are used for separating the valid addresses from multiplexed address/data signals and are controlled by ALE signal generated by 8086.

(8)

- \* Transceivers are the bidirectional buffers and sometimes they are called data amplifiers, which are controlled by 2-signals  $\overline{DEN}$  and  $\overline{DIR}$ 
  - The  $\overline{DEN}$  indicates valid data is available on data bus.
  - The  $\overline{DIR}$  indicates direction of data ie from/to Processor.
- \* The system contains memory for monitor and user program storage. usually EEPROMS are used for monitor storage and RAM's for user program storage.
- \* A system may contain I/O devices for communication with the Processor as well as some special purpose I/O devices.
- \* The clock generator also synchronizes some external signals with system clock.

→ Since it has 20 address lines and 16-data lines, the 8086 CPU requires 3-octal address latches and 2-octal data buffers for complete address and data separation.

The timing diagram can be categorized in two parts.

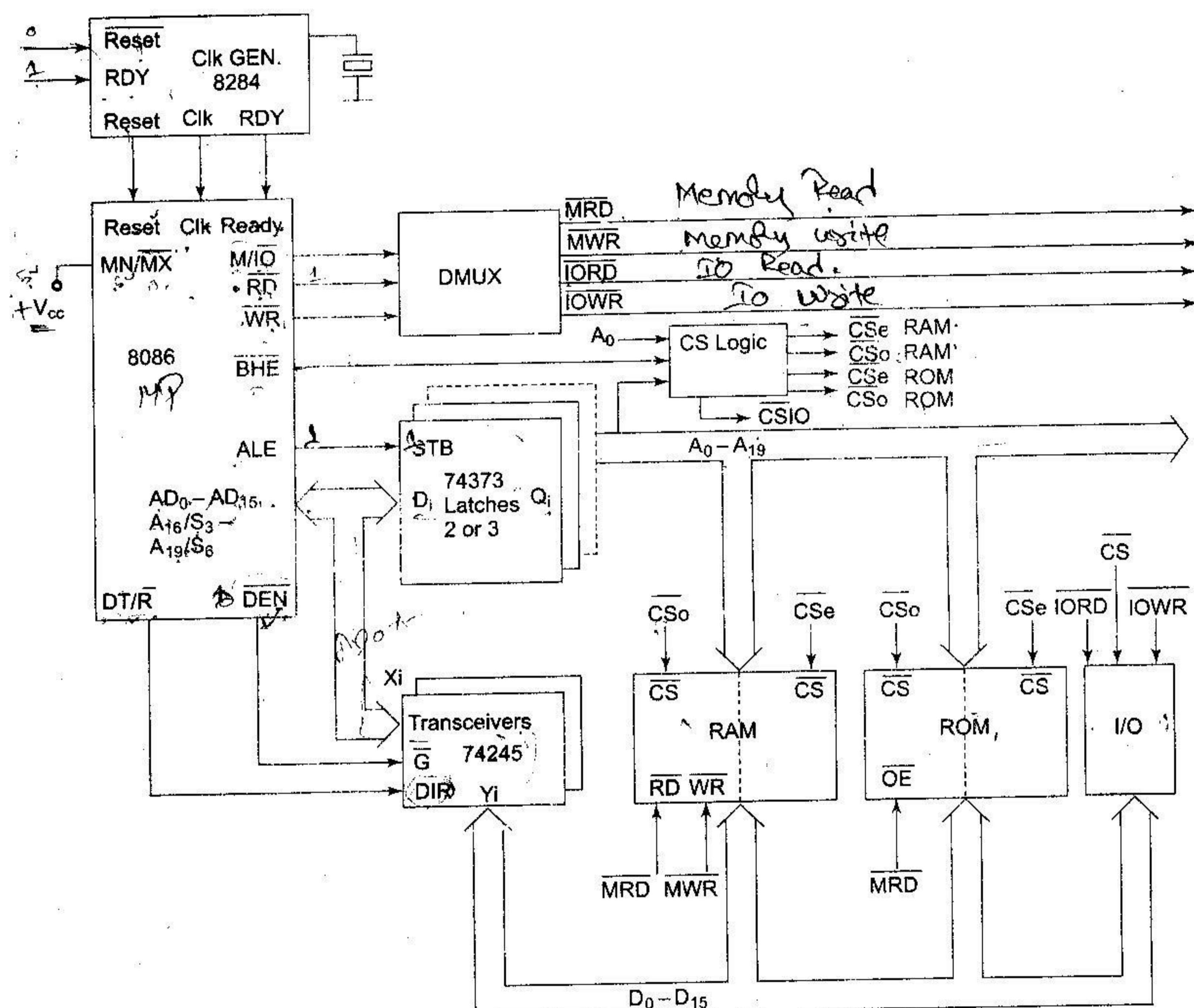
- i) Timing diagram for Read cycle
- ii) " " " write "

i) The Read cycle begins in  $t_1$  with the assertion of Address latch Enable (ALE) signal and  $M/\overline{IO}$  signal. During the negative going edge of this signal, the valid address is latched on local bus.

The BHE & A<sub>0</sub> signals address low, high (or) bytes.

From  $t_1$  to  $t_4$ , the  $M/\overline{IO}$  signal indicates a memory / I/O operation.

— The Processors: 8086/8088—Architectures, Pin Diagrams and Timing Diagrams —



**Fig. I.13 Minimum Mode 8086 System**

Figure 1.14(a) shows the read cycle while Fig. 1.14(b) shows the write cycle.

### I.8.1 HOLD Response Sequence

The HOLD pin is checked at the end of each bus cycle. If it is received active by the processor before T<sub>4</sub> of the previous cycle or during T<sub>1</sub> state of the current cycle, the CPU activates HLDA in the next clock cycle and for the succeeding bus cycles, the bus will be given to another requesting master. The control of the bus is not regained by the processor until the requesting master does not drop the HOLD pin low. When the request is dropped by the requesting master, the HLDA is dropped by the processor at the trailing edge of the next clock, as shown in Fig. 1.14 (c). The other conditions have already been discussed in the signal description section for the HOLD and HLDA signals.

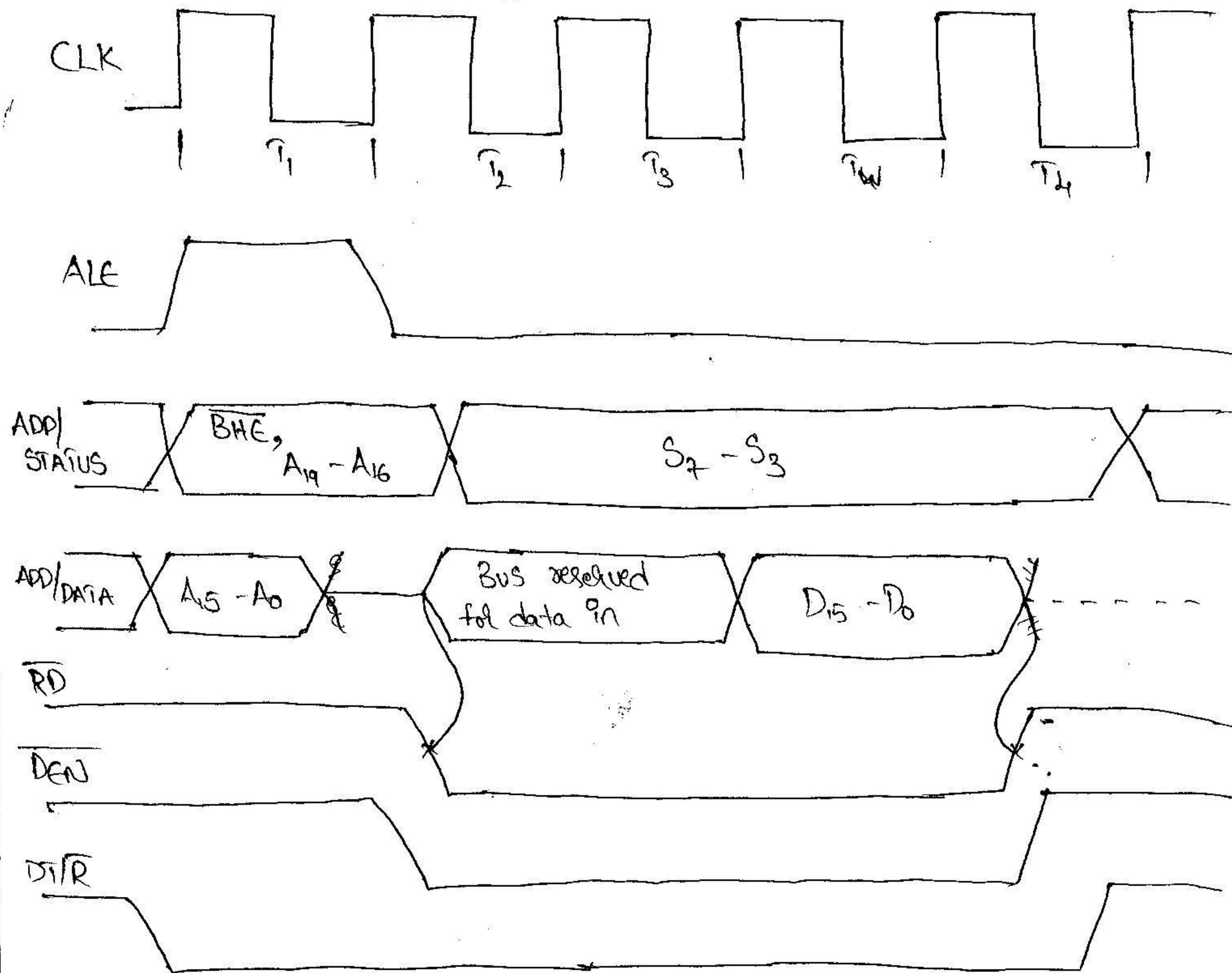
At  $t_2$ , the address is removed from local bus and it is sent to op. The bus is then tri-stated. The Read (RD) Control signal is also activated in  $t_2$ . The signal causes the addressed device to enable its data bus drivers.

~~After (RD) control signal is also activated at  $t_2$ .~~

After  $(\bar{RD})$  goes low, the valid data is available on the data bus. The addressed device will drive the READY line high. When the processor returns the read signal to high level the addressed device will again tri-state its bus drivers.

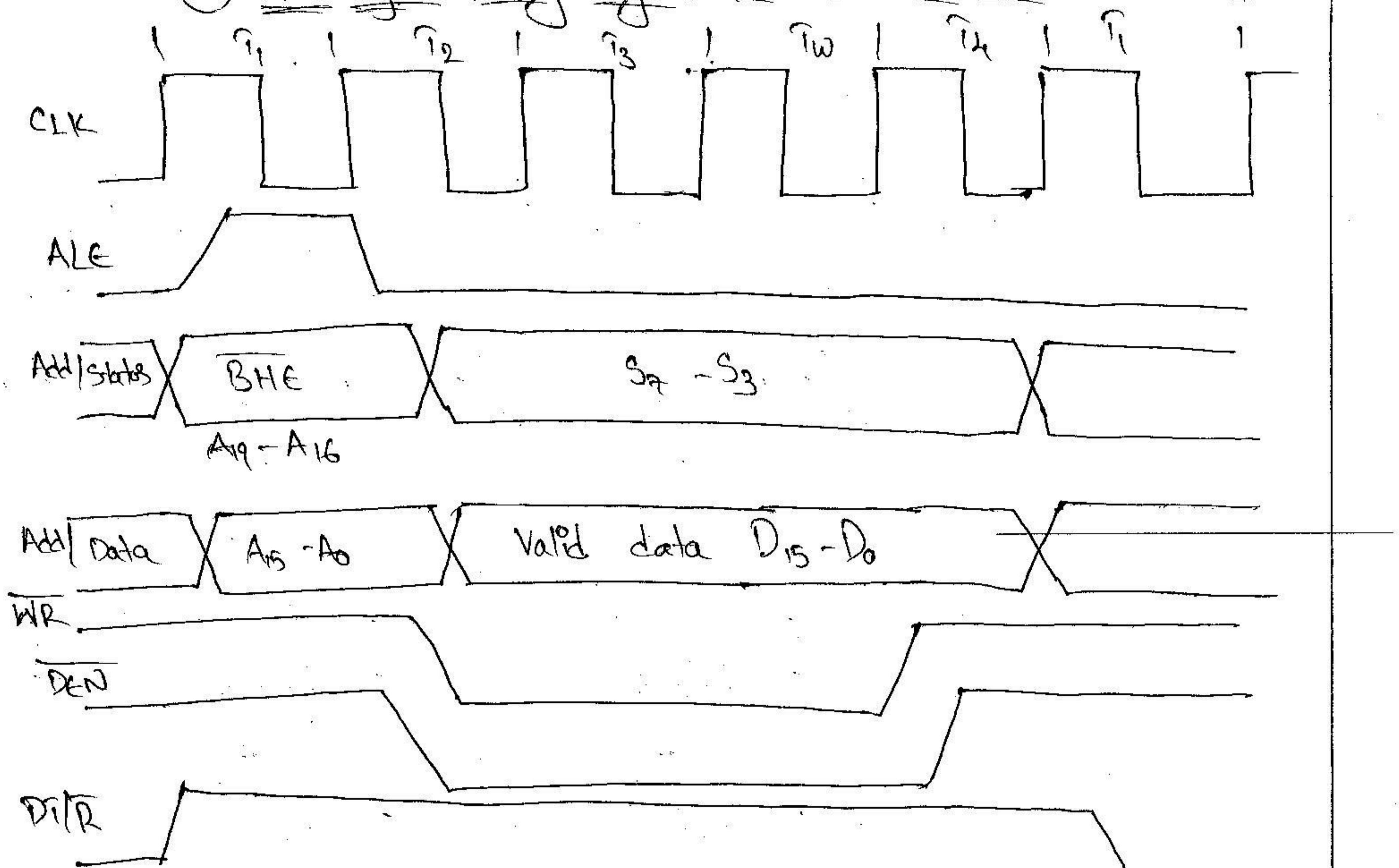
\* CS logic indicates chip select logic and 'e' and 'o'. Suffixes indicate even and odd address memory banks.

### a) Read Cycle Timing Diagram for minimum mode.



(b) Write Cycle Timing Diagram for minimum mode:

ii



ii) The write cycle begins with ALE and emission of the address. The M<sub>15</sub> signal is again asserted to indicate a memory (or) I/O operation. In T<sub>1</sub> after sending the address in T<sub>1</sub>, the processor sends the data to be written to the addressed location. The data remains on the bus until the middle of T<sub>4</sub> state. The WR becomes active at the beginning of T<sub>2</sub> (unlike RD it is some what delayed in T<sub>2</sub> to provide time for floating).

The BHE and A<sub>0</sub> signals are used to select the word byte or bytes of memory (or) I/O word to be read (or) written. The RD and WR signals indicates type of data transfer.

<u>M<sub>15</sub></u>	<u>RD</u>	<u>WR</u>	<u>transferred type</u>
0	0	1	I/O Read
0	1	0	I/O write
1	0	1	Memory read
,	,	0	Memory write

### 3) Maximum Mode 8086 System & Timings:

The Maximum mode, 8086 CPU is operated by grounding the MN/MX pin to ground. In this mode the Processor drives the status signals  $\overline{S_2}$ ,  $\overline{S_1}$ , and  $\overline{S_0}$ .

- \*) The bus controller derives the control signals using the status information.
- \*) In maximum mode, there may be more than one (or) more processors in system configuration. The other components in the system are the same as in minimum mode system.

\*) The basic functions of the bus controller chip 8288 is to drive control signals like RD and WR (for memory and I/O devices),  $\overline{DEN}$ ,  $\overline{D/I/R}$ , ALE etc, using the information made available by processor on status pins.

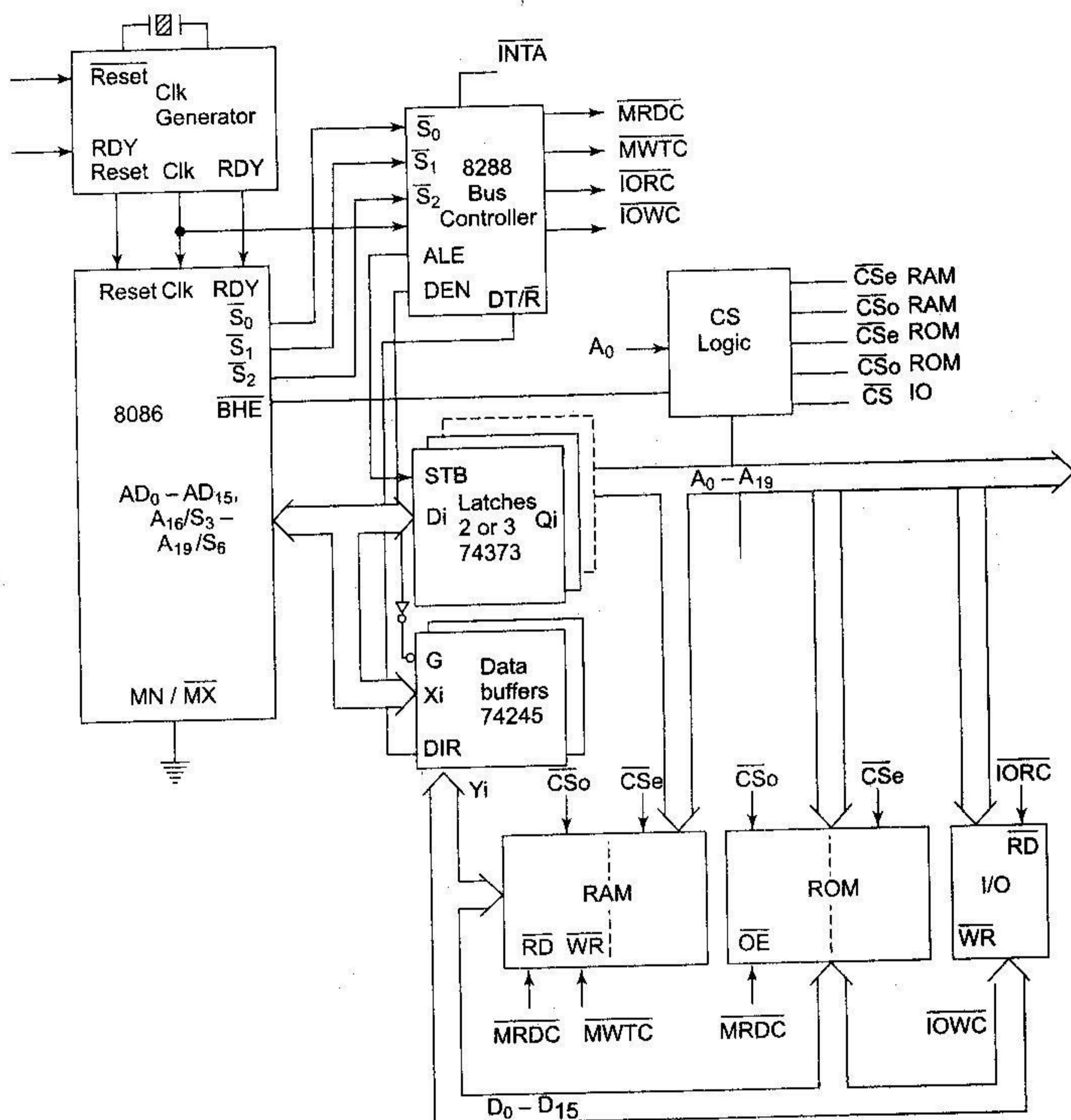
\*) The Bus Controller chip has op lines  $\overline{S_2}$ ,  $\overline{S_1}$ , and  $\overline{S_0}$  and CLK, which are driven by the CPU. It drives op ALE,  $\overline{DEN}$ ,  $\overline{D/I/R}$ ,  $\overline{MRDC}$ ,  $\overline{MWIC}$ ,  $\overline{AMWC}$ ,  $\overline{IORC}$ ,  $\overline{IOWC}$ , and  $\overline{AIOWC}$ .

\*) The  $\overline{AEN}$ ,  $\overline{IOB}$  and CEN pins are specially useful for multiprocessor systems.  $\overline{AEN}$  and  $\overline{IOB}$  are generally grounded. CEN pin is usually tied to +5V. The significance of MEC/ $\overline{PDEN}$  op depends upon state of the  $\overline{IOB}$  pin.

If  $\overline{IOB}$  is grounded it acts as master cascade enable to cascade 8259A. Else it acts as peripheral data used in multiple bus configuration. The  $\overline{INIA}$  pin used to issue two interrupt acknowledge pulses to the interrupt controller (8259) to an interrupting device.

TORC, IOWC are To Read Command and To write Command signals respectively. These signals enables an I/O interface to read (or) write the data from or to the addressed Port.

The MRDC, MWIC are memory read Command and memory write Command signals instruct the memory to accept or send data to or from the bus. For both of these write Command signals the advanced signals namely AIOWC and AMWIC are available. They also serve the same purpose, but are activated one clock cycle earlier than the IOWC and MWIC signals.



**Fig. 1.15 Maximum Mode 8086 System**

The maximum mode system timing diagrams are also divided in two portions as read (input) and write (output) timing diagrams. The address/data and address/status timings are similar to the minimum mode. ALE is asserted in  $T_1$ , just like minimum mode. The only difference lies in the status signals used and the available control and advanced command signals. Figure 1.16 (a) shows the maximum mode timings for the read operation while the Fig. 1.16 (b) shows the same for the write operation. The CS Logic block represents chip select logic and the 'e' and 'O' suffixes indicate even and odd address memory bank.

The Processors: 8086/8088—Architectures, Pin Diagrams and Timing Diagrams

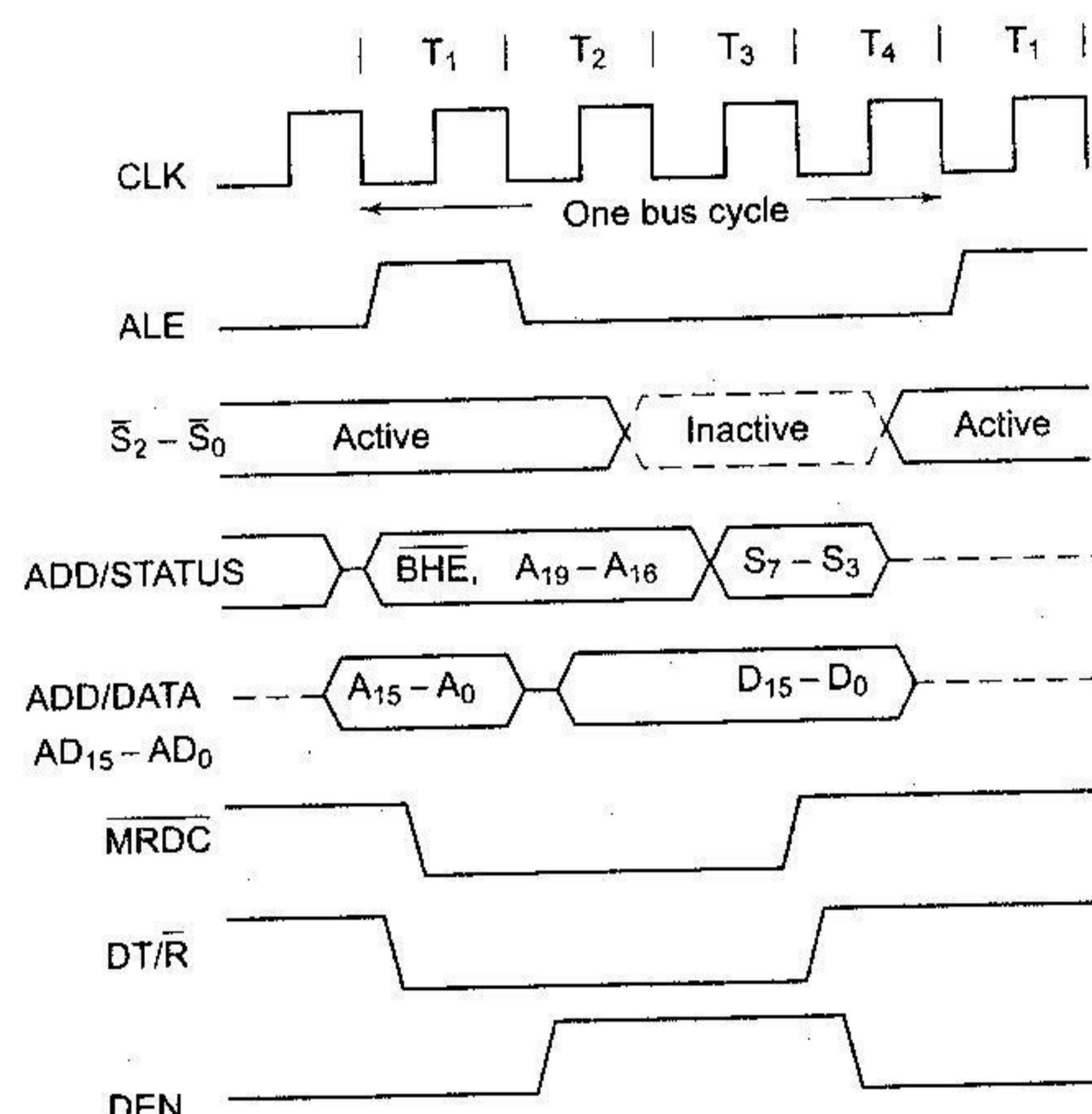


Fig. I.16 (a) Memory Read Timing in Maximum Mode

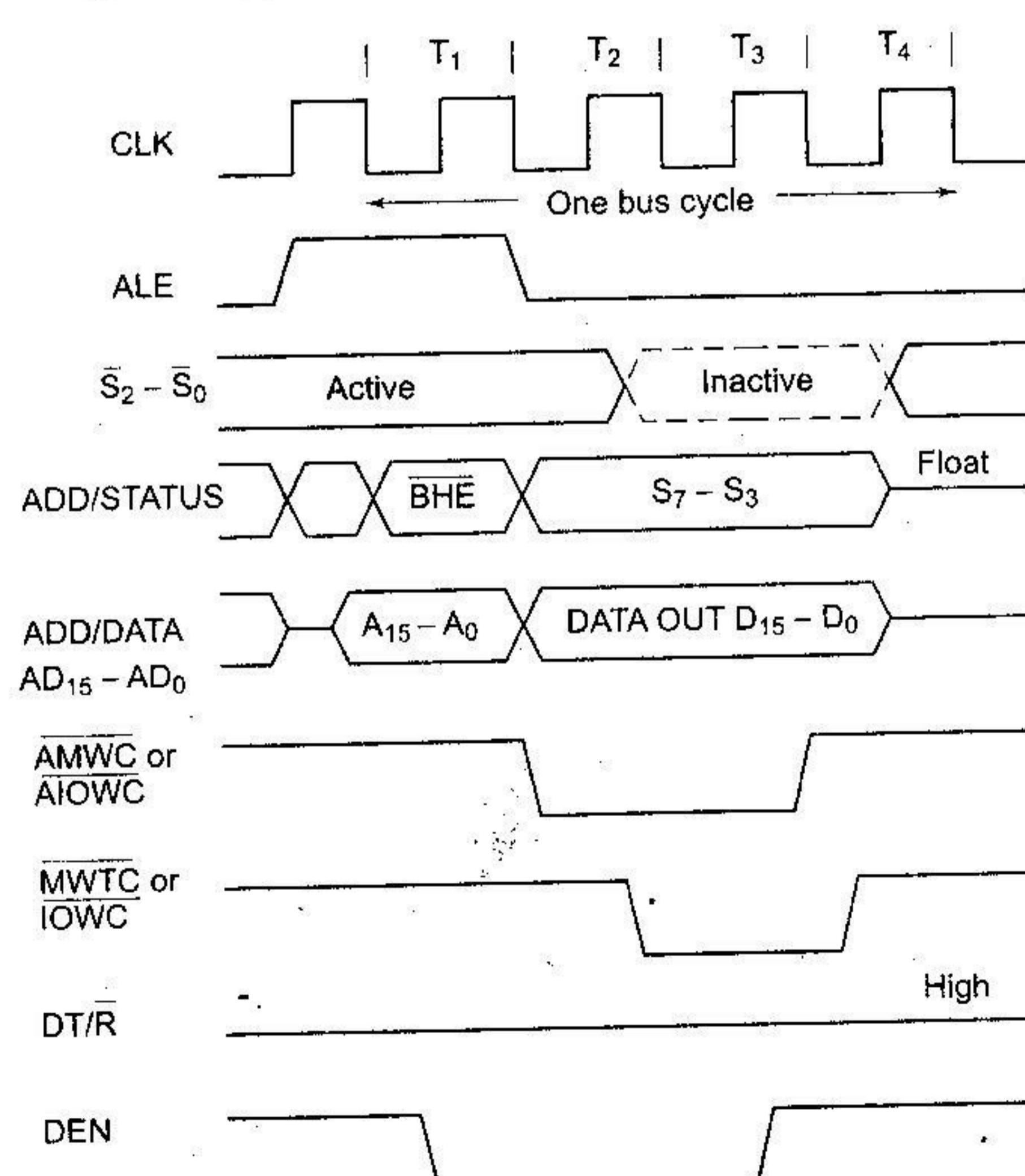


Fig. I.16(b) Memory Write Timing in Maximum Mode

) and  
imum  
gnals  
imum  
i. The  
ldress

### 3) Memory interfacing of 8086 CPU:-

The process in which how memory devices are physically connected to the processor nothing but memory interfacing.

→ In general memory devices are 2-types.

\* ROM's

\* RAM's.

→ ROM's are characterized in to 4 types

1) ROM → Read only memory

2) PROM → Programmable ROM

3) EEPROM → Erasable PROM

4) EEPROM → Electrically Erasable PROM.

→ RAM's are categorised in 2-types.

1) SRAM → Static RAM → flip flops

2) DRAM → Dynamic RAM

↳ Additional bit → Refreshing → Capacitors (8) array of capacitors.

for eg:- 4K x 8 (8) 4K byte memory contains 4096 locations,  
where each location contains 8-bit data and only ~~byte~~ 1 of the  
4096 locations can be selected at a time.

once a location is selected all the bits in it are  
accessible using a group of ~~selected~~ conductors called 'data bus'.

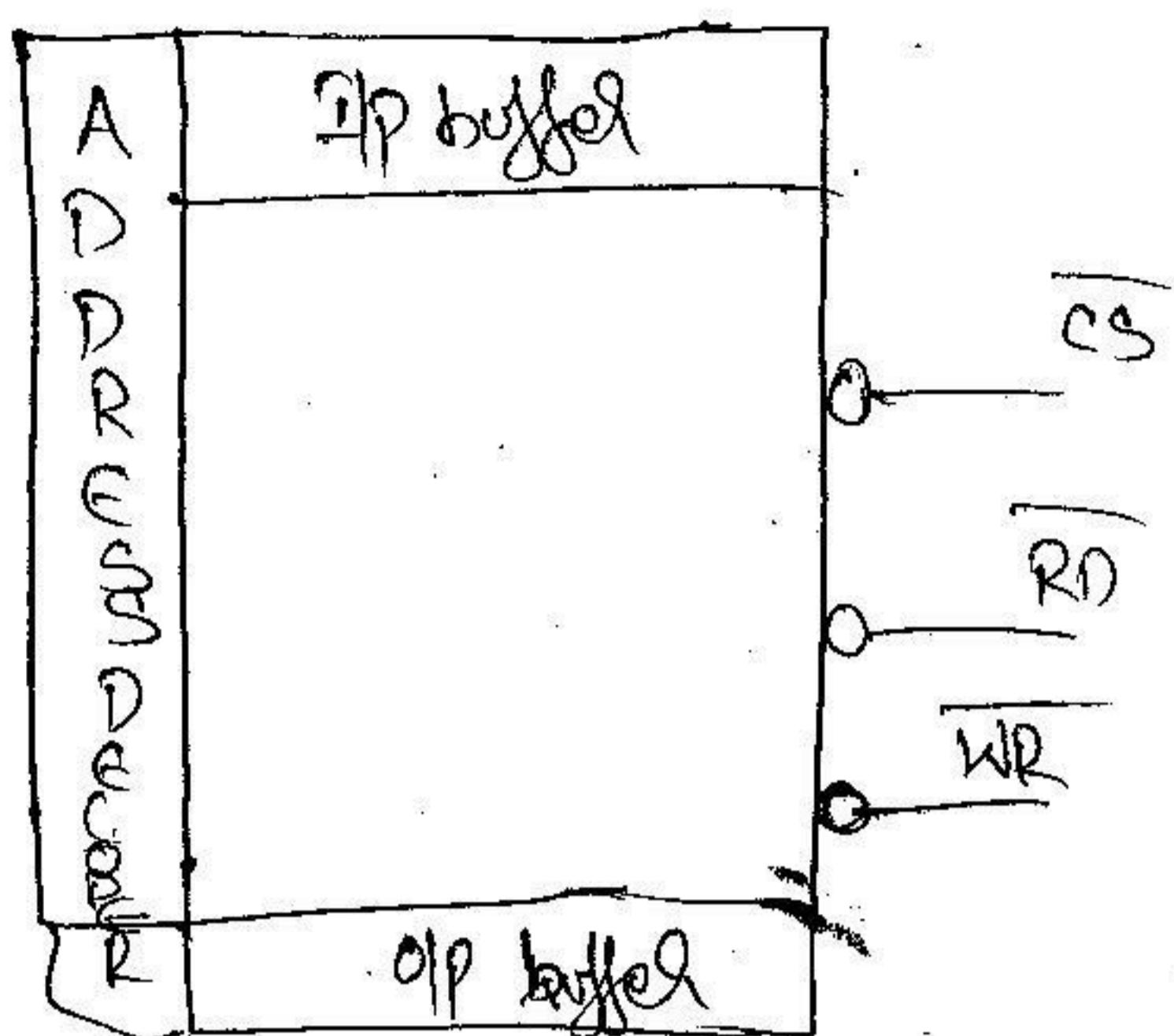
\* In general to address a memory location out of 'N' memory locations, we will require at least n-bits of address.

i.e. n-address lines where  $n = \log_2 N$

where  $2^n = N$ . { $\because N = \text{memory}$   
(n-address lines)}

## \* Structure of EEPROM (&) EPROM : (4KB)

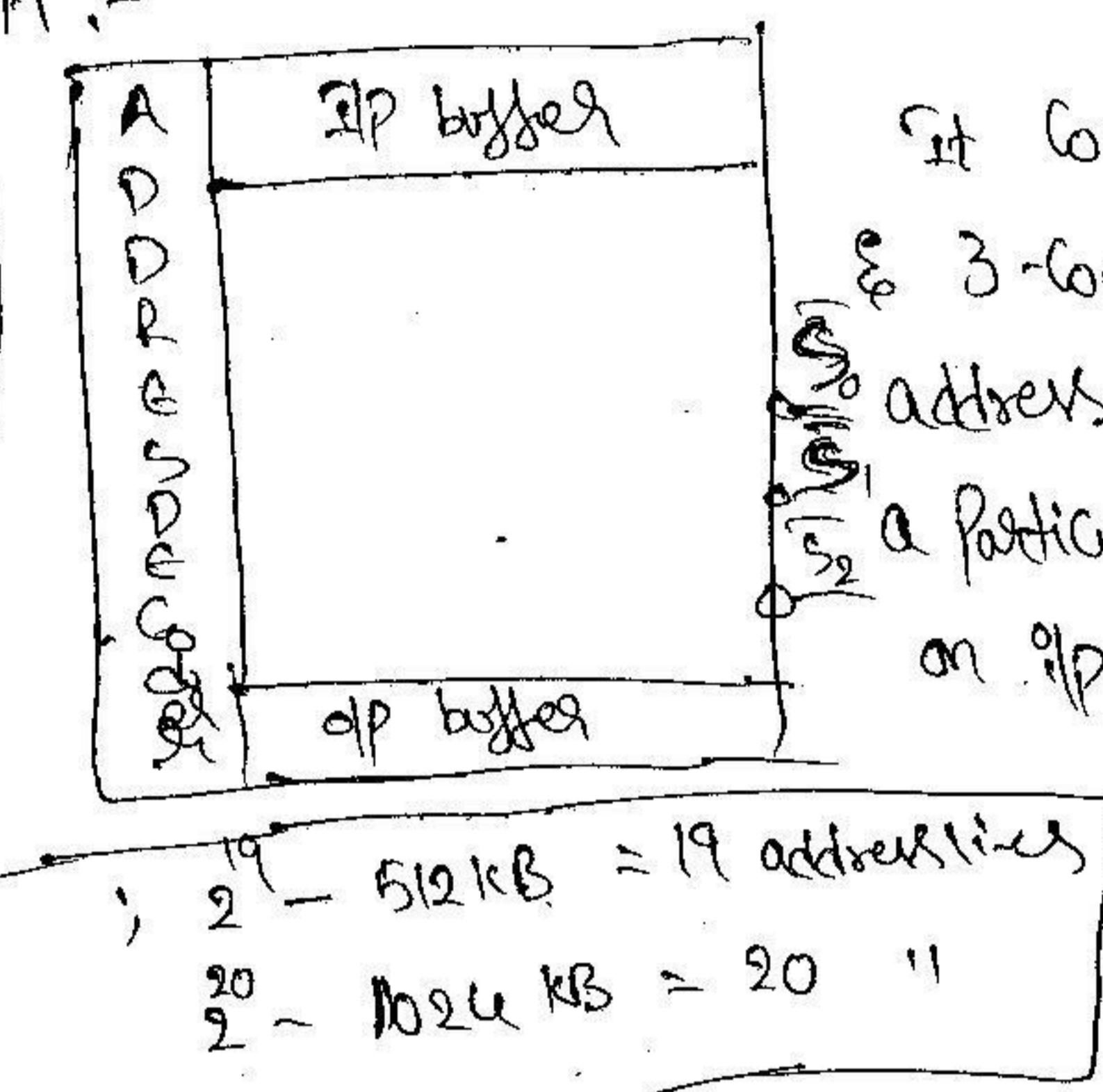
17



$$4KB \Rightarrow 2^2 \times 2^{10} \Rightarrow 2^{12} (\because 12 \rightarrow \text{address lines}) \\ 2^n \Rightarrow n \rightarrow \text{address}$$

## \* Structure of SRAM:-

$2^{10}$	- 1KB = 10 address lines
$2^{11}$	- 2KB = 11 "
$2^{12}$	- 4KB = 12 "
$2^{13}$	- 8KB = 13 "
$2^{14}$	- 16KB = 14 "
$2^{15}$	- 32KB = 15 "
$2^{16}$	- 64KB = 16 "
$2^{17}$	- 128KB = 17 "
$2^{18}$	- 256KB = 18 "



It consists of an address decoder & 3-control signals. Here the address decoded is used to select a particular memory location based on 8P address.

The general procedure of static memory interfacing with 8086 MP.

1. Arrange the available memory chips so as to obtain 16-bit data bus width. The upper 8-bit bank is called odd address memory bank and the lower 8-bit bank is called even address memory bank.
2. Connect available memory address lines of memory chips with those of the microprocessor and also connect the memory RD and WR 8P control signals to the corresponding processor control signals.
3. The remaining address lines of the microprocessor, BHE and A0 are fed for decoding the required chip select signals for odd and even memory banks. The CS of memory is derived from the op. of decoding circuit.

(Q) ~~Design two 2000 locations of 8086~~ → ~~Problem~~

- i) Interface two  $4K \times 8$  EPROMS and two  $4K \times 8$  RAM chips with 8086.

Ans - After RESET the CS & IP are initialised in the form address FFFF0H. The address must lie in the EPROM. The address of RAM may be selected anywhere in the 1MB address space of 8086. but we will select the RAM address such that the address map of system is continuous.

Address	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	
FFFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
:	:	:	EPROM	:	:	:	8Kx8	:	→ 8000 locations differ by 8												
FE000H	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
FDFFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
FC000H	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	

Note  $8KB \Rightarrow 2^3 \times 2^{10} \Rightarrow 13$  - Address lines ;

In this Problem Contains in total 4 (four)  $4K \times 8$  memory chips (RAM & ROM) which are arranged in parallel to obtain 16-bit data bus width.

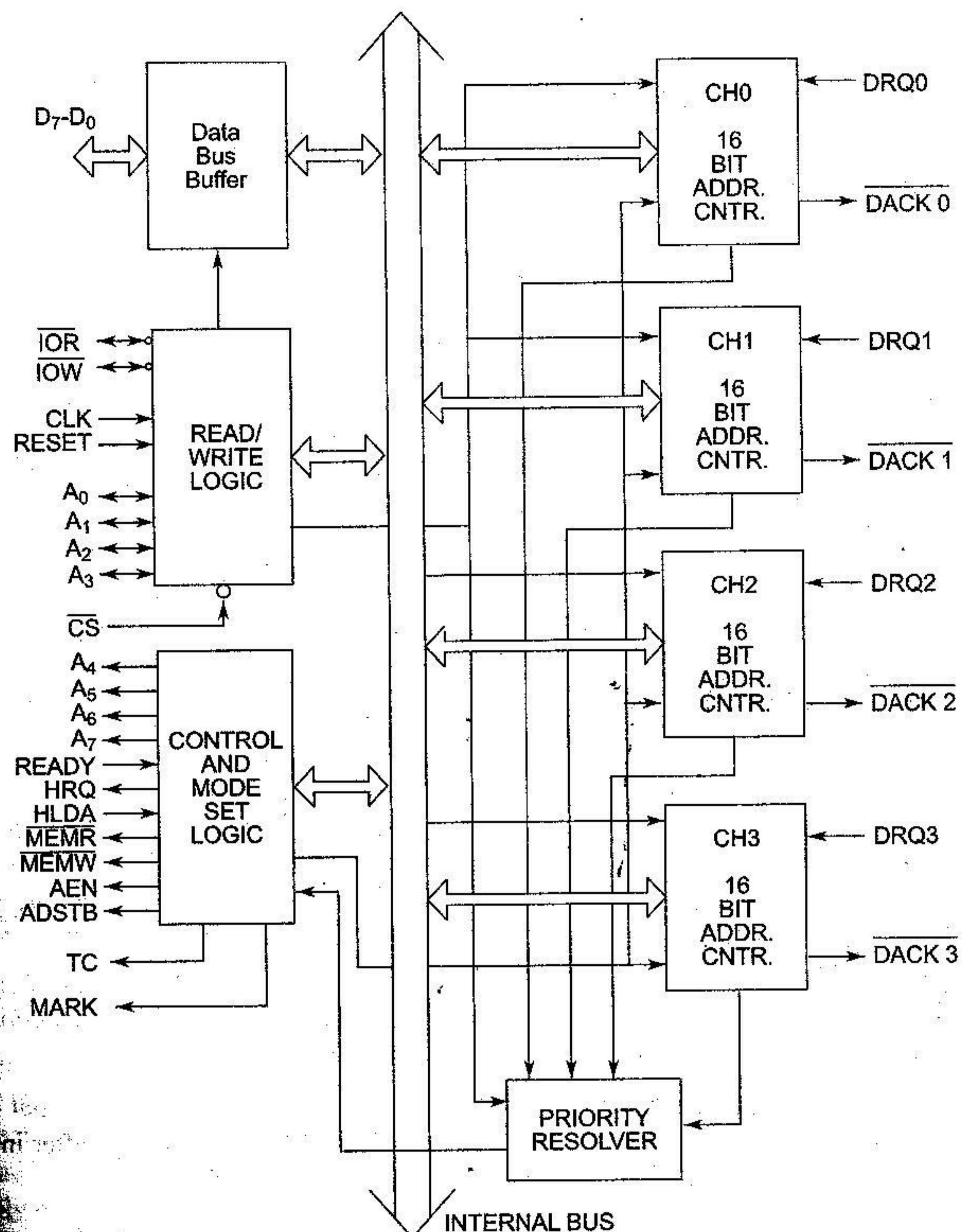


Fig. 7.1 Internal Architecture of 8257

ne modes of data transfer. In any interference free mode, data, address and control lines are available. A data transfer is initiated by one of the data transfer requests.

In their family of controllers, they have to access using local bus or  $\overline{RQ}/\overline{GT}$  pins. When a  $\overline{RQ}/\overline{GT}$  signal is asserted, and it completes the transfer, control of the bus back to CPU.

DMA channels are controlled by these channels. In each channel, there is a control unit, a priority resolver and the following sections.

Over four independent DMA channels, viz. DMA address, data, control and the channels, namely, CH0, CH1, CH2 and CH3. The CPU selects the channel. DMA address bits may be used.

a. The function of the control logic is accessed by the device and controlled by the device. It sends commands to transfer data. These commands are contained in the DMA address register.

channels of 8257. The data transfer is controlled by the data transfer requests. As this register contains 16-bits, the first 14-bits of the register are required for DMA cycle count, incremented by one at the end of each DMA cycle.

bits 14 and 15 of this register indicate the type of the DMA operation (transfer). If the device transfers data into the memory, the DMA operation is called DMA write operation. Bit 14 of the TC register in this case will be set to one and bit 15 will be set to zero. Table 7.2 gives details of DMA operation selection and the corresponding bit configuration of the bits 14 and 15 of the TC register.

**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 select various modes of operation. A DMA channel should not be enabled till the DMA address and terminal count register contain valid information, otherwise, an unwanted DMA request may start a DMA cycle, probably destroying the valid memory data.

Table 7.1 8257 Register Selection

Register	Byte	Address Inputs				F/L EW STOP	Bi-Directional Data Bus							
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>		D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
CH-0 DMA Address	LSB	0	0	0	0	0	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
	MSB	0	0	0	0	1	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>
CH-0 Terminal Count	LSB	0	0	0	1	0	C <sub>7</sub>	C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>
	MSB	0	0	0	1	1	Rd	Wr	C <sub>13</sub>	C <sub>12</sub>	C <sub>11</sub>	C <sub>10</sub>	C <sub>9</sub>	C <sub>8</sub>
CH-1 DMA Address	LSB	0	0	1	0	0	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
	MSB	0	0	1	0	1	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>
CH-1 Terminal Count	LSB	0	0	1	1	0	C <sub>7</sub>	C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>
	MSB	0	0	1	1	1	Rd	Wr	C <sub>13</sub>	C <sub>12</sub>	C <sub>11</sub>	C <sub>10</sub>	C <sub>9</sub>	C <sub>8</sub>
CH-2 DMA Address	LSB	0	1	0	0	0	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
	MSB	0	1	0	0	1	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>
CH-2 Terminal Count	LSB	0	1	0	1	0	C <sub>7</sub>	C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>
	MSB	0	1	0	1	1	Rd	Wr	C <sub>13</sub>	C <sub>12</sub>	C <sub>11</sub>	C <sub>10</sub>	C <sub>9</sub>	C <sub>8</sub>
CH-3 DMA Address	LSB	0	1	1	0	0	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
	MSB	0	1	1	0	1	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>
CH-3 Terminal Count	LSB	0	1	1	1	0	C <sub>7</sub>	C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>
	MSB	0	1	1	1	1	Rd	Wr	C <sub>13</sub>	C <sub>12</sub>	C <sub>11</sub>	C <sub>10</sub>	C <sub>9</sub>	C <sub>8</sub>
MODE SET (Programme only)	—	1	0	0	0	0	AL	TCS	EW	RP	EN3	EN2	EN1	EN0
STATUS (Read only)	—	1	0	0	0	0	0	0	0	UP	TC3	TC2	TC1	TC0

A10-A15 DMA Starting Address, C0-C13 Terminal Count Value (N-1), Rd & Wr-DMA verify (00), Write (01) or Read (10) cycle selection. AL Auto Load, TCS-TC STOP, EW-Extended Write, RP-Rotating Priority, EN3-EN0-Channel Mask Enable, UP-Update Flag, TC3-TC0-Terminal Count Status Bits.

The mode set register format is shown in Fig. 7.2. It is thus extremely important that the mode set register should be programmed by the CPU for enabling the DMA channels only after initializing the DMA address register and terminal count register appropriately.

Table 7.2 DMA Operation Selection Using A<sub>15</sub>/RD and A<sub>14</sub>/WR

Bit 15	Bit 14	Type of DMA Operation
0	0	Verify DMA Cycle
0	1	Write DMA Cycle
1	0	Read DMA Cycle
1	1	(Illegal)

The bits D<sub>0</sub>-D<sub>3</sub> enable one of the four DMA channels of 8257. For example, if D<sub>0</sub> is '1', channel 0 is enabled. If bit D<sub>4</sub> is set, rotating priority is enabled, otherwise, the normal, i.e. fixed priority is enabled. The normal and rotating priorities will be explained later in this text.

Update Flag -

If

If 1  
Enables Auto Load  
Enables TC Stop  
Enables Extended Write  
Enables Rotating Priority

If the TC STOP bit reached, and it further must be reprogrammed after the count reaches

The auto load bit, if software intervention while the channel 3 register address and the terminal are reloaded with the count set.

The extended write them earlier. This is usually not accessed within the request it to add one or into.

Status Register The contain the terminal control that the specific channel status is read by the CPU flag can only be cleared if update flag is set, the

D<sub>4</sub>:- Enabled Routing Priority:- This bit is set to 1 when we are selecting Routing Priority else we are selecting normal.

D<sub>5</sub>:- Enabled Extended write:- This bit is set to '1' then we are extending the 16 write (or) memory write operation that means we are introducing some wait states in eight cycles.

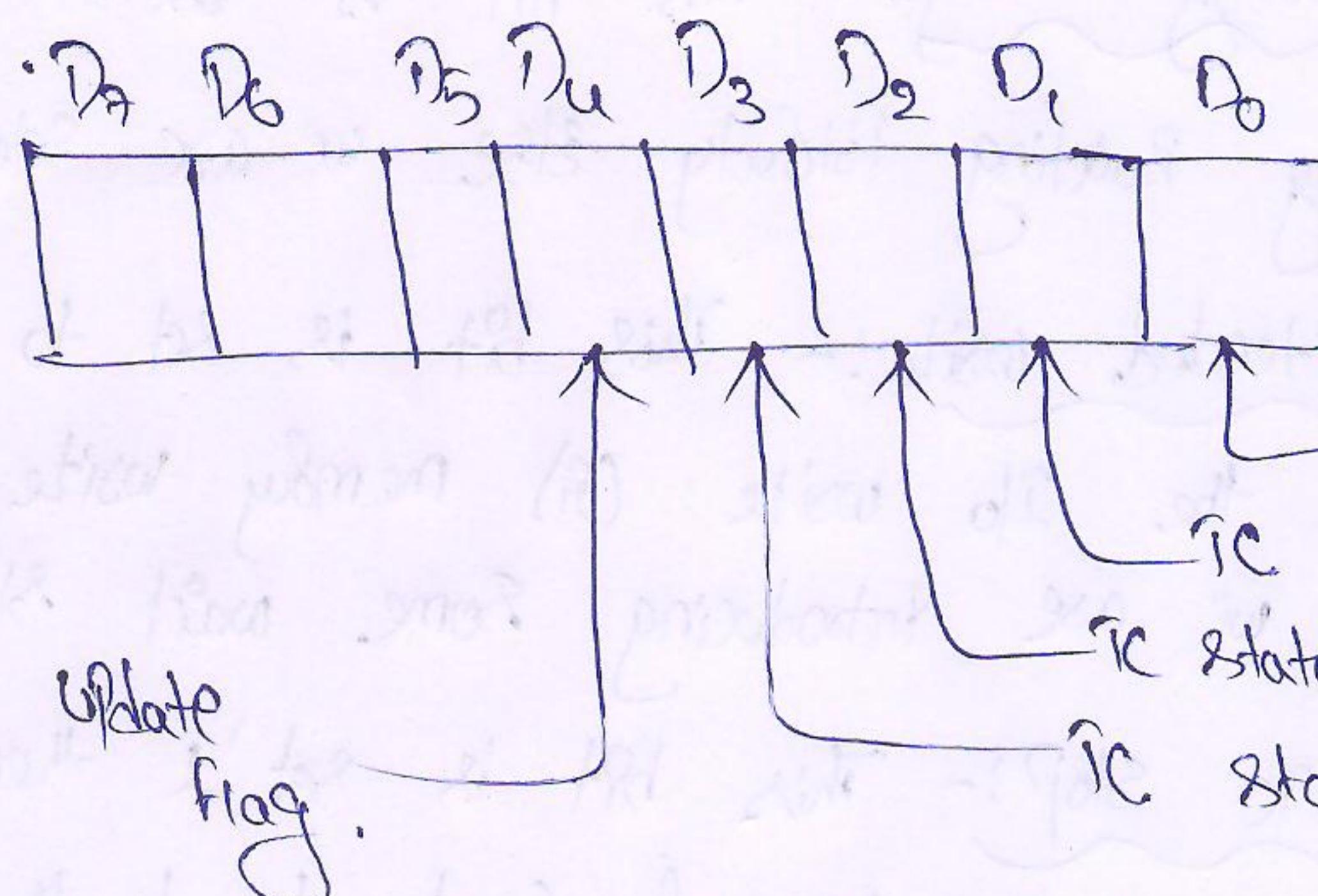
D<sub>6</sub>:- Enable IC Stop:- This bit is set '1' then particular channel is reached its terminal Count, due to that appropriate channel is disable.

D<sub>7</sub>:- Enable Auto Load:- This bit is set '1' then the channel 2 of 8257 is enters into the repeat block chaining operation mode.

\* ) Status Registers:- In this the back order 4-bits of this register contains the terminal Count status for the 4 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 CPU (or) 8257 is reset.

The update flag is not affected by read operation. This flag can only be cleared by resetting 8257 (or) by resetting (AL) bit of mode set register. If update flag is set, the contents of channel 3 registers are reloaded to corresponding registers of channel 2 when ever 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.



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 autoload 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 be read.

### 7.1.2 Data Bus Buffer, Read/Write Logic, Control Unit and Priority Resolver

The 8-bit, tristate, bidirectional buffer interfaces the internal bus of 8257 with the external system bus under the control of various control signals. In the slave mode, the read/write logic accepts the I/O Read or I/O Write signals, decodes the  $A_0$ - $A_3$  lines and either writes the contents of the data bus to the addressed internal register or reads the contents of the selected register depending upon whether IOW or IOR signal is activated. In master mode, the read/write logic generates the IOR and IOW signals to control the data flow to or from the selected peripheral. The control logic controls the sequences of operations and generates the required control signals like AEN, ADSTB, MEMR, MEMW, TC and MARK along with the address lines  $A_4$ - $A_7$ , in master mode. The priority resolver resolves the priority of the four DMA channels depending upon whether normal priority or rotating priority is programmed.

### 7.1.3 Signal Descriptions of 8257

Figure 7.4 shows pin configuration of 8257, followed by the functional description of each signal.

**DRQ<sub>0</sub>-DRQ<sub>3</sub>** These are the four individual channel DMA request inputs, used by the peripheral devices for requesting DMA services. The DRQ<sub>0</sub> has the highest priority while DRQ<sub>3</sub> has the lowest one, if the fixed priority mode is selected.

**DACK<sub>0</sub>-DACK<sub>3</sub>** 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.

**D<sub>0</sub>-D<sub>7</sub>** 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 D<sub>0</sub>-D<sub>7</sub> 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 D<sub>0</sub>-D<sub>7</sub> during the first clock cycle of the DMA cycle. During the rest of the period, data is available on the data bus.

IOR	1	40	= A <sub>7</sub>
IOW	2	39	= A <sub>6</sub>
MEMR	3	38	= A <sub>5</sub>
MEMW	4	37	= A <sub>4</sub>
MARK	5	36	= TC
READY	6	35	= A <sub>3</sub>
HLDA	7	34	= A <sub>2</sub>
ADSTB	8	33	= A <sub>1</sub>
AEN	9	32	= A <sub>0</sub>
HRQ	10	31	= Vcc
CS	11	8257	
CLK	12		
RESET	13		
DACK2	14		
DACK3	15		
DRQ <sub>3</sub>	16		
DRQ <sub>2</sub>	17		
DRQ <sub>1</sub>	18		
DRQ <sub>0</sub>	19		
GND	20		

Fig. 7.4 Pin Diagram of 8257

**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 as 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 disables all the DMA channels by clearing the mode register and tristates all the control lines.

**A<sub>0</sub>-A<sub>3</sub>** These are the four least significant address lines. In slave mode, they act as input which select 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.

**A<sub>4</sub>-A<sub>7</sub>** 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. In the non-cascaded 8257 systems, this is connected with HOLD pin of CPU. In the cascade mode, this pin of a slave is connected with a DRQ input line of the master 8257, while that of the master is connected with HOLD input of the CPU.

**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.

**ADSTB** 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. This also may be used to transfer the higher byte of the generated address over the data bus. If the 8257 is I/O mapped, then it should be used to disable the other I/O devices, when the DMA controller address is on the address bus.

**TC** Terminal count output indicates to the currently selected peripheral that the present DMA cycle is the last for the previously programmed data block. 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 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.

**MARK** The modulo 128 mark output indicates to the selected peripheral that the current DMA cycle is the 128th cycle since the previous MARK output. The mark will be activated after each 128 cycles or integral multiples of it from the beginning of the data block (the first DMA cycle), if the total number of the required DMA cycles ( $n$ ) is completely divisible by 128.

**V<sub>cc</sub>** 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).

## 7.2 DMA TRANSFERS AND OPERATIONS

The 8257 is able to accomplish three types of operations, viz. verify DMA operation, write operation and read operation. The complete operational sequence of 8257 is described using a state diagram in Fig. 7.5 for a single channel.

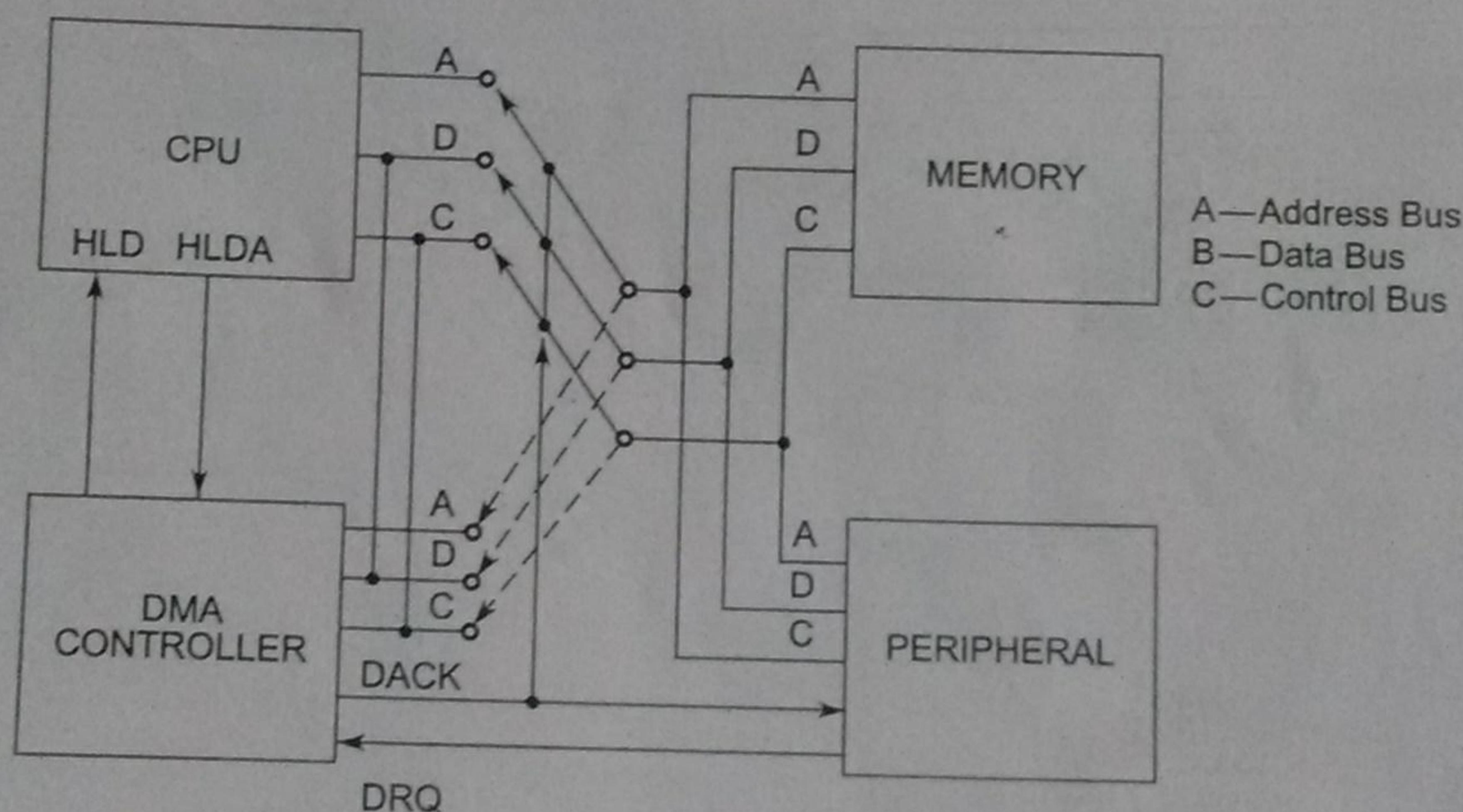
A single byte transfer using 8257 may be requested by an I/O device using any one of the 8257 DRQ inputs. In response, the 8257 sends HRQ signal to the CPU at its HLD input and waits for acknowledgement at the HLDA input. If the HLDA signal is received by the DMA controller, it indicates that the bus is available for the transfer. The DACK line of the used channel is pulled down by the DMA controller to indicate the I/O device that its request for the DMA transfer has been honoured by the CPU. The DMA controller generates the read and write commands to transfer the byte from/to the I/O device. The DACK line is pulled down when the transfer is over, to indicate the DMA controller that the transfer, as requested by the device, is over. The HRQ line is lowered by the DMA controller to indicate the CPU that it may regain the control of the bus. The DRQ must be high until acknowledged and must go low before S<sub>4</sub> state of the DMA operation state diagram to avoid another unwanted transfer.

If more than one channel requests service simultaneously, the transfer will occur as a *burst transfer*. This will be discussed further in case of 8237. No overhead is required in switching from one channel to another. In each S<sub>4</sub>, the DRQ lines are sampled and the highest priority request is recognized during the next transfer. Once the higher priority transfer is over; the lower priority transfer requests may be served, provided their DRQ lines are still active. The HRQ line is maintained active till all the DRQ lines go low.

The burst or continuous transfer, described above may be interrupted by an external device by pulling down the HLDA line. After each transfer, the 8257 checks the HLDA line. If it is found inactive,

### 7.2.3 Interfacing 8257 with 8086

Once a DMA controller is initialised by a CPU properly, 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 transfers). 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 tristated 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. A conceptual implementation of the system is shown in Fig. 7.7(a).



**Fig. 7.7(a)** Interfacing a Typical DMA Controller with a System

To explain the interfacing of 8257 with 8086 let us consider an interfacing example.

#### Problem 7.1

Interface DMA controller 8257 with 8086 so that the channel 0 DMA address register has an I/O address 80H and the mode set register has an address 88H. Initialize the 8257 with normal priority, TC stop and non-extended write. Autoload is not required. The transfer is to take place using channel 0. Write an ALP to move 2KB of data from a peripheral device to memory address 2000:5000H, with the above initialisation.

**Solution** Figure 7.7(b) shows interfacing connections of DMA controller 8257 with 8086. As the DMA controller can generate only 16-bit address, while the CPU generates 20-bit address, the four upper address bits are generated and latched on the bus externally using a latch 8212. The 8086 uses three more 8212 latches and two 74245 buffers to demultiplex the address and data buses. These latches are controlled using the AEN signal of the DMA controller. If the DMA controller is in master mode, these will be automatically disabled and if the DMA controller is in slave mode, i.e.