Microprocessor Signals and Machine Cycle

Second Stage

Microprocessor

Computer Engineering Department

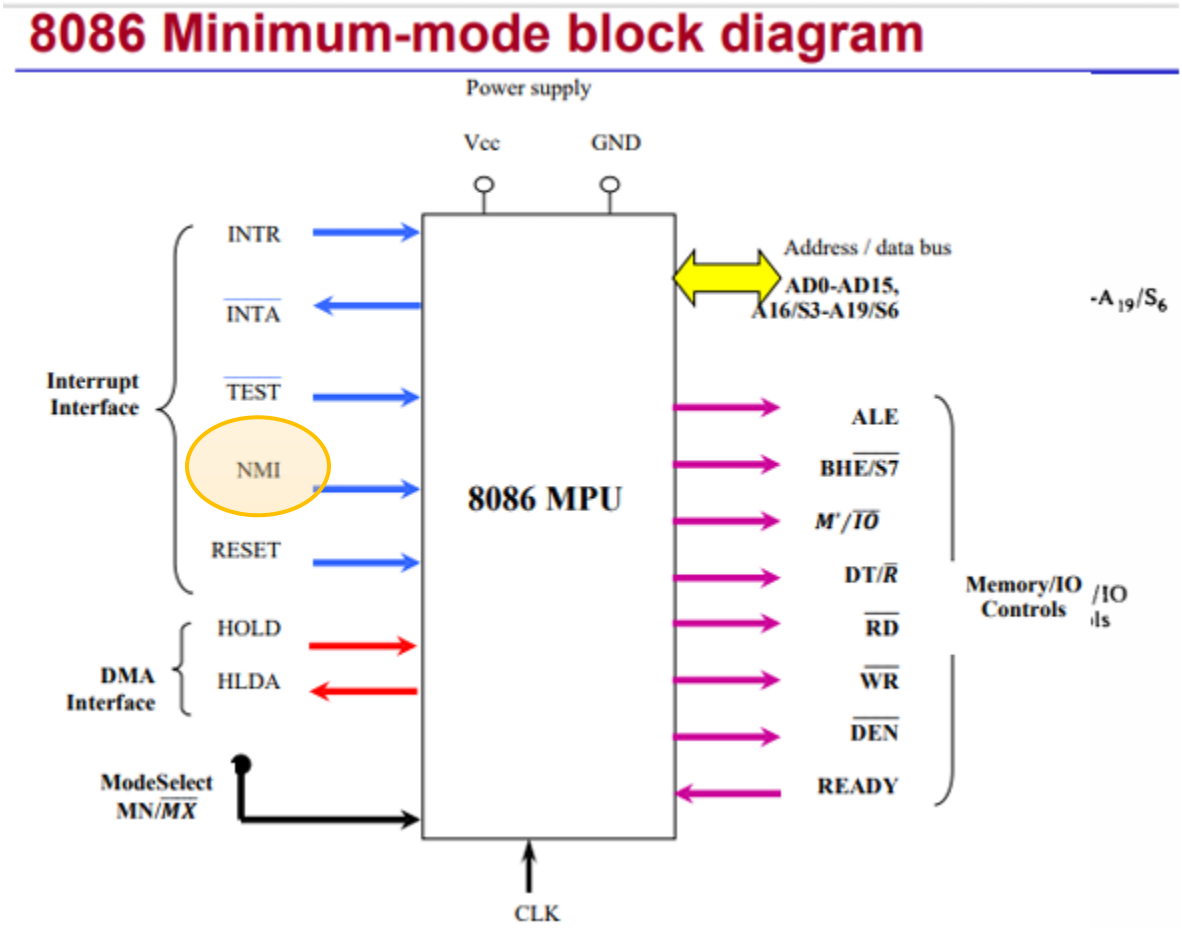
Lecture No. 3

Prepared by: Asst. Lect. Eng.
Hamza Waleed Hamza

8086 Microprocessor Signals

The Intel 8086 microprocessor can function in two distinct modes depending on the wiring of its MN/MX# (Minimum/Maximum) pin. In the **Minimum Mode (MN/MX# = 1)**, the 8086 acts as a standalone processor. It is designed for small, single processor systems where cost and simplicity are prioritized. In the **Maximum Mode (MN/MX# = 0)**, the 8086 is configured for multiprocessor or complex systems. It gives up its direct control over the bus to allow other processors (like the 8087 Math Coprocessor) to share resources.

The 8086 signals can be categorized in three groups. The first are the signals having common functions in minimum as well as maximum mode, the second are the signals which have special functions in minimum mode and third are the signals having special functions for maximum mode.



Tables below illustrate the signal required to operate in the minimum mode

The signals common for both minimum & maximum modes:

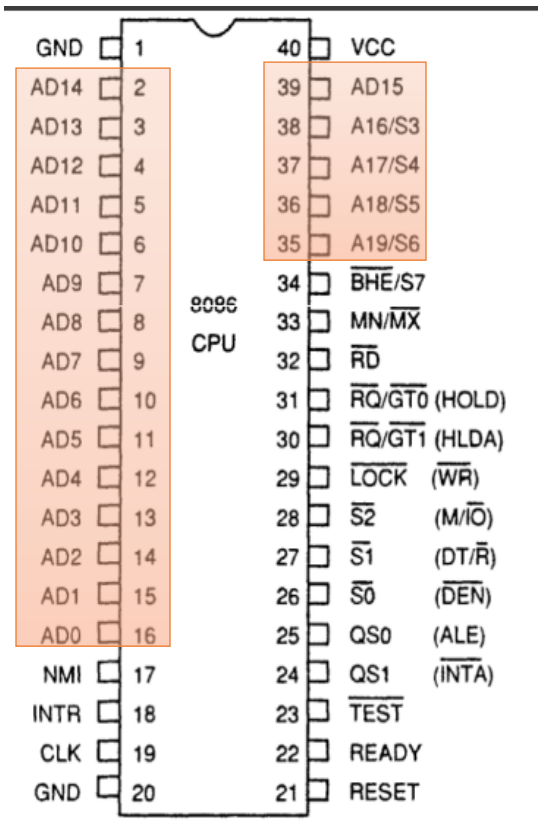
Common Signals		
Name	**Function**	**Type**
AD15-AD0	Address/Data Bus	Bidirectional , 3-state
A19/S6 - A16/S3	Address/Status	Output , 3-state
MN / $\overline{\text{MX}}$	Minimum/Maximum mode control	Input
$\overline{\text{RD}}$	Read Control	Output , 3-state
$\overline{\text{TEST}}$	Wait on test control	Input
$\overline{\text{READY}}$	Wait state control	Input
RESET	System reset	Input
CLK	System clock	Input
Vcc	+5V	Input
GND	Ground	Input
$\overline{\text{BHE}}$ /S7	Bus High Enable/Status	Input
INTR	Interrupt Request	Input
NIM	non maskable interrupt Request	Input

Minimum mode interface signals

Minimum mode Signals (MN / $\overline{\text{MX}}$ =Vcc)		
Name	**Function**	**Type**
(M/$\overline{\text{IO}}$)	Memory/IO Control	Output , 3-state
$\overline{\text{WR}}$	WRITE Control	Output , 3-state
ALE	Address Latch Enable	Output
DT/$\overline{\text{R}}$	Data Transmit/Receive	Output , 3-state
DEN	Data Enable	Output , 3-state
HOLD	Hold request	Input
HLDA	Hold Acknowledgment	Output
$\overline{\text{INTA}}$	Interrupt Acknowledgment	Output

Minimum Mode signals and Machine Cycles

❖ **Address/Data Bus**: The address bus is 20 bits long and consists of signal lines A0 (LSB) through A19 (MSB). However, only address lines A0 through A15 are used when accessing I/O. The data bus lines are multiplexed with address lines. For this reason, they are denoted as AD0 through AD15. Data line D0 is the LSB.



❖ **Status Signals**: The four most significant address lines A16 through A19 of the 8086 are multiplexed with status signals S3 through S6. These status bits are output on the bus at the same time that data are transferred over the bus lines AD0-AD7. S3 and S4 bits identify which the internal segment register was used to generate the physical address that was output on the address bus during the current bus cycle. Table below shows the S4 S3 code and the corresponding segment register. S5 is the logic of IF, S6 is always=0, AND S0,S1,S2 are used in maximum mode.

S4	S3	Address status
0	0	Alternate(relative to ES segment)
0	1	Stack (relative to SS Segment)
1	0	Code/None (relative to CS segment or a default zero)
1	1	Data (relative to DS segment)

❖ **Address Latch Enable (ALE)** :Itis an output signal provided by the 8086 and can be used to demultiplexing AD0 to AD15 in to address(A10 toA15) and data (D0 to D15).

Bus De-Multiplexing

The 8086 microprocessor has time-multiplexed 16-bit address/data bus AD_{15}-AD_0 and 4-bit address/status bus A_{19}/S_6-A_{16}/S_3. The ALE signal is used to latch the address of 8086. Usually, latch ICs are available with eight separate latches. Therefore, three latch ICs should be used for demultiplexing 20-bit address lines. Figure below shows the circuit diagram for latching 20-bit address lines using three 74LS373 latch ICs. In this arrangement, two ICs are fully utilized and one latch is partially used.

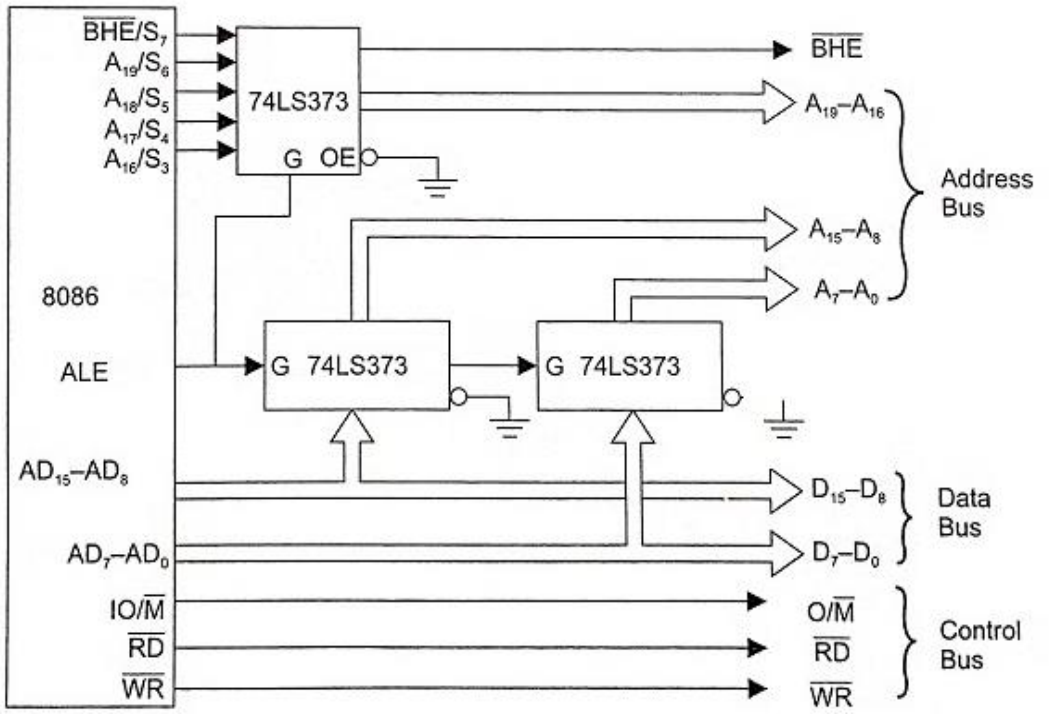
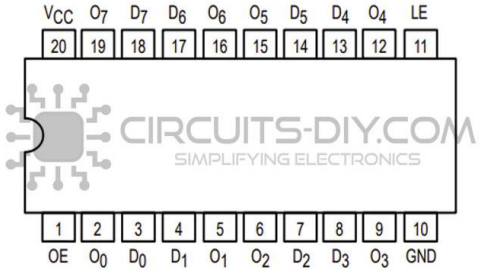
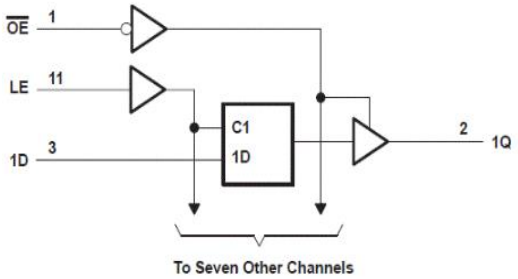
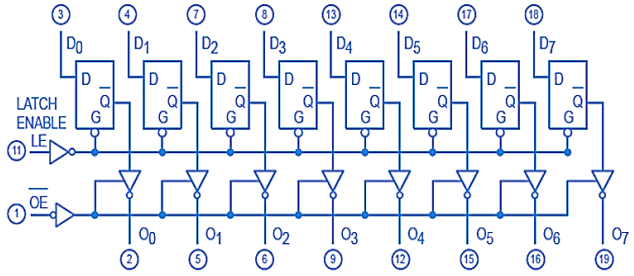


Fig. 5.28 *Demultiplexing of 20-bit address bus in 8086 processor*

74LS373 LATCH, BUFFER





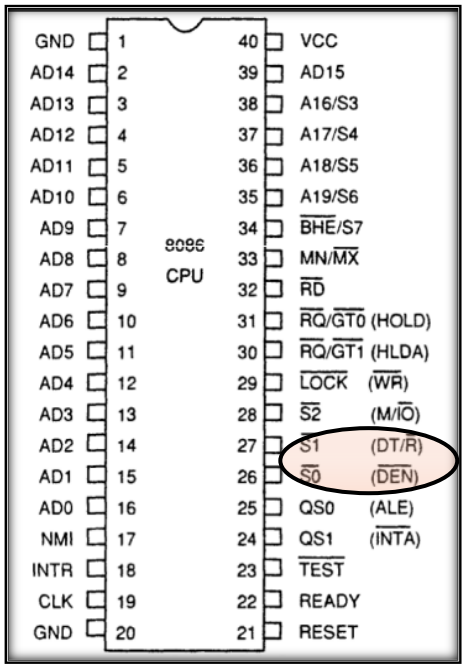


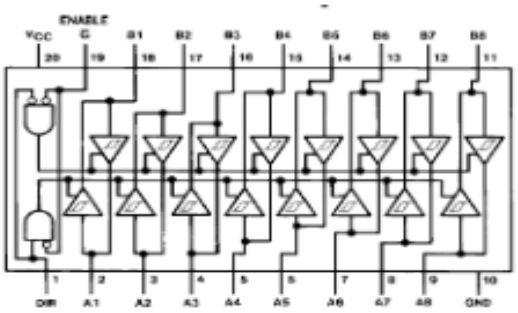
(Each Latch)

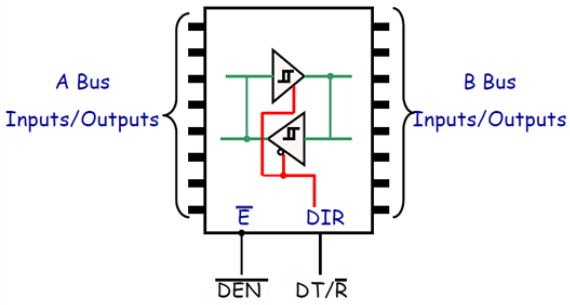
INPUTS			OUTPUT
$\overline{OE}$	LE	D	Q
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

- **DT/$\overline{R}$**: (Data Transmit/ Receive) Output signal from the processor to control the direction of data flow through the data transceivers
- **$\overline{DEN}$**: (Data Enable) Output signal from the processor used as output enable for the transceivers ALE (Address Latch Enable) Used to demultiplex the address and data lines using external latches

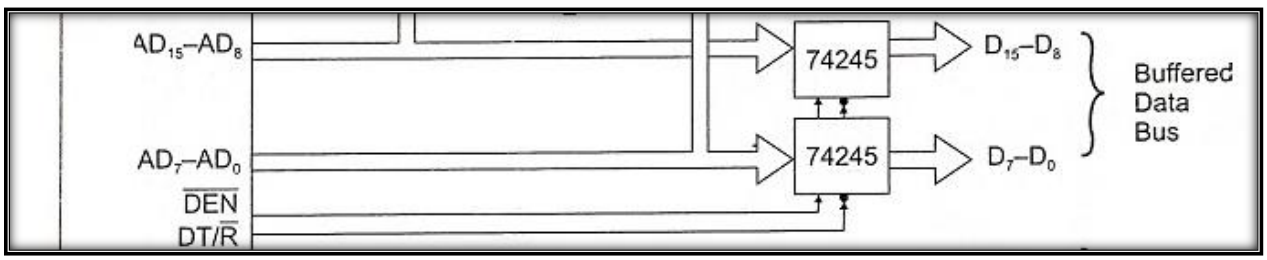
Both DT/**R** and DEN are controlled the direction of data transfer through data bus using IC 74LS245





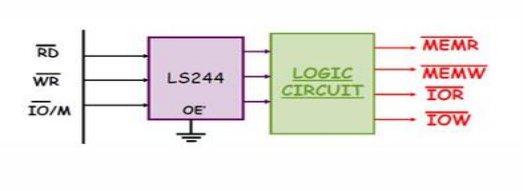


Enable $\overline{G}$	Direction Control DIR	Operation
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

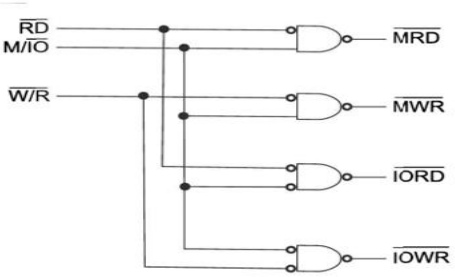


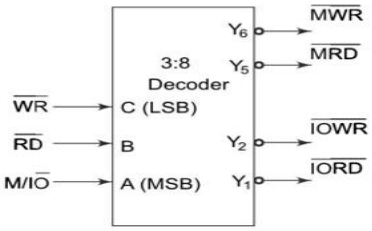
Memory I/O control signals logic

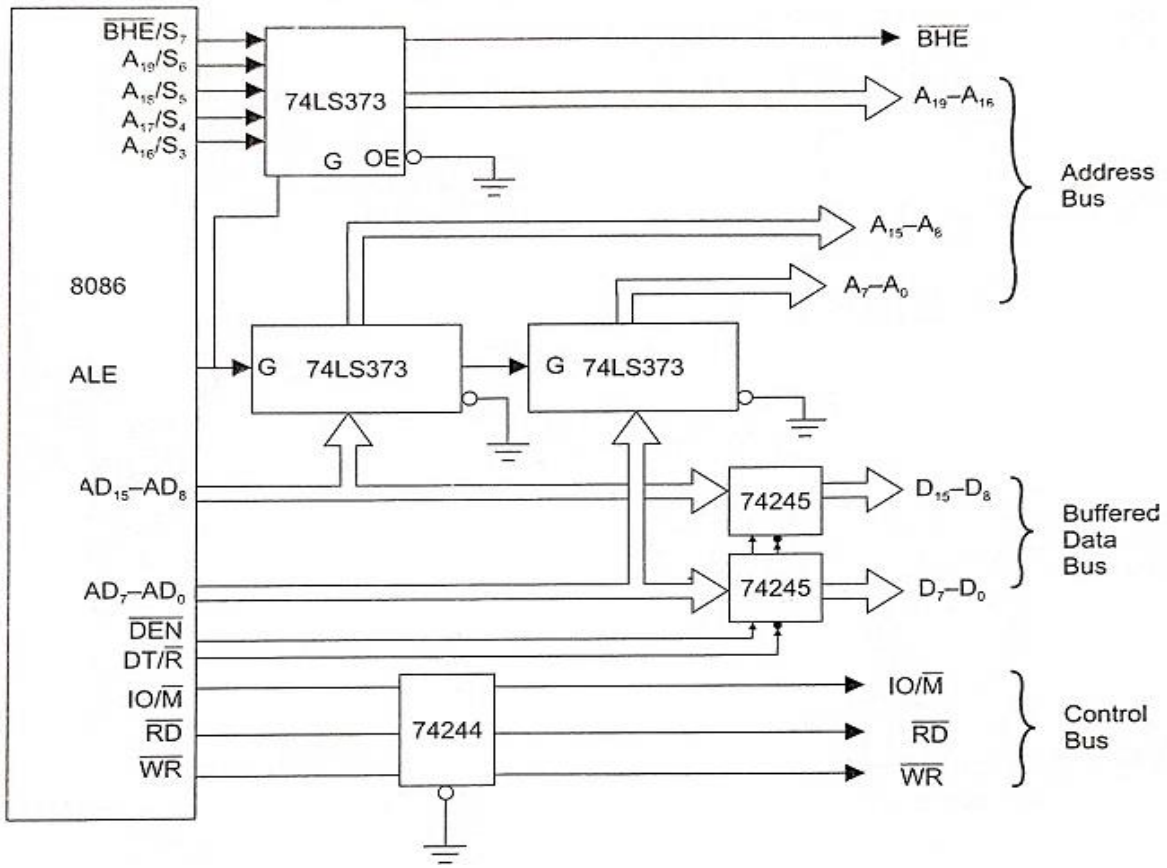
- **Read ($\overline{RD}$):**is logic 0 (low) when the data is read from memory or I/O location.
- **Memory/IO (M/$\overline{IO}$):** This is a status line when it is LOW, it indicates the CPU is having an I/O operation, and when it is HIGH, it indicates that the CPU is having a memory operation.
- **Write ($\overline{WR}$)** :Write control signal; asserted low Whenever processor writes data to memory or I/O port



M/IO'	RD'	WR'	FUNCTION
1	0	1	MEMR
1	1	0	MEMW
0	0	1	IOR
0	1	0	IOW

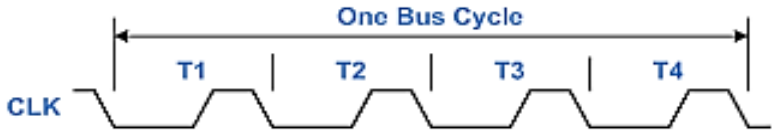






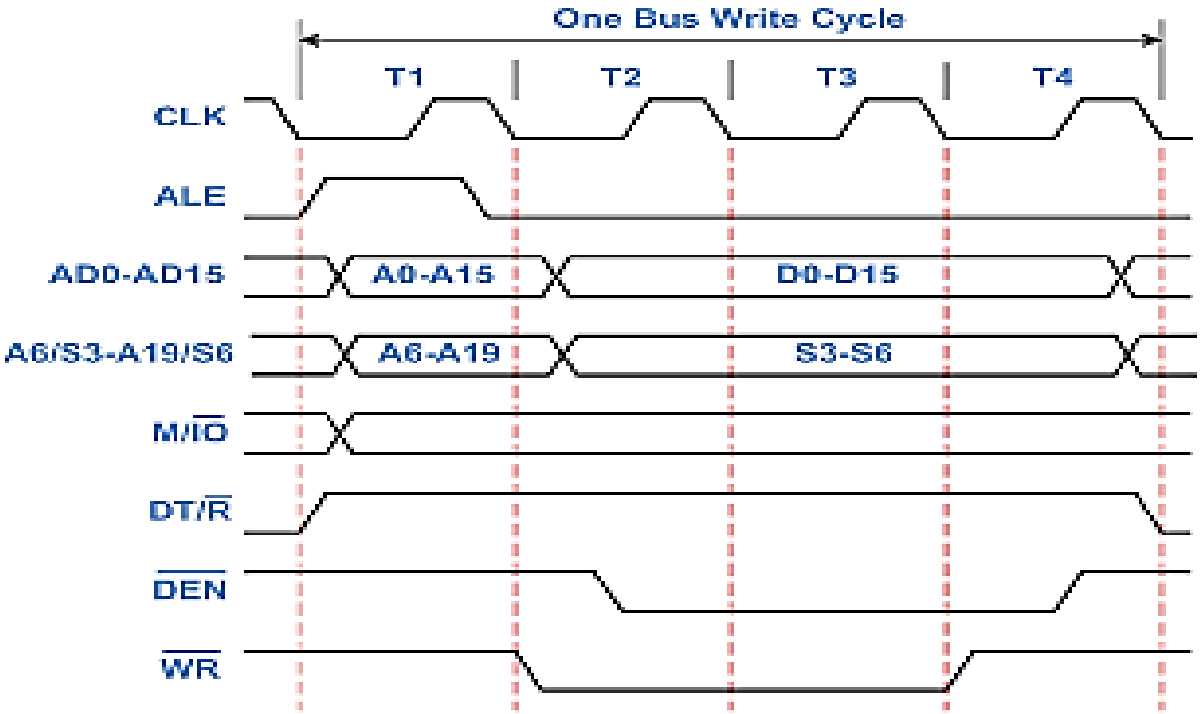
Machine Cycle

8086 access memory and I/O devices in periods called bus cycles. Each cycle equals 4 system-clocking periods (T states). If the clock is operated at 5 MHz, one 8086 bus cycle is complete in 800 ns.



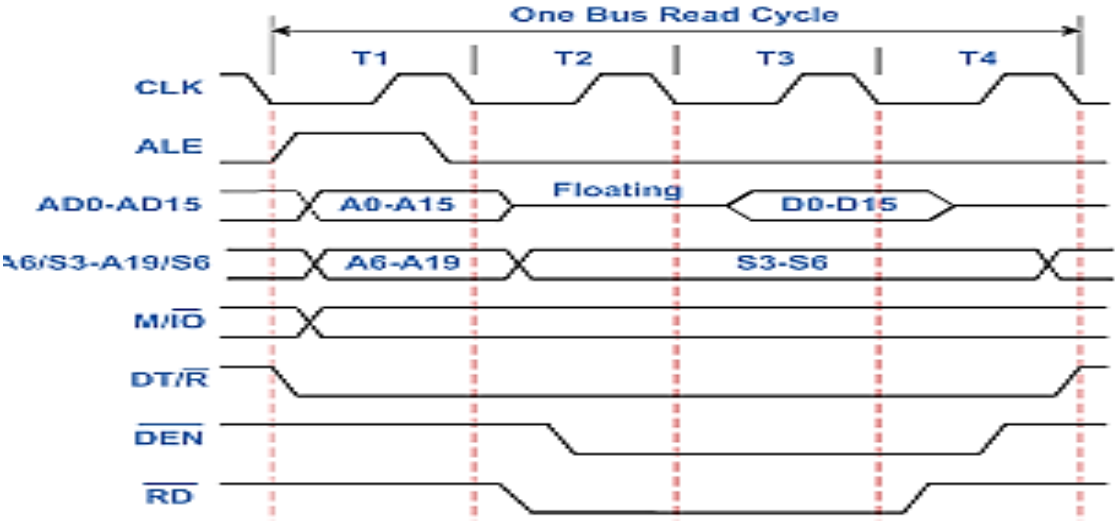
8086 Memory Write Cycle

- ***During T*** $_1$: The *address is placed on the Address/Data bus.*
- *Control signals M/* $\overline{\text{IO}}$*, ALE and DT/* $\overline{\text{R}}$ *specify memory or I/O, latch the address onto the address bus and set the direction of data transfer on data bus.*
- ***During T*** $_2$*:8086 issues the* $\overline{\text{WR}}$ *signal,* DEN*, for writing the data.*
    - $\overline{\text{DEN}}$ *enables the memory or I/O device to receive the data for writes.*
- ***During T*** $_3$: This *cycle is provided to allow memory to access data.*
- ***During T*** $_4$: All *bus signals are deactivated, in preparation for next bus cycle.*



8086 Memory Read Cycle

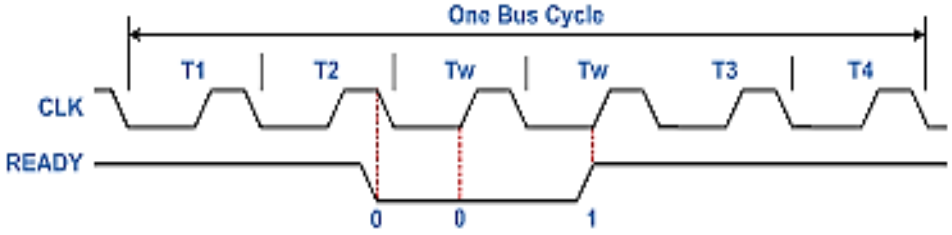
- ***During T*** $_1$ *:The address is placed on the Address/Data bus. Control signals M/*$\overline{\text{IO}}$*, ALE and DT/*$\overline{\text{R}}$ *specify memory or I/O, latch the address onto the address bus and set the direction of data transfer on data bus.*
- ***During T*** $_2$ *:8086 issues the* $\overline{\text{RD}}$ *signal,* $\overline{\text{DEN}}$ *enables the 8086 to receive the data for reads.*
- ***During T*** $_3$ *:This cycle is provided to allow memory to access data.*
- ***During T*** $_4$ *:All bus signals are deactivated, in preparation for next bus cycle.*



8086 Memory Read Cycle with Wait State

❖ **READY**: This is the acknowledgement from the slow device or memory that they have completed the data transfer. The signal made available by the devices is synchronized by the 8284A clock generator to provide ready input to the 8086.

- A wait state (Tw) is an extra clocking period inserted between T2 and T3.
- The READY input is sampled (checked) at the end of T2 and in the middle of Tw.
- If READY is a logic 0 at the end of T2, then T3 is delayed and Tw is inserted between T2 and T3.
- READY is next sampled at the middle of Tw to determine if the next state is Tw or T3. The READY signal is synchronized with clock using the 8284Aclock generator.



Memory address space and data organization

8086 can supports 1Mbyte of external memory that organized as individual bytes of data stored at consecutive addresses over the address range 00000_{16} to $FFFFF_{16}$. The 8086 can access any two consecutive bytes as a word of data. The lower-addressed byte is the least significant byte of the word, and the higher- addressed byte is its most significant byte.

The word of data is at an even-address boundary if its least significant byte is in even address. It's also called aligned word. The word of data is at an odd-address boundary if its least significant byte is in odd address. It's also called misaligned word, as shown in Fig 3. To store double word four locations are needed. The double word that it's least significant byte store at an address that is a multiple of 4 (e.g. 0, 4 , 8,....) as shown in Fig 4.

Example 1: For the 1Mbyte memory shown in Fig 2, storage location of address 00009_{16} contains the value 00000111_2=**7_{16}** , while the location of address 00010_{16} contains the value 01111101= **$7D_{16}$** . The 16-bit word **$225A_{16}$** is stored in the locations $0000C_{16}$ to $0000D_{16}$.

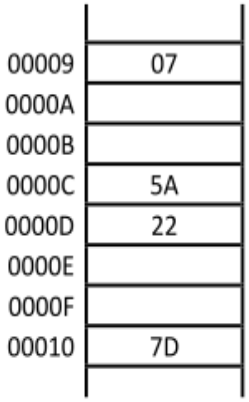
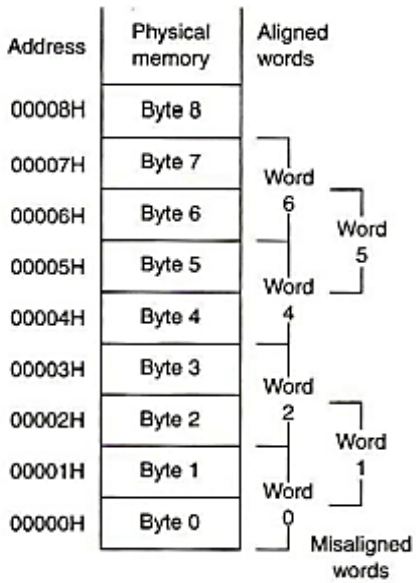
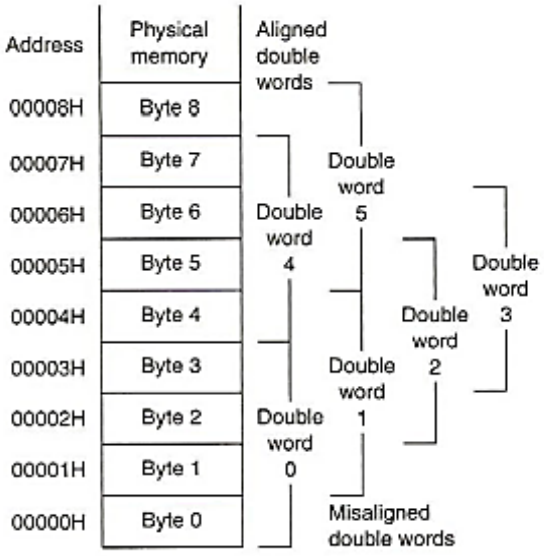


Fig2:Part of 1Mbyte memory



Aligned and misaligned word



Aligned and misaligned double word

Hardware Organization of the Memory Address Space

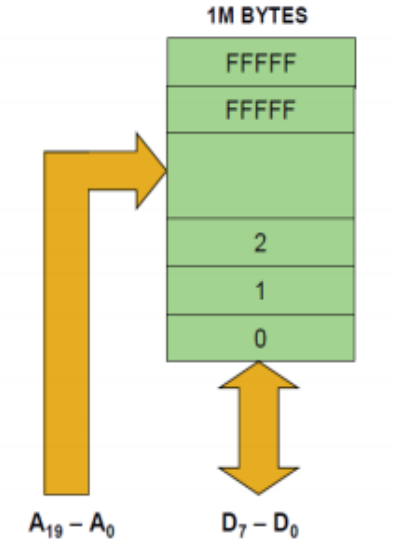
The 8088µP's memory address space as shown in figure below is implemented as single 1 Mbyte memory bank. The 8086µP's 1 Mbyte memory address space as shown in figure below is implemented as two independent 512 Kbyte banks:

♣ Data bytes associated with an even address (00000, 00002, …. , FFFFE) reside in the low bank.

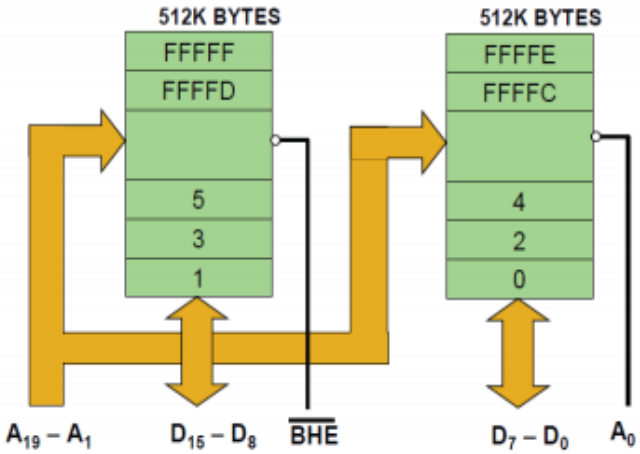
♣ Data bytes associated with an odd addresses (00001, 00003, ….. , FFFFF) reside in the high bank.

Address bits A1 through A19 select the location that is to be accessed. They are applied to both banks in parallel. A0 (BLE) and bank high enable ($\overline{BHE}$) are used as bank-select signals:

♣ A0 = 0 causes the low bank to be enabled. While BHE = 0 causes the high bank to be enabled. Each of the memory banks provides half of the 8086µP's 16-bit data bus. The lower bank transfers bytes of data over data lines D0 through D7, while data transfers for a high bank use D8 through D15.

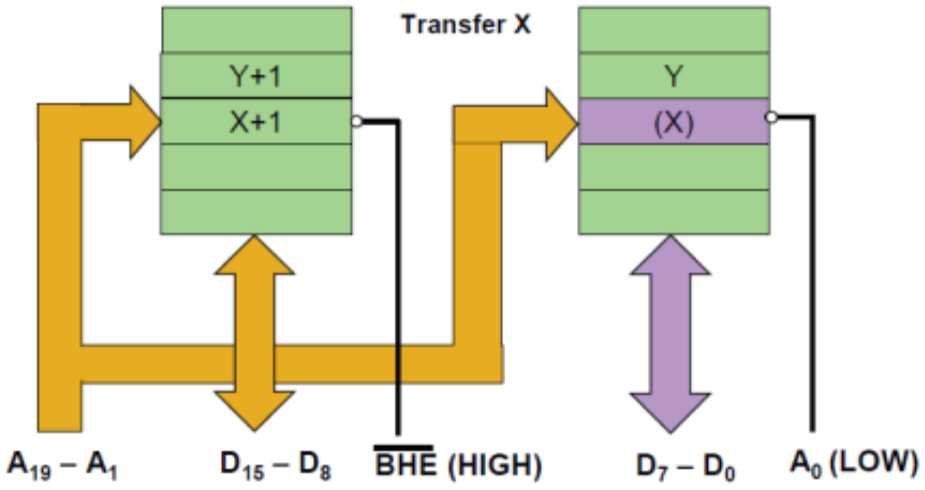


1Mx8 memory bank of the 8088



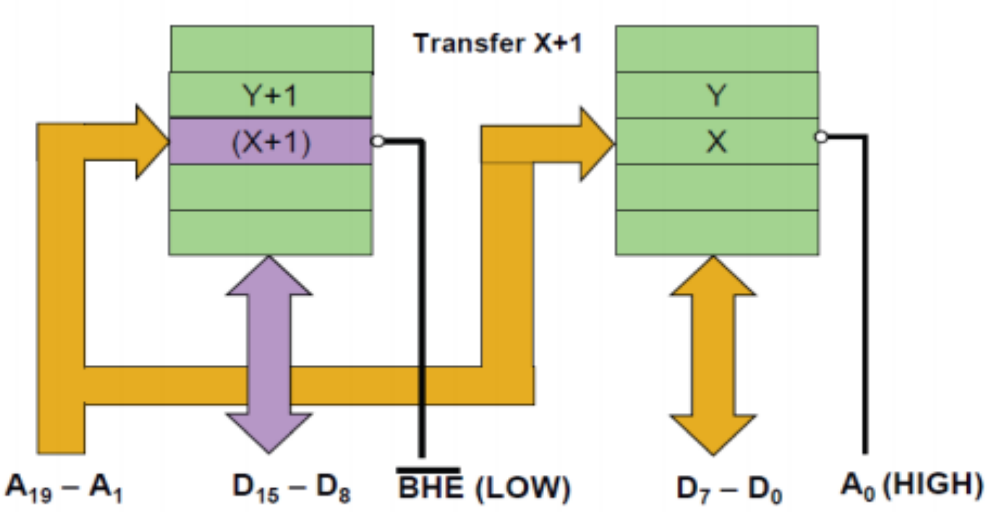
High and low memory banks of the 8086

Figure below shows how a byte transfer operation is performed to address X, an even-addressed storage location. A0 is set to logic 0 to enable the low bank of memory and $\overline{BHE}$ to logic 1 to disable the high bank. Data are transferred to or from the lower bank over data bus lines D0 - D7.



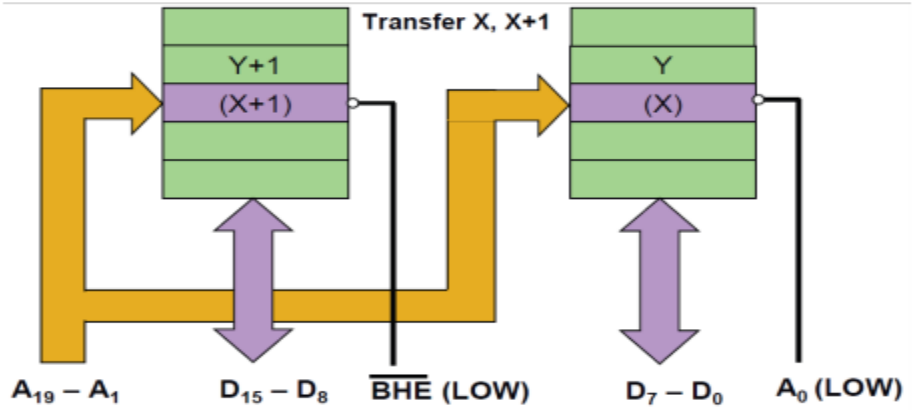
Even address byte transfer by the 8086

Figure below shows how a byte transfer operation is performed to an odd addressed storage location such as X + 1. A0 is set to logic 1 and $\overline{BHE}$ to logic 0. This enables the high bank of memory and disables the low bank. Data are transferred over bus lines D8 through D15. D8 represents the LSB.



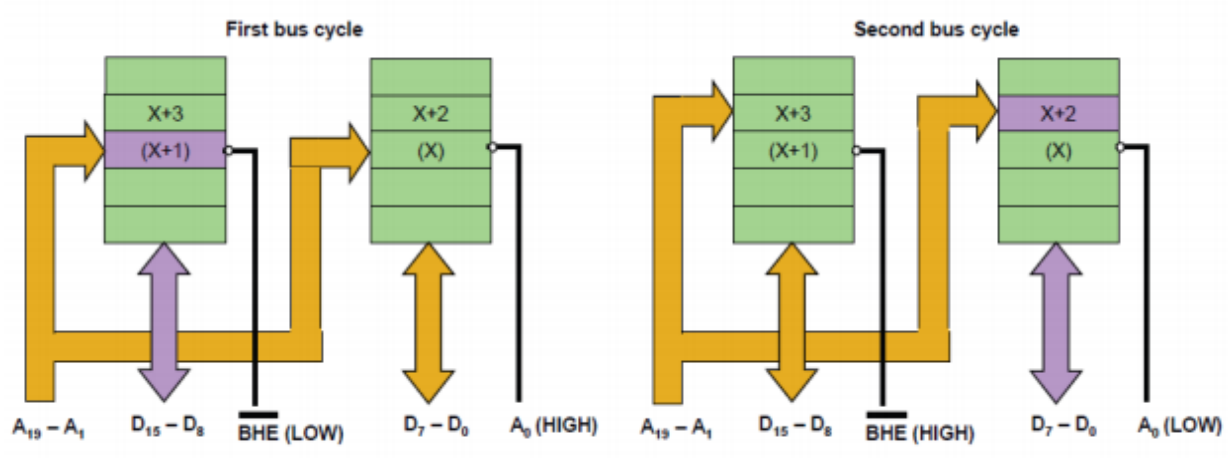
Odd address byte transfer by the 8086

Figure below illustrates how an aligned word (at even address X) is accessed. Both the high and low banks are accessed at the same time. Both A0 and $\overline{BHE}$ are set to 0. This 16-bit word is transferred over the complete data bus D0 through D15 in just one bus cycle.



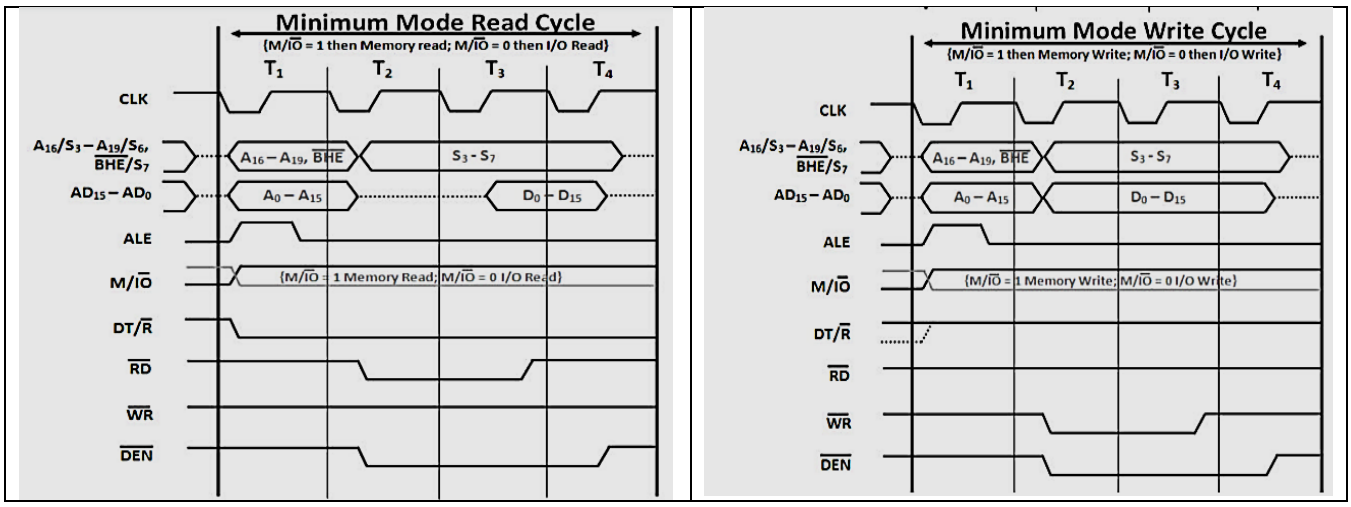
Even address word transfer by the 8086

Figure below illustrates how a misaligned word (at address X + 1) is accessed. Two bus cycles are needed. During the first bus cycle, the byte of the word located at address X + 1 in the high bank is accessed over D8 through D15. Even though the data transfer uses data lines D8 through D15, to the processor it is the low byte of the addressed data word. In the second memory bus cycle, the even byte located at X + 2 in the low bank is accessed over bus lines D0 through D7.

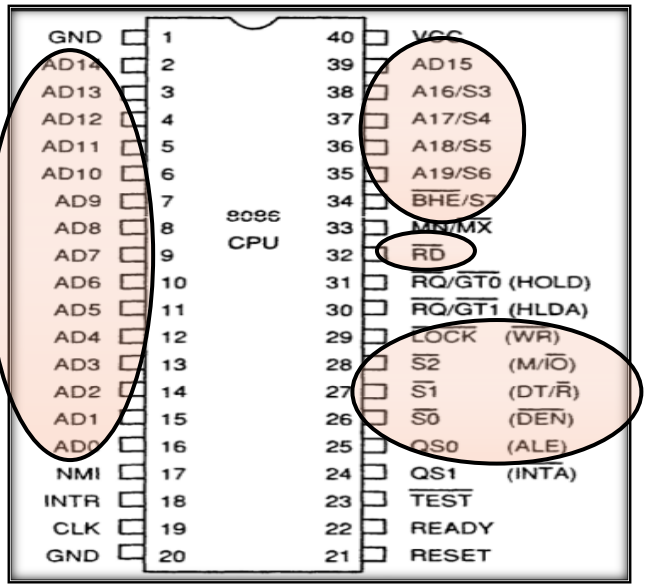


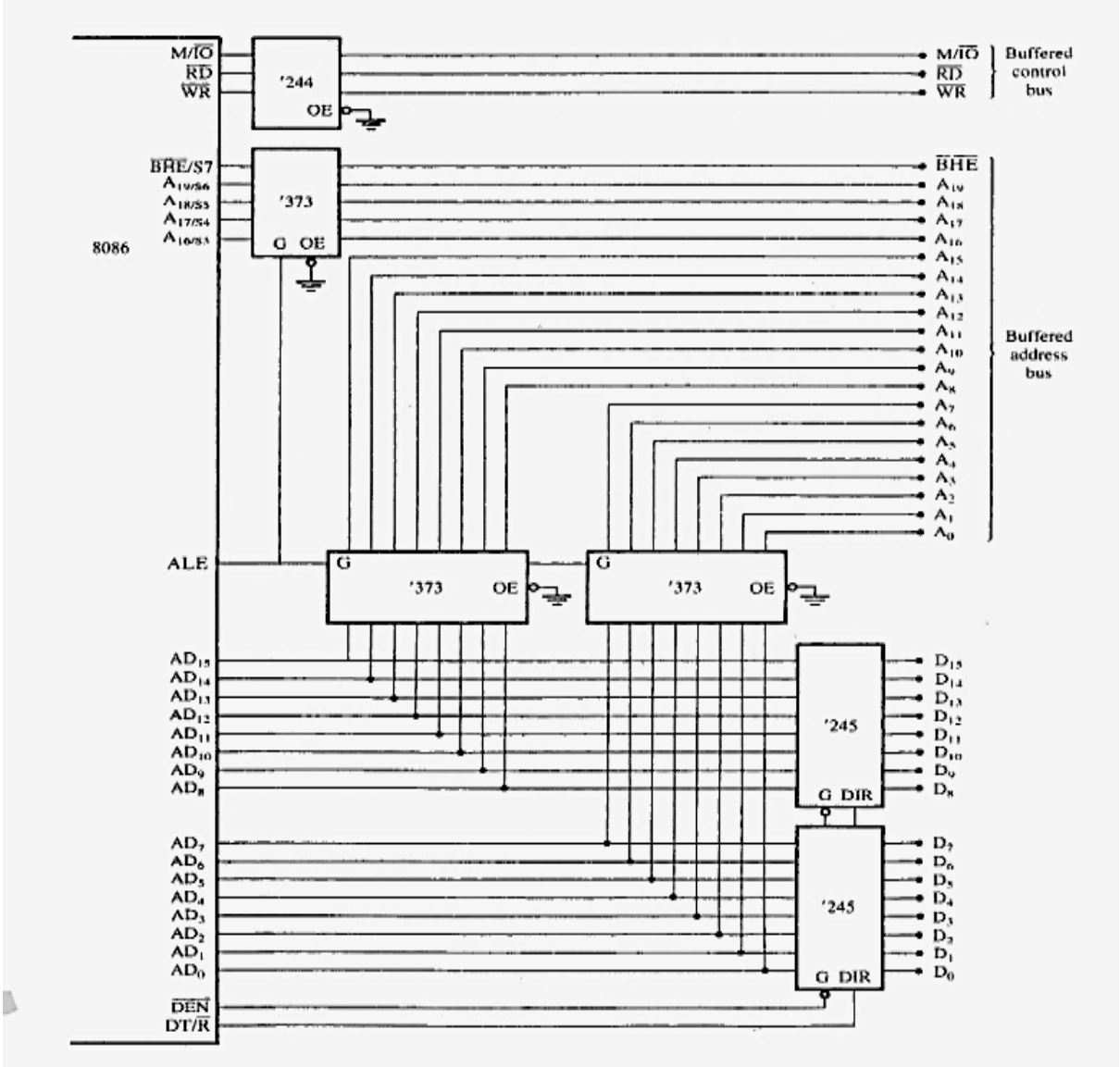
	Operation	$\overline{BHE}$	A_0	Data Lines Used
1	Read/ Write byte at an even address	1	0	$D_7 - D_0$
2	Read/ Write byte at an odd address	0	1	$D_{15} - D_8$
3	Read/ Write word at an even address	0	0	$D_{15} - D_0$
4	Read/ Write word at an odd address	0	1	$D_{15} - D_0$ in first operation byte from odd bank is transferred
	Read/ Write word at an Even address	1	0	$D_7 - D_0$ in first operation byte from odd bank is transferred

❖ **$\overline{\text{BHE}}$/S7**: The bus high enable is used to indicate the transfer of data over the higher order (D15-D8) data bus as shown in table. It goes low for the data transfer over D15-D8 and is used to derive chip selects of odd address memory bank or peripherals. $\overline{\text{BHE}}$ is low during T1 for read, write and interrupt acknowledge cycles, whenever a byte is to be transferred on higher byte of data bus. The status information is available during T2, T3 and T4. S7 is always a logic 1.



Fully buffered system bus in 8086 microprocessor





- **TEST**: input is tested by the 'WAIT' instruction. 8086 will enter a wait state after execution of the WAIT instruction and will resume execution only when the **TEST** is made low by an active hardware. This is used to synchronize an external activity to the processor internal operation.

- **RESET**: is the system set reset input signal. If this pin held HIGH for a minimum of four clocking periods causes to processor to reset itself and start execution from FFFF0H i.e reinitialize the system.
  - IP, DS, SS, ES and the instruction queue are cleared
  - CS = FFFFH

- **NIM**: is the non maskable interrupt input. A transition from low to high initiates the interrupt response at the end of the current instruction.

- **Interrupt Request (INTR) :** is a maskable interrupt input. This is a triggered input. This is sampled during the last clock cycles of each instruction to determine the availability of the request. If any interrupt request is pending (IF=1), the processor enters the interrupt acknowledge cycle (INTA becomes active) after the current instruction has complete execution.

- **HOLD**: Input signal to the processor form the bus masters as a request to grant the control of the bus. Usually used by the DMA controller to get the control of the bus.

- **HLDA** :(Hold Acknowledge) Acknowledge signal by the processor to the bus master requesting the control of the bus through HOLD. The acknowledge is asserted high, when the processor accepts HOLD.

- **Interrupt Acknowledge** ($\overline{\text{INTA}}$): This signal is used as a read strobe for interrupt acknowledge cycles. i.e. when it goes low, the processor has accepted the interrupt.