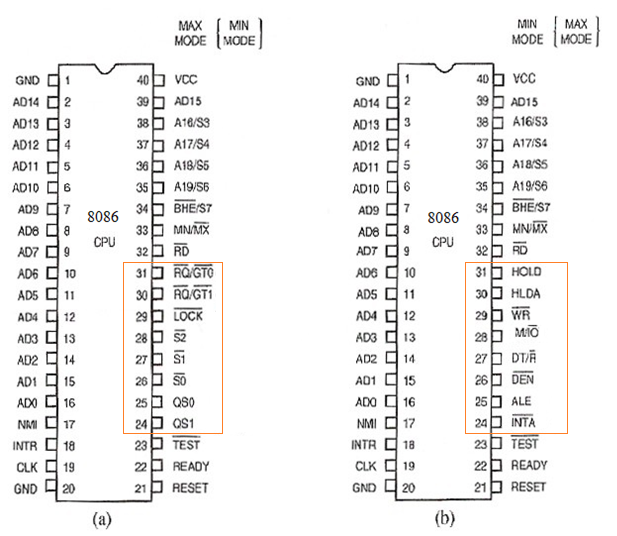
**UNIT-II**

**Hardware Features of 8086**

The first unit is mainly concerned with how a microcomputer is programmed. In that unit we worked with the programmer’s model of the 8086. This model shows features such as internal registers, number of address lines, number of data lines, and port addresses which we need to write programs. In this unit we will look at microcomputer hardware like bus signals, timing and circuit connections of an 8086.

**Pin Diagram of 8086: -**

****

**Pin Description of 8086: -** The 8086 Microprocessor has 40 pins. By usingthese pins 8086 communicate with external devices. These 40 pins are also called as 40 signals of 8086. The 40 signals are divided as follows:

1. Common Signals
2. Minimum Mode Signals
3. Maximum Mode Signals

In above classification 32-pins are common pins these pins perform same operation in both Minimum Mode as well as in Maximum Mode operations of 8086. But remaining 8-pins perform different functions in Minimum Mode and in Maximum Mode.

**1. Common Signals (32): -**

The functionality of these pins can’t change in any Mode of operation of 8086.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Pin no.** | **Type** | **Function** |
| AD15-AD0 | 2-16, 39 | I/O | **ADDRESS DATA BUS:** These lines constitute the time multiplexedmemory/IO address (T1) and data (T2, T3, Tw, T4) bus.  These lines are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge". |
| A19/S6,  A18/S5,  A17/S4,  A16/S3 | 35-38 | O | **ADDRESS/STATUS BUS:** During T1, these are the four most significantaddress lines for memory operations. During I/O operations, these lines are LOW. During memory and I/O operations, status information isavailable on these lines during T2, T3, Tw, and T4. S6 is always low.  The status of the interrupt enable flag bit (S5) is updated at the  beginning of each clock cycle. S4 and S3 indicates which segment register is presently being used for data accessing.  These lines float to 3-state OFF during local bus "hold acknowledge".   |  |  |  | | --- | --- | --- | | **S4** | **S3** | **Characteristics** | | 0(LOW) | 0 | Alternate Data (extra segment) | | 0 | 1 | Stack | | 1(HIGH) | 0 | Code or None | | 1 | 1 | Data | |
| /S7 | 34 | O | **BUS HIGH ENABLE/STATUS:** is used as during T1 clock cycle of an instruction execution. can be used in conjunction with AD0 to select memory banks. During T2, T3, Tw, and T4 clock cycles /S7 is used as S7 and it is always zero.   |  |  |  | | --- | --- | --- | |  | **A0** | **Indication** | | 0 | 0 | Whole word is received/transmitted [D15-D0data lines are active] | | 0 | 1 | Byte from/to Odd memory bank [D15-D8data lines are active] | | 1 | 0 | Byte from/to Even memory bank [D7-D0 data lines are active] | | 1 | 1 | none | |
|  | 32 | O | **Read** strobe indicates that the processor is performing memory or I/O read cycle. is active low during T2, T3 and Tw of any read cycle, and is guaranteed to remain HIGH inT2 until the 8086 local bus has floated. |
| READY | 22 | I | **READY**is the acknowledgement from the addressed memory or I/O device that it will complete the data transfer. The RDY signal from memory or I/O is synchronized by the 8284 clock generator to form READY. This signal can be used by only slowly operated external devices. If the logic level on this pin is equal to 0 then the processor will insert WAIT states between T3 and T4. If the logic level on this pin=1, the external device is ready to communicate with the processor |
| INTR | 18 | I | **INTERRUPT REQUEST**: is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the Processor should enter into an interrupt acknowledge operation. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH. |
|  | 23 | I | input is examined by the "wait" instruction.  If the input is LOW, execution continues, otherwise the processor waits in an"idle" state. |
| NMI | 17 | I | **NON-MASKABLE INTERRUPT**: is an edge triggered input which causes a type 2 interrupt. NMI is not maskable internally by software. A transition from a LOW to HIGH initiates the interrupt at the end of the current instruction. |
| RESET | 21 | I | **RESET:** causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. It restarts Execution from address FFFF0H when RESET returns LOW. RESET is internally synchronized.  **System Registers after RESET**   |  |  | | --- | --- | | **CPU component** | **Contents** | | Flags | Cleared | | Instruction Pointer | 0000H | | CS register | FFFFH | | DS register | 0000H | | SS register | 0000H | | ES register | 0000H | | Queue | Empty | |
| CLK | 19 | I | **CLOCK:** provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing. Range of clock rates  5MHz for 8086  8MHz for 8086\_2  10MHz for 8086\_1 |
| VCC | 40 |  | Vcc: is the +5V ±10% power supply pin. |
| GND | 1, 20 |  | GROUND |
| MN/ | 33 | I | MINIMUM/MAXIMUM. indicates what mode the processor is to operate in.If the logic level on this pin=0, then processor enters into MAXIMUM mode or **Multiprocessor mode**. If the logic level on this pin=1,then processor enters into MINIMUM mode or **Single Processor mode.** |

**2. Minimum mode signals (8): -**The pin MN/is connected to Vcc, and then the processor operates in this mode. The 8086 is operated in minimum mode in the systems where it is the only microprocessor on the system buses.

**Functions of 8086Minimum mode pins:**

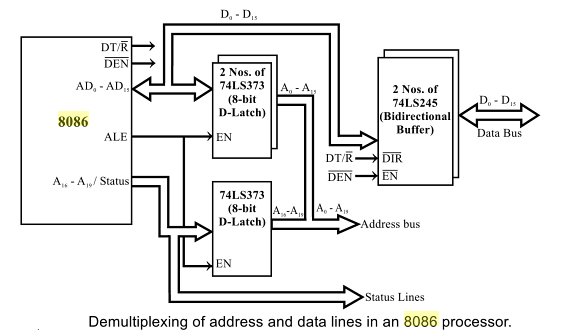
|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Pin no.** | **Type** | **Function** |
| M/ | 28 | O | This is a status line logically equivalent to in the maximum mode. When  M/ is Low, it indicates the CPU is having an I/O operation i.e. the processor is communicating with the external I/O devices.  M/ is High, it indicates the CPU is having a memory operation |
|  | 29 | O | **WRITE:** strobe indicates that the processor is performing a write memory or write I/ O cycle, depending on the state of the M/signal.  is active for T2, T3, andTw of any write cycle. |
|  | 24 | O | **:** is used as a read strobe for interrupt acknowledge cycles. It is active LOW during T2, T3, and Tw of each interrupt acknowledge cycle. |
| ALE | 25 | O | **ADDRESS LATCH ENABLE**: signal indicates the availability of the valid address on the multiplexed lines and is connected to the strobe input of the address latches.  It is a HIGH pulse active during clock low of T1 of any bus cycle. Note that ALE is never ﬂoated. |
| DT/ | 27 | O | **DATA TRANSMIT/ RECEIVE**: is needed in a system that desires to use a data bus transceiver. It is used to control the direction of data flow through the transceiver.  Logically, DT/ is equivalent to in the maximum mode, and its timing is the same as for M/(T = HIGH, R = LOW). |
|  | 26 | O | **DATA ENABLE**: is used to enable the transceivers.  For a read or cycle, it is active from the middle of T2 until the middle of T4, while for a write cycle, it is active from the beginning of T2 until the middle of T4. |
| HOLD  HLDA | 31  30 | I  O | **HOLD and HLDA** Signals are called DMA signals.The processor receives DMA request signals from the external DMA controller by using HOLD pin.The processor gives highest priority to the DMA request as compared to any other hardware Interrupt requests. Processor sends its acceptance information to DMA controller by using HLDA signal.It also gives control over the system bus to DMA controller. |

**3. Maximum Mode Signals(8): -**The pin MN/is connected to GND, and then the processor operates in this mode. The 8086 is operated in maximum mode in a system which has two or more microprocessors sharing the same buses.

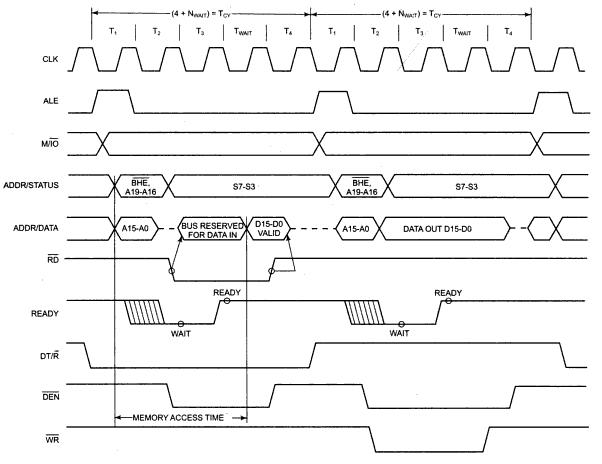
|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Pin no.** | **Type** | **Function** |
| **,,** | 26–28 | O | **STATUS**:active during T4,T1,and T2and is returned to the passive state (1,1,1) during T3 or TW. the status is used by the 8288 Bus Controller to generate all memory and I/O access control signals. Any change by S2,S1,or S0 during T4 isused to indicate the beginning of the bus cycle, and return to the passive state in T3 is used to indicate the end of a bus cycle.   |  |  |  |  | | --- | --- | --- | --- | |  |  |  | **Function** | | 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 | |
| **QS0,QS1** | 25,24 | O | **QUEUESTATUS**:The queue status is valid during the clock cycle, after which the queue operation is performed QS1 and QS0 provide status to allow external tracking of the internal 8086 instruction queue.   |  |  |  | | --- | --- | --- | | **QS1** | **QS0** | **Indication** | | 0 | 0 | No operation (all six registers of instruction queue are full) | | 0 | 1 | First byte of Opcode from Queue | | 1 | 0 | The instruction queue is empty (all IQ registers are cleared due to branching operation) | | 1 | 1 | Subsequent Byte from the Queue. | |
|  | 29 | O | Pin indicates that other system bus master will be prevented from gaining the system bus, while the signal is low. The signal is activated by the ‘LOCK’ prefix instruction and remains active until the completion of the next instruction. When the CPU is executing a critical instruction which requires the system bus, the LOCK prefix instruction ensures that other processors connected in the system will not gain the control of the bus. |
| **/;/** | 30, 31 | I/O | **(Request/Grant Signals): -**These pins are used by the other local bus master in Maximum Mode, to force the processor to release the local bus at the end of the processor current bus cycle. Each of the pin is bidirectional with RQ/GT0 having higher priority than RQ/GT1. Request/Grant sequence is as follows:  1.A pulse of one clock wide from another bus master requests the bus access to 8086.  2.During T4(current) or T1(next) clock cycle, a pulse one clock wide from 8086 to the requesting master, indicates that the 8086 has allowed the local bus to float and that it will enter the ‘hold acknowledge’ state at next cycle. The CPU bus interface unit is likely to be disconnected from the local bus of thesystem.  3.A one clock wide pulse from another master indicates to the 8086 that the hold request is about to end and the 8086 may regain control of the local bus at the next clock cycle. Thus each master to master exchange of the local bus is a sequence of 3 pulses. There must be at least one dead clock cycle after each bus exchange. The request and grant pulses are active low.For the bus request those are received while 8086 is performing memory or I/O cycle, the granting of the bus is governed by the rules as in case of HOLD and HLDA in minimum mode. |

**Demultiplexing of Address/Data Lines in an 8086 Processor**

* In order to Demultiplex the address/data lines (of the processor), the processor provides a signal called ALE (Address Latch Enable). The ALE is asserted high and then low by the processor at the beginning of every bus cycle. At the same time, the address is given out through AD0– AD15 lines and A16- A19/status lines. Demultiplexing of address/data lines and address/status lines using 8-bit D-latch 74LS373 is shown in Fig.



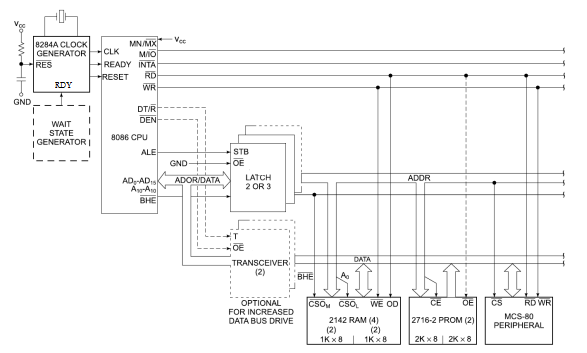
* The ALE is connected to the Enable Pin (EN) of the external 8-bit latches. When ALE is asserted high and then low, the addresses are latched into the output lines of the latch. It holds the address until the next bus cycle. After latching the address, the AD0- AD15 lines are free for data transfer and A16– A19/status lines are free for carrying status information. The first T-state of every bus cycle is used for address latching in 8086 and the remaining T states are used for reading or writing operation.

****

**Fig: Basic 8086 system timings**

**Minimum Mode operation of 8086 Processor: -**In a minimum mode 8086 system, the microprocessor 8086 is operated in minimum mode by connecting its MN/ pin to logic ‘1’ it means the pin is connected to the Vcc .

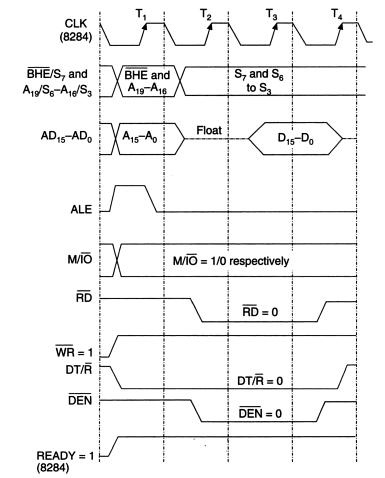
* In this mode, all the control signals are given out by the microprocessor chip itself. There is a single microprocessor in the minimum mode system.
* The remaining components in the system are latches, transceivers, clock generator, memory and I/O devices. Some type of chip selection logic may be required for selecting memory or I/O devices, depending upon the address map of the system.
* Latches are generally buffered output D-type flip-flops like 74LS373. They are used for separating the valid address from the multiplexed address/data signals and are controlled by the ALE signal generated by 8086.
* Transceivers are the bidirectional buffers and sometimes they are called as data amplifiers. They are required to separate the valid data from the time multiplexed address/data signals. They are controlled by two signals namely, and DT/.
* The signal indicates the direction of data, i.e. from or to the processor. The system contains memory for the monitor and users program storage.



**Figure shows the minimum mode configuration of 8086**

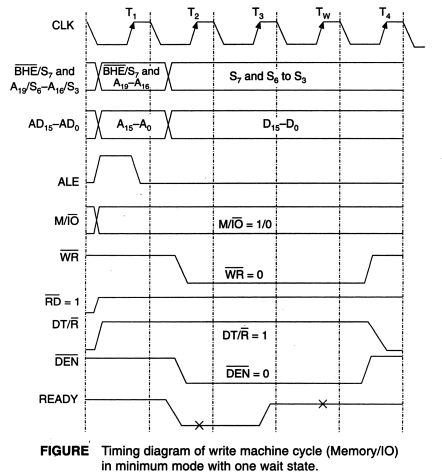
* Usually, EPROMs are used for monitor storage, while RAM for user’s program storage. A system may contain I/O devices.
* The working of the minimum mode configuration system can be better described in terms of the timing diagrams rather than qualitatively describing the operations.

**Read Timing Diagram for Minimum Mode Operation of 8086: -**



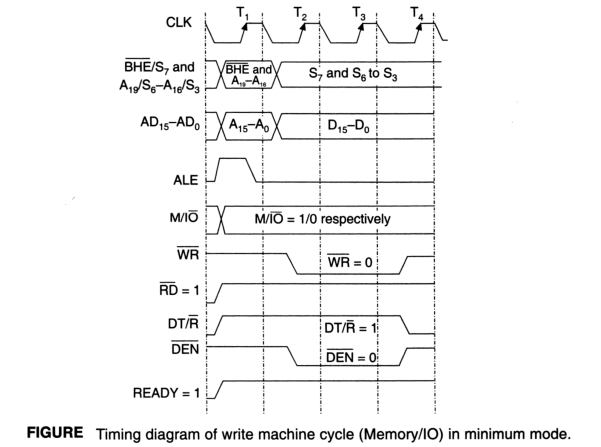
**Figure shows the timing diagram ofa read machine cycle in minimum mode**

1. The clock waveform at the top, CLK represents the clock signal sent to 8086 from 8284 clock generator.
2. One cycle of this clock is called a state. For reference purpose a state is measured from the falling edge of one clock pulse to the falling edge of the next clock pulse.
3. A basic microprocessor operation such as reading a byte from memory or writing a byte to a port is called a Machine cycle. A Machine cycle consists of several states.
4. The time a microprocessor requires to fetch and execute an entire instruction is referred to as an Instruction cycle. An Instruction cycleconsists of one or more Machine cycles.
5. The read cycle begins in T1 with the assertion of address latch enable (ALE) signal and also M / signal. ALE signal is connected to the enable input (STB) of the latches, so these latches will be enabled when ALE is high.
6. After the 8086 asserts ALE high, it sends out address of the memory location that it wants to read on the multiplexed Address-data and address-status lines.
7. The negative going edge of the ALE signal disables the latches. The address held on the latch outputs travels along the address bus to memory and port devices.
8. As the address information is held on the latches, the 8086 no longer needs to send out the address. Therefore at T2, the address is removed from ADDR/DATA lines and bus is tri-stated. At about the same time, the 8086 also removes the and A16-A19 information from the upper lines and sends out some status information on those lines.
9. The read () control signal is also activated in T2. The read () signal causes the address device to enable its data bus drivers.



1. If the READY input is low on or before T3 in a machine cycle then 8086 will insert one or more wait states between T3 and T4 in that machine cycle.
2. The 8086 will continue inserting WAIT states until the READY input is made high again.
3. During T1 of the machine cycle the 8086 asserts DT/ low to put the data buffers in the receive mode.
4. After the 8086 finishes using the data bus to send out the lower 16 address bits, it asserts low to enable the data bus buffers. The data put on the data bus by an addressed port or memory will then be able to come in through the buffers to the 8086 on the data bus.
5. To complete the cycle the 8086 brings the line high again.

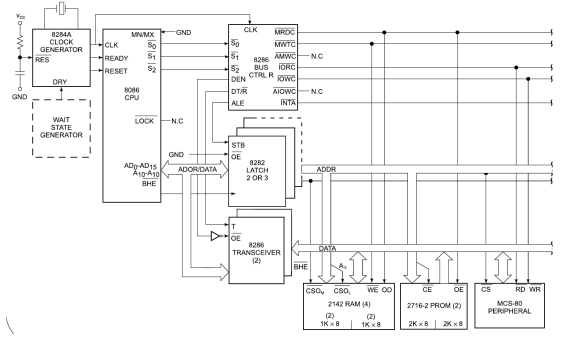
**Write Timing Diagram for Minimum Mode Operation of 8086: -**

****

1. The 4 processor clock cycles are called T states. Four cycles is the shortest time that the processor can use for carrying out a write or an output cycle.
2. At the beginning of T1, the processor outputs A16/S3-A19/S6, AD0 to AD15 and /S7.
3. The transitions of ALE signal from low to high, thereby allowing the address to pass through the transparent latches. The address, along with the signal is latched when ALE goes low, providing the latched address A0 to A19.
4. During T2, the processor removes the address and data, S3 to S6 status is output on the upper 4 address/status lines of the processor. Output data is driven out on the AD0 to AD15 lines.
5. Data bus transceivers are enabled away from the microprocessor (the WRITE direction) by the DT/ and signals.
6. The or signal is asserted at the beginning of T2.
7. The signals are maintained during T3.
8. During T4, the memory and I/O control lines are de-asserted.

**Maximum Mode operation of 8086 Processor: -** In the maximum mode, the 8086 is operated by connecting its MN/ pin to logic ‘0’ it means the pin is connected to the ground.

* In this mode, the processor derives the status signal **,,** . Another chip called bus controller derives the control signal using this status information.
* In the maximum mode, there may be more than one microprocessor in the system configuration.The components in the system are same as in the minimum mode system.



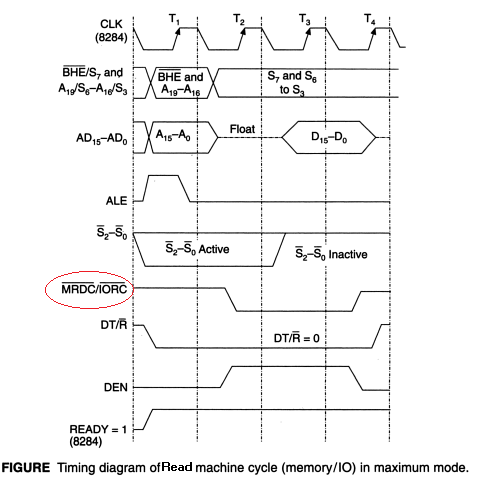
**Figure: Shows the maximum mode configuration of 8086**

The block diagram of 8288 consists of four blocks. These blocks are:

1. Status decoder
2. Command signal generator
3. Control logic
4. Control signal generator.

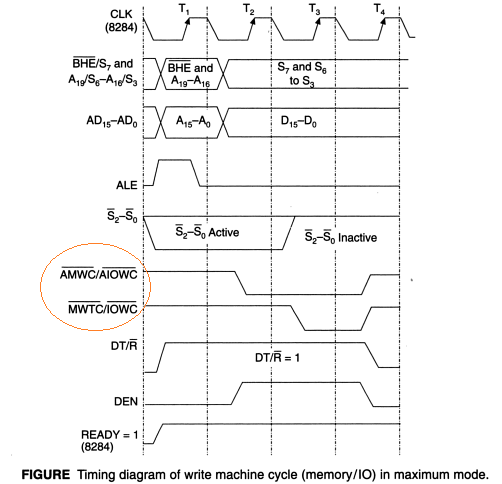
* The 8288 bus controller receives seven input signals. Out of these seven input signals, four inputs (,,and CLK) are from the 8086. The command logic decodes the three 8086 status lines ,and) to determine what command is to be issued. The CLK input provides the desired clock frequency to the bus controller.
* The remaining three input signals are the control signals CEN, IOB and . These control signals determine the operating modes of 8288.
* It derives the outputs ALE, DEN, DT/, , , , , and . The, IOB and CEN pins are especially useful for multiprocessor systems.
* The basic function of the bus controller chip IC8288 (DEMUX) is to derive control signals like and (for memory and I/O devices), , DT/, ALE etc. using the information by the processor on the status lines.
* and IOB are generally grounded. CEN pin is usually tied to +5V. The significance of the MCE/ output depends upon the status of the IOB pin.
* If IOB is grounded, it acts as master cascade enable to control cascade 8259A, else it acts as peripheral data enable used in the multiple bus configurations.
* pin used to issue two interrupt acknowledge pulses to the interrupt controller or to an interrupting device.
* , are I/O read command and I/O write command signals respectively.These signals enable an I/O interface to read or write the data from or to the address port.
* The , are memory read command and memory write command signals respectively and may be used as memory read or write signals.All these command signals instructs the memory to accept or send data from or to the bus.
* For both of these write command signals, the advanced signals namely and are available.

**Memory Read Timing Diagram for Maximum Mode Operation of 8086: -**

****

1. The 4 processor clock cycles are called T states. Four cycles is the shortest time that the processor can use for carrying out a read or an input cycle.
2. At the beginning of T1, the processor outputs A16/S3-A19/S6, AD0 to AD15 and /S7.
3. The transitions of ALE signal from low to high, thereby allowing the address to passthrough the transparent latches. The address, along with the signal is latched when ALE goes low, providing the latched address A0 to A19.
4. During T2, the processor removes the address and data, S3 to S6 status is output on the upper 4 address/status lines of the processor.
5. The AD0-AD15 signals are floated as inputs, waiting for data to be read.
6. Data bus transceivers are enabled towards the microprocessor (the READ direction) by the DT/ and signals.
7. The or signal is asserted.
8. The signals are maintained during T3. At the end of T3 the microprocessor samples the input data.
9. During T4, the memory and I/O control lines are de-asserted.

**Memory Write Timing Diagram for Maximum Mode Operation of 8086: -**



1. The 4 processor clock cycles are called T states. Four cycles is the shortest time that the processor can use for carrying out a write or an output cycle.
2. At the beginning of T1, the processor outputs A16/S3-A19/S6, AD0 to AD15and /S7.
3. The 8288 bus controller transitions the ALE signal from low to high, thereby allowing the address to pass through the transparent latches. The address, along with the BHE signal is latched when ALE goes low, providing the latched address A0 to A19.
4. During T2 the processor removes the address and data, S3 to S6 status is output on the upper 4 address/status lines of the processor.
5. Output data is driven out on the AD0 to AD15 lines.
6. Data bus transceivers are enabled away from the microprocessor (the WRITE direction) by the DT/ and signals.
7. The or signal is asserted at the beginning of T3.
8. The signals are maintained during T3.
9. During T4 the memory and I/O control lines are de-asserted.

**Interfacing: -**

“Interfacing” means designing of Hardware and writing of software, for the processor is communicating with the external devices (I/O devices and memory devices). There are two types of interfacing devices:

**i) Non-programmable Interfacing devices:**To work these devices there is no need of software code in the Program.

Ex: - 74LS373 Octal Latches,74LS245 Bidirectional buffers, 74LS138/139 Decoders etc.

**ii)Programmable Interfacing Devices:** - To work these devices there is a need of software code in the Program.

Ex: - 8255 PPI (Programmable Peripheral Interface), 8251 USART (Universal Synchronous Asynchronous Receiver and Transmitter), 8259 PIC (Programmable Interrupt Controller), 8237/57 DMAC (DMA Controller).

**Address Decoding**

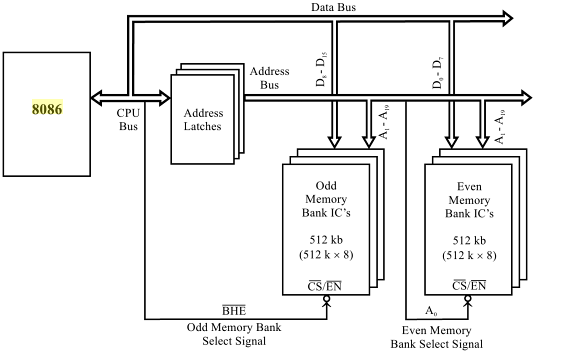
Address decoding is the process generation of Chip-select Signals for external devices by using unused address lines of the processor. There are 2 types of Address Decoding Methods.

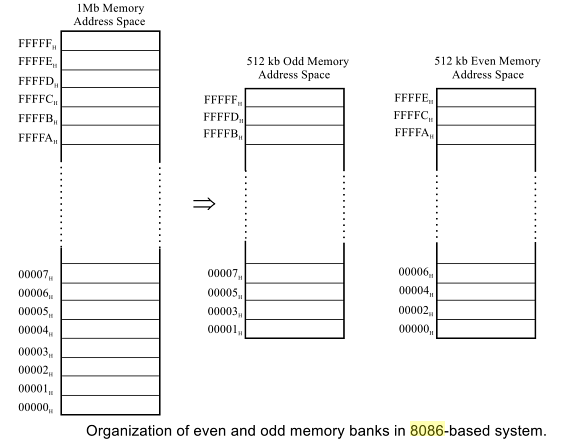
**1)Absolute Address Decoding Method: -** In this address decoding method,all the unused address lines of the processor are involved in the generation of chip-select signals for the external devices.By using this method each and every I/O device or a memory location have only one address i.e. Unique Address.

**2)Partial Address Decoding Method: -** In this address decoding method, only some of the unused address lines of the processor are involved in the generation of chip-select signals for the external devices. By using this method each and every I/O device or a memory location have more than one address, because some address lines logic levels we don’t know exactly.

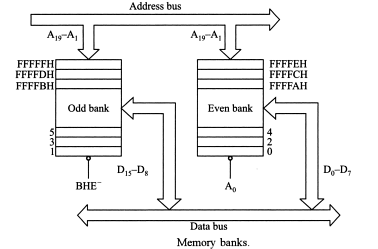
**PHYSICAL MEMORY ORGANIZATION IN 8086-BASED SYSTEM**

* The 8086 microprocessor provides a 20-bit address to memory. The memory is organized as a linear array of up to 1 MB, addressed from 00000H to FFFFFH.
* Physically, the one mega-byte (1Mb) of 8086 addressable memory space is divided into two banks: Even memory bank and Odd memory bank. Each bank will have an addressable space of 512 kilo bytes (512 kb). In 8086-based system the lower eight lines of data bus, Do- D7, are connected to even bank memory ICs and the upper eight lines of data bus, D8- D15 are connected to odd bank memory ICs. The processor provides two enable signals and A0. These two signals selectively allow reading from or writing into either an odd byte location, even byte location, or both. is used to enable the odd bank and the A0 address line is used to enable the even bank.





* A microprocessor-based system requires both EPROM and RAM. Hence, the available memory space has to be divided between EPROM and RAM. This choice depends on the systemdesigner as well as on the application for which the system is designed. For proper systemfunctioning, the system designer should allot equal address space in odd and even bank for bothEPROM and RAM.
* Some systems may require large memory space and so full memory space is utilized. But insome systems, the memory requirement may be less and in this case the full memory space is not utilized. When full memory space is not utilized for memory, then the unused memory addressescan be used for addressing I/O devices. Such I/O devices are called memory-mapped I/O devices and they can be accessed similar to that of memory device.
* The required EPROM memory capacity of the system can be implemented in two ICs (one for even and the other for odd bank) or in multiple ICs. Similarly, the RAM capacity of the system can be implemented in two ICs or in multiple ICs. This choice depends on the availability of memory IC and the system designer. Some example memory organisations for 8086 processor based-system are discussed in this section.



* **Case i : Byte access from even bank**

For read/write operation of a byte in even memory address, A0 is asserted low and is asserted high (i.e., A0 = 0 and = 1). Now the even bank alone is enabled and the data transfer take place through D0-D7 data lines.

* **Case ii : Byte access from odd bank**

For read/write operation of a byte in odd memory address, A0is asserted high and is asserted low (i.e., A0 = 1and = 0). Now odd bank alone is enabled and the data transfer take place through D8-D15 data lines.

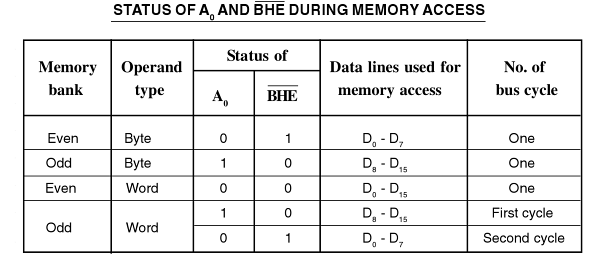
* **Case iii : Word access from even boundary**

For read/write operation of a word (16-bit) in even boundary (i.e., low byte in even address and high byte in next address (odd address)), both A0and are asserted low (i.e., A0 = 0 and = 0). Now both the memory banks are enabled simultaneously and the processor read/write the 16-bit operand in one bus cycle through D0-D15 data lines.

* **Case iv : Word access from odd boundary**

For read/write operation of a word (16-bit) in odd boundary (i.e., low byte in odd address and high byte in next address (even address)), the processor executes two bus cycles to read/write the word (16-bit) operand. In the first bus cycle A0 is asserted high and is asserted low (i.e., A0 = 1 and = 0). Now the odd bank alone is enabled and the low byte of 16-bit operand is read/write through D8-D15data lines. In the second bus cycle A0 is asserted low and is asserted high (i.e., A0 = 0 and = 1). Now the even bank alone is enabled and the high byte of 16-bit operand is read/write through D0-D7 data lines.

**The status of A0 and for byte and word memory access are listed in Table below**



**I/O ADDRESSING**

The I/O devices in 8086 may be interface with 8086 in two different ways, i.e. I/O mapped I/O and memory mapped I/O. These two methods of I/O interfacing give rise to two different I/O address spaces.

**I/O Mapped I/O**

The I/O Mapped I/O is also known as Isolated I/O. In this case the I/O devices are treated as I/O devices only. The I/O Mapped I/O interfacing may be of two types, viz. I/O devices having 8-bit port addresses and I/O devices having 16-bit addresses. In the first case, microprocessor 8086 can address up to 256 input devices as well as 256 output devices. In the second case, microprocessor 8086 can address up to 64K input devices as well as 64K output devices. The address of the I/O device is called the port address. The control signals used in this space are and . In this technique of interfacing, there are only two instructions for data transfer between an I/O device and microprocessor. These instructions are IN and OUT. When the I/O devices are having 8-bit port addresses then direct addressing mode will be used, whereas when the I/O devices are having 16-bit port addresses, the indirect addressing mode will be used and by default DX register is used to hold the I/O address.

**IN instruction**

The various formats of the IN instruction are:

IN AL, 8-bit port address

IN AX, 8-bit port address

IN AL, DX

IN AX, DX

These instructions are used to transfer data from the input device to the microprocessor. The data is transferred to the accumulator only. In the first instruction the port address is of 8-bit and the data transfer is also of 8-bit.

In the second case the port address is of 8-bit but the data to be transferred is of 16-bit. The data from the port number mentioned in the instruction will go to the AL register and the data from the next port address will go to the AH register. It is to be noted that I/O ports are also divided as even and odd ports just like that of memory.

In the third and fourth cases the port address is of 16-bit and indirect addressing, using DX register is used. In the third instruction data from the input port, mentioned in DX register will come to the AL register. In the fourth instruction data from the input port, mentioned in DX register will come to the AL register and data from the input port DX + 1 will come to the AH register.

To read data from an input port the microprocessor sends port address on the A0-A7 lines or A0-A15 lines. This 8- or 16-bit address is decoded and used to select IO device. Then the microprocessor generates the control signal, i.e. M/= 0 and = 0. The peripheral device, on receiving this control signal, will send the data on to the data bus.

**OUT instruction**

The various formats of the IN instruction are:

OUT 8-bit port address, AL

OUT 8-bit port address, AX

OUT DX, AL

OUT instruction is used to transfer the data from the microprocessor to the output device. These four formats of the OUT instruction are the same as that of the IN instruction with a difference that here the data is transferred from accumulator to the output port.

The data is transferred from the accumulator only. To write data to an output port the microprocessor sends port address on the A0—A7 lines or A0—A15 lines. This 8- or 16-bit address is decoded and used to select one IO device. Then the microprocessor generates the control signal, i.e. . M/= 0 and = 0. The peripheral device, on receiving this control signal, will read the data from the data bus.

**Memory Mapped I/O**

In memory mapped I/O, the I/O devices are allotted 20-bit addresses. This can be understood as that we have replaced a memory register by an I/O device. Now the I/O device has the address of a memory. Theoretically in memory mapped I/O we can interface 1M input output device with 8086. The I/O locations in memory mapped I/O is considered as a memory location. The memory related control signals (and ) are used to read and write from the I/O. All the memory related instructions are used to access an I/O. Moreover, arithmetic and logical operations can now directly performed with an I/O.

**Difference between I/O mapped I/O and memory mapped I/O**

|  |  |
| --- | --- |
| **I/O mapped I/O** | **Memory mapped I/O** |
| 8- or 16-bit port address. | 20-bit port address. |
| Maximum 256K I/O ports can be interfaced using direct addressing and 64K I/O ports can be interfaced using indirect addressing. | 1M I/O ports can be interfaced. |
| Less address decoding hardware is required as only 8 address bits are to be decoded. | More address decoding hardware is required because in this case 16 address bits are to be decoded. |
| and will be the control signals. IN and OUT are the only instructions for data communication between the microprocessor and the I/O devices. | and are the control signals. Any memory related instructions, such as MOV etc. can be used for data communication between the microprocessor and the I/O devices. |
| I/O can communicate only with the accumulator. | I/O can communicate with any of the registers, just Like a memory register. |
| I/O and memory can have the same address. | I/O and memory cannot have the same address. |
| There is no effect on memory size. | Memory size will reduce. |
| Arithmetic and logic operations cannot be performed directly on the data at I/O. | Arithmetic and logic operations can be performed directly on the data at I/O. |

**MEMORY TYPES**

1. ROM (Read only memory):

* ROM is the type of memory that does not lose its contents when power is turned off. It is also called non-volatile memory.

2. PROM (Programmable Memory):

* User programmable (one-time programmable) memory.
* If the information burned into PROM is wrong, it needs to be discarded since internal fuses are blown permanently.
* Special equipment needed: ROM burner or ROM programmer.

3. EPROM (Erasable Programmable ROM) 2,000 times:

* Allows making changes in the contents of PROM after it is burned.
* One can program the memory chip and erase it thousands of times.
* Erasing its contents can take up to 20 minutes; the entire chip is erased.
* All EPROM chips have a window that is used to shine ultraviolet (UV) radiation to erase its contents.
* Also referred to as UV-EPROM.

4. EEPROM (Electrically Erasable ROM) 500,000 times:

* Method of erasure is electrical.
* Moreover, one can select which byte to be erased.
* Cost per bit is much higher than for UV-EPROM.

5. Flash memory EPROM:

* First, the process of erasure of the entire contents takes less than a second, or one might say in a flash, hence its name: flash memory.
* When flash memory's contents are erased, the entire device is erased.
* Even though flash memories are writeable, like EPROM's. They find their widest use in microcomputer systems for storage of firmware.

6. RAM (Random Access Memory) Infinite times:

* RAM memory is called volatile memory since cutting off the power to the IC will mean the loss of data
* Also referred to as R/WM (Read and Write Memory).

**RAM MEMORY TYPES**

1. SRAM (Static RAM):

* Storage cells are made of flip-flops and, therefore, they do not require refreshing to keep their data.
* A cell handling one bit requires 6 or 4 transistors each, which is too many.
* SRAMS are widely used for cache memory and battery-backed memory systems.

2. DRAM (Dynamic RAM):

* + Uses MOS capacitors to store a bit.
  + Requires constant refreshing due to leakage (every 2ms-4ms).
  + Advantages:
* High density (capacity).
* Cheaper cost per bit.
* Lower power consumption.
  + Disadvantages:
* While it is being refreshed, data cannot be accessed.
* Larger access times.
* Too many pins due to large capacity.

**Address Decoding**: - Address decoding is the process generation of Chip-select Signals for external devices by using unused address lines of the processor. There are 2 types of Address Decoding Methods.

**1)Absolute Address Decoding Method: -** In this address decoding method,all the unused address lines of the processor are involved in the generation of chip-select signals for the external devices.By using this method each and every I/O device or a memory location have only one address i.e. Unique Address.

**2)Partial Address Decoding Method: -** In this address decoding method, only some of the unused address lines of the processor are involved in the generation of chip-select signals for the external devices. By using this method each and every I/O device or a memory location have more than one address, because some address lines logic levels we don’t know exactly.

**Memory Interfacing**

Semiconductor memories are organised as two dimensional arrays of memory locations. For example, 4K x 8 or 4K byte memory contains 4096 locations, where each location contains 8-bit data and only one 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 conductors called 'data bus'.

Obviously for addressing 4K bytes of memory twelve address lines are required. 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 = Log2N.

Thus if the microprocessor has n address lines, then it is able to address at the most N locations of memory, where 2n= N. However, if out of N locations only P memory locations are to be interfaced, then the least significant p address lines out of the available n lines can be directly connected from the microprocessor to the memory chip while the remaining (n-p) higher order address lines may be used for address decoding (as inputs to the chip selection logic). The memory address depends upon the hardware circuit used for decoding the chip select (). The output of the decoding circuit is connected with the pin of the memory chip.

The general procedure of static memory interfacing with 8086 is briefly described as follows:

I. 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', as described in memory organisation.

2. Connect available memory address lines of memory chips with those of the microprocessor and

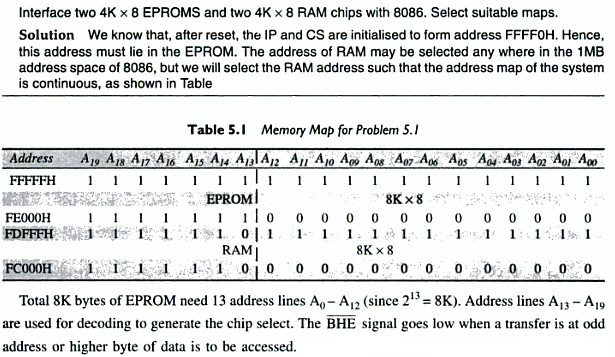
also connect the memory and inputs to the corresponding processor control signals.

Connect the 16-bit data bus of the memory bank with that of the microprocessor 8086.

3. The remaining address lines of the microprocessor and A0 are used for decoding the

required chip select signals for the odd and even memory banks. The of memory is derived from the O/P of the decoding circuit.

**Problem 1:**



**Problem 3**

