I/O interface circuits

TYPES OF INPUT/OUTPUT

The input/output system of the microprocessor allows peripherals to provide data or receive results of processing the data. This is done using I/O ports. The 8086 microcomputers can employ two different types of input/output (I/O):

- isolated I/O
- memory-mapped I/O.

These I/O methods differ in how I/O ports are mapped into the 8086's address spaces. Some microcomputer systems employ both kinds of I/O- that is, some peripheral ICs are treated as isolated I/O devices and others as memory- mapped I/O devices. Let us now look at each of these types of I/O.

Isolated Input/ Output Method:

When using isolated I/O in a microcomputer system, the I/O devices are treated separate from memory. This is achieved because the software and hardware architectures of the 8086 support separate memory and I/O address spaces. Figure-1- illustrates these memory and I/O address spaces.

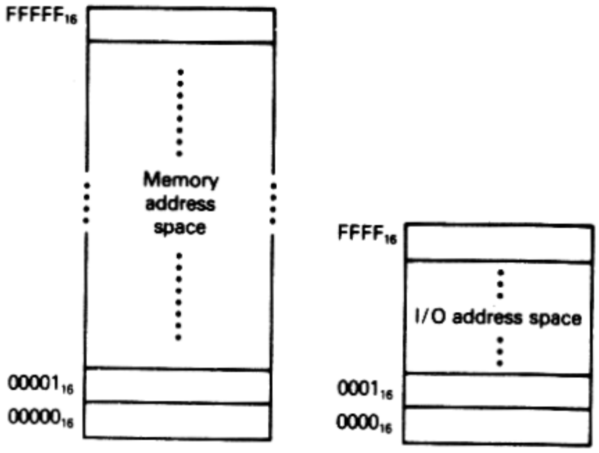


Fig-1-

In our study of 8086 software architecture , We found that information in memory or at I/O ports is organized as bytes of data; that the memory address space contains IM consecutive byte addresses in the range 00000H through FFFFFH; and that the I/O address space contains 64K consecutive byte addresses in the range 0000H through FFFFH.

Figure -2-shows a more detailed map of this I/O address space.

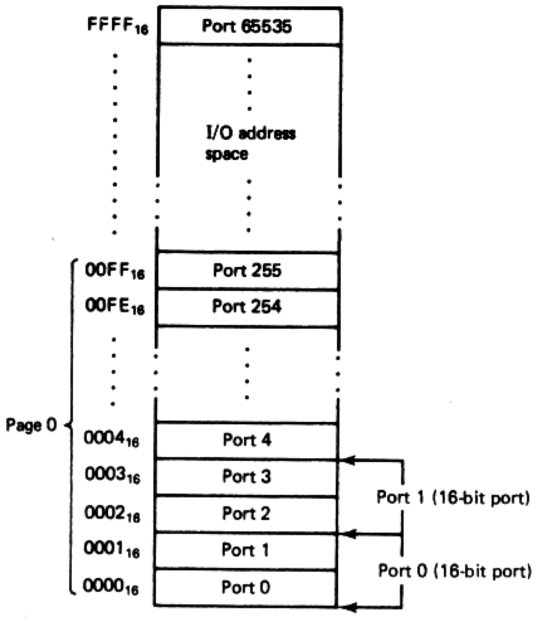


Fig-2-

Here we find that the bytes of data in two consecutive I/O addresses could be accessed as word-wide data. For instance, I/O addresses 0000H, 0001H, 0002H, and 0003H can be treated as independent **byte**-wide I/O ports. Ports 0 and 1 may be considered together as **word**-wide port 0.

Note that the part of the I/O address space in Fig.-2- from address 0000H through 00FFH is referred to as page 0. Certain I/O instructions can only perform operations to ports in this part of the address range. Other I/O instructions can input or output data for ports anywhere in the I/O address space.

This isolated method of I/O offers some advantages: First, the complete 1Mbyte memory address space is available for use with memory. Second, special instructions have been provided in the instruction set of the 8086 to perform isolated I/O input and output operations.

A disadvantage of this type of I/O is that all input and output data transfers must take place between the AL or AX register and the I/O port.

Memory—Mapped Input/ Output Method:

I/O devices can be placed in the memory address space of the microcomputer as well as in the independent I/O address space. In this case, the MPU looks at the I/O port as though it is a storage location in memory. For this reason, the method is known as memory-mapped I/O. When I/O is configured in this way, instructions that affect data in memory are used instead of the special input/output instructions. This is an advantage in that many more instructions and addressing modes are available to perform I/O operations. In addition, I/O transfers can now take place between an I/O port and an internal register other than just AL or AX. However, this also leads to a disadvantage. That is, the memory instructions tend to execute slower than those specifically designed for isolated I/O. Therefore, a memory—mapped I/O routine may take longer to perform than an equivalent program using the input/output instructions.

Another disadvantage of using this method is that part of the memory address space is lost.

ISOLATED INPUT/OUTPUT INTERFACE

The way in which the MPU deals with input/output circuitry is similar to the way in which it interfaces with memory circuitry.

- There is an I/O interface circuitry for minimum mode. This interface also use the signals ALE, $\overline{BHE}$, $M/\overline{IO}$, $\overline{RD}$, $\overline{WR}$, $DT/\bar{R}$ and $\overline{DEN}$.
- There is an I/O interface circuitry for maximum mode. This interface uses the 8288 bus controller.
- Through this I/O interface, the MPU can input or output data in **bit**, **byte** or **word**.
- Unlike the memory interface, just the 16 least significant lines of the address bus (A15through A0) are used.
- The logic levels of signals A0 and $\overline{BHE}$ determine whether data are input/ output for an odd-addressed byte-wide port, even-addressed byte-wide port, or a word-wide port.
- Input/output operations are performed using special input and output instructions.

Figure-3-(a) shows the isolated I/O interface of a minimum – mode and fig-3-(b) show the maximum – mode 8086 – based microcomputer system.

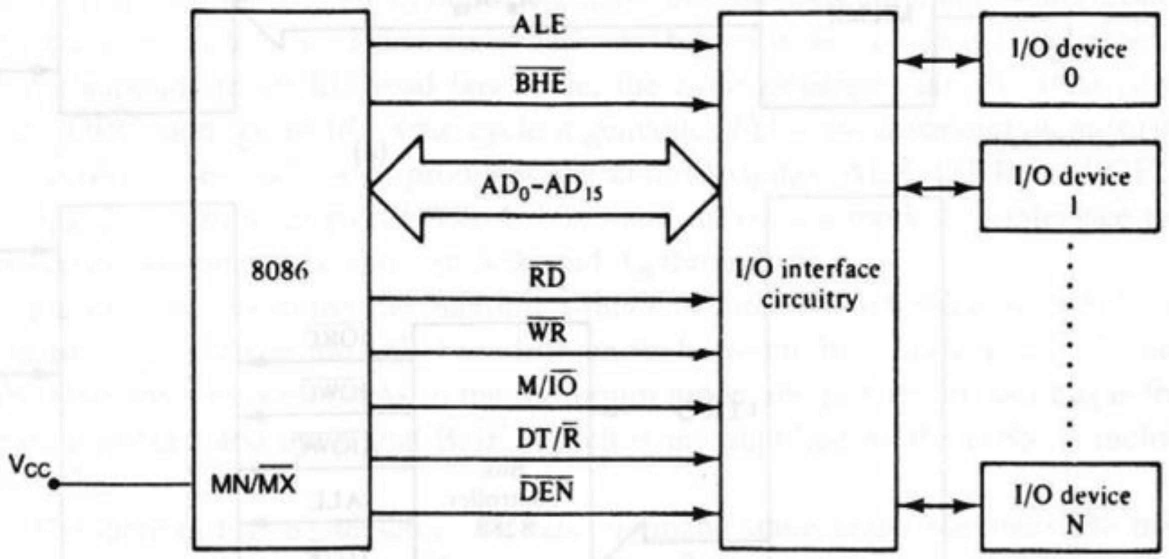


Fig-3-(a)

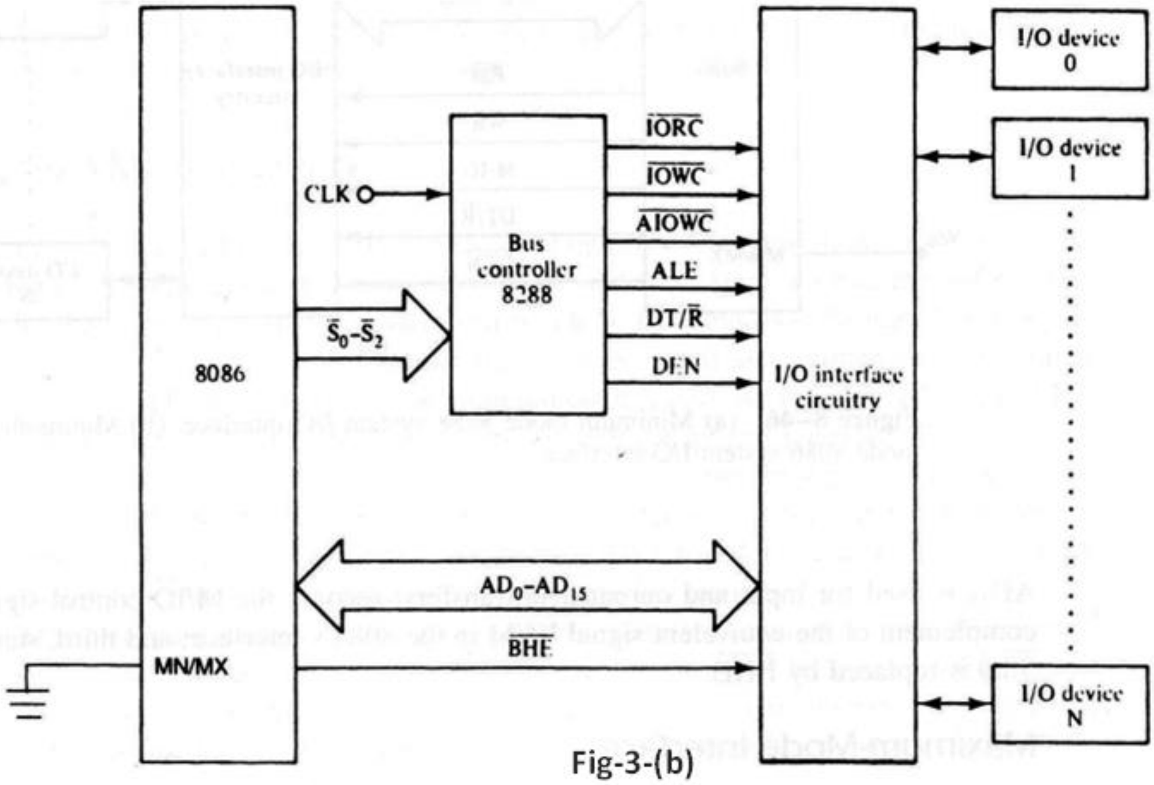
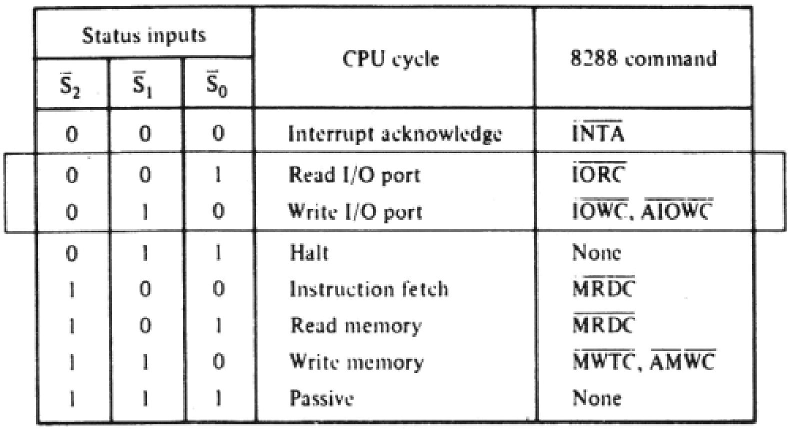


Fig-3-(b)



Status inputs			CPU cycle	8288 command
$\bar{S}_2$	$\bar{S}_1$	$\bar{S}_0$		
0	0	0	Interrupt acknowledge	$\overline{INTA}$
0	0	1	Read I/O port	$\overline{IORC}$
0	1	0	Write I/O port	$\overline{IOWC}$, $\overline{AIOWC}$
0	1	1	Halt	None
1	0	0	Instruction fetch	$\overline{MRDC}$
1	0	1	Read memory	$\overline{MRDC}$
1	1	0	Write memory	$\overline{MWTC}$, $\overline{AMWC}$
1	1	1	Passive	None

The table above shows the bus command status codes together with the command signals that they produce. Those for I/O bus cycles are highlighted. The MPU indicates that data are to be input (read I/O port) by code $\overline{S}_2\overline{S}_1\overline{S}_0$ = 001. This code causes the bus controller to produce control output I/O read command ($\overline{IORC}$). There is one other code that represents an output bus cycle, the write I/O port code $\overline{S}_2\overline{S}_1\overline{S}_0$ = 010. It produces two output command signals: I/O write cycle ($\overline{IOWC}$) and advanced I/O write cycle ($\overline{AIOWC}$). These command signals are used to enable data from the I/O ports onto the system bus during an input operation and from the MPU to the I/O ports during an output operation.

INPUT/OUTPUT DATA TRANSFERS

Input/output data transfers in the 8086 microcomputers can be either byte-wide or word-wide. The port that is accessed for input or output of data is selected by an I/O address. This address is specified as part of the instruction that performs the I/O operation. I/O addresses are 16 bits in length and are output by the 8086 to the I/O interface over bus lines ADO through AD15. The most significant address lines, A16 through A19, are held at the 0 logic level during the address period (T1) of all I/O bus cycles. Since 16 address lines are used to address I/O ports, the 8086's I/O address space consists of 64K byte-wide I/O ports.

The 8086 signals to external circuitry that the address on the bus is for an I/O port instead of a memory location by switching the $M/\overline{IO}$ control line to the 0 logic level. This signal is held at the 0 level during the complete input or output bus cycle.

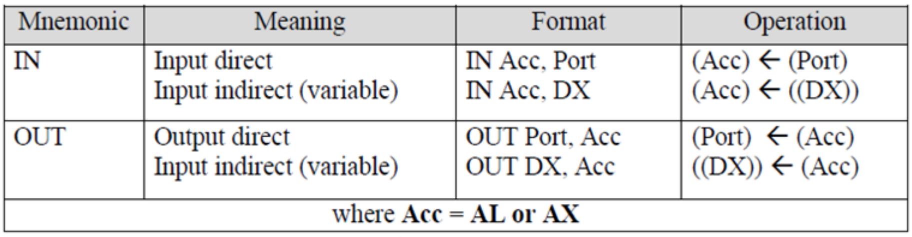
The logic levels of signals A_0 and $\overline{BHL}$ determine whether data are input/output for an odd-addressed byte-wide port, even-addressed byte-wide port, or a Word-wide port. For example, if $A_0 \overline{BHL}$ = 10, an odd-addressed byte-wide I/O port is accessed. Byte data transfers to a port at an even address are performed over bus lines D0 through D7 and those to an odd-addressed port are performed over D8 through D15. Data transfers to byte-wide I/O ports always take place in one bus cycle.

Word data transfers between the 8086 and I/O devices are accompanied by the code $A_0 \overline{BHL}$ = 00 and are performed over the complete data bus, D0 through D15. A word transfer can require either one or two bus cycles. To ensure that just one bus cycle is required for the word data transfer, word-wide I/O ports should be aligned at even-address boundaries.

INPUT/OUTPUT INSTRUCTIONS

The 8086 microprocessors that employ isolated I/O, using special input and output instructions together with the I/O port addressing modes.

These instructions, in (**IN**) and out (**OUT**), are listed in Fig.-4-. Their mnemonics and formats are provided together with a brief description of their operations.



Mnemonic	Meaning	Format	Operation
IN	Input direct Input indirect (variable)	IN Acc, Port IN Acc, DX	(Acc) ← (Port) (Acc) ← ((DX))
OUT	Output direct Input indirect (variable)	OUT Port, Acc OUT DX, Acc	(Port) ← (Acc) ((DX)) ← (Acc)
where **Acc = AL or AX**			

Fig-4-

Note that there are two different forms of IN and OUT instructions:

1- **Direct I/O instructions:**

- The address of the I/O port is specified as part of the instruction.
- Eight bits are provided for this direct address. For this reason, its value is limited to the address range form 00H to FFH
- This range is referred to as **Page 0** in the I/O address space.

2- **Variable I/O instructions**:

- The instructions use a 16-bit address that resides in the DX register within the MPU.
- The value in DX is not an offset.

All data transfers take place between an I/O device and the MPU's accumulator register. For this reason, this method of performing I/O is known as accumulator I/O. Byte transfers involve the AL register, and word transfers the AX register.

Ex: **IN AL, FEH**

Execution of this instruction causes the contents of the byte-wide I/O port at address FEH of the I/O address space to be input to the AL register. This data transfer takes place in one input bus cycle.

Ex: Write a sequence of instructions that will output the data FFH to a byte-wide output port at address ABH of the I/O address space.

Solution First, the AL register is loaded with FFH as an immediate operand in the instruction

```
MOV AL, FFH
```

Now the data in AL can be output to the byte-wide output port with the instruction

```
OUT ABH, AL
```

Ex: inputs the contents of the byte-wide input port at A000H of the I/O address space into AL and then saves it in BL.

```
MOV DX, A000H
IN AL, DX
MOV BL, AL
```

Ex: Write a series of instructions that will output FFH to an output port located at address B000H of the I/O address space.

Solution

The DX register must first be loaded with the address of the output port. This is done with the instruction

```
MOV DX, B000H
```

Next, the data that are to be output must be loaded into AL with the instruction

```
MOV AL, FFH
```

Finally, the data are output with the instruction

```
OUT DX, AL
```

Ex: Data are to be read in from two byte-wide input ports at addresses AAH and A9Hand then output as a word to a word-wide output port at address B000H. Write a sequence of instructions to perform this input/output operation.

Solution

We can first read in the byte from the port at address AAH into AL and move it to AH. This is done with the instructions

```
IN AL, AAH
MOV AH, AL
```

Now the other byte, which is at port A9H, can be read into AL by the instruction

```
IN AL, A9H
```

The word is now held in AX. To write out the word of data, we load DX with the address B000H and use a variable output instruction. This leads to the following:

```
MOV DX, B000H
OUT DX, AX
```

INPUT/OUTPUT BUS CYCLES

The waveforms of the 8086's input and output bus cycles are shown in Fig.-5-(a) and fig-5-(b), respectively. Looking at the input and output bus cycle waveforms, we see The 8086 switches $M/\overline{IO}$ it to logic 0 to indicate that an I/O bus cycle is in progress. It is maintained at the 0 logic level for the duration of the I/O bus cycle.

the 8086 outputs the signal $\overline{BHL}$ along with the address in T-state T1. As in memory cycles, the address is output together with ALE during clock period T1.

For the input bus cycle, $\overline{DEN}$ is switched to logic 0 to signal the I/O interface circuitry when to put the data onto the bus and the 8086 reads data off the bus during period T3.

On the other hand, for the output bus cycle in Fig.-5-(b), the 8086 puts write data on the bus late in T2 and maintains it during the rest of the bus cycle. This time $\overline{WR}$ switches to logic 0 to signal the I/O system that valid data are on the bus.

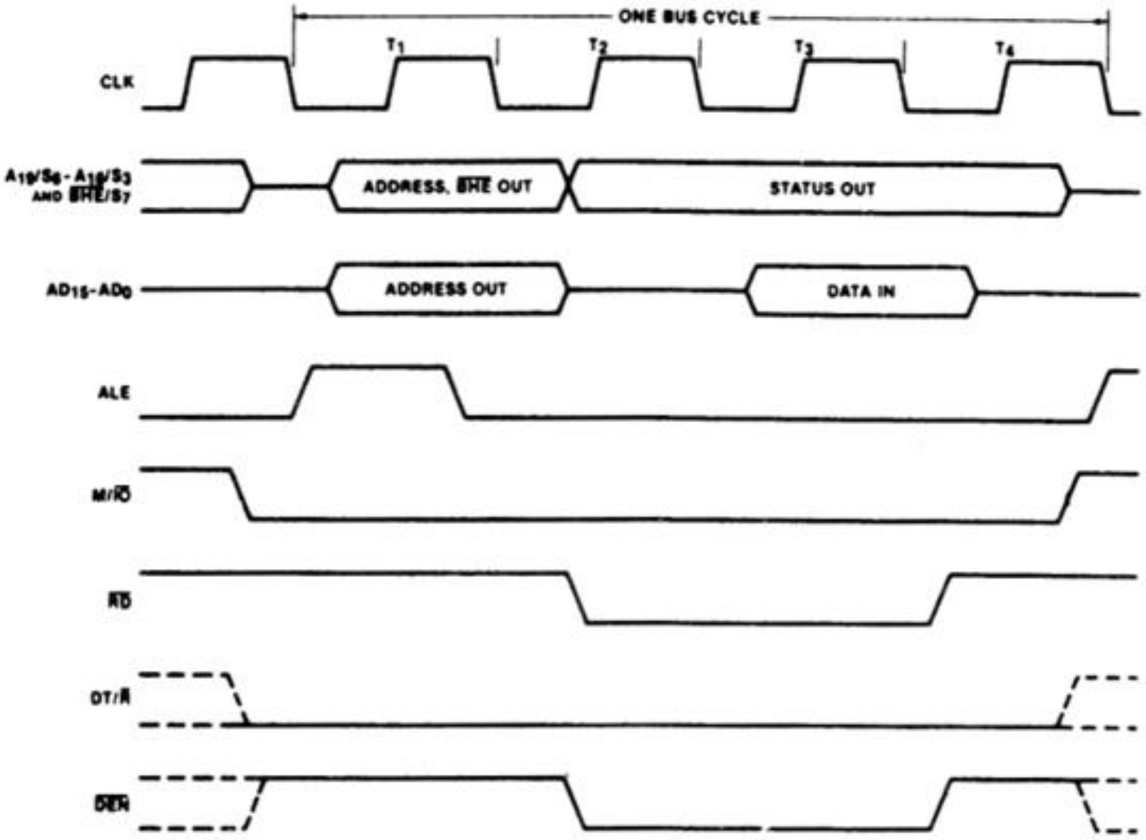

ONE BUS CYCLE
T1
T2
T3
T4
CLK
A19/S6-A16/S3 AND BHE/S7
ADDRESS, BHE OUT
STATUS OUT
AD15-AD0
ADDRESS OUT
DATA IN
ALE
M/IO
RD
DT/R
DEN

Fig-5-(a)

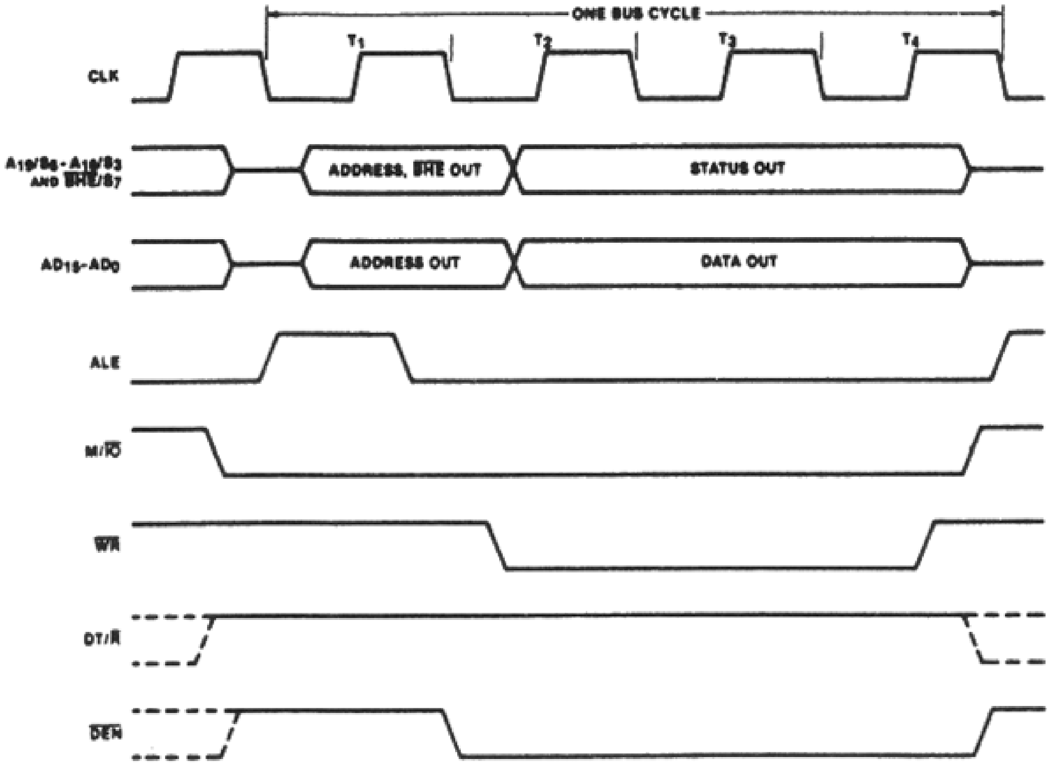

ONE BUS CYCLE
T1
T2
T3
T4
CLK
A19/S6-A16/S3 AND BHE/S7
ADDRESS, BHE OUT
STATUS OUT
AD15-AD0
ADDRESS OUT
DATA OUT
ALE
M/IO
WR
DT/R
DEN

Fig-5-(b)

MEMORY-MAPPED INPUT/ OUTPUT INTERFACE:

I/O devices can be placed in the memory address space of the microcomputer as will as in the independent I/O address space. In this case, the MPU looks at the I/O port as though it is a storage location in memory. For example, in Figure-6- the 4096 memory addresses in the range form E0000H through E0FFFH are assigned to I/O devices.

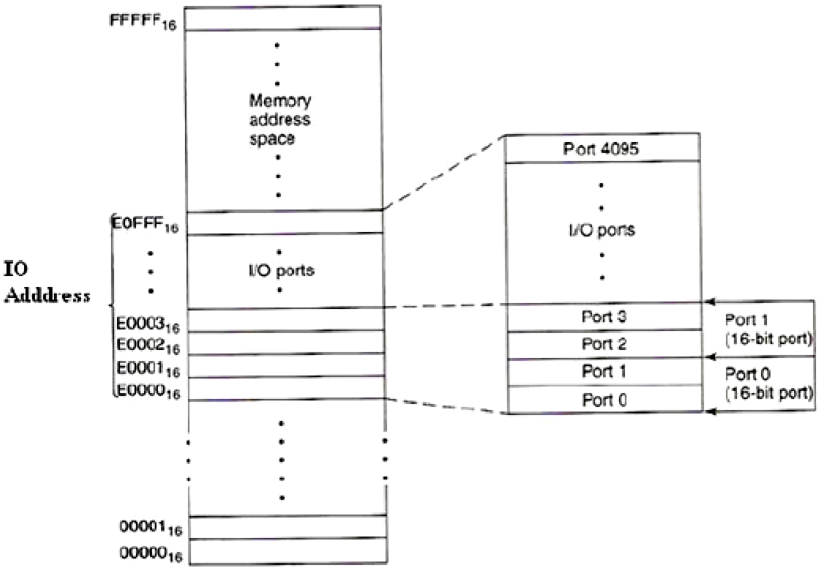


Fig-6-

The memory – mapped I/O interface is similar to memory interface that we describes before. The isolated I/ O and memory-mapped I/ O differ in terms of: -

1- The number of address lines used in identifying an I/O device.
2- The type of control line used to enable the device.
3- The instruction used for data transfer.