Input Devices Interfacing, Output Devices Interfacing

Second Stage

Microprocessor

Computer Engineering Department

Lecture No. 5

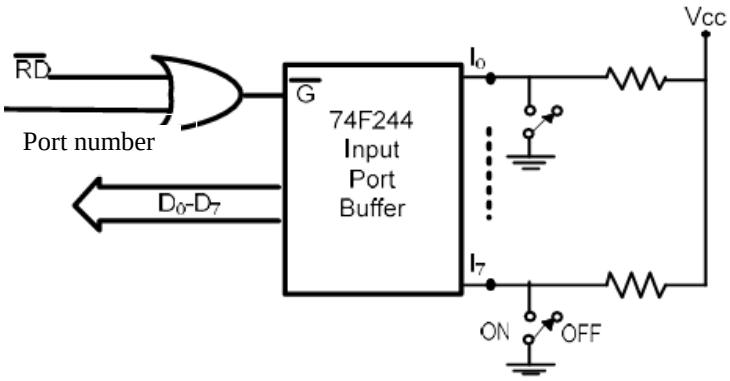
Prepared by: Asst. Lect. Eng.
Hamza Waleed Hamza

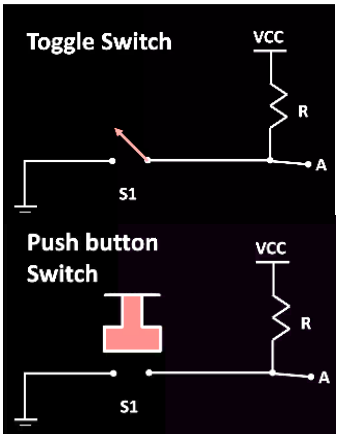
Interfacing I/O Device

Any application of a microprocessor based system requires the transfer of data between external circuitry to the microprocessor and microprocessor to the external circuitry. User can give information to the microprocessor based system using key board and user can see the result or output information from the microprocessor based system with the help of display device. The transfer of data between microprocessor and outside world is called input/output data transfer or I/O data transfer. This data transfer is done with the help of I/O ports.

Input port:

It is used to read data from the input device. The simplest form of input port is a tri state buffer as shown in the figure below. Its output is available only when enable signal is active. When microprocessor wants to read data from the input device, the control signals from the microprocessor activates the buffer by asserting enable input of the buffer.



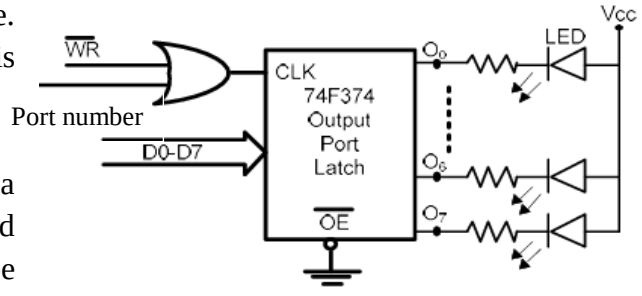


Input Port

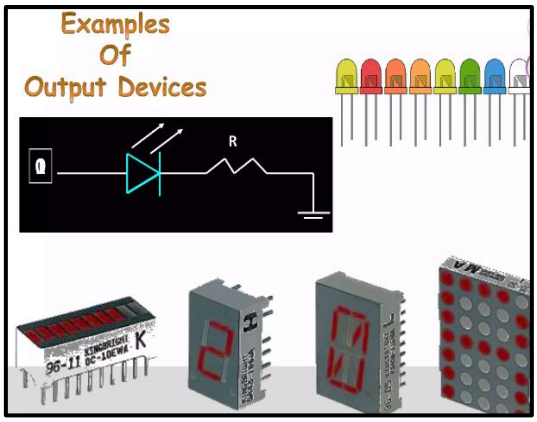
I/O address (also called Port number) appears on the address bus. The address lines are decoded to generate a signal that is active when the particular port is being accessed. The control (Enable) for these buffers is connected to the result of combining the address signal and the READ signal.

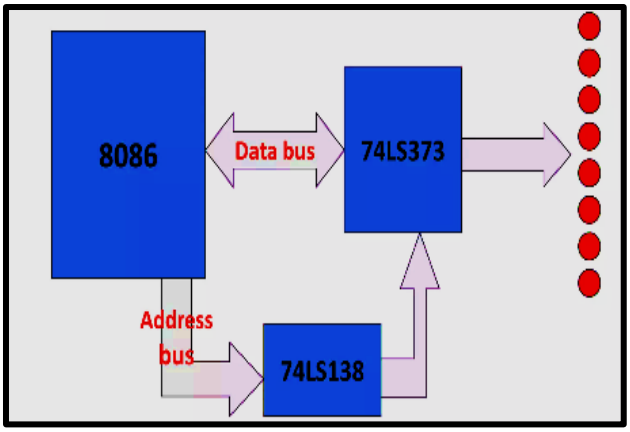
Output port:

Output ports are used to send data to the output device. Output devices are usually slow. Also, the output is usually expected to continue appearing on the output device for a long period of time. Given that the data will only be present on the data lines for a very short period (microseconds), it has to be latched externally. To do this the external latch should be enabled when the port's address is present on the address bus, and WR is set low. The resulting signal would be active when the output device is being accessed by the microprocessor as shown:



Output Port



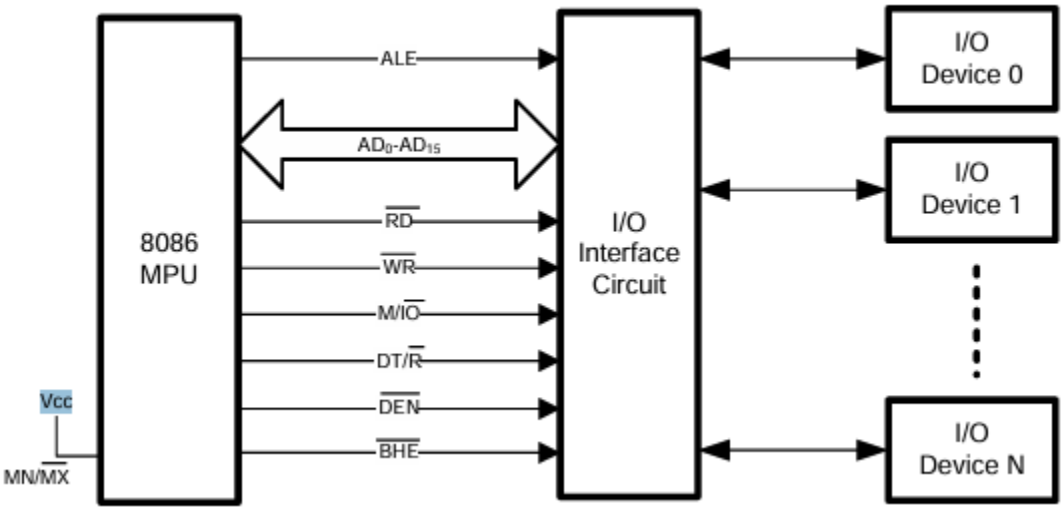


Interfacing Techniques and address Decoding

Minimum Mode Interface

This interface use the following control signals

- ALE = pulse to logic 1 tells bus interface circuitry to latch I/O address
- $\overline{RD}$ = logic 0 tells the I/O interface circuitry that an input (read) is in progress
- $\overline{WR}$= logic 0 tells the I/O interface circuitry that an output (write) is in progress
- $\overline{IO}$/ M = logic 0 tells I/O interface circuits that the data transfer operation is for the I/O subsystem
- DT/$\overline{R}$ = sets the direction of the data bus for input (read) or output (write) operation
- $\overline{DEN}$= enables the interface between the I/O subsystem and MPU data bus

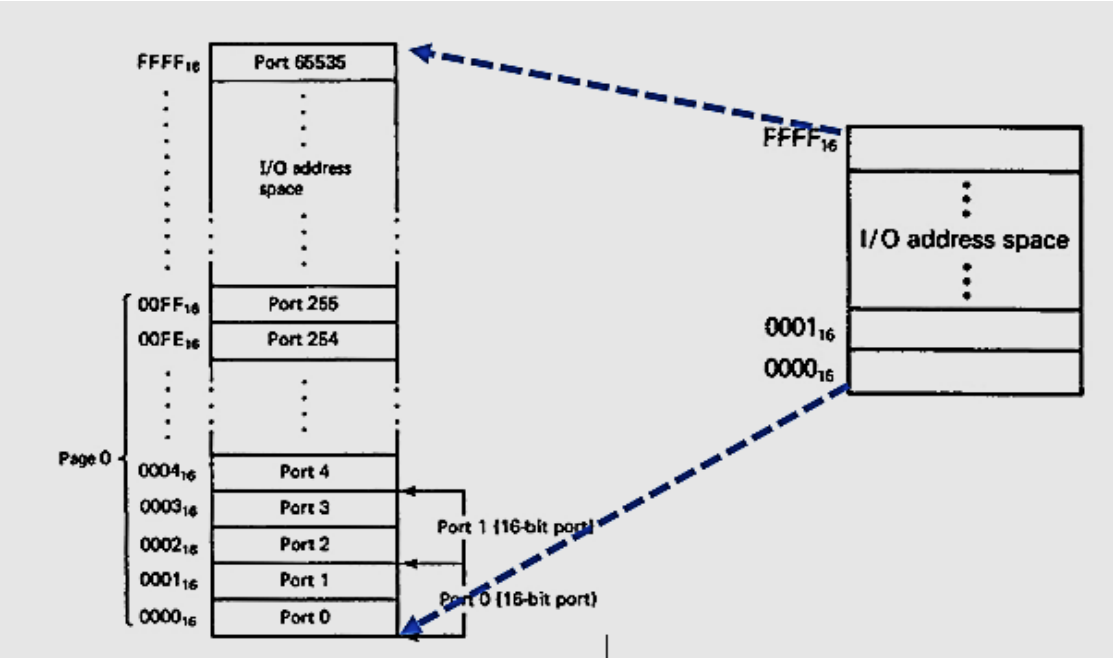


Minimum mode 8086 system I/O Interface.

8088/8086 architecture implements two interfacing techniques; Isolated I/O and Memory-mapped I/O.

Isolated I/O mapped: I/O devices are treated separate from memory.

- The part of the I/O address space from address 0000H through 00FFH is referred to as
- Page 0 as shown in figure below.
- Supports byte and word I/O ports
- 64K independent byte-wide I/O ports
- 32K independent aligned word-wide I/O ports



Isolated I/O port.

Advantages of Isolated I/O

1. Complete memory address space available for use by memory
2. Special instructions have been provided in the instruction set of the 8086 to perform isolated I/O operation. This instructions tailored to maximize performance.

Disadvantage of Isolated I/O

All inputs/outputs must take place between an I/O port and accumulator (AL or AX) Register

Isolated Interfacing basic instructions

There are different forms of I/O instructions: the direct I/O instructions as below:

1- **IN and OUT** transfer data between an I/O device and the microprocessor's accumulator (AL, AX), it can transfer byte or word.

Mnemonic	Meaning
IN AL, port no.	A byte is input into AL from port no.
IN AX, port no.	A word is input into AX from port no.
IN AL, DX	A byte is input into AL from the port addressed by DX
IN AX, DX	A word is input into AX to the port addressed by DX
OUT AL, port no.	A byte is output from AL to port no.
OUT AX, port no.	A word is output from AX to port no.
OUT AL, DX	A byte is output from AL to the port addressed by DX
OUT AX, DX	A word is output from AX to the port addressed by DX

Example:

```
IN AL,0C8H          ;Input a byte from port 0C8H to AL
IN AX, 34H          ;Input a word (two byte) from port 34H, 35H to AX
OUT 3BH, AL         ;Copy the contents of the AL to port 3Bh
OUT 2CH,AX          ;Copy the contents of the AX to port 2CH, 2DH
MOV DX, 0FF78H      ;Initialize DX point to port
IN AL, DX           ;Input a byte from a 8 bit port 0FF78H to AL
IN AX, DX           ;Input a word from 16 bit port to 0FF78H,0FF79H to AX.
```

Example: write a program that will output FFH to an output port addressed by B000H of the I/O address space.

Solution:

```
MOV DX, B000H
MOV AL, FF
OUT DX, AL
```

Example: Data are to be read from two byte-wide input ports at addresses AAH and A9H and then output as a word to a word-wide output port at address B000H. Write a program to perform this input/output operation.

Solution:

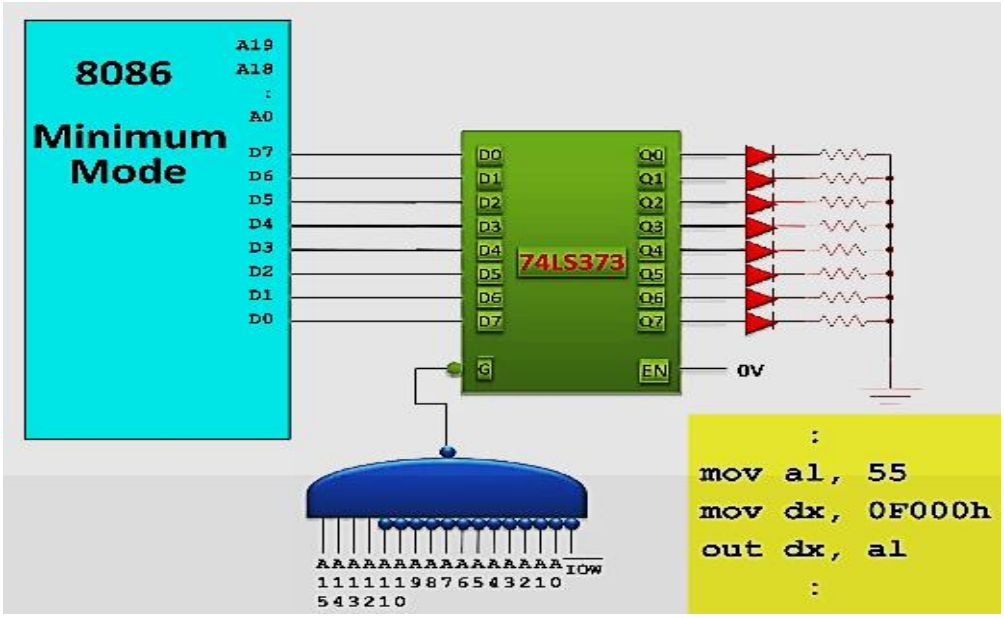
```
IN  AL, AAH
MOV AH, AL
IN  AL, A9H
MOV DX, B000H
OUT DX, AX
```

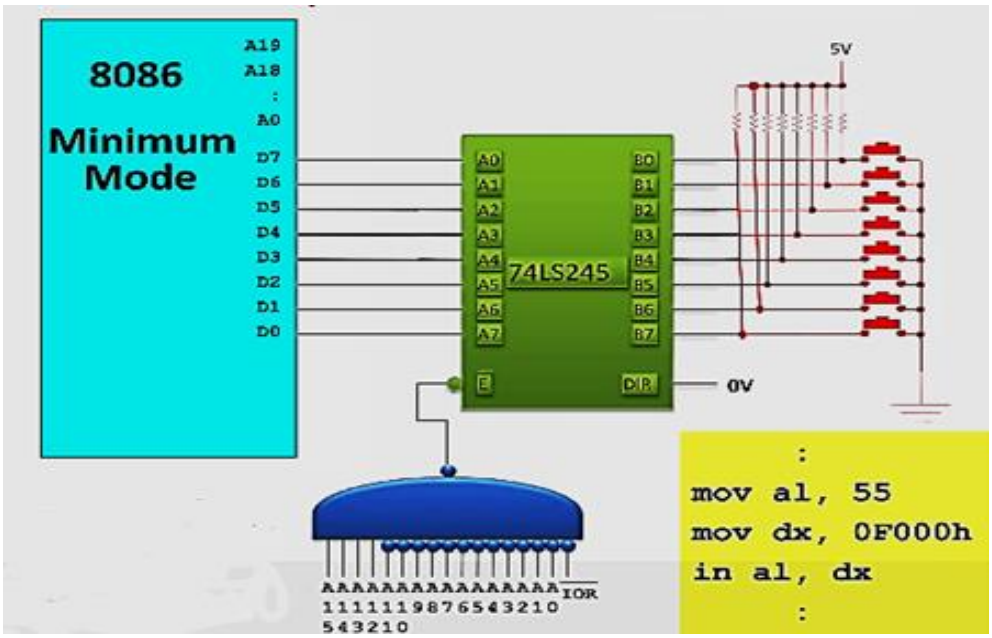
EX: For the following circuits;

a- Write a program to output the content of AL to the output port , what will be the status of the LEDs.

b- Write a program to transfer the status of the switches to AL, what will be the content of AL.

C- Repeat b if all switches are closed. d- Repeat a if the content of AL =AAH.





2-**INS and OUTS**: these instructions transfer data between memory and the I/O device. The port address is stored in DX and the memory address is located by ES:DI for the INS instruction and by DS:SI for the OUTS instruction.

INS / INSB / INSW — Input String Instructions

- DX holds the port address
- DI (or ES:DI) points to the destination in memory
- After transfer: DI increments or decrements depending on the Direction Flag (DF)

Example: Transfer a byte from port 60H into memory location 2000H

```
MOV DX, 60H      ; port address
MOV DI, 2000H    ; memory location
INSB
```

OUTS / OUTSB / OUTSW — Output String Instructions

- DX = port address
- SI (or DS:SI) = data source
- After transfer: SI increments or decrements based on DF

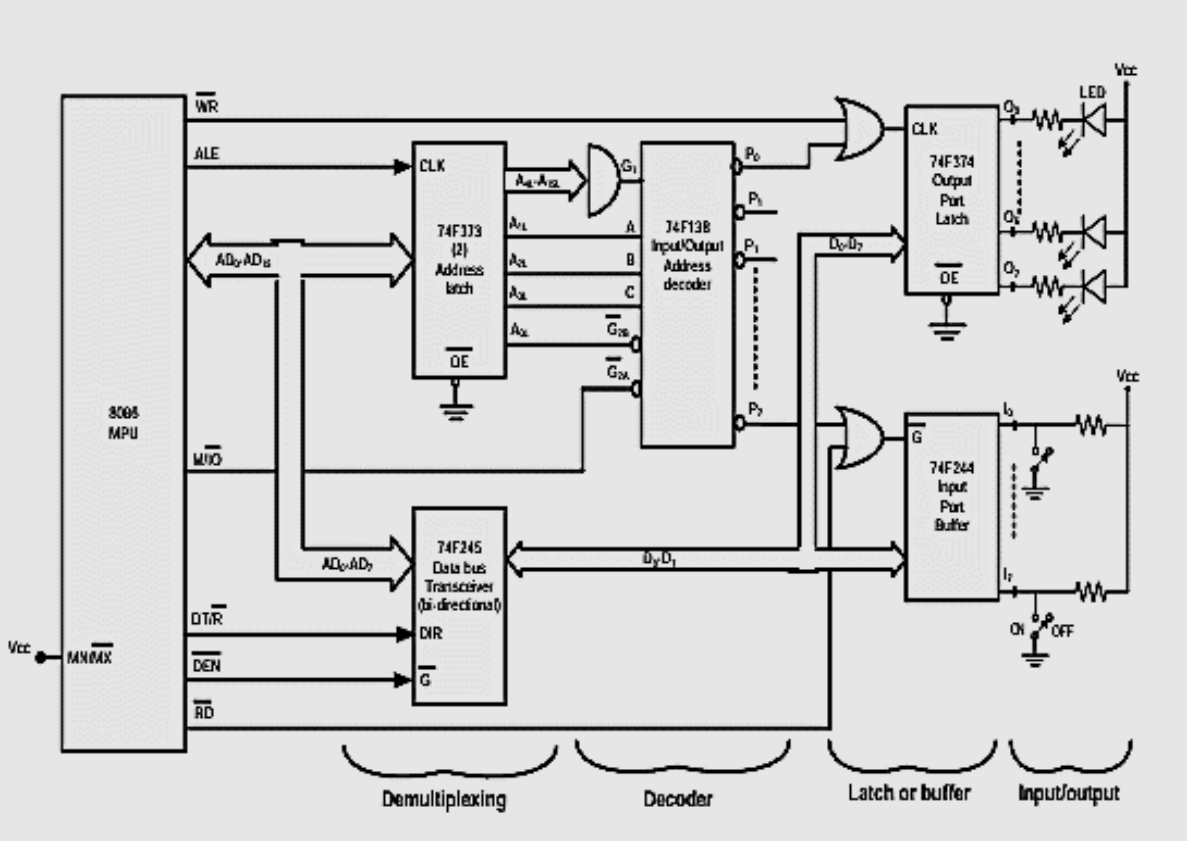
Example: Output 1 word from memory to port 378H.

```
MOV DX, 378H     ; printer port
MOV SI, 3000H
OUTSW
```

Example: Byte-Wide Input and Output Ports using Isolated I/O

Figure below shows a circuits diagram of a byte-wide input and output ports (8 bit) using isolated I/O for an 8086 based microcomputer system.

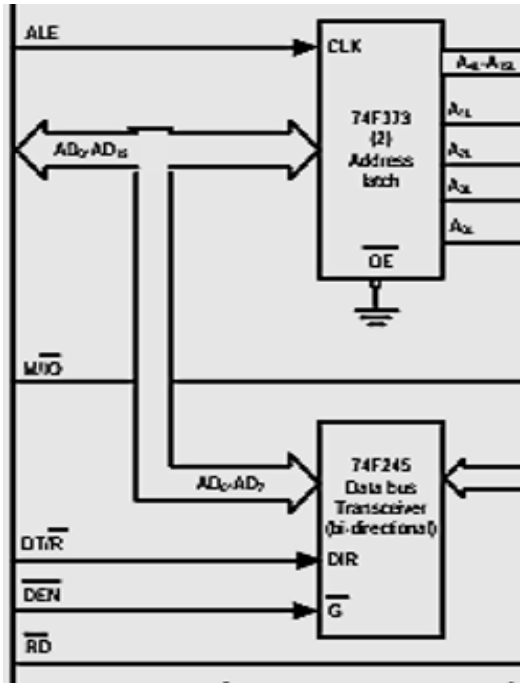
A) Analyze the circuit.



Demultiplexing circuit:

 Two 74F373 octal latches are used to form a 16-bit address latch. These devices latch the address A0 through A15 synchronously with the ALE pulse.

NOTE: address lines A16 through A19 are not involved in the I/O interface.

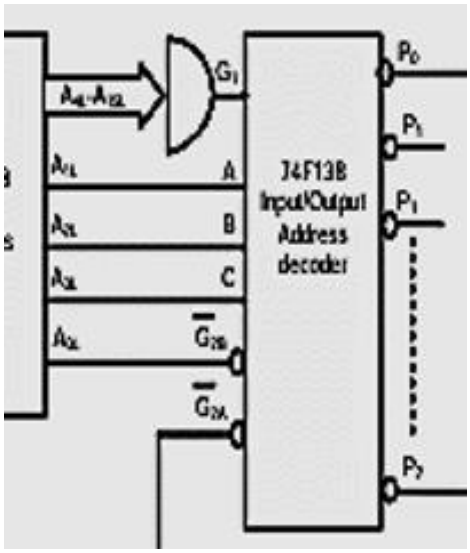


 Data bus transceiver buffer in 8086 system is implemented using 74F245 octal bus IC's, where the control inputs 'DIR ' and ' G' is used to control the data flow .

 G input is used to enable the buffer operation, whereas DIR input selects the direction of intended data transfer.

Decoder circuit:

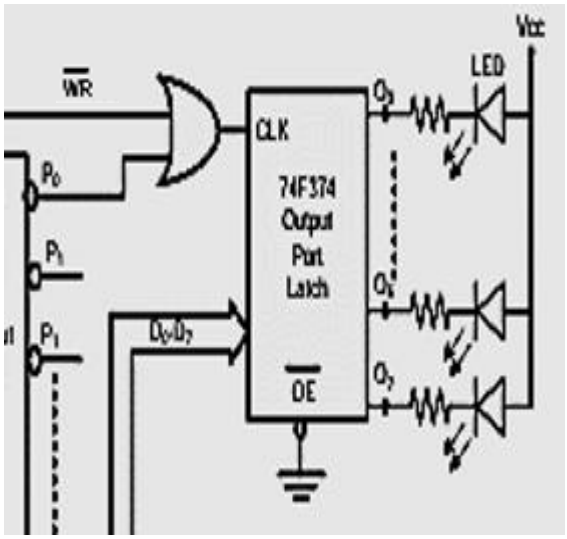
 A 74F138 (3 Line-to- 8 Line decoder) is used for decoder circuit. Address lines $A4_L$-$A15_L$ are applied to AND gate, output of AND gate and address line $A0_L$ provide two of the three enable inputs of the 74F138 input/output address decoder. These signals are applied to enable input G_1 and $\overline{G_{2B}}$ respectively.



- The decoder requires one more enable signal at its $\overline{G_{2A}}$ input, which is supplied by the complement of M/$\overline{IO}$.
- The enable inputs must be $\overline{G_{2A}}$ $\overline{G_{2B}}$ G_1=001 to enable the decoder for operation.
- The condition $G_{\overline{2B}}$=0 corresponds to an even address, and $\overline{G_{2A}}$=0 represent the fact that an I/O bus cycle is in progress. G_1 =1 is achieved if the output of AND gate is logic 1
- The three address lines A_{3L} A_{2L} A_{1L} are applied to select inputs CBA of the 74F138 decoder.
- When the decoder is enabled, the P output corresponding to these select inputs switches combination.

The output port

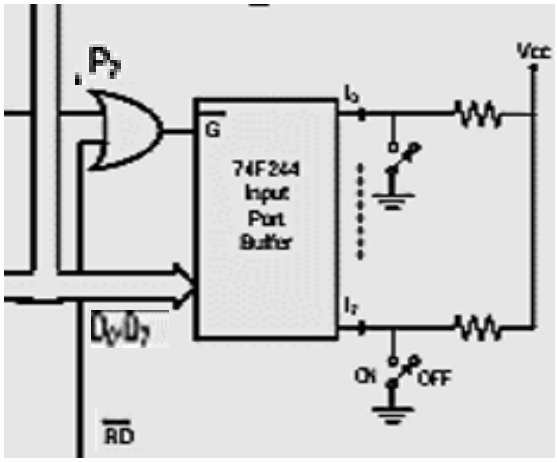
- For output circuit we need to store the output data so we use latch, for this purpose
  - 74F374 device is selected. The gate at the CLK input of 74F374 has its inputs P_0 and $\overline{WR}$.
  - When valid output data are on the bus, $\overline{WR}$ switches to logic 0. Since P_0 is also 0, the CLK input of the 74F374 for port 0 switches to logic 0.
  - At the end of the $\overline{WR}$ pulse, the clock switches from 0 to 1, a positive transition. This causes the data on D_0 –D_7 to be latched and become available at output lines O_0 –O_7 of port 0.
  - $\overline{OE}$ =0 to enable the output , the latched data appears at the appropriate port outputs.



The input port

Buffer is used with input ports. The 74F244 octal buffer is used to implement the port. The outputs of the buffer are applied to the data bus for transfering switches status to the MPU. This buffer has three-state outputs.

- The gate at the $\overline{G}$ input of 74F244 has its inputs P7 and $\overline{RD}$, when an input bus cycle is in progress, $\overline{RD}$ switches to logic 0.
- Since P7 is also 0, the G input of the 74F244 switches to logic 0 and the outputs of 74F244 are enabled.
- In this case, the logic levels at inputs I_0 – I_7 are passed onto data bus lines D_0 – D_7 respectively. This byte of data is carried through the enable data bus transceiver to the data bus of the 8086. As part of the input operation, the 8086 reads this byte of data into the AL register.



B) Determine the input/output Ports Addresses

(a) To enable Port 0 (P0) :

Input of decoder ABC = 000 → A_{1L} A_{2L} A_{3L} = 000

A_{15L}	A_{14L}	A_{13L}	A_{12L}	A_{11L}	A_{10L}	A_{9L}	A_{8L}	A_{7L}	A_{6L}	A_{5L}	A_{4L}	A_{3L}	A_{2L}	A_{1L}	A_{0L}
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

I/O address of Port 0 = FFF0H

(B) To enable Port 7 (P7) :

Input of decoder ABC = 111 $\rightarrow A_{1L}\,A_{2L}\,A_{3L}$ = 111

A_{15L}	A_{14L}	A_{13L}	A_{12L}	A_{11L}	A_{10L}	A_{9L}	A_{8L}	A_{7L}	A_{6L}	A_{5L}	A_{4L}	A_{3L}	A_{2L}	A_{1L}	A_{0L}
1	1	1	**1**	**1**	1	**1**	1	1	**1**	1	1	**1**	1	1	0

I/O address of Port 0 = FFFEH

C) Write a program that inputs the contents of input port 7 to the memory DS:A000H.

Solution: The address of Port P7 =FFFEH. The instruction sequence needed to input the byte is:

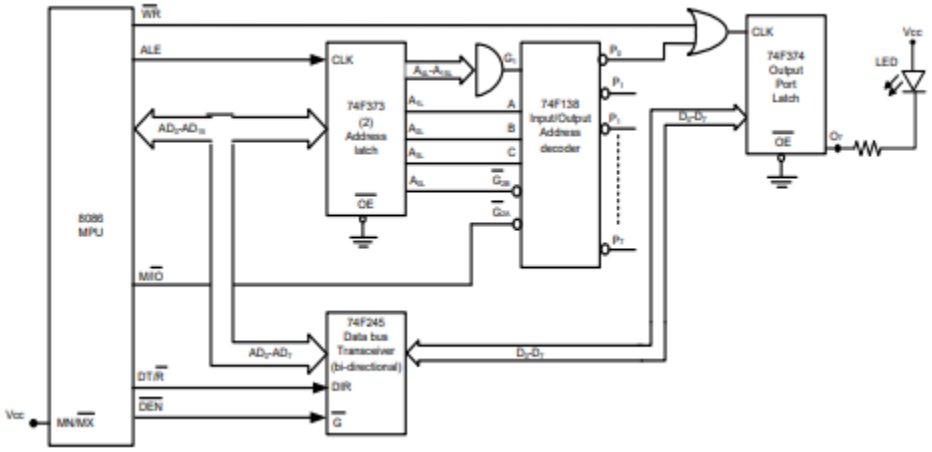
```
MOV DX, FFFEH
IN AL, DX
MOV [A000], AL
```

D) Write a program that outputs contents of memory location DS:8000H to output port 0.

Solution: The address of Port P0 =FFF0H. The instruction sequence needed to output the contents of memory location DS:8000H to output port 0 is:

```
MOV DX, FFF0H
MOV  AL , [8000]
OUT DX , AL
```

E) Write instructions sequence to make the LED blink in figure below:



Solution: we must write a program that first makes O7 logic 0 to turn on the LED, delays for a short period of time, and then switches O7 back to 1 to turn off the LED. This piece of program can run as a loop to make the LED continuously blink. This is done as follows:

Sequence of instructions needed to initialize0 7 to logic 0.

```
                MOV DX, FFF0H   ; Initialize address of port0
                MOV AL, 00H     ; Load data with bit 7 as logic 0
ON_OFF:  OUT DX, AL             ; Output the data to port 0
```

Delay for a short period of time so as to maintain the data written to the LED

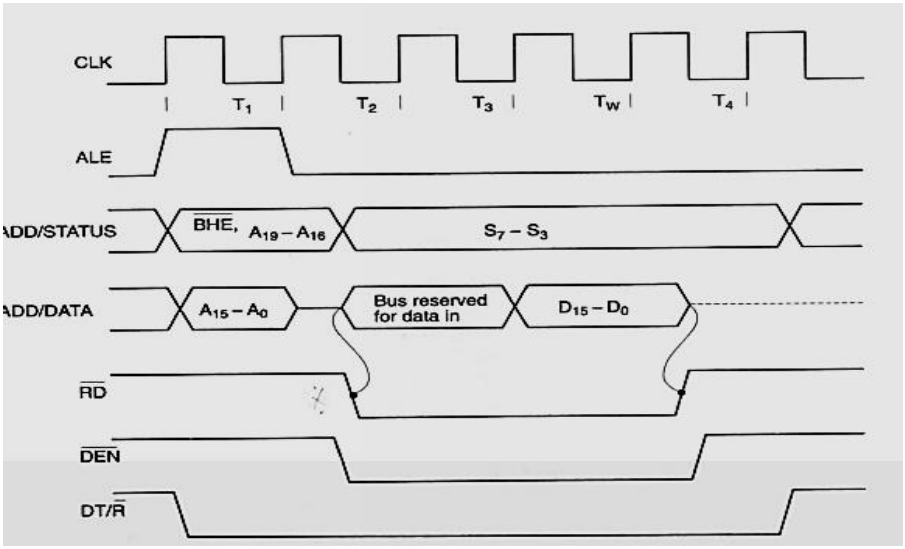
```
        MOV CX, FFFFH   ; Load delay count of FFFFH
HERE:   LOOP HERE       ; Time delay loop
```

The value in bit 7 of AL is complemented to 1 and then a jump is performed to return to the output operation that writes the data to the output port:

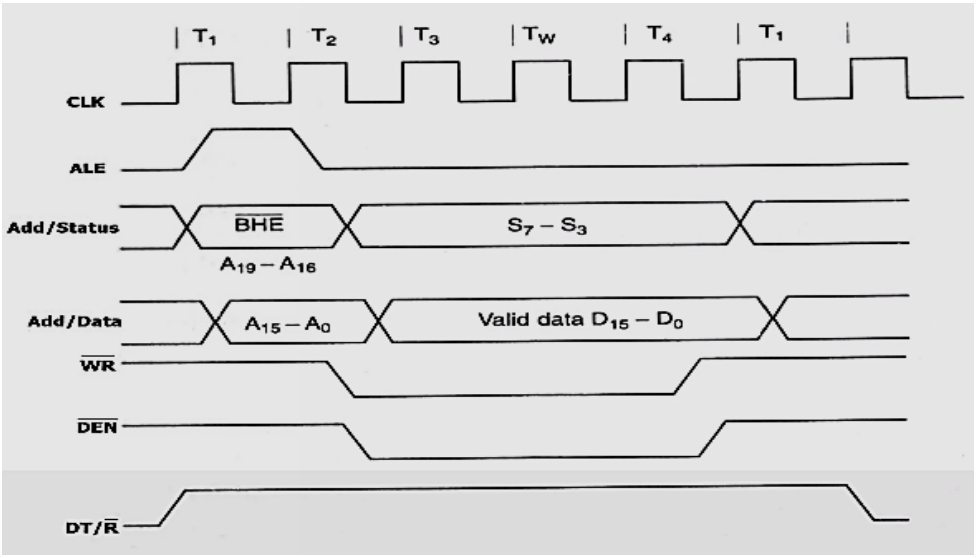
```
XOR  AL, 80H    ; Complement bit 7 of AL
JMP ON_OFF      ; Repeat to Output the new bit 7
```

Input/Output Bus Cycles

Input Read Cycle

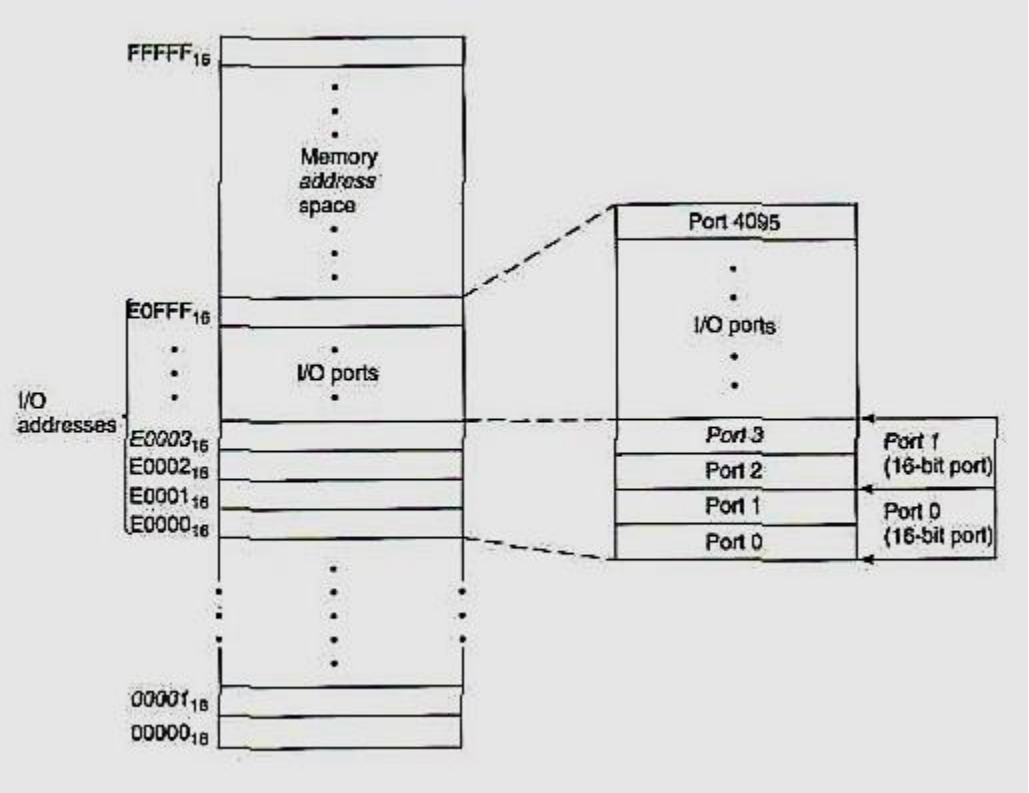


Output Write Cycle



Memory-mapped input/output : a part of the memory address space is dedicated to I/O devices

- For Example: (E0000H-E0FFFH) → 4096 memory addresses assigned to I/O ports
- E0000H, E0001H, and E0002H correspond to byte wide ports 0,1, and 2
- E0000H and E0001H correspond to word-wide port 0 at address E0000H
- For memory-mapped I/O, decoding is identical to memory decoding.
- For memory-mapped I/O, read and write cycles are identical to memory decoding.



memory mapped I/O

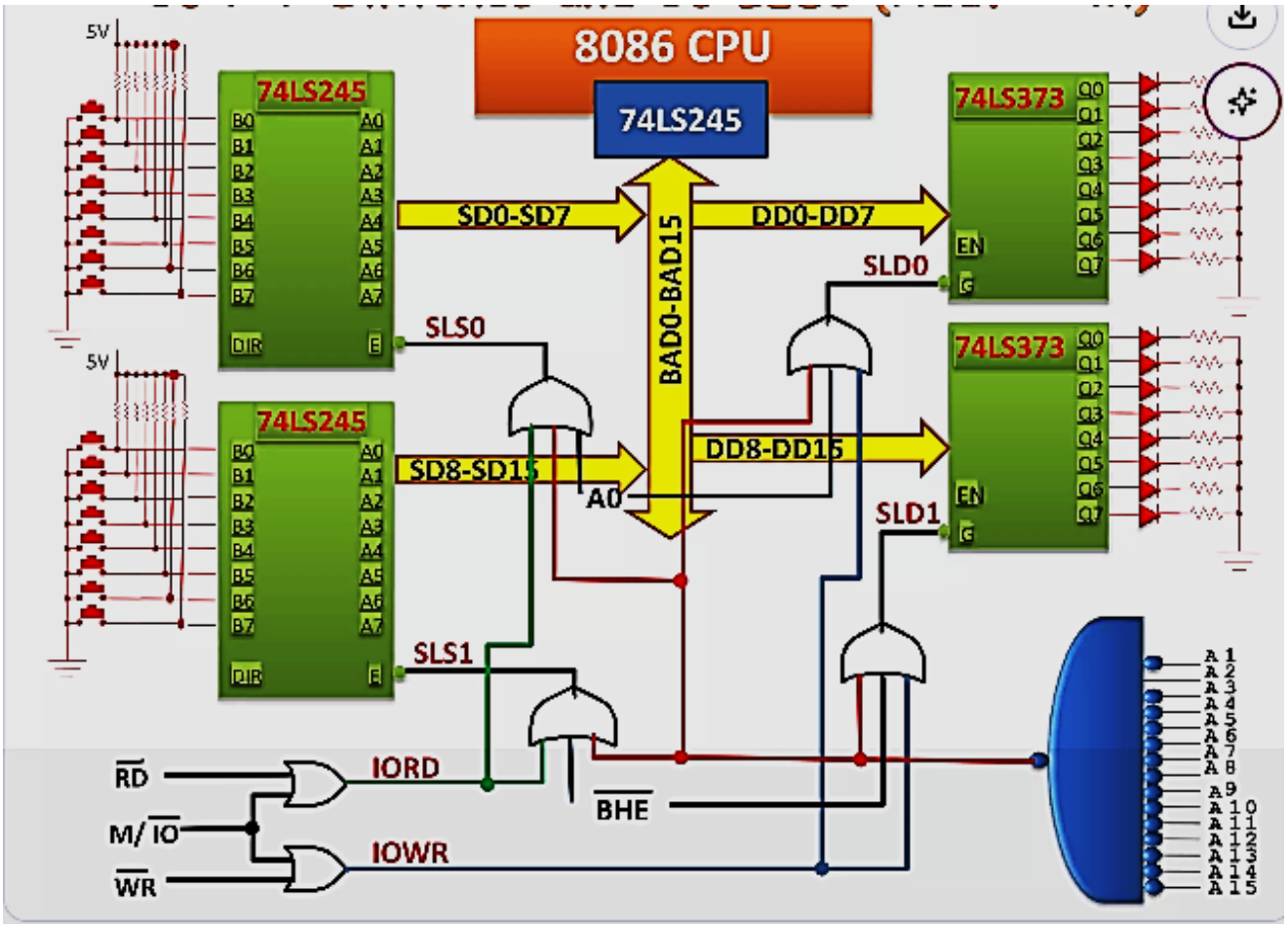
Advantages of memory mapped I/O

1. (MOV, ADD, AND, etc.) can be used to perform I/O operations.
2. I/O transfers can take place between I/O port and any of the registers.

Disadvantage of memory mapped I/O

1. Memory instructions perform slower
2. Part of the memory address space cannot be used to implement memory

EX: Interfacing 16 bit I/O port using memory mapped as in figure below:



M/IO	RD	WR	BHE	A0	A15 –A3	A2	A1	SLD1	SLD0	SLS1	S
1	X	X	X	X	X	X	X	1	1	1	1
0	0	1	0	0	0	1	0	1	1	0	0
0	0	1	1	0	0	1	0	1	1	1	0
0	0	1	0	1	0	1	0	1	1	0	1
0	1	0	0	0	0	1	0	0	0	1	1
0	1	0	1	0	0	1	0	1	0	1	1
0	1	0	0	1	0	1	0	0	1	1	1
X	X	X	X	X	X	0	X	1	1	1	1

BHE#	A0/BLE	Selection
0	0	Whole word (16-bits)
0	1	High byte to/from odd address
1	0	Low byte to/from even address
1	1	No selection

Differences between Isolated I/O and Memory Mapped I/O:

Isolated I/O	Memory Mapped I/O
Isolated I/O used separated memory space	Memory mapped I/O uses memory from the main memory
Limited instructions can be used such IN, OUT, INS, OUTS	Any instructions which references to memory can be used. (MOV, AND, XCHG, SUB, ….)
Faster because I/O instruction is specifically designed to run faster than memory instruction	Slower because memory instruction execute slower than the special I/O instructions
The memory address space is not affected	Part of the memory address space is lost
The addresses for isolated I/O devices are called ports	Memory mapped I/O devices are treated as memory locations on the memory map.