Input and Output

- Input & Output (I/O) devices provide the means by which a computer system can interact with the outside worlds.

- An I/O device can be a purely input device (e.g. KB, Mouse), a purely output device (printer, screen), or both input and output device like (e.g. disk).

- Regardless of the intended purpose of I/O devices, all communication with these devices must involve the system bus. However, I/O devices are not directly connected to the system bus. Instead, there is usually, On I/O controller that acts as an interface between the system and the I/O devices.

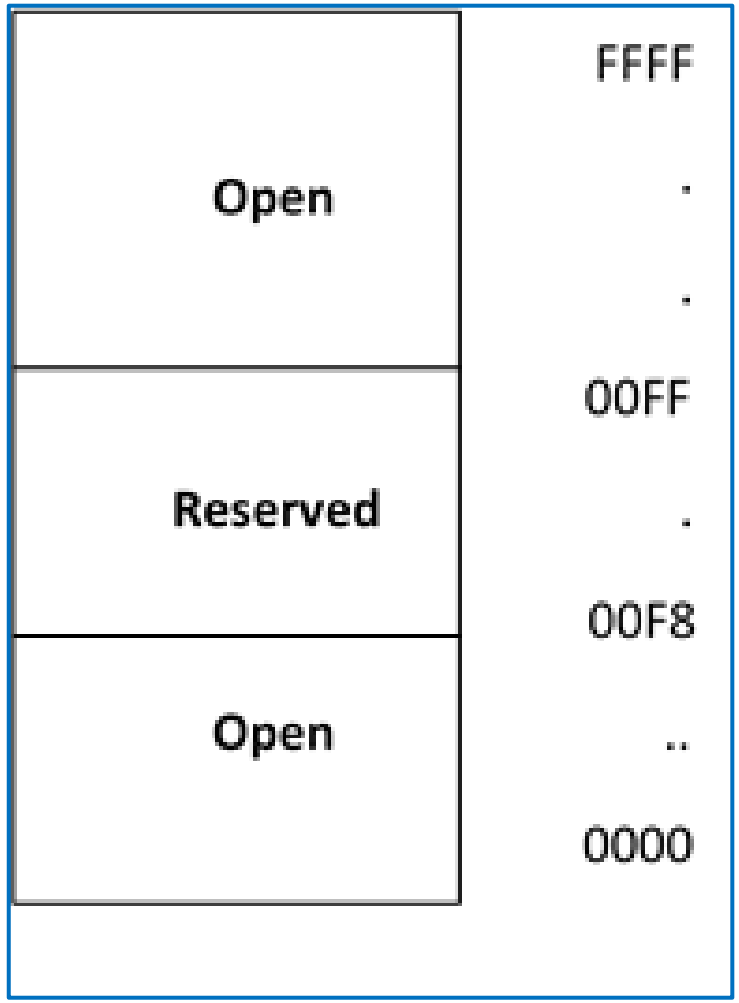
Accessing I/O Devices

- As programmer, you can have direct control to any of the I/O devices (through their associated I/O controller).
- It is a waste of time and effort if every one had to develop their own routines to access I/O devices. In addition system resource could be abused either intentionally or accidentally. For instance, and improper disk drive could erase the content of a disk due to a bug in the driver routine.
- To avoid this problem and to provide a standard way of accessing I/O devices, OS provide routine to convent all access I/O devices. Typically, access to I/O devices can be obtain from two layer of system software, the basic I/O system (BIOS) and the OS,BIOS is ROM resident and is a collection of routine that control the I/O devices. Both provide access to routine that control I/O devices through a mechanism called INT (interrupt).

I/O Address Space and Data Transfer

- As we know I/O ports in the 8086 MPU 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. The address is specified as port of the instruction that performs the I/O operation.
- I/O addresses are 16 bit in length and are output by the 8086 to the I/O interface over bus lines AD0 through AD15, the most significant bit A16-A19 of the memory address are held at the 0 logic (not used).
- Below Figure 19 show a map of I/O address space of the 8086 system. This is an independent 64-KB address space that is dedicated for I/O devices. Notice that its address range is from 000016-FFFF16. Moreover, notice that the eight ports located from address 00F8 to 00FF are specified as reserved. These port addresses are reserved by Intel for use in their future HW and SW products.

I/O Address Space



I/O Address Space and Data Transfer

- Data transfer between the MPU and I/O devices are performed over the data bus. Word transfer take place over the complete data bus D0 to D15, and can required either one or two bus cycle.
- Ports: a port is a device that connects the processor to the external world through a port processor, receive a signal from an input device and send a signal to an output device.

Input / Output Instruction

- The instruction set contains one type of instruction that transfer information to an I/O device (OUT) and another to read information from an I/O device (IN).

Instruction	Meaning	Format	Operation
IN	Input direct	IN ACC, PORT	ACC ← PORT
	Input indirect	IN ACC, DX	ACC ← (DX)
OUT	Output direct	OUT PORT,ACC	PORT ← ACC
	Output indirect	OUT DX, ACC	(DX) ← ACC

ACC = AL or AX

Input / Output Instruction

- Ex 1: write a sequence of inst that will output FF16 to a byte wide output port at address AB16 of the I/O addresses space.
- Solution: first the AL register is loaded with FF16 as an immediate operand in the instruction
- MOV AL, 0FFH
- Now the data in AL can be output to the byte wide output port with the instruction
- OUT 0ABH, AL

Input / Output Instruction

- Ex 2: write a series of instruction that will output FF16 to an output port located at address B00016 of the I/O address space

- Solution: the DX register must first be loaded with the address of the output port
- MOV DX, 0B000H
- Next, the data that is to be output must be loaded into AL
- MOV AL, 0FFH
- Finally, the data are output with the instruction
- OUT DX, AL

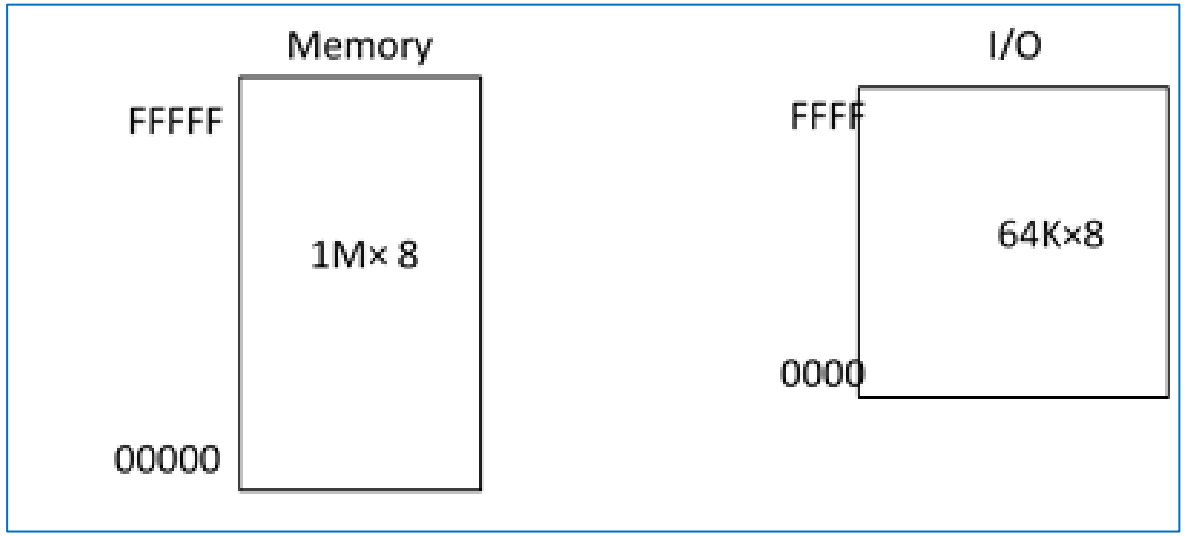
- Ex 3: data are to be read in from two byte wide input port at address AA16 and A916 respectively, and then output to a word wide output port at address B00016. Write a sequence of instruction to perform this I/O operation:
- Solution: we first read in a byte from the port at address AA16 into AL and move it to AH
- IN AL, 0AAH
- MOV AH, AL
- The other byte can be read into AL
- IN AL, 0A9H
- To writhe out the word of data in AX, we can load DX with the addressB00016 and use a variable output instruction
- MOV DX, 0B000H
- OUT DX, AX

Isolated and Memory I/O

- There are two different method of interfacing I/O to the MPU.
- In the isolated I/O scheme, the IN, OUT instruction transfer data between the MPU (ACC or memory) and the I/O device.

Isolated I/O

- It is the most common I/O transfer techniques. The addressed for insolated I/O device, called ports, are separate from the memory. Because the ports are separate from the memory, because the ports are separate. The user can expand the memory to its full size without using any of memory space for I/O device.
- A disadvantage of isolated I/O is that, the data transferred between I/O and the MPU must be accessed by the IN, OUT instruction.



Memory - Map I/O

- Unlike isolated I/O, memory mapped I/O does not use the IN or OUT instruction.
- Instead, it uses any instruction that transfer data between the MPU and memory. A memory mapped I/O device is treated as a memory location in memory map.
- The main advantage of memory-mapped I/O is that any memory transfer instruction can be used to access the I/O
- The main disadvantage is that a portion of the memory systems used as the I/O map. This reduced the amount of memory available to application.