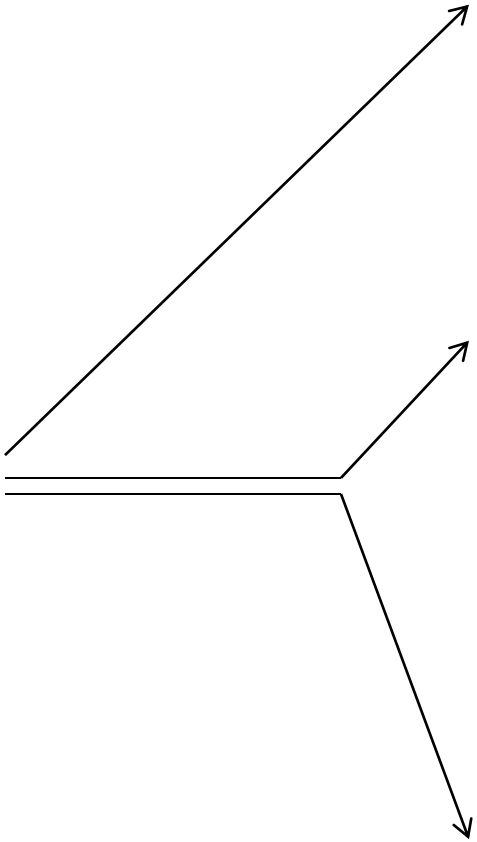
Interfacing memory and i/o ports

Memory



Processor Memory

- Registers inside a microcomputer
- Store data and results temporarily
- No speed disparity
- Cost ↑

Primary or Main Memory

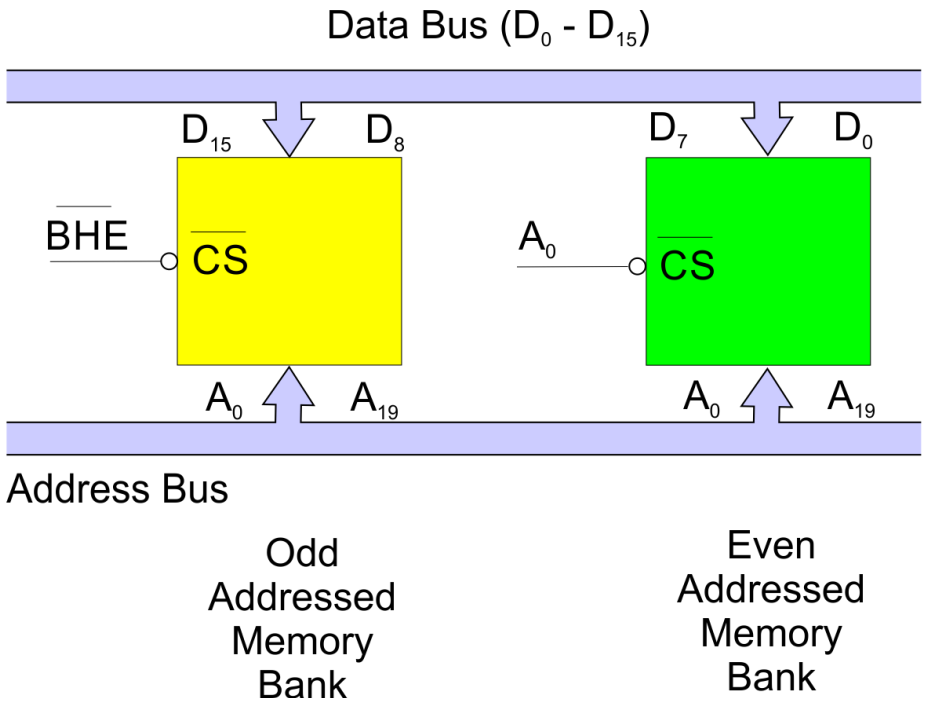
- Storage area which can be directly accessed by microprocessor
- Store programs and data prior to execution
- Should not have speed disparity with processor ⇒ Semi Conductor memories using CMOS technology
- ROM, EPROM, Static RAM, DRAM

Secondary Memory

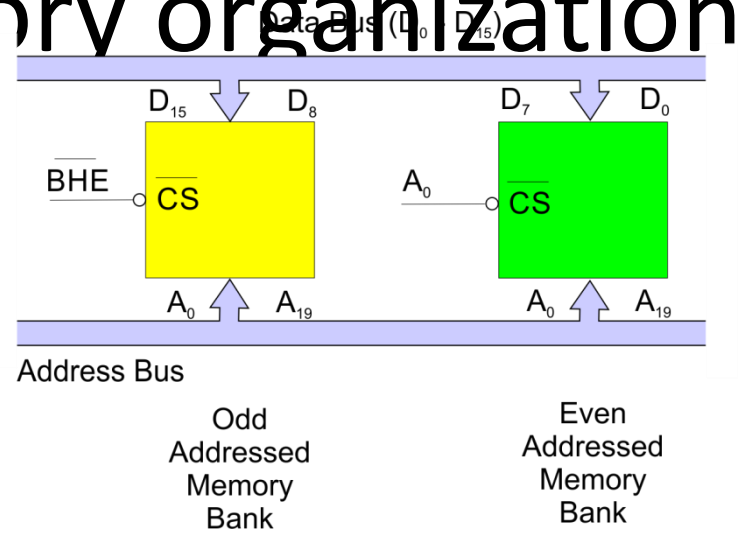
- Storage media comprising of slow devices such as magnetic tapes and disks
- Hold large data files and programs: Operating system, compilers, databases, permanent programs etc.

Memory organization in 8086

- **Memory IC's : Byte oriented**
- **8086 : 16-bit**
- **Word : Stored by two consecutive memory locations; for LSB and MSB**
- **Address of word : Address of LSB**
- **Bank 0 : $A_0 = 0 \Rightarrow$ Even addressed memory bank**
- **Bank 1 : $\overline{BHE} = 0 \Rightarrow$ Odd addressed memory bank**



Memory organization in 8086



	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
		1	0	$D_7 - D_0$ in first operation byte from odd bank is transferred

Memory organization in 8086

- **Available memory space = EPROM + RAM**
- **Allot equal address space in odd and even bank for both EPROM and RAM**
- **Can be implemented in two IC's (one for even and other for odd) or in multiple IC's**

Interfacing SRAM and EPROM

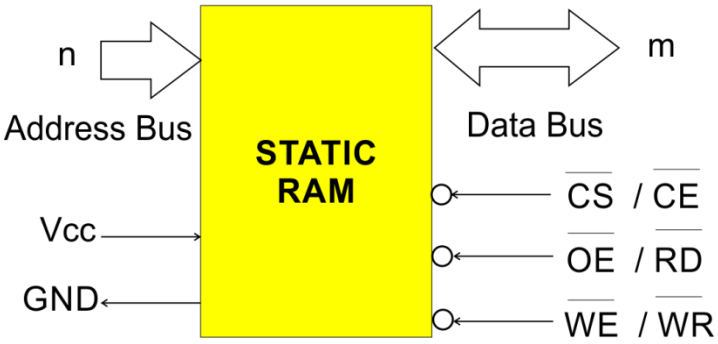
- Memory interface ⇒ Read from and write in to a set of semiconductor memory IC chip
- EPROM ⇒ Read operations
- RAM ⇒ Read and Write

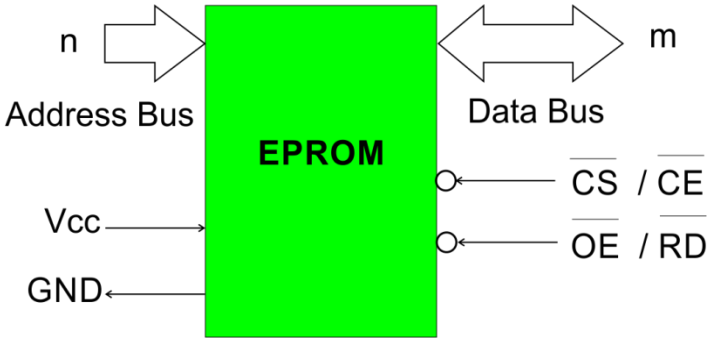
In order to perform read/ write operations,

- Memory access time < read / write time of the processor
- Chip Select (CS) signal has to be generated
- Control signals for read / write operations
- Allot address for each memory location

Interfacing SRAM and EPROM

Typical Semiconductor IC Chip





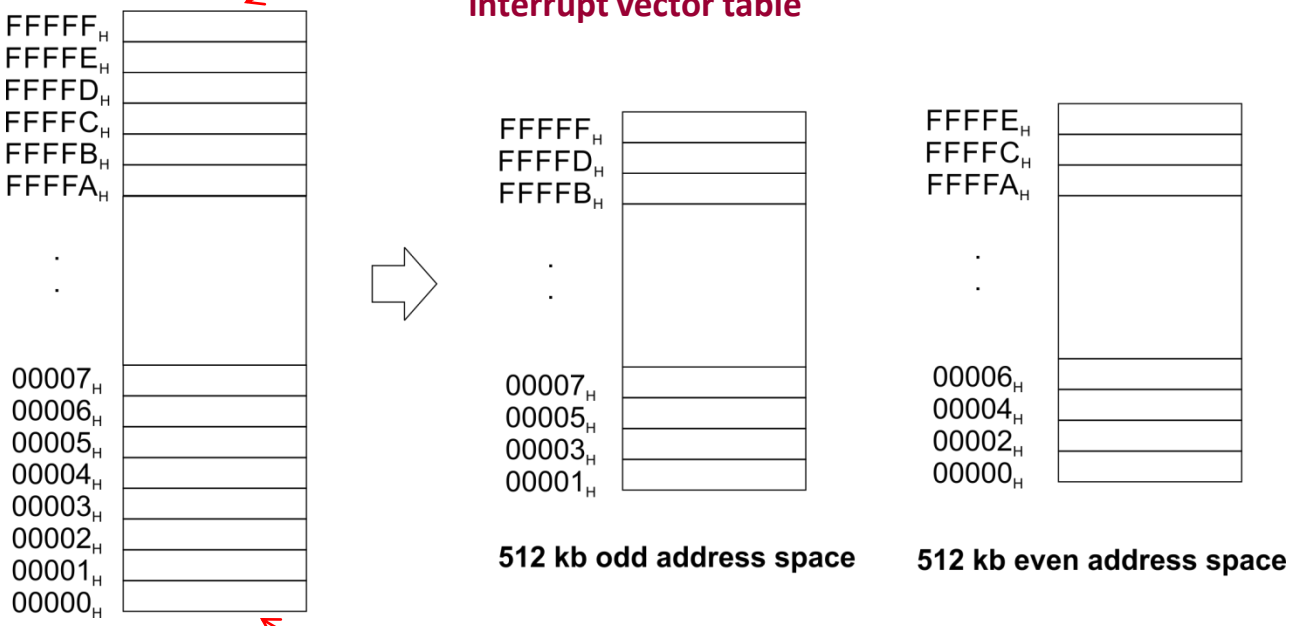
No of Address pins	Memory capacity			Range of address in hexa
	In Decimal	In kilo	In hexa	
20	2^{20}= 10,48,576	1024 k = 1M	100000	00000 to FFFFF

Interfacing SRAM and EPROM

Memory map of 8086

EPROM's are mapped at $FFFFF_H$

⇒ Facilitate automatic execution of monitor programs and creation of interrupt vector table



RAM are mapped at the beginning; 00000H is allotted to RAM

Interfacing SRAM and EPROM

Monitor Programs

⇒ Programing 8279 for keyboard scanning and display refreshing

⇒ Programming peripheral IC's 8259, 8257, 8255, 8251, 8254 etc

⇒ Initialization of stack

⇒ Display a message on display (output)

⇒ Initializing interrupt vector table

Note :

8279 Programmable keyboard/ display controller

8257 DMA controller

8259 Programmable interrupt controller

8255 Programmable peripheral interface

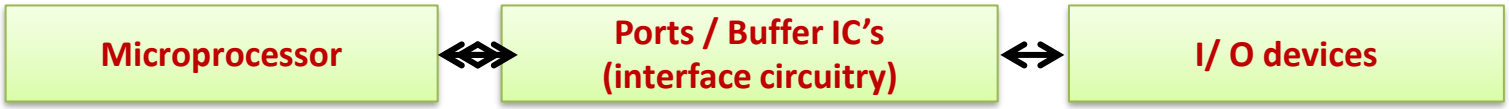
Interfacing I/O and peripheral devices

I/O devices

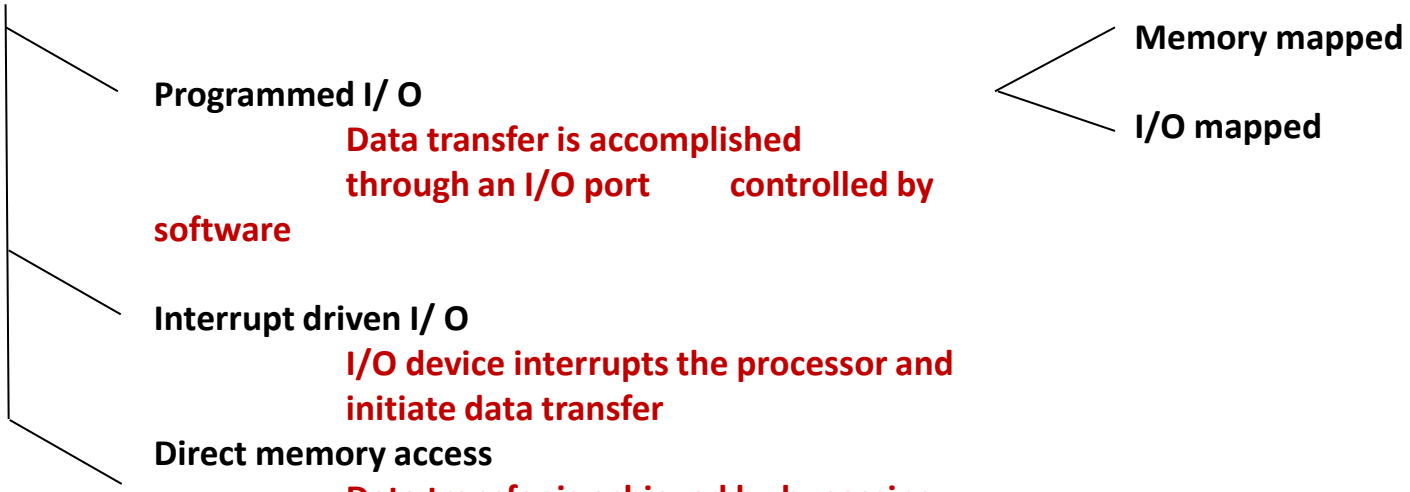
⇒ For communication between microprocessor and outside world

⇒ Keyboards, CRT displays, Printers, Compact Discs etc.

⇒



⇒ Data transfer types



8086 and 8088 comparison

Memory mapping	I/O mapping
20 bit address are provided for I/O devices	8-bit or 16-bit addresses are provided for I/O devices
The I/O ports or peripherals can be treated like memory locations and so all instructions related to memory can be used for data transmission between I/O device and processor	Only IN and OUT instructions can be used for data transfer between I/O device and processor
Data can be moved from any register to ports and vice versa	Data transfer takes place only between accumulator and ports
When memory mapping is used for I/O devices, full memory address space cannot be used for addressing memory. ⇒ Useful only for small systems where memory requirement is less	Full memory space can be used for addressing memory. ⇒ Suitable for systems which require large memory capacity
For accessing the memory mapped devices, the processor executes memory read or write cycle. ⇒ M / $\overline{IO}$ is asserted high	For accessing the I/O mapped devices, the processor executes I/O read or write cycle. ⇒ M / $\overline{IO}$ is asserted low