The 8086 Memory Interface

Memory Devices

- Simple or complex, every microprocessor-based system has a memory system.
- Almost all systems contain four common types of memory:
  - Read only memory (ROM)
  - Flash memory (EEPROM)
  - Static Random access memory (SARAM)
  - Dynamic Random access memory (DRAM).
- Before attempting to interface memory to the microprocessor, it is essential to understand the operation of memory components.

Memory Pin Connections

Figure (1) shows a general form diagram of ROM and RAM pins. Pin connections common to all memory devices are:

- Address connections
- Data connections
- Selection connections
- Control connections

Address connections: All memory devices have address inputs that select a memory location within the memory device. Address inputs are labeled from A_0 to A_n

Data connections: All memory devices have a set of data outputs or input/outputs. Today many of them have bi-directional common I/O pins.

Selection connections: Each memory device has an input that selects or enables the memory device. This kind of input is most often called a chip select ($\overline{\mathbf{CS}}$), chip enable ($\overline{\mathbf{CE}}$) or simply select ($\overline{\mathbf{S}}$) input.

- RAM memory generally has at least one $\overline{\mathbf{CS}}$ or $\overline{\mathbf{S}}$ input and ROM at least one $\overline{\mathbf{CE}}$.
- If the $\overline{\mathbf{CE}}$, $\overline{\mathbf{CS}}$, $\overline{\mathbf{S}}$ input is active the memory device perform the read or write.
- If it is inactive the memory device cannot perform read or write operation.
- If more than one $\overline{\mathbf{CS}}$ connection is present, all most be active to perform read or write data.

Control connections:

- A ROM usually has only one control input, while a RAM often has one or two control inputs.
- The control input most often found on the ROM is the output enable ($\overline{\mathbf{OE}}$) or gate ($\overline{\mathbf{G}}$), this allows data to flow out of the output data pins of the ROM.

- A RAM memory device has either one or two control inputs. If there is one control input it is often called $\mathbf{R}/\overline{\mathbf{W}}$.
- This pin selects a read operation or a write operation only if the device is selected by the selection input ($\overline{\mathbf{CS}}$)

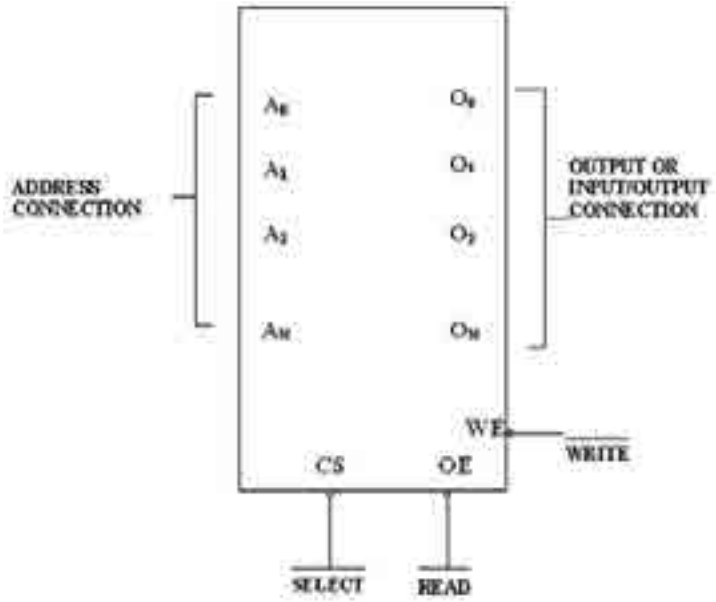


Fig. 1: Memory Component

Read-only memory (ROM)

- Read-only memory (**ROM**) permanently stores programs/data resident to the system, and must not change when power disconnected
- Often called nonvolatile memory, because its contents *do not* change even if power is disconnected.
- A device we call a ROM is purchased in mass quantities from a manufacturer. programmed during fabrication at the factory
- The **EPROM** (erasable programmable read-only memory) is programmed in the field on a device called an EPROM programmer.
- Also erasable if exposed to high-intensity ultraviolet light, depending on the type of EPROM.
- The **PROM** (programmable read-only memory) is also programmed in the field by burning open tiny NIchrome or silicon oxide fuses. Once it is programmed, it cannot be erased.
- A newer type of read-mostly memory (**RMM**) is called the flash memory.
- Flash memory is also often called an **EEPROM** (electrically erasable programmable ROM) or **EAROM** (electrically alterable ROM) or a **NOVRAM** (nonvolatile RAM)
- Electrically erasable in the system, but they require more time to erase than normal RAM.
- The flash memory device is used to store setup information for systems such as the video card in the computer.

Static Random Access Memory (SRAM)

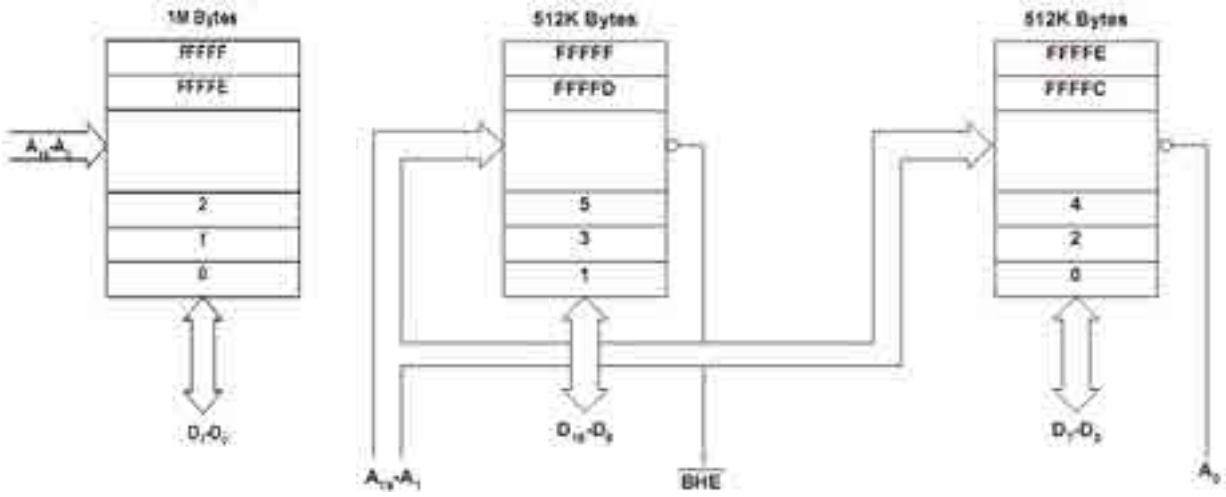
- A Static RAM is a volatile memory device which means that the contents of the memory array will be lost if power is removed.
- Unlike a dynamic memory device, the static memory does not require a periodical refresh cycle and generally runs much faster than a dynamic memory device.
- Static RAM is used when the size of the read/write memory is relatively small, today, a small memory is less than 1M byte.
- The main difference between ROM and RAM is that RAM is written under normal operation, whereas ROM is programmed outside the computer and normally is only read.

Dynamic Random Access Memory (DRAM)

- Available up to 256M X 8 (2G bits).
- DRAM is essentially the same as SRAM, except that it retains data for only 2 or 4 ms on an integrated capacitor.
- After 2 or 4 ms, the contents of the DRAM must be completely rewritten *(refreshed)*, because the capacitors, which store a logic 1 or logic 0, lose their charges.

8086 Memory Interface

- The memory address space of the 8086-based microcomputers has different logical and physical **organizations** (see **Fig. 2**).



(a) **(b)**

Fig. 2: (a) Logical memory organization, and (b) Physical memory organization (high and low memory banks) of the 8086 microprocessor.

- **Logically**, memory is implemented as a single **1M × 8 memory chunk**. The byte-wide storage locations are assigned consecutive addresses over the range from 00000H through FFFFFH
- **Physically**, memory is implemented as **two** independent **512 Kbyte banks**: the **low (even) bank** and the **high (odd) bank**. Data bytes associated with an even address (00000H,

00002H, etc.) reside in the low bank, and those with odd addresses (00001H, 00003H, etc.) reside in the high bank.

- Address bits A_1 through A_{19} select the storage location that is to be accessed. They are applied to both banks in parallel. **A_0** and bank high enable (**$\overline{\text{BHE}}$**) are used as **bank-select** signals.
- The memory locations 00000-FFFFF are designed as odd and even bytes. To distinguish between odd and even bytes, the CPU provides a signal called **$\overline{\text{BHE}}$** (bus high enable). **$\overline{\text{BHE}}$** and A_0 are used to select the odd and even byte, as shown in the table below.

$\overline{\text{BHE}}$	**A0**	**Function**
0	0	Choose both odd and even memory bank
0	1	Choose only odd memory bank
1	0	Choose only even memory bank
1	1	None is chosen

Minimum mode Memory Interface

- Figure (3) show block diagram of minimum mode 8086 memory interface.

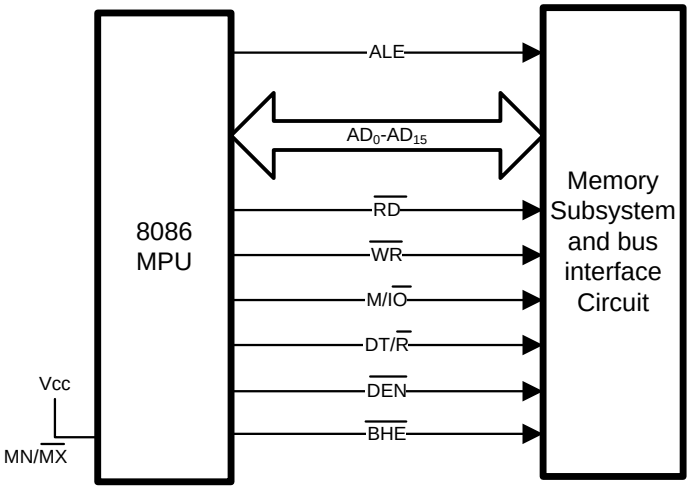


Fig. 3: Minimum mode memory interface

- The control signals provided to support the interface to the memory subsystem are **ALE**, **$M/\overline{IO}$**, **$DT/\overline{R}$**, **$\overline{RD}$**, **$\overline{WR}$**, **DEN** and **$\overline{BHE}$**
- When ***Address latch enable*** (**ALE**) is **logic 1** it signals that a **valid address** is on the bus. This address can be latched in external circuitry on the **1-to-0 edge** of the pulse at ALE.
- **$M/\overline{IO}$** (*memory*/IO) and **$DT/\overline{R}$** tells external circuitry whether a memory or I/O transfer is taking place over the bus, and whether the 8086 will transmit or receive data over the bus.

- The *bank high enable* ($\overline{\mathbf{BHE}}$) signal is used as a **memory enable signal** for the **most significant byte** half of the data bus, **D8** through **D15**.
- The signals $\overline{\mathbf{WR}}$ (*write*) and $\overline{\mathbf{RD}}$ (*read*) identify that a write or read bus cycleis in progress.
- **DEN** (*data enable*), is also supplied. It enables external devices to supply data to the microprocessor.

Maximum mode Memory Interface

- Figure (4) show block diagram of maximum mode memory interface.
- In maximum mode the 8086 not directly provides all control signal to support the memory interface.
- Instead, an external Bus Controller (8288) provides memory commands and control signals as shown in table (5) in lecture (8).

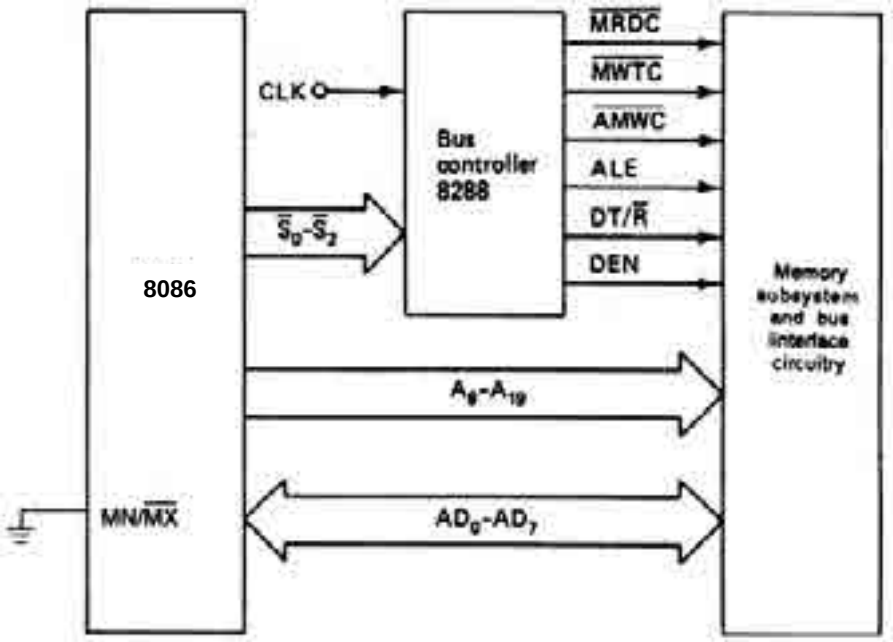


Fig. 4: Maximum mode memory interface

Memory expansion

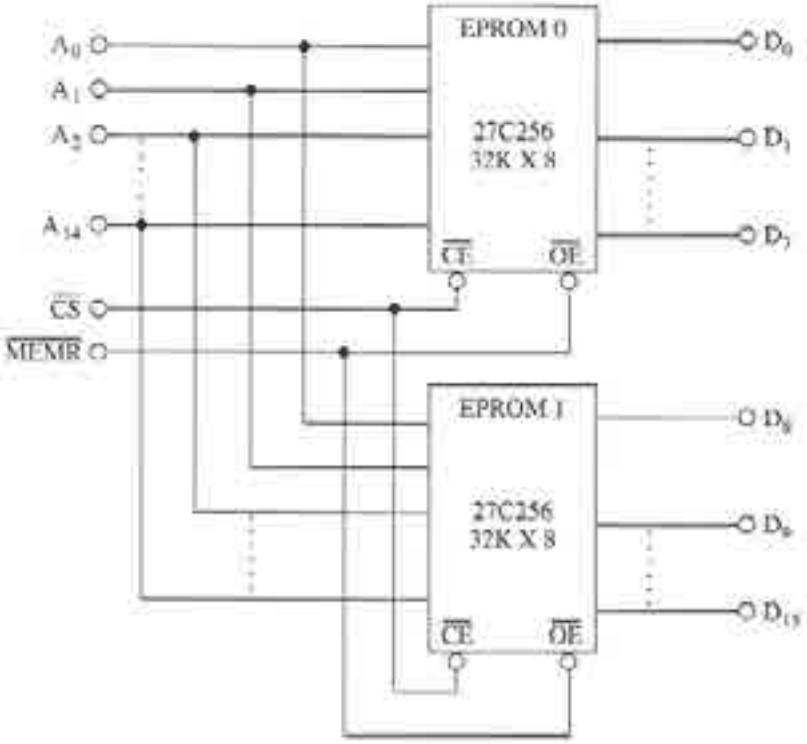
In many applications, the microcomputer system requirement for memory is greater than what is available in a single device. There are two basic reasons for expanding memory capacity:

1. The byte-wide length is not large enough
2. The total storage capacity is not enough bytes.

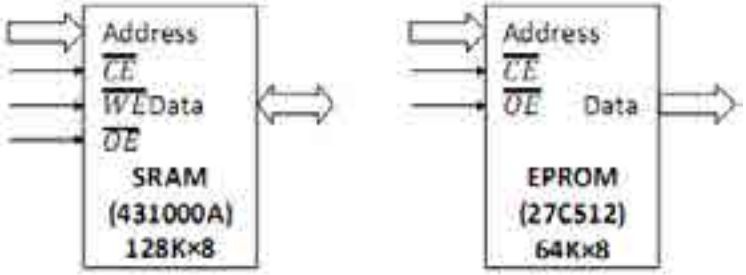
Both of these expansion needs can be satisfied by interconnecting a number of ICs.

.

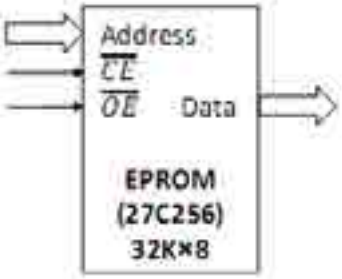
Example 1: show how to implement 32K× 16 EPROM using two 32K×8 EPROM?
Solution:



Example 2: Design 8086's memory system consisting of 512K bytes of RAM memory and 128K bytes of ROM use the devices in figure below. RAM memory is to reside over the address range 00000H through 7FFFFH and the address range of the ROM is to be A0000H through BFFFFH



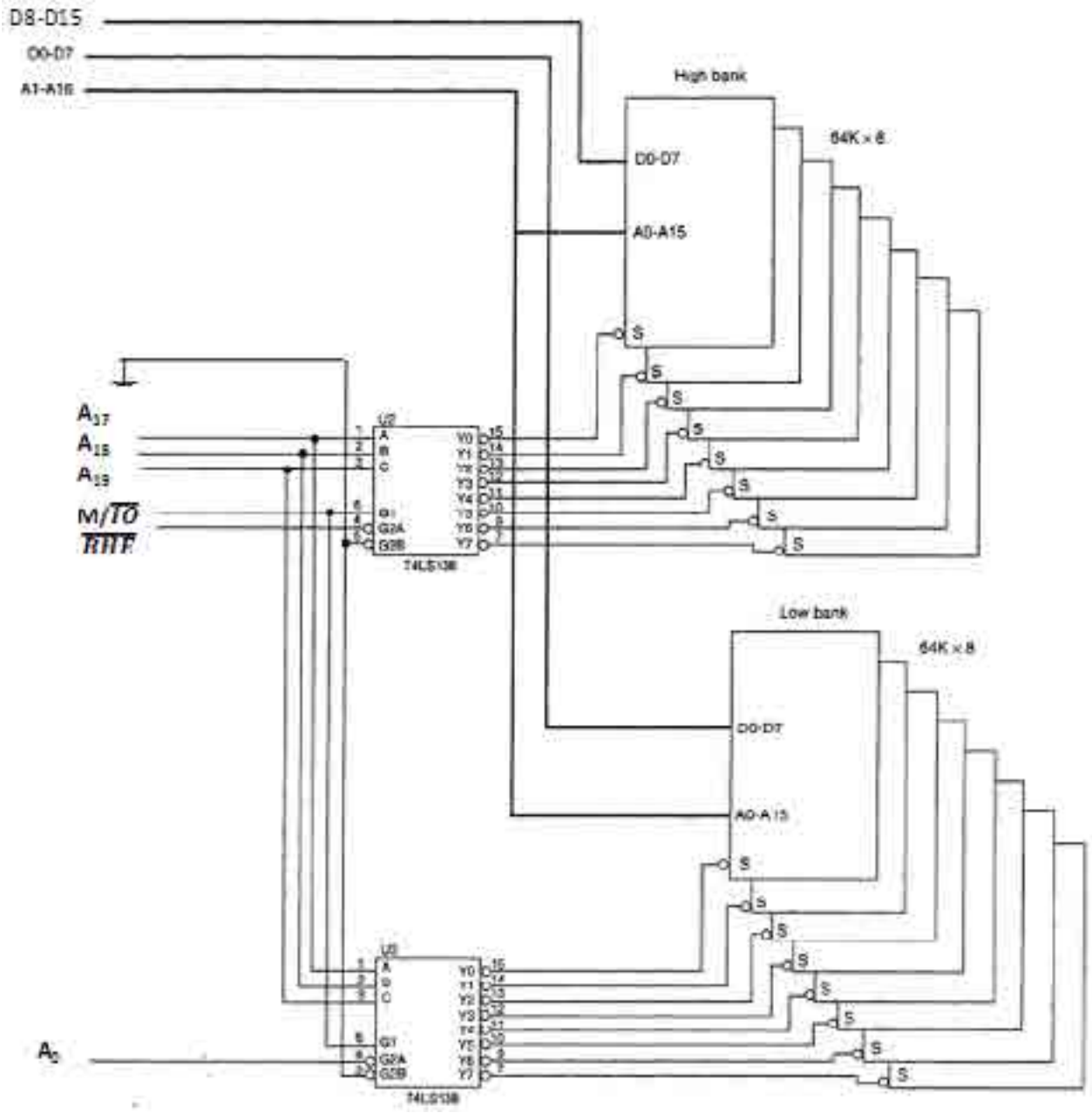
Example 3: Design 8086's memory system consisting of 64K bytes of ROM memory, make use of the devices in figure below. The memory is to reside over the address range 60000H through 6FFFFH



.

Example 4: Design a 8086 memory system consisting of 1Mbytes, Using 64K× 8 memory.

Solution:



Example5: show how to implement 64K× 8 EPROM using two 32K×8 EPROM?

Example5: show how to implement 32K× 32 EPROM using four 32K×8 EPROM?