

**VIVEKANANDA COLLEGE
THAKURPUKUR
KOLKATA-700063**

NAAC ACCREDITED 'A' GRADE



Topic: Basic Concepts of Memory Interfacing

Course Title: Microprocessor and its Application

Paper: CMS-A-CC-4-10-TH

Unit: Microprocessor Architecture and Memory Interfacing

Semester: 4th Semester

Name of the Teacher: Amitava Biswas

Name of the Department: Computer Science

Introduction:

Memory is an integral part of a microprocessor system, and in this section, we will discuss how to interface a memory device with the microprocessor. The Memory Interfacing in 8085 is used to access memory quite frequently to read instruction codes and data stored in memory. This read/write operations are monitored by control signals. The microprocessor activates these signals when it wants to read from and write into memory. In the last section we have already seen the memory read and memory write machine cycles, and status of the RD, WR and IO/M status signals for read/write operation. In the following section we will see memory structure and its requirements, concepts in Memory Interfacing in 8085 and interfacing examples.

As we know that any system which process digital data needs the facility for storing the data. Interfacing is a technique to be used for connecting the Microprocessor to Memory.

Now a days Semiconductor memories are used for storing purpose. There are some of the advantages of the semiconductor memory:

- Small size
- High speed
- Better reliability
- Low cost

Generally, RAM or ROM is used for memory interfacing.

Terminologies:

1. Memory:-A memory is a digital IC which stores the data in binary form.
2. Memory Size:-The number of location and number of bits per word will vary from memory to memory. For example, If a particular memory chip is capable of storing M words with each word having N-bits. Then the size of the memory will be $M \times N$.
3. Interfacing a ROM memory of 4096×8 with 8085 Microprocessor:-

Given memory size = 4096×8

$4096 = 2^{12}$.

So 12 lines will be used for interfacing. A0 to A11

In this system A0 to A11 lines of Microprocessor will be connected to the address lines of the memory. and D0 to D7 of the 8085 microprocessor will be connected to the data bus of the memory. As we know that the it is EPROM, so only RD pin is connected to the microprocessor. There is not the facility for writing data.

Memory Structure and its Requirements:

As mentioned earlier, read/write memories consist of an array of registers, in which each register has unique address. The size of the memory is $N \times M$ as shown in Fig. 4.13 (a) where N is the number of registers and M is the word length, in number of bits.

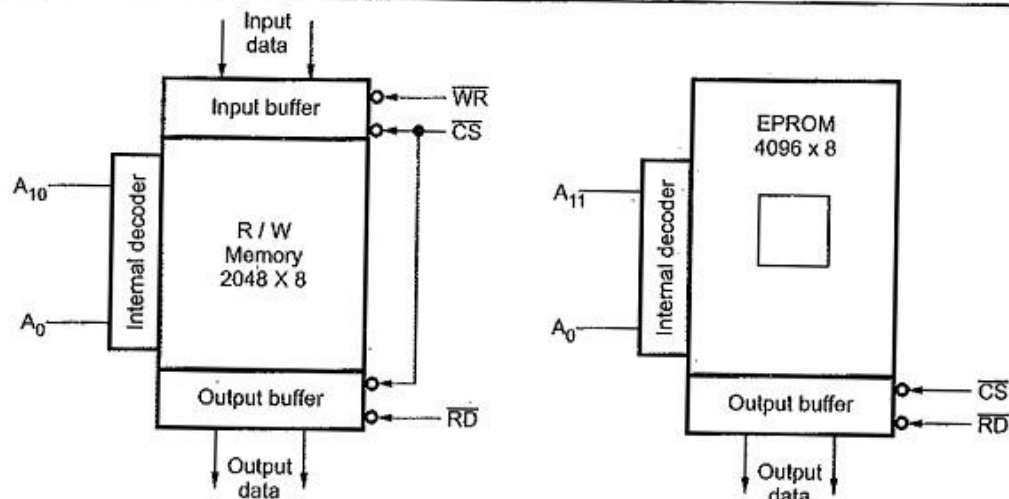


Fig. 4.13 (a) Logic diagram for RAM Fig. 4.13 (b) Logic diagram for EPROM

Basic Concepts in Memory Interfacing:

1. Microprocessor 8085 can access 64Kbytes memory since address bus is 16-bit. But it is not always necessary to use full 64Kbytes address space. The total memory size depends upon the application.
2. Generally EPROM (or EPROMs) is used as a program memory and RAM (or RAMs) as a data memory. When both, EPROM and RAM are used, the total address space 64Kbytes is shared by them.
3. The capacity of program memory and data memory depends on the application.
4. It is not always necessary to select 1 EPROM and 1 RAM. We can have multiple EPROMs and multiple RAMs as per the requirement of application.
5. We can place EPROM/RAM anywhere in full 64 Kbytes address space. But program memory (EPROM) should be located from address 0000H since reset address of 8085 microprocessor is 0000H.
6. It is not always necessary to locate EPROM and RAM in consecutive memory For example : If the mapping of EPROM is from 0000H to 0FFFH, it is not must to locate RAM from 1000H. We can locate it anywhere between 1000H and FFFFH. Where to locate memory component totally depends on the application.

The memory interfacing requires to :

- **Select the chip**
- **Identify the register**
- **Enable the appropriate buffer.**

Microprocessor system includes memory devices and I/O devices. It is important to note that microprocessor can communicate (read/write) with only one device at a time, since the data, address and control buses are common for all the devices. In order to communicate with memory or I/O devices, it is necessary to decode the address from the microprocessor. Due to this each device (memory or I/O) can be accessed independently. The following section describes common address decoding techniques.

Address Decoding Techniques :

- **Absolute decoding/Full Decoding**
- **Linear decoding/Partial Decoding**

Absolute decoding:

In absolute decoding technique, all the higher address lines are decoded to select the memory chip, and the memory chip is selected only for the specified logic levels on these high-order address lines; no other logic levels can select the chip. Fig. 4.14 shows the

Memory Interfacing in 8085 with absolute decoding. This addressing technique is normally used in large memory systems.

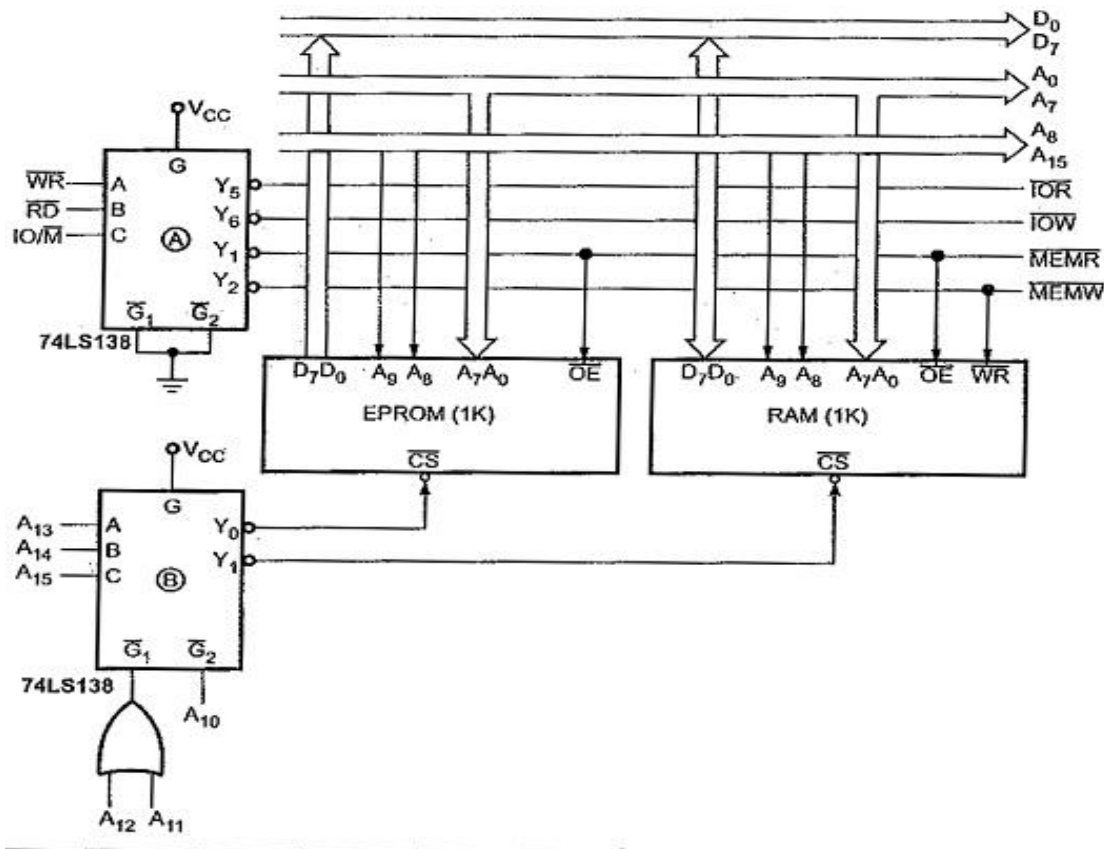


Fig. 4.14 Absolute decoding technique

Linear decoding:

In small systems, hardware for the decoding logic can be eliminated by using individual high-order address lines to select memory chips. This is referred to as linear decoding. Fig. 4.15 shows the addressing of RAM with linear decoding technique. This technique is also called **partial decoding**. It reduces the cost of decoding circuit, but it has a drawback of multiple addresses (shadow addresses).

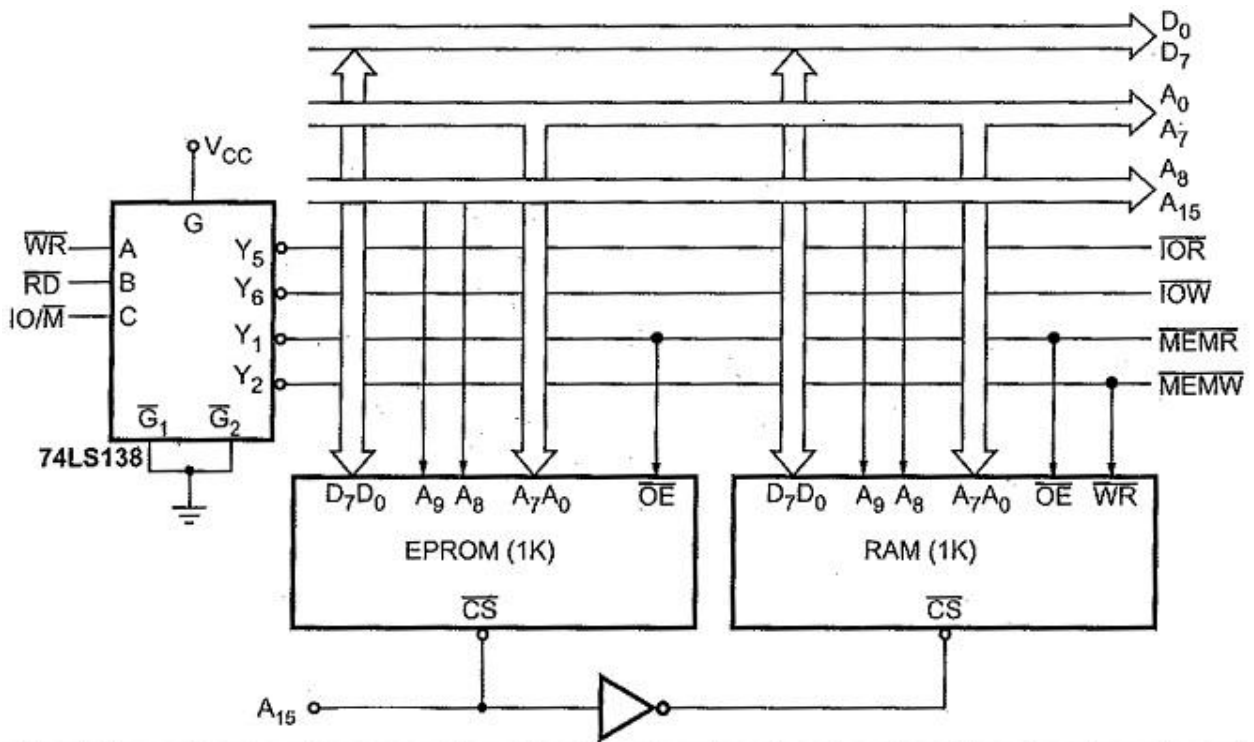


Fig. 4.15 Linear decoding

Fig. 4.15 shows the addressing of RAM with linear decoding technique. A_{15} address line, is directly connected to the chip select signal of EPROM and after inversion it is connected to the chip select signal of the RAM. Therefore, when the status of A_{15} line is 'zero', EPROM gets selected and when the status of A_{15} line is 'one' RAM gets selected. The status of the other address lines is not considered, since those address lines are not used for generation of chip select signals.

Wait State Generator:

We have seen that wait states can be introduced in the machine cycles using READY signal to interface slower memory devices. In this section we are going to see the circuits for wait state generator and applications of those circuits to interface slower memories.

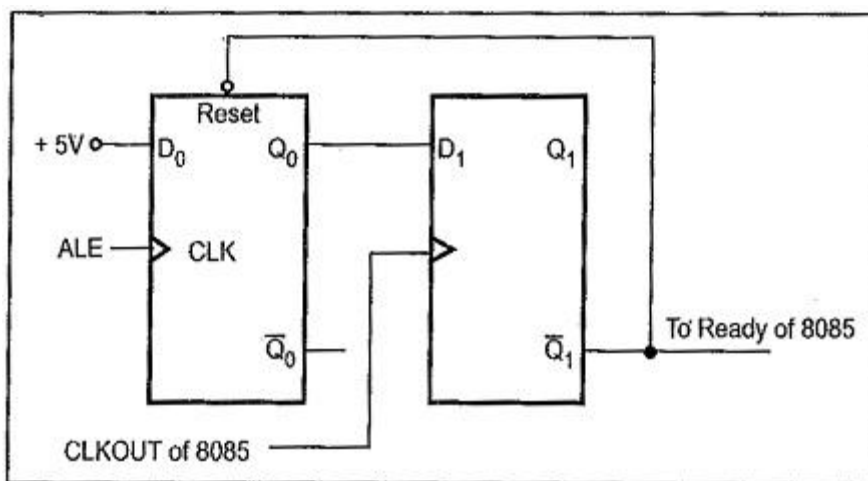


Fig. 4.22 Wait state generator

The Fig. 4.22 shows the circuit for wait state generator. This circuit generates single wait state in every machine cycle. The circuit operation is given below.

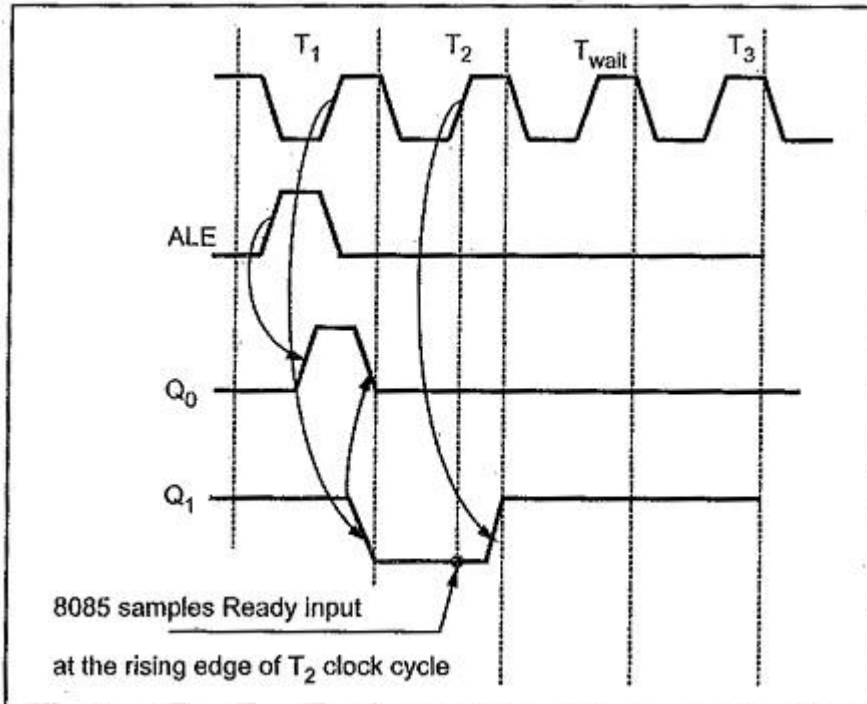


Fig. 4.23 Timing waveforms for wait state generator circuit

In T_1 , the ALE signal is activated, which makes output Q_0 of the first D flip-flop high, assuming RESET (R) is one. In the next clock pulse i.e. in the next low to high transition of CLKOUT signal of the 8085, the output Q_1 of the second D flip-flop goes low. The low on the Q_1 output initiates 8085 to enter wait state and resets the first D flip-flop making its Q_0 output low, since Q_1 is connected to the ready input of 8085 and to the reset input of the first D flip-flop. At next low to high transition of CLKOUT, Q of the second D flip-flop goes high making READY pin high and inactivating reset input of the first flip-flop. Fig. 4.23 shows the timing waveforms for this circuit.

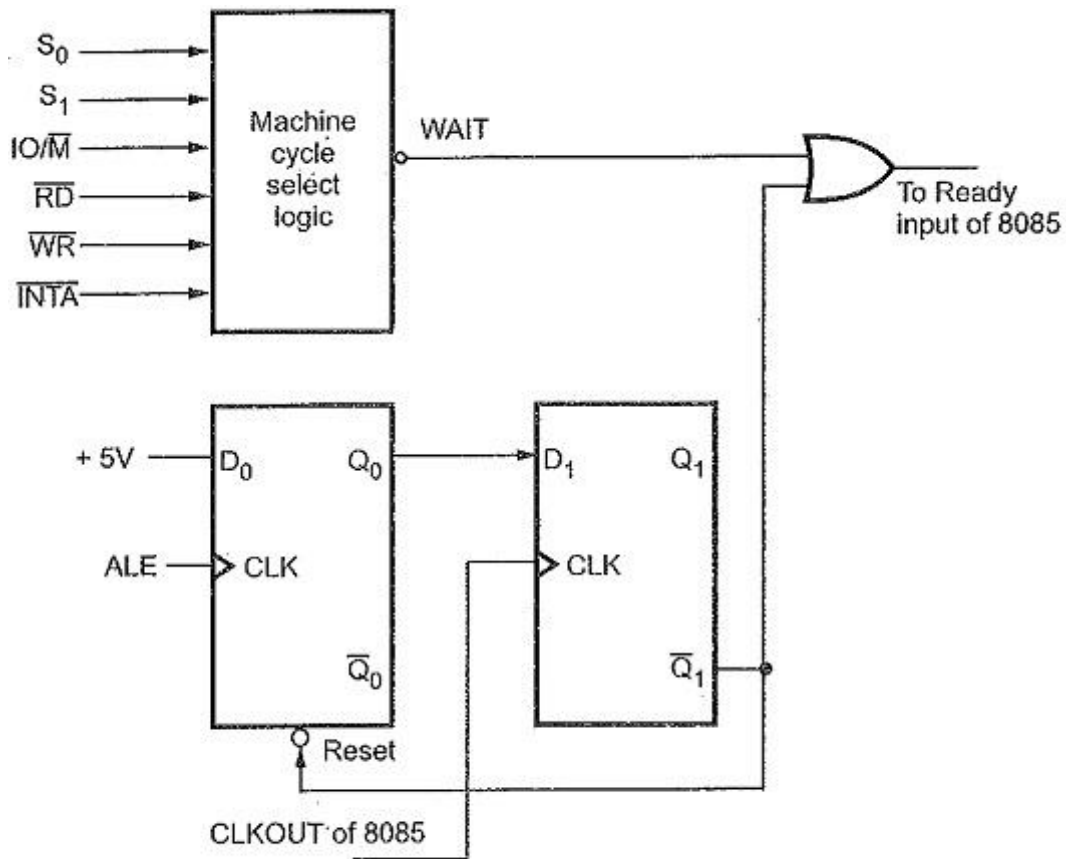


Fig. 4.24 Modified Wait State Generator Circuit

This circuit can be modified to introduce wait states for a particular machine cycle. The modified circuit of wait state generator is shown in Fig. 4.24. The modified circuit will generate a single WAIT state for any machine [cycle](#) for which the WAIT output of the machine cycle select circuit is logic 0.