

# Module 2

## Embedded Processors and Memory

Lesson

12

Memory-Interfacing

## Instructional Objectives

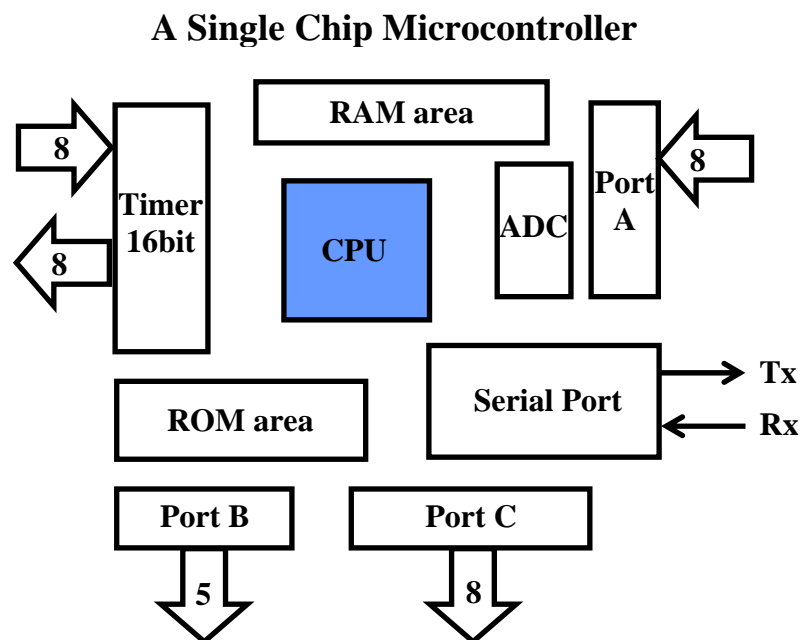
After going through this lesson the student would learn

- Requirement of External Memory
- Different modes of a typical Embedded Controller
- Standard Control Signals for Memory Interface
- A typical Example

## Pre-Requisite

Digital Electronics, Microprocessors

## 12.1 Introduction



**CPU: The processing module of the microcontroller**

**Fig. 12.1 The basic architecture of a Microcontroller**

Fig. 12.1 shows the internal architecture of single chip microcontroller with internal RAM as well as ROM. Most of these microcontrollers do not require external memory for simpler tasks. The program lengths being small can easily fit into the internal memory. Therefore it often provides single chip solutions. However the amount of internal memory cannot be increased beyond a certain limit because of the following reasons.

- Power Consumption
- Size

The presence of extra memory needs more power consumption and hence higher temperature rise. The size has to be increased to house the additional memory. The need for extra memory

space arises in some specific applications. Fig. 12.2 shows the basic block diagram of memory interface to a processor.

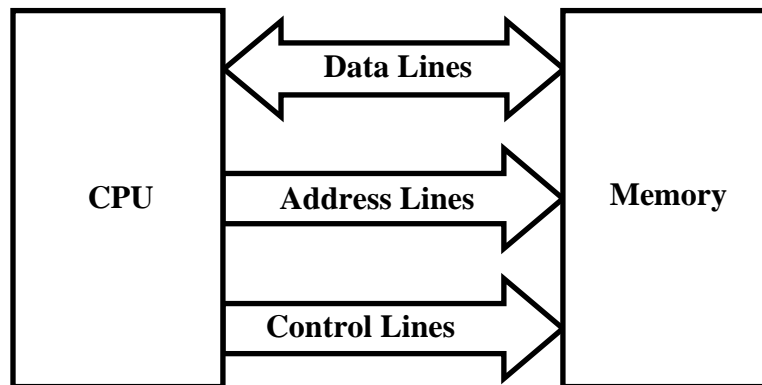


Fig. 12.2 The Memory Interface

## 12.2 External Memory Interfacing to PIC18F8XXX family of microcontrollers

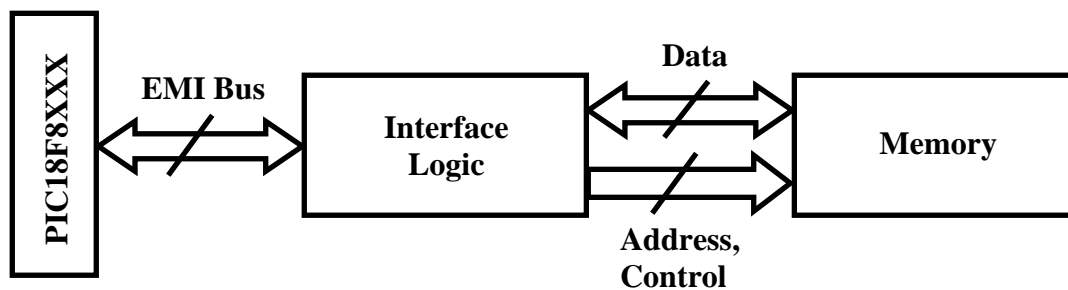


Fig. 12.3 External Memory Interface Diagram

The above family of microcontroller can have both on-chip as well as off chip external memory. At times the on-chip memory is a programmable flash type. A special register inside the microcontroller can be programmed (by writing an 8 bit or 16-bit binary number) for using this external memory in various modes. In case of the PIC family the following modes are possible

### Microcontroller Mode

The processor accesses only on-chip FLASH memory. External Memory Interface functions are disabled. Attempts to read above the physical limit of the on-chip FLASH causes a read of all '0's (a NOP instruction).

### Microprocessor Mode

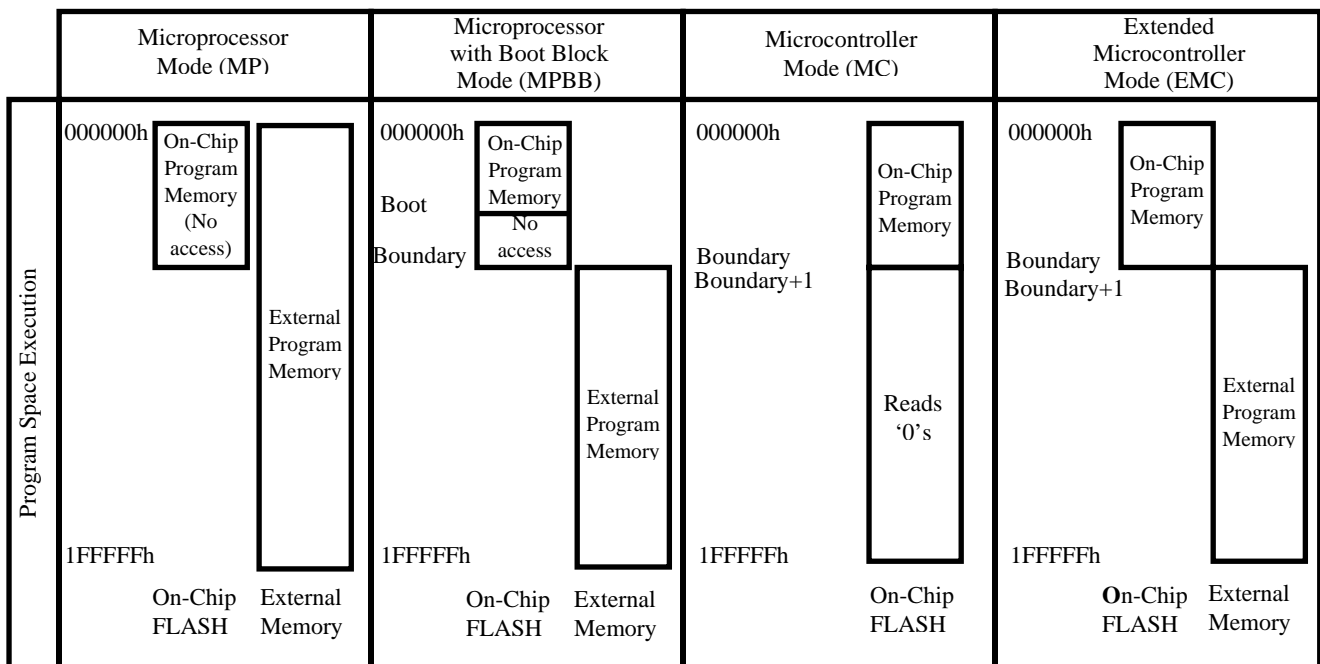
The processor permits execution and access only through external program memory; the contents of the on-chip FLASH memory are ignored.

## Microprocessor with Boot Block mode

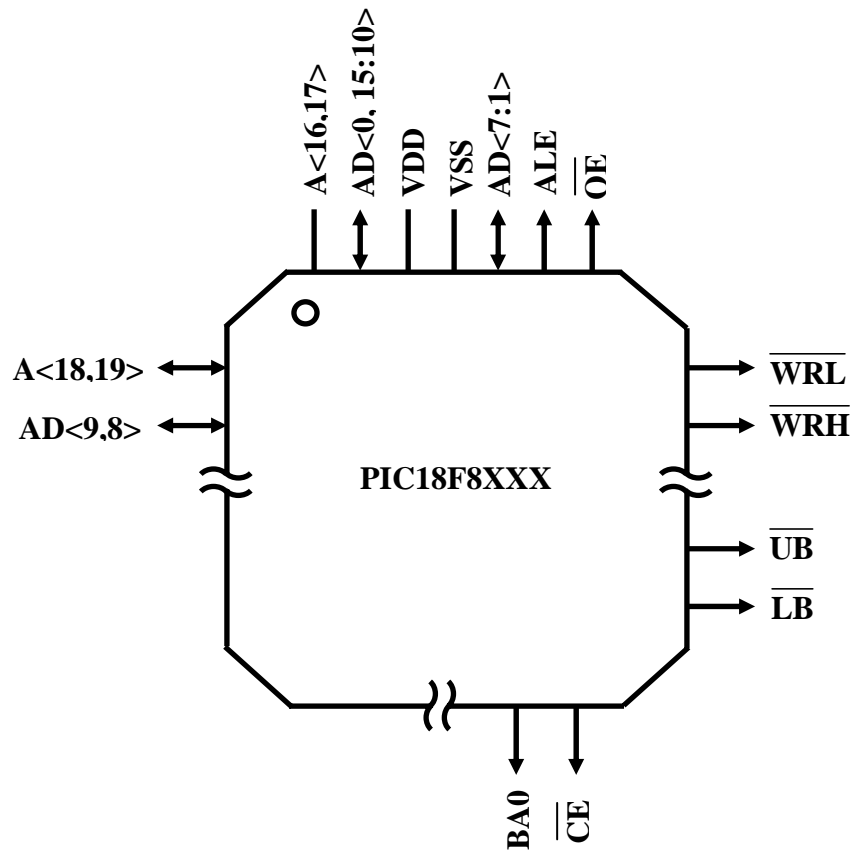
The processor accesses on-chip FLASH memory within only the boot block. The boot block size is device dependent and is located at the beginning of program memory. Beyond the boot block, external program memory is accessed all the way up to the 2-MByte limit. Program execution automatically switches between the two memories as required.

## Extended Microcontroller Mode

The processor accesses both internal and external program memories as a single block. The device can access its entire on-chip FLASH memory; above this, the device accesses external program memory up to the 2-MByte program space limit. As with Boot Block mode, execution automatically switches between the two memories as required.



**Fig. 12.4 The memory Map in different modes**



**Fig. 12.5 The address, data and control lines of the PIC18F8XXX microcontroller required for external memory interfacing**

The address, data and control lines of a PIC family of microcontroller is shown in Fig. 12.5 and are explained below.

AD0-AD15: 16-bit Data and 16 bits of Address are multiplexed

A16-A19: The 4 most significant bits of the address

ALE: Address Latch Enable Signal to latch the multiplexed address in the first clock cycle

$\overline{\text{WRL}}$  Write Low Control Pin to make the memory write the lower byte of the data when it is low

$\overline{\text{WRH}}$  Write High Control Pin to make the memory write the higher byte of the data when it is low

$\overline{\text{OE}}$  Output Enable is made low when valid data is made available to the external memory

$\overline{\text{CE}}$  Chip enable line is made low to access the external memory chip

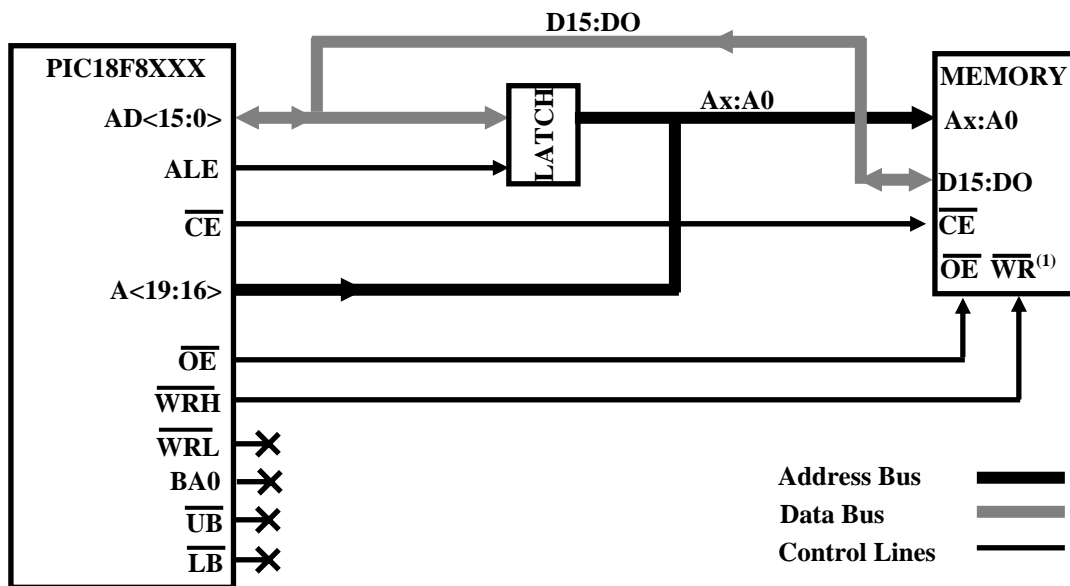
BA0: Byte Address 0

$\overline{\text{LB}}$  Lower Byte Enable Control is kept low when the lower byte is available for the memory.

$\overline{\text{UB}}$  Upper Byte Enable Control is kept low when the upper byte is available for the memory.

The microcontroller has a 16-bit wide bus for data transfer. These data lines are shared with address lines and are labeled AD<15:0>. Because of this, 16 bits of latching are necessary to demultiplex the address and data. There are four additional address lines labeled A<19:16>. The PIC18 architecture provides an internal program counter of 21 bits, offering a capability of 2 Mbytes of addressing.

There are seven control lines that are used in the External Memory Interface: ALE,  $\overline{\text{WRL}}$ ,  $\overline{\text{WRH}}$ ,  $\overline{\text{OE}}$ ,  $\overline{\text{CE}}$ ,  $\overline{\text{LB}}$ ,  $\overline{\text{UB}}$ . All of these lines except  $\overline{\text{OE}}$  may be used during data writes. All of these lines except  $\overline{\text{WRL}}$  and  $\overline{\text{WRH}}$  may be used during fetches and reads. The application will determine which control lines are necessary. The basic connection diagram is shown in Fig. 12.6. The 16-bit byte select mode is shown here.



**Fig. 12.6 The connection diagram for external memory interface in 16-bit byte select mode**

The PIC18 family runs from a clock that is four times faster than its instruction cycle. The four clock pulses are a quarter of the instruction cycle in length and are referred to as Q1, Q2, Q3, and Q4. During Q1, ALE is enabled while address information A<15:0> are placed on pins AD<15:0>. At the same time, the upper address information A<19:16> are available on the upper address bus. On the negative edge of ALE, the address is latched in the external latch. At the beginning of Q3, the  $\overline{\text{OE}}$  output enable (active low) signal is generated. Also, at the beginning of Q3, BA0 is generated. This signal will be active high only during Q3, indicating the state of the program counter Least Significant bit. At the end of Q4,  $\overline{\text{OE}}$  goes high and data (16-bit word) is fetched from memory at the low-to-high transition edge of  $\overline{\text{OE}}$ . The timing diagram for all signals during external memory code execution and table reads is shown in Fig. 12.7.

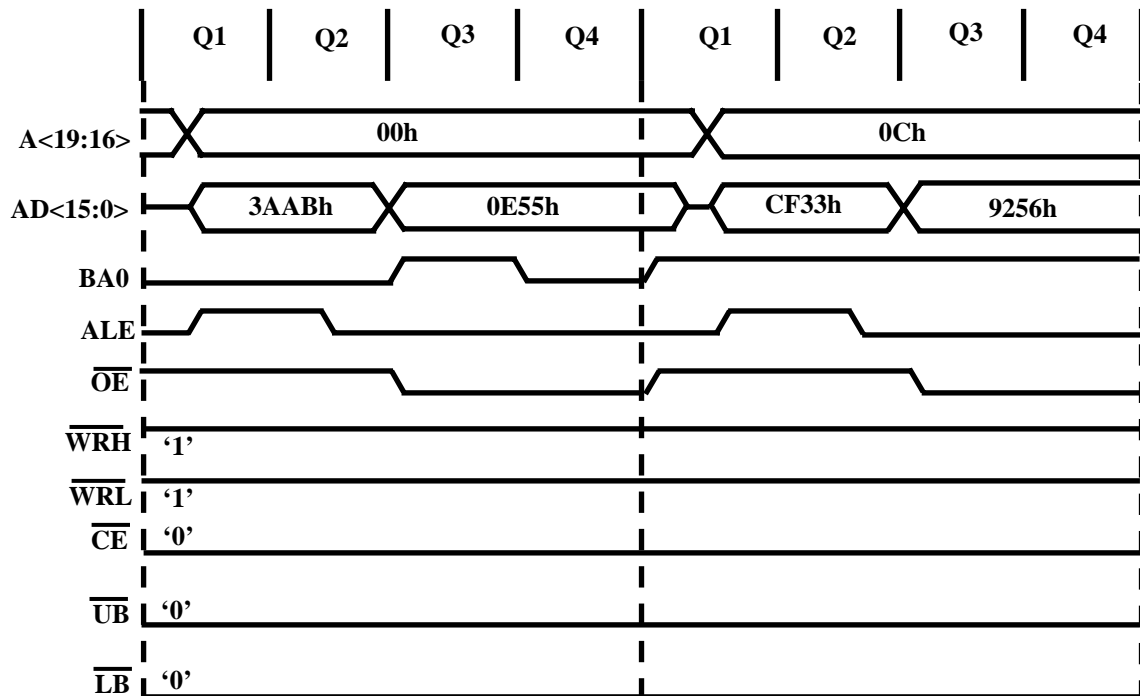


Fig. 12.7 Timing Diagram for Memory Read

## 12.3 Conclusion

This lesson discussed a typical external memory interface example for PIC family of microcontrollers. A typical timing diagram for memory read operation is presented.

## 12.4 Questions

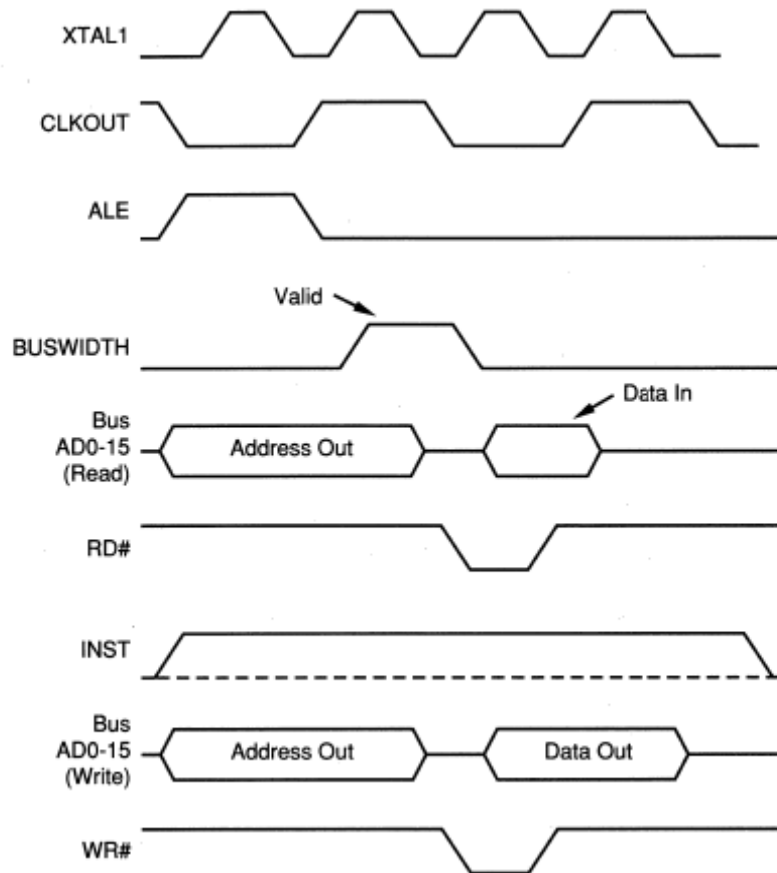
Q1. Draw the read timing diagram for a typical memory operation

**Ans:**

*Refer to text*

Q2. Draw the read timing diagram for a typical memory operation





**16-bit Write Operation in MCS96 family refer Lesson10 and 11**

Source:<http://www.nptel.ac.in/courses/Webcourse-contents/IIT%20Kharagpur/Embedded%20systems/Pdf/Lesson-12.pdf>