

Interfacing to external memory.

The 8051 microcontroller is a popular microcontroller used in various applications, ranging from small embedded systems to larger industrial applications. While the 8051 has a limited amount of internal memory, it is possible to extend the memory capacity by interfacing it with external memory devices.

External memory interfacing in 8051 microcontroller involves connecting external memory devices such as RAM and ROM to the microcontroller to provide additional memory space. This allows the microcontroller to execute larger and more complex programs, store more data, and perform more complex operations.

External memory interfacing typically involves connecting the memory devices to the microcontroller through a data bus and an address bus. The data bus is used to transfer data between the microcontroller and the memory device, while the address bus is used to select a specific memory location in the memory device.

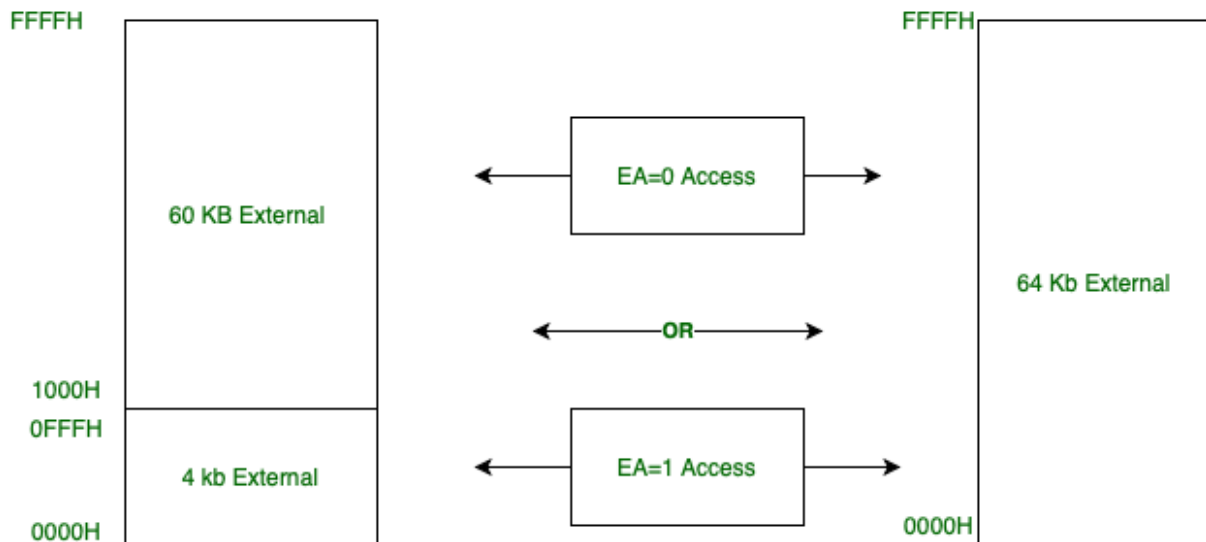
To interface with external memory, the 8051 microcontroller uses dedicated pins such as ALE (Address Latch Enable), PSEN (Program Store Enable), and RD (Read) and WR (Write) signals. These signals are used to control the flow of data between the microcontroller and the external memory device. For minimal memory applications, the 8051 has internal data and code memory. In such a position. For certain applications, this memory capacity will not be adequate. To expand the memory space of the 8051 micro-controller, we must bind external ROM/EPROM and RAM. We also understand that ROM serves as program memory and RAM serves as data memory. Let's take a look at how the 8051 accesses these memories.

Why need External Memory Interfacing in 8051 Microcontroller ?

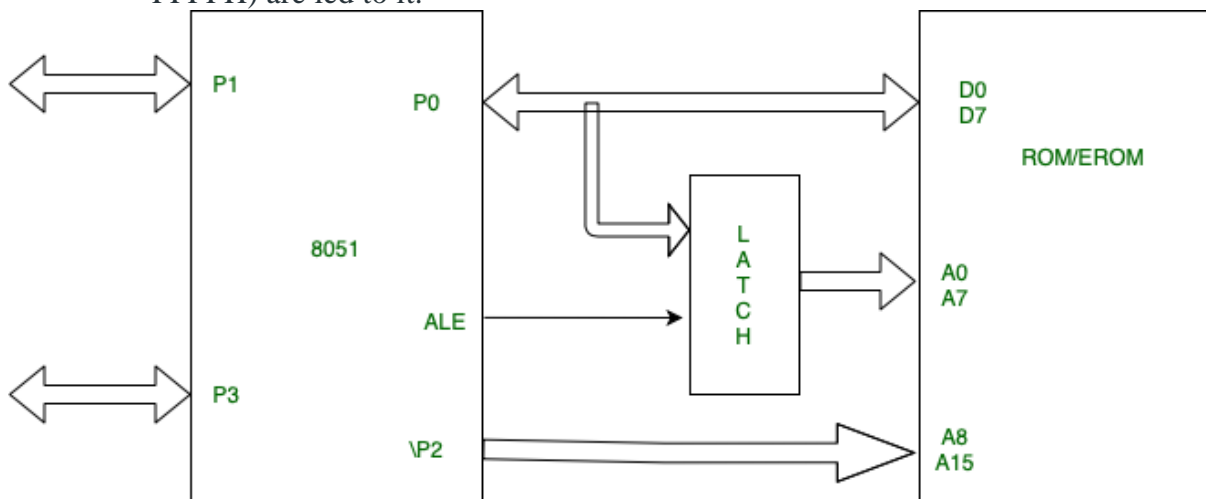
External memory interfacing is necessary in the 8051 microcontroller for several reasons:

1. Limited internal memory: The 8051 microcontroller has a limited amount of internal memory, including 128 bytes of RAM and 4KB of on-chip ROM. This memory may not be sufficient for some applications that require larger program memory or more data storage.
2. Larger programs: For applications that require larger programs, such as complex algorithms or multiple functions, external memory interfacing can provide the necessary program memory space to store these programs.
3. Data storage: Applications that require the storage of large amounts of data, such as data logging or data analysis, may require external memory interfacing to store the data.
4. Flexibility: External memory interfacing provides greater flexibility in the design of embedded systems, allowing for customization and adaptability to meet specific application requirements.
5. Cost-effective: External memory devices such as RAM and ROM are relatively inexpensive, making it cost-effective to interface them with the microcontroller instead of using more expensive on-chip memory.

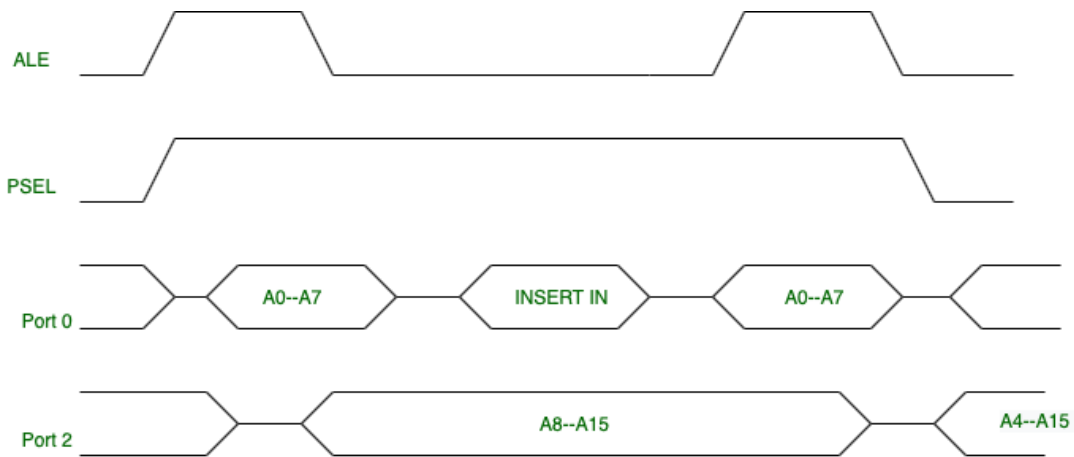
External Program Memory :



- The program fetches to addresses 0000H through 0FFFH are directed to the internal ROM in the 8051 when the EA pin is attached to Vec, and program fetches to addresses 1000H through FFFFH are directed to the external ROM/EPROM. When the EA pin is grounded, all addresses fetched by the program (0000H to FFFFH) are led to it.



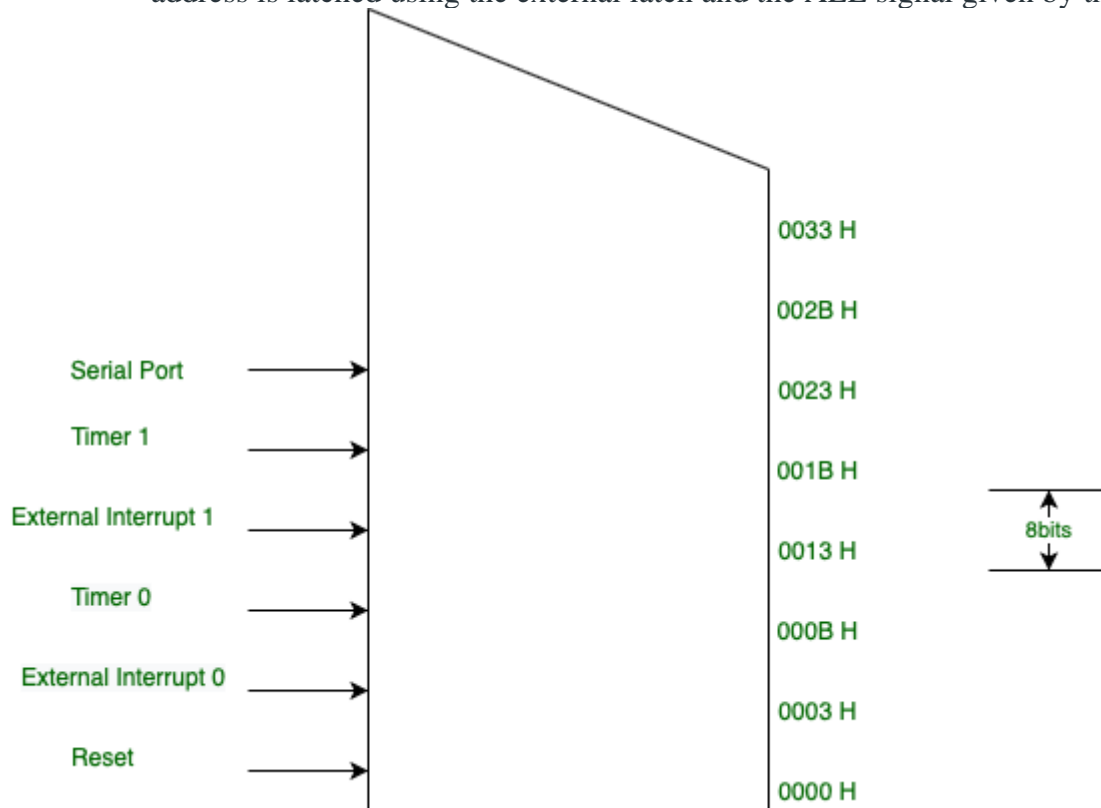
- ROM/EPROM that is external to the device. As seen in Fig. 1, the PSEN signal is used to trigger output e external ROM/EPROM.
- Port 0 is used as a multiplexed address/bus, as seen in Fig2. In the initial T-cycle, it provides a lower order 8-bit address, and later it is used as a data bus. The external latch and the ALE signal provided by the 8051 are used to latch the 8-bit address.



timing waveform for external data memory write cycle

Timing Waveform for external program memory

- Remote ROM/EPROM(Read Only Memory/Electronic Programmable ROM/Electronic Programmable ROM/Electronic Program The PSEN signal is used to activate output e external ROM/EPROM, as seen in Fig. 1.
- As seen in Fig2, port 0 is used as a multiplexed address/bus. It supplies a lower-order 8-bit address in the first T-cycle and later serves as a data bus. The 8-bit address is latched using the external latch and the ALE signal given by the 8051.



- External Interrupt 0 is 0003H, Timer 0 is 000BH, External Interrupt 1 is 0013H, Timer 1 is 001BH, and so on. If an interrupt is to be used, the operation routine for it must be in the same place as the interrupt. If the interrupt isn't used, the service location may be used as general-purpose program memory.

Instructions to Access External ROM / Program Memory :

This table is explaining the instructions to access external ROM/program memory.

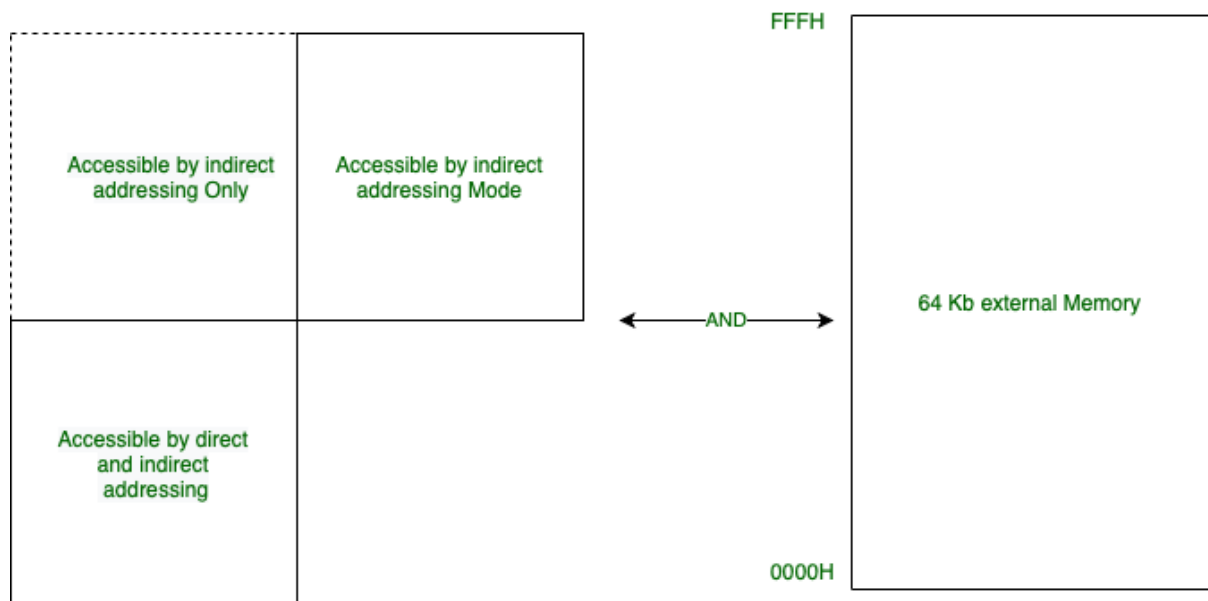
Mnemonic	Operation
MOVC A, @ A+DPTR	Copy the contents of the external ROM address formed by adding A and the DPTR, to A
MOVC A, @ A + PC	This operation will do copy This operation contents of the external ROM address formed by adding A and the PC, to A.

External Memory Interfacing :

- Up to 64 k-bytes of additional data memory can be addressed by the 8051. The external data memory is accessed using the “MOVX” instruction.
- The 8051’s internal data memory is split into three sections: Lower 128 bytes, Upper 128 bytes, and SFRs. While they are physically distinct bodies, the upper addresses and SFRs share the same block of address space, 80H by FFH.
- The upper address space is only accessible via indirect addressing, and SFRs are only accessible via direct addressing, as seen in Higher address space, on the other hand, can be reached using either direct or indirect addressing.

Internal Data Memory

External Data Memory



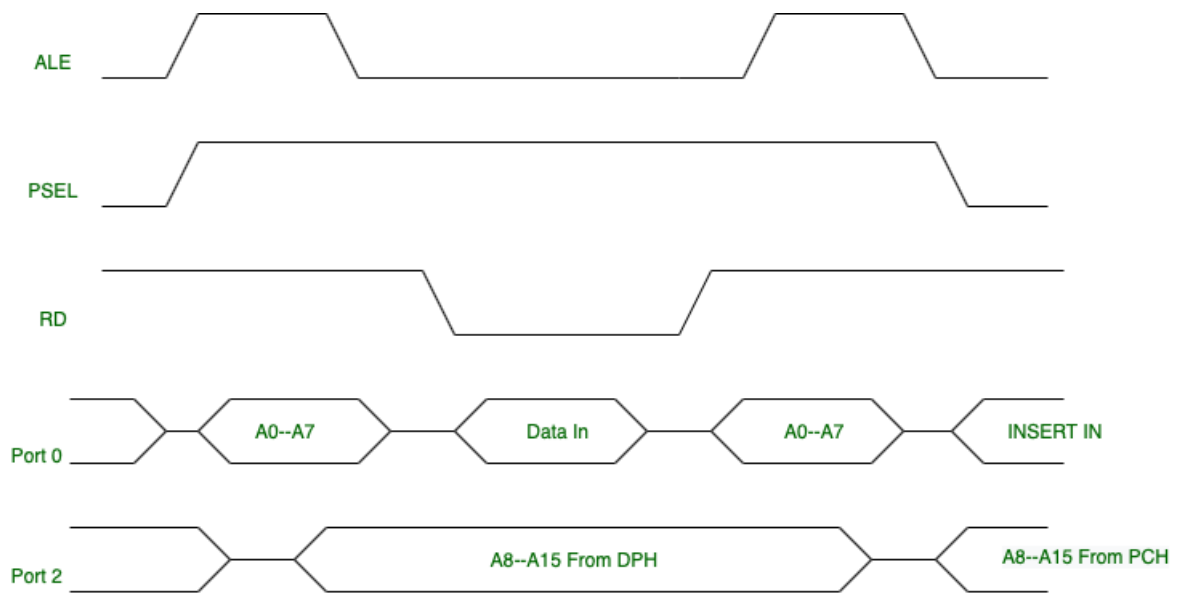
A map of the 8051 data memory Fig. 5

- Fig. 7 a and b show the timing waveform for external data memory read and write cycles, respectively.



timing waveform for external data memory read cycle

7(a)



timing waveform for external data memory write cycle

7(b)

Instructions to Access External Data Memory :
 The Table explains the instruction to access external data memory.

Mnemonic	Operation
MOVX A, @Rp	In this operation, it will copy the contents of the external address in Rp to A.
MOVX @DPTR, A	Copy the contents of the external address in DPTR to A.
MOVX @Rp, A	Copy data from A to the external address in Rp
MOVX DPTR, A	Copy data from A to the external address in DPTR.

Important Points to Remember in Accessing External Memory :

- In the Case of accessing external memory, All external data moves with external RAM or ROM involve the A register.
- While accessing external memory, R can address 256 bytes and DPTR can address 64 k-bytes
- MOV X instruction is used to access external RAM or I/O addresses