Address Decoding and Memory Mapping: Memory address decoding is nothing but to assign an address for each location in the memory chip. The data stored in the memory is accessed by specifying its address. Memory address can be decoded in two ways:

i) Absolute or Fully decoding and ii) Linear Select or Partial decoding

There are many advantages in absolute address decoding.

i) Each memory location has only one address, there is no duplication in the address
ii) Memory can be placed contiguously in the address space of the microprocessor
iii) Future expansion can be made easily without disturbing the existing circuitry

There are few disadvantages in this method

i) Extra decoders are necessary
ii) Some delay will be produced by these extra decoders.

The main advantage of linear select decoding is its simplified decoding circuit. This reduces the hardware design cost. But there are many disadvantages in this decoding.

i) Multiple addresses are provided for the same location
ii) Complete memory space of the microprocessor is not efficiently used
iii) Adding or interfacing ICs with already existing circuitry is difficult.

Absolute Address Decoding: The 8085 microprocessor has 16 address lines. Therefore it can access $2^{16}$ locations in the physical memory. If all these lines are connected to a single memory device, it will decode these 16 address lines internally and produces 216 different addresses from 0000H to FFFFH so that each location in the memory will have a unique address.

![Fig. 1.6 Memory Address](image)

Above diagram shows the various memory addresses used in Microprocessor. If more than one chips are used then some logic must be used to select one particular chip. This is done with the help of decoder.

74LS138 address decoder to generate the chip select signals for each memory block. In this
decoder when the address lines $A_{13}$, $A_{14}$ and $A_{15}$ are 000, the output line $Y_0$ will be activated as shown in Fig 1.7. This in turn selects the first memory block. Similarly when these lines are 001 ($C=0$, $B=0$ and $A=1$) $Y_1$ will be activated and the second memory block will be selected.

Fig 1.7: Memory block decoder

In this type of memory interfacing, all the address lines (A0 to A15) have been used. Each location in the memory will have a single address. This type of address decoding is called as absolute or fully decoded addressing.

Fig. 1.8: Role of CS signal
According to the value of Ao and A1, any one register will be selected and to select one memory chip we need one chip select signal CS signal as shown in the next diagram.

If CS’ is ‘0’ memory 1 will be selected else memory 2 will be selected. And the complete picture of the interfacing is shown below.

The simple view of RAM is that it is made up of registers that are made up of flip-flops (or memory elements). The number of flip-flops in a “memory register” determines the size of the memory word. ROM on the other hand uses diodes instead of the flip-flops to permanently hold the information. For the microprocessor to access (Read or Write) information in memory (RAM or ROM), it needs to do the following:

Select the right memory chip (using part of the address bus). Identify the memory location (using the rest of the address bus). Access the data (using the data bus).
**Tri-State Buffers:** An important circuit element that is used extensively in memory. This buffer is a logic circuit that has three states: Logic 0, logic1, and high impedance. When this circuit is in high impedance mode it looks as if it is disconnected from the output completely. This circuit has two inputs and one output. The first input behaves like the normal input for the circuit. The second input is an “enable”. If it is set high, the output follows the proper circuit behaviour. If it is set low, the output looks like a wire connected to nothing.

**Input /Output Devices:** Parallel Interfacing: There are two ways to interface 8085 with I/O devices in parallel data transfer mode: Memory Mapped IO and IO mapped IO.

Memory mapped I/O: It considers them like any other memory location. They are assigned a 16-bit address within the address range of the 8085. The exchange of data with these devices follows the transfer of data with memory. The user uses the same instructions used for memory.

I/O mapped I/O: It treats them separately from memory: I/O devices are assigned a “port number” within the 8-bit address range of 00H to FFH. The user in this case would access these devices using the IN and OUT instructions only.
IO mapped IO V/s Memory Mapped IO:

<table>
<thead>
<tr>
<th>Memory Mapped IO</th>
<th>IO mapped IO</th>
</tr>
</thead>
<tbody>
<tr>
<td>• IO is treated as memory.</td>
<td>• IO is treated IO.</td>
</tr>
<tr>
<td>• 16-bit addressing.</td>
<td>• 8-bit addressing.</td>
</tr>
<tr>
<td>• More Decoder Hardware.</td>
<td>• Less Decoder Hardware.</td>
</tr>
<tr>
<td>• Can address $2^{16}=64k$ locations.</td>
<td>• Can address $2^8=256$ locations.</td>
</tr>
<tr>
<td>• Less memory is available.</td>
<td>• Whole memory address space is available.</td>
</tr>
<tr>
<td>• Memory Instructions are used.</td>
<td>• Special Instructions are used like IN, OUT.</td>
</tr>
<tr>
<td>• Memory control signals are used.</td>
<td>• Special control signals are used.</td>
</tr>
<tr>
<td>• Arithmetic and logic operations can be performed on data.</td>
<td>• Arithmetic and logic operations cannot be performed on data.</td>
</tr>
<tr>
<td>• Data transfer b/w register and IO.</td>
<td>• Data transfer b/w accumulator and IO.</td>
</tr>
</tbody>
</table>