UNIT IV – INTERFACING TECHNIQUES AND PERIPHERALS

Memory and I/O Interfacing Concepts, Interfacing with LEDs, 7-segment displays, LCD , Interfacing ADC and DAC (e.g., ADC0808/0809, DAC0800) , Sensor Interfacing (Temperature, IR, Ultrasonic), Stepper Motor and DC Motor Interfacing, Serial Communication Protocols – USART, RS232 , I2C, SPI (Conceptual Introduction)

MEMORY INTERFACING

The interfacing of the 8086 microprocessor to memory is the main step in the design of a microprocessor-based system. Since 8086 has 20 address lines, a total of 1MB memory space can be interfaced. Various types of memory chips such as RAM, ROM, EPROM may be connected within this space.

It is not necessary that the address space of any microprocessor based system should start only from 00000. It may start from any boundary of 1 KB, i.e. the address bits A_0 to A_9 must be zero at the starting address.

The first task is to allocate the address range to different memory chips and devise the address decoding circuit. Based on the address line, the chip select signal to select the particular memory chip may be generated. The block diagram of address decoder 74LS138 is shown in figure.

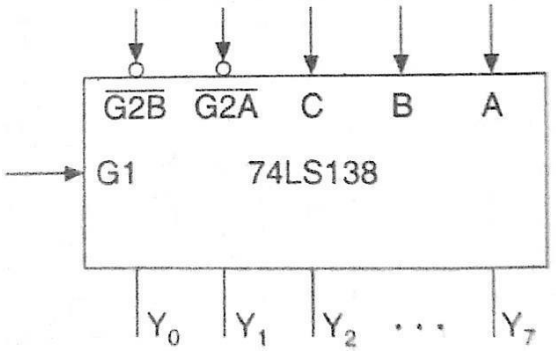

Fig: Block diagram of 74LS138

The decoder is selected when G1.($\overline{G2B}$. $\overline{G2A}$ $\overline{A}$) =1. On selection, the decoder generates the signals Y0 to Y7 based on the information available at A,B and C pins, and based on the truth table shown in figure.

Inputs			Outputs							
C	**B**	**A**	**Y0**	**Y1**	**Y2**	**Y3**	**Y4**	**Y5**	**Y6**	**Y7**
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1

0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

Fig: Truth table of 74LS138 decoder

Example 1: 2KB chip interfacing

Consider a 2KB chip 2716/6116 is to be interfaced to the 8086 starting from 00000 to 007FFH. The bits A_{19} to A_{11} in address information will always be zero. Thus, we may connect address lines to 74LS138 pins in the following manner.

$$A=A_{11},\ B=A_{12},\ C=A_{13}$$

$$\overline{G2}\ \overline{A}=A_{14},\ \overline{G2B}=A_{15}$$

$$G1 = (\overline{A}1\overline{6} \cdot \overline{A}1\overline{7} \cdot \overline{A}1\overline{8} \cdot \overline{A}1\overline{9}) \cdot (M/\overline{IO})$$

Since A,B,C are zero, Y0 will be low which can be used as a active low chip select ($\overline{CS}$) signal for the memory chip. For the next address range which starts from 00800H, A11 will become 1. Thus, the pin A of 74LS138 decoder will become 1 for the address range starting from 00800H and this will make Y1 low which can be used to select the next memory chip as shown in figure.

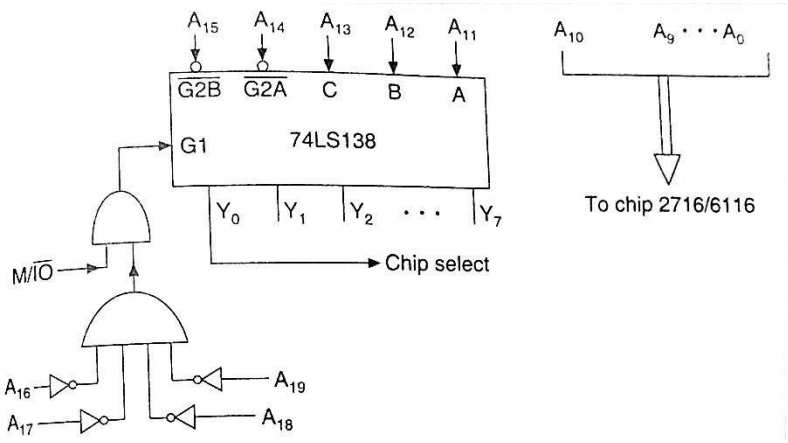

Fig:Generation of chip select signal for memory

Example 2: 4KB chip interfacing (separate decoder system)

The address range is 00000 to 00FFFH. i.e. total 4KB.

Step 1: Bit pattern of starting address and ending address are A19 – A0

00000H = 00000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

00FFFH= 00000 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bits which do not undergo change are A19 to A12. These bits maybe utilized for G1, $\overline{G2A}$ and $\overline{G2B}$.

Bits A12, A13, A14 may be connected to A,B,C input pins respectively.

Step 2: There may be different combinations for G1, $\overline{G2A}$ and $\overline{G2B}$. One possible combination is

$$\overline{G2A} = A15 \text{ OR } A16 \text{ OR } (M/\overline{IO})' = (\overline{A15} . \overline{A16} . (M/\overline{IO}))'$$

$$\overline{G2B} = A17 \text{ OR } A18 \text{ OR } A19 = (\overline{A17} . \overline{A18} . \overline{A19})'$$

For lower decoder, G1= $\overline{A0}$

For upper decoder, G1=BHE

Step 3:

A=A12, B=A13, C=A14

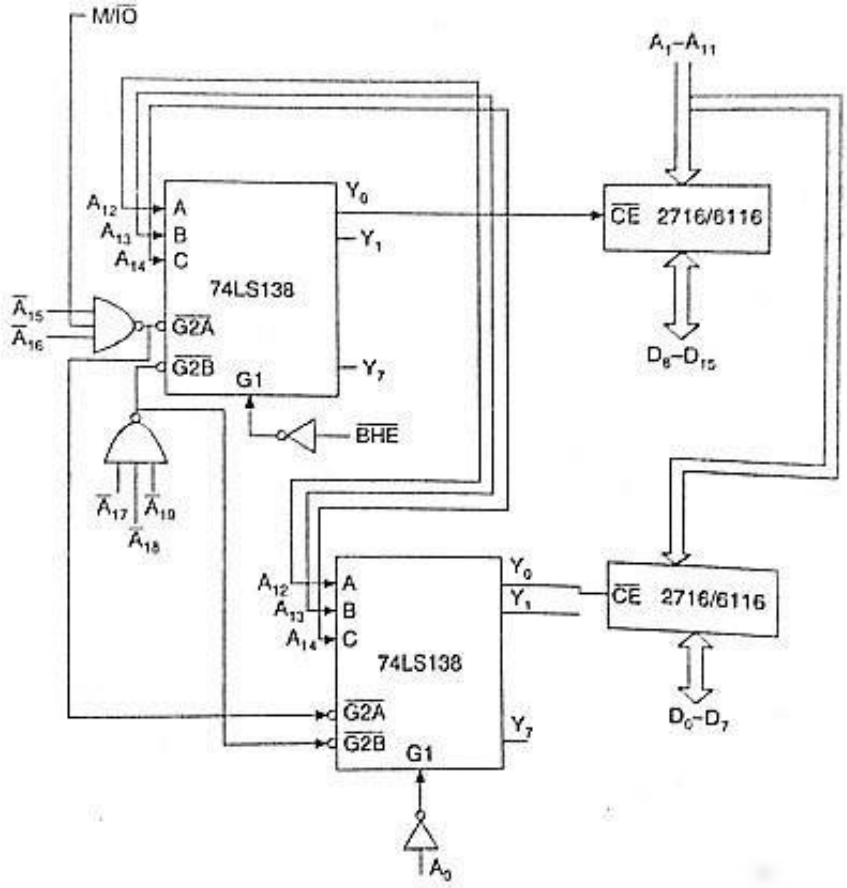

Fig : Memory address decoding (separate decoder system)

Thus the upper memory chip 2716/6116 (2KB) will be part of high-order memory bank whereas the lower memory chip 2716/6116 (2KB) will be part of low-order memory bank. Other output lines of decoders, i.e. Y1 to Y7 can be used to connect similar chips. Thus the total memory capacity of 32 KB may be interfaced in this way.

Example 3: 8KB chip interfacing

Consider another example in which the 8KB memory chip 2764/6164 is to be interfaced starting from address FA000H. Bit patterns of starting address FA000H and end address FBFFFH have to be represented.

Step 1: Bit pattern of starting address and ending address are A19 – A0

FA000H=1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

FBFFFH=1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bits which do not undergo change are A14 to A19. Thus these bits maybe utilized for G1, $\bar{G}\bar{Z}\bar{A}$ and $\bar{G}\bar{Z}\bar{B}$.

Bits A12,A13,A14 may be connected to A,B,C input pins respectively.

Step 2: There may be different combinations for G1, $\bar{G}\bar{Z}\bar{A}$ and $\bar{G}\bar{Z}\bar{B}$. One possible combination is G1=M/$\bar{I}\bar{O}$ OR A15

$$\bar{G}\bar{Z}^{-}{}^{-}\bar{A}=\bar{A}\bar{1}\bar{9}\text{OR}\bar{A}\bar{1}\bar{8}$$

$$\bar{G}\bar{Z}\bar{B}=\bar{A}\bar{1}\bar{6}\text{OR}^{-}\ \bar{A}\bar{1}\bar{7}$$

Step 3:

$$A \rightarrow A12\,,B \rightarrow A13\,,C \rightarrow A14$$

Memory can be organized into two different memory chip for higher order address bank and lower order address bank. This may be done in two ways:

(a) Separate decoder for high- and low- order banks.

(b) Single decoder for both the banks.

I/O INTERFACING

There are two methods of interfacing I/O devices with 8086. They are:

1) I/O mapped I/O
2) Memory mapped I/O

1) I/O mapped I/O:

It is also called isolated I/O scheme i.e. the I/O locations are isolated from the memory system in a separate I/O address space. The isolated I/O and memory mapped I/O address space for the 8086 are shown below.

Address Address Address

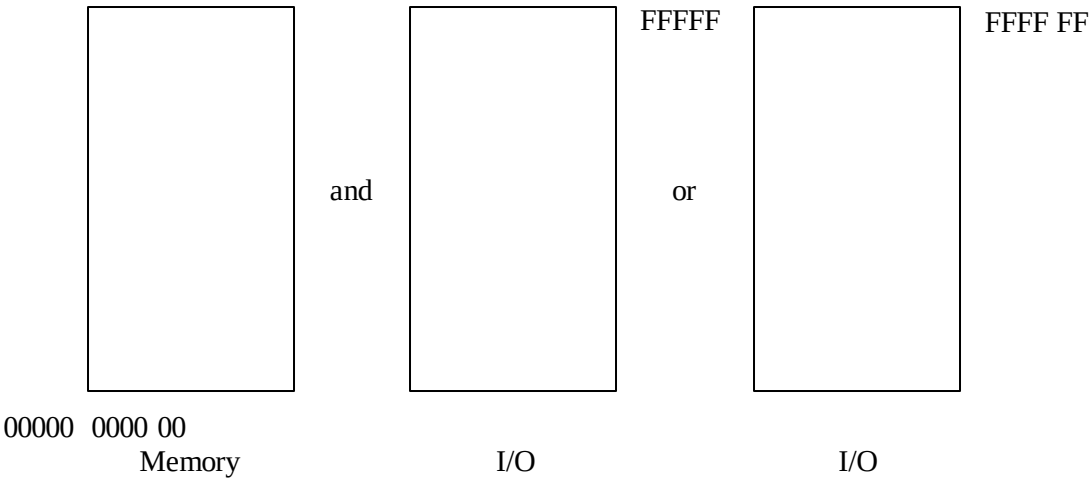

Fig : I/O mapped I/O

In the isolated I/O scheme, the address for the isolated I/O devices called ports is separate from the memory. Hence the user can expand the full size to (1MB) without using any of its address space for the I/O devices.

The **disadvantages** of this scheme is that the data is transferred between the 8086 and the I/O devices only by the IN and OUT instructions.

2) Memory mapped I/O:

The memory mapped I/O scheme does not use IN and OUT instructions. Instructions that transfer data between microprocessor and the memory are used to transfer data between the 8086 and I/O devices.

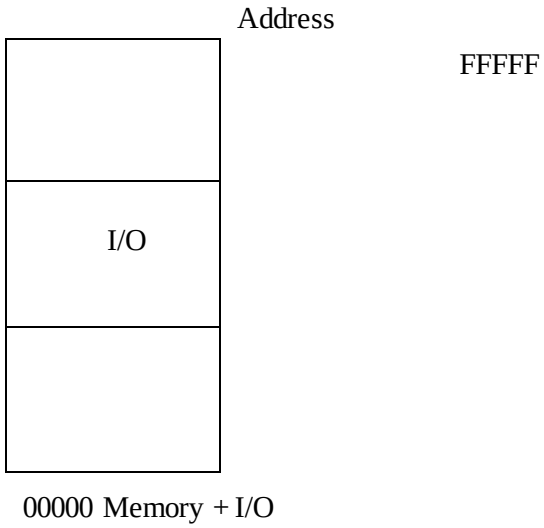

Fig: Memory and I/O maps for the 8086

The **advantage** of this scheme is that there are memory transfer instructions in the 8086 and all of them can be used to access the I/O device. The same control signals used for accessing the memory ($\overline{M}$ $\overline{R}$ $\overline{D}$ $\overline{C}$ and $\overline{M}$ $\overline{W}$ $\overline{T}$ $\overline{C}$) in the maximum

mode are used to access the I/O devices. This reduces the additional hardware required to generate the control signals.

The main **disadvantage** of this scheme is that a portion of the memory systems is used as the I/O map. This reduces the amount of memory available.

Fig: Asynchronous mode transmit and receive format

IO mapped IO V/s Memory Mapped IO

IO Mapped IO	Memory Mapped IO
IO is treated IO.	IO is treated as memory.
8-bit addressing.	16-bit addressing.
Less Hardware.	More Hardware.
Can address 28=256 locations.	Can address 216=64k locations.
Whole memory address space is available.	Less memory is available.