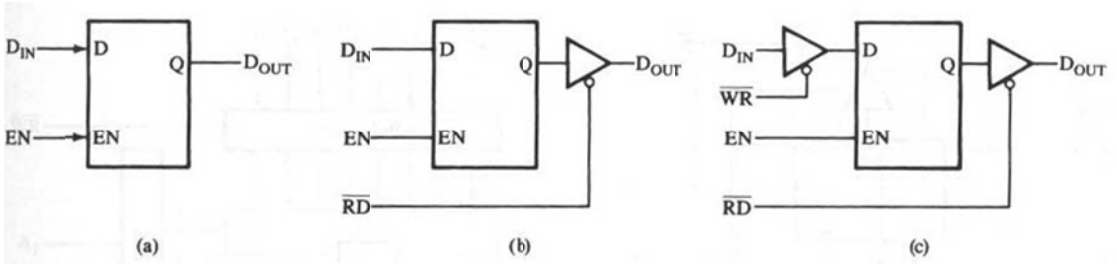
2.4 Memory

Memory is and essential component of microcomputer system; it stores binary instructions and data for the microprocessor. They can be classified in two groups: prime (or main) memory and storage memory. The R/W memory is made up of registers, and each register can use this memory to hold programs and store data. On the other hand, the ROM stores information permanently in the form of diodes.

2.4.1 Flip-Flop or Latch as a Storage Element.

What is memory?

It is a circuit that can store bits. A flip-flop or latch is a basic element of memory. To write or store a bit in the latch, we need an input data bit and an enable signal.



This latch, which can store one binary bit, is called a memory cell.

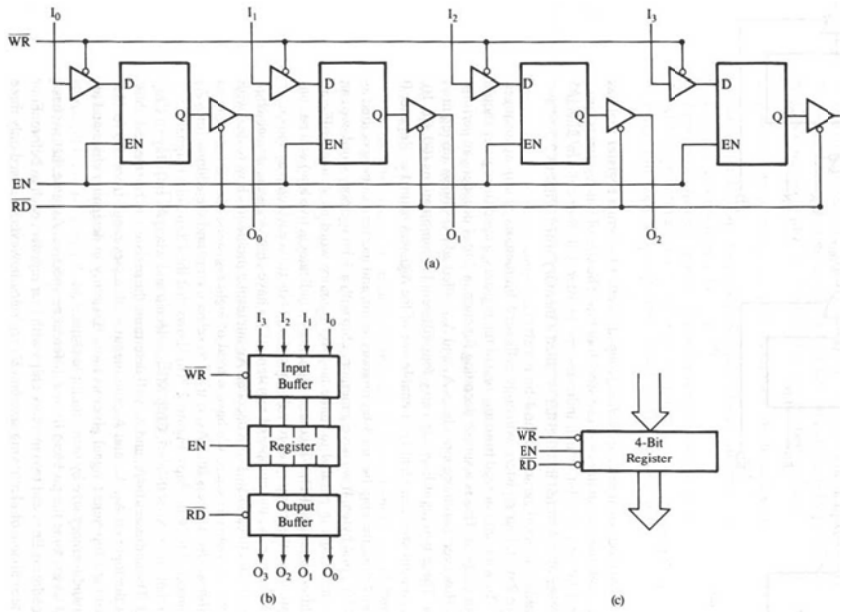


Figure shows four latches as a 4 bit Register. The number of bits stored in a register is called a memory word.

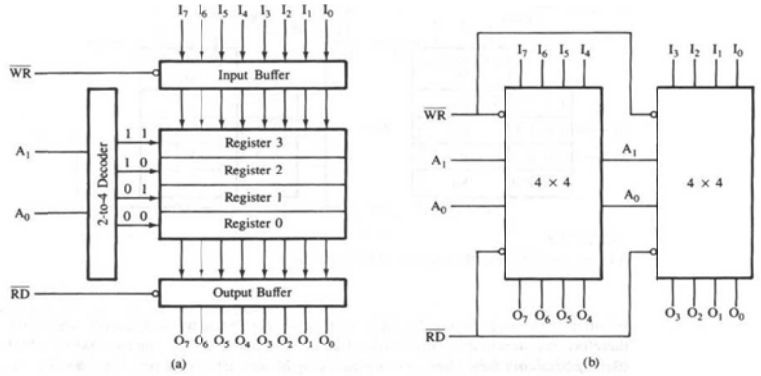
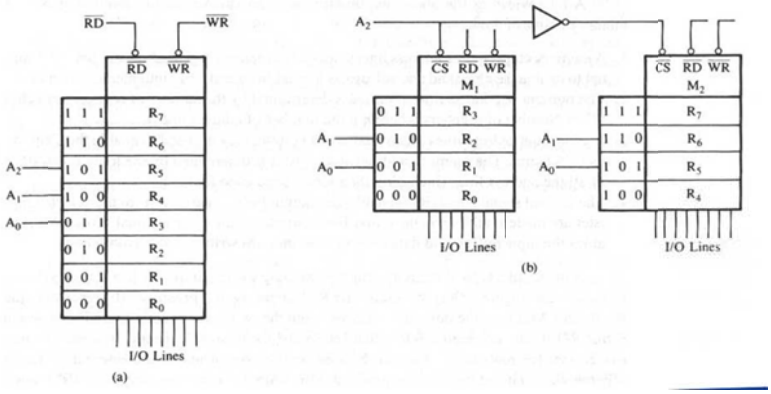


FIGURE
(a) 4 × 8 Bit Register; (b) Two 4 × 4 Bit Registers



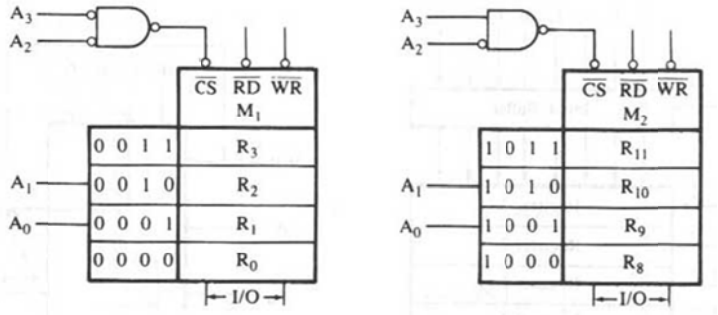
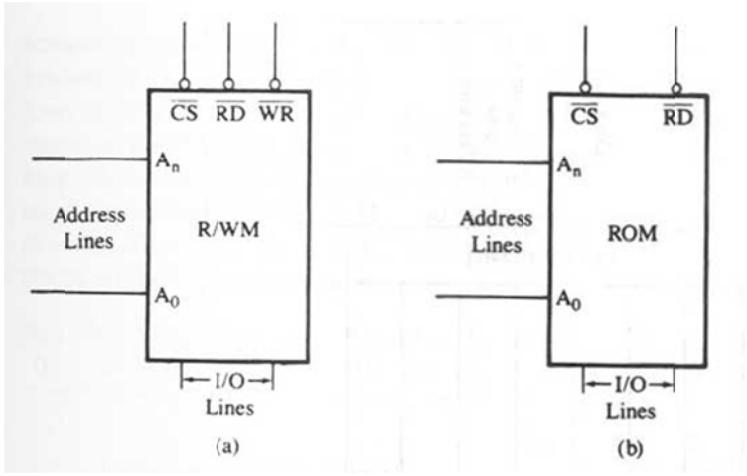


FIGURE
Addressing Eight Registers with Four Address Lines

The requirements of a memory chip.

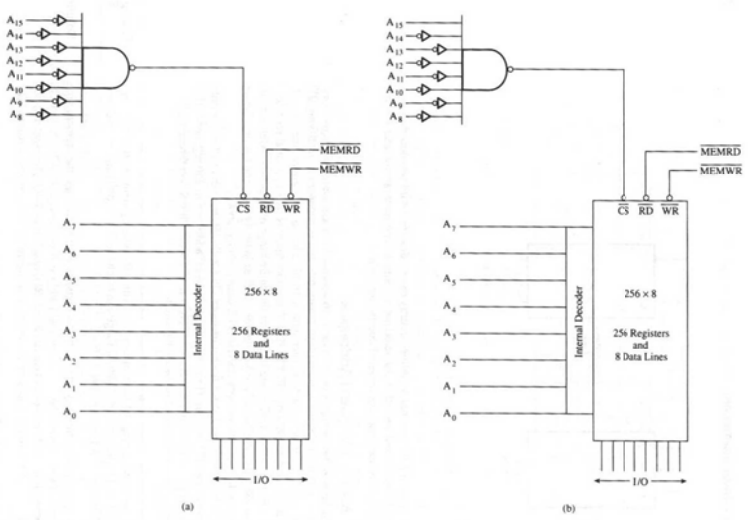
1. A memory chip requires address lines to identify a memory register, a chip select CS signal.
2. The number of address lines required is determined by the number of registers in a chip (2^n)
3. If additional address lines are available in a system, chip select signal is used.

4. The control signal Read ($\overline{RD}$) enables the output buffer. The control signal Write ($\overline{WR}$) enables the input buffer.



2.4.2 Memory Map and Address

Typically, in an 8- bit microprocessor system, 16 address lines are available for memory. It is capable of identifying 2^{16} (65,536) memory registers, each register with a 16 bit address. The entire memory addresses can range from 0000 to FFFF in Hex.



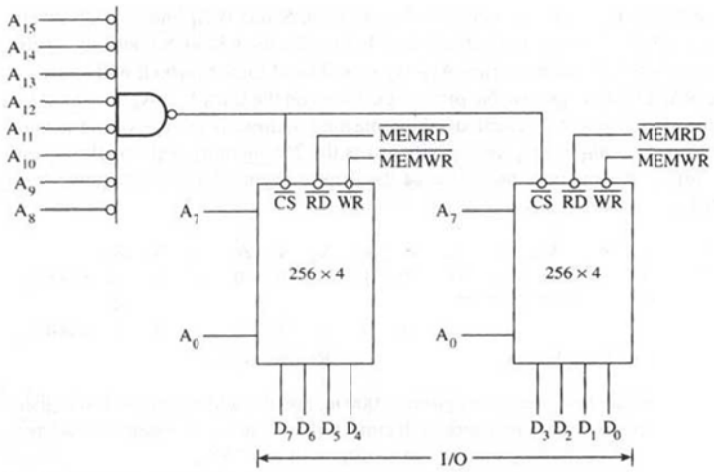


FIGURE
Memory Circuit Using 256 × 4 Memory Chips

2.4.3 How the MPU writes into and Reads from Memory.

1. Places the 16 bit address on the address of the memory location where a byte is to be stored. The interfacing logic of the memory chip decodes the address and selects the memory register to be written into.
2. Places the byte on the data bus.
3. Send the control signal Memory. Write to enable the input buffer of the memory and then stores the byte.

To read from memory the step are similar to that of writing into memory, except the order of step 2 and 3.

2. The MPU send the control signal Memory Read to enable the output buffer of the memory chip.
3. The memory chip places the data byte on the data bus, and the MPU reads the data byte

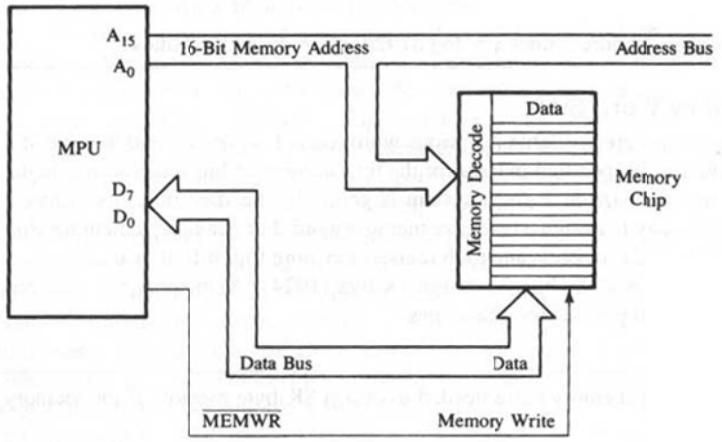


FIGURE
Memory Write Operation

2.5 Input and Output (I/O) Devices

Input/output devices are the means through which the MPU communications with "the outside world". The MPU accepts binary data as input from devices such as keyboards and A/D converters and sends data to output devices such as LEDs or printers.

There are two different methods by which an MPU can identify I/O devices:

- 8 bit address
- 16 bit address

2.5.1 I/Os with 8-bit Addresses (Peripheral-Mapped I/O)

The MPU uses eight address lines to identify an input or an output device. The MPU can identify 256 input and 256 output devices with addresses ranging from 00_H to FF_H.

The steps in communicating with I/O device can be summarized as follows:

1. The MPU places an 8 bit address on the address bus, which is decodes by the external decode logic.
2. The MPU sends a control signal (I/O Read or I/O Write) to enable the I/O device.
3. Data are transferred on the data bus.

2.5.2 I/Os with 16 bit Addresses (Memory-Mapped I/O)

The MPU uses 16 address lines to identify an I/O device; an I/O is connected as if it is a memory register. In memory-mapped I/O, the MPU uses the same control signals

(Memory Read or Memory Write) and instructions as there of memory and follows the same steps as when it is accessing a memory register.

2.5.3 How does the system work?

In figure above let us assume that the memory address of the first R/W memory location is 8000H and the address of the fan output port is 43H.

The example of the two byte instruction that tells the processor to turn on the fan with the address 43H.

Memory	Hex Code	Code explanation
8000	D3	Out put (turn on)
8001	43	The fan

To execute this one instruction, the MPU performs the following operations.

1. MPU must read the binary code of the out instruction stored in location 8000H. To read the code D3, the MPU places the address 8000H on the address bus, enables the memory chip by asserting the Memory Read ($\overline{\text{MEMRD}}$) signal, and fetches the code D3H using the data bus. This is called the fetch operation.
2. The code D3H in the instruction decoder and it is interpreted as a two-byte instruction. The MPU places the next address 8001H on the address bus, asserts the $\overline{\text{MEMRD}}$ signal, and fetches the address 43H. Now it has read the complete instruction, and is ready to execute the instruction.
3. The MPU places the address of the fan output port (43H) on the address bus, sends a byte to turn on the fan using the data bus, and asserts the IOWR signal to enable the output port.