Lecture-4

MEMORY:

It is a storage device. It stores program instructions, data and the results. There are two kind of memories; semiconductor memories & magnetic memories. Semiconductor memories are faster, smaller, and lighter and consume less power. Semiconductor memories are used as the main memory of a computer. Magnetic memories are slow but they are cheaper than semiconductor memories. Magnetic memories are used as the secondary memories of a computer for bulk storage of data and information's. With the development in technology, semiconductor memories are used everywhere.

If a memory stores N- words of information each word being of m bits, we say it is a N x m memory. E.g. 8x4 memory means there are 8 words and each word containing 4- bit of information (called nibble). 8 words are stored at 8-memory locations and these memory locations are clearly identified by 8 unique addresses.

Table: Formulation of Memory Address

A_2	A_1	A_0	Decimal Equivalent	Memory Location	Contents of the memory location
0	0	0	0	0	M(0)
0	0	1	1	1	M(1)
0	1	0	2	2	M(2)
0	1	1	3	2	M(3)
1	0	0	4	3	M(4)
1	0	1	5	5	M(5)
1	1	0	6	6	M(6)
1	1	1	7	7	M(7)

Addresses are formulated by bit combination available in wires known as address lines. To identify 8-memory locations, 3 address lines designated $A_2A_1A_0$ are required. The memory locations identification and the corresponding contents stored are shown in table. M(0) is the content of memory location '0' and it has 4 bits here. M(1) is the content of memory location '1' and so on. It can also be represented as shown in fig.2.1.

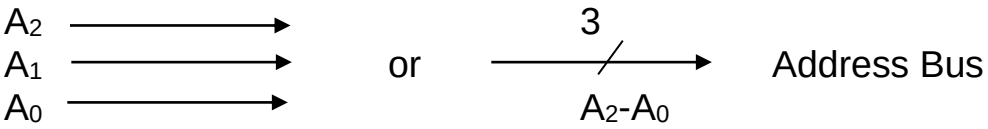

Fig.2.1 Representation of Address Bus

The three address lines A_2 A_1 A_0 together is known as address bus. It is a unidirectional bus. The microprocessor always sends the addresses.

In general, an Nxm memory shall have 'k' address lines designated A_{k-1} A_{k-2} A_{k-3} …… $A_2A_1A_0$ such that 'k' is the smallest integer satisfying the inequality $2^k \geq N$. e.g. 200x8 memory shall have 200 memory locations. Each location contains 8 bit of information. To identify 200 memory locations we require a minimum of 8 (=k) lines designated A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0. However k = 8 address lines can identify a total of 256 memory location starting from $(0000\ 0000)_2$ to $(1111\ 1111)_2$ or 00_H to FF_H. But we are using only 200 memory locations and rests of the locations are redundant. The 200 memory locations shall be identified starting from $(00000000)_2$ to $(11000111)_2$. The other combinations $(11001000)_2$ to $(11111111)_2$ are not used in this memory & are redundant addresses. Since it is too tiring & boring

to use binary numbers for identifying the addresses we normally make use of hexadecimal number notation. E.g. 200 memory locations are identified starting from 00_H to $C7_H$ and $C8_H$ to FF_H are redundant memory locations. Using 10 address lines designated as $A_9A_8A_7$……..$A_2A_1A_0$, one can directly address 2^{10} = 1024 memory locations. This is known as 1k memory locations.

The capacity of a memory is specified terms of the maximum number of words the memory can store. In general, if the memory has k-bit address and each word is of length m, then the memory has a capacity of 2^k x m bits, organized as 2^k words each of m-bits. If k=10, then the memory can store 1024 words or 1k words. Intel 808A microprocessor has 16-address lines. Therefore it can address directly 2^{16} memory locations.

2^{16} memory locations = 2^8 x 2^{10} memory locations

= 64k memory locations

= 65536 memory locations.

Thus, 8-bit microprocessor provides a maximum of 2^{16} or 64k memory addresses ranging from 0000 to $FFFF_H$.

Development of Memory:

Let us see how semiconductor memories are developed. The smallest unit of information a digital system can store is a binary digit which has a logic value of ‘0’ or ‘1’. A bit of data is stored in an electronic device called a flip–flop or a 1-bit register. A flip – flop is a general memory and has two stable states in which it can remain indefinitely as long as the operating power is not interrupted. The

output can be changed only if the input signals allow for it. A very simple type of flip–flop is D- type flip-flop as shown in fig.2.2.

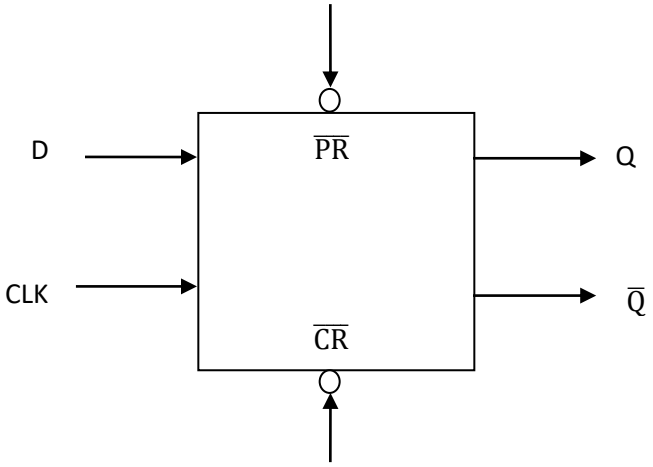

Fig.2.2 D Flip Flop

It has a single data input D and two outputs, Q and $\overline{Q}$. Output Q represents the state of flip- flop; $\overline{Q}$ represents the complement of the flip-flop's state. The logic value at a flip-flop's D input when a clock signal (CLK) occurs is stored in the flip- flop. If the stored value is equal to 1 (Q = 1) the flip–flop is set. If the stored value is equal to 0 (Q = 0) the flip–flop is clear.

The logical operation of a D type flip–flop is expressed by the characteristic equation $Q_{n+1} = D_n$. This equation indicates that the output of a D- type flip – flop, after the occurrence of a clock pulse, Q_{n+1} is equal to the logic value of the D–input before the occurrence of the clock pulse D_n. But D–type flip-flop differs with regard to the precise time at which the clock pulse causes the input data to be accepted, the output to change in accordance with the input, and the output to be held or latched.

Two clock pulse or strobes are shown in fig.2.3

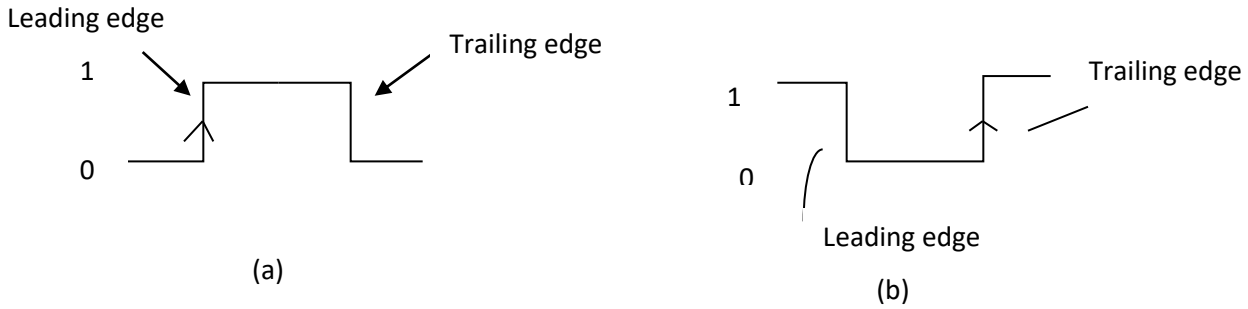

Fig.2.3 Rising Edge Active Clock Pulses

Positive clock pulse: This signal is logic 0 in its quiescent state, makes a transition to logic 1 remains at logic 1 momentarily, and then returns to logic 0. The leading edge of the pulse is a 0 to 1 or positive transition and the trailing edge is a 1 to 0 or negative transition.

Negative clock pulse: The quiescent value of this signal is logic 1 and it makes a momentary negative transition to logic 0 followed by a positive transition back to logic 1. A positive transition is also referred to as a rising edge, and a negative transition is also referred as the falling edge.

An edge triggered D–type flip-flop latches the logic value at the D input during the clock pulse's transition from one logic value to the other. The sensitivity of the flip-flop to the transition (edge) of the clock is indicated on the flip-flop logic symbol by a dynamic indicator, a triangle '>' at the clock input. Positive edge triggered flip–flops latch on the positive transition of the clock. Negative edge triggered Flip-flops latch on the negative transition of the clock.

If the clock pulse of fig (a) is applied to the positive edge pulse triggered flip-flop, the data is latched at the leading edge of the pulse. If the clock pulse of fig (b) is applied to a positive edge triggered flip-

flop, the data is latched at the trailing edge of the pulse. This is shown in fig.2.4. Note that in the edge triggered flip-flop, the input is accepted, and the output changes and is latched during a single clock transition.

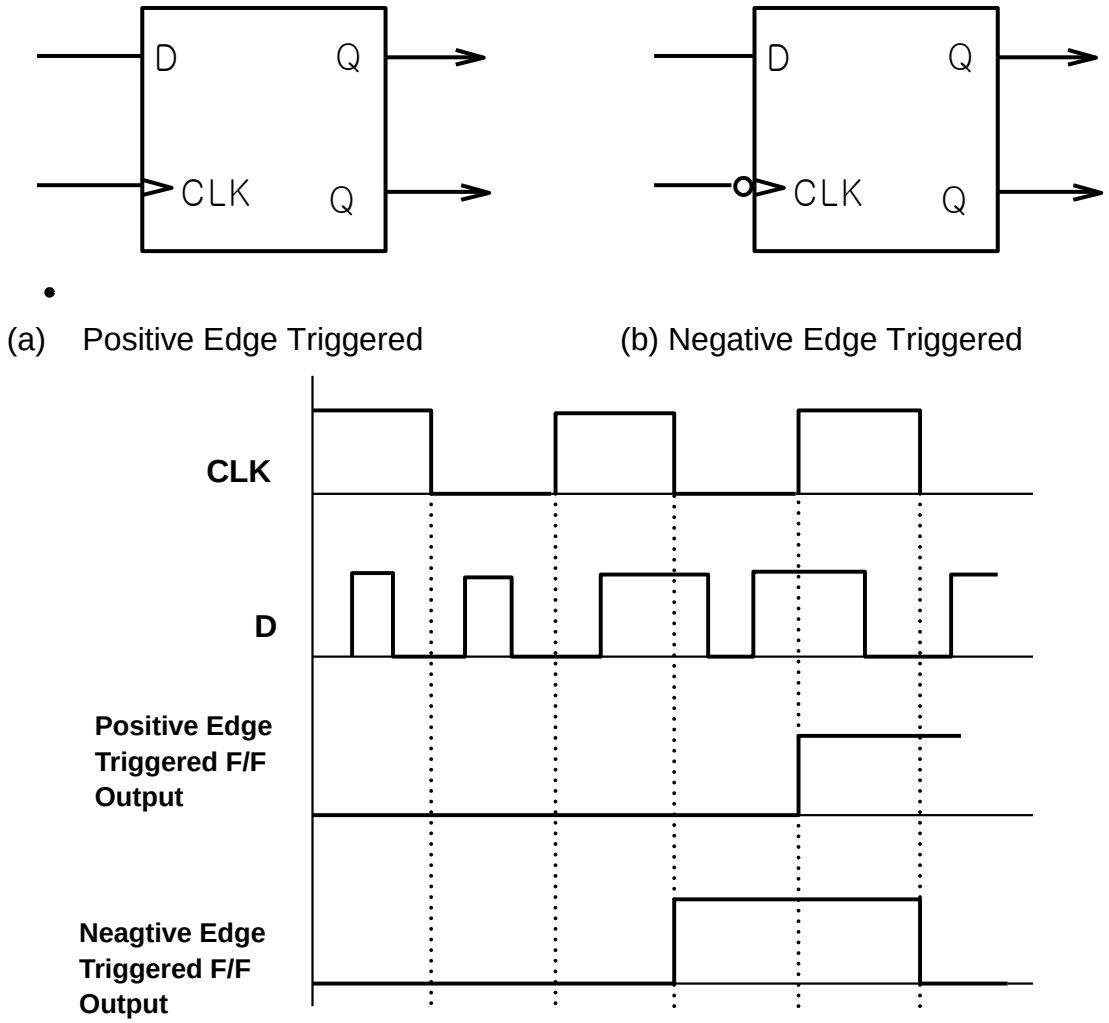

Fig.2.4 Edge Triggered D Flip-Flops and Outputs

A level triggered flip-flop usually referred to simply as a latch. It has a clock input that is sensitive to the level of the clock signal. The output of a positive level triggered D-flip-flop follows the D input when the clock signal is high. When the edge makes a transition from 1 to

0, the data present at the D input is latched. The output of a negative level triggered D flip-flop follows the input when the clock is logic ‘0’ and latches the input on a 0 to 1 transition. Thus for a level triggered flip-flop, the output follows the input, when the clock is at the trigger level. This is shown in fig.2.5. During this condition the flip-flop is referred to as being transparent. The input data is latched on the transition from the trigger level to the quiescent level.

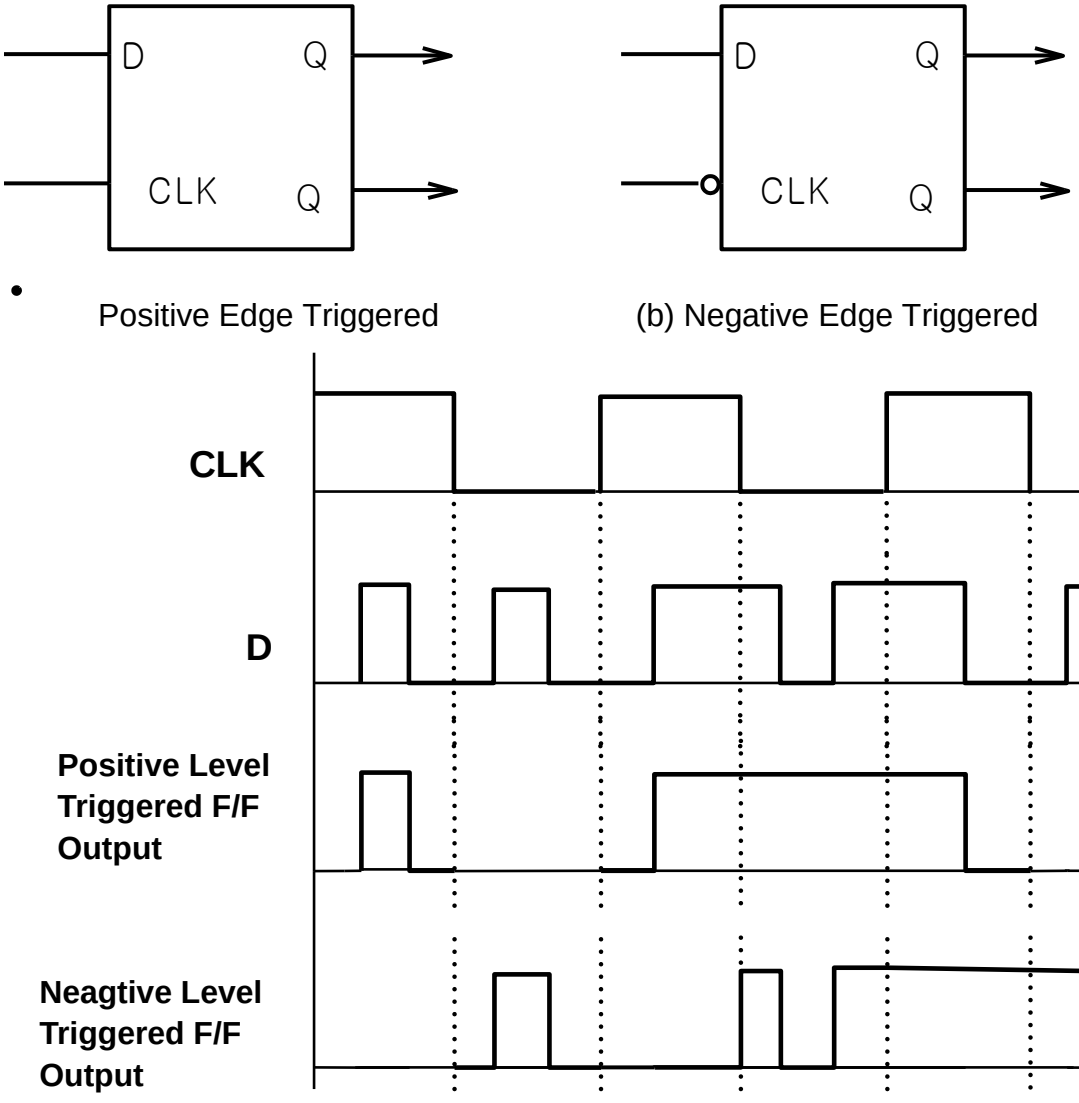

Fig.2.5 Level Triggered D Flip-Flops and Outputs

The D flip-flop shows two additional inputs common to most ICs $\overline{PR}$ (Preset) & $\overline{CR}$ (Clear). Both the inputs are active low signals. Preset & clear are asynchronous input; they affect the state of the flip-flop independent of the clock's level or transition. Thus preset & clear have override influence on clock & synchronous input. Logic '0' at the clear input clears it for proper operation. Preset & clear inputs are not strobed simultaneously.

Examples of latches:

Typical examples of transparent latches are 74LS373 & the Intel 8282 shown in fig.2.6(a) & (b). Both are functionally similar; however, they are not pin compatible. These octal latches are suitable to latch 8-bit data.

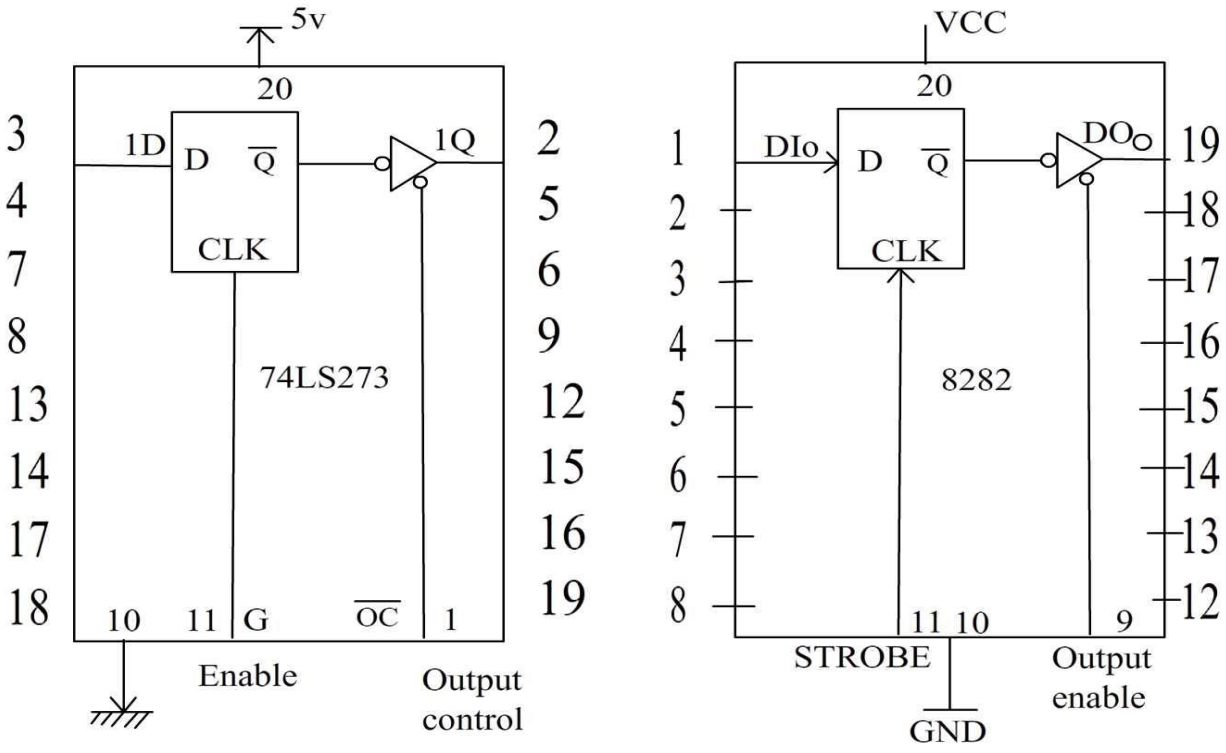

Fig.2.6 Schematic Diagram of (a) 74LS273 Latch (b) Intel 8282 Latch

Function table:

Output control	Enable G	Input	Output
L	H	H	H
L	H	L	L
L	L	X	Q
H	X	X	High-Z

Output enable	STB	Input DI	Output
L	H	H	H
L	H	L	L
L	L	X	Data latch
H	X	X	L

These devices include eight D latches with tri-state buffers. They require two input signals; enable (G) & the output control $\overline{OE}$ for the 74LS373 which are synchronous to the strobe (STB) & the output control $\overline{OE}$ for the 8282. The enable is an active high signal connected to the clock signal input of the flip-flop. When this signal goes low, data are latched from the data bus. When the output control is low (active) the data latched is accessible to the display devices.