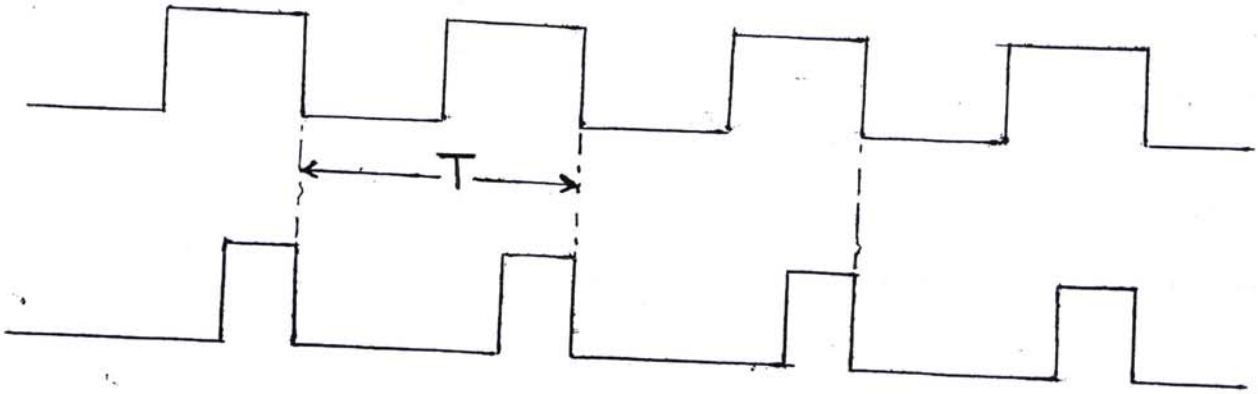
Sequential Systems

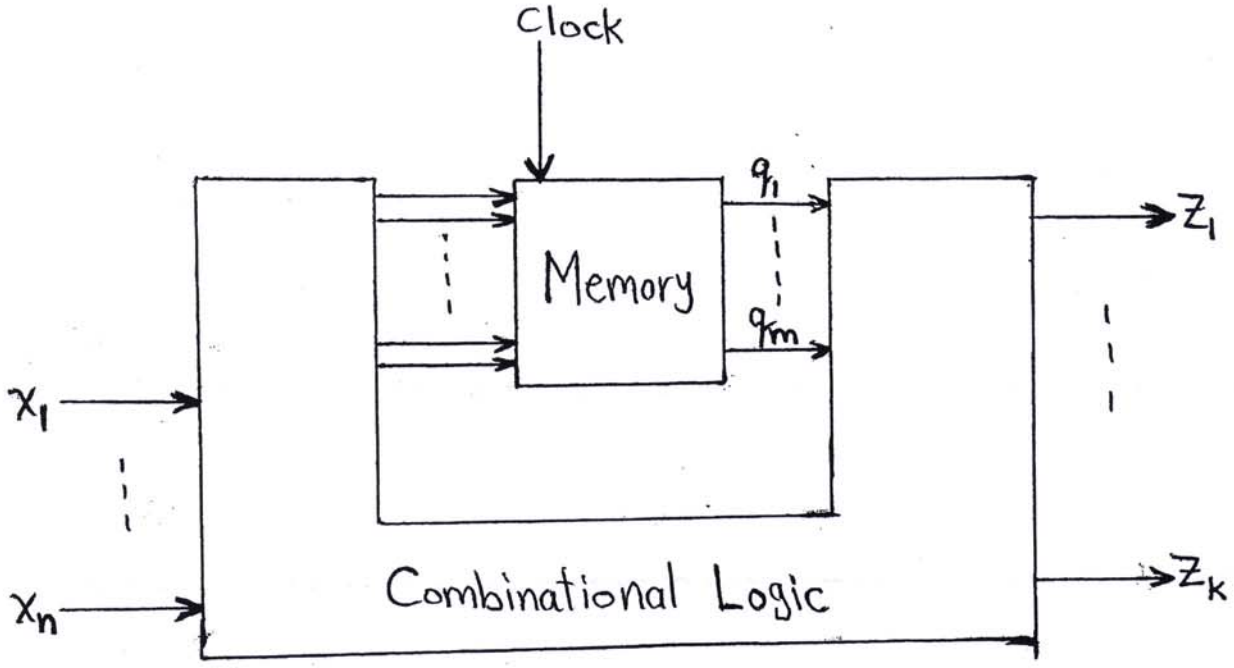
Up to now everything has been combinational, that is the output at any instant of time depends only on what the inputs are at that time. While in sequential systems the output will depend not only on the present input but also on the past history, what has happened earlier.

We will deal almost exclusively with clocked systems (sometimes referred to as synchronous). A clock is just a signal that alternates (over time) between 0 and 1 at a regular rate. The first clock signal is 0 half of the time and 1 half of the time. The second, it is 1 for a shorter part of the cycle.



The period of the signal (T) is the length of one cycle. The frequency is the inverse (1/T). A frequency of 200 M Hz corresponds to a period of 5 n sec.

The following block diagram is a conceptual view of a synchronous sequential system. A sequential system consists of a set of memory devices and some combinational logic. This diagram depicts a system with n inputs (x's), in addition to the clock, K outputs (Z's), and m binary storage devices (q's). Each memory device may need one or two input signals. Many systems have only one input and one output.



Some new terminology to allow you to more easily describe sequential systems.

State: What is stored in memory. It is stored in binary devices.

State table: Shows for each input combination and each state, what the output is and what the

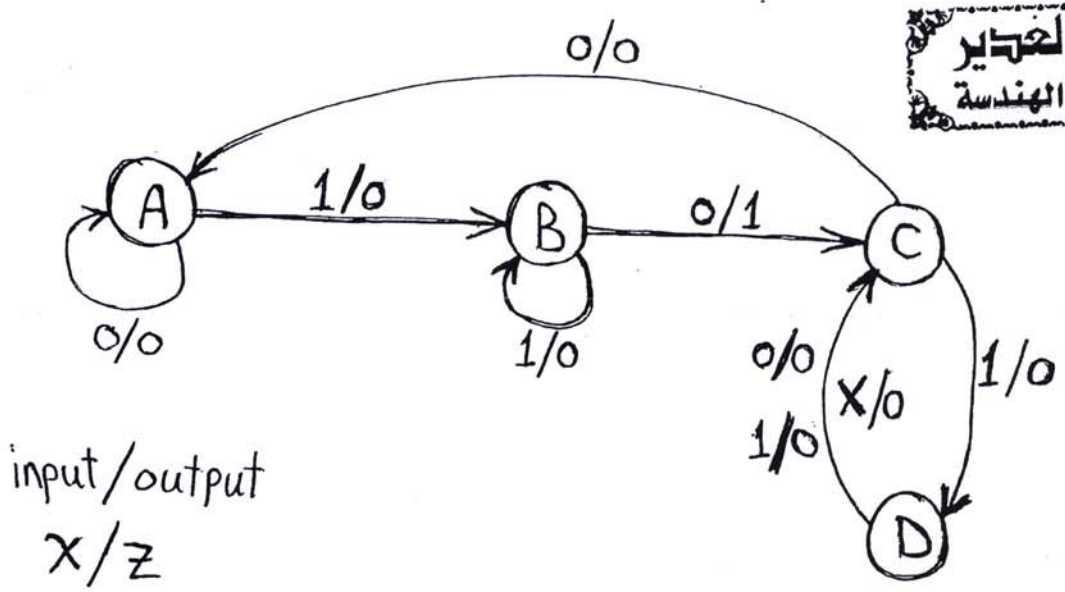
next state is, that is, what is to be stored in memory after the next clock.

State diagram: a graphical representation of the state table.

example:

Present state q	next State q^* X=0	X=1	output Z X=0	X=1
A	A	B	0	0
B	C	B	1	0
C	A	D	0	0
D	C	C	0	0

"State table"



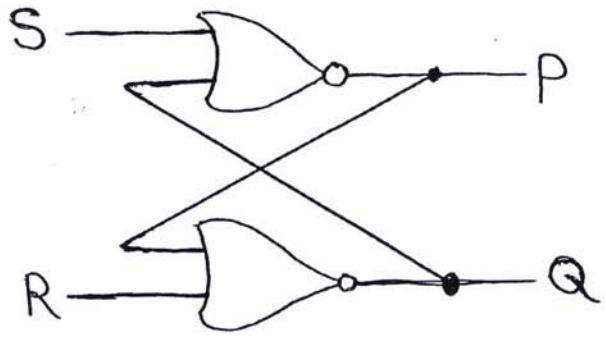
"State diagram"

Latches

A latch is a binary storage device, composed of two or more gates, with feedback, that is, for the simplest two-gate latch, the output of each gate is connected to the input of the other gate.

$$P = \overline{(S + Q)}$$

$$Q = \overline{(R + P)}$$



The normal storage state is both inputs 0 (inactive).

If S and R are 0, then

$$P = \bar{Q} \quad , \quad Q = \bar{P}$$

The latch can store either 0 or a 1. Thus, the P output is usually just labeled $\bar{Q}$; the letter S is used to indicate set, that is, store 1 in the latch.

If $S = 1$ and $R = 0$, then

$$P = \overline{(1 + Q)} = \bar{1} = 0$$

$$Q = \overline{(0 + 0)} = \bar{0} = 1$$

Thus, a 1 is stored in the latch (on line Q). Similarly, if the reset line, R, is made 1 and $S = 0$,

$$Q = \overline{(1 + P)} = \bar{1} = 0$$

$$P = \overline{(0 + 0)} = \bar{0} = 1$$

Finally, the latch is not operated with both S and R active, since, if S=1 and R=1,

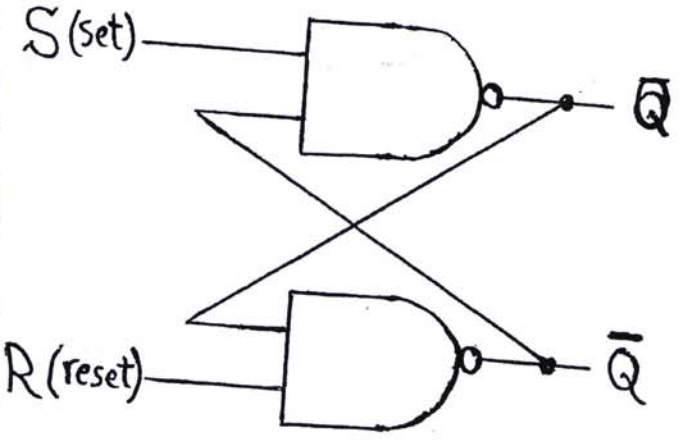
$$P = \overline{(1+Q)} = \bar{1} = 0$$

$$Q = \overline{(1+P)} = \bar{1} = 0$$

Both outputs would be **0** (not the complement of each other).

S	R	Q	$\bar{Q}$	
1	0	1	0	
0	0	1	0	(after S=1, R=0)
0	1	0	1	
0	0	0	1	(after S=0, R=1)
1	1	0	0	

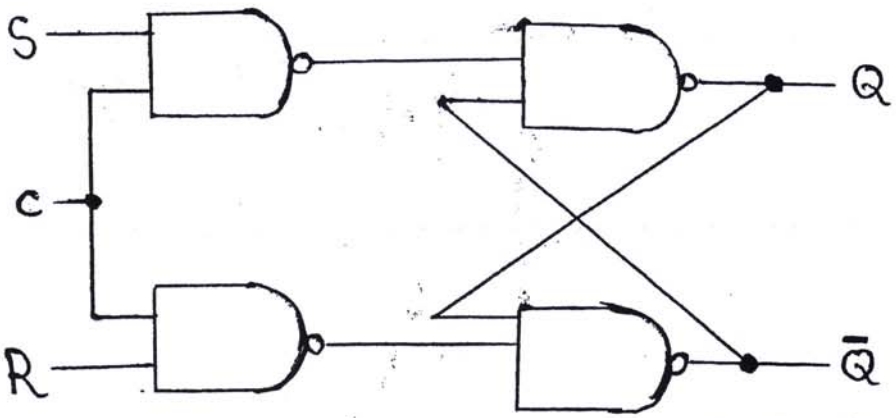
The SR latch with two cross-coupled NAND gates is shown below. It operates with both inputs normally at 1 unless the state of the latch has to be changed.



S	R	Q	$\bar{Q}$	
1	0	0	1	
1	1	0	1	(after S=1, R=0)
0	1	1	0	
1	1	1	0	(after S=0, R=1)
0	0	1	1	

Comparing the NAND with NOR latch note that the input signals for the NAND latch requires the complement of those values used for the NOR latch. NAND latch is sometimes referred to as an $\overline{S}$-$\overline{R}$ latch.

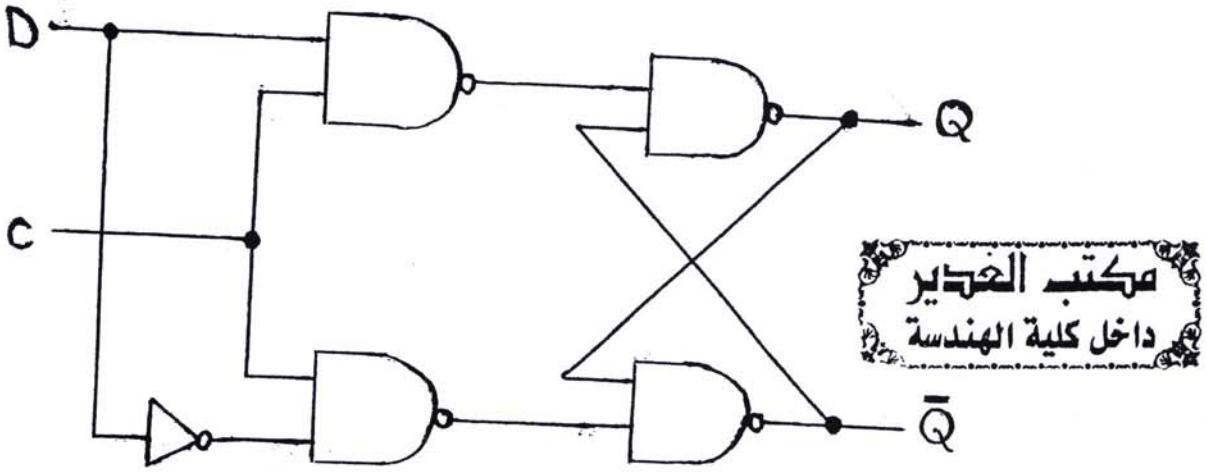
The operation of the basic SR latch can be modified by providing an additional control input that determines when the state of the latch can be changed. The control input C acts as an enable signal for the other two inputs. The output of the NAND gates stay at the logic 1 level as long as the control input remains at 0.



C	S	R	Next state of $\overline{Q}$
0	X	X	No change
1	0	0	No change
1	0	1	Q=0; Reset state
1	1	0	Q=1; Set state
1	1	1	Indeterminate

D Latch

One way to eliminate the undesirable condition of the indeterminate state in the SR latch is to ensure that inputs S and R are never equal to 1 at the same time. This is done in the D latch shown below. This latch has only two inputs: D (data) and C (control). The D input goes directly to the S input and its complement is applied to the R input.



The D latch receives the designation from its ability to hold data in its internal storage. It is suited for use as a temporary storage for binary information between a unit and its environment.

C	D	Next state of Q
0	X	No change
1	0	Q=0; Reset state
1	1	Q=1; Set state

"function table"

The binary information present at the data input of the D latch is transferred to the Q output when the control input is enabled. When the control input is disabled,

the binary information that was present at the data input at the time the transition occured is retained at the Q output until the control input is enabled again.

Flip-Flops

A flip-flop is a clocked binary storage device, that is, a device that stores either a 0 or a 1. The state of the system (that is, what is in memory) changes on the transition of the clock. For some flip flops, that change takes place when the clock goes from 1 to 0; that is referred to as trailing-edge triggered. For others, that changes takes place when the clock goes from 0 to 1; that is referred to as leading-edge triggered.

trailing

leading

Flip flops have one or two outputs. One output is the state of the flip flop. If there are two, the other output is the complement of the state.

D flip-flop

The simplest flip flop is the D flip flop. The name comes from Delay, since the output is just the input delayed until the next active clock transition. The next state of the D flip flop is the value of D before the clock transition.