Chapter5-Synchronous Sequential Logic

Lecture1-Introduction to Sequential Circuits, Study Latches

Objectives

- Introduction to Sequential Circuits
- Comparison of Combinational and Sequential Circuits
- Study Synchronous and Asynchronous Sequential Circuits
- Design Different Types of Latches

Chapter Contents

Sequential Circuit Definitions

Latches

Flip-flops

Analysis of Clocked Circuits

State Reduction and Assignment

Design Procedure

Combinational vs Sequential Circuits

Combinational

- The outputs are all the times dependent on the current inputs and change as and when inputs change.
- Consist of logic gates with no memory elements i.e memory- less.
- Defined by a set of Boolean equations.
- Faster in speed because of parallel mode of operation.
- Combinational circuits require more hardware
- Easier to design and simple in handling.
- Parallel Adder is a combinational circuit.

Sequential

- The outputs depend not only on the inputs but also on the present state of storage elements.
- Consist of storage elements besides logic gates.
- Defined by flip-flop input equations, state equations, and output equations.
- Slower than combinational circuits because of serial mode of operation.
- Sequential circuits require less hardware
- Comparatively harder to design because of clocking problems.
- Serial adder is a sequential circuit.

Sequential Circuits

- A sequential circuit consists of combinational circuits to which storage elements are connected to form a feedback path.
  - Storage elements are latches or flip-flops. These are single-bit storage devices and store binary information. The binary information stored in these in these elements defines the state of sequential circuit at that time.
  - Outputs of a sequential circuit are a function of the inputs and the internal state of the storage elements.
  - Next state of the storage elements is also a function of external inputs and the present state.
  - The term Present state is used to mean the Q output of the latch or flip-flop at the time the input signals are applied(or changed).
  - The term Next state is used to mean the state of the Q output after the latch or flip-flop has reacted to these input signals.
- State refers to the currently known condition of the circuit including what data is stored in the storage elements.
- A sequential circuit is specified by a time sequence of inputs, outputs, and internal states.

Sequential Circuit- Block Diagram

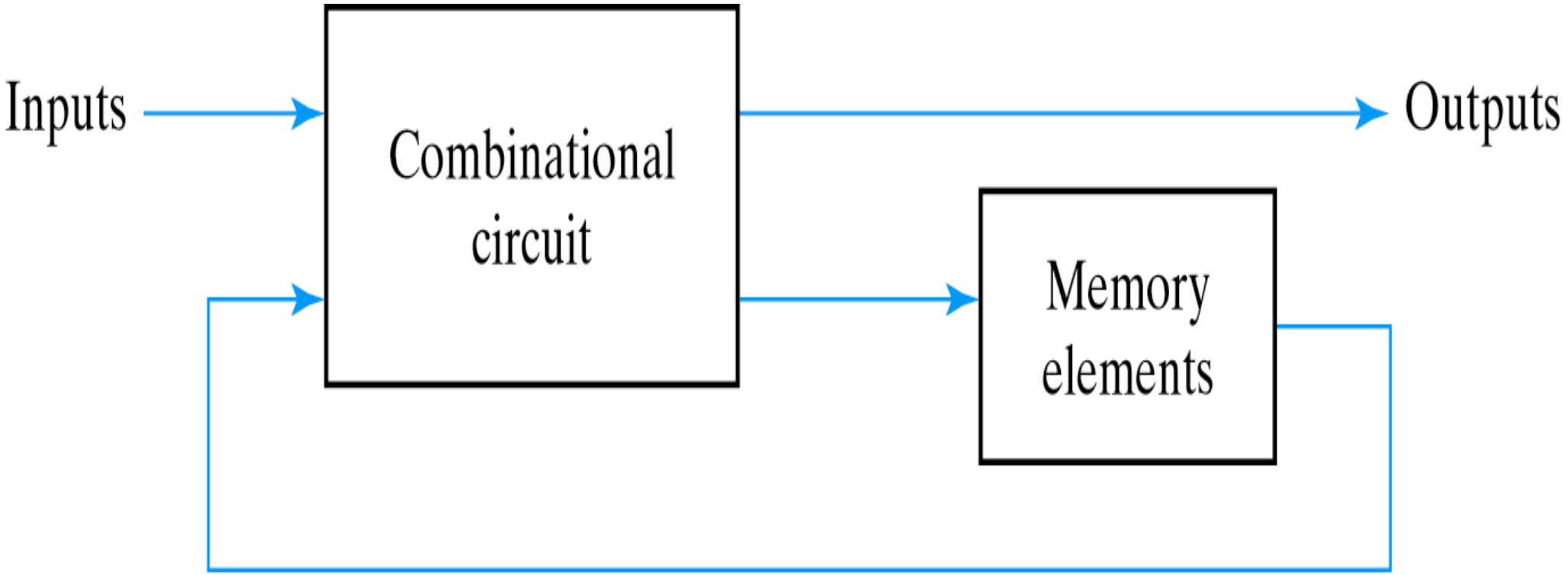


Fig. 5-1 Block Diagram of Sequential Circuit

Sequential Circuit Types

- There are two types of sequential circuits Synchronous and Asynchronous :
  - A synchronous sequential circuit is a system whose behavior can be defined from the knowledge of its signals at discrete instants of time
    - Storage elements in synchronous circuits are affected at discrete instants of time
    - Synchronization is achieved by a timing device called a clock generator that provides a periodic train of clock pulses.
    - Synchronous sequential circuits that use clock pulses in the inputs of storage elements are called clocked sequential circuits.
    - The storage elements used in clocked sequential circuits are called flip-flops.
  - An asynchronous sequential circuit depends on the input signals at any instant of time and the order in which the inputs change
    - Storage elements in asynchronous circuits are usually time-delay devices.
    - The storage capability of time-delay devices is due to the time it takes for the signal to propagate through the device.
    - It is also regarded as combinational circuit with feedback.

Synchronous Vs Asynchronous Sequential Circuits

Synchronous

- In synchronous circuits, memory elements are clocked flip-flops.
- In synchronous circuits, the change in input signals can affect memory elements upon active edges of clock i.e edge-triggered.
- The maximum operating speed of clock depends on time delays involved.
- Synchronous sequential circuits are easier to design.
- More stable because all the components are driven from a common reference signal, that is clock. The inputs are sampled and output changes in relation to clock.

Asynchronous

- In asynchronous circuits, memory elements are either unclocked flip-flops i.e latches or time delay elements.
- In asynchronous circuits, the change in input signals can affect memory elements at any instant of time.
- Because of absence of clock, asynchronous circuits can operate faster than synchronous circuits.
- Asynchronous sequential circuits are difficult to design.
- Less stable and prone to momentary false outputs i.e glitches.

Synchronous Vs Asynchronous Sequential Circuits

Synchronous

- An independent signal oscillating at a known frequency is called clock.
- An edge-triggered D flip-flop, constructed from several latches is synchronous circuit.

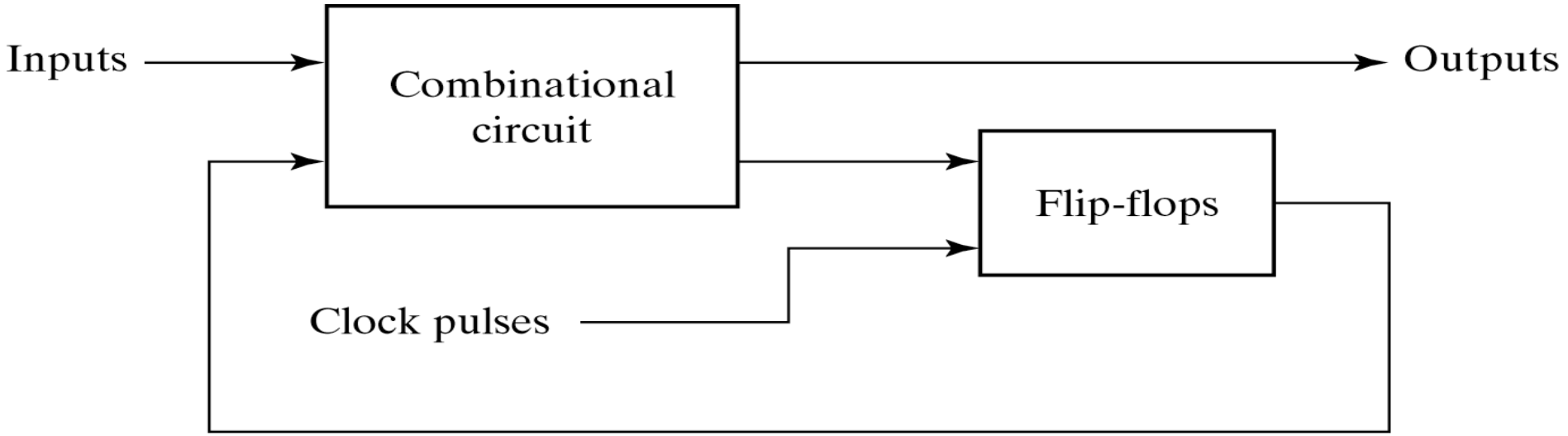
Asynchronous

- Asynchronous sequential circuits lie at the heart of every synchronous circuit.
- The basic R-S latch is asynchronous circuit.

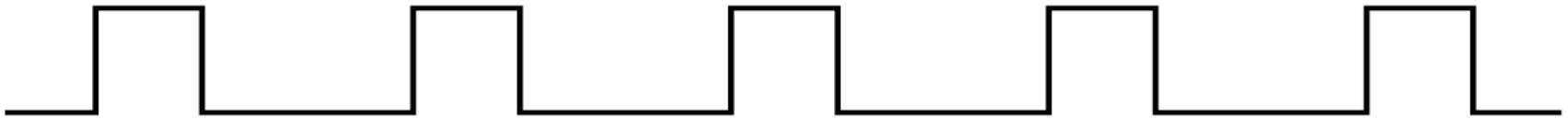
Synchronous Sequential Circuit

- A synchronous sequential circuit employs signals that affect the storage elements at only discrete instants of time.
- Synchronization in synchronous sequential circuits is achieved by timing device called a clock generator
- A clock generator generates a periodic train of clock pulses
- The clock pulses are distributed throughout the system in such a way that storage elements are affected only with the arrival of each pulse
- Clock pulses are employed with other circuitry that indicates the changes to the storage elements.
- Synchronous sequential circuits that use clock pulses in the inputs of storage elements are called clocked sequential circuits.
- The design of synchronous circuits is feasible because they seldom manifest instability problems and their timing is easily broken down into independent discrete steps, each of which can be considered separately.
- The memory elements used in these circuits are flip-flops

Block Diagram of Synchronous Sequential Circuit



(a) Block diagram



(b) Timing diagram of clock pulses

Fig. 5-2 Synchronous Clocked Sequential Circuit

Flip Flops

- The storage element used in clocked sequential circuits are called flip flops.
  - A flip flop is a binary storage device capable of storing one bit of information. It changes output in response to clock input and not in response to data input.
  - In a stable state, the output of a flip-flop is either 0 or 1.
  - Multiple flip flops can be used to store multiple bits.
  - Flip flops receive their inputs from a combinational circuit and a clock signal.
  - A Flip flop maintains a binary state indefinitely until directed by an input signal to switch states.
  - The transition from one state to the next occurs only at predetermined intervals dictated by the clock pulses.
- A flip flop circuit can maintain a binary state indefinitely until directed by an input signal to switch states.
- Flip flops are categorized by the number of inputs they possess and the manner in which the inputs affect the binary state.

Latches

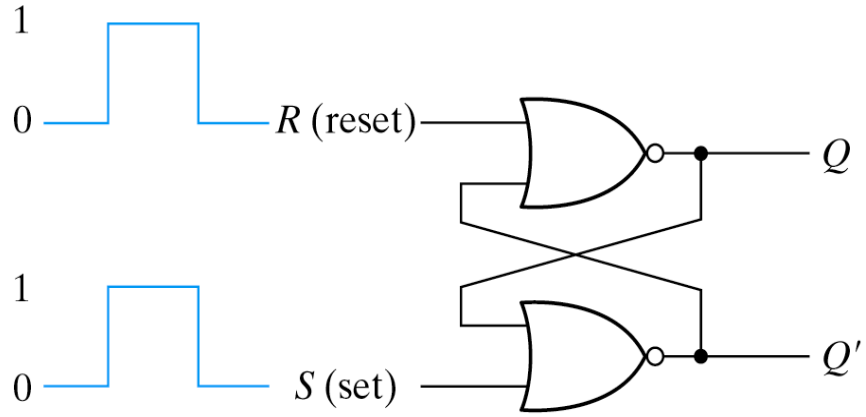
- Latches are the storage elements that operate with signal levels (rather than signal transitions).
- These are single-bit storage devices that are either without clock or may have a control input and change output in response to changes in input data as long as signal level is active.
- Latches are the building blocks for all flip flops.
- Although latches are useful for storing binary information and for the design of asynchronous sequential circuits, they are not practical for use in synchronous sequential circuits.
- Because they are building blocks of flip-flops

SR Latch

- The SR latch is a circuit constructed with two cross-coupled NOR gates or two cross-coupled NAND gates.
  - There are two inputs:
    - S is for set
    - R is for reset
  - There are two complementing outputs Q and Q’

SR Latch Logic (NOR Gates)

- The SR latch with NOR Gates has two useful states
  - The value stored in the output Q determines the state of a latch.
  - When Q = 1 and Q′ = 0 the latch is said to be in a set state.
  - When Q = 0 and Q′ = 1 the latch is said to be in a reset state.
  - Q and Q′ are normally complement each other under normal operation of the latch.
    - When both inputs are equal to 1 at the same time, an undefined state with both outputs equal to 0 occurs



(a) Logic diagram

S	R	Q	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	

(b) Function table

Fig. 5-3 *SR* Latch with NOR Gates

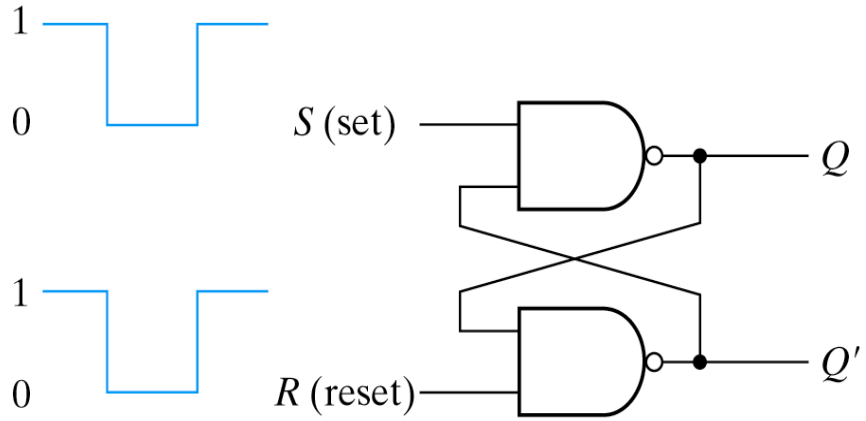
SR Latch (NOR) Logic

- Normally the two inputs to the SR latch (NOR) are left at 0.
- If the state needs to be changed then the application of a momentary 1 to the S input causes the latch to go to the set state.
  - The S input must go back to 0 before any other changes to avoid the occurrence of the undefined state. The circuit remains in set state.
- A momentary 1 to the R input causes the latch to go to the reset state.
  - The R input must go back to 0 before any other changes to avoid the occurrence of the undefined state. The circuit remains in reset state.
- If a 1 is applied to both S and R then the outputs both go to 0, the undefined state.

S	R	Q	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	

SR Latch Logic (NAND Gates)

- The SR latch with NAND Gates has two useful states
  - When Q = 1 and Q' = 0 the latch is said to be in a set state.
  - When Q = 0 and Q' = 1 the latch is said to be in a reset state.
  - Q and Q' are normally complement of each other. When both inputs are equal to 0 at the same time, an undefined state with both outputs equal to 1 occurs



(a) Logic diagram

S	R	Q	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	

(b) Function table

Fig. 5-4 *SR* Latch with NAND Gates

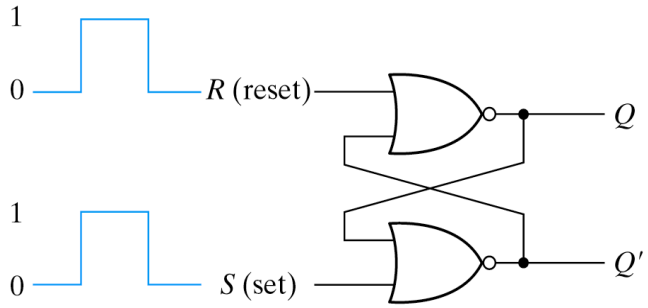
SR Latch (NAND) Logic

- Normally the two inputs to the SR latch (NAND) are left at 1.
- If the state needs to be changed then the application of a momentary 0 to the S input causes the latch to go to the set state.
  - The S input must go back to 1 before any other changes to avoid the occurrence of the undefined state. The circuit remains in set state.
- A momentary 0 to the R input causes the latch to go to the reset state.
  - The R input must go back to 1 before any other changes to avoid the occurrence of the undefined state. The circuit remains in reset state.
- If a 0 is applied to both S and R then the outputs both go to 1, the undefined state.

S	R	Q	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	

NAND Vs NOR Implementations

- The input signals for the NAND latch requires the complement of those values used for the NOR latch
- Because the NAND latch requires a 0 signal it is sometimes called a S´-R´ latch.

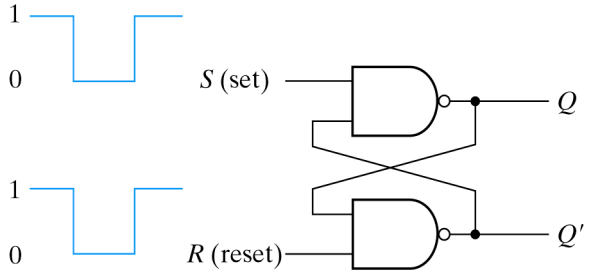


(a) Logic diagram

S	R	Q	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	

(b) Function table

Fig. 5-3 *SR* Latch with NOR Gates



(a) Logic diagram

S	R	Q	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	

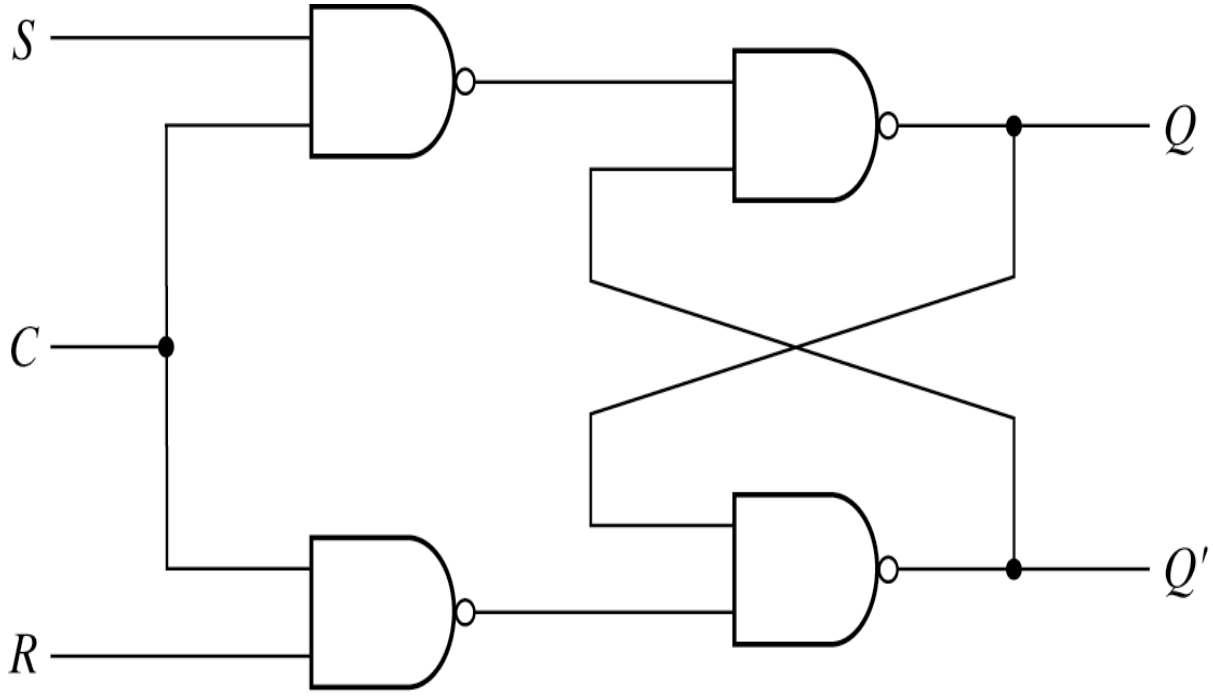
(b) Function table

Fig. 5-4 *SR* Latch with NAND Gates

SR Latch Modification

- 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 following example uses the C enable input with 1 allowing inputs S and R to flow through and 0 disallowing the flow of the inputs S and R. In other words S and R are allowed to change the flip-flop only when C = 1 and If C=0, S and R can't change output
  - It consists of basic SR latch and two additional NAND gates
  - The control input C acts as an enable input 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. This is quiescent (inactive) state.
  - When the control input goes to 1, the information for S or R is allowed to effect the SR latch (active state)
  - The set state is reached with S=1, R=0 and C=1. To change to the reset state, the inputs must be S=0, R=1 and C=1. In either cases when C returns to 0, the circuit remains in its current state

SR Latch With Control Input



(a) Logic diagram

C	S	R	Next state of 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

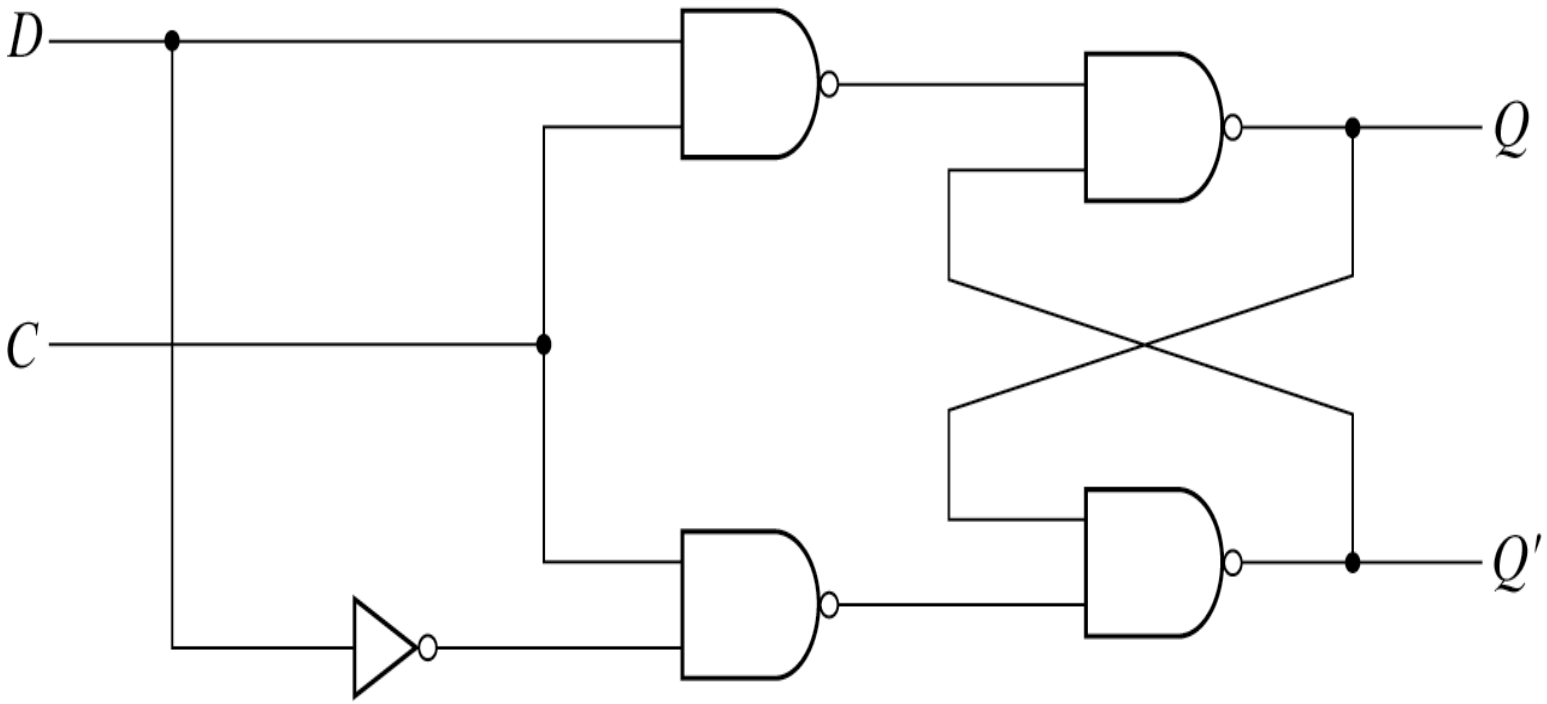
(b) Function table

Fig. 5-5 SR Latch with Control Input

D Latch

- A D latch modifies the SR latch to avoid the undesirable condition of the indeterminate state by ensuring that S and R are never equal to 1 at the same time.
  - There are only two inputs:
    - D is the data input
    - C is the control input
  - The D input goes directly to the S input and its complement goes to the R input.
  - When control input C is left at 0 the state of the latch remains constant (regardless of value of D).
  - If C = 1 and D = 1 the output Q goes to 1, placing the circuit in the set state.
  - If C = 1 and D = 0 the output Q goes to 0, placing the circuit in the reset state.

D Latch Logic Diagram



(a) Logic diagram

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

(b) Function table

Fig. 5-6 D Latch

Notes on D Latches

- The D Latch has the ability to hold data in its internal storage.
- The data input of the D latch is transferred to the Q output when the control input is enabled.
  - The output follows changes in the data input as long as the control input is enabled.
  - For this reason, the latch is also called the transparent latch.
- When the control input is disabled, the output of the latch remains in the state it was in just prior to the control input being disabled.

Graphic Symbols for Latches

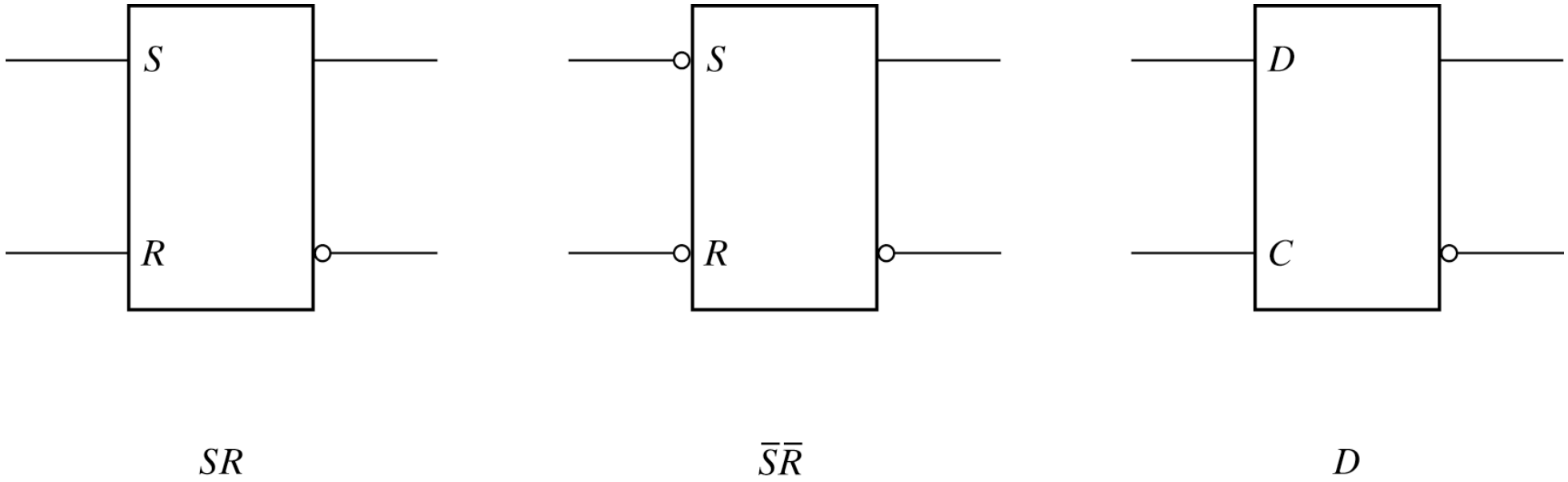


Fig. 5-7 Graphic Symbols for Latches

The End