Digital System Design

Lecture 16

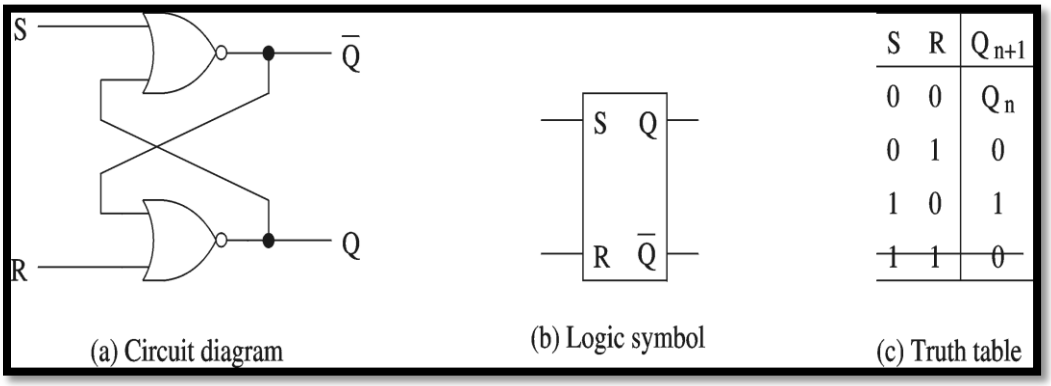
Sequential Circuits

Latches and Flip-Flops

Objectives:

1. ***RS latch implementation using A NOR gate.***
2. ***RS latch using NAND gates.***
3. ***Clocked RS latch uses a clock signal.***
4. ***D Latch***

1. RS latch implementation using A NOR gate



- **SR** latch have
  - two inputs **S** and **R**:
    - **S** is called **set** and it is used to produce **HIGH** on **Q** (i.e. store **1**).
    - **R** is called **reset** and it is used to produce **LOW** on **Q** (i.e. store **0**).
  - **Q'** is **Q** complementary output.
- The ***output*** of the **SR** latch depends on **current** as well as **previous inputs or state** (can be changed as soon as its inputs change).
- The ***truth table*** of **RS** latch is shown below:

S	*R*	*Q*	*Q +*	*Comment*
0	*0*	*0*	*0*	*Hold state*
0	*0*	*1*	*1*	
0	*1*	*X*	*0*	*Reset state*
1	*0*	*X*	*1*	*Set state*
1	*1*	*X*	*Invalid*	*Avoid state*

- The output equations:
  - $Q = \overline{(R + \bar{Q})}$
  - $\bar{Q} = \overline{(S + Q)}$
- ***Cases*** according to the truth table:
  - ***Case 1.1:*** $S = 0, R = 0$ and initial condition: $Q = 1, \bar{Q} = 0$.
    The output Q after input is applied would be:
    $$Q = \overline{(R + \bar{Q})} = 1$$
    And the complement output would be:
    $$\bar{Q} = \overline{(S + Q)} = 0$$
  - ***Case 1.2:*** $S = 0, R = 0$ and initial condition: $Q = 0, \bar{Q} = 1$.
    The output Q after input is applied would be:
    $$Q = \overline{(R + \bar{Q})} = 0$$
    And the complement output would be:
    $$\bar{Q} = \overline{(S + Q)} = 1$$

When both S and R inputs are LOW, the output is retained as before (i.e. there is no state change).

  - ***Case 2.1:*** $S = 1, R = 0$ and initial condition: $Q = 1, \bar{Q} = 0$.
    The output Q after input is applied would be:
    $$Q = \overline{(R + \bar{Q})} = 1$$
    And the complement output would be:
    $$\bar{Q} = \overline{(S + Q)} = 0$$
  - ***Case 2.2:*** $S = 1, R = 0$ and initial condition: $Q = 0, \bar{Q} = 1$.
    The output Q after input is applied would be:
    $$Q = \overline{(R + \bar{Q})} = 1$$
    And the complement output would be:
    $$\bar{Q} = \overline{(S + Q)} = 0$$

When S is HIGH and R is LOW, the output Q is HIGH (i.e. set mode).

  - ***Case 3.1:*** $S = 0, R = 1$ and initial condition: $Q = 1, \bar{Q} = 0$.
    The output Q after input is applied would be:
    $$Q = \overline{(R + \bar{Q})} = 0$$
    And the complement output would be:
    $$\bar{Q} = \overline{(S + Q)} = 1$$
  - ***Case 3.2:*** $S = 0, R = 1$ and initial condition: $Q = 0, \bar{Q} = 1$.
    The output Q after input is applied would be:
    $$Q = \overline{(R + \bar{Q})} = 0$$

And the complement output would be:

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

When S is LOW and R is HIGH, the output Q is LOW (Reset mode).

- ***Case 4.1:*** $S = 1, R = 1$ and initial condition: $Q = 1$, $\bar{Q} = 0$.
  The output Q after input is applied would be:

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

  And the complement output would be:

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

- ***Case 4.2:*** $S = 1, R = 1$ and initial condition: $Q = 0$, $\bar{Q} = 1$.
  The output Q after input is applied would be:

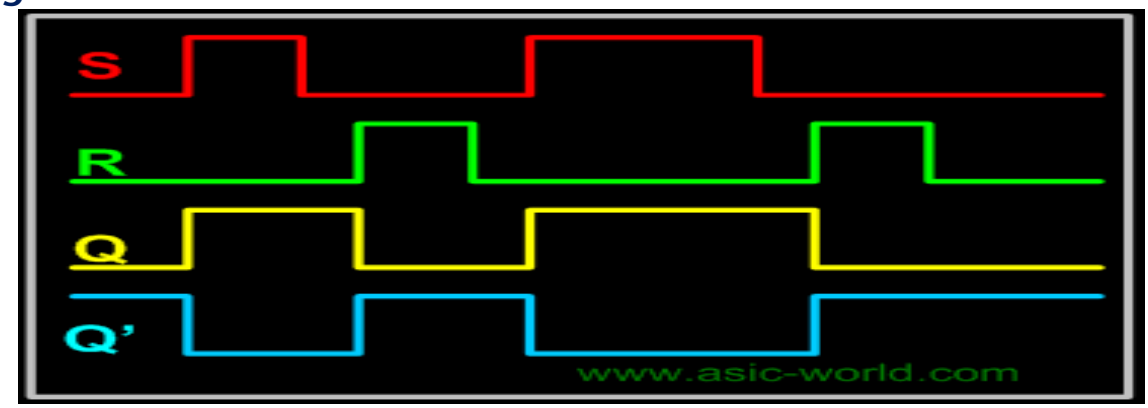
$$Q = \overline{(R + \bar{Q})} = 0$$

  And the complement output would be:

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

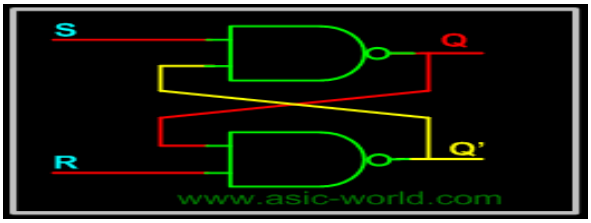
When both S and R inputs are HIGH, the outputs are invalid (undefined state).

- The **waveform** below shows the operation of ***RS Latch based NOR gates.***



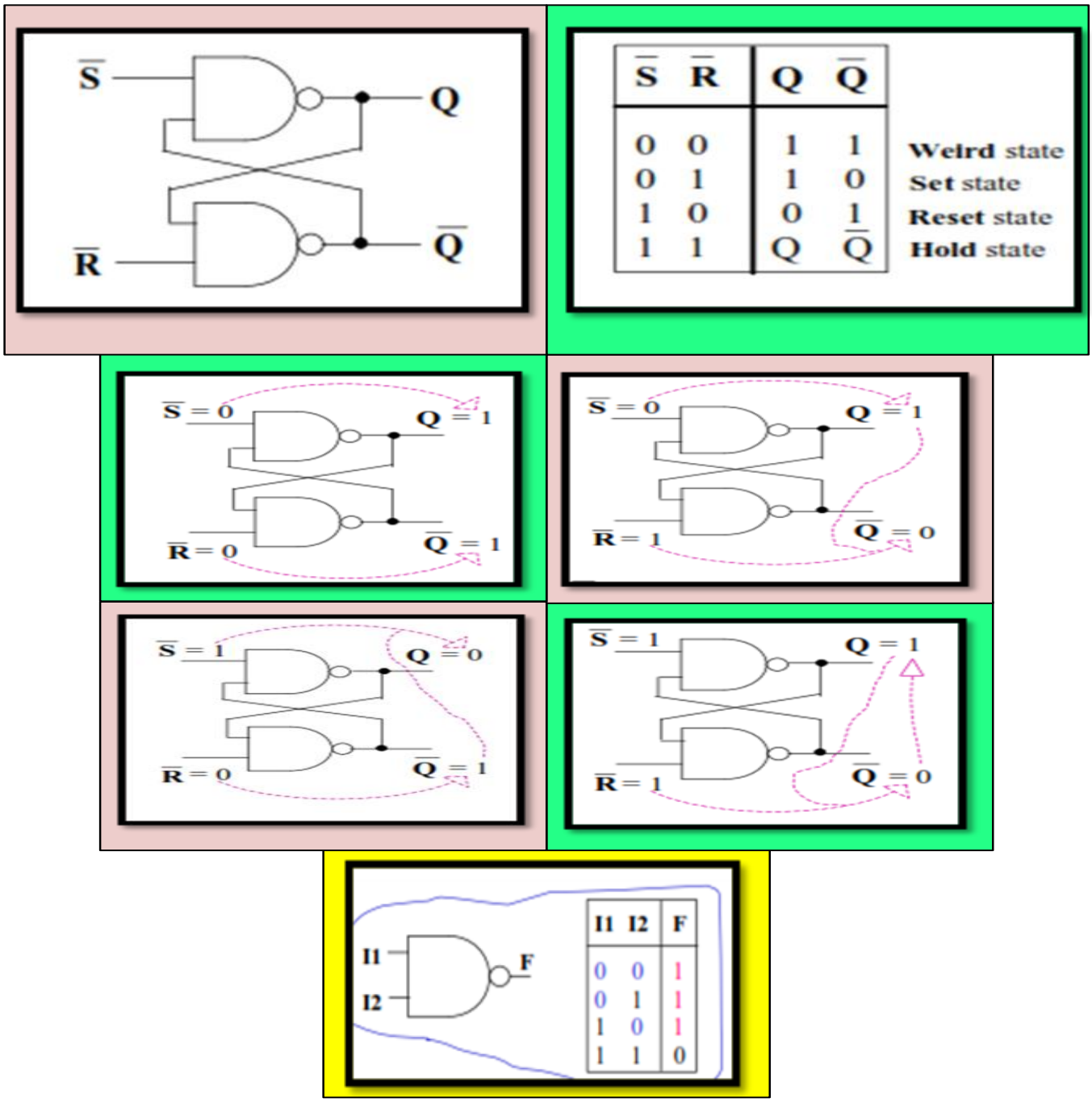
2. RS latch using NAND gates

- It is possible to construct the RS latch using ***NAND gates***.
- The circuit and Truth table of RS latch using NAND is shown below:

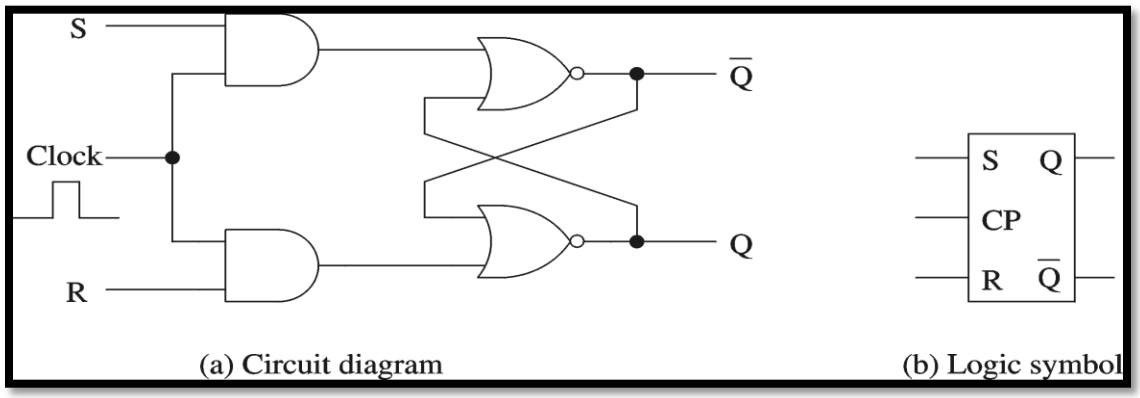


S	R	Q	Q +
1	1	0	0
1	1	1	1
0	1	X	0
1	0	X	1
0	0	X	Invalid

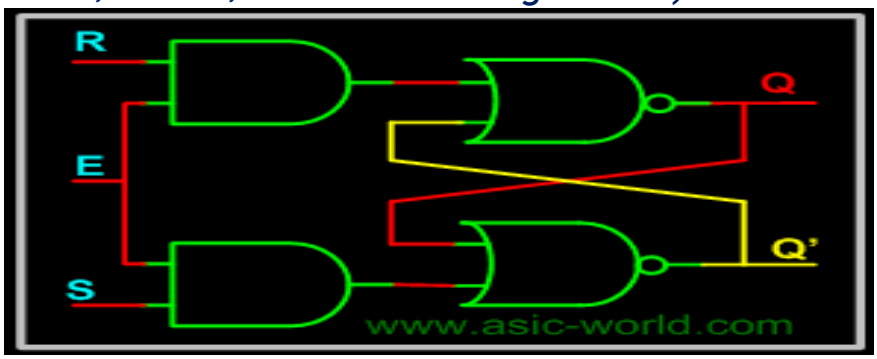
When $R = 0$ and $S = 0$ becomes the invalid case.



3. Clocked RS latch uses a clock signal



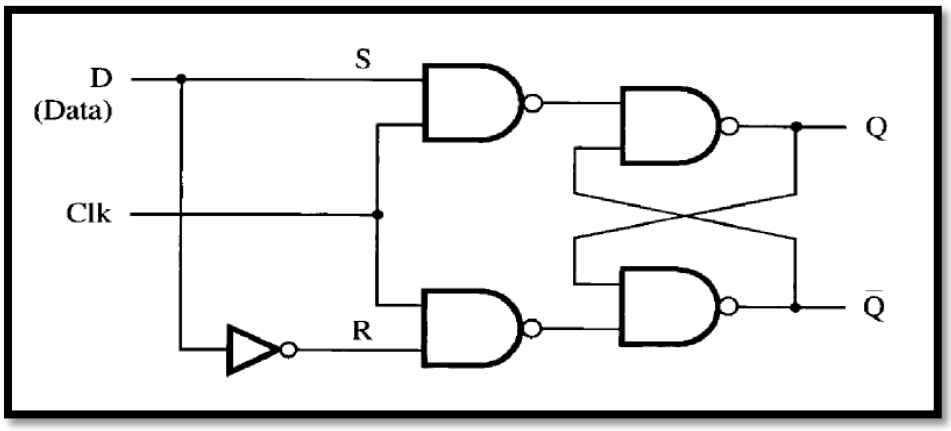
- The circuit below shows the level sensitive ***RS*** latch. Control signal "***Enable***" ***E*** is used to gate the input ***S*** and ***R*** to the ***RS*** Latch.
  - When Enable ***E*** is ***HIGH***, both the ***AND*** gates act as ***buffers*** and thus ***R*** and ***S*** appears at the ***RS*** latch input and it functions like a normal ***RS*** latch.
  - When Enable ***E*** is ***LOW***, it drives ***LOW*** to both inputs of ***RS*** latch (***:*** $S = 0, R = 0$, i.e. the not change mode).



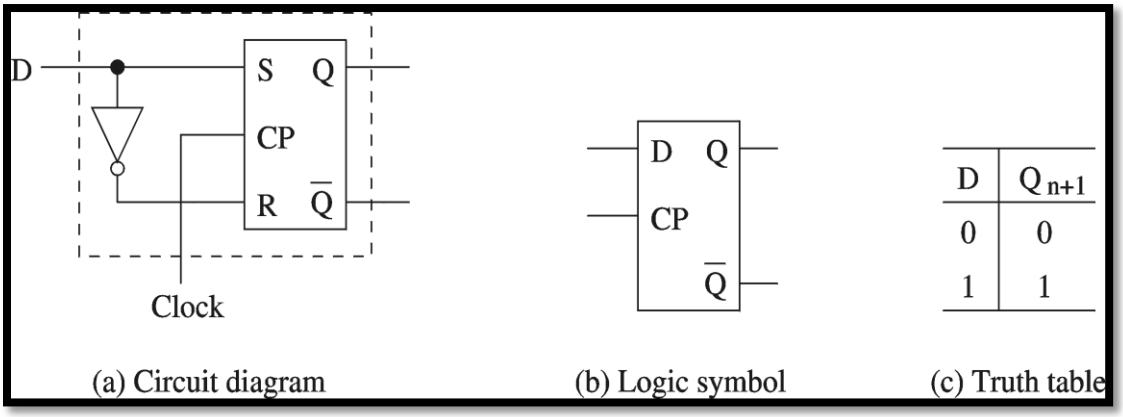
4. D Latch

- An RS-flip flop is rarely used in actual sequential logic; however, it is the fundamental building block for the very useful.
- The RS latch contains an invalid state; to eliminate this state we can ensure that S and R are never equal (Avoids the SR = 11 state).
- To eliminate this state, we connect S and R together with an inverter. Thus we have D Latch.
- The D-flip flop has only a single data input, this input is called D or Data input.
- That data input is connected to the S input of an RS-flip flop, while the inverse of D is connected to the R input.
- D latch is called delay flip-flop or delay latch also.

- Below is the truth table and circuit of D latch.



D	Q	Q+
1	X	1
0	X	0



- In real world designs (ASIC/FPGA Designs) only D latches/Flip-Flops are used.
- To allow the flip flop to be in a holding state, a D-flip flop has a second input called ``Enable."