# CSE 4205
# Digital Logic Design

# Sequential Logic

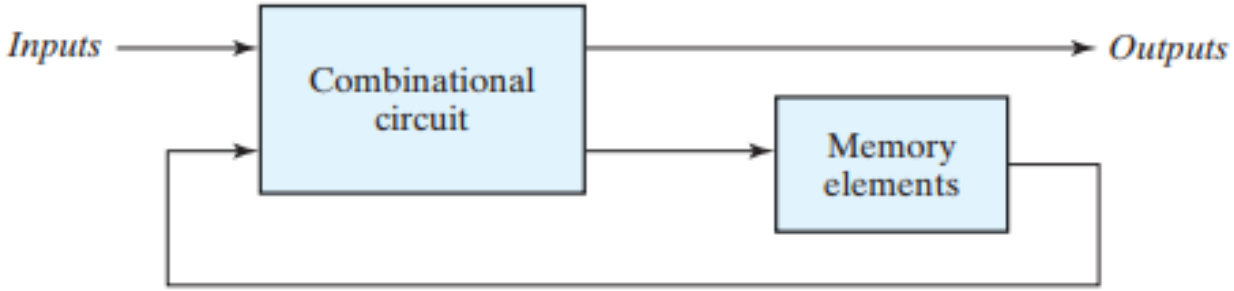
**Course Teacher: Md. Hamjajul Ashmafee**

**Lecturer, CSE, IUT**

**Email: ashmafee@iut-dhaka.edu**

# Introduction

- So far digital circuits considered – **combinational circuit** – output at any instant entirely dependent on the inputs presents at that time.
- Most systems in practice also include memory element – **sequential circuit/ logic**



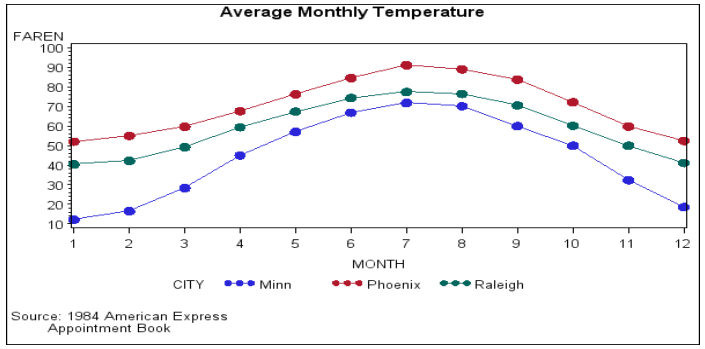

- **Memory elements** are connected to form a feedback path – device to store binary information within them.

# *Introduction…*

- The **binary information** stored in memory at any given time – **state** of the sequential circuit
- The sequential circuits receive binary information from **external inputs.**
- These **inputs** as well as **present state of the memory element**  the binary value at the output terminals
- **Present inputs and states** also determine the conditions for changing the state in the memory elements
- *Present output = F (present input, present state)*
- *Next state of memory element = F (present input, present state)*

# *Introduction…*

- Sequential circuit – **time sequence** of inputs, outputs and internal states




- ***Two** types sequential circuit* – depending on **timing** of their signal

❑**Synchronous sequential circuit**

Behavior defined based on the signal at **discrete** instants of time

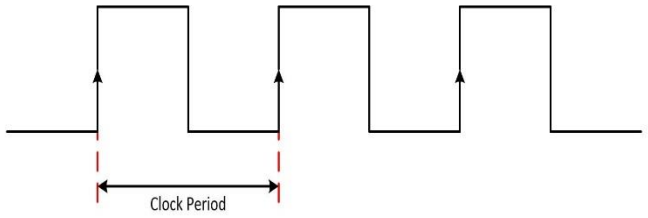
❑**Asynchronous sequential circuit**

Behavior defined based on the **order** of input signals at **any instant** of time

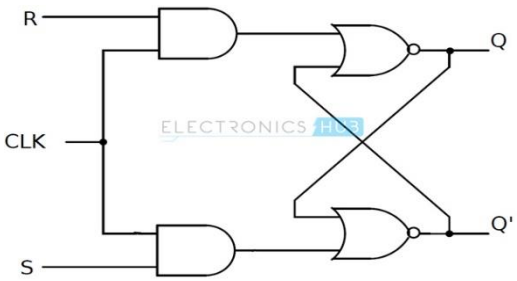
# *Introduction…*

- **Memory elements** in Asynchronous Sequential Circuit – time delay device (intentionally added)
- **Time delay device** – *takes a finite time* for the signal to propagate through the device
- In practice, **internal propagation delay** of logic gates is sufficient duration to produce necessary delay so it is possible to avoid physical time delay units
- This propagation delays through logic gates in memory unit constitute the required memory, thus an asynchronous sequential circuit regarded as a combinational circuit with feedback. (feedback - instability problem)

# *Introduction…*




- Synchronous sequential circuit – employs signals **at discrete instants of time** – use pulses of limited duration (pulse amplitude – 0 and 1)
- Inputs from different sources – have separate independent sources of pulse – **arrival time may be delayed** (unpredictable) – separate the pulses slightly (not synchronized) – **unreliable operation**
- Binary signal – **voltage amplitude**
- Synchronization achieved through a timing device – followed by all components (synchronized) – **master clock generator** (train of clock pulses) – clock pulse ANDed with specific signal
- **Clocked Sequential circuit**
  - Most frequently used
  - No instability problem



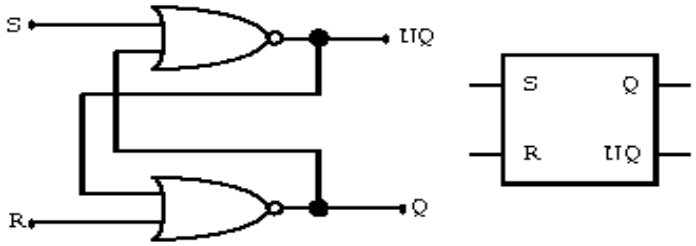
# *Flip flops*

- **Memory elements** in clocked sequential circuit – **flip-flop** – binary cell that stores one bit of information
- Flip-flop – **two outputs** – normal and complement value of the bit stored
- Flip flop – **maintains binary state** indefinitely (based on power supply)
- Differences in flip-flops – based on number of inputs and the manner in which the inputs affect the binary state

# Basic Flip-flop Circuit

- Constructed from NAND and NOR gates which form basic flip-flops
- Two inputs – Reset and Set; Two outputs – Q and Q'
- Q=1, Q'=0 : set (1) state and Q=0, Q'=1 : reset (0) state
- Both inputs should remain in "no change" state unless the state has to be changed and "undefined" state should be avoided

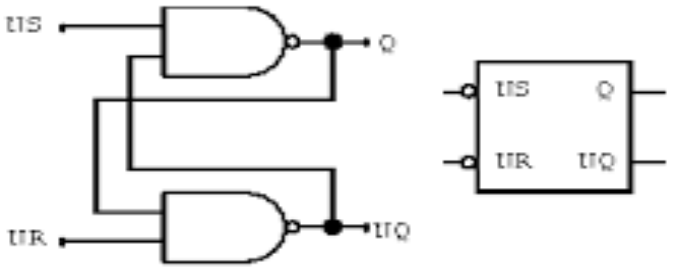
RS flip-flop




| $S$ | $R$ | $Q$ | $\overline{Q}$ |
|---|---|---|---|
| 0 | 0 | no change | |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | undefined | |

**Controlled by 1, if any input= 1, output =0: NOR TT**

$\overline{RS}$ flip-flop



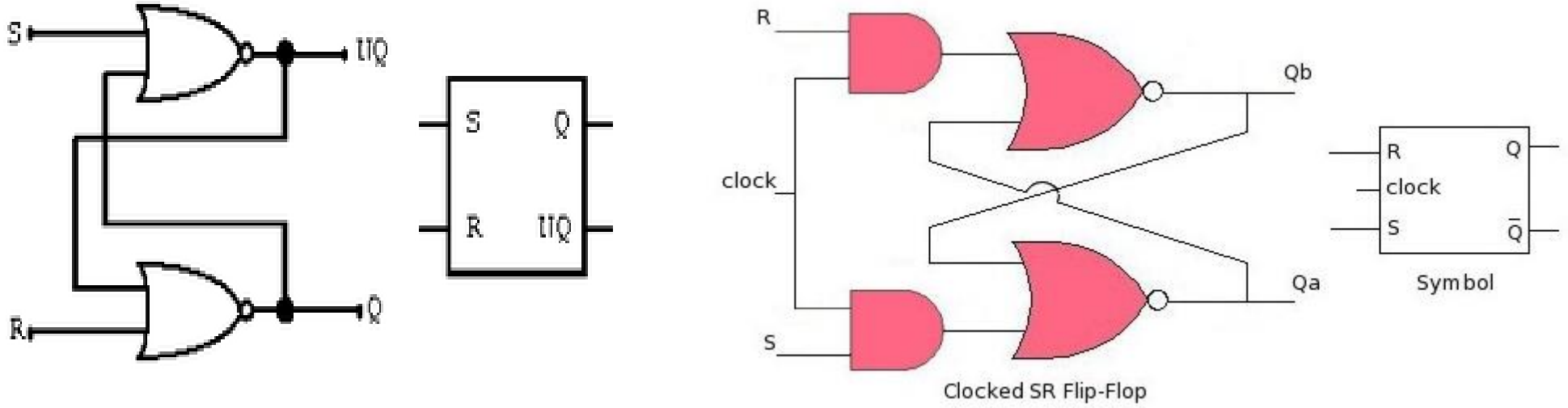

| $S$ | $R$ | $Q$ | $\overline{Q}$ |
|---|---|---|---|
| 0 | 0 | undefined | |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | no change | |

**Controlled by 0, if any input= 0, output =1: NAND TT**

# *Flip flop and Latches*

A **latch** is asynchronous, and the outputs can change as soon as the inputs.

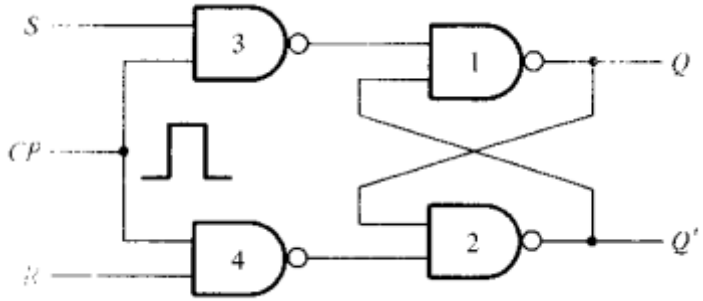
A **flip-flop**, on the other hand, is edge-triggered and only changes state when a control signal goes from high to low or low to high.

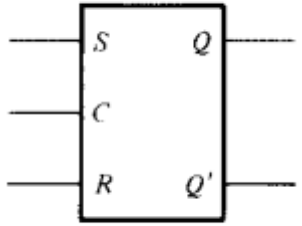


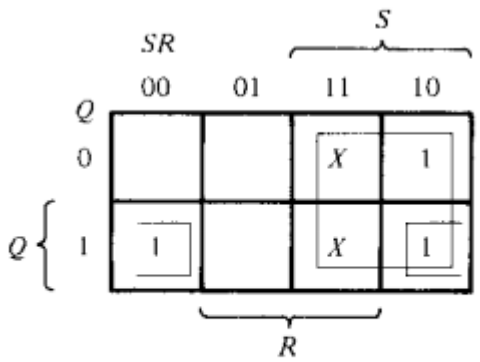
# *RS flip-flop*

- **Clocked circuit** – flip-flop will response when it is aligned with clock pulse
- **Clock pulse** – remarked as a triangle in the graphic symbol – Clocks are dynamic indicators – inputs will be responsive at clock's transitions
- **Q** = present state and **Q (t+1)** = next state
- **Characteristic table** summarize the operation of the flip-flop. (R, S, Q – i/p)
- **Graphic symbol** has three inputs (R,S,CP) and two outputs (Q and Q')
- **Characteristic equation** specifies the value of the next state as function of present state and inputs derived from the characteristic table. (R, S, Q – i/p)
- **An extra expression** is also needed to be constrained for indeterminate state (don't care in k-map)

# RS flip-flop…




$Q(t+1) = S + R'Q$

$SR = 0$

| *En* | *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 |

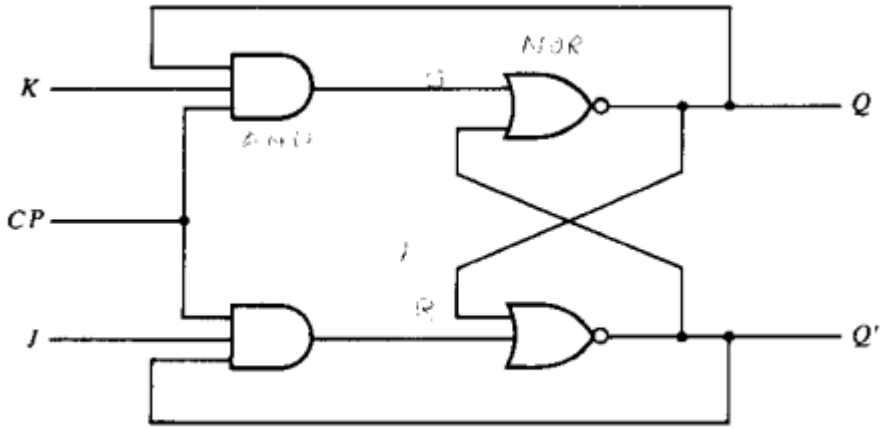
(b) Function table

| Q | S | R | Q (t+1) |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | Indeterminate |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | Indeterminate |

# *JK flip-flop*

- A **refinement of the RS** flip-flop to avoid indeterminate state
- **J behaves like set (S)** and **K behaves like reset/clear (R)**
- Instead of being indeterminate state it work as **toggle/complement** when both J and K are given input (1) – Q to Q'.
- **Q is ANDed with K and CP** and **Q' is ANDed with J and CP,** it ensures that it will be cleared if previously set and set if previously clear.
- JK flip-flop behaves like a RS flip-flop *except both of them are 1.*
- *When both of them are 1,* it will allow **one AND gate** to transmit which is connected to the **"current memory state=1"**
- When CP=J=K=1, causes repeated and continuous transitions of the output (Race around condition)

# JK flip-flop…



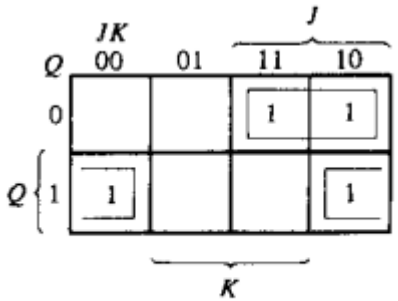

(a) Logic diagram

| Q | J | K | Q(t + 1) |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Flip-Flop Characteristic Tables

**JK Flip-Flop**

| J | K | Q(t + 1) | |
|---|---|---|---|
| 0 | 0 | $Q(t)$ | No change |
| 0 | 1 | 0 | Reset |
| 1 | 0 | 1 | Set |
| 1 | 1 | $Q'(t)$ | Complement |

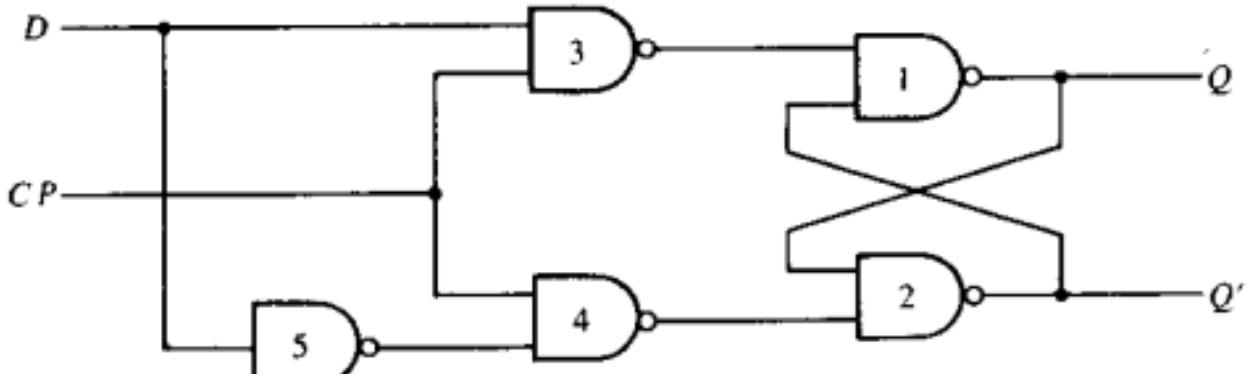



$Q(t+1) = JQ' + K'Q$

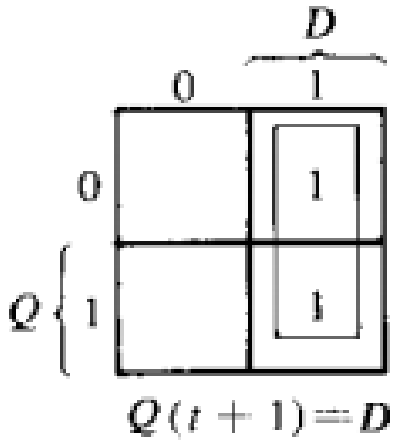
# *D Flip Flop*

- **Modification** of the clocked RS flip-flop
- D goes **directly to S** and as **complement (inverter) to R**, reducing the number of inputs
- *If CP = 0*, it will be independent of S and R inputs and both will be 1 for main circuit resulting in memory state.
- If **D=1**, switch the flip-flop to the **set state**
- If **D=0**, switch the flip-flop to the **reset/clear state**
- D flip-flop receives the this designation from its ability to transfer **"data"** into flip-flop

# D Flip Flop…

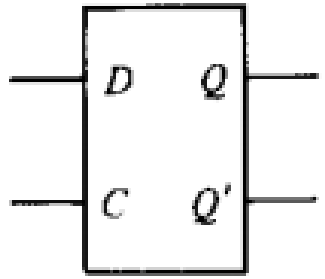



| Q | D | Q(t + 1) |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |




**D Flip-Flop**

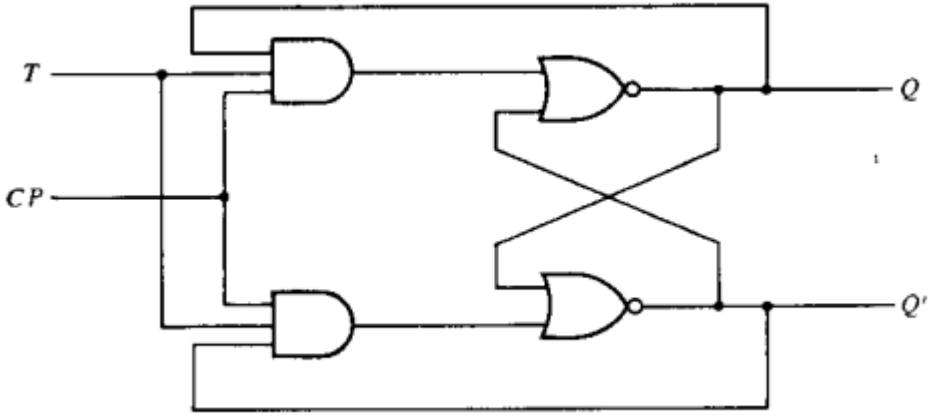
| D | Q(t + 1) | |
|---|---|---|
| 0 | 0 | Reset |
| 1 | 1 | Set |



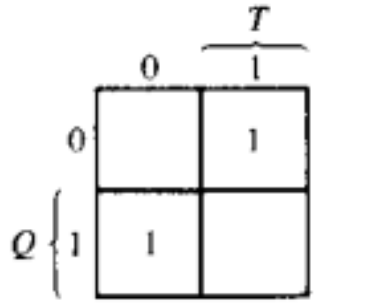
# *T flip flop*

- Single input version of JK flip-flop
- T flip-flop is obtained from a JK flip-flop if both inputs are tied together
- The designation of T comes from the ability of the flip-flop to “toggle” or change the state
- When T=1, it will complement the present state (CP=1).

# T flip flop…




| Q | T | Q(t + 1) |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |




$$Q(t+1) = TQ' + T'Q$$

**T Flip-Flop**

| T | Q(t + 1) | |
|---|---|---|
| 0 | $Q(t)$ | No change |
| 1 | $Q'(t)$ | Complement |