SNS COLLEGE OF ENGINEERING
(Autonomous)
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

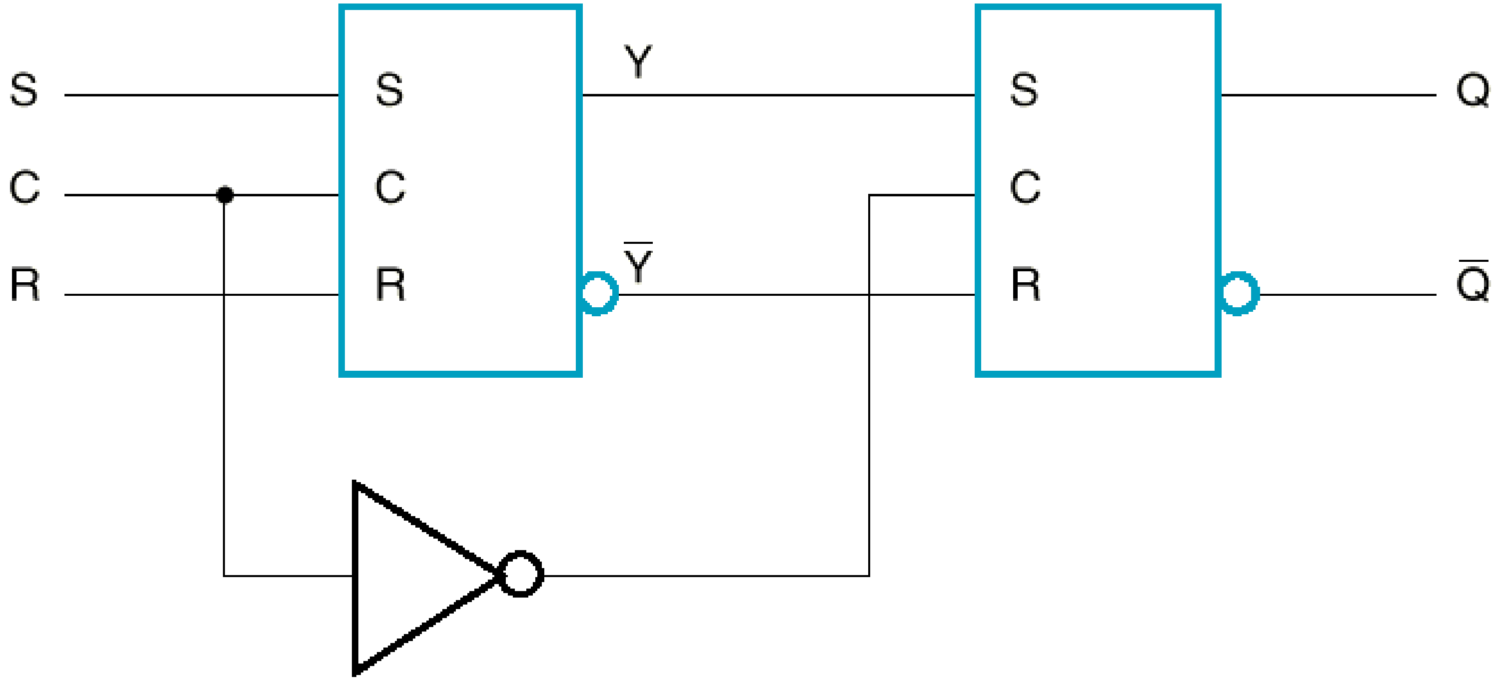
Sequential Circuits

Flip-Flops

Flip-Flops

- Latches are “transparent” (= any change on the inputs is seen at the outputs immediately when C=1).
- This causes synchronization problems.
- Solution: use latches to create flip-flops that can respond (update) only on specific times (instead of any time).
- Types: RS flip-flop and D flip-flop

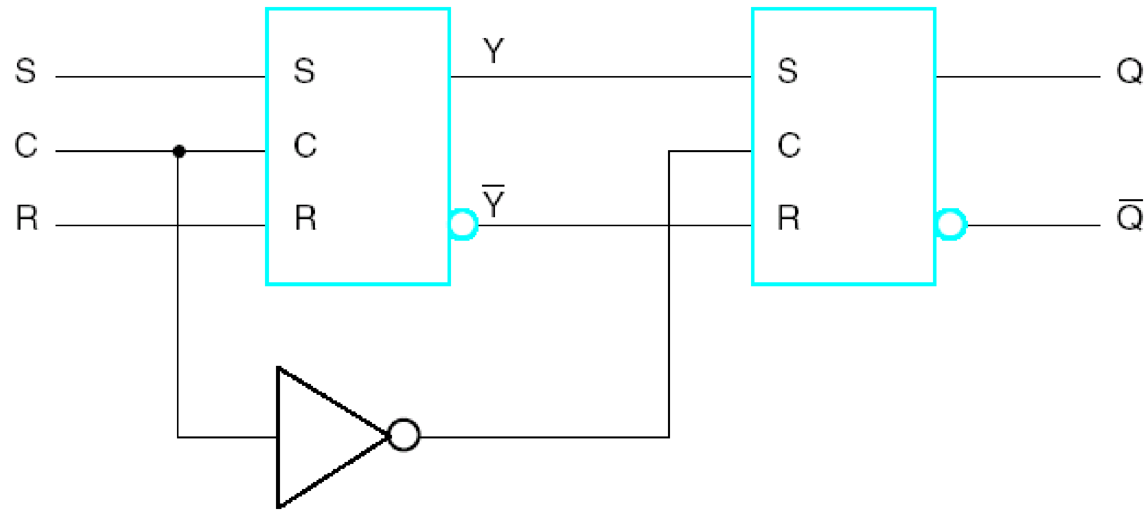
Master-Slave FF configuration using SR latches



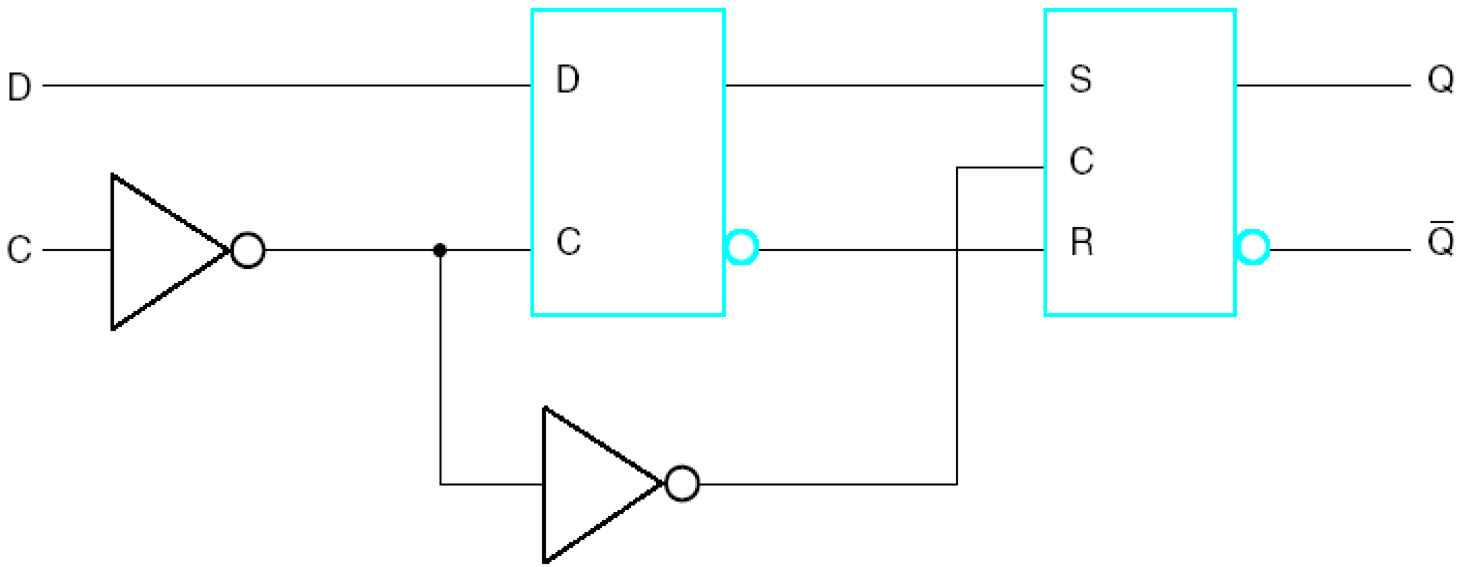
Master-Slave FF configuration using SR latches (cont.)

S	R	CLK	Q	Q'	
0	0	1	Q_0	Q_0'	Store
0	1	1	0	1	Reset
1	0	1	1	0	Set
1	1	1	1	1	Disallowed
X	X	0	Q_0	Q_0'	Store

- When C=1, master is enabled and stores *new* data, slave stores *old* data.
- When C=0, master's state passes to enabled slave, master not sensitive to new data (disabled).



D Flip-Flop



Characteristic Tables

- Defines the logical properties of a flip-flop (su
  as a truth table does for a logic gate).
- Q(t) – present state at time t
- Q(t+1) – next state at time t+1

Characteristic Tables (cont.)

SR Flip-Flop

S	R	Q(t+1)	Operation
0	0	Q(t)	No change/Hold
0	1	0	Reset
1	0	1	Set
1	1	?	Undefined/Invalid

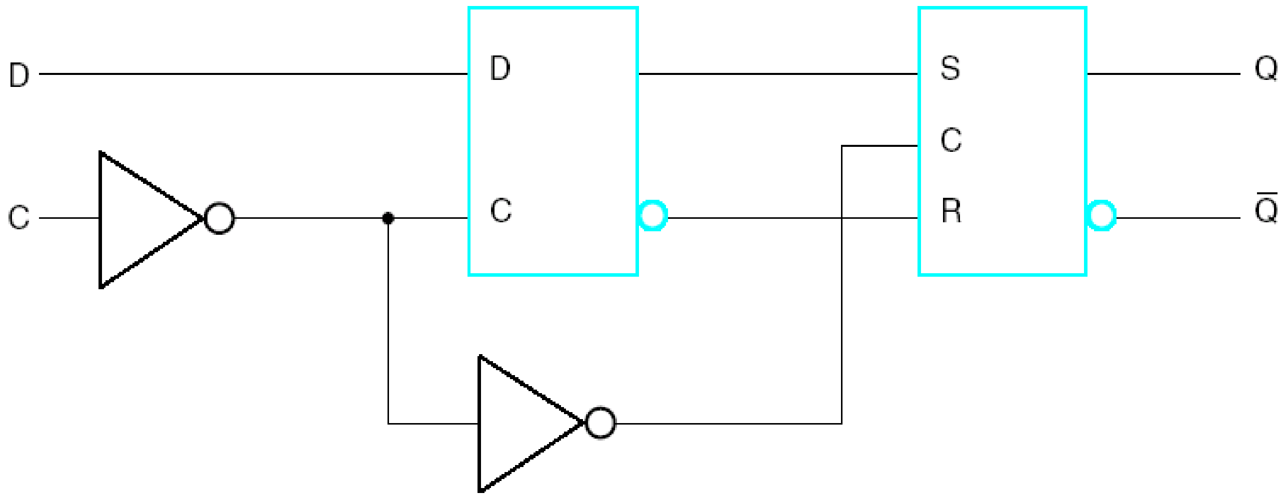
Characteristic Tables (cont.)

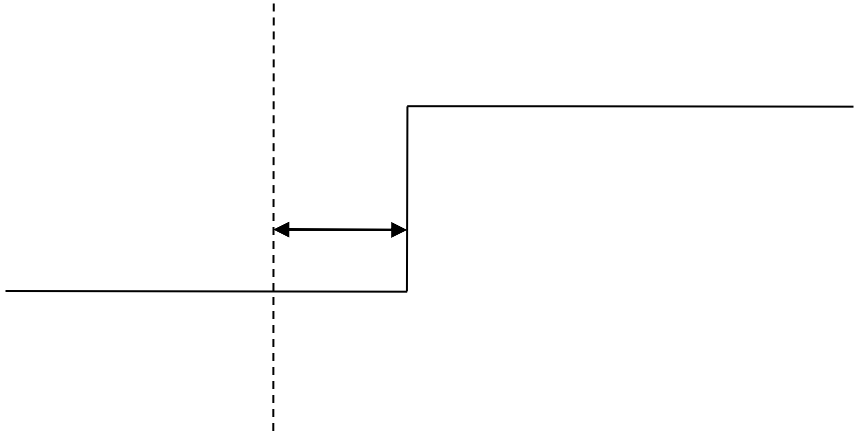
D Flip-Flop

D	Q(t+1)	Operation
0	0	Set
1	1	Reset

Characteristic Equation: $Q(t+1) = D(t)$

D Flip-Flop Timing Parameters





Setup time

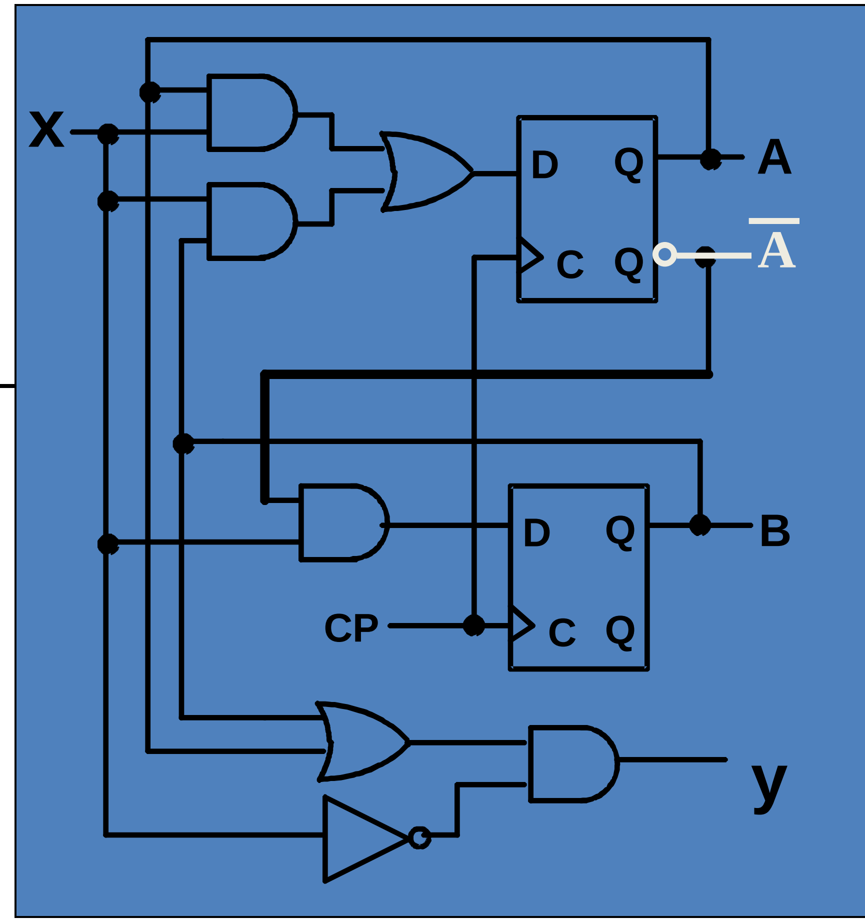
Sequential Circuit Analysis

- ***Analysis***: Consists of obtaining a <u>suitable</u> description that demonstrates the <u>time sequence</u> of inputs, outputs, and states.
- Logic diagram: Boolean gates, flip-flops (of any kind), and appropriate interconnections.
- The logic diagram is derived from any of the following:
  - Boolean Equations (FF-Inputs, Outputs)
  - State Table
  - State Diagram

Example

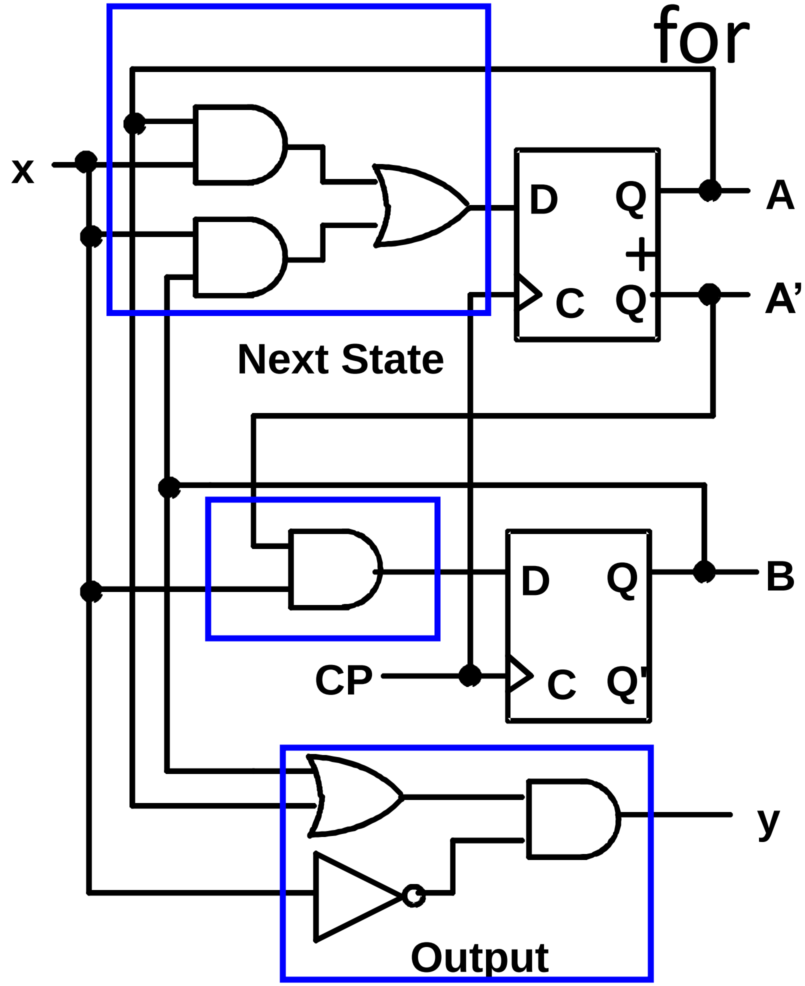
- Input: x(t)
- Output: y(t)
- State: (A(t), B(t))
- What is the Output

What is the Next State



Example (continued)

- Boolean equations for the functions:
  - $A(t+1) = A(t)x(t)$ $B(t)x(t)$
  - $B(t+1) = A'(t)x(t)$
  - $y(t) = x'(t)(B(t) + A(t))$



State Table Characteristics

- *State table* – a multiple variable table with the following four sections:
  - *Present State* – the values of the state variables for each allowed state.
  - *Input* – the input combinations allowed.
  - *Next-state* – the value of the state at time (t+1) based on the present state and the input.
  - *Output* – the value of the output as a function of the present state and (sometimes) the input.

From the viewpoint of a truth table:

- the inputs are Input, Present State
- and the outputs are Output, Next State

Example: State Table

- The state table can be filled in using the next state and output equations:
  - $A(t+1) = A(t)x(t) + B(t)x(t)$
  - $B(t+1) = \overline{A}(t)x(t);$
  - $y(t) = \overline{x}(t)(B(t) + A(t))$

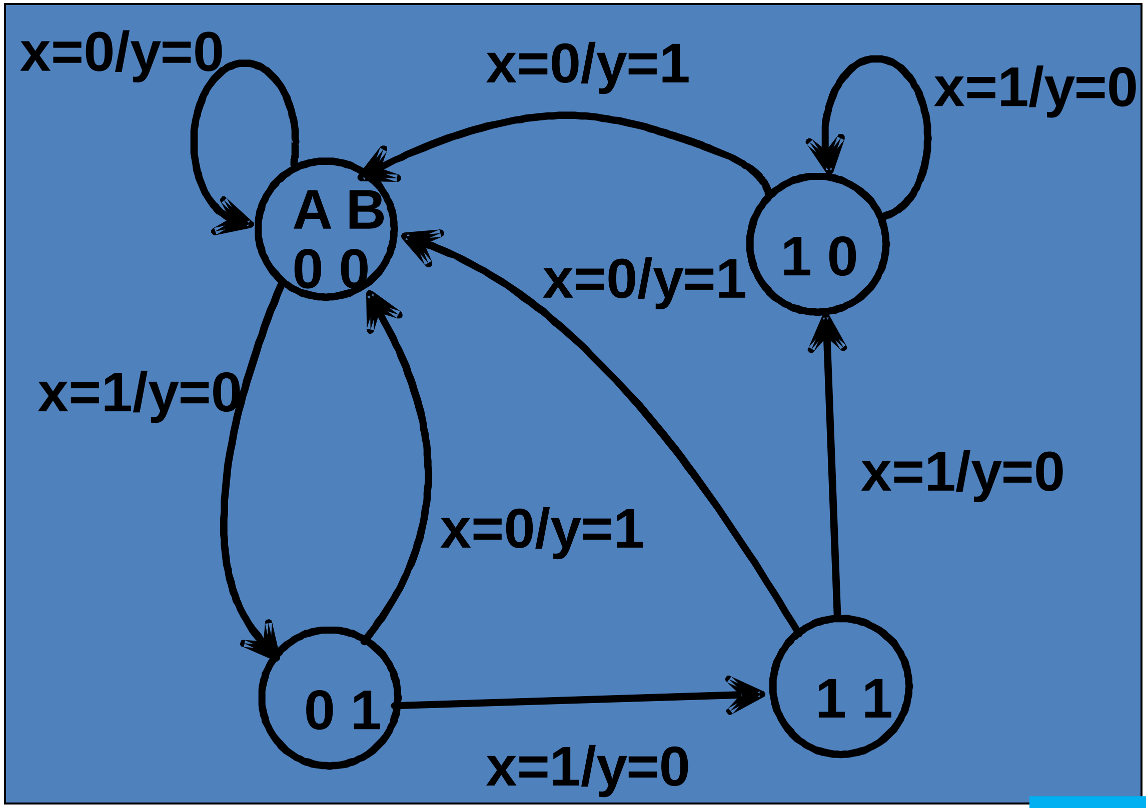
Present State		Input	Next State		Output
A(t)	B(t)	x(t)	A(t+1)	B(t+1)	y(t)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

State Diagrams

- The sequential circuit function can be represented in graphica form as a state diagram with the following components:
  - A circle with the state name in it for each state
  - A directed arc from the Present State to the Next State for each state transition
  - A label on each directed arc with the Input values which causes the state transition, and
  - A label:
    - On each circle with the output value produced, or
    - On each directed arc with the output value produced.

Example: State Diagram

- Diagram gets confusing for large circuits
- For small circuits, usually easier to understand than the state table



THANK YOU