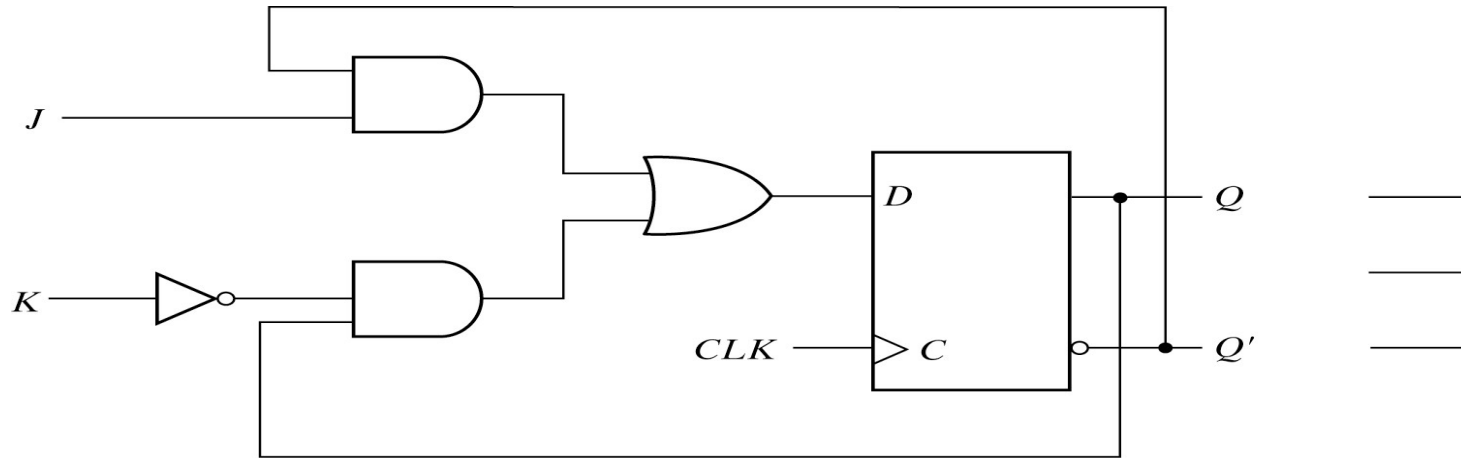


# **Sequential Circuit (JK & T Flip Flop) Characteristic Equations**

## Overview

- Latches respond to trigger **levels** on control inputs
  - **Example: If  $G = 1$ , input reflected at output**
- Difficult to precisely time when to store data with latches
- **Flip flips** store data on a **rising** or **falling** trigger edge.
  - Example: control input transitions from 0 -> 1, data input appears at output
  - Data remains stable in the flip flop until until next rising edge.
- Different types of flip flops serve different functions
- Flip flops can be defined with **characteristic functions**.

## Positive Edge-Triggered J-K Flip-Flop



Created from D flop

J sets

K resets

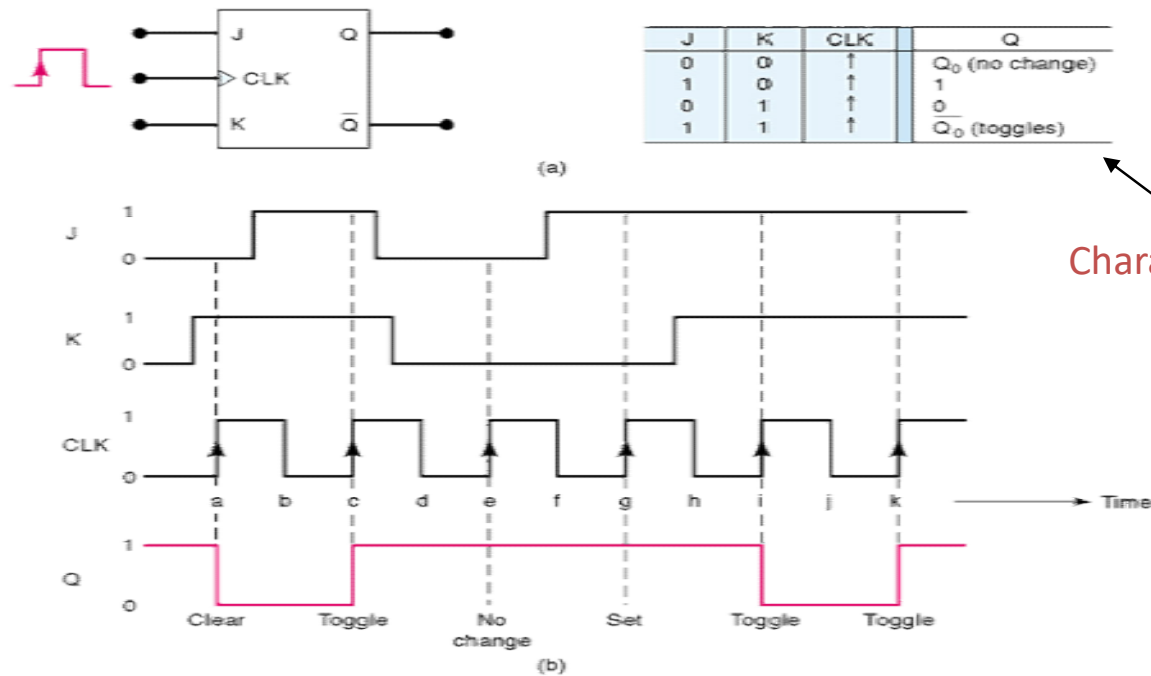
J=K=1 -> invert output

J	K	CLK	Q	Q'
0	0	↑	$Q_0$	$Q'_0$
0	1	↑	0	1
1	0	↑	1	0
1	1	↑	TOGGLE	TOGGLE

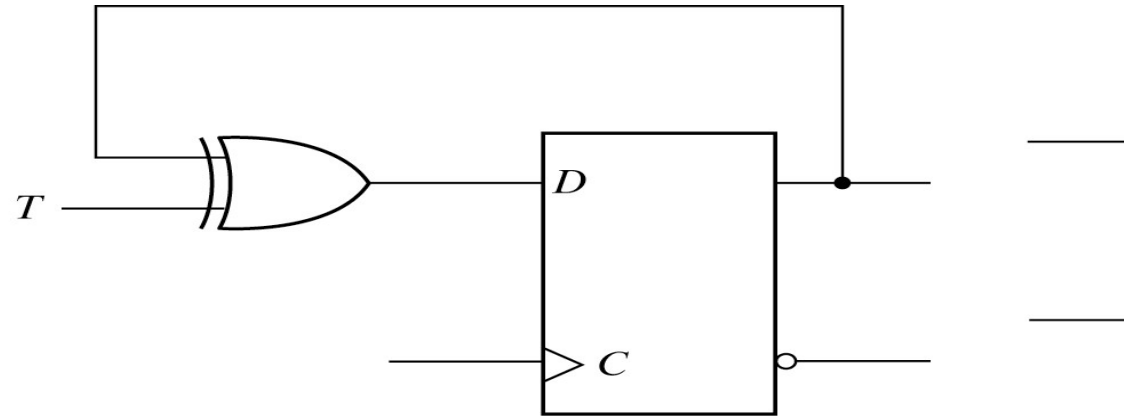
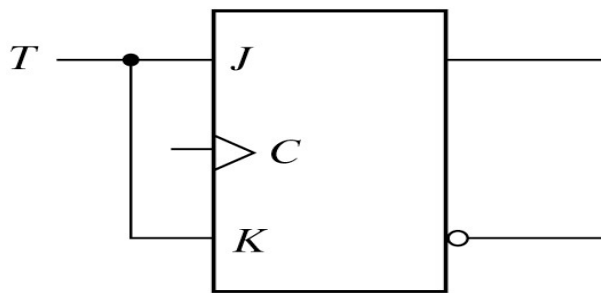
# Clocked J-K Flip Flop

Two data inputs, J and K

J -> set, K -> reset, if J=K=1 then toggle output



## Positive Edge-Triggered T Flip-Flop

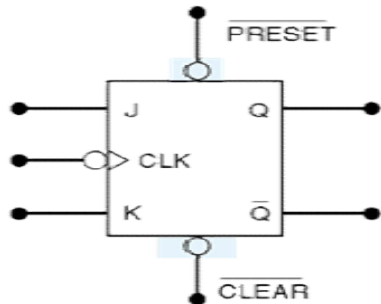


Created from D flop  
 $T=0 \rightarrow$  keep current  
 $K$  resets  
 $T=1 \rightarrow$  invert current

T	C	Q	Q'
0	↑	$Q_0$	$Q_0'$
1	↑	TOGGLE	

## Asynchronous Inputs

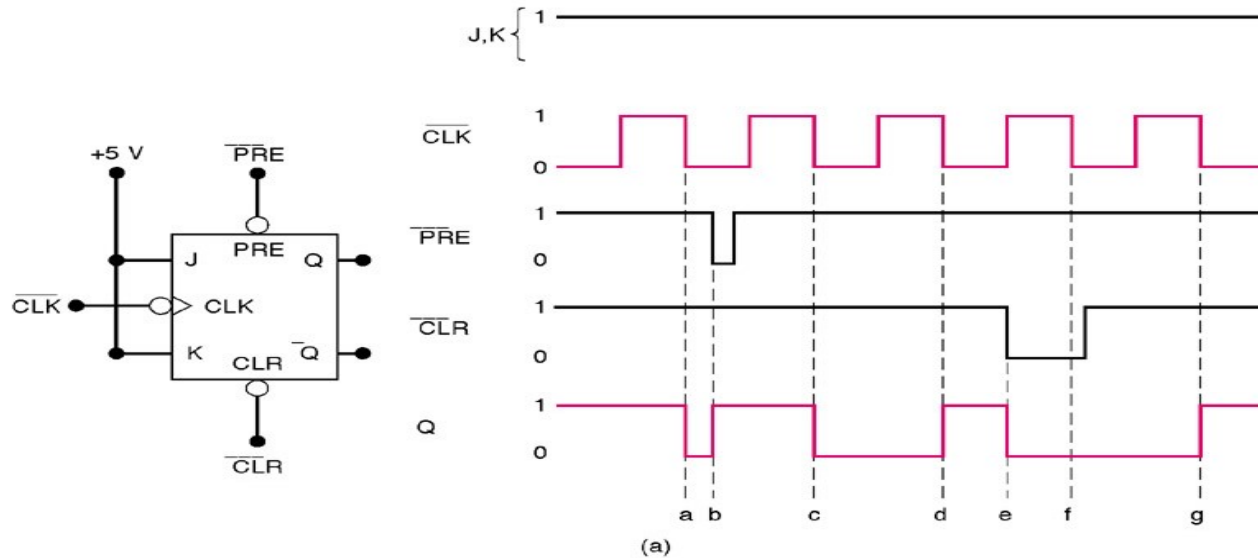
- J, K are synchronous inputs
  - Effects on the output are synchronized with the *CLK* input.
- Asynchronous inputs operate independently of the synchronous inputs and clock
  - Set the FF to 1/0 states *at any time*.



PRESET	CLEAR	FF response
1	1	Clocked operation*
0	1	Q = 1 (regardless of CLK)
1	0	Q = 0 (regardless of CLK)
0	0	Not used

\*Q will respond to J, K, and CLK

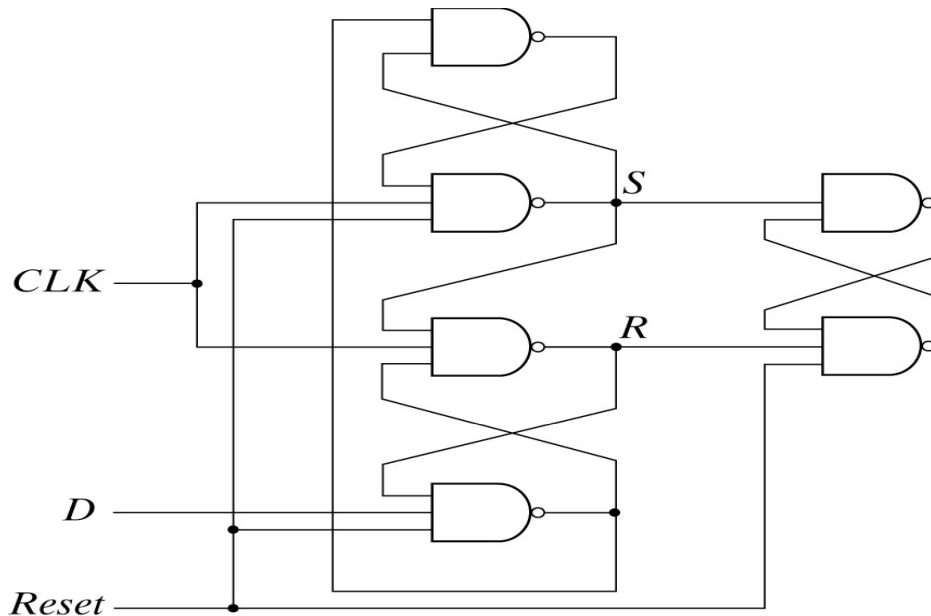
# Asynchronous Inputs



Point	Operation
a	Synchronous toggle on NGT of CLK
b	Asynchronous set on $\overline{PRE} = 0$
c	Synchronous toggle
d	Synchronous toggle
e	Asynchronous clear on $\overline{CLR} = 0$
f	$\overline{CLR}$ over-rides the NGT of CLK
g	Synchronous toggle

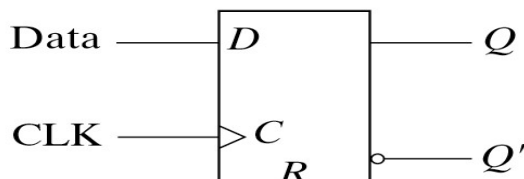
(b)

# Asynchronous Inputs



(a) Circuit diagram

- Note reset signal (R) for D flip flop
- If  $R = 0$ , the output Q is cleared
- This event can occur at any time, regardless of the value of the CLK

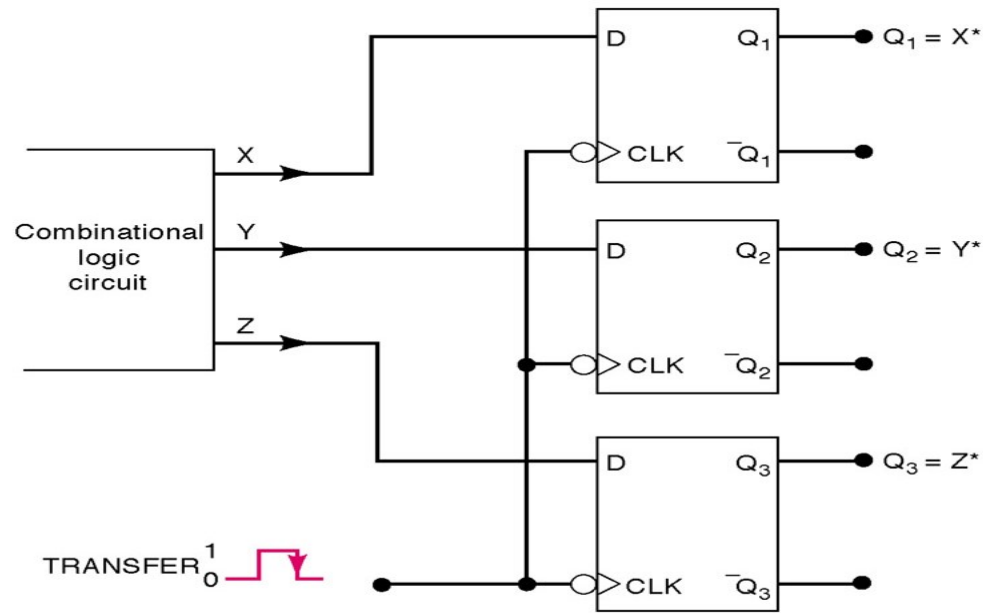


<u>R</u>	<u>C</u>
0	X
1	↑
1	↑



## Parallel Data Transfer

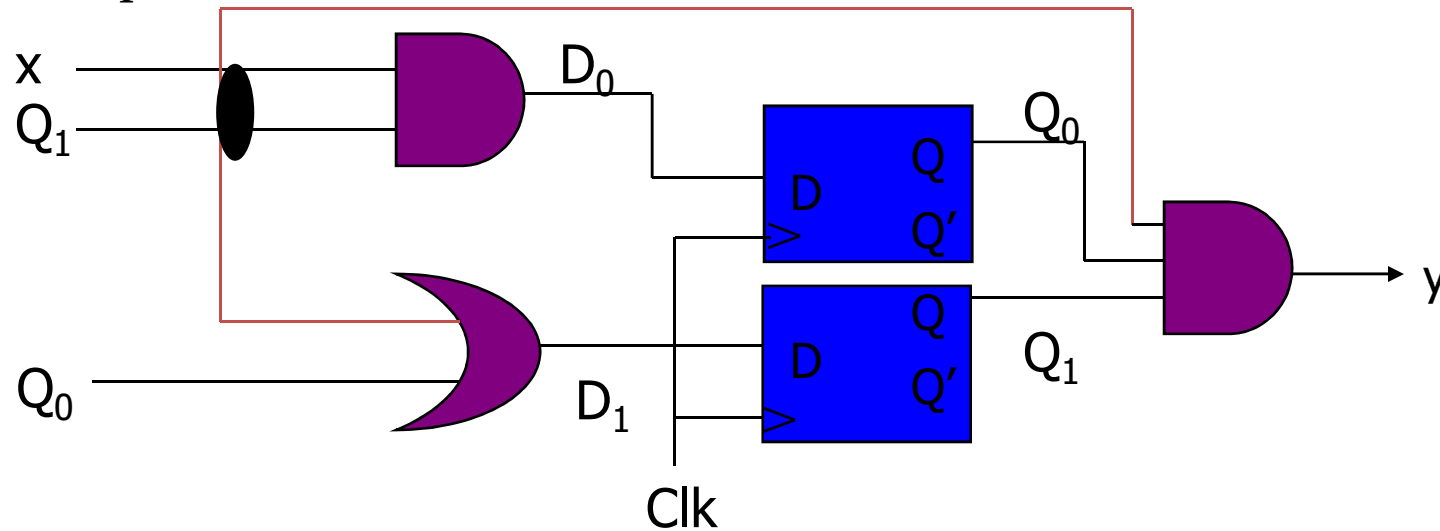
Flip flops store outputs from combinational logic  
Multiple flops can store a collection of data



\*After occurrence of NGT

## Flip Flop State

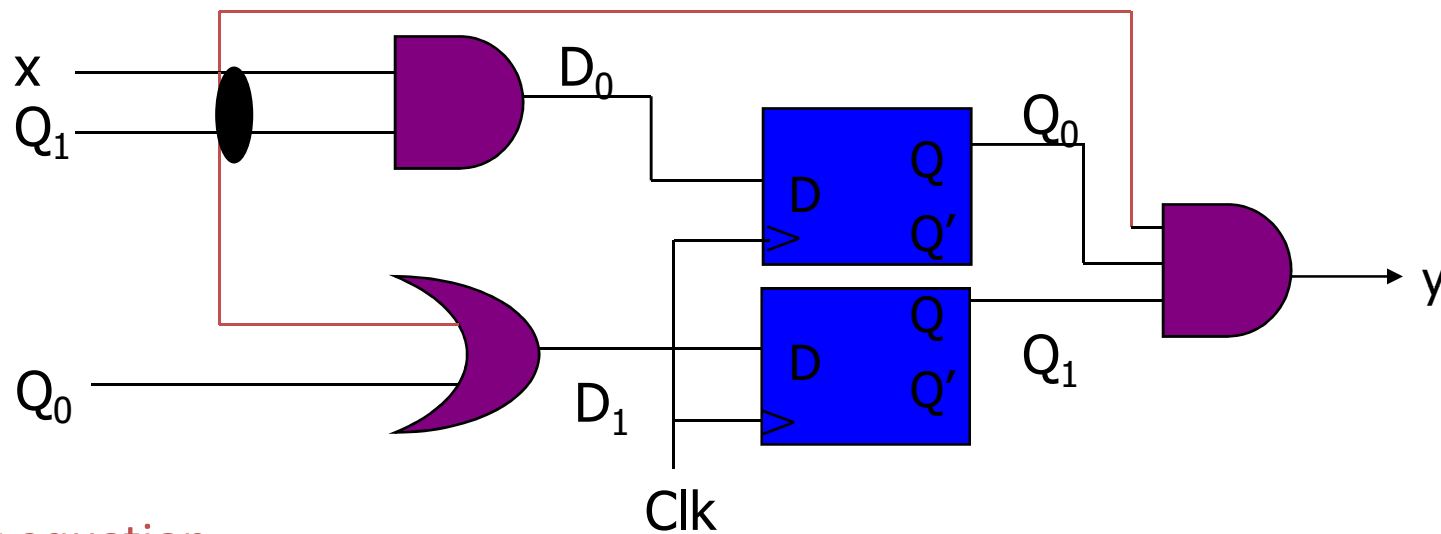
Behavior of clocked sequential circuit can be determined from inputs, outputs and FF state



$$y(t) = x(t)Q_1(t)Q_0(t)$$
$$Q_0(t+1) = D_0(t) = x(t)Q_1(t)$$
$$Q_1(t+1) = D_1(t) = x(t) + Q_0(t)$$

## Output and State Equations

Next state dependent on previous state.



Output equation

$$y(t) = x(t)Q_1(t)Q_0(t)$$

State equations

$$Q_0(t+1) = D_0(t) = x(t)Q_1(t)$$

$$Q_1(t+1) = D_1(t) = x(t) + Q_0(t)$$

## State Table

- Sequence of outputs, inputs, and flip flop states enumerated in state table
- **Present state** indicates current value of flip flops
- **Next state** indicates state after next rising clock edge
- **Output** is output value on current clock edge

State Table

Present State	Next State		Output	
	x=0	x=1	x=0	x=1
0 0	00	10	0	0
0 1	10	10	0	0
1 0	00	11	0	0
1 1	1 0	11	0	1

$Q_1(t) \quad Q_0(t)$        $Q_1(t+1) \quad Q_0(t+1)$

## State Table

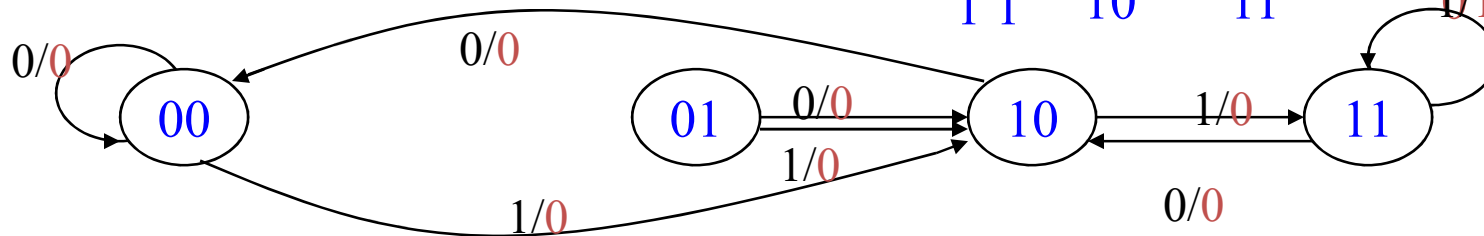
- All possible input combinations enumerated
- All possible state combinations enumerated
- Separate columns for each output value.
- Sometimes easier to designate a symbol for each state.

Let: $s_0 = 00$ $s_1 = 01$ $s_2 = 10$ $s_3 = 11$	Present	Next State		Output	
	State	x=0	x=1	x=0	x=1
	$s_0$	$s_0$	$s_2$	0	0
	$s_1$	$s_2$	$s_2$	0	0
	$s_2$	$s_0$	$s_3$	0	0
	$s_3$	$s_2$	$s_3$	0	1

## State Diagram

- Circles indicate current state
- Arrows point to **next state**
- For  $x/y$ ,  $x$  is input and  $y$  is output

Present State	Next State		Output	
	x=0	x=1	x=0	x=1
0 0	00	10	0	0
0 1	10	10	0	0
1 0	00	11	0	0
1 1	10	11	0	1



## State Diagram

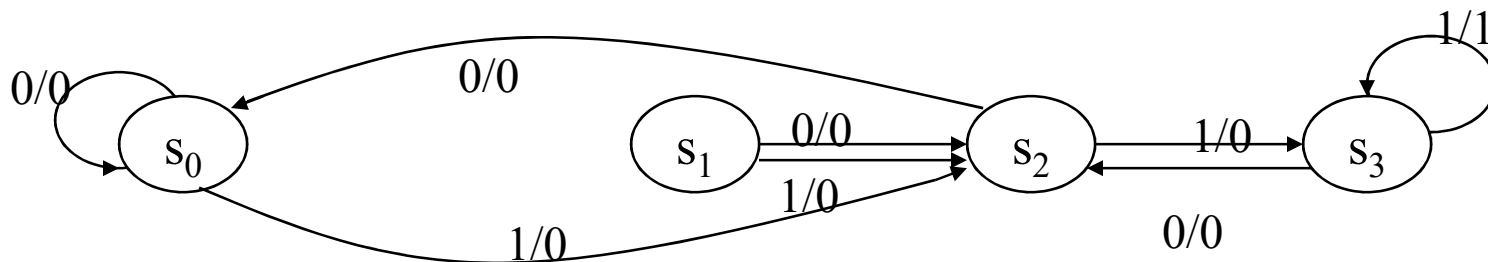
Each state has two arrows leaving

One for  $x = 0$  and one for  $x = 1$

Unlimited arrows can enter a state

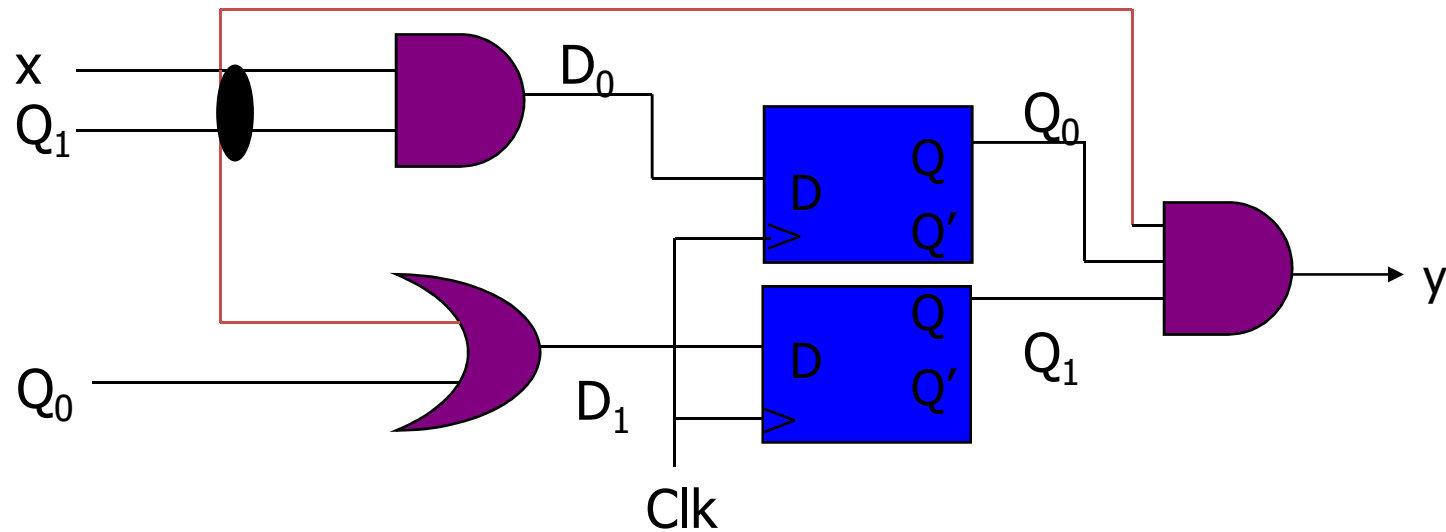
Note use of state names in this example

Easier to identify



## Flip Flop Input Equations

Boolean expressions which indicate the input to the flip flops.



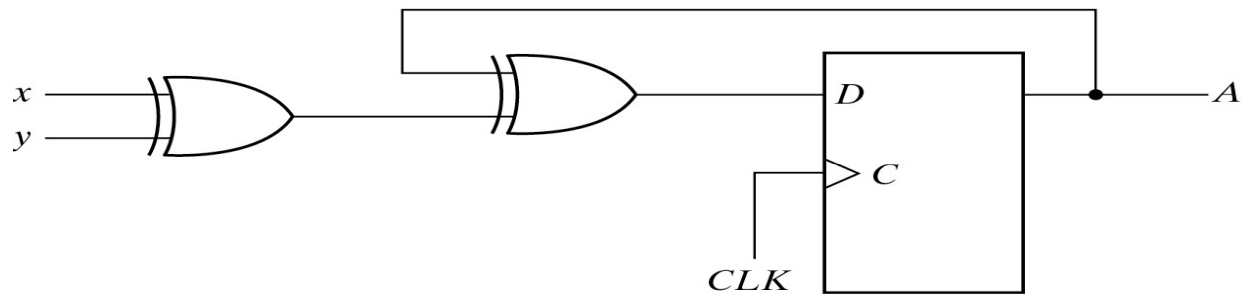
$$D_{Q_0} = xQ_1$$
$$D_{Q_1} = x + Q_0$$

Format implies type of flop used



# Analysis with D Flip-Flops

Identify flip flop input equations  
Identify output equation



(a) Circuit diagram



## Summary

- Flip flops contain state information
- State can be represented in several forms:
  - State equations
  - State table
  - State diagram
- Possible to convert between these forms

Cont..

- Circuits with state can take on a finite set of values
  - Finite state machine
- Two types of “machines”
  - Mealy machine
  - Moore machine

THANK YOU

## Summary

- Flip flops are powerful storage elements
  - They can be constructed from gates and latches!
- D flip flop is simplest and most widely used
- Asynchronous inputs allow for clearing and presetting the flip flop output
- Multiple flops allow for data storage
  - The basis of computer memory!
- Combine storage and logic to make a computation circuit