Synchronous Sequential Logic

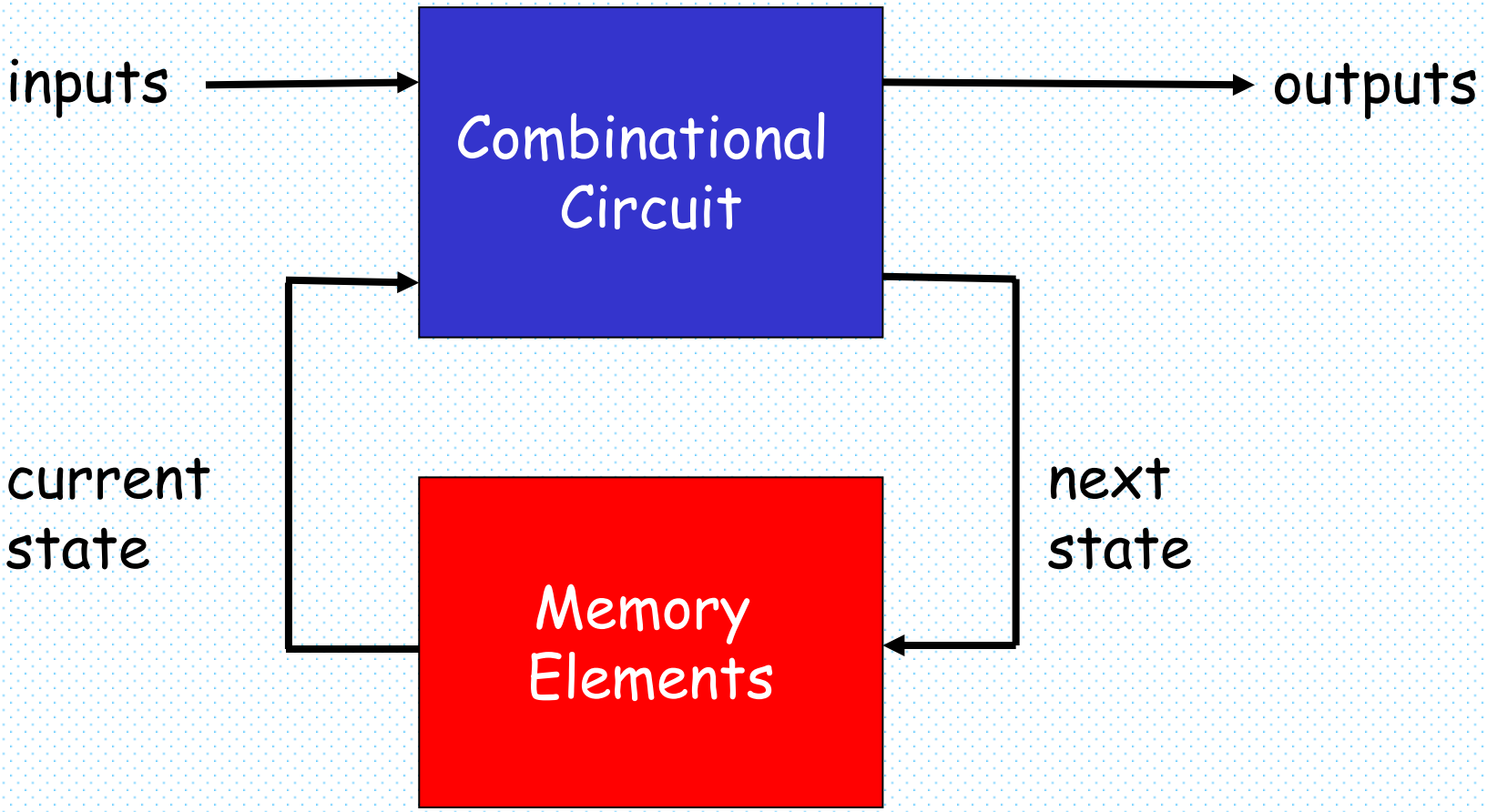
Logic and Digital System Design - CS 303

Erkay Savaş
Sabanci University

Sequential Logic

- Digital circuits we have learned, so far, have been combinational
  - no memory,
  - outputs are entirely defined by the "current" inputs
- However, many digital systems encountered everyday life are sequential (i.e. they have memory)
  - the memory elements remember past inputs
  - outputs of sequential circuits are not only dependent on the current input but also the state of the memory elements.

Sequential Circuits Model



current state is a function of past inputs

Classification - 1

- Two types of sequential circuits

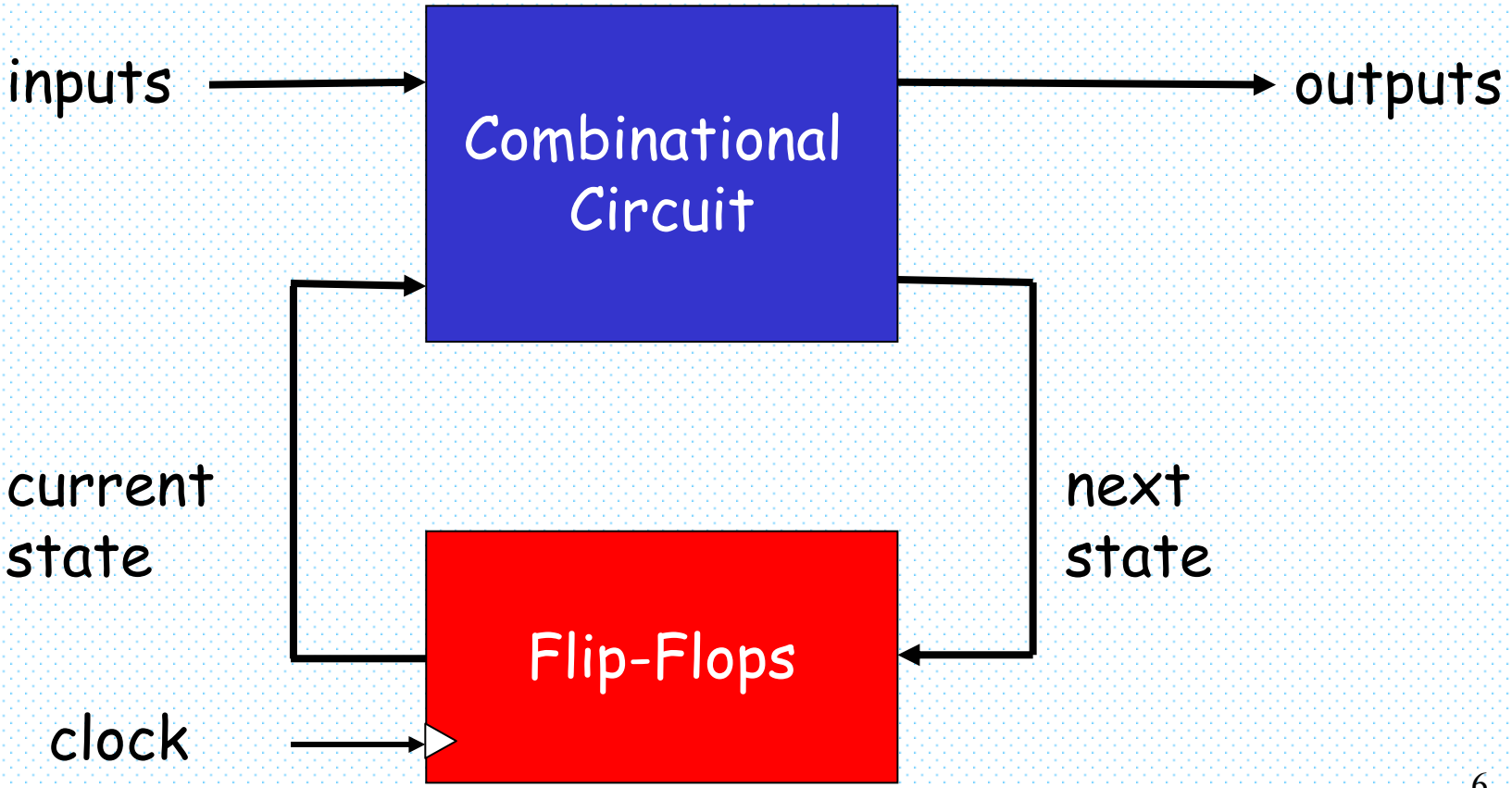
1. Synchronous
   - Signals affect the memory elements at discrete instants of time.
   - Discrete instants of time requires synchronization.
   - Synchronization is usually achieved through the use of a common clock.
   - A "clock generator" is a device that generates a periodic train of pulses.

Classification - 2

1. Synchronous
   - The state of the memory elements are updated with the arrival of each pulse
   - This type of logical circuit is also known as clocked sequential circuits.
2. Asynchronous
   - No clock
   - behavior of an asynchronous sequential circuits depends upon the input signals at any instant of time and the order in which the inputs change.
   - Memory elements in asynchronous circuits are regarded as time-delay elements

Clocked Sequential Circuits

- Memory elements are flip-flops which are logic devices capable of storing one bit of information each.



Clocked Sequential Circuits

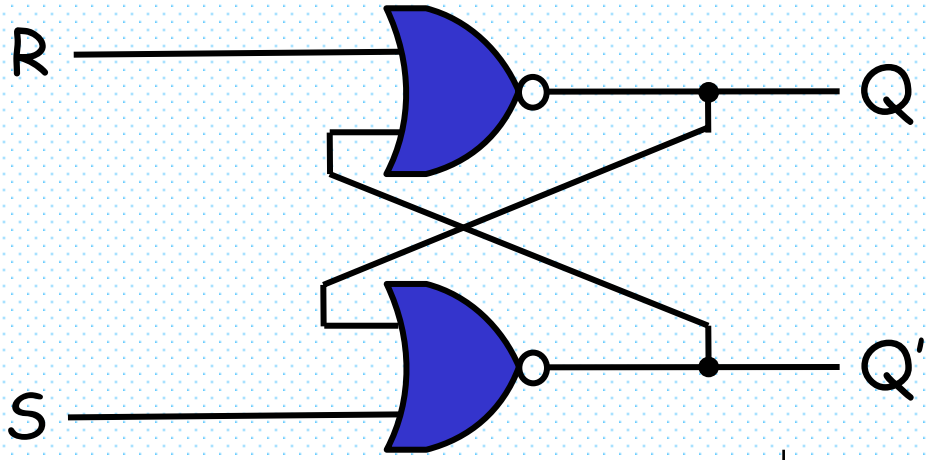
- The outputs of a clocked sequential circuit can come from the combinational circuit, from the outputs of the flip-flops or both.
- The state of the flip-flops can change only during a clock pulse transition
  - i.e. low-to-high and high-to-low
  - clock edge
- When the clock is at logic-0, the flip-flop output does not change
- The transition from one state to the other occurs at the clock edge.

Latches

- The most basic types of memory elements are not flip-flops, but latches.
- A latch is a memory device that can maintain a binary state indefinitely.
- Latches are, in fact, asynchronous devices and they usually do not require a clock to operate.
- Therefore, they are not directly used in clocked synchronous sequential circuits.
- They rather be used to construct flip-flops.

SR-Latch

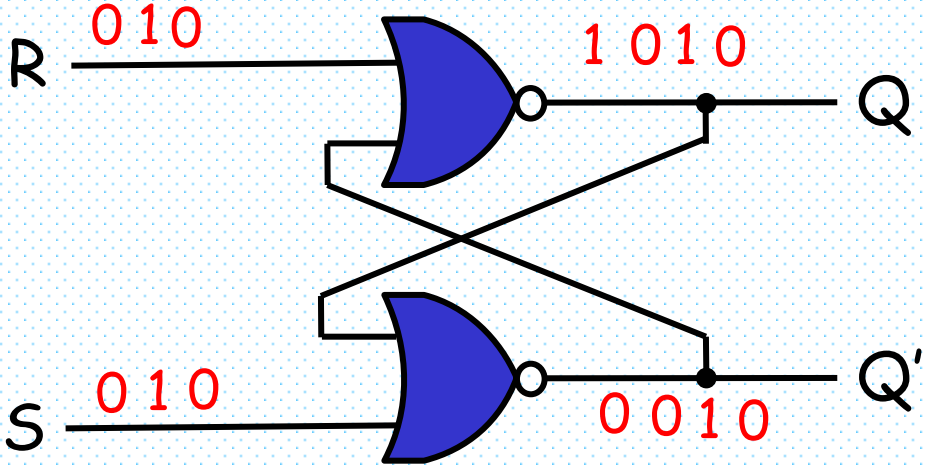
- A circuit is made of cross-coupled NOR (or NAND) gates



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	Undefined

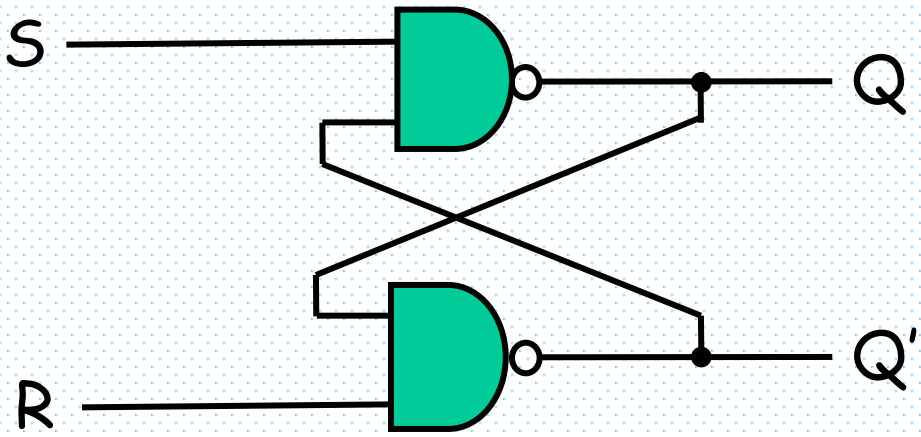
Undefined State of SR-Latch

- S = R = 1 may result in an undefined state
  - the reason being is that the next state is unpredictable when both S and R goes to 0 at the same time.
  - It may oscillate
  - Or the outcome state will depend on which of S and R goes to 0 first.



it oscillates

SR-Latches with NAND Gates

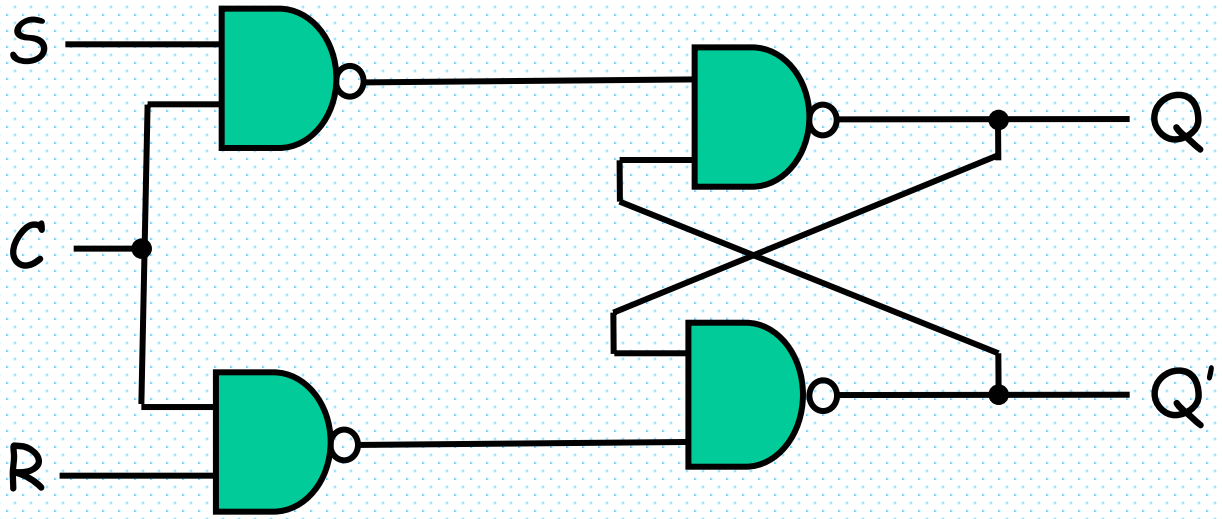


Also known as S'R'-latch

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	Undefined

SR-Latch with Control Input

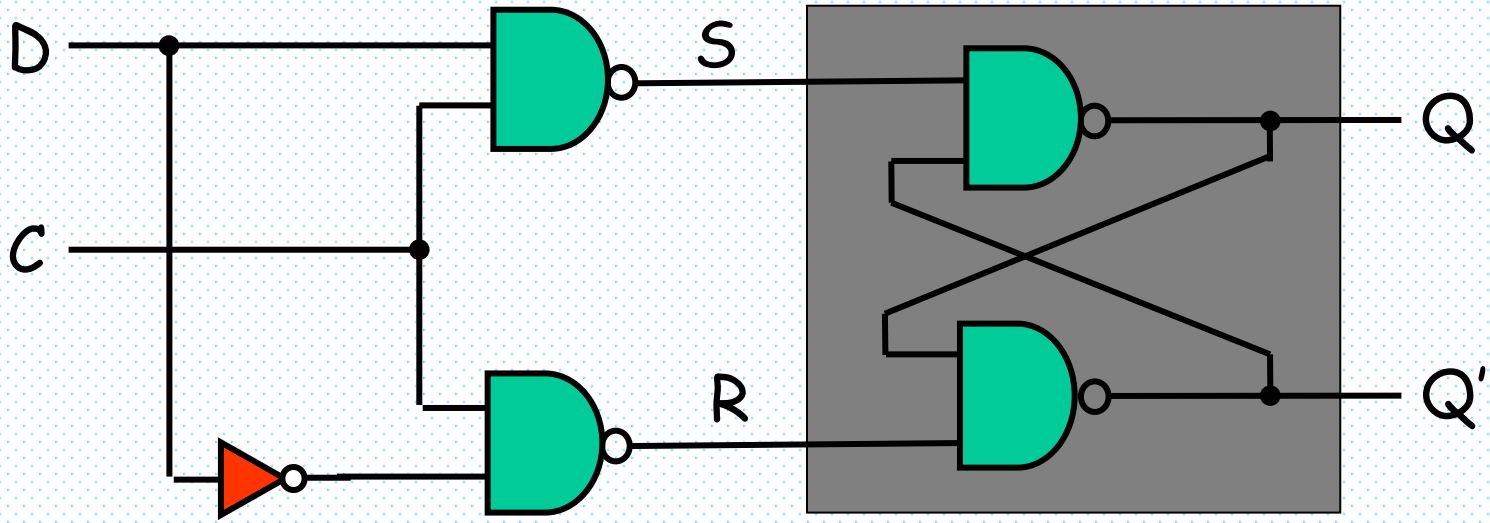
- Control inputs allow the changes at S and R to change the state of the latch.



C	S	R	Q	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	

D-Latch

- SR latches are seldom used in practice because the indeterminate state may cause instability
- Remedy: D-latches

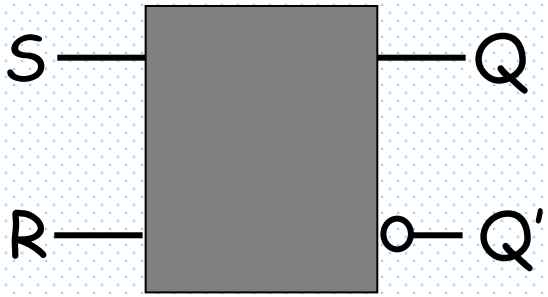


This circuit guarantees that the inputs to the SR-latch is always complement of each other when $C = 1$.

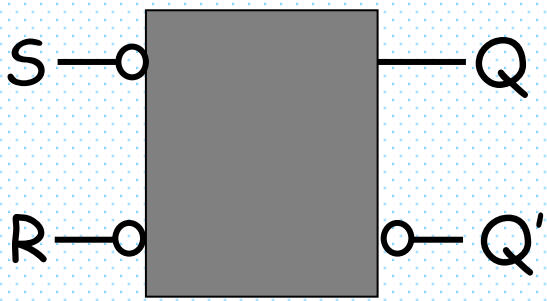
D-Latch

C	D	Next state of Q
0	X	No change
1	0	Q = 0; reset state
1	1	Q = 1; set state

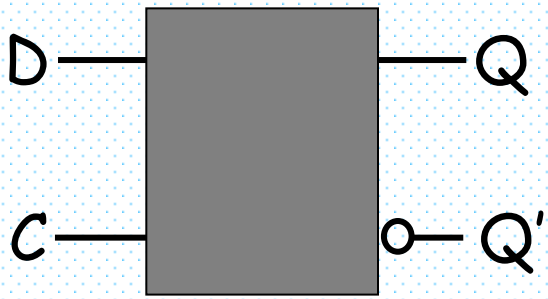
- We say that the D input is sampled when C = 1



SR-latch



S'R'-latch



D-latch

D-Latch as a Storage Unit

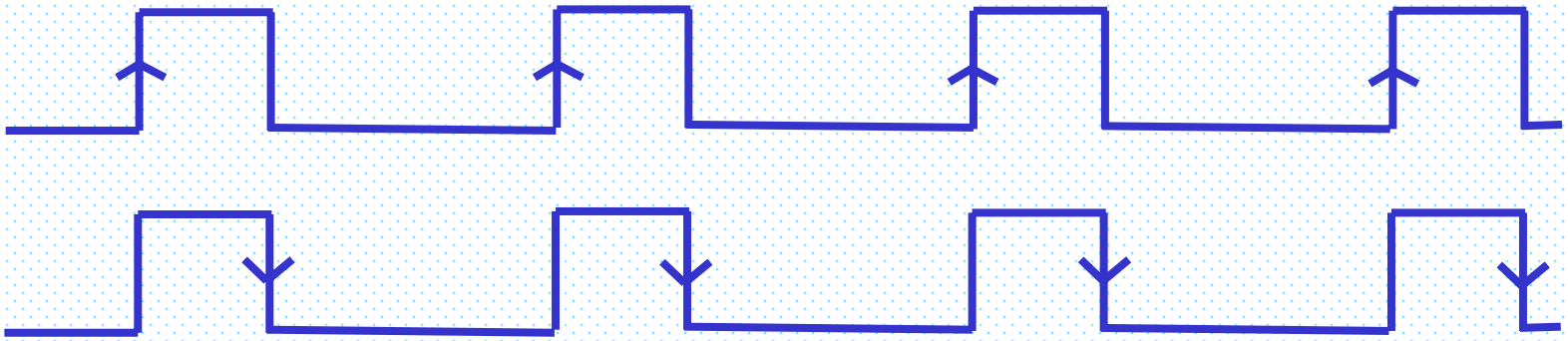
- D-latches can be used as temporary storage
- The input of D-latch is transferred to the Q output when *C* = 1
- When *C* = 0 the binary information is retained.
- We call latches level-sensitive devices.
  - So long as *C* remains at logic-1 level, any change in data input will change the state and the output of the latch.
  - Level sensitive latches may suffer from a serious problem.
- A memory devices that are sensitive to the rising or falling edge of control input is called flip-flops.

Need for Flip-Flops - 1

- Remember in a latch
  - there is a feedback path from the output to the input of the circuit.
  - When control input remains at logic-1 for a period of time
    - The state transition occurs as soon as C becomes 1
    - The new state appears at the outputs of the latches
    - This output is connected to the input
    - Since the input changes again, the state of the latches may change again
    - This may lead to a situation where the state of latches keeps changing so long as C = 1
  - This is why we need edge-sensitive devices.

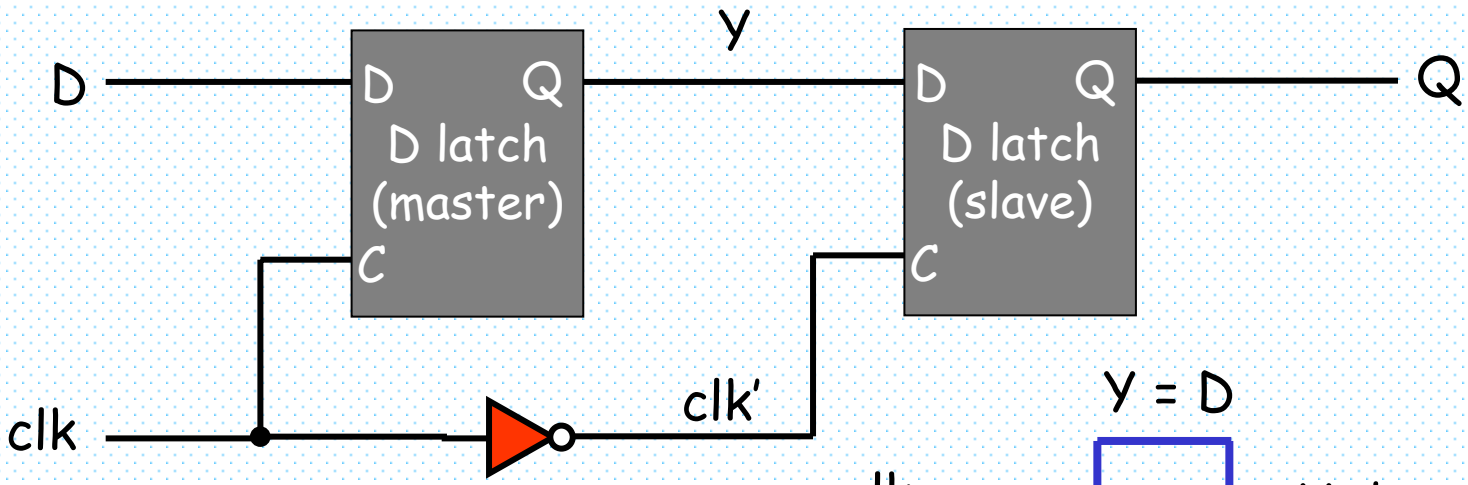
Need for Flip-Flops - 1

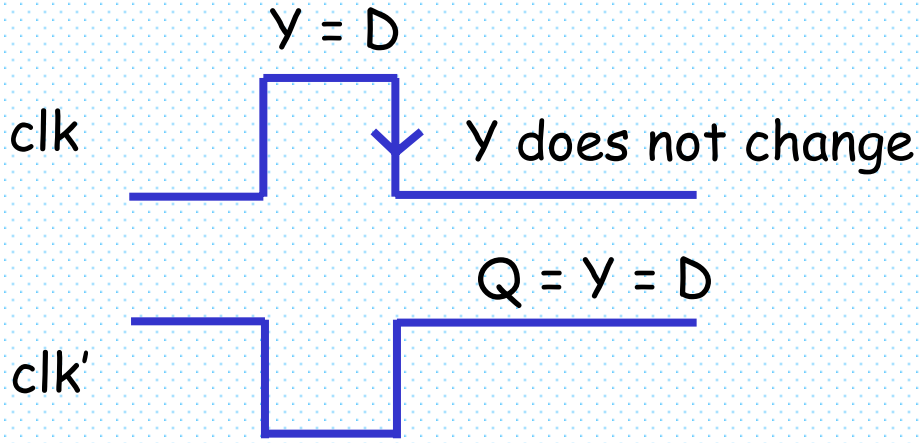
- Another problem
  - We want the states of the memory elements to change synchronously
  - What we need is memory elements that should respond to the changes in input at certain points in time.
  - This is the very characteristics of synchronous circuits.
  - To this end, we use flip-flops that change states during a signal transition of control input (clock)



Edge-Triggered D Flip-Flop

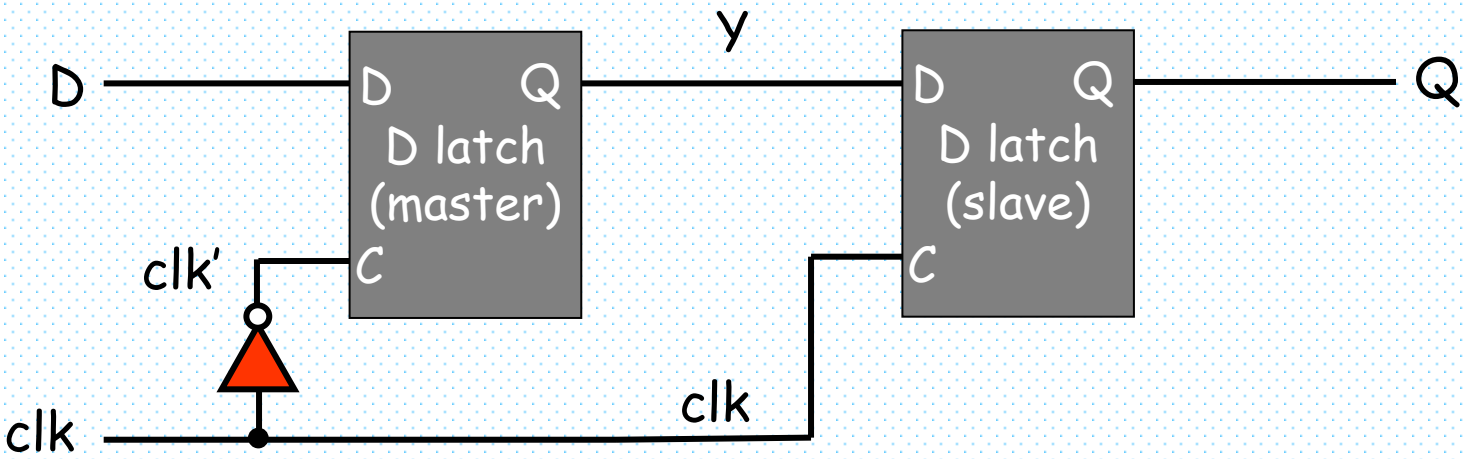
- An edge-triggered D flip-flop can be constructed using two D latches

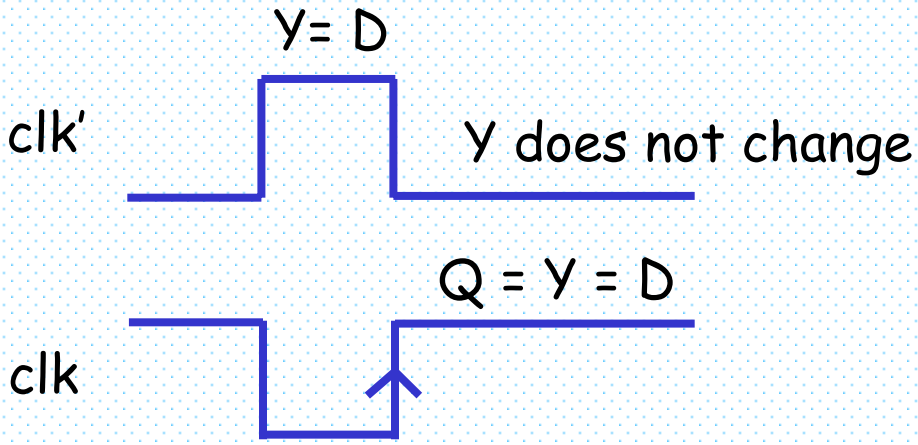




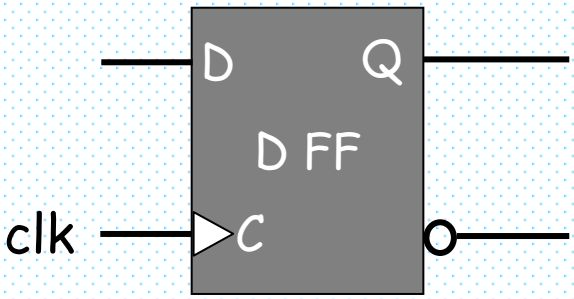
Negative edge-triggered D flip-flop

Positive Edge-Triggered D Flip-Flop

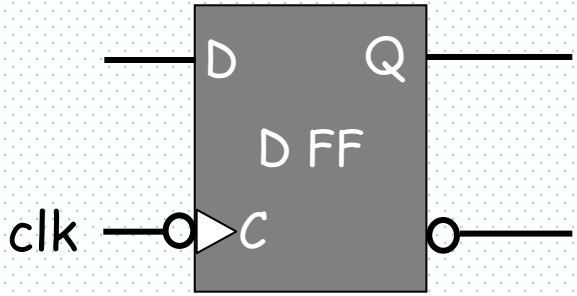




Symbols for D Flip-Flops



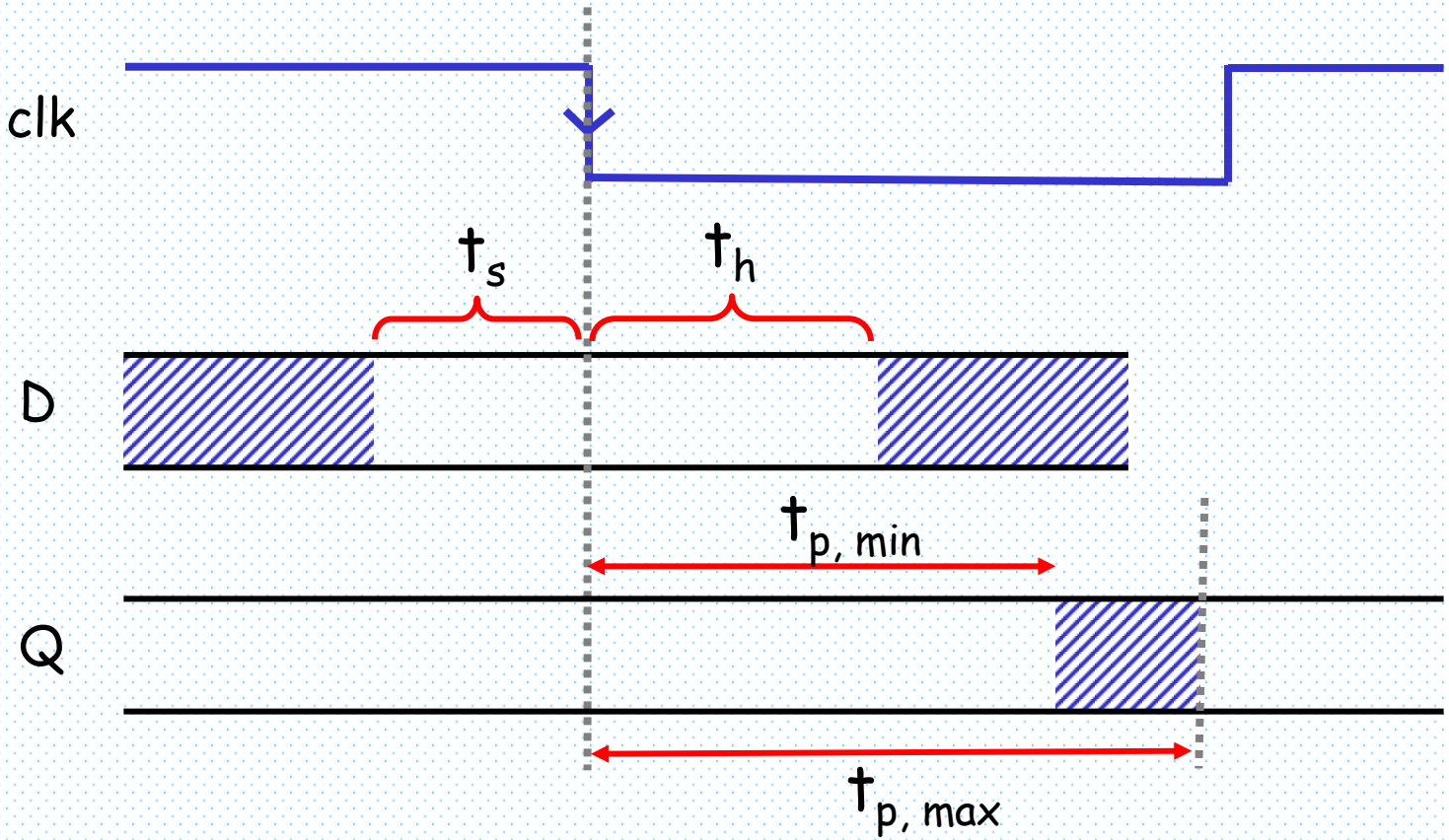
Positive edge-triggered
D Flip-Flop



Negative edge-triggered
D Flip-Flop

Setup & Hold Times - 1

- Timing parameters are associated with the operation of flip-flops
- Recall Q gets the value of D in clock transition



Setup & Hold Times - 2

- Setup time, t_s
  - The change in the input D must be made before the clock transition.
  - Input D must maintain this new value for a certain minimum amount time.
  - If a change occurs at D less than t_s second before the clock transition, then the output may not acquire this new value.
  - It may even demonstrate unstable behavior.
- Hold time, t_h,
  - Similarly the value at D must be maintained for a minimum amount of time (i.e. t_h) after the clock transition.

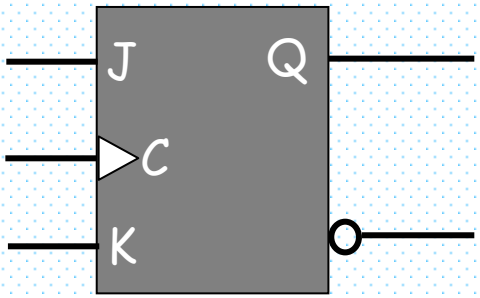
Propagation Time

- Even if setup and hold times are achieved, it takes some time the circuit to propagate the input value to the output.
- This is because of the fact that flip-flops are made of logic gates that have certain propagation times.

Other Flip-Flops

- D flip-flop is the most common
  - since it requires the fewest number of gates to construct.
- Two other widely used flip-flops
  - JK flip-flops
  - T flip-flops
- JK flip-flops
  - Three FF operations
    1. Set
    2. Reset
    3. Complement

JK Flip-Flops



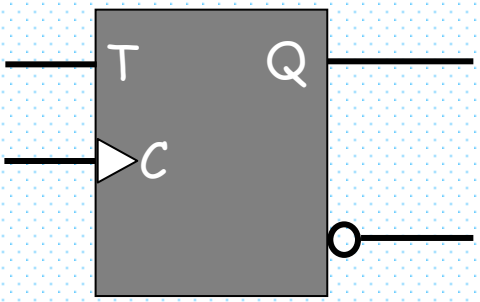
J	K	Q(t+1)	next state
0	0	Q(t)	no change
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	Complement

Characteristic Table

- Characteristic equation
  - Q(t+1) = JQ'(t) + K'Q(t)

T (Toggle) Flip-Flop

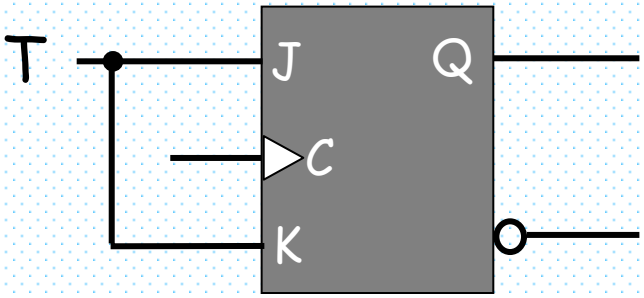
- Complementing flip-flop

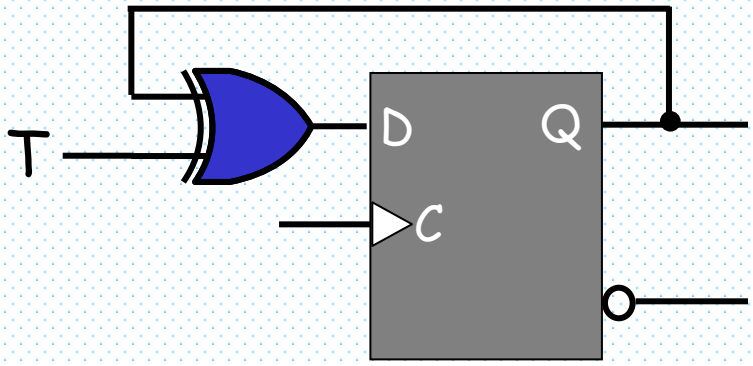


T	Q(t+1)	next state
0	Q(t)	no change
1	Q'(t)	Complement

Characteristic Table

- Characteristic equation
  - $Q(t+1) = T \oplus Q(t) = TQ'(t) + T'Q(t)$





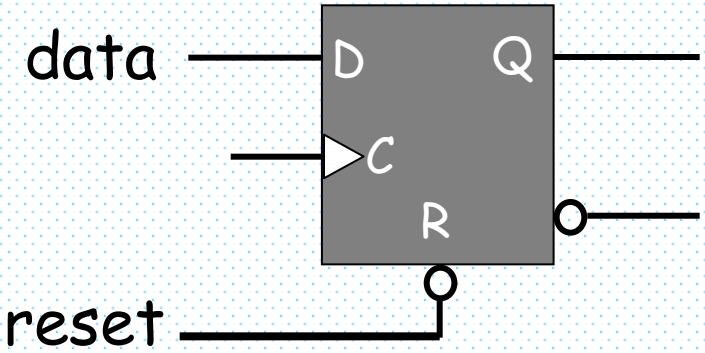
Characteristic Equations

- The logical properties of a flip-flop can be expressed algebraically using characteristic equations
- D flip-flop
  - $Q(t+1) = D$
- JK flip-flop
  - $Q(t+1) = JQ'(t) + K'Q(t)$
- T flip-flop
  - $Q(t+1) = T \oplus Q(t)$

Asynchronous Inputs of Flip-Flops

- They are used to force the flip-flop to a particular state independent of clock
  - "Preset" (direct set) set FF state to 1
  - "Clear" (direct reset) set FF state to 0
- They are especially useful at startup.
  - In digital circuits when the power is turned on, the state of flip-flops are unknown.
  - Asynchronous inputs are used to bring all flip-flops to a known "starting" state prior to clock operation.

Asynchronous Inputs



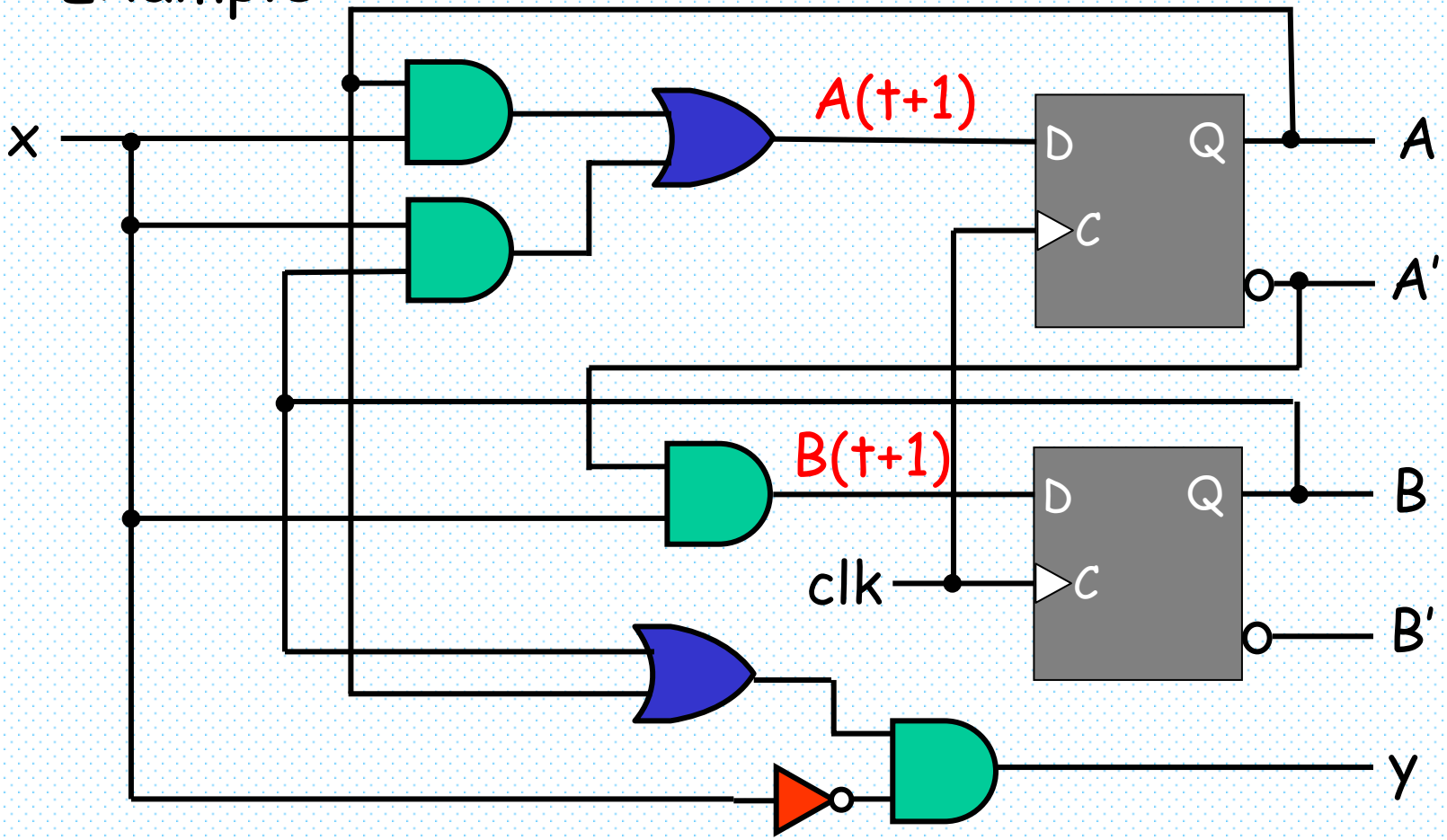
R	C	D	Q	Q′	
0	X	X	0	1	Starting State
1	↑	0	0	1	
1	↑	1	1	0	

Analysis of Clocked Sequential Circuits

- Goal:
  - to determine the behavior of clocked sequential circuits
  - "Behavior" is determined from
    - Inputs
    - Outputs
    - State of the flip-flops
  - We have to obtain
    - (state) table
    - (state) diagram
    - Boolean expression for next state and output
  - They must include time sequence information

State Equations

- Also known as "transition equations"
  - specify the next state as a function of the present state and inputs
- Example



Example: Output and State Equations

- Flip-Flop input (excitation) equations
  - Same as state equations in D flip-flops
  - $A(t+1) = x\,A(t) + x\,B(t)$
    $= xA + xB$
  - $B(t+1) = x\,A'(t)$
    $= xA'$
- Output equation
  - $y(t) = [A(t) + B(t)]\,x'$
    $= (A + B)\,x'$

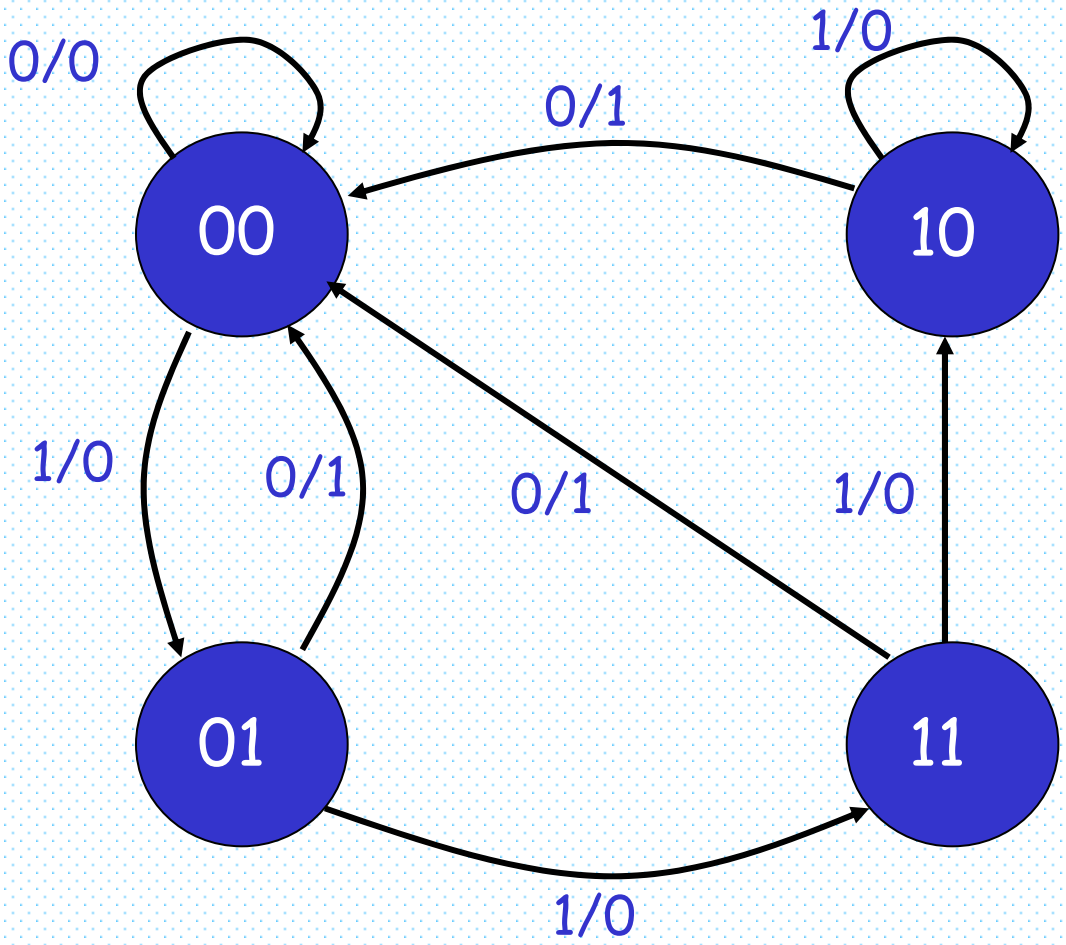
Example: State (Transition) Table

$A(t+1) = xA + xB$ $B(t+1) = xA'$ $y = (A + B)x'$

Present state		input	Next state		output
A	B	x	A	B	y
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

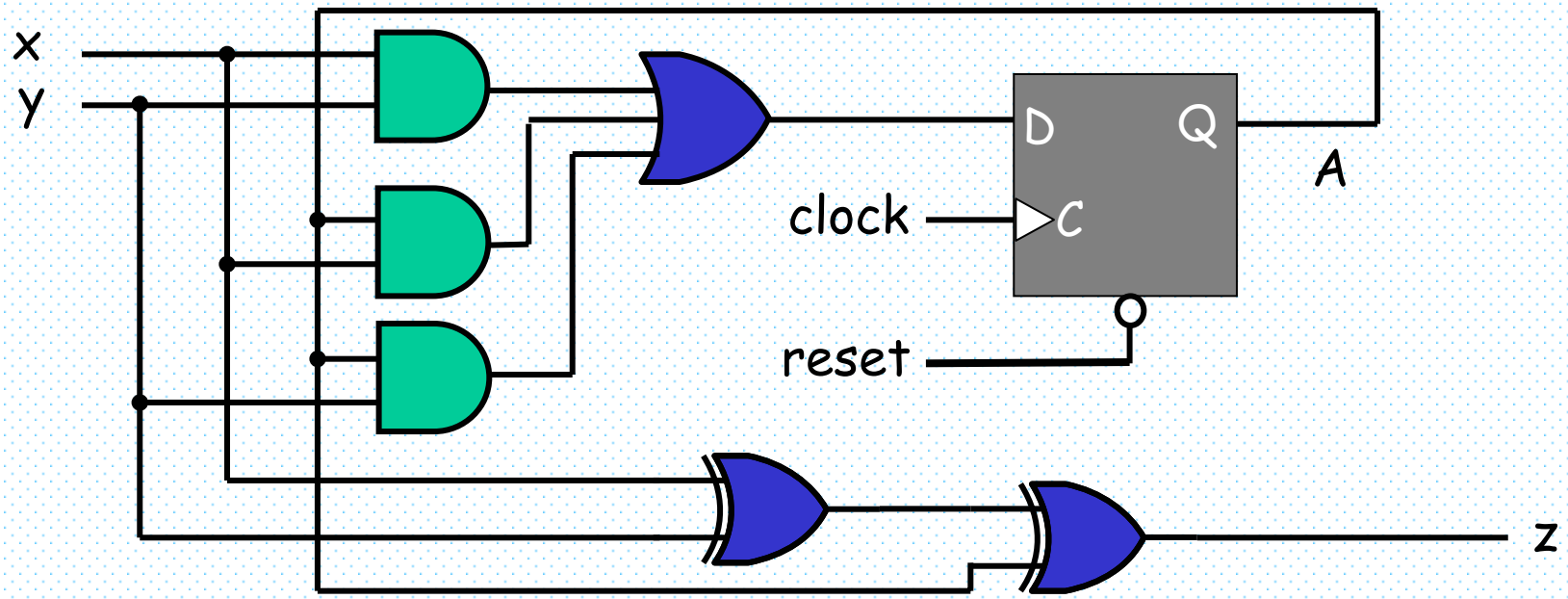
A sequential circuit with m FFs and n inputs needs 2^{m+n} rows in the transition table

Example: State Diagram



State diagram provides the same information as state table

Analysis with D Flip-Flops - 1



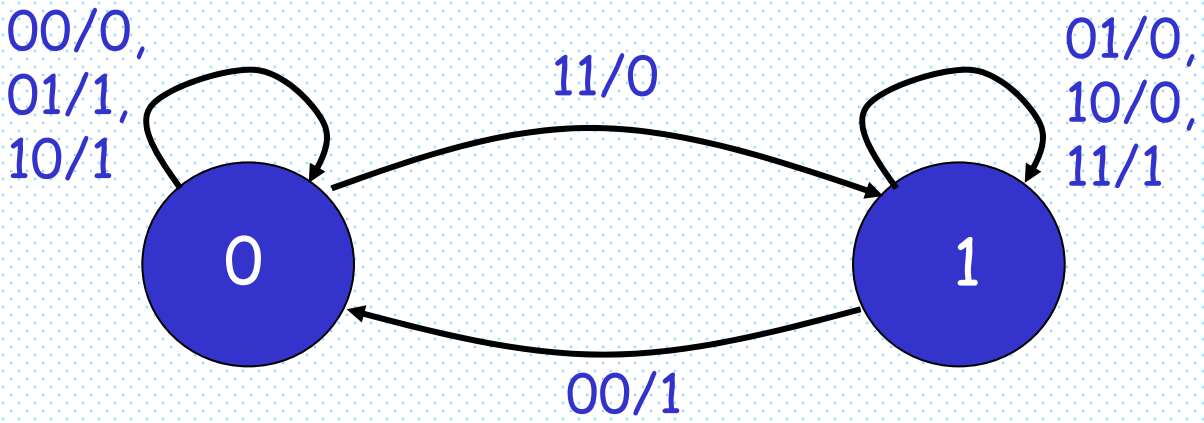
- Flip-Flop input equation
  - $D_A = xy + xA + yA$
  - $Q(t+1) = D_A$
- Output equation
  - $z = x \oplus y \oplus A$

Analysis with D Flip-Flops - 2

State Table

Present state	Inputs		next state	output
A	x	y	A	z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Analysis with D Flip-Flops - 3

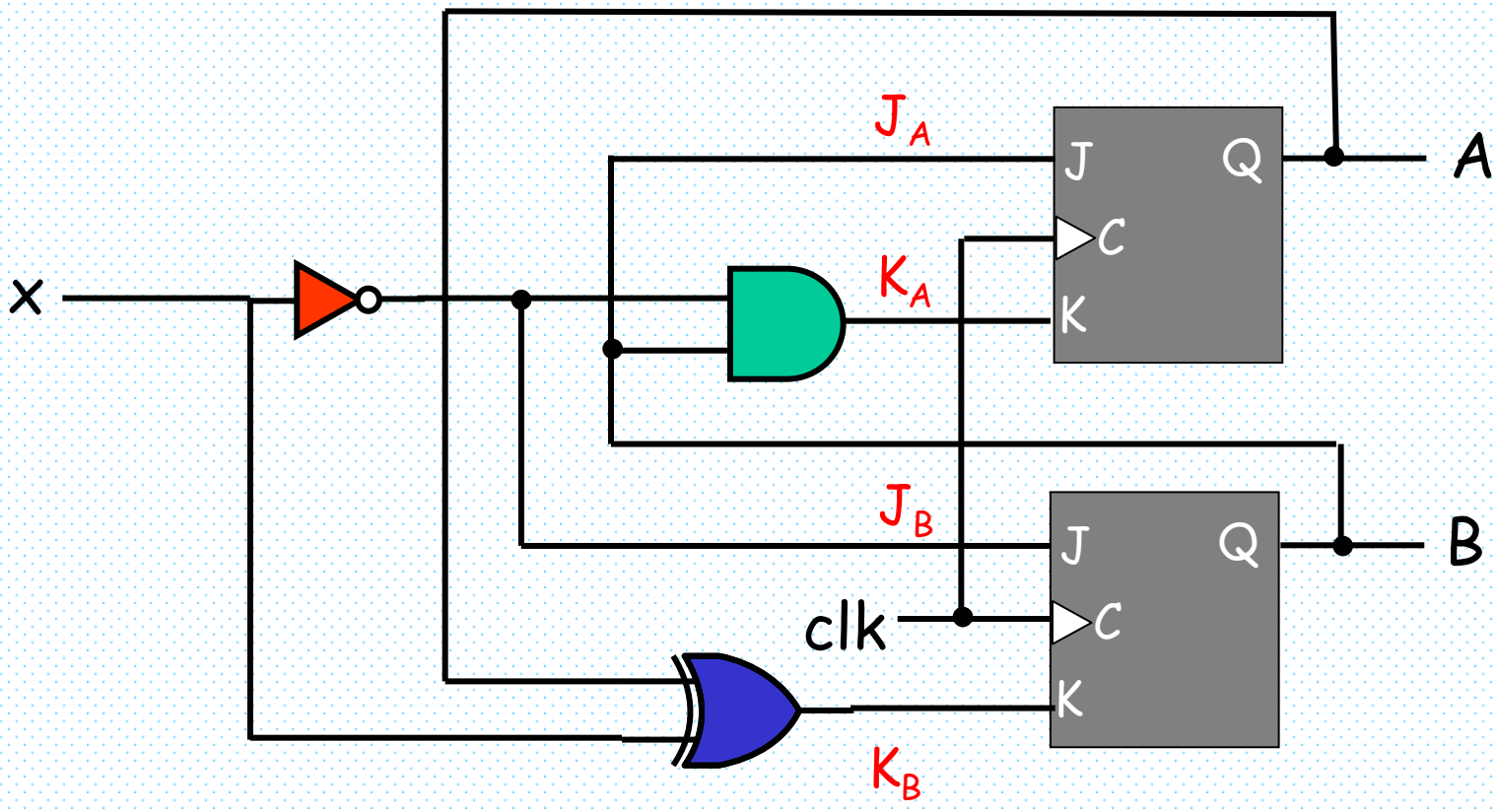


It is a serial adder.

Analysis with JK Flip-Flops

- For a D flip-flop, the state equation is the same as the flip-flop input equation
  - $Q(t+1) = D_Q$
- For JK flip-flops, situation is different
  - Goal is to find state equations
  - Method
    1. determine flip-flop input equations
    2. List the binary values of each input equation
    3. Use the corresponding flip-flop characteristic table to determine the next state values in the state table

Example: Analysis with JK FFs



- Flip-flop input equations
  - $J_A = B$ and $K_A = x'B$
  - $J_B = x'$ and $K_B = x \oplus A$

Example: Analysis with JK FFs

- $J_A = B$ and $K_A = x'B$
- $J_B = x'$ and $K_B = x \oplus A$

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

present State		input	next state		FF inputs			
A	B	x	A	B	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

Example: Analysis with JK FFs

- Characteristic equations
  - $A(t+1) = J_A A' + K_A' A$
  - $B(t+1) = J_B B' + K_B' B$
- Input equations
  - $J_A = B$ and $K_A = x'B$
  - $J_B = x'$ and $K_B = x \oplus A$
- State equations
  - $A(t+1) = BA' + (x'B)'A$
    $= BA' + (x + B')A = BA' + AB' + Ax$
  - $B(t+1) = x'B' + (x \oplus A)'B$
    $= x'B' + xAB + x'A'B$

State Diagram