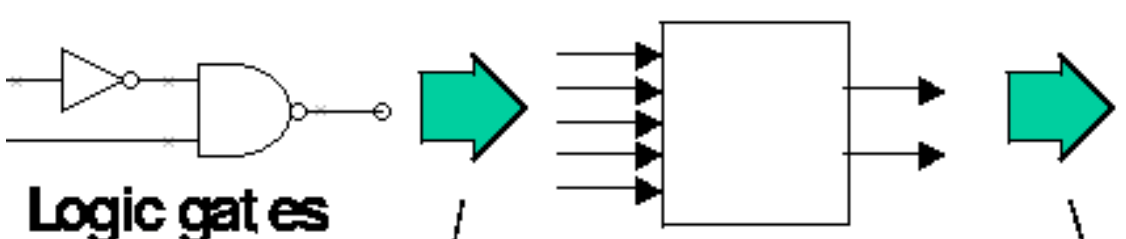
Algorithm = Logic + Control

Sequential Circuits Problems



Sequential Logic

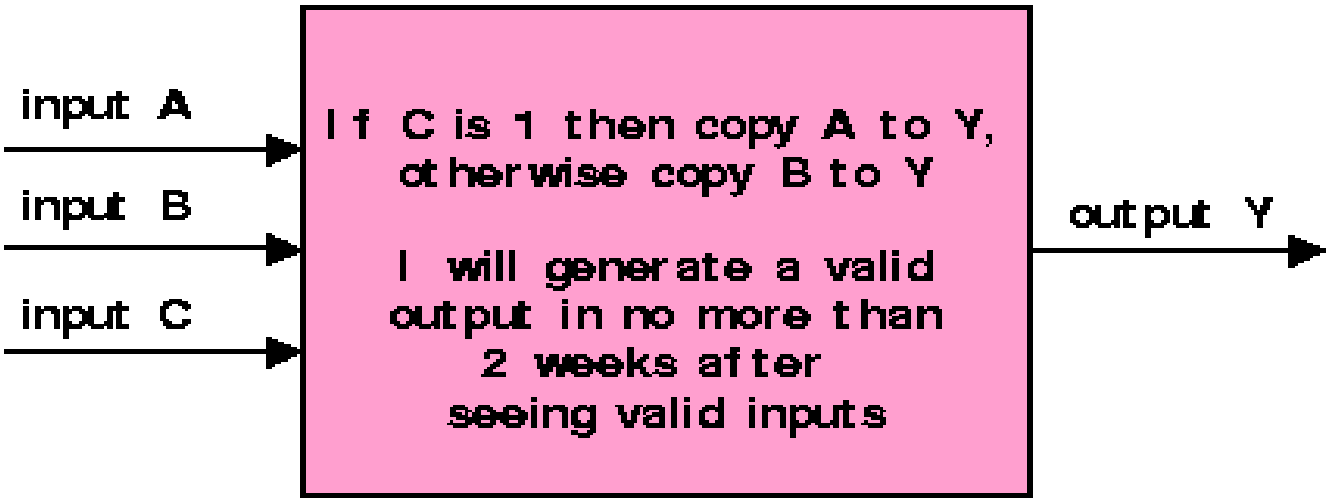
Acyclic connections
Composable blocks
Design:

- truth tables
- sum-of-products
- simplification
- muxes, ROMs, PLAs

Storage & state
Dynamic discipline
Finite-state machines
Metastability
Throughput & latency
Pipelining

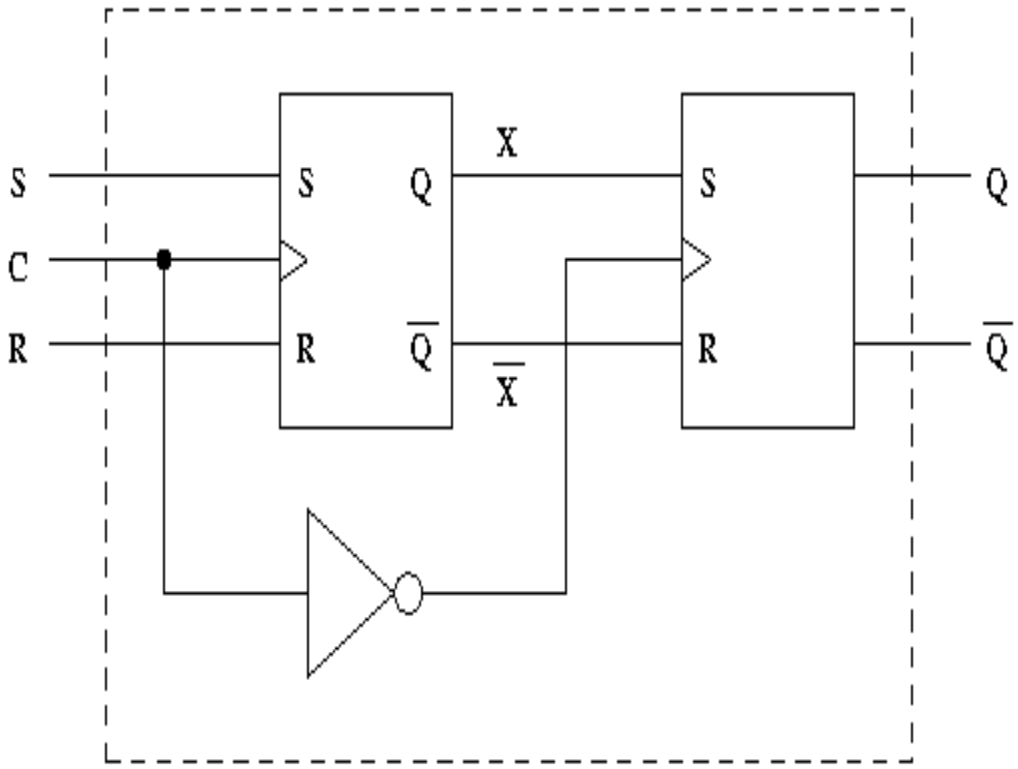
- A *combinational device* is a circuit element that has
  - one or more digital *inputs*
  - one or more digital *outputs*
  - a *functional specification* that details the value of each output for every possible combination of valid input values
  - a *timing specification* consisting (at minimum) of an upper bound t_{pd} on the required time for the device to compute the specified output values from an arbitrary set of stable, valid input values

Static discipline



SR Master-Slave Flip-Flop

The SR Master-Slave Flip-Flop is constructed from two SR latches and an inverter

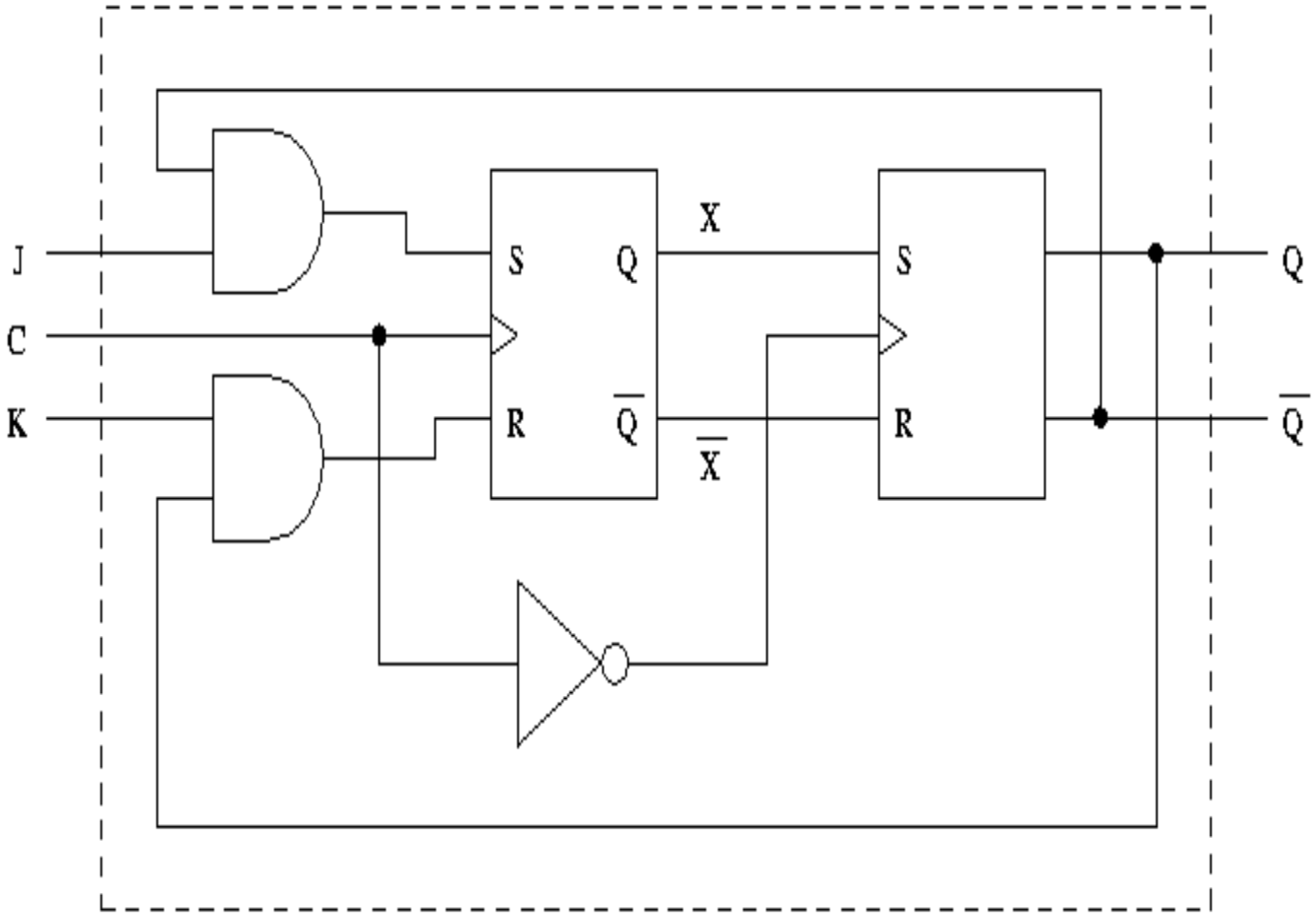


The left hand latch is called the *master* and the right hand latch the *slave*.

When the enable line goes high, the inputs to the circuit are passed through the master latch to its outputs (X and $\overline{X}$). They are not passed through the slave, though, as when the enable line is high, the enable input to the slave latch is low. Once the enable line goes low, the enable input to the slave latch then goes high and the values of X and $\overline{X}$ are transferred to the outputs from the circuit and the state is changed.

JK Flip-Flop

The *JK flip-flop* is constructed from and is similar in operation to the SR flip-flop. The circuit is



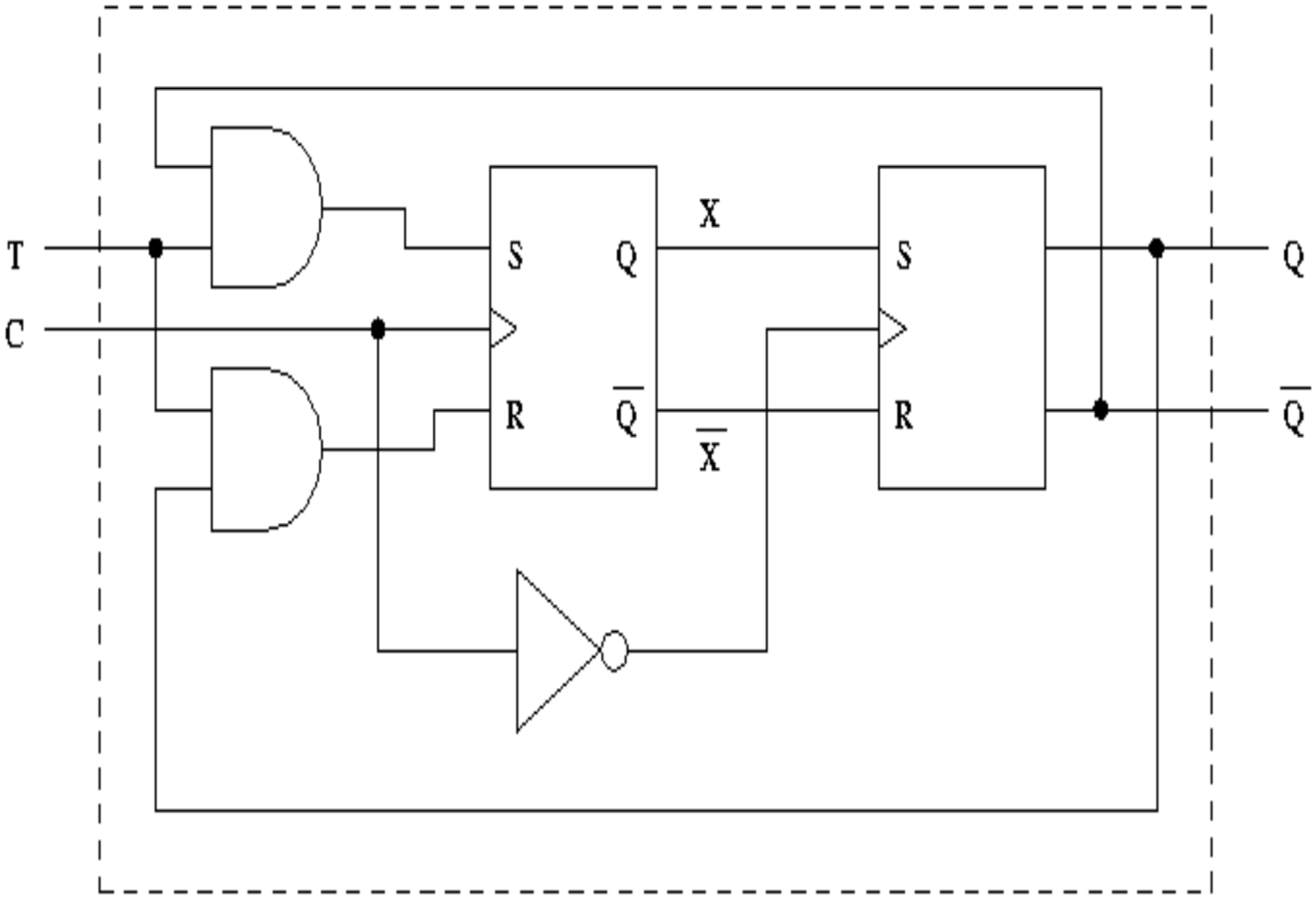
J	K	Q	Q^+
0	0	x	Q
0	1	x	0
1	0	x	1
1	1	x	$\overline{Q}$

For three of the four possible input combinations the operation is identical to the SR. The difference lies in the case where the two inputs are both 1. In this case the flip-flop toggles its output i.e. $Q \rightarrow \overline{Q}$. The next state table for this circuit is

T-type Flip-Flop

The fourth (and last) type of flip-flop is the *T-type*. The T-type is a single input flip-flop like the D-type. It inverts its input each time its input goes high.

The T type flip-flop is constructed by tying the two input lines in a JK flip-flip together.



The next state table is

T	Q	Q^+
0	x	Q
1	x	$\overline{Q}$

State Transition Tables - 1

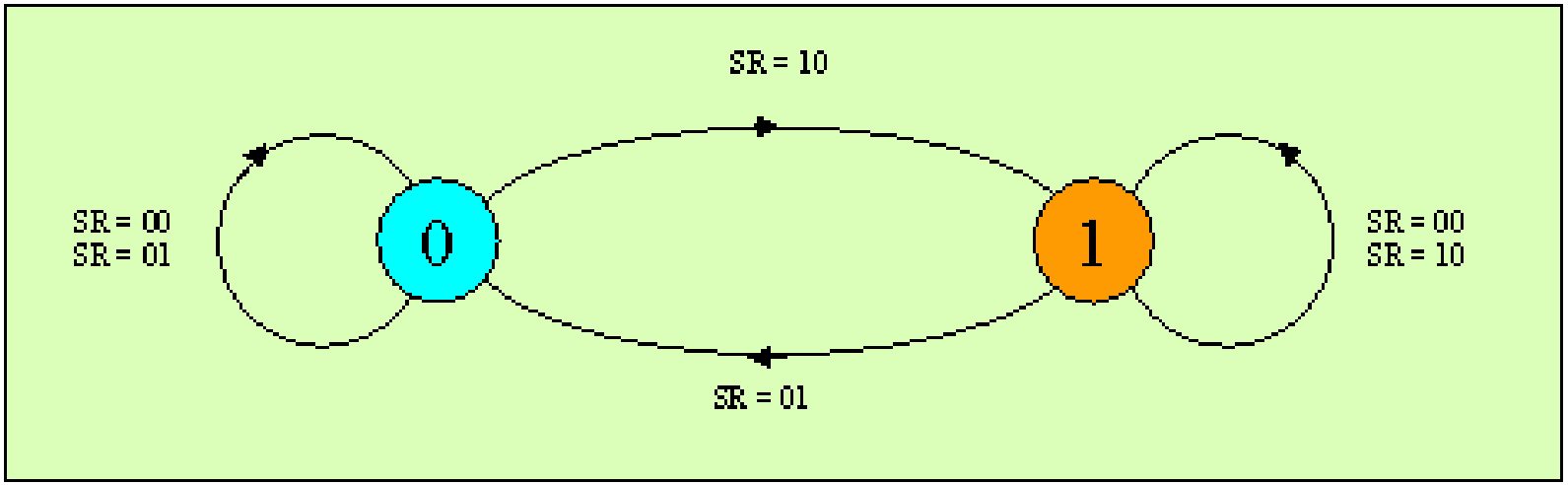
Inputs		Outputs		
Q0	S	R	Q	Q	Notes
0	0	0	0	1	
0	0	1	0	1	
0	1	0	1	0	
0	1	1	*	*	1
1	0	0	1	0	
1	0	1	0	1	
1	1	0	1	0	
1	1	1	*	*	1

Representation 1

Inputs		Outputs		Notes
S	R	Q	Q	
0	0	Q	Q	
1	0	1	0	
0	1	0	1	
1	1	*	*	1

Representation 2

State Transition Diagram for Q



State Transition Tables - 2

- In most general form STTs indicate *start state*, *next state* and *condition for transition*
- Next state for bi-stables is same as *Q output*
- Representation 2 is compacted form of representation 1
  - Q indicates that final state is same as initial state

- Note 1: When both S and R are 1 the output is unstable and undefined (*) - *not same as don't care!*

- There are other kinds of flip-flops which we will return to shortly

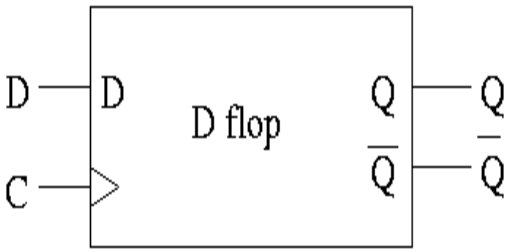
Characteristic Table

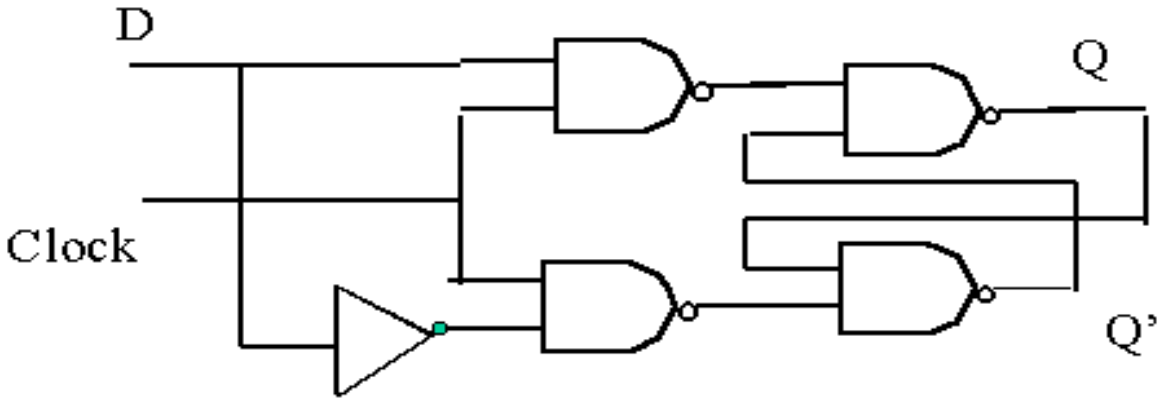
- The characteristic table shows the state transition for every set of inputs.
- Example: SR latch

S	R	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	?

JK flipflop
(more compact)

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	¬ Q(t)

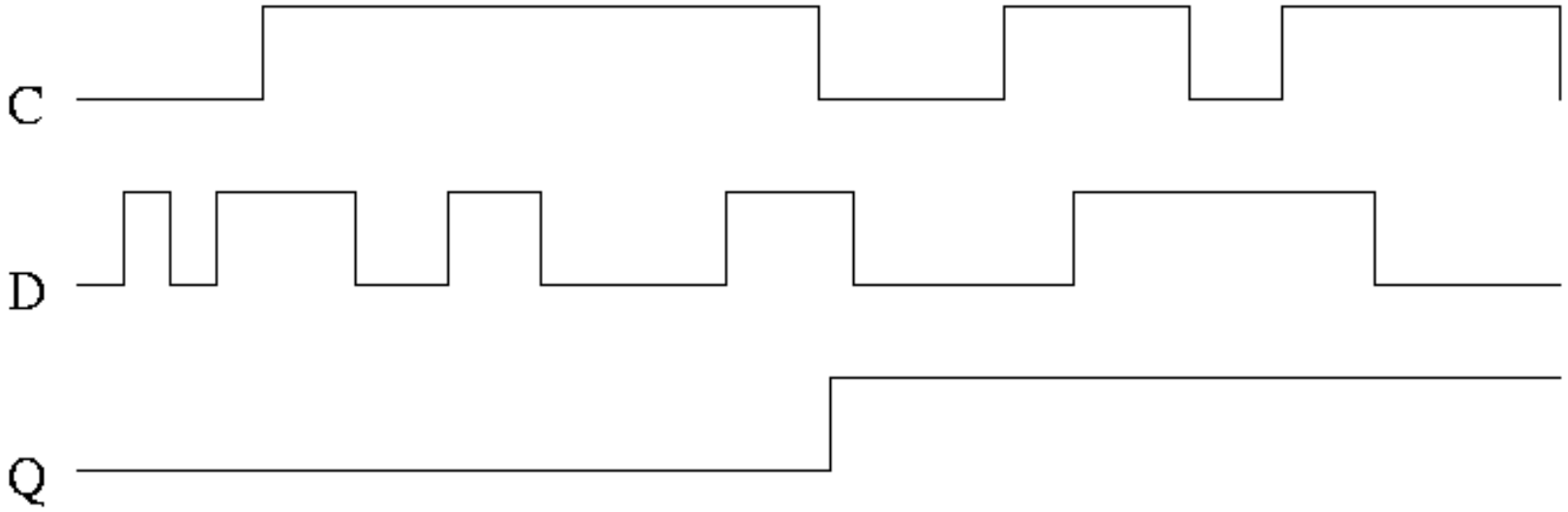




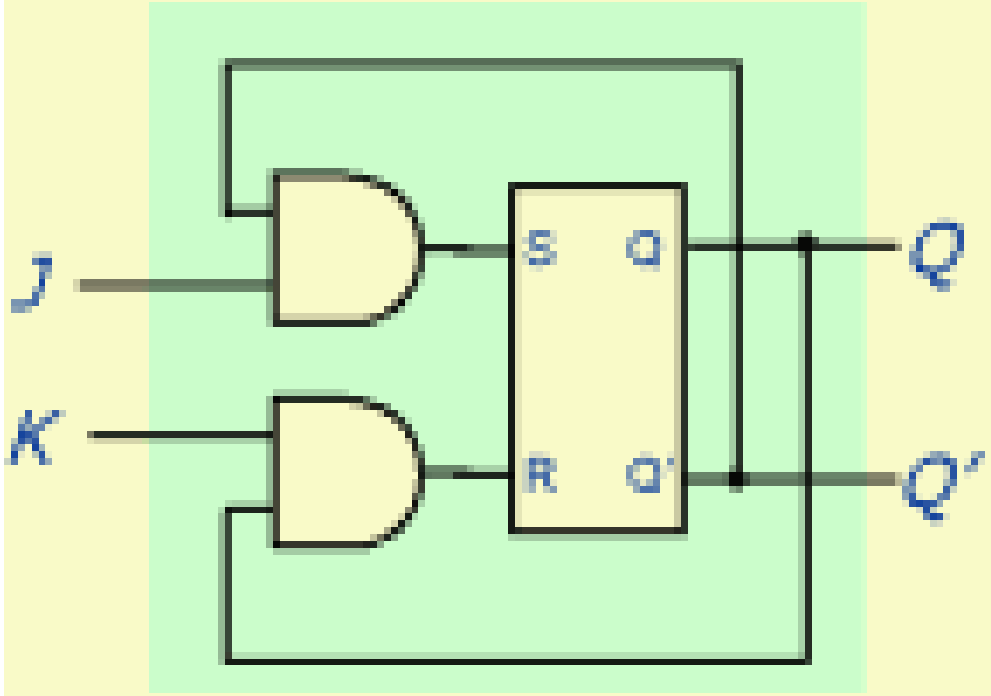
D Latch

C	D	Next State
0	x	No Change
1	0	0
1	1	1

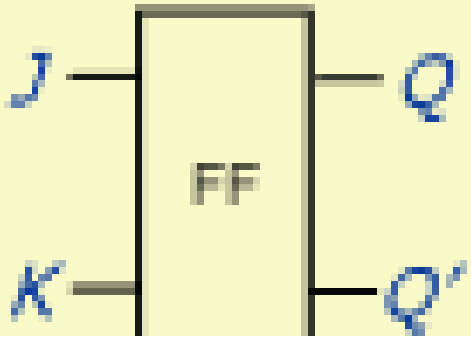
When clock is high the latch is *transparent*, are observed from the output.



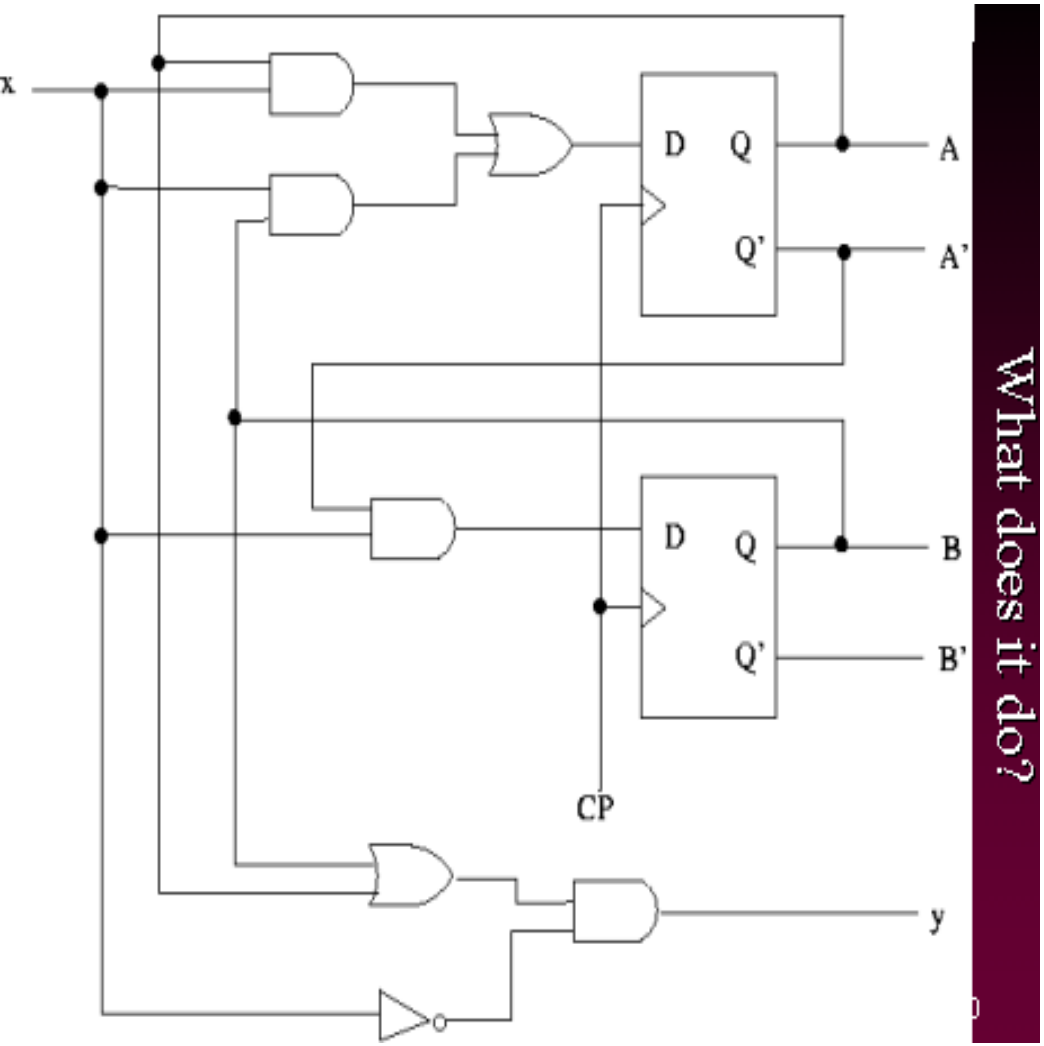
The JK Flip-Flop



J(t)	K(t)	Q(t)	Q(t+2δ)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



$$Q(t+2\delta) = Q(t)K'(t) + Q'(t)J(t)$$



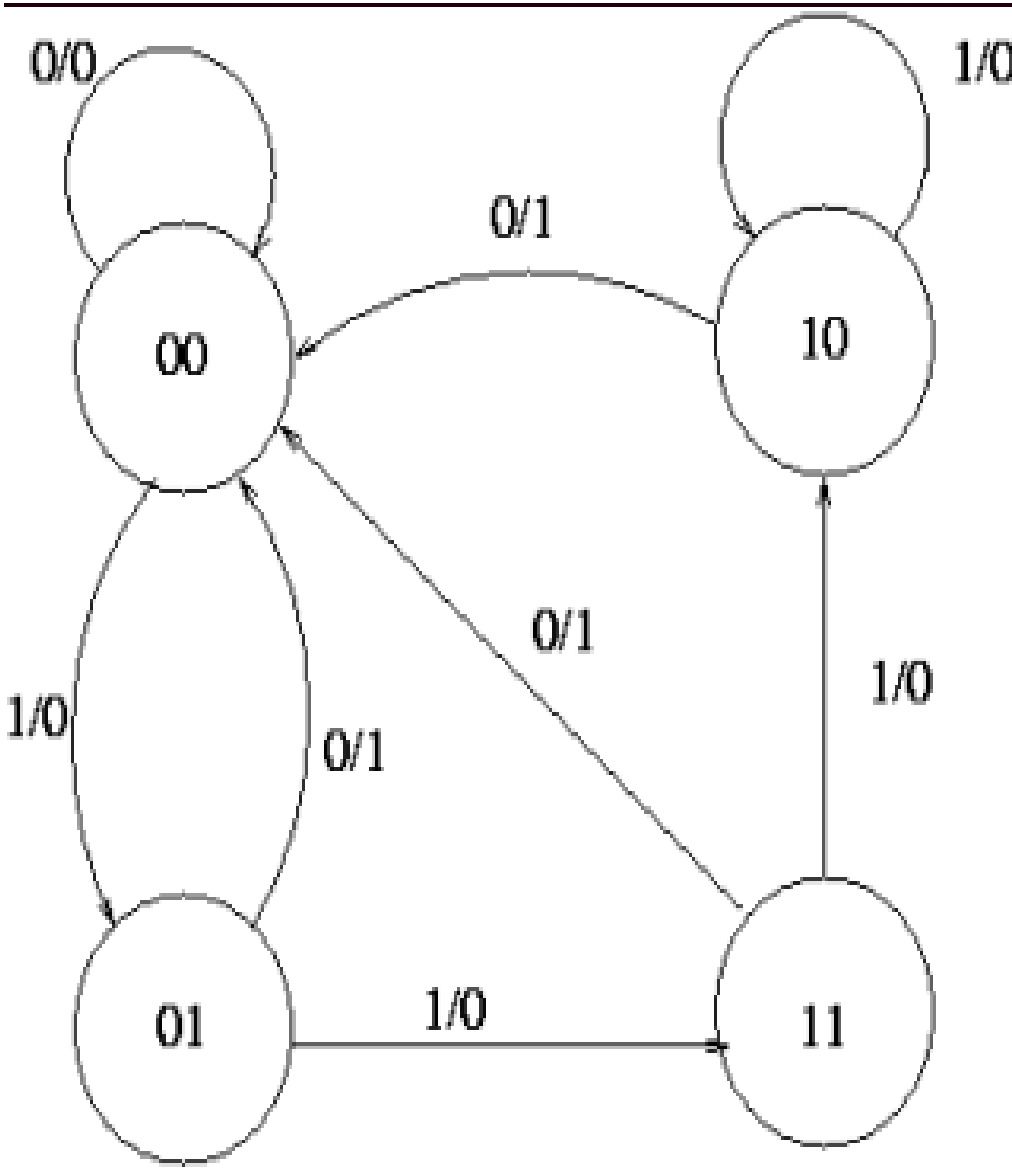
What does it do?

Transition table

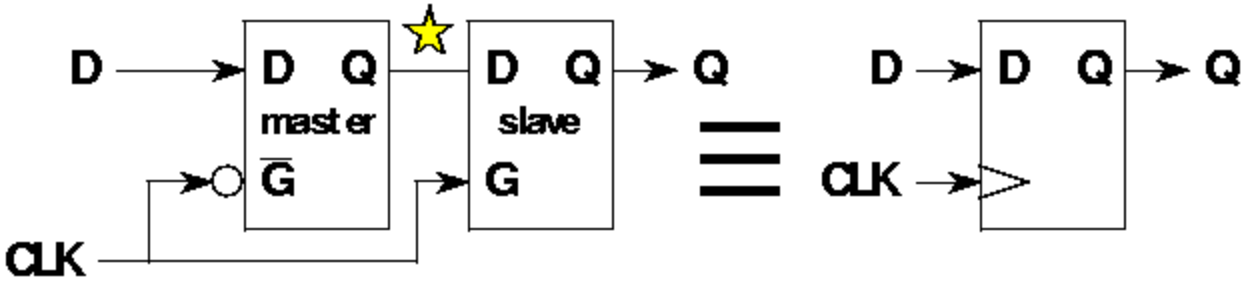
A(t)	B(t)	x	A(t+1)	B(t+1)	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

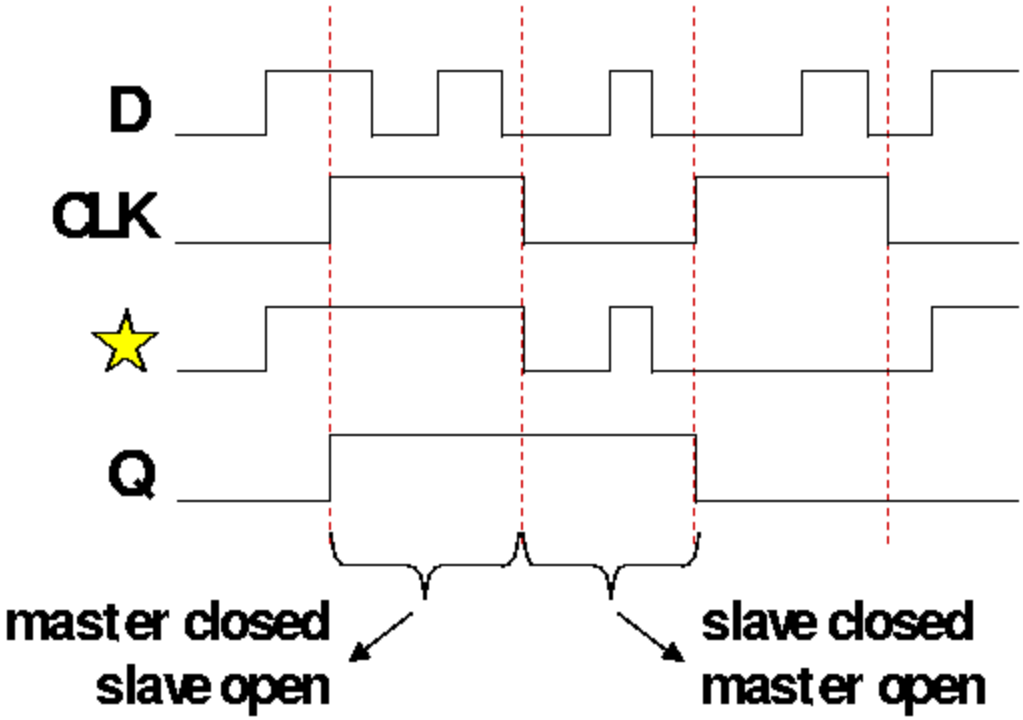
Transition table

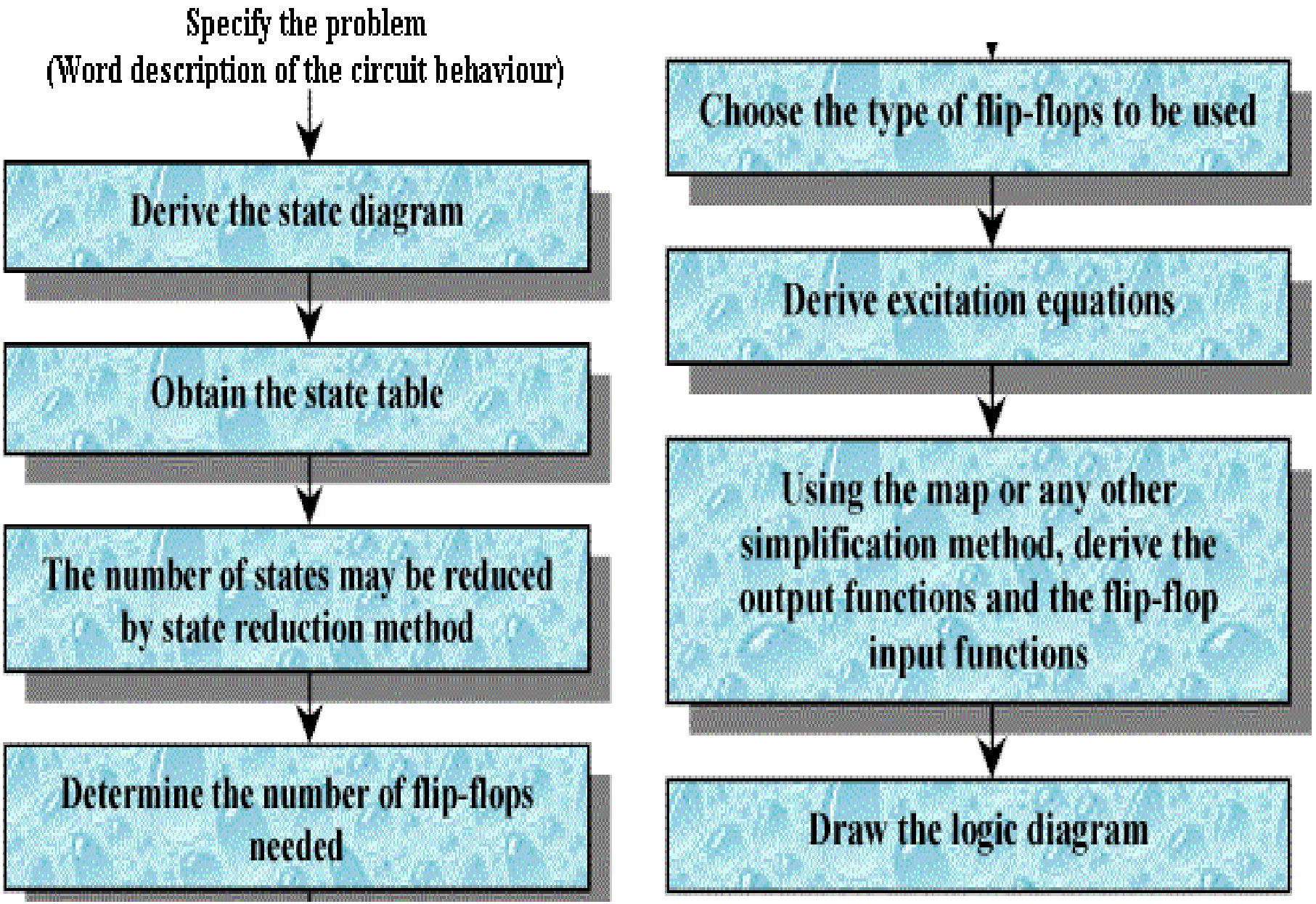
A(t)	B(t)	x	A(t+1)	B(t+1)	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0



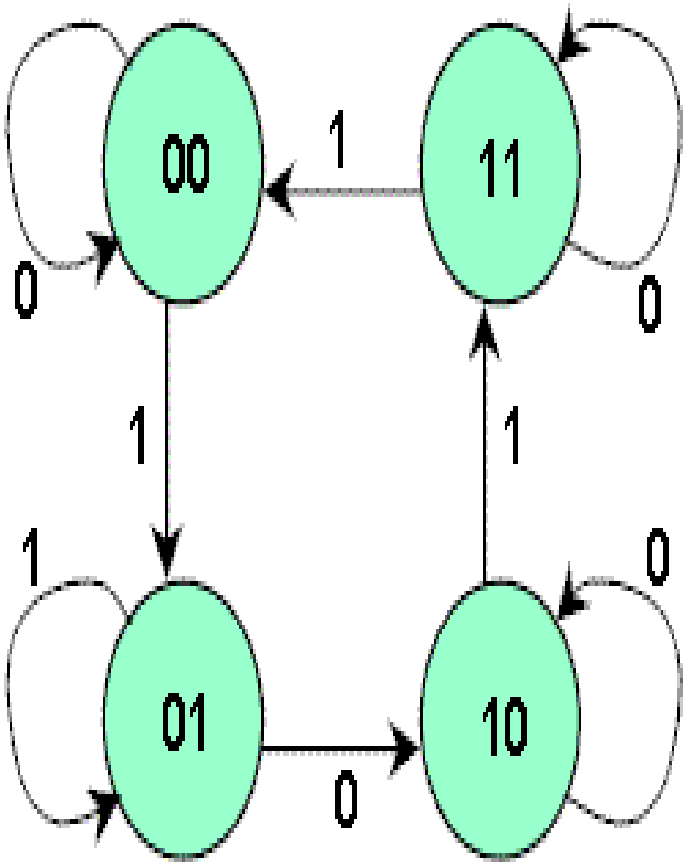
Flip Flop Waveforms






Specify the problem
(Word description of the circuit behaviour)
Derive the state diagram
Obtain the state table
The number of states may be reduced by state reduction method
Determine the number of flip-flops needed
Choose the type of flip-flops to be used
Derive excitation equations
Using the map or any other simplification method, derive the output functions and the flip-flop input functions
Draw the logic diagram

We wish to design a synchronous sequential circuit whose state diagram is shown in Figure. The type of flip-flop to be use is J-K



Two flip-flops are needed to represent the four states and are designated Q0Q1. The input variable is labelled x.

Present State	Next State	
Q0 Q1	x = 0	x = 1
0 0	0 0	0 1
0 1	1 0	0 1
1 0	1 0	1 1
1 1	1 1	0 0

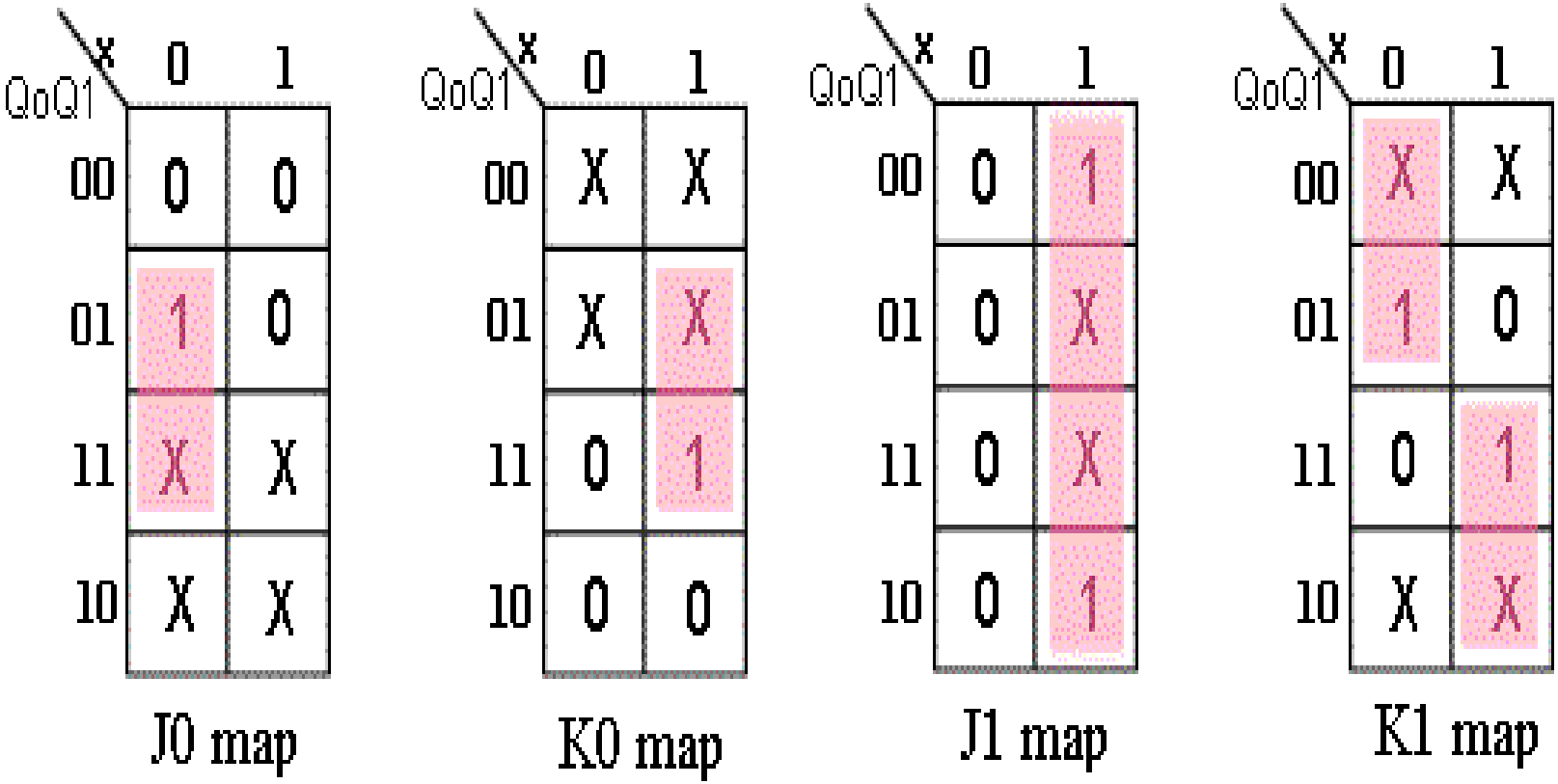
. Excitation table for JK flip-flop

Output Transitions Q → Q(next)	Flip-flop inputs J K
0 → 0	0 X
0 → 1	1 X
1 → 0	X 1
1 → 1	X 0

Excitation table of the circuit

Present State Q0 Q1	Next State Q0 Q1	Input x	Flip-flop Inputs J0K0	J1K1
0 0	0 0	0	0 X	0 X
0 0	0 1	1	0 X	1 X
0 1	1 0	0	1 X	X 1
0 1	0 1	1	0 X	X 0
1 0	1 0	0	X 0	0 X
1 0	1 1	1	X 0	1 X
1 1	1 1	0	X 0	X 0
1 1	0 0	1	X 1	X 1

The simplified Boolean functions for the combinational circuit can now be derived



The flip-flop input functions are derived:

$$J0 = Q1^*x' \qquad K0 = Q1^*x$$

$$J1 = x \qquad K1 = Q0'^*x' + Q0^*x = Q0 \odot x$$

Note: the symbol $\odot$ is exclusive-NOR.

The logic diagram is drawn in Figure 15.

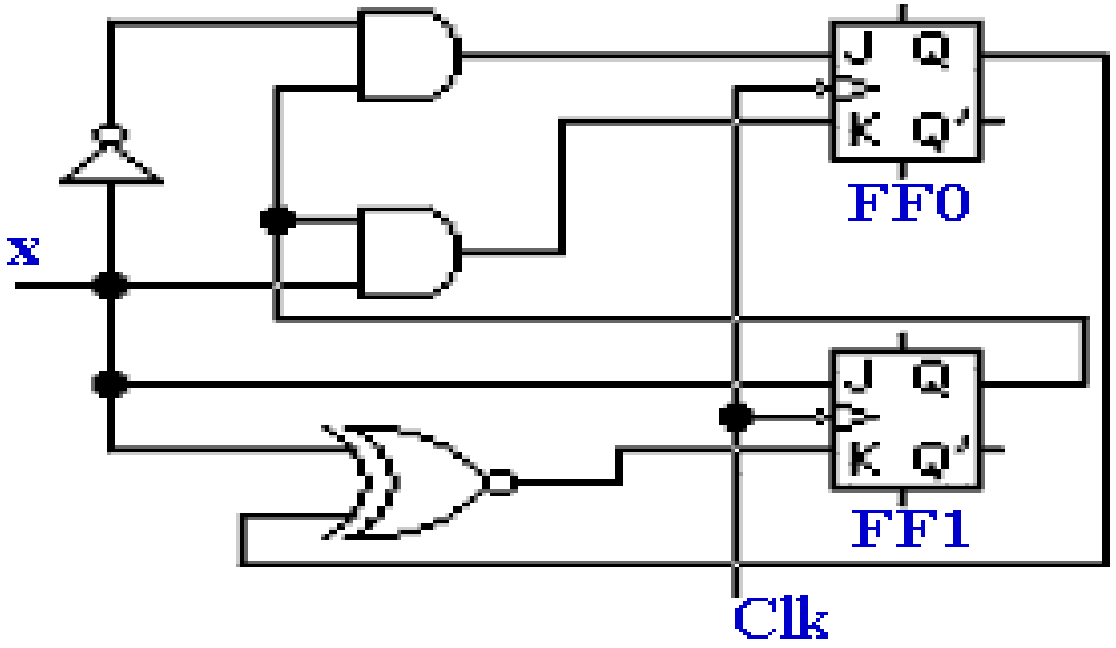


Figure 15. Logic diagram of the sequential circuit.

Example

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

- Implement the following sequence:

  $000 \rightarrow 110 \rightarrow 100 \rightarrow 111 \rightarrow 000 \rightarrow \ldots$

- Excitation table:

A_2	A_1	A_0	JA_2	KA_2	JA_1	KA_1	JA_0	KA_0
0	0	0	1	X	1	X	0	X
1	1	0	X	0	X	1	0	X
1	0	0	X	0	1	X	1	X
1	1	1	X	1	X	1	X	1
0	0	0	1	X	1	X	0	X

Solving for JA_i and KA_i

A_2	A_1	A_0	JA_2	KA_2	JA_1	KA_1	JA_0	KA_0
0	0	0	1	X	1	X	0	X
1	1	0	X	0	X	1	0	X
1	0	0	X	0	1	X	1	X
1	1	1	X	1	X	1	X	1

$$JA_2 = \overline{A_2} \qquad JA_1 = \overline{A_1} \qquad JA_0 = A_2 \oplus A_1$$

$$KA_2 = A_0 \qquad KA_1 = A_1 \qquad KA_0 = A_0$$

Second Example

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

- Implement the following sequence:

  $000 \rightarrow 010 \rightarrow 001 \rightarrow 101 \rightarrow 110 \rightarrow 000 \rightarrow \ldots$

- Excitation table:

A_2	A_1	A_0	JA_2	KA_2	JA_1	KA_1	JA_0	KA_0
0	0	0	0	X	1	X	0	X
0	1	0	0	X	X	1	1	X
0	0	1	1	X	0	X	X	0
1	0	1	X	0	1	X	X	1
1	1	0	X	1	X	1	0	X
0	0	0	1	X	1	X	0	X

Solving for JA_i and KA_i

A_2	A_1	A_0	JA_2	KA_2	JA_1	KA_1	JA_0	KA_0
0	0	0	0	X	1	X	0	X
0	1	0	0	X	X	1	1	X
0	0	1	1	X	0	X	X	0
1	0	1	X	0	1	X	X	1
1	1	0	X	1	X	1	0	X

$$JA_2 = A_0 \qquad JA_1 = A_2 + \overline{A_0} \qquad JA_0 = A_2 \oplus A_1$$

$$KA_2 = A_1 \qquad KA_1 = A_1 \qquad KA_0 = A_2$$

How do we determine the combinatorial ciccuit?

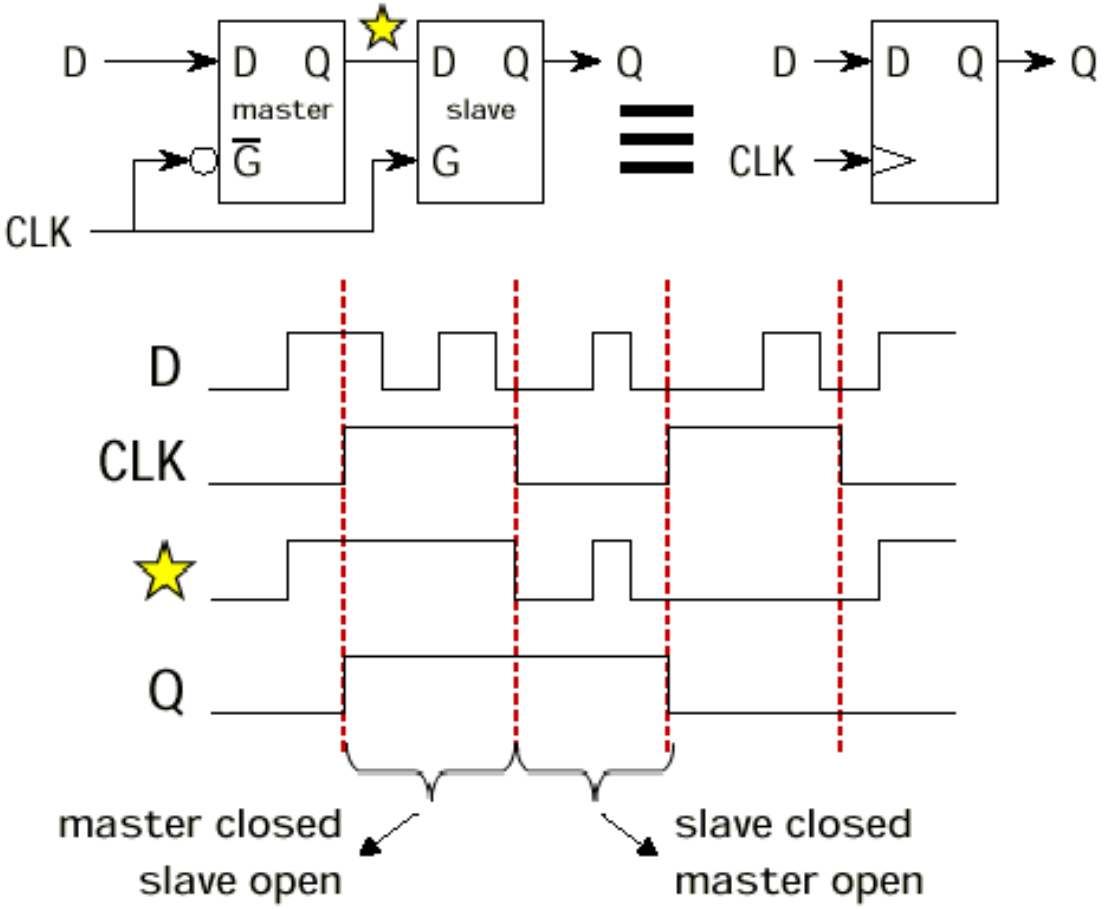
- This circuit has three inputs, I, R, and the current A.
- It has one output, DA, which is the desired *next* A.
- So we draw a truth table, as before.
- For convenience I added the label Next A to the DA column

Current A	I	R	Next A DA <--
0	0	0	0
1	0	0	1
0	1	0	1
1	1	0	0
x	x	1	0

i.e. to what must I set DA in order to get the desired Next A for the next cycle.

But this table is simply the truth table for the combinatorial circuit.

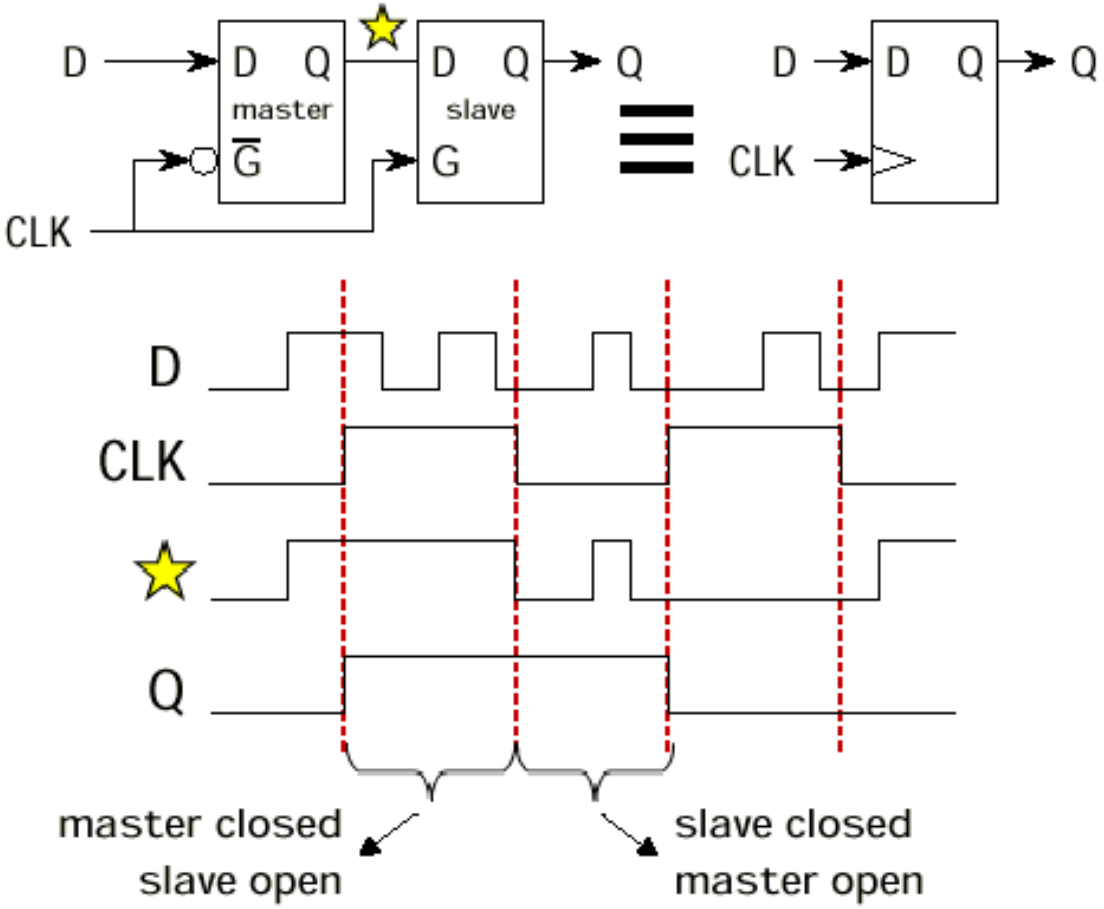
Edge-triggered Flip Flop



Idea:

- only one latch "transparent" at any time
  → no combinational path through flip flop
- Q only changes shortly after 0 →1

Edge-triggered Flip Flop

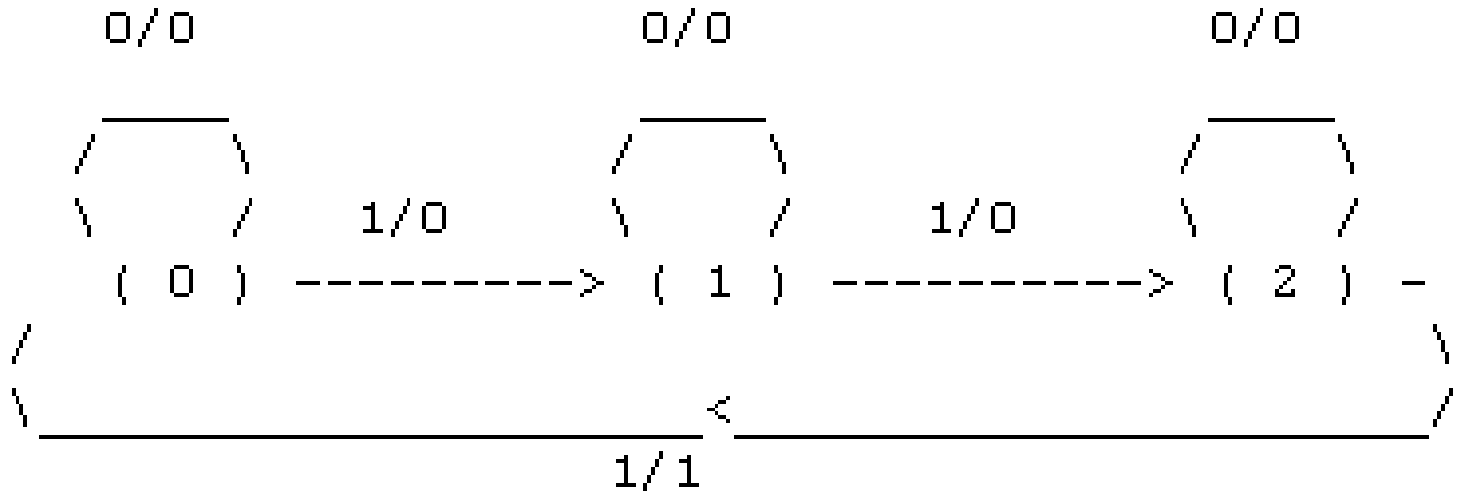


Idea:

- only one latch "transparent" at any time
  → no combinational path through flip flop
- Q only changes shortly after 0 →1

A divide-by-three counter which outputs one 1 for every 3 1's seen as input (not necessarily in succession.) After outputting a 1, it starts counting all over again. 1.

To build this, will need three states, corresponding to 0, 1, or 2 1's seen so far.



3. State table:

Current State		Input	Next State		Output	Excitation			
Q1	Q0	Enable (E)	Q1	Q0	Carry(C)	J1	K1	J0	K0
0	0	0	0	0	0	0	-	0	-
0	0	1	0	1	0	0	-	1	-
0	1	0	0	1	0	0	-	-	0
0	1	1	1	0	0	1	-	-	1
1	0	0	1	0	0	-	0	0	-
1	0	1	0	0	1	-	1	0	-
1	1	0	-	-	-	-	-	-	-
1	1	1	-	-	-	-	-	-	-

J1 = (Q0)E K1 = E J0 = (Q1')E K0 = E

output: C = (Q1)E

Designing with *JK* Flip-Flops

- The design of a sequential circuit with other than the D type is complicated by the fact that the flip-flop input equations for the circuit must be derived indirectly from the state table. When D-type flip-flops are employed, the input equations are obtained directly from the next state. This is not the case for *JK* and other types of flip-flops. In order to determine the input equations for these flip-flops, it is necessary to derive a functional relationship between the state table and the input equations.

Flip-Flop Excitation Tables

- A table that lists the required inputs for a given change of state is known as an *excitation table.* Example of an excitation table is shown below:

TABLE 4-10
Flip-Flop Excitation Tables

(a) *JK* Flip-Flop				(b) *SR* Flip-Flop			
$Q(t)$	$Q(t+1)$	J	K	$Q(t)$	$Q(t+1)$	S	R
0	0	0	X	0	0	0	X
0	1	1	X	0	1	1	0
1	0	X	1	1	0	0	1
1	1	X	0	1	1	X	0

(c) *D* Flip-Flop			(d) *T* Flip-Flop		
$Q(t)$	$Q(t+1)$	D	$Q(t)$	$Q(t+1)$	T
0	0	0	0	0	0
0	1	1	0	1	1
1	0	0	1	0	1
1	1	1	1	1	0

Flip-Flop Excitation Tables (cont)

The excitation table show four different types of flip-flops. Each table has a column for the present state $Q(t)$, a column for the next state $Q(t + 1)$, and a column for each flip-flop input to show how the required transition is achieved. The symbol X in the table represents a don't-care condition, which means that it does not matter whether the input is 0 or 1.

Flip-Flop Excitation Tables (cont)

The excitation table for the D flip-flop shows that the next state is always equal to the D input and is independent of the present state. This can be represented algebraically:

$$D = Q(t + 1)$$

Design Procedure

- The design procedure for sequential circuits with *JK* flip-flops is the same as that for sequential circuits with D flip-flops, except that the input equations must be evaluated from the present-state to next-state transition derived from the excitation table.

Design Procedure (cont)

- The advantage of using *JK*-type flip-flops when designing sequential circuits is that there are so many don’t-care entries indicates that the combinational circuit for the input equations is likely to be simpler, because don’t-care minterms usually help in obtaining simpler expressions.

Design Procedure (cont)

In order to perform the simulation, a clock, as well as the input signals R and X, is required. In doing the simulation of any sequential circuit, sufficient time must be provided in the clock period for each of the following:

1. All flip-flops and inputs to change;
2. The effects of these changes to propagate through the combinational logic of the circuit to the flip-flop inputs; and
3. The setup of the flip-flops for the next clock edge to occur.