Digital Systems I

Instructor: Mohammad Ghamari

Electrical and Computer Engineering Department

These slides are modified version of the slides by Nozar Tabrizi, Introduction to Digital Circuits

Chapter 7

Memory Cells and Analysis of Sequential Circuits

So far, **Combinational** **Logic**:

Output depends on only current inputs

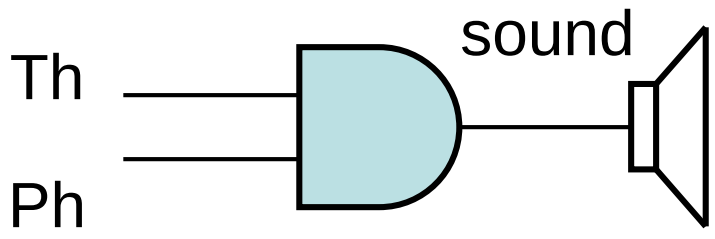
Or

There is NO memory.

Example: Chemical plant (see next slide)

Example

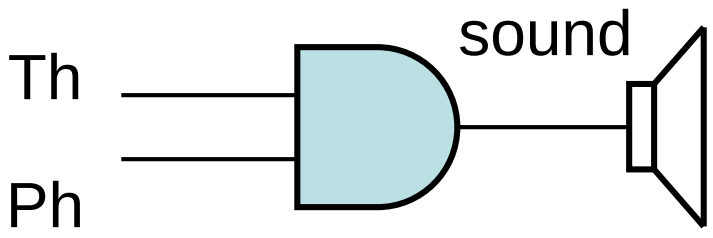
Alarm must go off only if both pressure & temperature are high.



It works

Example

Alarm must sound only if pressure & temperature are high, provided that temperature goes up first.



Does not work!

Since current inputs are not enough, circuit has to know (remember) something from the past as well.

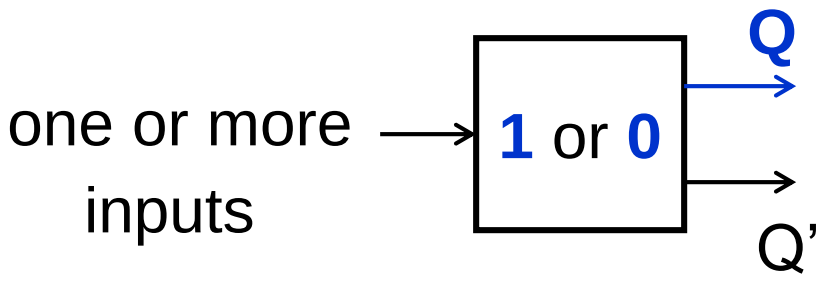
Sequential Logic (as opposed to Combinational Logic)

Output depends on current inputs as well as previous inputs, or

Keeps a history from past, or

Needs **MEMORY**.

Memory Cells (writable & readable)



Q' = Complement of Q

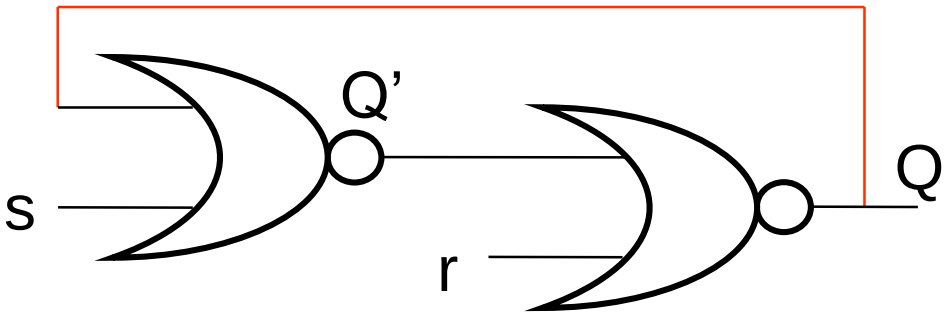
Q (output) shows content or state of the cell.

Cell is in *read mode* as long as Q is accessible.

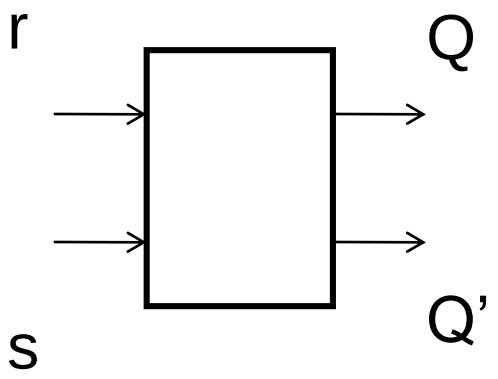
Different types of memory cells have their own ways or RULES for *write* operation (Set & Reset).

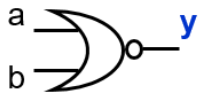
r-s latch

Logic Diagram



Symbol

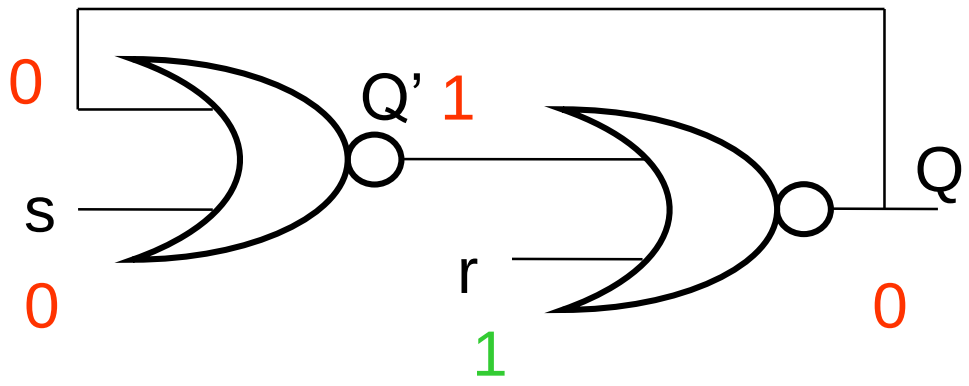


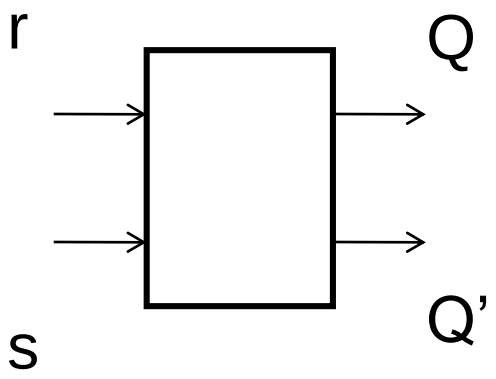


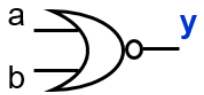
a	b	y
0	0	1
0	1	0
1	0	0
1	1	0

r-s latch (r = 1, s = 0)

Reset Mode



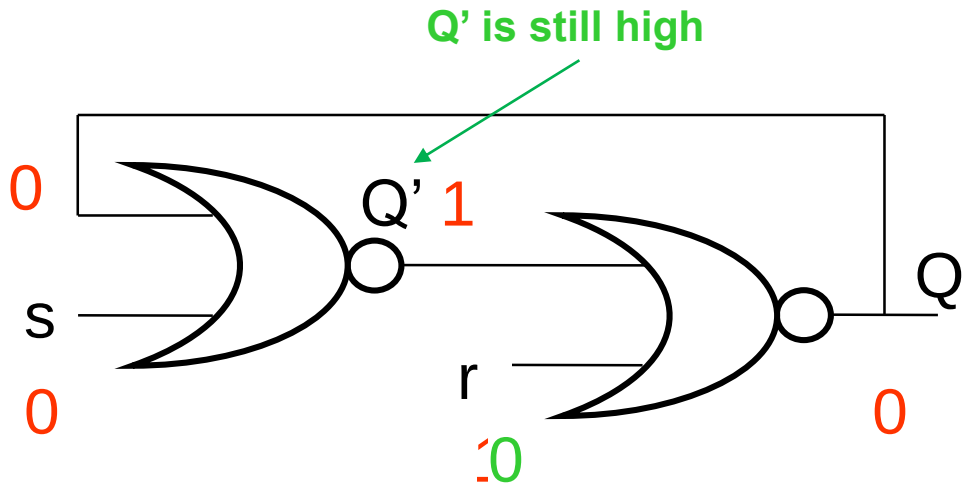


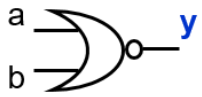


a	b	y
0	0	1
0	1	0
1	0	0
1	1	0

r-s latch (r = s = 0)

Hold Mode

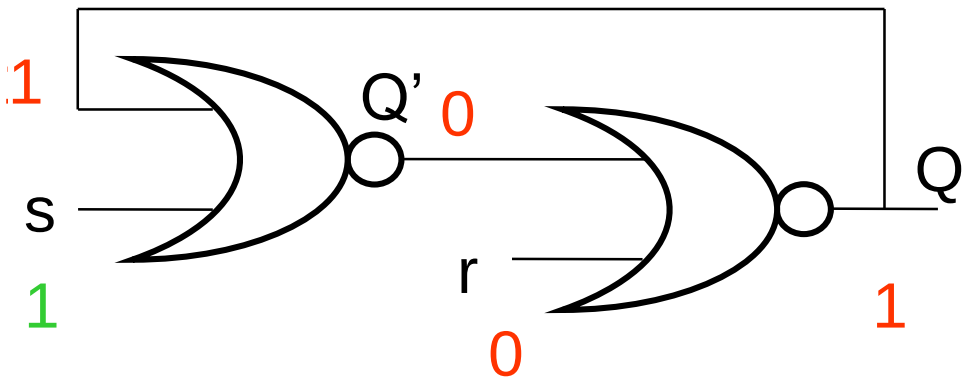


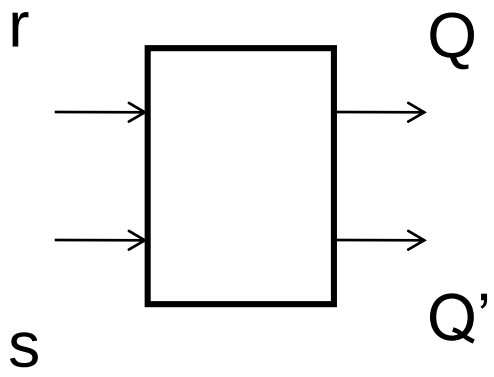


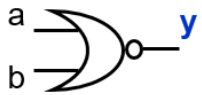
a	b	y
0	0	1
0	1	0
1	0	0
1	1	0

r-s latch (r = 0, s = 1)

Set Mode



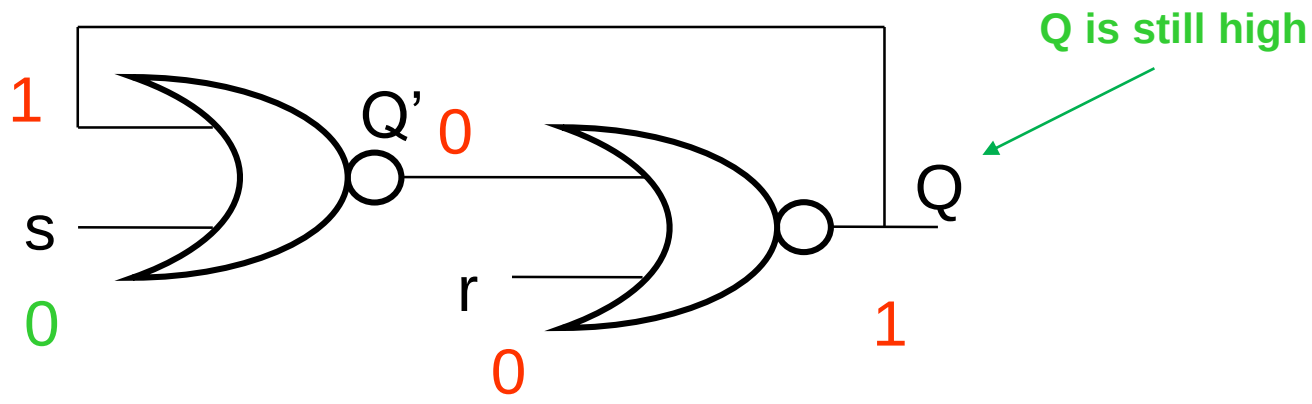


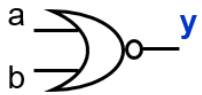


a	b	y
0	0	1
0	1	0
1	0	0
1	1	0

r-s latch (r = s = 0)

Hold Mode



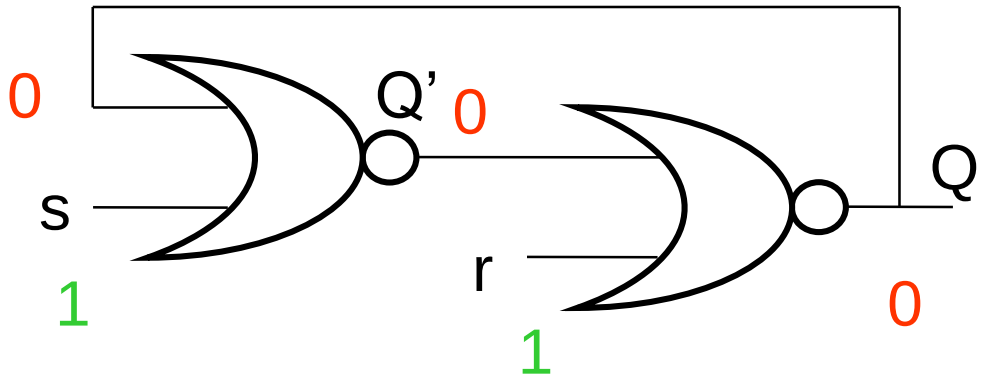


a	b	y
0	0	1
0	1	0
1	0	0
1	1	0

r-s latch (r = s = 1)

Do not use!

Results in Q=Q'=0



r-s latch characteristic table (read from top to bottom)

Same input generates different outputs!

r s	Q	Q’	Mode
1 0	0	1	Reset
0 0	**0**	**1**	**Hold**
0 1	1	0	Set
0 0	**1**	**0**	**Hold**
1 1	0	0	Don’t use

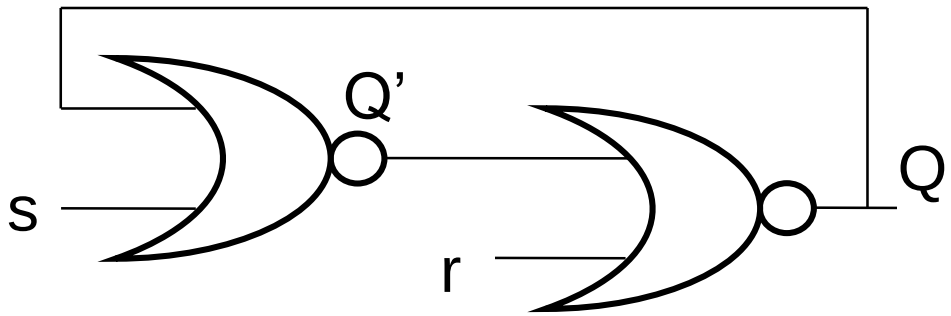
r-s latch characteristic table (reworded)

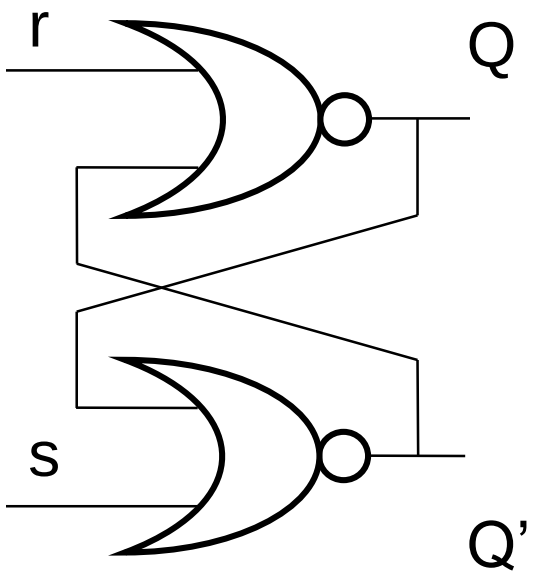
r s	Q	Q’	Mode
0 0	NC	NC	Hold
0 1	1	0	Set
1 0	0	1	Reset
1 1	0	0	Don’t use

NC: No change

If both inputs are at logic low, the output does not change.

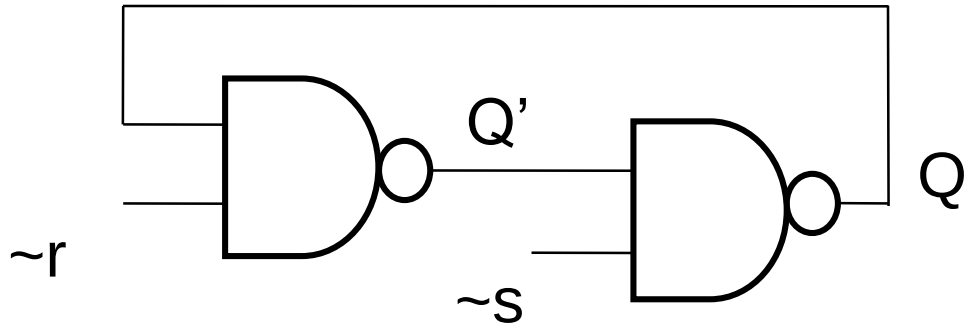
Two different drawings



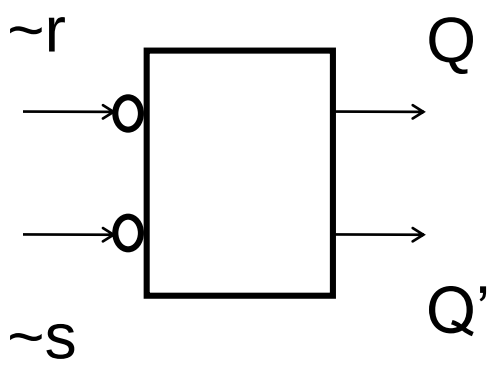


NAND-based r-s latch

Logic Diagram



Symbol

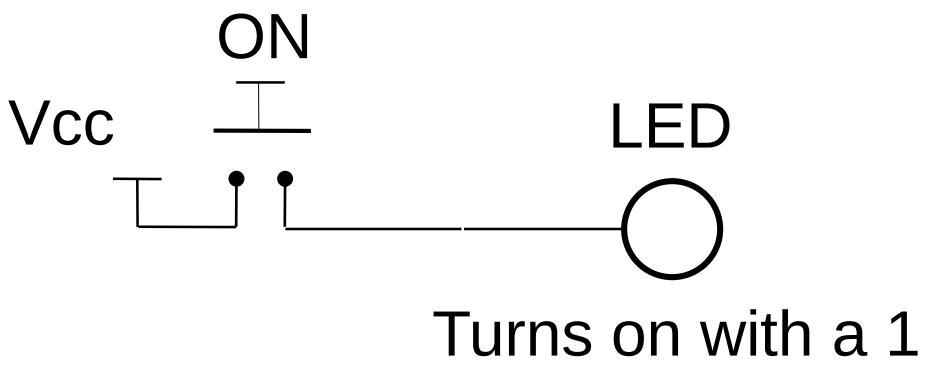


Characteristic table (~r ~s latch)

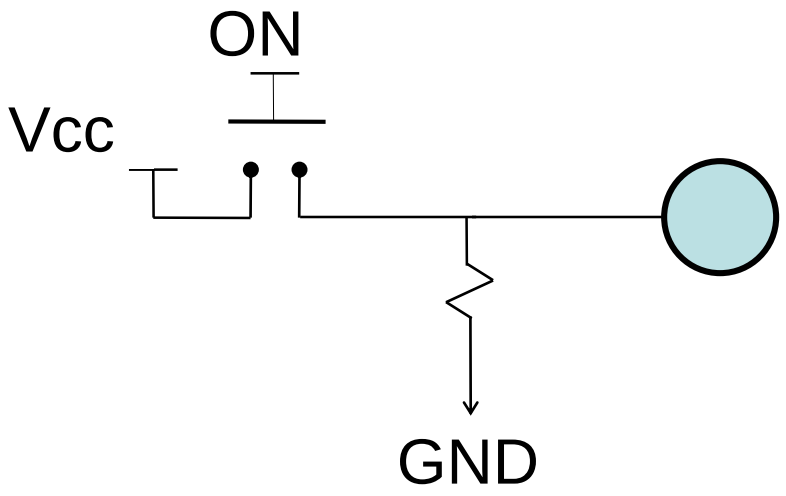
~r	~s	Q	Q’	Mode
0	0	1	1	Don’t use
0	1	0	1	Reset
1	0	1	0	Set
1	1	NC	NC	Hold

Example 1.

Push-button has no memory

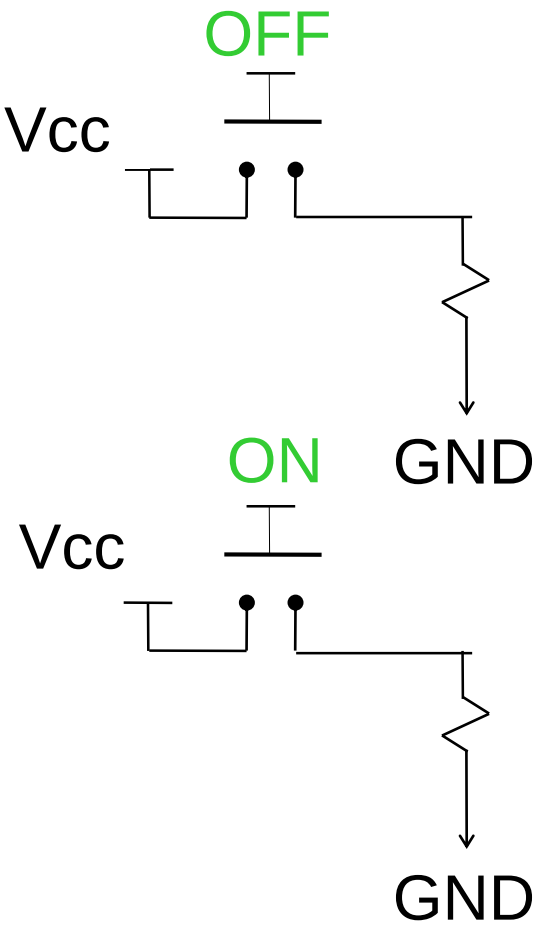


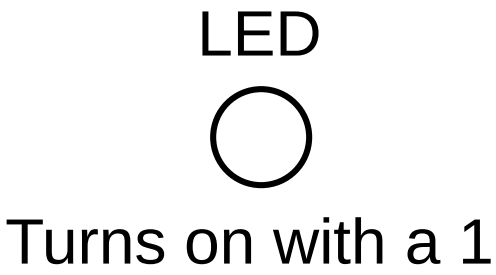
The Vcc connection overpowers the GND connection (due to the resistance in the GND connection)



Example 1. (Cont'd)

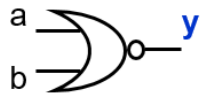
Add an electronic memory to two (memoryless) push-buttons to let the resulting system remember the most recent button that was pushed, hence to be able to turn an off LED on or vice versa, by pressing the right button.



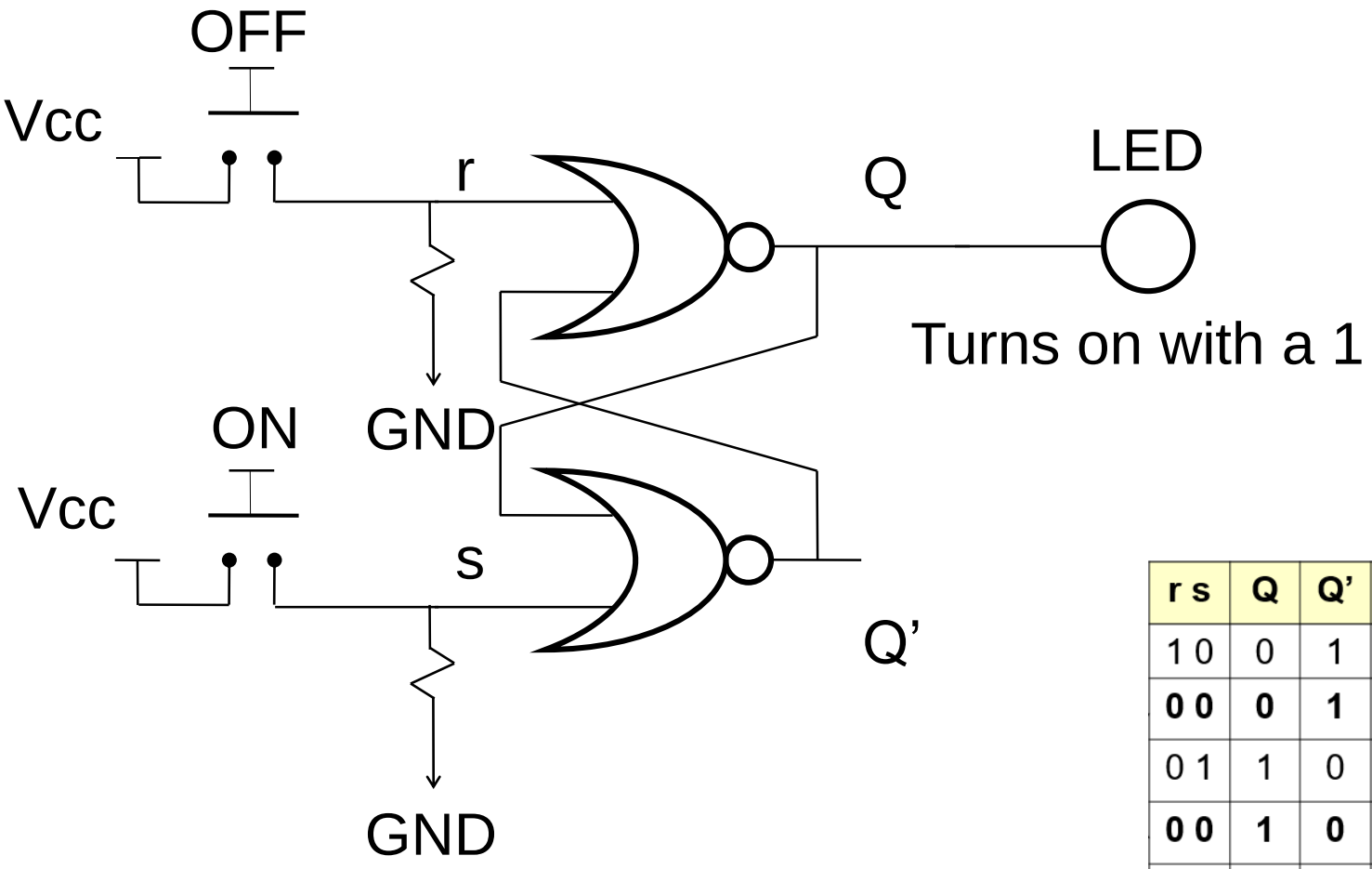


Example 1. (Cont'd)

ON/OFF push-buttons with electronic memory



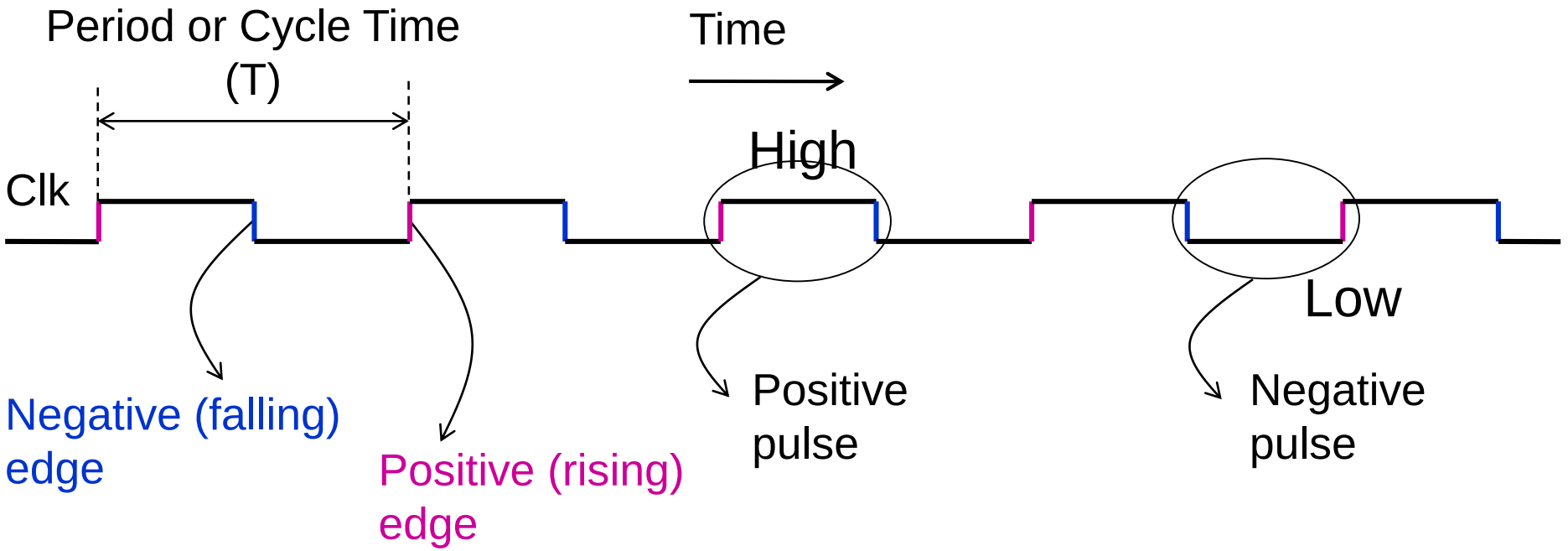
a	b	y
0	0	1
0	1	0
1	0	0
1	1	0



r	s	Q	Q'	Mode
1	0	0	1	Reset
0	**0**	**0**	**1**	**Hold**
0	1	1	0	Set
0	**0**	**1**	**0**	**Hold**
1	1	0	0	Don't use

In *synchronous* sequential logic (to be covered in Dig I)

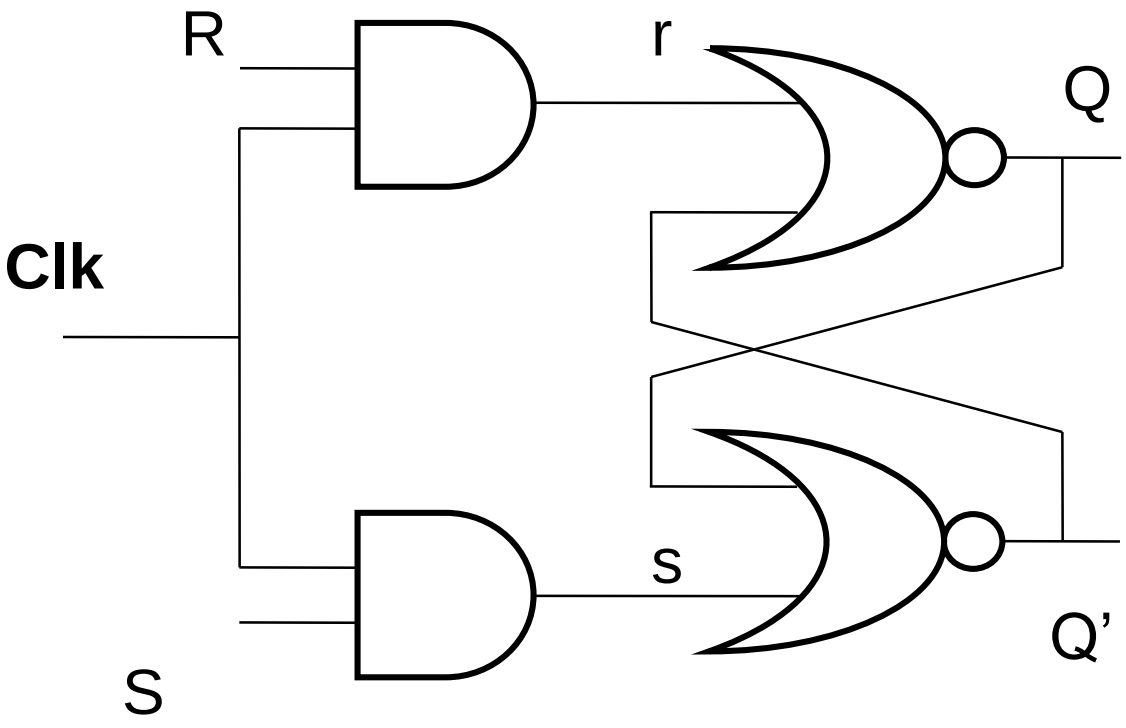
all write operations into memory cells are carried out simultaneously, thus we use a **global synchronizing** signal called *clock* (or *Clk*).



Frequency, f = 1/T

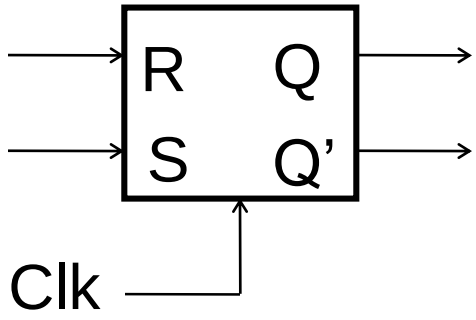
Duty cycle = High/T

(Clocked) R-S latch



Logic Diagram

Symbol



- Any changes to the content of the latch may only take place while clk is asserted.
- Deasserted clk keeps the memory cell in the hold mode, no matter what values S and R take.

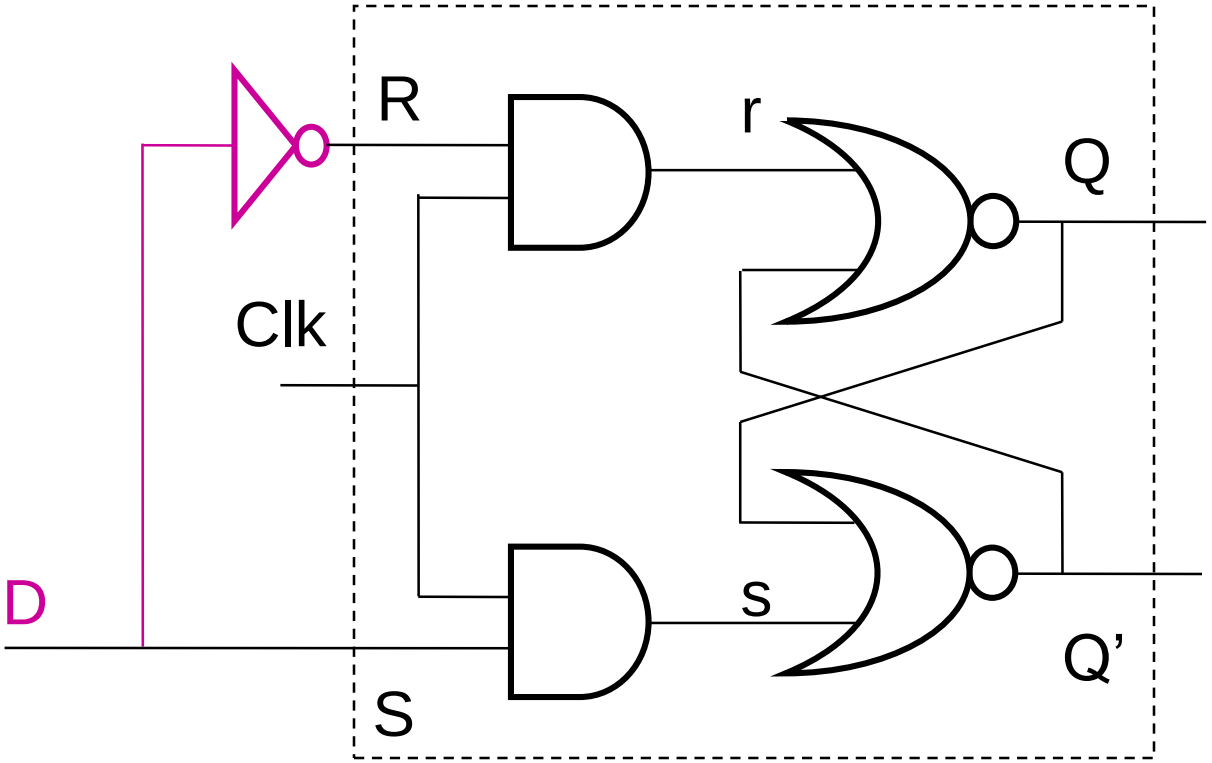
(Clocked) R-S latch: characteristic table

Do not forget Clock!

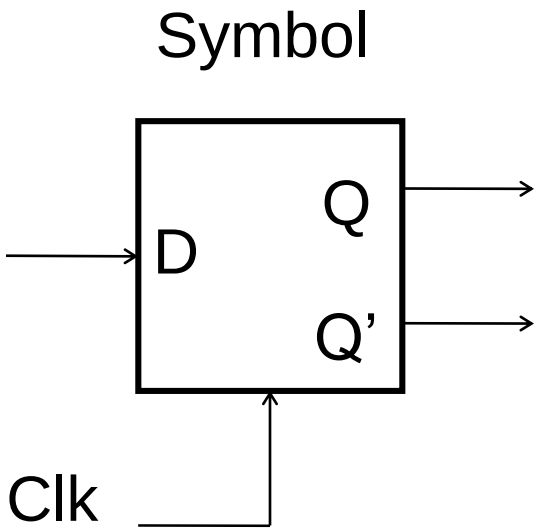
R	S	Q	Q’	Mode
0	0	NC	NC	Hold
0	1	1	0	Set
1	0	0	1	Reset
1	1	0	0	Don’t use

For example, the third row reads if S=0 and R=1 while clk=1, the memory cell is reset.

D latch



Logic Diagram



R = S'

D = S

D latch: Characteristic Table

Do not forget Clk!

R	S	Q	Q’	Mode
0	0	NC	NC	Hold
0	1	1	0	Set
1	0	0	1	reset
1	1	0	0	Don’t use

D	Q	Q’
0	0	1
1	1	0

R = S’

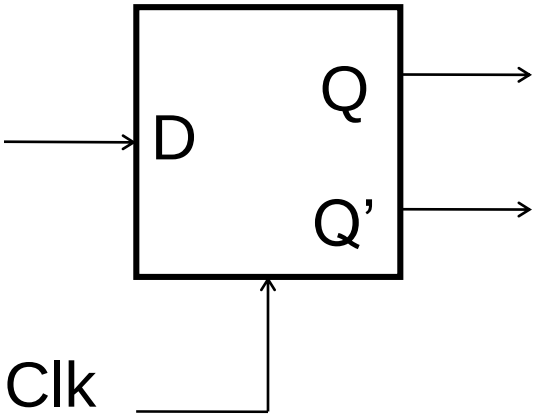
D = S

R ≠ S

D latch: timing diagram (ideal)

Symbol

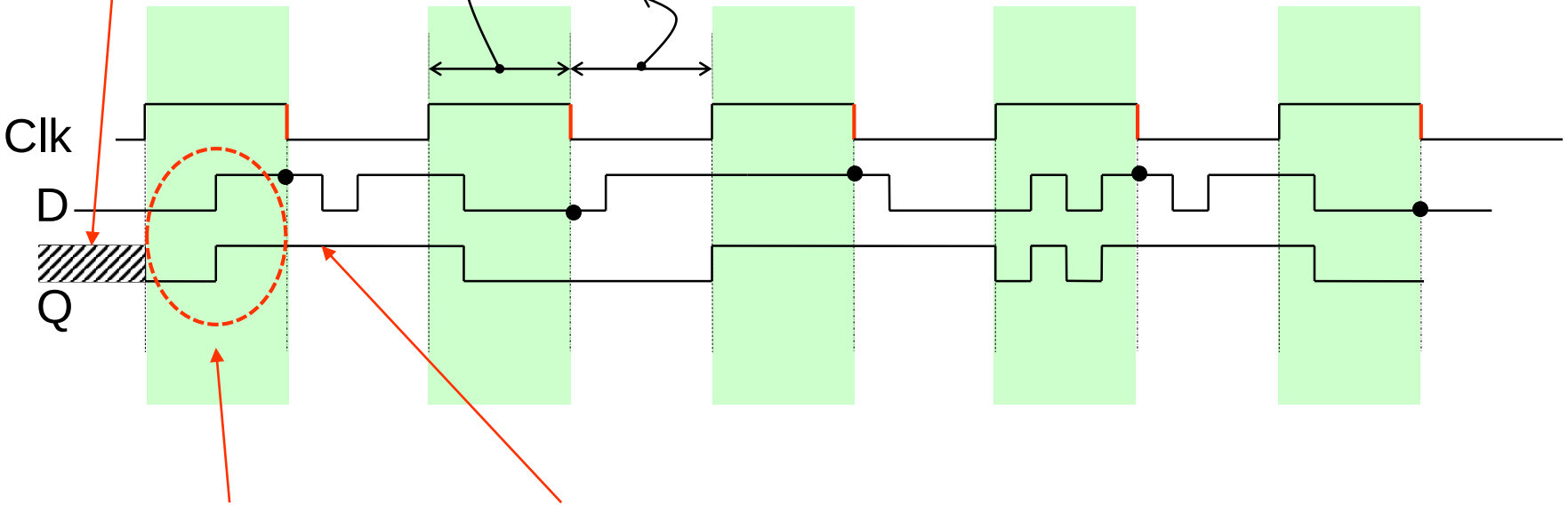
D	Q	Q’
0	0	1
1	1	0



The content (state) of the latch is unknown before the first positive edge

Transparent
Q = D

Hold



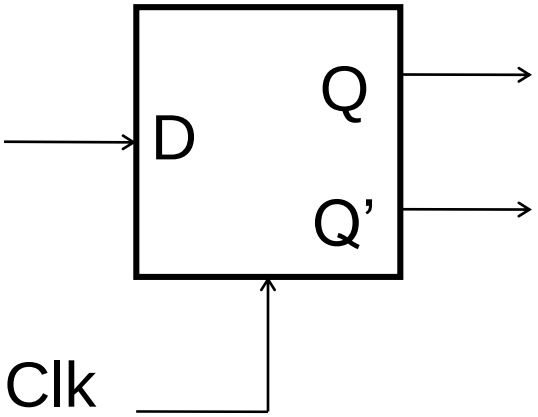
Output Q exactly follows input D

With the falling edge of the clock signal, the latch leaves the transparent mode and samples and holds input D

D latch: timing diagram (real)

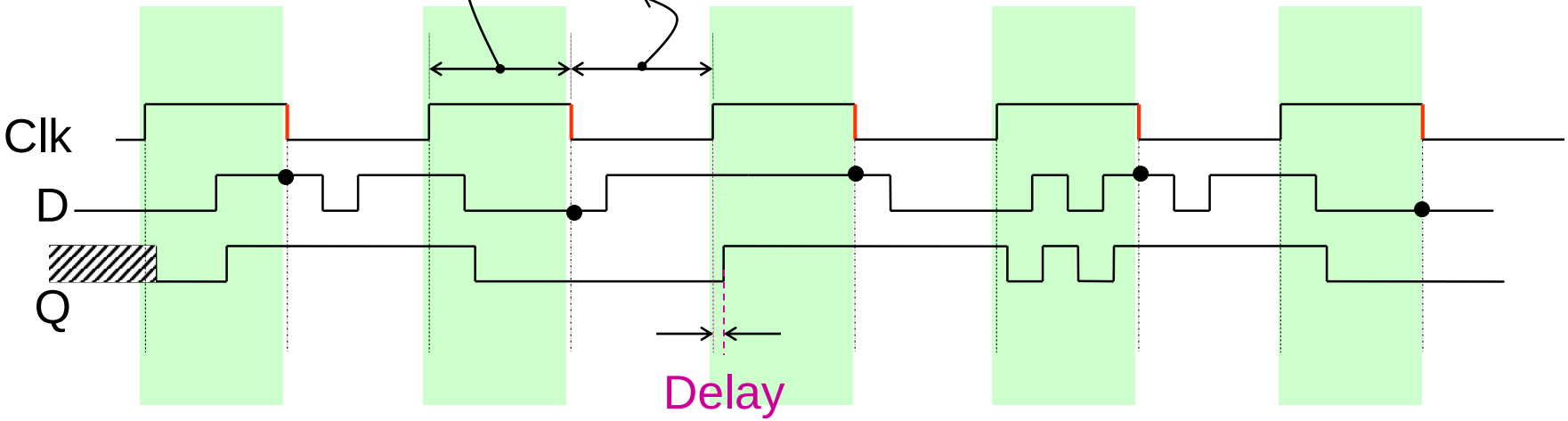
Symbol

D	Q	Q’
0	0	1
1	1	0



Transparent
Q = D

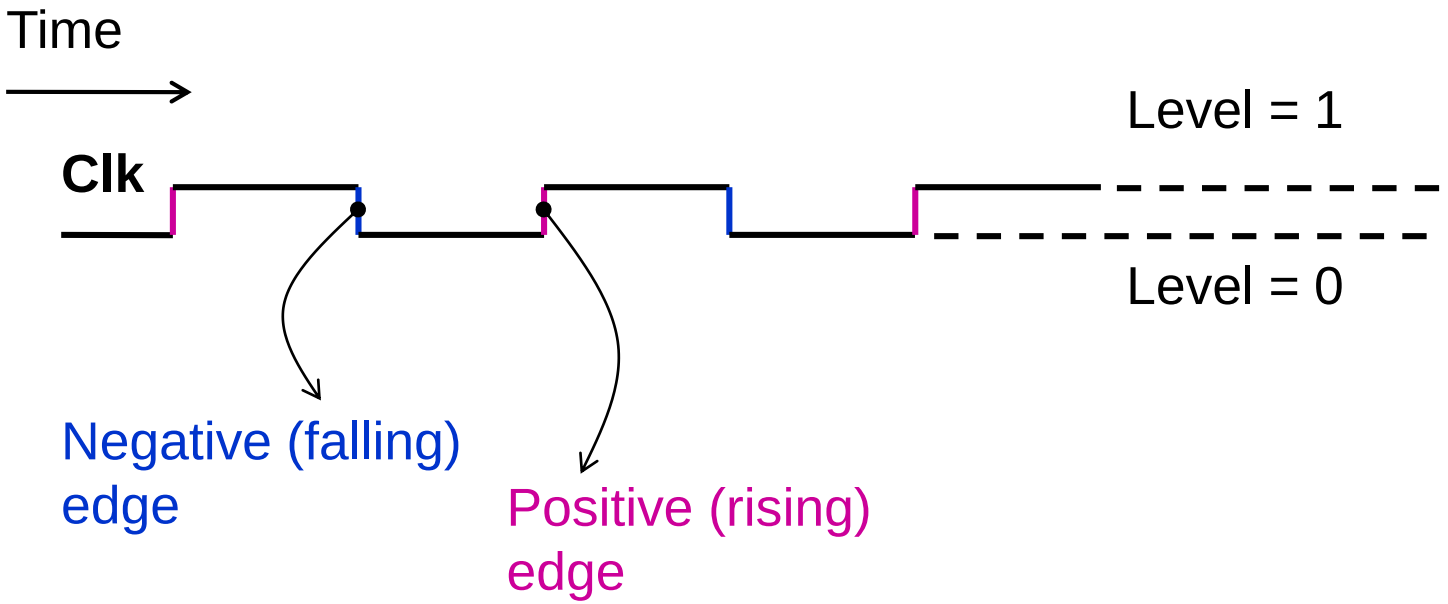
Hold



Flip Flops or Flops (FFs):

Edge-Triggered Memory Cells

(React to **edge** as opposed to **level**)



A clock edge is necessary to write into a FF.

- A FF is a memory cell, so it can be *read* and *written* as well.
- FFs are always in read mode, but each type of FF has its own rules for write operation.
- A **positive-edge triggered** FF needs a positive edge (*active* edge) of clock (as part of its rules of write) to do write operation.
- N**egative-edge triggered** FF: a similar story.
- Assume that all FFs are positive-edge triggered unless otherwise specified.
- By a clock edge we mean an active edge, unless otherwise specified.

If clock is *gated*, FF's content will never change. (No Edge, No Write Operation!)

Clock is a *global signal*, so all write operations into FFs are carried out simultaneously, and synchronized with clock's active edges.

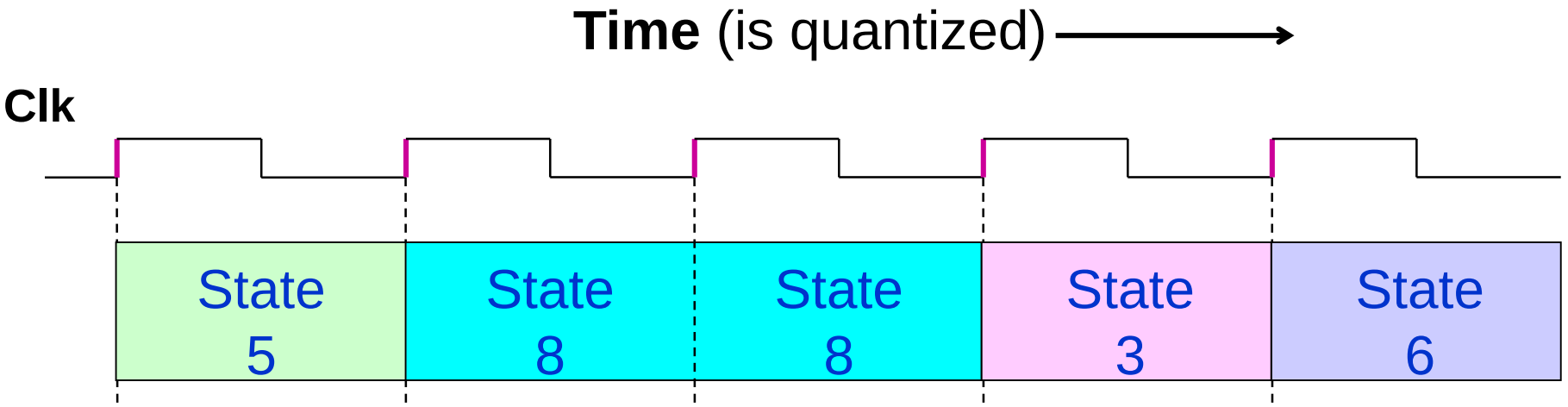
Output of a FF is content or state of that FF; these 3 terms are used interchangeably.

Definition: State of a sequential circuit at a given time is comprised of states of all FFs in that system at that time.

So, in any time instant, a sequential circuit has a state, which is what the circuit remembers from the past.

Conclusion: **State** of a system may only change with an active edge of clock.

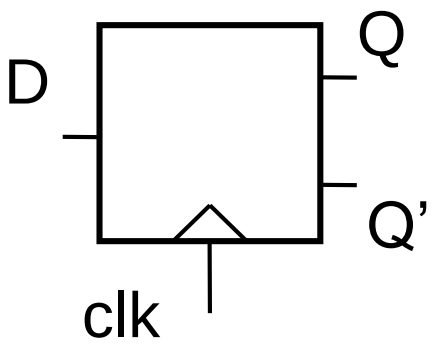
Each state last at least one clock period or one time quantum



We will come back here soon.

D flip-flops
(D-FFs)

D-FF



D	Q^{n+1}
0	0
1	1

First Row: If a clock edge arrives while D=0, the FF will store this value.

Second Row: If a clock edge arrives while D=1, the FF will store this value.

$Q^{n+1} = D$

Symbol

Characteristic Table
(Do not forget Clk)

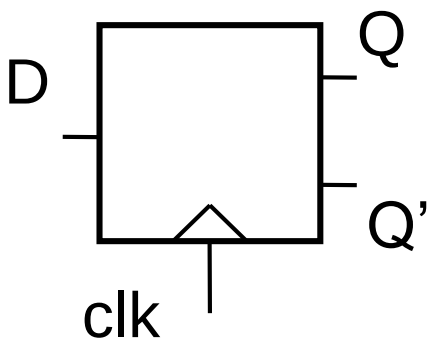
Characteristic Equation

Q^n or simply Q: current state
D: (current) input
Q^{n+1}: next state

When D=0:

- If its content was 0 before the clock edge, it would remain 0 after the clock edge.
- If it was 1 before the clock edge, it would change to 0 after the clock edge.

D-FF



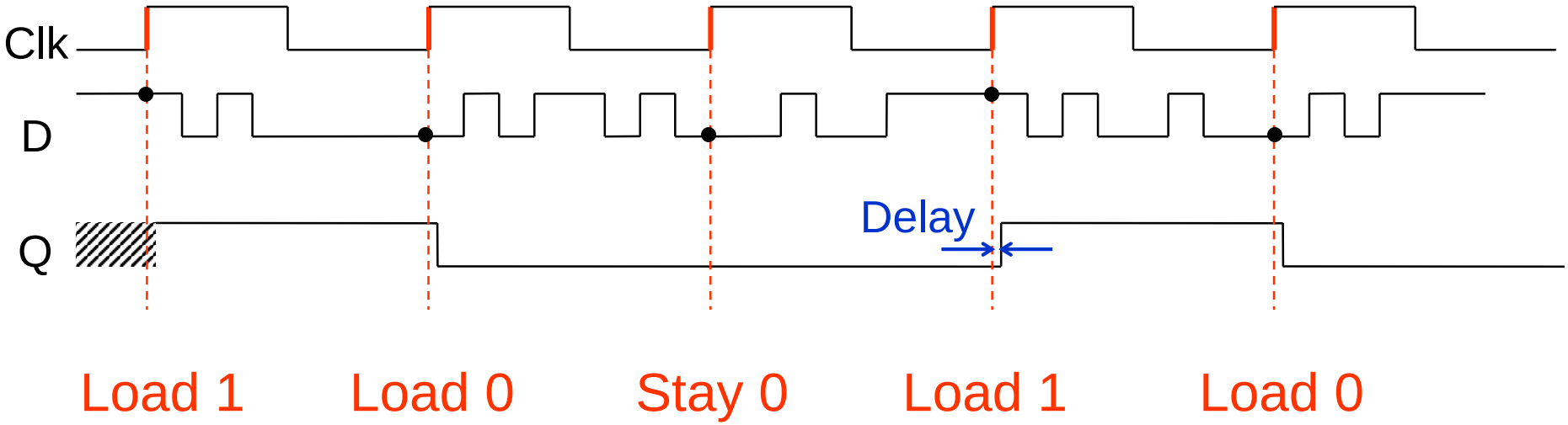
Symbol

Q^n	Q^{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Excitation Table

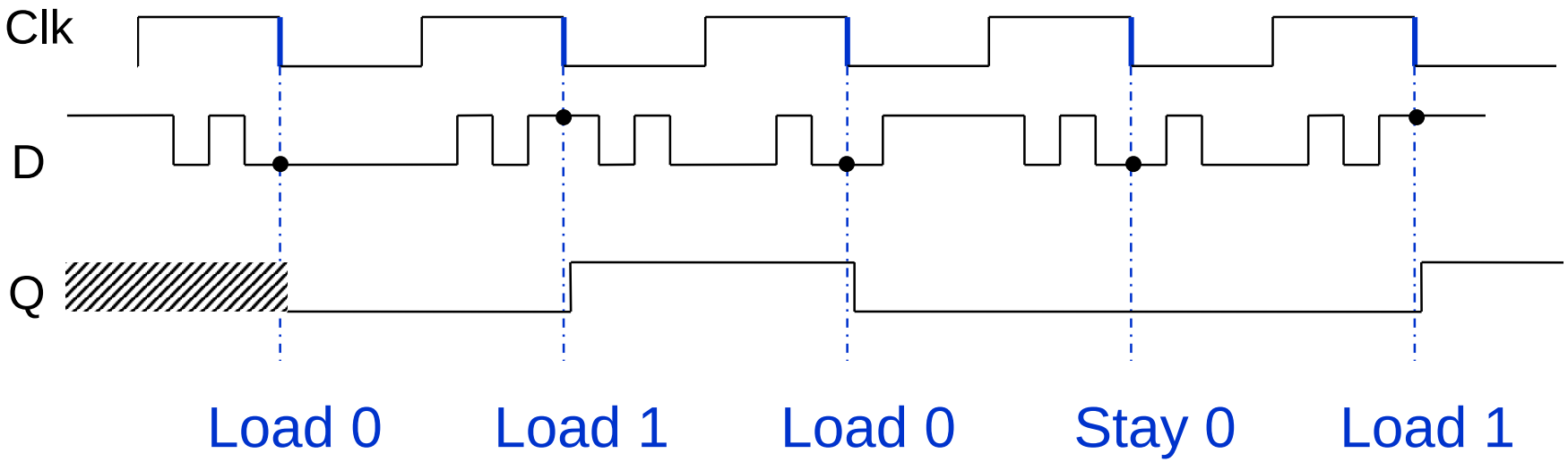
Same amount of information in characteristic table, characteristic equation or excitation table.

Timing Diagram: Positive-Edge-Triggered D-FF



- If a clock edge arrives while D=1, the FF will store this value.
- If a clock edge arrives while D=0, the FF will store this value.

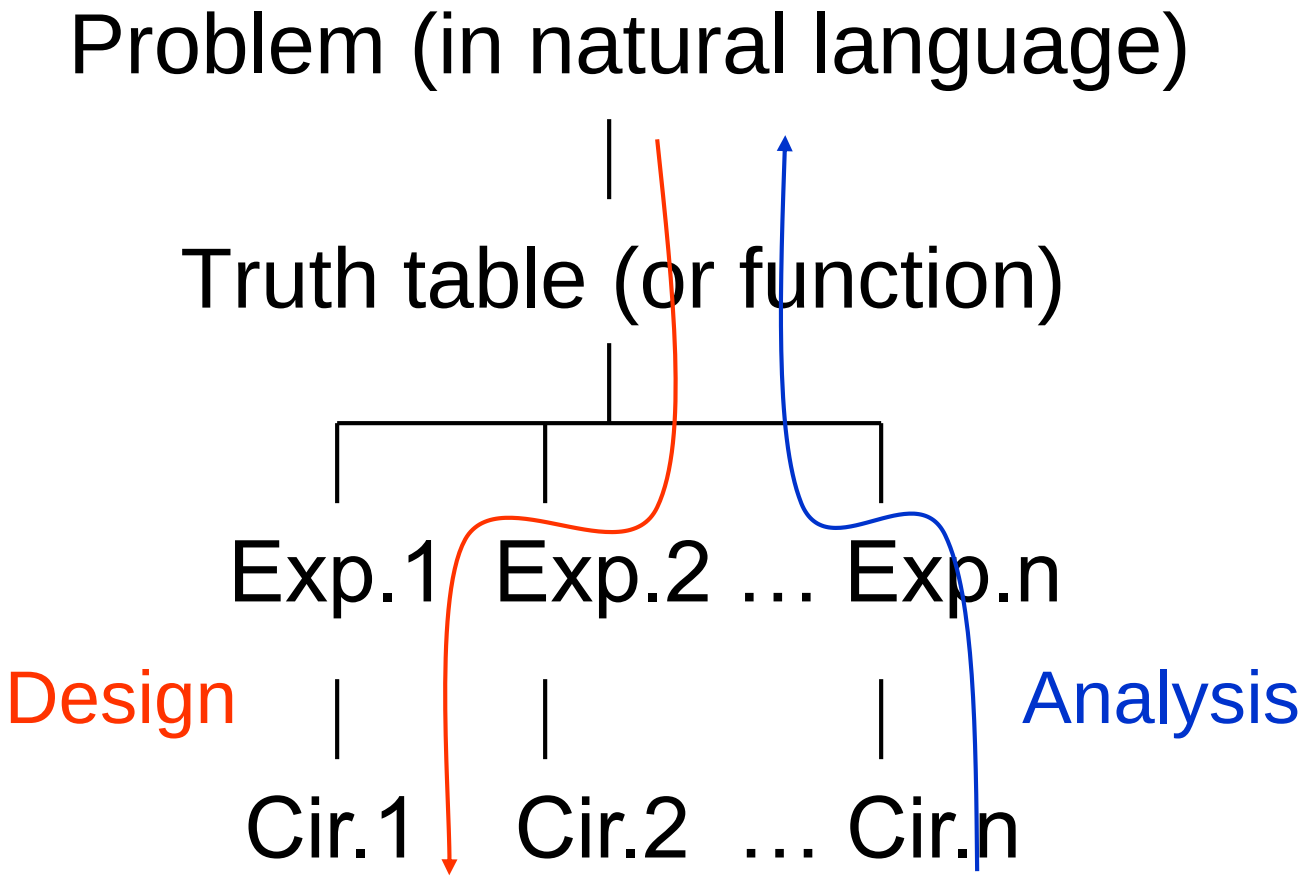
Timing Diagram: Negative-Edge-Triggered D-FF

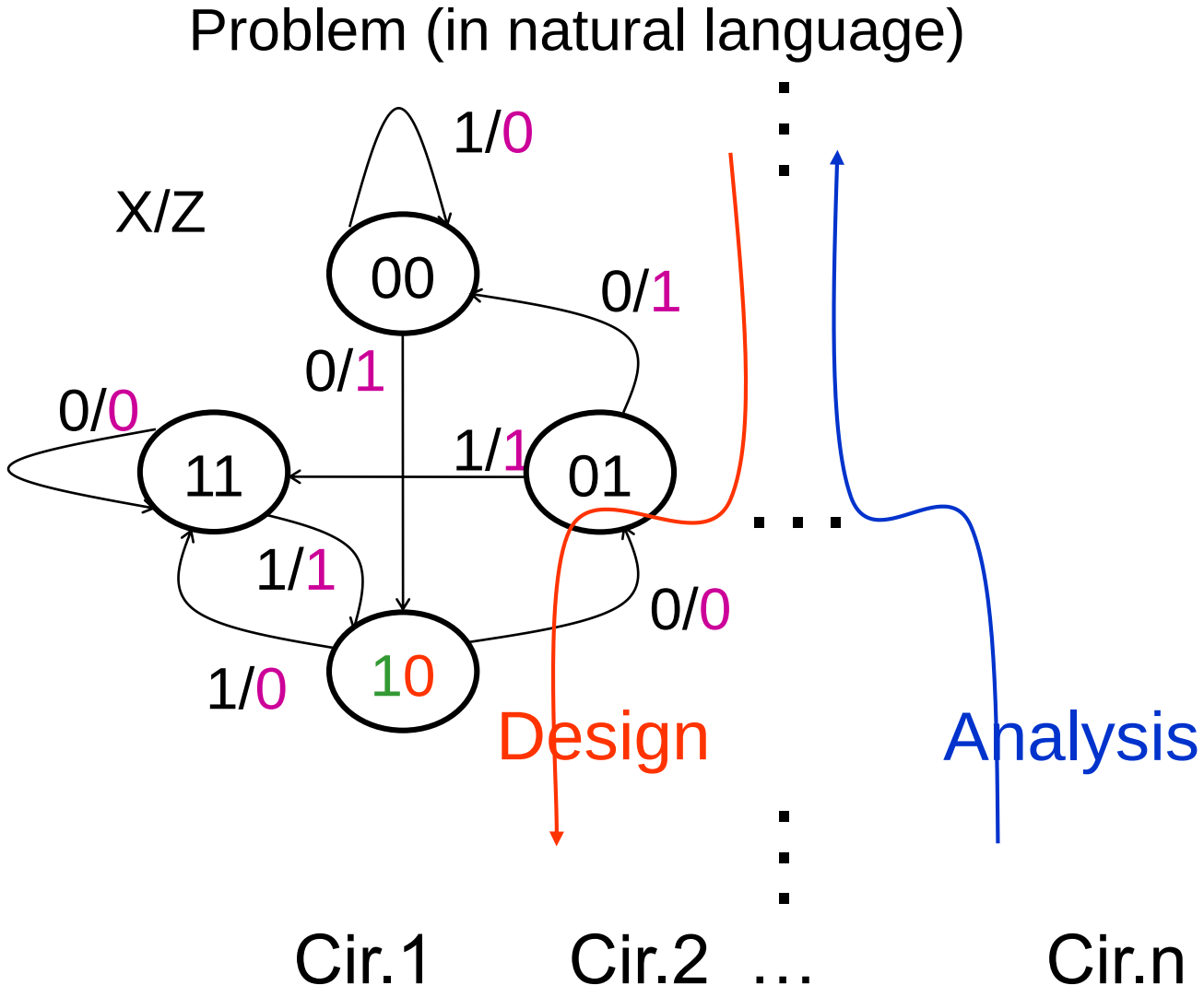


- If a clock edge arrives while D=1, the FF will store this value.
- If a clock edge arrives while D=0, the FF will store this value.

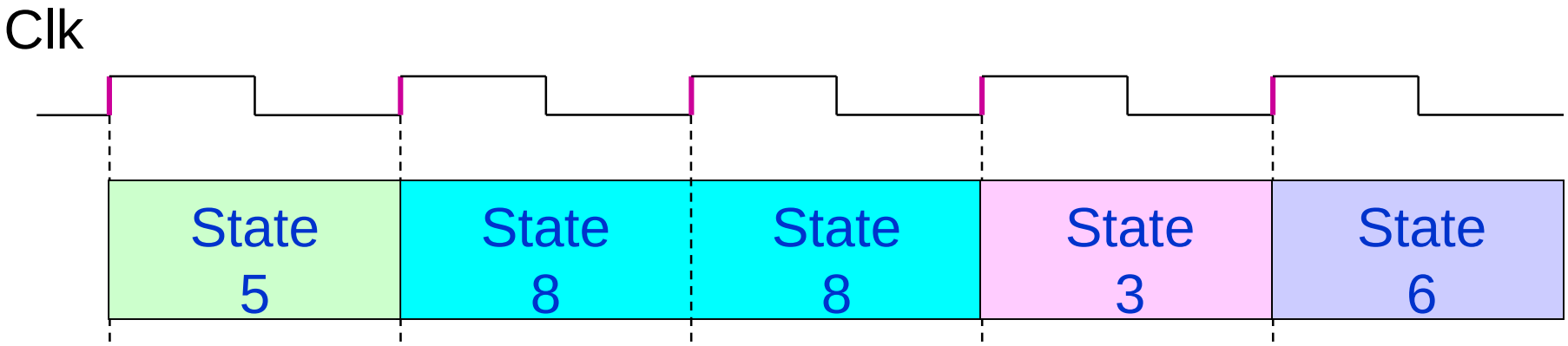
Analysis of Sequential Circuits

Remember: (in combinational logic)

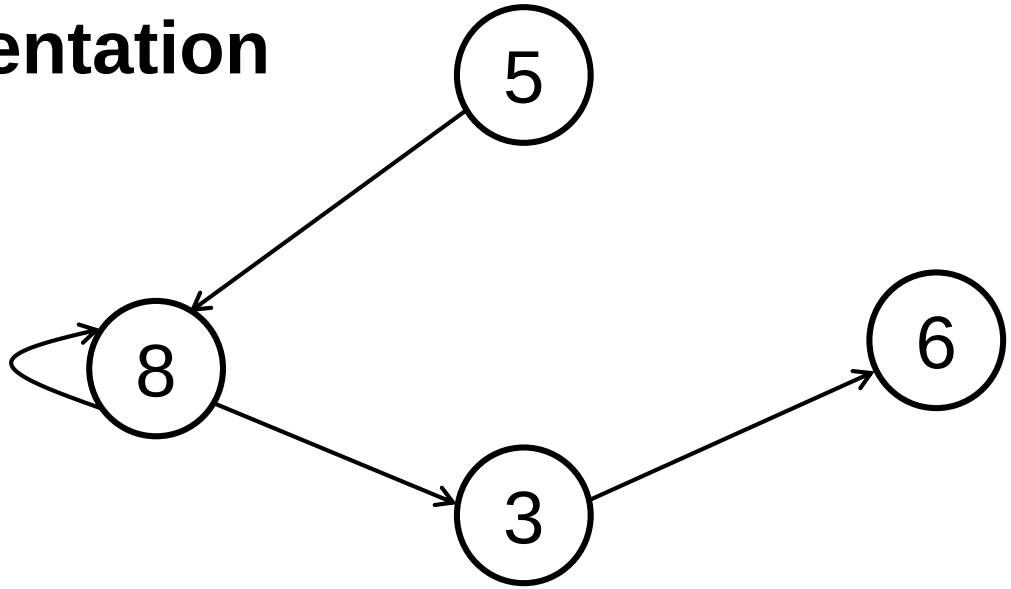




In analysis, the goal is to obtain a state graph or state diagram, the highest level of description for state machines

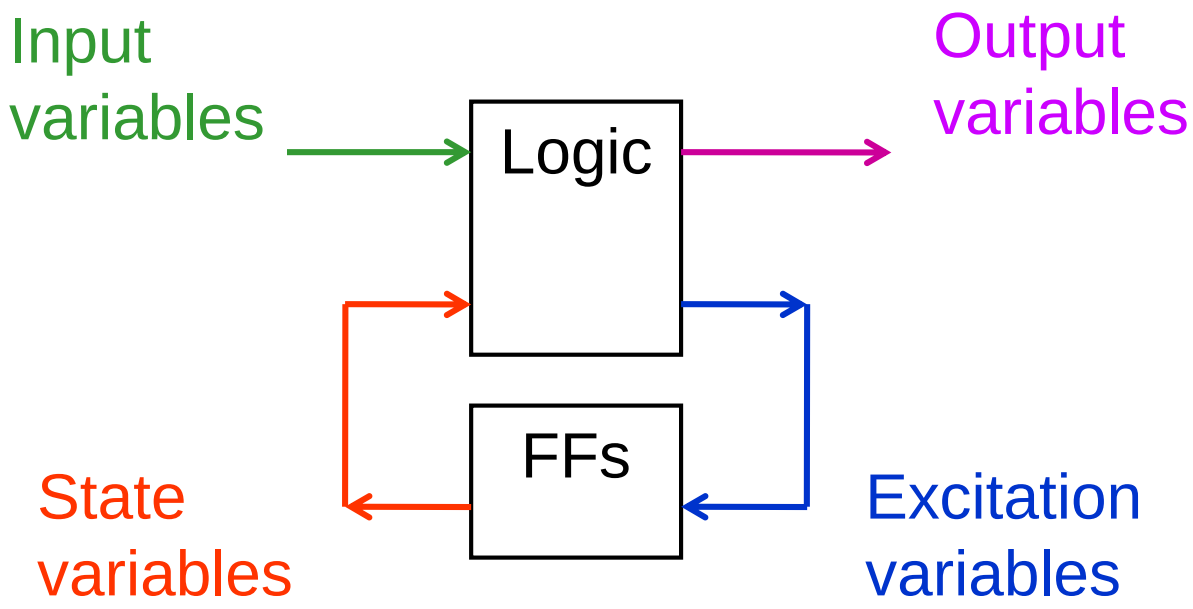


Different Representation



Notice the concept of choice: 8 to 8, or 8 to 3

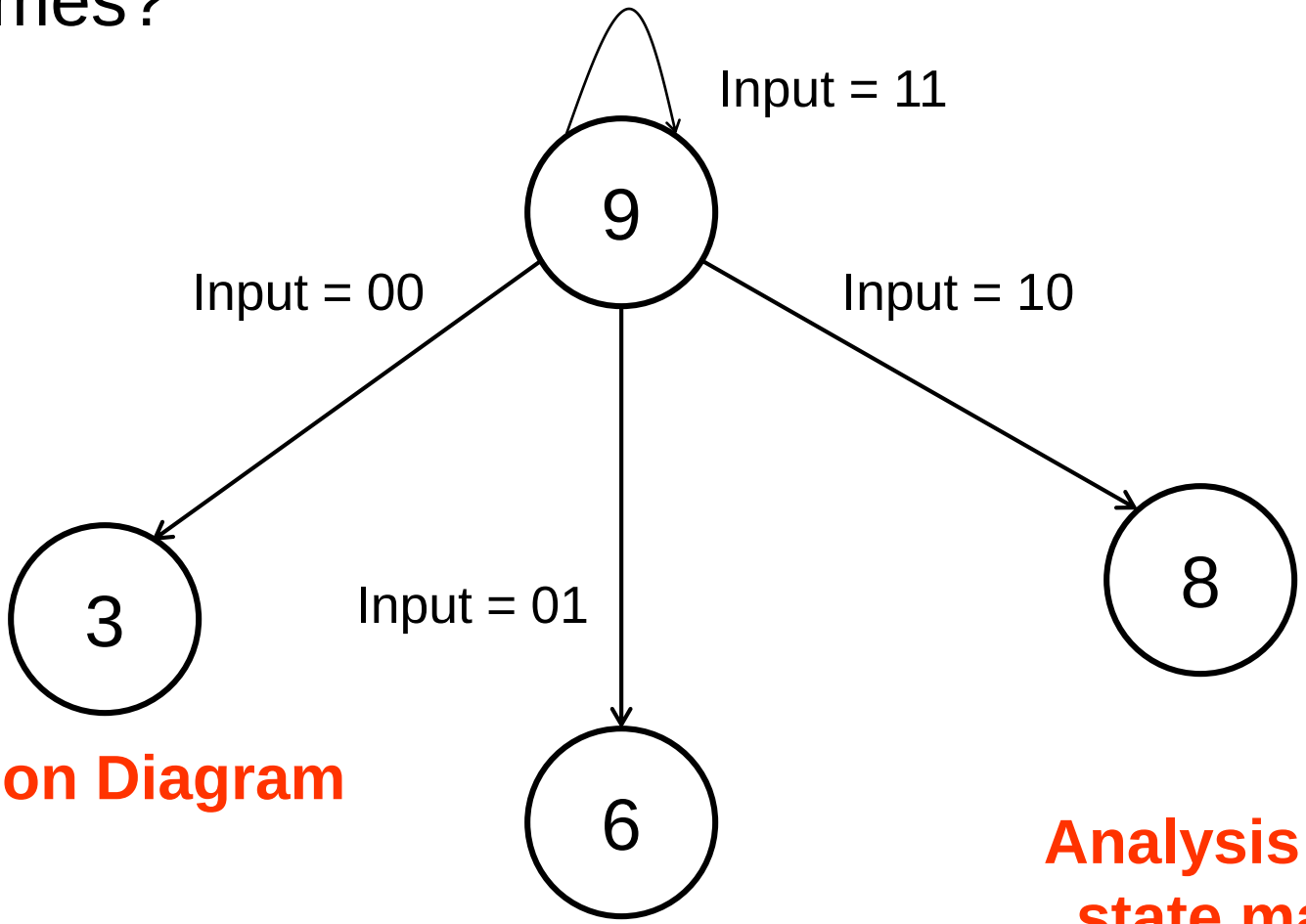
Finite-State Machines



Number of states = $2^{\#\text{ of FFs}}$

We need to answer this question:

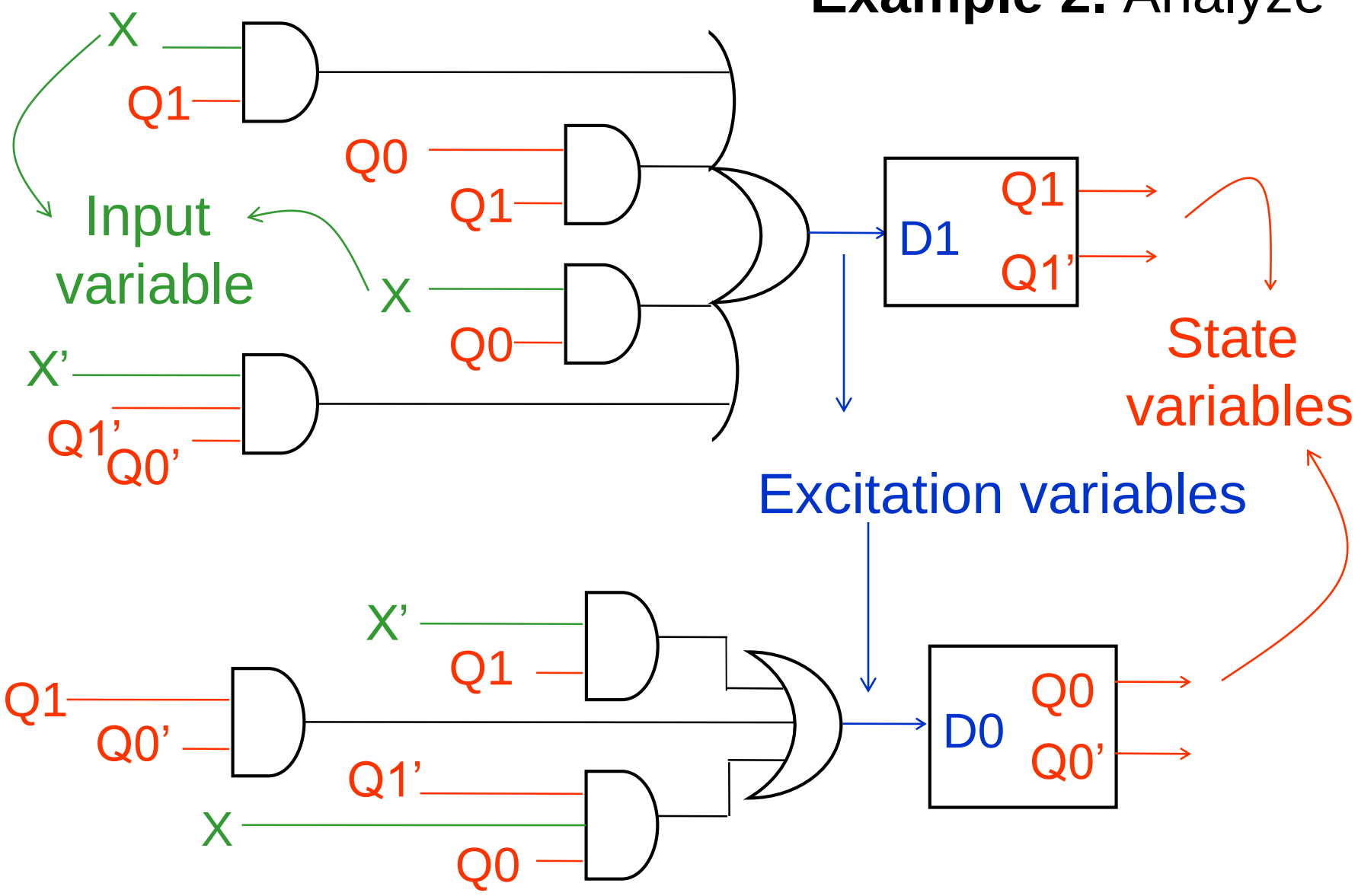
Starting with *any* state, such as 9, which state(s) may immediately follow 9 when next clock edge comes?

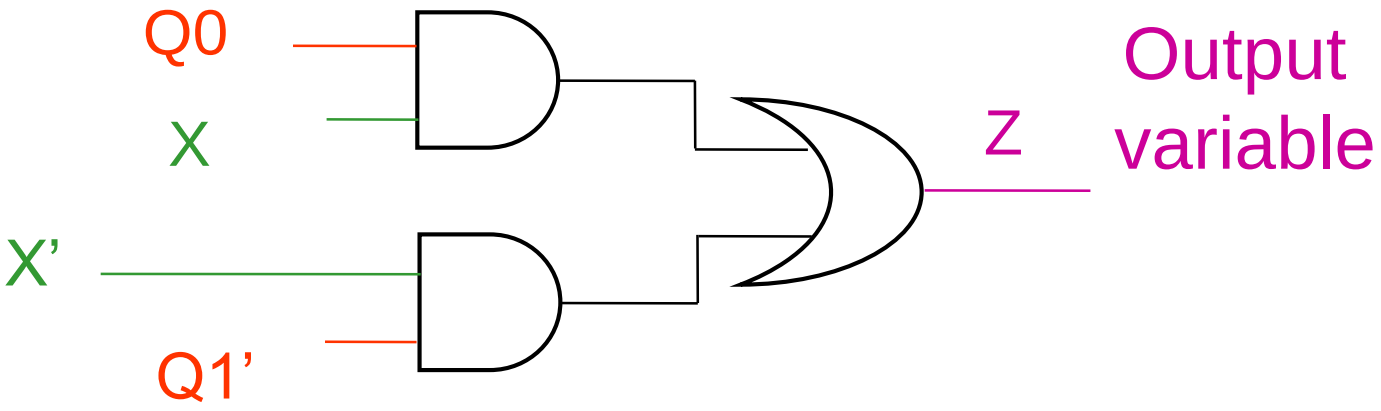


Transition Diagram

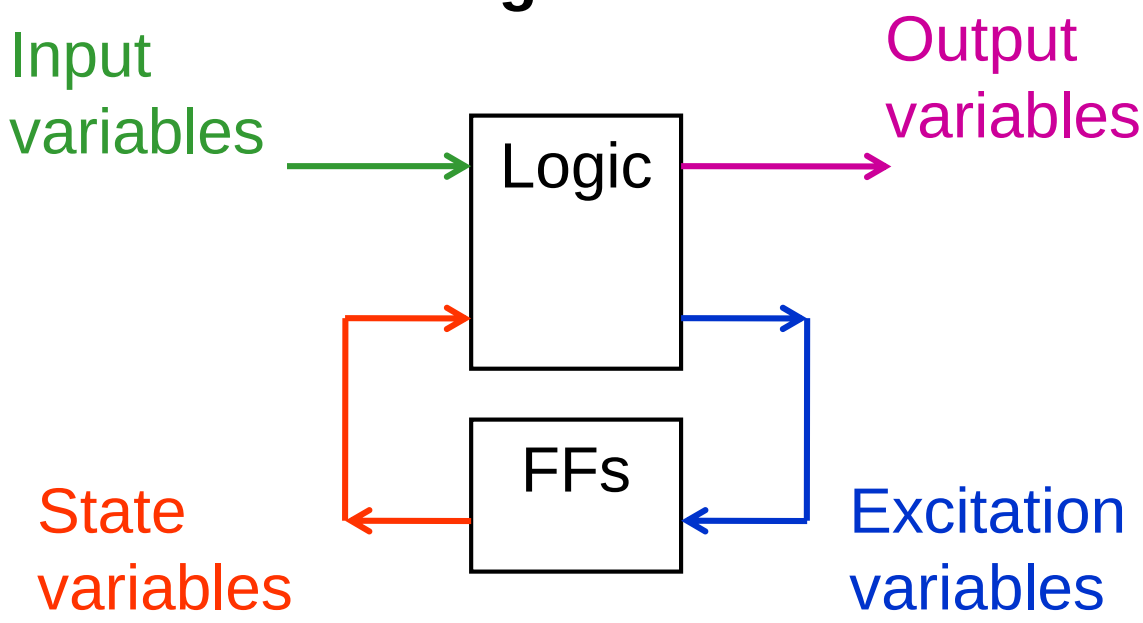
Analysis of finite state machines

Example 2. Analyze

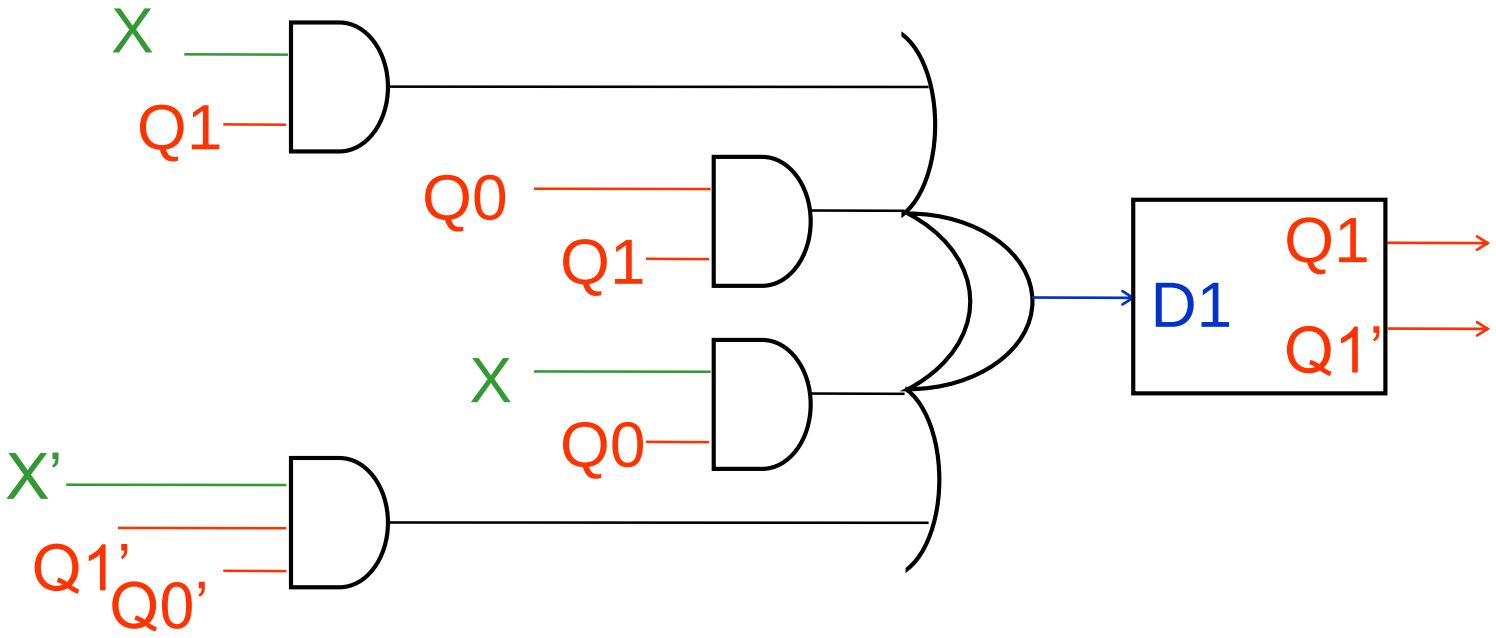



Q0
X
X'
Q1'
Z
Output
variable

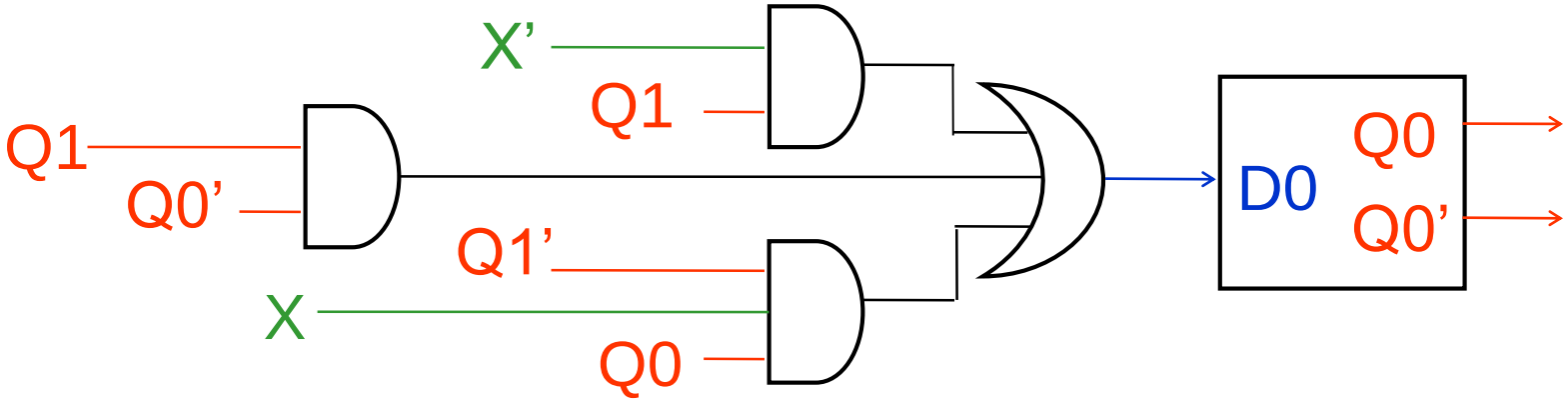
In general


Input
variables
Logic
Output
variables
FFs
State
variables
Excitation
variables

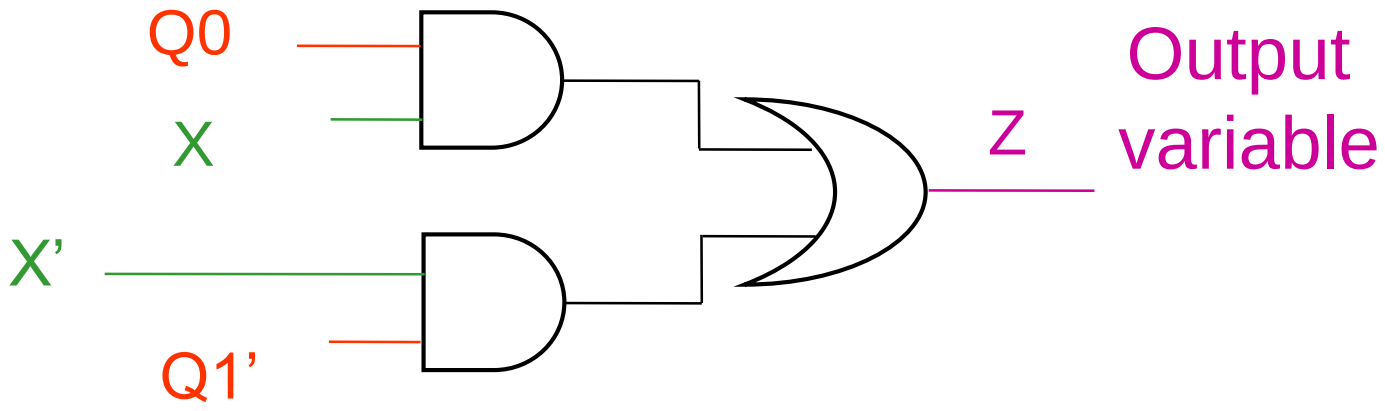
1) Excitation equations & output equation



$D1 = X.Q1 + Q1.Q0 + X.Q0 + X'.Q1'.Q0'$



$D0 = X'.Q1 + Q1.Q0' + X.Q1'.Q0$



$$Z = X.Q0 + X'.Q1'$$

2) Use excitation & output equations to obtain …

Excitation maps

Q1Q0 \ X	0	1
00	**1**	0
01	0	1
11	1	1
10	**0**	1

D1

Q1Q0 \ X	0	1
00	0	0
01	0	**1**
11	1	0
10	1	1

D0

Output map

Q1Q0 \ X	0	1
00	1	0
01	1	1
11	0	1
10	0	0

Z

D1 = X.Q1 + Q1.Q0 + X.Q0 + X'.Q1'.Q0'

D0 = X'.Q1 + Q1.Q0' + X.Q1'.Q0

Z = X.Q0 + X'.Q1'

Q1Q0 \ X	0	1
00	**1**	0
01	0	1
11	1	1
10	**0**	1

D1

Q1Q0 \ X	0	1
00	0	0
01	0	**1**
11	1	0
10	1	1

D0

$Q^{n+1} = D$

$Q^{n+1} = D$

Copy

Copy

3) Partial Transition Tables

Q1Q0 \ X	0	1
00	**1**	0
01	0	1
11	1	1
10	**0**	1

$Q1^{n+1}$

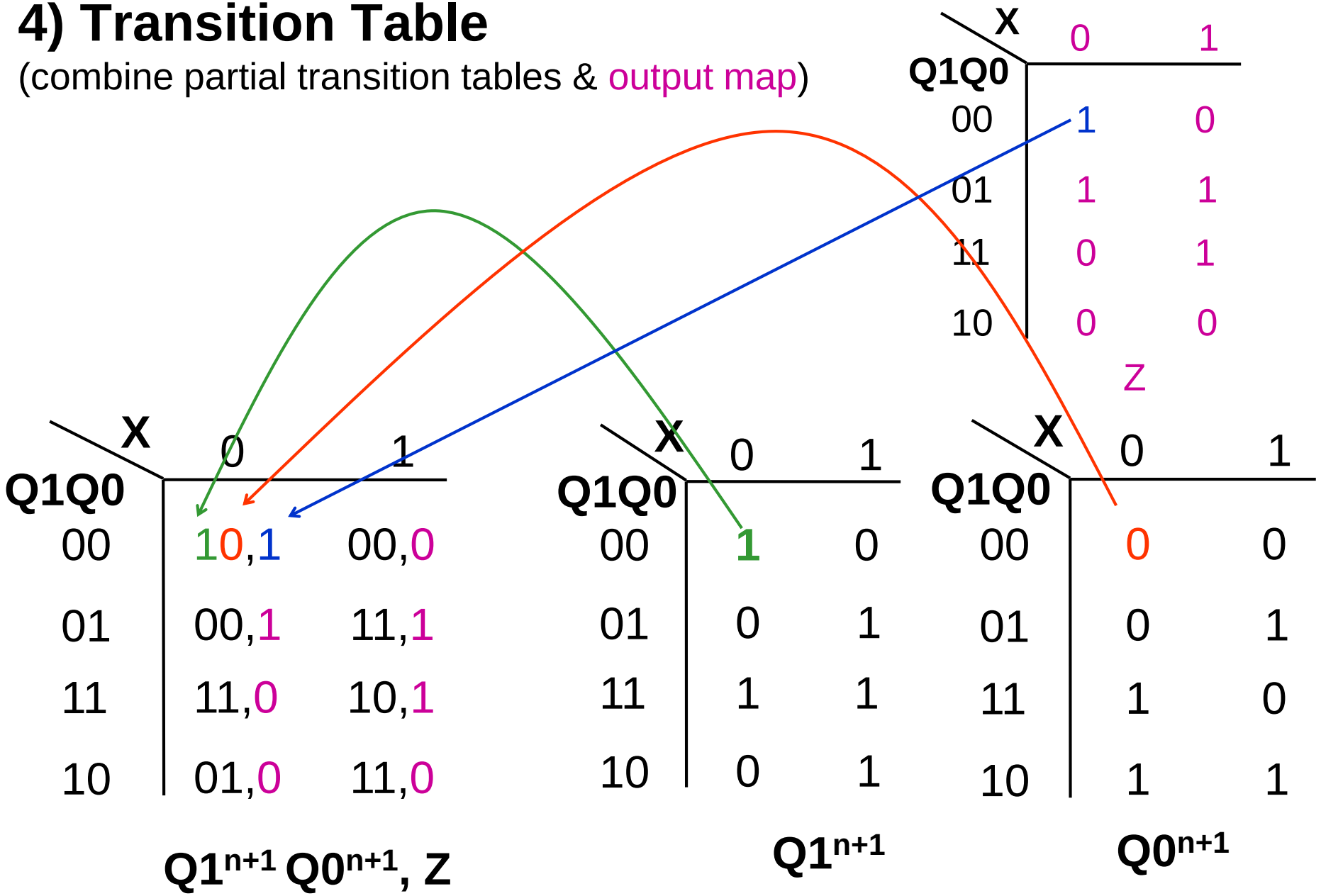
Q1Q0 \ X	0	1
00	0	0
01	0	**1**
11	1	0
10	1	1

$Q0^{n+1}$

For D-FFs, partial transition tables = excitation maps

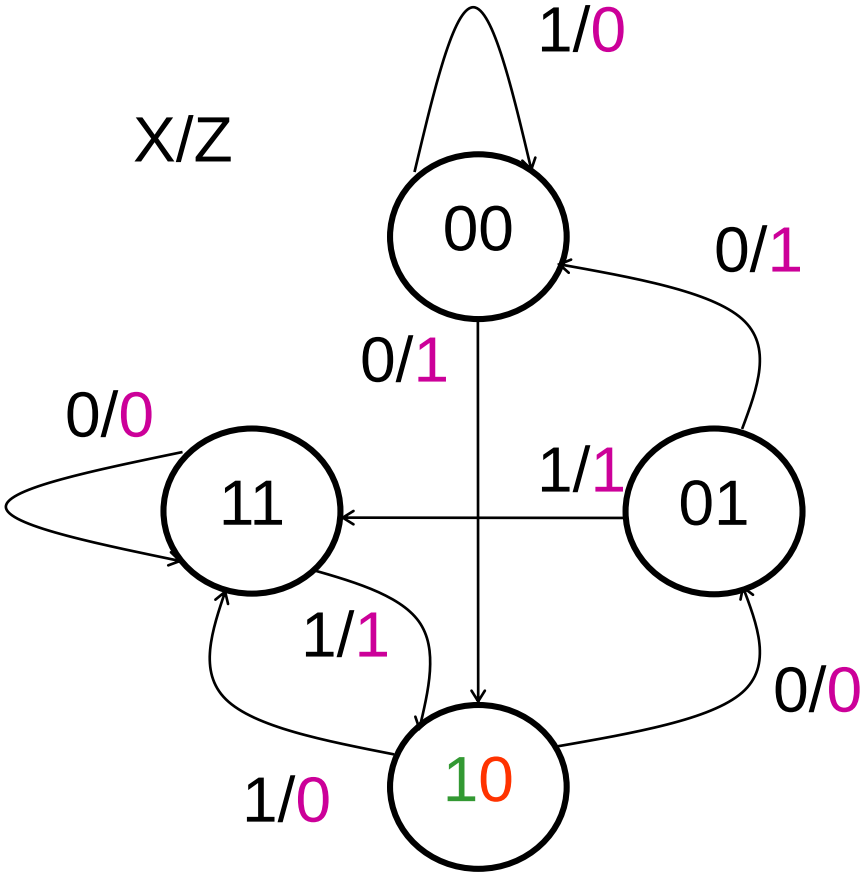
4) Transition Table

(combine partial transition tables & output map)



5) Transition Diagram or Graph

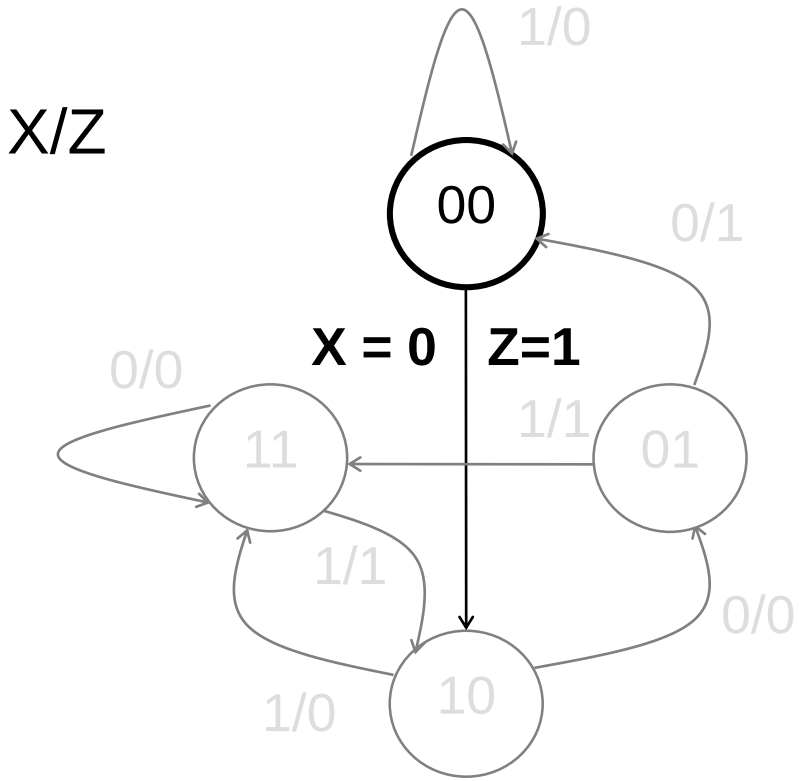
(graphical representation)



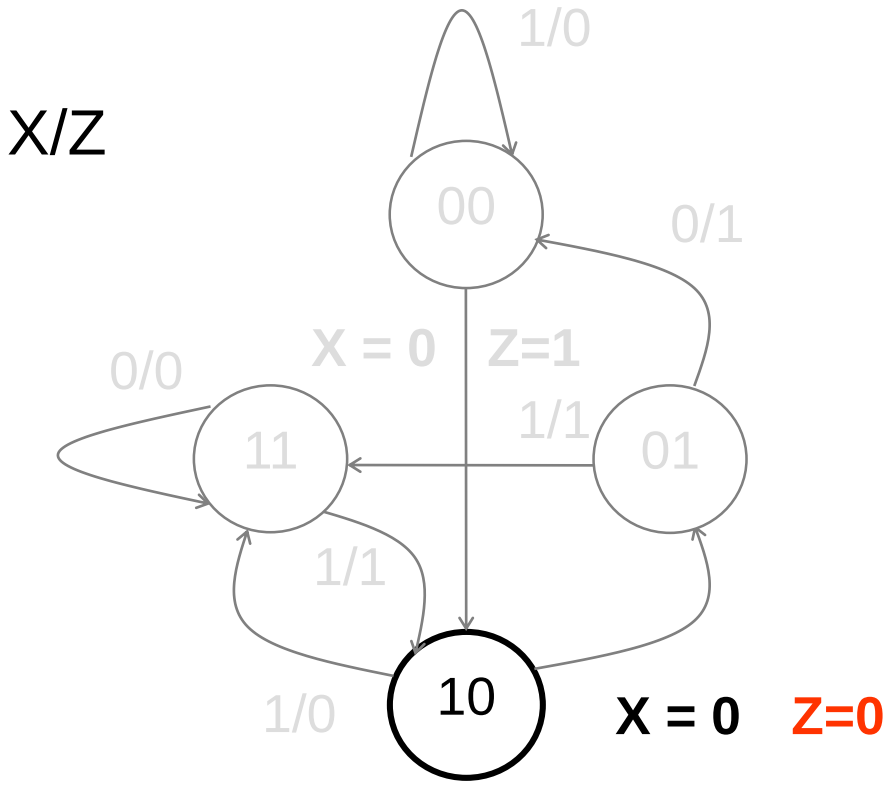
X / Q1Q0	0	1
00	10,1	00,0
01	00,1	11,1
11	11,0	10,1
10	01,0	11,0

$Q1^{n+1}$ $Q0^{n+1}$, Z

5) Transition Diagram …



5) Transition Diagram …

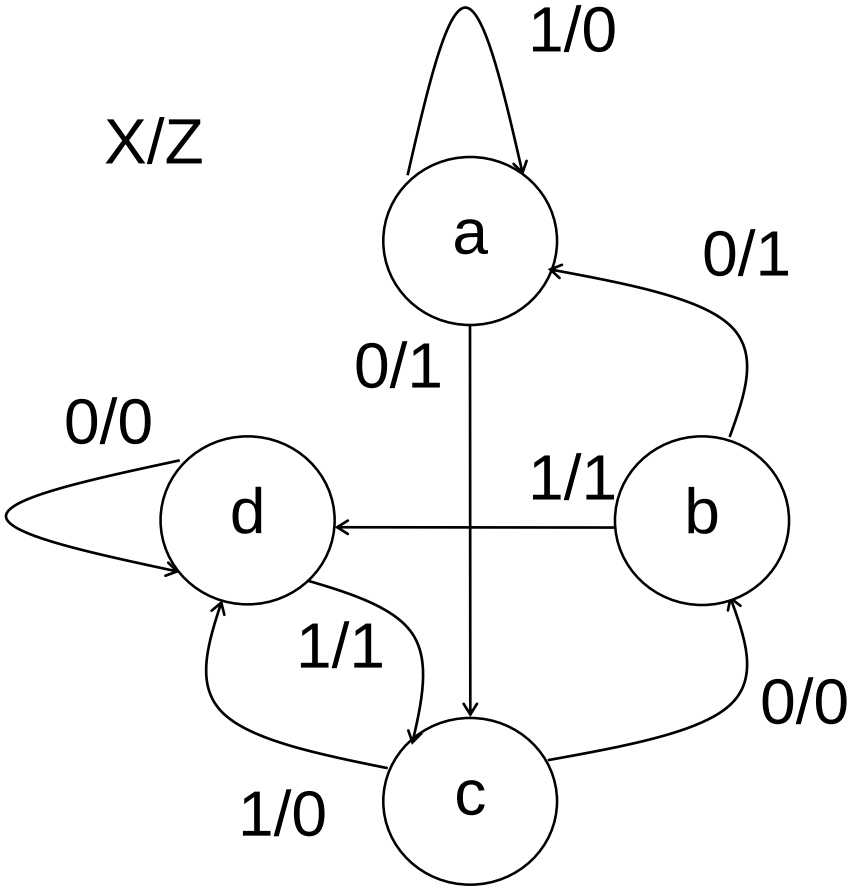


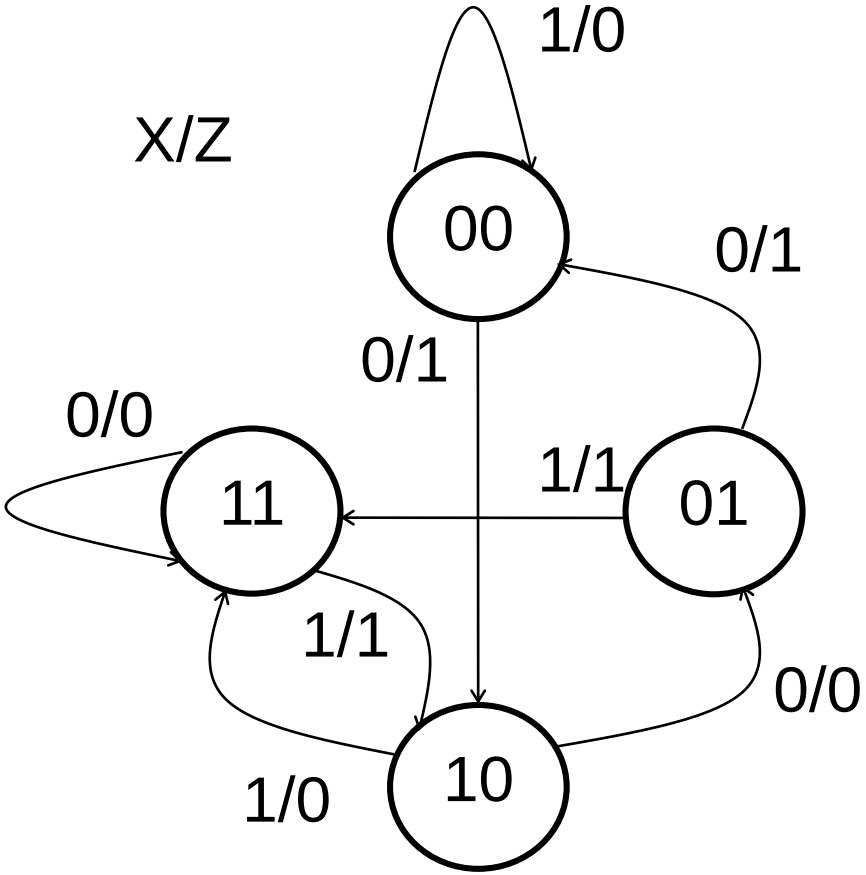
6) State Table (assign arbitrary symbolic codes)

Q1Q0	Q	X = 0	X = 1
00	a	c, 1	a, 0
01	b	a, 1	d, 1
11	d	d, 0	c, 1
10	c	b, 0	d, 0

Q^{n+1}, Z

7) State Diagram (graphical representation)





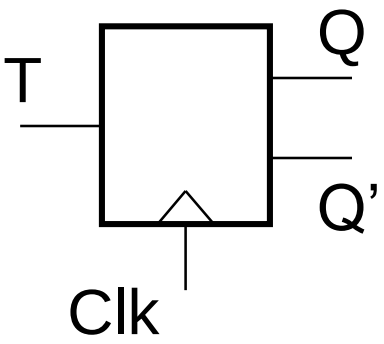
00	01	10	11
a	b	c	d

T-FFs

(Toggle FFs)

T-FF

Symbol



Characteristic Table

T	Q^{n+1}
0	Q^n
1	Q'^n

Characteristic Equation

$$Q^{n+1} = T\,Q'^n + T'\,Q^n$$

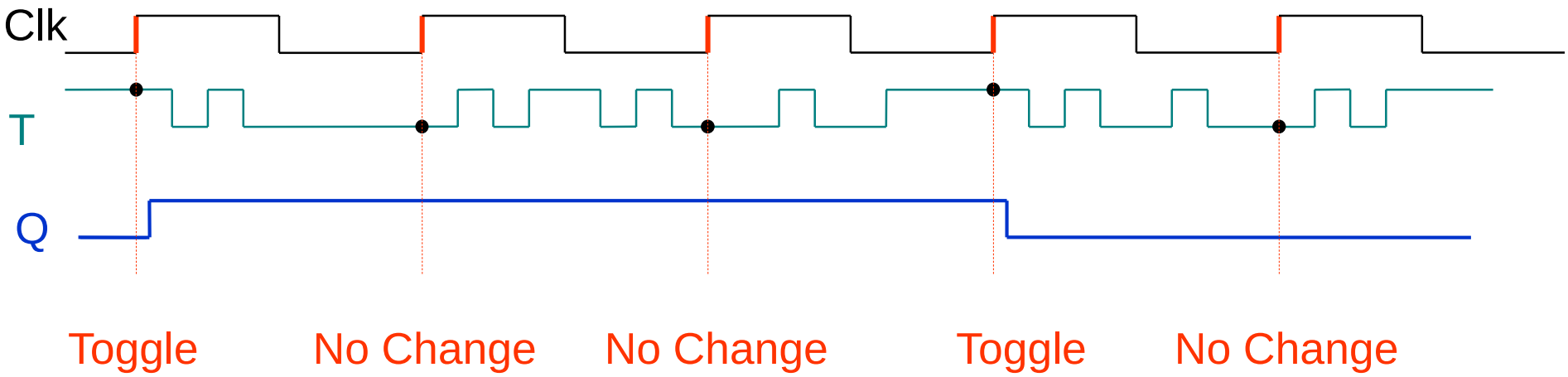
First Row: If a clock edge arrives while T=0, then the content of FF remains unchanged.

Second Row: If a clock edge arrives while T=1, then the content of FF will change or toggles.

Excitation Table

Q^n	Q^{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Timing diagram: positive-edge-triggered T-FF



First Row: If a clock edge arrives while T=0, then the content of FF remains unchanged.

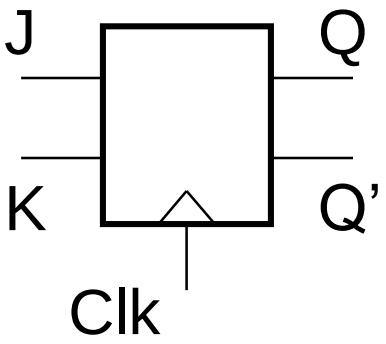
Second Row: If a clock edge arrives while T=1, then the content of FF will change or toggles.

END

JK-FFs

JK-FF

Symbol



Characteristic Table

J	K	Q^{n+1}	Mode
0	0	Q^n	NC
0	1	0	Reset
1	0	1	Set
1	1	Q'^n	Toggle

Characteristic Equation

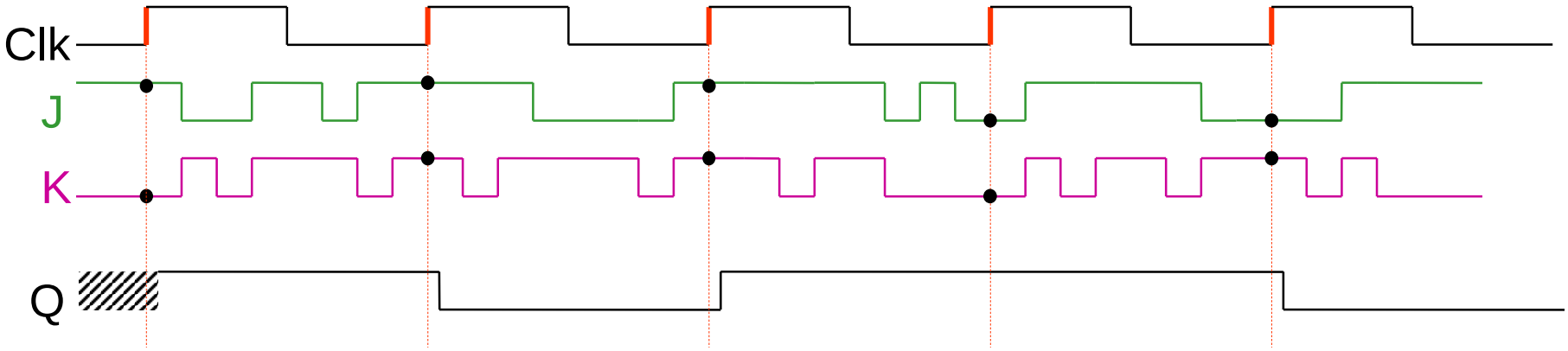
$$Q^{n+1} = K' Q^n + J Q'^n$$

J: Set
K: Reset

Q^n	Q^{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Excitation Table

Timing diagram: positive-edge-triggered JK-FF



Set Toggle Toggle No Change Reset

J	K	Q^{n+1}	Mode
0	0	Q^n	NC
0	1	0	Reset
1	0	1	Set
1	1	Q'^n	Toggle