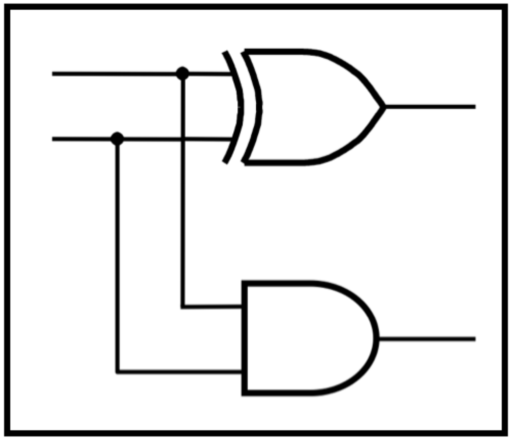


CprE 2810: Digital Logic

Instructor: Alexander Stoytchev

http://www.ece.iastate.edu/~alexs/classes/

Registers

CprE 2810: Digital Logic
Iowa State University, Ames, IA

Administrative Stuff

- **Homework 8 is due on Monday Oct 27 @ 10pm.**
- **Homework 9 will be due on Monday Nov 3 @ 10pm.**
- **The second midterm exam is next week (Friday October 31).**

Administrative Stuff

- Midterm Exam #2

- When: Friday, October 31 @ 4:20pm.

- Where: This room

- What: Chapters 1, 2, 3, 4 and 5

- The exam will be closed book but open notes
(you can bring up to 3 sheets of handwritten notes).

Midterm 2: Format

- The exam will be out of 130 points
- You need 95 points to get an A for this exam
- It will be great if you can score more than 100 points.
  - but you can't roll over your extra points ☹

Midterm 2: Topics

- K-maps for 2, 3, and 4 variables
- Binary Numbers and Hexadecimal Numbers
- 1's complement and 2's complement representation
- Addition and subtraction of binary numbers
- Circuits for adders and fast adders, delay calculation

- Single and Double precision IEEE floating point formats
- Converting a real number to the IEEE format
- Converting a floating point number to base 10

- Multiplexers (circuits and function)
- Synthesis of logic functions using multiplexers
- Shannon's Expansion Theorem

Midterm 2: Topics

- Decoders (circuits and function)
- Demultiplexers
- Encoders (binary and priority)
- Code Converters and Comparison Circuits

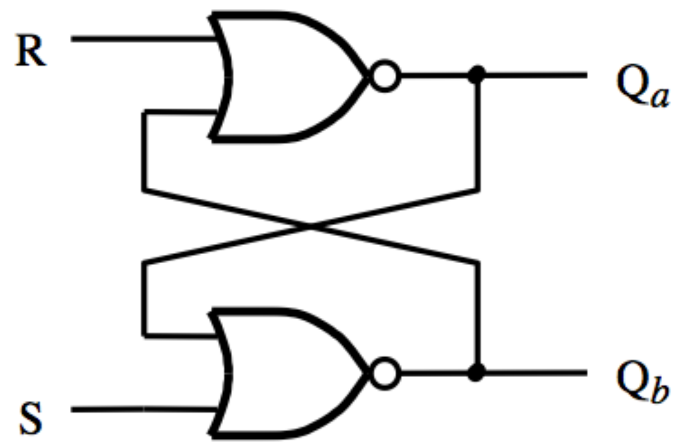
- Synthesis of logic circuits using adders, multiplexers, encoders, decoders, and basic logic gates
- Synthesis of logic circuits given constraints on the available building blocks that you can use

- Latches (circuits, behavior, timing diagrams)
- Flip-Flops (circuits, behavior, timing diagrams)
- Registers and Register Files
- Counters
- Something from Star Wars

Quick Review

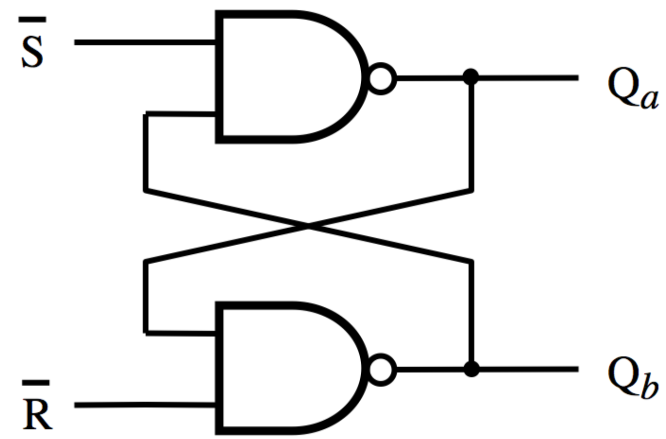
Basic Latch with NAND Gates

Basic Latch
(with NOR Gates)



SR Latch

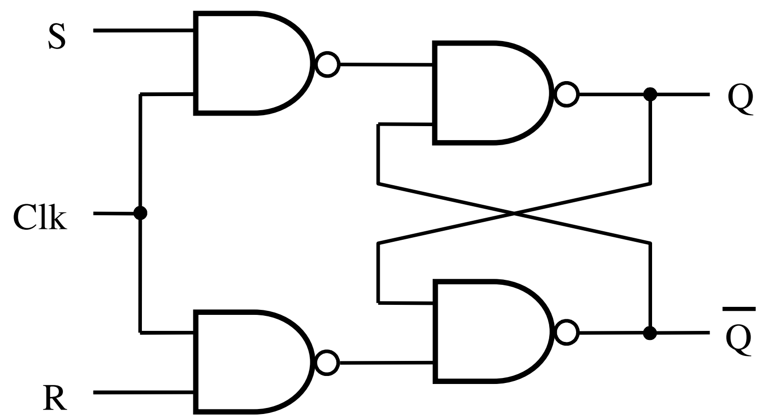
Basic Latch
(with NAND Gates)



$\overline{S}\,\overline{R}$ Latch

Gated SR Latch

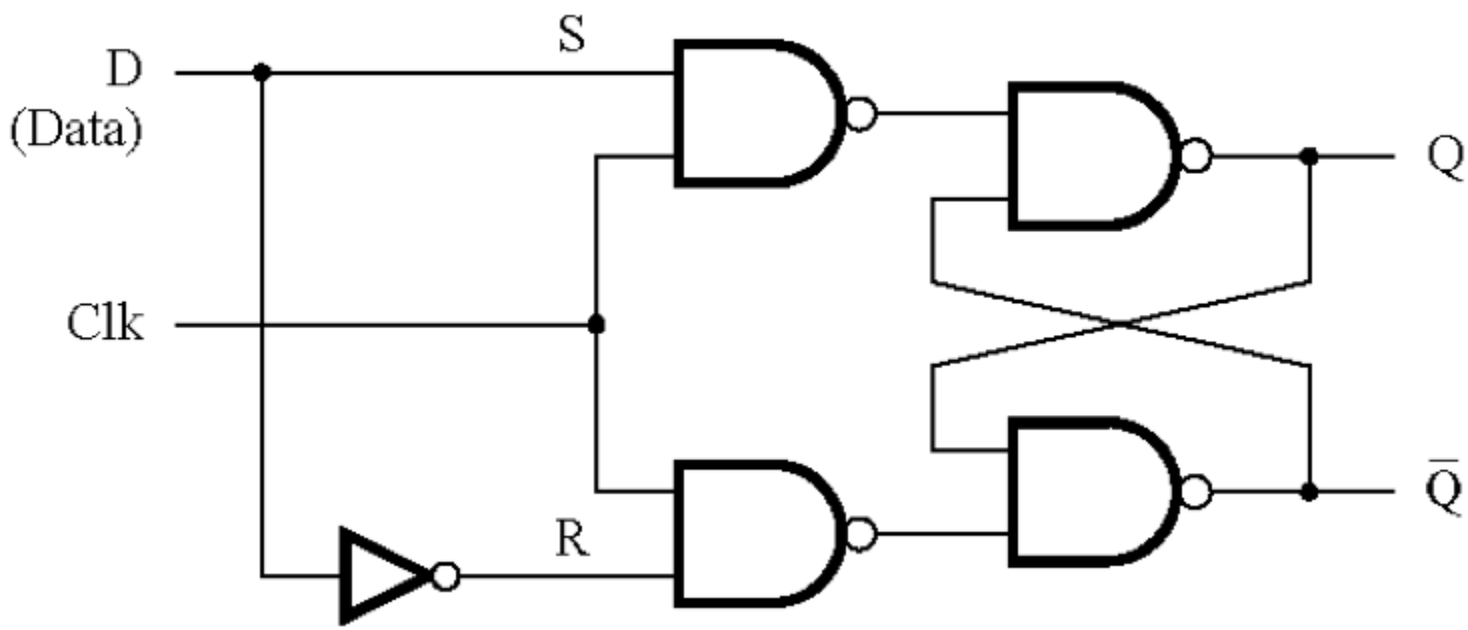
Gated SR latch with NAND gates



[Figure 5.6 from the textbook]

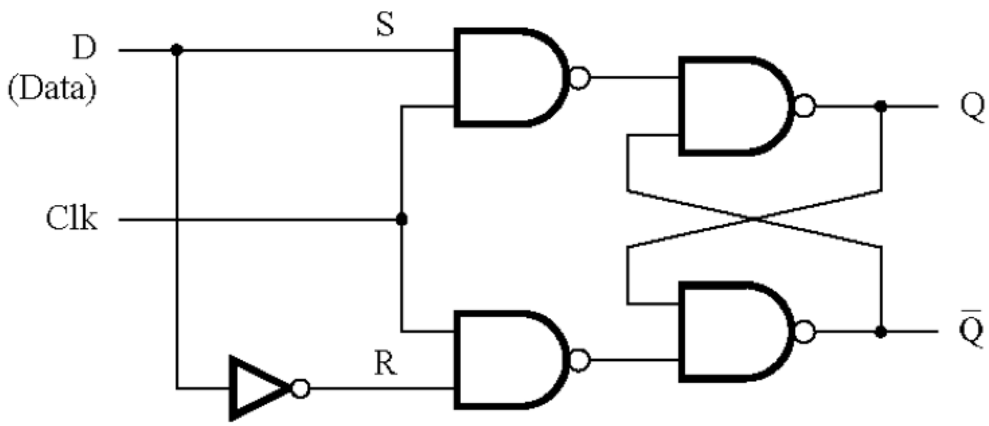
Gated D Latch

Circuit Diagram for the Gated D Latch



[Figure 5.7a from the textbook]

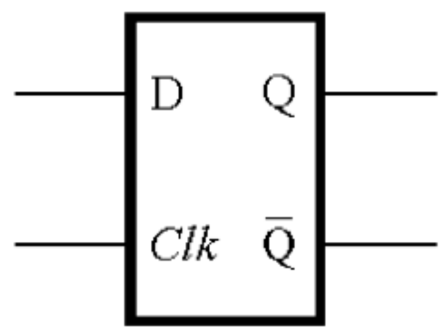
Gated D Latch: Circuit Diagram, Characteristic Table, and Graphical Symbol



Clk	D	$Q(t+1)$
0	x	$Q(t)$
1	0	0
1	1	1

Note that it is now impossible to have S=R=1.

When Clk=1 the output follows the D input.
When Clk=0 the output cannot be changed.

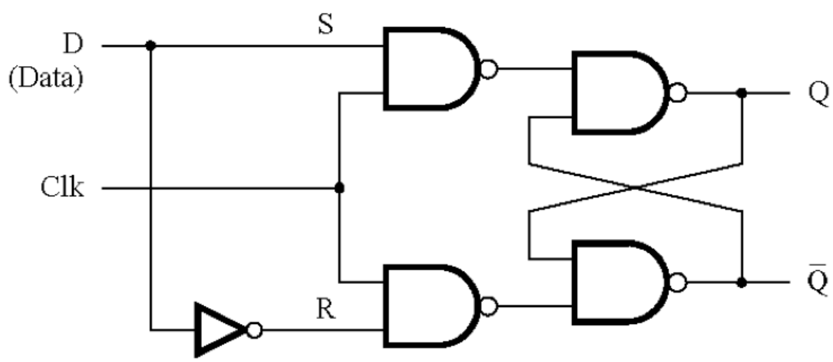


[Figure 5.7a,b from the textbook]

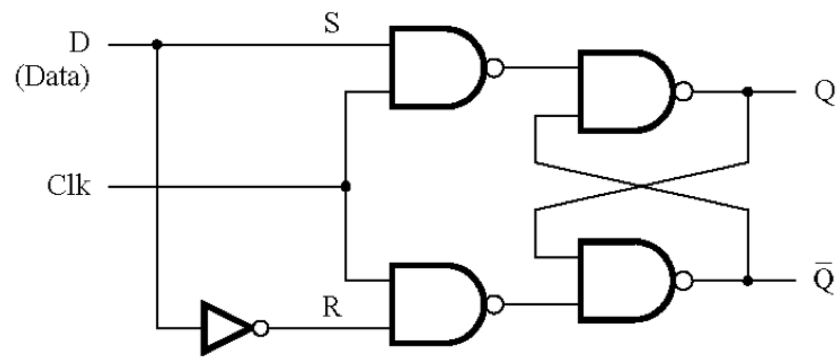
D Flip-Flop

It takes two D latches to construct one D flip-flop

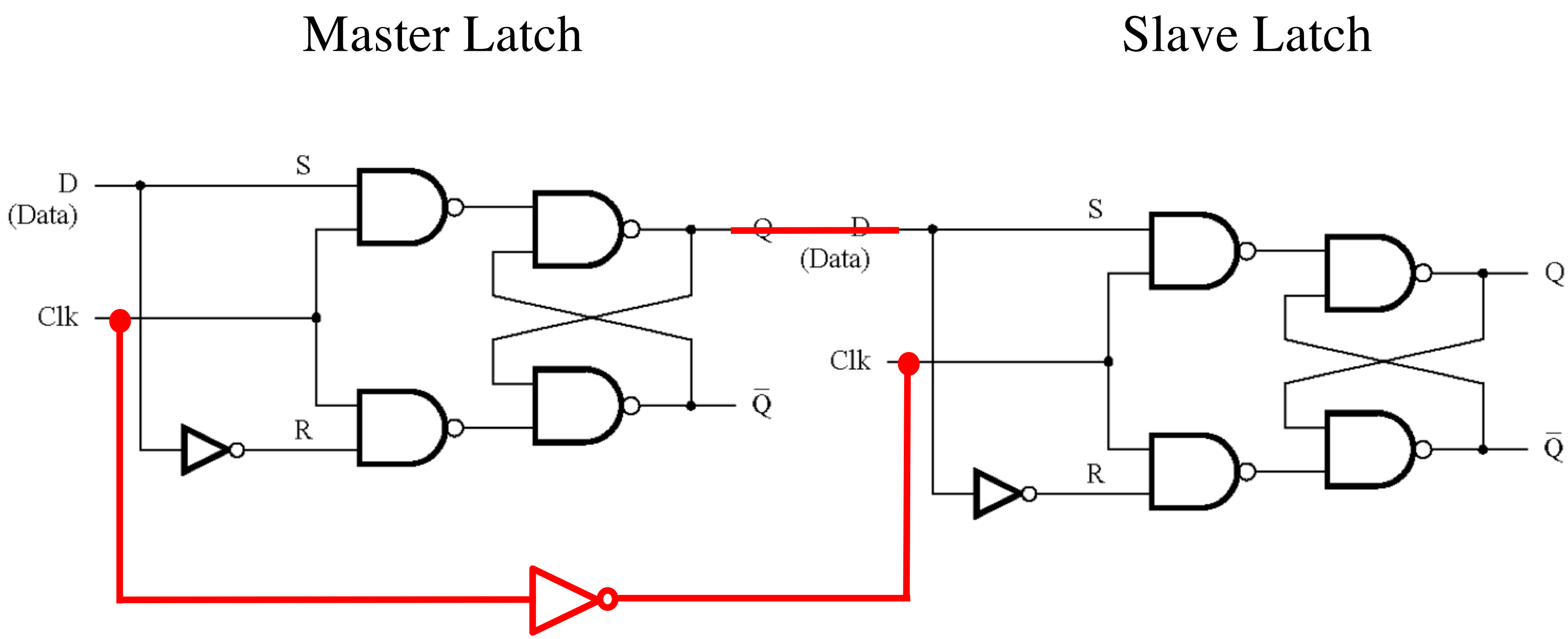
Master Latch



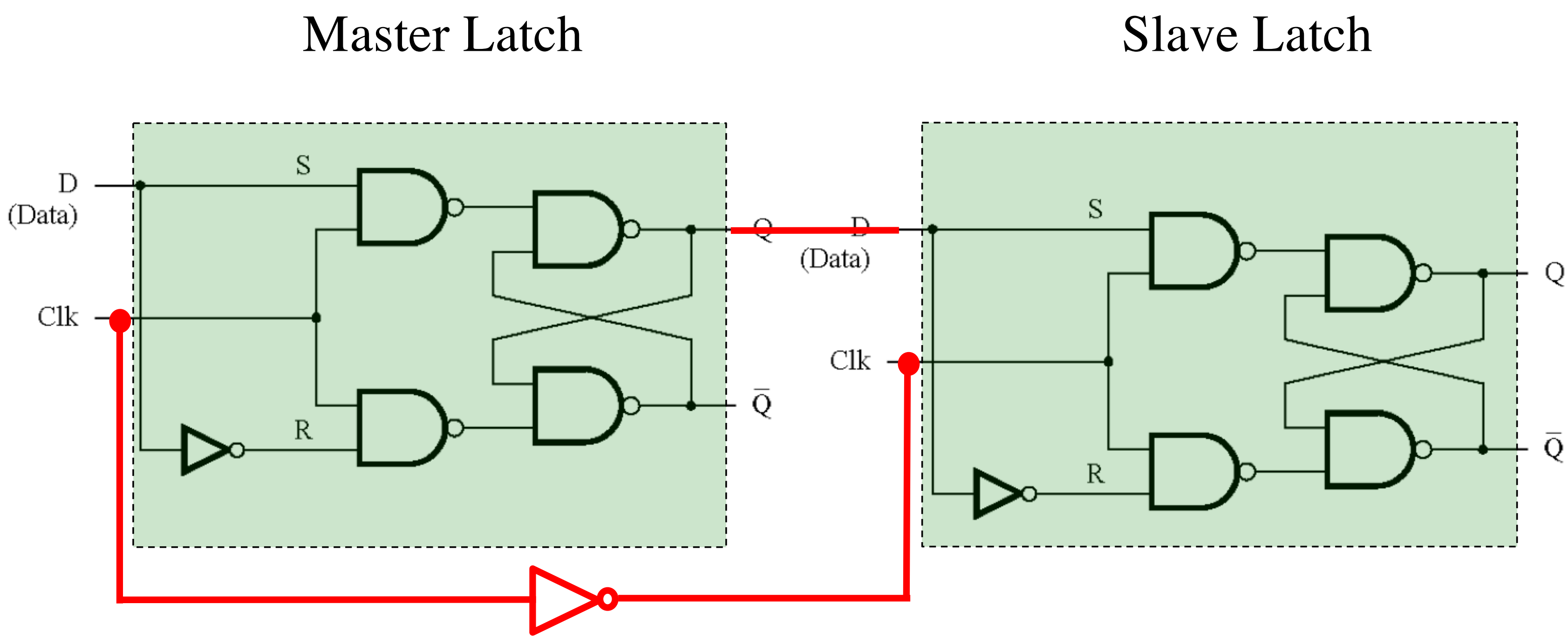
Slave Latch



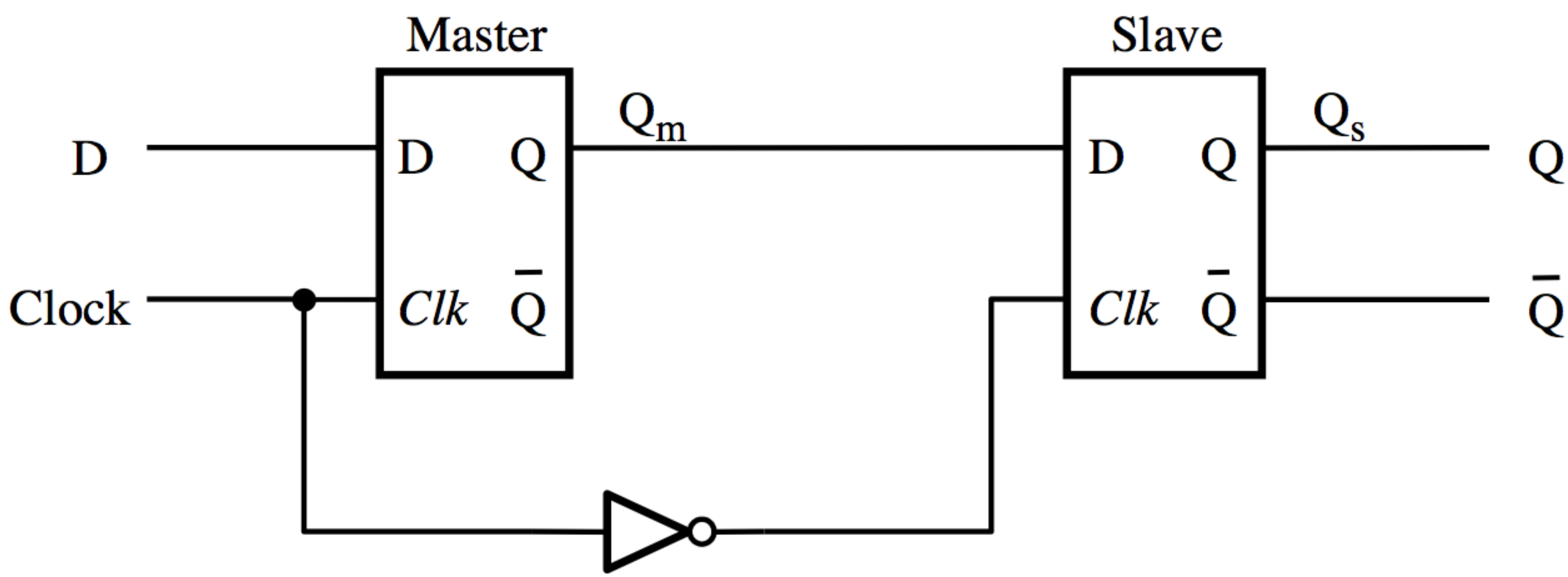
It takes two D latches to construct one D flip-flop



It takes two D latches to construct one D flip-flop



Constructing a Master-Slave D Flip-Flop From Two D Latches

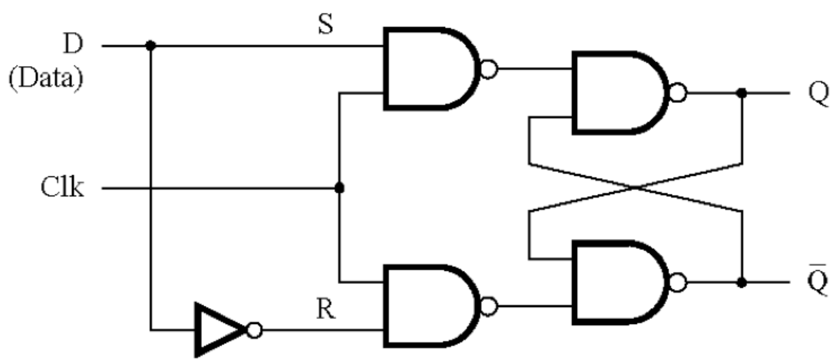


[Figure 5.9a from the textbook]

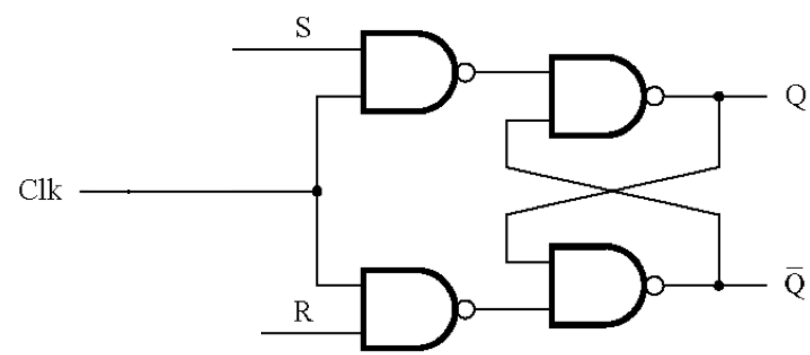
Constructing a Master-Slave D Flip-Flop From one D Latch and one Gated SR Latch

(This version uses one less NOT gate)

Master

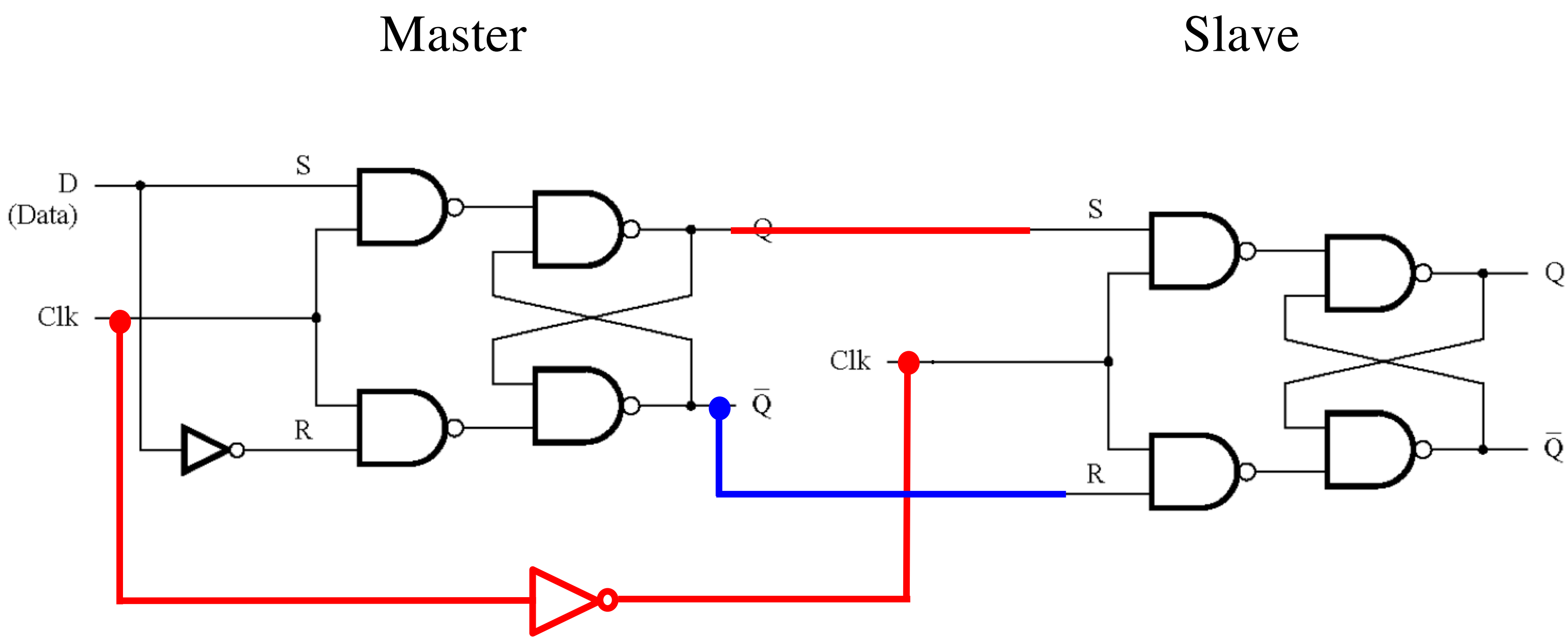


Slave



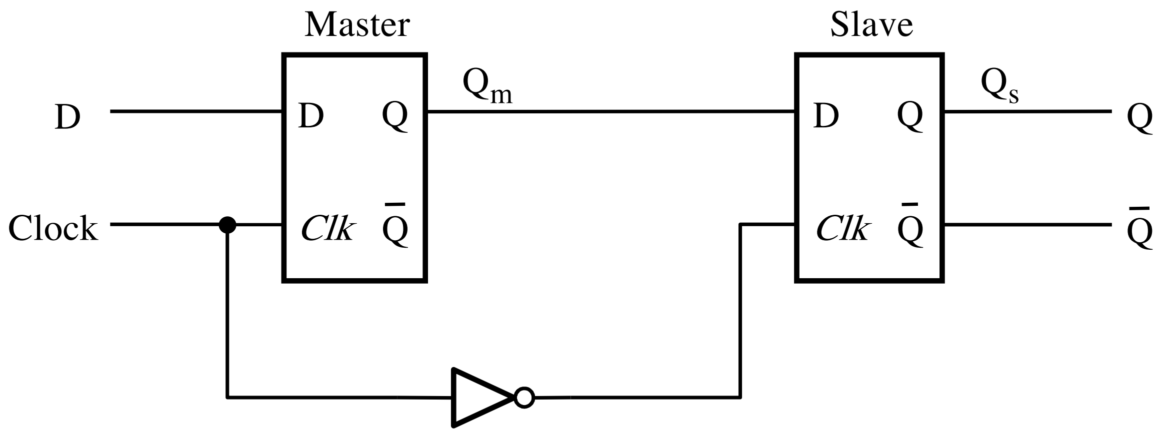
Constructing a Master-Slave D Flip-Flop From one D Latch and one Gated SR Latch

(This version uses one less NOT gate)

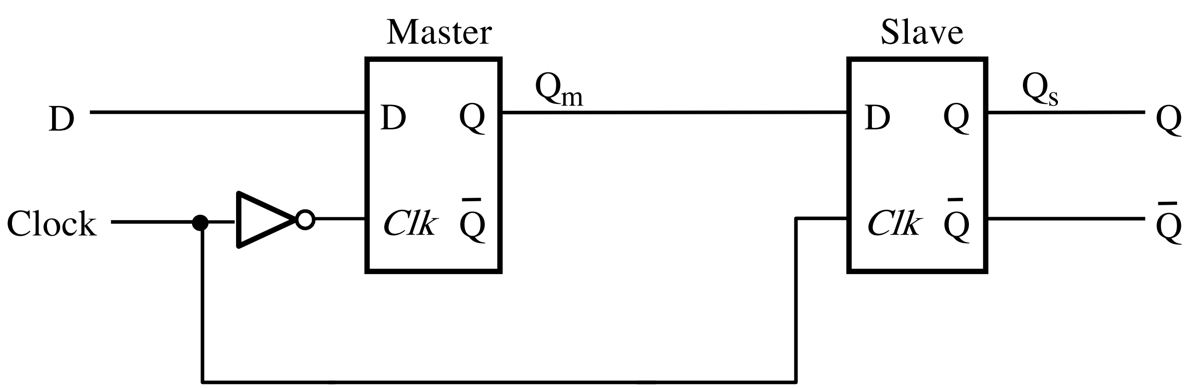


Positive-Edge vs. Negative-Edge D Flip-Flop

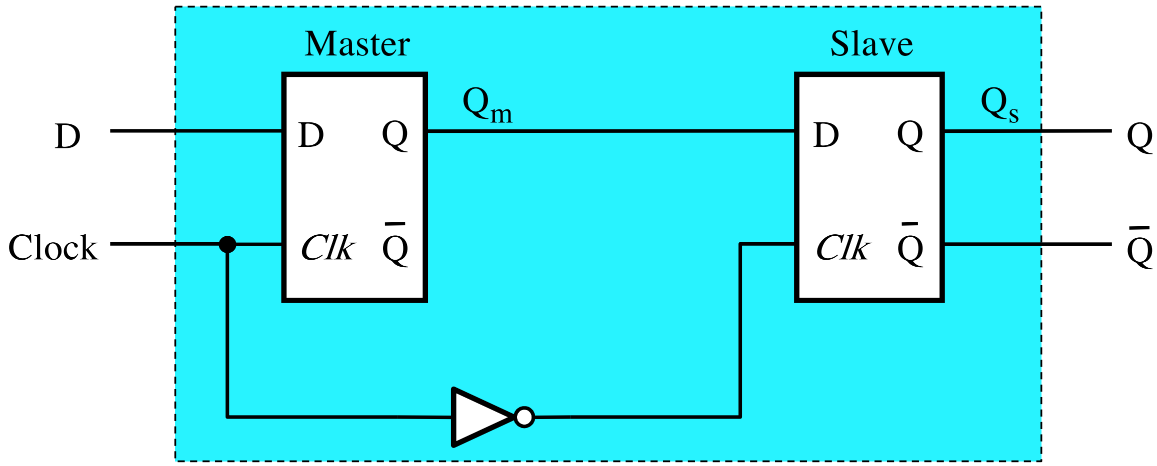
Negative-Edge-Triggered Master-Slave D Flip-Flop

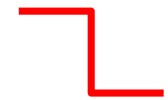


Positive-Edge-Triggered Master-Slave D Flip-Flop



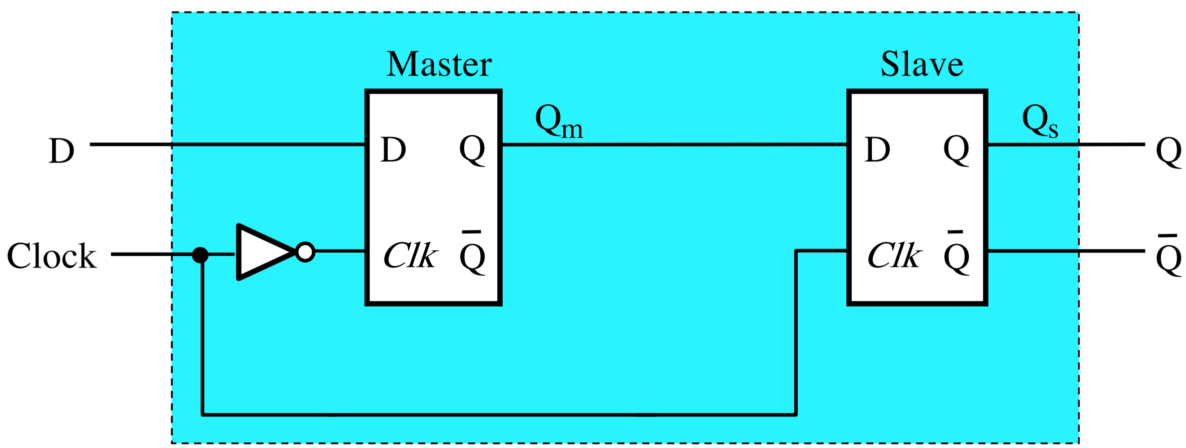
Negative-Edge-Triggered Master-Slave D Flip-Flop

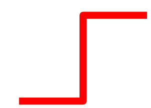




negative edge
(of the Clock)

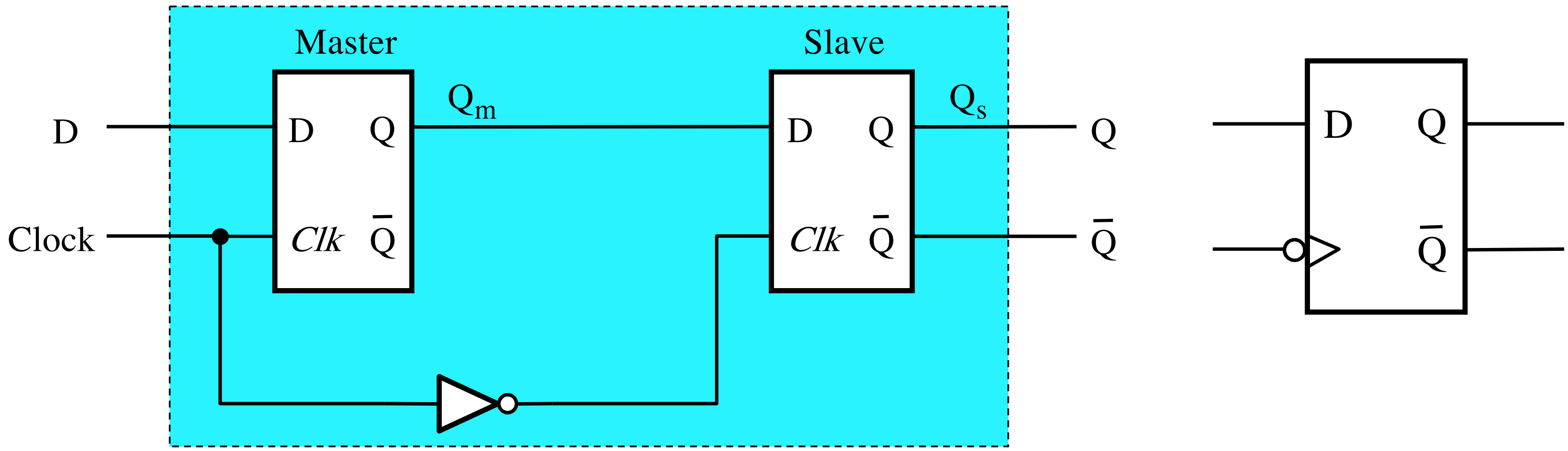
Positive-Edge-Triggered Master-Slave D Flip-Flop



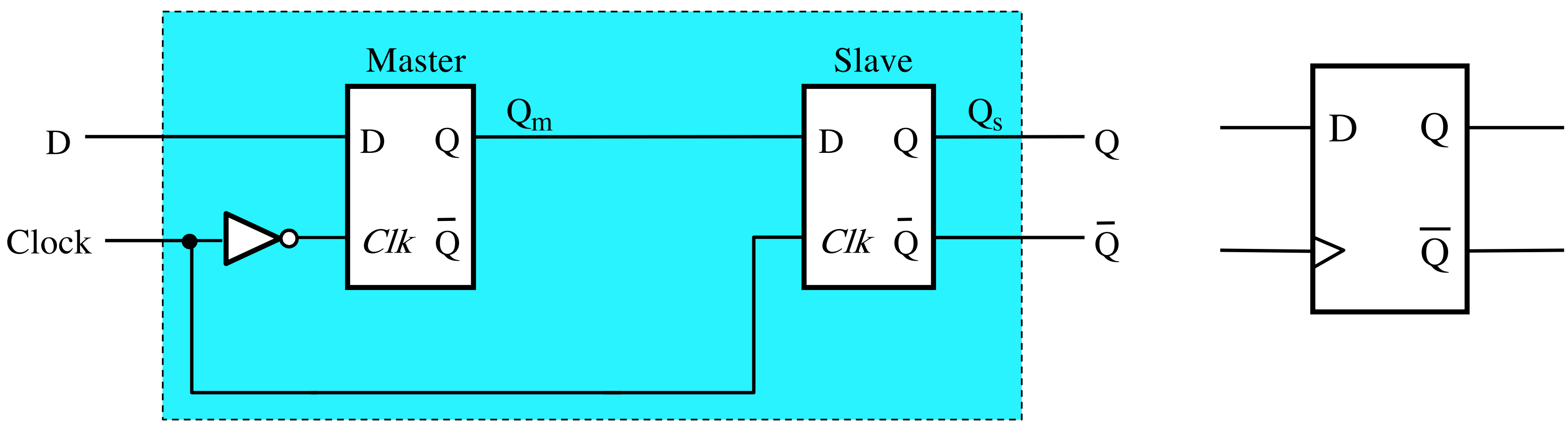


positive edge
(of the Clock)

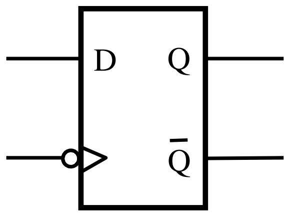
Negative-Edge-Triggered Master-Slave D Flip-Flop



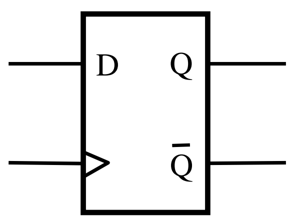
Positive-Edge-Triggered Master-Slave D Flip-Flop



Negative-Edge-Triggered Master-Slave D Flip-Flop

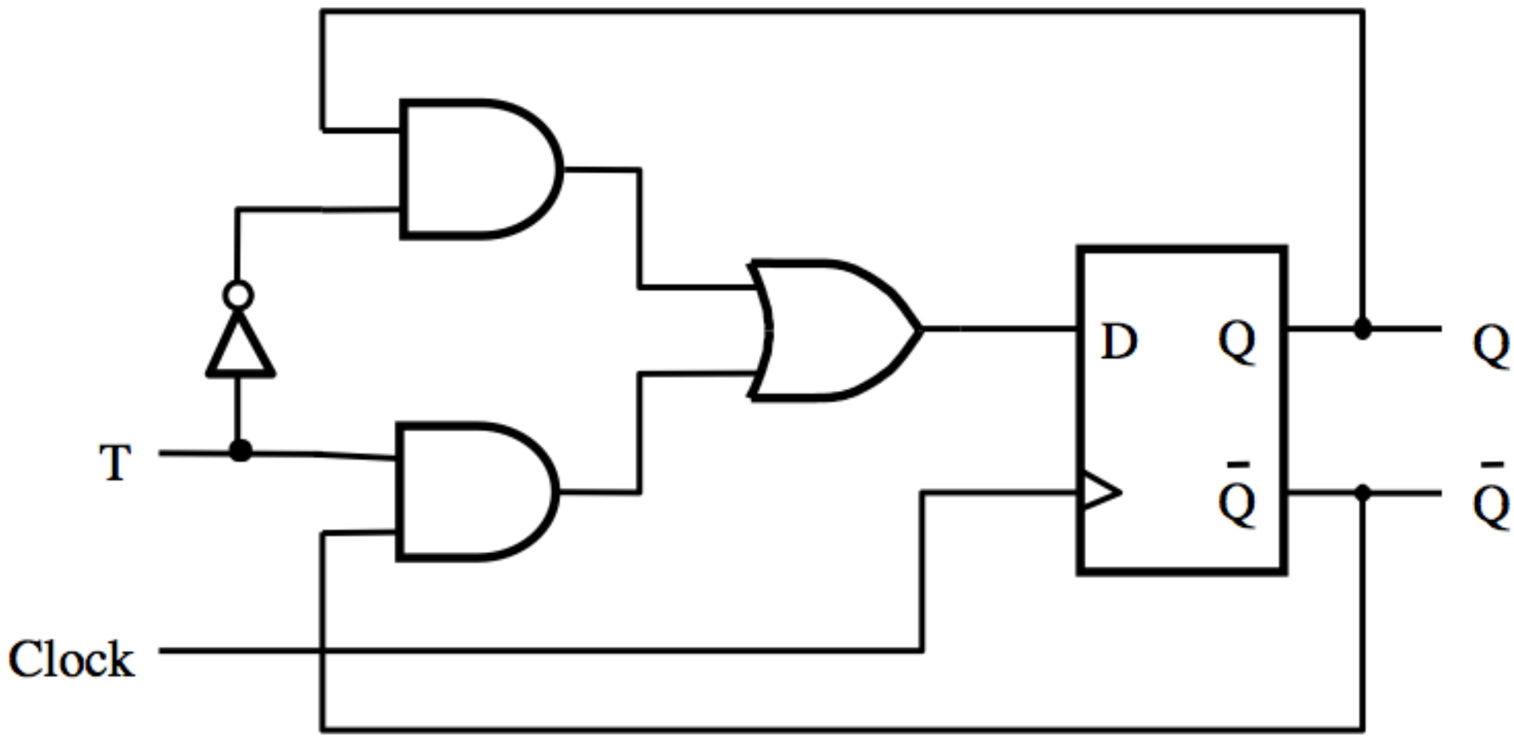


Positive-Edge-Triggered Master-Slave D Flip-Flop



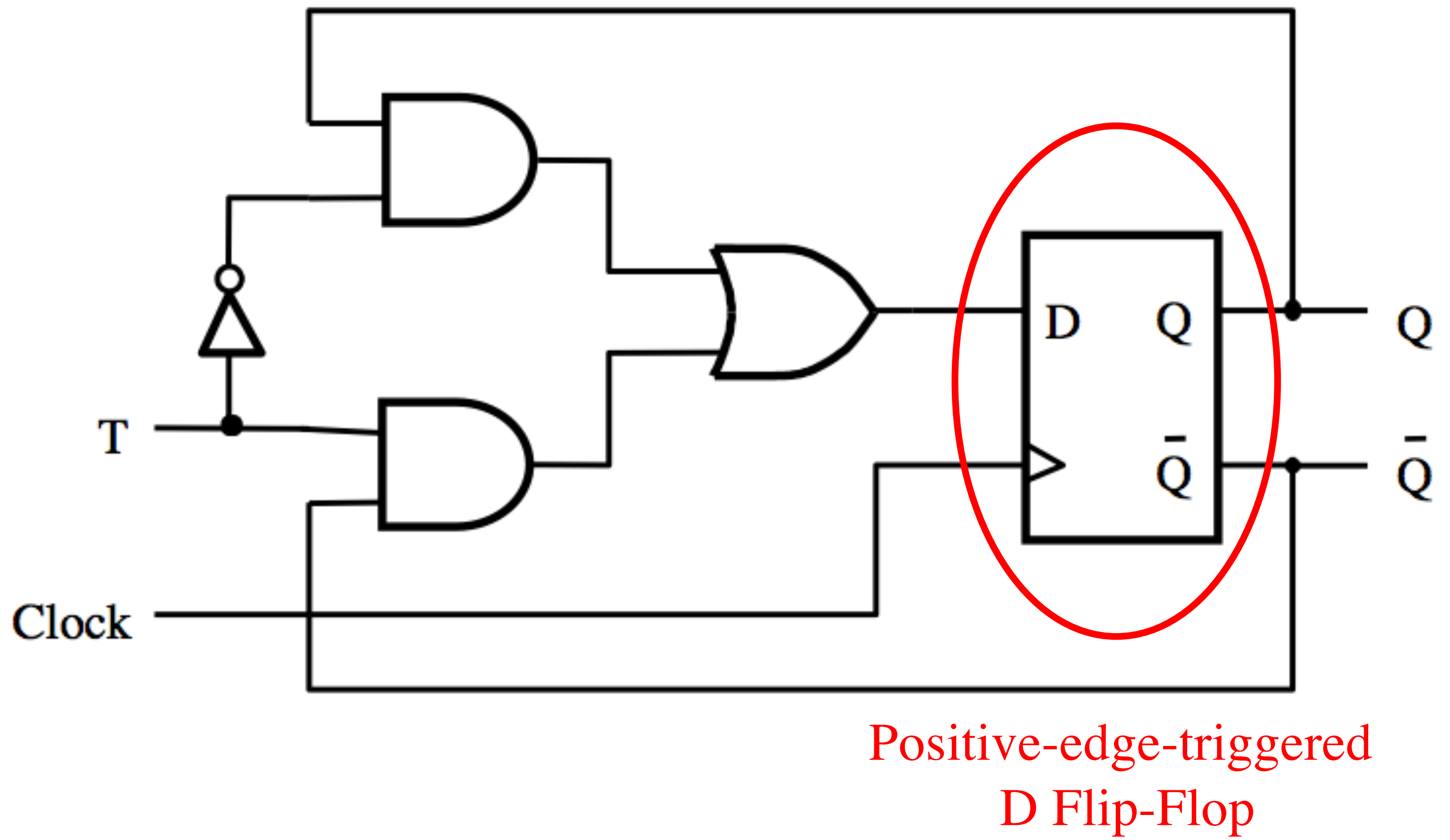
T Flip-Flop

T Flip-Flop



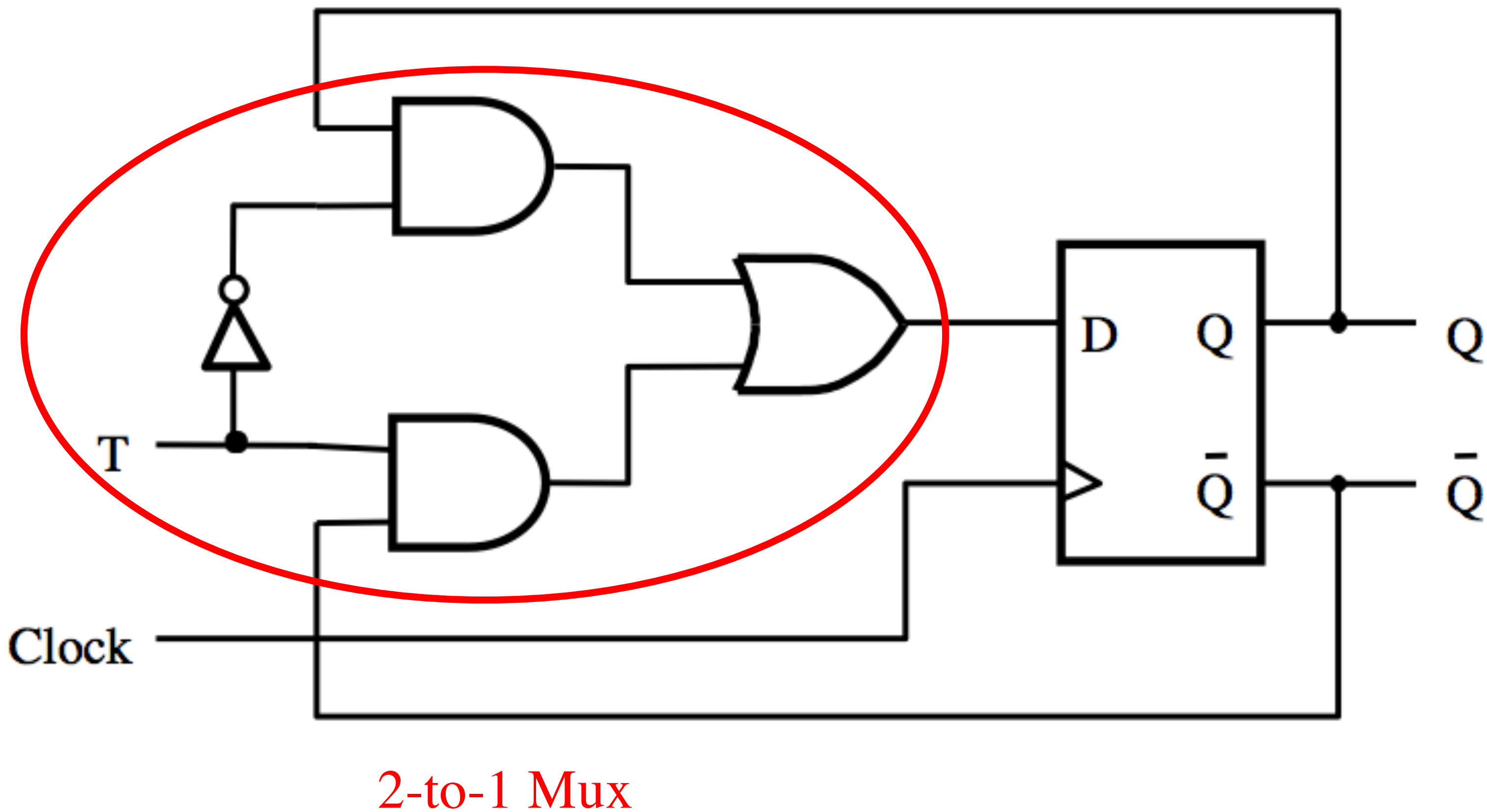
[Figure 5.15a from the textbook]

T Flip-Flop



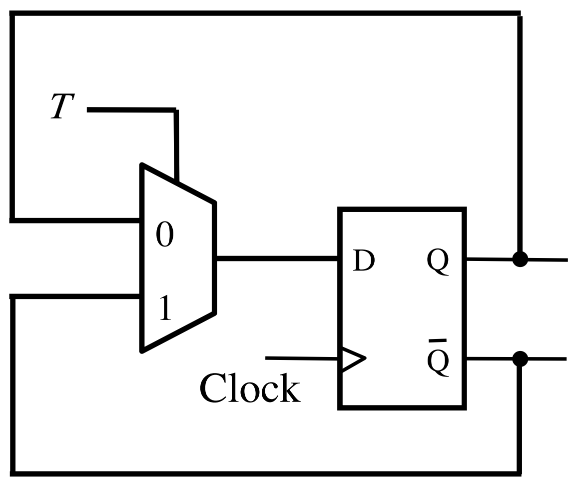
[Figure 5.15a from the textbook]

T Flip-Flop

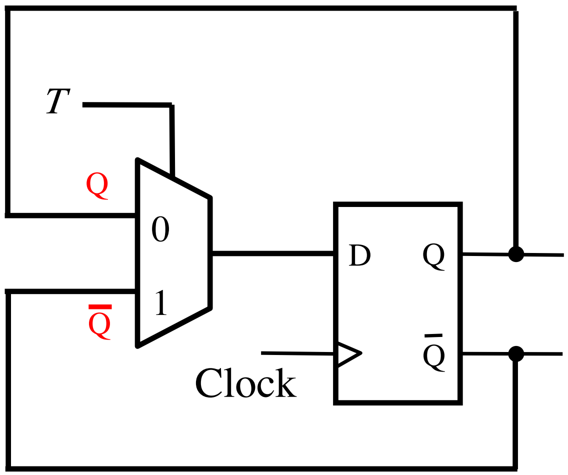


[Figure 5.15a from the textbook]

Another Way to Draw a T Flip-Flop

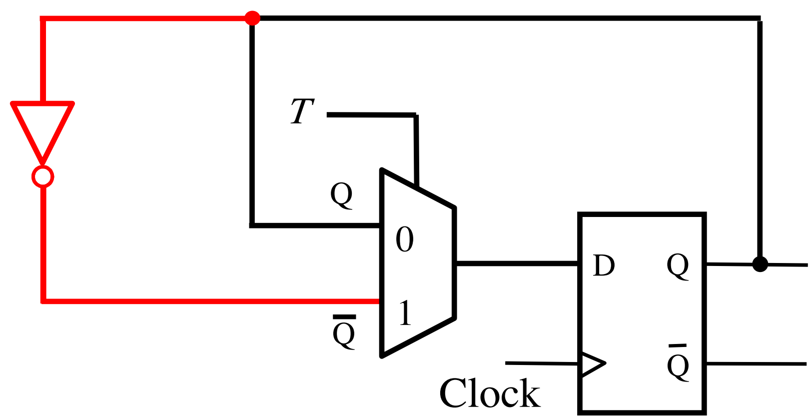


Another Way to Draw a T Flip-Flop

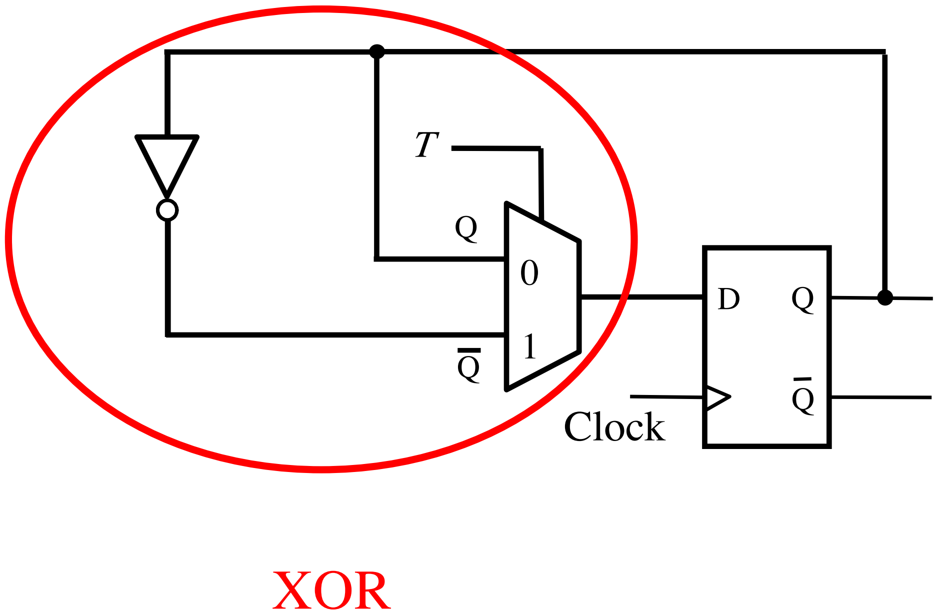


Note that the two inputs to the multiplexer are inverses of each other.

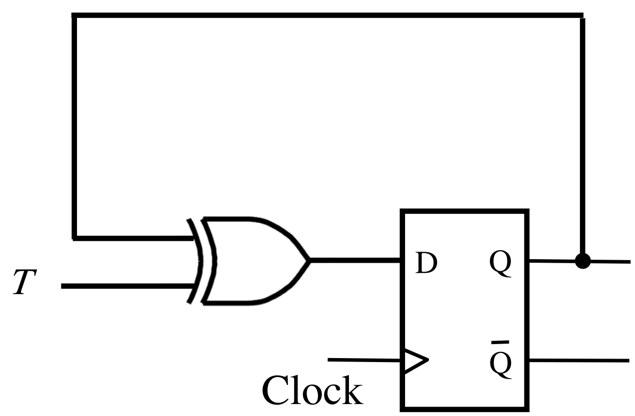
Another Way to Draw This



Another Way to Draw This



Yet Another Way to Draw a T Flip-Flop



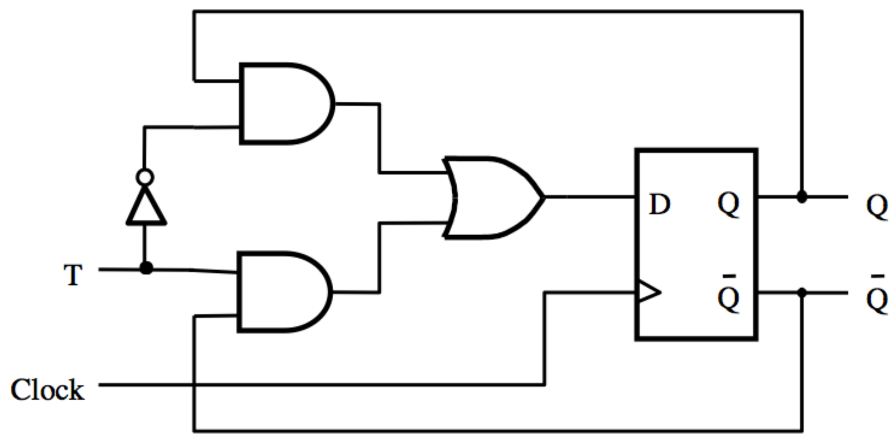
T Flip-Flop
(how it works)

If T=0 then it stays in its current state

If T=1 then it reverses its current state

In other words the circuit “toggles” its state when T=1. This is why it is called T flip-flop.

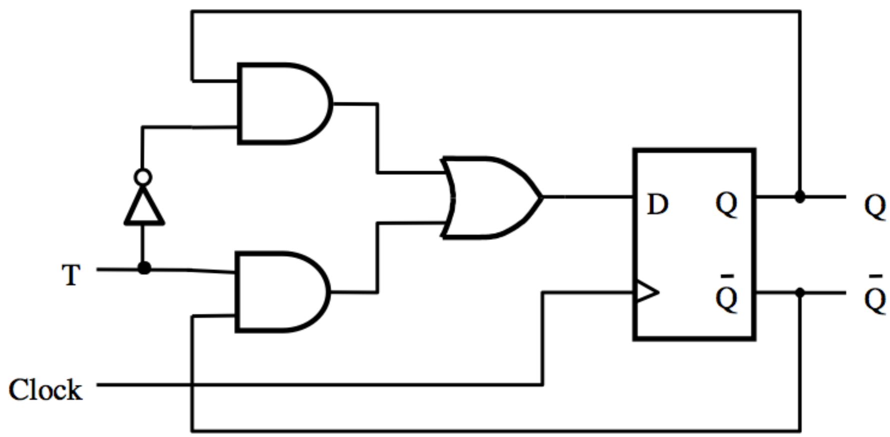
T Flip-Flop
(circuit and truth table)

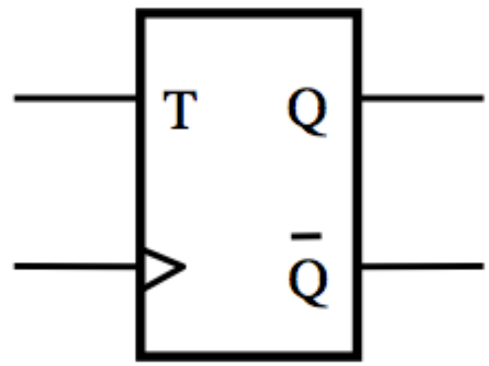


T	$Q(t+1)$
0	$Q(t)$
1	$\overline{Q}(t)$

[Figure 5.15a,b from the textbook]

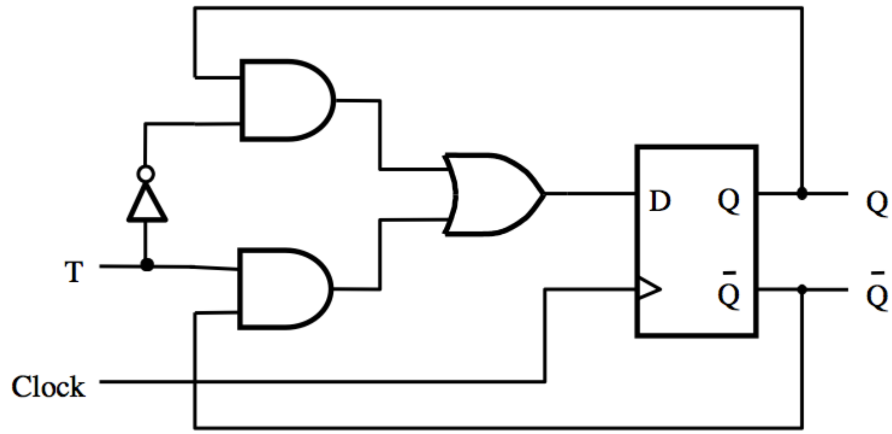
T Flip-Flop
(circuit and graphical symbol)

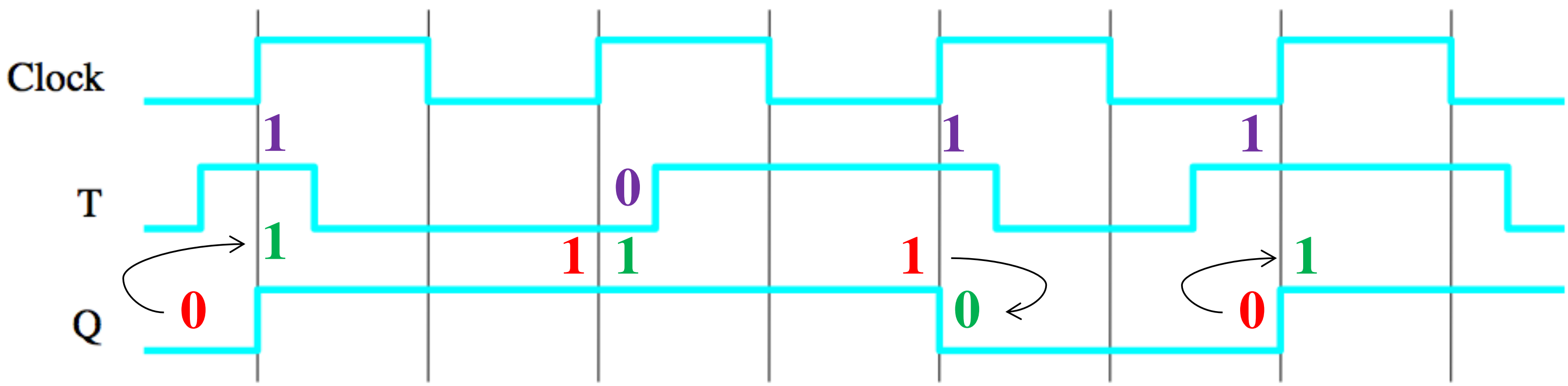




[Figure 5.15a,c from the textbook]

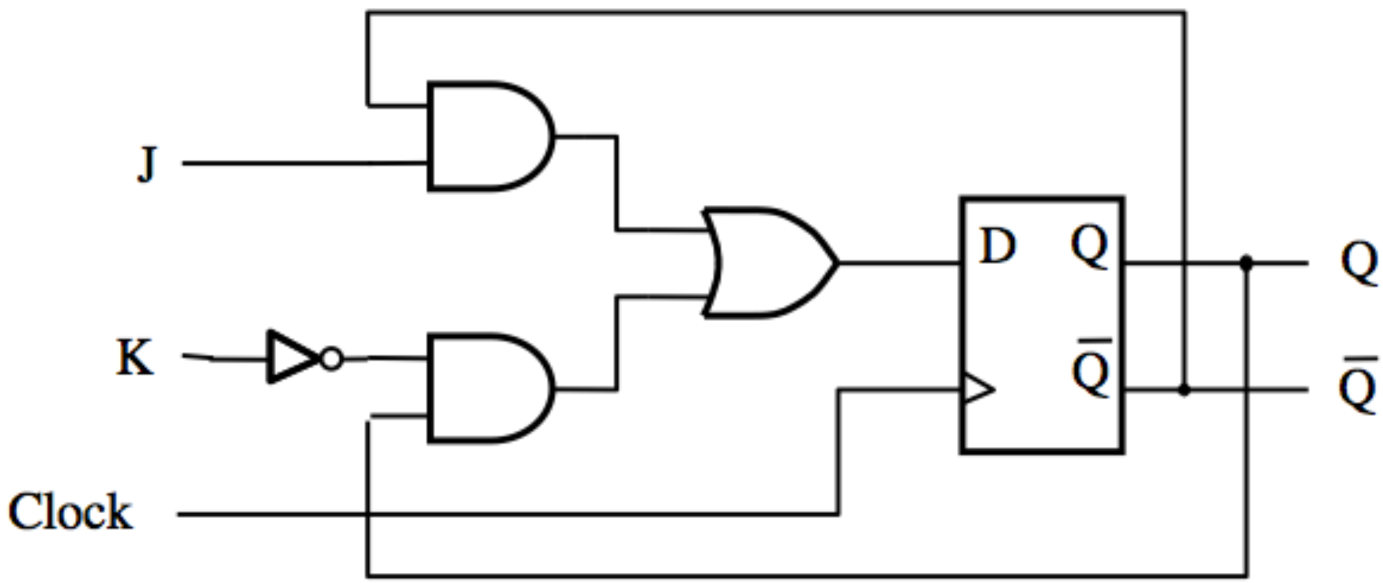
T Flip-Flop (Timing Diagram)





JK Flip-Flop

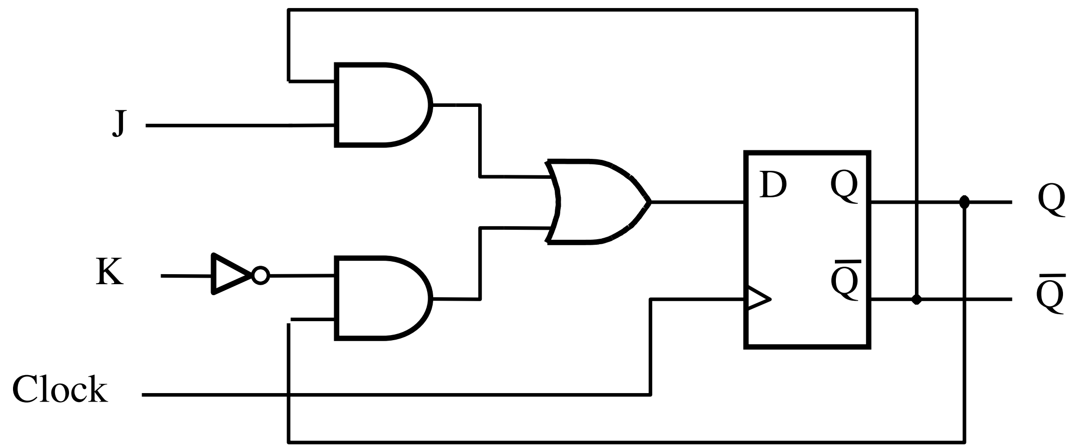
JK Flip-Flop



$$D = J\overline{Q} + \overline{K}Q$$

[Figure 5.16a from the textbook]

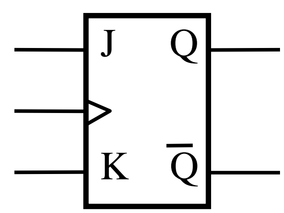
JK Flip-Flop



(a) Circuit

J	K	Q(t+1)	
0	0	Q(t)	Hold
0	1	0	Reset
1	0	1	Set
1	1	$\overline{Q}(t)$	Toggle

(b) Truth table



(c) Graphical symbol

[Figure 5.16 from the textbook]

JK Flip-Flop
(how it works)

A more versatile flip-flop

If J=0 and K=0 it stays in the same state

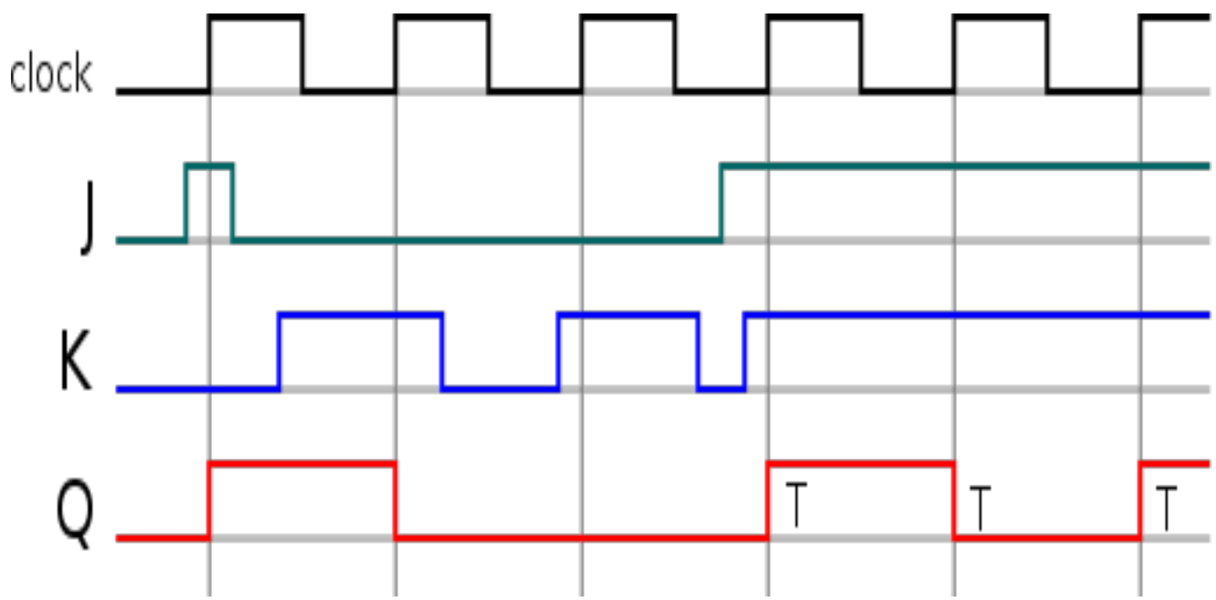
If J=1 and K=0 it sets the output Q to 1

If J=0 and K=1 it resets the output Q to 0

If J=1 and K=1 it toggles the output Q

If J=K then it behaves like a T flip-flop

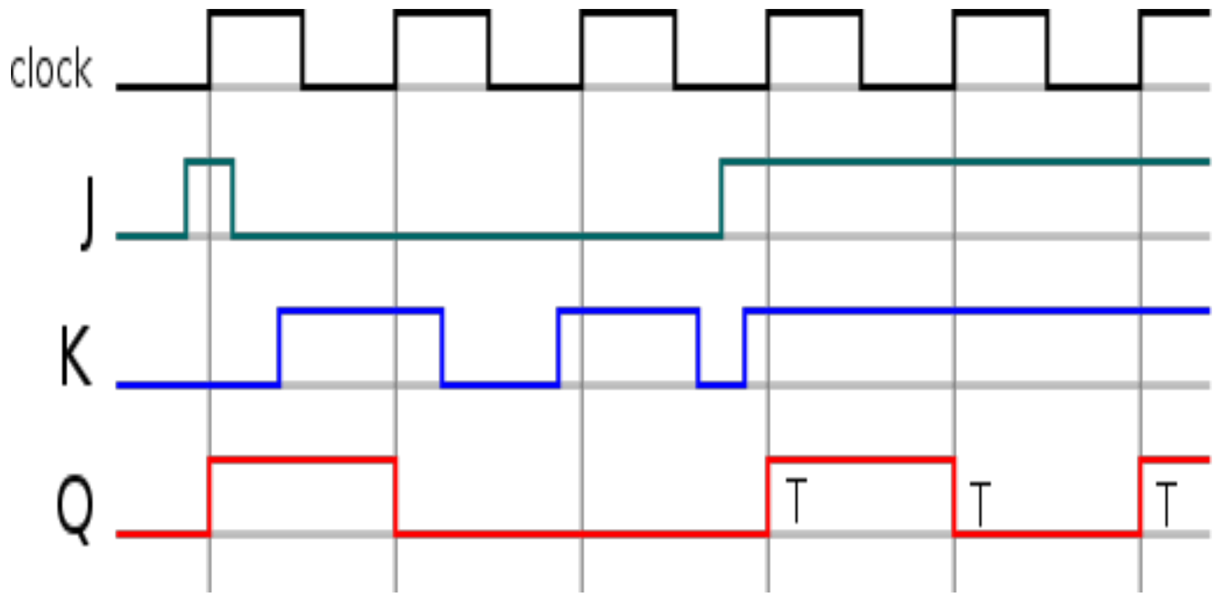
JK Flip-Flop
(timing diagram)



J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\overline{Q}$(t)

[https://en.wikipedia.org/wiki/Flip-flop_(electronics)]

JK Flip-Flop
(timing diagram)



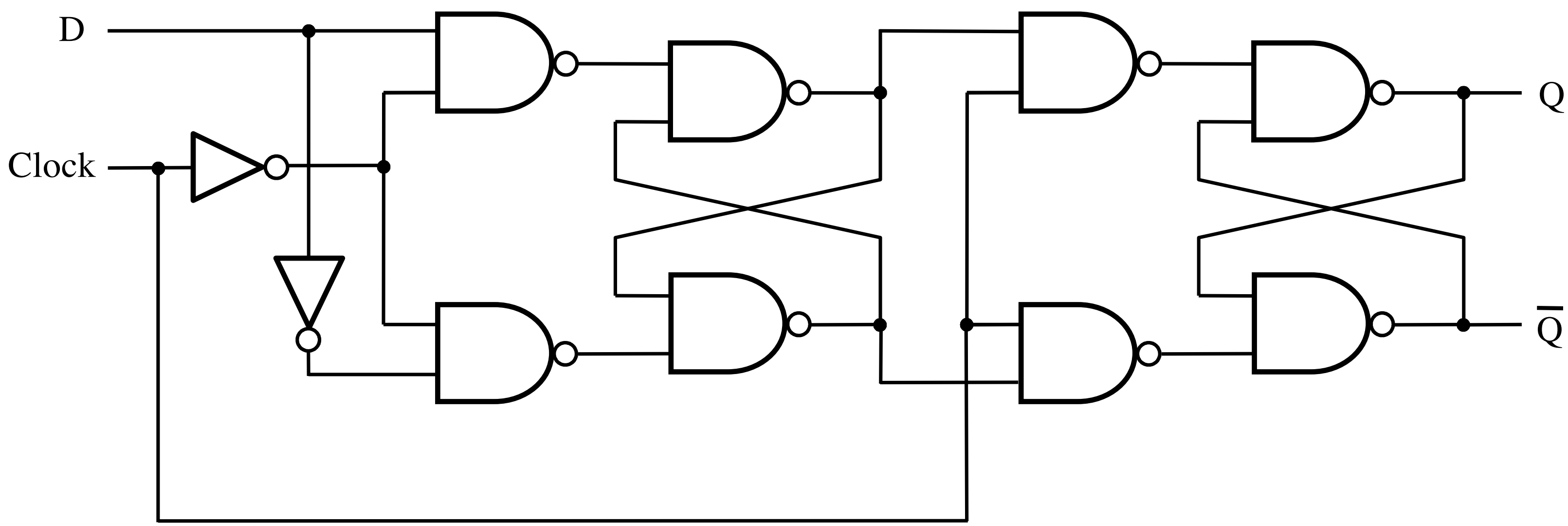
set reset

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

[https://en.wikipedia.org/wiki/Flip-flop_(electronics)]

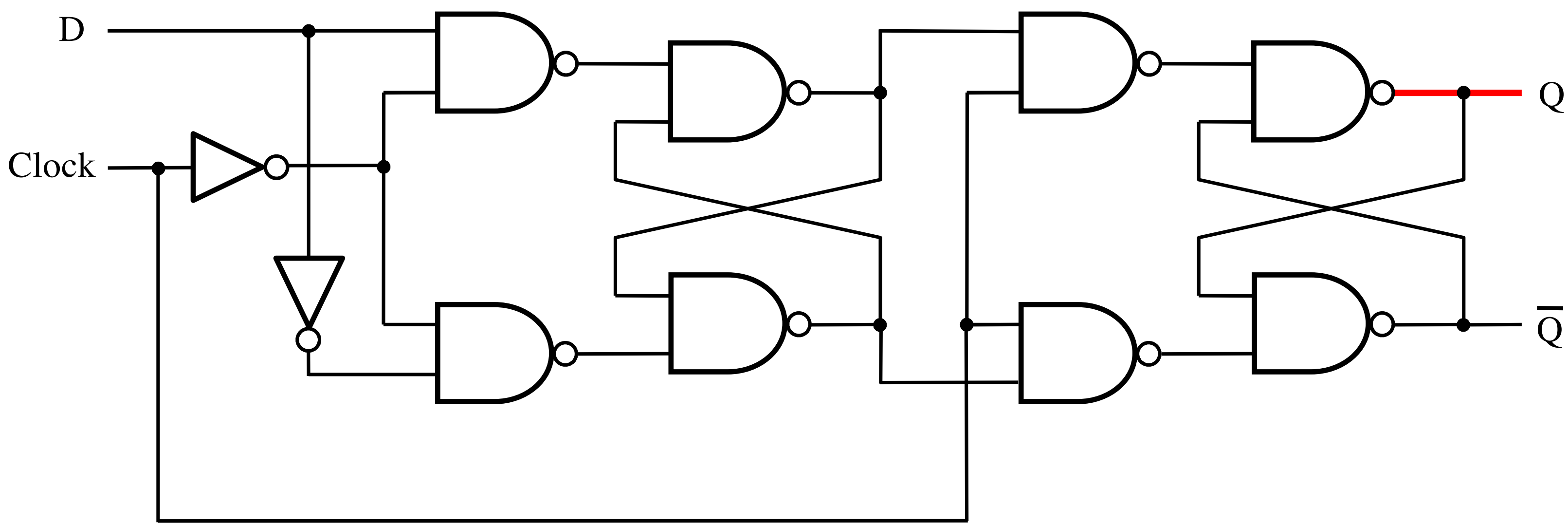
Complete Wiring Diagrams

The Complete Wiring Diagram for a Positive-Edge-Triggered D Flip-Flop

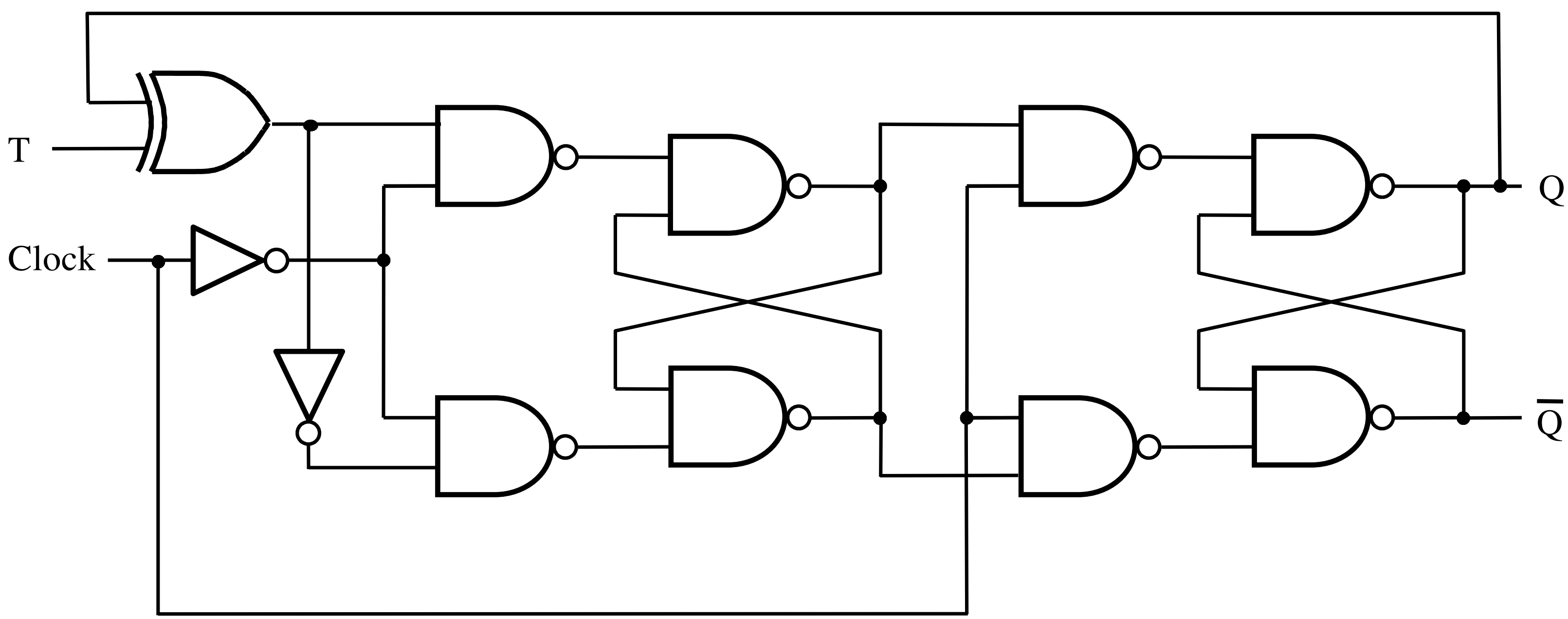


The Complete Wiring Diagram for a Positive-Edge-Triggered D Flip-Flop

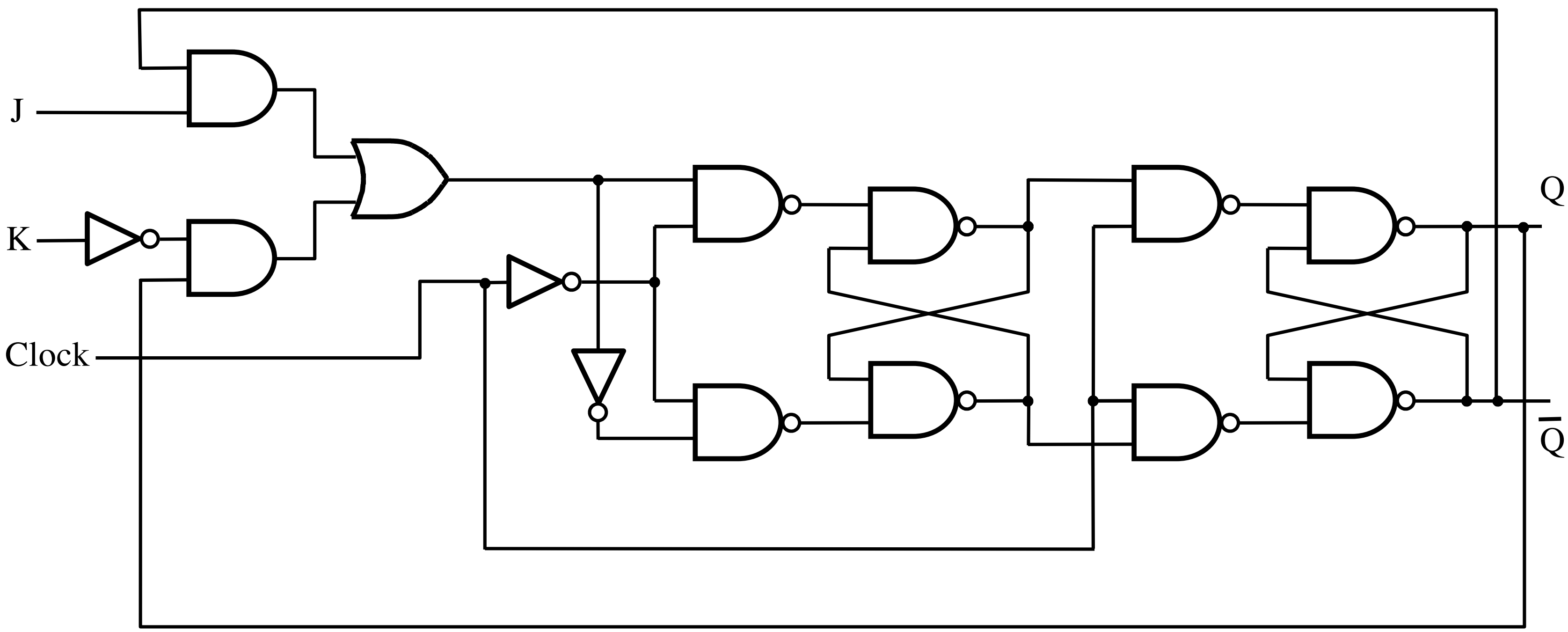
We need all of this just to store 1 bit!



The Complete Wiring Diagram for a Positive-Edge-Triggered T Flip-Flop

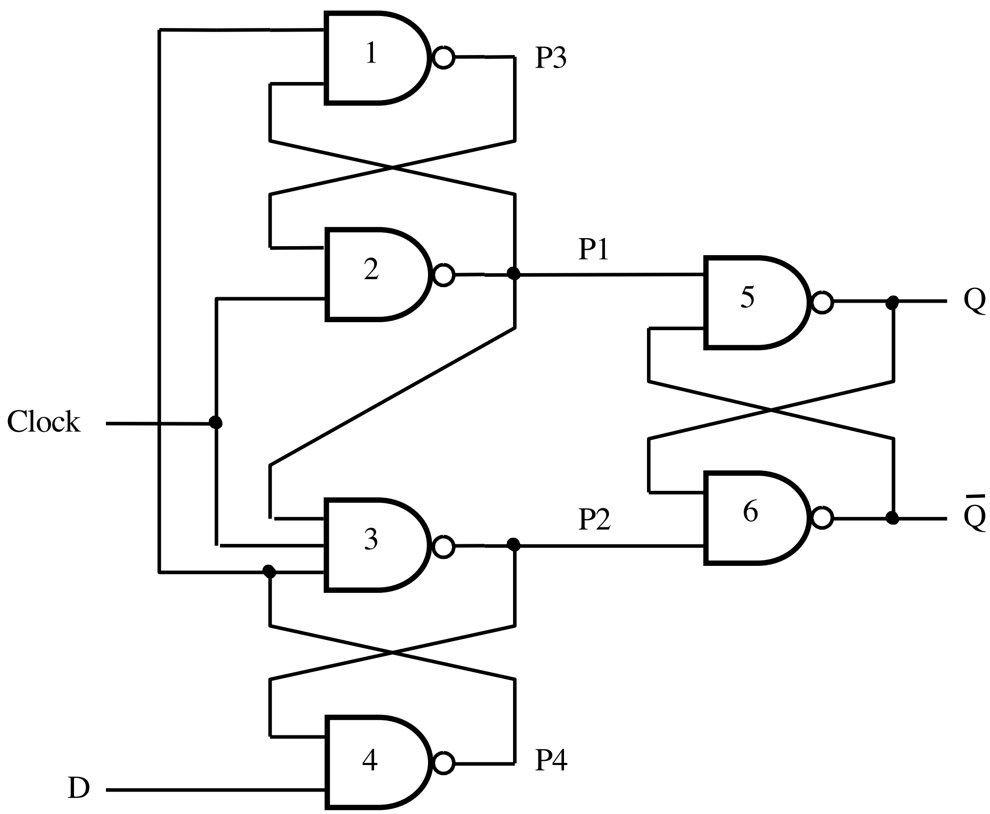


The Complete Wiring Diagram for a Positive-Edge-Triggered JK Flip-Flop

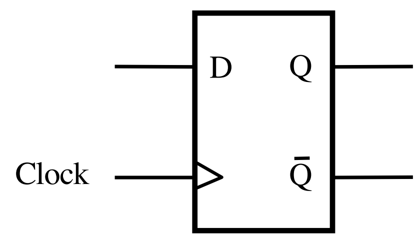


An alternative D Flip-Flop Design

A positive-edge-triggered D flip-flop



(a) Circuit

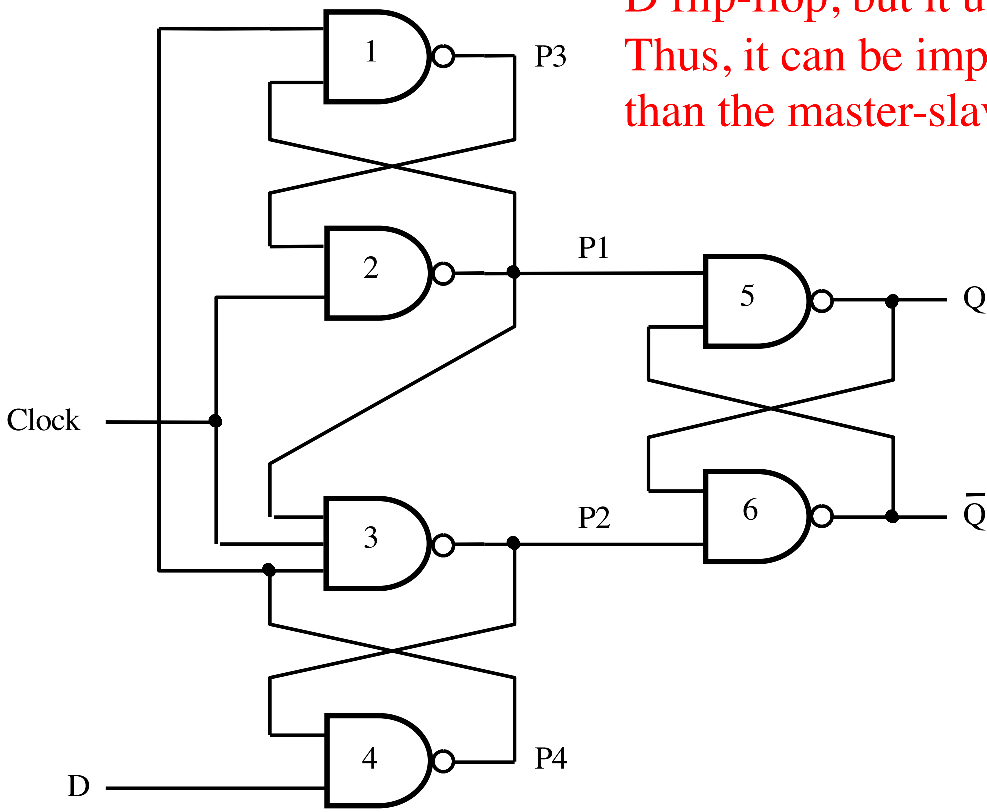


(b) Graphical symbol

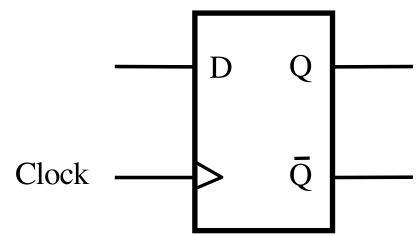
[Figure 5.11 from the textbook]

A positive-edge-triggered D flip-flop

This circuit behaves like a positive-edge-triggered D flip-flop, but it uses only 6 NAND gates.
Thus, it can be implemented with fewer transistors than the master-slave D flip-flop.



(a) Circuit



(b) Graphical symbol

[Figure 5.11 from the textbook]

Registers

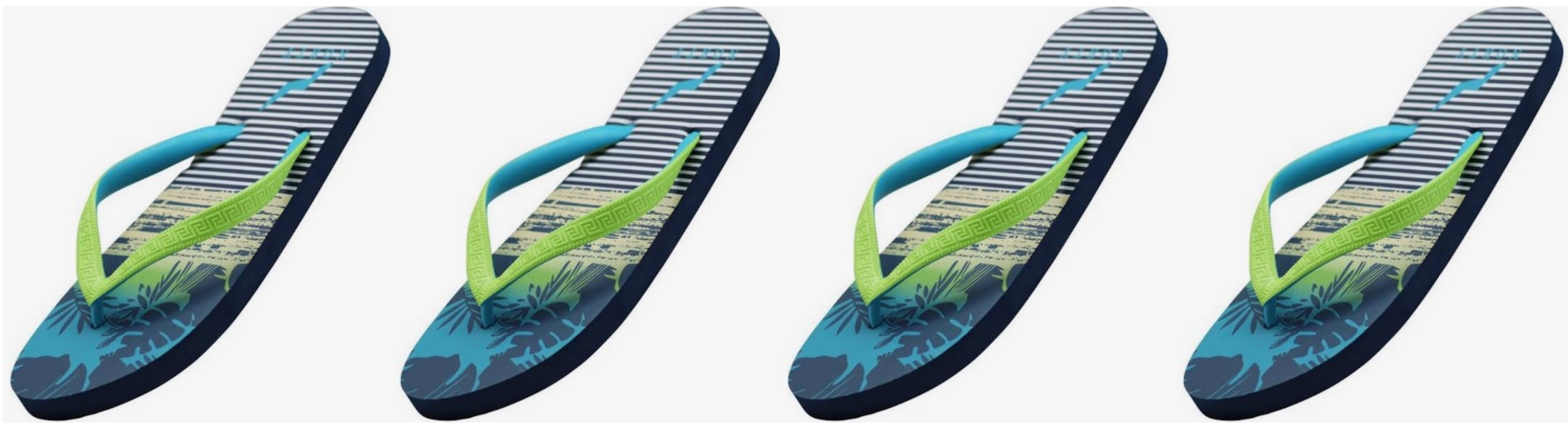
Register
(Definition)

An n-bit structure consisting of flip-flops.

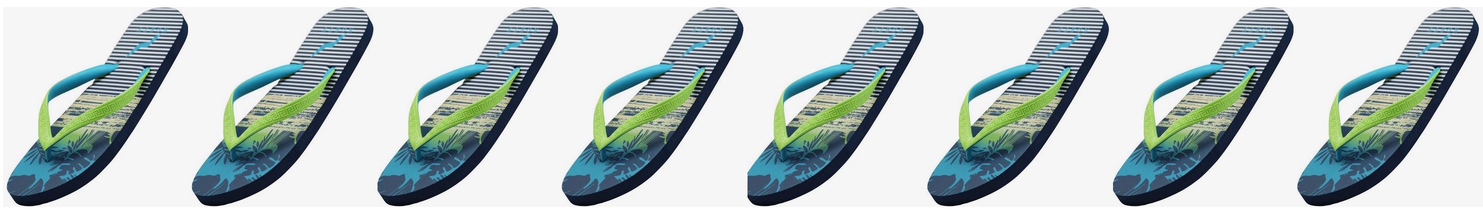
Flip-Flop

[https://www.ebay.com/itm/223956811053?chn=ps&mkevt=1&mkcid=28&var=522710005842]

4-bit Register

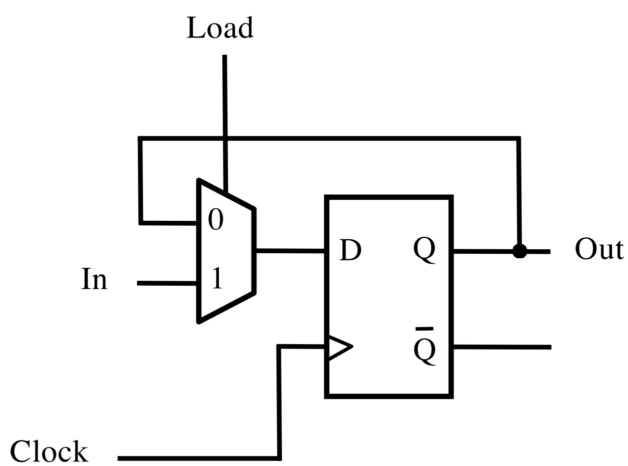


8-bit Register

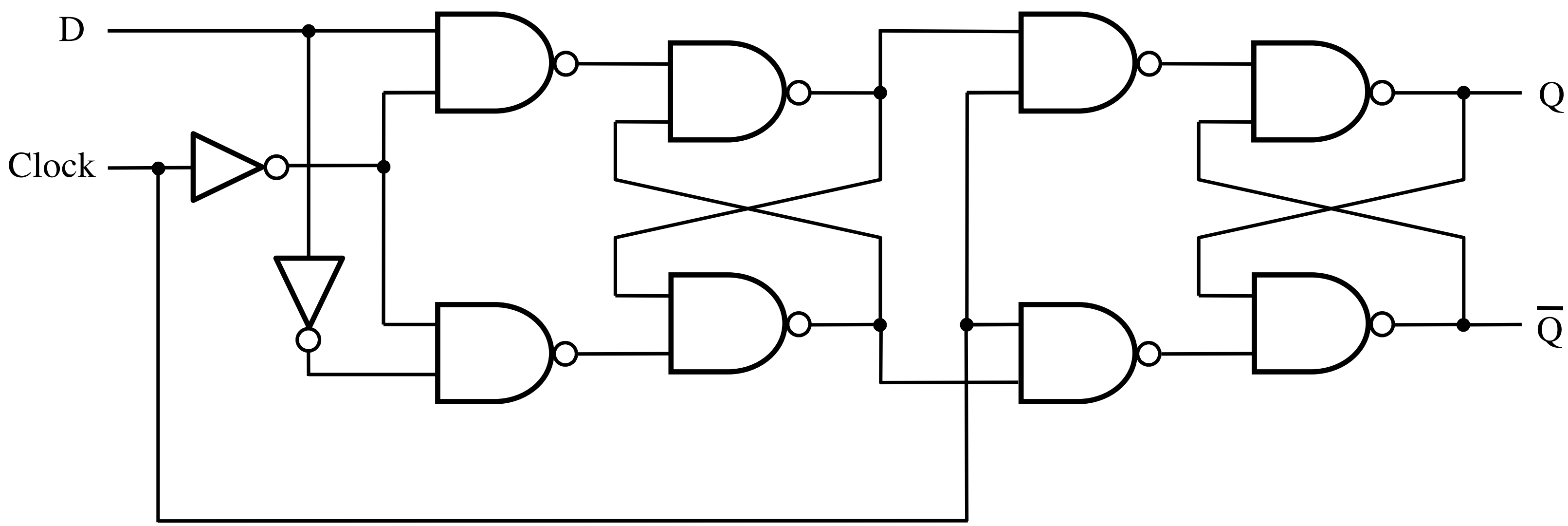


Parallel-Access Register

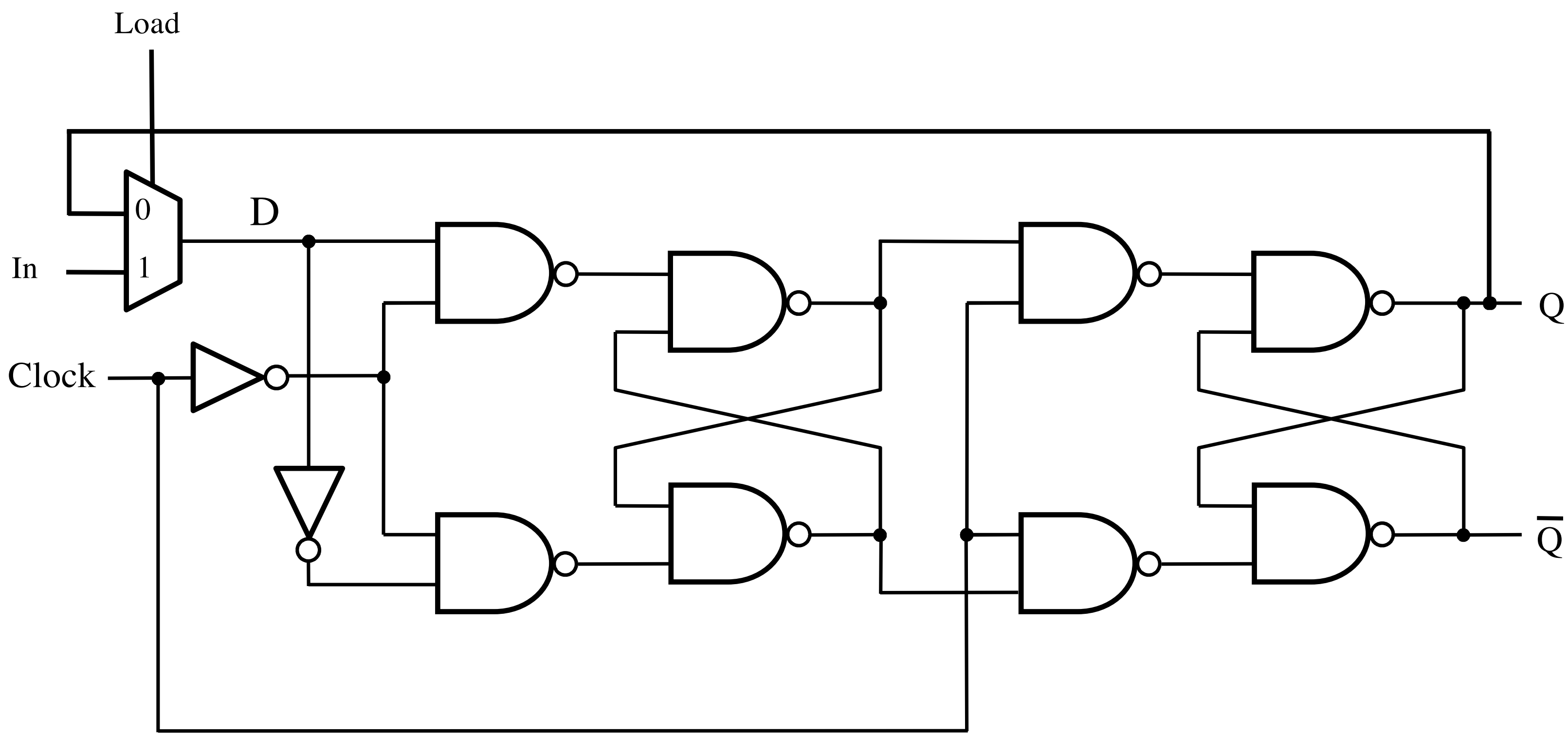
1-Bit Parallel-Access Register



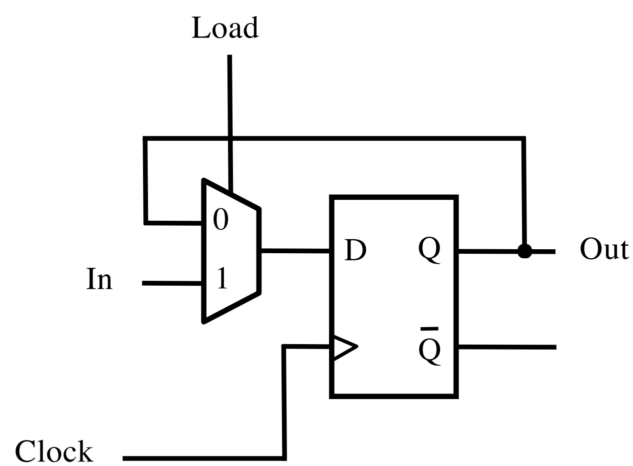
Positive-Edge-Triggered D Flip-Flop



1-bit Parallel-Access Register



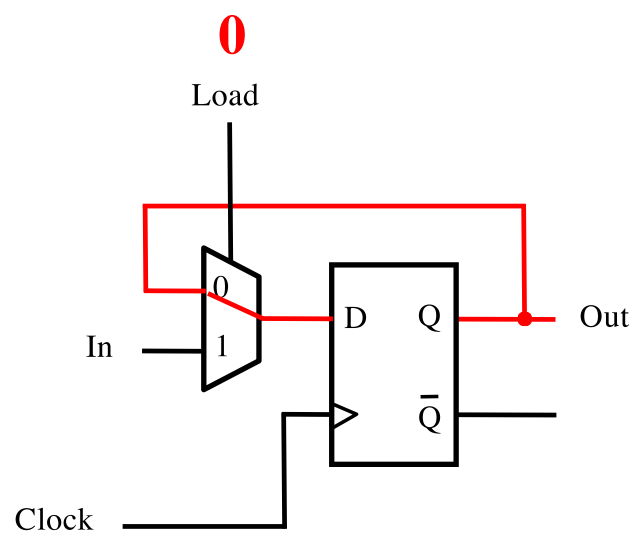
1-Bit Parallel-Access Register



The 2-to-1 multiplexer is used to select whether to load a new value into the D flip-flop or to retain the old value.

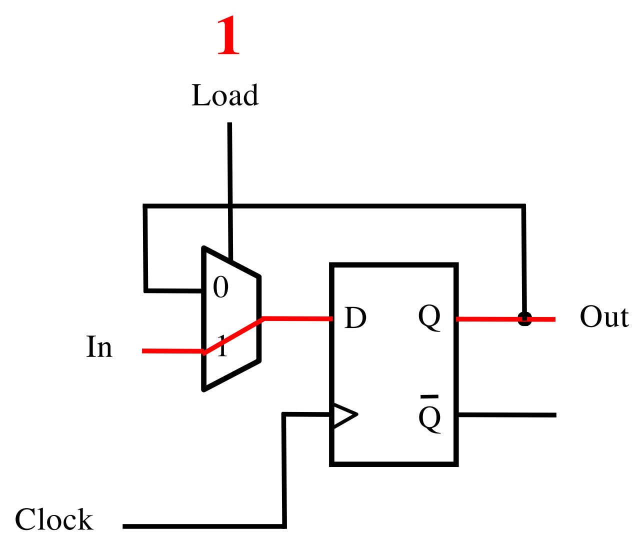
The output of this circuit is the Q output of the flip-flop.

1-Bit Parallel-Access Register



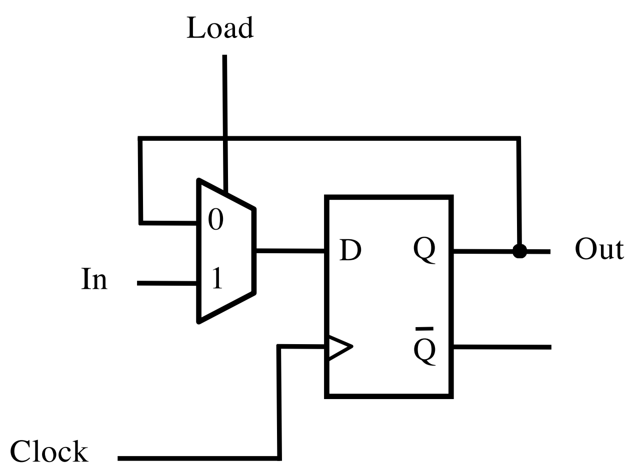
If Load = 0,   then retain the old value.

1-Bit Parallel-Access Register



If Load = 1,   then load the new value from In.

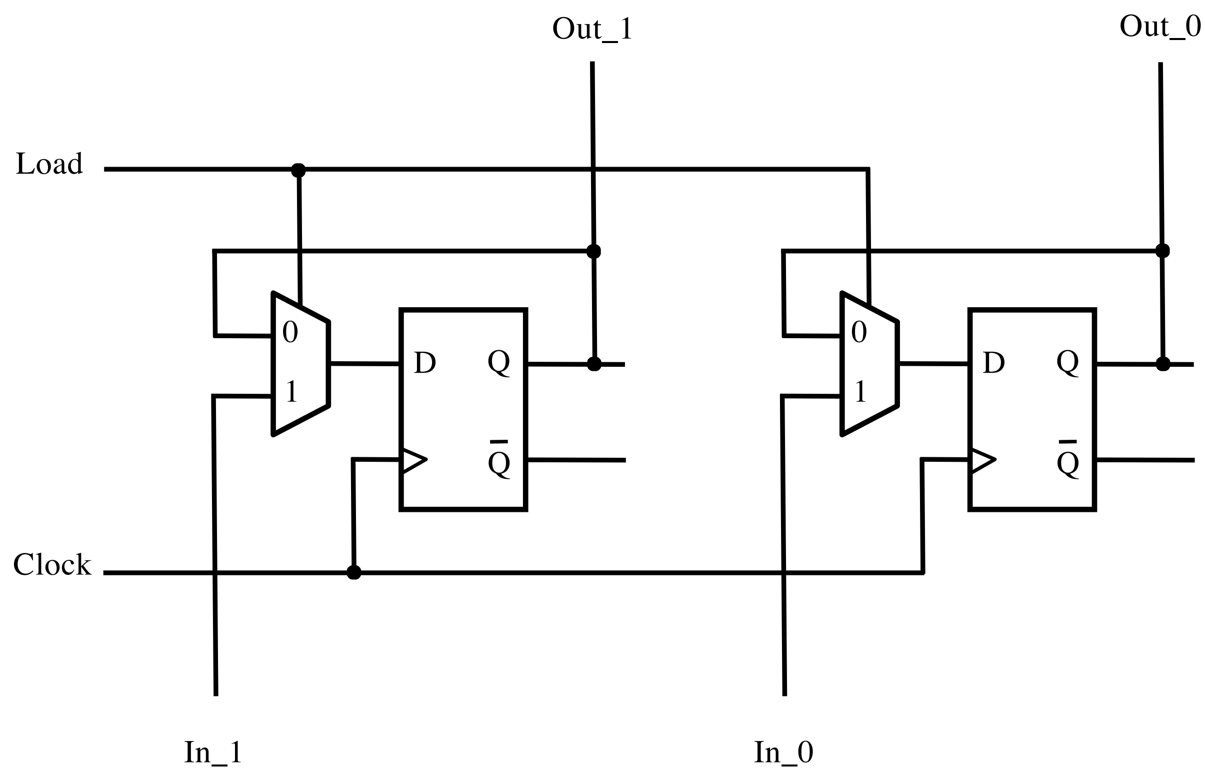
1-Bit Parallel-Access Register



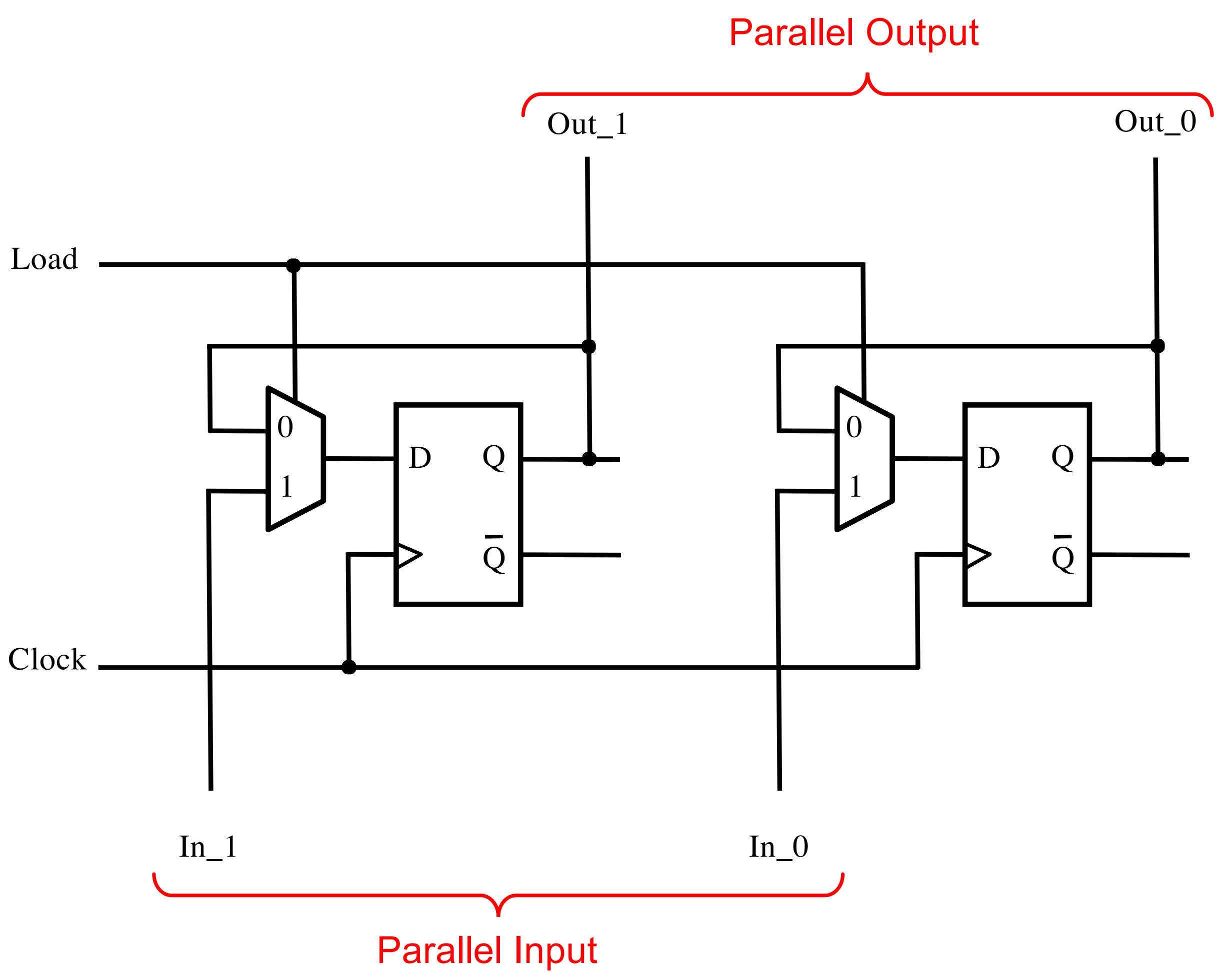
If Load = 0,   then retain the old value.

If Load = 1,   then load the new value from In.

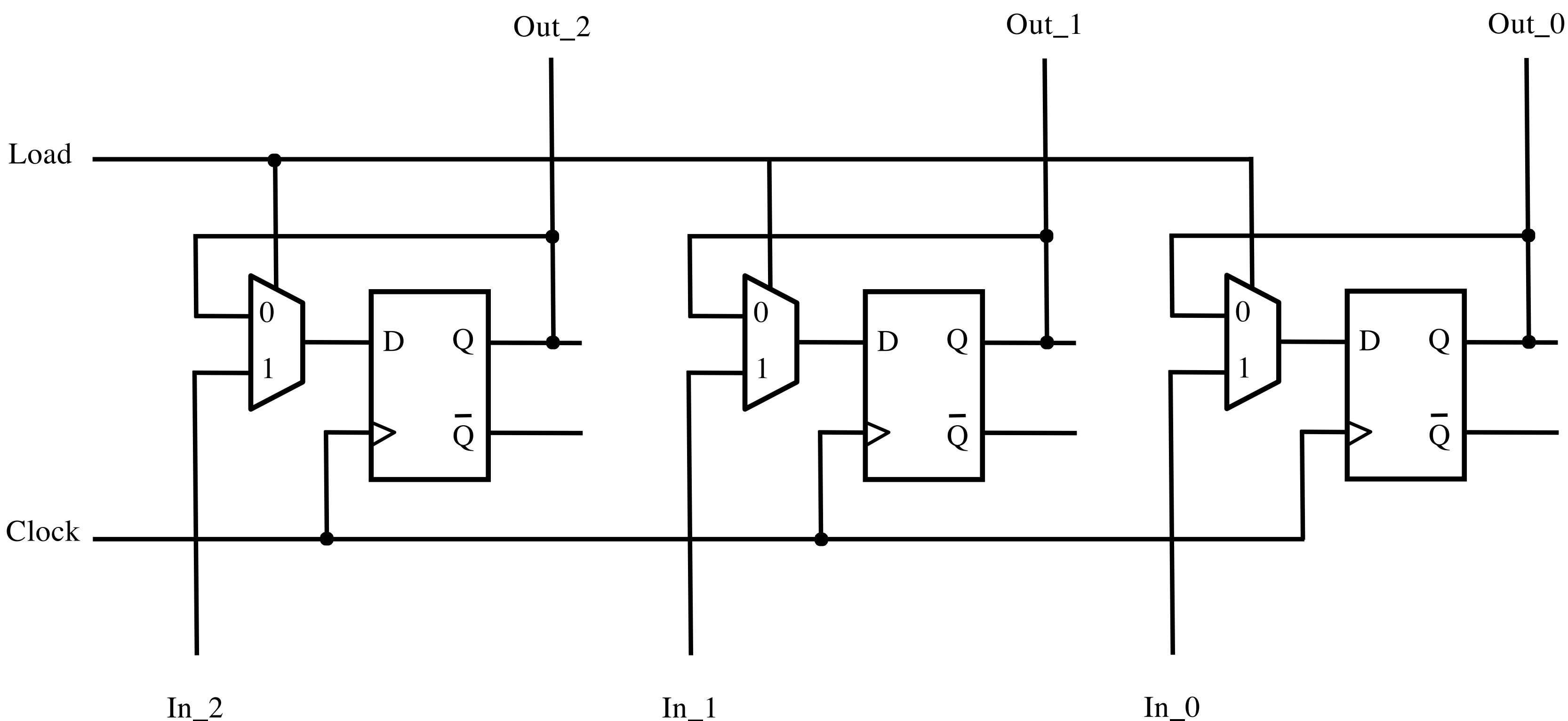
2-Bit Parallel-Access Register



2-Bit Parallel-Access Register

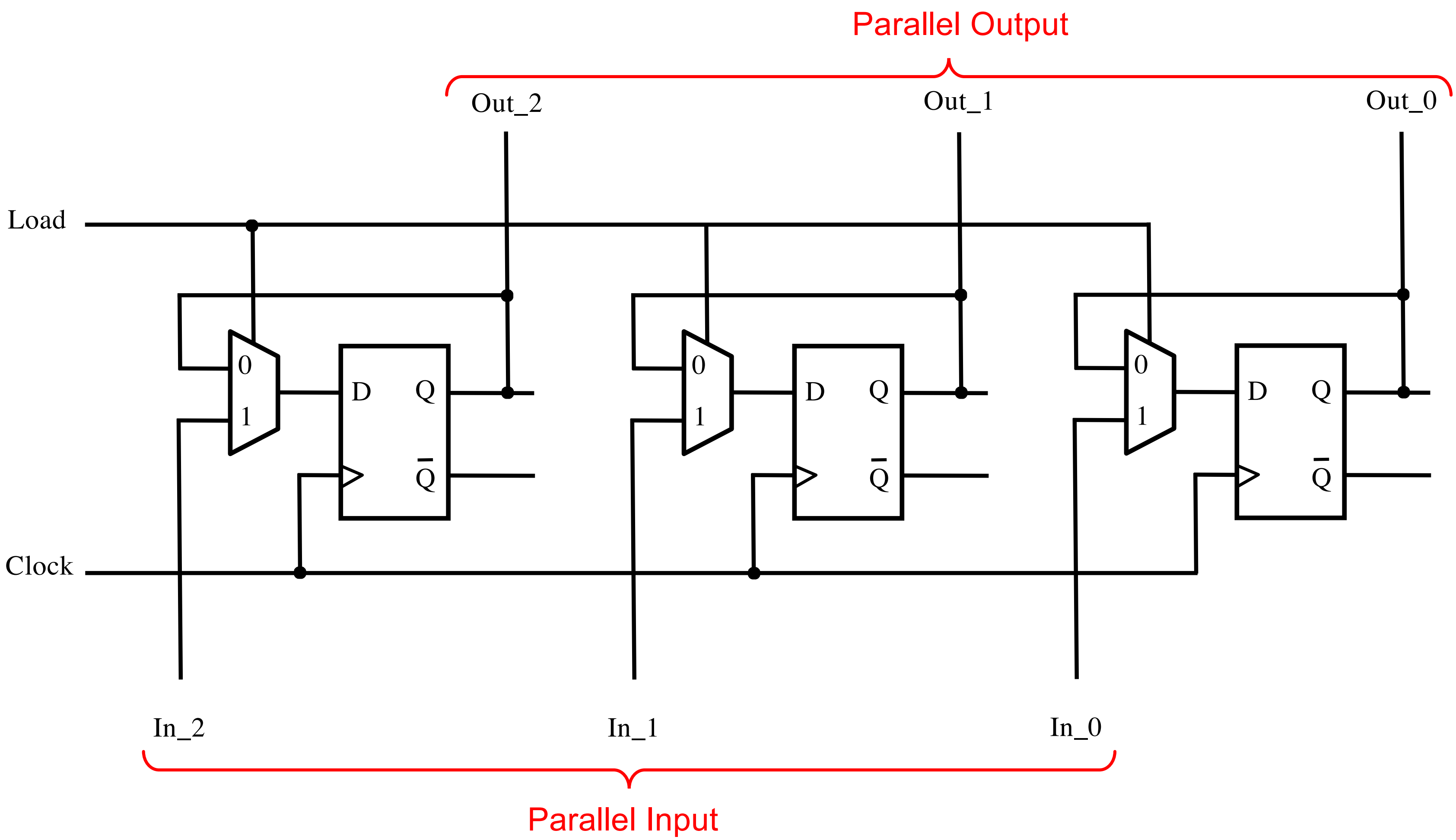


3-Bit Parallel-Access Register

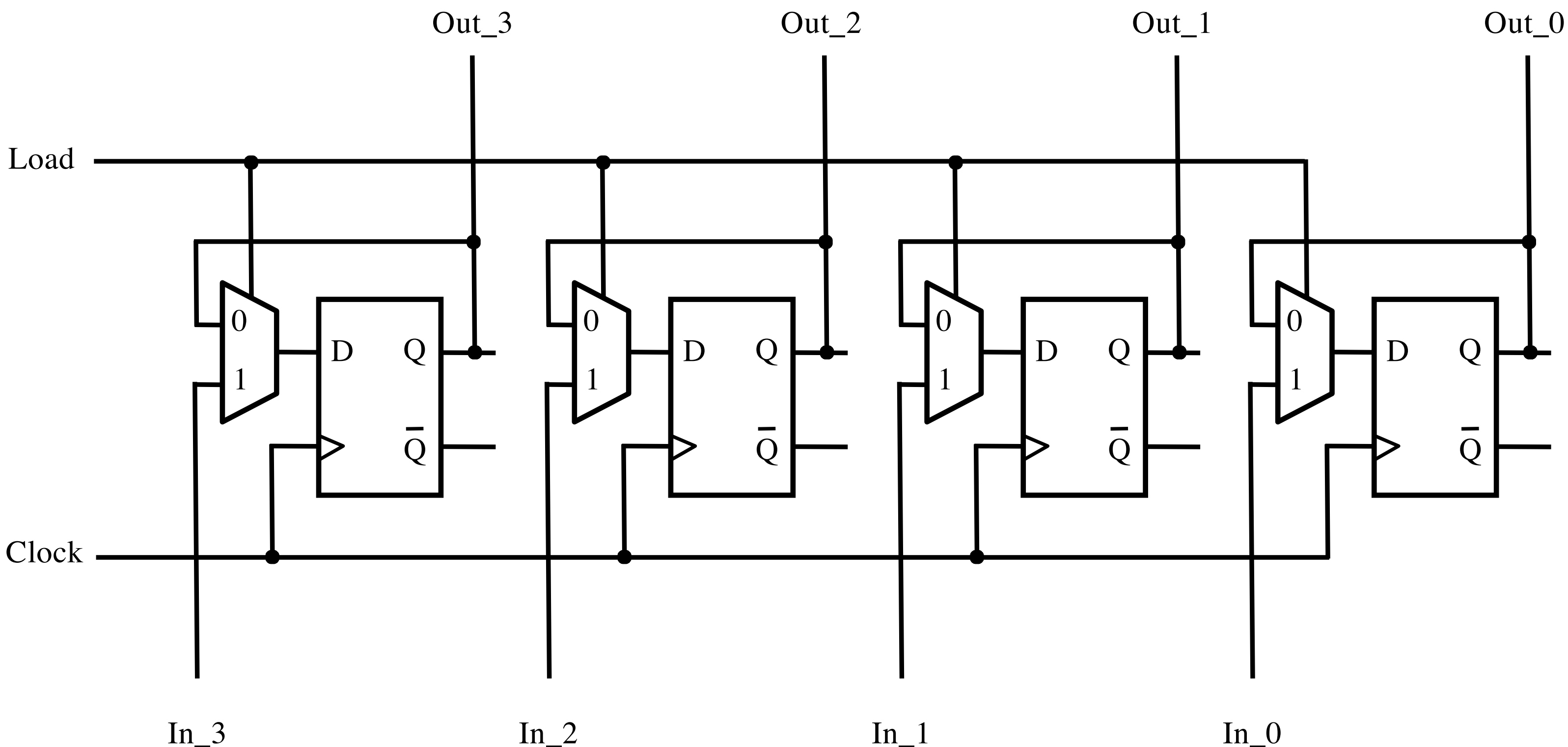


Notice that all flip-flops are on the same clock cycle.

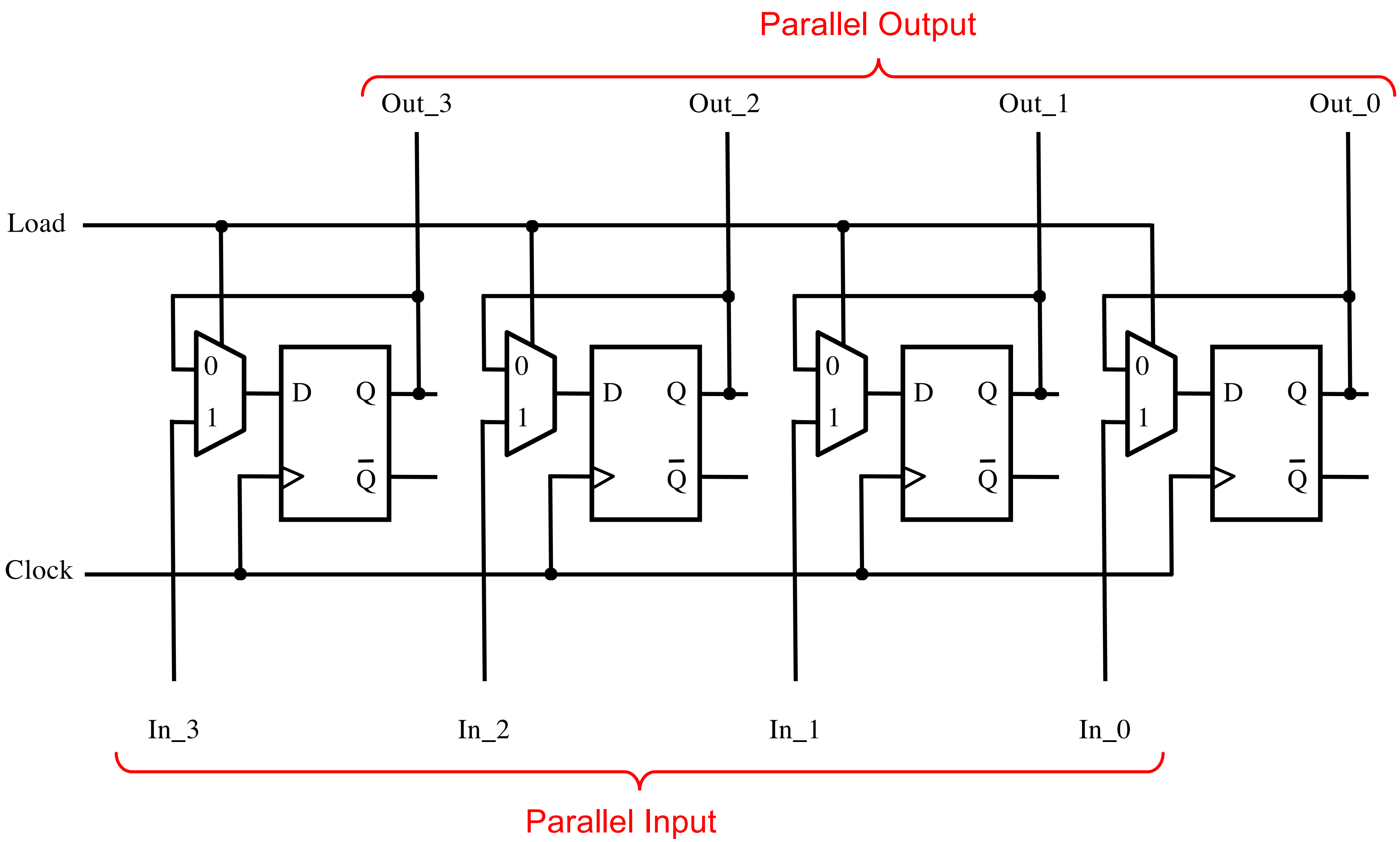
3-Bit Parallel-Access Register



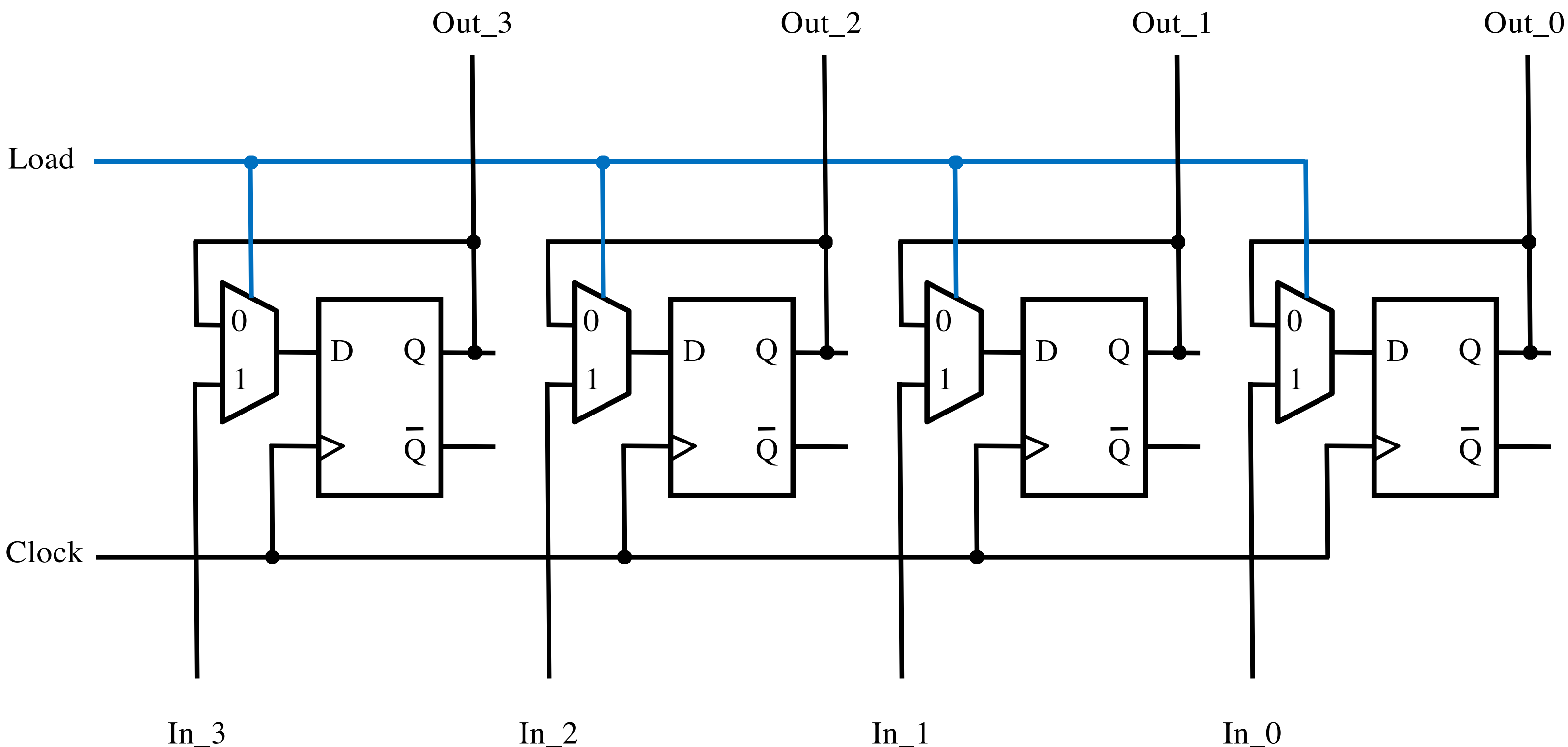
4-Bit Parallel-Access Register



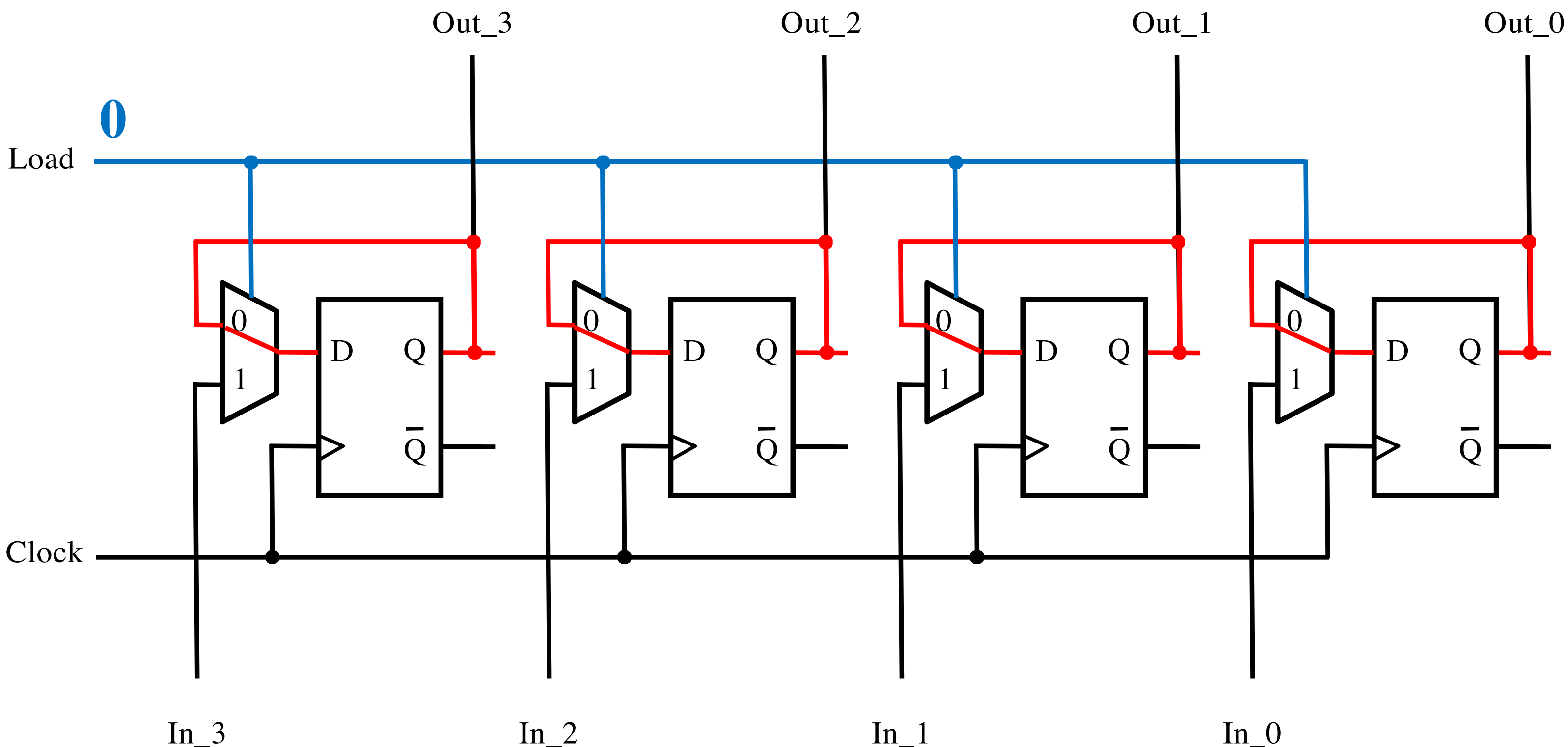
4-Bit Parallel-Access Register



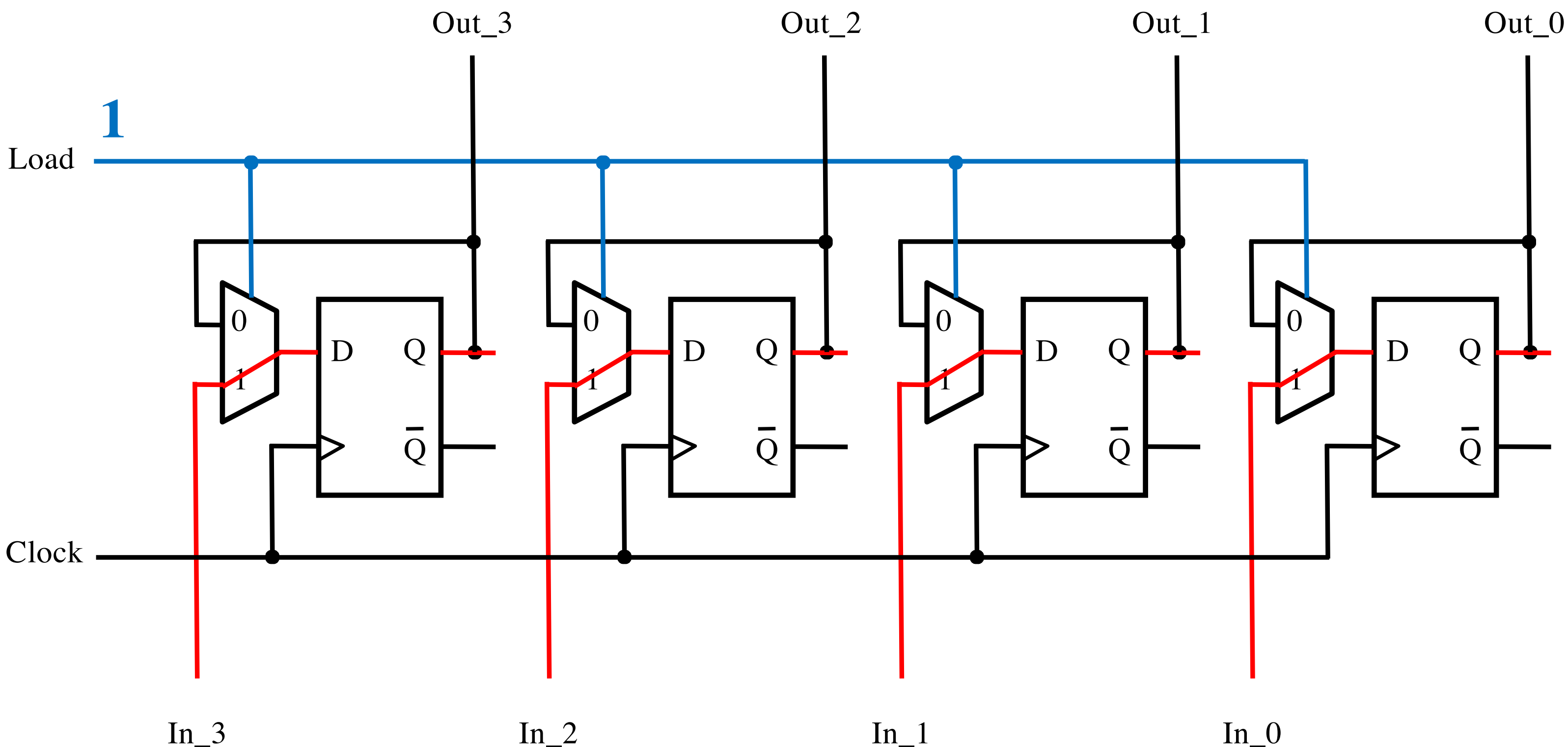
4-Bit Parallel-Access Register



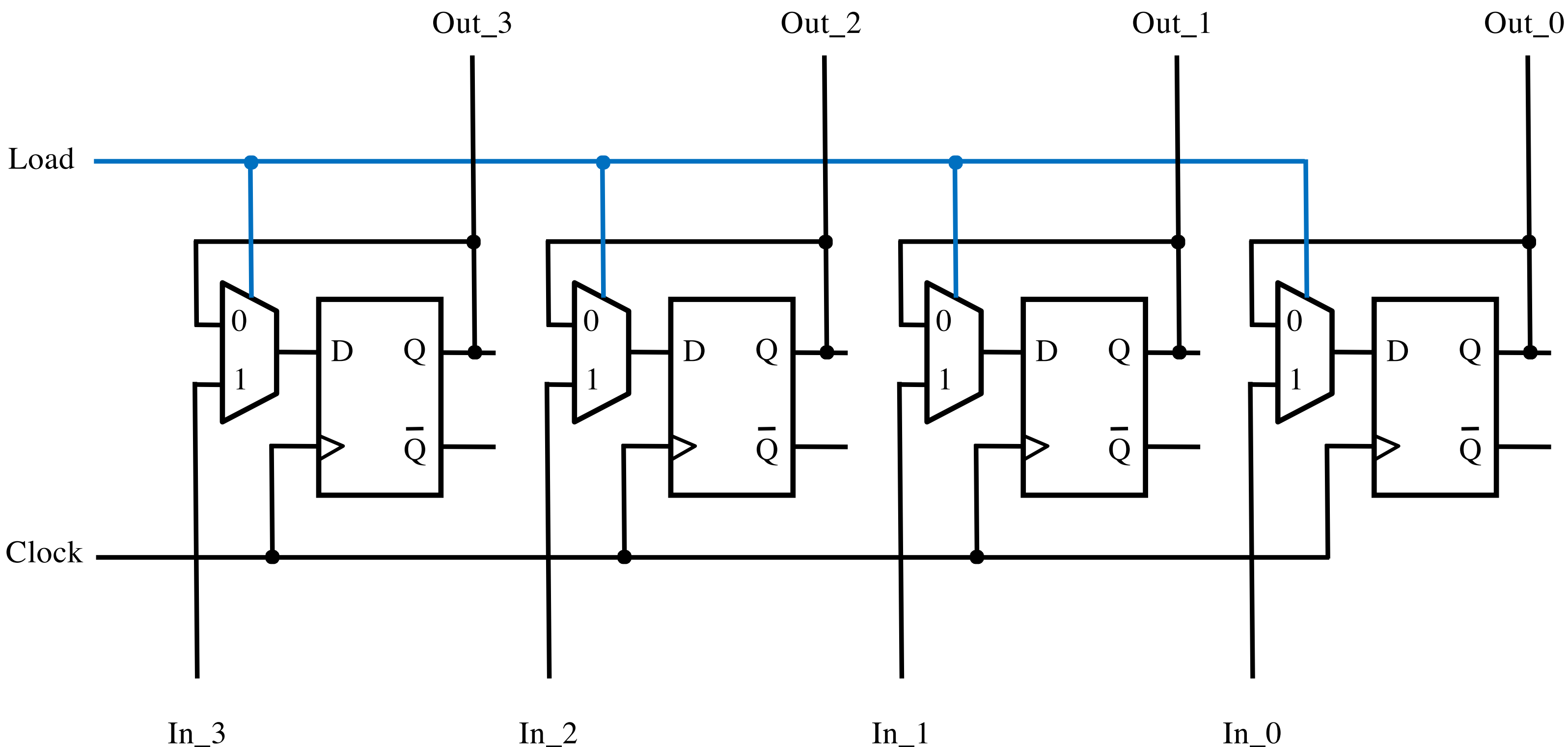
4-Bit Parallel-Access Register



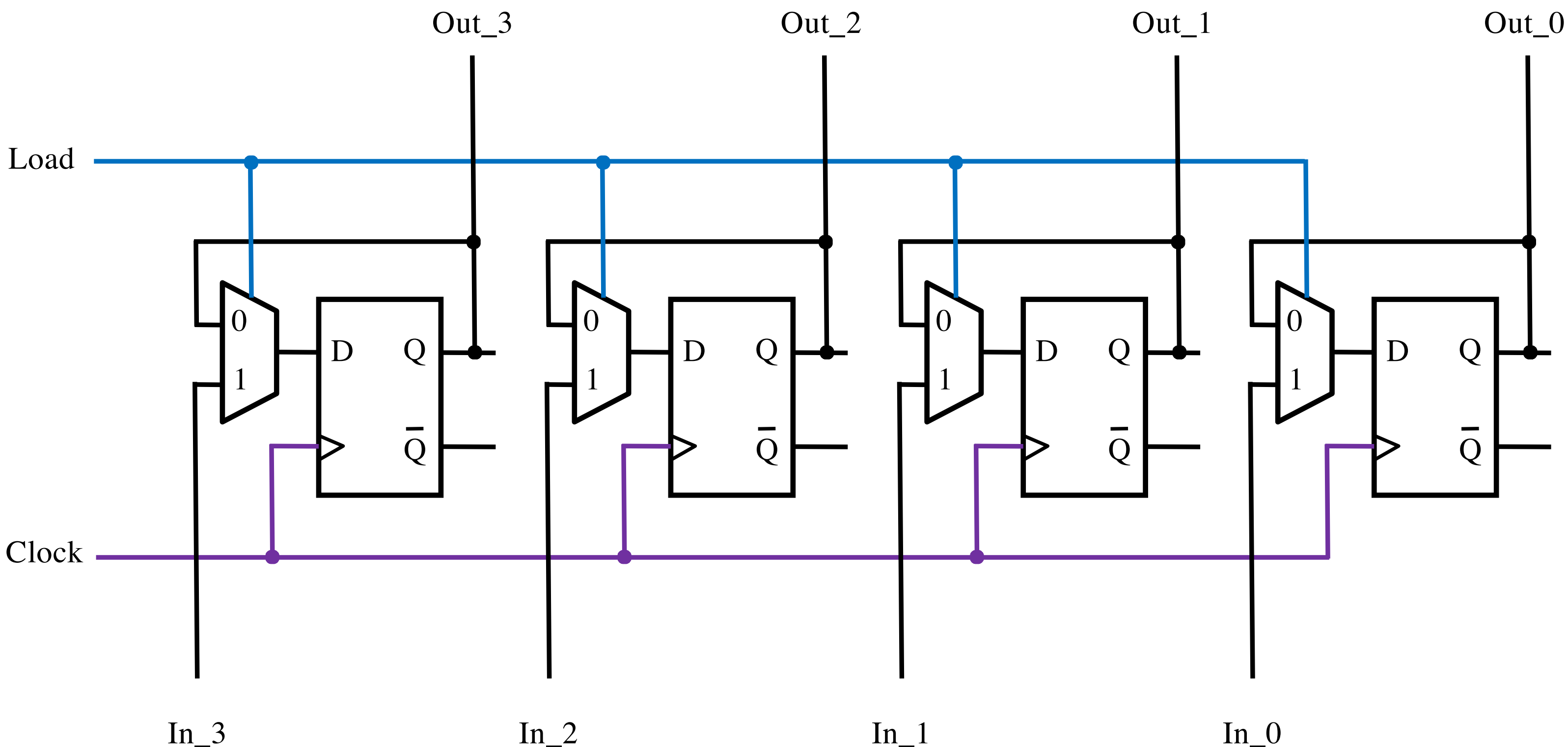
4-Bit Parallel-Access Register



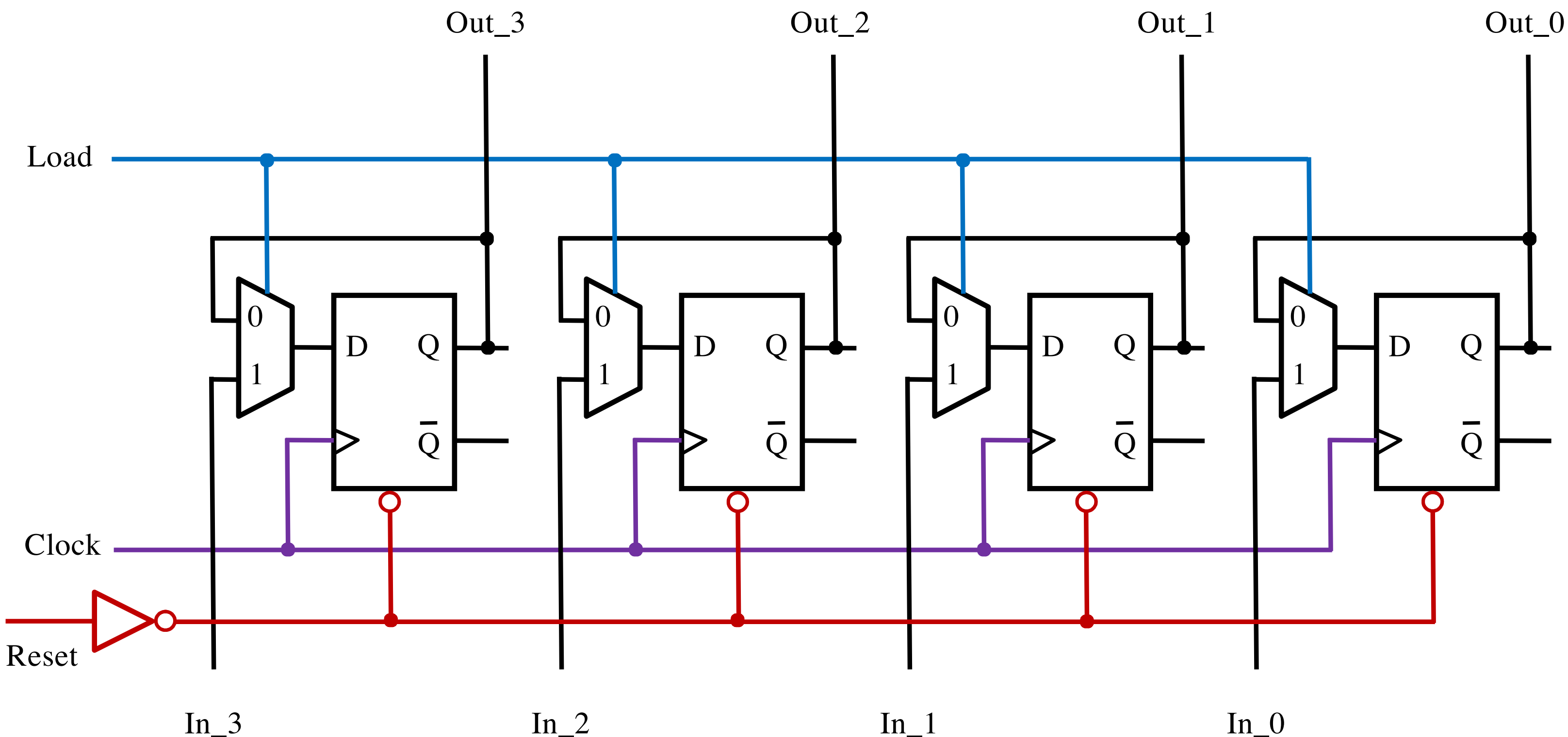
4-Bit Parallel-Access Register



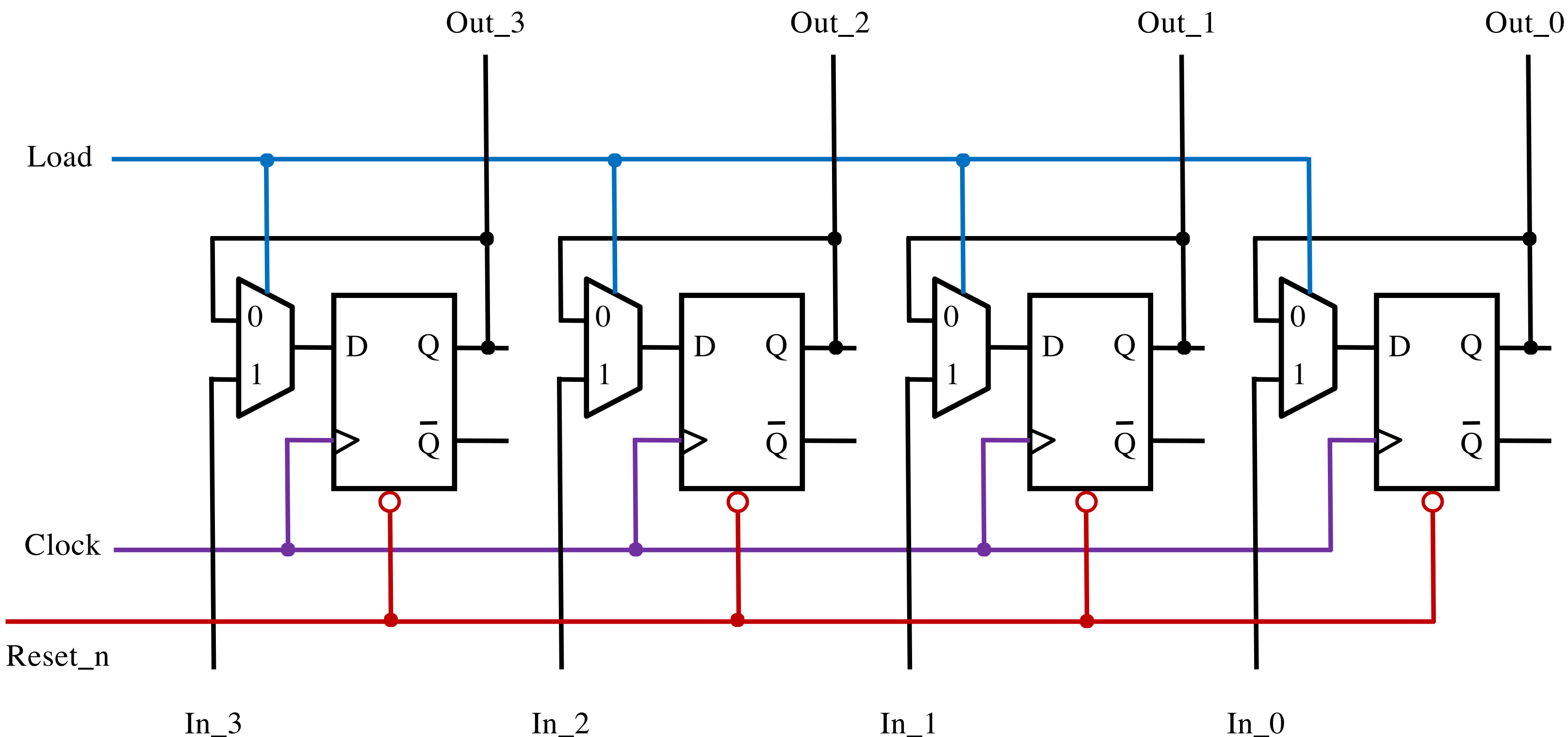
4-Bit Parallel-Access Register



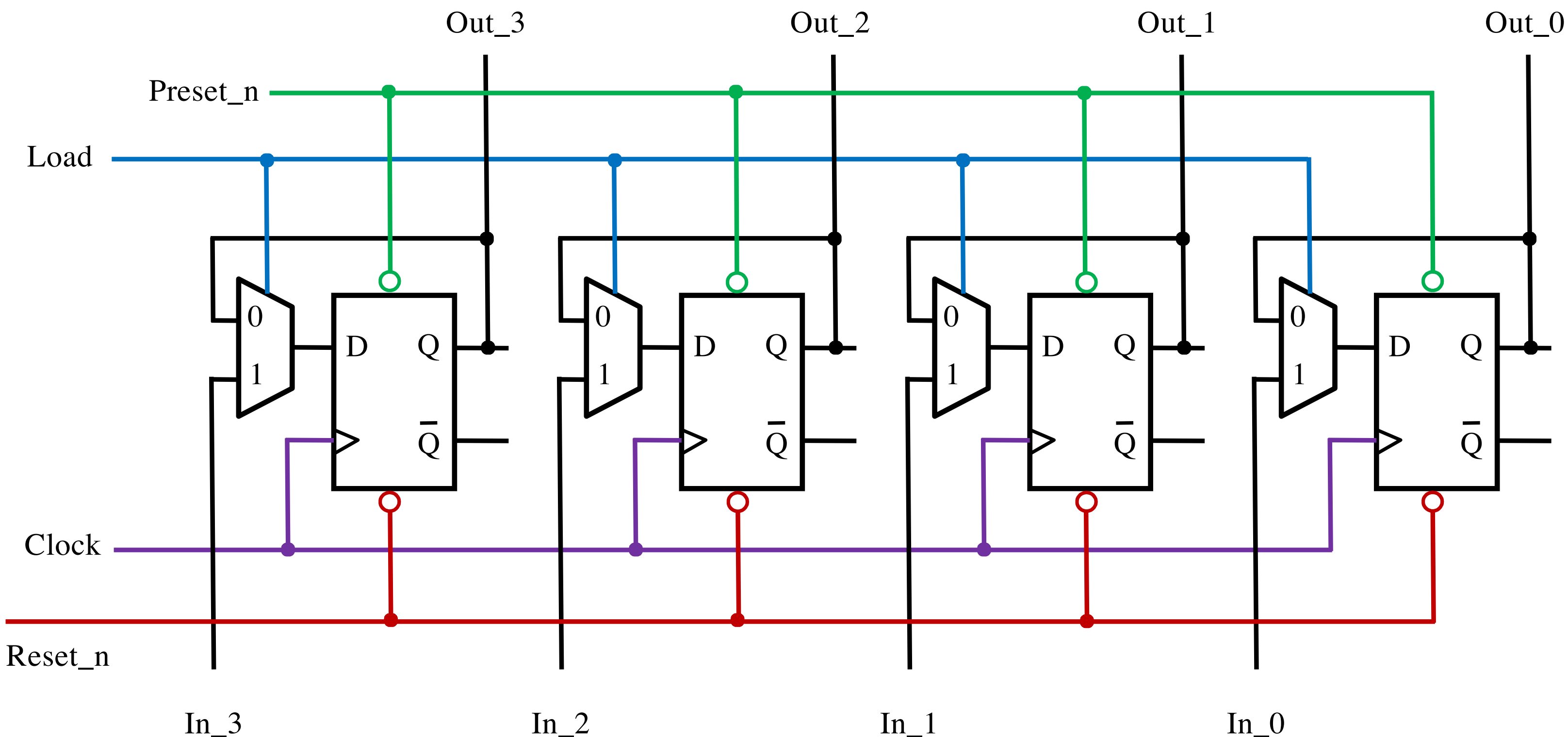
4-Bit Parallel-Access Register



4-Bit Parallel-Access Register

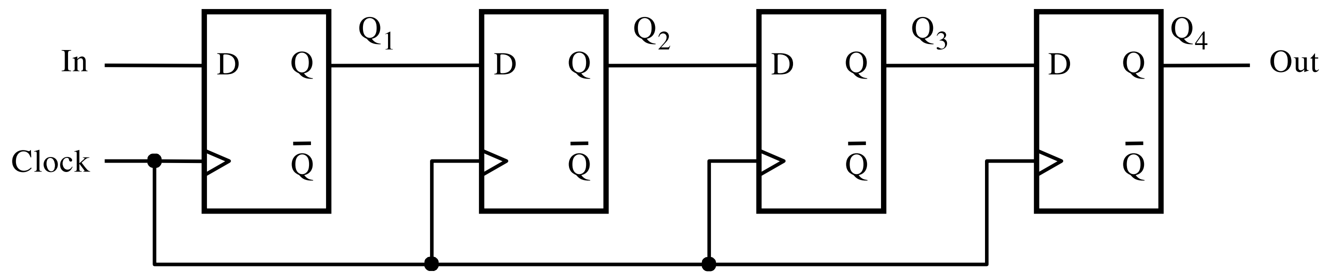


4-Bit Parallel-Access Register



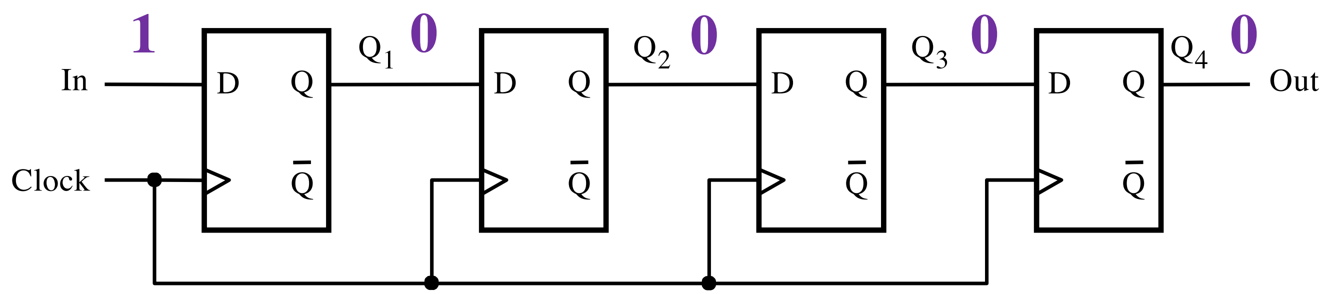
Shift Register

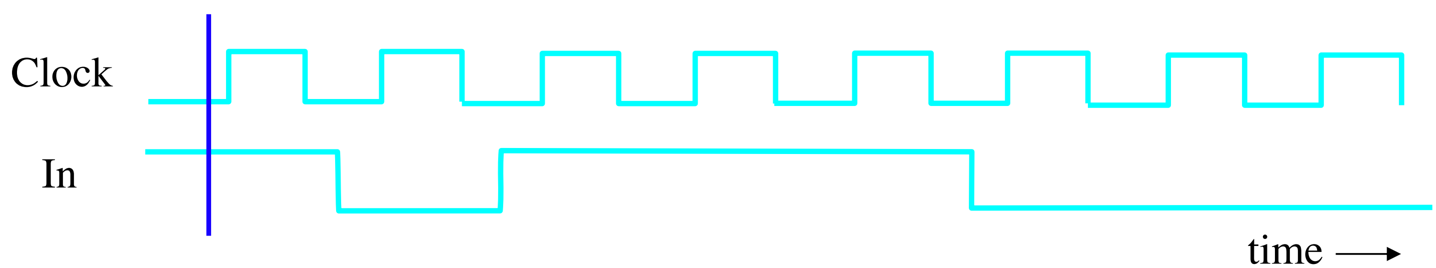
A simple shift register



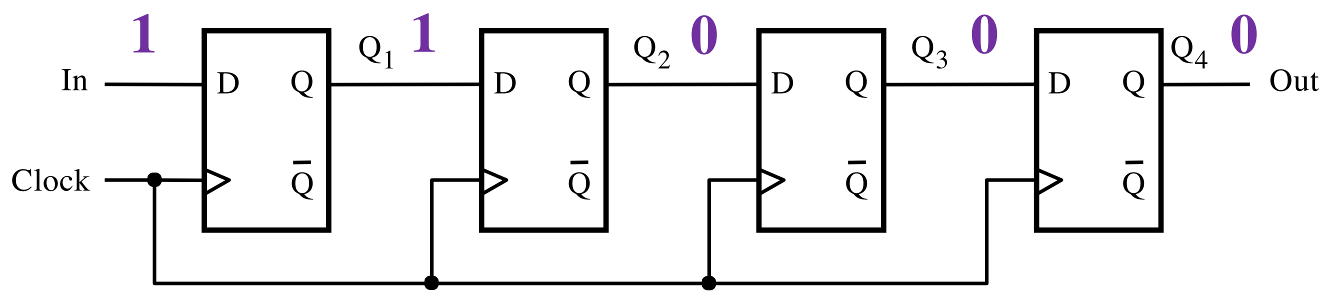
[Figure 5.17a from the textbook]

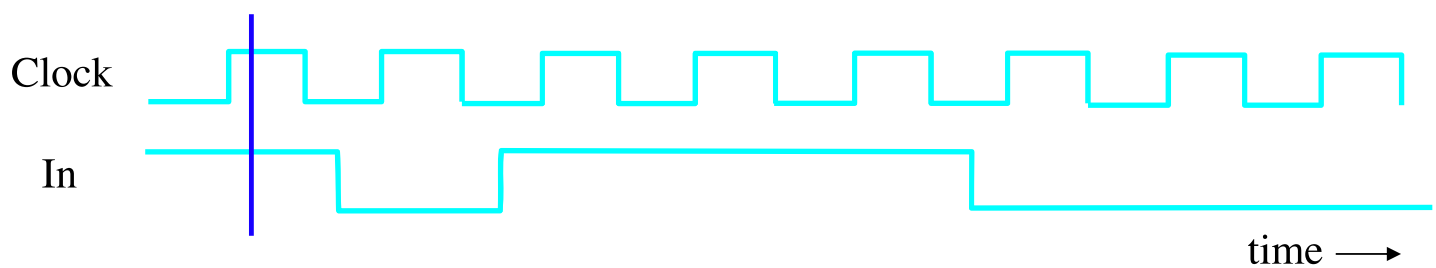
Shift Register Simulation



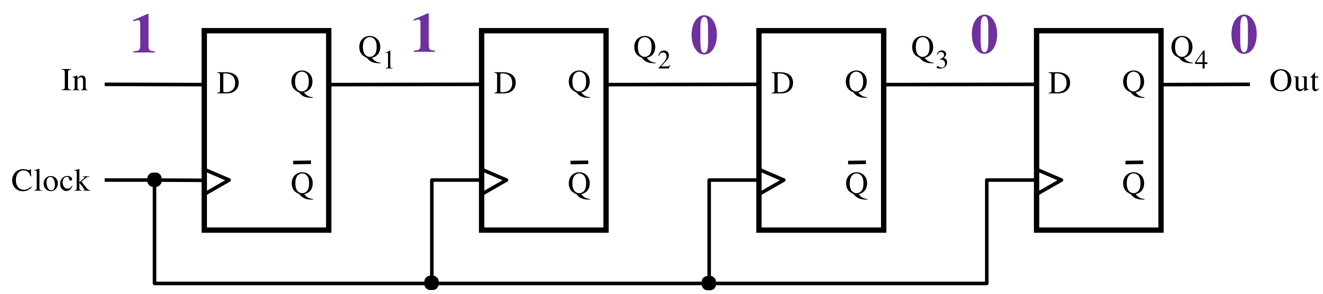


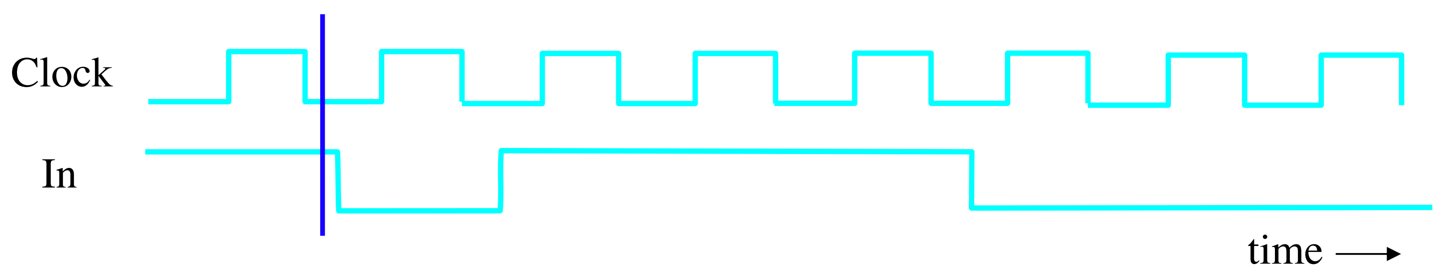
Shift Register Simulation



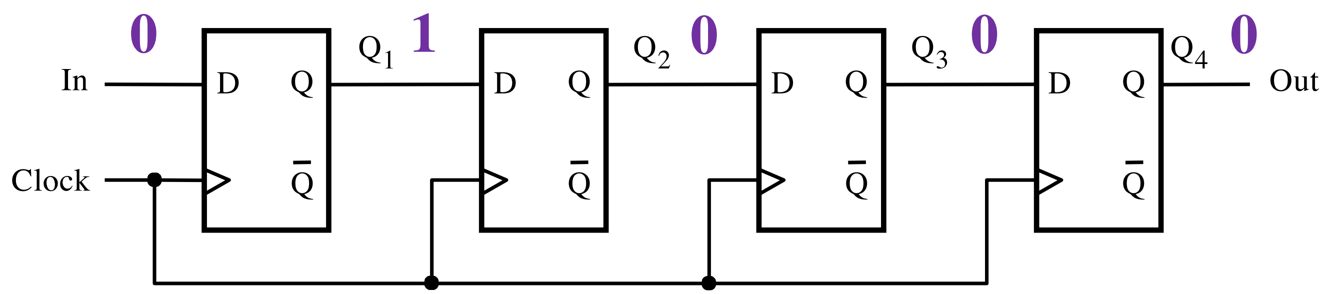


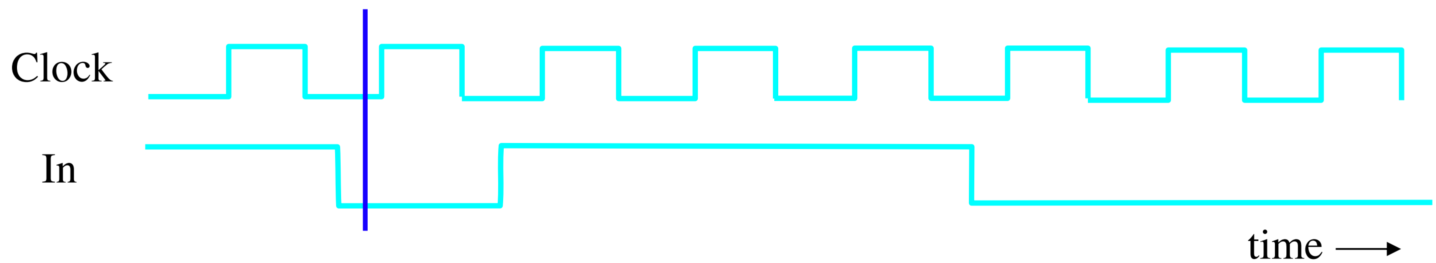
Shift Register Simulation



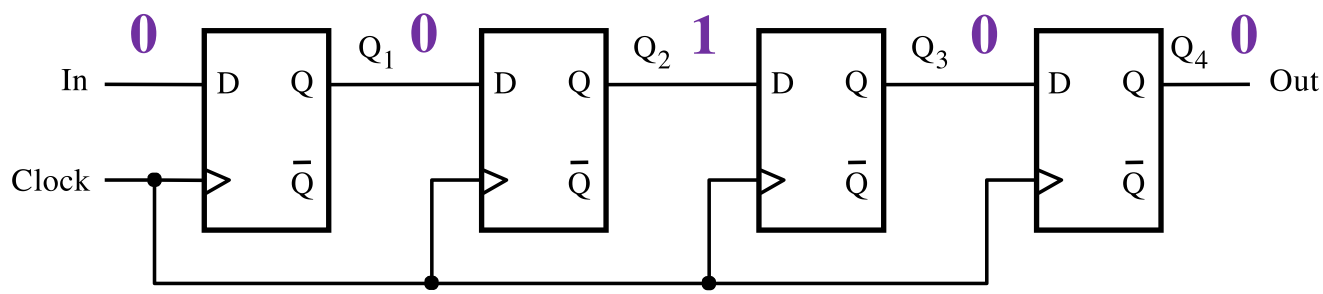


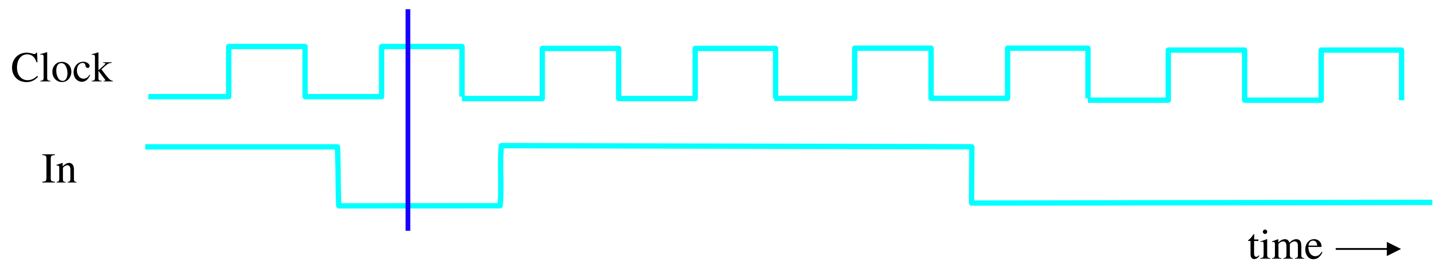
Shift Register Simulation



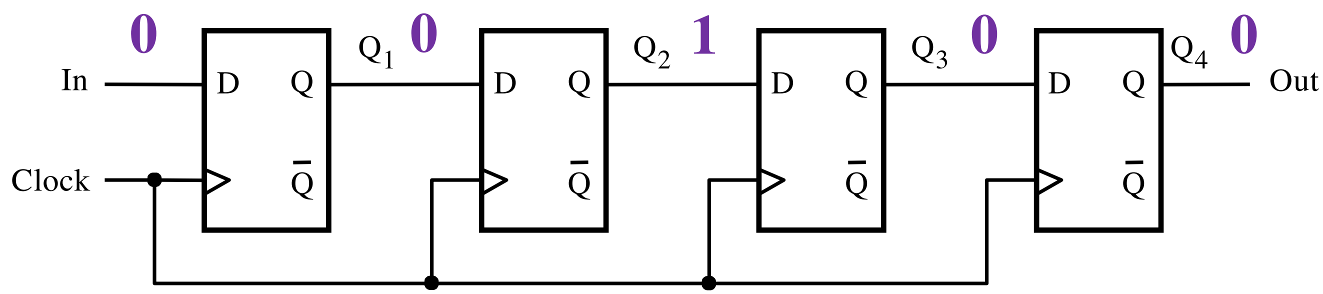


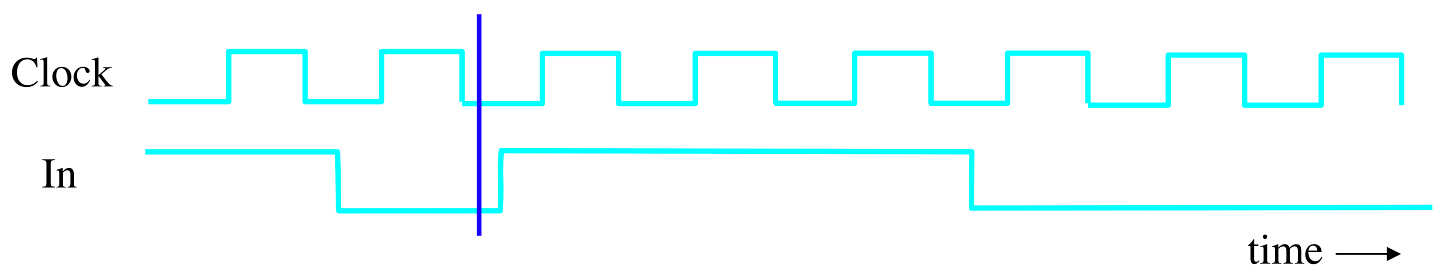
Shift Register Simulation



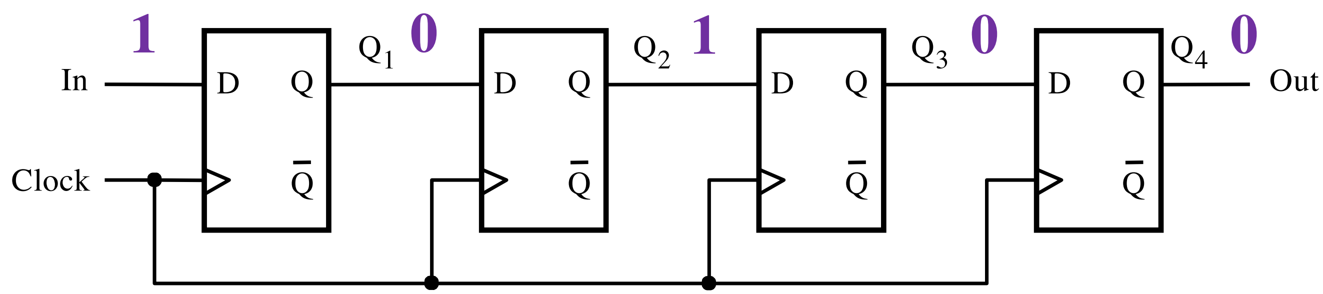


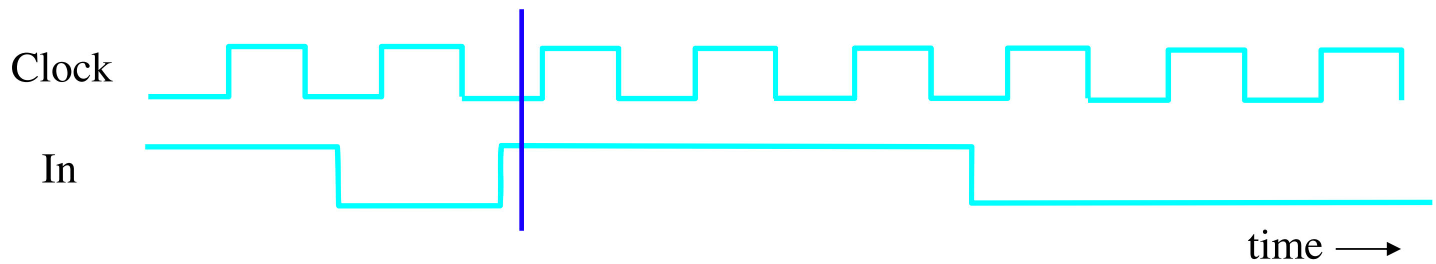
Shift Register Simulation



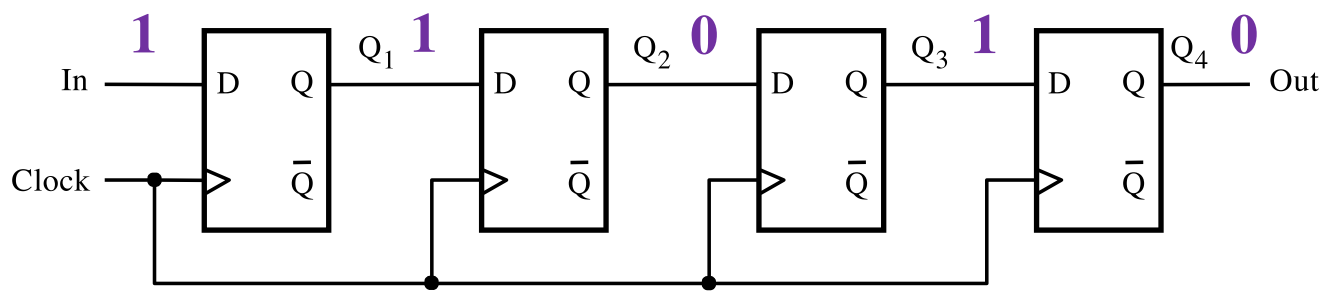


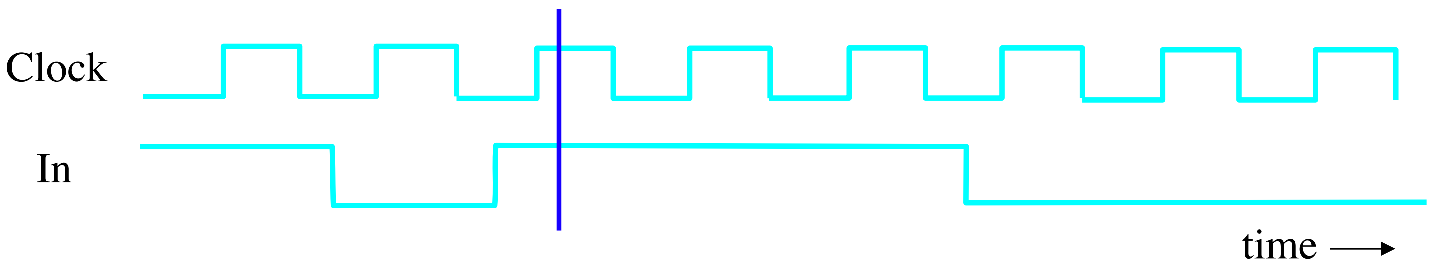
Shift Register Simulation



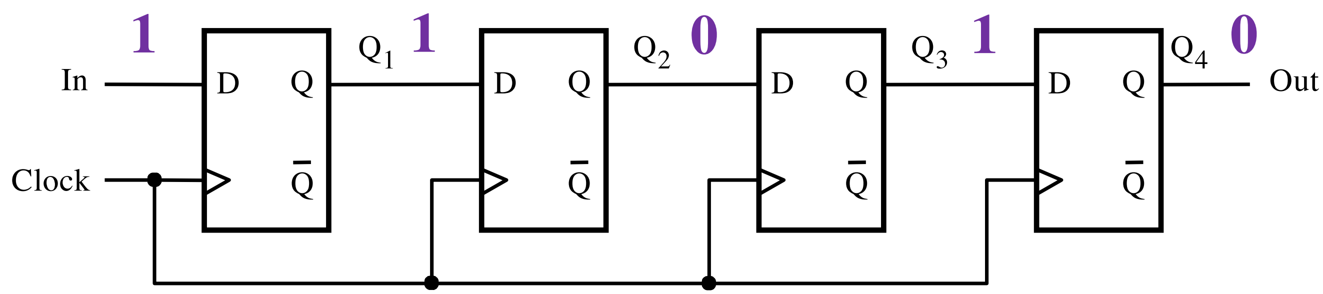


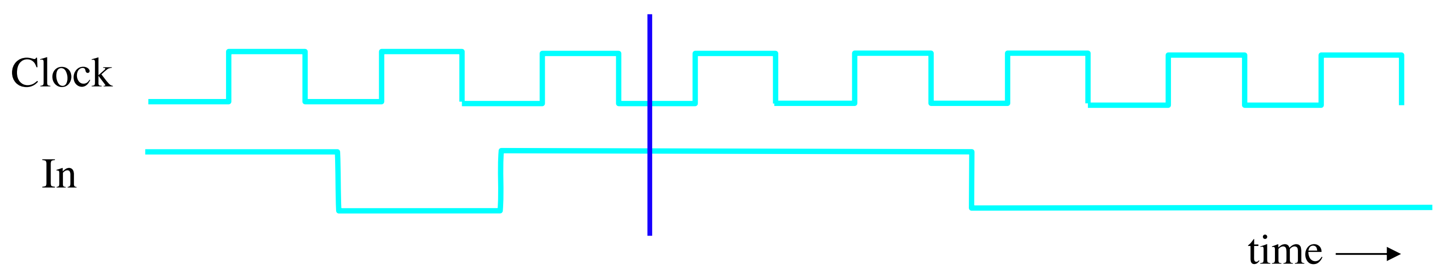
Shift Register Simulation



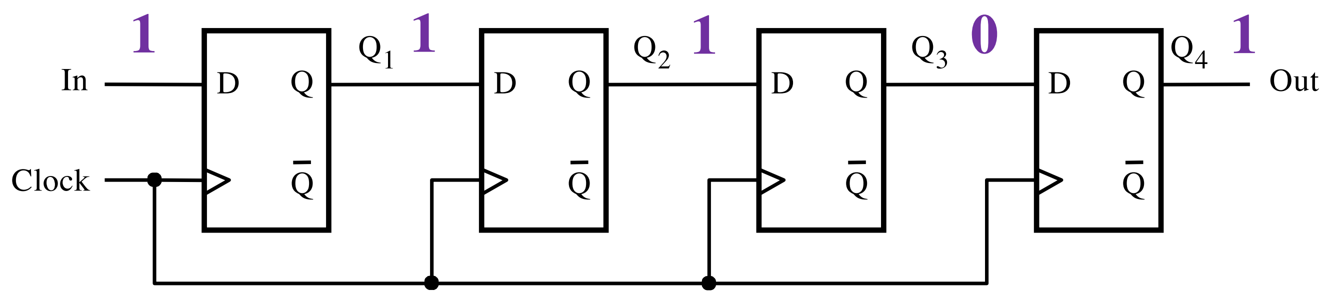


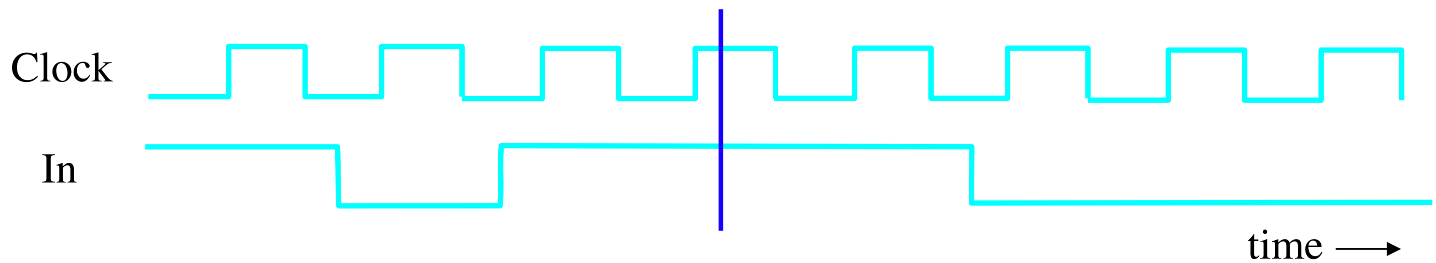
Shift Register Simulation



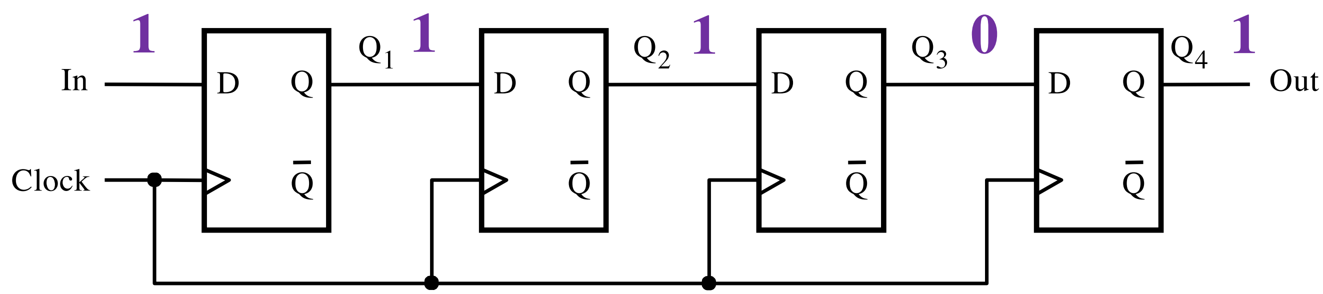


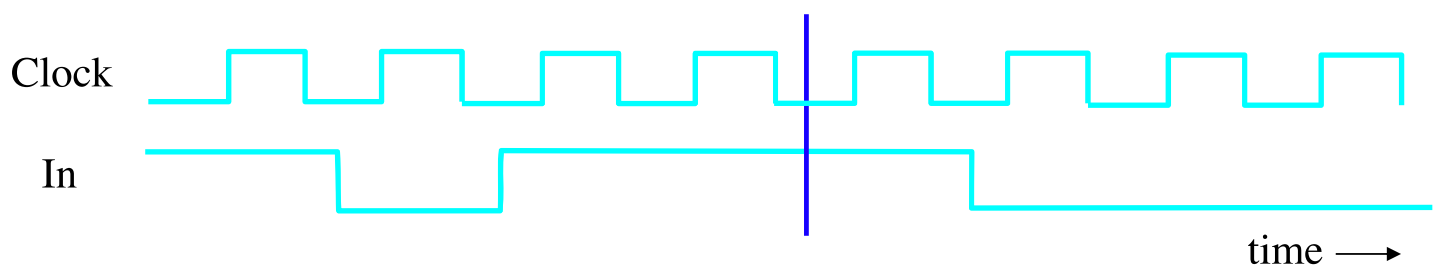
Shift Register Simulation



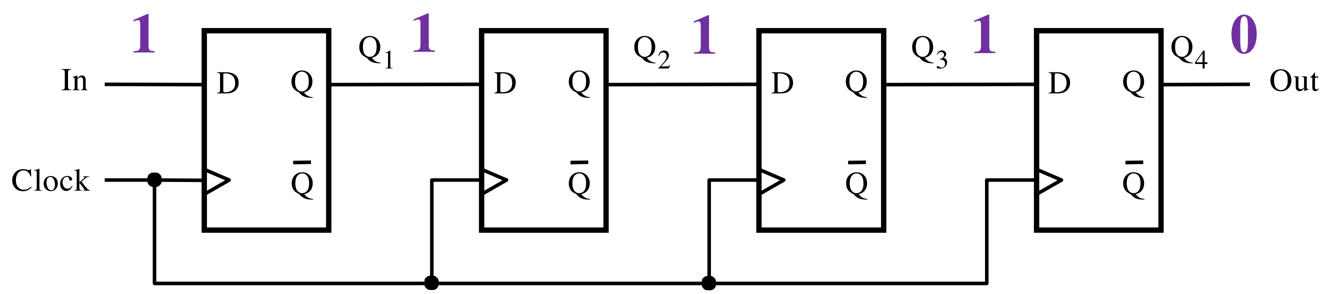


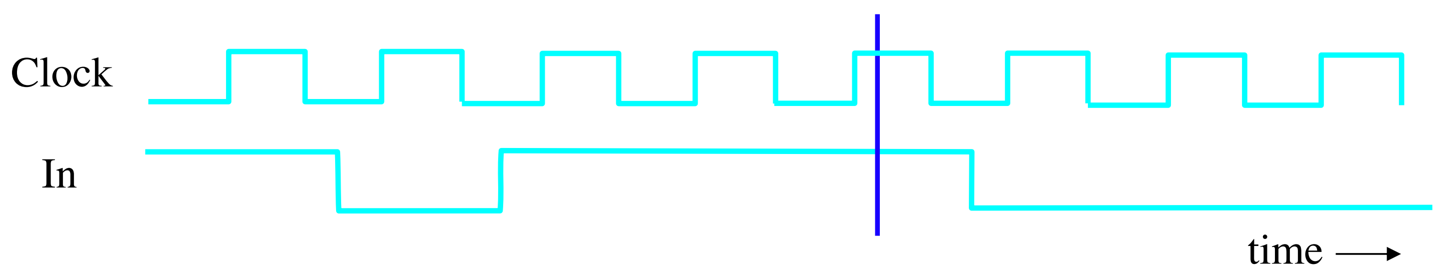
Shift Register Simulation



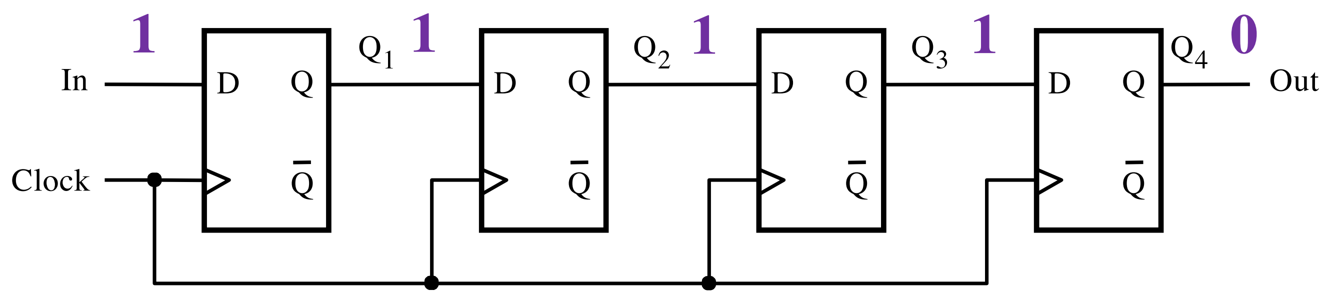


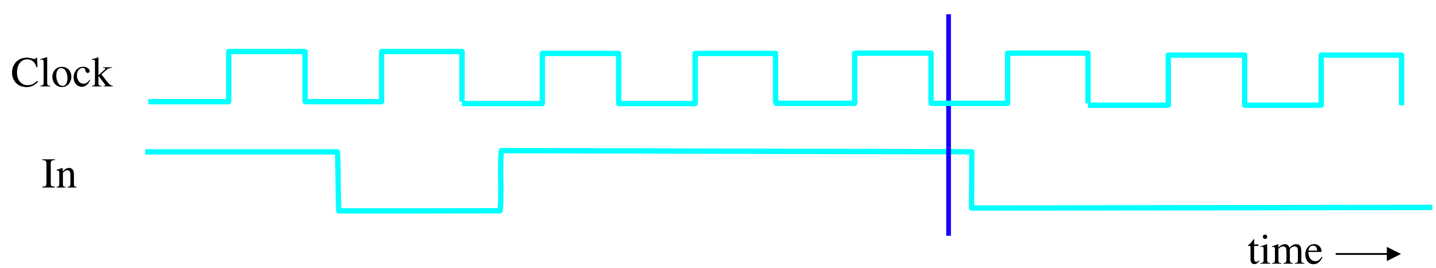
Shift Register Simulation



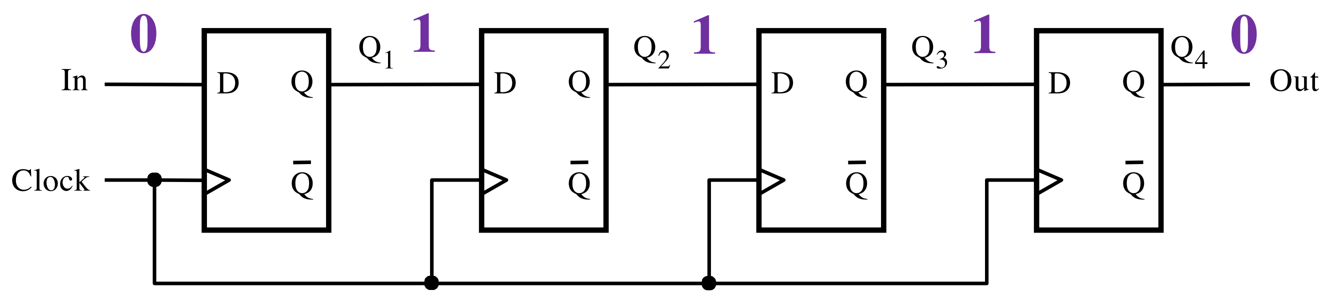


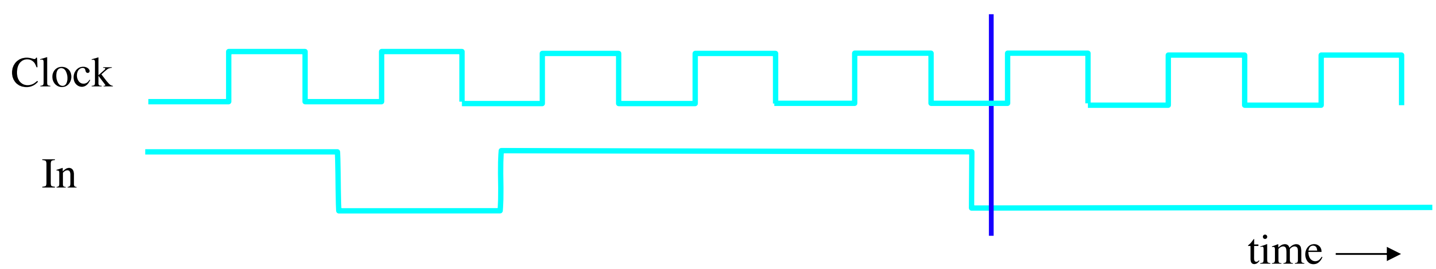
Shift Register Simulation



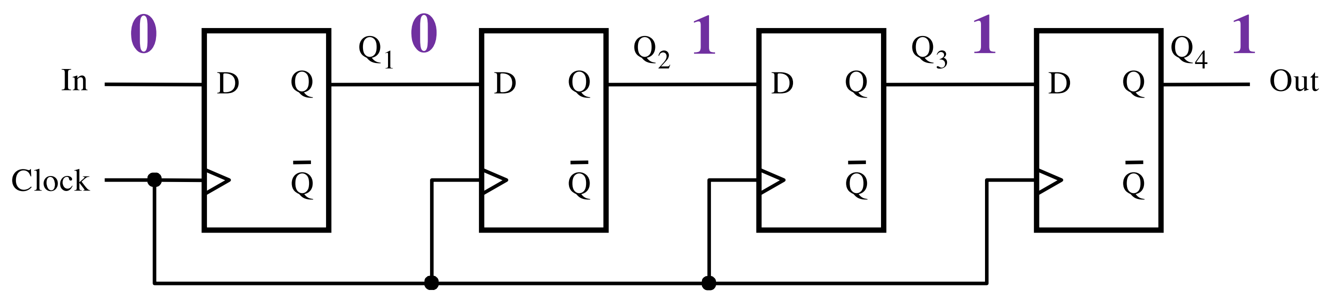


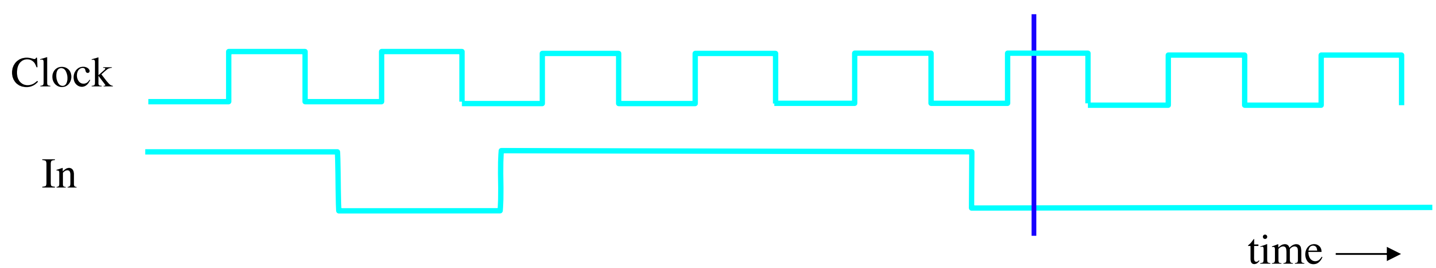
Shift Register Simulation



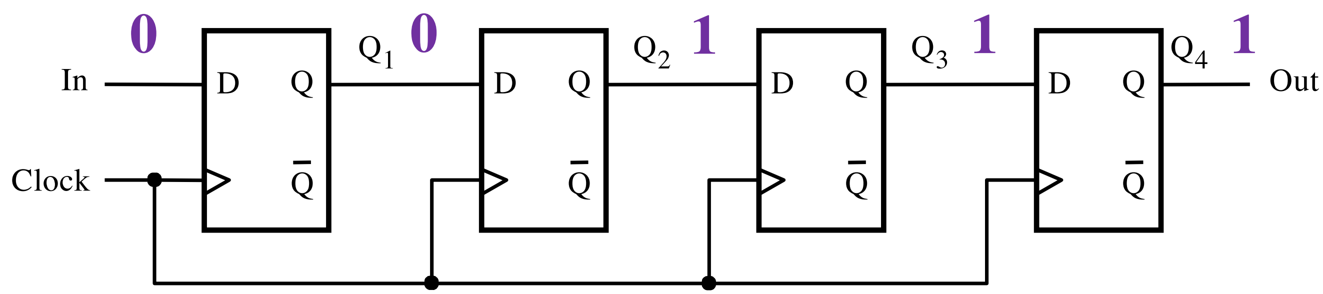


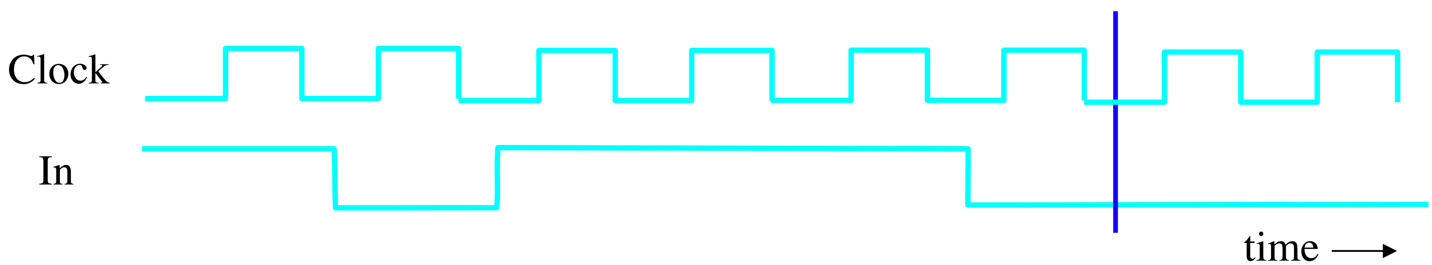
Shift Register Simulation



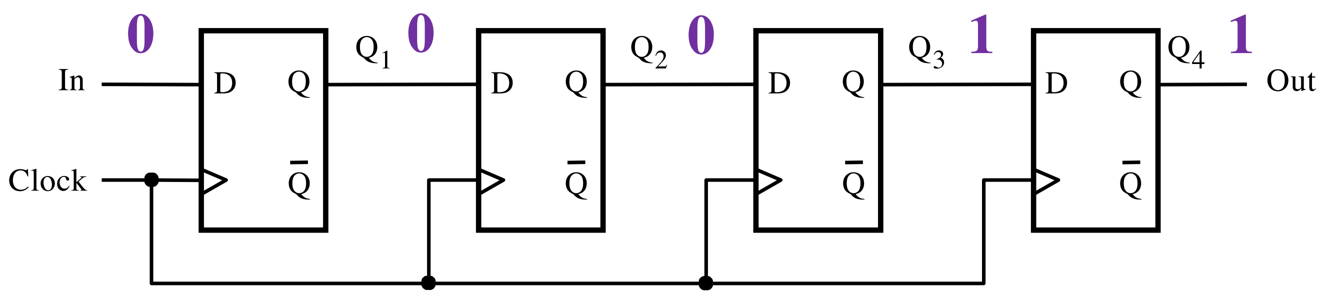


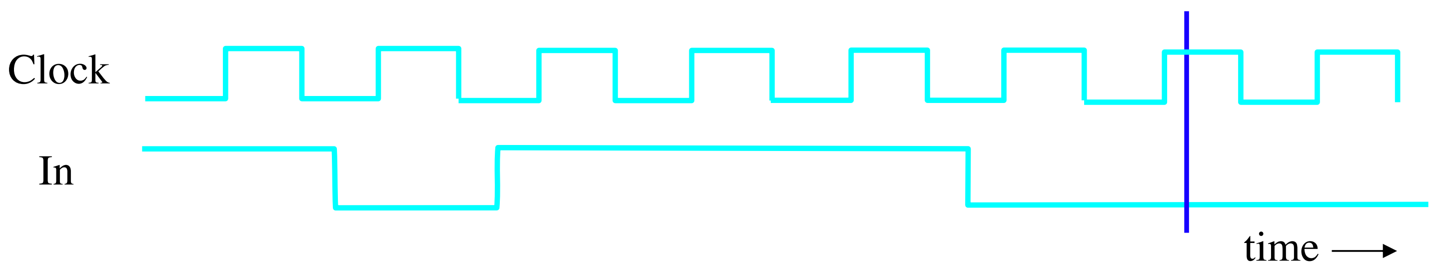
Shift Register Simulation



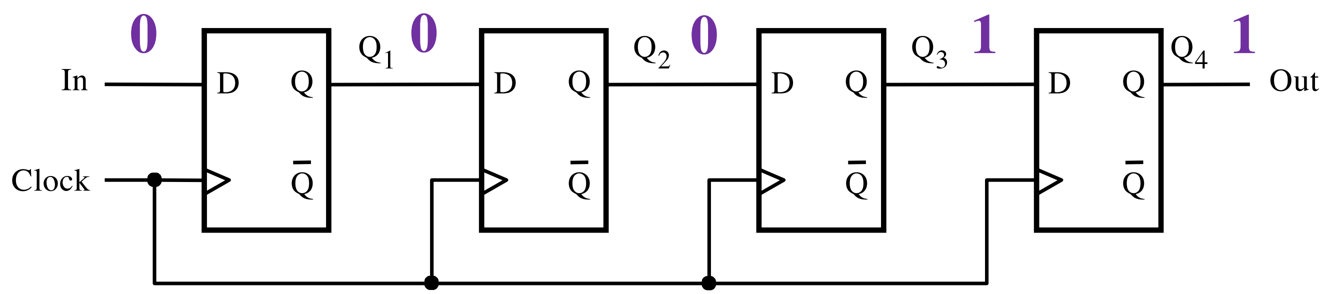


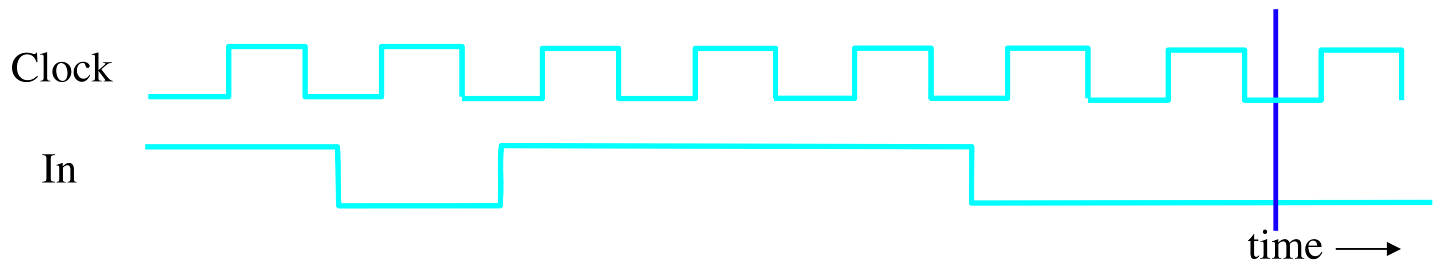
Shift Register Simulation



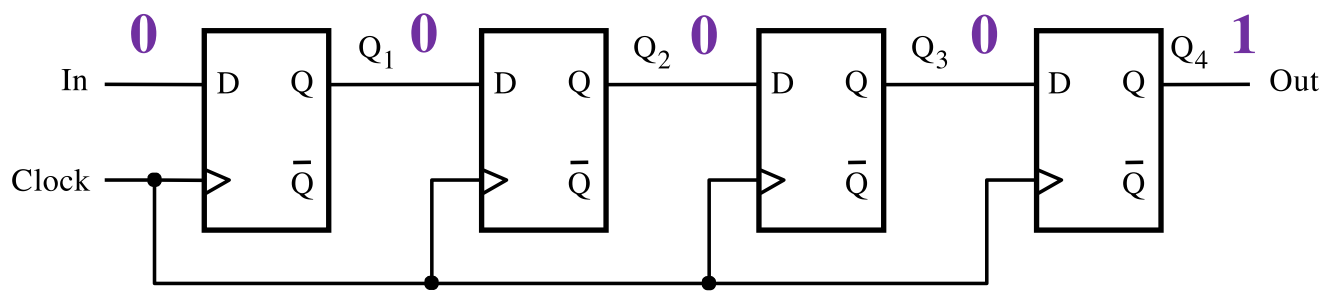


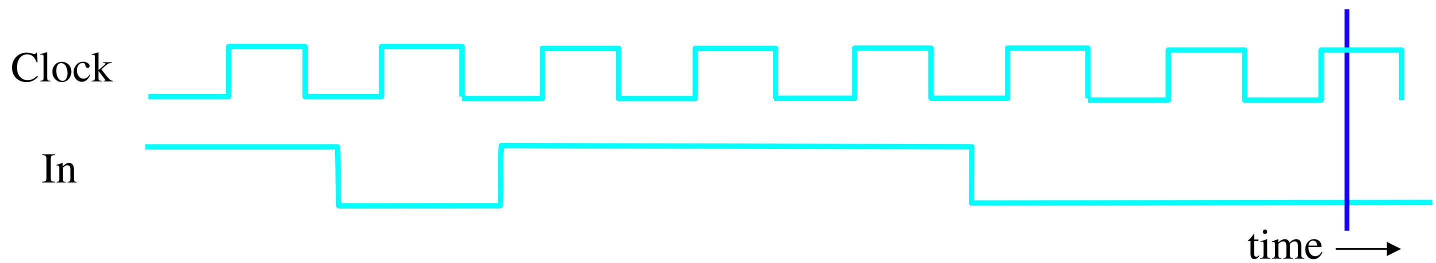
Shift Register Simulation



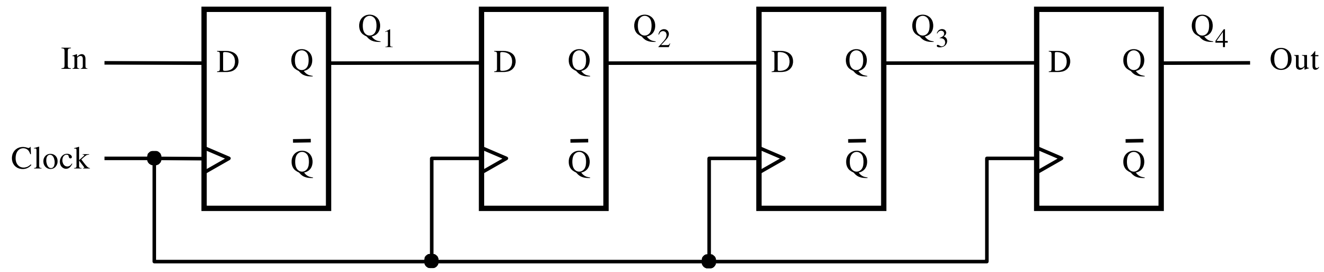


Shift Register Simulation





A simple shift register



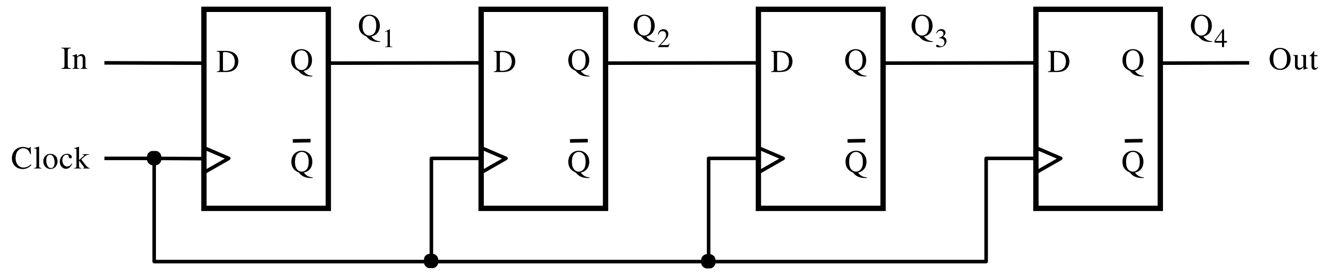
(a) Circuit

time		In	Q_1	Q_2	Q_3	Q_4 = Out
	t_0	1	0	0	0	0
	t_1	0	1	0	0	0
	t_2	1	0	1	0	0
	t_3	1	1	0	1	0
	t_4	1	1	1	0	1
	t_5	0	1	1	1	0
	t_6	0	0	1	1	1
	t_7	0	0	0	1	1

(b) A sample sequence

[Figure 5.17 from the textbook]

A simple shift register



(a) Circuit

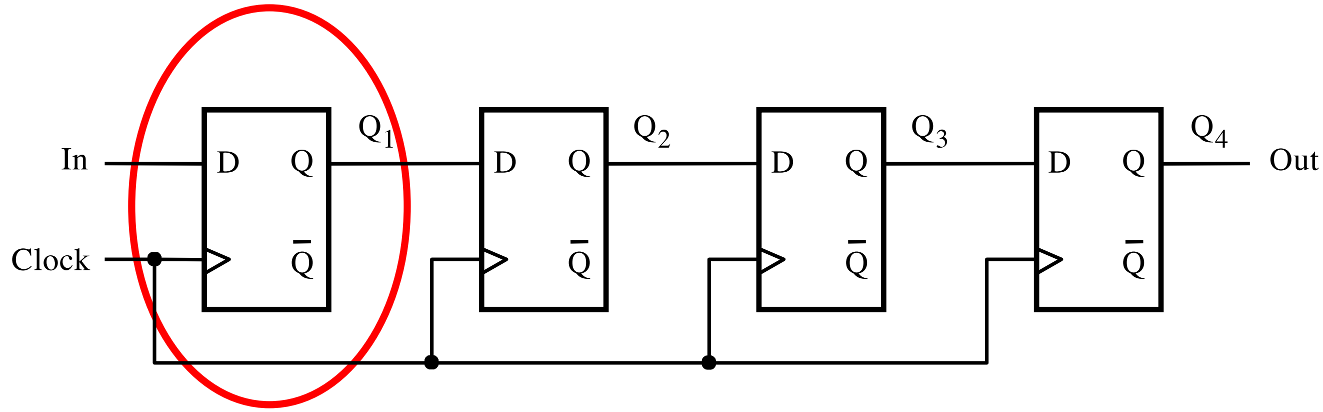
time

	In	Q_1	Q_2	Q_3	Q_4 = Out
t_0	1	0	0	0	0
t_1	0	1	0	0	0
t_2	1	0	1	0	0
t_3	1	1	0	1	0
t_4	1	1	1	0	1
t_5	0	1	1	1	0
t_6	0	0	1	1	1
t_7	0	0	0	1	1
	0	0	0	0	1

The simulation goes one step further

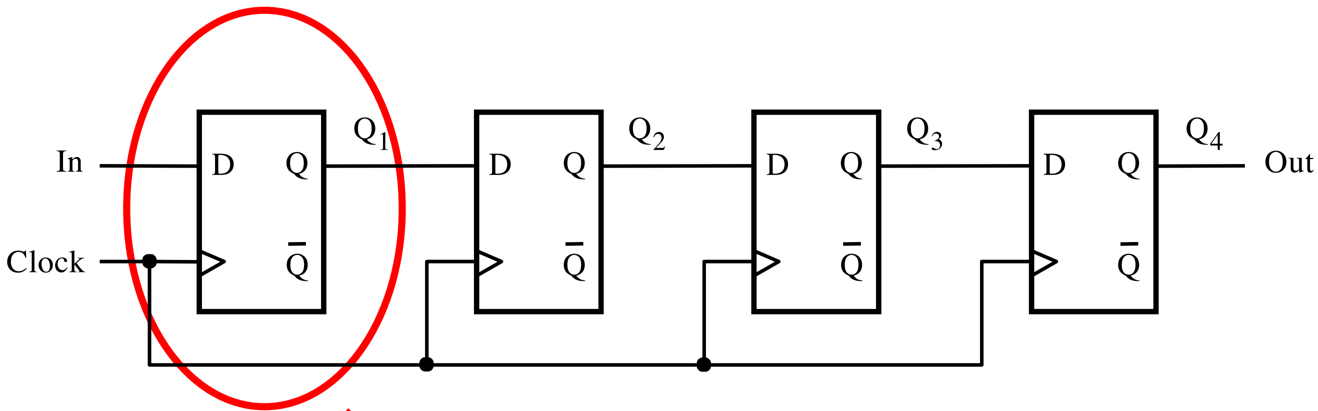
[Figure 5.17 from the textbook]

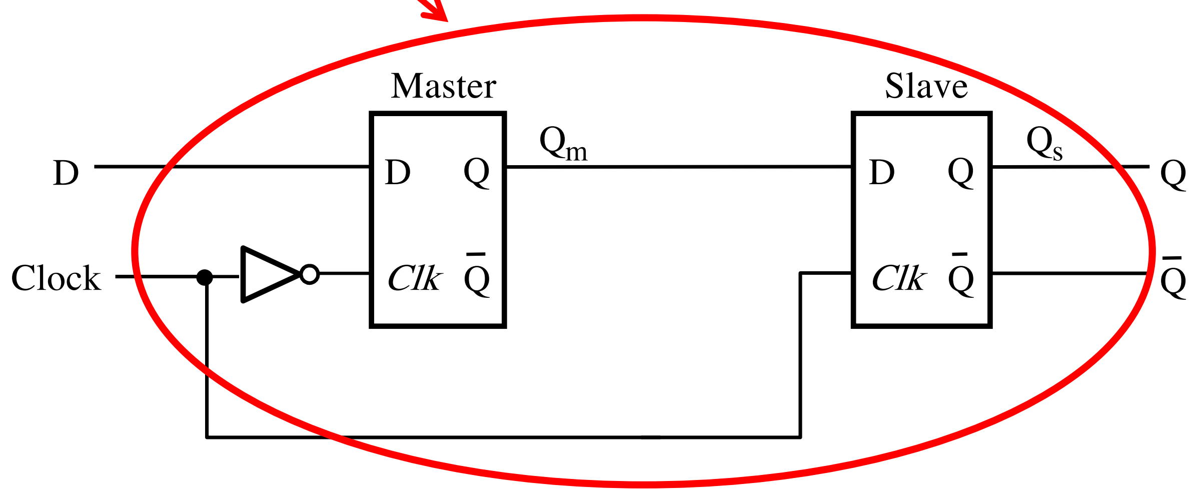
A simple shift register



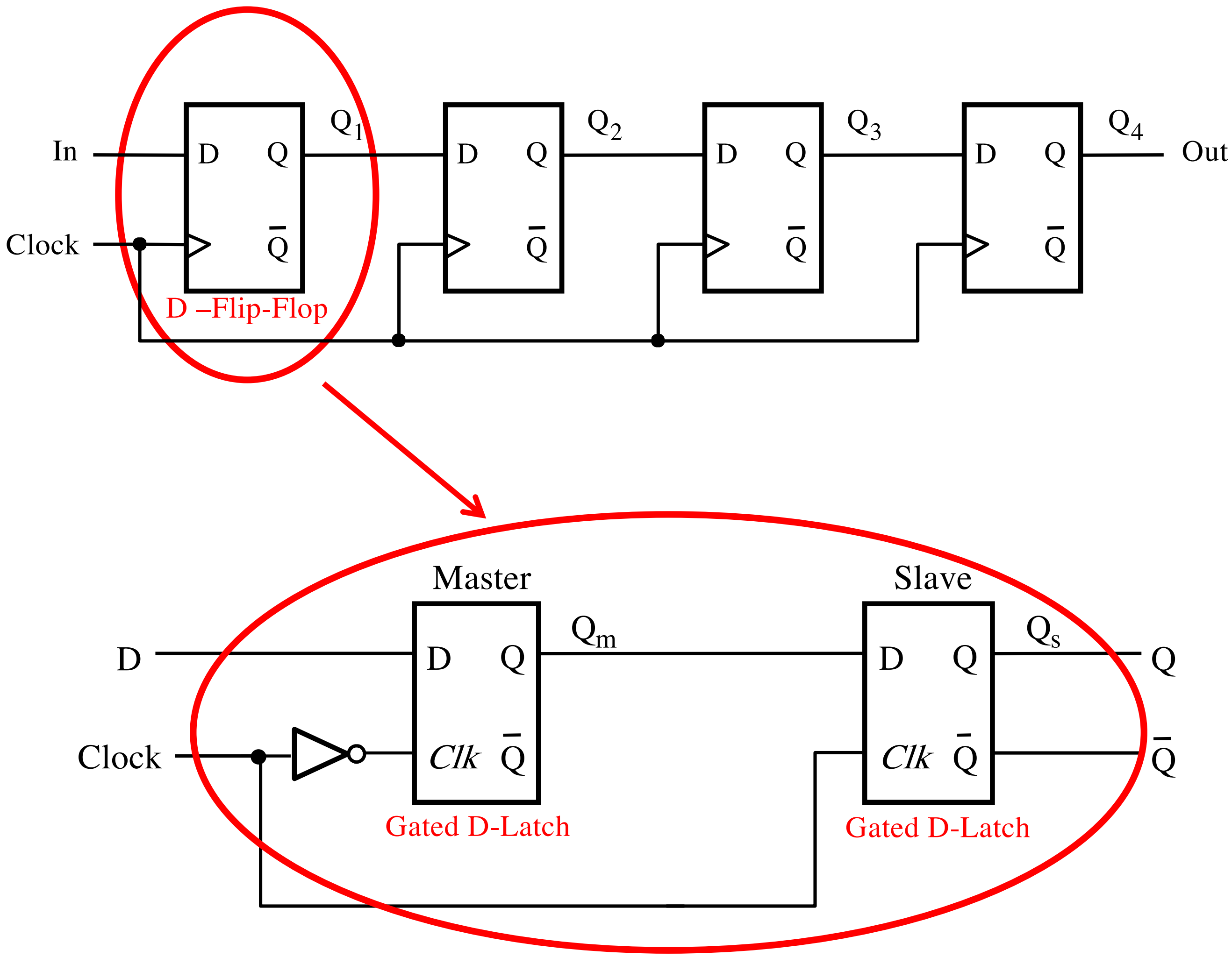
Positive-edge-triggered
D Flip-Flop

A simple shift register

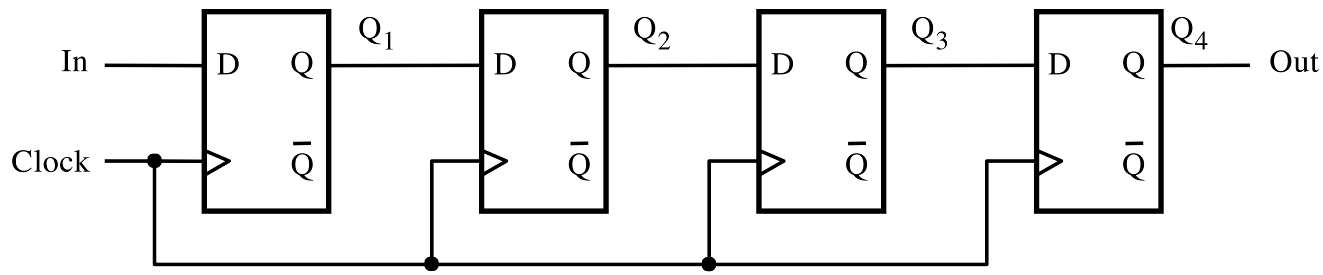




A simple shift register



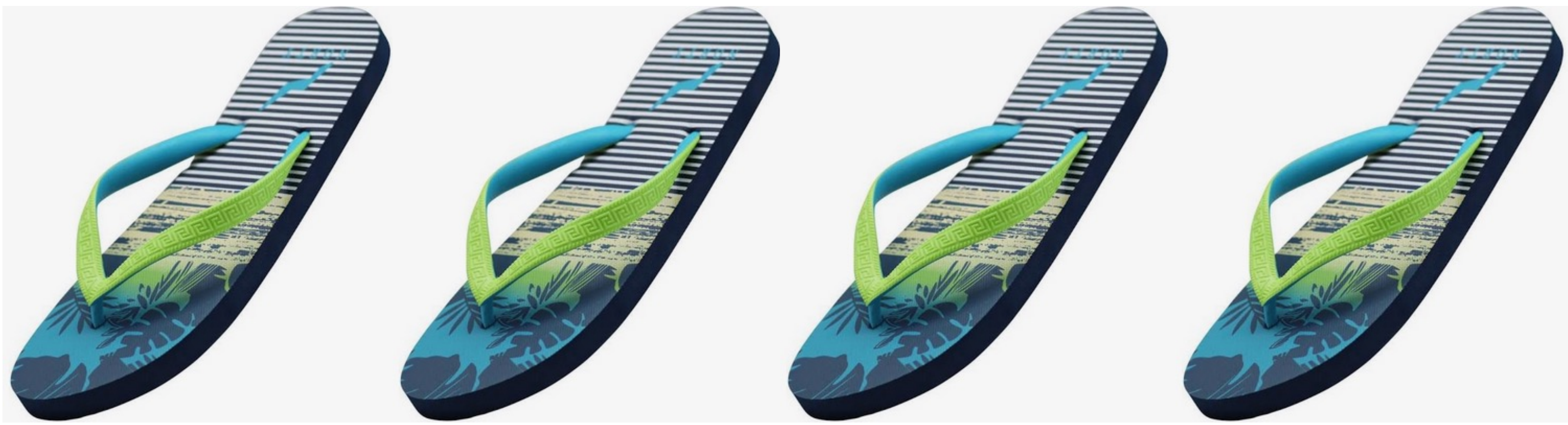
A simple shift register



You need 2 latches to make a flip-flop

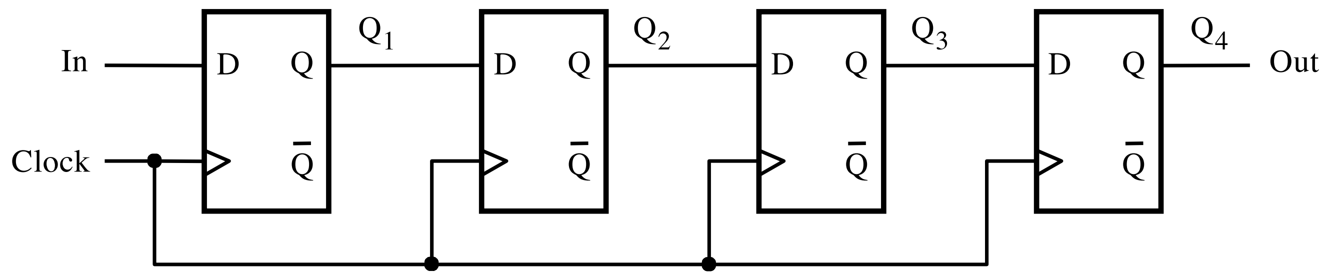
=

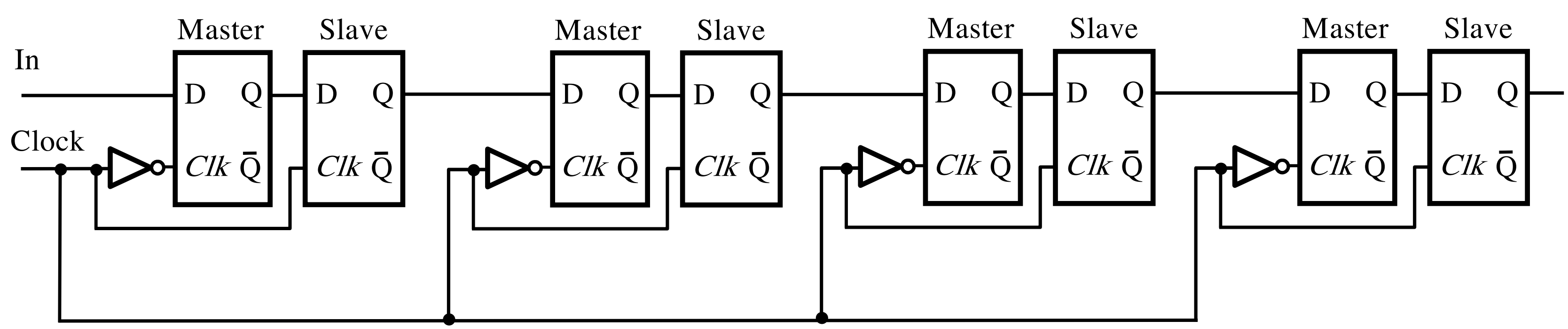
4-bit Register



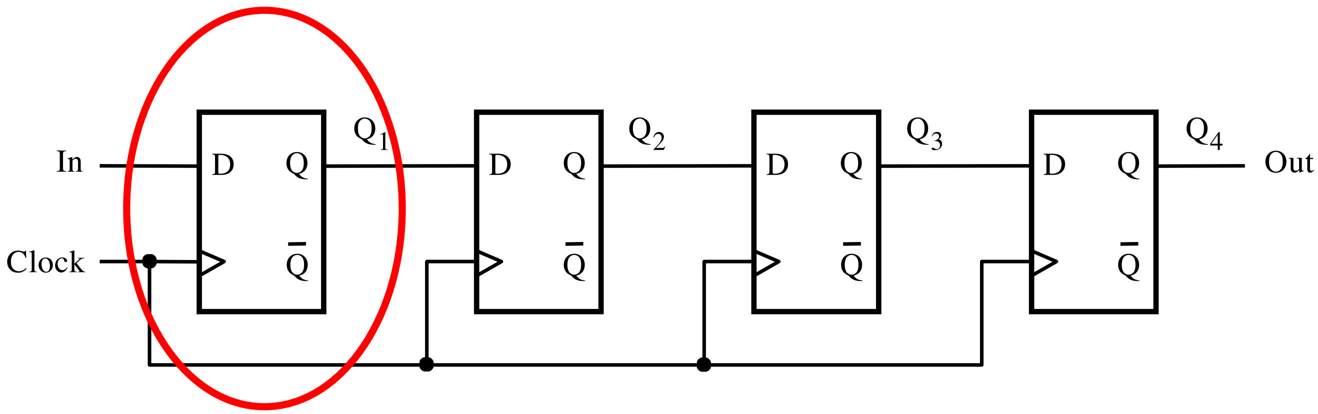
4-bit Register

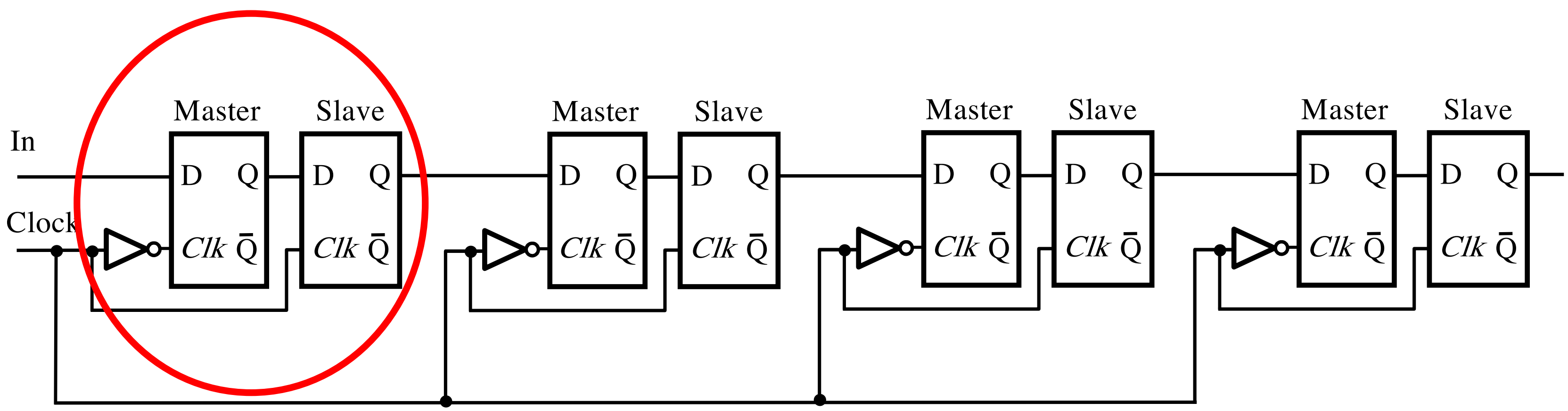
A simple shift register



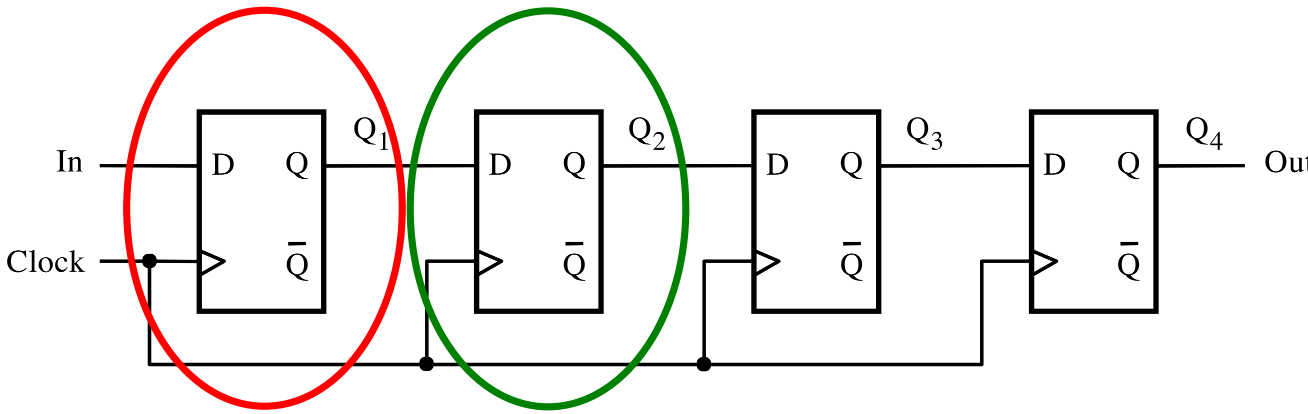


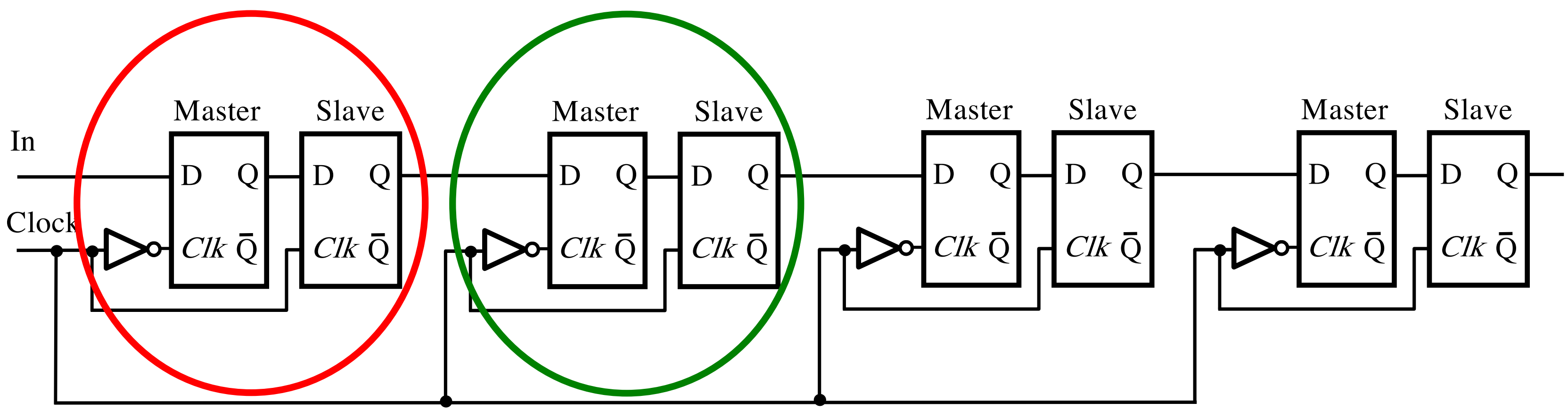
A simple shift register



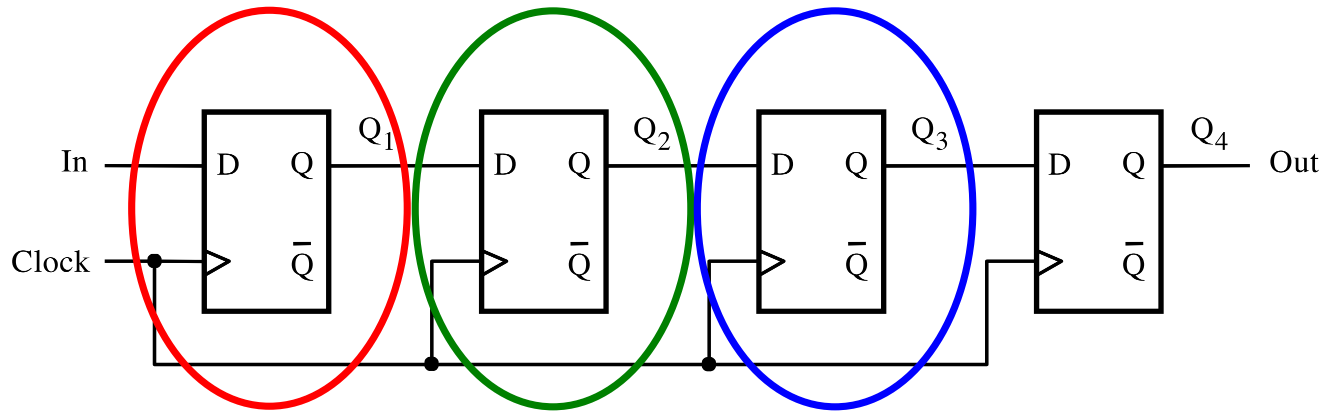


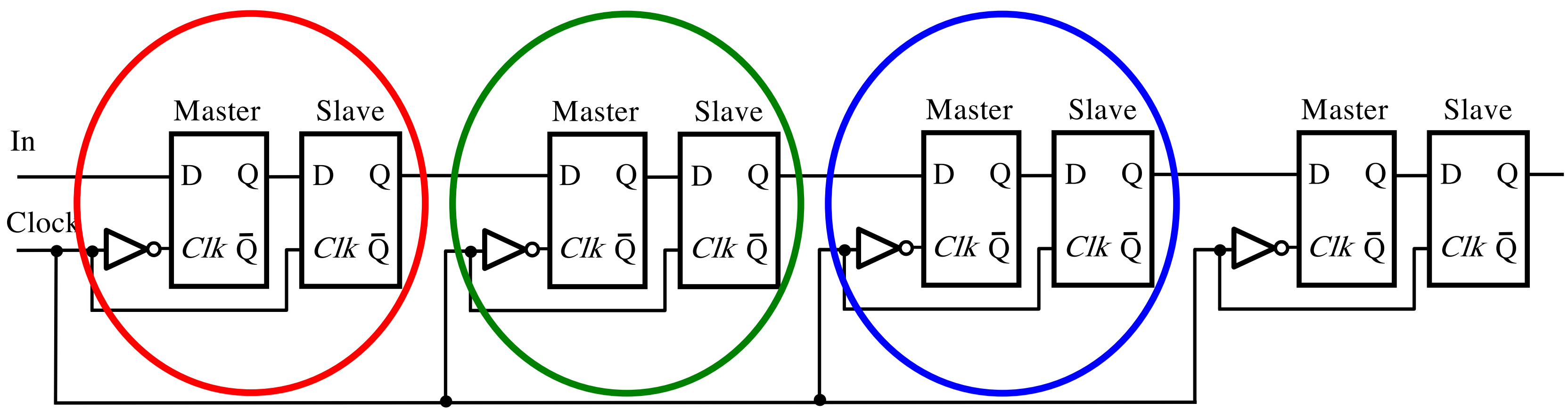
A simple shift register



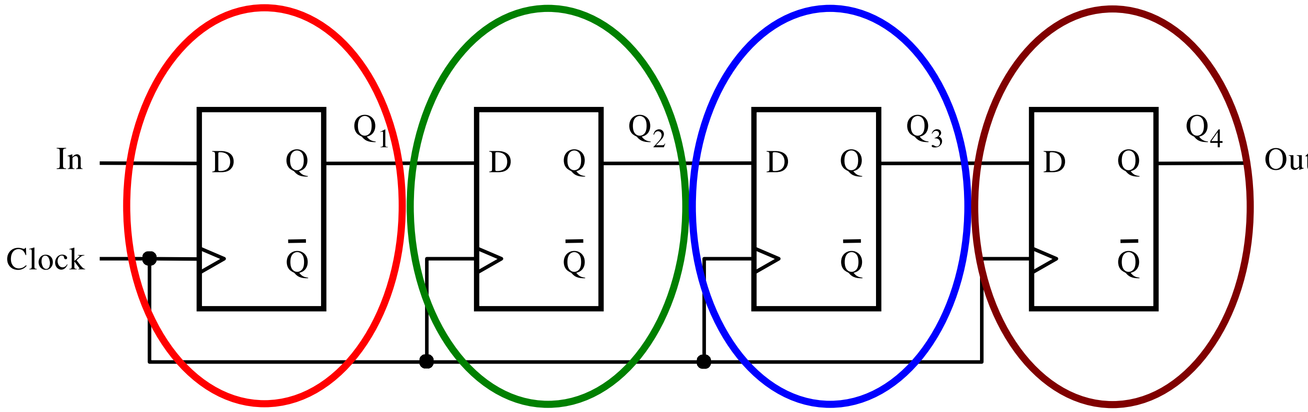


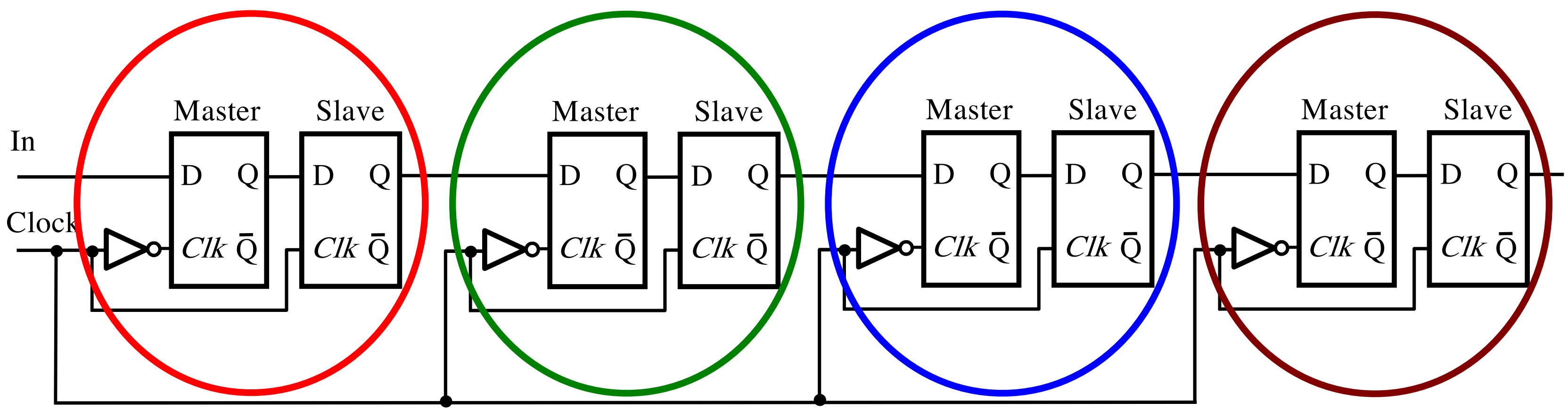
A simple shift register



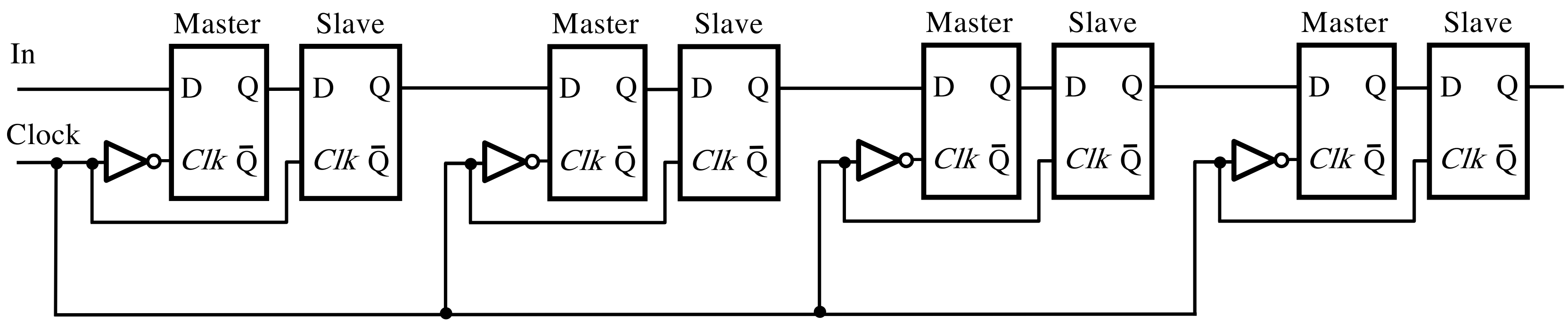


A simple shift register

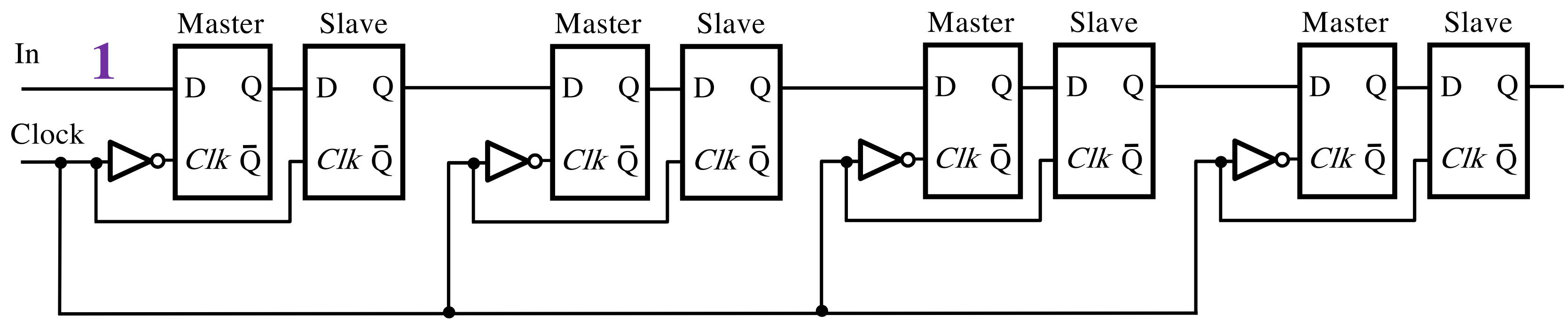




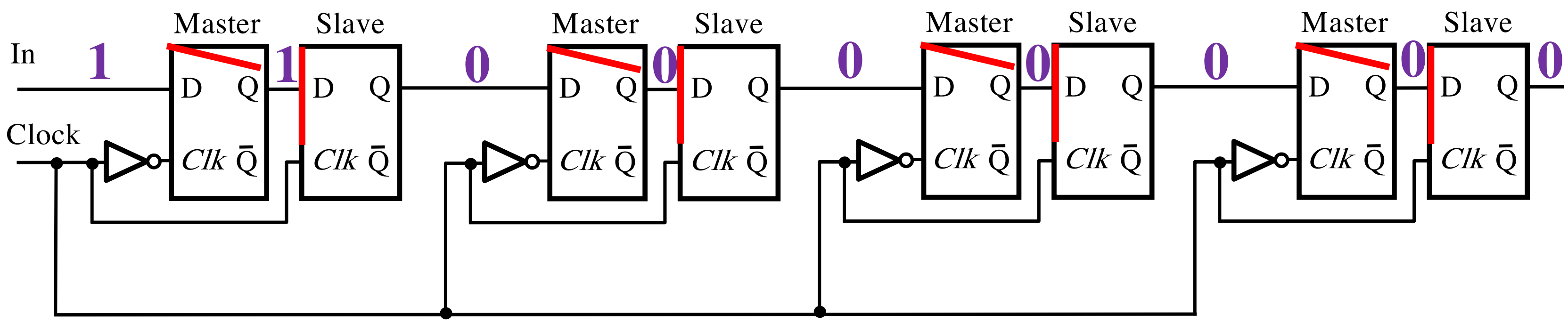
Simulating a shift register

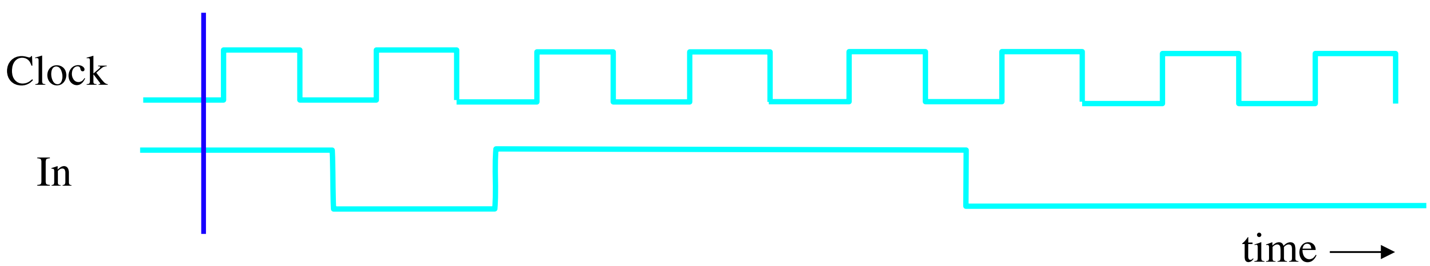


Simulating a shift register

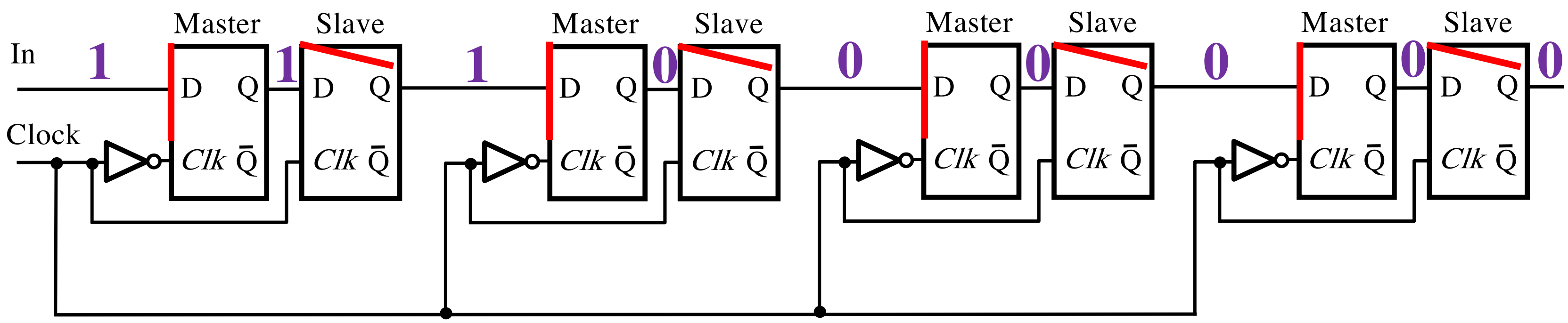


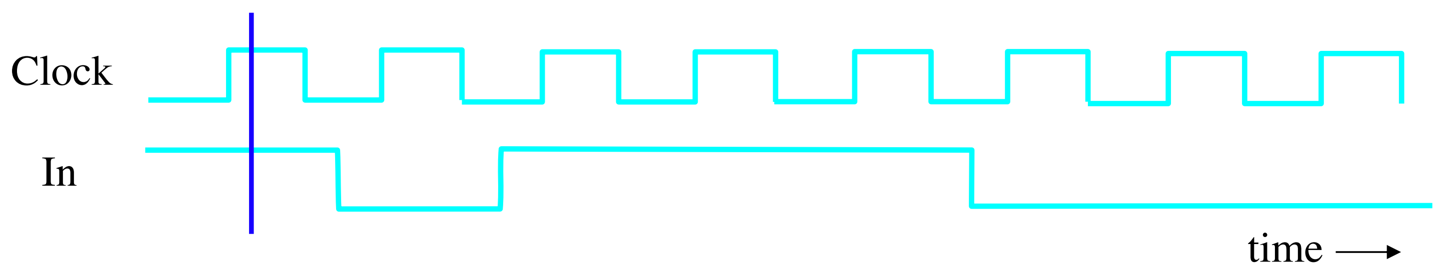
Simulating a shift register



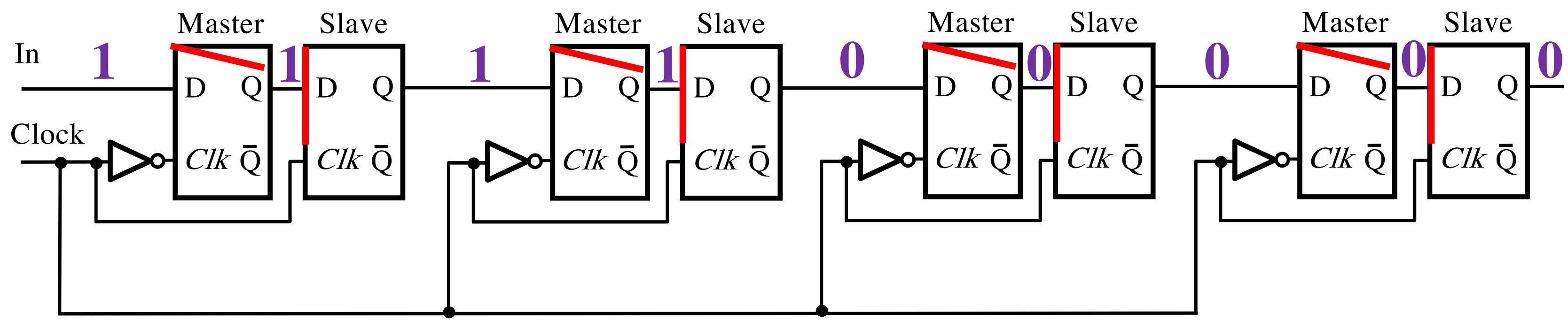


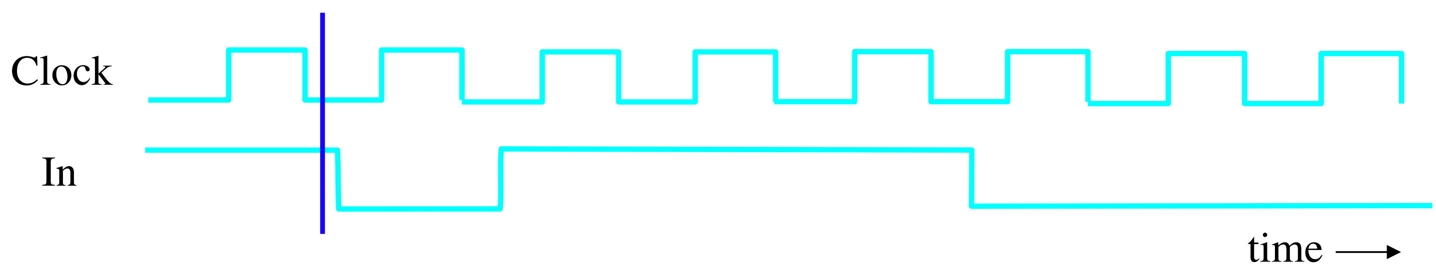
Simulating a shift register



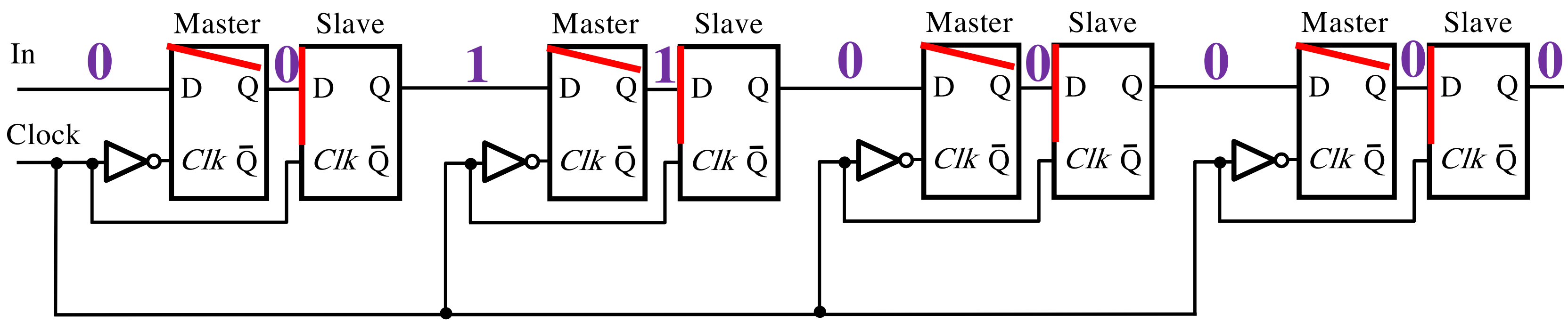


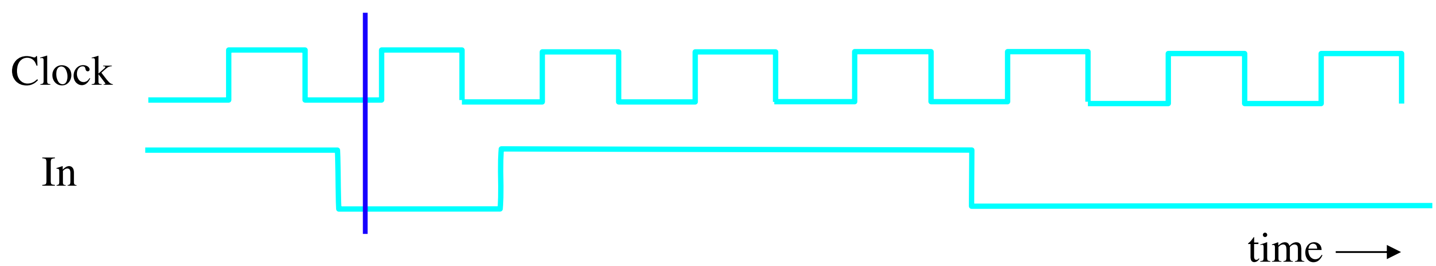
Simulating a shift register



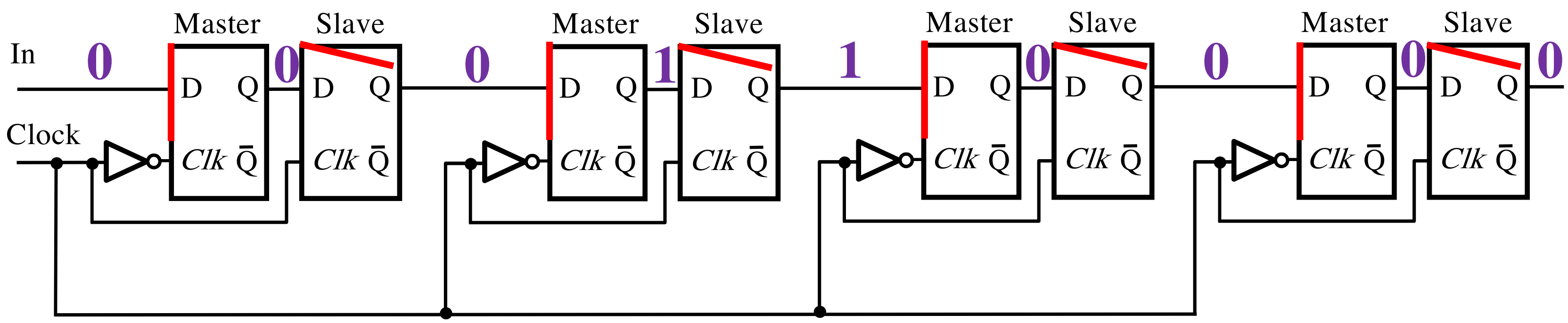


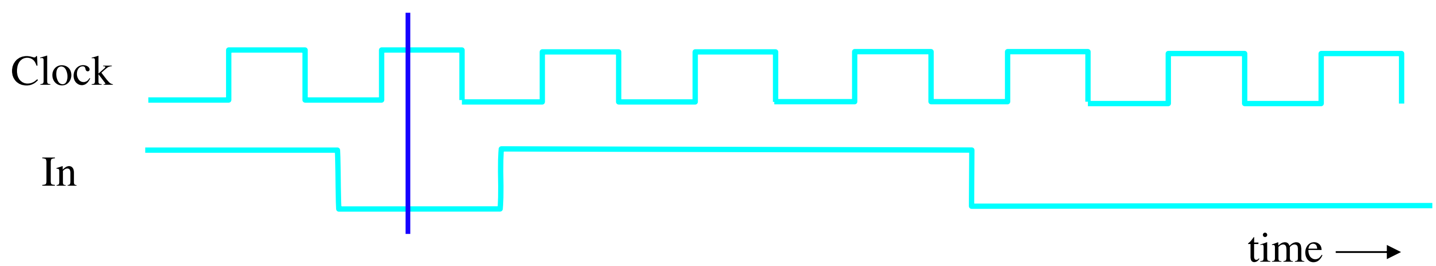
Simulating a shift register



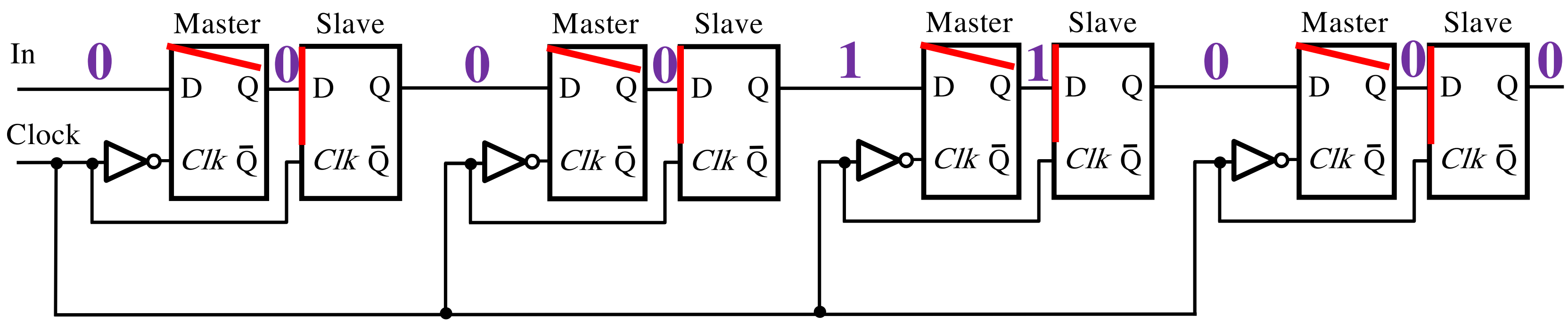


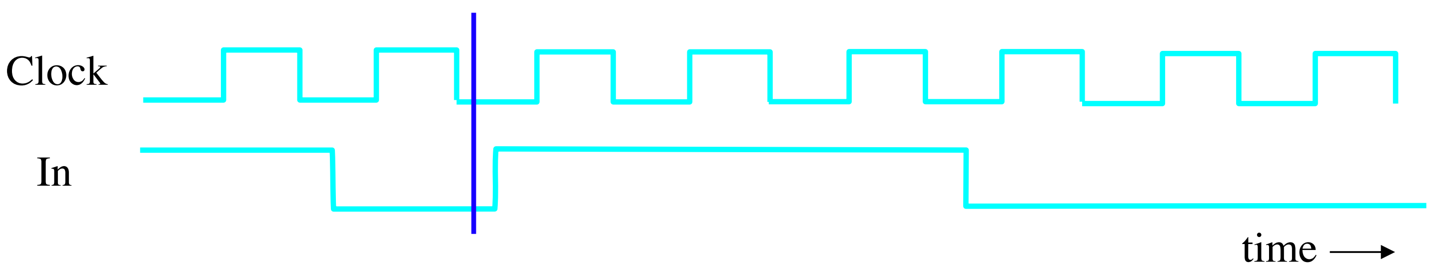
Simulating a shift register



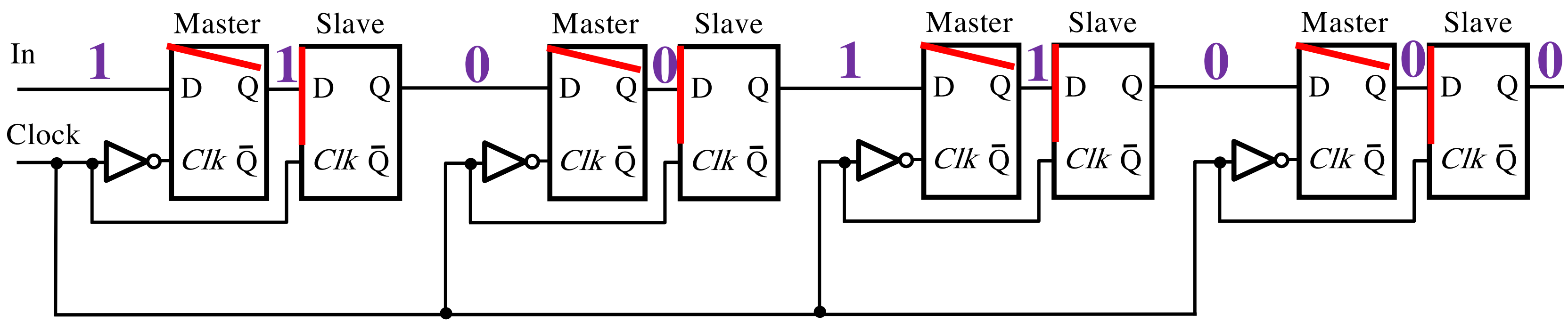


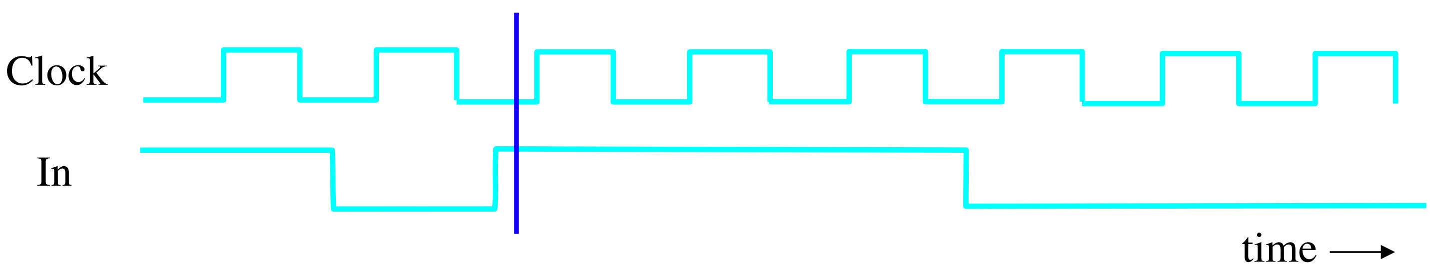
Simulating a shift register



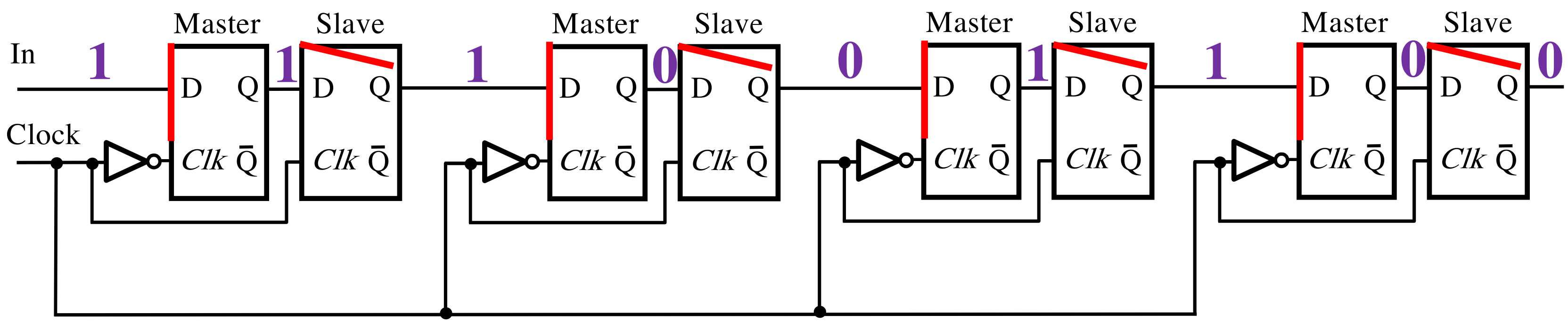


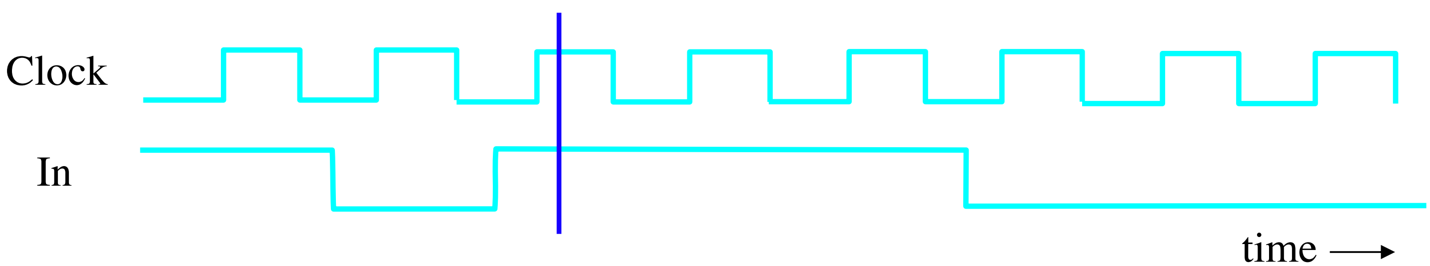
Simulating a shift register



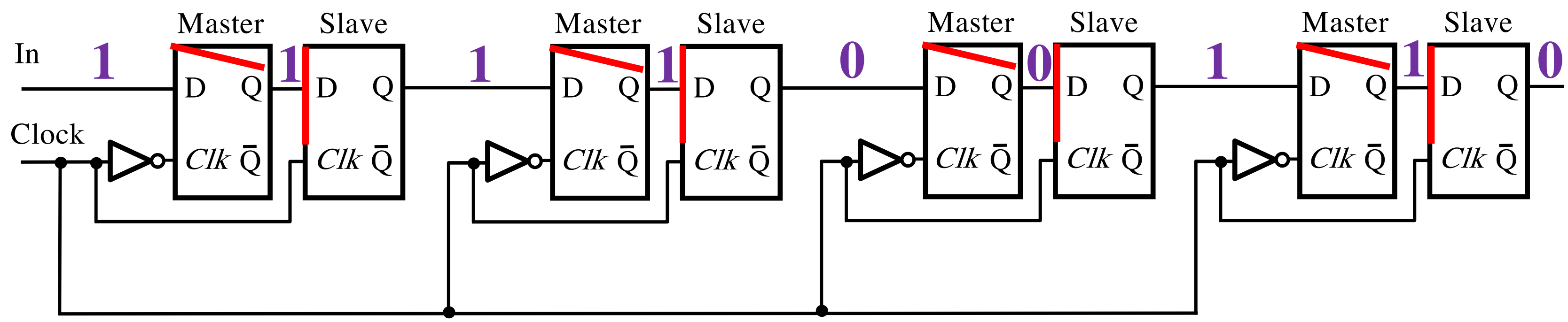


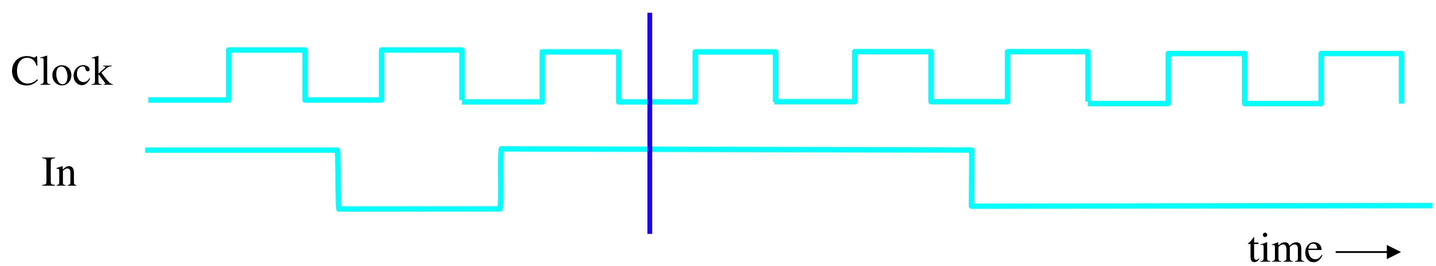
Simulating a shift register



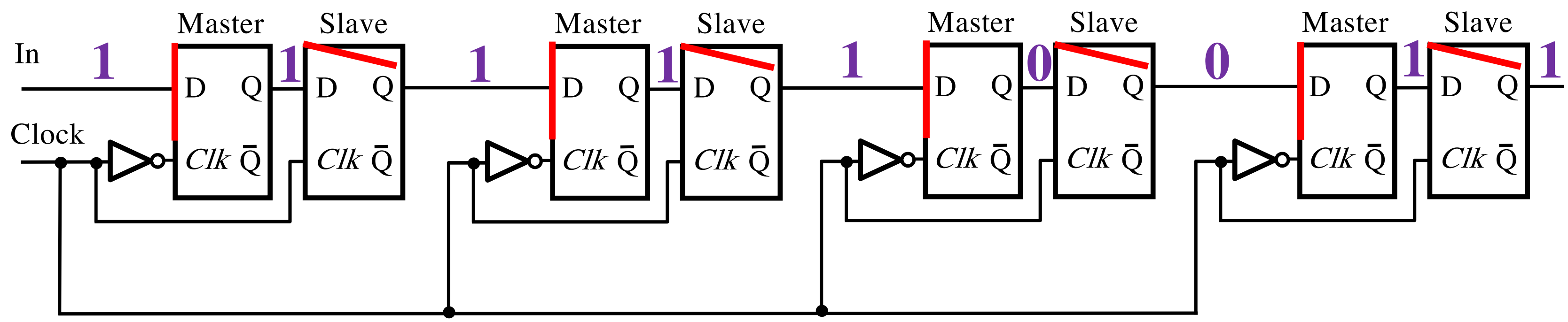


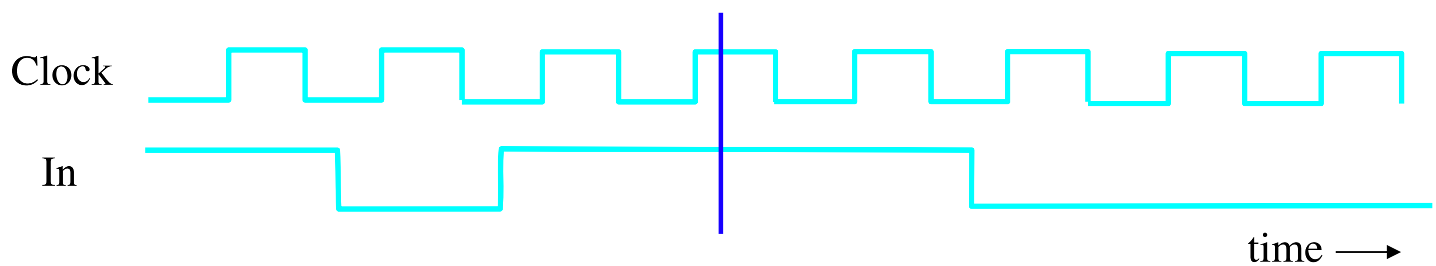
Simulating a shift register



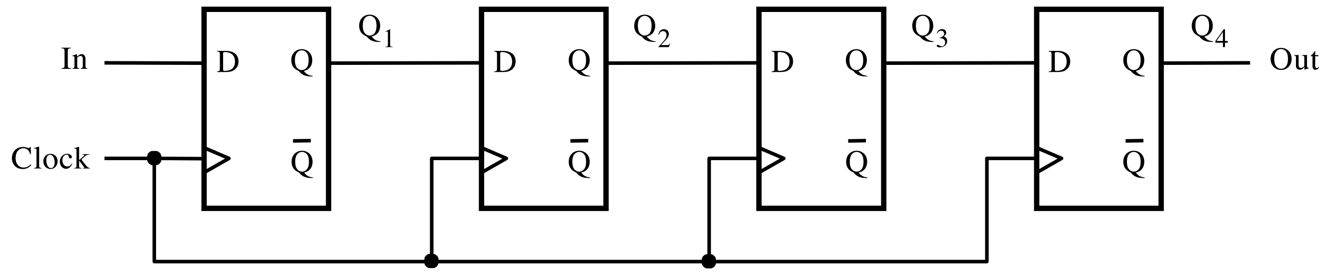


Simulating a shift register





A simple shift register



(a) Circuit

	In	Q_1	Q_2	Q_3	Q_4 = Out
t_0	1	0	0	0	0
t_1	0	1	0	0	0
t_2	1	0	1	0	0
t_3	1	1	0	1	0
t_4	1	1	1	0	1
t_5	0	1	1	1	0
t_6	0	0	1	1	1
t_7	0	0	0	1	1

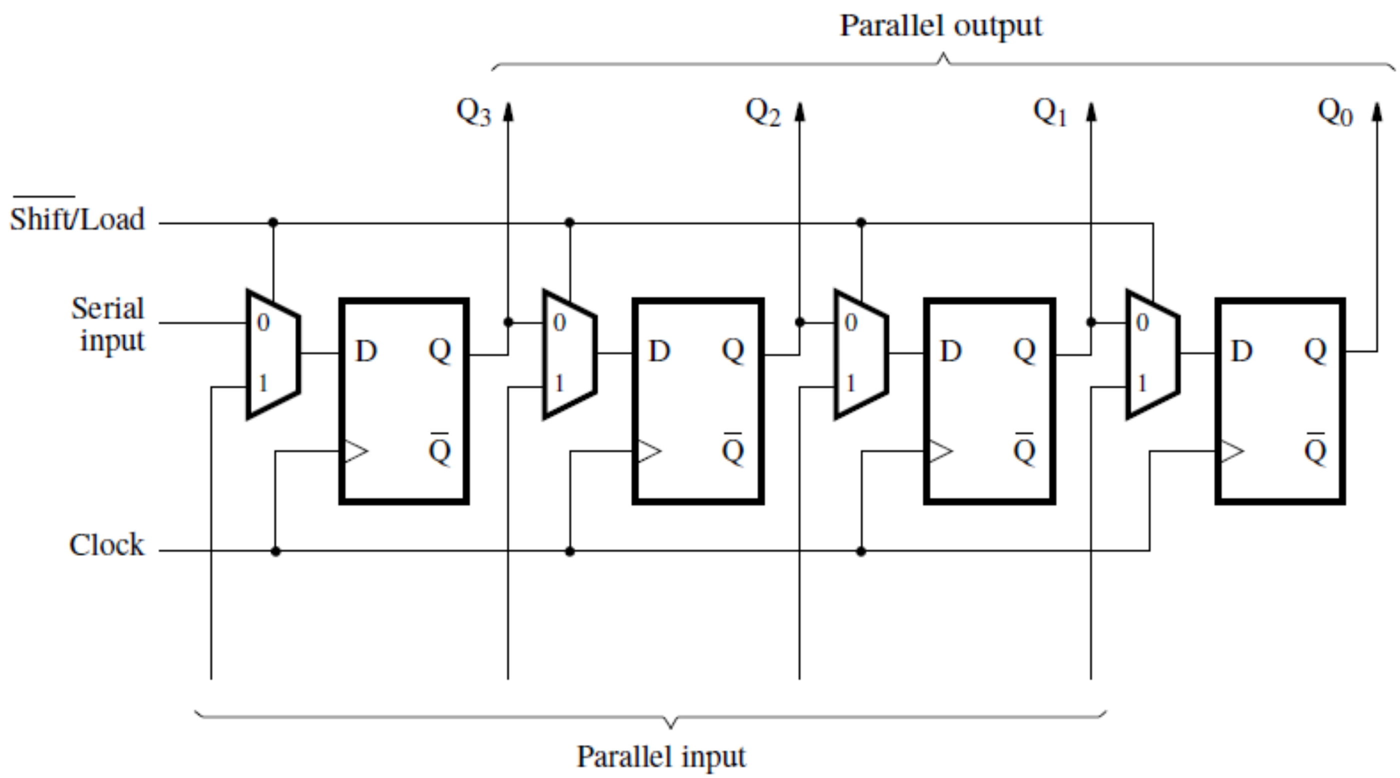
This simulation goes only up to here

(b) A sample sequence

[Figure 5.17 from the textbook]

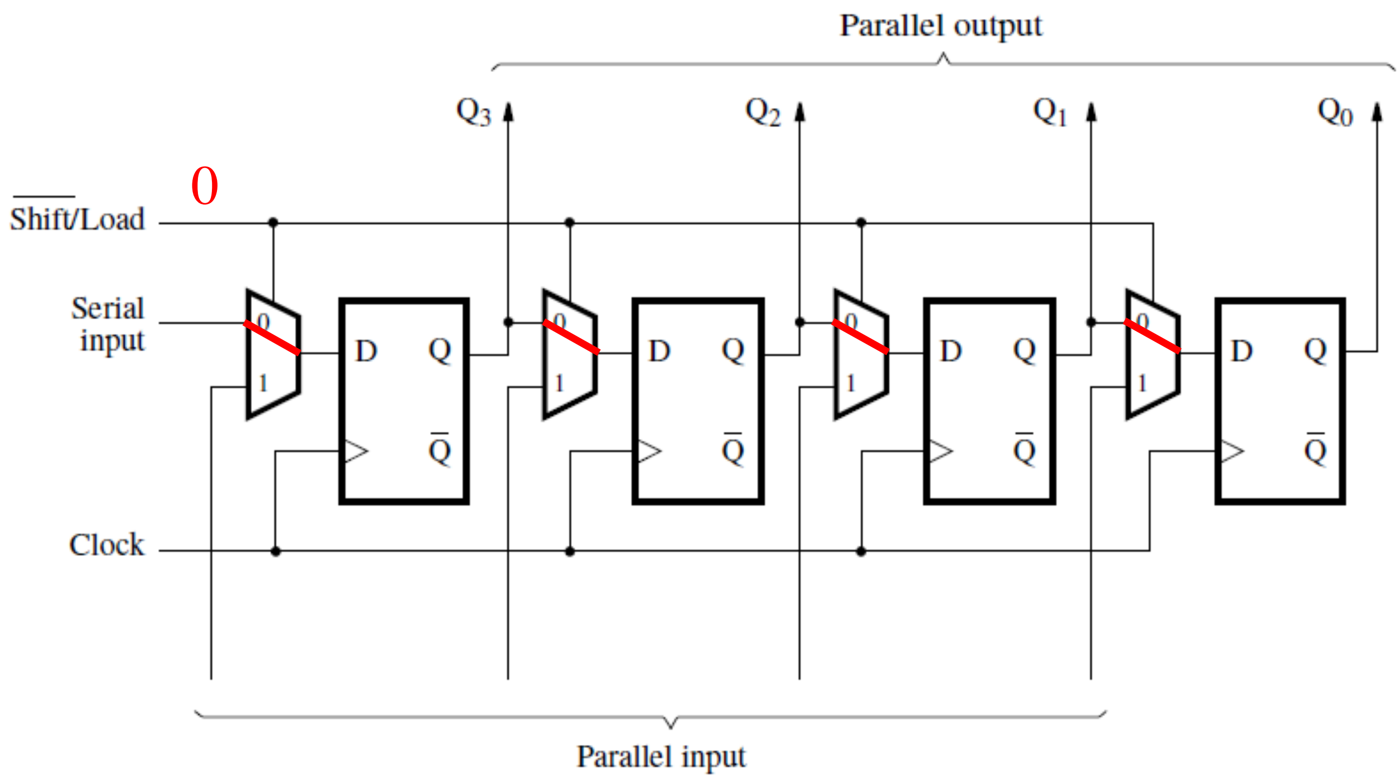
Parallel-Access Shift Register

Parallel-access shift register



[Figure 5.18 from the textbook]

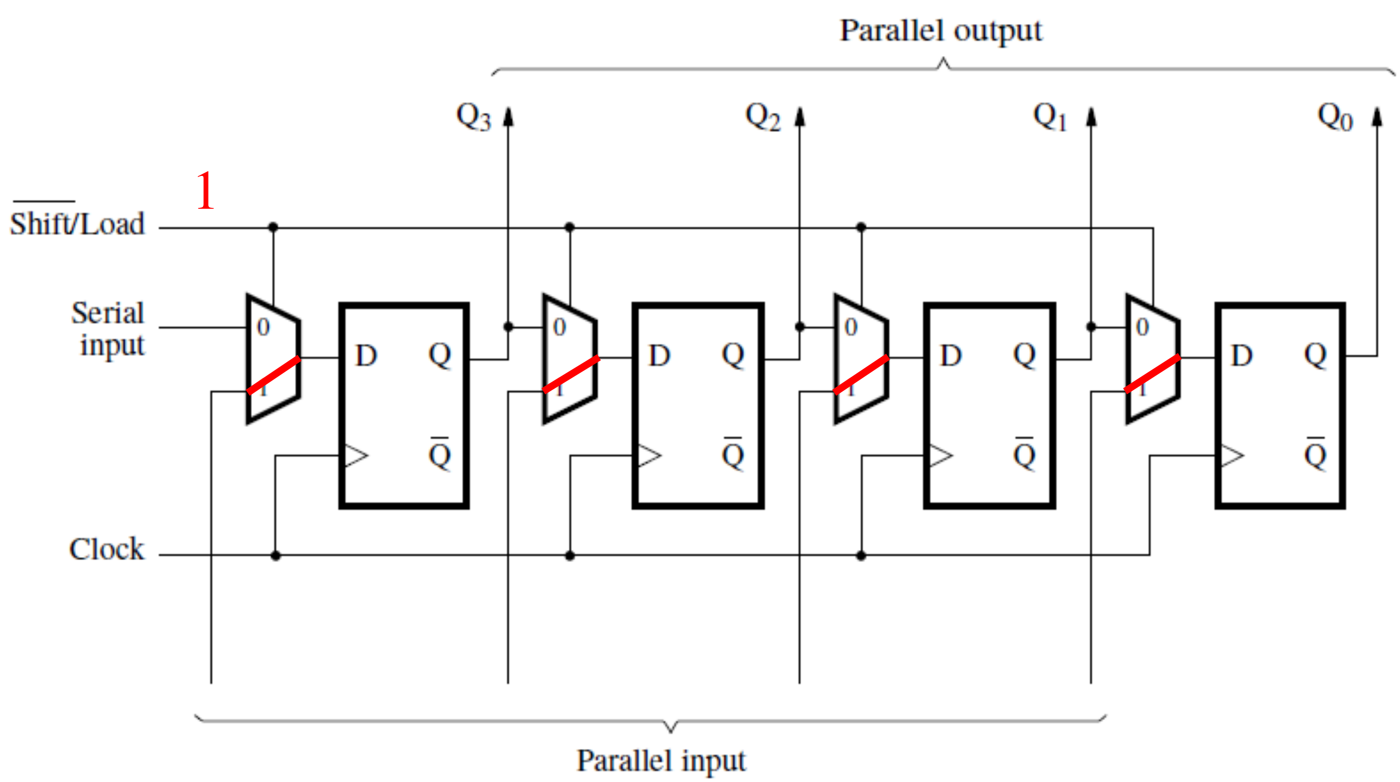
Parallel-access shift register



When Load=0, this behaves like a shift register.

[Figure 5.18 from the textbook]

Parallel-access shift register

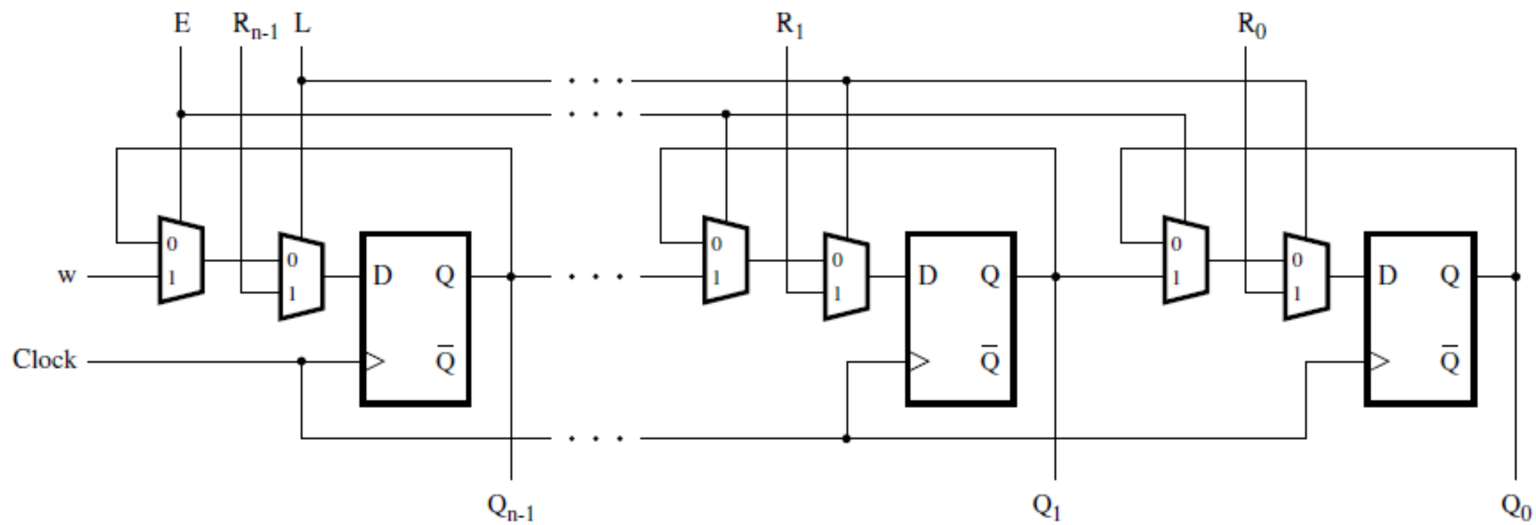


When Load=1, this behaves like a parallel-access register.

[Figure 5.18 from the textbook]

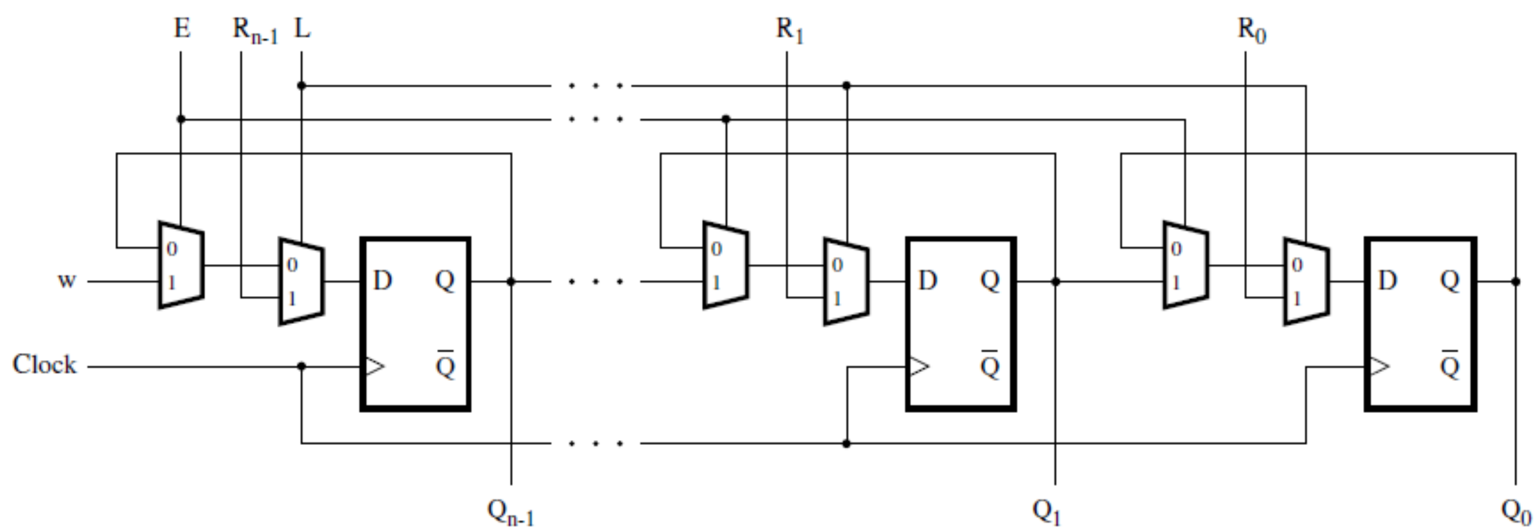
Shift Register With Parallel Load and Enable

A shift register with parallel load and enable control inputs



[Figure 5.59 from the textbook]

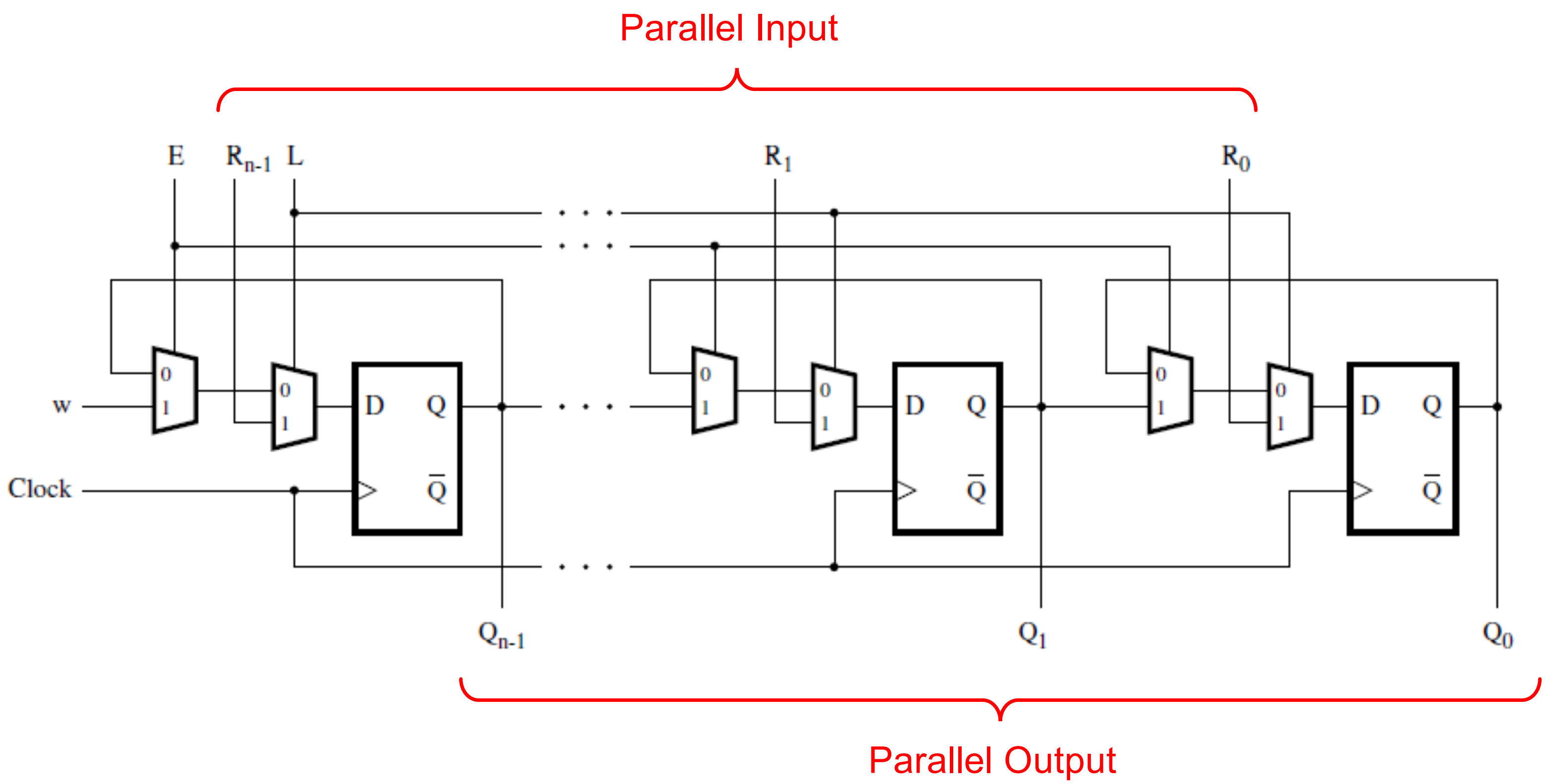
A shift register with parallel load and enable control inputs



The directions of the input and output lines are switched relative to the previous slides.

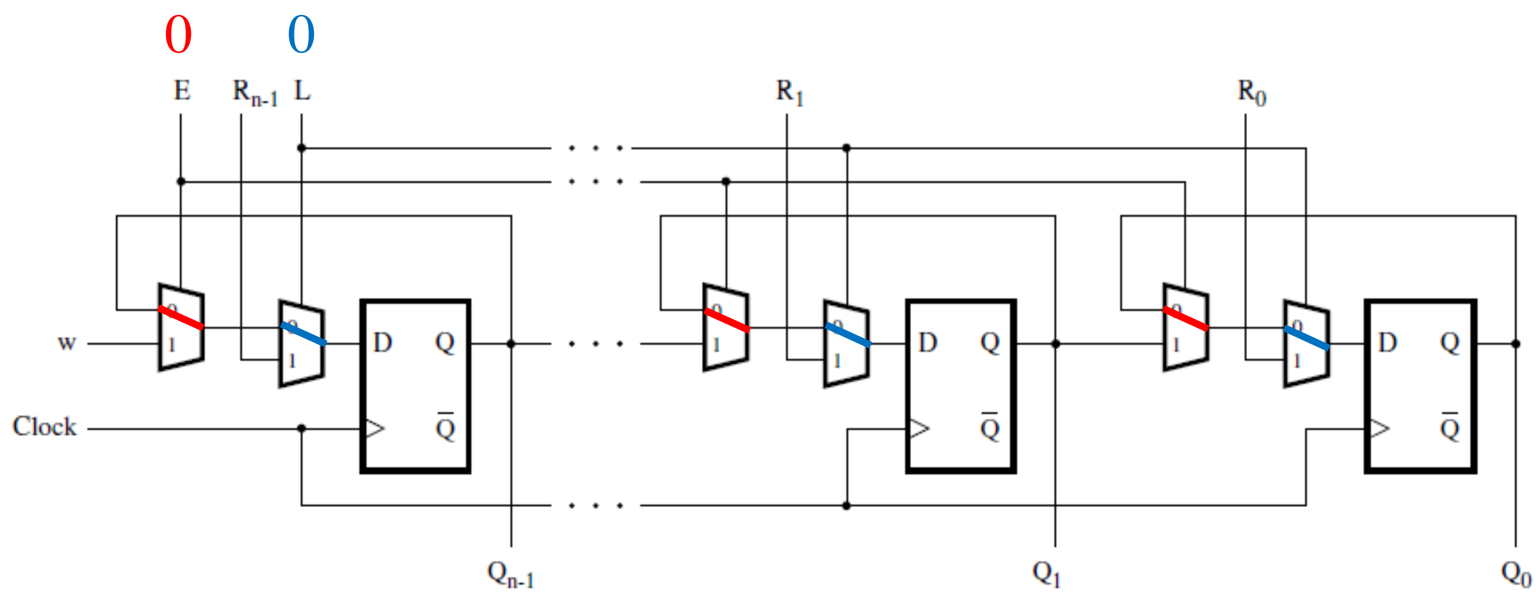
[Figure 5.59 from the textbook]

A shift register with parallel load and enable control inputs



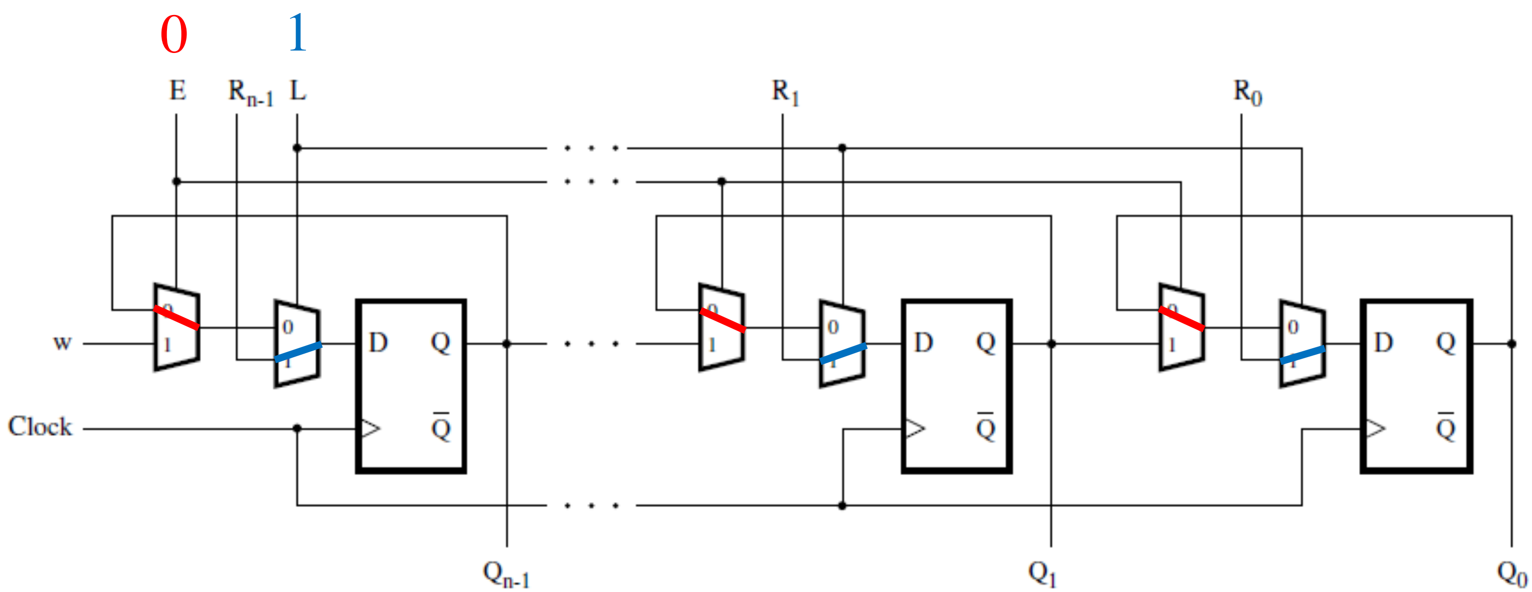
[Figure 5.59 from the textbook]

A shift register with parallel load and enable control inputs



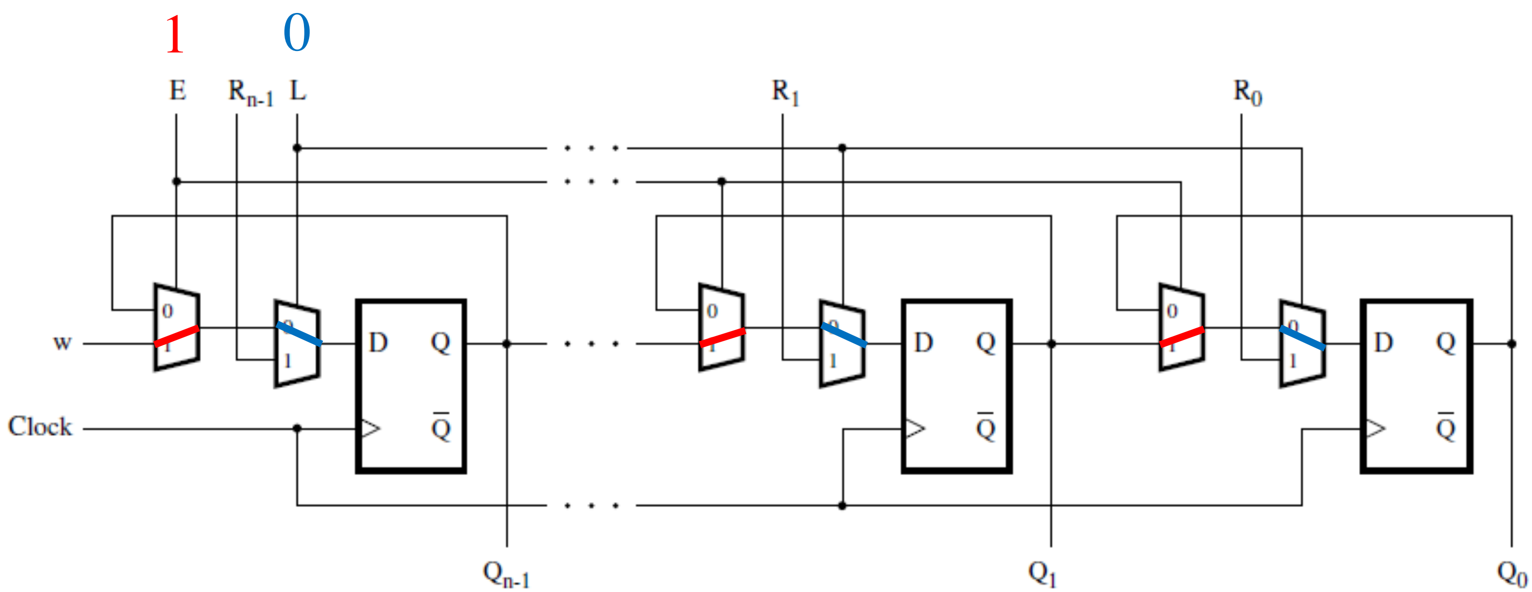
[Figure 5.59 from the textbook]

A shift register with parallel load and enable control inputs



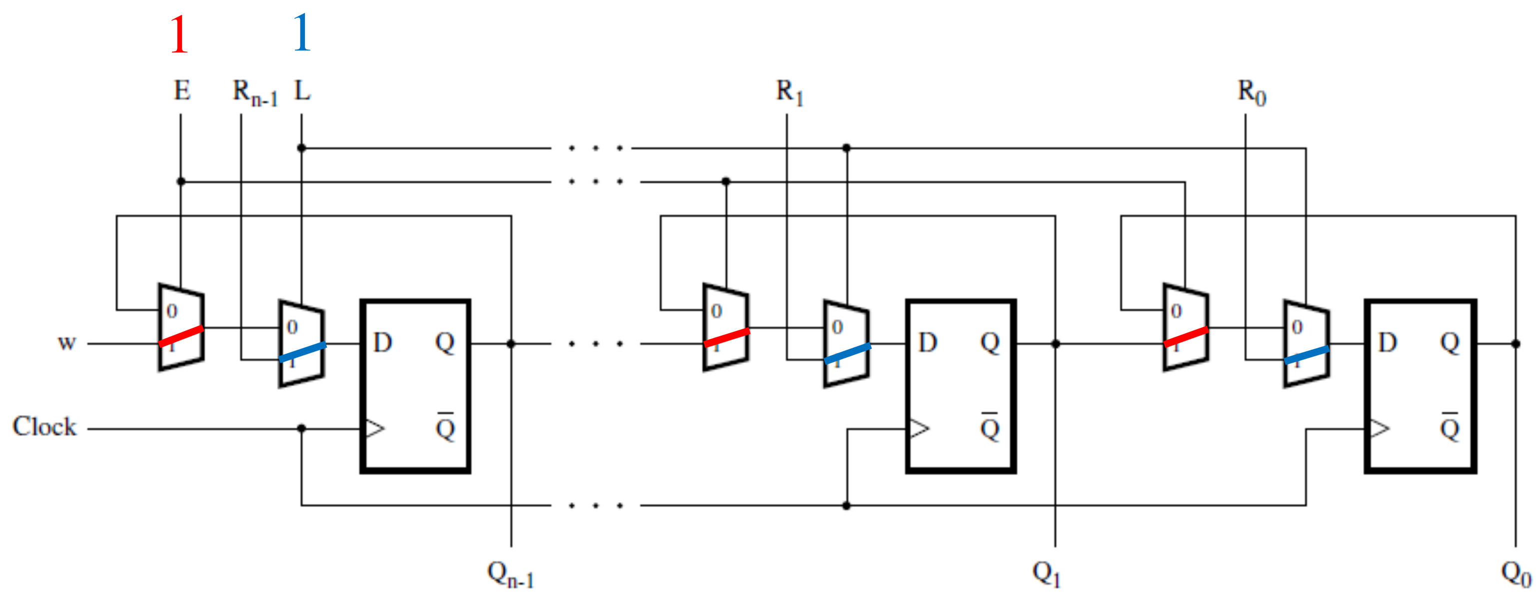
[Figure 5.59 from the textbook]

A shift register with parallel load and enable control inputs



[Figure 5.59 from the textbook]

A shift register with parallel load and enable control inputs



[Figure 5.59 from the textbook]

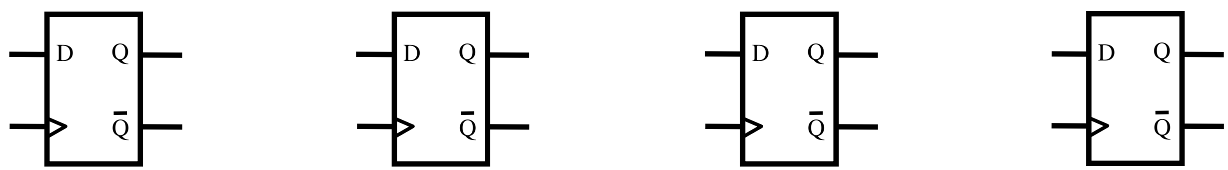
Parallel-access
shift left / right register

Parallel-access shift left/right register

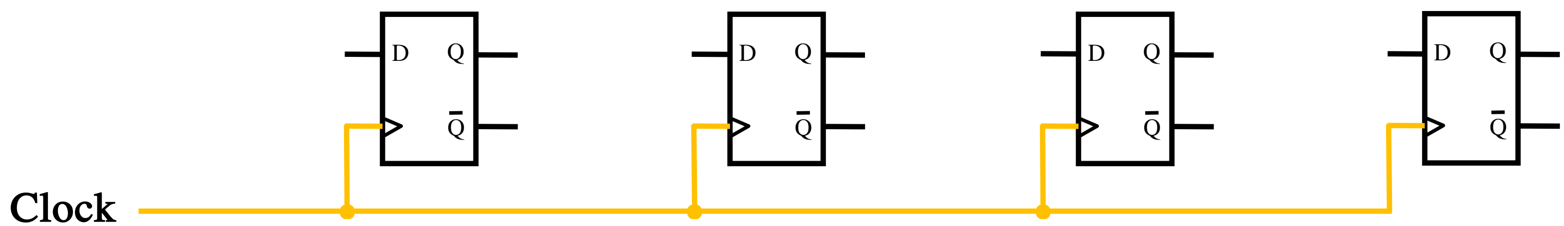
Complete the following circuit diagram to implement a 4-bit register that has both parallel load and shift left/right functionality. The register has two control inputs (C1 and C0), four parallel input lines (I3, I2, I1, and I0), and four output lines (Q3, Q2, Q1, and Q0). Depending on the values of C1 and C0, the register performs one of the following four operations:

C_1	C_0	Operation
0	0	Hold the current value (i.e., Q_3 Q_2 Q_1 Q_0 are not changed)
0	1	Shift left (i.e., new $Q_3=Q_2$, new $Q_2=Q_1$, new $Q_1=Q_0$, new $Q_0=I_0$)
1	0	Shift right (i.e., new $Q_3=I_3$, new $Q_2=Q_3$, new $Q_1=Q_2$, new $Q_0=Q_1$)
1	1	Load new data (i.e., new $Q_3=I_3$, new $Q_2=I_2$, new $Q_1=I_1$, new $Q_0=I_0$)

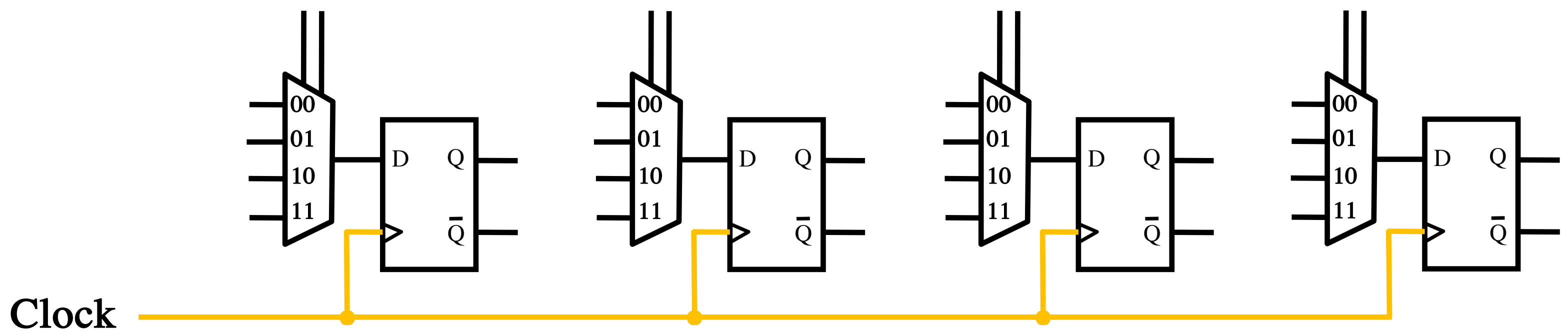
Parallel-access shift left/right register



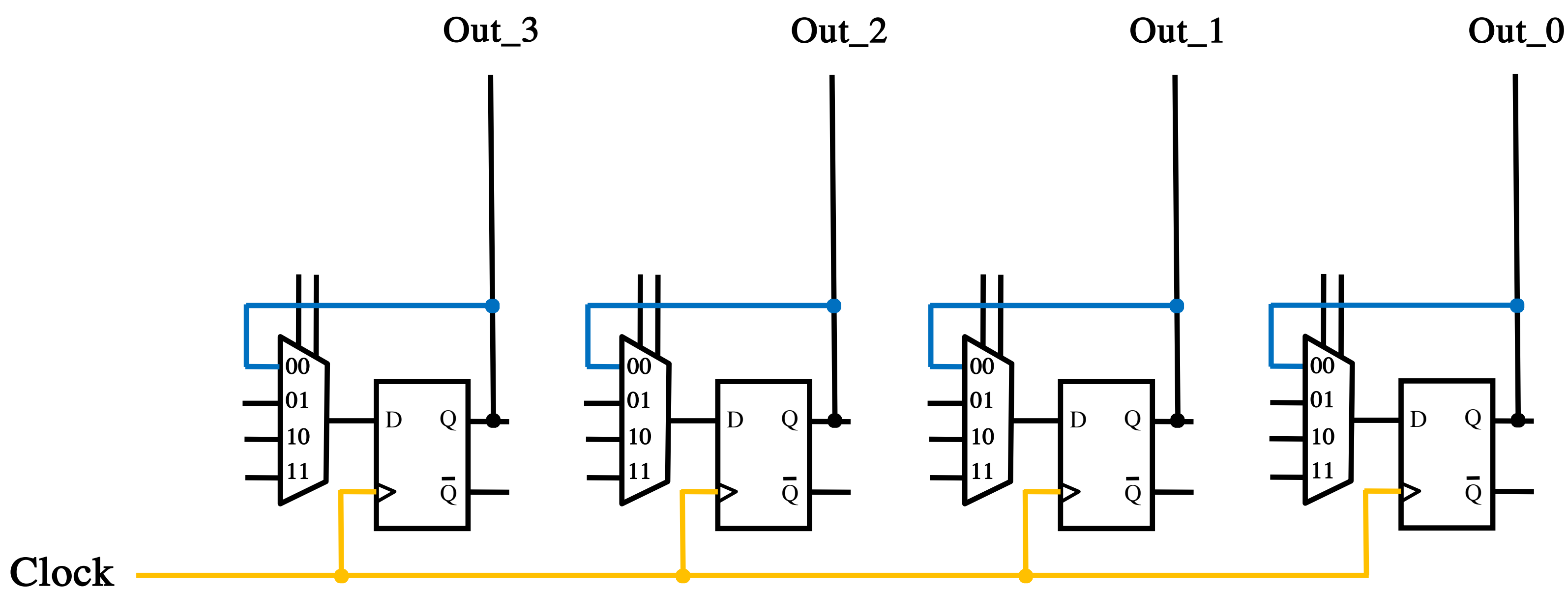
Parallel-access shift left/right register



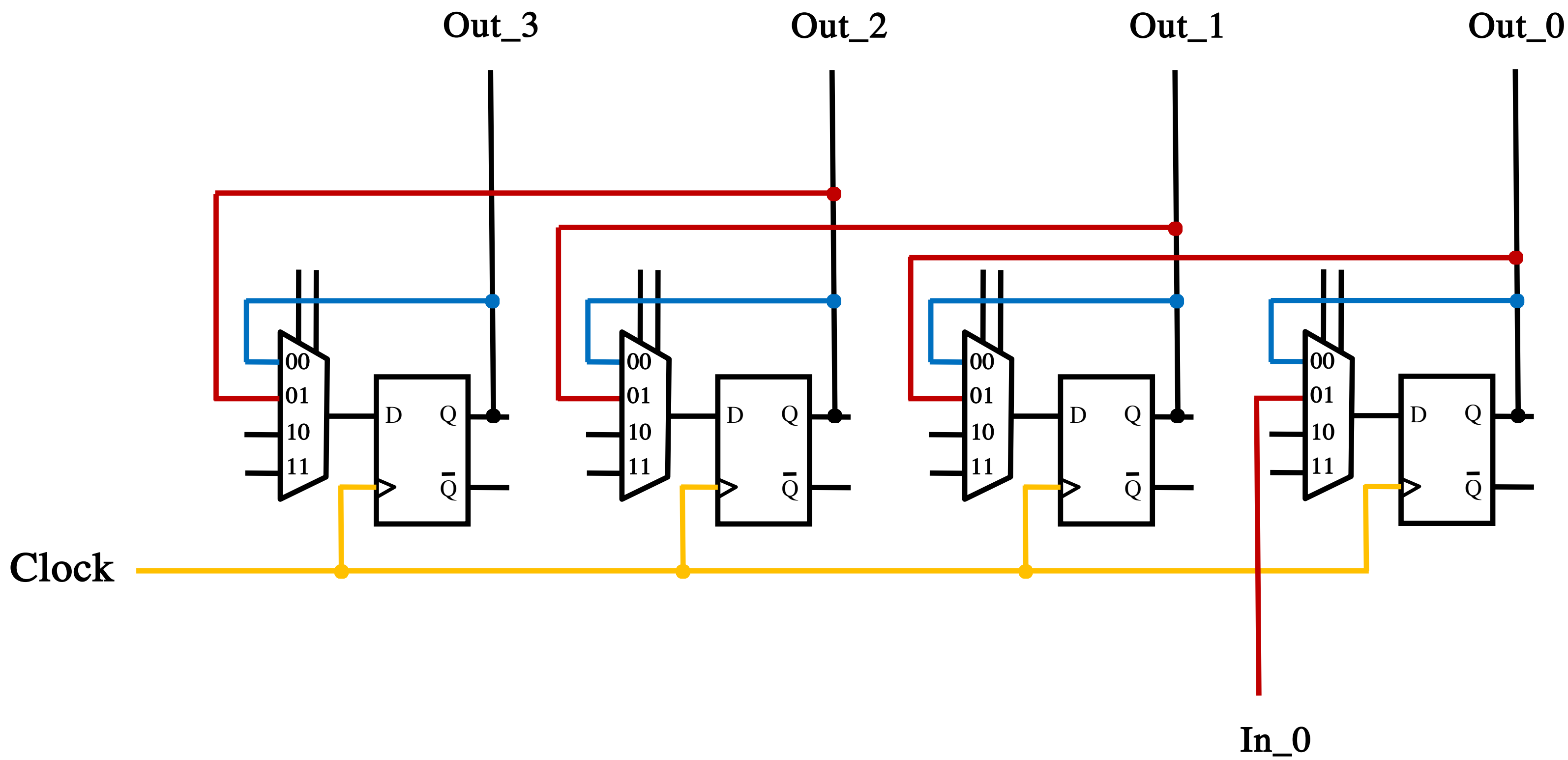
Parallel-access shift left/right register



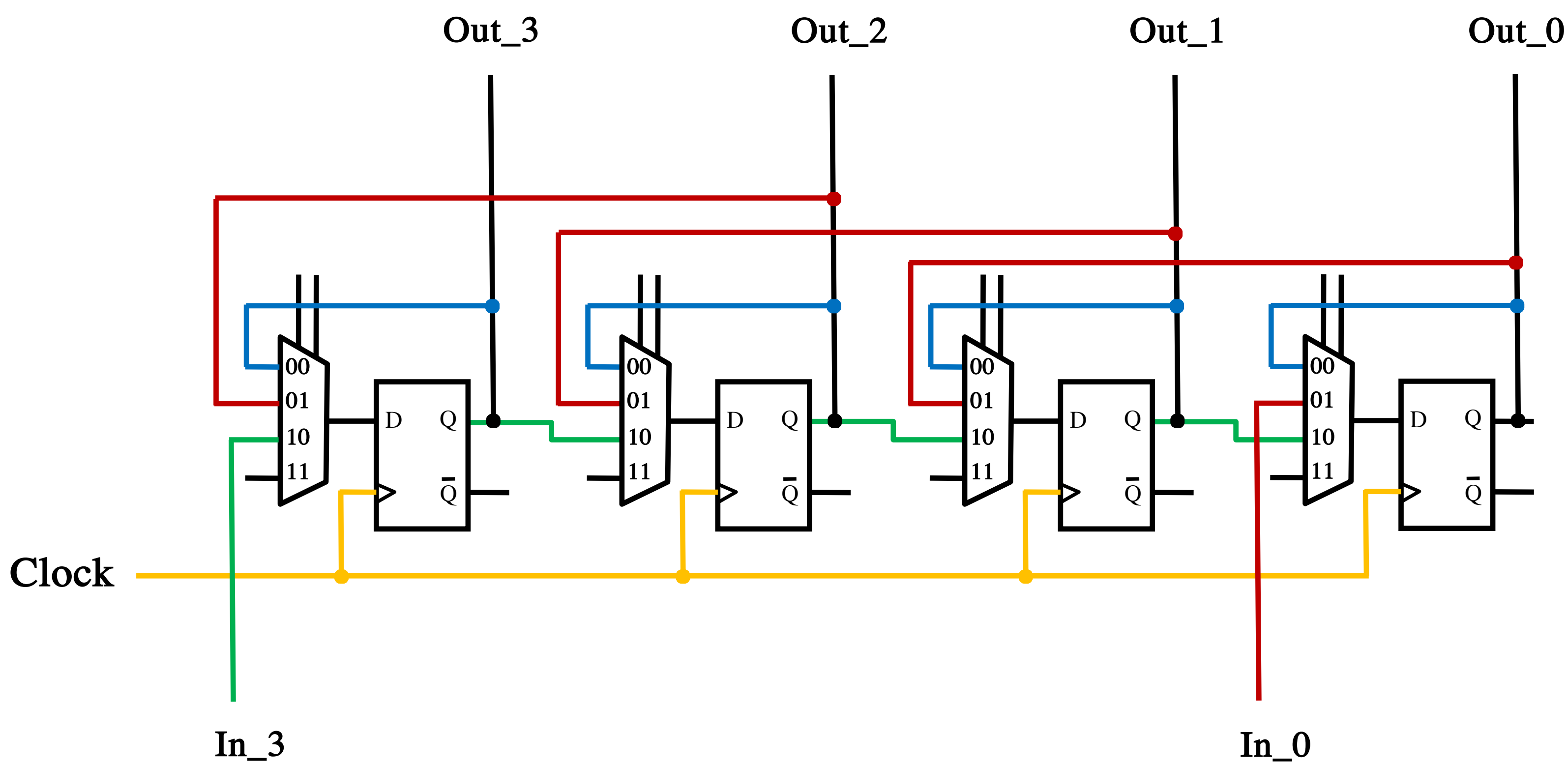
Parallel-access shift left/right register



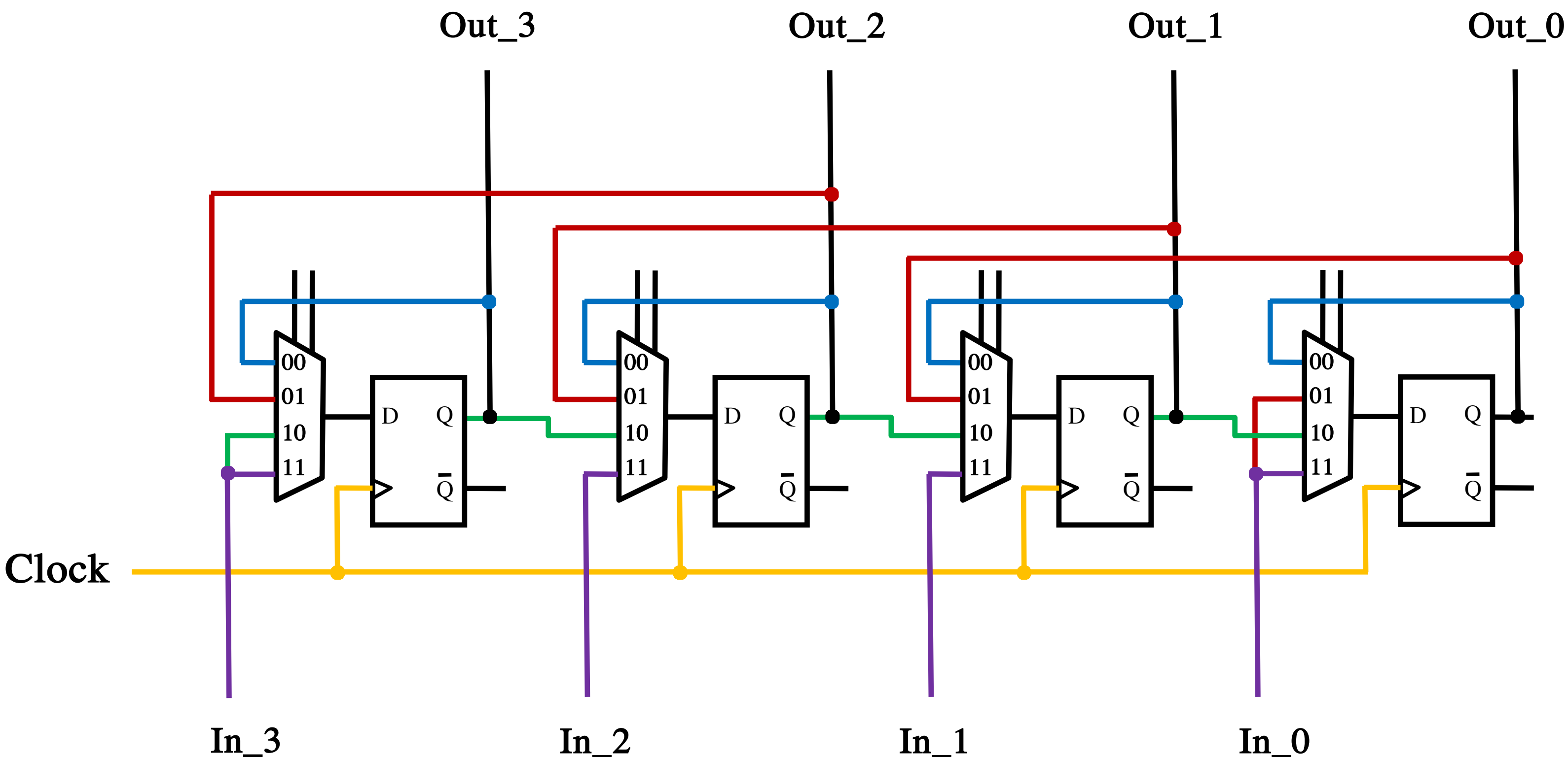
Parallel-access shift left/right register



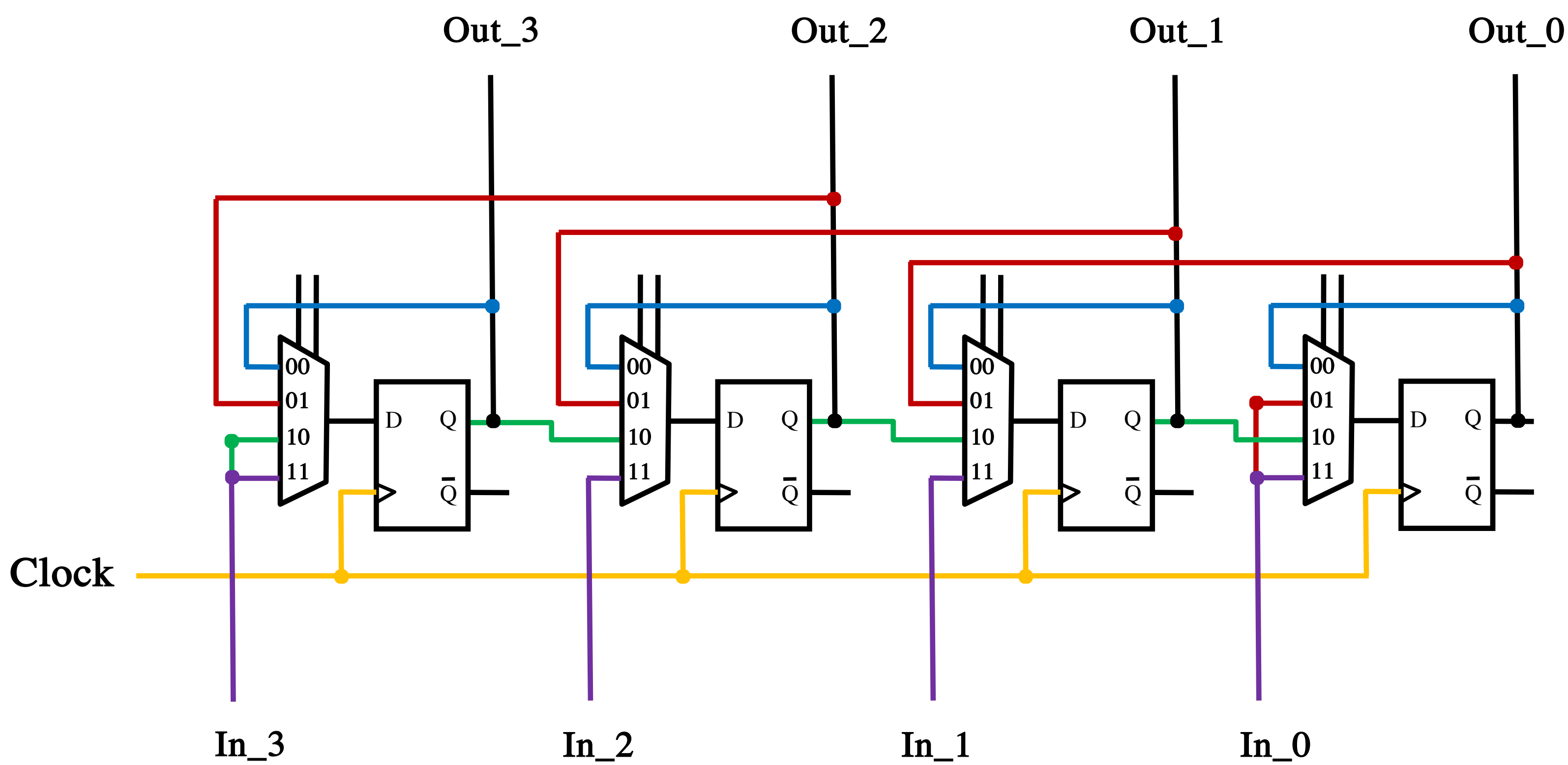
Parallel-access shift left/right register



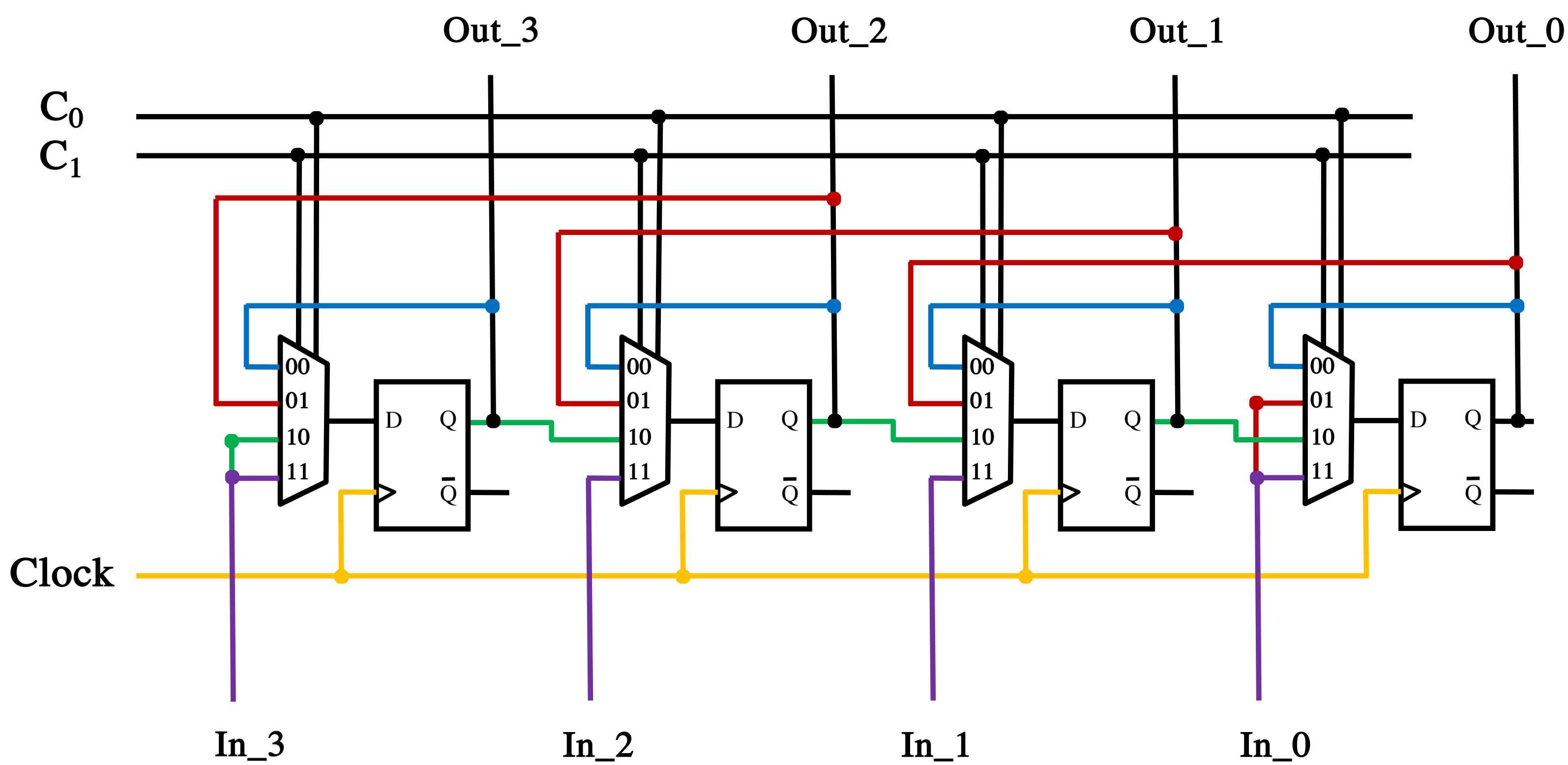
Parallel-access shift left/right register



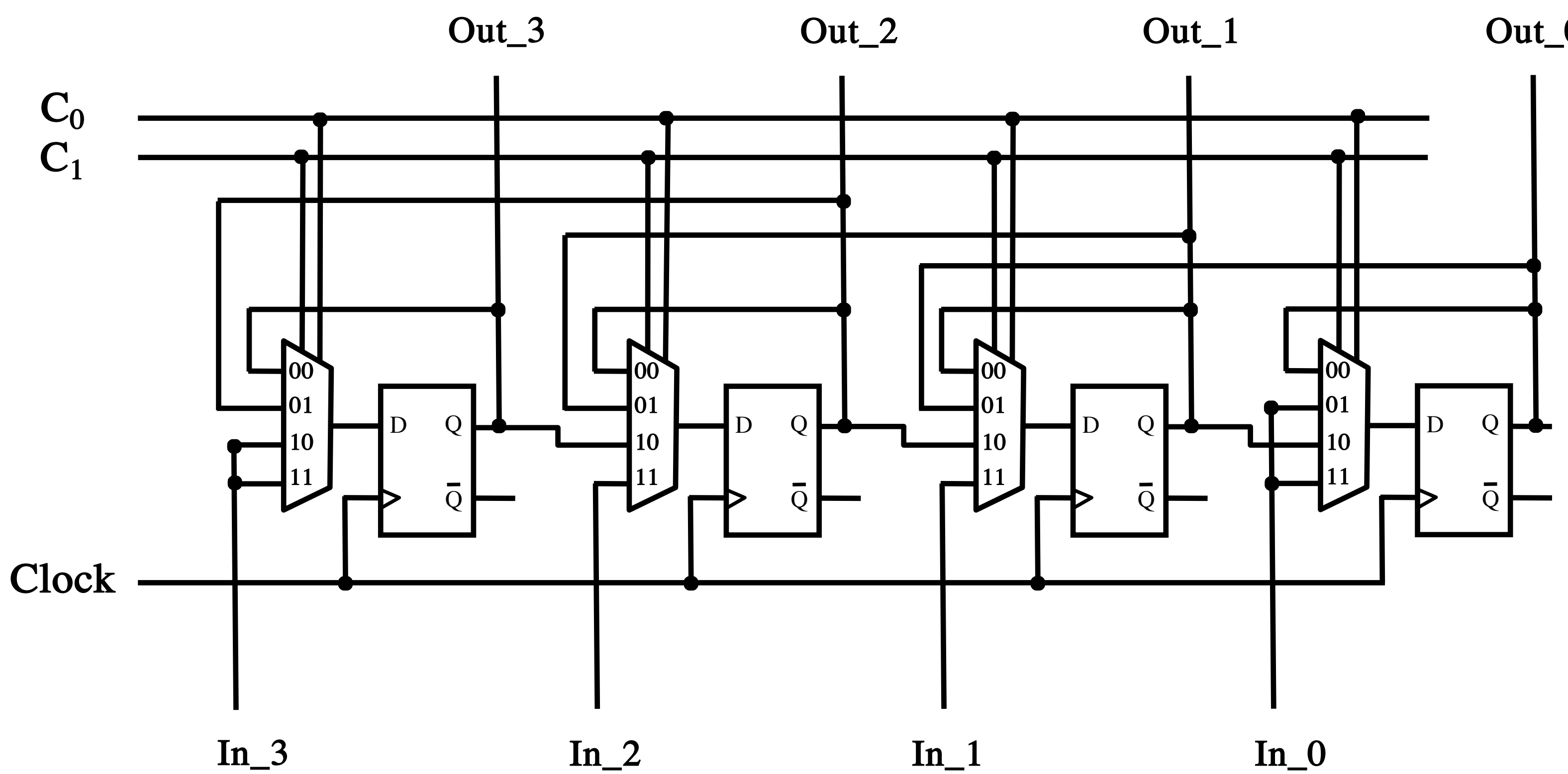
Parallel-access shift left/right register



Parallel-access shift left/right register



Parallel-access shift left/right register



Questions?

THE END