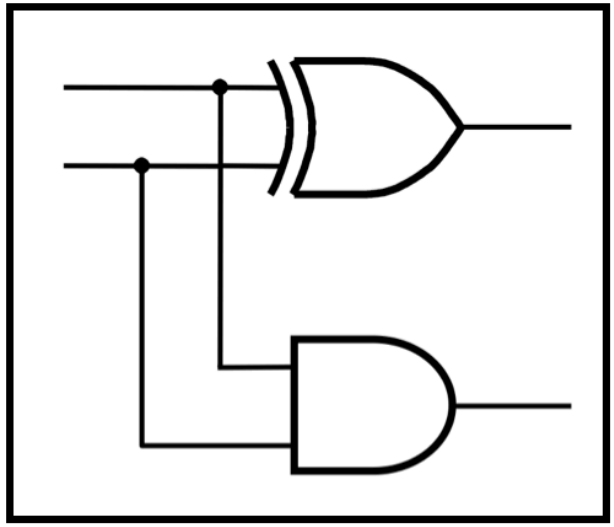


CprE 281: Digital Logic

Instructor: Alexander Stoytchev

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

Registers and Counters

CprE 281: Digital Logic
Iowa State University, Ames, IA

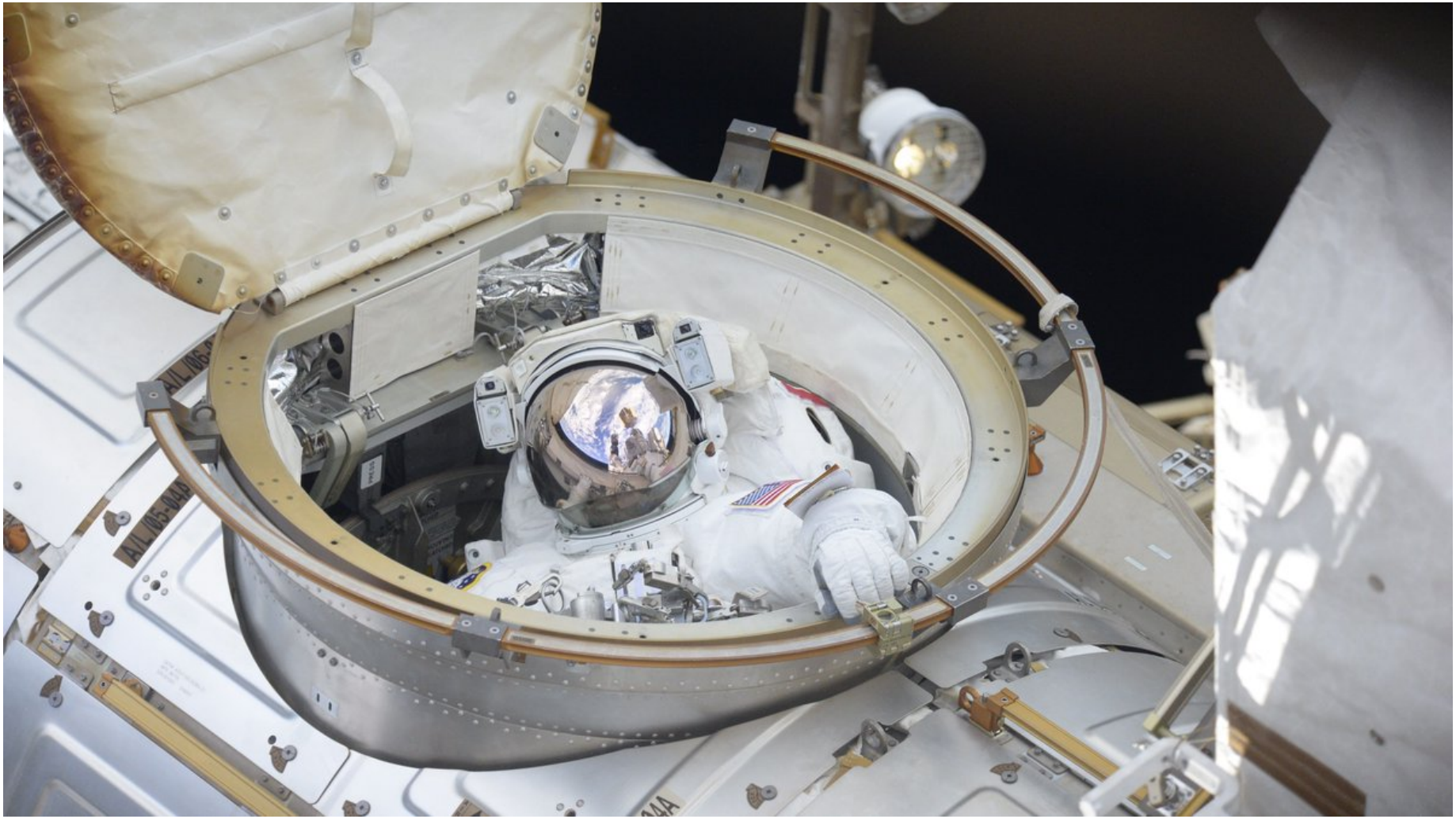
Flip-Flop Analogy

(Airlock)

[https://en.wikipedia.org/wiki/Quest_Joint_Airlock]

[https://fineartamerica.com/featured/1-astronaut-ingresses-the-airlock-hatch-stocktrek-images.html]

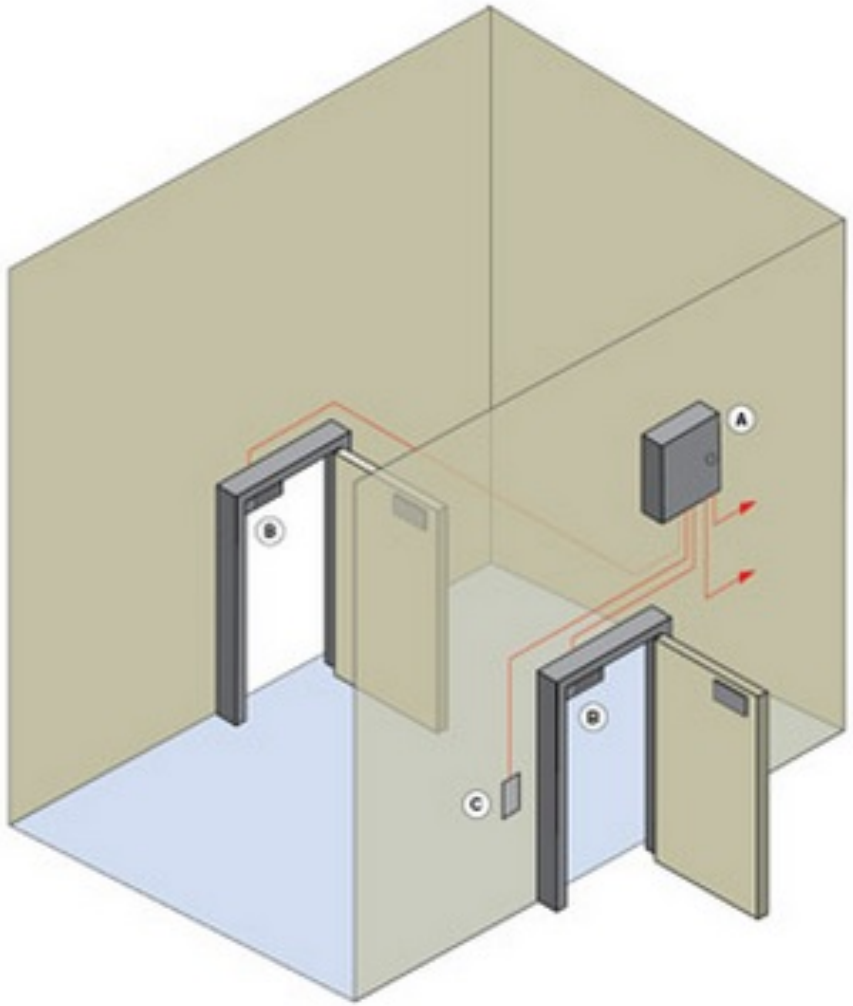
[https://www.allposters.com/-sp/Astronaut-Ingresses-the-Airlock-Hatch-on-the-International-Space-Station-Posters_i12684251_.htm]



[https://www.spaceflightinsider.com/missions/iss/astronauts-install-high-definition-cameras-outside-international-space-station/]

[https://en.wikipedia.org/wiki/Quest_Joint_Airlock]

Airlock on Earth



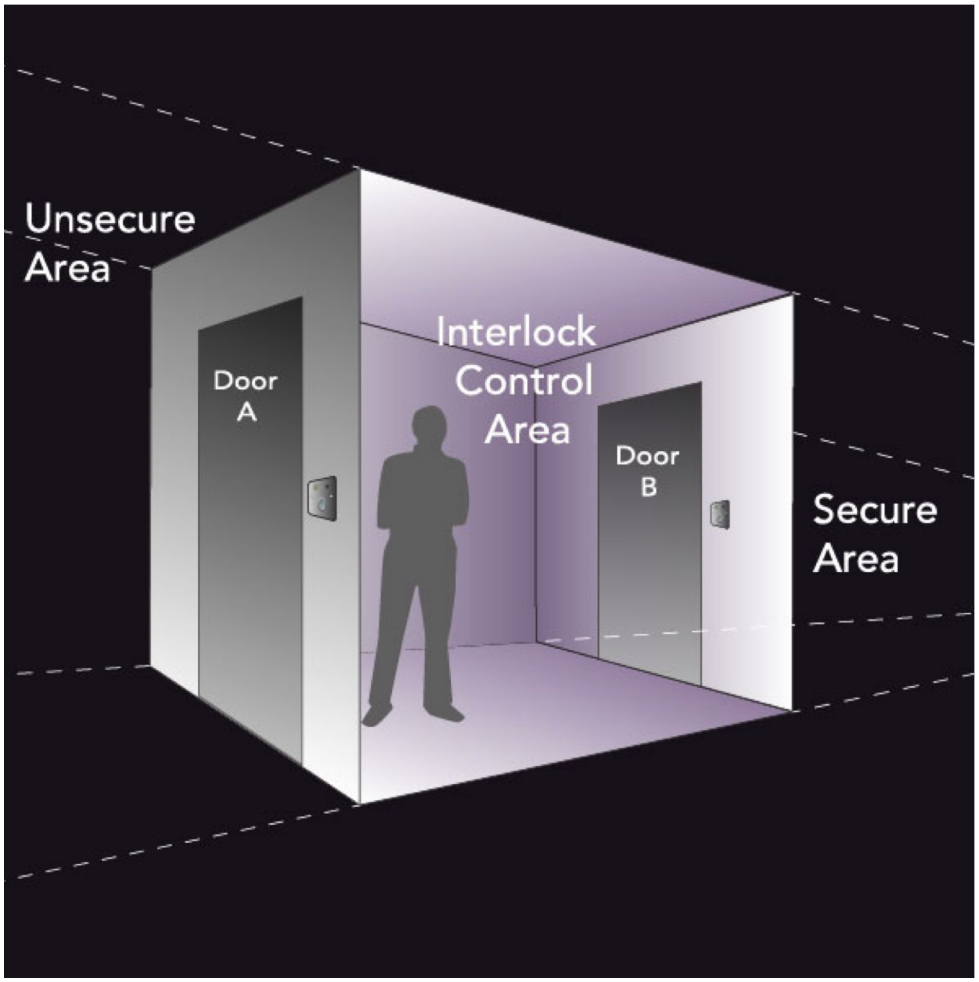
[https://www.maglocks.com/sdc-two-door-airlockmantrapcleanroom-level-1-kit.html]

D Flip-Flop Analogy

Outer Door
Will Not Unlock When
Inner Door is Open

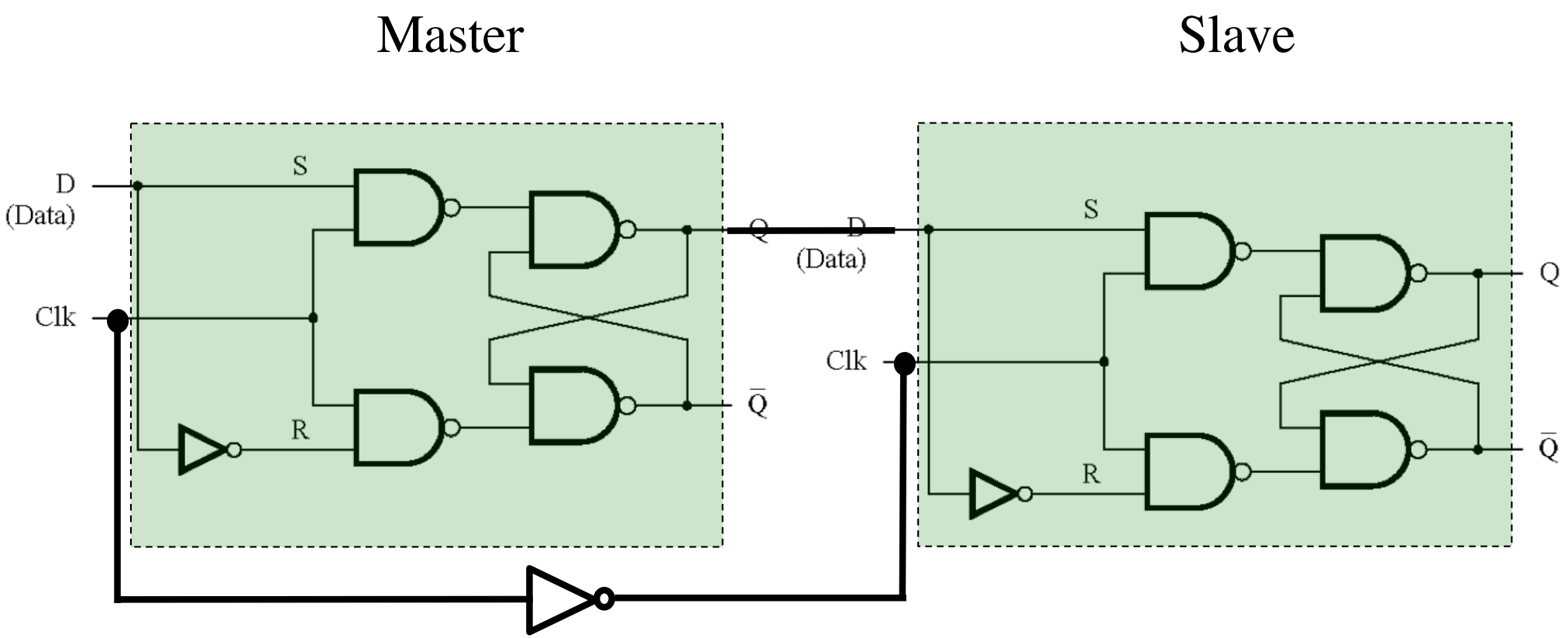
Inner Door
Will Not Unlock When
OuterDoor is Open

[https://www.nortechcontrol.com/solutions/people/door-interlock/]

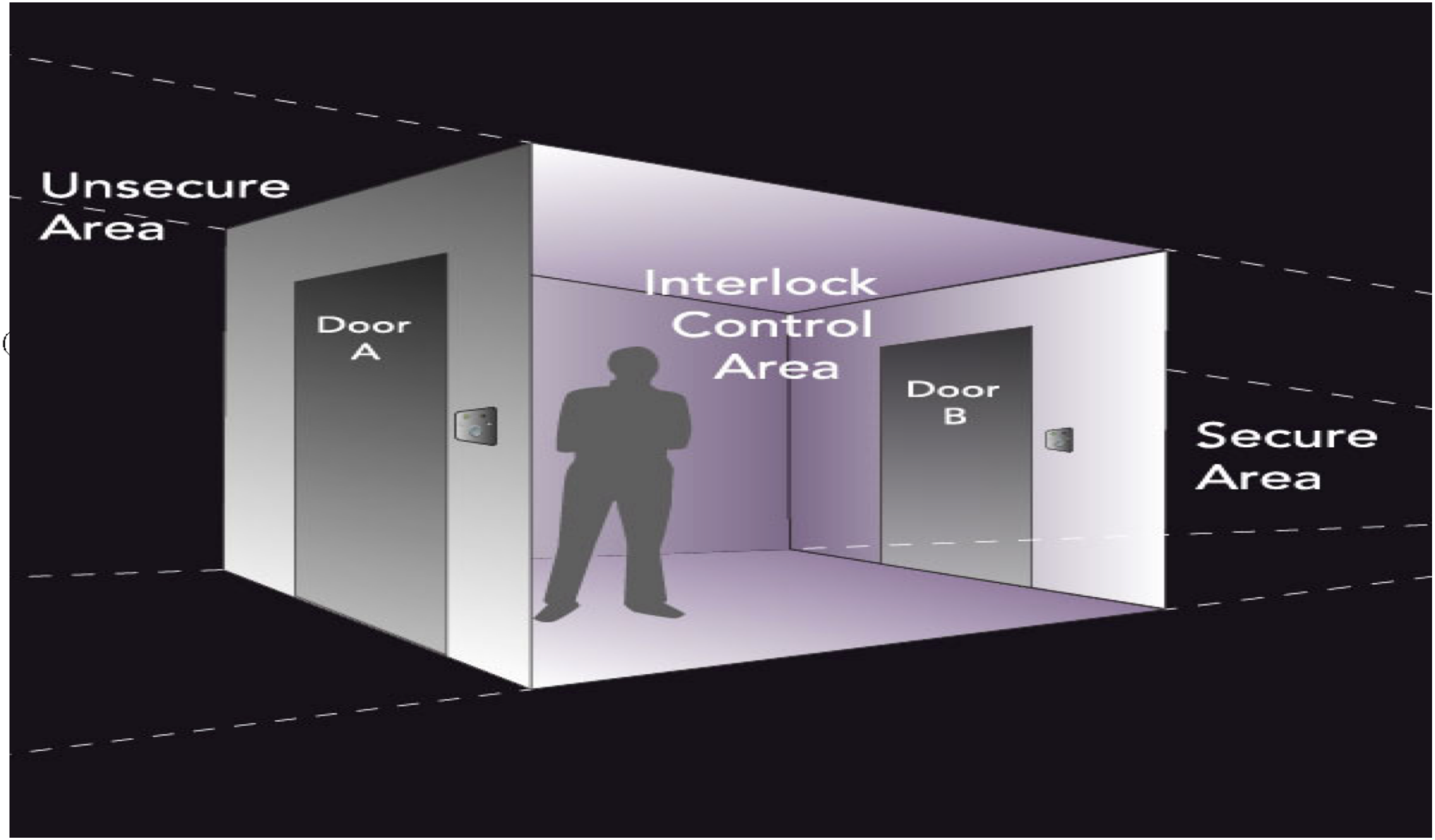


[https://www.insafe.com/activa-pro-interlock/]

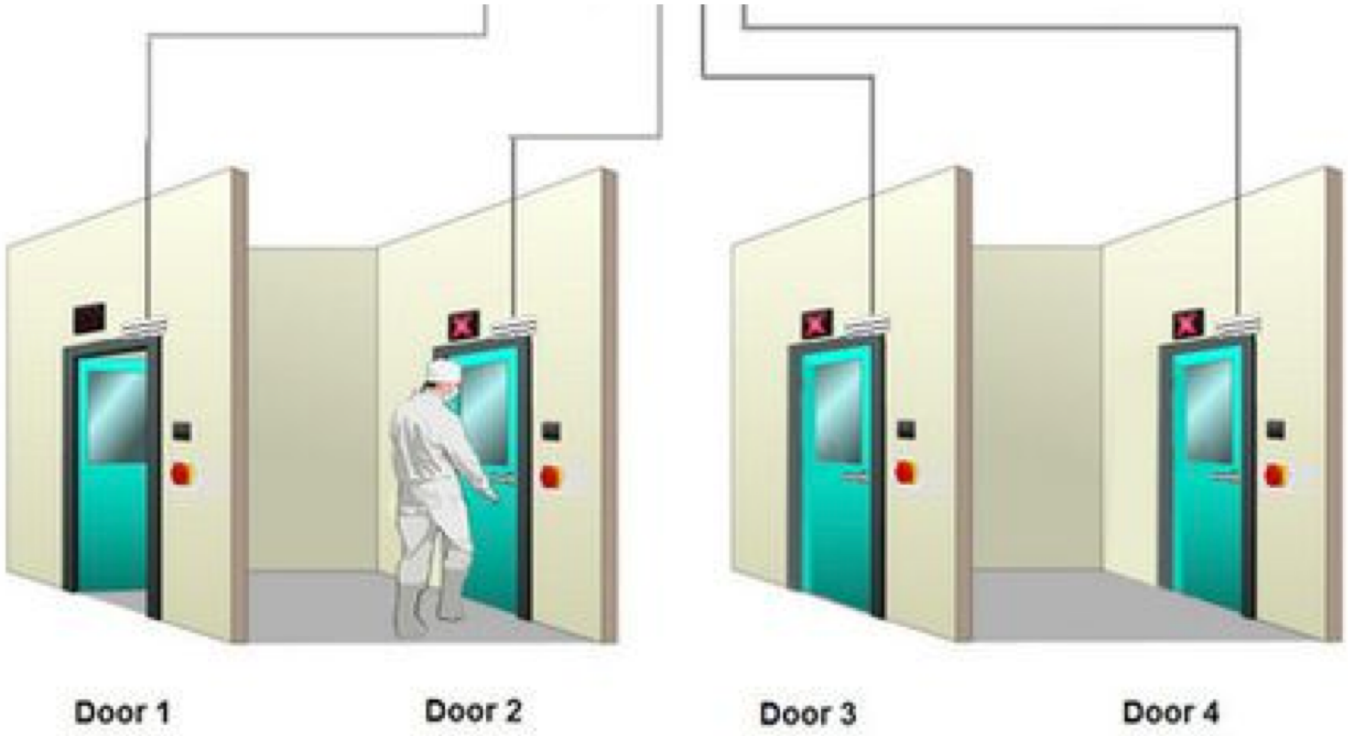
D Flip-Flop Analogy



D Flip-Flop Analogy



Shift-Register Analogy



[https://tiimg.tistatic.com/fp/1/004/777/door-interlock-controllers-by-radix-566.jpg]

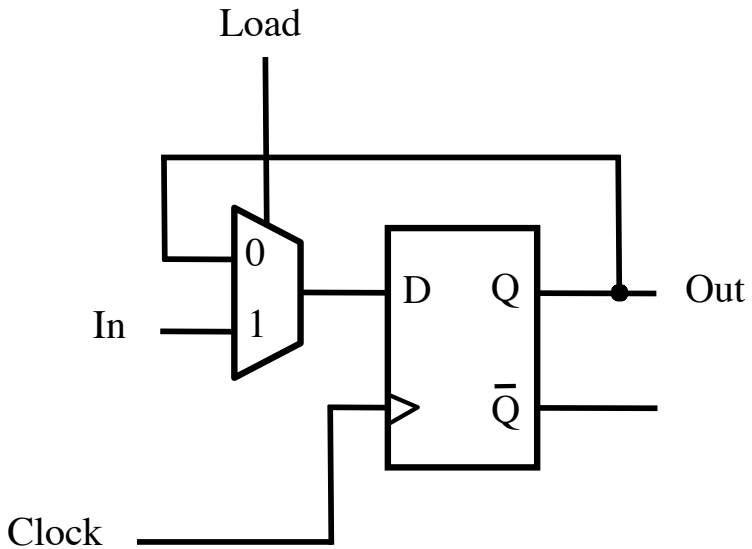
Registers

Register
(Definition)

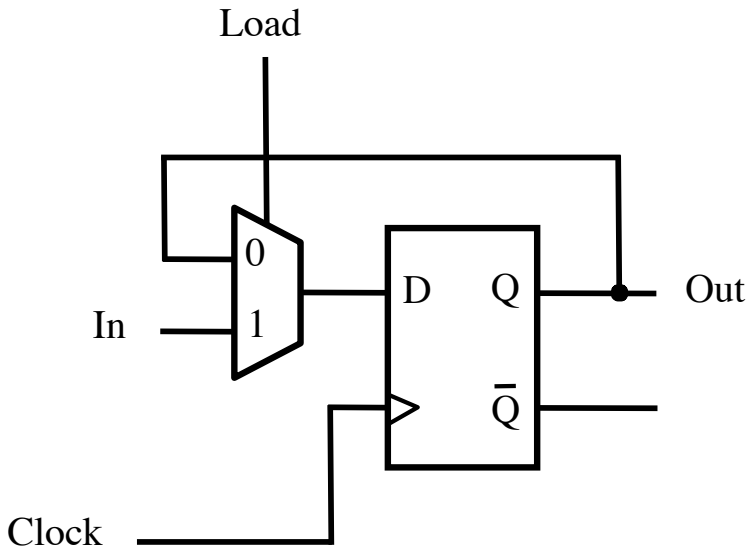
An n-bit structure consisting of flip-flops

Parallel-Access Register

1-Bit Parallel-Access Register



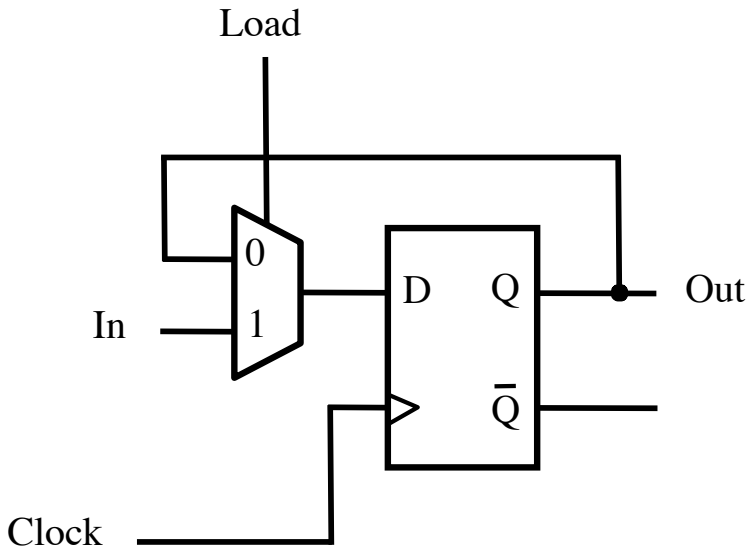
1-Bit Parallel-Access Register



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.

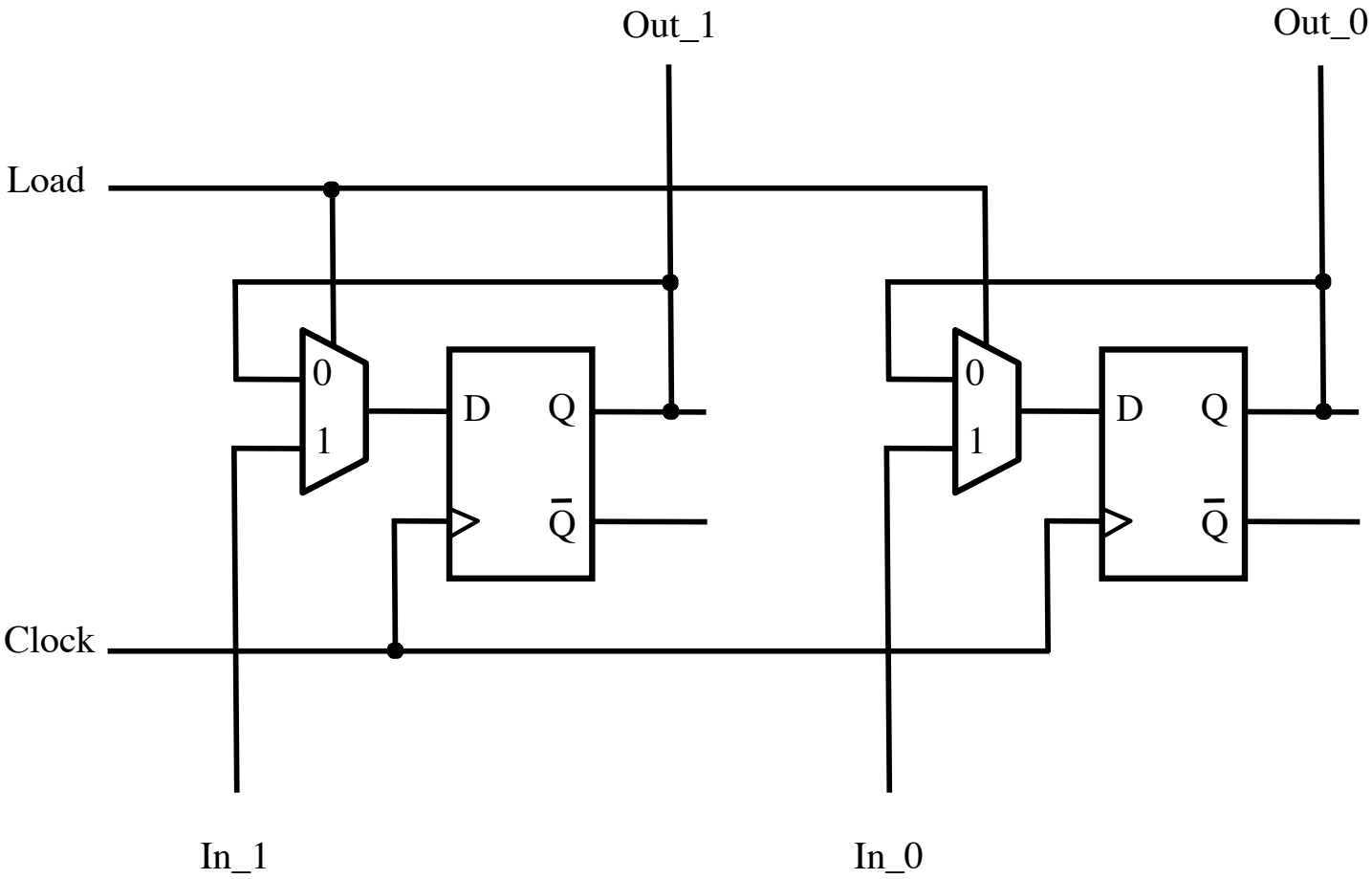
1-Bit Parallel-Access Register



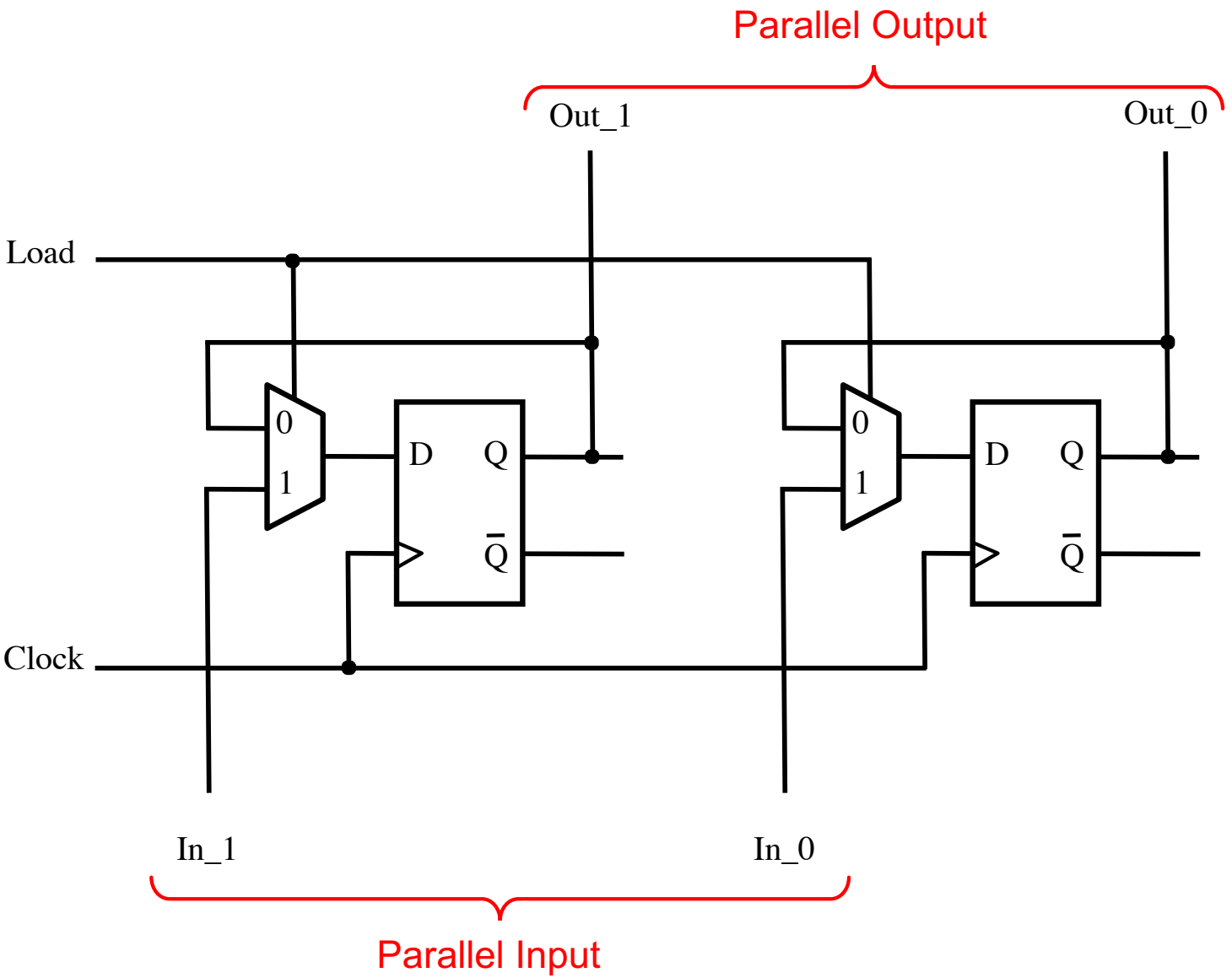
If Load = 0, then retain the old value.

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

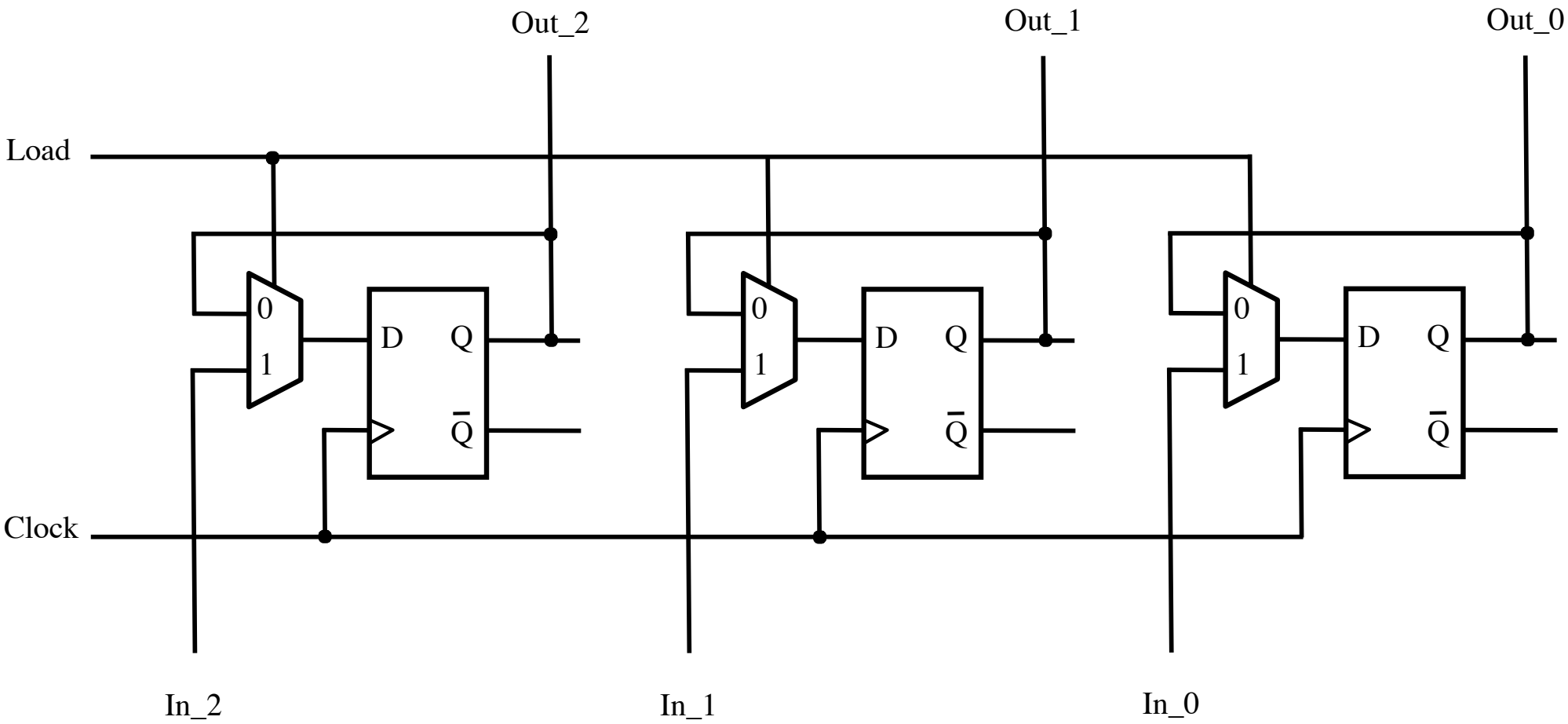
2-Bit Parallel-Access Register



2-Bit Parallel-Access Register

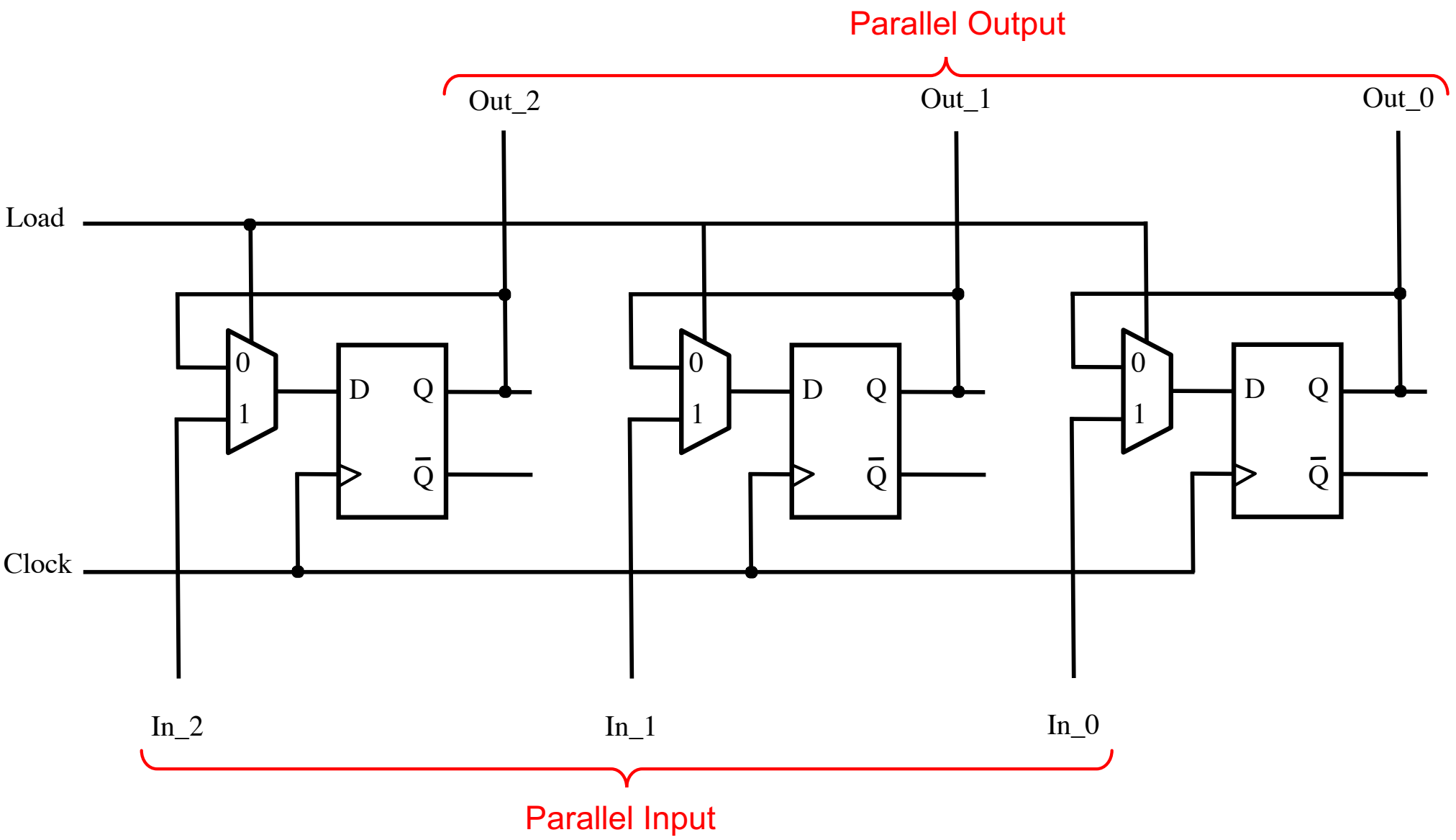


3-Bit Parallel-Access Register

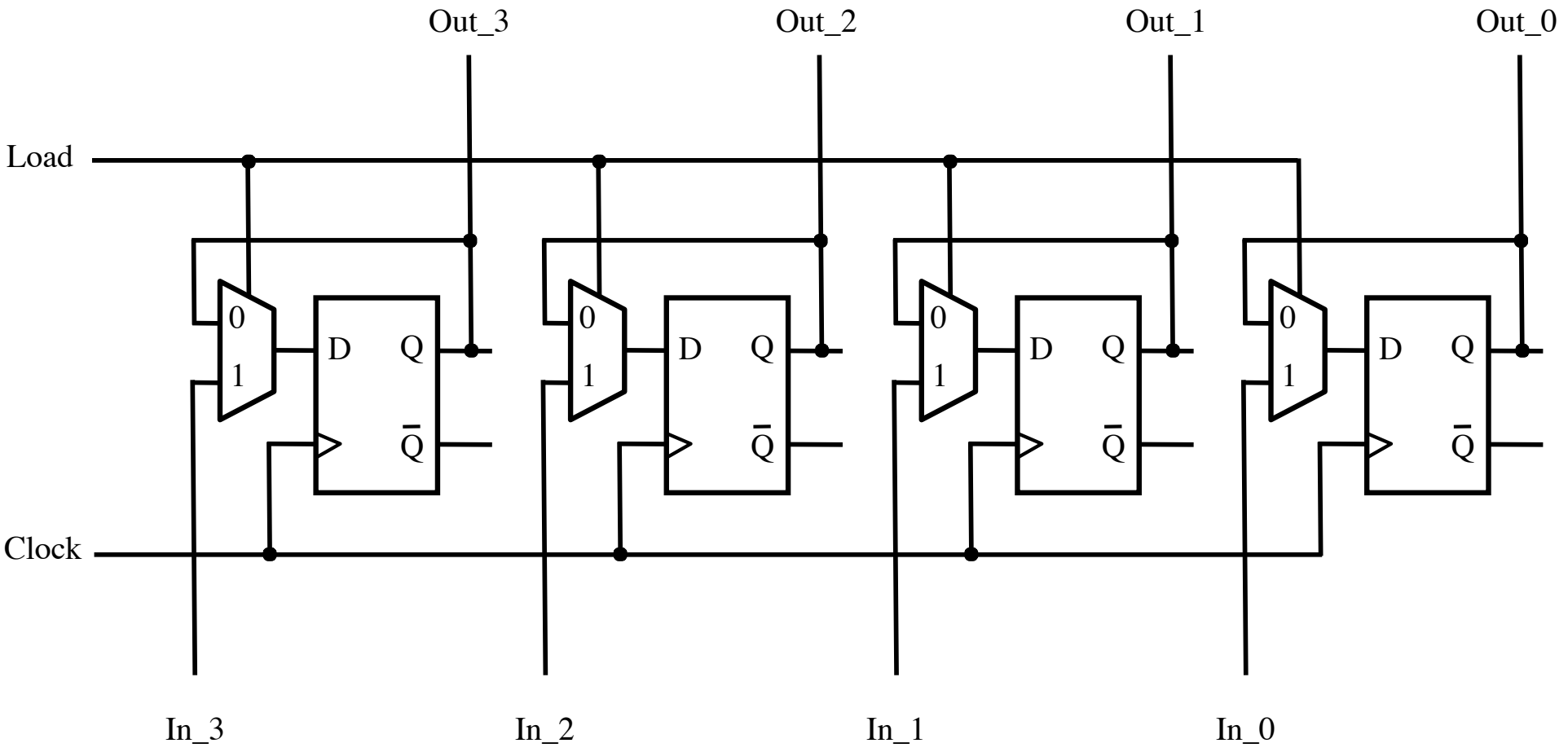


Notice that all flip-flops are on the same clock cycle.

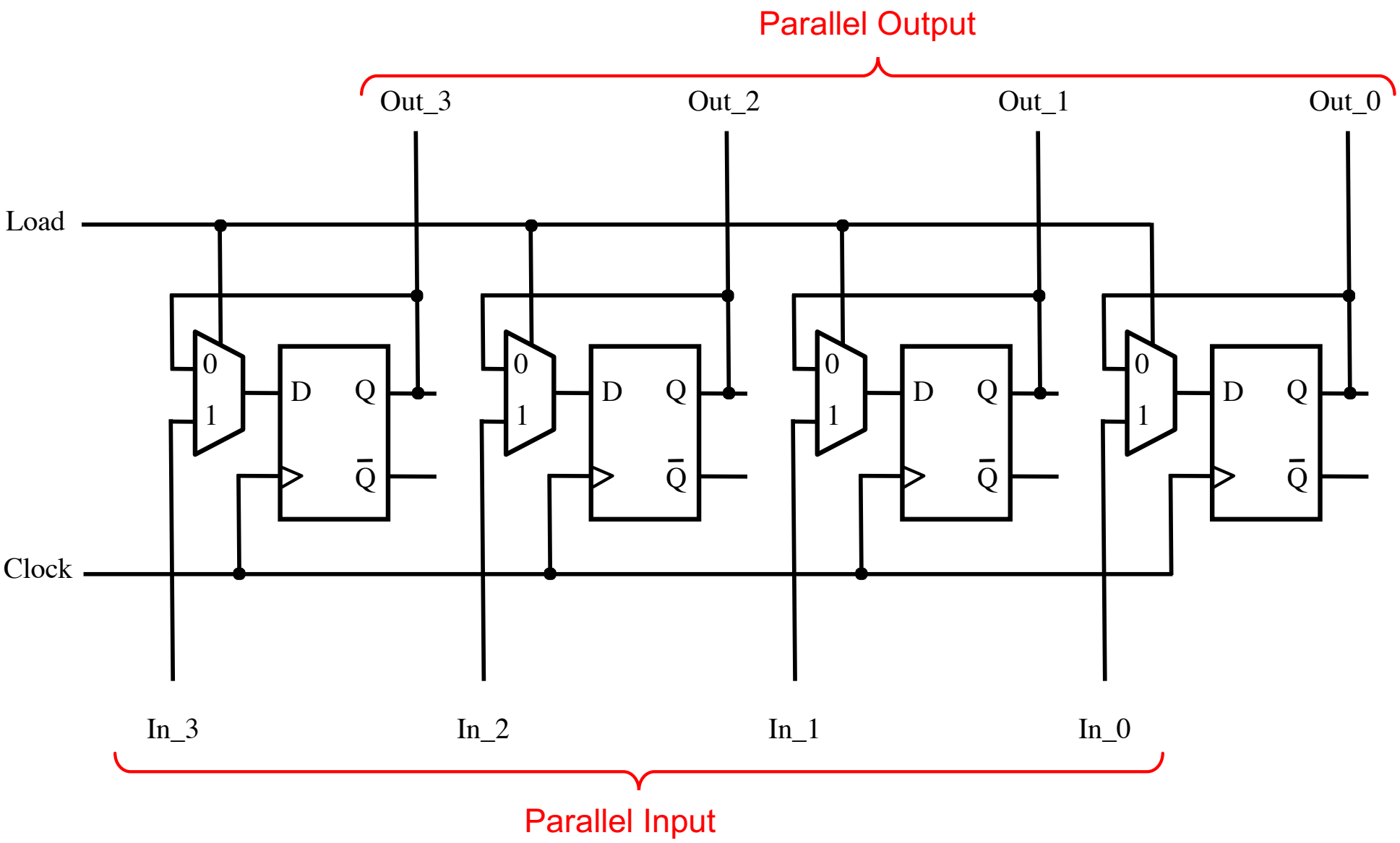
3-Bit Parallel-Access Register



4-Bit Parallel-Access Register

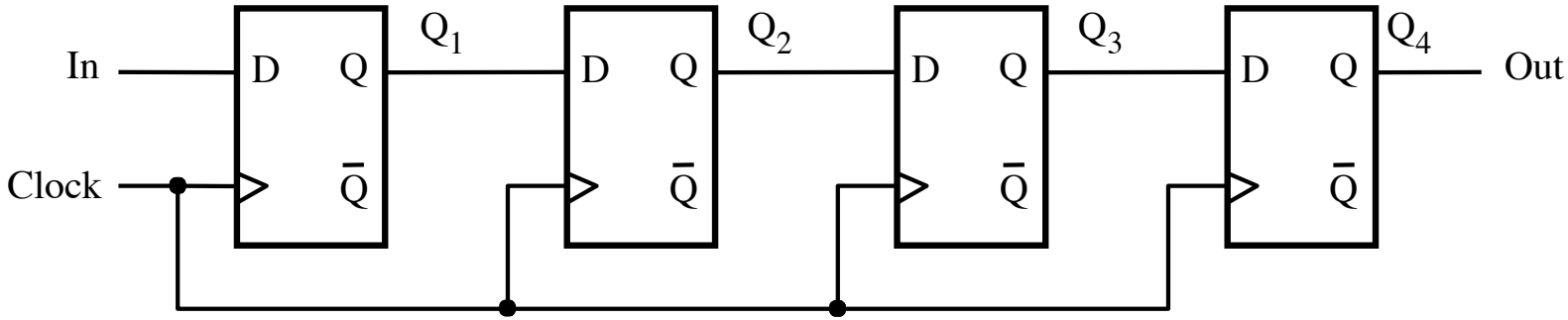


4-Bit Parallel-Access Register



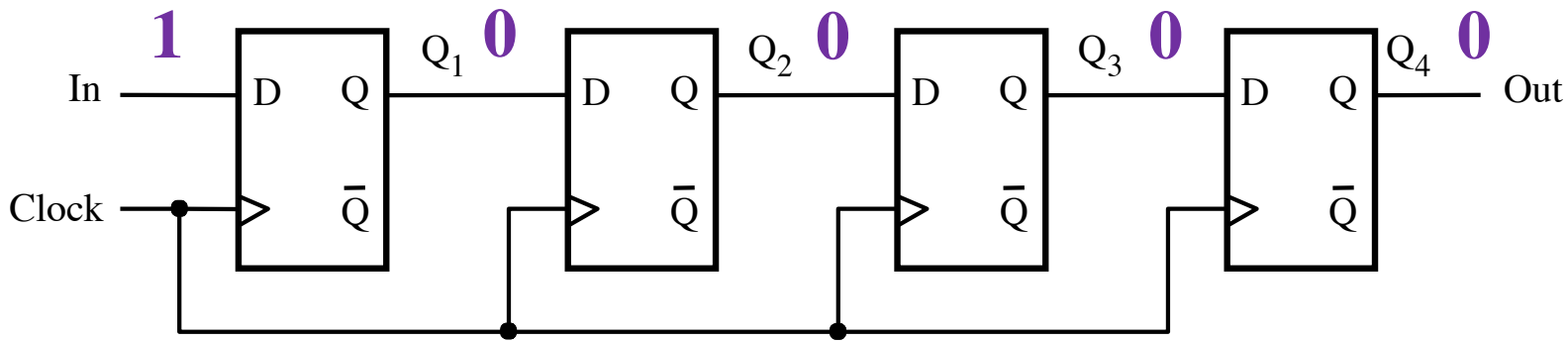
Shift Register

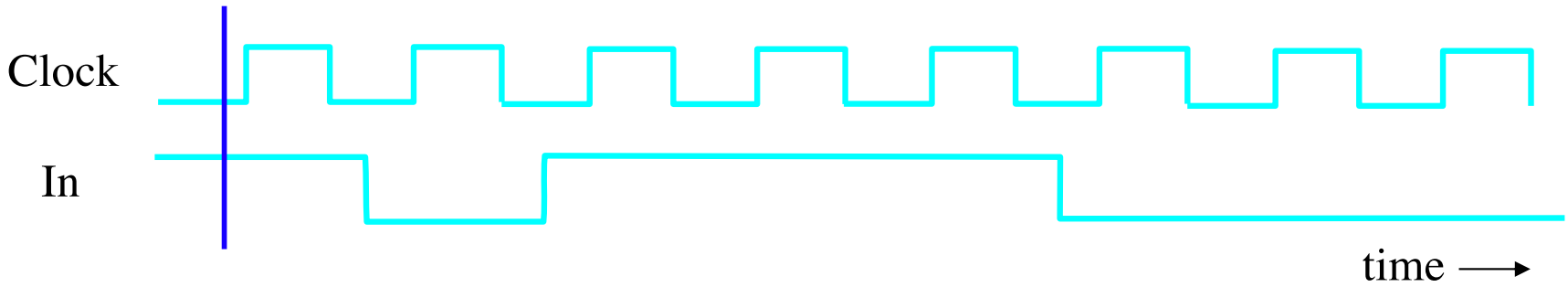
A simple shift register



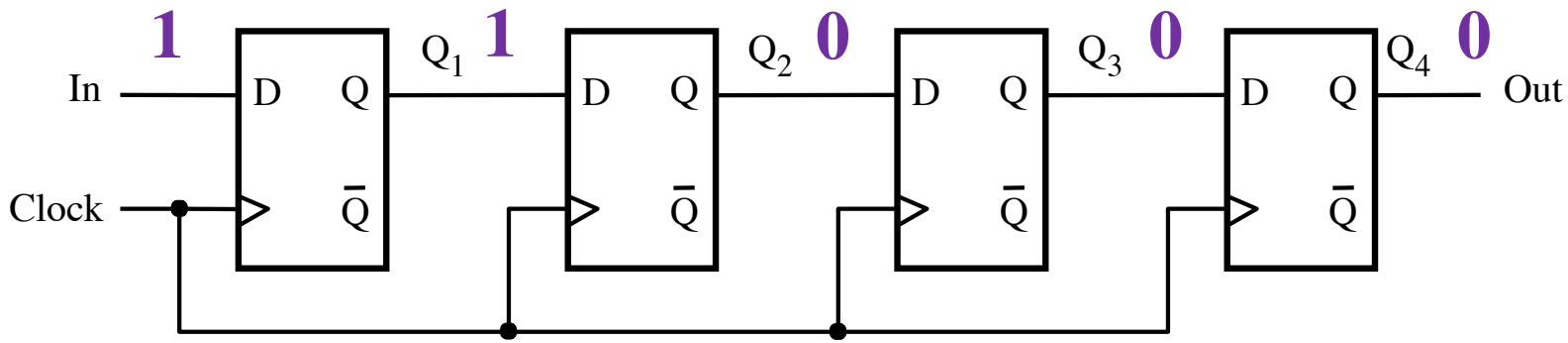
[Figure 5.17a from the textbook]

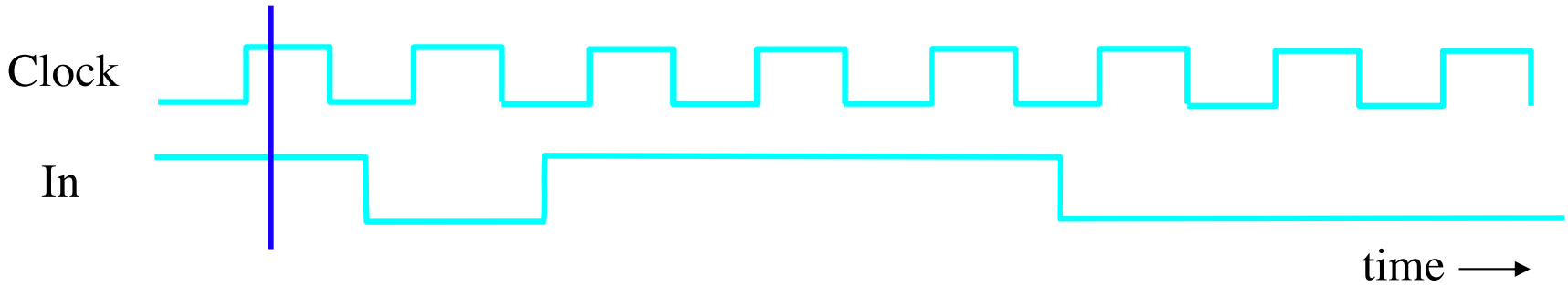
Shift Register Simulation



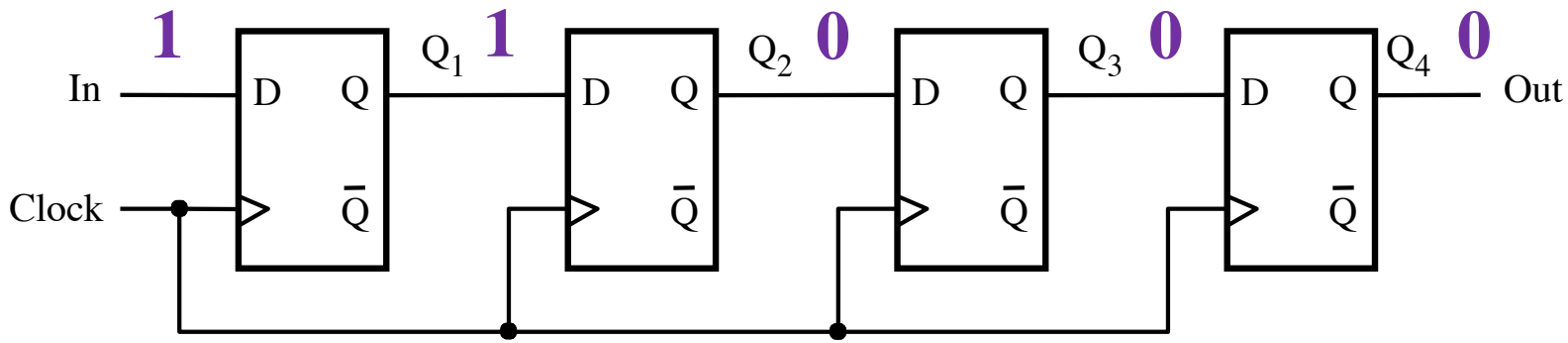


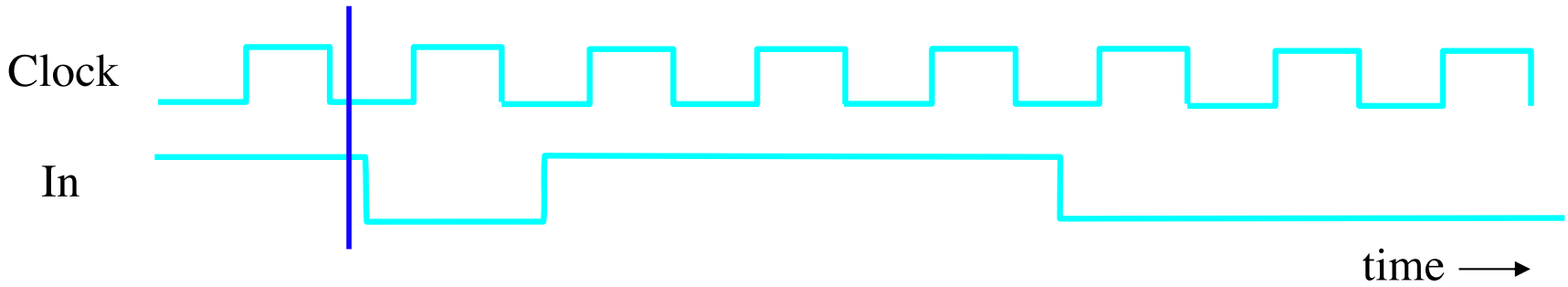
Shift Register Simulation



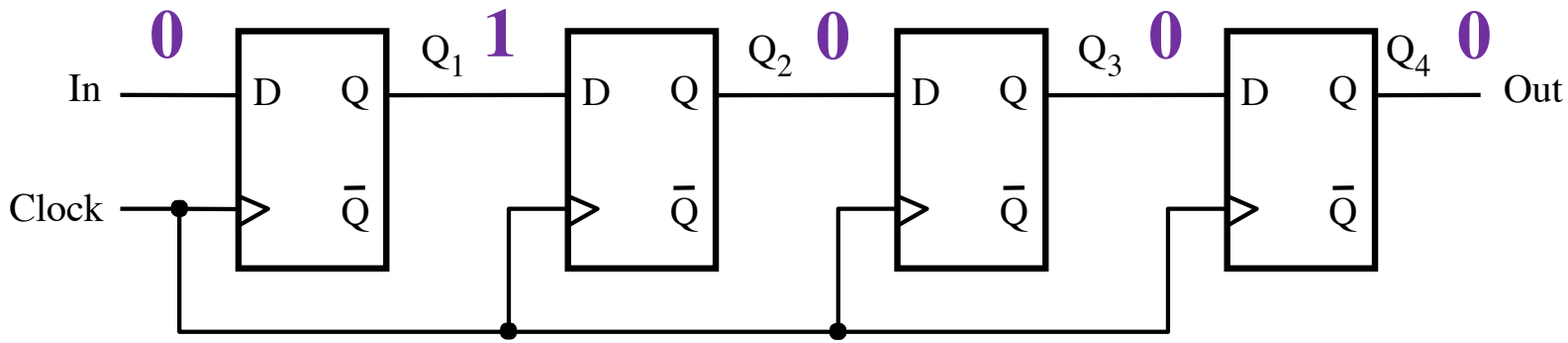


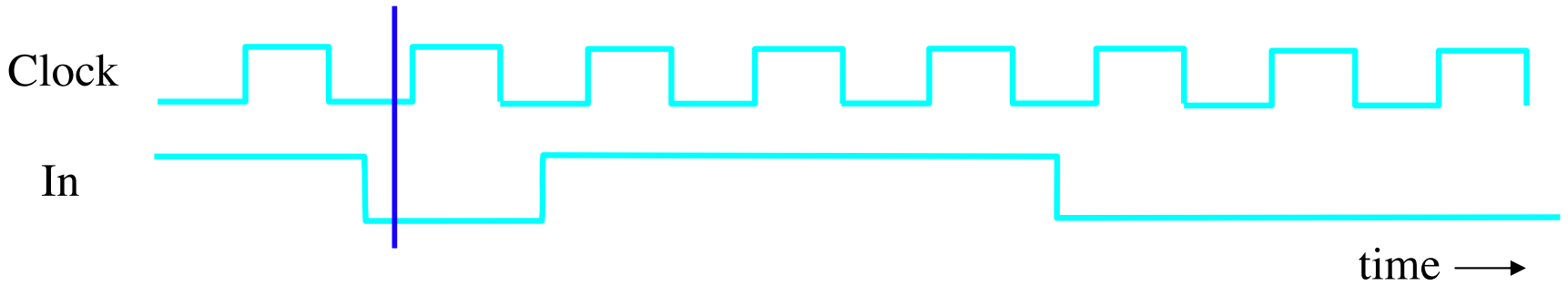
Shift Register Simulation



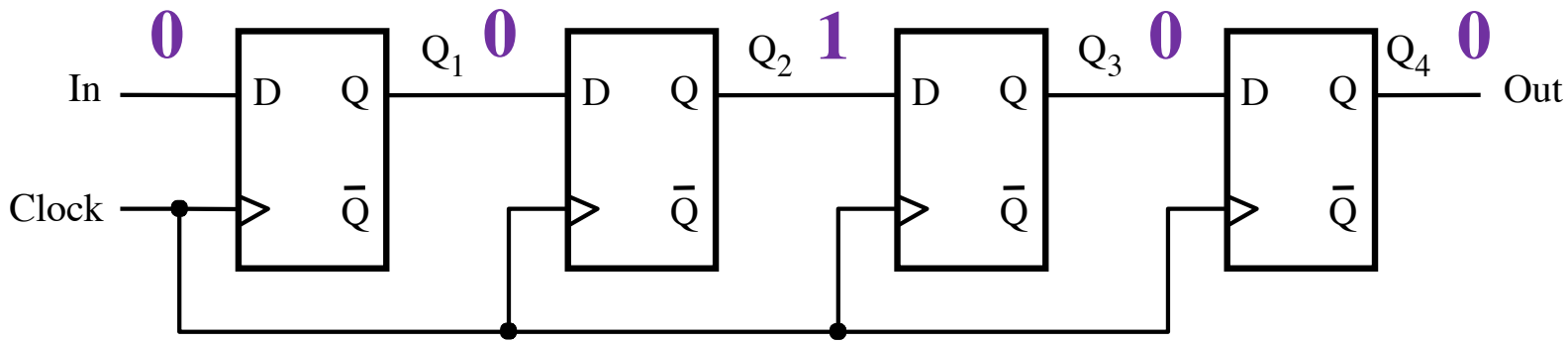


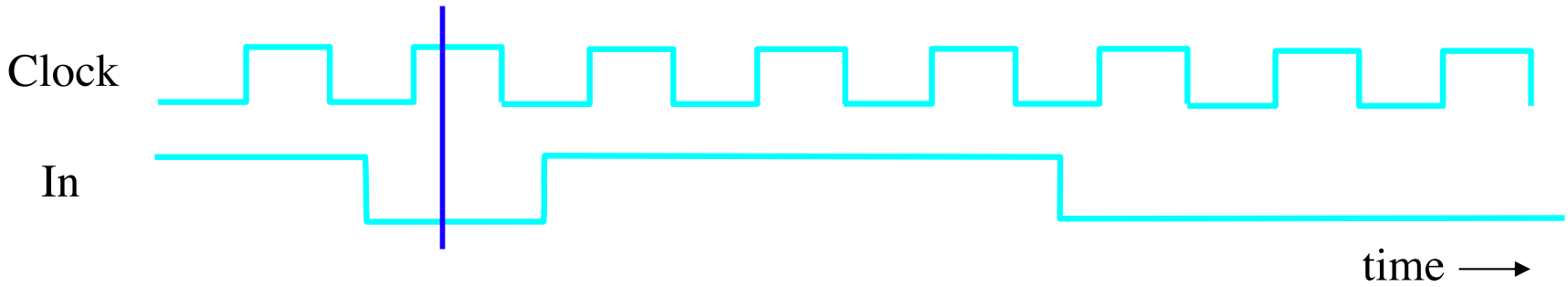
Shift Register Simulation



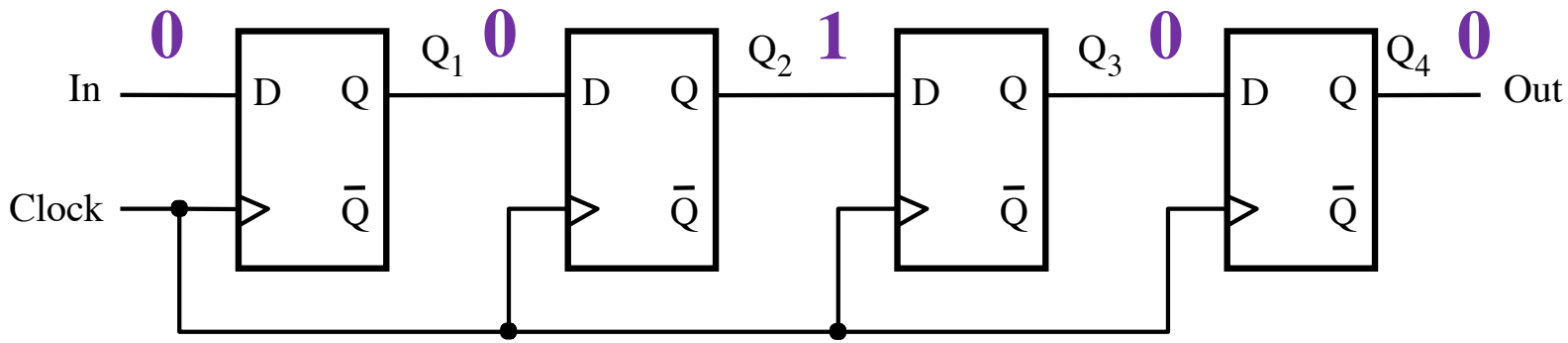


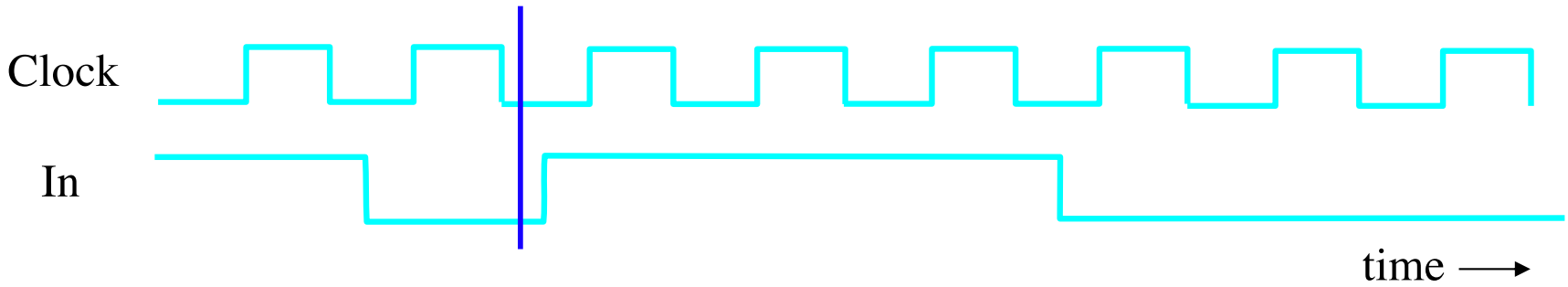
Shift Register Simulation



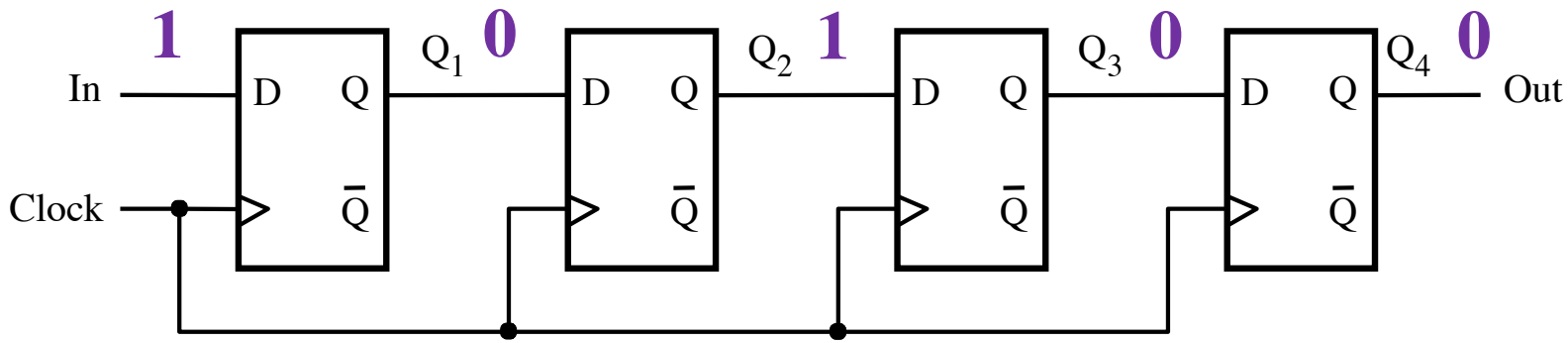


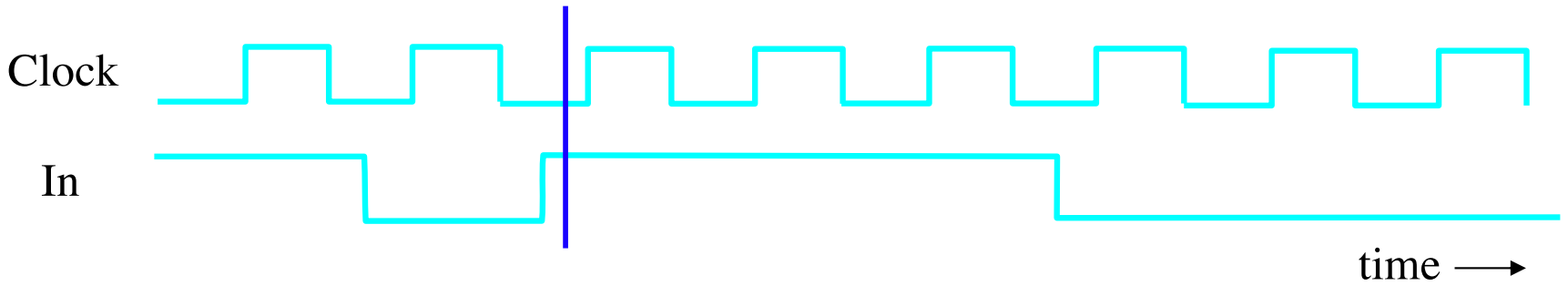
Shift Register Simulation



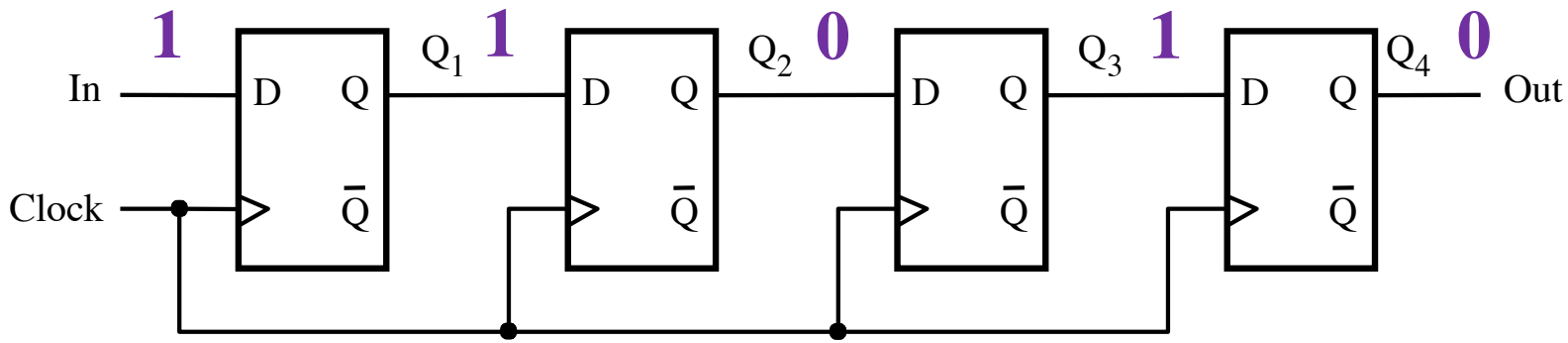


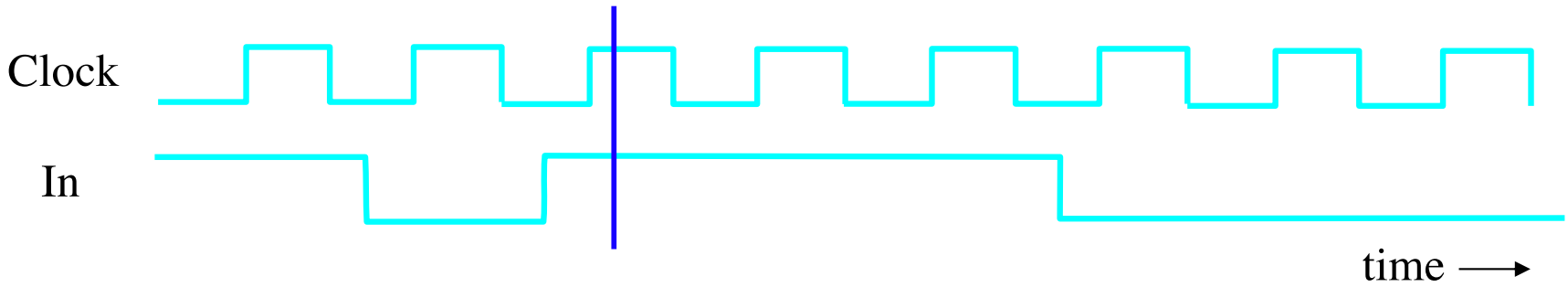
Shift Register Simulation



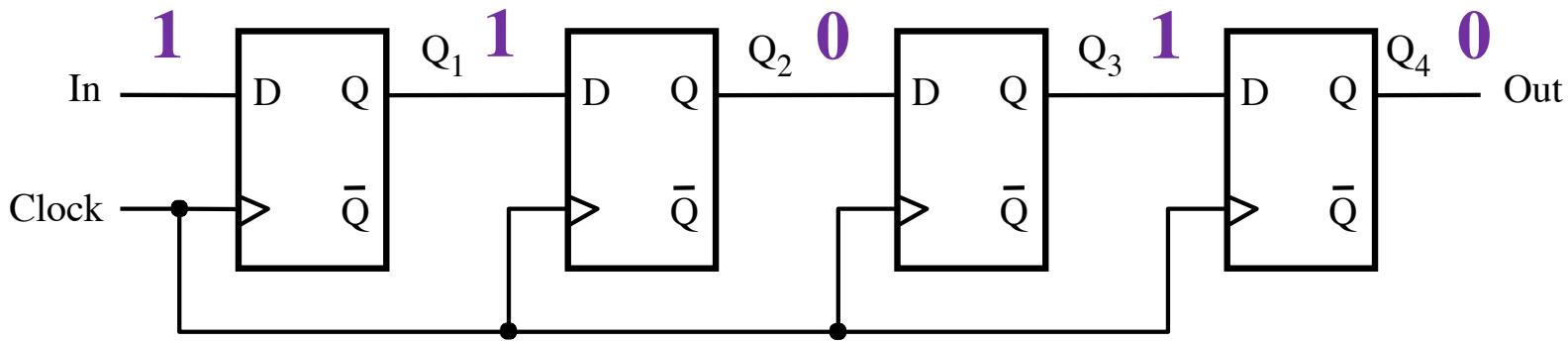


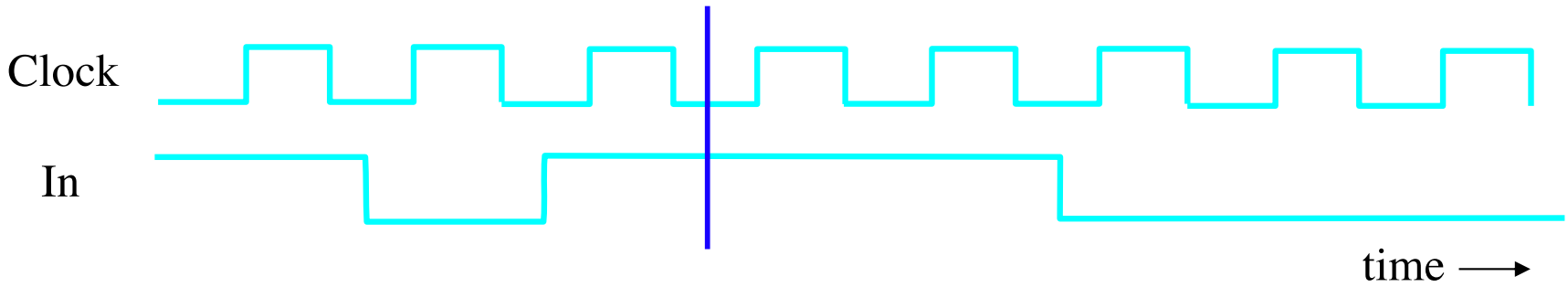
Shift Register Simulation



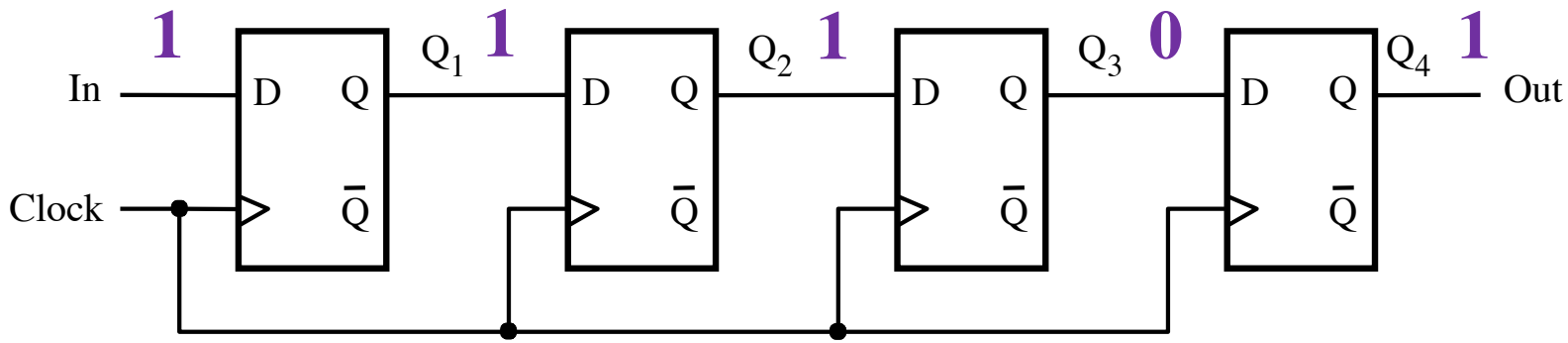


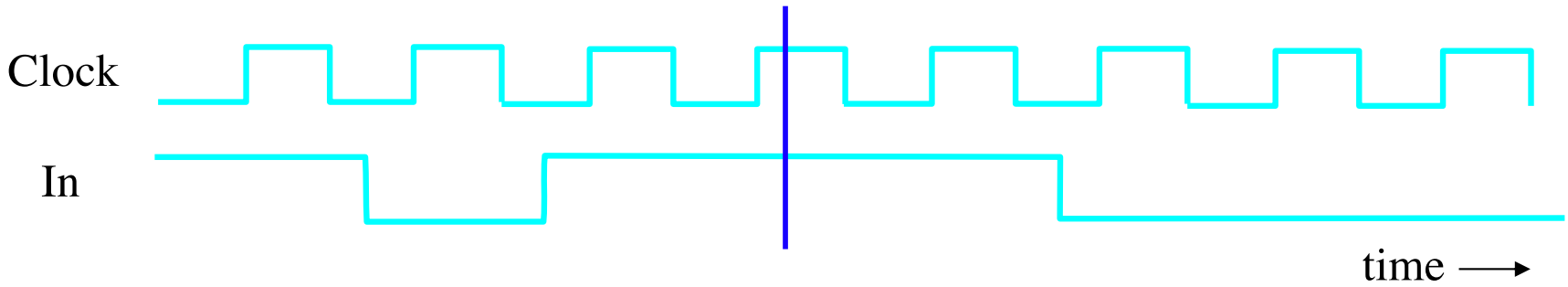
Shift Register Simulation



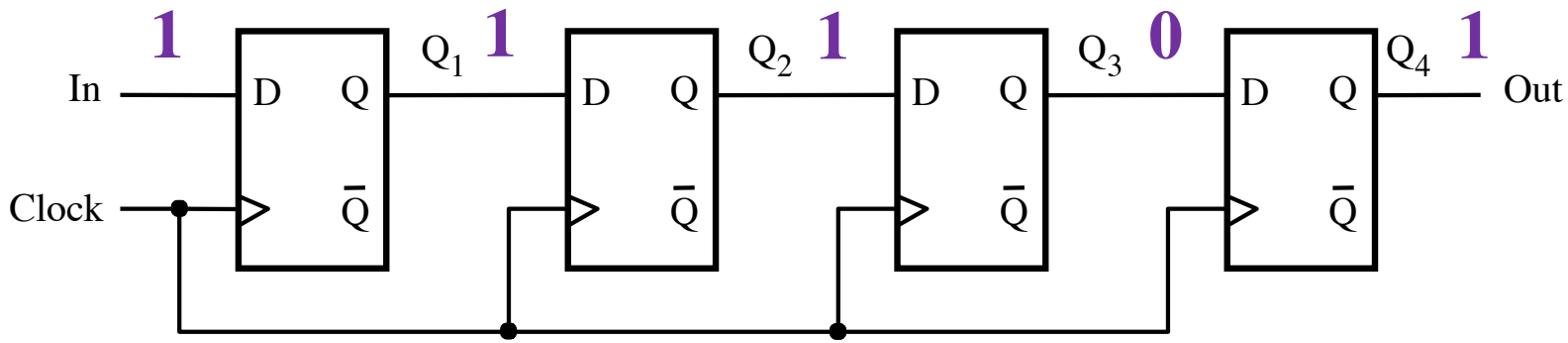


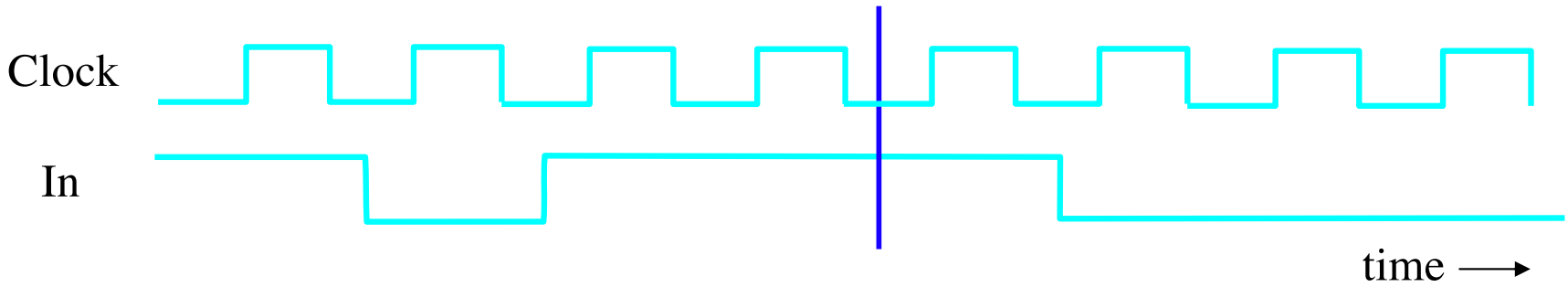
Shift Register Simulation



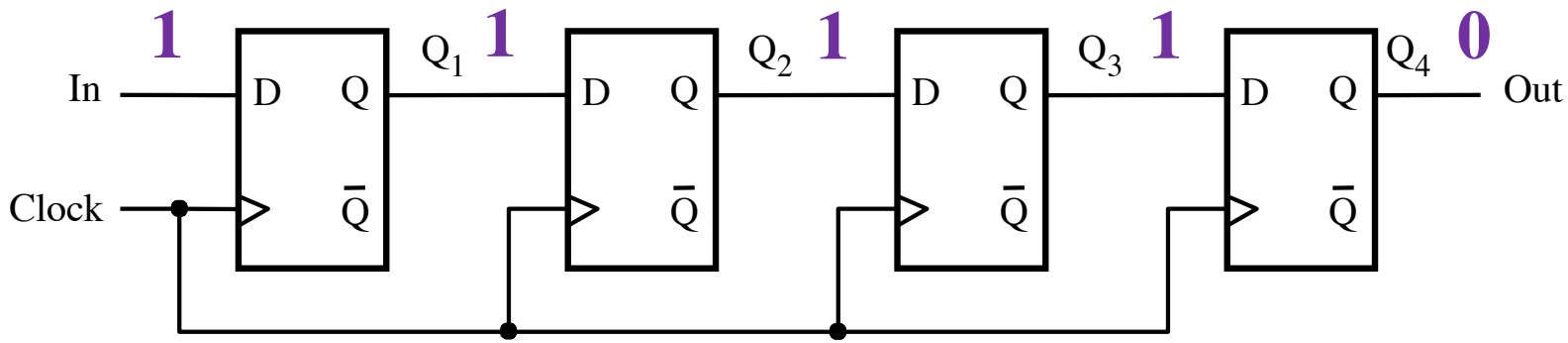


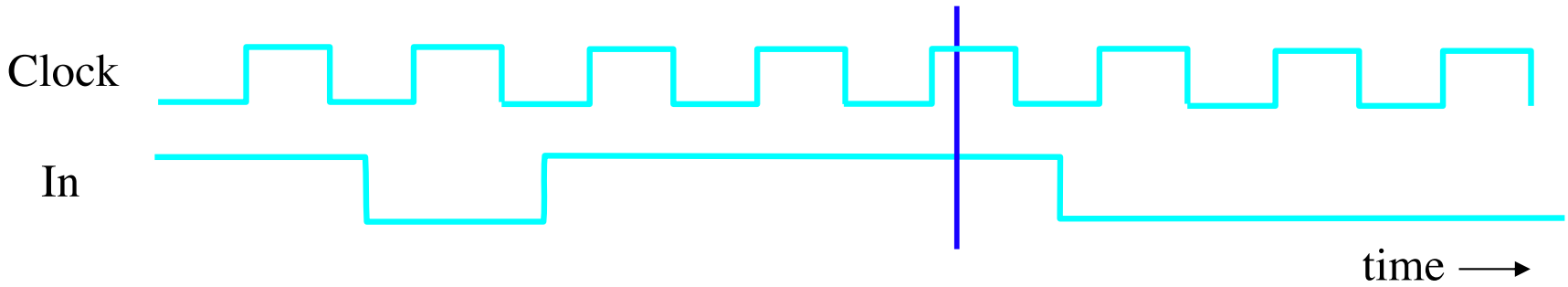
Shift Register Simulation



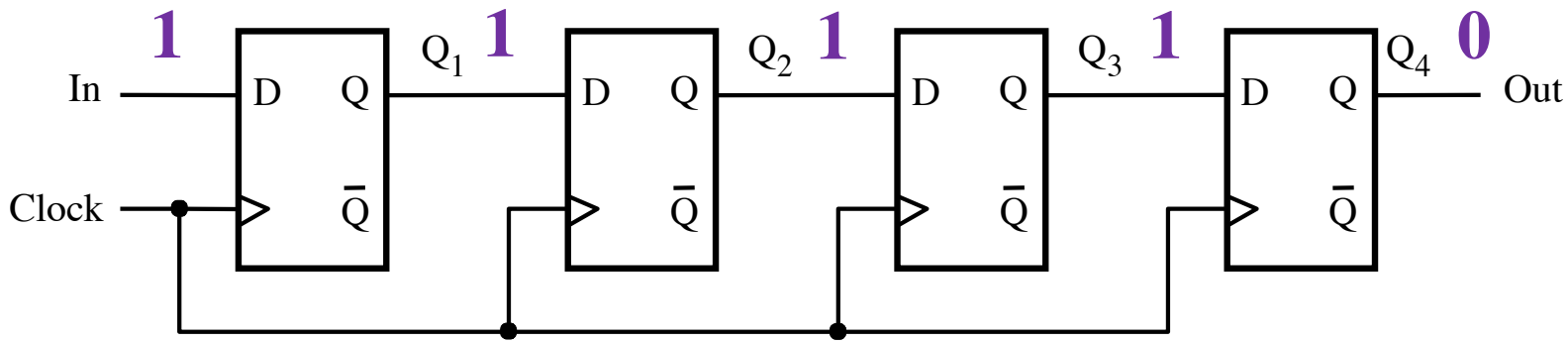


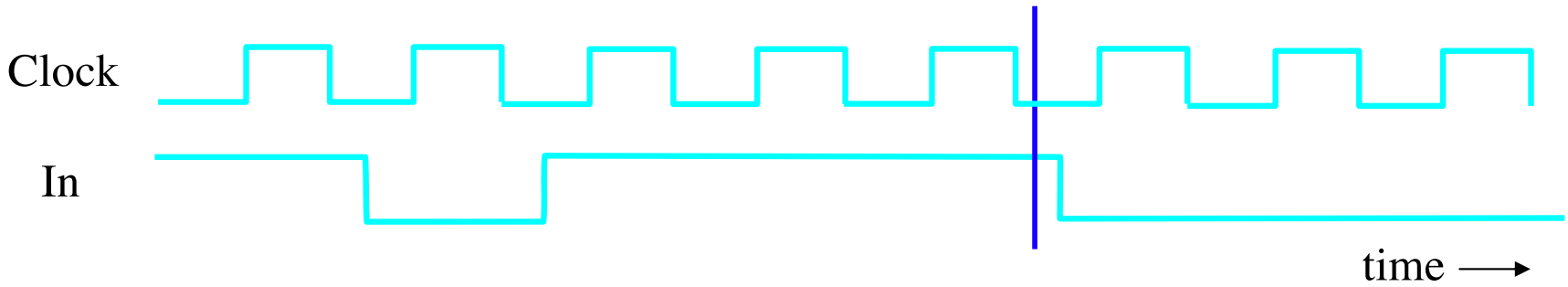
Shift Register Simulation



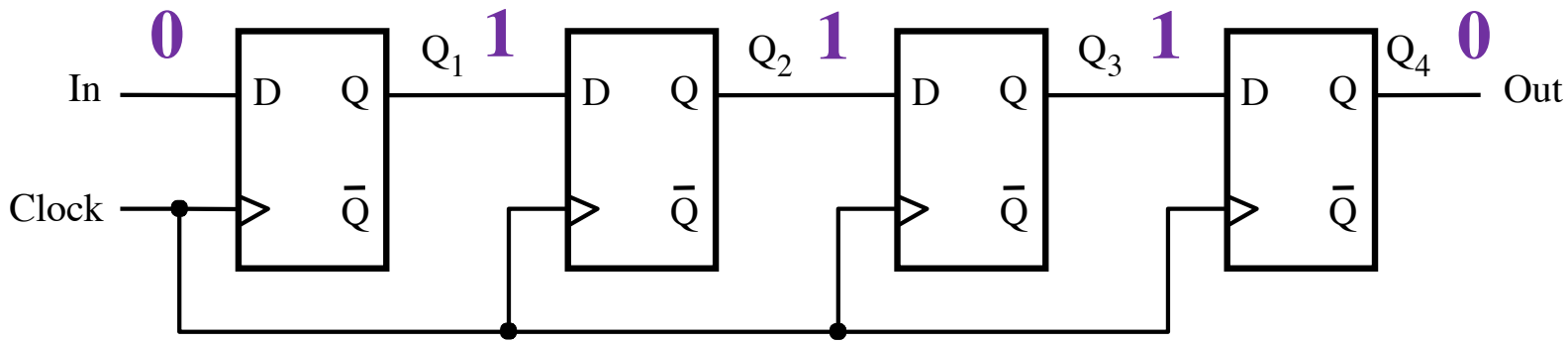


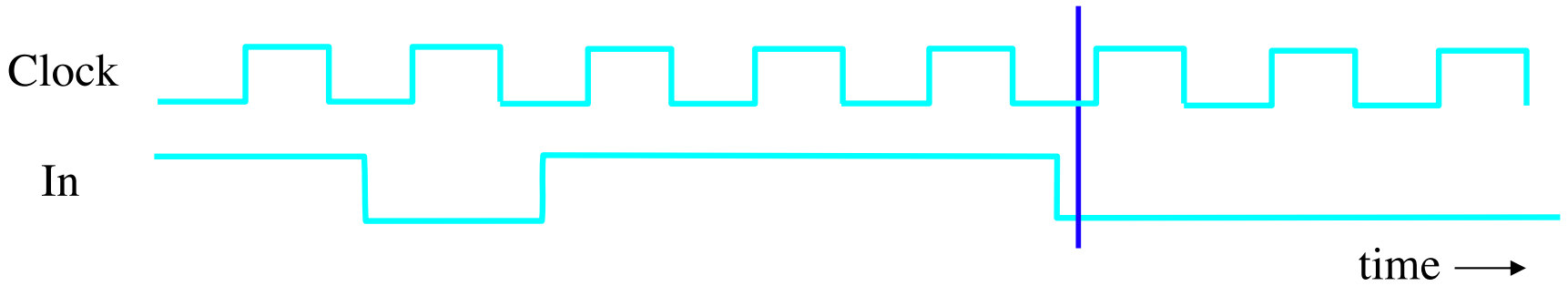
Shift Register Simulation



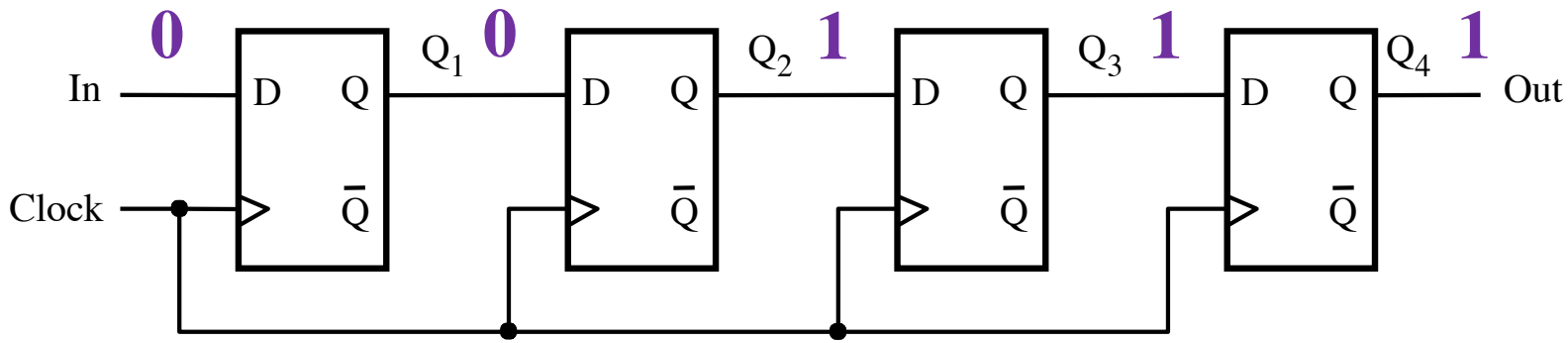


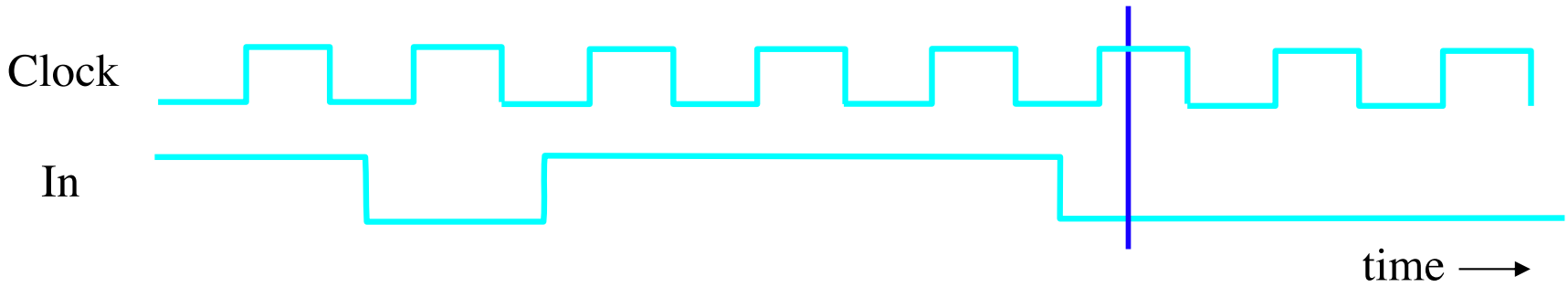
Shift Register Simulation



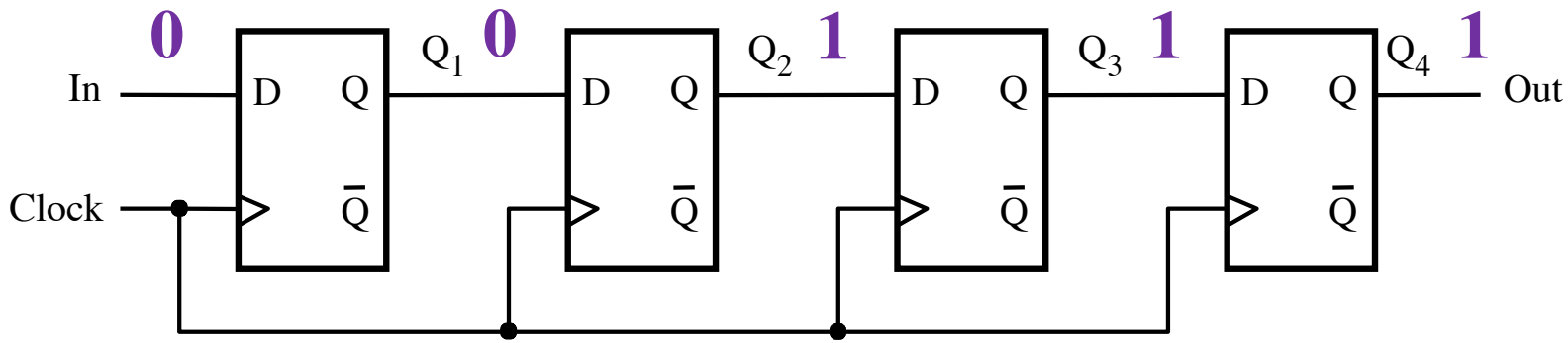


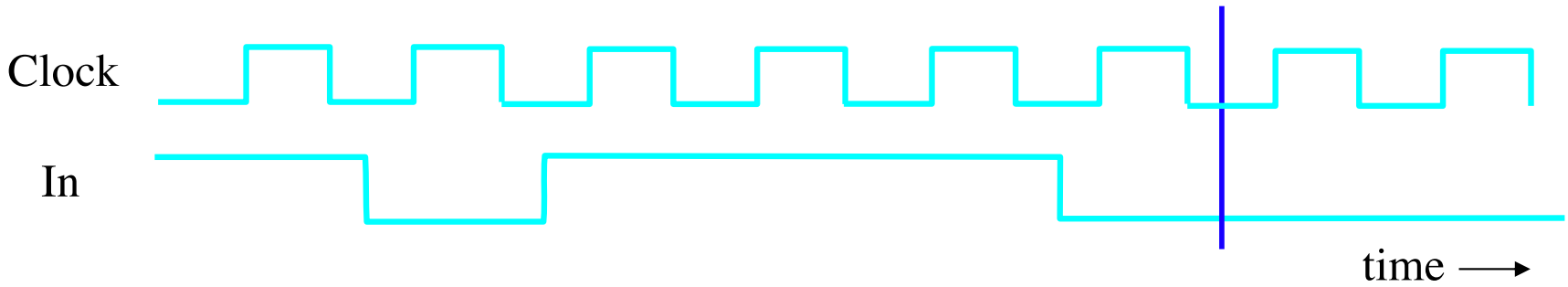
Shift Register Simulation



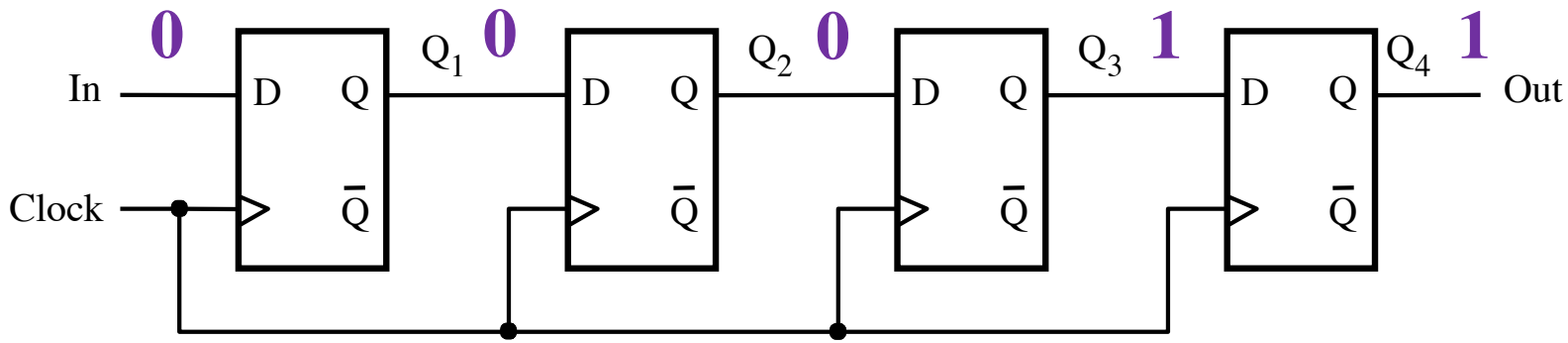


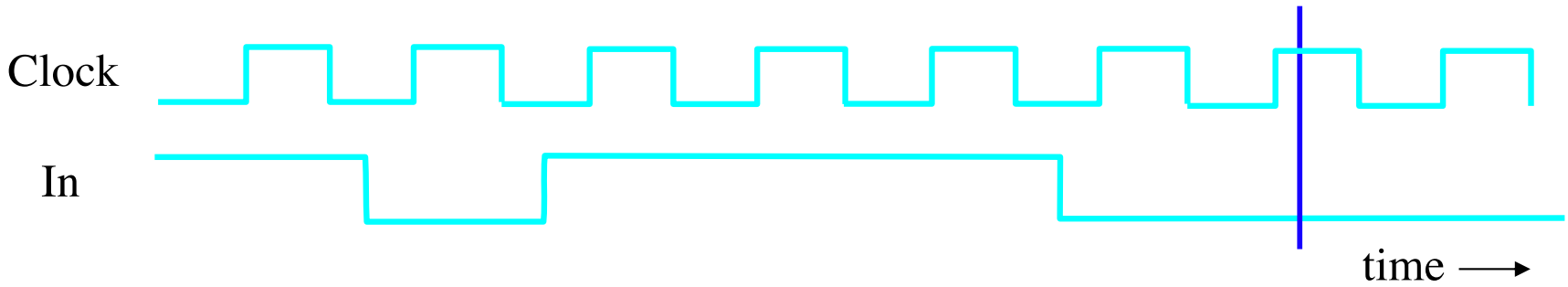
Shift Register Simulation



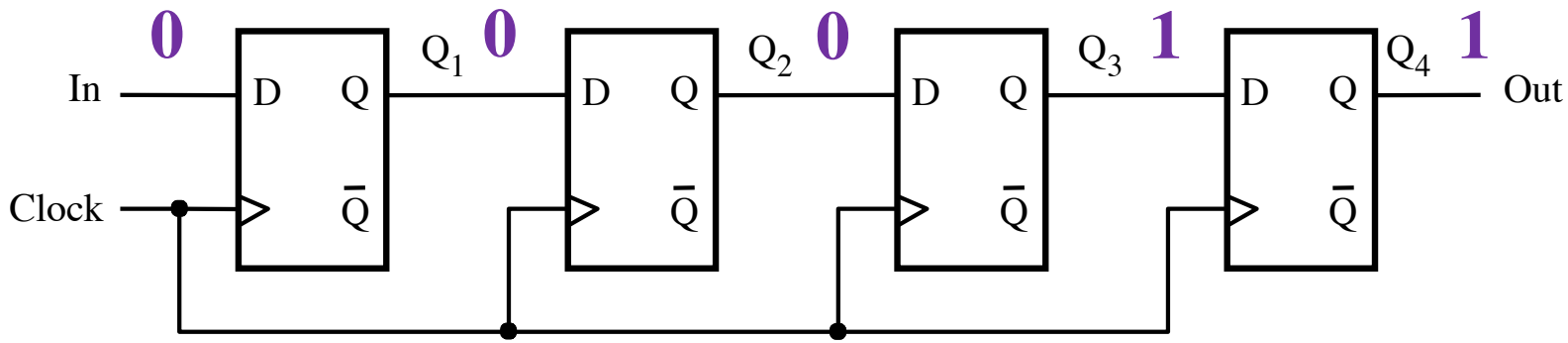


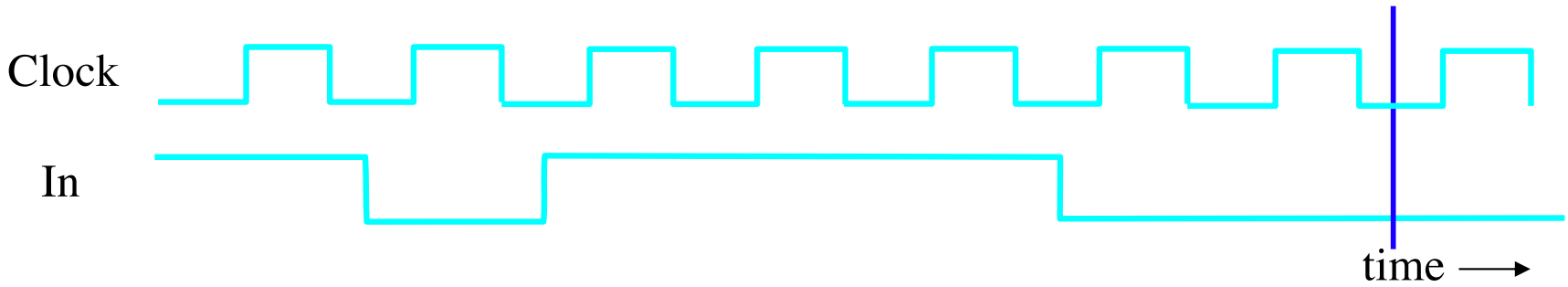
Shift Register Simulation



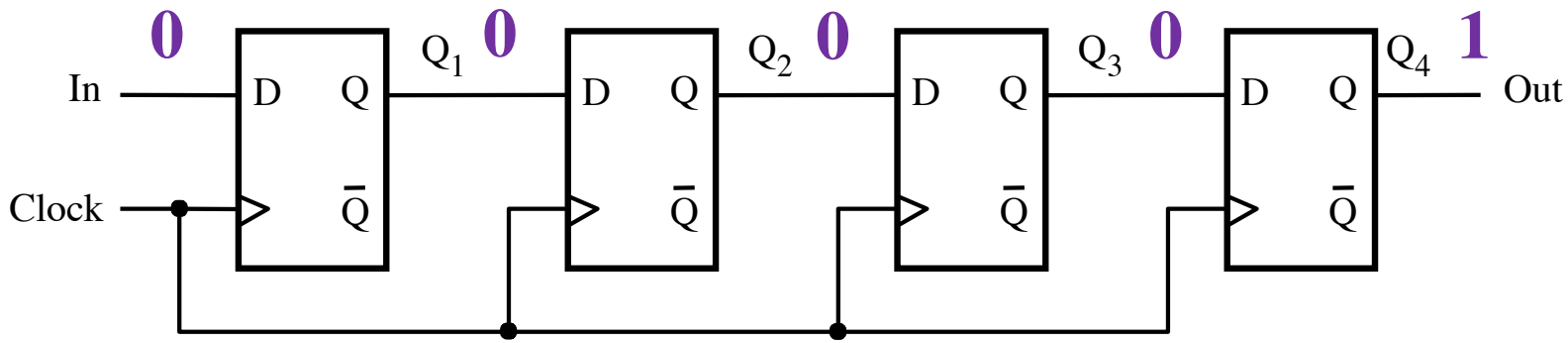


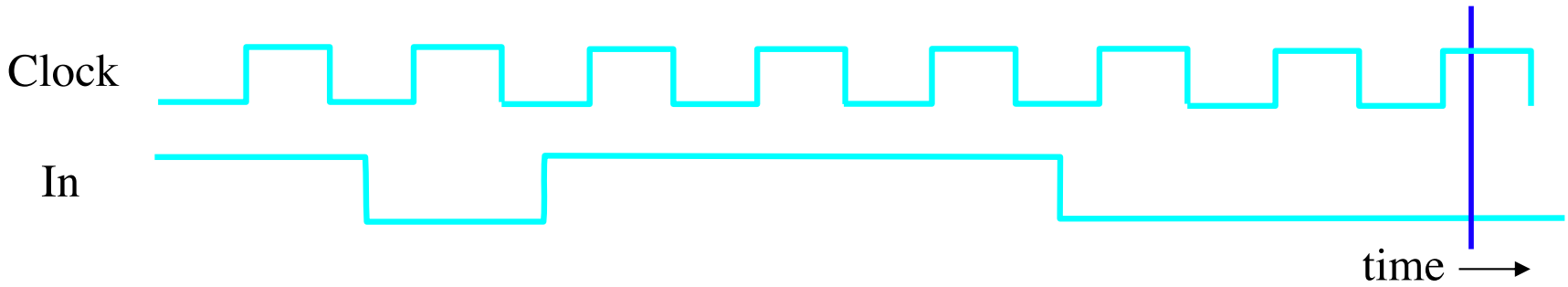
Shift Register Simulation



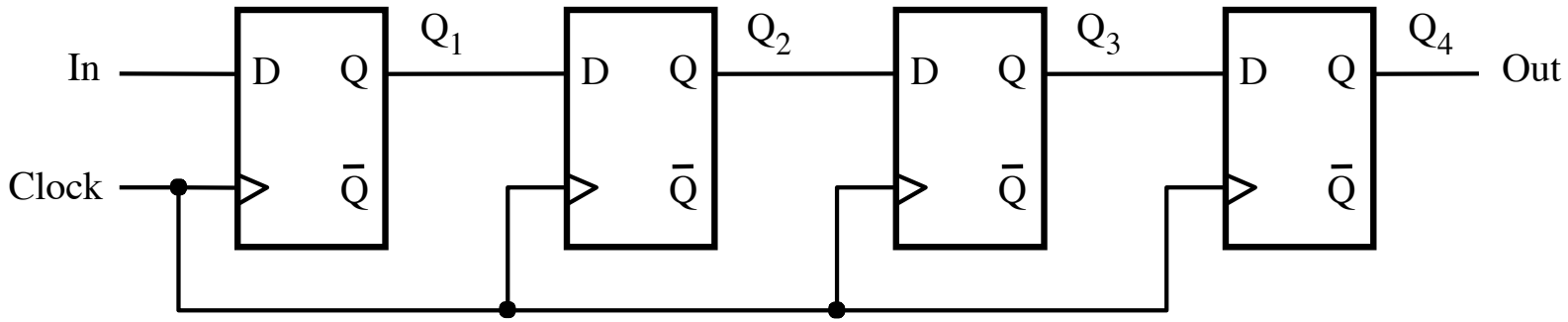


Shift Register Simulation





A simple shift register



(a) Circuit

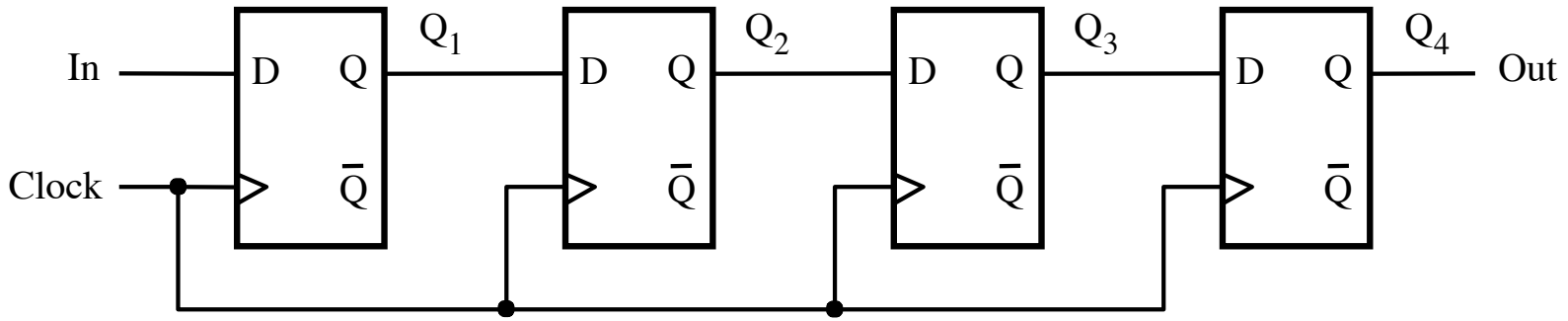
time

	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

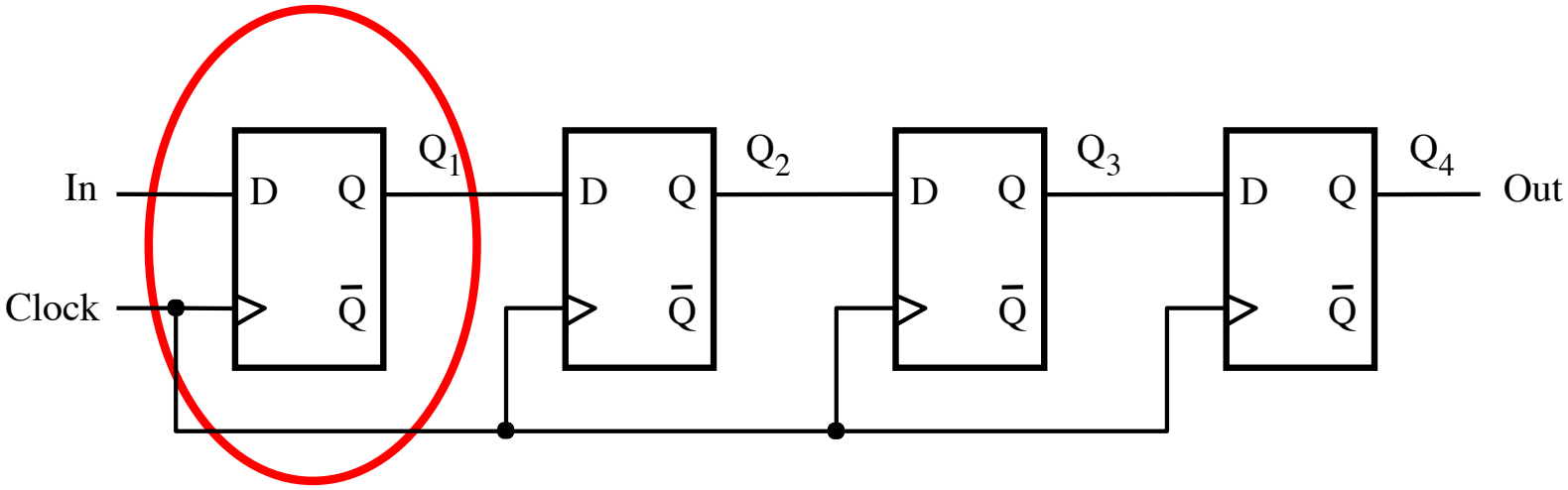
time

	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

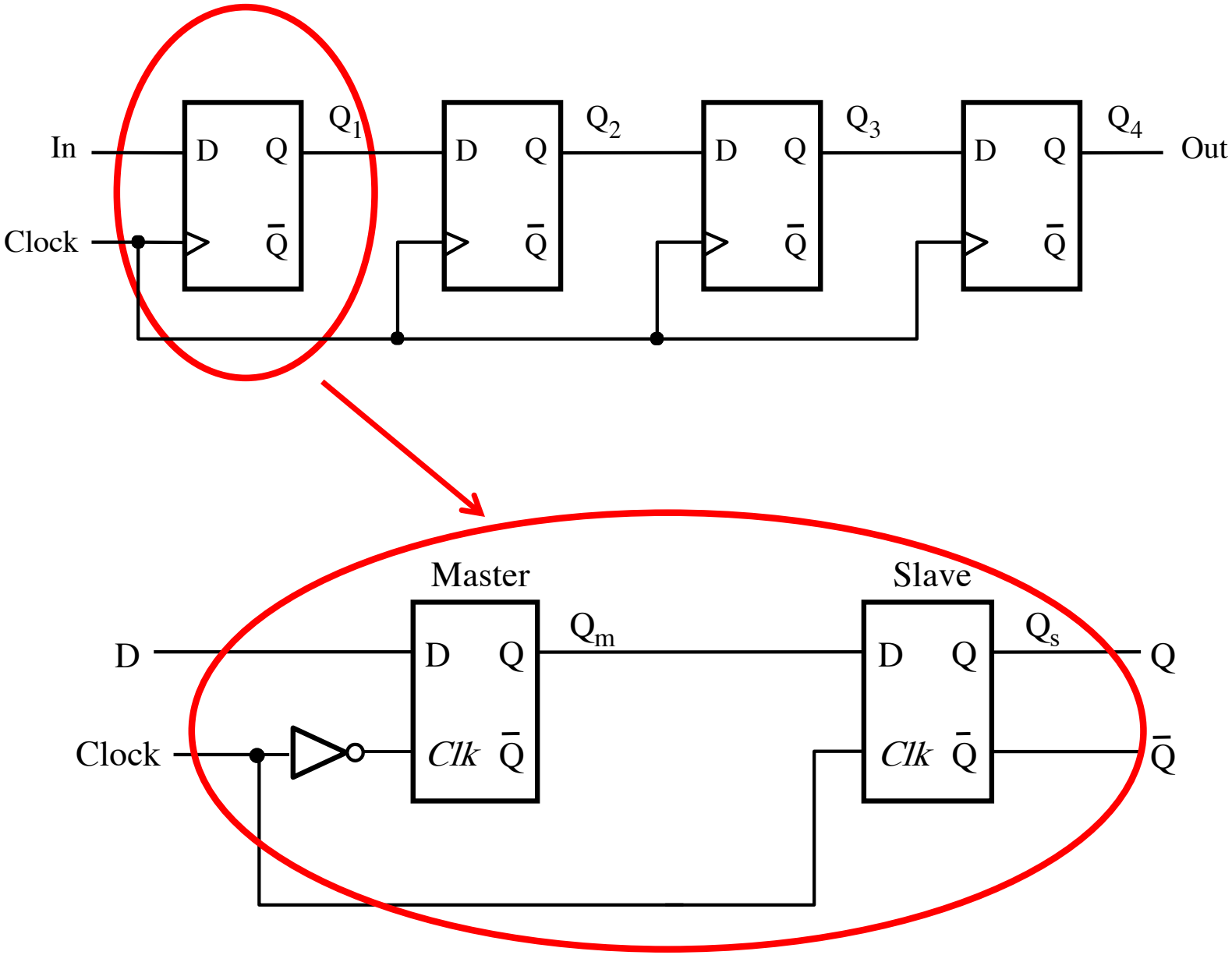
[Figure 5.17 from the textbook]

A simple shift register

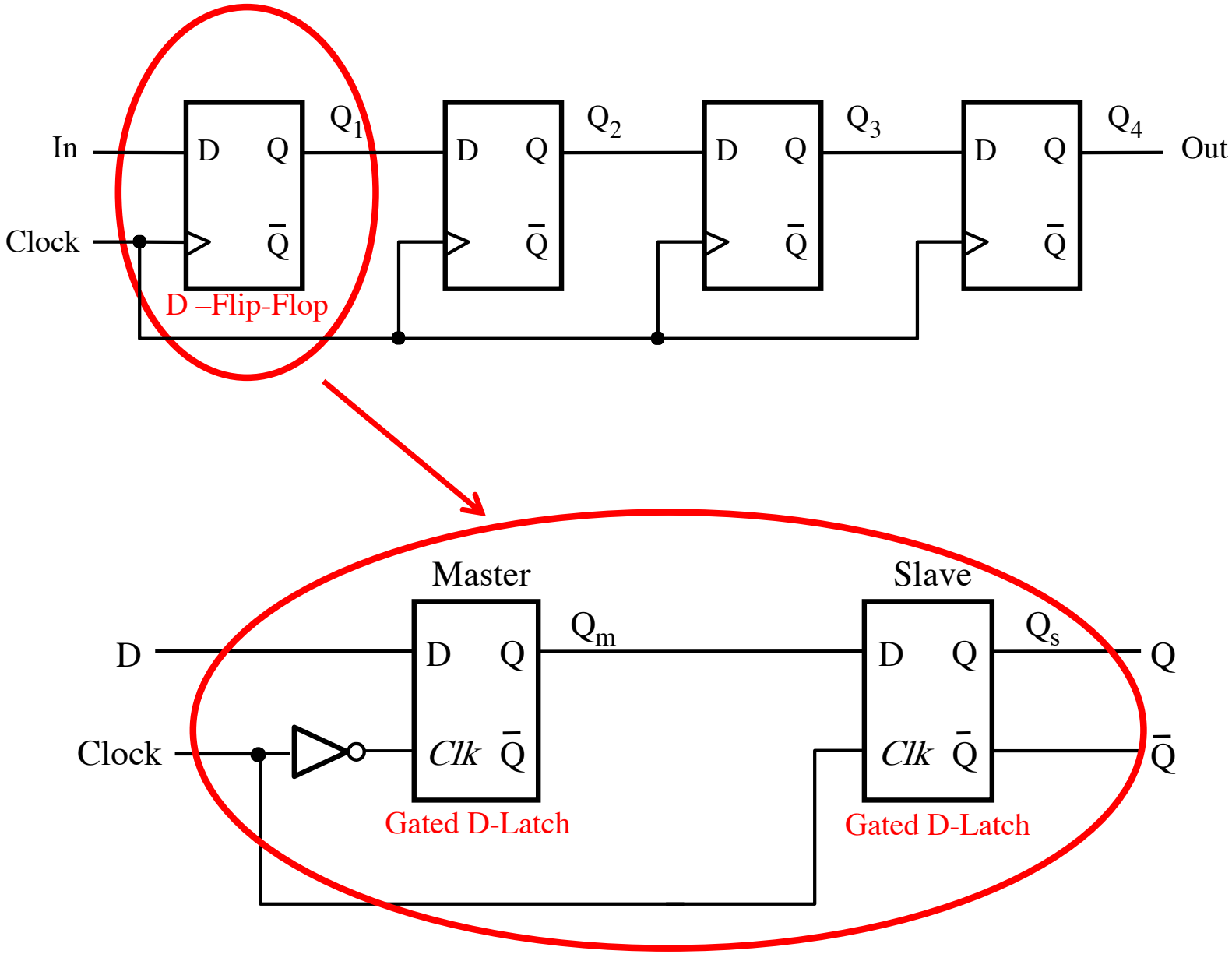


Positive-edge-triggered
D Flip-Flop

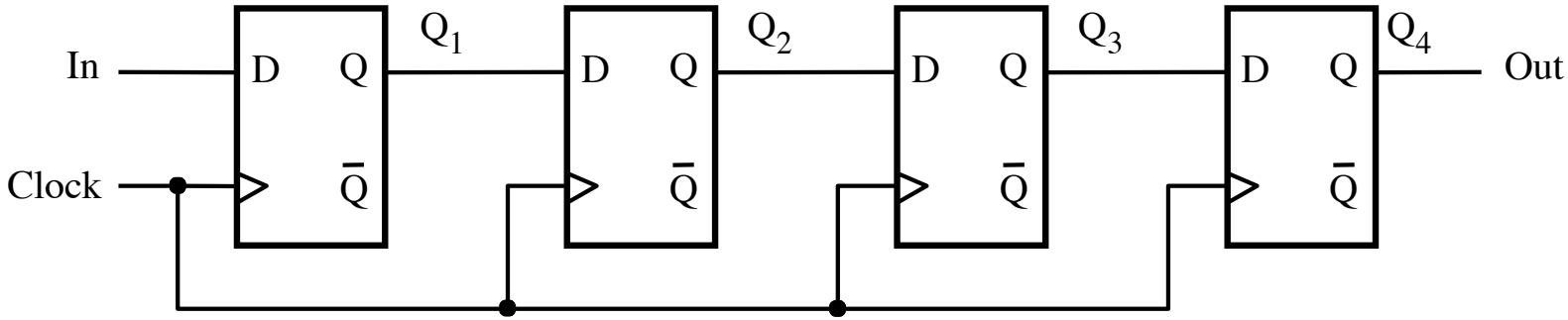
A simple shift register



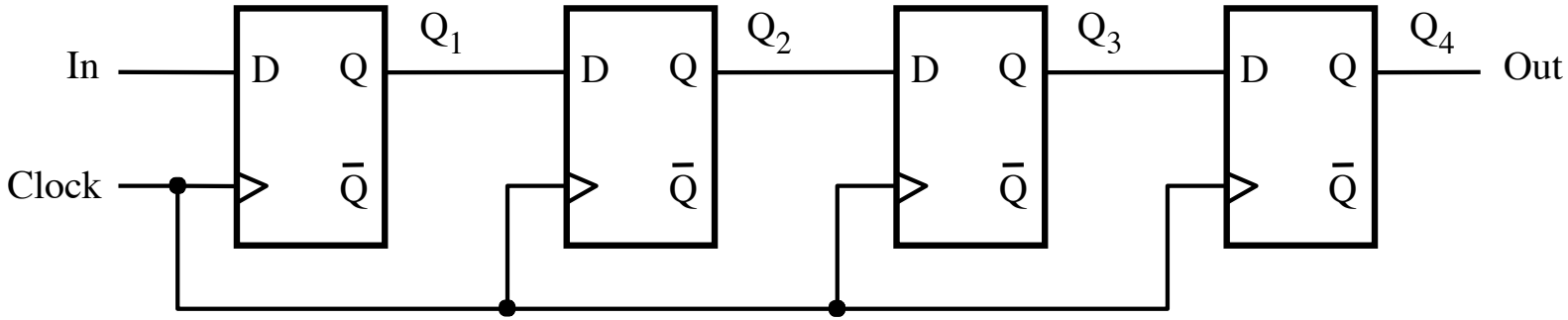
A simple shift register

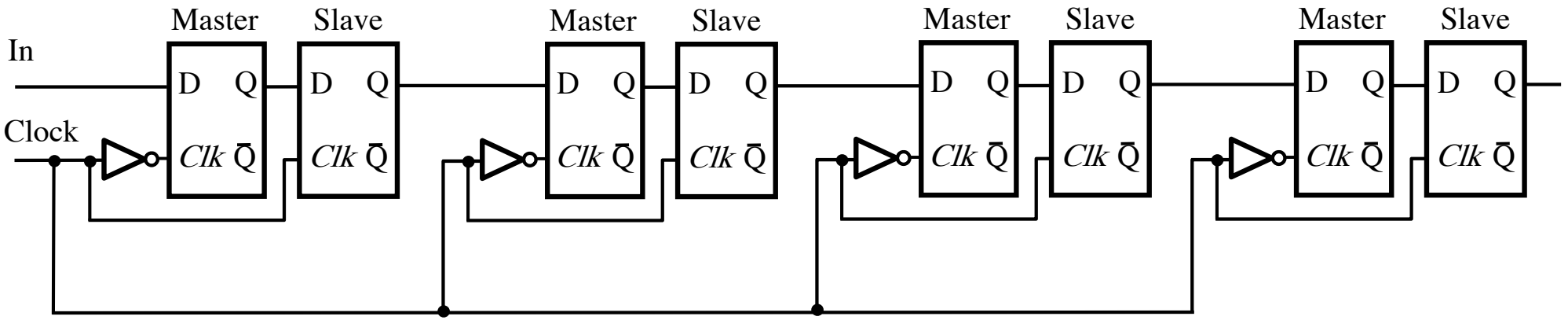


A simple shift register

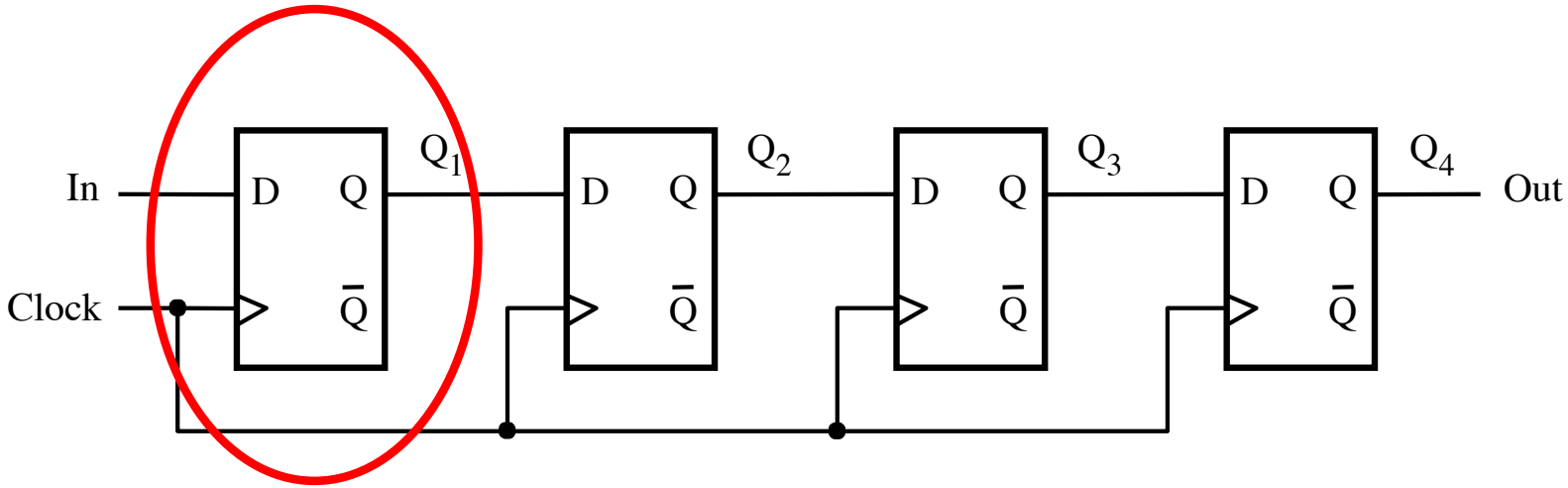


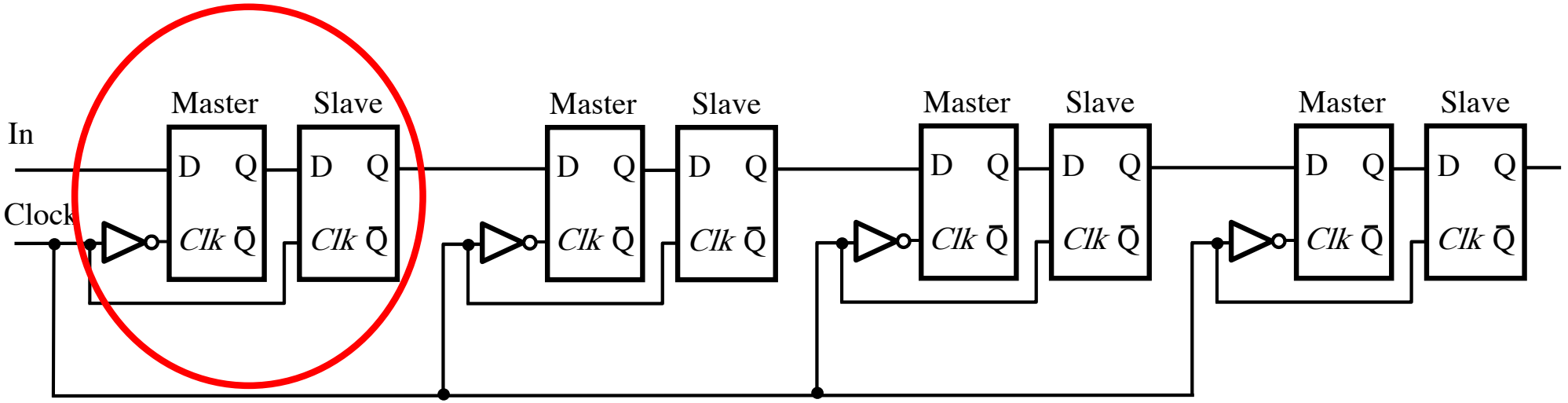
A simple shift register



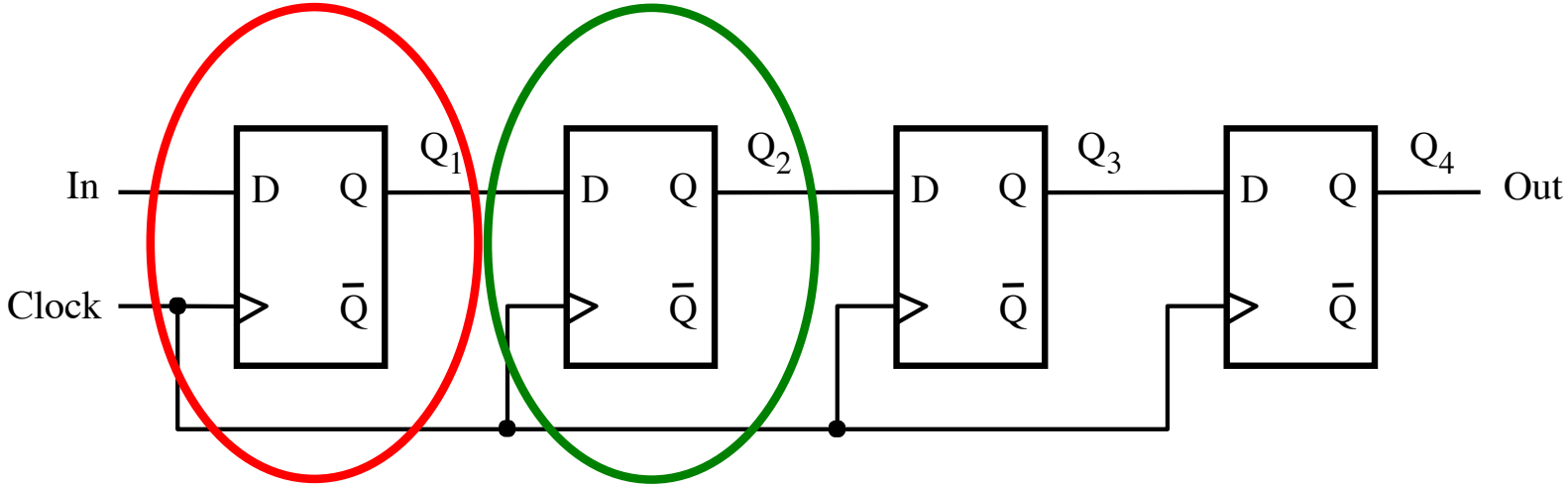


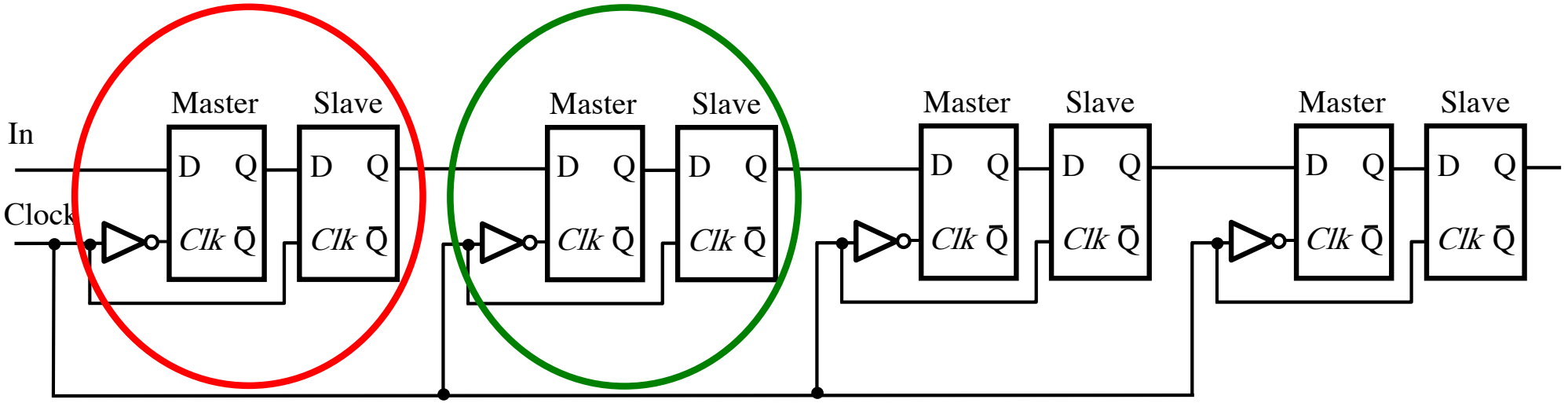
A simple shift register



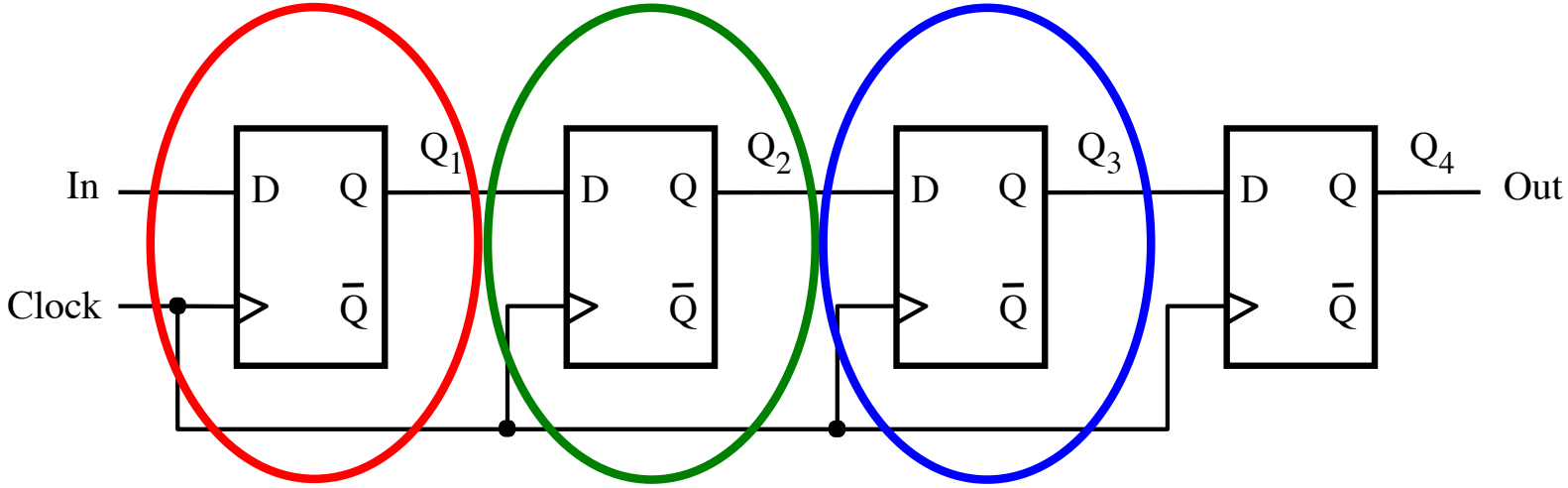


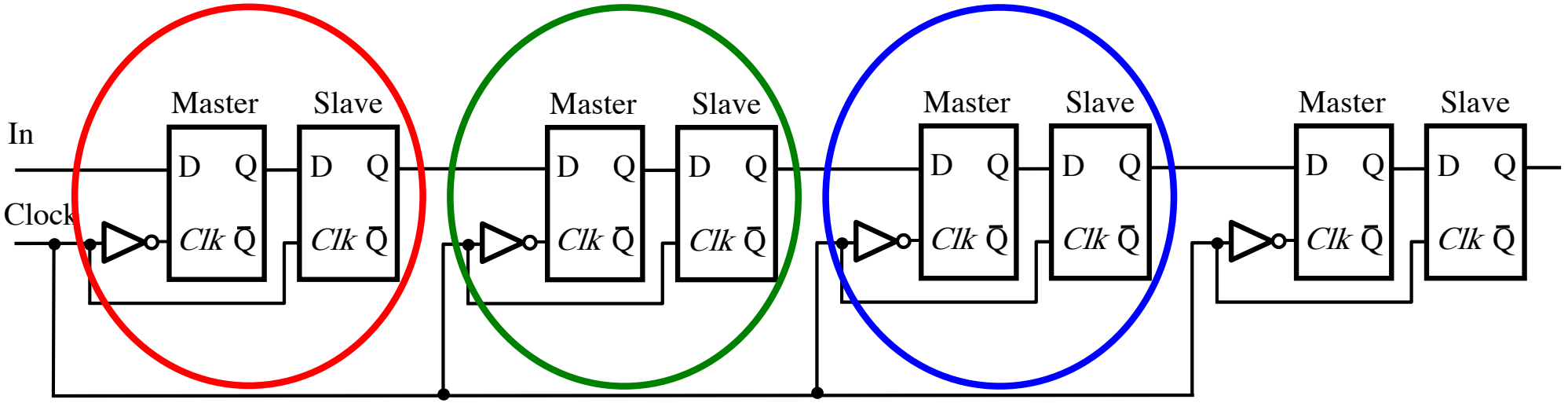
A simple shift register



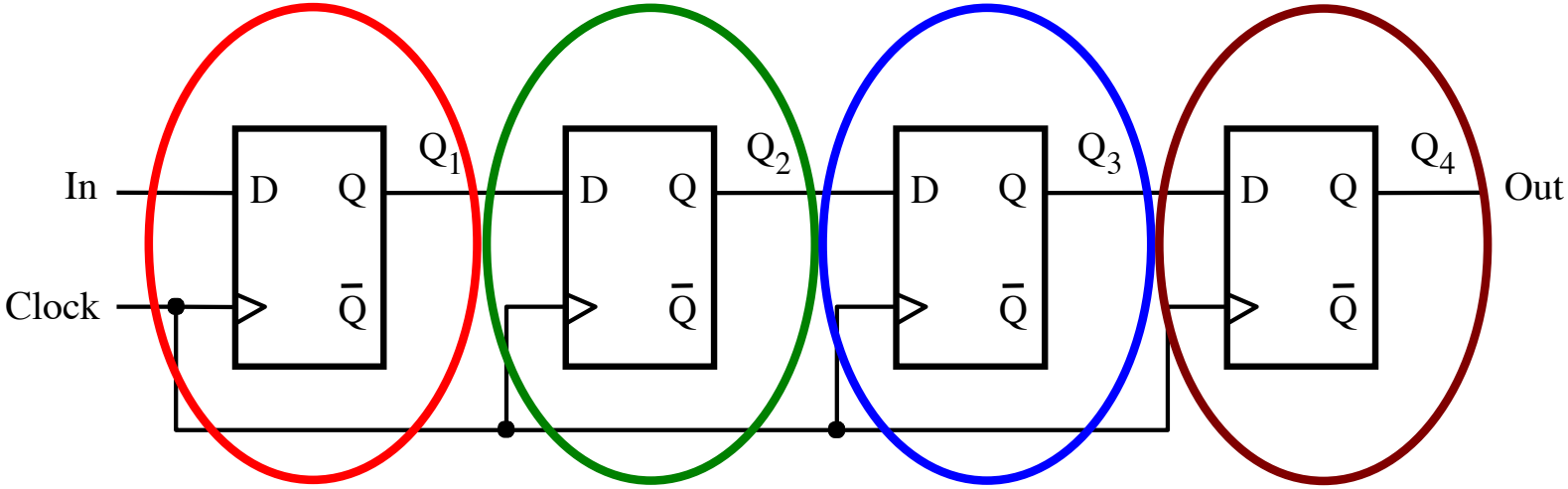


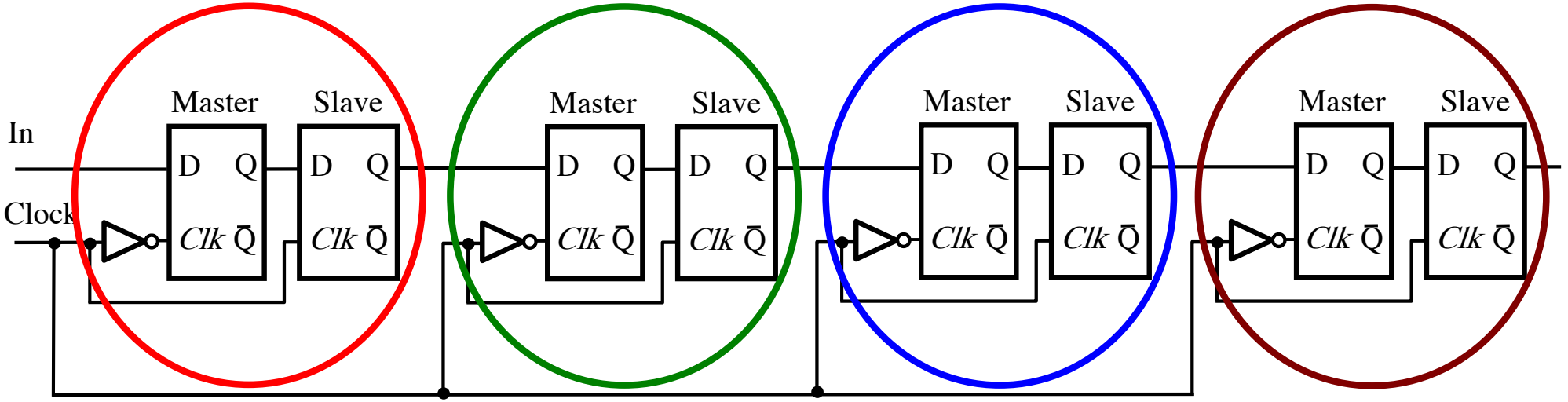
A simple shift register



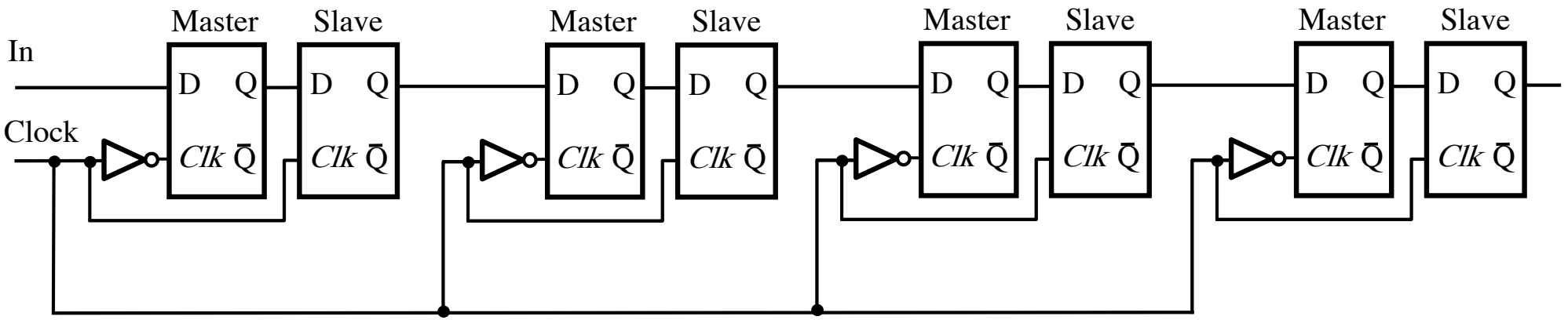


A simple shift register

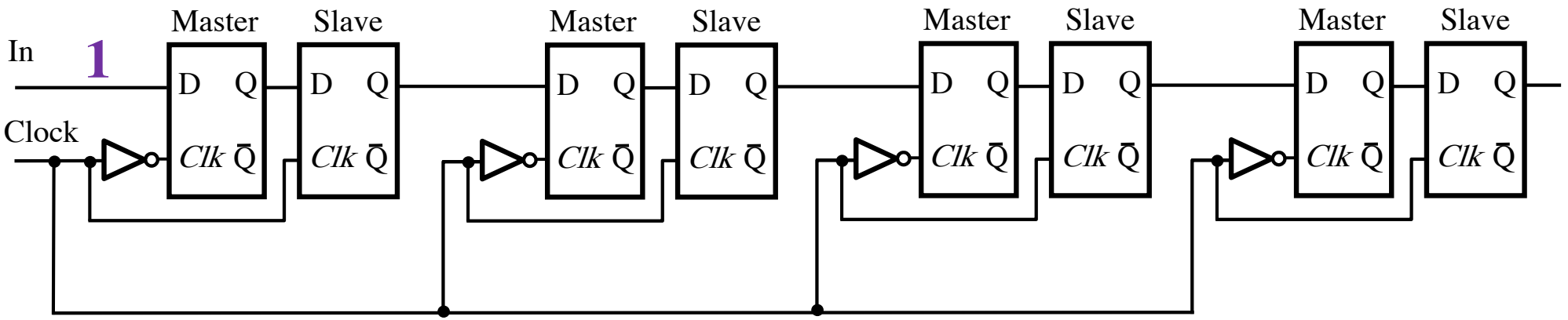




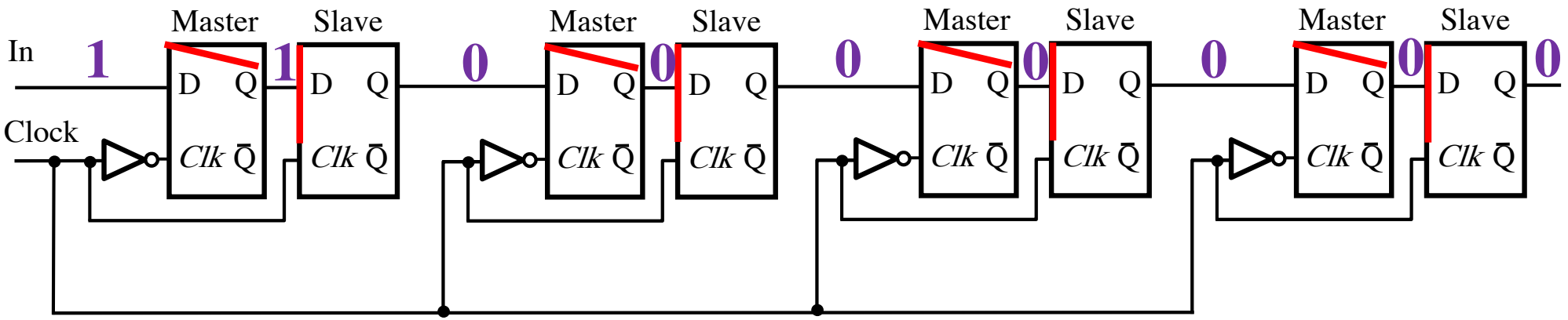
Simulating a shift register

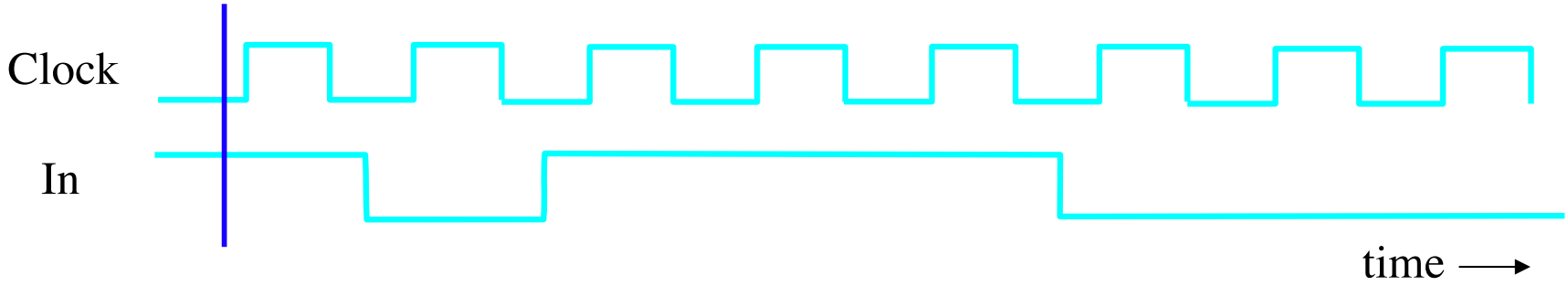


Simulating a shift register

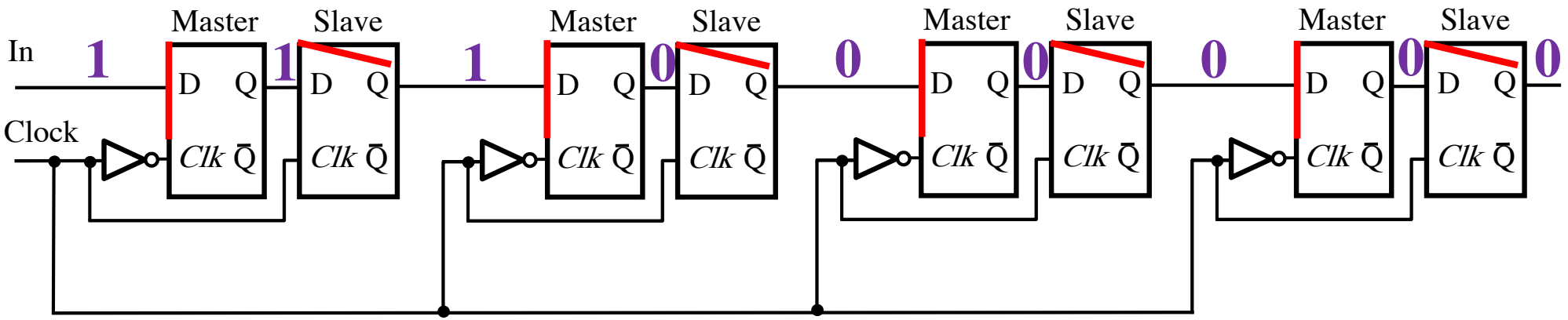


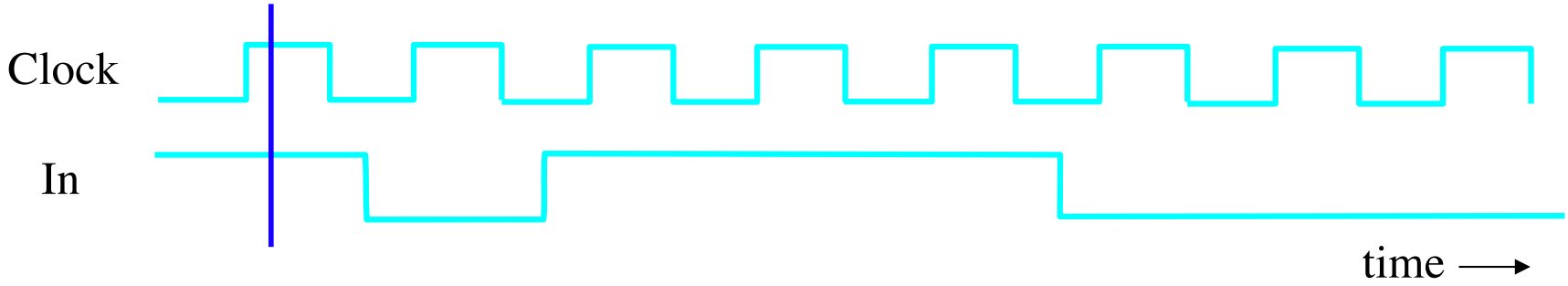
Simulating a shift register



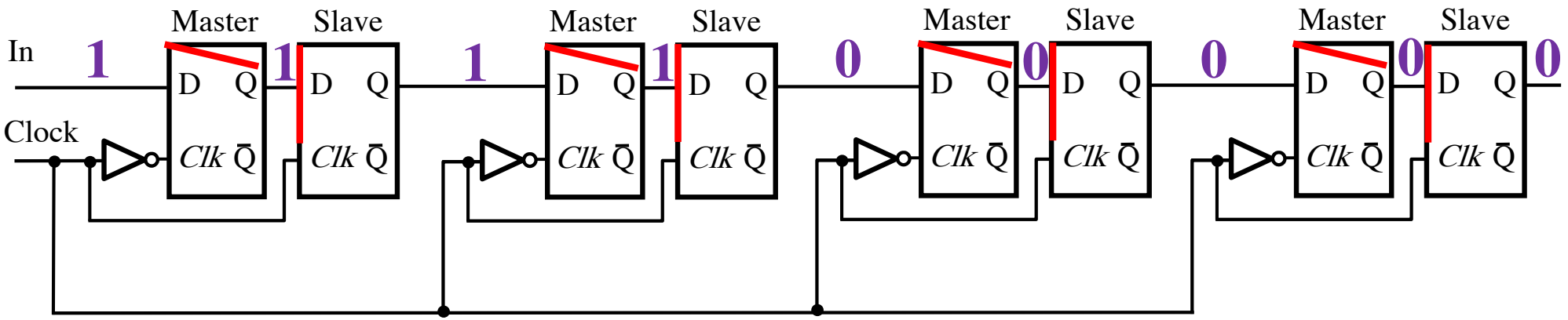


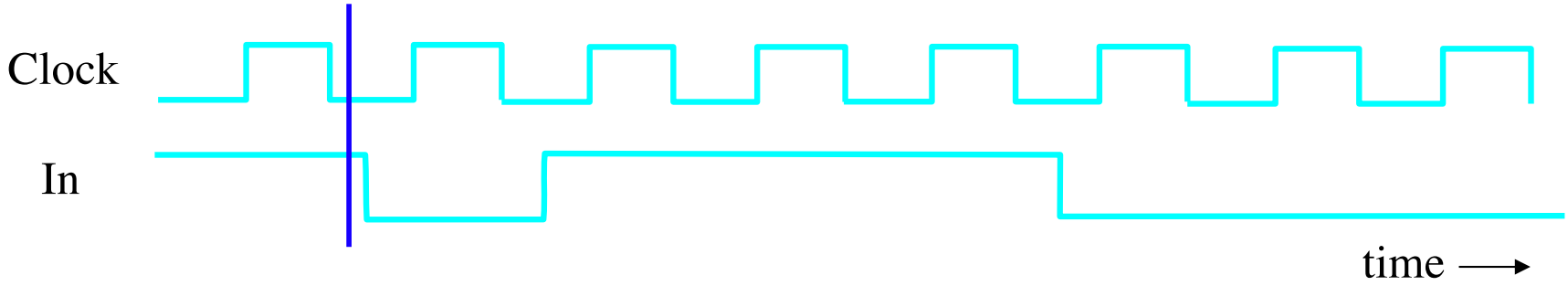
Simulating a shift register



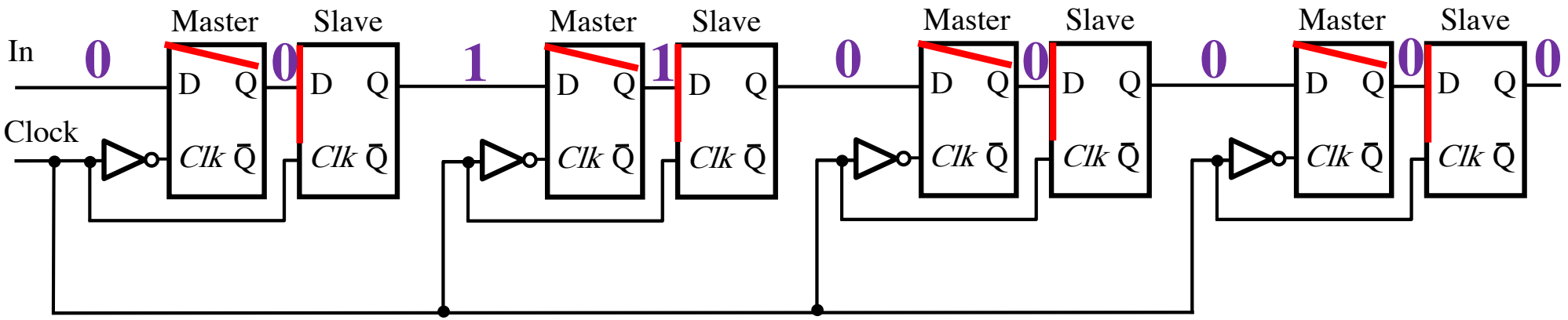


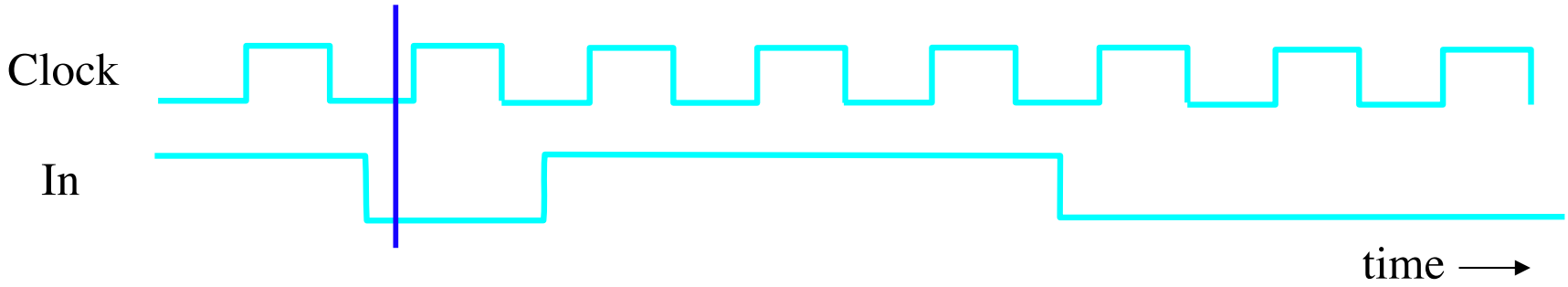
Simulating a shift register



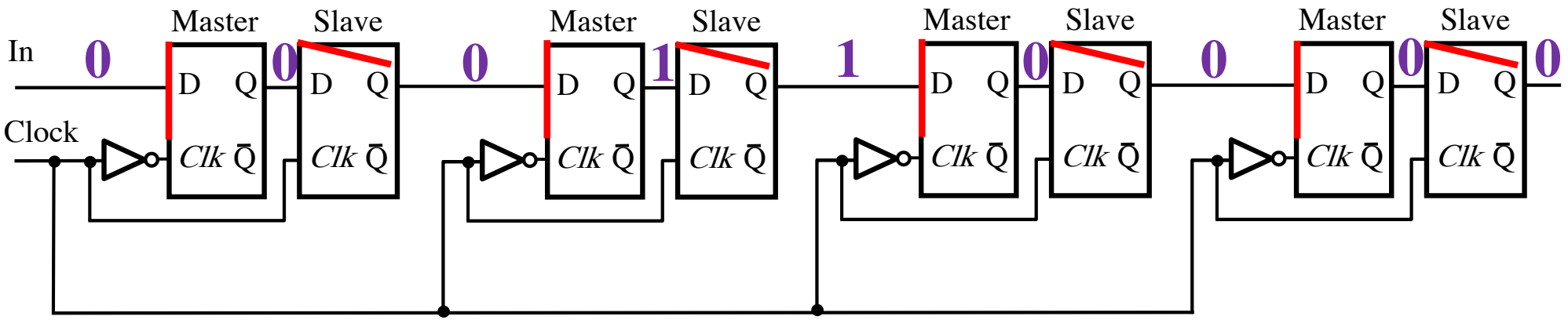


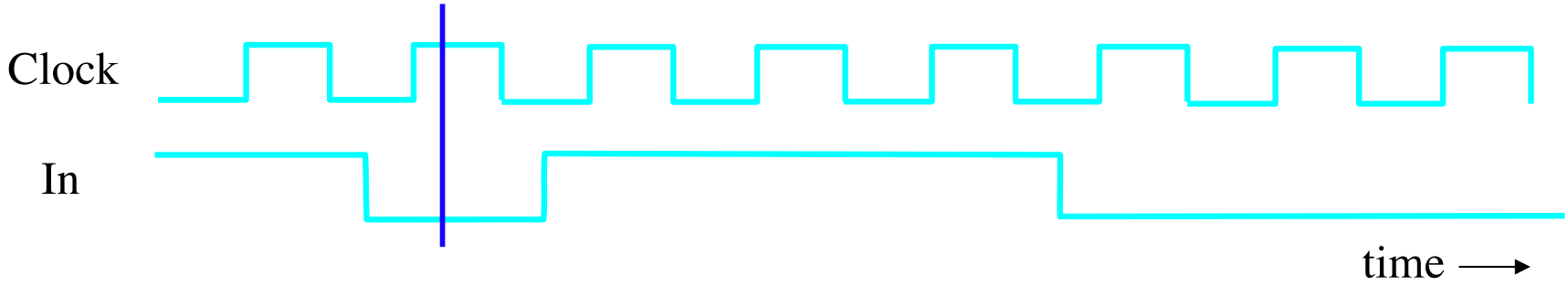
Simulating a shift register



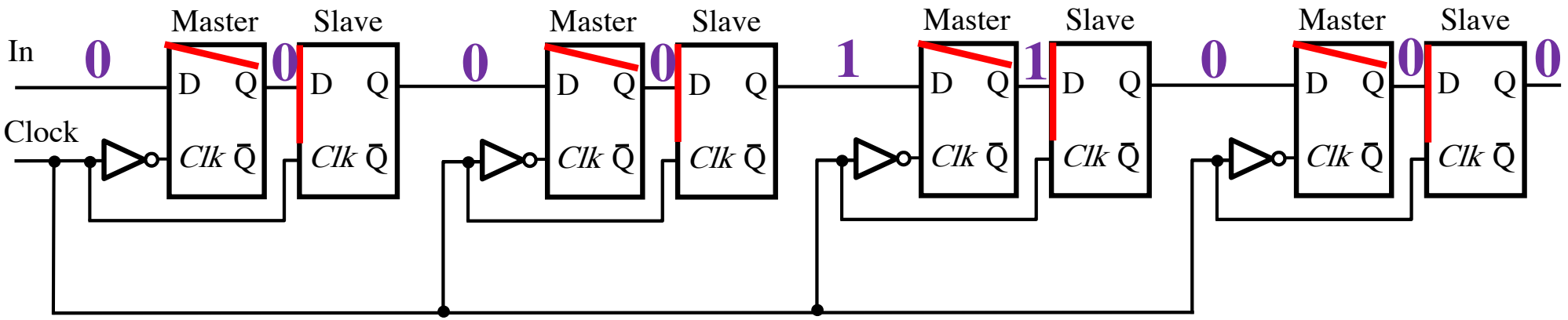


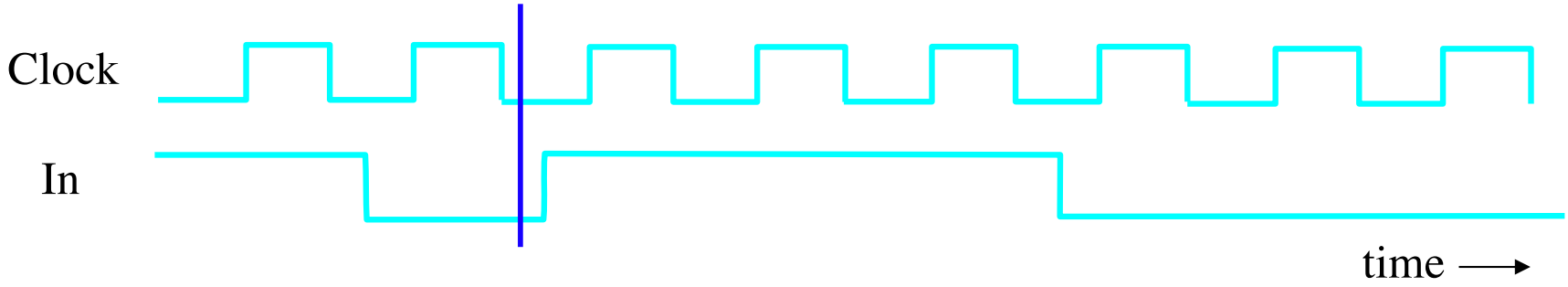
Simulating a shift register



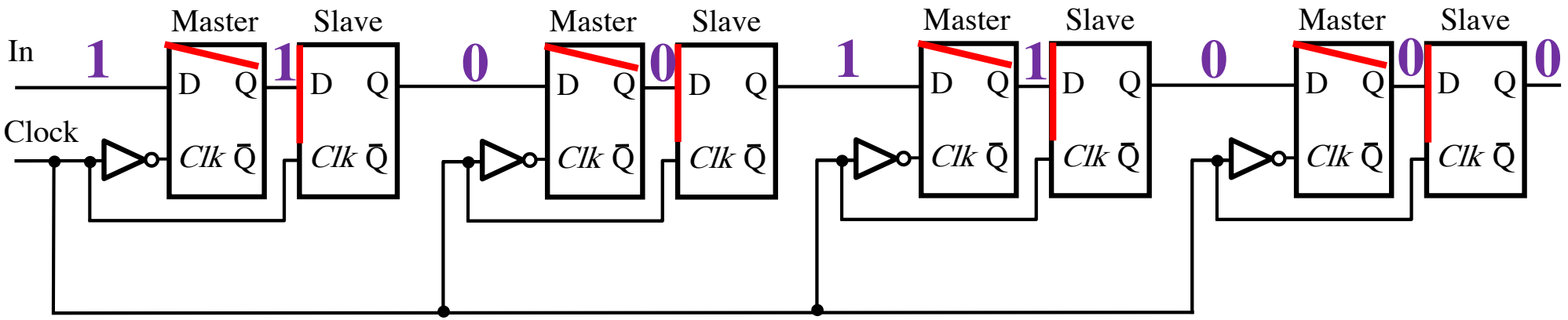


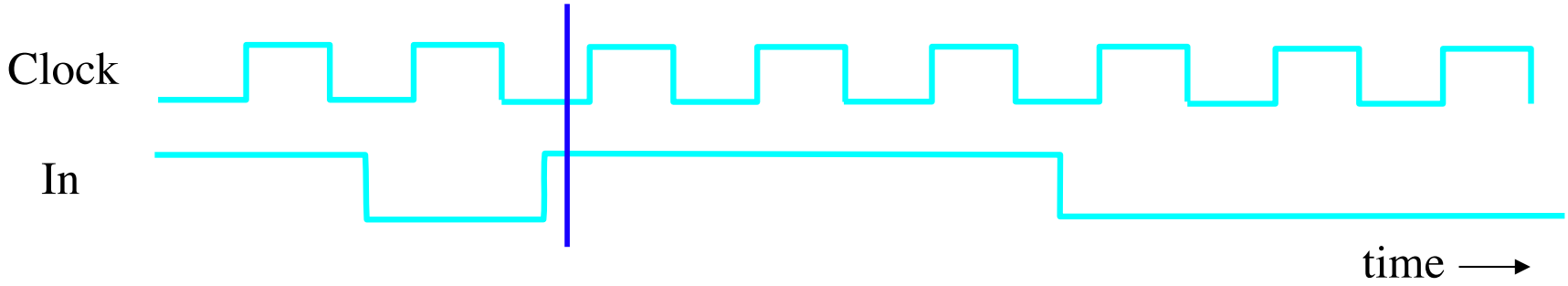
Simulating a shift register



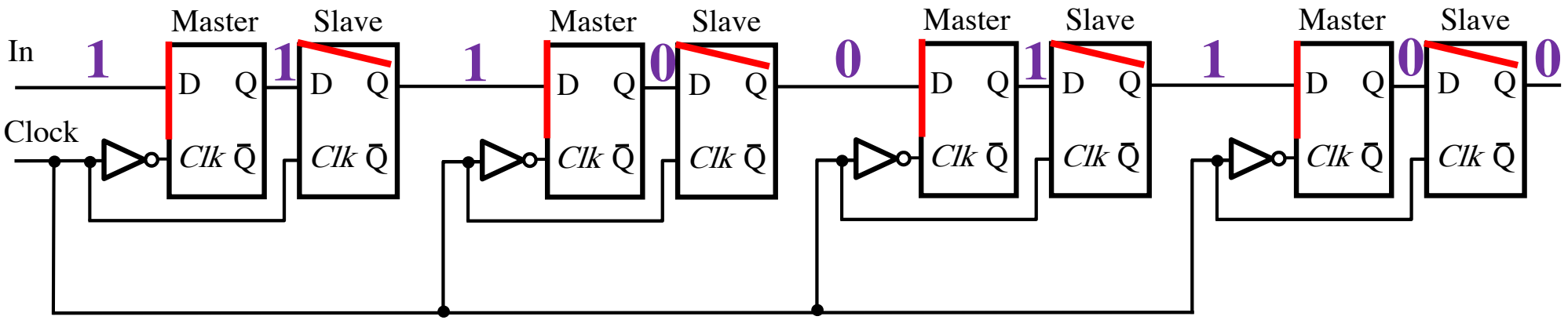


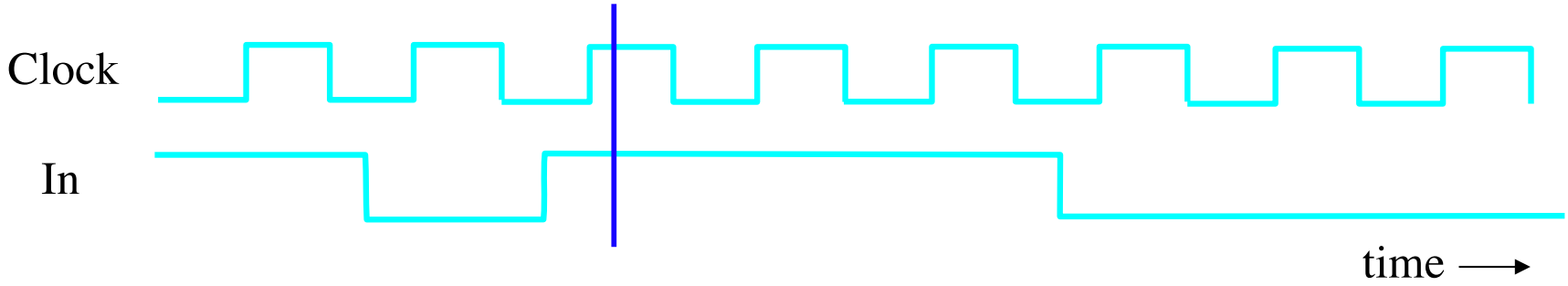
Simulating a shift register



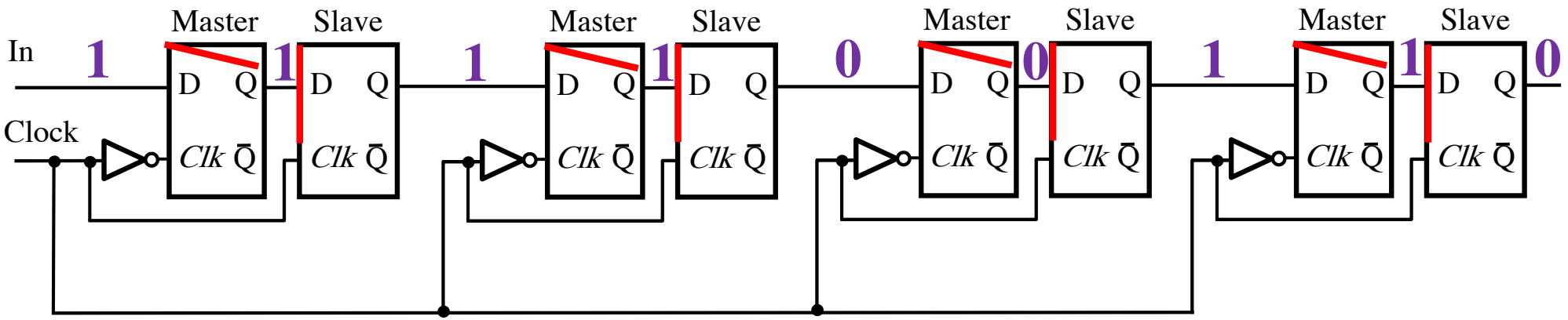


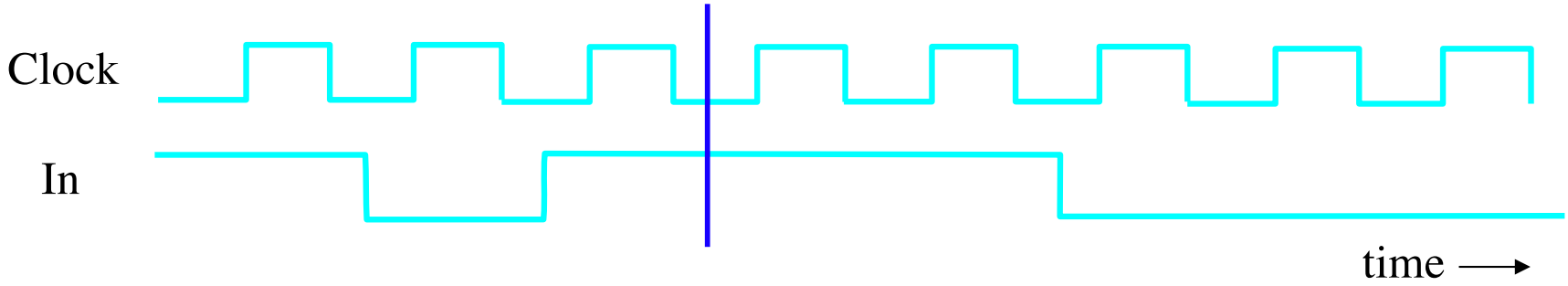
Simulating a shift register



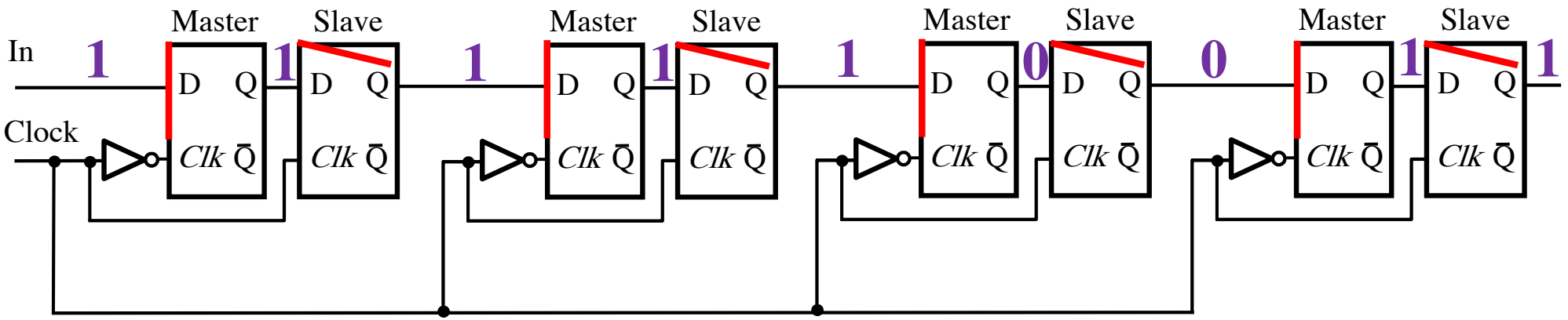


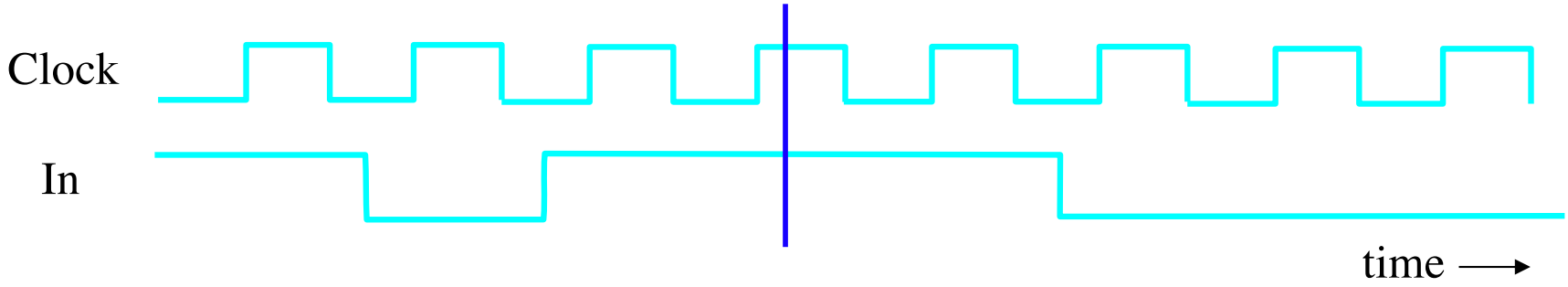
Simulating a shift register



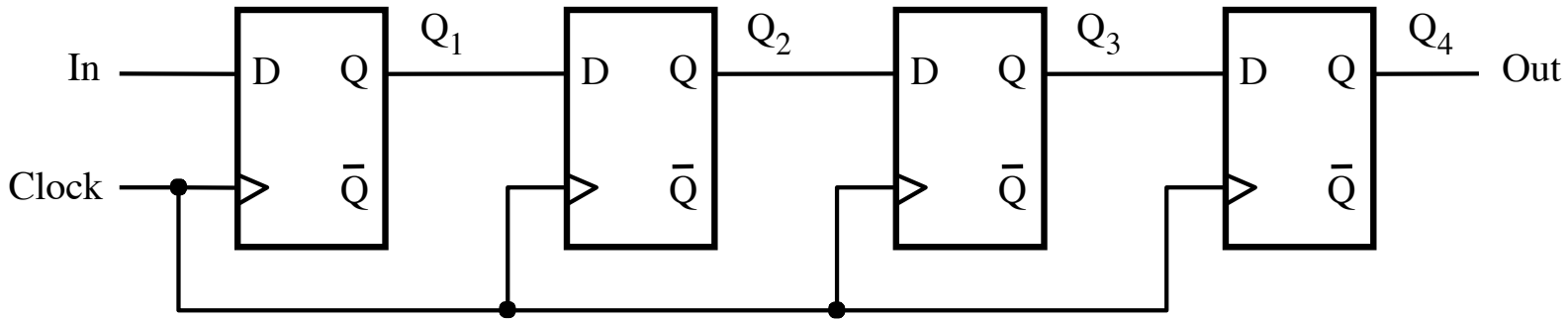


Simulating a shift register





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

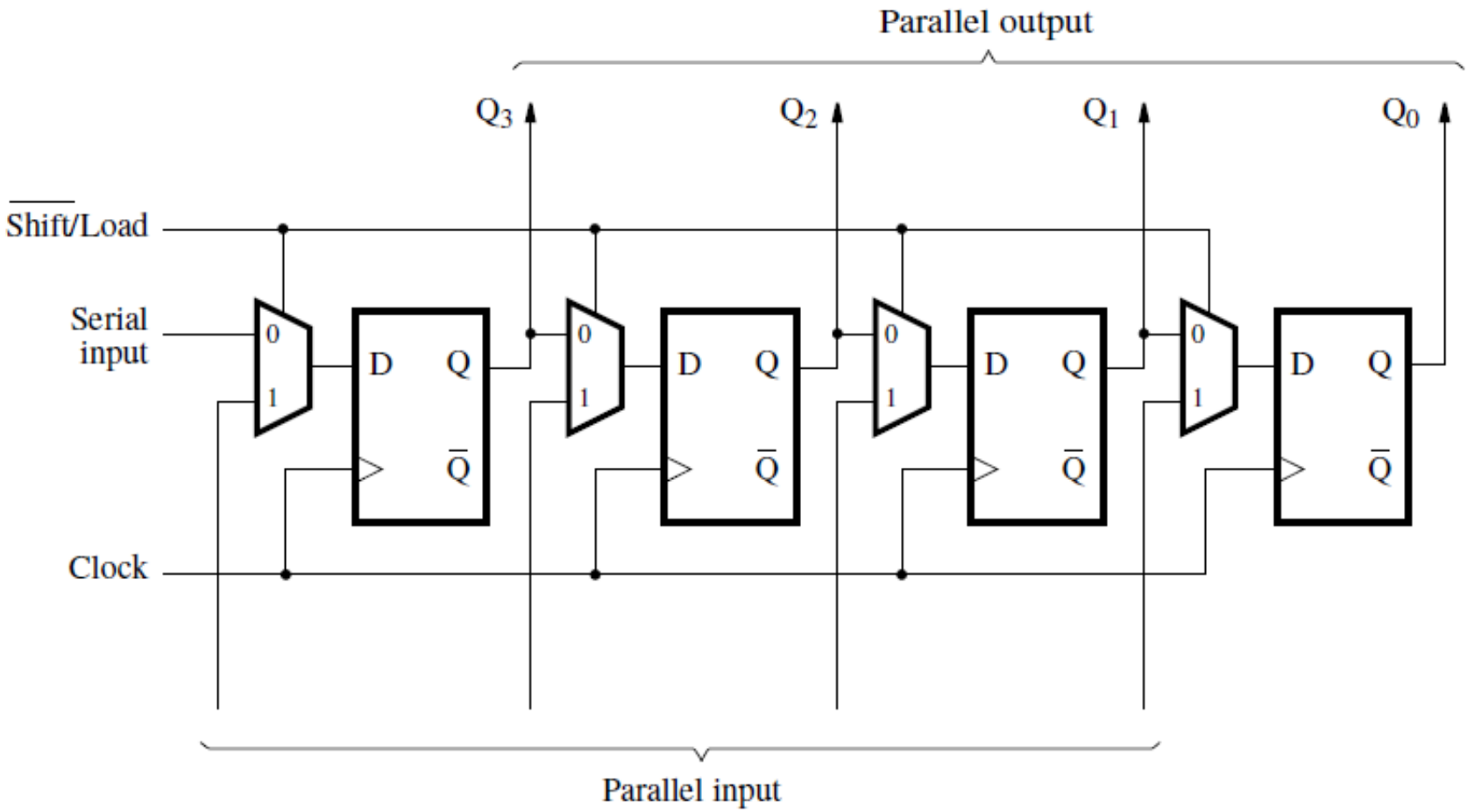
(b) A sample sequence

This simulation goes only up to here

[Figure 5.17 from the textbook]

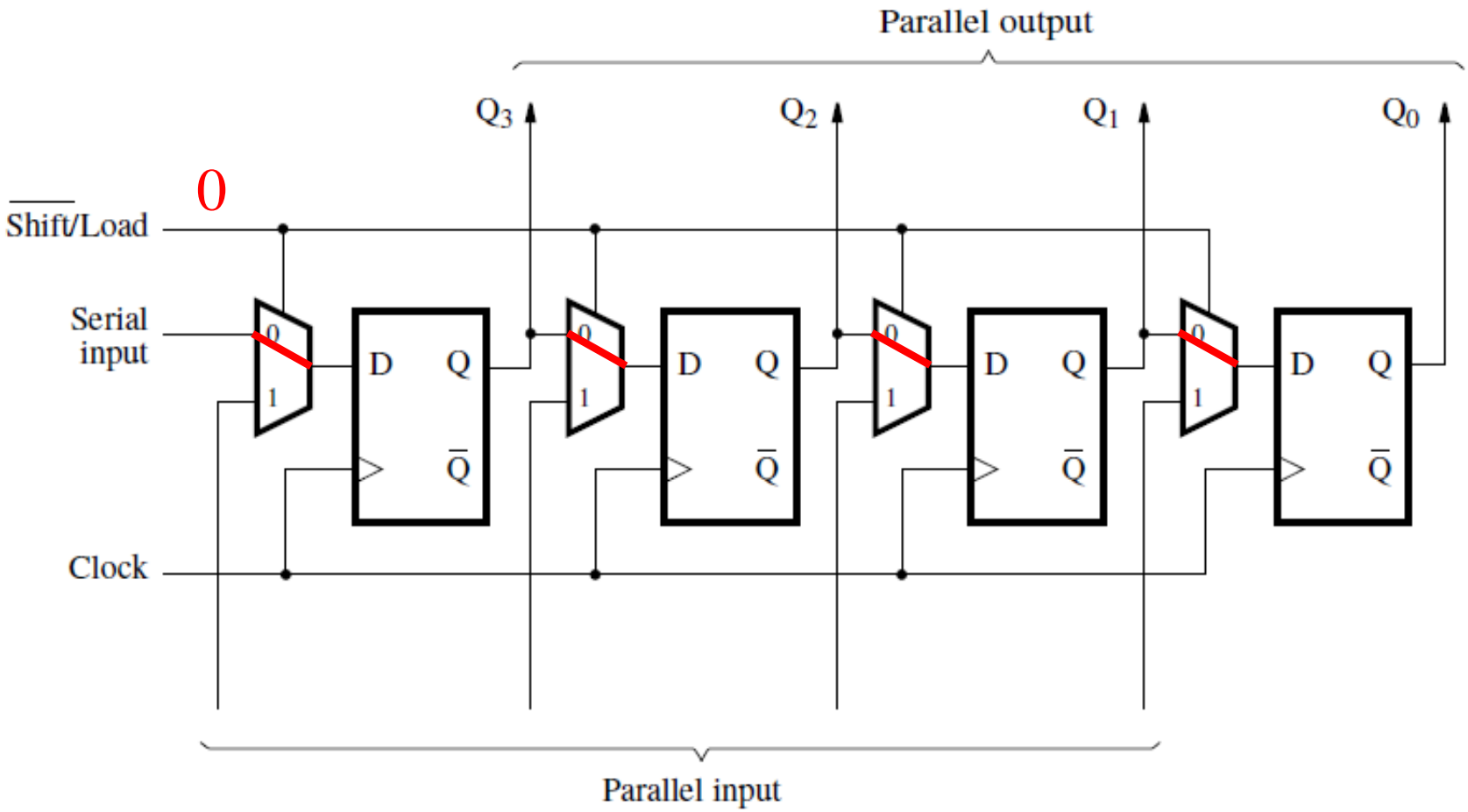
Parallel-Access Shift Register

Parallel-access shift register



[Figure 5.18 from the textbook]

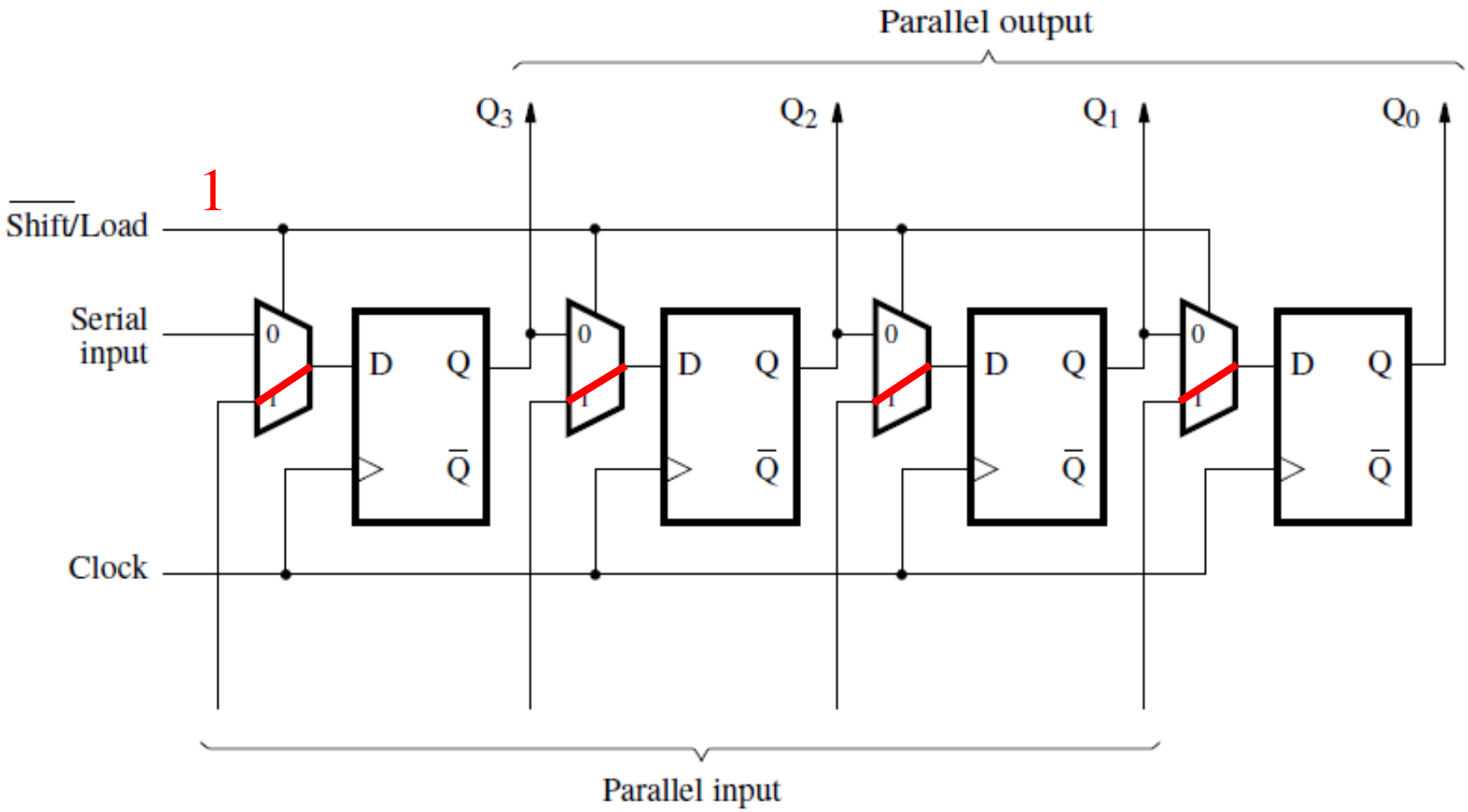
Parallel-access shift register



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

[Figure 5.18 from the textbook]

Parallel-access shift register

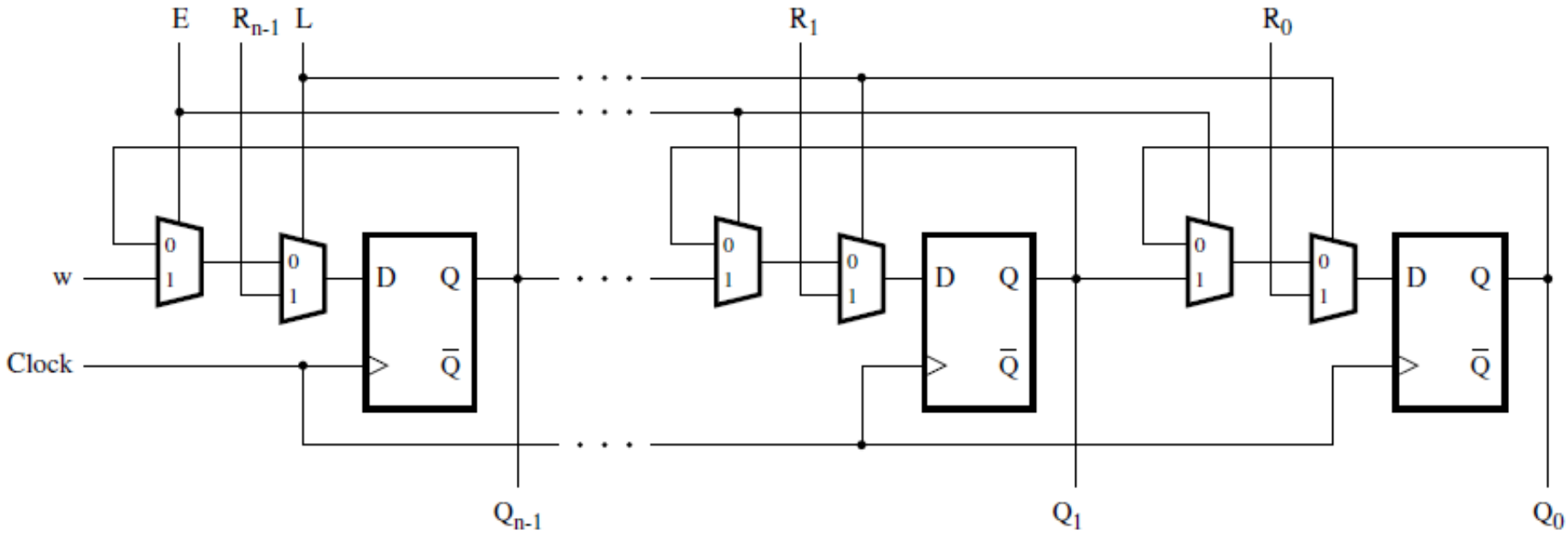


When Load=1, this behaves like a parallel-access register.

[Figure 5.18 from the textbook]

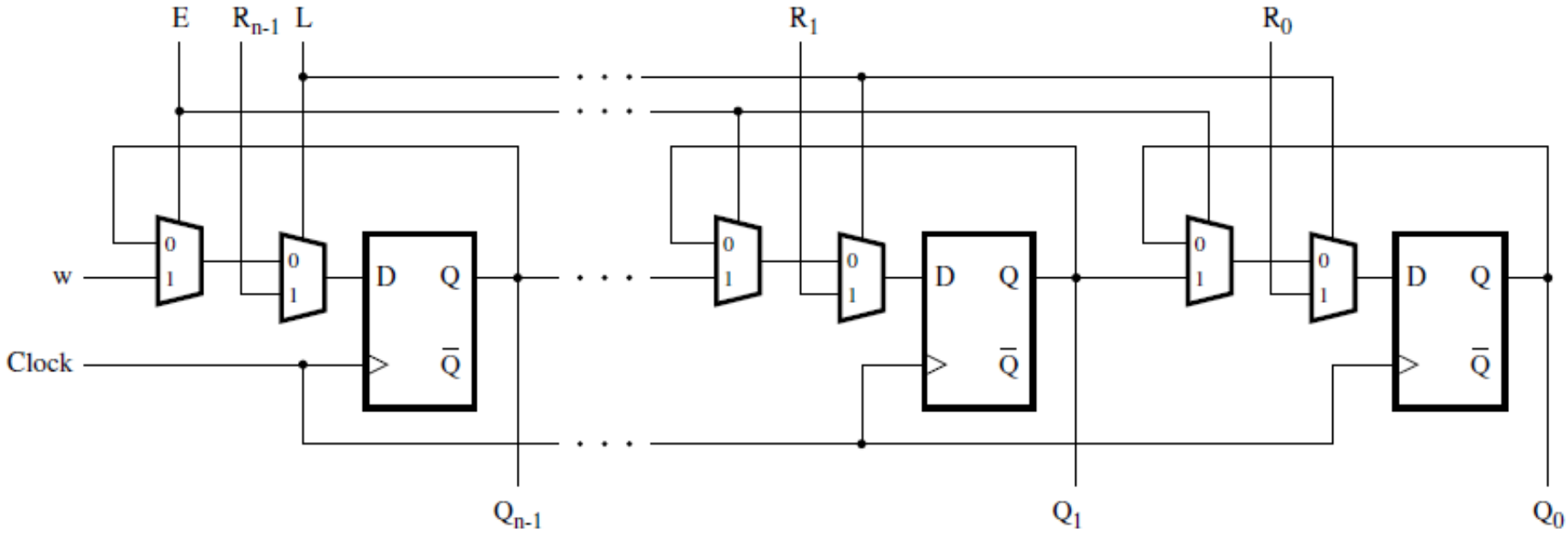
Shift Register With Parallel Load and Enable

A shift register with parallel load and enable control inputs



[Figure 5.59 from the textbook]

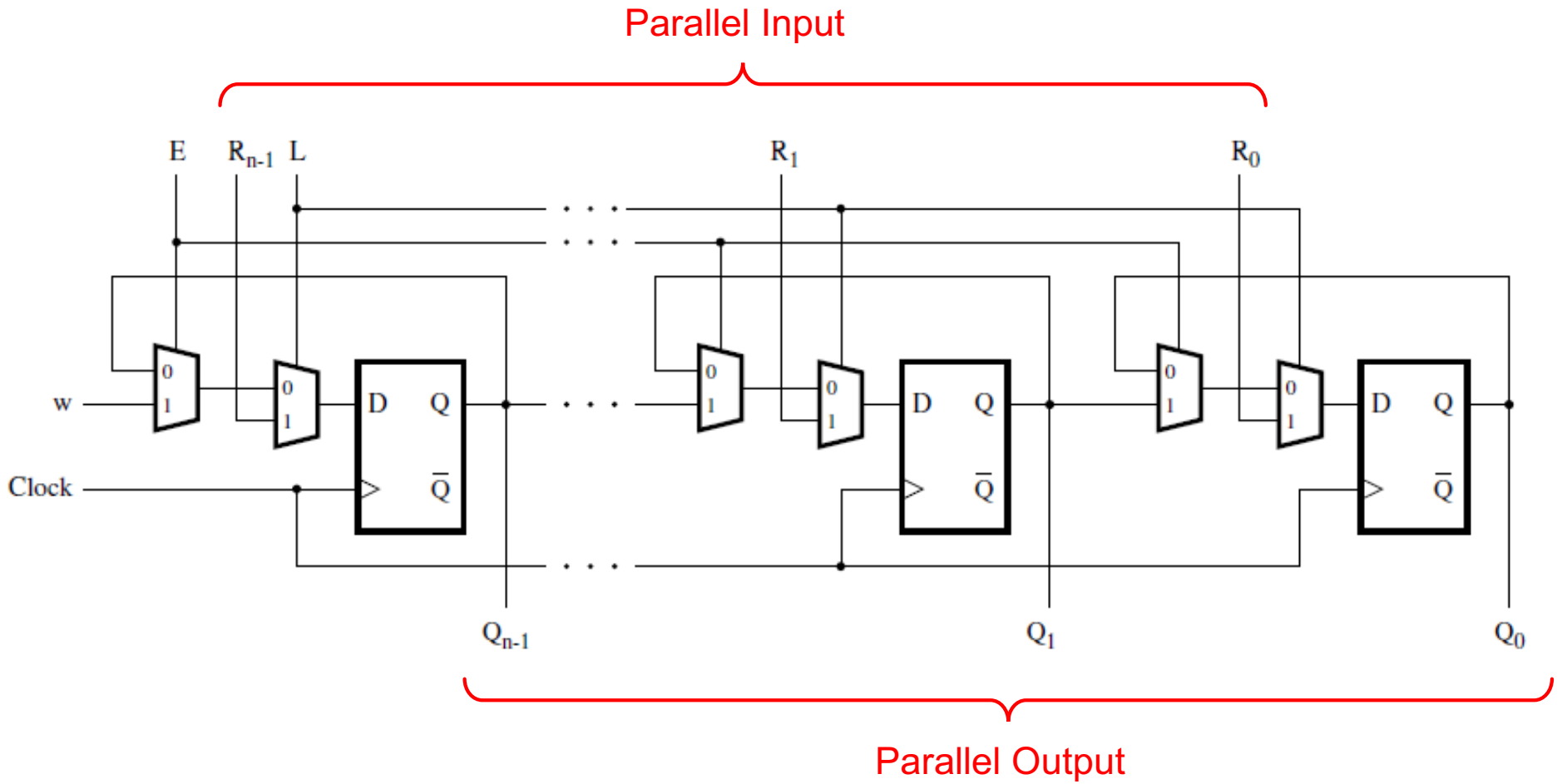
A shift register with parallel load and enable control inputs



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

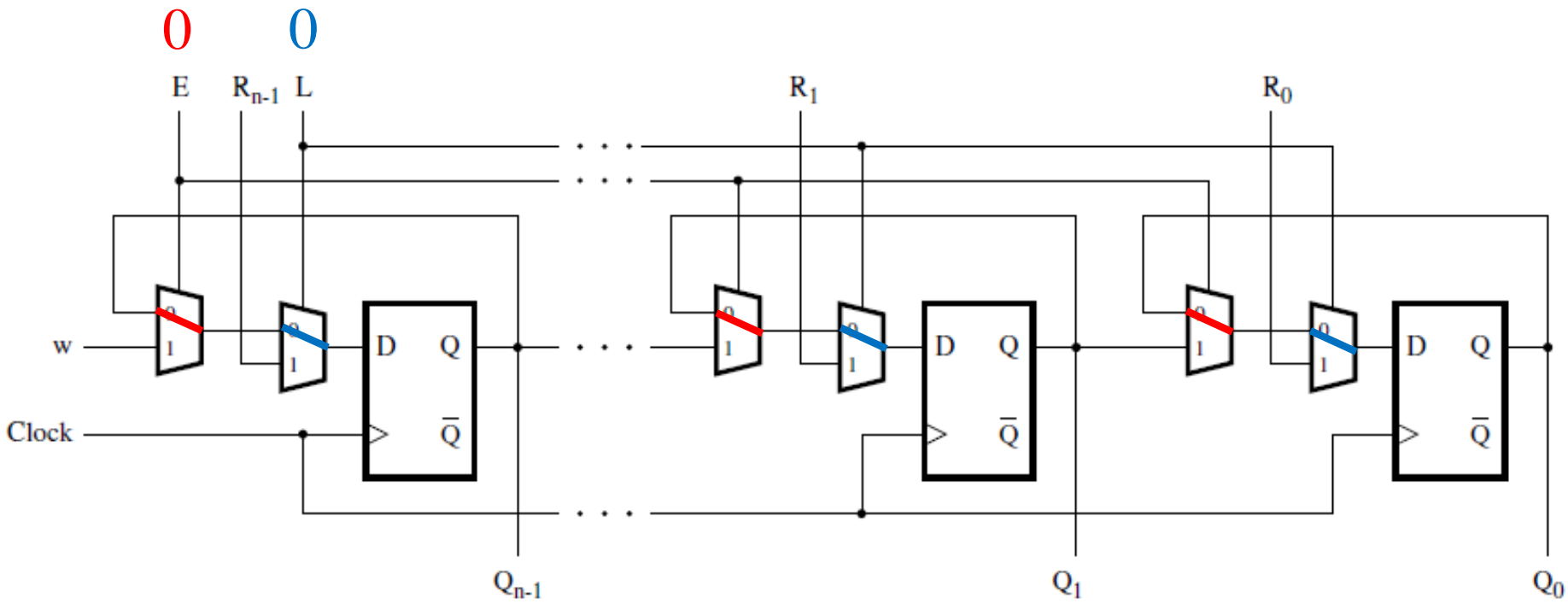
[Figure 5.59 from the textbook]

A shift register with parallel load and enable control inputs



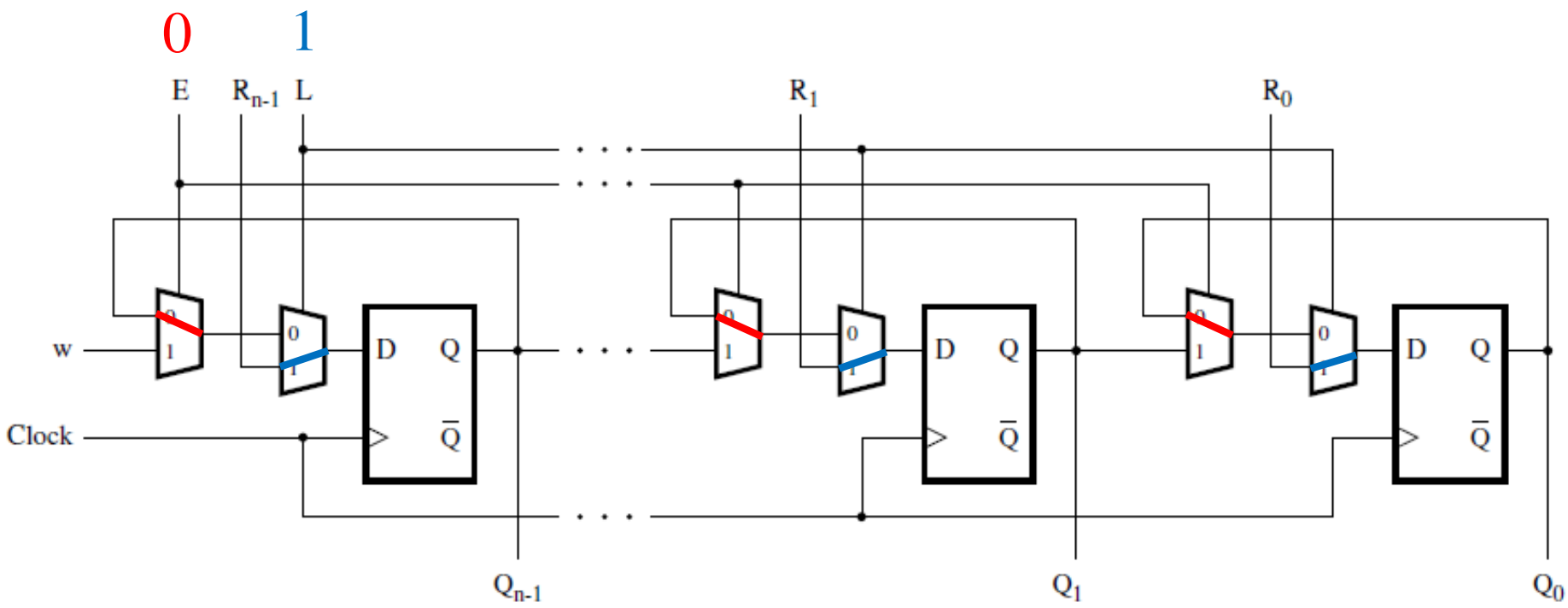
[Figure 5.59 from the textbook]

A shift register with parallel load and enable control inputs



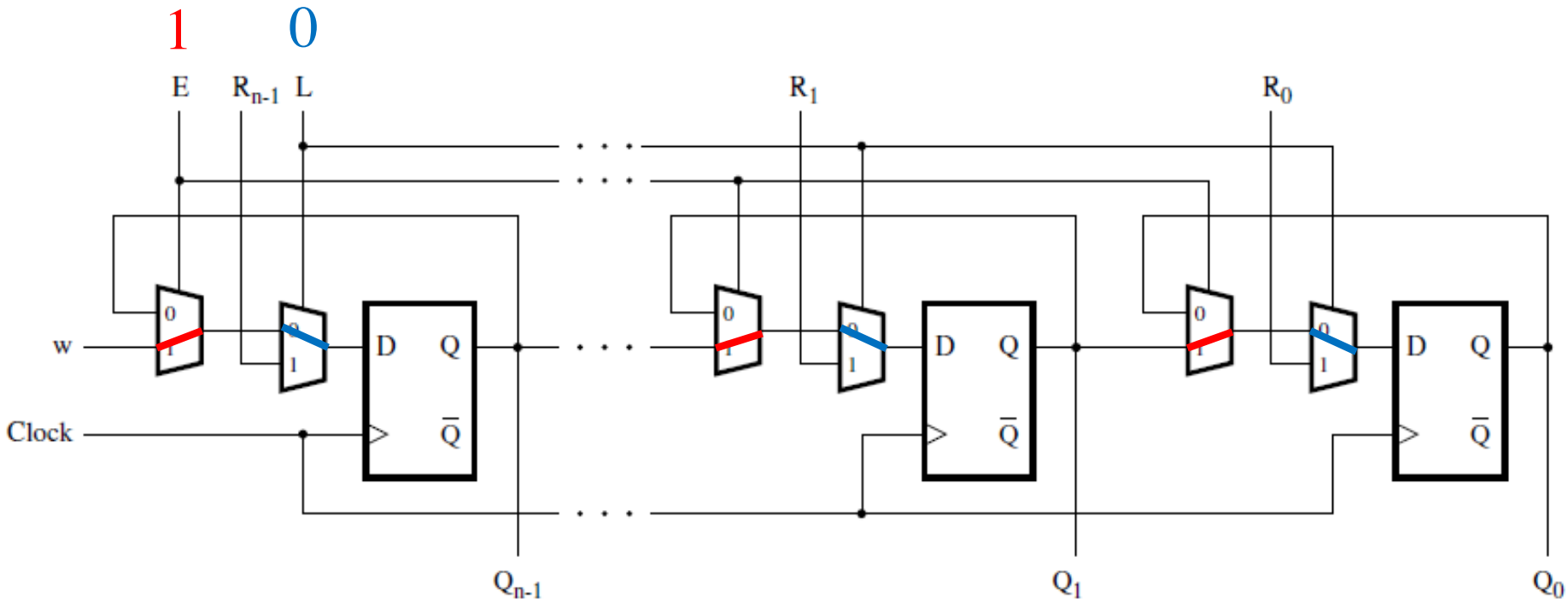
[Figure 5.59 from the textbook]

A shift register with parallel load and enable control inputs



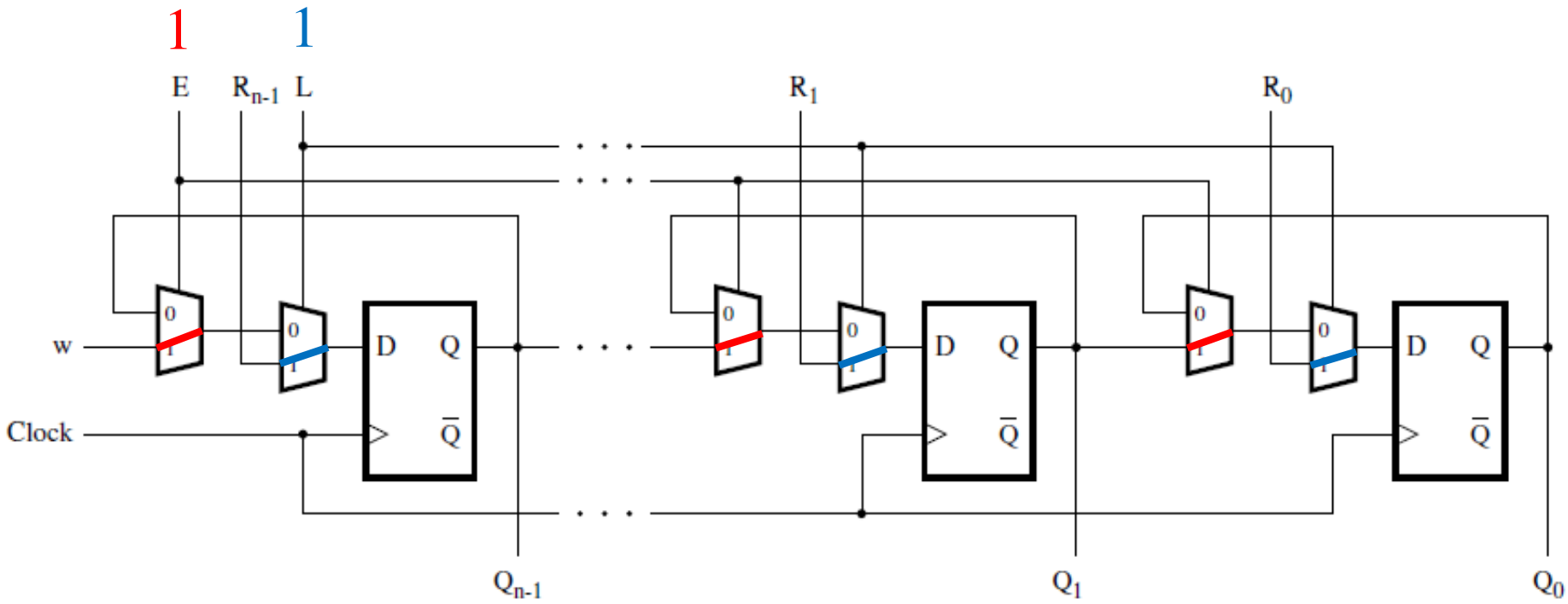
[Figure 5.59 from the textbook]

A shift register with parallel load and enable control inputs



[Figure 5.59 from the textbook]

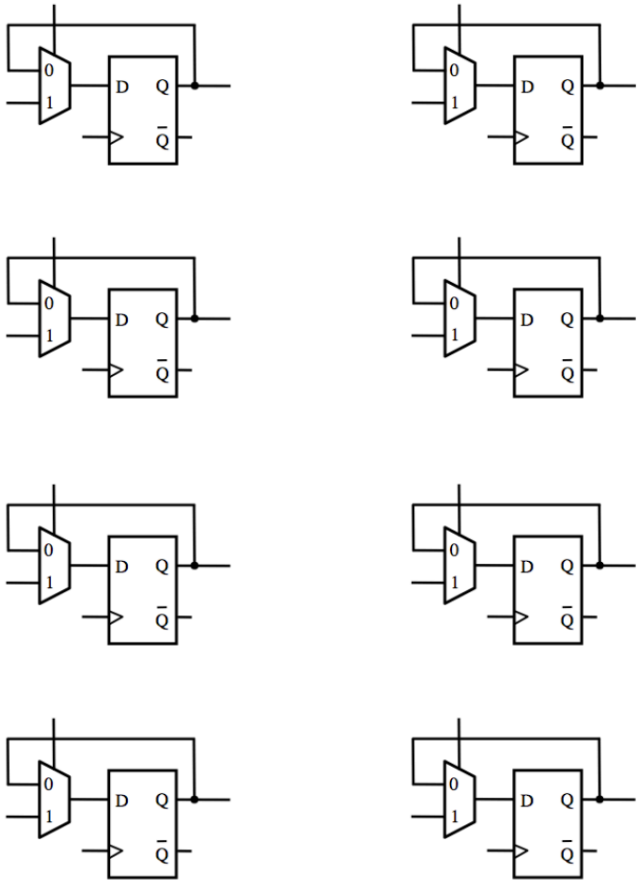
A shift register with parallel load and enable control inputs

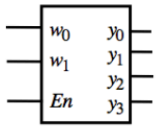


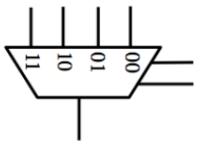
[Figure 5.59 from the textbook]

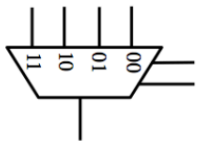
Register File

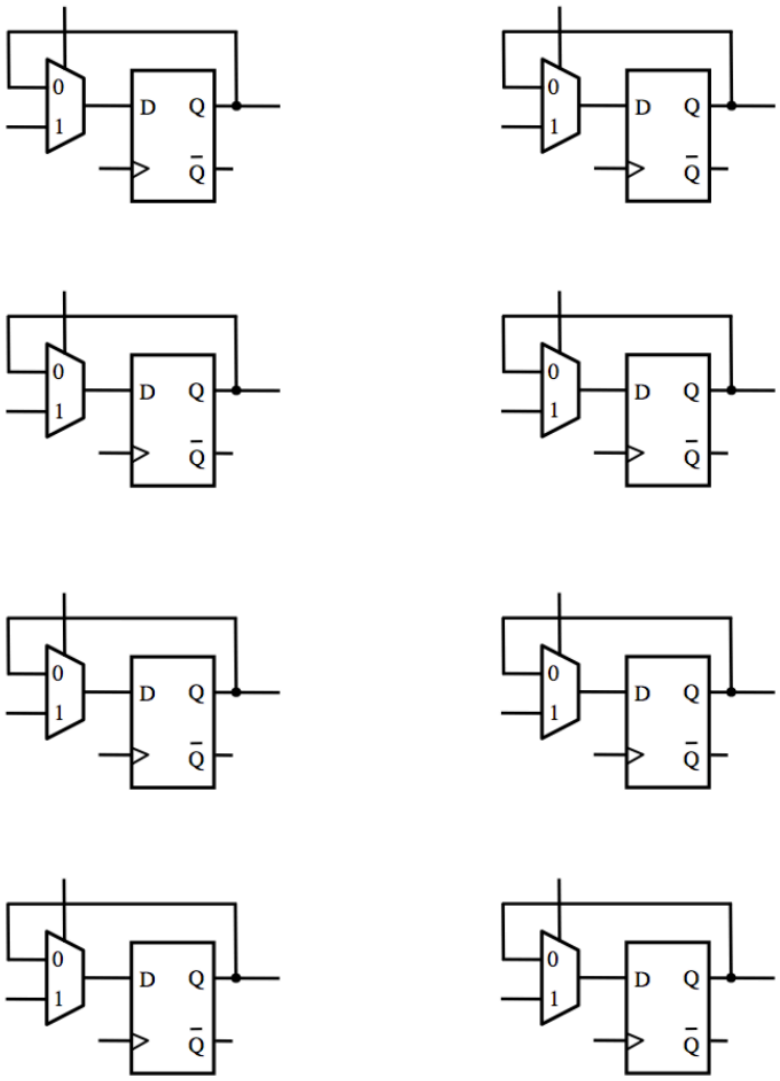
Complete the following circuit diagram to implement a register file with four 2-bit registers, one write port, one read port, and one write enable line.

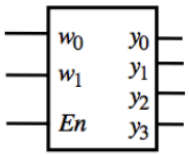


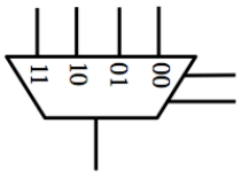


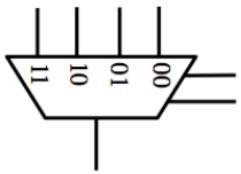


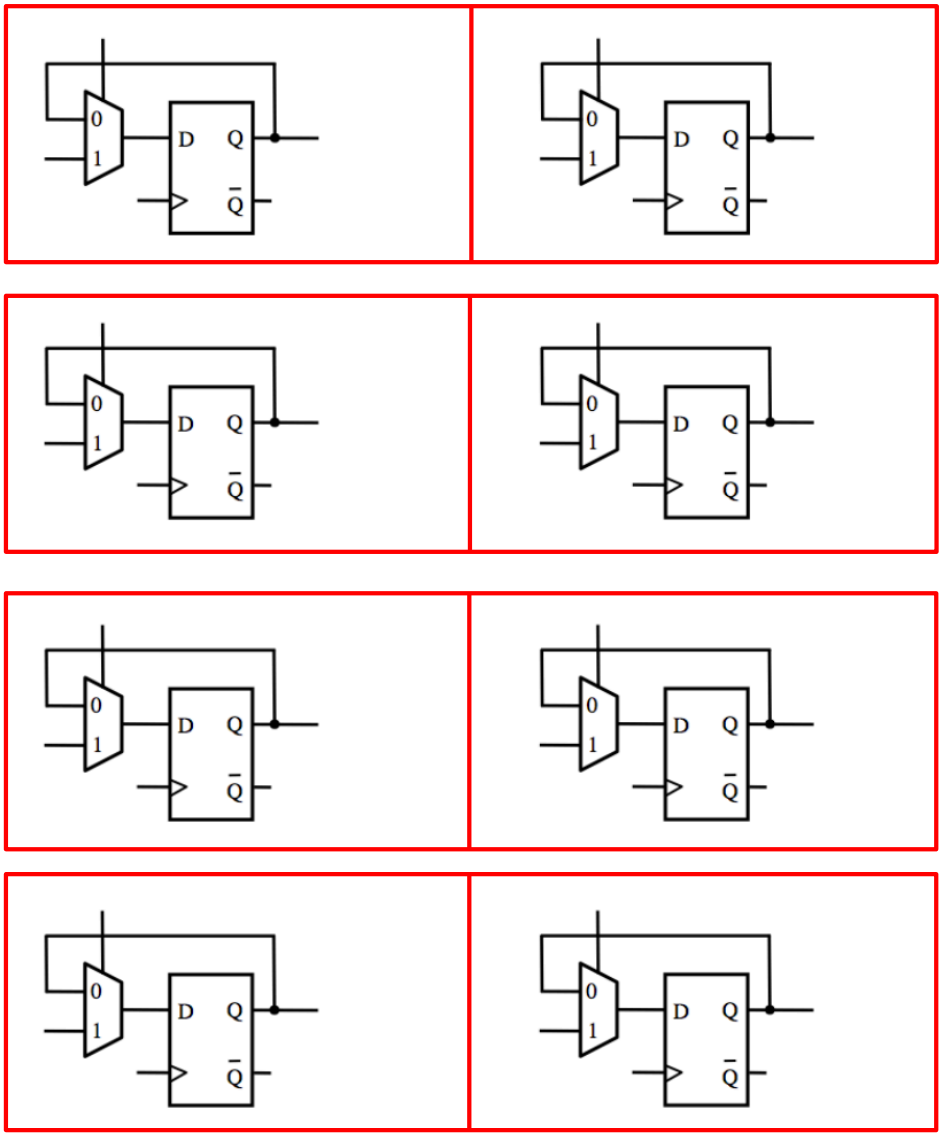


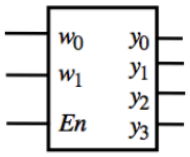

0
1
D
Q
Q

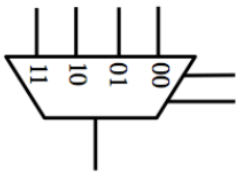

w_0
w_1
En
y_0
y_1
y_2
y_3

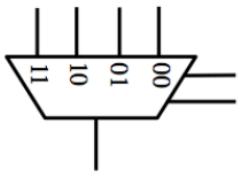

11
10
01
00

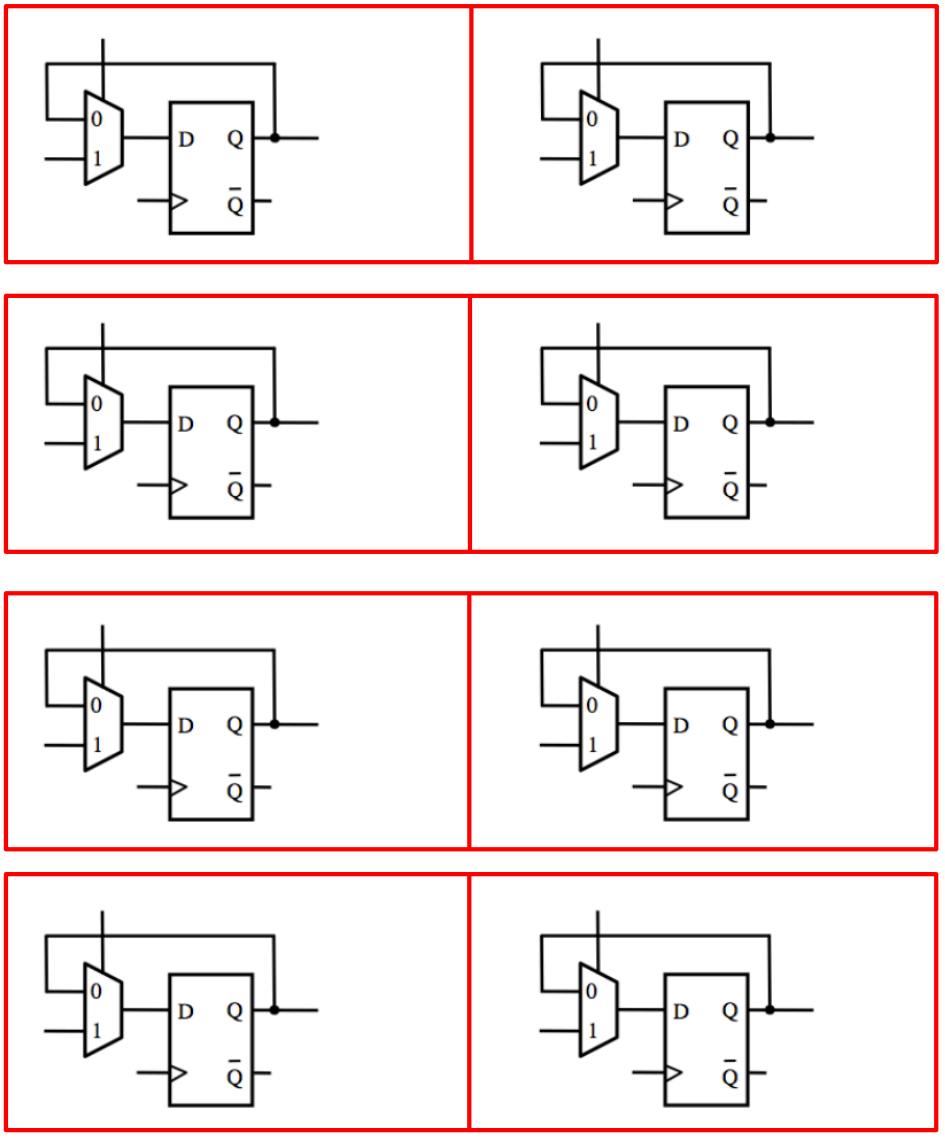

11
10
01
00

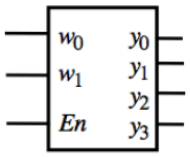

Register 0
Register 1
Register 2
Register 3
D
Q
0
1

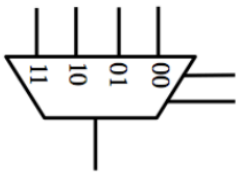

w_0
w_1
En
y_0
y_1
y_2
y_3

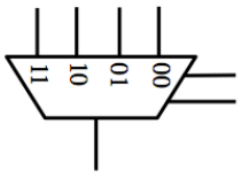

11
10
01
00

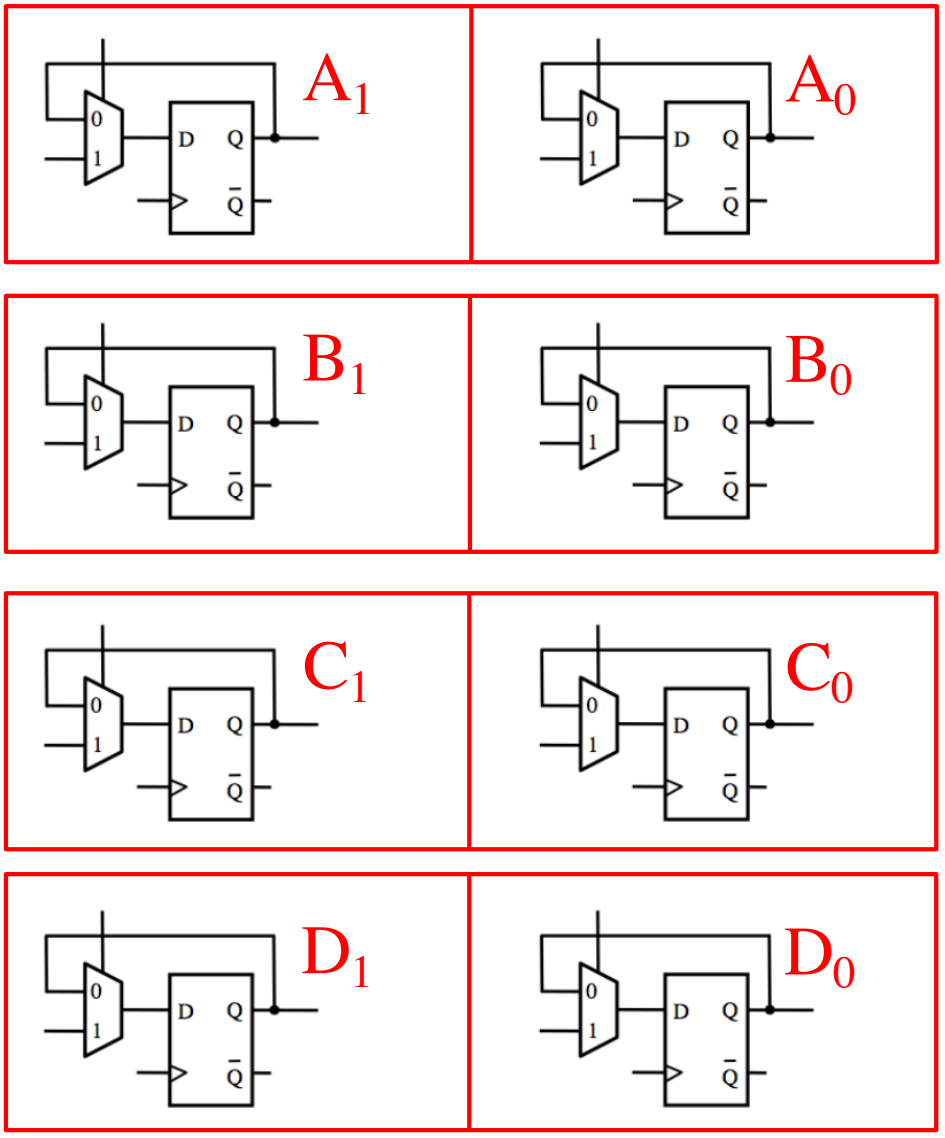

11
10
01
00

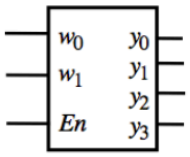

Register A
Register B
Register C
Register D
0
1
D
Q
Q̄

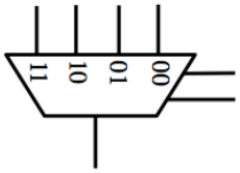

w_0
w_1
En
y_0
y_1
y_2
y_3

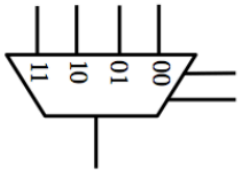

11
10
01
00

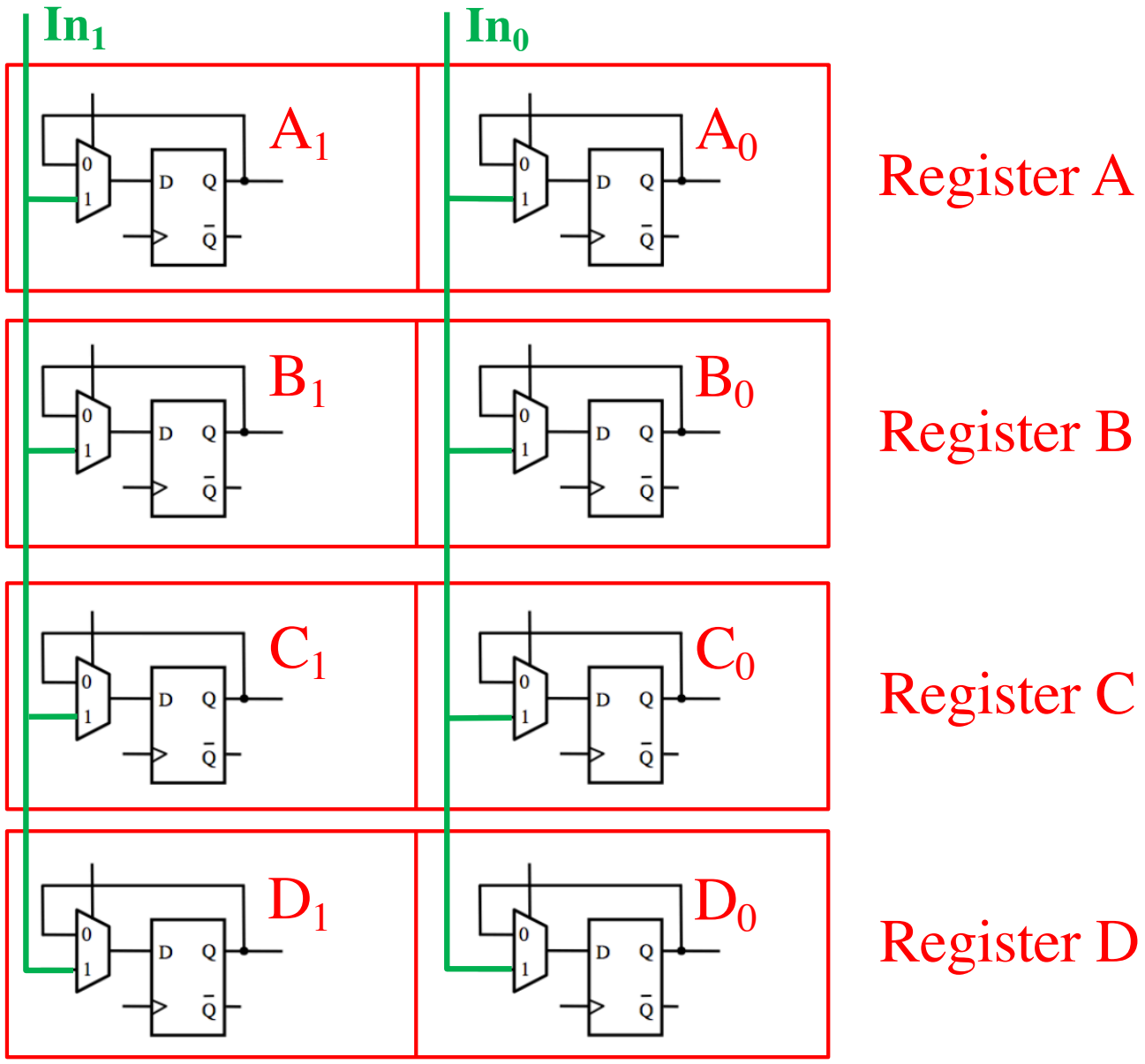

11
10
01
00

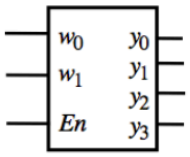

A1
A0
Register A
B1
B0
Register B
C1
C0
Register C
D1
D0
Register D
D
Q
0
1

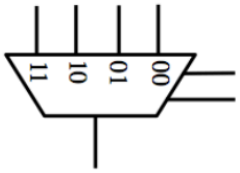

w0
w1
En
y0
y1
y2
y3

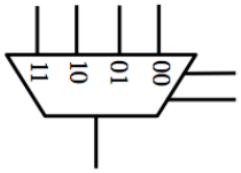

11
10
01
00

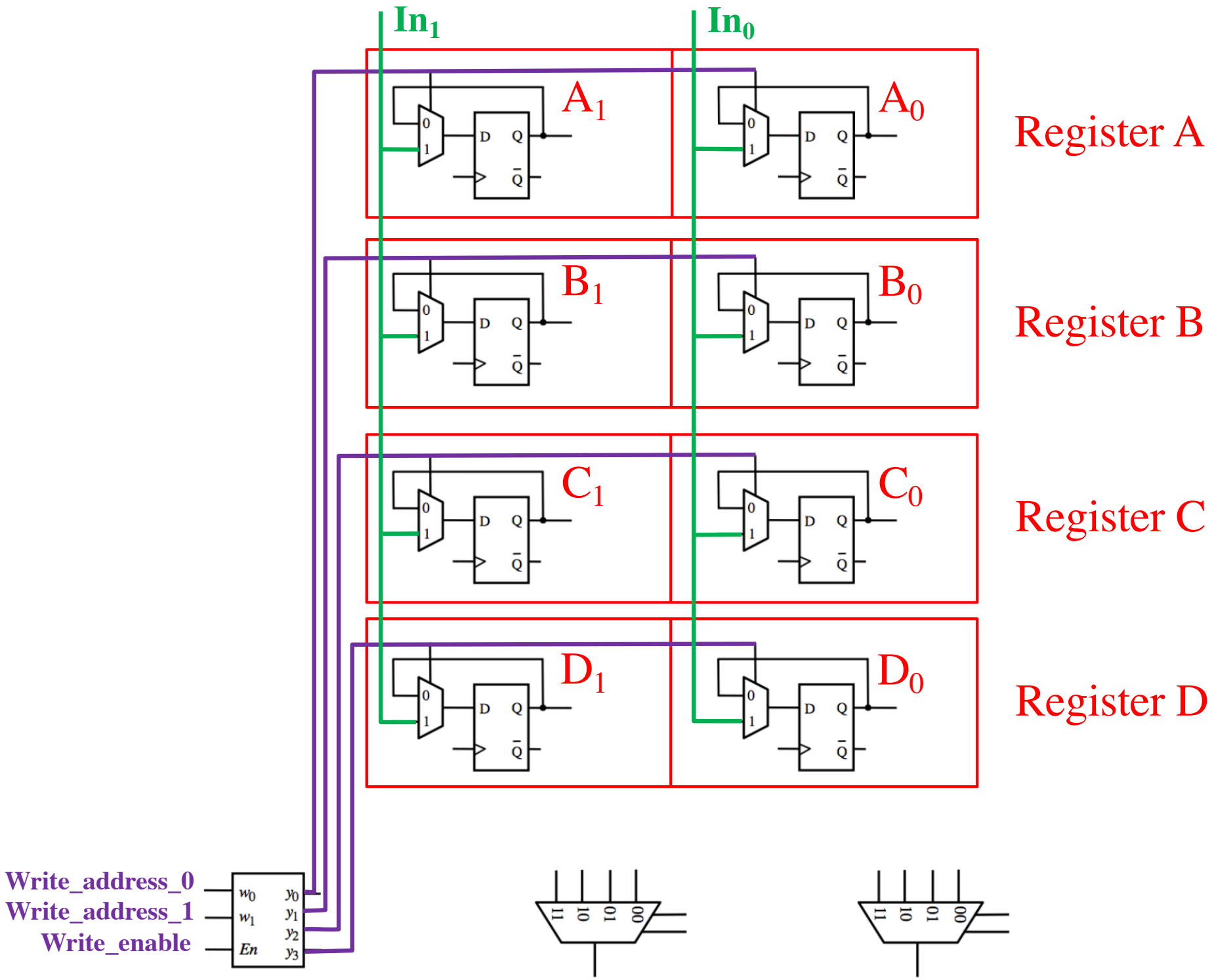

11
10
01
00

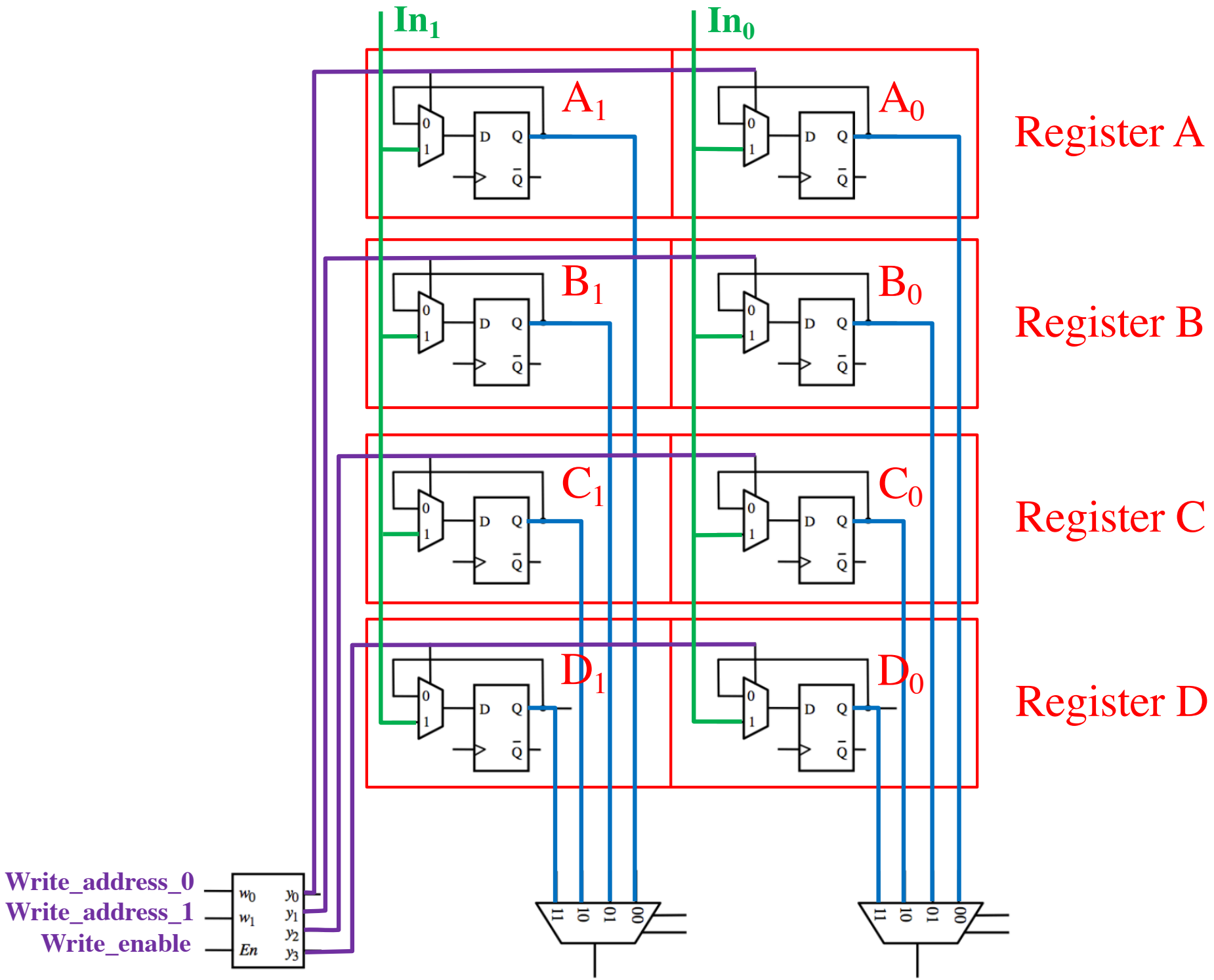

In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
D
Q
Q̄
0
1
Register A
Register B
Register C
Register D

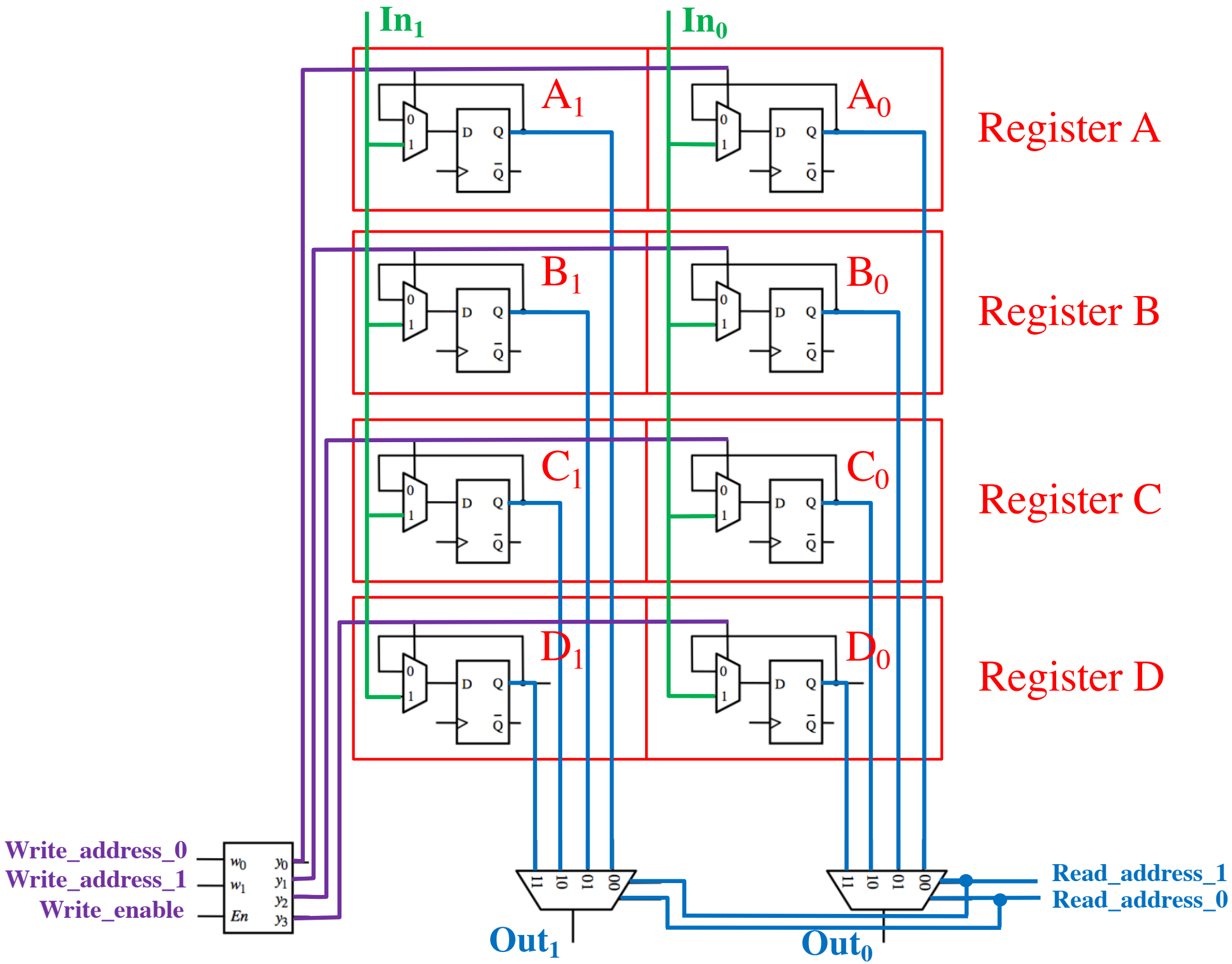

w0
w1
En
y0
y1
y2
y3

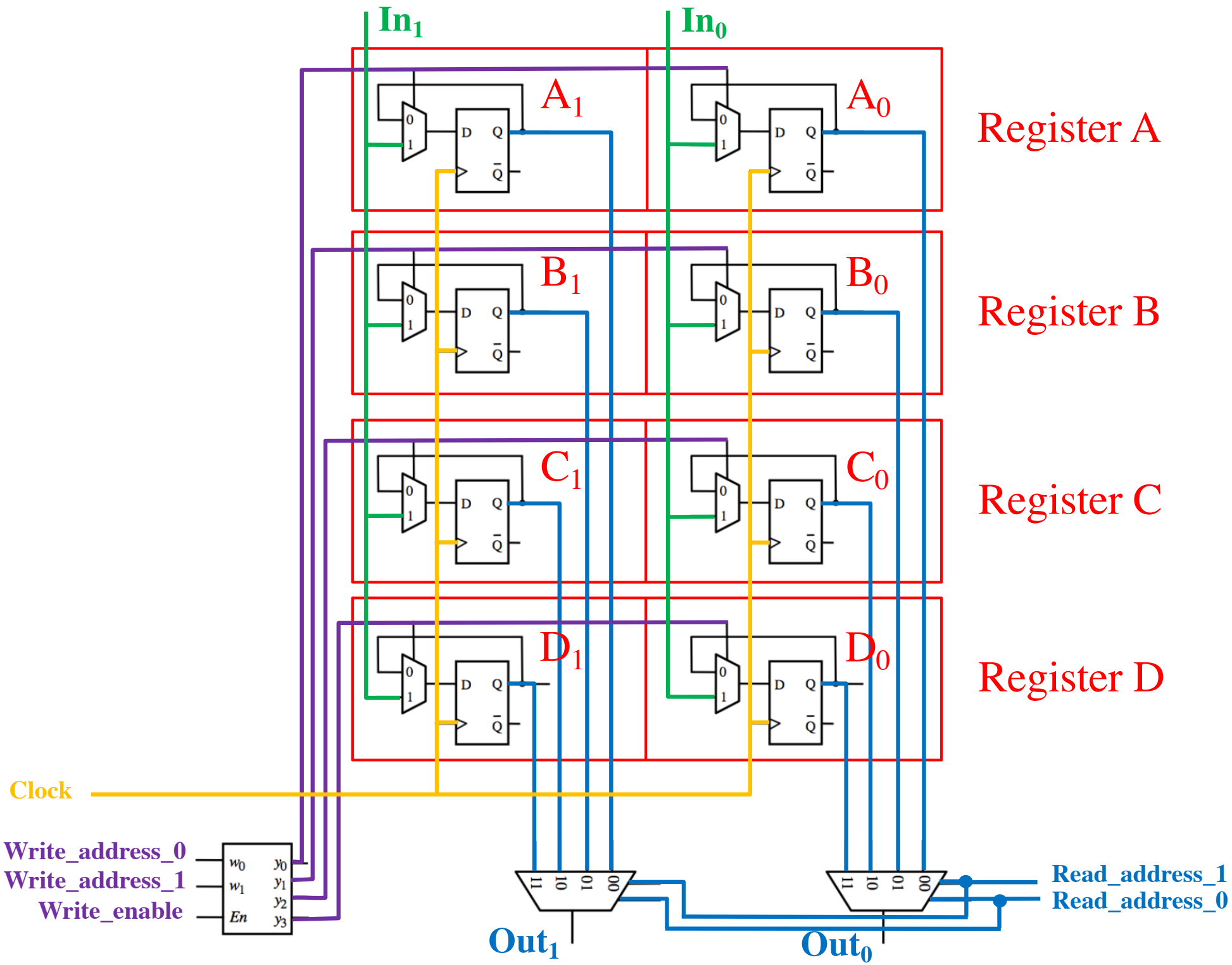

11
10
01
00

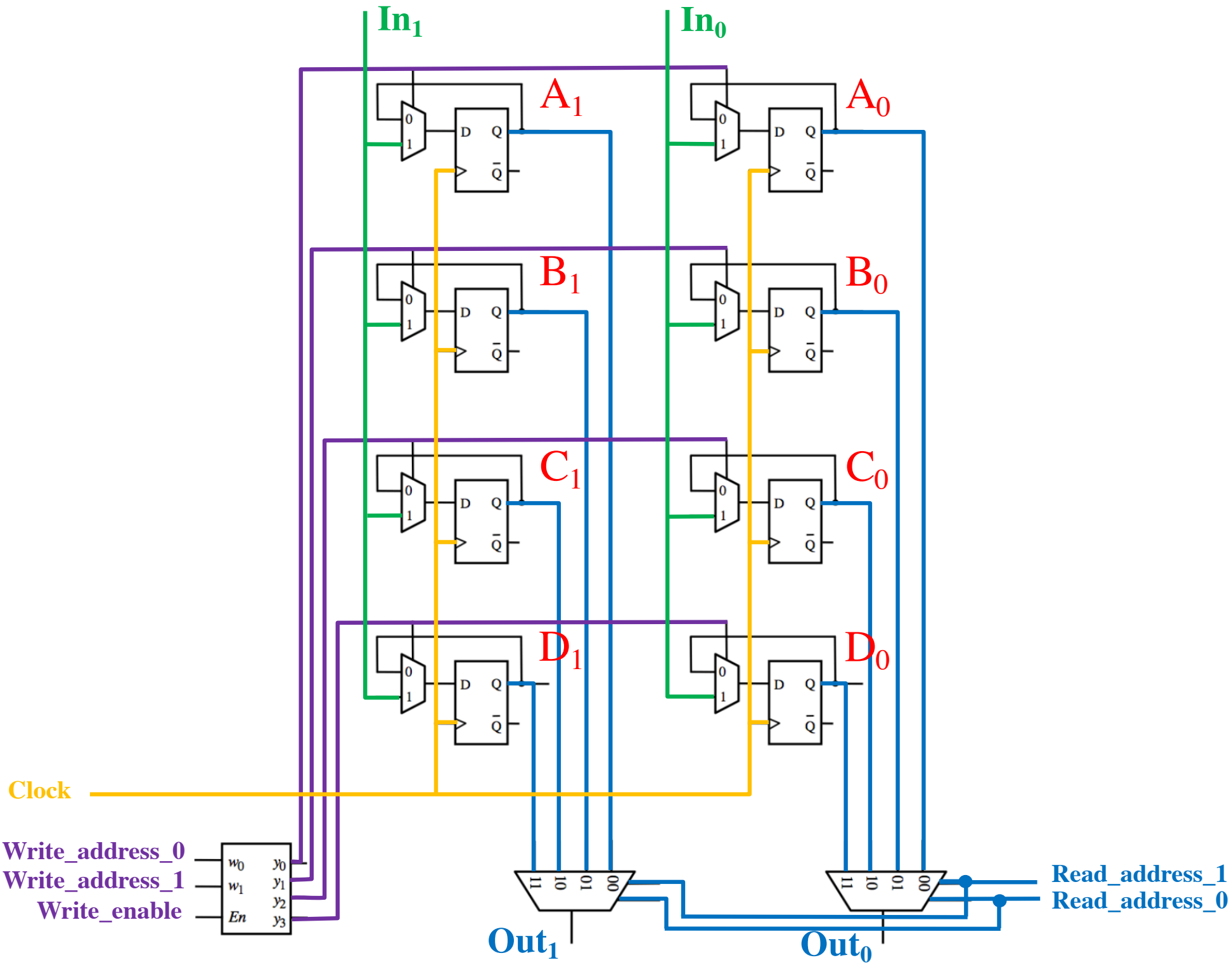

11
10
01
00


In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
Register A
Register B
Register C
Register D
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11
10
01
00
11
10
01
00


In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
Register A
Register B
Register C
Register D
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11
10
01
00

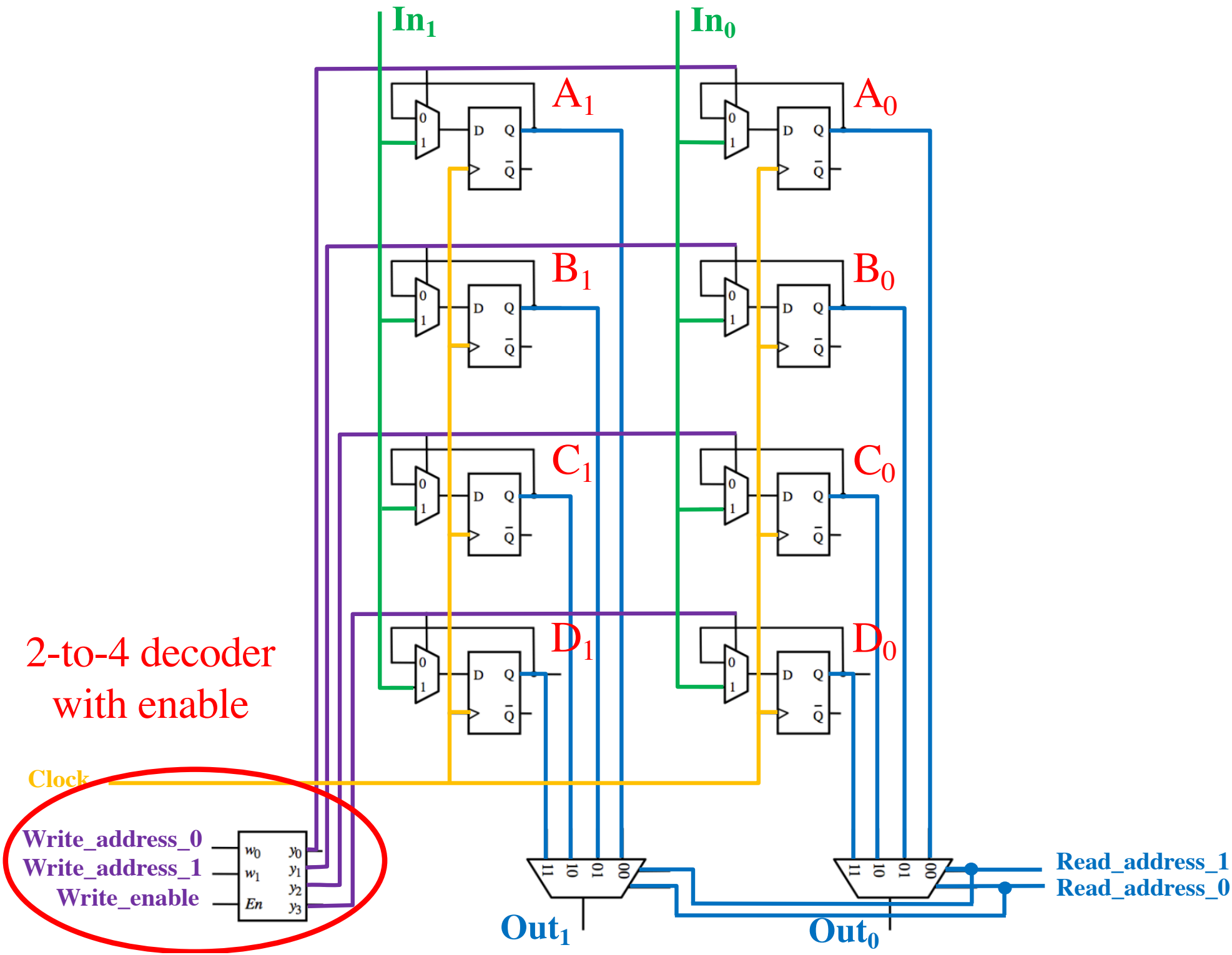

In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
Register A
Register B
Register C
Register D
D
Q
Q̄
0
1
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11
10
01
00
Out1
Out0
Read_address_1
Read_address_0


In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
Register A
Register B
Register C
Register D
Clock
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11
10
01
00
Out1
Out0
Read_address_1
Read_address_0

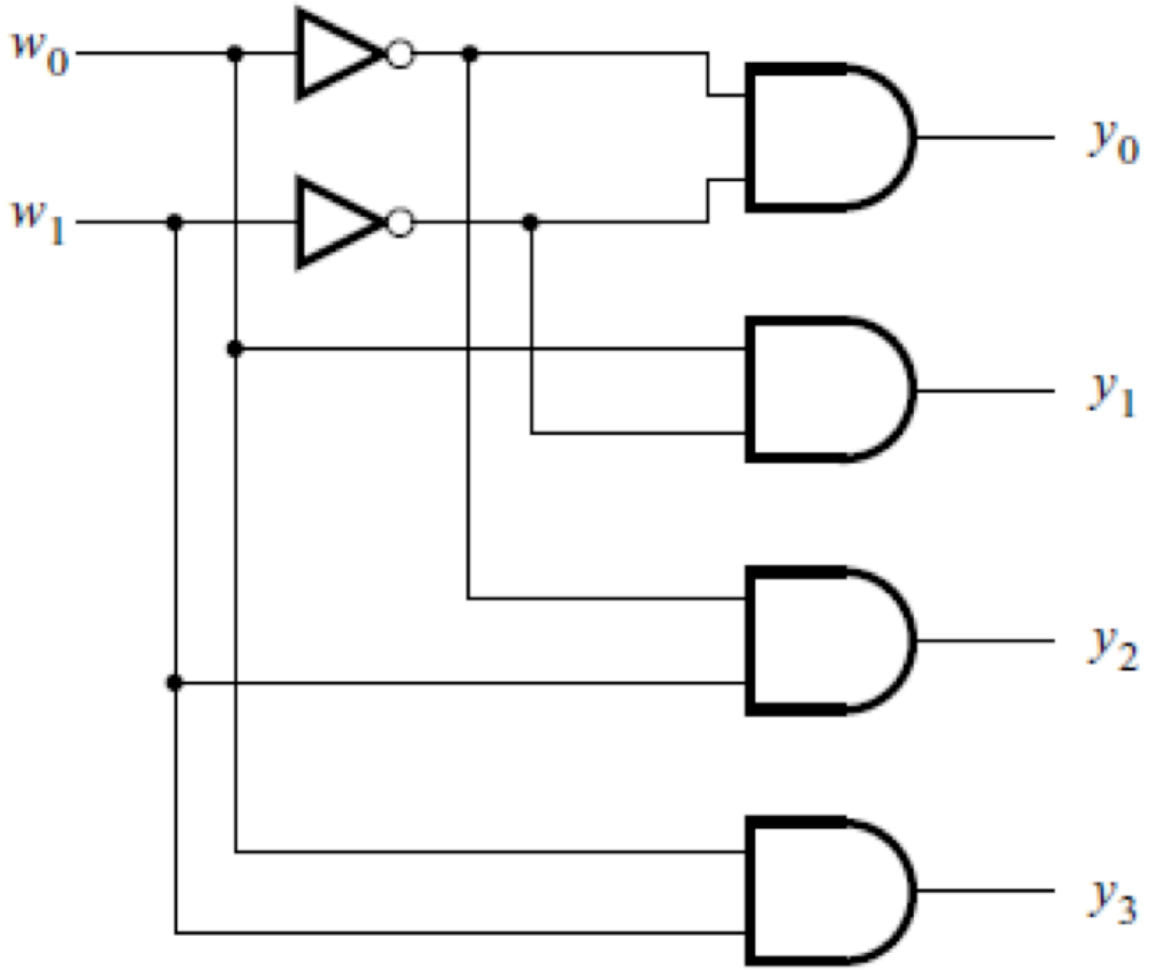

In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
D
Q
Q̄
0
1
Clock
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11
10
01
00
Out1
Out0
Read_address_1
Read_address_0

Components of the Register File

2-to-4 Decoder

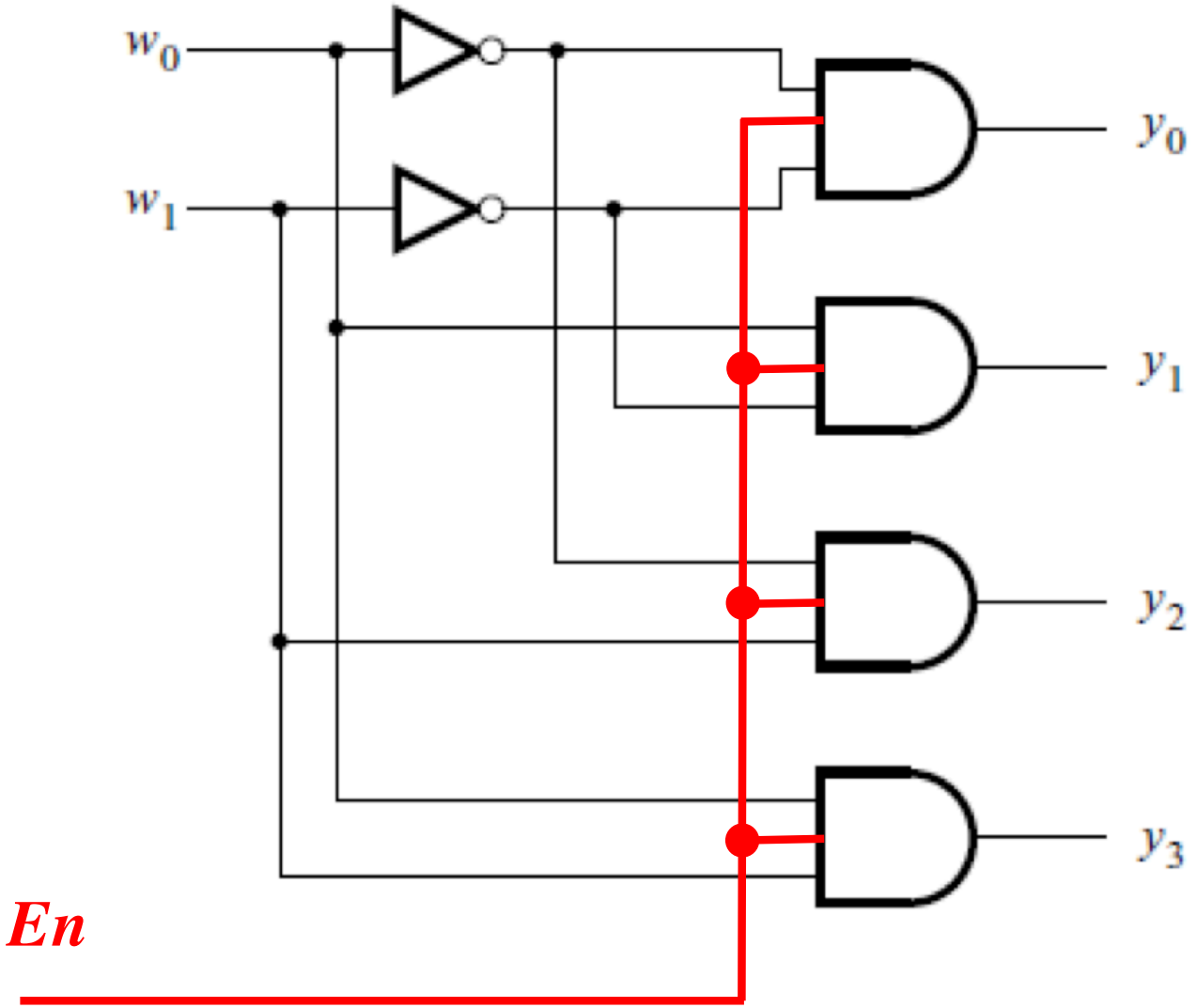

In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
2-to-4 decoder
with enable
Clock
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
Out1
Out0
Read_address_1
Read_address_0

2-to-4 Decoder

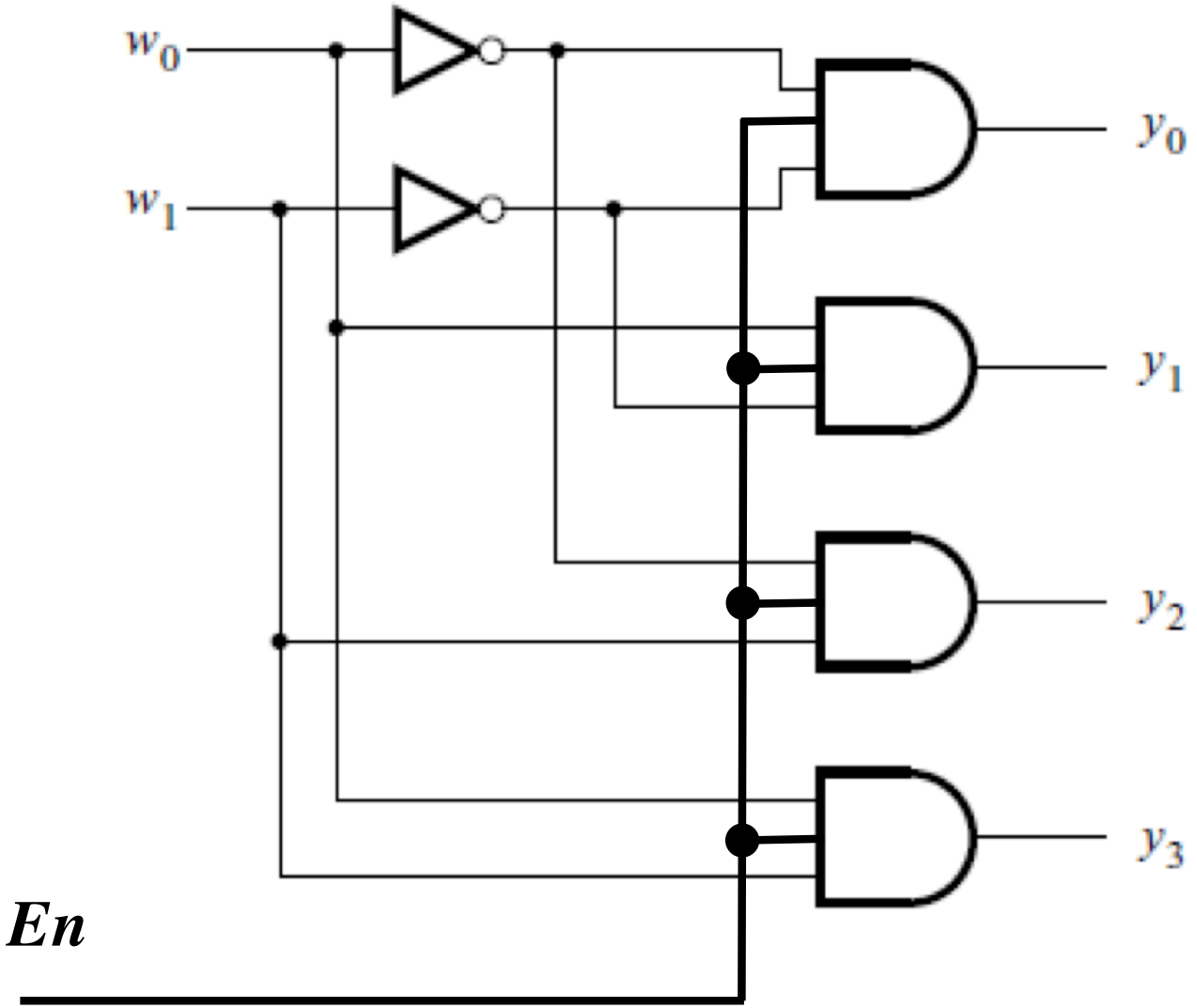


[Figure 4.14c from the textbook]

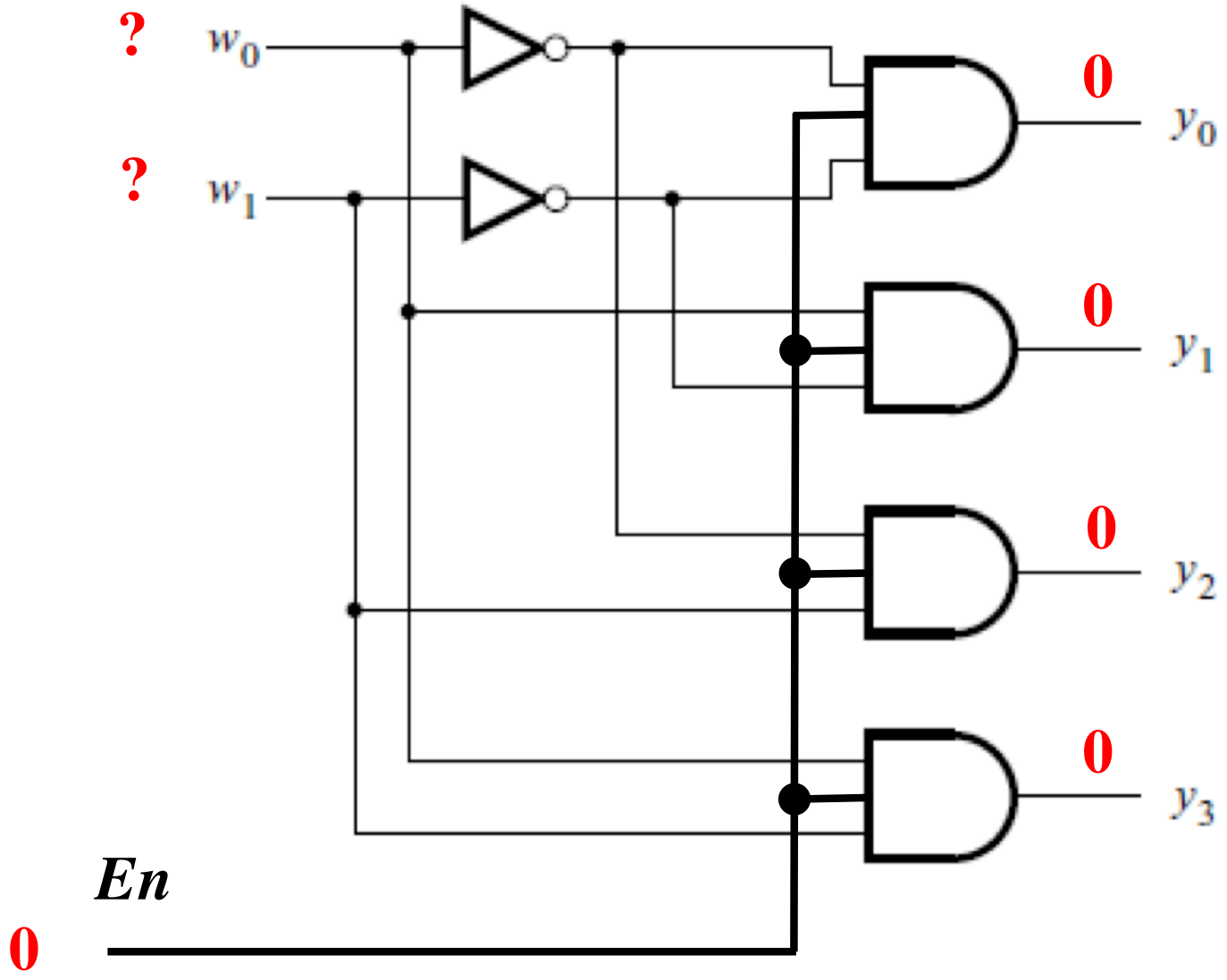
2-to-4 Decoder with Enable Input



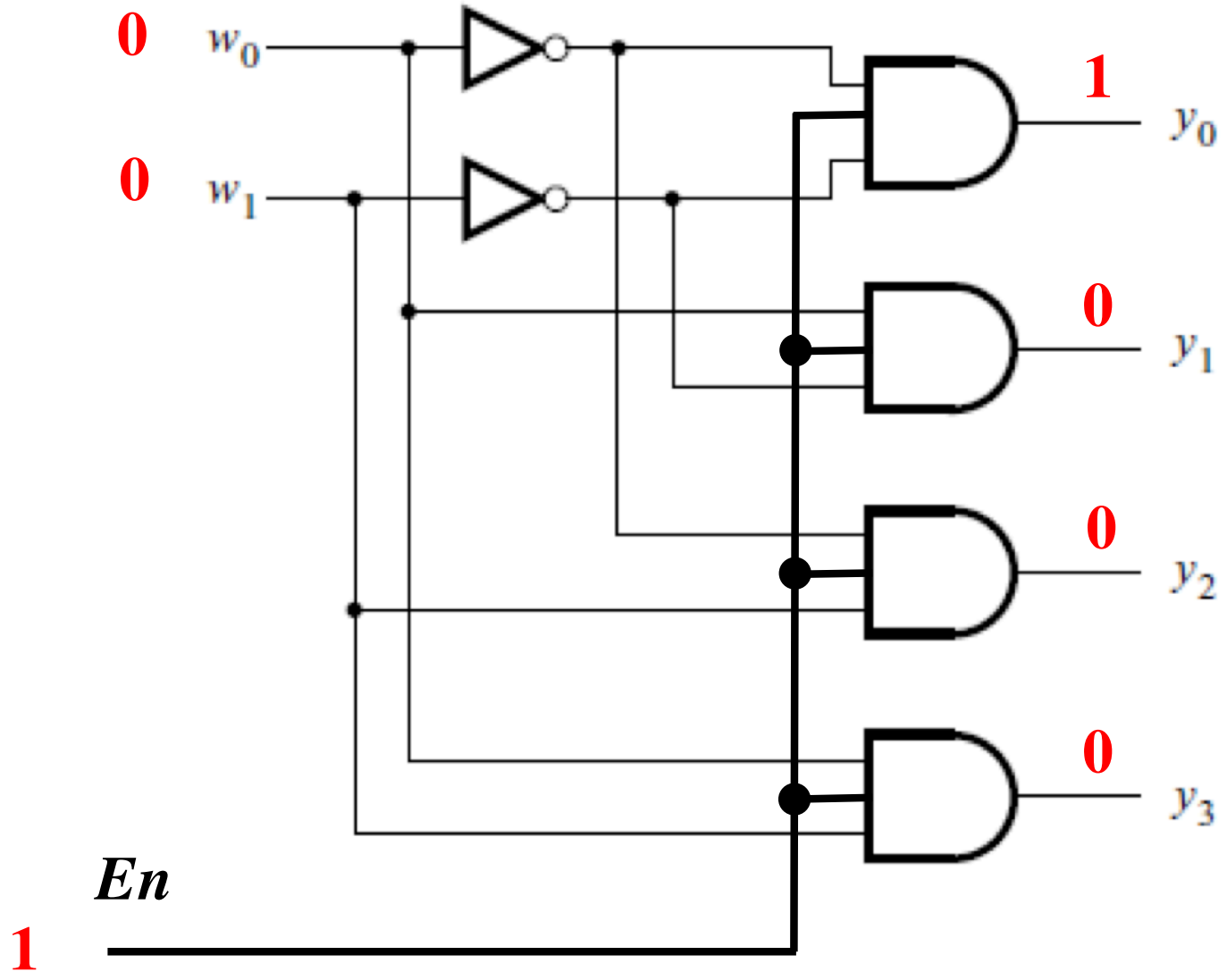
2-to-4 Decoder with Enable Input



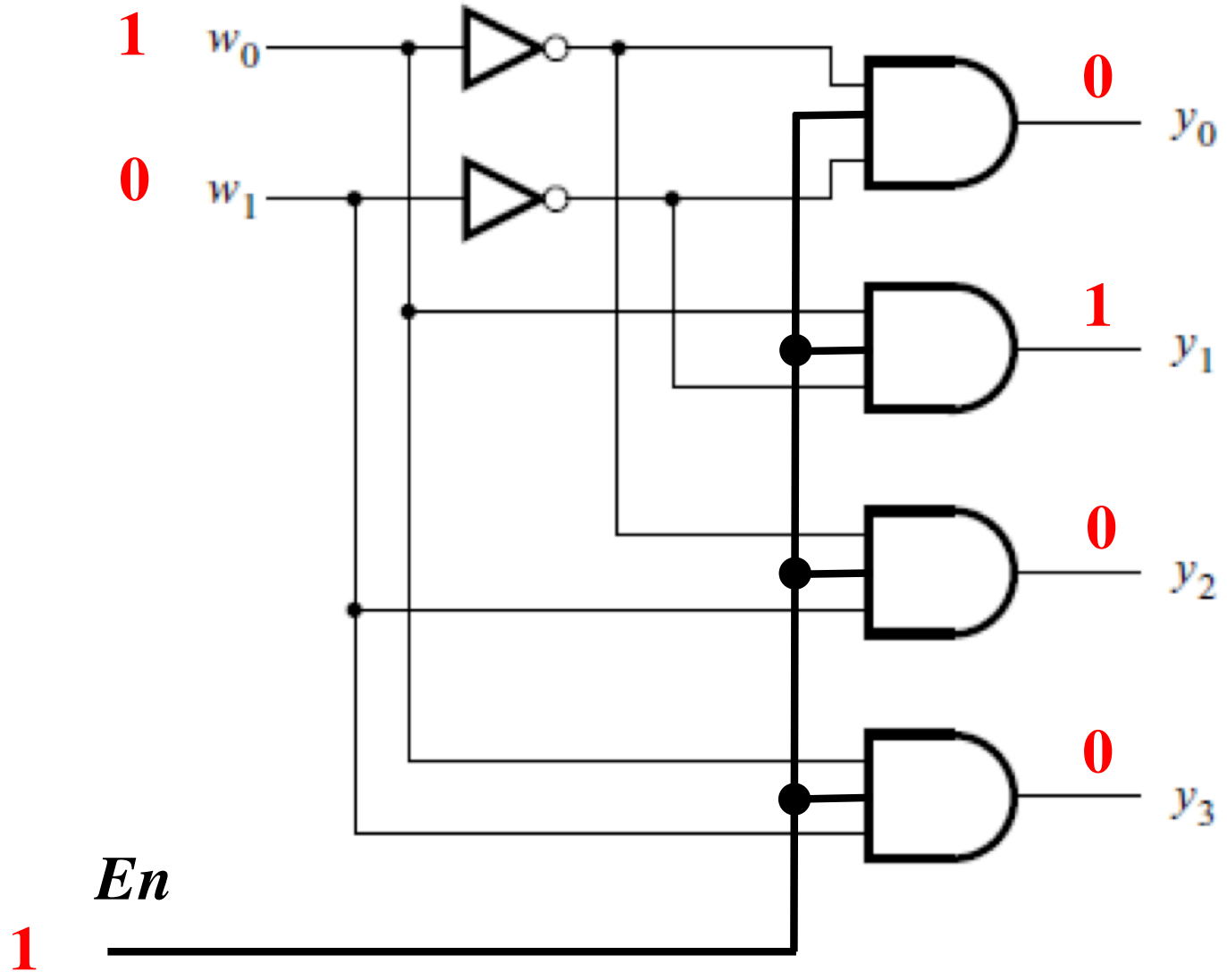
2-to-4 Decoder with Enable Input



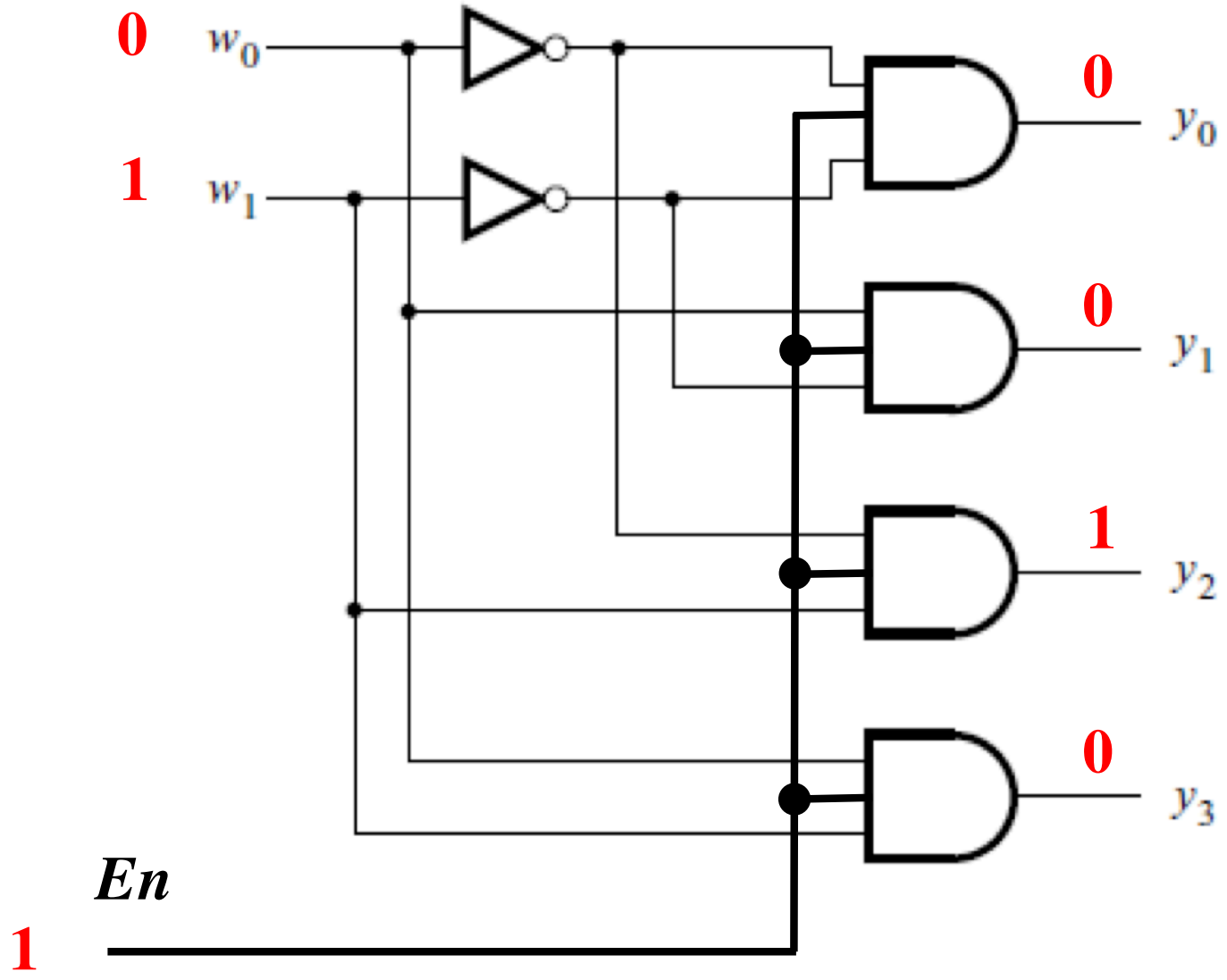
2-to-4 Decoder with Enable Input



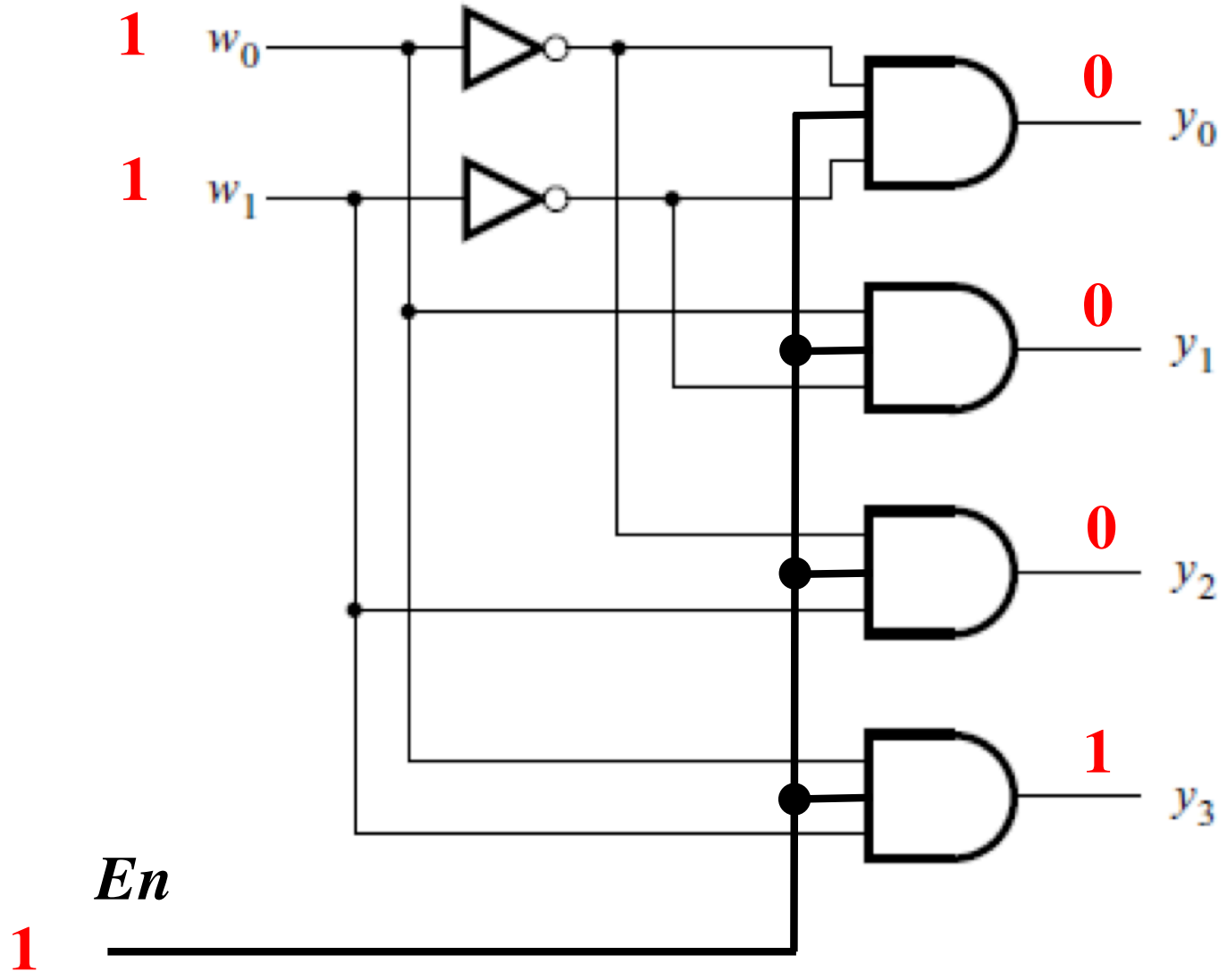
2-to-4 Decoder with Enable Input



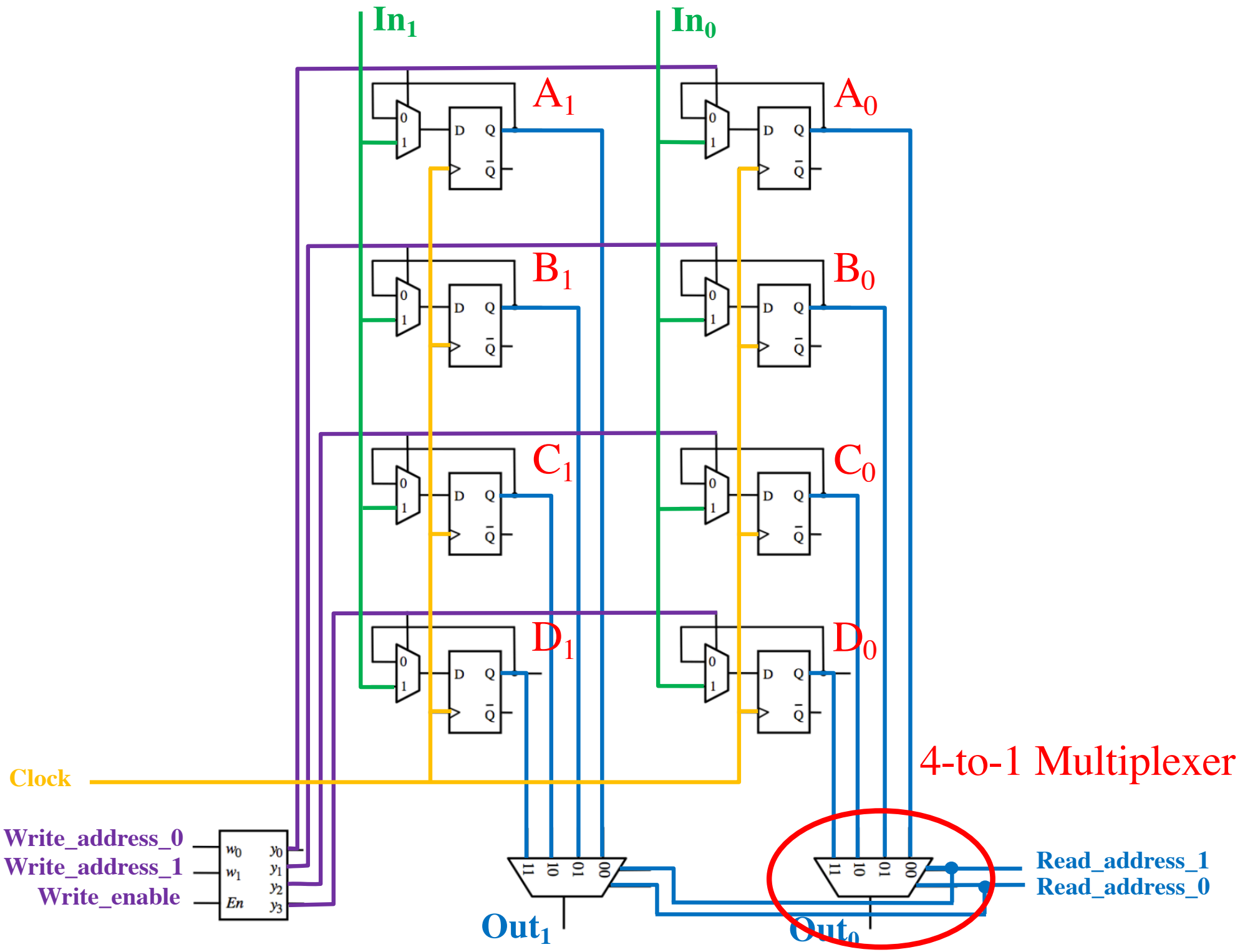
2-to-4 Decoder with Enable Input

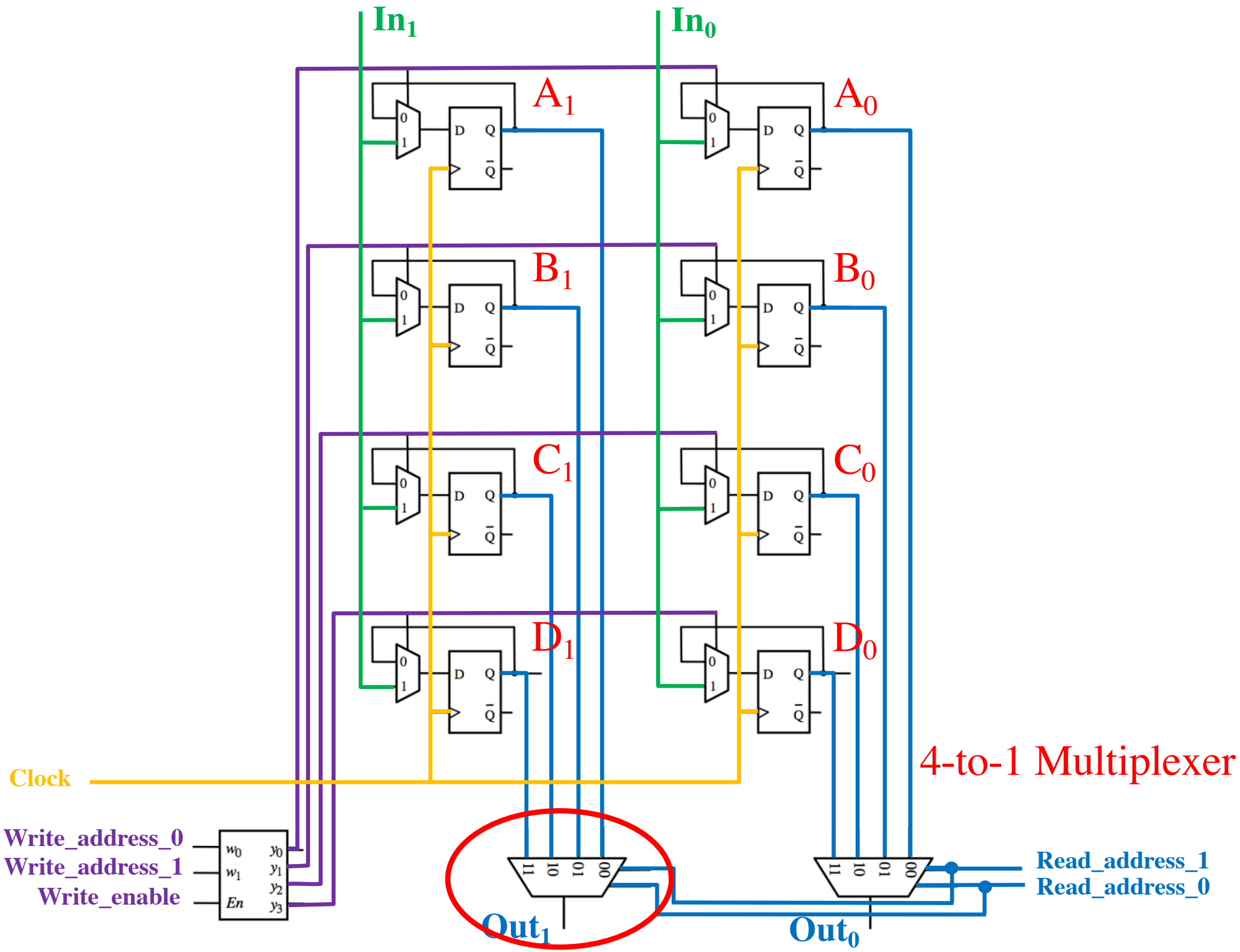


2-to-4 Decoder with Enable Input

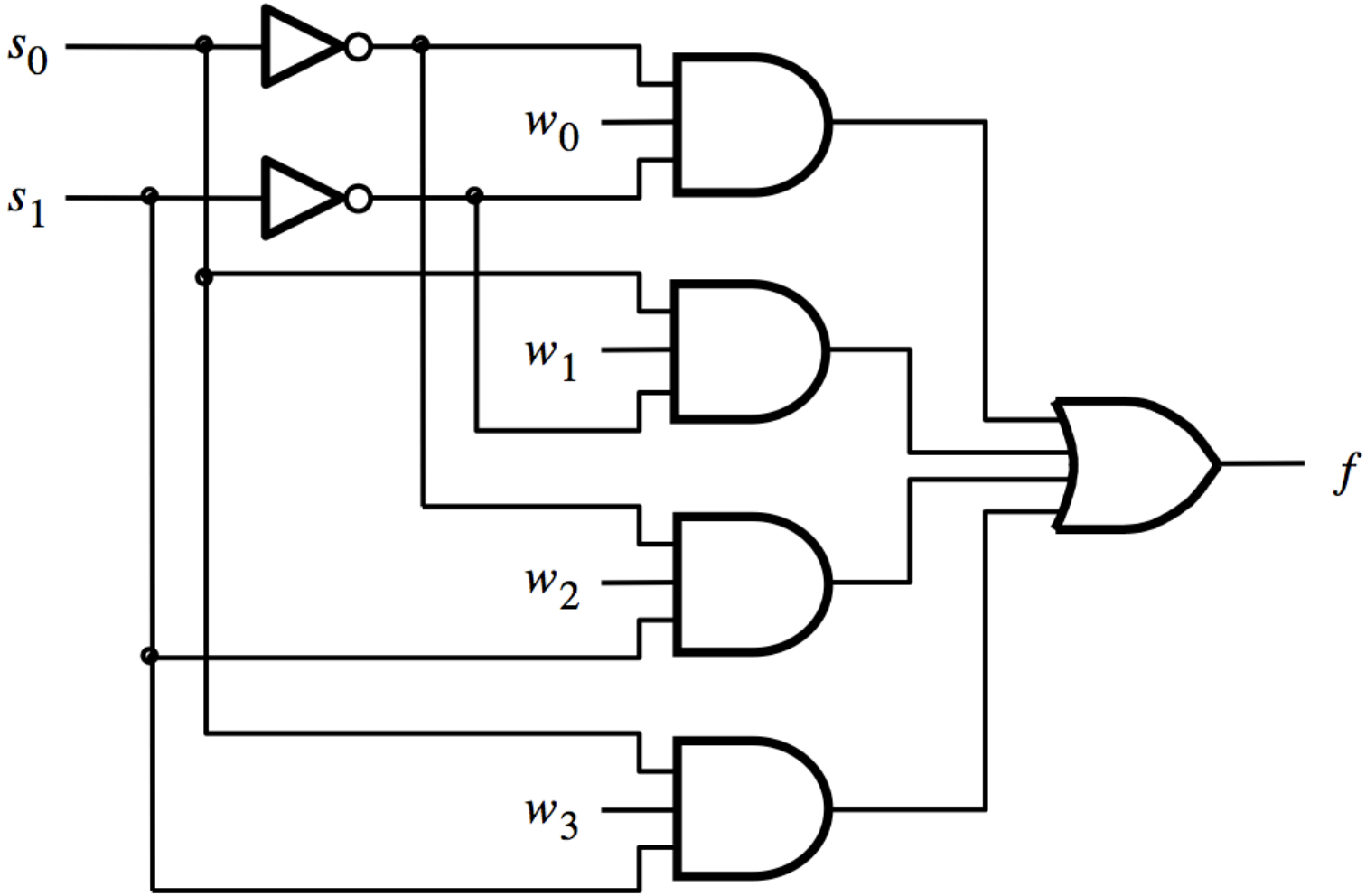


4-to-1 Multiplexer


In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
D
Q
Q̄
0
1
Clock
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11
10
01
00
Out1
Out0
4-to-1 Multiplexer
Read_address_1
Read_address_0


In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
D
Q
Q̄
0
1
Clock
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11
10
01
00
Out1
Out0
4-to-1 Multiplexer
Read_address_1
Read_address_0

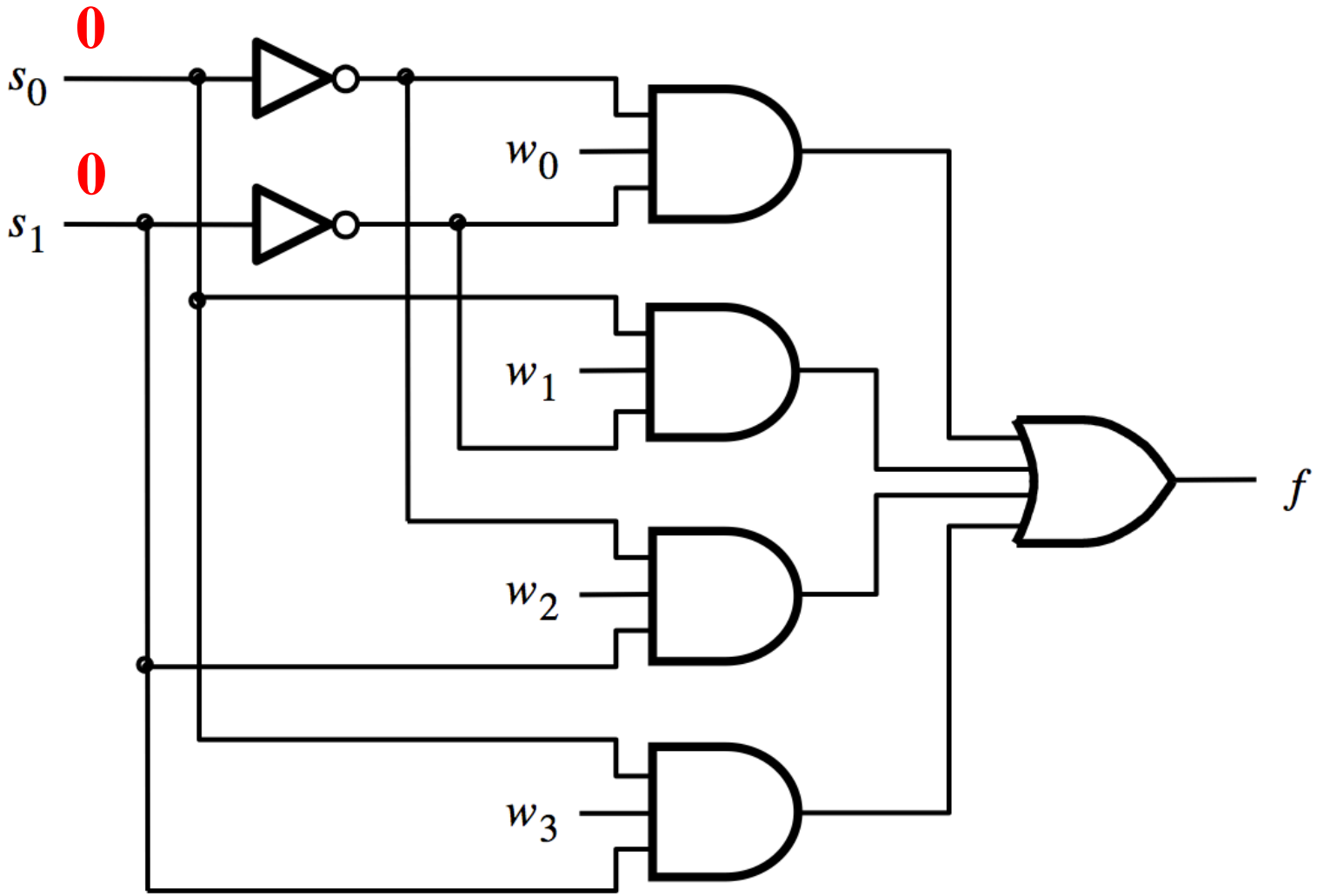
4-to-1 Multiplexer



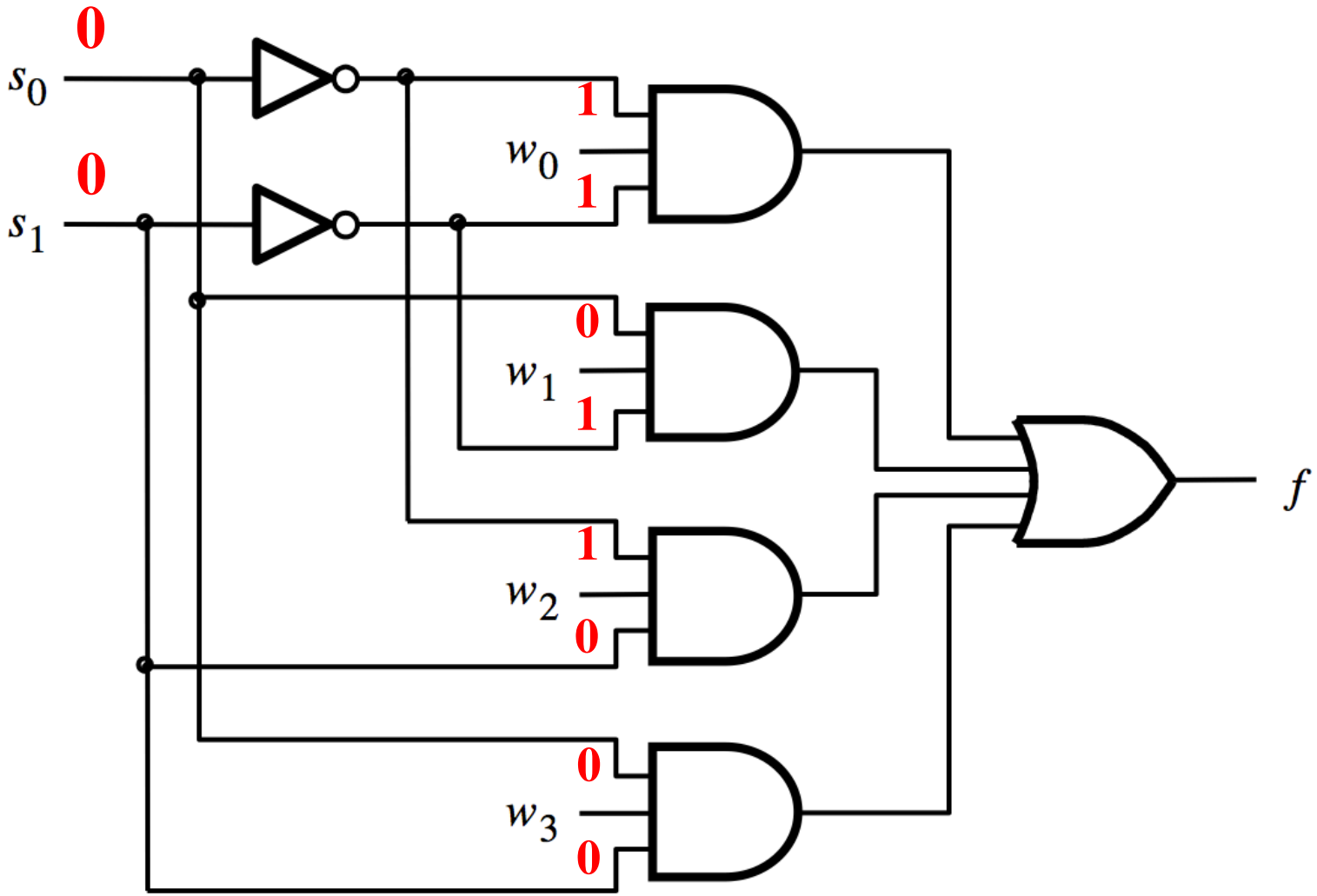
$$f = \overline{s_1}\,\overline{s_0}\,w_0 + \overline{s_1}\,s_0\,w_1 + s_1\,\overline{s_0}\,w_2 + s_1\,s_0\,w_3$$

[Figure 4.2c from the textbook]

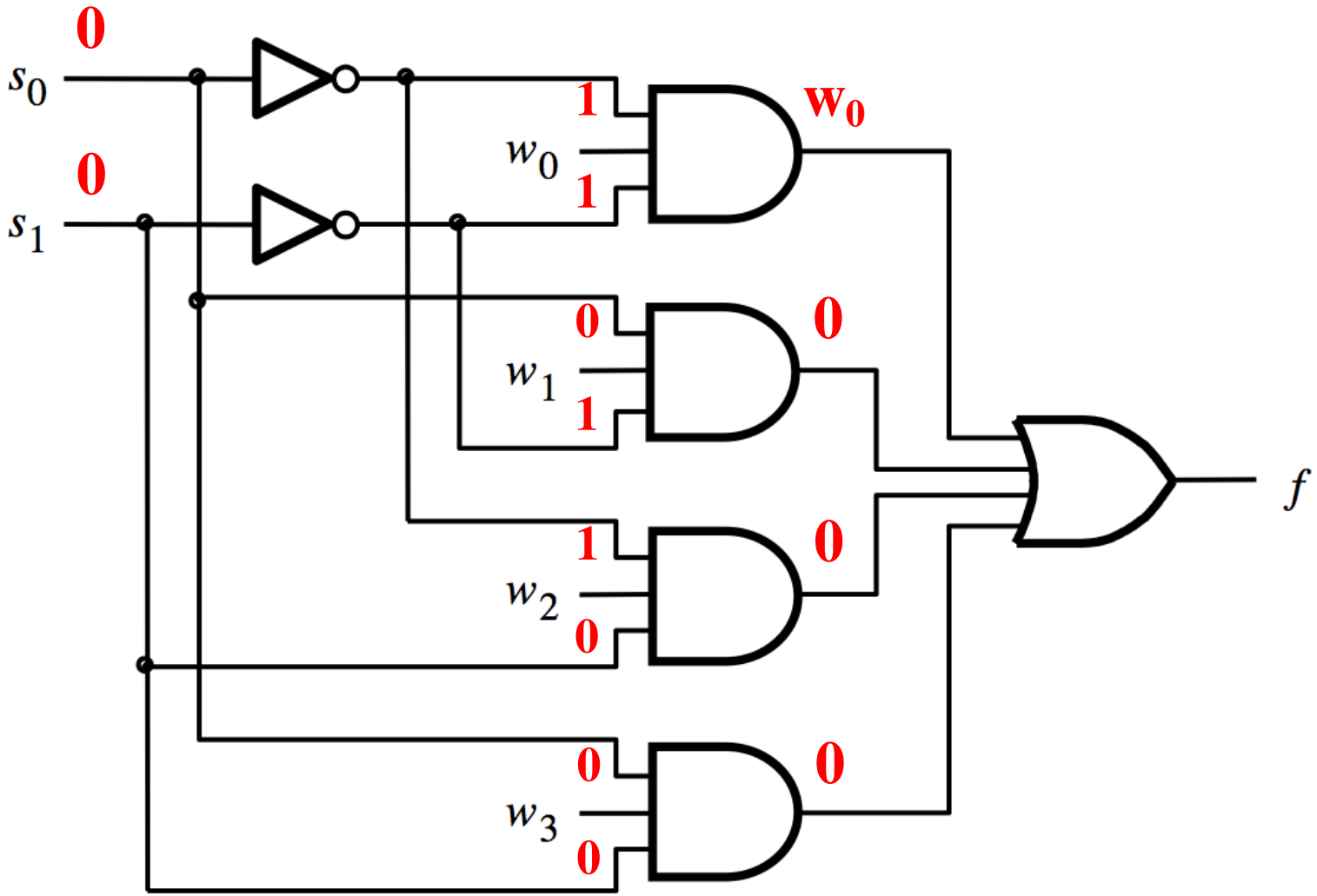
Analysis of the 4-to-1 Multiplexer (s_1=0 and s_0=0)



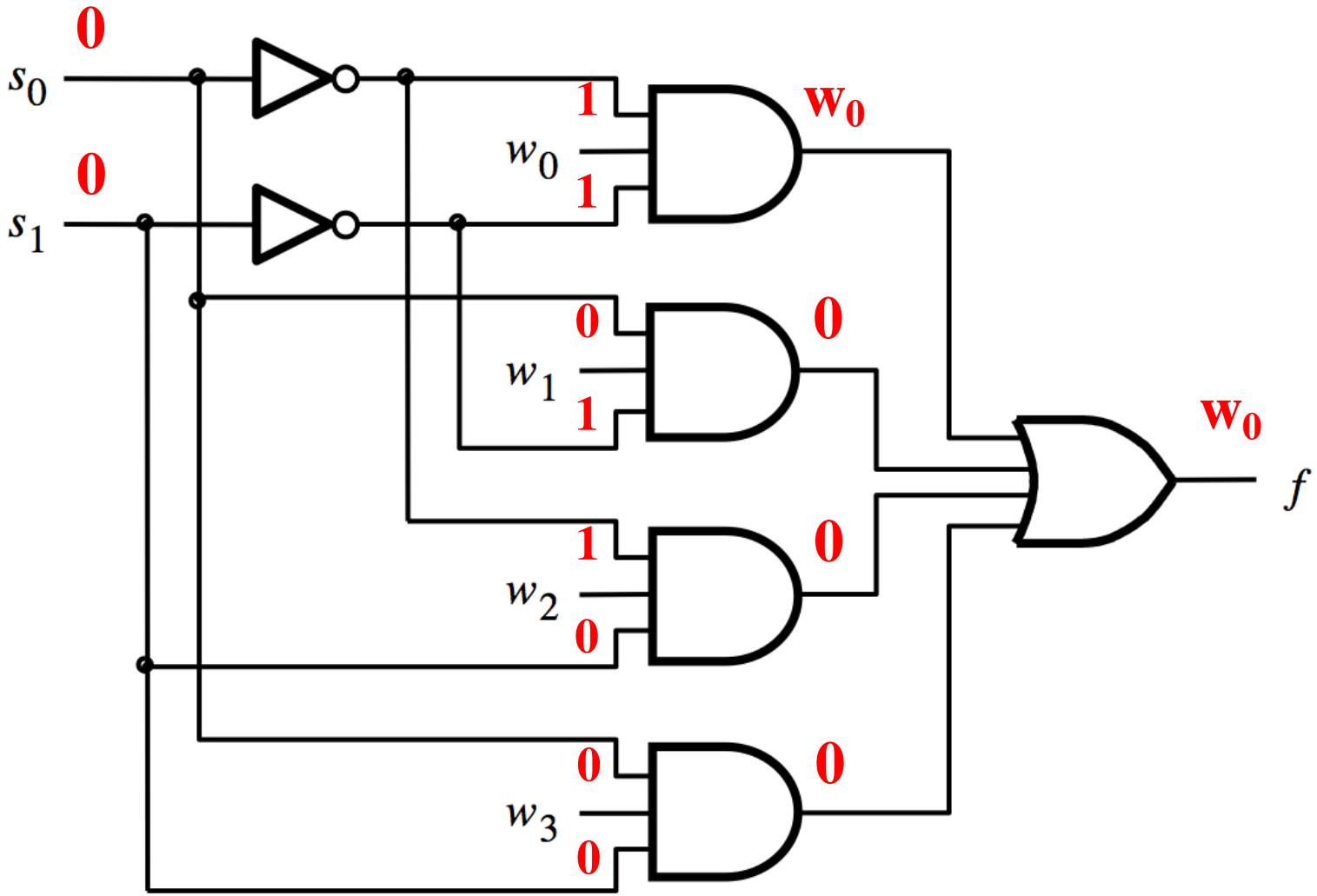
Analysis of the 4-to-1 Multiplexer (s_1=0 and s_0=0)



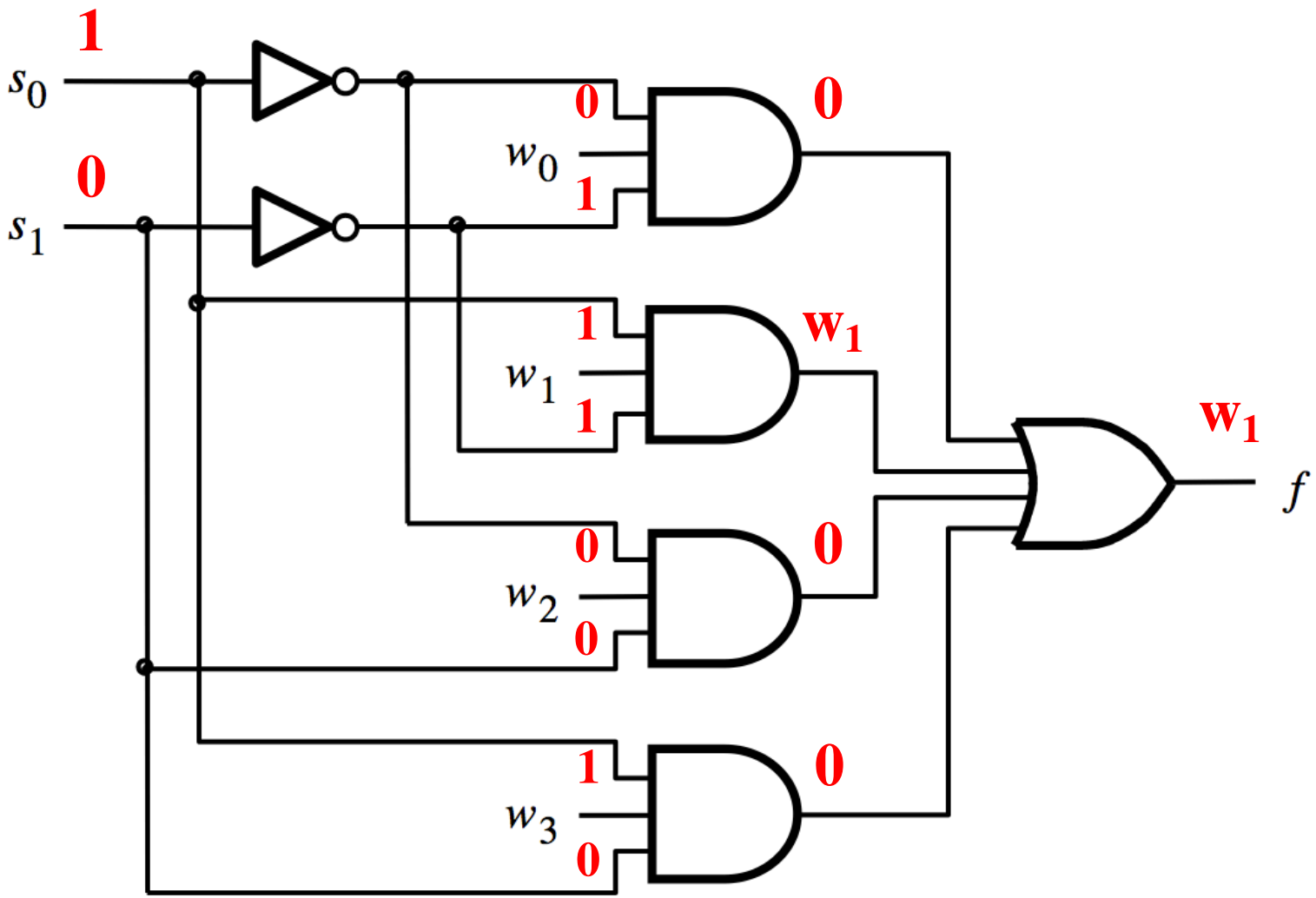
Analysis of the 4-to-1 Multiplexer (s_1=0 and s_0=0)



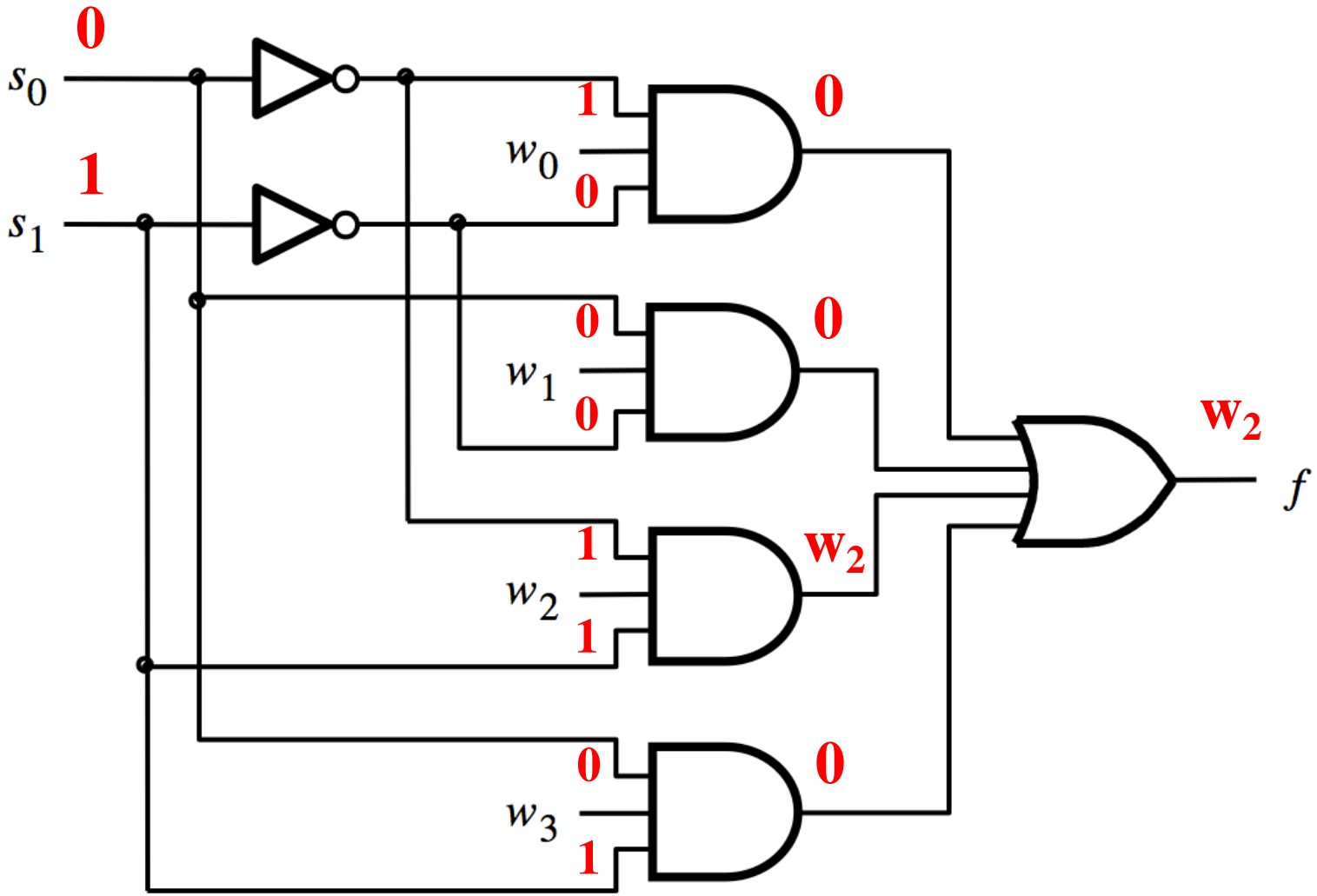
Analysis of the 4-to-1 Multiplexer (s_1=0 and s_0=0)



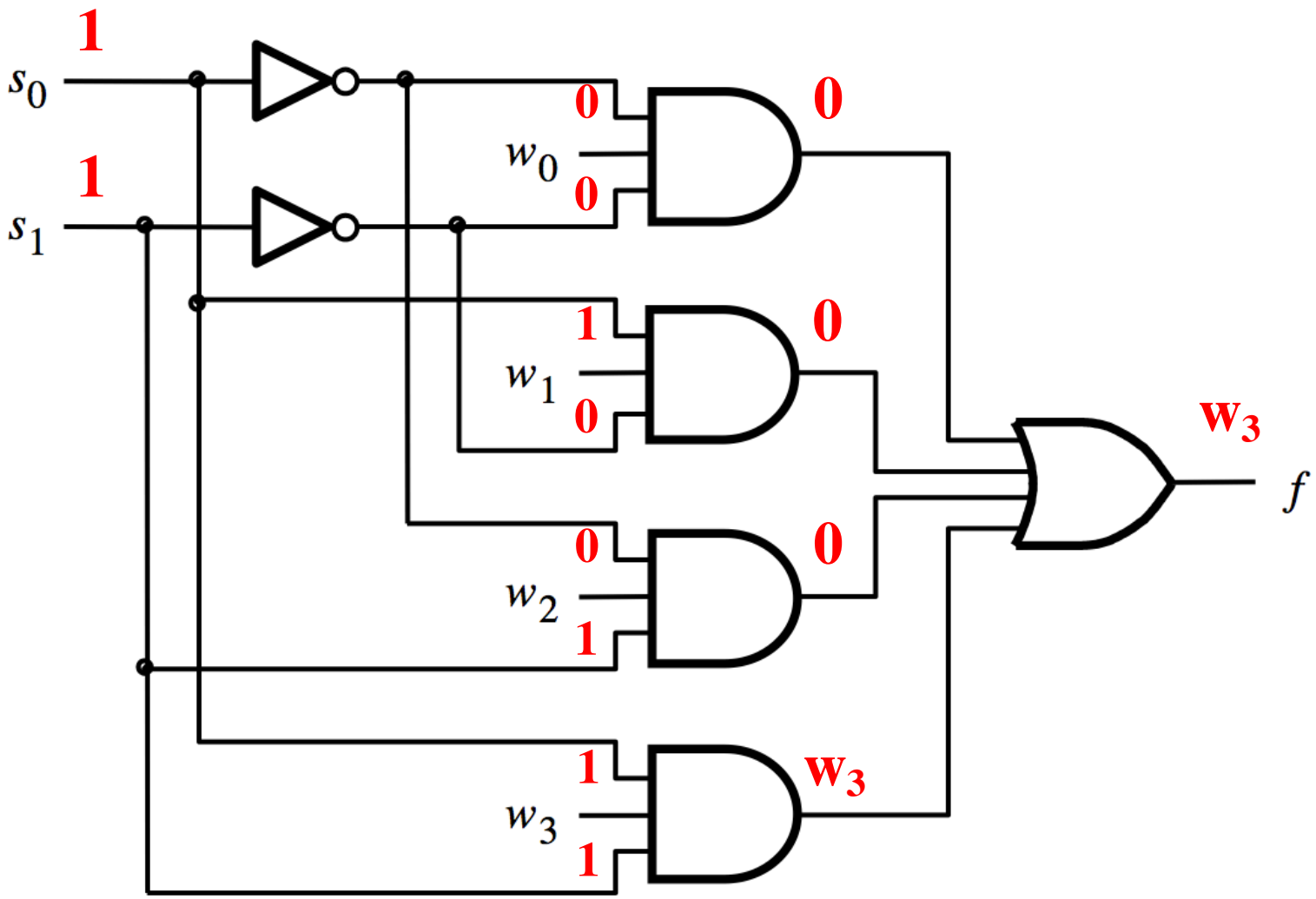
Analysis of the 4-to-1 Multiplexer (s_1=0 and s_0=1)



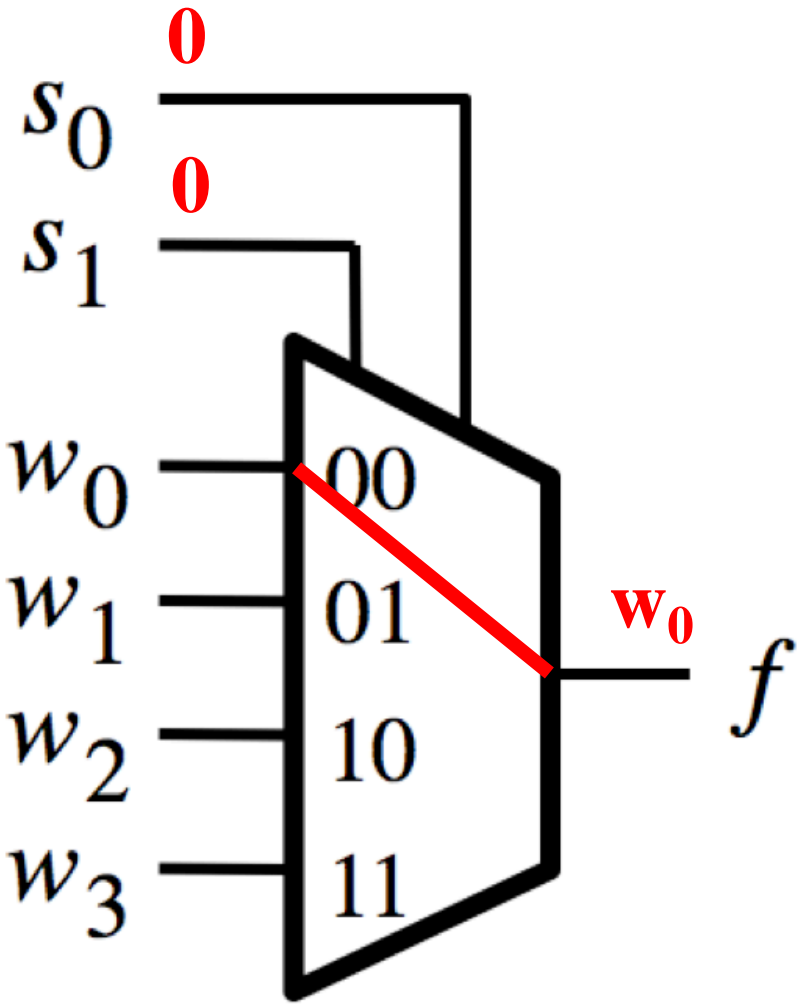
Analysis of the 4-to-1 Multiplexer (s_1=1 and s_0=0)



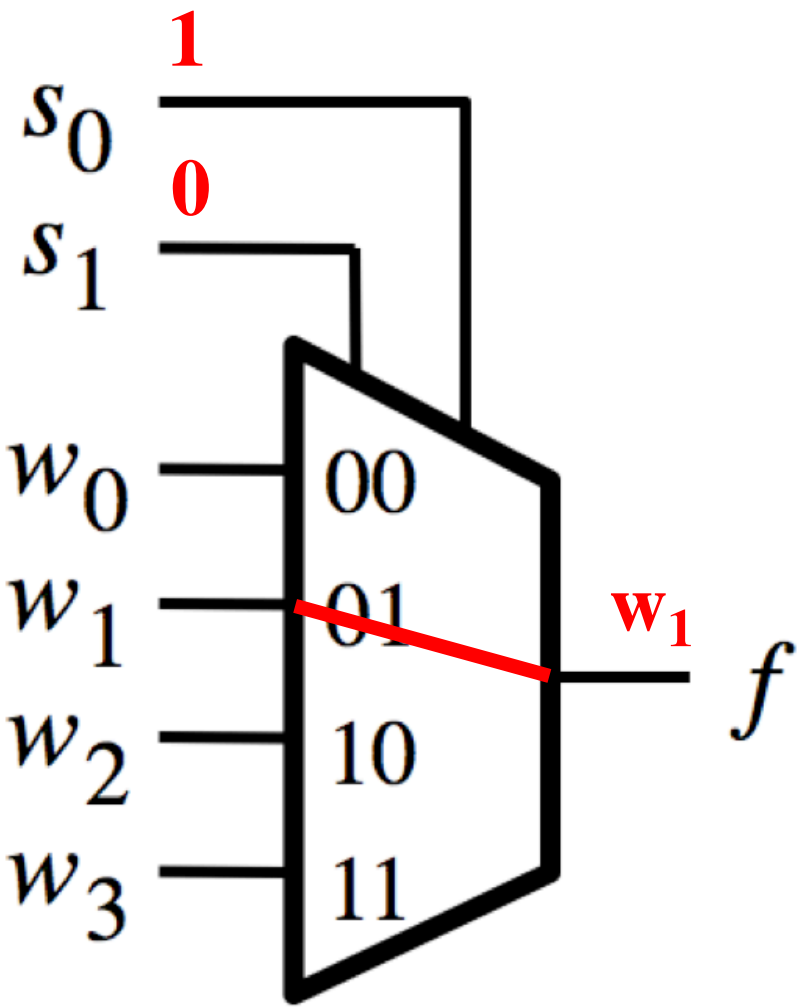
Analysis of the 4-to-1 Multiplexer (s_1=1 and s_0=1)



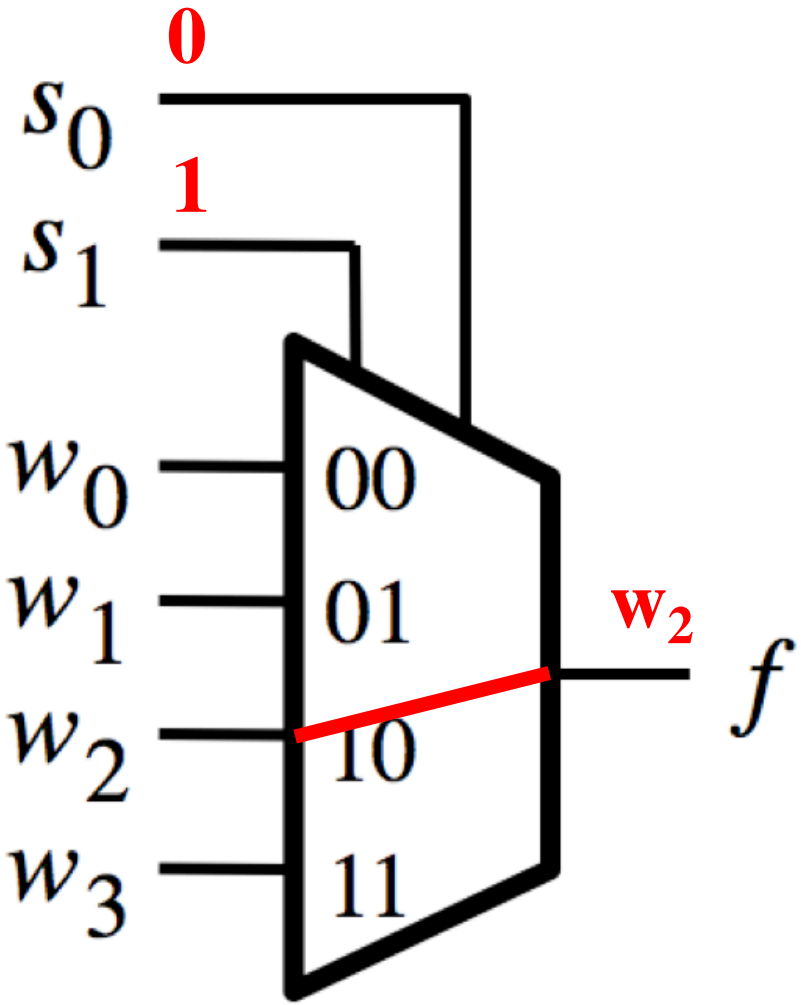
Analysis of the 4-to-1 Multiplexer (s_1=0 and s_0=0)



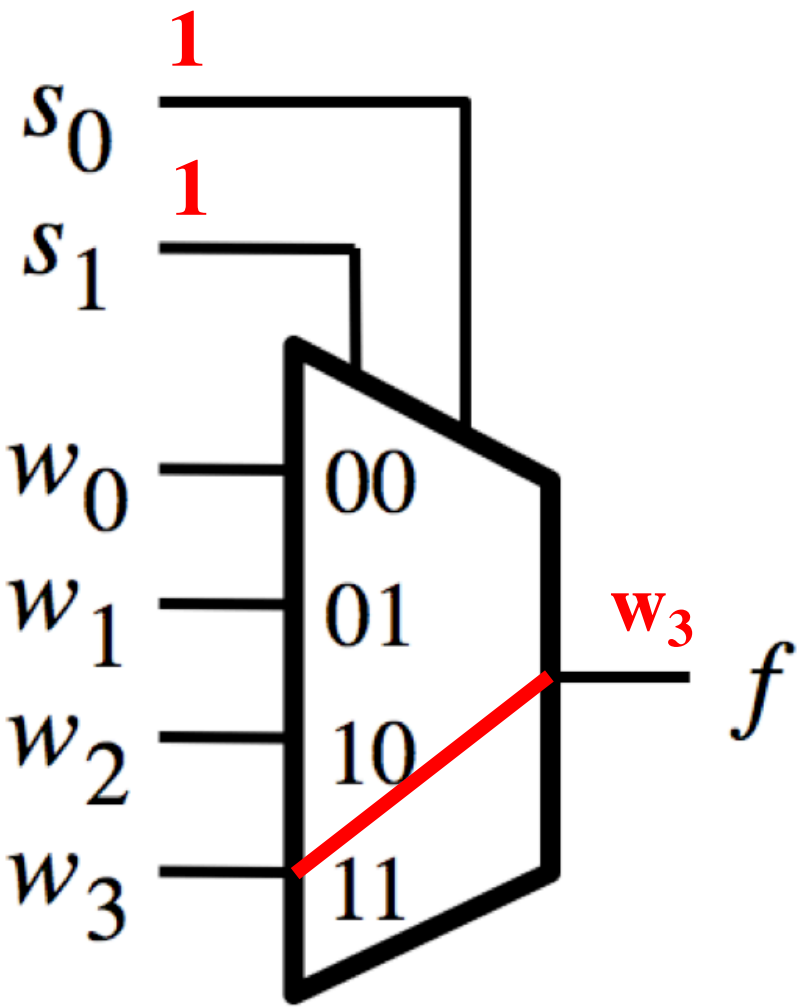
Analysis of the 4-to-1 Multiplexer (s_1=0 and s_0=1)



Analysis of the 4-to-1 Multiplexer (s_1=1 and s_0=0)

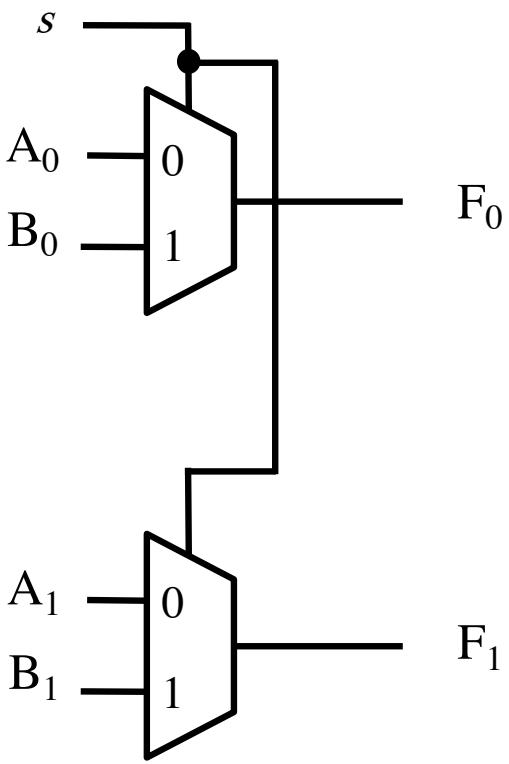


Analysis of the 4-to-1 Multiplexer
(s_1=1 and s_0=1)

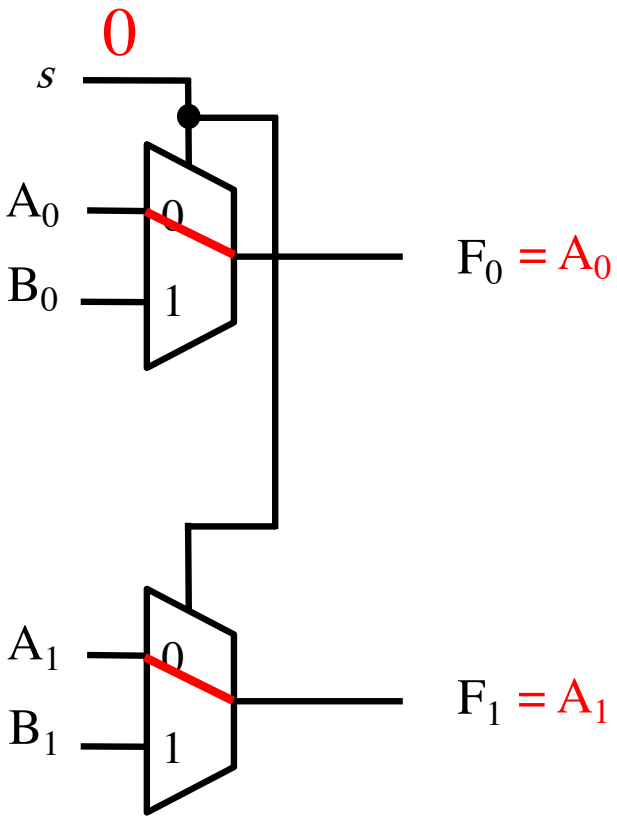


Multiplexer Tricks
(select one of two 2-bit numbers)

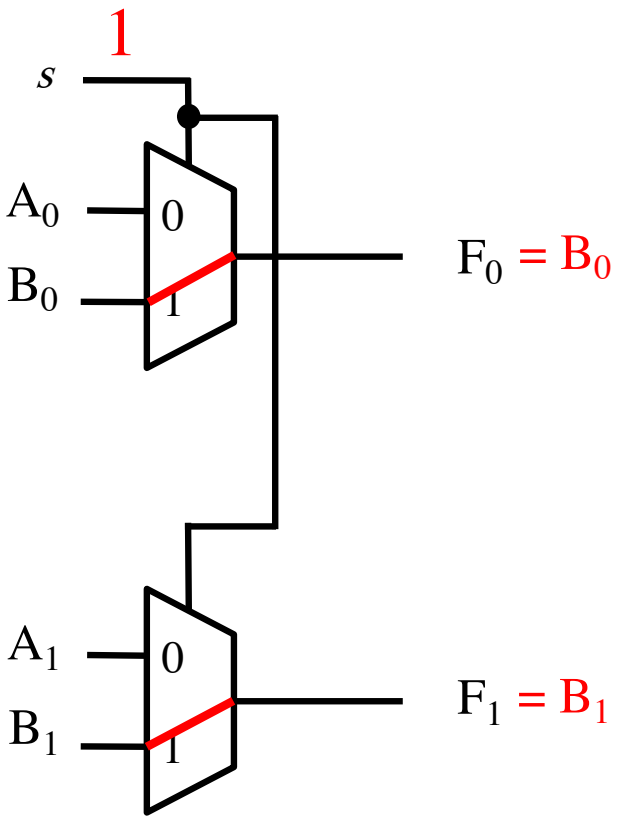
Select Either $A=A_1A_0$ or $B=B_1B_0$



Select Either $A=A_1A_0$ or $B=B_1B_0$

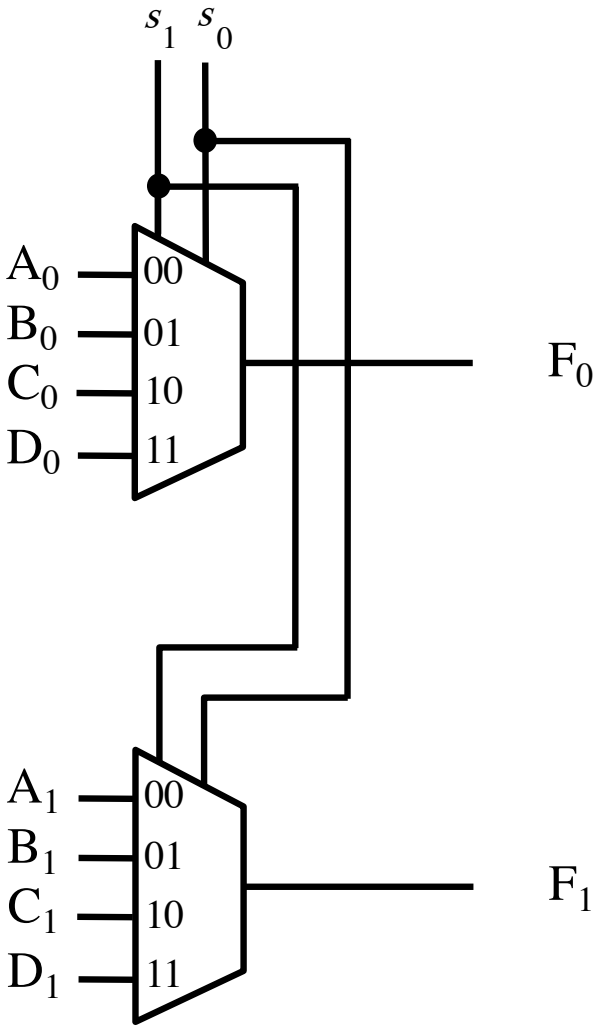


Select Either $A=A_1A_0$ or $B=B_1B_0$

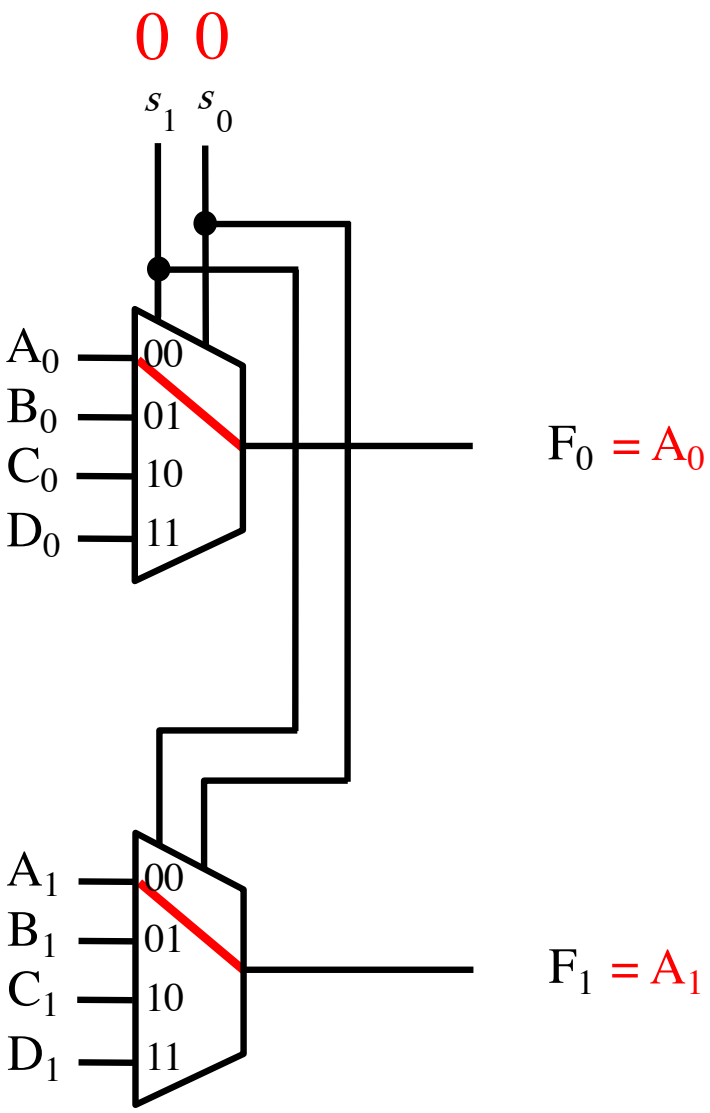


Multiplexer Tricks
(select one of four 2-bit numbers)

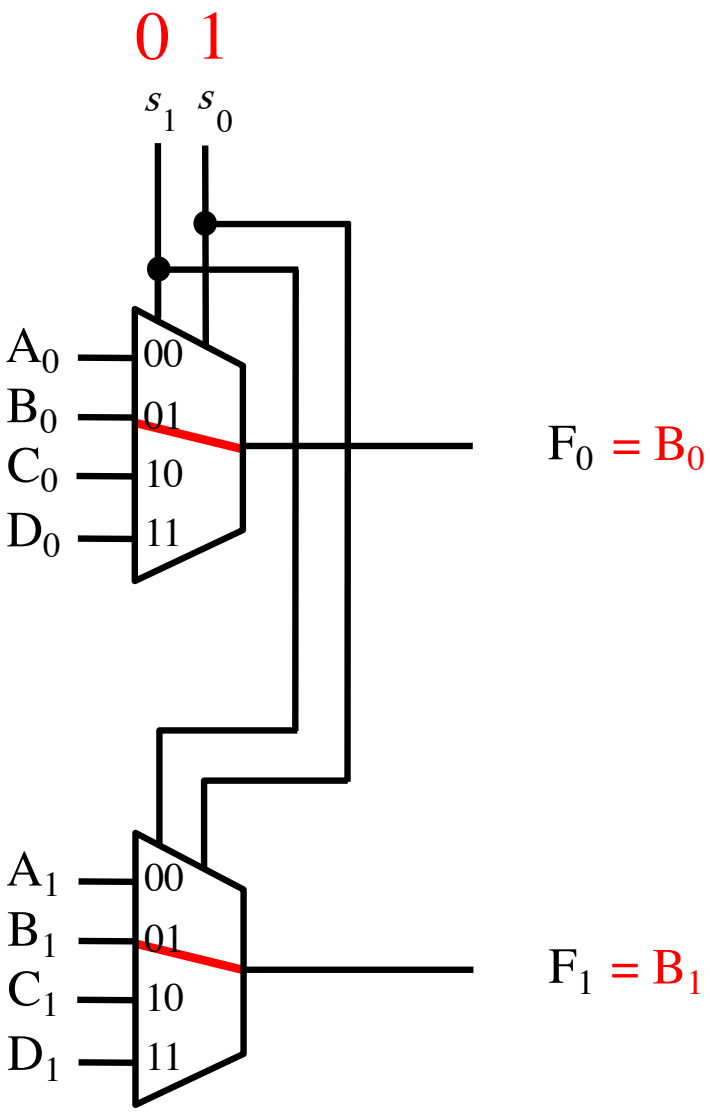
Select $A=A_1A_0$ or $B=B_1B_0$ or $C=C_1C_0$ or $D=D_1D_0$



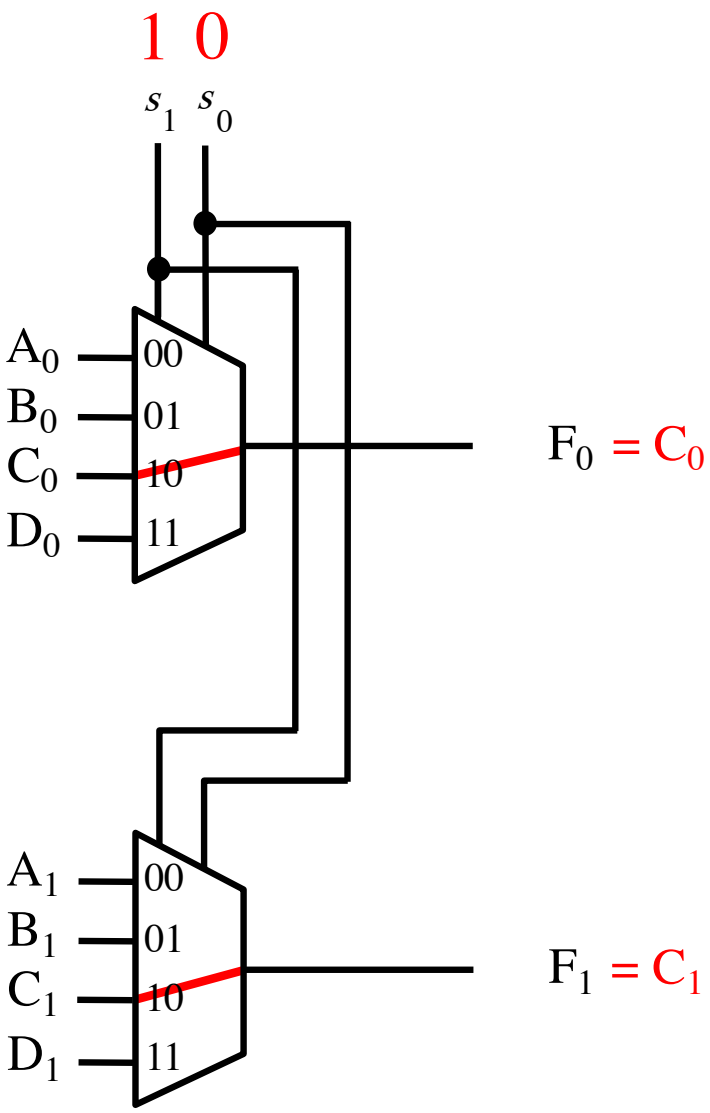
Select $A=A_1A_0$ or $B=B_1B_0$ or $C=C_1C_0$ or $D=D_1D_0$



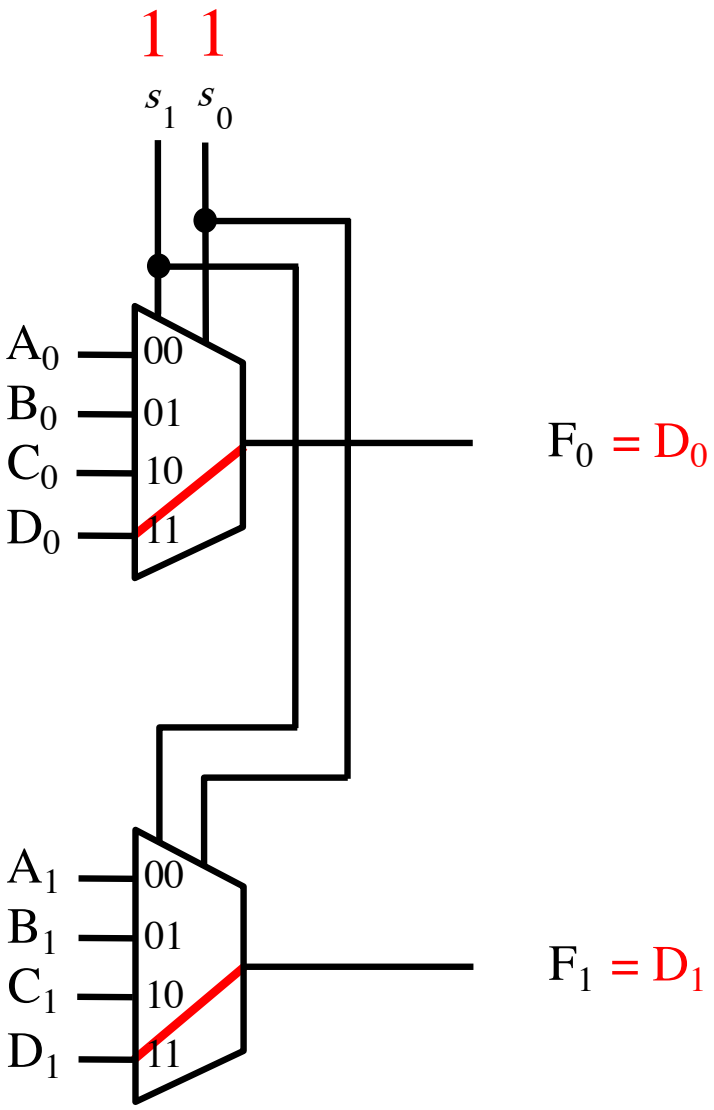
Select $A=A_1A_0$ or $B=B_1B_0$ or $C=C_1C_0$ or $D=D_1D_0$



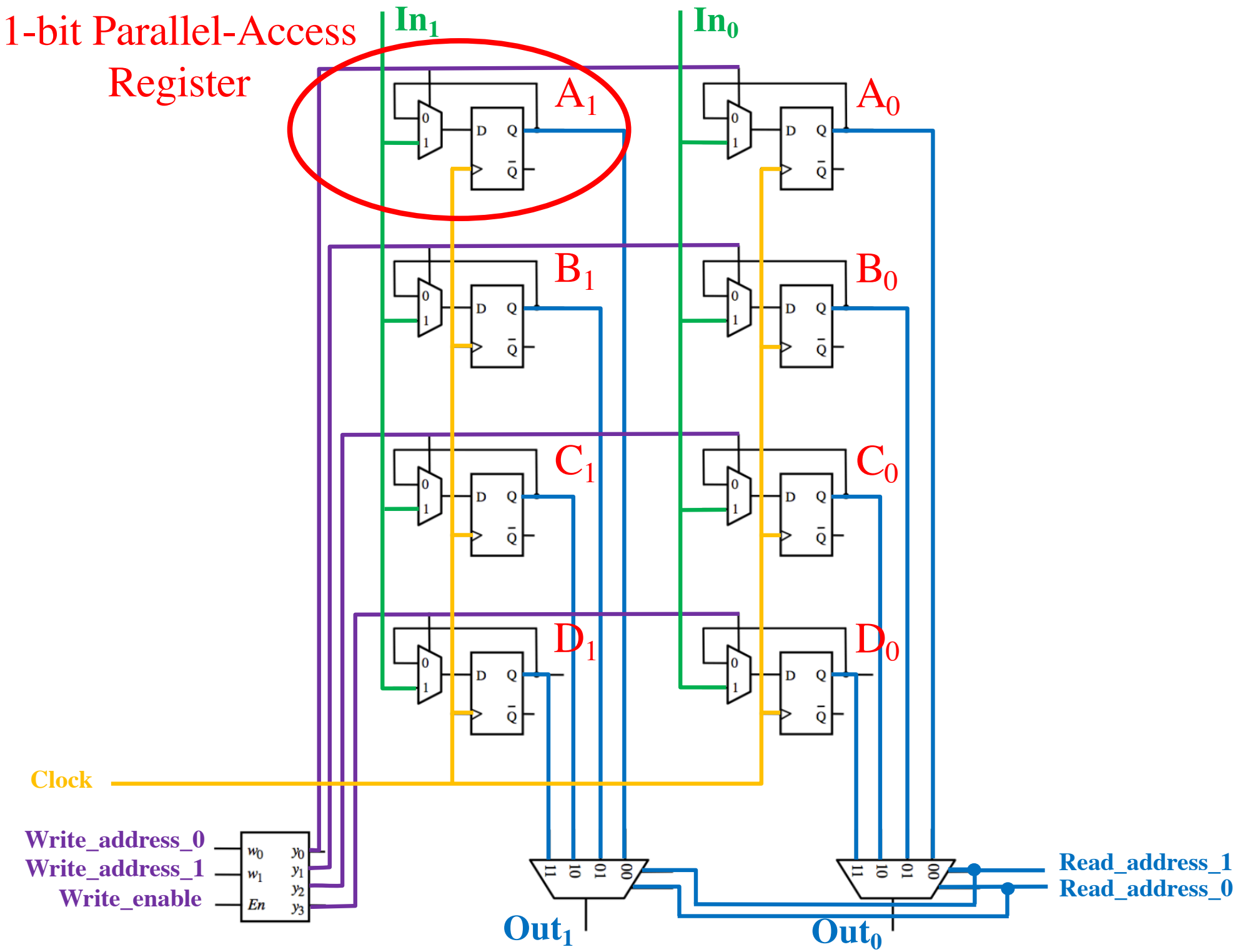
Select $A=A_1A_0$ or $B=B_1B_0$ or $C=C_1C_0$ or $D=D_1D_0$

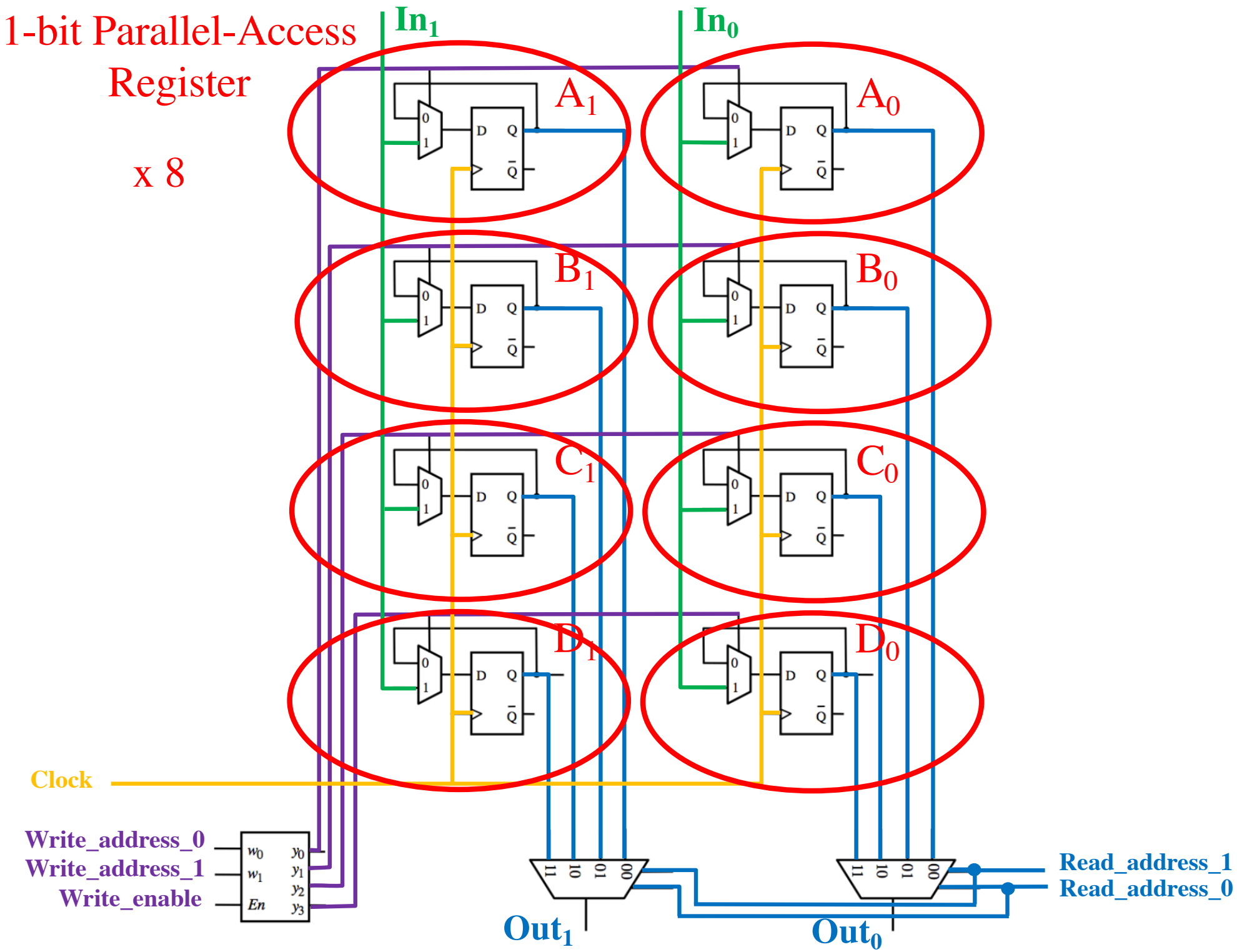


Select $A=A_1A_0$ or $B=B_1B_0$ or $C=C_1C_0$ or $D=D_1D_0$

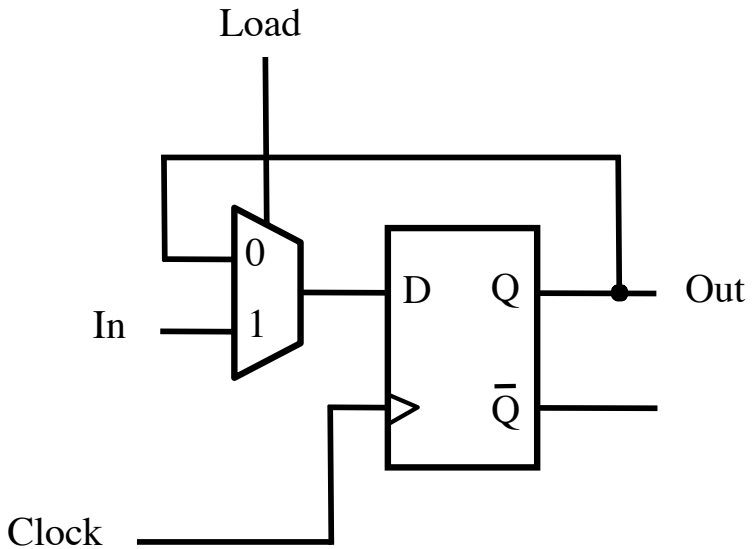


1-bit Parallel-Access Register

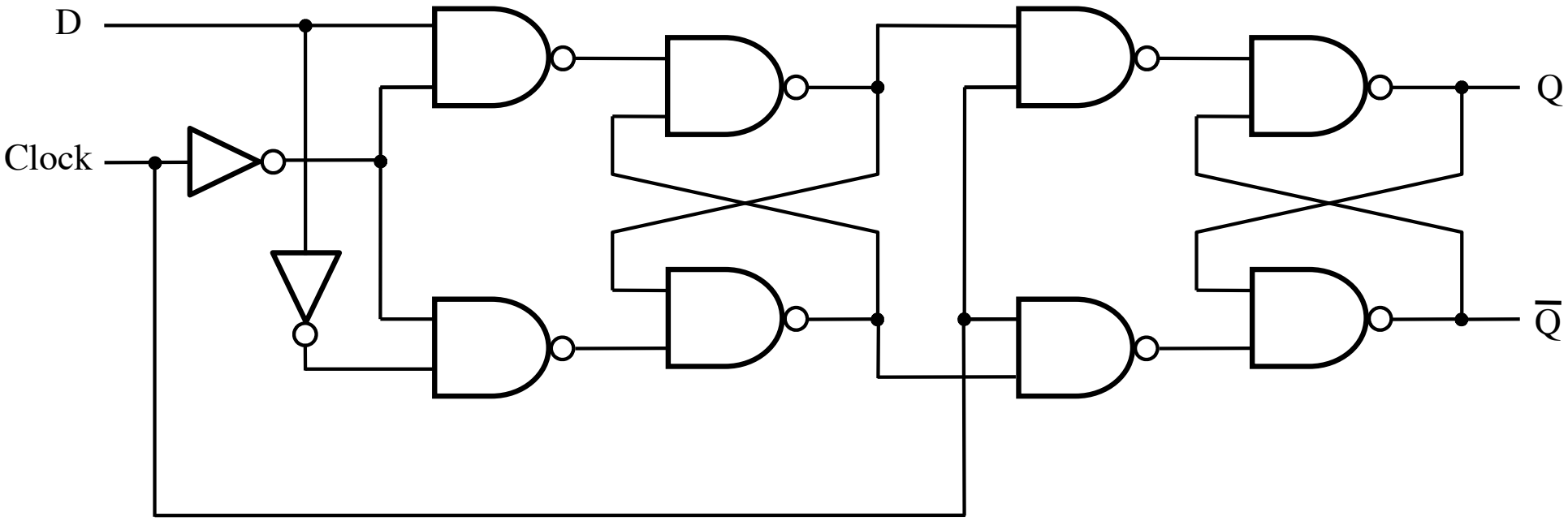

1-bit Parallel-Access Register
In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
D
Q
Q̄
0
1
Clock
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11
10
01
00
Out1
Out0
Read_address_1
Read_address_0


1-bit Parallel-Access Register
x 8
In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
D
Q
Q̄
0
1
Clock
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11
10
01
00
Out1
Out0
Read_address_1
Read_address_0

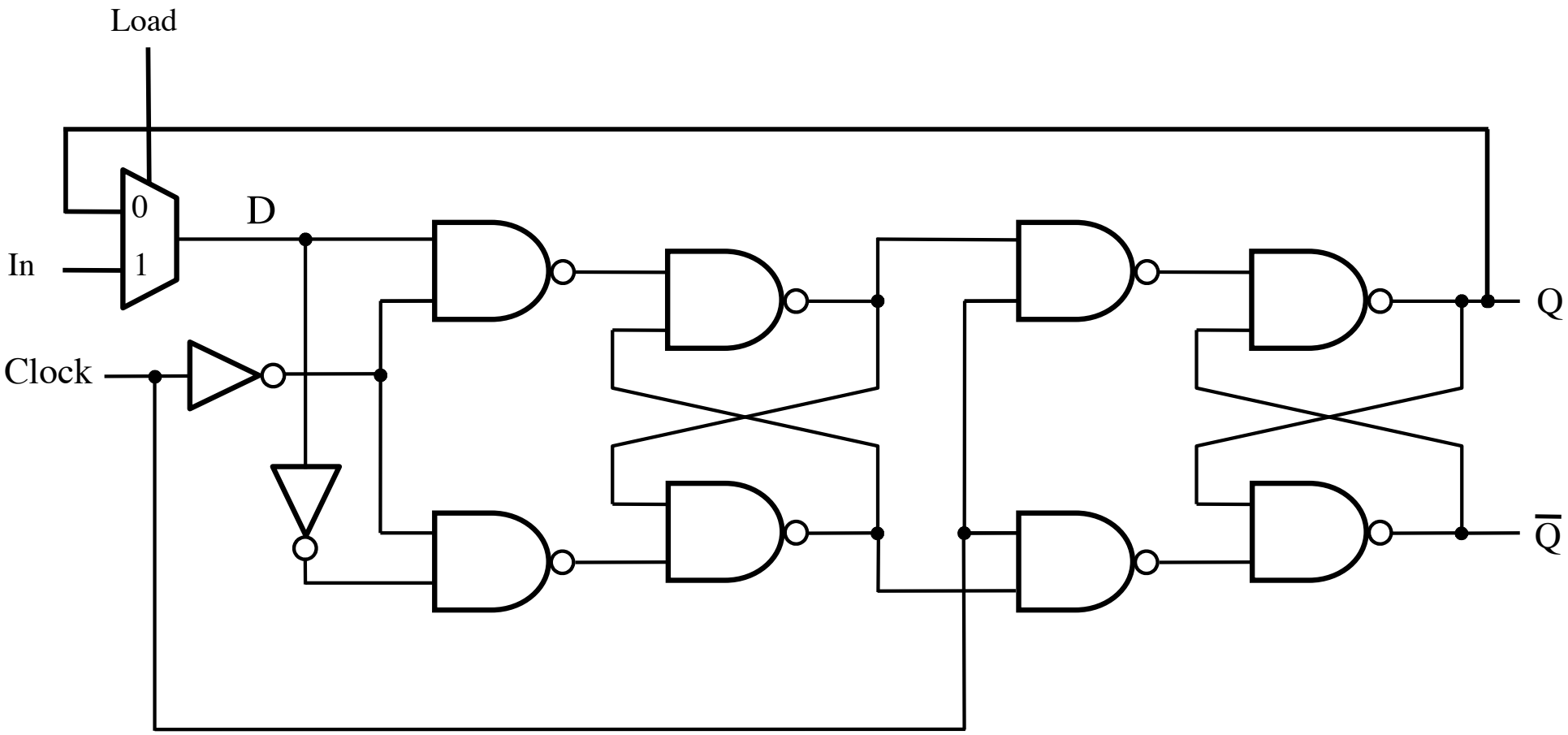
1-Bit Parallel-Access Register



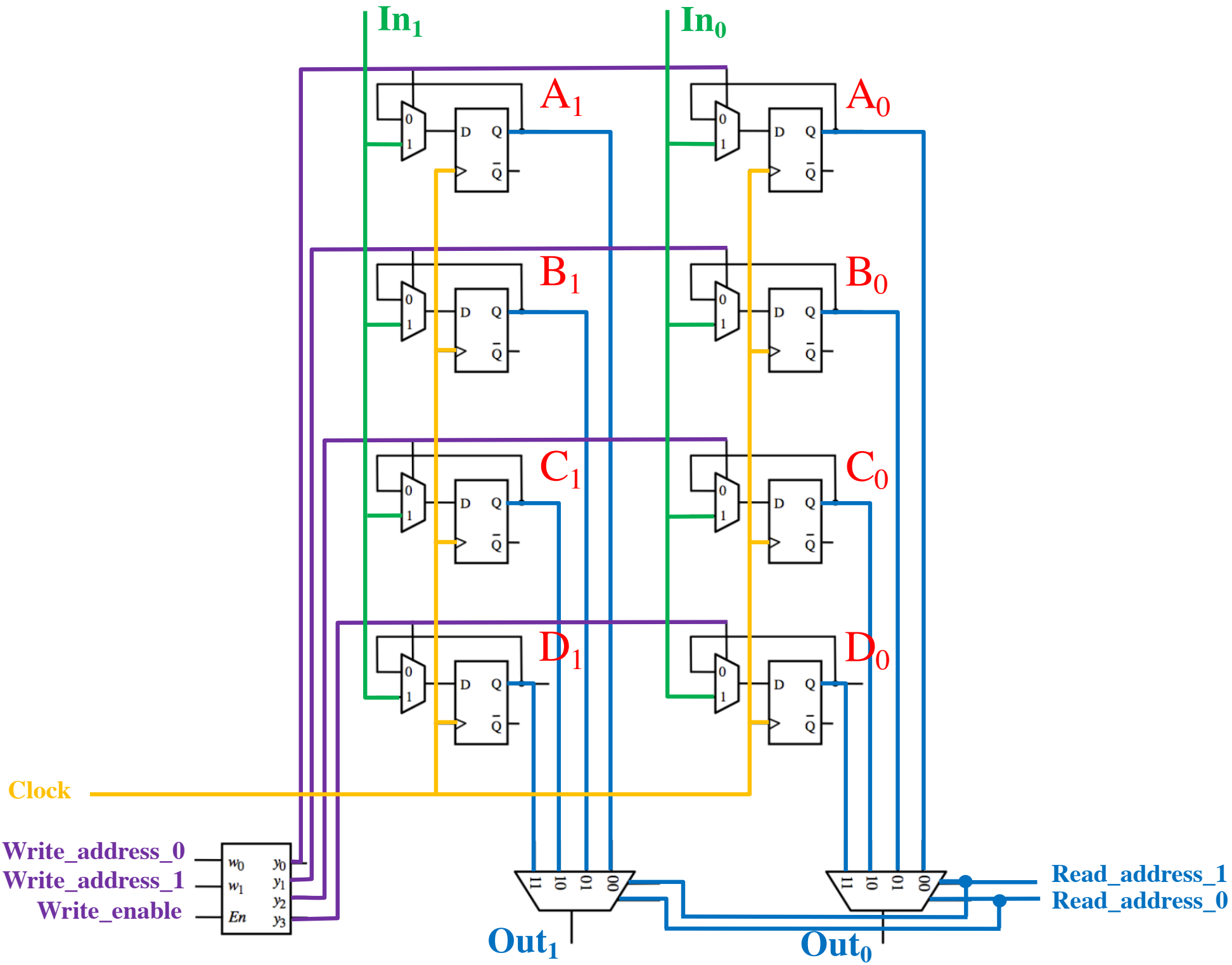
Positive-Edge-Triggered D Flip-Flop



1-bit Parallel-Access Register

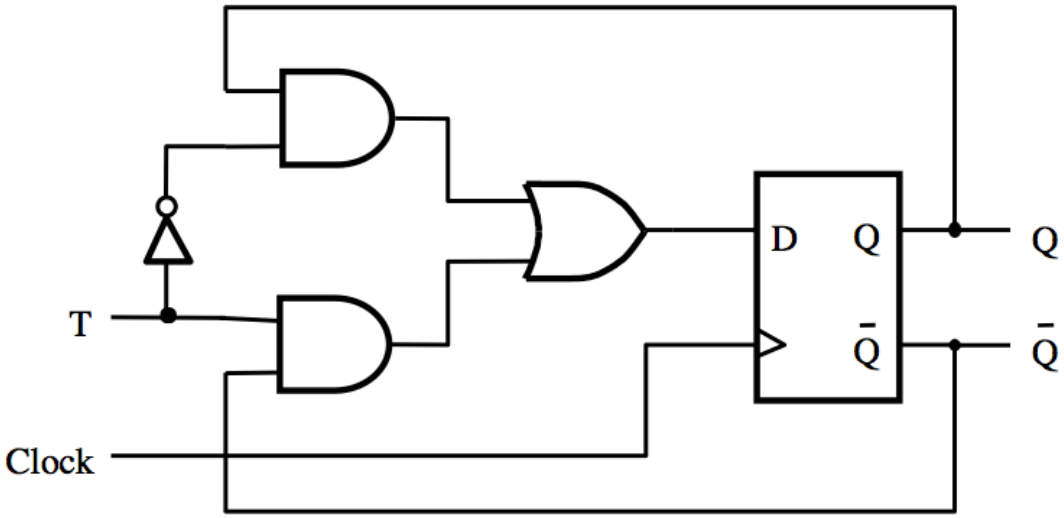


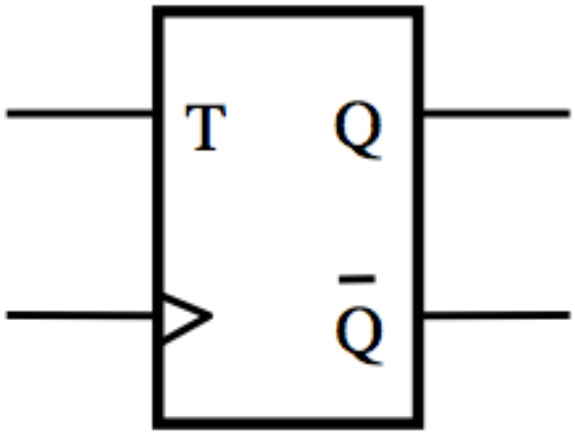
Putting it all Together


In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
D
Q
Q̄
0
1
Clock
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11
10
01
00
Out1
Out0
Read_address_1
Read_address_0

Counters

T Flip-Flop
(circuit and graphical symbol)

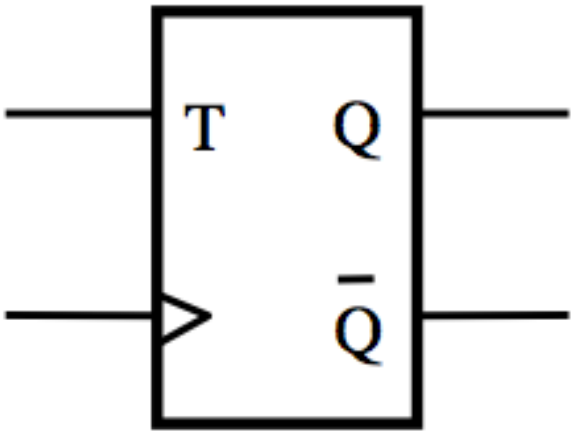




[Figure 5.15a,c from the textbook]

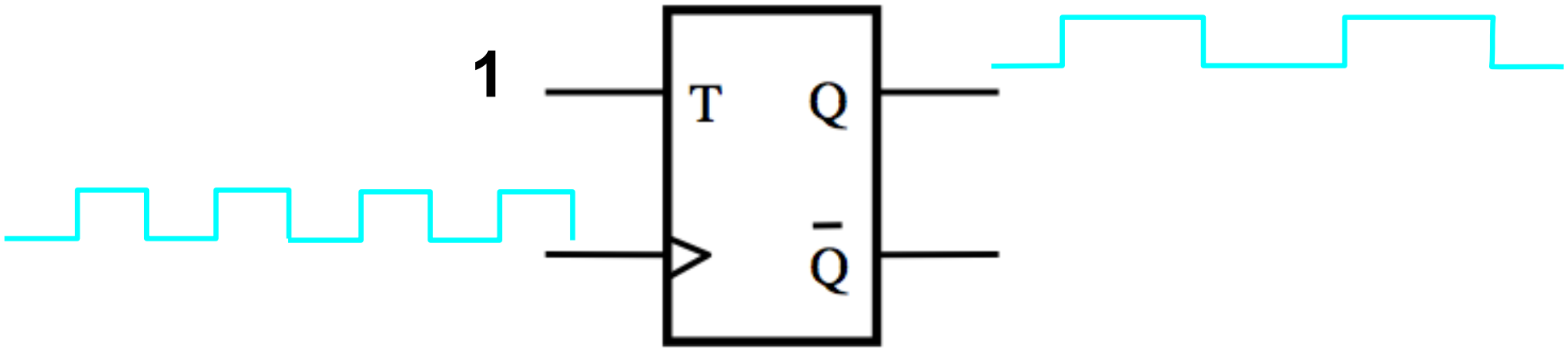
The output of the T Flip-Flop

divides the frequency of the clock by 2

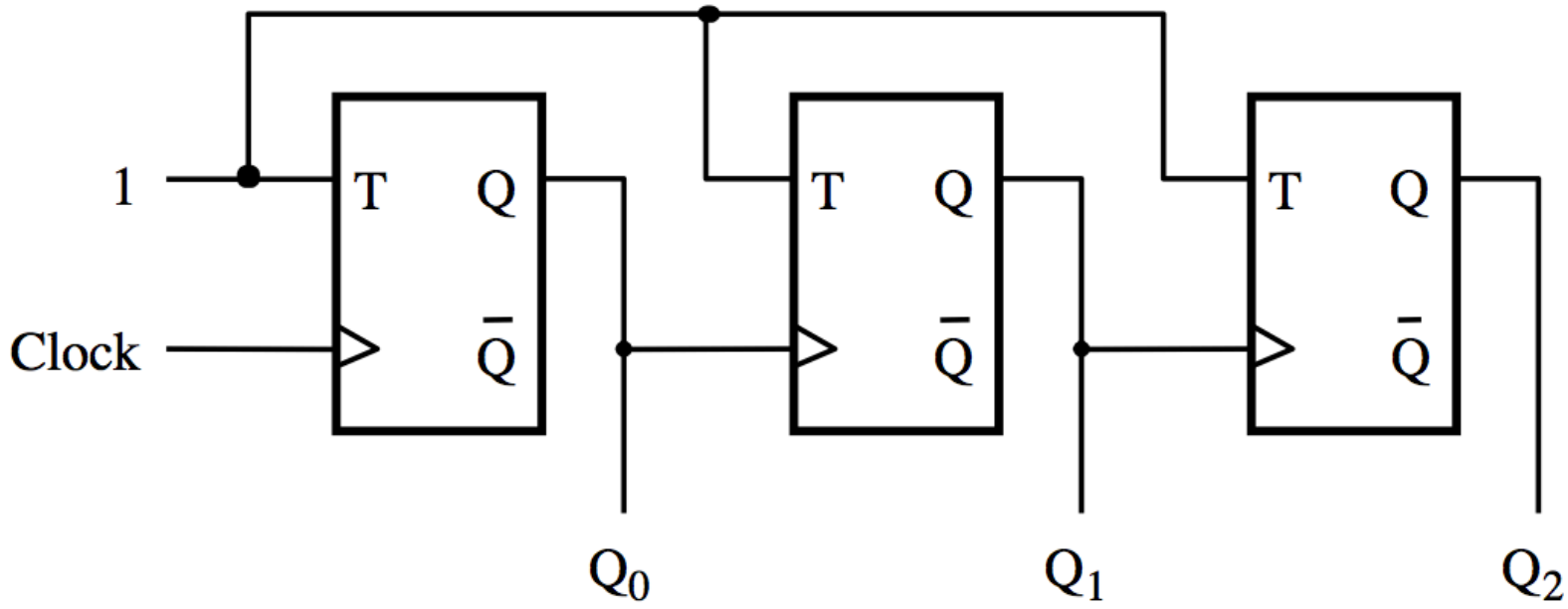


The output of the T Flip-Flop

divides the frequency of the clock by 2

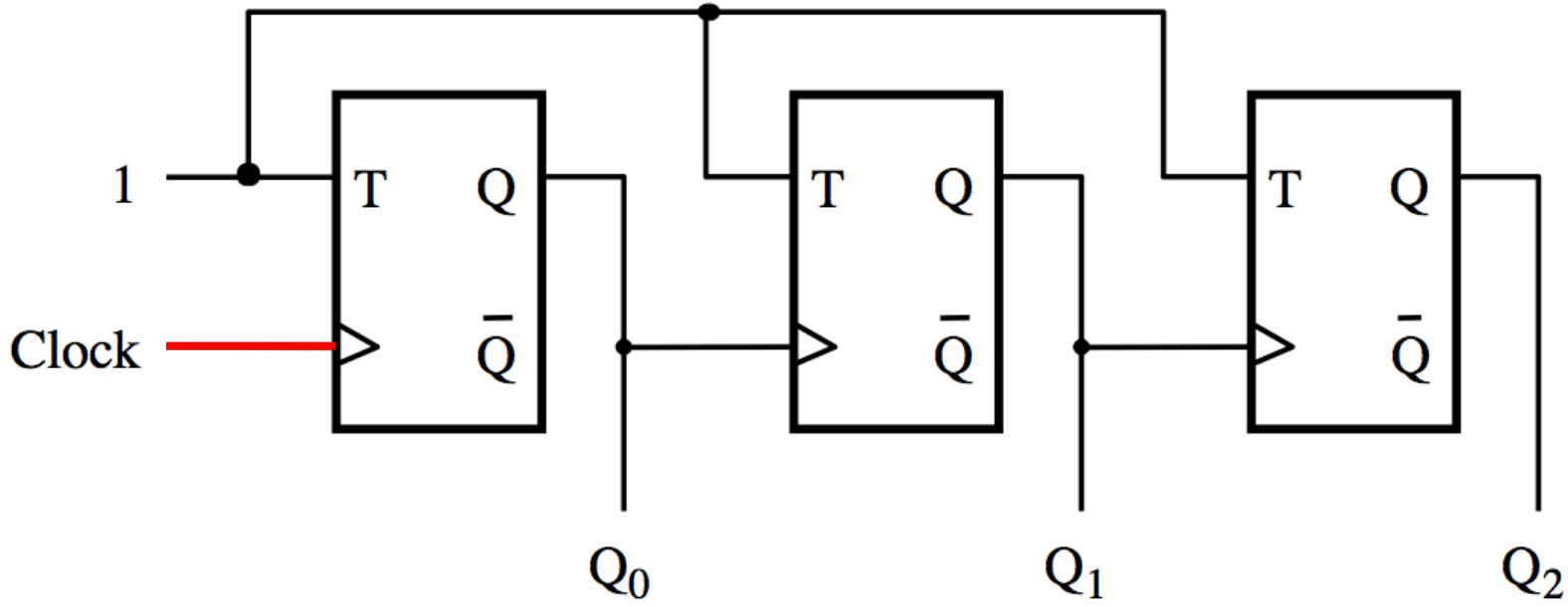


A three-bit down-counter



[Figure 5.20 from the textbook]

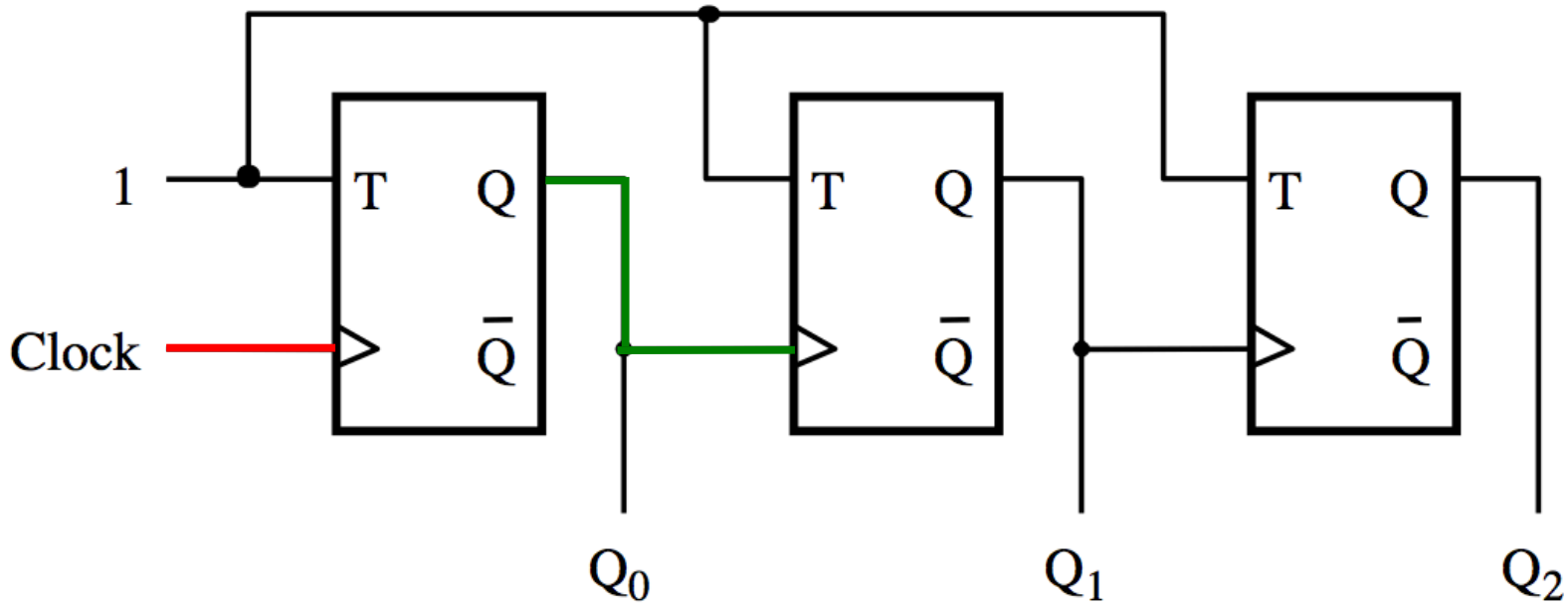
A three-bit down-counter



The first flip-flop changes
on the positive edge of the clock

[Figure 5.20 from the textbook]

A three-bit down-counter

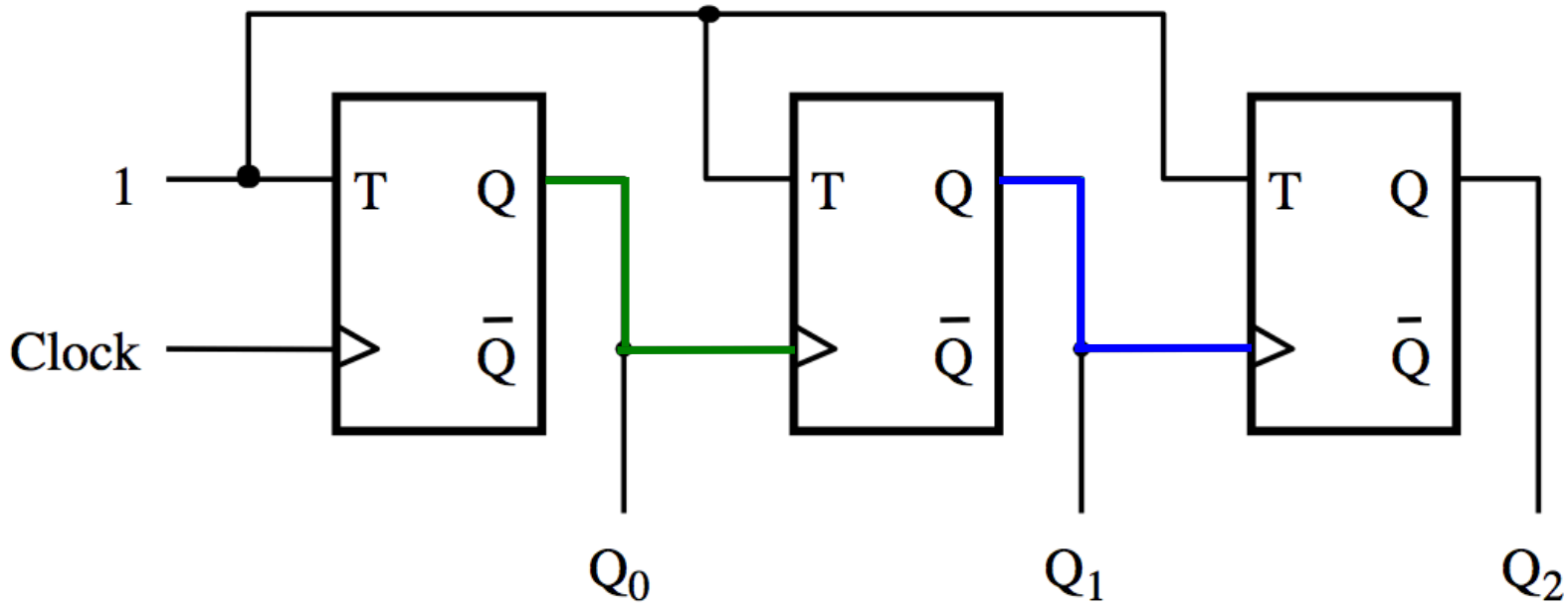


The first flip-flop changes on the positive edge of the clock

The second flip-flop changes on the positive edge of Q_0

[Figure 5.20 from the textbook]

A three-bit down-counter



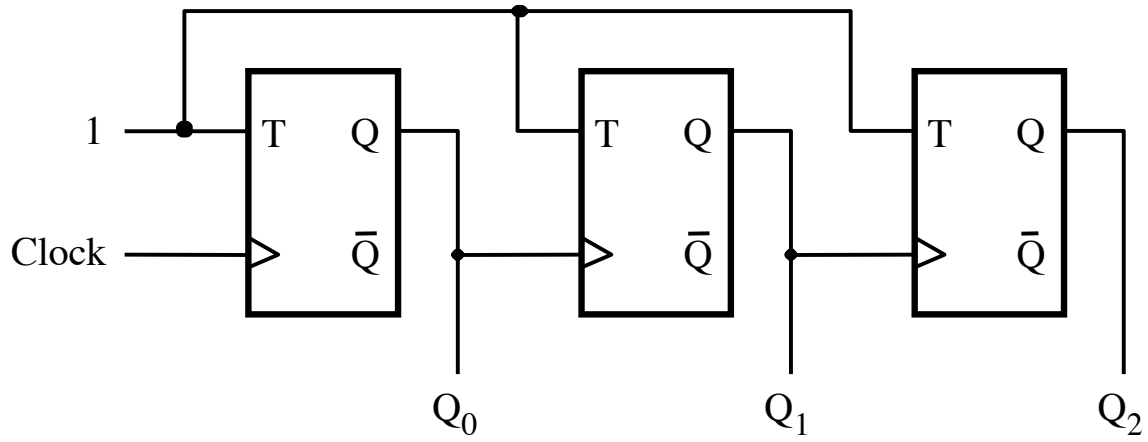
The first flip-flop changes on the positive edge of the clock

The second flip-flop changes on the positive edge of Q_0

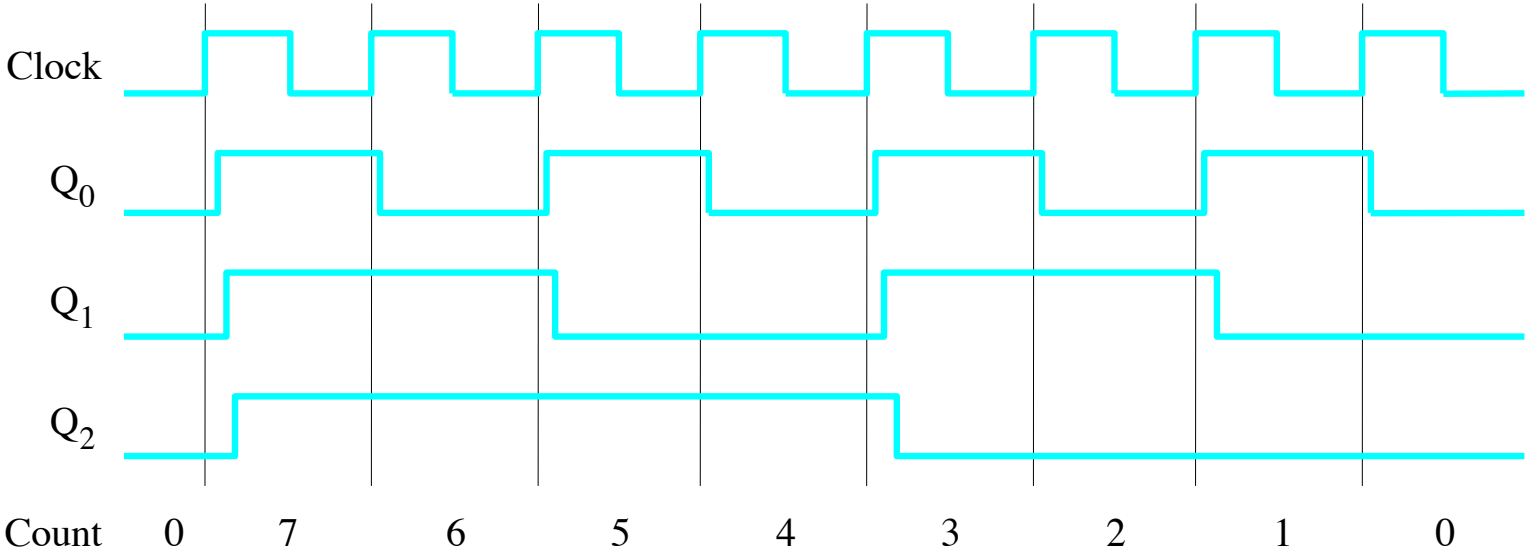
The third flip-flop changes on the positive edge of Q_1

[Figure 5.20 from the textbook]

A three-bit down-counter



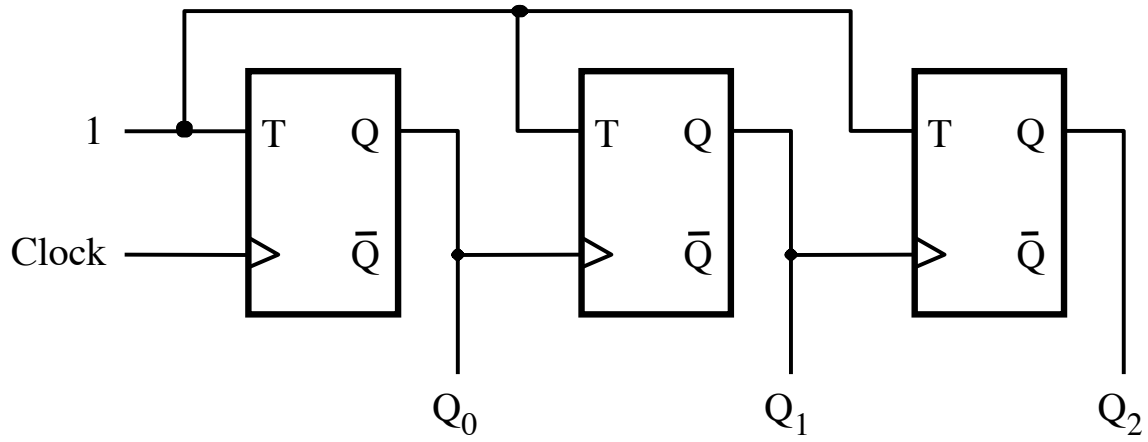
(a) Circuit



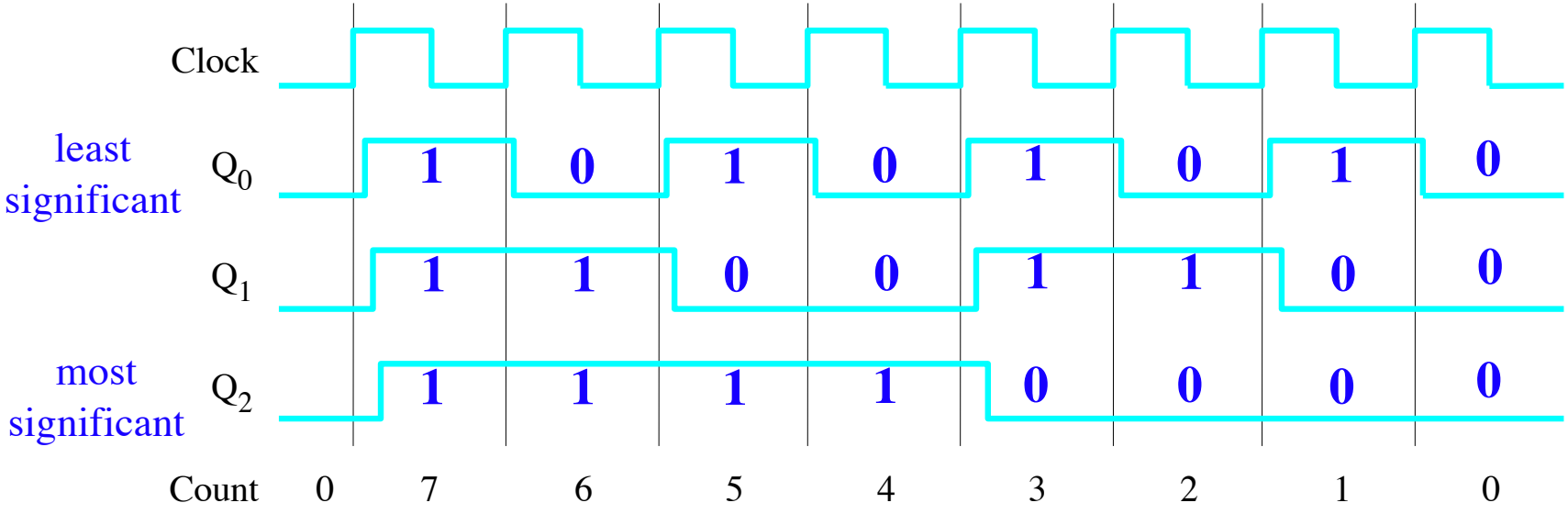
(b) Timing diagram

[Figure 5.20 from the textbook]

A three-bit down-counter

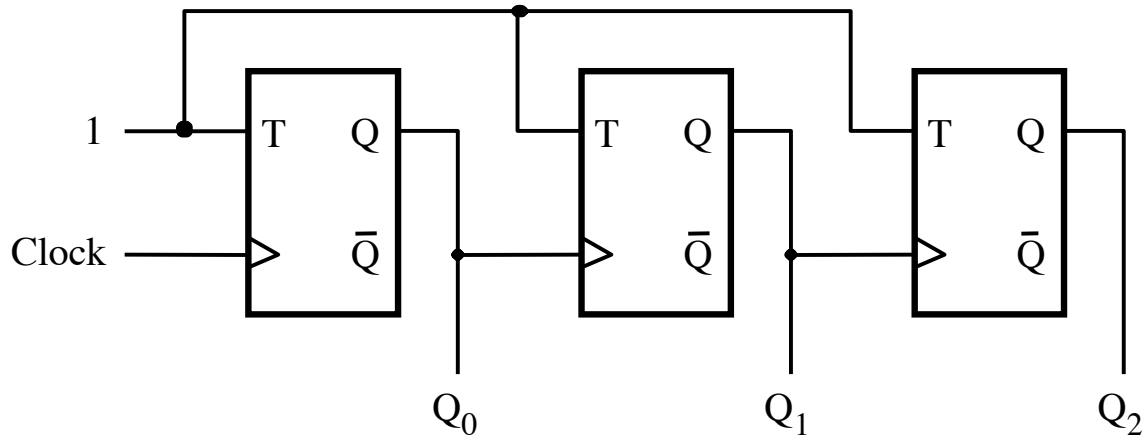


(a) Circuit

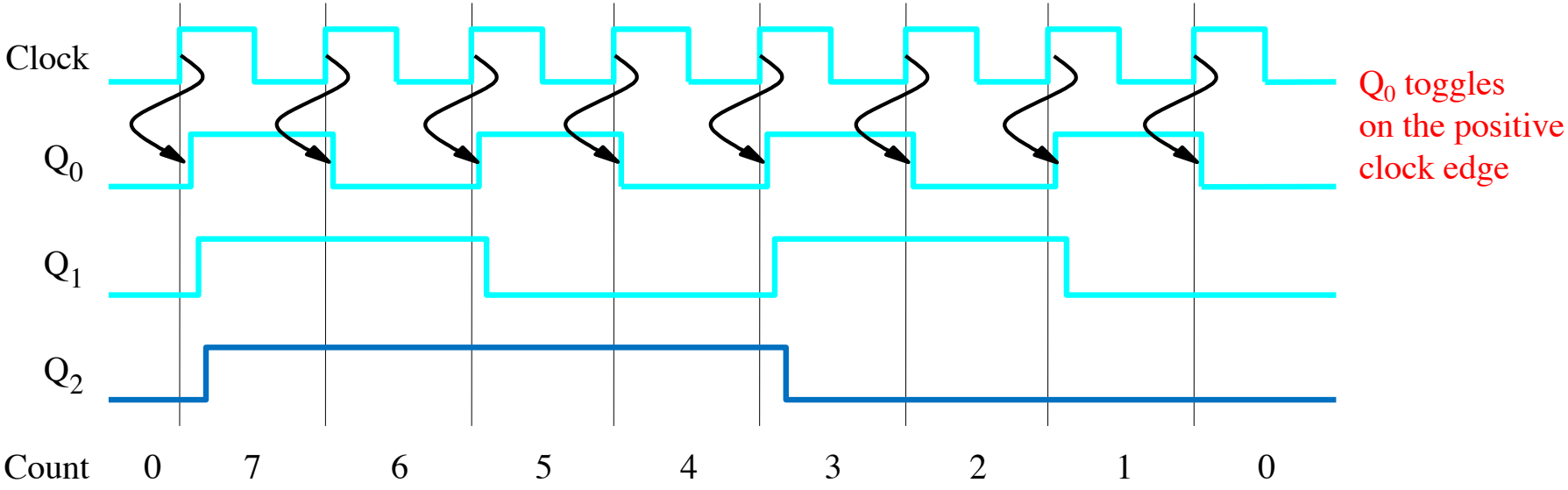


(b) Timing diagram

A three-bit down-counter

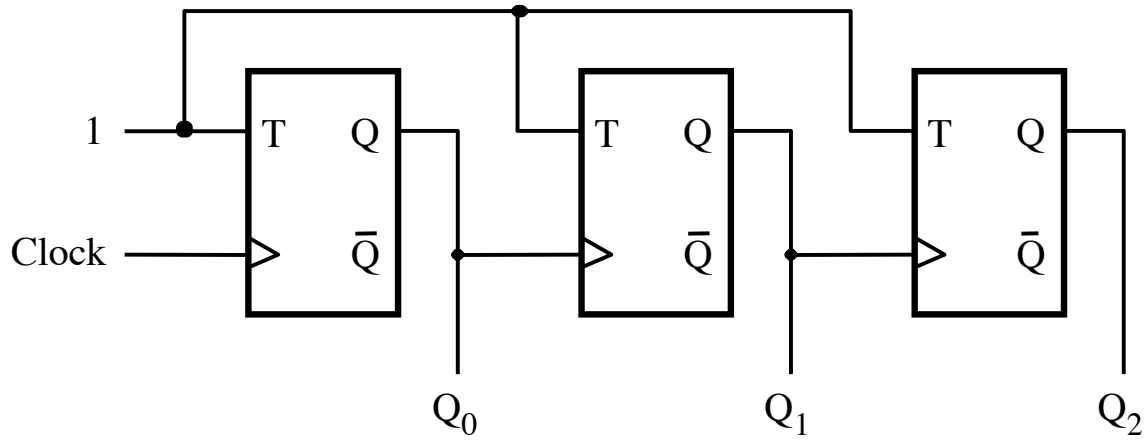


(a) Circuit

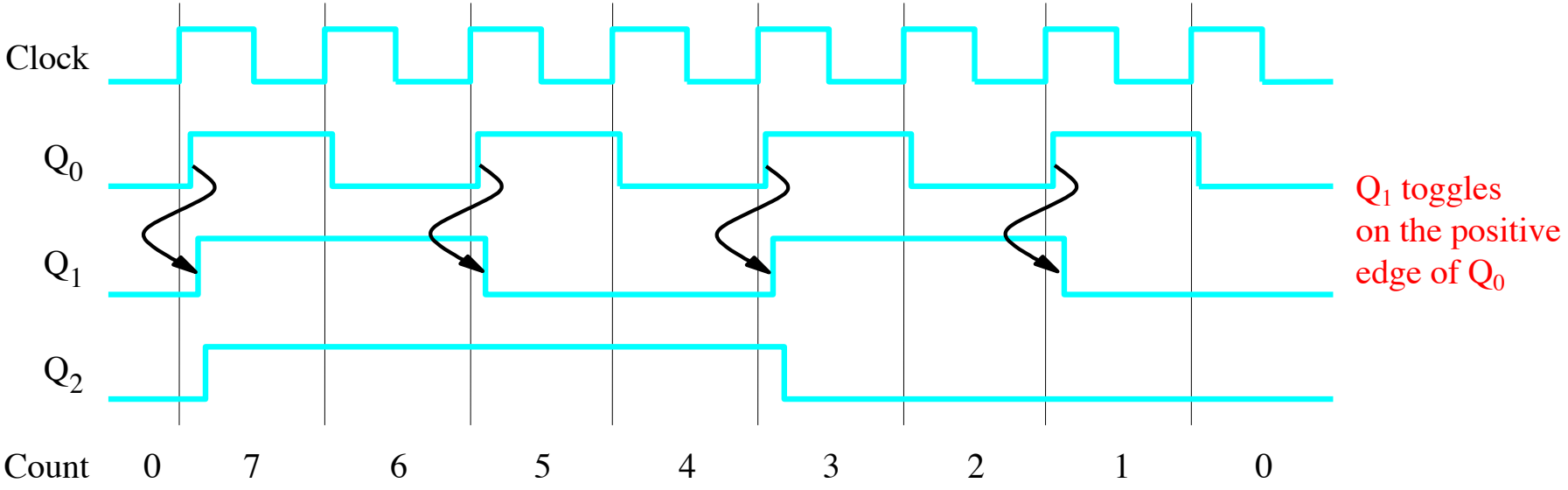


(b) Timing diagram

A three-bit down-counter

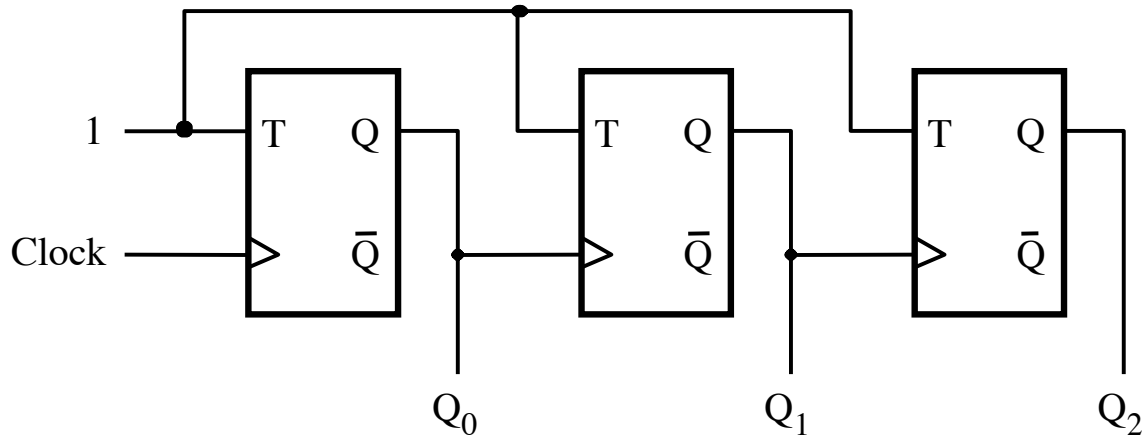


(a) Circuit

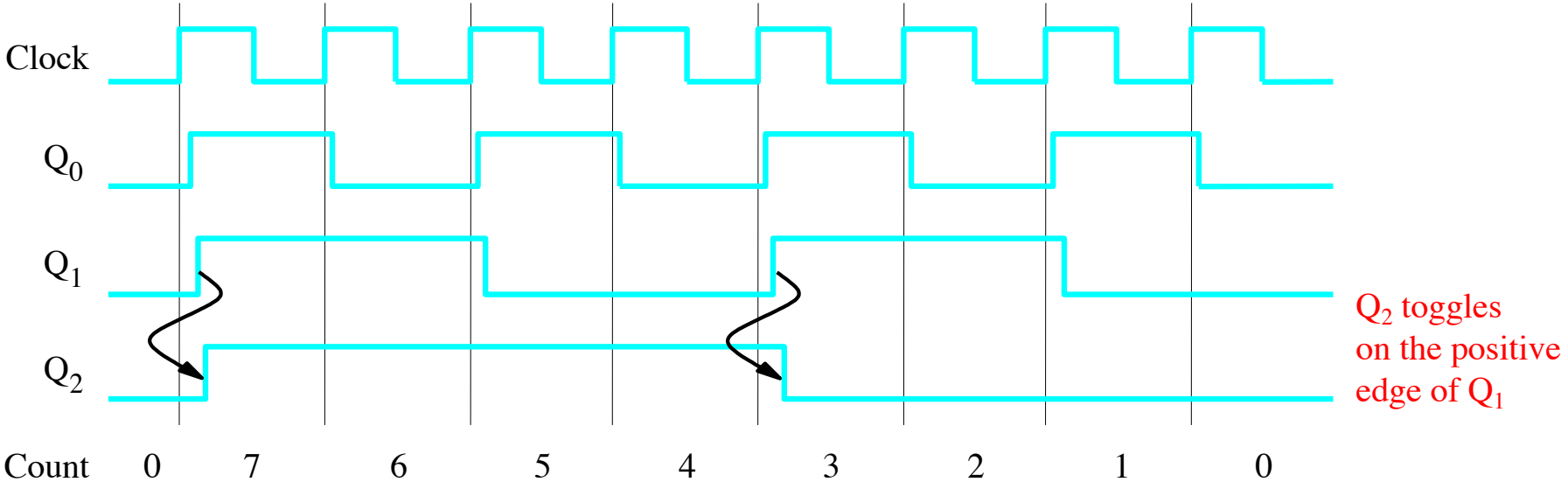


(b) Timing diagram

A three-bit down-counter

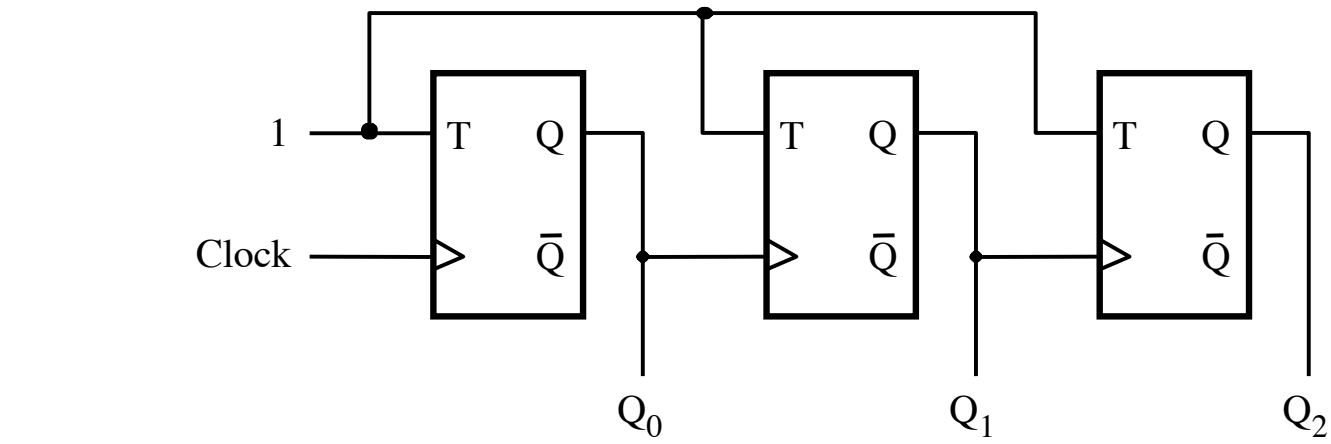


(a) Circuit

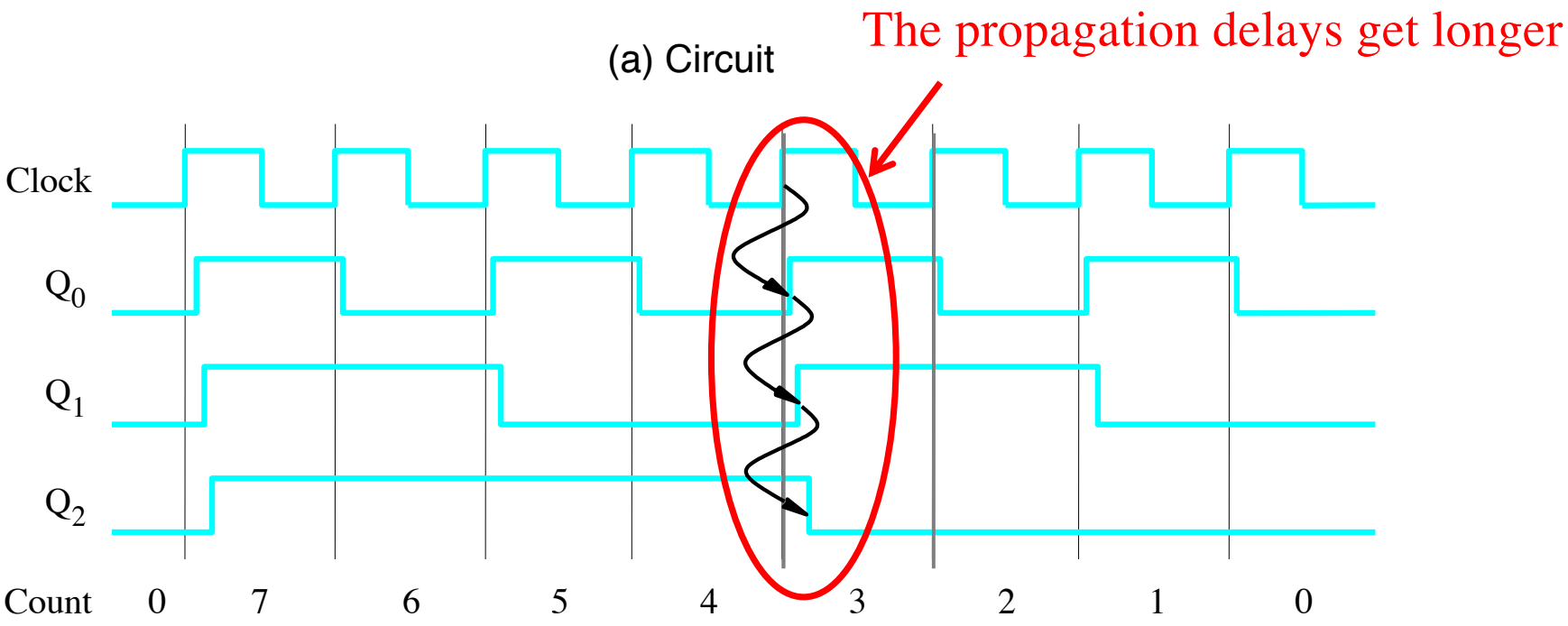


(b) Timing diagram

A three-bit down-counter

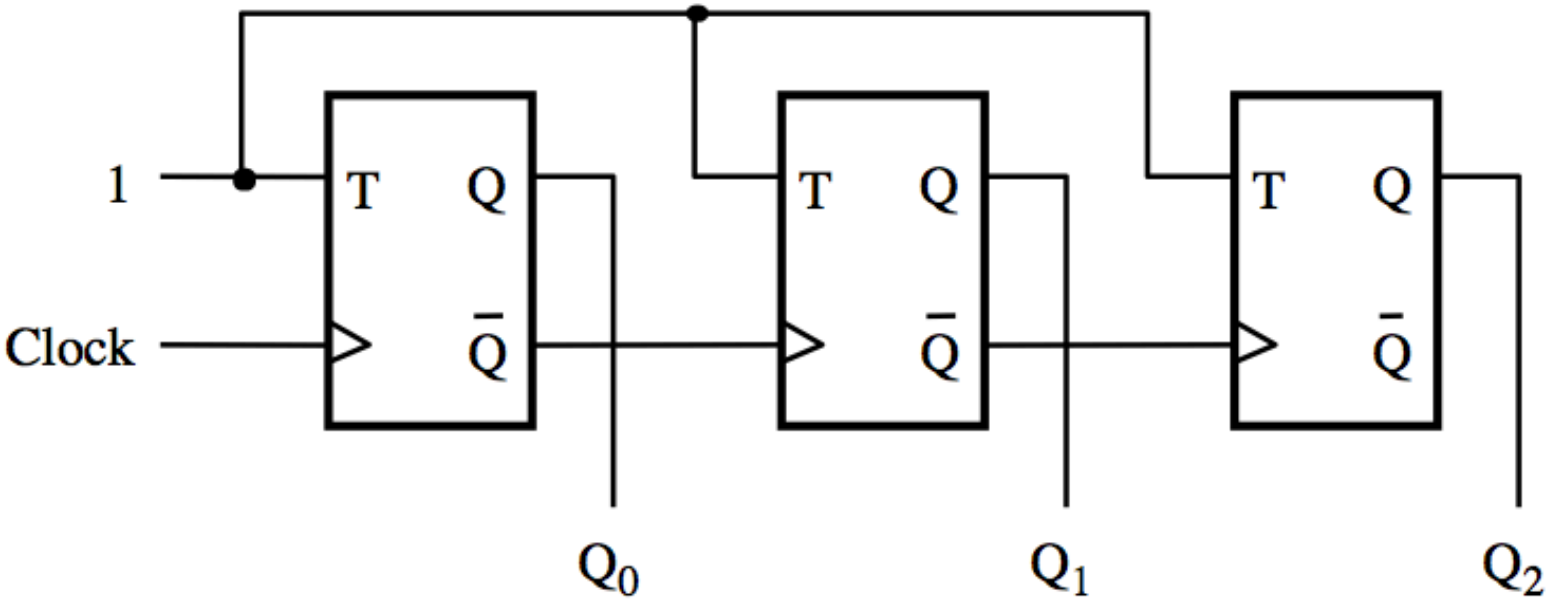


(a) Circuit



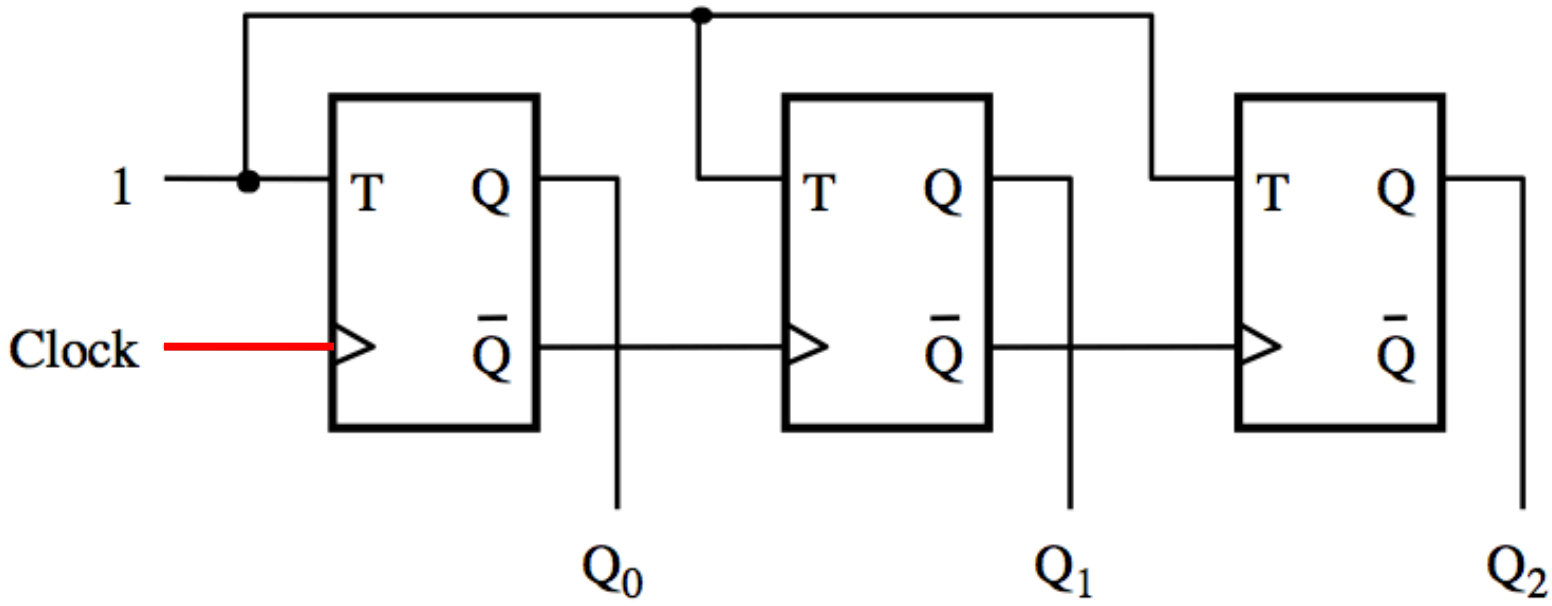
(b) Timing diagram

A three-bit up-counter



[Figure 5.19 from the textbook]

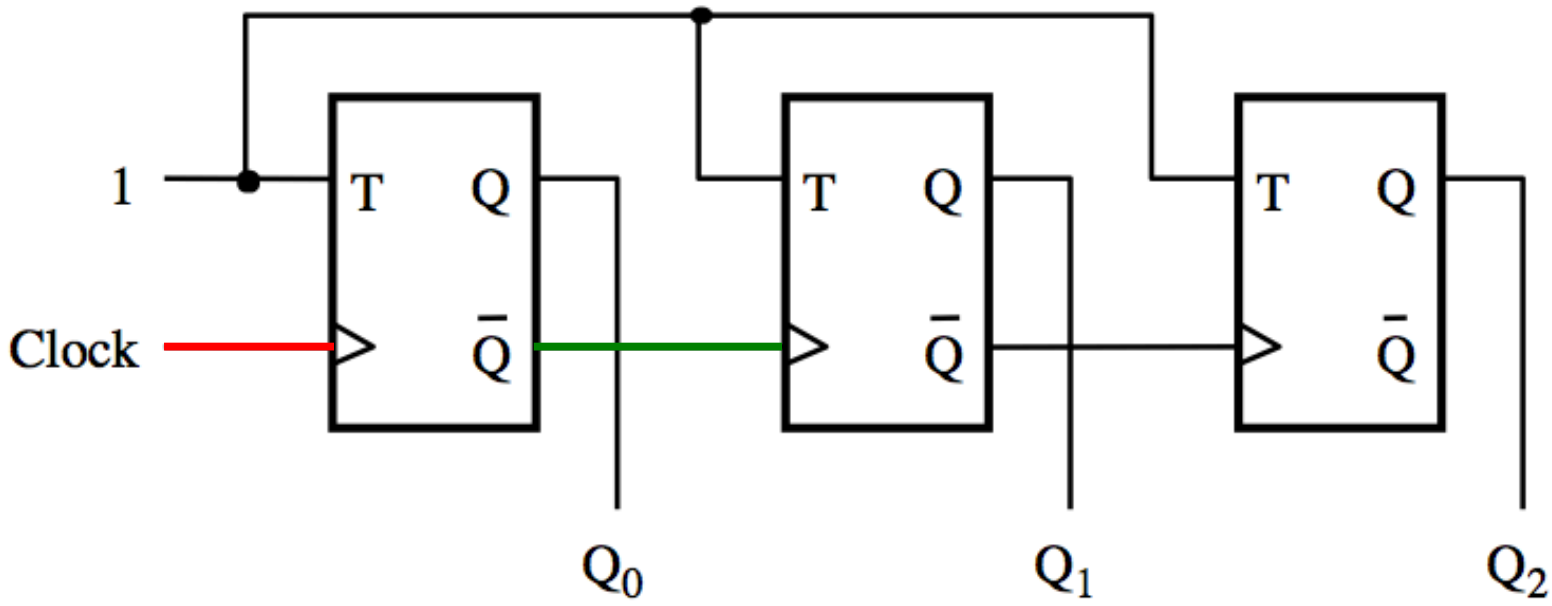
A three-bit up-counter



The first flip-flop changes
on the positive edge of the clock

[Figure 5.19 from the textbook]

A three-bit up-counter

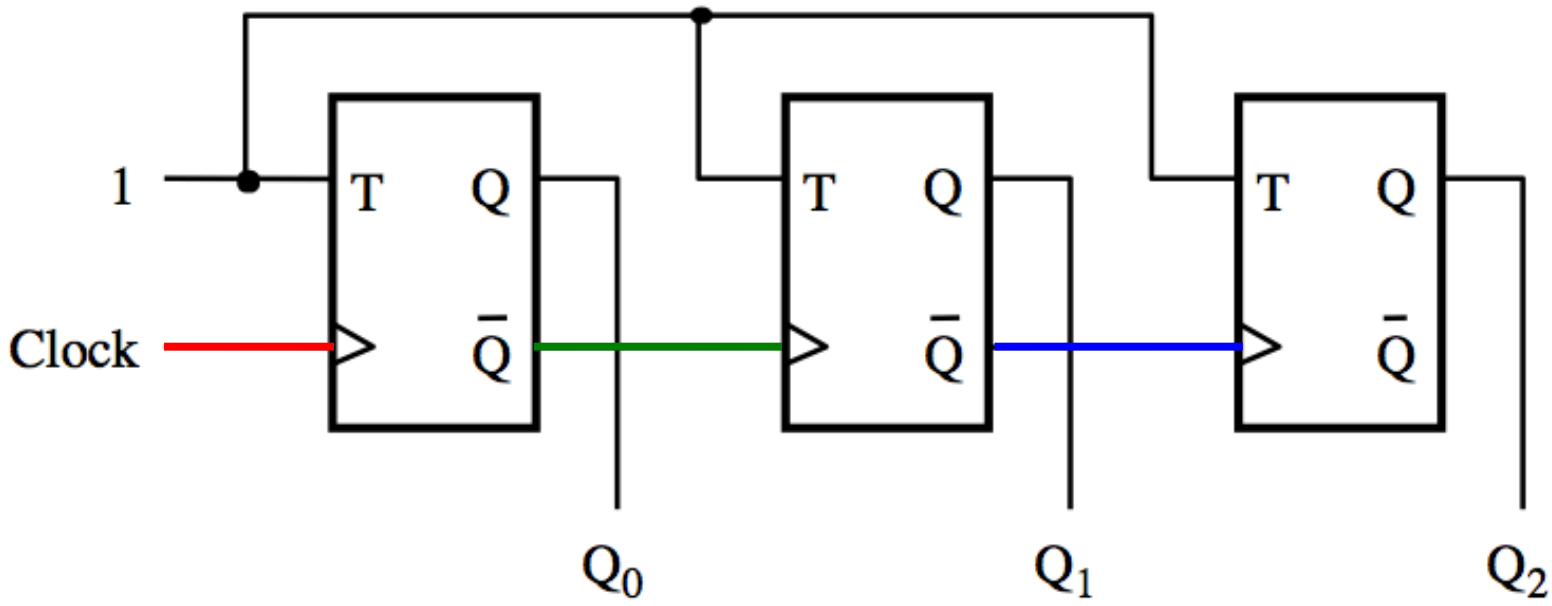


The first flip-flop changes
on the positive edge of the clock

The second flip-flop changes
on the positive edge of $\overline{Q}_0$

[Figure 5.19 from the textbook]

A three-bit up-counter



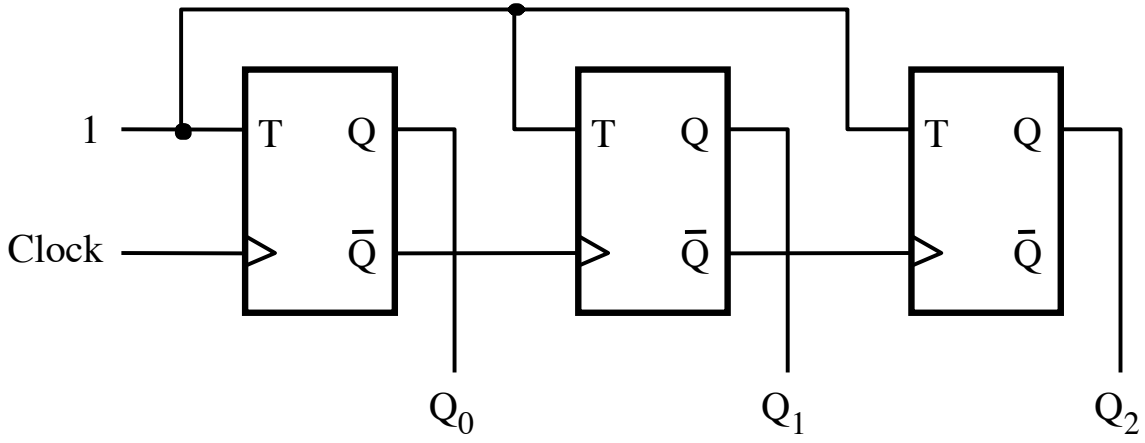
The first flip-flop changes on the positive edge of the clock

The second flip-flop changes on the positive edge of $\overline{Q}_0$

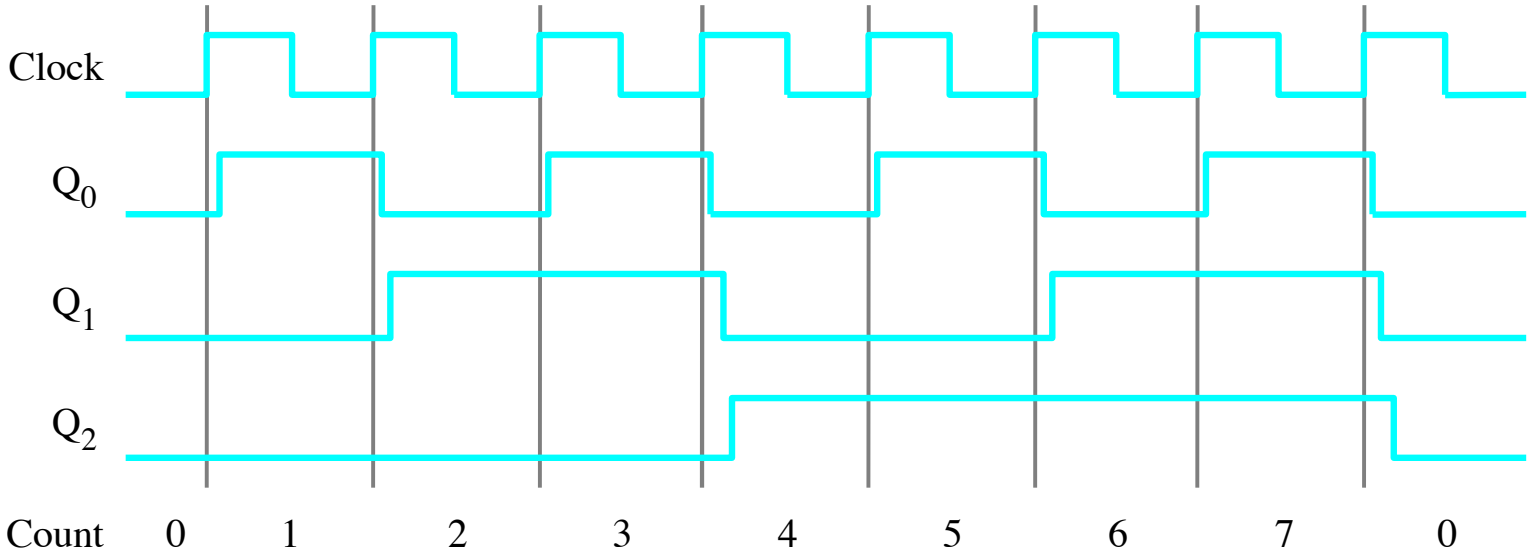
The third flip-flop changes on the positive edge of $\overline{Q}_1$

[Figure 5.19 from the textbook]

A three-bit up-counter



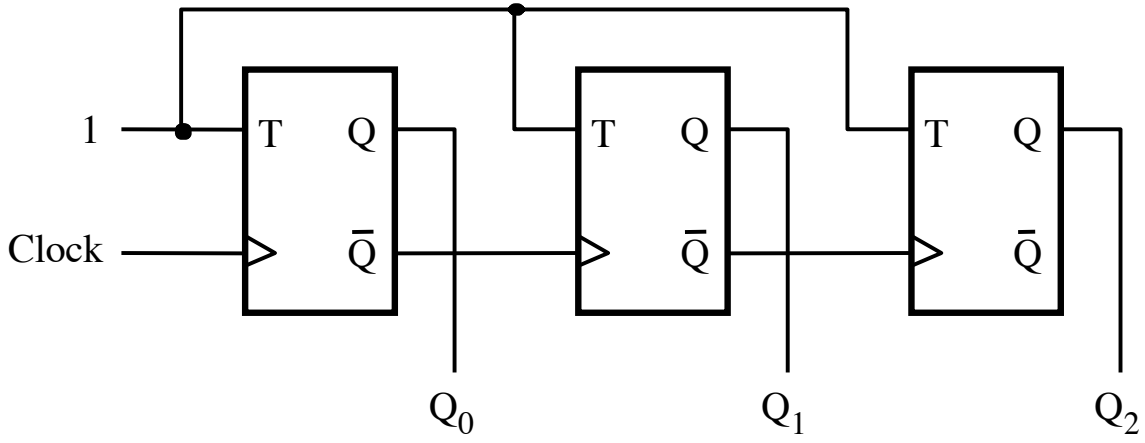
(a) Circuit



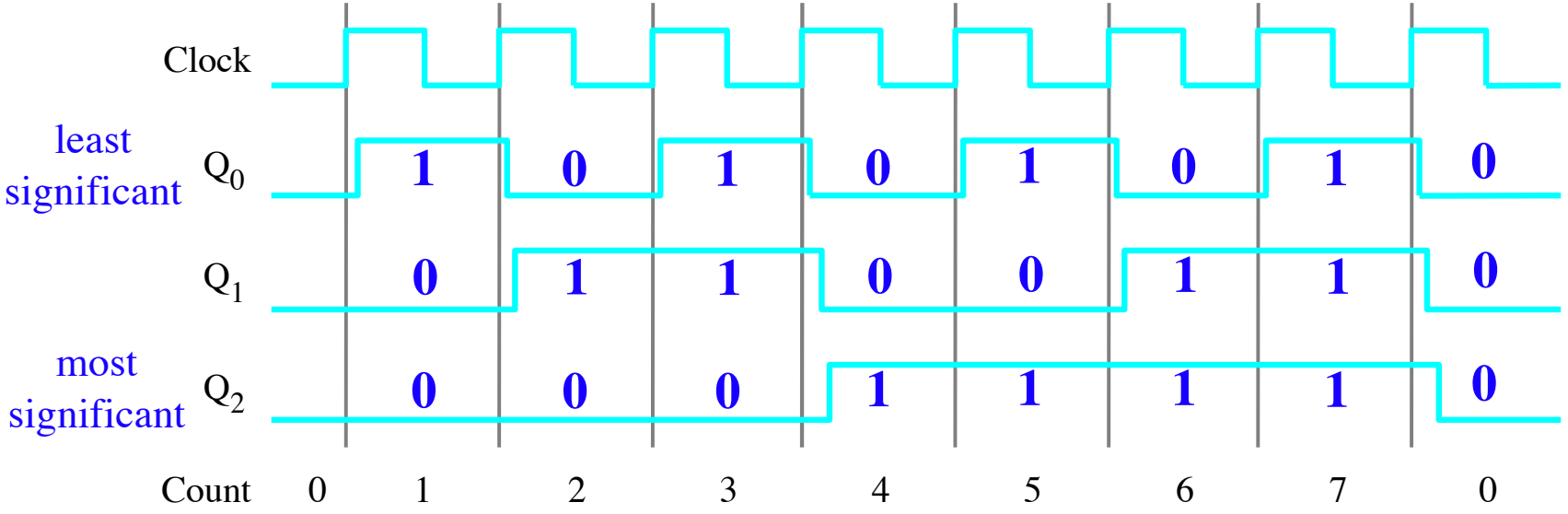
(b) Timing diagram

[Figure 5.19 from the textbook]

A three-bit up-counter



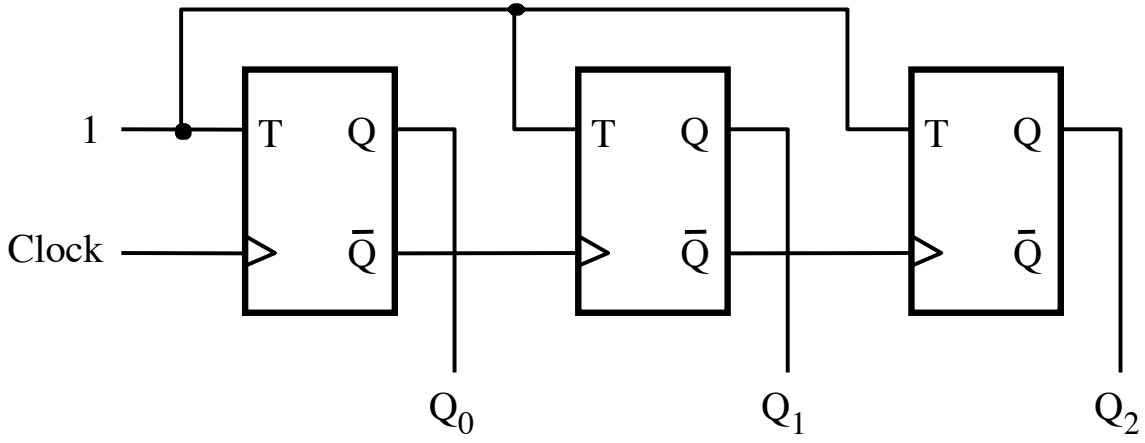
(a) Circuit



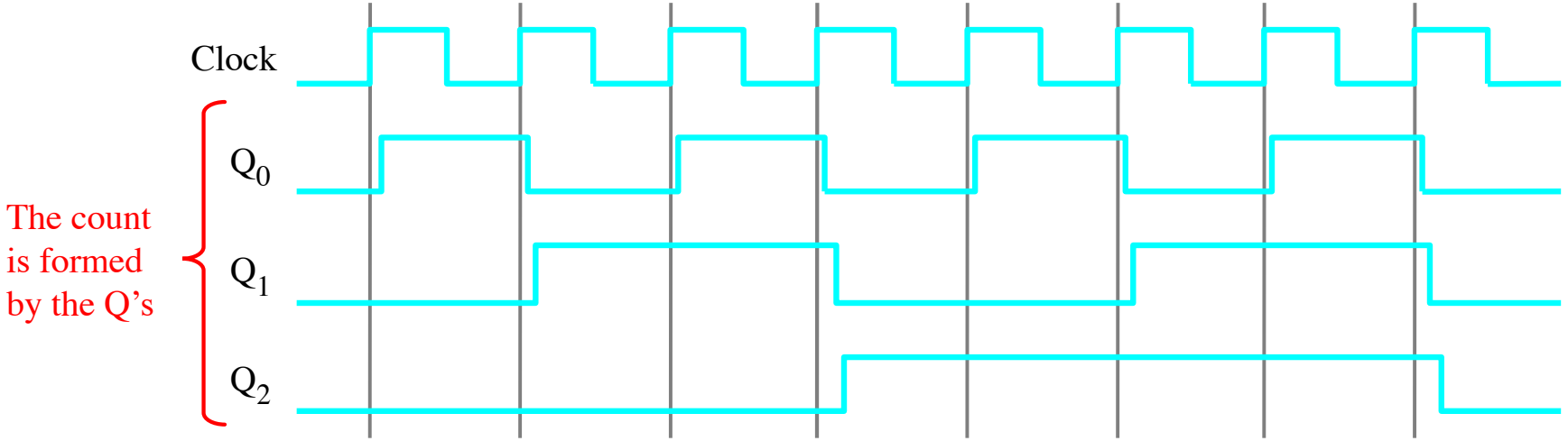
(b) Timing diagram

[Figure 5.19 from the textbook]

A three-bit up-counter



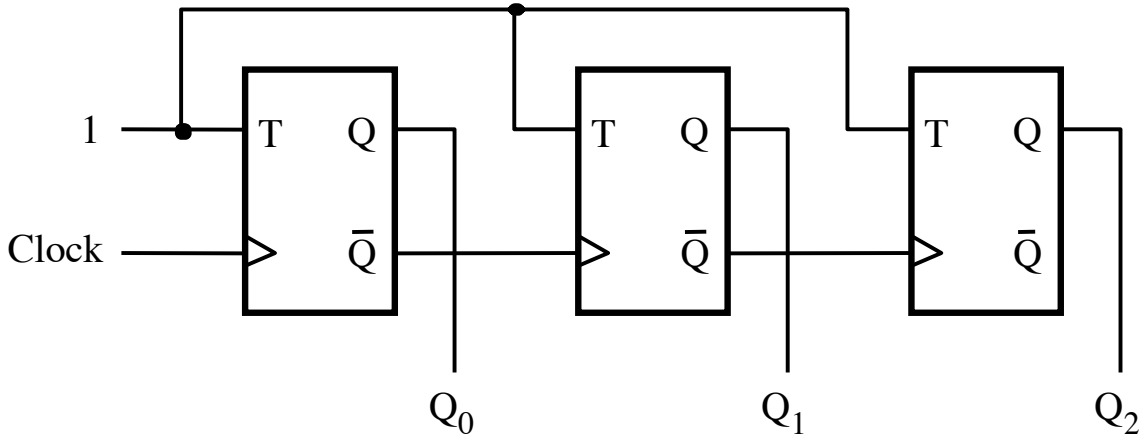
(a) Circuit



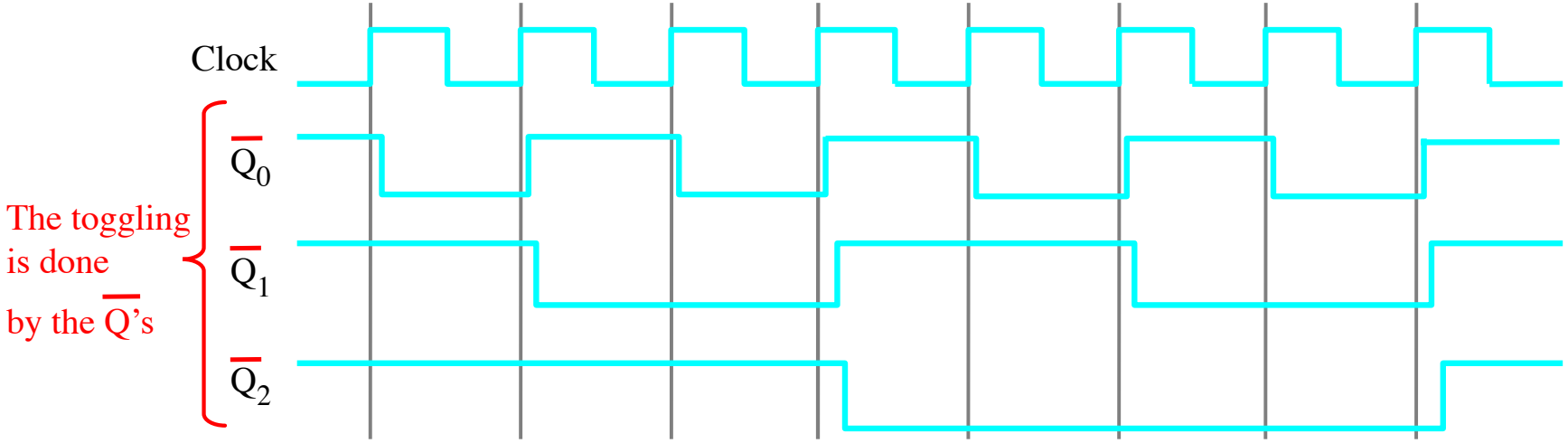
(b) Timing diagram

[Figure 5.19 from the textbook]

A three-bit up-counter

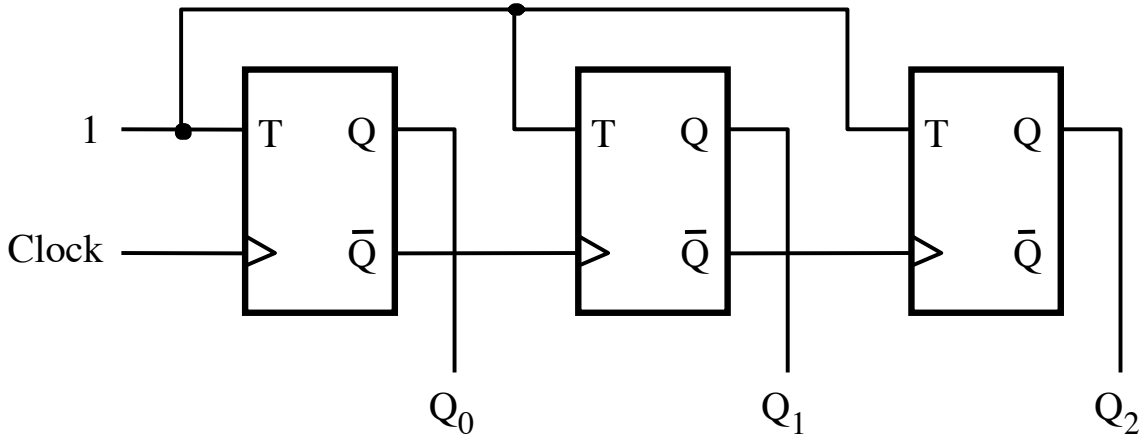


(a) Circuit

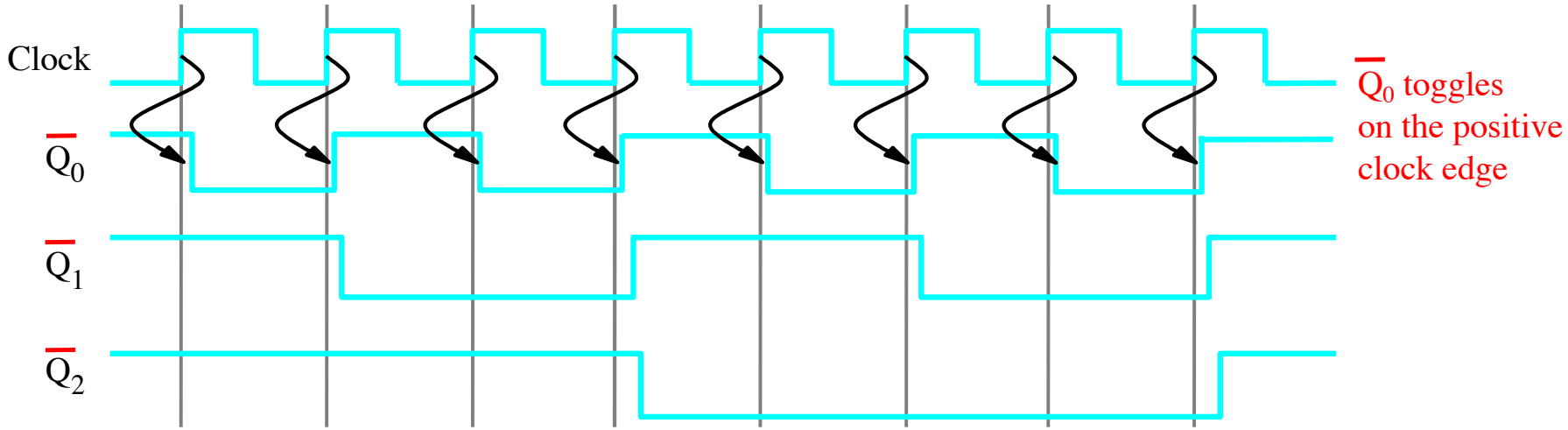


(b) Timing diagram

A three-bit up-counter

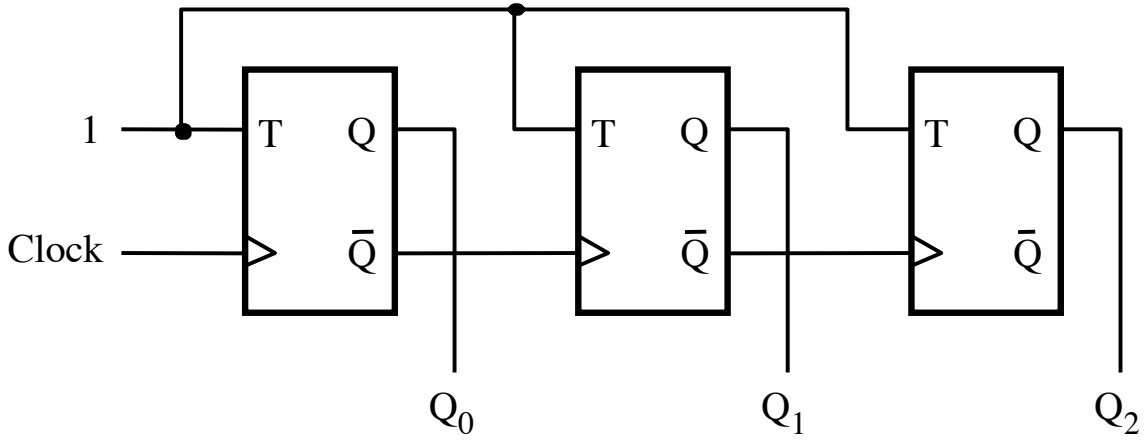


(a) Circuit

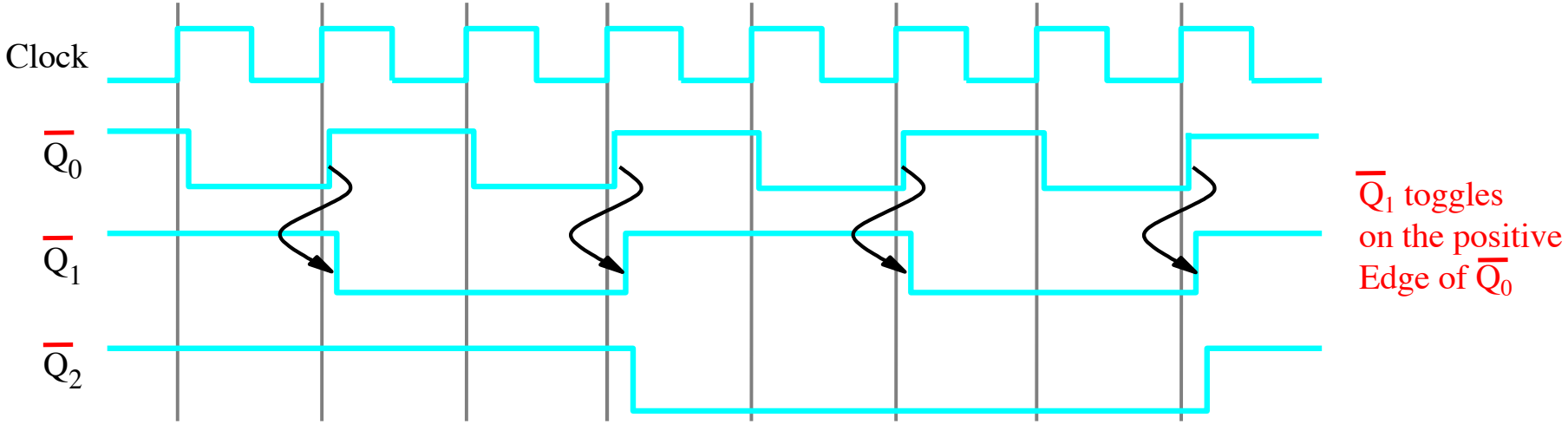


(b) Timing diagram

A three-bit up-counter

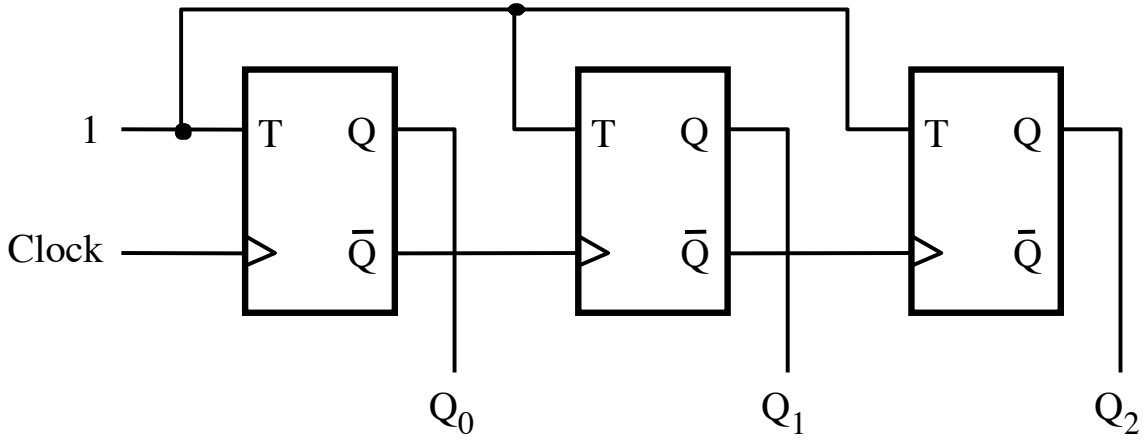


(a) Circuit

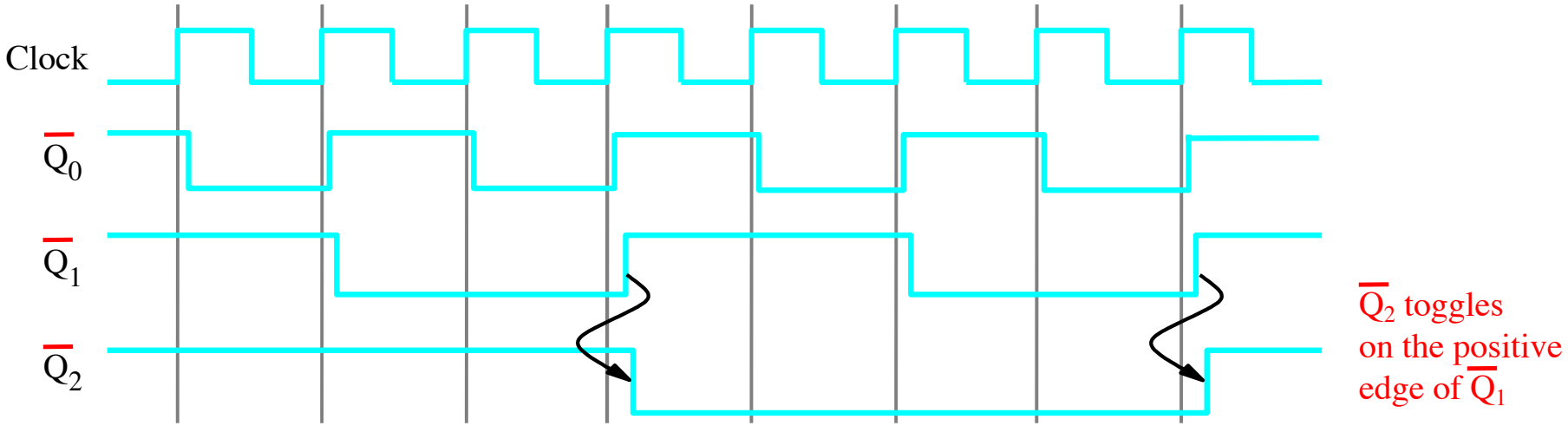


(b) Timing diagram

A three-bit up-counter

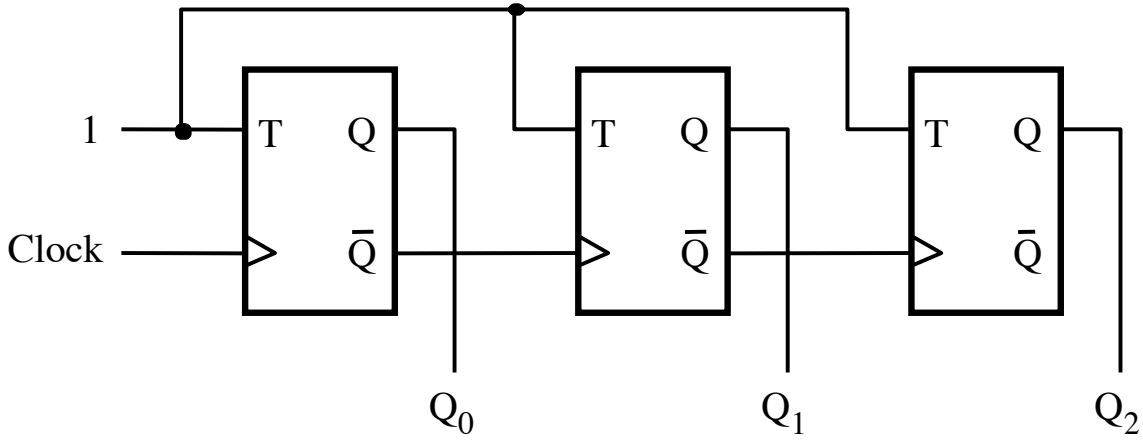


(a) Circuit



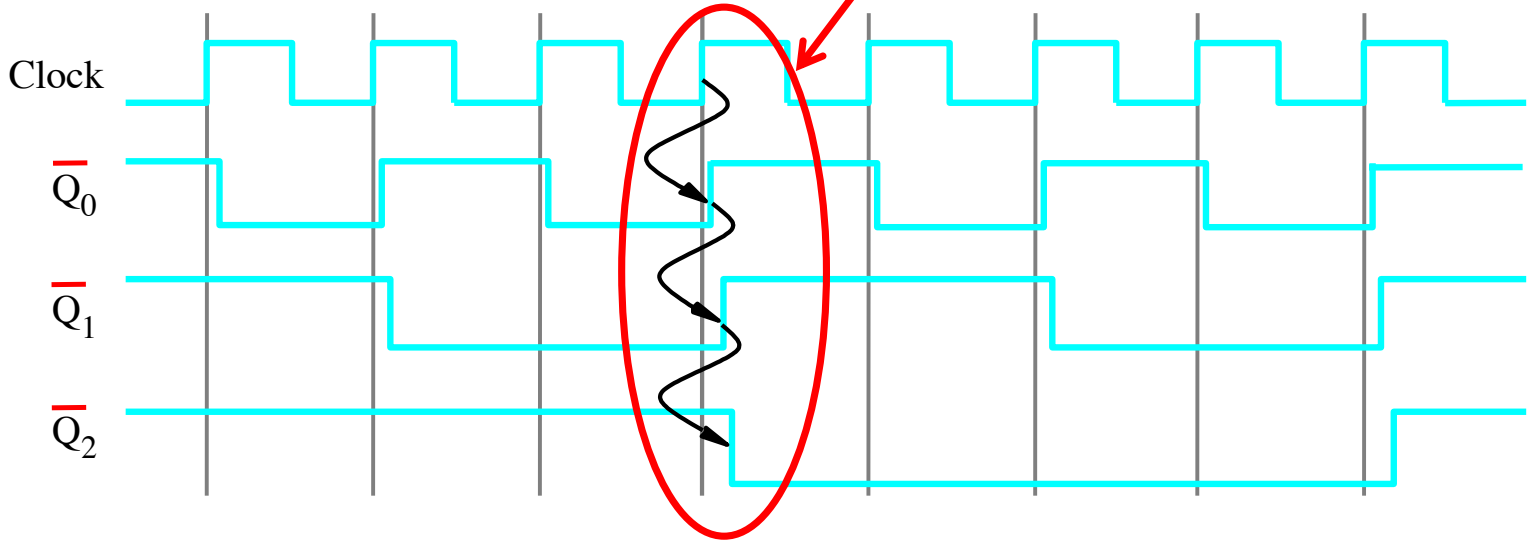
(b) Timing diagram

A three-bit up-counter



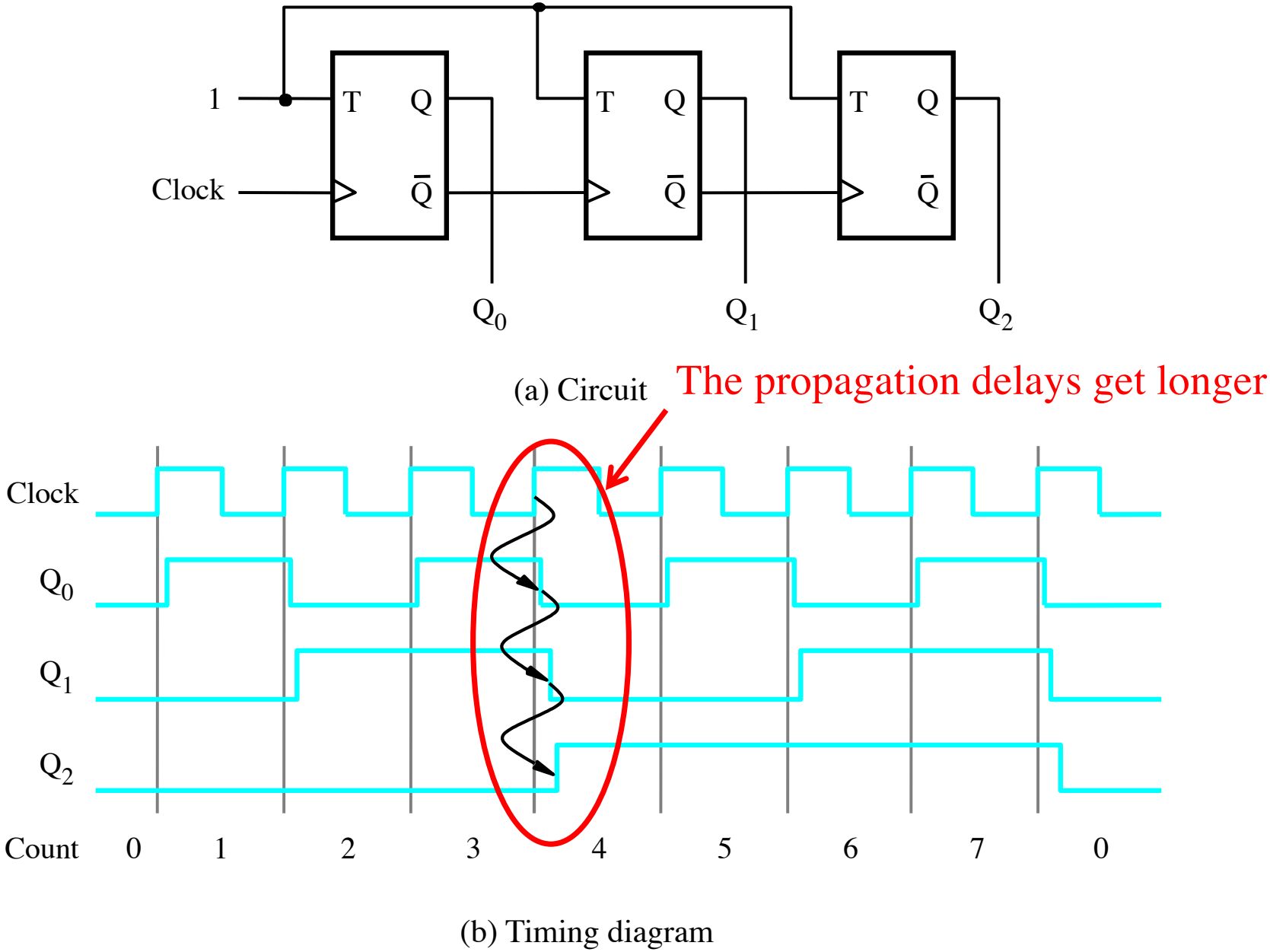
(a) Circuit

The propagation delays get longer



(b) Timing diagram

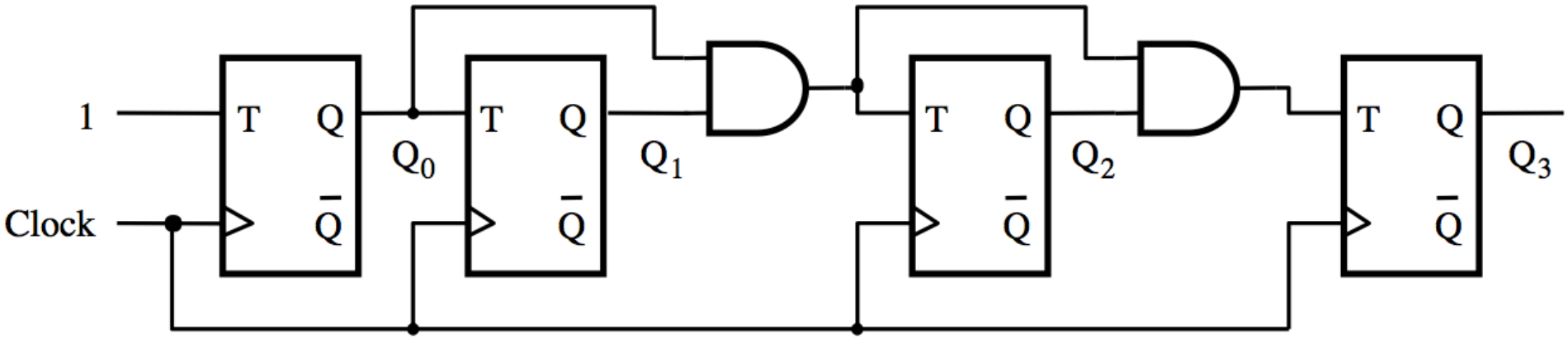
A three-bit up-counter



[Figure 5.19 from the textbook]

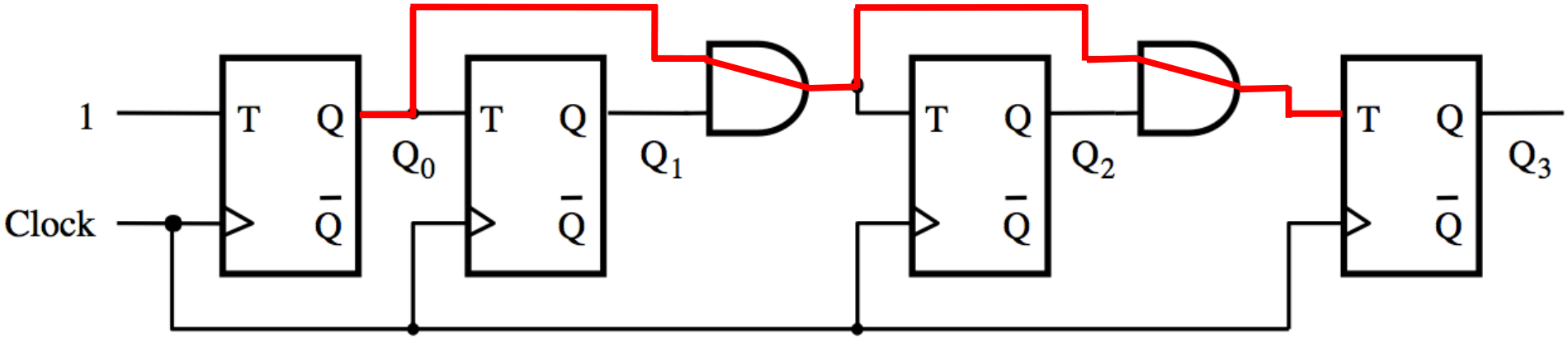
Synchronous Counters

A four-bit synchronous up-counter



[Figure 5.21 from the textbook]

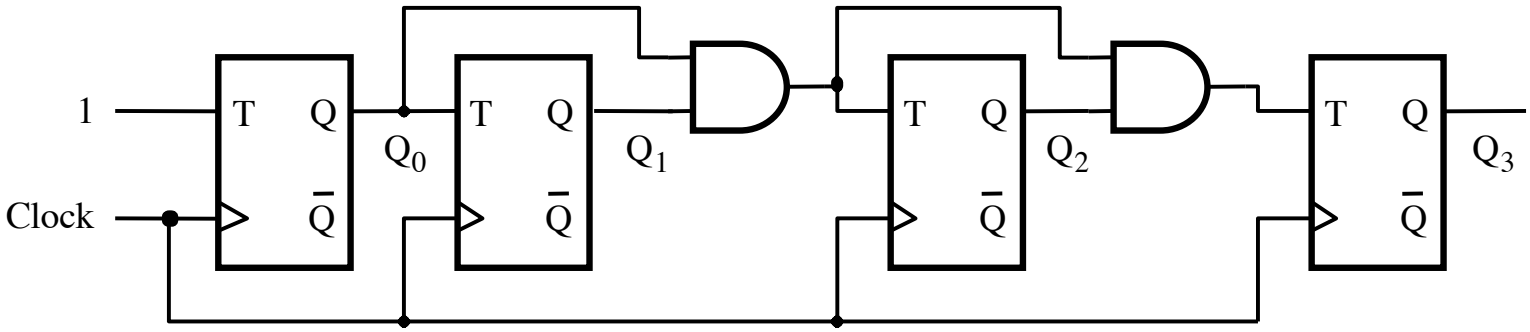
A four-bit synchronous up-counter



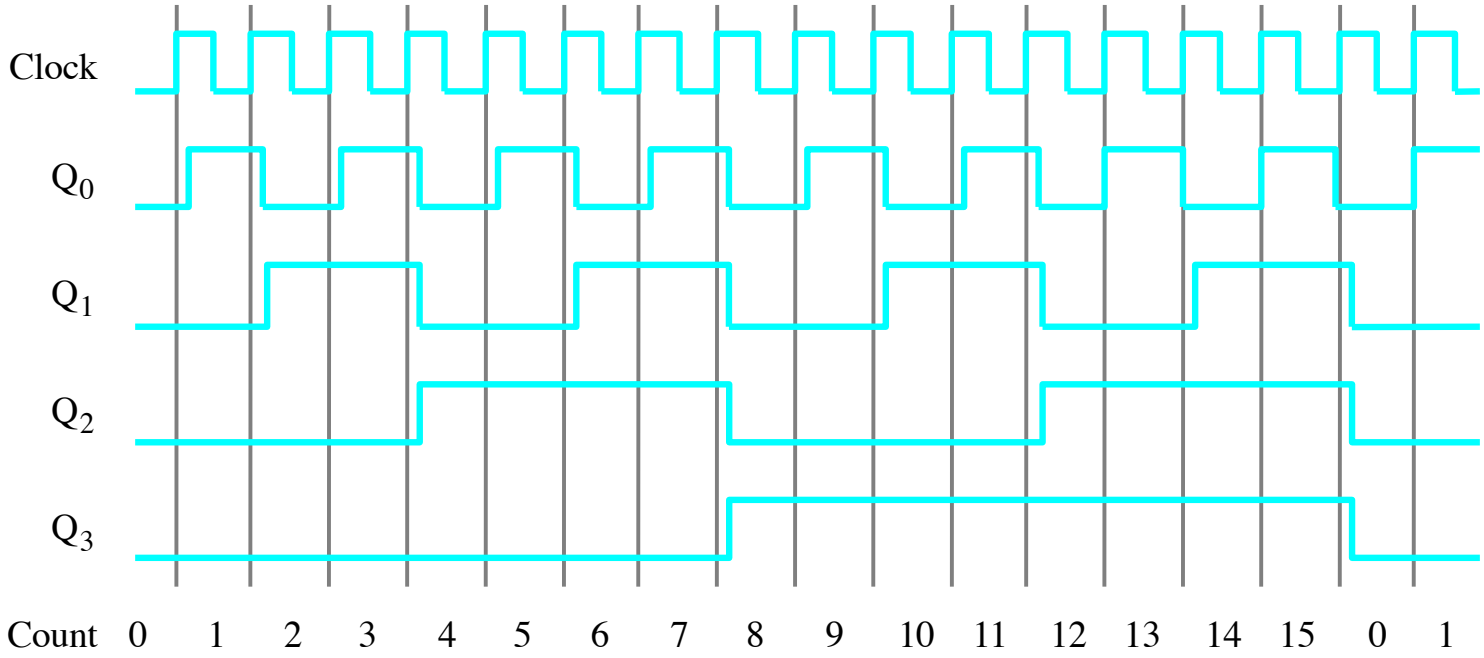
The propagation delay through all AND gates combined must not exceed the clock period minus the setup time for the flip-flops

[Figure 5.21 from the textbook]

A four-bit synchronous up-counter



(a) Circuit

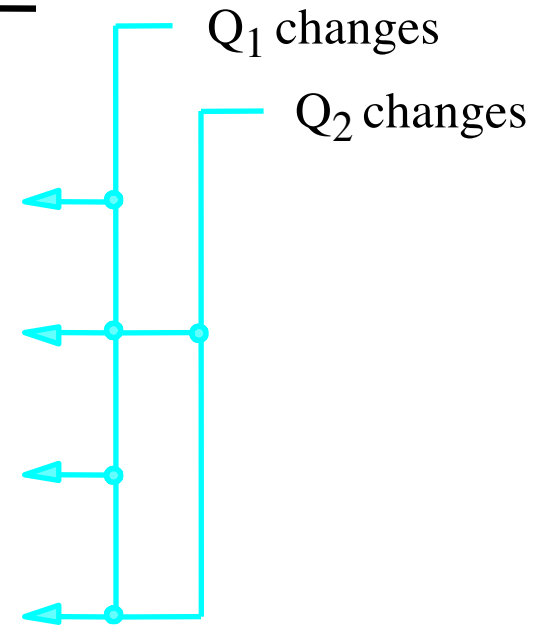


(b) Timing diagram

[Figure 5.21 from the textbook]

Derivation of the synchronous up-counter

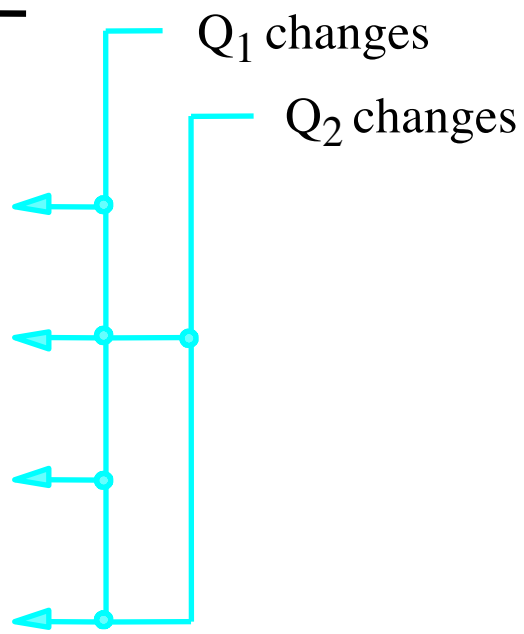
Clock cycle	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0



[Table 5.1 from the textbook]

Derivation of the synchronous up-counter

Clock cycle	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0



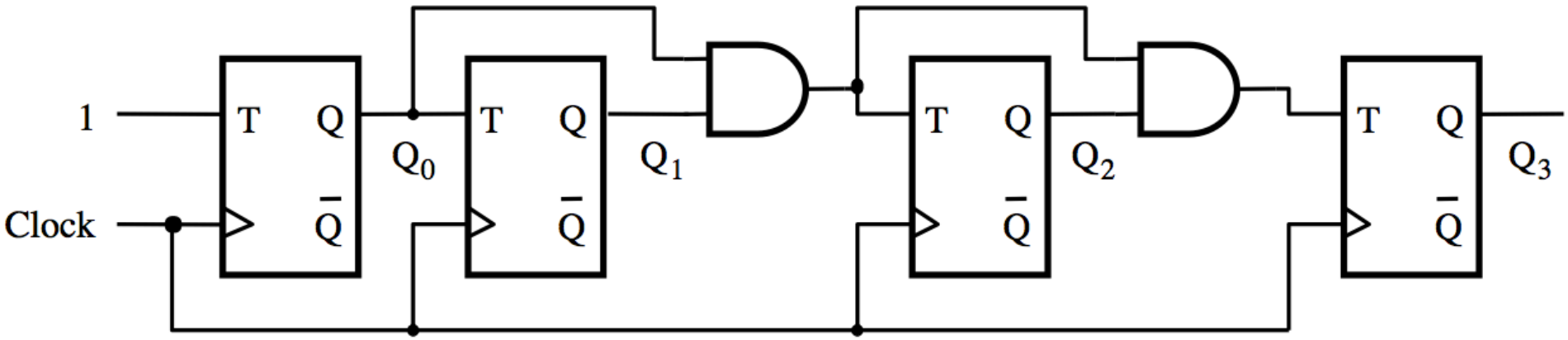
$T_0 = 1$

$T_1 = Q_0$

$T_2 = Q_0\ Q_1$

[Table 5.1 from the textbook]

A four-bit synchronous up-counter



$T_0 = 1$

$T_1 = Q_0$

$T_2 = Q_0 Q_1$

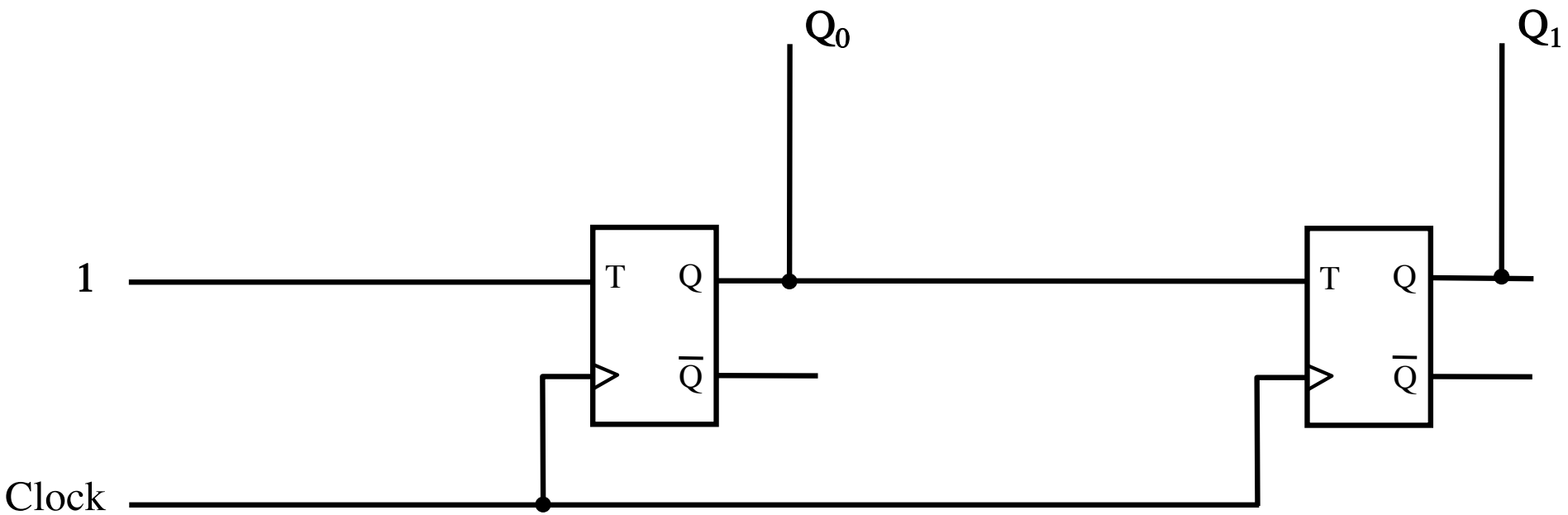
[Figure 5.21 from the textbook]

In general we have

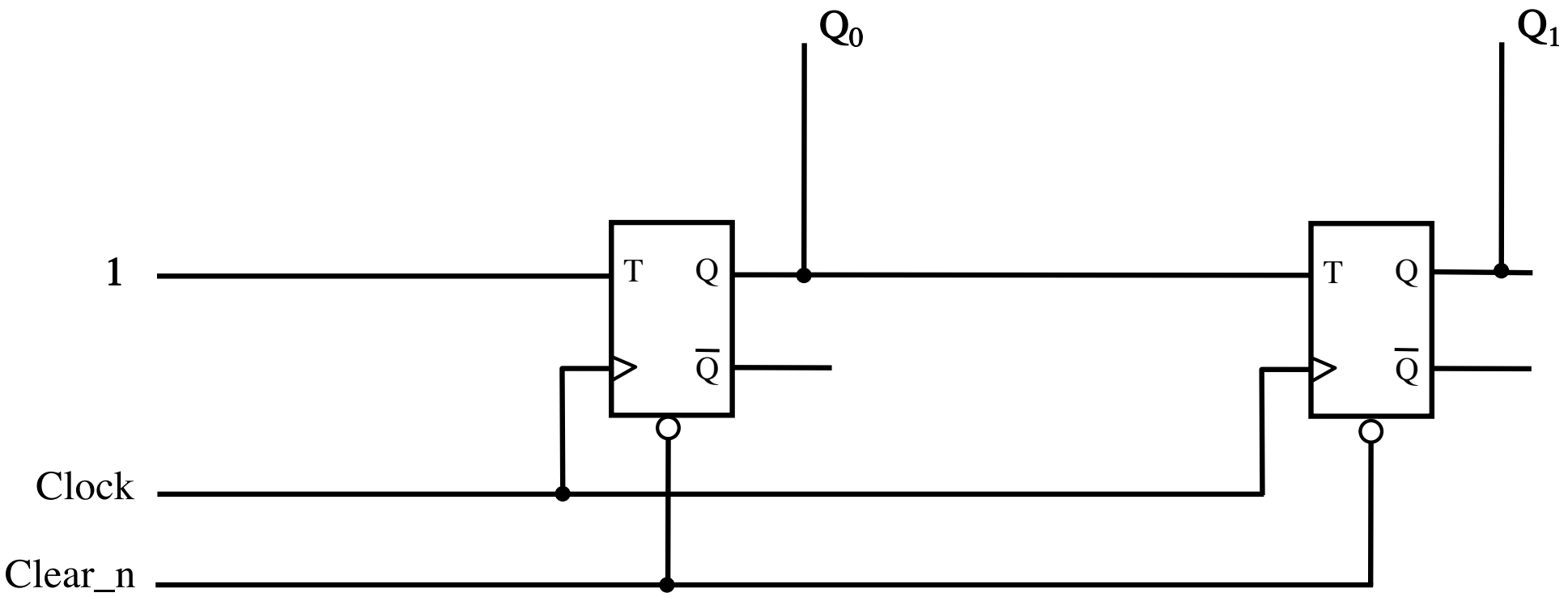
$$T_0 = 1$$
$$T_1 = Q_0$$
$$T_2 = Q_0\, Q_1$$
$$T_3 = Q_0\, Q_1\, Q_2$$
$$\cdots$$
$$T_n = Q_0\, Q_1\, Q_2 \cdots Q_{n-1}$$

Synchronous v.s. Asynchronous Clear

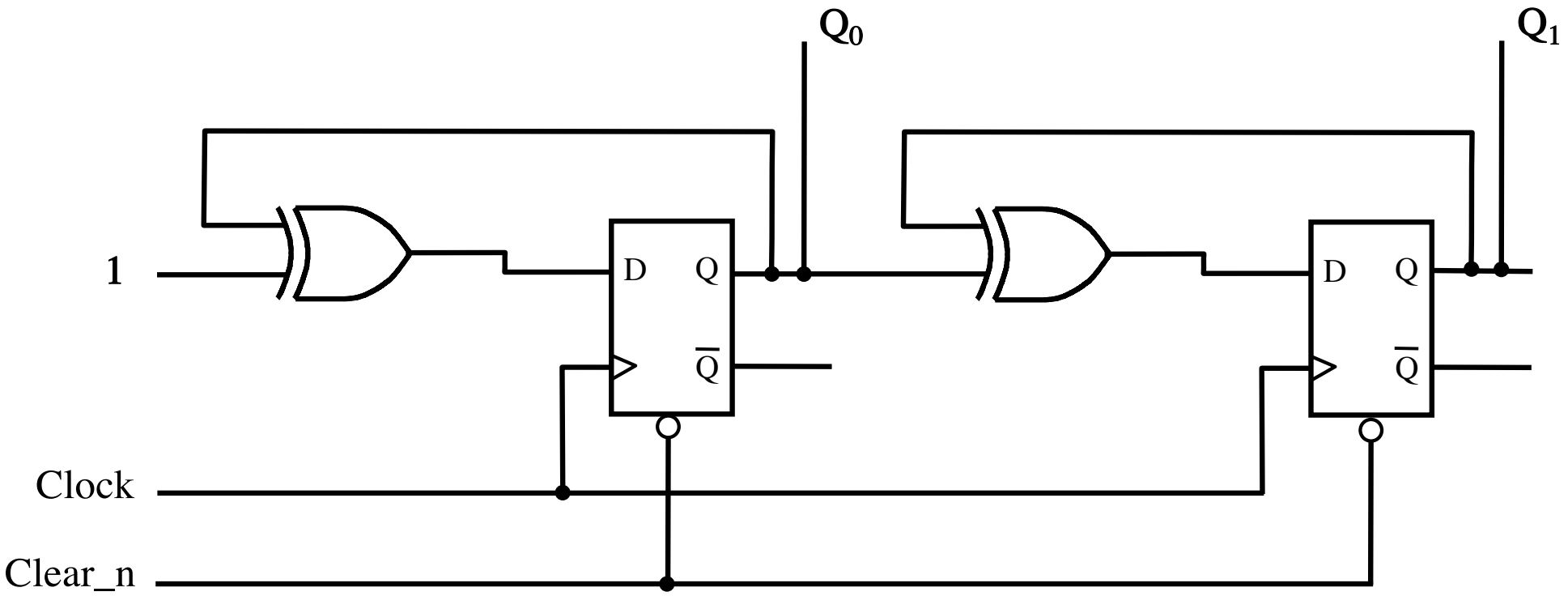
2-Bit Synchronous Up-Counter (without clear capability)



2-Bit Synchronous Up-Counter (with asynchronous clear)

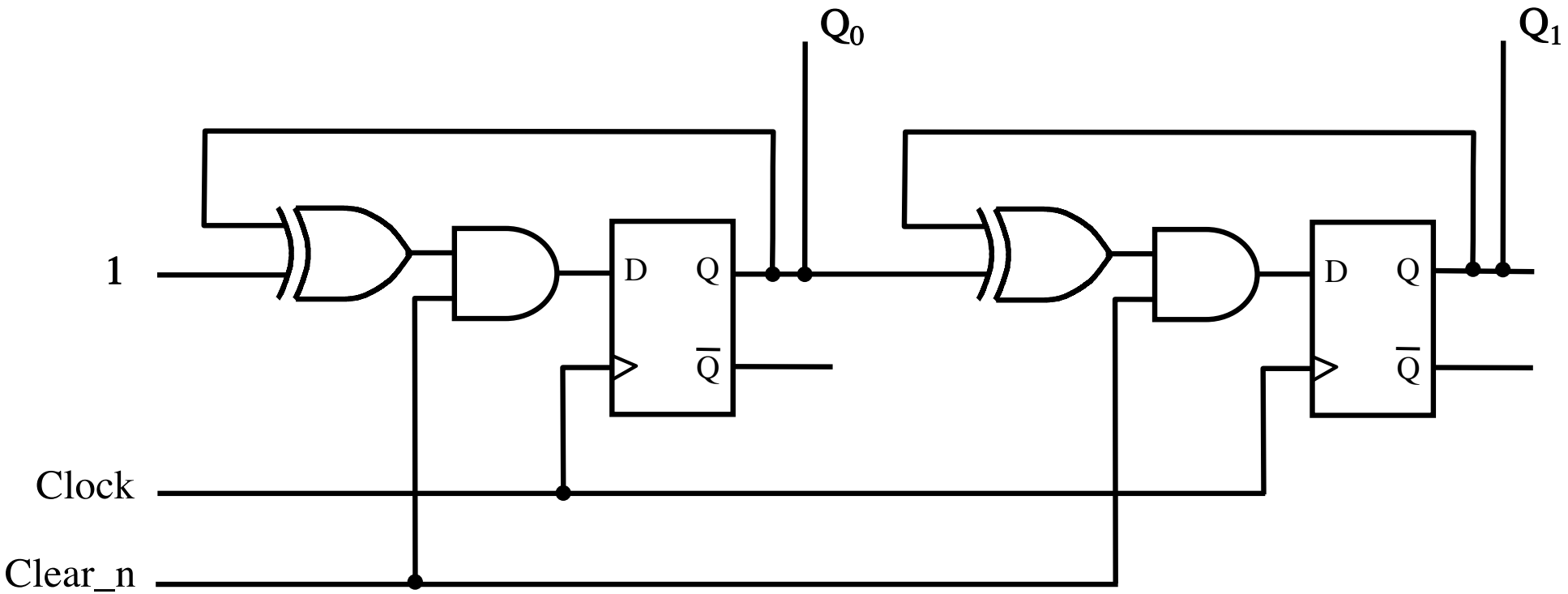


2-Bit Synchronous Up-Counter (with asynchronous clear)



This is the same circuit but uses D Flip-Flops.

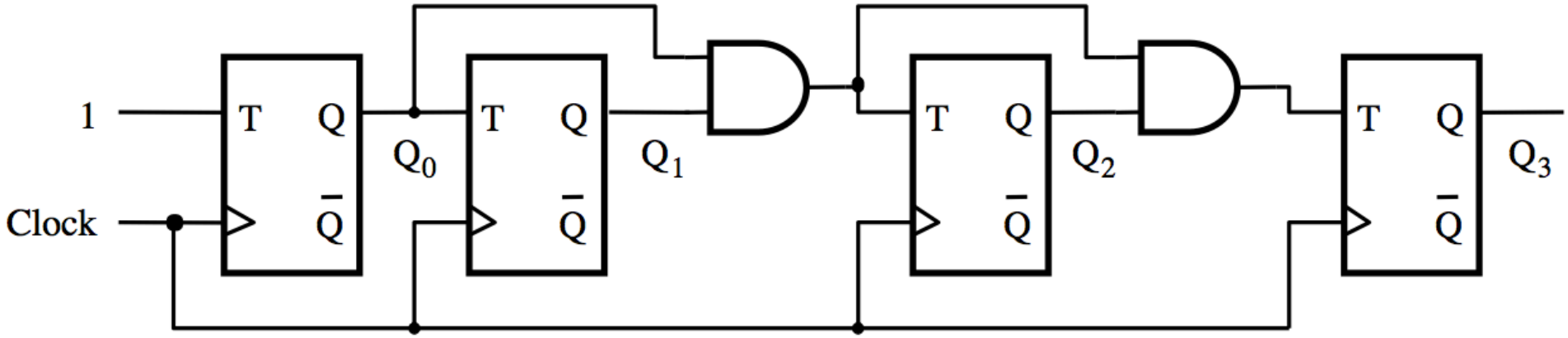
2-Bit Synchronous Up-Counter (with synchronous clear)



This counter can be cleared only on the positive clock edge.

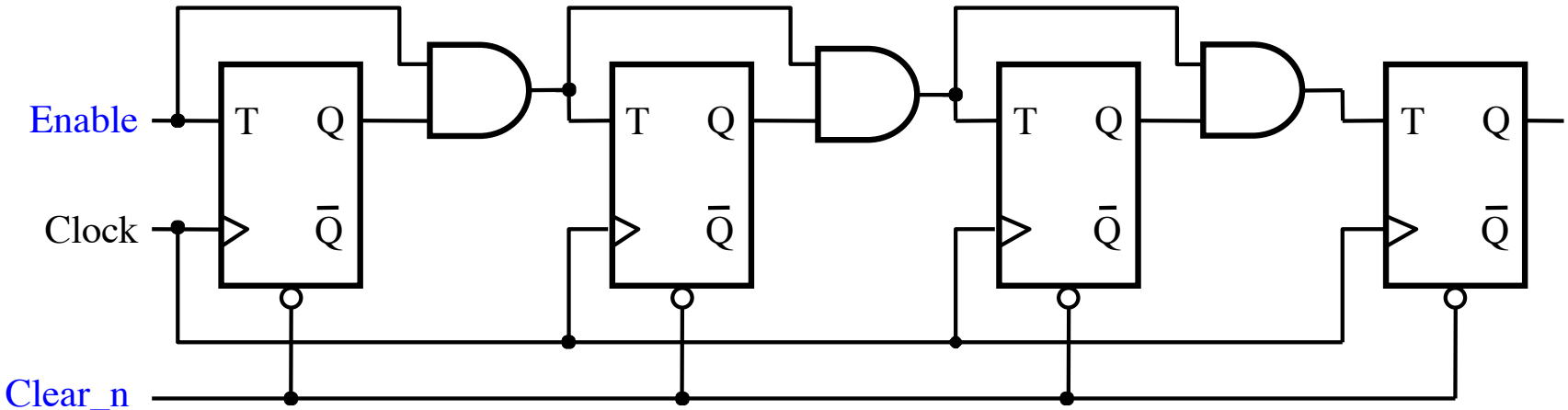
Adding Enable Capability

A four-bit synchronous up-counter



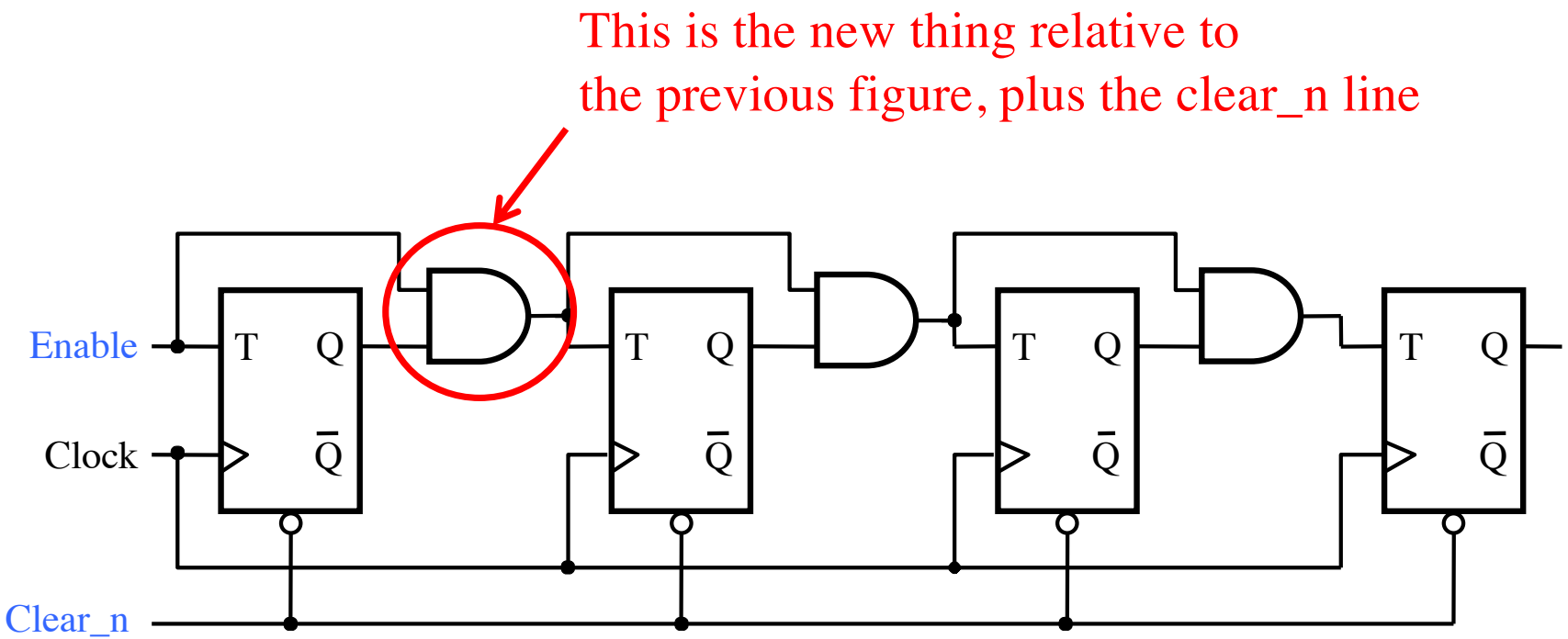
[Figure 5.21 from the textbook]

Inclusion of Enable and Clear Capability



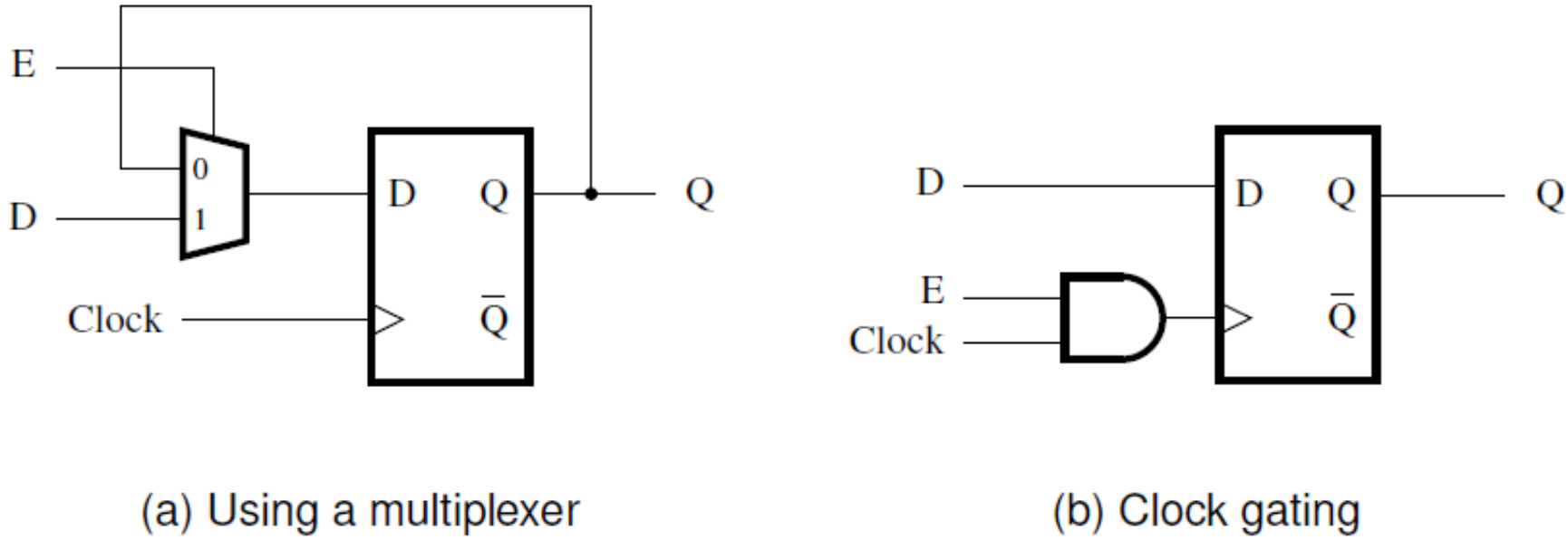
[Figure 5.22 from the textbook]

Inclusion of Enable and Clear Capability



[Figure 5.22 from the textbook]

Providing an enable input for a D flip-flop



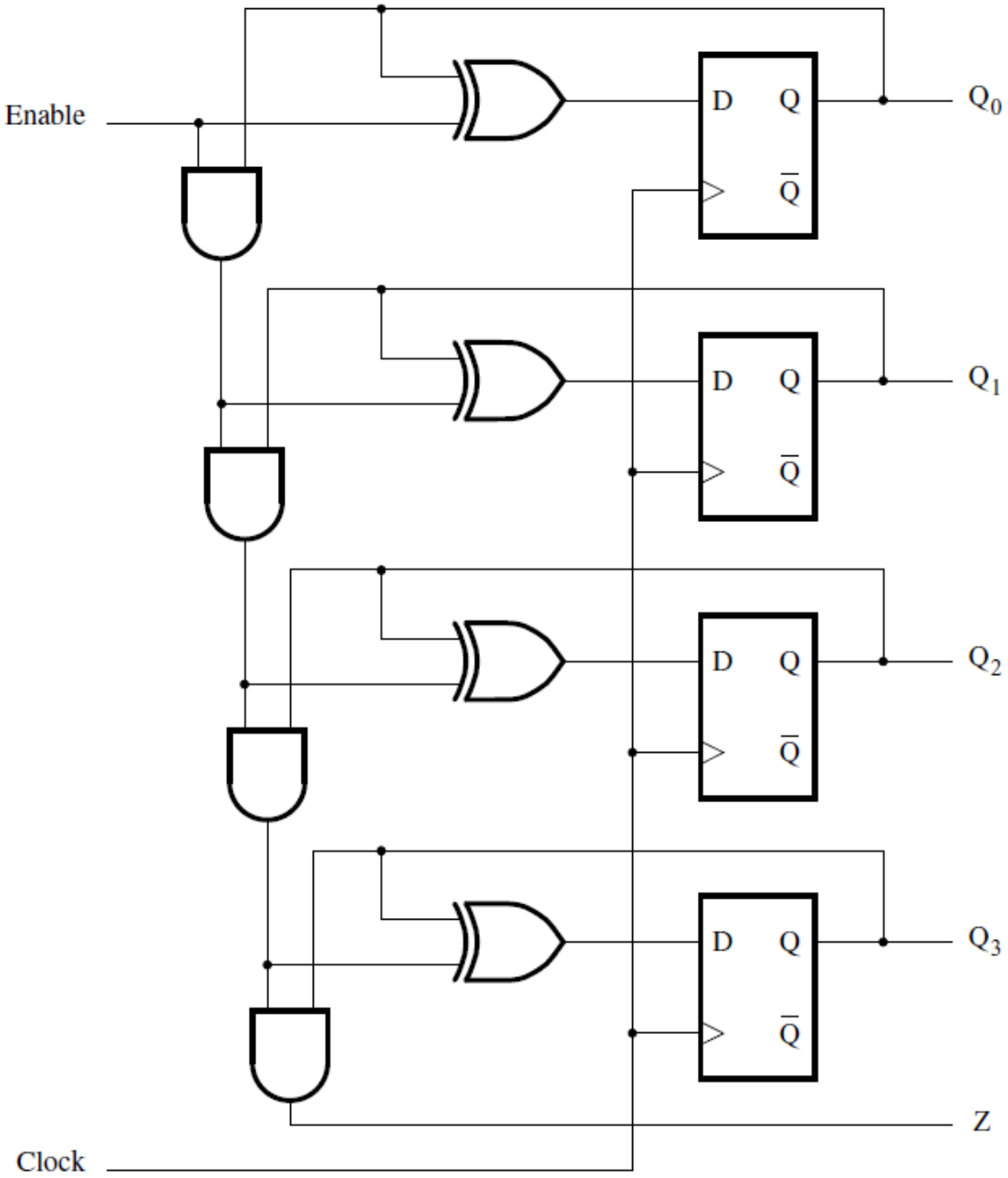
(a) Using a multiplexer

(b) Clock gating

[Figure 5.56 from the textbook]

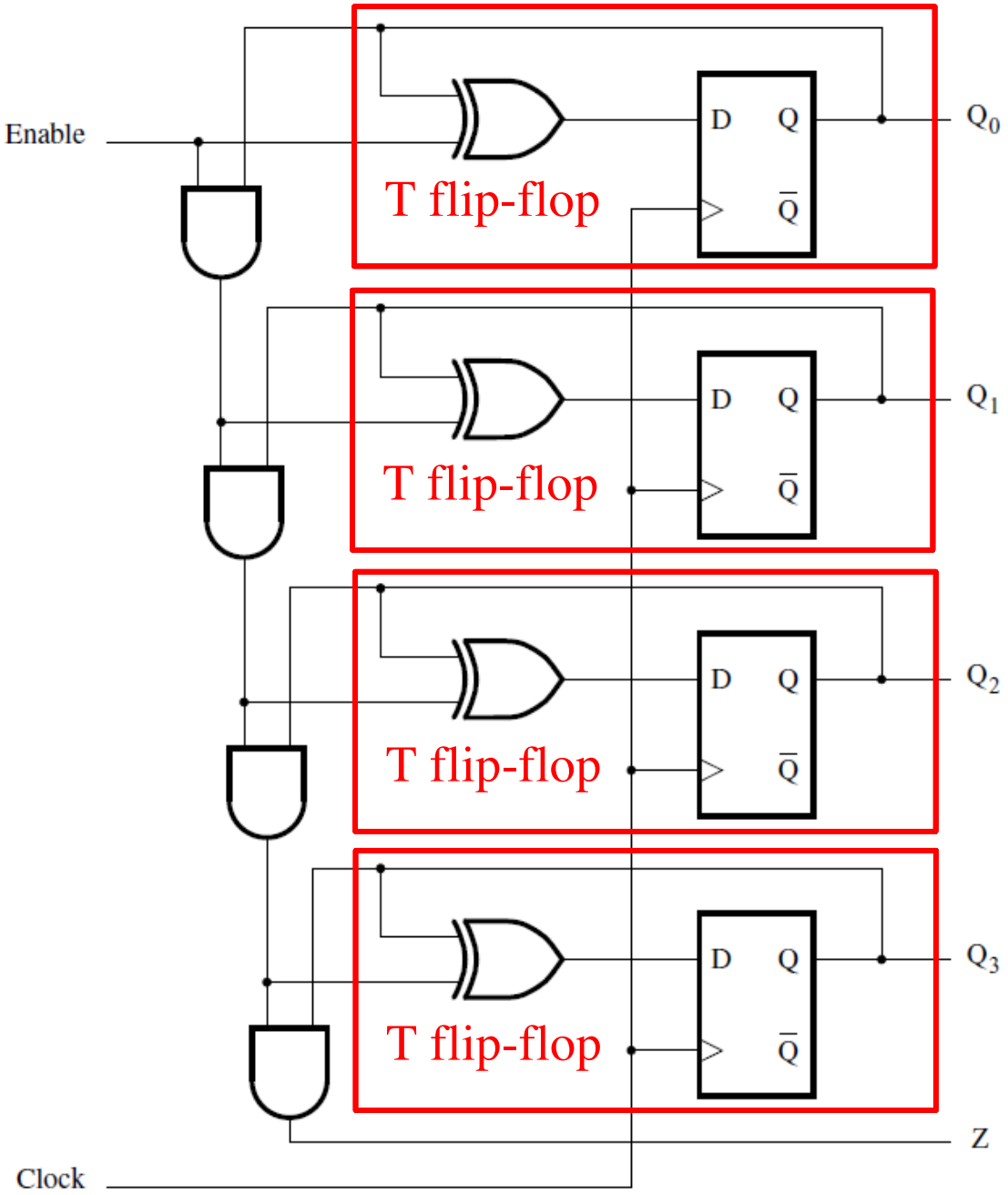
Synchronous Counter
(with D Flip-Flops)

A 4-bit up-counter with D flip-flops



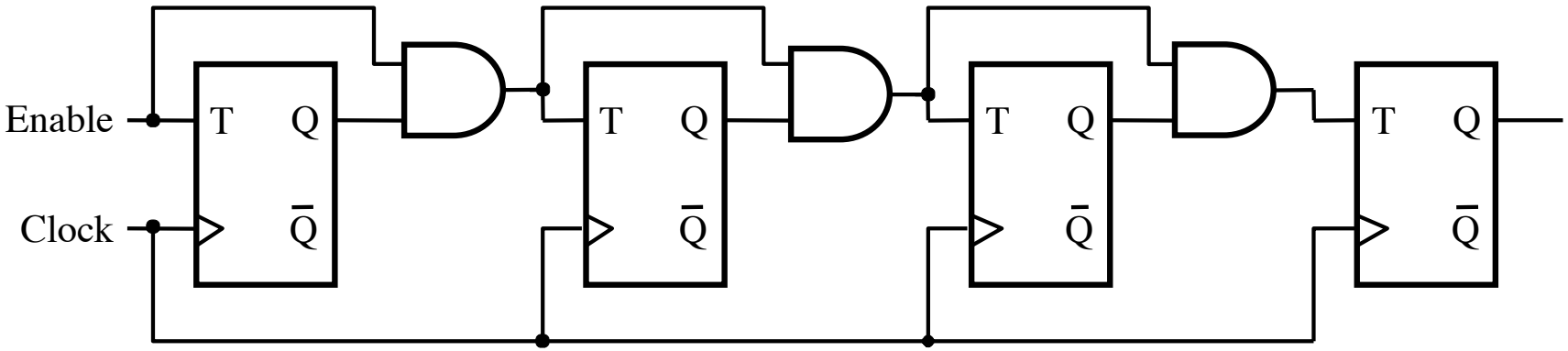
[Figure 5.23 from the textbook]

A 4-bit up-counter with D flip-flops

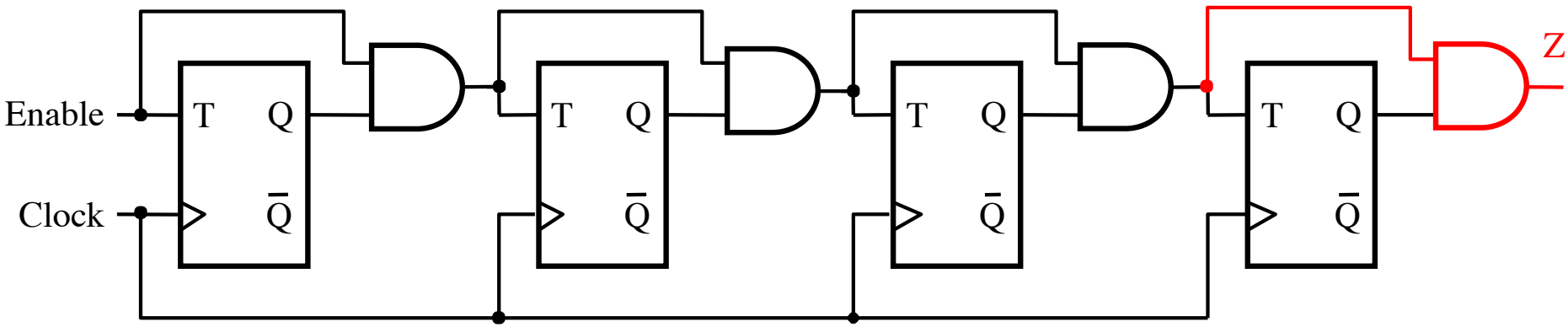


[Figure 5.23 from the textbook]

Equivalent to this circuit with T flip-flops



Equivalent to this circuit with T flip-flops

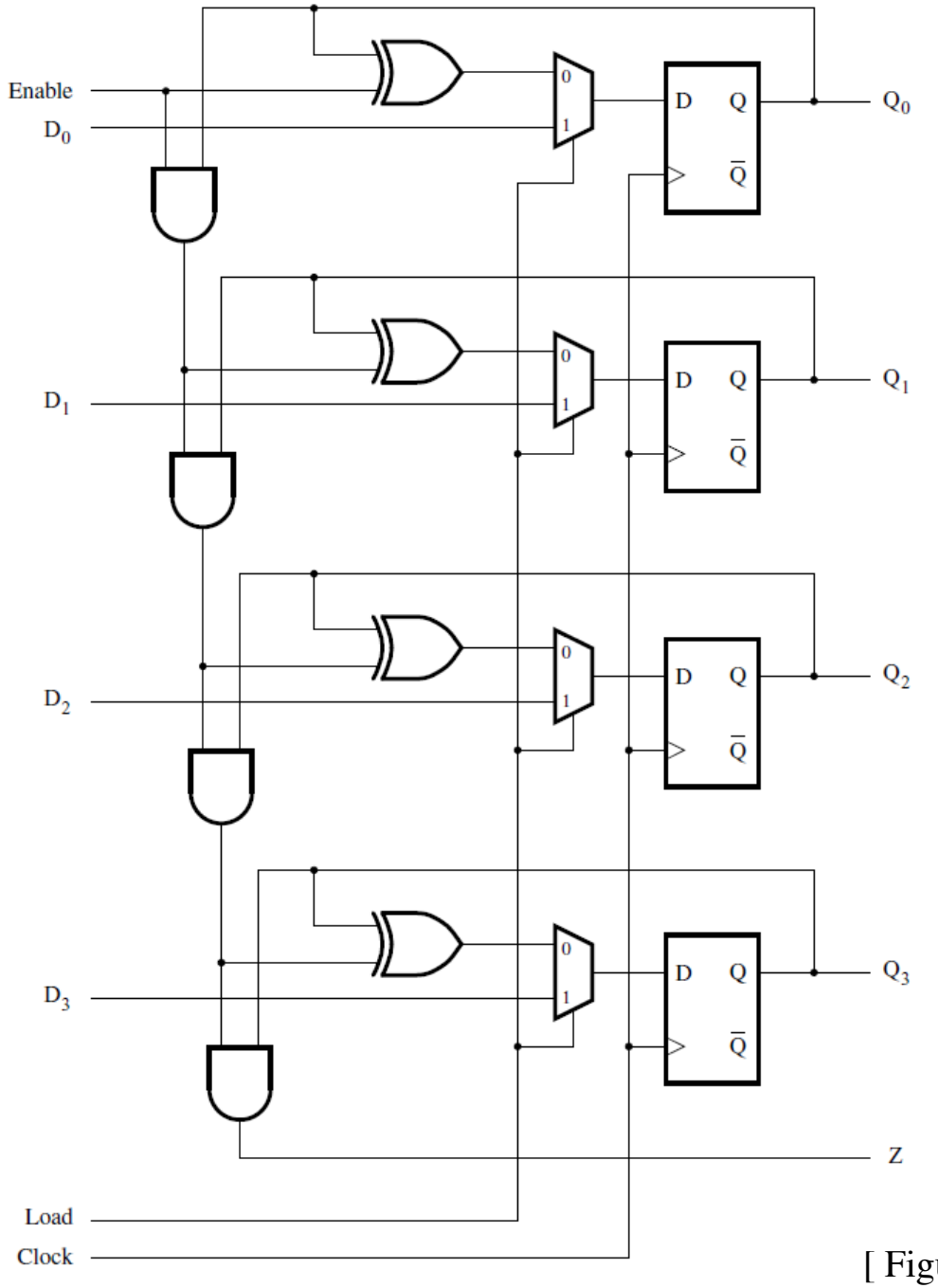


But has one extra output called Z, which can be used to connect two 4-bit counters to make an 8-bit counter.

When Z=1 the counter will go to 0000 on the next clock edge, i.e., the outputs of all flip-flops are currently 1 (maximum count value).

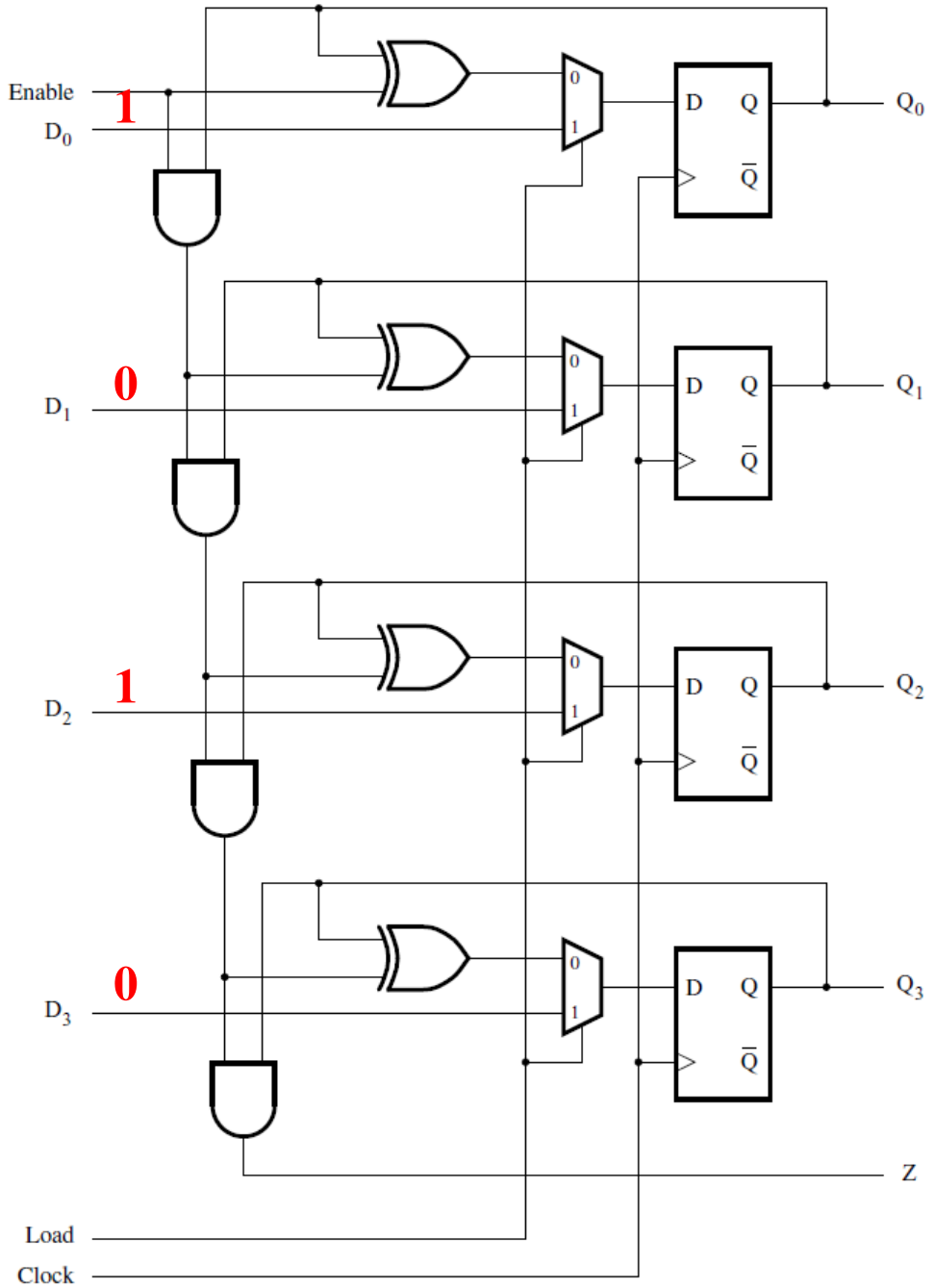
Counters with Parallel Load

A counter with parallel-load capability



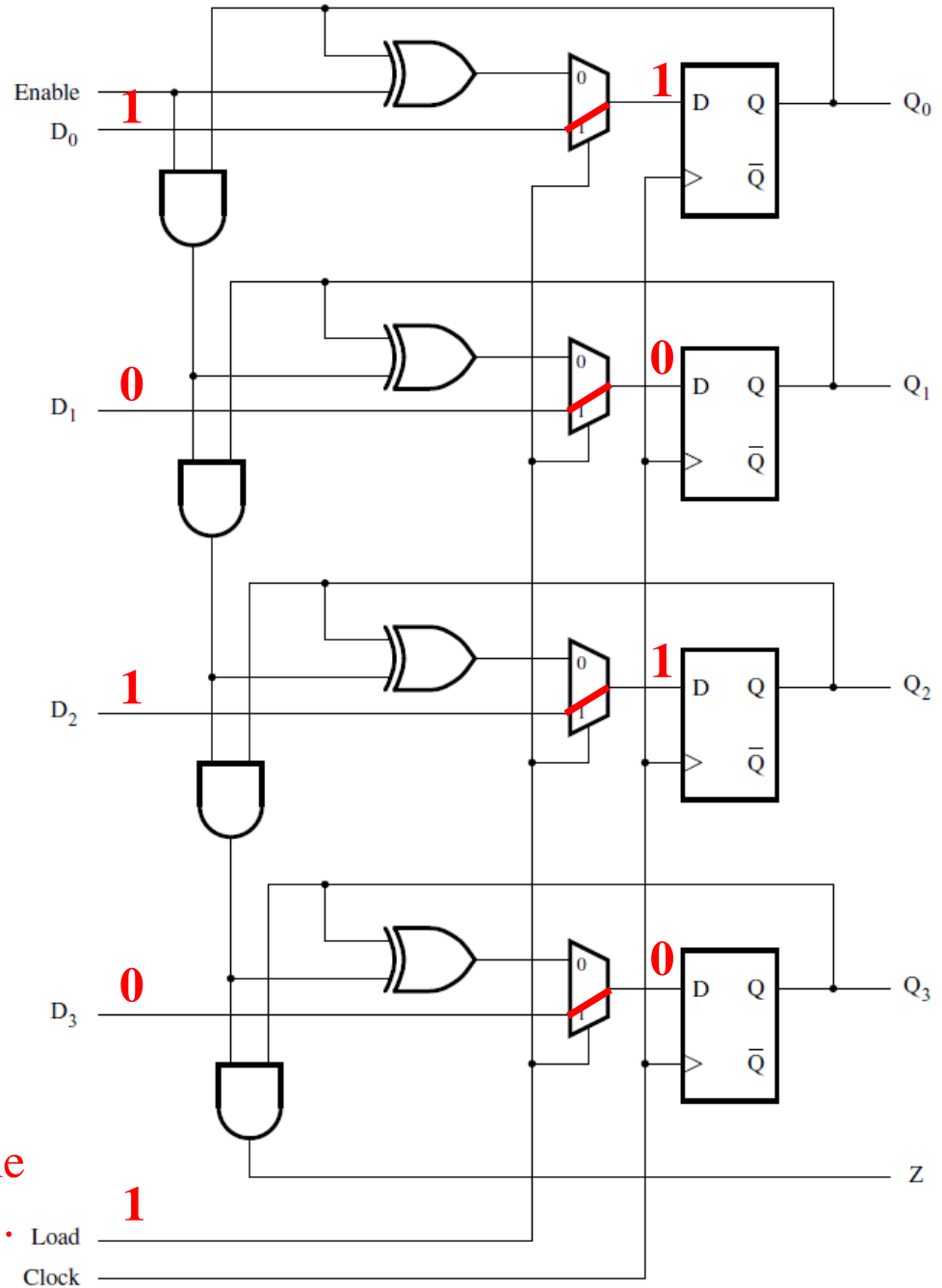
[Figure 5.24 from the textbook]

How to load the initial count value



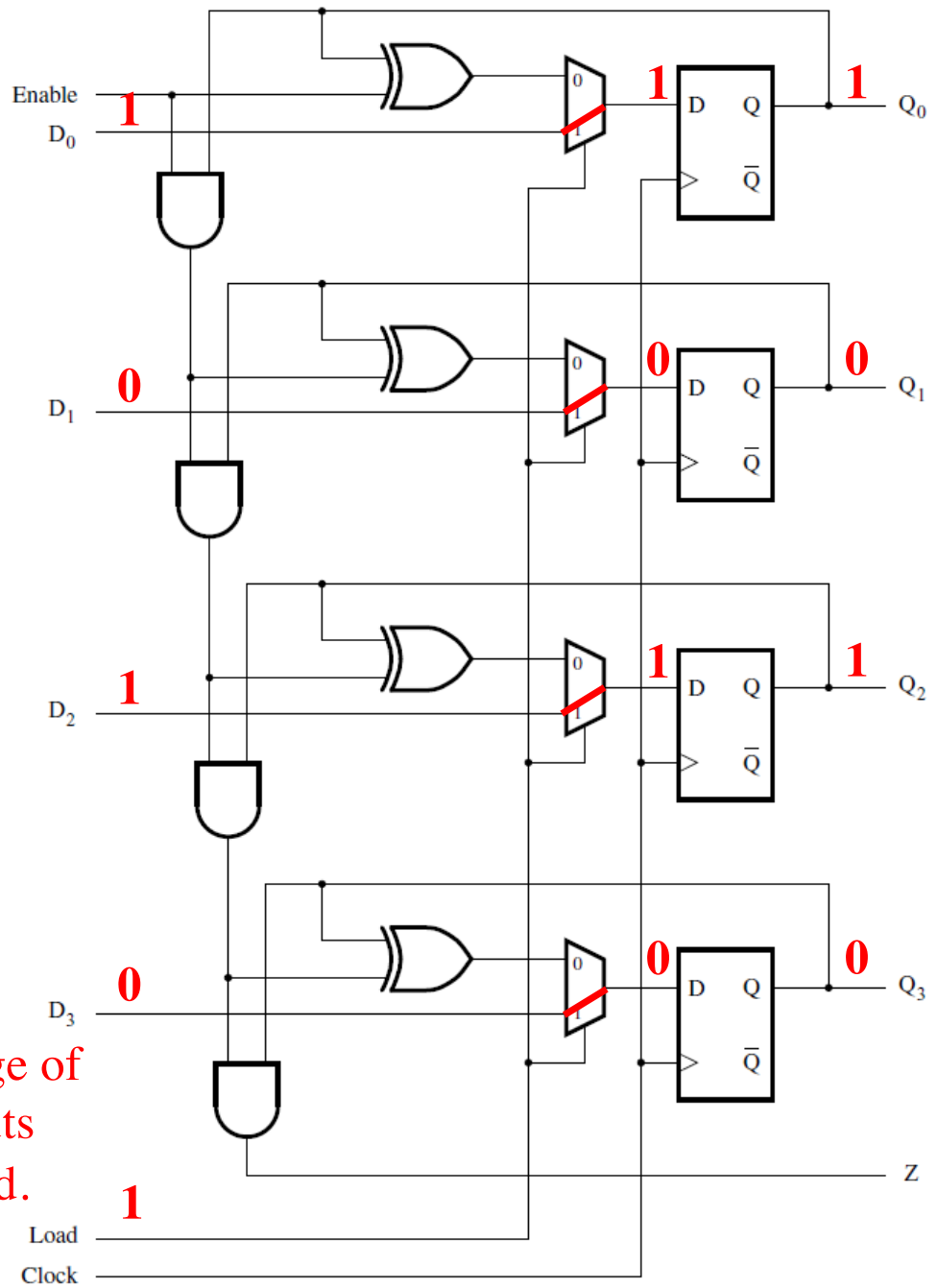
Set the initial count on the parallel load lines (in this case 5).

How to zero a counter



Set "Load" to 1, to open the "1" line of the multiplexers.

How to zero a counter



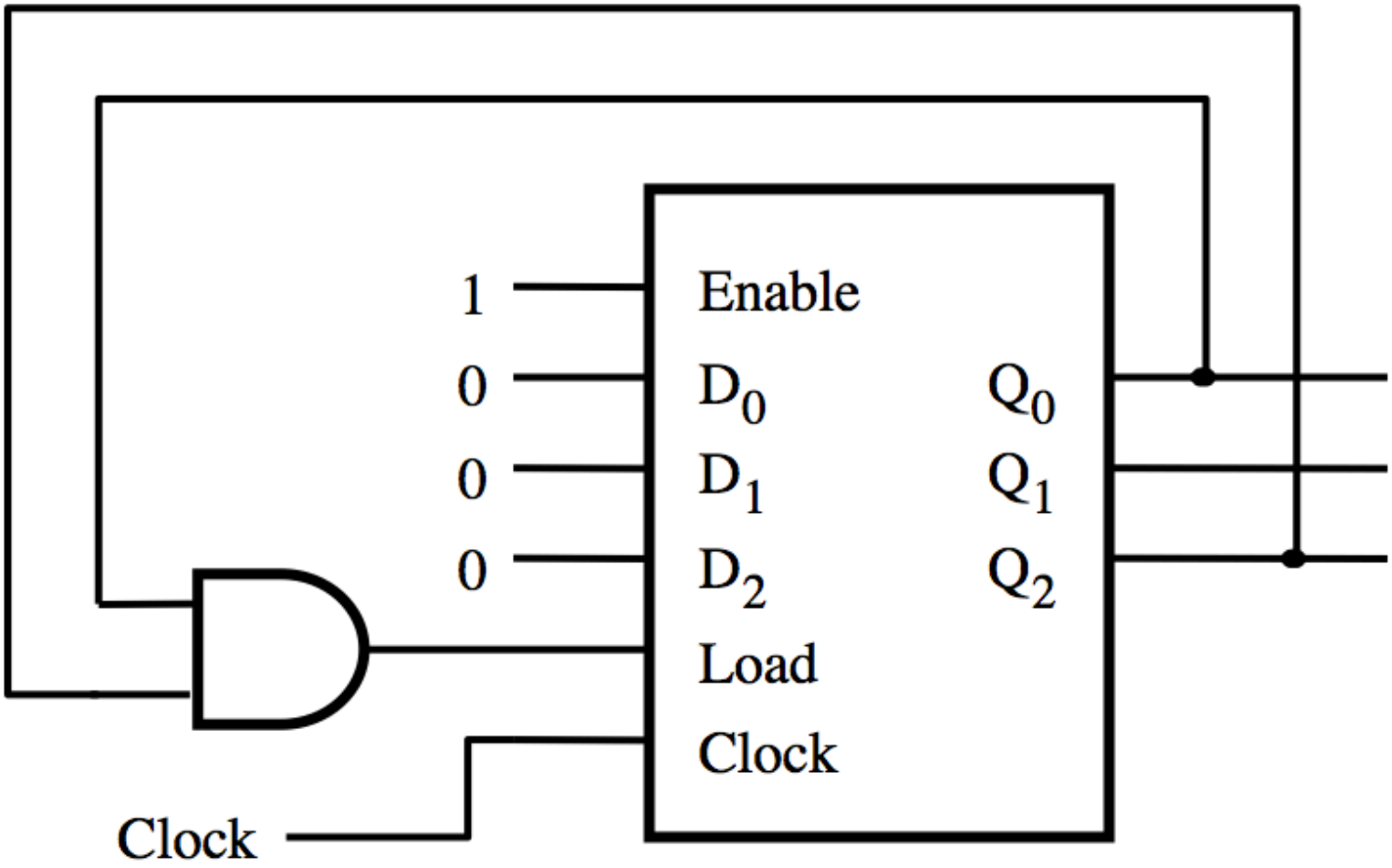
When the next positive edge of the clock arrives, the outputs of the flip-flops are updated.

Reset Synchronization

Motivation

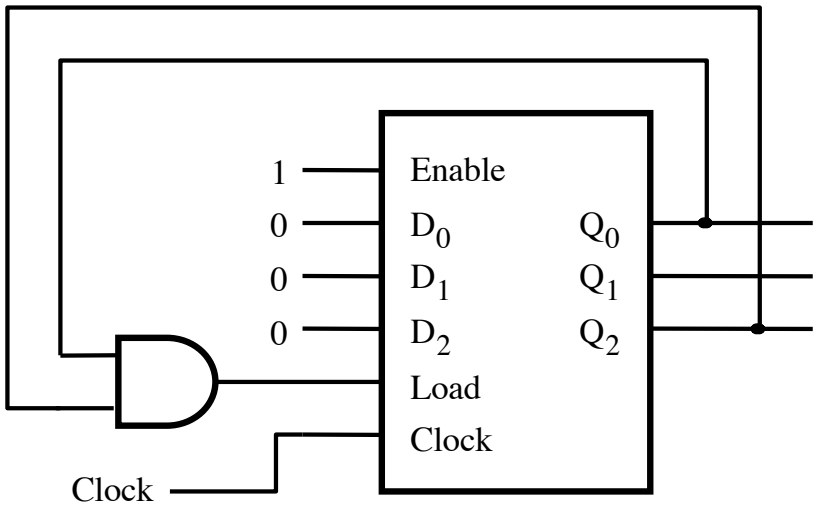
- An n-bit counter counts from 0, 1, …, 2^n-1
- For example a 3-bit counter counts up as follow
  - 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, …
- What if we want it to count like this
  - 0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5, 0, 1, …
- In other words, what is the cycle is not a power of 2?

What does this circuit do?

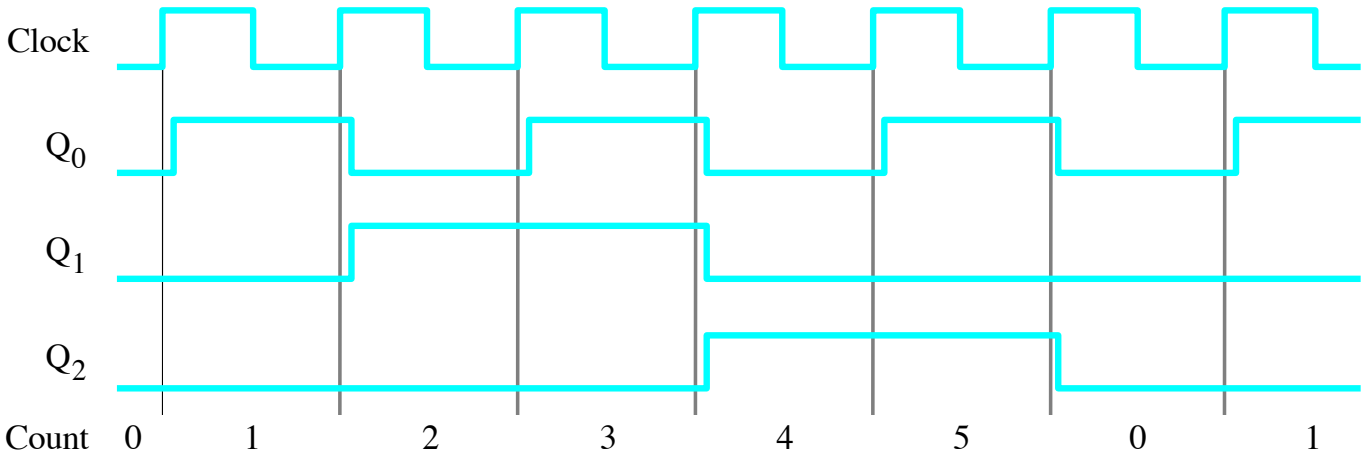


[Figure 5.25a from the textbook]

A modulo-6 counter with synchronous reset



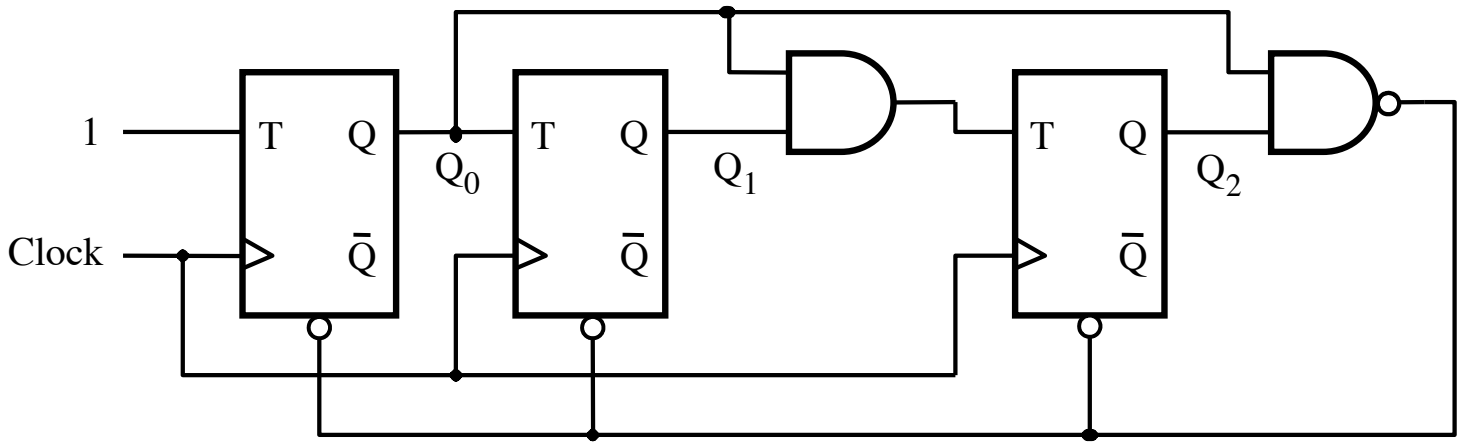
(a) Circuit



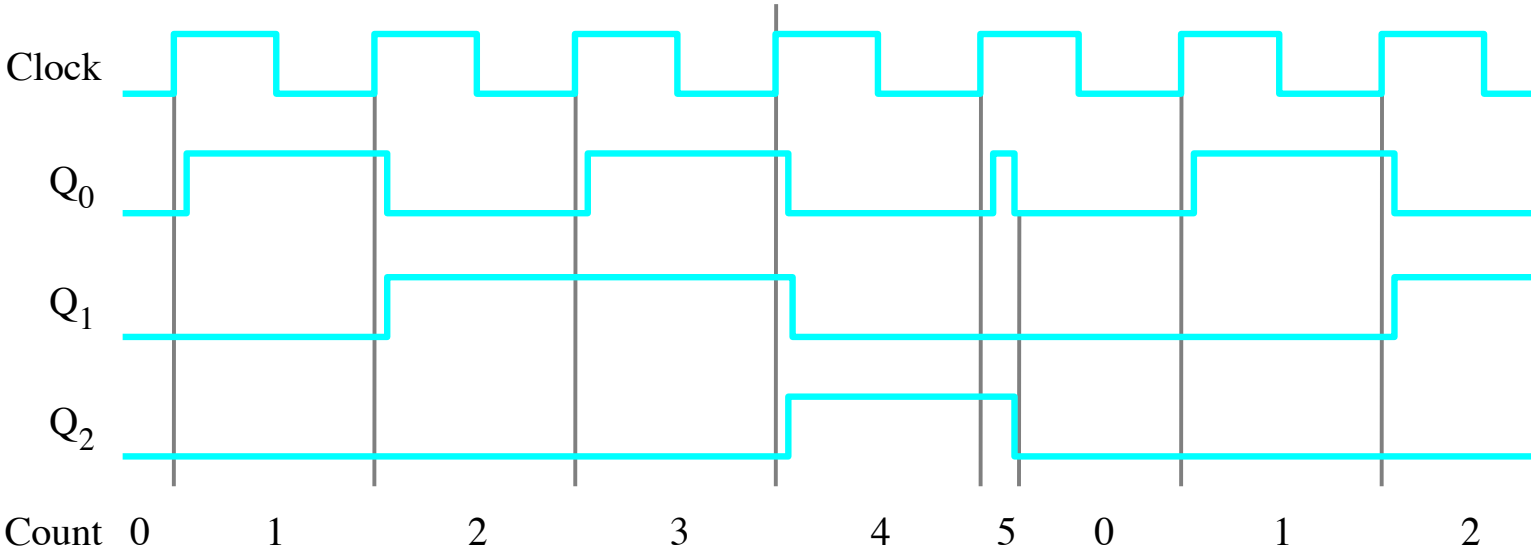
(b) Timing diagram

[Figure 5.25 from the textbook]

A modulo-6 counter with asynchronous reset



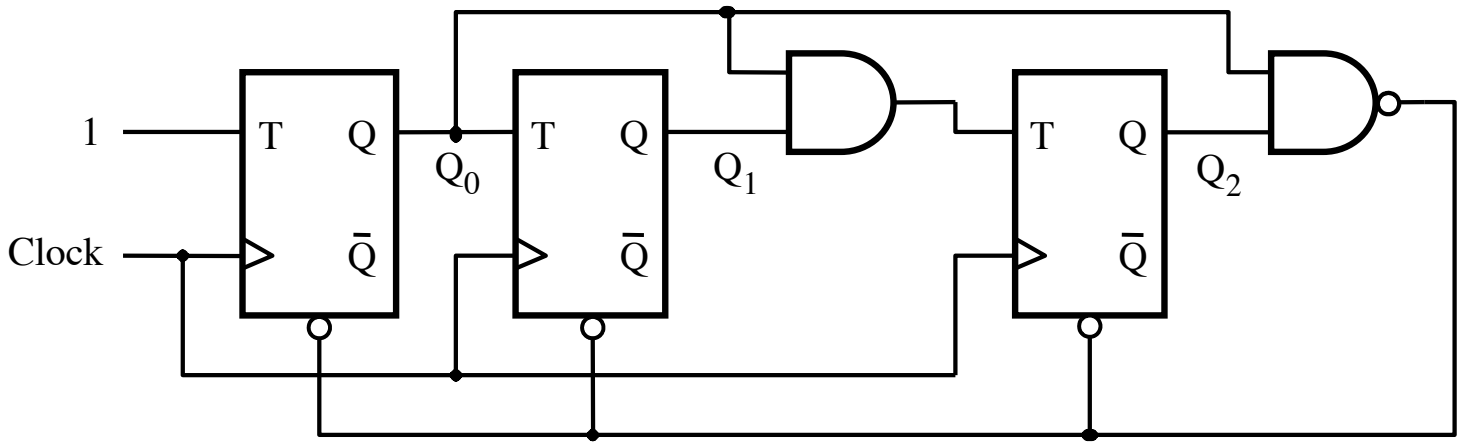
(a) Circuit



(b) Timing diagram

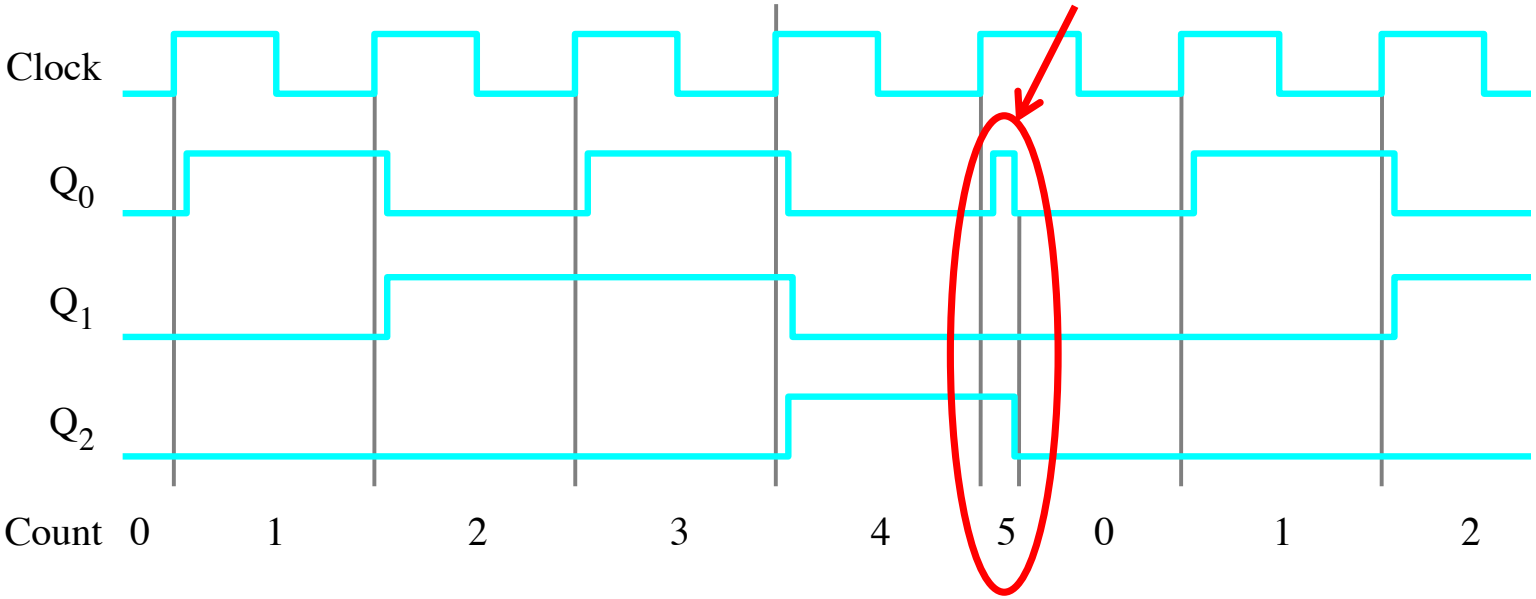
[Figure 5.26 from the textbook]

A modulo-6 counter with asynchronous reset



(a) Circuit

The number 5 is displayed
for a very short amount of time



(b) Timing diagram

[Figure 5.26 from the textbook]

Questions?

THE END