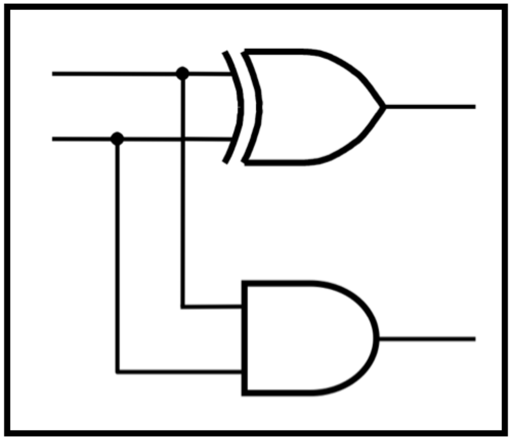


CprE 2810: Digital Logic

Instructor: Alexander Stoytchev

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

Counters

CprE 2810: Digital Logic
Iowa State University, Ames, IA
Copyright © Alexander Stoytchev

Administrative Stuff

- Homework 9 is due on Monday Nov 3 @ 10pm.
- Homework 10 will be due on Monday Nov 10 @ 10pm.
- The second midterm exam is this Friday
  (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).

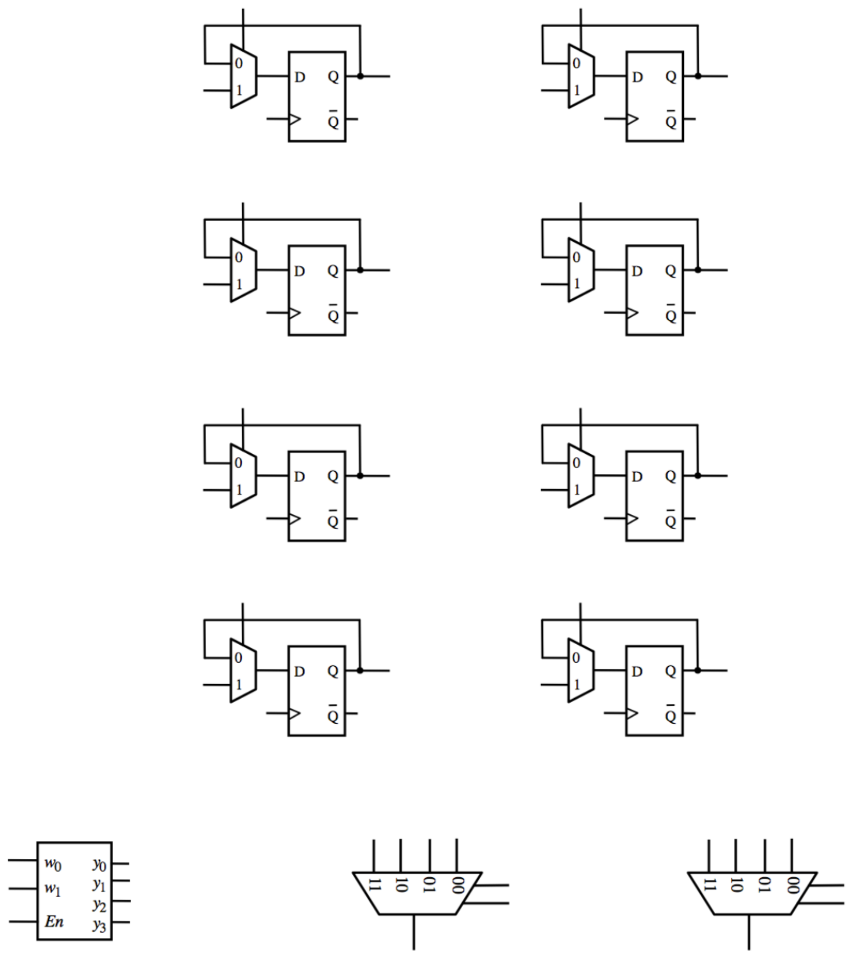
Quick Review

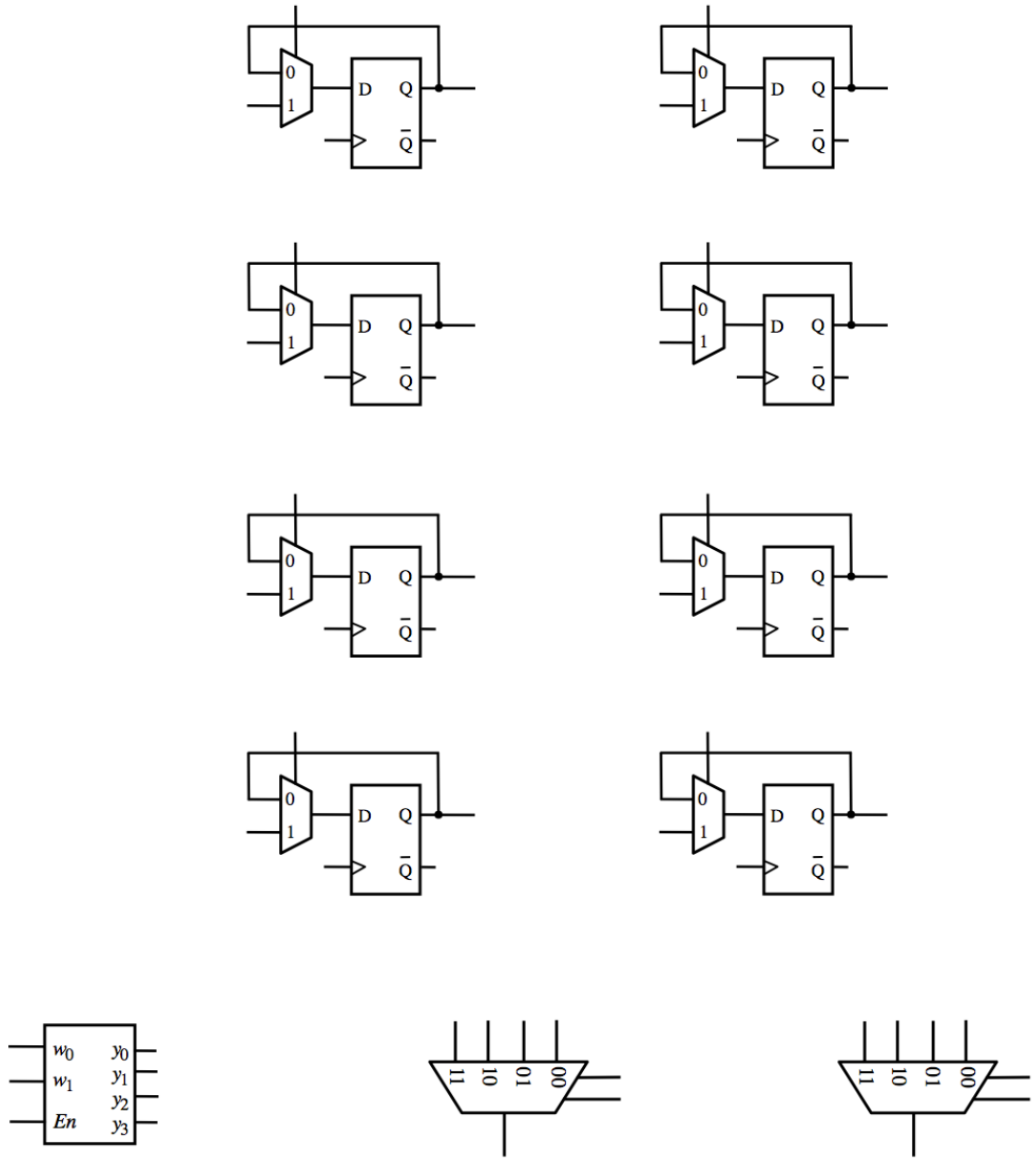
Register File

Register File (Definition)

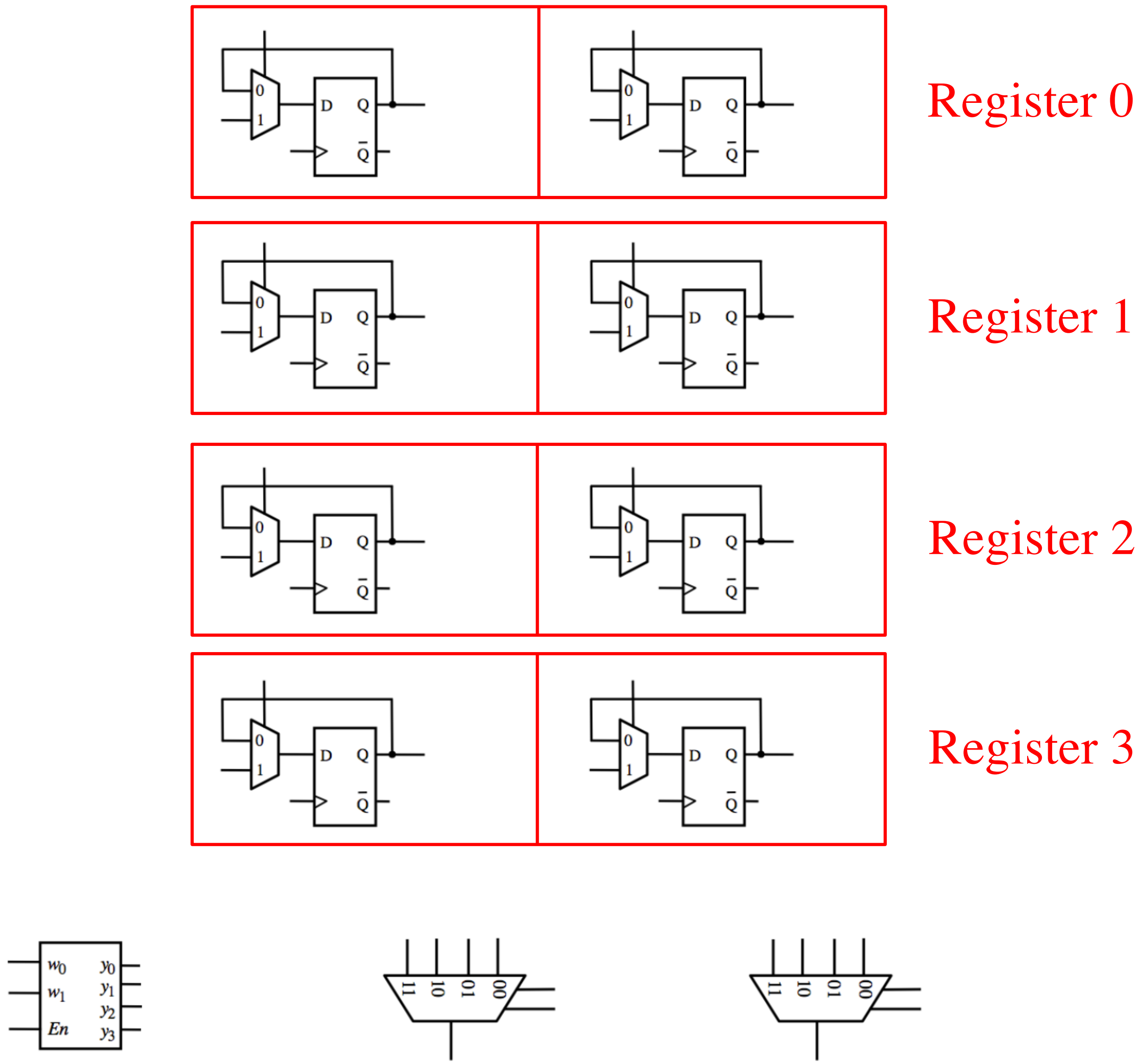
- **A circuit that can store several binary numbers, which can be read or written independently.**
- **Each number is stored in a separate n-bit register.**
- **To write: a decoder selects which resister is enabled for writing. An input bus provides the values.**
- **To read: a set of multiplexers select which register will be read and copied to the output bus.**
- **Some register files come with two read ports.**
  In those designs the multiplexer circuitry is doubled.

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.

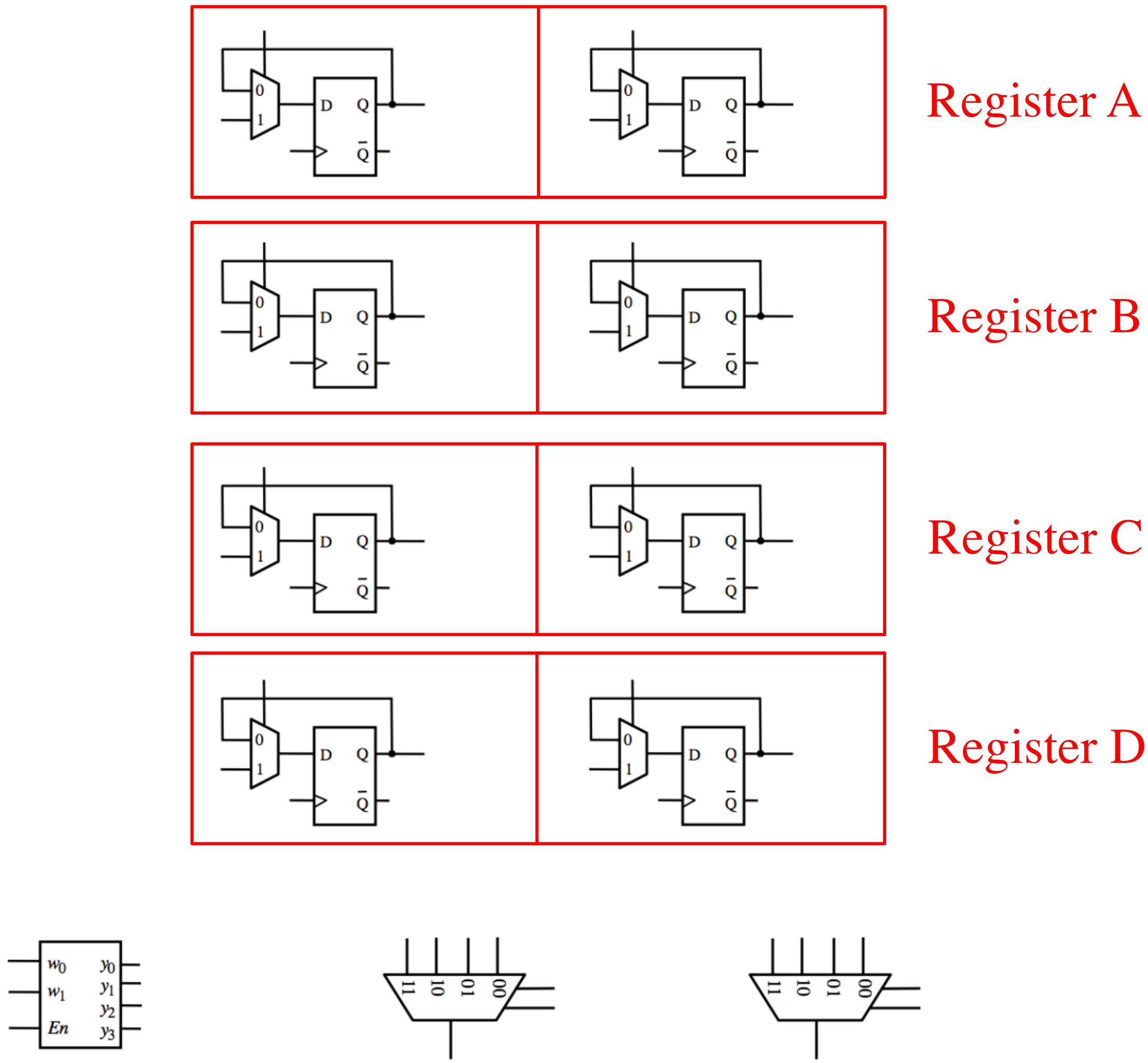


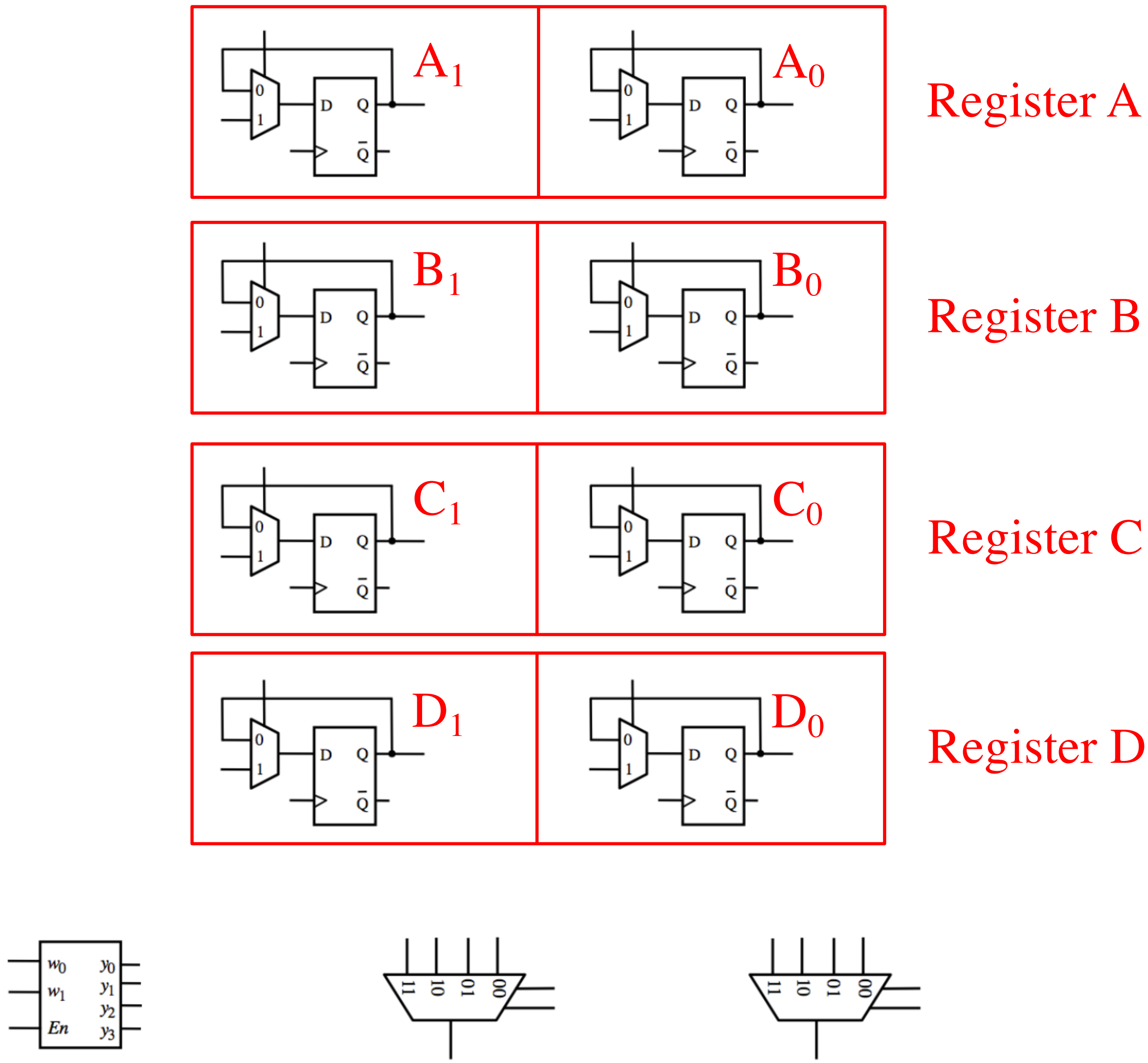


D
Q
$\overline{Q}$
0
1
w_0
w_1
En
y_0
y_1
y_2
y_3
11
10
01
00

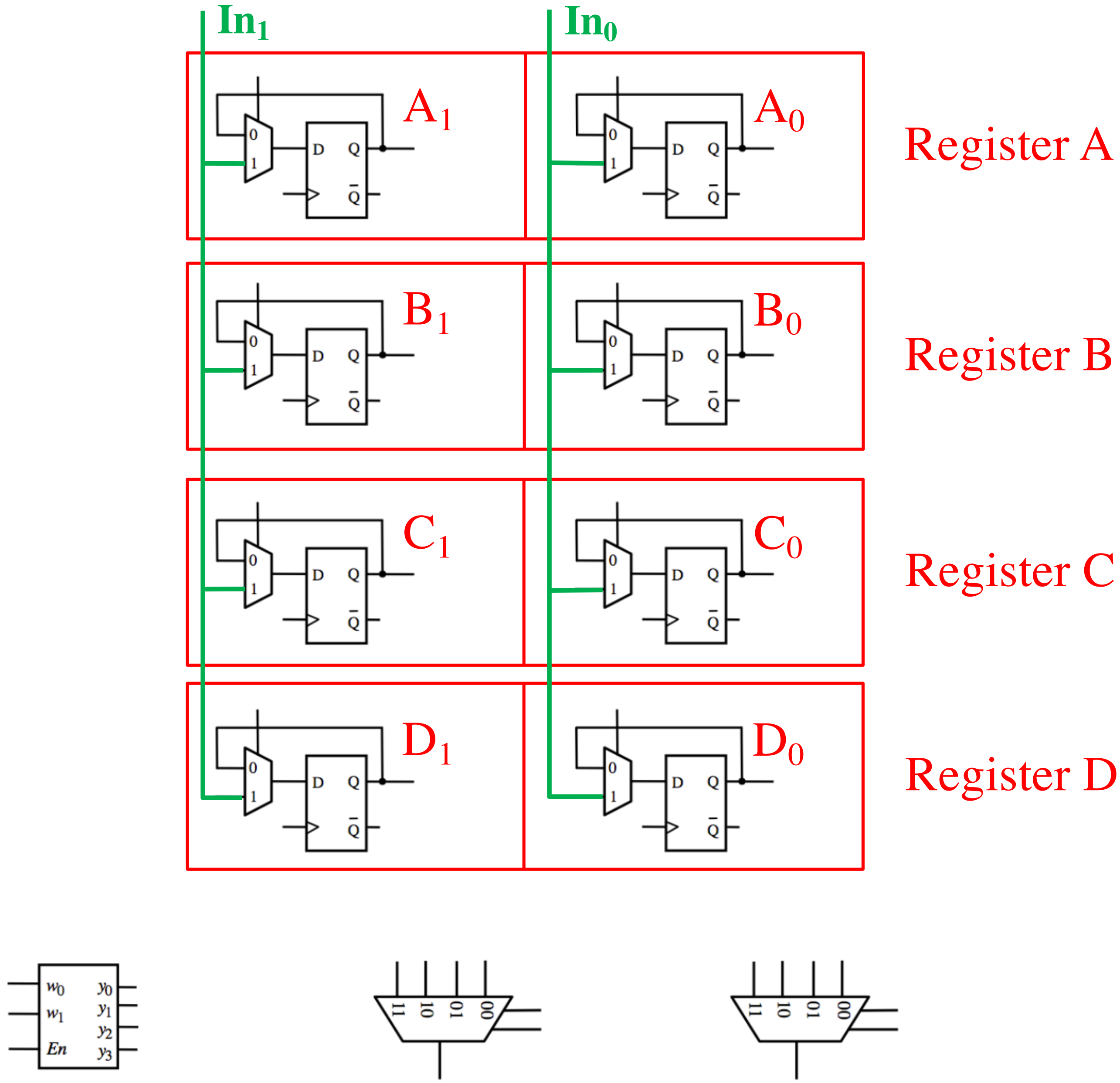


Register 0
Register 1
Register 2
Register 3
w0
w1
En
y0
y1
y2
y3
11
10
01
00

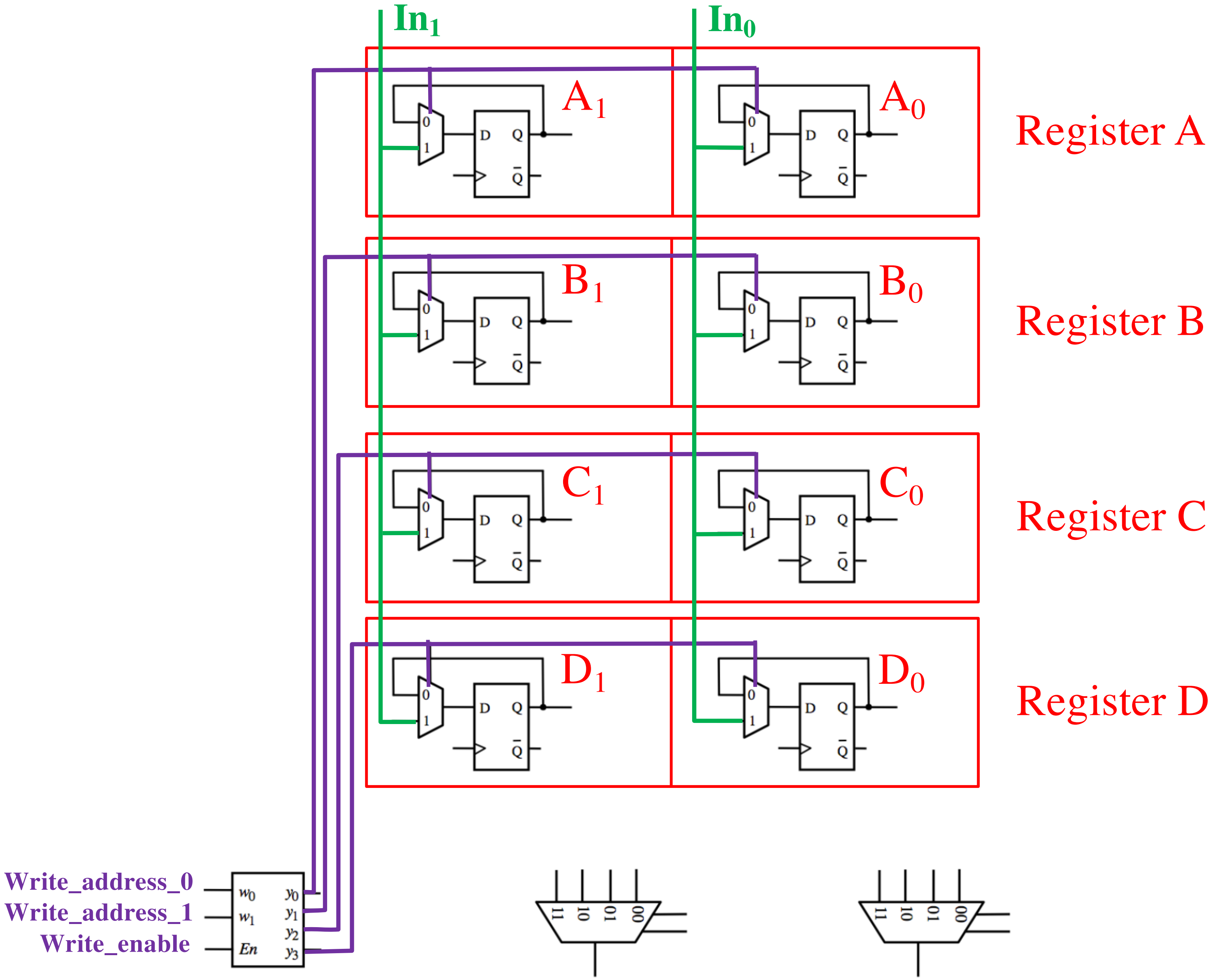

Register A
Register B
Register C
Register D
D
Q
0
1
w0
w1
En
y0
y1
y2
y3
11
10
01
00



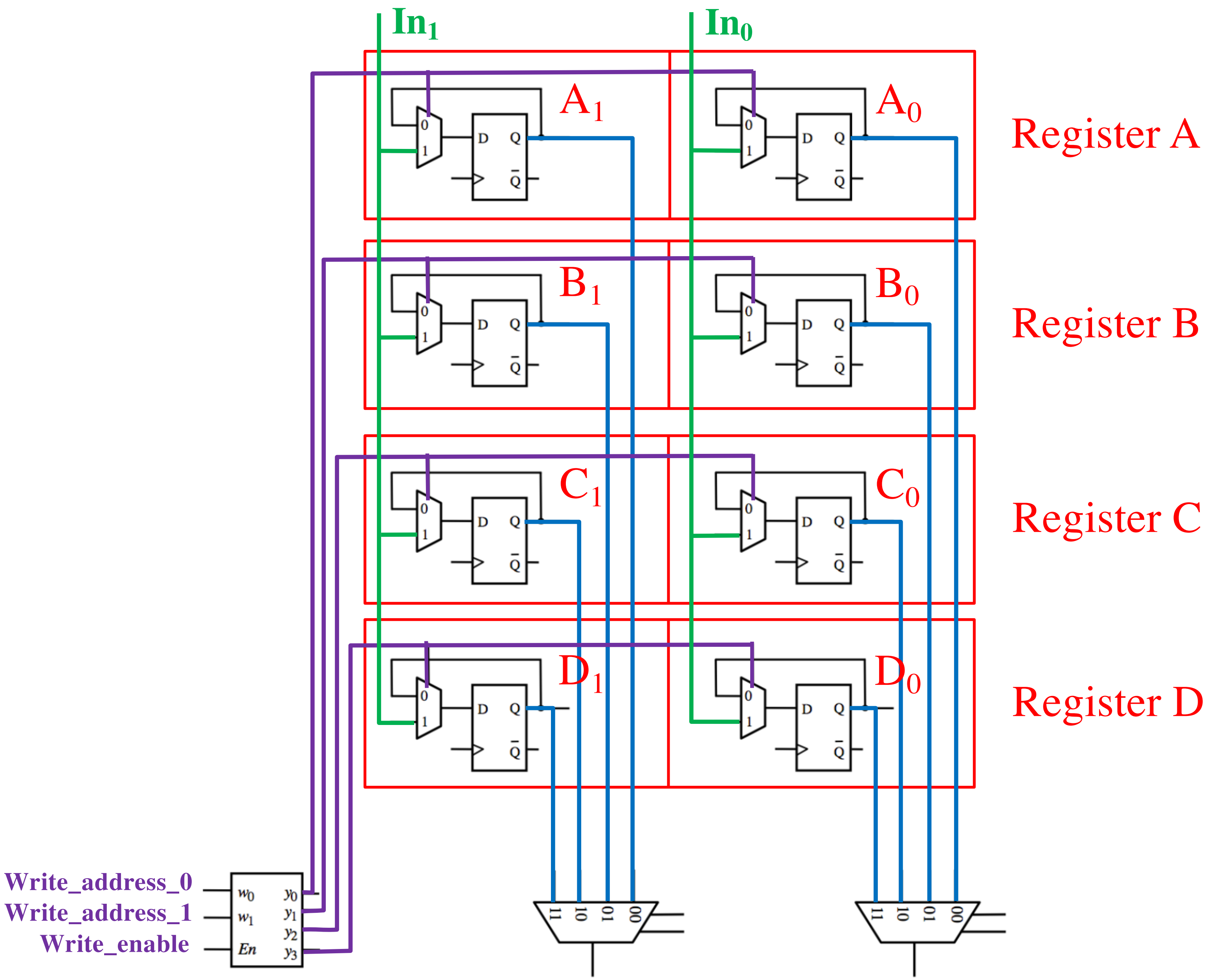
A1
A0
Register A
B1
B0
Register B
C1
C0
Register C
D1
D0
Register D
w0
w1
En
y0
y1
y2
y3
11
10
01
00
11
10
01
00



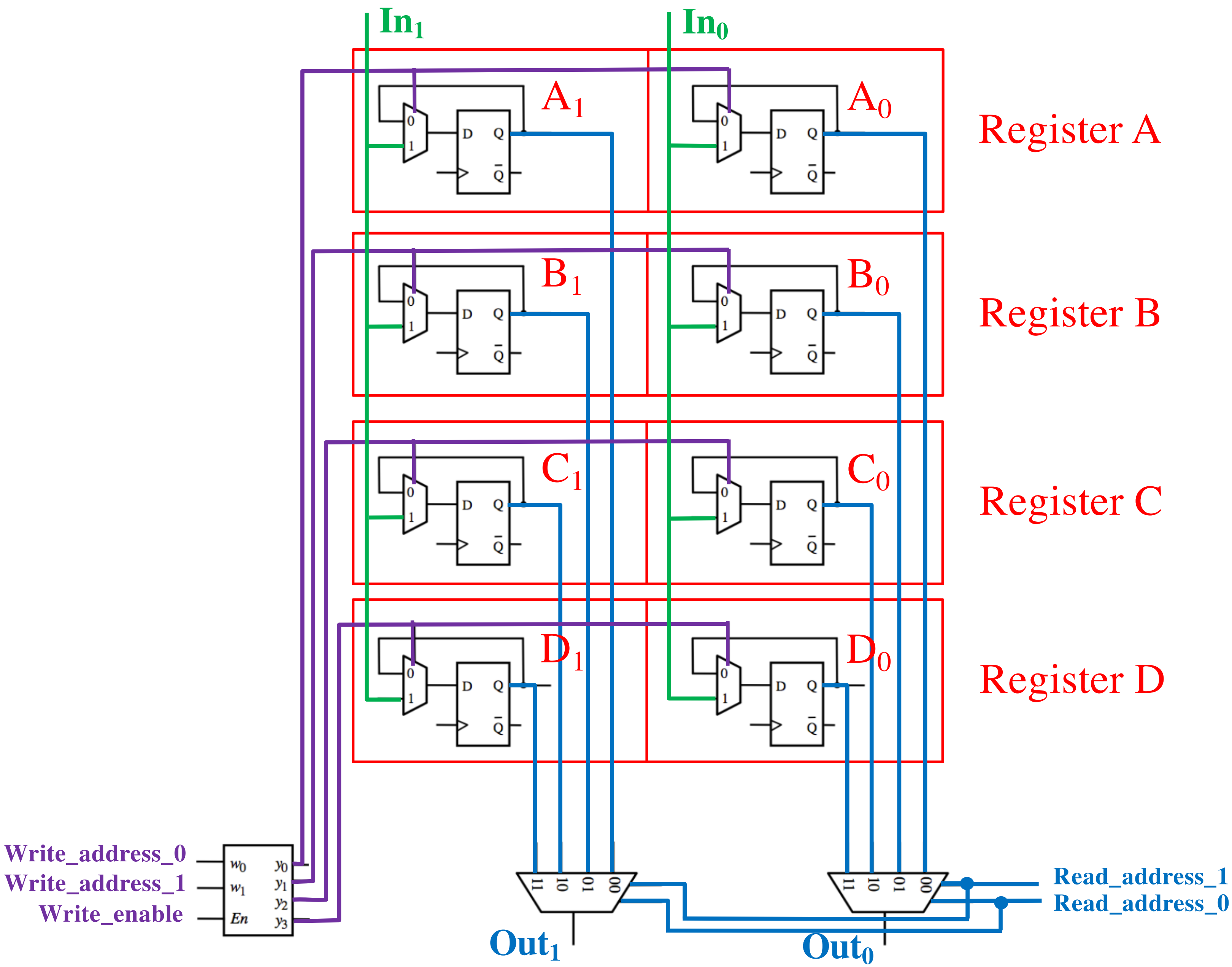
In_1
In_0
A_1
A_0
Register A
B_1
B_0
Register B
C_1
C_0
Register C
D_1
D_0
Register D
w_0
w_1
En
y_0
y_1
y_2
y_3
11 10 01 00
11 10 01 00



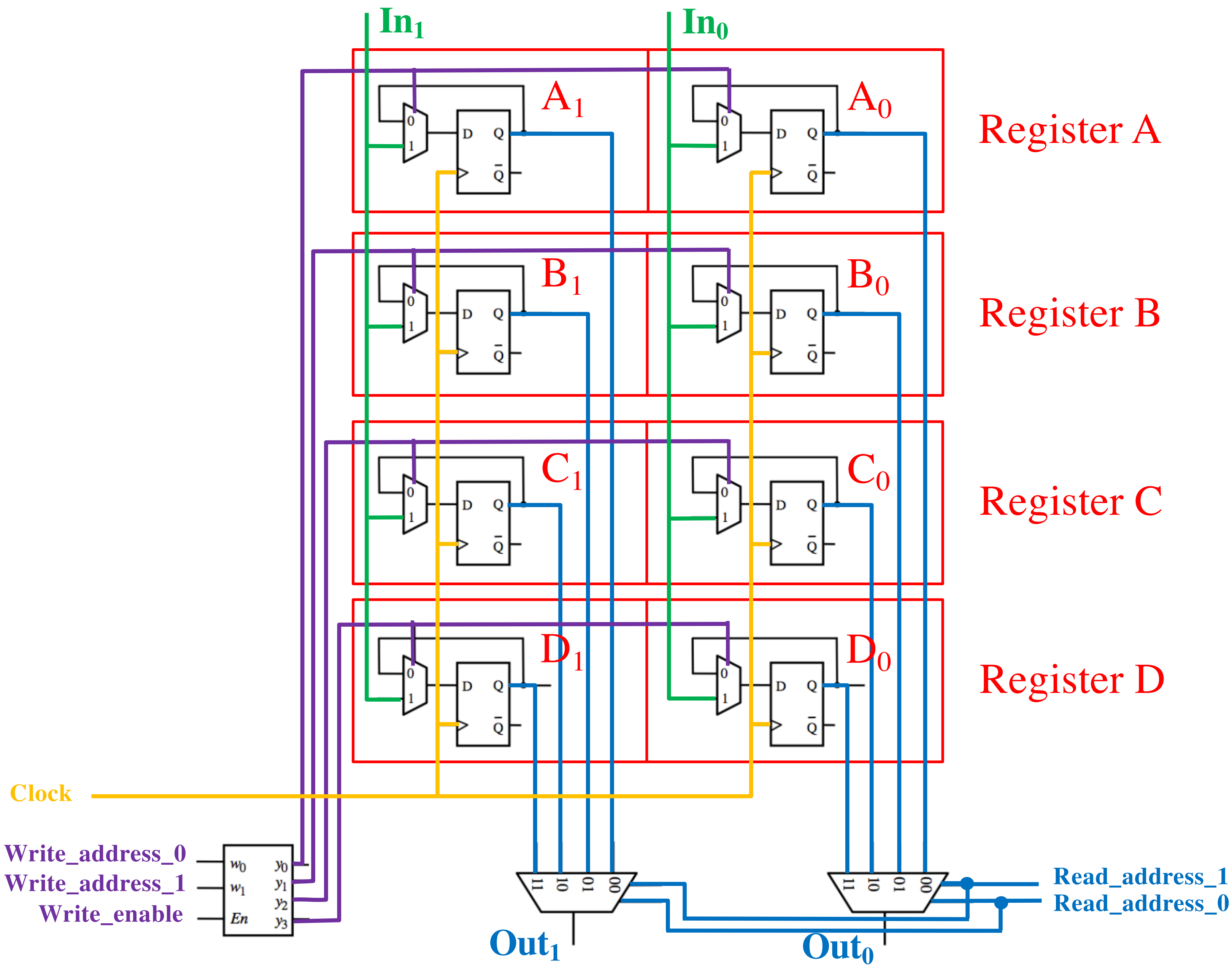
In1
In0
A1
A0
Register A
B1
B0
Register B
C1
C0
Register C
D1
D0
Register D
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11
10
01
00



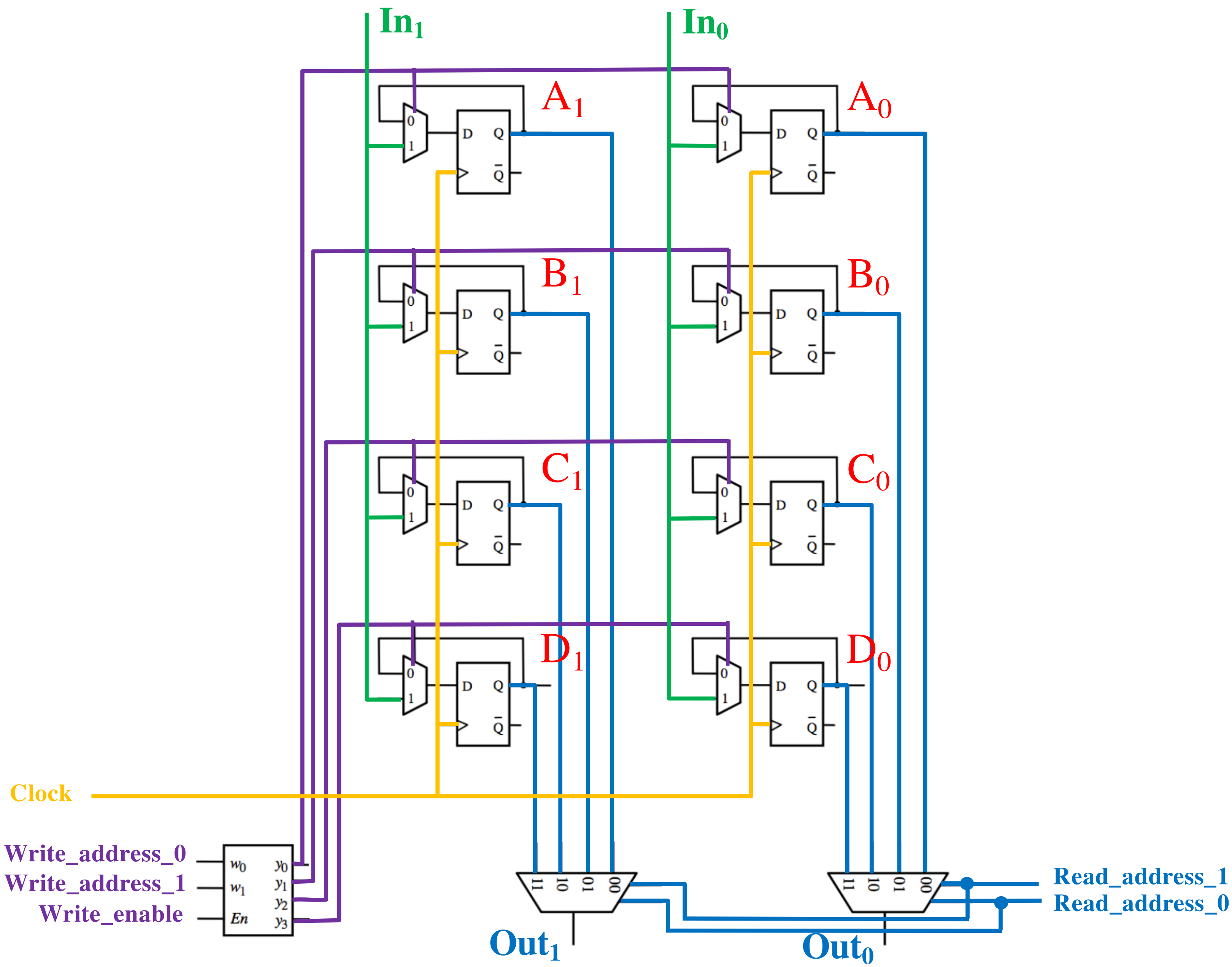
In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
Register A
Register B
Register C
Register D
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
D
Q
0
1
11
10
01
00



In1
In0
A1
A0
Register A
B1
B0
Register B
C1
C0
Register C
D1
D0
Register D
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

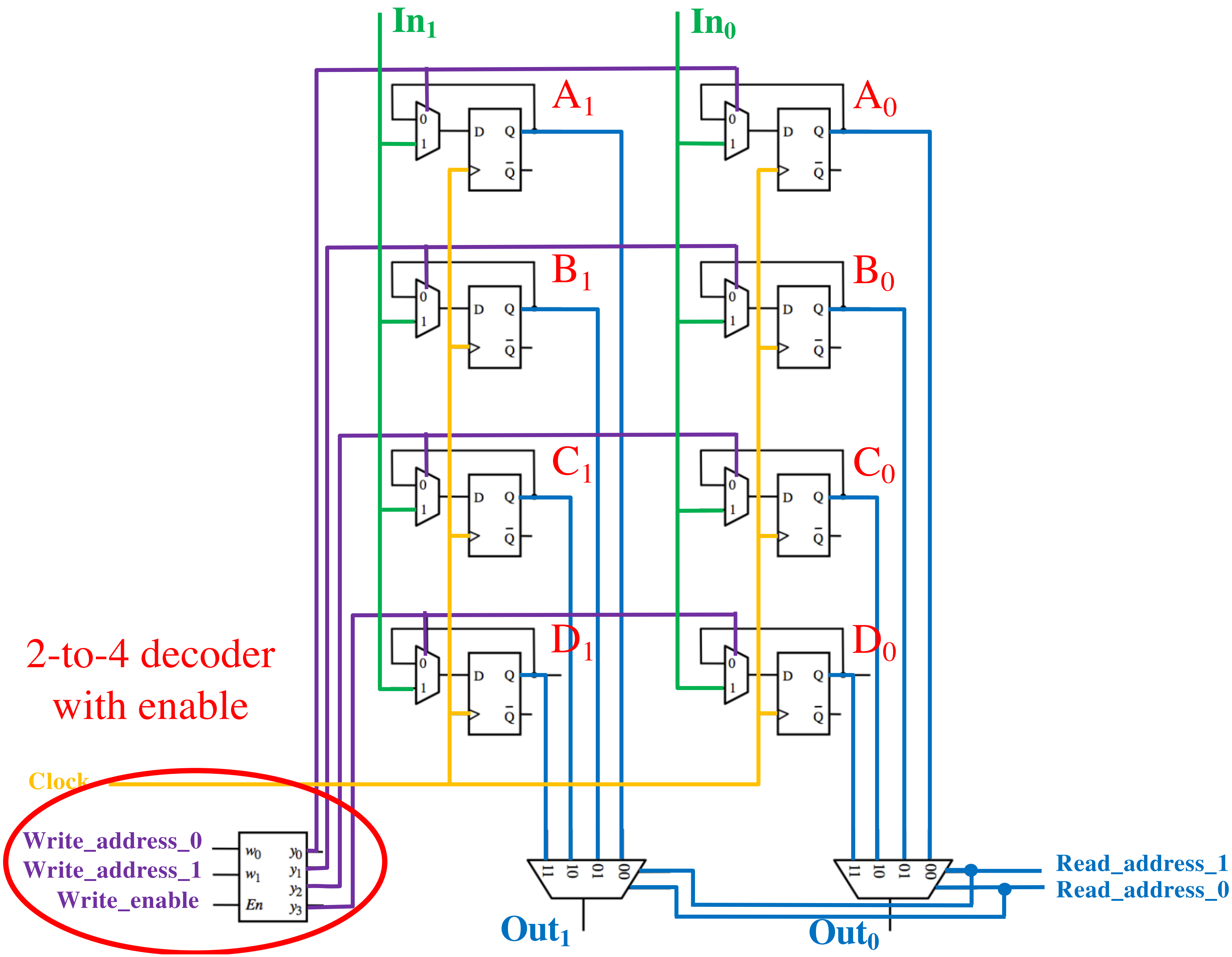


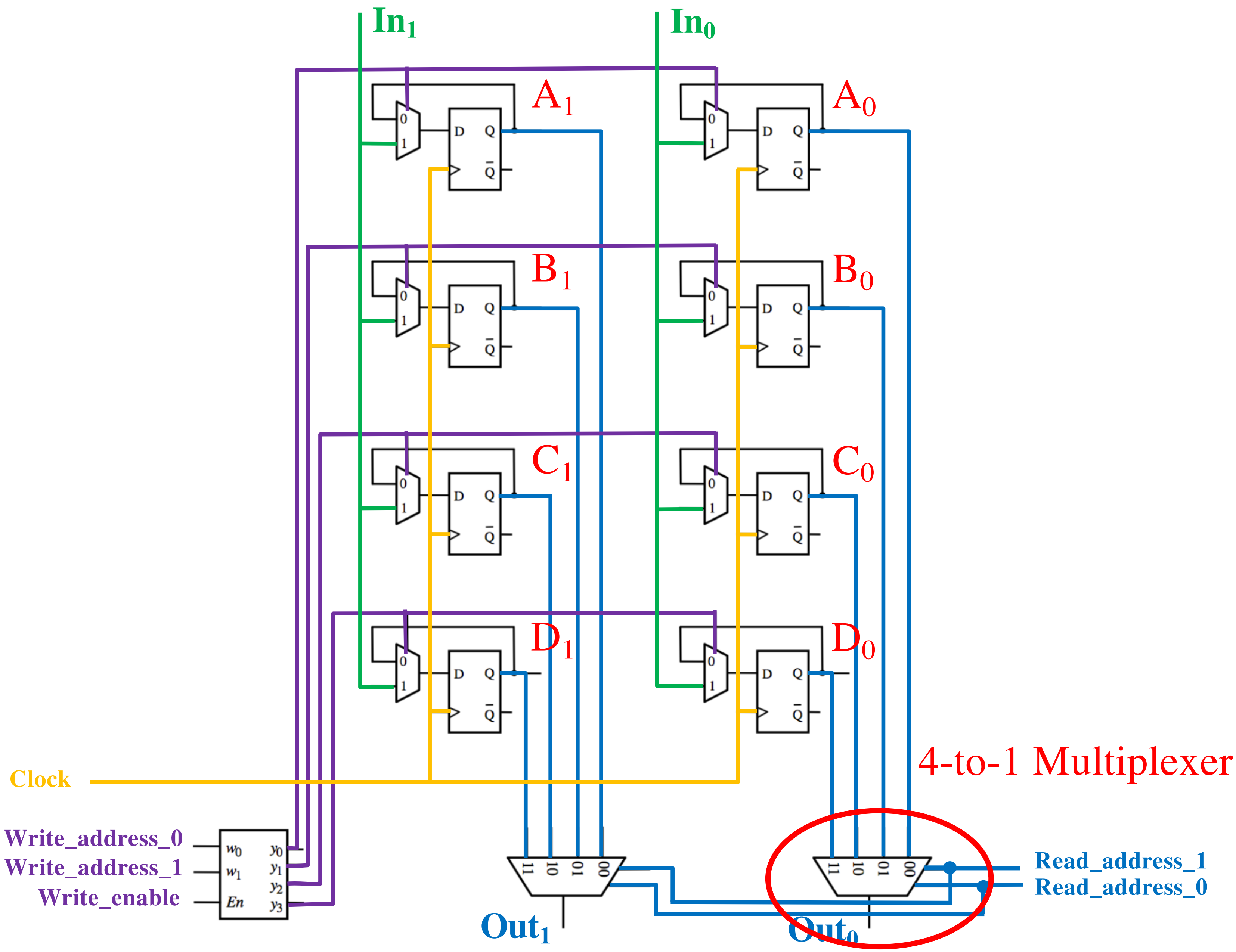
In1
In0
A1
A0
Register A
B1
B0
Register B
C1
C0
Register C
D1
D0
Register D
Clock
Write_address_0
Write_address_1
Write_enable
Read_address_1
Read_address_0
Out1
Out0



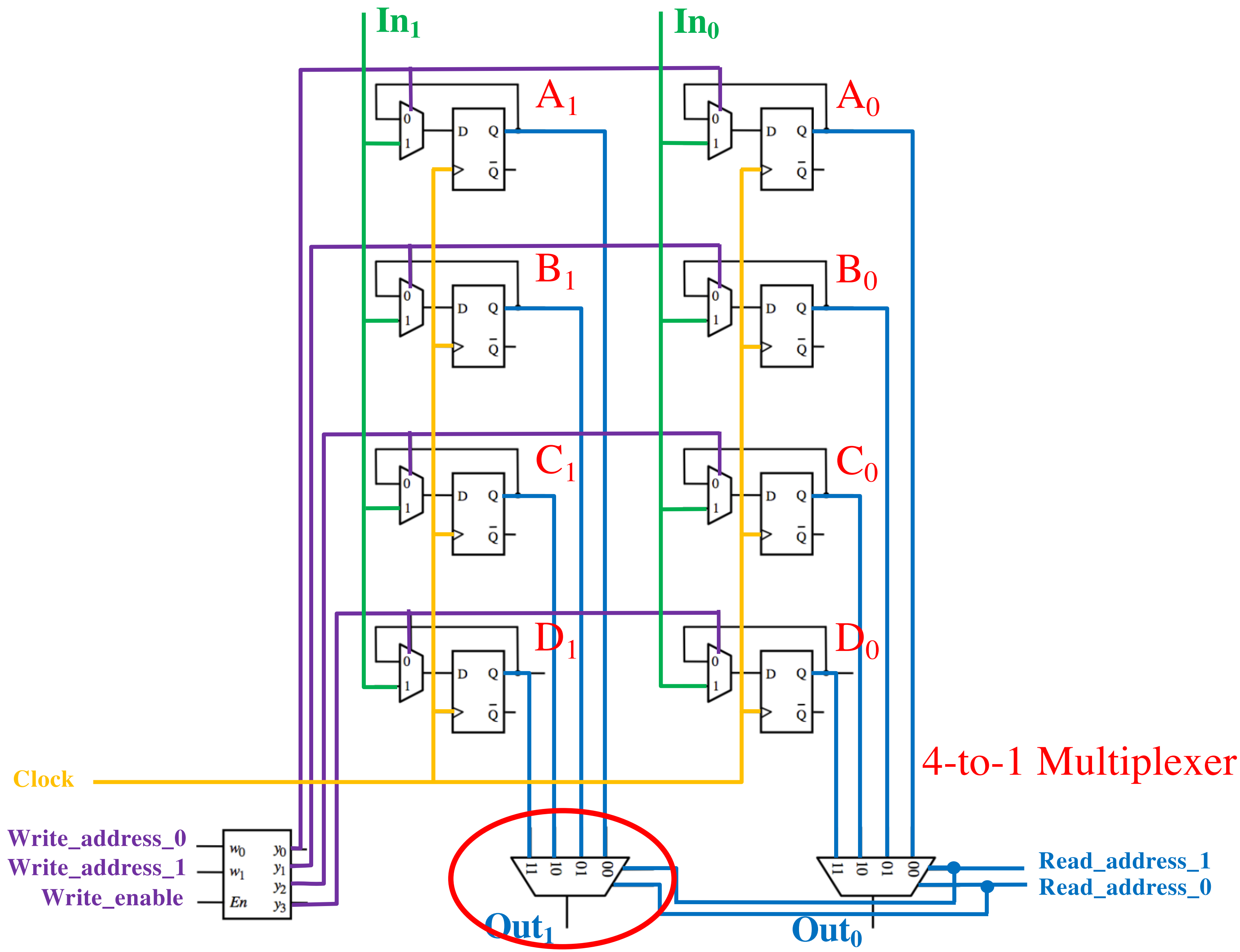
In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
D Q
Q
0
1
Clock
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11 10 01 00
Out1
Out0
Read_address_1
Read_address_0

Components of the Register File

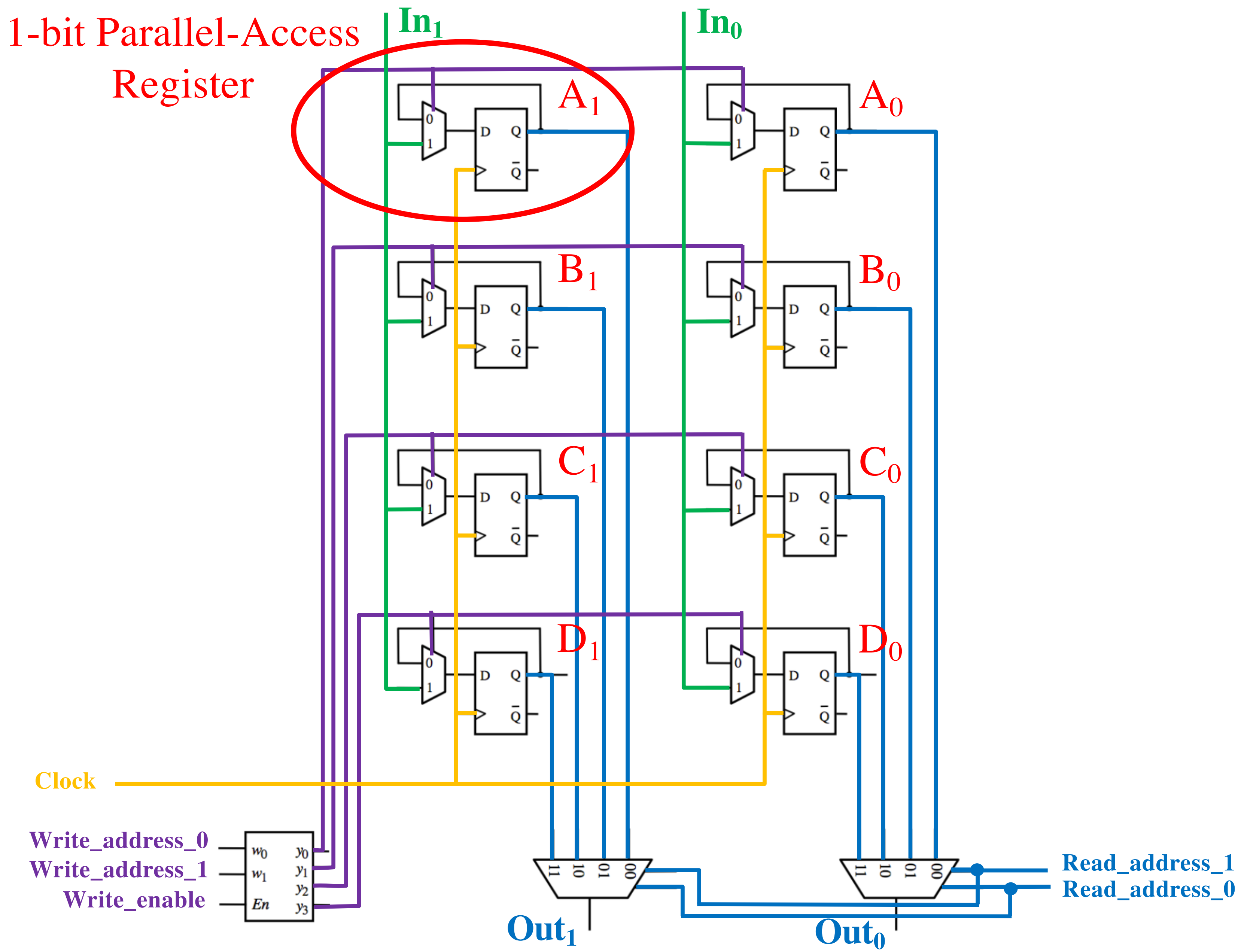

In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
2-to-4 decoder
with enable
Clock
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11
10
01
00
Out1
Out0
Read_address_1
Read_address_0

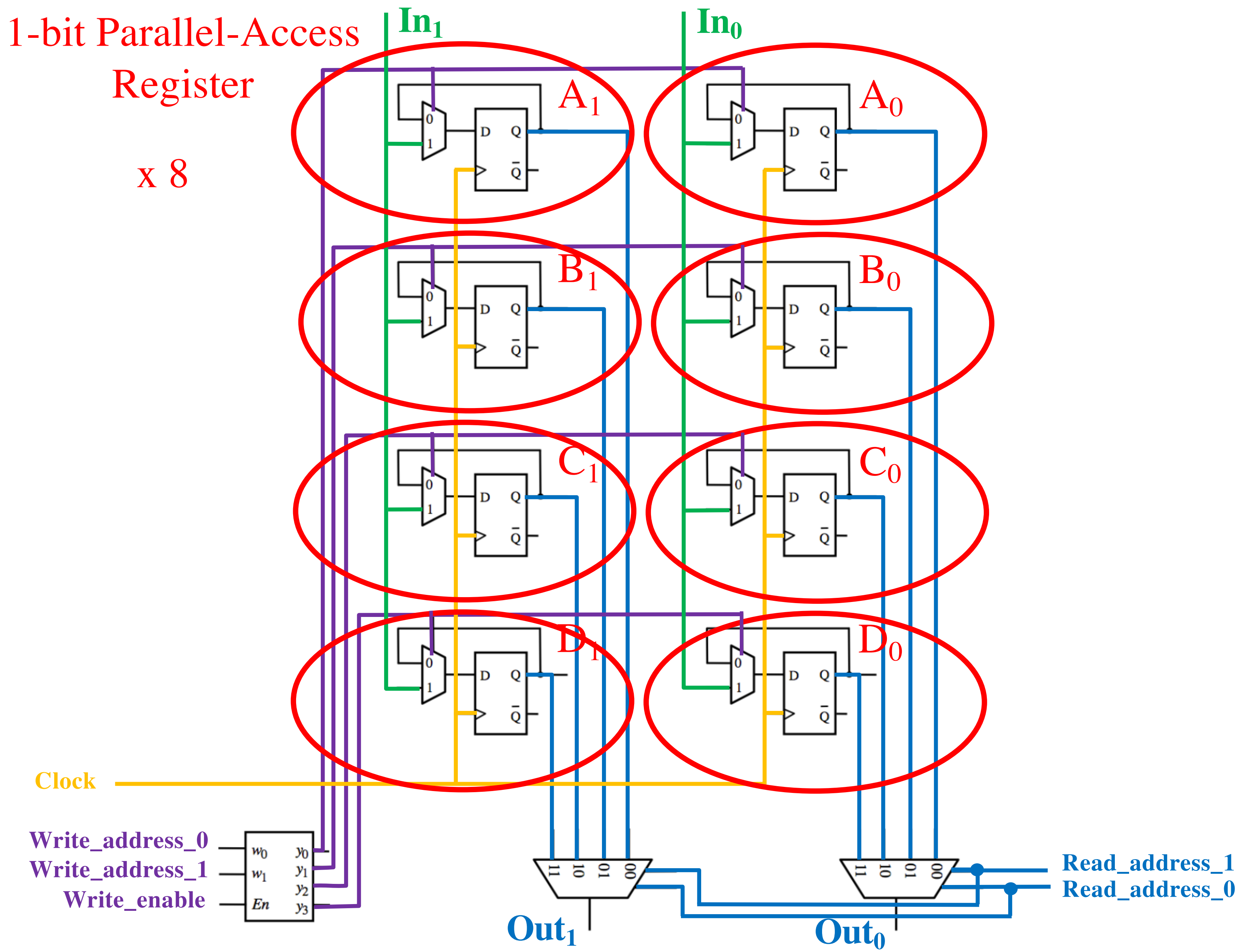


In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
D
Q
Clock
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11
10
01
00
Out1
Out0
4-to-1 Multiplexer
Read_address_1
Read_address_0

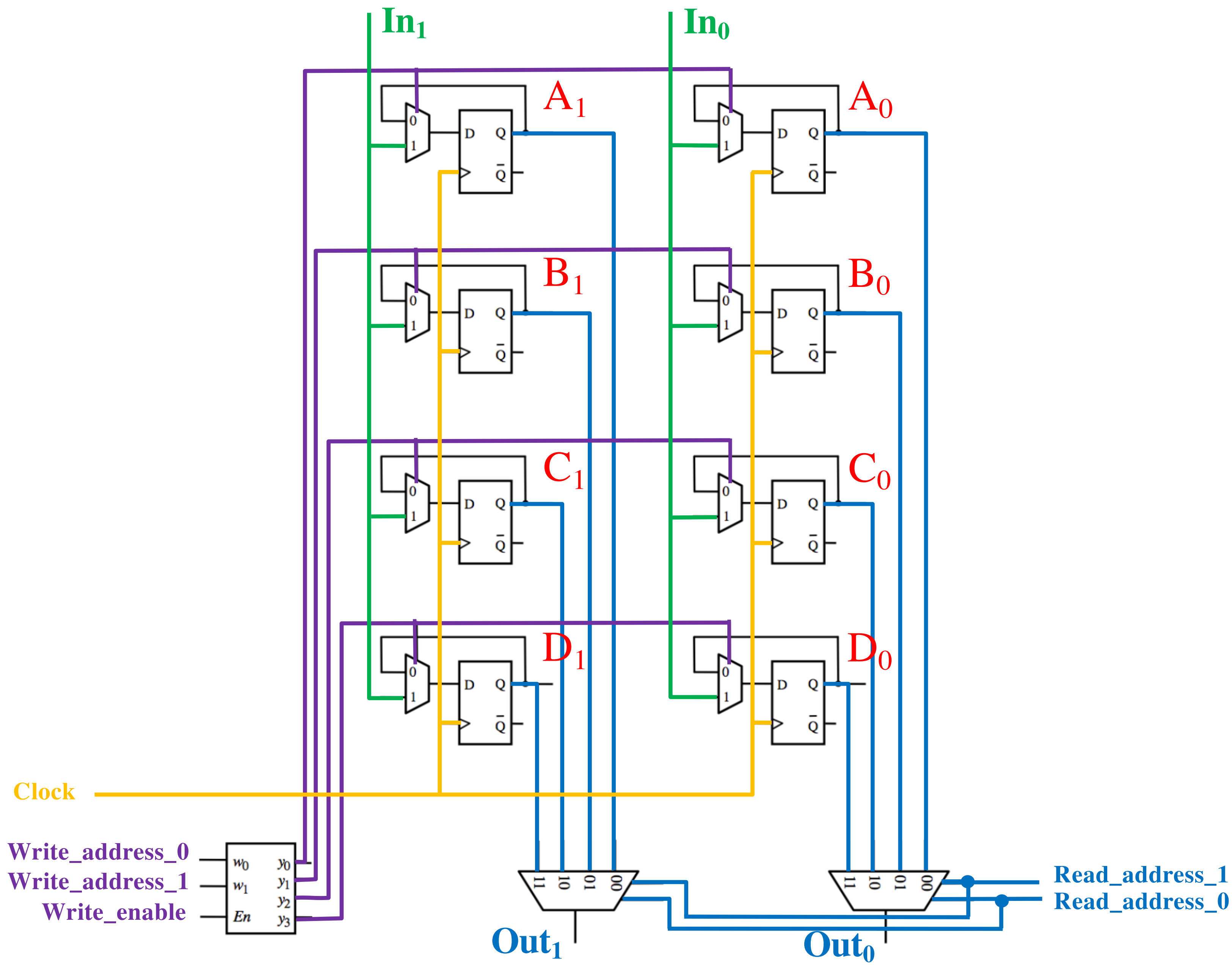


In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
D
Q
Q̄
0
1
4-to-1 Multiplexer
Clock
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11
10
01
00
Out1
Out0
Read_address_1
Read_address_0

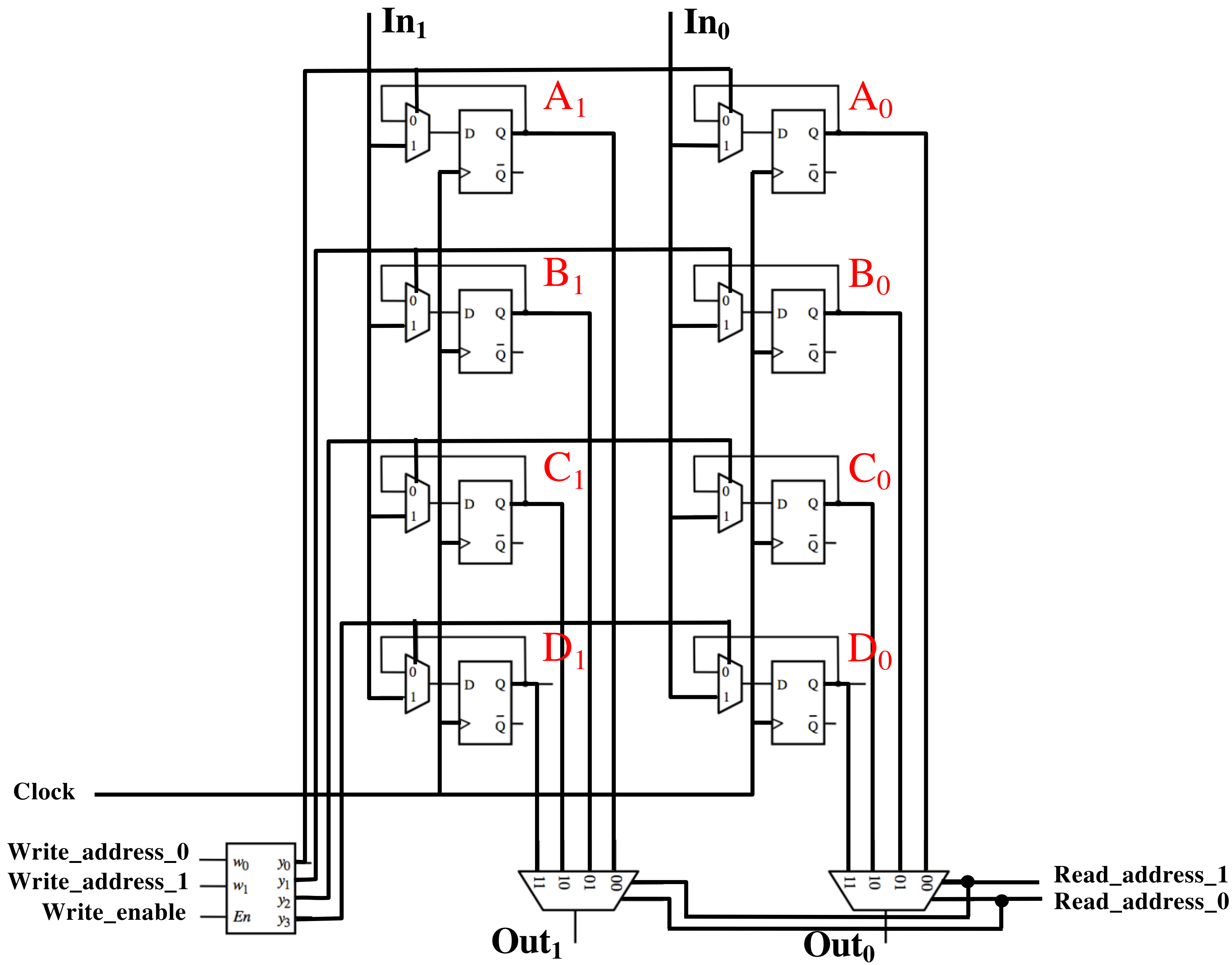

1-bit Parallel-Access Register
In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
D
Q
Q̄
0
1
Clock
Write_address_0
Write_address_1
Write_enable
w0
w1
En
y0
y1
y2
y3
11
10
01
00
Out1
Out0
Read_address_1
Read_address_0


1-bit Parallel-Access Register
x 8
In_1
In_0
A_1
A_0
B_1
B_0
C_1
C_0
D_1
D_0
Clock
Write_address_0
Write_address_1
Write_enable
w_0
w_1
En
y_0
y_1
y_2
y_3
11 10 01 00
Out_1
Out_0
Read_address_1
Read_address_0

Putting it all Together



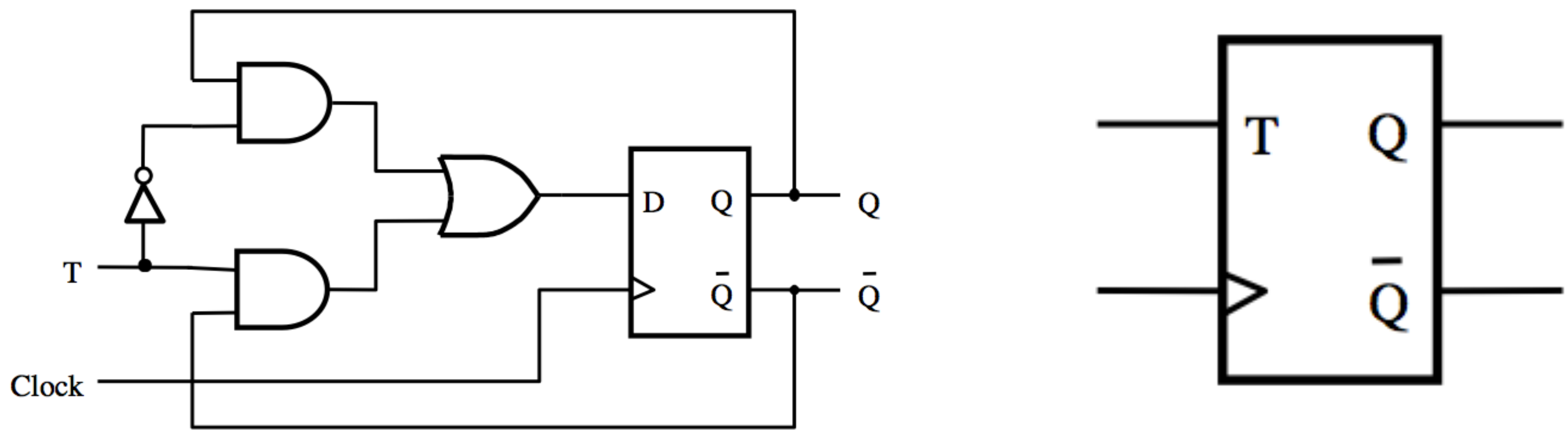
In_1
In_0
A_1
A_0
B_1
B_0
C_1
C_0
D_1
D_0
D Q $\bar{Q}$
0 1
Clock
Write_address_0
Write_address_1
Write_enable
w_0 w_1 En
y_0 y_1 y_2 y_3
11 10 01 00
Out_1
Out_0
Read_address_1
Read_address_0



In1
In0
A1
A0
B1
B0
C1
C0
D1
D0
D
Q
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

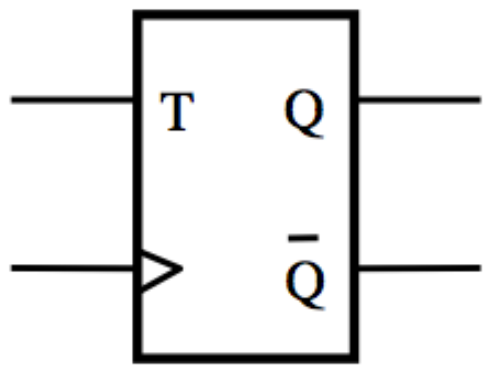
T Flip-Flop
(circuit and graphical symbol)



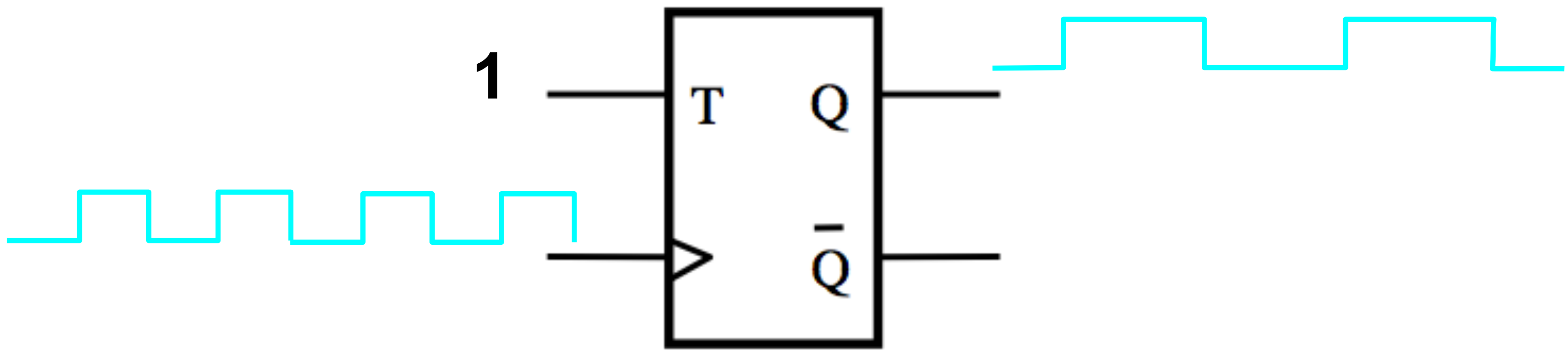
[Figure 5.15a,c from the textbook]

The output of the T Flip-Flop

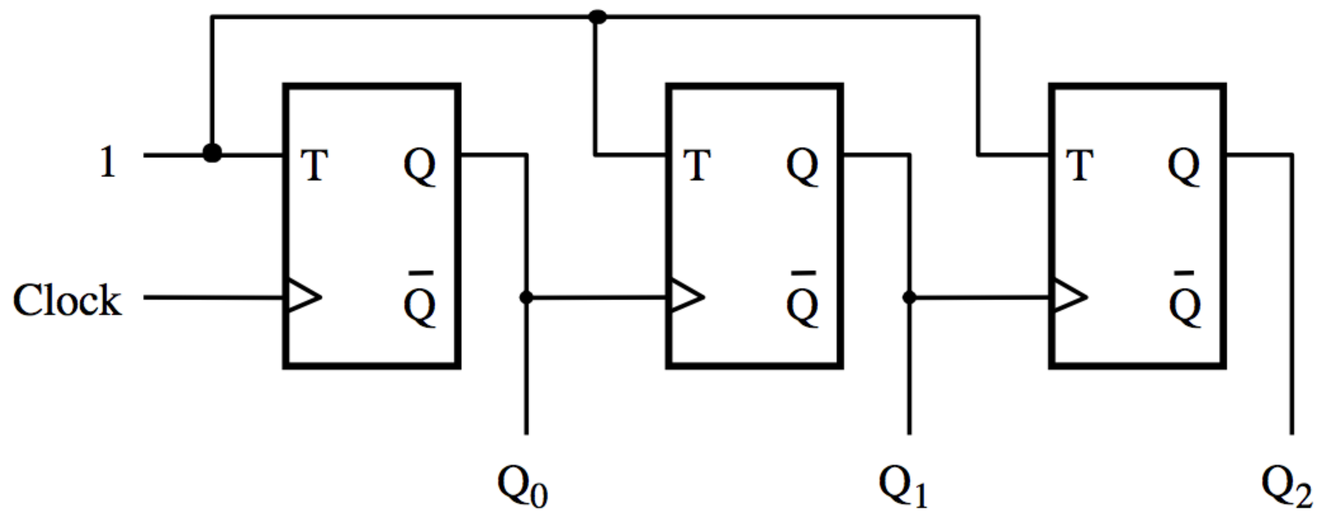
divides the frequency of the clock by 2



The output of the T Flip-Flop
divides the frequency of the clock by 2

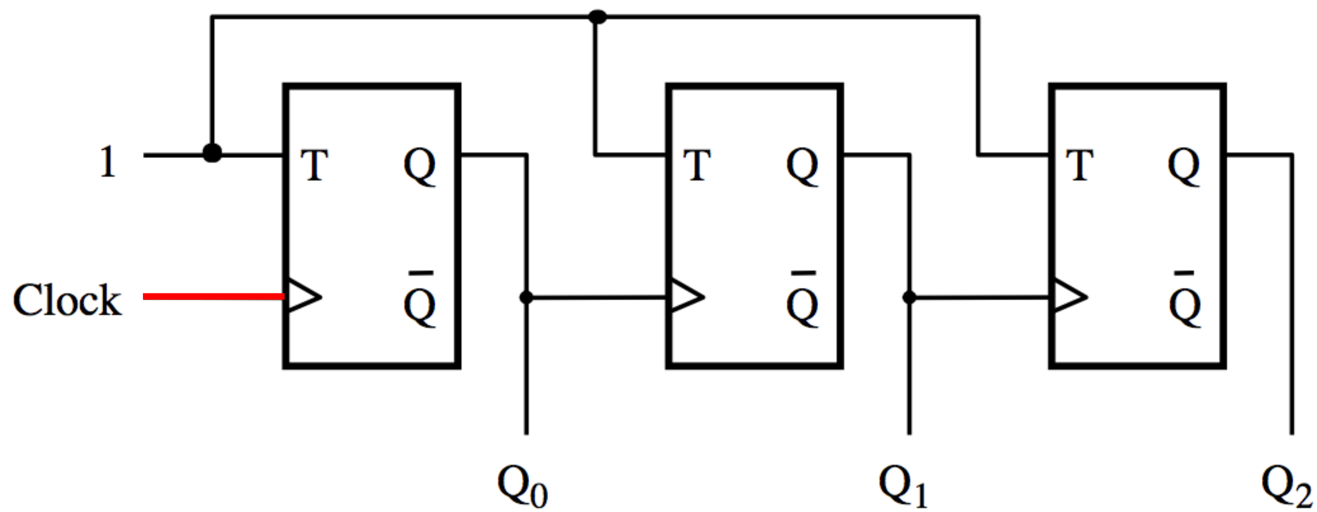


A three-bit down-counter



[Figure 5.20 from the textbook]

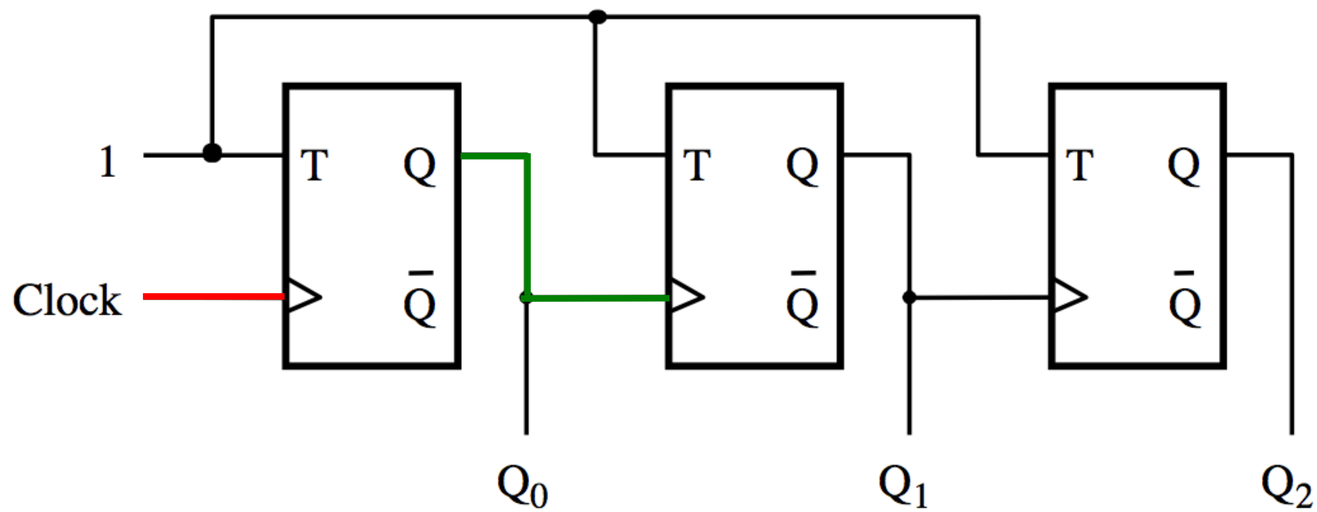
A three-bit down-counter



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

[Figure 5.20 from the textbook]

A three-bit down-counter

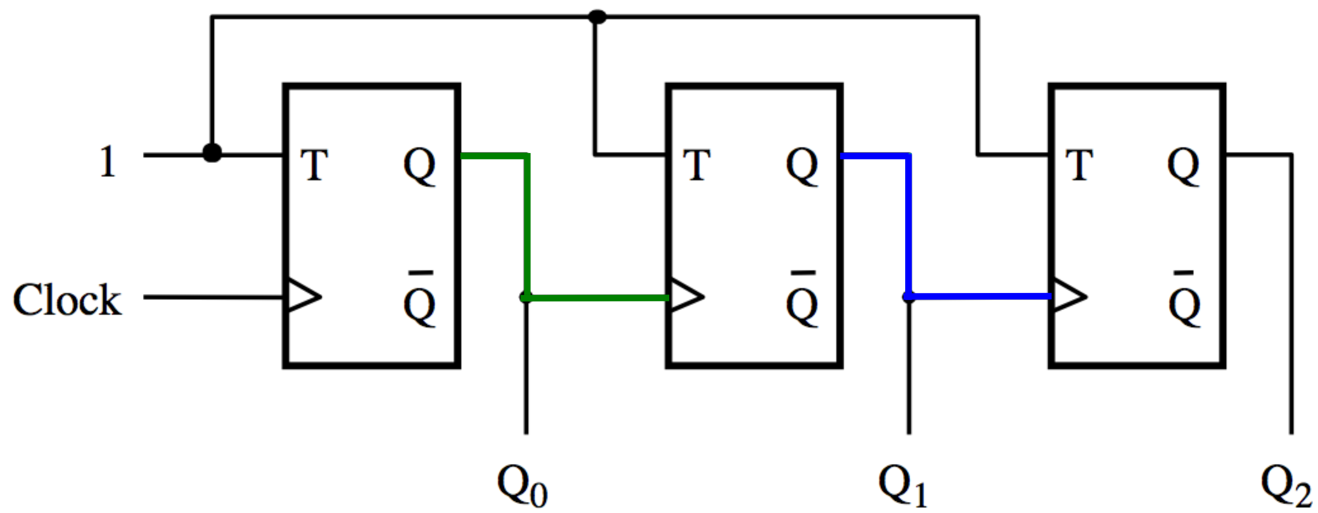


The first flip-flop changes
on the positive edge of the clock

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

[Figure 5.20 from the textbook]

A three-bit down-counter



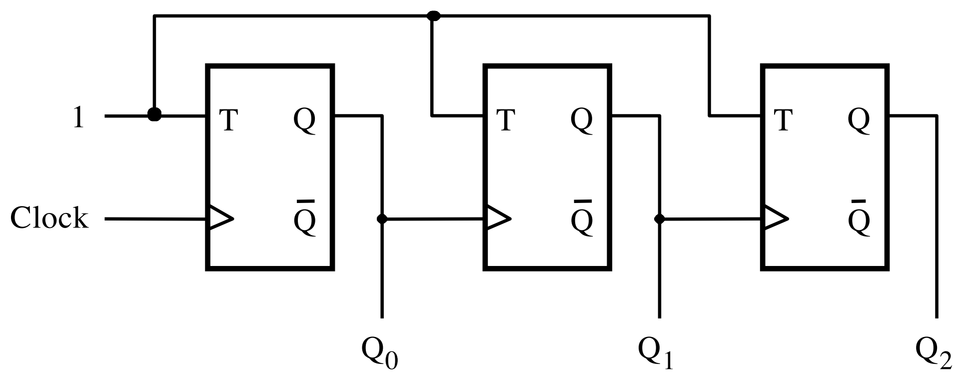
The first flip-flop changes on the positive edge of the clock

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

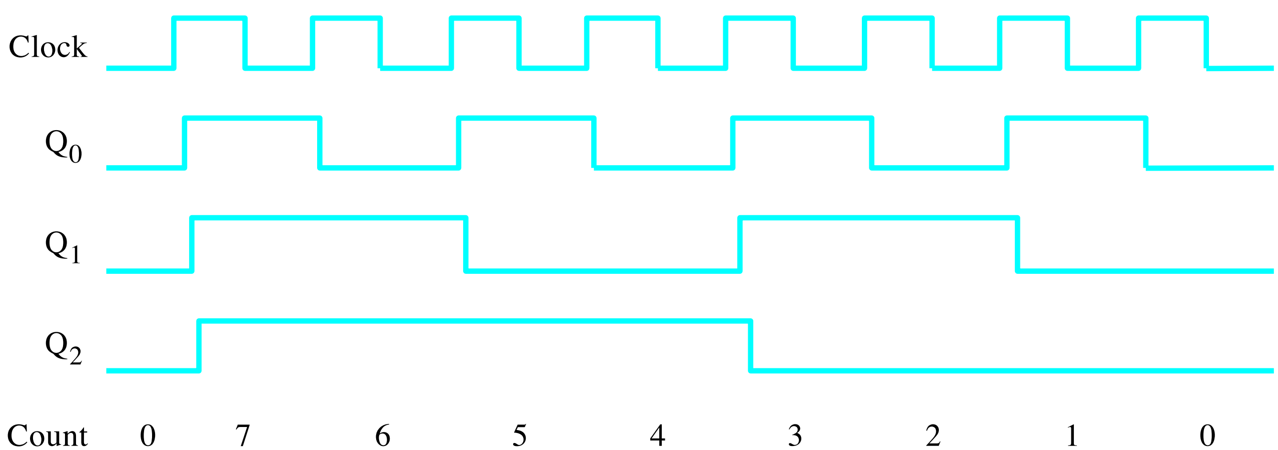
The third flip-flop changes on the positive edge of Q_1

[Figure 5.20 from the textbook]

A three-bit down-counter



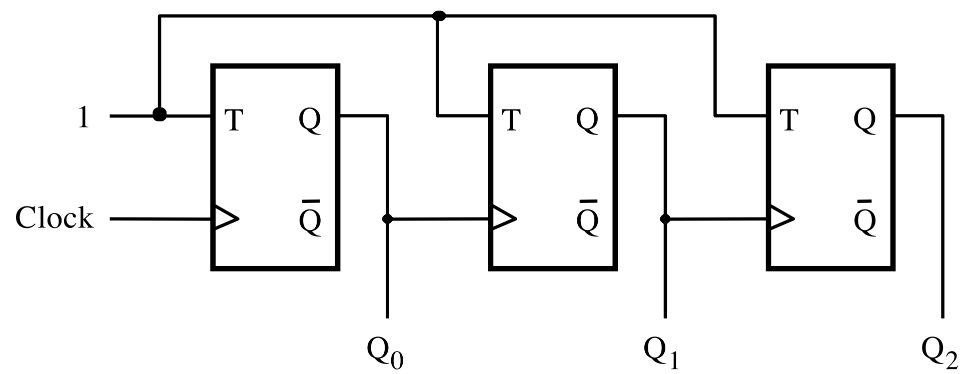
(a) Circuit



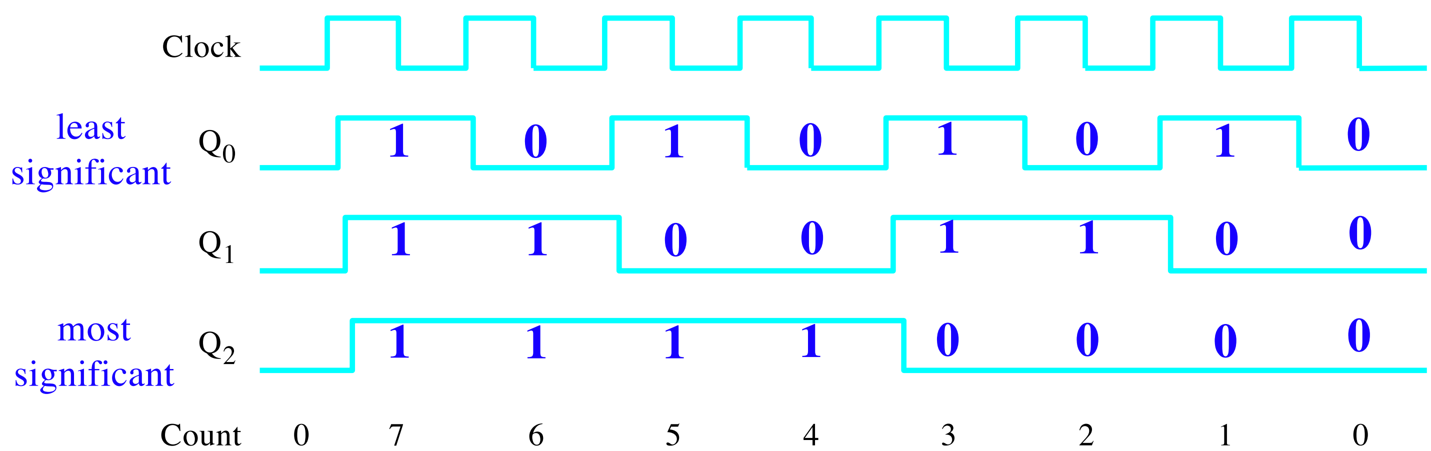
(b) Timing diagram

[Figure 5.20 from the textbook]

A three-bit down-counter

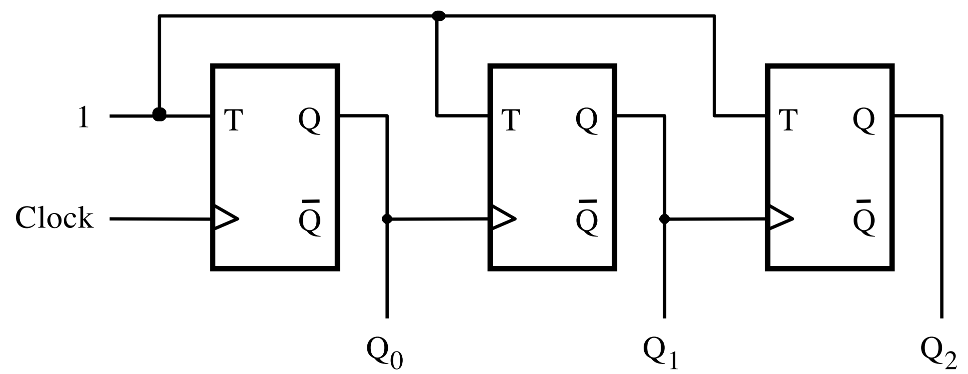


(a) Circuit

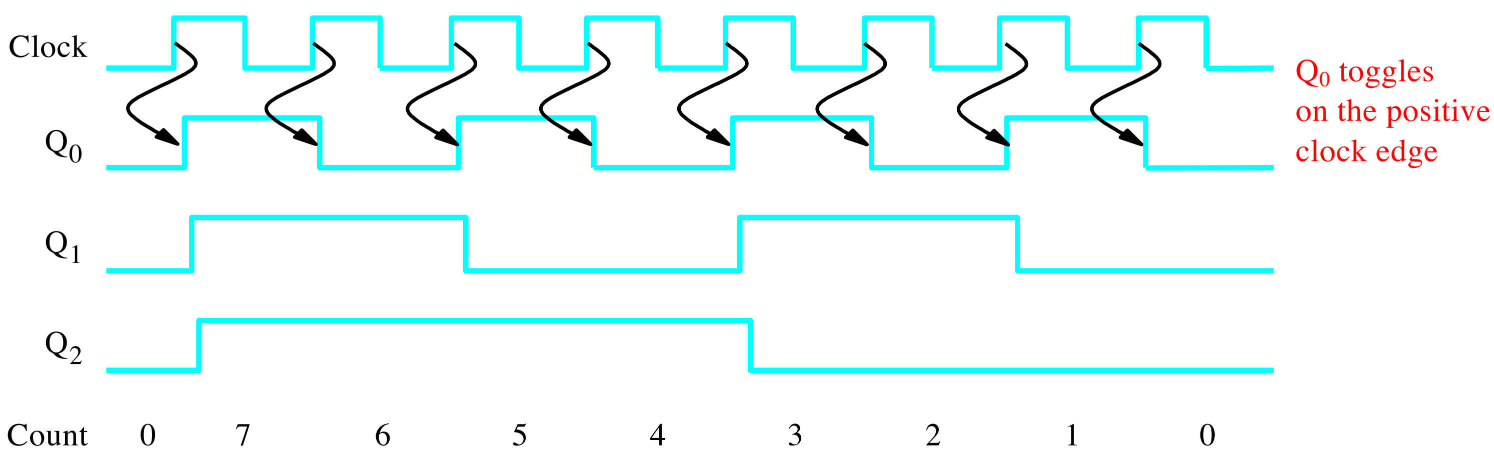


(b) Timing diagram

A three-bit down-counter

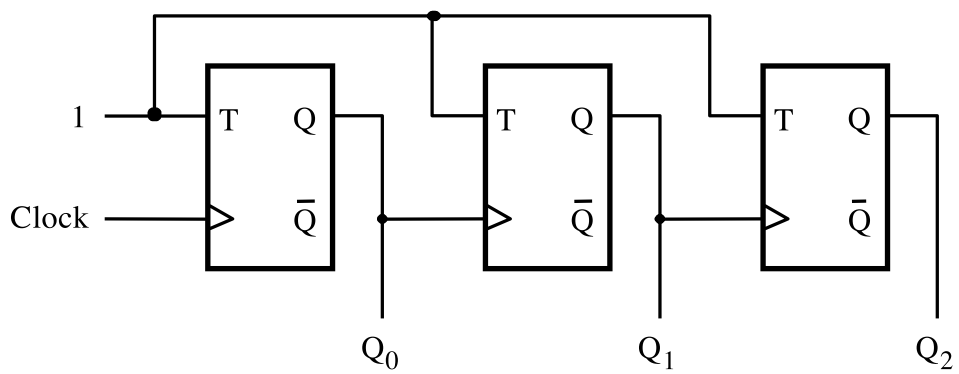


(a) Circuit

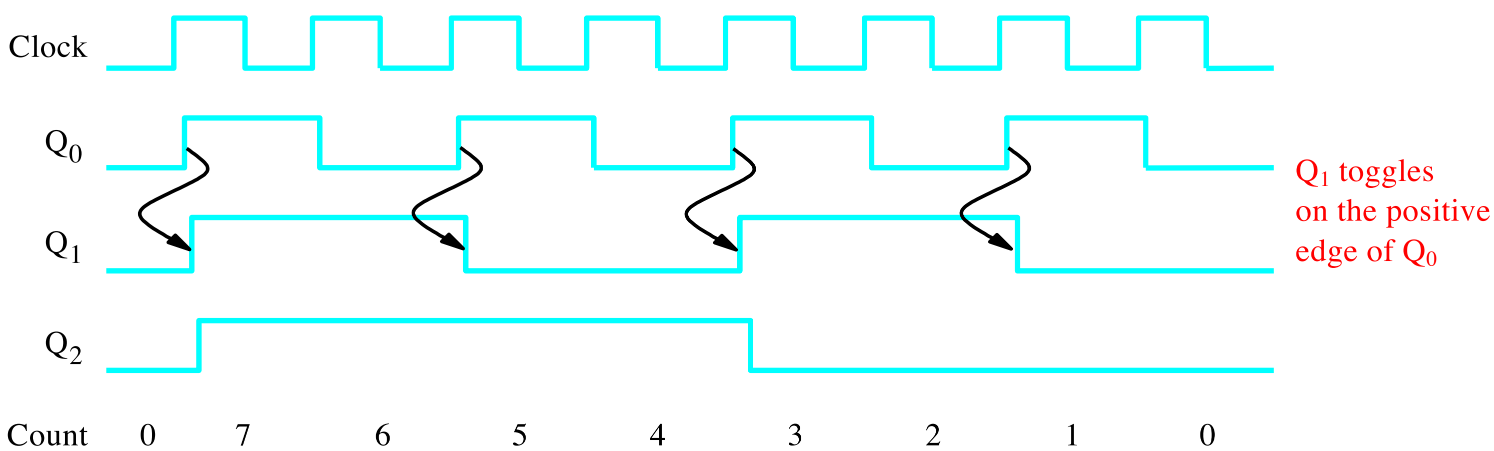


(b) Timing diagram

A three-bit down-counter

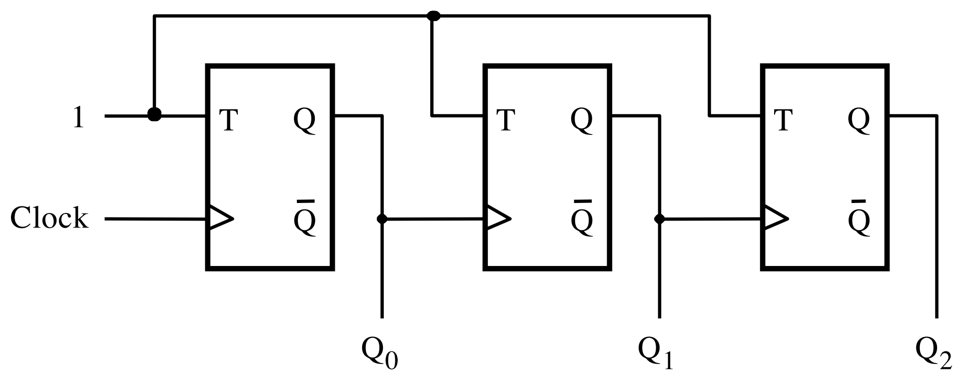


(a) Circuit

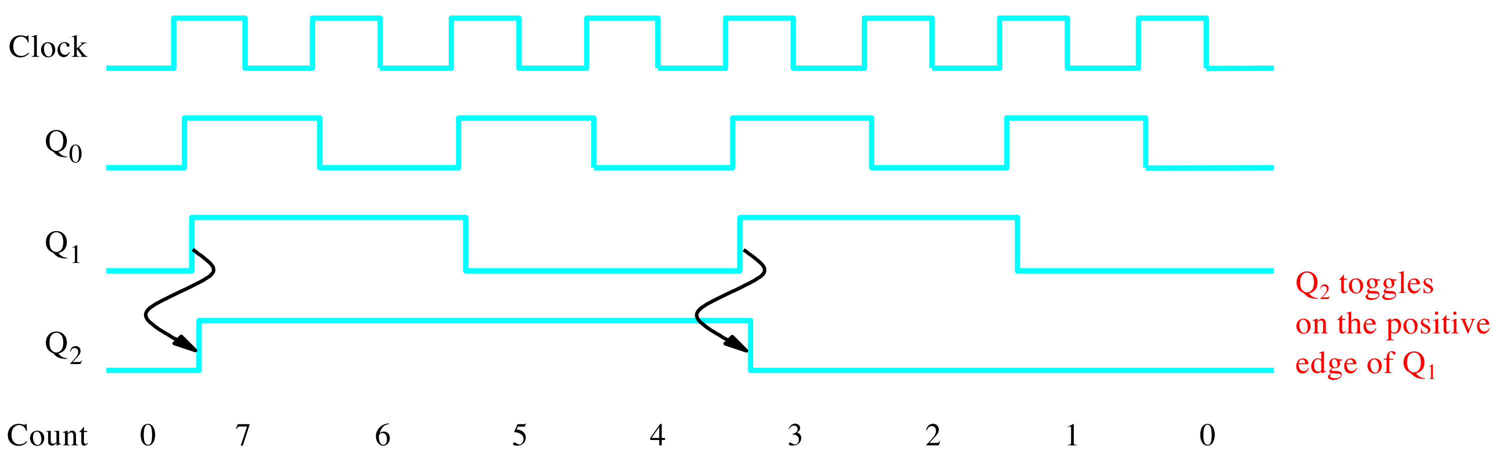


(b) Timing diagram

A three-bit down-counter

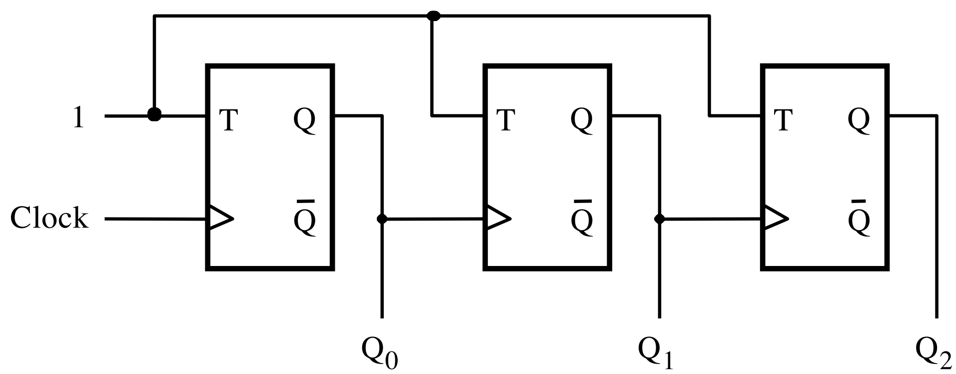


(a) Circuit



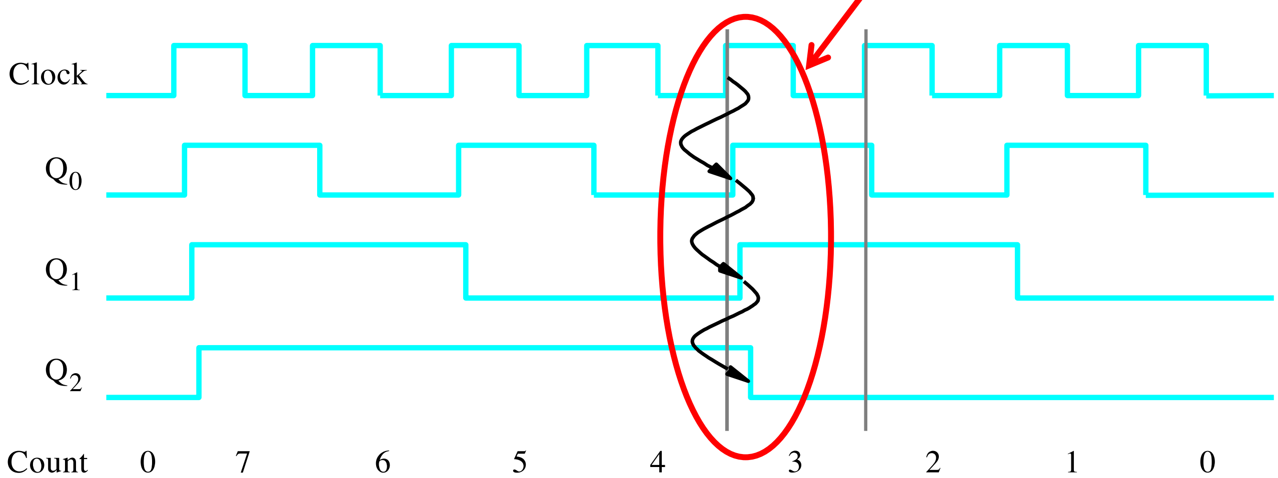
(b) Timing diagram

A three-bit down-counter



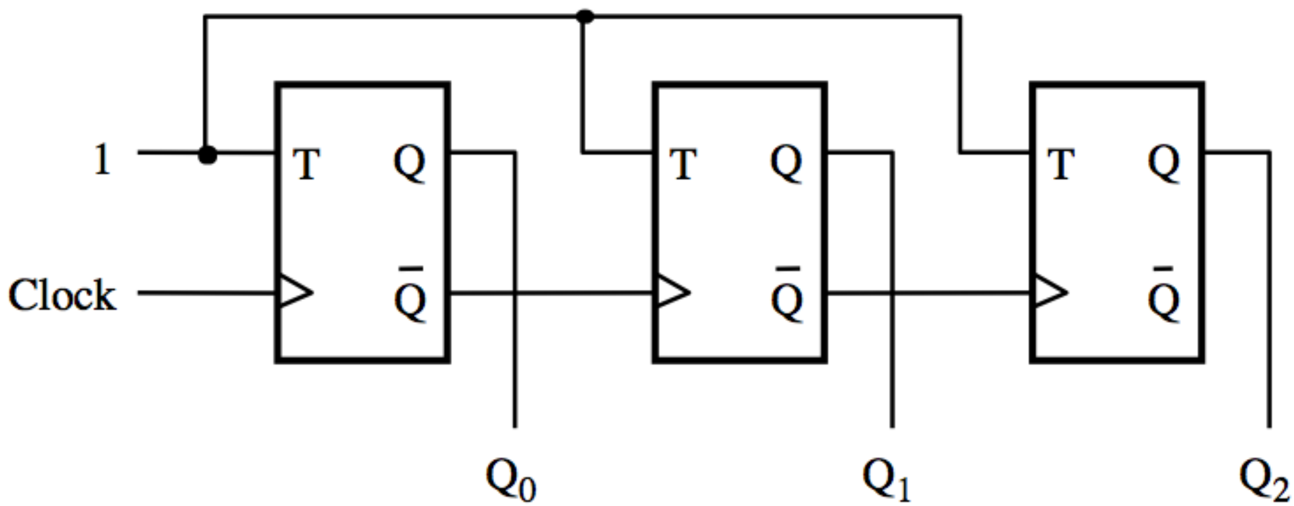
(a) Circuit

The propagation delays get longer



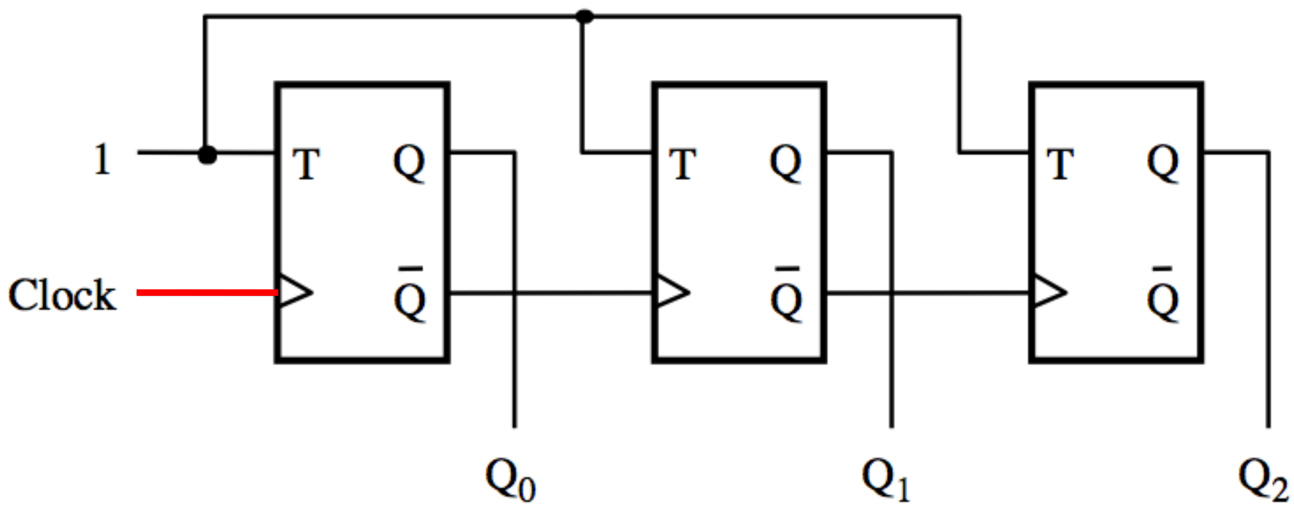
(b) Timing diagram

A three-bit up-counter



[Figure 5.19 from the textbook]

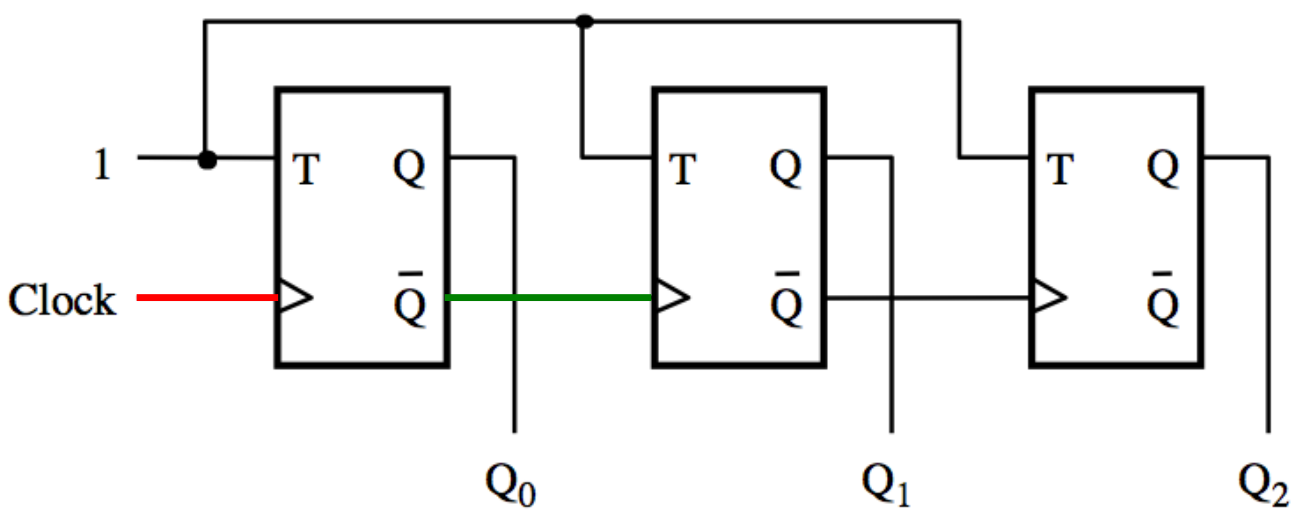
A three-bit up-counter



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

[Figure 5.19 from the textbook]

A three-bit up-counter

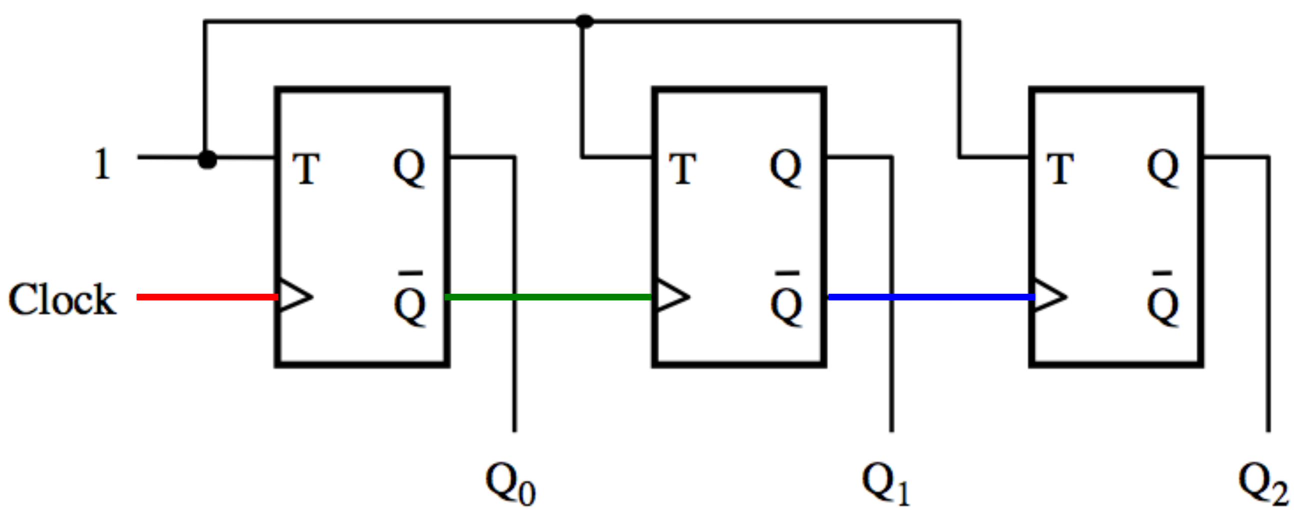


The first flip-flop changes
on the positive edge of the clock

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

[Figure 5.19 from the textbook]

A three-bit up-counter



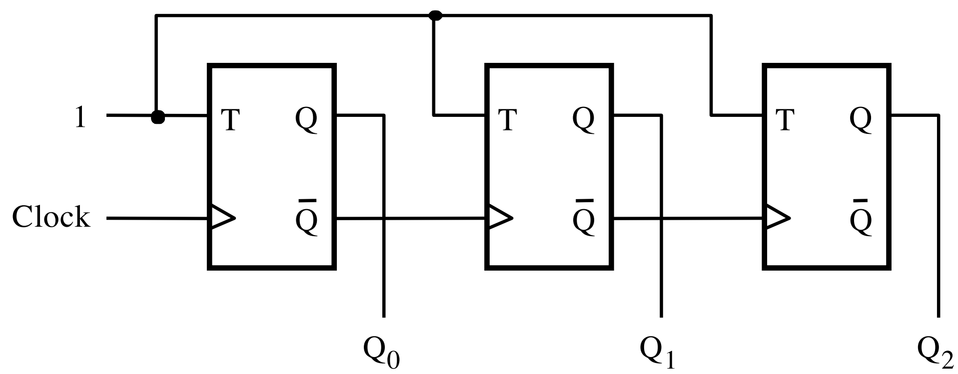
The first flip-flop changes on the positive edge of the clock

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

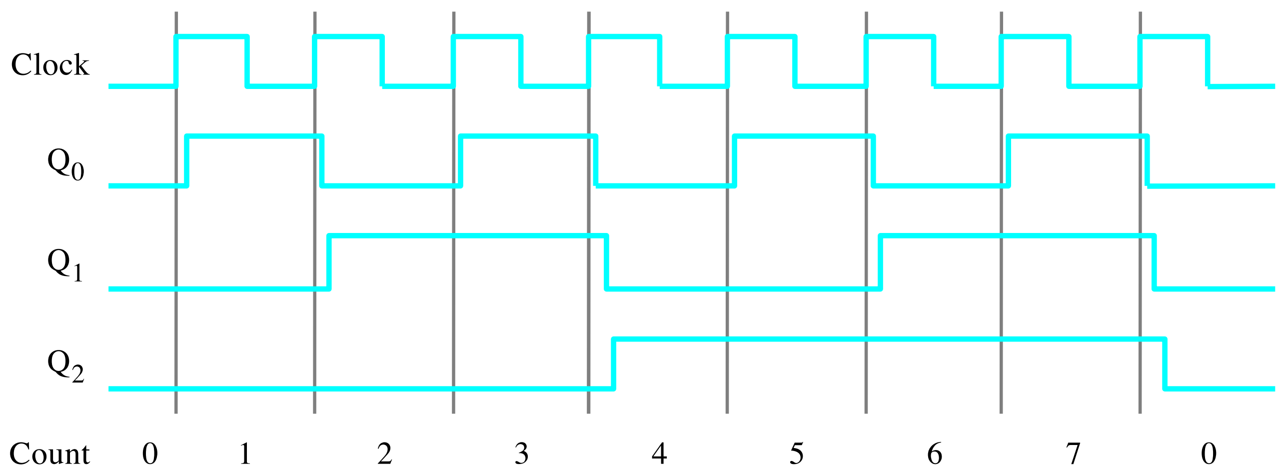
The third flip-flop changes on the positive edge of $\overline{Q}_1$

[Figure 5.19 from the textbook]

A three-bit up-counter



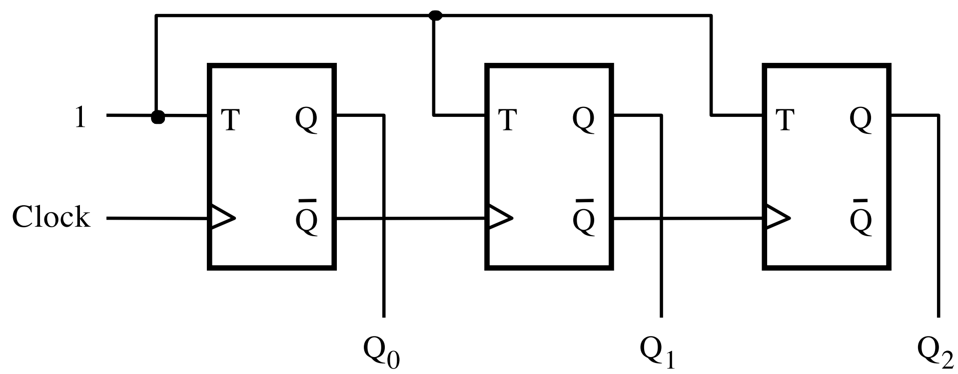
(a) Circuit



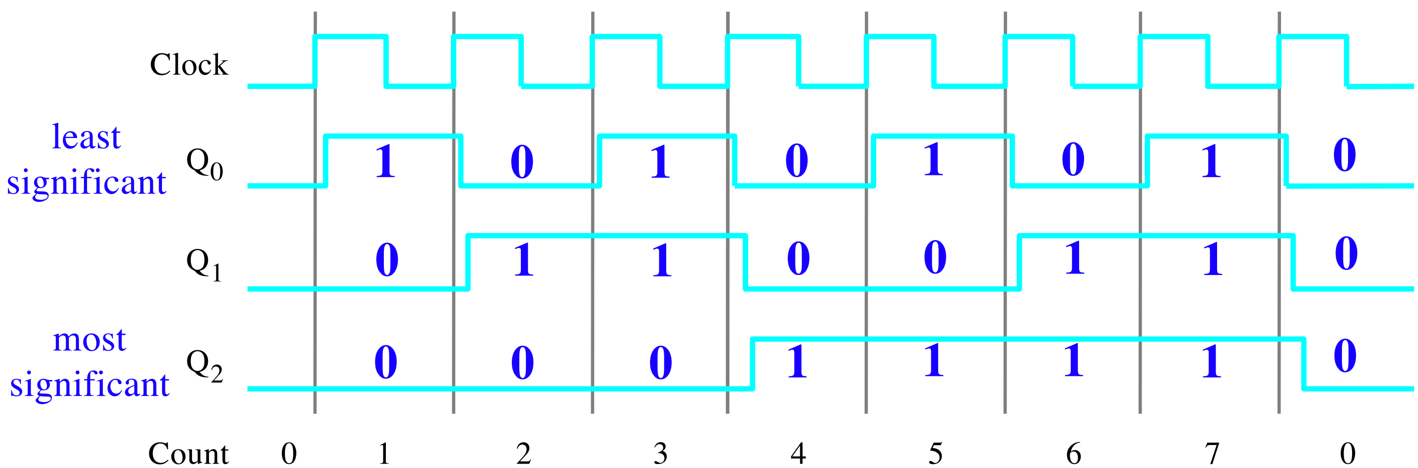
(b) Timing diagram

[Figure 5.19 from the textbook]

A three-bit up-counter



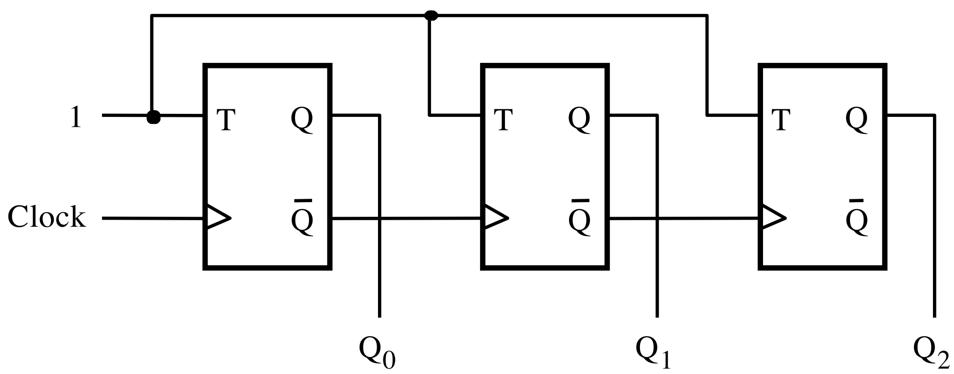
(a) Circuit



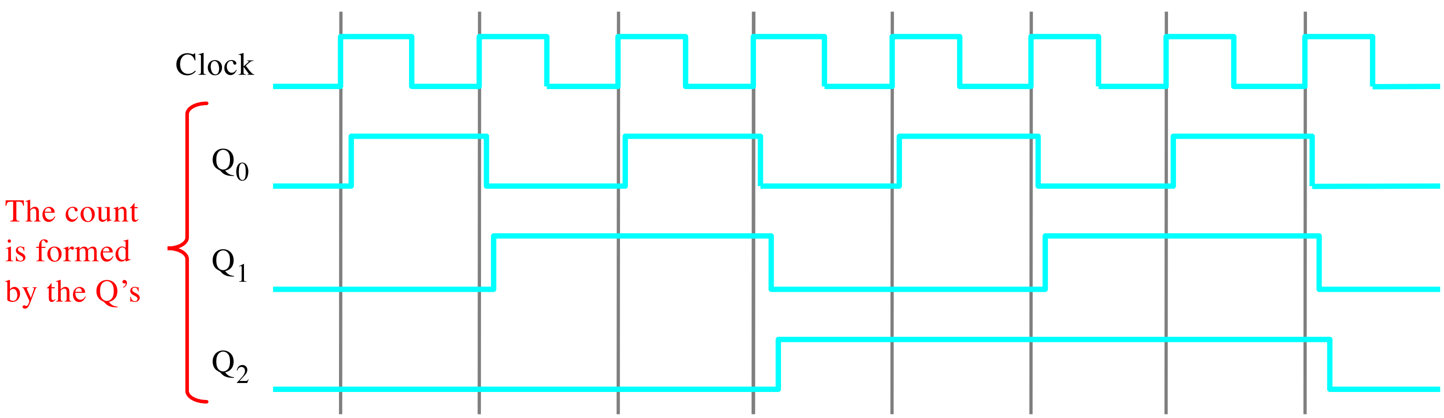
(b) Timing diagram

[Figure 5.19 from the textbook]

A three-bit up-counter



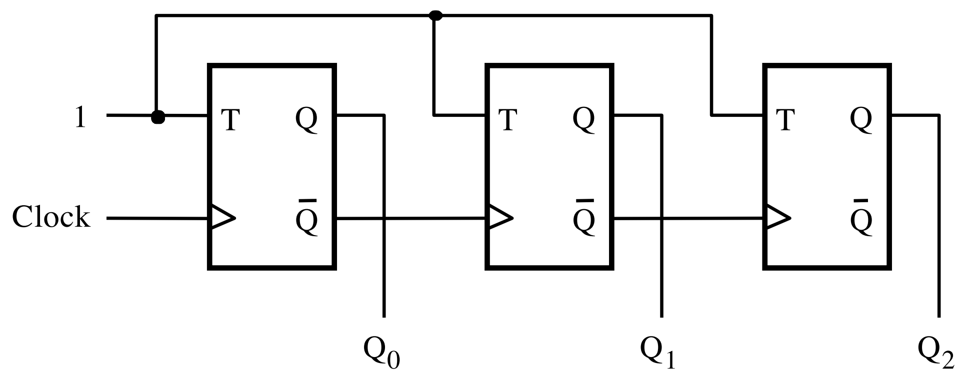
(a) Circuit



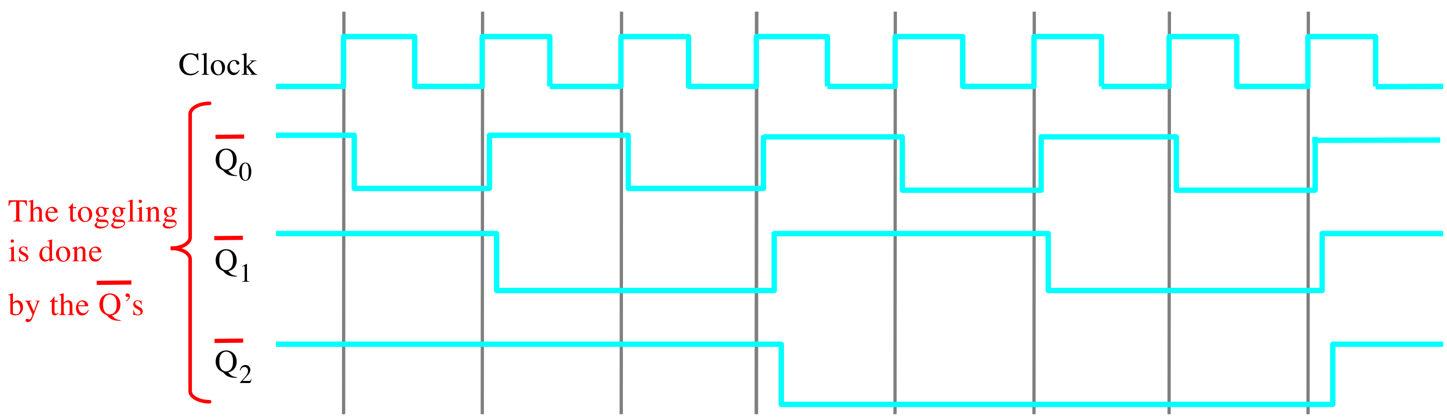
(b) Timing diagram

[Figure 5.19 from the textbook]

A three-bit up-counter

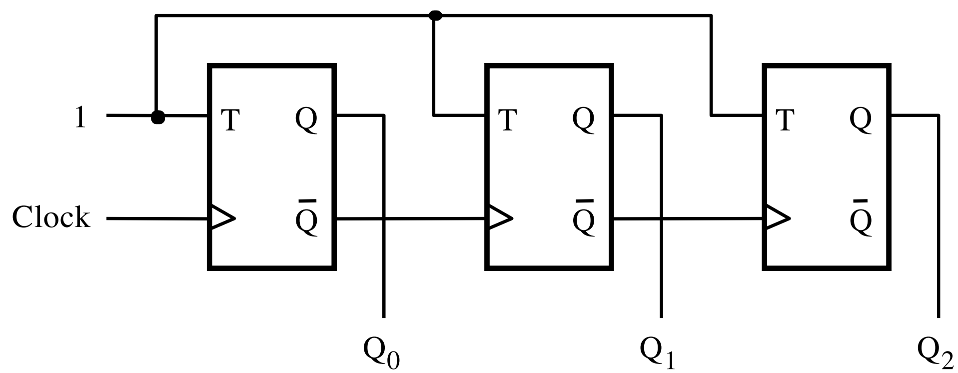


(a) Circuit

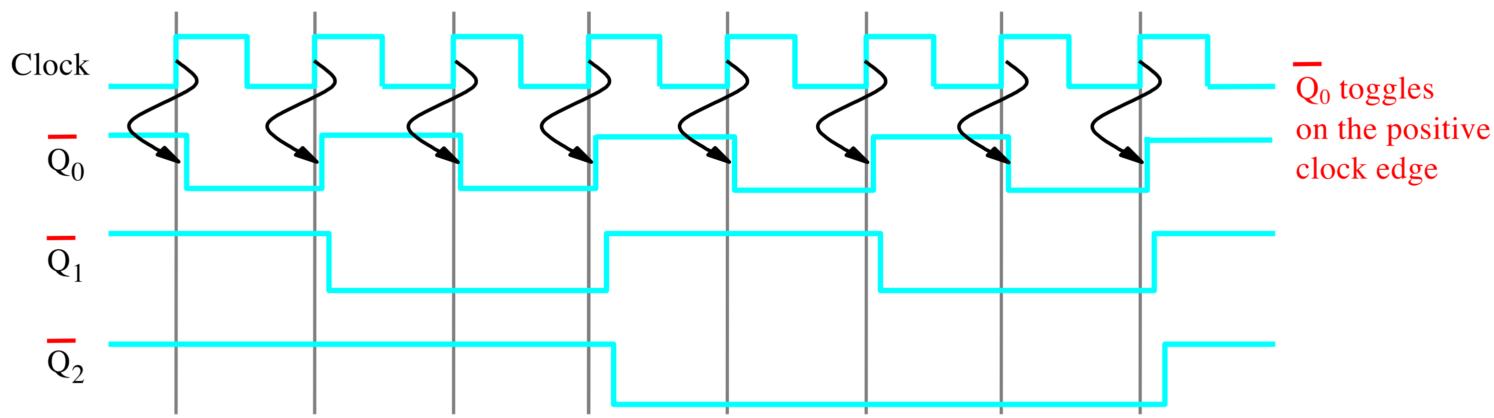


(b) Timing diagram

A three-bit up-counter

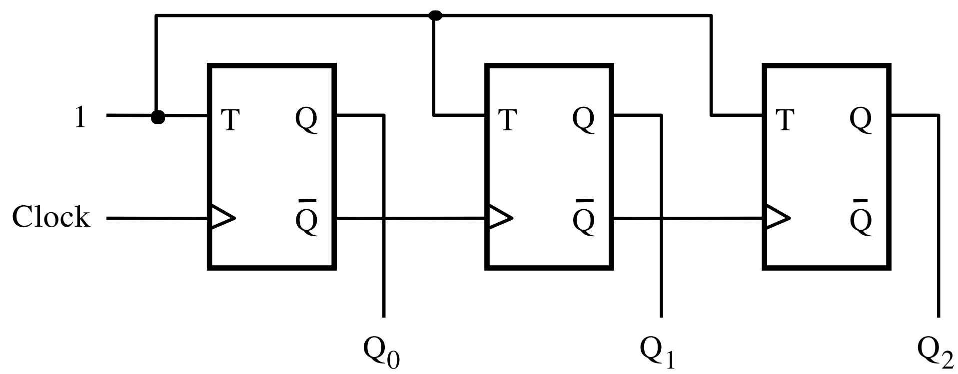


(a) Circuit

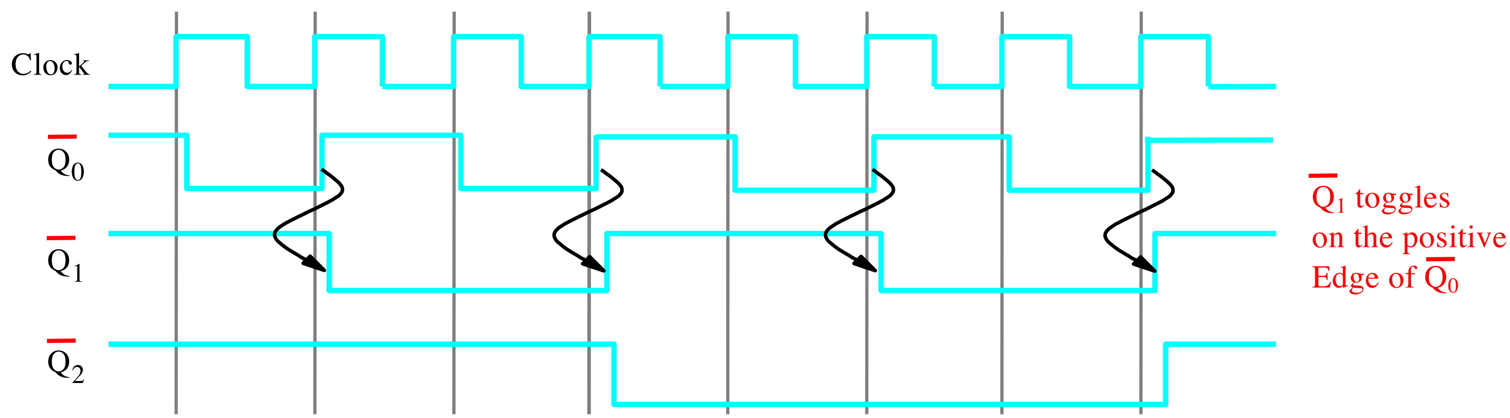


(b) Timing diagram

A three-bit up-counter

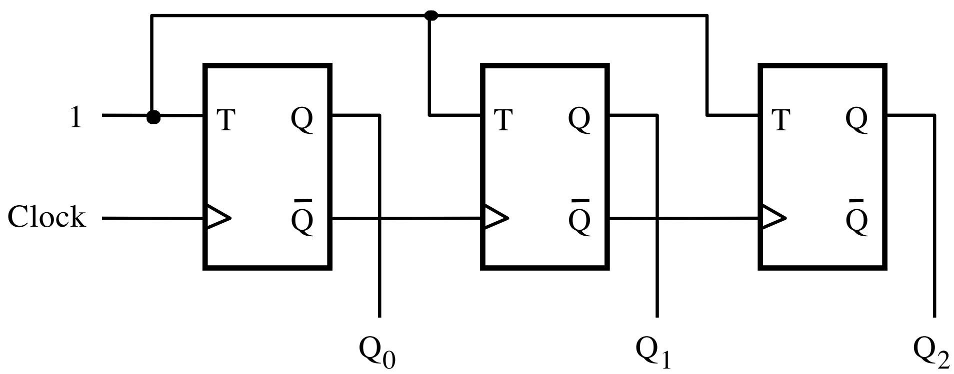


(a) Circuit

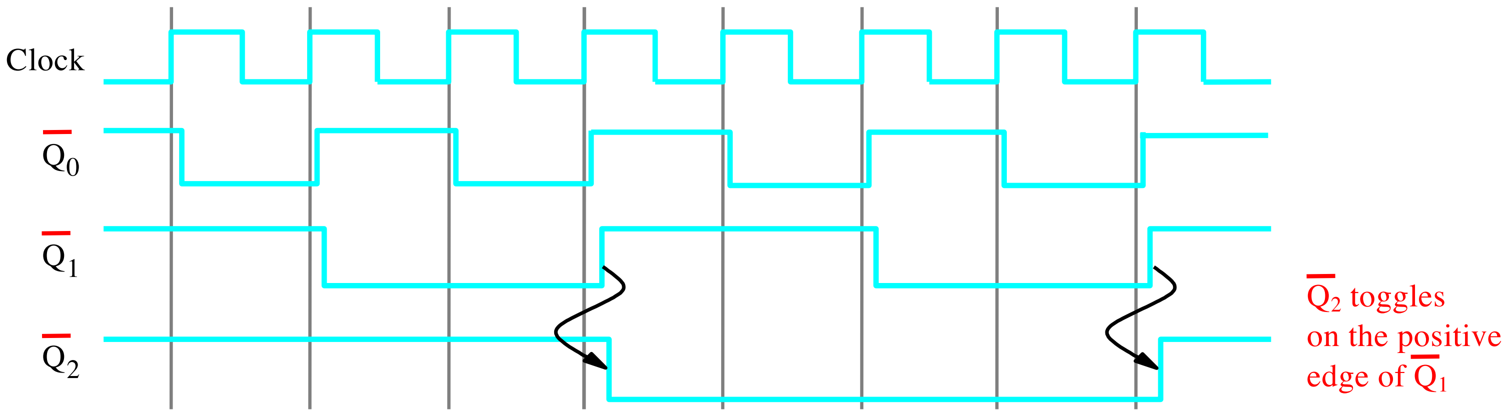


(b) Timing diagram

A three-bit up-counter

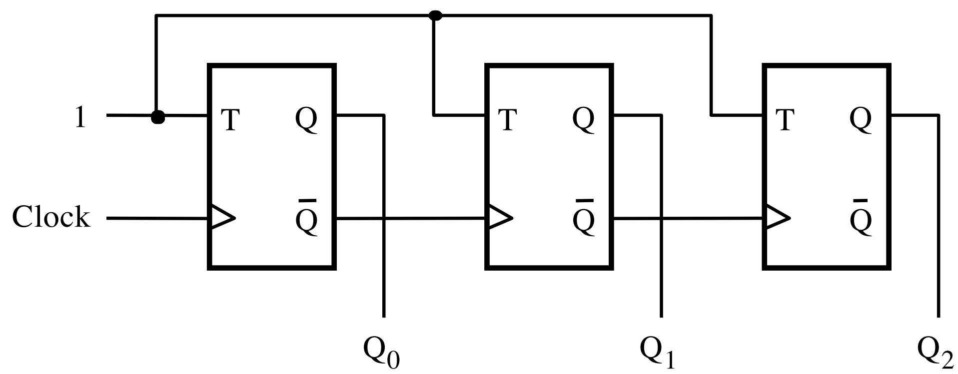


(a) Circuit



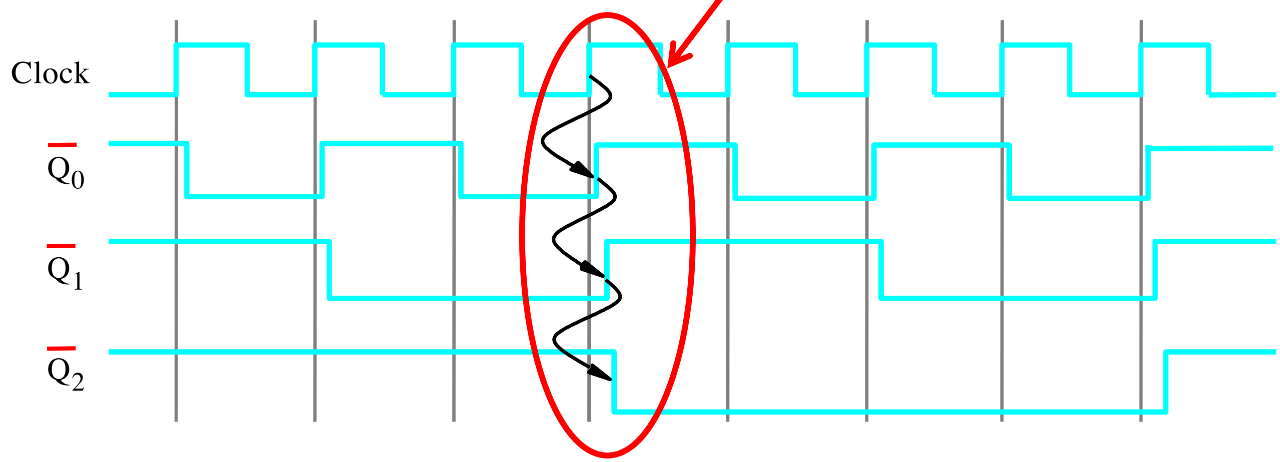
(b) Timing diagram

A three-bit up-counter



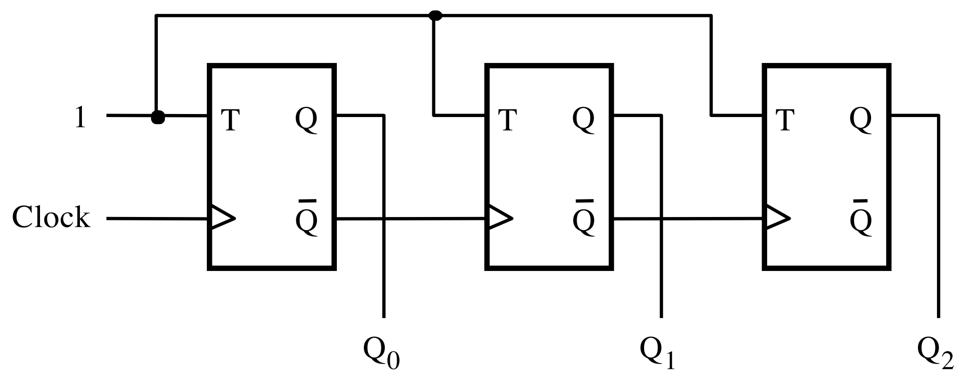
(a) Circuit

The propagation delays get longer



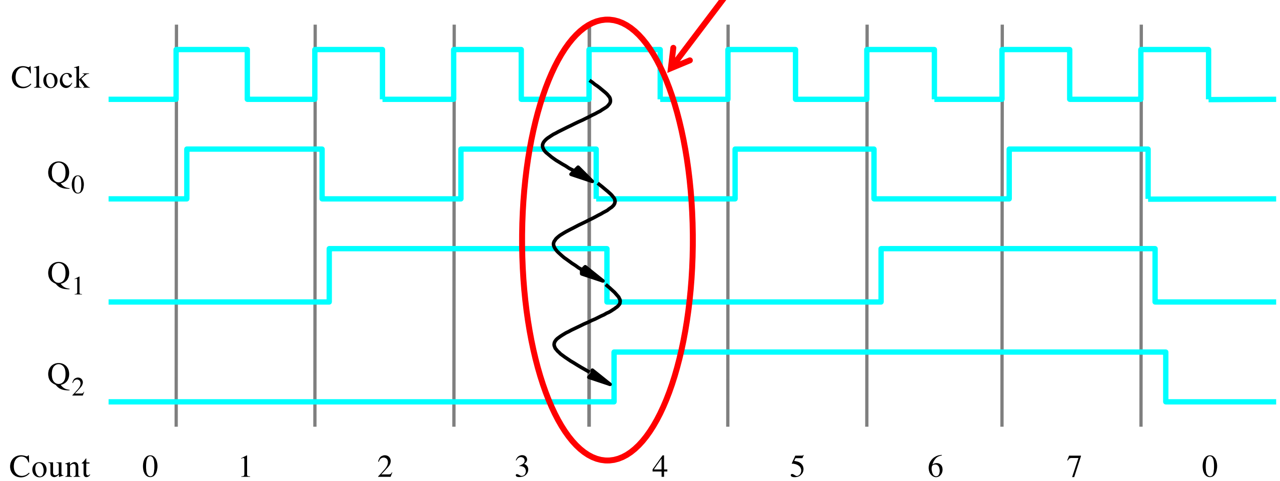
(b) Timing diagram

A three-bit up-counter



(a) Circuit

The propagation delays get longer

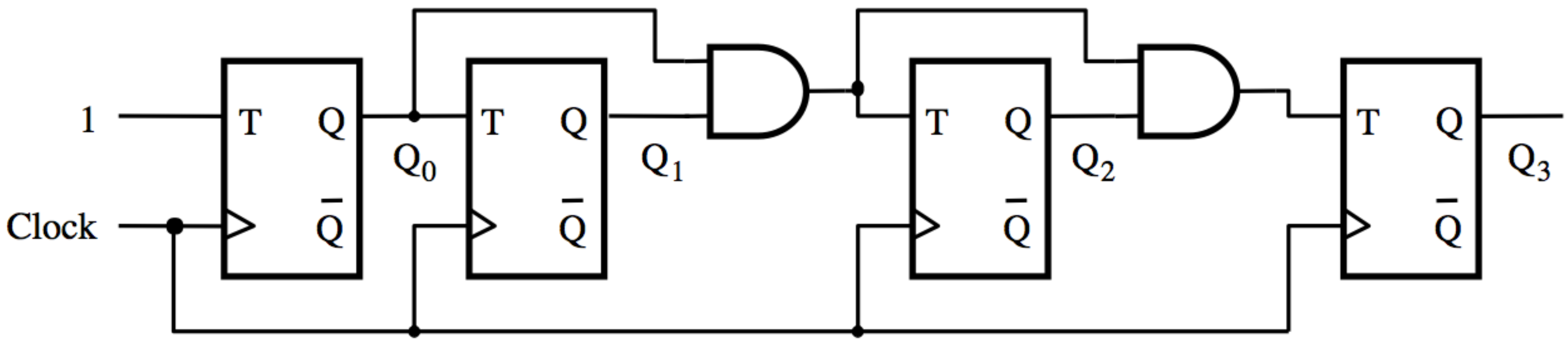


(b) Timing diagram

[Figure 5.19 from the textbook]

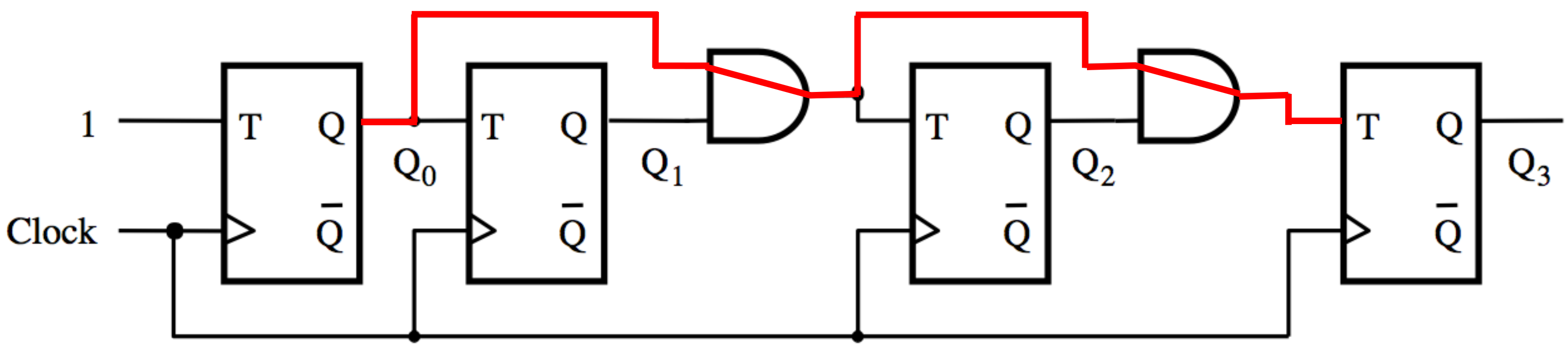
Synchronous Counters

A four-bit synchronous up-counter



[Figure 5.21 from the textbook]

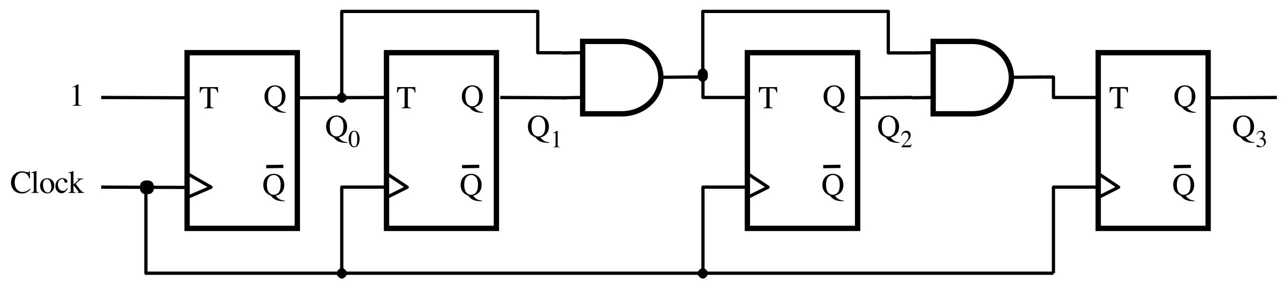
A four-bit synchronous up-counter



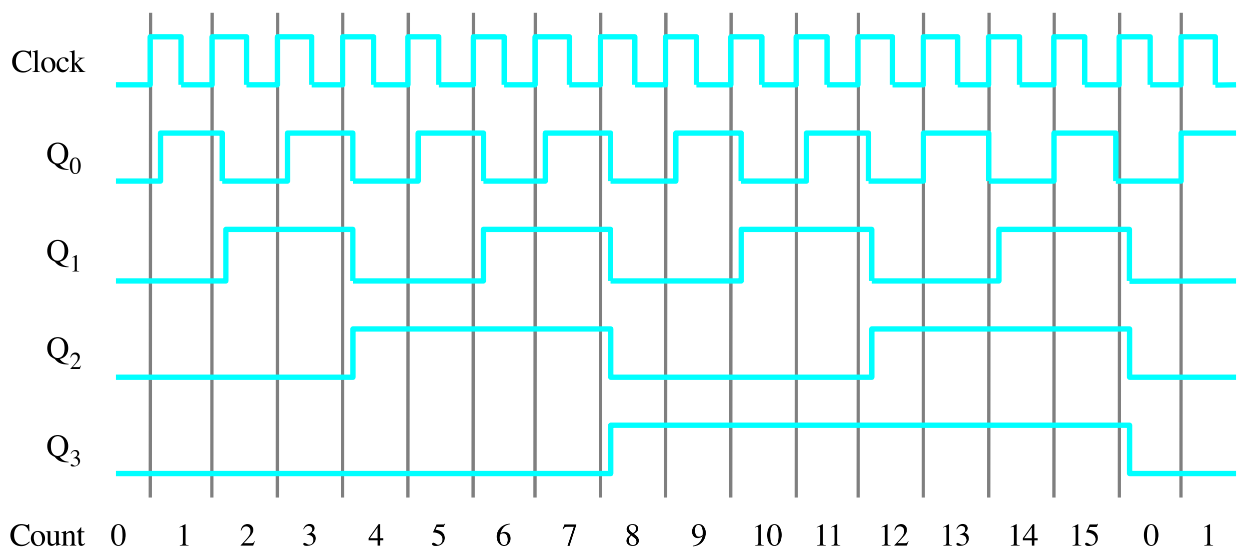
The propagation delay through all AND gates combined must not exceed the clock period minus the setup time for the flip-flops

[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

Q_1 changes

Q_2 changes

[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

Q_1 changes

Q_2 changes

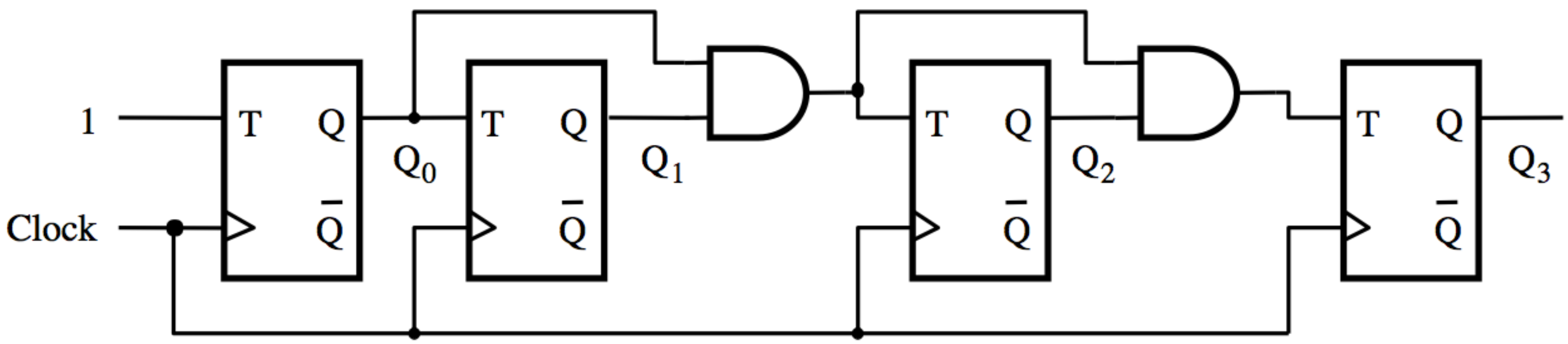
$T_0 = 1$

$T_1 = Q_0$

$T_2 = Q_0 Q_1$

[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

$T_0 = 1$

$T_1 = Q_0$

$T_2 = Q_0\ Q_1$

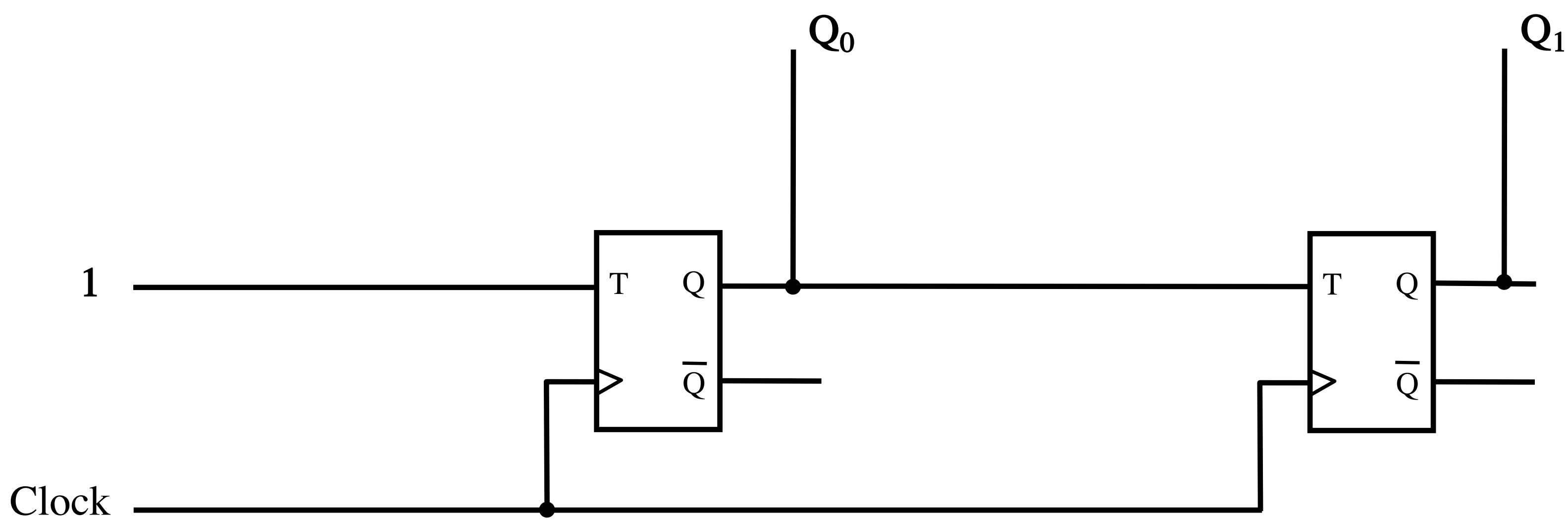
$T_3 = Q_0\ Q_1\ Q_2$

$\ldots$

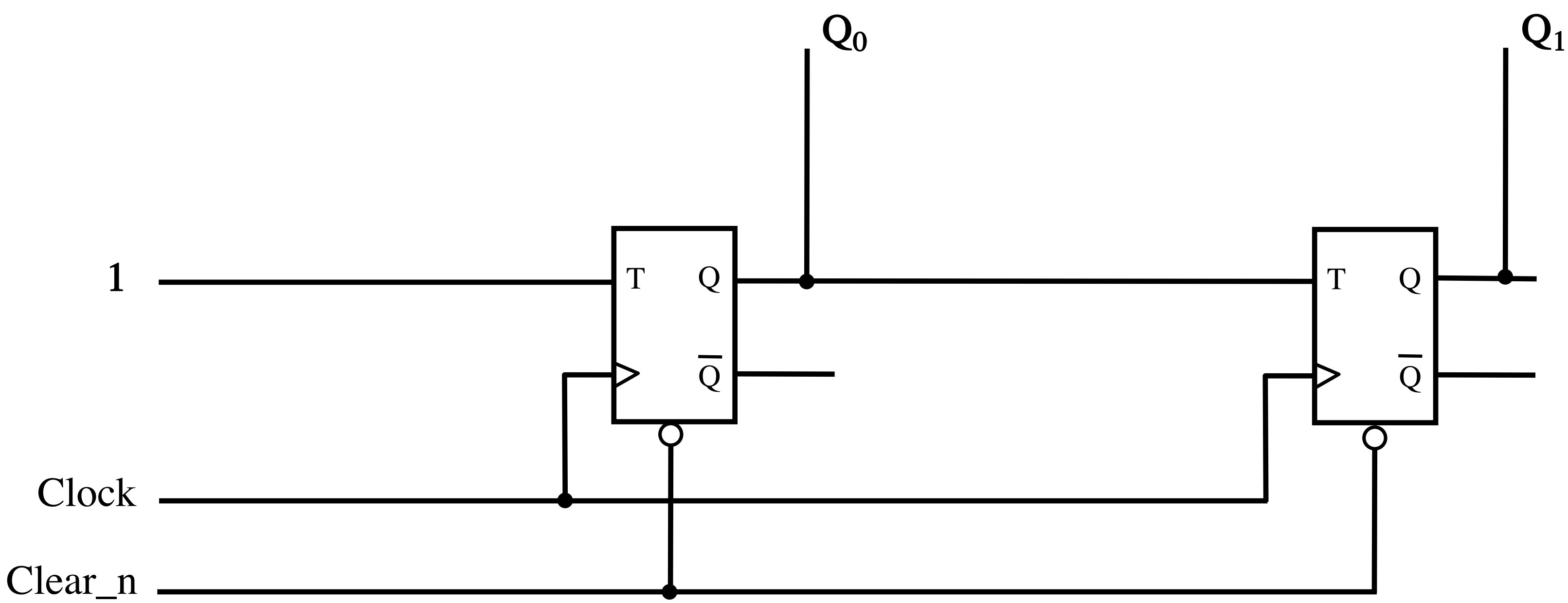
$T_n = Q_0\ Q_1\ Q_2\ \ldots Q_{n-1}$

Synchronous v.s. Asynchronous Clear

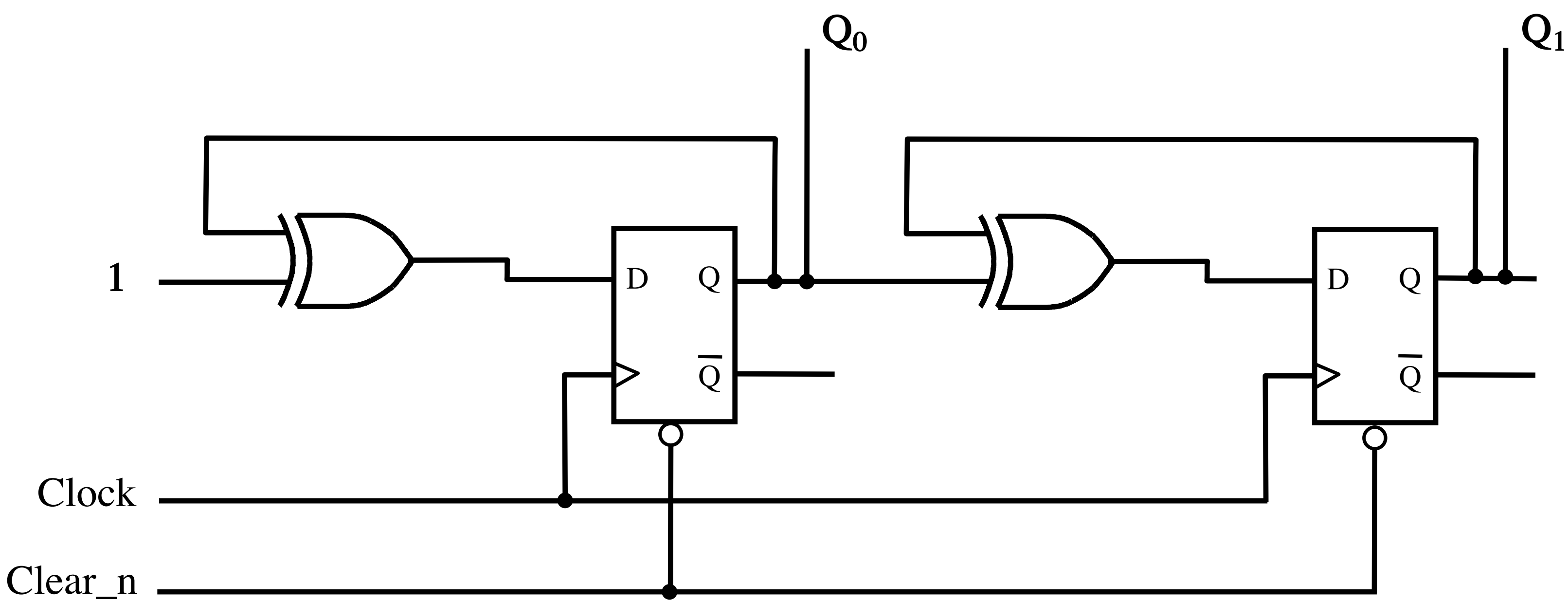
2-Bit Synchronous Up-Counter (without clear capability)



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

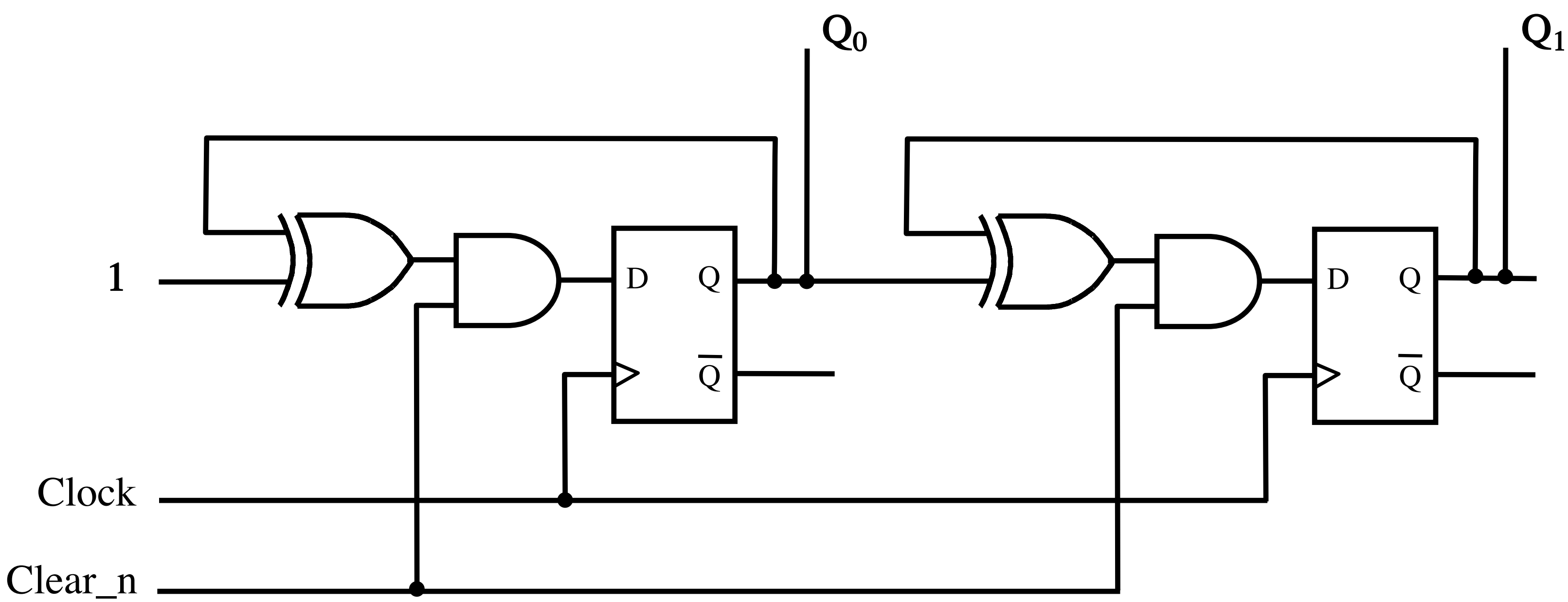


2-Bit Synchronous Up-Counter (with asynchronous clear)



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

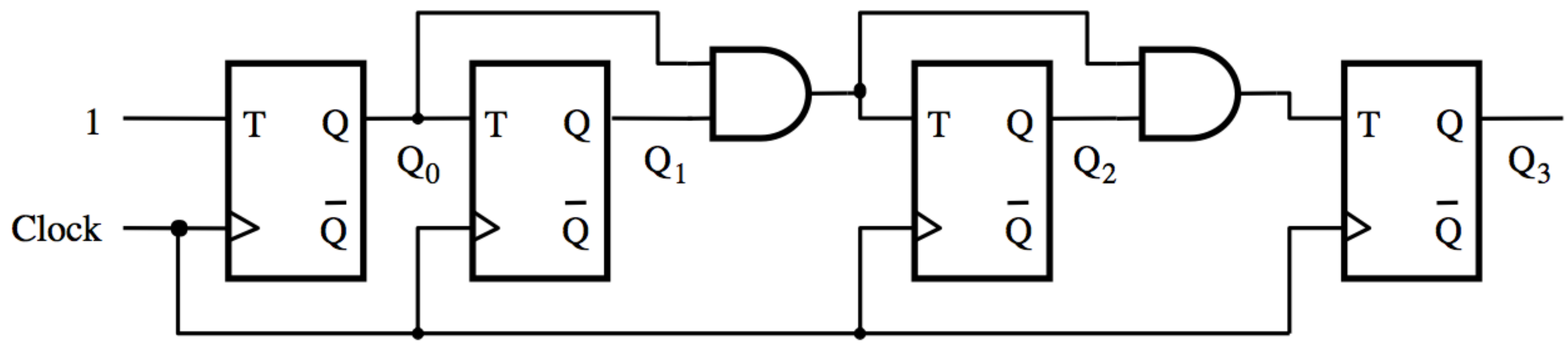
2-Bit Synchronous Up-Counter (with synchronous clear)



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

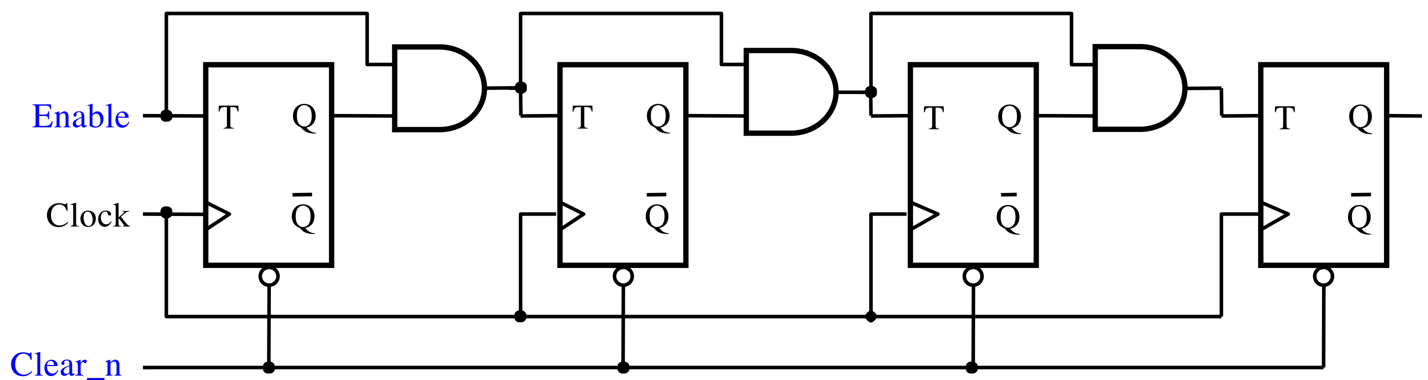
Adding Enable Capability

A four-bit synchronous up-counter



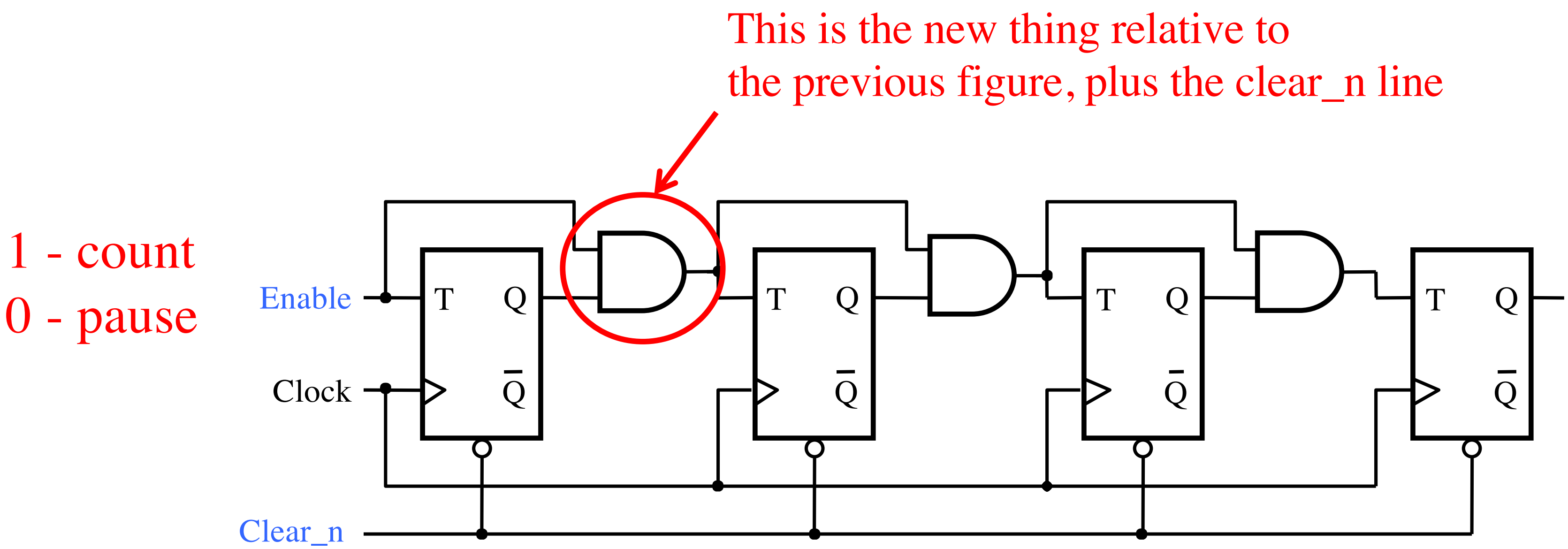
[Figure 5.21 from the textbook]

Inclusion of Enable and Clear Capability



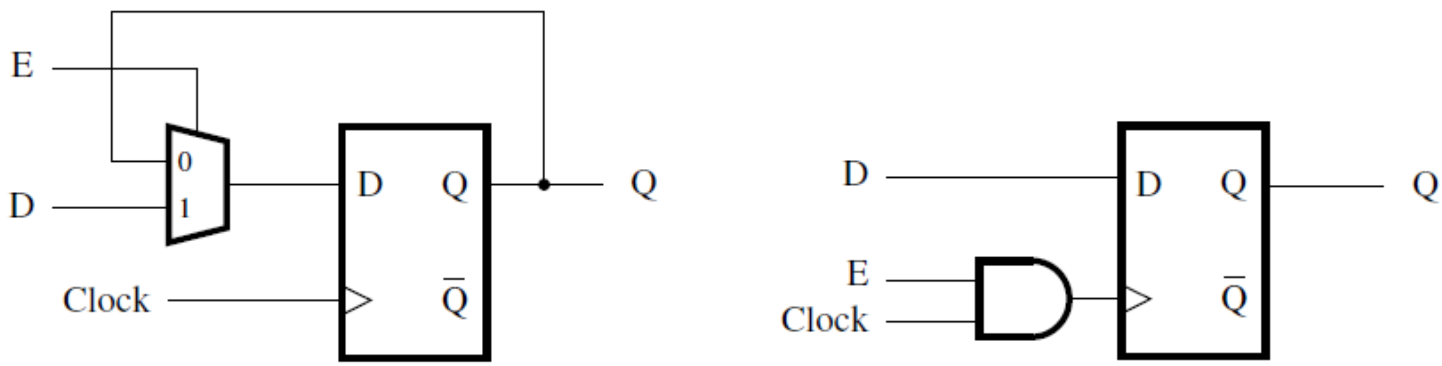
[Figure 5.22 from the textbook]

Inclusion of Enable and Clear Capability



[Figure 5.22 from the textbook]

Providing an enable input for a D flip-flop



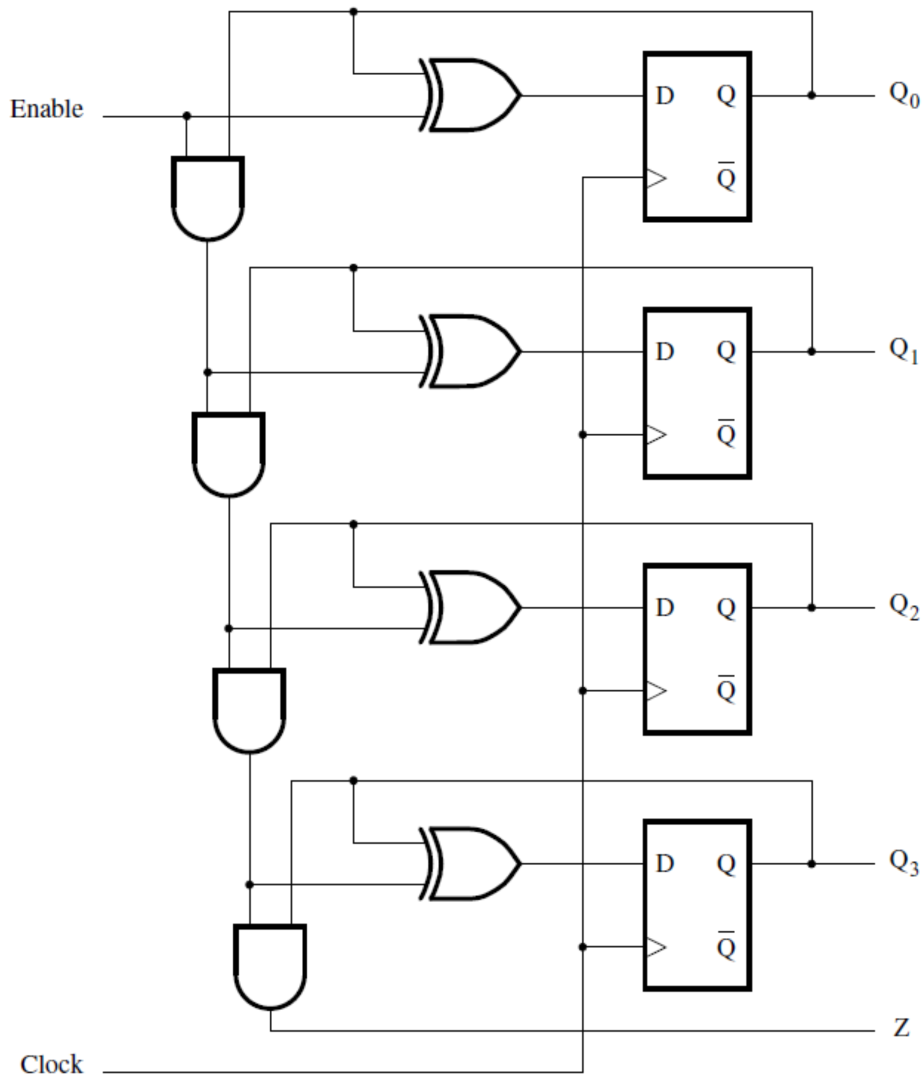
(a) Using a multiplexer

(b) Clock gating

[Figure 5.56 from the textbook]

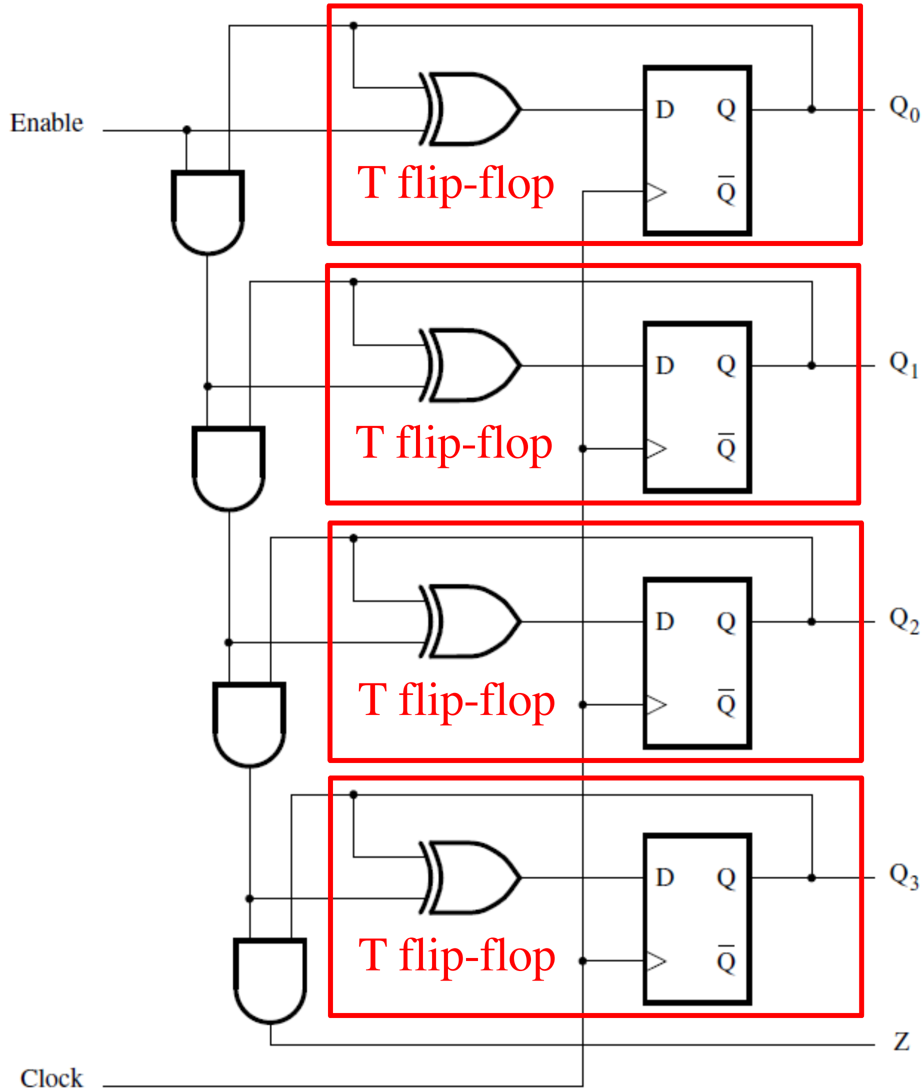
Synchronous Counter
(with D Flip-Flops)

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



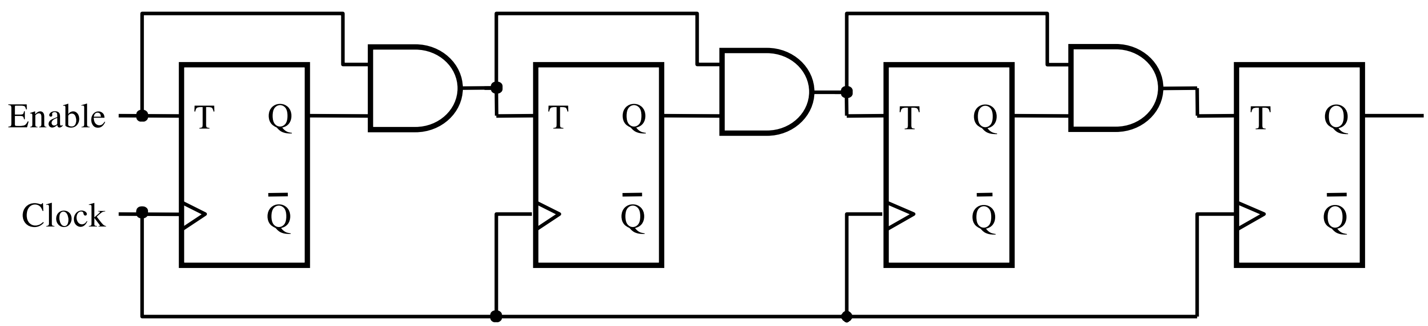
[Figure 5.23 from the textbook]

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

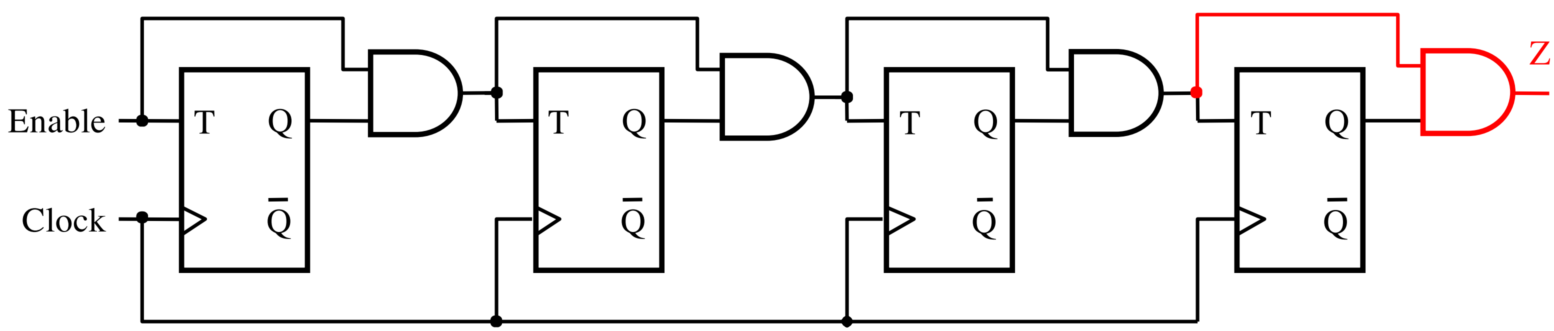


[Figure 5.23 from the textbook]

Equivalent to this circuit with T flip-flops



Equivalent to this circuit with T flip-flops

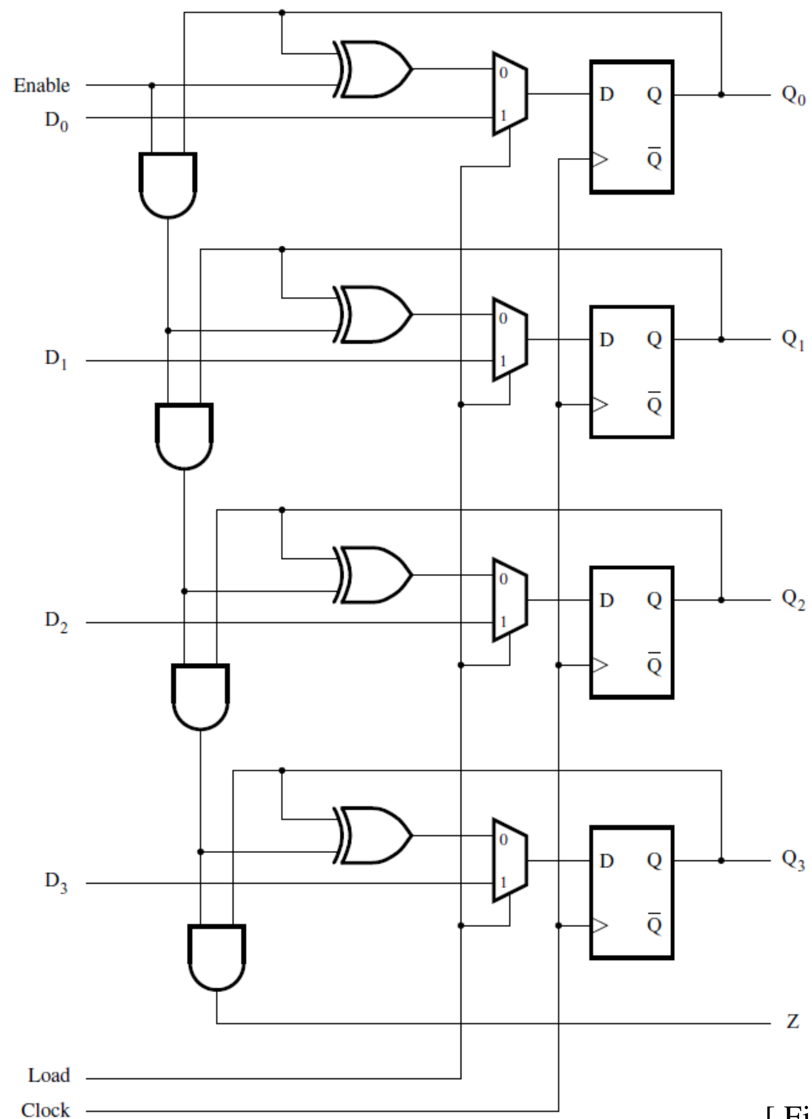


But has one extra output called Z, which can be used to connect two 4-bit counters to make an 8-bit counter.

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).

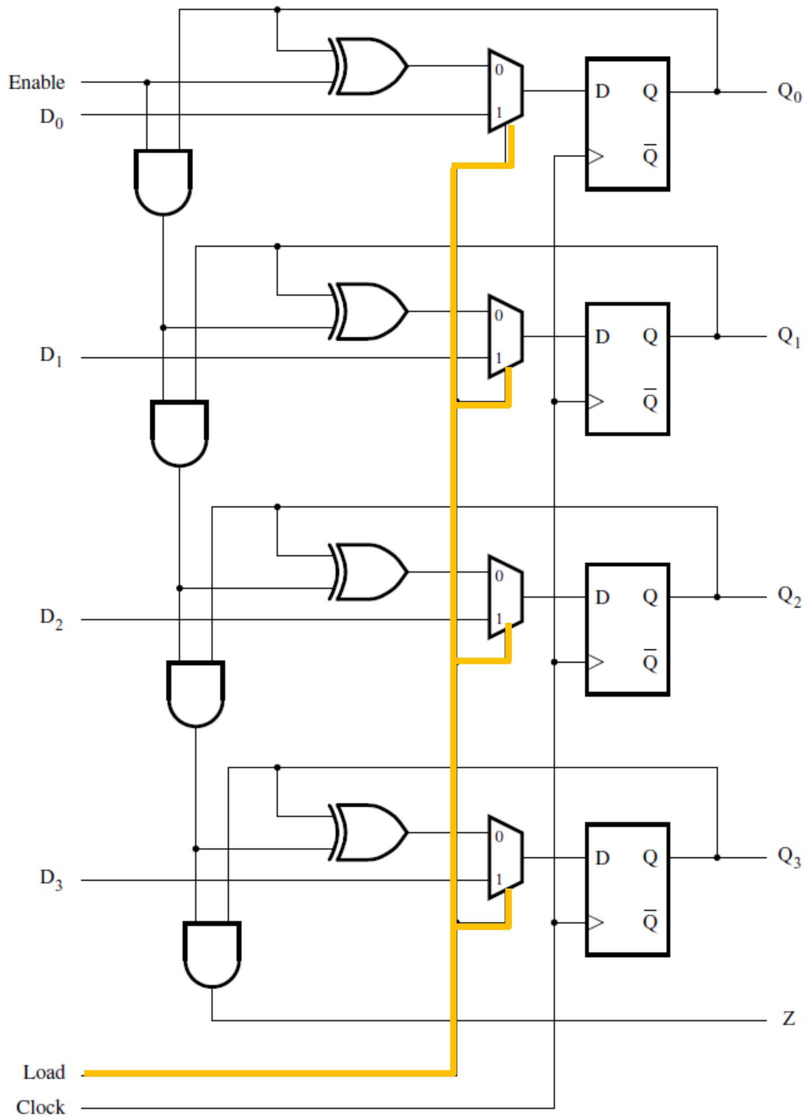
Counters with Parallel Load

A counter with parallel-load capability

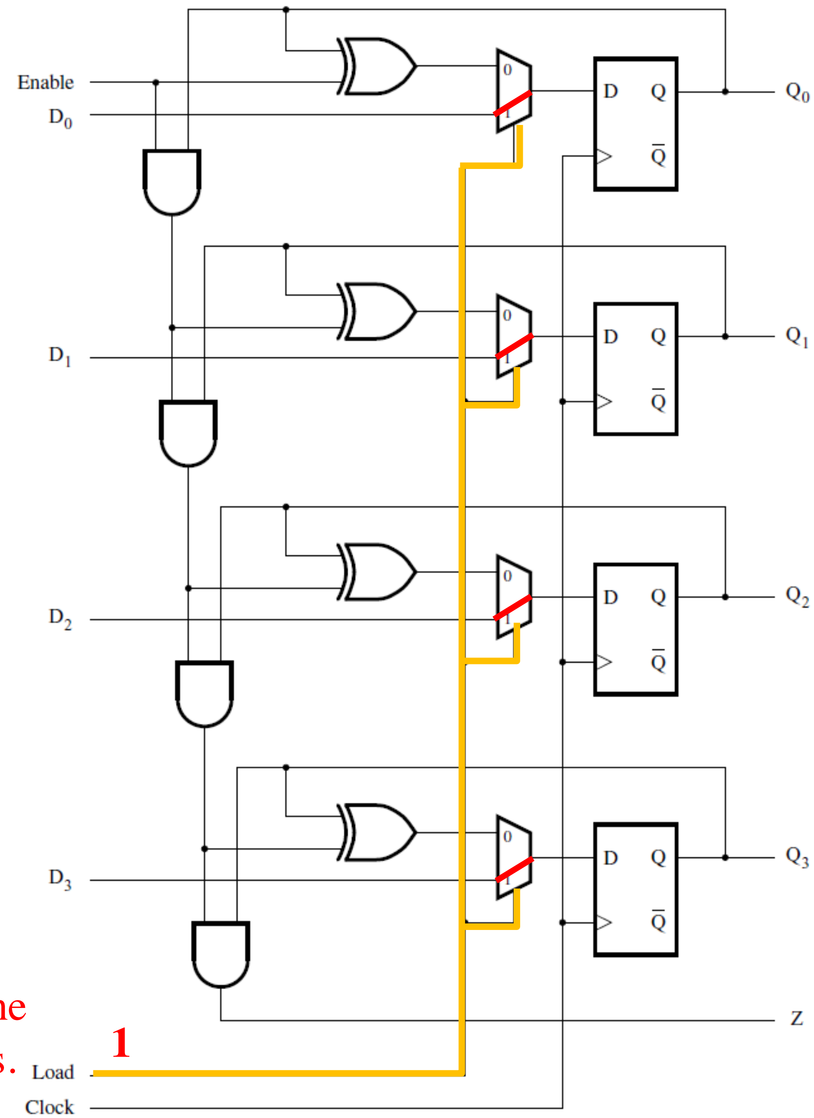


[Figure 5.24 from the textbook]

How to load the initial count value



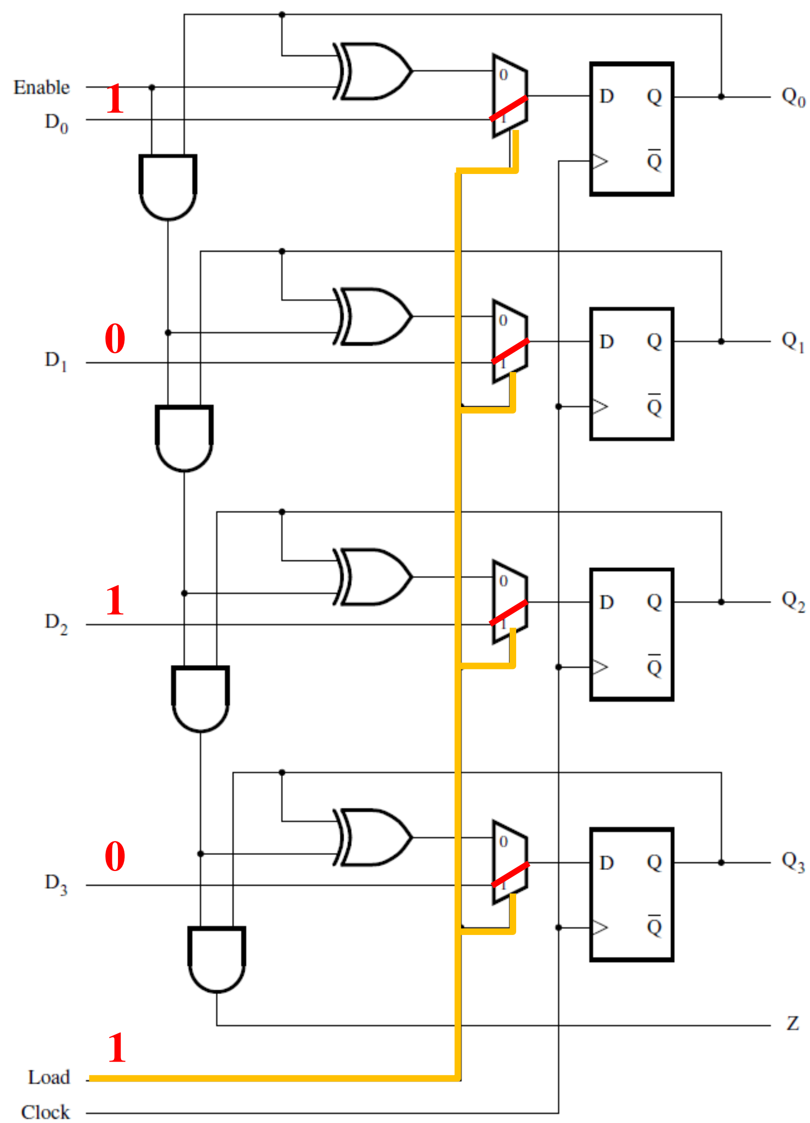
How to load the initial count value



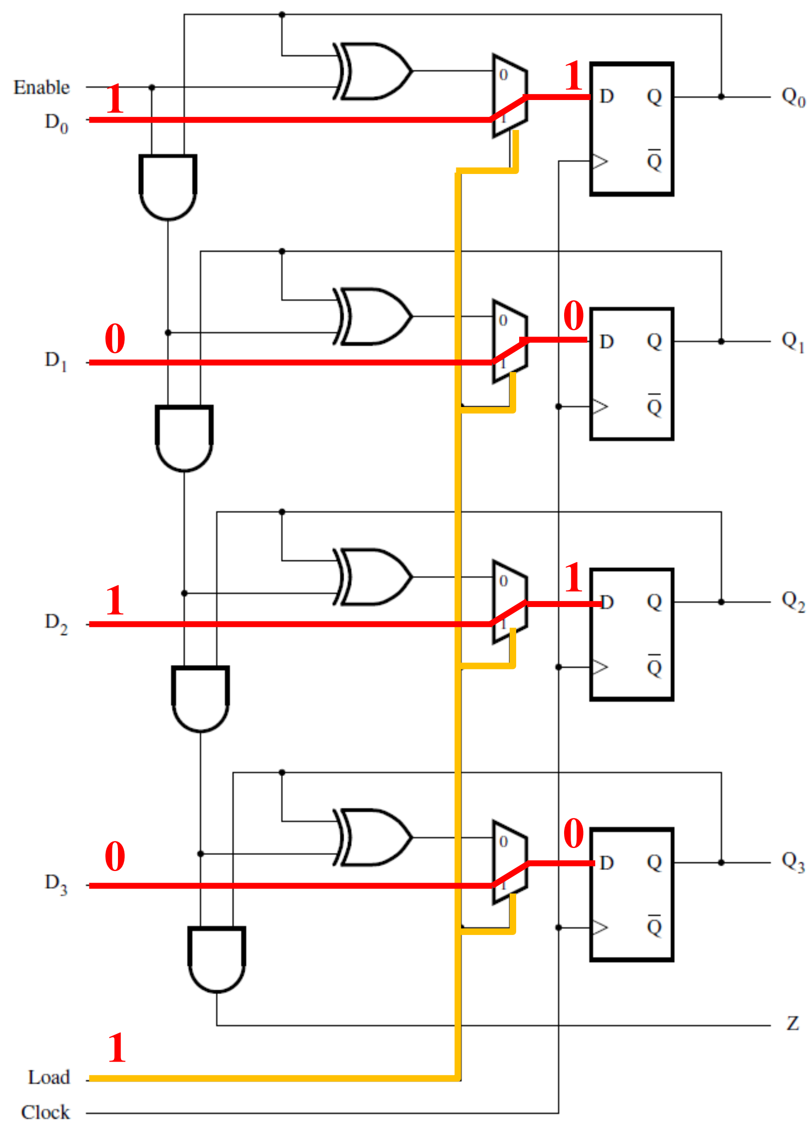
Set "Load" to 1, to open the "1" line of the multiplexers.

How to load the initial count value

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

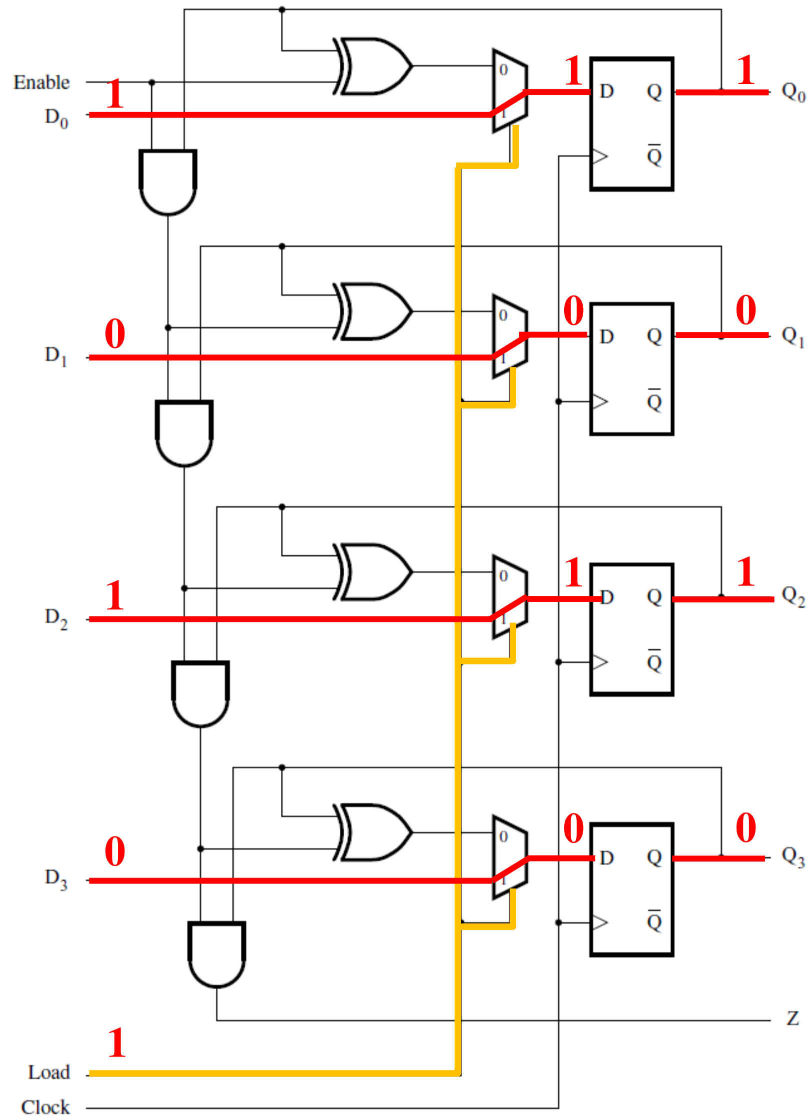


How to load the initial count value



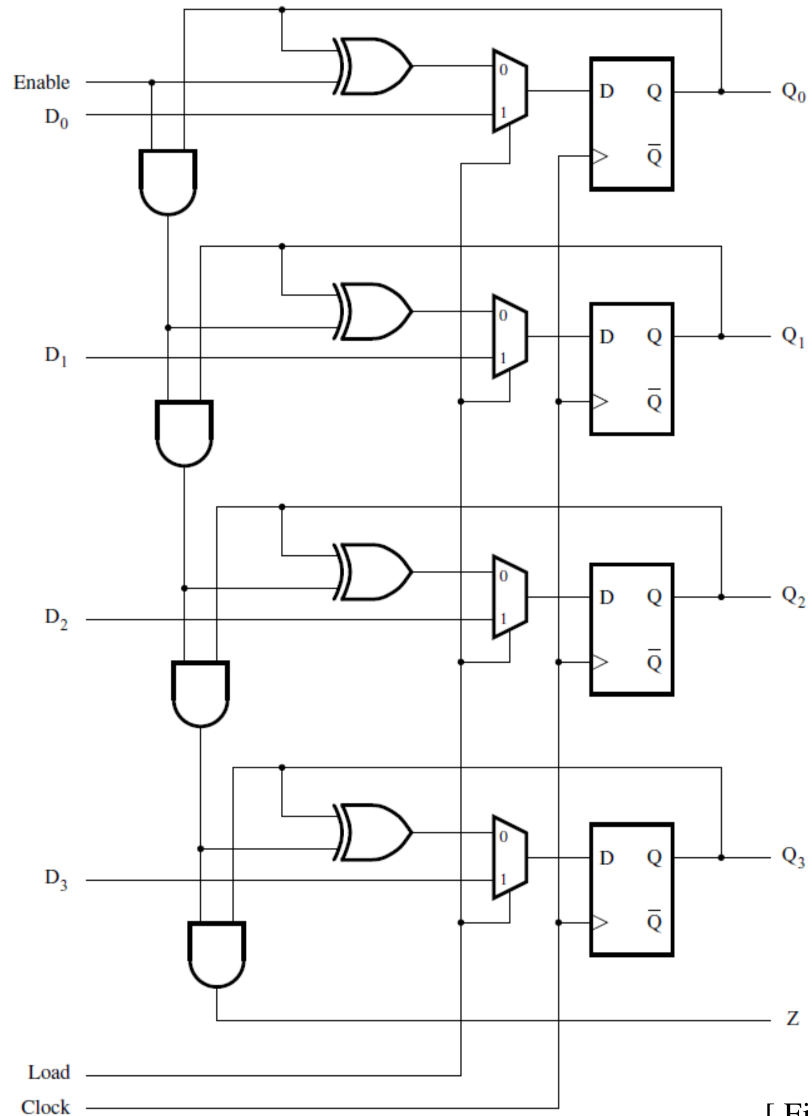
These bits propagate
to the inputs of
the D Flip-Flops

How to load the initial count value



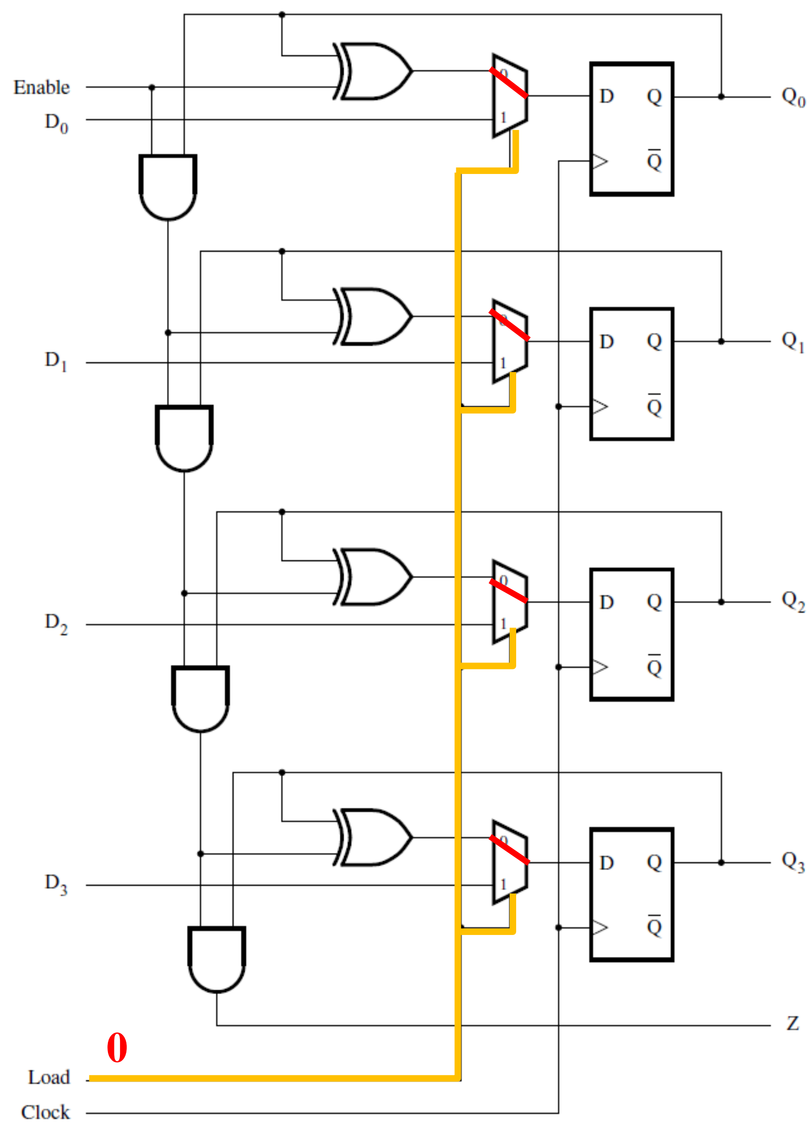
When the next positive edge of the clock arrives, the outputs of the flip-flops are updated.

A counter with parallel-load capability



[Figure 5.24 from the textbook]

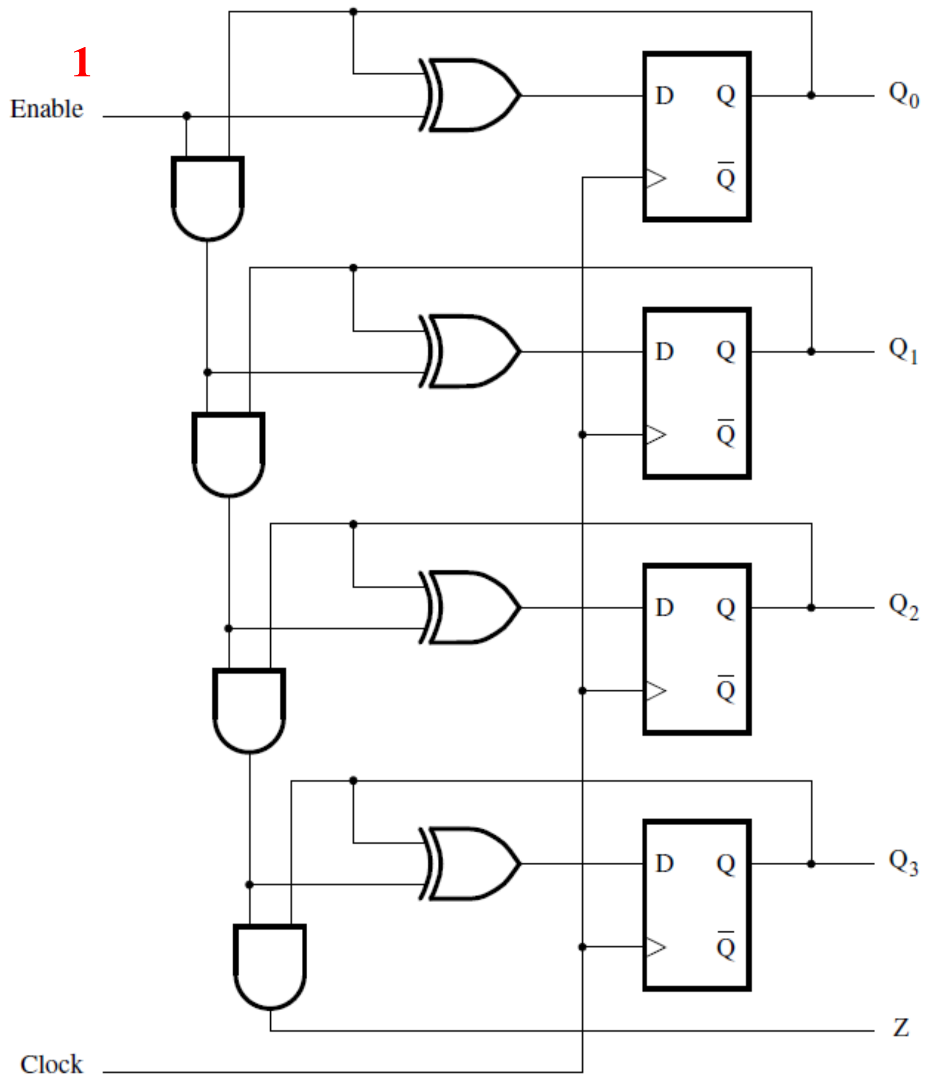
How to run this as a counter



When "Load" is 0 this reduces to the previous counter circuit. As if the MUXes are not there.

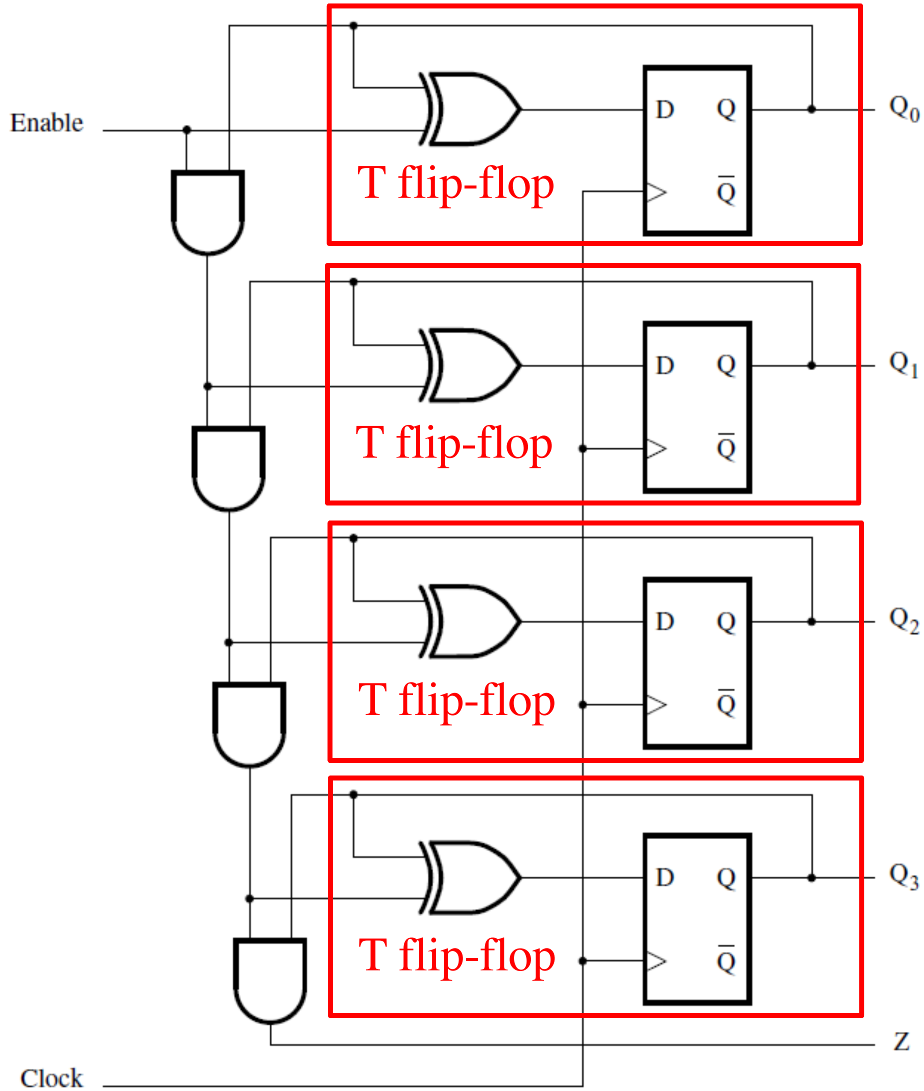
It reduces to this 4-bit up-counter

Make sure to enable it.



[Figure 5.23 from the textbook]

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



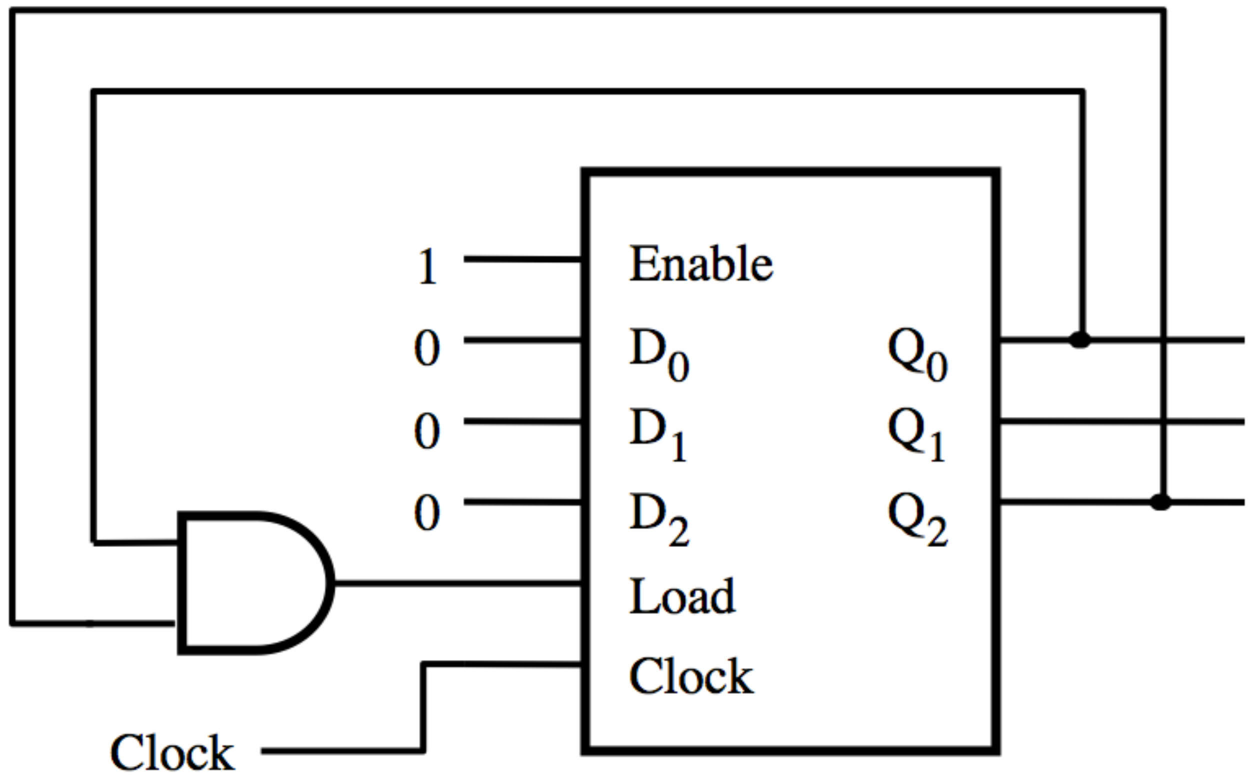
[Figure 5.23 from the textbook]

Reset Synchronization

Motivation

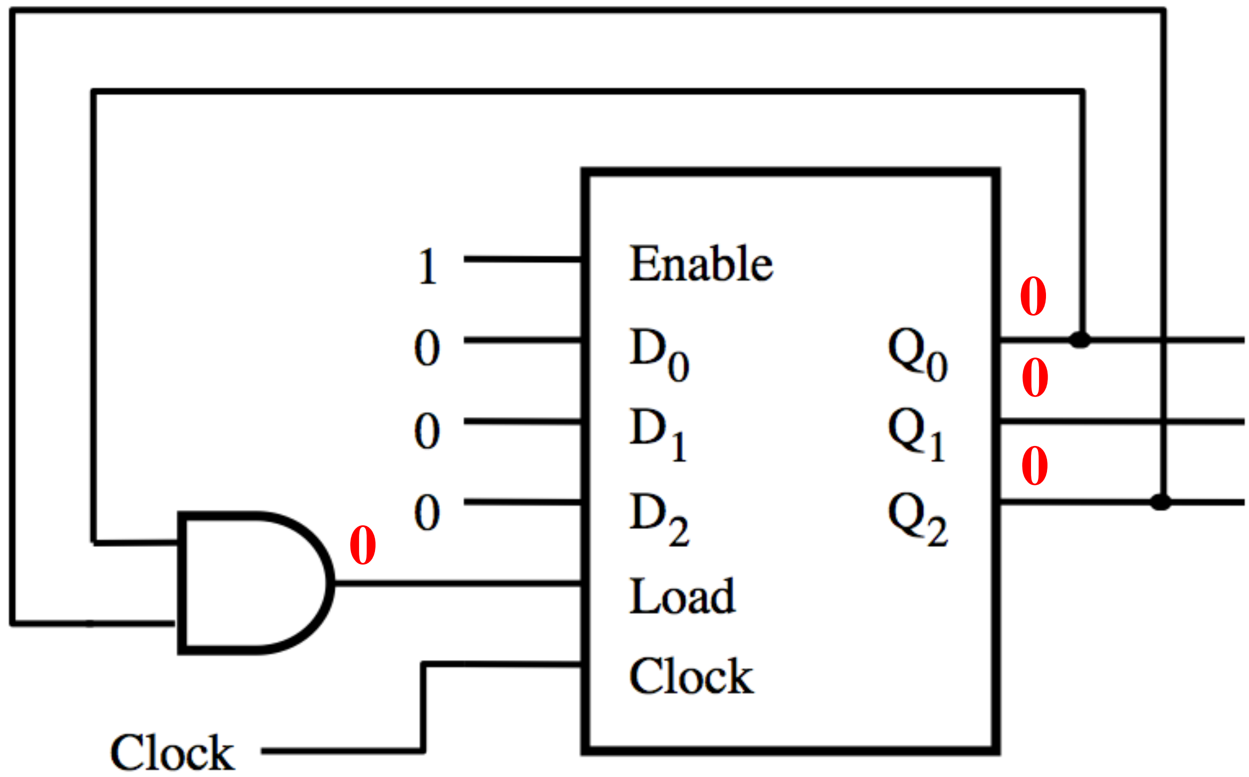
- An n-bit counter counts from 0, 1, …, 2^n-1
- For example a 3-bit counter counts up as follow
  - 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, …
- What if we want it to count like this
  - 0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5, 0, 1, …
- In other words, what is the cycle is not a power of 2?

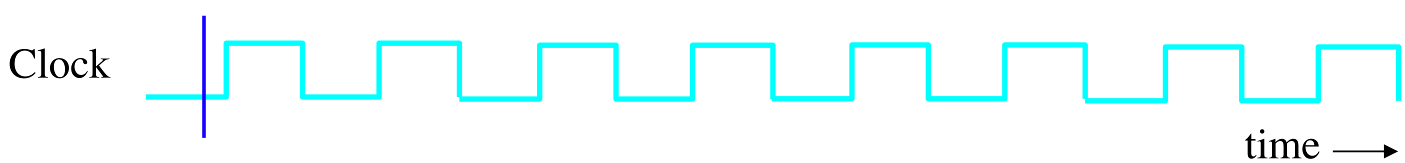
What does this circuit do?



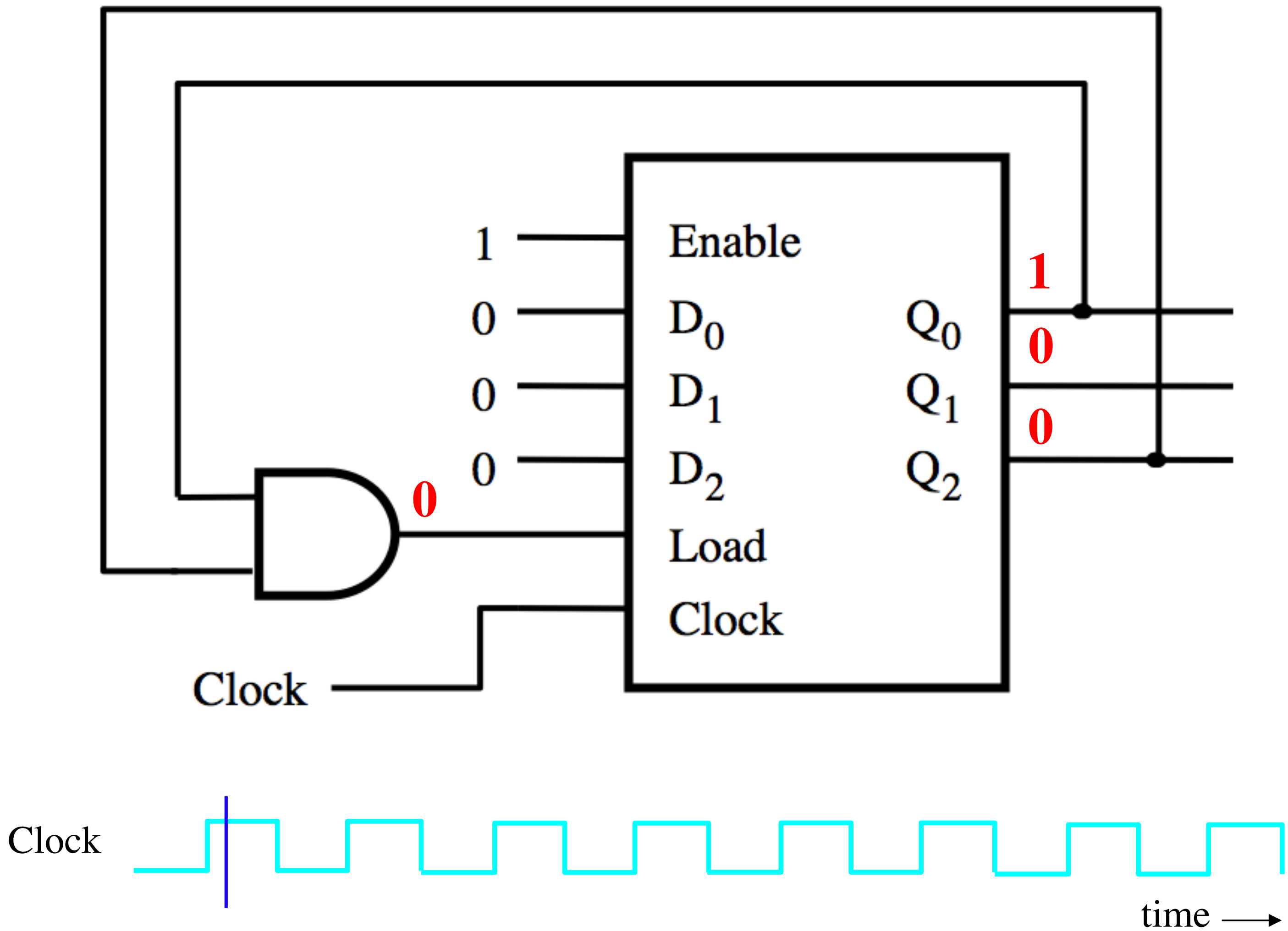
[Figure 5.25a from the textbook]

What does this circuit do?

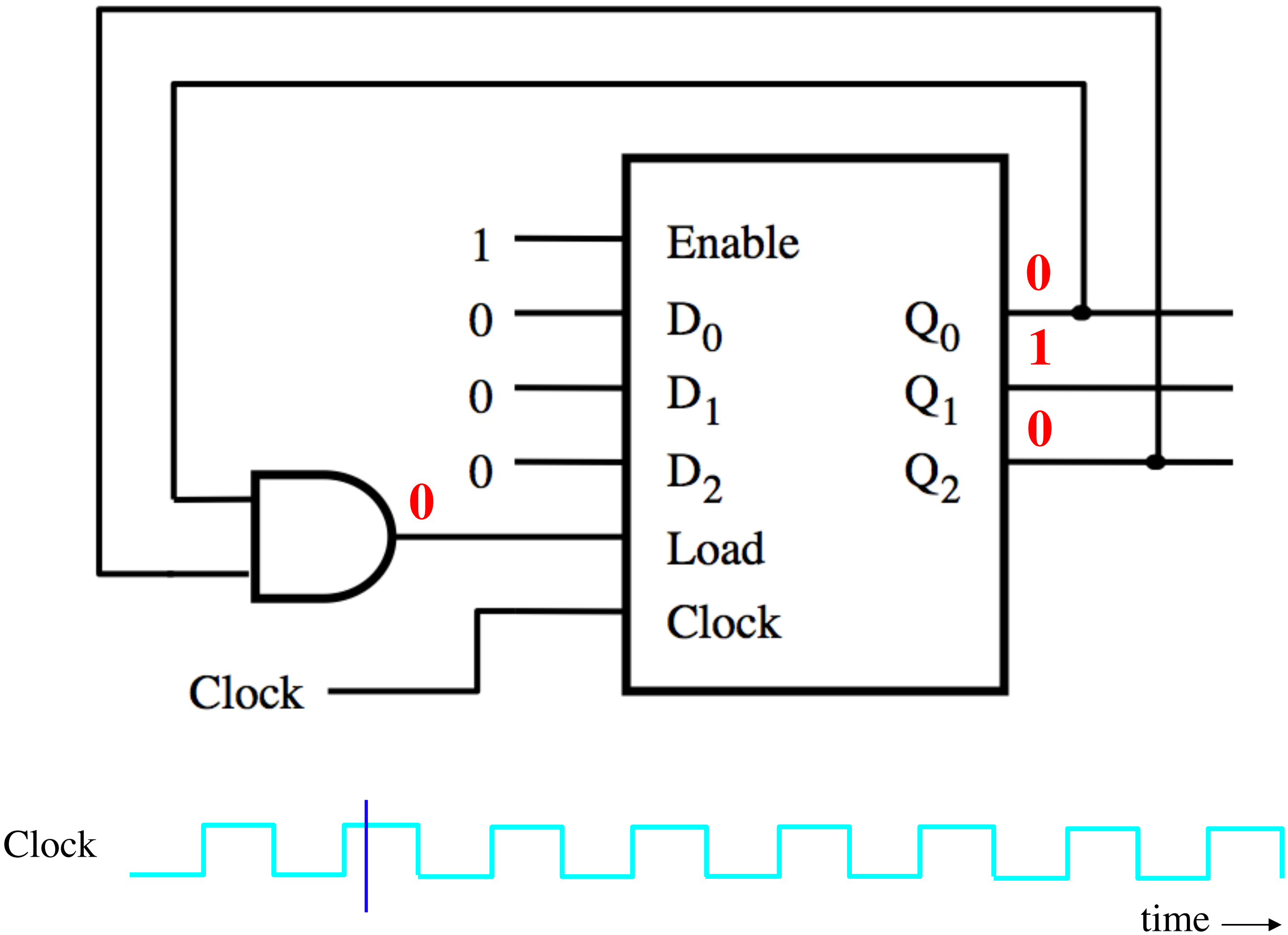




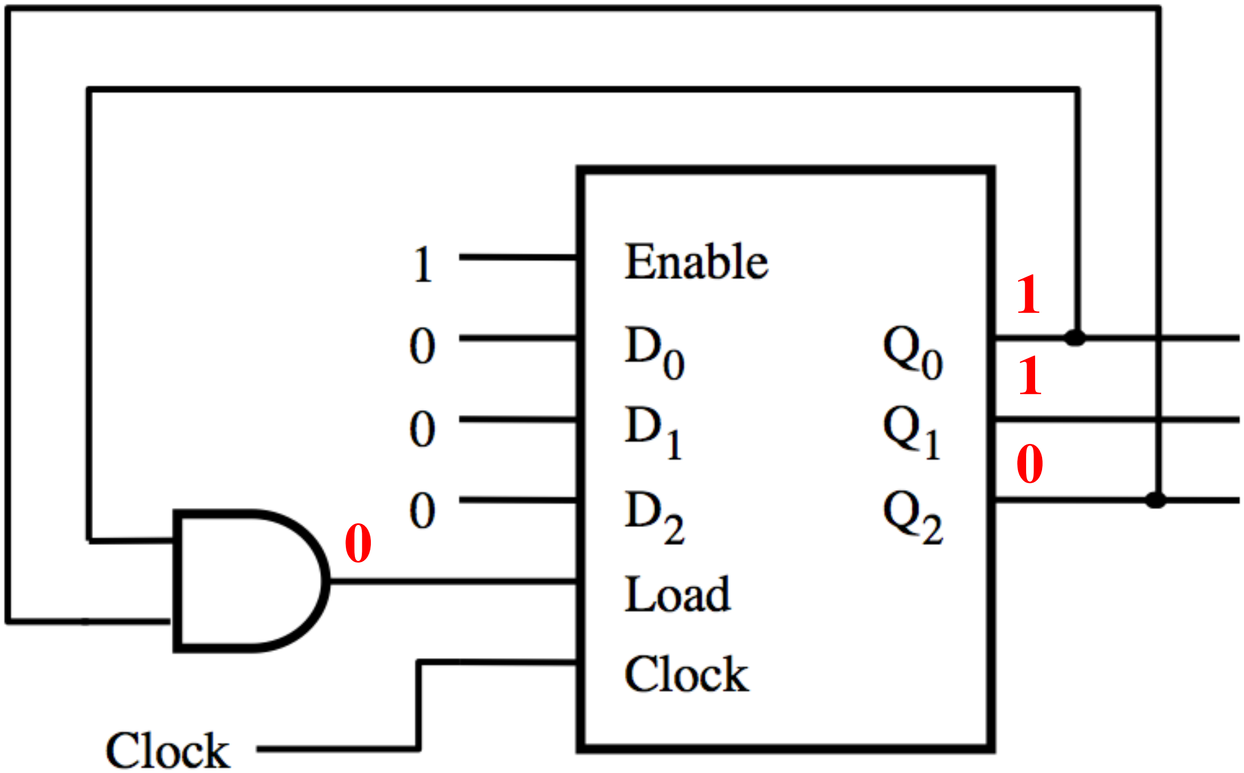
What does this circuit do?

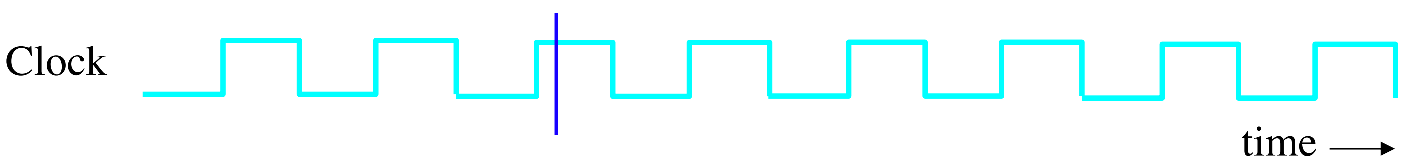


What does this circuit do?

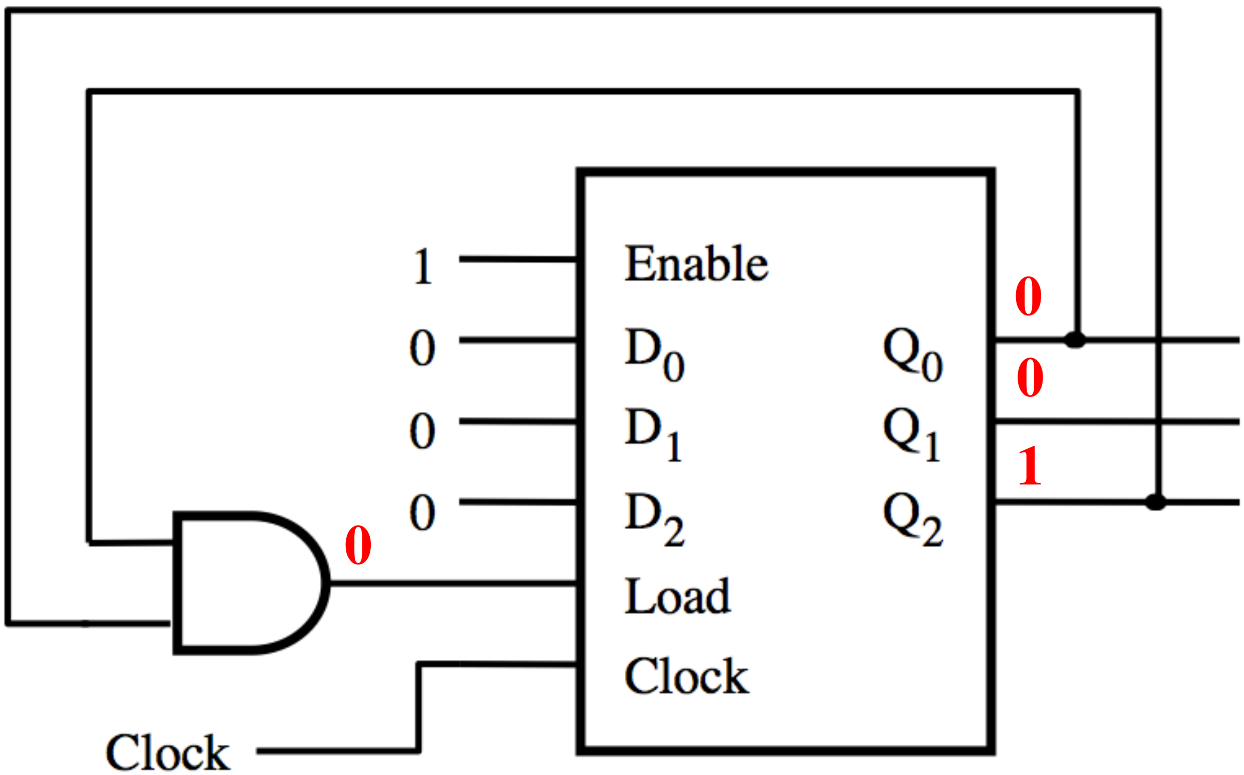


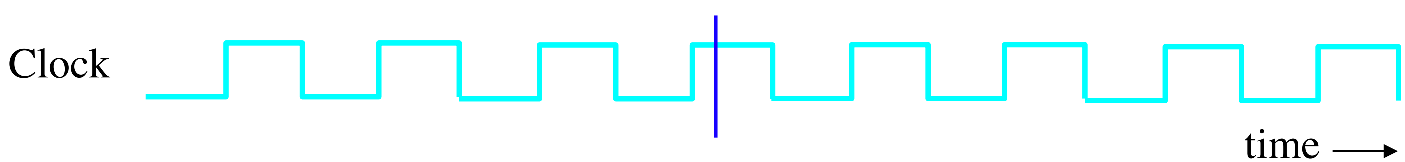
What does this circuit do?



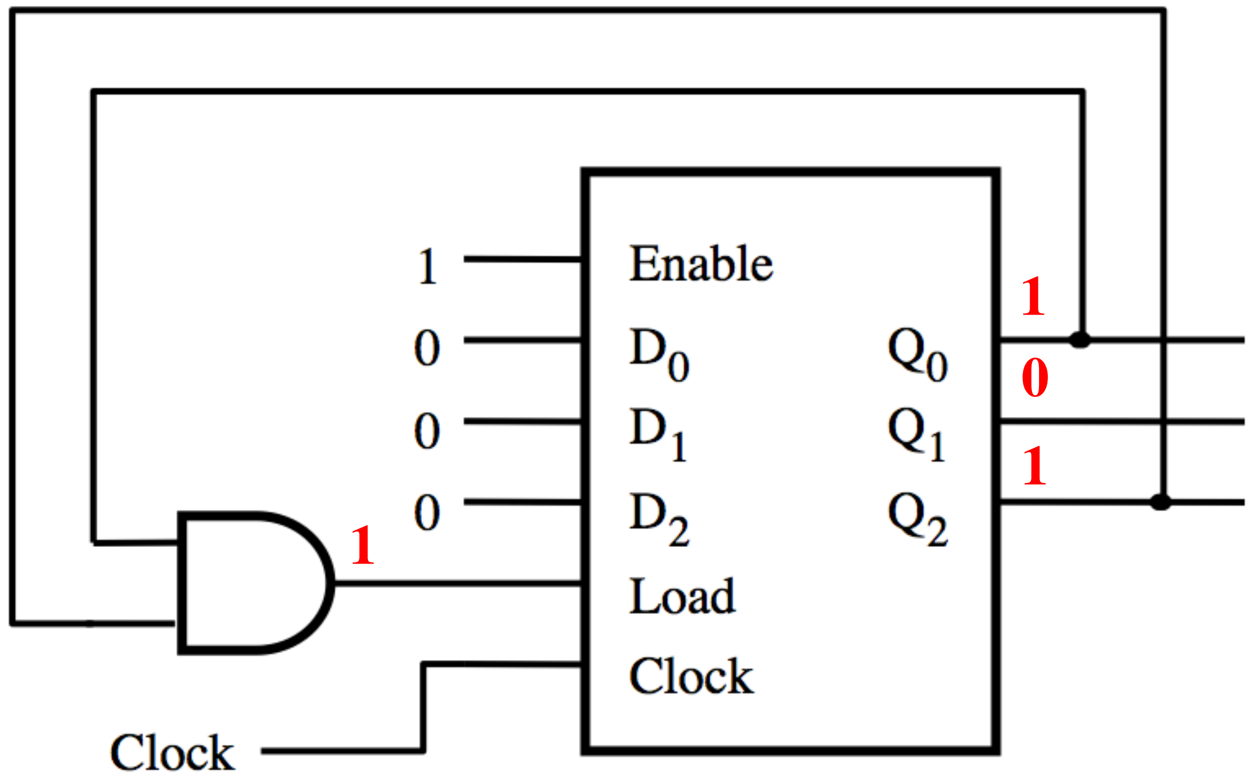


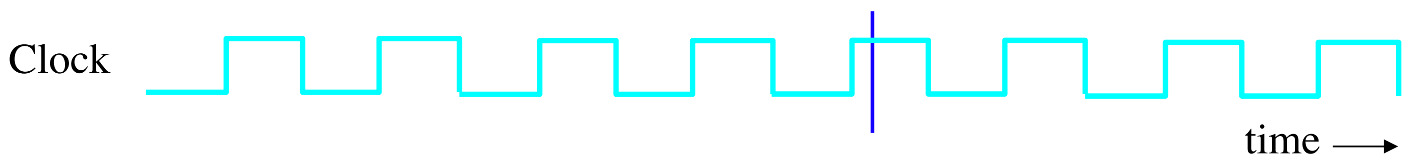
What does this circuit do?



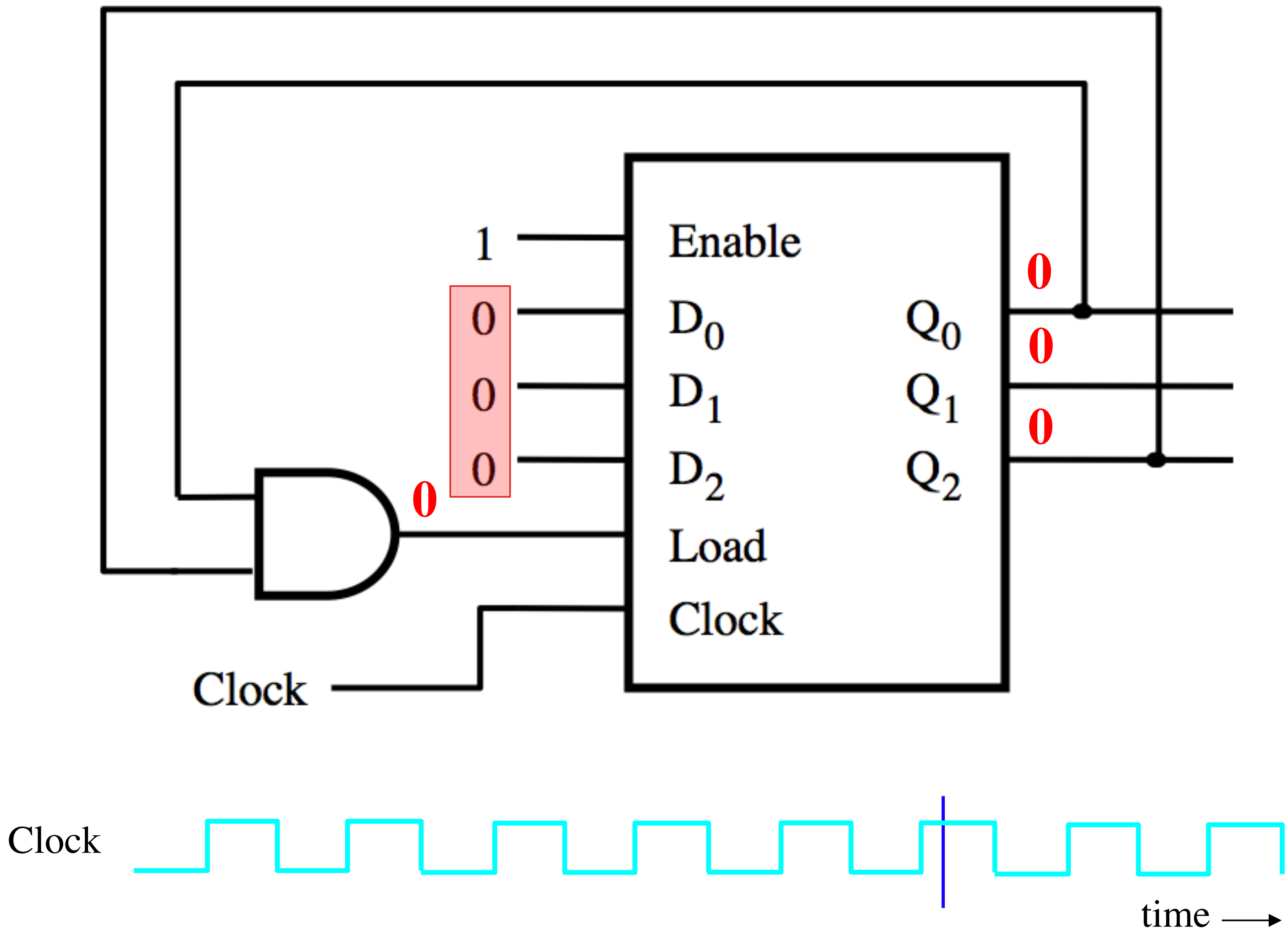


What does this circuit do?

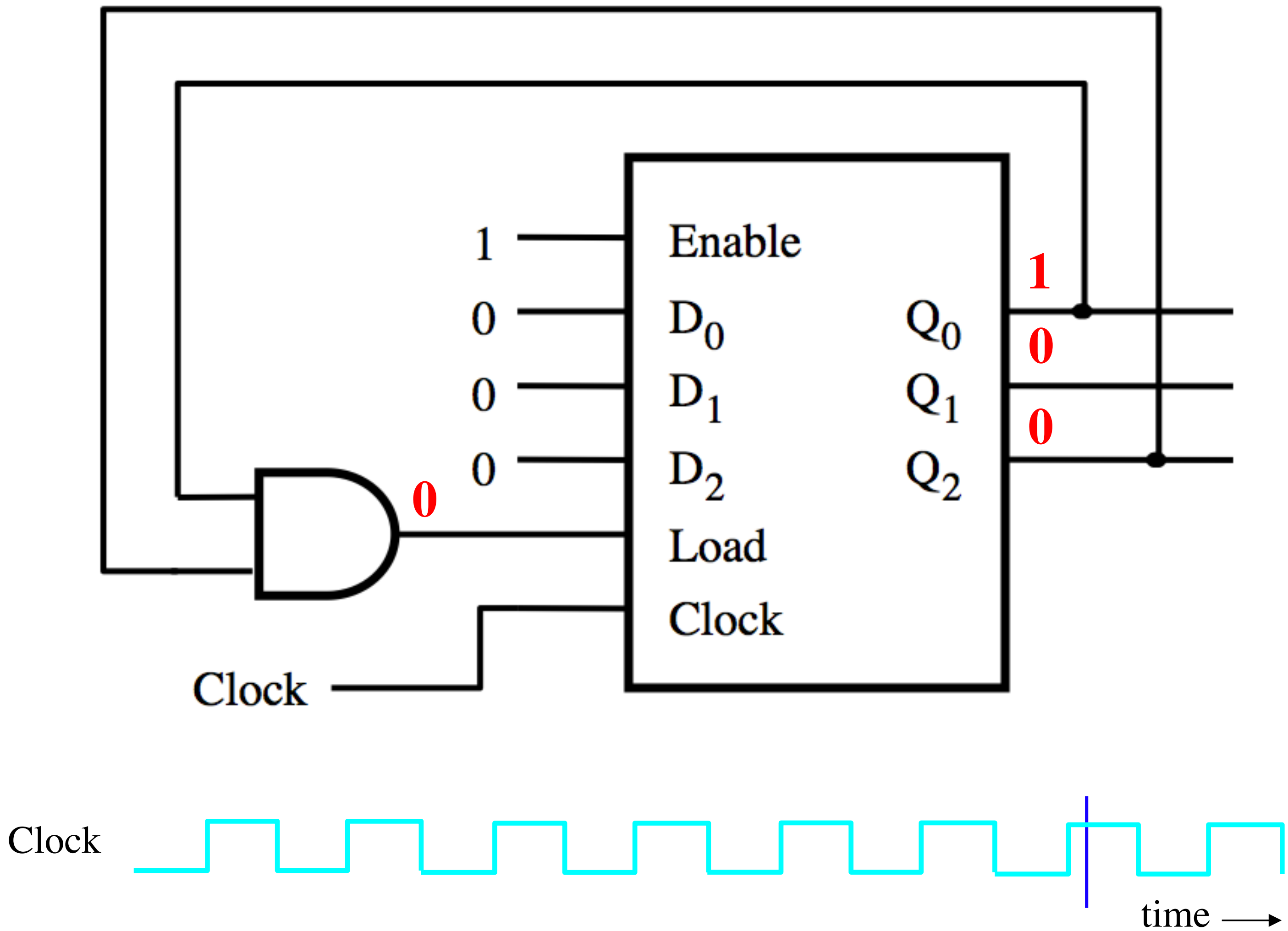




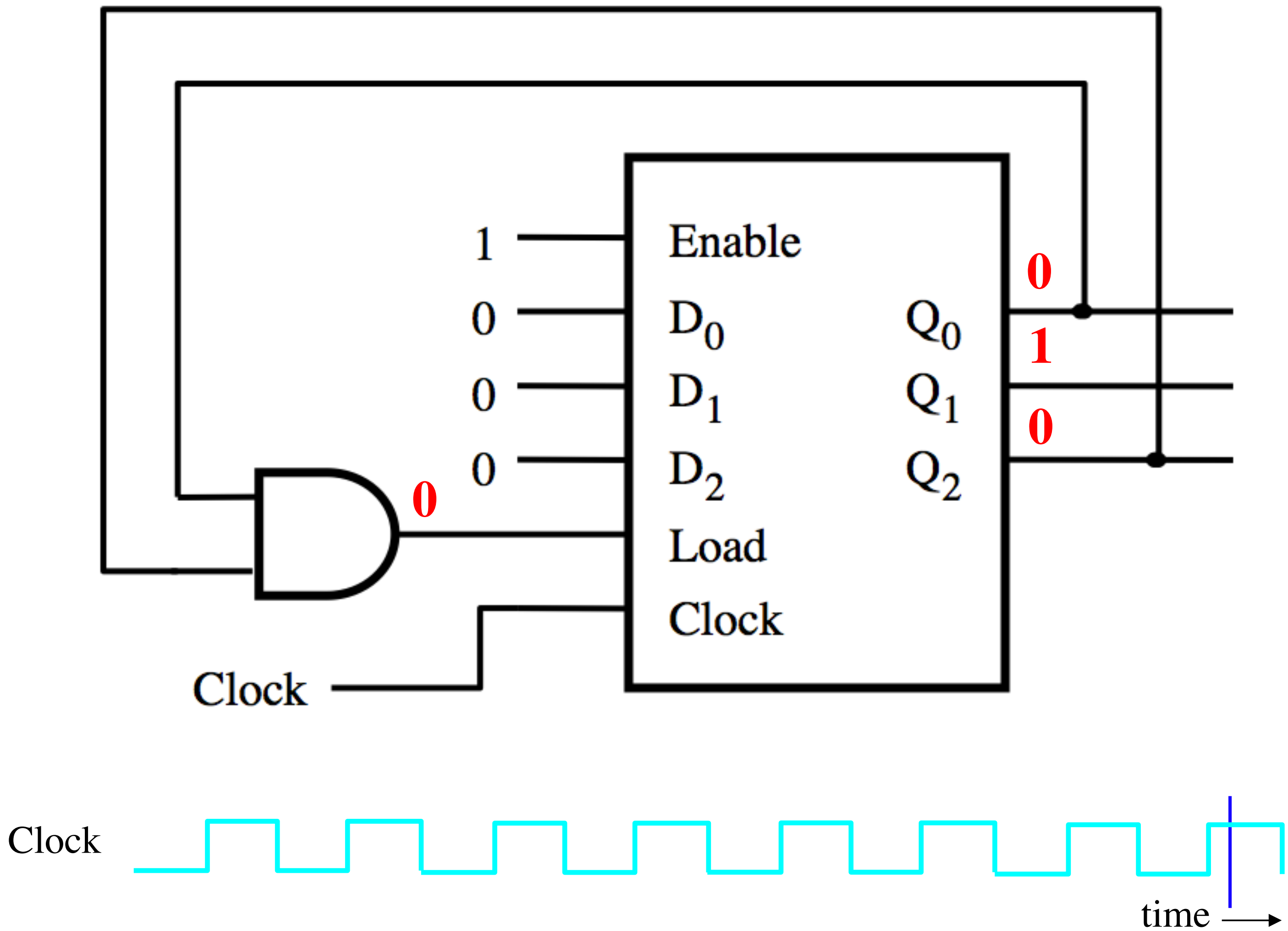
What does this circuit do?



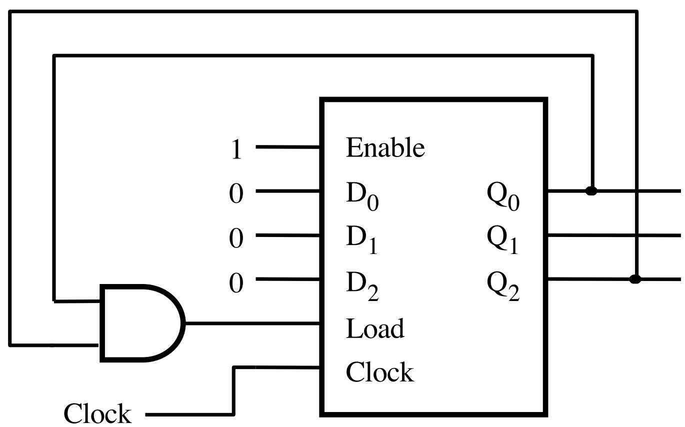
What does this circuit do?



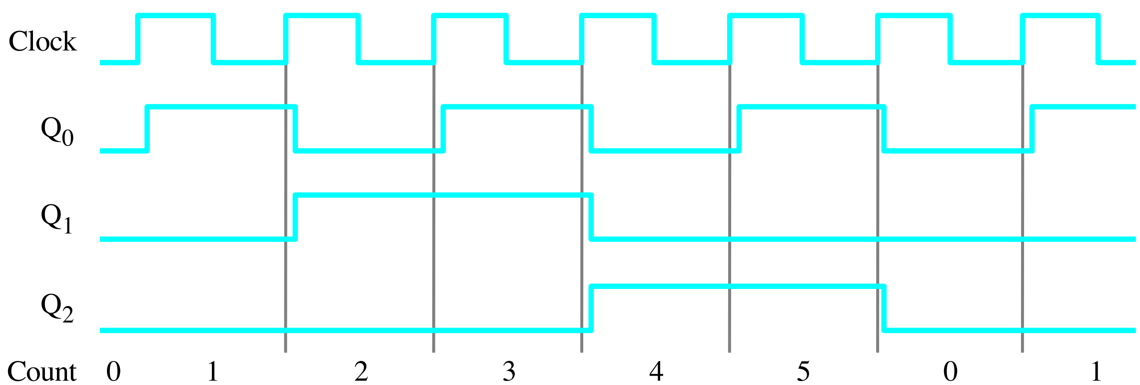
What does this circuit do?



A modulo-6 counter with synchronous reset



(a) Circuit

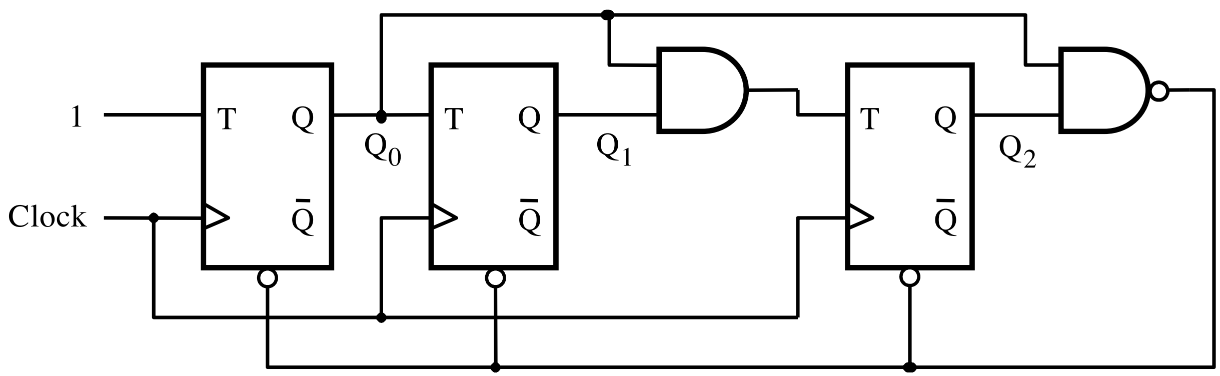


(b) Timing diagram

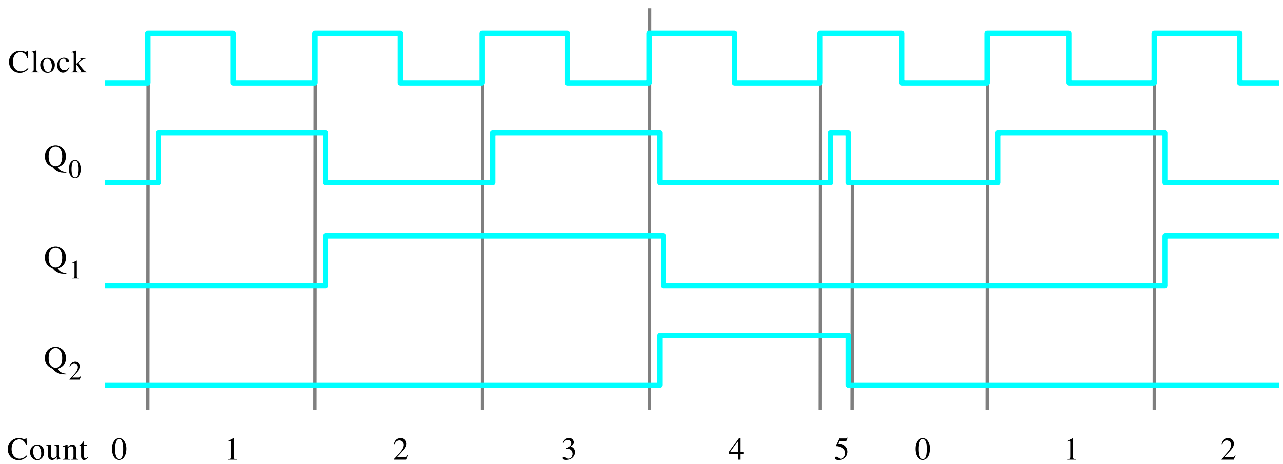
[Figure 5.25 from the textbook]

Asynchronous Reset

A modulo-6 counter with asynchronous reset



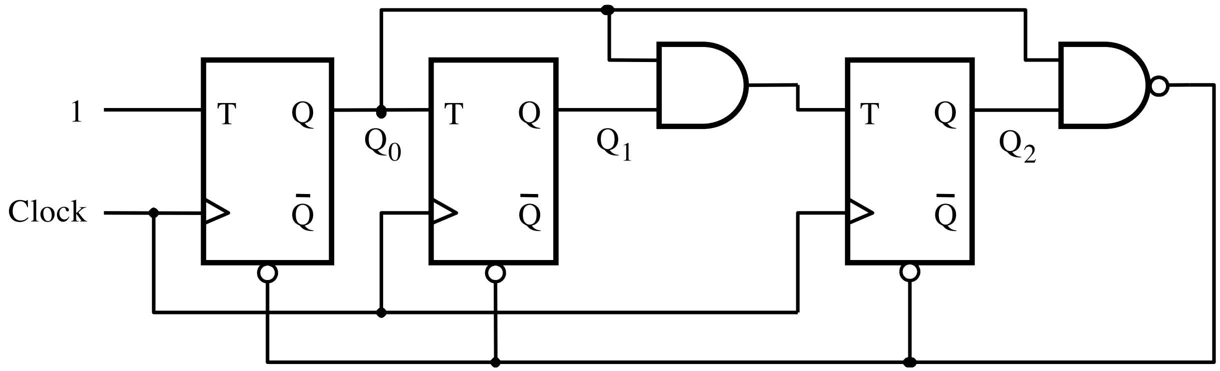
(a) Circuit



(b) Timing diagram

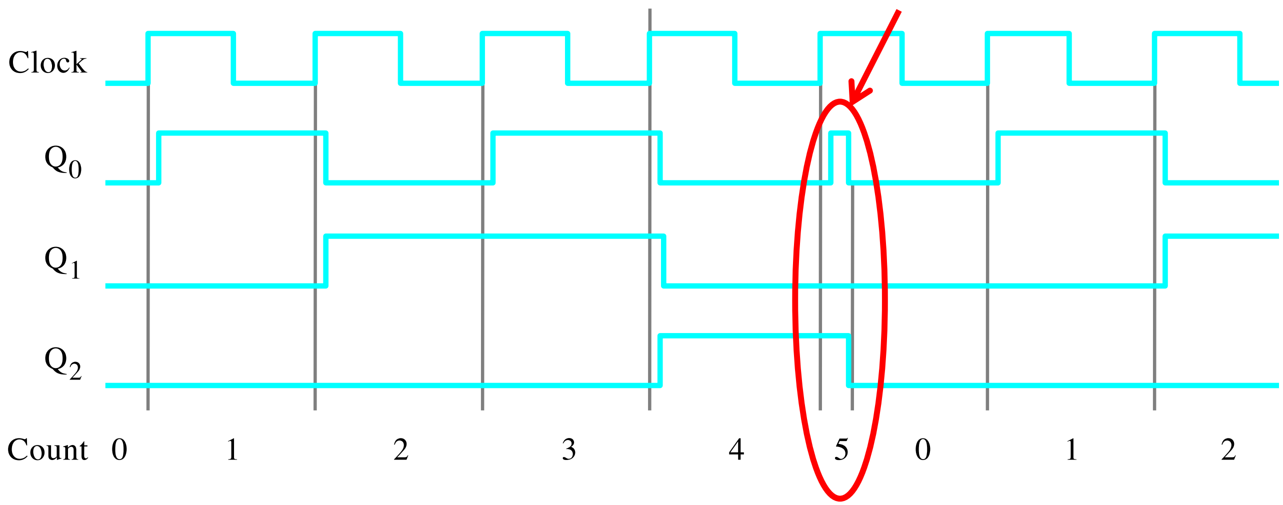
[Figure 5.26 from the textbook]

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