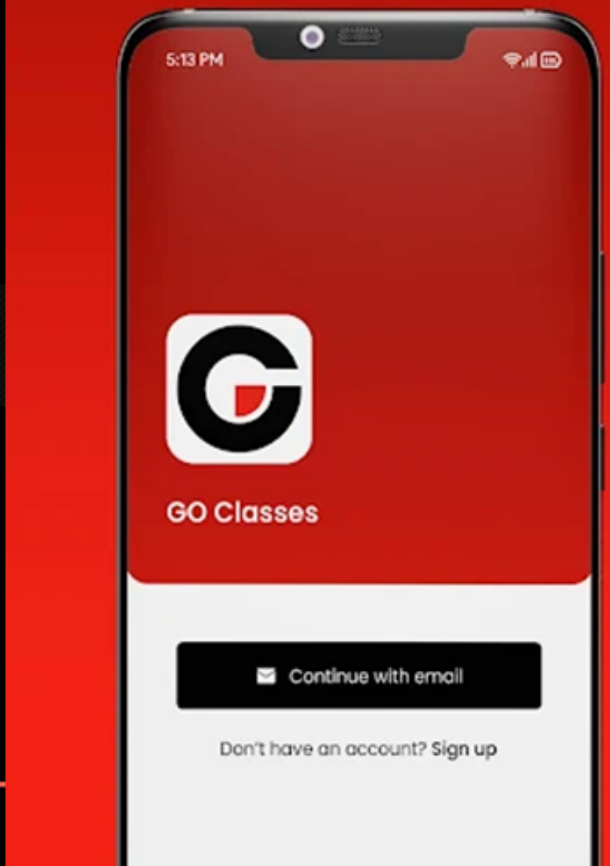
Sequential Circuits: Synchronous Counters

Ring Counter
Johnson Counter
(special counting sequences & special purpose counters)

Download the GO Classes Android App:

https://play.google.com/store/apps/details?id=com.goclasses.courses

Search "GO Classes" on Play Store.



Recap:

Synchronous Counters

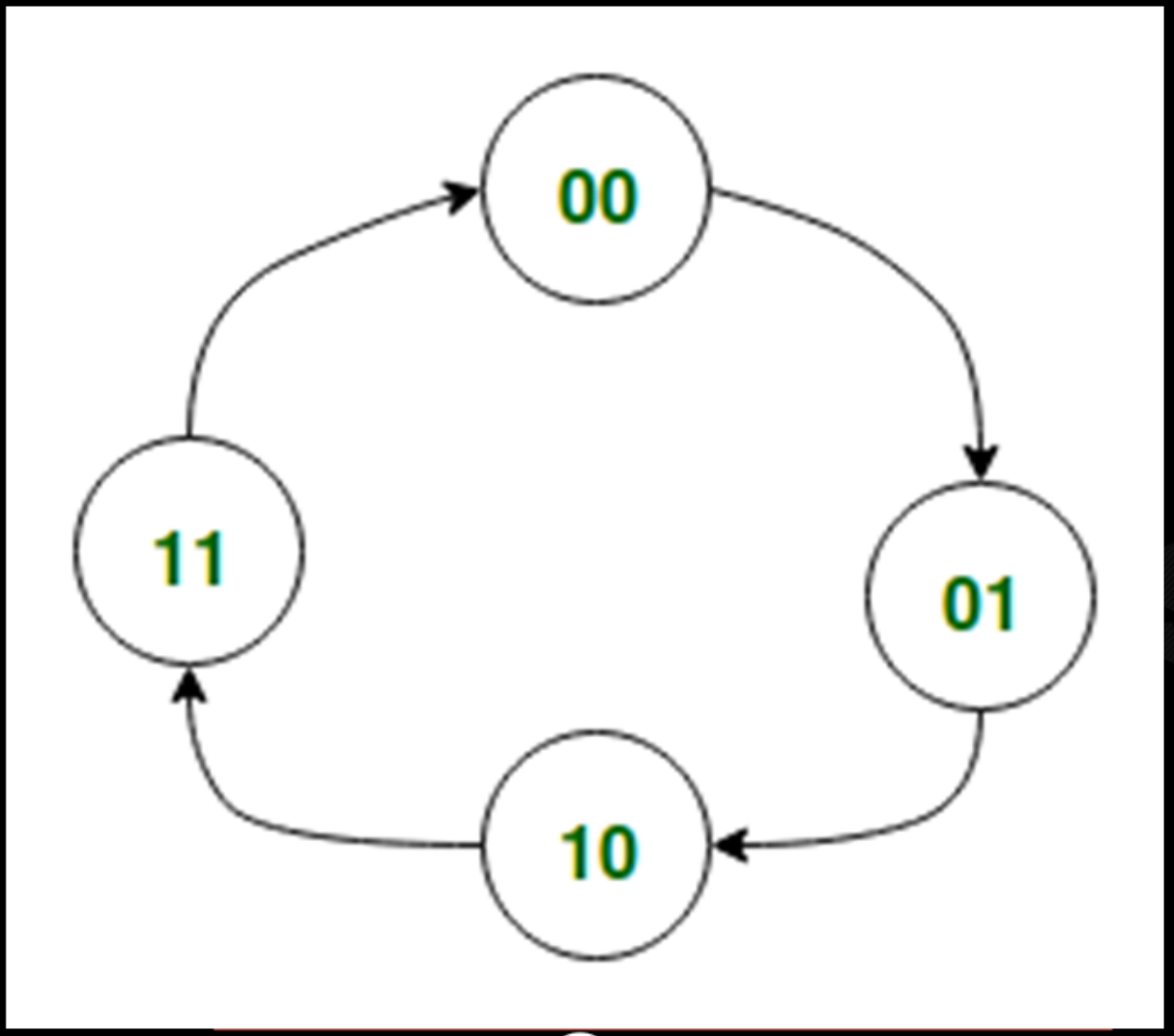
OTHER COUNTERS

Counters can be designed to generate any desired sequence of states. A divide-by-N counter (also known as a modulo-N counter) is a counter that goes through a repeated sequence of N states. The sequence may follow the binary count or may be any other arbitrary sequence. Counters are used to generate timing signals to control the sequence of operations in a digital system. Counters can also be constructed by means of shift registers. In this section, we present a few examples of nonbinary counters.

Counters can be designed to generate any desired sequence of states.

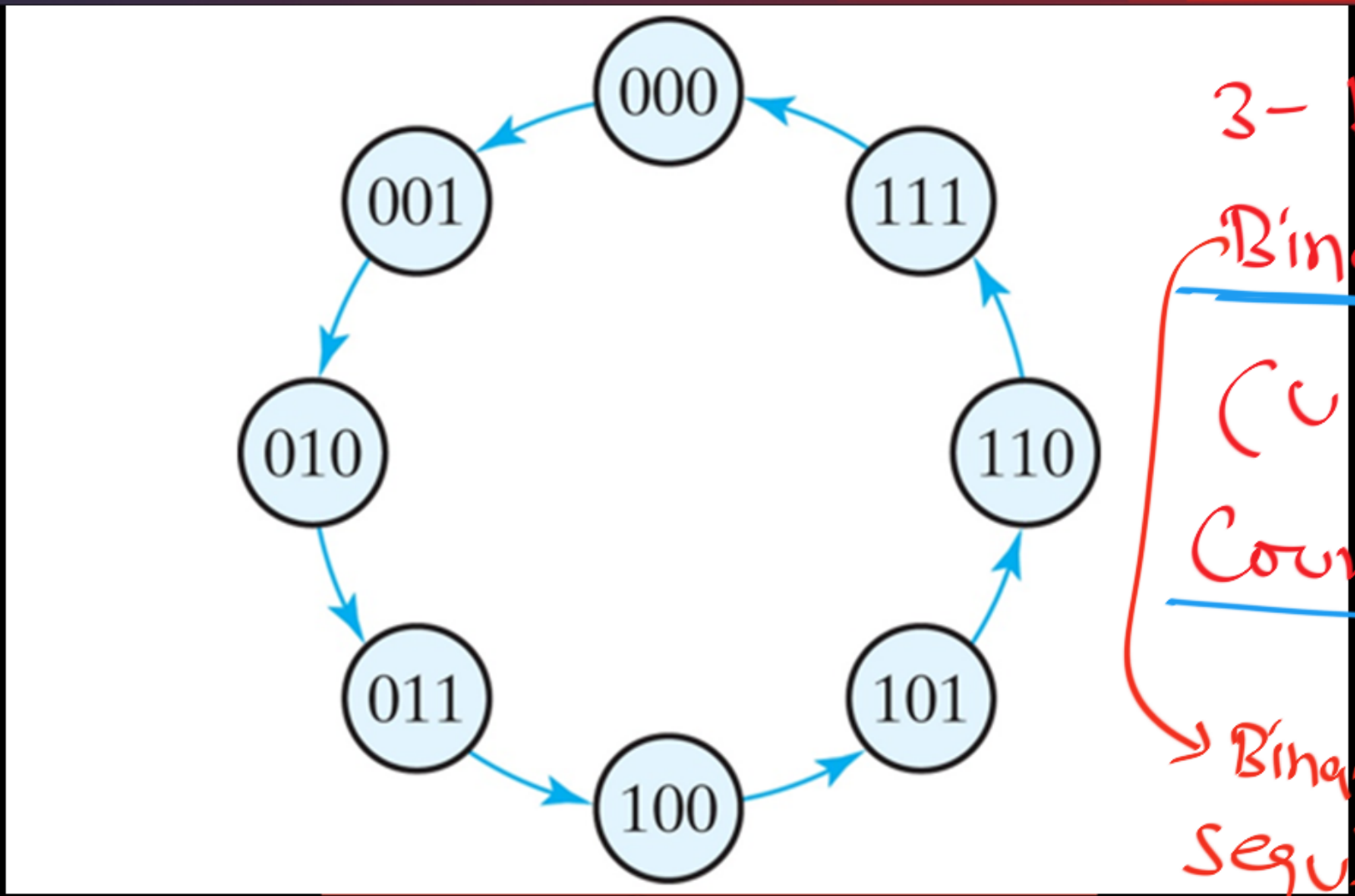
A divide-by- N counter (also known as a modulo- N counter) is a counter that goes through a repeated sequence of N states.

The sequence may follow the binary count or may be any other arbitrary sequence.



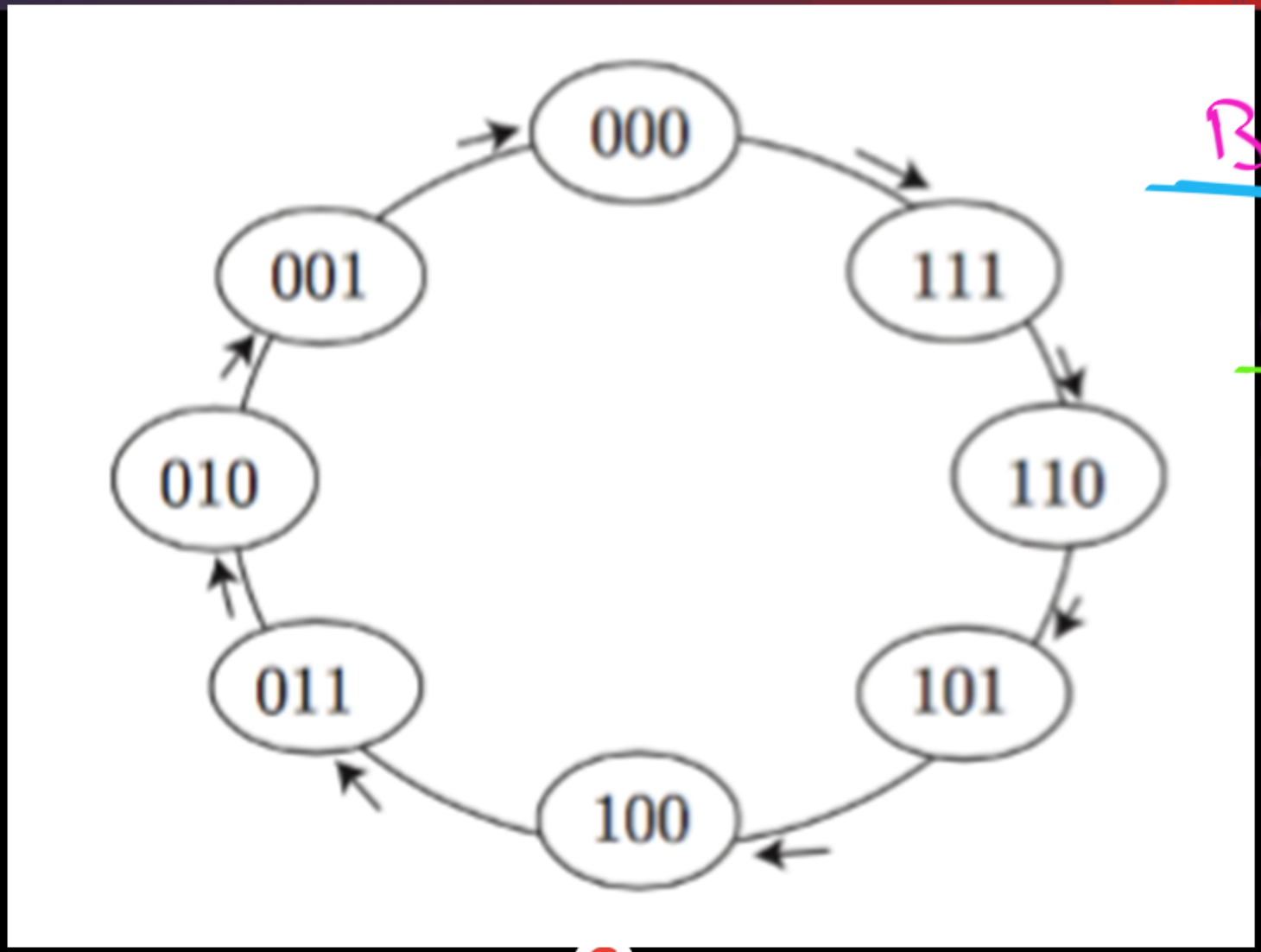
2-bit
Binary
(UP)
Counter

(mod-4
Counter)



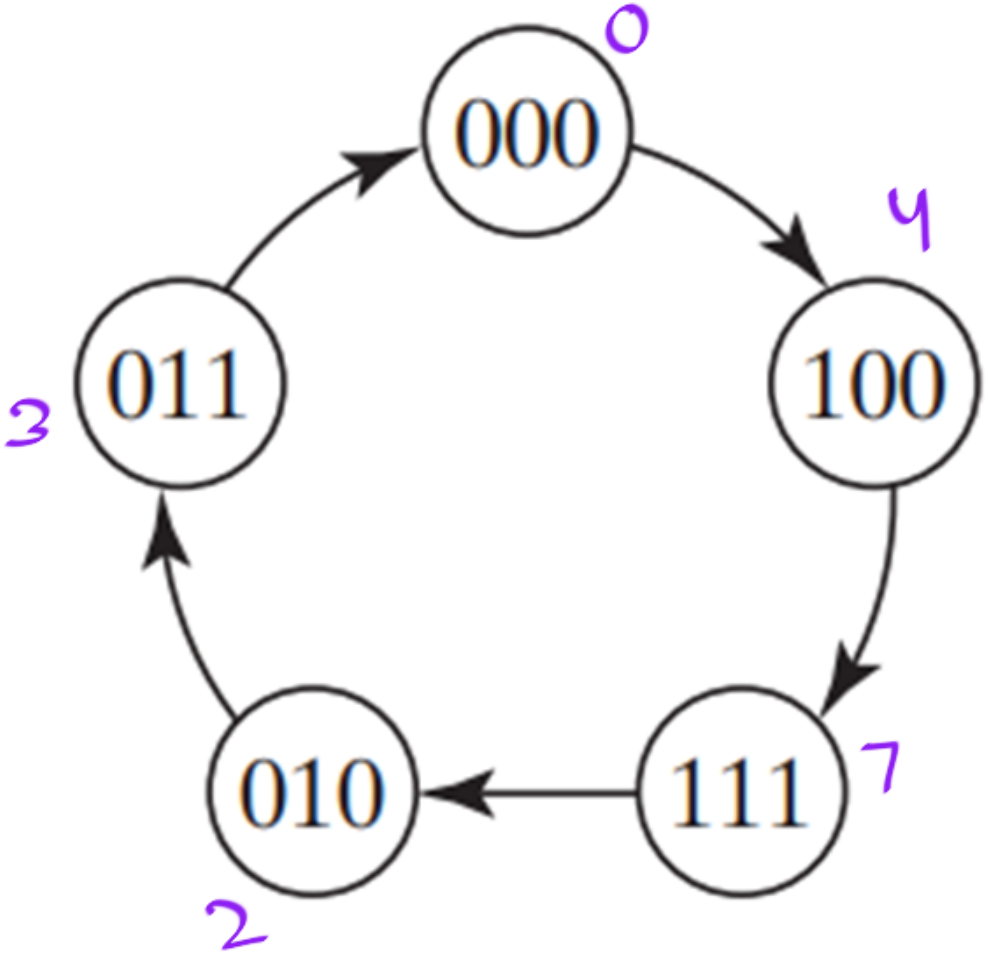
3- bit Binary (UP) Counter

Binary Sequence (UP or Down)



Binary (Down) Counter

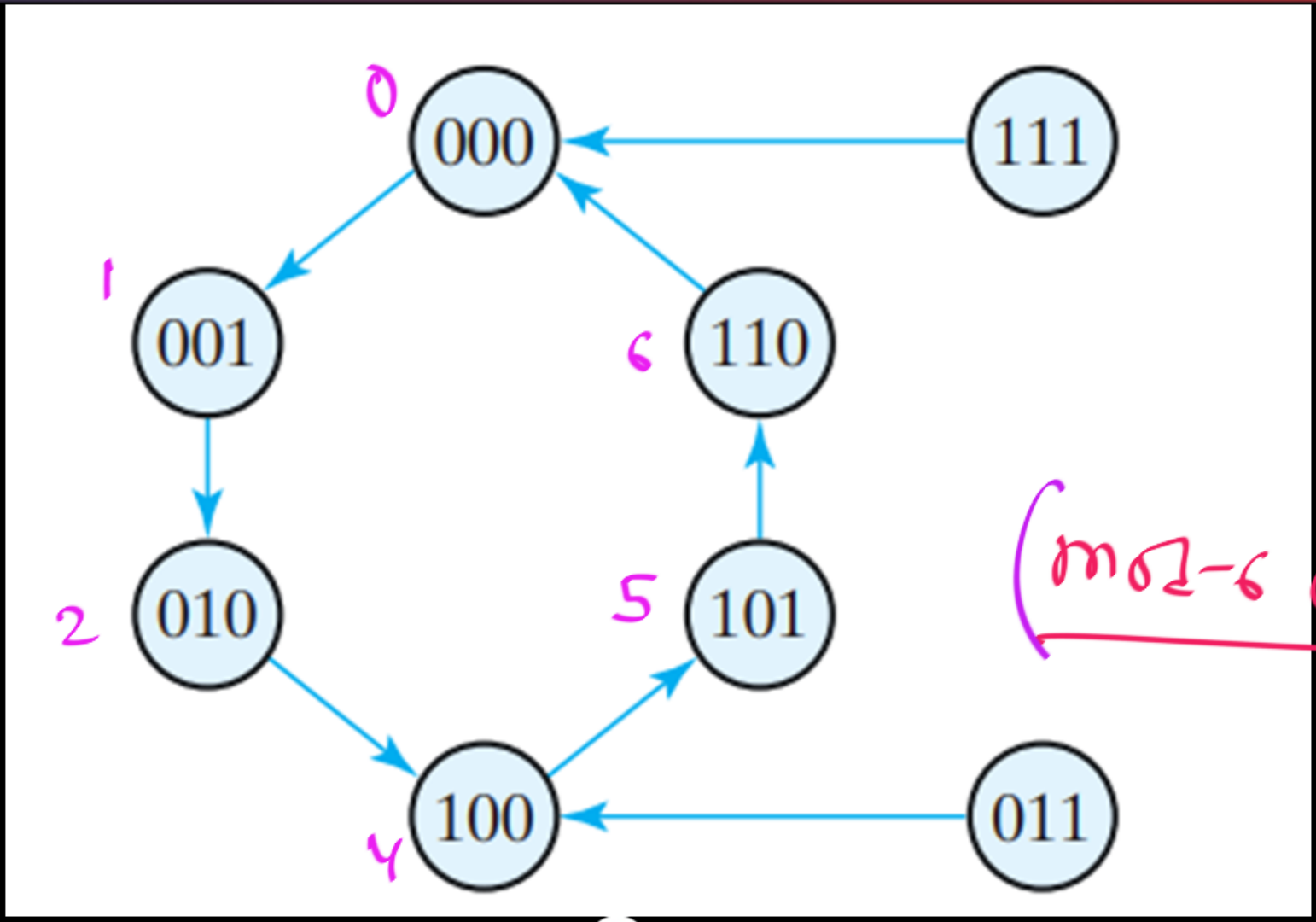
(mod-8 counter



Not binary count → following

→ Arbitrary sequence of states

(mod-5 Counter)


0
000
111
1
001
6
110
2
010
5
101
4
100
011
(mod-6 Counter)

Counters can be designed to generate any desired sequence of states.

A divide-by- N counter (also known as a modulo- N counter) is a counter that goes through a repeated sequence of N states.

The sequence may follow the binary count or may be any other arbitrary sequence.

Ring/Johnson Counter: The Purpose

```c
#include<stdio.h>
int main()
{
    int a=10, b=2, sum;
    sum = a + b;
    printf("Sum = %d",sum);
}
```

Compiler

C- program for addition

Assembly Program

Write an Assembly program to add two 8-bit numbers.

$C = A + B$

```
lds   r16, A      ; 1. Load variables
lds   r17, B
add   r16, r17    ; 2. Do something
sts   C, r16      ; 3. Store answer
```

COA:

1. fetch (memory → Register)
2. Decode
3. Execute
4. Access memory
5. write Back

In main memory

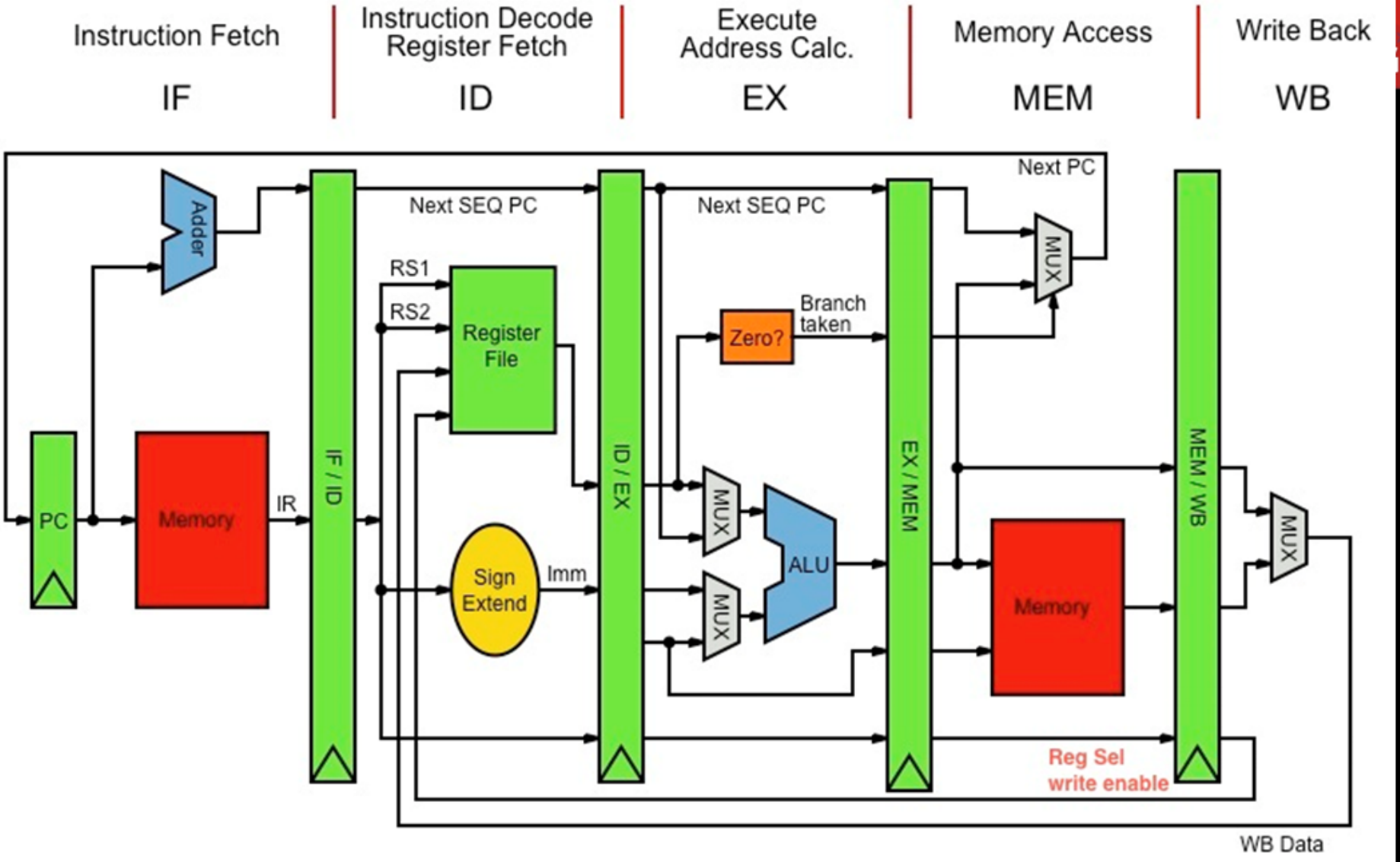
Assembly Program

```
lds   r16, A      ; 1. Load variables
lds   r17, B
add   r16, r17    ; 2. Do something
sts   C, r16      ; 3. Store answer
```

The Purpose:

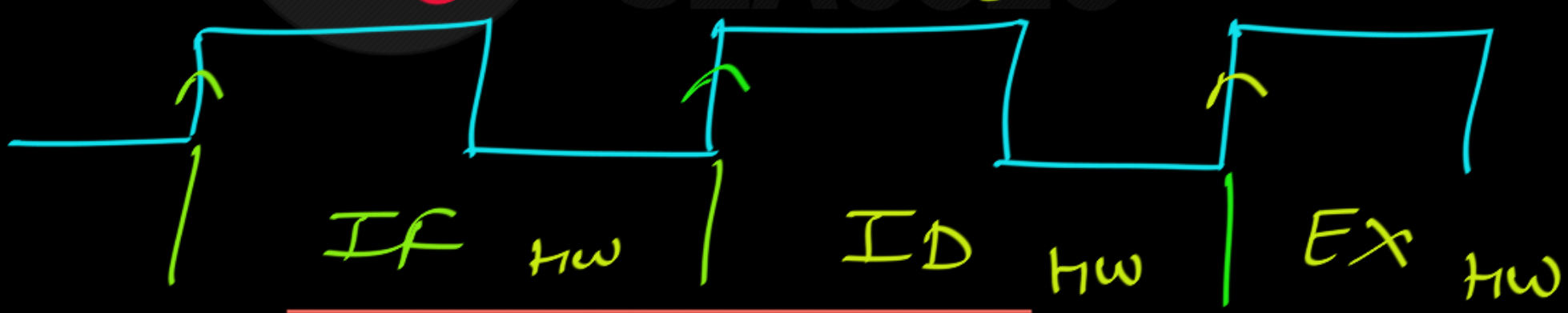
Inside CPU, Instruction execution happens in phases(stages), typically 5 stages.

- Five stages, one step per stage
  1. **IF**: Instruction fetch from (instruction) memory — HW1
  2. **ID**: Instruction decode & register read — HW 2
  3. **EX**: Execute operation or calculate address — HW 3
  4. **MEM**: Access (data) memory operand — HW 4
  5. **WB**: Write result back to register — HW 5


Instruction Fetch
IF
Instruction Decode
Register Fetch
ID
Execute
Address Calc.
EX
Memory Access
MEM
Write Back
WB
Adder
Next SEQ PC
Next SEQ PC
Next PC
MUX
RS1
RS2
Register
File
Branch
taken
Zero?
PC
Memory
IR
IF / ID
ID / EX
EX / MEM
MEM / WB
MUX
MUX
ALU
Sign
Extend
Imm
Memory
MUX
Reg Sel
write enable
WB Data

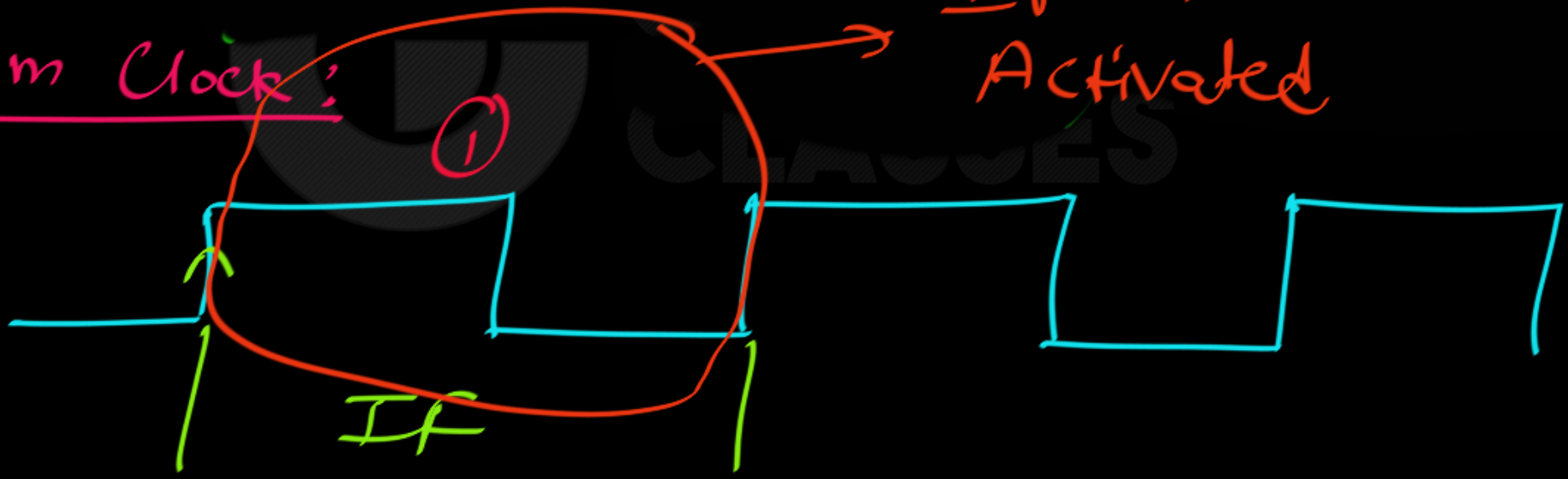
	Clock Number								
	1	2	3	4	5	6	7	8	9
Instruction i	IF	ID	EX	MEM	WB				

System Clock : → Computer (Your) → 2.3 GHz



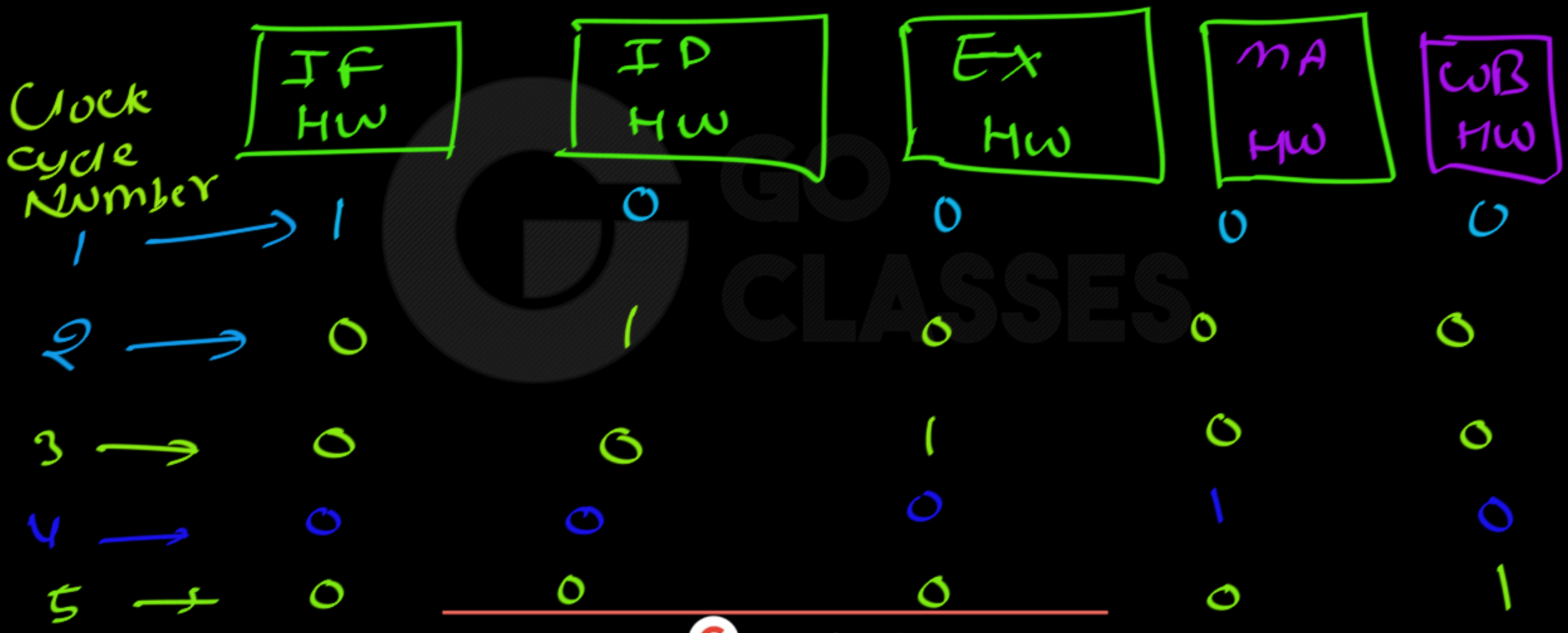
	Clock Number								
	1	2	3	4	5	6	7	8	9
Instruction i	IF	ID	EX	MEM	WB				

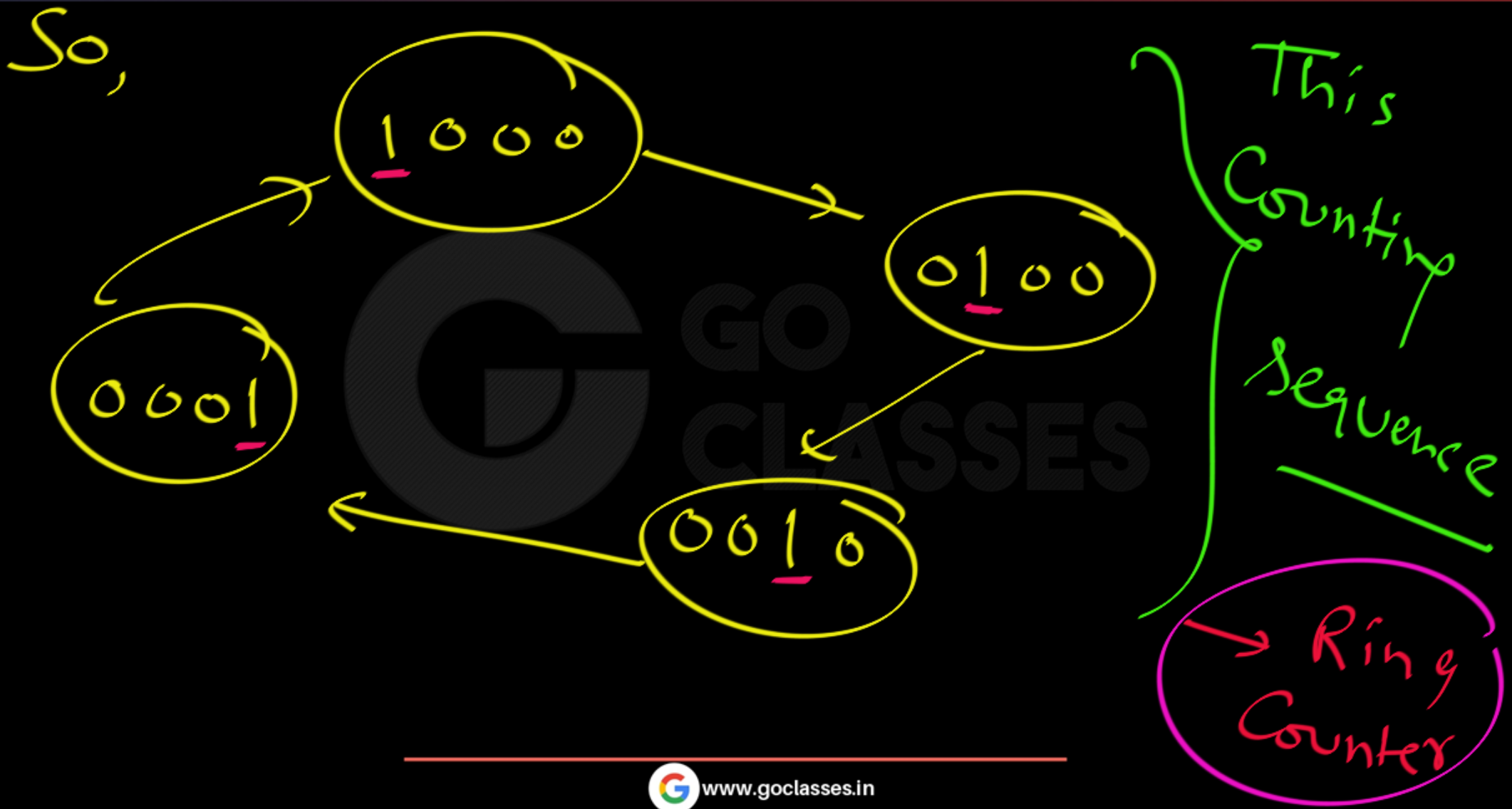
System Clock :



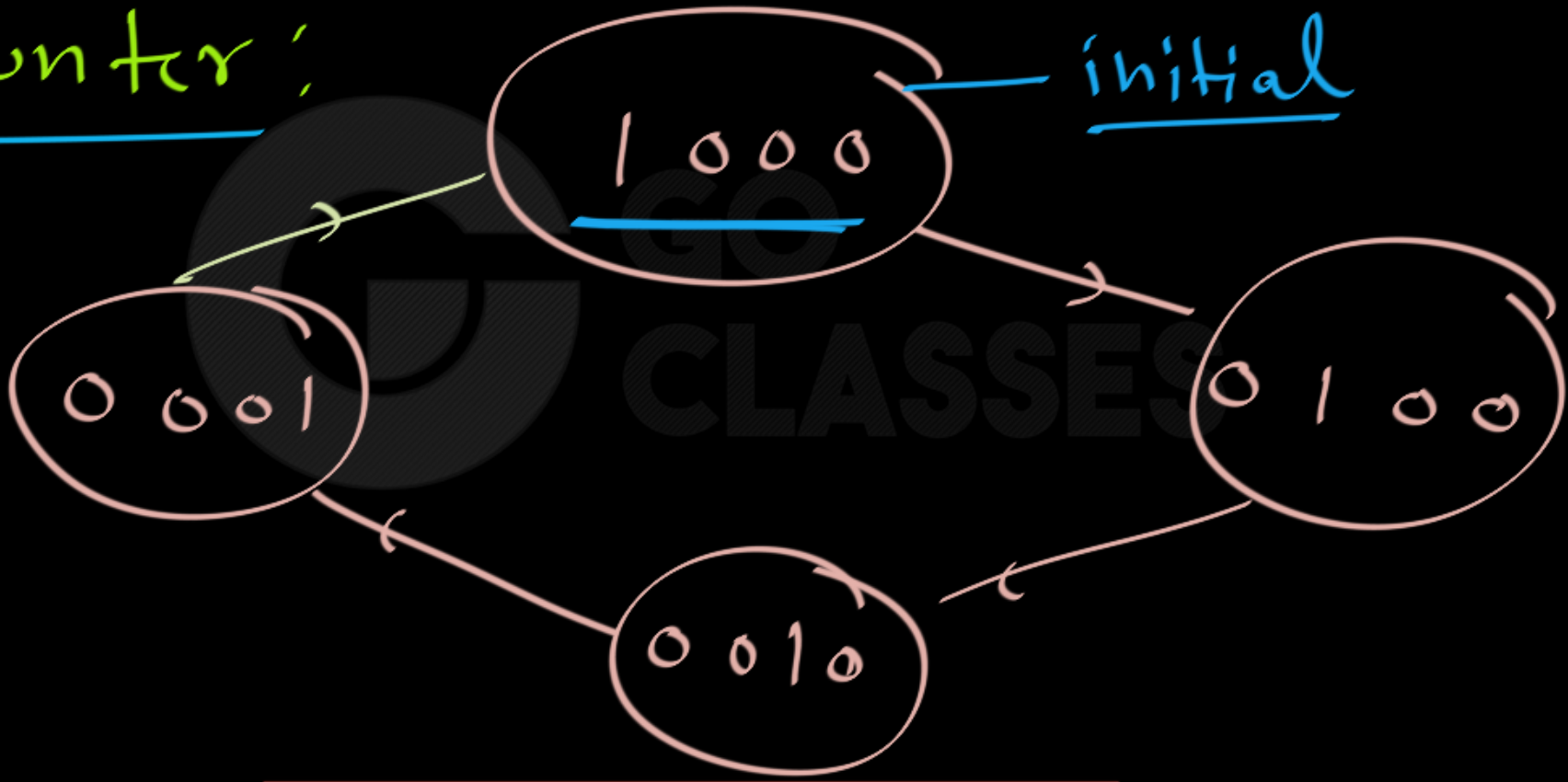
IF HW Activated

Conclusion :


IF HW
ID HW
Ex HW
MA HW
WB HW
Clock cycle Number
1 → 1 0 0 0 0
2 → 0 1 0 0 0
3 → 0 0 1 0 0
4 → 0 0 0 1 0
5 → 0 0 0 0 1


So,
1000
0100
0001
0010
This Counting Sequence
Ring Counter

The Purpose of Ring / Johnson Counter :



The Purpose:

Ring/Johnson Counter is used to generate Timing signals that control the sequence of operations in a digital system

The Purpose:

Ring/Johnson Counter is used to generate Timing signals that control the sequence of operations in a digital system

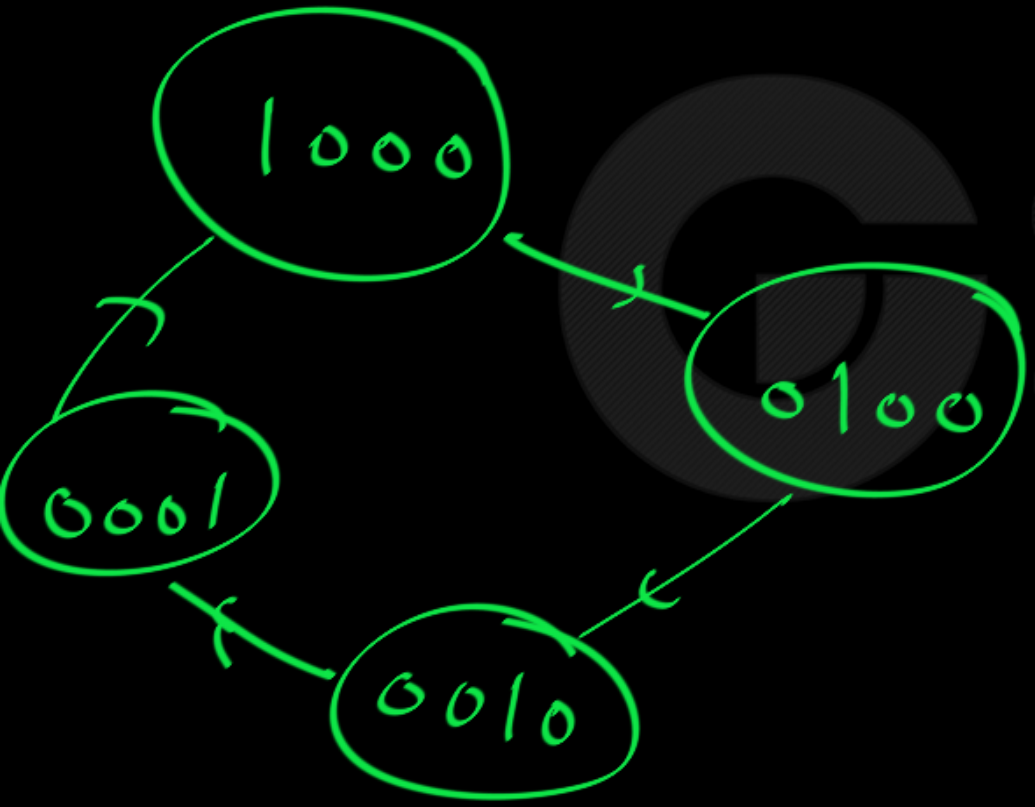
Next Topic:

Ring Counter

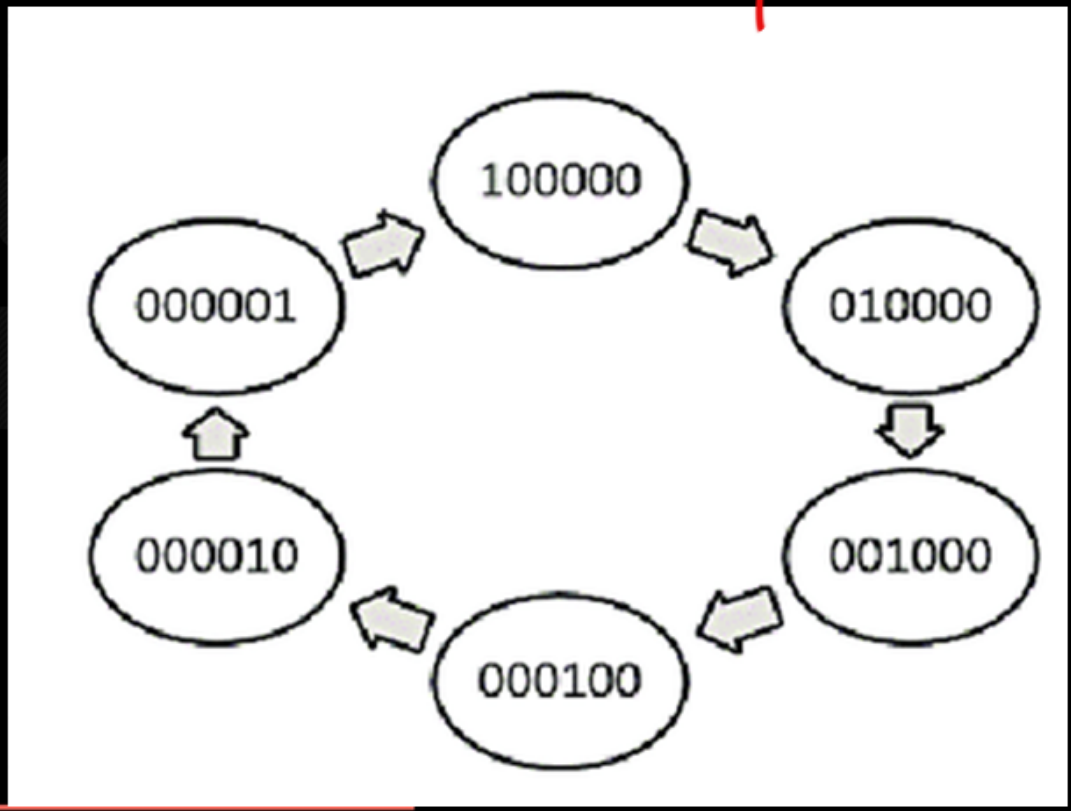
(Straight Ring Counter)

Ring Counter: The Definition

4 bit Ring Counter



6 bit Ring Counter



Ring Counter Definition:

A ring counter is a circular shift register with only one flip-flop being set at any particular time; all others are cleared.

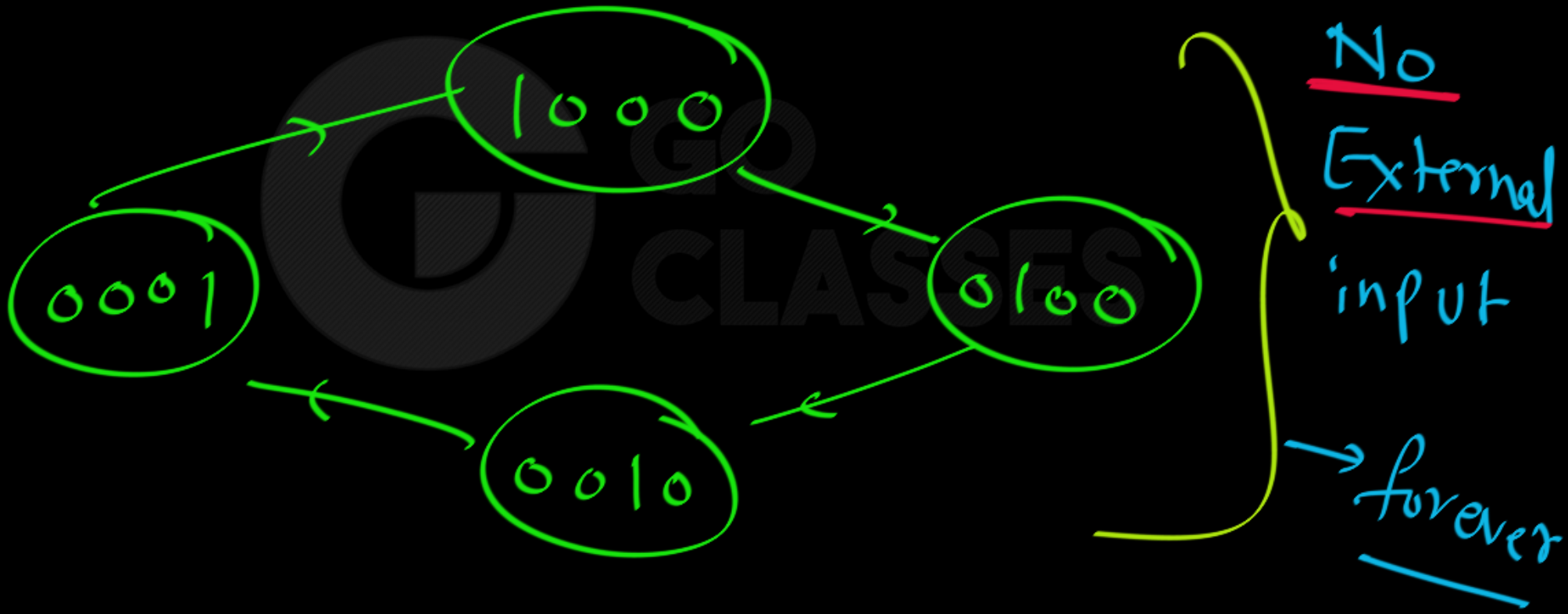
The single bit is shifted from one flip-flop to the next to produce the sequence of timing signals.

4-bit ring counter

- There is only one 1 on the outputs of the four flip-flops
- The counting sequence is: 1000, 0100, 0010, 0001, 1000, ...

Ring Counter: The Implementation

Implement the Counting Sequence;

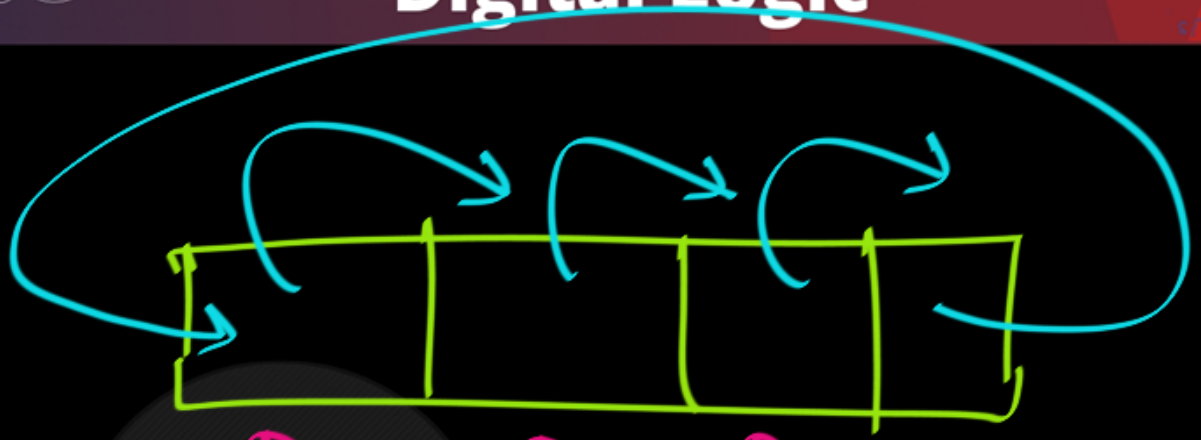


Ring Counter:

Implementation 1:

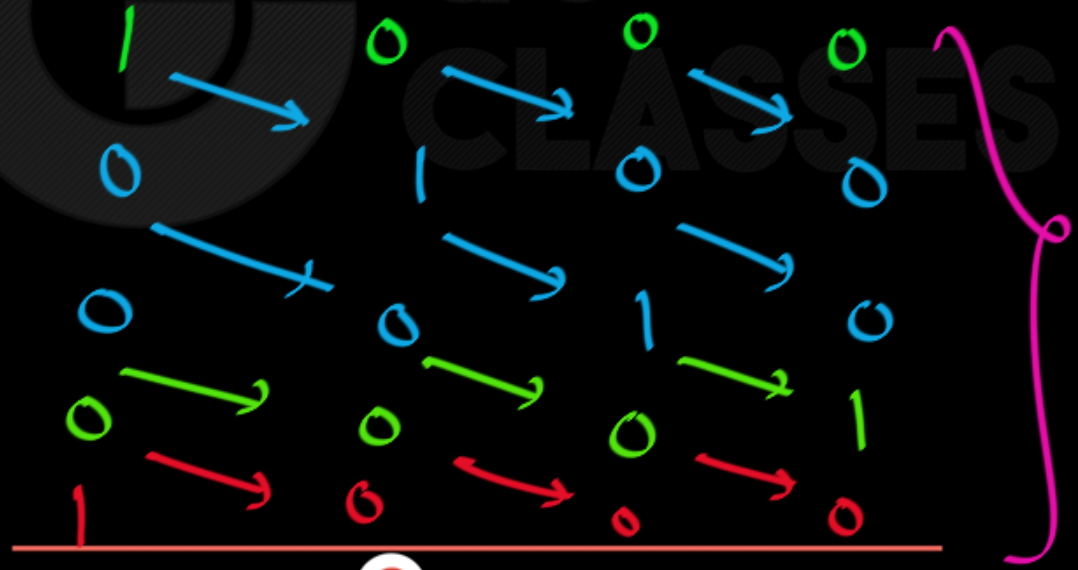
Using Shift register

Idea :



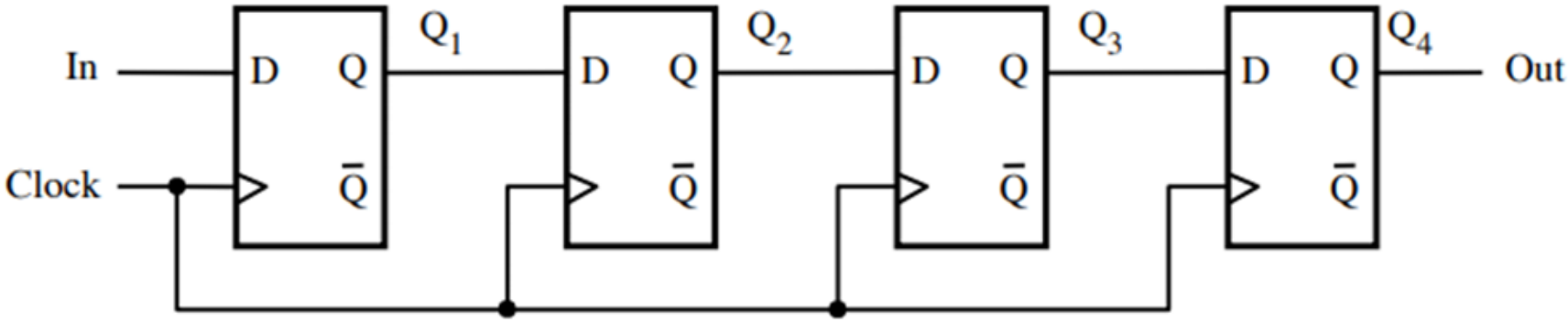
→ Ring Counter

'initially :

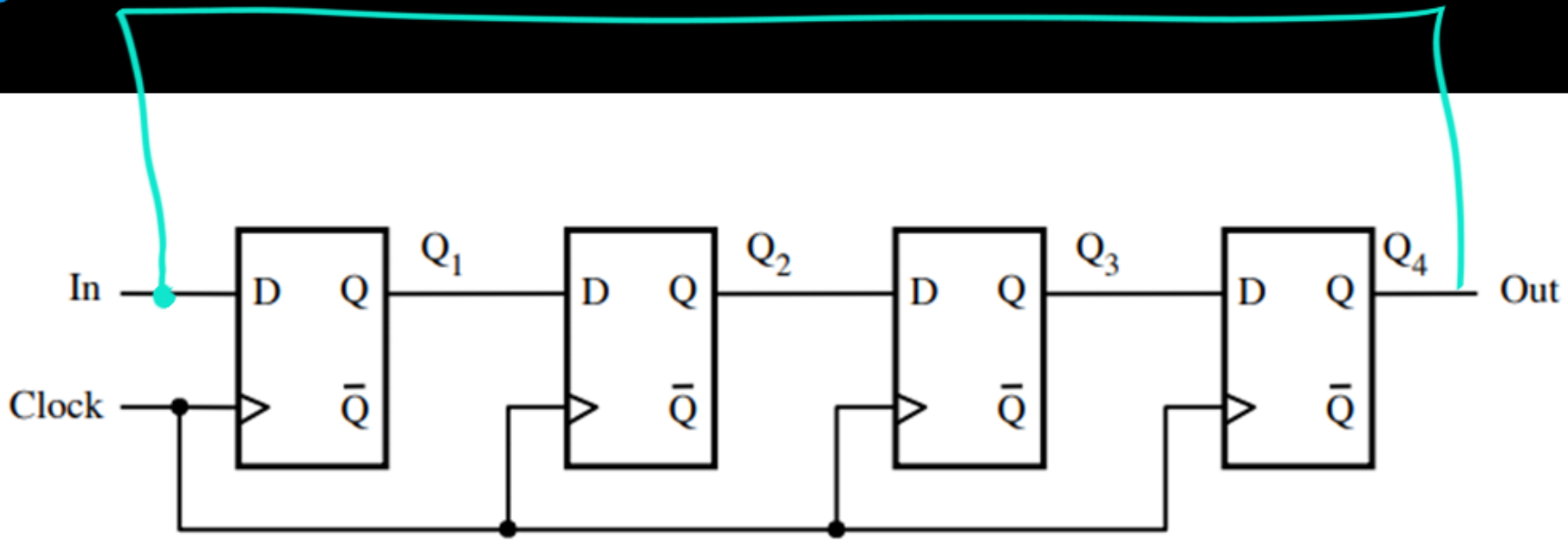


To 'implement

Shift Register:

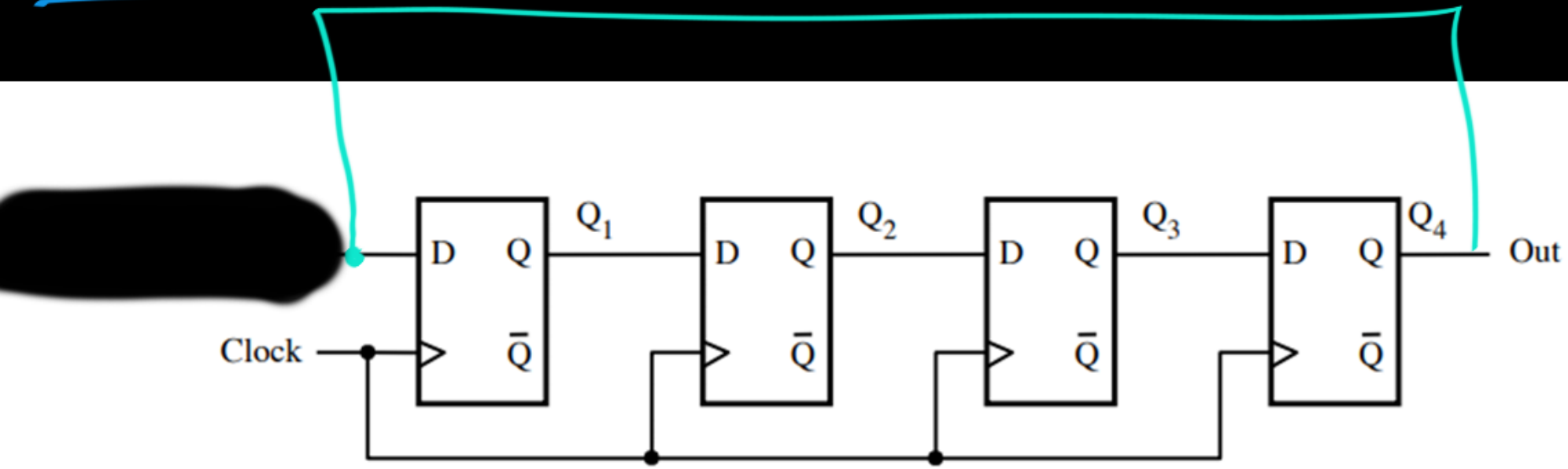


Step 1:

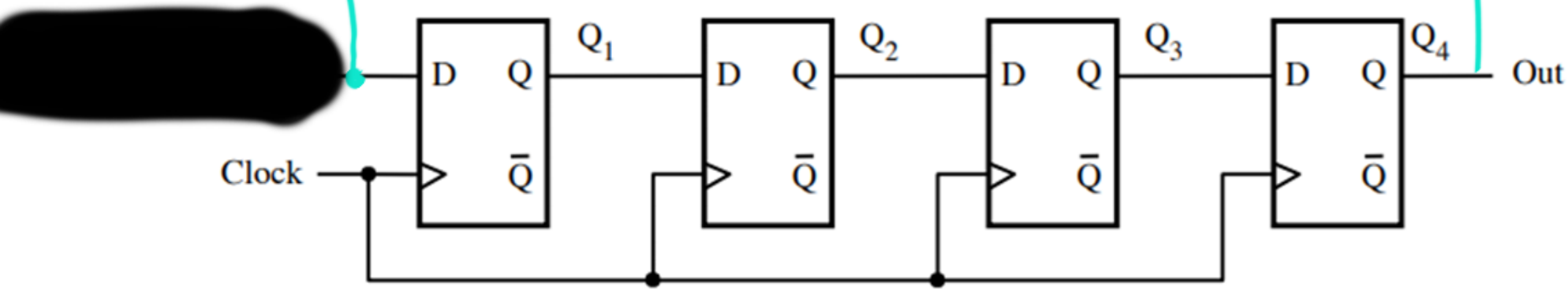


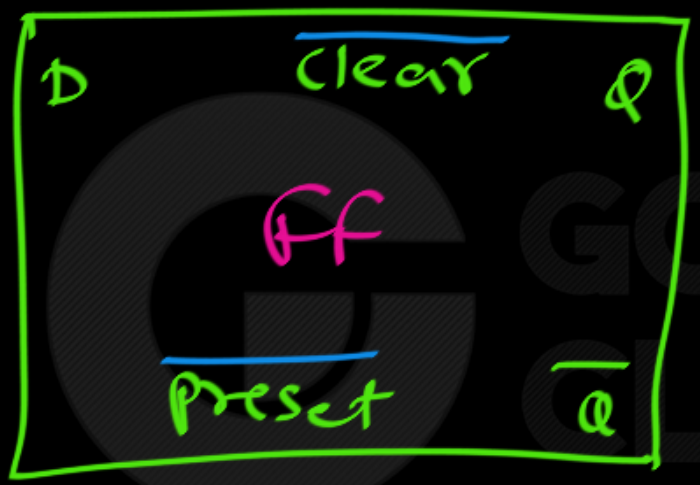
Shift Register ⟶ Ring Counter

Step2 :→ Remove Exteral input



Step 3 :→ initialize to 1000 ⇒ How ??





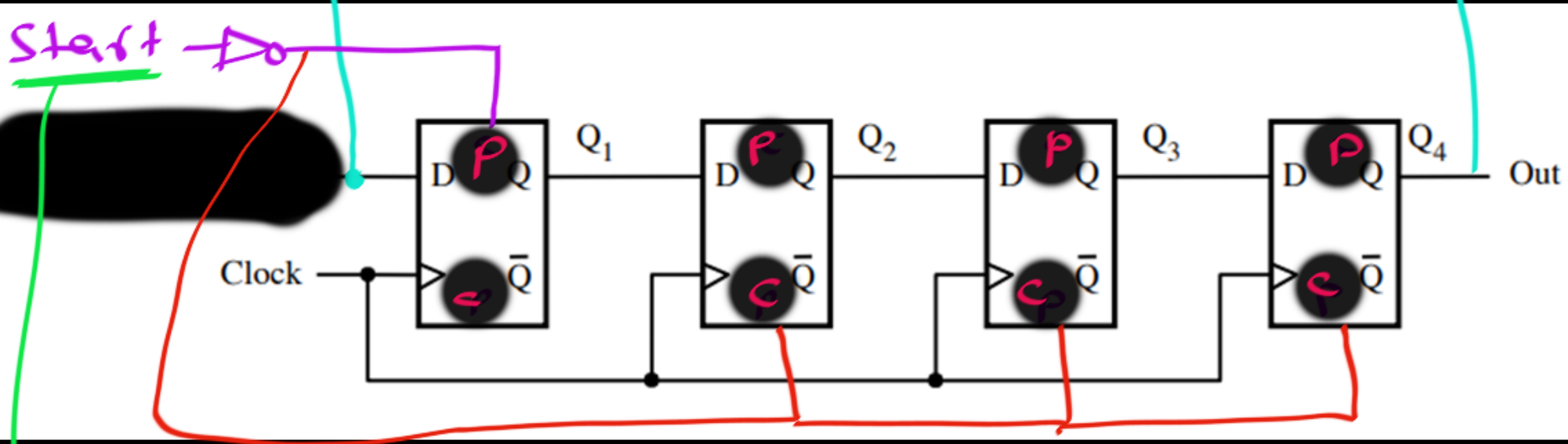
ff Contains Two Asynchronous inputs

① $\overline{clear}$

② $\overline{preset}$

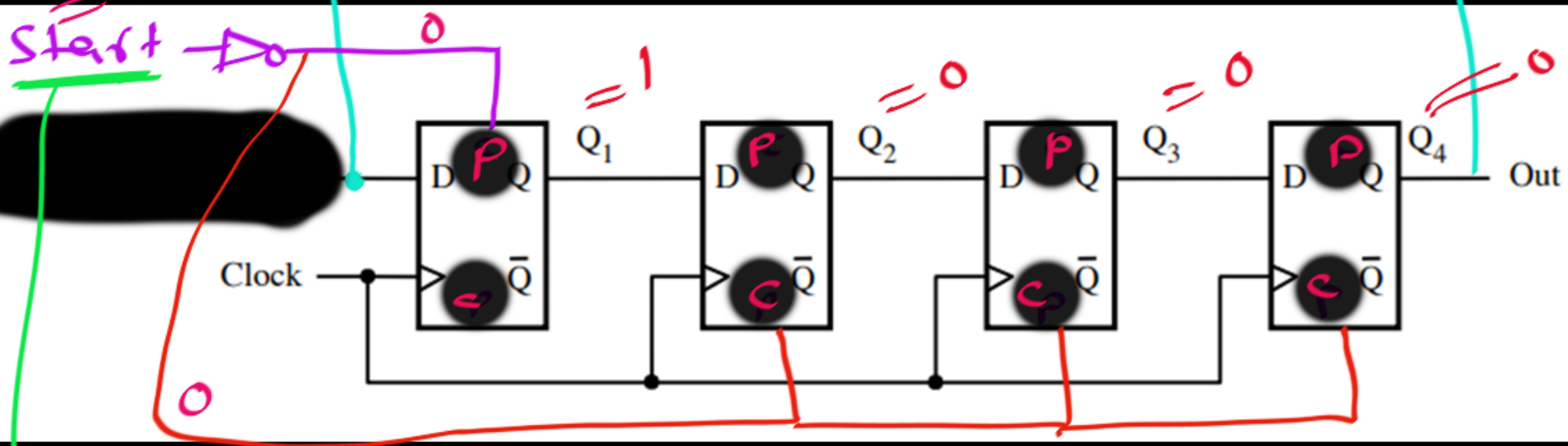
Active low inputs

Step 3 :→ initialize to 1000



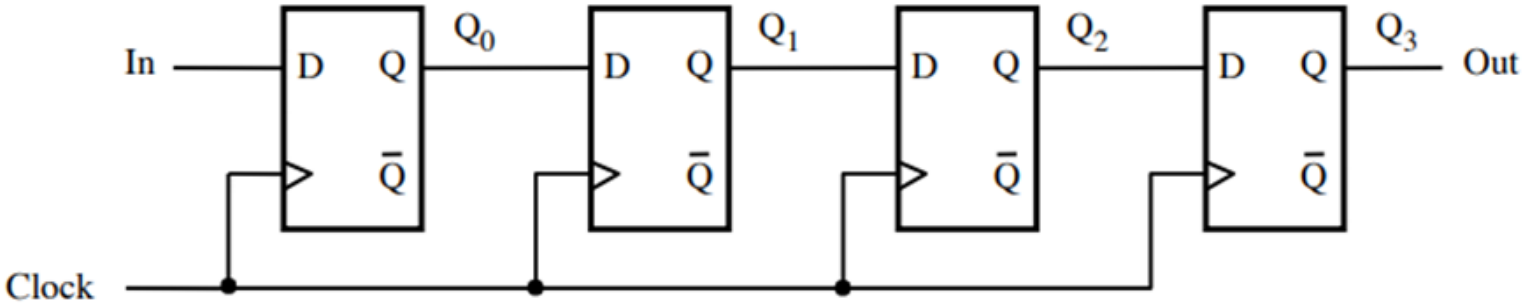
→ only purpose to Initialize (or re-start the Counting sequence)

Step 3 :→ initialize to 1000



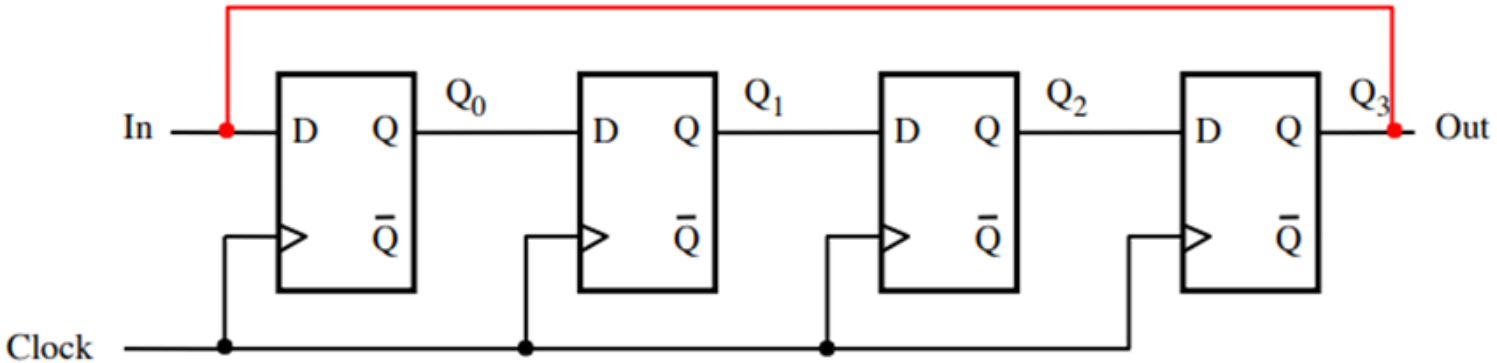
→ only purpose to Initialize (or re-start the Counting sequence)

How to build a 4-bit ring counter



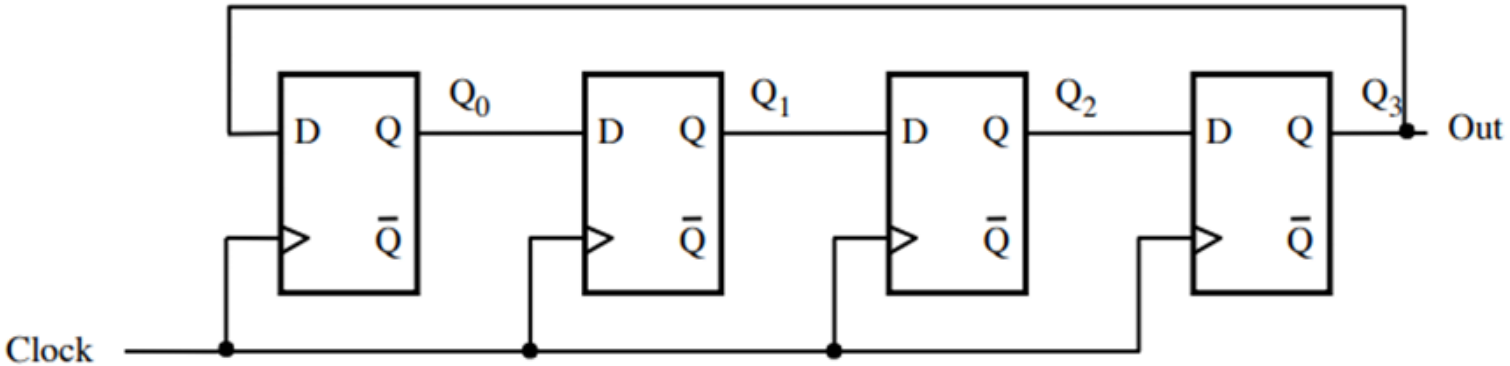
To build a ring counter we start with a shift register.

How to build a 4-bit ring counter



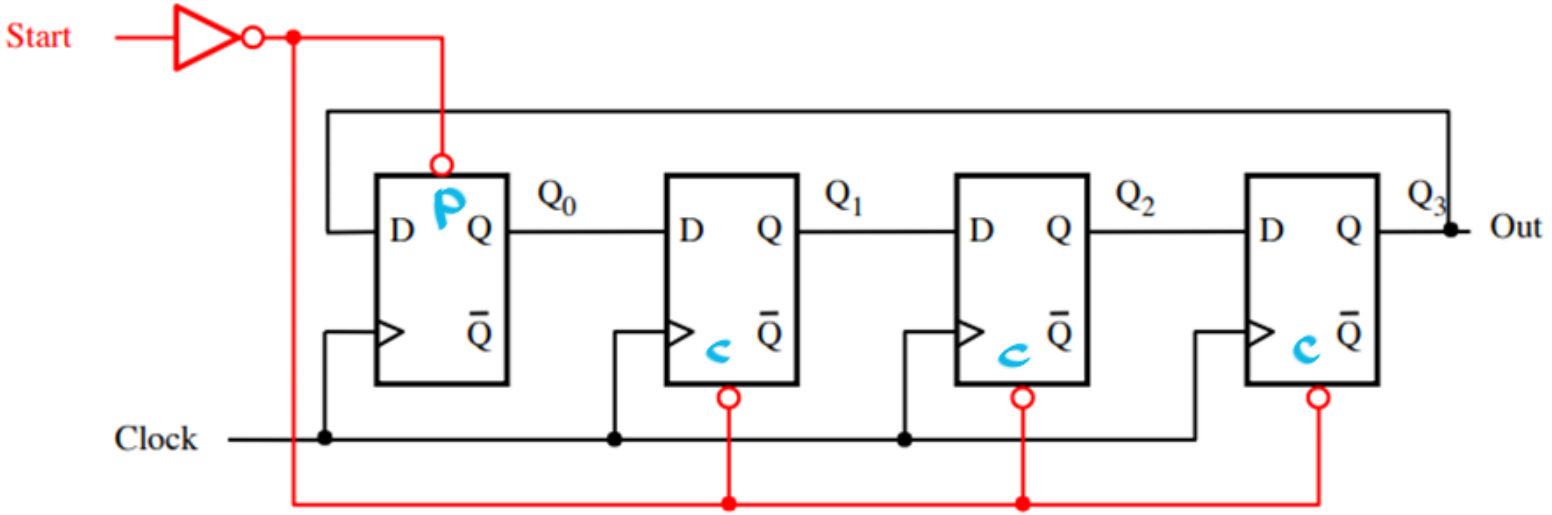
Next, add a loop from the last flip-flop to the first…

How to build a 4-bit ring counter



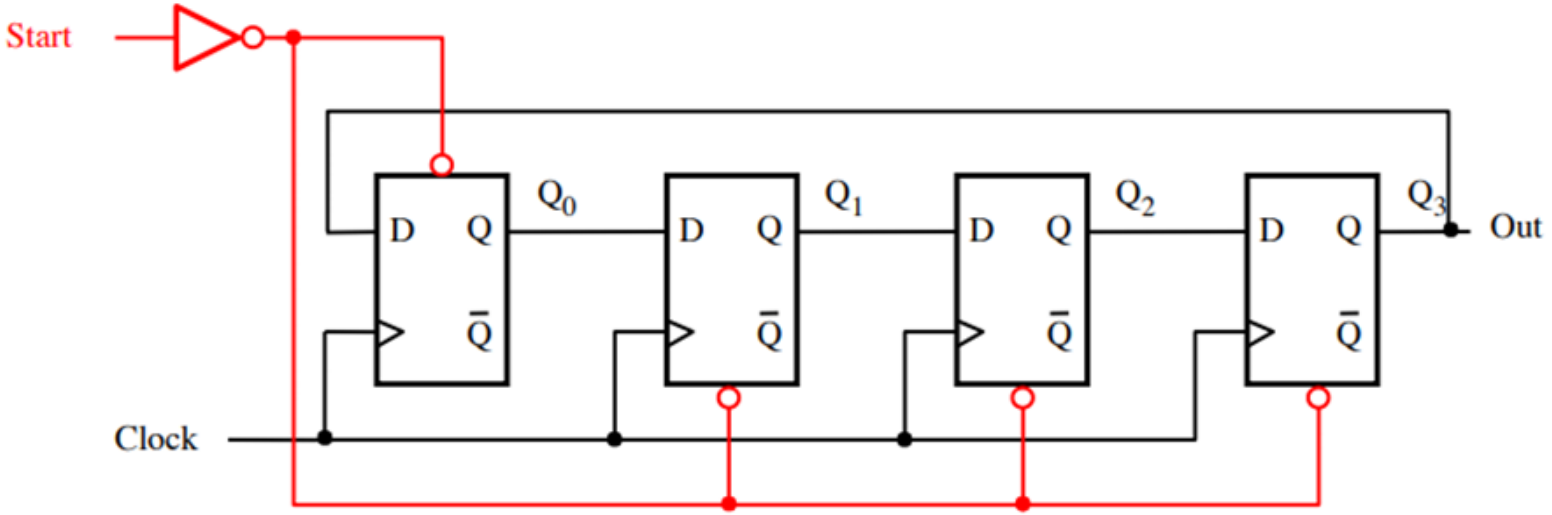
... and remove the In input line.

How to build a 4-bit ring counter



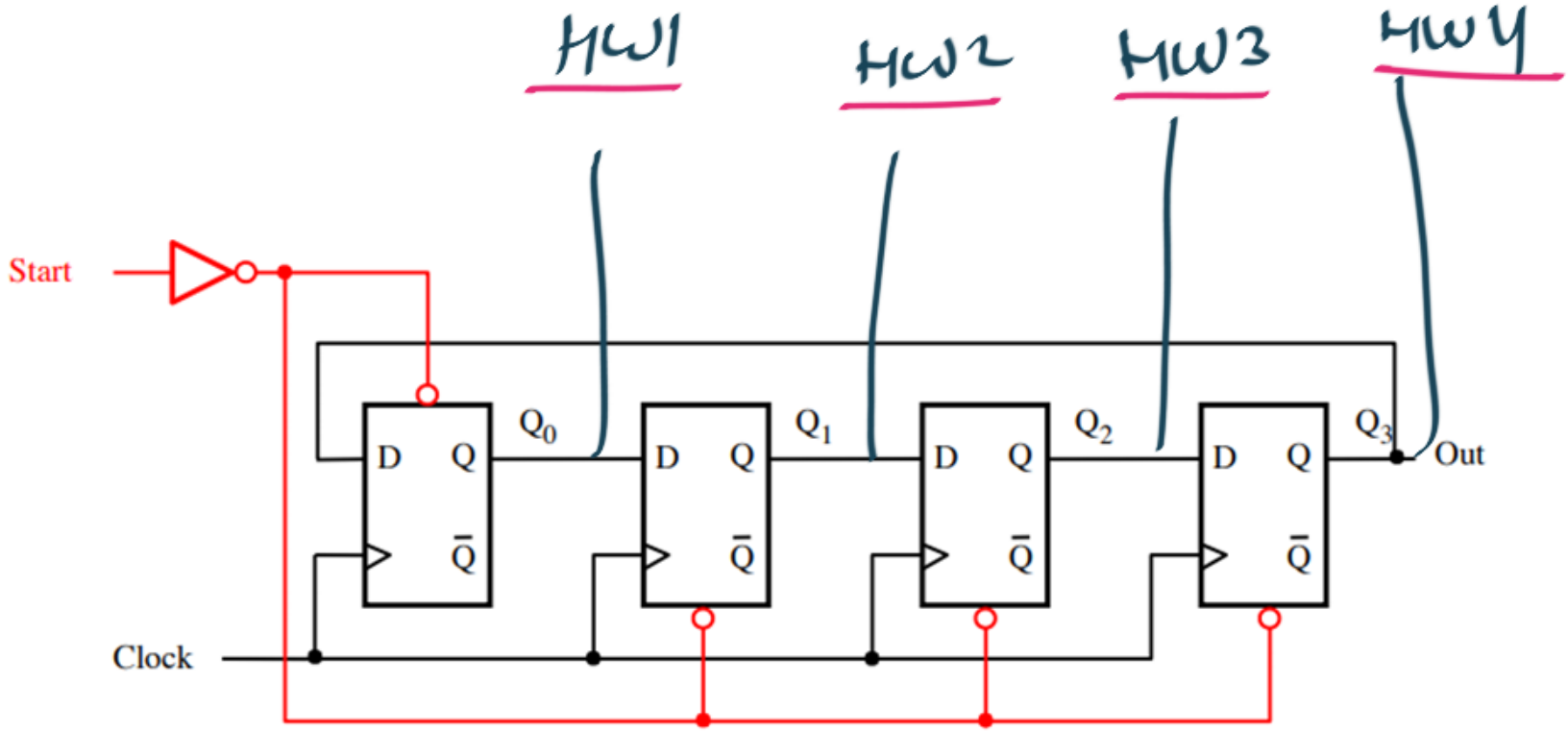
Also, add a start input that goes inverted to preset_n of the first flip=flop and to clear_n of all remaining flip-flops.

How to build a 4-bit ring counter



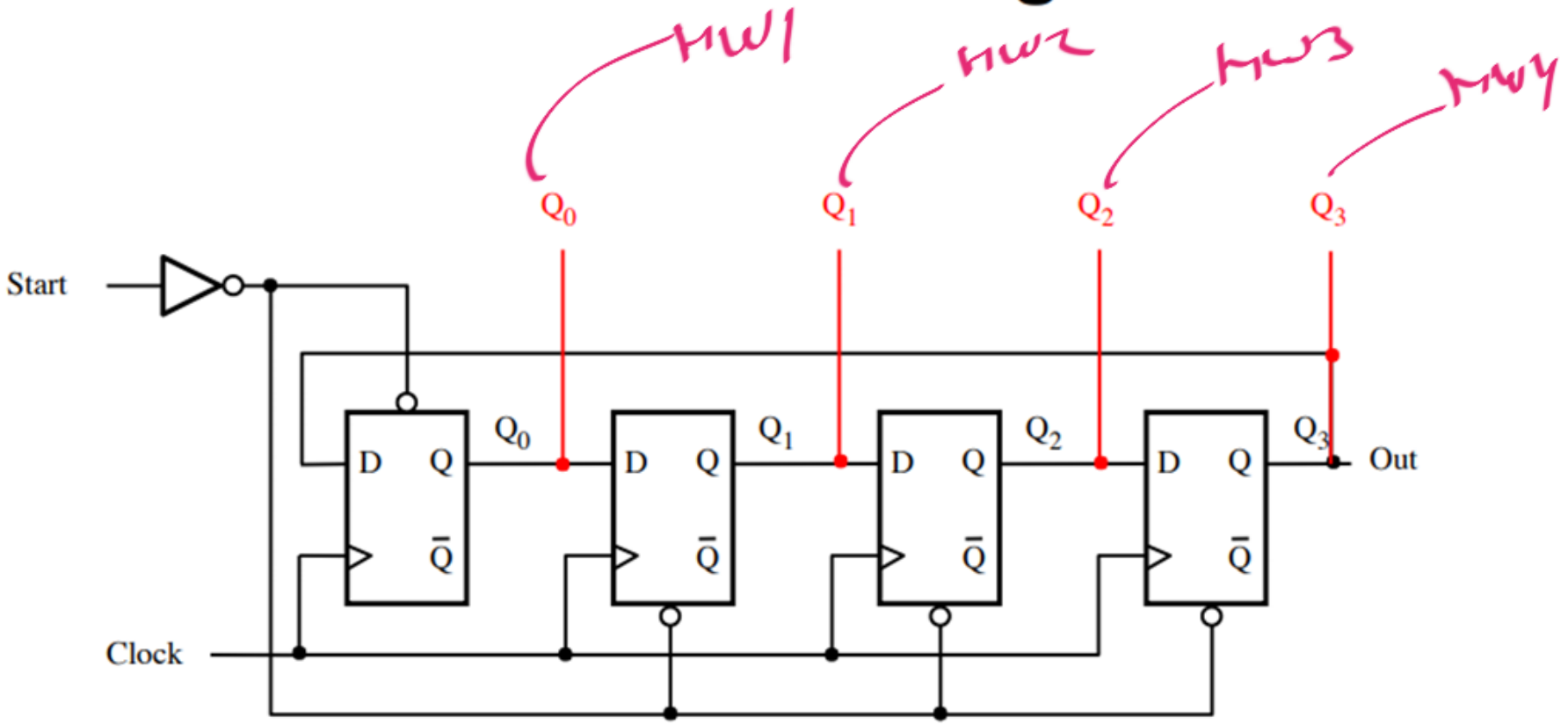
Also, add a start input that goes inverted to preset_n of the first flip=flop and to clear_n of all remaining flip-flops.

How to build a 4-bit ring counter



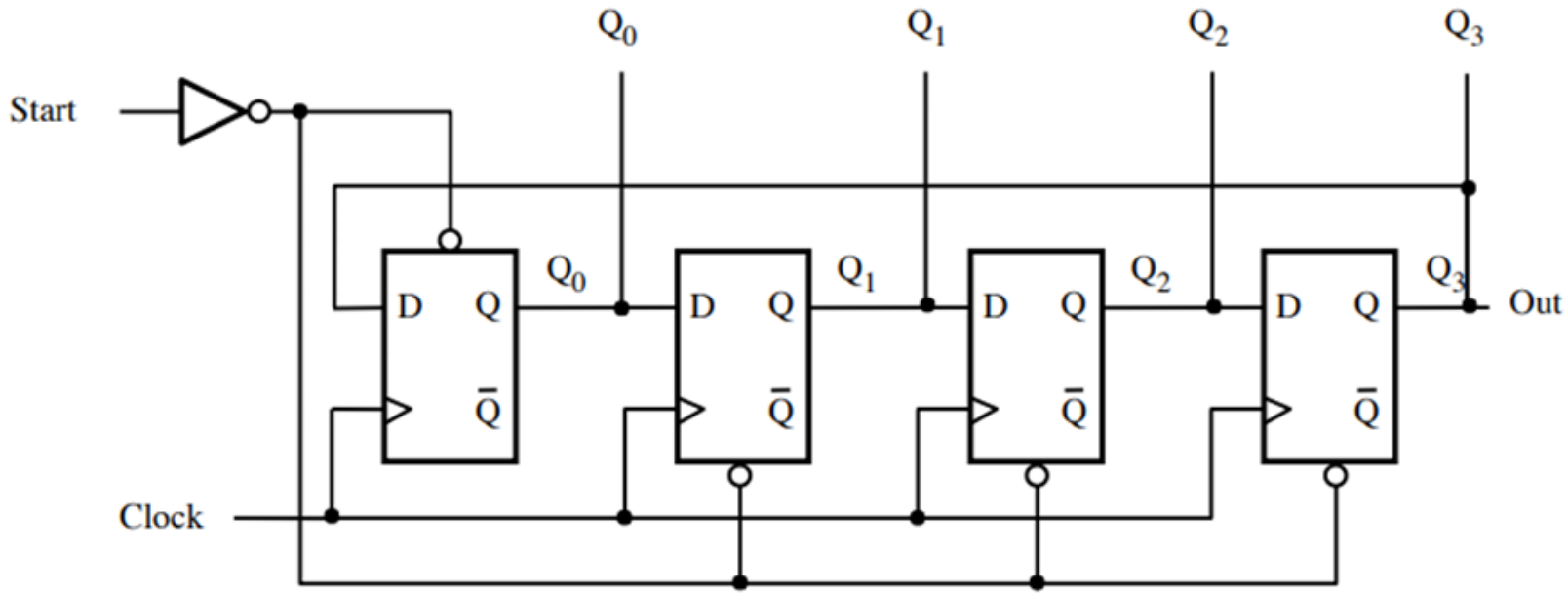
Also, add a start input that goes inverted to preset_n of the first flip=flop and to clear_n of all remaining flip-flops.

How to build a 4-bit ring counter



Finally, extend the output lines that form the count number.

How to build a 4-bit ring counter



This is the final circuit diagram.

4-bit ring counter

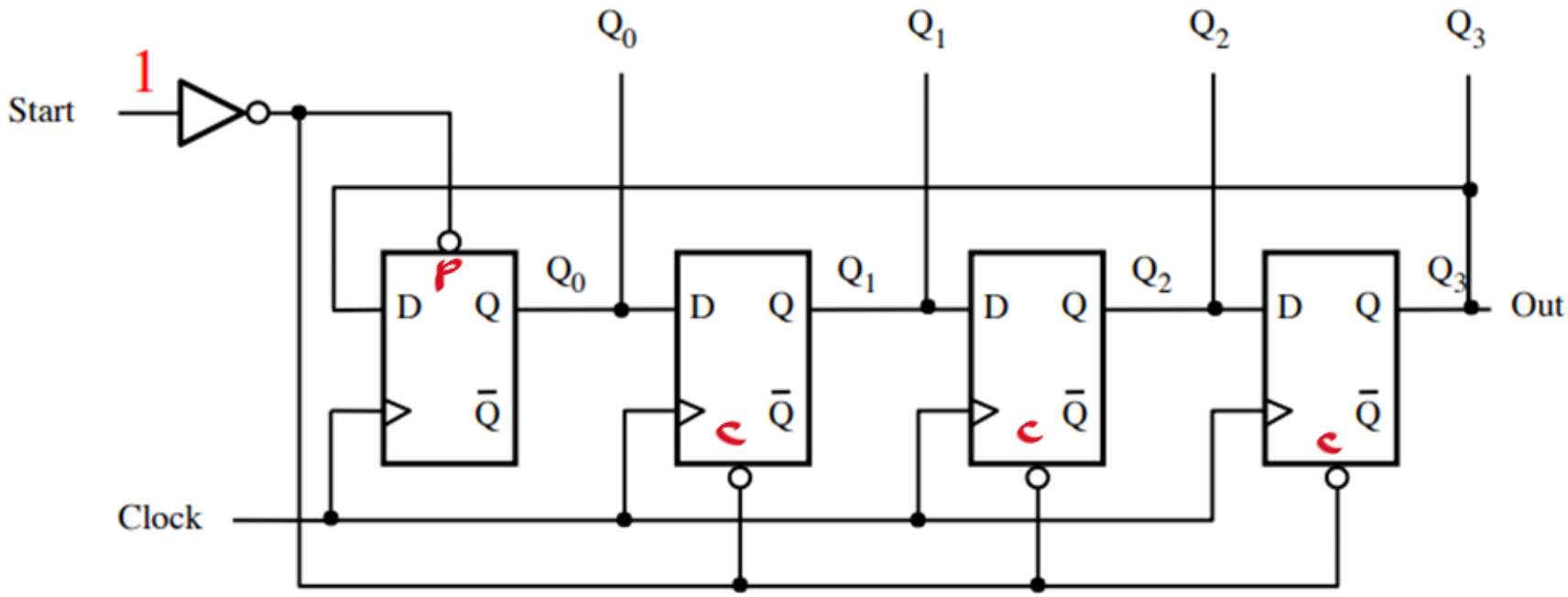
- There is only one 1 on the outputs of the four flip-flops
- The counting sequence is: 1000, 0100, 0010, 0001, 1000, …
- To reset the counter
  - set start to 1 for a short period of time
  - This sets the four outputs to 1000

A shift register is constructed using D-type flip-flops where the output of one flip-flop is connected to the input of another flip-flop.

With ring counters, the output of the last flip-flop is fed to the input of the first flip-flop.

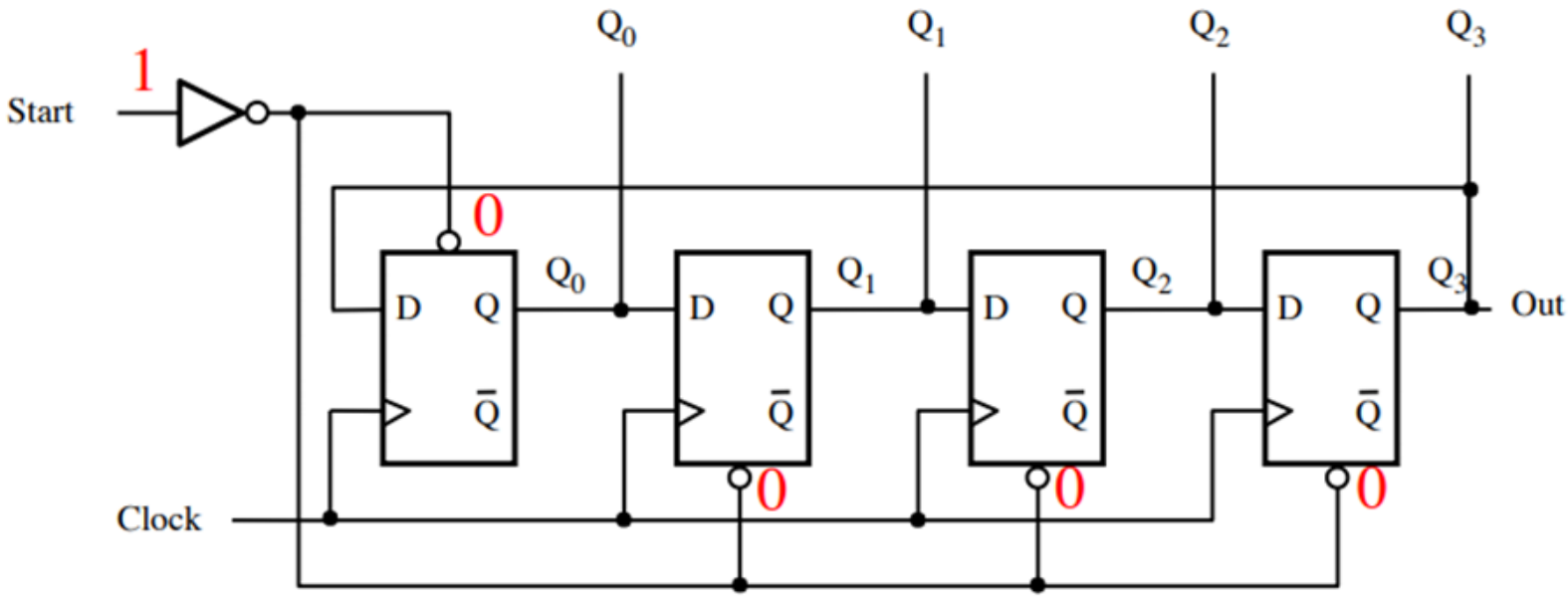
Ring Counter: Implementation 1: Using Shift register Working

4-bit ring counter: How does it work



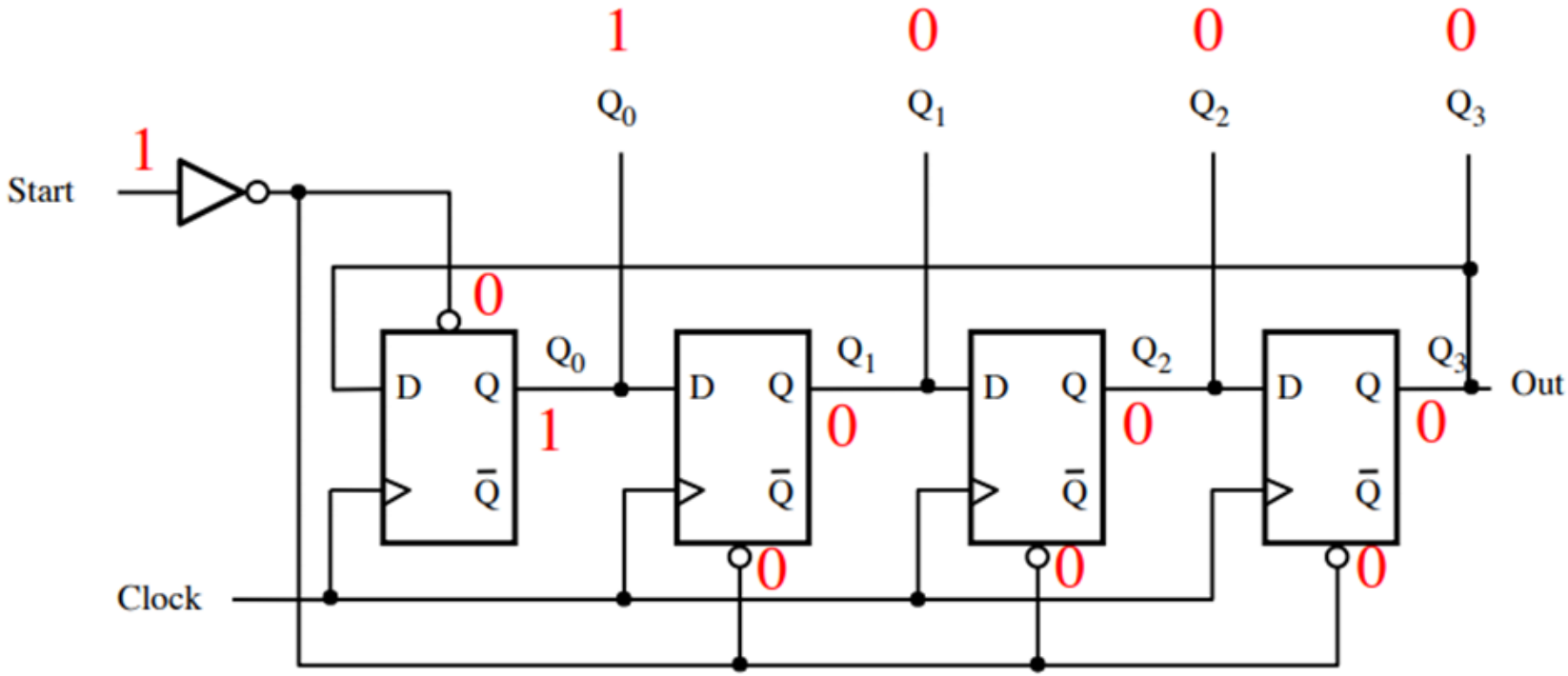
To initialize the counter set Start to 1.

4-bit ring counter: How does it work



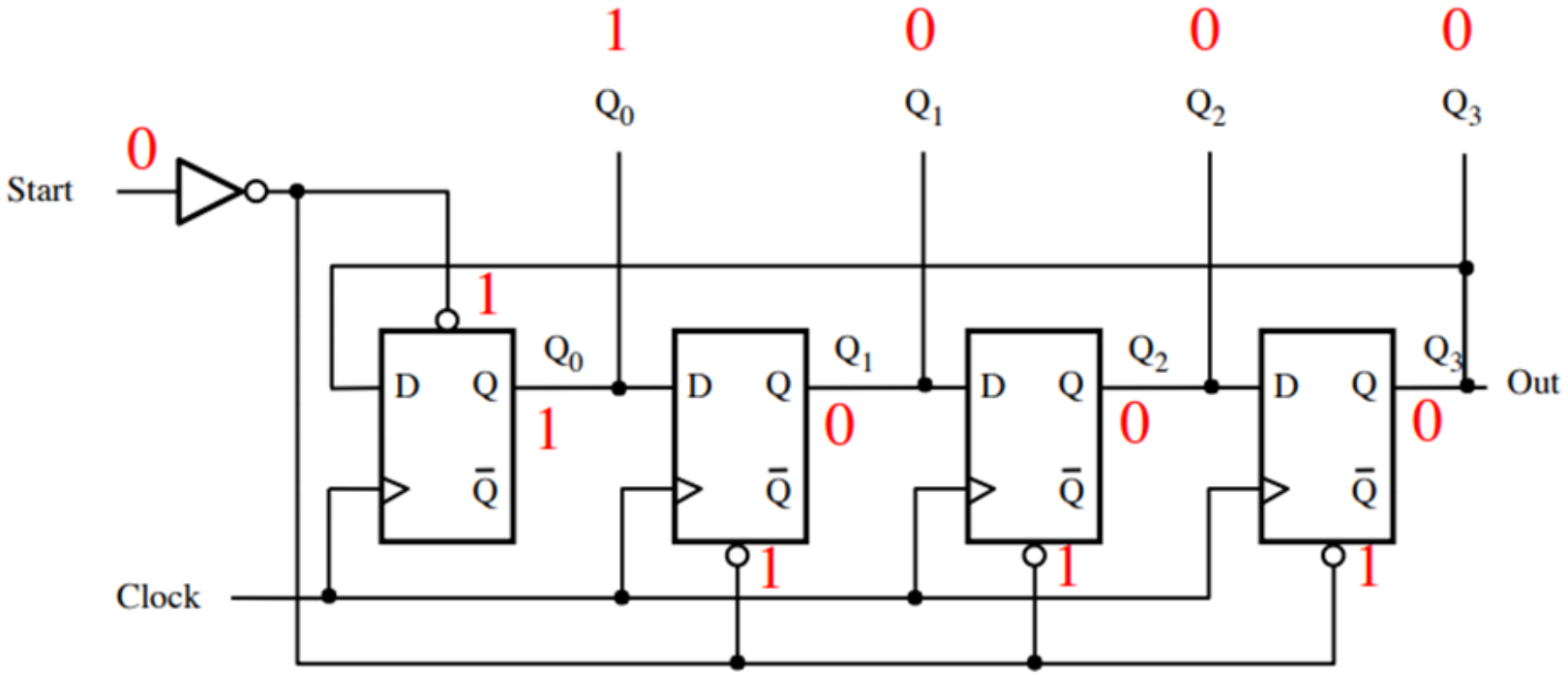
After the NOT gate, this 1 goes as 0 to preset_n of the first flip-flop and to clear_n of all remaining flip-flops.

4-bit ring counter: How does it work



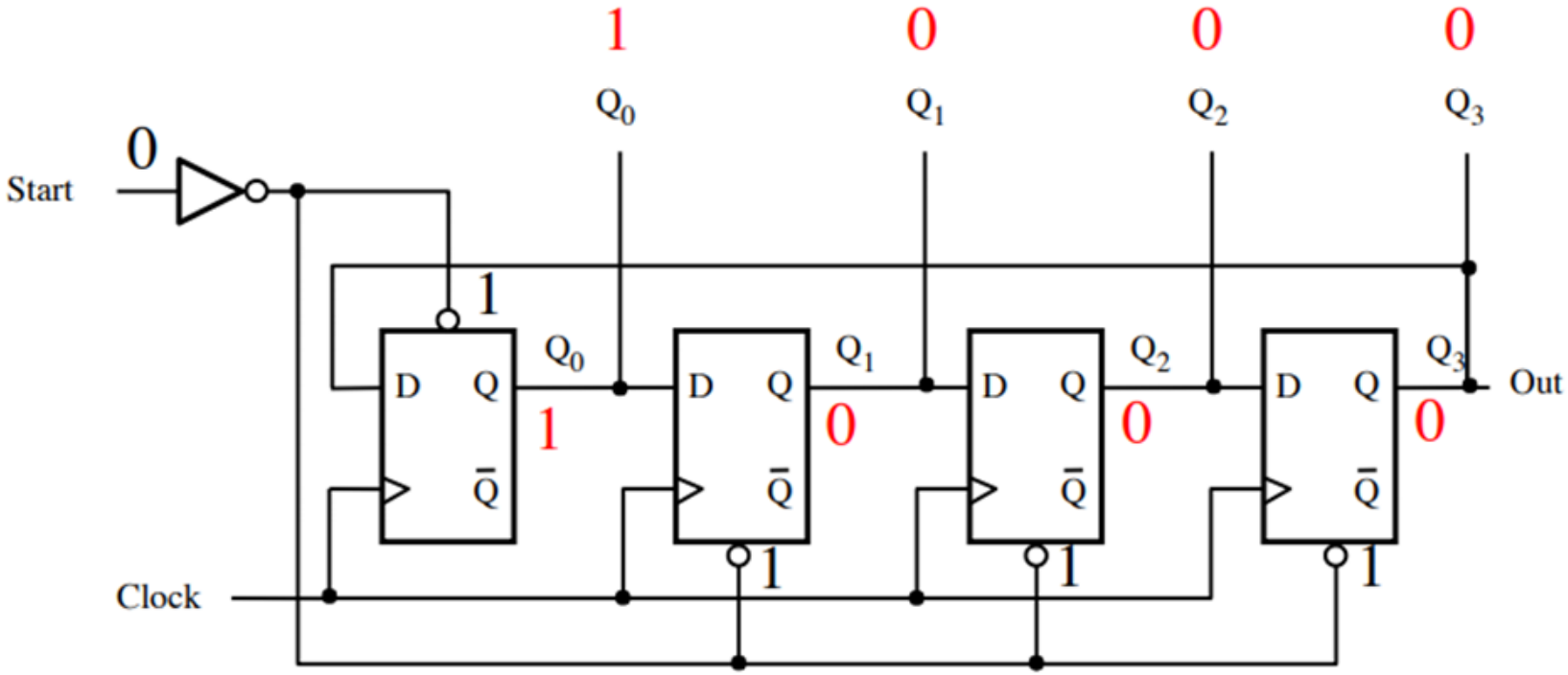
This sets the output pattern to 1000,
i.e., only the first bit is one and the rest are zeros.

4-bit ring counter: How does it work



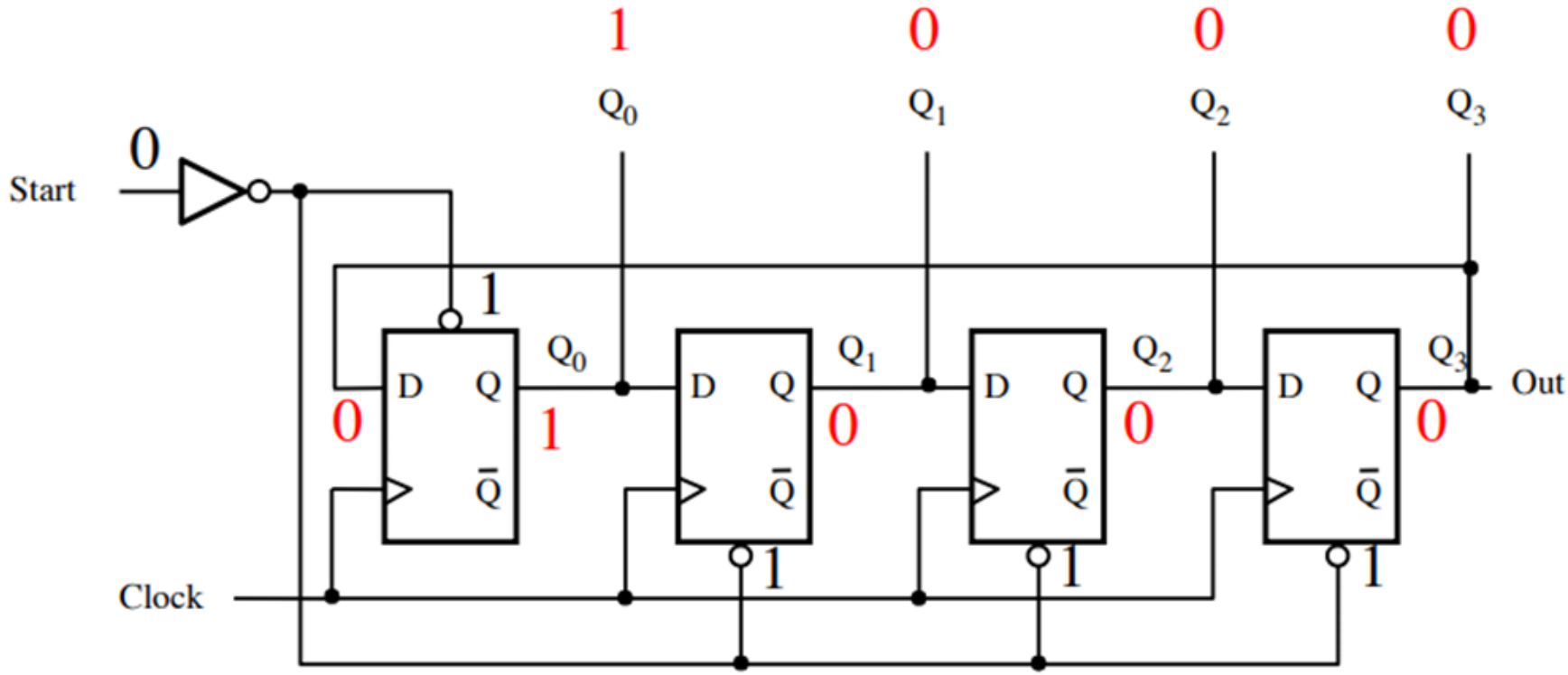
Setting Start to 0 has no effect on the outputs,
because both preset_n and clear_n are sensitive only to 0.

4-bit ring counter: How does it work



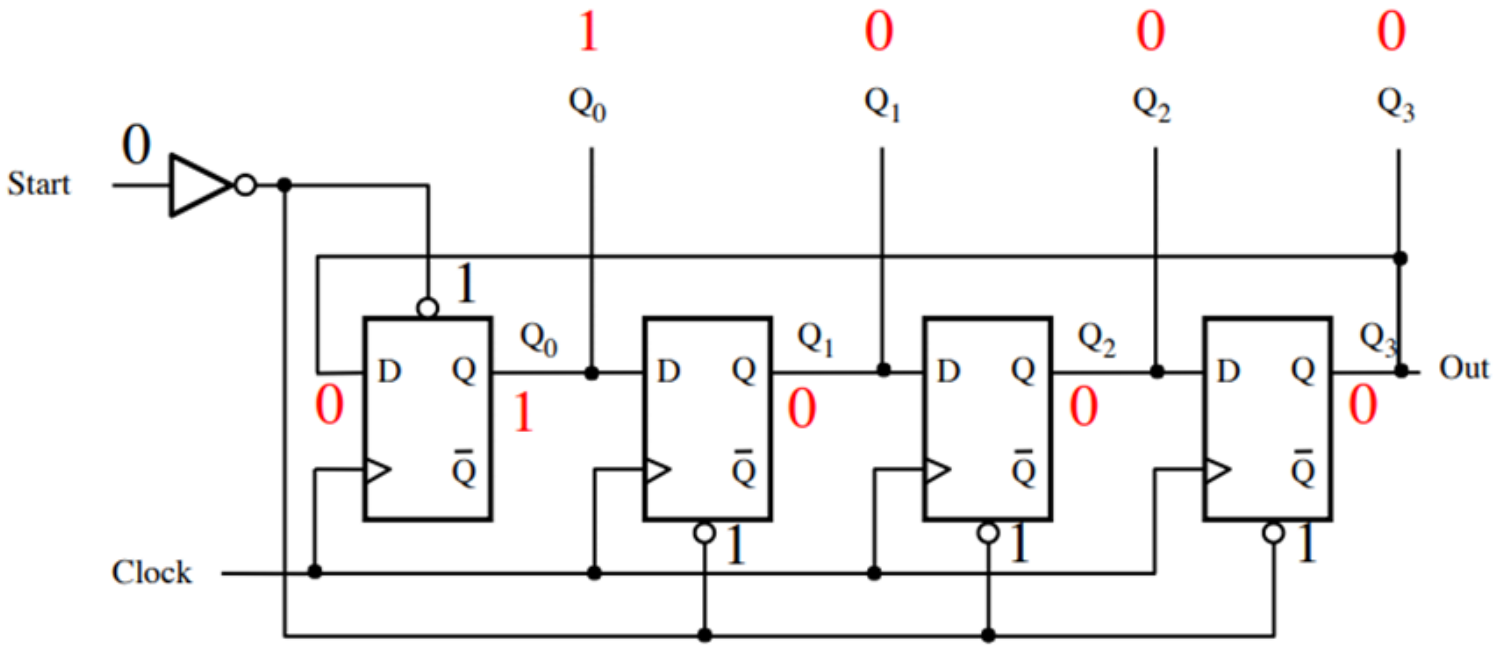
The initialization does not depend on the clock since both preset_n and clear_n bypass the gates of the latches in the flip-flops.

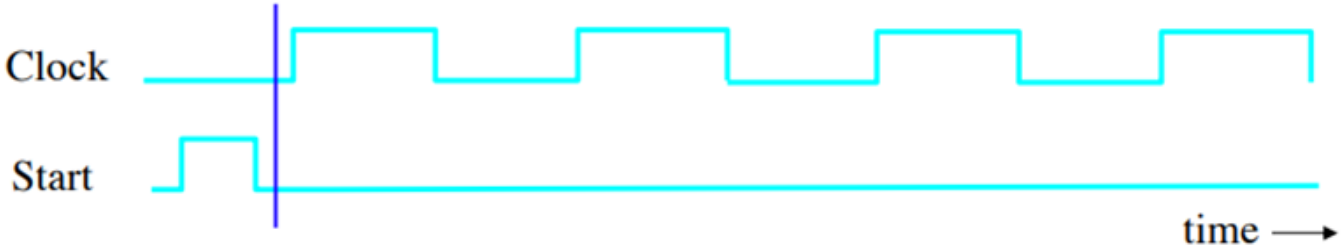
4-bit ring counter: How does it work



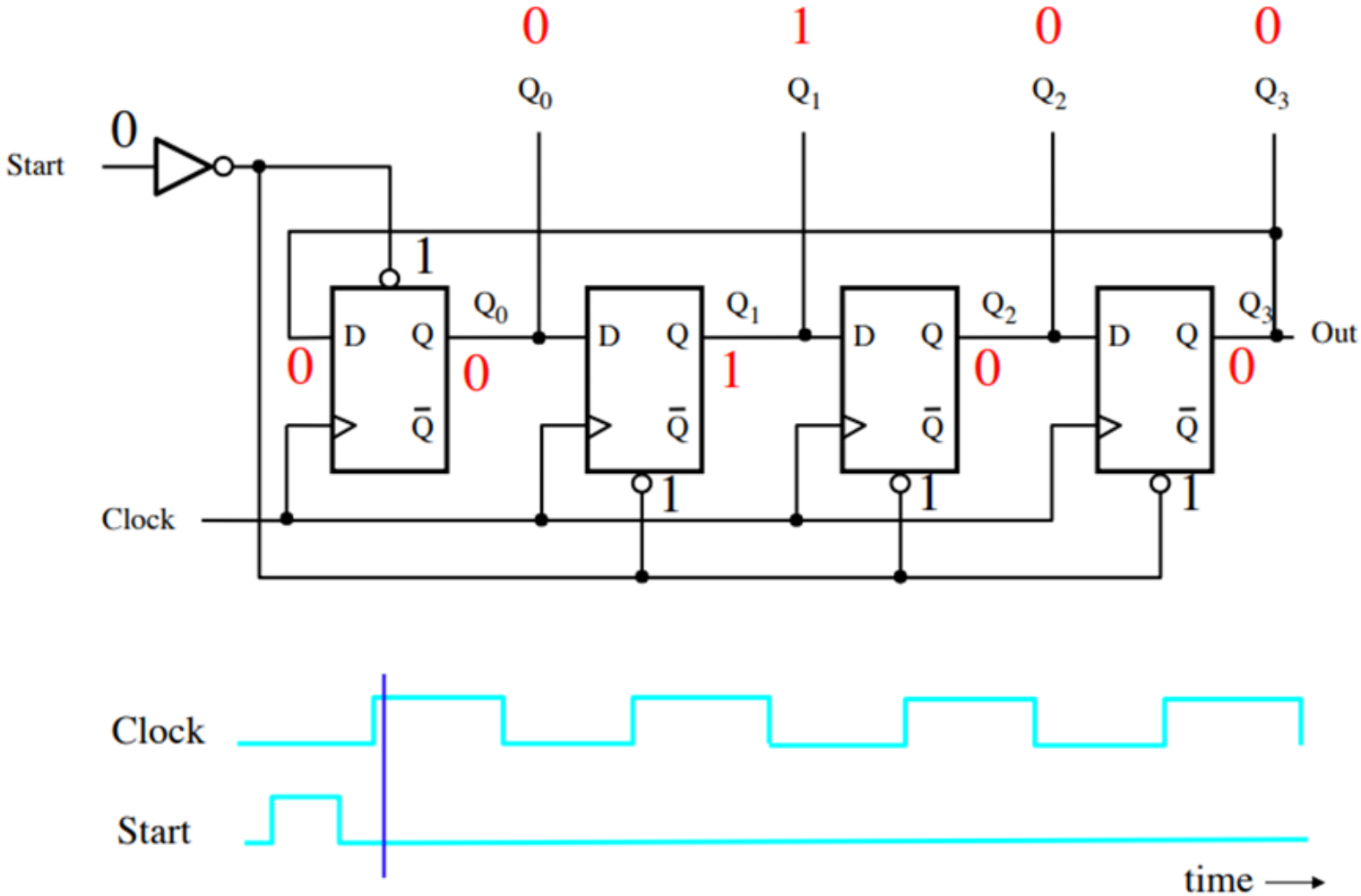
That last 0 loops back to the D input of the first flip-flop.

4-bit ring counter: How does it work

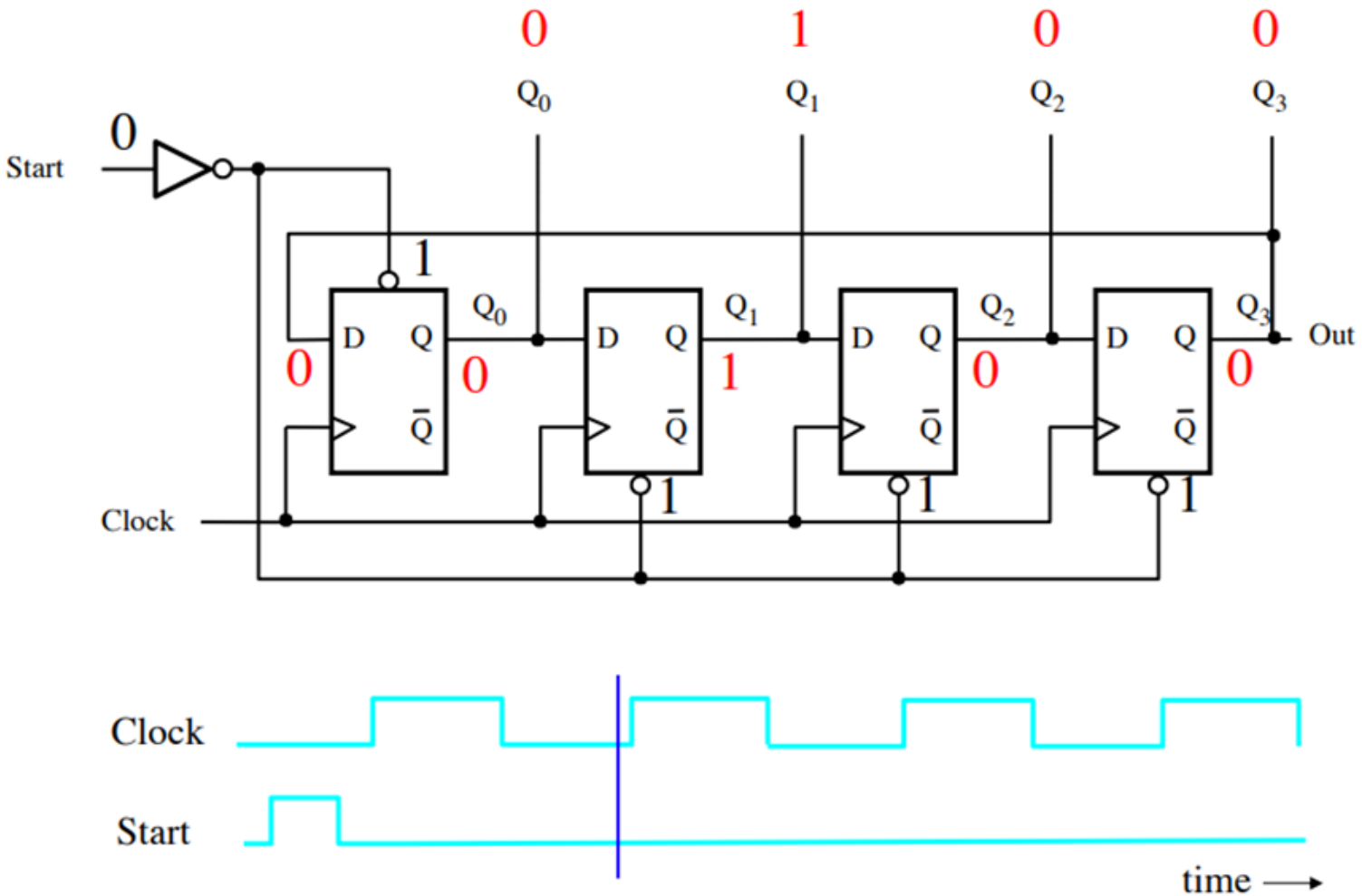




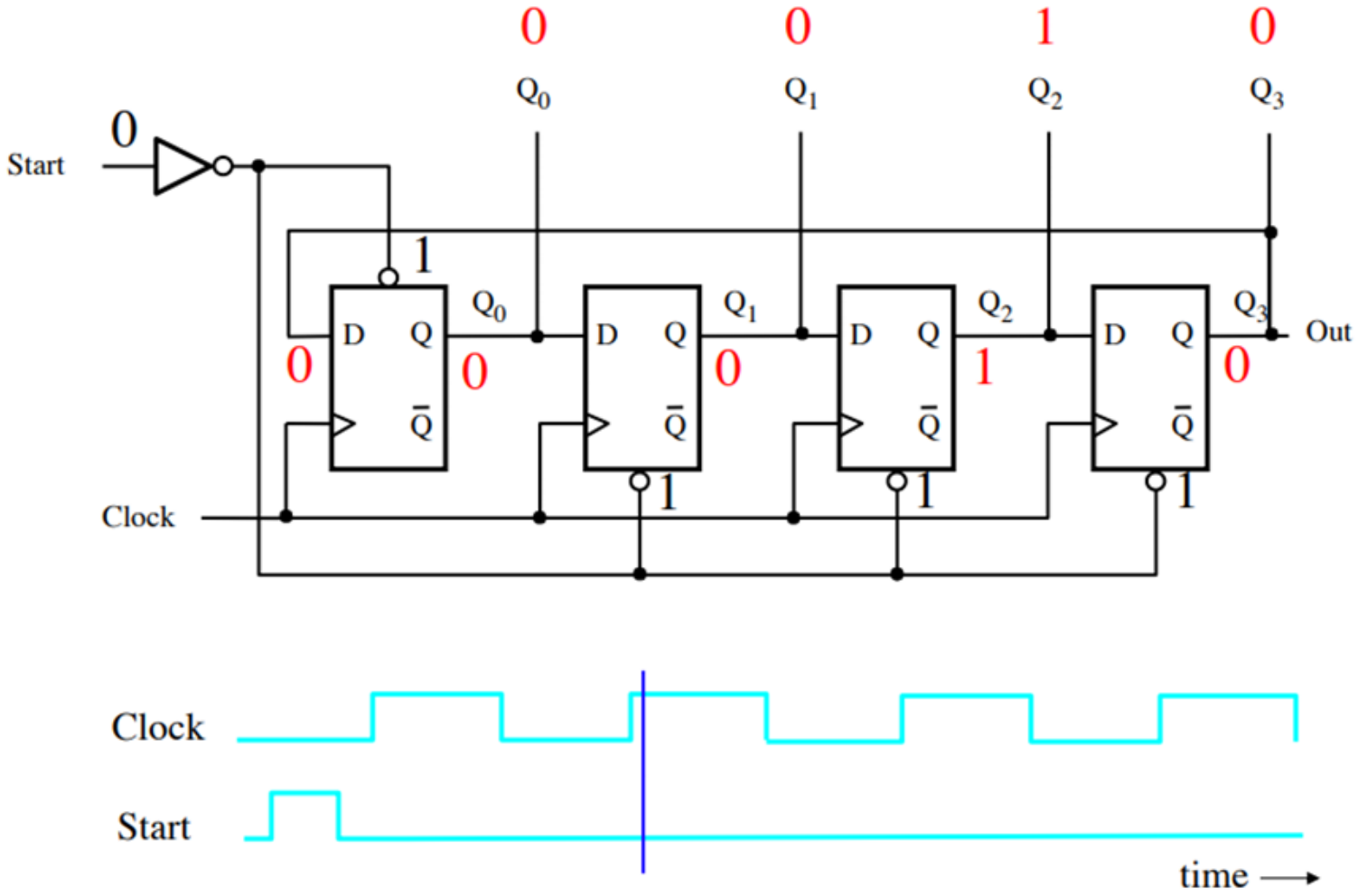
4-bit ring counter: How does it work



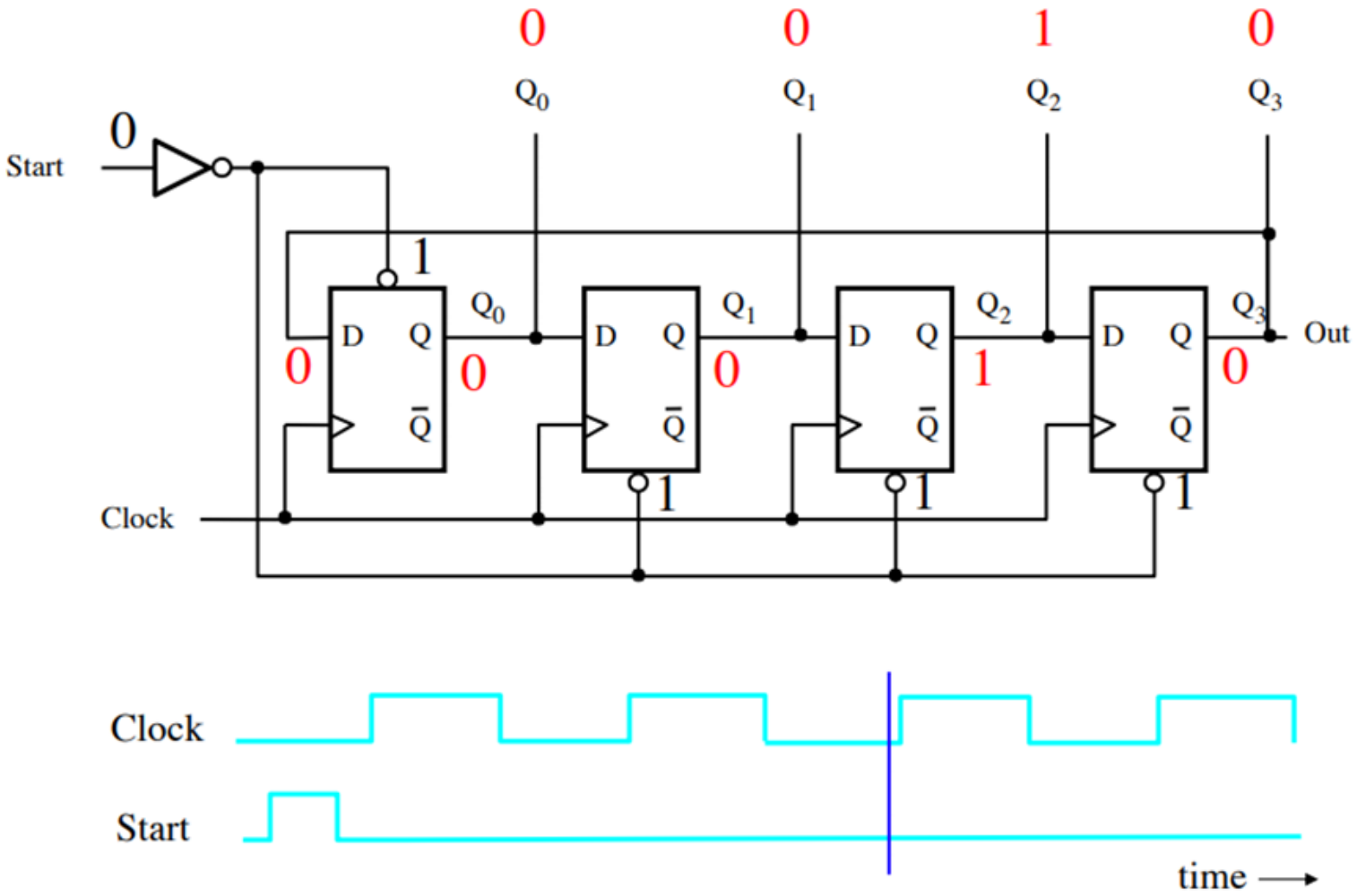
4-bit ring counter: How does it work



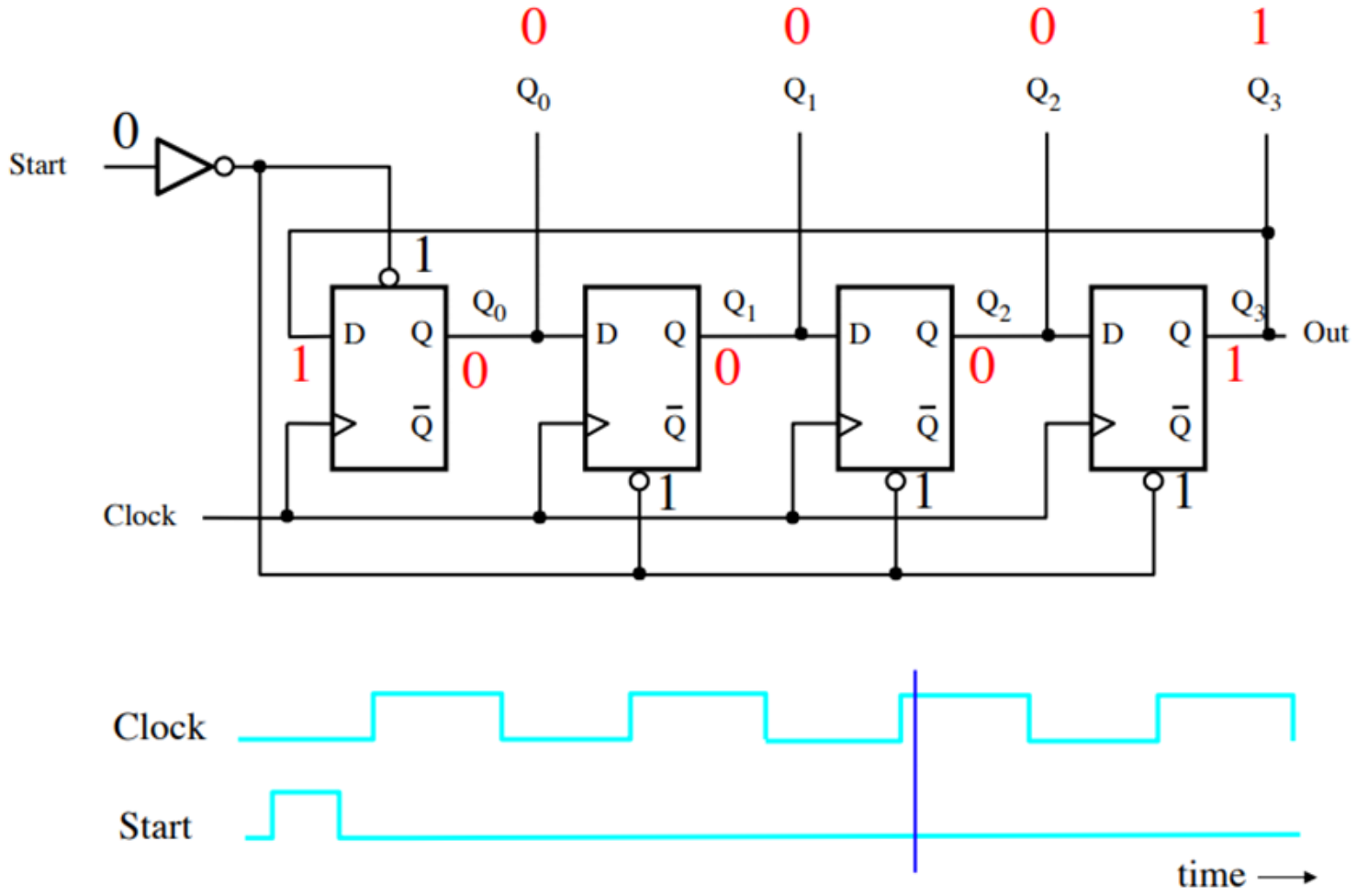
4-bit ring counter: How does it work



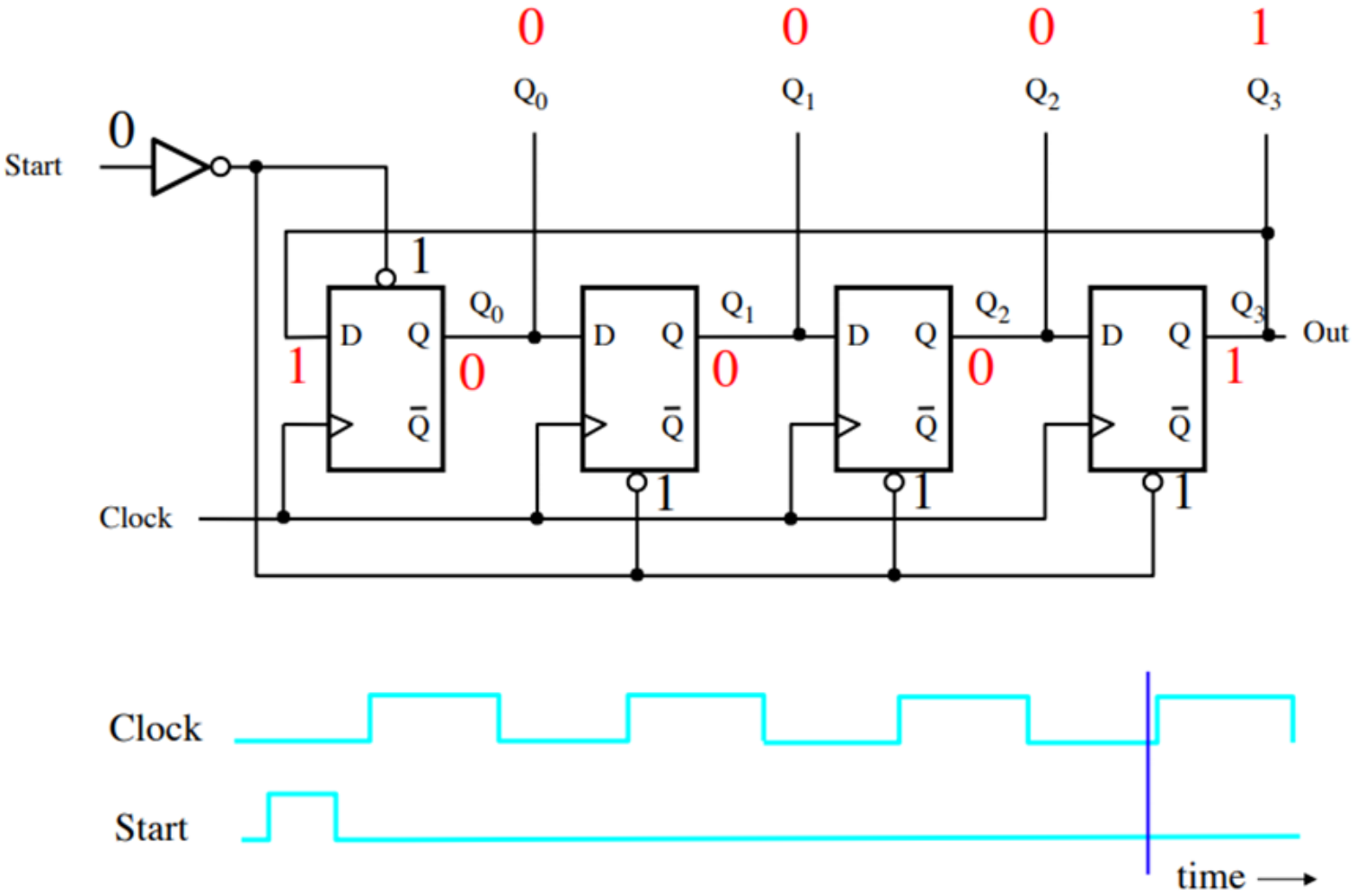
4-bit ring counter: How does it work



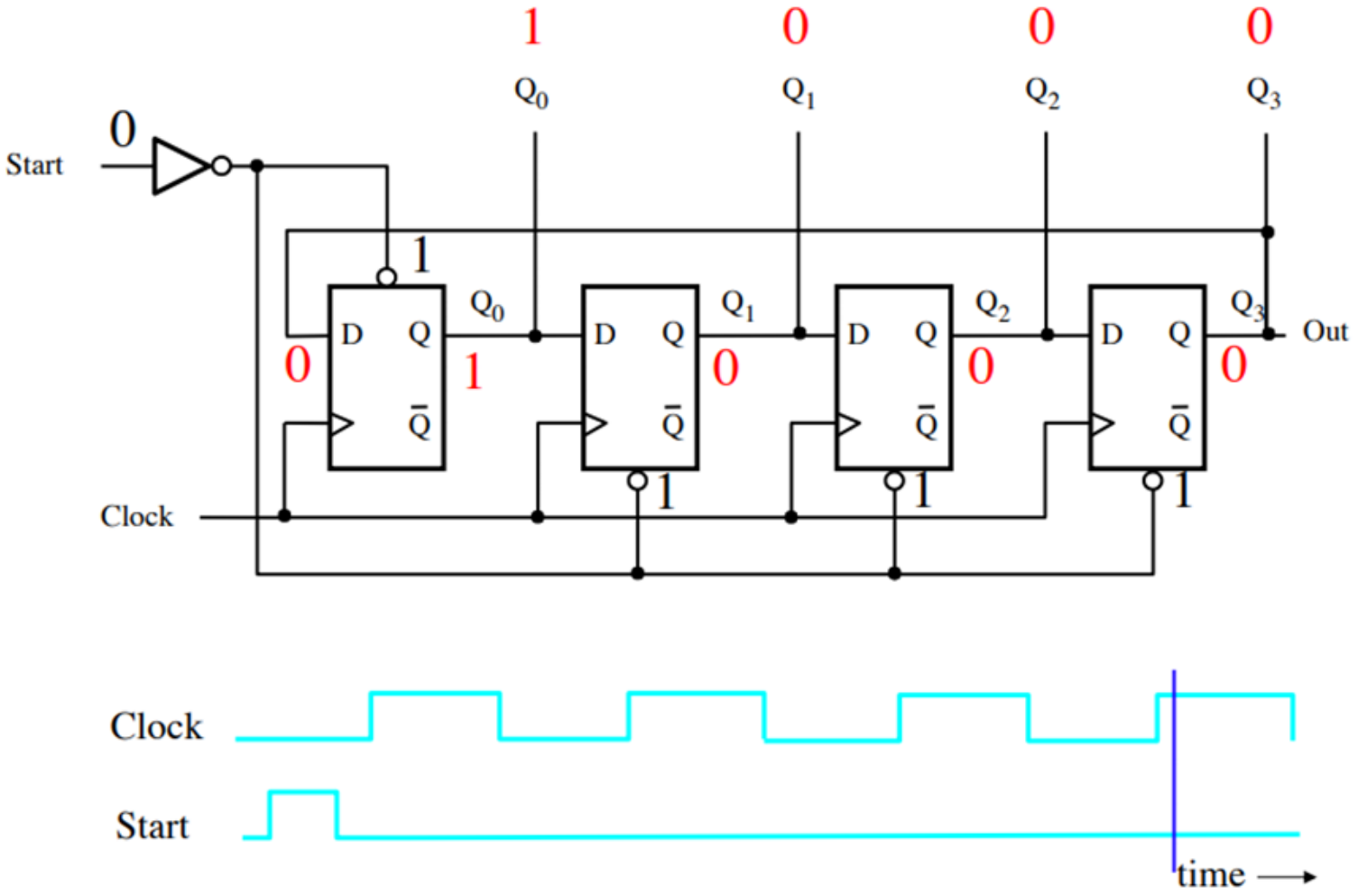
4-bit ring counter: How does it work



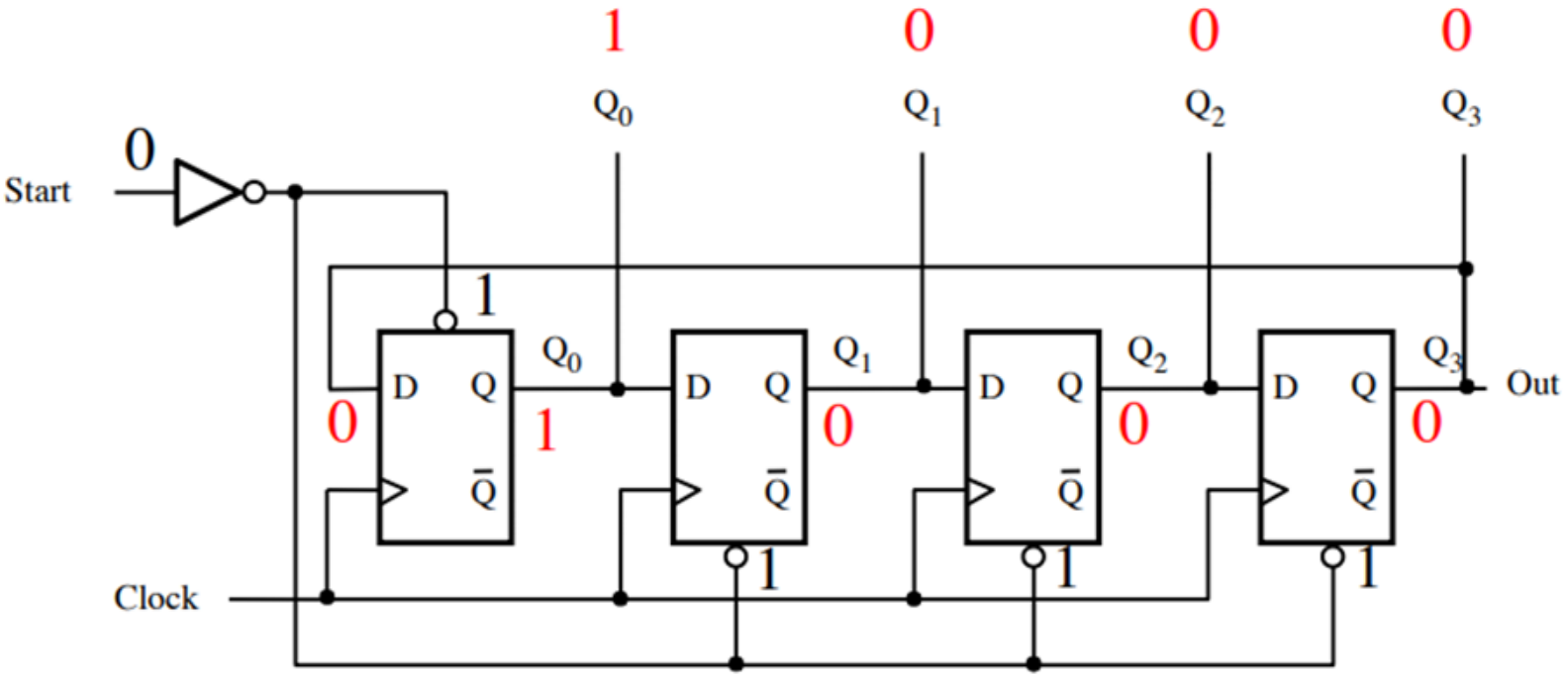
4-bit ring counter: How does it work



4-bit ring counter: How does it work

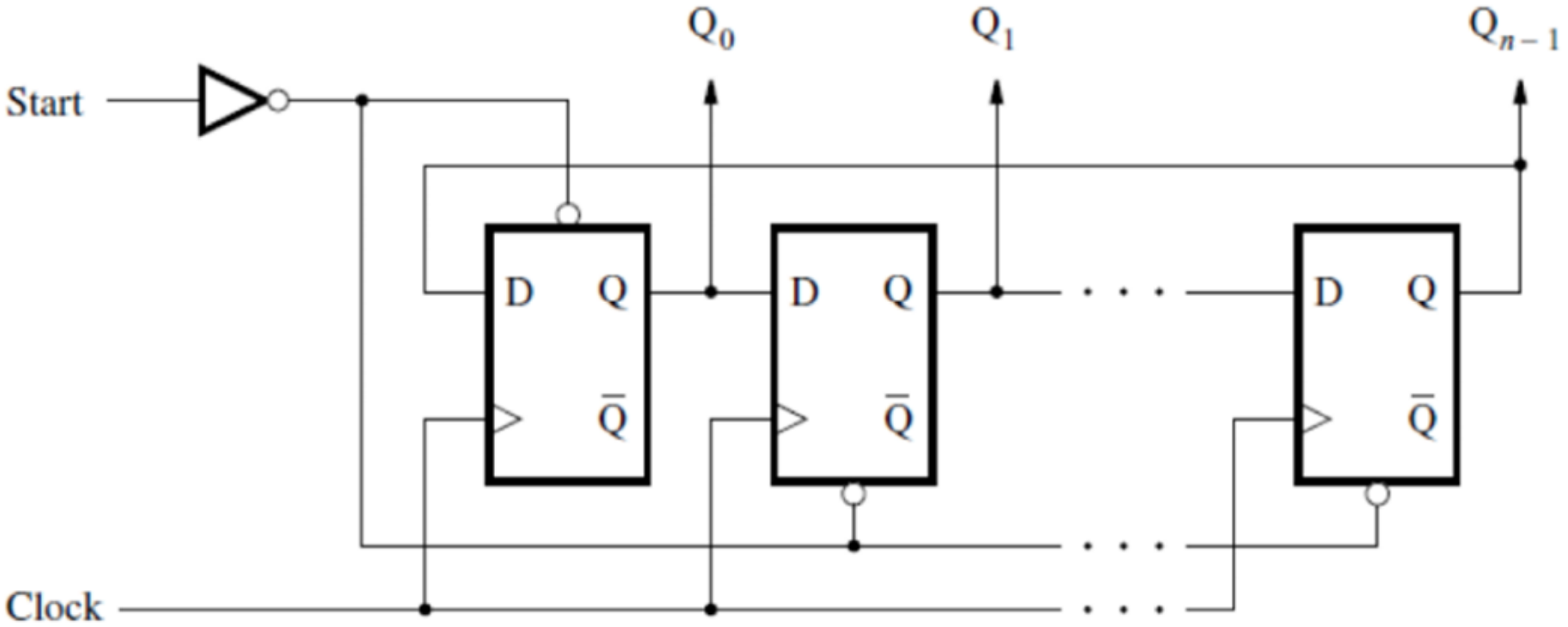


4-bit ring counter: How does it work

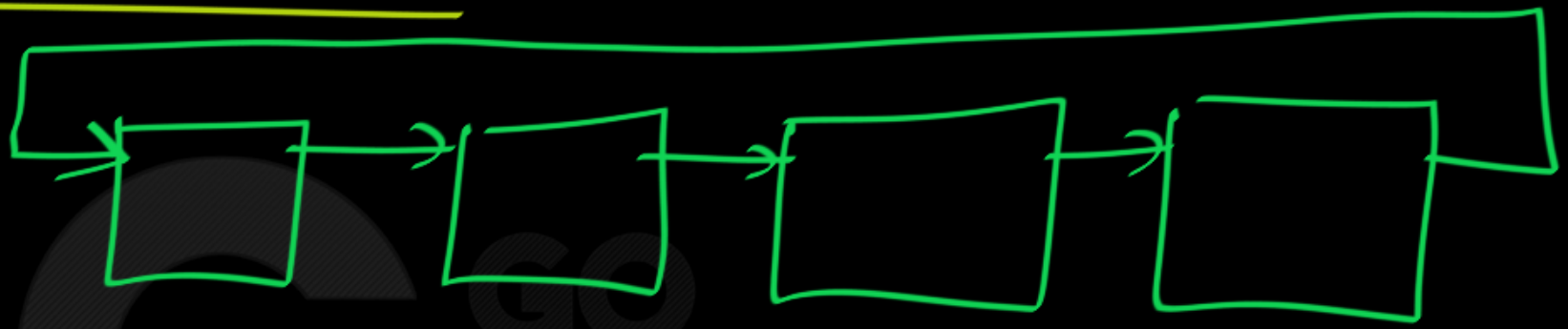


It is back to the start of the counting sequence, which is: 1000, 0100, 0010, 0001.

n-bit ring counter

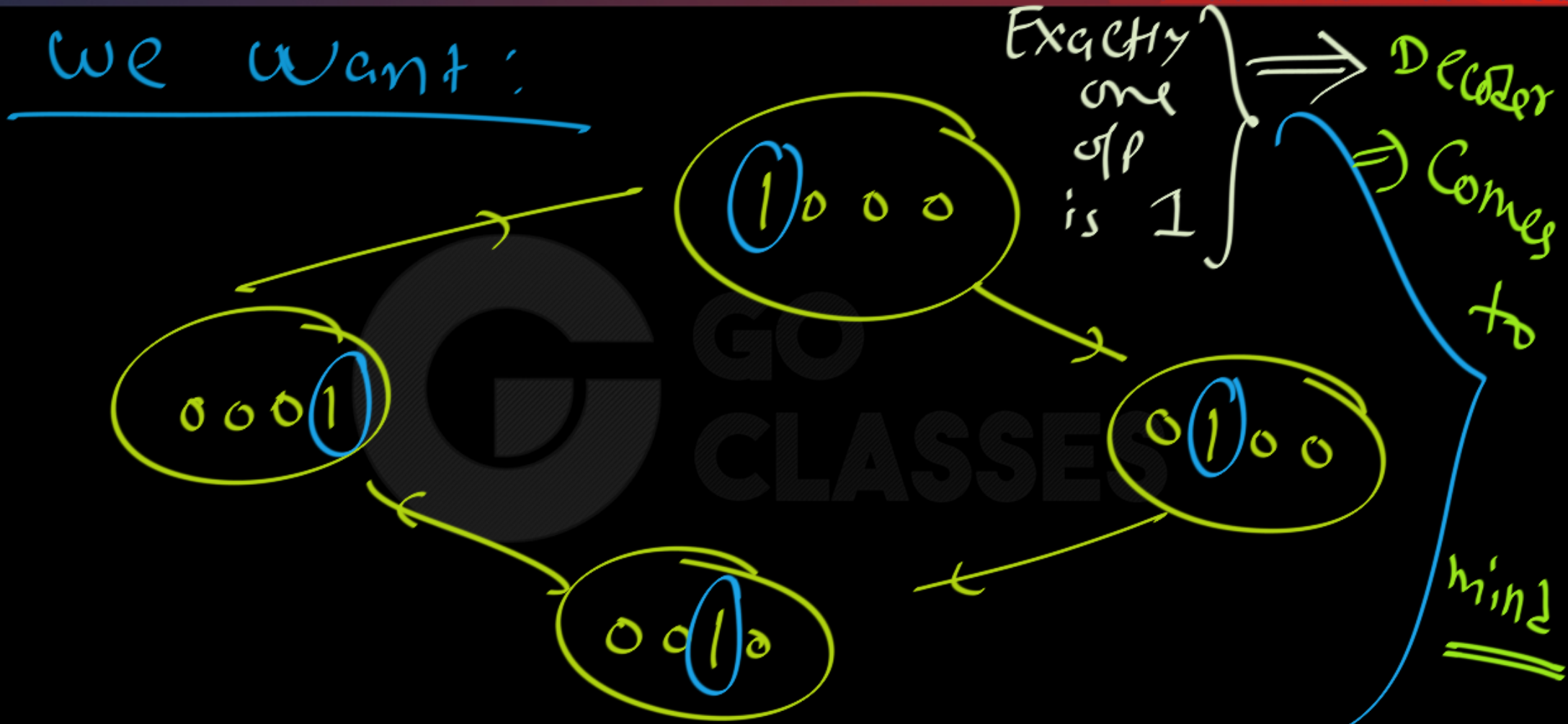


4-bit Ring Counter : ⟶ 4 - flipflops



n - bit Ring Counter using Shift Register ⇒ n - flipflops

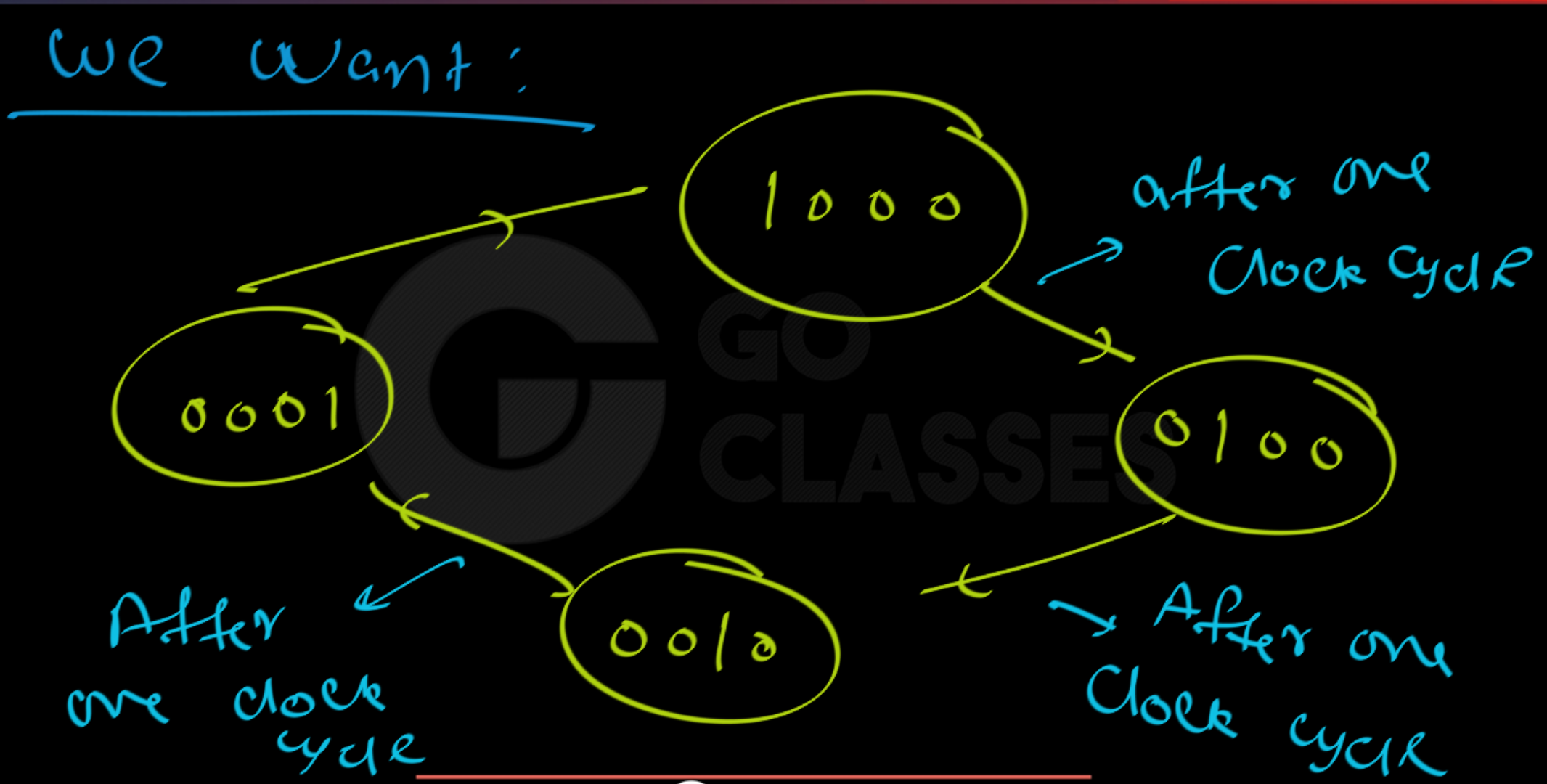
Ring Counter: Implementation 2: Using a Counter with a Decoder


We want :
Exactly one op is 1
Decoder
Comes
to
mind
1000
0001
0100
0010

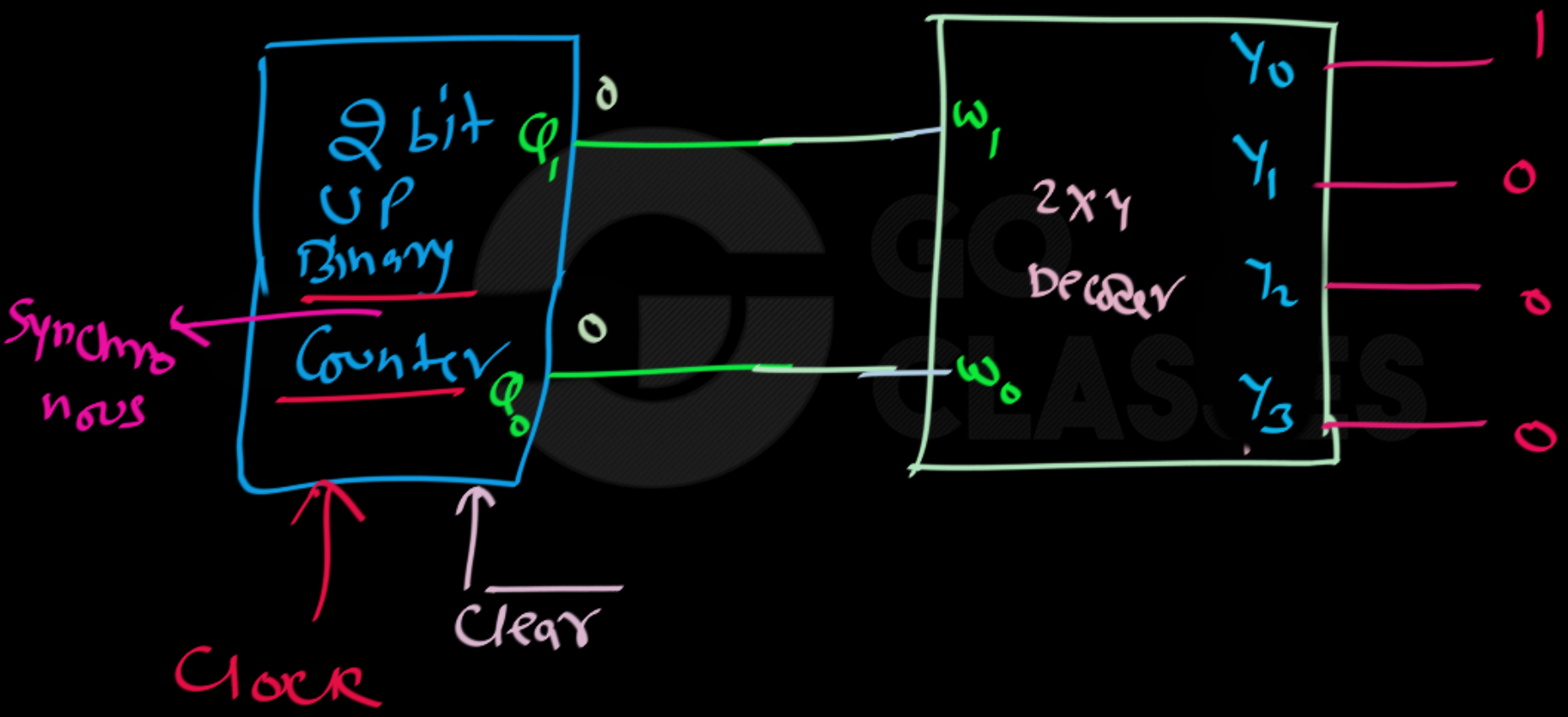
But Decoder is Combinational ckt

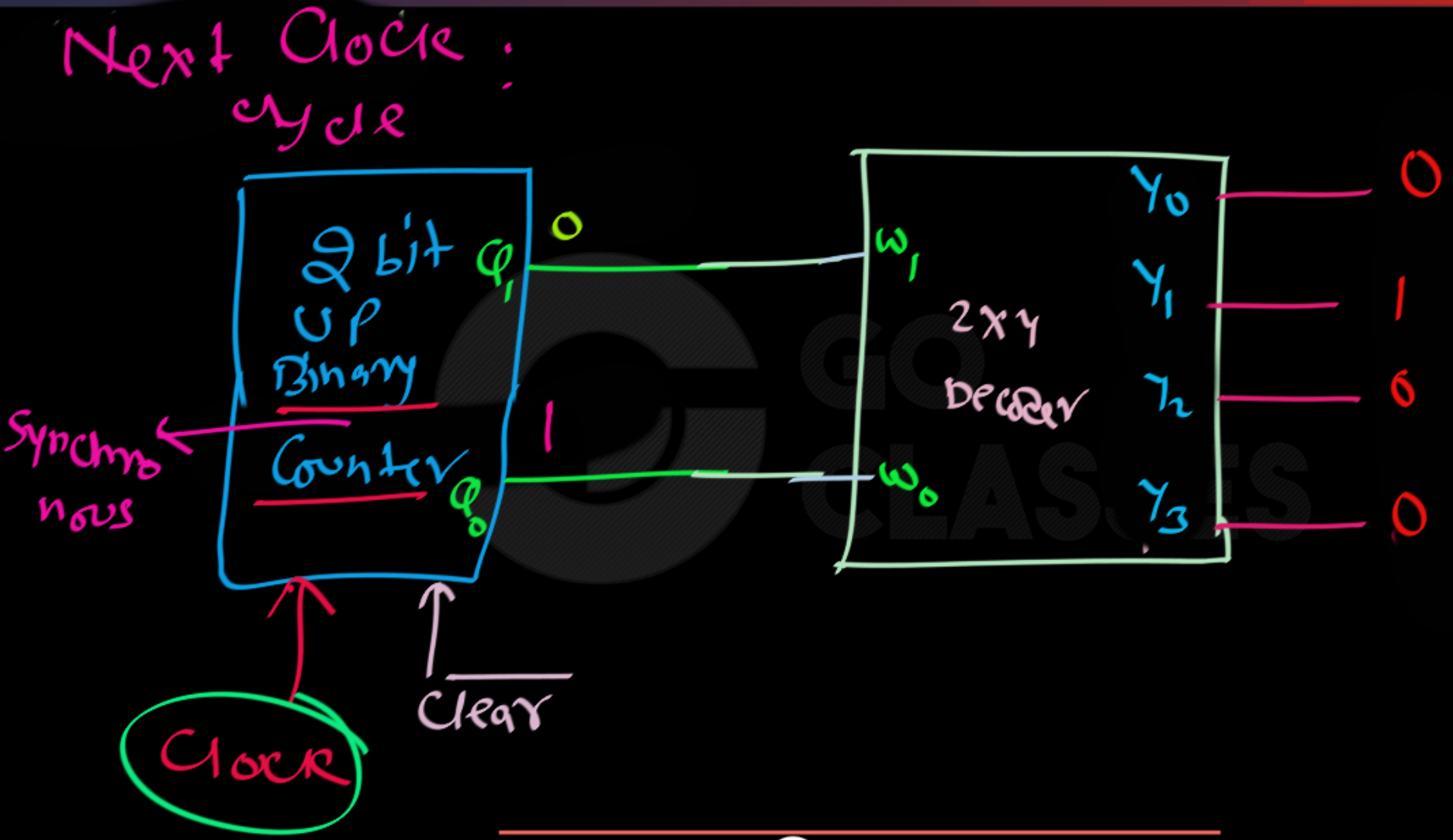
i.e. without Clock.

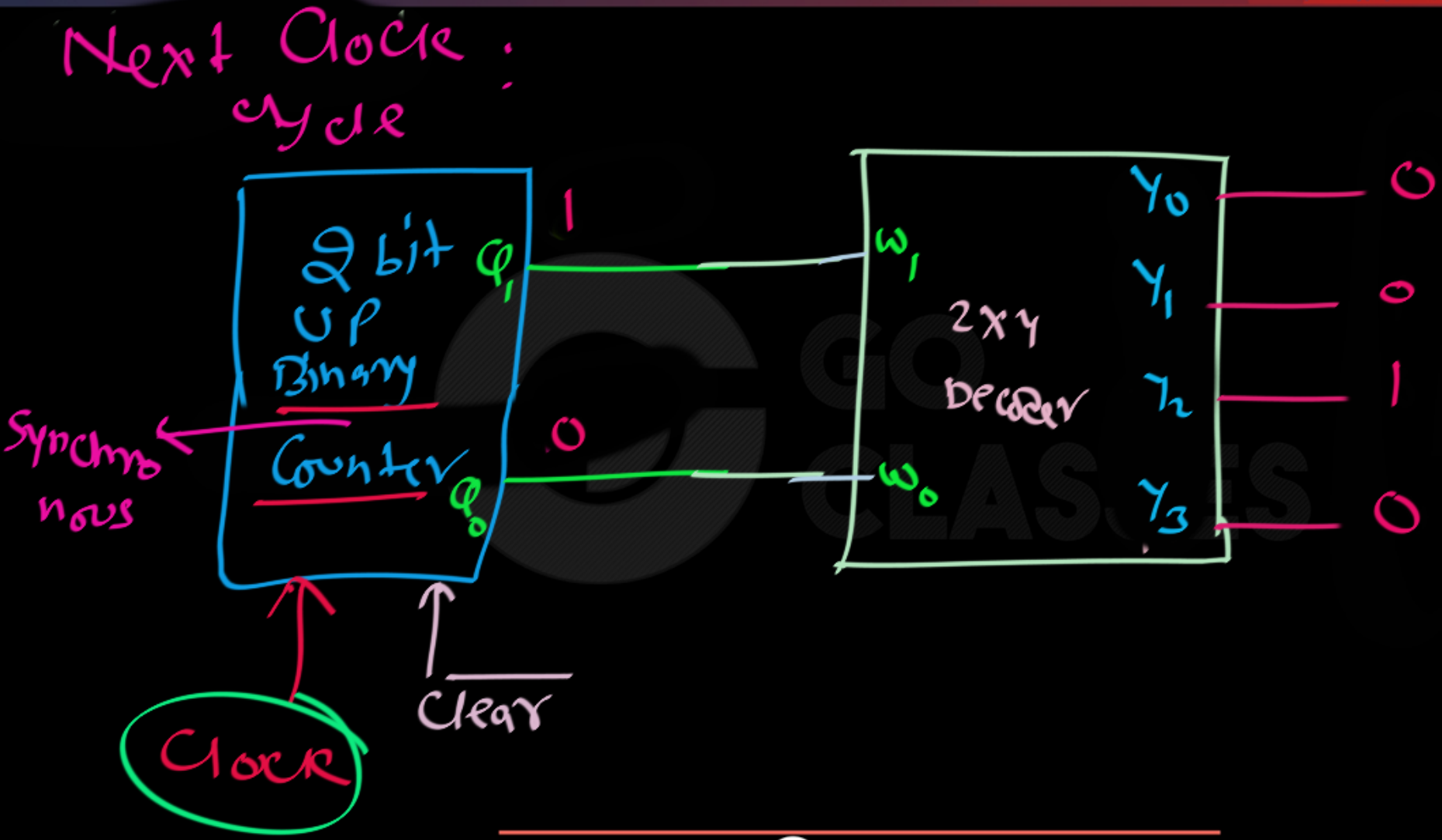
for Ring Counter, we need Synchronization using Clock.

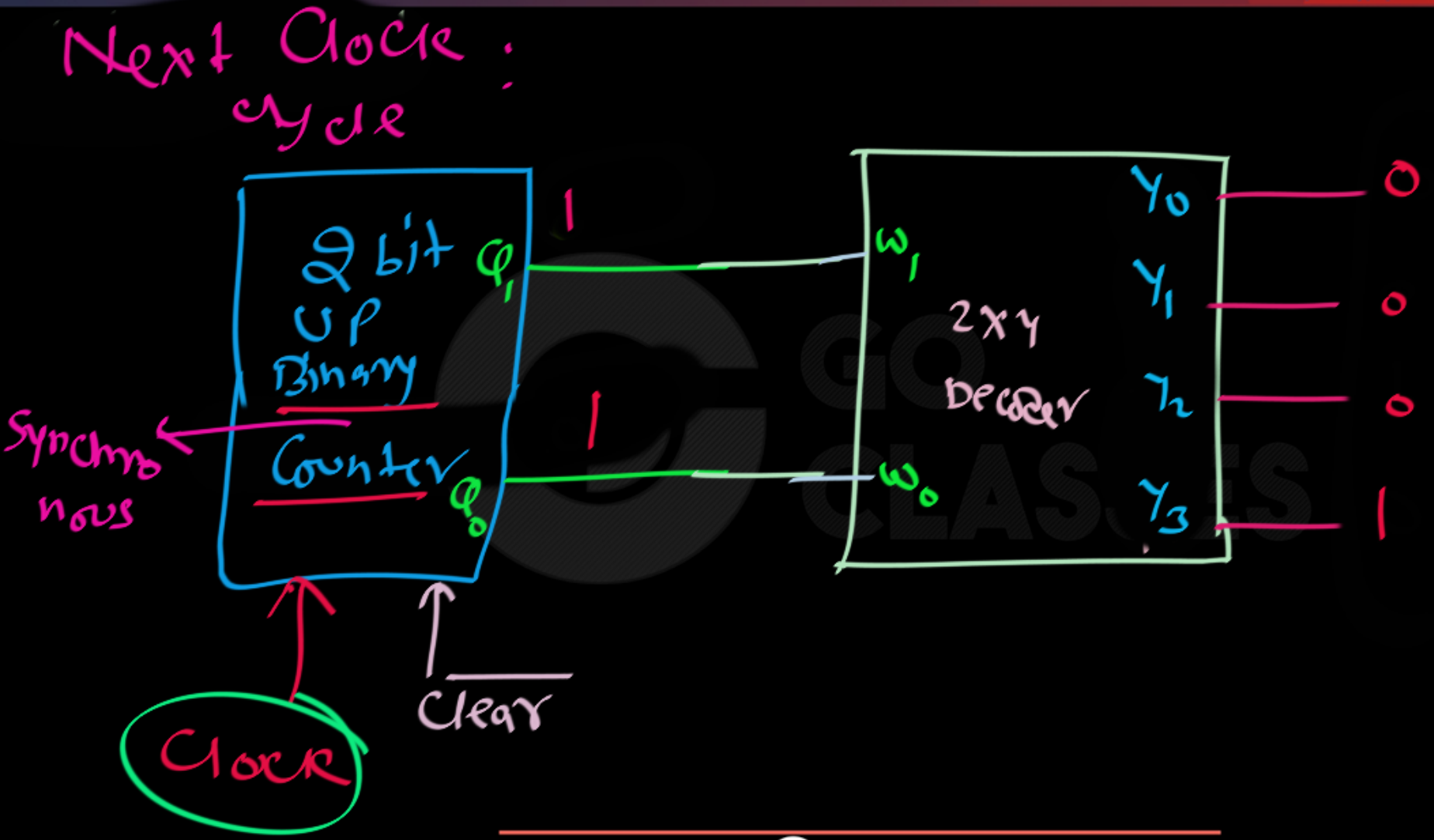

We want :
1000
after one Clock cycle
0001
0100
0010
After one clock cycle
After one Clock cycle

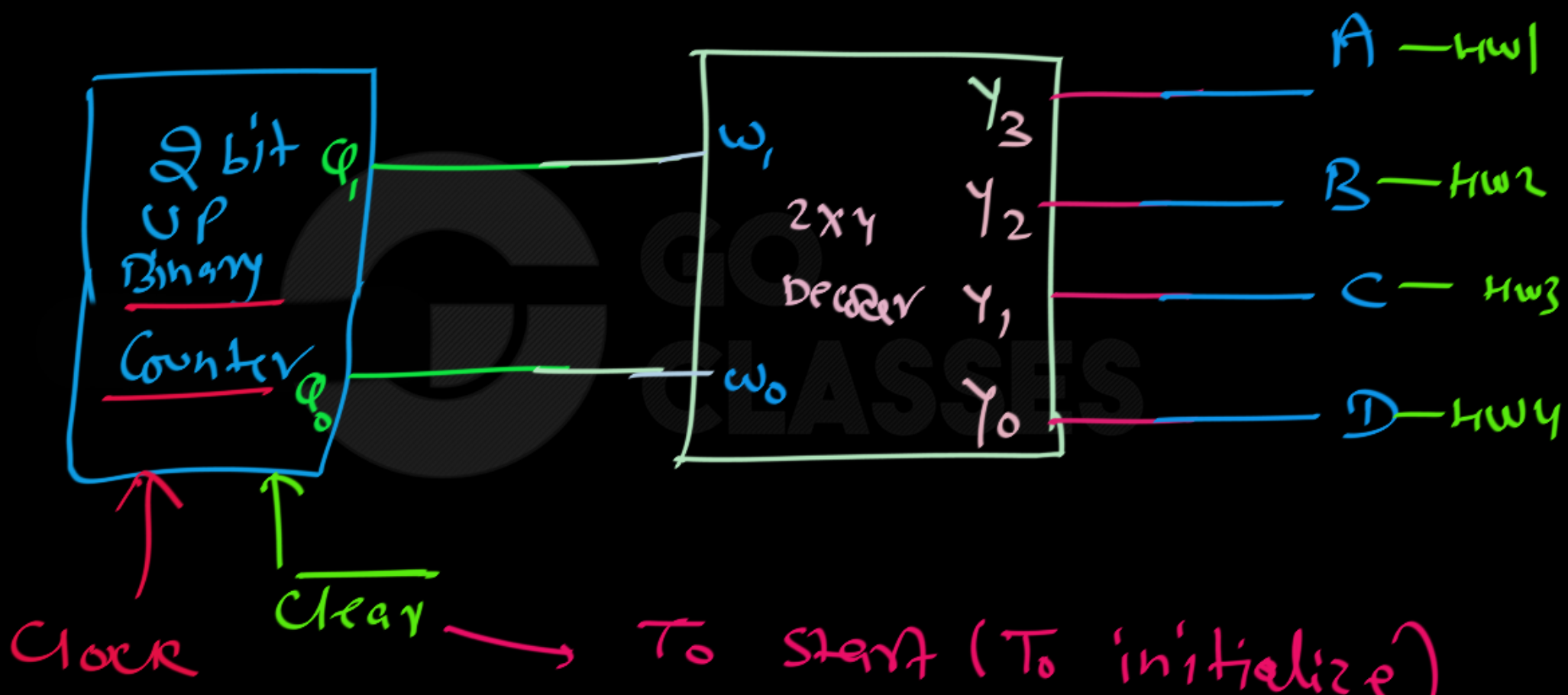
initialize :

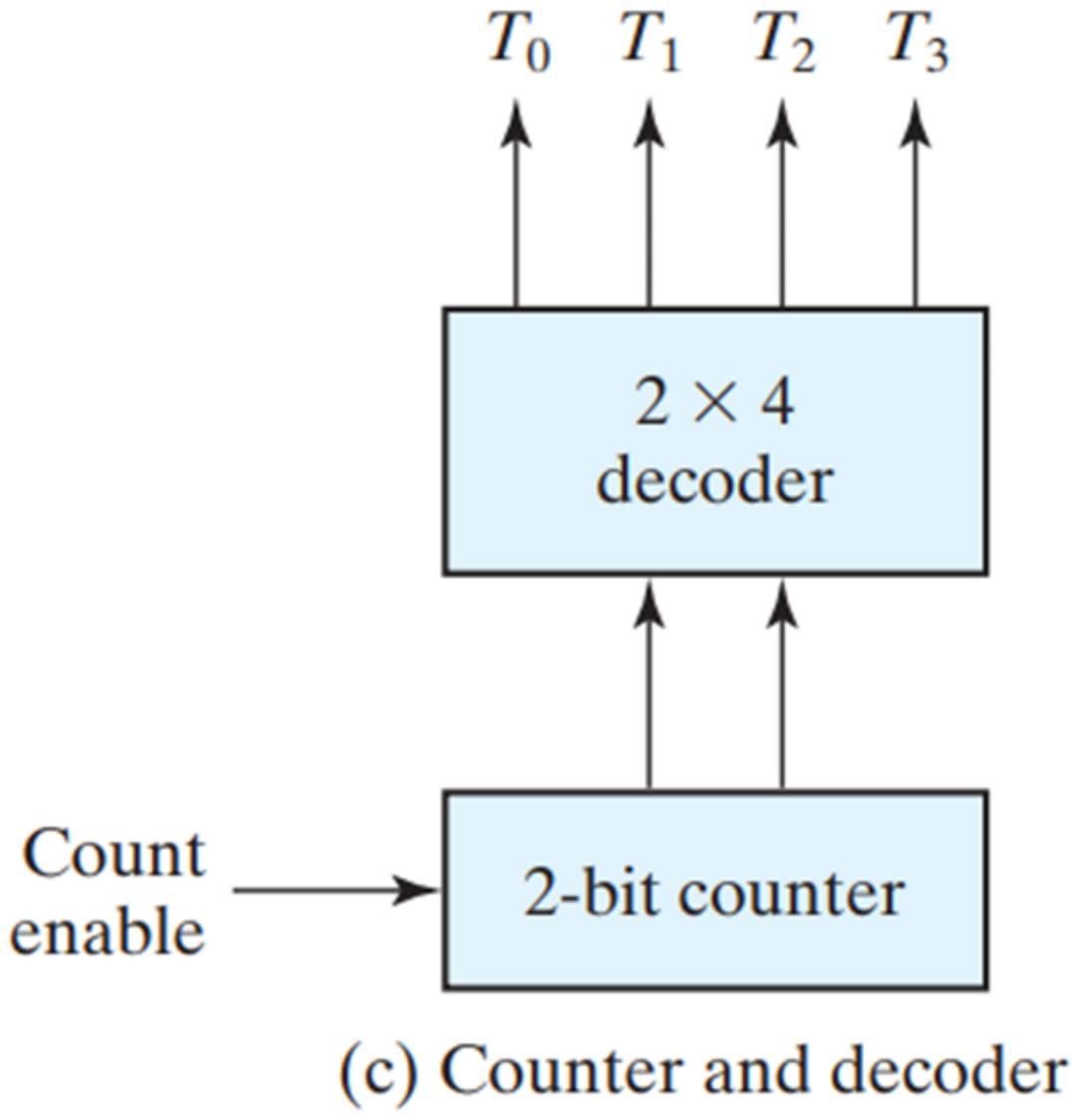

2 bit
UP
Binary
Counter
Synchro
nous
Q_1
Q_0
0
0
W_1
W_0
2x4
Decoder
Y_0
Y_1
Y_2
Y_3
1
0
0
0
Clock
Clear


Next Clock cycle :
2 bit UP Binary Counter
Synchronous
Q_1
Q_0
0
1
W_1
W_0
2x4 Decoder
Y_0
Y_1
Y_2
Y_3
0
1
0
0
Clock
Clear


Next Clock cycle :
2 bit UP Binary Counter
Synchronous
Q_1
Q_0
1
0
W_1
W_0
2x4 Decoder
Y_0
Y_1
Y_2
Y_3
0
0
1
0
Clear
Clock


Next Clock cycle :
2 bit
UP
Binary
Counter
Synchro nous
Q_1
Q_0
1
1
2x4
Decoder
W_1
W_0
Y_0
Y_1
Y_2
Y_3
0
0
0
1
Clear
Clock


2 bit
UP
Binary
Counter
Q_1
Q_0
W_1
W_0
2x4
Decoder
Y_3
Y_2
Y_1
Y_0
A — HW1
B — HW2
C — HW3
D — HW4
Clock
Clear
To start (To initialize)

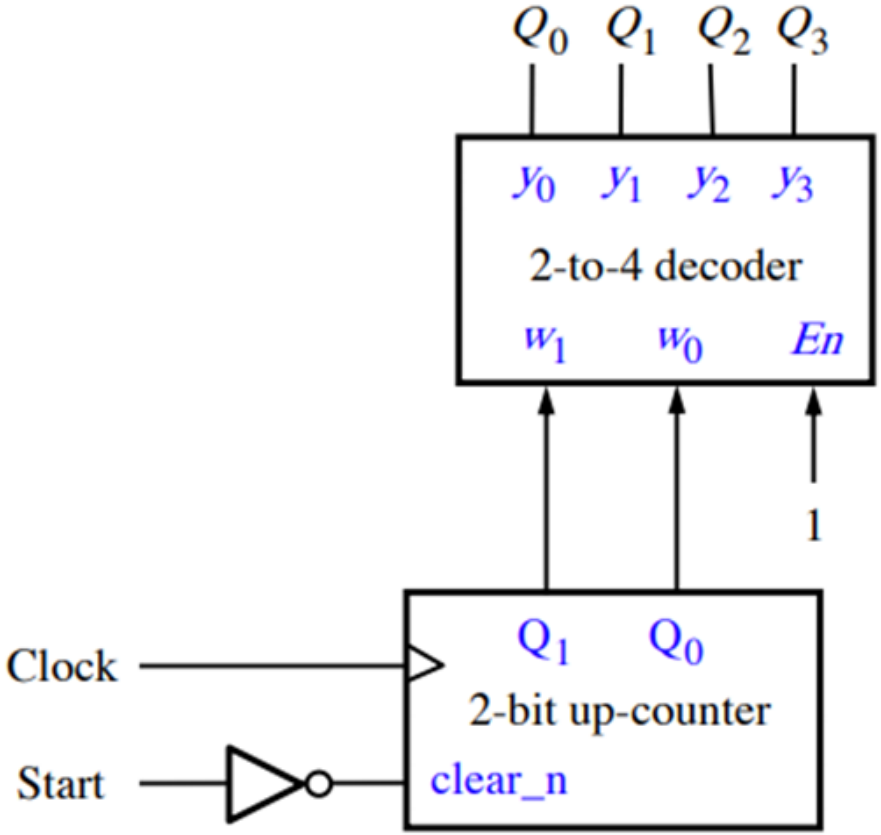


(c) Counter and decoder

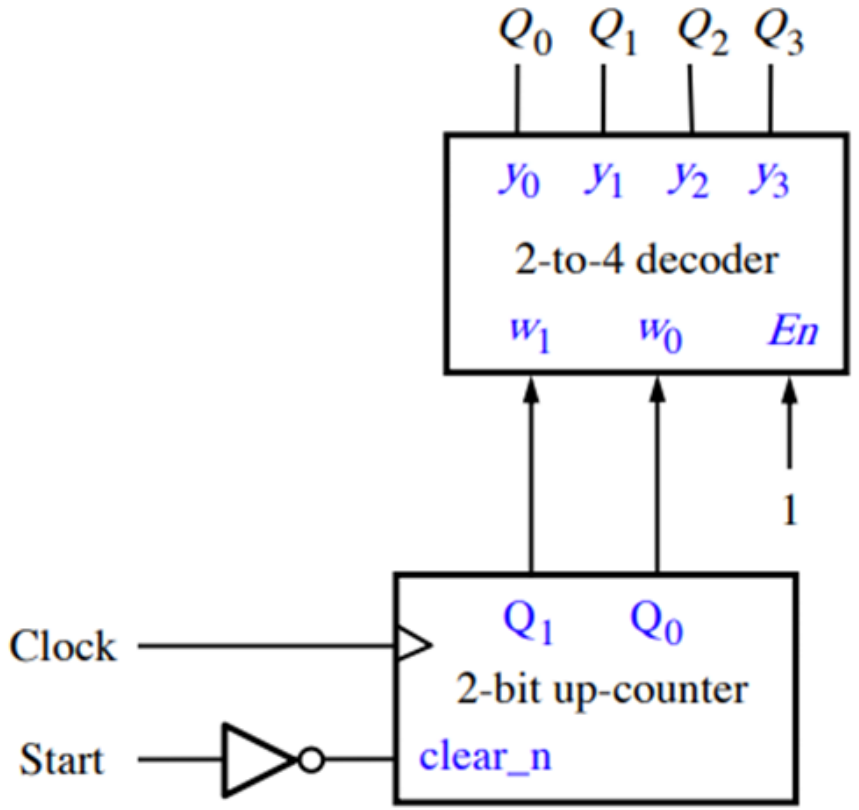
Alternative version of a 4-bit ring counter

- This implementation uses 2-bit up-counter followed by a 2-to-4 decoder.
- The counter cycles through 00, 01, 10, 11, 00, …
- Recall that the outputs of the decoder are one-hot encoded. Thus, there is only one 1 on its outputs.
- Because the output of the counter is the input to the decoder, the outputs of the decoder cycle through: 1000, 0100, 0010, 0001, 1000, …
- This is the counting sequence for a ring counter.

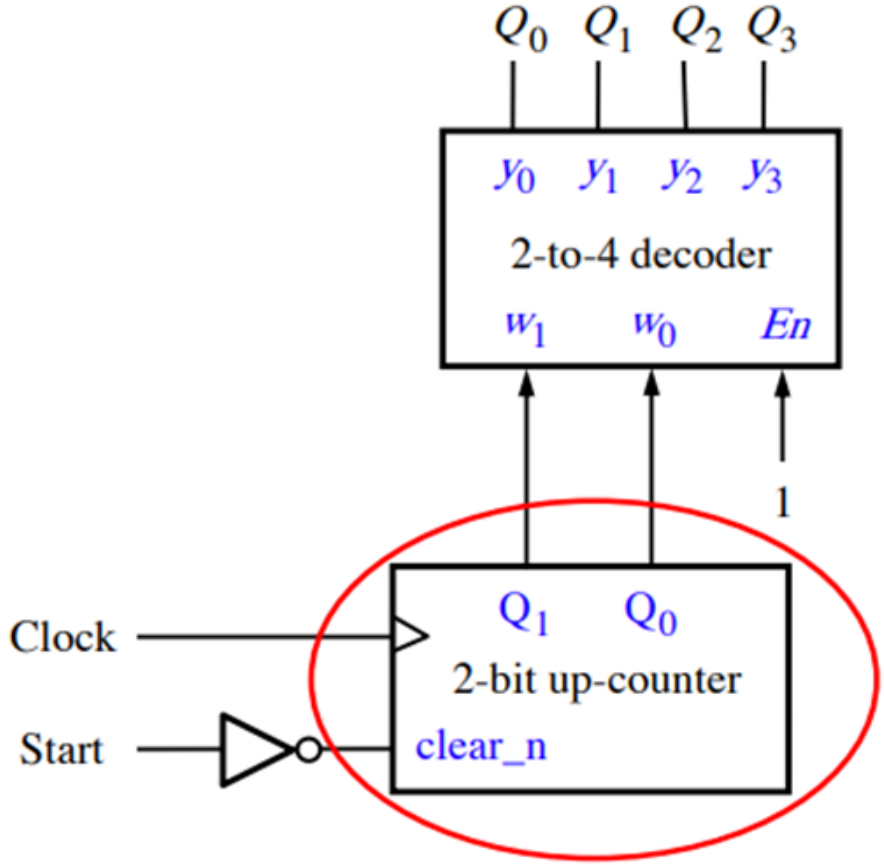
Alternative version of a 4-bit ring counter



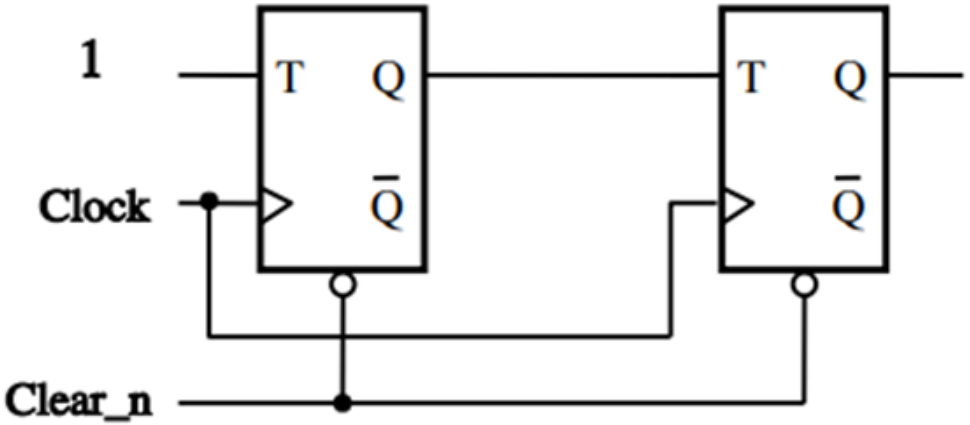
What are the components?



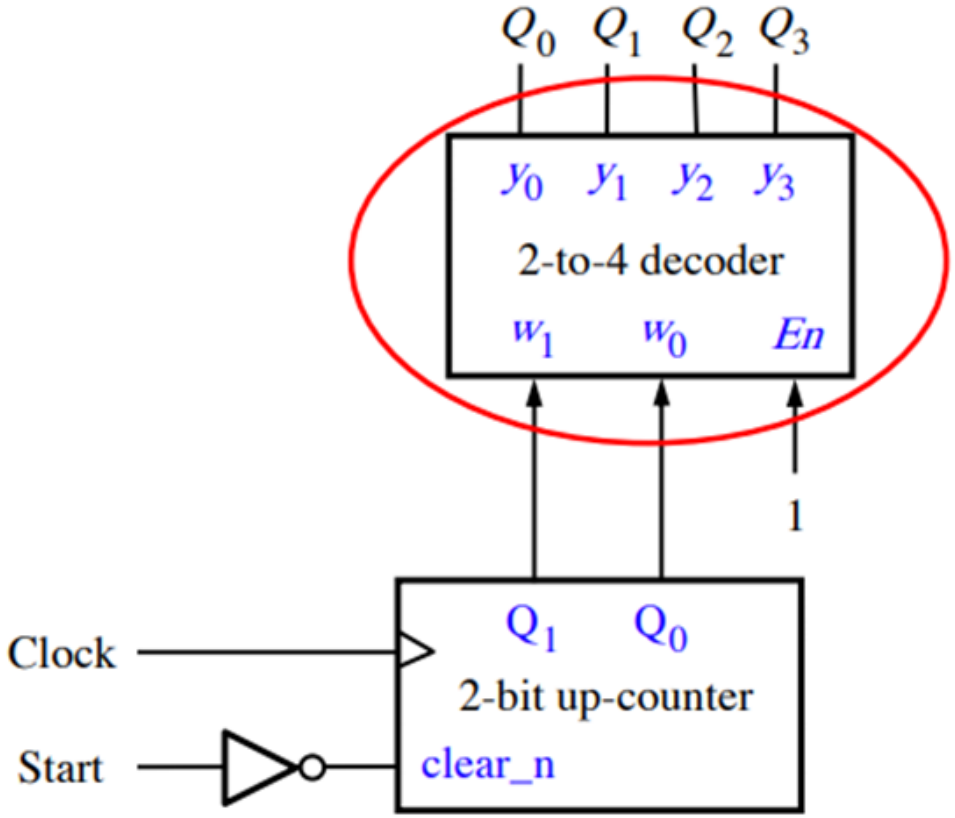
2-Bit Synchronous Up-Counter



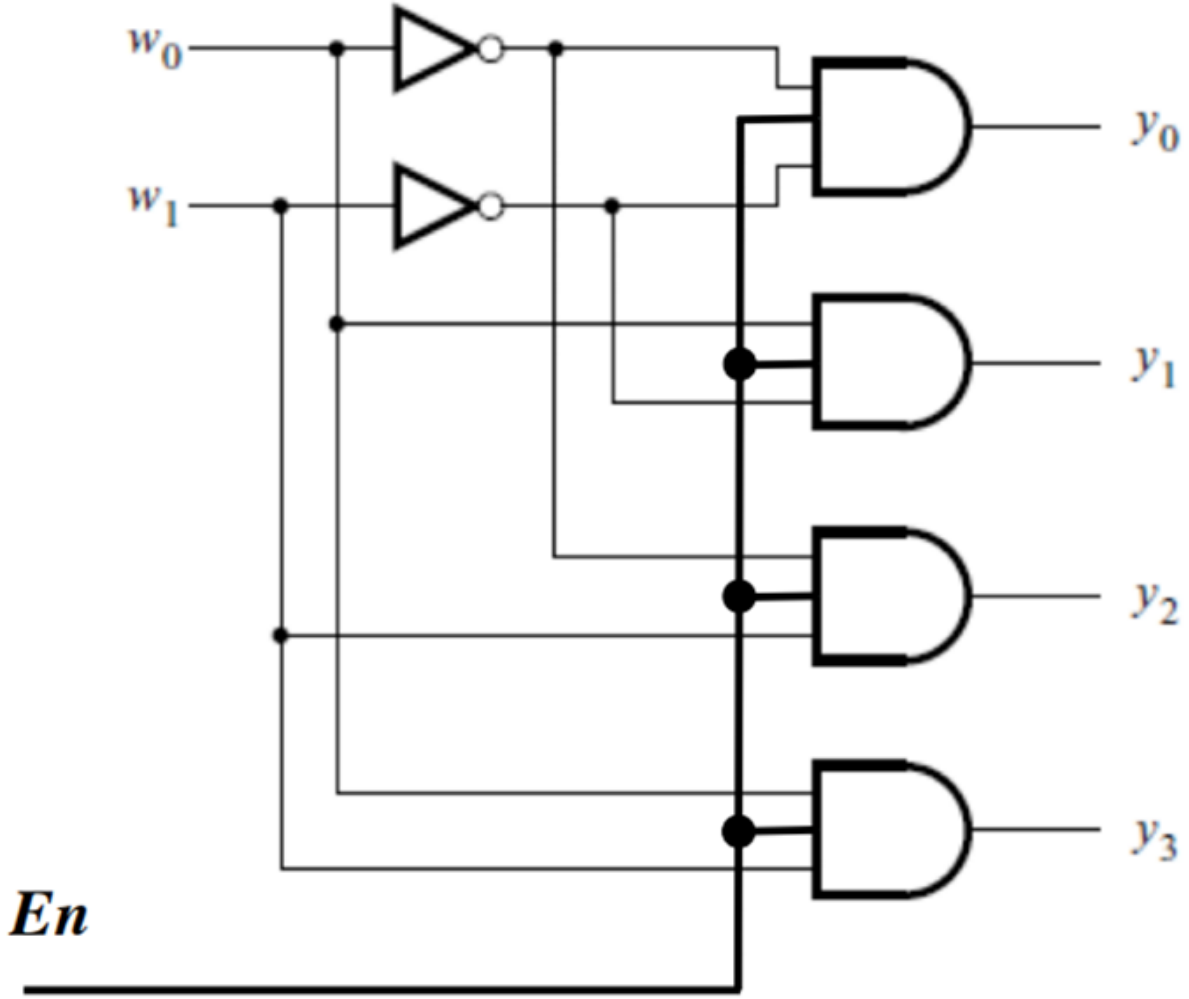
2-Bit Synchronous Up-Counter



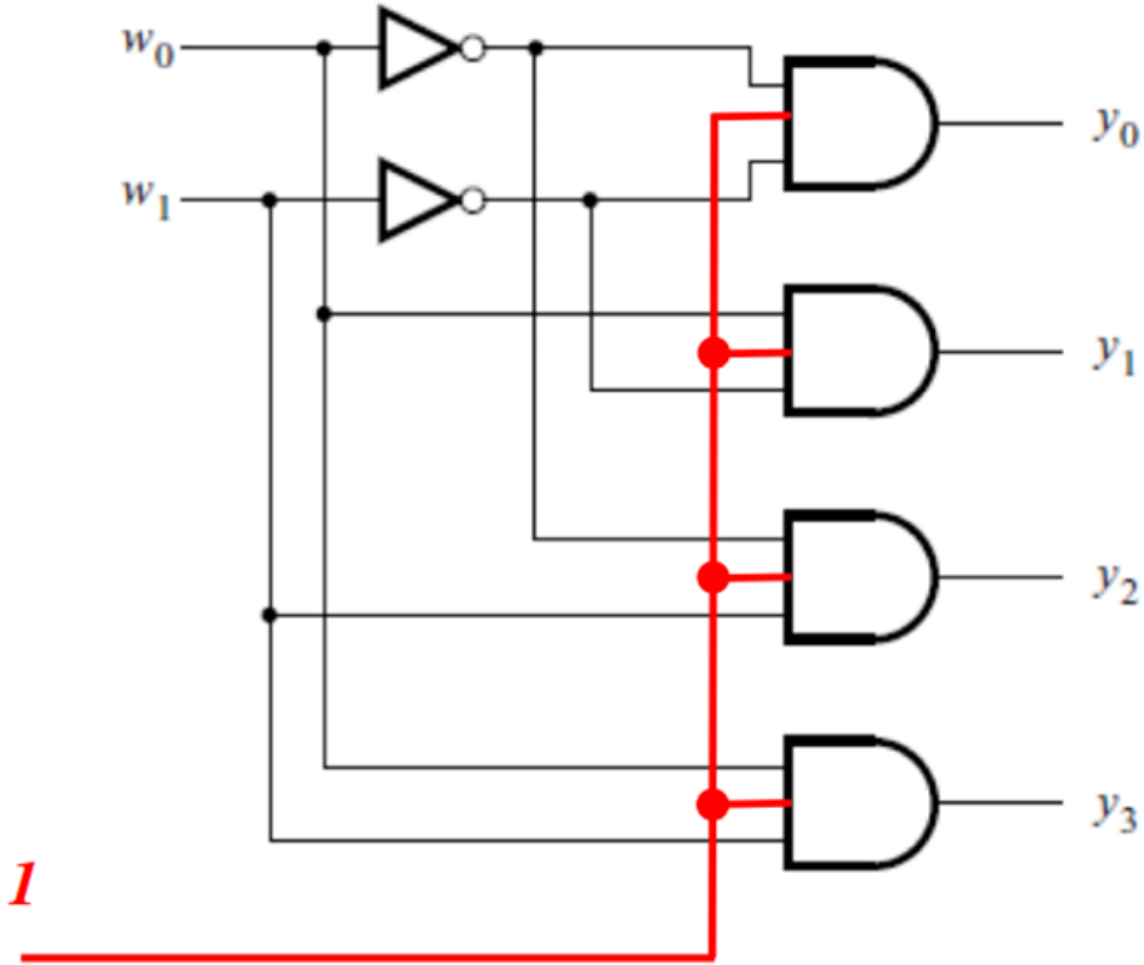
2-to-4 Decoder with Enable Input



2-to-4 Decoder with Enable Input

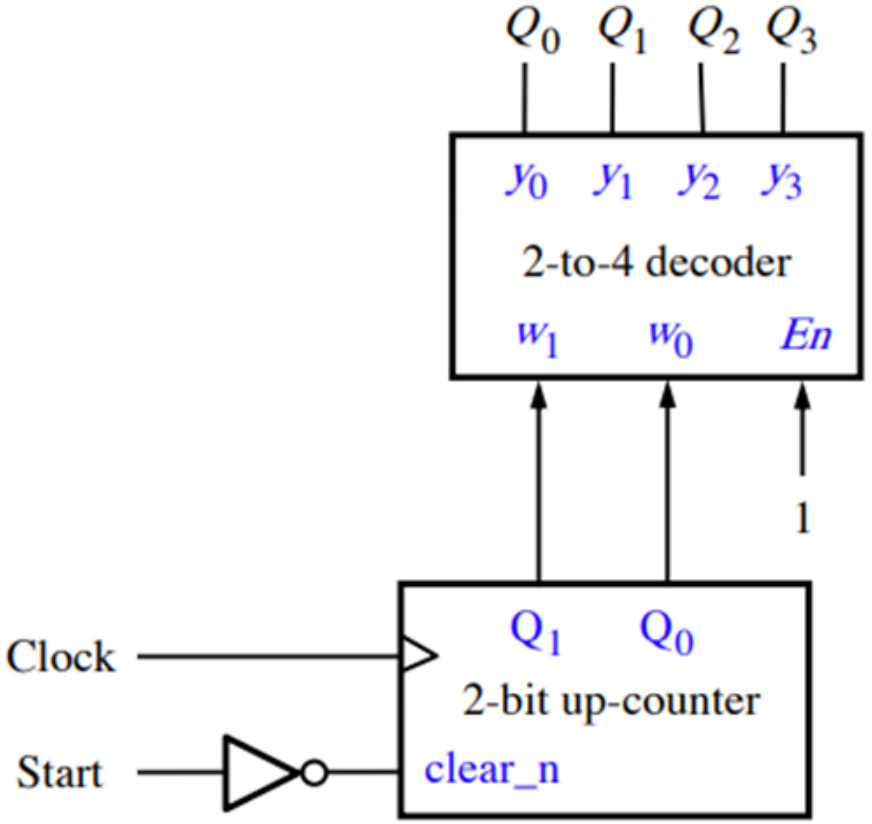


2-to-4 Decoder with Enable Input

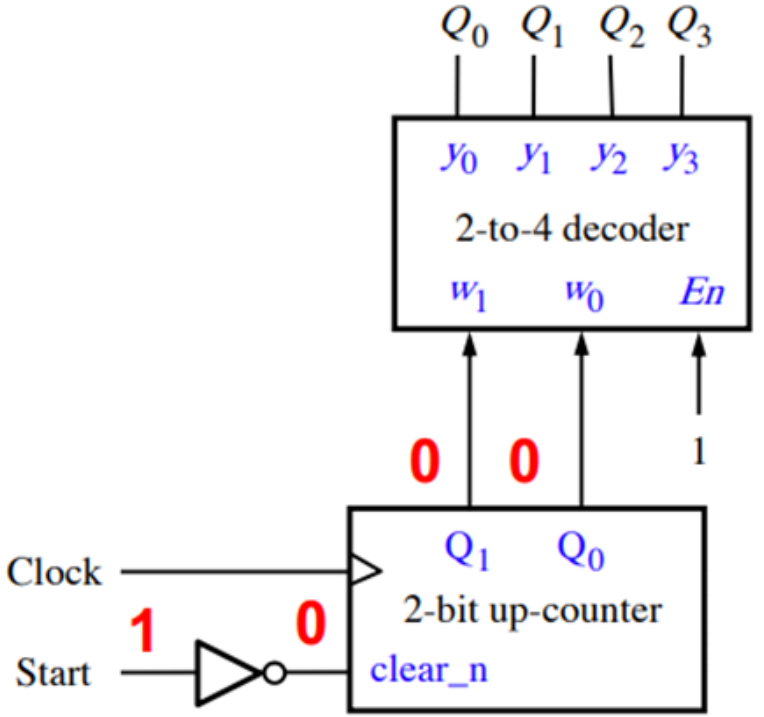


(always enabled in this example)

How Does It Work?

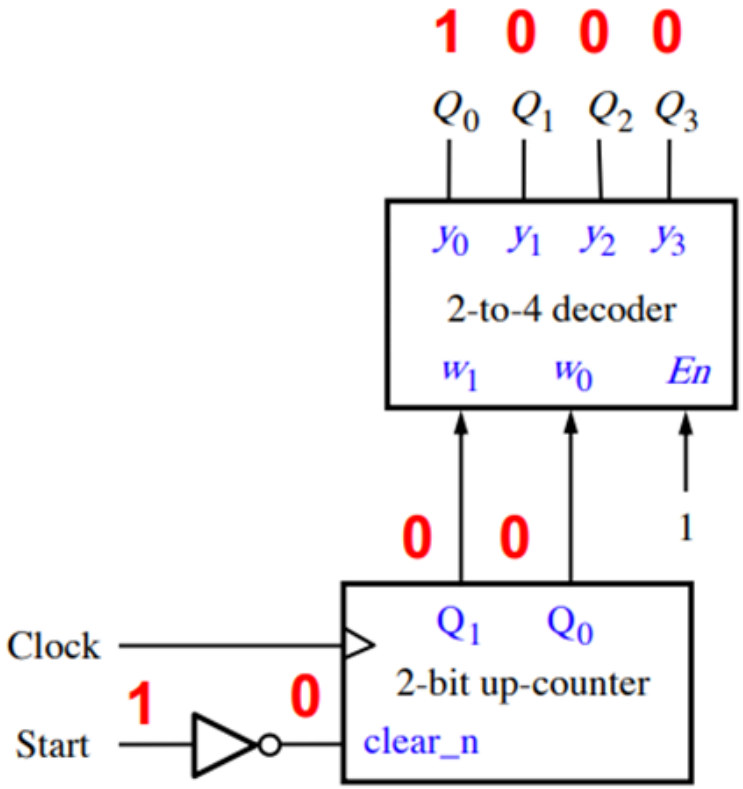


How Does It Work?



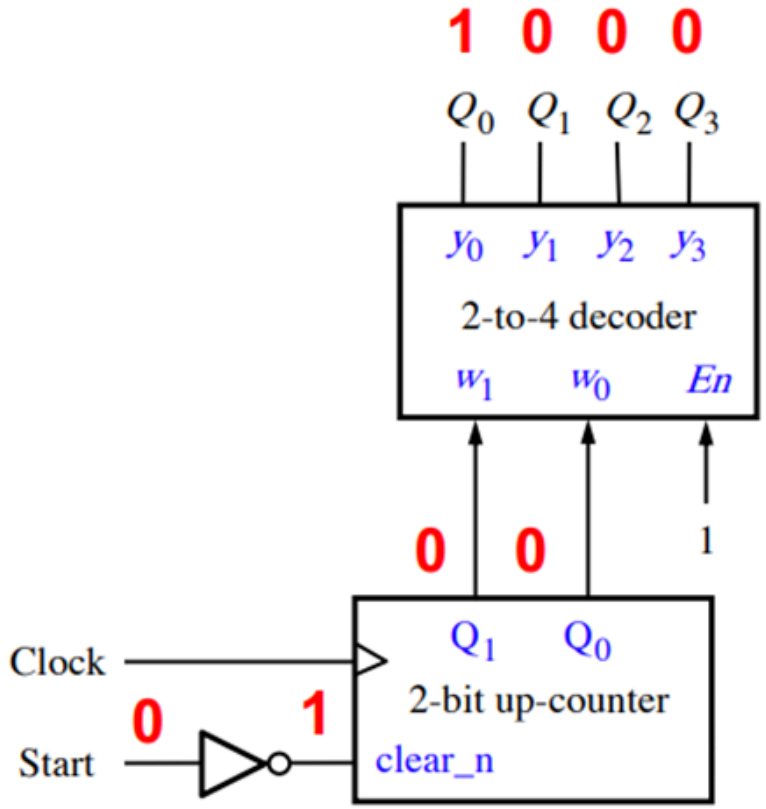
To initialize this circuit set Start to 1, which sets the counter to 00.

How Does It Work?



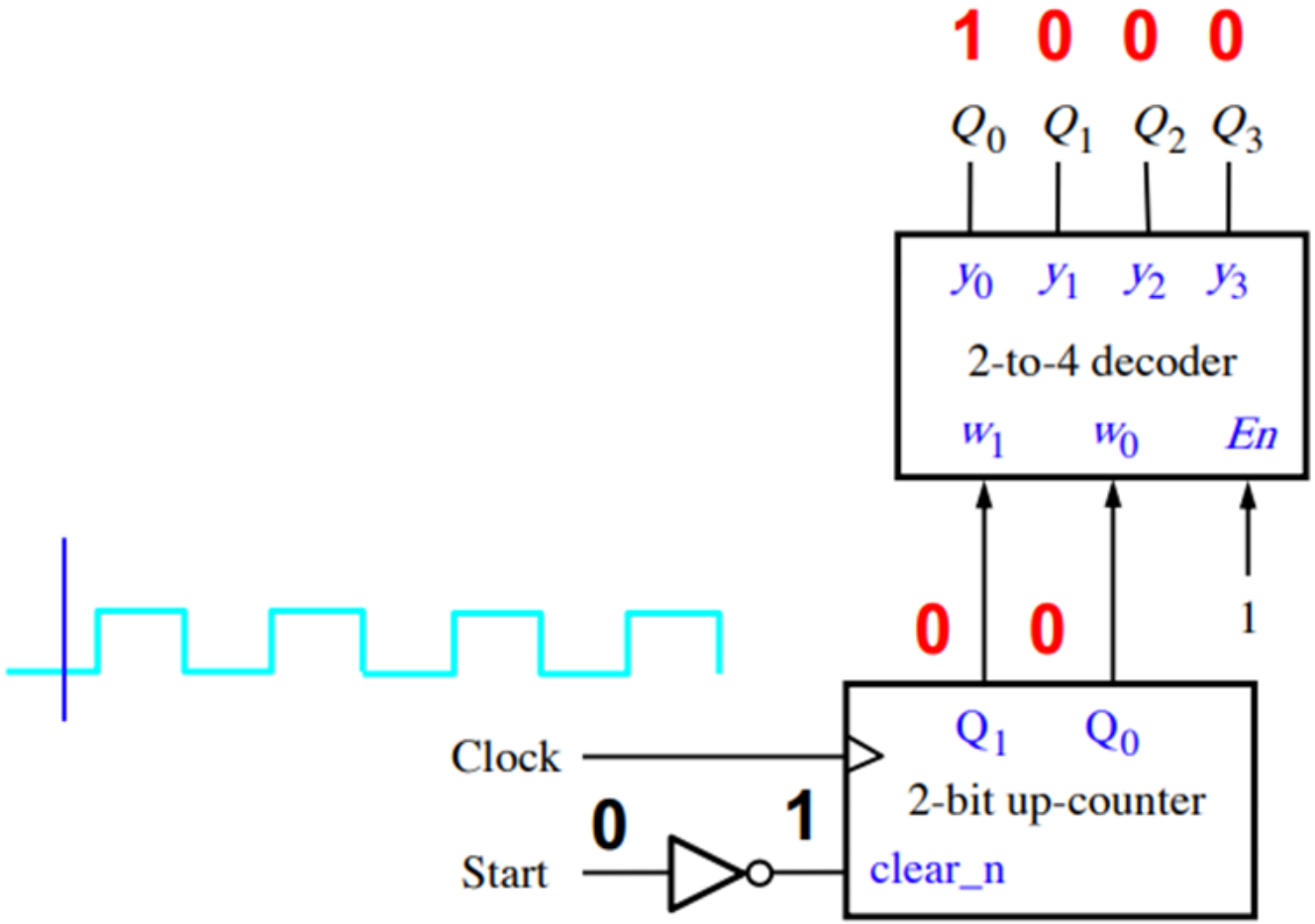
This sets the outputs of the decoder to the start of the counting sequence.

How Does It Work?

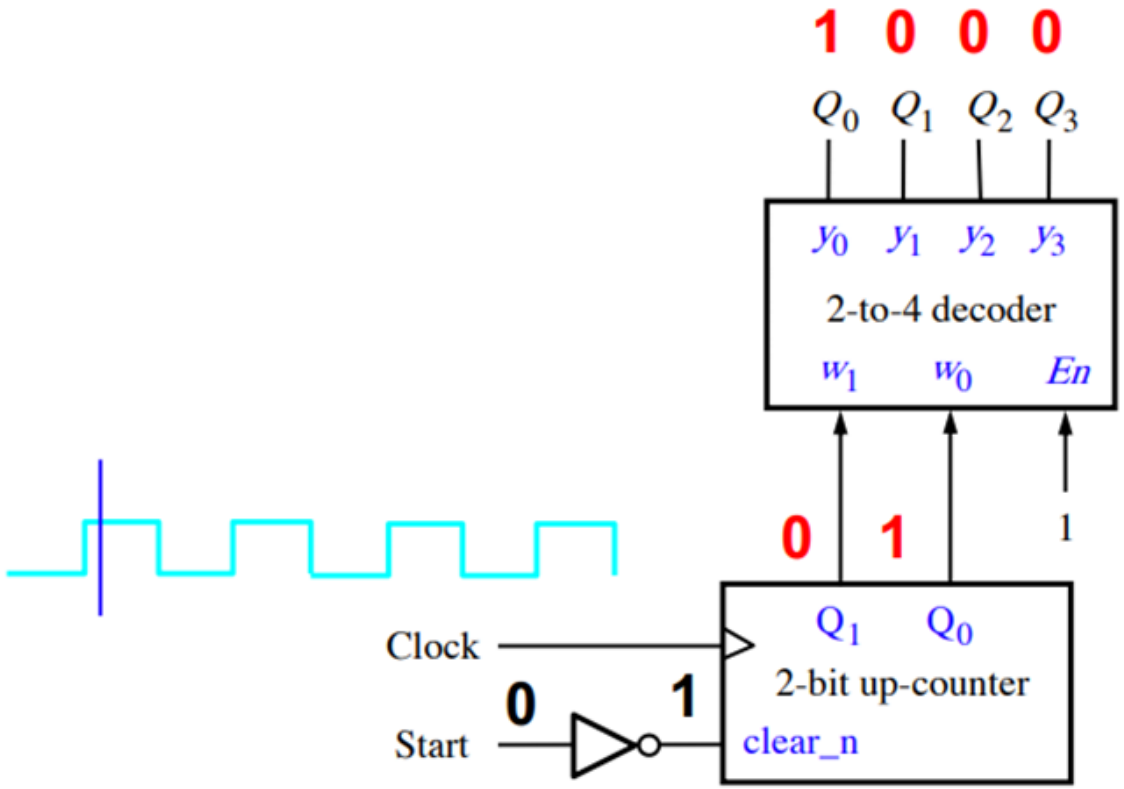


When Start is equal to 0, clear_n has no effect.

How Does It Work?

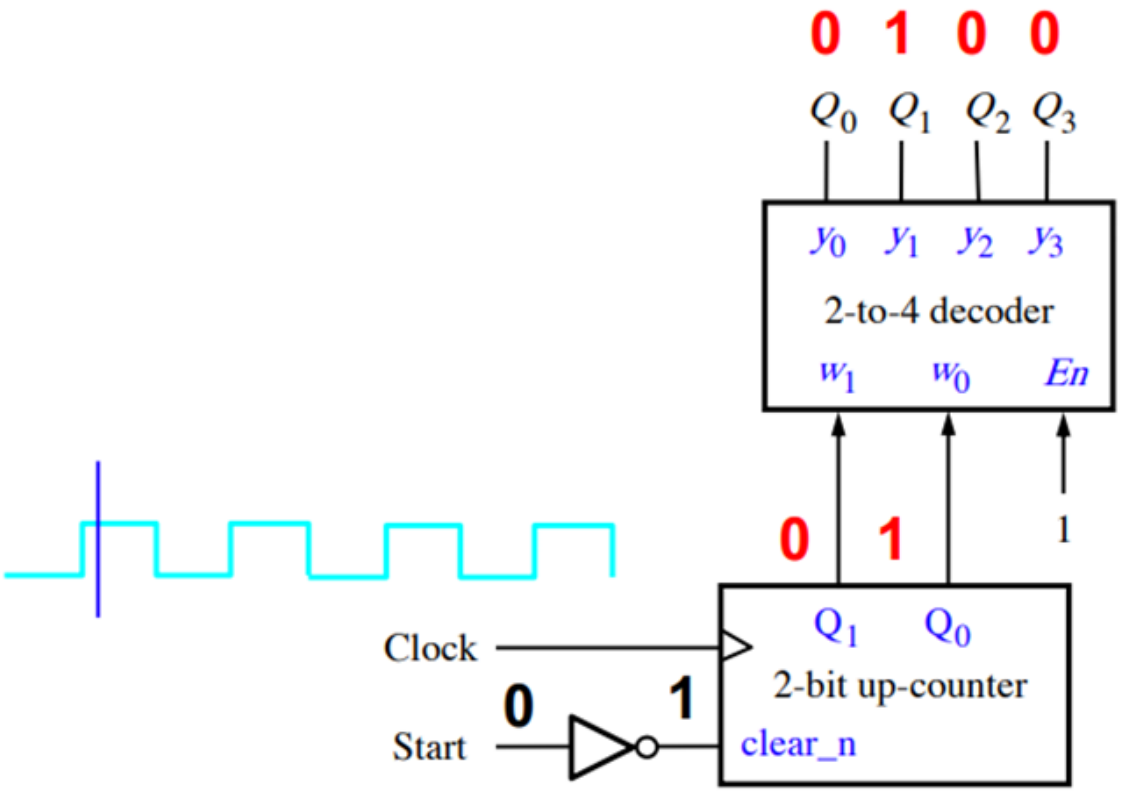


How Does It Work?



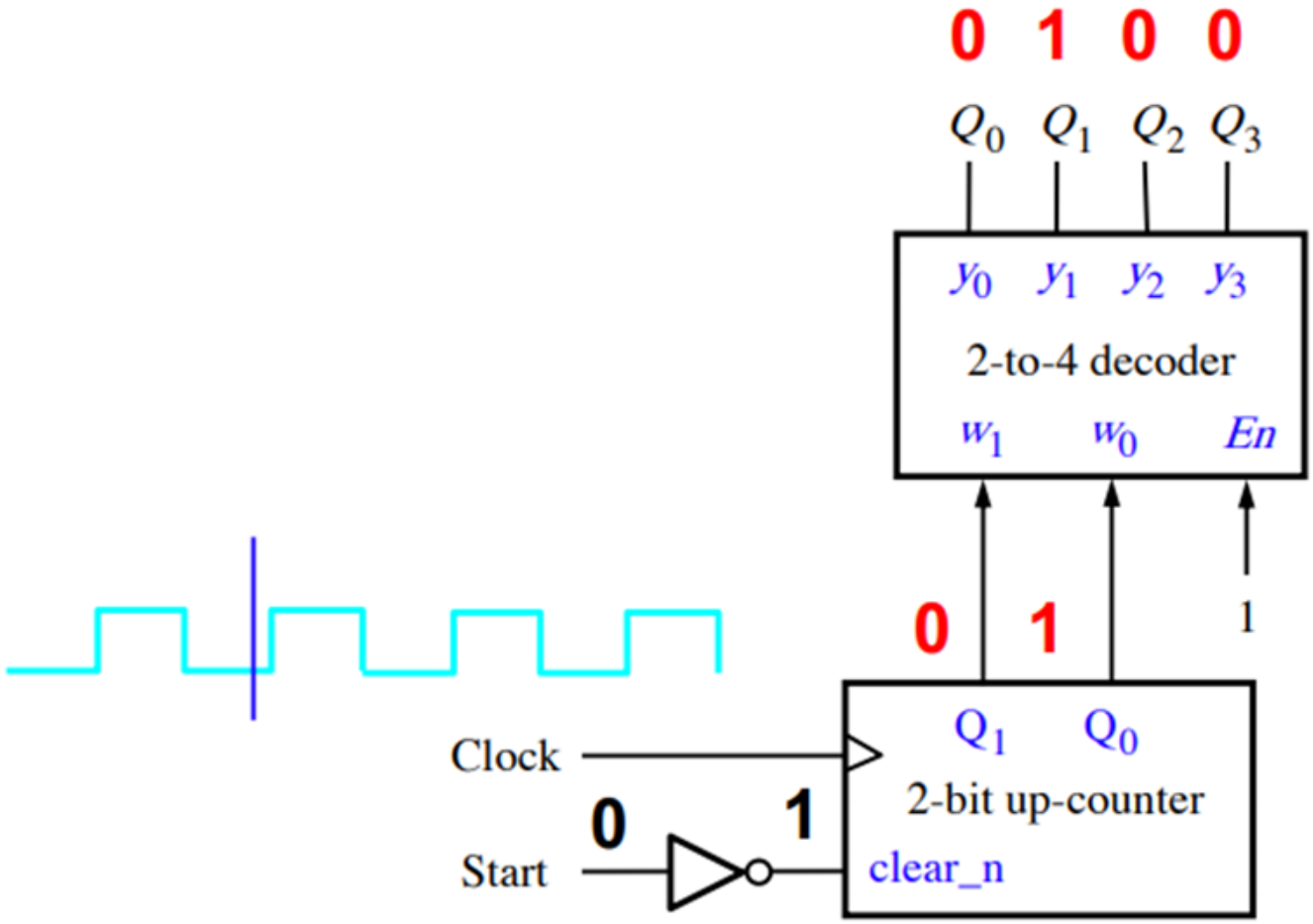
The counter increments the count on the positive clock edge ...

How Does It Work?

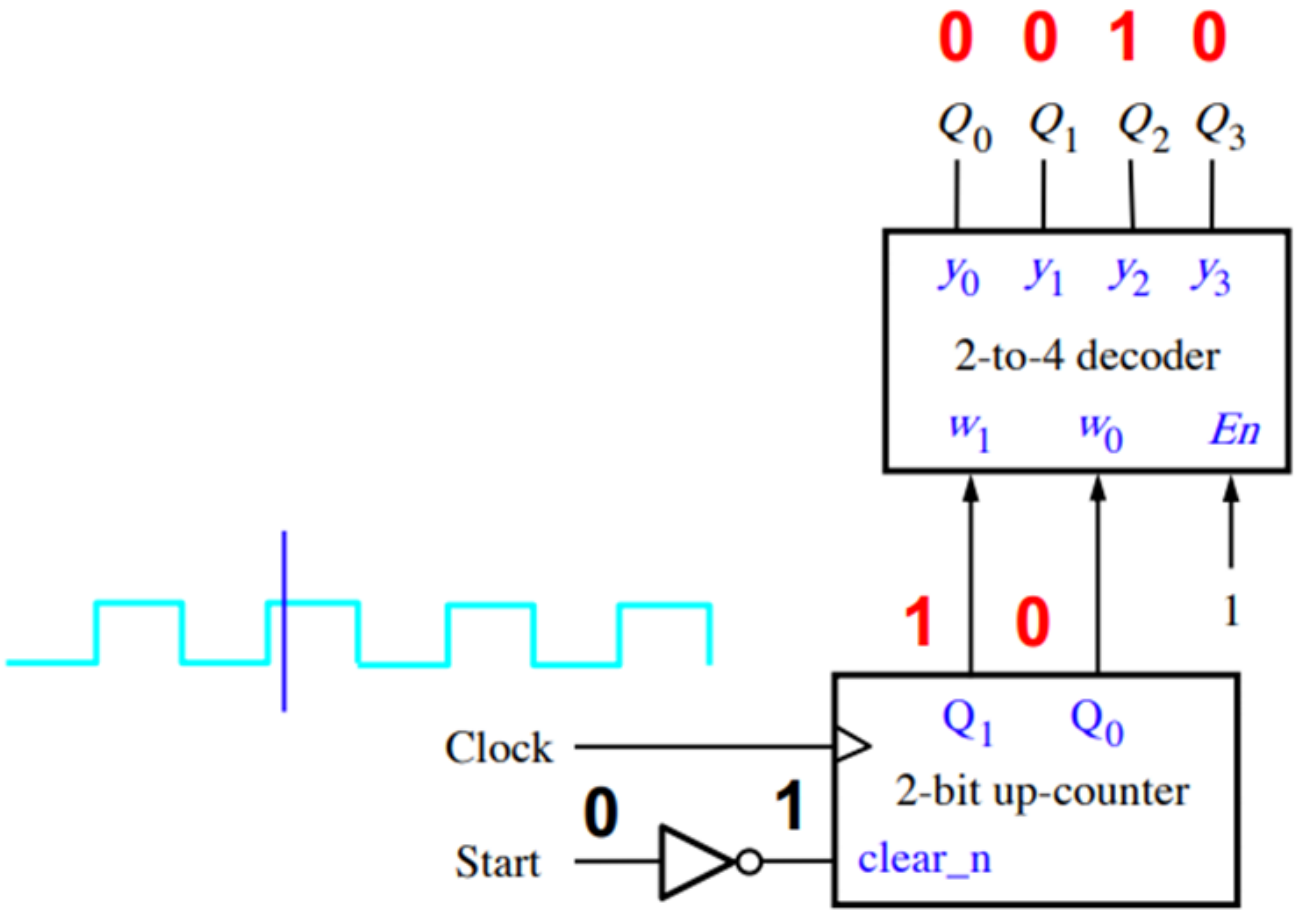


… this updates the outputs of the decoder (which are one hot encoded).

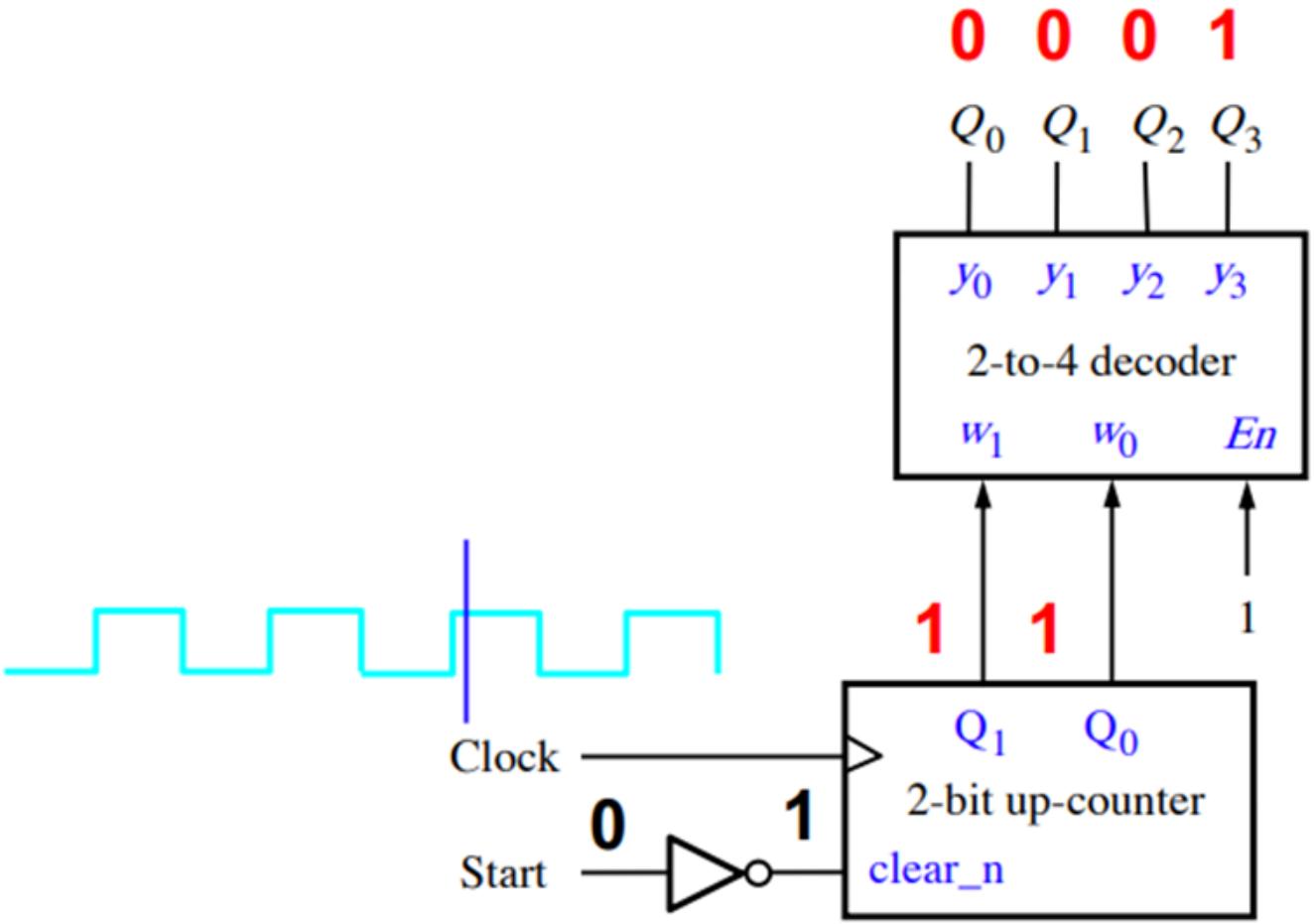
How Does It Work?



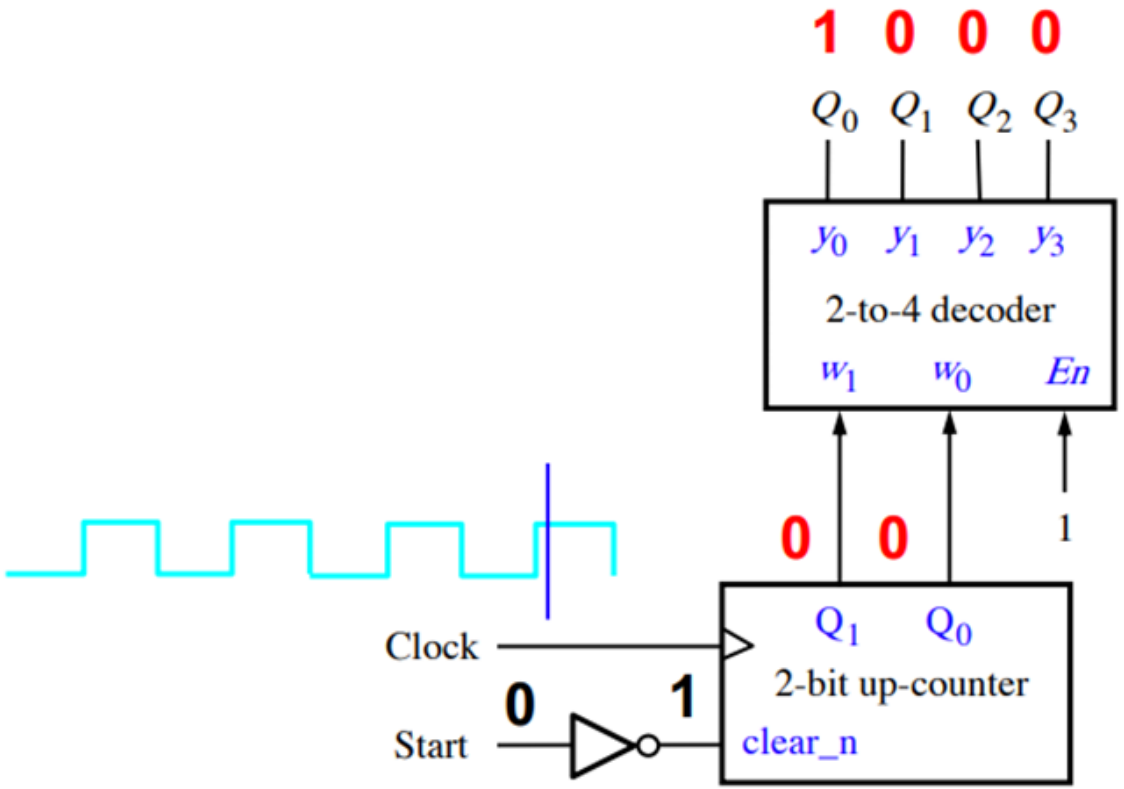
How Does It Work?



How Does It Work?



How Does It Work?



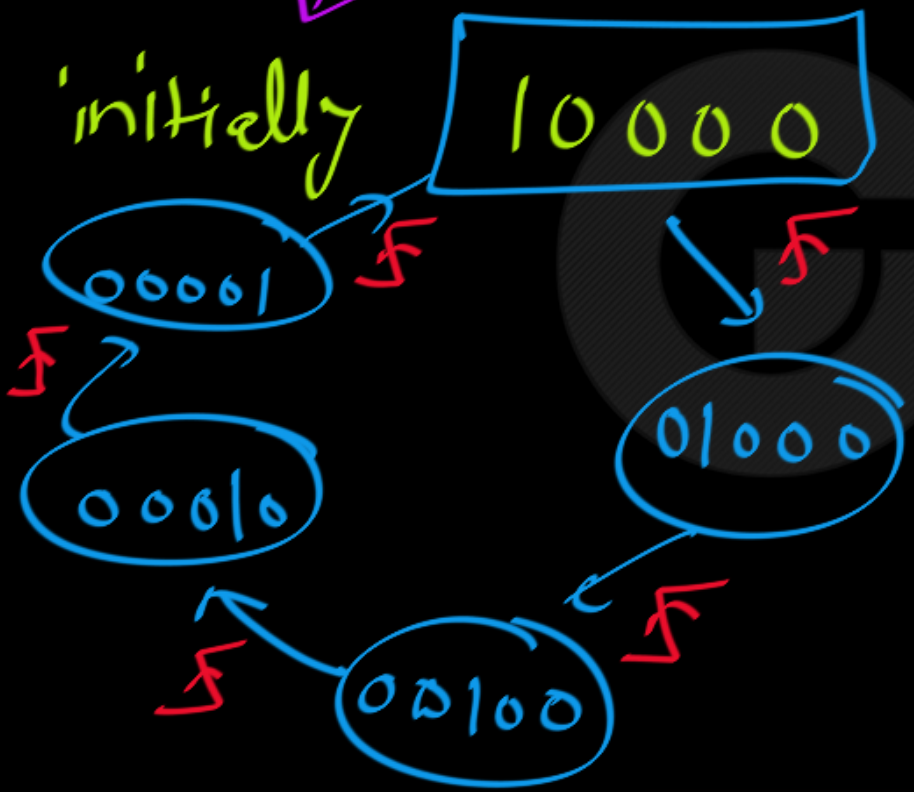
It is back to the start of the counting sequence, which is: 1000, 0100, 0010, 0001.

Ring Counter: The Conclusion

To generate 2^n timing signals, we need either a shift register with 2^n flip-flops or an n-bit binary counter together with an n-to-2^n-line decoder. For example, 16 timing signals can be generated with a 16-bit shift register connected as a ring counter or with a 4-bit binary counter and a 4-to-16-line decoder. In the first case, we need 16 flip-flops. In the second, we need 4 flip-flops and 16 four-input AND gates for the decoder. It is

Ring Counter: → Special Purpose Counter

'initially 10000



Generation of timing signal to control the "sequence of operations" in a Digital circuit.

Two Standard implementations :

n-bit Ring Counter :

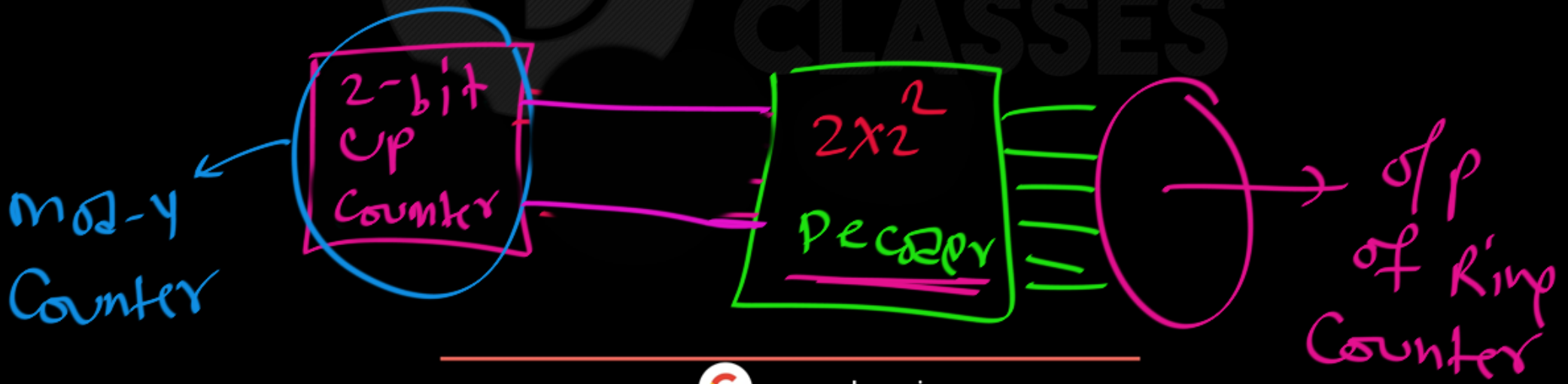
(1) using Shift Register :

n-FF needed

Two Standard implementations :

2^2-bit Ring Counter :

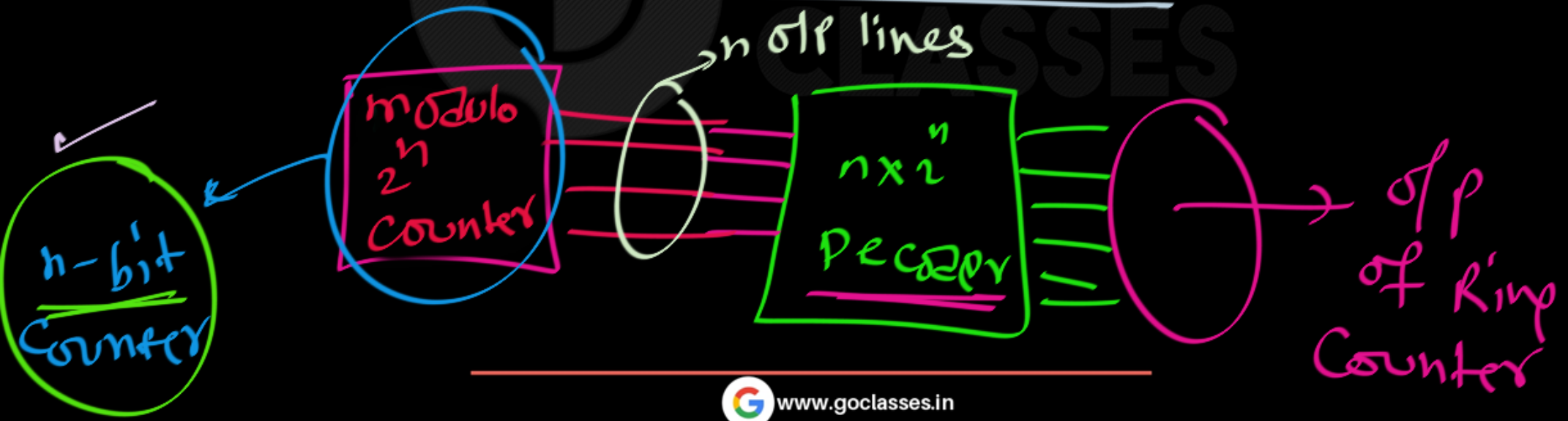
(2) using Counter with Decoder :



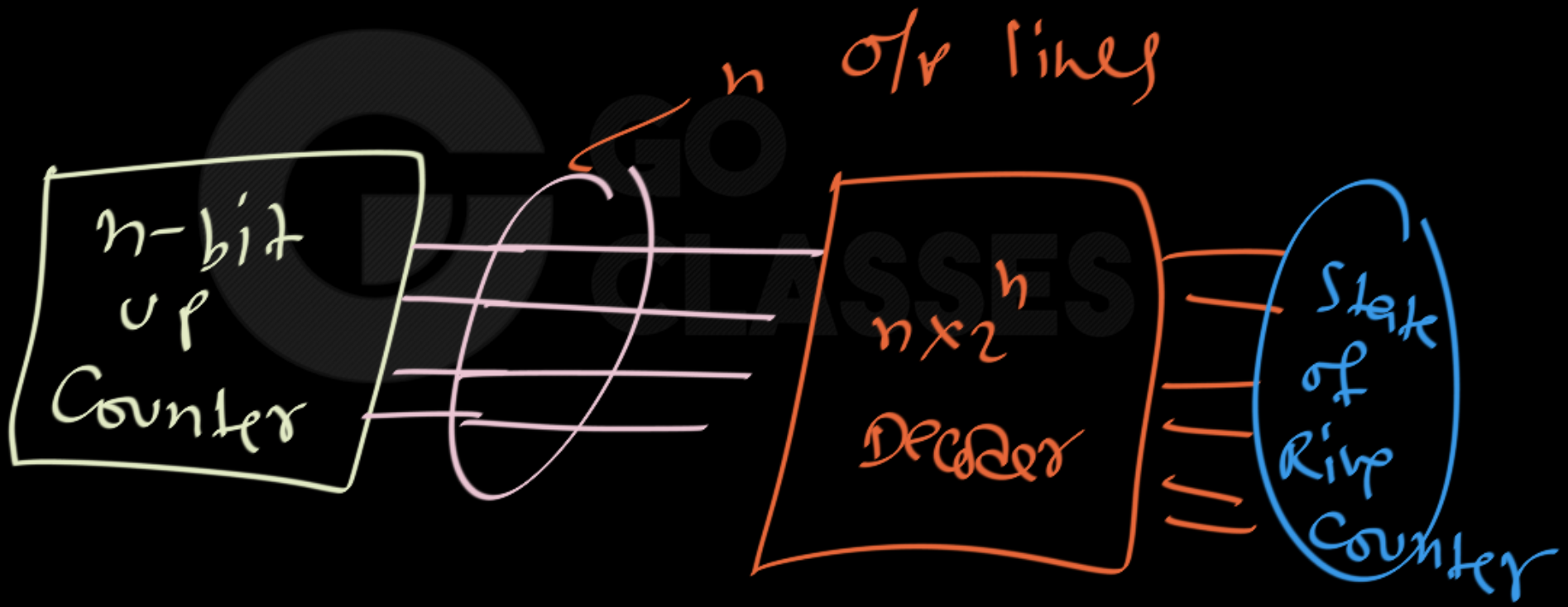
Two Standard implementations :

2^n - bit Ring Counter :

(2) using Counter with Decoder :



2^n - bit Ring Counter :



Note:

n-bit binary up counter:

$\equiv$ mod-2^n Counter

(n o/p lines ; n-FFs)

Next Topic:

Twisted Ring Counter

Twisted Ring Counter
or
Switch-Tail ring counter
or
Mobius Counter

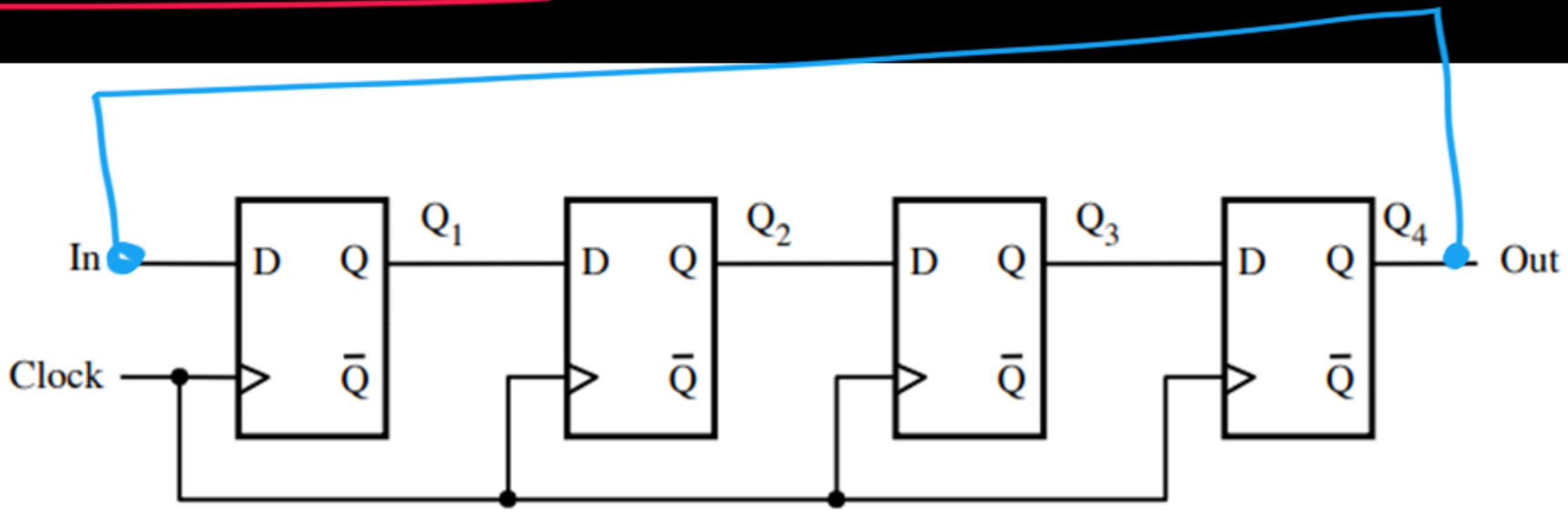
} Same

Is “Johnson counter” Same as Twisted Ring Counter ?

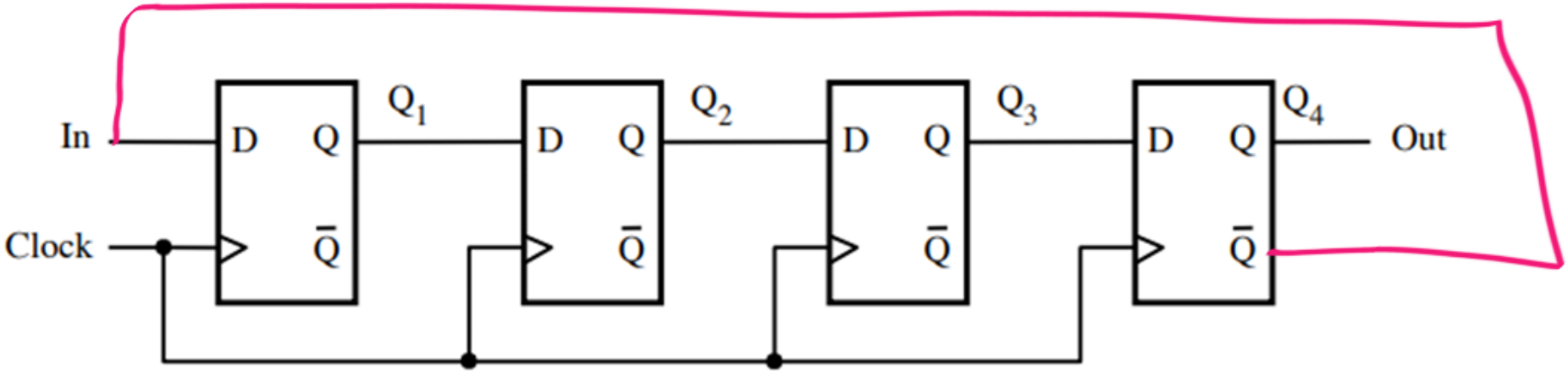
Are these two same Or is there any difference??

We will see at the end.. For now, assume that both are same.

Ring Counter : (Straight Ring Counter)

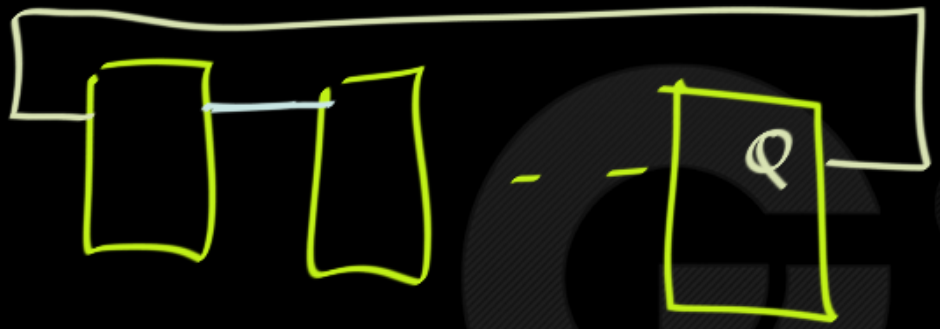


Twisted Ring Counter ;

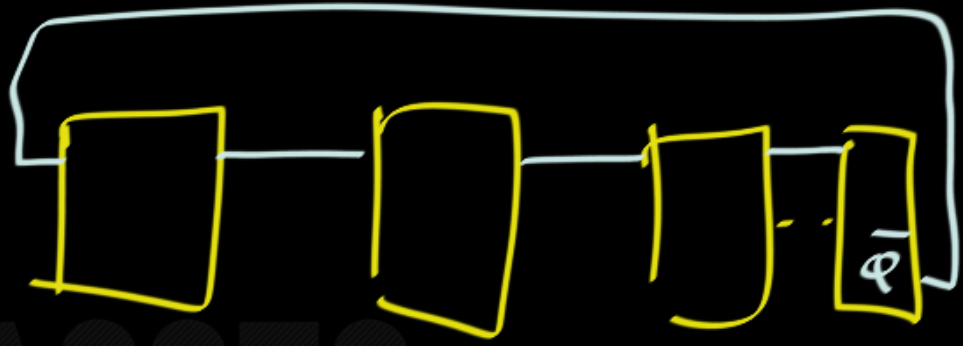


Why?

n - ffs



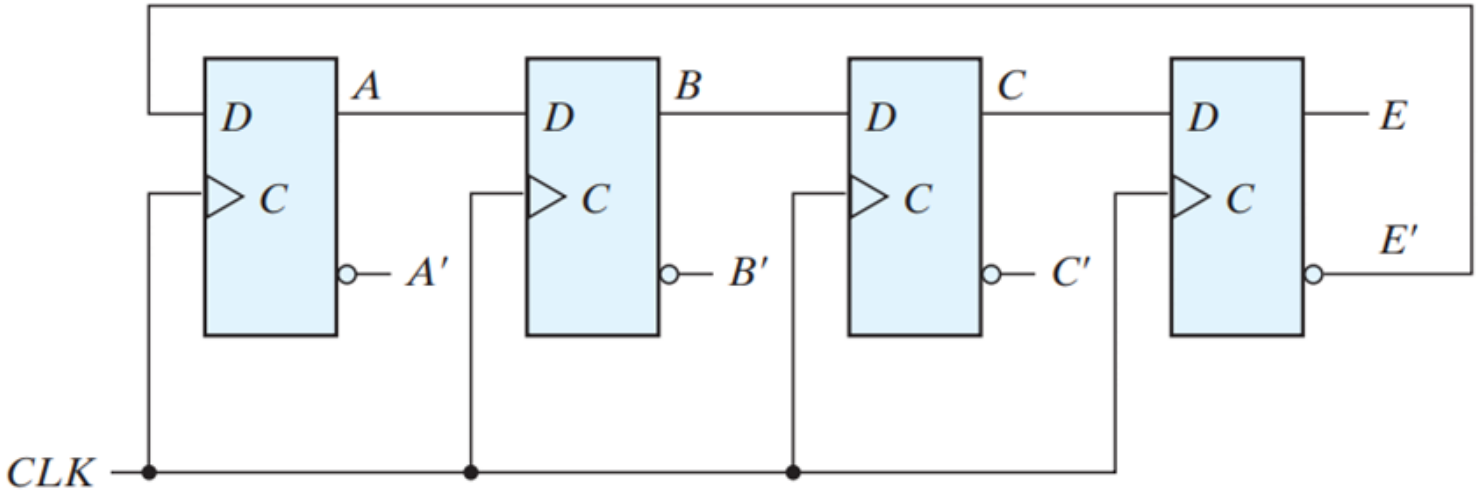
n states



2n states

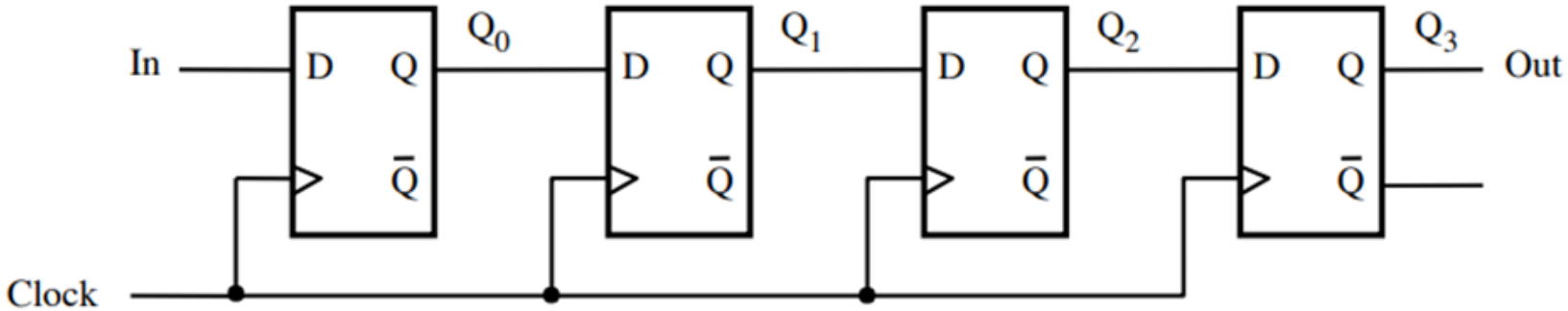
Johnson Counter

A *k*-bit ring counter circulates a single bit among the flip-flops to provide *k* distinguishable states. The number of states can be doubled if the shift register is connected as a *switch-tail* ring counter. A switch-tail ring counter is a circular shift register with the complemented output of the last flip-flop connected to the input of the first flip-flop. Figure 6.18(a) shows such a shift register. The circular connection is made from the



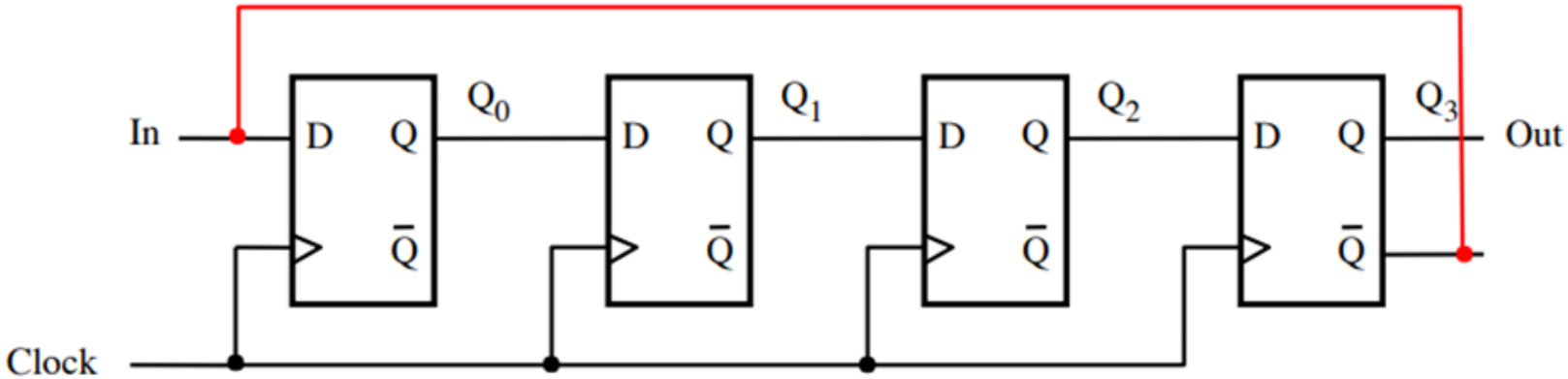
(a) Four-stage switch-tail ring counter

How to build a 4-bit Johnson counter



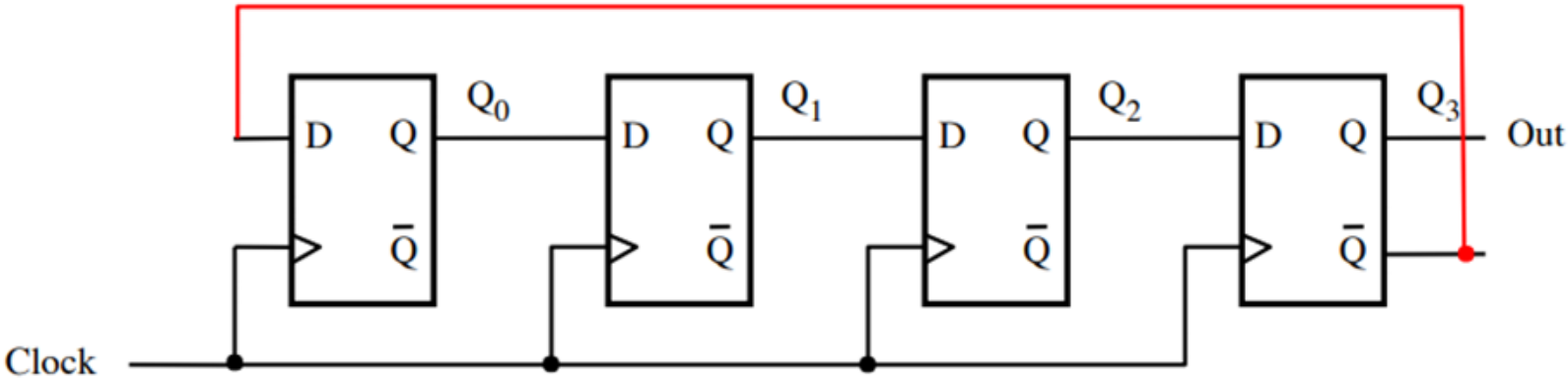
To build a Johnson counter start with a shift register.

How to build a 4-bit Johnson counter



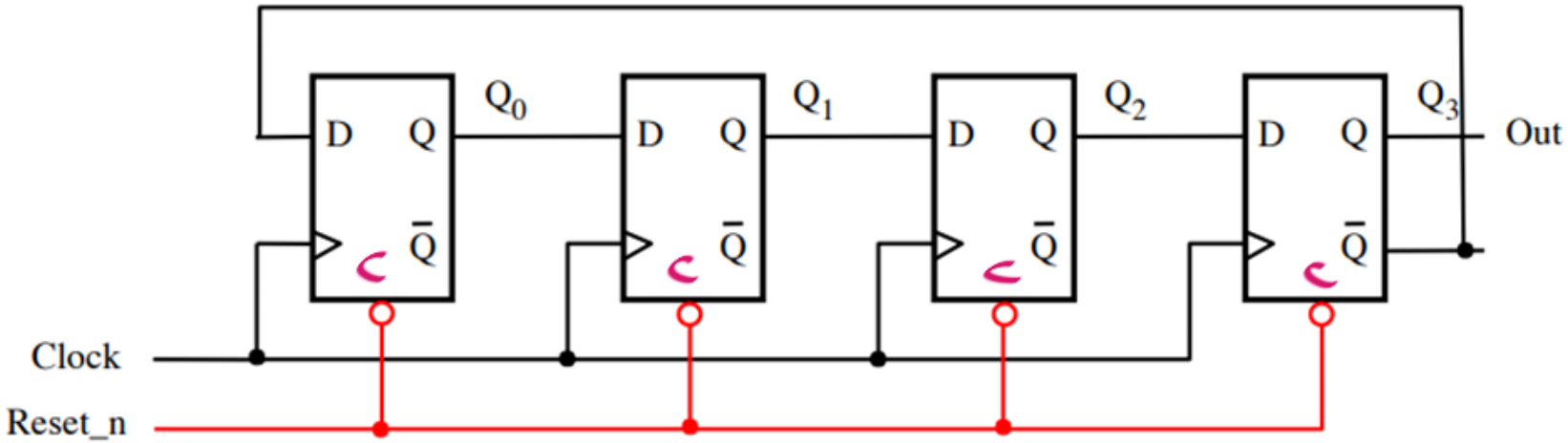
Next, add a loop from the $\overline{Q}$ output of the last flip-flop to the first…

How to build a 4-bit Johnson counter



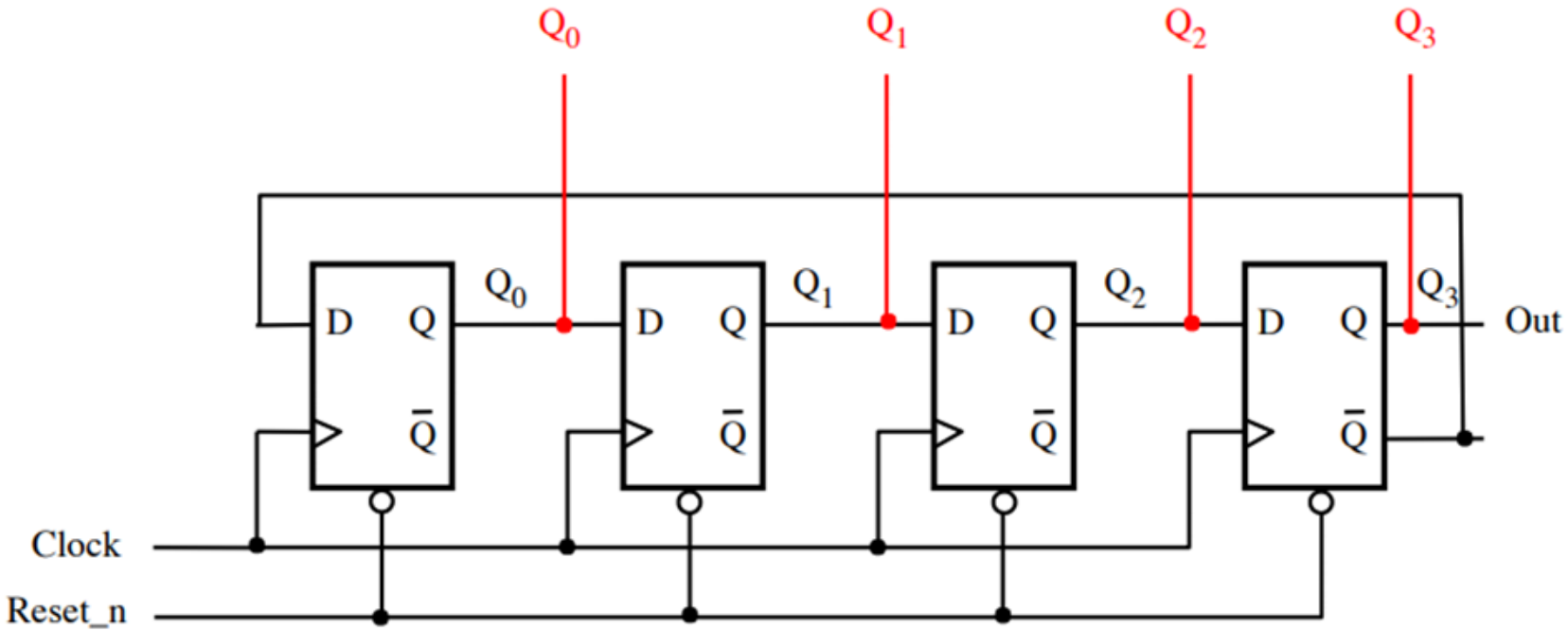
... and remove the In input line.

How to build a 4-bit Johnson counter



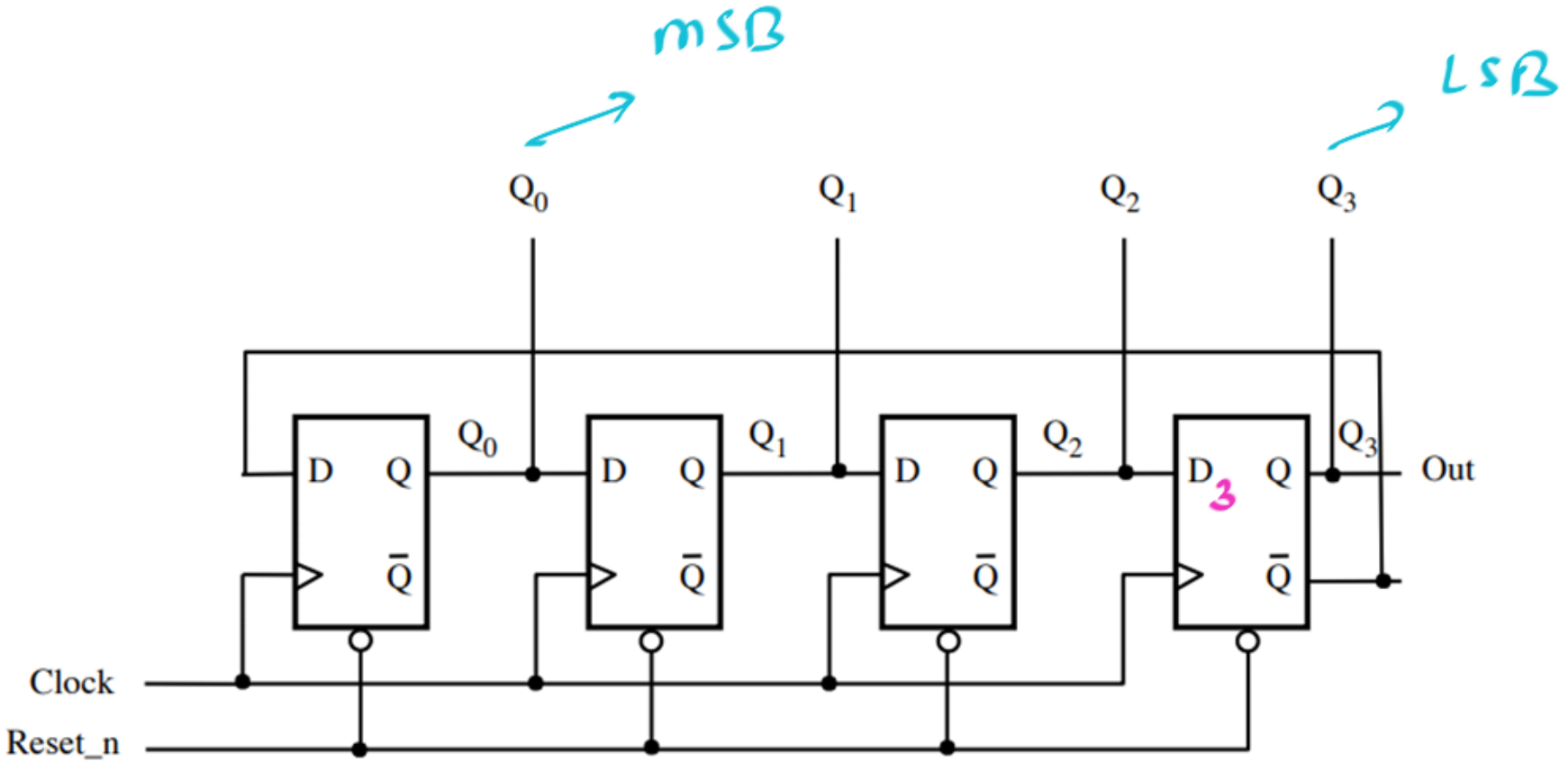
Also, add a Reset_n line that goes to clear_n of all flip-flops.

How to build a 4-bit Johnson counter

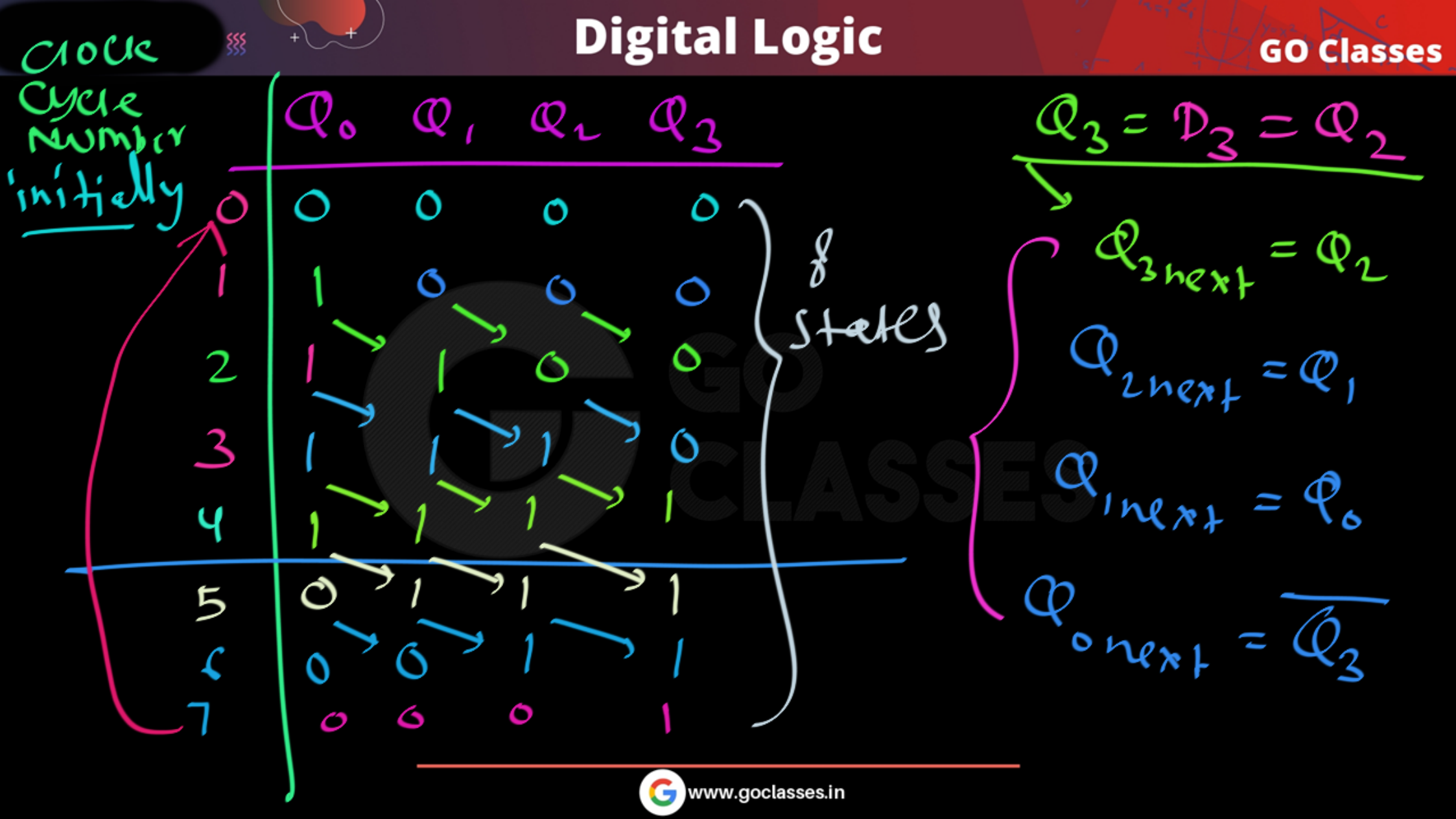


Finally, extend the output lines that form the count number.

How to build a 4-bit Johnson counter



This is the final circuit diagram.

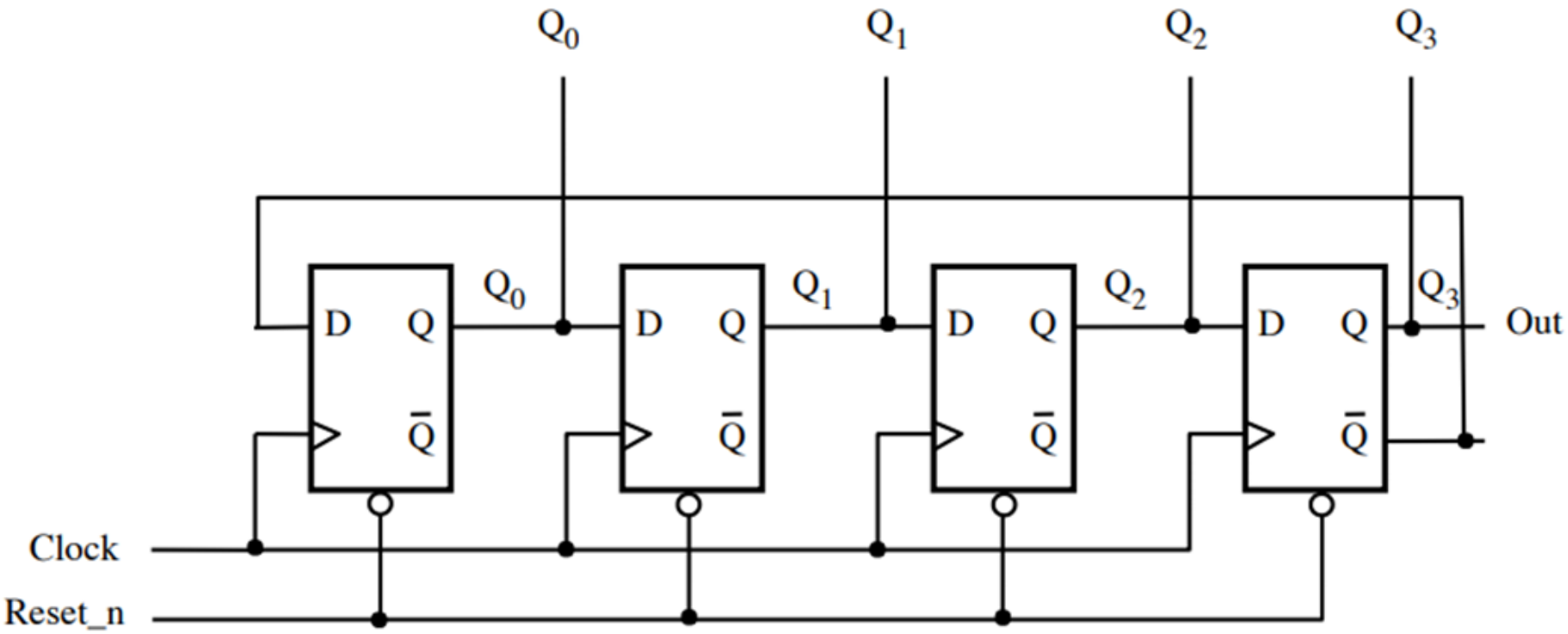

Clock Cycle Number initially
Q_0 Q_1 Q_2 Q_3
0 0 0 0 0
1 1 0 0 0
2 1 1 0 0
3 1 1 1 0
4 1 1 1 1
5 0 1 1 1
6 0 0 1 1
7 0 0 0 1
8 states
$Q_3 = D_3 = Q_2$
$Q_{3next} = Q_2$
$Q_{2next} = Q_1$
$Q_{1next} = Q_0$
$Q_{0next} = \overline{Q_3}$

4-Bit Johnson Counter

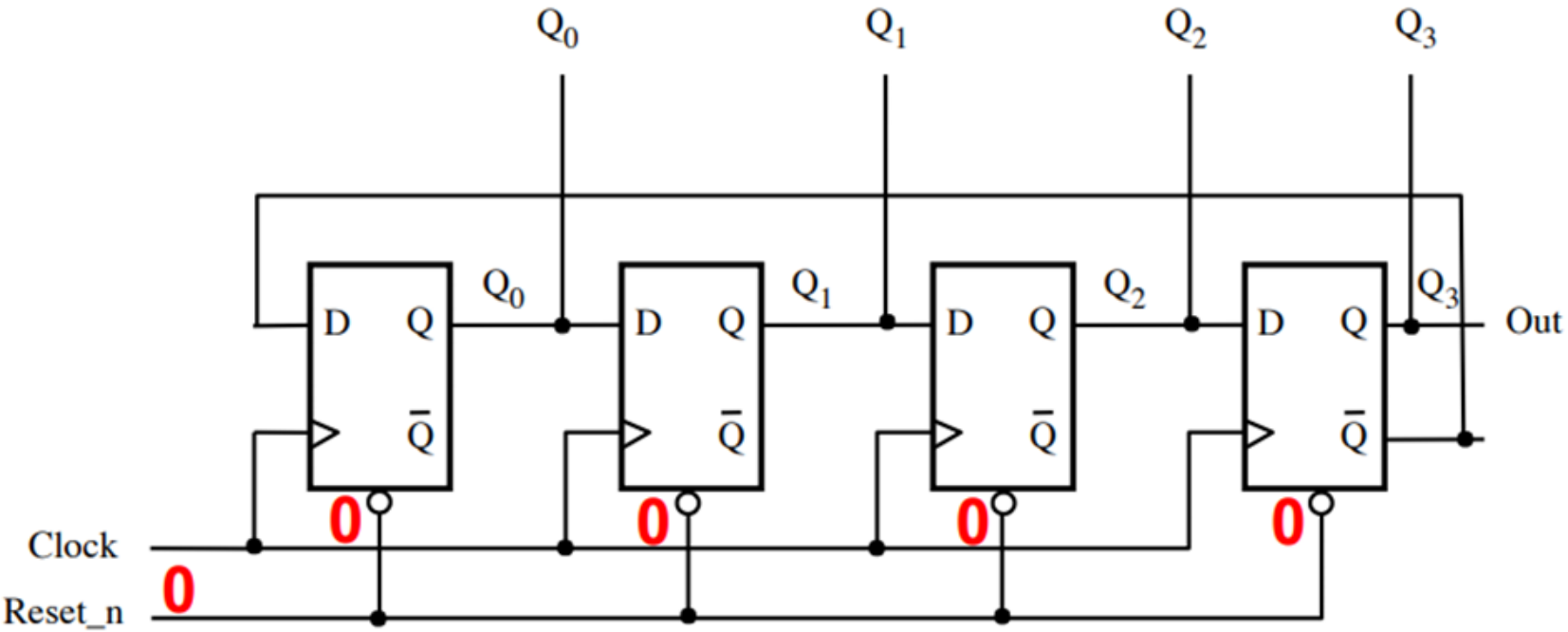
- Only 1-bit changes at a time
- Start with a reset of all flip-flops
- The counting sequence is:
  0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001, 0000
- An n-bit Johnson counter has a counting sequence of length 2n

Johnson Counter Working

Initialization: How does it work?

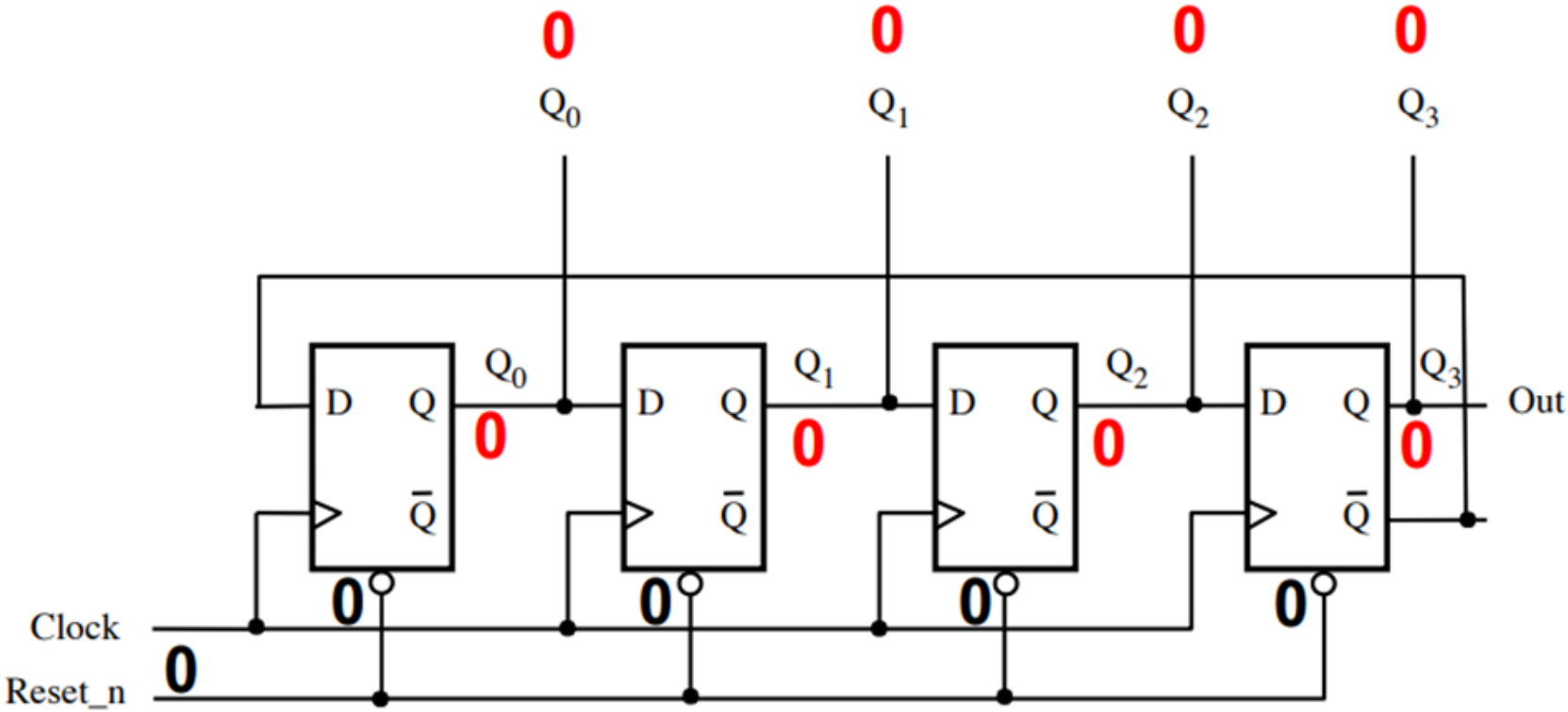


Initialization: How does it work?



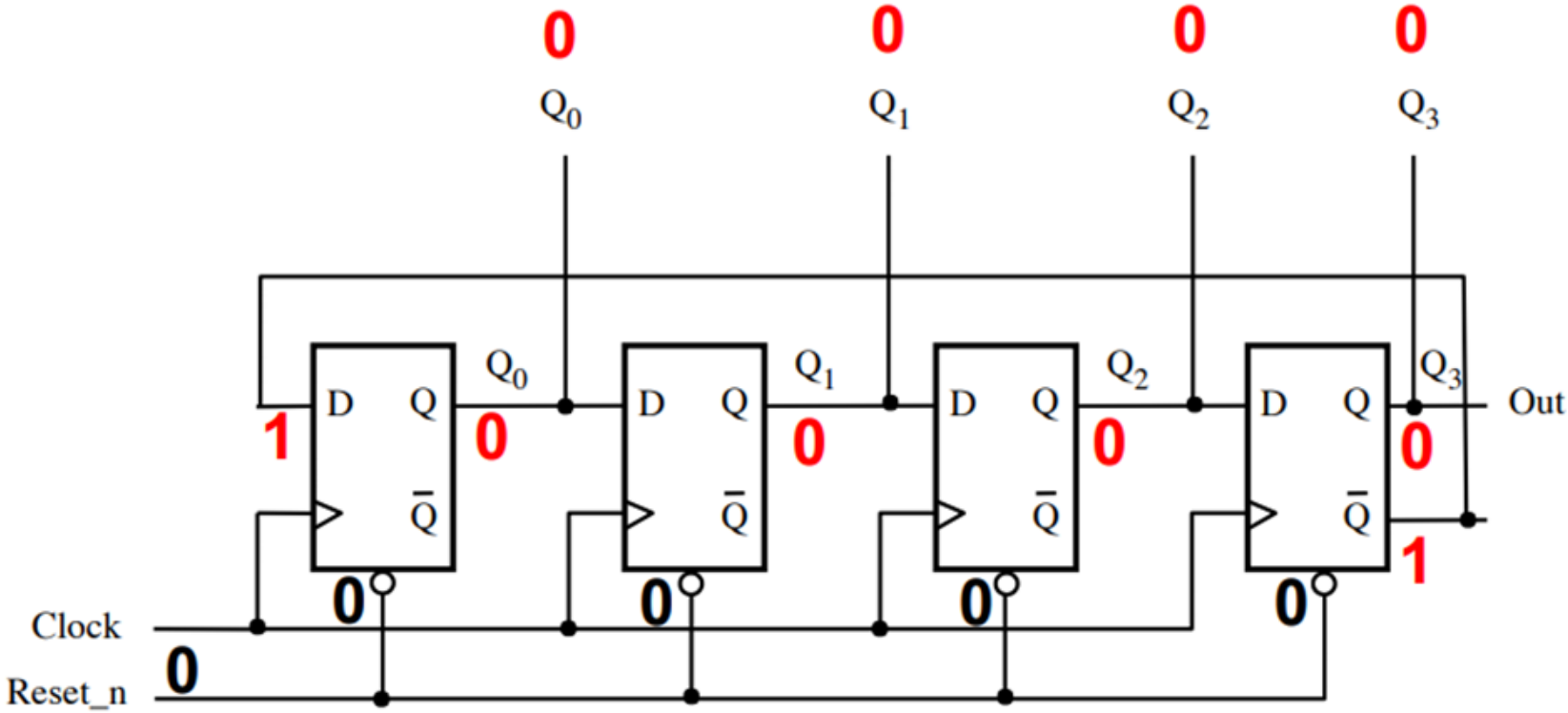
To initialize the Johnson counter set Reset_n to 0 ...

Initialization: How does it work?



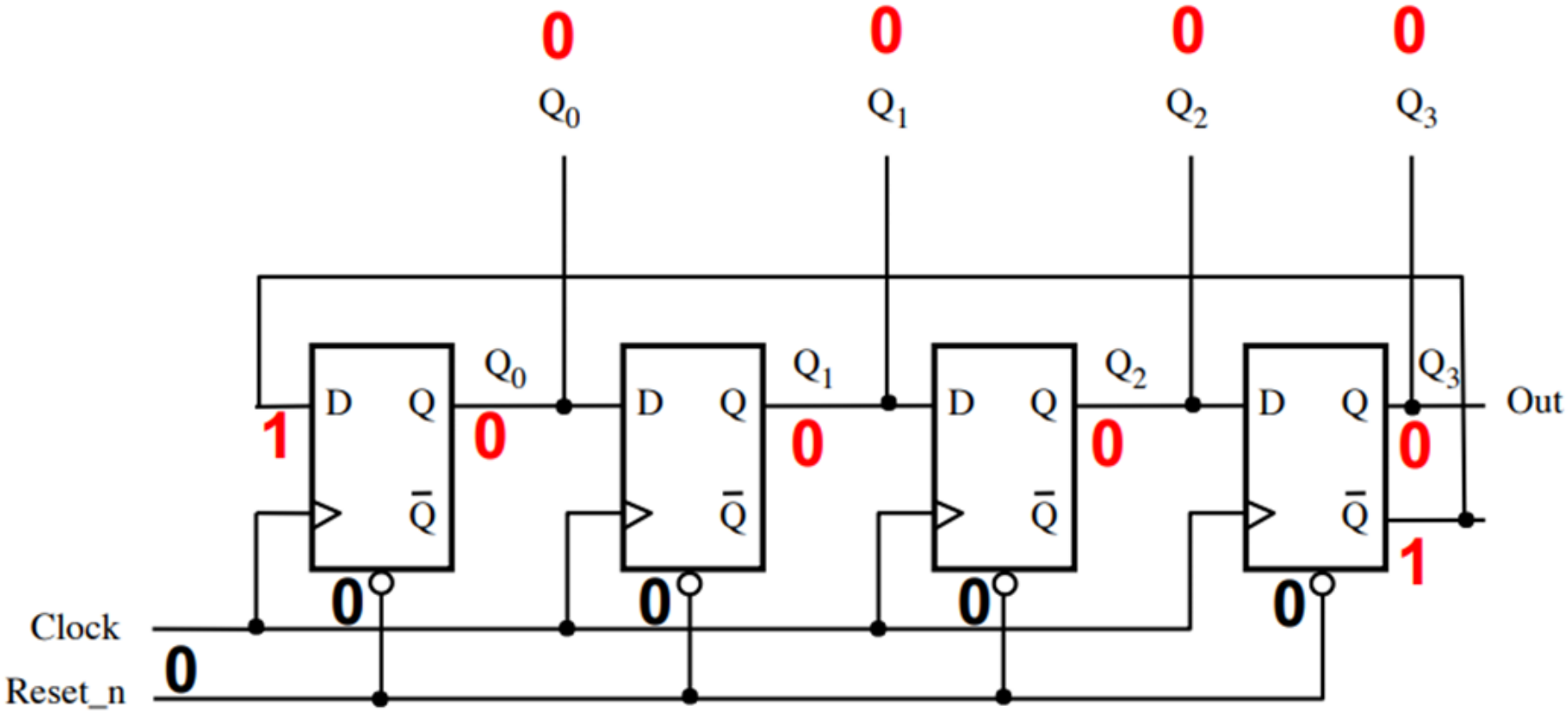
... this zeros the outputs of all flip-flops ...

Initialization: How does it work?

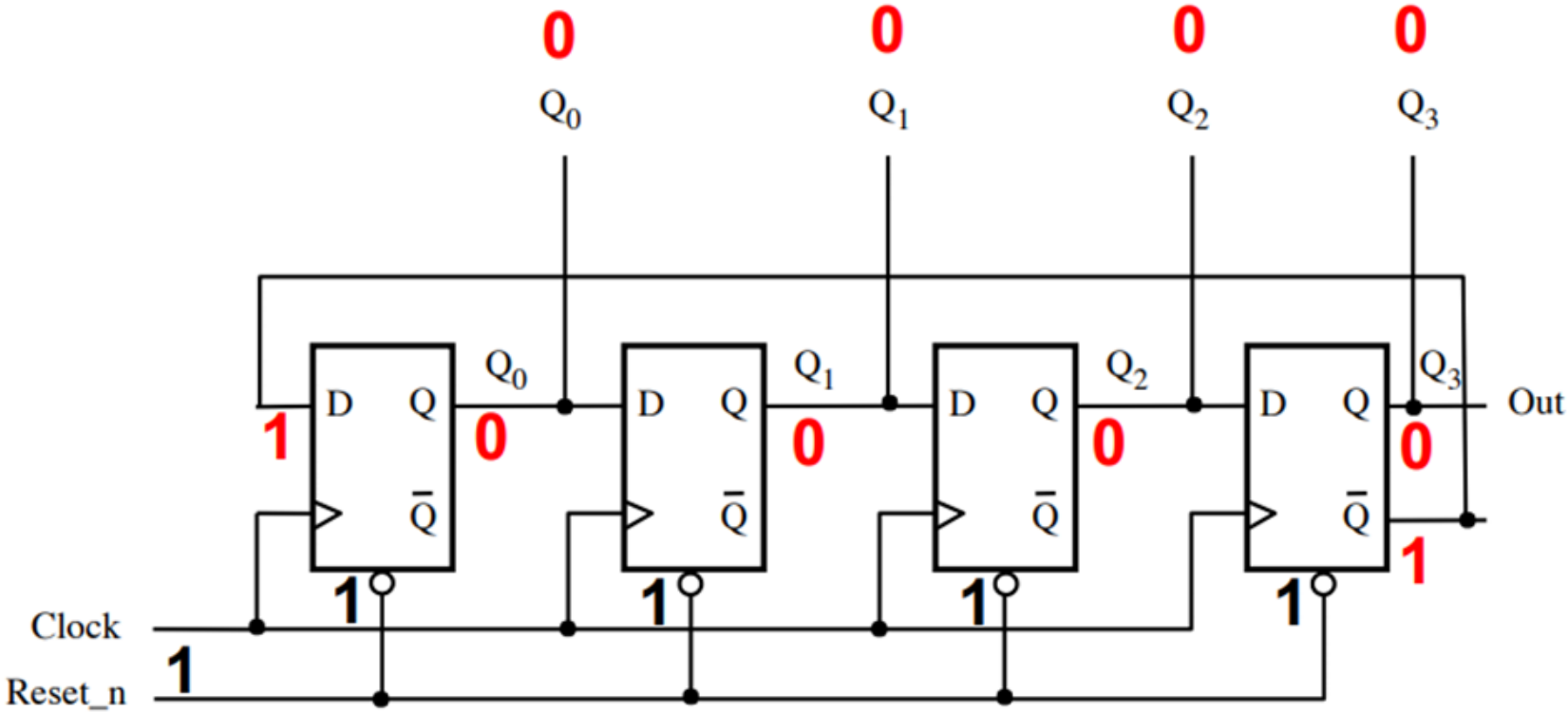


... and also sets the $\overline{Q}$ output of the last flip-flop to 1.

Counting: How does it work?

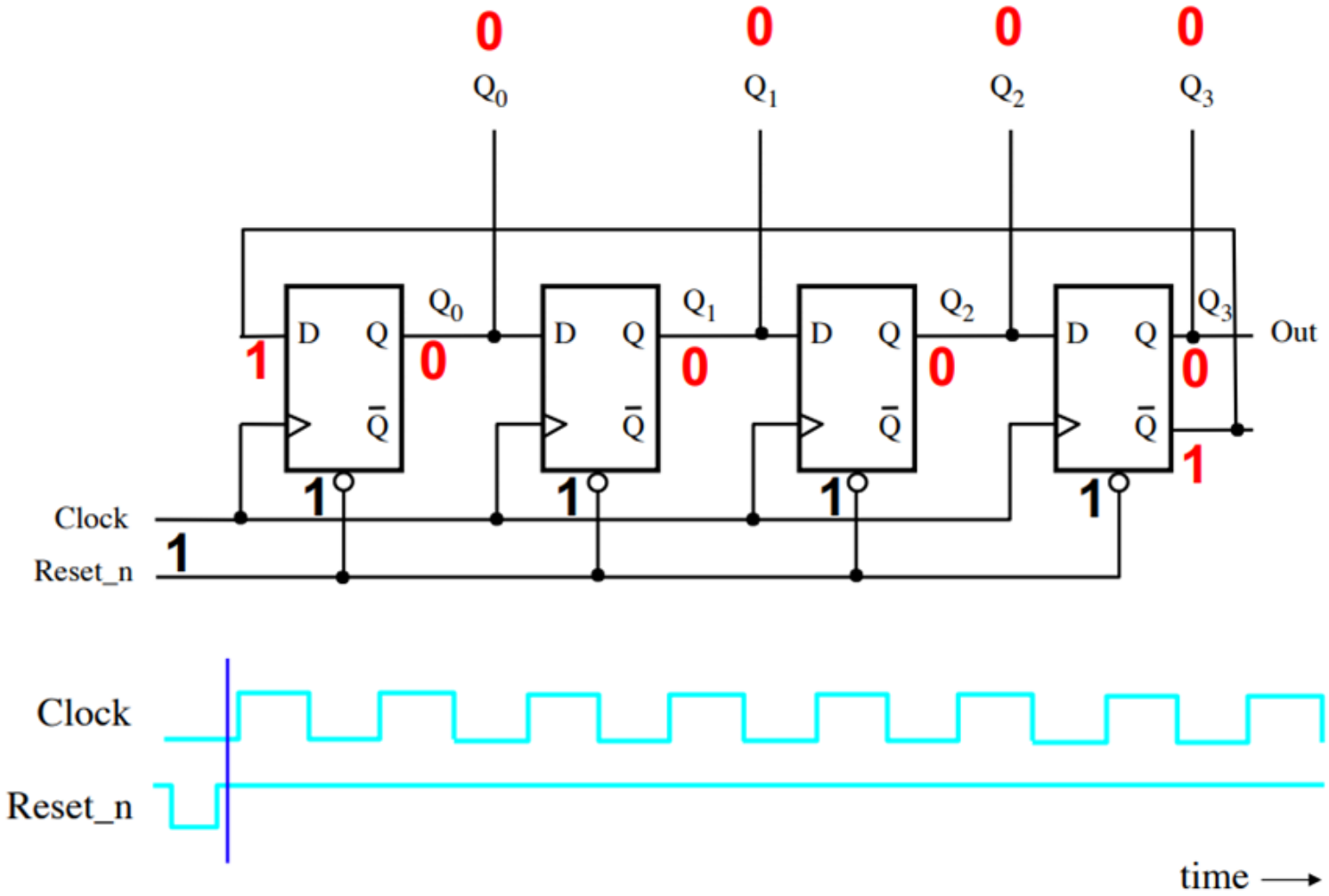


Counting: How does it work?

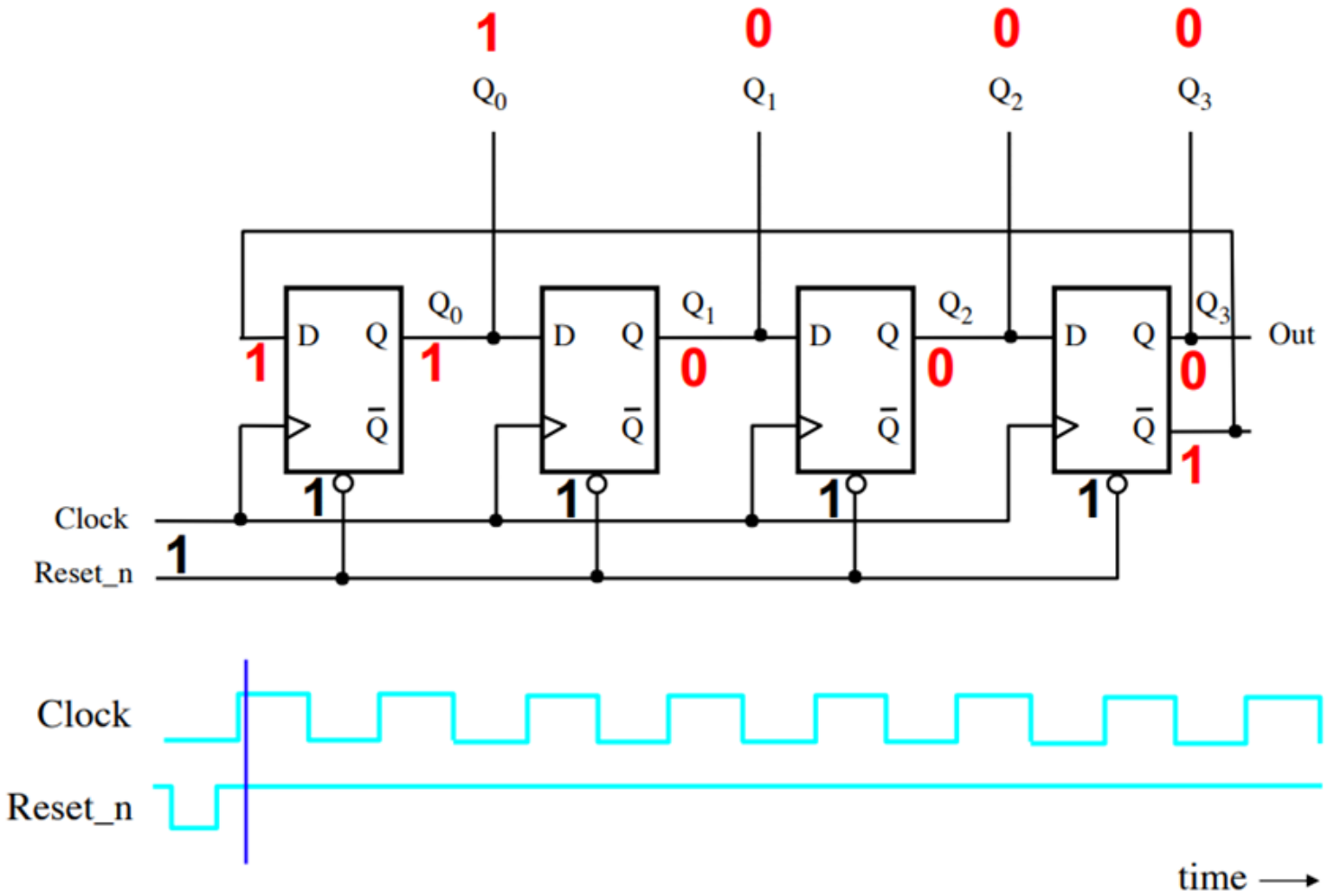


To start counting, Reset_n needs to be set to 1.

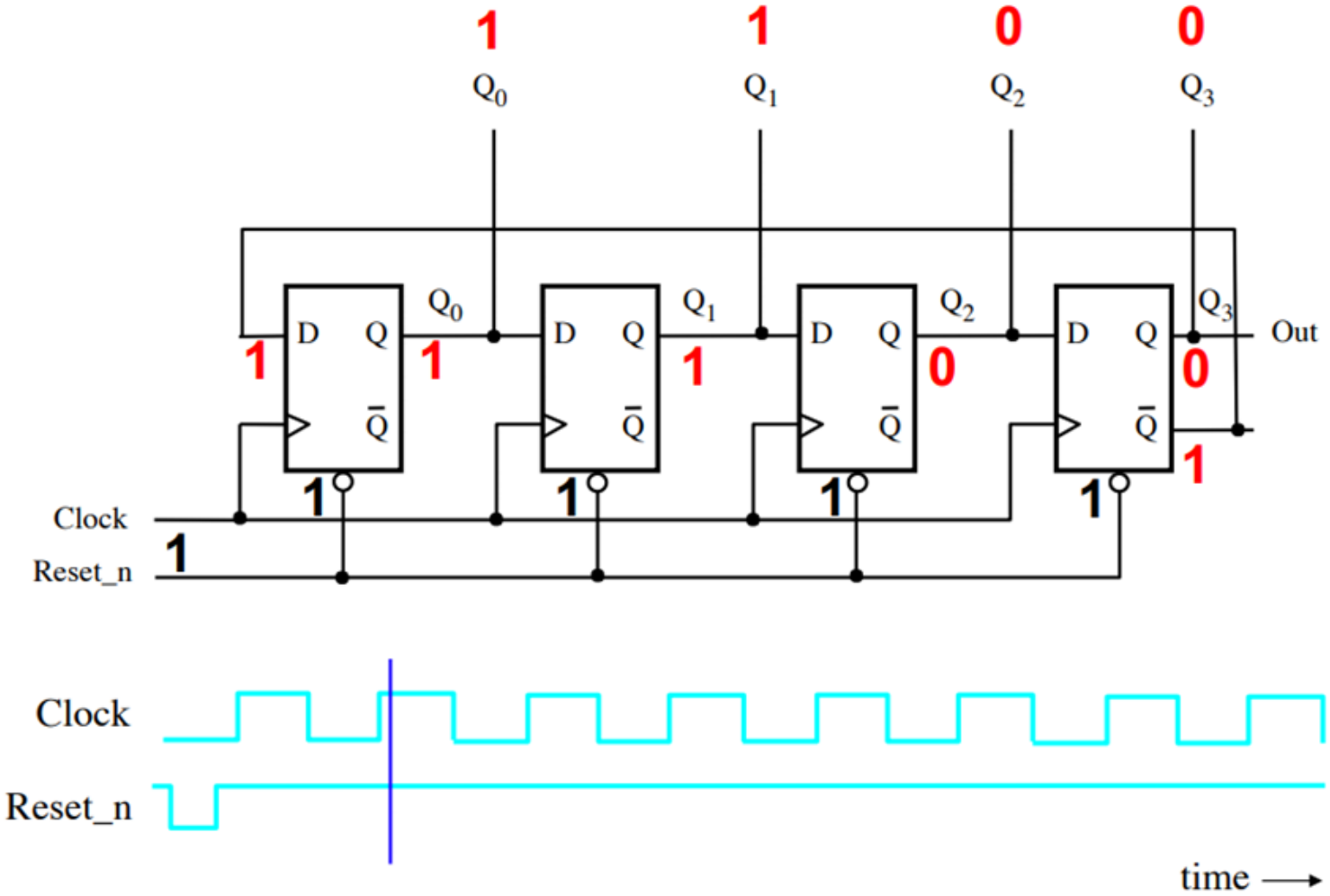
Counting: How does it work?



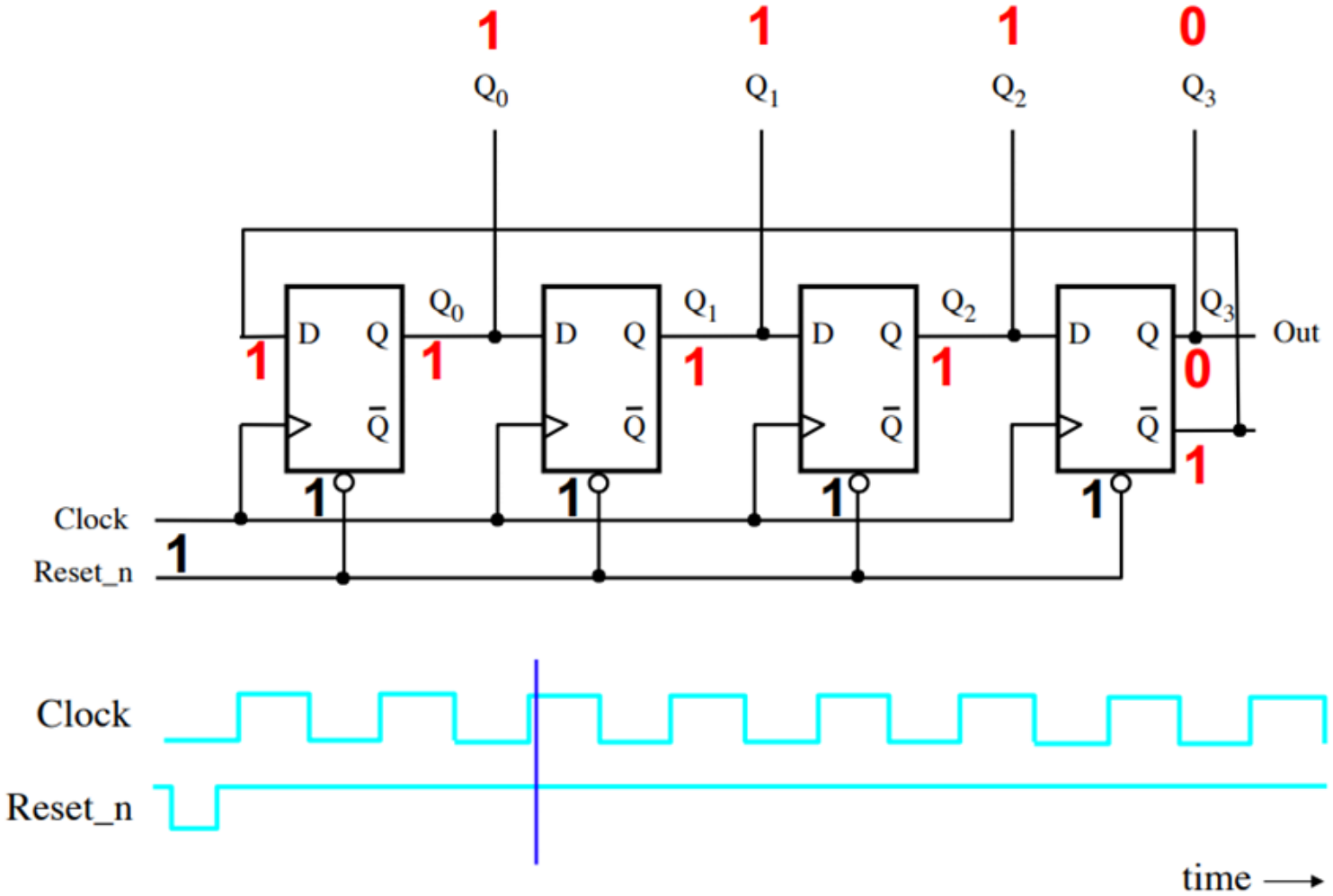
Counting: How does it work?



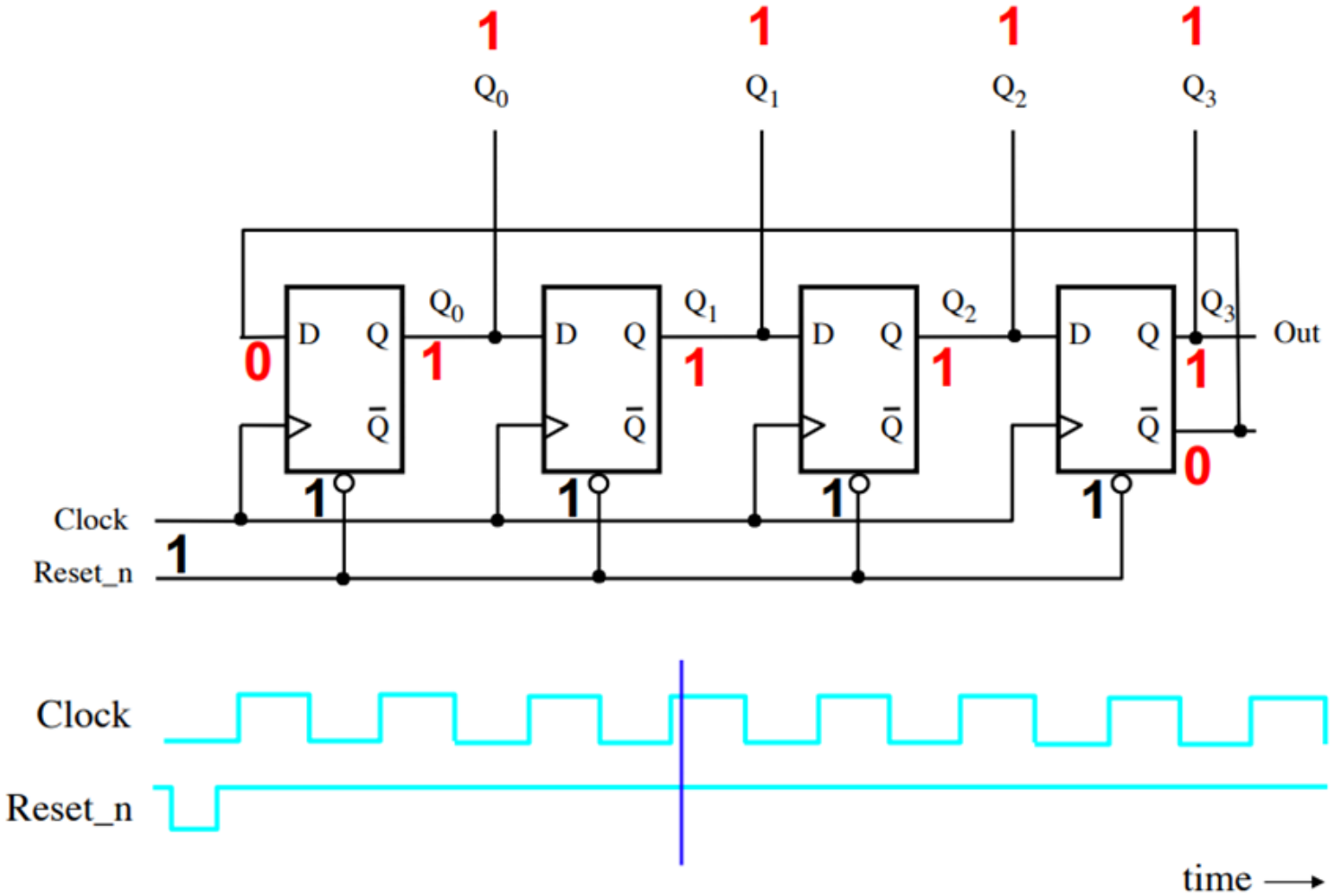
Counting: How does it work?



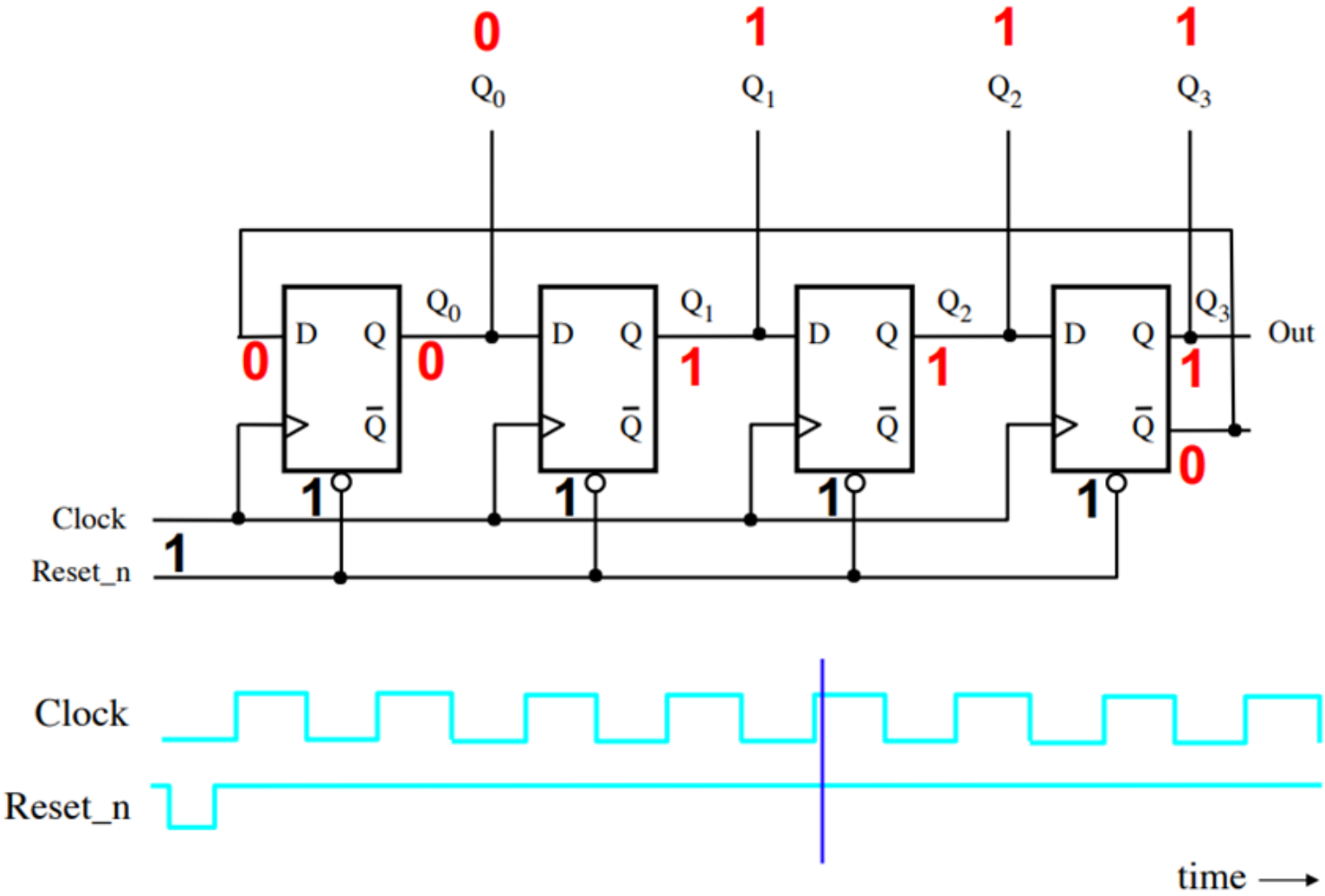
Counting: How does it work?



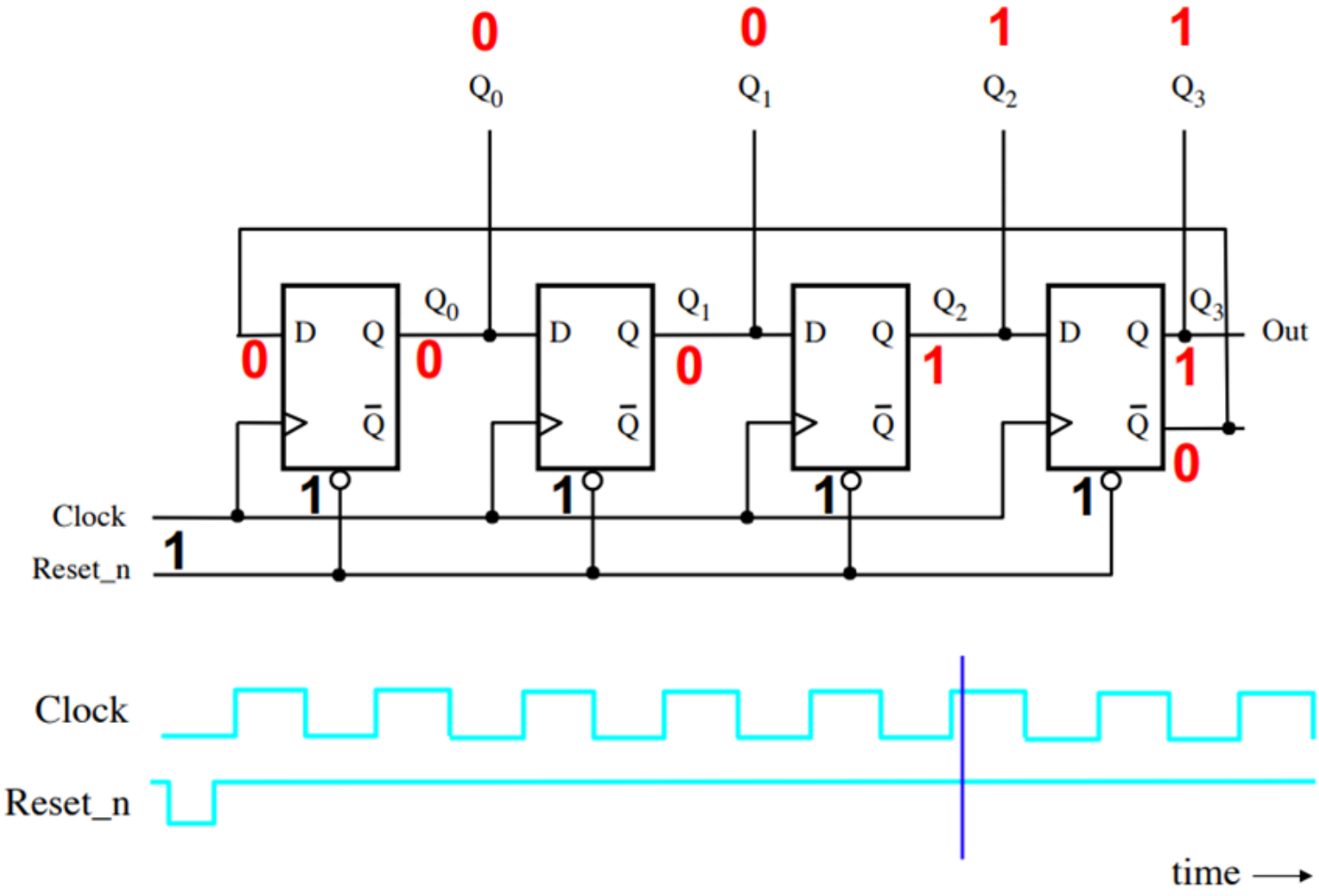
Counting: How does it work?



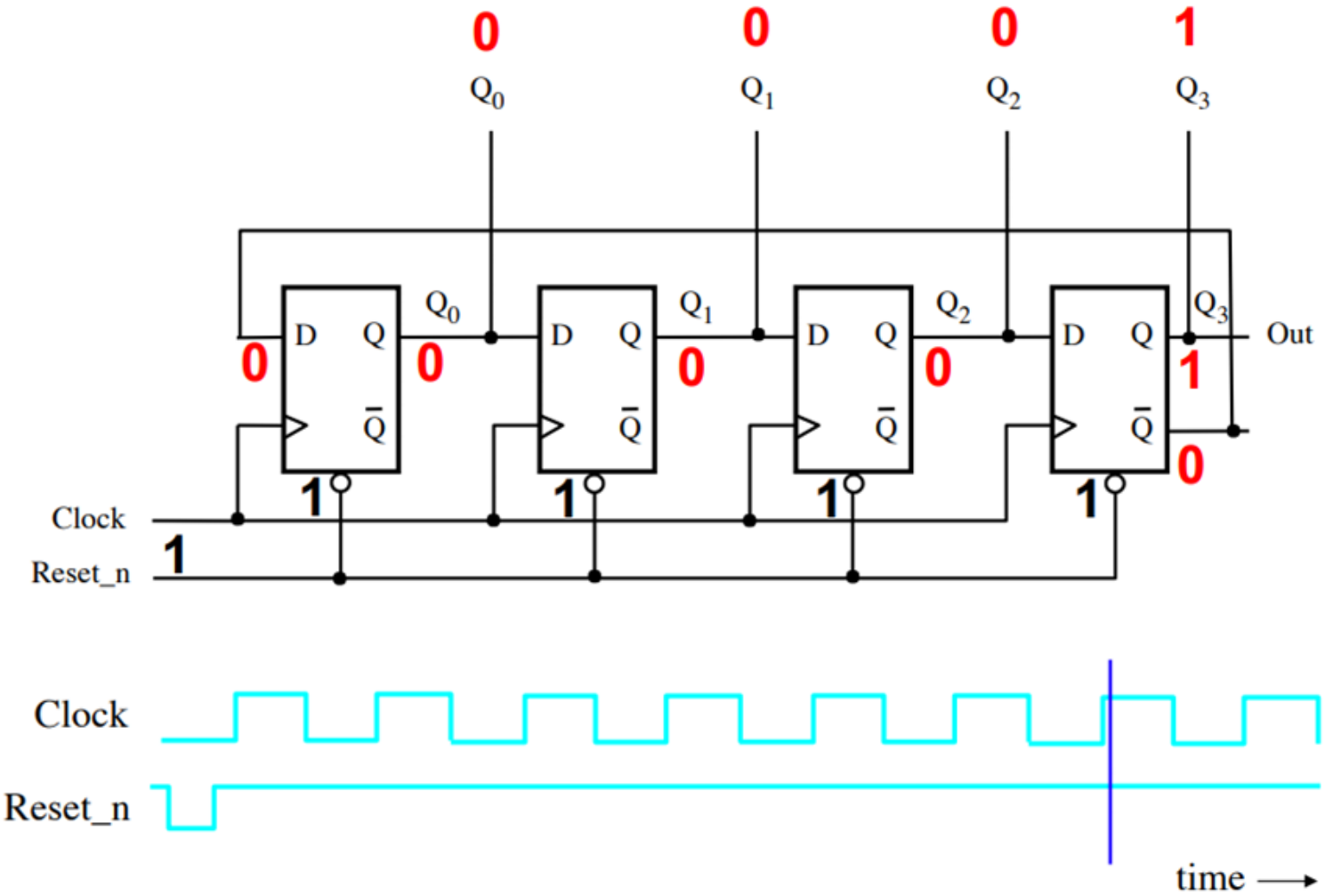
Counting: How does it work?



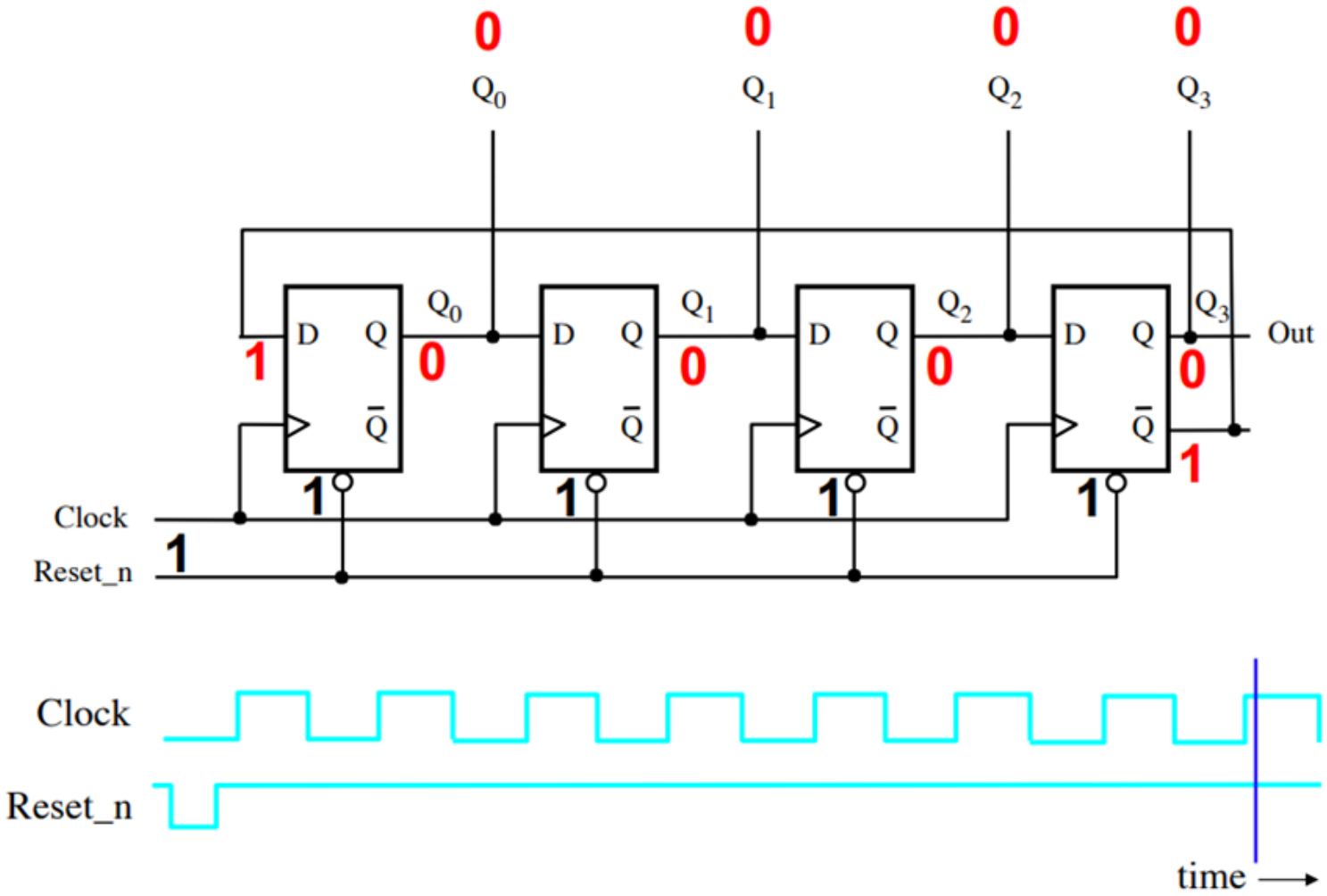
Counting: How does it work?



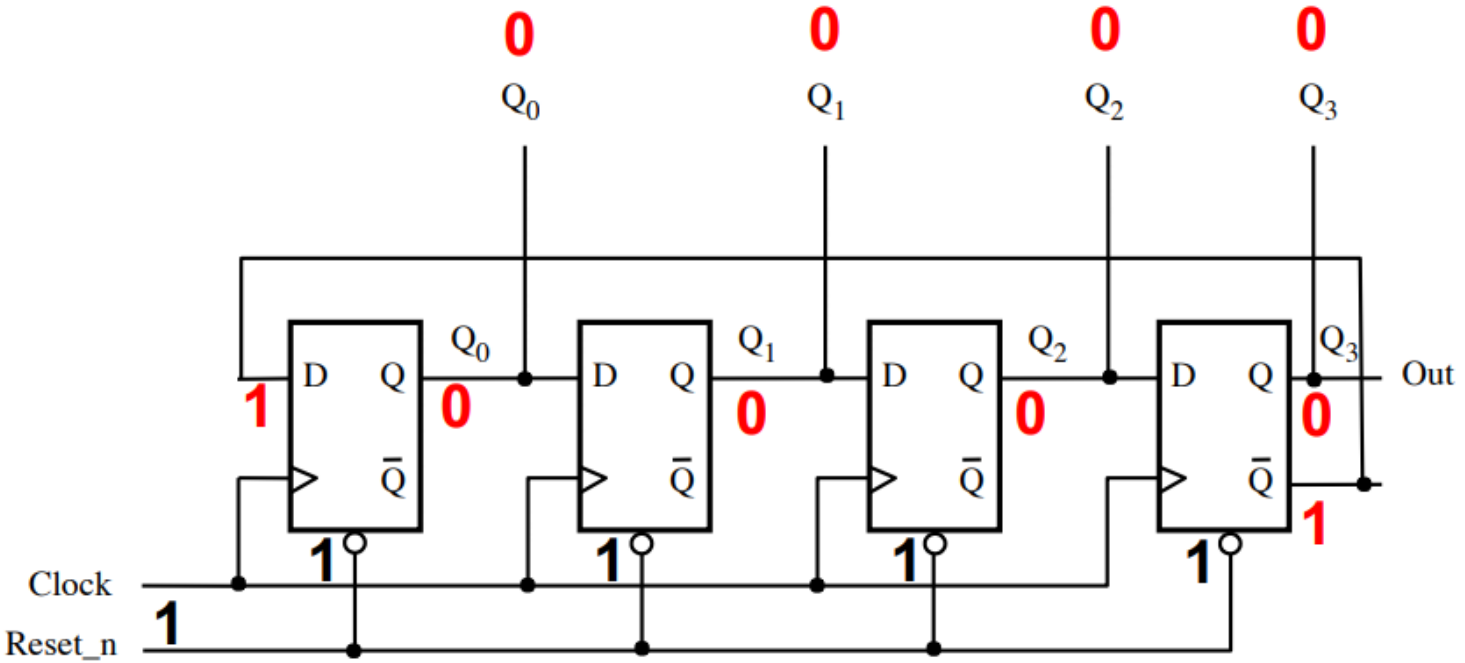
Counting: How does it work?



Counting: How does it work?

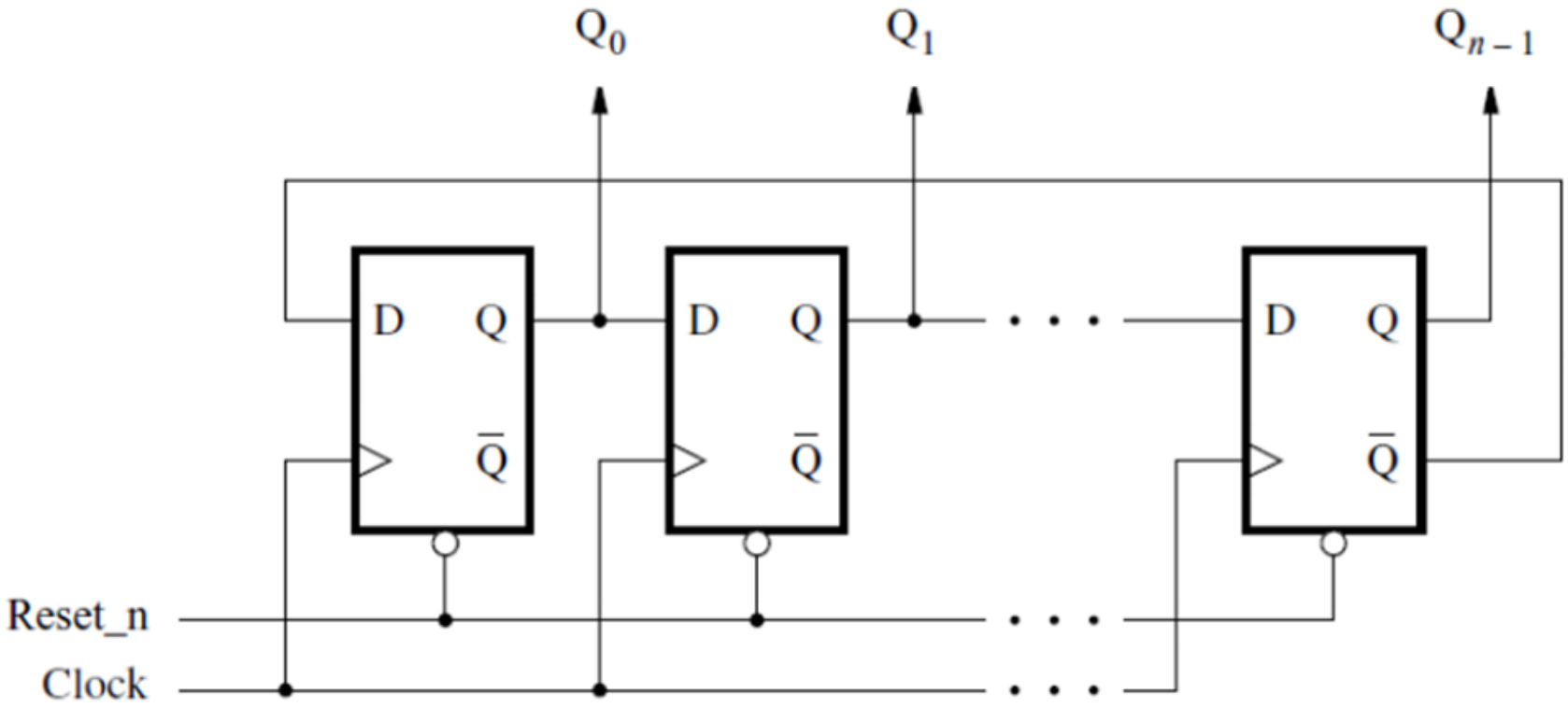


Counting: How does it work?

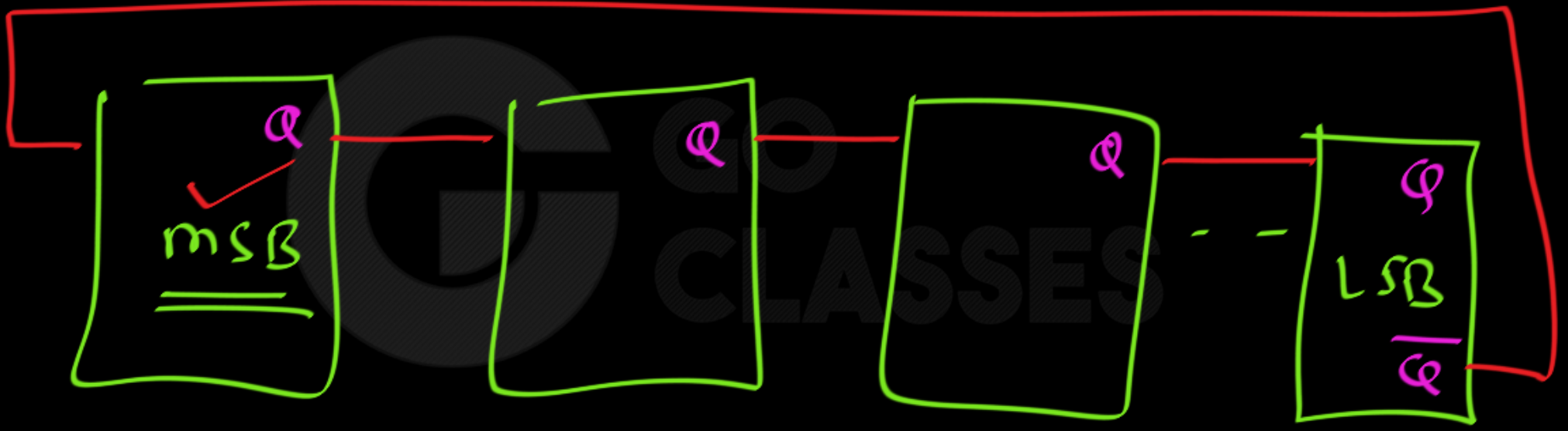


It is back to the start of the counting sequence, which is:
0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001.

n-bit Johnson Counter



Conclusion of Twisted Ring Counter :



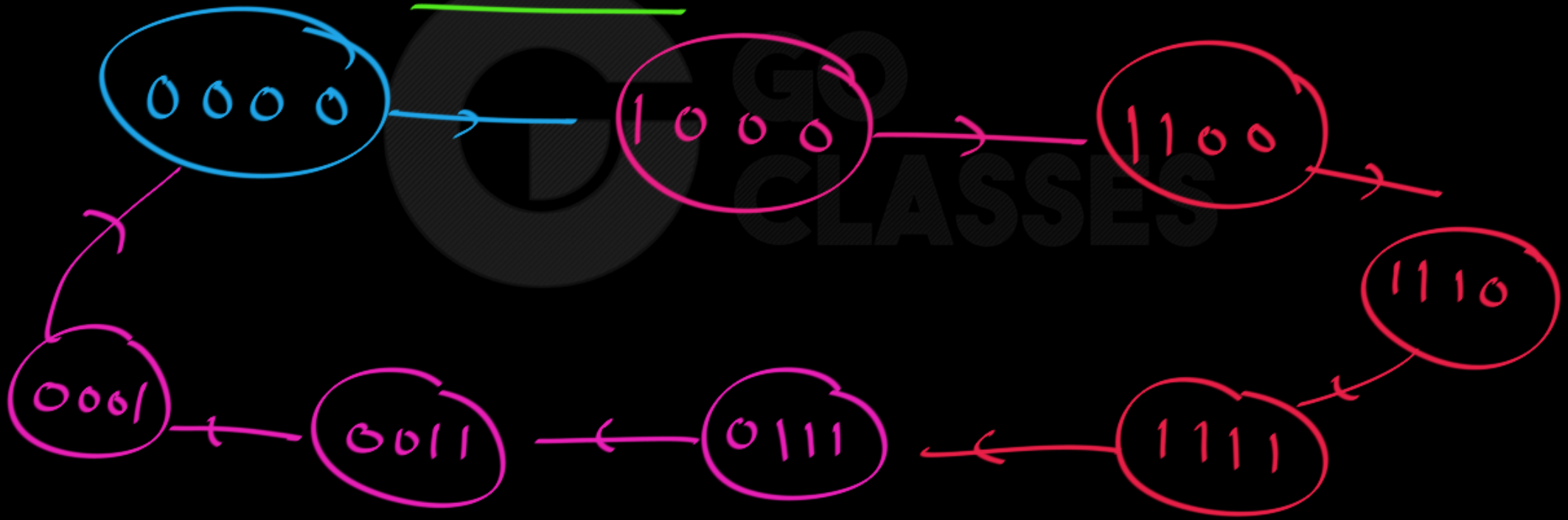
Conclusion of Twisted Ring Counter:

(1) initialization: 0000

(2) n - FFs ⇒ 2n states (Double of Ring Counter)

Conclusion of Twisted Ring Counter:

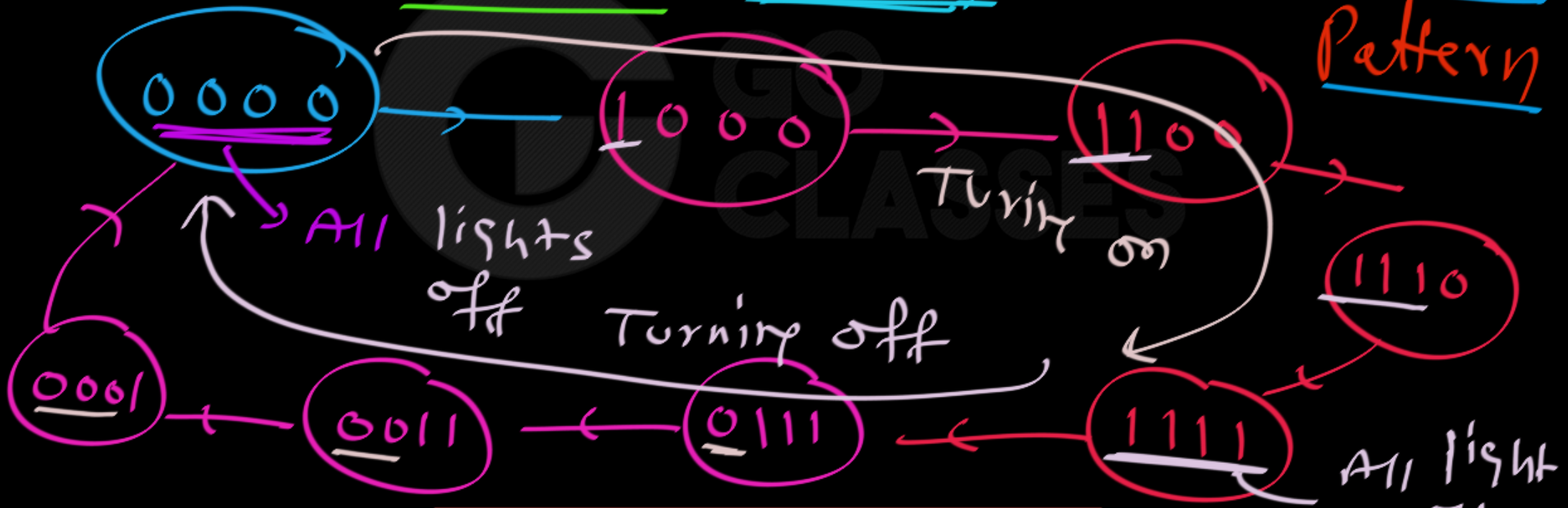
Sequence : 4-FFs



Conclusion of Twisted Ring Counter:

Sequence: 4-FFs

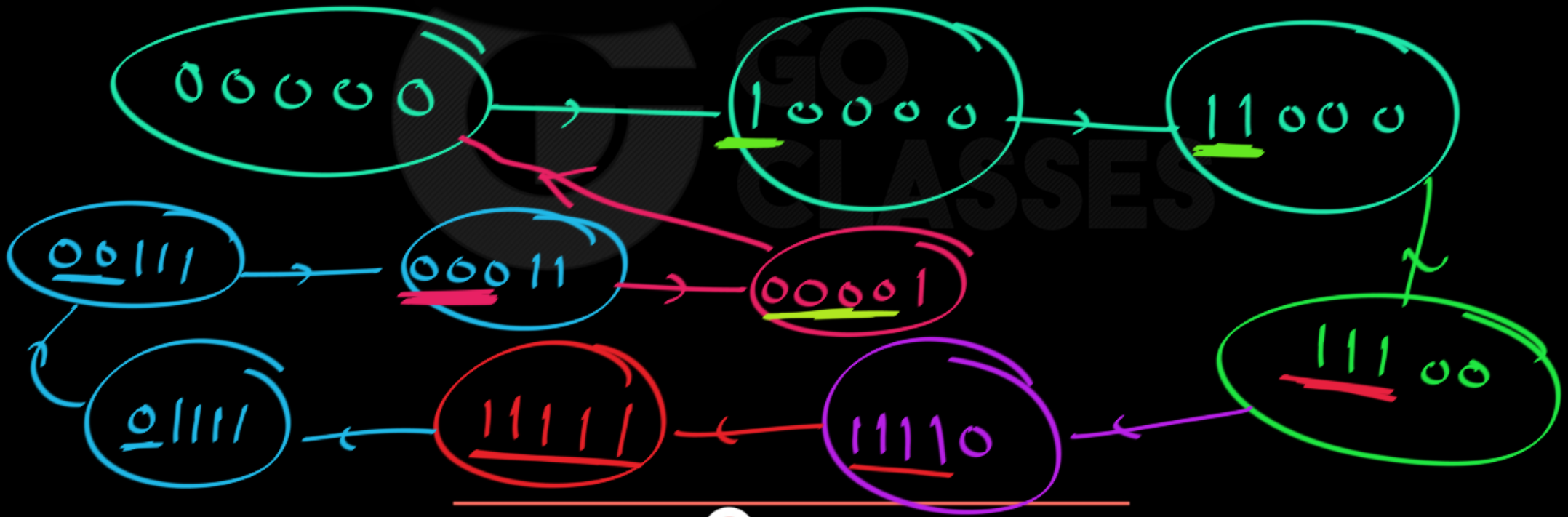
Analogy: Diwali lights Pattern



Conclusion of Twisted Ring Counter:

Sequence:

5-ffs:



Conclusion of Twisted Ring Counter:

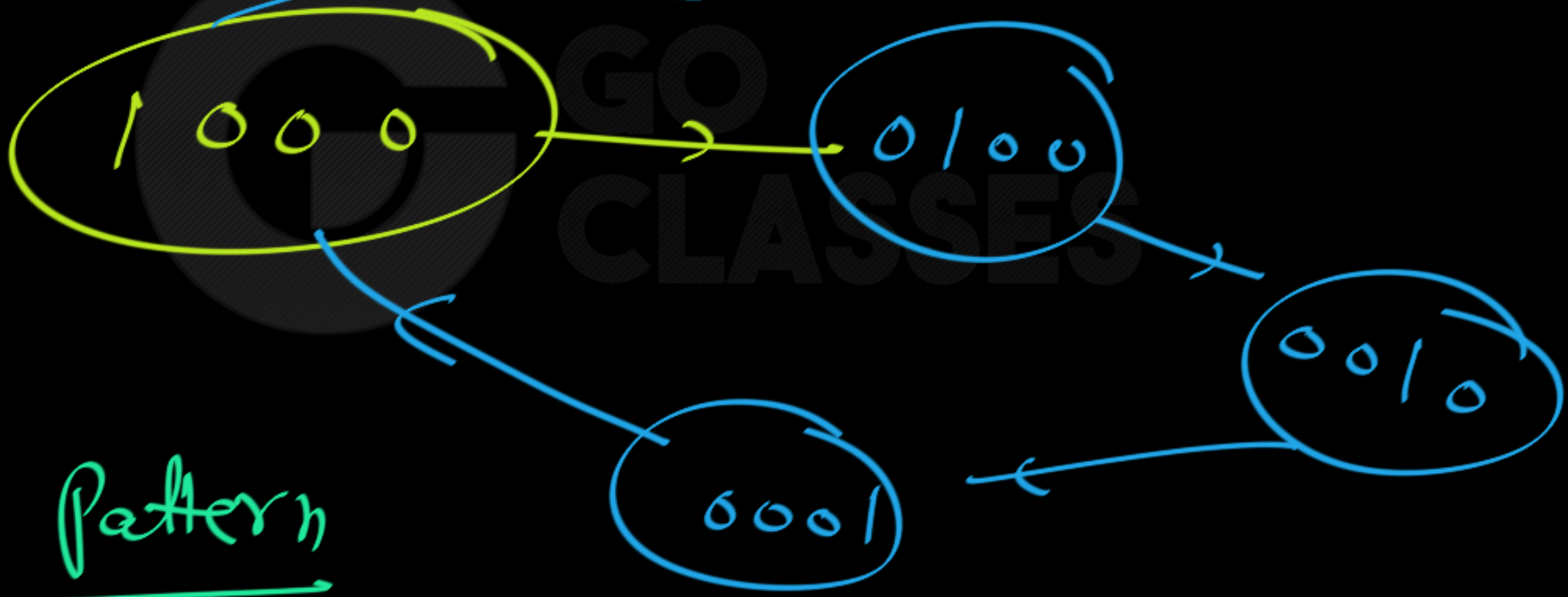
Sequence:

5-ffs:

Conclusion of Ring Counter:

4-FFs:

initial



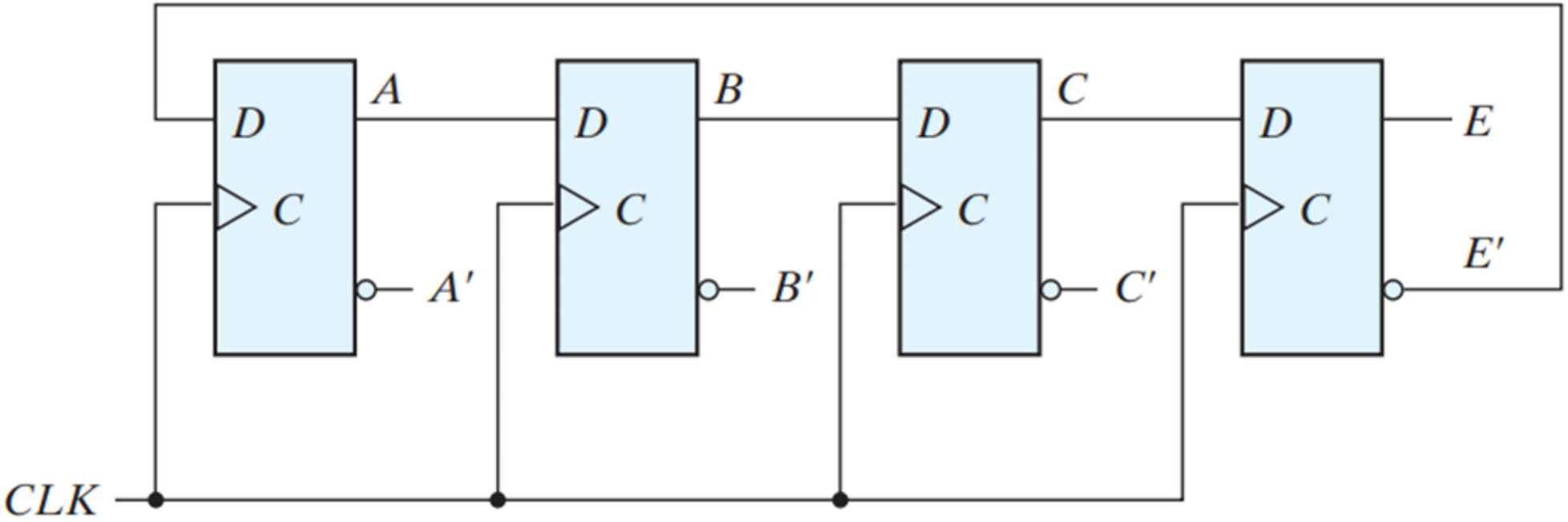
Another Diwali Lights Pattern

A NOTE on Johnson Counter

Twisted Ring Counter
or
Switch-Tail ring counter
or
Mobius Counter

} Same

Different Names of Same thing



(a) Four-stage switch-tail ring counter

Twisted Ring Counter

Is "Johnson counter" Same as Twisted Ring Counter ?

Are these two same Or is there any difference??

We will see Now....

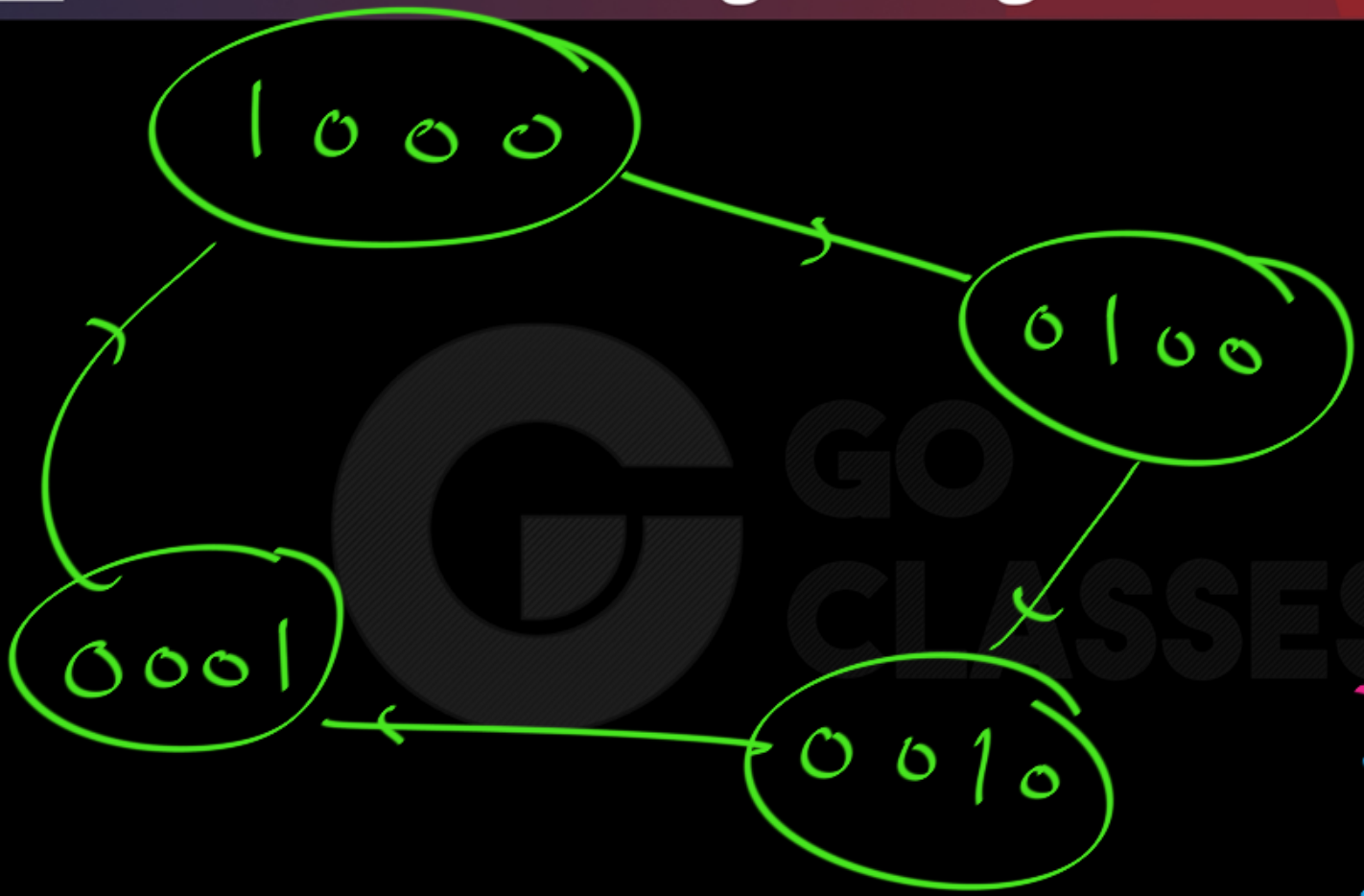
Ring Counter

- Straight Ring Counter
- Johnson Counter

by Default; Ring Counter ≡ Straight Ring Counter

Recall (Ring/Johnson Counter)

The Purpose

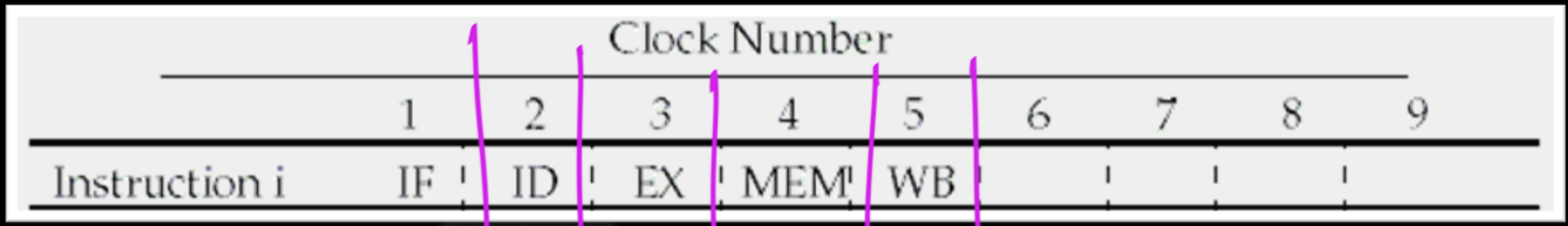


Purpose :

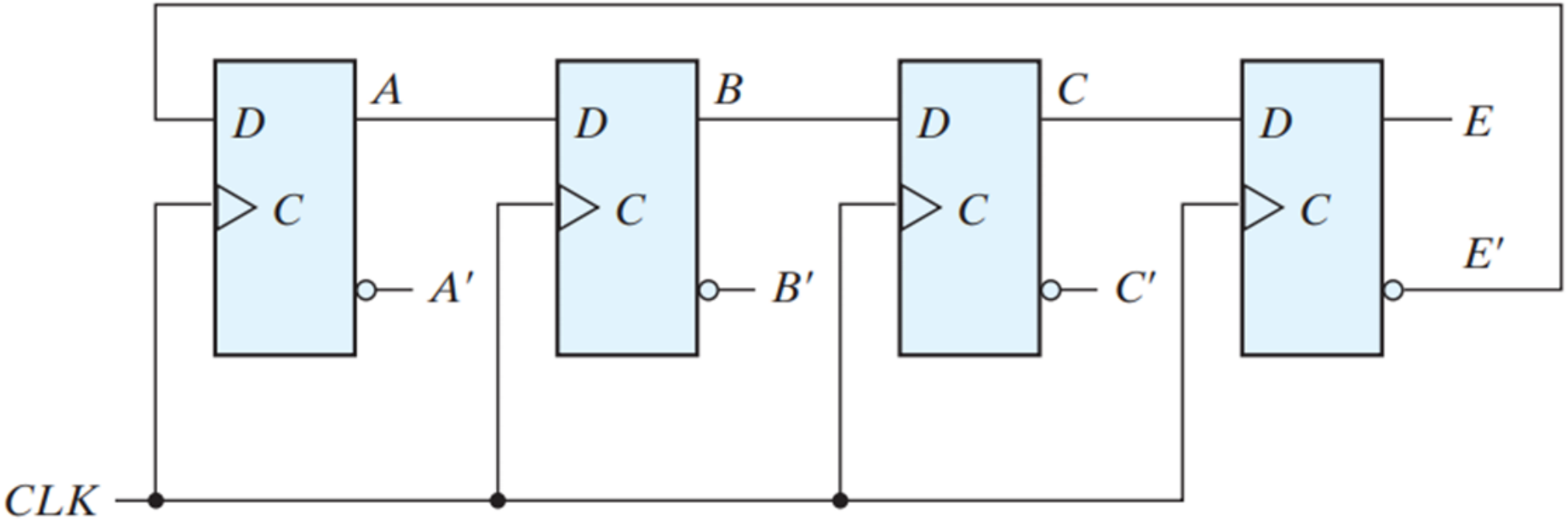
↓ only one circuit turn on at any time

Straight Ring Counter :

→ Purpose fulfilled ✓



Purpose of Ring Counter



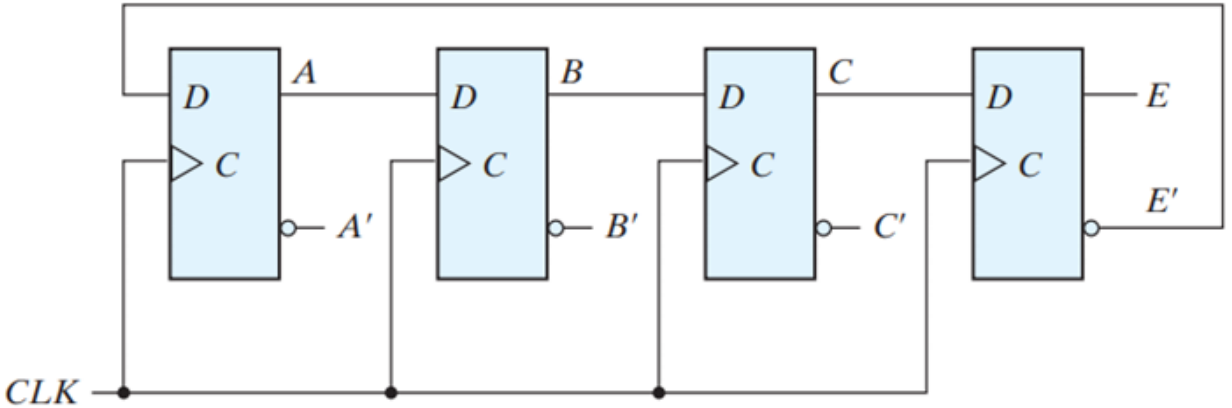
(a) Four-stage switch-tail ring counter

Twisted Ring Counter

Does Twisted Ring Counter Complete Our Purpose??? → No

Statement :

With n-ffs, Ring Counter generates n states; Johnson Counter generates 2n states.



(a) Four-stage switch-tail ring counter

Sequence number	Flip-flop outputs				AND gate required for output
	A	B	C	E	
1	0	0	0	0	$A'E'$
2	1	0	0	0	AB'
3	1	1	0	0	BC'
4	1	1	1	0	CE'
5	1	1	1	1	AE
6	0	1	1	1	$A'B$
7	0	0	1	1	$B'C$
8	0	0	0	1	$C'E$

(b) Count sequence and required decoding

FIGURE 6.18
Construction of a Johnson counter

A Johnson counter is a k-bit switch-tail ring counter with $2k$ decoding gates to provide outputs for $2k$ timing signals. The decoding gates are not shown in Fig. 6.18, but are specified in the last column of the table. The eight AND gates listed in the table, when connected to the circuit, will complete the construction of the Johnson counter. Since each gate is enabled during one particular state sequence, the outputs of the gates generate eight timing signals in succession.

The decoding of a k-bit switch-tail ring counter to obtain $2k$ timing signals follows a regular pattern. The all-0's state is decoded by taking the complement of the two extreme flip-flop outputs. The all-1's state is decoded by taking the normal outputs of the two extreme flip-flops. All other states are decoded from an adjacent 1, 0 or 0, 1 pattern in the sequence. For example, sequence 7 has an adjacent 0, 1 pattern in flip-flops B and C. The decoded output is then obtained by taking the complement of B and the normal output of C, or $B'C$.

Source: Morris Mano

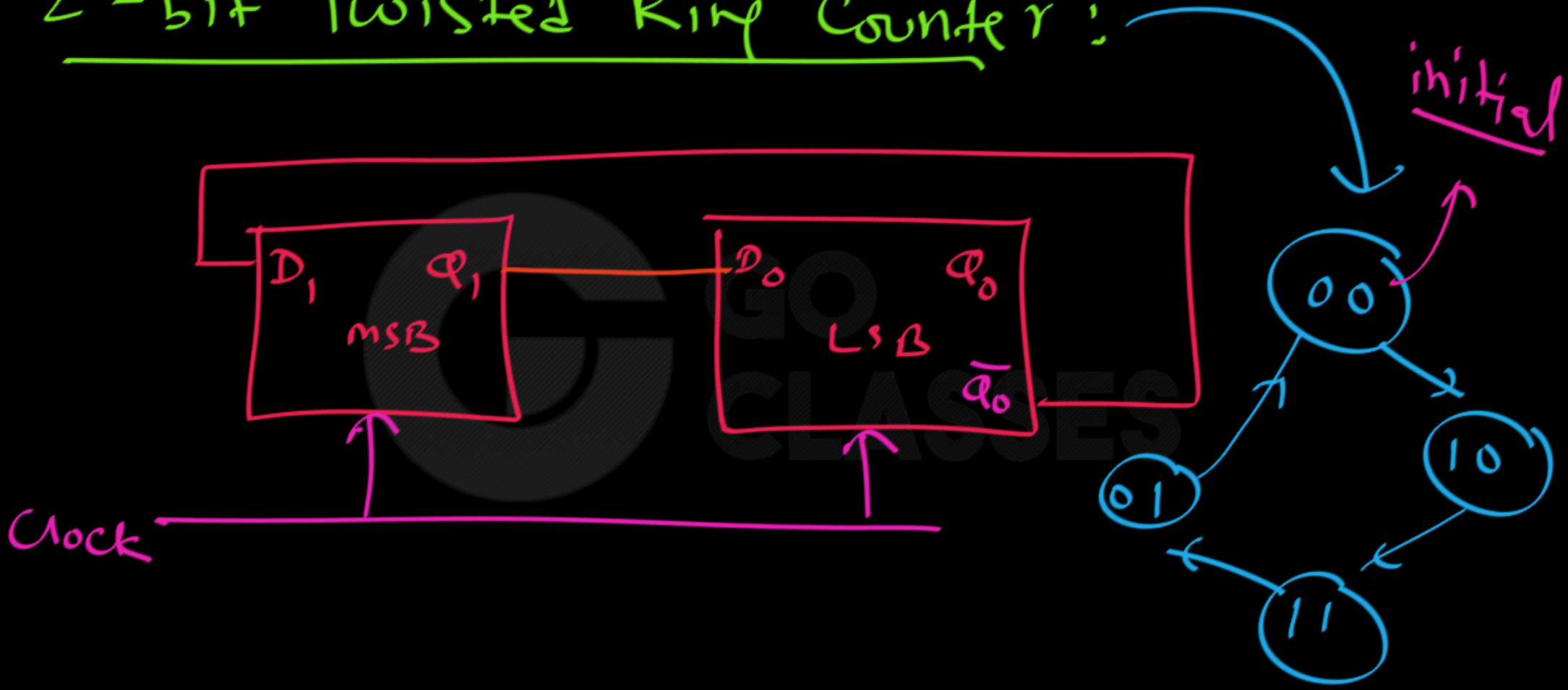
A Johnson counter is a k-bit switch-tail ring counter with $2k$ decoding gates to provide outputs for $2k$ timing signals. The decoding gates are not shown in Fig. 6.18, but are specified in the last column of the table. The eight AND gates listed in the table, when connected to the circuit, will complete the construction of the Johnson counter. Since each gate is enabled during one particular state sequence, the outputs of the gates generate eight timing signals in succession.

The decoding of a k-bit switch-tail ring counter to obtain $2k$ timing signals follows a regular pattern. The all-0's state is decoded by taking the complement of the two extreme flip-flop outputs. The all-1's state is decoded by taking the normal outputs of the two extreme flip-flops. All other states are decoded from an adjacent 1, 0 or 0, 1 pattern in the sequence. For example, sequence 7 has an adjacent 0, 1 pattern in flip-flops B and C. The decoded output is then obtained by taking the complement of B and the normal output of C, or $B'C$.

Source: Morris Mano

Johnson Counter :

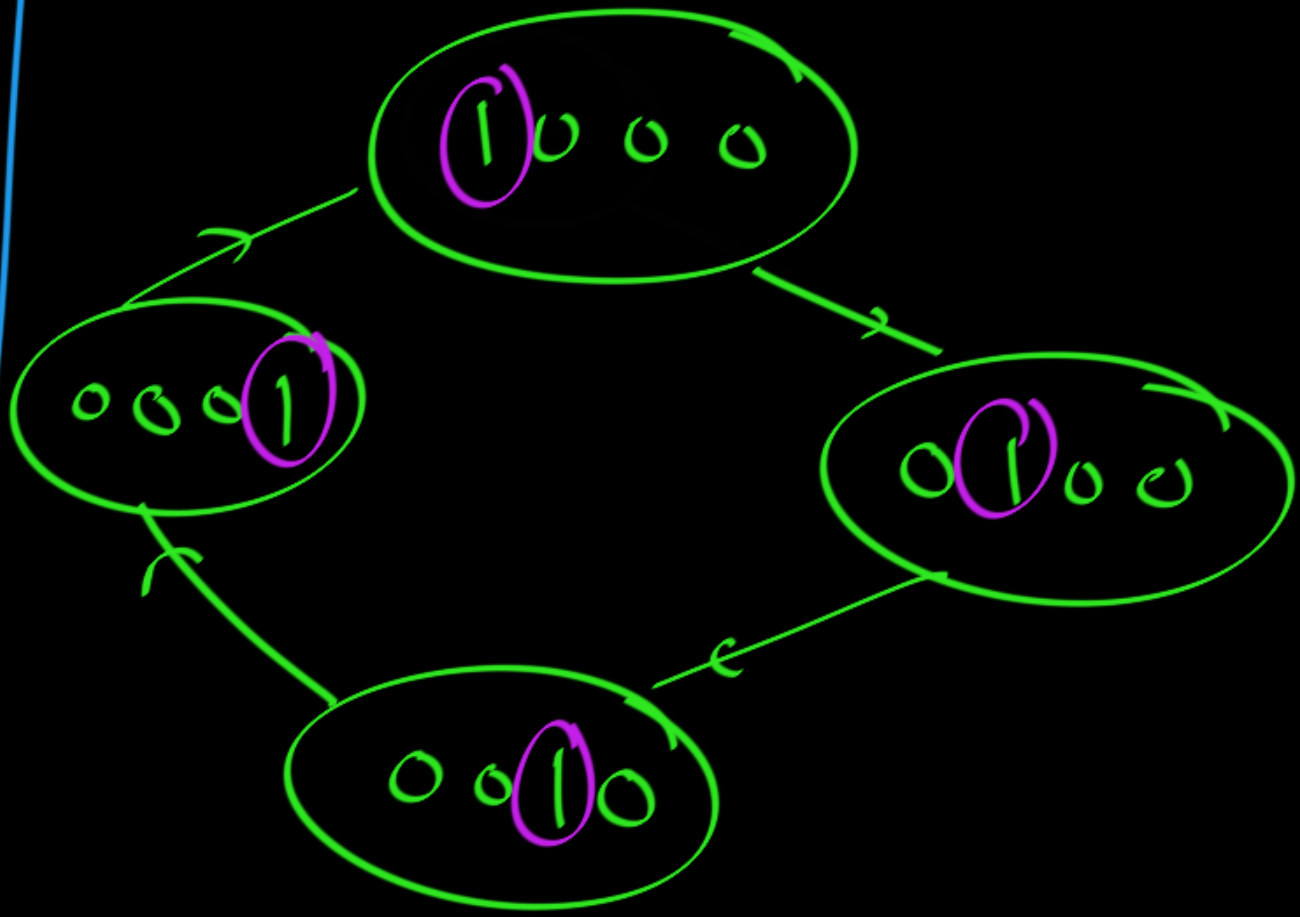
2-bit Twisted Ring Counter :

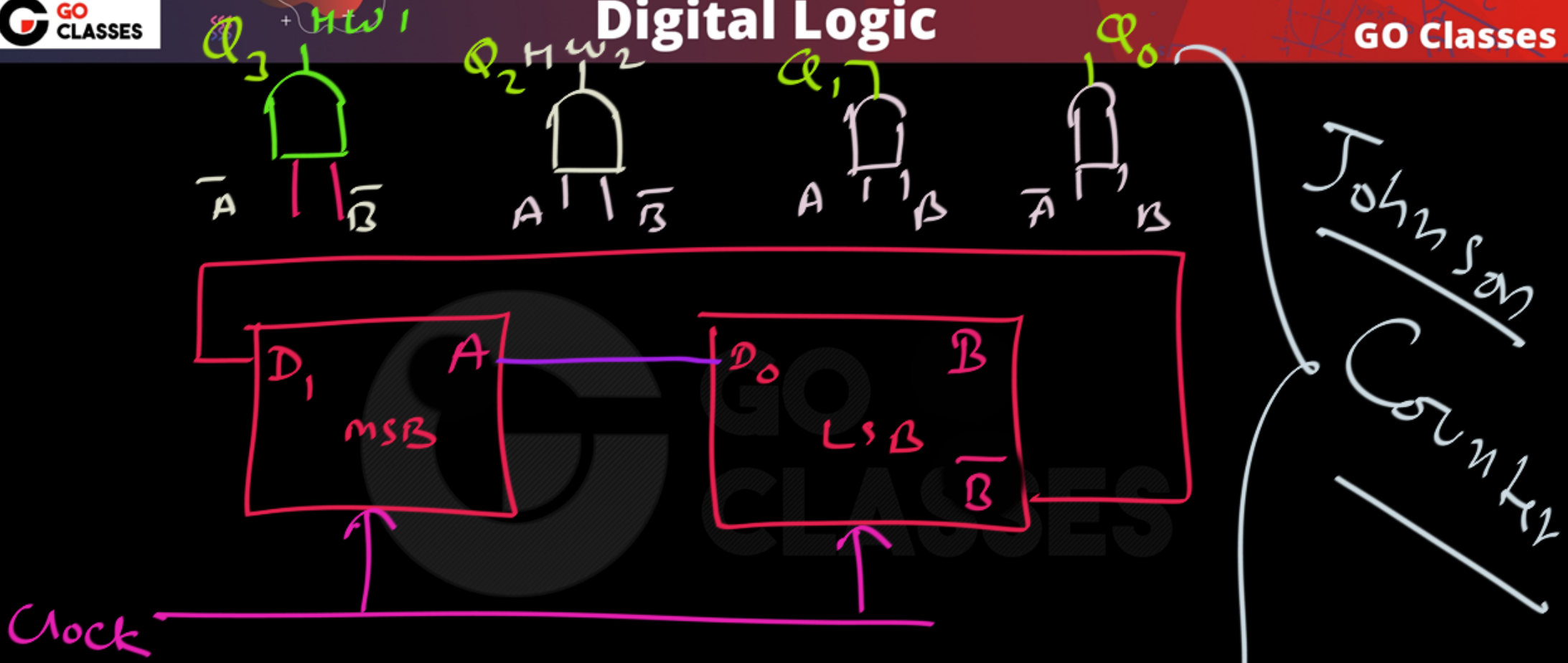


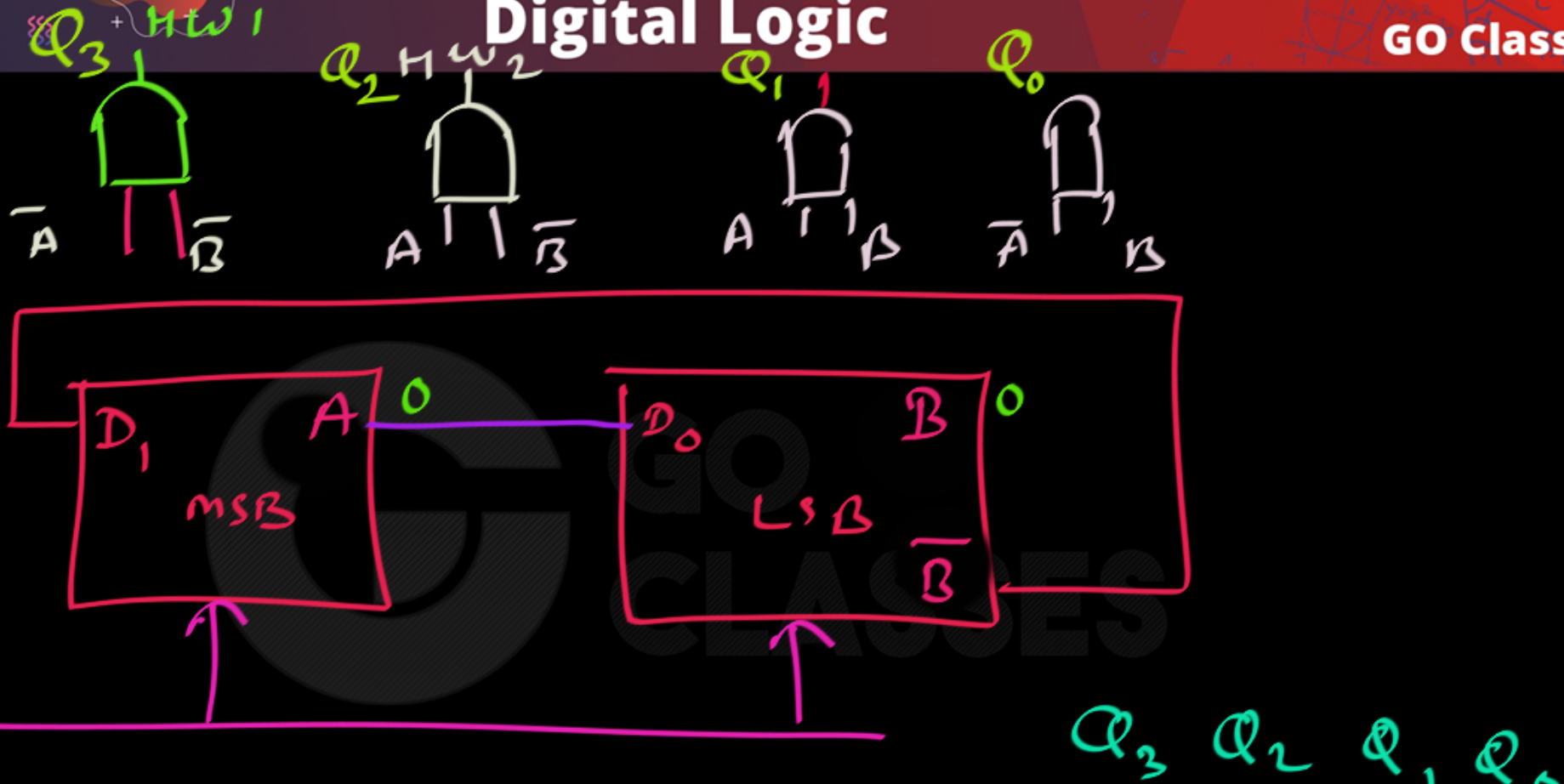
2-bit Twisted Ring Counter

A	B
0	0
1	0
1	1
0	1

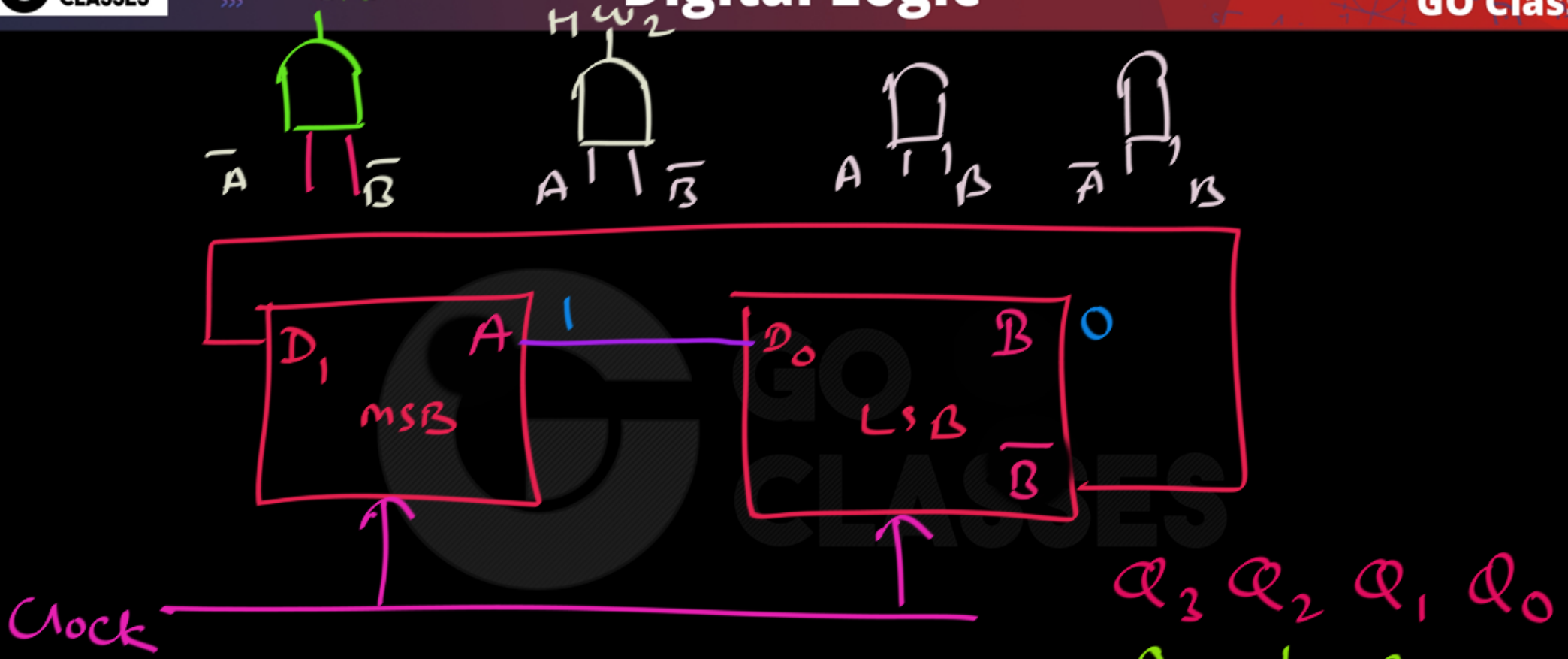
Johnson Counter

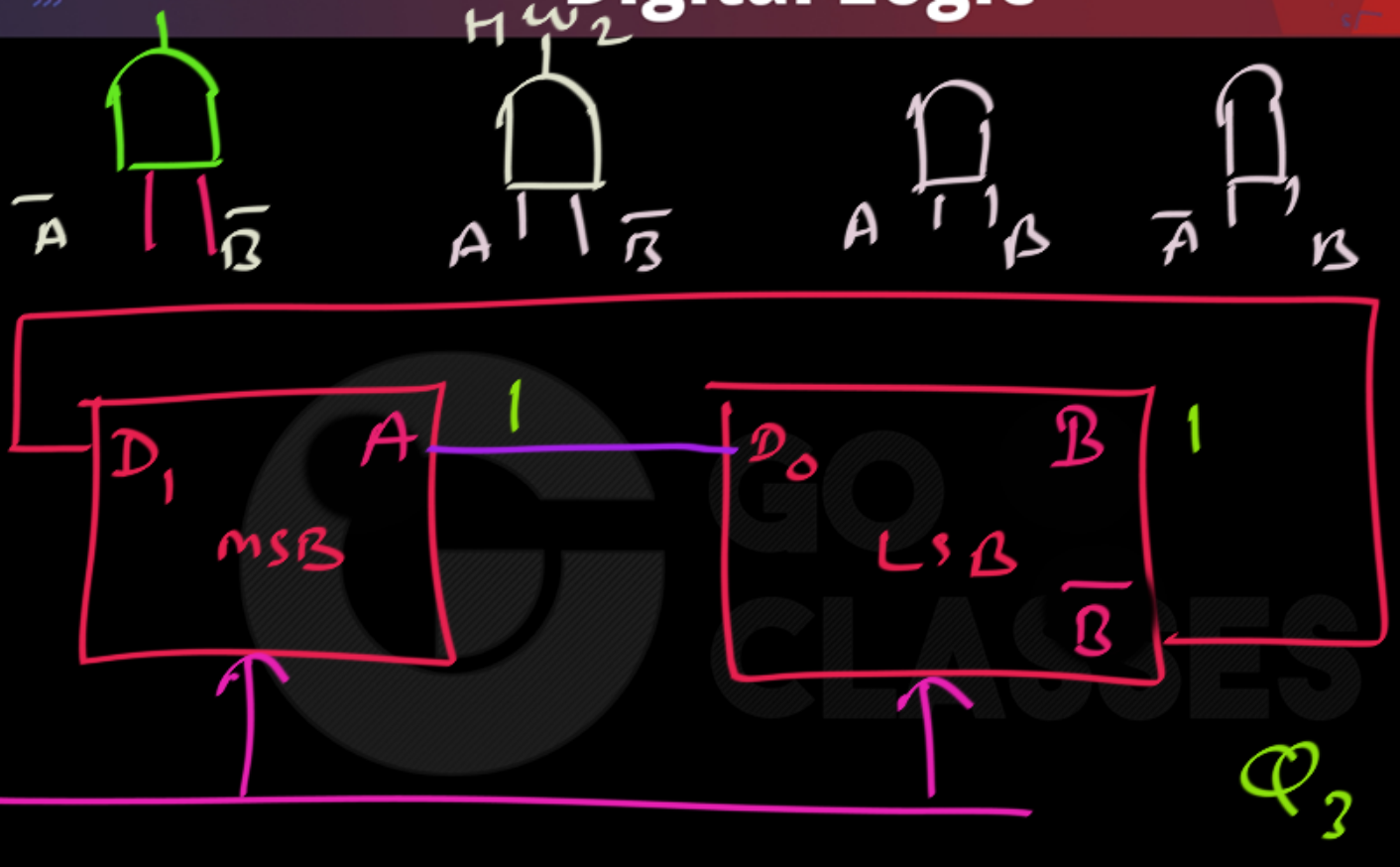



Q3
HW1
Q2
HW2
Q1
Q0
$\bar{A}$
$\bar{B}$
A
$\bar{B}$
A
B
$\bar{A}$
B
D1
A
MSB
D0
B
LSB
$\bar{B}$
Clock
Johnson
Counter



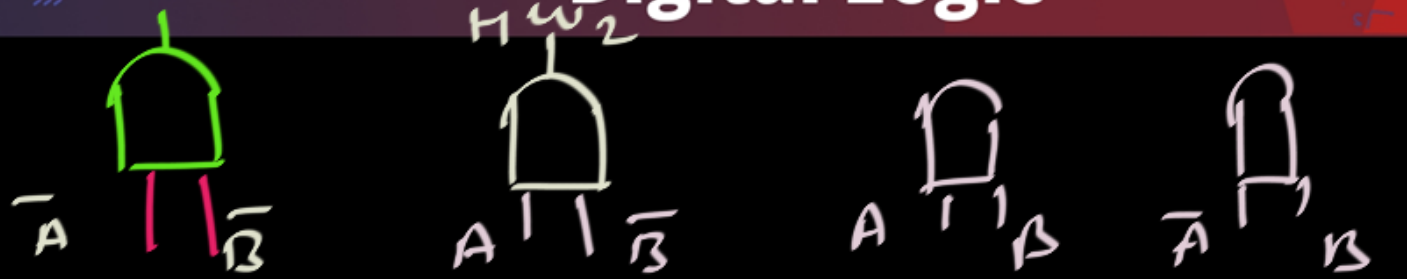
Q_3	Q_2	Q_1	Q_0
1	0	0	0

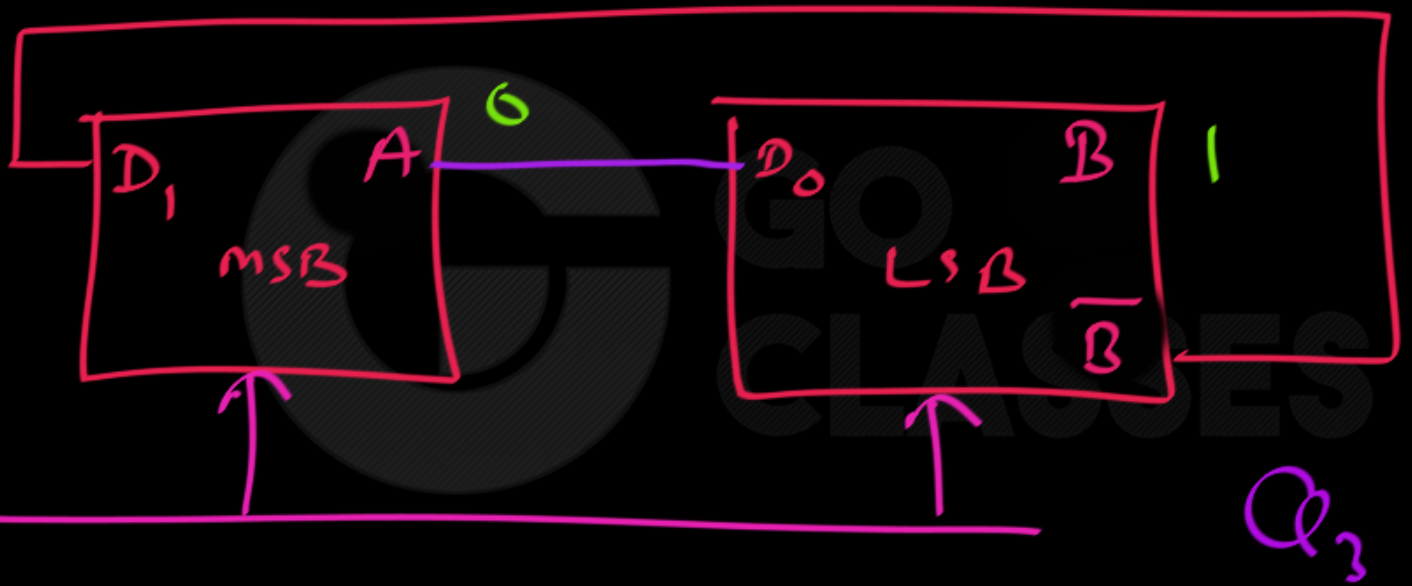

HW 1
HW 2
$\bar{A}$
$\bar{B}$
A
$\bar{B}$
A
B
$\bar{A}$
B
D_1
A
1
MSB
D_0
B
0
LSB
$\bar{B}$
Clock
Q_3 Q_2 Q_1 Q_0
0 1 0 0



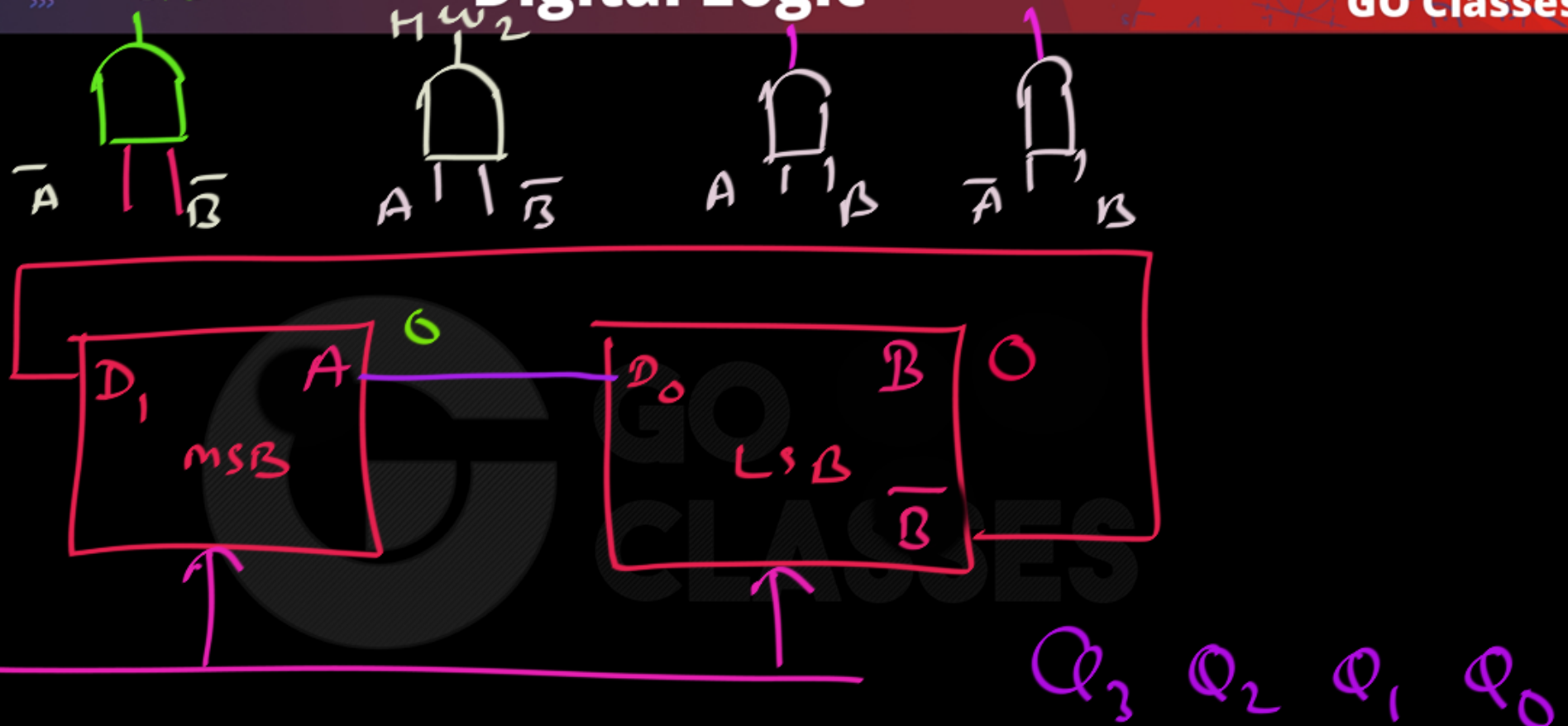
Q_3 Q_2 Q_1 Q_0

0 0 1 0



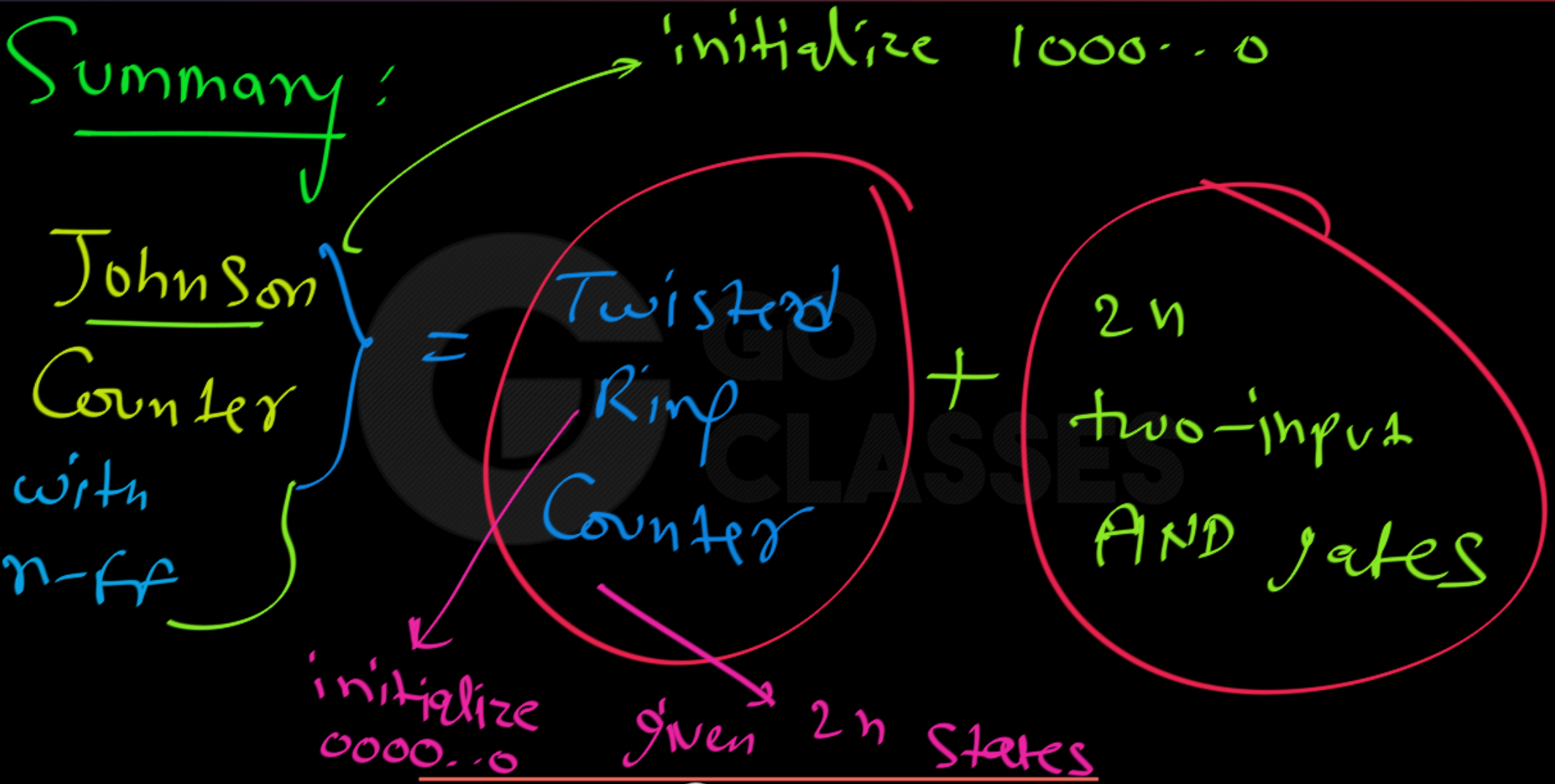


Q_3	Q_2	Q_1	Q_0
0	0	0	1



Q_3 Q_2 Q_1 Q_0

1 0 0 0


Summary :
initialize 1000..0
Johnson Counter with n-ff
=
Twisted Ring Counter
+
2n two-input AND gates
initialize 0000..0
given 2n states

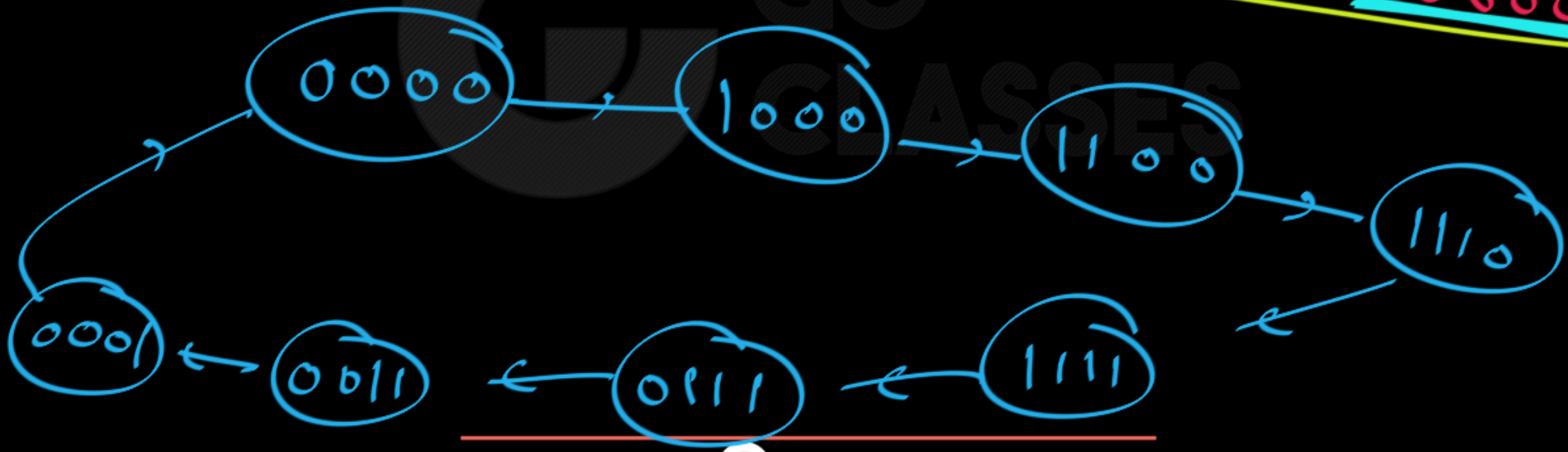
Summary :

(2) In Johnson Counter; only one AND gate o/p is at any point of time.

Summary :

(3) 4-bit Twisted Ring Counter → 8 states

initialize with 0000 :



Summary :

(3) 4-bit Johnson Counter :

8 states

Summary :

4-bit Johnson Counter :

↓

4-bit Twisted Ring Counter

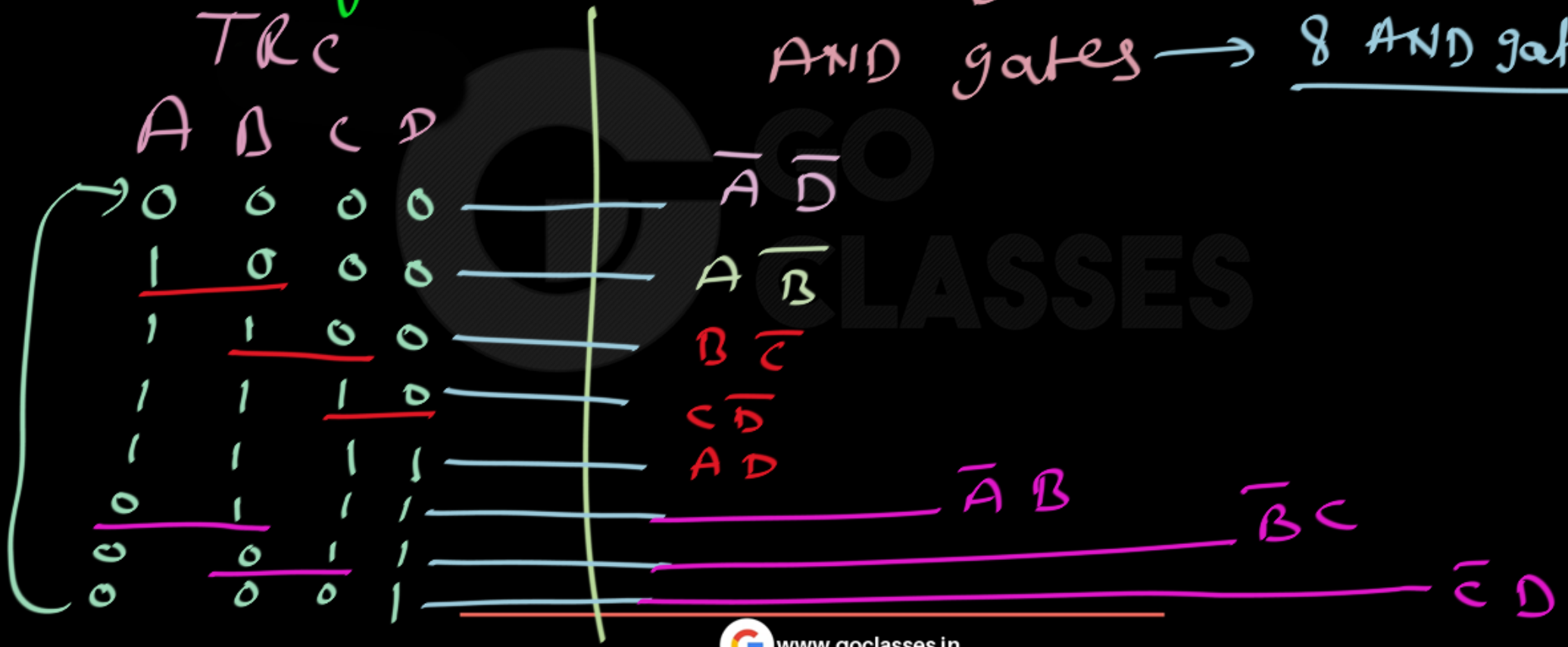
+

8 AND (two-input) gates

Summary :

4-bit Johnson Counter :

AND gates → 8 AND gates



Summary:

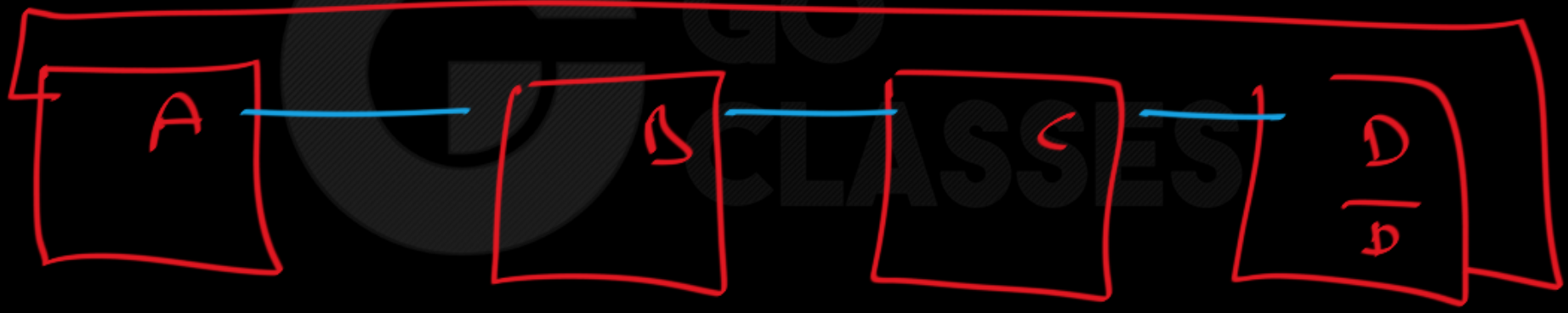
Idea to create Connections for AND gates:

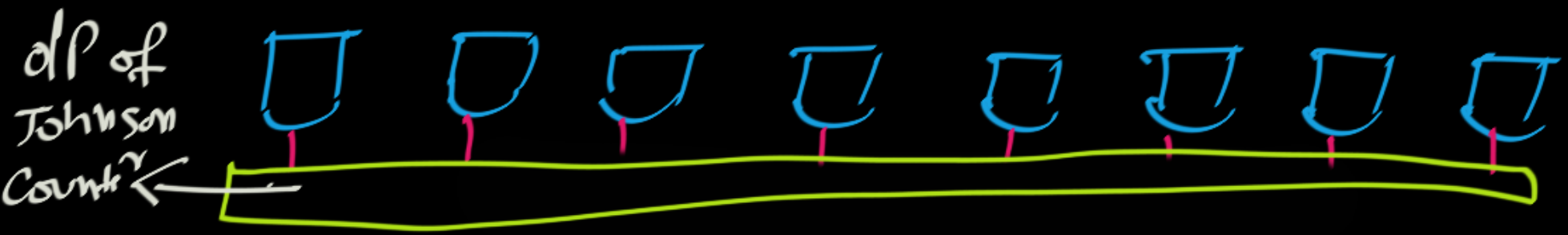
1st Occurrence of

1 0 OR 0 1 OR Extreme inputs

Summary:

4 bit Johnson Counter:





Note :

Some Authors :

Twisted Ring Counter = Johnson Counter

Note :

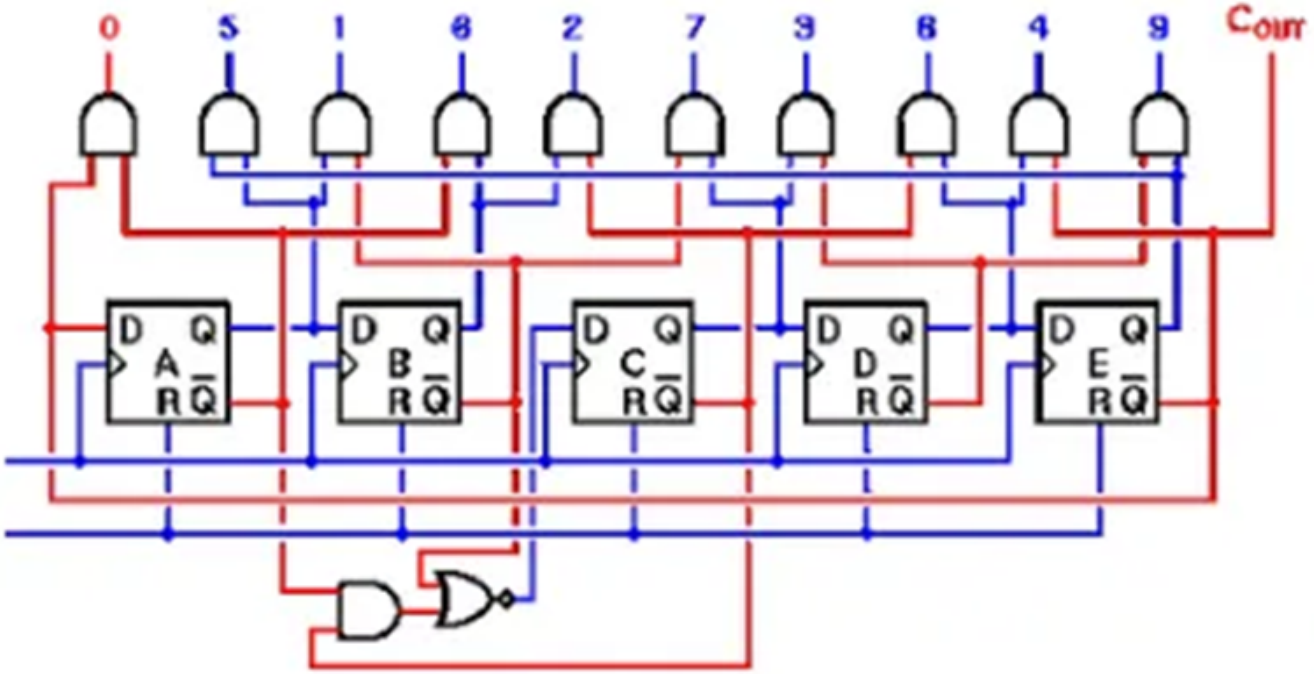
Morris Mano :

Twisted Ring Counter $\neq$ Johnson Counter

3 Johnson Counter

Another form of ring counter is created by feeding back the complement of the contents of the last flip-flop to the input of the first flip-flop. This is called a *twisted ring counter*, but is better known as the *Johnson counter*. The alternative term Möbius counter is found in many books and articles because the Johnson counter resembles the famous Möbius strip [4]. For example, in a 5-flip-flop Johnson counter with an initial register contents (or state) of 00000, the repeating sequence is 00000, 10000, 11000, 11100, 11110, 11111, 01111, 00111, 00011, 00001. When observing the pattern, it can be seen that any changes between succeeding states, only one flip-flop changes state. As a result, any of these states is directly, spike-free decodable with only a two-input gate [5, 6].

What is Johnson Counter?



Final Conclusion:

To generate 2^n timing signals, we need either a shift register with 2^n flip-flops or an n-bit binary counter together with an n-to-2^n-line decoder. For example, 16 timing signals can be generated with a 16-bit shift register connected as a ring counter or with a 4-bit binary counter and a 4-to-16-line decoder. In the first case, we need 16 flip-flops. In the second, we need 4 flip-flops and 16 four-input AND gates for the decoder. It is also possible to generate the timing signals with a combination of a shift register and a decoder. That way, the number of flip-flops is less than that in a ring counter, and the decoder requires only two-input gates. This combination is called a *Johnson counter*.