



This Lecture :

Sequential Circuits: Counters

Synchronous

Counters(part-1)



Recap:

Synchronous Counters

Vs

Ripple Counters



WHAT IS MODULUS N COUNTER ?

- COUNTERS ARE SEQUENTIAL LOGIC DEVICES THAT FOLLOW A PREDETERMINED SEQUENCE OF COUNTING STATES WHICH ARE TRIGGERED BY AN EXTERNAL CLOCK (CLK) SIGNAL
- THE NUMBER OF STATES OR COUNTING SEQUENCES THROUGH WHICH A PARTICULAR COUNTER ADVANCES BEFORE RETURNING ONCE AGAIN BACK TO ITS ORIGINAL FIRST STATE IS CALLED THE **MODULUS (MOD)**. IN OTHER WORDS, THE MODULUS (OR JUST MODULO) IS THE NUMBER OF STATES THE COUNTER COUNTS AND IS THE DIVIDING NUMBER OF THE COUNTER.

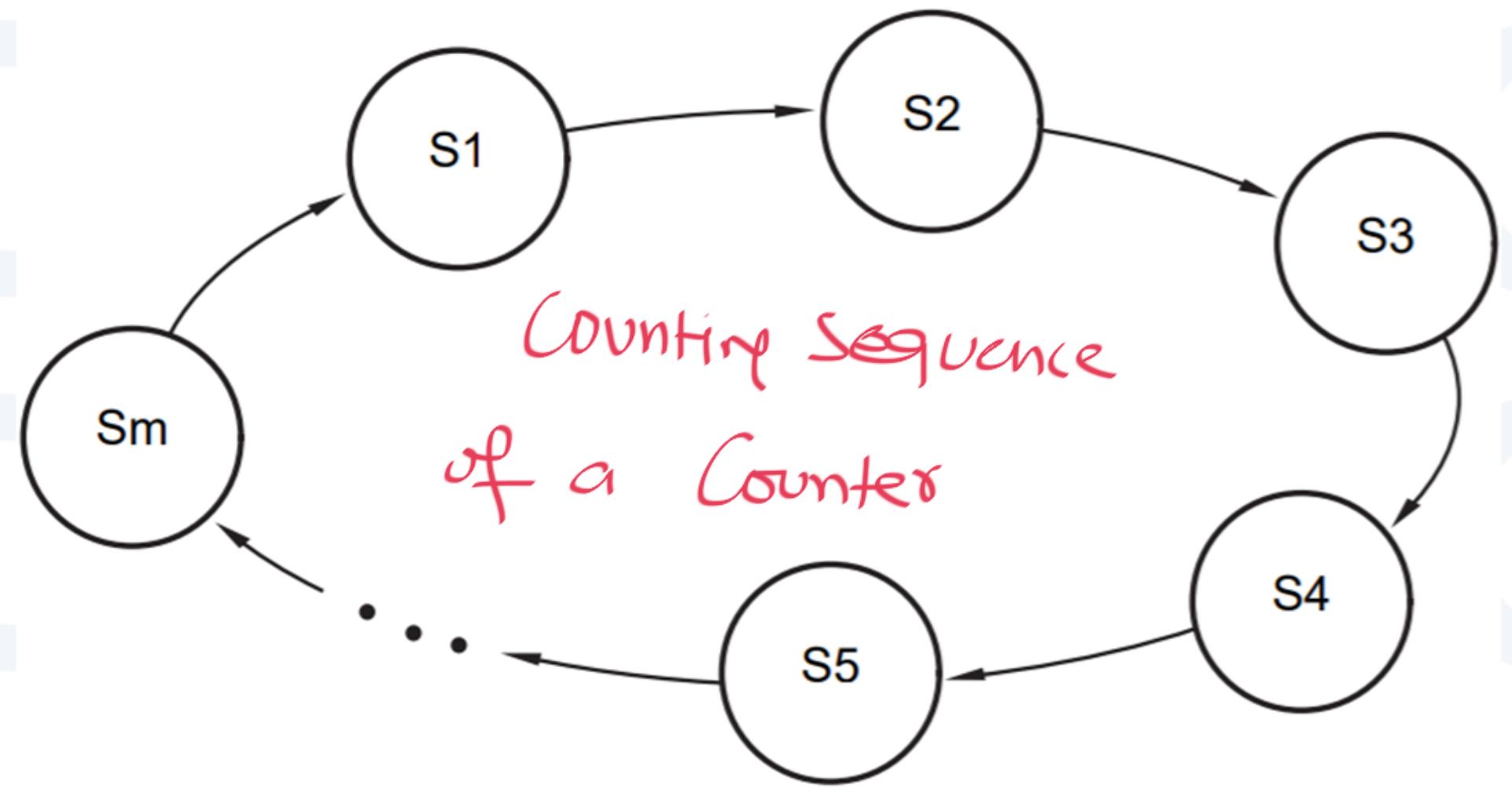
DETERMINING THE MODULUS :

- *MOD COUNTERS* ARE DEFINED BASED ON THE NUMBER OF STATES THAT THE COUNTER WILL SEQUENCE THROUGH BEFORE RETURNING BACK TO ITS ORIGINAL VALUE
- FOR EXAMPLE, A 2-BIT COUNTER THAT COUNTS FROM 00_2 TO 11_2 IN BINARY, HAS A MODULUS VALUE OF 4 ($00 \rightarrow 01 \rightarrow 10 \rightarrow 11$, RETURN BACK TO 00) SO WOULD THEREFORE BE CALLED A MODULO-4,. NOTE ALSO THAT IT HAS TAKEN 4 CLOCK PULSES TO GET FROM 00 TO 11 .
- THE MAXIMUM NUMBER OF POSSIBLE OUTPUT STATES (MAXIMUM MODULUS) FOR THE COUNTER IS: 2^N



A counter with m states in the counting sequence is called a modulo- m counter or, sometimes, a divide-by- m counter.

A counter with a nonpower-of-2 modulus has extra states that are not used in normal operation.



Types

Synchronous

all the data bits
change
synchronously with
the application of a
clock signal

Asynchronous

independent of
the input clock so
the data bits
change state at
different times
one after the
other



ASynchronous Counter

Way of Studying :

Pure Analytical

Analysis



Synchronous Counter \Rightarrow best way to study :

① Analysis : Given Circuit \rightarrow Analyse

② Design : Counting Sequence \rightarrow Design the Circuit (Implement)

③ Some special Counting Sequences

- 111
Counters
- { ① Ring Counter → fix Counting Sequence
② Johnson Counter → 11 Counting Sequence



Next Topic:

Analysis of Synchronous Sequential Circuits (When NO External Inputs)

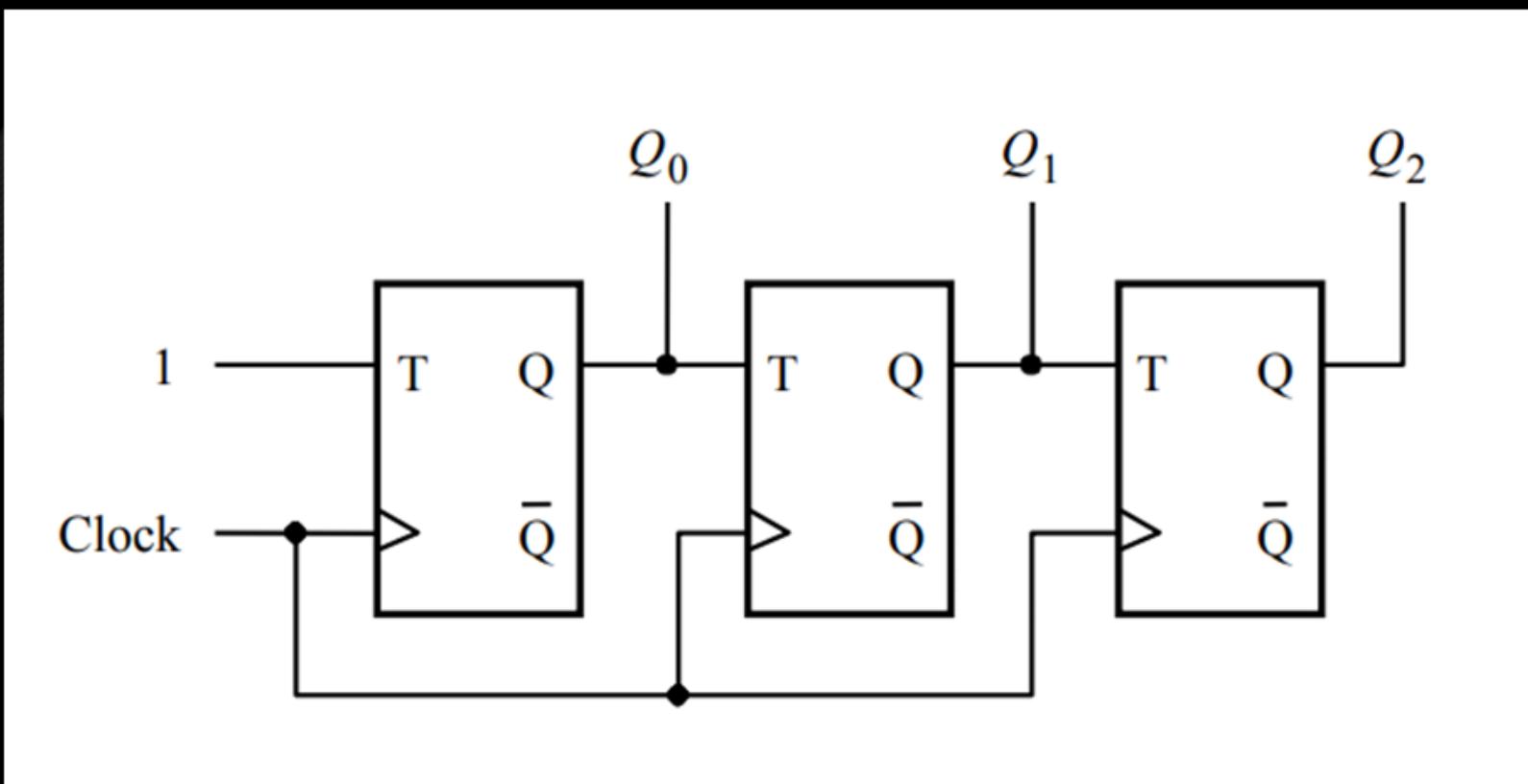


Analysis of Any Synchronous Sequential Circuit
is Very Easy (When NO External Inputs):

1. Find out the Next State from the present state.
2. Find out the state diagram, state transition table etc.



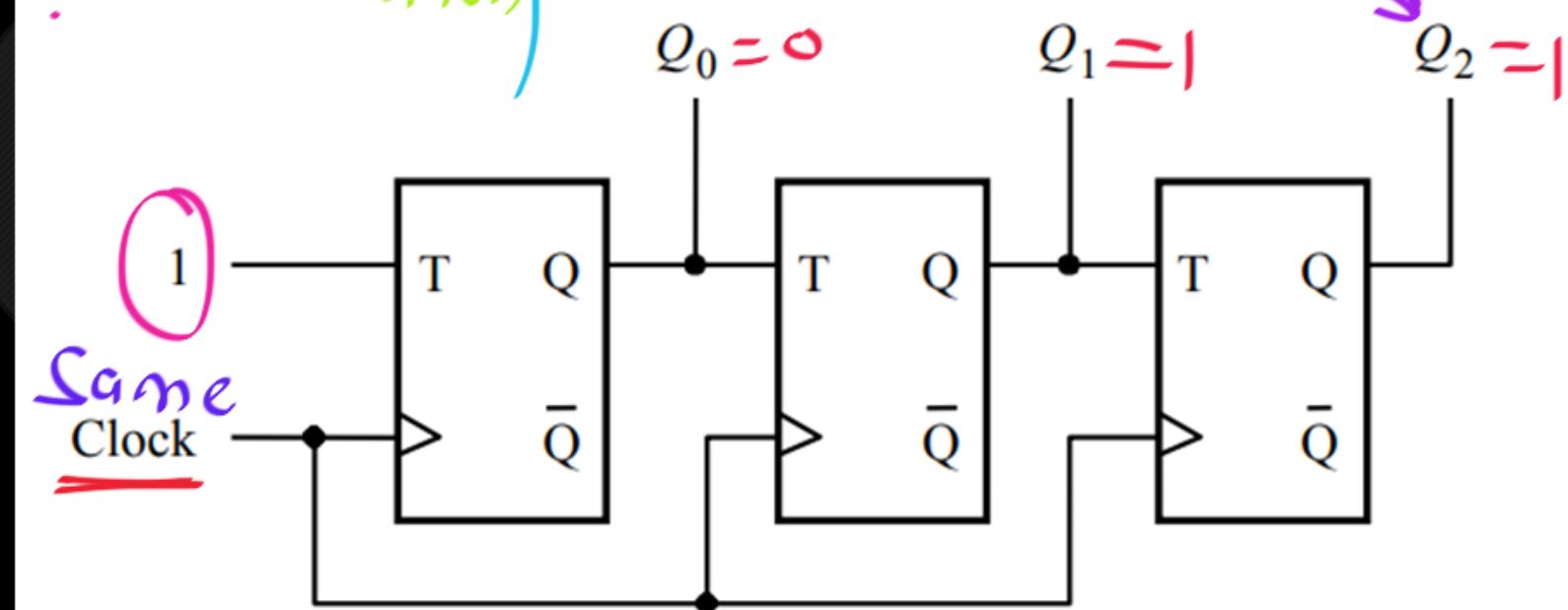
Q: Analyse the given synchronous sequential circuit. Find out the next state if initial state is $Q_2Q_1Q_0 = 000$



State of
this circuit

$Q_2 Q_1 Q_0$
1 1 0

✓ synchronous sequential
ff output state
Combination

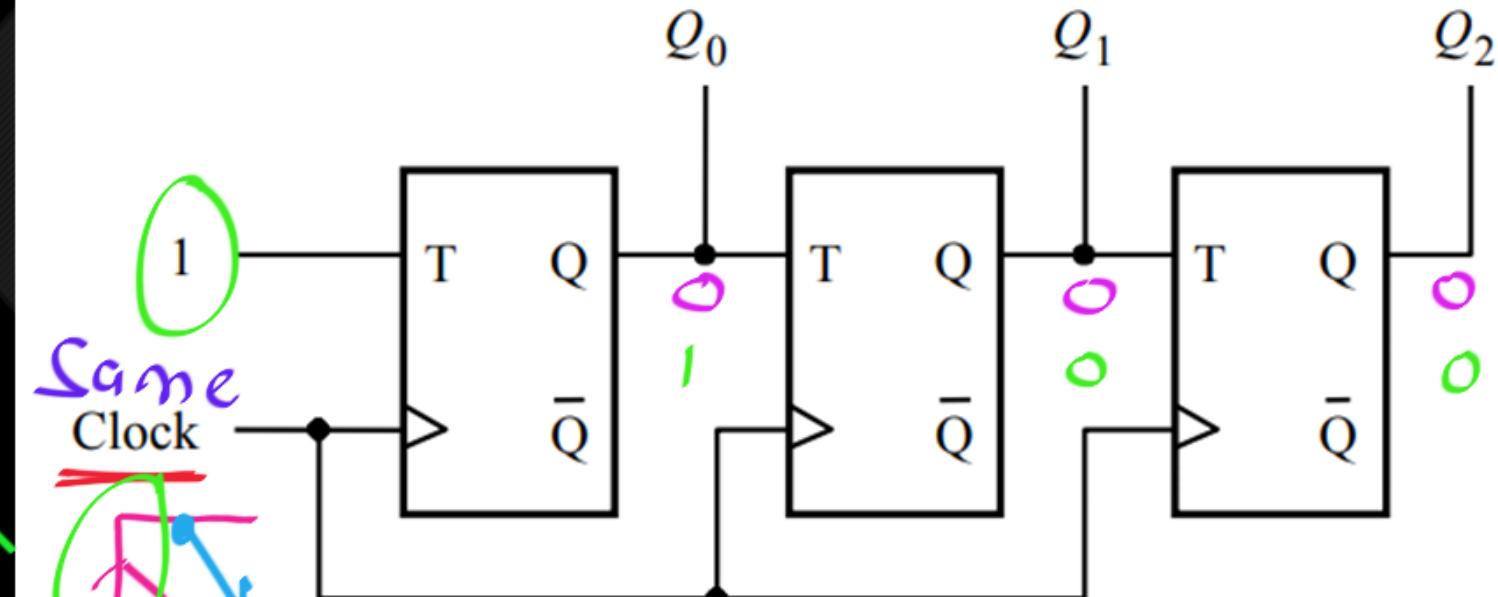


Q: Analyse the given synchronous sequential circuit. Find out the next state if initial state is

$$Q_2 Q_1 Q_0 = 000$$

↓
MSB ↓LSB

$$Q_2 Q_1 Q_0 \\ 0 \ 0 \ 0$$



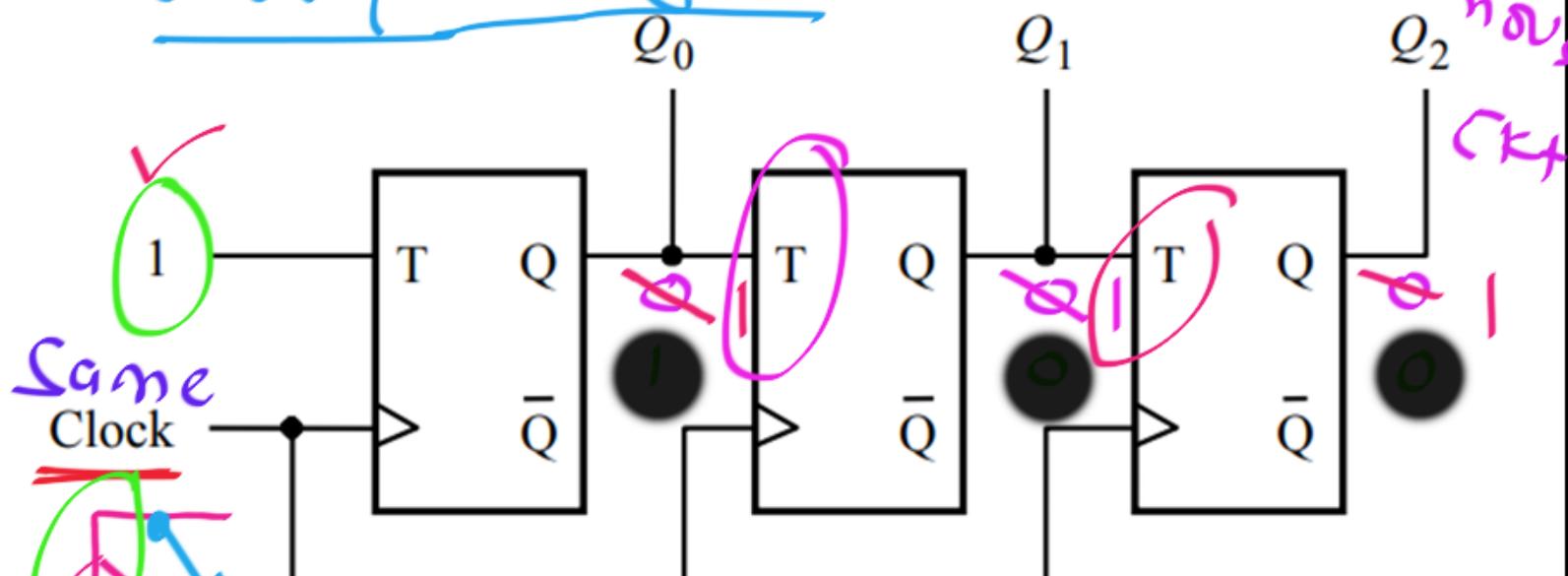
Q: Analyse the given synchronous sequential circuit. Find out the next state if initial state is

$$Q_2 Q_1 Q_0 = \underline{000}$$

MSB LSB

$$\underline{Q_2 Q_1 Q_0} \\ 000$$

Wrong Analysis \Rightarrow Reason: synchronous

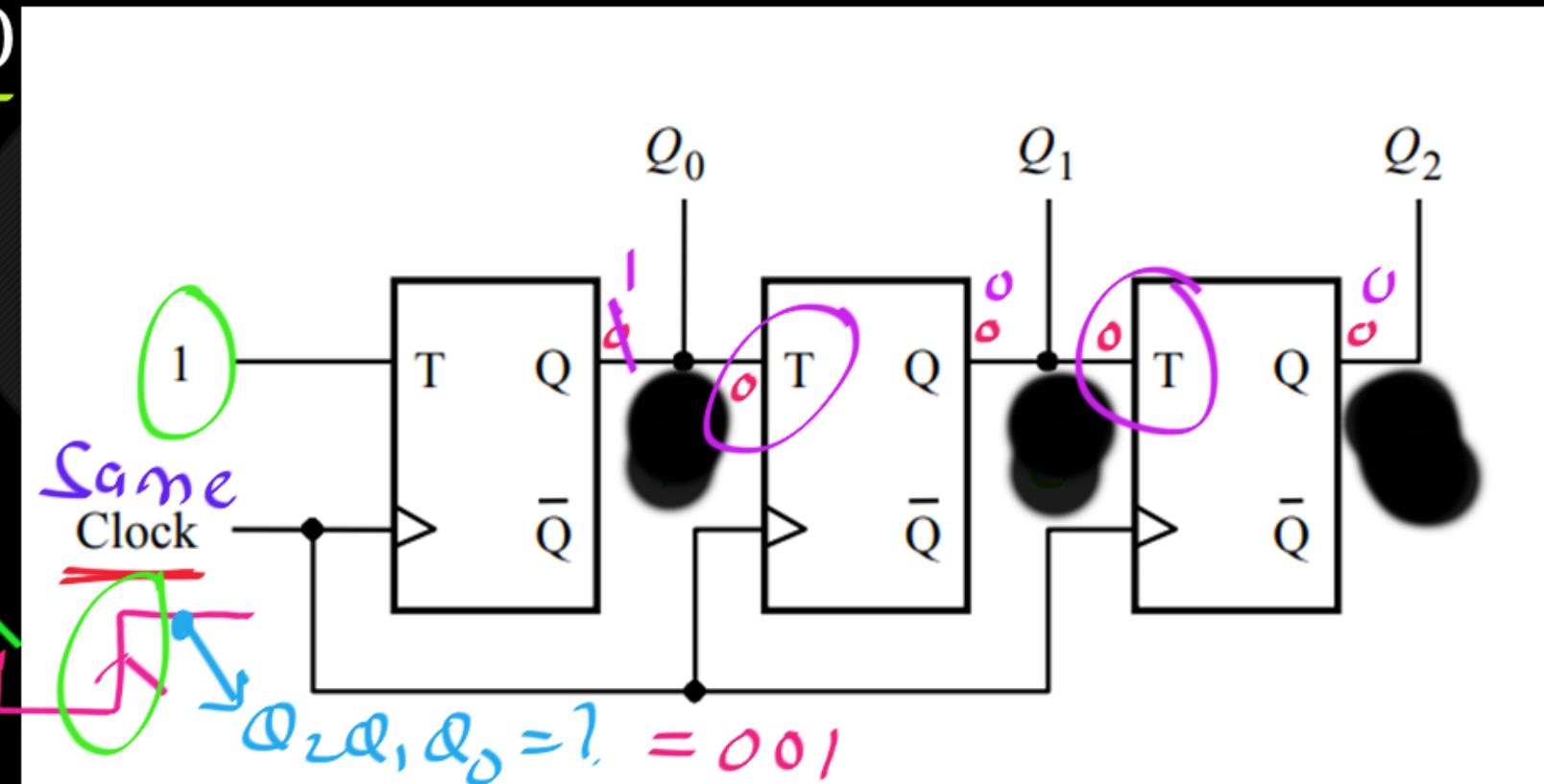


Q: Analyse the given synchronous sequential circuit. Find out the next state if initial state is

$$Q_2 Q_1 Q_0 = 000$$

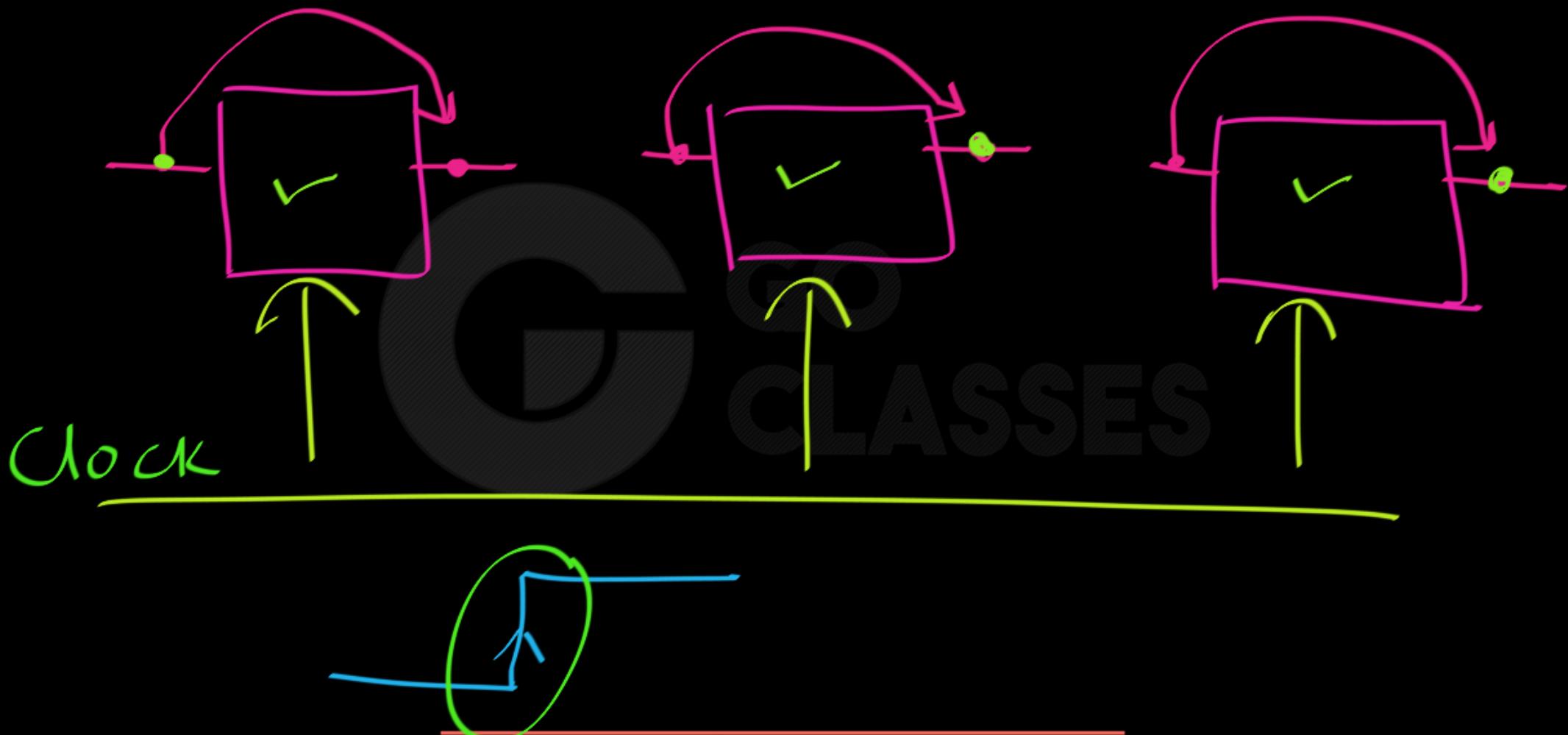
↓
MSB ↓LSB

$$Q_2 Q_1 Q_0 \\ 0 \ 0 \ 0$$



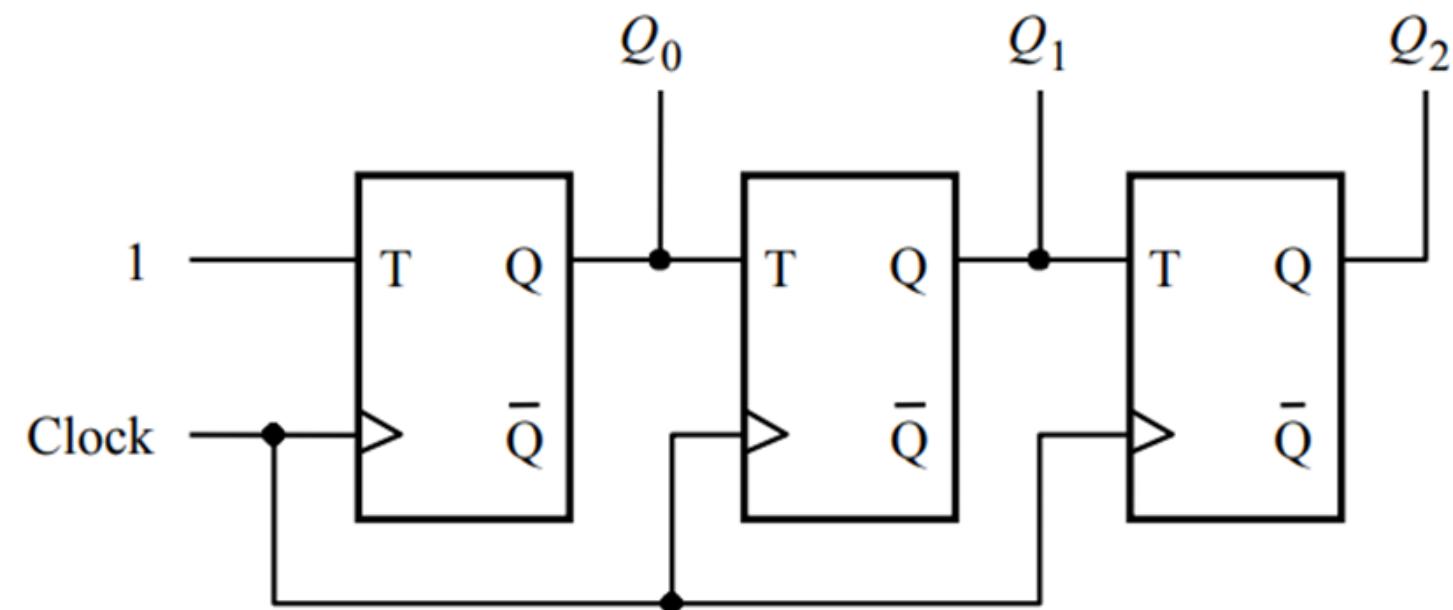


Digital Logic



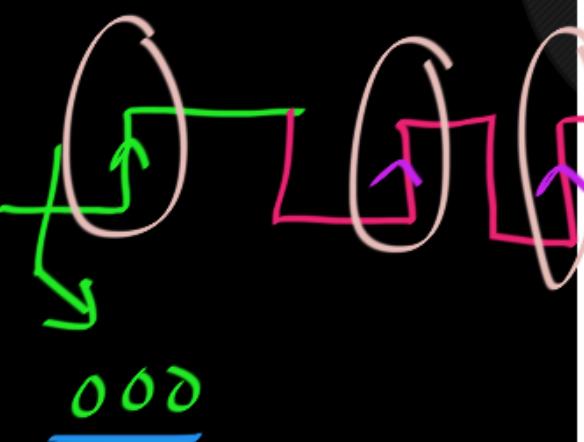
Q: Analyse the given synchronous sequential circuit. Find out the after 4 clock cycles if initial state is

$Q_2Q_1Q_0 = 000$

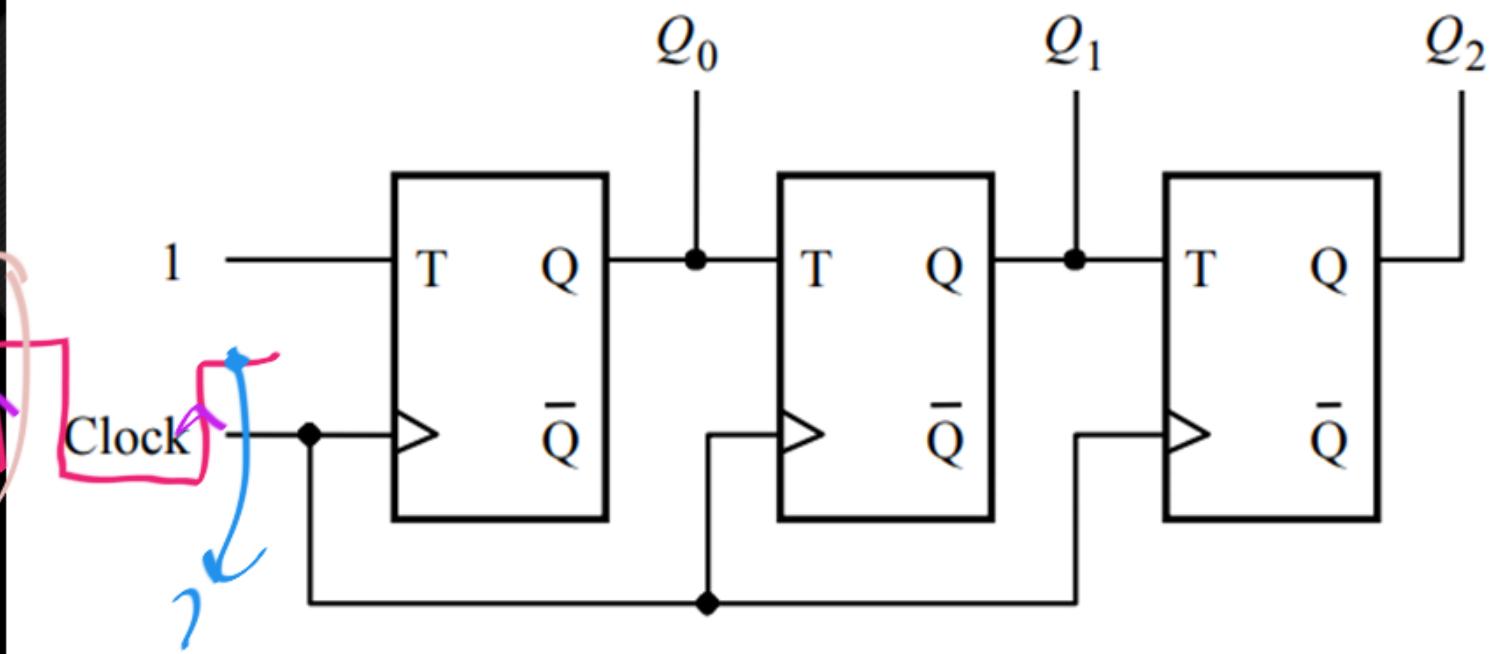


Q: Analyse the given synchronous sequential circuit. Find out the after 4 clock cycles if initial state is

$$\underline{Q_2 Q_1 Q_0 = 000}$$



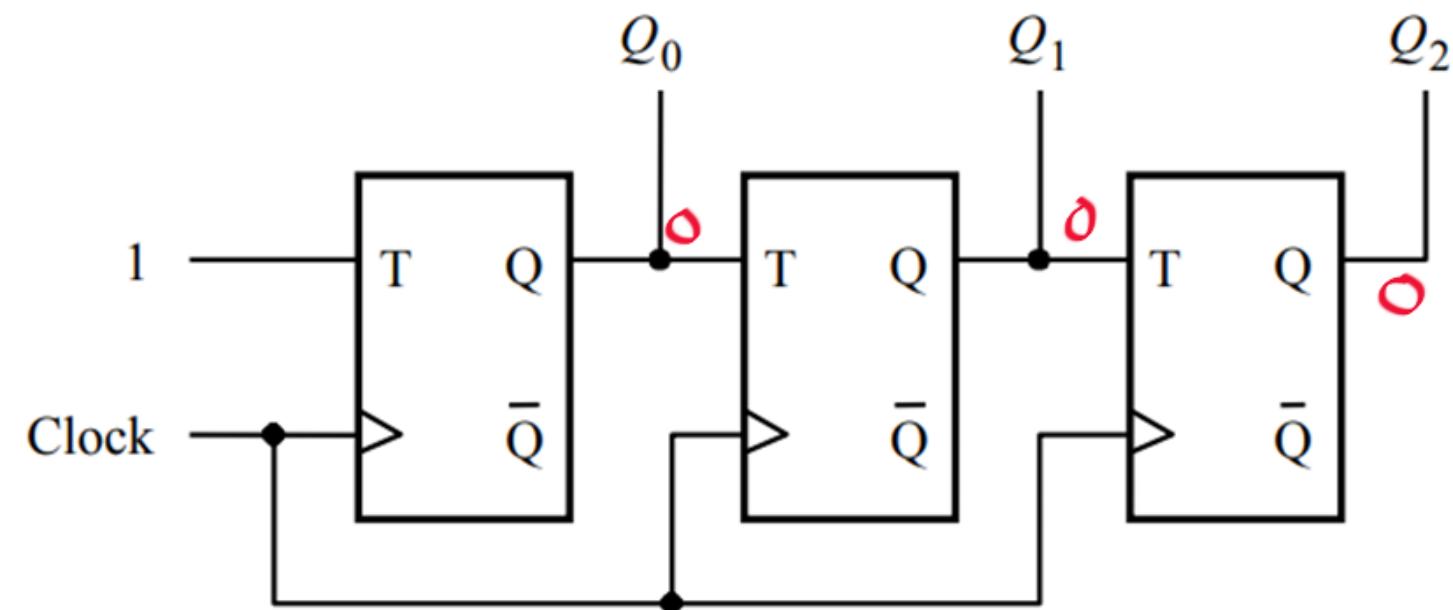
After 4 Rising Edges of Clock



Q: Analyse the given synchronous sequential circuit. Find out the after 4 clock cycles if initial state is

$$\underline{Q_2 Q_1 Q_0 = 000}$$

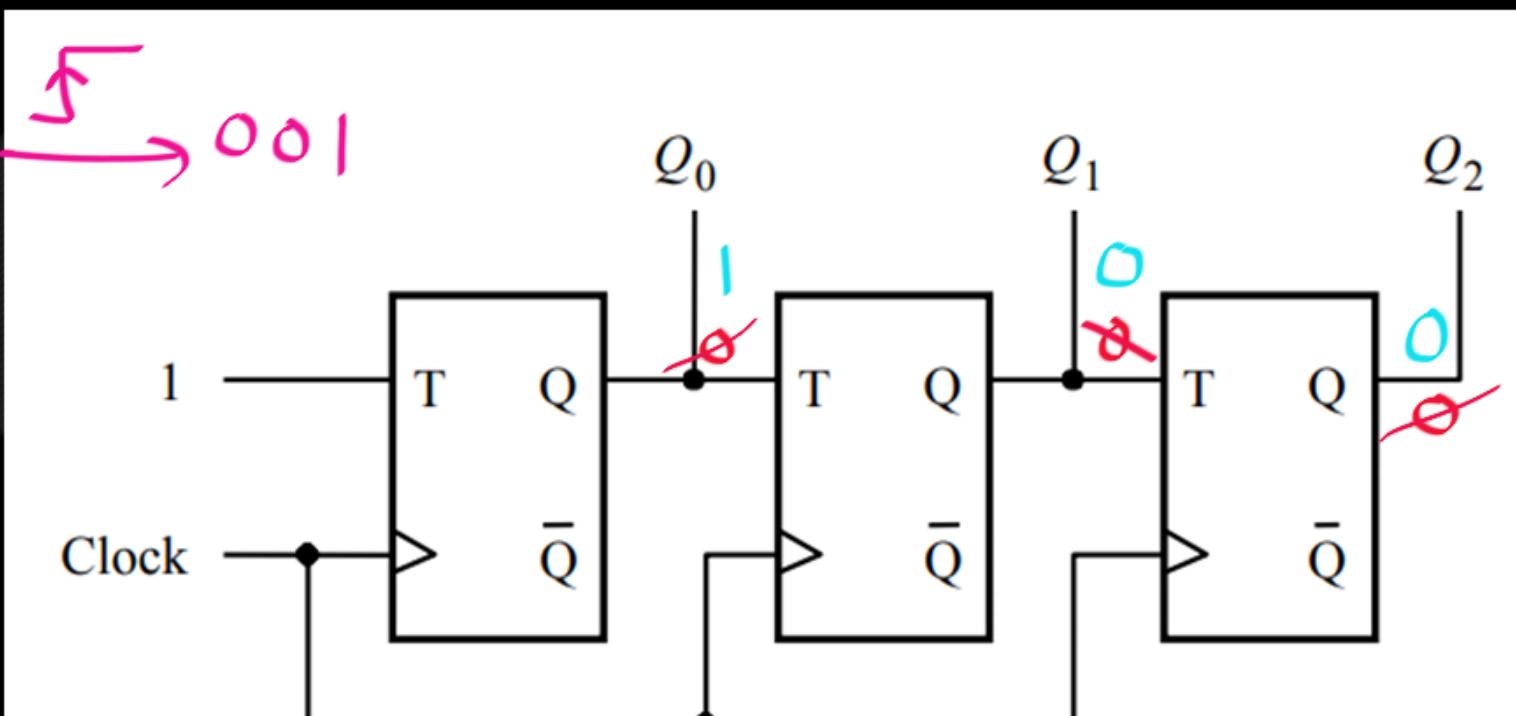
Initial :



Q: Analyse the given synchronous sequential circuit. Find out the after 4 clock cycles if initial state is

$$\underline{Q_2 Q_1 Q_0 = 000}$$

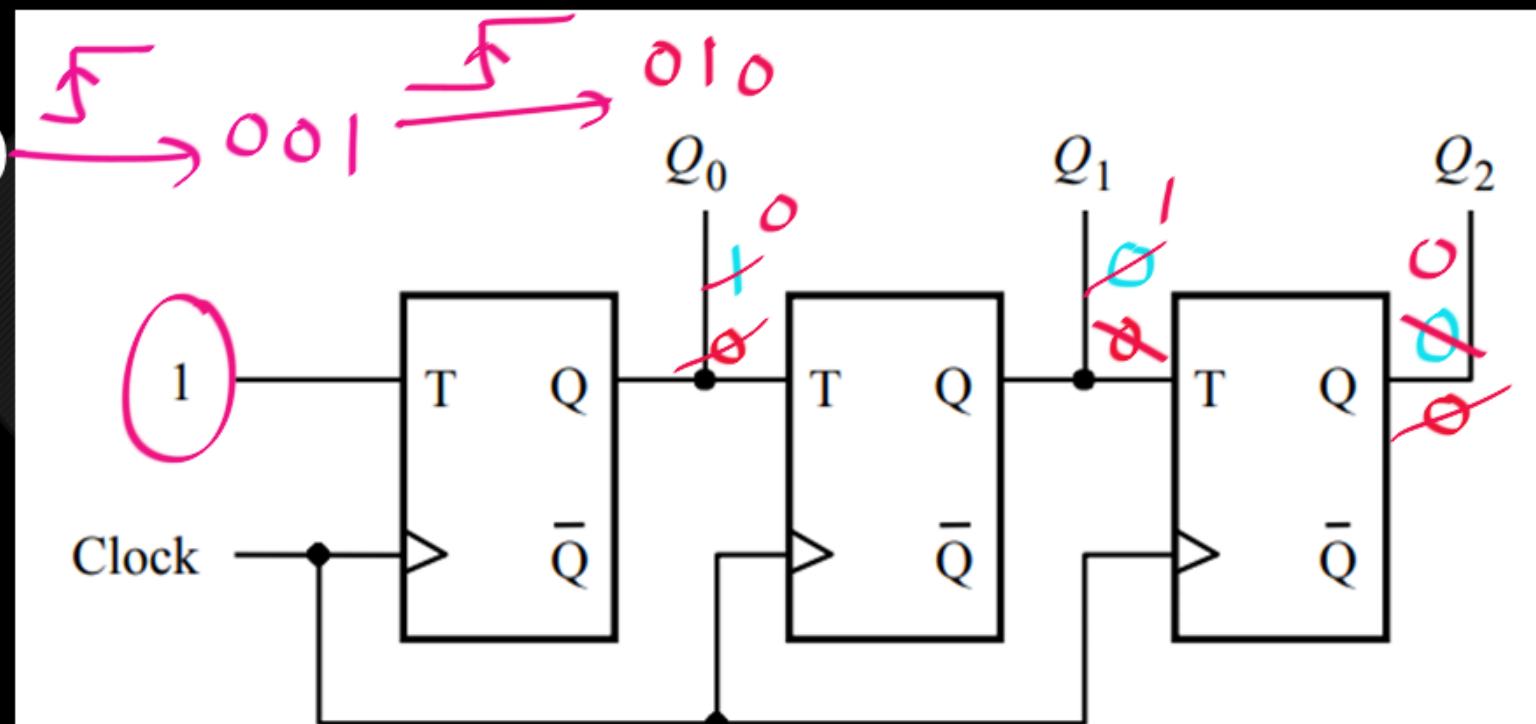
After
one
Positive
Edge of Clock



Q: Analyse the given synchronous sequential circuit. Find out the after 4 clock cycles if initial state is

$$\underline{Q_2 Q_1 Q_0 = 000}$$

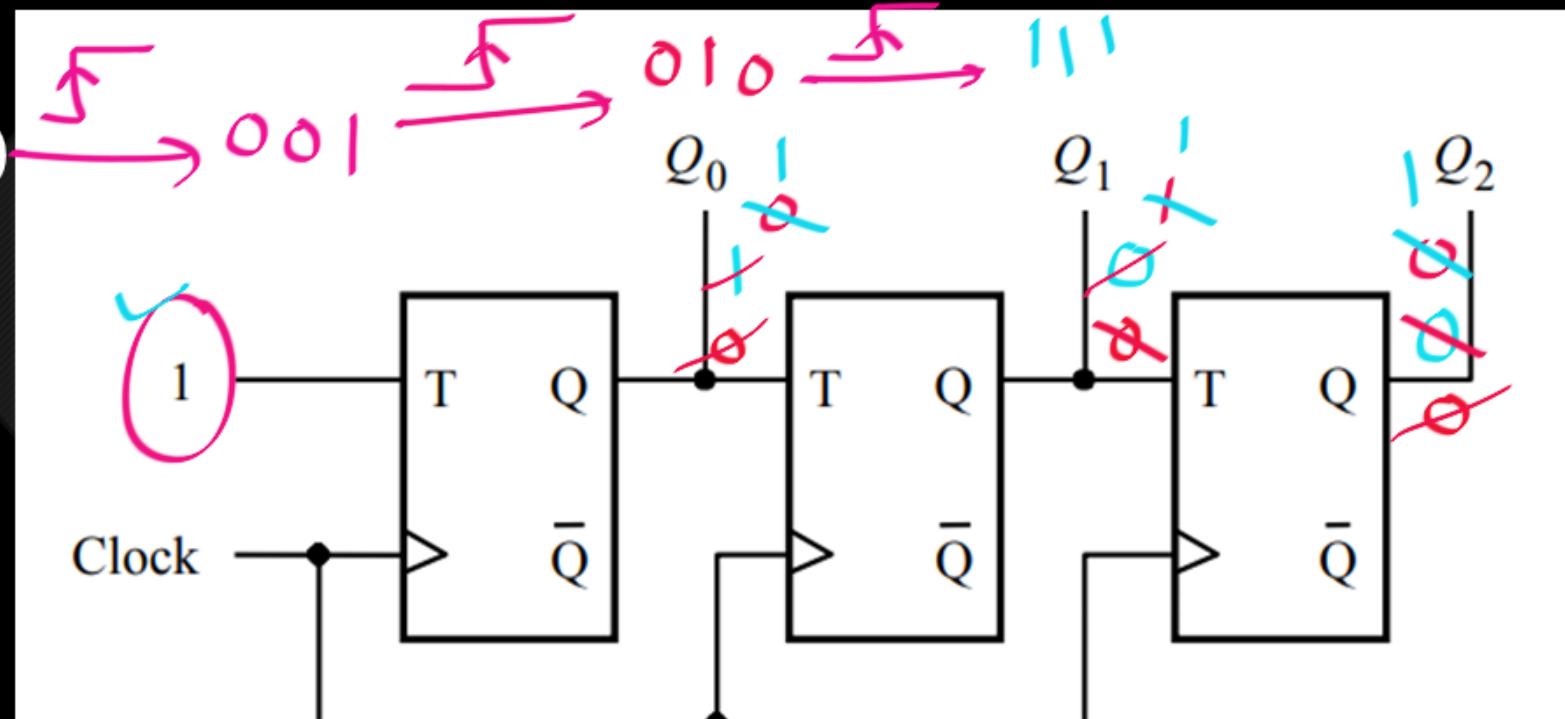
After :
one
Positive
Edge of clock



Q: Analyse the given synchronous sequential circuit. Find out the after 4 clock cycles if initial state is

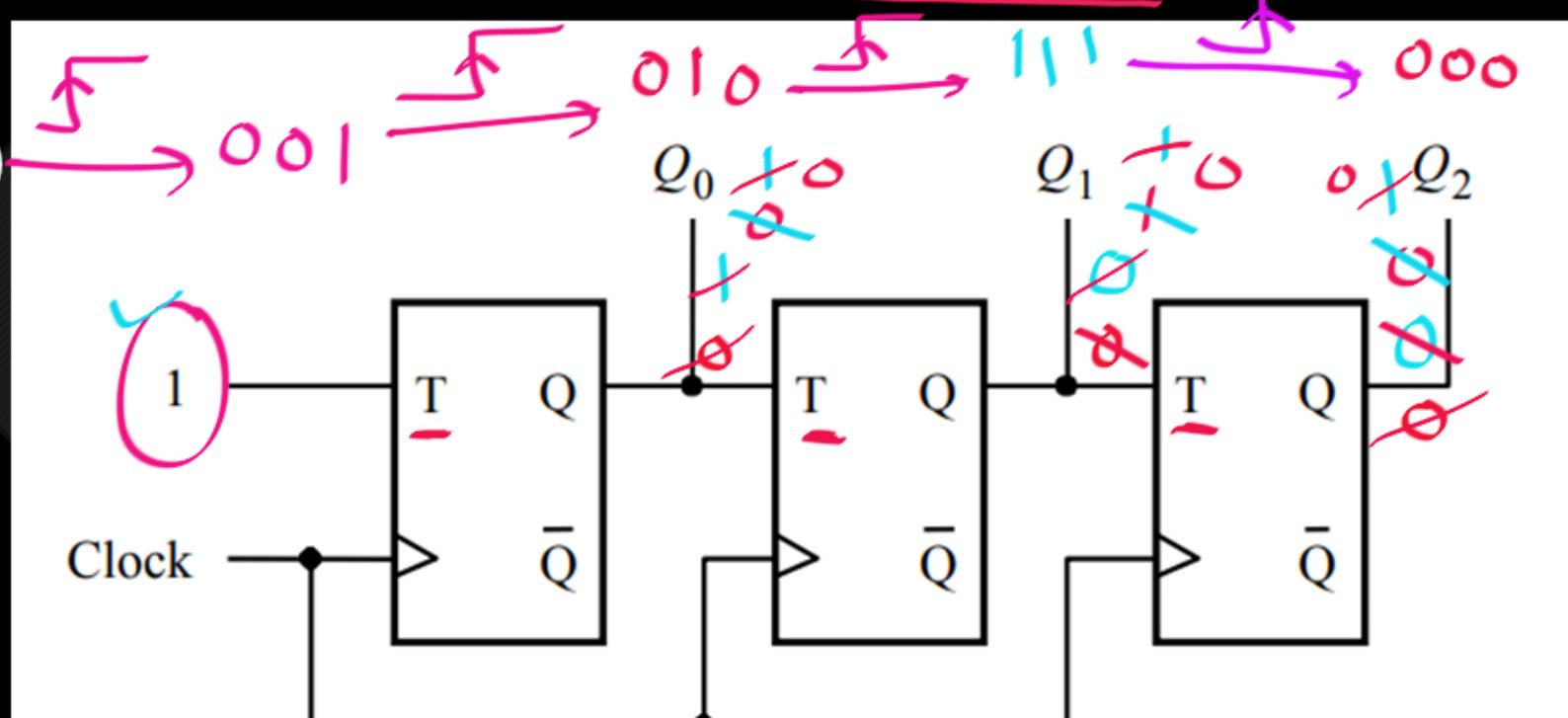
$$\underline{Q_2 Q_1 Q_0 = 000}$$

After :
one
Positive
Edge of clock



Q: Analyse the given synchronous sequential circuit. Find out the after 4 clock cycles if initial state is $Q_2Q_1Q_0 = 000$

After
one
Positive
Edge of clock

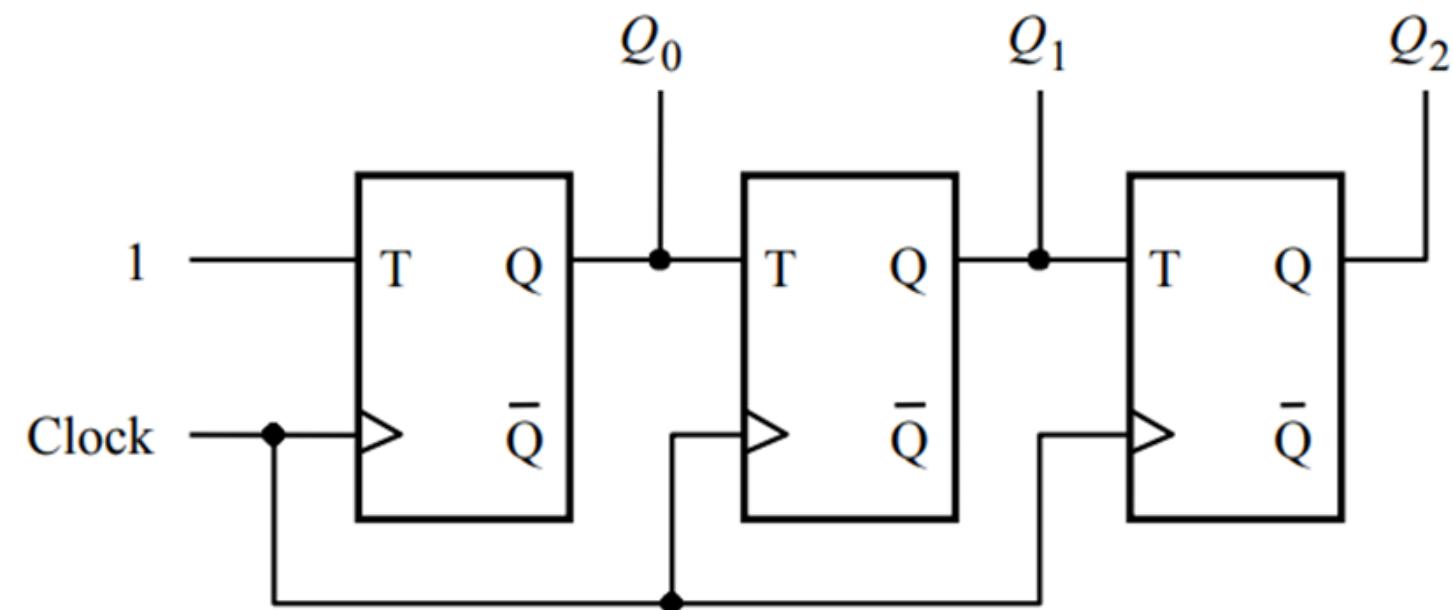


Q: Analyse the given synchronous sequential circuit. Find out the after 4 clock cycles if initial state is

$$\underline{Q_2 Q_1 Q_0 = 000}$$

Ans:

$Q_2 \ Q_1 \ Q_0$
0 0 0



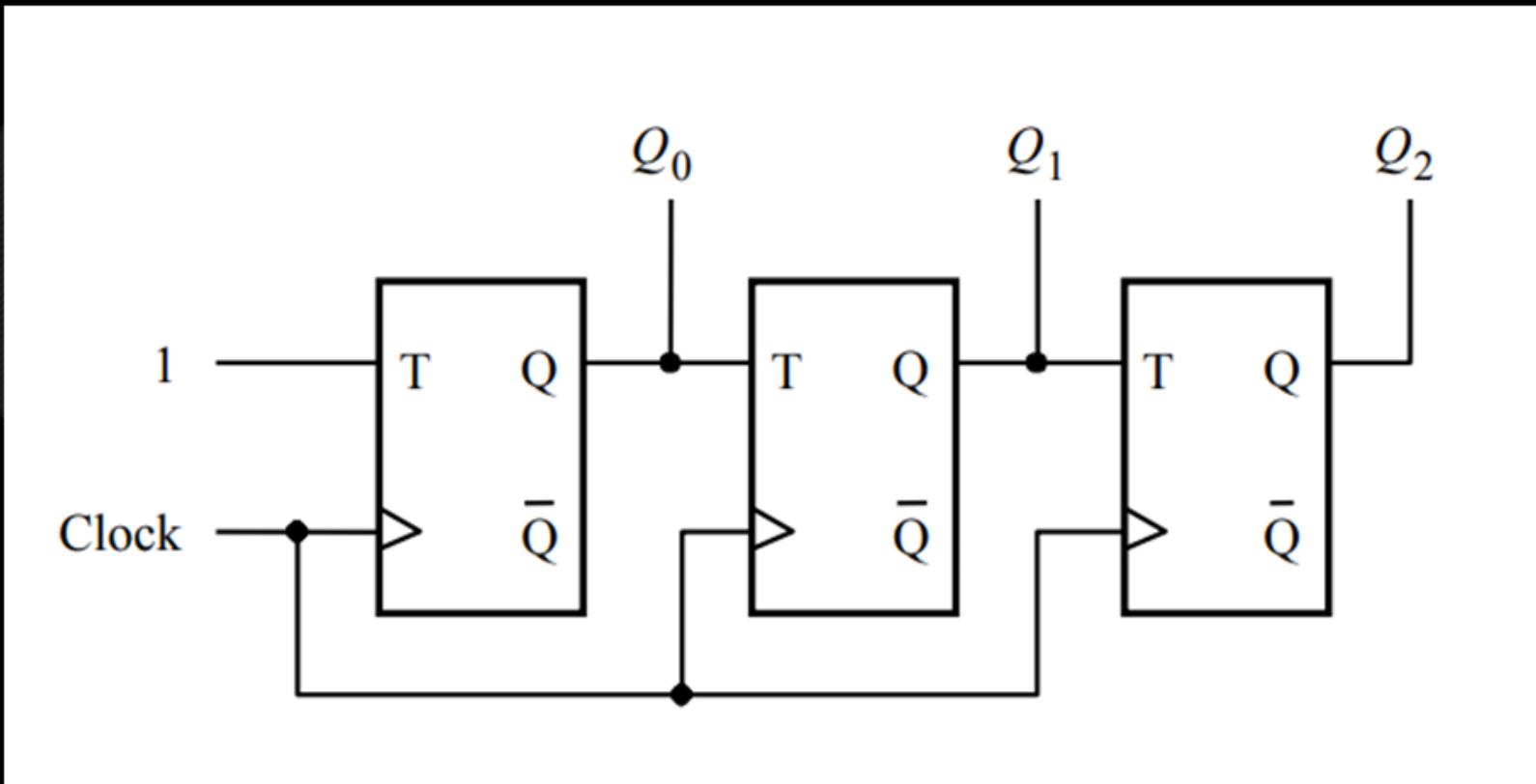
	Q_2	Q_1	Q_0	
<u>Initial</u>	0	0	0	
$t=1$	0	0	1	✓
$t=2$	0	1	0	✓
	1	1	1	✓
	0	0	0	✓

After one fl
of clock)

Another fl

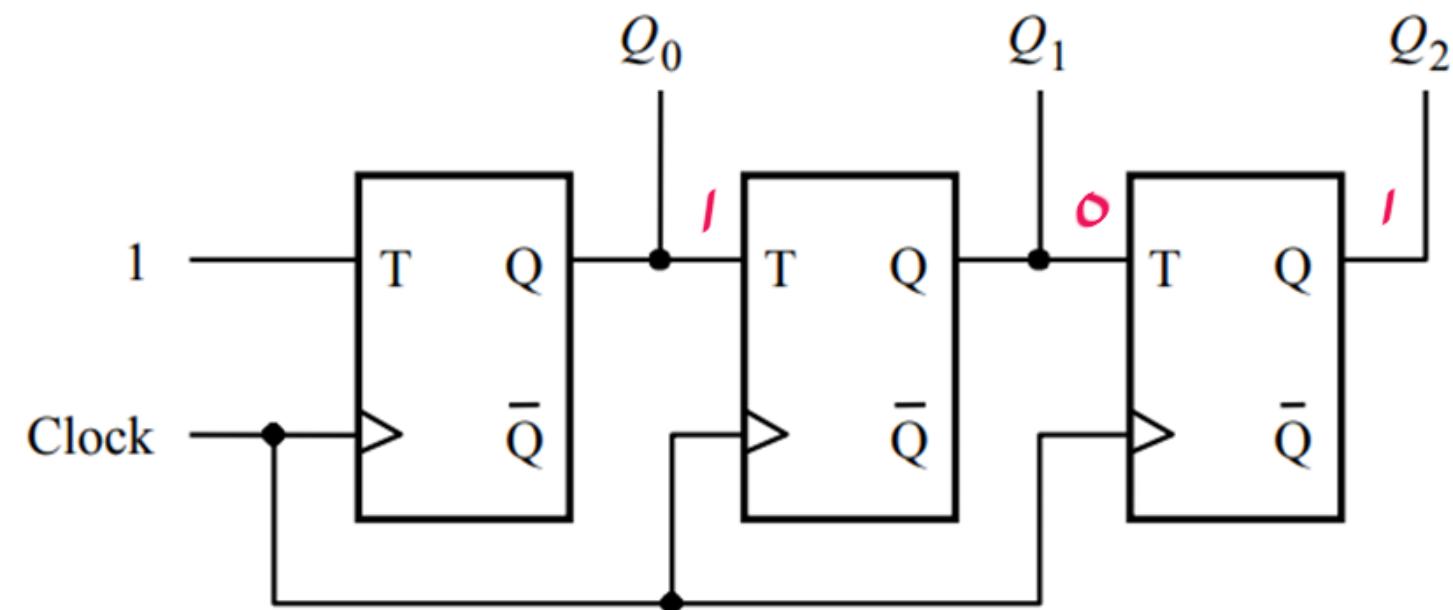


Q: Analyse the given synchronous sequential circuit. Find out the next state if initial state is $Q_2Q_1Q_0 = 101$



Q: Analyse the given synchronous sequential circuit. Find out the next state if initial state is $Q_2Q_1Q_0 = \underline{101}$

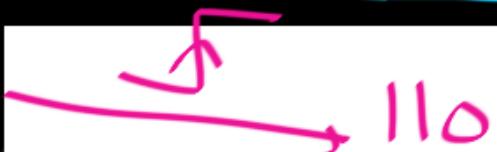
initial :

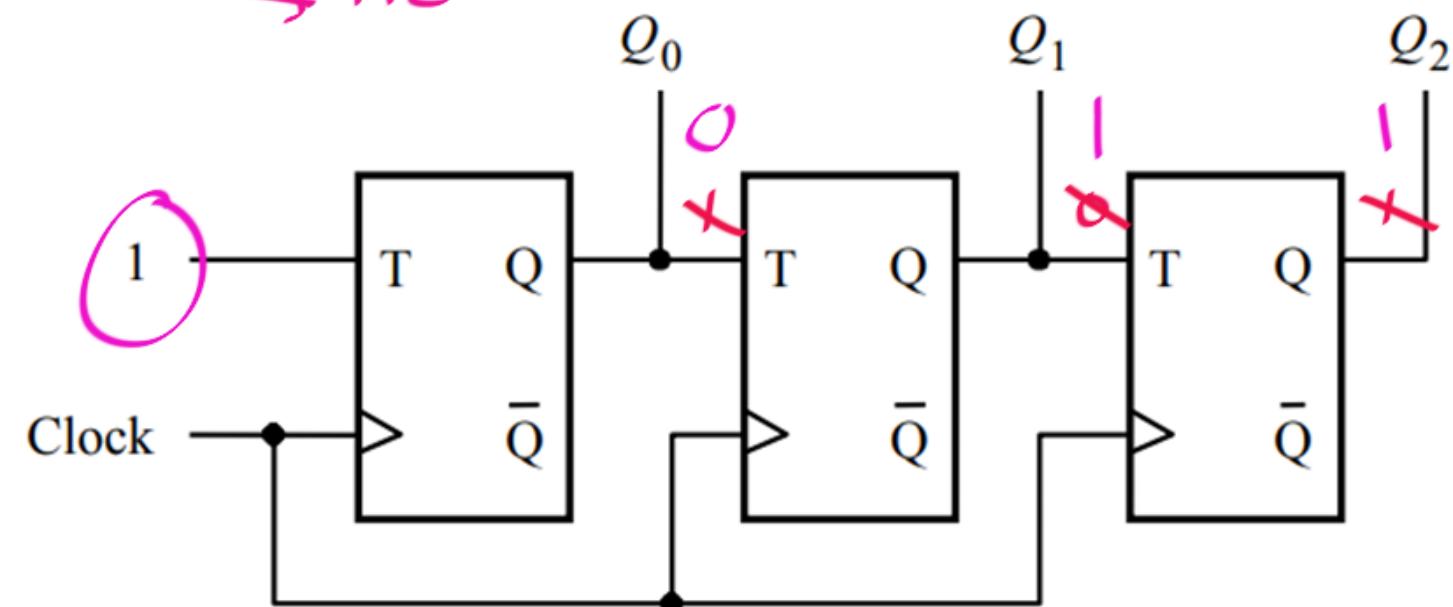


Q: Analyse the given synchronous sequential circuit. Find out the next state if initial state is

$$Q_2 Q_1 Q_0 = \underline{101}$$

After
one
Clock
Pulse

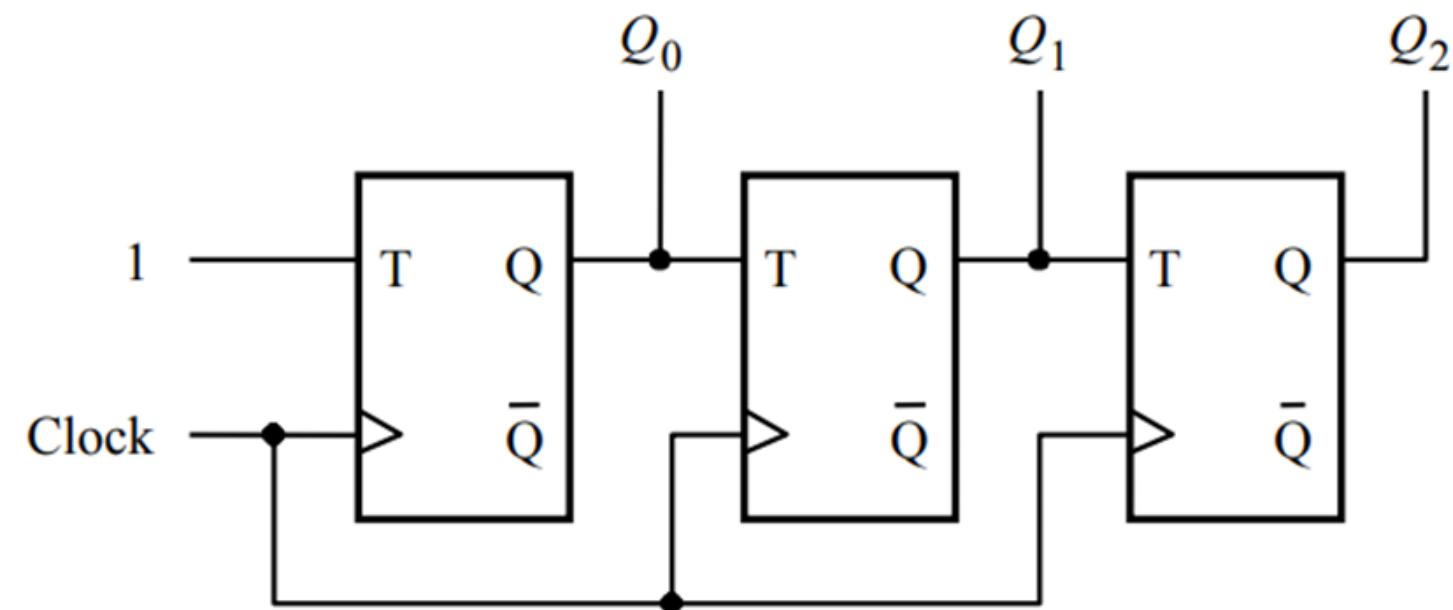
 101 → 110





Q: Analyse the given synchronous sequential circuit. Find out the after 4 clock cycles if initial state is

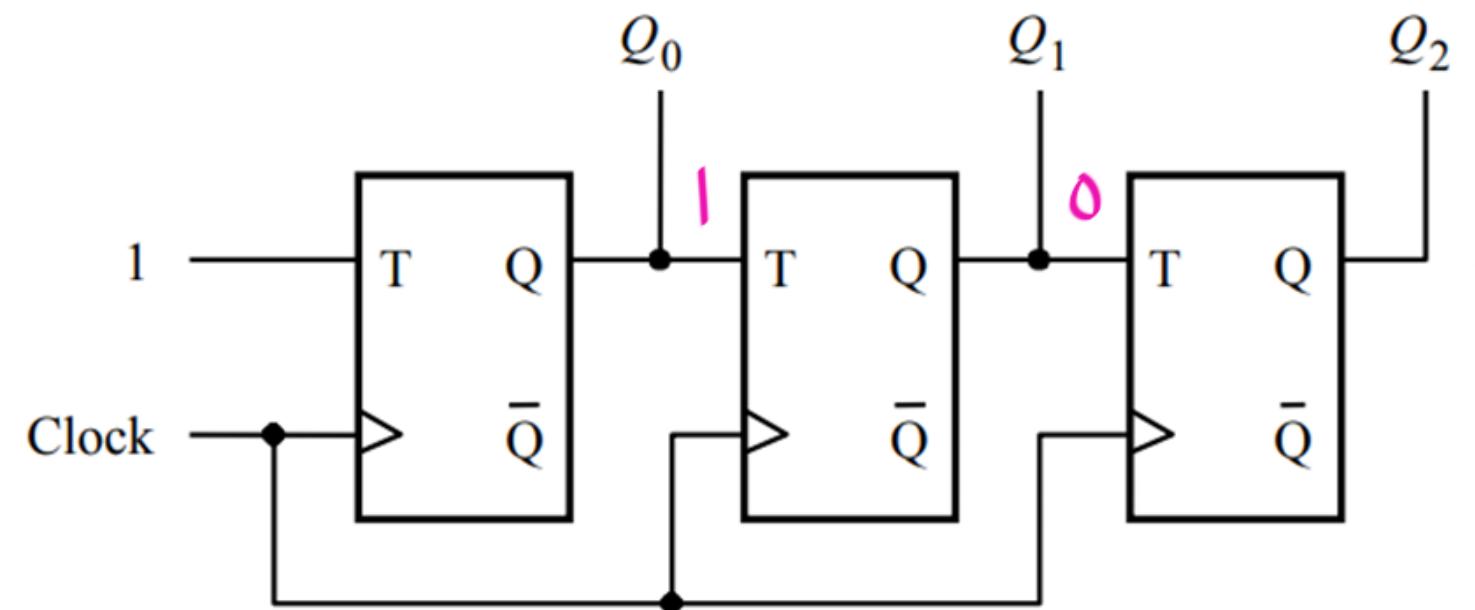
$$Q_2 Q_1 Q_0 = 101$$



Q: Analyse the given synchronous sequential circuit. Find out the after 4 clock cycles if initial state is

$$Q_2 Q_1 Q_0 = 101$$

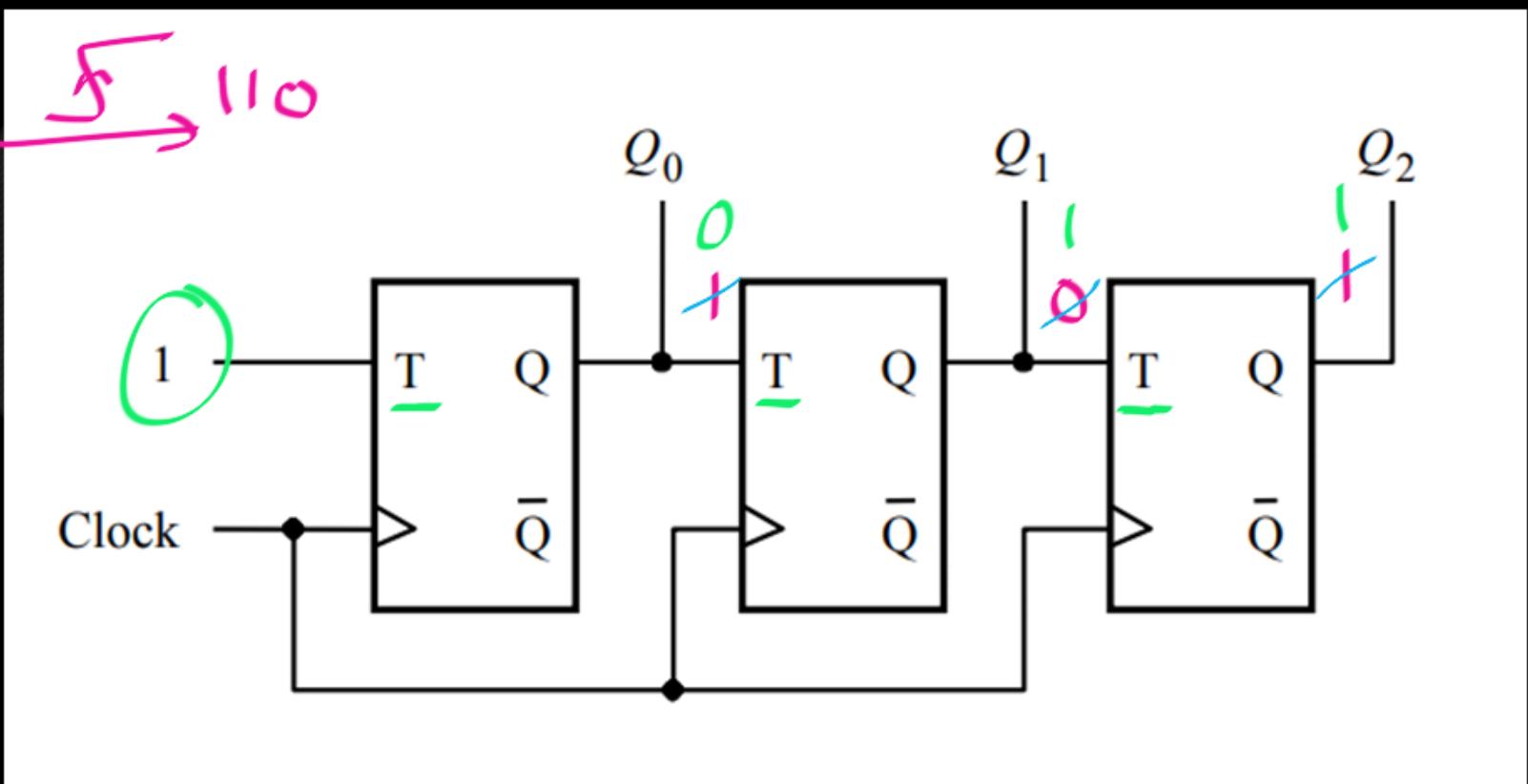
Initial :



Q: Analyse the given synchronous sequential circuit. Find out the after 4 clock cycles if initial state is

$$Q_2 Q_1 Q_0 = 101$$

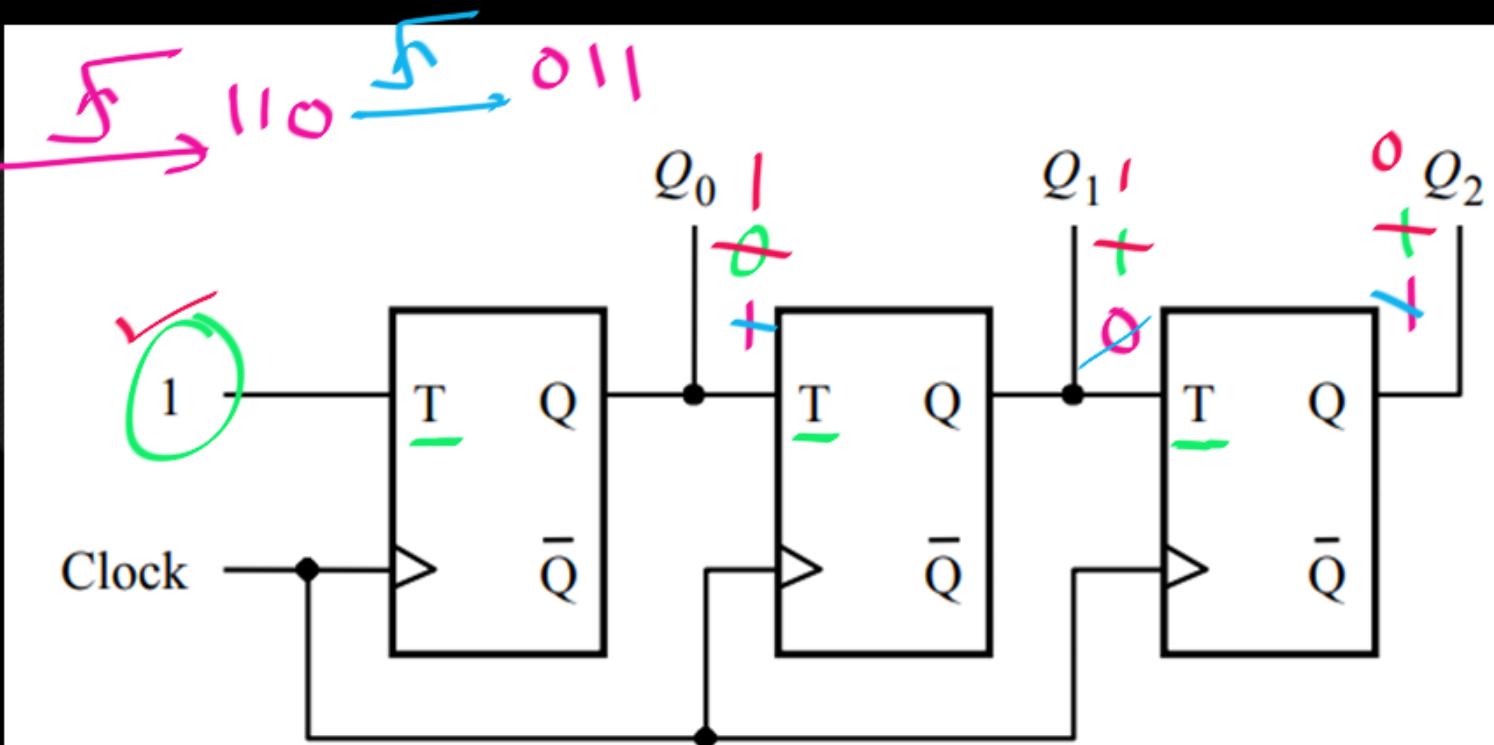
~~After one :
F~~



Q: Analyse the given synchronous sequential circuit. Find out the after 4 clock cycles if initial state is

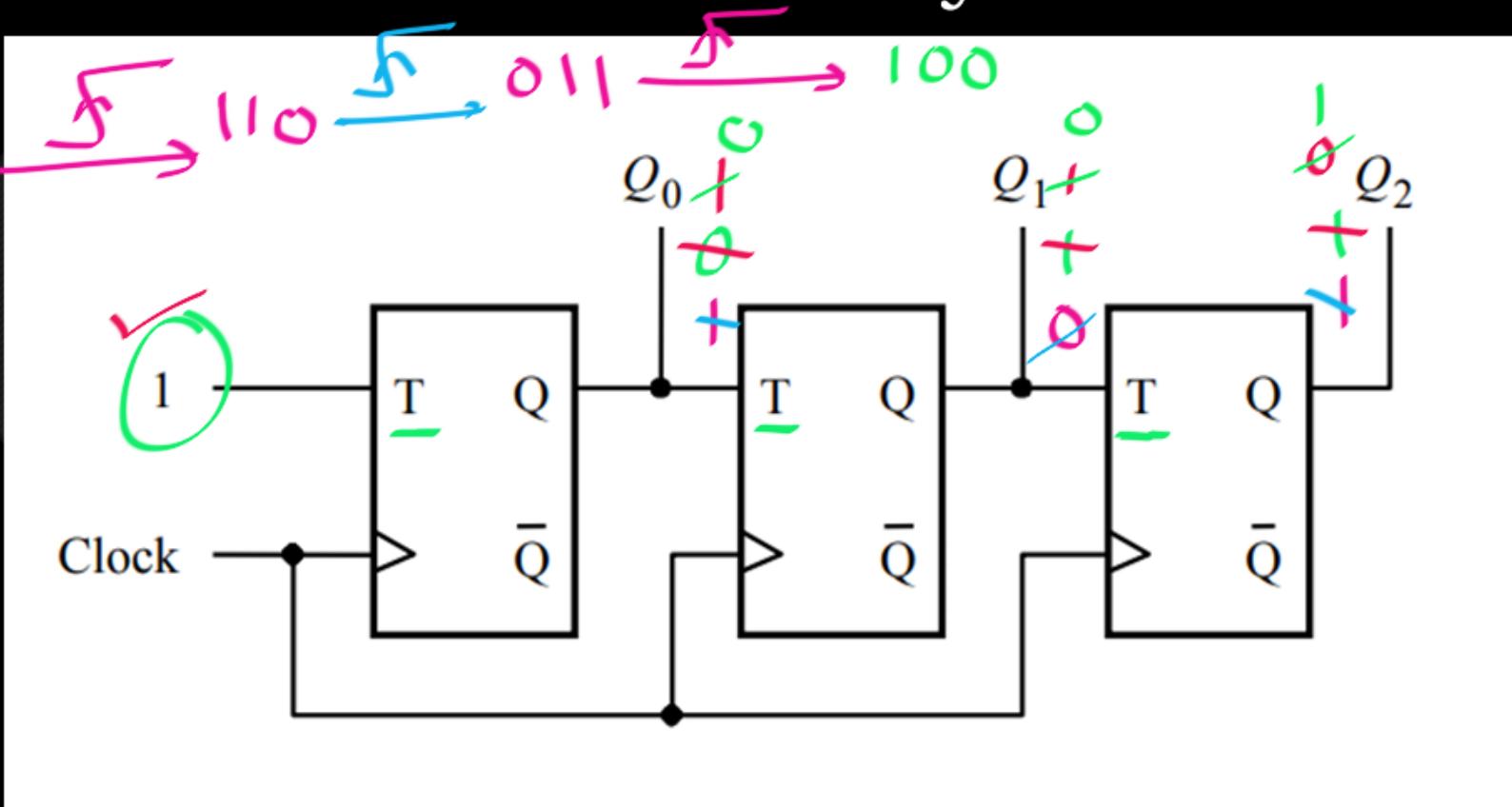
$$Q_2 Q_1 Q_0 = 101$$

After
Another :
 $\xrightarrow{\Delta}$



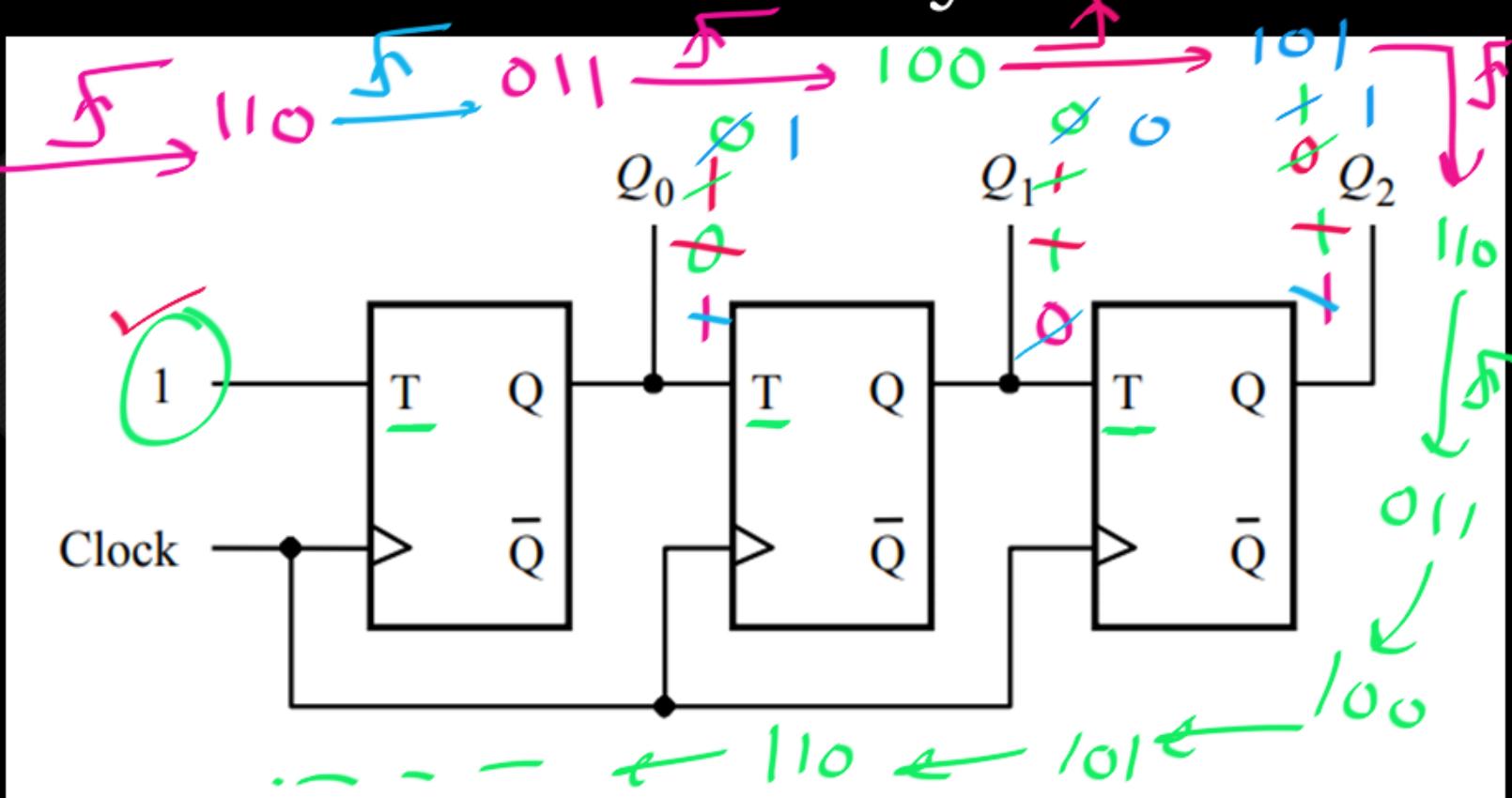
Q: Analyse the given synchronous sequential circuit. Find out the after 4 clock cycles if initial state is $Q_2Q_1Q_0 = 101$

After
Another :



Q: Analyse the given synchronous sequential circuit. Find out the after 4 clock cycles if initial state is $Q_2Q_1Q_0 = 101$

~~After
Another :~~





So far,

One type of Analysis :

Useful when if we
need to find State after small
no. of cycle.



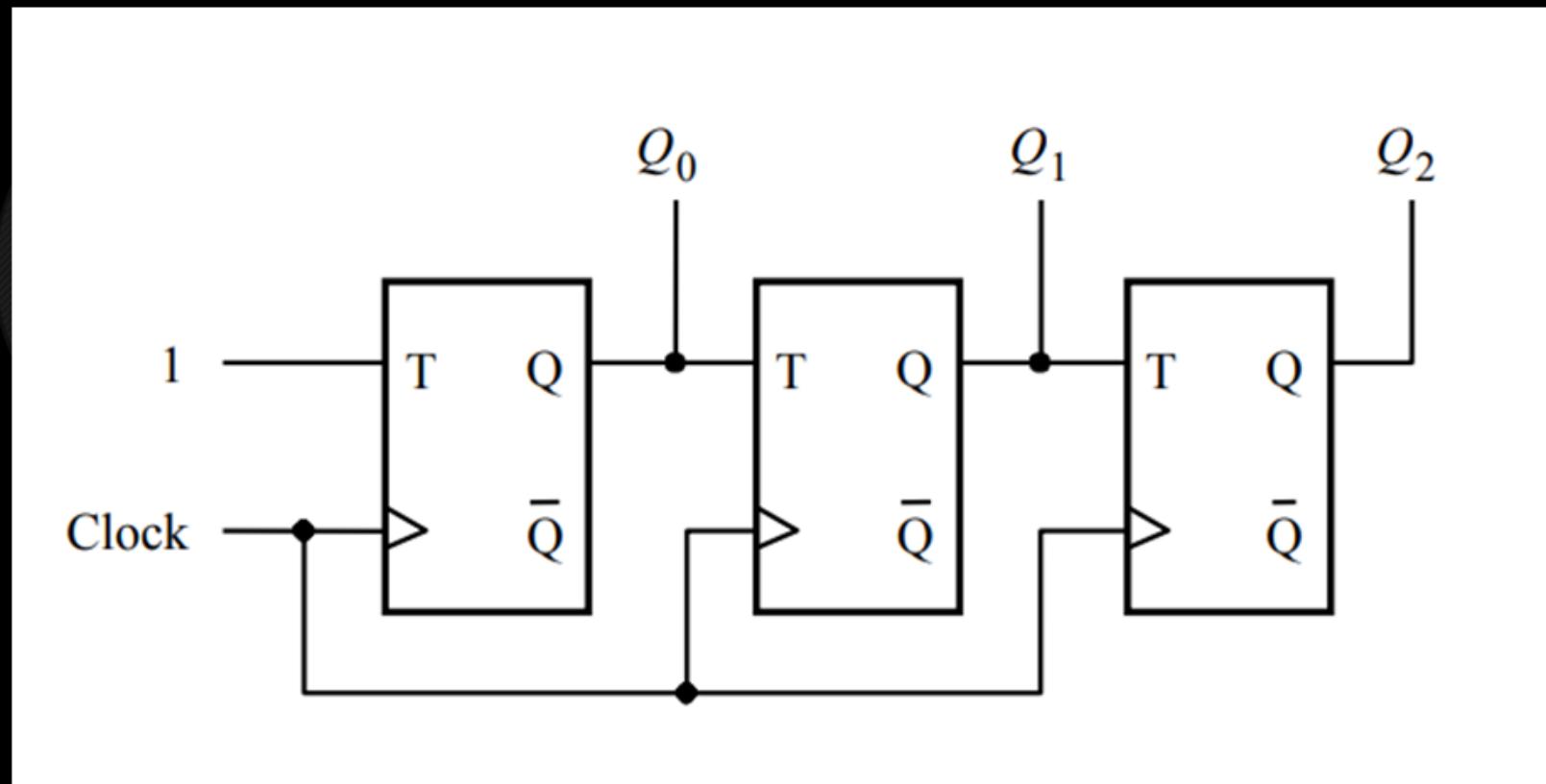
Let's see "Complete Analysis"

of the Synchronous Ckt.

Complete Analysis of Any Synchronous Sequential Circuit (When No External Input):

1. Find out FlipFlop Inputs in terms of Present state.
2. Find out the Next State from FlipFlip Input Combination.
3. Find out the Next State Equation in terms of Present States.
4. Find out Sate Diagram, State Transition table.

Q: Complete Analyse the given synchronous sequential circuit:





Present State			Next State			Q_{ONext}
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	
0	0	0	?			
0	0	1	?	?		
0	1	0	?	?		
0	1	1	?	?		
1	0	0	?			
1	0	1	?			
1	1	0	?			
1	1	1	?			

find Next
state.

Target

Q: Complete Analyse the given synchronous sequential circuit:

Toggles in every cycle

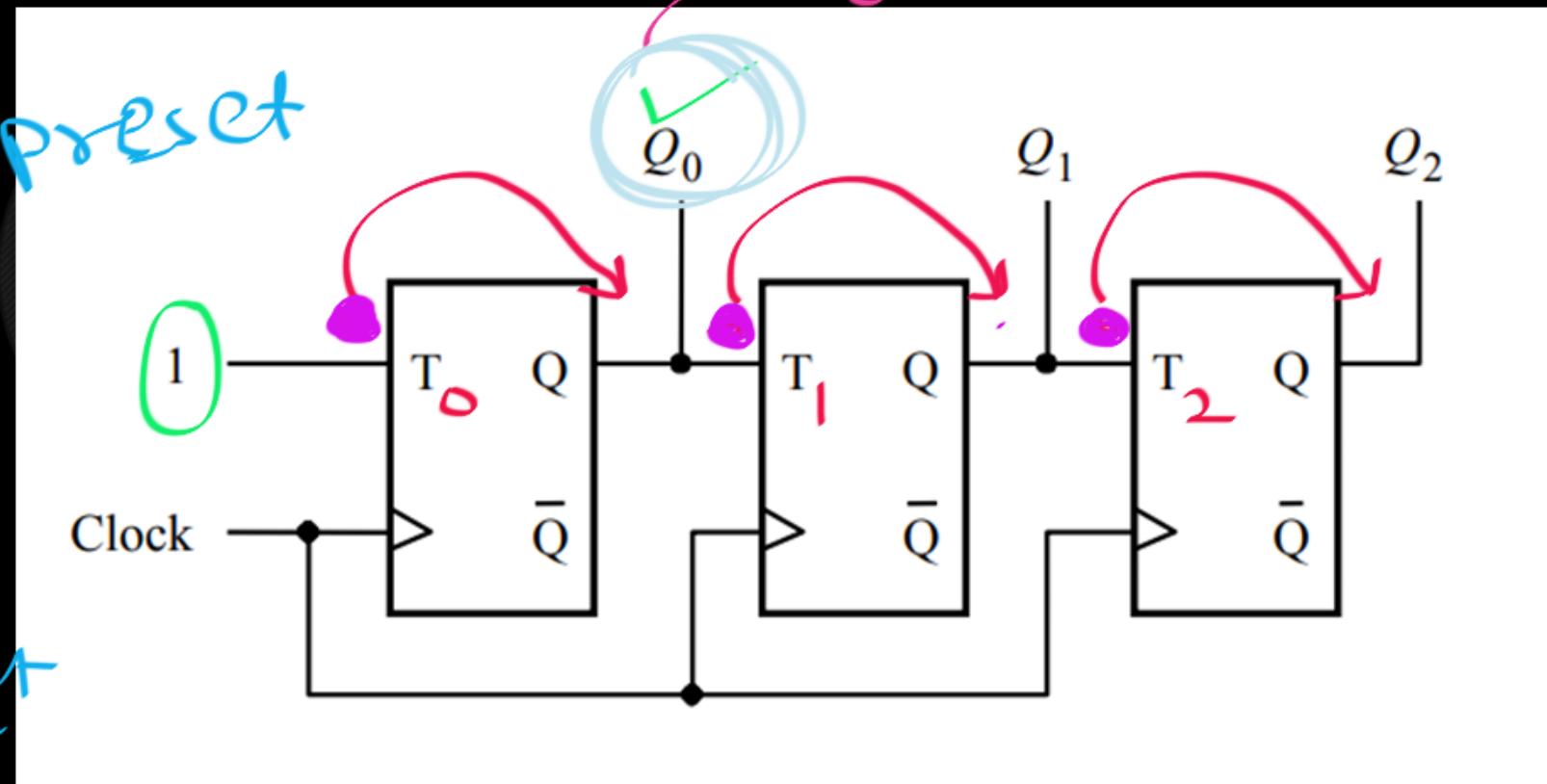
$$\overline{T_0} = 1$$

preset

$$\overline{T_1} = Q_0$$

$$\overline{T_2} = Q_1$$

preset

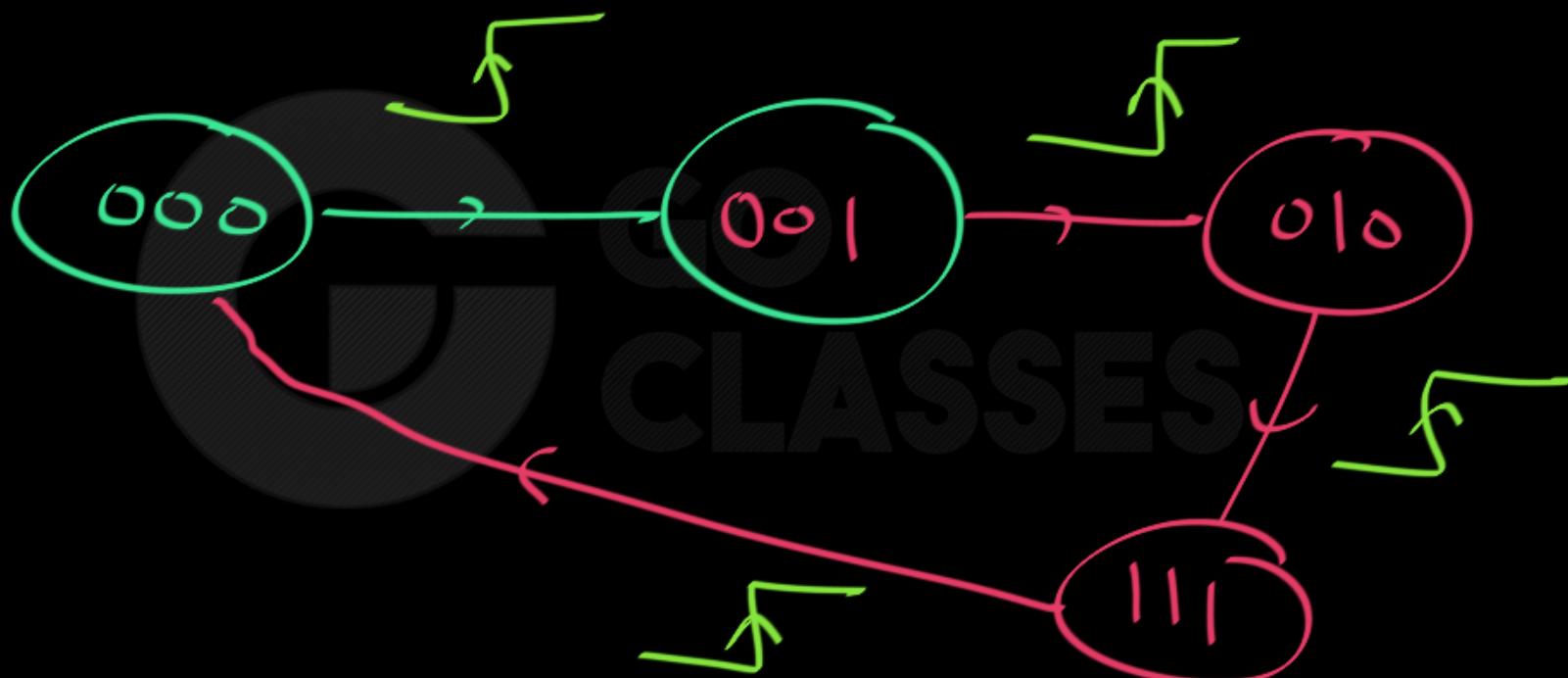




Present state	Next state	flip flop inputs
$Q_2 \ Q_1 \ Q_0$	$Q_2^+ \ Q_1^+ \ Q_0^+$	$T_2 \ T_1 \ T_0$
0 0 0	0 0 1	0 0 1
0 0 1	0 1 0	0 1 0
0 1 0	1 0 1	1 0 1
0 1 1	1 0 0	1 0 0
1 0 0	1 0 1	0 0 1
1 0 1	1 1 0	0 1 0
1 1 0	0 1 1	1 1 0
1 1 1	0 0 0	1 0 1

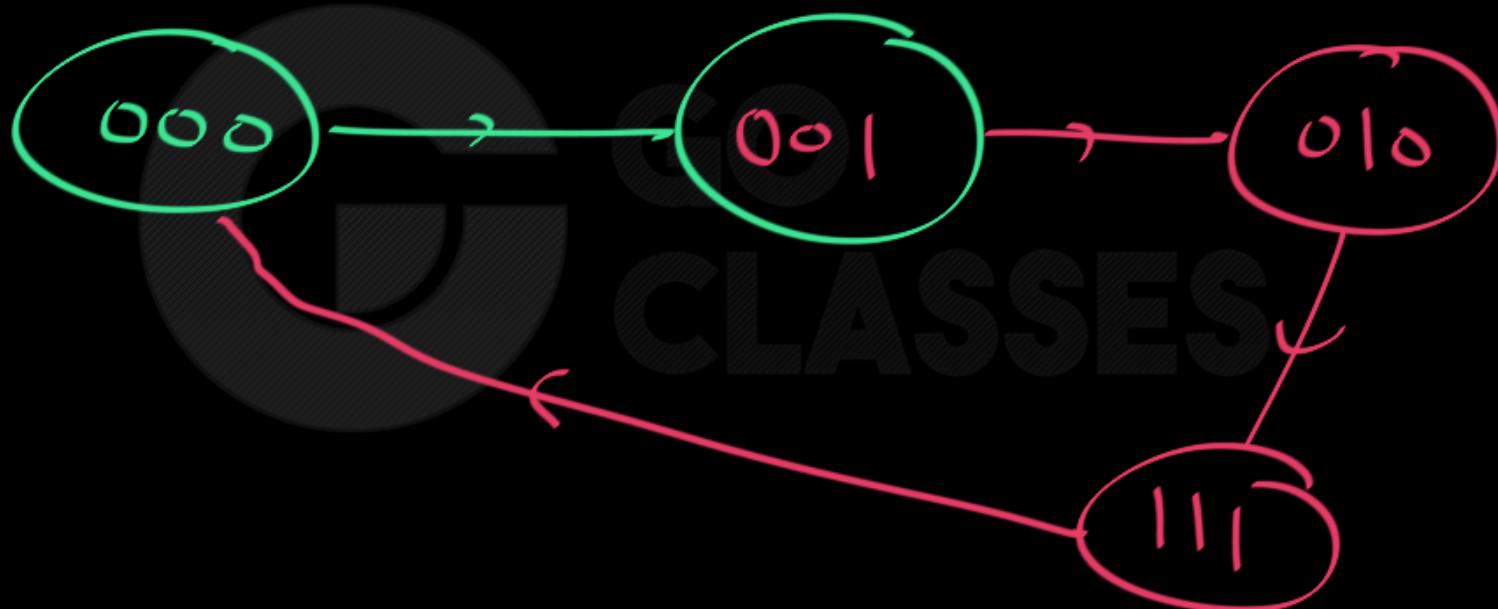


State Diagram:



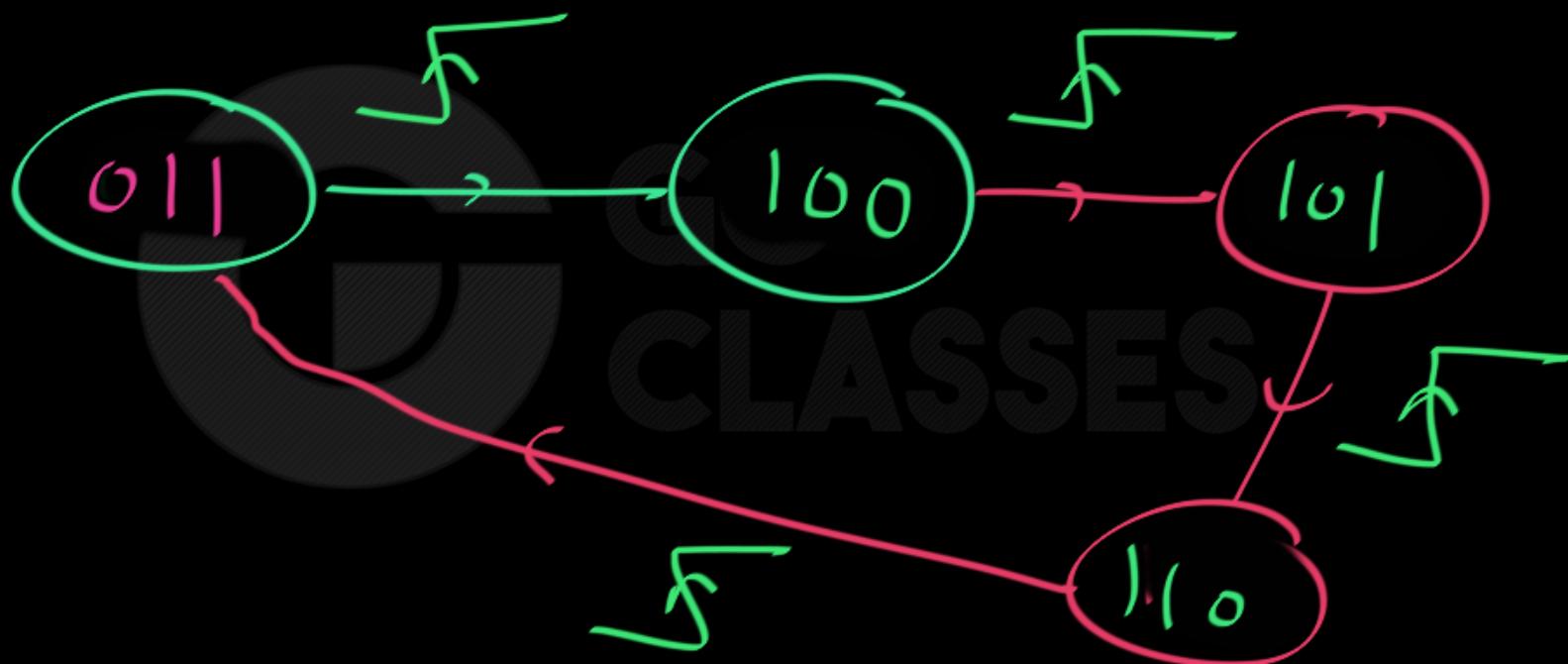


State Diagram:



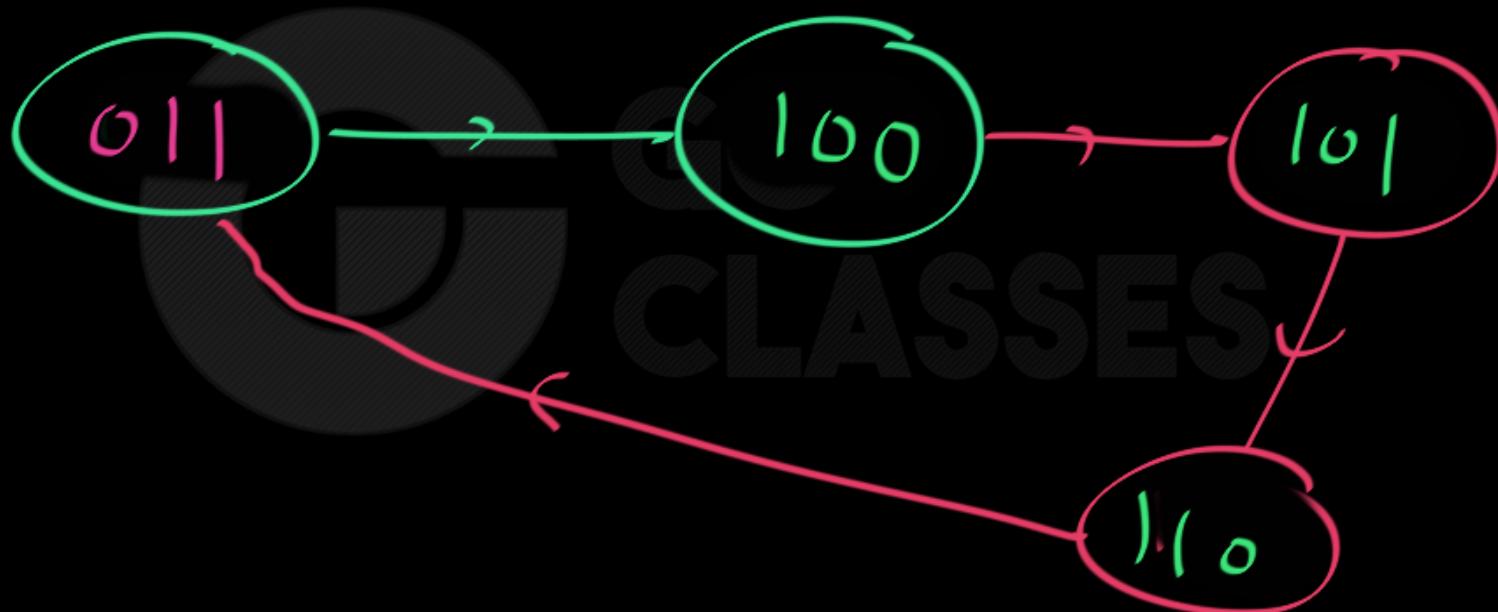


State Diagram:





State Diagram:





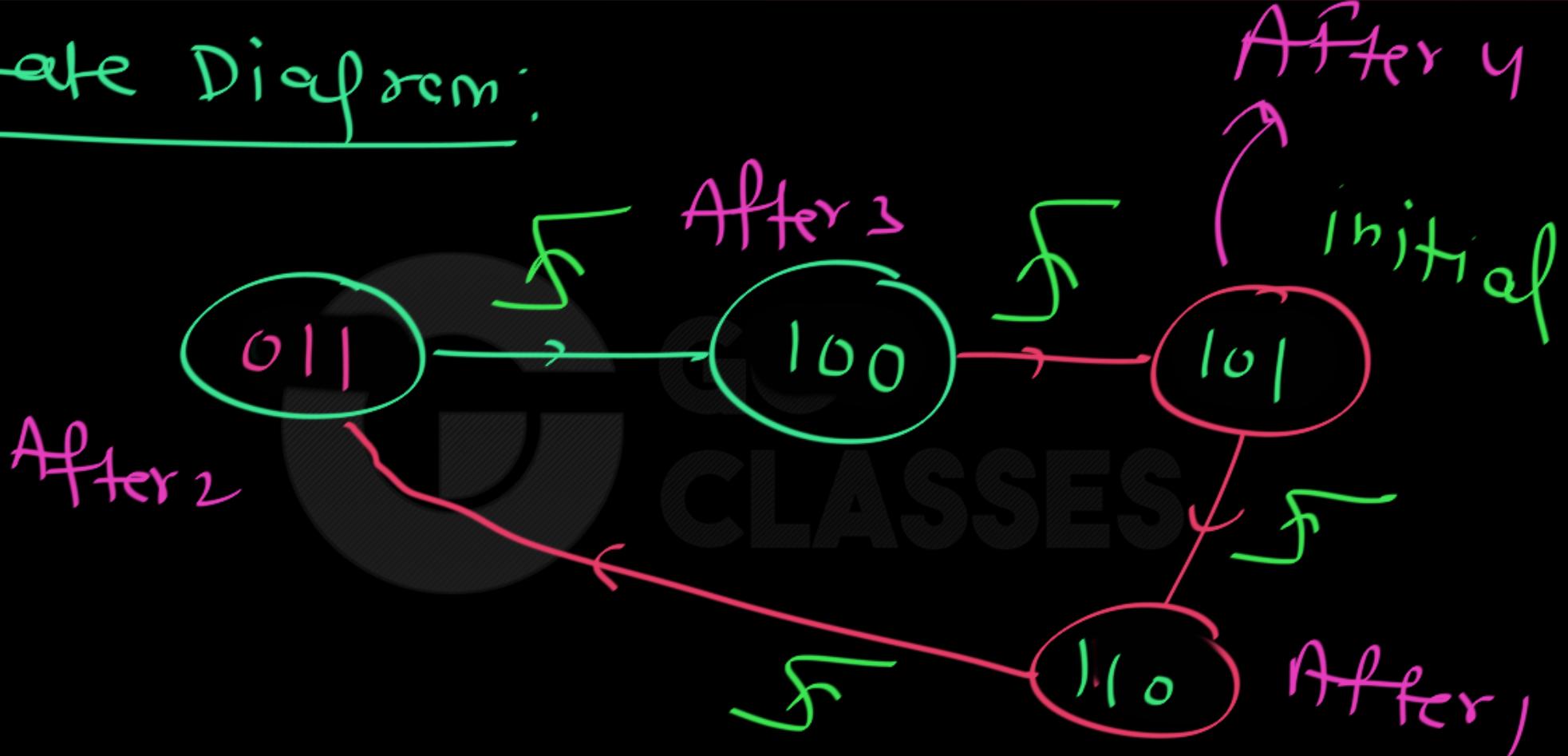
initial

 $Q_2 \ Q_1 \ Q_0$ 

find state after 4 clock cycles ?



State Diagram:





initial

 $Q_2 \ Q_1 \ Q_0$ 

find state after 4 clock cycles ?

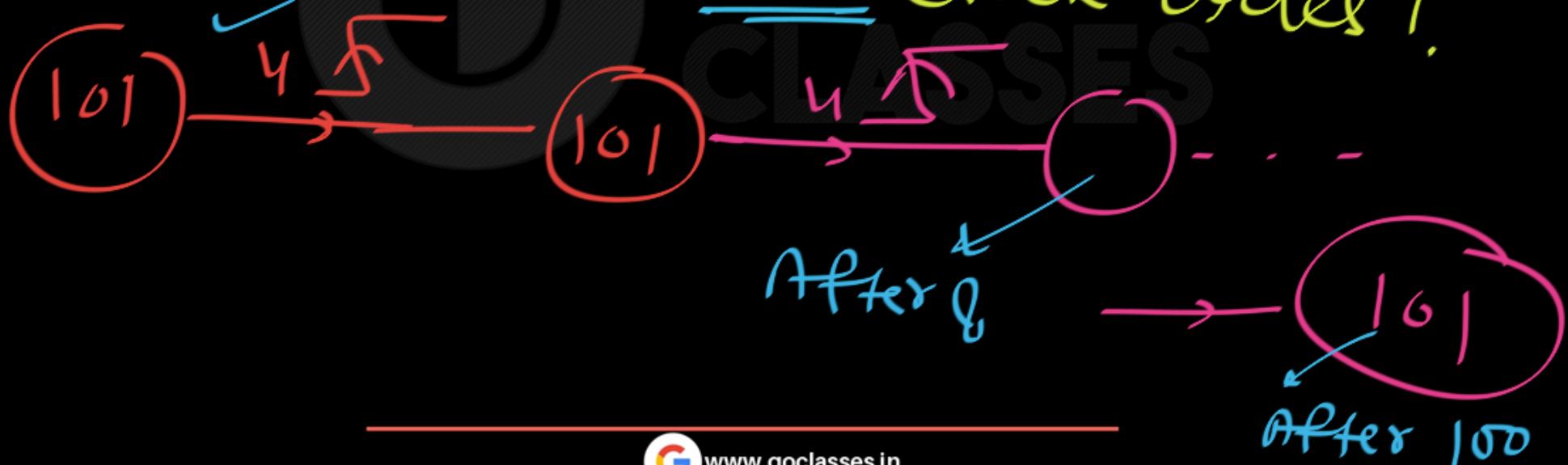
1 0 1

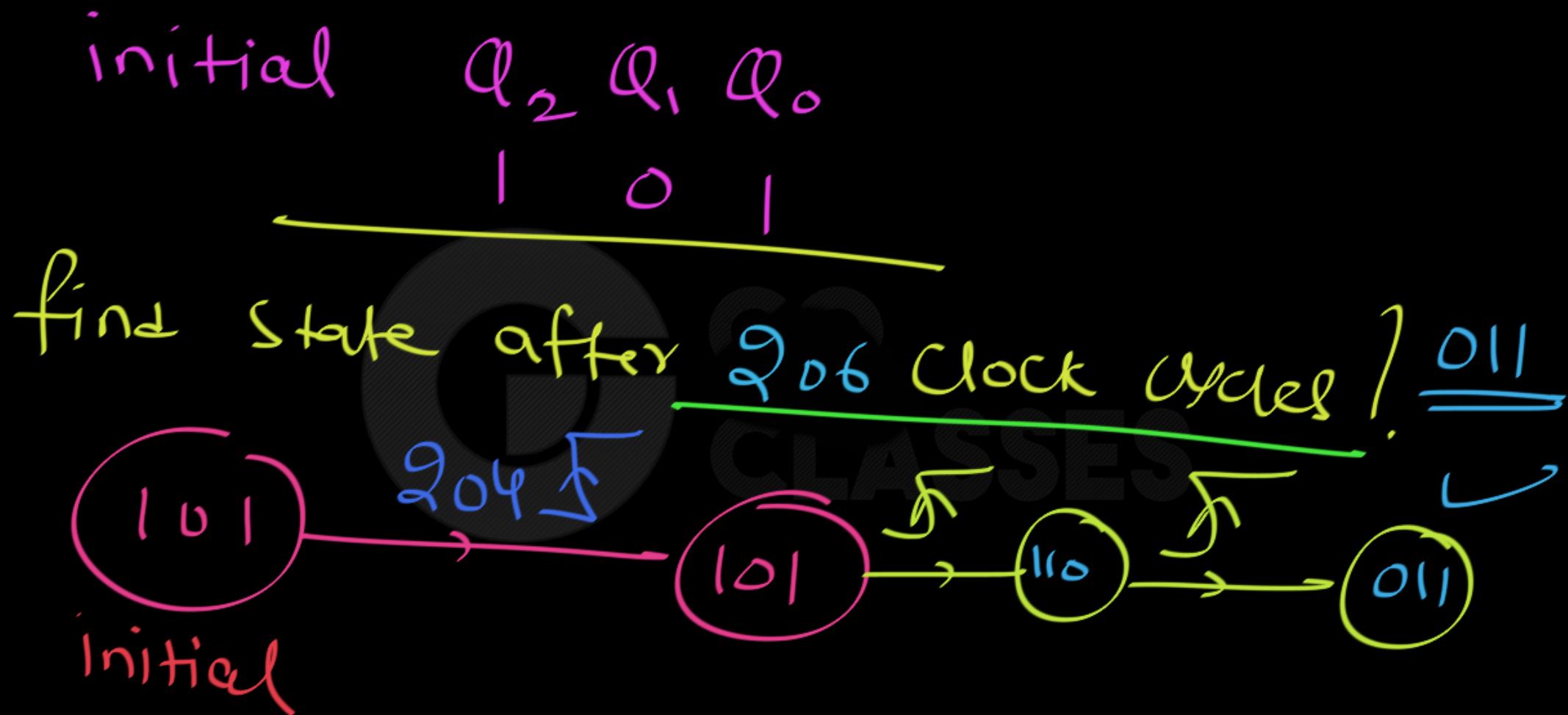
initial

$Q_2 \ Q_1 \ Q_0$

1 0 1

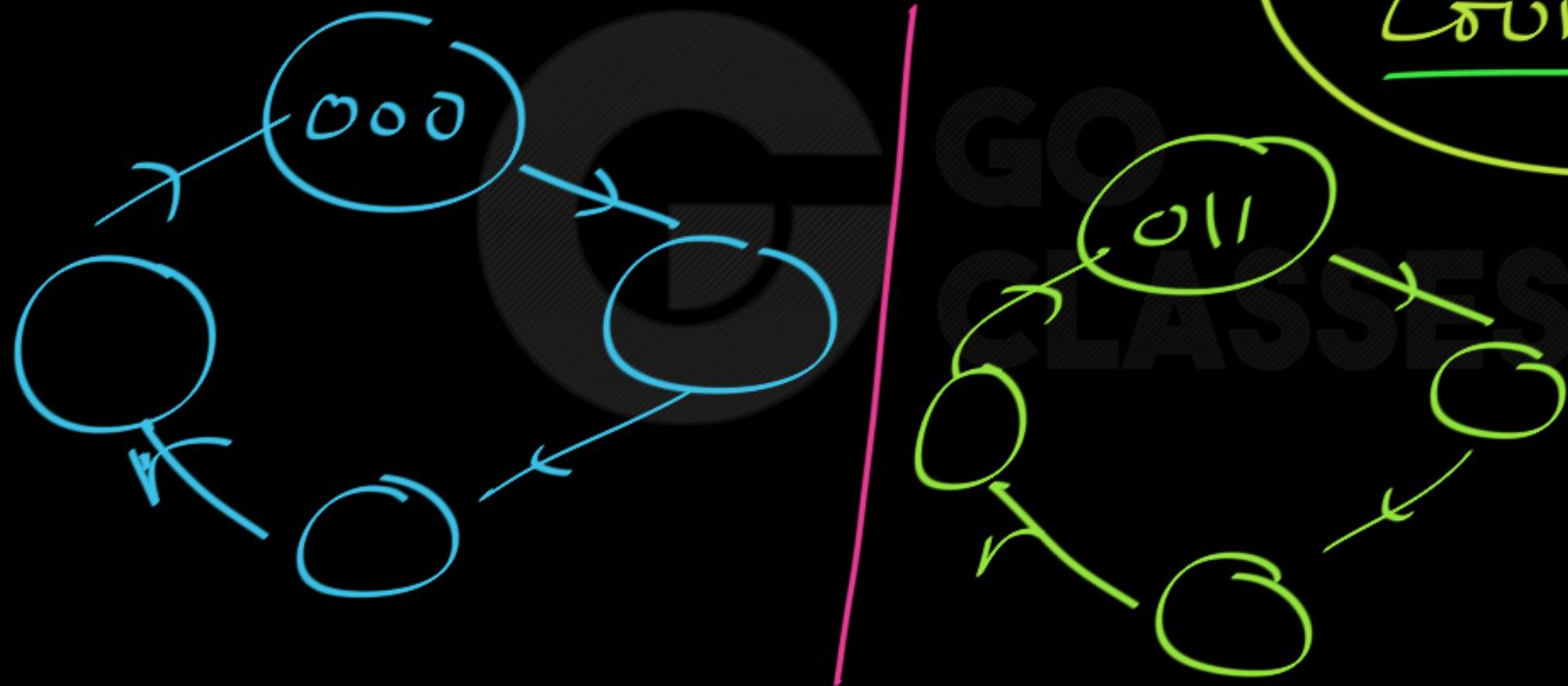
find state after 100 clock cycles !







Previous Counter:



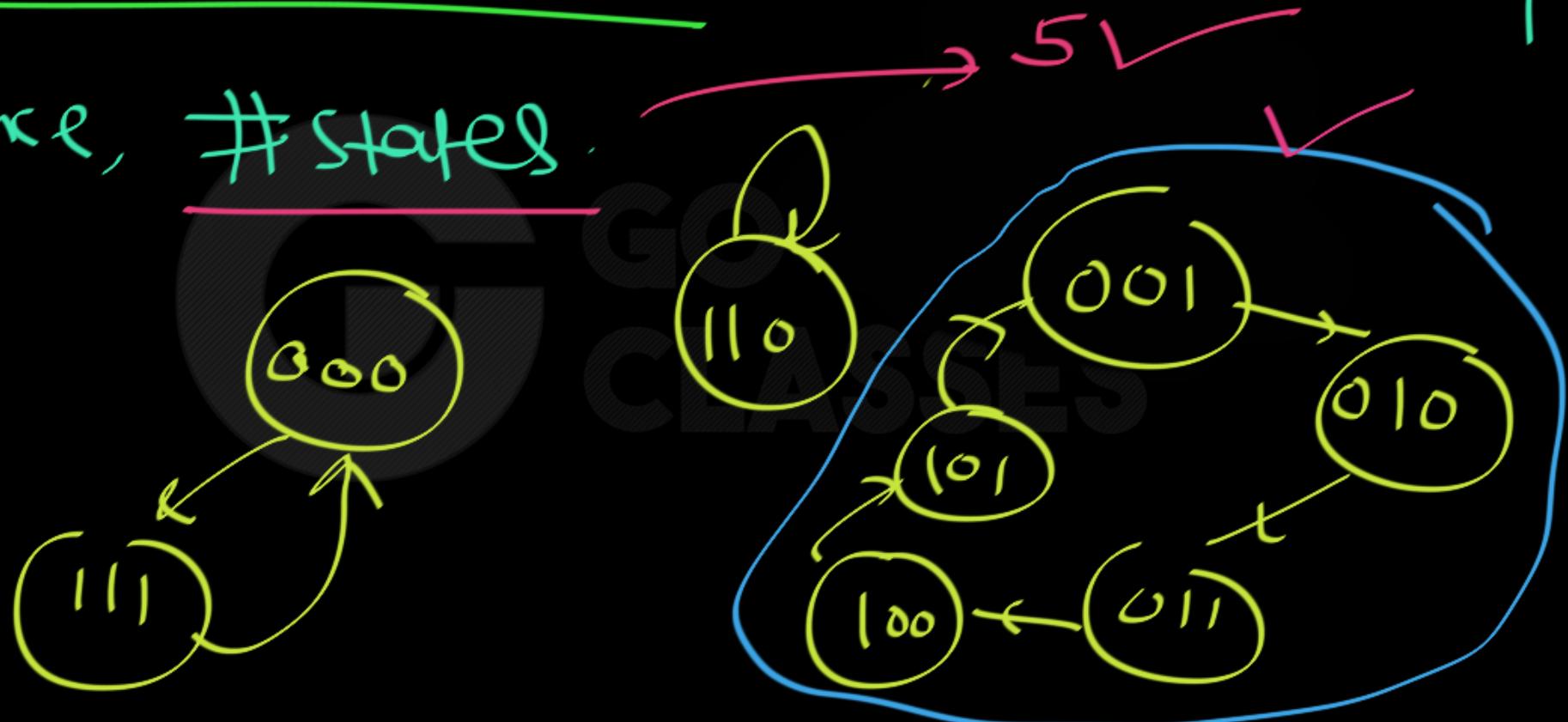


Mod - of a Counter \Rightarrow In a Repeating Sequence, # States.



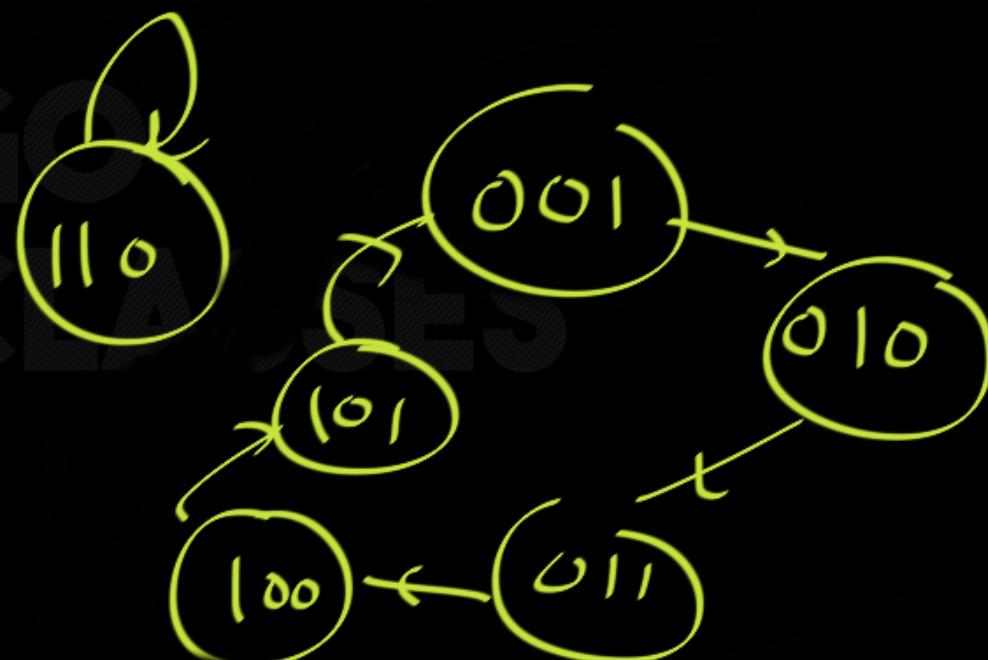
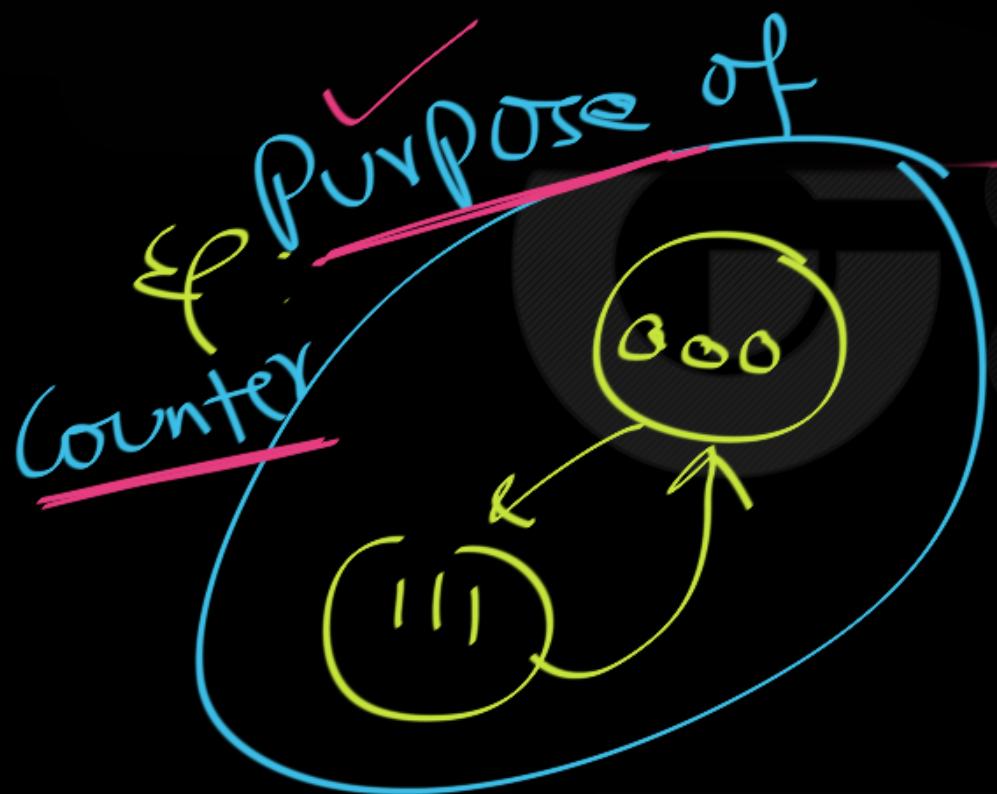
Mod- of this Counter \Rightarrow In a Repeating Sequence, # States

Ex:



mod-of this Counter \Rightarrow

2 ✓



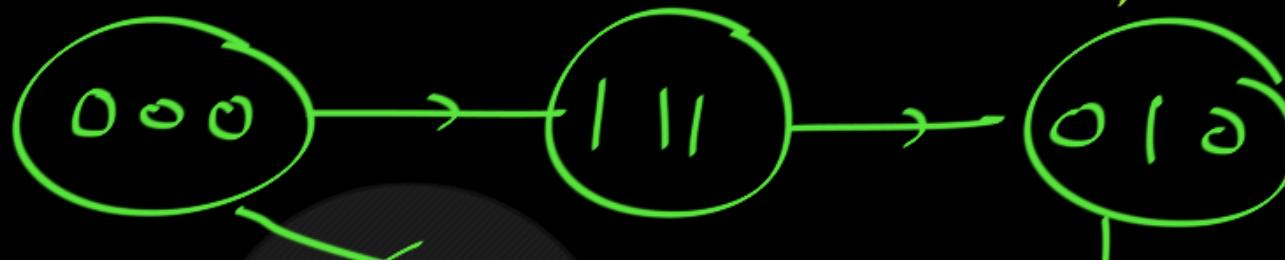
Specially Designed Counters

have a Purpose

fixed Counting Sequence

↑
Ring Counter
Johnson "",
Binary "

Eg: Create a Counter for:



Implementation:

Possible



→ 8 states possible

1 → 3
2 → 4
5 → 6
7 → 0

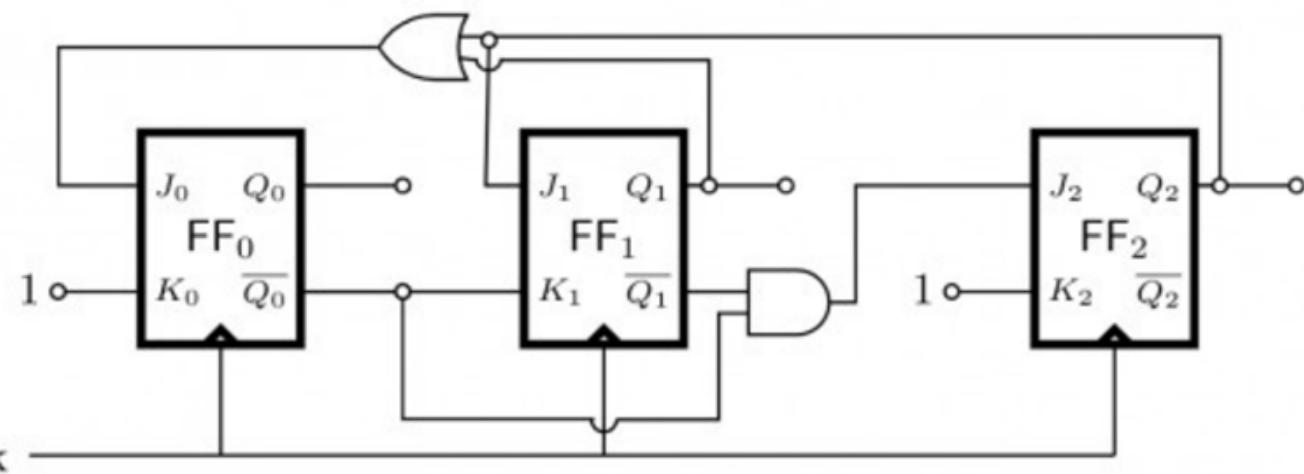
no. of counter = 3

Treat remaining states as Don't Cares

4.30.2 Sequential Circuit: GATE CSE 1990 | Question: 5-c top b<https://gateoverflow.in/85400>

For the synchronous counter shown in Fig.3, write the truth table of Q_0 , Q_1 , and Q_2 after each pulse, starting from $Q_0 = Q_1 = Q_2 = 0$ and determine the counting sequence and also the modulus of the counter.

tests.g:



tests.gatecse.in

tests.gatecse.in

goclasses.in

tests.gatecse.in

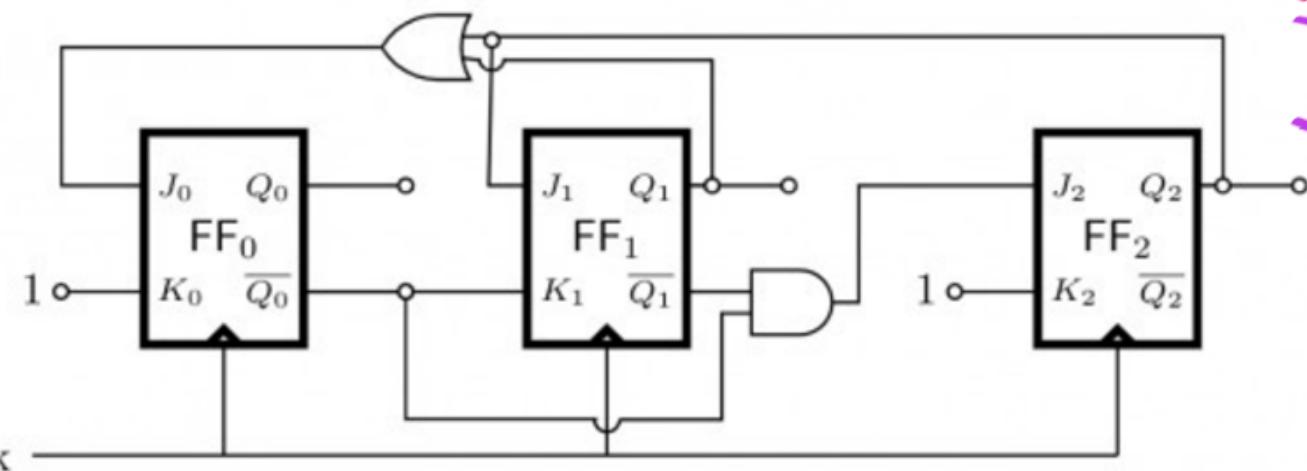


4.30.2 Sequential Circuit: GATE CSE 1990 Question: 5-c top b

Subjective GATE Exam<http://gateoverflow.in/85400>

For the synchronous counter shown in Fig.3, write the truth table of Q_0 , Q_1 , and Q_2 after each pulse, starting from $Q_0 = Q_1 = Q_2 = 0$ and determine the counting sequence and also the modulus of the counter.

Convention
(msb)



tests.gatecse.in

goclasses.in

tests.gatecse.in



Analysis 1 :

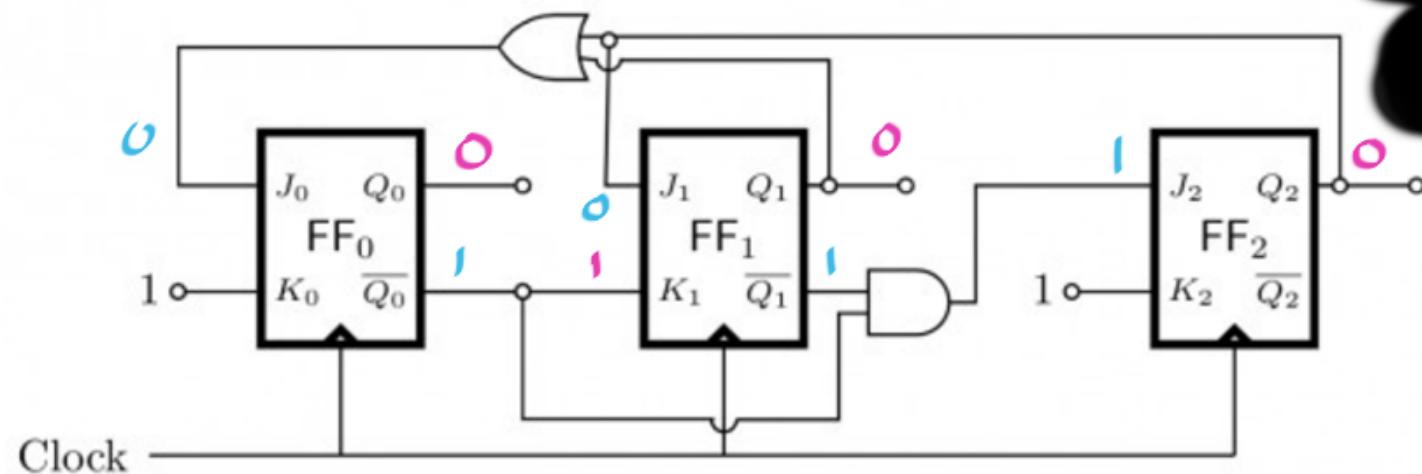
Start with $Q_2 \ Q_1 \ Q_0$

0 0 0



F J

Initial : *tests.g*



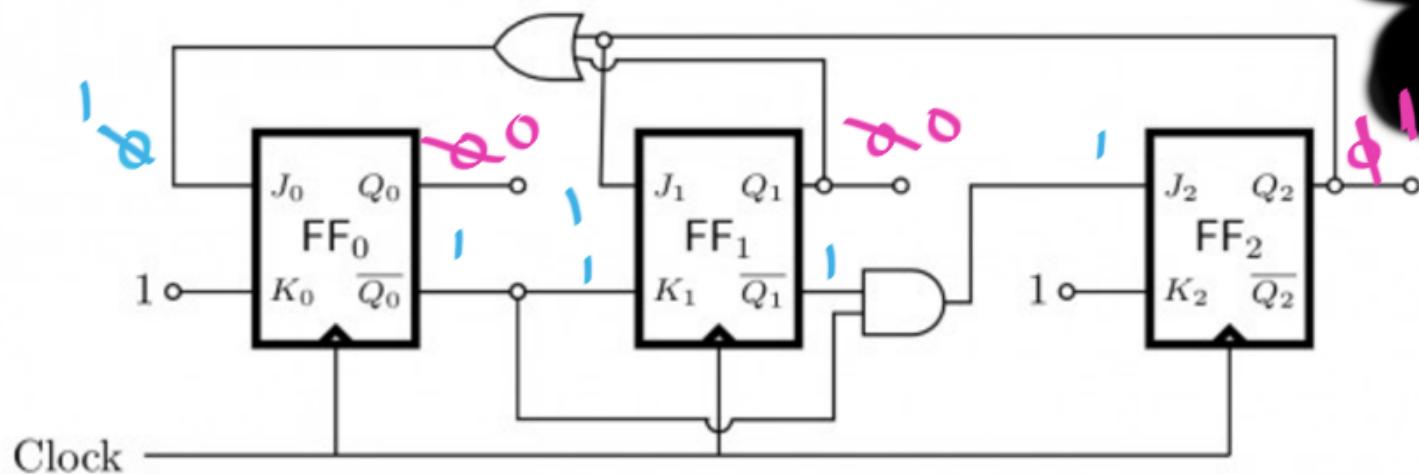
tests.gatecse.in

goclasses.in

tests.gatecse.in

$Q_2 Q_1 Q_0$
000 → 100 →
MSB

After
one
clock



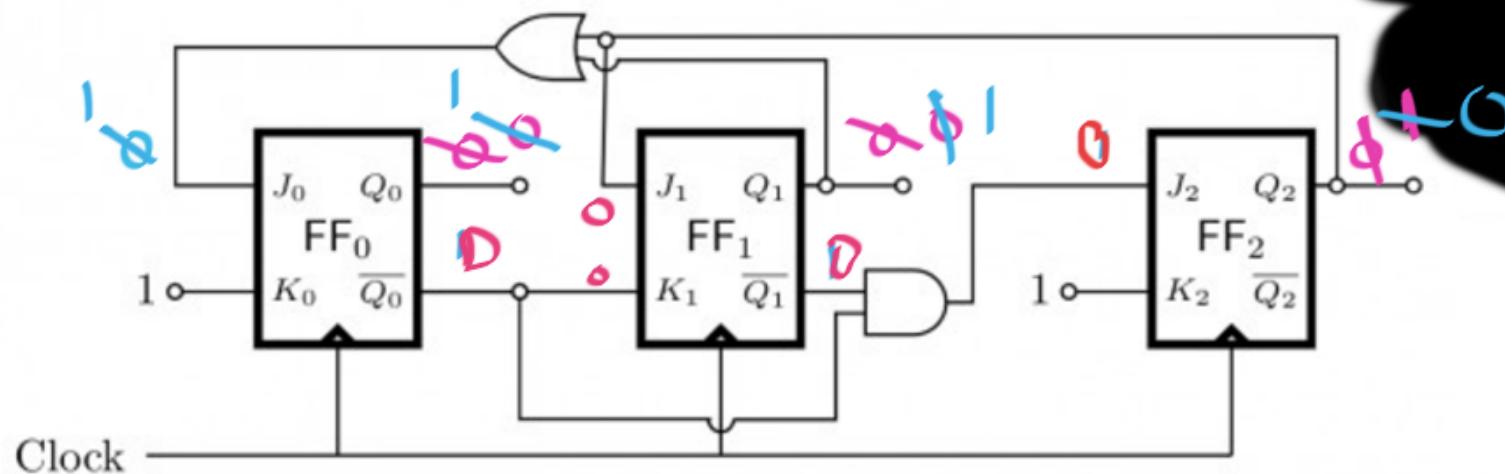
tests.gatecse.in

goclasses.in

tests.gatecse.in



After one step



tests.gatecse.in

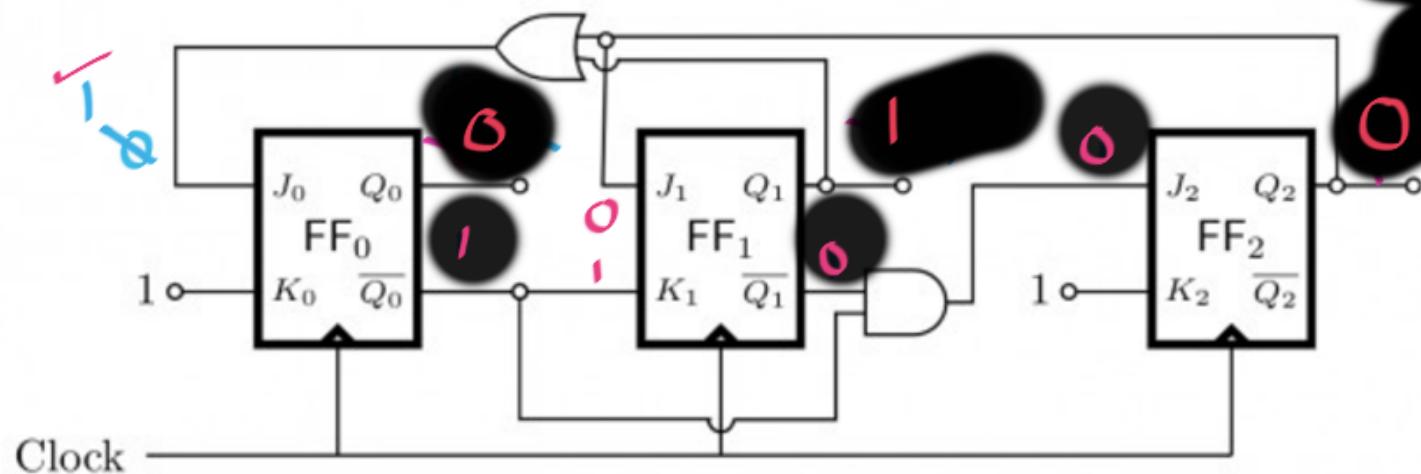
goclasses.in

tests.gatecse.in

$Q_2 Q_1 Q_0$
000 → 100 →
Ans

0 | 1 → 0 | 0 →

After tests.g:
one
5



tests.gatecse.in

goclasses.in

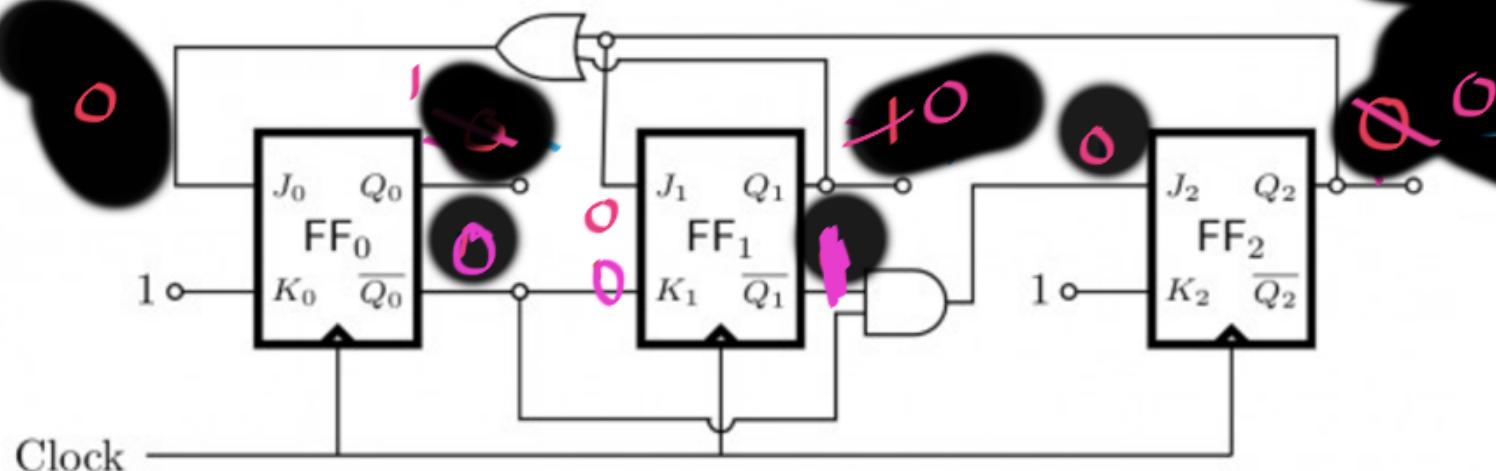
tests.gatecse.in

$Q_2 Q_1 Q_0$
000 → 100 → 011 → 010 → 001
MSB

After

one

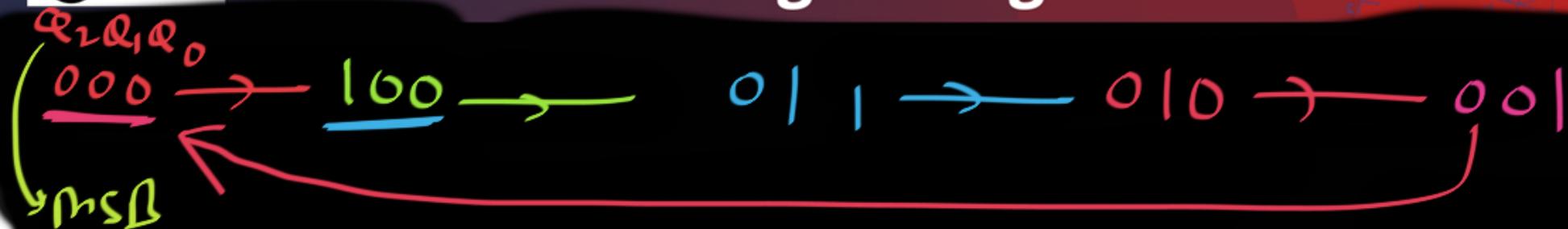
step



tests.gatecse.in

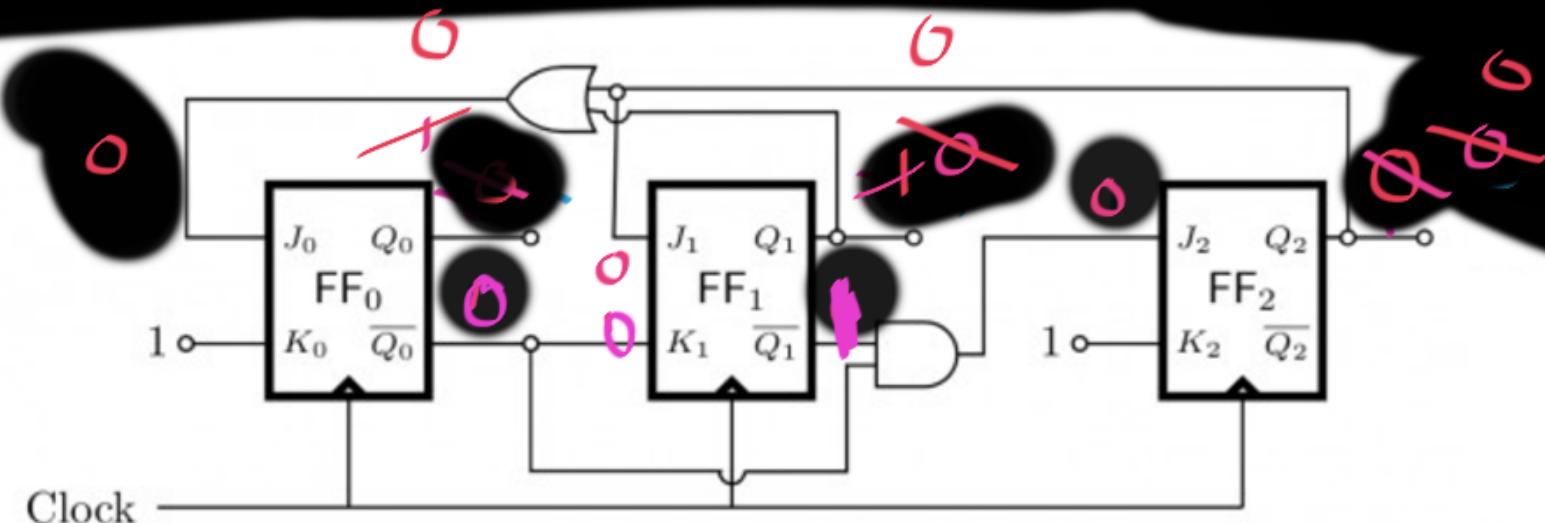
goclasses.in

tests.gatecse.in



After tests.g:

one
5



tests.gatecse.in

goclasses.in

tests.gatecse.in

4.30.2 Sequential Circuit: GATE CSE 1990 Question: 5-c top b

Subjective GATE Exam<http://gateoverflow.in/85400>

For the synchronous counter shown in Fig.3, write the truth table of Q_0 , Q_1 , and Q_2 after each pulse, starting from $Q_0 = Q_1 = Q_2 = 0$ and determine the counting sequence and also the modulus of the counter.

$$\overline{J_0} = Q_2 + Q_1$$

tests.gatecse.in

$$J_1 = Q_2$$

✓

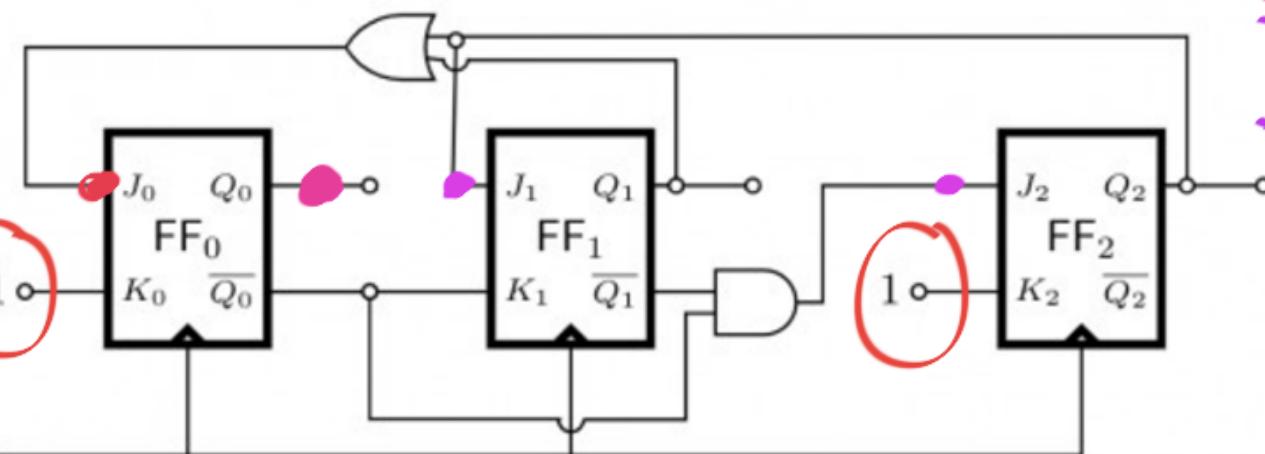
$$K_1 = \overline{Q}_0$$

✓

$$\overline{J}_2 = \overline{Q}_1, \overline{Q}_0$$

tests.gatecse.in

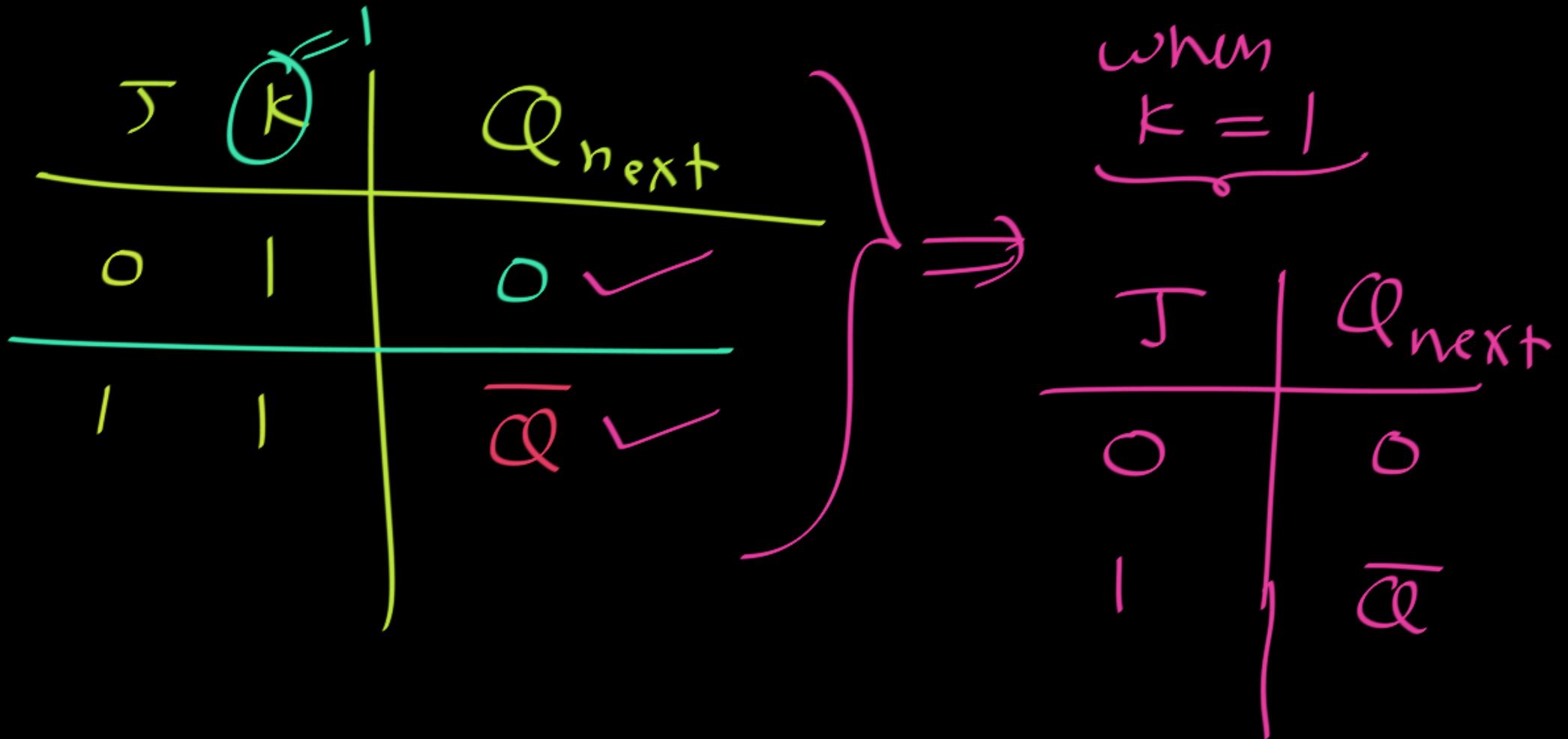
✓



Convention (msb)
tests.gatecse.in

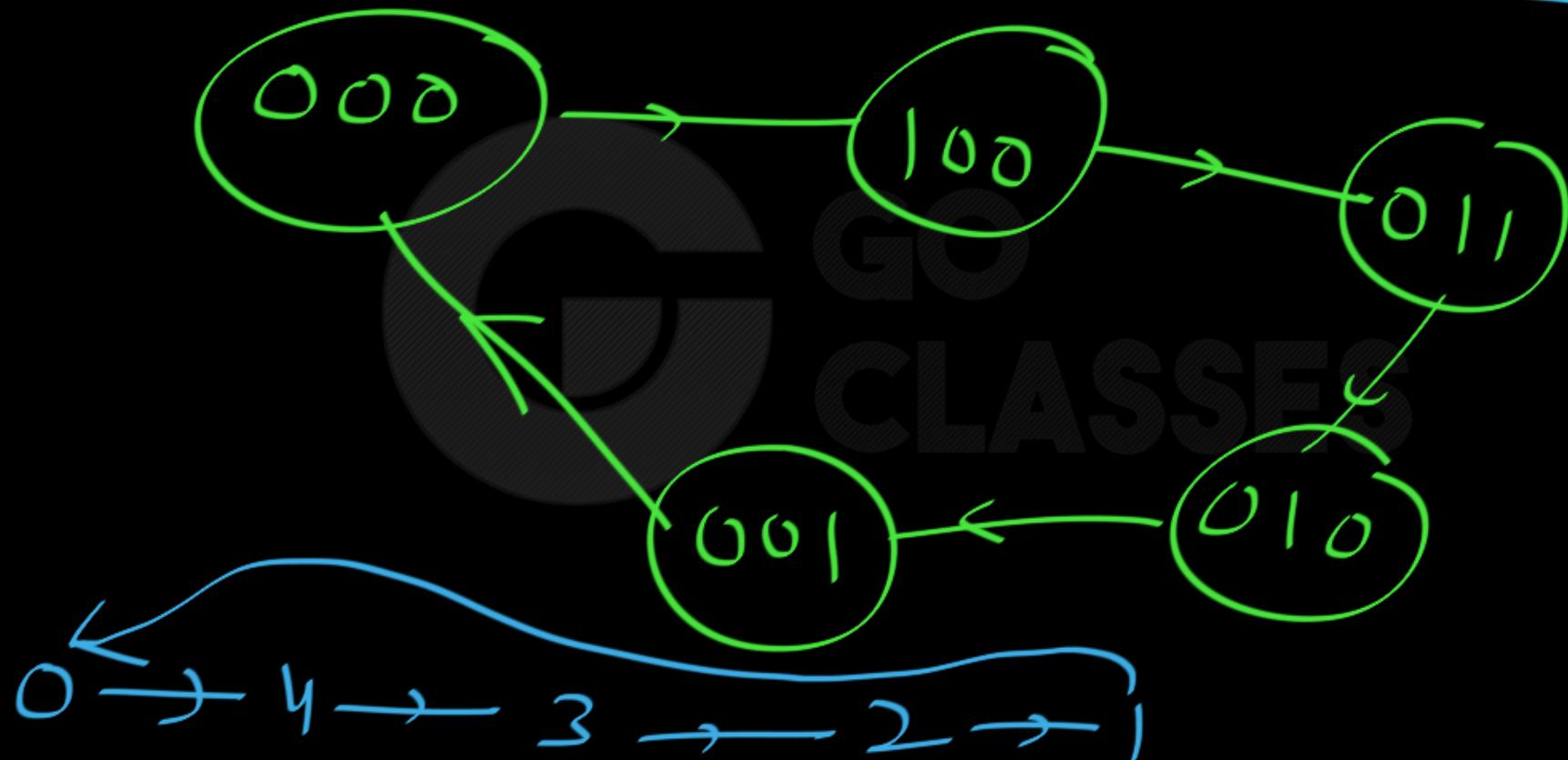
goclasses.in

tests.gatecse.in



Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	T_2	t_2	J_1	k_1	J_0	k_0
0	0	0	1	0	0	1	✓	0	1	0	1
0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	1	1	0	1	0	0	0	1	1	1	1
1	0	0	0	1	1	1	✓	1	1	1	1
1	0	1	0	1	0	0	0	1	0	1	0
1	1	0	0	0	1	0	0	1	1	1	1
1	1	1	0	1	0	0	0	1	0	1	0

State Diagram:

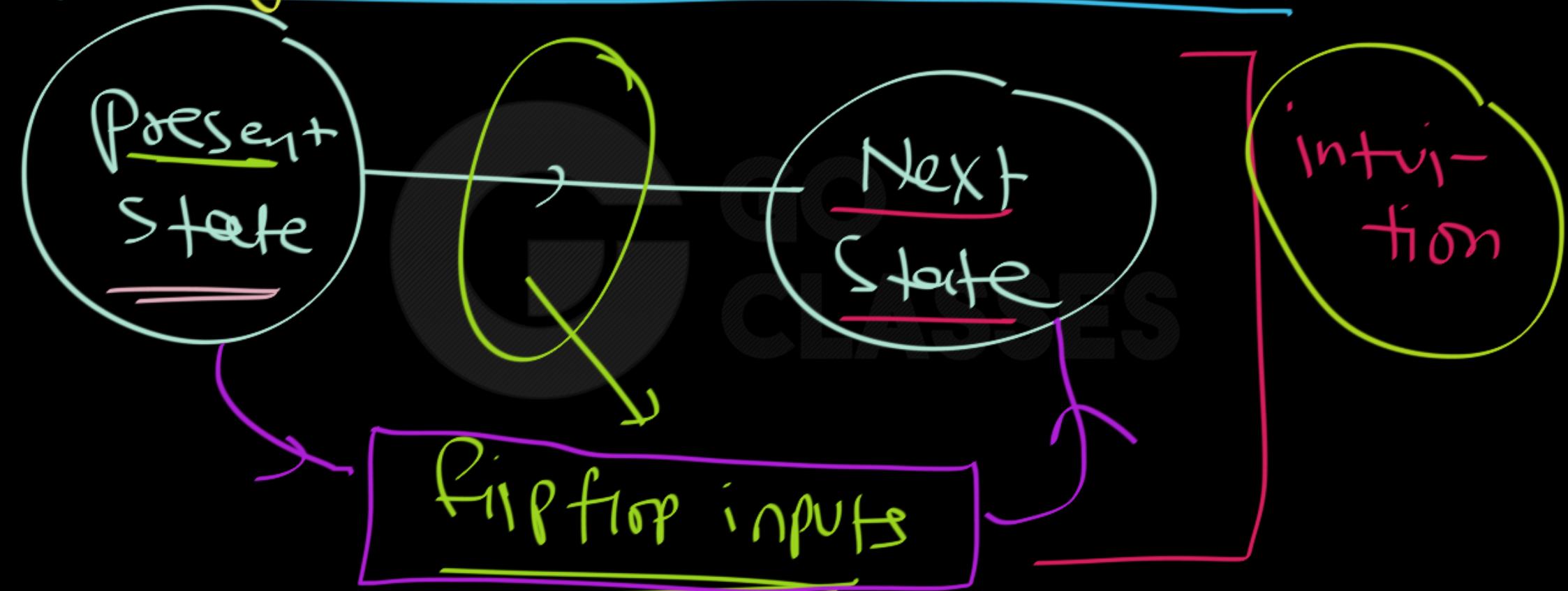


m₀₂ 5 Counter

Analysis of Any Synchronous CKT :

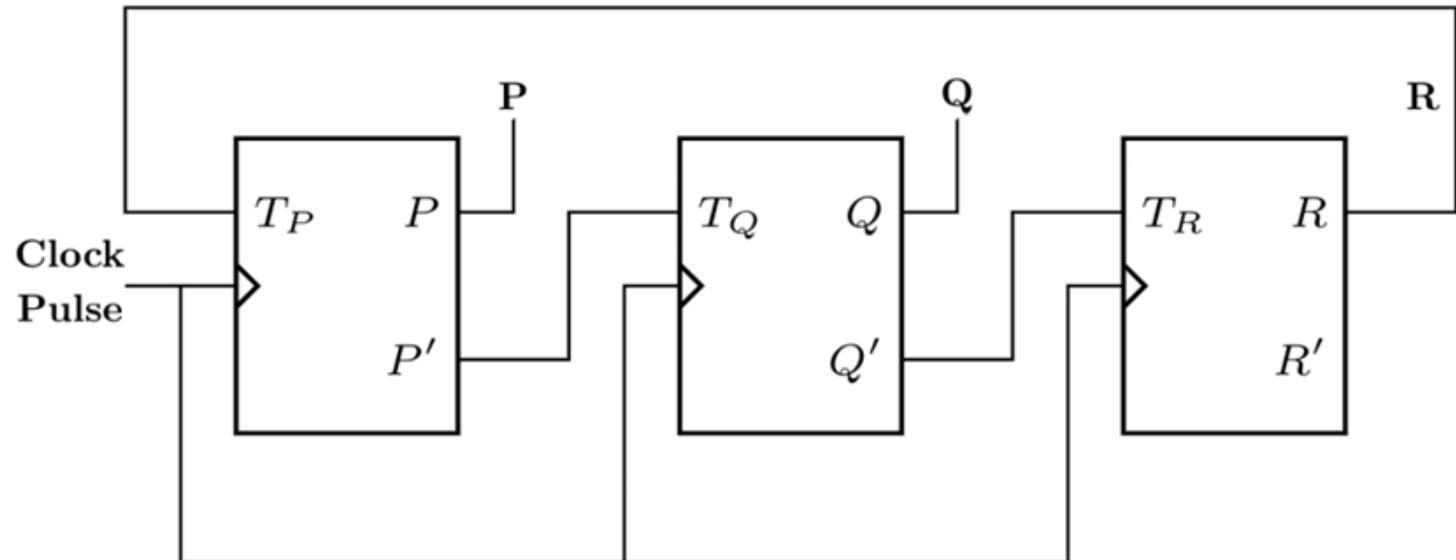
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	flip flop inputs $T_i = f(\text{Present States})$
Present States			Next State			$\left\{ \begin{array}{l} \text{we want} \\ \text{find this Next State} \end{array} \right.$

Analysis of Any Synchronous CKT :





Consider a 3-bit counter, designed using T flip-flops, as shown below:



tests.gatecse.in

Assuming the initial state of the counter given by PQR as 000, what are the next three states?

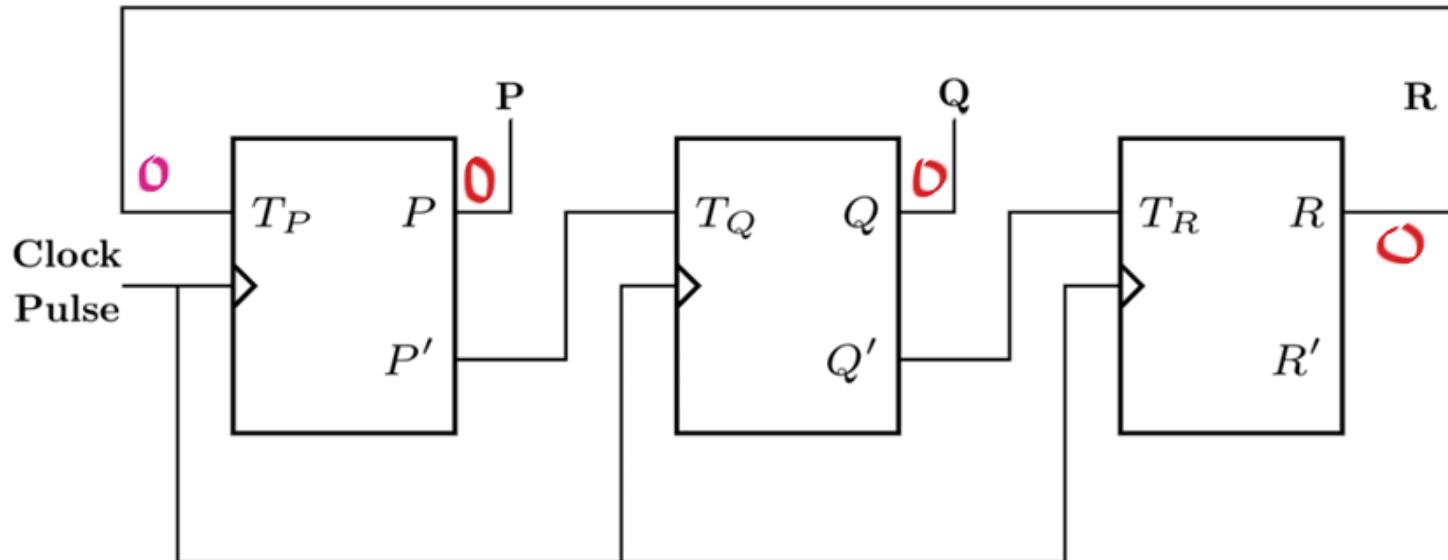
- A. 011, 101, 000
- B. 001, 010, 111
- C. 011, 101, 111
- D. 001, 010, 000

goclasses.in

tests.gatecse.in



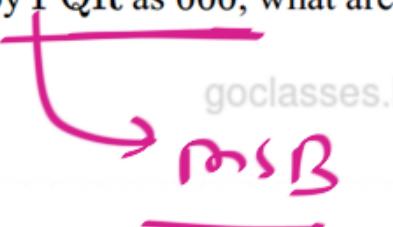
Consider a 3-bit counter, designed using T flip-flops, as shown below:



initial
P Q R
0 0 0

Assuming the initial state of the counter given by PQR as 000 , what are the next three states?

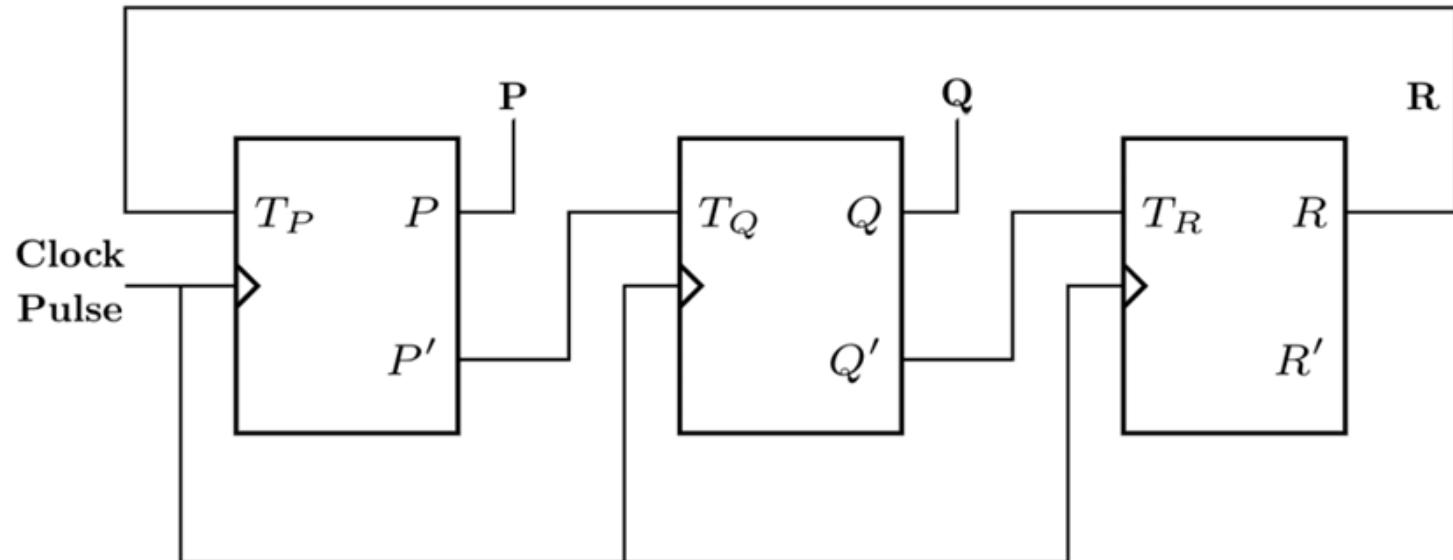
- A. 011, 101, 000
- B. 001, 010, 111
- C. 011, 101, 111
- D. 001, 010, 000

 goclasses.in
MSB

tests.gatecse.in



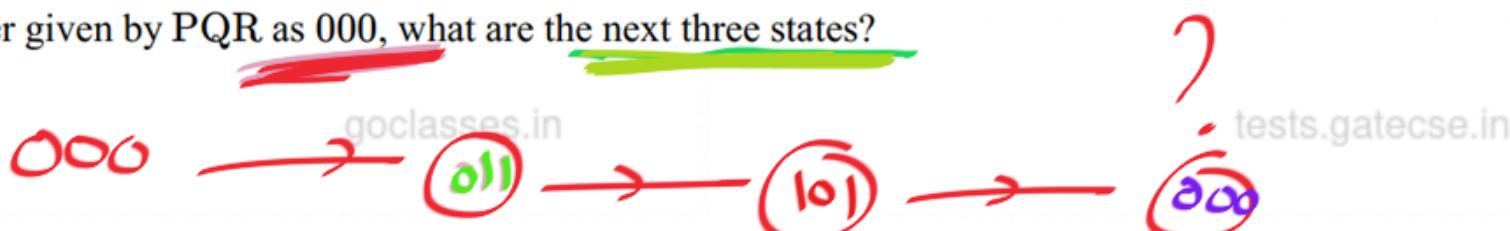
Consider a 3-bit counter, designed using T flip-flops, as shown below:



$$\left. \begin{aligned} T_P &= R \\ T_Q &= P' \\ T_R &= \bar{Q} \end{aligned} \right\} \text{tests.gatecse.in}$$

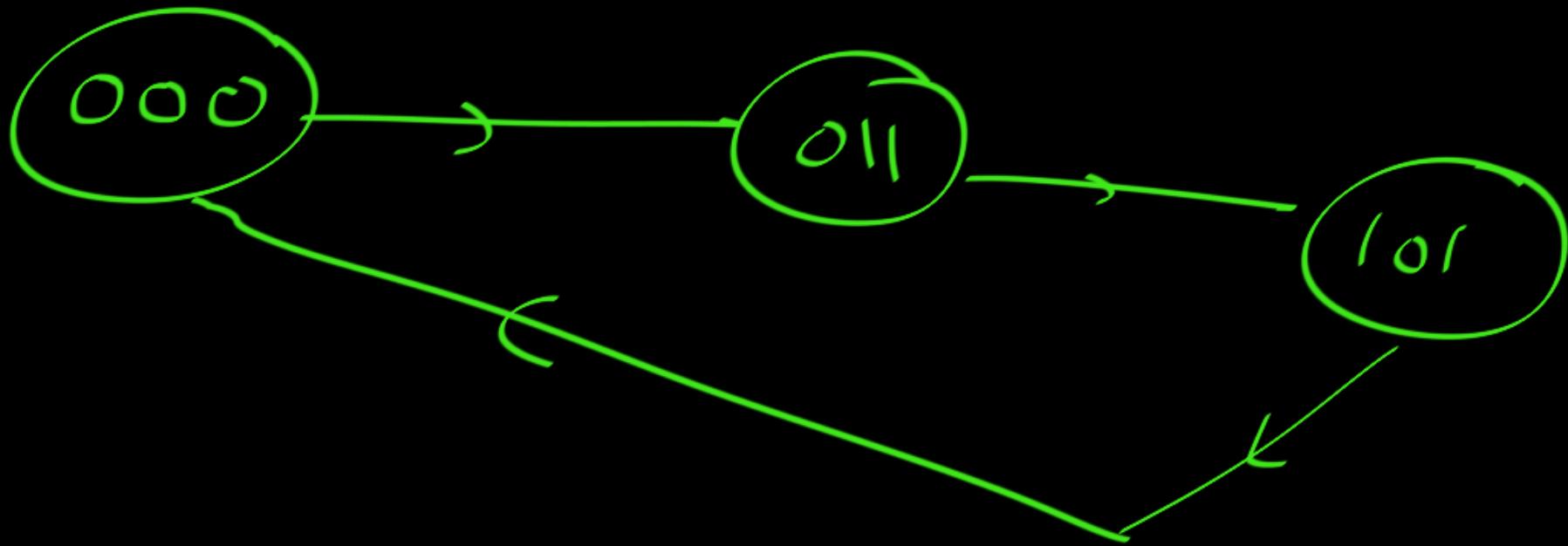
Assuming the initial state of the counter given by PQR as 000, what are the next three states?

- ✓ A. 011, 101, 000
 B. 001, 010, 111
 C. 011, 101, 111
 D. 001, 010, 000



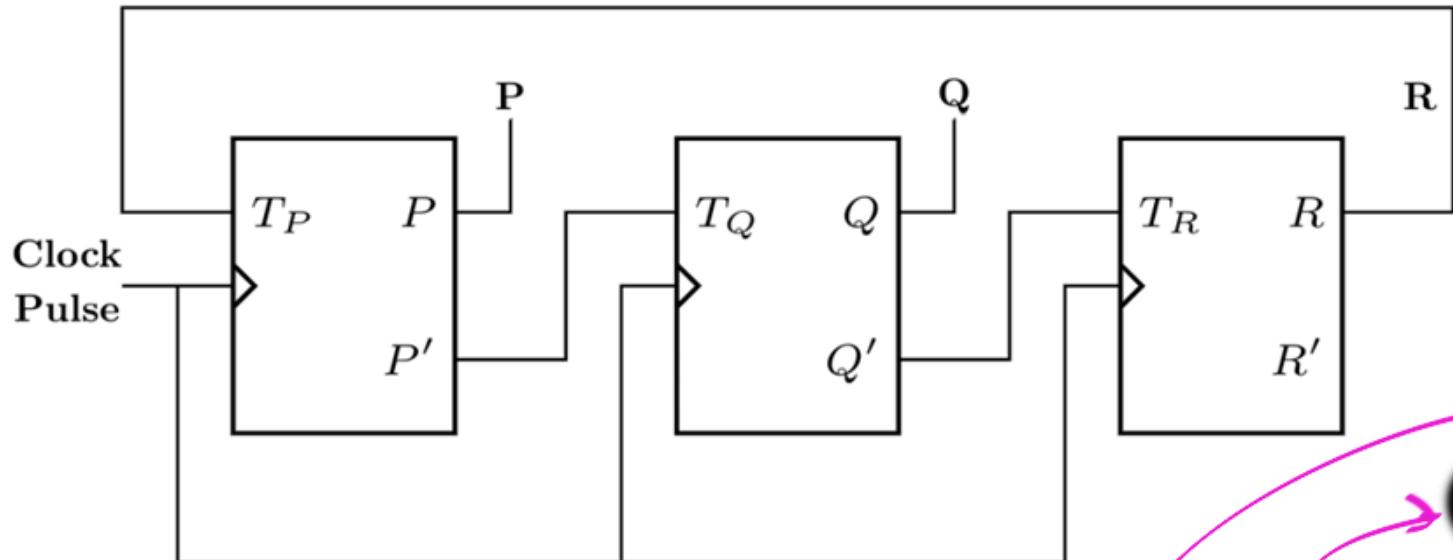
Analysis 2: (Restricted) → focus on what you want

P	Q	R	P^+	Q^+	R^+	T_P	T_Q	T_R
0	0	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	1	0	0	0	1	0	1





Consider a 3-bit counter, designed using T flip-flops, as shown below:



tests.gatecse.in
Complete Analysis

Assuming the initial state of the counter given by PQR as 000, what are the next states?

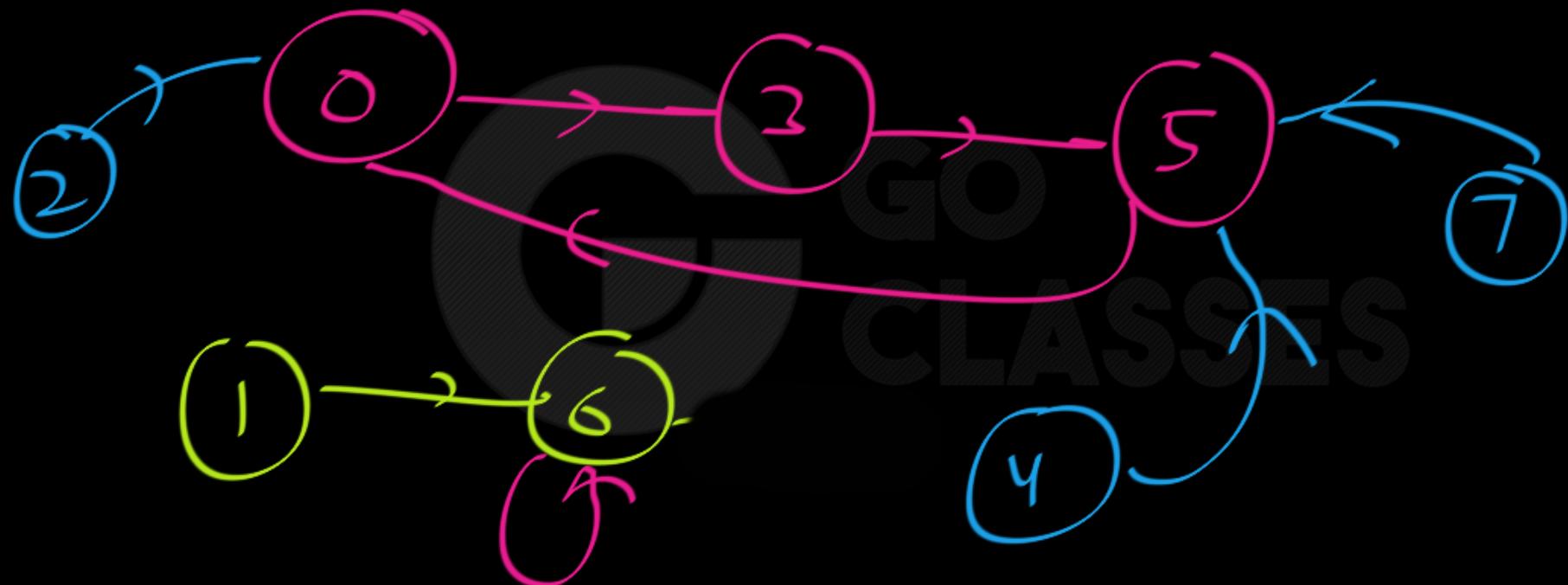
- A. 011, 101, 000
- B. 001, 010, 111
- C. 011, 101, 111
- D. 001, 010, 000

Analysis 2 : (Complete Analysis)

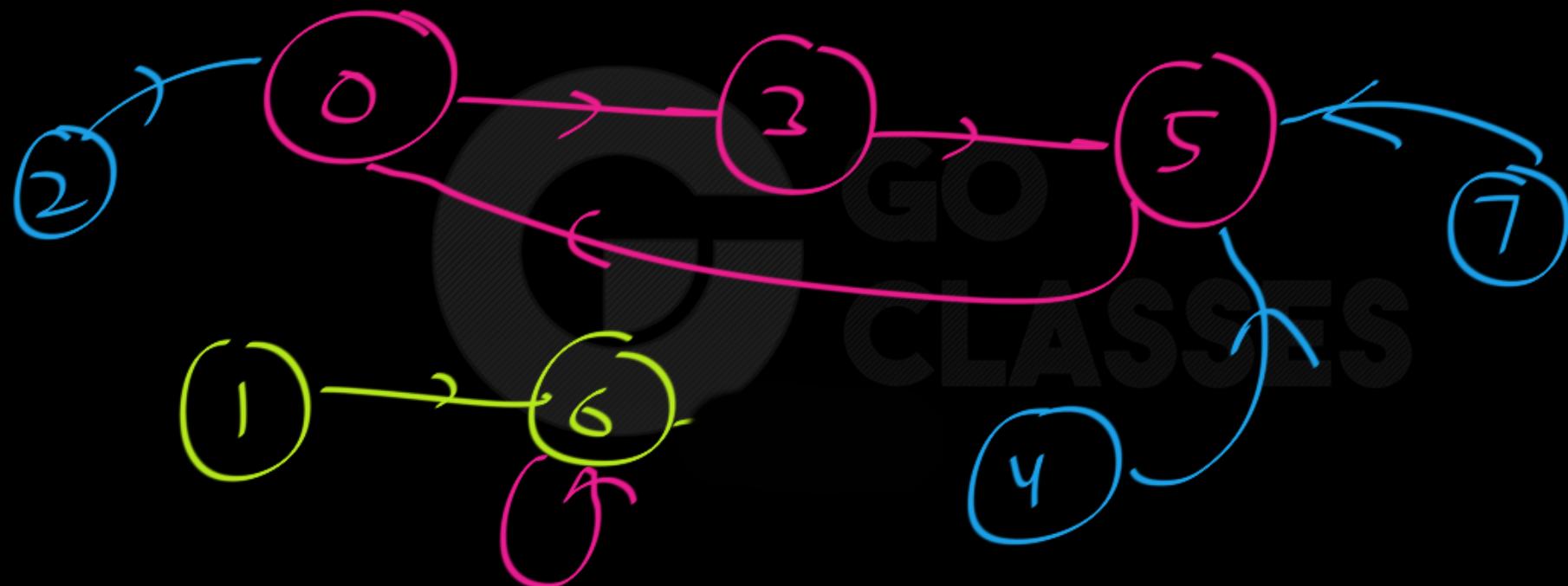
P	Q	R	P^+	Q^+	R^+	T_P	T_Q	\bar{P}	\bar{Q}	\bar{R}
0	0	0	0	1	1					
0	0	1	1	1	0					
0	1	0	0	0	0					
0	1	1	1	0	1					
<u> </u>			1	0	1					
1	0	0	1	0	1	0	0	0	0	1
1	0	1	0	0	0	1	0	0	1	1
1	1	0	0	1	0	0	1	0	0	0
<u> </u>			0	1	0					



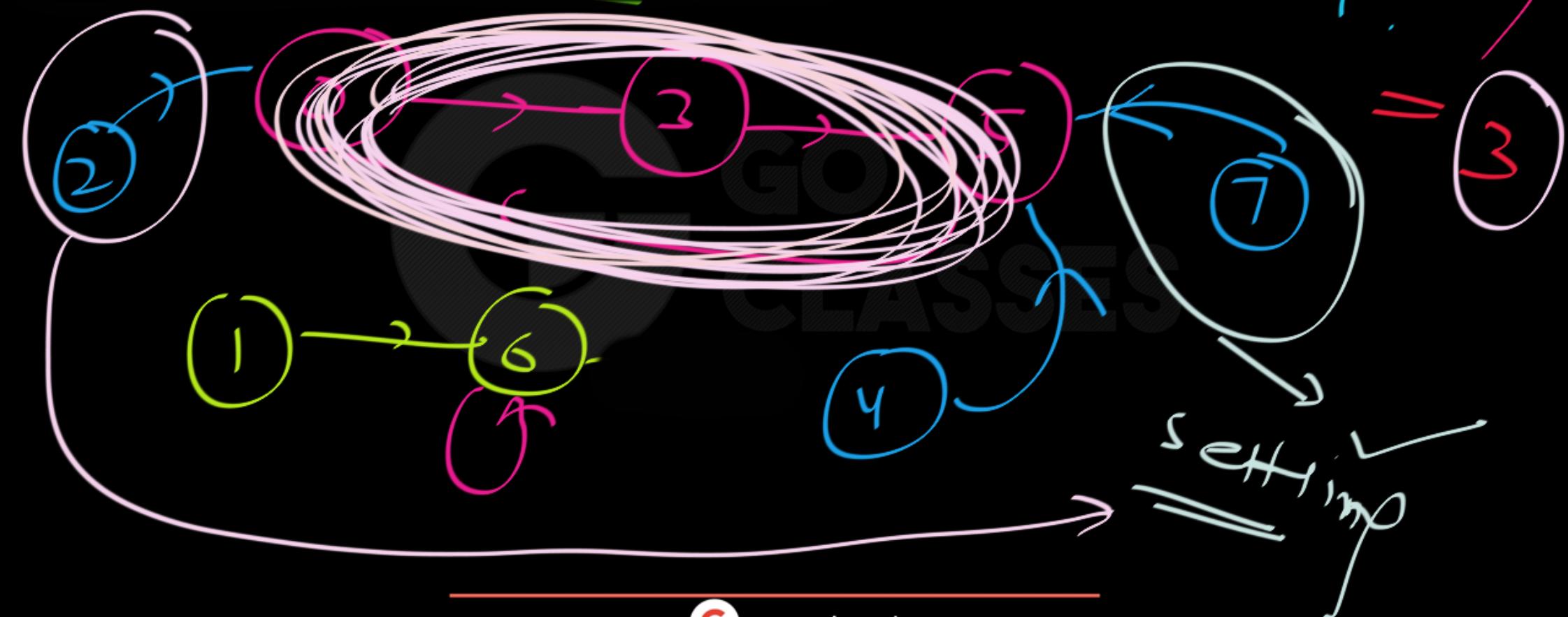
State Diagram:



Modulus of this Counter ? $\Rightarrow 3 \checkmark$



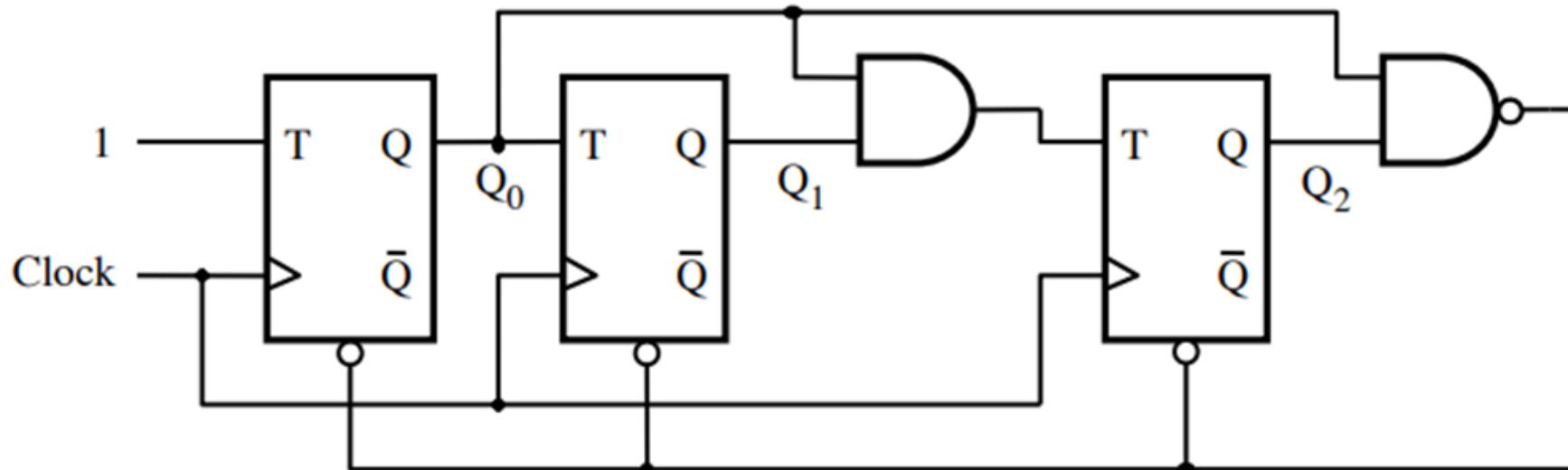
Magnitude of this Counter (if initial state was 7):



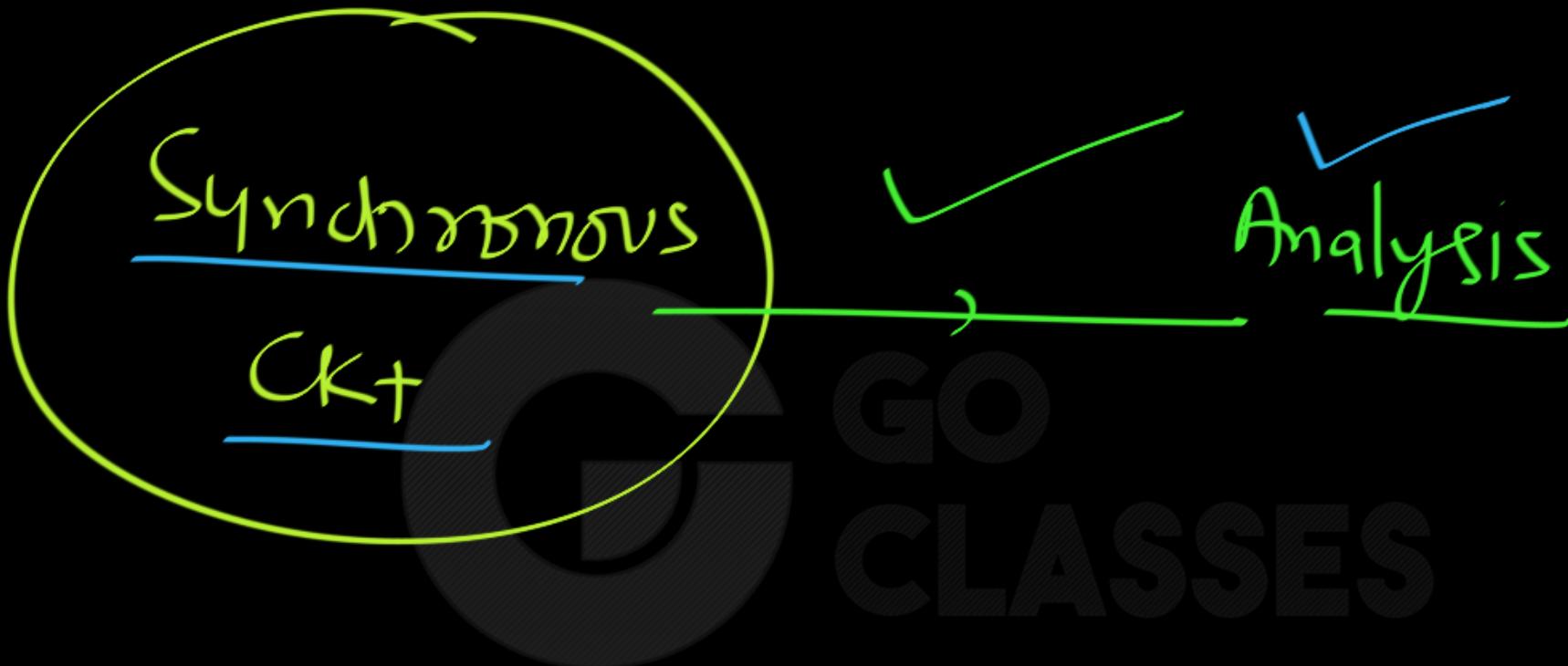


Q: Complete Analyse the given synchronous sequential circuit:

HW ✓



(a) Circuit





Now:





Reverse Engineering of Analysis

= Design



GO
CLASSES



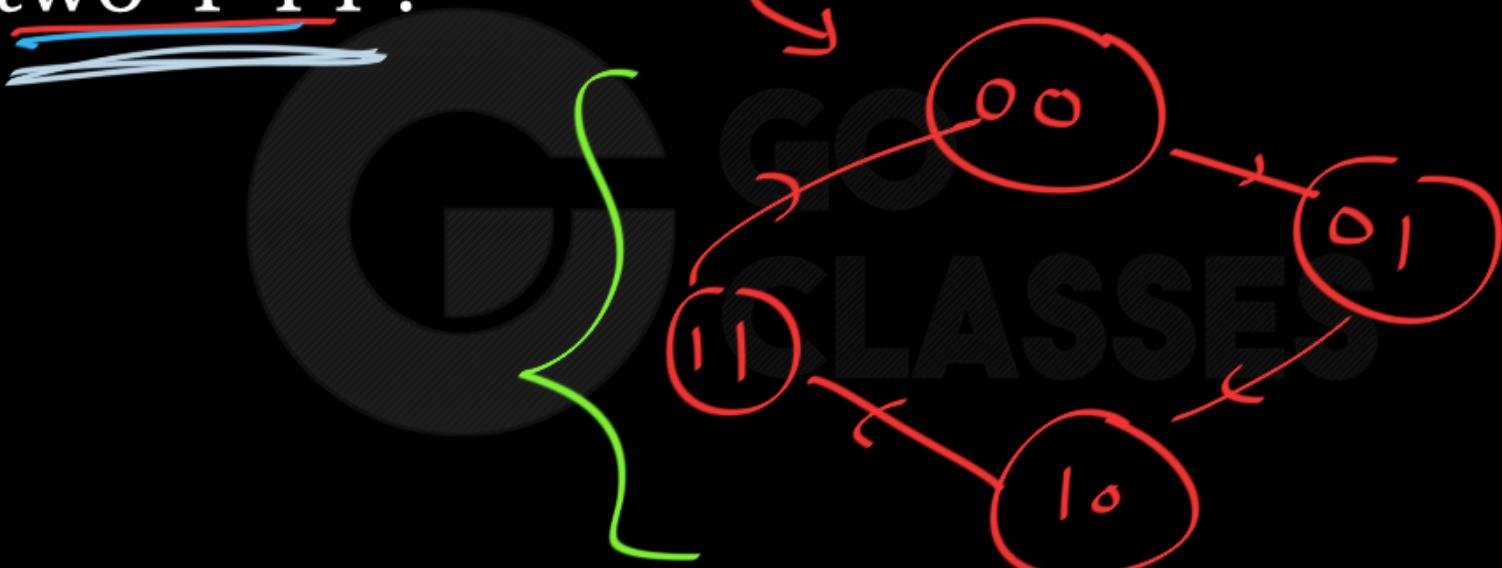
Next Topic:

Designing of Synchronous Sequential Circuits (When NO External Inputs)

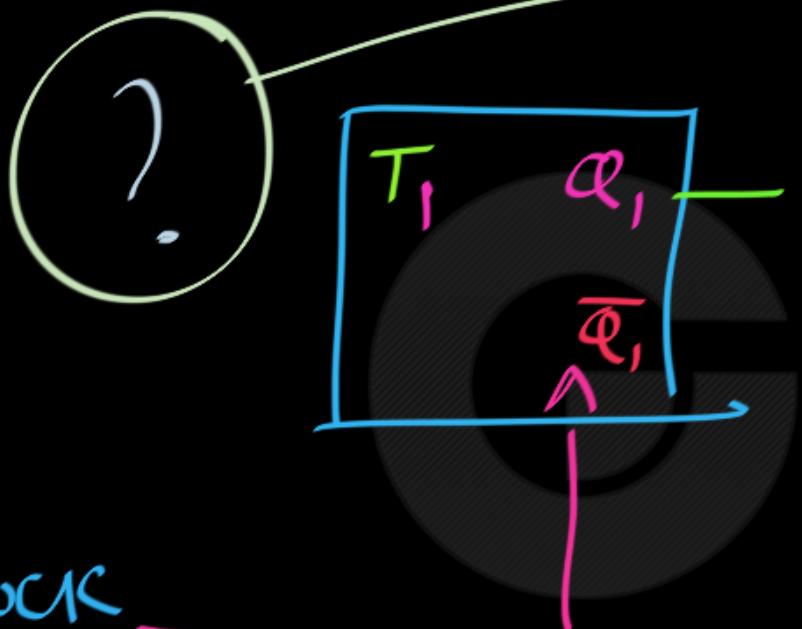


Q:

Design a 2-bit Binary (up) synchronous counter using two T-FF.



Sketch of Counter:



Combinational Circuit



Clock

To find: $T_1 = f(\text{Present State})$; $T_0 = f(\text{Present State})$

Present state

Next state

flip flop inputs

Q_1 Q_0

Q_1^+ Q_0^+

T_1 T_0

0	0
0	1
1	0
1	1

0 → 0

1 → 1

1 → 0

0 → 0

0 → 1

1 → 0

0 → 1

0 → 0

} , } , }

Present state

Next state

flip flop inputs

Q_1

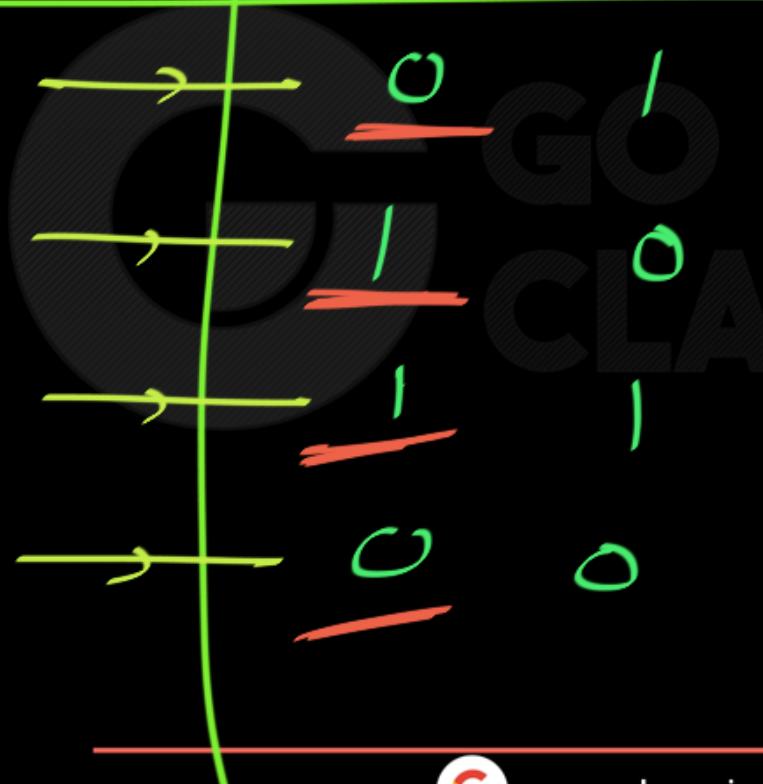
Q_0

Q_1^+ Q_0^+

T_1

T_0

<u>0</u>	0
<u>0</u>	1
1	0
1	1



0

1

0

1

Present state

Q_1

Q_0

Next state

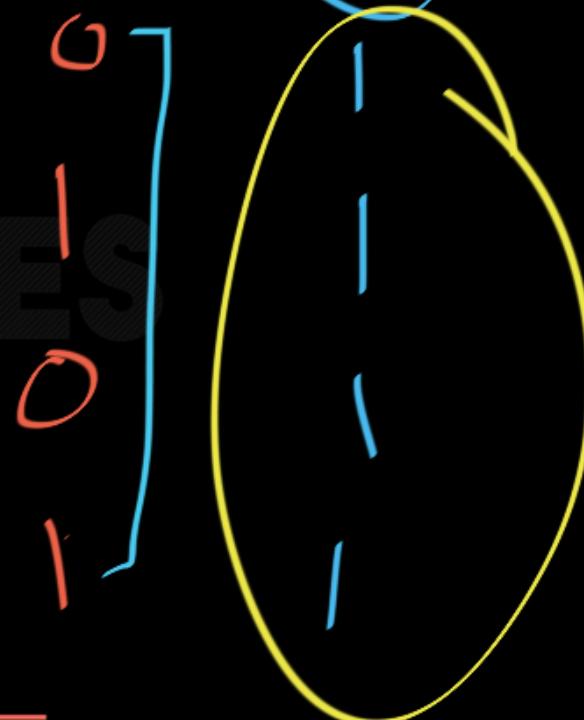
Q_1^+ Q_0^+

flip flop inputs

T_1

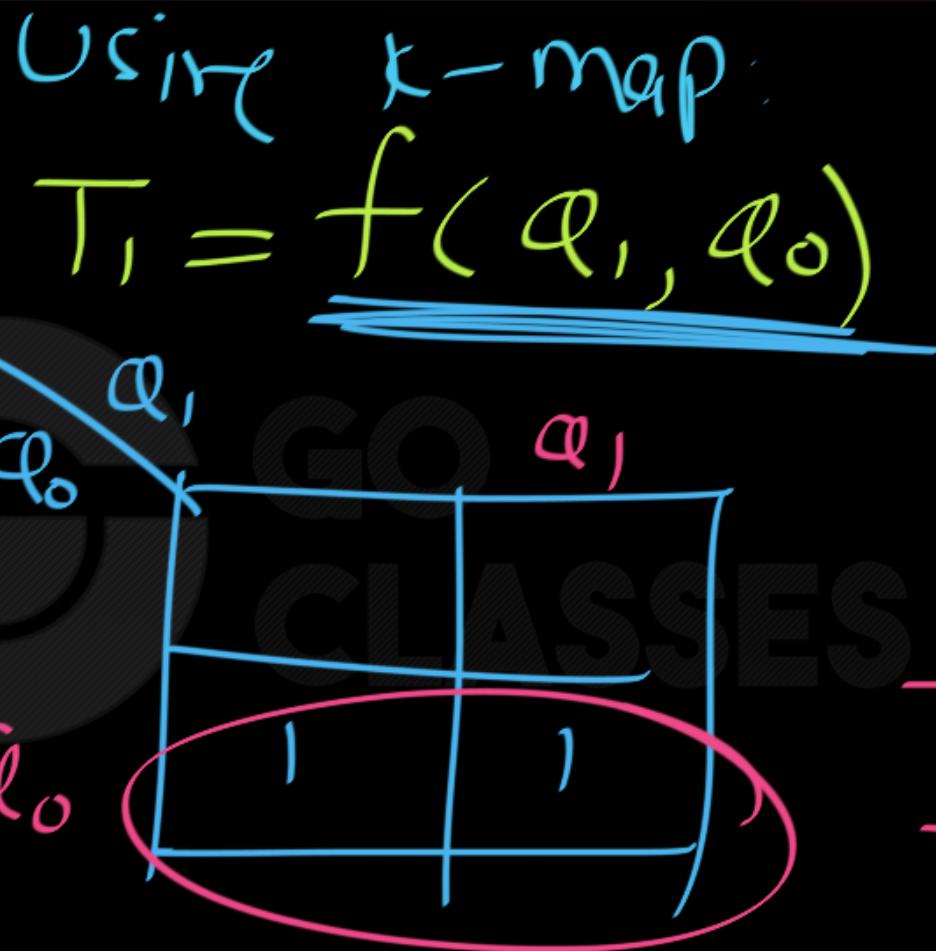
T_0

Q_1	Q_0	Q_1^+	Q_0^+
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

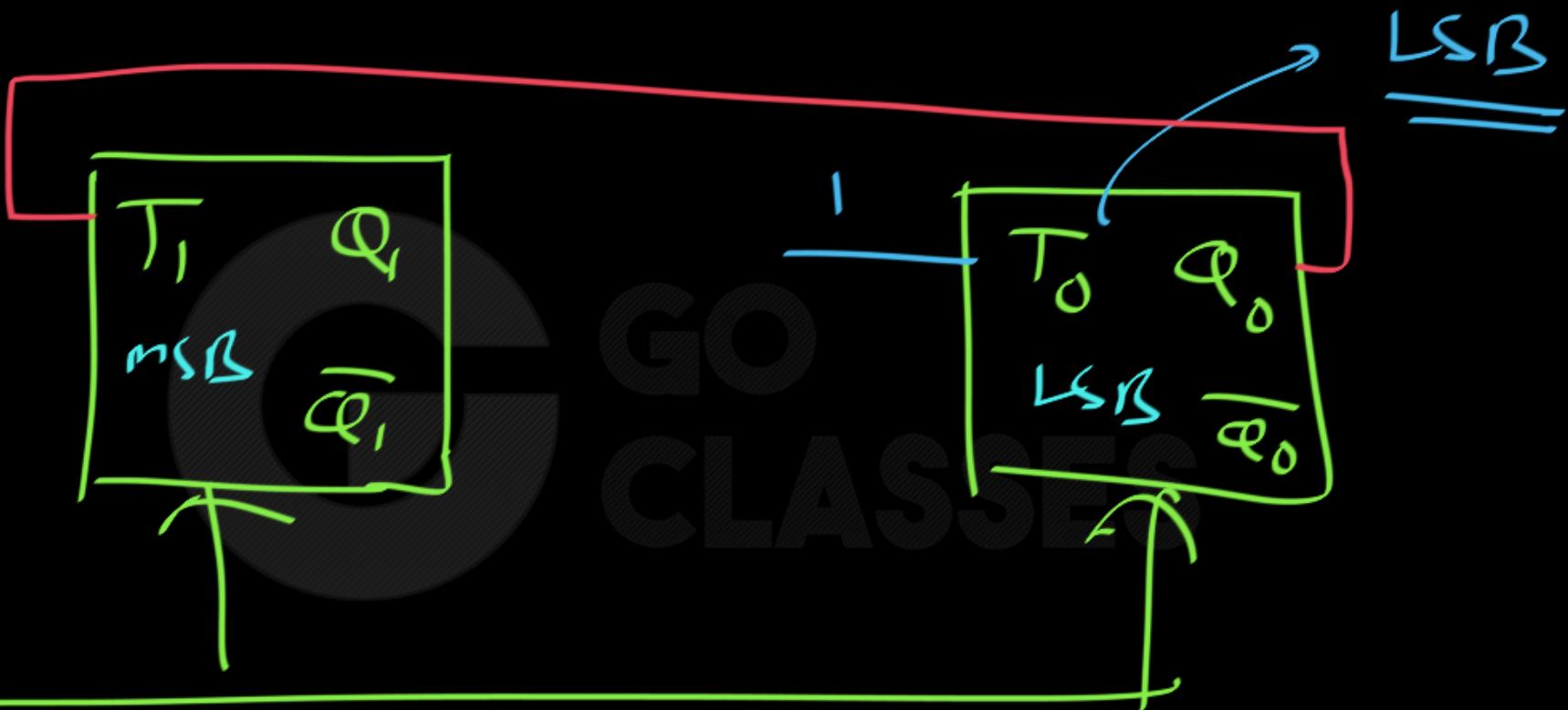


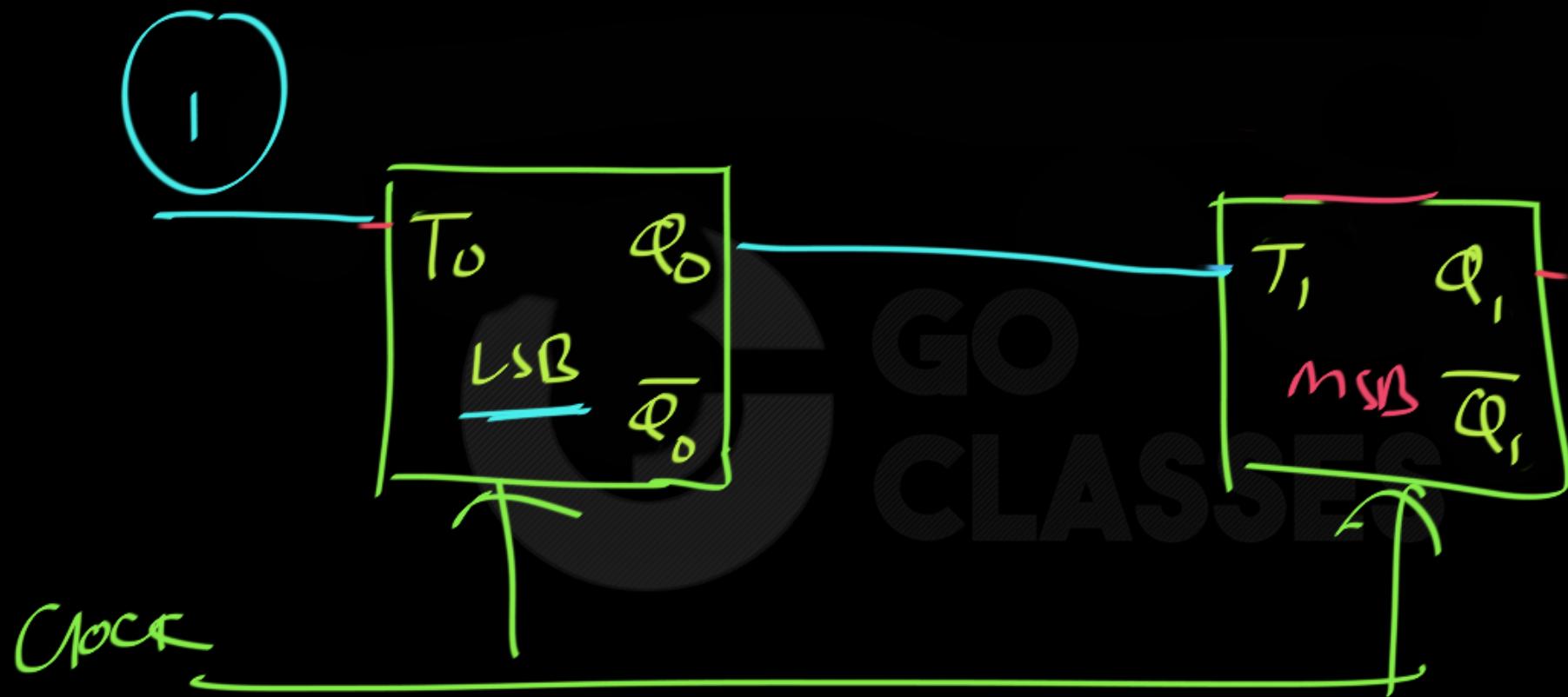


$$\begin{aligned}T_0 &= 1 \\T_1 &= \overline{Q_0} \\LSB(T_0) &= 1 \\MSB(T_1) &= \overline{Q_0}\end{aligned}$$



$$T_1 = \overline{Q_0}$$

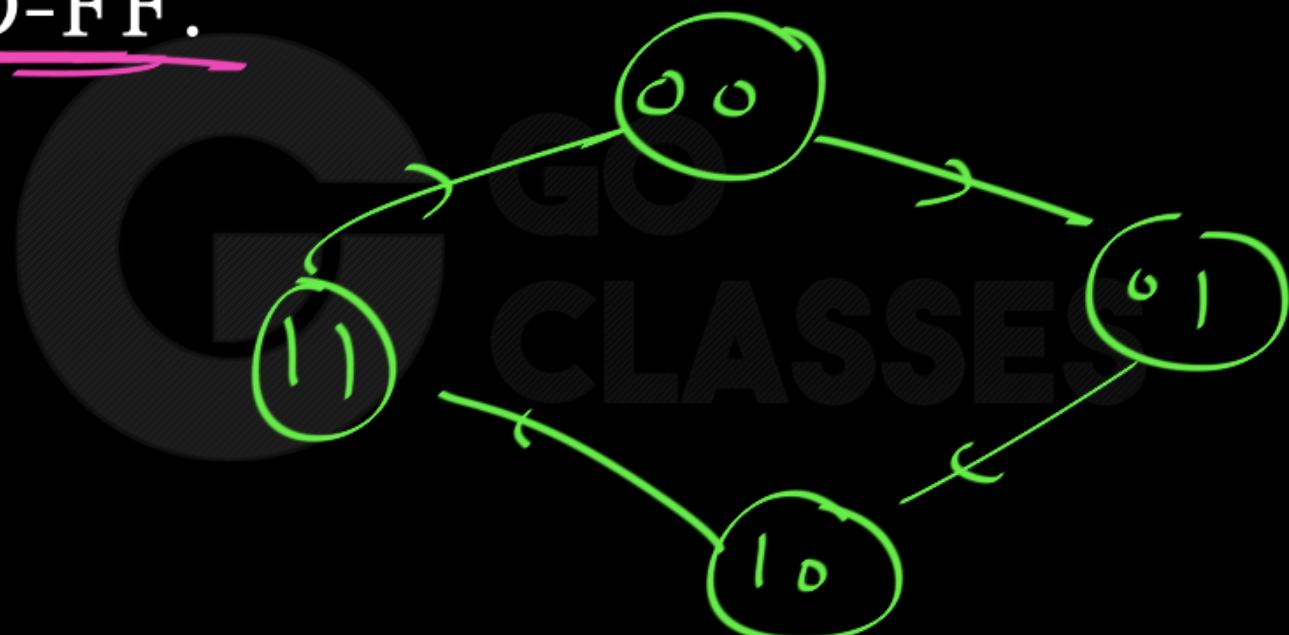




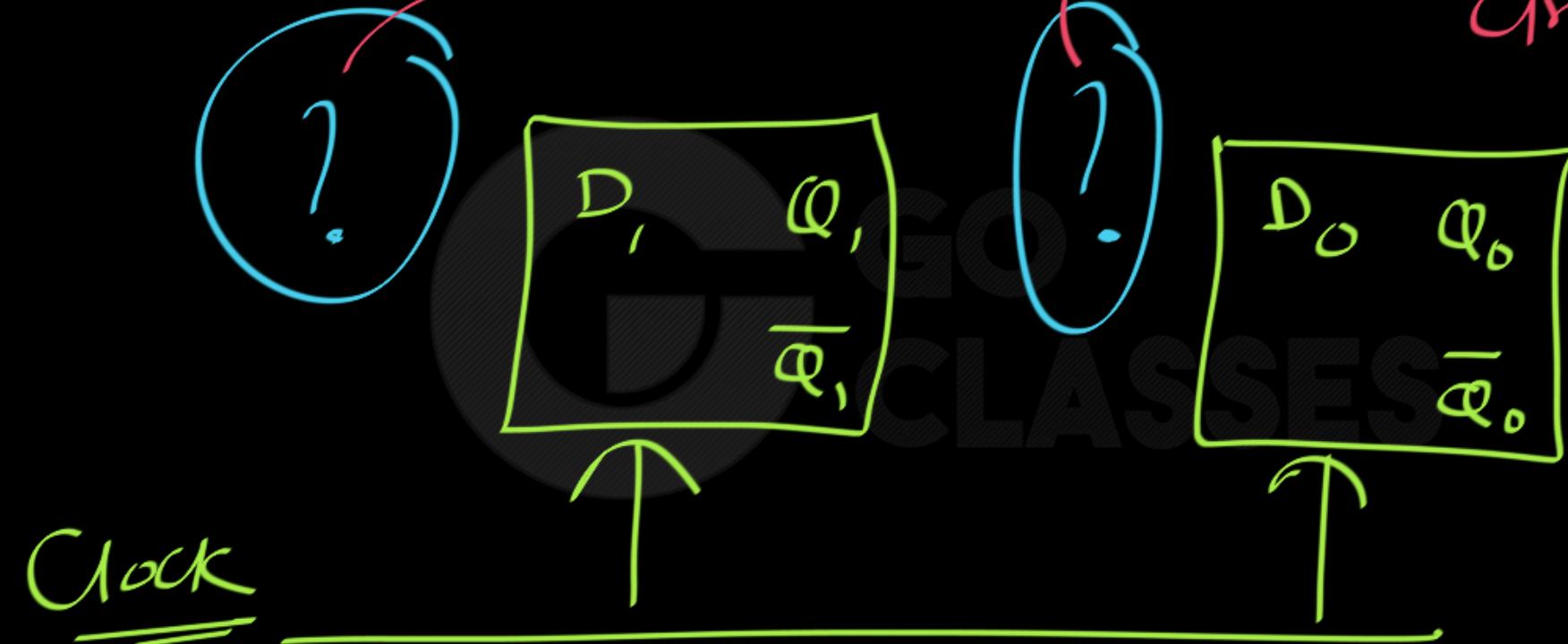


Q:

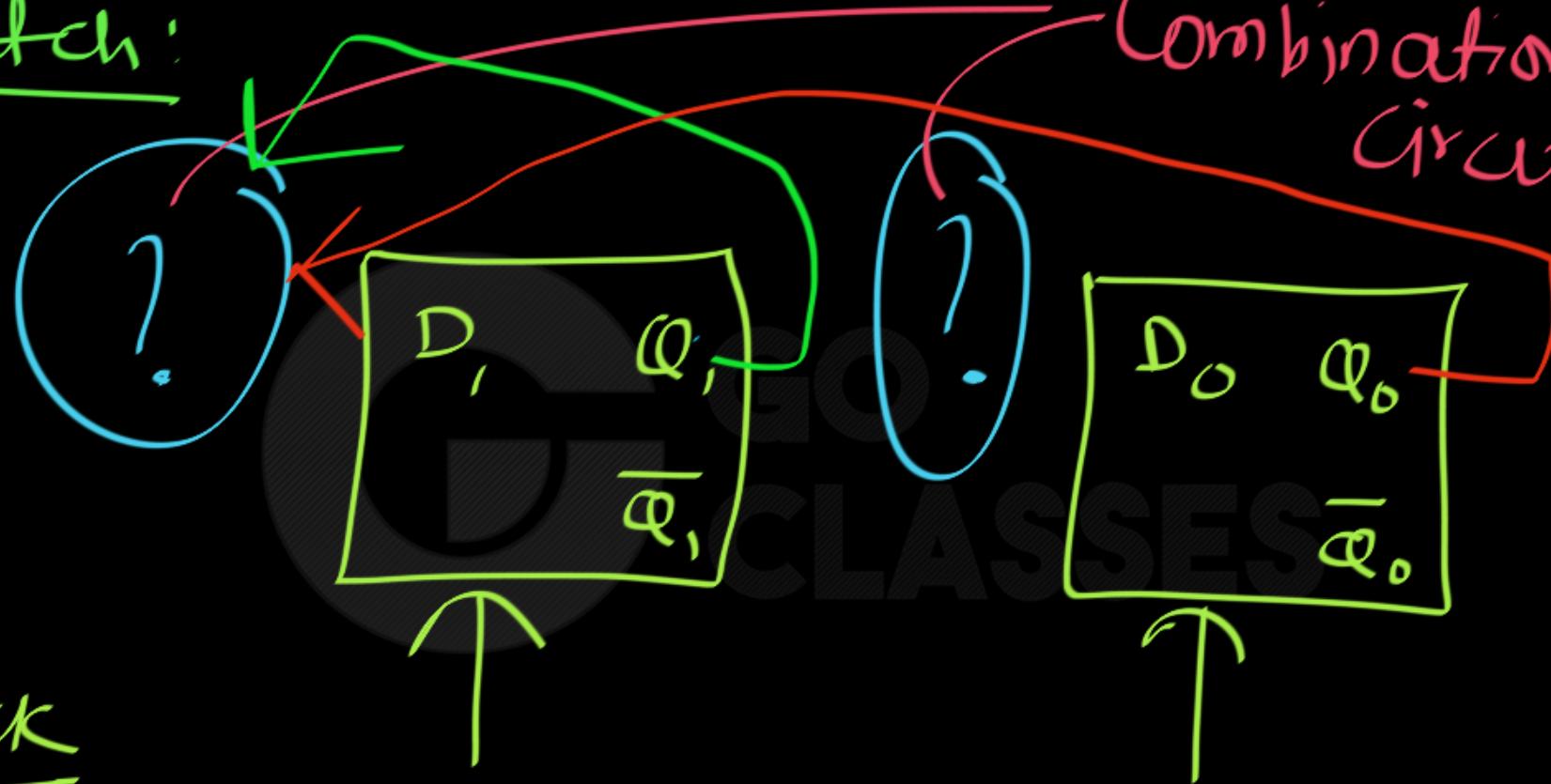
Design a 2-bit Binary (up) synchronous counter
using two D-FF.



Sketch:



Sketch:

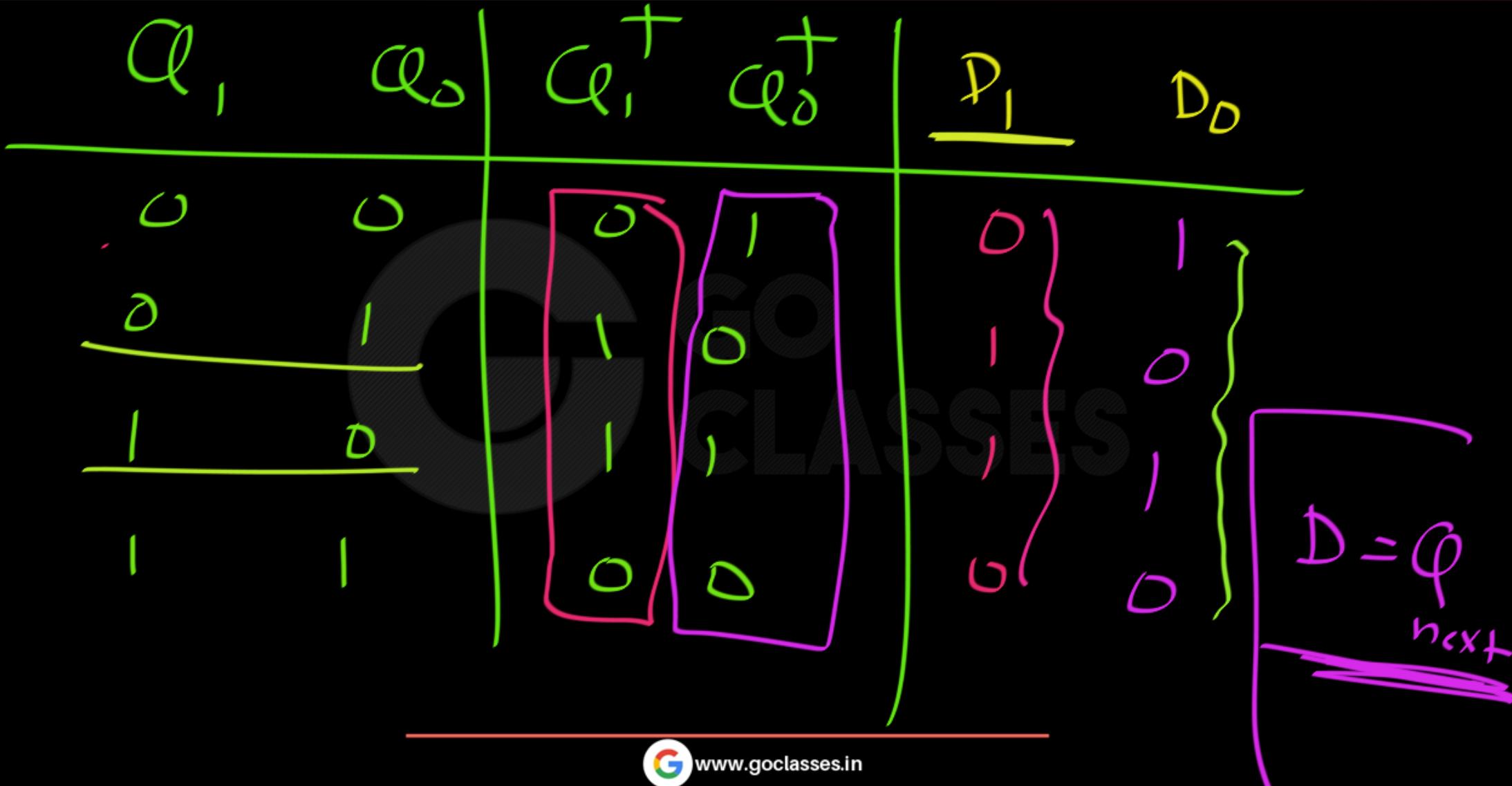


Clock

$$D_1 = f(Q_1, Q_0) ; D_0 = f(Q_1, Q_0)$$

Q_1	Q_0	Q_1^+	Q_0^+	D_1	D_0
0	0	0	1		
0	1	1	0		
1	1	0	1		
1	0	0	0		

Excitation Table





$$D_1 = Q_0 \oplus Q_1$$

~~$$D_0 = \overline{Q_0}$$~~

K-map

Q_1

Q_0

0 1

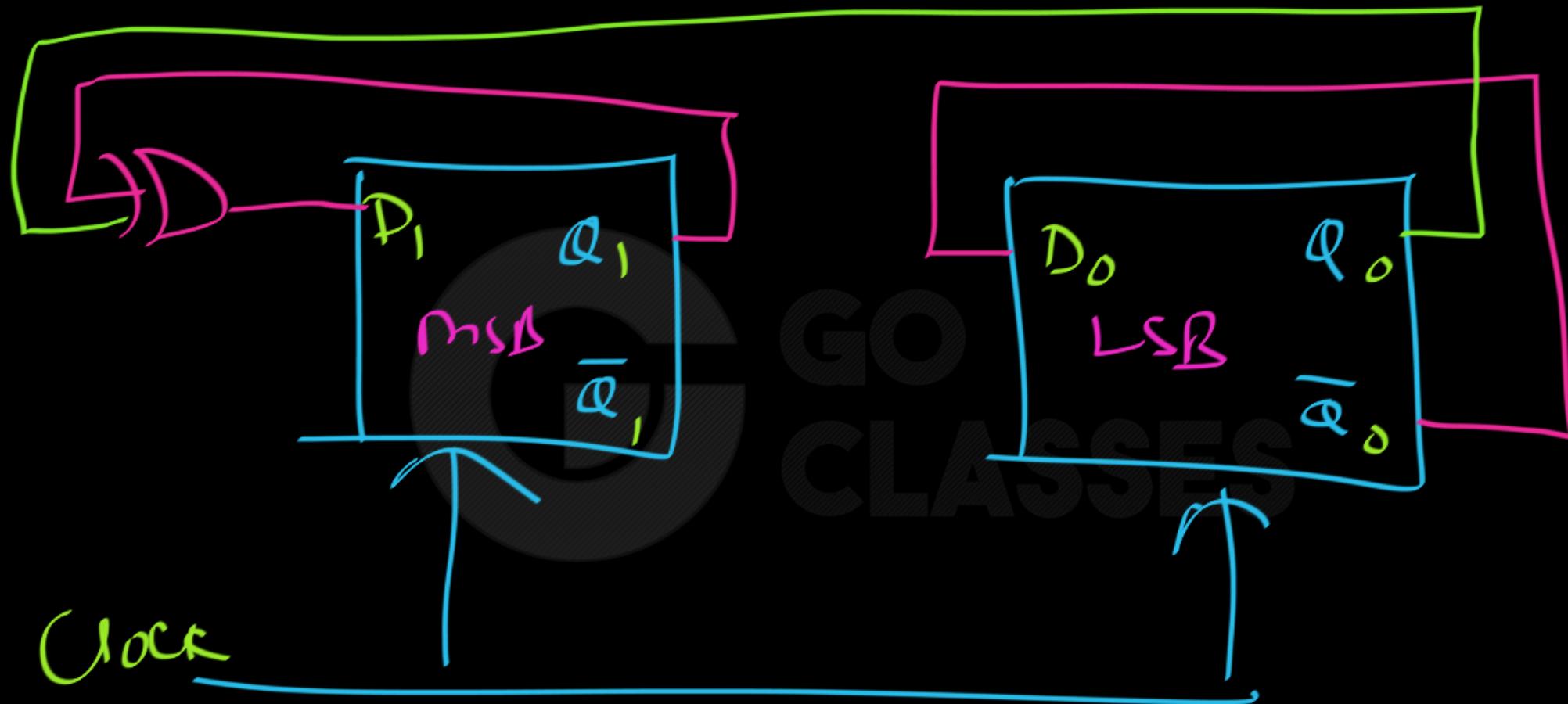
1 0

Q_1

D_1

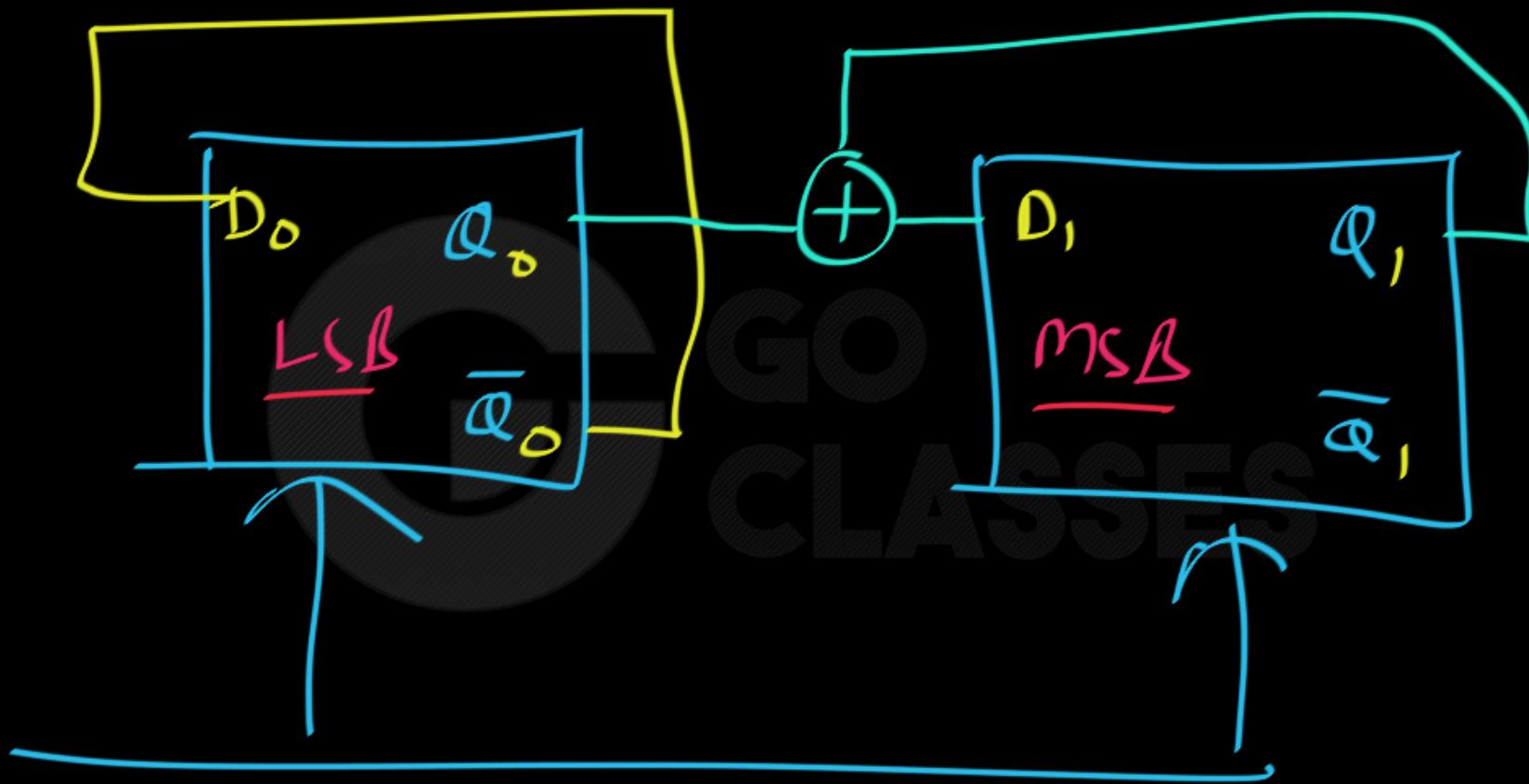
$$D_1 = \overline{Q_0} Q_1 + \overline{Q_1} Q_0$$

$$= Q_0 \oplus Q_1$$





Digital Logic

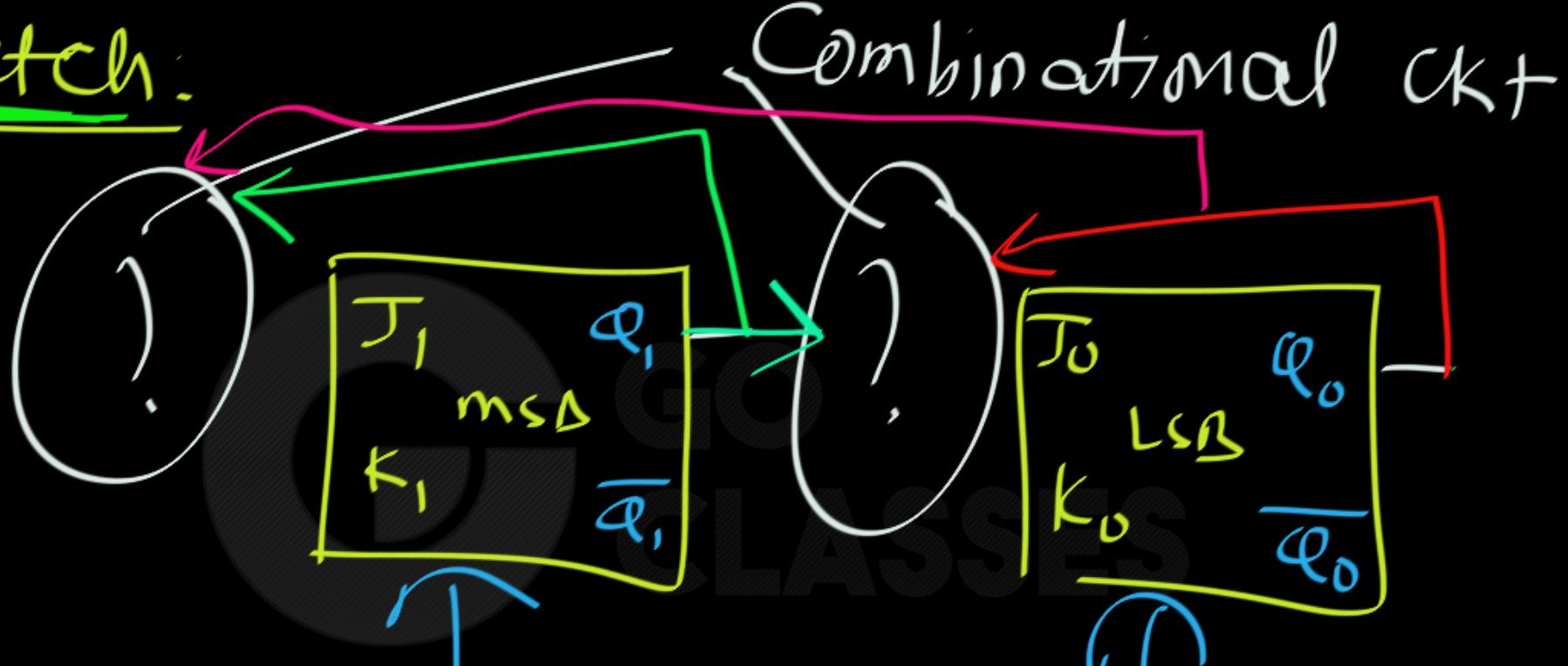




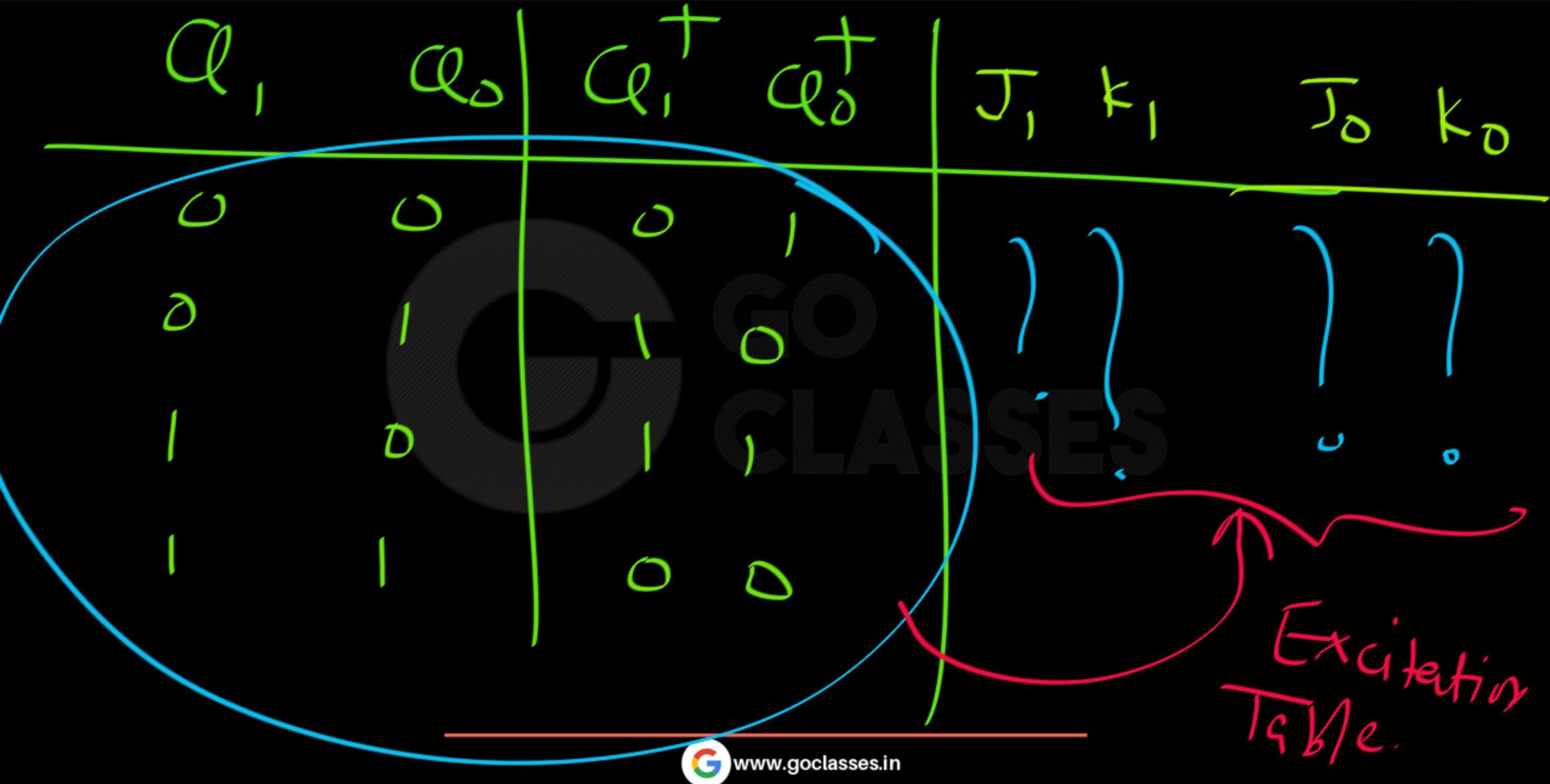
Q:

Design a 2-bit Binary (up) synchronous counter using two JK-FF.



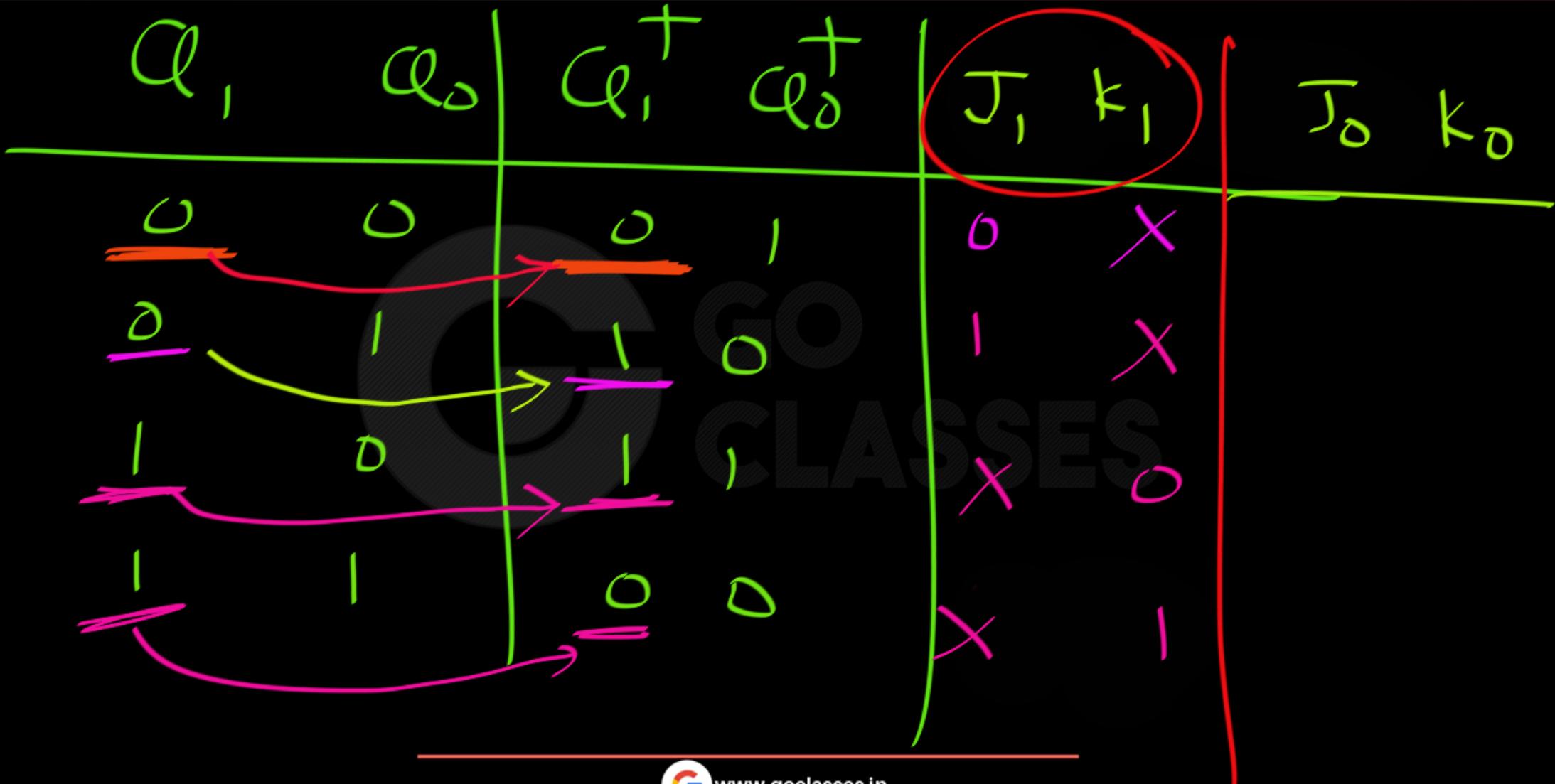
Sketch:clock

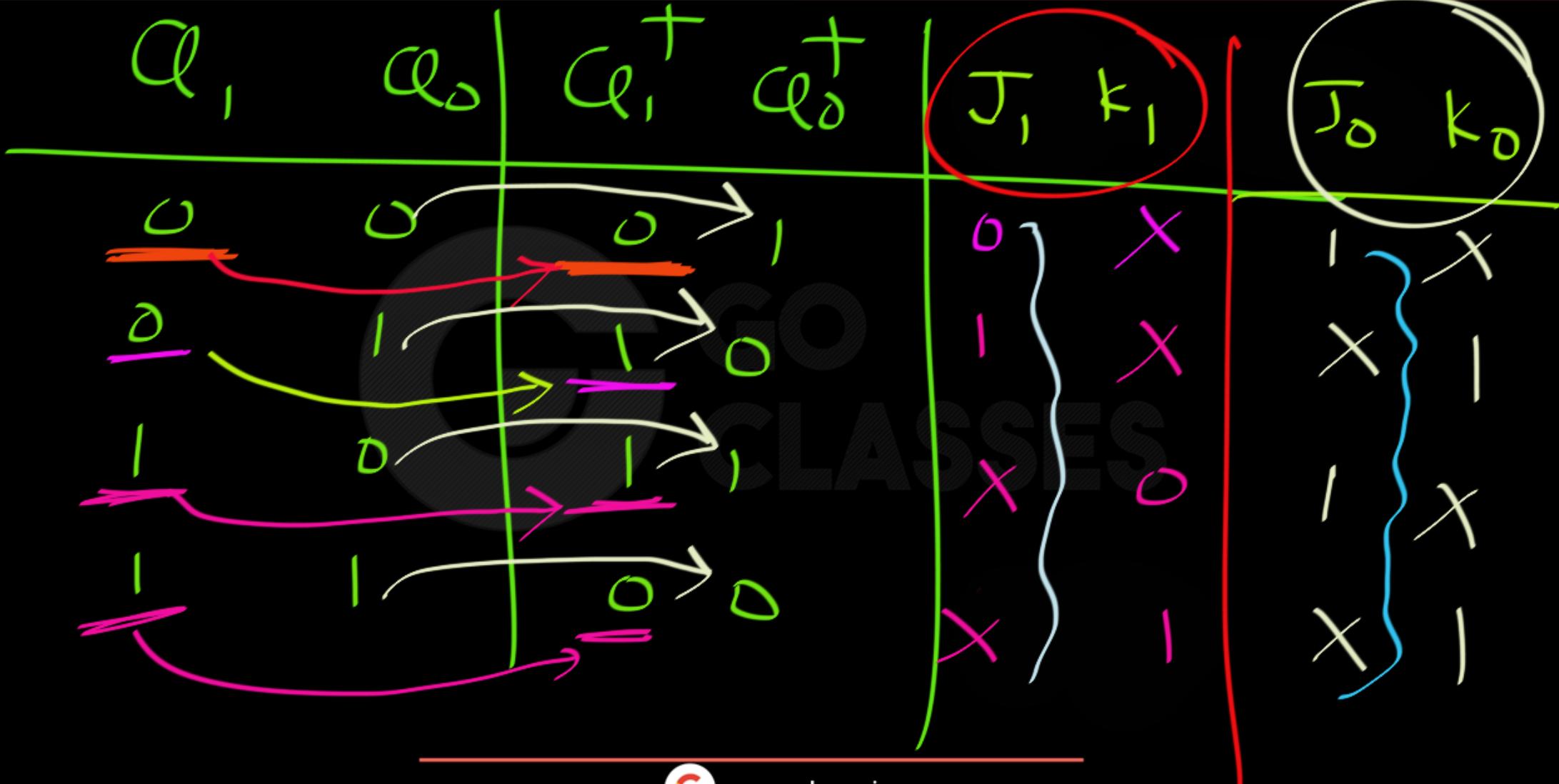
$$\frac{J_0, K_0, K_1, J_1}{f(Q_0, Q_1)} = f(Q_0, Q_1)$$



Execution Time of JK

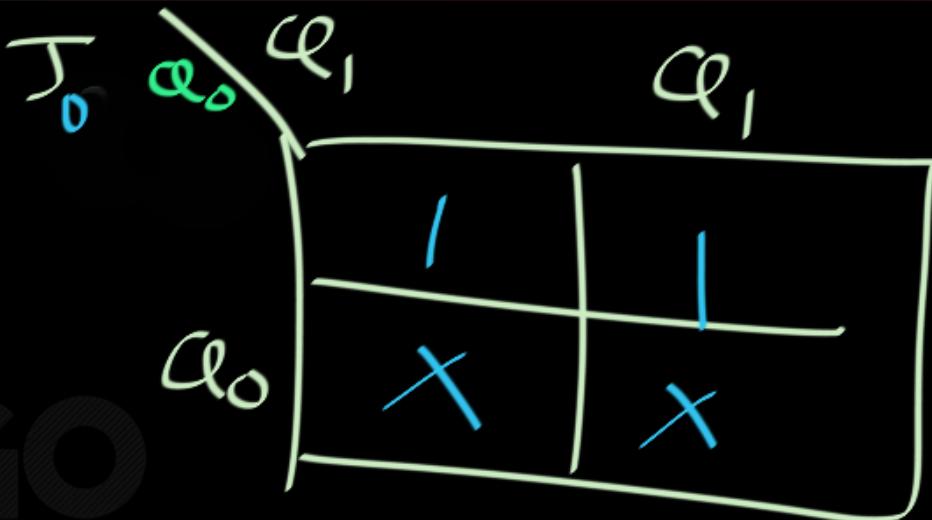
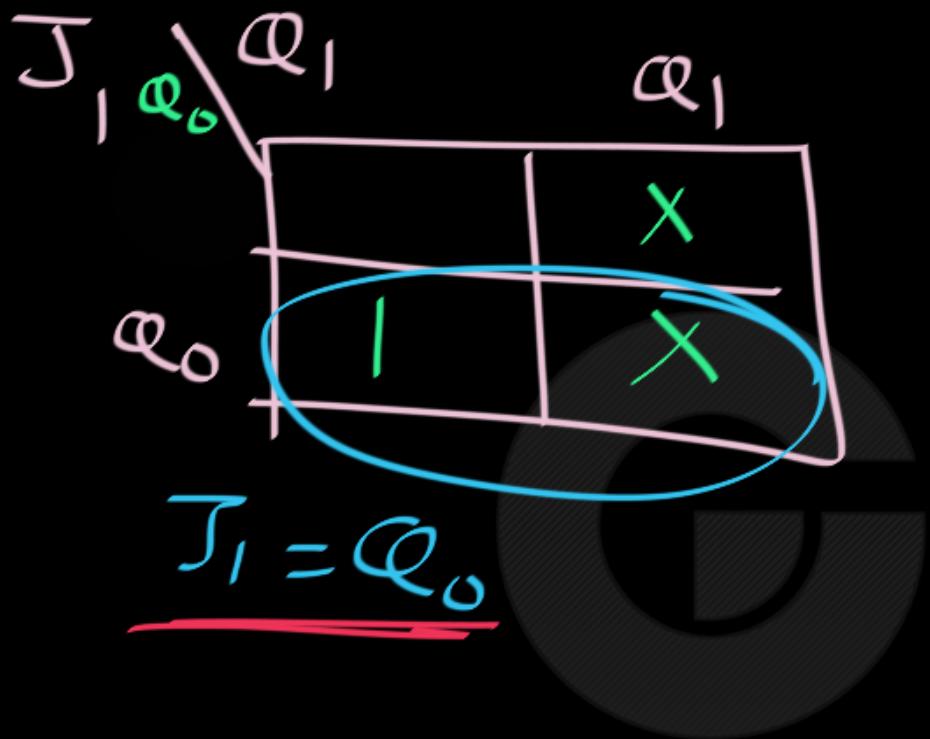
Q	Q_n	J	K
0 → 0		0	X
0 → 1		1	X
1 → 0		X	1
1 → 1		X	0





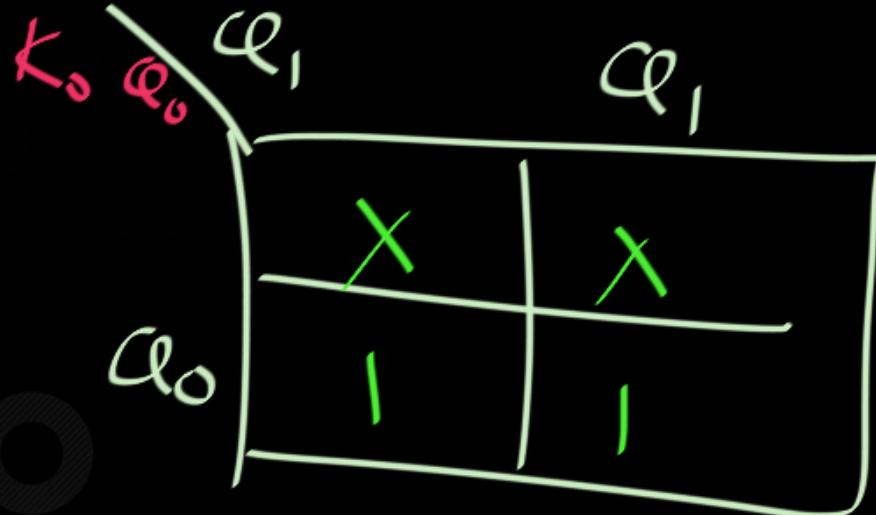
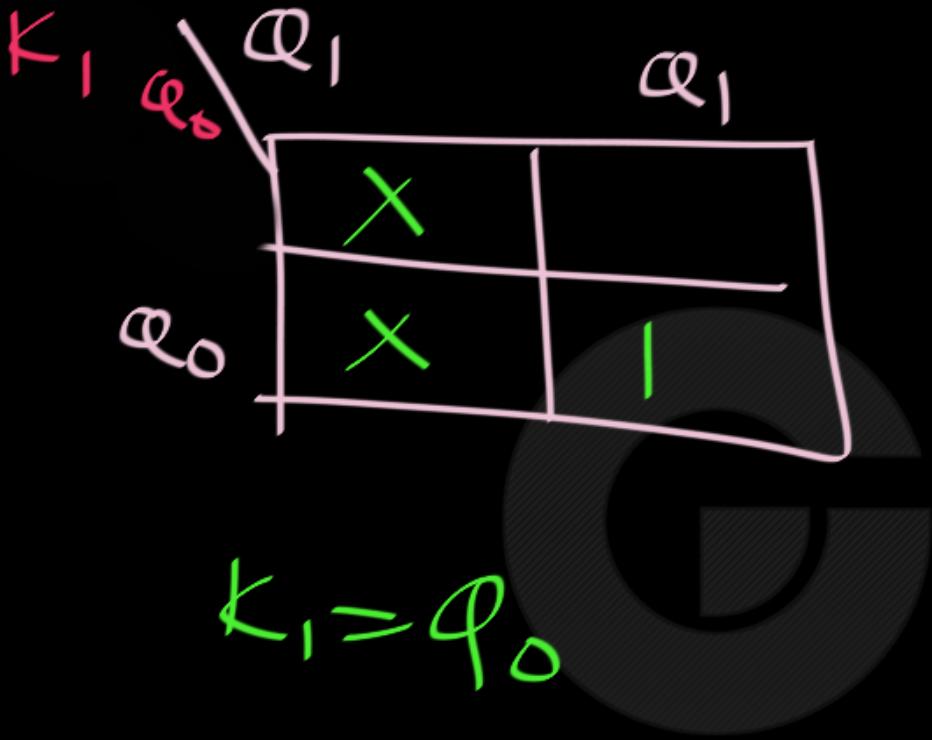


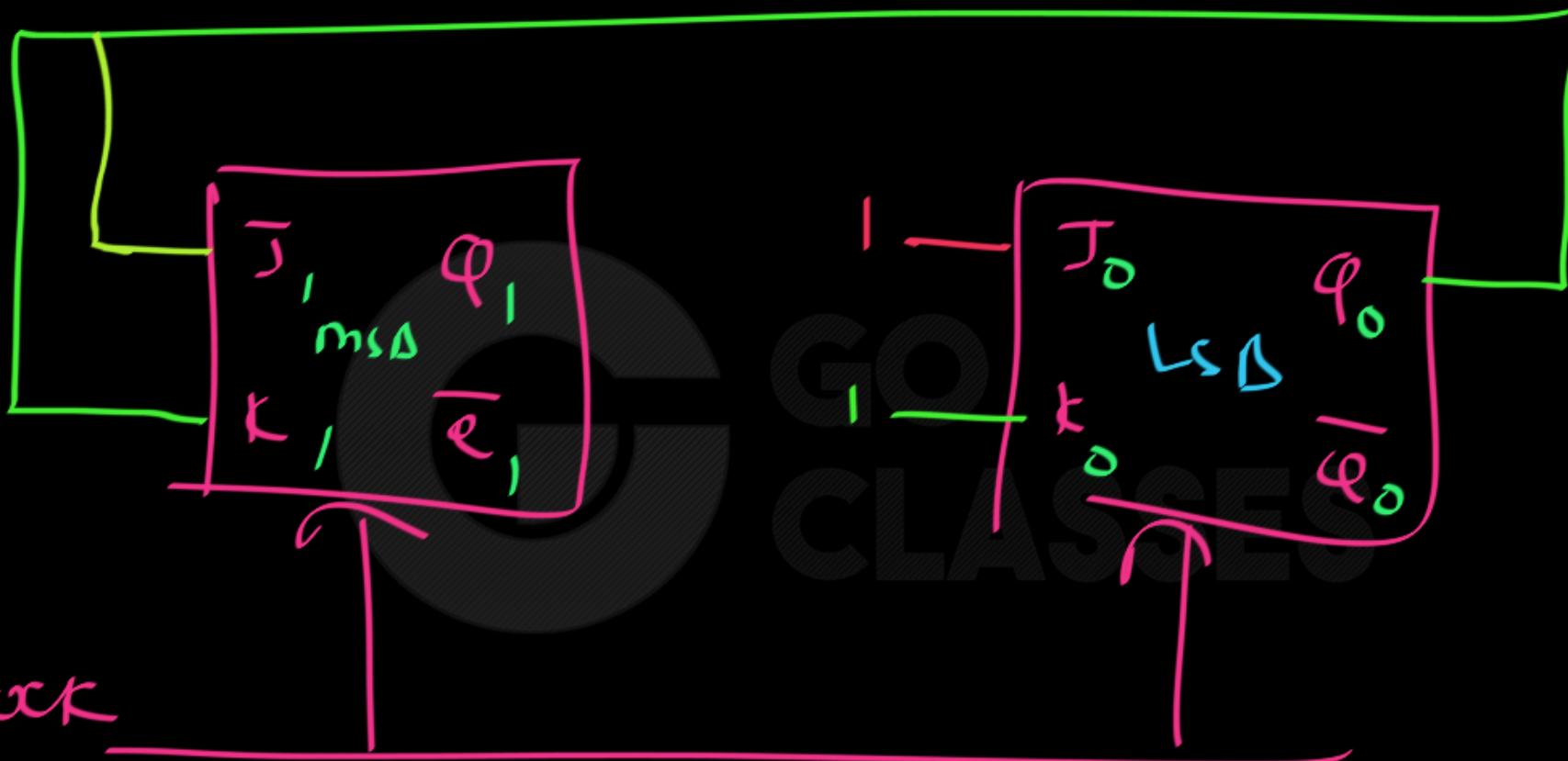
Digital Logic

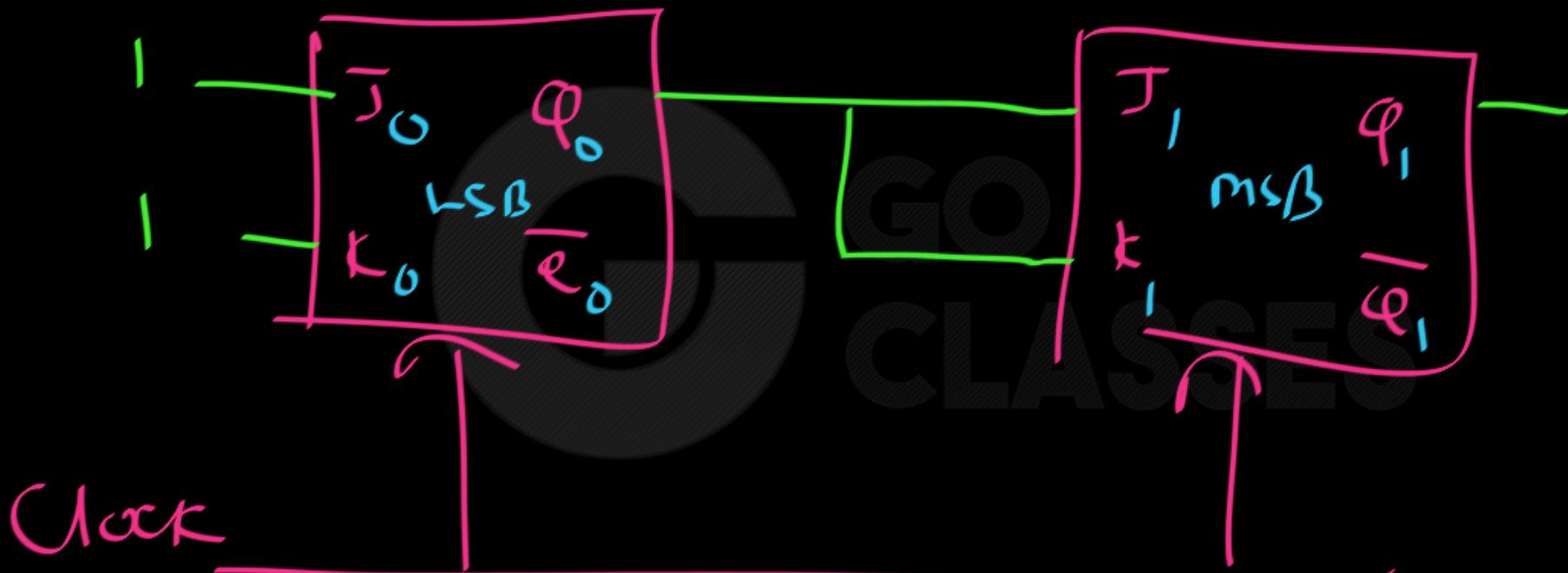




Digital Logic



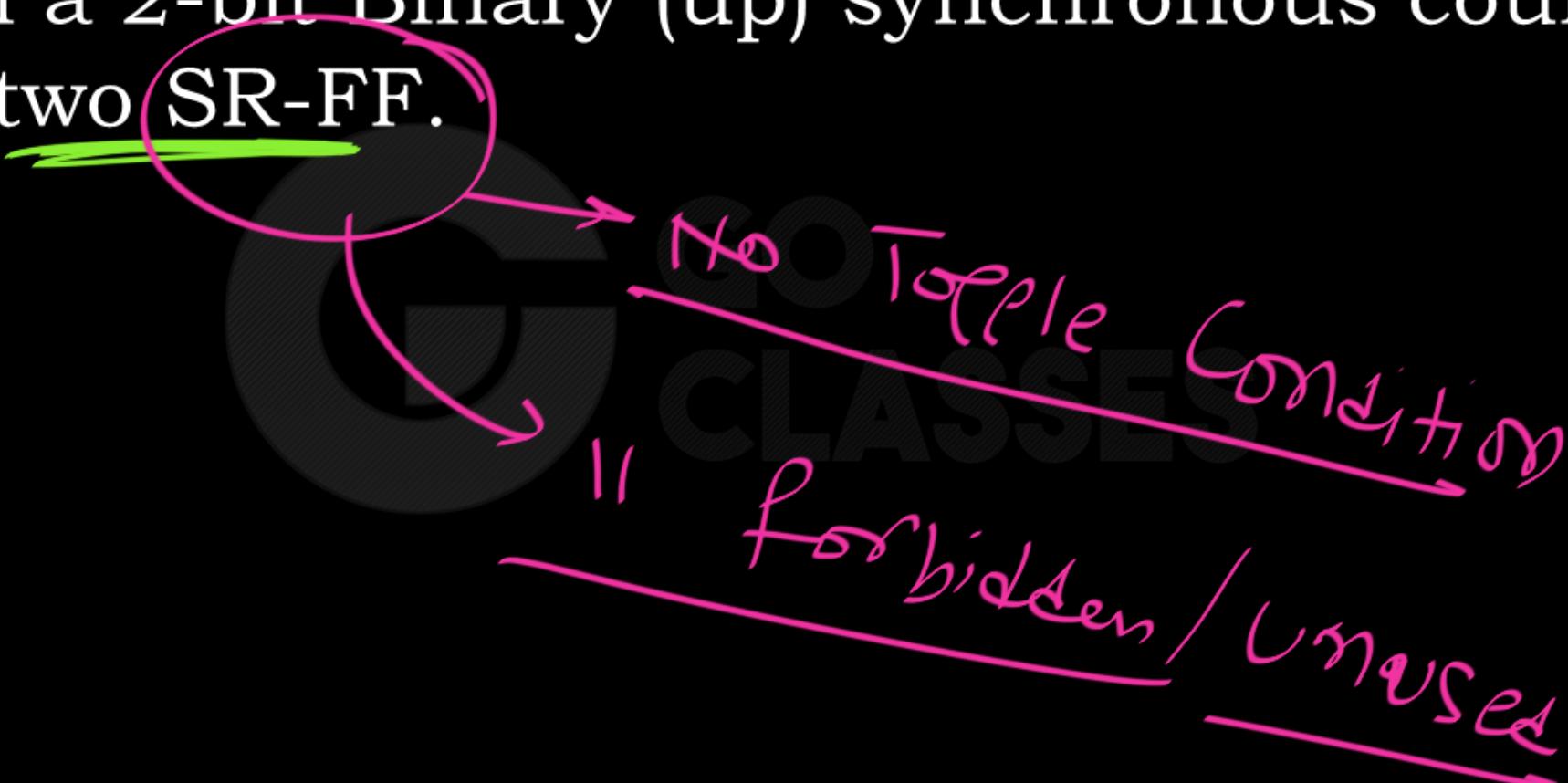


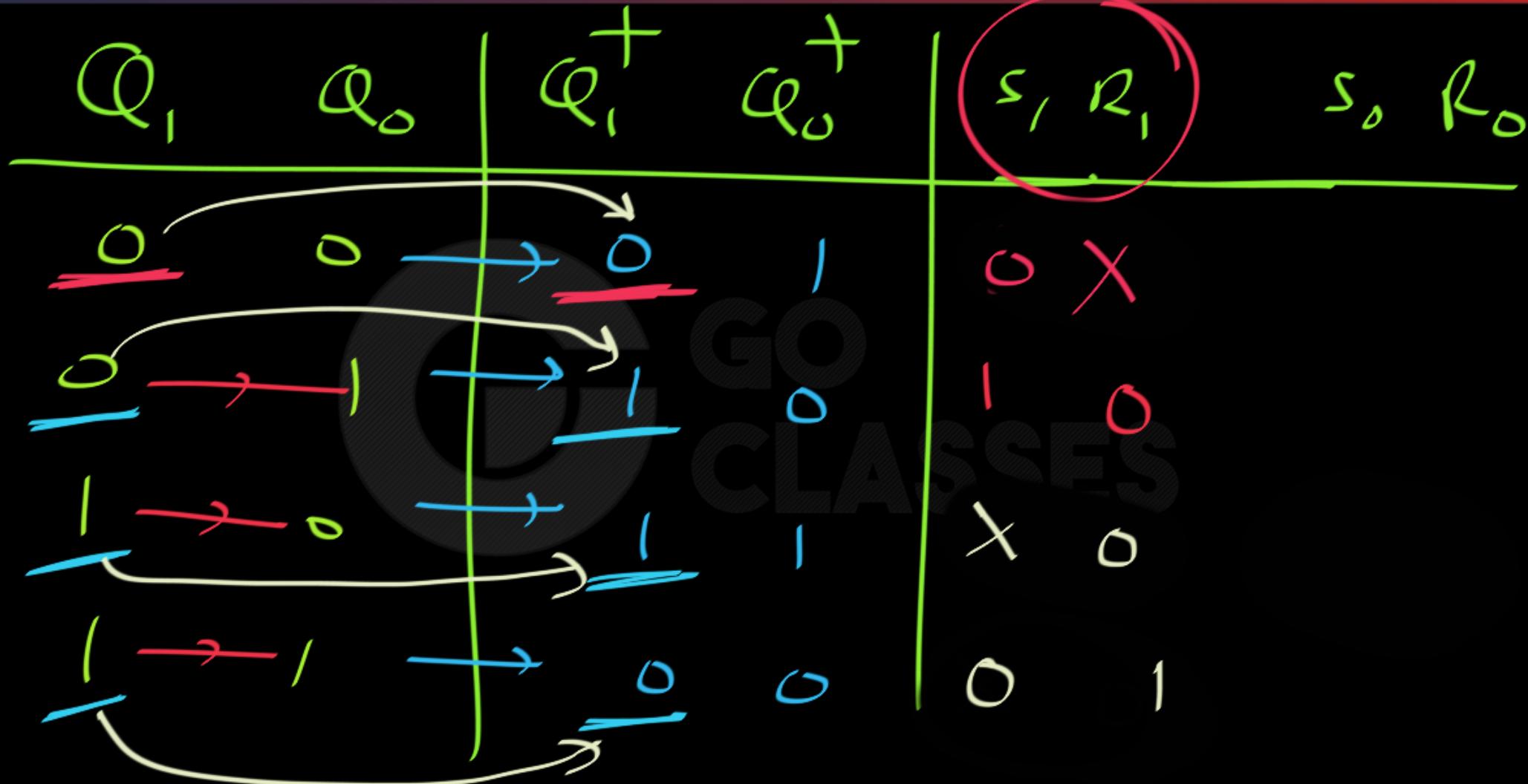


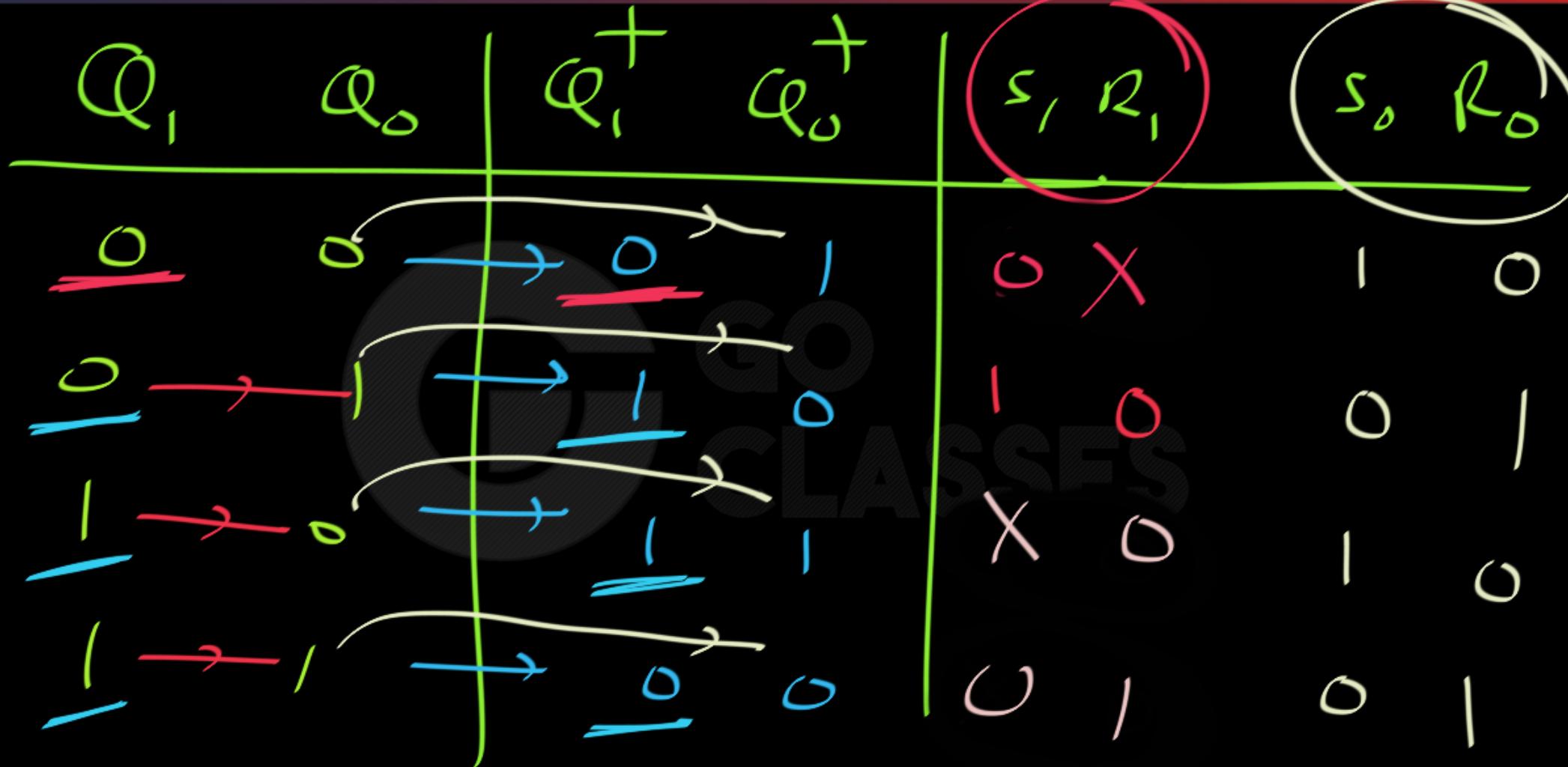


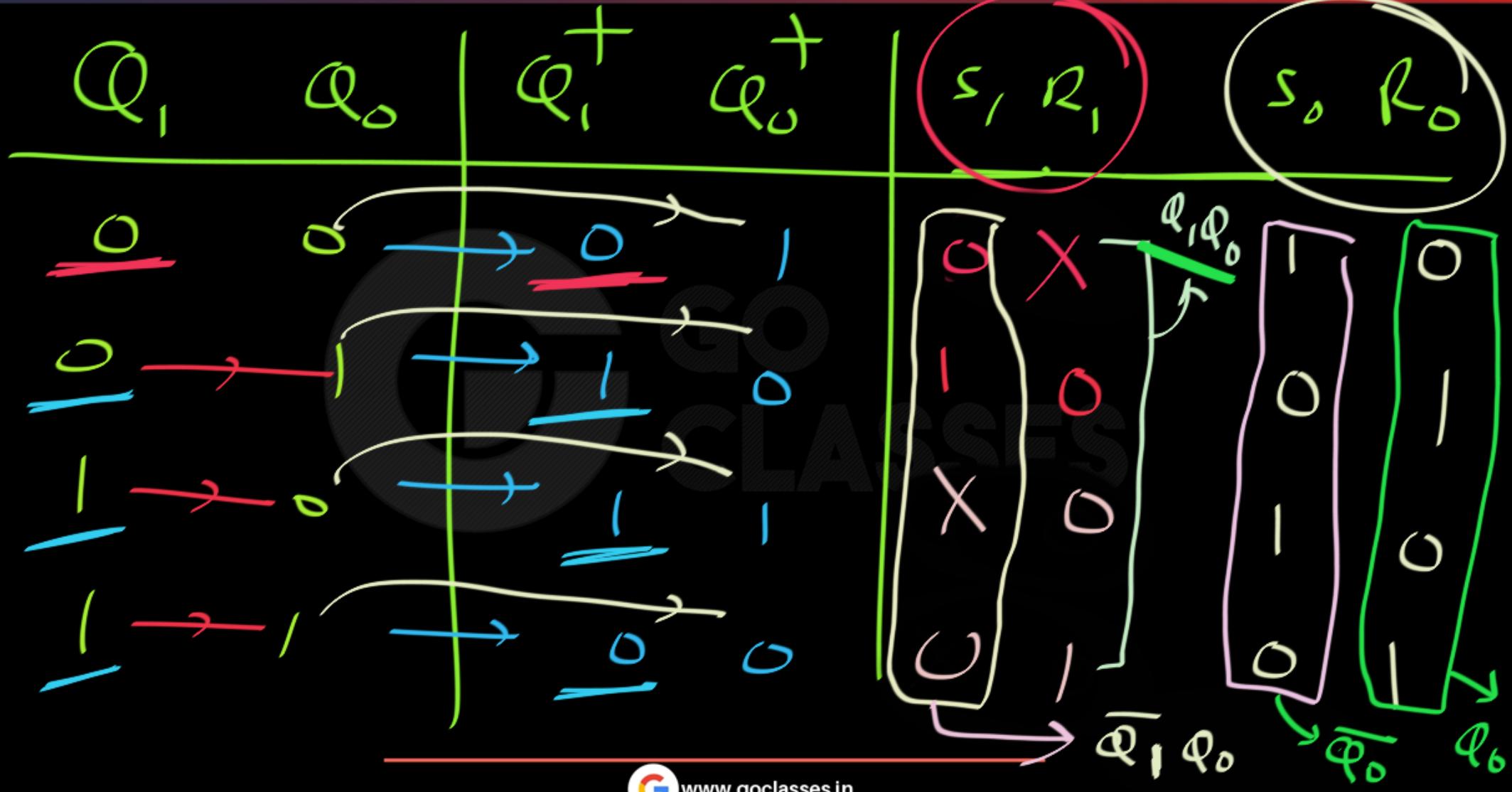
Q:

Design a 2-bit Binary (up) synchronous counter using two SR-FF.











$$\underbrace{S_1, S_0, R_1, R_0}_{\text{Inputs}} = f(Q_1, Q_0) \quad \text{Outputs}$$

Using k-map:

GO
CLASSES

Q:

Hw ✓

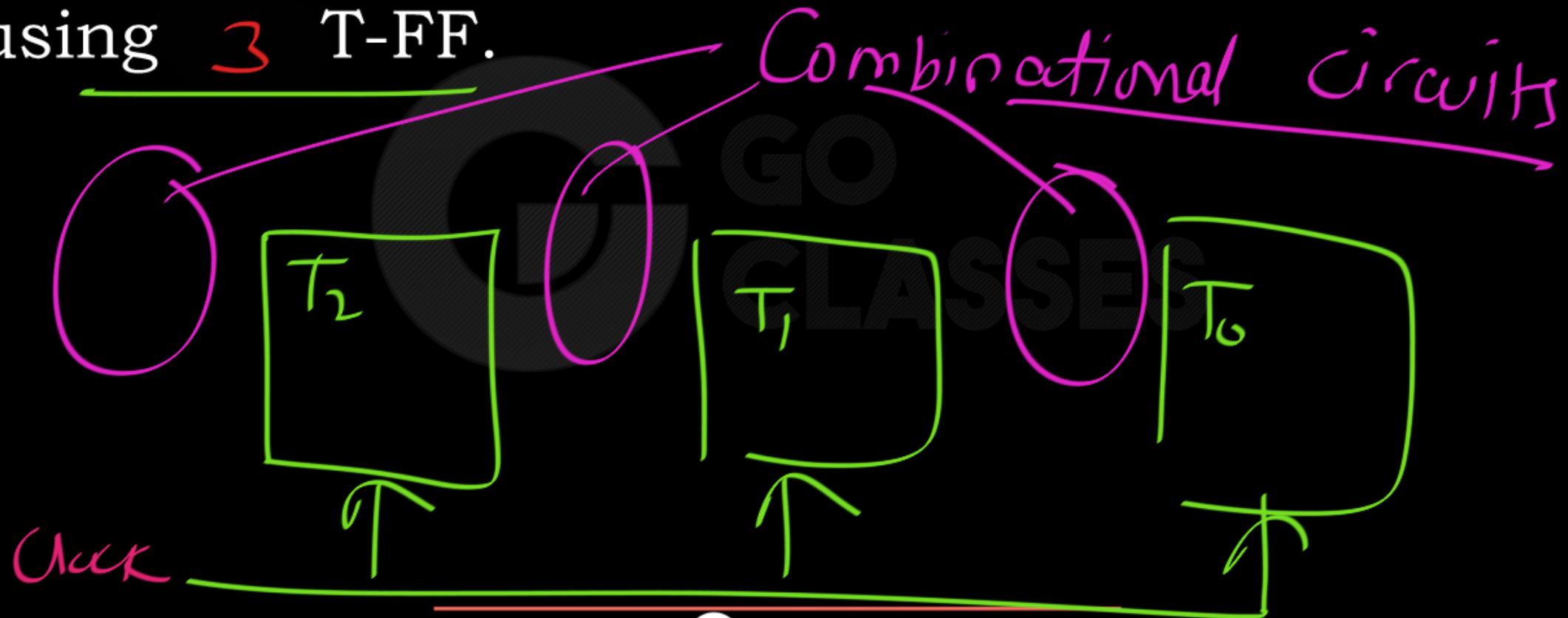
Design a 3-bit Binary (up) synchronous counter
using 3 D-FF.



Q: ✓



Design a 3-bit Binary (up) synchronous counter using 3 T-FF.



Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	T_2	T_1	T_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	0
1	0	1	1	1	0	0	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	1	1	1

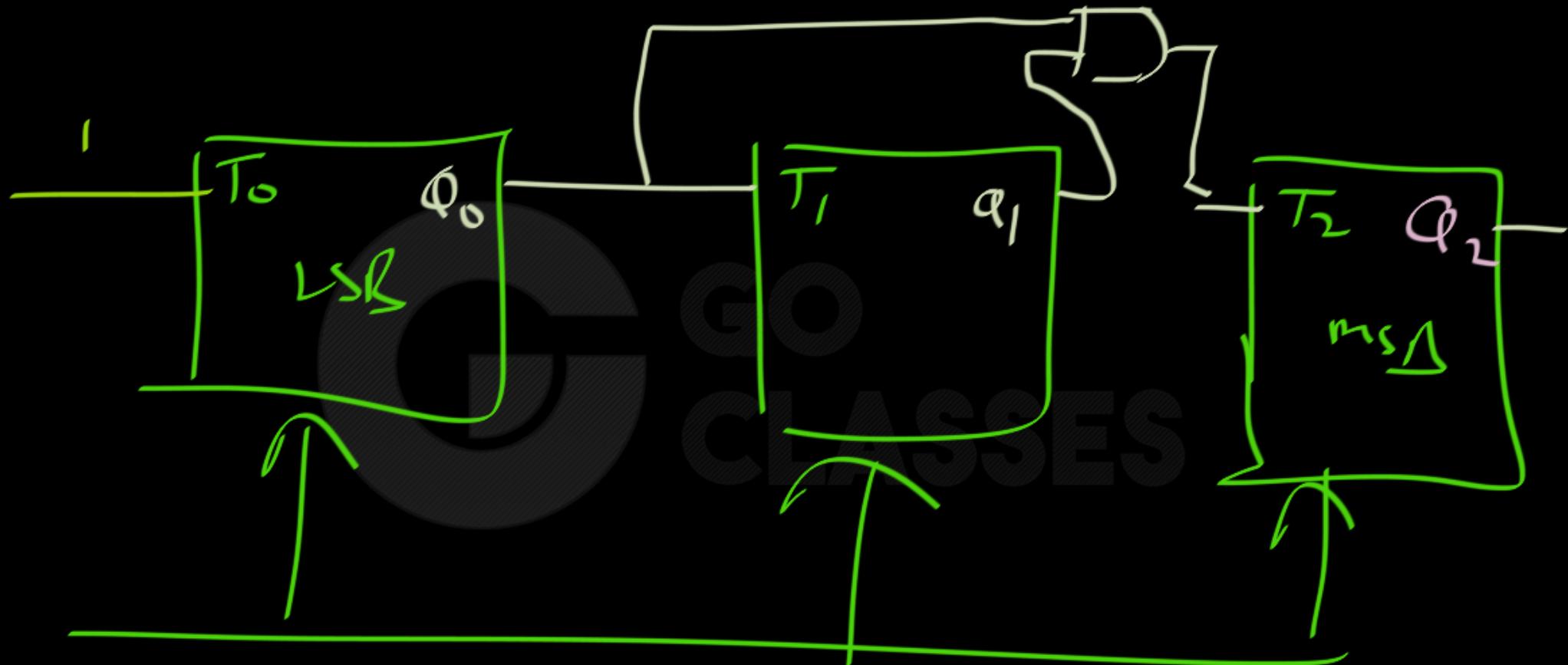


$$T_2 = \overline{Q_2} Q_1 Q_0 + Q_2 Q_1 Q_0 = \underline{Q_1 Q_0}$$

$$T_1 = Q_0$$

$$\overline{T_0} = 1$$







Using

D - FF

JK - FF

SR - FF

Messy (crowded)CircuitGO
CLASSES



Q:

Design a 3-bit Binary (up) synchronous counter using 3 T-FF (Use Only LOGIC, Analysis).





Digital Logic

$$\begin{array}{r} z \\ - \\ \underline{x} \\ \hline \end{array} \quad \begin{array}{r} y \\ - \\ \underline{y} \\ \hline \end{array} \quad \begin{array}{r} 0 \\ - \\ \underline{1} \\ \hline \end{array}$$

Diagram illustrating the addition of binary numbers:

- Top row: $x = 0100$, $y = 0010$, $z = 0010$ (sum)
- Middle row: $x = 0001$, $y = 0001$, $z = 0000$ (sum)
- Bottom row: $x = 1000$, $y = 1000$, $z = 0100$ (sum)

Annotations:

- Red arrows point from the bottom row to the middle row.
- A green arrow points from the bottom row to the top row.
- A pink arrow points from the bottom row to the bottom row.

$$\begin{array}{r} 0000 \\ 0001 \\ 0010 \\ \hline 0111 \end{array}$$

Diagram illustrating the addition of binary numbers:

- Top row: $x = 0000$, $y = 0001$, $z = 0001$ (sum)
- Middle row: $x = 0001$, $y = 0010$, $z = 0010$ (sum)
- Bottom row: $x = 0010$, $y = 0111$, $z = 1000$ (sum)

Annotations:

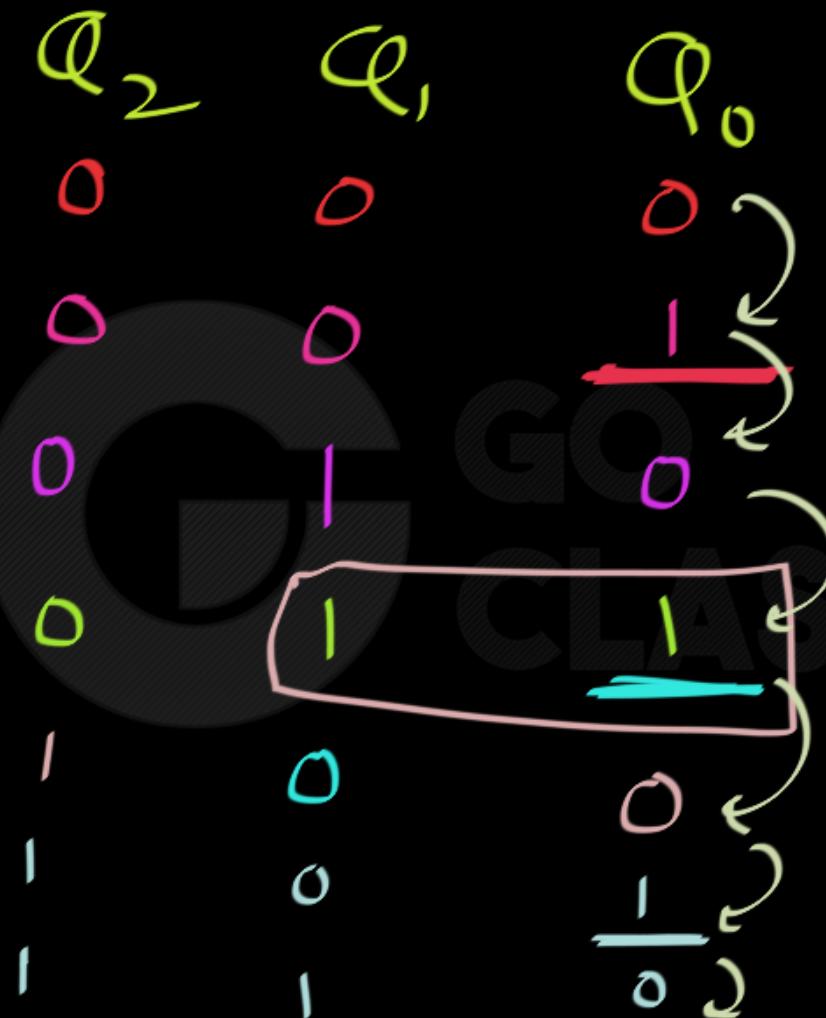
- Red arrows point from the bottom row to the middle row.
- A green arrow points from the bottom row to the top row.
- A pink arrow points from the bottom row to the bottom row.

$$\begin{array}{c} n := \Sigma \\ \downarrow \\ \underline{n+1 := \Sigma} \end{array}$$

A diagram consisting of three curved arrows originating from the left side of the first equation and pointing towards the bottom equation. The top arrow is red, the middle arrow is green, and the bottom arrow is blue.



Digital Logic

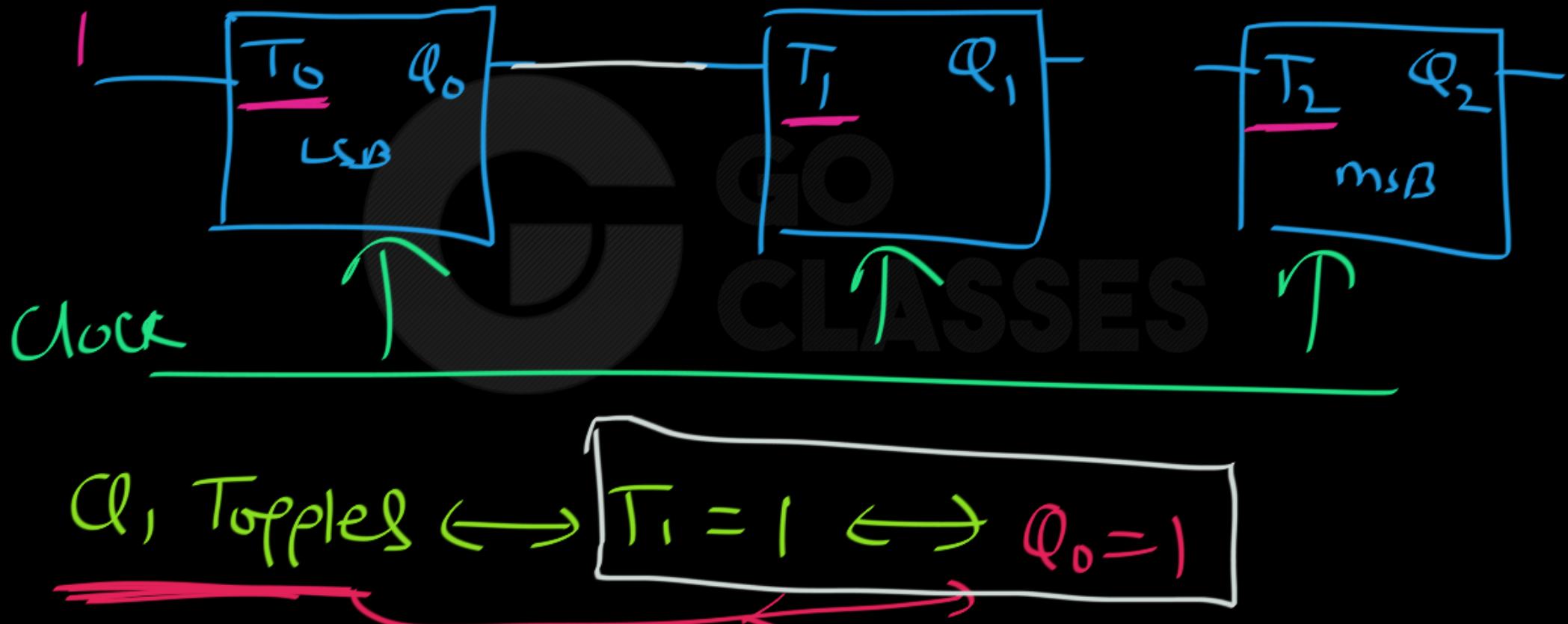


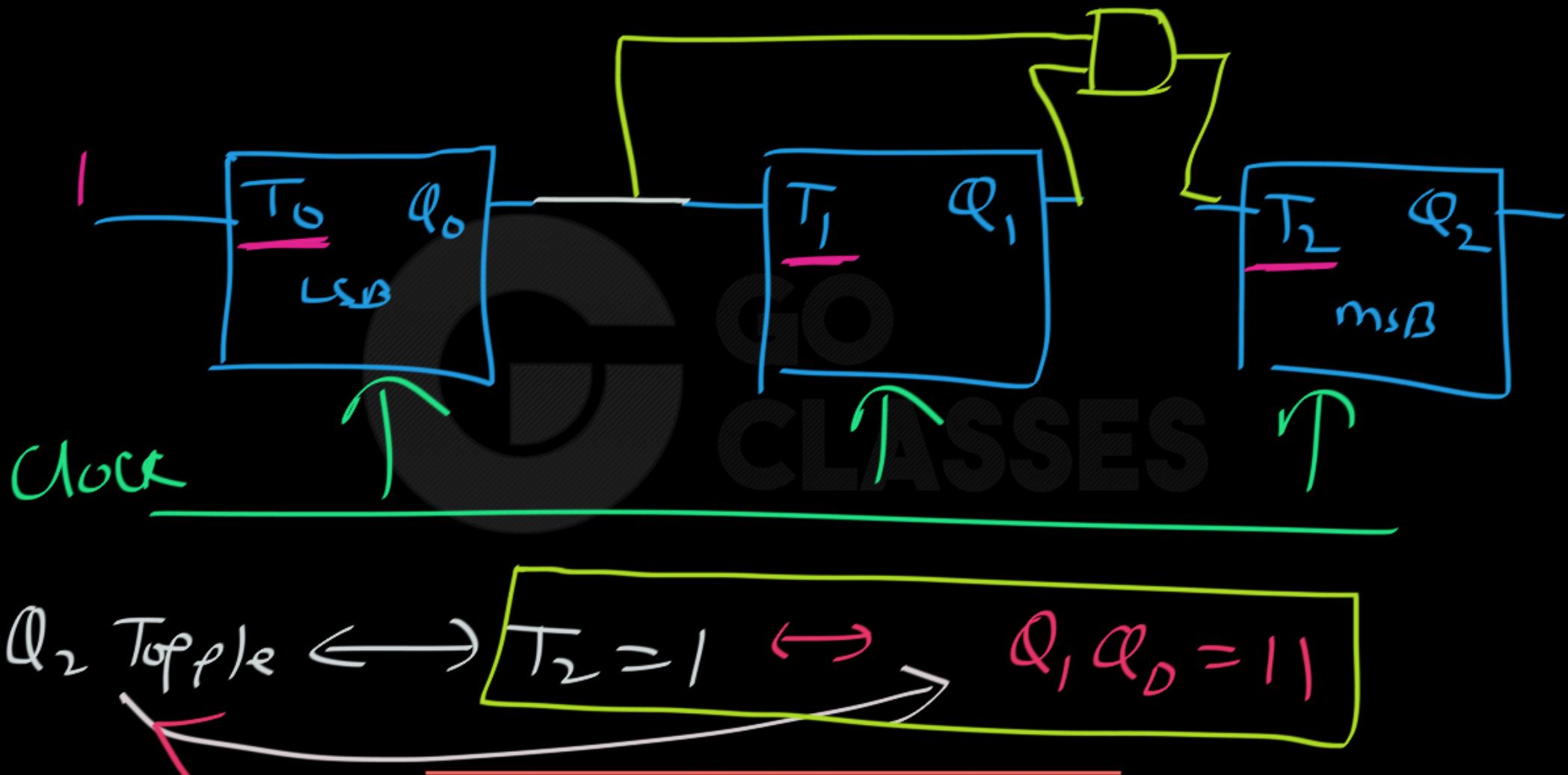
$Q_o = LSR = \frac{\text{must change in}}{\text{Every clock cycle}}$

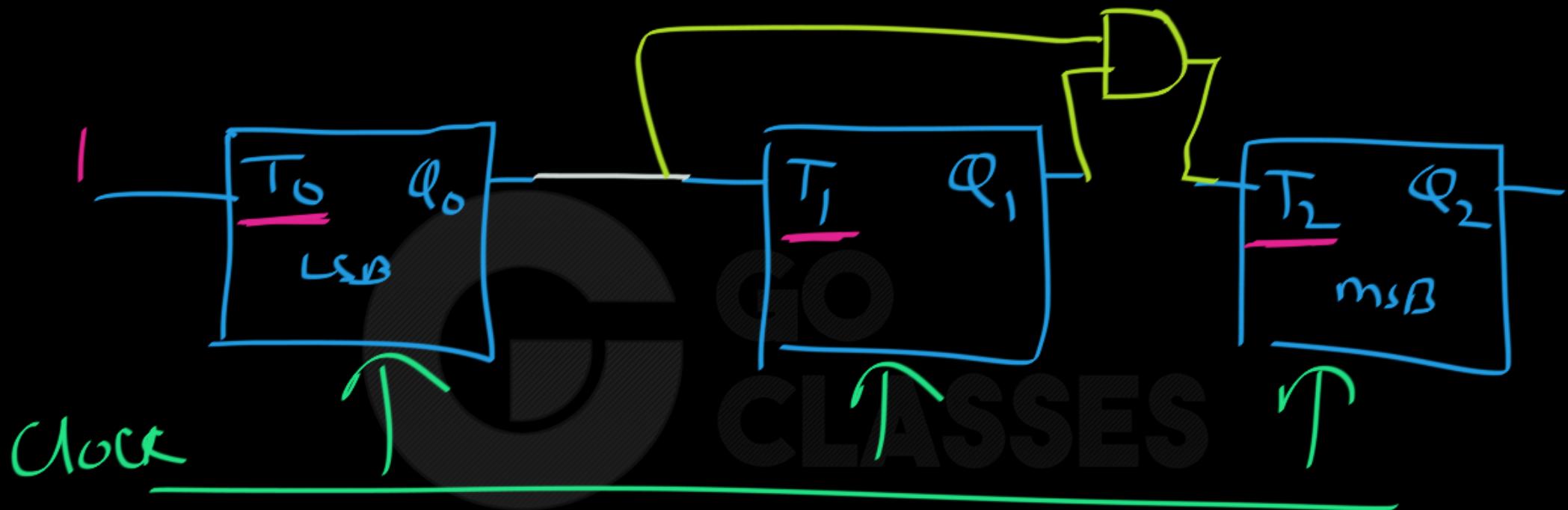
$Q_1 = \text{change when } Q_o = 1$

$Q_2 = \text{change when}$

$Q_o = 1$
 $Q_1 = 1$



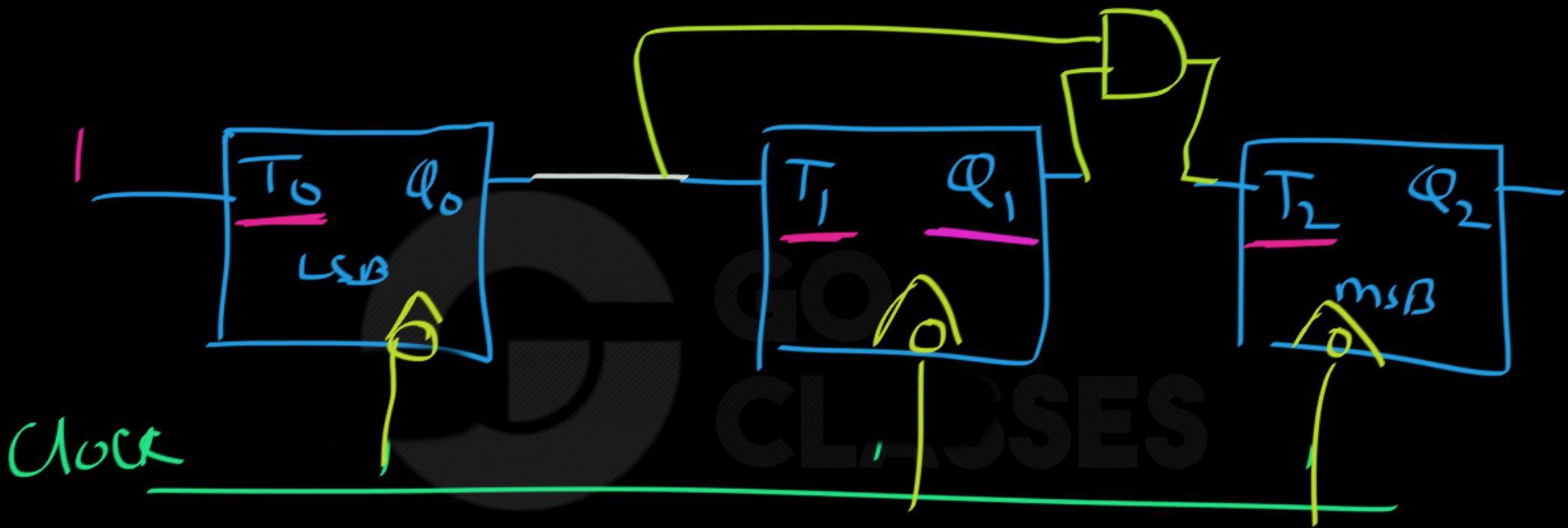


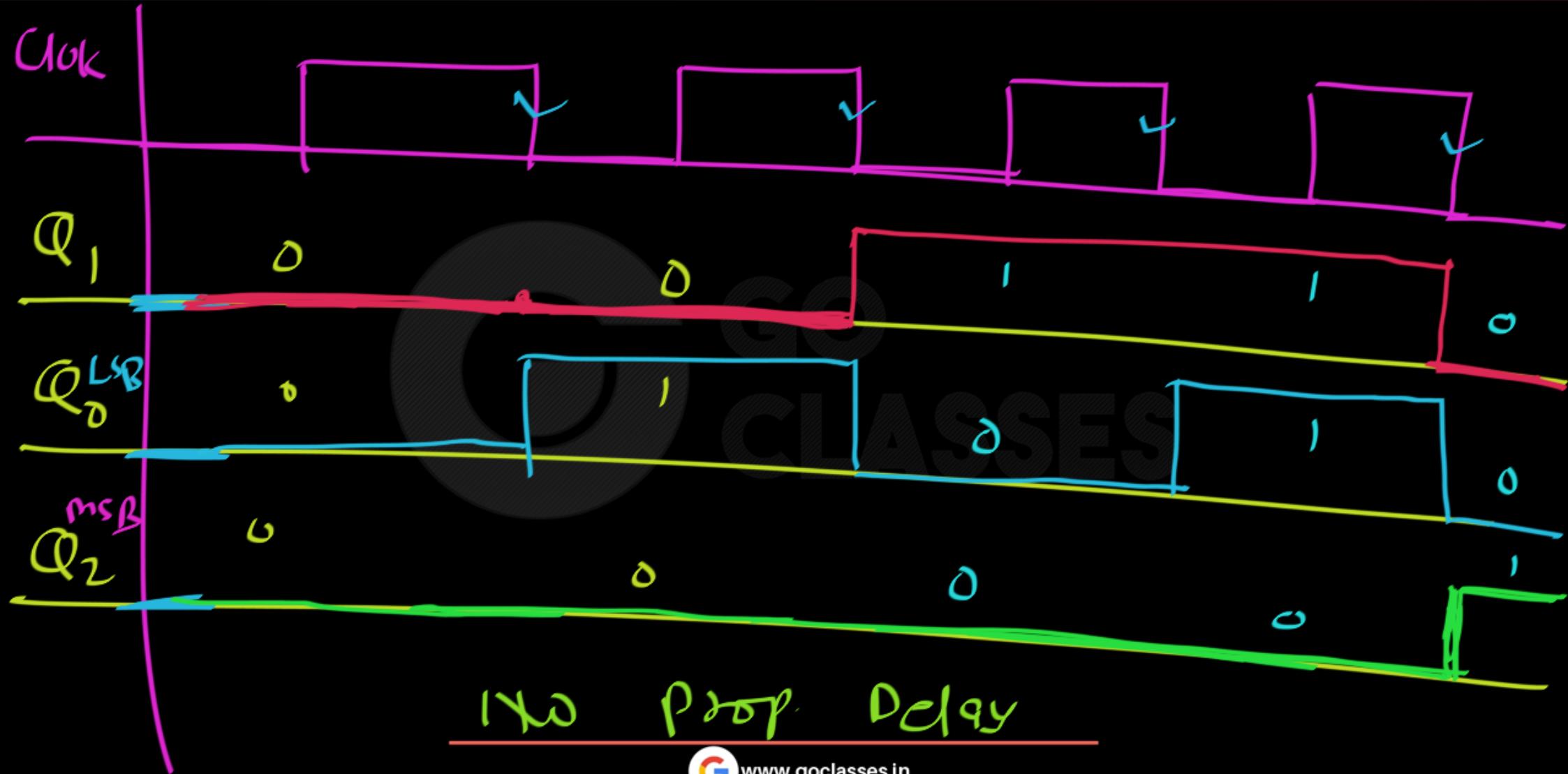




In Synch. CKts: flip flop's

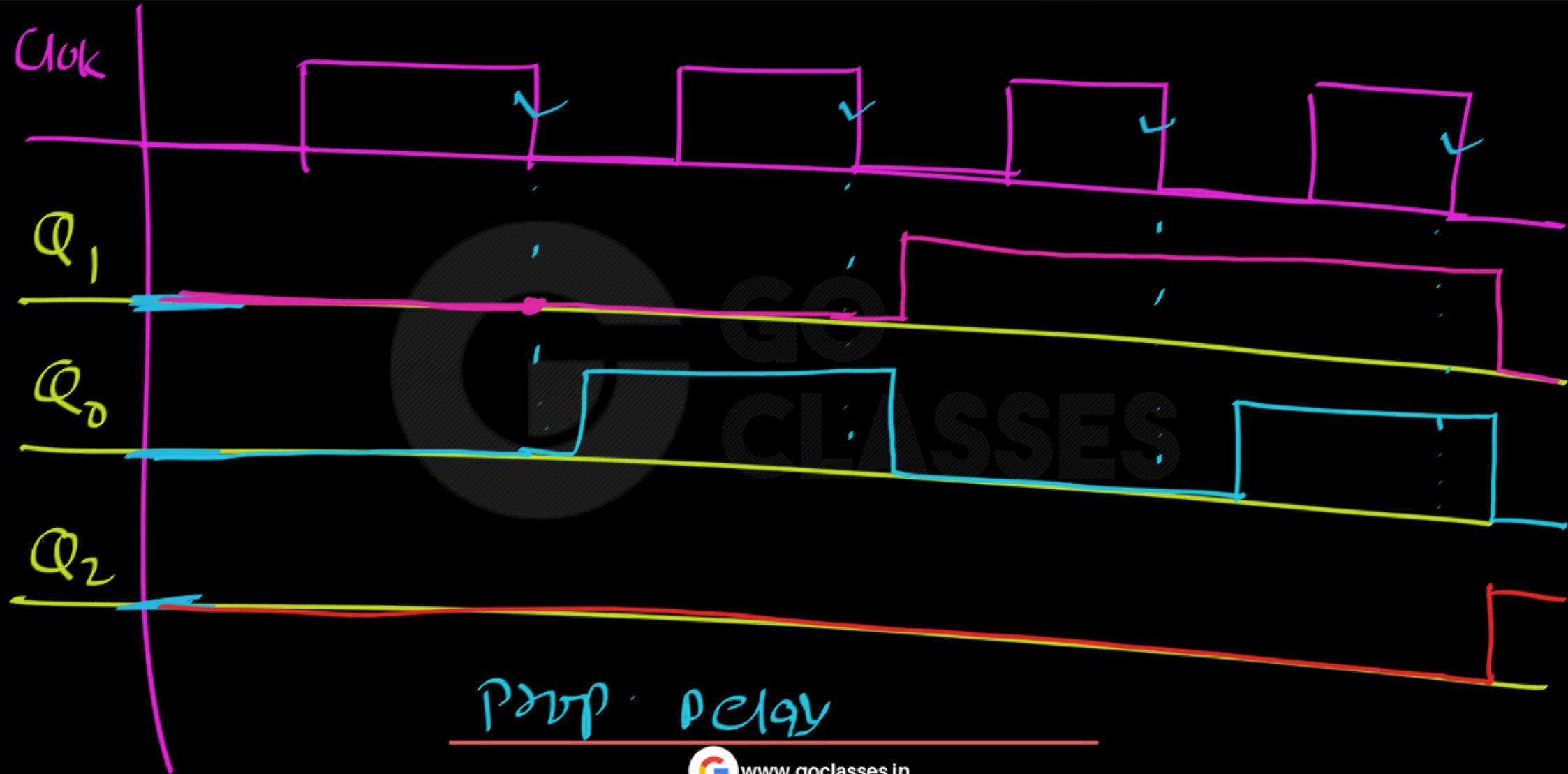
Clock Triggering does not change
the circuit





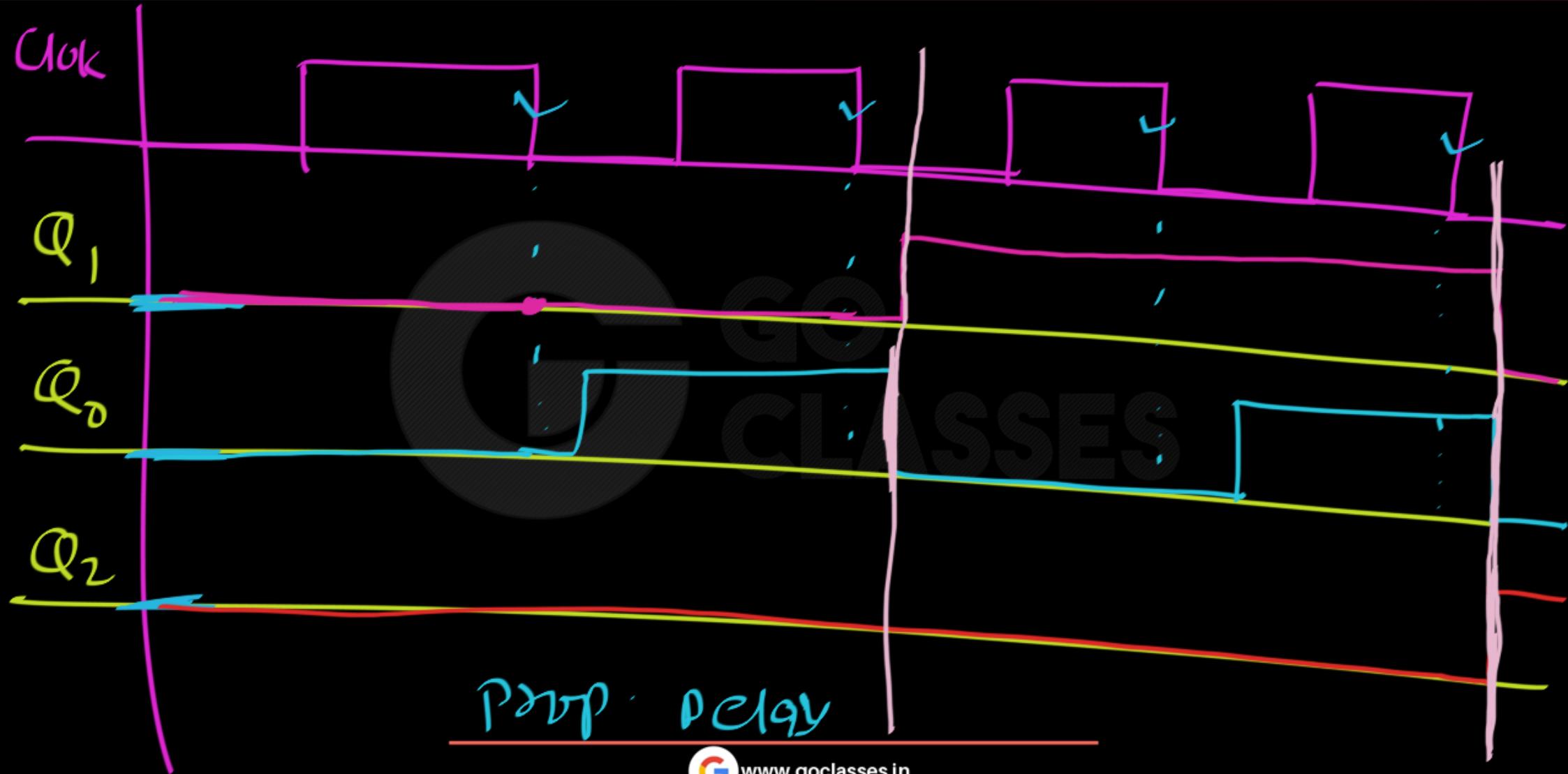


Digital Logic





Digital Logic



When FFs have prop. delay
Synch Counter Vs Ripple Counter.

No Transient state



Why?

All FFs Affect
at the same time.

Transient state

When FFs have prop. delay
Synch Counter Vs Ripple Counter.

No Transient State

In every FF stage
same prop. Delay.
Does not depend
on #FF.

Transient State

Prop. Delay increase
from 1st FF to last FF
Period Time(Clock) ≥ #FF × t_{pd}

When FFs have prop. delay V_s

Synch Counter Ripple Counter.

$$\text{time period (clock)} \geq t_{pd}$$

$$\text{time period (clock)} \geq n t_{pd}$$

In every FF stage
same prop. Delay.
Does not depend
on #FF.

~~prop. delay increases from 1st FF to last FF~~
Period Time (clock) $\geq \#FF \times t_{pd}$



Ripple Counter :

Highest frequency
of Clock $\frac{1}{\#ff \times t_{PL}}$



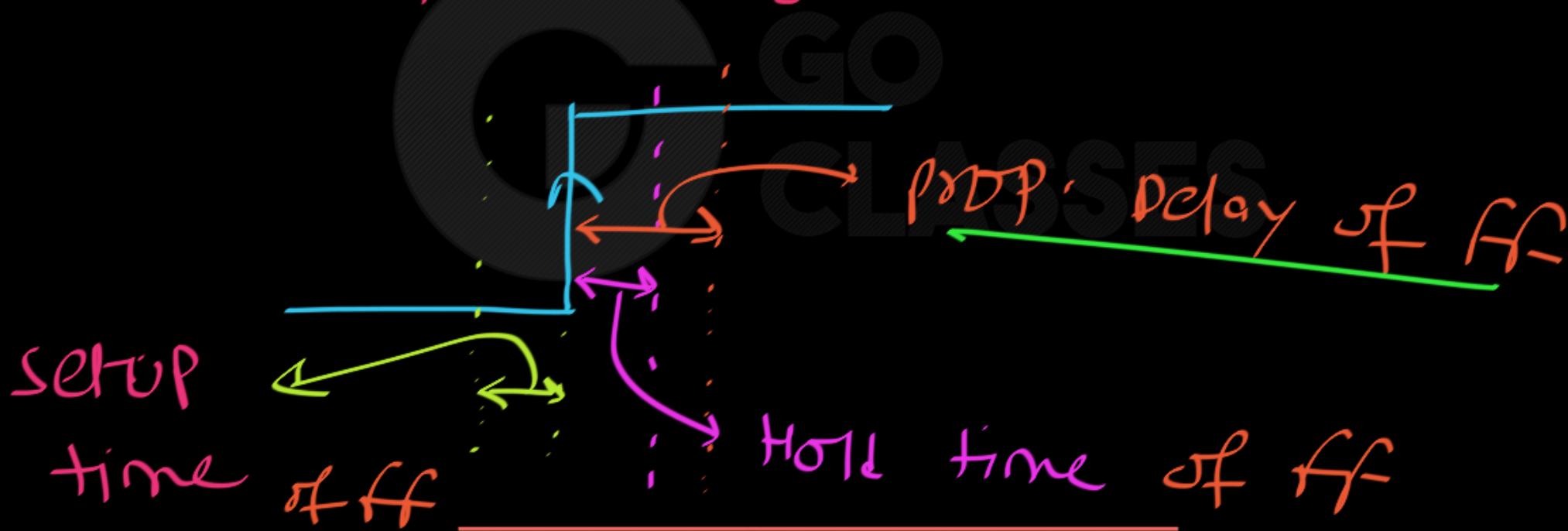
Synch. Counter:

Highest frequency
of Clock



If setup time, Hold time are

Also Given !



Setup Delay , Hold time

one of them covers another.
(usually PD covers)
Hold time
so whichever is maximum, take it.

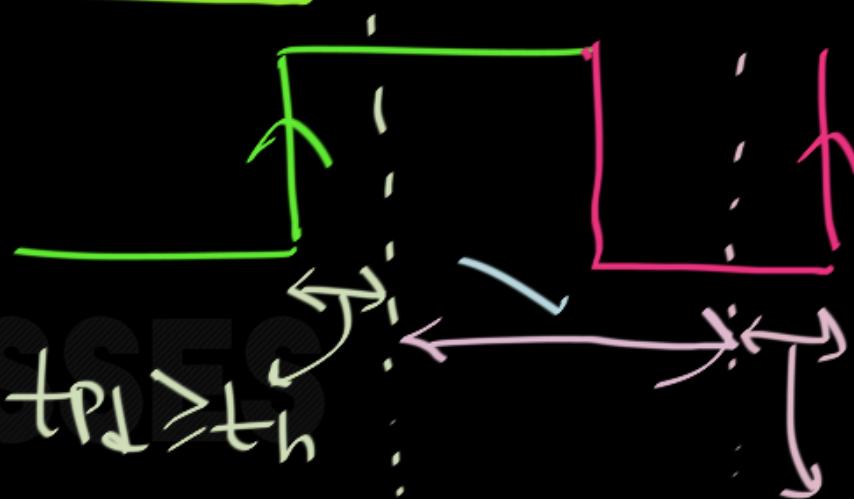
If setup time, Hold time are

Also Given

Syncho Counter:

Time Period of Clock

$$\geq [t_{PD} + t_{S\!P}]$$





If setup time, Hold time are

Also Given

t_o

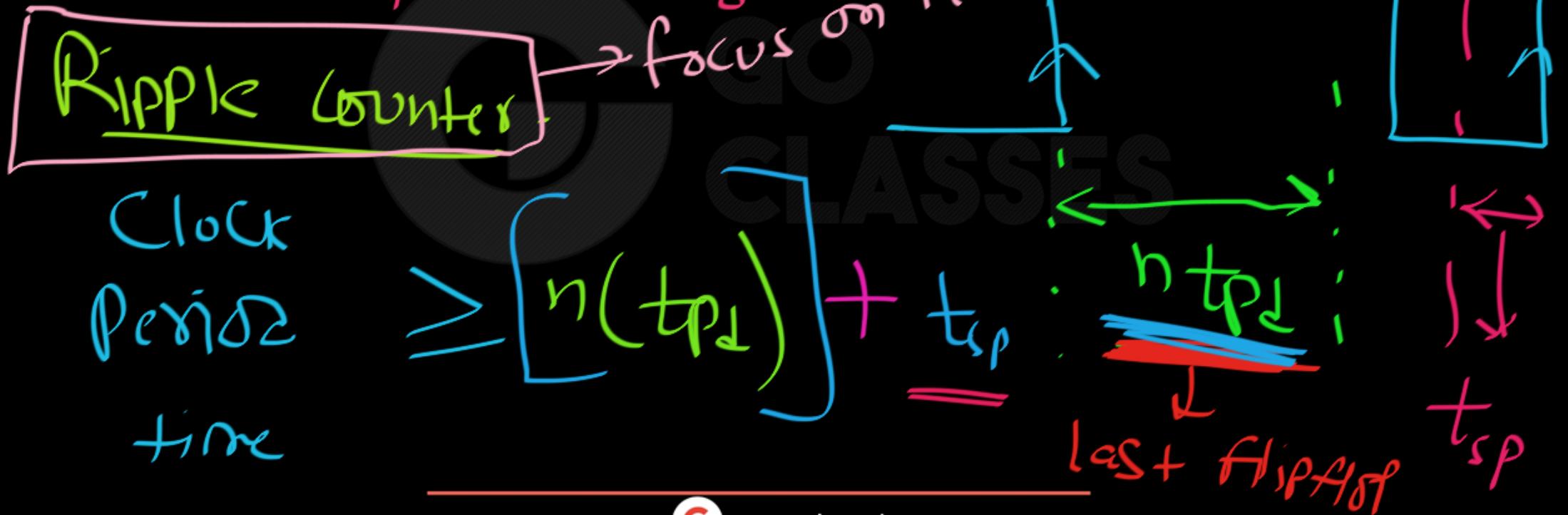
GO
CLASSES

Syncho Counter:

$$\text{max freq} \leq \frac{1}{(t_{pd} + t_{sp})}$$

If setup time, Hold time are

Also Given focus on last flipflop





If setup time, Hold time are

Also Given

Ripple Counter:

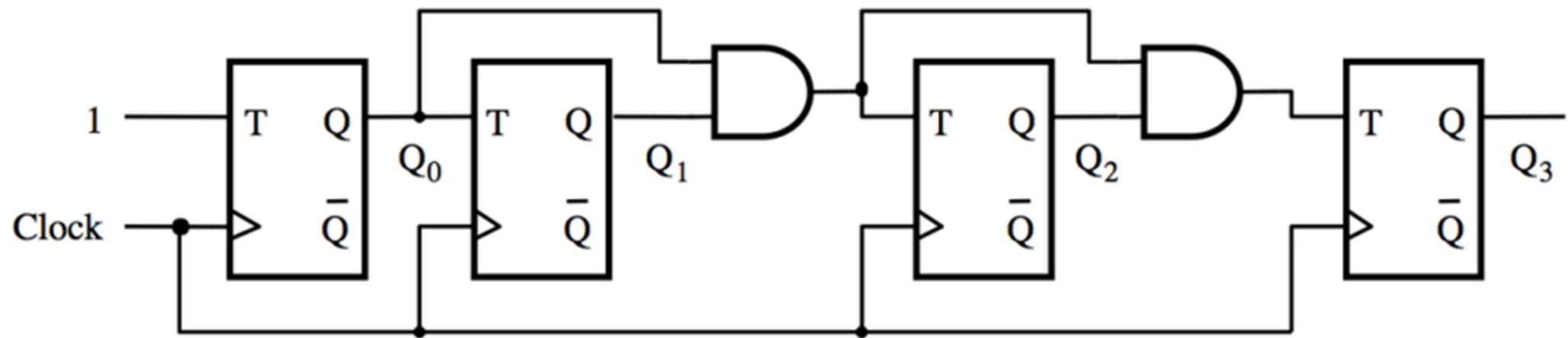
$$\text{max freq} \leq \frac{1}{(n t_{PL}) + t_{SP}}$$



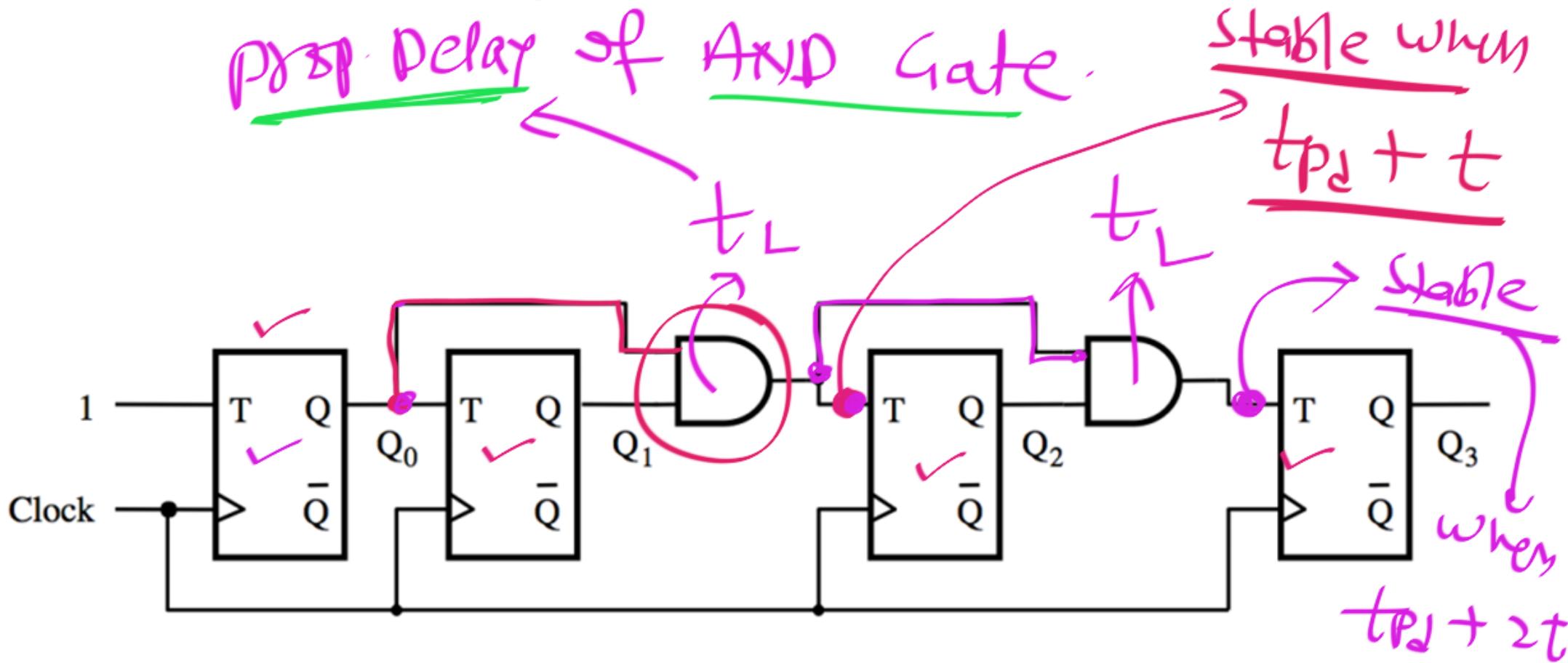
Next Topic:

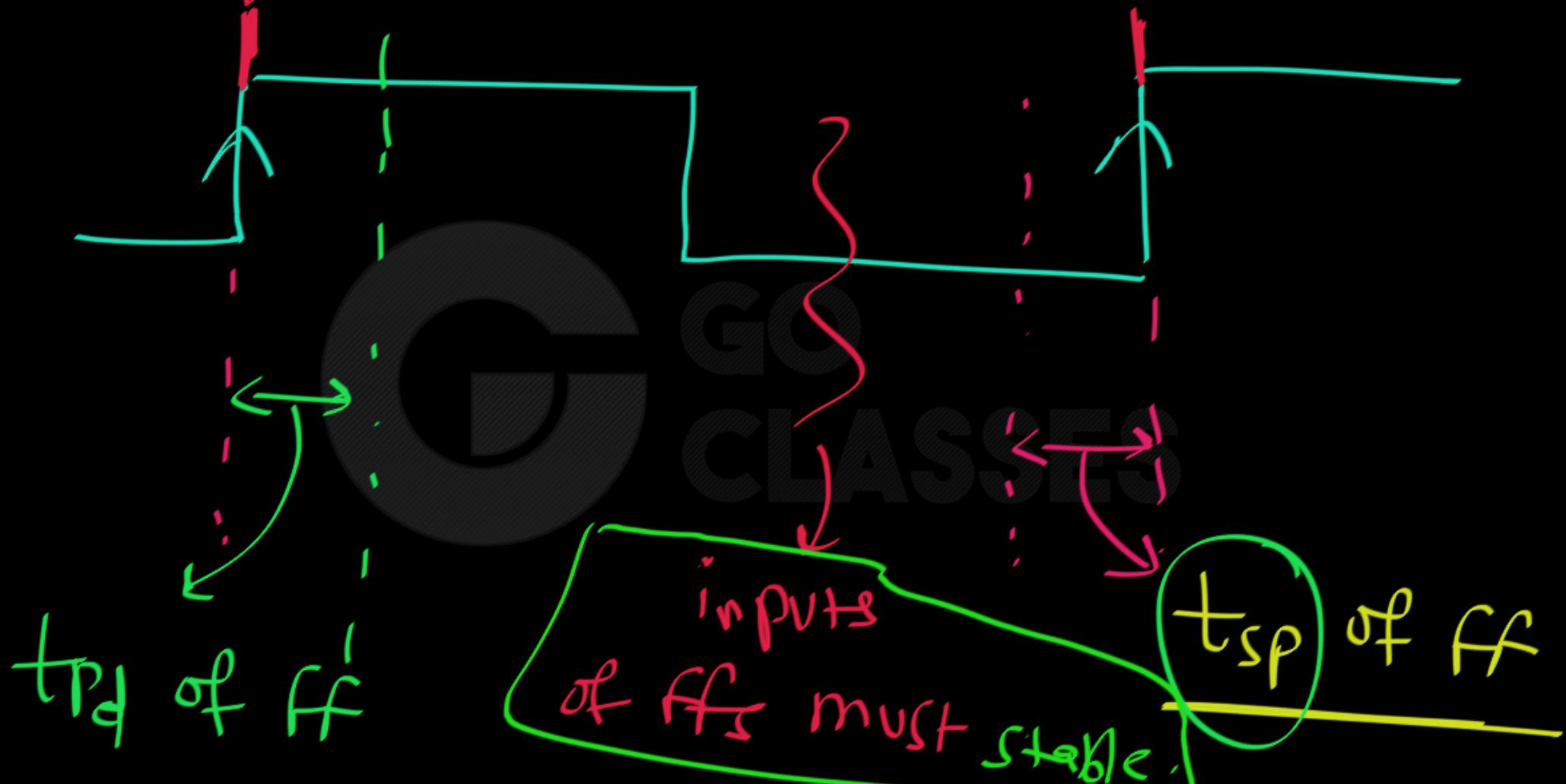
Maximum Frequency of Synchronous Sequential Circuits

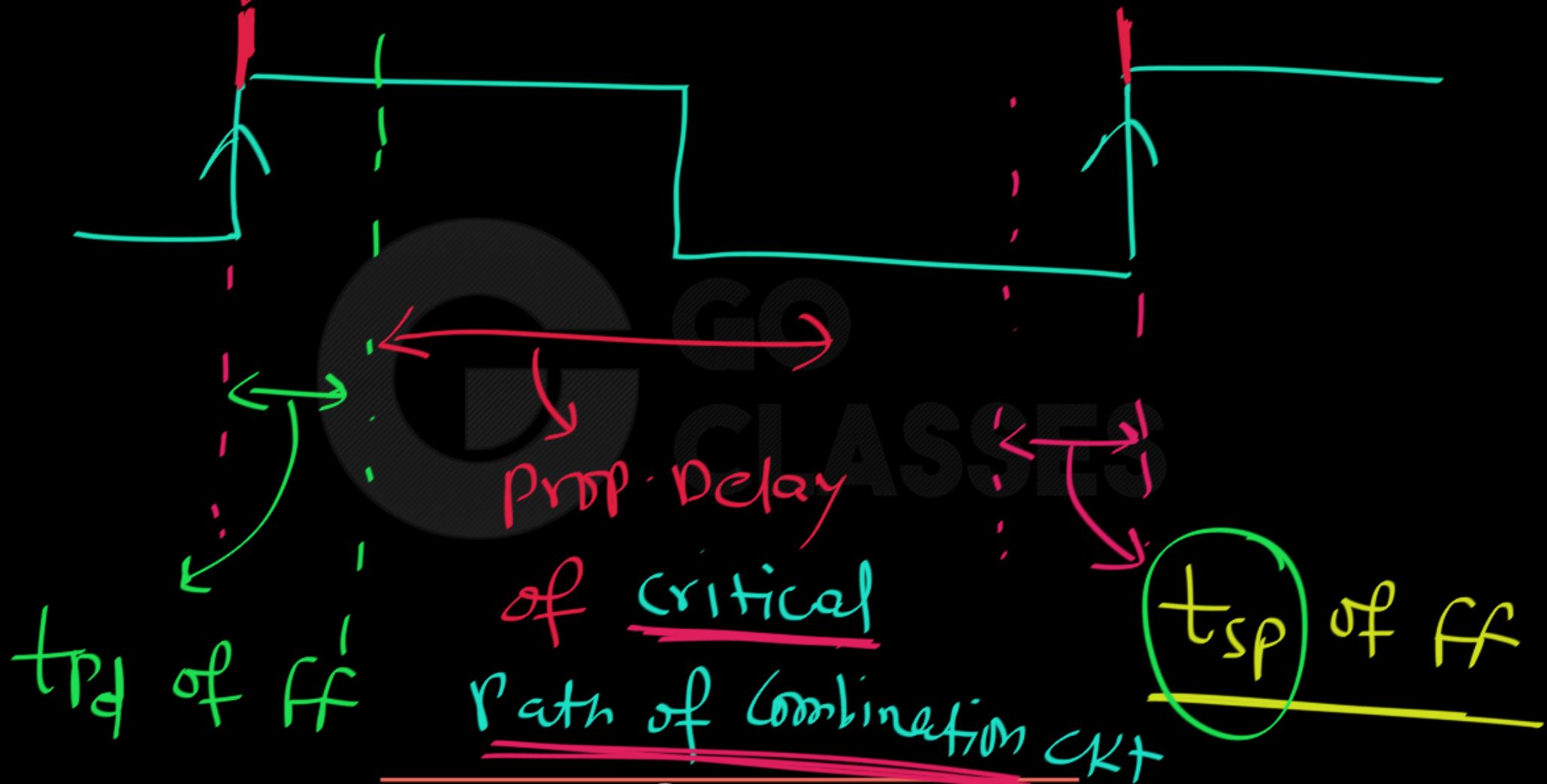
A four-bit synchronous up-counter



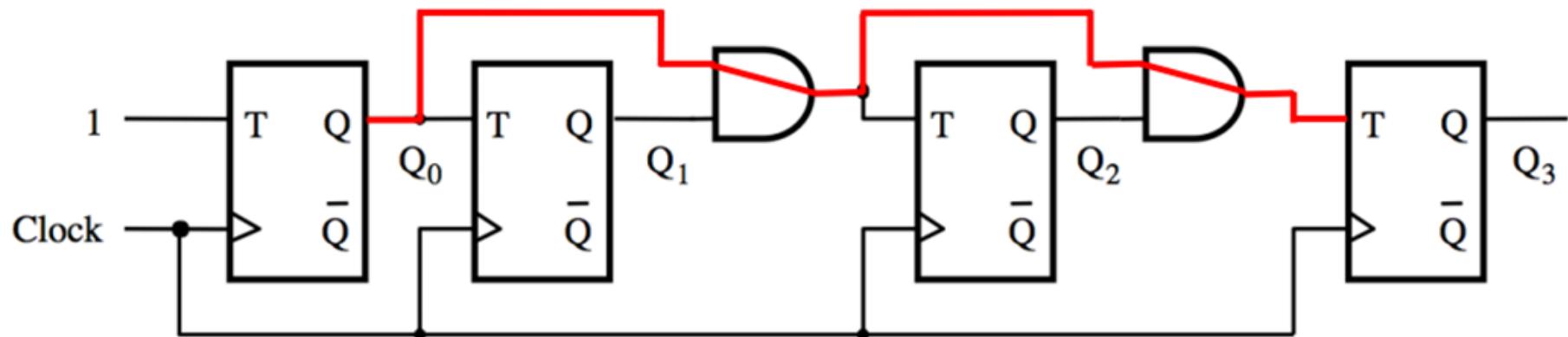
A four-bit synchronous up-counter





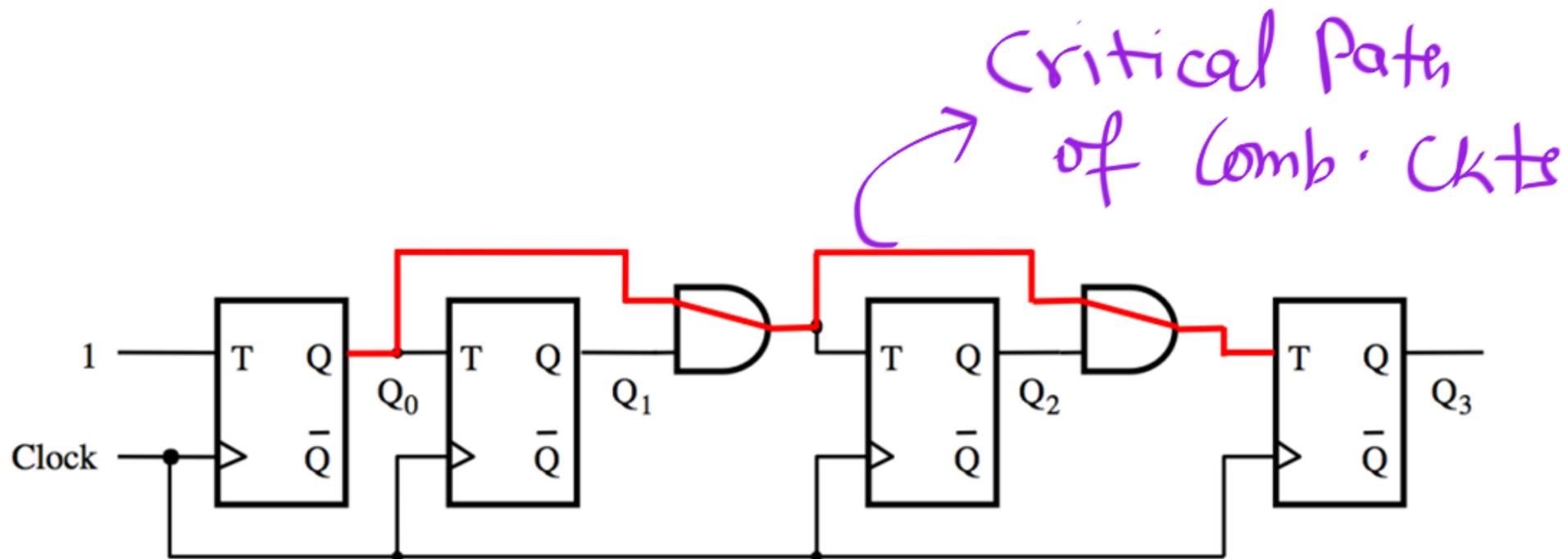


A four-bit synchronous up-counter



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

A four-bit synchronous up-counter

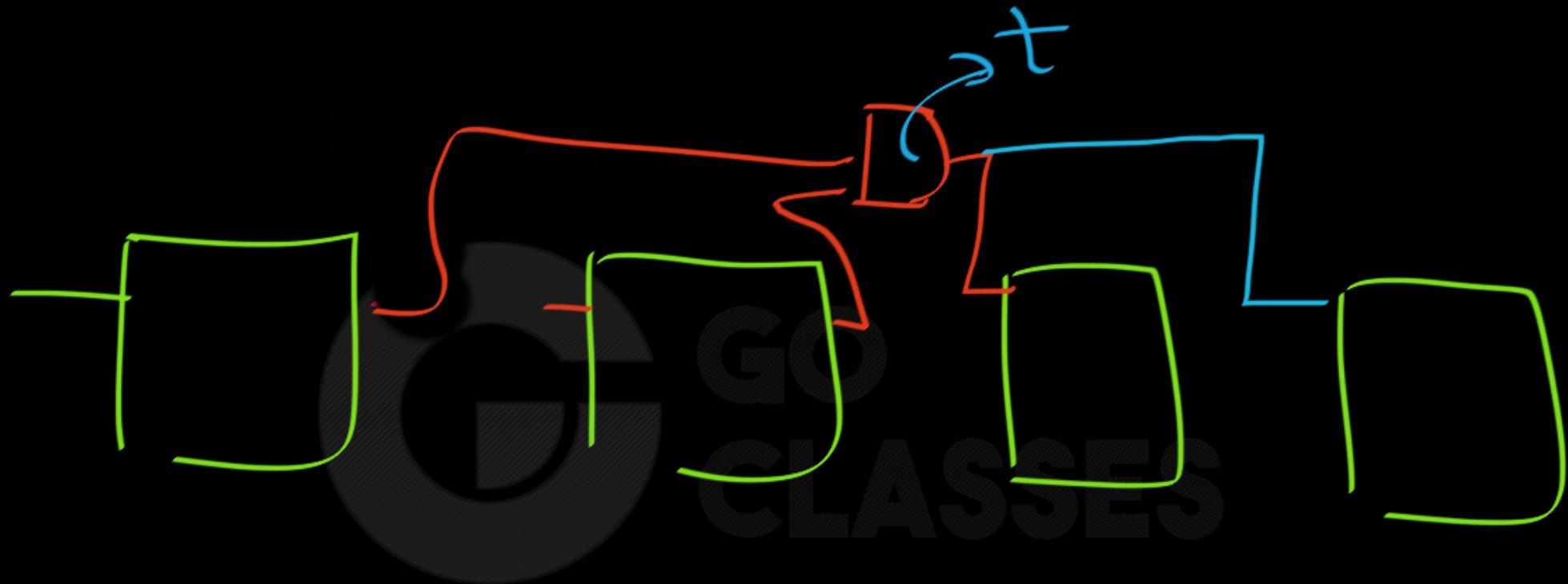


The propagation delay through all AND gates combined must not exceed the clock period minus the setup time for the flip-flops



Syn. Counter:

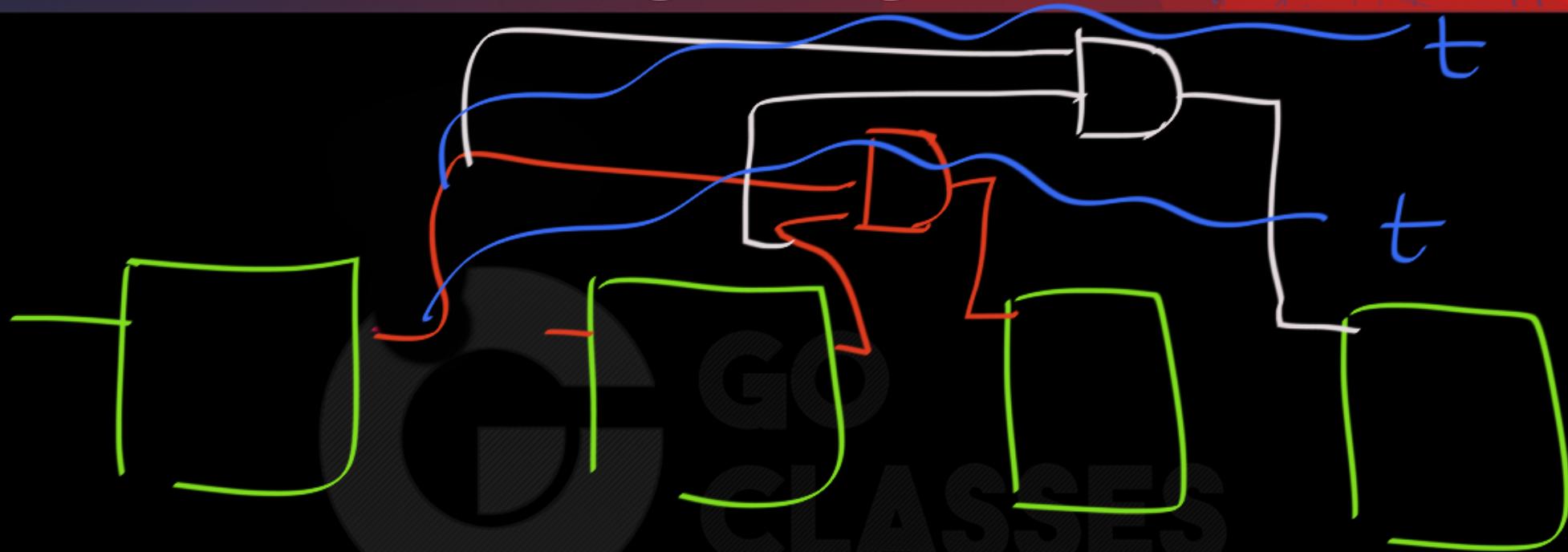
time period of clock \geq $(t_{PD} \text{ of FF}) + (t_{SP} \text{ of FF}) + \text{critical Path PD of logic gates}$



$$\text{Clock } t \geq t_{Pd} + t_{SP} + t$$



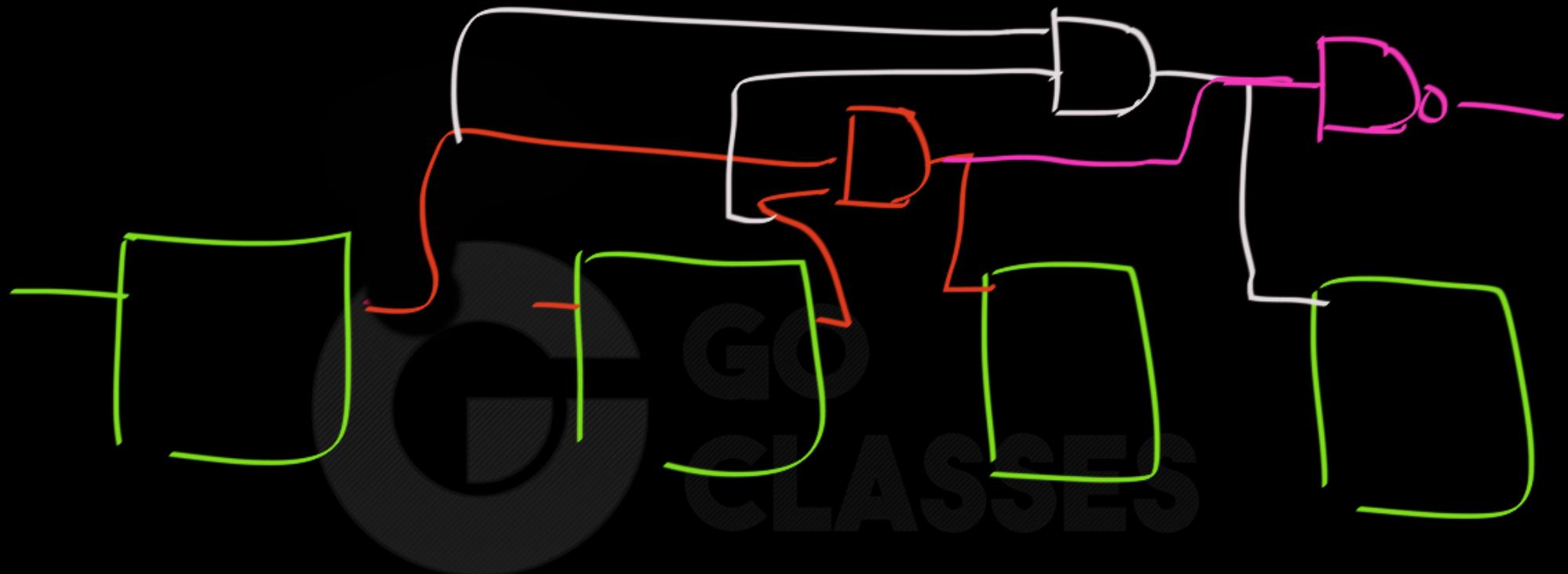
$$\text{Clock } t \geq (t_{PLH} + t_{PHL}) + 2t$$



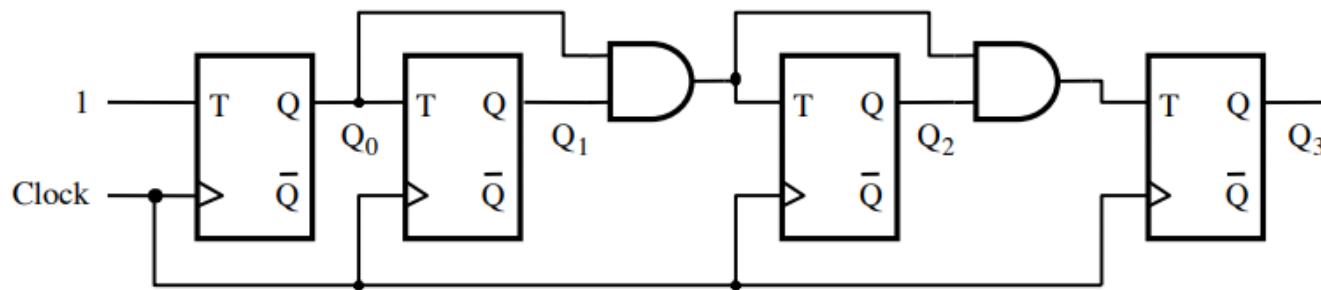
$$\underline{\text{Clock}} + \underline{t} \geq \left(t_{PD} \right)_{FF} + \left(t_{SP} \right)_{FF} + \underline{t}_{\text{critical path}}$$



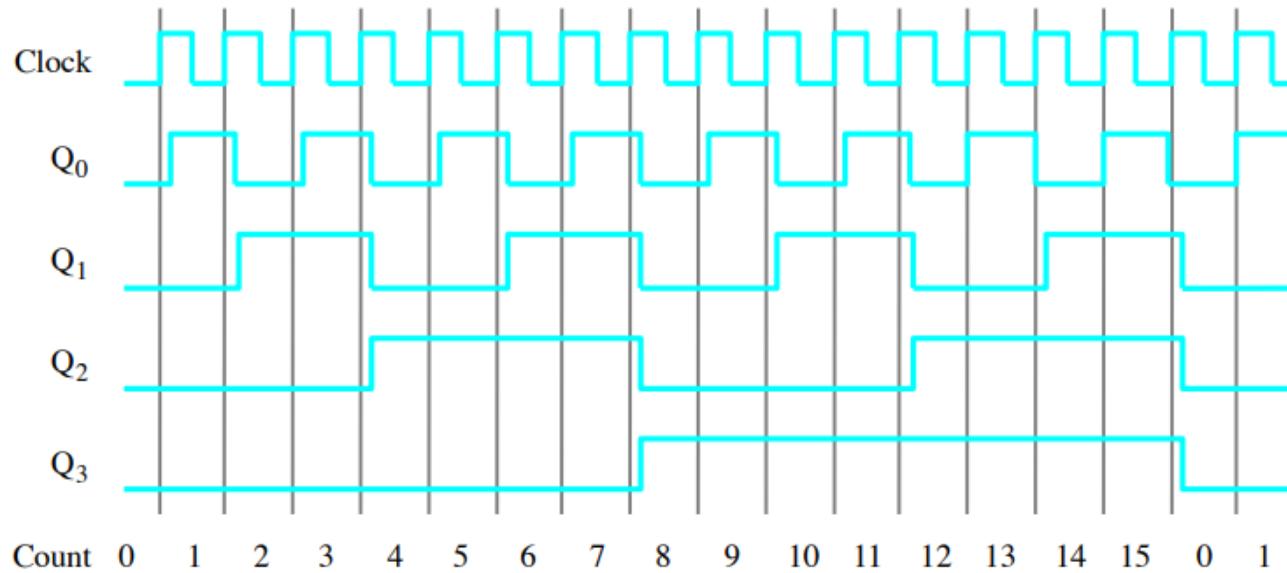
Digital Logic



$$\text{Clock}_t \geq - + - + (2+)$$



(a) Circuit



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

The timing diagram illustrates the state transitions of the counter. It shows two vertical blue lines representing the outputs Q₁ and Q₂. The Q₁ line has a change at cycle 2 and another at cycle 6. The Q₂ line has a change at cycle 4 and another at cycle 8. Arrows point from the table entries to these specific transition points on the lines.

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

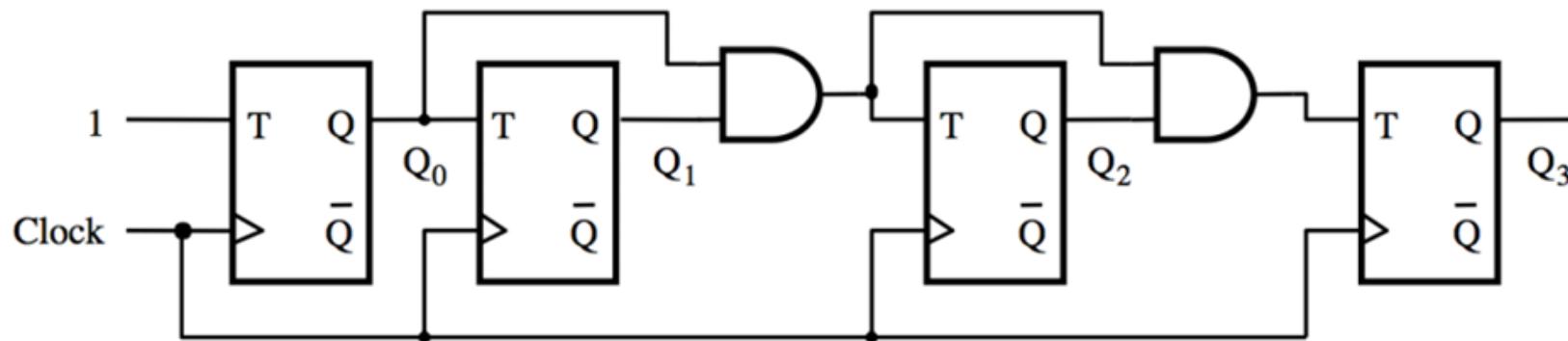
The timing diagram illustrates the state transitions of the counter. The vertical axis represents time, and the horizontal axis represents the states of Q₂, Q₁, and Q₀. Arrows point from the state changes in the table to the corresponding points on the timing diagram. The diagram shows that Q₁ changes at the 2nd, 4th, 6th, and 8th clock cycles, while Q₂ changes at the 4th, 6th, and 8th clock cycles.

$$T_0 = 1$$

$$T_1 = Q_0$$

$$T_2 = Q_0 Q_1$$

A four-bit synchronous up-counter



$$T_0 = 1$$

$$T_1 = Q_0$$

$$T_2 = Q_0 Q_1$$



In general we have

$$T_0 = 1$$

$$T_1 = Q_0$$

$$T_2 = Q_0 Q_1$$

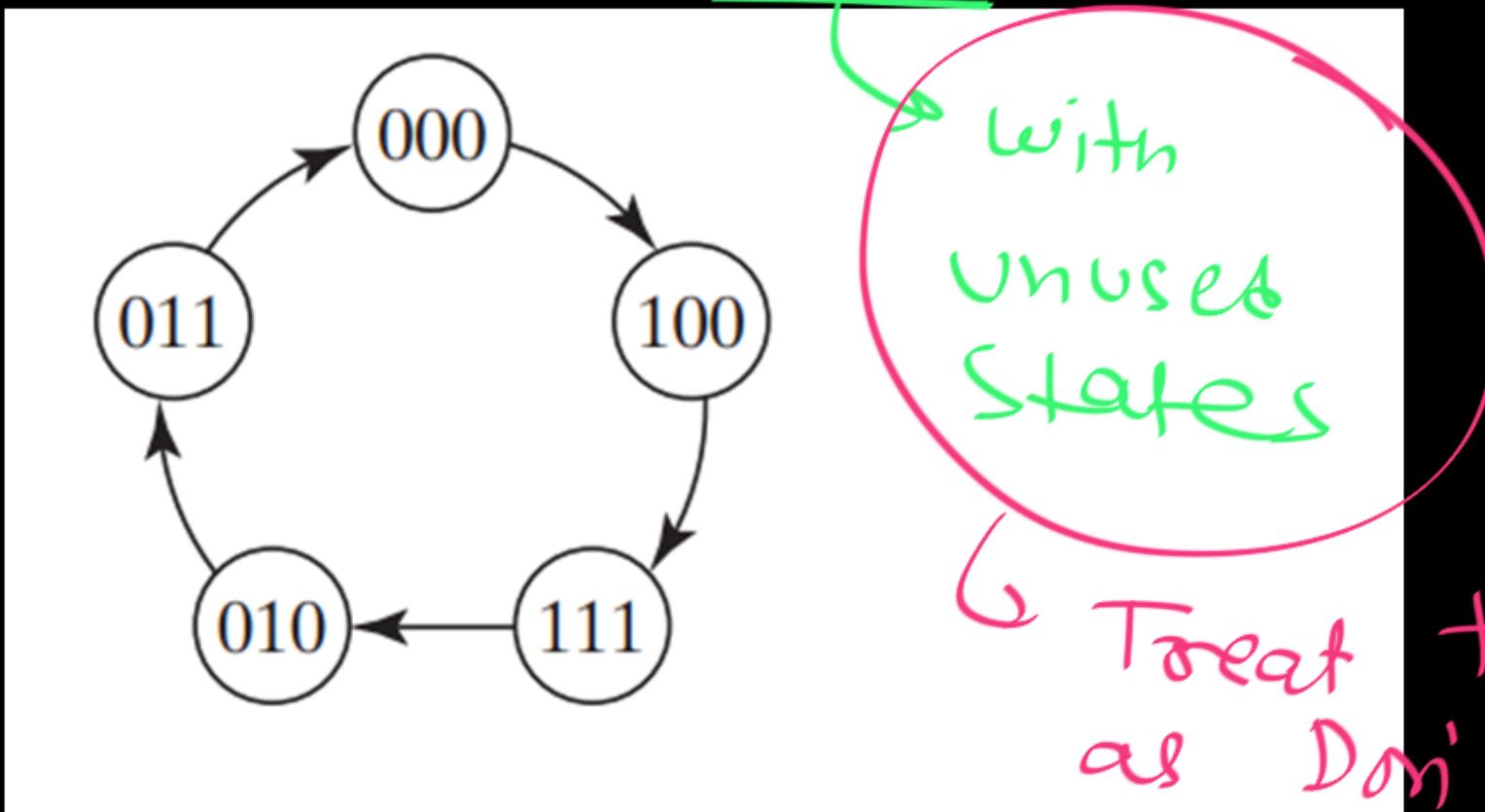
$$T_3 = Q_0 Q_1 Q_2$$

...

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

Q: — HW — Video Solution ✓ ✓

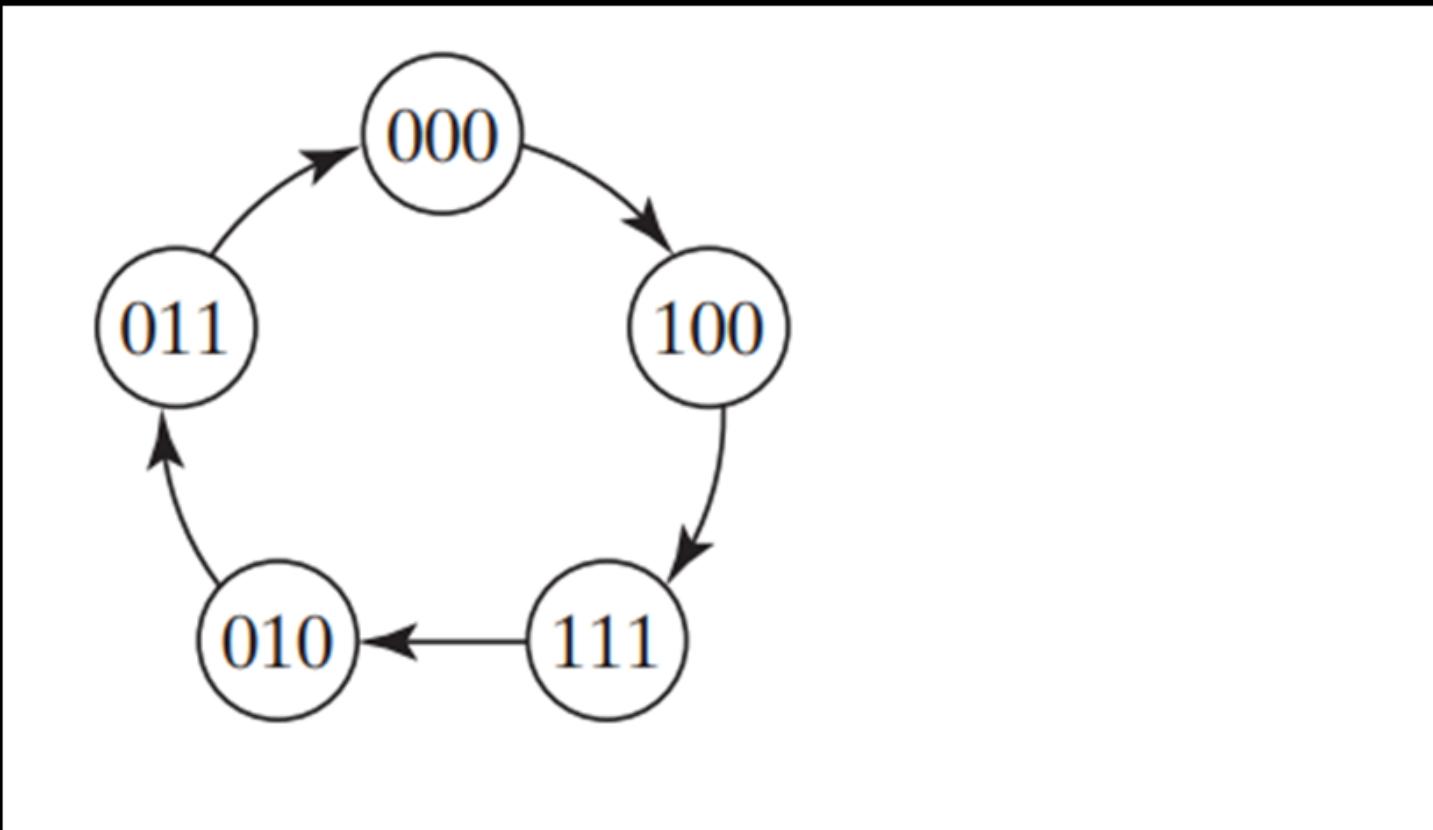
Design a the following counter using T-FFs.



Q:



Design a the following counter using D-FFs.



Q:

HW

Design a the following counter using JK-FFs.

