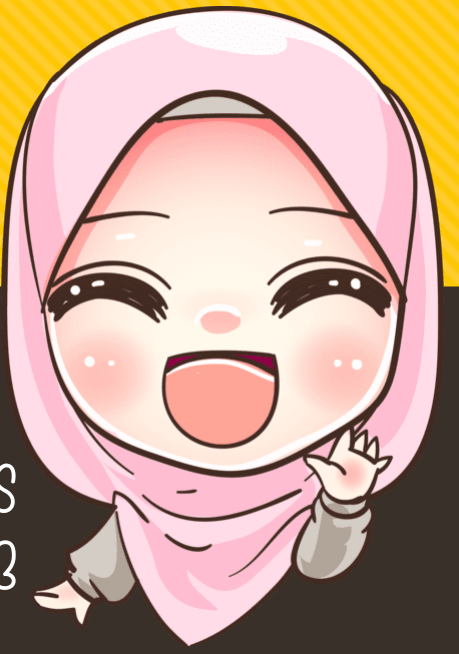
MMU
MULTIMEDIA UNIVERSITY

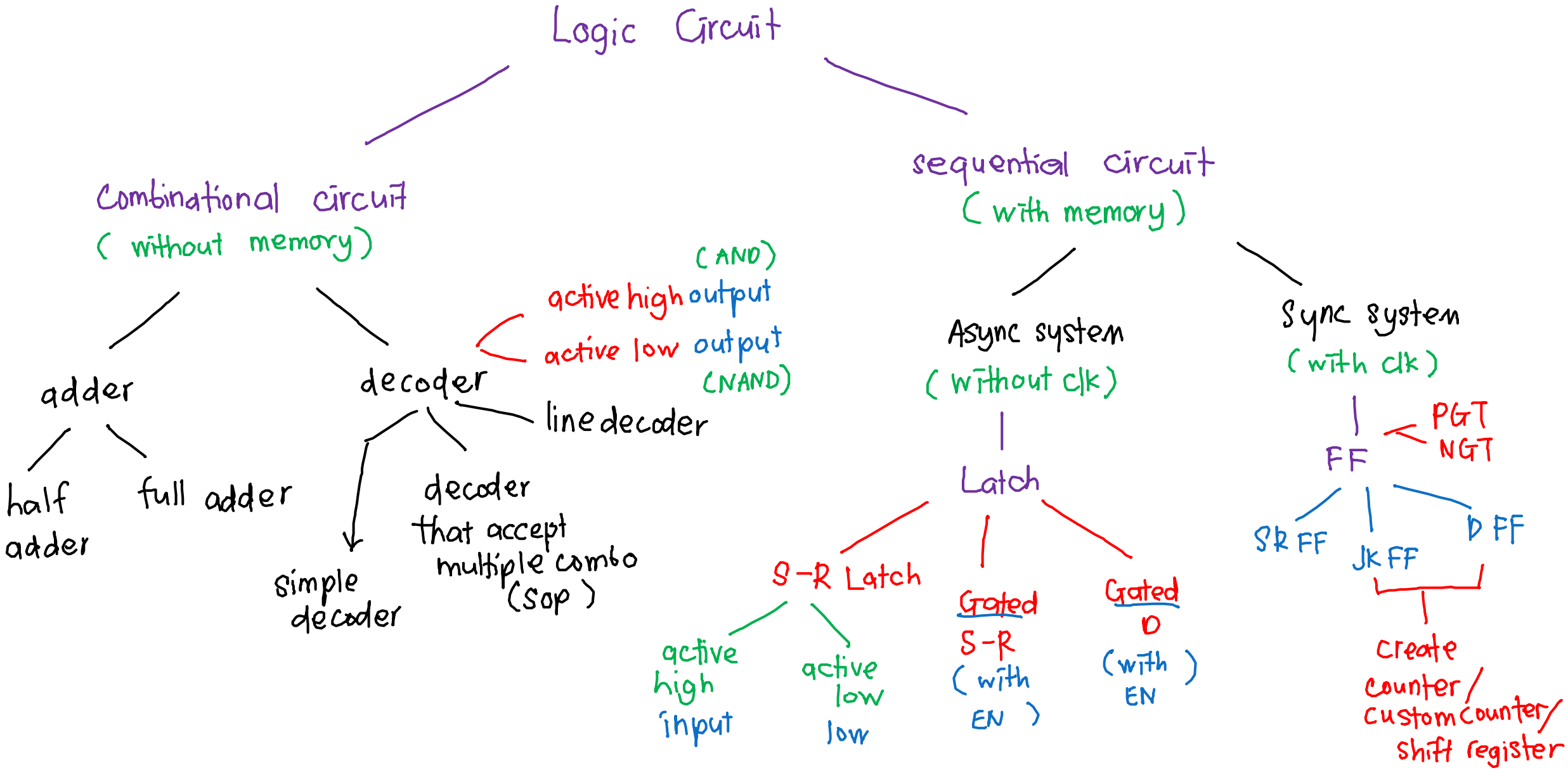
TUTORIAL 9 : COUNTER

APPLIED KNOWLEDGE QUESTIONS



PDS0101: INTRODUCTION TO DIGITAL SYSTEMS

TRI 2, 2022-2023

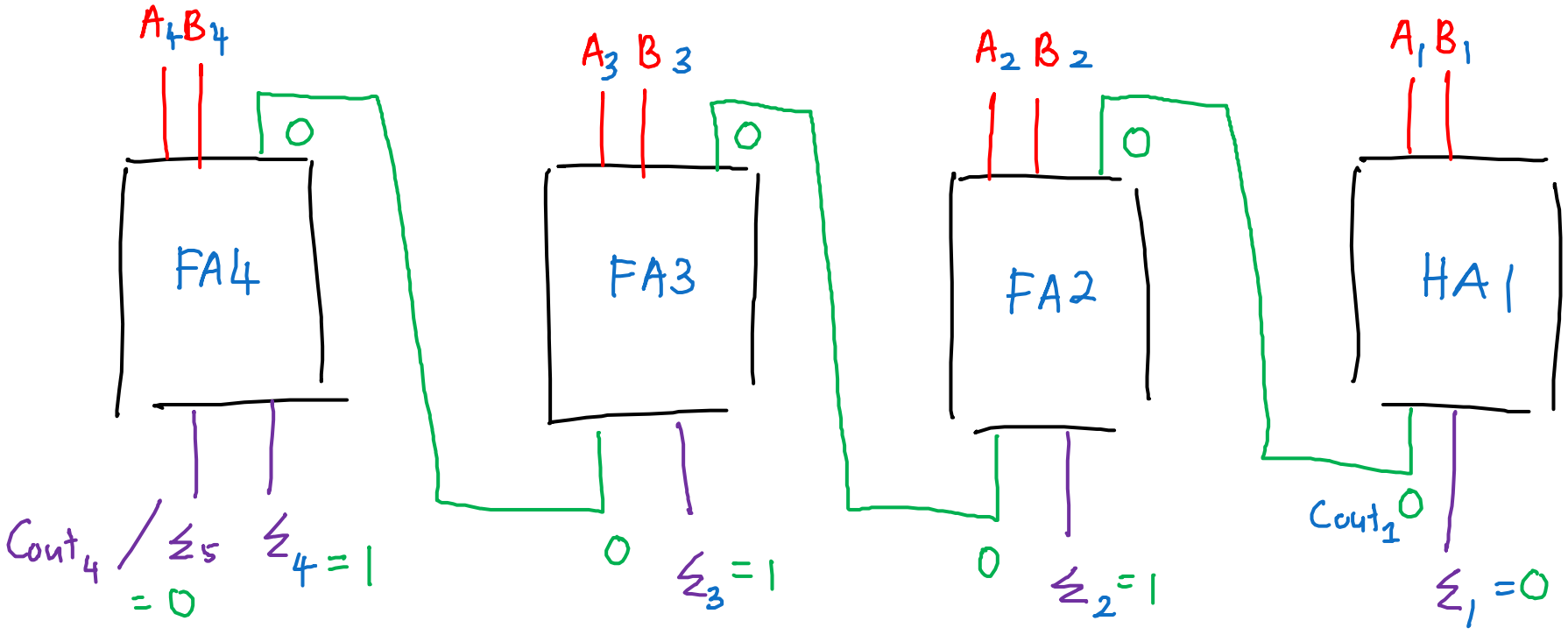

Logic Circuit
Combinational circuit
(without memory)
adder
half adder
full adder
decoder
(AND)
active high output
active low output
(NAND)
line decoder
simple decoder
decoder that accept multiple combo (sop)
sequential circuit
(with memory)
Async system
(without clk)
Latch
S-R Latch
active high input
active low low
Gated S-R (with EN)
Gated D (with EN)
Sync system
(with clk)
FF
PGT
NGT
SR FF
JK FF
D FF
create counter/ custom counter/ shift register

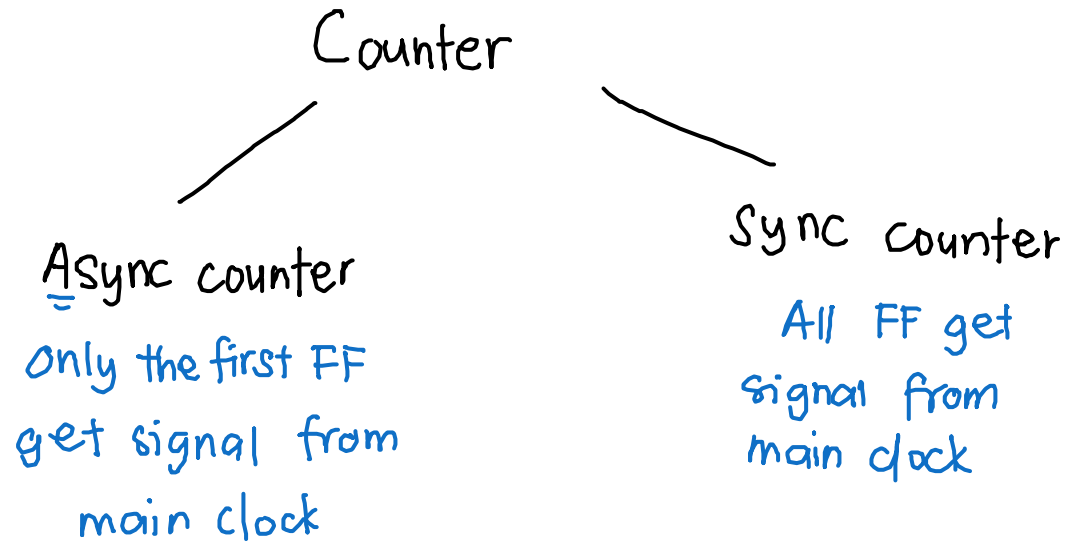
Adder

binary addition

Draw parallel adder for $4_{10} + 10_{10}$ using halfadder and full adder

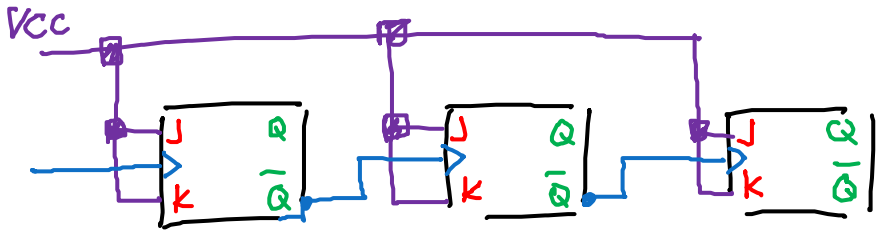
0100 1010



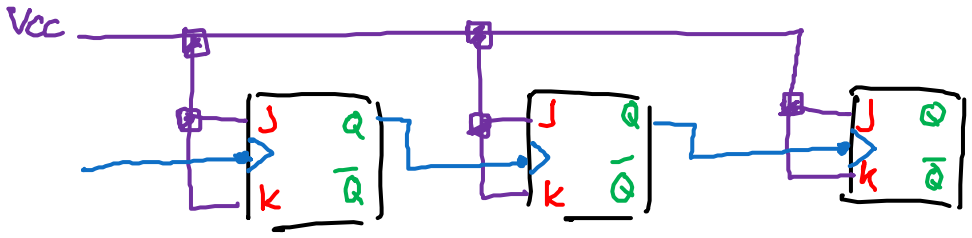

Counter
Async counter
Only the first FF get signal from main clock
Sync counter
All FF get signal from main clock

Async counter

PGT 3 bit Async Counter



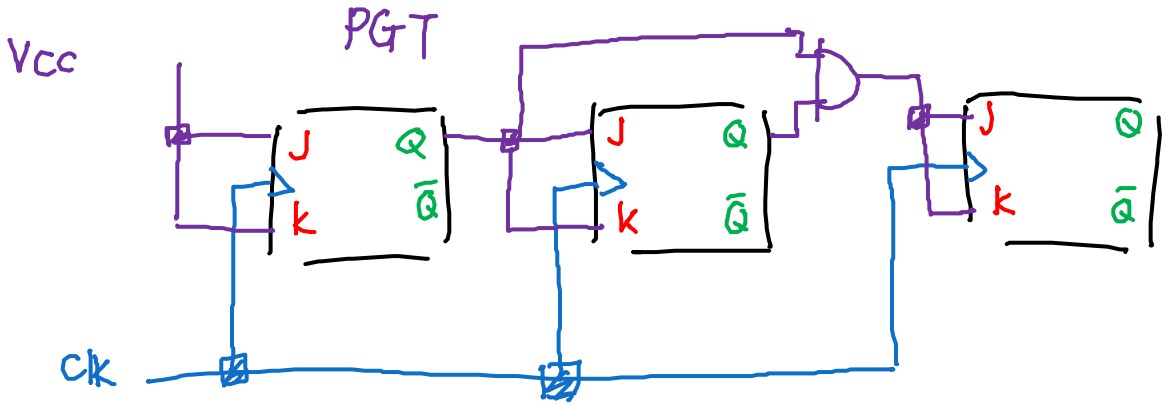
NGT 3 bit Asyn counter



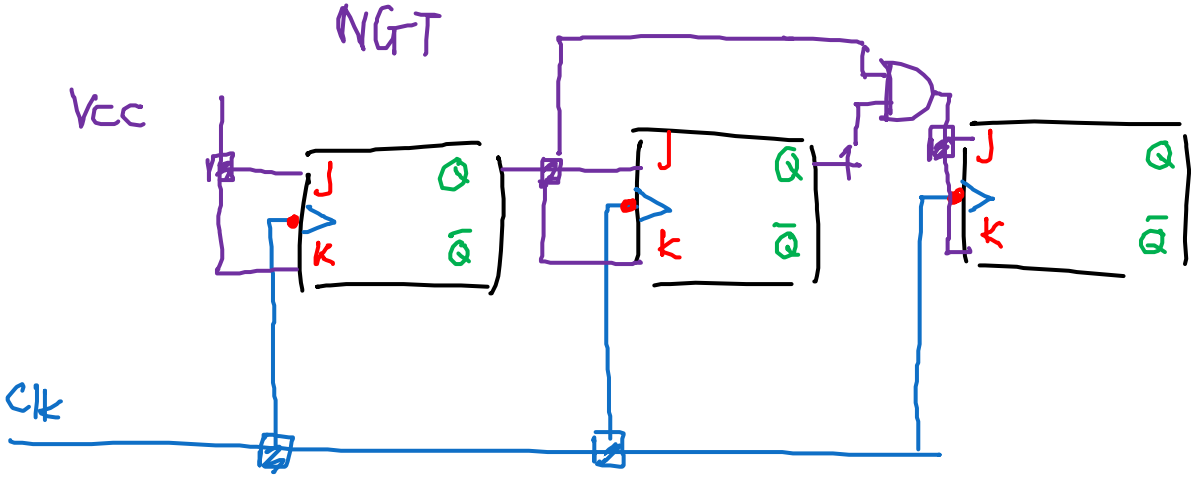
3 bit Counter

Sync Counter

PGT



NGT



PGT SR FF

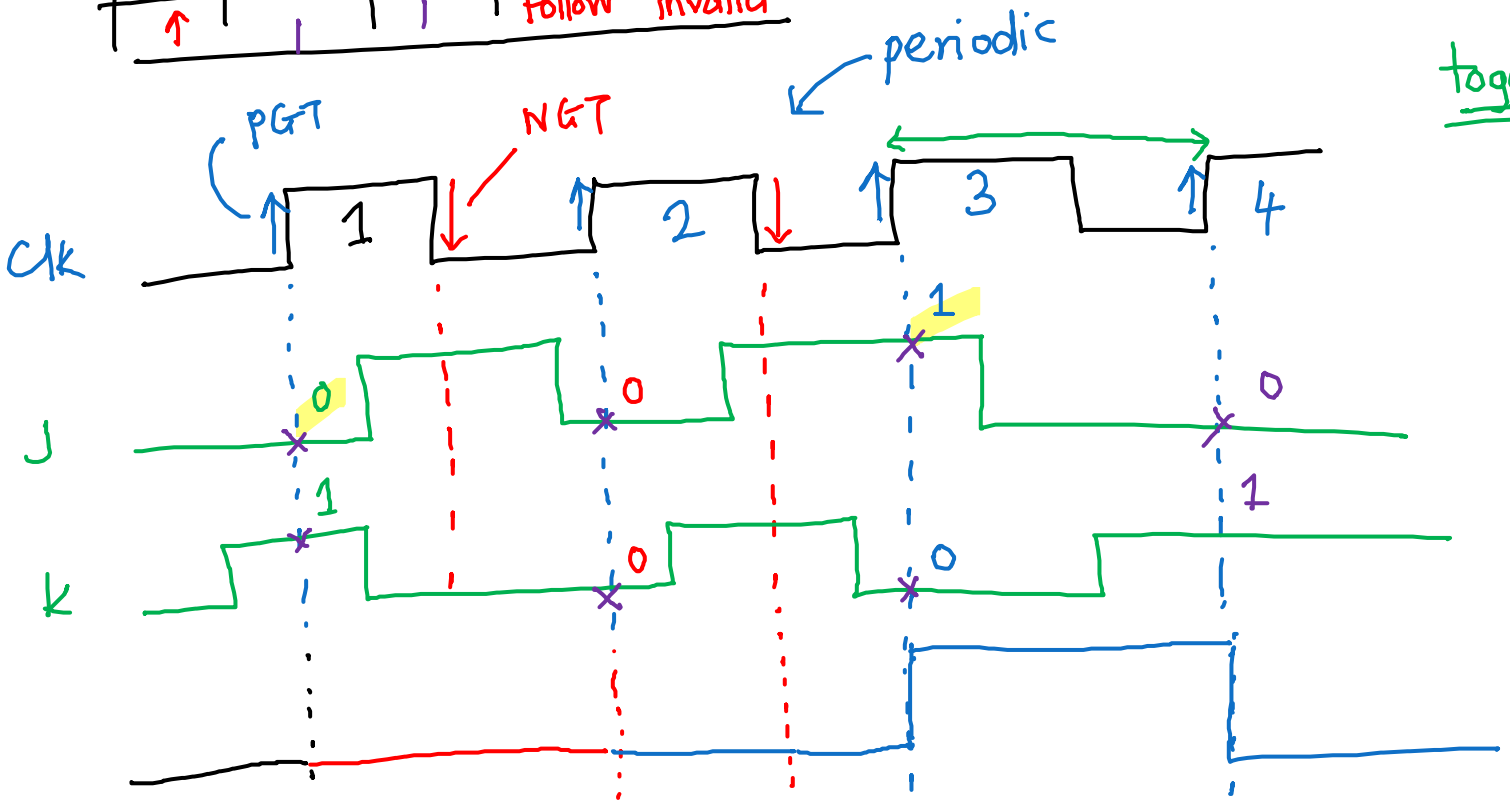
Q follows S when 01, 10

Clk	S	R	Output	State
↑	0	0	follow	Rest
↑	0	1	Q = 0	Reset
↑	1	0	Q = 1	set
↑	1	1	follow	invalid

NGT JK FF

Q follow J when 01, 10

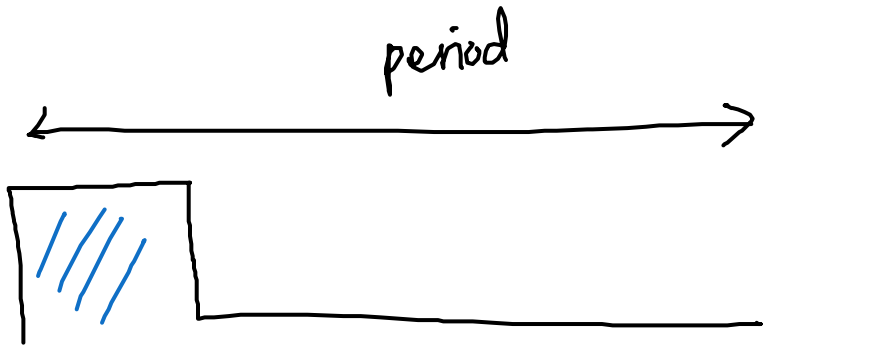
Clk	J	K	Output	State
↓	0	0	follow	Rest
↓	0	1	Q = 0	Reset
↓	1	0	Q = 1	Set
↓	1	1	toggle	set/reset



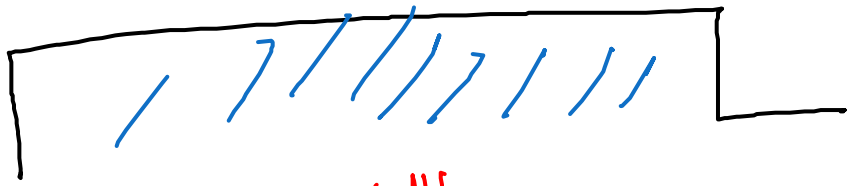
toggle 0 → 1 set

1 → 0 reset

Q = 0



duty cycle



pulse width

QUESTIONS 1

PGT
NGT

mod -12

Show how an **asynchronous counter** with a **modulus of 12** can be constructed using **flip-flops** (JK)

0 — 11 ; reset 12 (1100)
↓
1011

Mod-12 = how many FF needed??

Mod-12 requires **4 FFs** minimum to implement

$2^4 = 16$

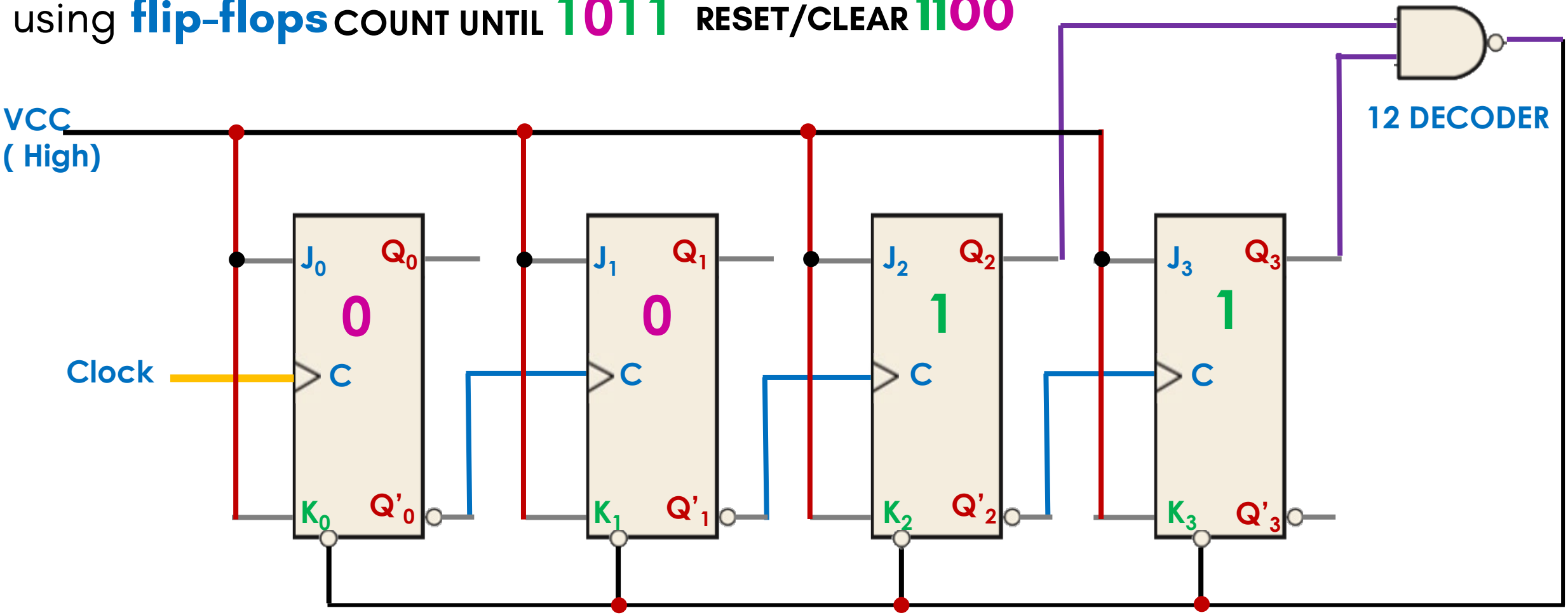
STATE	COUNT	Q_3	Q_2	Q_1	Q_0
1	0	0	0	0	0
2	1	0	0	0	1
3	2	0	0	1	0
4	3	0	0	1	1
5	4	0	1	0	0
6	5	0	1	0	1
7	6	0	1	1	0
8	7	0	1	1	1
9	8	1	0	0	0
10	9	1	0	0	1
11	10	1	0	1	0
12	11	1	0	1	1
13	12	1	1	0	0
14	13	1	1	0	1
15	14	1	1	1	0
16	15	1	1	1	1

Counter recycles HERE

CLEAR / RESET

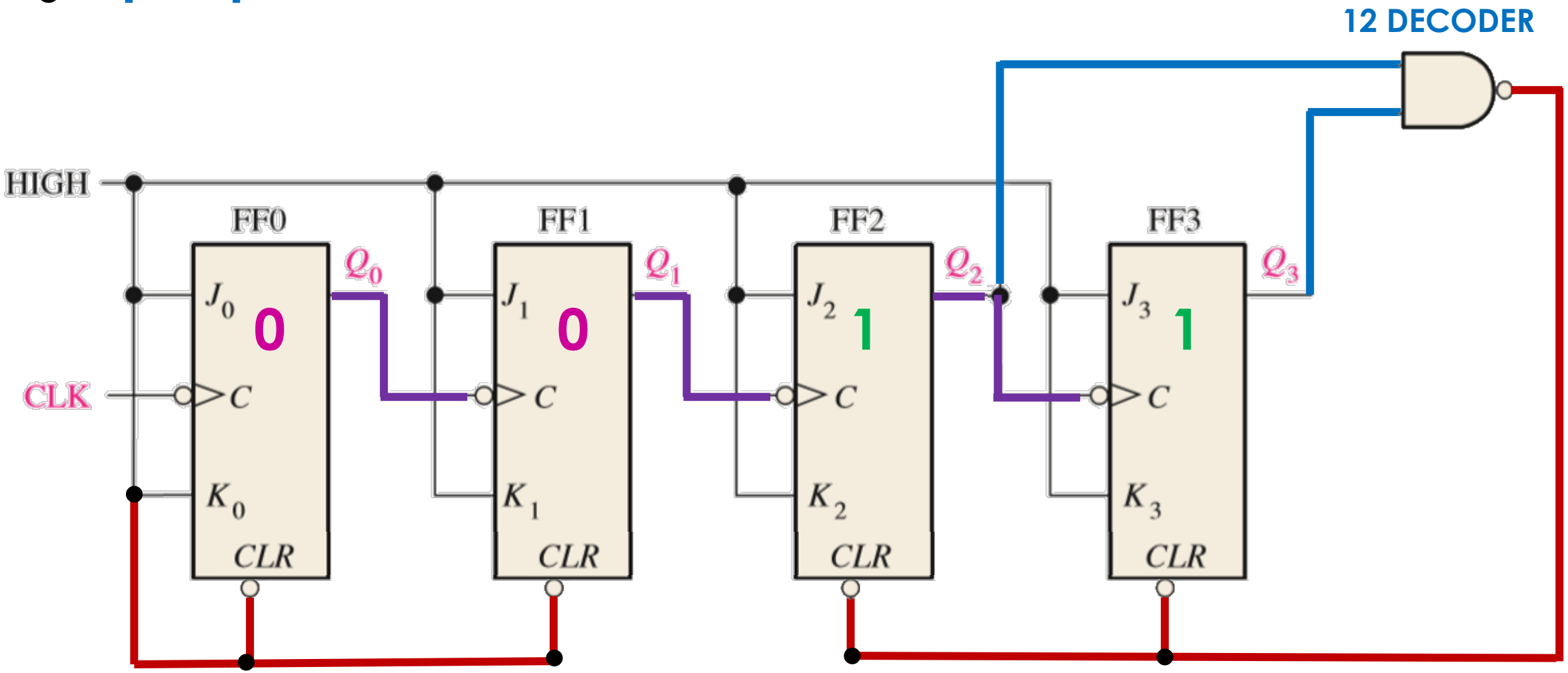
QUESTIONS 1

Show how an **asynchronous counter** with a **modulus of 12** can be constructed using **flip-flops** COUNT UNTIL **1011** RESET/CLEAR **1100**



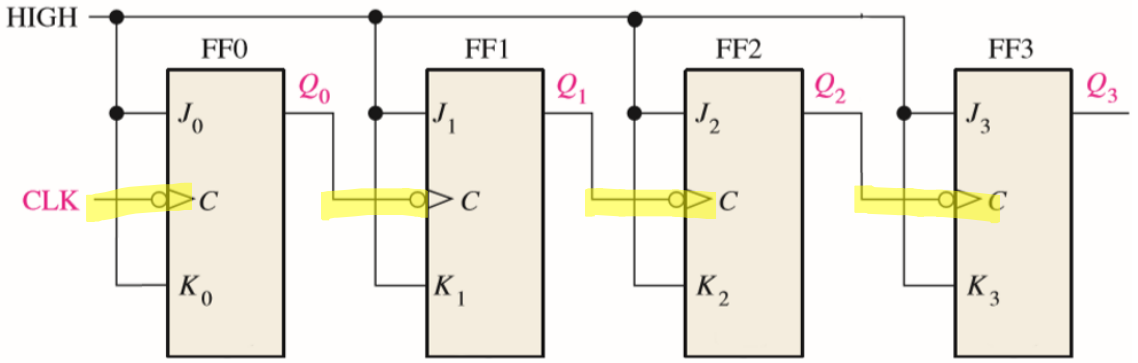
QUESTIONS 1

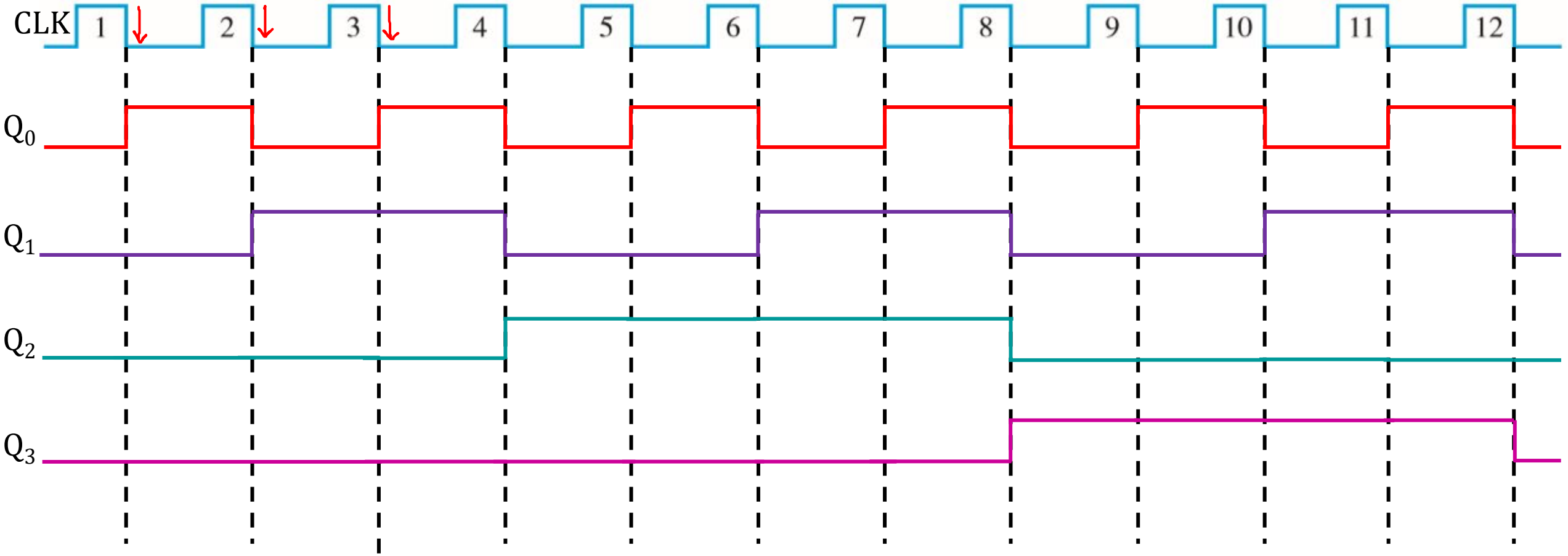
Show how an **asynchronous counter** with a **modulus of 12** can be constructed using **flip-flops** COUNT UNTIL 1011 RESET/CLEAR 1100



QUESTIONS 1

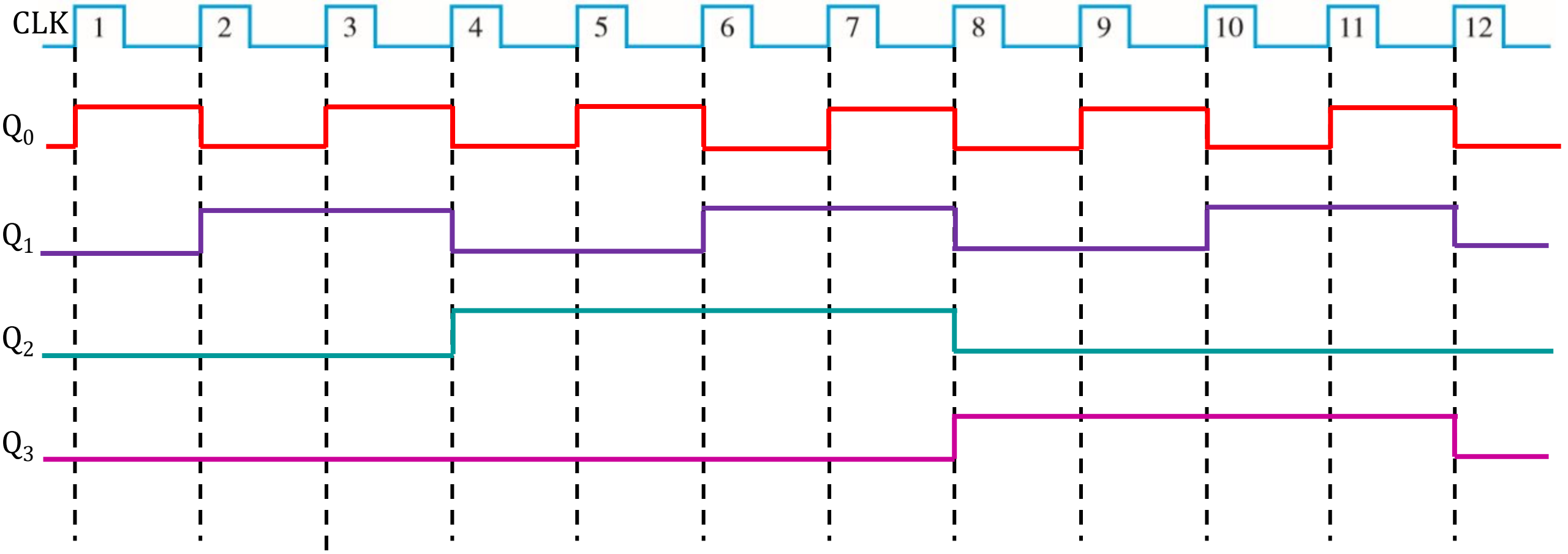
Show how an **asynchronous counter** with a **modulus of 12** can be constructed using **timing diagram**





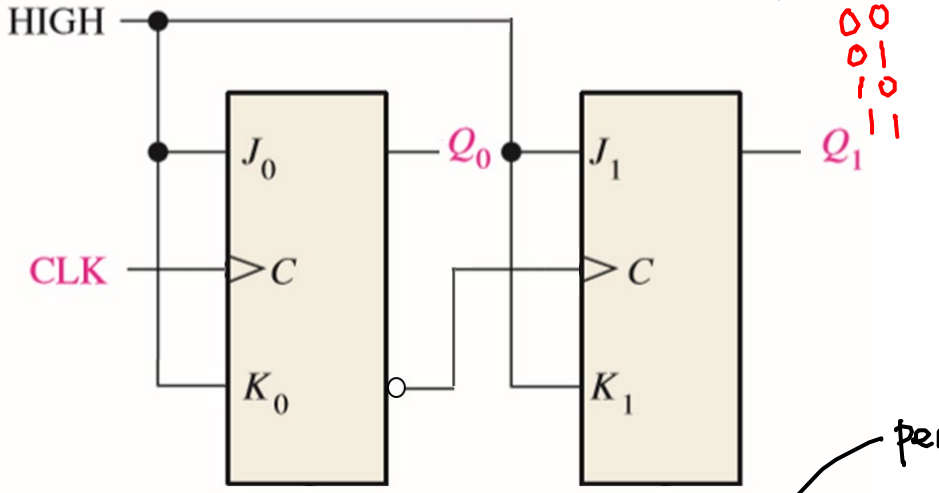
QUESTIONS 1

Show how an **asynchronous** **counter** with a **modulus of 12** can be constructed using **timing diagram**



QUESTIONS 2

For the **ripple counter** shown below, show the complete **timing diagram** for the **outputs** at Q_0 and Q_1 for **eight clock pulses**



2 bit
$2^2 = 4$
00
01
10
11

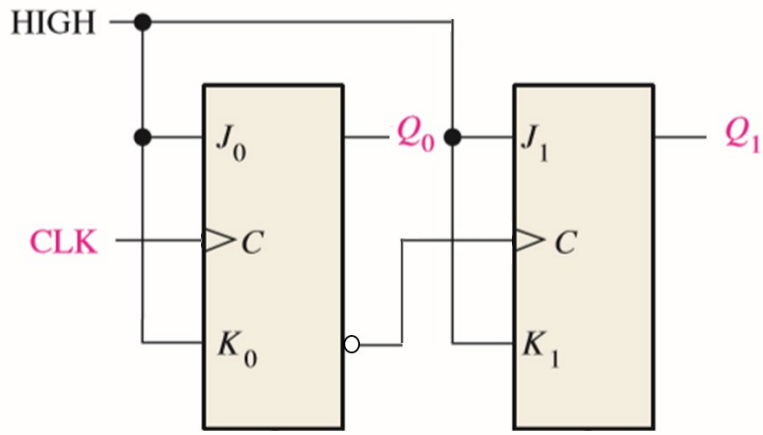
Clock Pulse	Q_1	Q_0
0	0	0
1	0	1
2	1	0
3	1	1
4	0	0
5	0	1
6	1	0
7	1	1
8	0	0

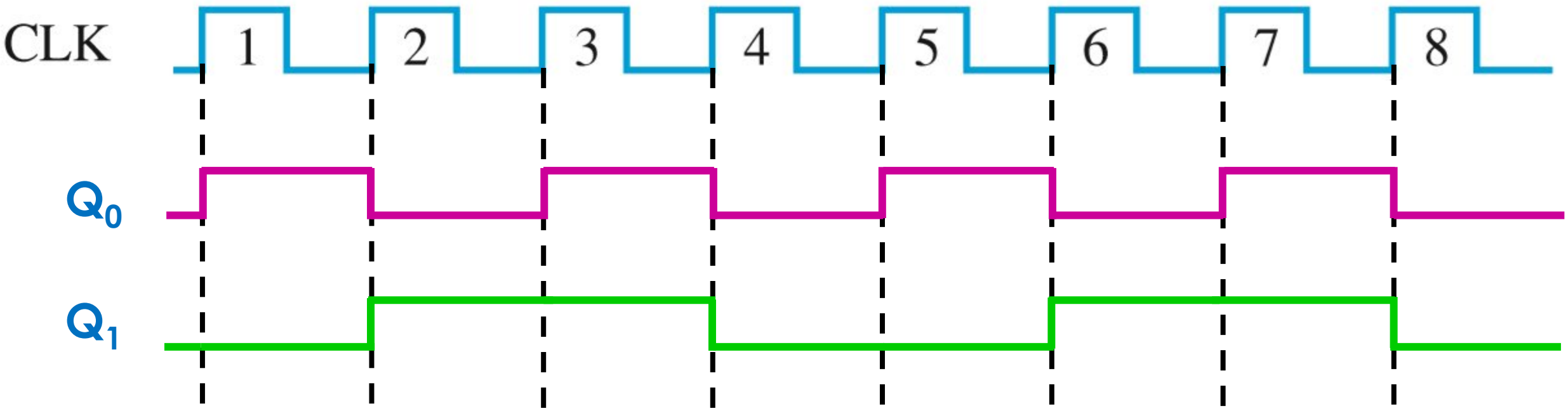
Clock Pulse	0	1	2	3	4	5	6	7	8
Q_0	0	1	0	1	0	1	0	1	0
Q_1	0	0	1	1	0	0	1	1	0

periodic

QUESTIONS 2

For the **ripple counter** shown below, show the complete **timing diagram** for the **outputs** at Q_0 and Q_1 for **eight clock pulses**

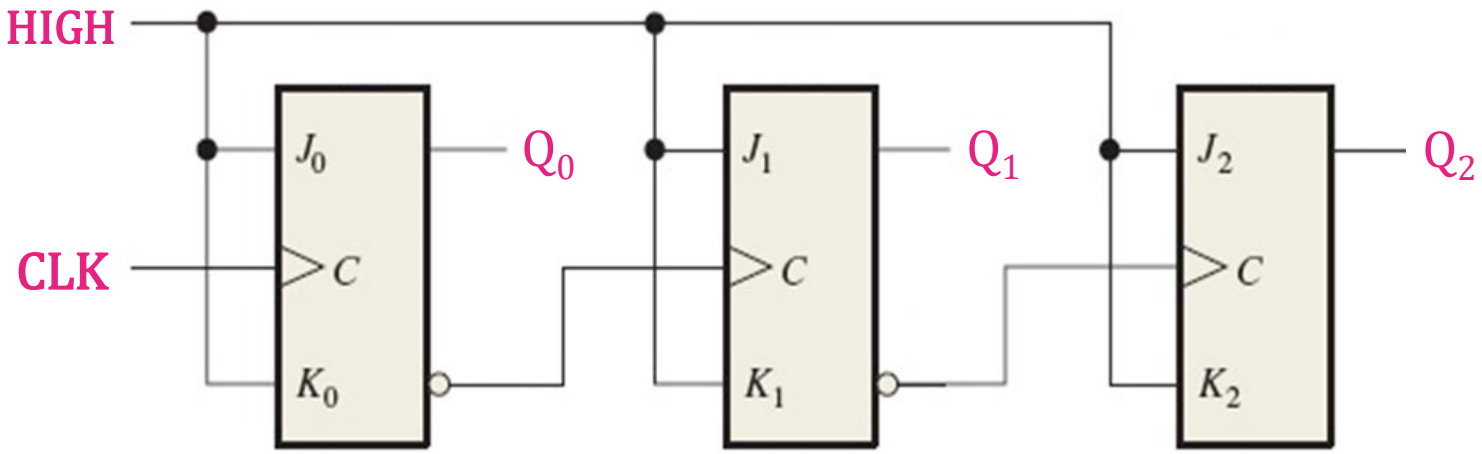




QUESTIONS 3

For the **counter** below, show the complete **timing diagram** for the **output** waveforms at Q_0, Q_1 and Q_2 for **sixteen clock pulses**

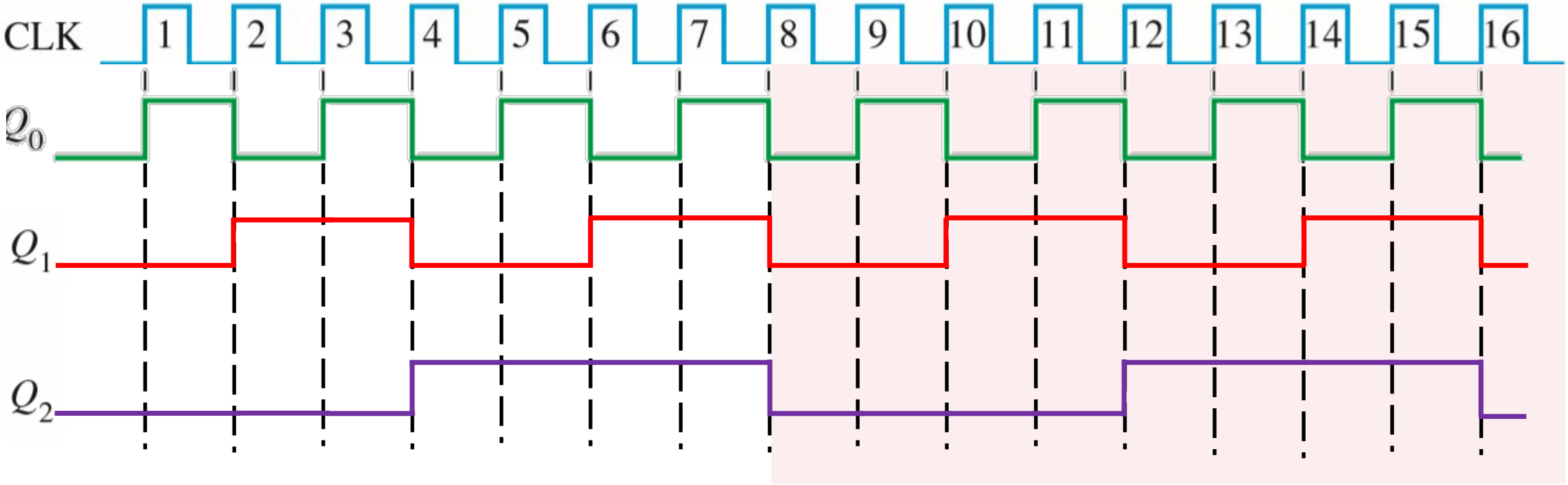
3 bit (000 – 111) $2^3 = 8$



Clock Pulse	Q_2	Q_1	Q_0
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			

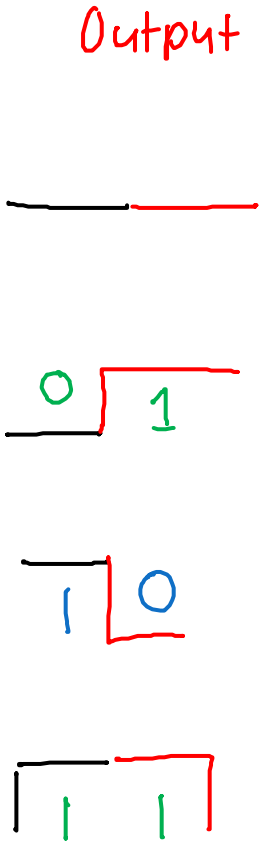
QUESTIONS 3

For the **counter** below, show the complete **timing diagram** for the **output** waveforms at **Q_0, Q_1** and **Q_2** for **sixteen clock pulses**



CURRENT - NEXT TABLE
(JK TRANSITION TABLE)

Output		Input	
Current	Next	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



Input

J = 0 ; K = 0
J = 0 ; K = 1

J = 0 ; K = X

J = 1 ; K = 0
J = 1 ; K = 1

J = 1 ; K = X

J = 0 ; K = 1
J = 1 ; K = 1

J = X ; K = 1

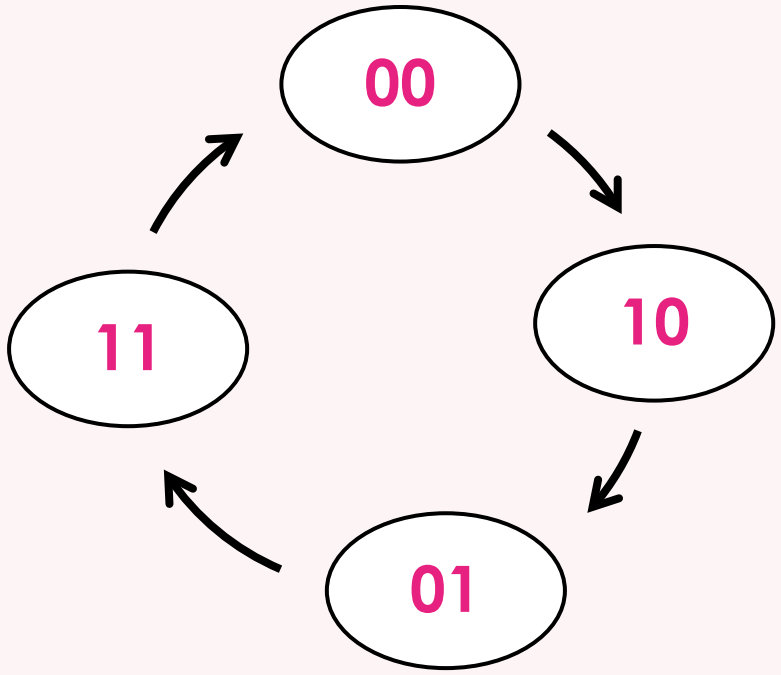
J = 1 ; K = 0
J = 0 ; J = 0

J = X ; K = 0

QUESTIONS 4

Design a **counter** to produce the following cyclic sequence 00 → 10 → 01 → 11

STEP 1: DRAW TRANSITION STATE DIAGRAM



STEP 2 : COMPLETE CURRENT-NEXT TABLE

CURRENT STATE		NEXT STATE	
A	B	A	B
0	0	1	0
1	0	0	1
0	1	1	1
1	1	0	0

QUESTIONS 4

Design a **counter** to produce the following cyclic sequence **00→10→01→11**

OUTPUT TRANSITIONS		FLIP-FLOP INPUTS	
Q_N	Q_{N+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

STEP 3 : COMPLETE JK PARTS IN THE TABLE

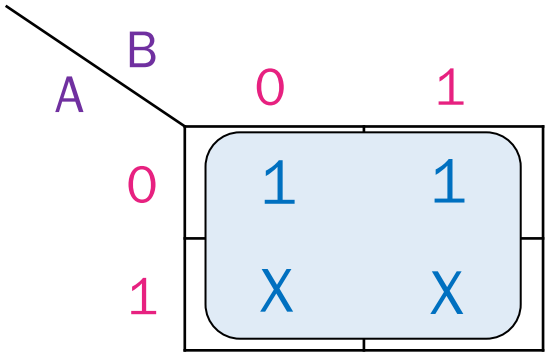
CURRENT STATE		NEXT STATE		JK INPUT			
A	B	A	B	J_A	K_A	J_B	K_B
0	0	1	0	1	X	0	X
1	0	0	1	X	1	1	X
0	1	1	1	1	X	X	0
1	1	0	0	X	1	X	1

QUESTIONS 4

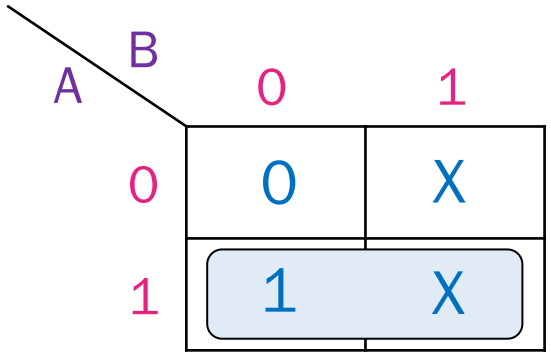
Design a **counter** to produce the following cyclic sequence **00→10→01→11**

STEP 4 : SIMPLIFY WITH K-MAP

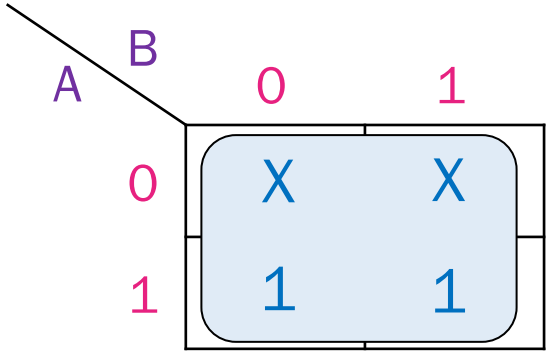
CURRENT STATE		JK INPUT			
A	B	J_A	K_A	J_B	K_B
0	0	1	X	0	X
1	0	X	1	1	X
0	1	1	X	X	0
1	1	X	1	X	1



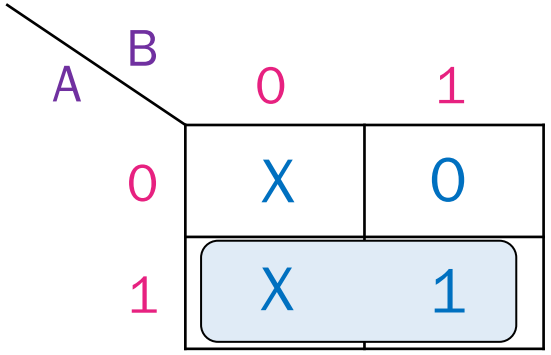
$J_A = 1$



$J_B = A$



$K_A = 1$

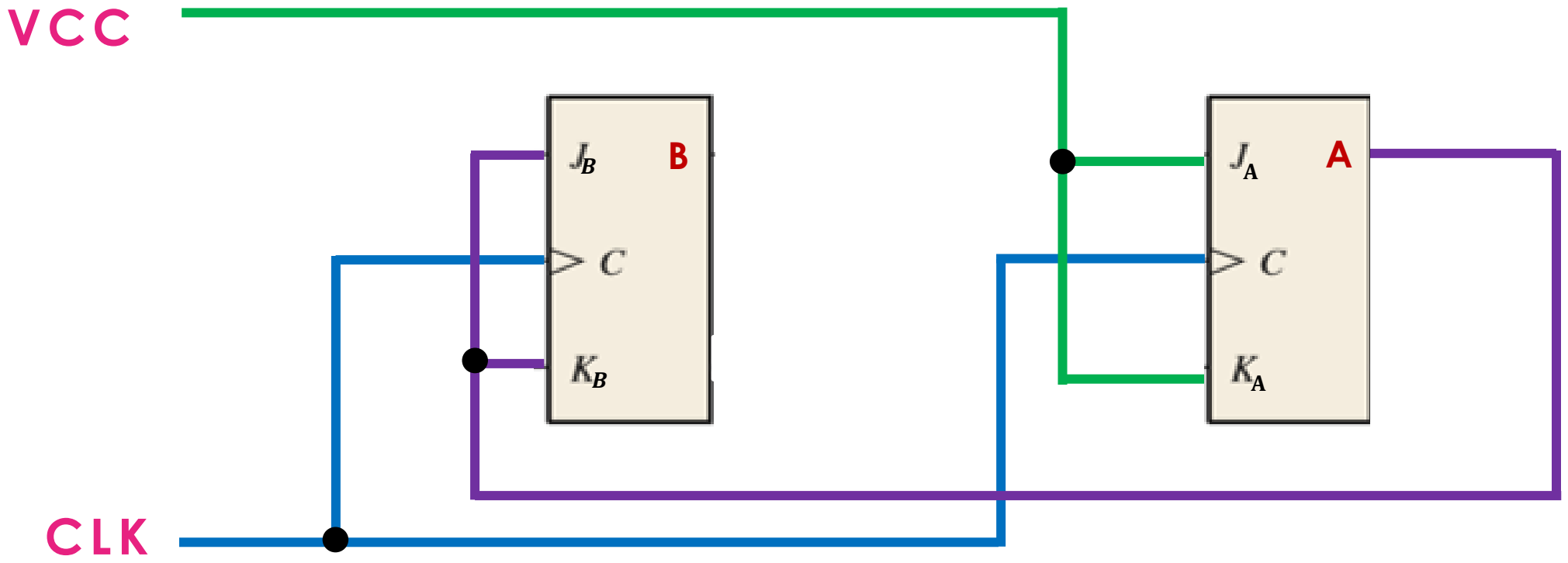


$K_B = A$

QUESTIONS 4

Design a **counter** to produce the following cyclic sequence **00→10→01→11**

STEP 5 : DRAW CIRCUIT

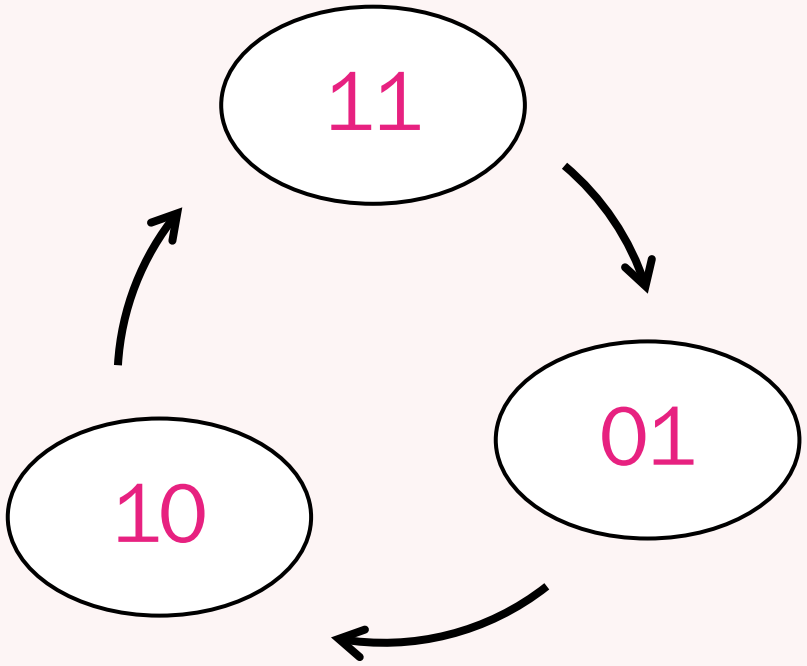


$J_B = A$ $K_B = A$ $J_A = 1$ $K_A = 1$

QUESTIONS 5

Alter the **counter** from (4) so that it only implements the stages **11** , **01** and **10** in cycle

STEP 1: DRAW TRANSITION STATE DIAGRAM



STEP 2 : COMPLETE CURRENT -NEXT TABLE

CURRENT STATE		NEXT STATE	
Q_1	Q_0	Q_1	Q_0
1	1	0	1
0	1	1	0
1	0	1	1
0	0	X	X

QUESTIONS 5

Alter the **counter** from (4) so that it only implements the stages **11** , **01** and **10** in cycle

OUTPUT TRANSITIONS		FLIP-FLOP INPUTS	
Q_N	Q_{N+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

STEP 3 : COMPLETE JK PARTS IN THE TABLE

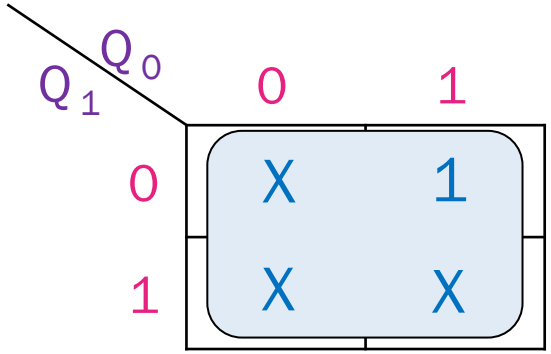
CURRENT STATE		NEXT STATE		JK INPUT			
Q_1	Q_0	Q_1	Q_0	J_1	K_1	J_0	K_0
1	1	0	1	X	1	X	0
0	1	1	0	1	X	X	1
1	0	1	1	X	0	1	X
0	0	X	X	X	X	X	X

QUESTIONS 5

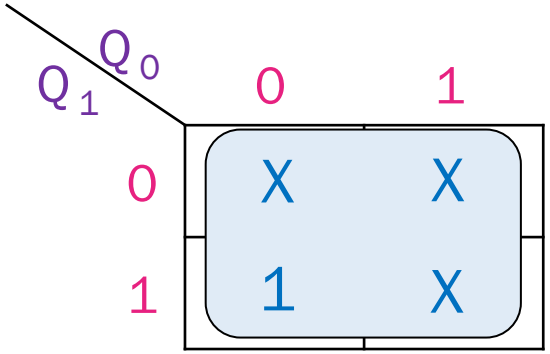
Alter the **counter** from (4) so that it only implements the stages **11** , **01** and **10** in cycle

STEP 4 : SIMPLIFY WITH K-MAP

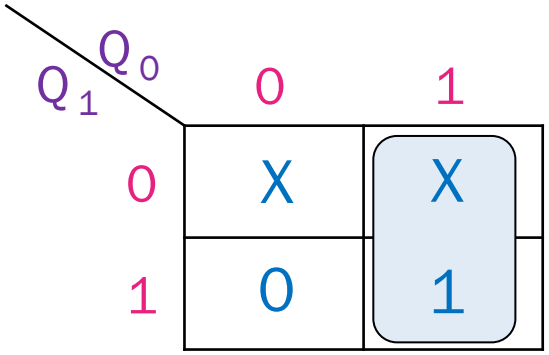
CURRENT STATE		JK INPUT			
Q_1	Q_0	J_1	K_1	J_0	K_0
1	1	X	1	X	0
0	1	1	X	X	1
1	0	X	0	1	X
0	0	X	X	X	X



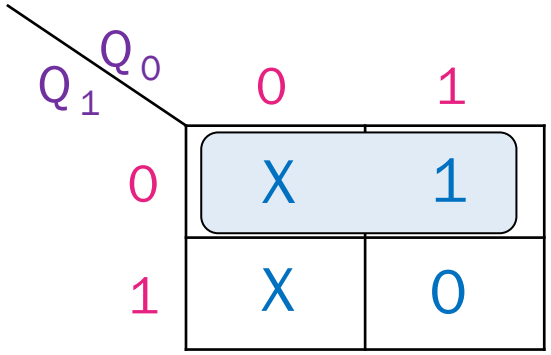
$J_1 = 1$



$J_0 = 1$



$K_1 = Q_0$

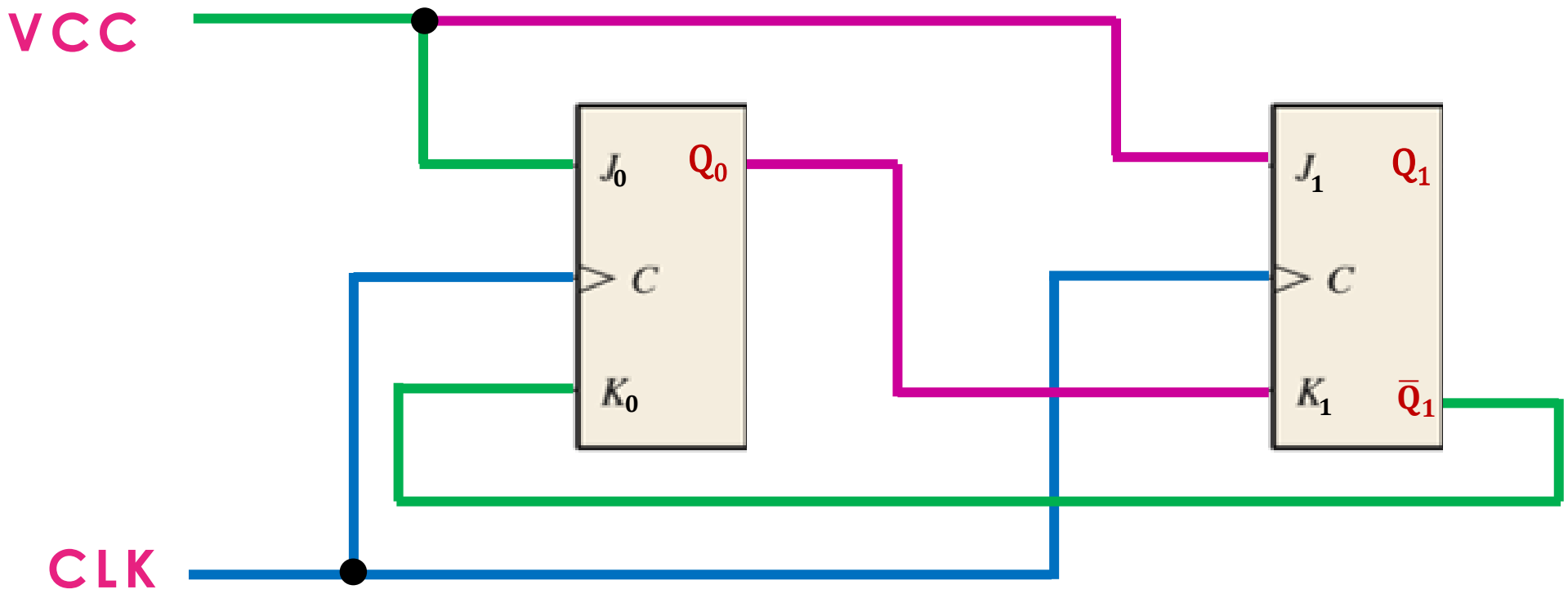


$K_0 = \overline{Q}_1$

QUESTIONS 5

Alter the **counter** from (4) so that it only implements the stages **11** , **01** and **10** in cycle

STEP 5 : DRAW CIRCUIT

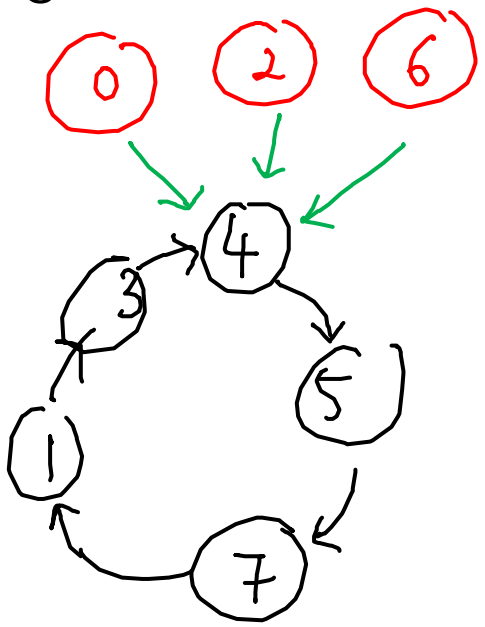


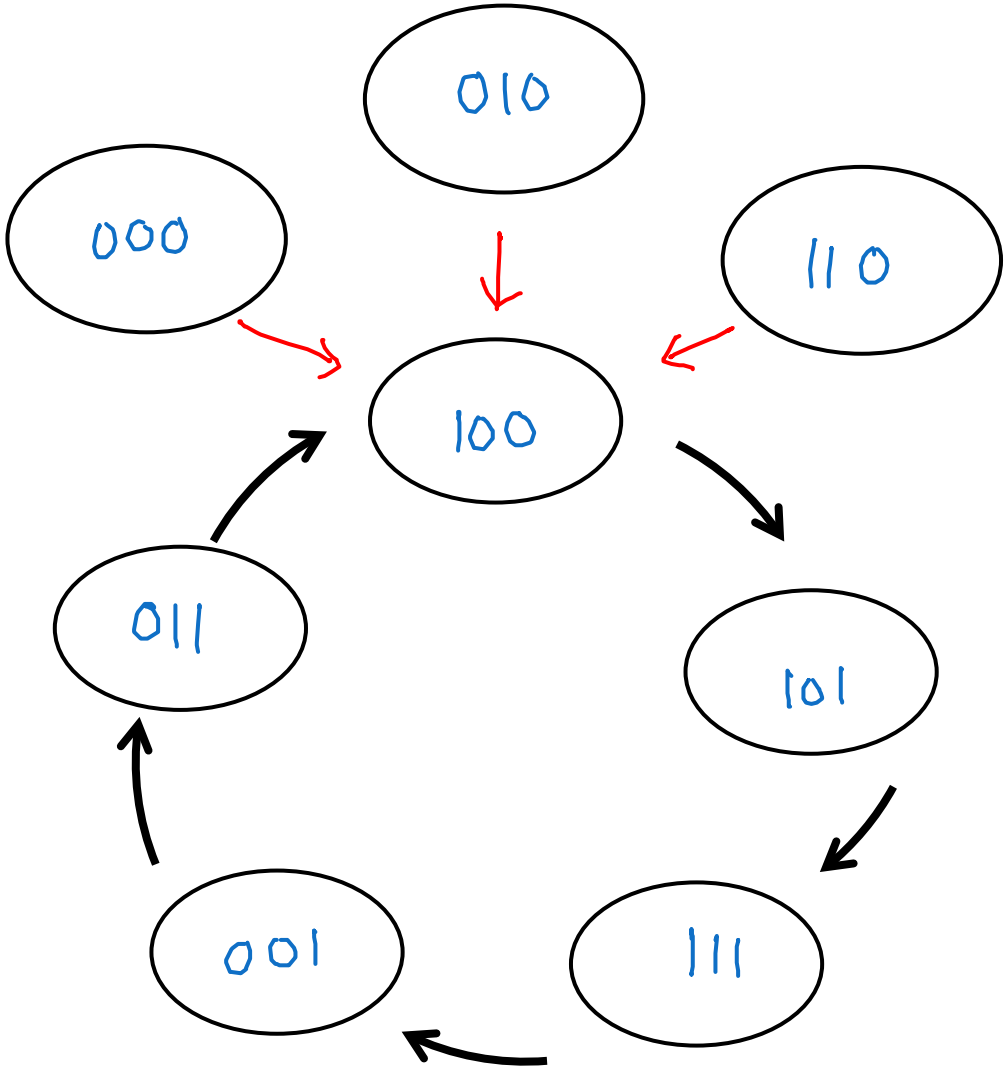
$J_0 = 1$ $K_0 = \bar{Q}_1$ $J_1 = 1$ $K_1 = Q_0$

APPLIED KNOWLEDGE QUESTIONS

7. Design a counter using **J-K FFs** that follow the cyclic sequence **4,5,7,1,3**. Find the minimum number of FFs required to implement the counter and **any unwanted FF** sequences in the counter go to **4**

Decimal	Binary		
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



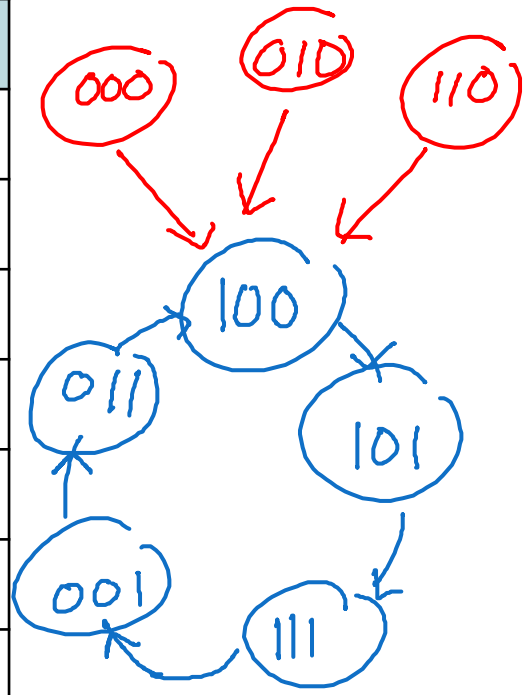


STEP 2 : COMPLETE CURRENT-NEXT TABLE
STEP 3 : COMPLETE JK PARTS IN THE TABLE

OUTPUT TRANSITIONS		FLIP-FLOP INPUTS	
Q_N	Q_{N+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Present

CURRENT STATE			NEXT STATE			FF2		FF1		FF0	
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	1	0	0	1	X	0	X	0	X
0	0	1	0	1	1	0	X	1	X	X	0
0	1	0	1	0	0	1	X	X	1	0	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	1	X	0	1	X	X	0
1	1	0	1	0	0	X	0	X	1	0	X
1	1	1	0	0	1	X	1	X	1	X	0



100 101
101 111
111 001

STEP 4 : SIMPLIFY WITH K-MAP

CURRENT STATE			FF2	
A	B	C	J_A	K_A
0	0	0	1	X
0	0	1	0	X
0	1	0	1	X
0	1	1	1	X
1	0	0	X	0
1	0	1	X	0
1	1	0	X	0
1	1	1	X	1

J_A

AB \ C	0	1
00	1	0
01	1	1
11	X	X
10	X	X

$J_A = \overline{C} + B$

K_A

AB \ C	0	1
00	X	X
01	X	X
11	0	1
10	0	0

$K_A = BC$

STEP 4 : SIMPLIFY WITH K-MAP

CURRENT STATE			FF1	
A	B	C	J_B	K_B
0	0	0	0	X
0	0	1	1	X
0	1	0	X	1
0	1	1	X	1
1	0	0	0	X
1	0	1	1	X
1	1	0	X	1
1	1	1	X	1

J_B

AB \ C	0	1
00	0	1
01	X	X
11	X	X
10	0	1

$J_B = C$

K_B

AB \ C	0	1
00	X	X
01	1	1
11	1	1
10	X	X

$K_B = 1$

STEP 4 : SIMPLIFY WITH K-MAP

CURRENT STATE			FF0	
A	B	C	J_C	K_C
0	0	0	0	X
0	0	1	X	0
0	1	0	0	X
0	1	1	X	1
1	0	0	1	X
1	0	1	X	0
1	1	0	0	X
1	1	1	X	0

Jc

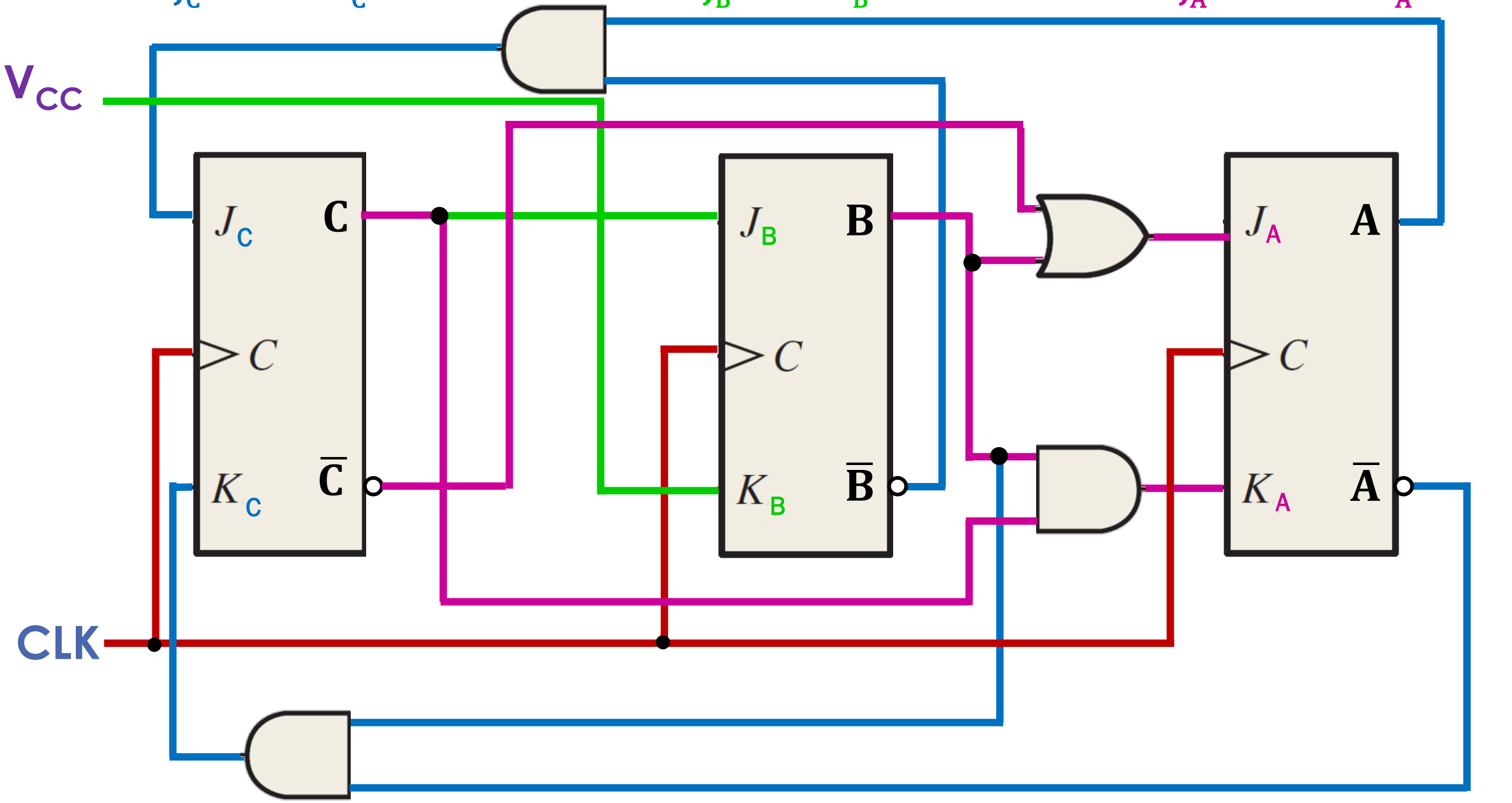
AB \ C	0	1
00	0	X
01	0	X
11	0	X
10	1	X

$J_C = A\bar{B}$

Kc

AB \ C	0	1
00	X	0
01	X	1
11	X	0
10	X	0

$K_C = \bar{A}B$


$J_C = A\bar{B}$ $K_C = \bar{A}B$
$J_B = C$ $K_B = 1$
$J_A = B + \bar{C}$ $K_A = BC$
V_{CC}
J_C
C
C
$\bar{C}$
K_C
J_B
B
C
K_B
$\bar{B}$
J_A
A
C
K_A
$\bar{A}$
CLK