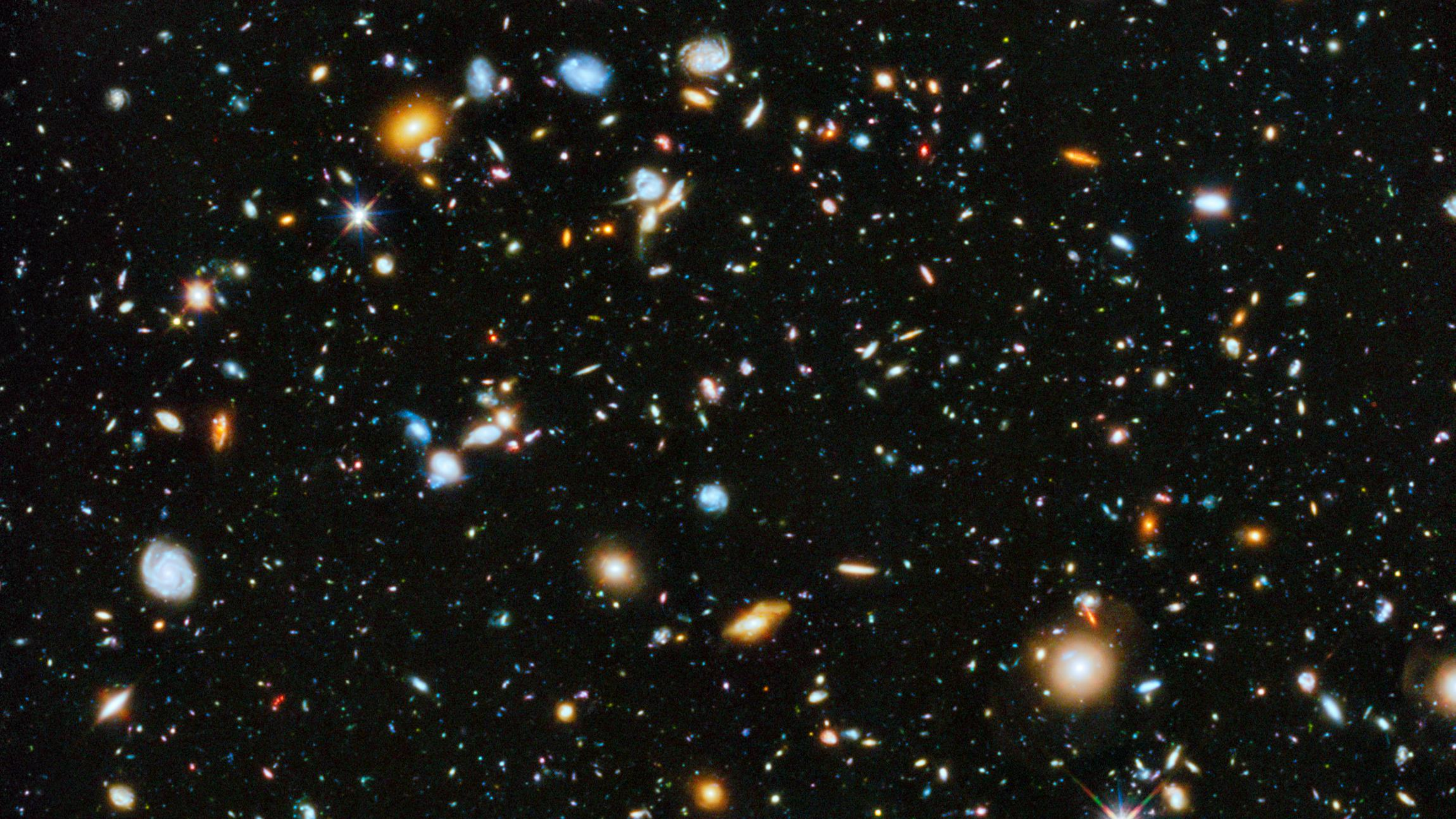
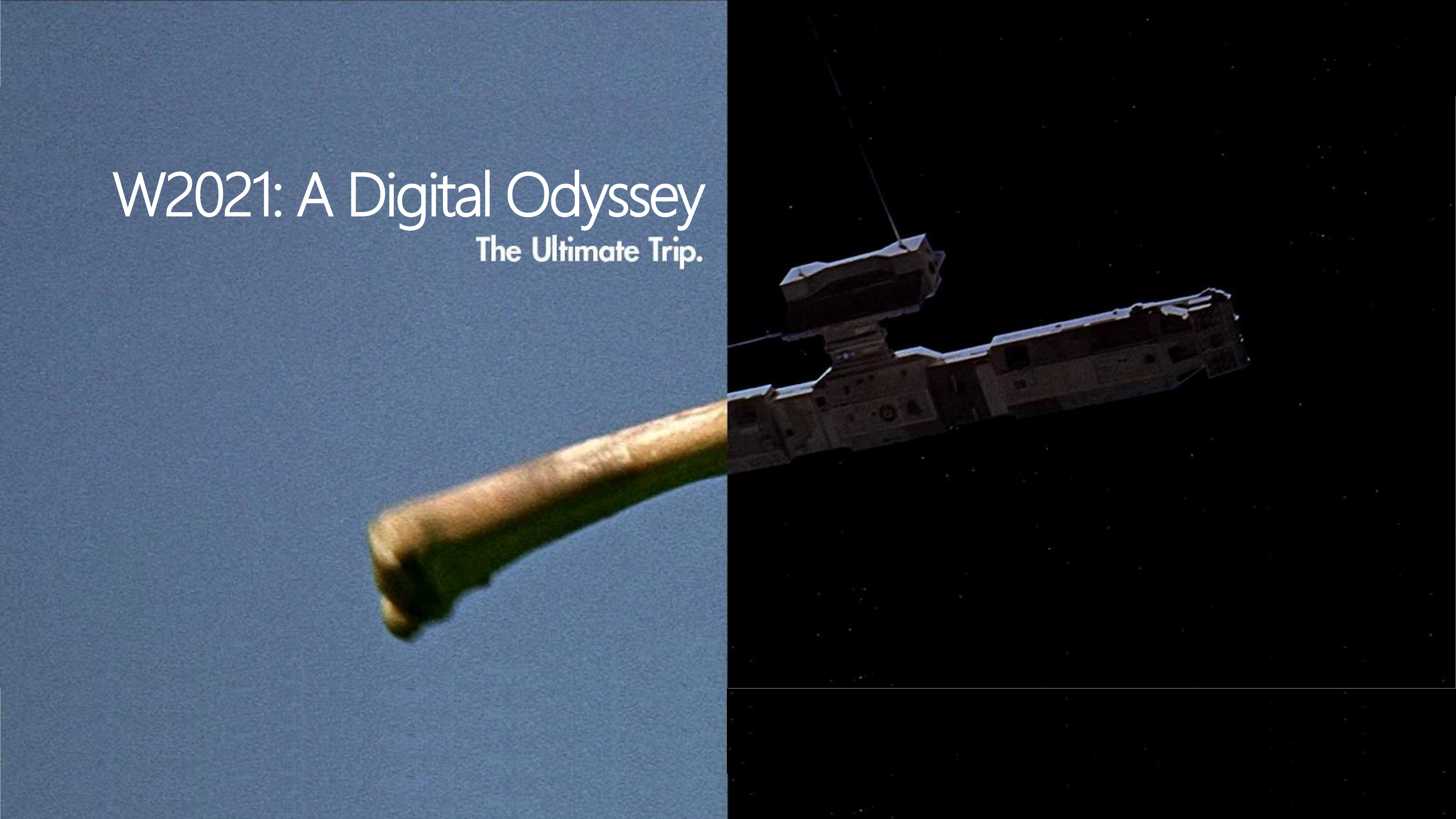
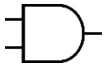
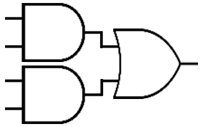
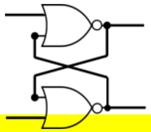


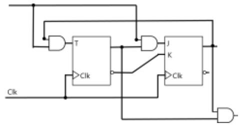

W2021: A Digital Odyssey
The Ultimate Trip.

Number Systems | $(12)_{10} \rightarrow (1100)_2$

| Logic Gates | 

| Combinational Logic | 

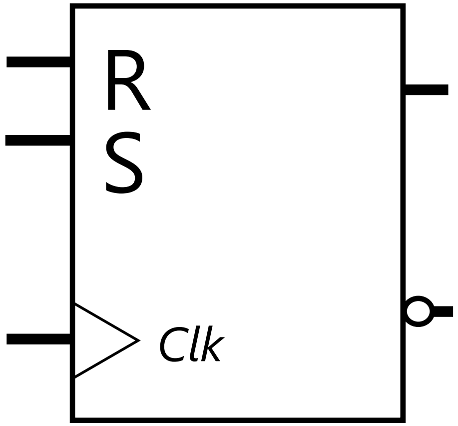
| Flip-Flops | 

| Sequential Logic | 

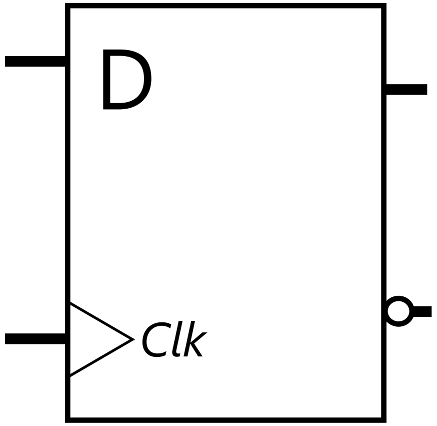
| Computer Systems

Flip-Flop

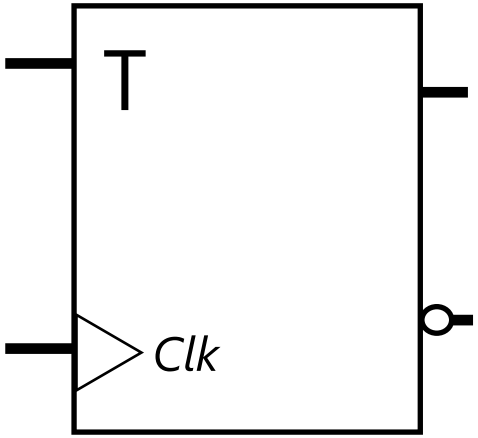
A single edge triggered latch



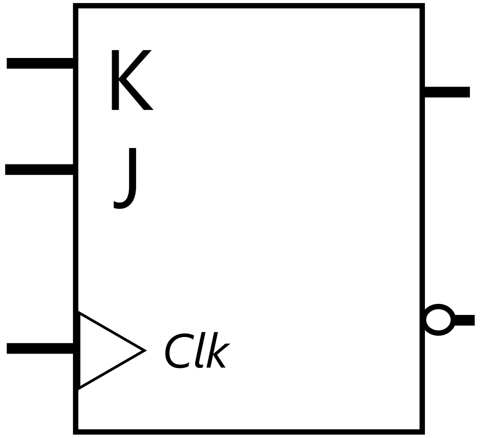
S	R	Q
0	0	Q_t
0	1	0
1	0	1
1	1	X



D	Q
0	0
1	1

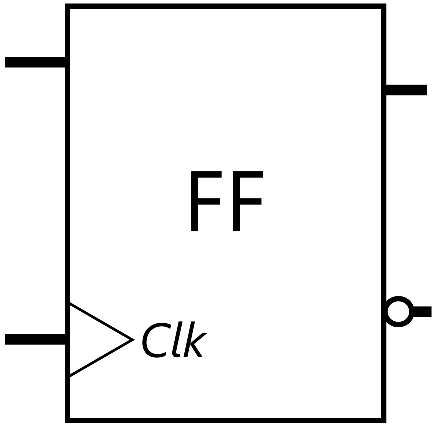


T	Q
0	Q_t
1	Q'_t



J	K	Q
0	0	Q_t
0	1	0
1	0	1
1	1	Q'_t

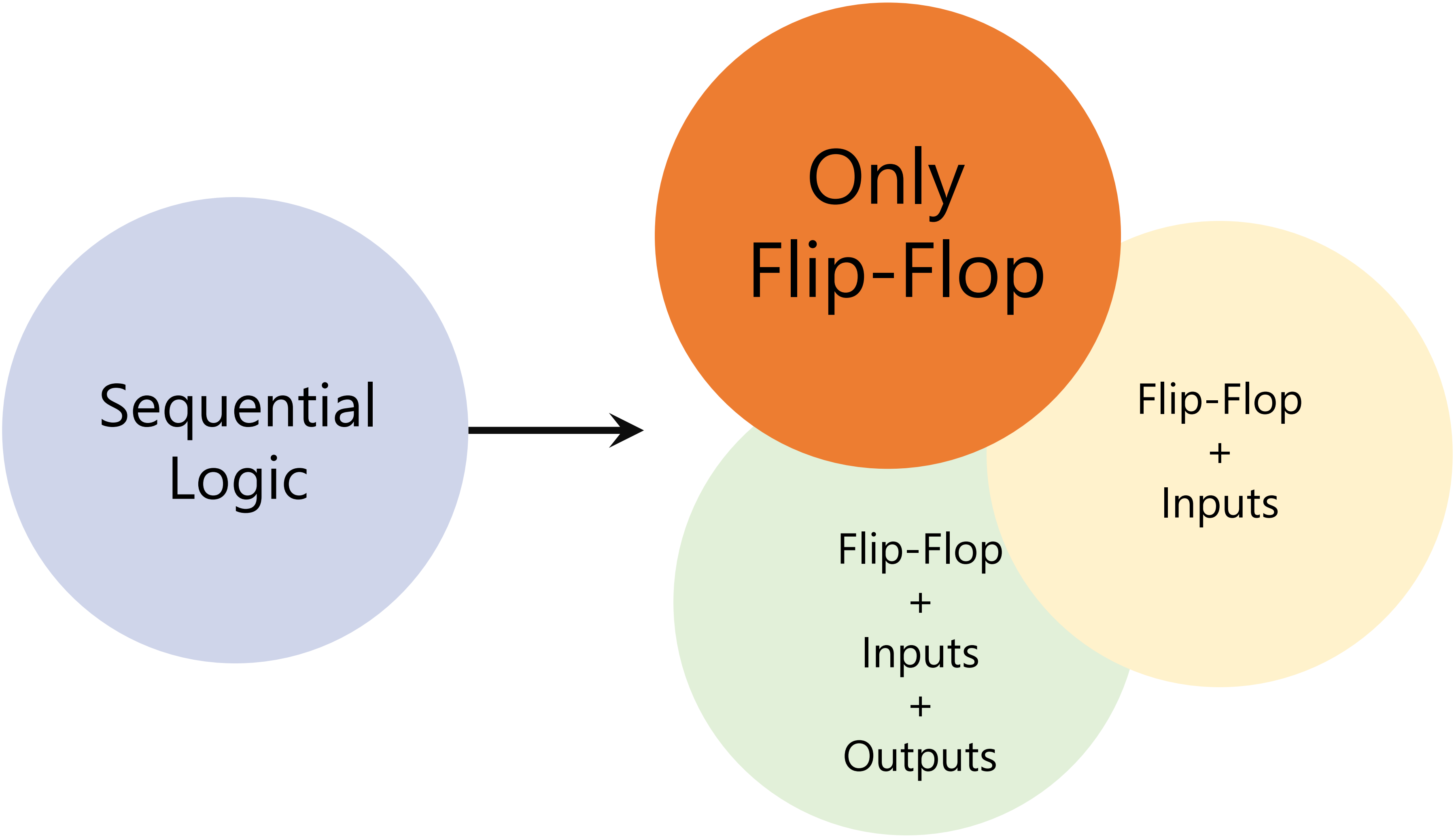
Single Edge
Positive



Analysis: Given a sequential circuit, show the behavior

vs.

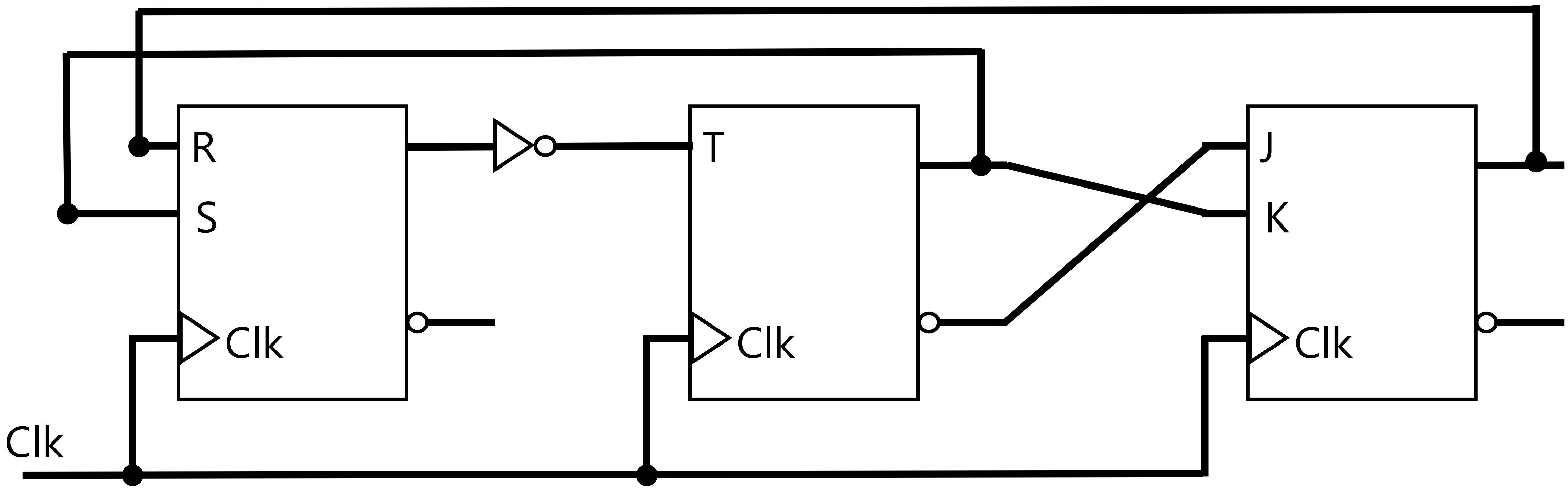
Design: Given a behavior, build the sequential circuit


Sequential
Logic
Only
Flip-Flop
Flip-Flop
+
Inputs
Flip-Flop
+
Inputs
+
Outputs

Analysis (Recap)

0. Is the circuit sequential or combinational? Any FF or feedback → Sequential
1. What are the flip-flops? RS, D, T, JK, or mixed (e.g., 2 JK, 1 RS, …)
2. What are the state combinations? $2^{\#FF}$
3. Form "State" table:
   a) Columns: for each FF, two columns:
      - one for current state,
      - one for next state
   b) Rows: for each state combination
      - In total: $2^{\#FF}$
4. Fill the state table for next state columns based on:
   a) the current state
   b) the inputs to the FFs
5. Form State Transition Diagram
6. (Optional) Analyze paths and states in state transition diagram

Analysis
by an example


R
S
Clk
T
Clk
J
K
Clk
Clk

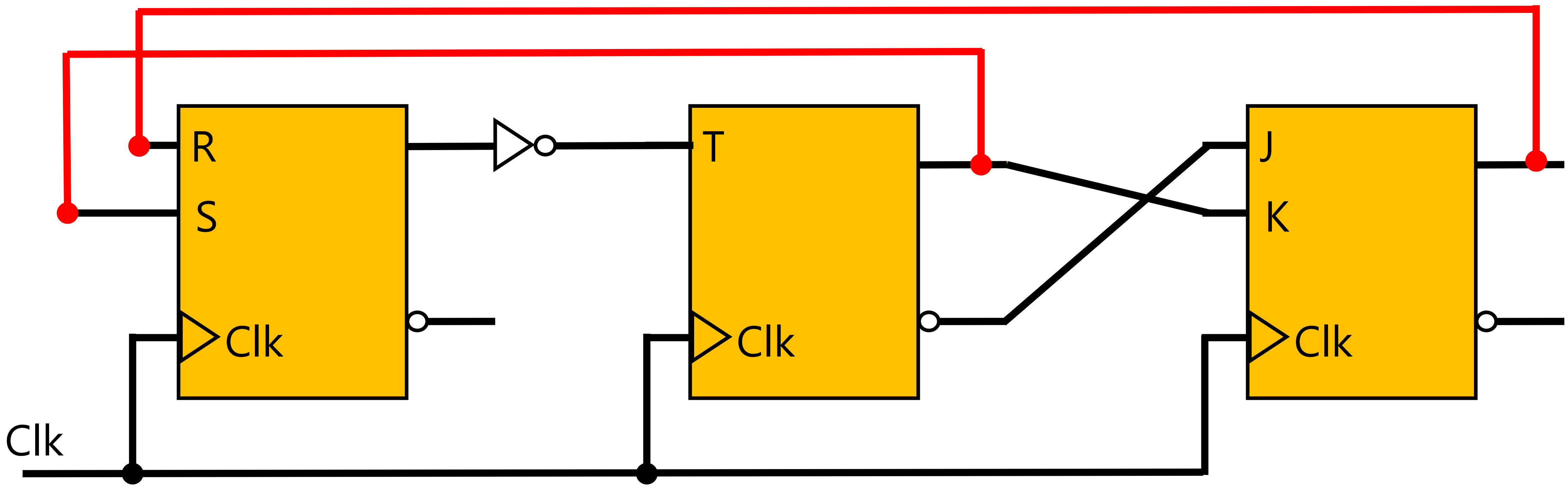
Analysis

0) Is it sequential circuit?

At least one FF → Yes

At least one feedback → Yes

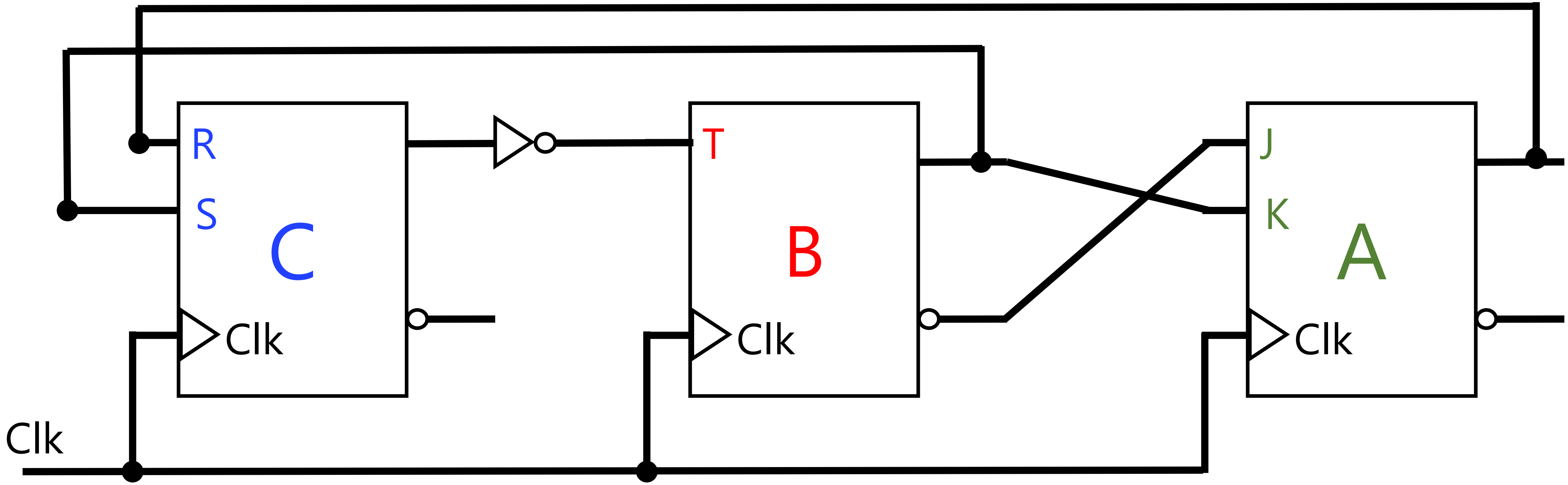
Otherwise → No


R
S
Clk
T
Clk
J
K
Clk
Clk

Analysis

1) What are the FFs?

1.1. We pick a name for each FF
1.2. We note the type of FF

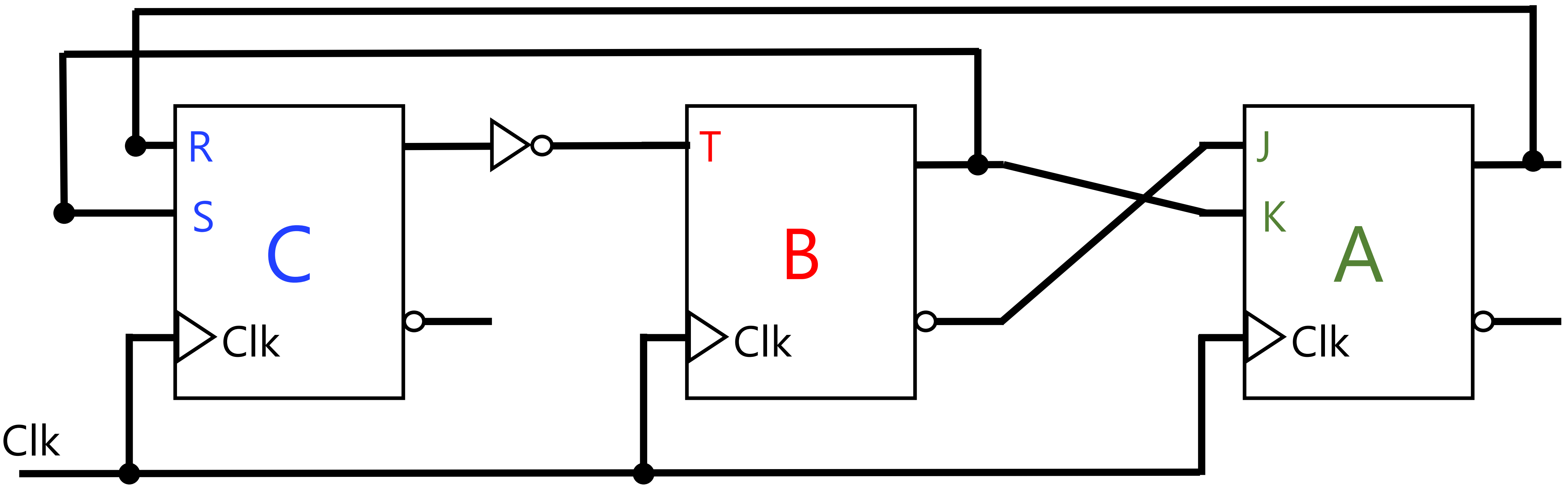

R
S
C
Clk
T
B
Clk
J
K
A
Clk
Clk

Analysis

2) What are the state combinations (possibilities)?

Each FF can have {0,1} states

In total, $2^{\#FFs}$

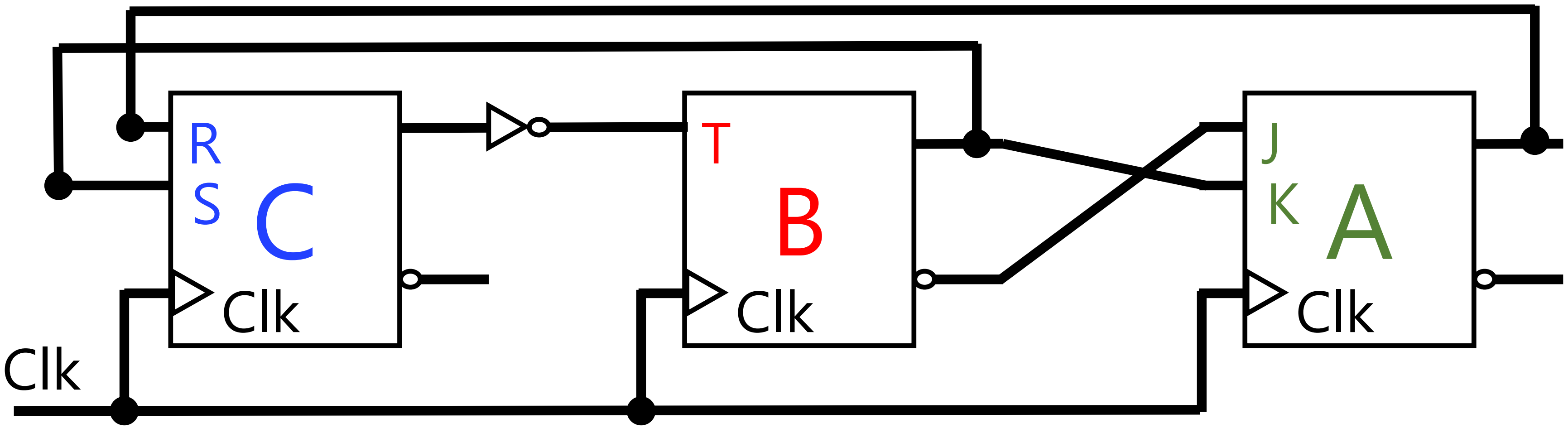


#FFs = 3 → 2^3 = 8 combinations

Analysis

3) Form a ‘State’ Table

3.1. For each FF, one column for current state
3.2. For each FF, one column for next state
3.3. For each combination of current state one row

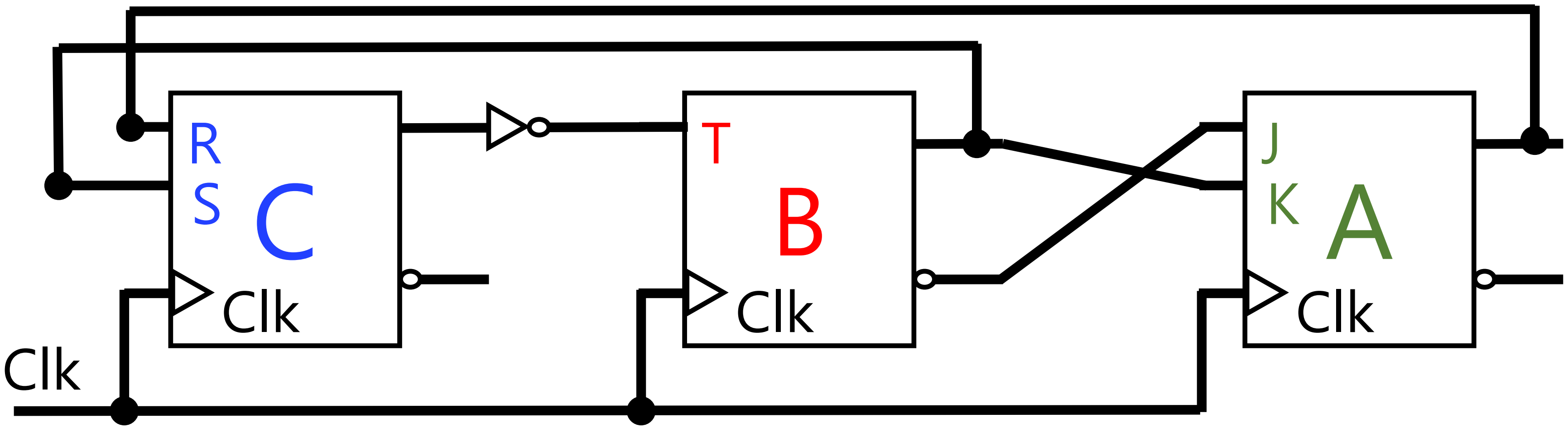


Q(T)			Q(T+1)		
C	B	A	C	B	A

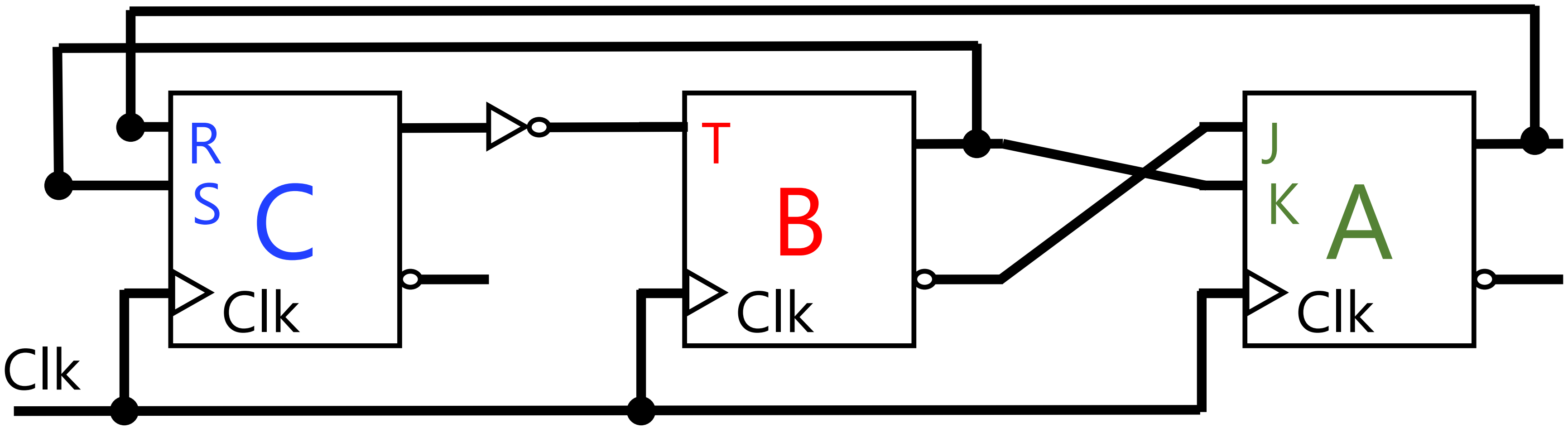
Analysis
4) Fill the ‘State’ table

For each FF, we determine the next state based on

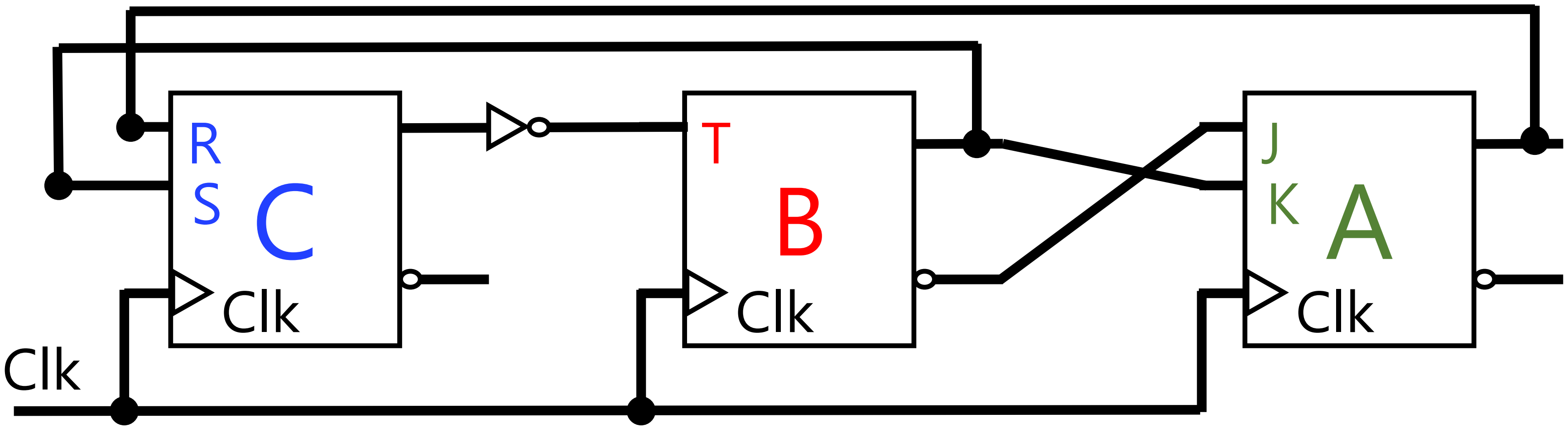
I) current state
II) the current value of inputs to the FF



Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0			$Q_A(T)=0$ $J_A=Q'_B(T)=1$ $K_A=Q_B(T)=0$ ----------------- Set Action: 1
0	0	1			
0	1	0			
0	1	1			
1	0	0			



Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0		$Q_B(T)=0$ $T_B=Q'_C(T)=1$ -------------- Comp. $(Q_B(T)) = 1$	1
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			

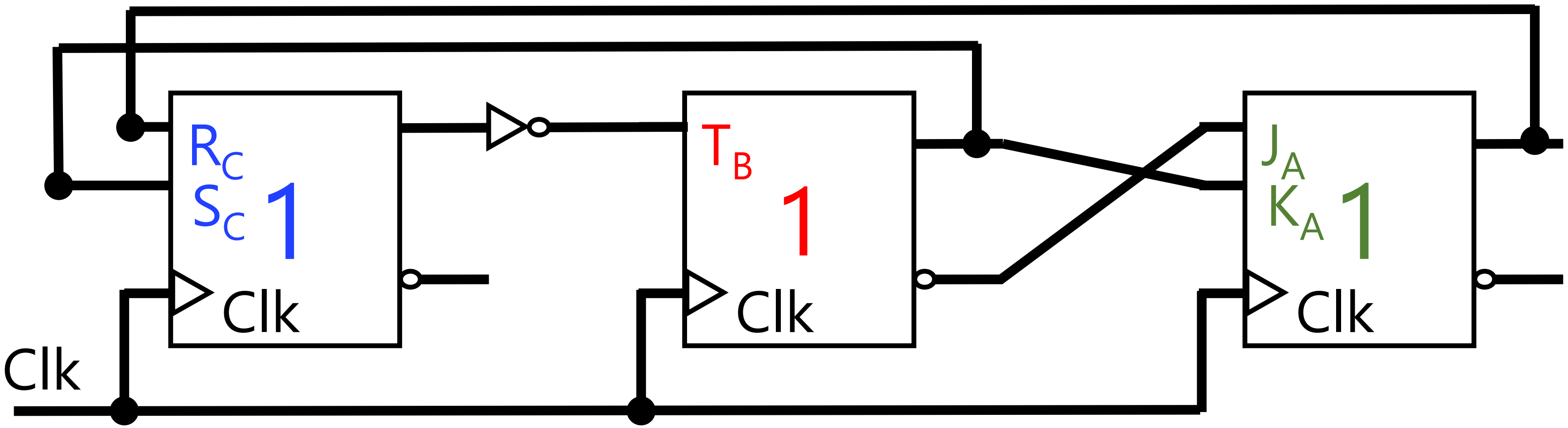


Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	$Q_C(T)=0$ $R_C=Q_A(T)=0$ $S_C=Q_B(T)=0$ --------------- Store $Q_C(T)$ = 0	1	1
0	0	1			
0	1	0			
0	1	1			
1	0	0			

Analysis

$$Q_A(T) = A,\ Q'_A(T) = A'$$

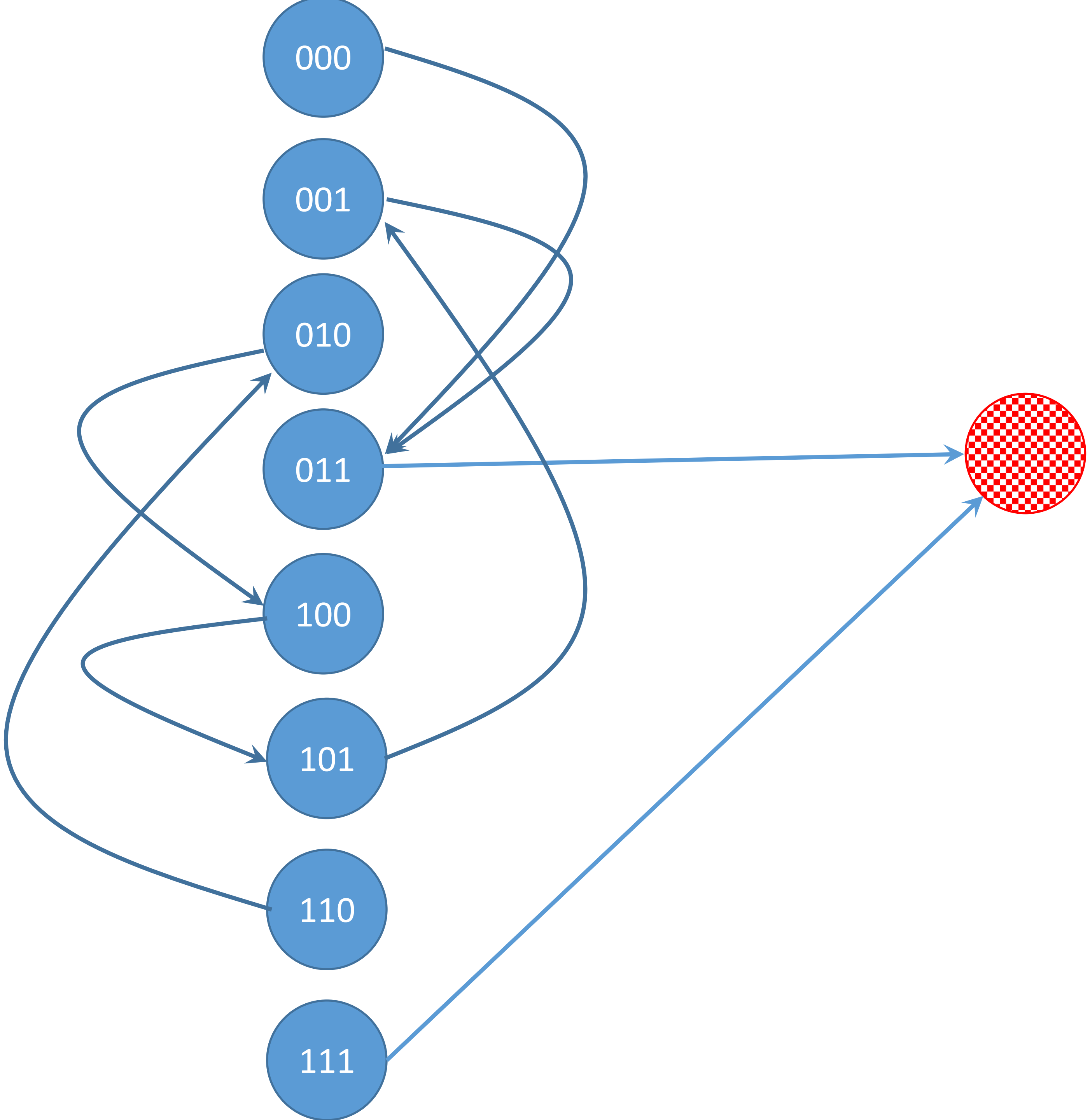
For simplicity, the current status of a FF can be assume to be as a binary variable

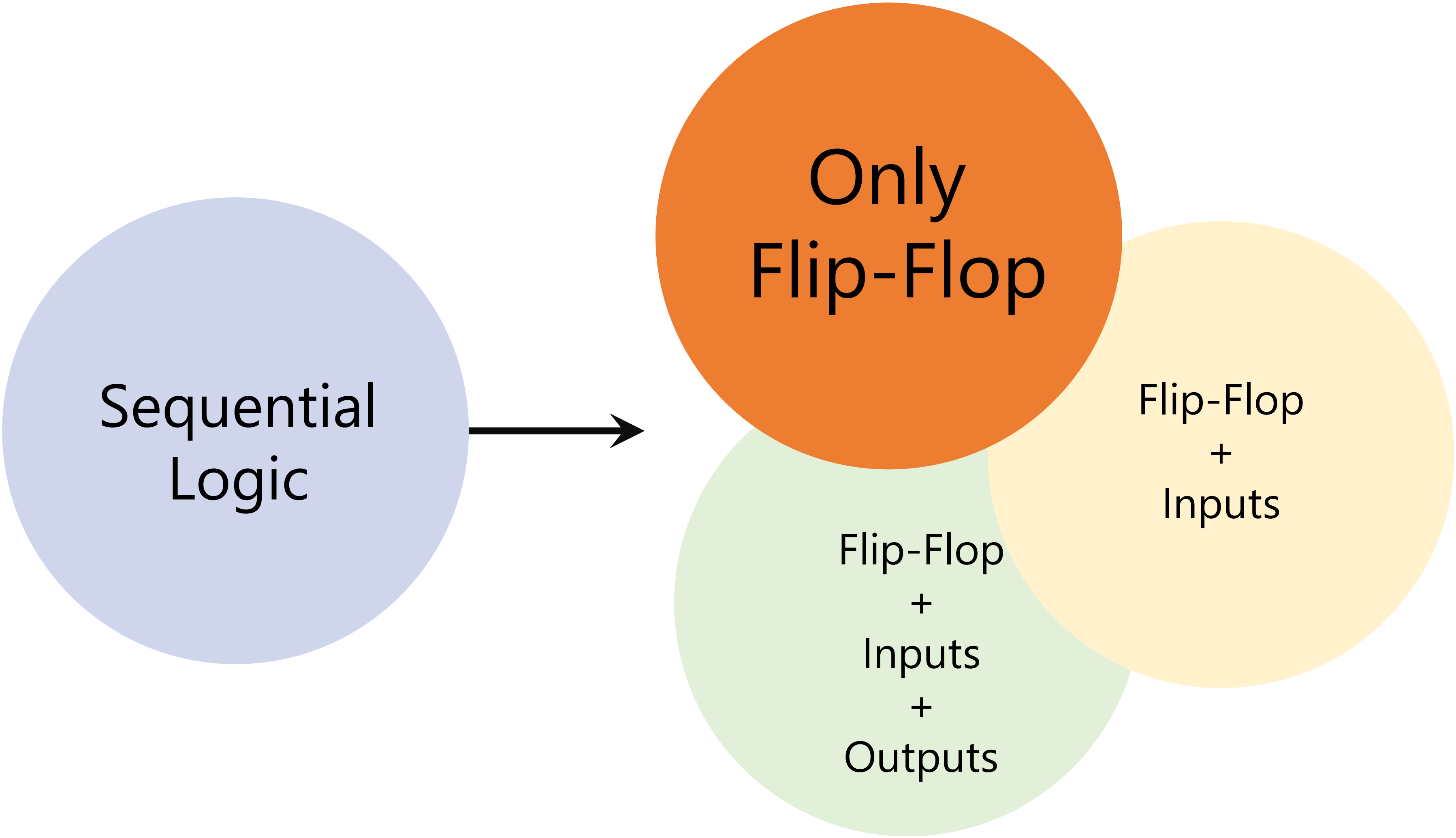


Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	x	0	0
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	0	1	0
1	1	1	x	1	0

Analysis
5) State Transition Diagram

5.1. for each state combination (each row), a node
5.2. from one state (node) to another state, a directed edge


000
001
010
011
100
101
110
111


Sequential
Logic
Only
Flip-Flop
Flip-Flop
+
Inputs
Flip-Flop
+
Inputs
+
Outputs

Design
by an example

Counter

Count from 0 to N

Design

0. Do we need combinational logic or sequential logic?

Do we need memory?

Counter

Count from 0 to N

0 → 1 → 2 → 3 → … → N-1 → N

Counter

Count from 0 to N

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow \ldots \rightarrow N\text{-}1 \rightarrow N$

At each step, we have to see at number we are and then move to next number: $i \rightarrow i+1$

Counter

Count from 0 to N

We need a storage to store current number.

We need a sequential circuit!

Design

1. How many storage (flip-flops)?

Depends on the storage you need to store the current state in binary system!

Counter

Count from 0 to N

N = 7

1 → 2 → 3 → 4 → 5 → 6 → 7

000 → 001 → 010 → 011 → 100 → 101 → 110 → 111

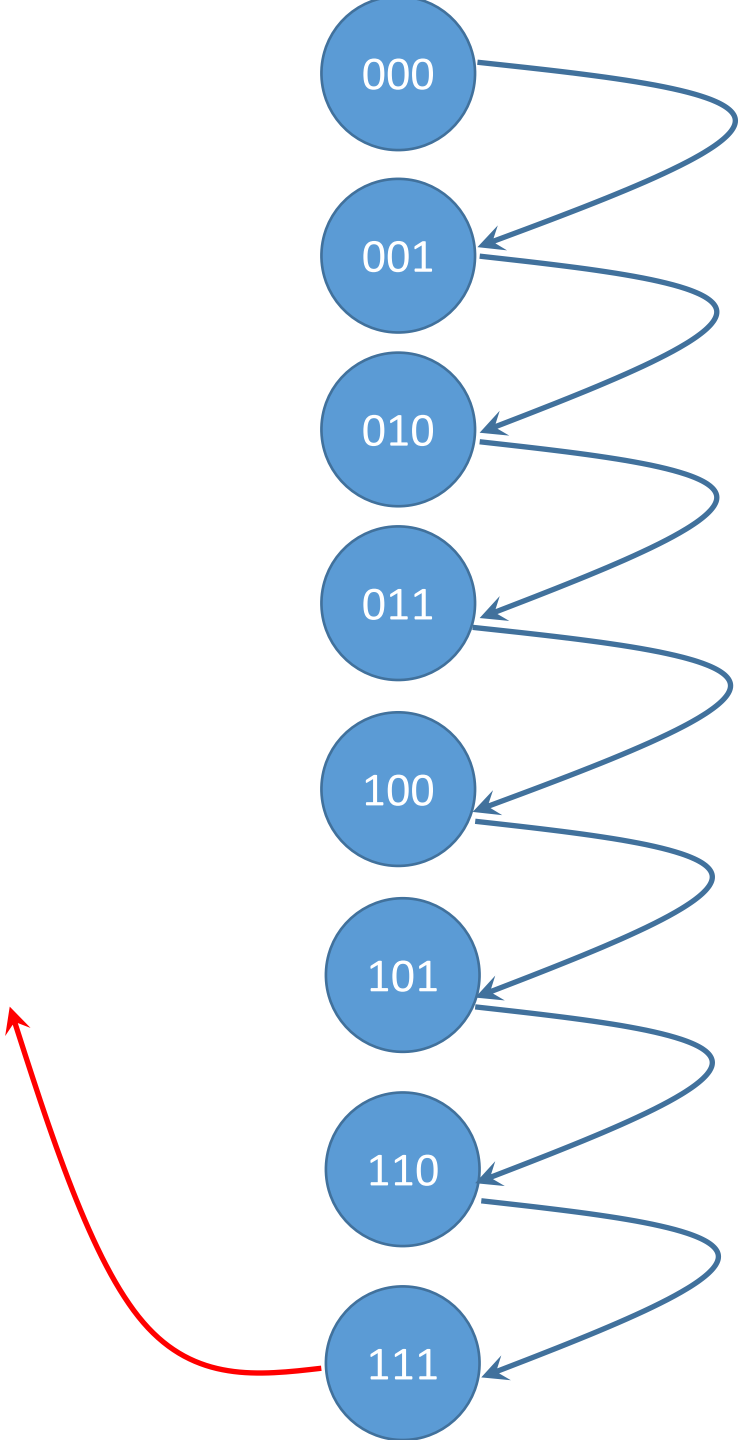
For each intermediate state, we need 3 bits → 3 flip-flops

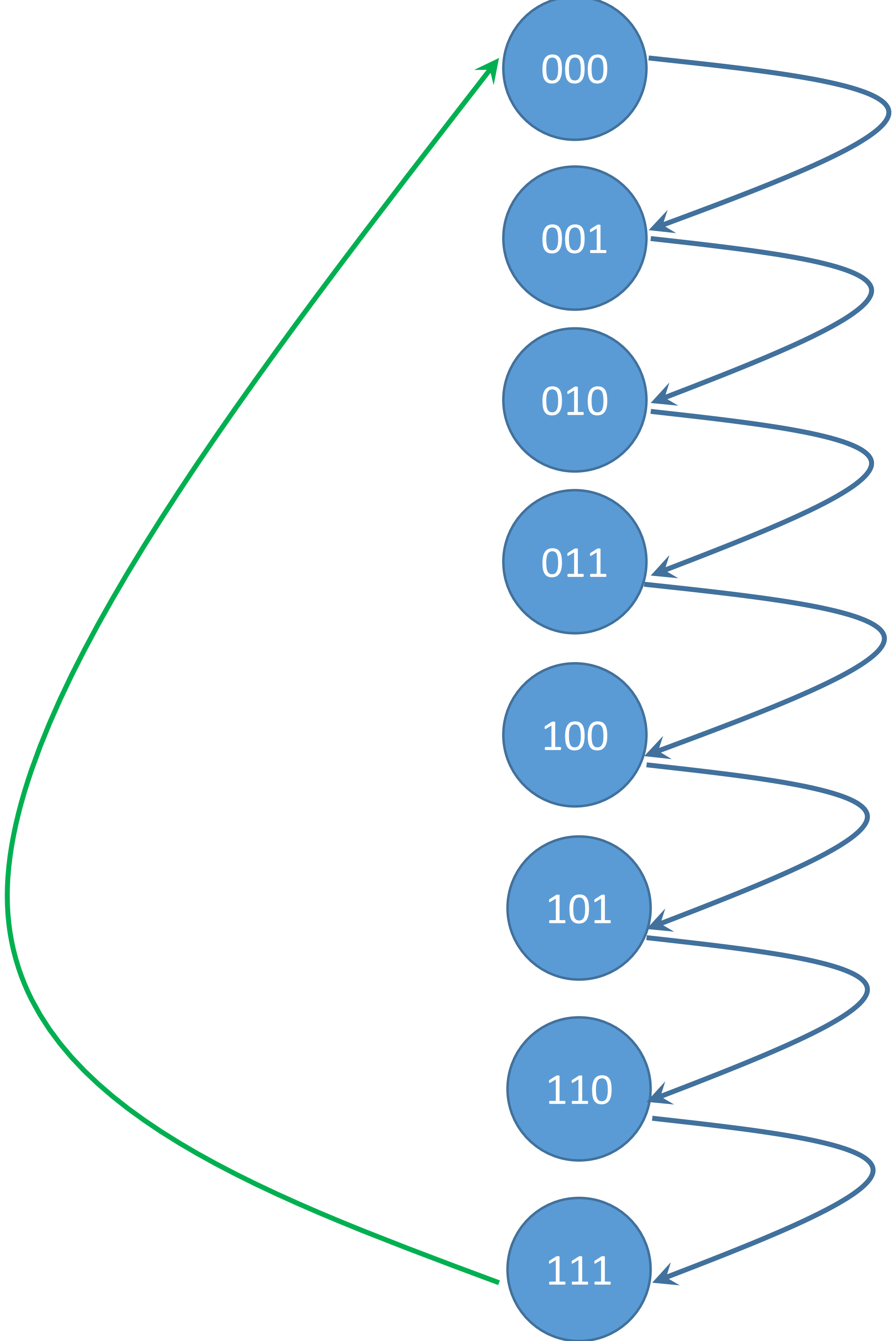
Design

2. Form the state (transition) diagram

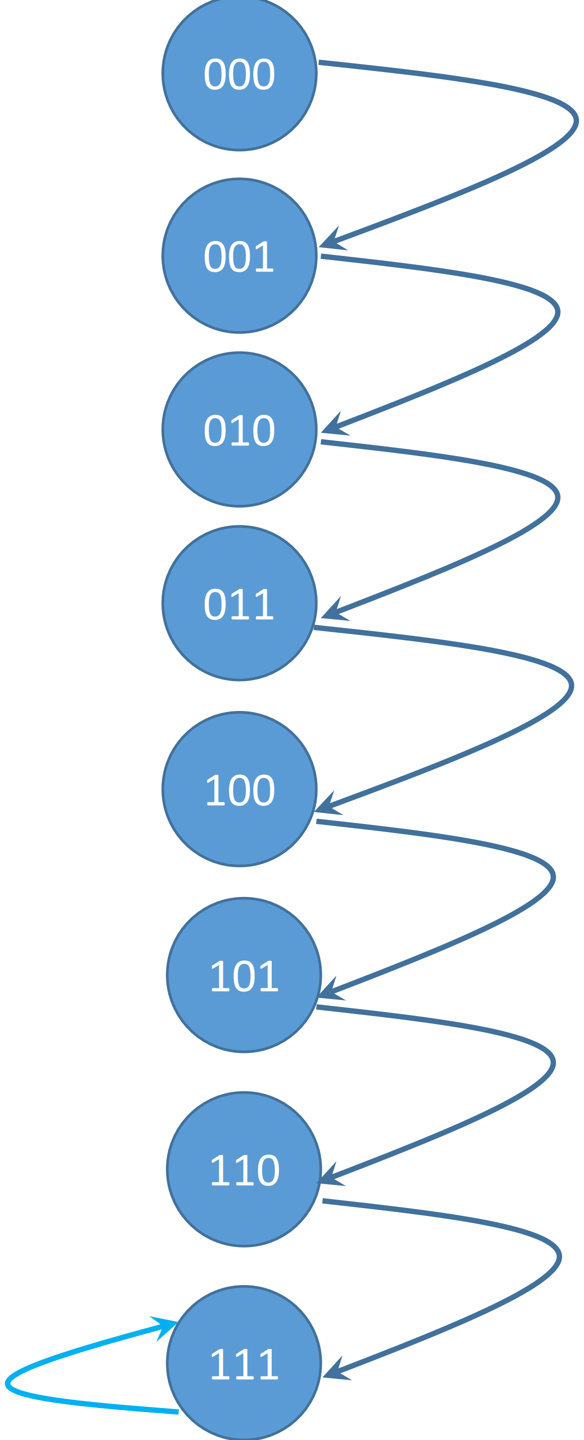
Same as analysis,

- For each state → one node
- For each state transition to next state → a directed edge


000
001
010
011
100
101
110
111



Loop to the beginning!

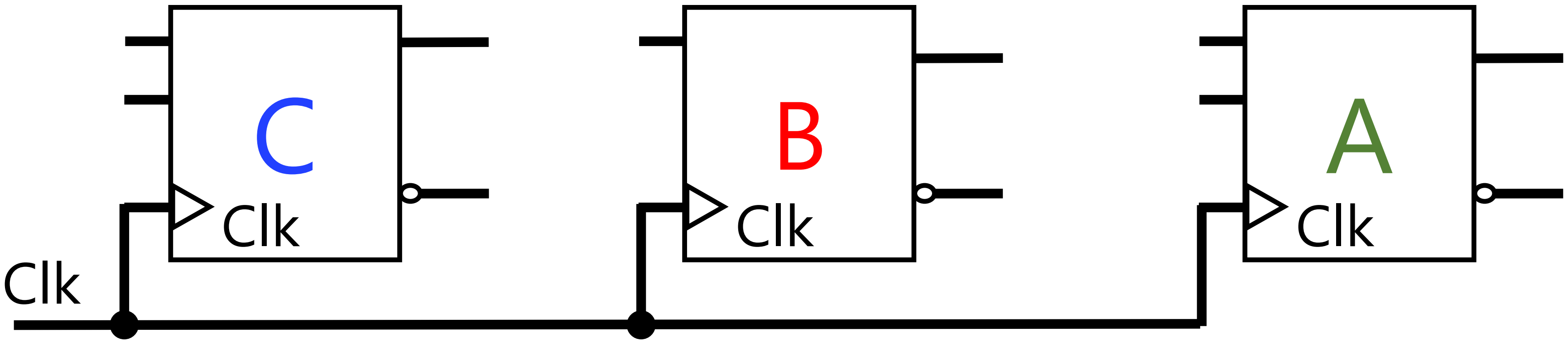


Stuck in 7
Just one time counter!

Design
3. Form the state table

Same as analysis, two columns for each flip-flop (storage unit)

a) One for current state Q(T)
b) One for next state Q(T+1)

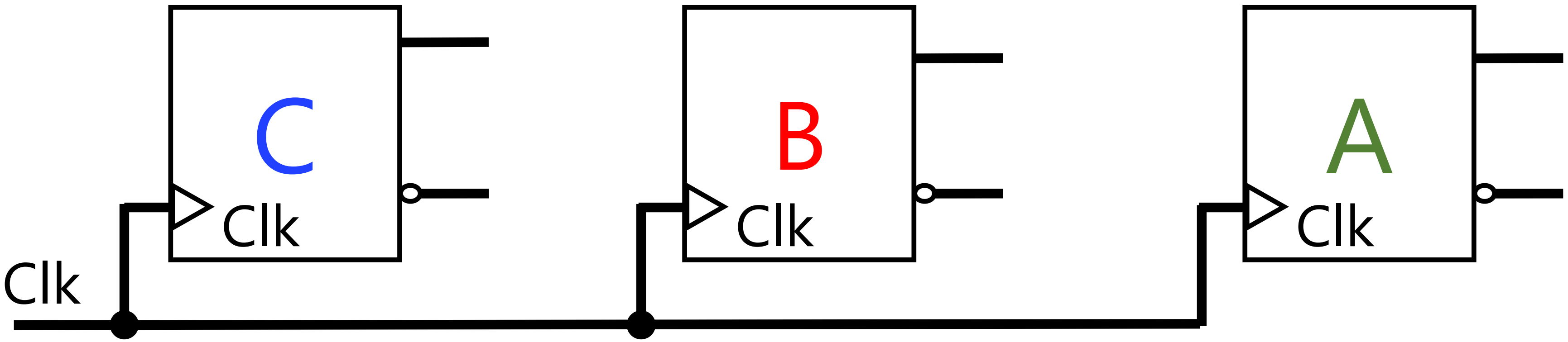


Q(T)			Q(T+1)		
C	B	A	C	B	A

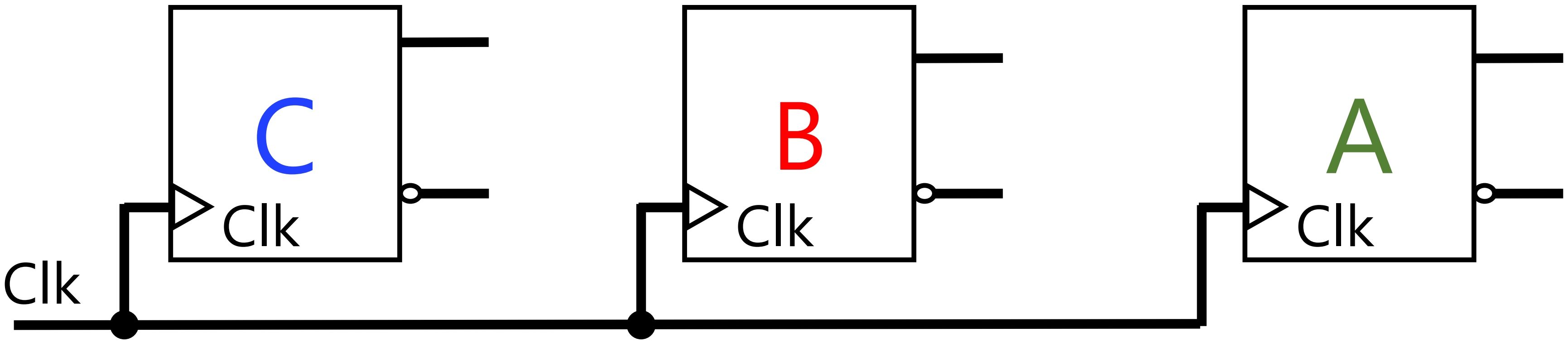
Design

4. Fill the state table

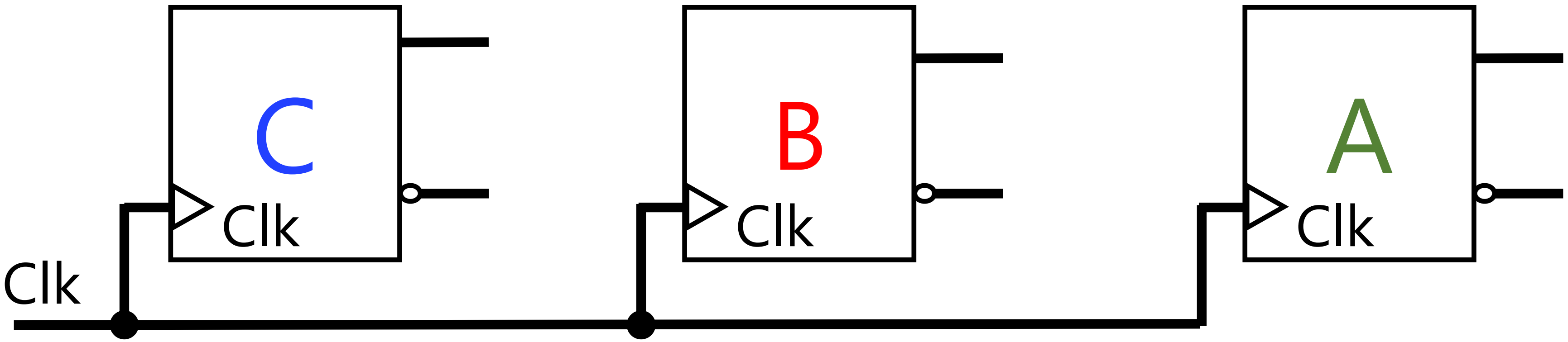
Unlike analysis, here we already know what is going to be the next state Q(T+1) based on current state Q(T)



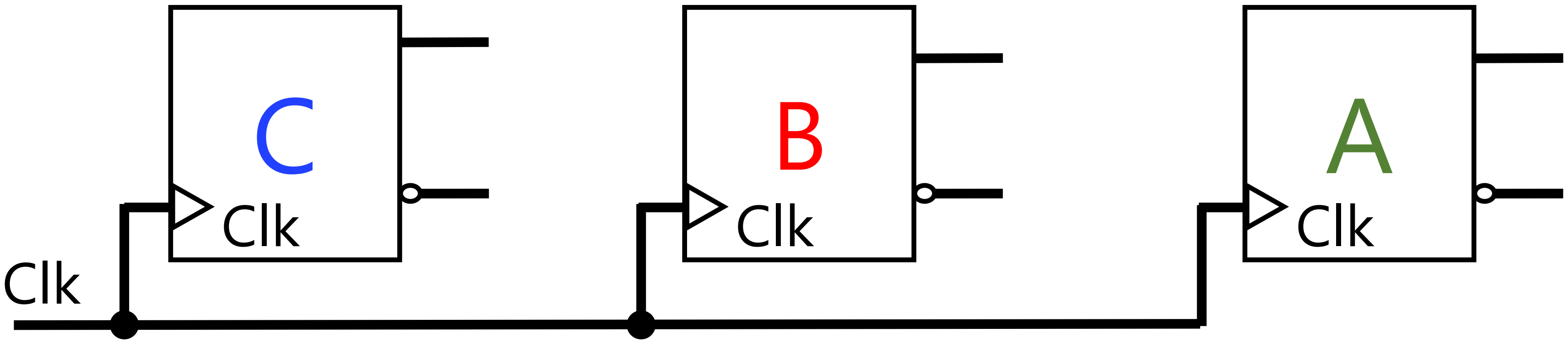
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	0	1
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			



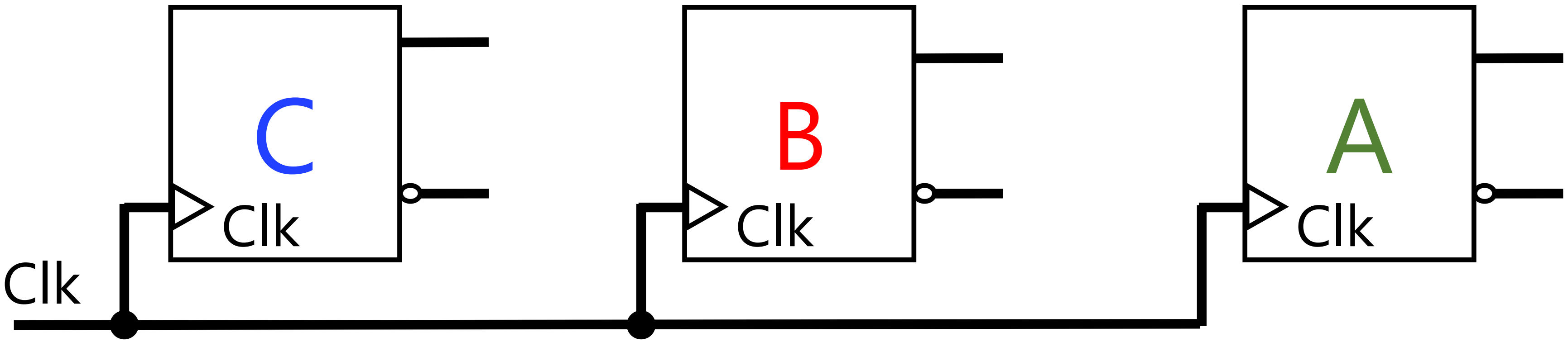
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			



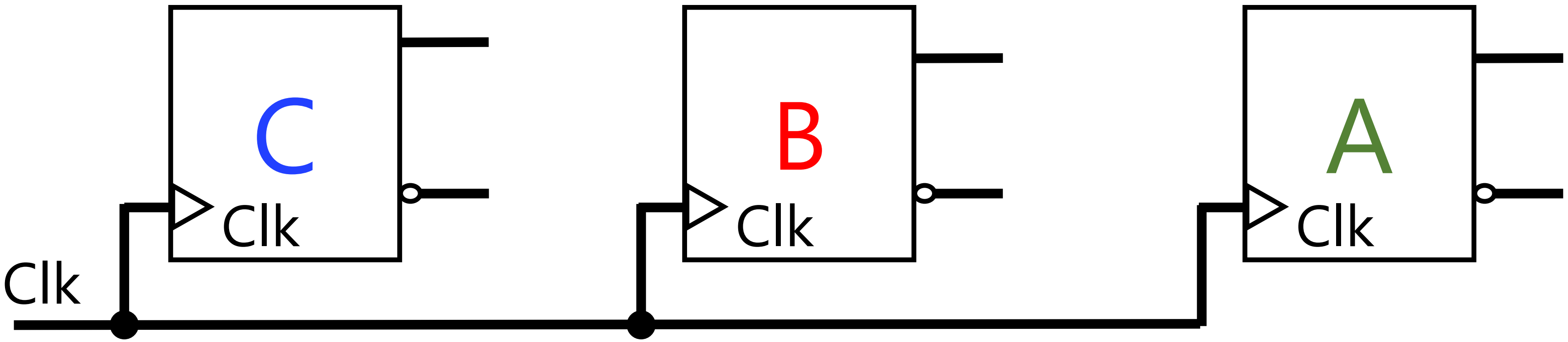
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			



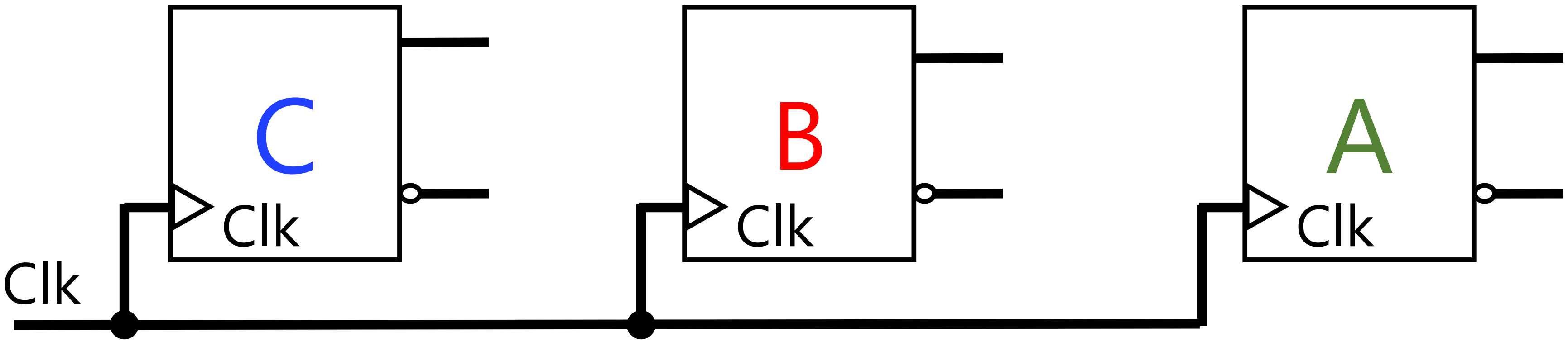
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0			
1	0	1			
1	1	0			
1	1	1			



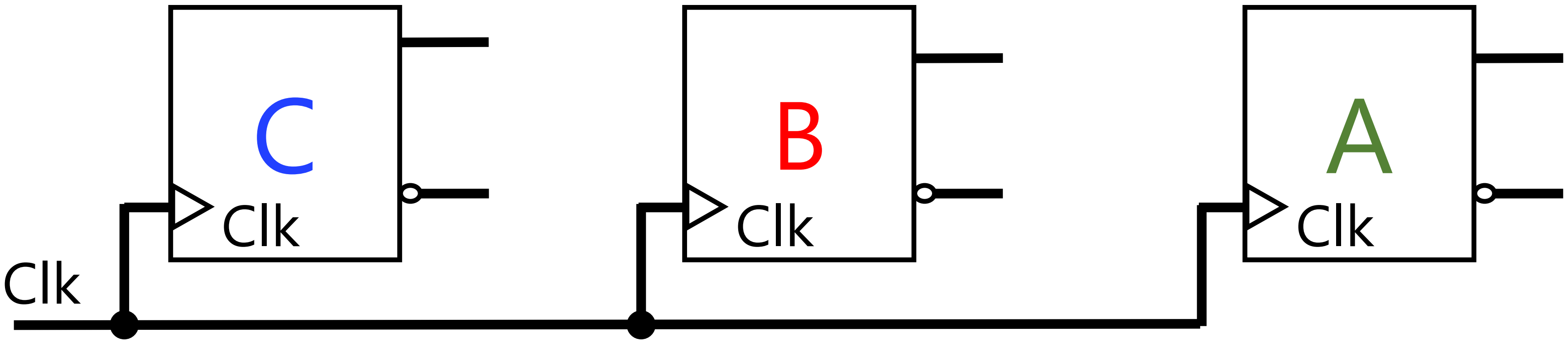
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1			
1	1	0			
1	1	1			



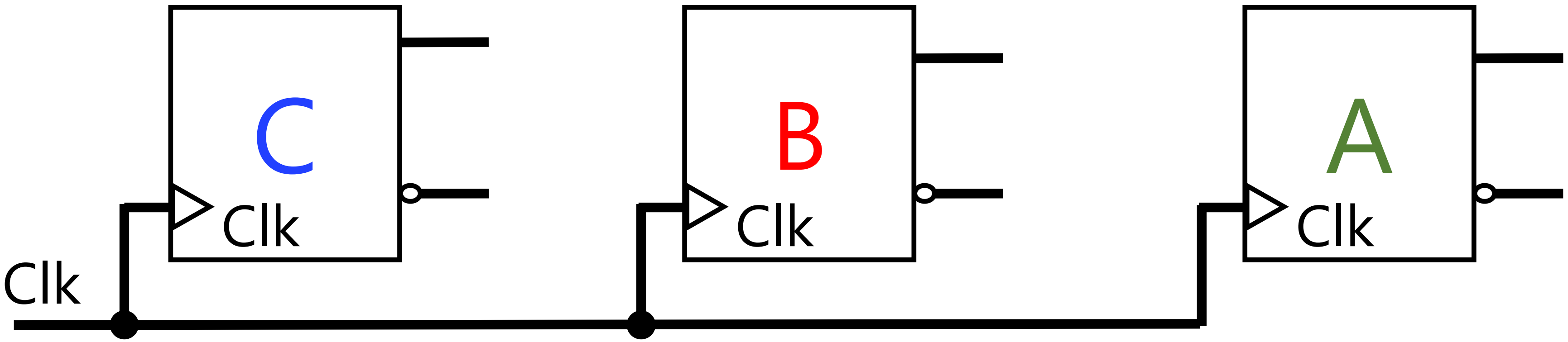
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0			
1	1	1			



Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1			

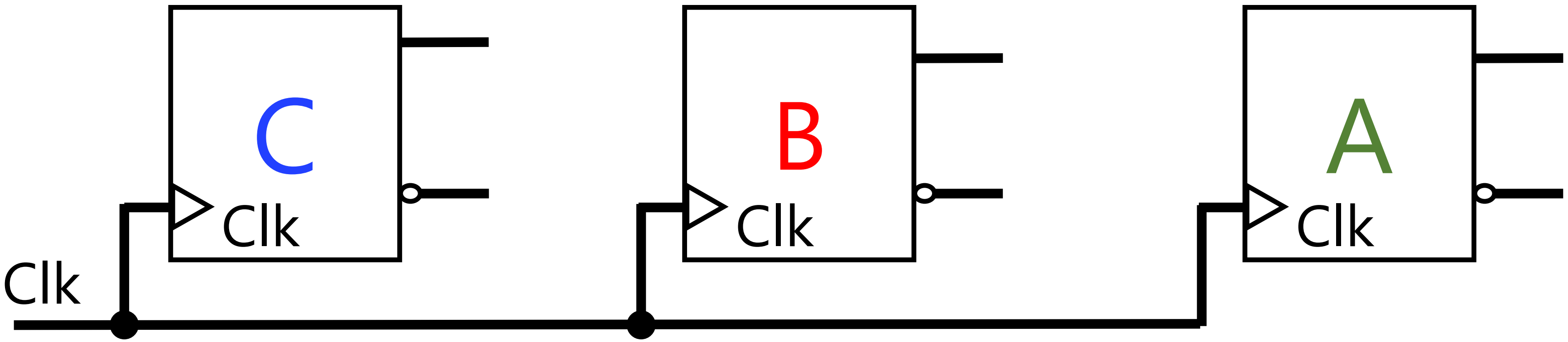


Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	?	?	?



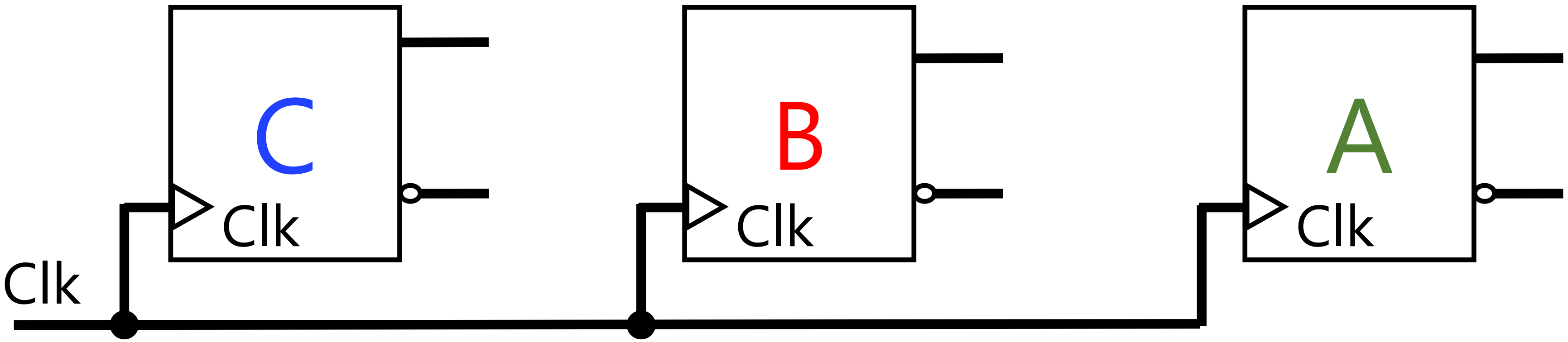
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	[illegible]	1
1	0	1	1	[illegible]	0
1	1	0	1	[illegible]	1
1	1	1	0	0	0

Loop to the beginning!



Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	[illegible]	1
1	0	1	1	[illegible]	0
1	1	0	1	[illegible]	1
1	1	1	1	1	1

Stuck in 7
Just one time counter!



Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	[illegible]	1
1	0	1	1	[illegible]	0
1	1	0	1	1	1
1	1	1	0	0	0

Our Design Choice!

Design

5. What type of storage (flip-flop)?

RS, D, T, JK, or Mixed

Design

5. What type of storage (flip-flop)?

RS, D, T, JK, or Mixed

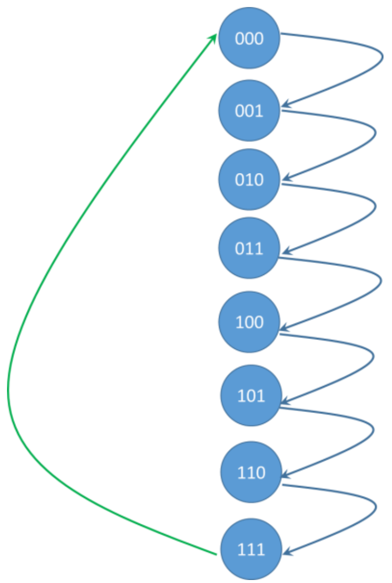
In terms of design, does <u>not</u> matter.

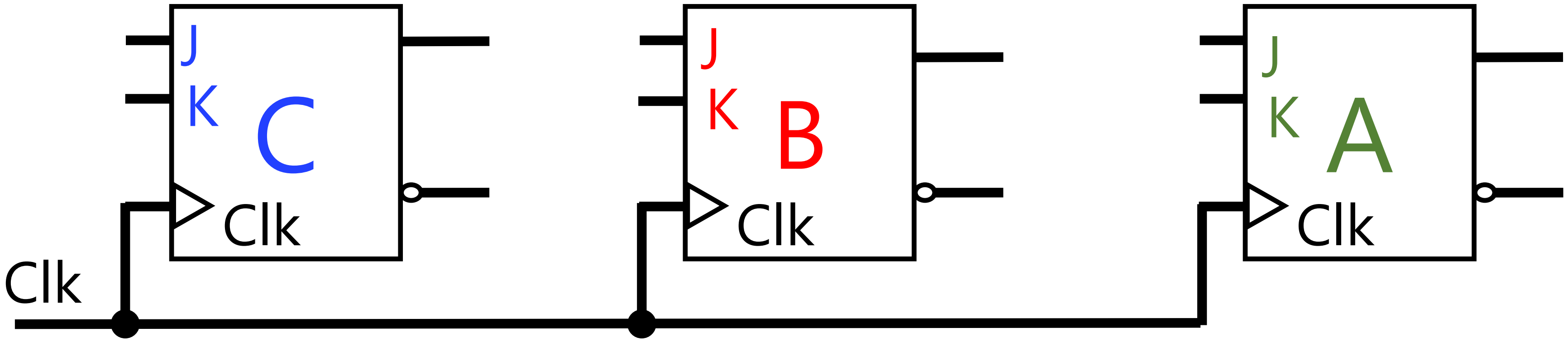
In terms of **efficiency**, matters!

Counter

Count from 0 to N=7

Let's select JK, the complete FF.





Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Design

6. Boolean expression for the flip-flops' input?

input equations, aka, *excitation* equations

Counter

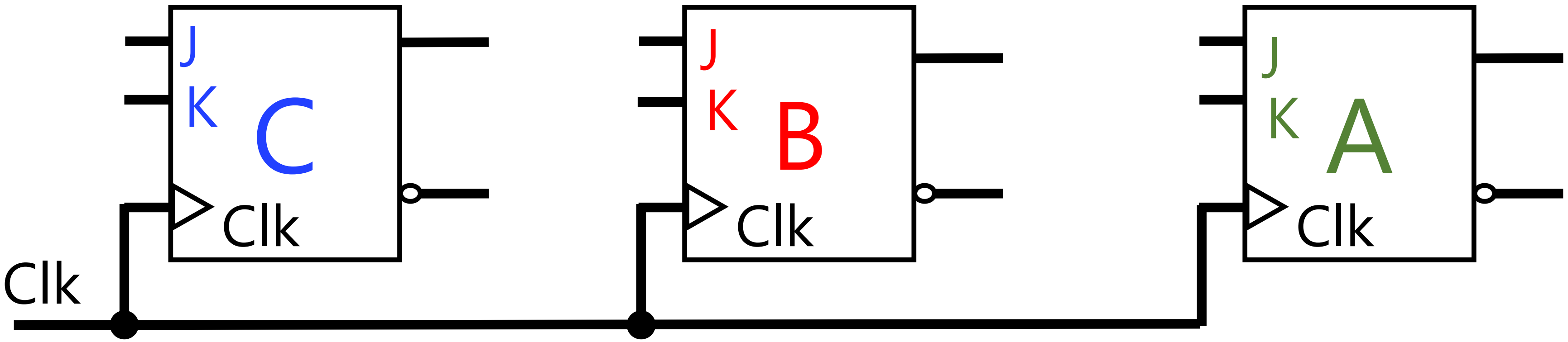
Count from 0 to N=7

A	$J_A=$	$K_A=$
B	$J_B=$	$K_B=$
C	$J_C=$	$K_C=$

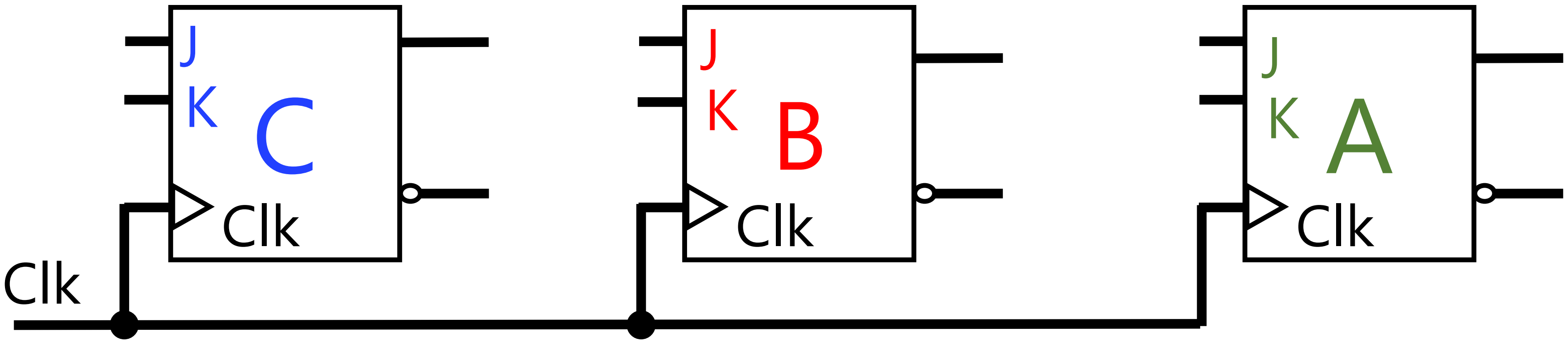
Counter

Count from 0 to N=7

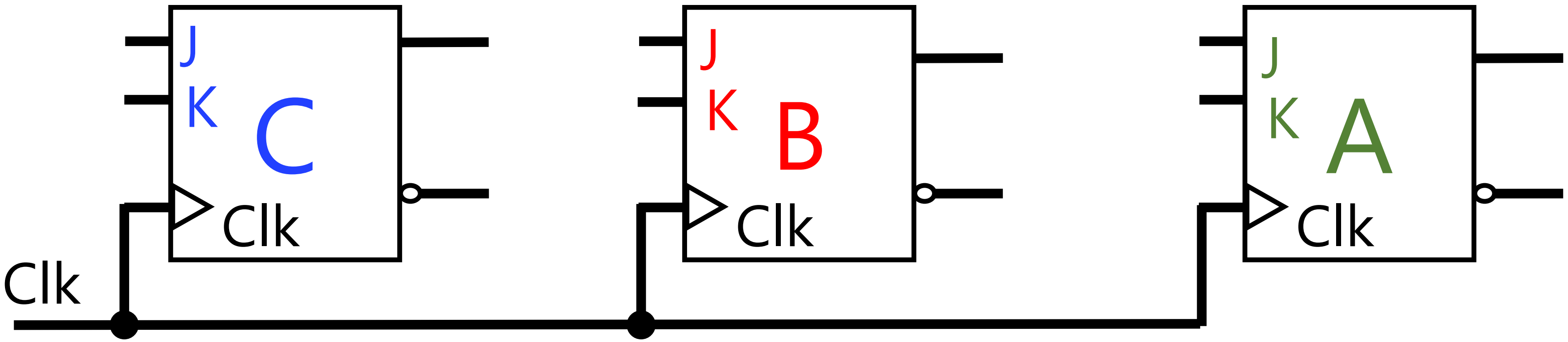
A	$J_A=$	$K_A=$
B	$J_B=$	$K_B=$
C	$J_C=$	$K_C=$



Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

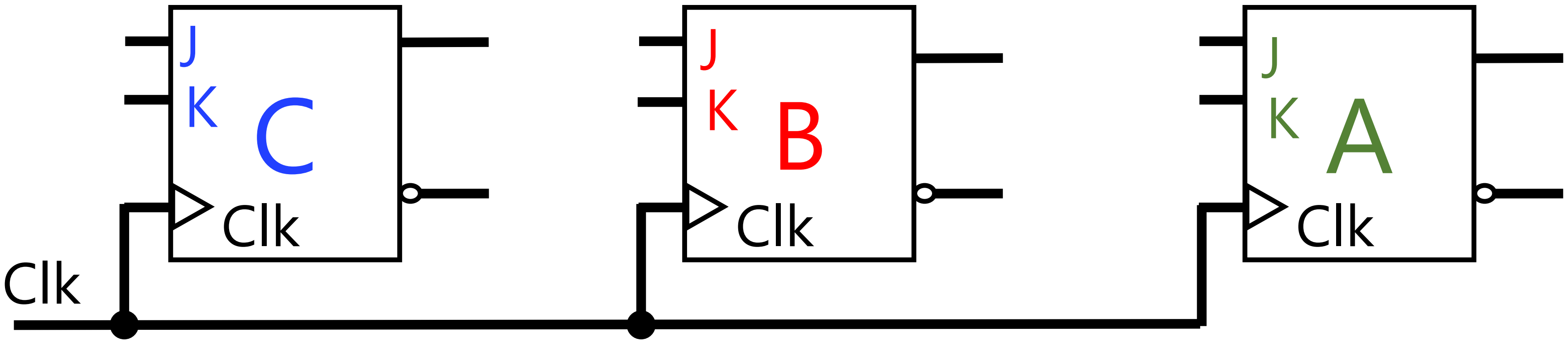


Q(T)			Q(T+1)			Not part of state table!		
C	B	A	C	B	A	Action	J_A	K_A
0	0	0	0	0	1			
0	0	1	0	1	0			
0	1	0	0	1	1			
0	1	1	1	0	0			
1	0	0	1	0	1			
1	0	1	1	1	0			
1	1	0	1	1	1			
1	1	1	0	0	0			

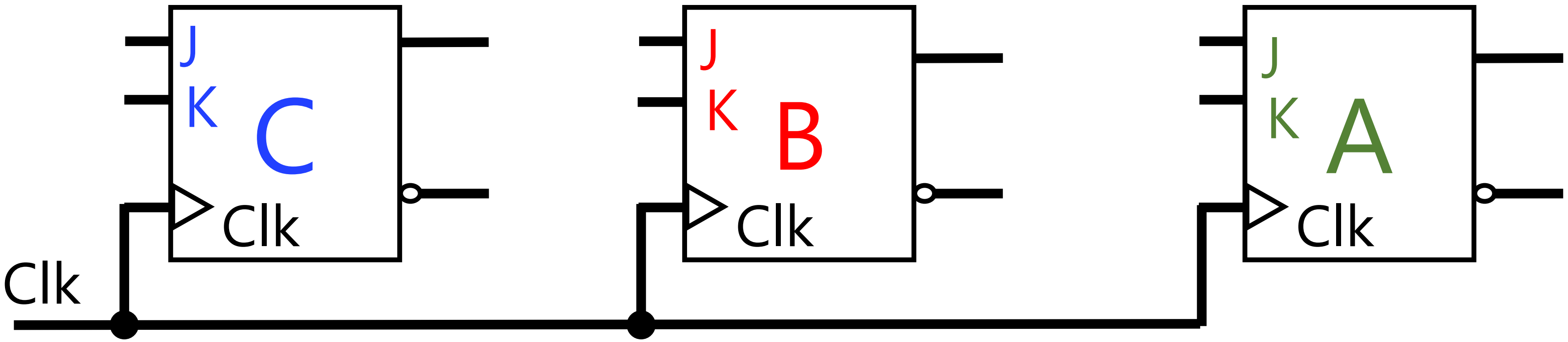


Q(T)			Q(T+1)			Not part of state table!		
C	B	A	C	B	A	Action	J_A	K_A
0	0	0	0	0	1	Set	1	0
0	0	1	0	1	0			
0	1	0	0	1	1			
0	1	1	1	0	0			
1	0	0	1	0	1			
1	0	1	1	1	0			
1	1	0	1	1	1			
1	1	1	0	0	0			

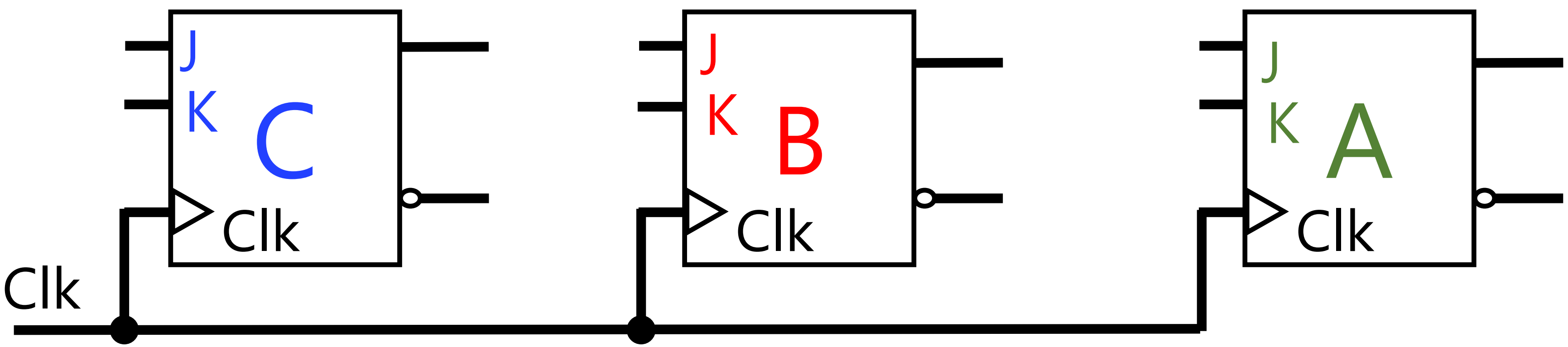
OR



Q(T)			Q(T+1)			Not part of state table!		
C	B	A	C	B	A	Action	J_A	K_A
0	0	0	0	0	1	Comp	1	1
0	0	1	0	1	0			
0	1	0	0	1	1			
0	1	1	1	0	0			
1	0	0	1	0	1			
1	0	1	1	1	0			
1	1	0	1	1	1			
1	1	1	0	0	0			

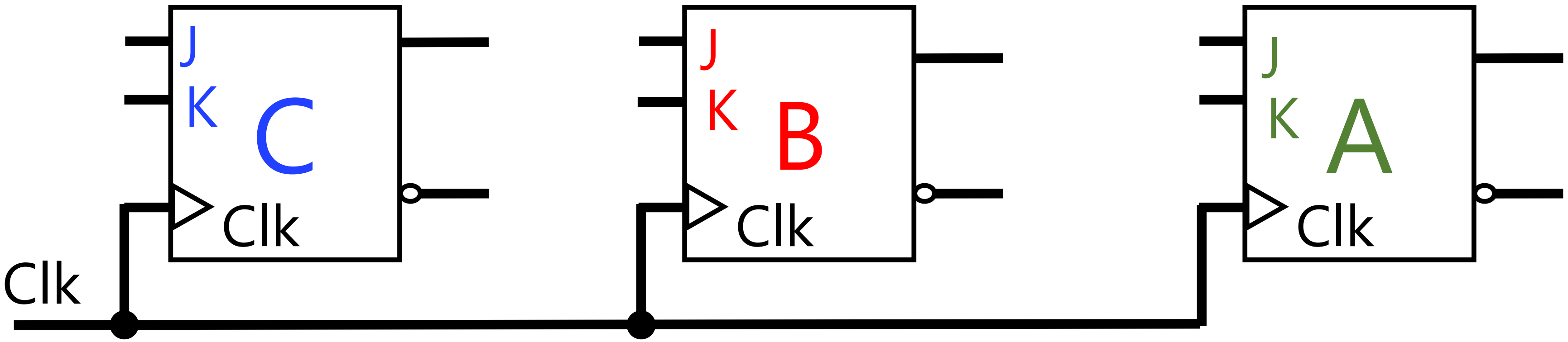


Q(T)			Q(T+1)			Not part of state table!		
C	B	A	C	B	A	Action	J_A	K_A
0	0	0	0	0	1	Set/Comp	1	0/1 → x
0	0	1	0	1	0			
0	1	0	0	1	1			
0	1	1	1	0	0			
1	0	0	1	0	1			
1	0	1	1	1	0			
1	1	0	1	1	1			
1	1	1	0	0	0			

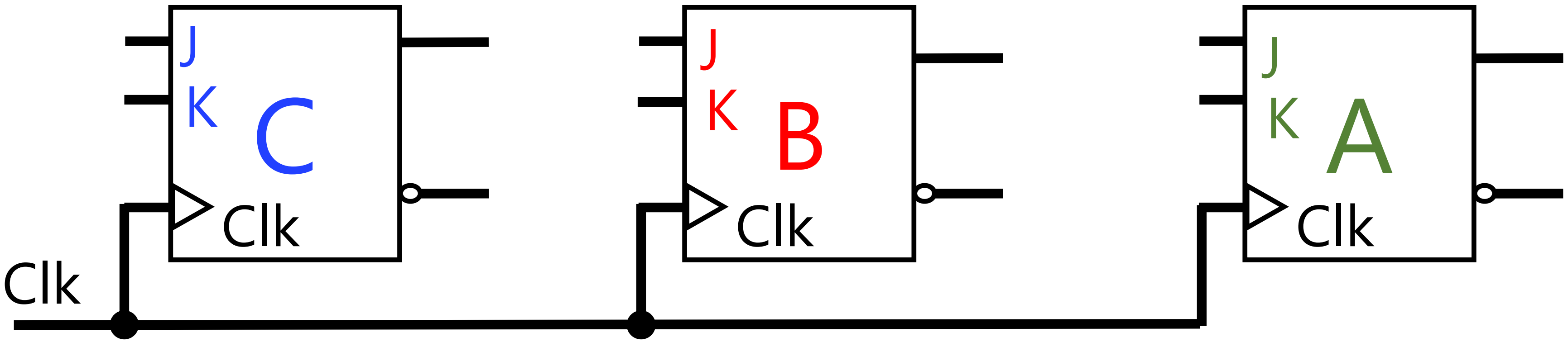


Q(T)			Q(T+1)			Not part of state table!		
C	B	A	C	B	A	Action	J_A	K_A
0	0	0	0	0	1	Set/Comp	1	x
0	0	1	0	1	0	Reset	0	1
0	1	0	0	1	1			
0	1	1	1	0	0			
1	0	0	1	0	1			
1	0	1	1	1	0			
1	1	0	1	1	1			
1	1	1	0	0	0			

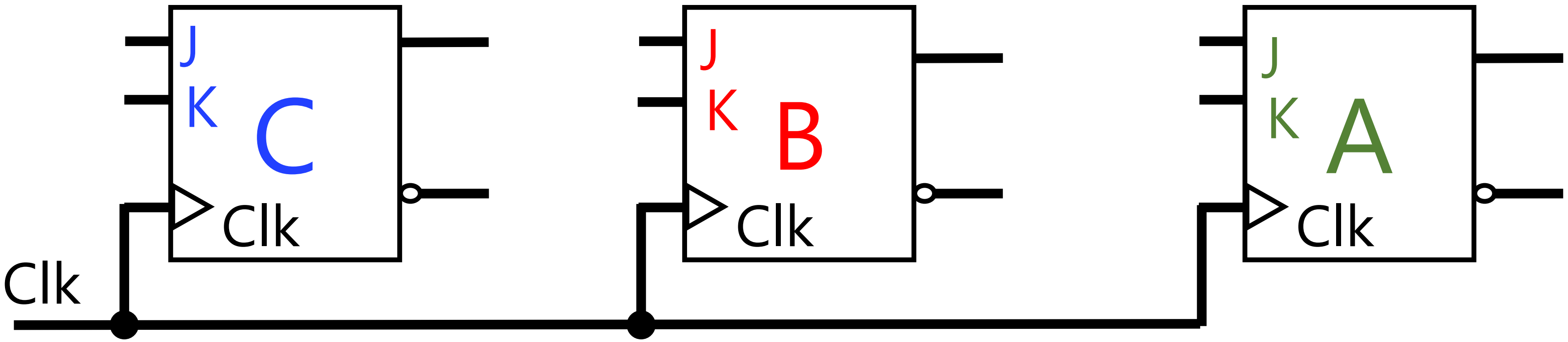
OR



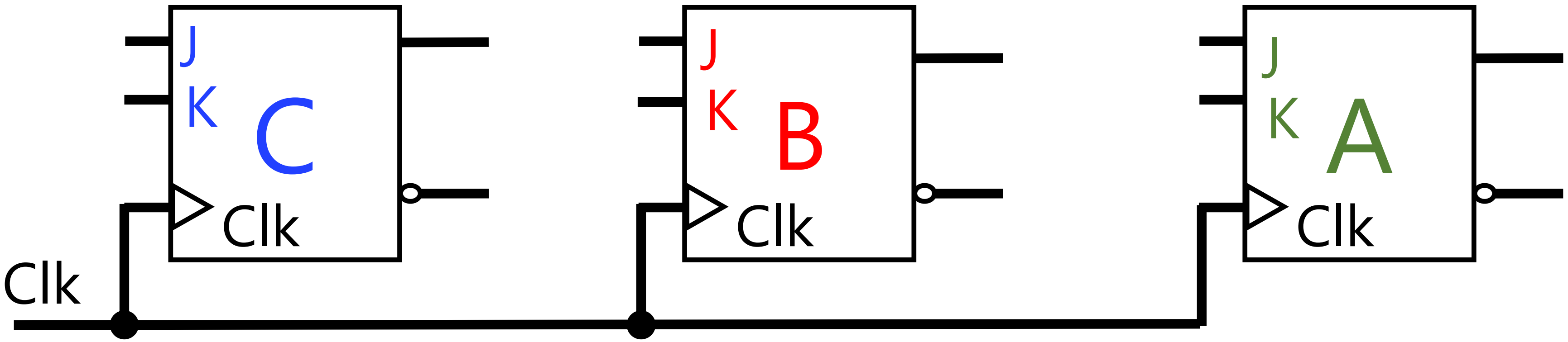
Q(T)			Q(T+1)			Not part of state table!		
C	B	A	C	B	A	Action	J_A	K_A
0	0	0	0	0	1	Set/Comp	1	x
0	0	1	0	1	0	Comp.	1	1
0	1	0	0	1	1			
0	1	1	1	0	0			
1	0	0	1	0	1			
1	0	1	1	1	0			
1	1	0	1	1	1			
1	1	1	0	0	0			



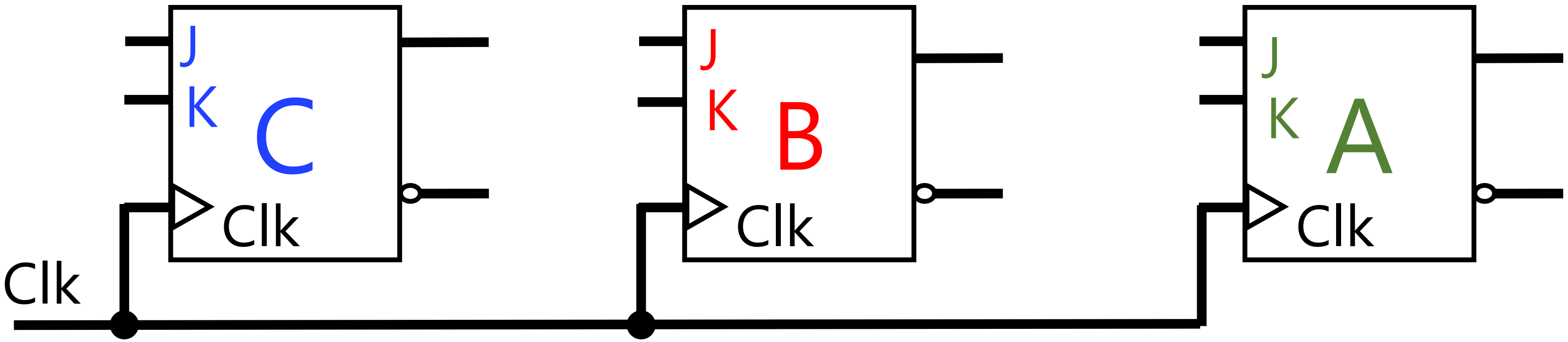
Q(T)			Q(T+1)			Not part of state table!		
C	B	A	C	B	A	Action	J_A	K_A
0	0	0	0	0	1	Set/Comp	1	x
0	0	1	0	1	0	Reset/Comp	x	1
0	1	0	0	1	1			
0	1	1	1	0	0			
1	0	0	1	0	1			
1	0	1	1	1	0			
1	1	0	1	1	1			
1	1	1	0	0	0			



Q(T)			Q(T+1)			Not part of state table!		
C	B	A	C	B	A	Action	J_A	K_A
0	0	0	0	0	1	Set/Comp	1	x
0	0	1	0	1	0	Reset/Comp	x	1
0	1	0	0	1	1	Set/Comp	1	x
0	1	1	1	0	0			
1	0	0	1	0	1			
1	0	1	1	1	0			
1	1	0	1	1	1			
1	1	1	0	0	0			



Q(T)			Q(T+1)			Not part of state table!		
C	B	A	C	B	A	Action	J_A	K_A
0	0	0	0	0	1	Set/Comp	1	x
0	0	1	0	1	0	Reset/Comp	x	1
0	1	0	0	1	1	Set/Comp	1	x
0	1	1	1	0	0	Reset/Comp	x	1
1	0	0	1	0	1			
1	0	1	1	1	0			
1	1	0	1	1	1			
1	1	1	0	0	0			



Q(T)			Q(T+1)			Not part of state table!		
C	B	A	C	B	A	Action	J_A	K_A
0	0	0	0	0	1	Set/Comp	1	x
0	0	1	0	1	0	Reset/Comp	x	1
0	1	0	0	1	1	Set/Comp	1	x
0	1	1	1	0	0	Reset/Comp	x	1
1	0	0	1	0	1	Set/Comp	1	x
1	0	1	1	1	0	Reset/Comp	x	1
1	1	0	1	1	1	Set/Comp	1	x
1	1	1	0	0	0	Reset/Comp	x	1

Counter

Count from 0 to N=7

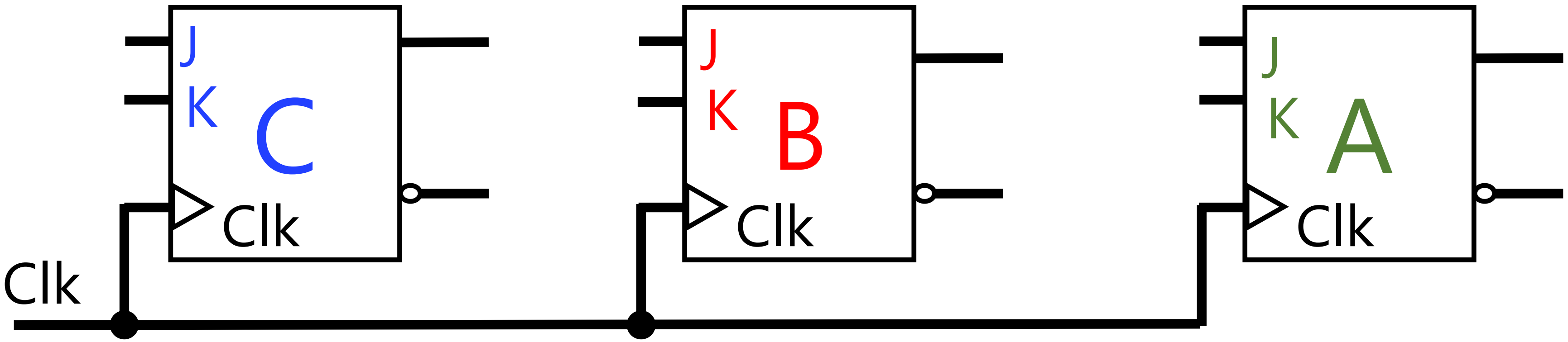
Flip-Flops

A	$J_A = F(C,B,A) = \sum(0,2,4,6) + d(1,3,5,7)$	$K_A = F(C,B,A) = \sum(1,3,5,7) + d(0,2,4,6)$
B	$J_B =$	$K_B =$
C	$J_C =$	$K_C =$

Counter

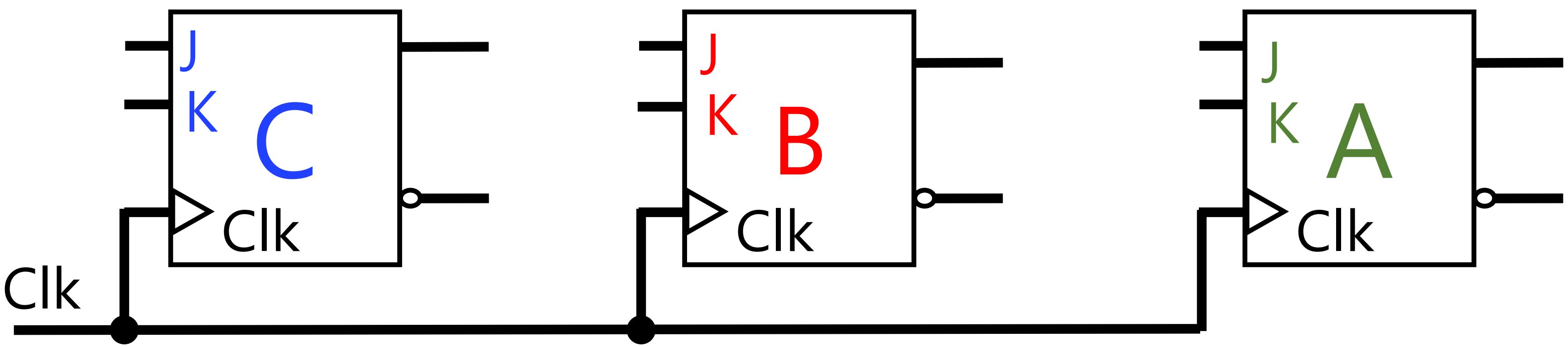
Count from 0 to N=7

A	$J_A = F(C,B,A) = \sum(0,2,4,6) + d(1,3,5,7)$	$K_A = F(C,B,A) = \sum(1,3,5,7) + d(0,2,4,6)$
B	$J_B =$	$K_B =$
C	$J_C =$	$K_C =$

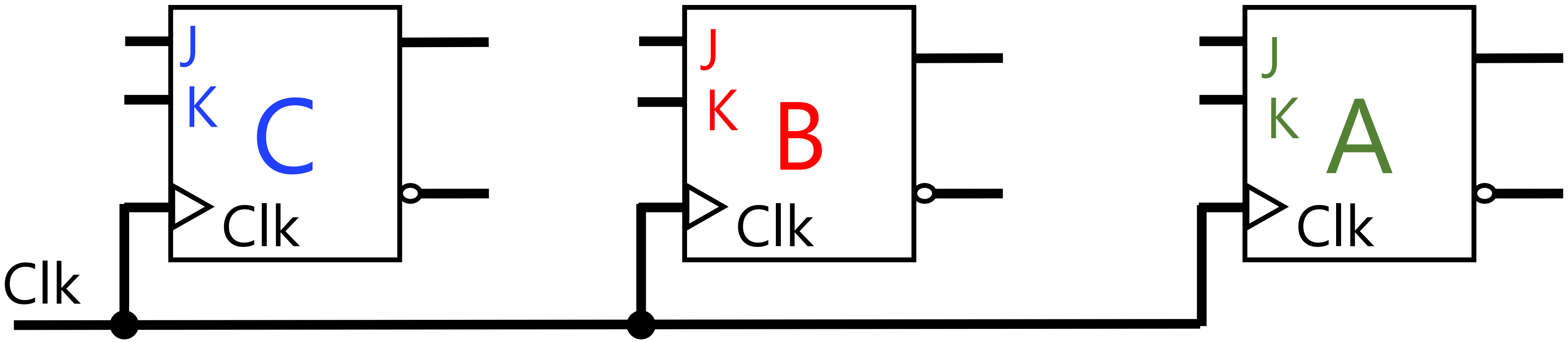


Q(T)			Q(T+1)			Not part of state table!		
C	B	A	C	B	A	Action	J_B	K_B
0	0	0	0	0	1	Store	0	0
0	0	1	0	1	0			
0	1	0	0	1	1			
0	1	1	1	0	0			
1	0	0	1	0	1			
1	0	1	1	1	0			
1	1	0	1	1	1			
1	1	1	0	0	0			

OR



Q(T)			Q(T+1)			Not part of state table!		
C	B	A	C	B	A	Action	J_B	K_B
0	0	0	0	0	1	Reset	0	1
0	0	1	0	1	0			
0	1	0	0	1	1			
0	1	1	1	0	0			
1	0	0	1	0	1			
1	0	1	1	1	0			
1	1	0	1	1	1			
1	1	1	0	0	0			



Q(T)			Q(T+1)			Not part of state table!		
C	B	A	C	B	A	Action	J_B	K_B
0	0	0	0	0	1	Store/Reset	0	x
0	0	1	0	1	0	Set/Comp	1	x
0	1	0	0	1	1	Store/Set	x	0
0	1	1	1	0	0	Reset/Comp	x	1
1	0	0	1	0	1	Store/Reset	0	x
1	0	1	1	1	0	Set/Comp	1	x
1	1	0	1	1	1	Store/Set	x	0
1	1	1	0	0	0	Reset/Comp	x	1

Counter

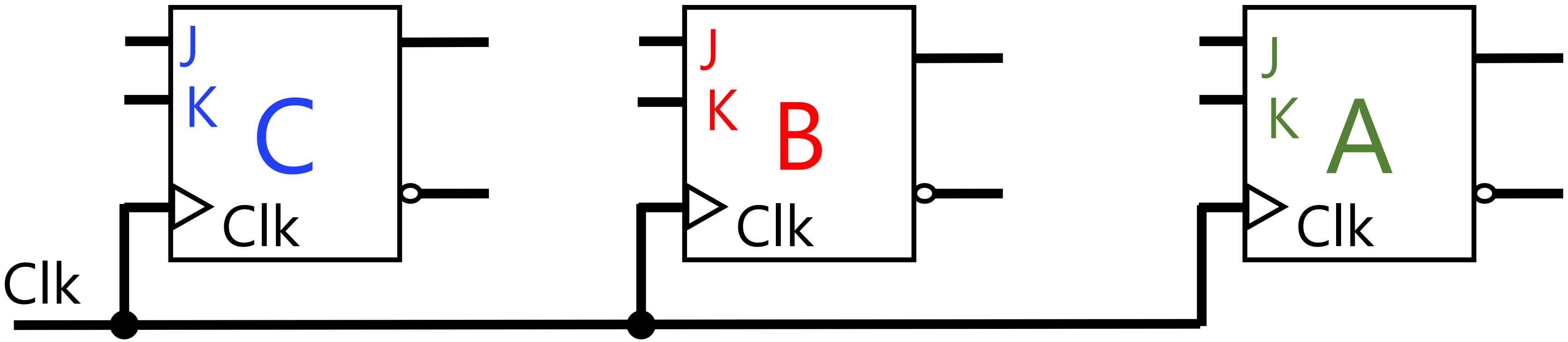
Count from 0 to N=7

A	$J_A=F(C,B,A)=\sum(0,2,4,6)+d(1,3,5,7)$	$K_A= F(C,B,A)=\sum(1,3,5,7)+d(0,2,4,6)$
B	$J_B=F(C,B,A)= \sum(1,5)+d(2,3,6,7)$	$K_B=F(C,B,A)= \sum(3,7)+d(0,1,4,5)$
C	$J_C=$	$K_C=$

Counter

Count from 0 to N=7

A	$J_A=F(C,B,A)=\sum(0,2,4,6)+d(1,3,5,7)$	$K_A= F(C,B,A)=\sum(1,3,5,7)+d(0,2,4,6)$
B	$J_B=F(C,B,A)= \sum(1,5)+d(2,3,6,7)$	$K_B=F(C,B,A)= \sum(3,7)+d(0,1,4,5)$
C	$J_C=$	$K_C=$



Q(T)			Q(T+1)			Not part of state table!		
C	B	A	C	B	A	Action	J_C	K_C
0	0	0	0	0	1	Store/Reset	0	X
0	0	1	0	1	0	Store/Reset	0	X
0	1	0	0	1	1	Store/Reset	0	X
0	1	1	1	0	0	Comp/Set	1	X
1	0	0	1	0	1	Store/Set	X	0
1	0	1	1	1	0	Store/Set	X	0
1	1	0	1	1	1	Store/Set	X	0
1	1	1	0	0	0	Comp/Reset	X	1

Counter

Count from 0 to N=7

A	$J_A=F(C,B,A)=\sum(0,2,4,6)+d(1,3,5,7)$	$K_A= F(C,B,A)=\sum(1,3,5,7)+d(0,2,4,6)$
B	$J_B=F(C,B,A)= \sum(1,5)+d(2,3,6,7)$	$K_B=F(C,B,A)= \sum(3,7)+d(0,1,4,5)$
C	$J_C=F(C,B,A)=\sum(3)+d(4,5,6,7)$	$K_C=F(C,B,A)= \sum(7)+d(0,1,2,3)$

Design

7. Minimization of input (*excitation*) equations

Counter

Count from 0 to N=7

3-Variable K-Map

A	$J_A = F(C,B,A) = \sum(0,2,4,6) + d(1,3,5,7)$	$K_A = F(C,B,A) = \sum(1,3,5,7) + d(0,2,4,6)$
B	$J_B = F(C,B,A) = \sum(1,5) + d(2,3,6,7)$	$K_B = F(C,B,A) = \sum(3,7) + d(0,1,4,5)$
C	$J_C = F(C,B,A) = \sum(3) + d(4,5,6,7)$	$K_C = F(C,B,A) = \sum(7) + d(0,1,2,3)$

C \ BA	00	01	11	10
0	0 m_0	0 m_1	1 m_3	0 m_2
1	X m_4	X m_5	X m_7	X m_6

$J_C=F(C,B,A)=\sum(3)+d(4,5,6,7)$
$J_C=BA$

C \ BA	00	01	11	10
0	X m_0	X m_1	X m_3	X m_2
1	0 m_4	0 m_5	1 m_7	0 m_6

$K_C=F(C,B,A)= \sum(7)+d(0,1,2,3)$
$K_C=BA$

C \ BA	00	01	11	10
0	0 m_0	1 m_1	X m_3	X m_2
1	0 m_4	1 m_5	X m_7	X m_6

$J_B=F(C,B,A)= \sum(1,5)+d(2,3,6,7)$
$J_B=A$

C \ BA	00	01	11	10
0	X m_0	X m_1	0 m_3	1 m_2
1	X m_4	X m_5	0 m_7	1 m_6

$K_B=F(C,B,A)= \sum(3,7)+d(0,1,4,5)$
$K_B=A'$

C \ BA	00	01	11	10
0	1 m_0	X m_1	X m_3	1 m_2
1	1 m_4	X m_5	X m_7	1 m_6

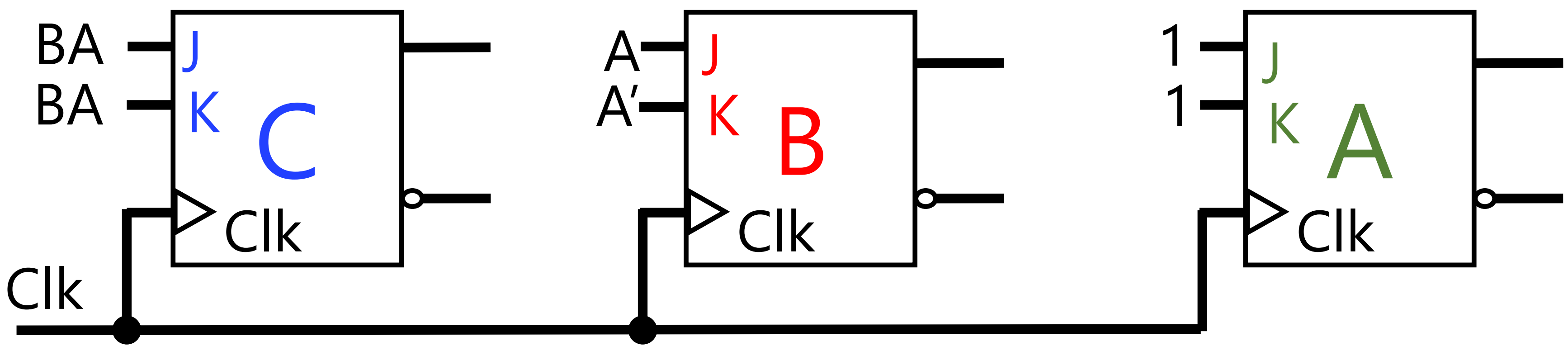
$J_A=F(C,B,A)=\sum(0,2,4,6)+d(1,3,5,7)$
$J_A=1$

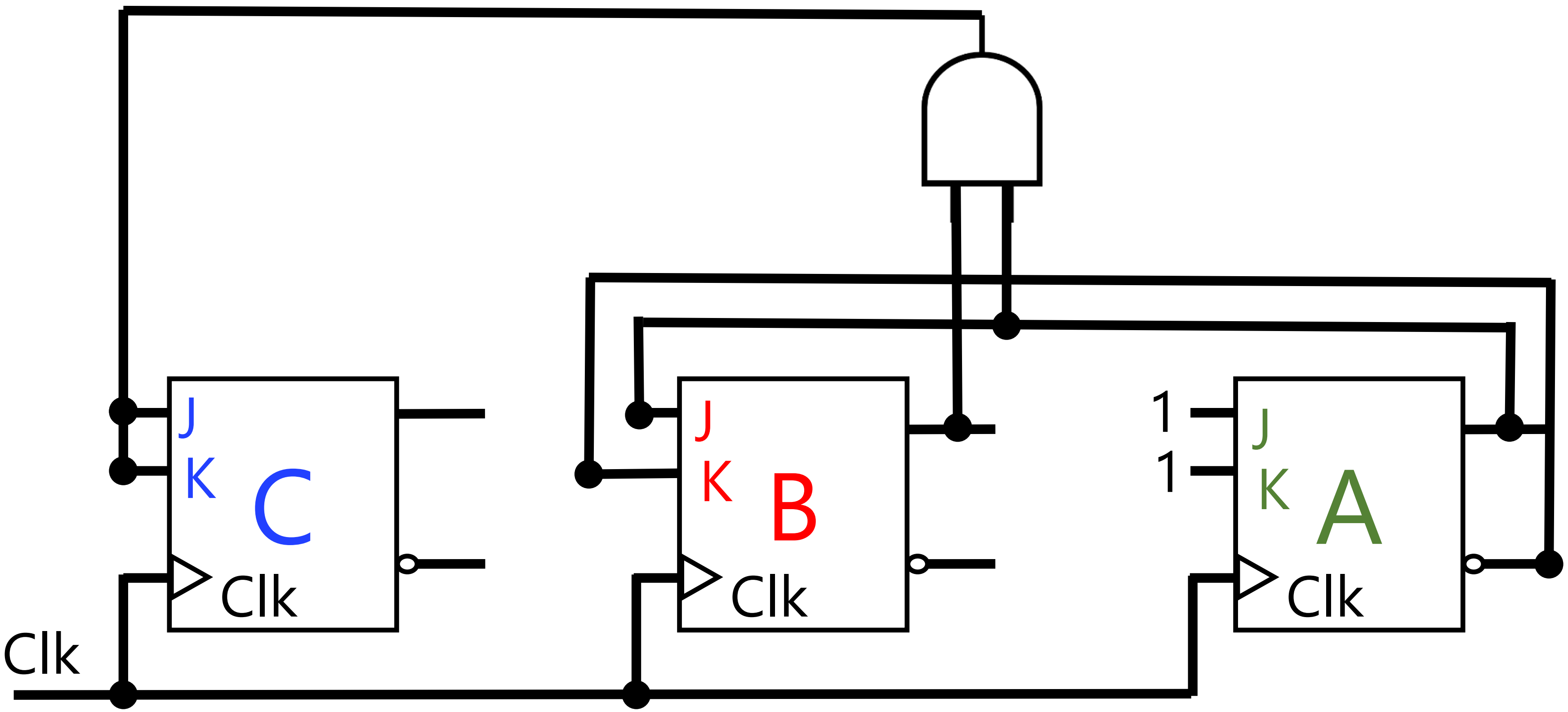
C \ BA	00	01	11	10
0	X m_0	1 m_1	1 m_3	X m_2
1	X m_4	1 m_5	1 m_7	X m_6

$K_A= F(C,B,A)=\sum(1,3,5,7)+d(0,2,4,6)$
$K_A=1$

Design

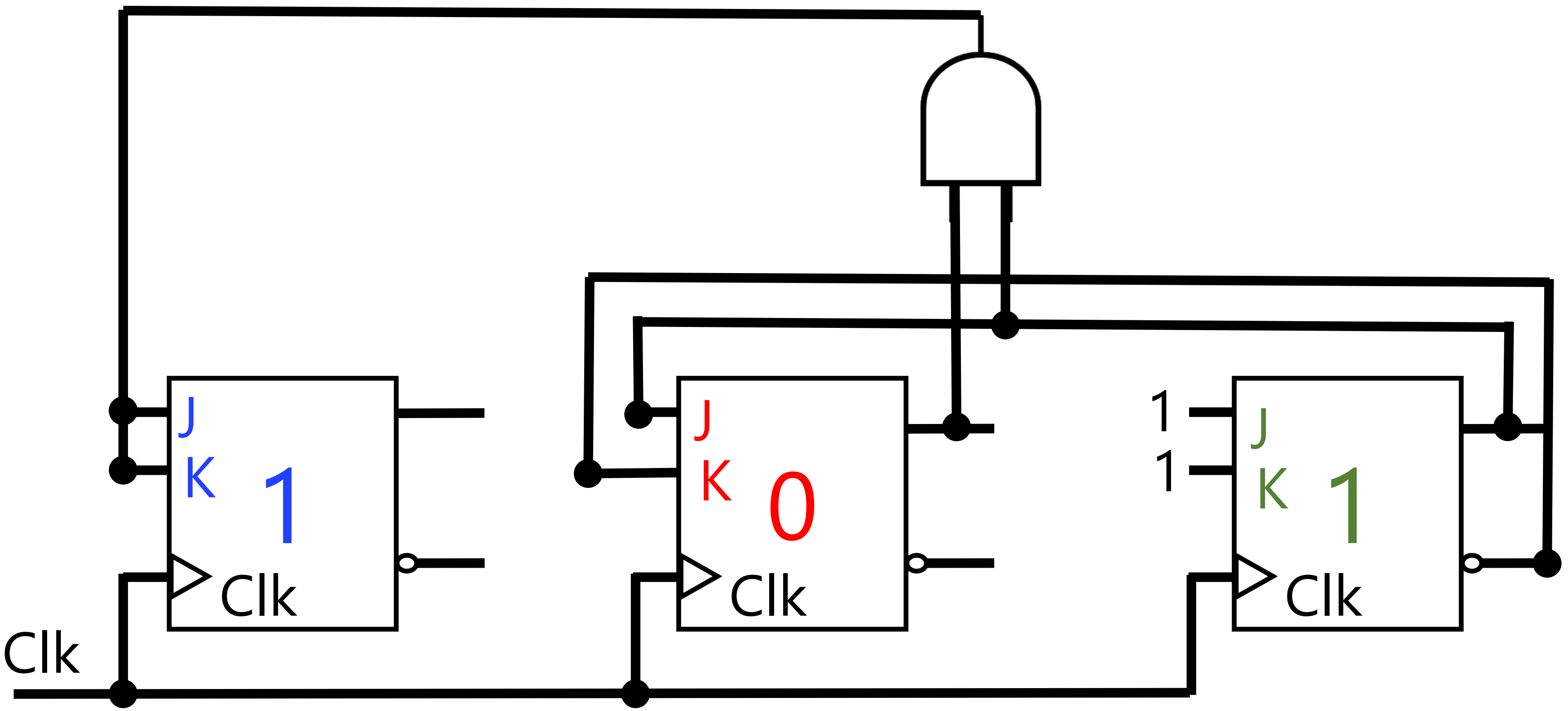
8. Draw/Sketch Logic Circuit


BA
BA
J
K
C
Clk
A
A'
J
K
B
Clk
1
1
J
K
A
Clk
Clk

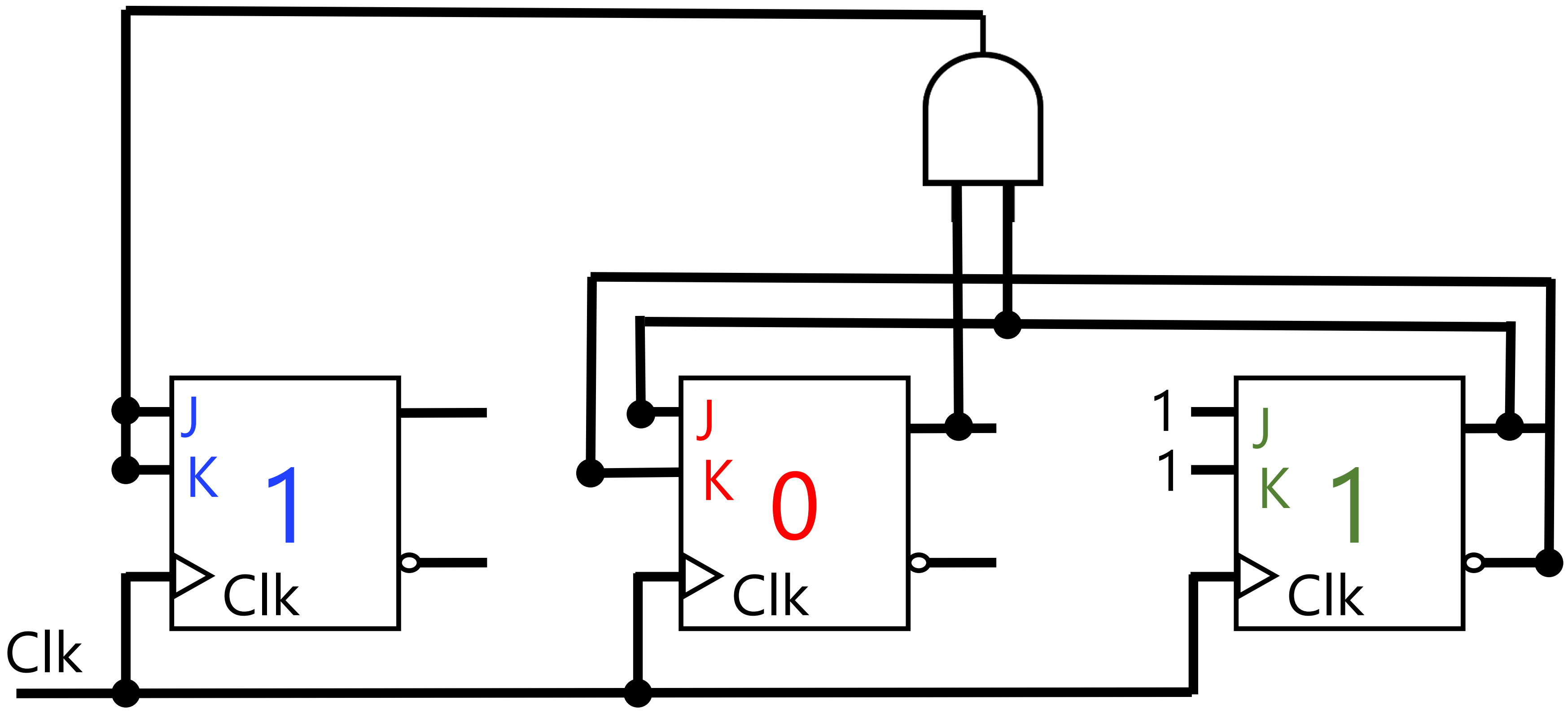

J
K
C
Clk
J
K
B
Clk
1
1
J
K
A
Clk
Clk

Design

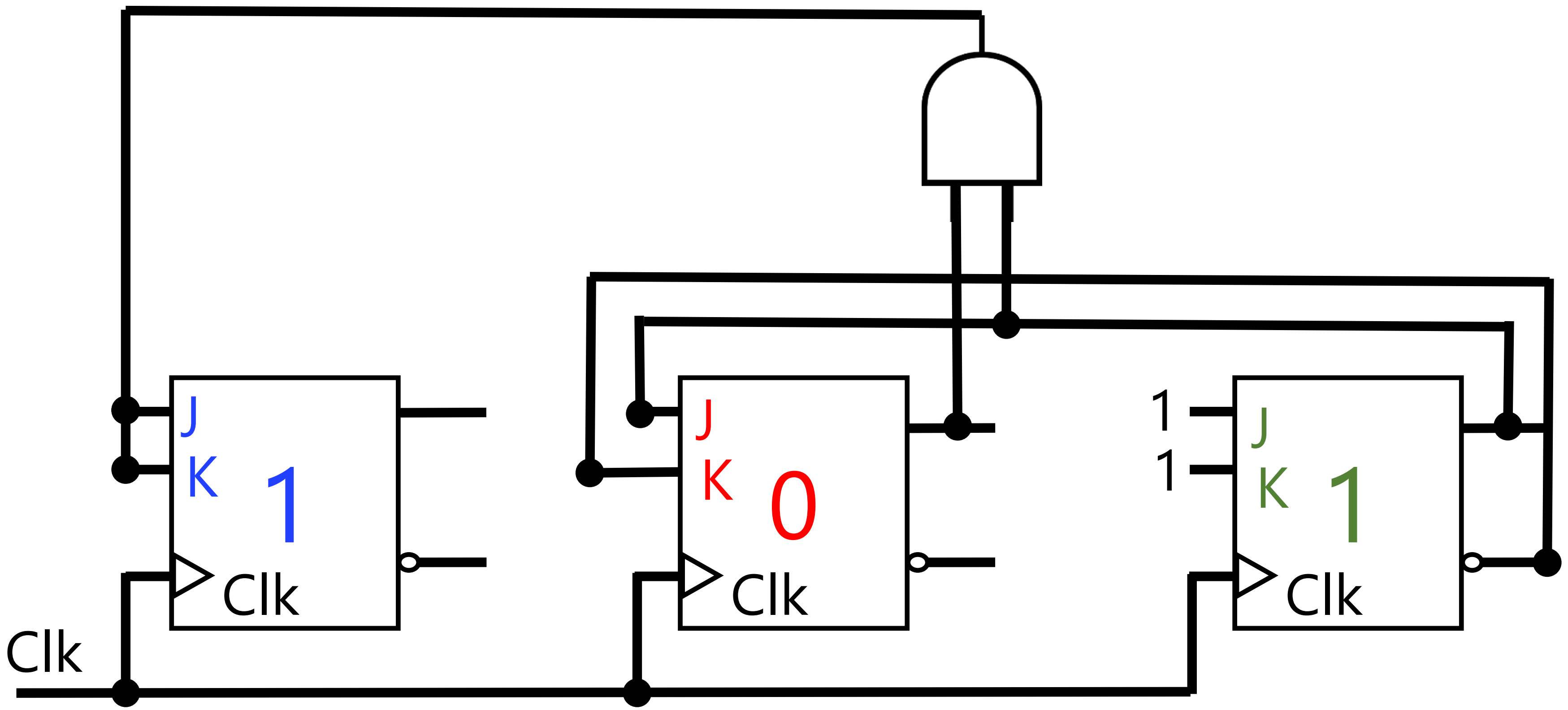
9. (Optional) Test



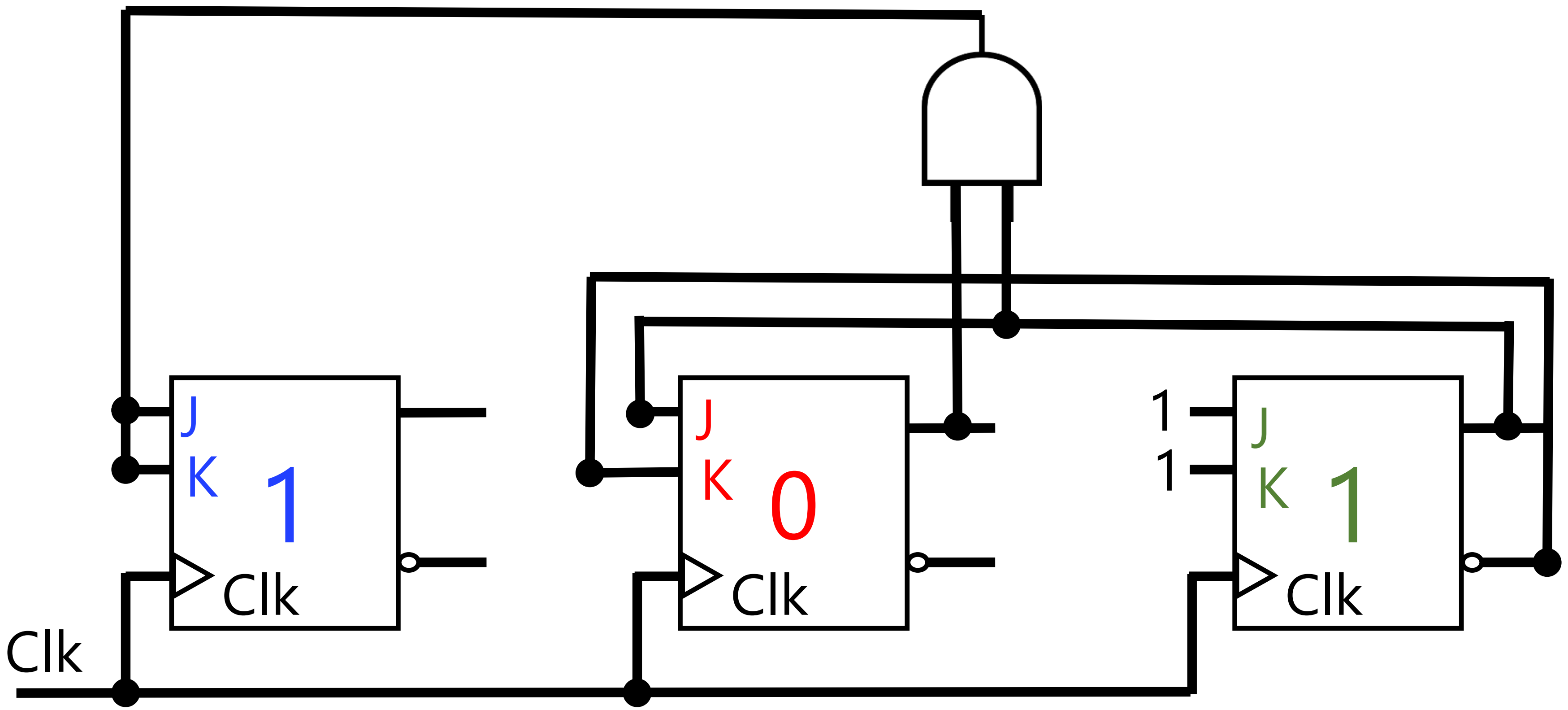
Q(T)			Q(T+1)		
C	B	A	C	B	A
1	0	1	?	?	?
5 → ?					



Q(T)			Q(T+1)		
C	B	A	C	B	A
1	0	1	?	?	?
5 → ?					

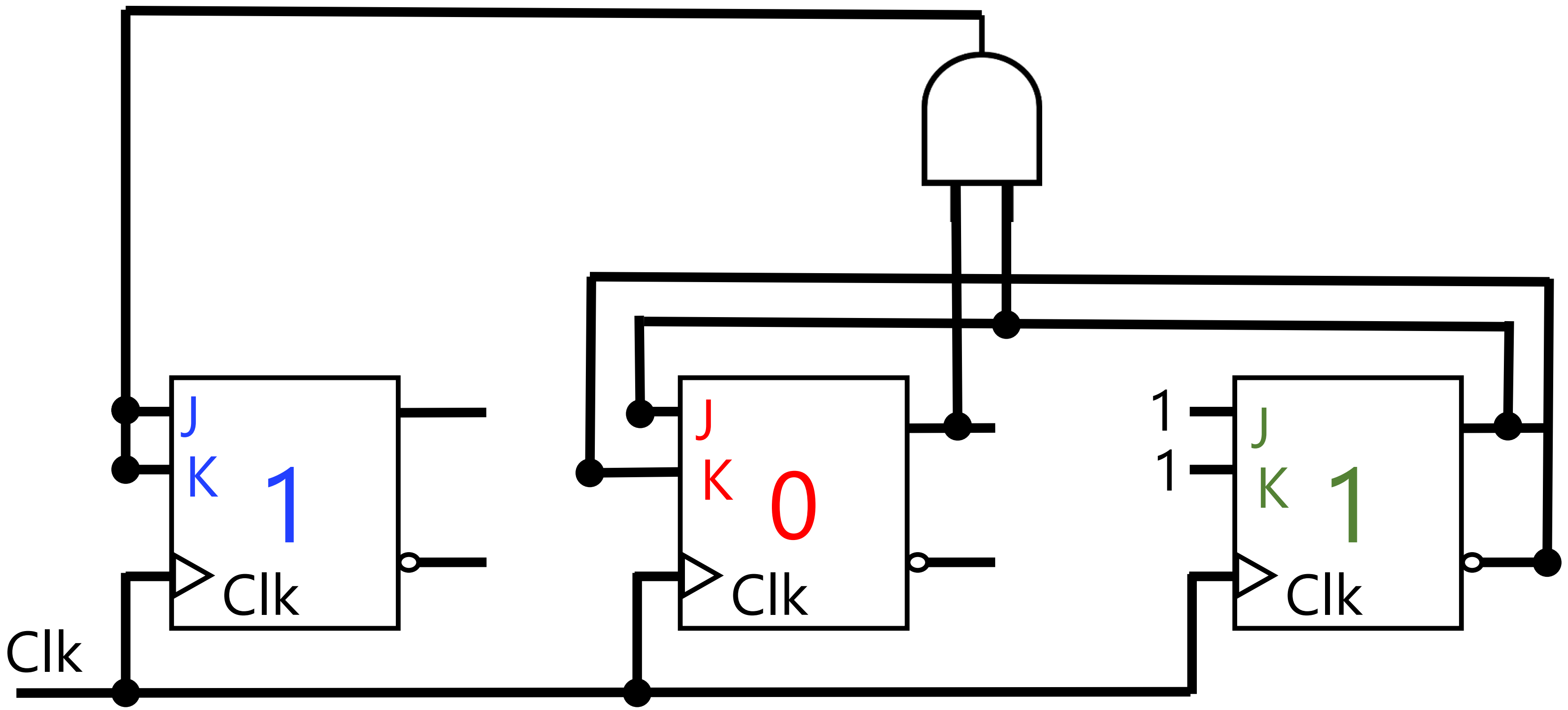


Q(T)			Q(T+1)		
C	B	A	C	B	A
1	0	1	?	?	$A=1, J_A=1, K_A=1$ ----------- Comp. → 0
5 → ?					

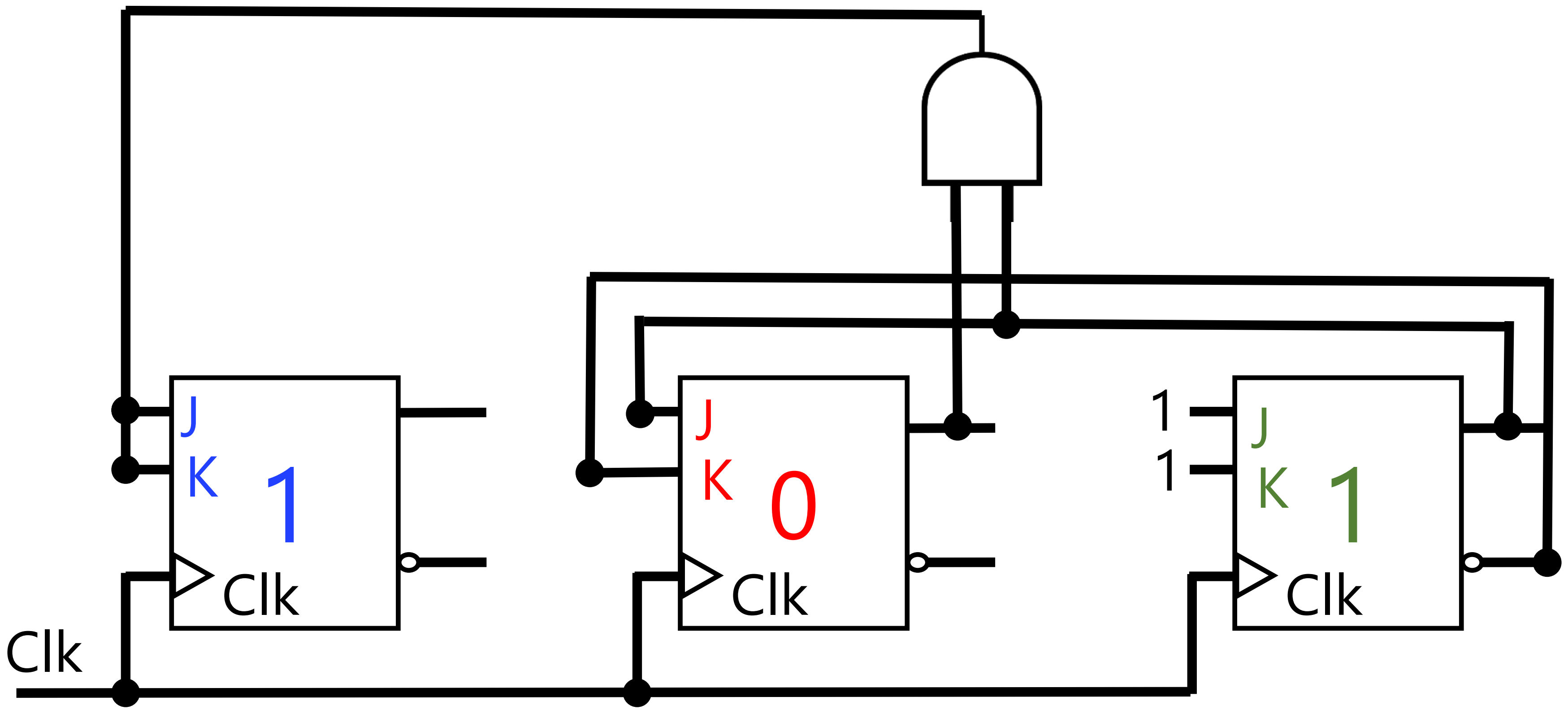


Q(T)			Q(T+1)		
C	B	A	C	B	A
1	0	1	?	?	0

5 → ?

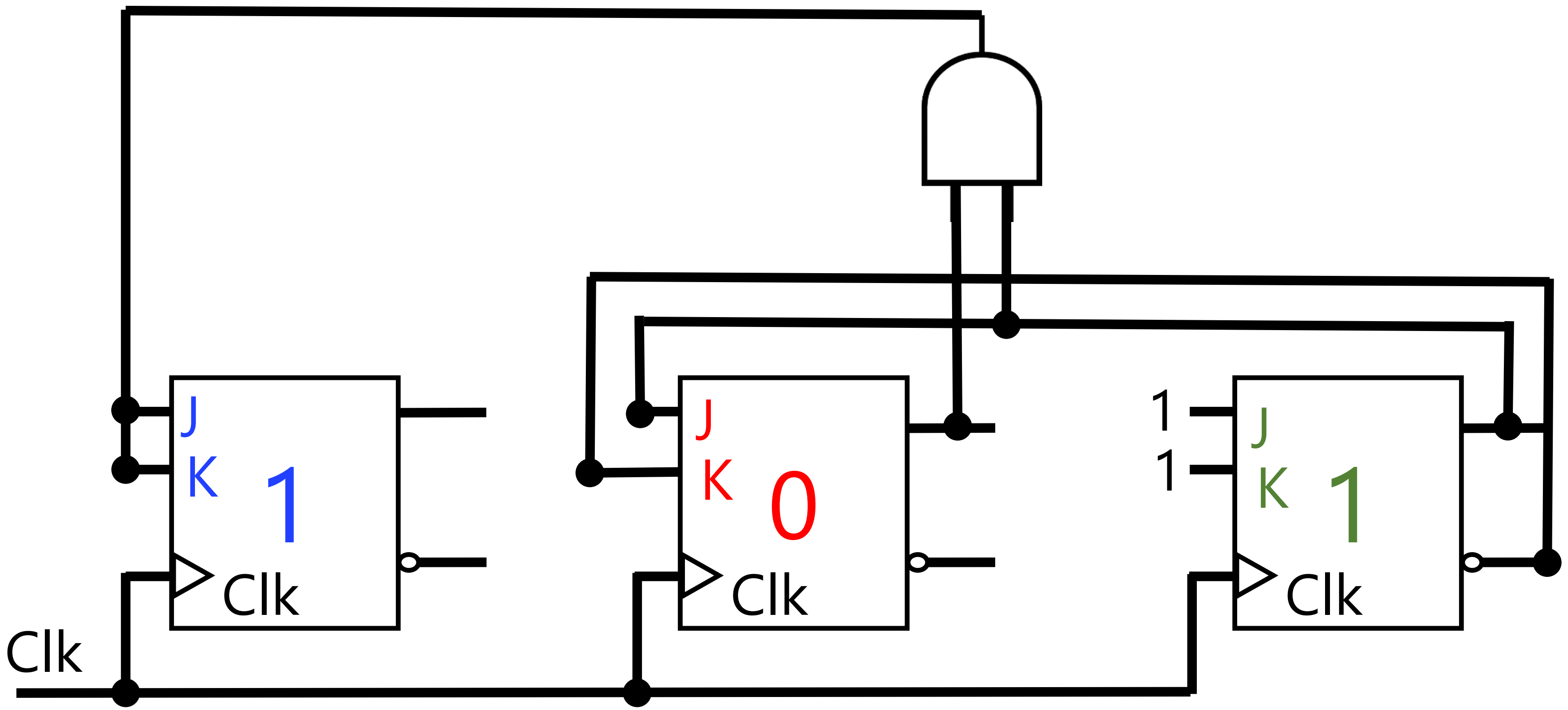


Q(T)			Q(T+1)		
C	B	A	C	B	A
1	0	1	?	B=0, J_B=A=1, K_B=A'=0 --------- Set → 1	0
5 → ?					

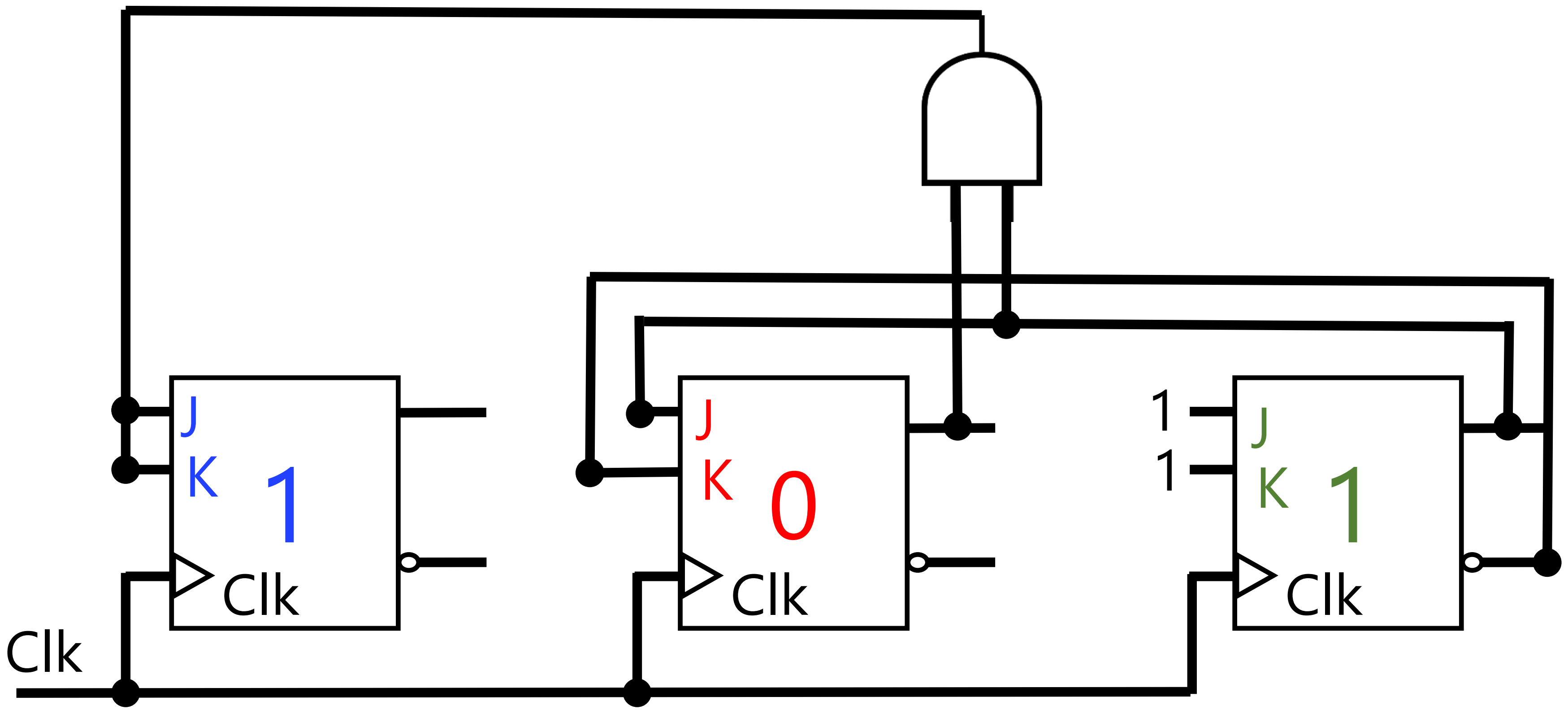


Q(T)			Q(T+1)		
C	B	A	C	B	A
1	0	1	?	1	0

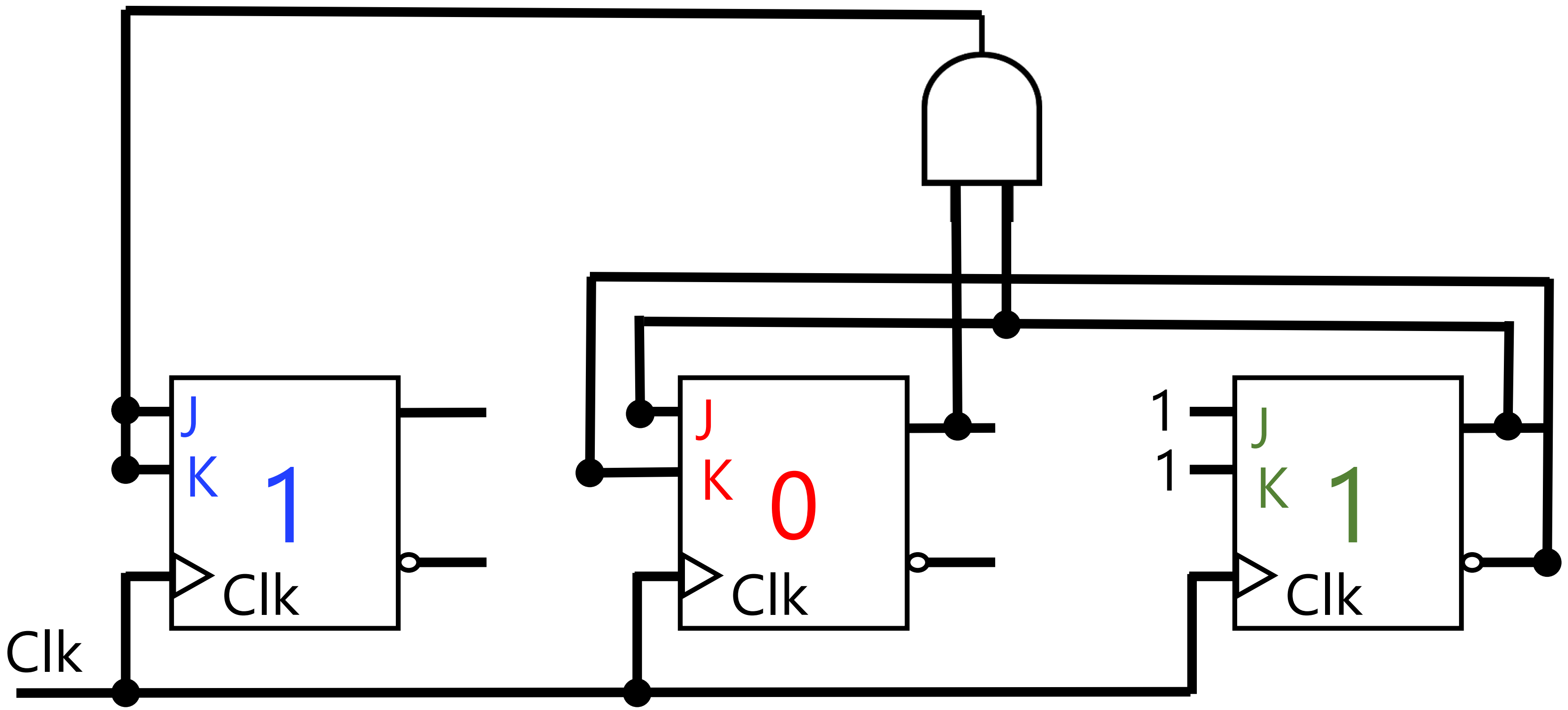
5 → ?



Q(T)			Q(T+1)		
C	B	A	C	B	A
1	0	1	?	1	0
5 → ?					



Q(T)			Q(T+1)		
C	B	A	C	B	A
1	0	1	C=1, J_C=BA=01=0 K_C=BA=01=0 ---------------- Store → 1	1	0
5 → ?					



Q(T)			Q(T+1)		
C	B	A	C	B	A
1	0	1	1	1	0
5 → 6					

Design (Recap)

0. Do we need combinational logic or sequential logic? Do we need memory?
1. How many storage (flip-flops)? #FF
2. Form the state (transition) diagram
3. Form the state table
4. Fill the state table
5. What type of storage (flip-flop)? RS, D, T, JK, or Mixed
6. Input (*excitation*) equations for each FF
7. Minimization of input (*excitation*) equations
8. Draw/Sketch Logic Circuit
9. (Optional) Test

Design (Recap)

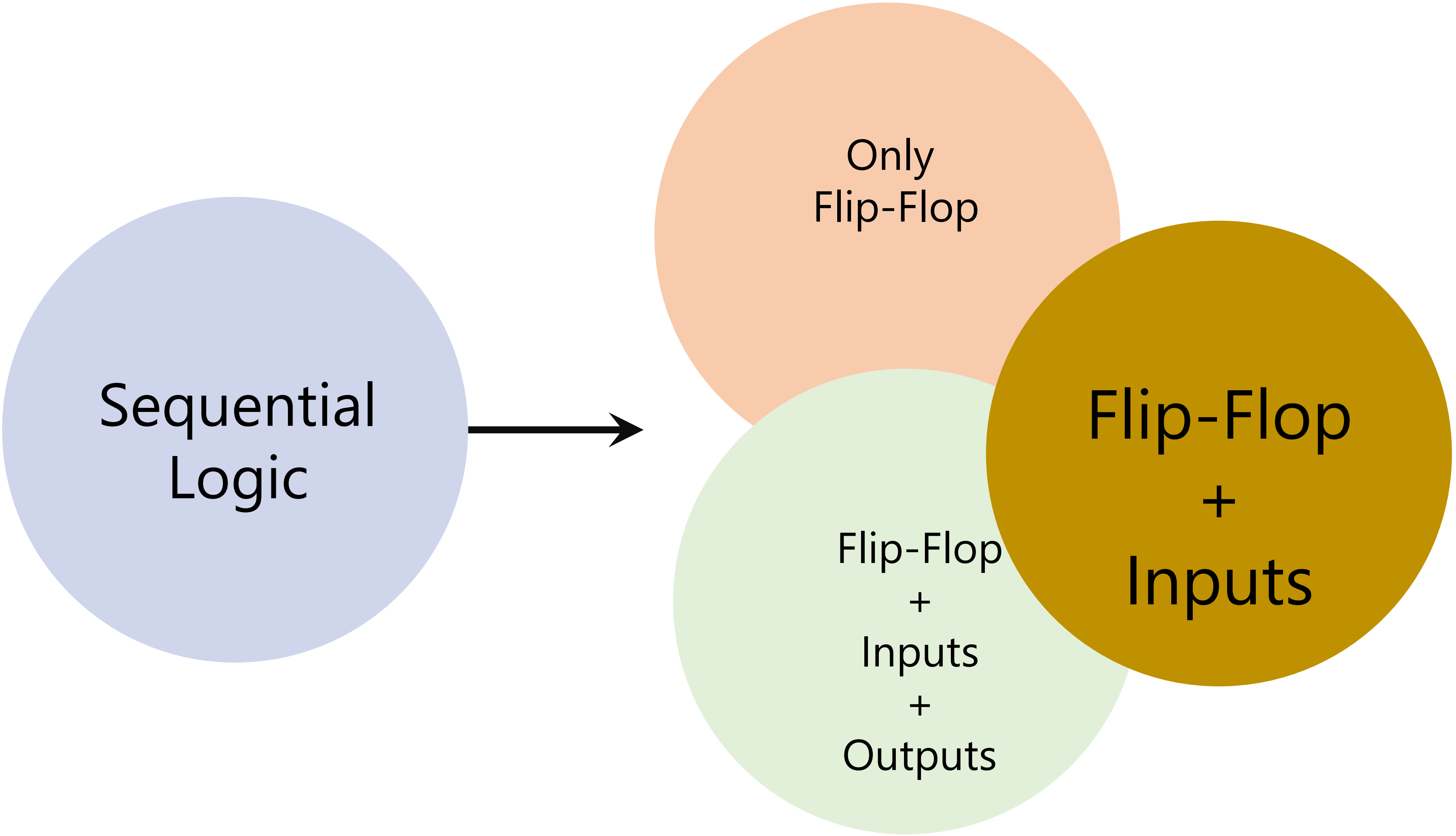
0. Do we need combinational logic or sequential logic? Do we need memory?
1. How many storage (flip-flops)? #FF
2. Form the state (transition) diagram
3. Form the state table
4. Fill the state table
5. What type of storage (flip-flop)? RS, D, T, JK, or Mixed
6. Input (*excitation*) equations for each FF
7. Minimization of input (*excitation*) equations
8. Draw/Sketch Logic Circuit
9. (Optional) Test

Design (Advanced)

0. Do we need combinational logic or sequential logic? Do we need memory?
1. How many storage (flip-flops)? #FF
2. Form the state (transition) diagram
2.1. State Reduction
3. Form the state table
4. Fill the state table
5. What type of storage (flip-flop)? RS, D, T, JK, or Mixed
6. Input (*excitation*) equations for each FF
7. Minimization of input (*excitation*) equations
8. Draw/Sketch Logic Circuit
9. (Optional) Test

Theory of Automata

COMP-2140: Computer Languages, Grammars, and Translators



Sequential Logic
Only Flip-Flop
Flip-Flop + Inputs + Outputs
Flip-Flop + Inputs