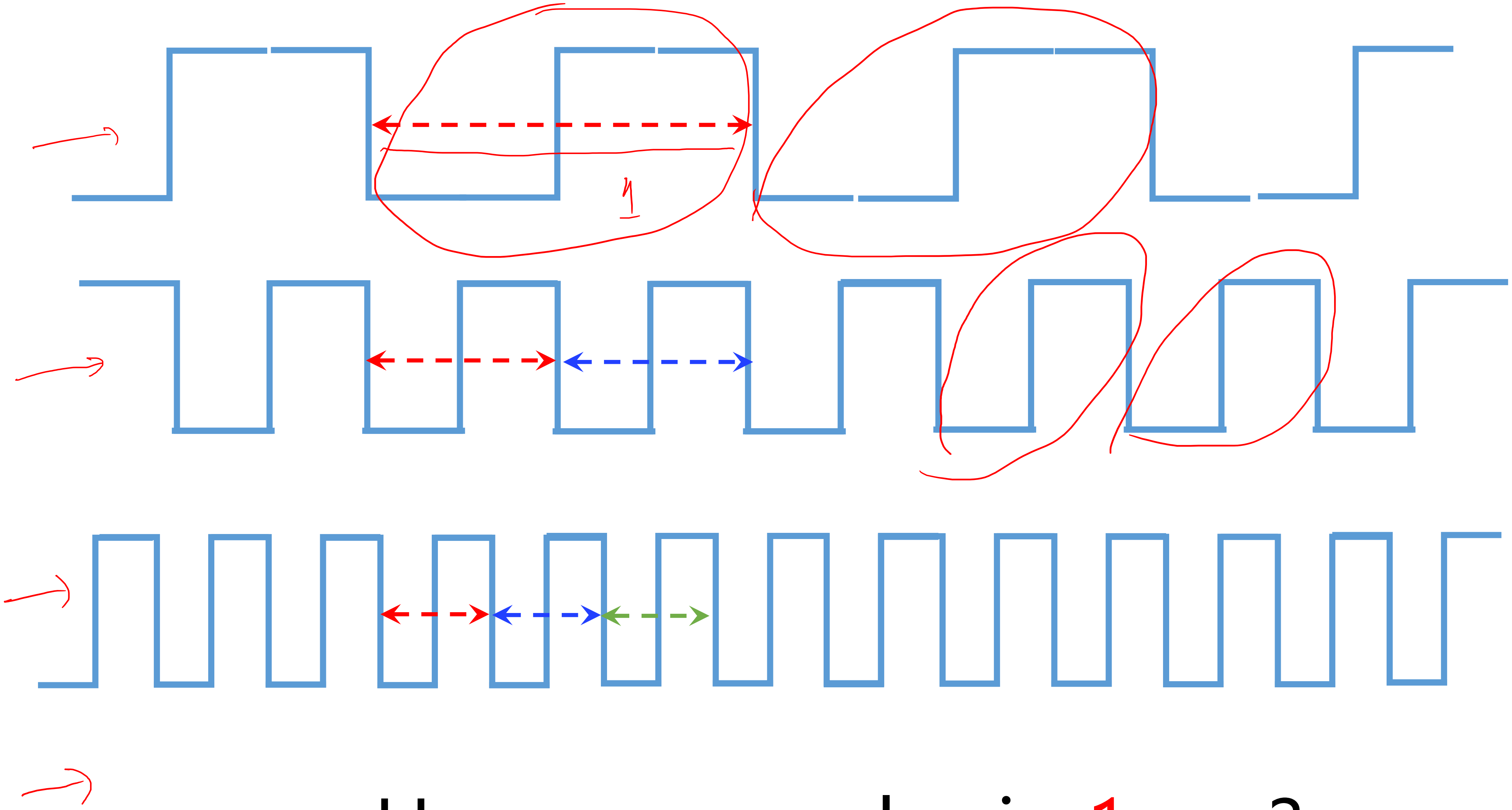
Student Evaluations of Teaching (SET) are now available. This is your chance to provide feedback on your courses!
SETs do not apply for Continuing Education courses such as IB,AQ, and ELIP courses.
Instructions for completing SET: http://ask.uwindsor.ca/app/answers/detail/a_id/176

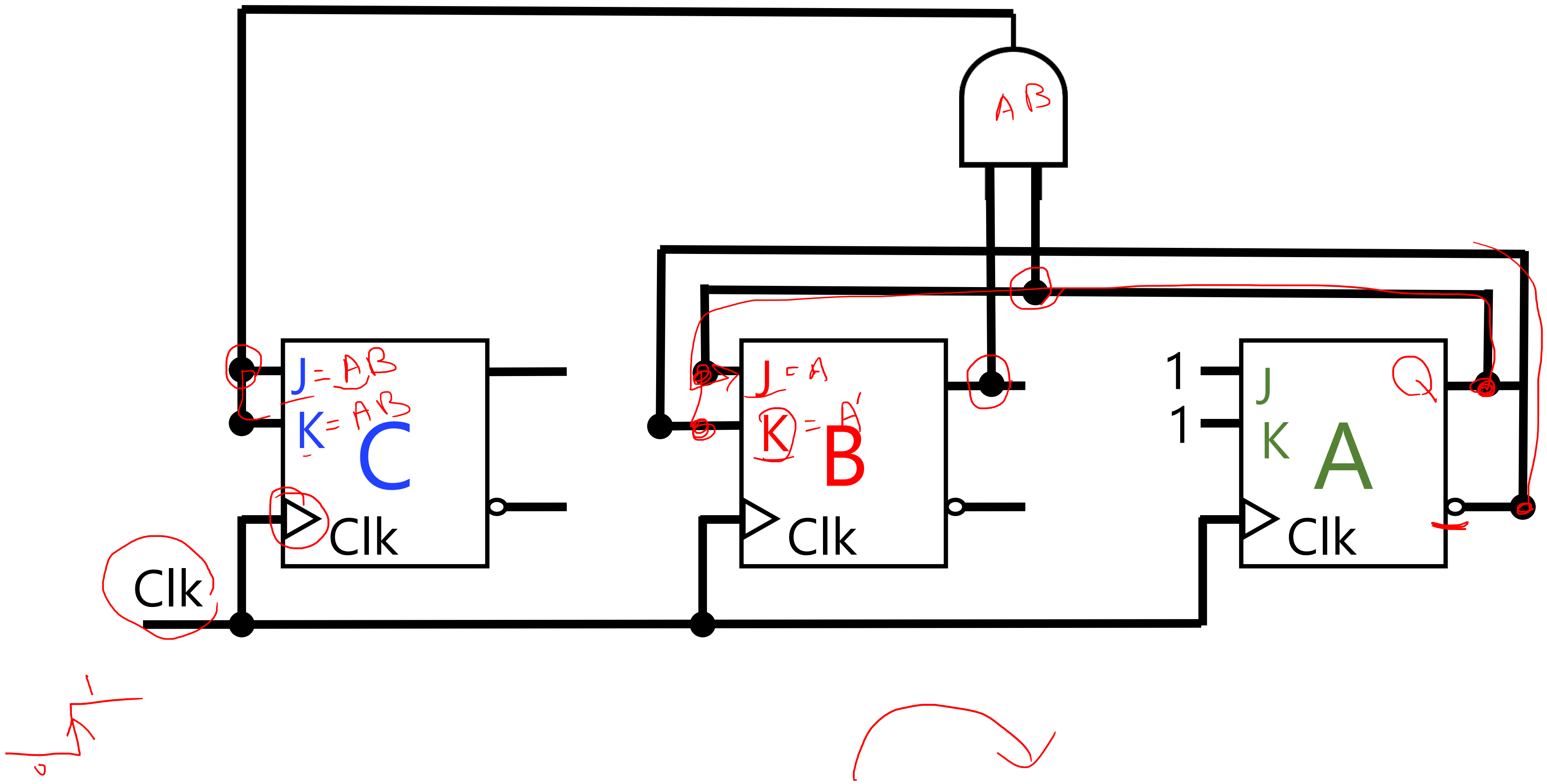
Clock (revisited)

shortened as *clk*

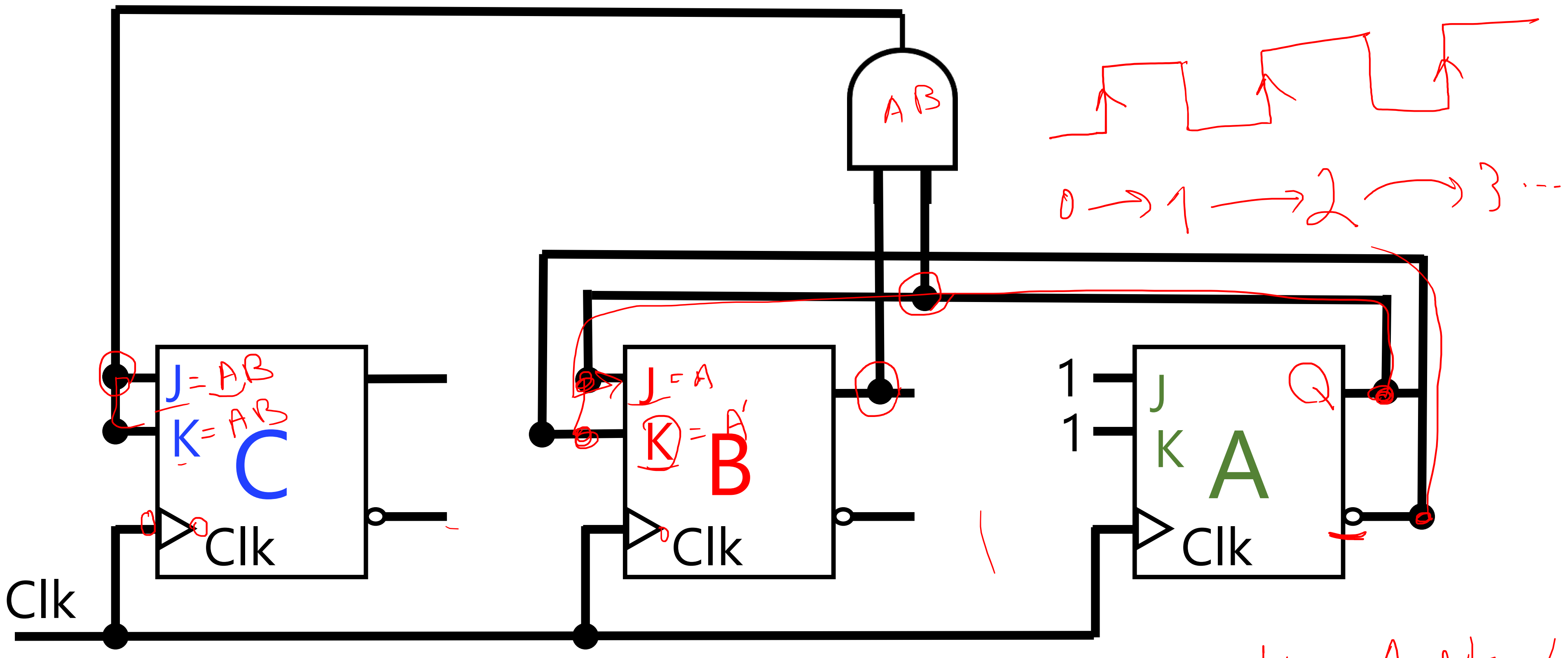
timing device that generates a train of pulses



How many pulse in 1 sec?



Counter 0 to 7

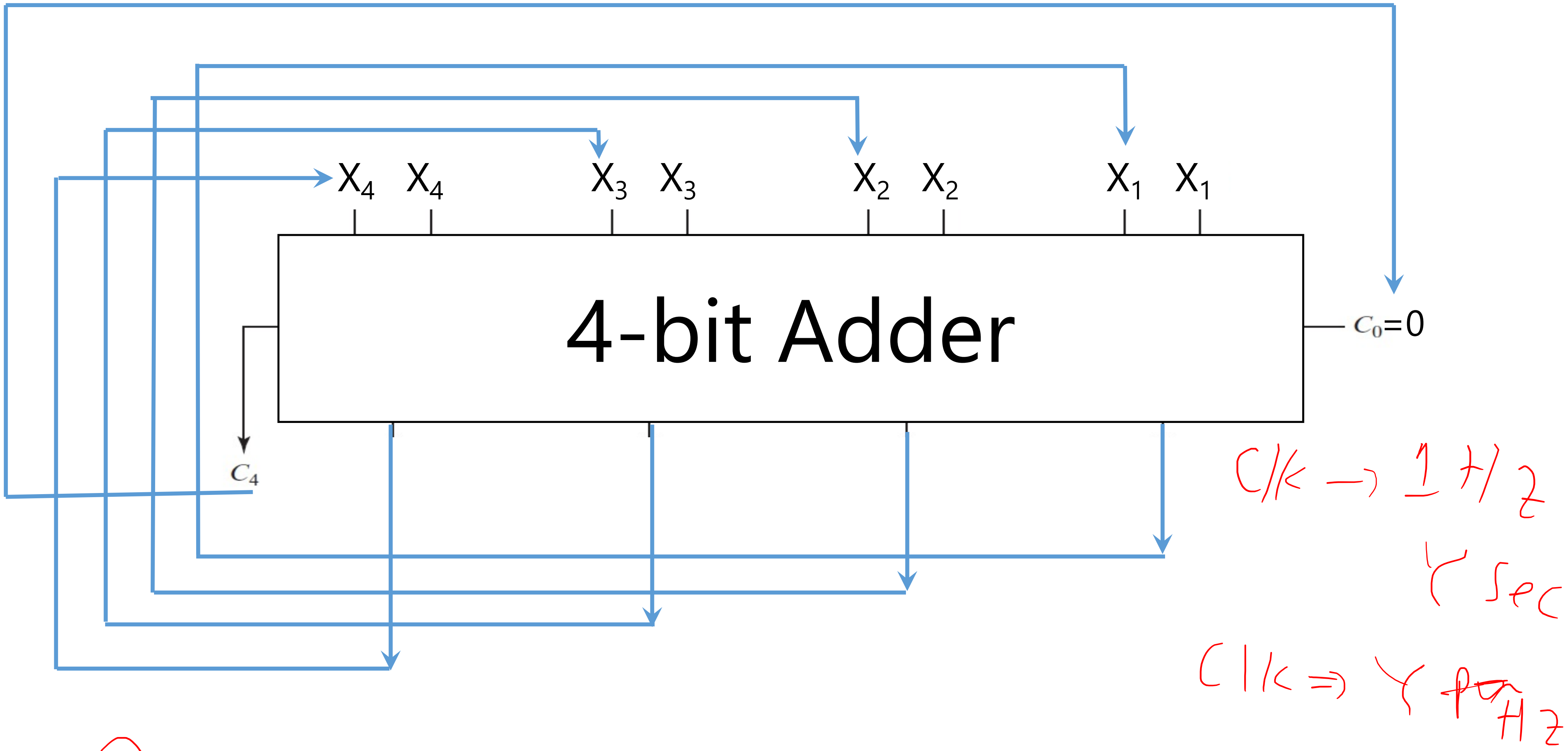


clk → 1 puls/se
→ 1 Hz
⇒ $\Delta t = 7$ sec

Counter 0 to 7

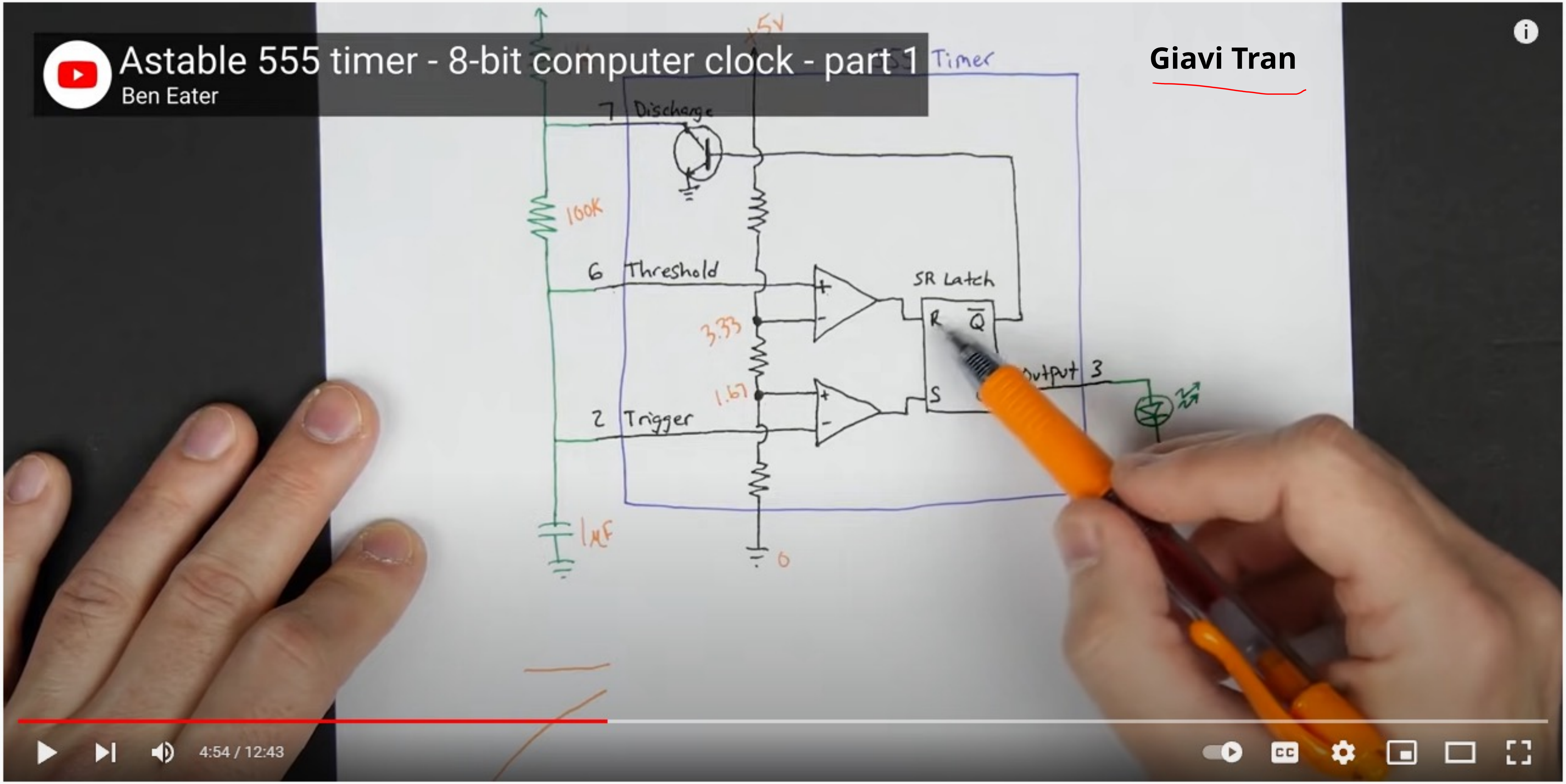
How many pulses? Elapsed time?

clk → 7 puls/se
→ 7 Hz



$X \times Y = X + \ldots + X$ → When to stop?

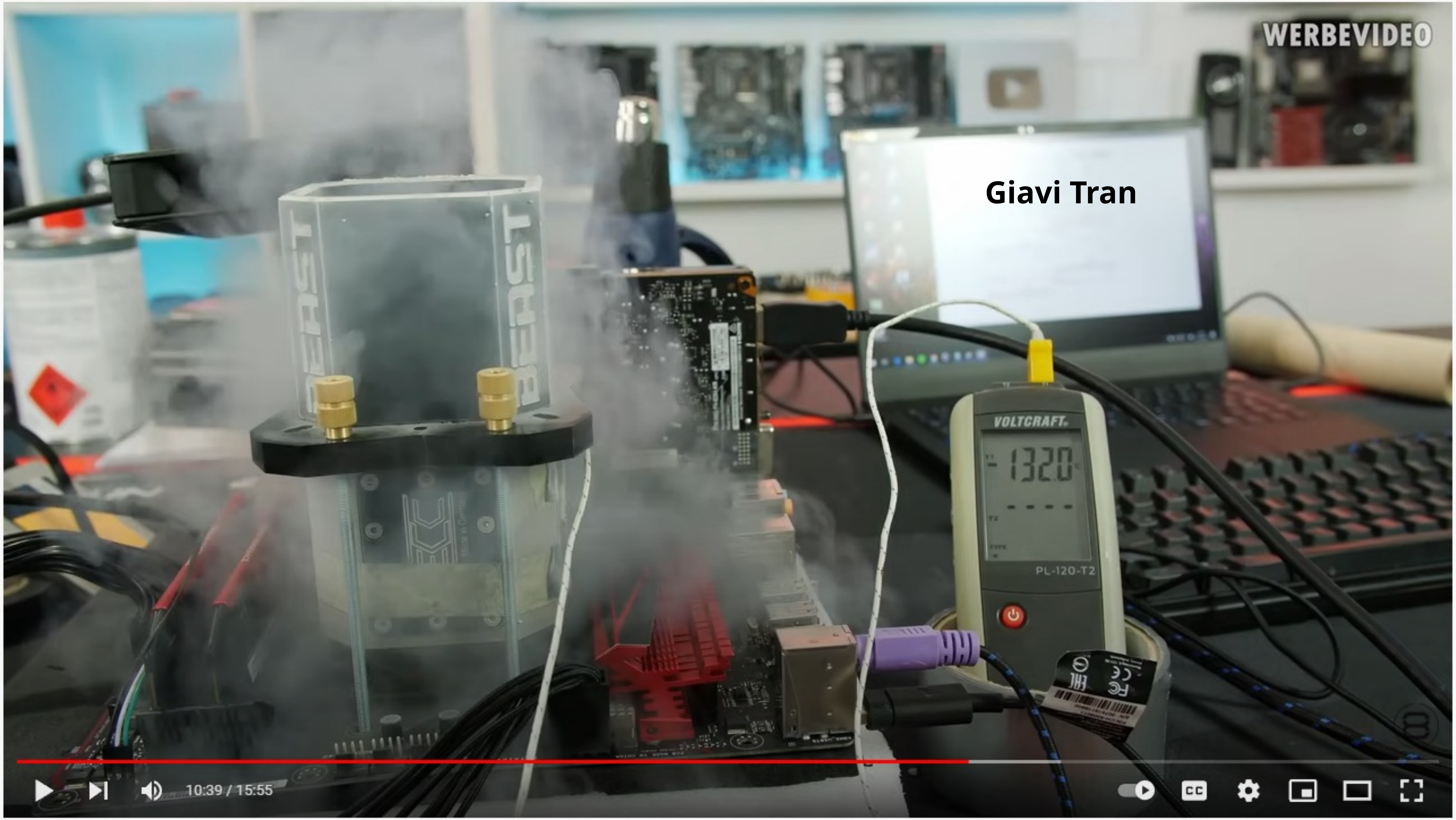
Feedback → Sequential Logic



Water Cooling a TI-84 Graphing Calculator!

3,263,682 views • May 27, 2020

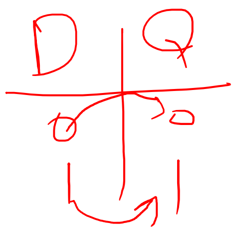
134K DISLIKE SHARE CLIP SAVE ...

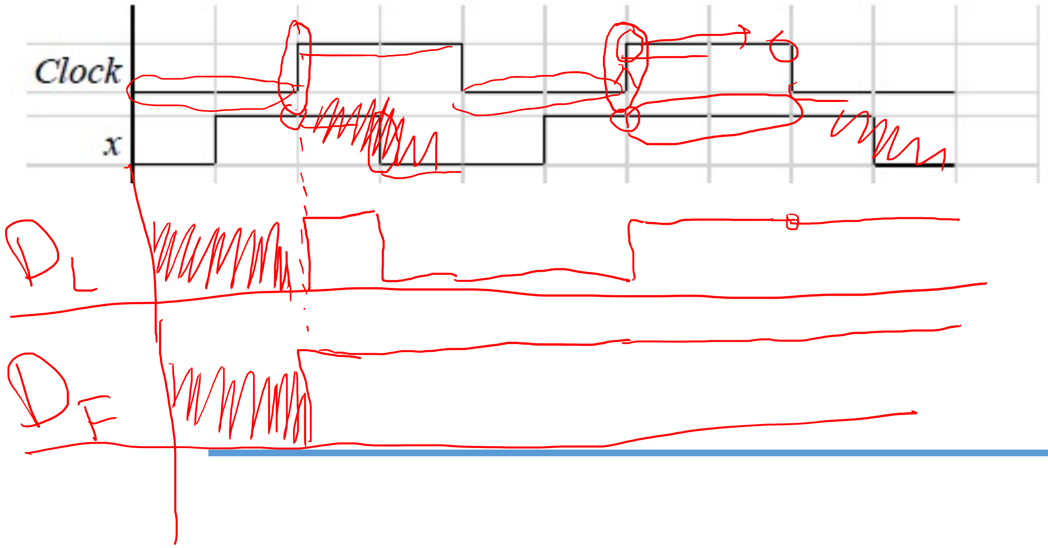


We Overclocked an AMD CPU to over 8.1 GHz !!

3,138,659 views • Apr 22, 2020

71K DISLIKE SHARE THANKS CLIP SAVE ...





Latch

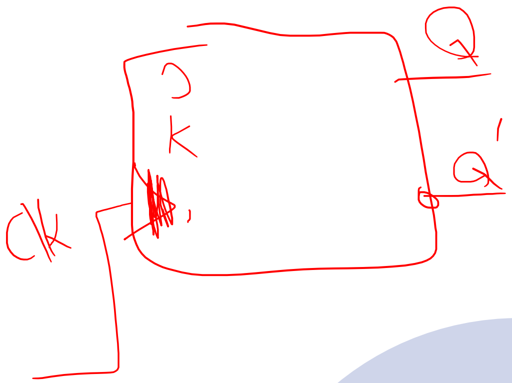
Active high (defa-
low

vs.

FlipFlop

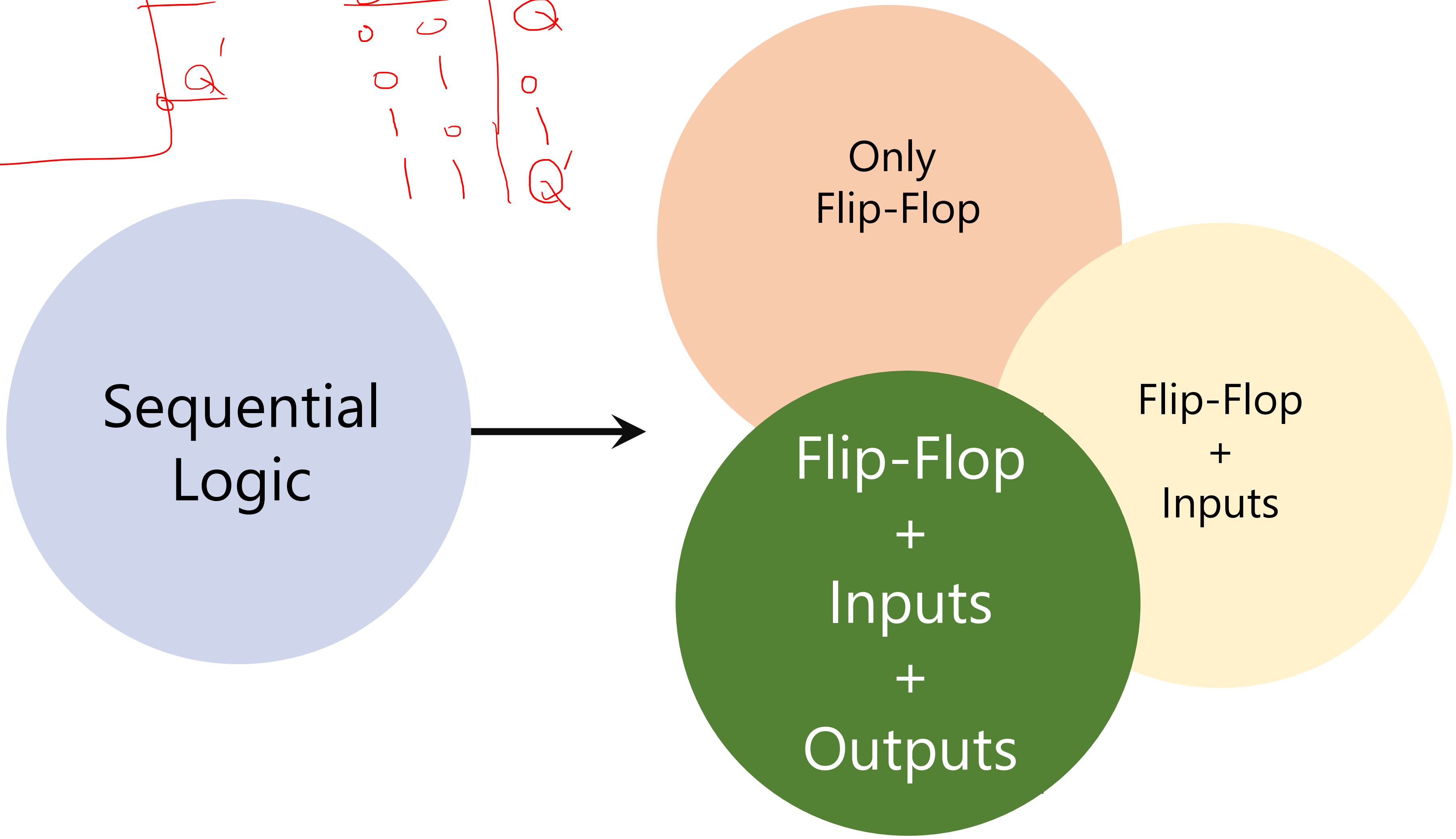
Pos edges (defa)
neg edge
both edges

Practice!

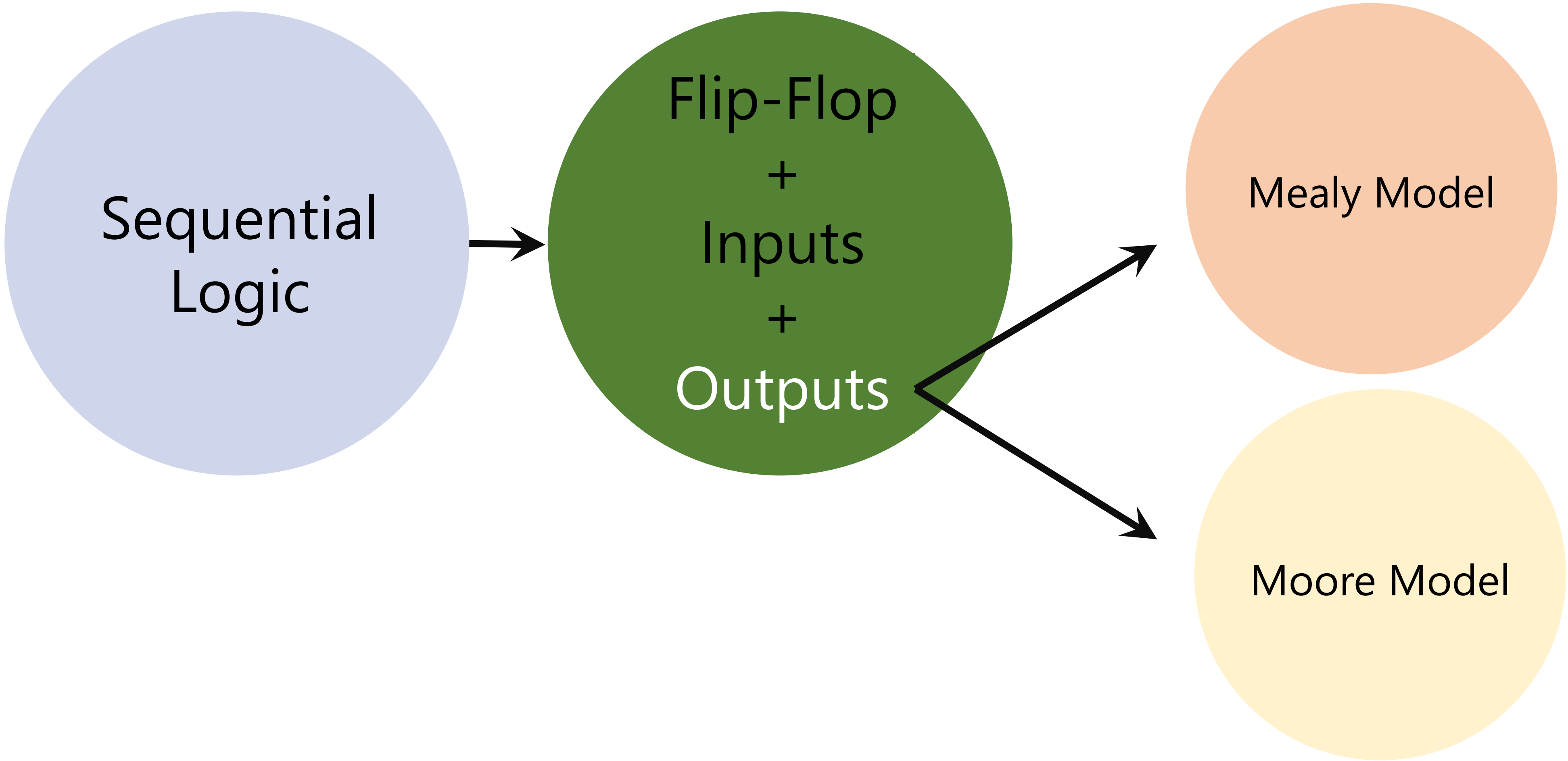


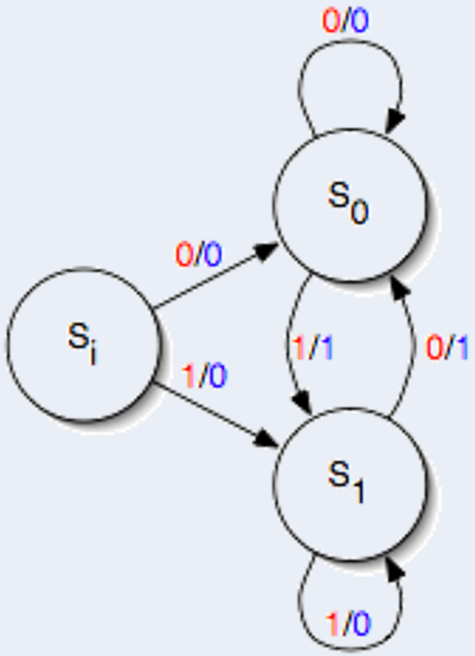
S R

J	K	Q
0	0	Q
0	1	0
1	0	1
1	1	Q'



Analysis
vs.
Design


Sequential Logic
Flip-Flop
+
Inputs
+
Outputs
Mealy Model
Moore Model

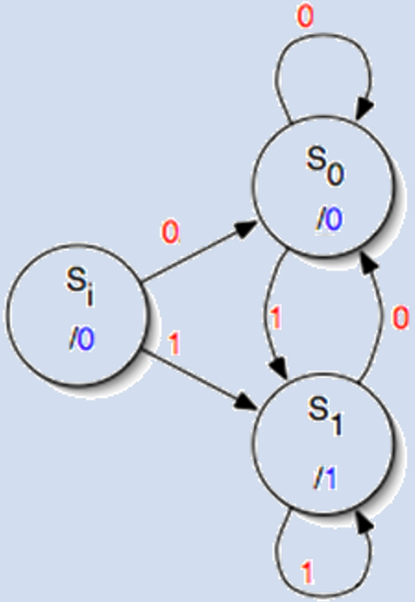


George H. Mealy
(1927 – 2010)
Mathematician and Computer Scientist
Invented Mealy Machine
Also a pioneer of modular programming

Outputs = Function(Current State, Inputs)

Edward Forrest Moore
(1925 – 2003)
Mathematician and Computer Scientist
Inventor of the Moore Machine
Also an early pioneer of artificial life

Outputs = Function(Current State, ~~Inputs~~)



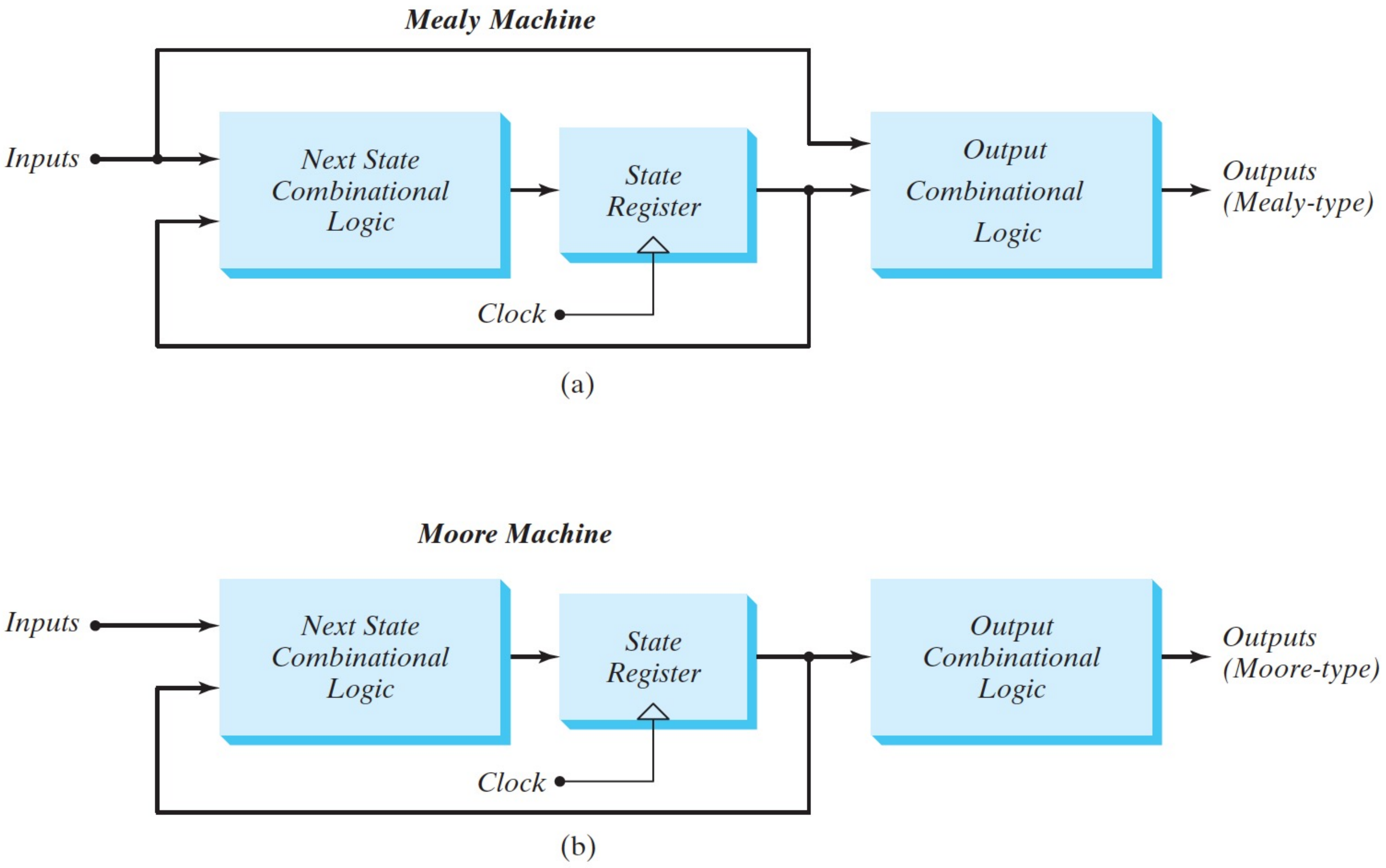
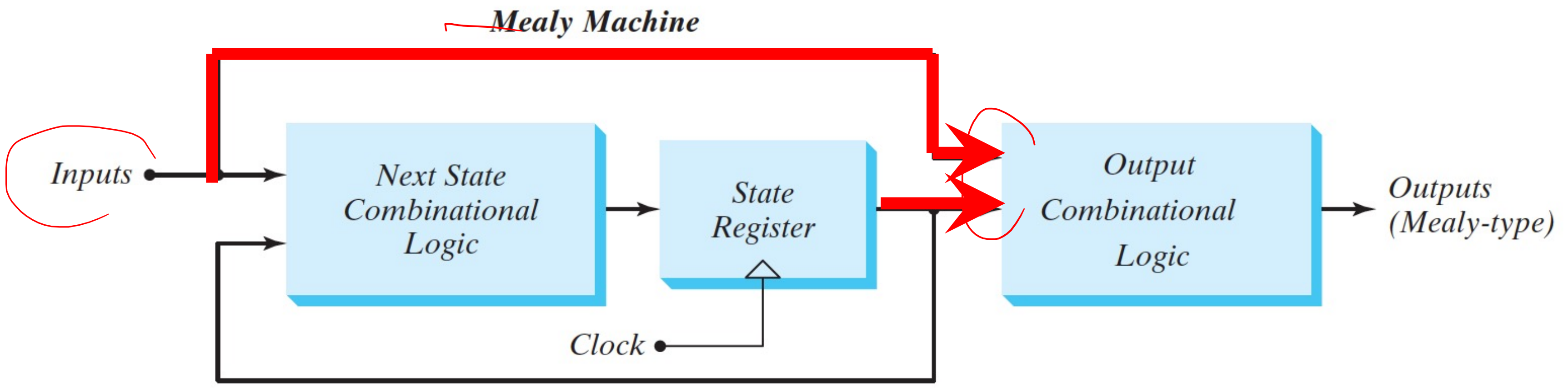
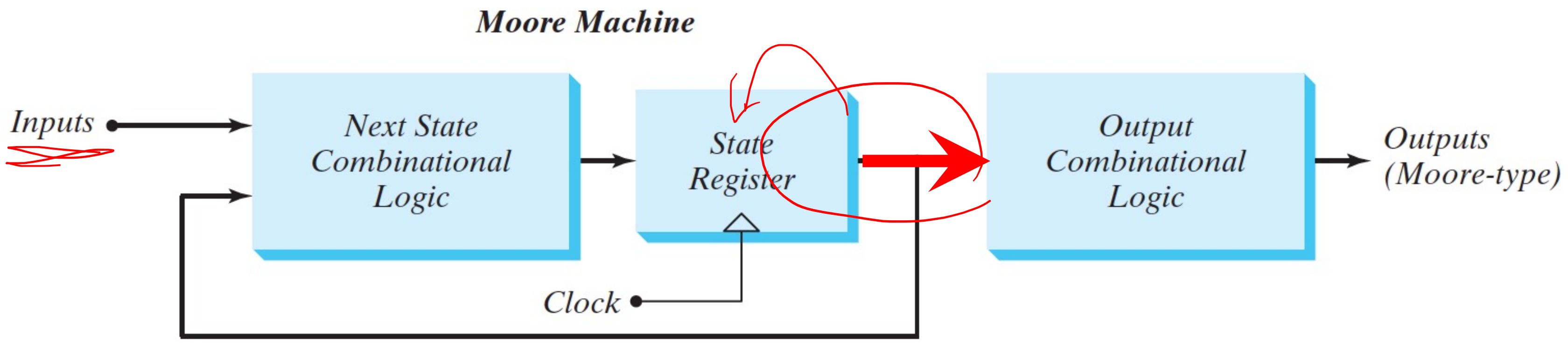


FIGURE 5.21
Block diagrams of Mealy and Moore state machines



(a)



(b)

FIGURE 5.21
Block diagrams of Mealy and Moore state machines

Sequential Logic

Flip-Flop
+
Inputs
+
Outputs

Mealy Model

Analysis (Moore model in output)

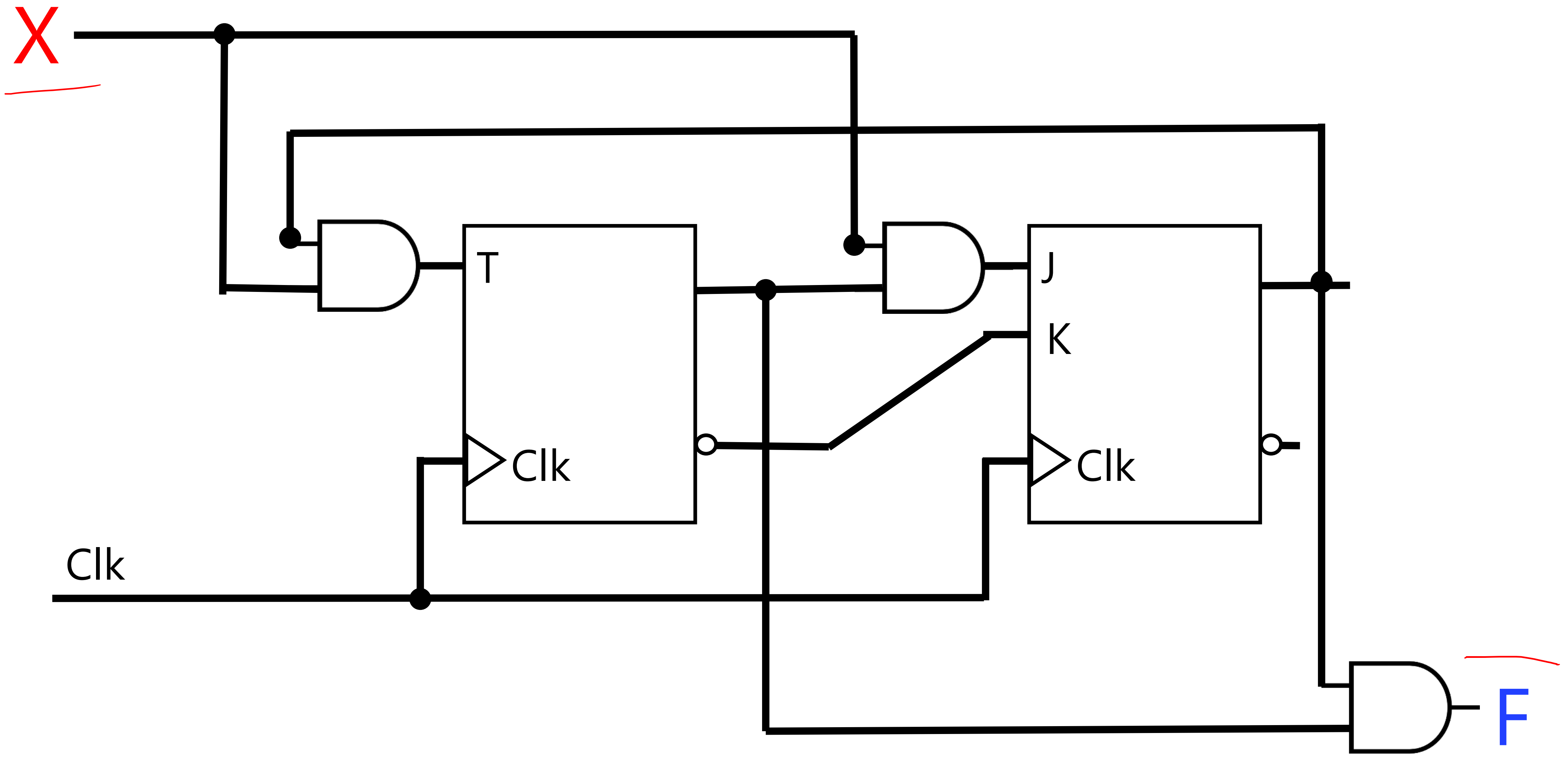
by an example

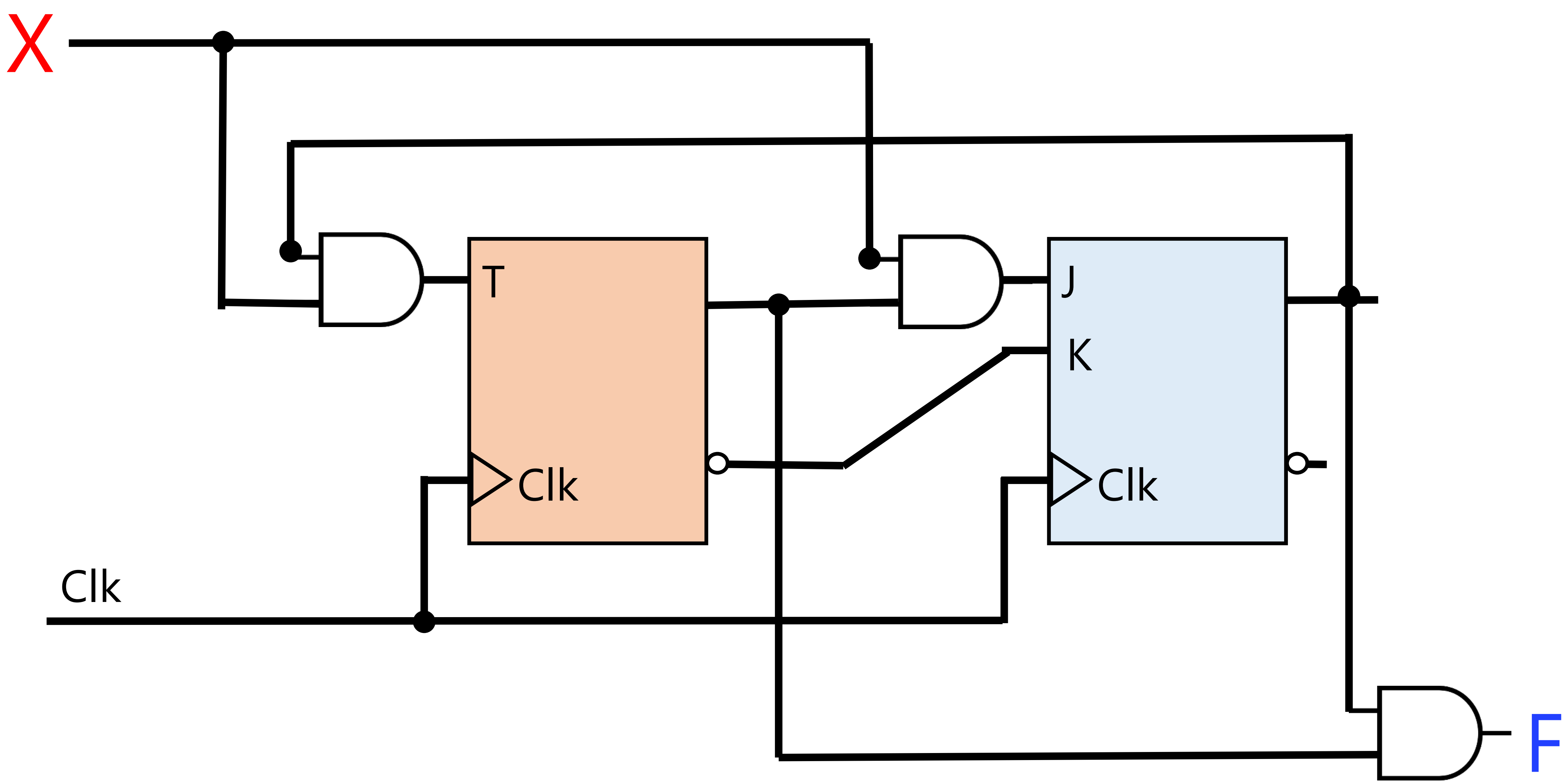
Analysis (Recap)

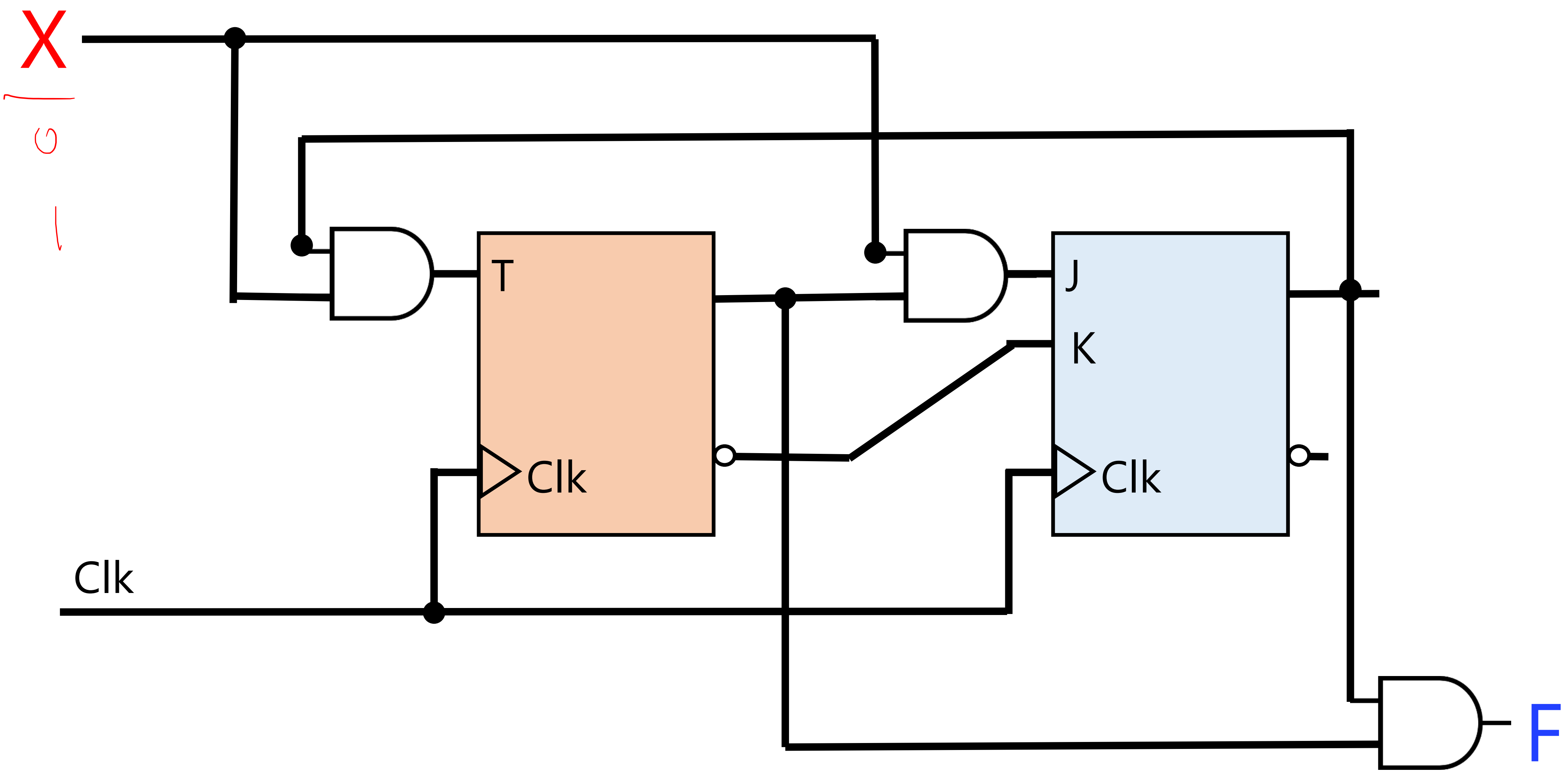
0. Is the circuit sequential or combinational? Any FF or feedback → Sequential
1. What are the flip-flops? RS, D, T, JK, or mixed (e.g., 2 JK, 1 RS, …)
2. What are the state combinations? $2^{\#FF}$
3. Form "State" table:
   a) Columns: for each FF, two columns:
      - one for current state,
      - 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 (+ Input + Moore Model Output)

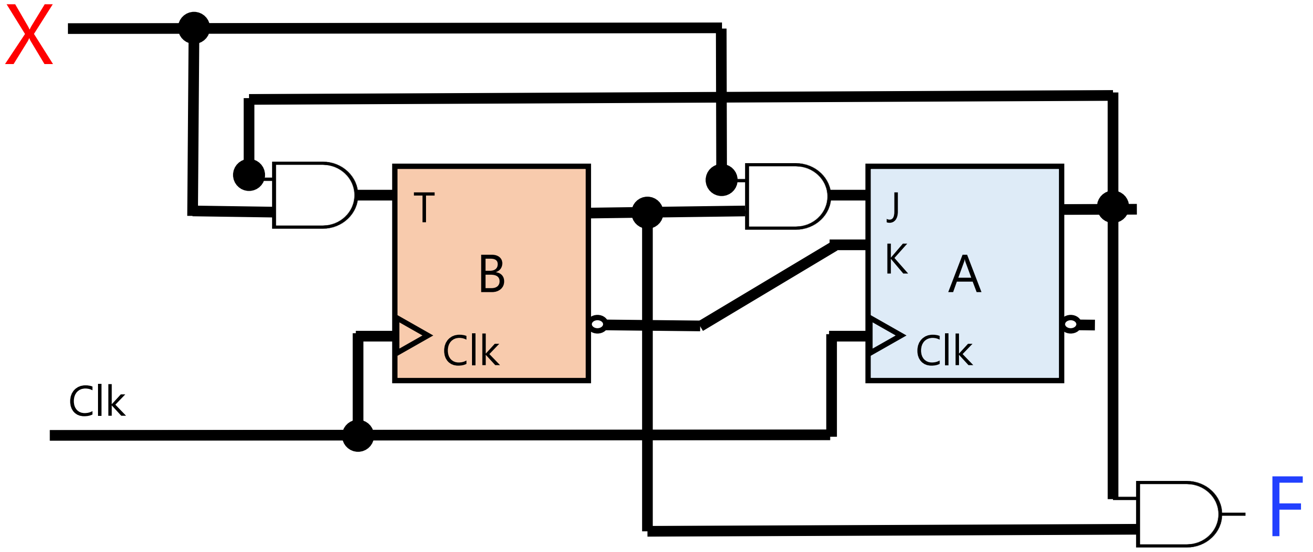
0. Is the circuit sequential or combinational? Any FF or feedback → Sequential
1. What are the flip-flops? RS, D, T, JK, or mixed (e.g., 2 JK, 1 RS, ...)
2. What are the state combinations? $\cancel{2}^{\#FF + \#inputs}$
3. Form "State" table:
   a) Columns: for each FF, two columns:
      - one for current state,
      - one for next state
   b) Rows: for each state combination
      - In total: $\cancel{2}^{\#FF}$
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


X
T
Clk
J
K
Clk
Clk
F


X
T
Clk
J
K
Clk
Clk
F

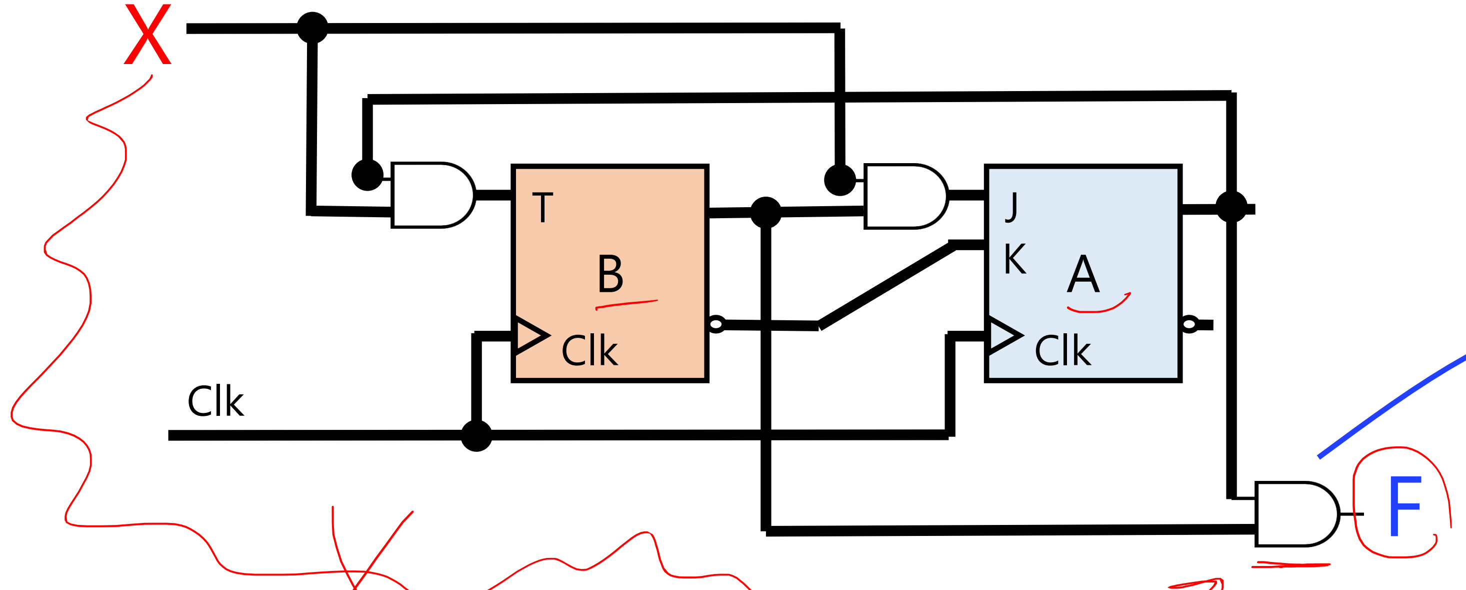


#FFs + #Inputs = 2+1 $\rightarrow$ 2^3 = 8 combinations



Inputs	Q(T)		Q(T+1)		Outputs
X	B	A	B	A	F

Alternative State Table



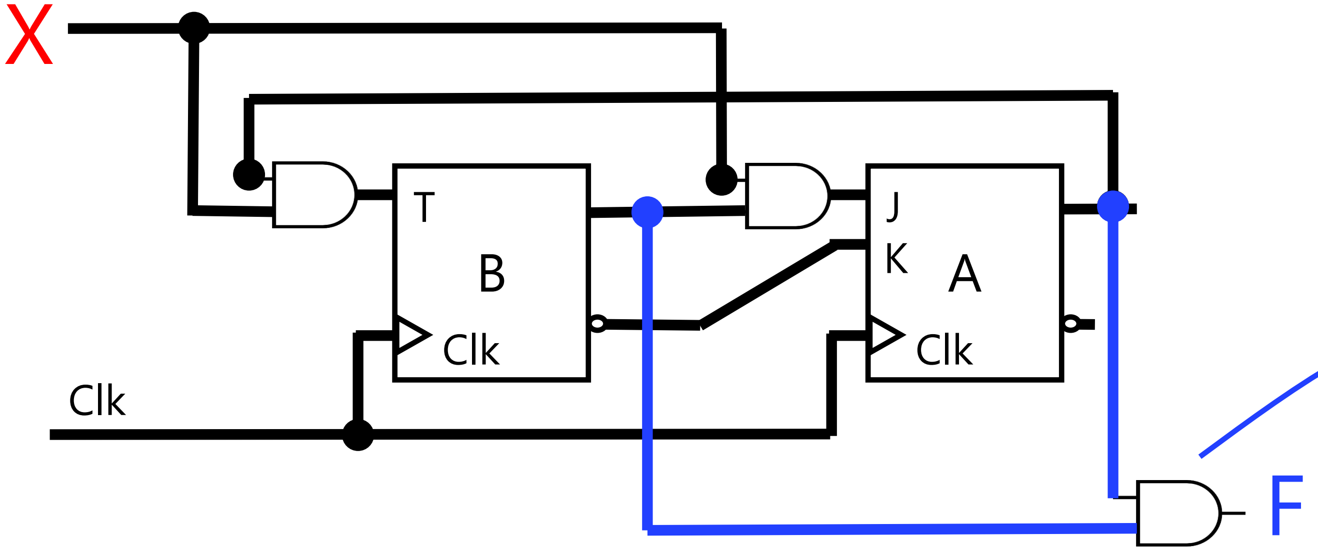
Q(T)		Q(T+1) when X=0		Q(T+1) when X=1		Outputs
B	A	B	A	B	A	F=AB

Moore model:
does not depend on X

Analysis (Recap)

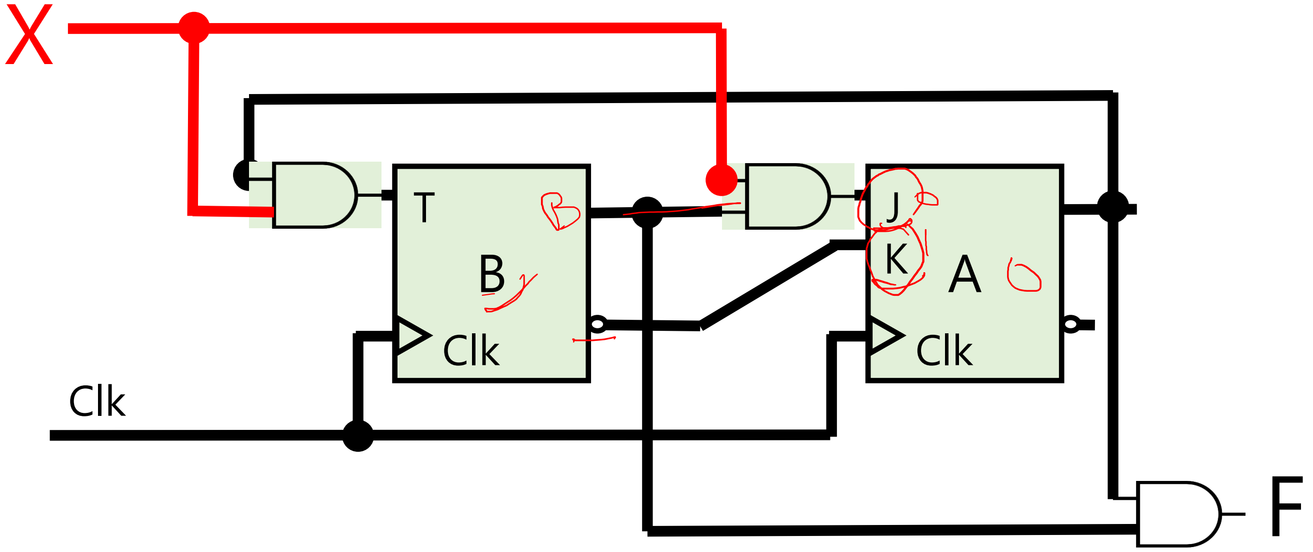
0. Is the circuit sequential or combinational? Sequential
1. What are the flip-flops? T, JK
2. What are the state combinations? $2^{\#FF} \times 2^{\#inputs} = 2^{\#FF+\#inputs} = 2^3 = 8$
3. Form “State” table:
    a) Columns:
      - For each FF, two columns: one for current state, one for next state
      - For each input, one column
      - For each output, one column

    b) Rows: See item 2
4. Fill the state table for
    a) next state columns
    b) the output value

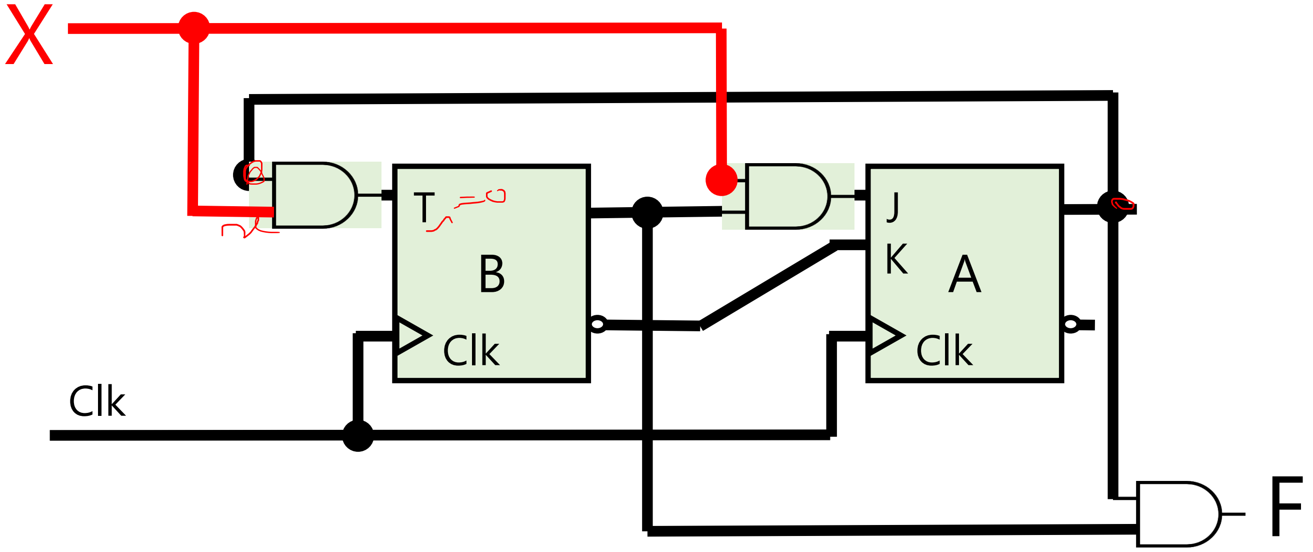


Inputs	Q(T)		Q(T+1)		Outputs
X	B	A	B	A	F=BA
0	0	0			0
0	0	1			0
0	1	0			0
0	1	1			1
1	0	0			0
1	0	1			0
1	1	0			0
1	1	1			1

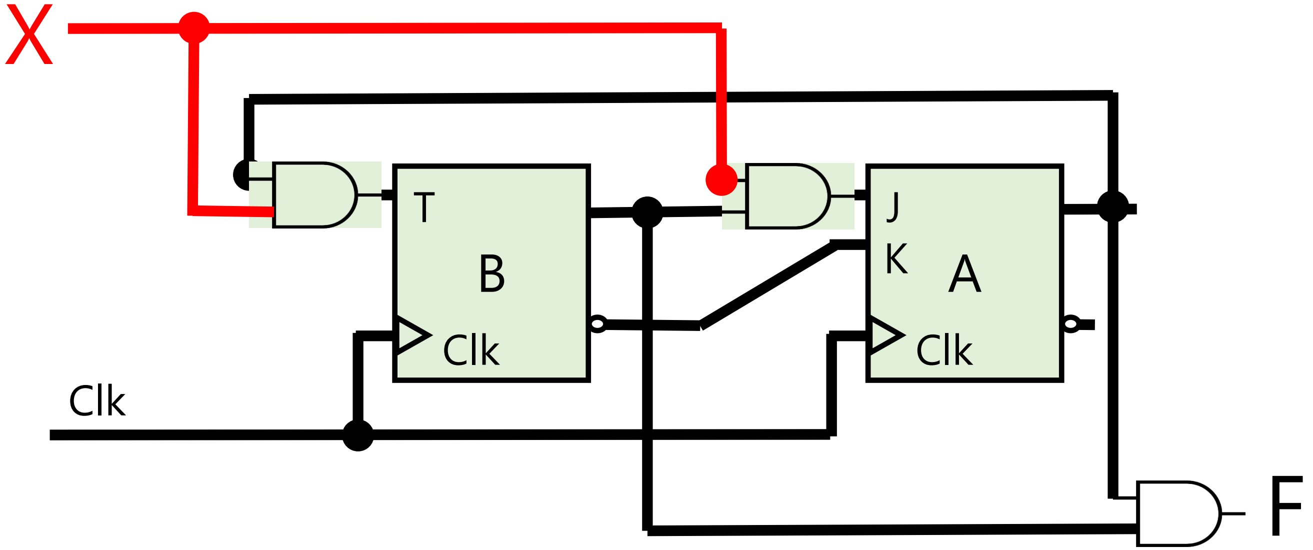
Moore Model
Only depends on current state
X is not involved!



Inputs	Q(T)		Q(T+1)		Outputs
X	B	A	B	A	F=BA
0	0	0		A=0 $J_A=XB=00=0$ $K_A=B'=0'=1$ -------------- Reset → 0	0
0	0	1			0
0	1	0			0
0	1	1			1
1	0	0			0



Inputs	Q(T)		Q(T+1)		Outputs
X	B	A	B	A	F=BA
0	0	0	B=0 T_B=XA=00=0 -------------- Store → 0	0	0
0	0	1			0
0	1	0			0
0	1	1			1
1	0	0			0
1	0	1			0

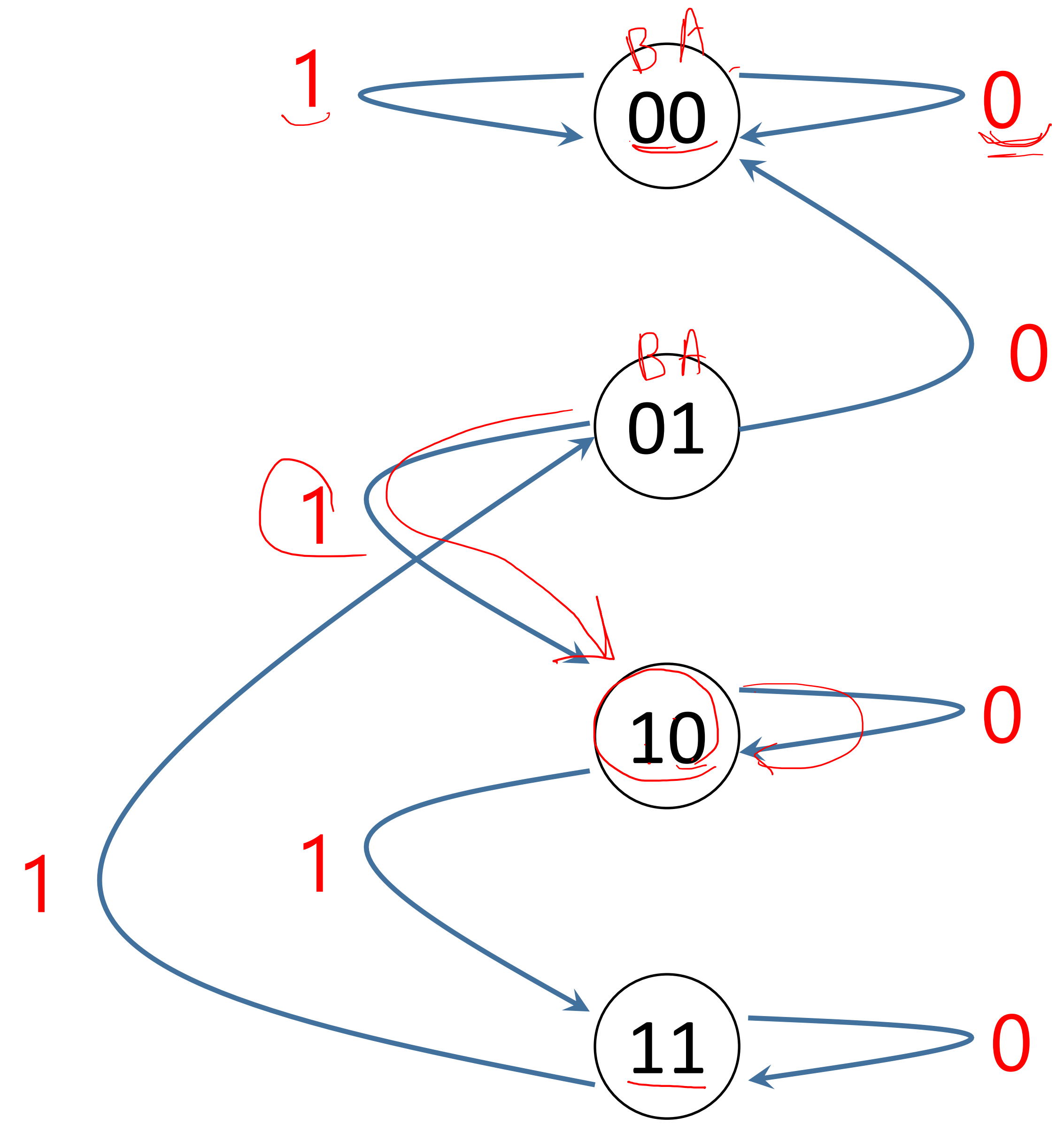


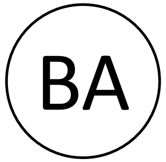
Inputs	Q(T)		Q(T+1)		Outputs
X	B	A	B	A	F=BA
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	1	1	0
1	1	1	0	1	1

Analysis (Recap)

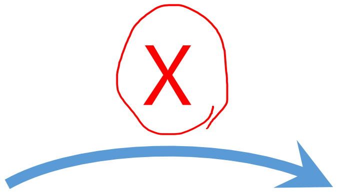
0. Is the circuit sequential or combinational? Sequential
1. What are the flip-flops? T, JK
2. What are the state combinations? $2^{\#FF} \times 2^{\#inputs} = 2^{\#FF+\#inputs} = 2^3 = 8$
3. Form "State" table:
   a) Columns:
      - For each FF, two columns: one for current state, one for next state
      - For each input, one column
      - For each output, one column

   b) Rows: See item 2
4. Fill the state table for
   a) next state columns
   b) the output value
5. Form state (transition) diagram
   a) nodes for states, directed edges for transitions between states
   b) labels for edges by the value of input
   c) labels for nodes by the value of output

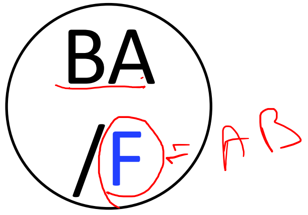


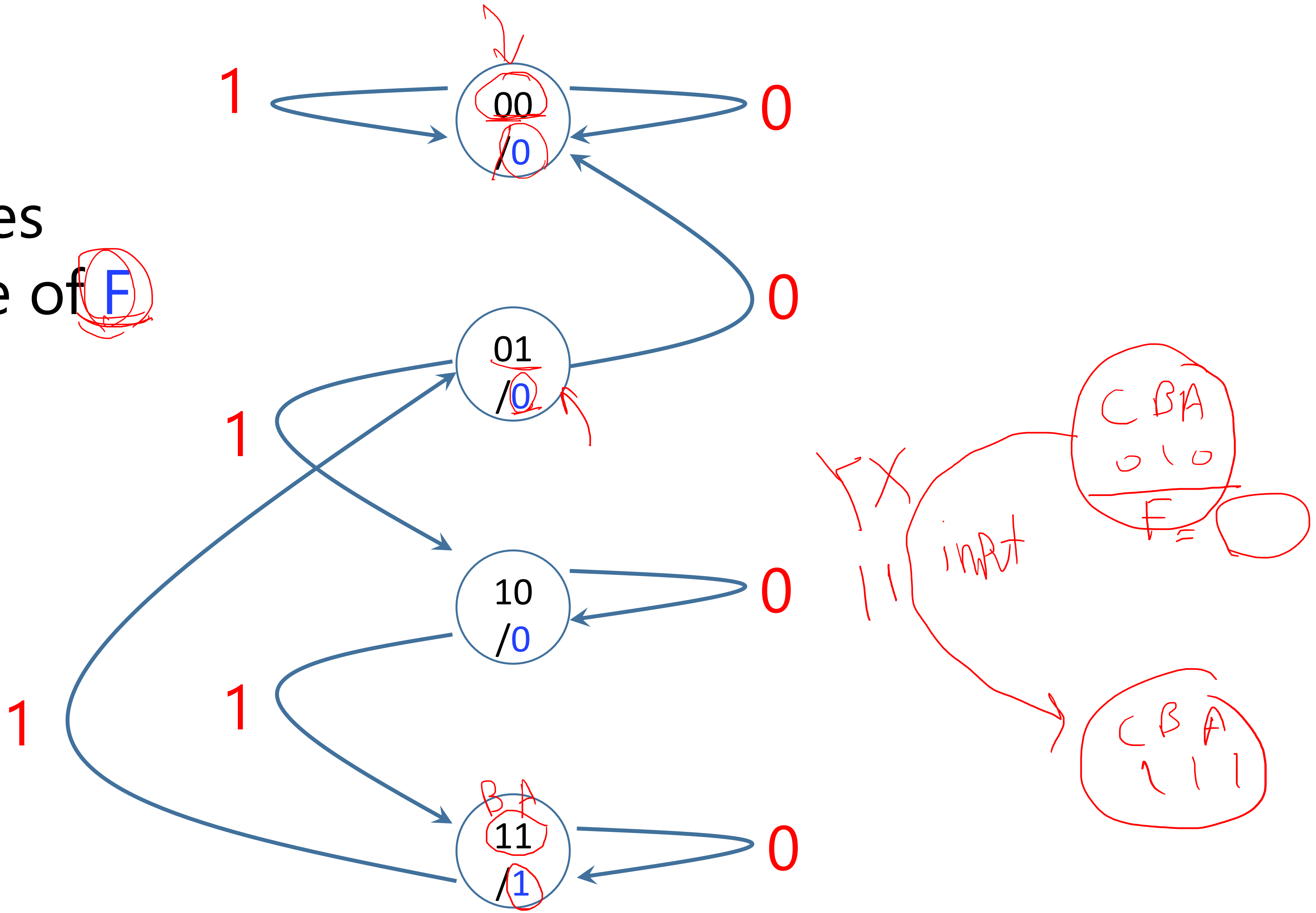


Labels on edges based on value of X



Labels on nodes based on value of F

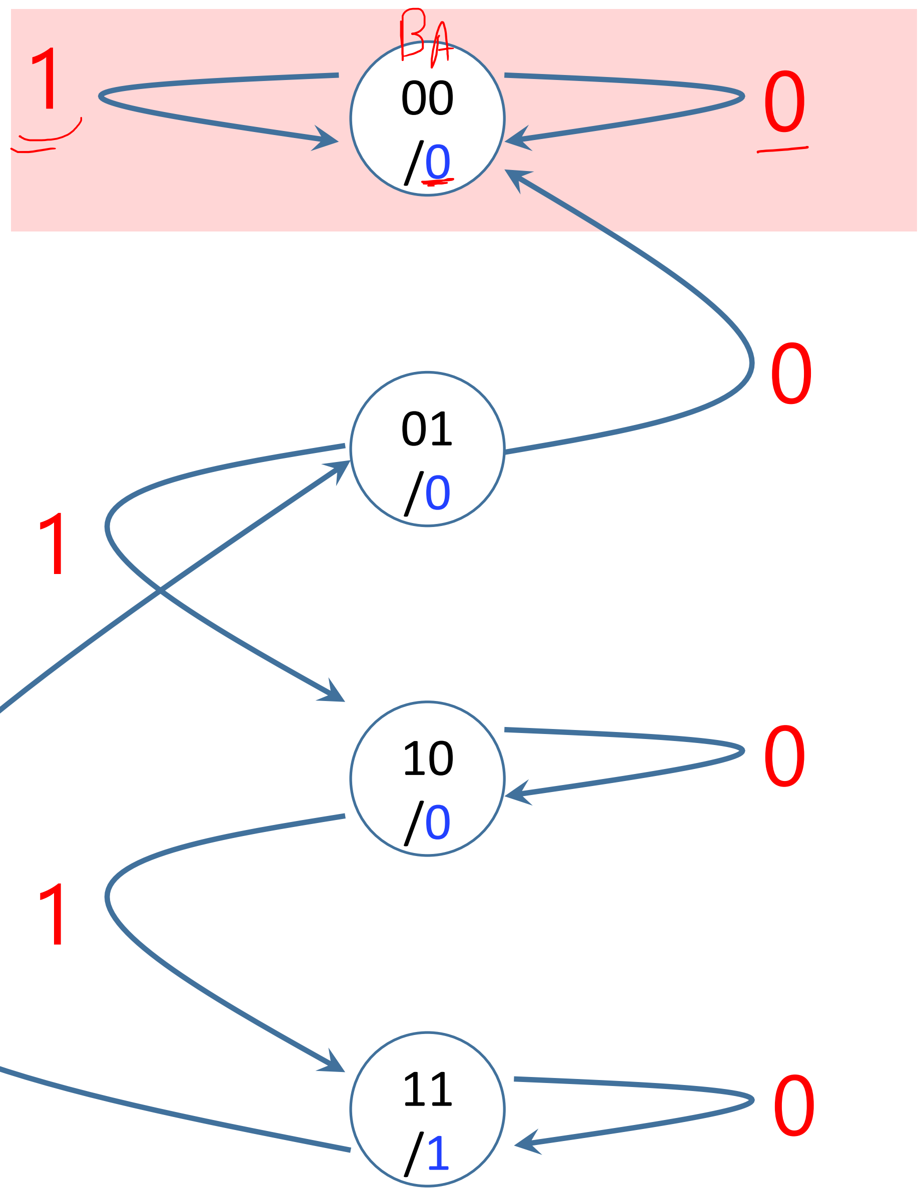




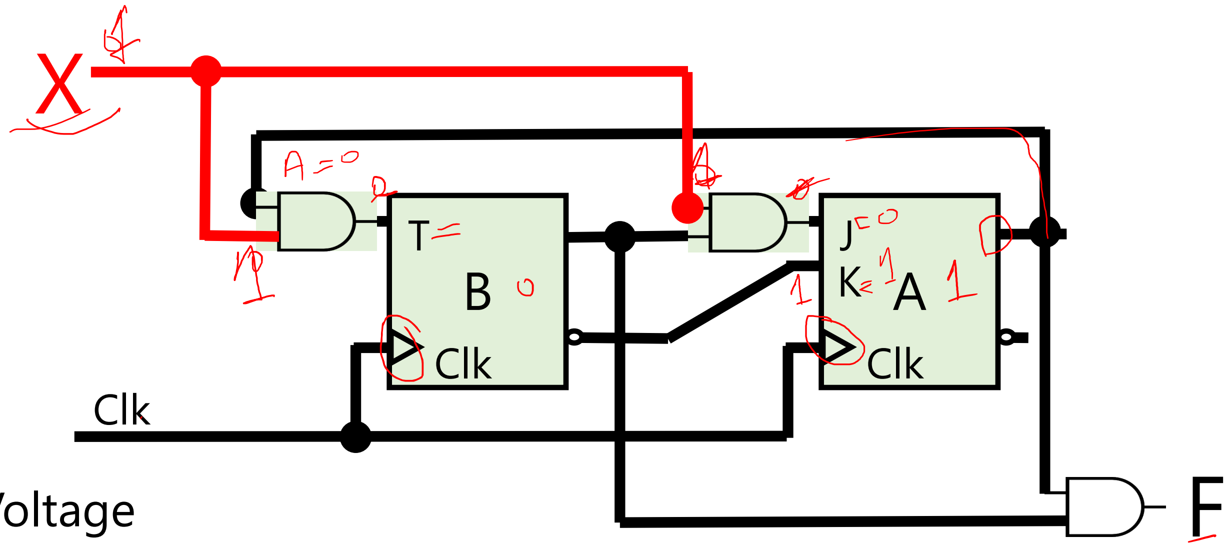
Analysis

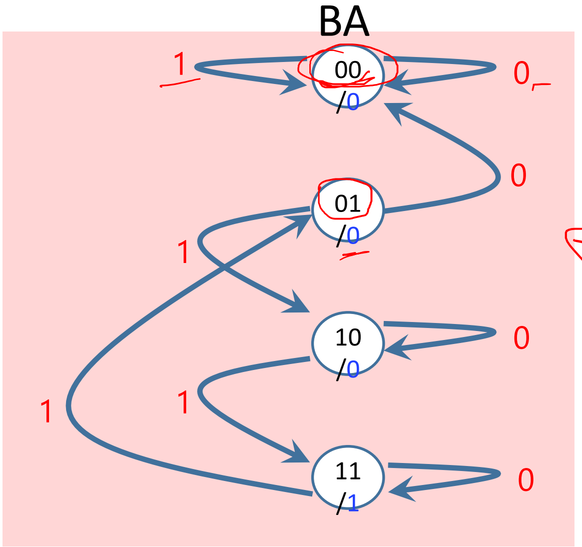
6) (*Optional*) Path on State Transitions

Life lock!

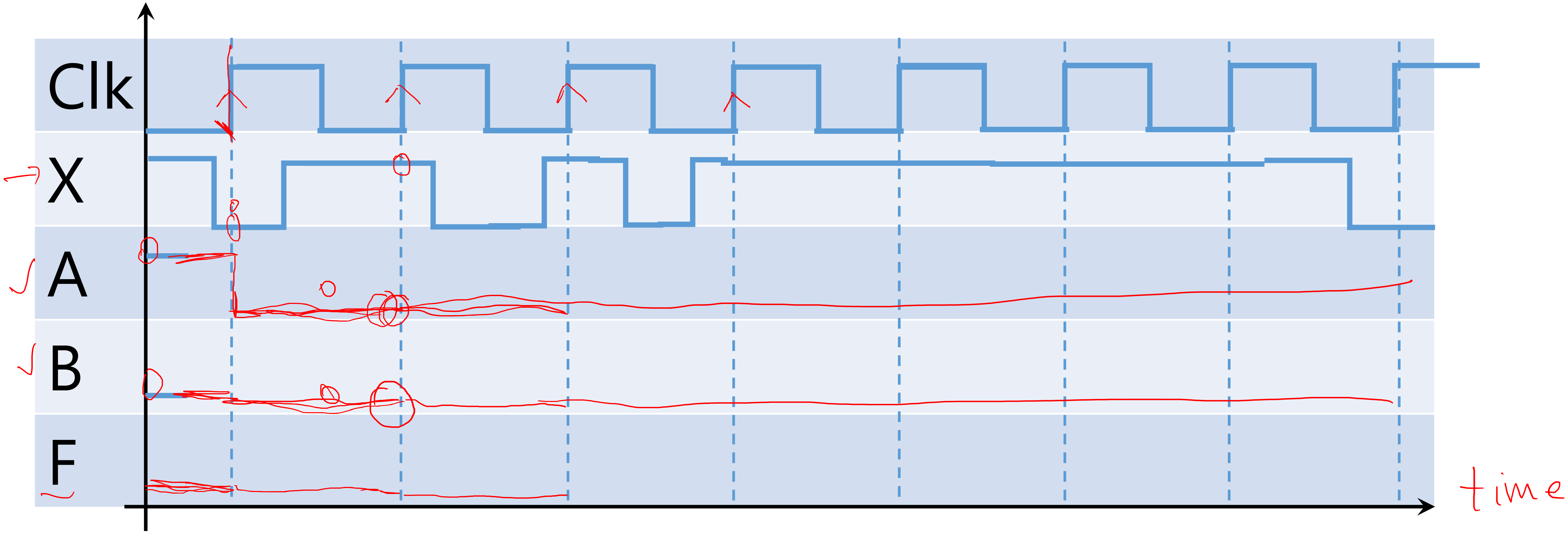


Synchronization

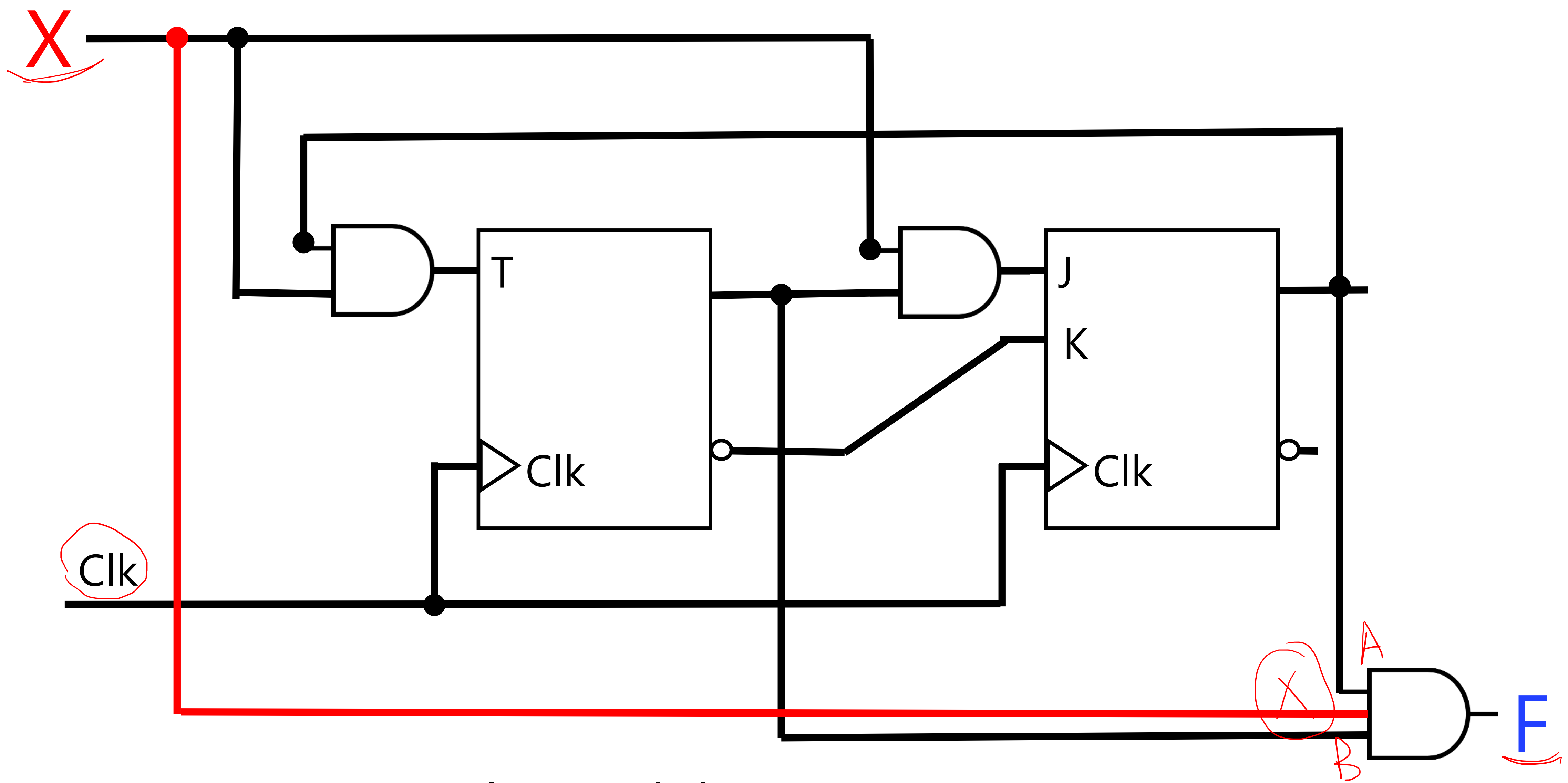

X
A=0
T=
B
Clk
J=0
K=1
A
Clk
Clk
F


BA
00
/0
01
/0
10
/0
11
/1
1
0
0
1
0
1
1
0

State Table

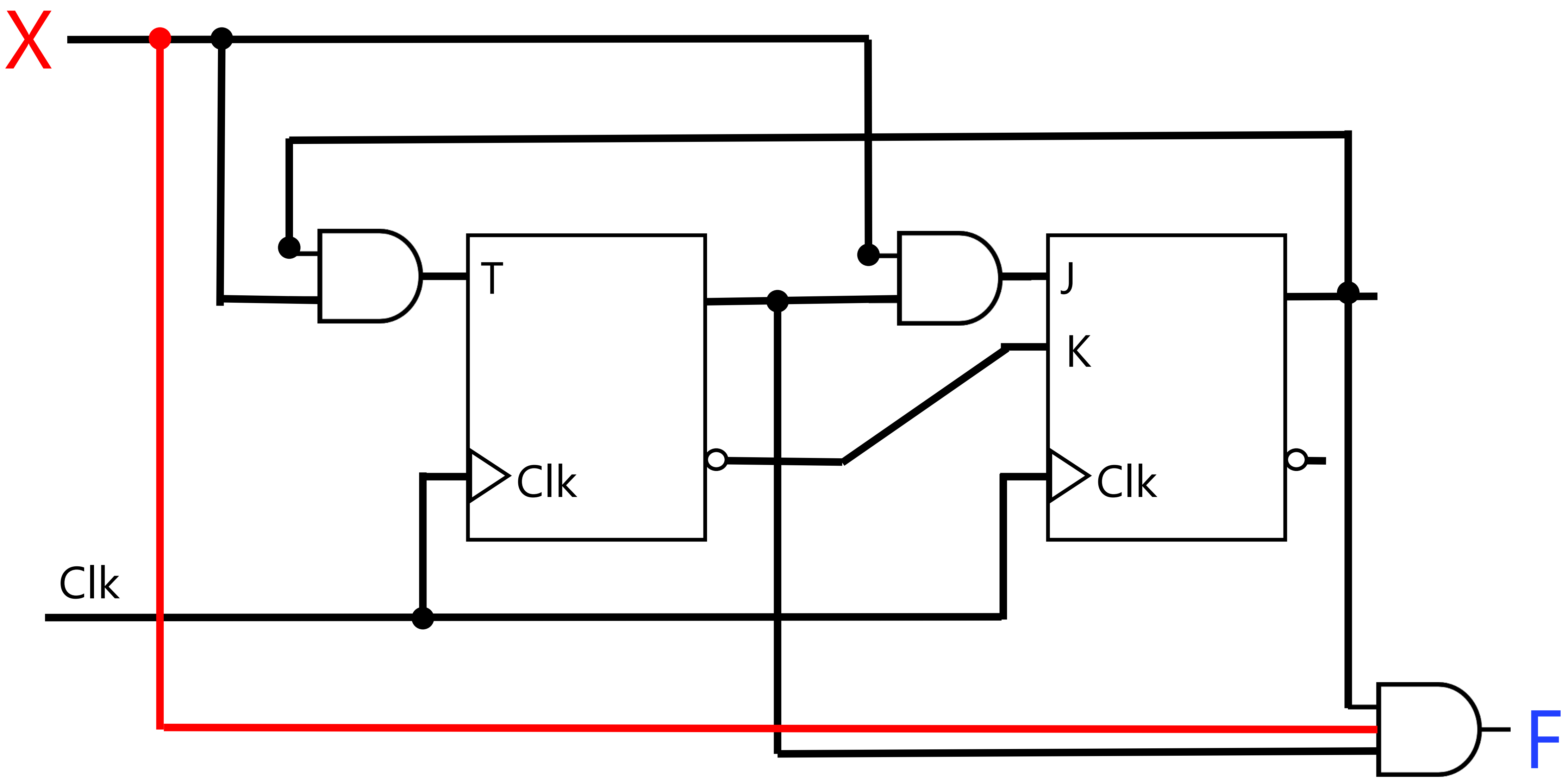

Voltage
Clk
X
A
B
F
time

Asynchronized



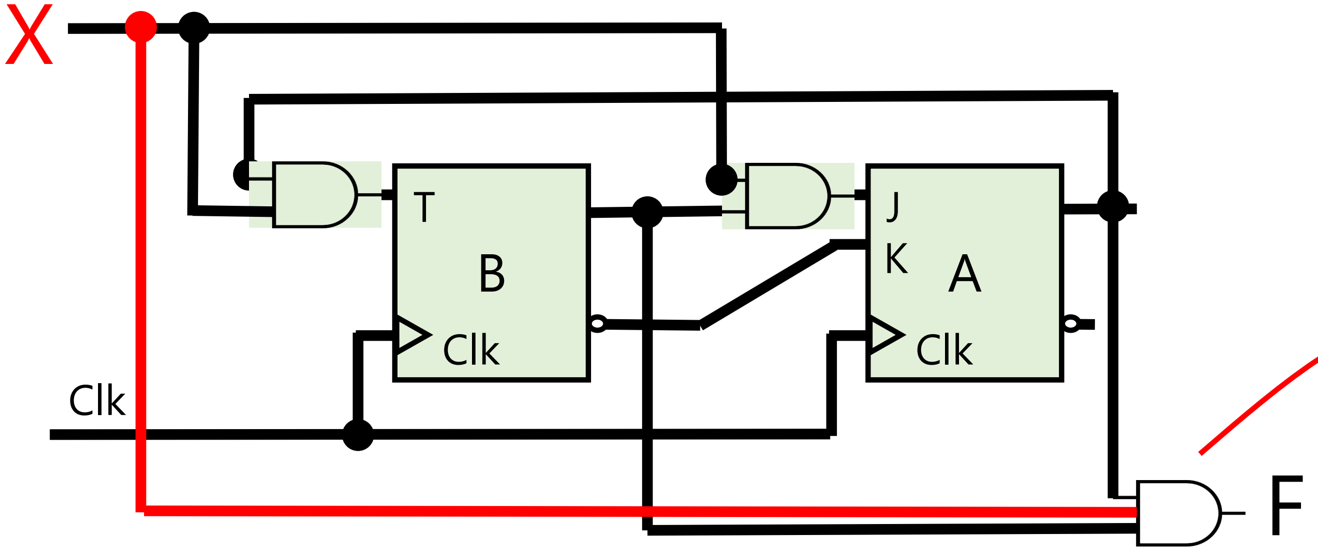
Mealy Model

F depends on X



Mealy Model

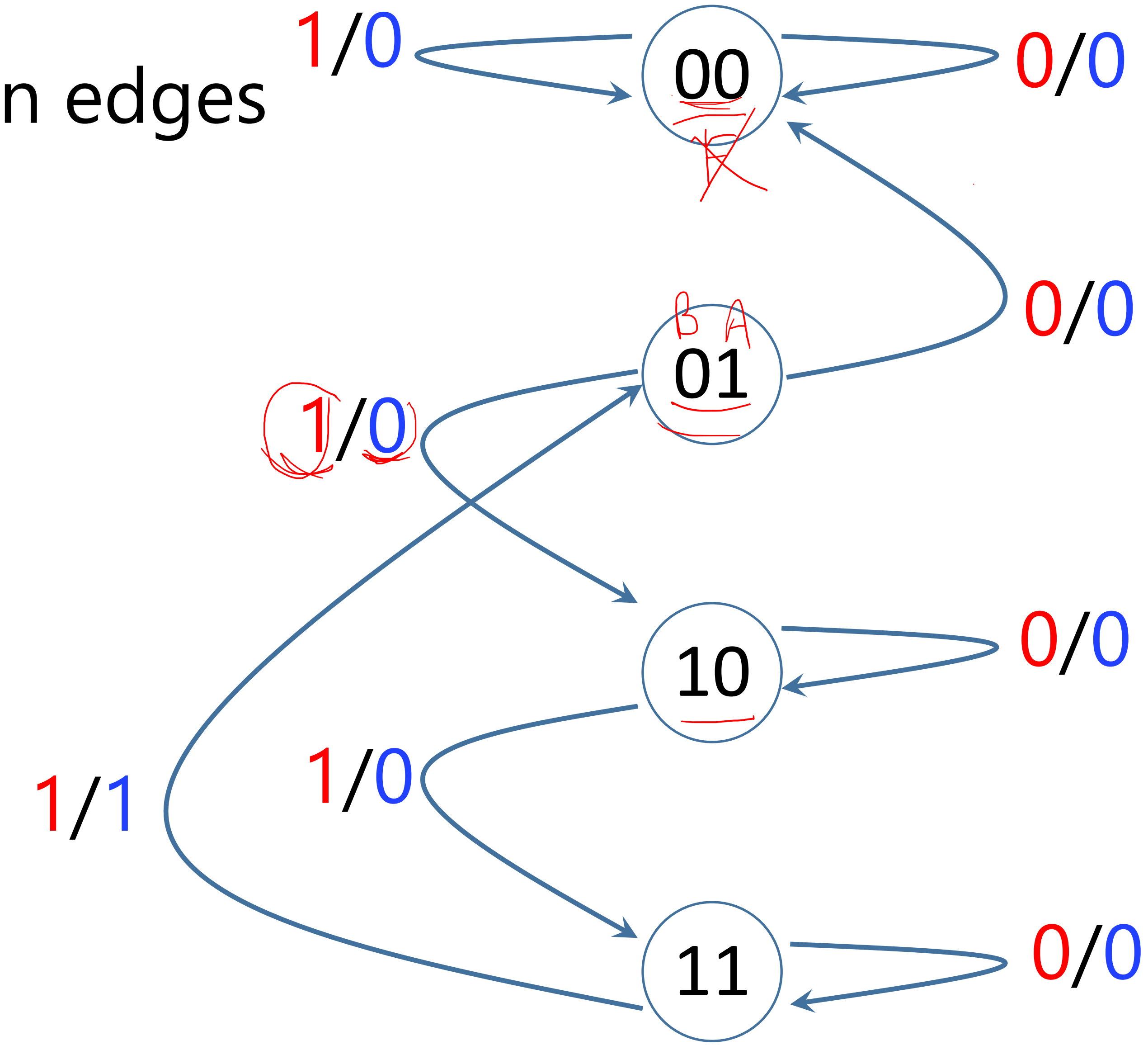
F depends on X, so, F can change out of Clk synchronization!

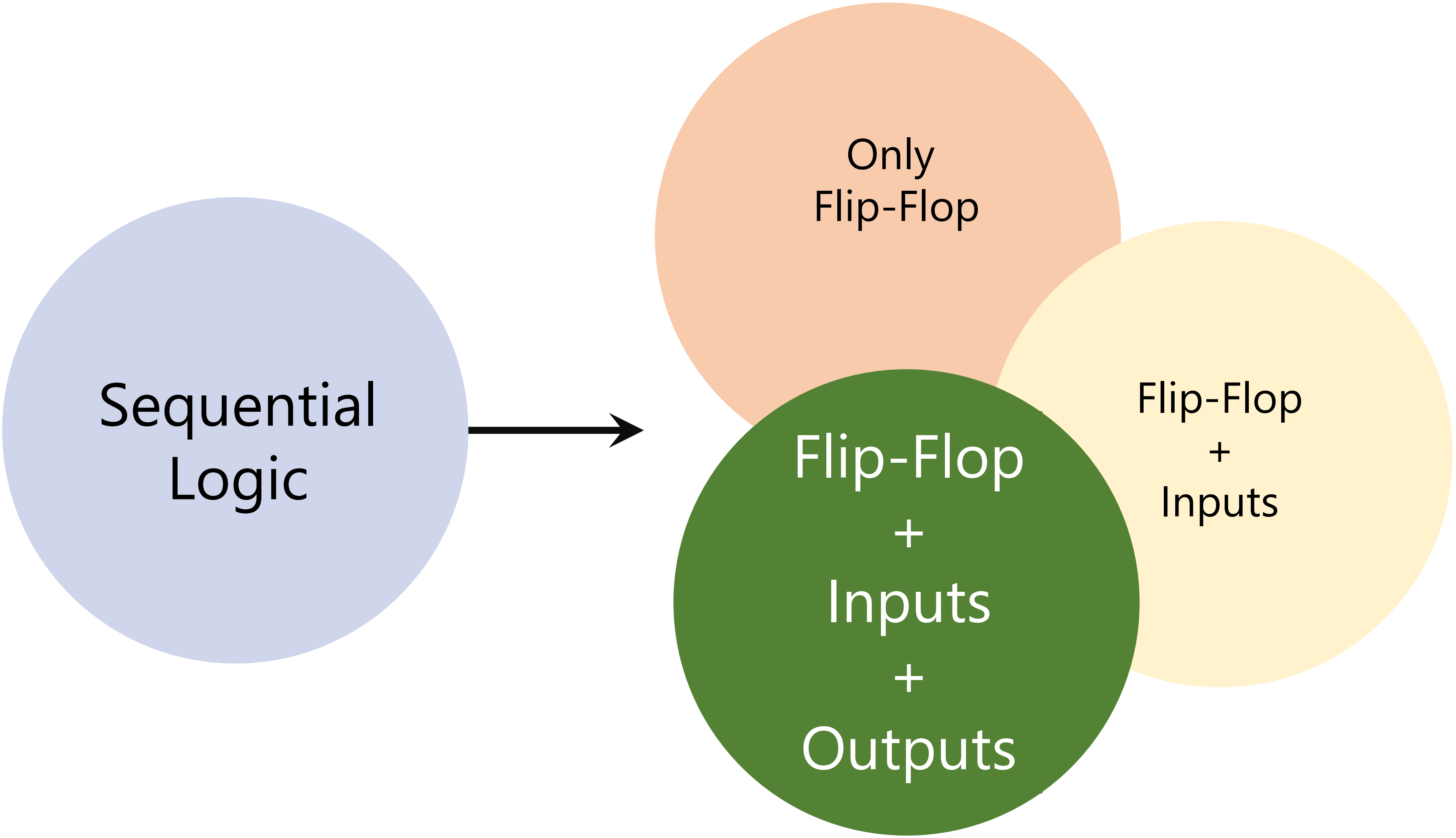


Inputs	Q(T)		Q(T+1)		Outputs
X	B	A	B	A	F=XBA
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	1	1	0
1	1	1	0	1	1

Labels go on edges for F:

X/F




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

Design
by an example

$0/1 \rightarrow 1/0 \rightarrow 2/1$

$1/0 \leftarrow$

$\rightarrow 2/1 \rightarrow 3/0 \rightarrow 4/1$

Counter Up-Down

$2 \leftarrow 3 \leftarrow$

Switch to count up from i to i+1

Switch to count down from i to i-1

Turn on the light if the current number is even

Design (Recap)

0. Do we need combinational logic or sequential logic? Do we need memory?

Counter up/down

0 → 1 → 2 → 3 → … → N-1 → N

N → N-1 → … → 3 → 2 → 1 → 0

Design (Recap)

0. Do we need combinational logic or sequential logic? Do we need memory?
1. How many storage (flip-flops)? #FF

Counter up/down

N=7

000→001→010→011→100→101→110→111

000←001←010←011←100←101←110←111

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

Design (Recap)

0. Do we need combinational logic or sequential logic? Do we need memory?
1. How many storage (flip-flops)? #FF
2. How many input and output?

Counter up/down

N=7

000→001→010→011→100→101→110→111

000←001←010←011←100←101←110←111

We need to switch between up and down → 1 binary variable

X=0 Count Up [illegible]

X=1 Count Down

Counter up/down
N=7

X=0: 000→001→010→011→100→101→110→111

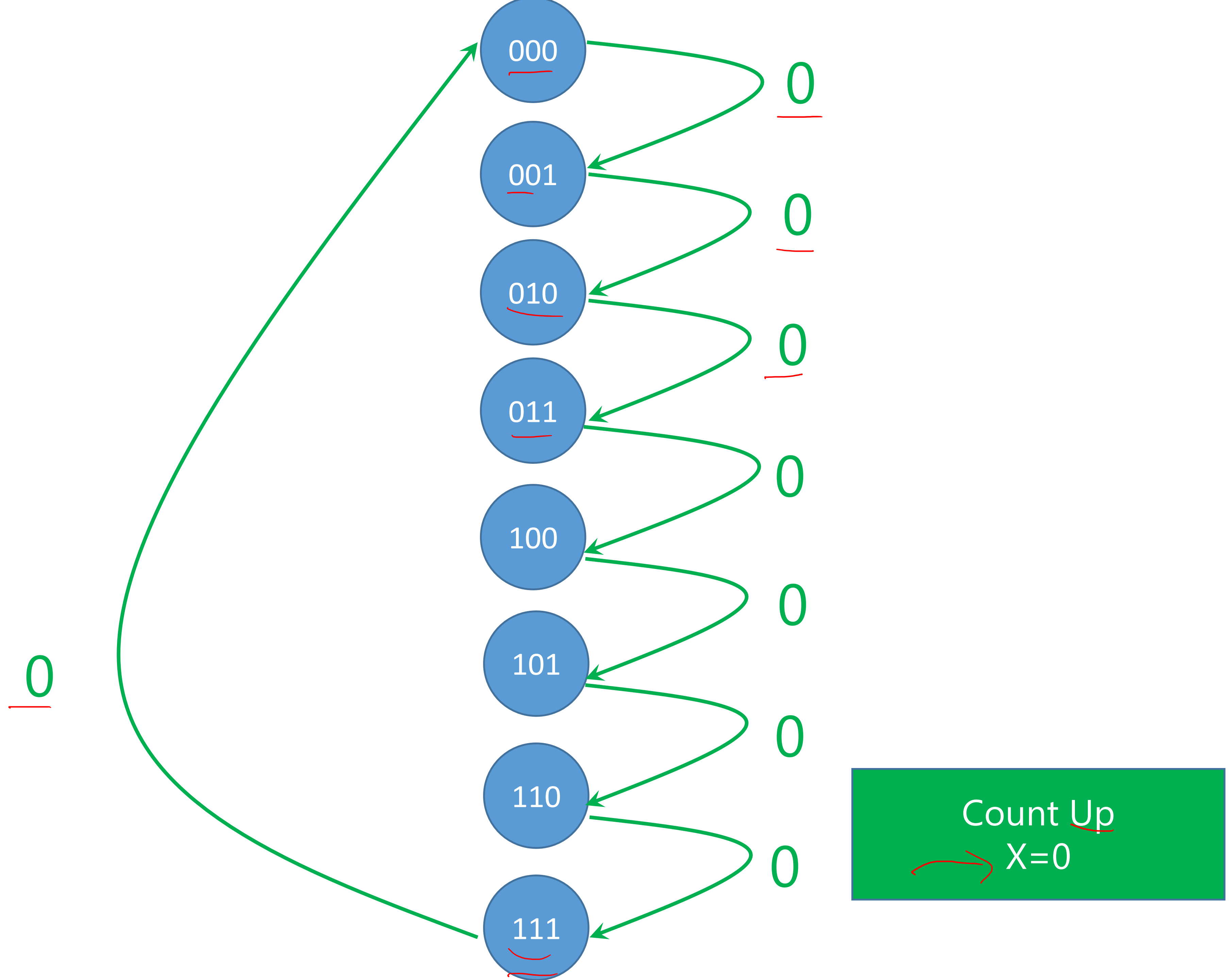
X=1: 111→110→101→100→011→010→001→000

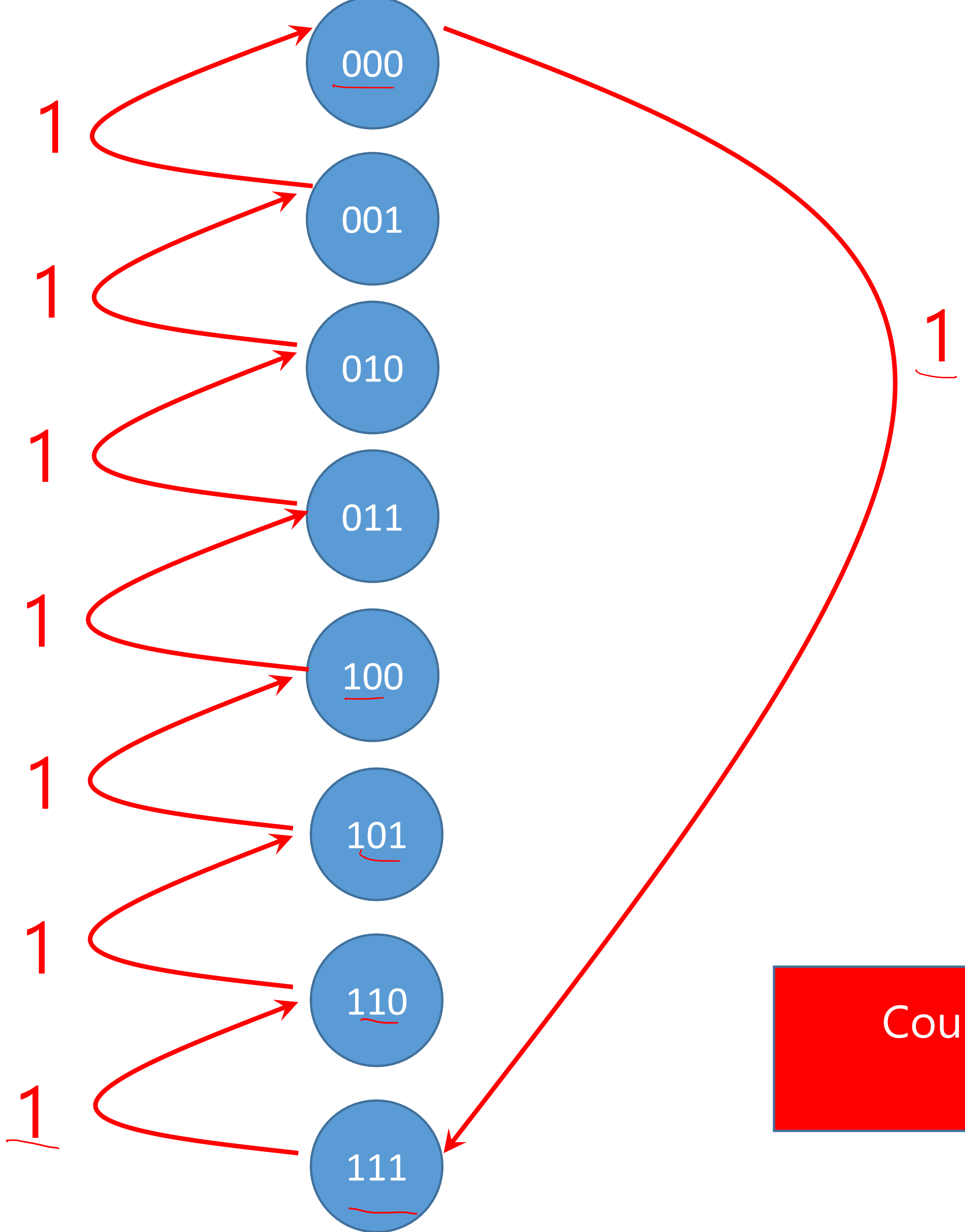
We need to turn on the light when the number is even

→ 1 binary variable → F =0

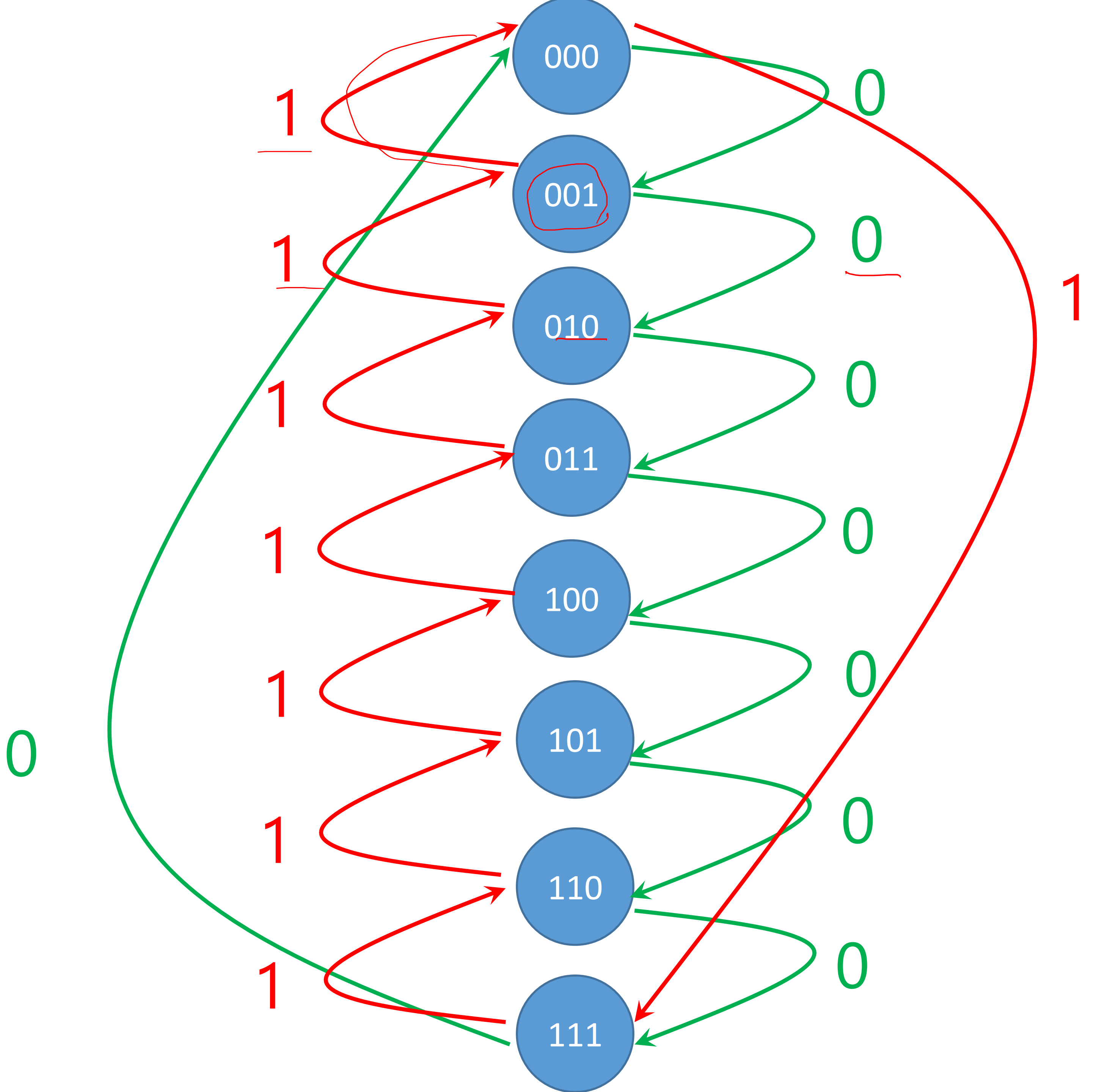
Design (Recap)

0. Do we need combinational logic or sequential logic? Do we need memory?
1. How many storage (flip-flops)? #FF
2. How many input and output?
3. Form the state (transition) diagram


000
001
010
011
100
101
110
111
0
0
0
0
0
0
0
0
Count Up
X=0



Count Down
X=1

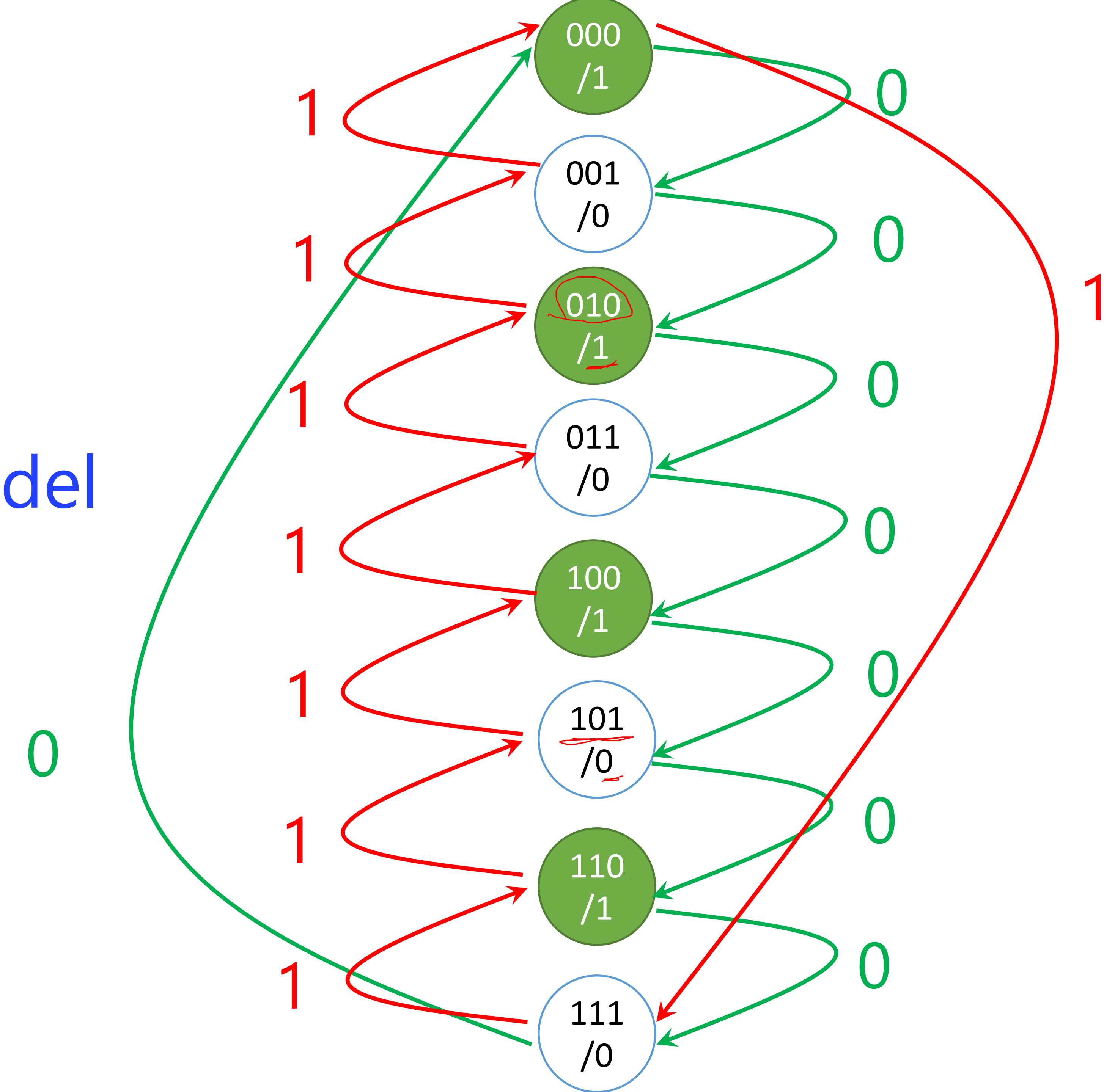


000
001
010
011
100
101
110
111
1
1
1
1
1
1
1
1
0
0
0
0
0
0
0
0

F=1 : even

F=0 : else

Moore model



Design (Recap)

0. Do we need combinational logic or sequential logic? Do we need memory?
1. How many storage (flip-flops)? #FF
2. How many input and output?
3. Form the state (transition) diagram
4. Form the state table
5. Fill the state table

Inputs	Q(T)			Q(T+1)			Outputs
X	C	B	A	C	B	A	F
0	0	0	0	0	0	1	
0	0	0	1	0	1	0	
0	0	1	0	0	1	1	
0	0	1	1	1	0	0	
0	1	0	0	1	0	1	
0	1	0	1	1	1	0	
0	1	1	0	1	1	1	
0	1	1	1	0	0	0	
1	0	0	0	1	1	1	
1	0	0	1	0	0	0	
1	0	1	0	0	0	1	
1	0	1	1	0	1	0	
1	1	0	0	0	1	1	
1	1	0	1	1	0	0	
1	1	1	0	1	0	1	
1	1	1	1	1	1	0	

Count Up
X=0

Count Down
X=1

Inputs	Q(T)			Q(T+1)			Outputs
X	C	B	A	C	B	A	F
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	1	1
0	0	1	1	1	0	0	0
0	1	0	0	1	0	1	1
0	1	0	1	1	1	0	0
0	1	1	0	1	1	1	1
0	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1
1	0	0	1	0	0	0	0
1	0	1	0	0	0	1	1
1	0	1	1	0	1	0	0
1	1	0	0	0	1	1	1
1	1	0	1	1	0	0	0
1	1	1	0	1	0	1	1
1	1	1	1	1	1	0	0

F depends only on current state of CBA

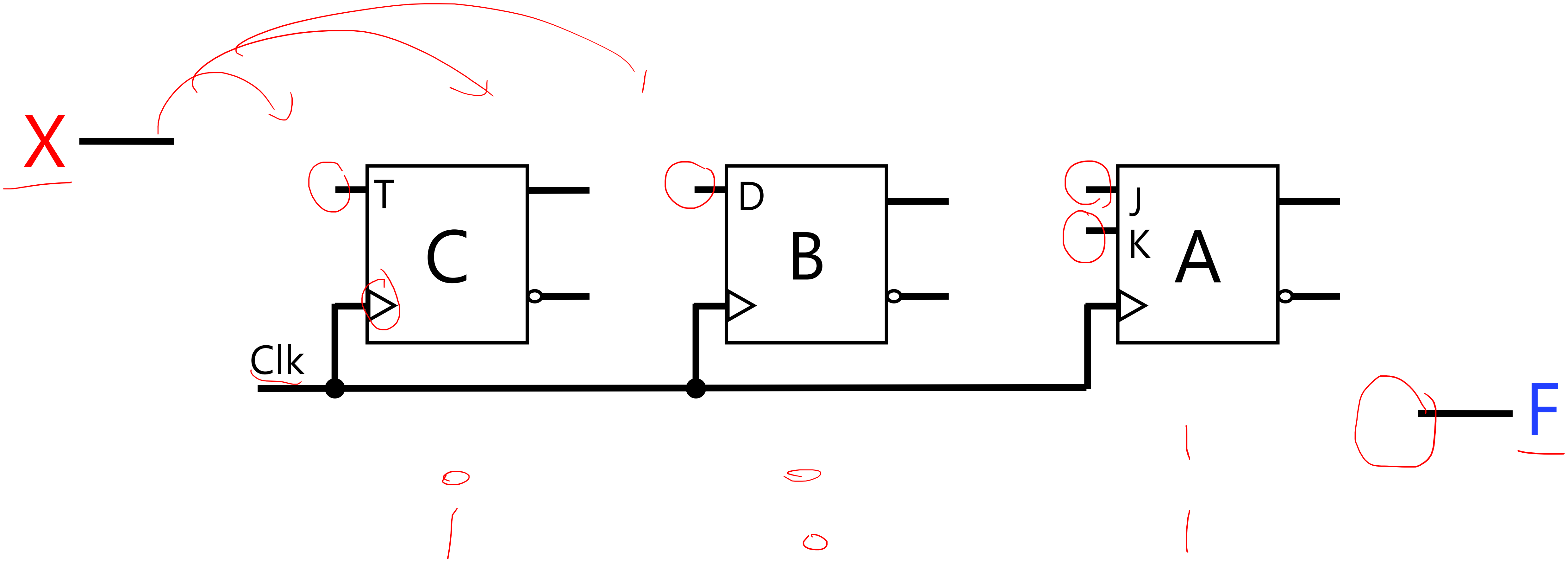
Inputs	Q(T)			Q(T+1)			Outputs
X	C	B	A	C	B	A	F
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	1	1
0	0	1	1	1	0	0	0
0	1	0	0	1	0	1	1
0	1	0	1	1	1	0	0
0	1	1	0	1	1	1	1
0	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1
1	0	0	1	0	0	0	0
1	0	1	0	0	0	1	1
1	0	1	1	0	1	0	0
1	1	0	0	0	1	1	1
1	1	0	1	1	0	0	0
1	1	1	0	1	0	1	1
1	1	1	1	1	1	0	0

Design (Recap)

0. Do we need combinational logic or sequential logic? Do we need memory?
1. How many storage (flip-flops)? #FF
2. How many input and output?
3. Form the state (transition) diagram
4. Form the state table
5. Fill the state table
6. What type of storage (flip-flop)? RS, D, T, JK, or Mixed

Counter Up/Down

Let’s pick mix FFs: 1×T-FF, 1×D-FF, 1×JK-FF


X
T
C
D
B
J
K
A
Clk
F

Design (Recap)

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

Excitation Equations		
A	$J_A=F(X, C, B, A)$	$K_A=F(X, C, B, A)$
B	$D_B=F(X, C, B, A)$	
C	$T_C=F(X, C, B, A)$	

Input

Flip-Flops

Excitation Equations		
A	J_A=F(X, C, B, A)	K_A=F(X, C, B, A)
B	D_B=	
C	T_C=	

Inputs	Q(T)			Q(T+1)			Outputs	Excitation for A		
X	C	B	A	C	B	A	F	Action	J_A	K_A
0	0	0	0	0	0	1	1	Set/Cmp m_0	1	X
0	0	0	1	0	1	0	0	Rst/Cmp	X	1
0	0	1	0	0	1	1	1	Set/Cmp	1	X
0	0	1	1	1	0	0	0	Rst/Cmp	X	1
0	1	0	0	1	0	1	1	Set/Cmp	1	X
0	1	0	1	1	1	0	0	Rst/Cmp	X	1
0	1	1	0	1	1	1	1	Set/Cmp	1	X
0	1	1	1	0	0	0	0	Rst/Cmp	X	1
1	0	0	0	1	1	1	1	Set/Cmp	1	X
1	0	0	1	0	0	0	0	Rst/Cmp	X	1
1	0	1	0	0	0	1	1	Set/Cmp	1	X
1	0	1	1	0	1	0	0	Rst/Cmp	X	1
1	1	0	0	0	1	1	1	Set/Cmp	1	X
1	1	0	1	1	0	0	0	Rst/Cmp	X	1
1	1	1	0	1	0	1	1	Set/Cmp	1	X
1	1	1	1	1	1	0	0	Rst/Cmp	X	1

Excitation Equations		
A	$J_A=\sum(0,2,4,6,8,10,12,14)+d(1,3,5,7,9,11,13,15)$	$K_A=\sum(1,3,5,7,9,11,13,15)+d(0,2,4,6,8,10,12,14)$
B	$D_B=$	
C	$T_C=$	

Inputs	Q(T)			Q(T+1)			Outputs	Excitation for B	
X	C	B	A	C	B	A	F	Action	D_B
0	0	0	0	0	0	1	1	Rst	0
0	0	0	1	0	1	0	0	Set	1
0	0	1	0	0	1	1	1	Set	1
0	0	1	1	1	0	0	0	Rst	0
0	1	0	0	1	0	1	1	Rst	0
0	1	0	1	1	1	0	0	Set	1
0	1	1	0	1	1	1	1	Set	1
0	1	1	1	0	0	0	0	Rst	0
1	0	0	0	1	1	1	1	Set	1
1	0	0	1	0	0	0	0	Rst	0
1	0	1	0	0	0	1	1	Rst	0
1	0	1	1	0	1	0	0	Set	1
1	1	0	0	0	1	1	1	Set	1
1	1	0	1	1	0	0	0	Rst	0
1	1	1	0	1	0	1	1	Rst	0
1	1	1	1	1	1	0	0	Set	1

Excitation Equations		
A	$J_A=\sum(0,2,4,6,8,10,12,14)+d(1,3,5,7,9,11,13,15)$	$K_A=\sum(1,3,5,7,9,11,13,15)+d(0,2,4,6,8,10,12,14)$
B	$D_B=\sum(1,2,5,6,8,11,12,15)$ *Never "don't care condition" happens in D-FF*	
C	$T_C=$	

Inputs	Q(T)			Q(T+1)			Outputs	Excitation for C	
X	C	B	A	C	B	A	F	Action	T_C
0	0	0	0	0	0	1	1	Store	0
0	0	0	1	0	1	0	0	Store	0
0	0	1	0	0	1	1	1	Store	0
0	0	1	1	1	0	0	0	Comp	1
0	1	0	0	1	0	1	1	Store	0
0	1	0	1	1	1	0	0	Store	0
0	1	1	0	1	1	1	1	Store	0
0	1	1	1	0	0	0	0	Comp	1
1	0	0	0	1	1	1	1	Comp	1
1	0	0	1	0	0	0	0	Store	0
1	0	1	0	0	0	1	1	Store	0
1	0	1	1	0	1	0	0	Store	0
1	1	0	0	0	1	1	1	Comp	1
1	1	0	1	1	0	0	0	Store	0
1	1	1	0	1	0	1	1	Store	0
1	1	1	1	1	1	0	0	Store	0

Excitation Equations		
A	$J_A=\sum(0,2,4,6,8,10,12,14)+d(1,3,5,7,9,11,13,15)$	$K_A=\sum(1,3,5,7,9,11,13,15)+d(0,2,4,6,8,10,12,14)$
B	$D_B=\sum(1,2,5,6,8,11,12,15)$ *Never "don't care condition" happens in D-FF*	
C	$T_C=\sum(3,7,8,12)$ *Never "don't care condition" happens in T-FF*	

Minimization for excitation equations

?-Variable K-Map

Excitation Equations		
A	$J_A=\sum(0,2,4,6,8,10,12,14)+d(1,3,5,7,9,11,13,15)$	$K_A=\sum(1,3,5,7,9,11,13,15)+d(0,2,4,6,8,10,12,14)$
B	$D_B=\sum(1,2,5,6,8,11,12,15)$ *Never "don't care condition" happens in D-FF*	
C	$T_C=\sum(3,7,8,12)$ *Never "don't care condition" happens in T-FF*	

Minimization

4-Variable K-Map

F(X, C, B, A)

XC \ BA	00	01	11	10
00	0 m_0	0 m_1	1 m_3	0 m_2
01	0 m_4	0 m_5	1 m_7	0 m_6
11	1 m_{12}	0 m_{13}	0 m_{15}	0 m_{14}
10	1 m_8	0 m_9	0 m_{11}	0 m_{10}

$T_C = \sum(3,7,8,12)$
$= X'BA + XB'A'$

XC \ BA	00	01	11	10
00	0 m_0	1 m_1	0 m_3	1 m_2
01	0 m_4	1 m_5	0 m_7	1 m_6
11	1 m_{12}	0 m_{13}	1 m_{15}	0 m_{14}
10	1 m_8	0 m_9	1 m_{11}	0 m_{10}

$D_B = \sum(1,2,5,6,8,11,12,15)$
$= X'B'A + X'BA' + XB'A' + XBA$
$= X'(B'A + BA') + X(B'A' + BA)$
$= X'(B \oplus A) + X(B \odot A)$
$= X \oplus B \oplus A$

XC \ BA	00	01	11	10
00	1 m_0	X m_1	X m_3	1 m_2
01	1 m_4	X m_5	X m_7	1 m_6
11	1 m_{12}	X m_{13}	X m_{15}	1 m_{14}
10	1 m_8	X m_9	X m_{11}	1 m_{10}

$J_A = \sum(0,2,4,6,8,10,12,14) +$
$d(1,3,5,7,9,11,13,15) = 1$

XC \ BA	00	01	11	10
00	X m_0	1 m_1	1 m_3	X m_2
01	X m_4	1 m_5	1 m_7	X m_6
11	X m_{12}	1 m_{13}	1 m_{15}	X m_{14}
10	X m_8	1 m_9	1 m_{11}	X m_{10}

$K_A = \sum(1,3,5,7,9,11,13,15) +$
$d(0,2,4,6,8,10,12,14) = 1$

Excitation Equations		
A	$J_A=1$	$K_A=1$
B	$D_B=F(X,C,B,A)=X\oplus B\oplus A$	
C	$T_C=F(X,C,B,A)=X'BA+XB'A'$	

Design (Recap)

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

Inputs	Q(T)			Q(T+1)			Outputs
X	C	B	A	C	B	A	F = SoP
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	1	1
0	0	1	1	1	0	0	0
0	1	0	0	1	0	1	1
0	1	0	1	1	1	0	0
0	1	1	0	1	1	1	1
0	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1
1	0	0	1	0	0	0	0
1	0	1	0	0	0	1	1
1	0	1	1	0	1	0	0
1	1	0	0	0	1	1	1
1	1	0	1	1	0	0	0
1	1	1	0	1	0	1	1
1	1	1	1	1	1	0	0

F(C, B, A)

Moore

Inputs	Q(T)			Q(T+1)			Outputs
X	C	B	A	C	B	A	Y
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	1	1
0	0	1	1	1	0	0	0
0	1	0	0	1	0	1	1
0	1	0	1	1	1	0	0
0	1	1	0	1	1	1	1
0	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1
1	0	0	1	0	0	0	0
1	0	1	0	0	0	1	1
1	0	1	1	0	1	0	0
1	1	0	0	0	1	1	1
1	1	0	1	1	0	0	0
1	1	1	0	1	0	1	1
1	1	1	1	1	1	0	0

$$F = F(C,B,A) = \sum(0,2,4,6)$$

Inputs	Q(T)			Q(T+1)			Outputs
X	C	B	A	C	B	A	Y
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	1	1
0	0	1	1	1	0	0	0
0	1	0	0	1	0	1	1
0	1	0	1	1	1	0	0
0	1	1	0	1	1	1	1
0	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1
1	0	0	1	0	0	0	0
1	0	1	0	0	0	1	1
1	0	1	1	0	1	0	0
1	1	0	0	0	1	1	1
1	1	0	1	1	0	0	0
1	1	1	0	1	0	1	1
1	1	1	1	1	1	0	0

~~Incorrect but if we make a mistake that X is involved:~~

~~$F = F(X,C,B,A) = \sum(0,2,4,6,8,10,12,14)$~~

Minimization for output Y

3-Variable K-Map

~~Incorrectly 4-Variable K-Map~~

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

$$F = F(C,B,A) = \sum(0,2,4,6)$$
$$= A'$$

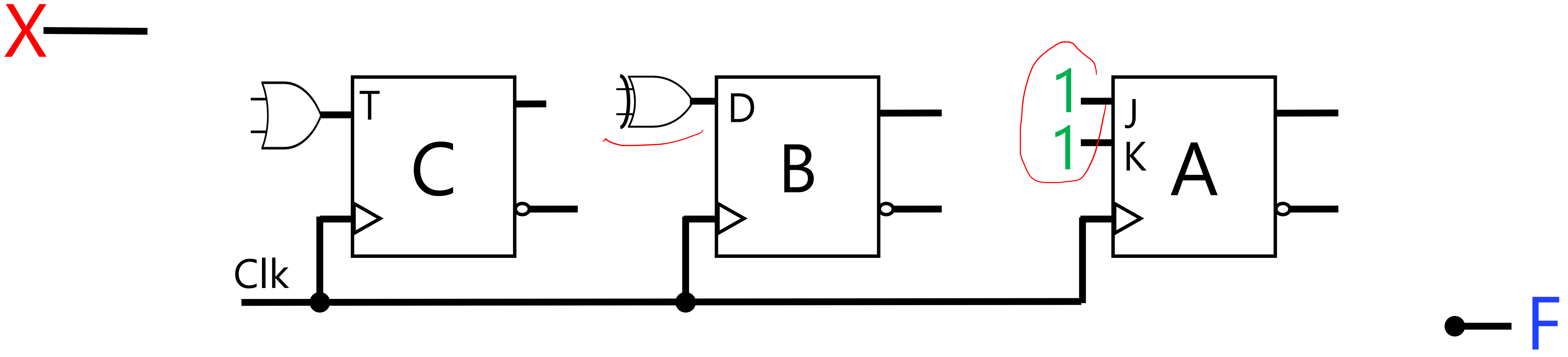
Even if we consider the input, because Y does not depend on X in the state table, it will disappear in simplification!

XC \ BA	00	01	11	10
00	1 (m_0)	0 (m_1)	0 (m_3)	1 (m_2)
01	1 (m_4)	0 (m_5)	0 (m_7)	1 (m_6)
11	1 (m_{12})	0 (m_{13})	0 (m_{15})	1 (m_{14})
10	1 (m_8)	0 (m_9)	0 (m_{11})	1 (m_{10})

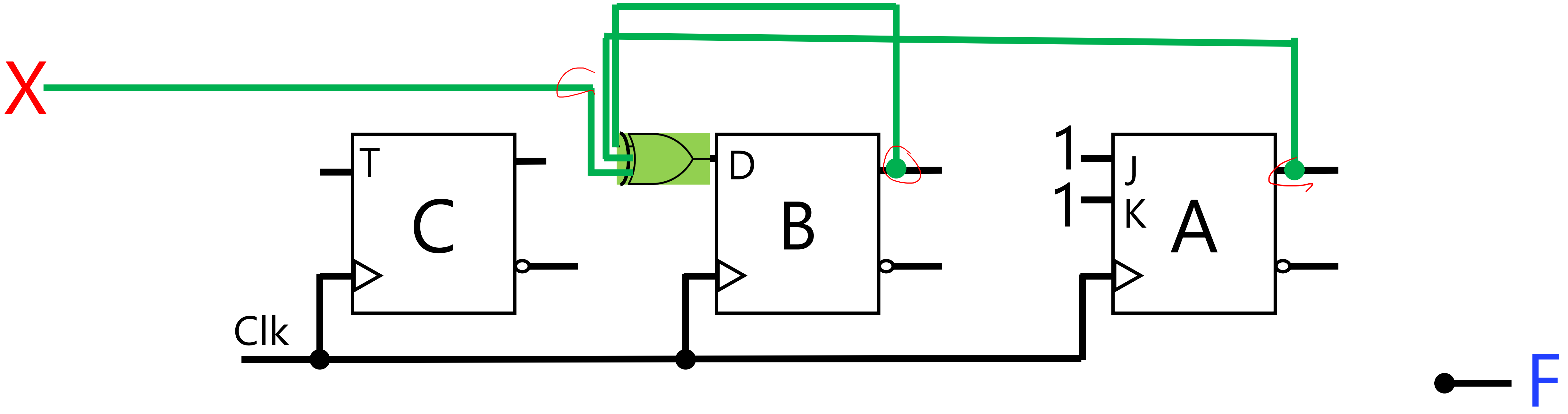
$$F = F(X,C,B,A) = \sum(0,2,4,6,8,10,12,14)$$
$$= A'$$

Design (Recap)

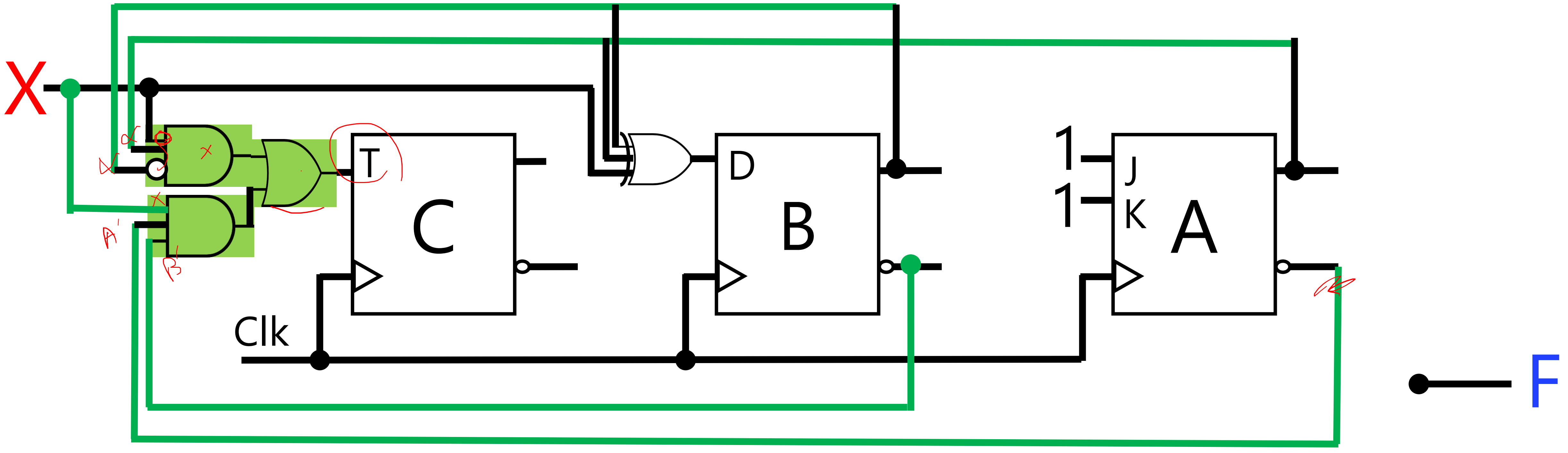
0. Do we need combinational logic or sequential logic? Do we need memory?
1. How many storage (flip-flops)? #FF
2. How many input and output?
3. Form the state (transition) diagram
4. Form the state table
5. Fill the state table
6. What type of storage (flip-flop)? RS, D, T, JK, or Mixed
7. Input (*excitation*)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



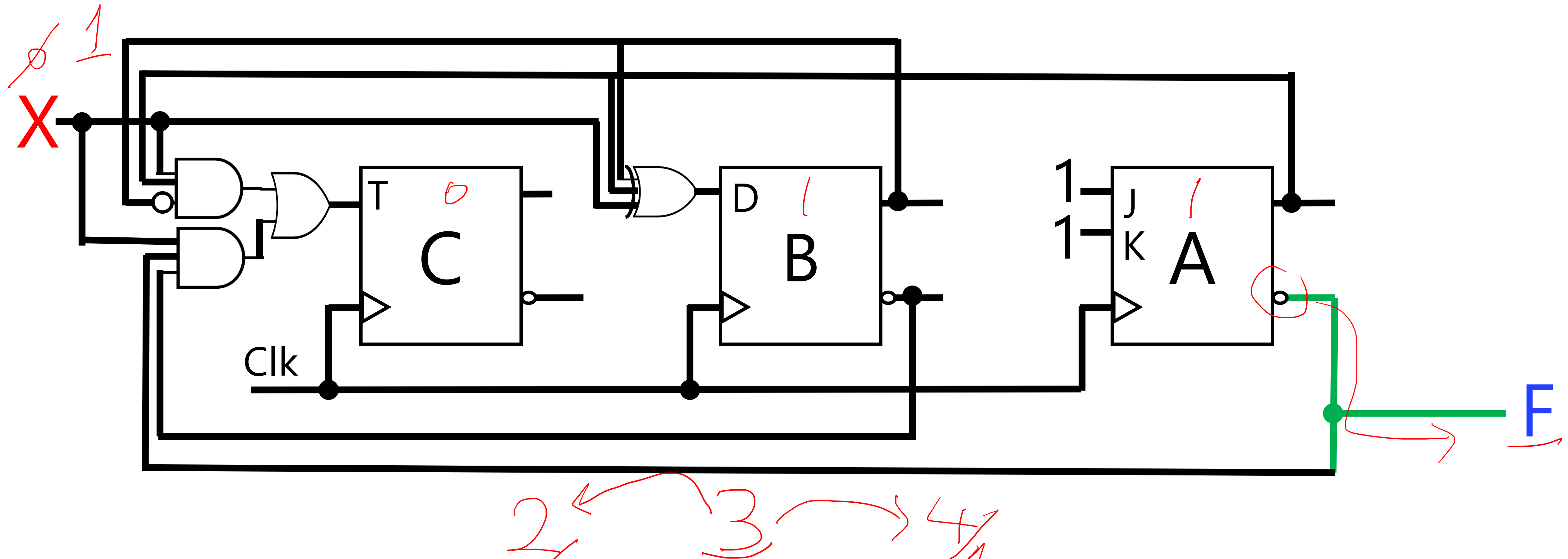
A	$J_A=1$	$K_A=1$
B	$D_B=F(X,C,B,A)=X \oplus B \oplus A$	
C	$T_C=F(X,C,B,A)=X'BA + XB'A'$	
F	$F(C,B,A) = A'$	



A	$J_A=1$	$K_A=1$
B	$D_B=F(X,C,B,A)=X\oplus B\oplus A$	
C	$T_C=F(X,C,B,A)=X'BA + XB'A'$	
F	$F(C,B,A) = A'$	

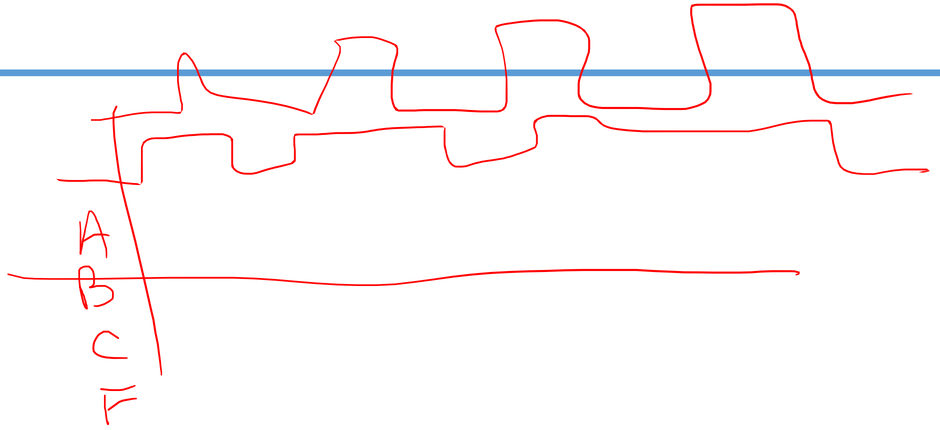


A	$J_A=1$	$K_A=1$
B	$D_B=F(X,C,B,A)=X \oplus B \oplus A$	
C	$T_C=F(X,C,B,A)=X'BA + XB'A'$	
F	$F(C,B,A) = A'$	



A	$J_A=1$	$K_A=1$
B	$D_B=F(X,C,B,A)=X\oplus B\oplus A$	
C	$T_C=F(X,C,B,A)=X'BA + XB'A'$	
F	$F(C,B,A) = A'$	

Synchronization
Practice Timing Diagram



Counter Up-Down
Example Mealy Model

Counting up and number is even
Counting up and number is odd
Counting down and number is odd
Counting down and number is even

Counter Up-Down

X=0: Count up from i to i+1

X=1: Count down from N i to i -1

F = 1 when counting up (X=0) and number is even

F = 0 when counting up (X=0) and number is odd

F = 1 when counting down (X=1) and number is odd

F = 0 when counting down (X=1) and number is even

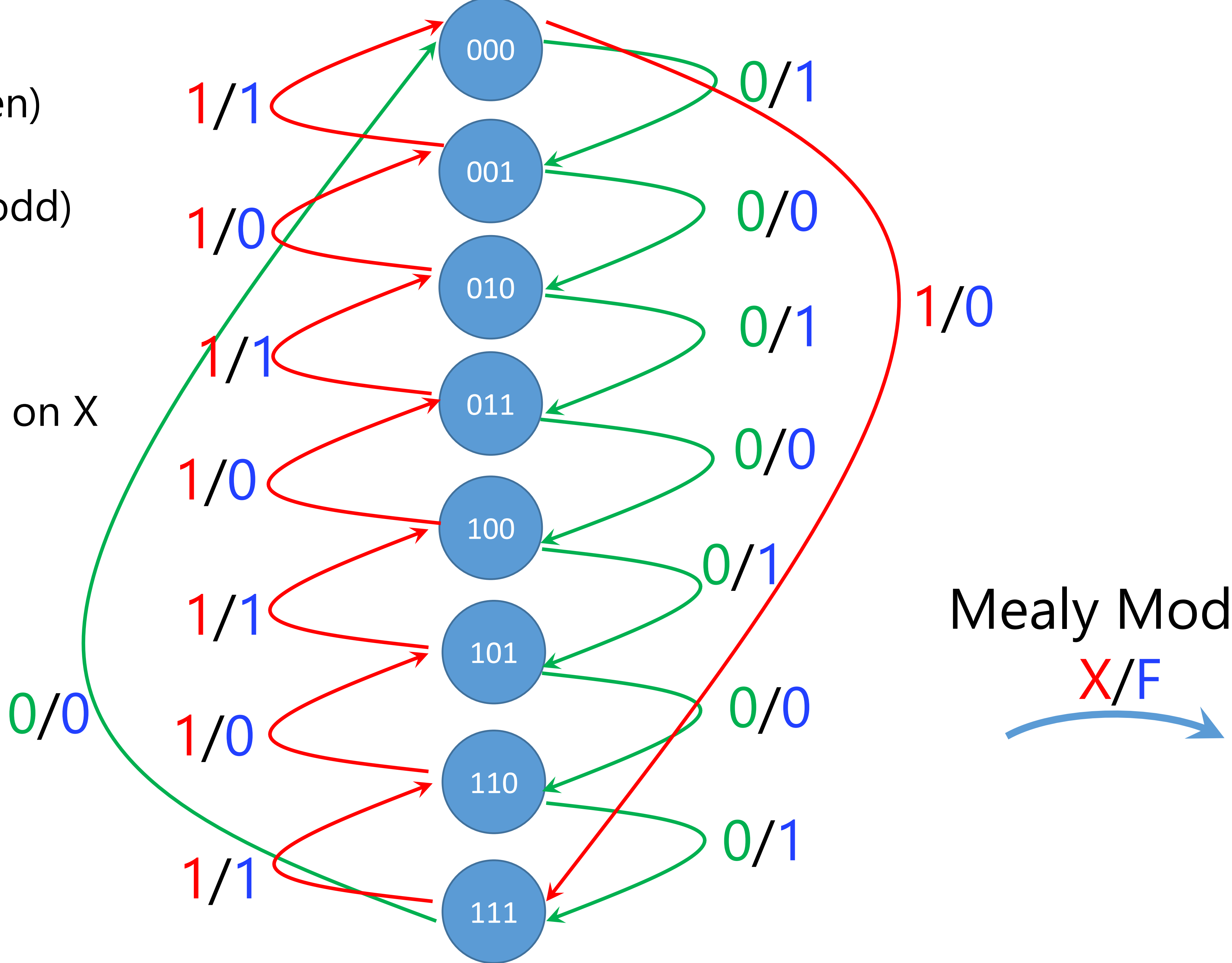
F=1: (up AND even)

OR

(down AND odd)

F=0 : else

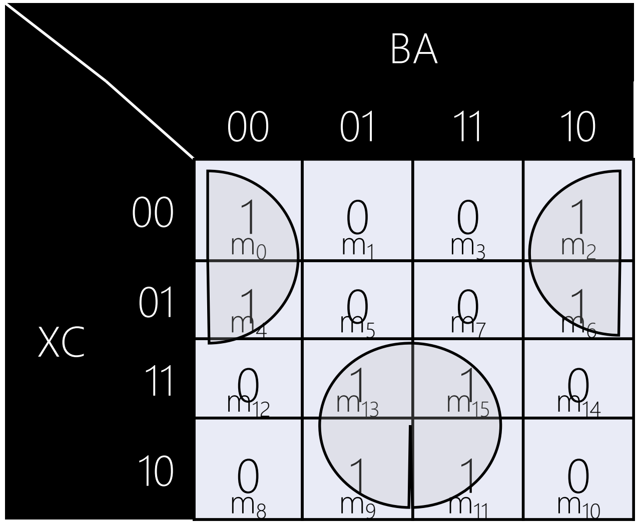
Y clearly depends on X



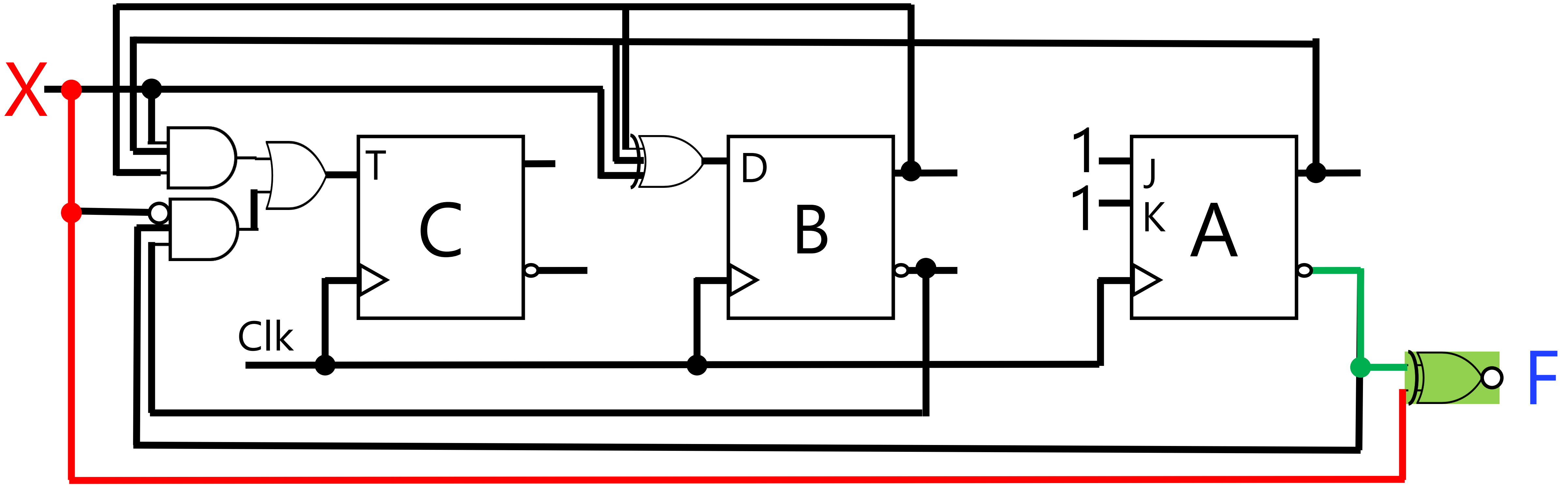
Inputs	Q(T)			Q(T+1)			Outputs
X	C	B	A	C	B	A	F
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	1	1
0	0	1	1	1	0	0	0
0	1	0	0	1	0	1	1
0	1	0	1	1	1	0	0
0	1	1	0	1	1	1	1
0	1	1	1	0	0	0	0
1	0	0	0	1	1	1	0
1	0	0	1	0	0	0	1
1	0	1	0	0	0	1	0
1	0	1	1	0	1	0	1
1	1	0	0	0	1	1	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	0
1	1	1	1	1	1	0	1

$$Y = F(\underline{X},C,B,A) = \sum(0,2,4,6,9,11,13,15)$$

Minimization for output F
4-Variable K-Map



$$F = F(\underline{X},C,B,A) = \sum(0,2,4,6,9,11,13,15)$$
$$= X'A' + XA$$
$$= X \odot A$$



A	$J_A=1$ $K_A=1$
B	$D_B=F(X,C,B,A)=X\oplus B\oplus A$
C	$T_C=F(X,C,B,A)=X'BA + XB'A'$
F	$F(C,B,A) = X\odot A$

Asynchronization
Practice Timing Diagram