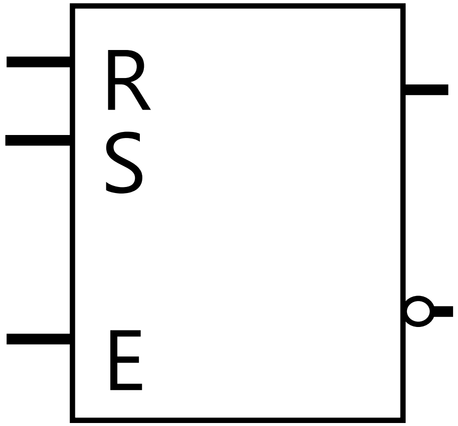
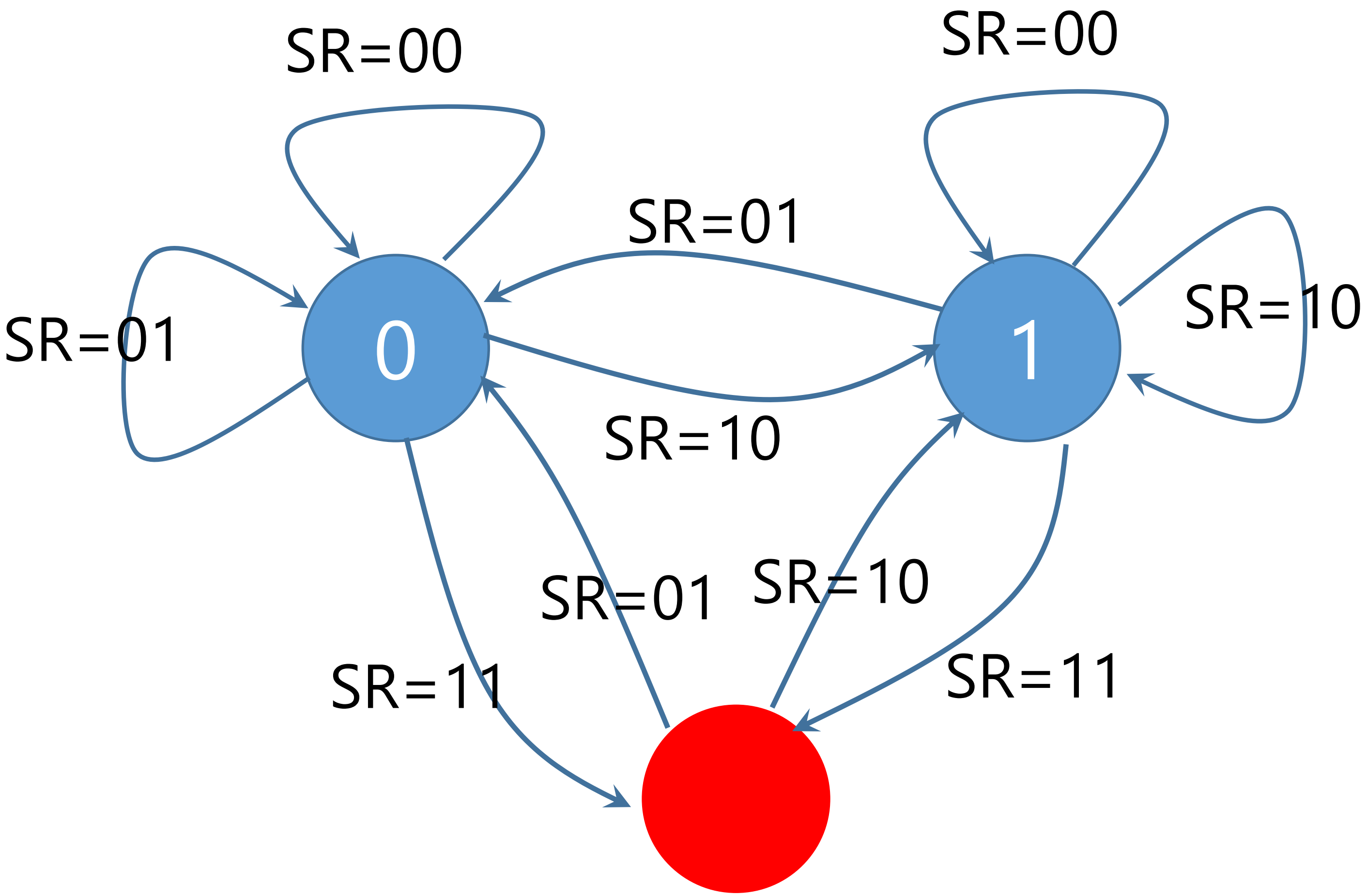
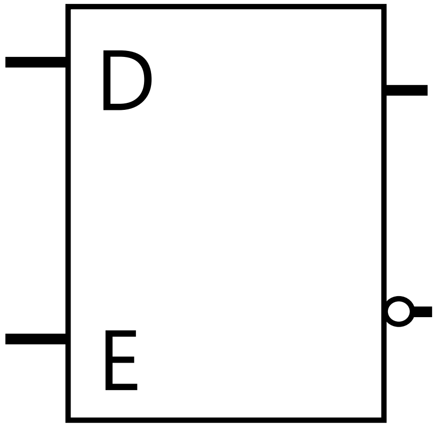
Flip-Flop

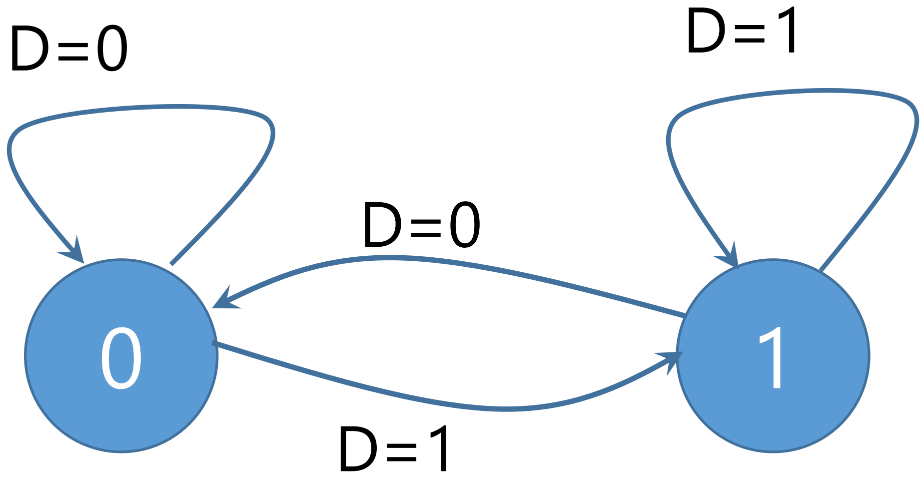


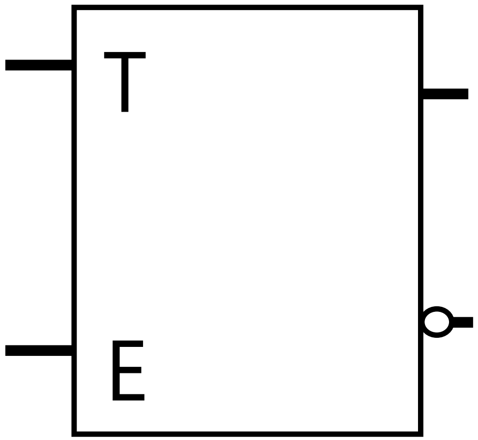
S	R	Q
0	0	Q_t
0	1	0
1	0	1
1	1	X



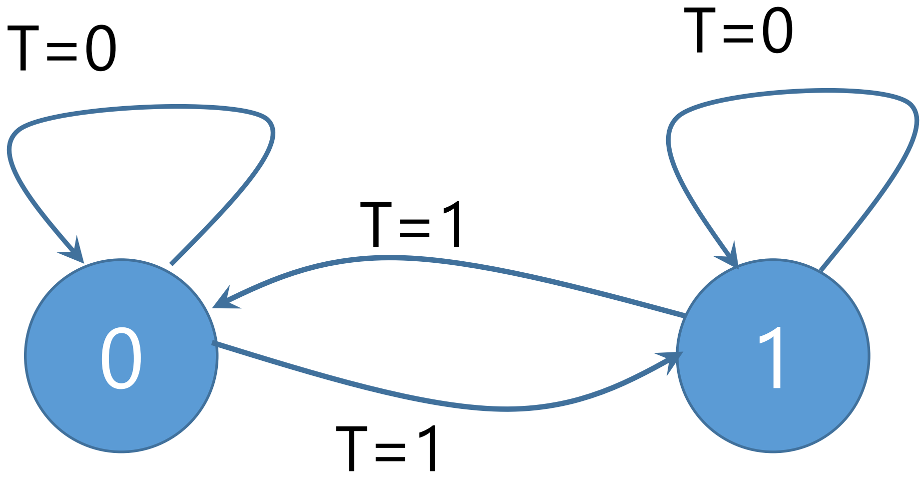


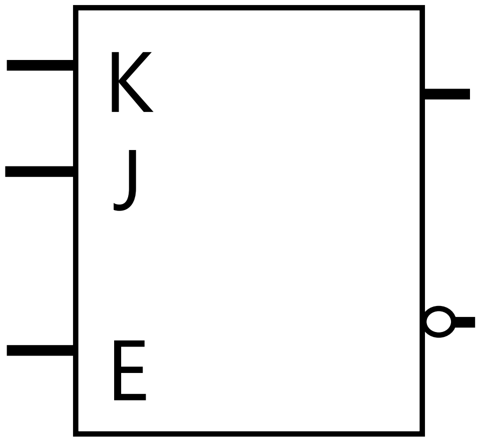
D	Q
0	0
1	1



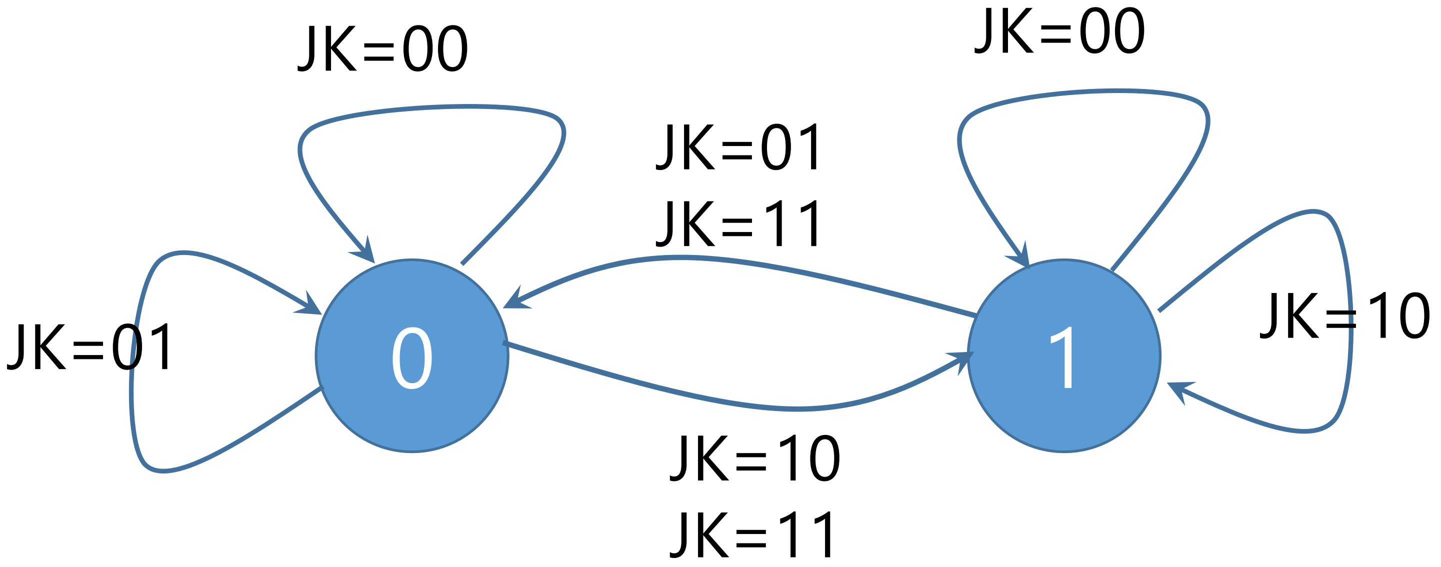


T	Q
0	Q_t
1	Q'_t

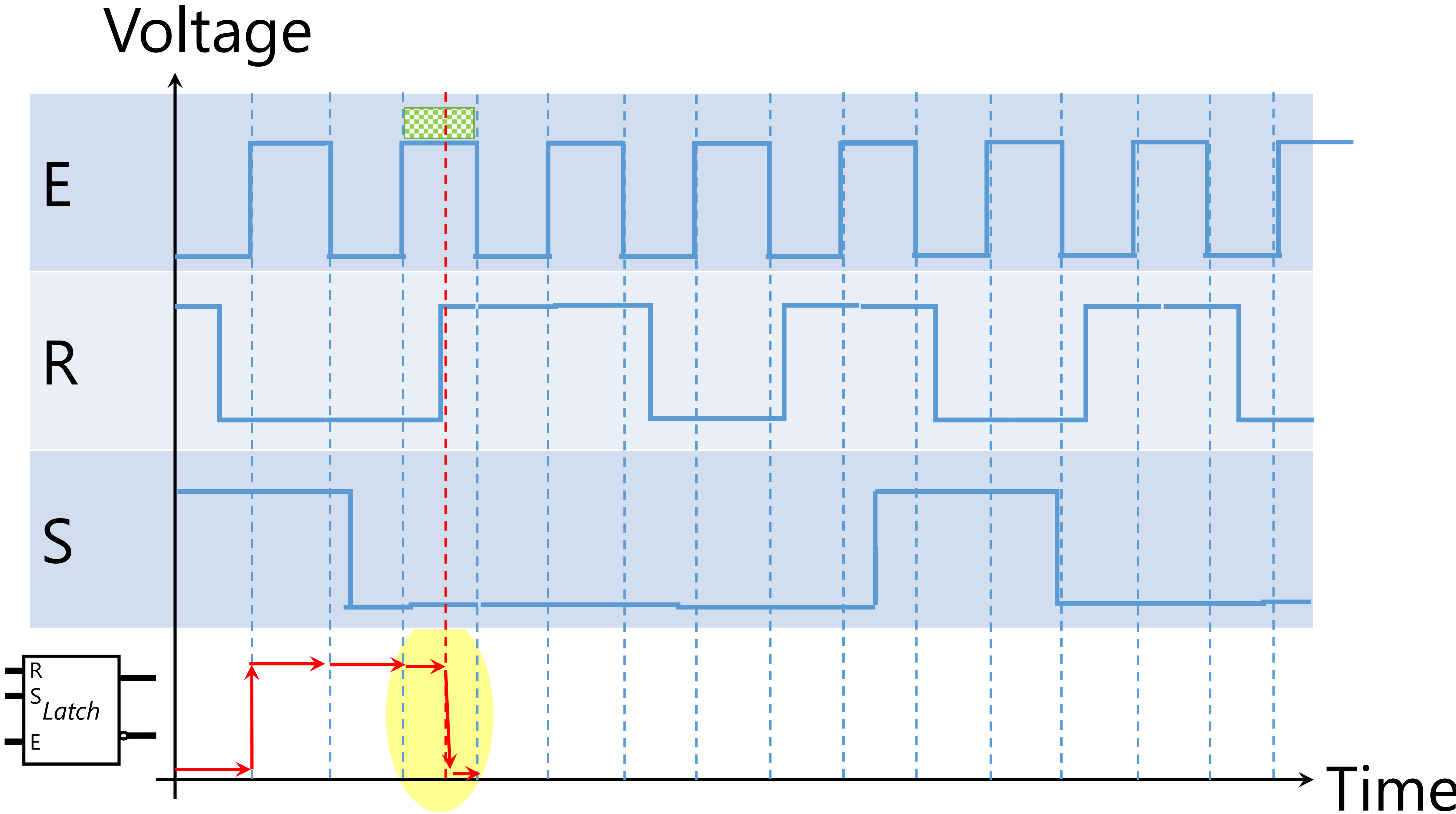




J	K	Q
0	0	Q_t
0	1	0
1	0	1
1	1	Q'_t



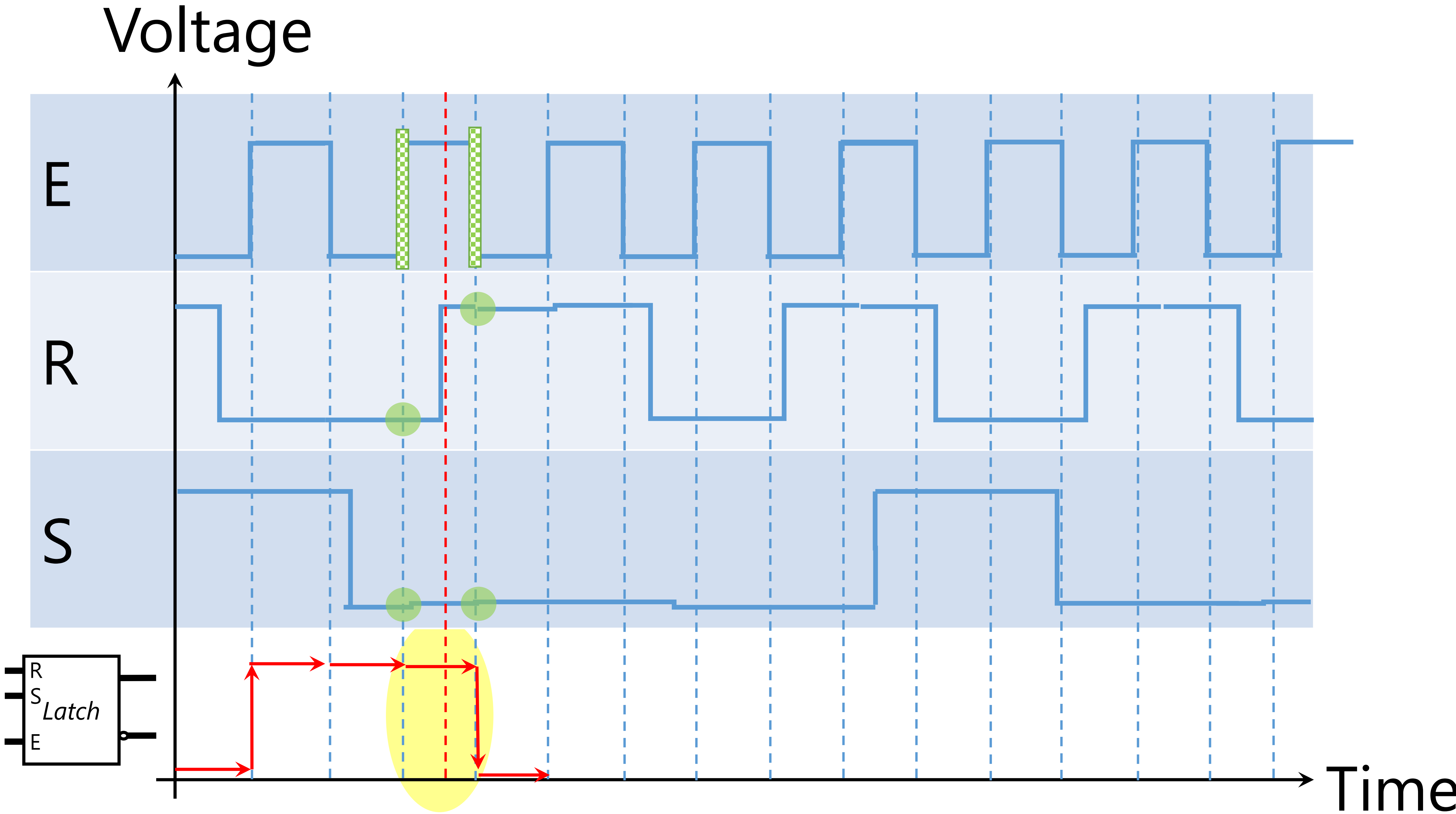
Flip-Flop



Voltage
E
R
S
R
S
Latch
E
Time

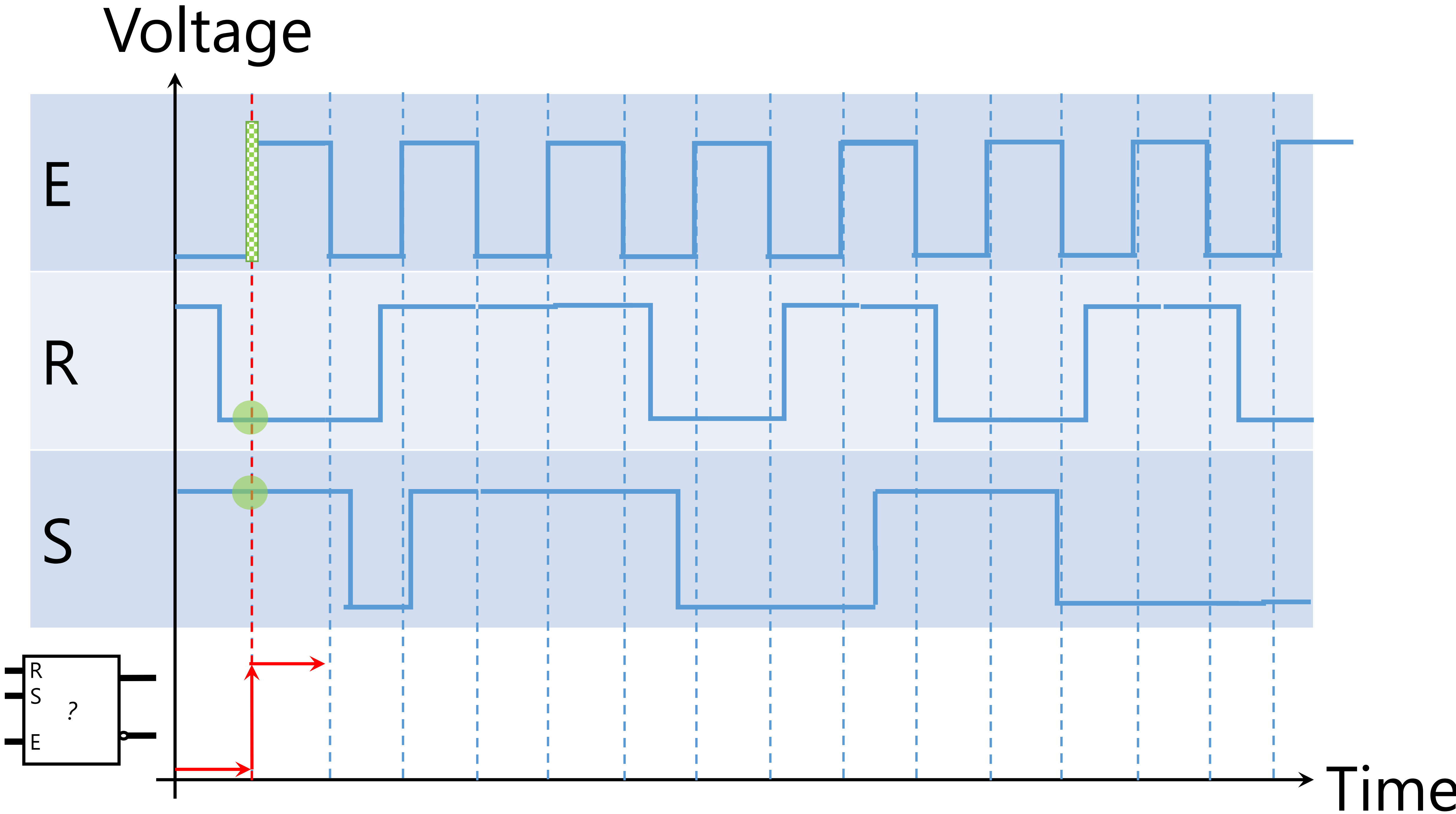
Active High (low) is controlling the change to only a specific time period, but is it possible to reduce it to a moment?


Peephole

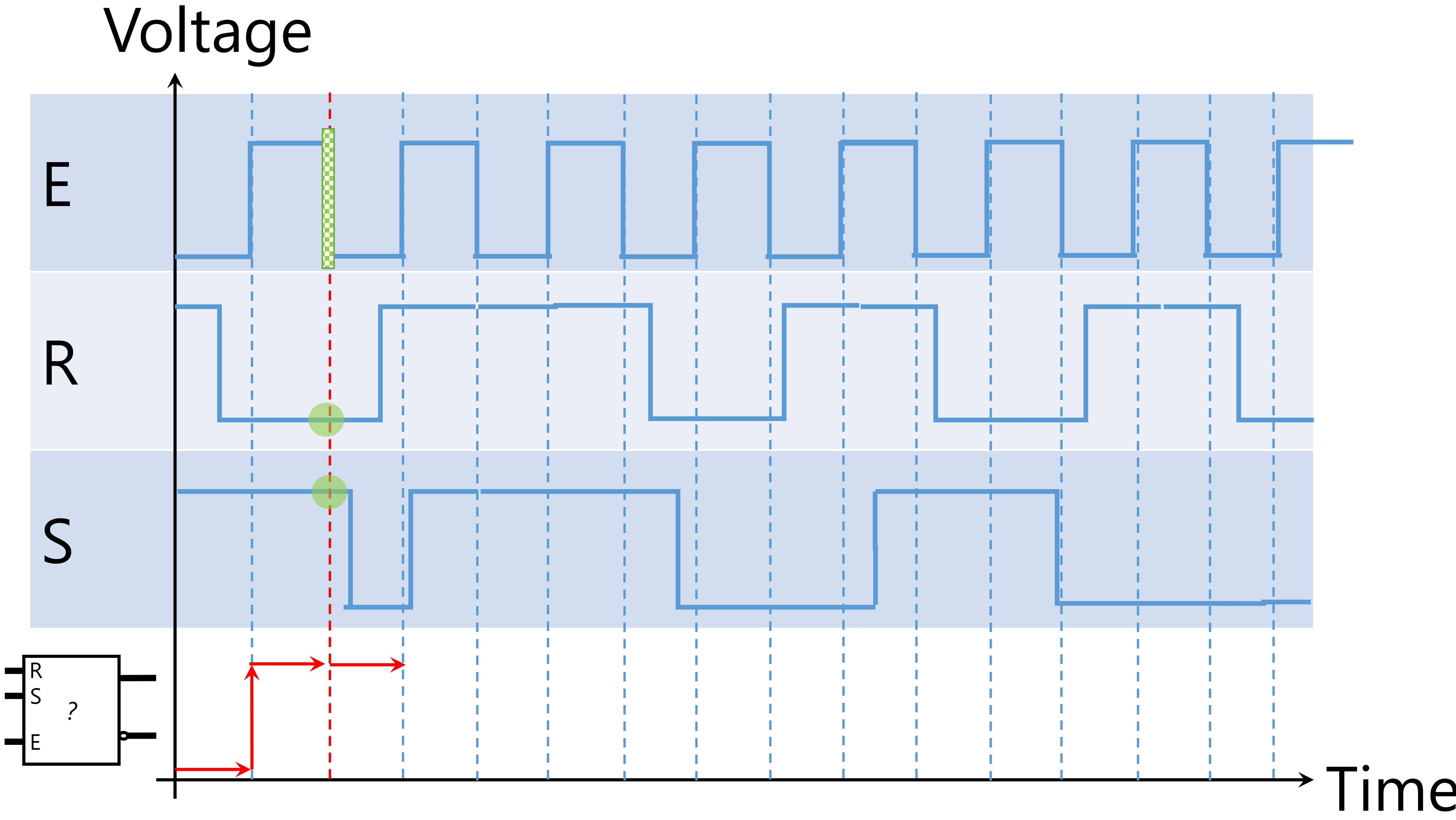


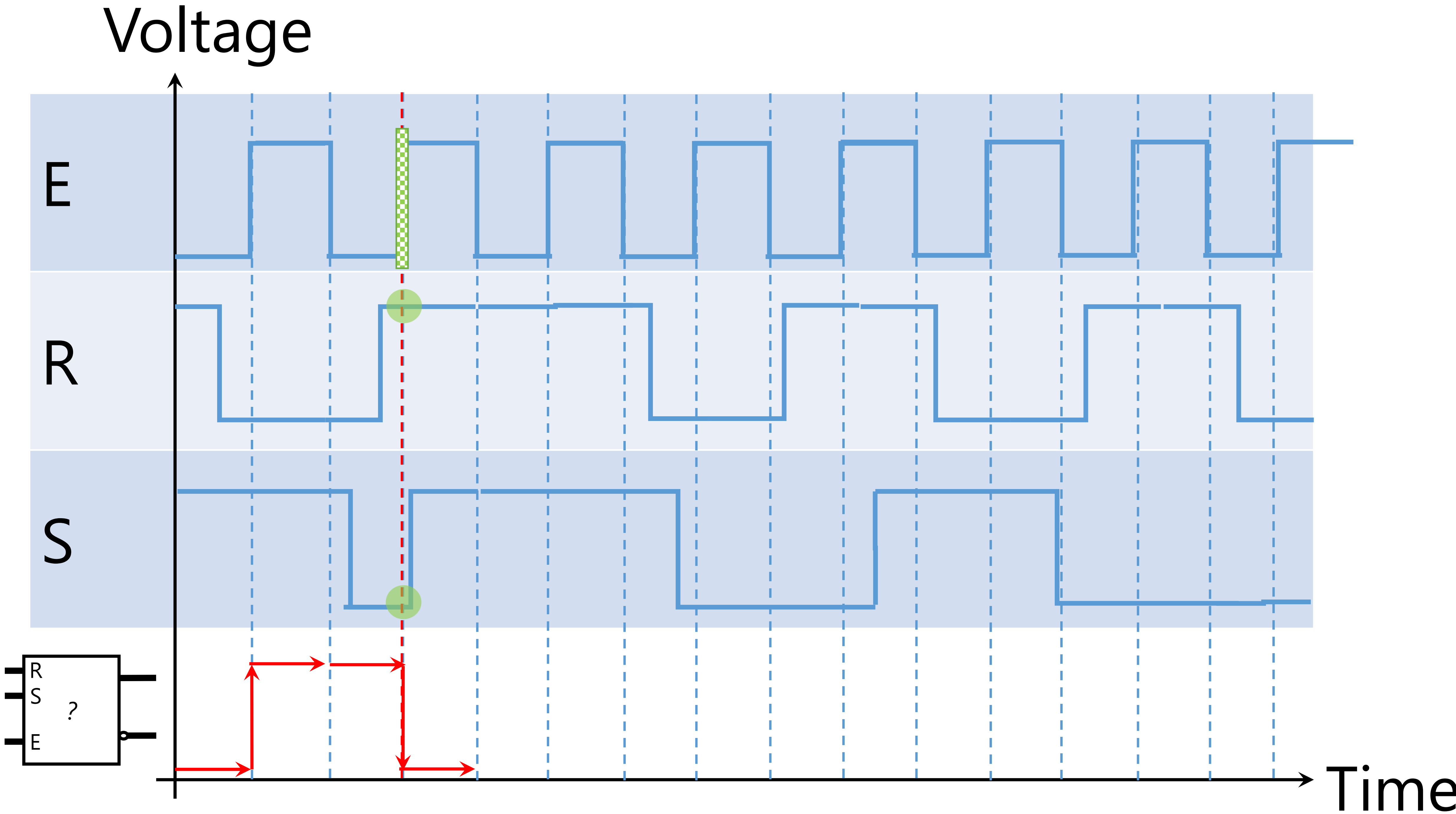
Voltage
E
R
S
R
S
Latch
E
Time

Level vs. Edge



Voltage
E
R
S
R
S
?
E
Time


Voltage
E
R
S
R
S
?
E
Time



Voltage
E
R
S
R
S
?
E
Time

Both Edge

From one edge to the next edge, locked!

Both vs. Single Edge

Single Edge
Positive vs. Negative

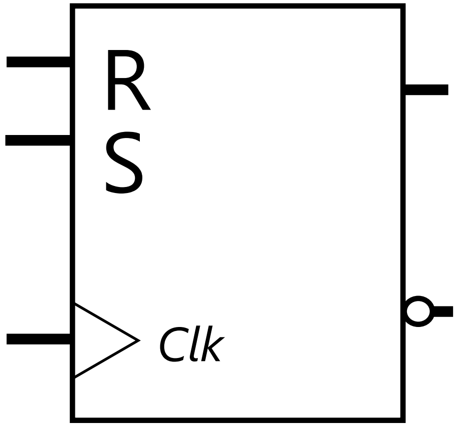
(b) Positive-edge response

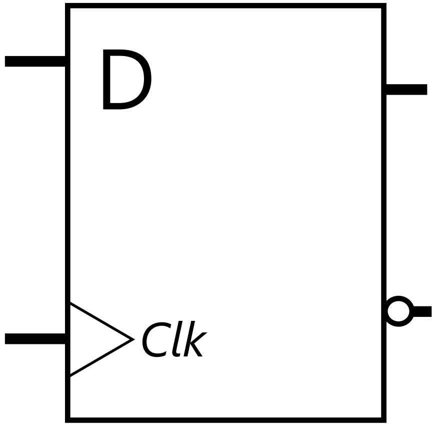
(c) Negative-edge response

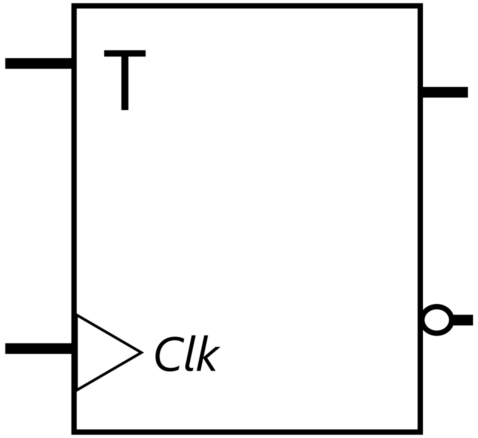
FIGURE 5.8
Clock response in latch and flip-flop

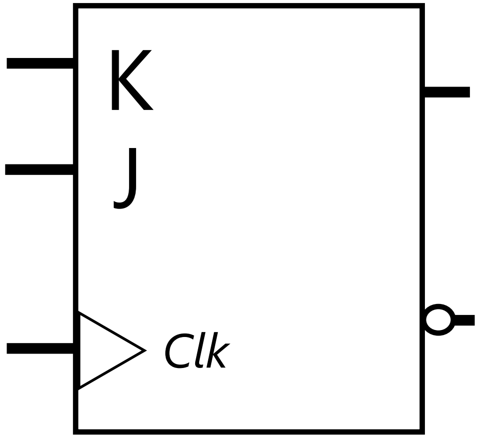
Flip-Flop

A single edge enabled latch









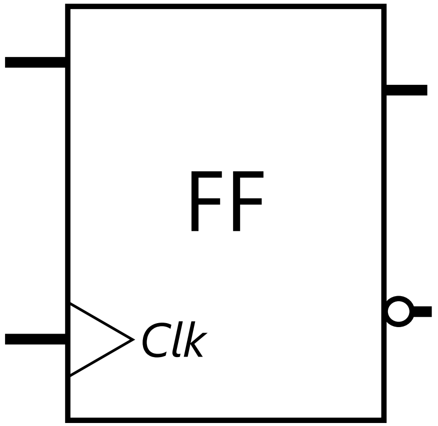
S	R	Q
0	0	Q_t
0	1	0
1	0	1
1	1	X

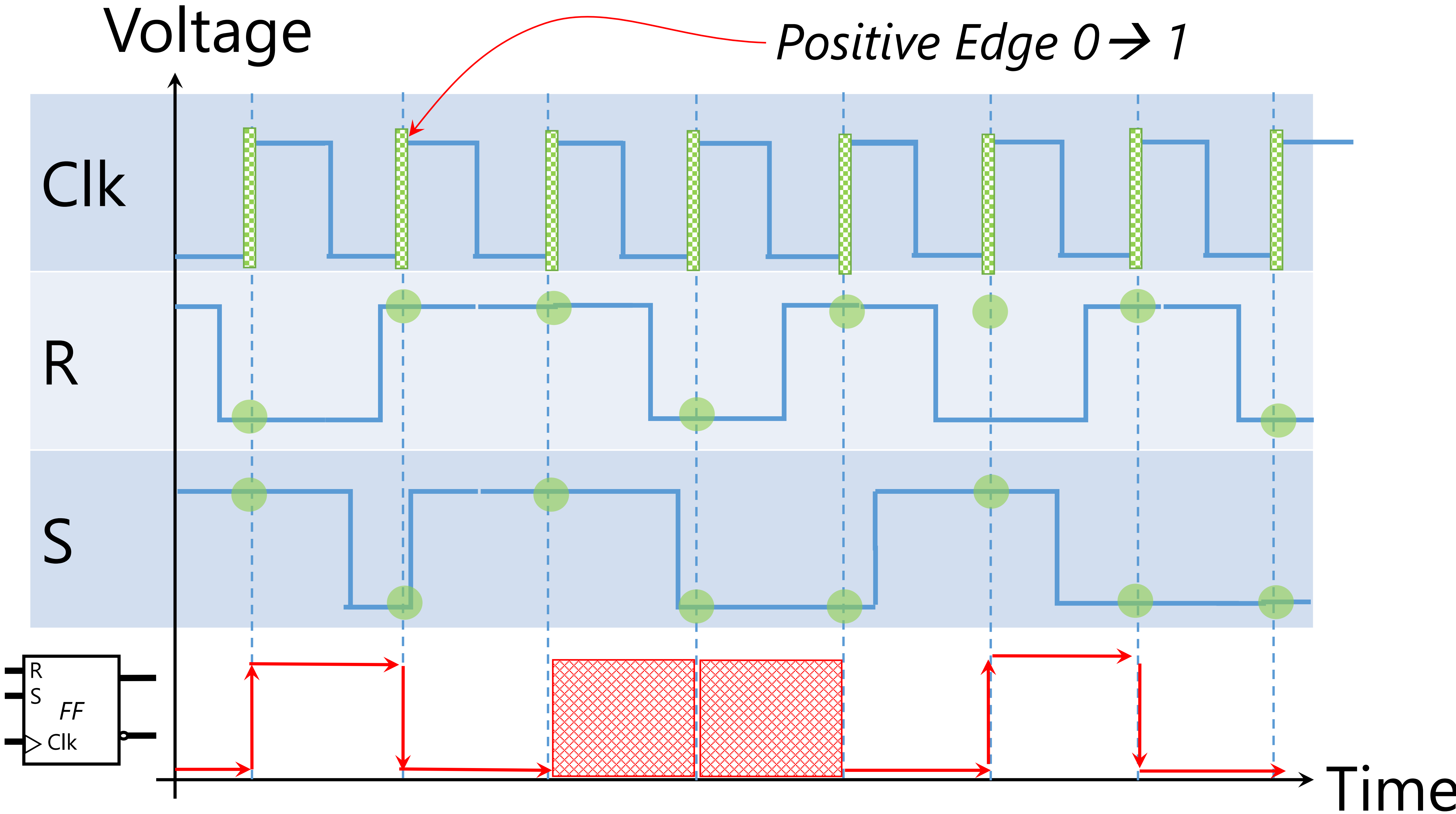
D	Q
0	0
1	1

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

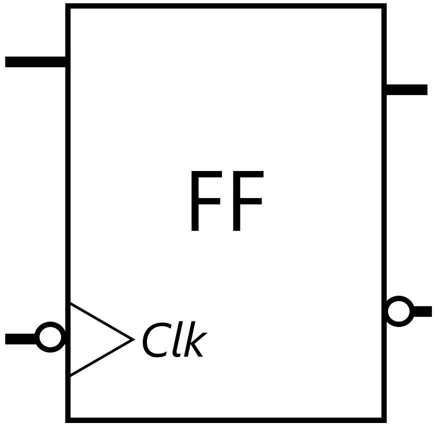
Single Edge
Positive

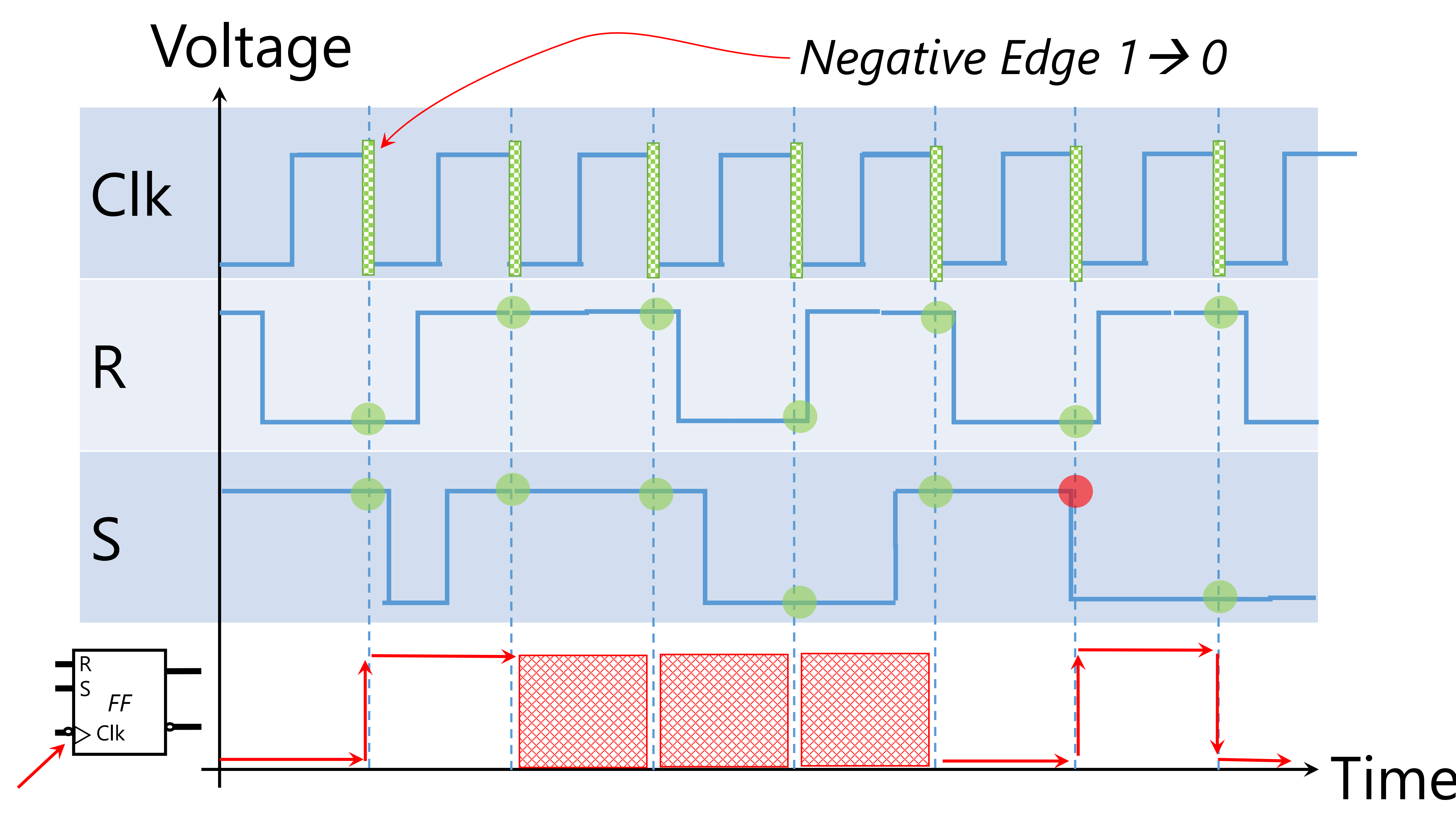




Voltage
Positive Edge 0→ 1
Clk
R
S
R
S
FF
Clk
Time

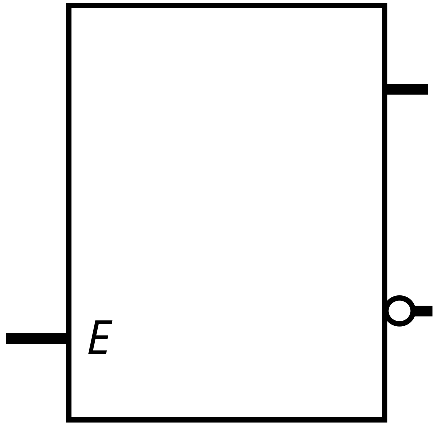
Single Edge
Negative

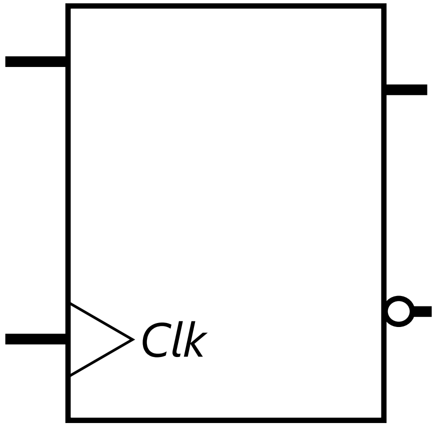


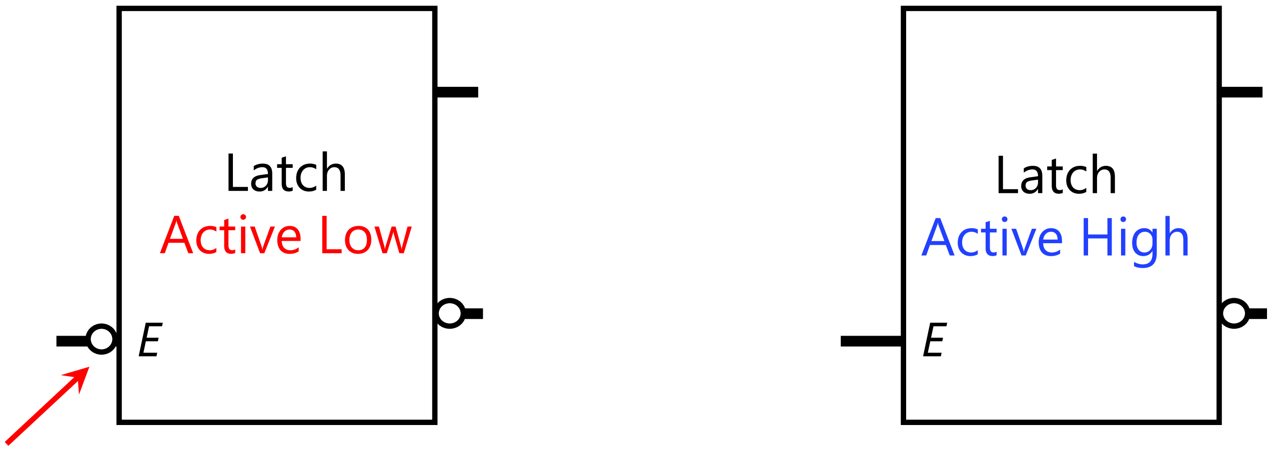


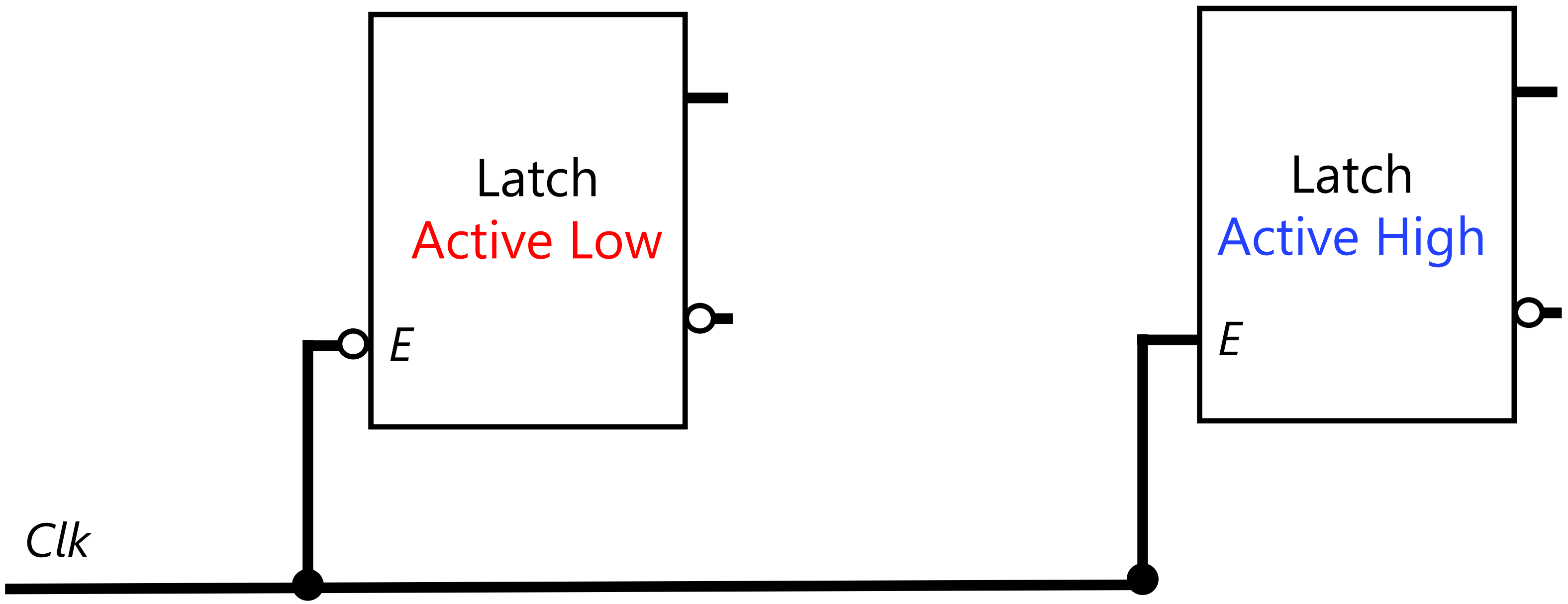
Voltage
Negative Edge 1 → 0
Clk
R
S
R
S
FF
Clk
Time

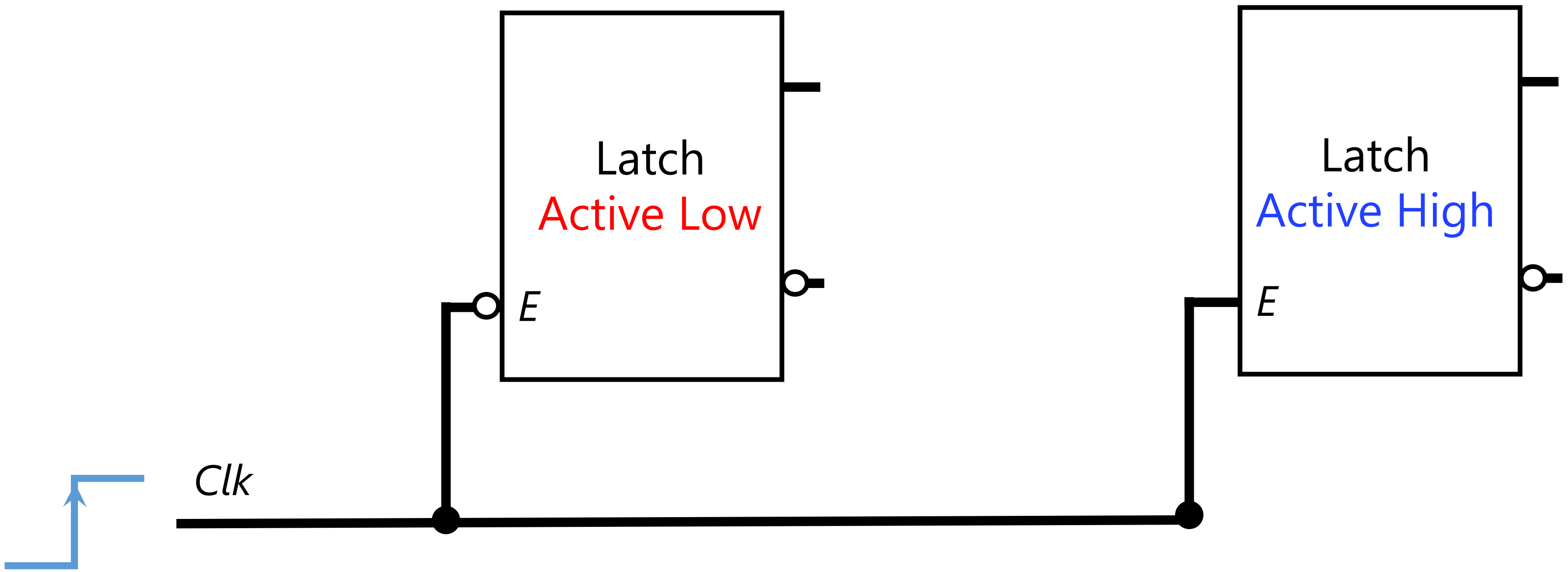
Latch → Flip-Flop

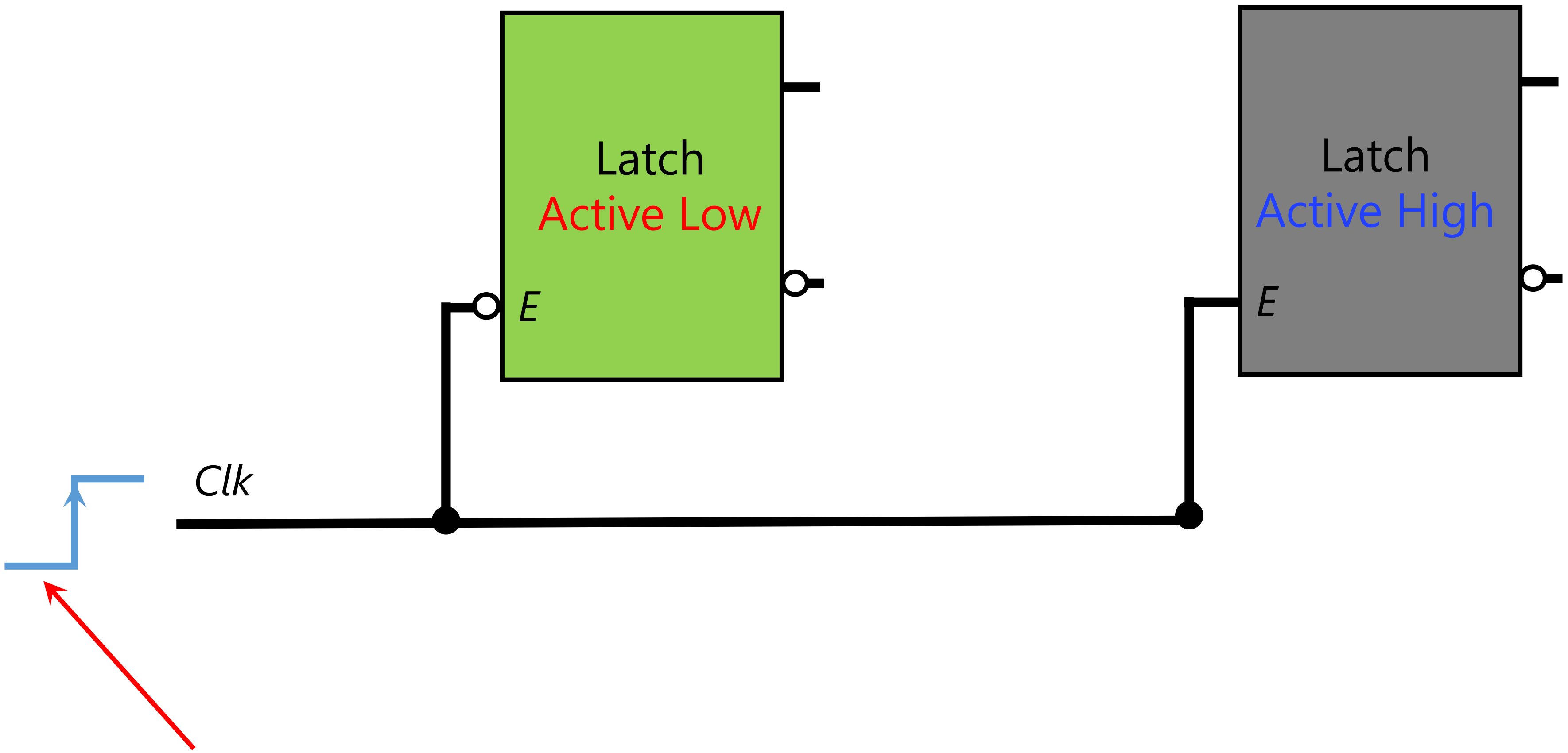


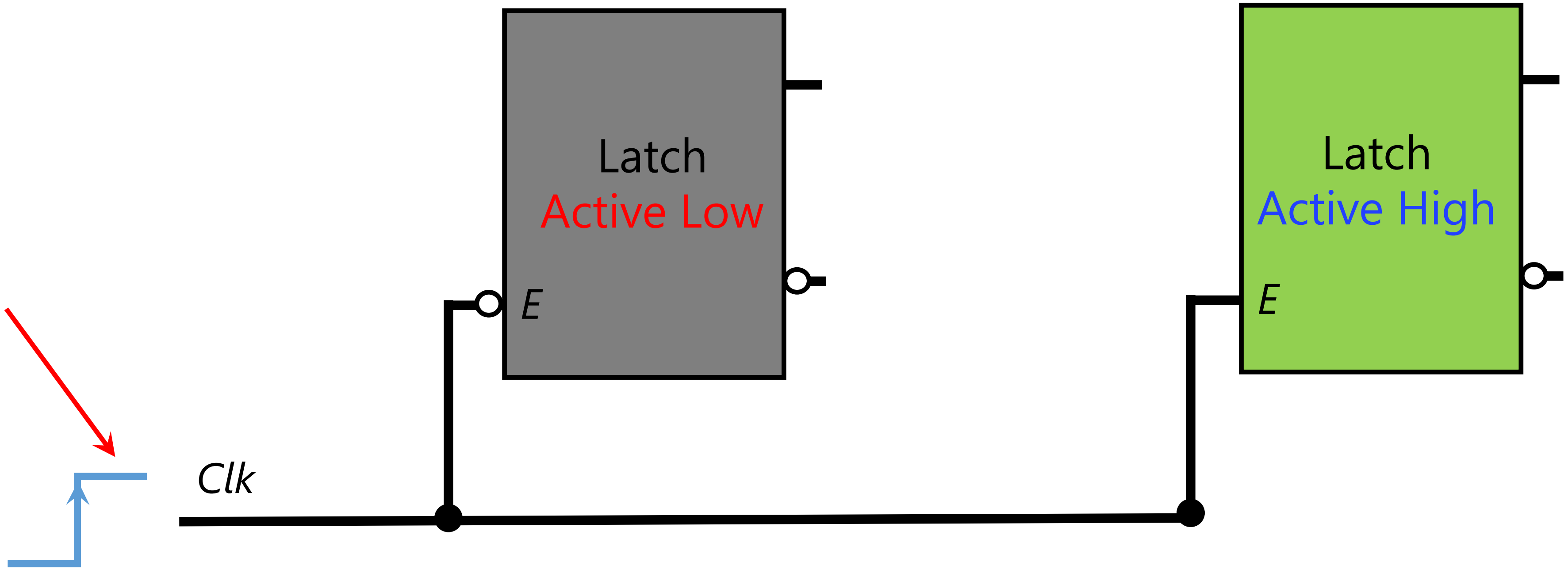



Latch
Active Low
E
Latch
Active High
E

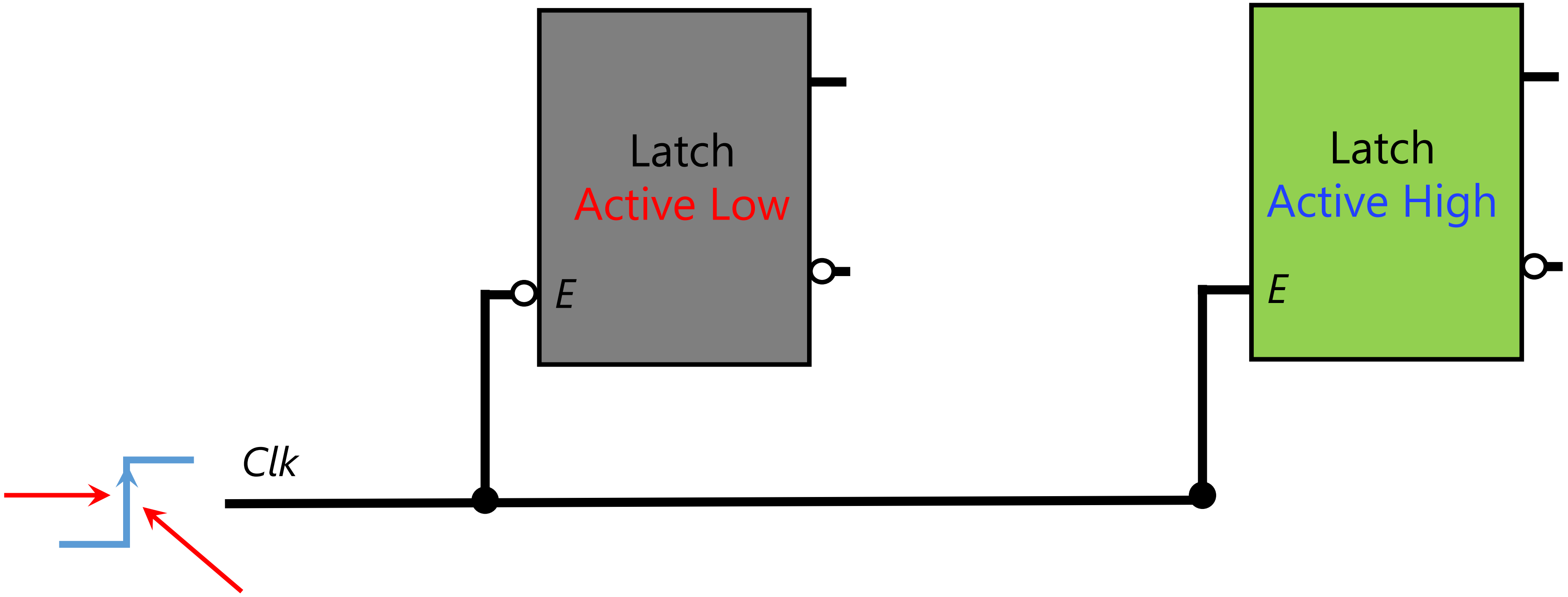

Latch
Active Low
E
Latch
Active High
E
Clk


Latch
Active Low
E
Latch
Active High
E
Clk

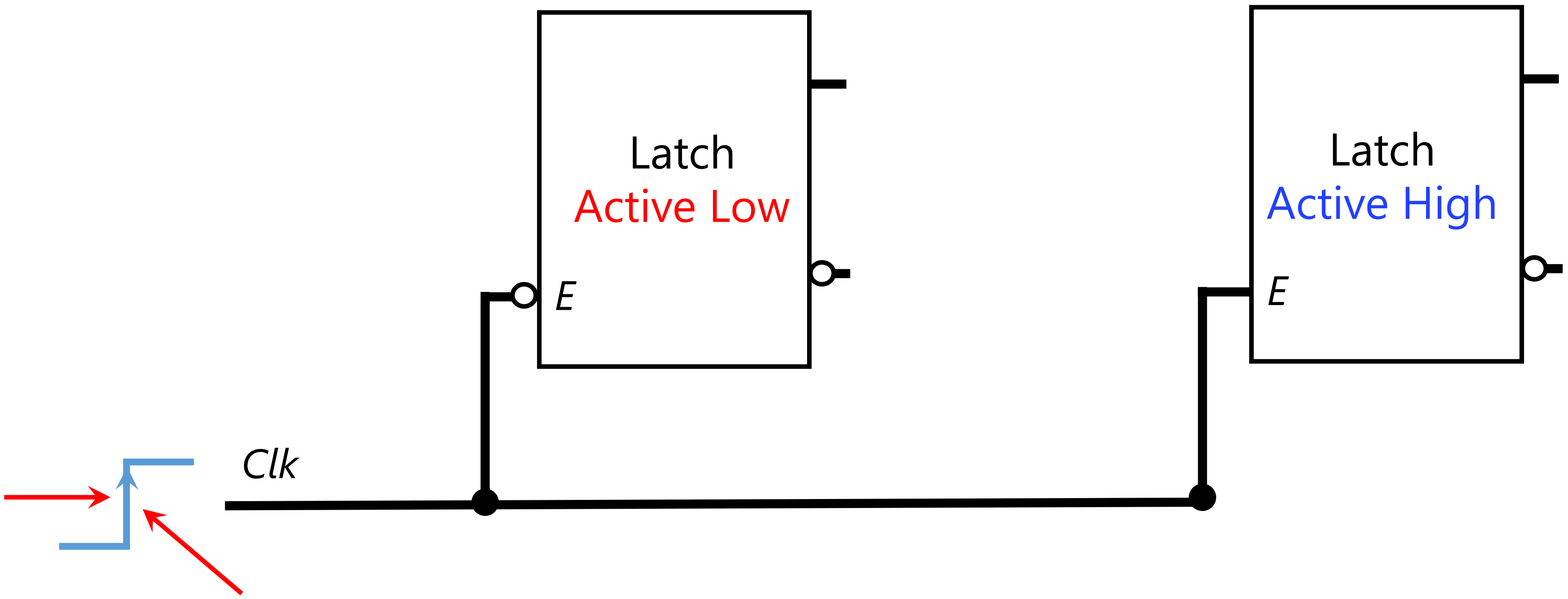

Latch
Active Low
E
Latch
Active High
E
Clk


Latch
Active Low
E
Latch
Active High
E
Clk

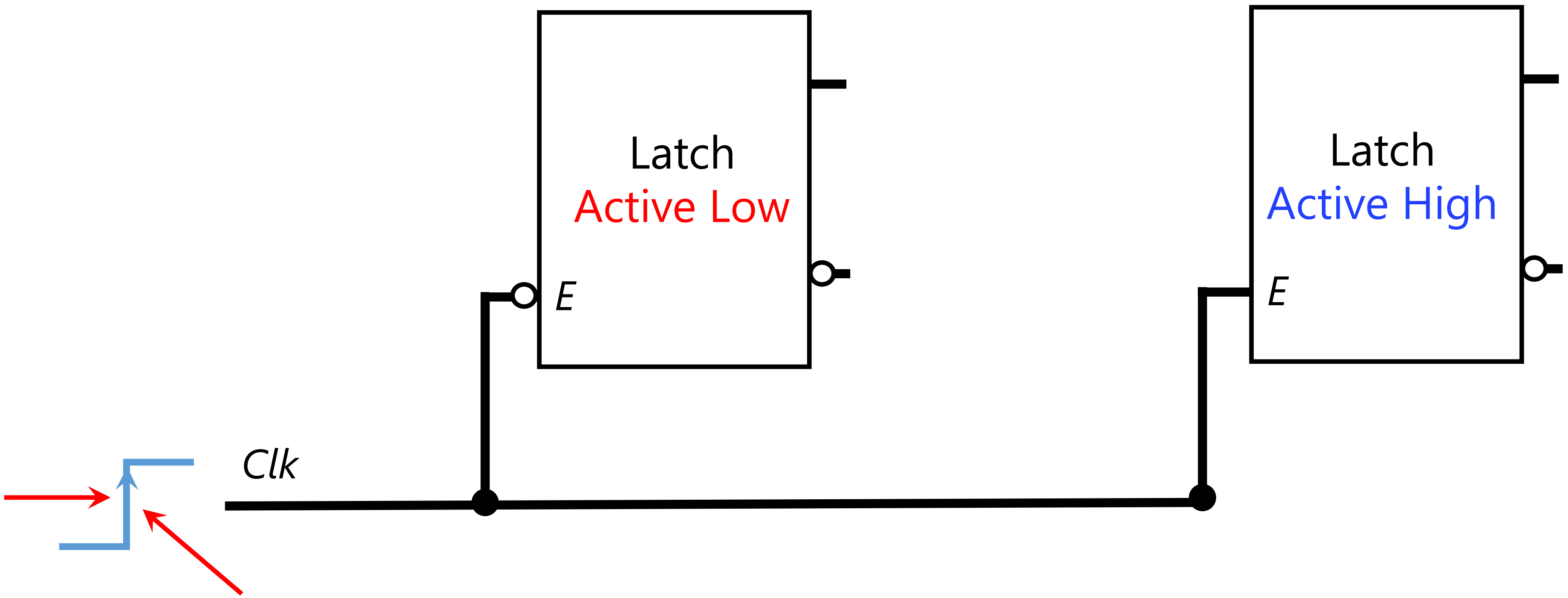
Opening the first → Closing the second
Closing the first → Opening the second



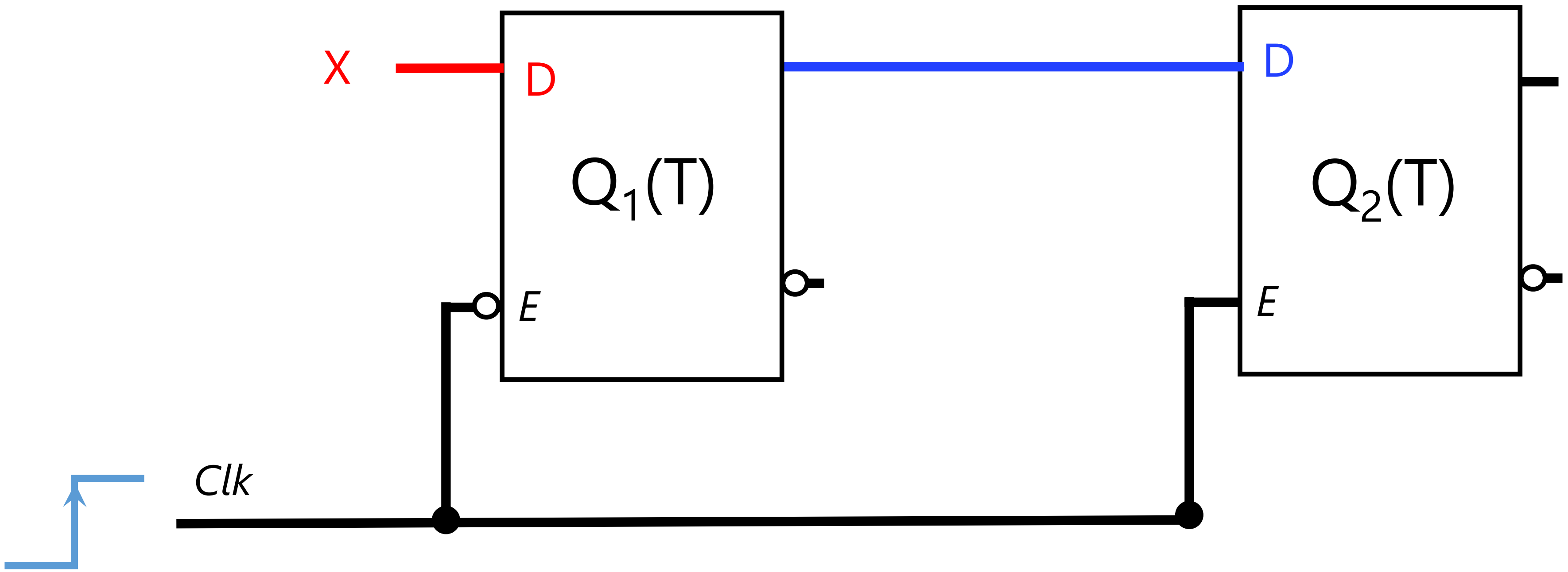
One “quick” moment that the first latch is becoming stable and closing the door!
Instead, the second latch is going to accept change.



One “quick” moment that the first latch is becoming stable and closing the door!
Instead, the second latch is going to accept change.



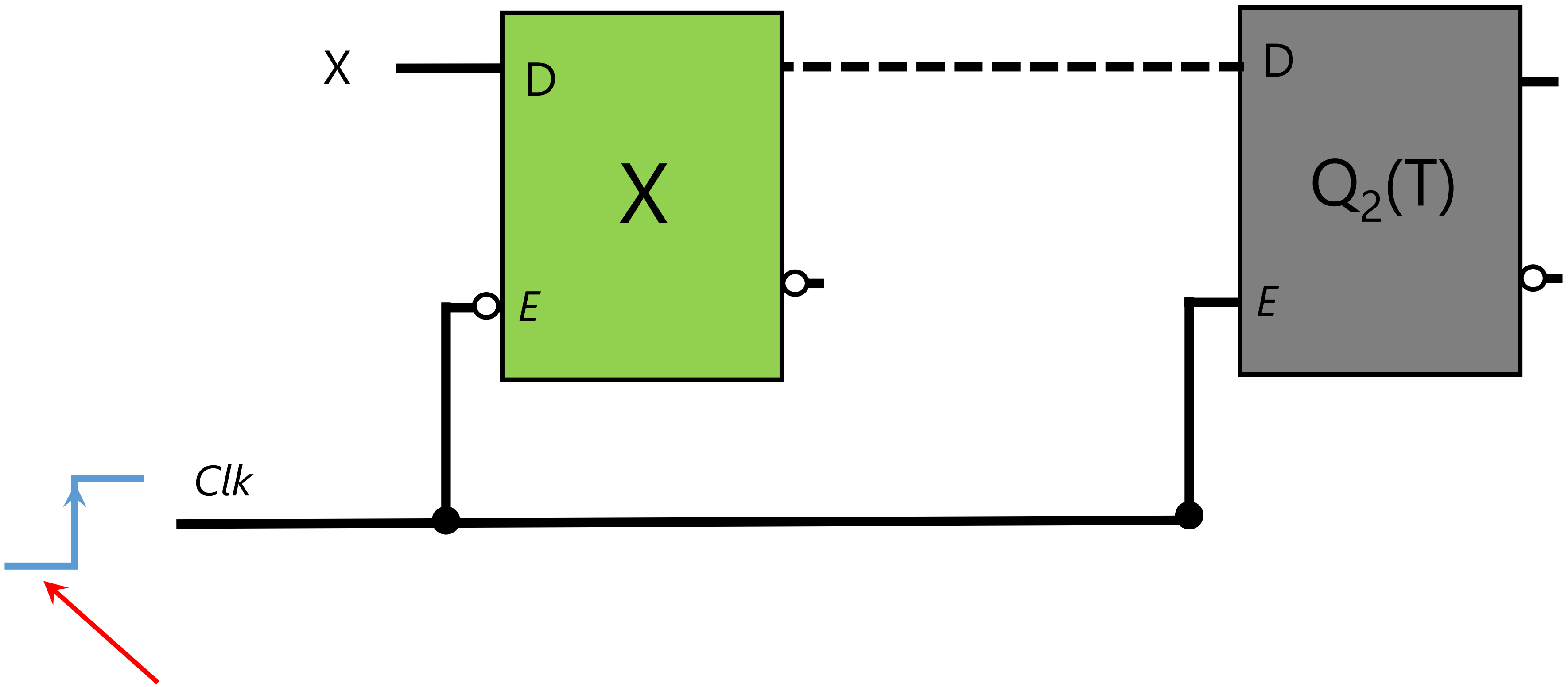
One “quick” moment that the first latch is becoming stable and closing the door!
Instead, the second latch is going to accept change.
Let’s force the second latch accept change from the first latch only!


X
D
Q1(T)
E
D
Q2(T)
E
Clk

Remember: In D latch, when enabled, whatever in input changes the state:

$X=0 \rightarrow Q=0$

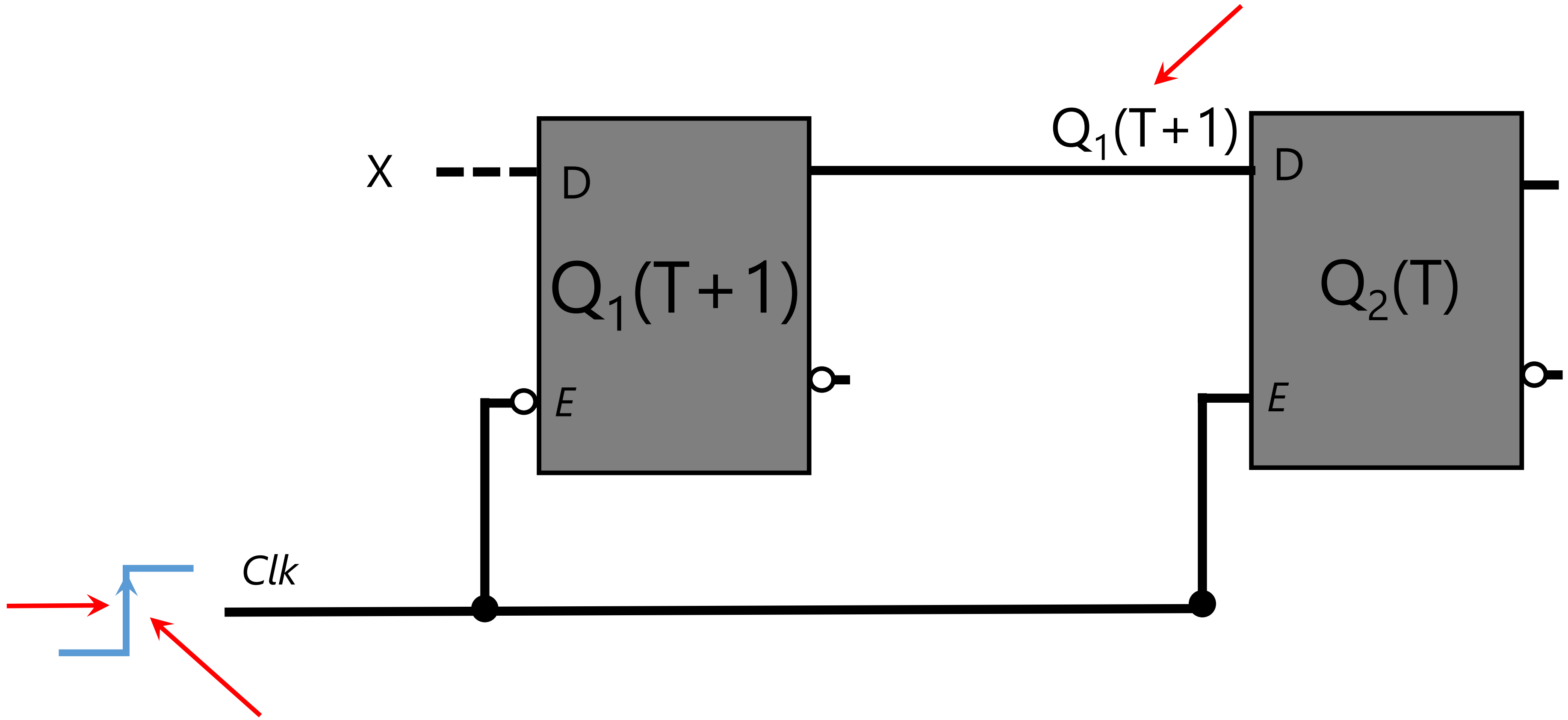
$X=1 \rightarrow Q=1$



Remember: In D latch, when enabled, whatever in input changes the state:

$X=0 \rightarrow Q=0$

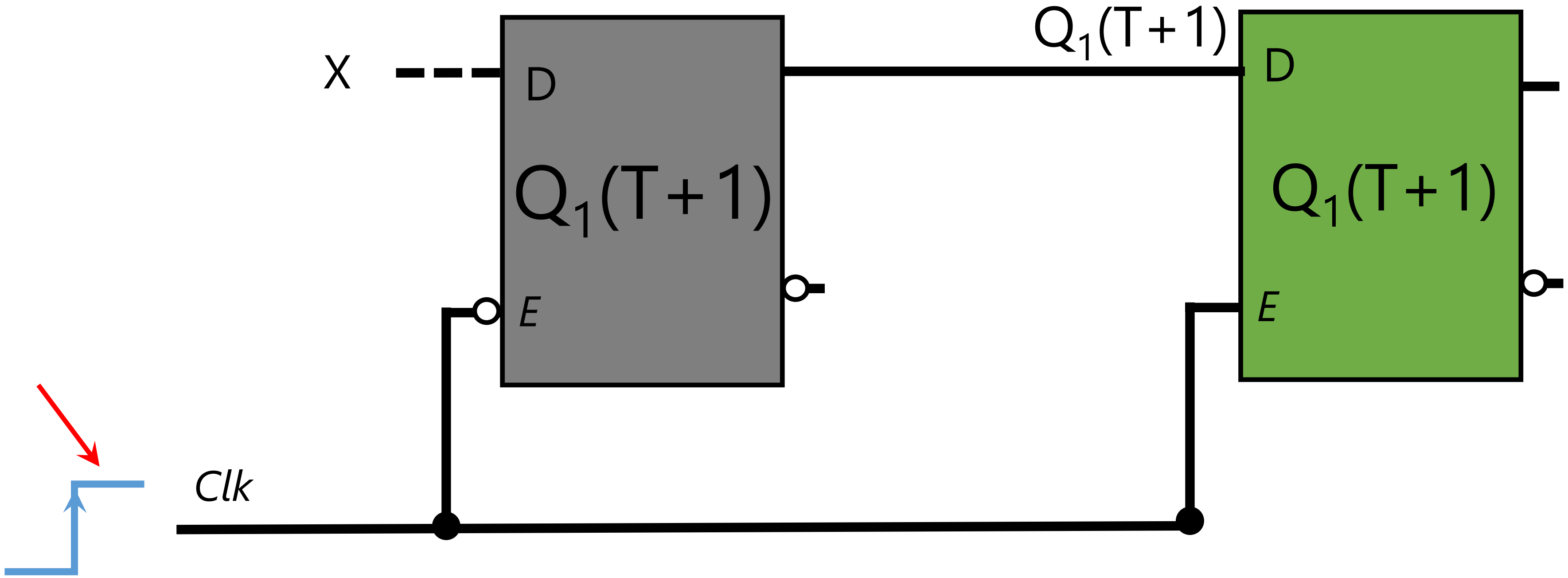
$X=1 \rightarrow Q=1$

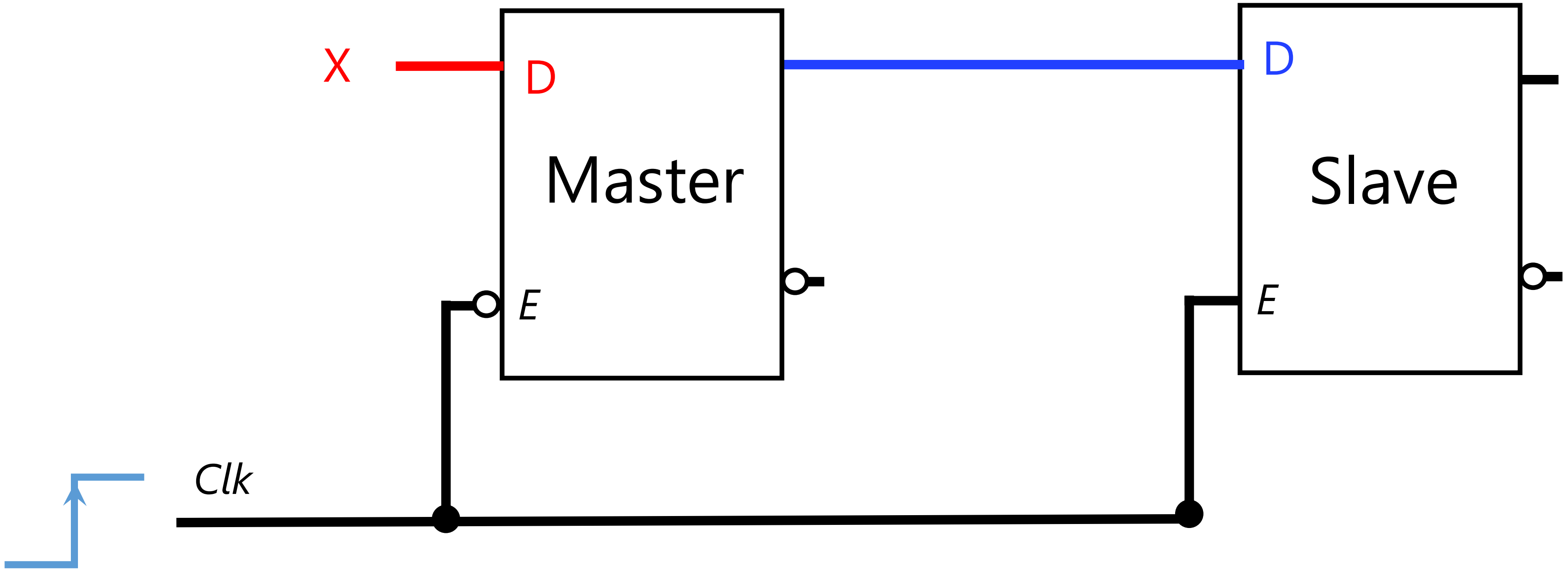


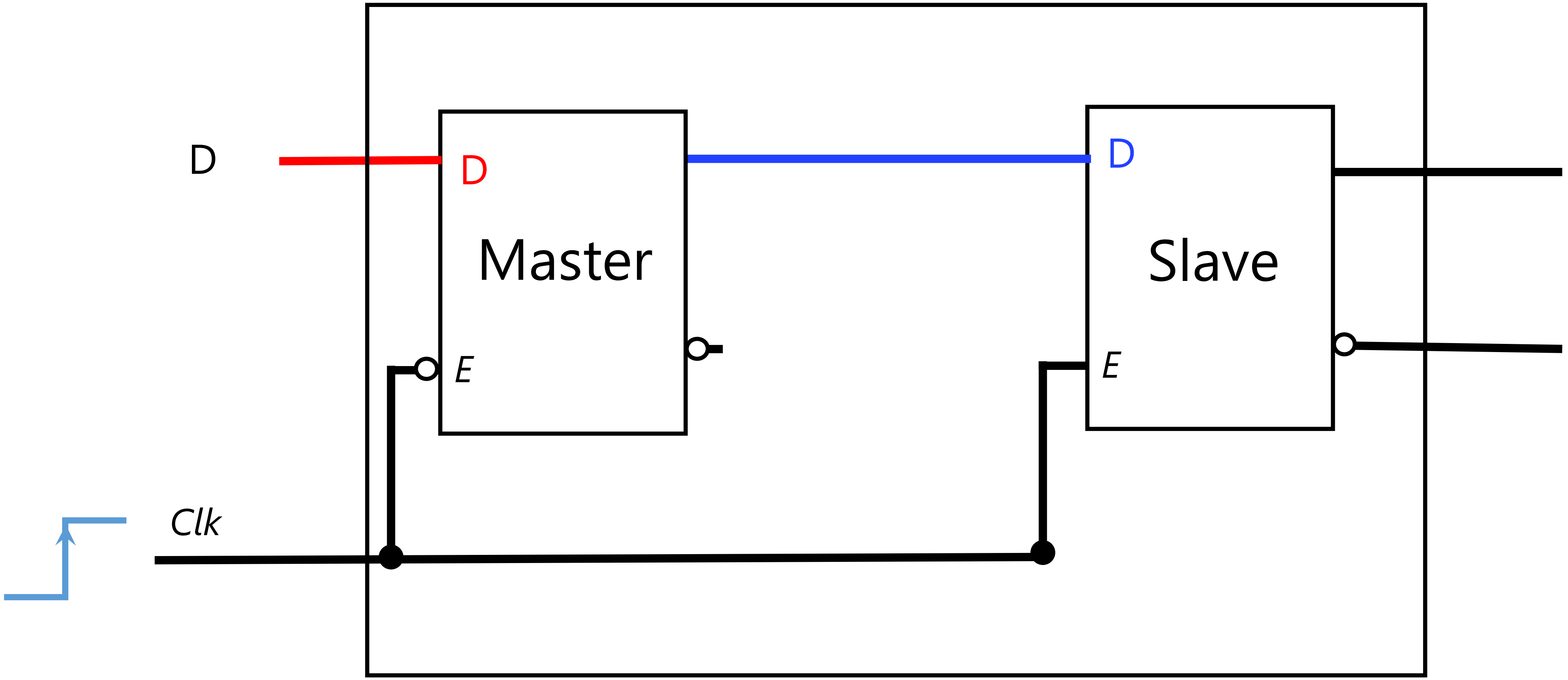
Remember: In D latch, when enabled, whatever in input changes the state:

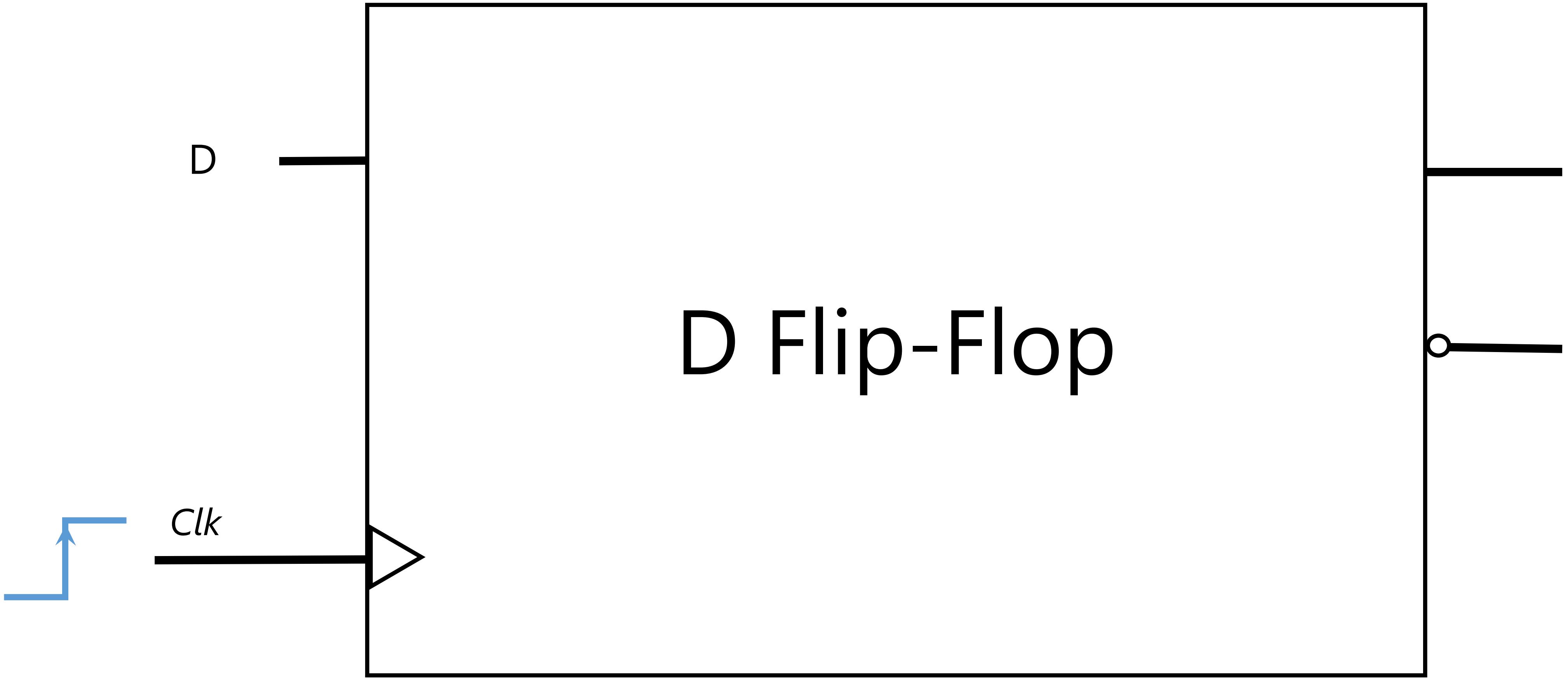
$X=0 \rightarrow Q=0$

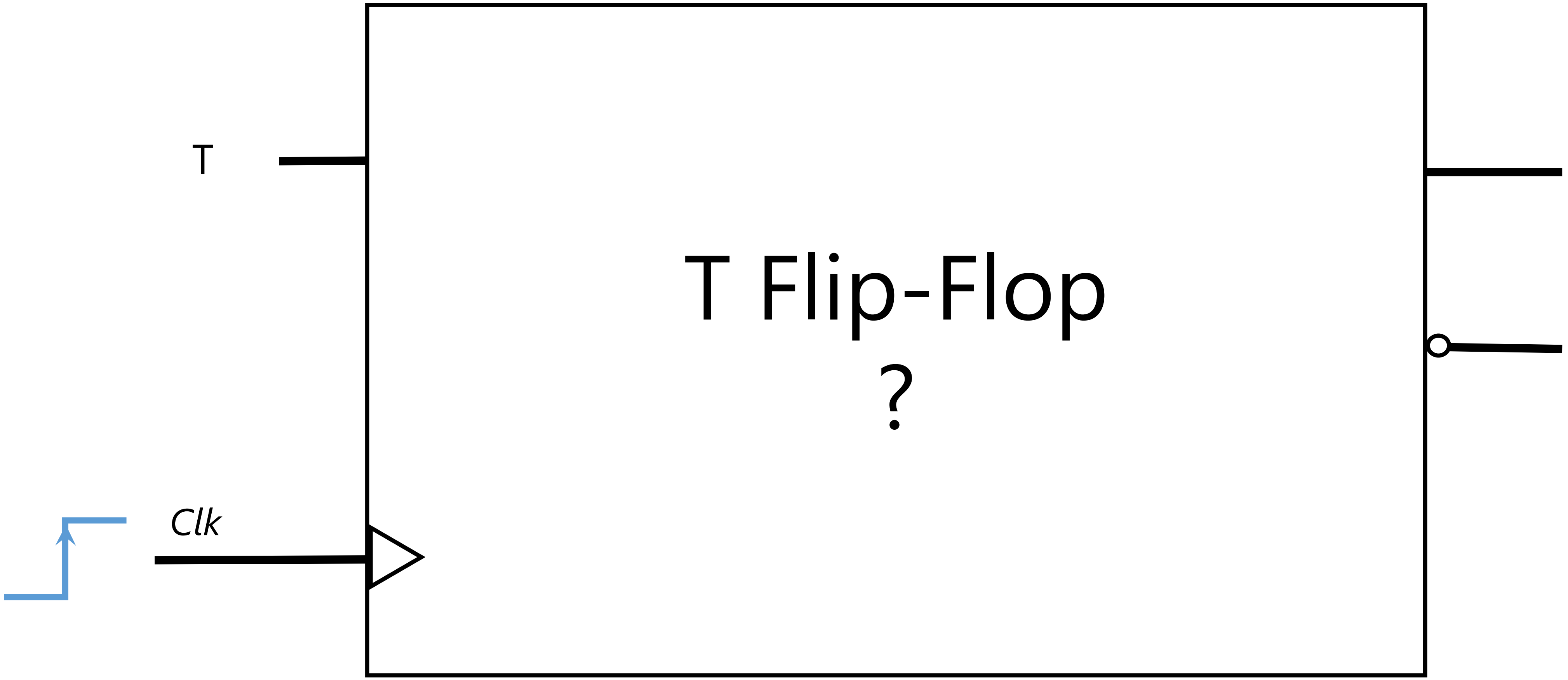
$X=1 \rightarrow Q=1$

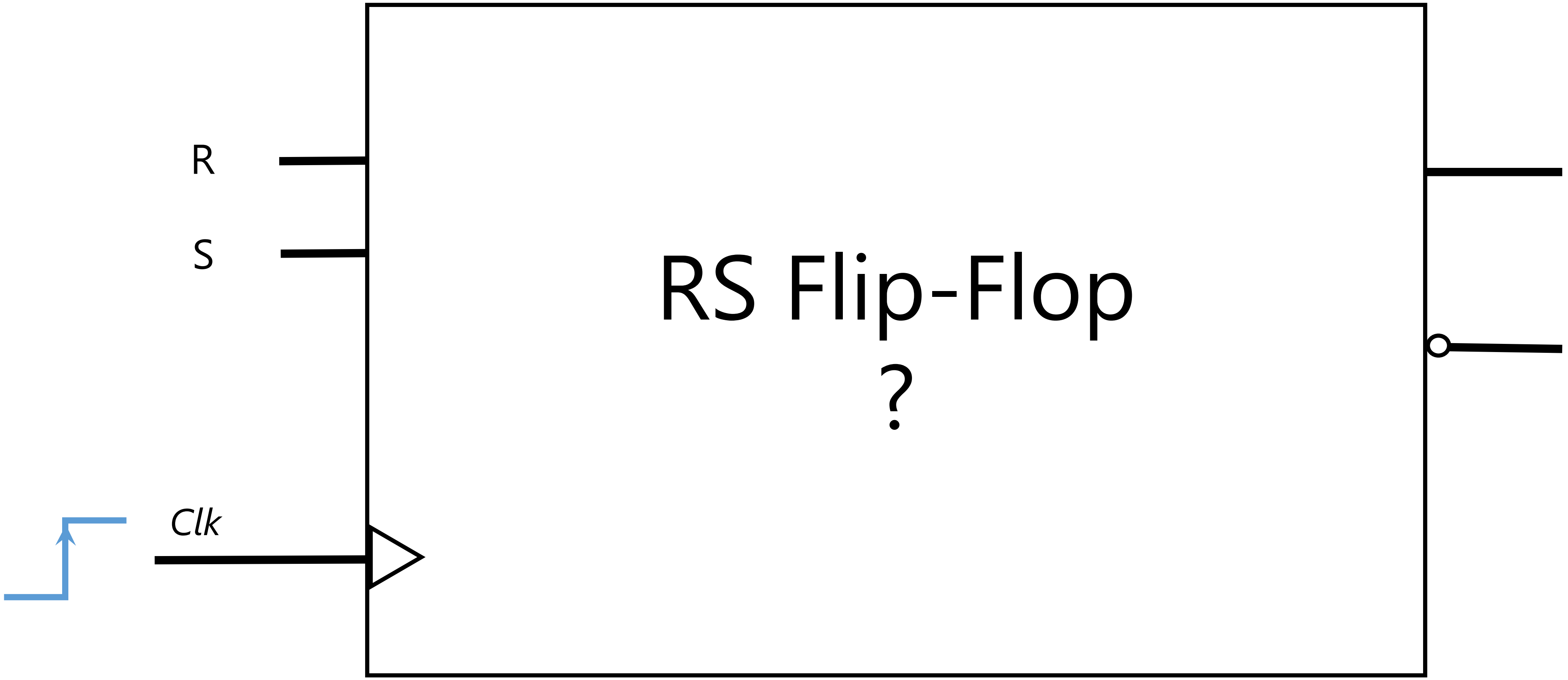


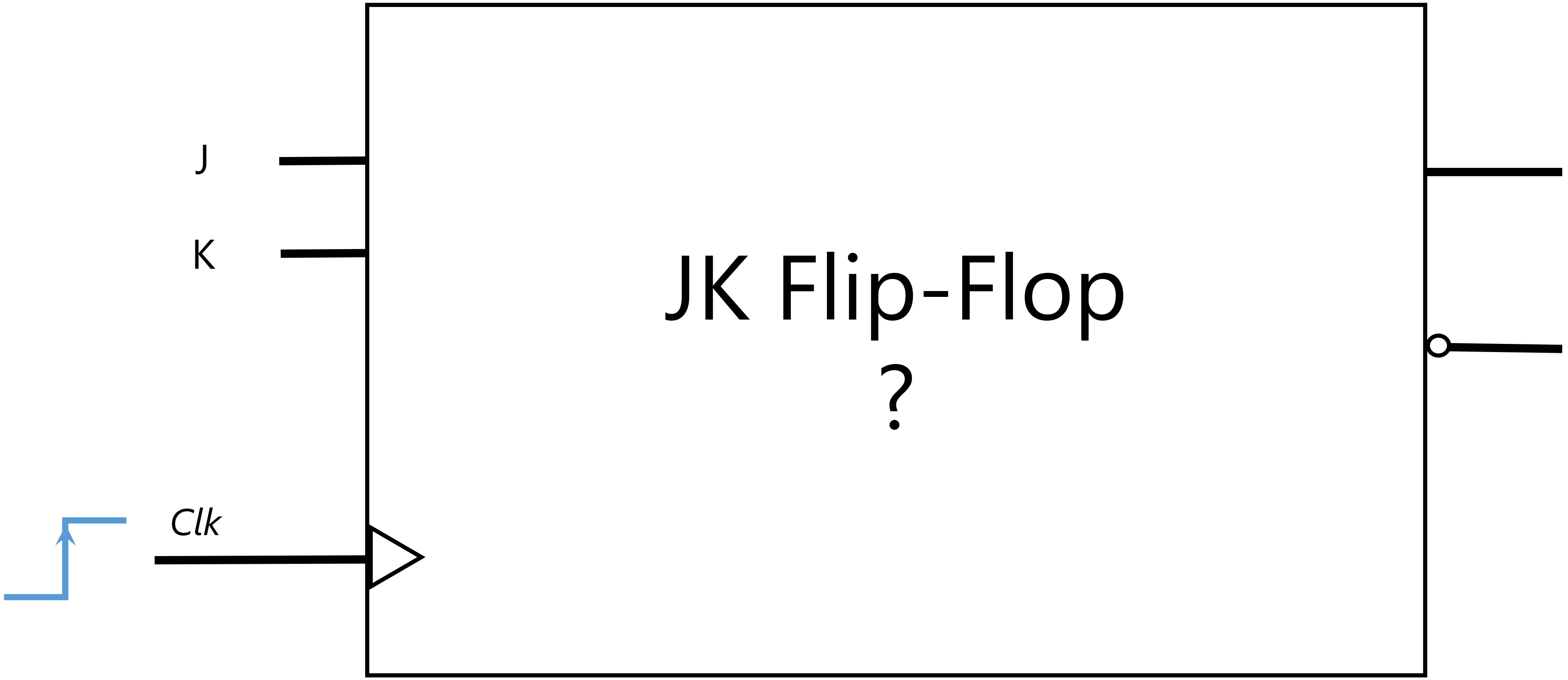

X
D
Master
E
D
Slave
E
Clk


D
D
Master
E
D
Slave
E
Clk


D
D Flip-Flop
Clk


T
Clk
T Flip-Flop
?


R
S
Clk
RS Flip-Flop
?


J
K
JK Flip-Flop
?
Clk

We have our ideal memory unit: Flip-Flop
Let's build larger memory units!

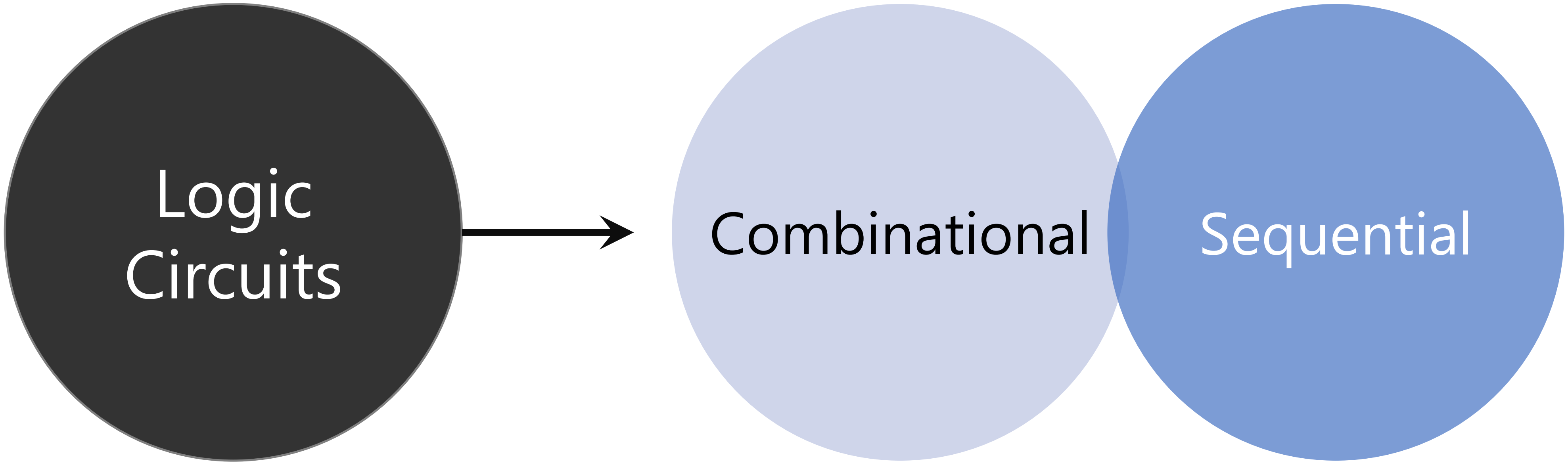
We have our ideal memory unit: Flip-Flop
Let’s build sequential (logic) circuits!

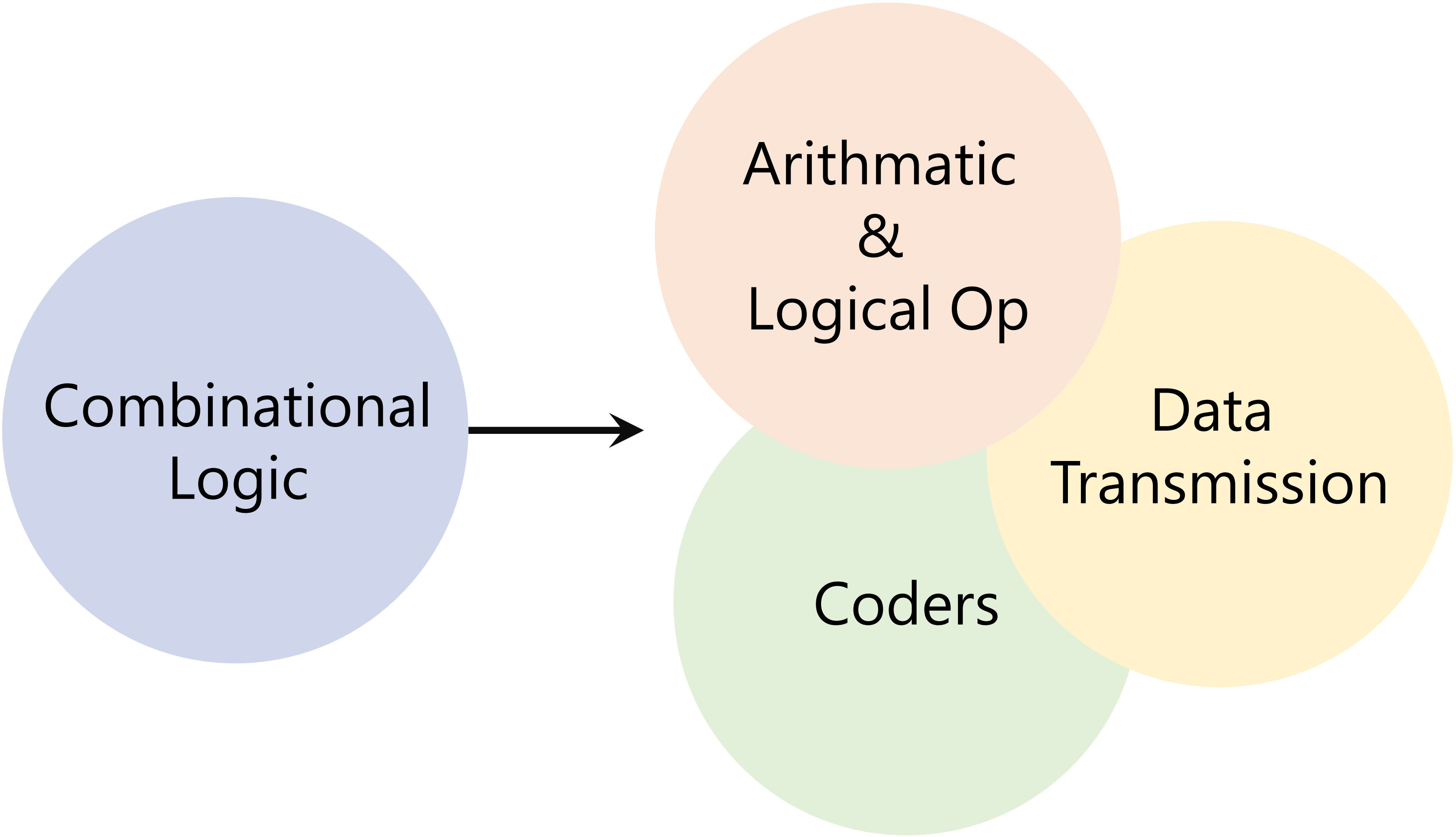
Analysis
vs.
Design

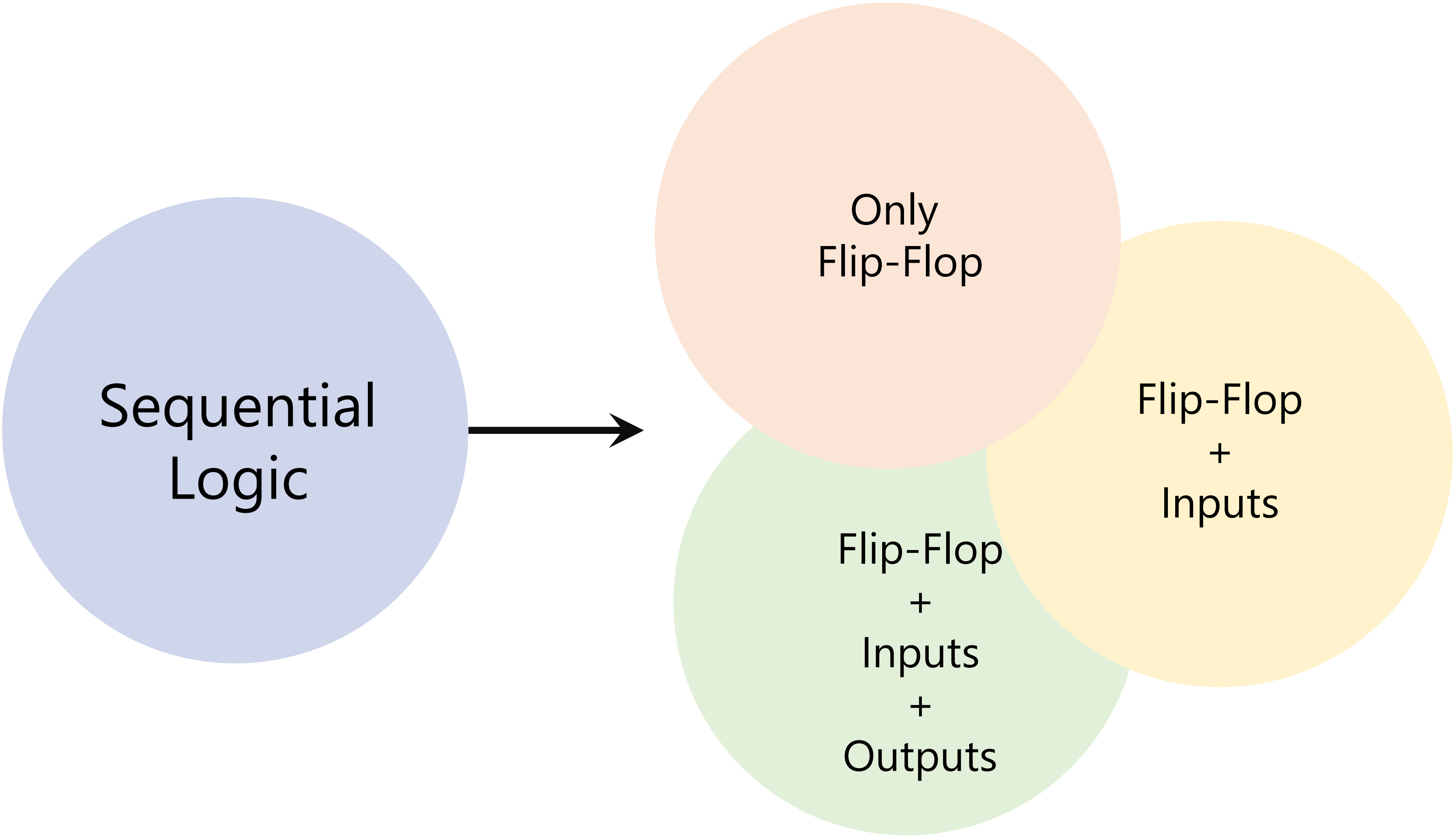
Analysis: Given a sequential circuit, show the behavior

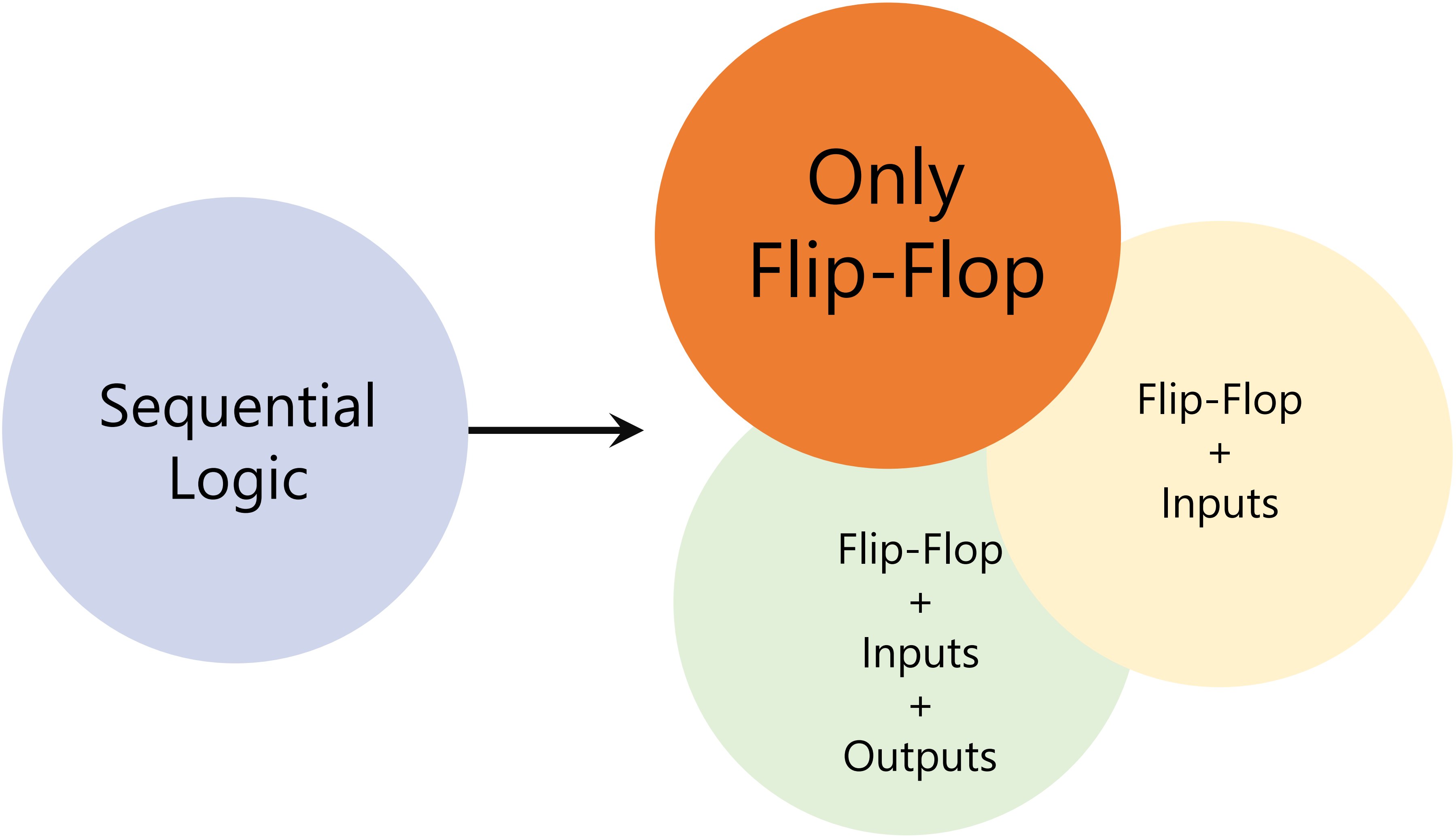
vs.

Design: Given a behavior, build the sequential circuit


Logic
Circuits
Combinational
Sequential

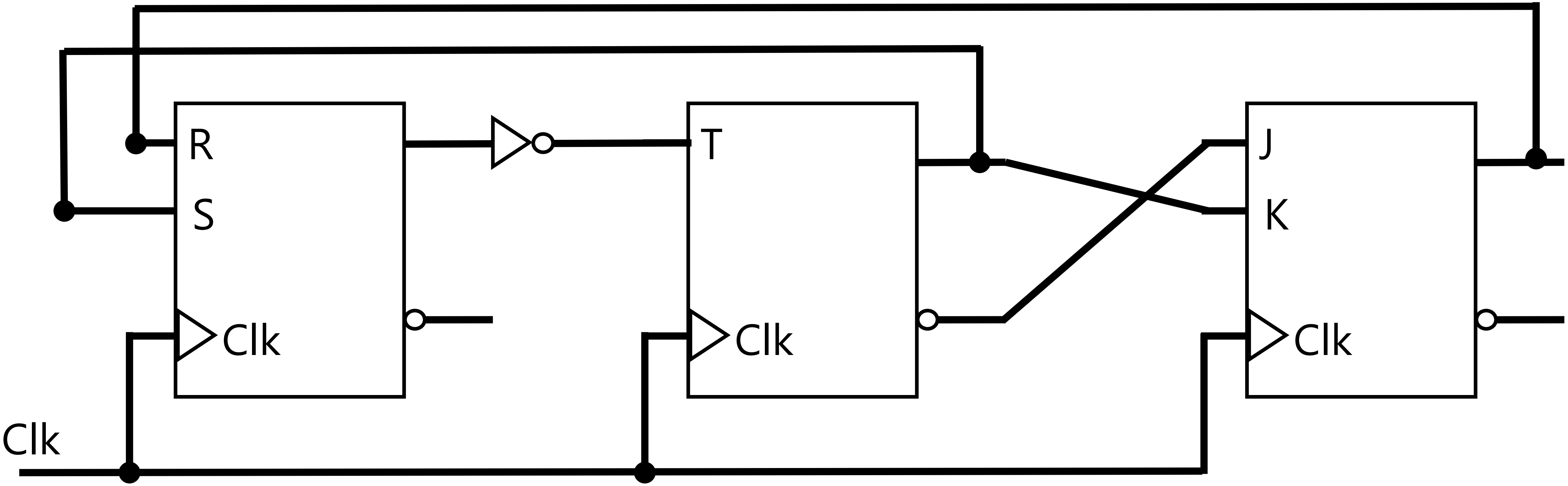

Combinational
Logic
Arithmatic
&
Logical Op
Data
Transmission
Coders


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


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

Analysis

by an example


R
S
Clk
T
Clk
J
K
Clk
Clk

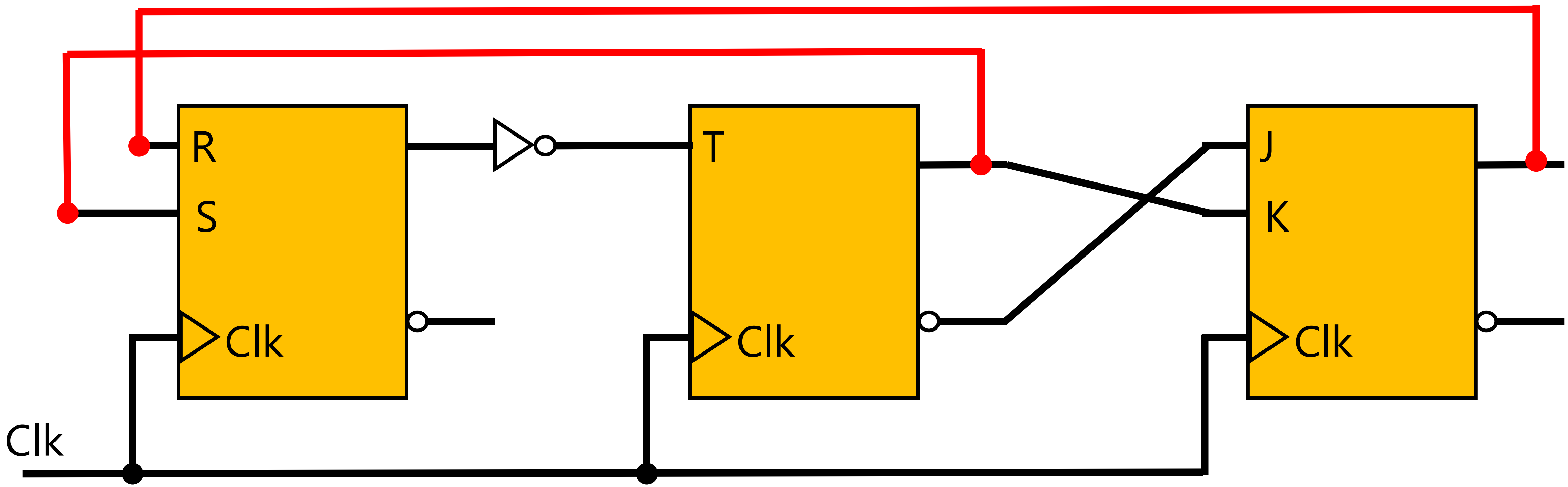
Analysis

0) Is it sequential circuit?

At least one FF → Yes

At least one feedback → Yes

Otherwise → No



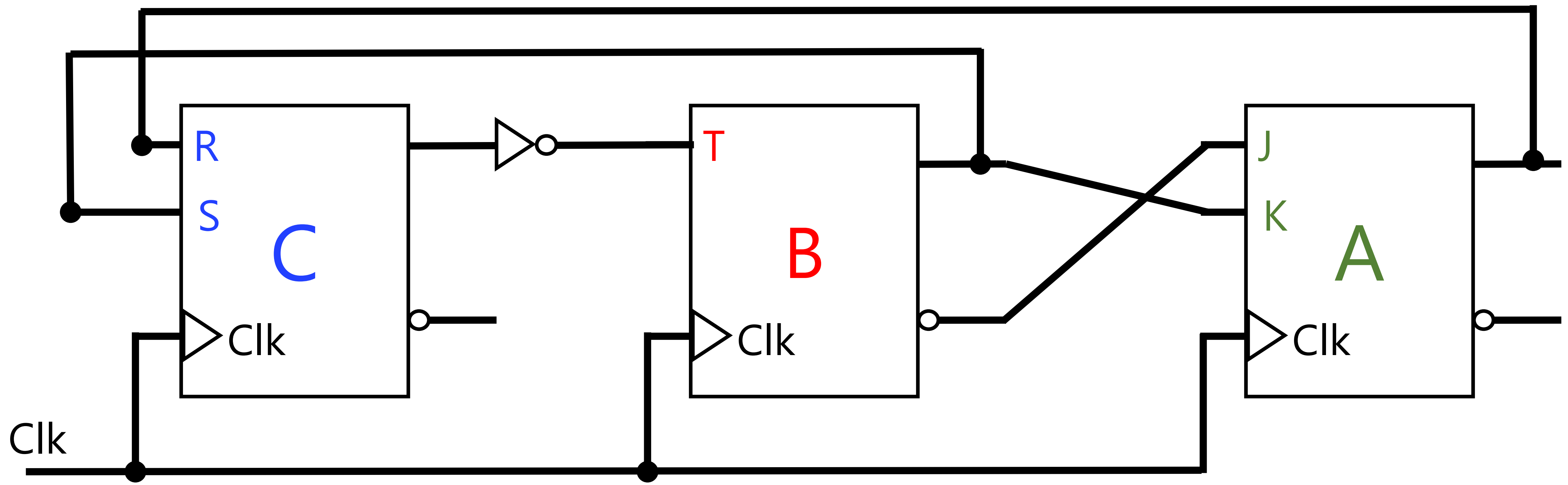
R
S
Clk
T
Clk
J
K
Clk
Clk

Analysis

1) What are the FFs?

1.1. We pick a name for each FF

1.2. We note the type of FF

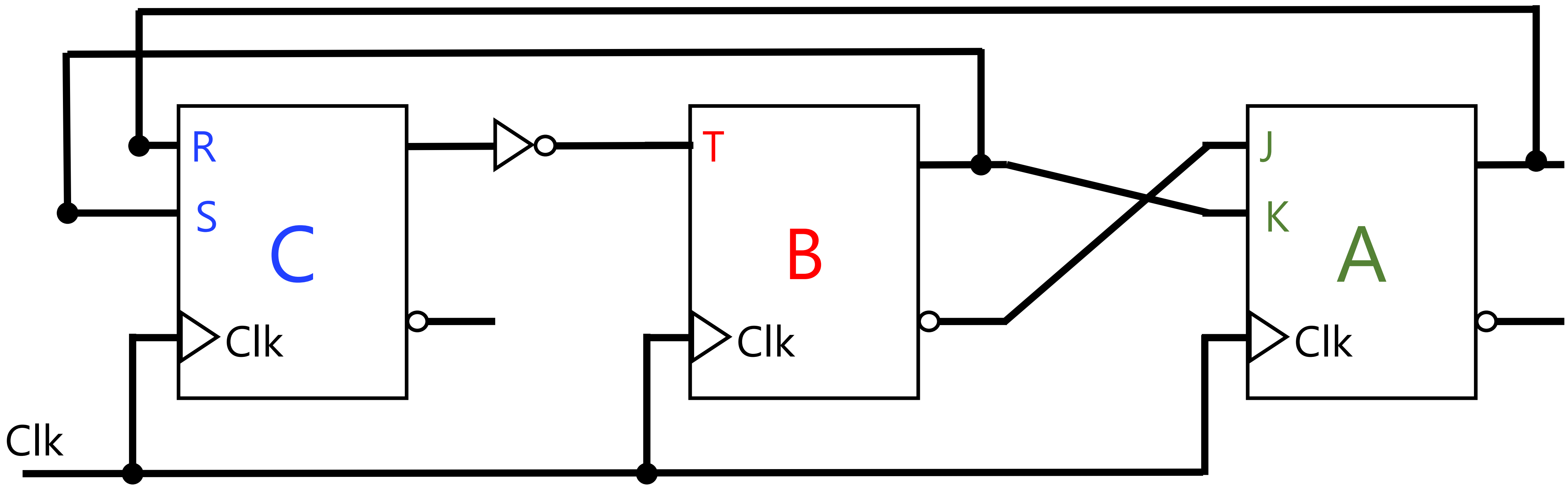

R
S
C
Clk
T
B
Clk
J
K
A
Clk
Clk

Analysis

2) What are the state combinations (possibilities)?

Each FF can have {0,1} states

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



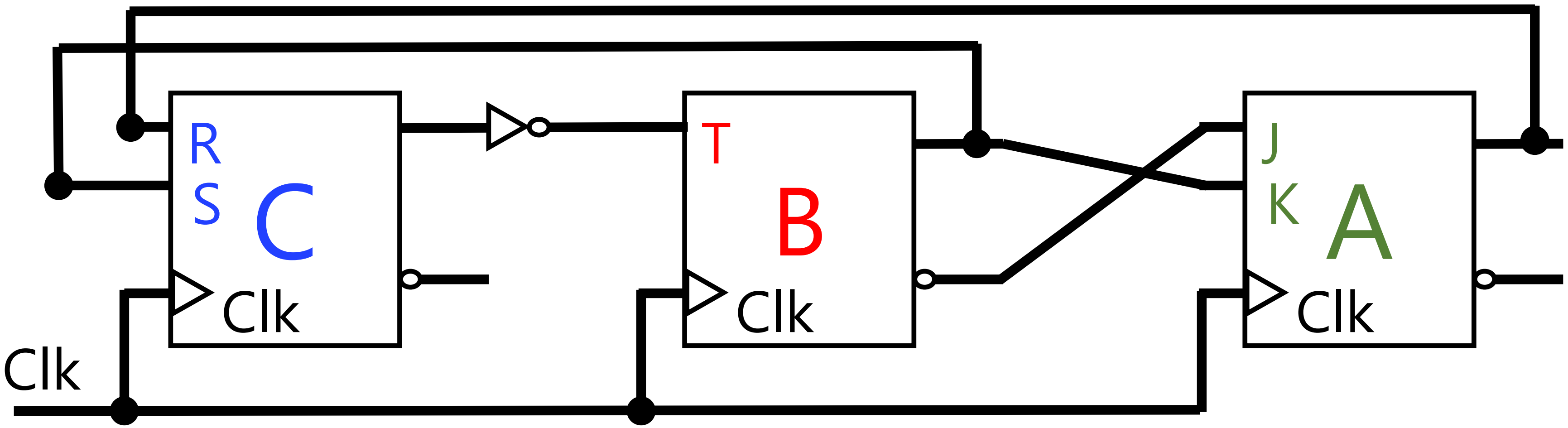
#FFs = 3 → 2^3 = 8 combinations

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



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

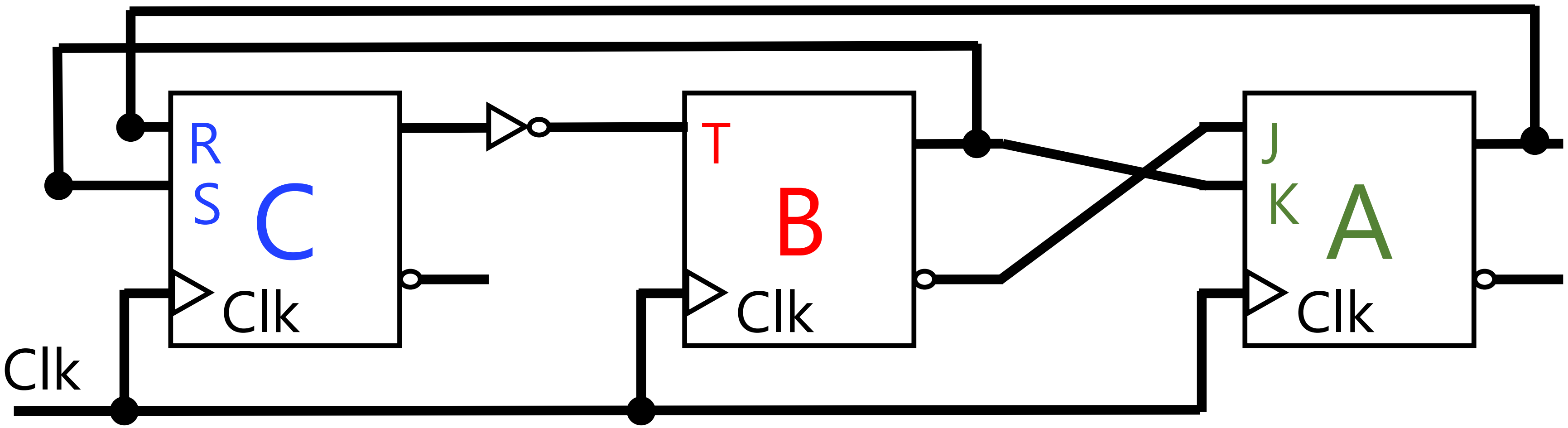
Analysis

3) Fill the ‘State’ table

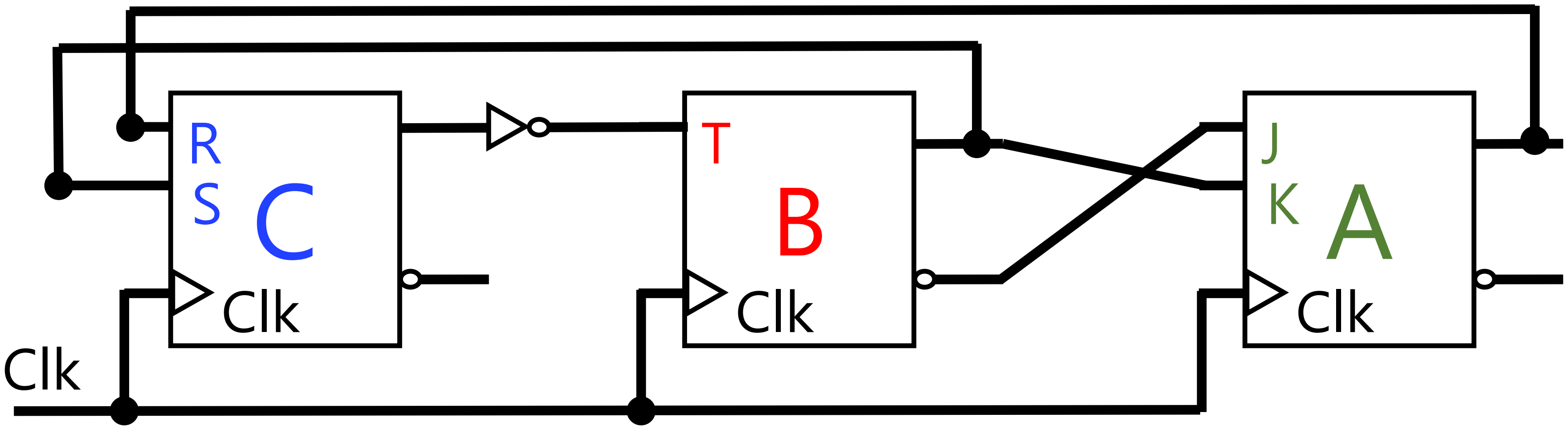
For each FF, we determine the next state based on

I) current state

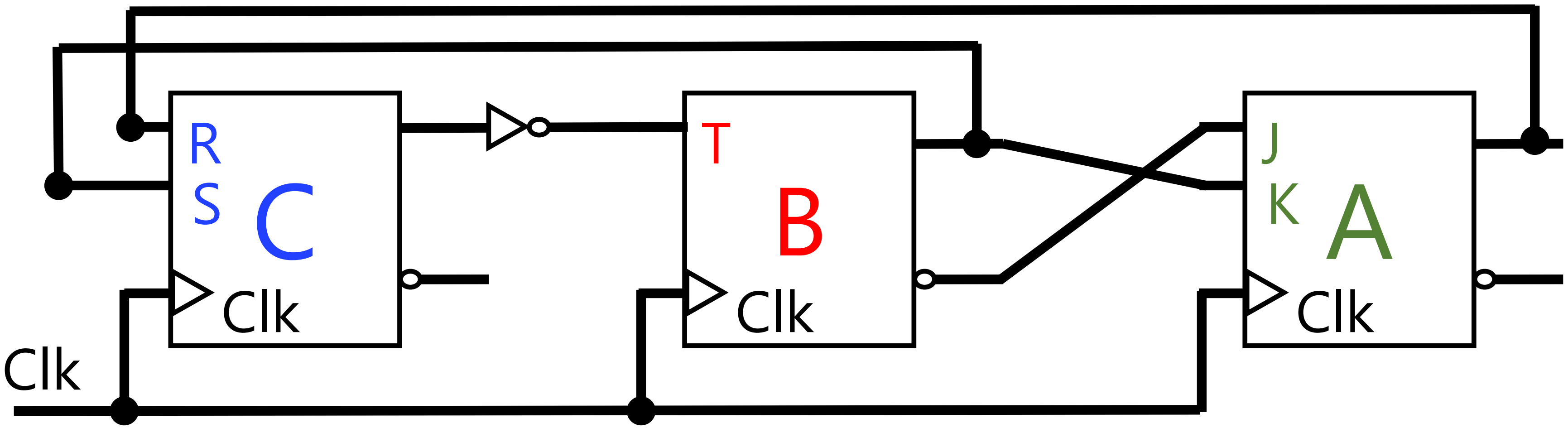
II) the current value of inputs to the FF



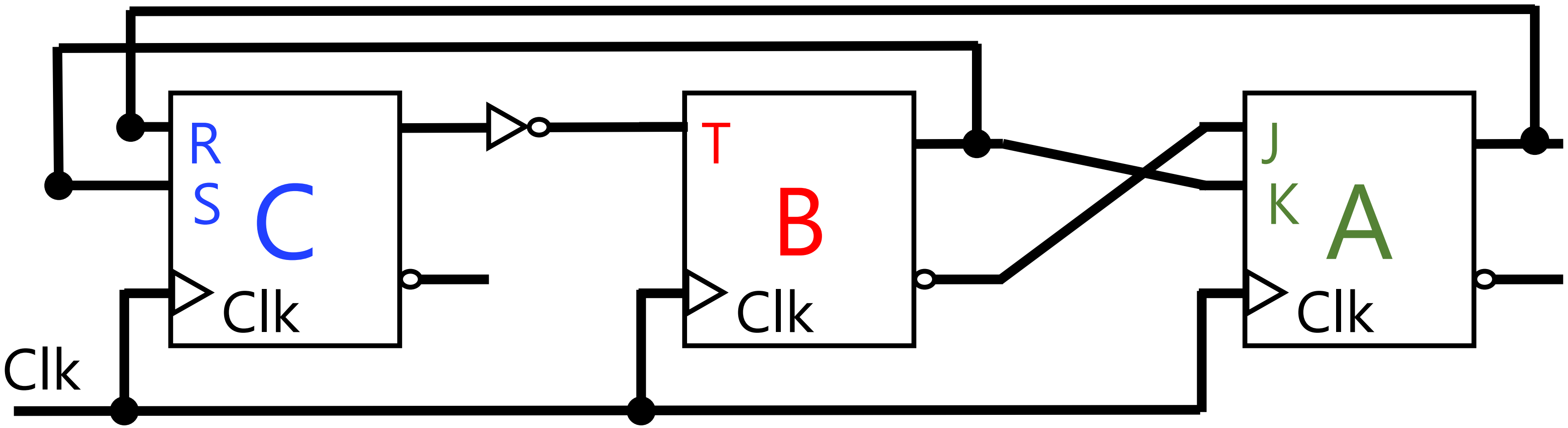
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			



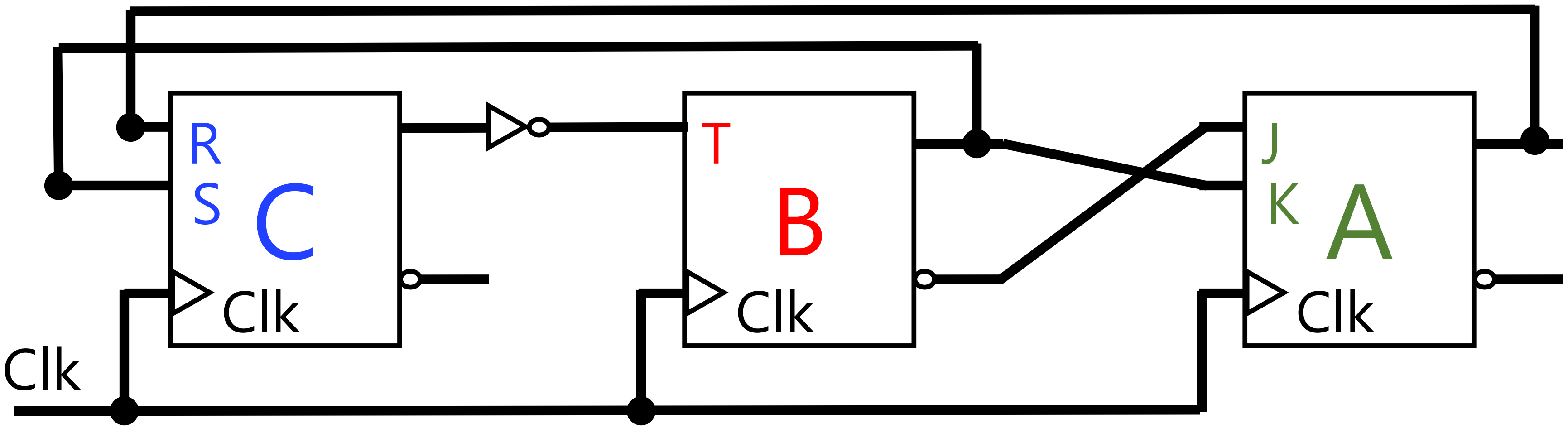
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0			?
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			



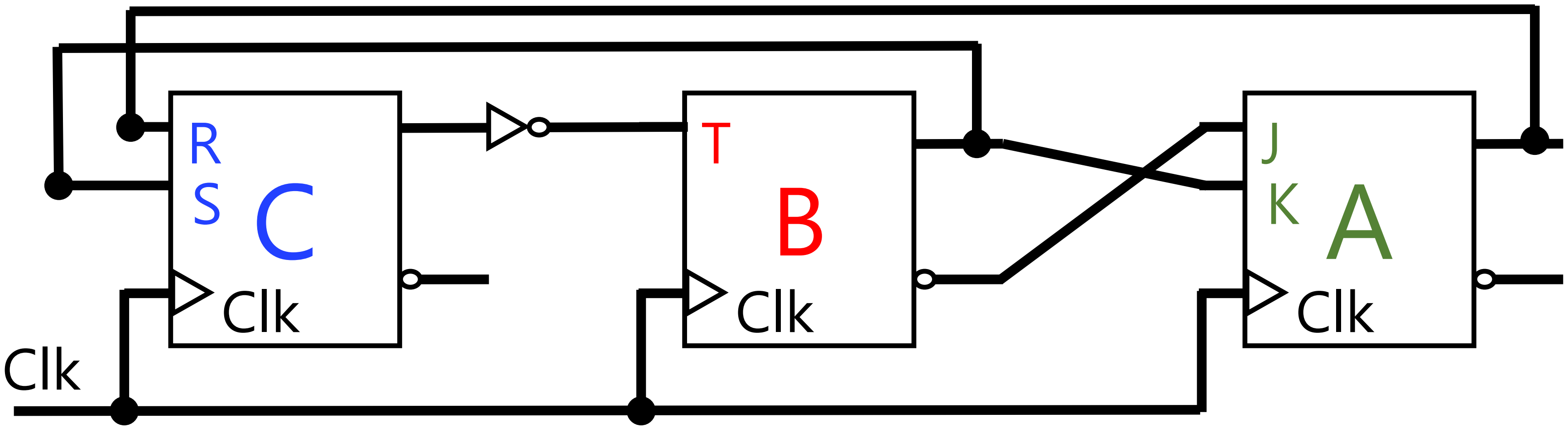
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0			$Q_A(T)=0$ $J_A=Q'_B(T)=1$ $K_A=Q_B(T)=0$
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			



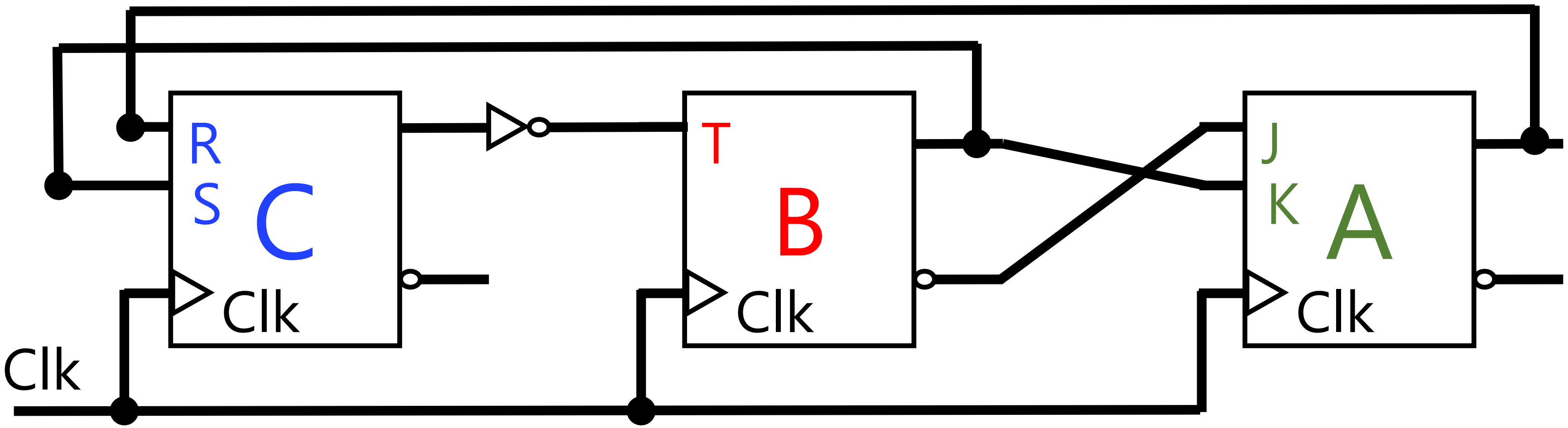
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0			$Q_A(T)=0$ $J_A=Q'_B(T)=1$ $K_A=Q_B(T)=0$ ----------------- Set Action: 1
0	0	1			
0	1	0			
0	1	1			
1	0	0			



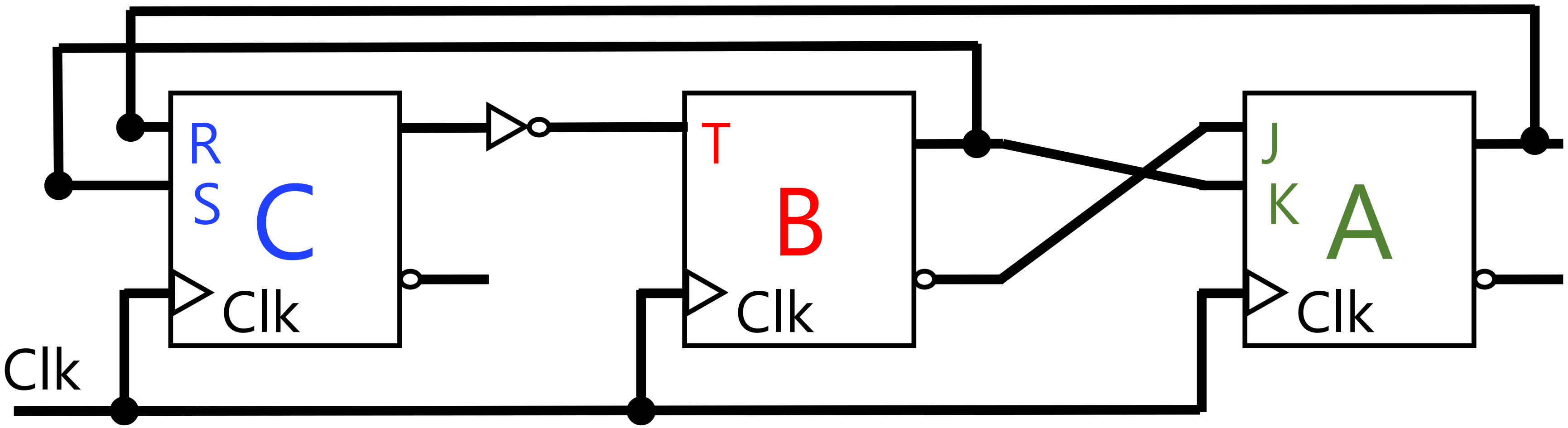
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0			1
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			



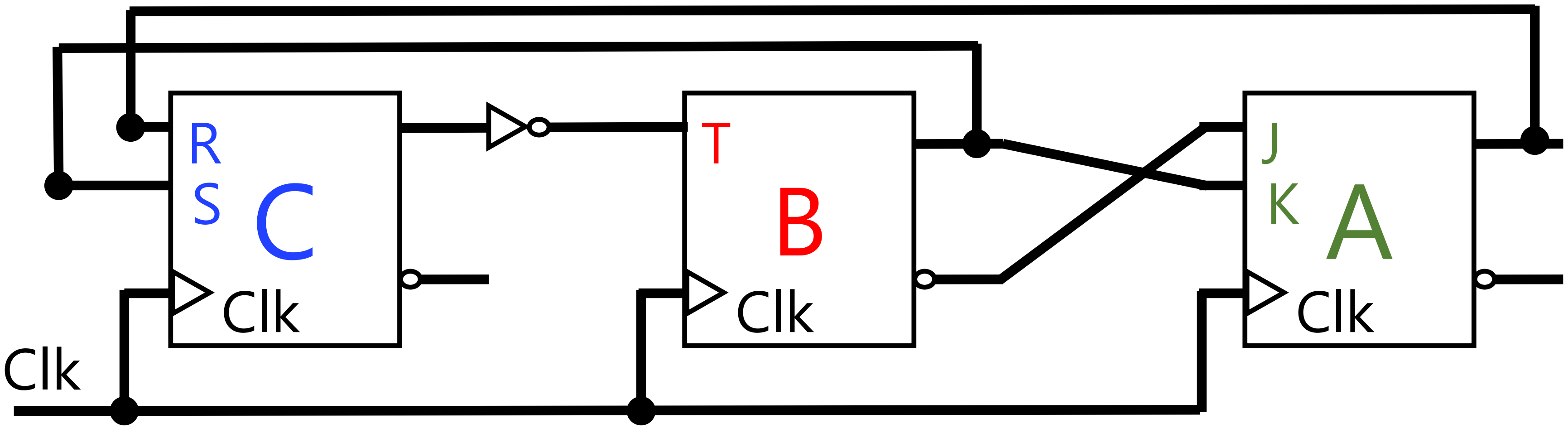
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0		?	1
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			



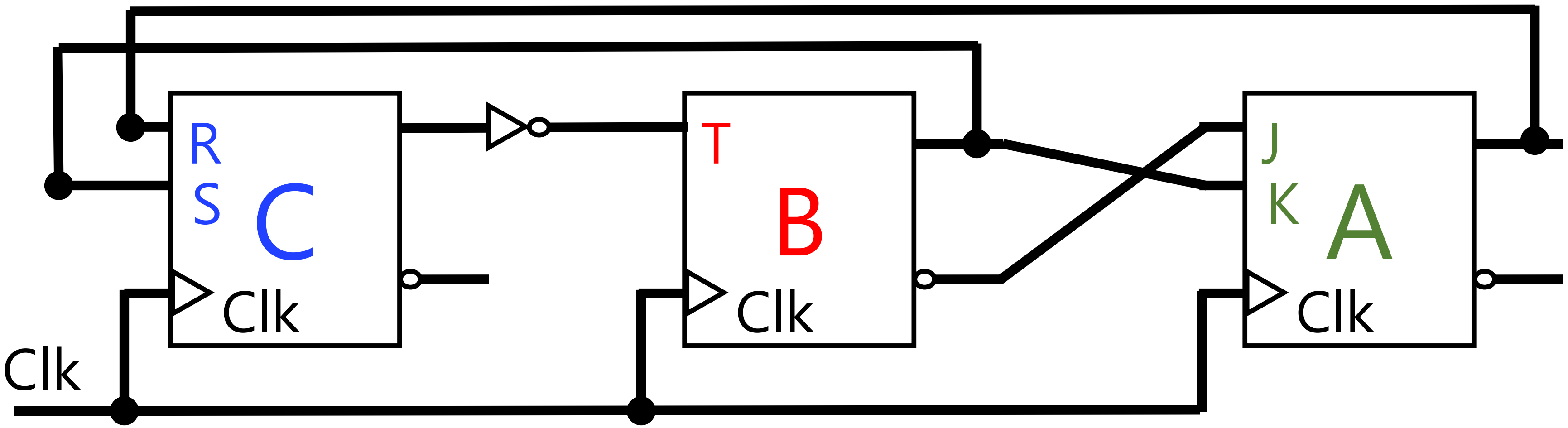
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0		$Q_B(T)=0$ $T_B=Q'_C(T)=1$	1
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			



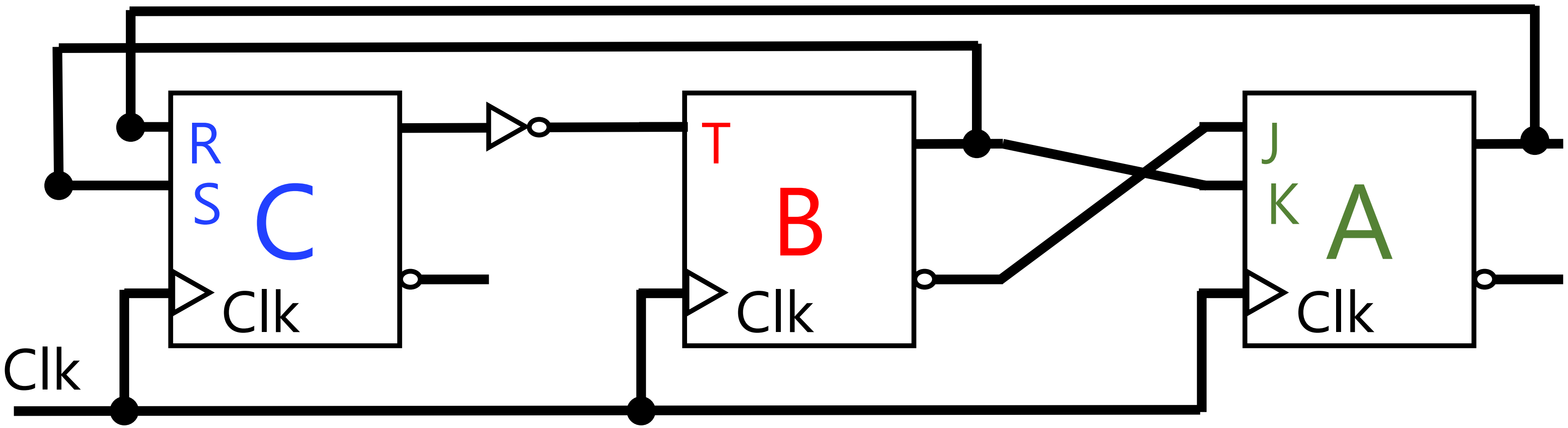
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0		$Q_B(T)=0$ $T_B=Q'_C(T)=1$ -------------- Comp. $(Q_B(T)) = 1$	1
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			



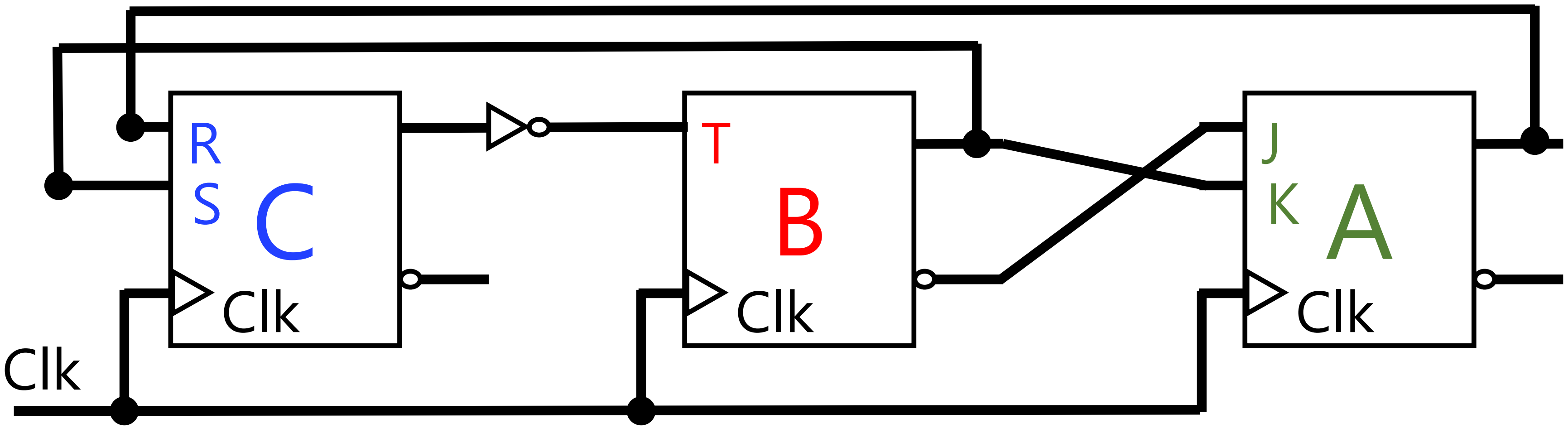
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	?	1	1
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			



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



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			

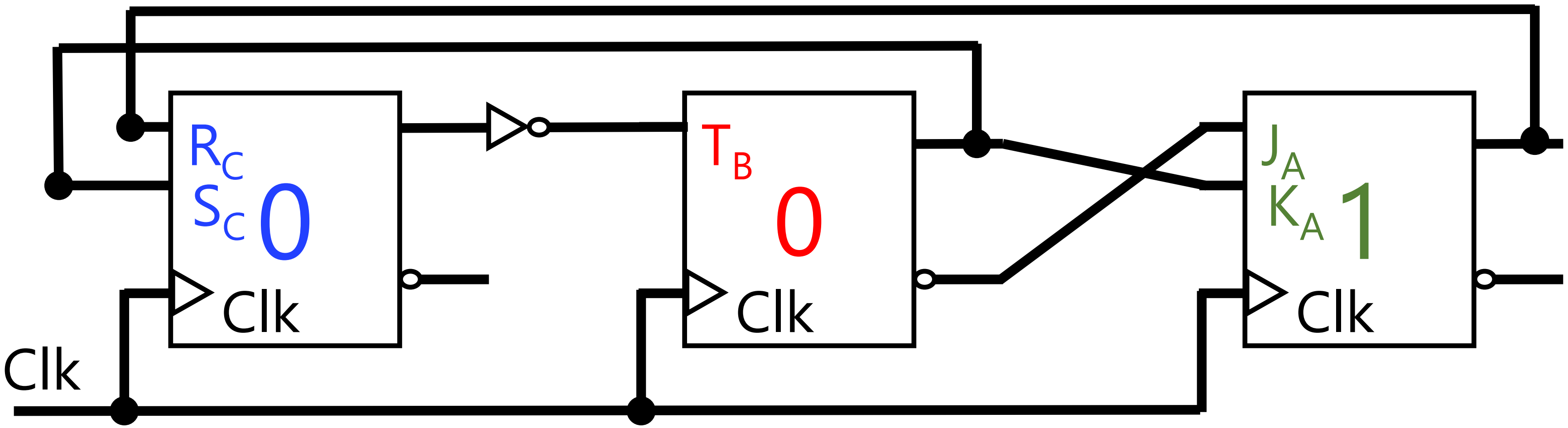


Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	1	1
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

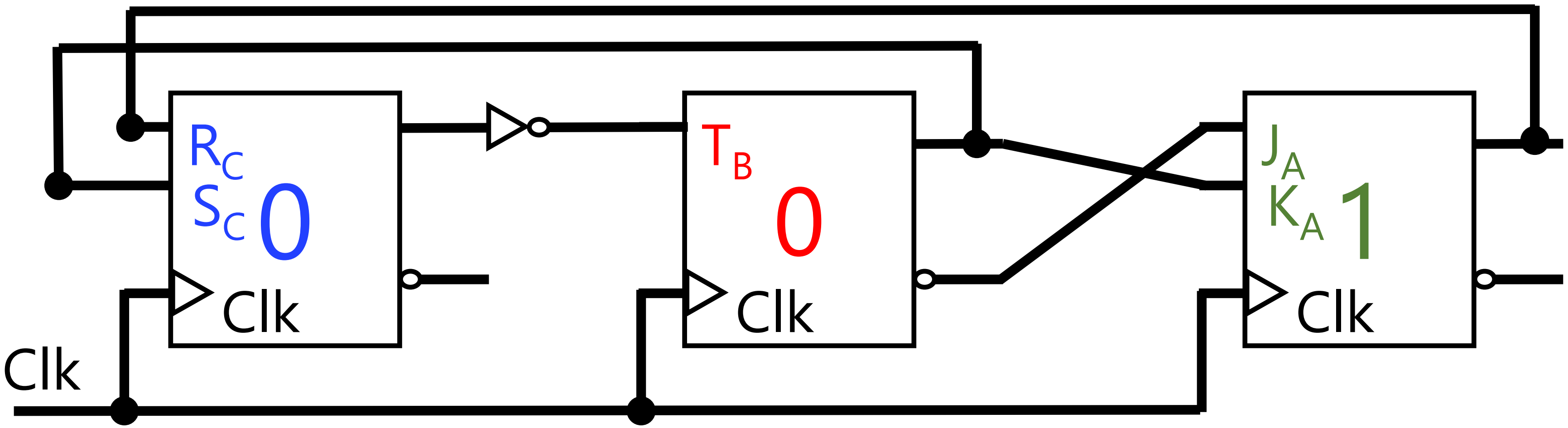
Analysis

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

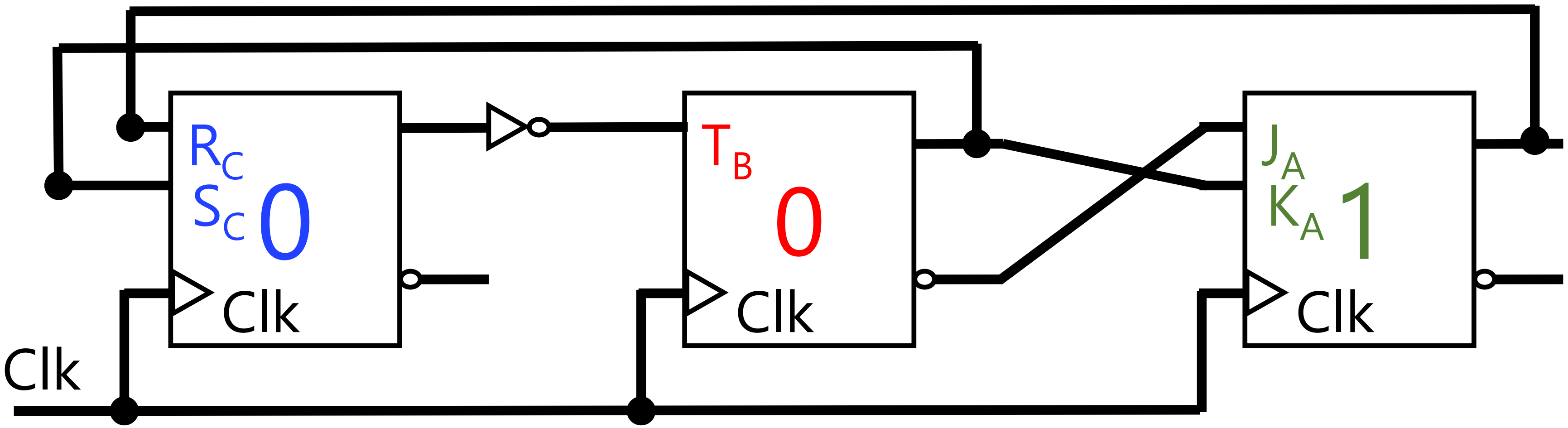
For simplicity, the current status of a FF can be assume to be as a binary variable



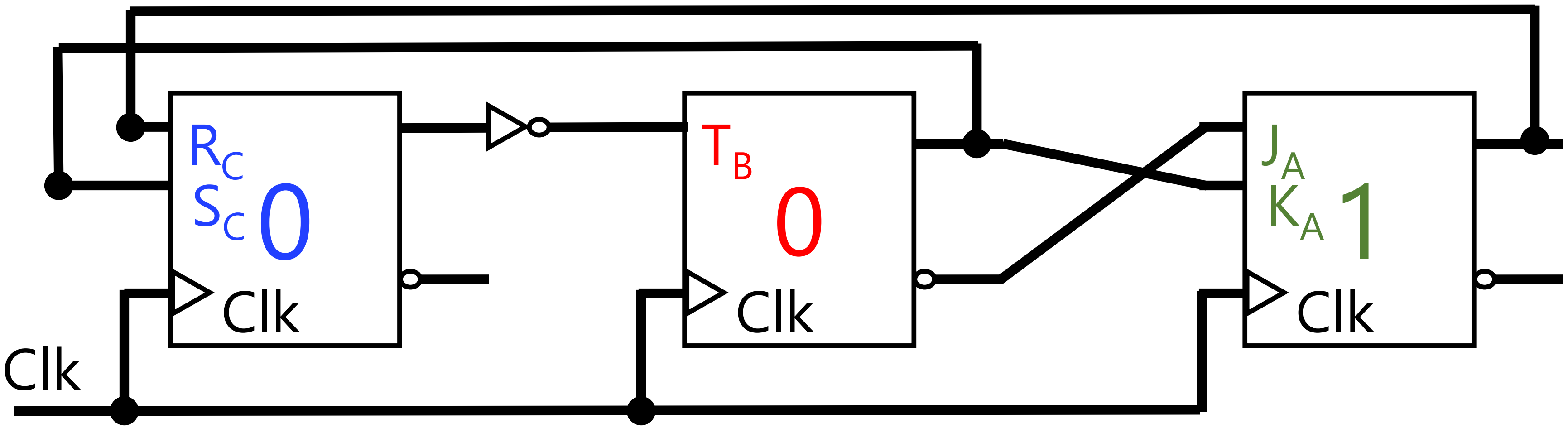
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	1	1
0	0	1			A=1 J_A=B′=1 K_A=B=0 --------------- Set Action: 1
0	1	0			
0	1	1			
1	0	0			



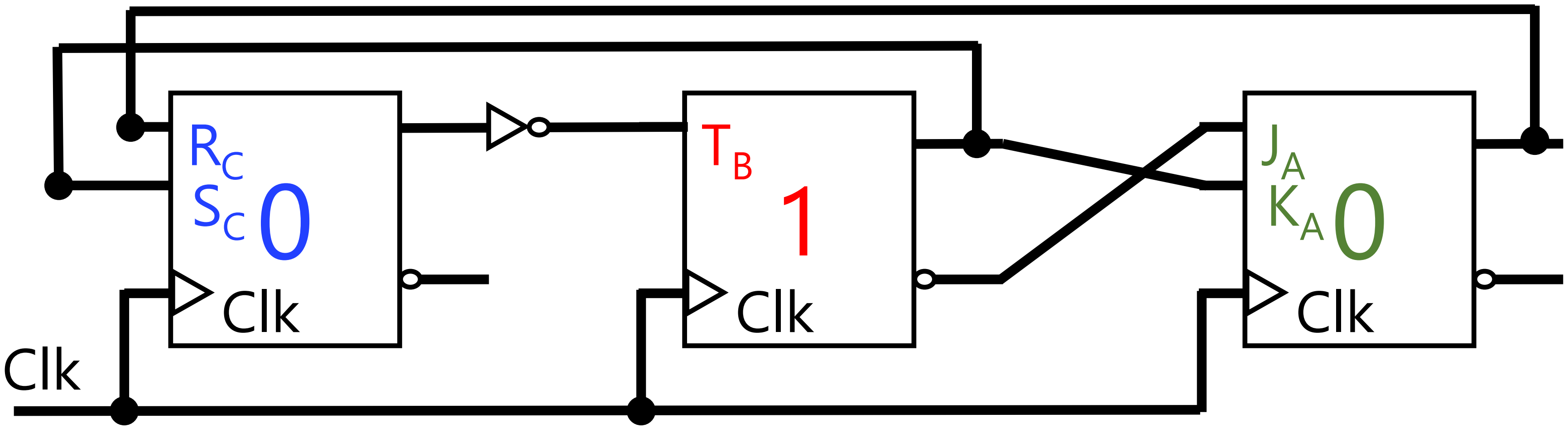
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	1	1
0	0	1		B=0 T_B=C′=1 ------------- Comp. Action: 1	1
0	1	0			
0	1	1			
1	0	0			
1	0	1			



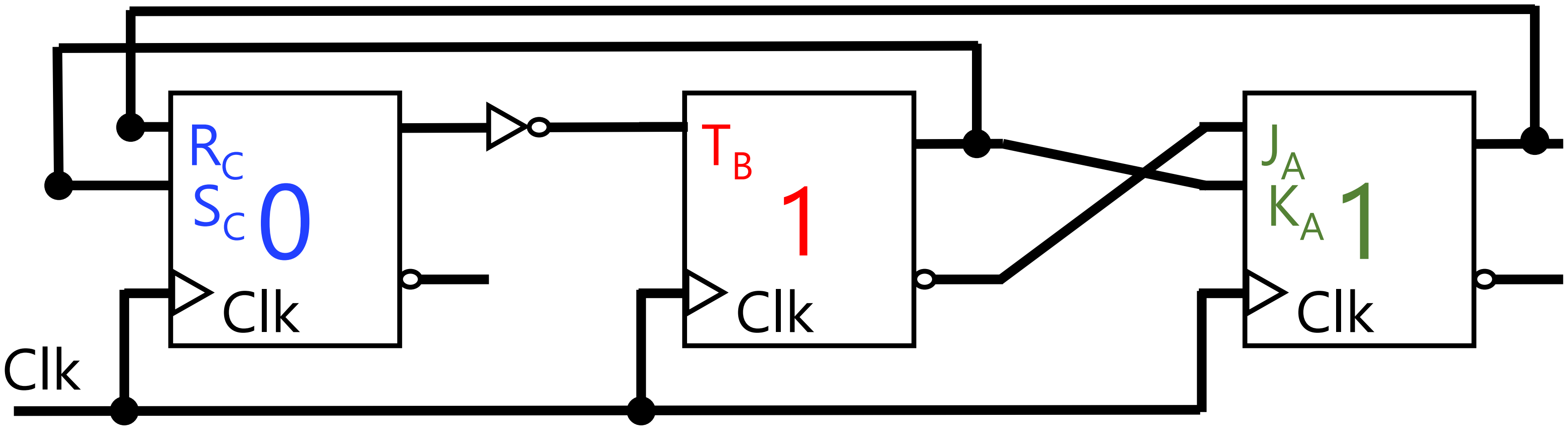
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	1	1
0	0	1	C=0 R_C=A=1 S_C=B=0 --------------- Reset Action: 0	1	1
0	1	0			
0	1	1			
1	0	0			



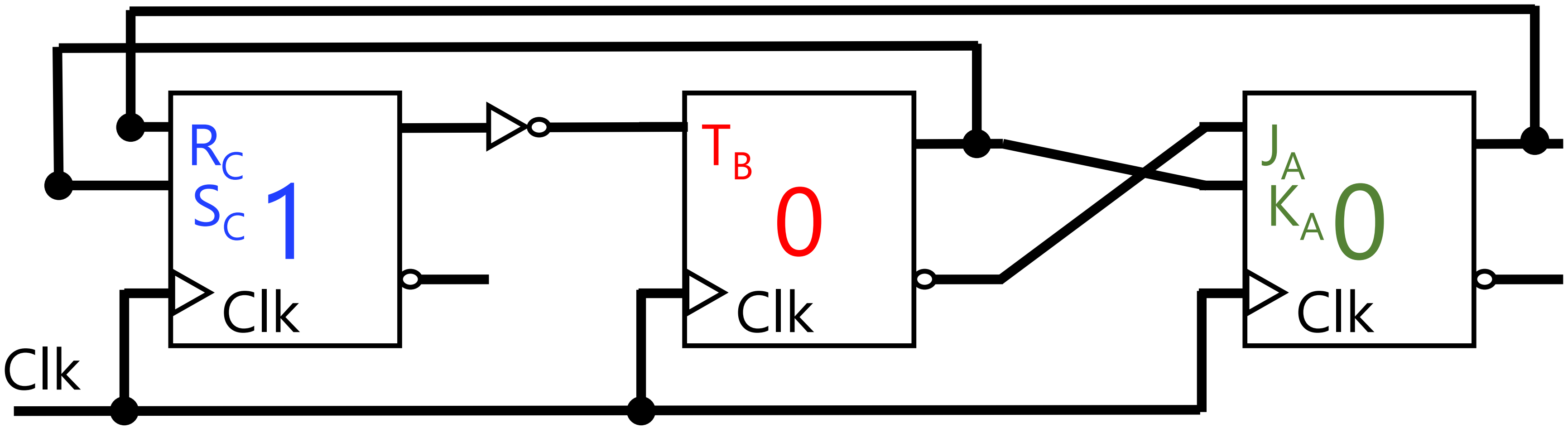
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			



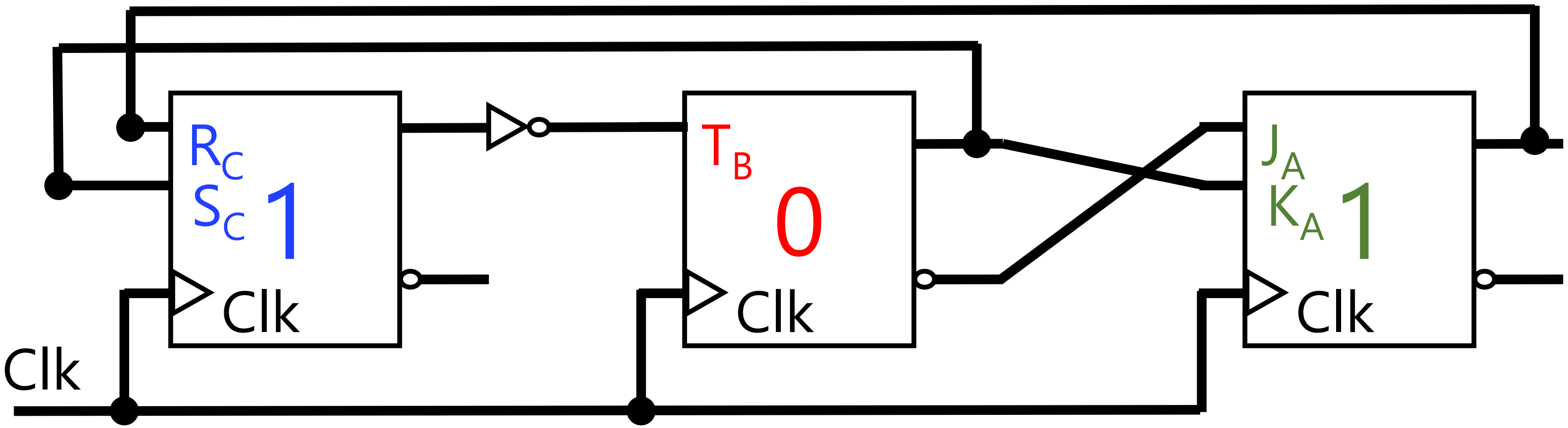
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			



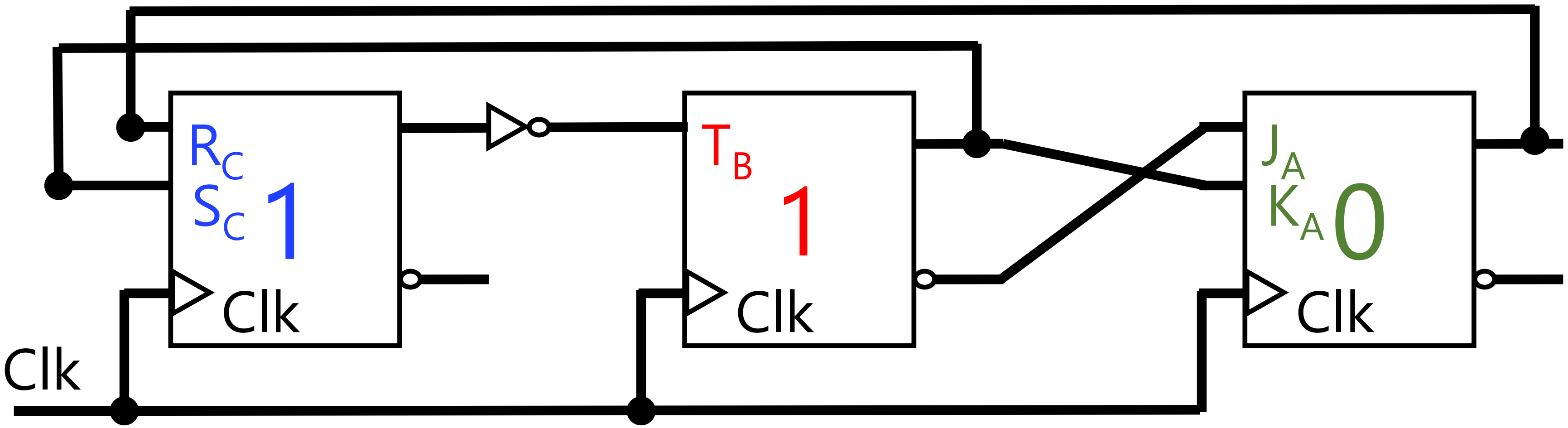
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	x	0	0
1	0	0			
1	0	1			
1	1	0			
1	1	1			



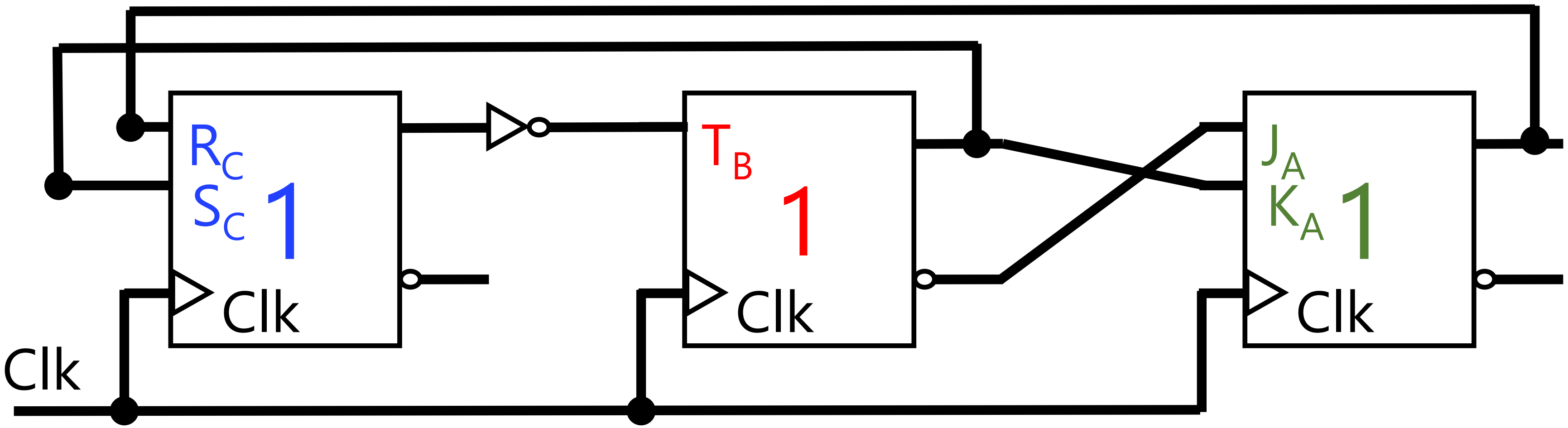
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	x	0	0
1	0	0	1	0	1
1	0	1			
1	1	0			
1	1	1			



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			
1	1	1			



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			



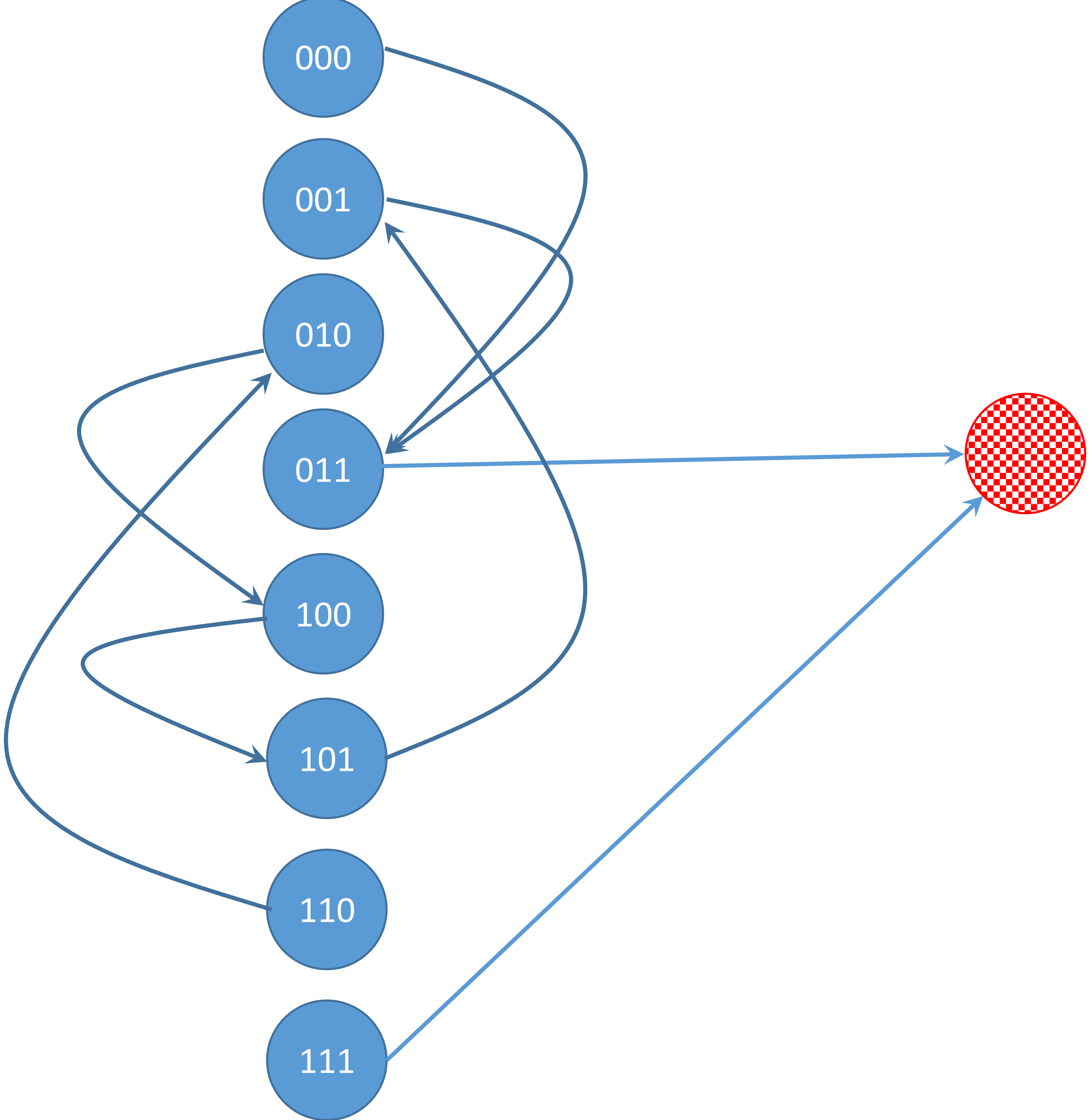
Q(T)			Q(T+1)		
C	B	A	C	B	A
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	x	0	0
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	0	1	0
1	1	1	x	1	0

Analysis

4) State Transition Diagram

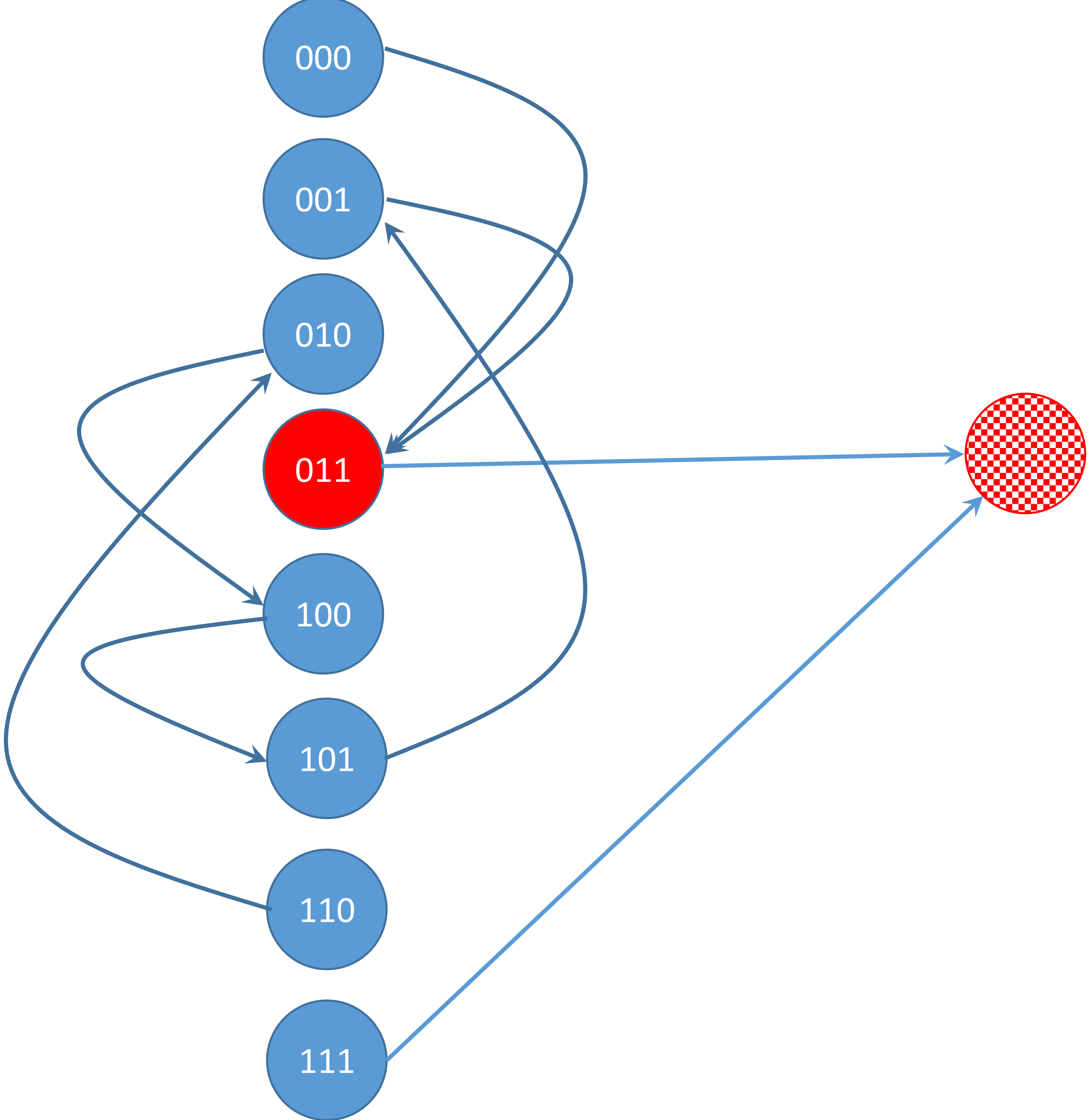
4.1. for each state combination (each row), a node

4.2. from one state (node) to another state, a directed edge


000
001
010
011
100
101
110
111

Analysis

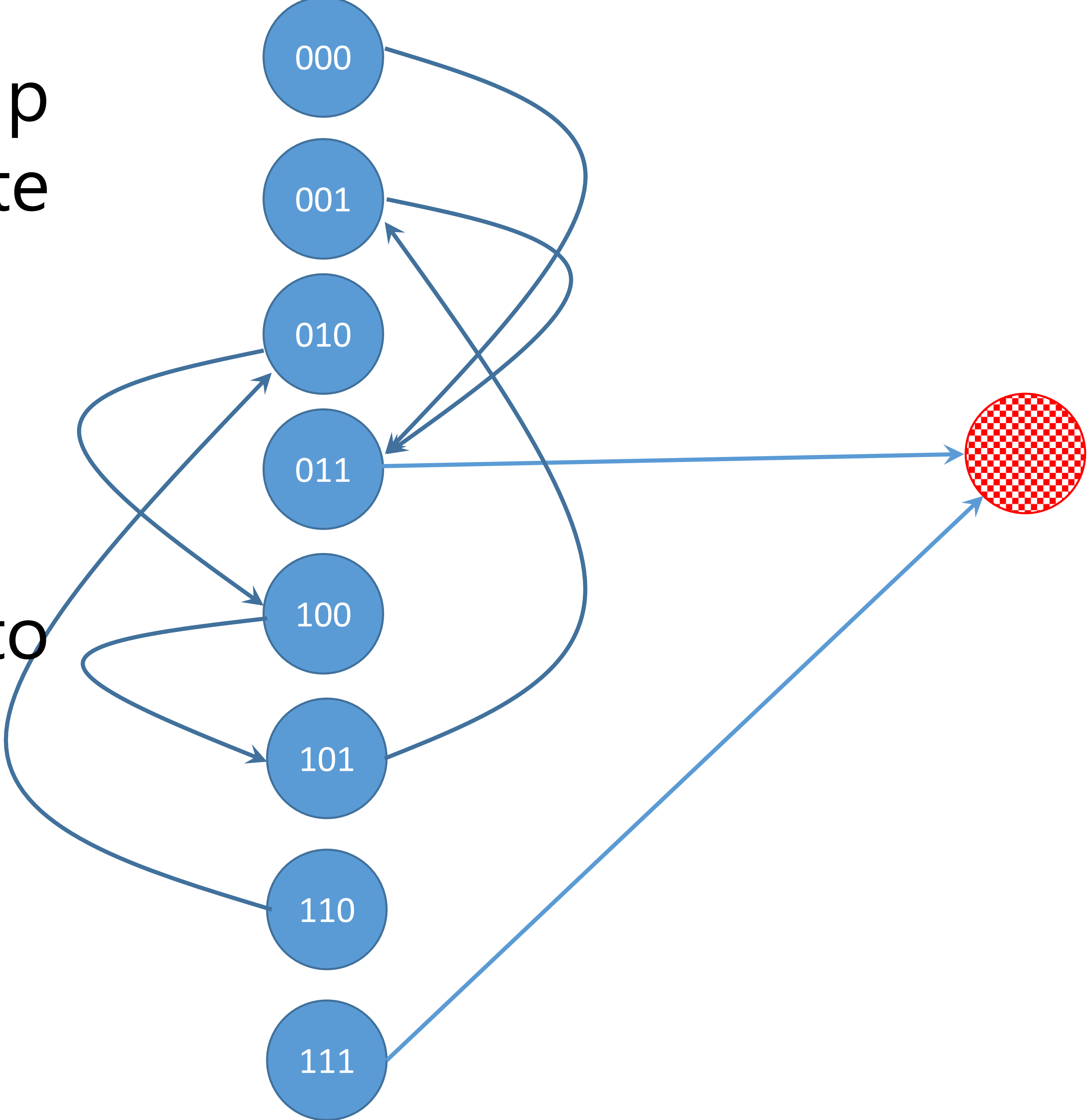
5) (*Optional*) Path on State Transitions

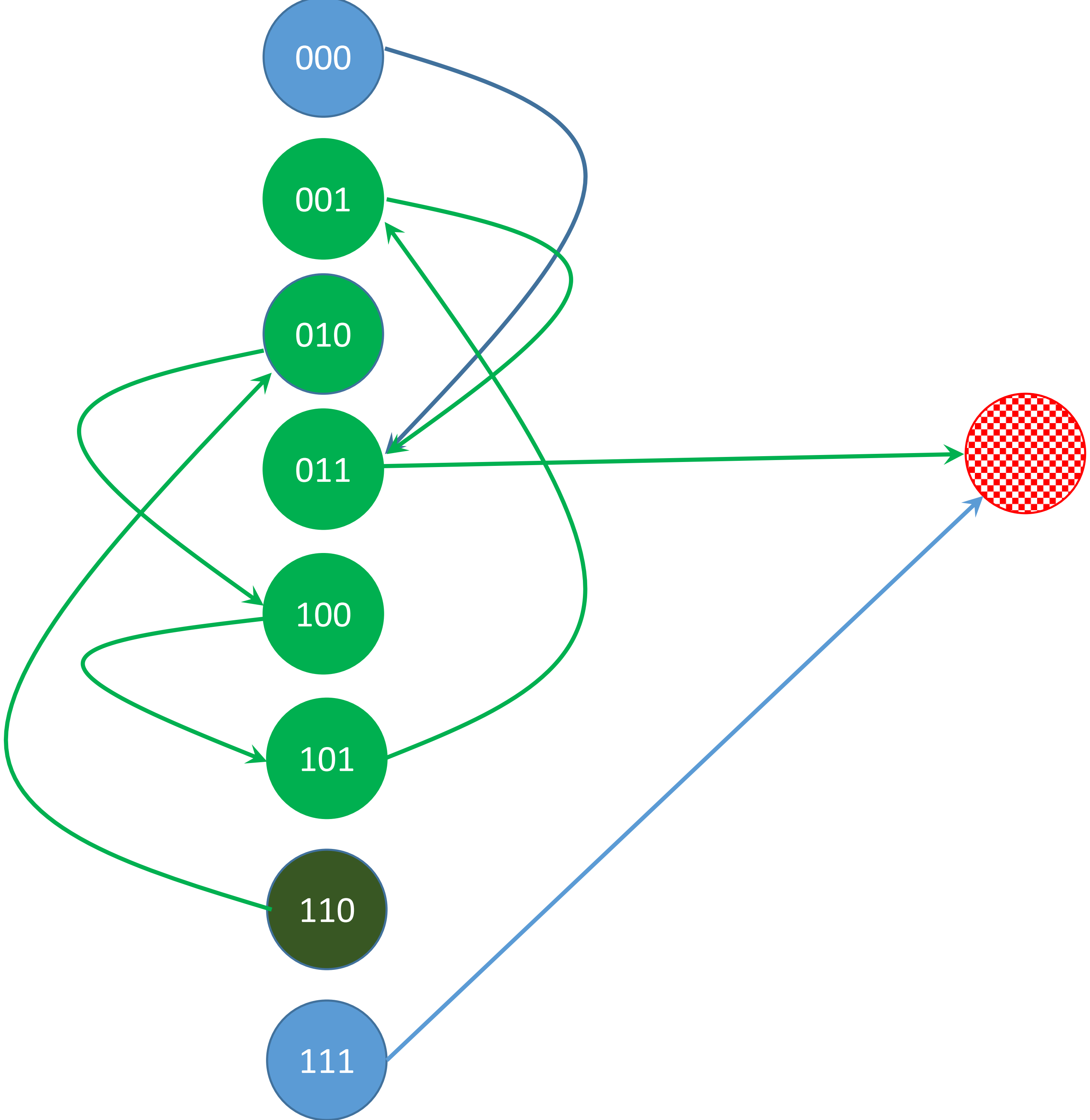

000
001
010
011
100
101
110
111

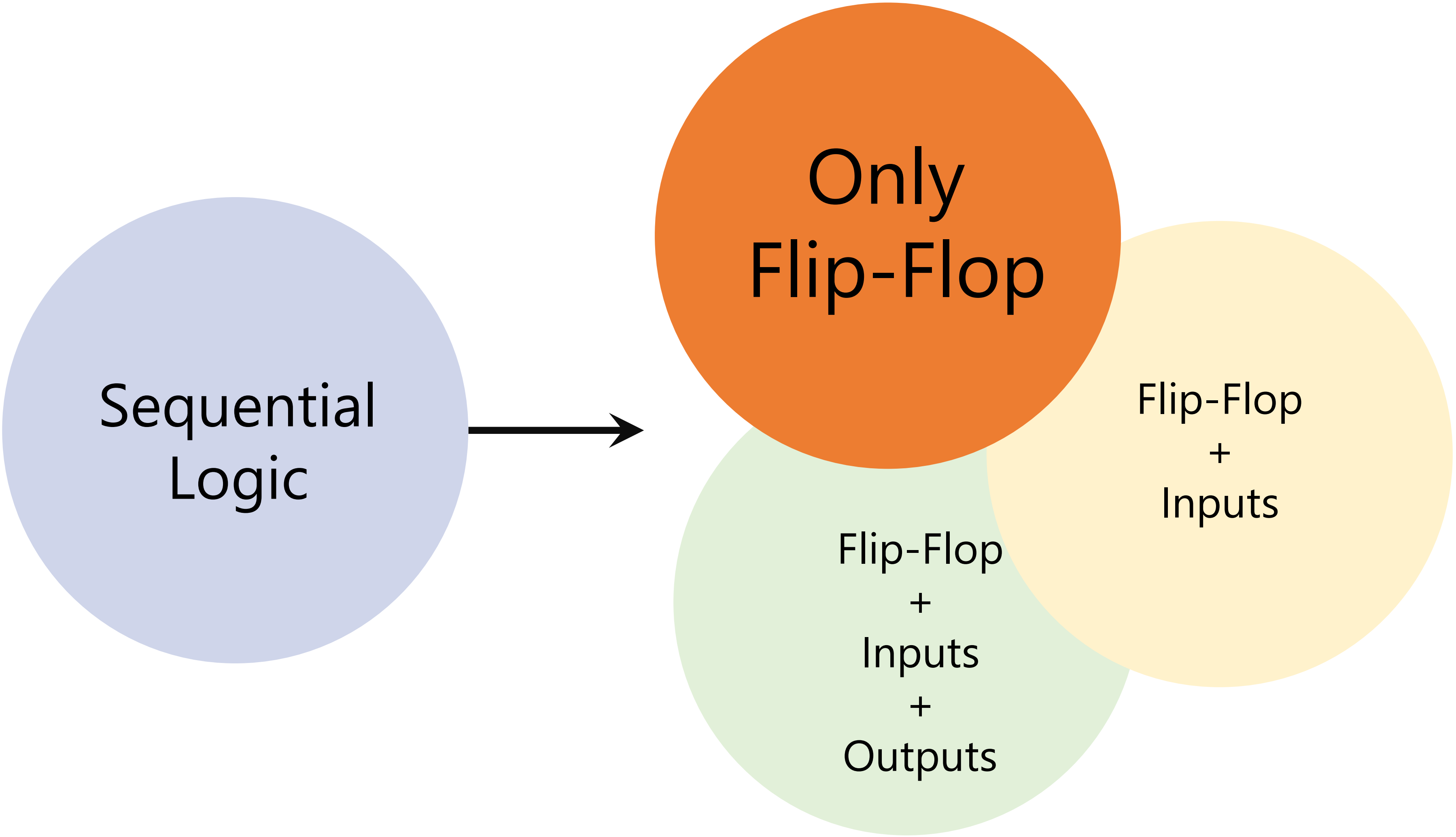
All the paths end up with indeterminate state

→

The circuit needs to be improved!




000
001
010
011
100
101
110
111


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

Design
by an example