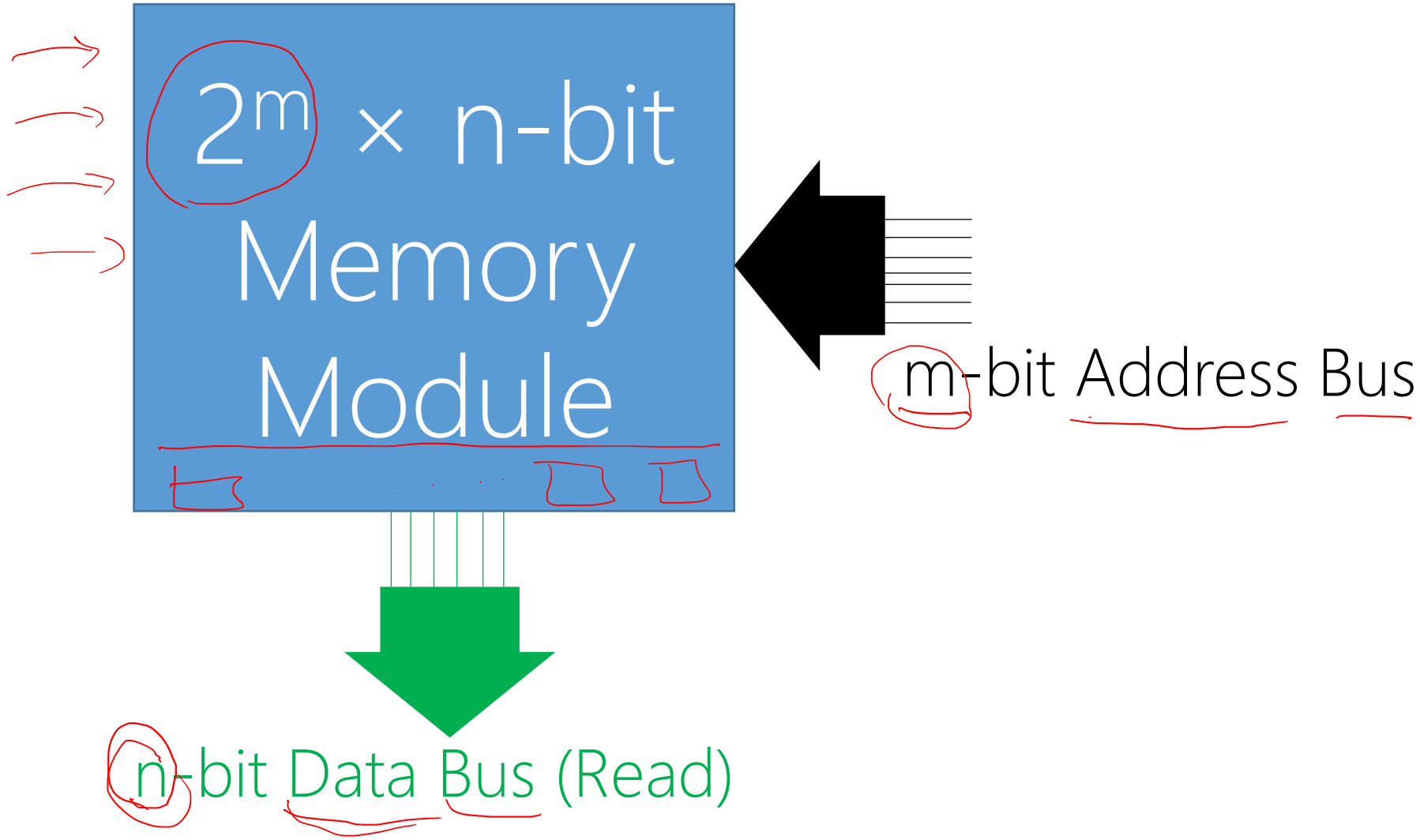
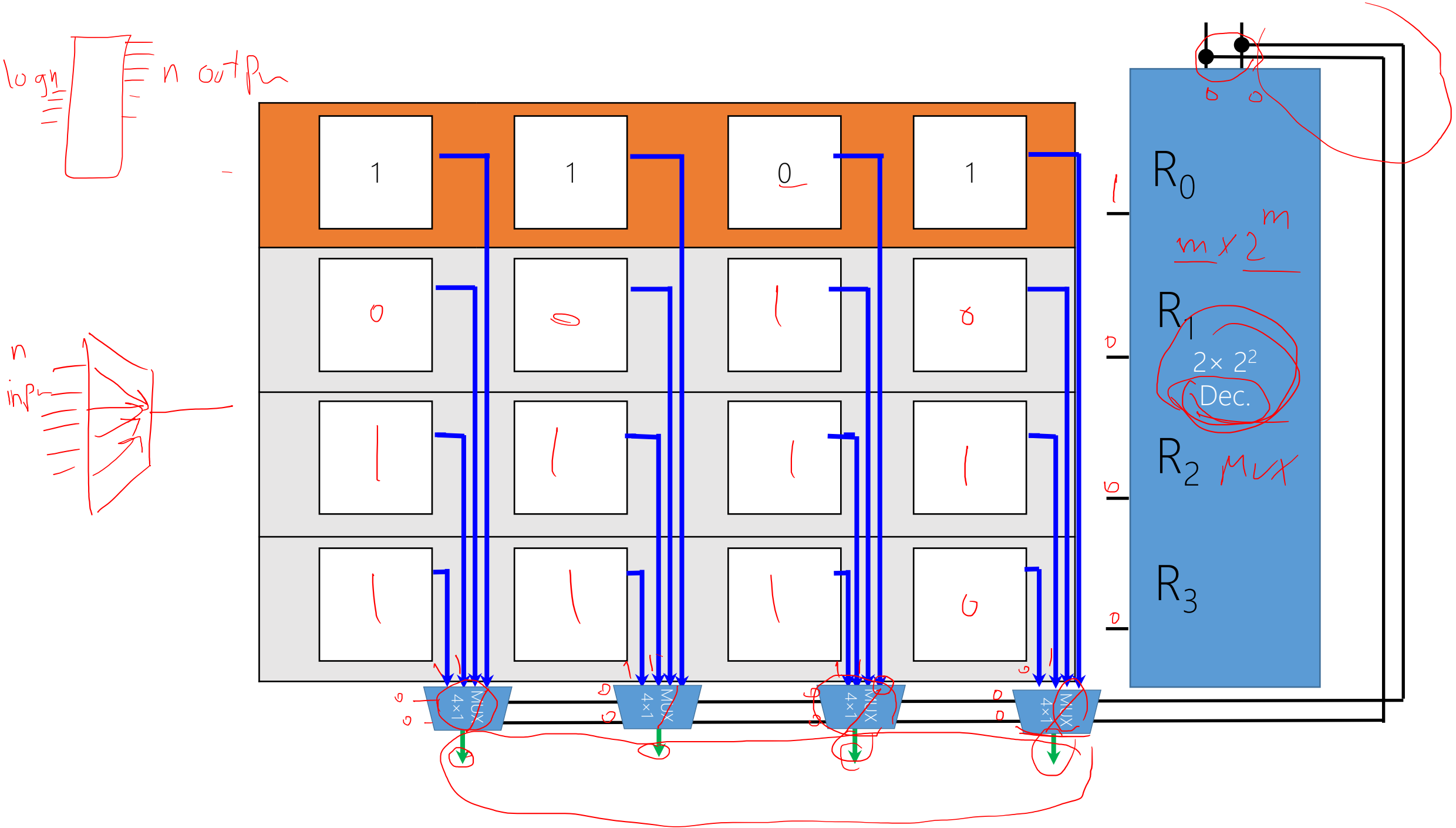
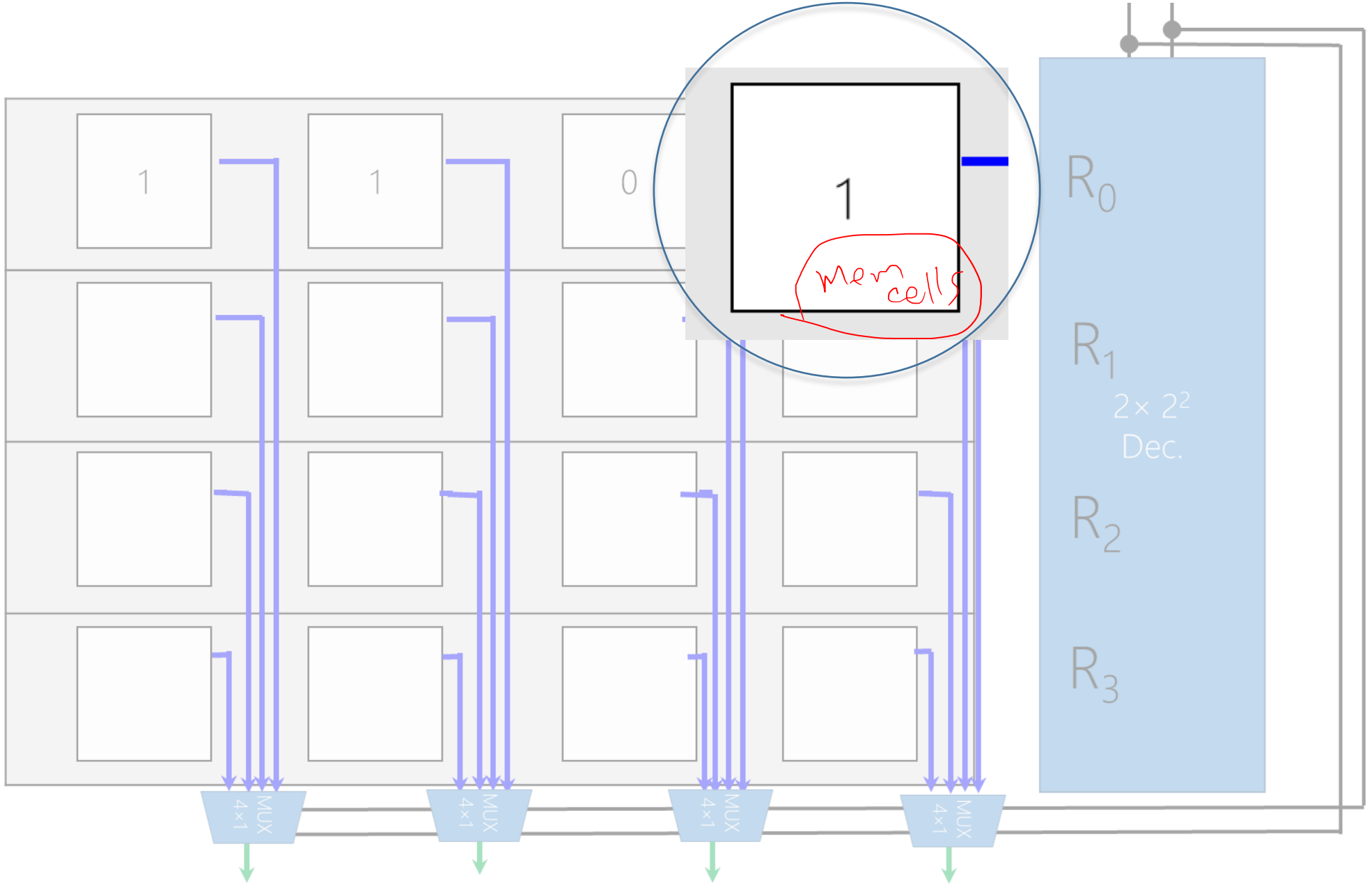
Memento, Christopher Nolan
Guy Pearce, Carrie-Anne Moss, Joe Pantoliano
September 2000
Budget $4.5 m
Box office $40 m

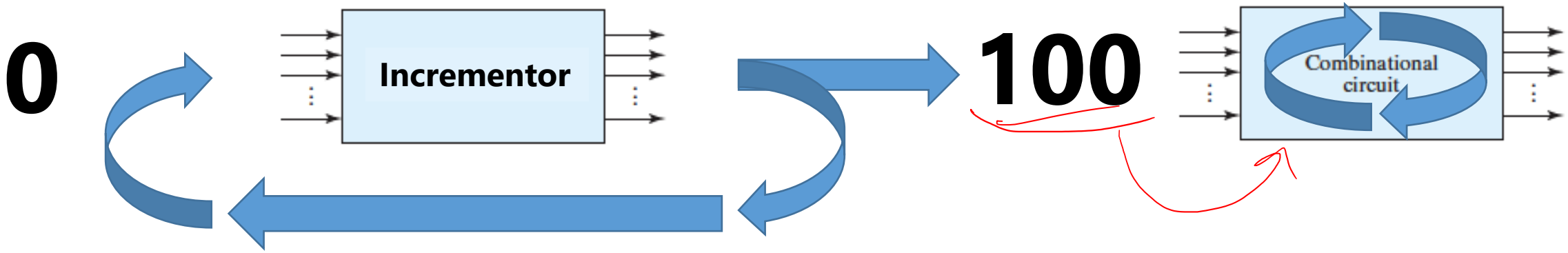

$2^m \times$ n-bit Memory Module
m-bit Address Bus
n-bit Data Bus (Read)

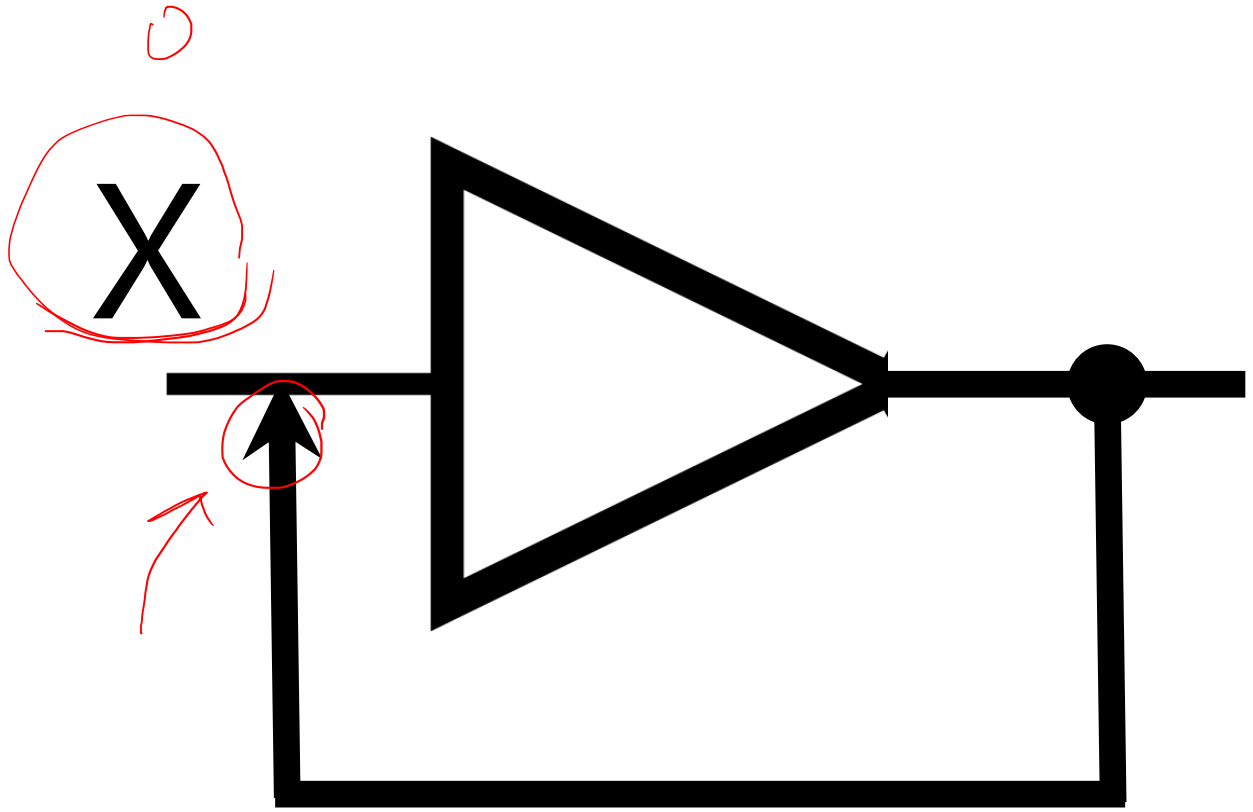

1
1
0
1
0
0
1
0
1
1
1
1
1
1
1
0
MUX 4×1
MUX 4×1
MUX 4×1
MUX 4×1
R_0
R_1
2×2^2
Dec.
R_2
R_3


1
1
0
1
mem cells
R_0
R_1
2×2^2
Dec.
R_2
R_3
MUX 4×1
MUX 4×1
MUX 4×1
MUX 4×1

Sequential Logic

Counter: 0 → 1 → 2 → …

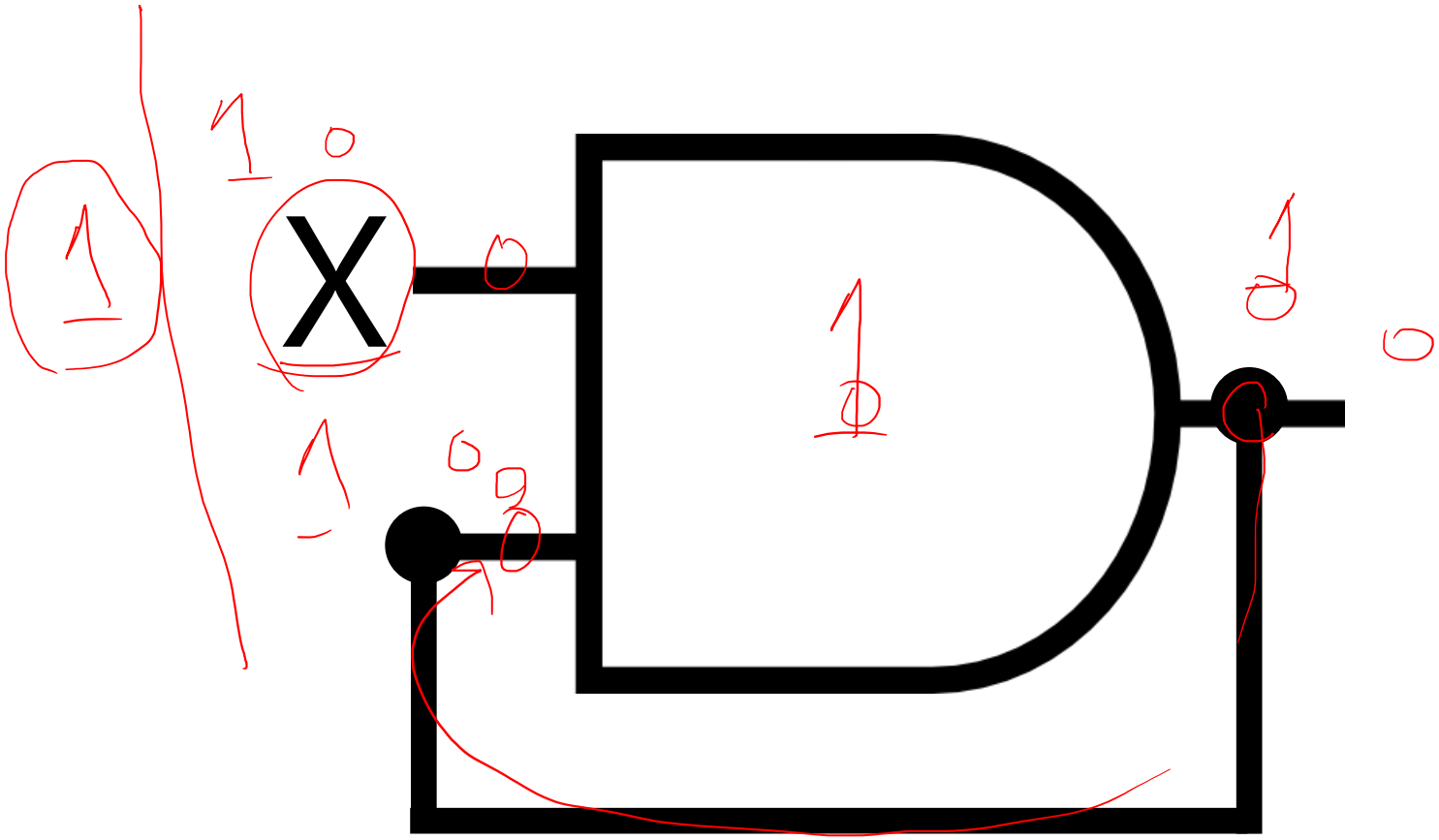




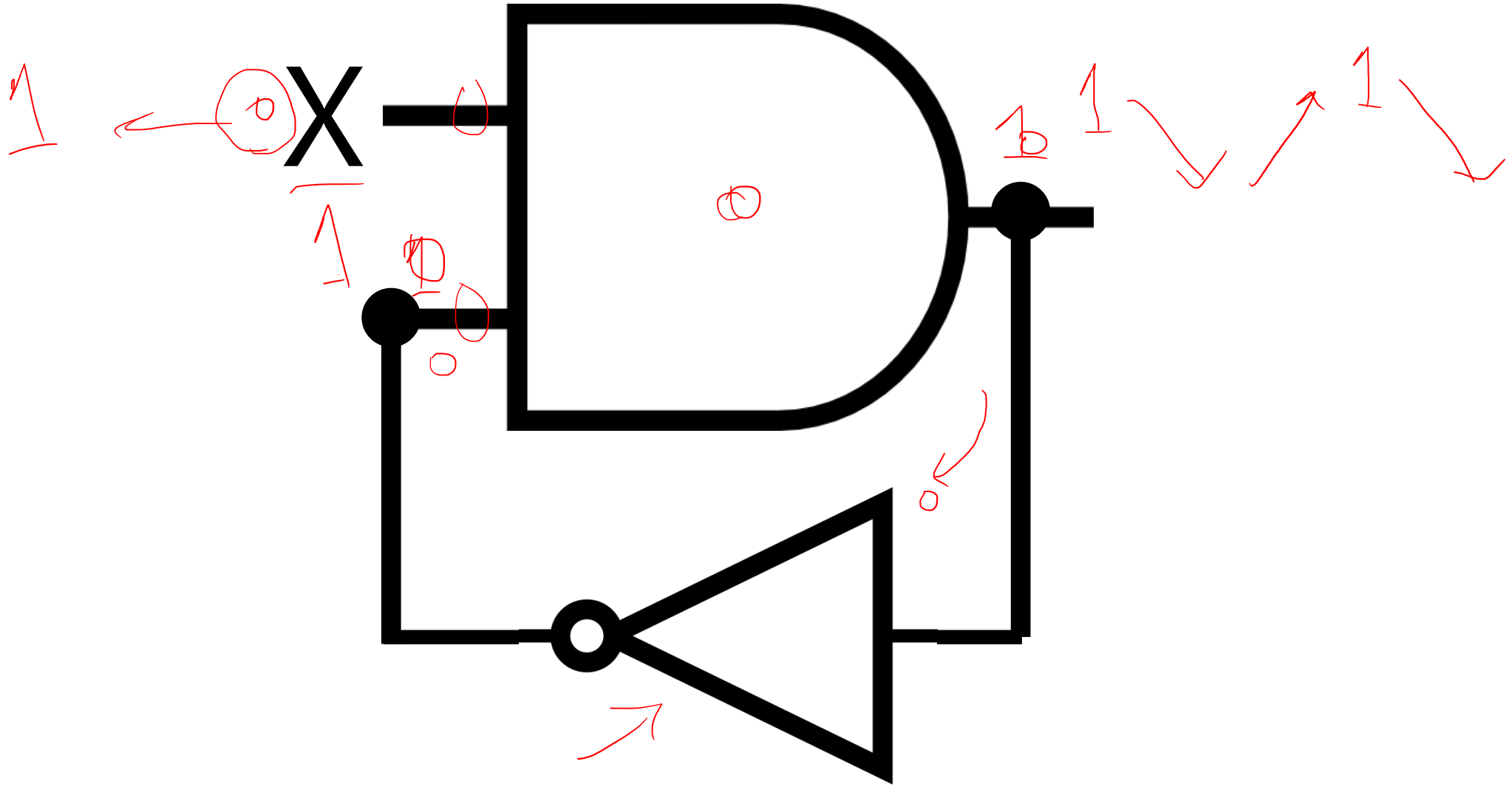
This design is logically incorrect! Why?

Buffer gate accepts one input.

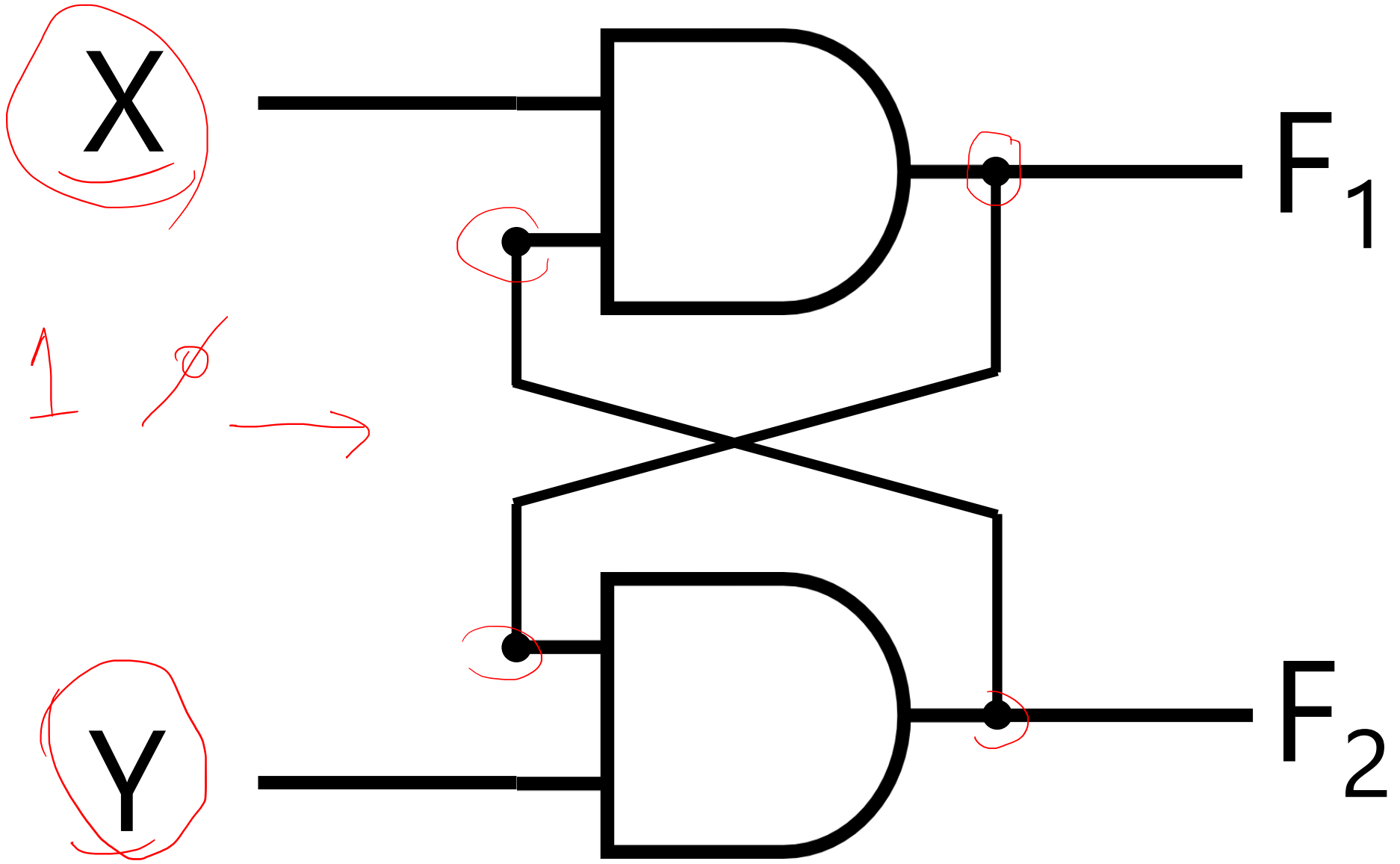
We have two inputs: X and X_{T-1}

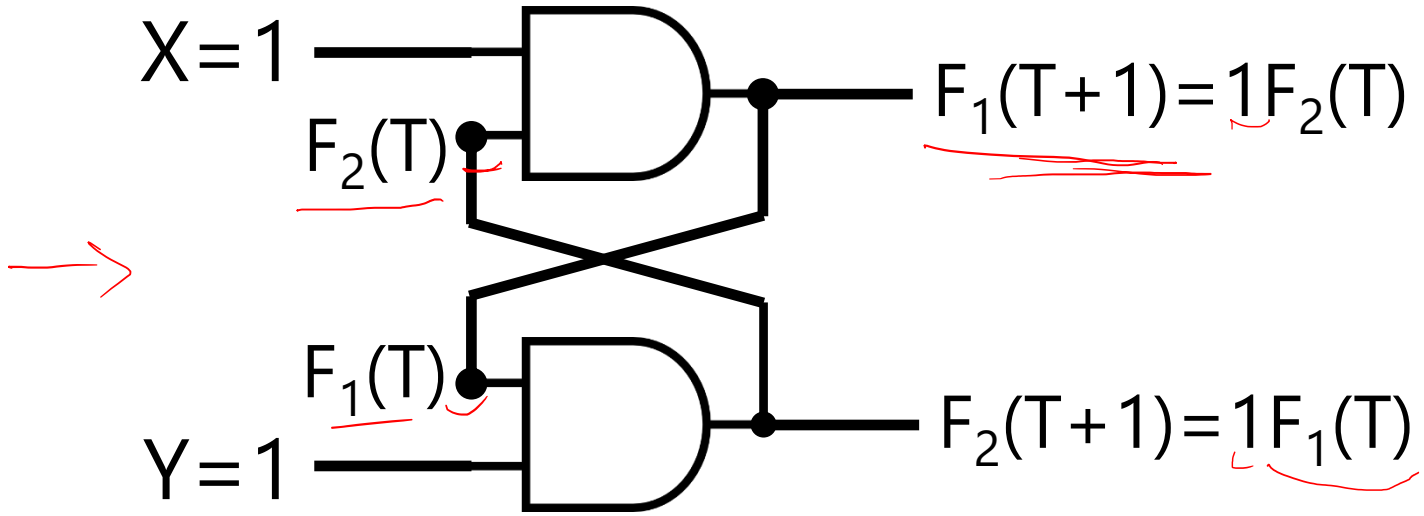


This design is logically correct!
But what's the problem?

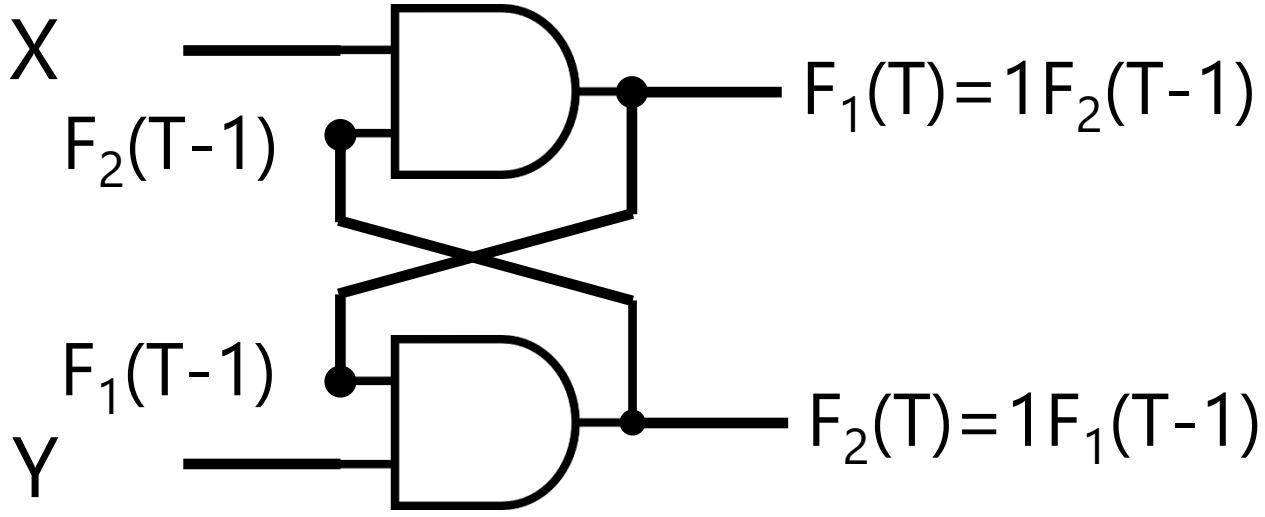


This design is also logically correct.
But what's the problem?

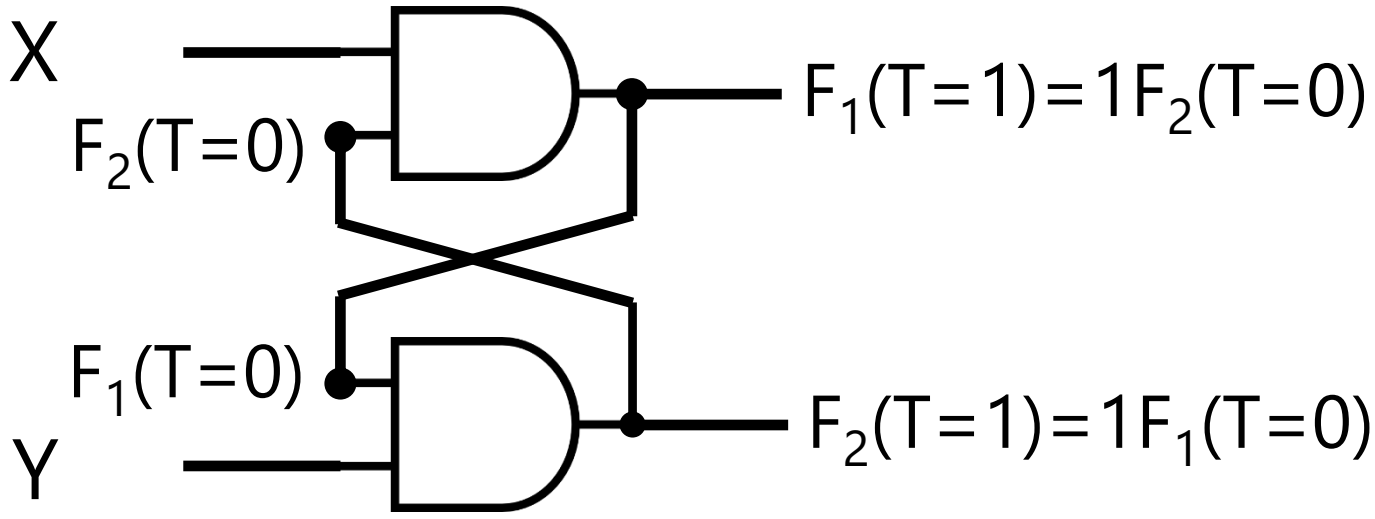

X
Y
F1
F2



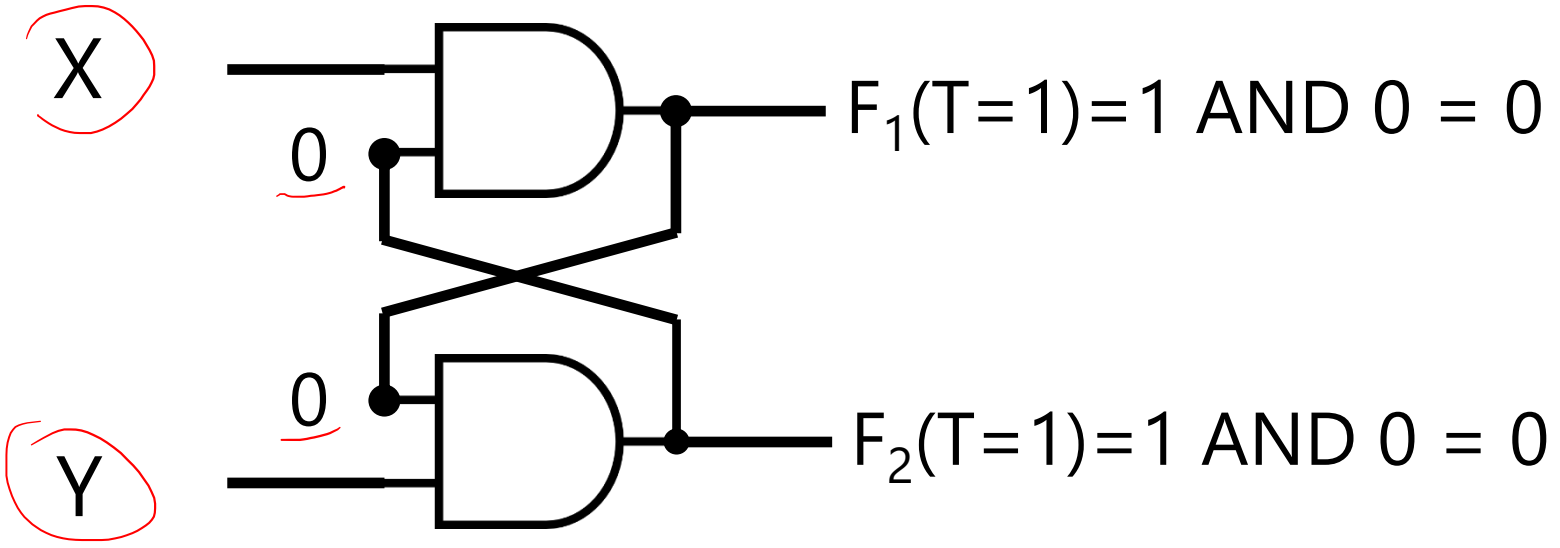
Y	X	$F_1(T+1)$	$F_2(T+1)$
0	0	0	0
0	1	$F_2(T)$	0
1	0	0	$F_1(T)$
1	1	$F_2(T)$	$F_1(T)$



Y	X	$F_1(T)$	$F_2(T)$
0	0	0	0
0	1	$F_2(T-1)$	0
1	0	0	$F_1(T-1)$
1	1	$F_2(T-1)$	$F_1(T-1)$

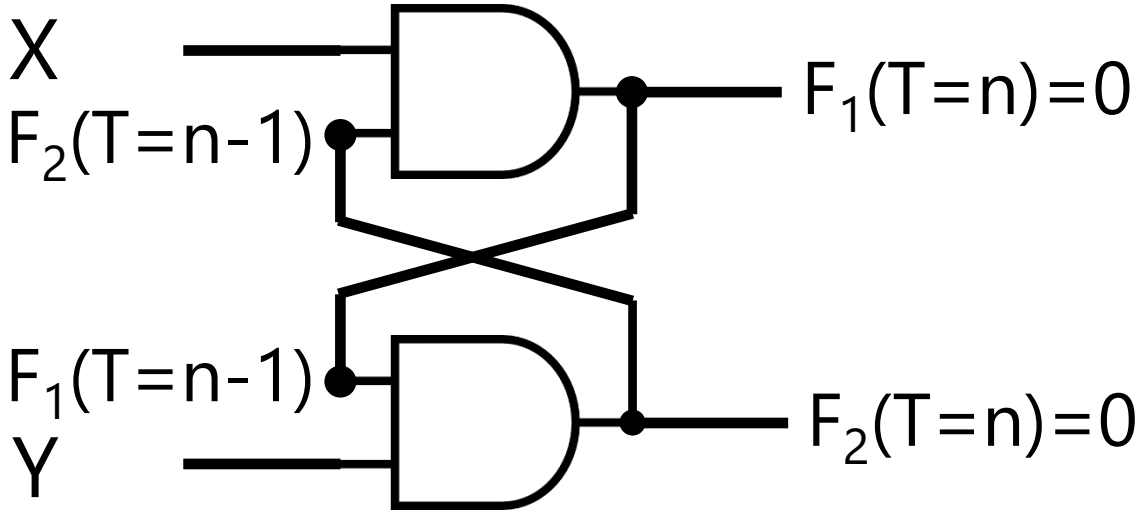


Y	X	$F_1(T=1)$	$F_2(T=1)$
0	0	0	0
0	1	$F_2(T=0)$	0
1	0	0	$F_1(T=0)$
1	1	$F_2(T=0)$	$F_1(T=0)$



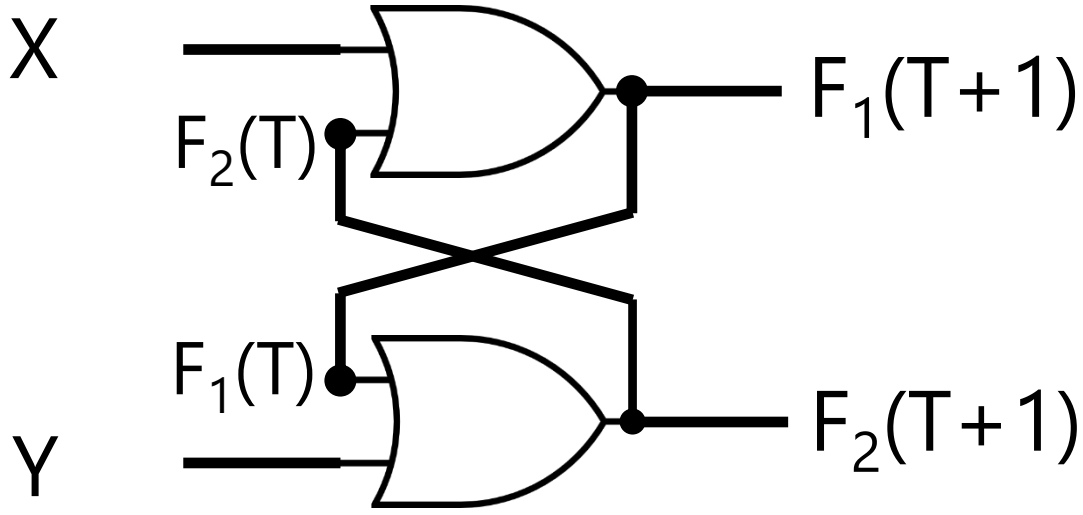
by default, in positive logic F(T=0) = 0

Y	X	$F_1(T=1)$	$F_2(T=1)$
0	0	0	0
0	1	0	0
1	0	0	0
1	1	0	0



When it goes to 00 state, never recover to 1!

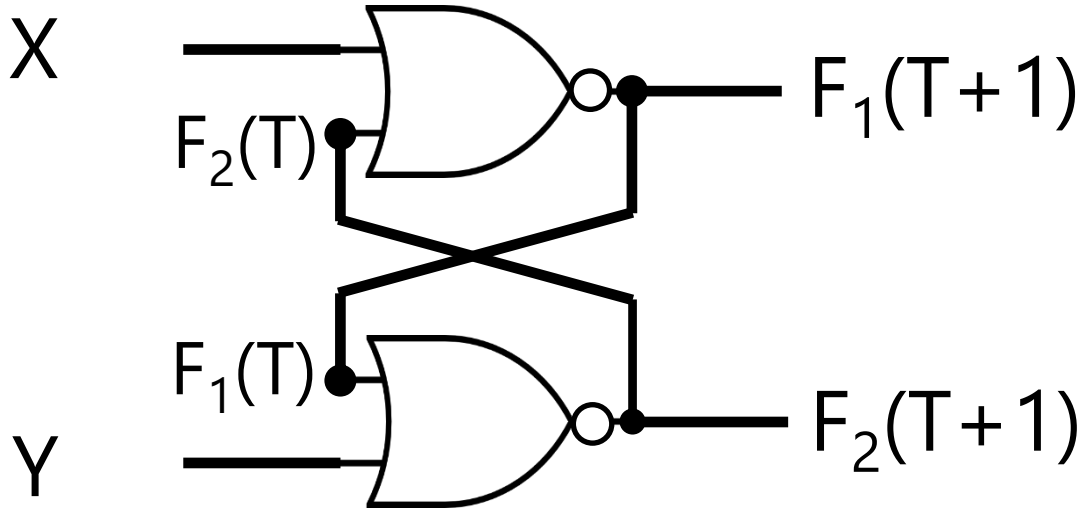
Y	X	$F_1(T=n)$	$F_2(T=n)$
0	0	0	0
0	1	0	0
1	0	0	0
1	1	0	0



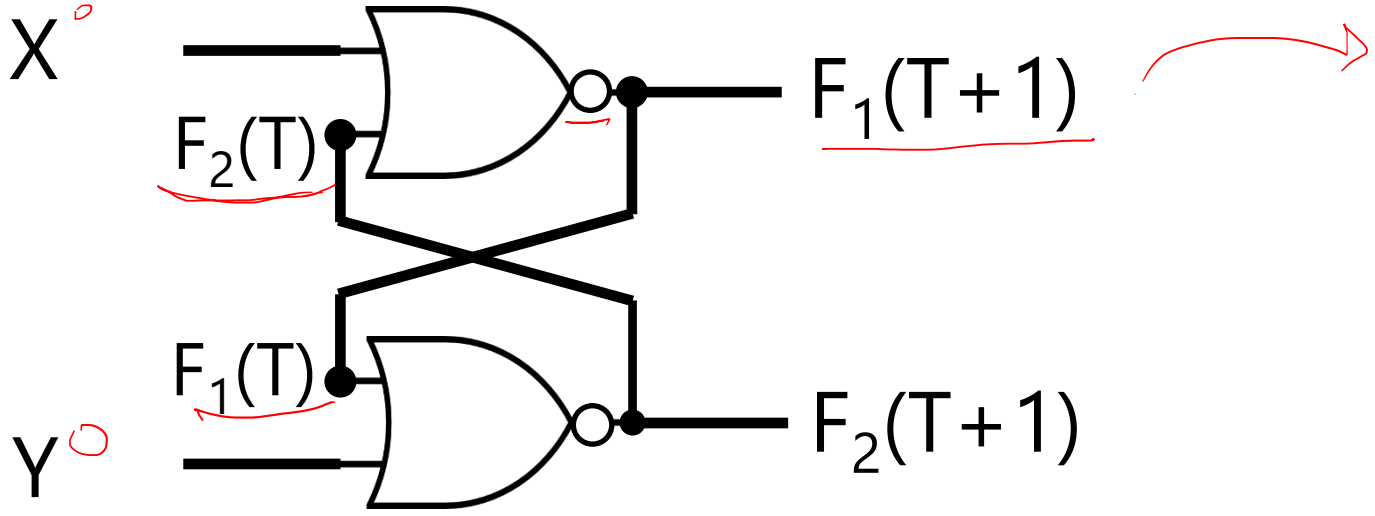
How about OR gates? Do they solve the problem? No! Why?

Y	X	$F_1(T+1)$	$F_2(T+1)$
0	0		
0	1		
1	0		
1	1		

Try NOR gates

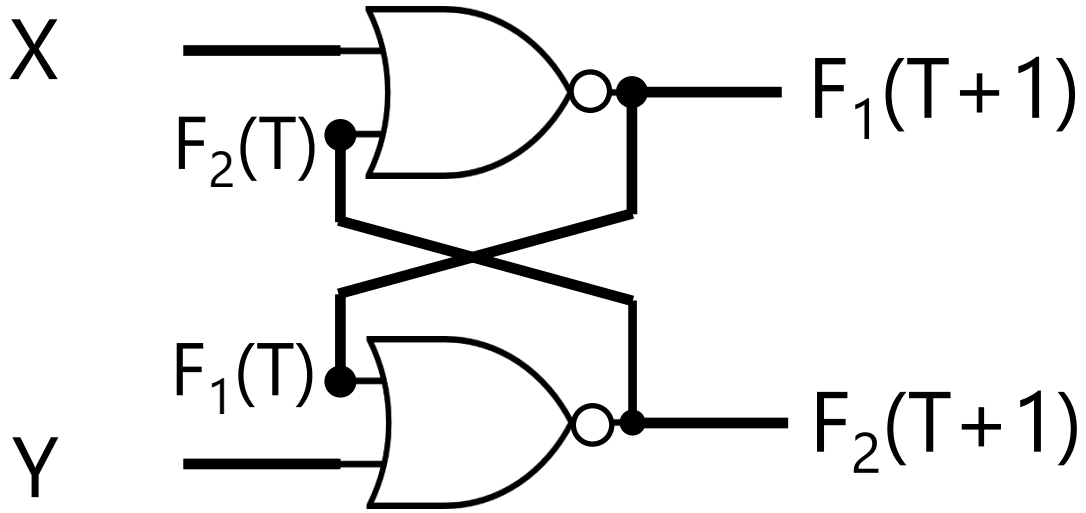


Y	X	$F_1(T+1)$	$F_2(T+1)$	$F_1(T+2)$	$F_2(T+2)$
0	0	$F'_2(T)$	$F'_1(T)$	$F'_2(T+1)$	$F'_1(T+1)$
0	1	0	$F'_1(T)$	0	$F'_1(T+1)$
1	0	$F'_2(T)$	0	$F'_2(T+1)$	0
1	1	0	0	0	0



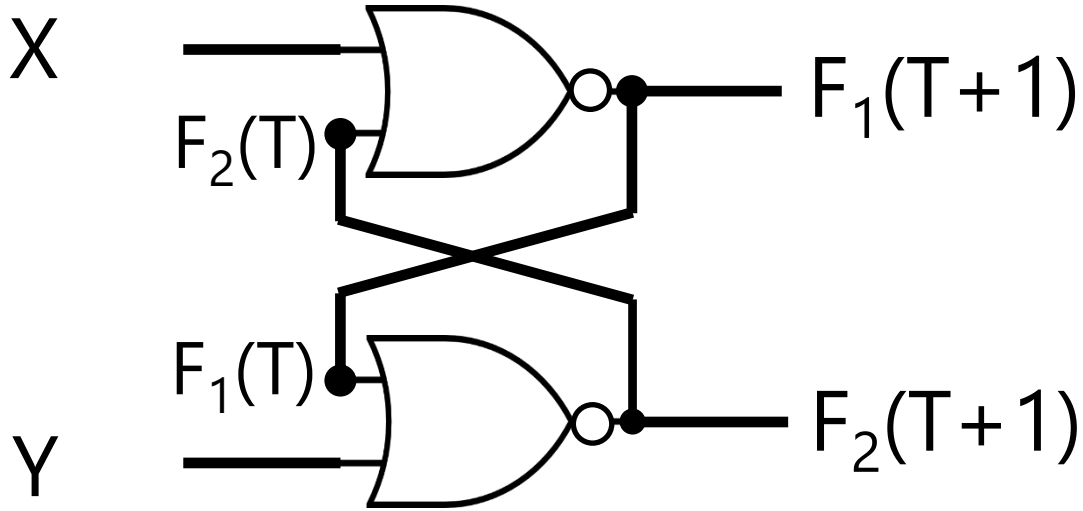
WOW! If wait longer, it stores the current state!

Y	X	$F_1(T+1)$	$F_2(T+1)$	$F_1(T+2)$	$F_2(T+2)$
0	0	$F'_2(T)$	$F'_1(T)$	$F_1(T)$	$F_2(T)$
0	1	0	$F'_1(T)$	0	1
1	0	$F'_2(T)$	0	1	0
1	1	0	0	0	0



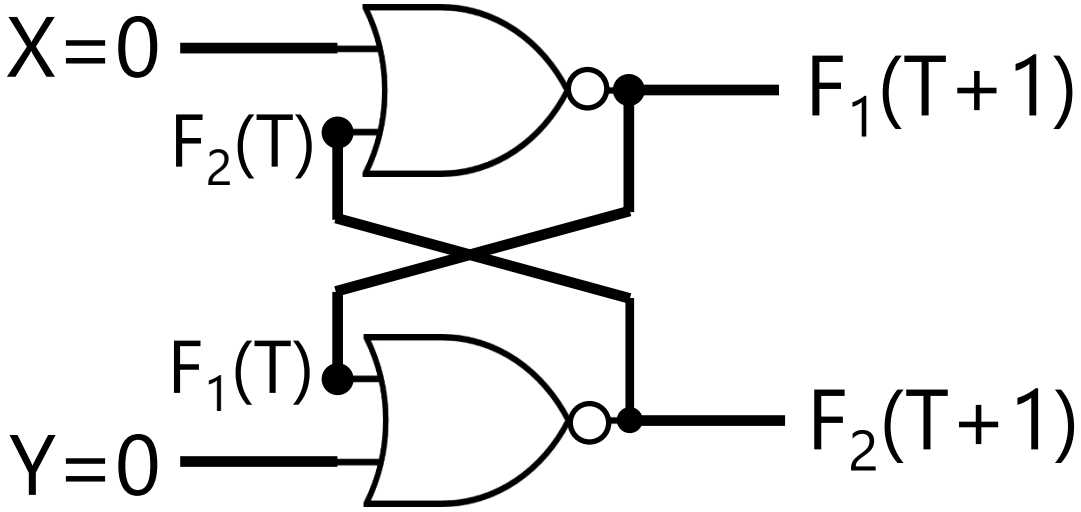
1 2 3 4

Y	X	$F_1(T+1)$	$F_2(T+1)$	$F_1(T+2)$	$F_2(T+2)$	$F_1(T+3)$	$F_2(T+3)$	$F_1(T+4)$	$F_2(T+4)$
0	0	$F'_2(T)$	$F'_1(T)$	$F_1(T)$	$F_2(T)$	$F'_2(T)$	$F'_1(T)$	$F'_2(T+3)$	$F'_1(T+3)$
0	1	0	$F'_1(T)$	0	1	0	1	0	$F'_1(T+3)$
1	0	$F'_2(T)$	0	1	0	0	0	$F'_2(T+3)$	0
1	1	0	0	0	0	0	0	0	0



Y	X	$F_1(T+1)$	$F_2(T+1)$	$F_1(T+2)$	$F_2(T+2)$	$F_1(T+3)$	$F_2(T+3)$	$F_1(T+4)$	$F_2(T+4)$
0	0	$F'_2(T)$	$F'_1(T)$	$F_1(T)$	$F_2(T)$	$F'_2(T)$	$F'_1(T)$	$F_1(T)$	$F_2(T)$
0	1	0	$F'_1(T)$	0	1	0	1	0	1
1	0	$F'_2(T)$	0	1	0	0	0	1	0
1	1	0	0	0	0	0	0	0	0

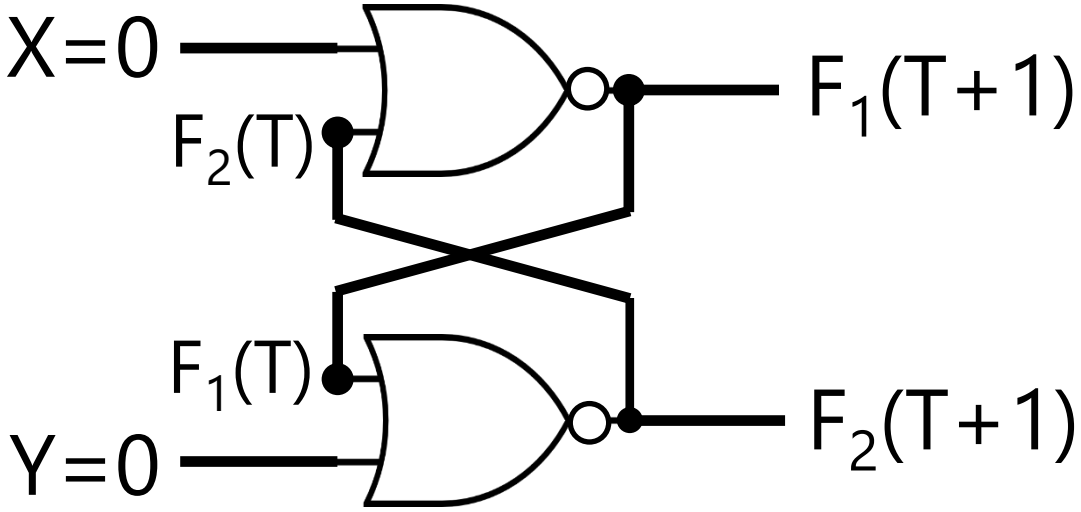
Store/Hold Action



Y	X	$F_1(T+1)$	$F_2(T+1)$	$F_1(T+2)$	$F_2(T+2)$	$F_1(T+3)$	$F_2(T+3)$	$F_1(T+4)$	$F_2(T+4)$
0	0	$F'_2(T)$	$F'_1(T)$	$F_1(T)$	$F_2(T)$	$F'_2(T)$	$F'_1(T)$	$F_1(T)$	$F_2(T)$
0	1	0	$F'_1(T)$	0	1	0	1	0	1
1	0	$F'_2(T)$	0	1	0	0	0	1	0
1	1	0	0	0	0	0	0	0	0

Store/Hold Action

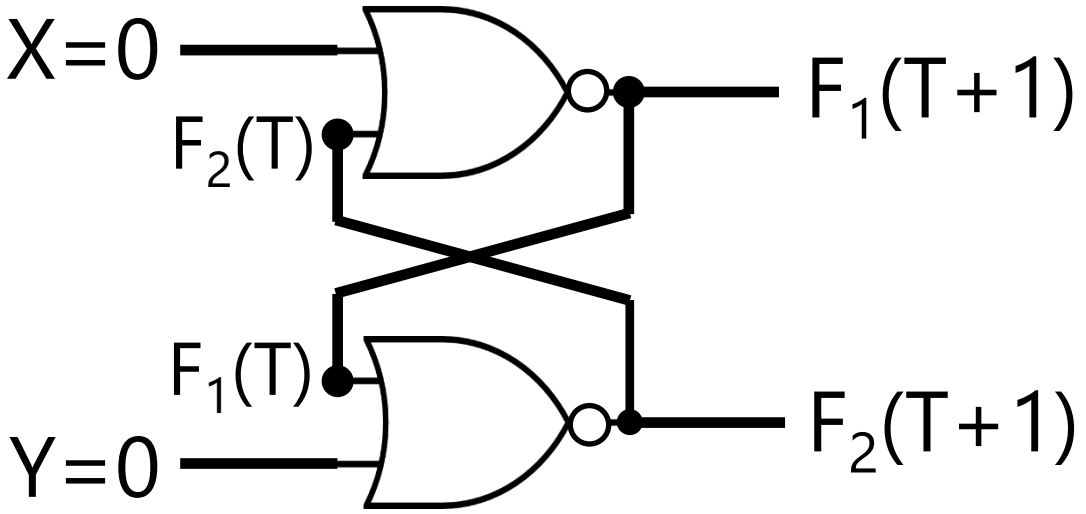
$2T \rightarrow T$



Y	X	$F_1(T+2)$	$F_2(T+2)$
0	0	$F_1(T)$	$F_2(T)$
0	1	0	1
1	0	1	0
1	1	0	0

Store/Hold Action

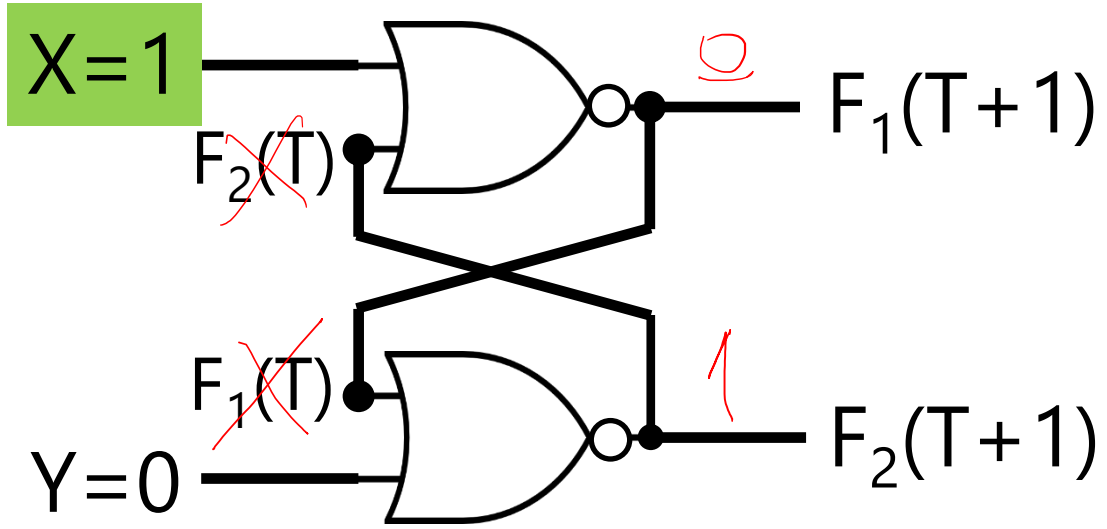
$2T \rightarrow T$



Y	X	$F_1(T+1)$	$F_2(T+1)$
0	0	$F_1(T)$	$F_2(T)$
0	1	0	1
1	0	1	0
1	1	0	0

Reset Action

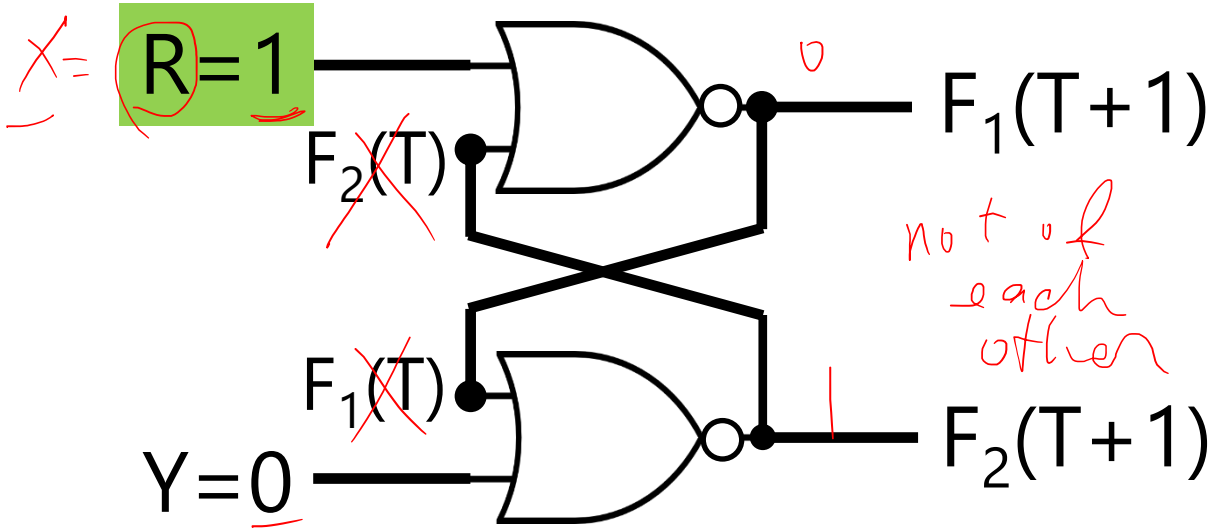
$2T \rightarrow T$



Y	X	$F_1(T+1)$	$F_2(T+1)$
0	0	$F_1(T)$	$F_2(T)$
0	1	0	1
1	0	1	0
1	1	0	0

Reset Action

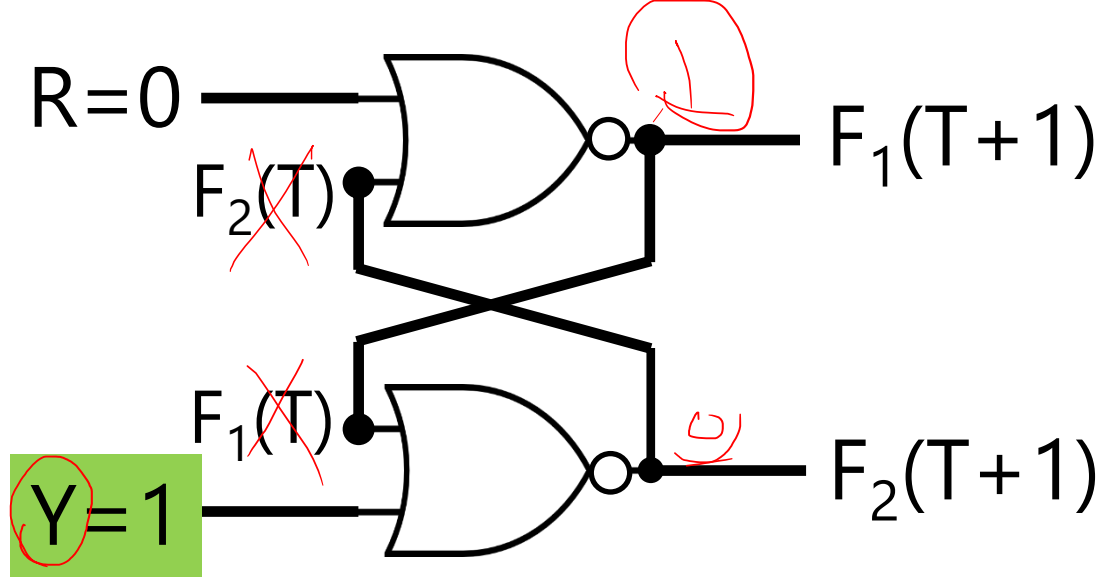
$2T \rightarrow T$



Y	R	$F_1(T+1)$	$F_2(T+1)$
0	0	$F_1(T)$	$F_2(T)$
0	1	0	1
1	0	1	0
1	1	0	0

Set Action

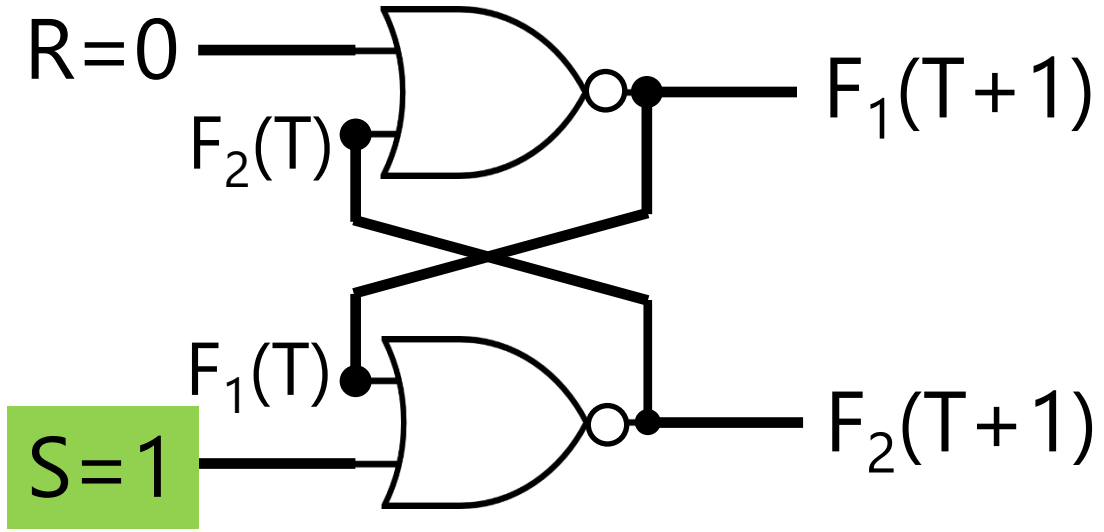
$2T \rightarrow T$



Y	R	$F_1(T+1)$	$F_2(T+1)$
0	0	$F_1(T)$	$F_2(T)$
0	1	0	1
1	0	1	0
1	1	0	0

Set Action

$2T \rightarrow T$

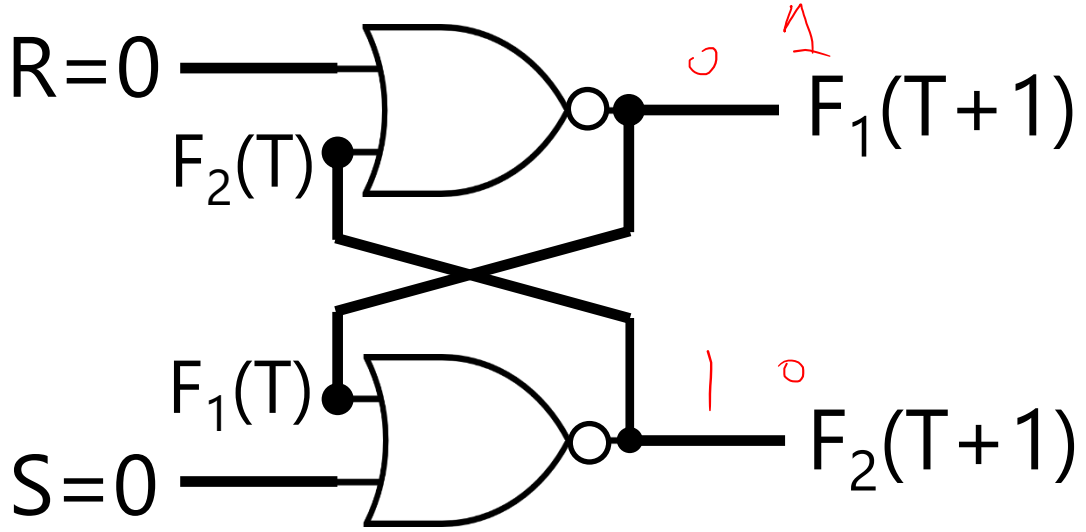


S	R	$F_1(T+1)$	$F_2(T+1)$
0	0	$F_1(T)$	$F_2(T)$
0	1	0	1
1	0	1	0
1	1	0	0

Store/Hold Action

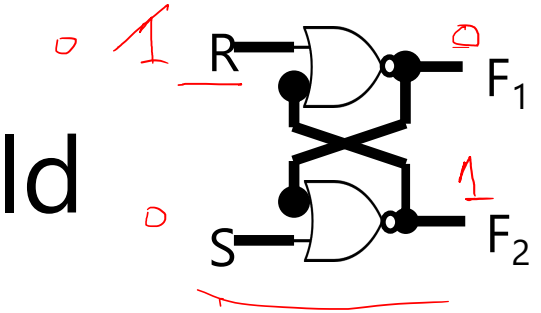
2T → T

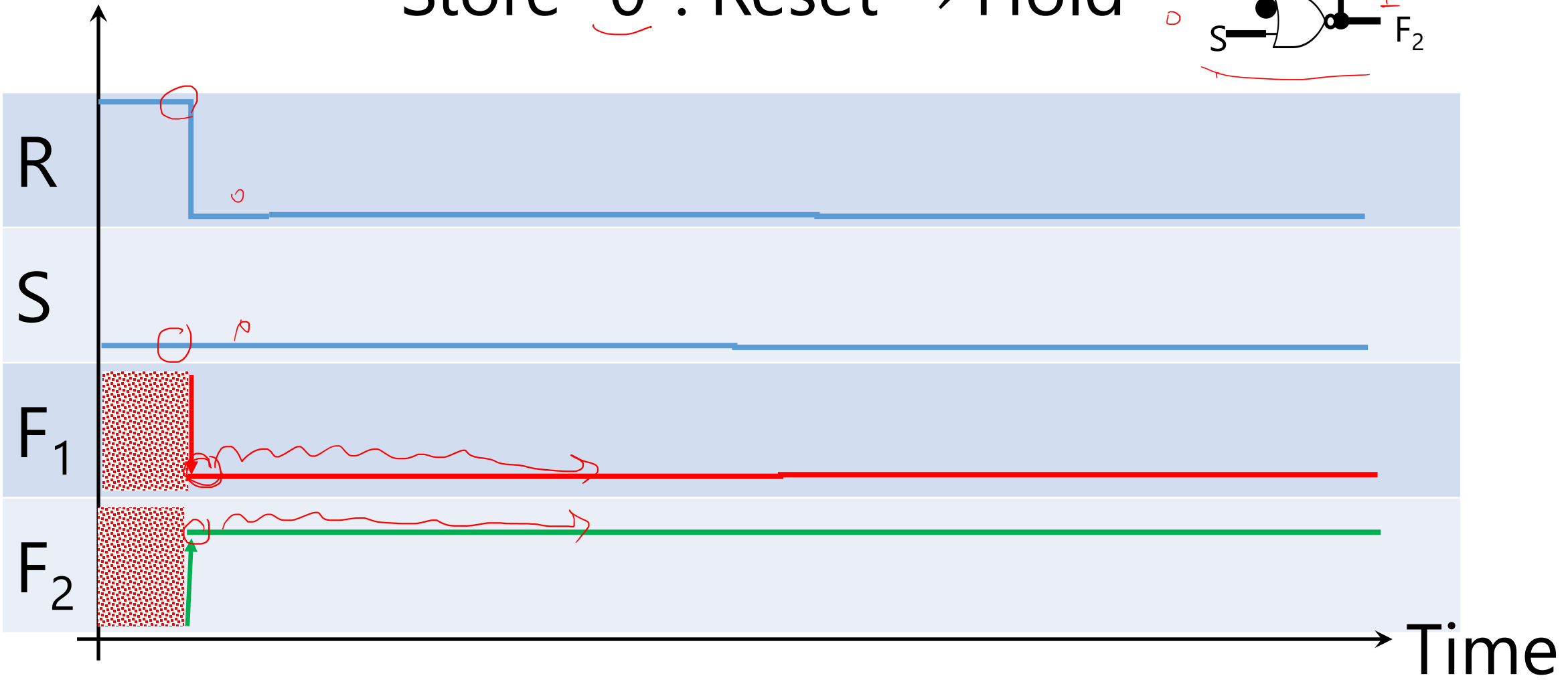
No Set & No Reset



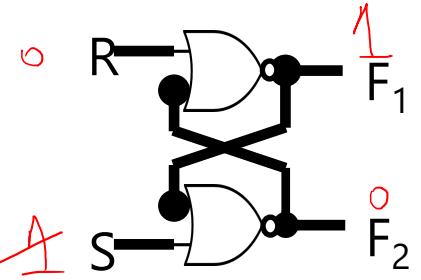
S	R	$F_1(T+1)$	$F_2(T+1)$
0	0	$F_1(T)$	$F_2(T)$
0	1	0	1
1	0	1	0
1	1	0	0

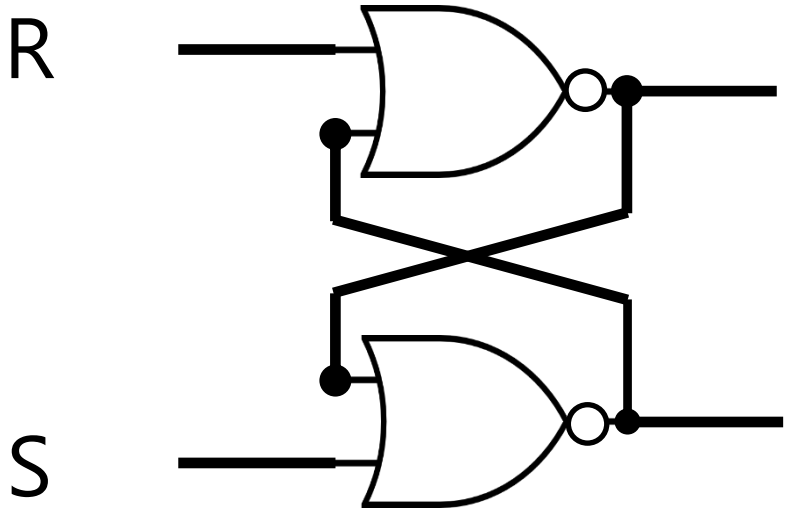
Store "0": Reset → Hold





Store "1": Set → Hold

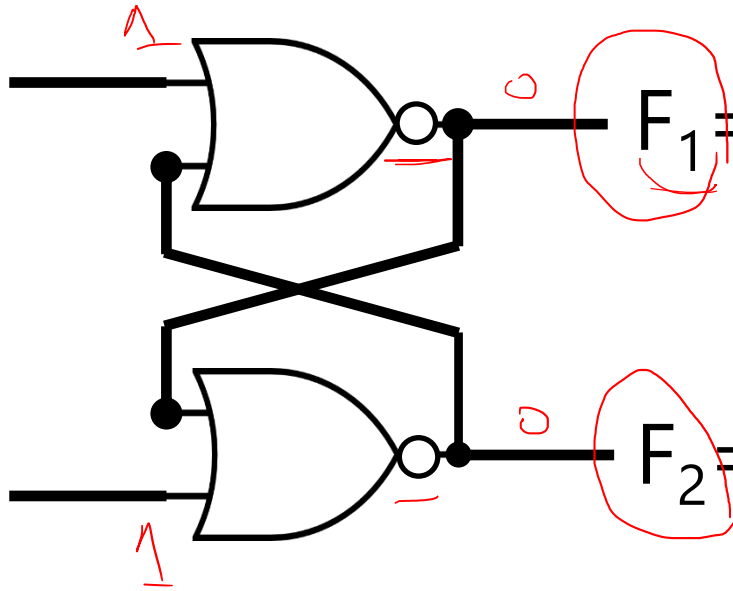




$F_1 = F'_2 = Q \rightarrow$ State

$F_2 = F'_1 = Q' \rightarrow$ NOT State

S	R	Q	Q′
0	0	Q_t	Q'_t
0	1	0	1
1	0	1	0
1	1	0	0



$F_1 = F'_2 = Q \rightarrow$ State

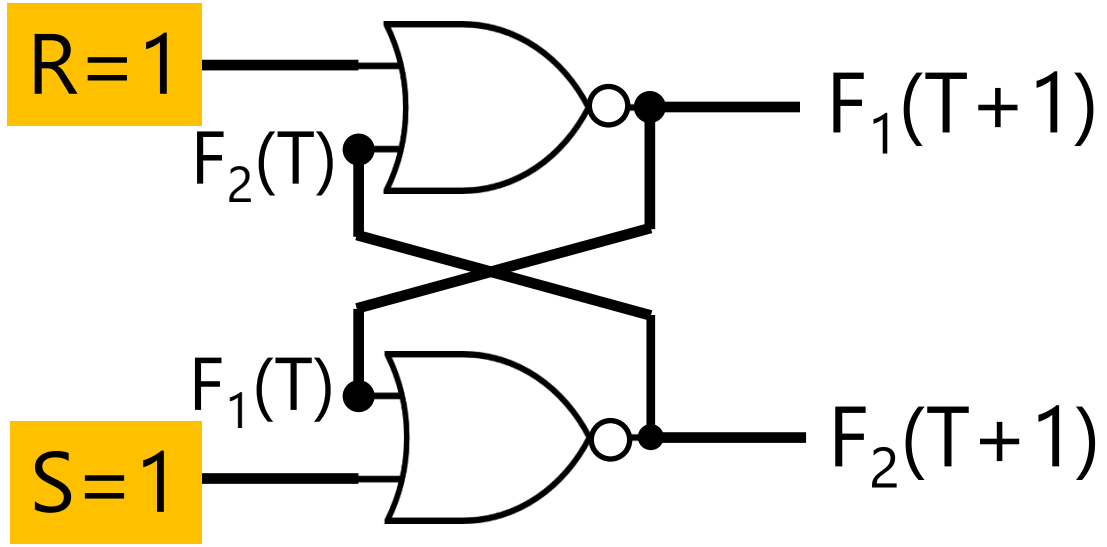
$F_2 = F'_1 = Q' \rightarrow$ NOT State

S	R	Q	Q′
0	0	Q_t	Q'_t
0	1	0	1
1	0	1	0
1	1	0	0

Forbidden Action

2T → T

~~*Set & Reset*~~

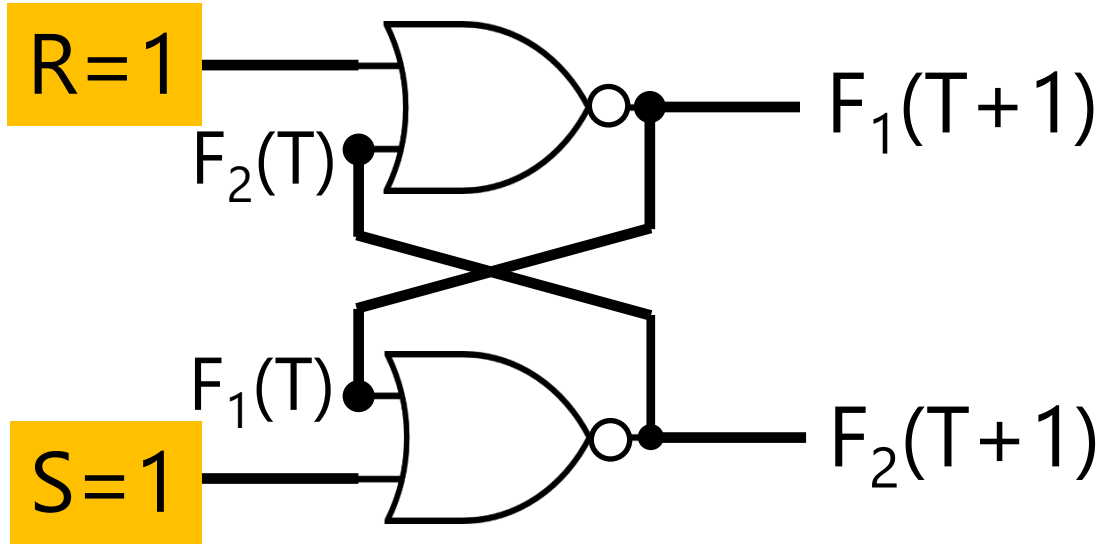


S	R	$F_1(T+1)$	$F_2(T+1)$
0	0	$F_1(T)$	$F_2(T)$
0	1	0	1
1	0	1	0
1	1	0	0

Forbidden Action

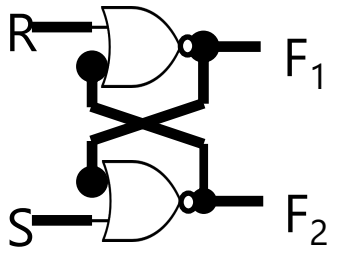
2T → T

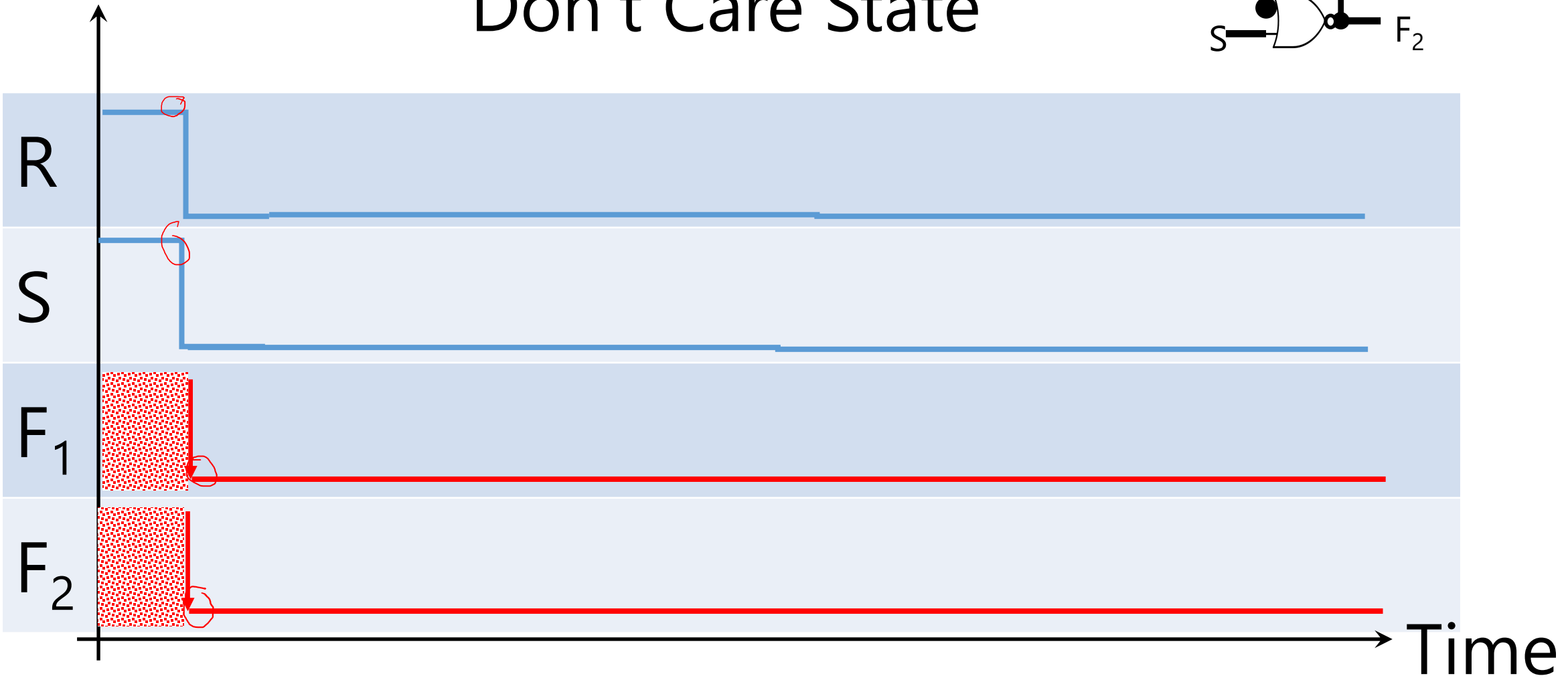
~~*Set & Reset*~~



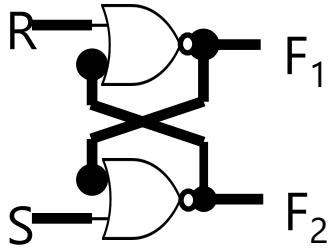
S	R	$F_1(T+1)$	$F_2(T+1)$
0	0	$F_1(T)$	$F_2(T)$
0	1	0	1
1	0	1	0
1	1	X	X

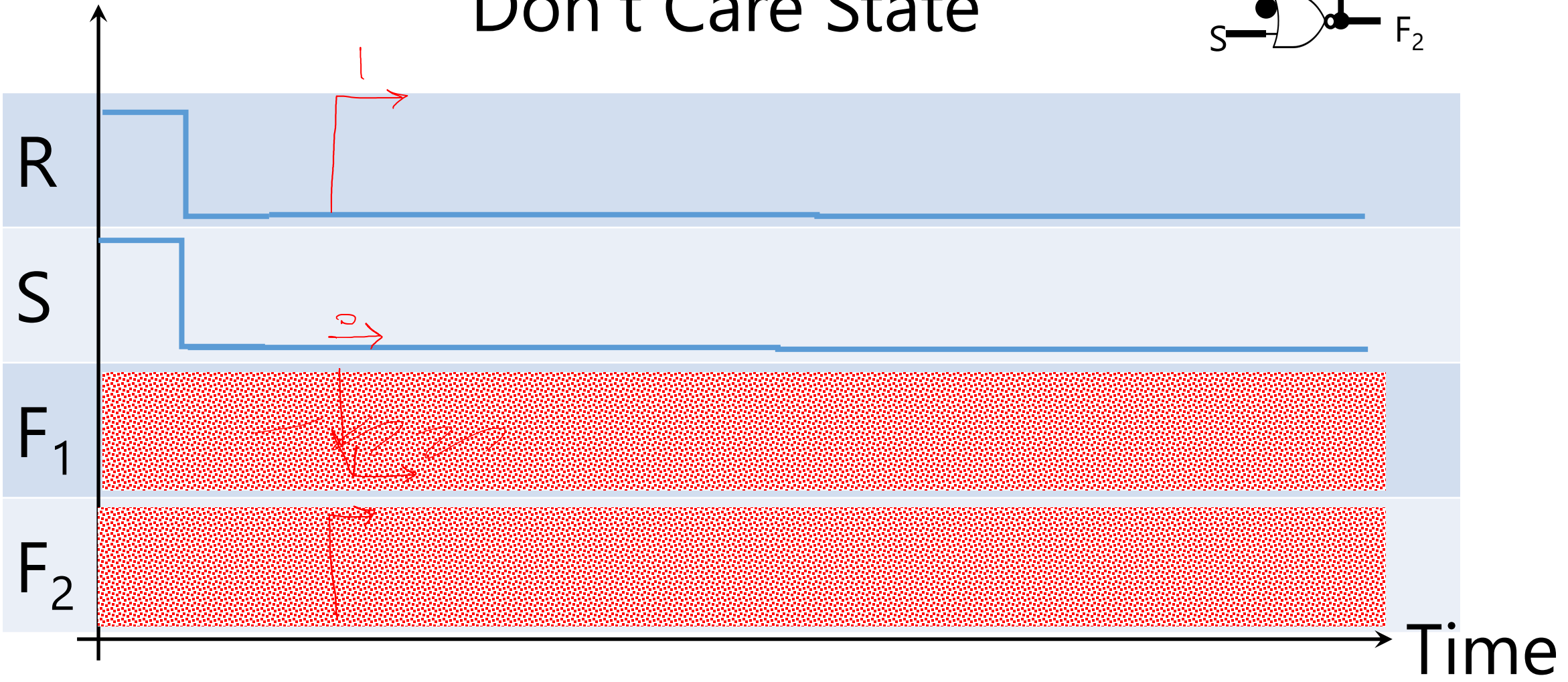
Don’t Care State

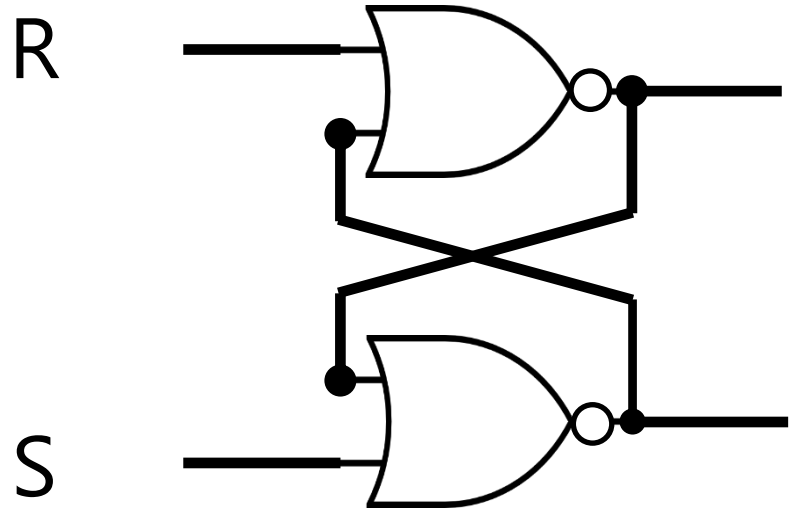




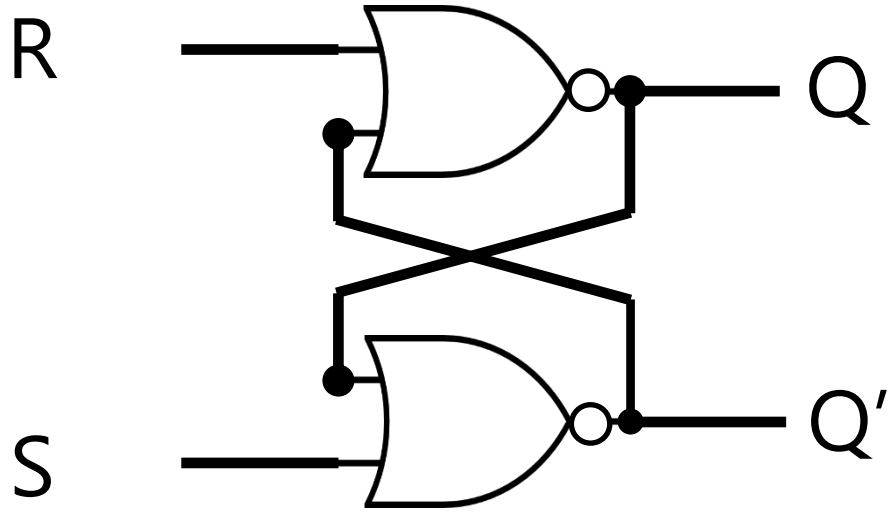
Don’t Care State



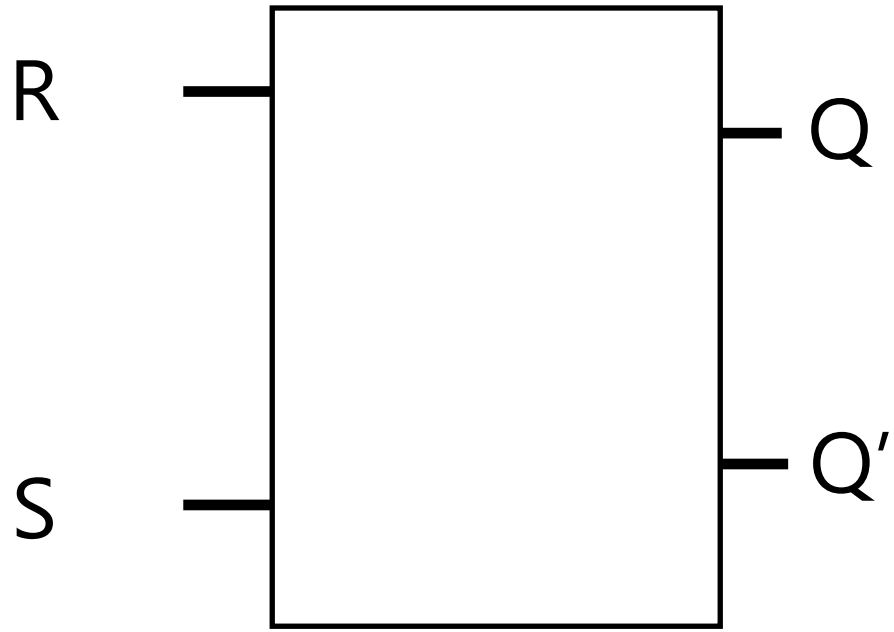




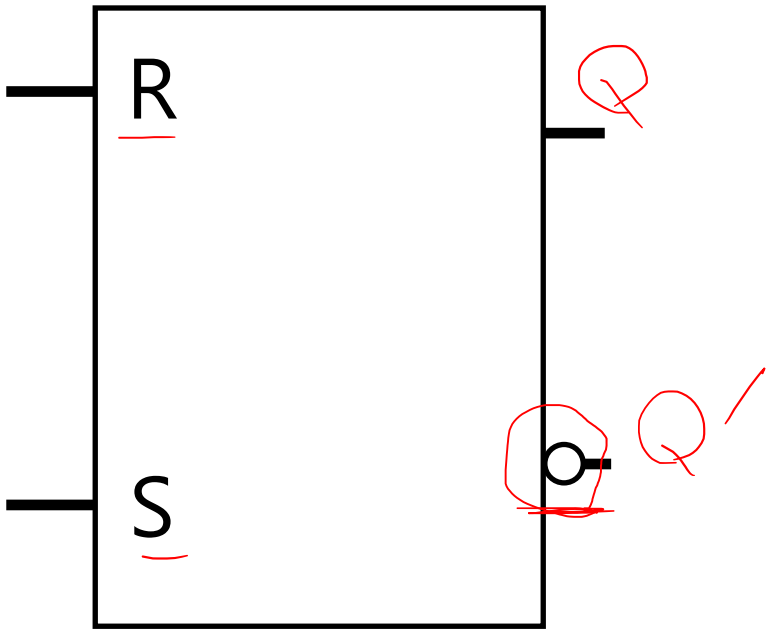
S	R	Q	Q′
0	0	Q_t	Q'_t
0	1	0	1
1	0	1	0
1	1	X	X



	S	R	Q	Q′
Hold	0	0	Q_t	Q'_t
Reset	0	1	0	1
Set	1	0	1	0
	1	1	x	x



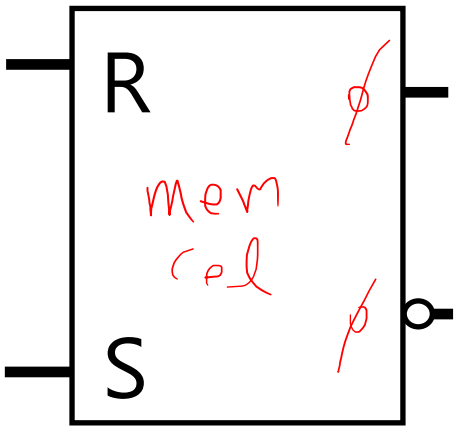
	S	R	Q	Q′
Hold	0	0	Q_t	Q'_t
Reset	0	1	0	1
Set	1	0	1	0
	1	1	X	X



	S	R	Q	Q′
Hold	0	0	Q_t	Q'_t
Reset	0	1	0	1
Set	1	0	1	0
	1	1	X	X

State Transition Diagram

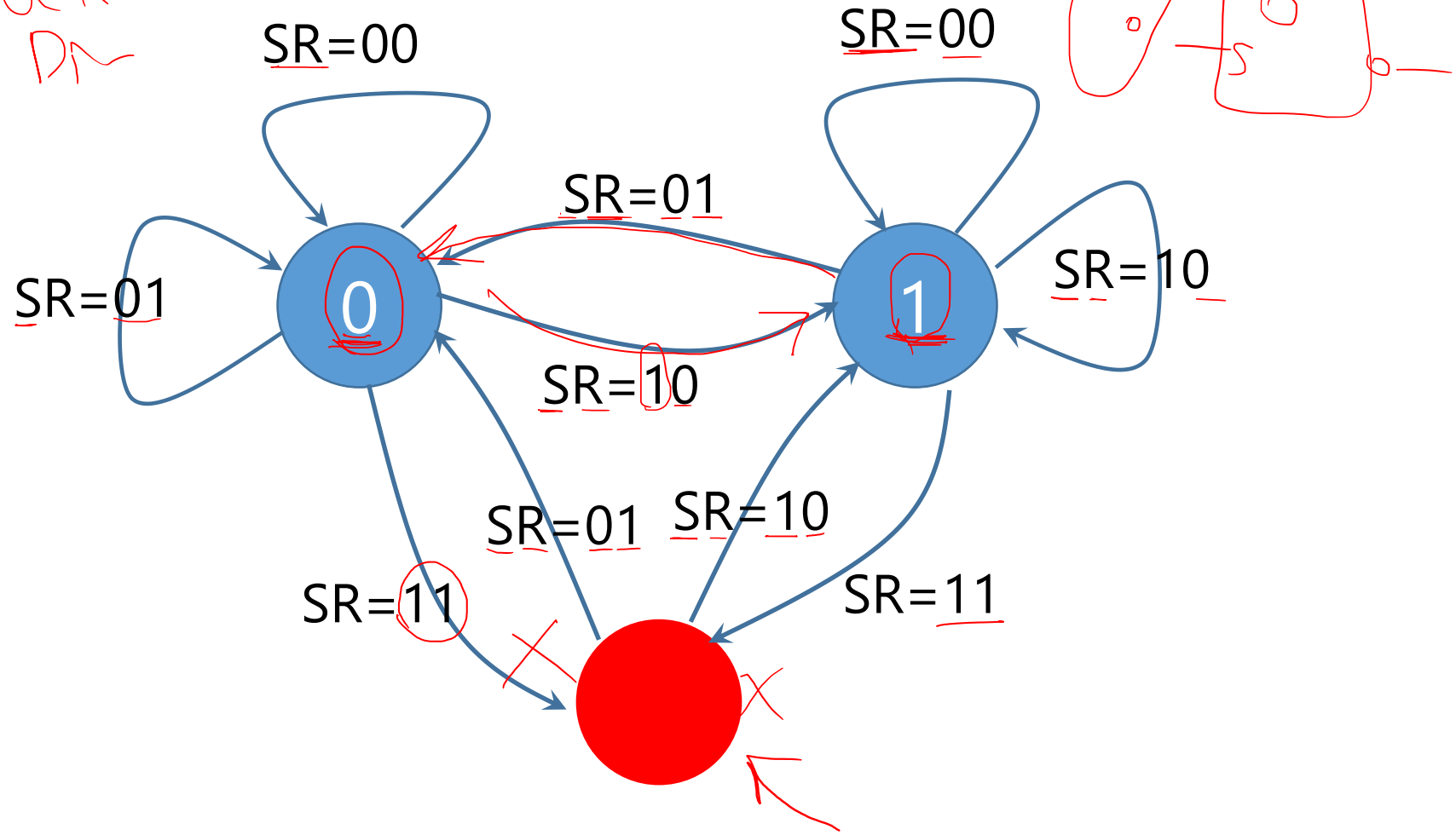
Life Cycle

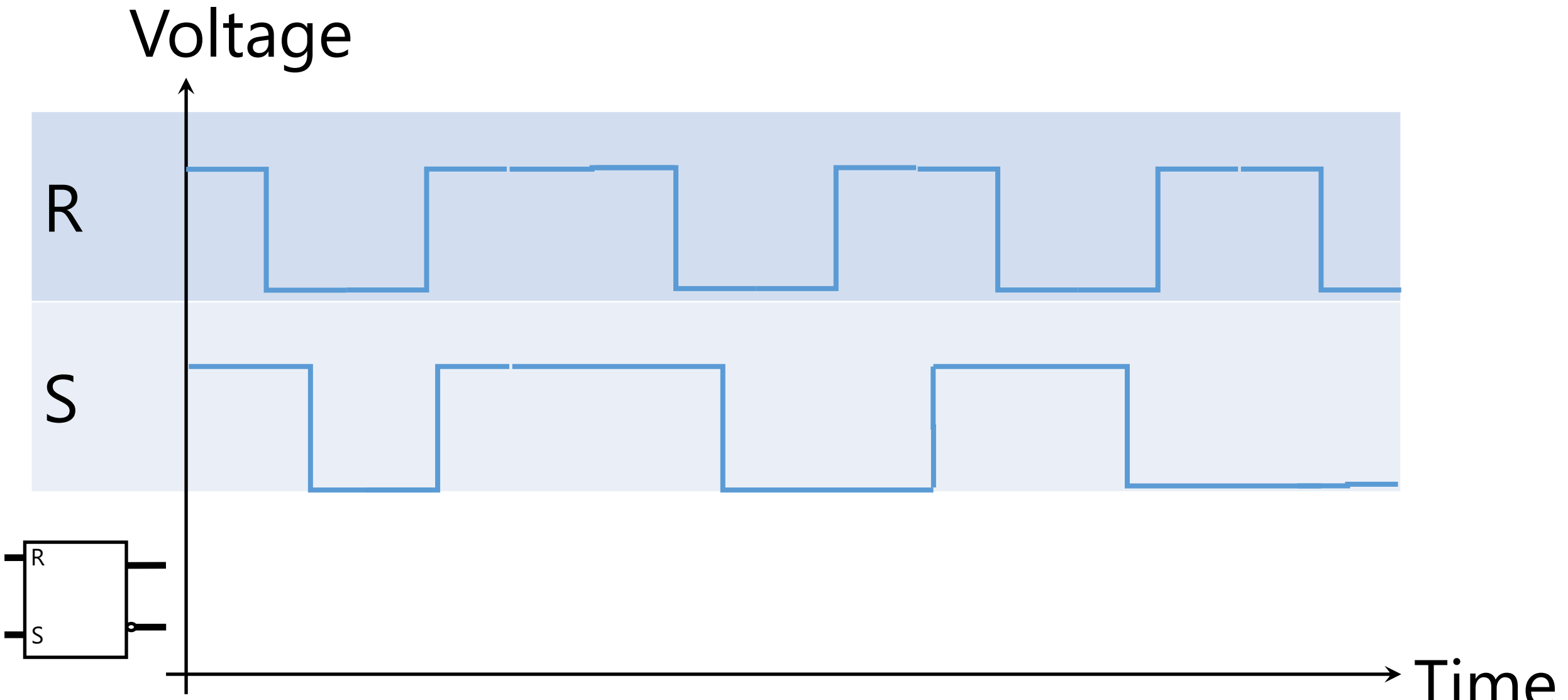


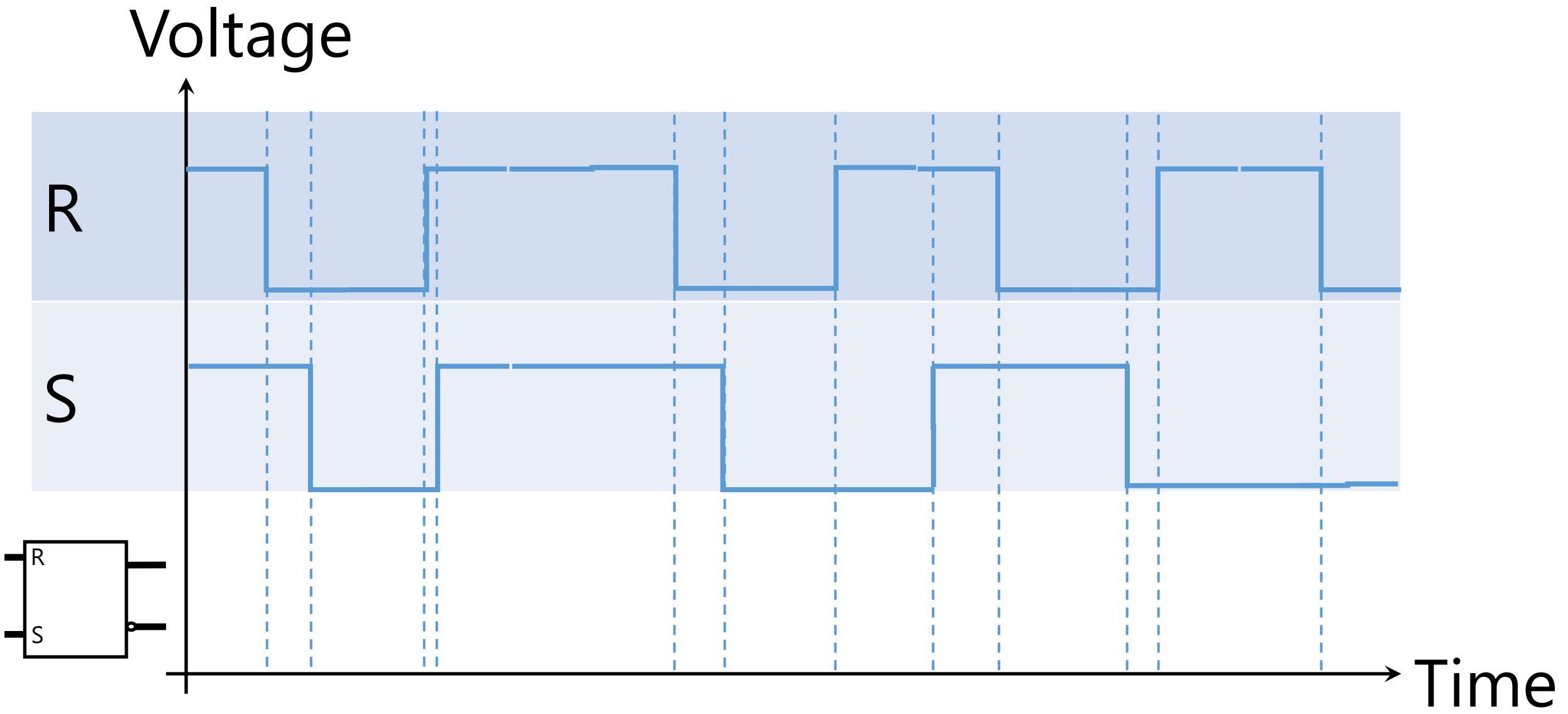
Block Diagram

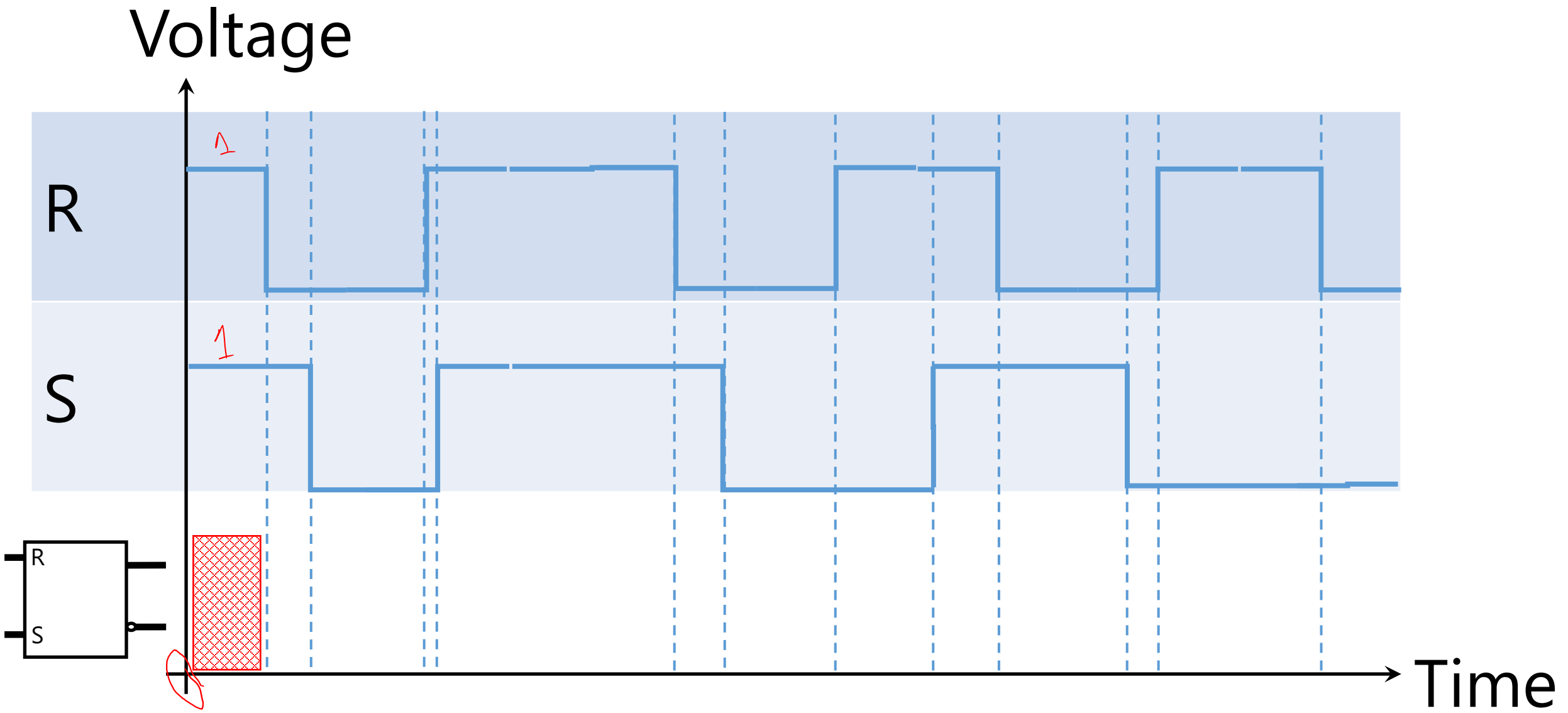
Characteristic Table

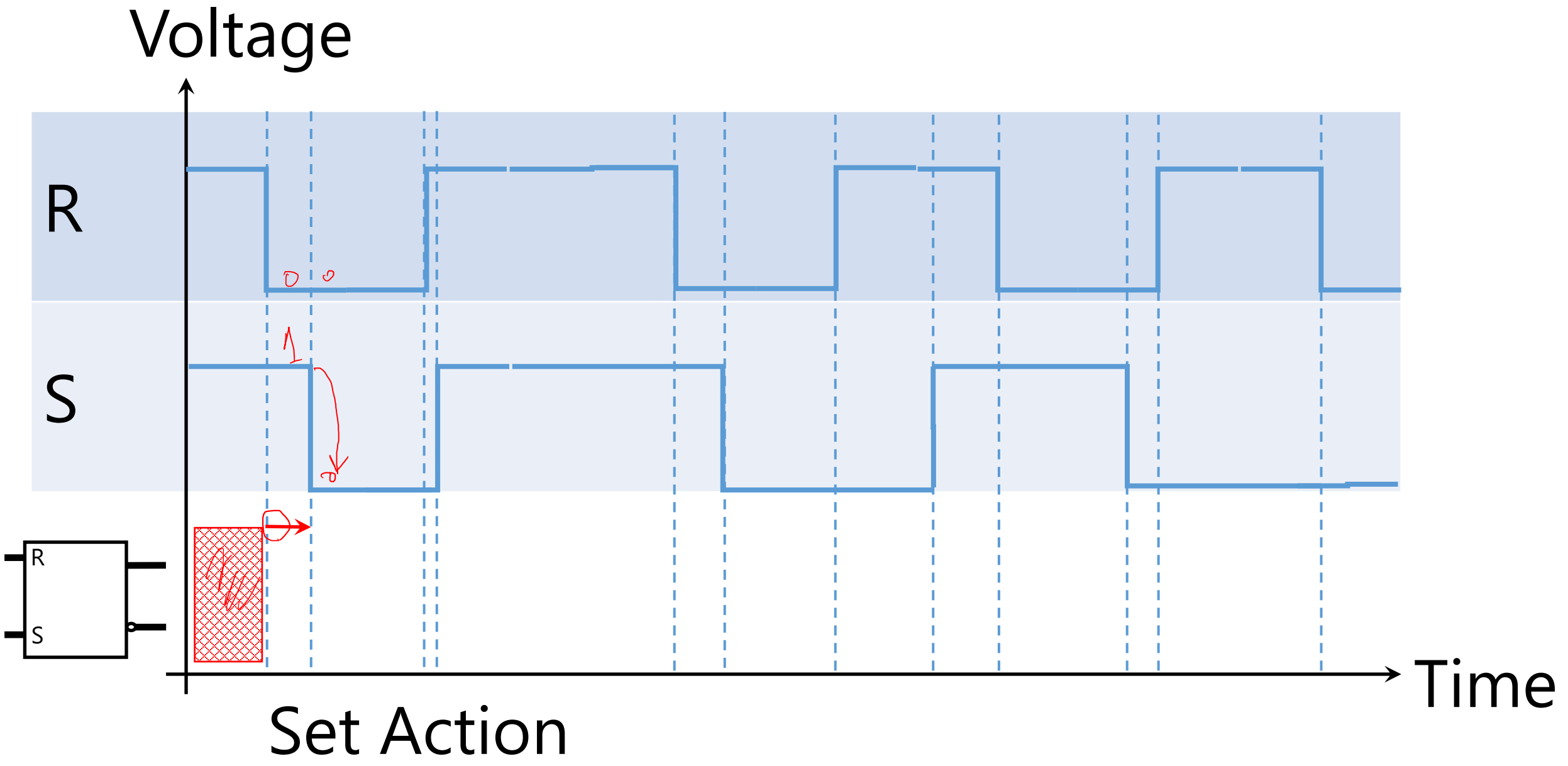
S	R	Q
0	0	Q_t
0	1	0
1	0	1
1	1	X

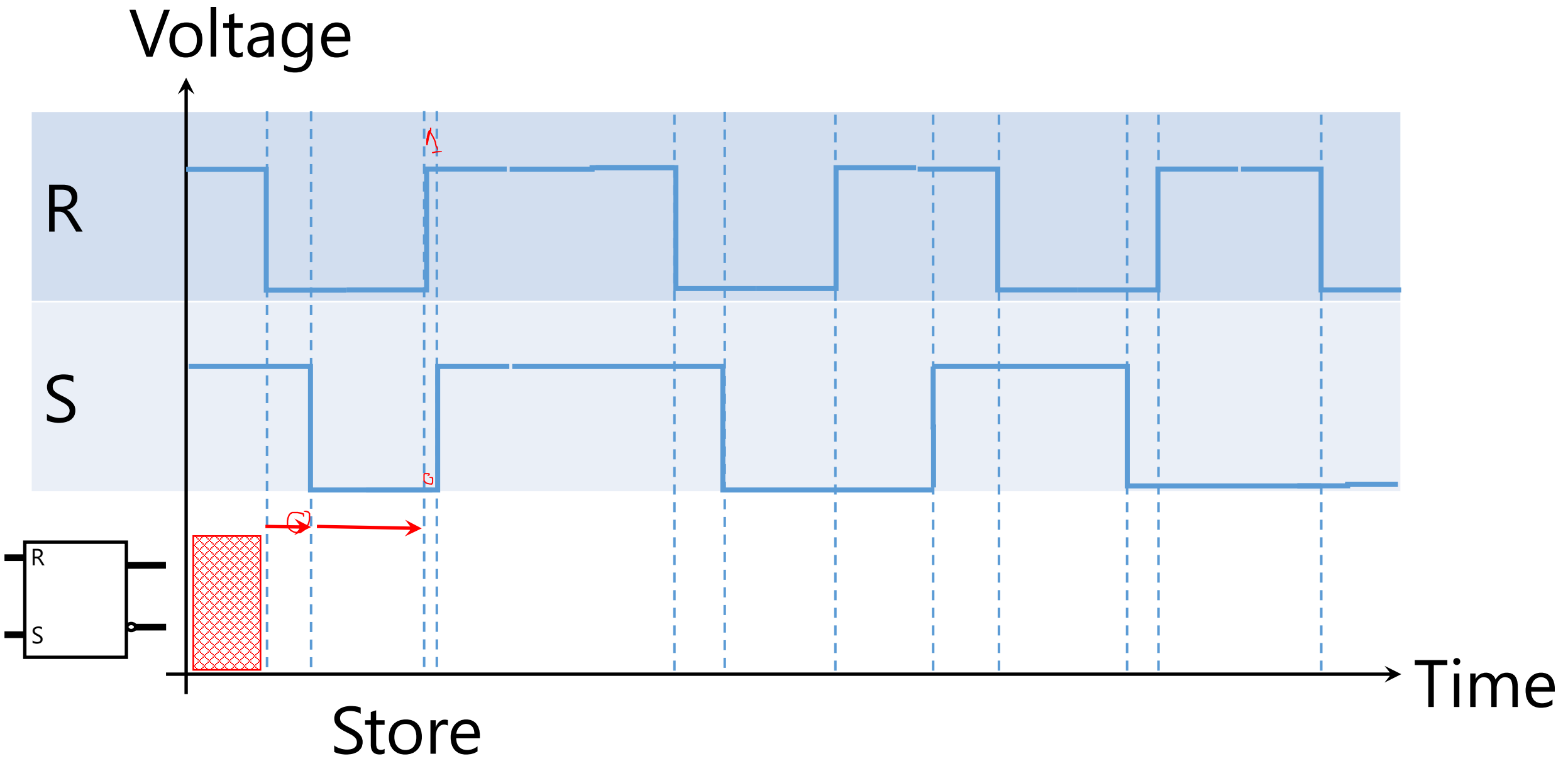


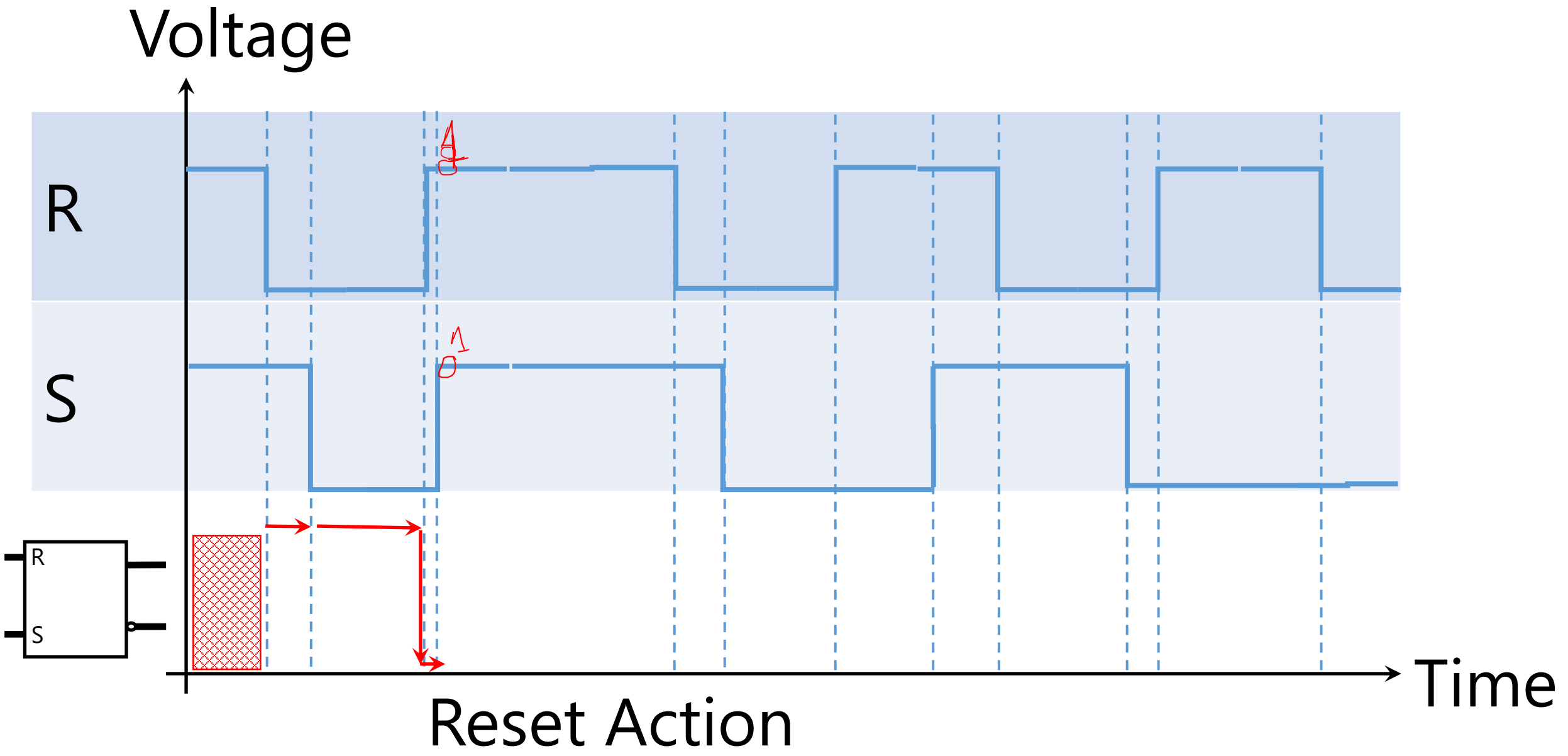

Voltage
R
S
R
S
Time

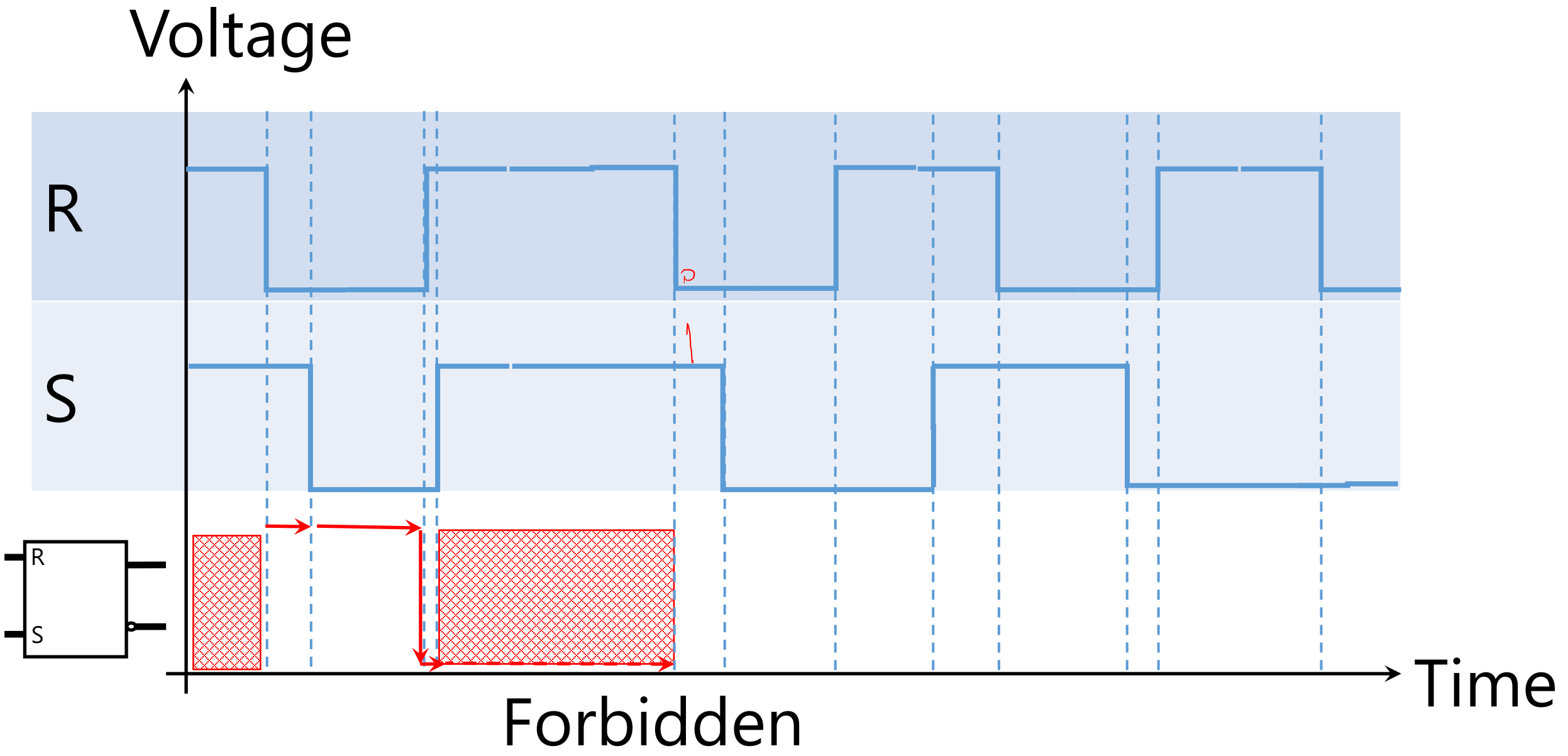

Voltage
R
S
R
S
Time

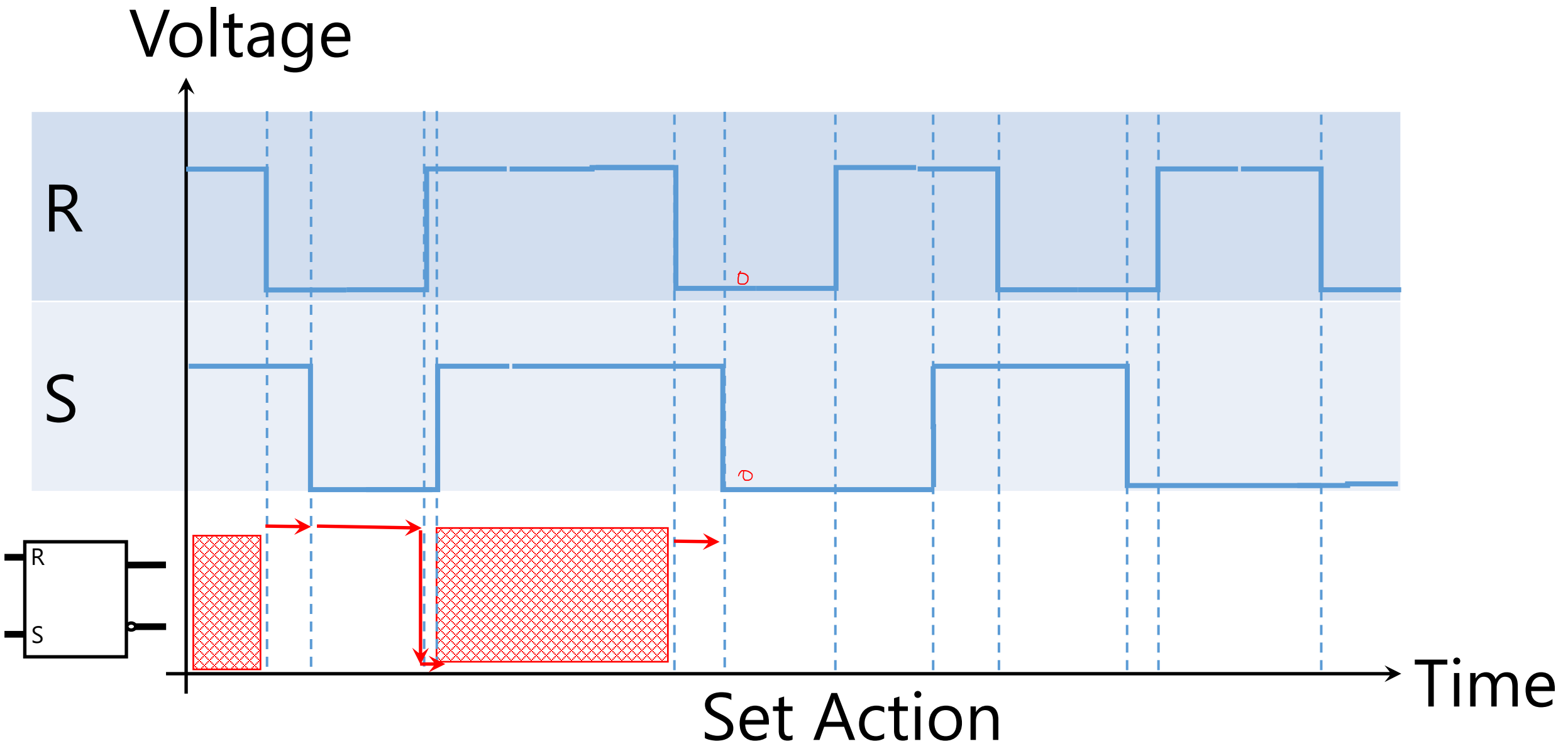

Voltage
R
S
1
1
R
S
Time


Voltage
R
S
R
S
Time
Set Action

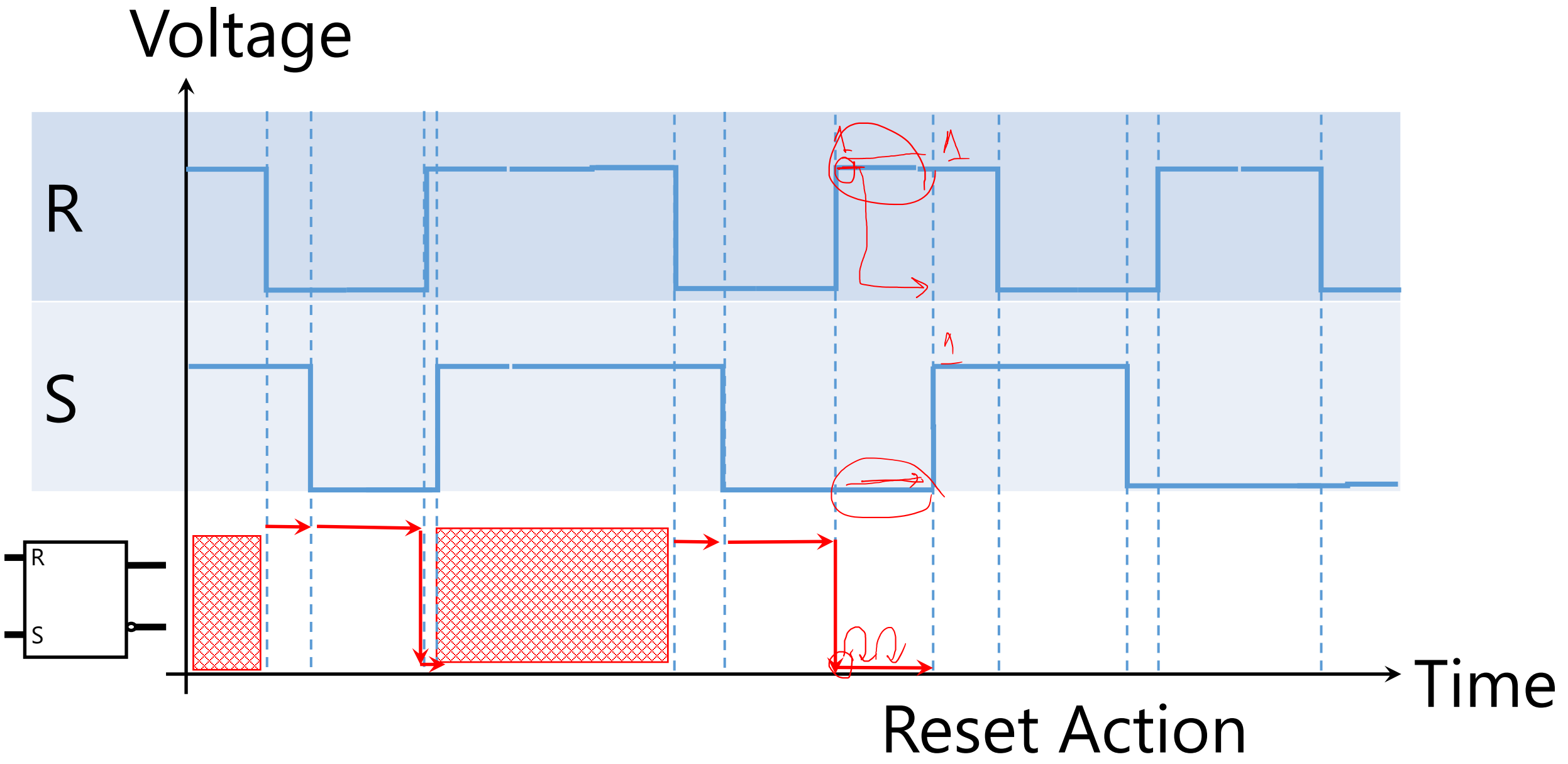

Voltage
R
S
R
S
Time
Store

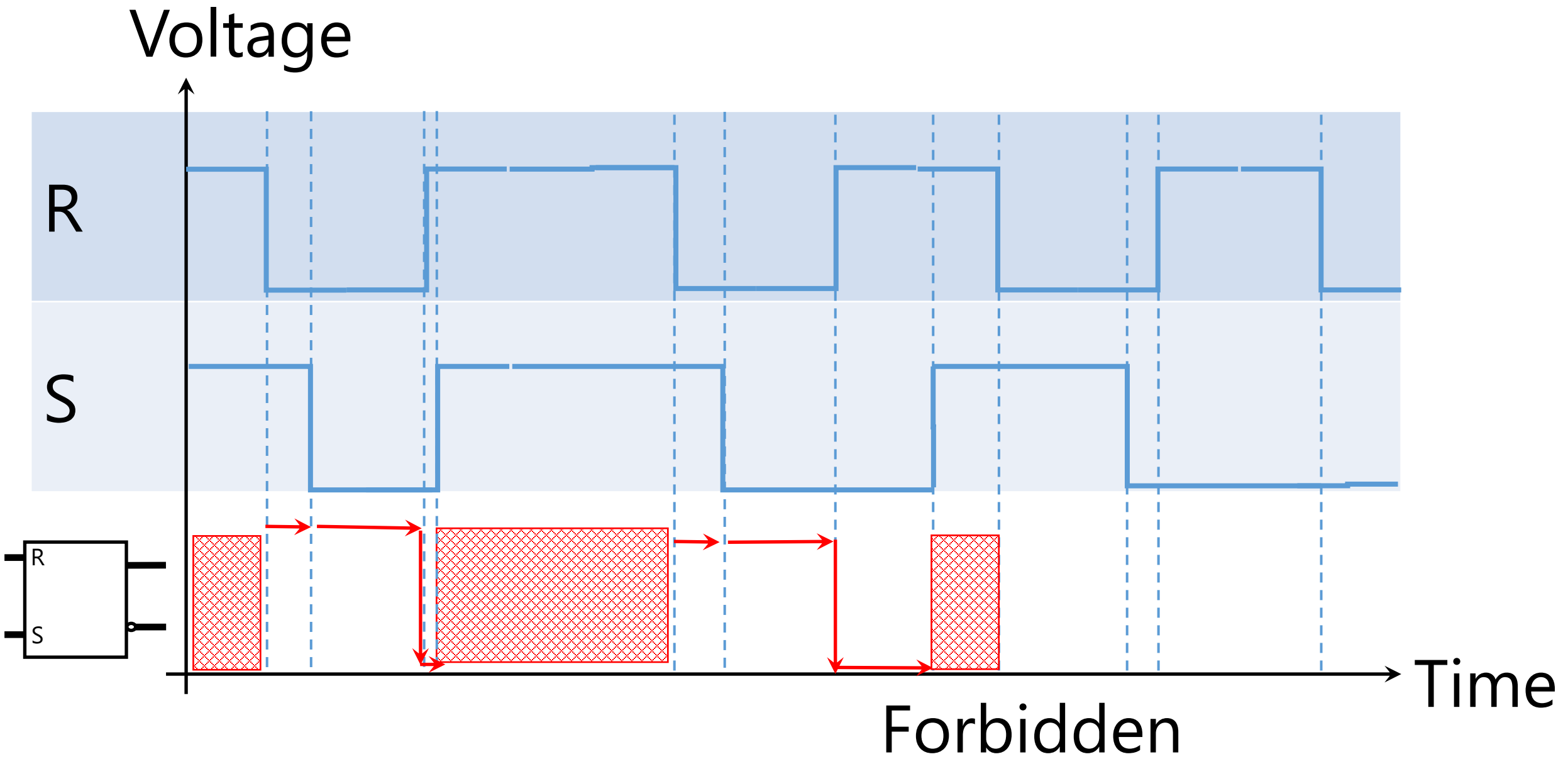

Voltage
R
S
R
S
Time
Reset Action

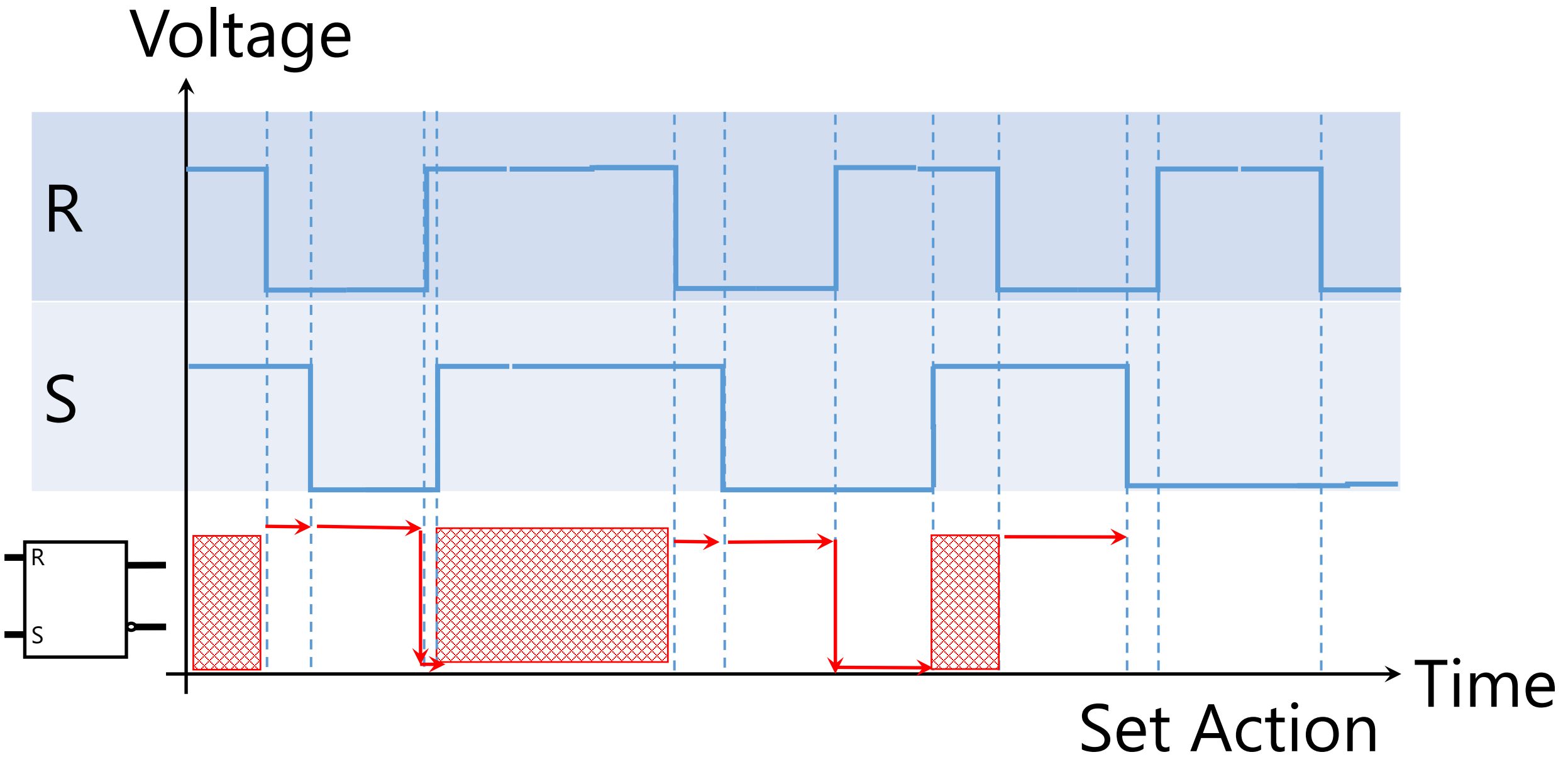

Voltage
R
S
R
S
Time
Forbidden

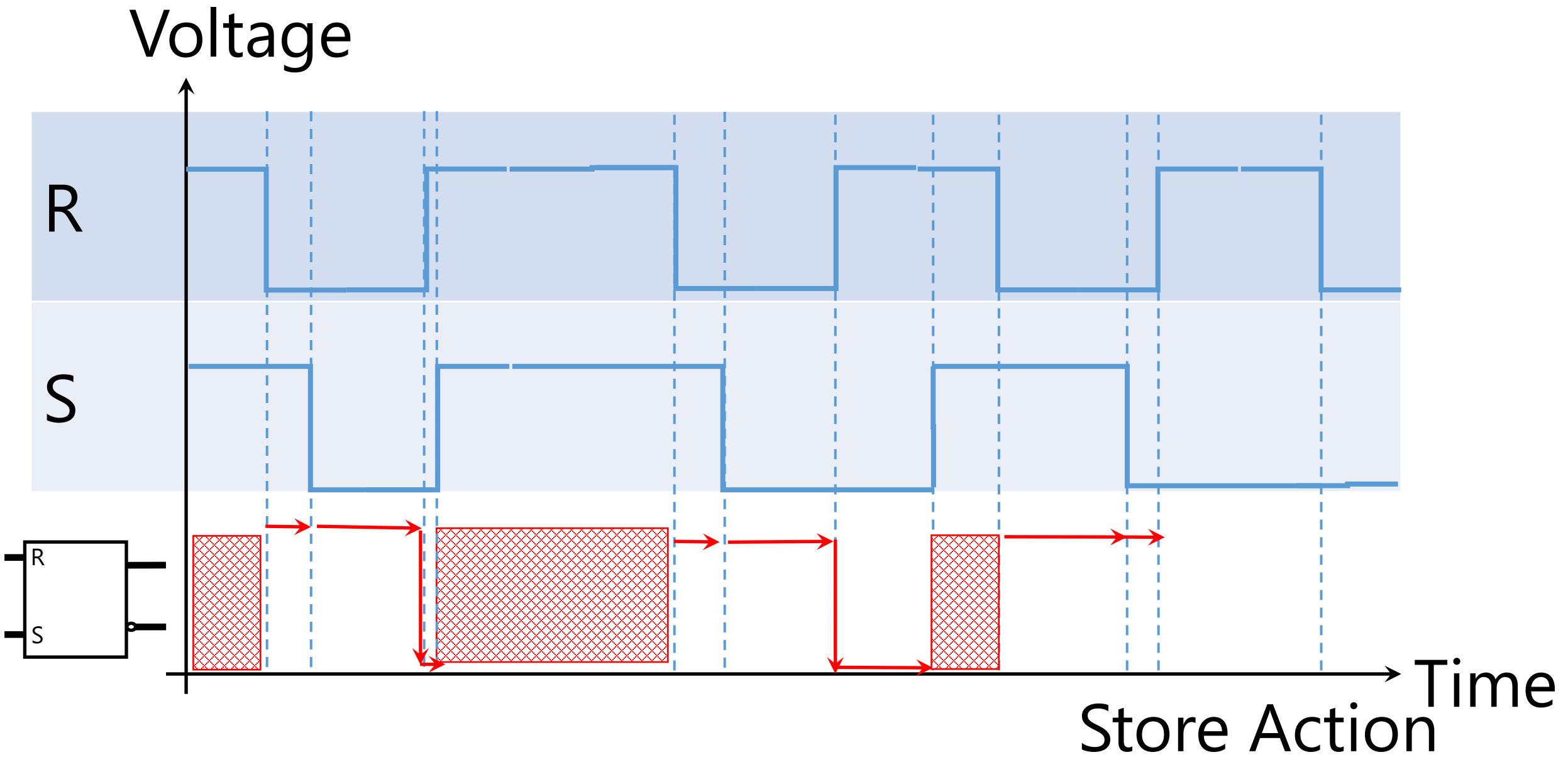

Voltage
R
S
R
S
Time
Set Action

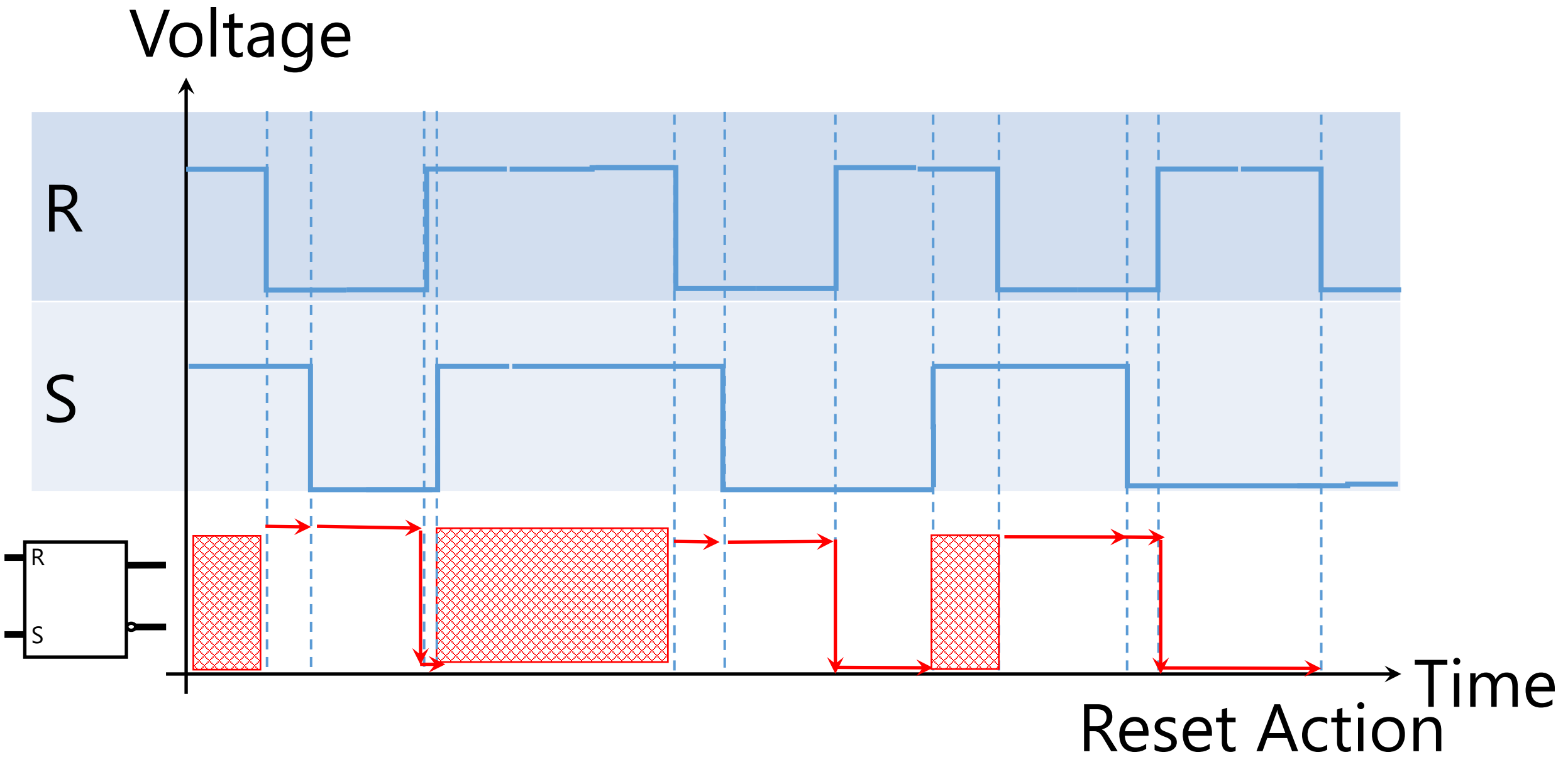

Voltage
R
S
R
S
Time
Store Action

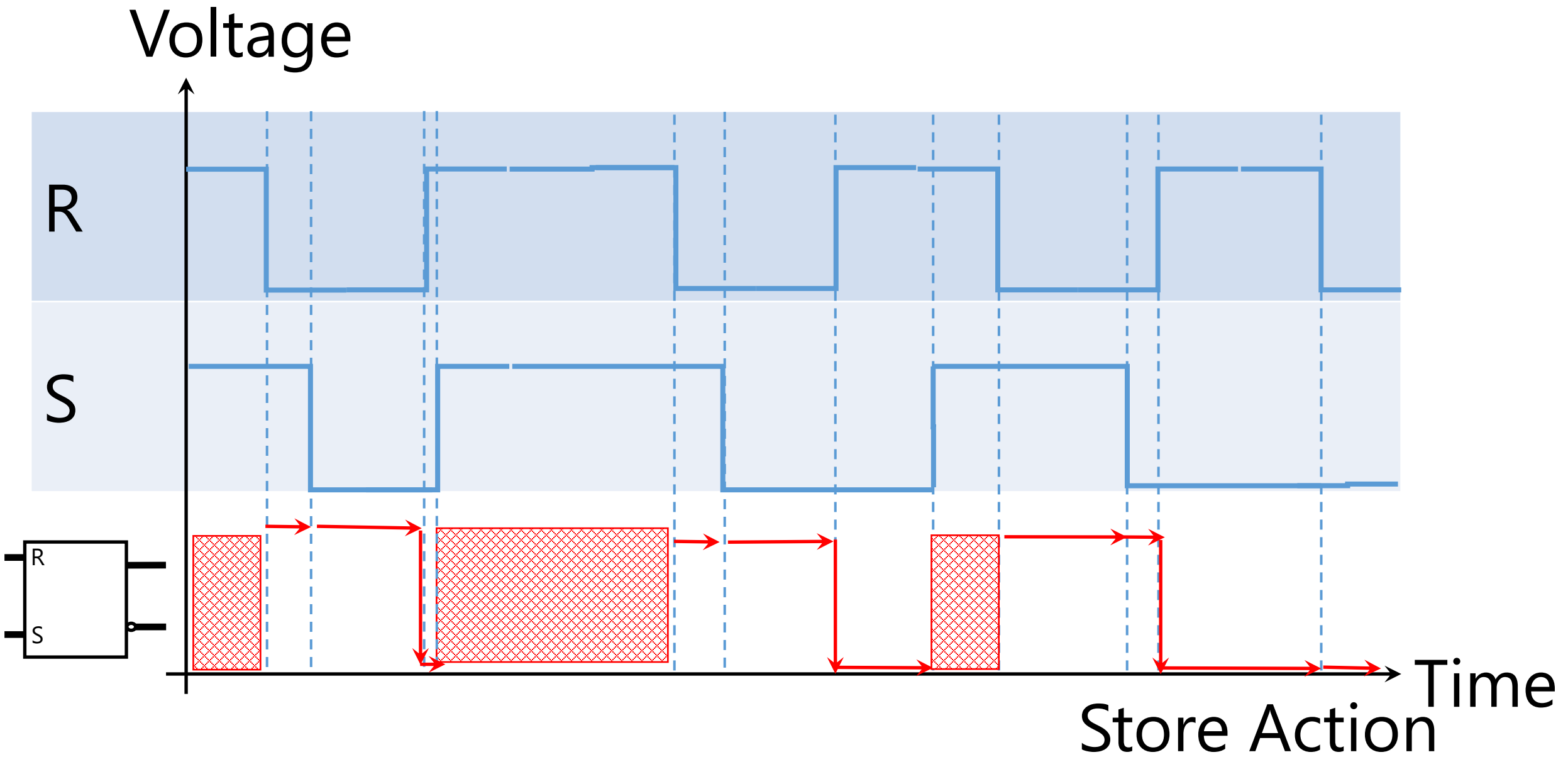

Voltage
R
S
R
S
Time
Reset Action


Voltage
R
S
R
S
Time
Forbidden


Voltage
R
S
R
S
Time
Set Action


Voltage
R
S
R
S
Time
Store Action

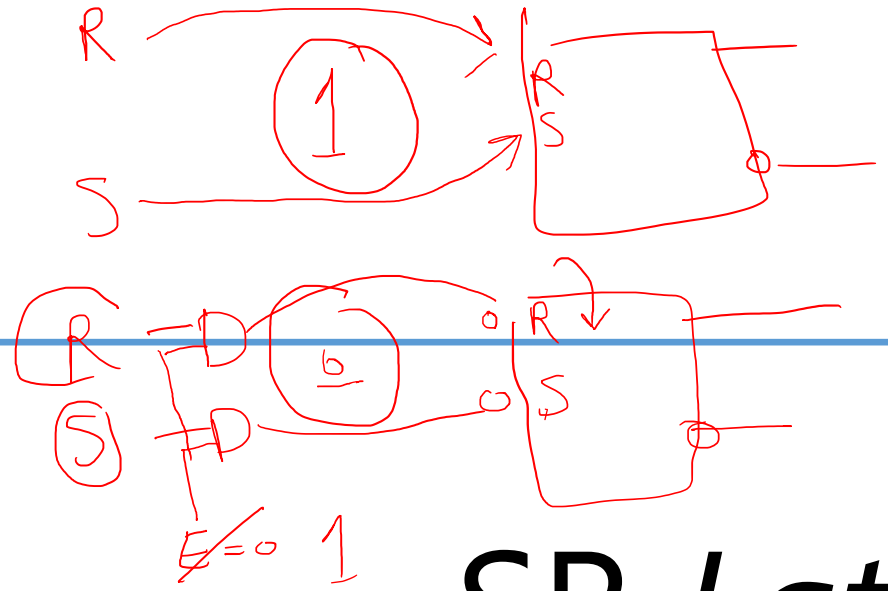

Voltage
R
S
R
S
Time
Reset Action


Voltage
R
S
R
S
Time
Store Action

S and R control how the state changes.

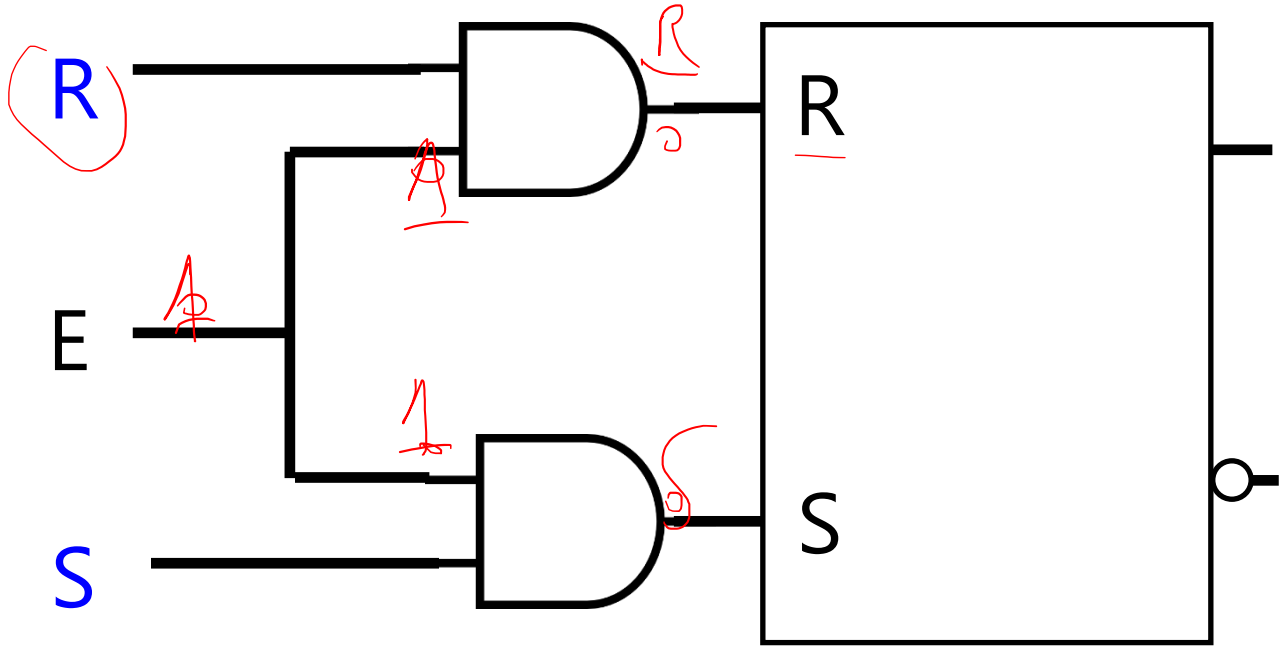
Put a gate/latch on when change applies

SR w/ enable input

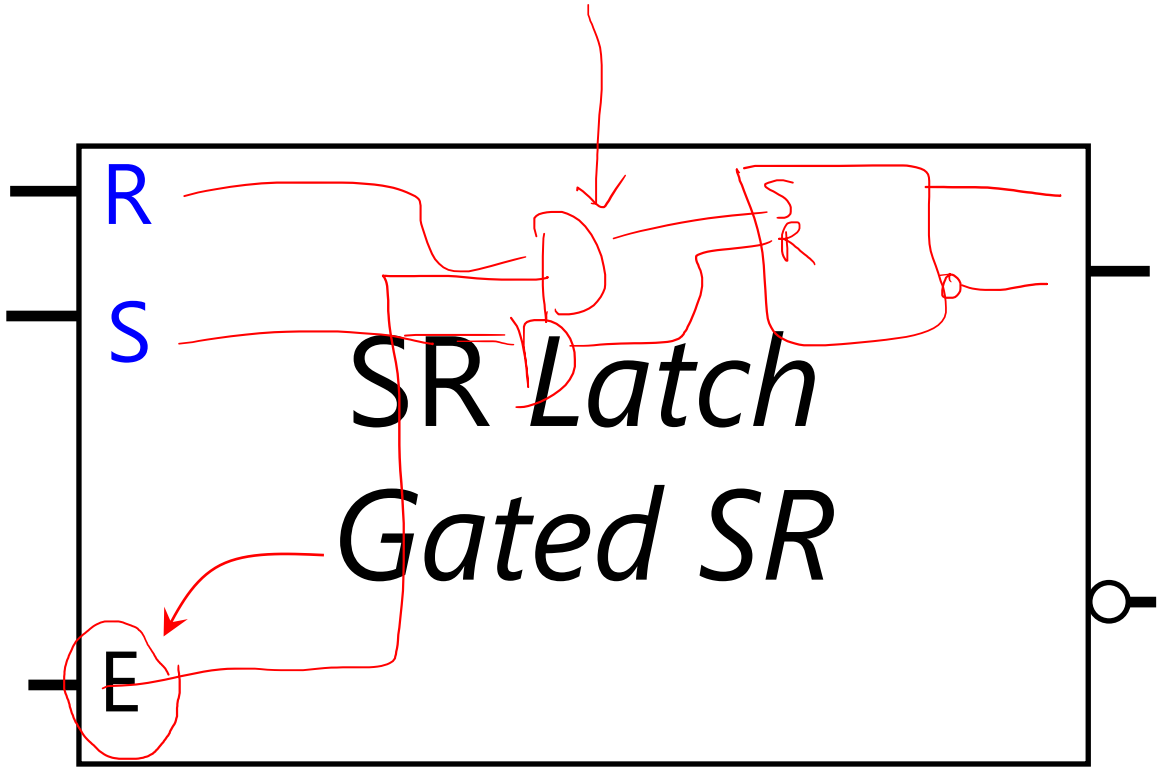


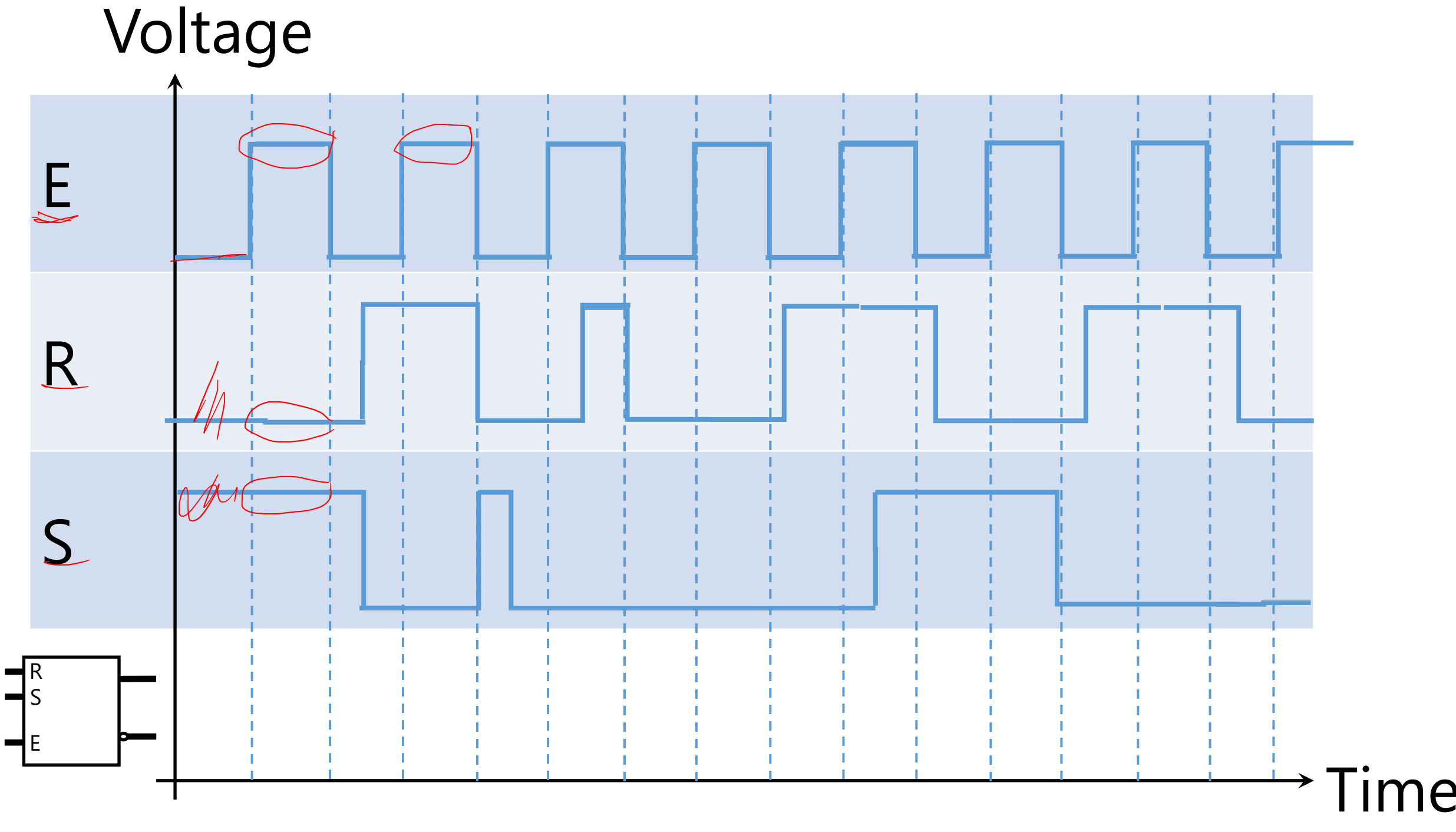
SR *Latch*

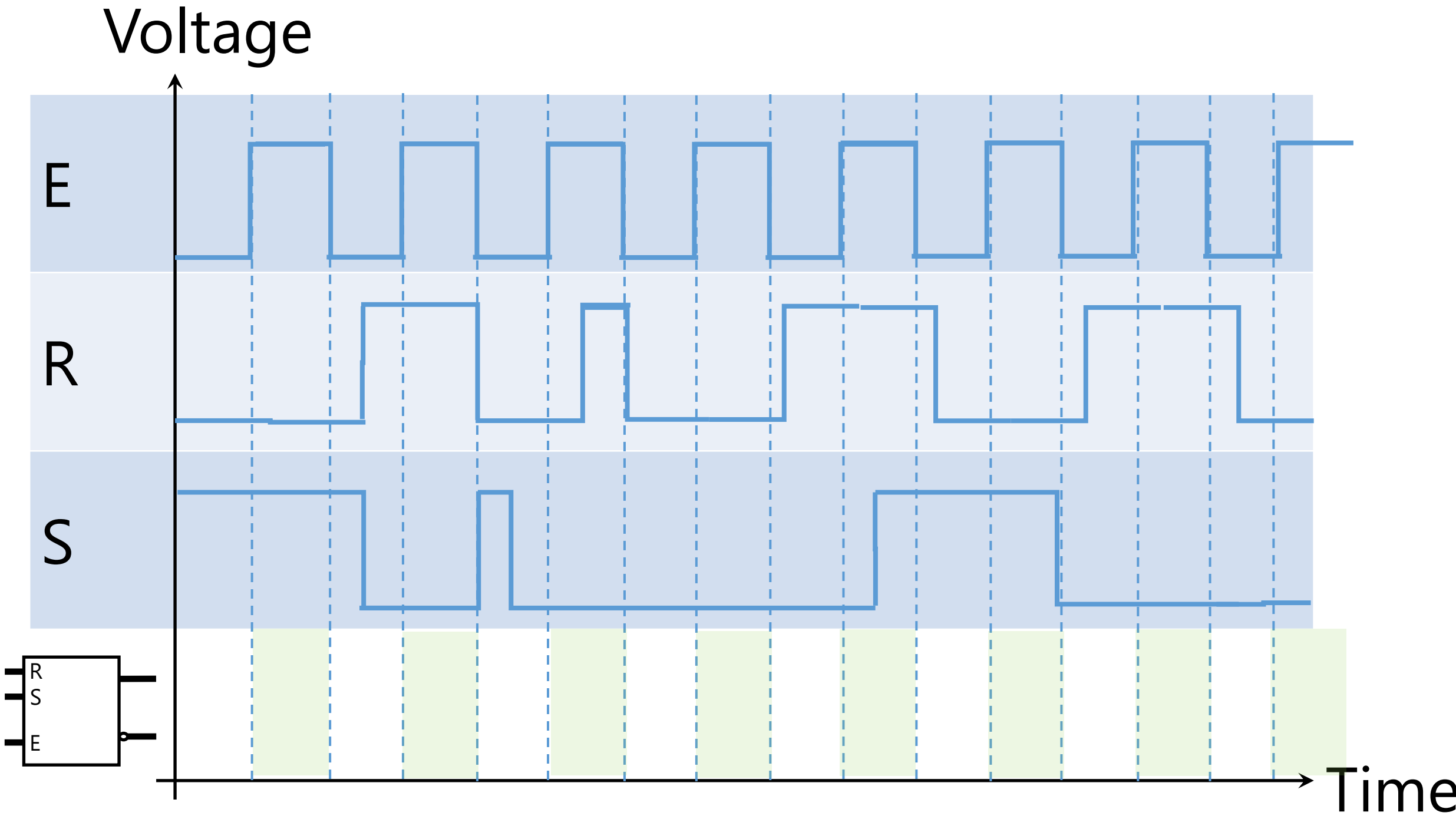
Gated SR

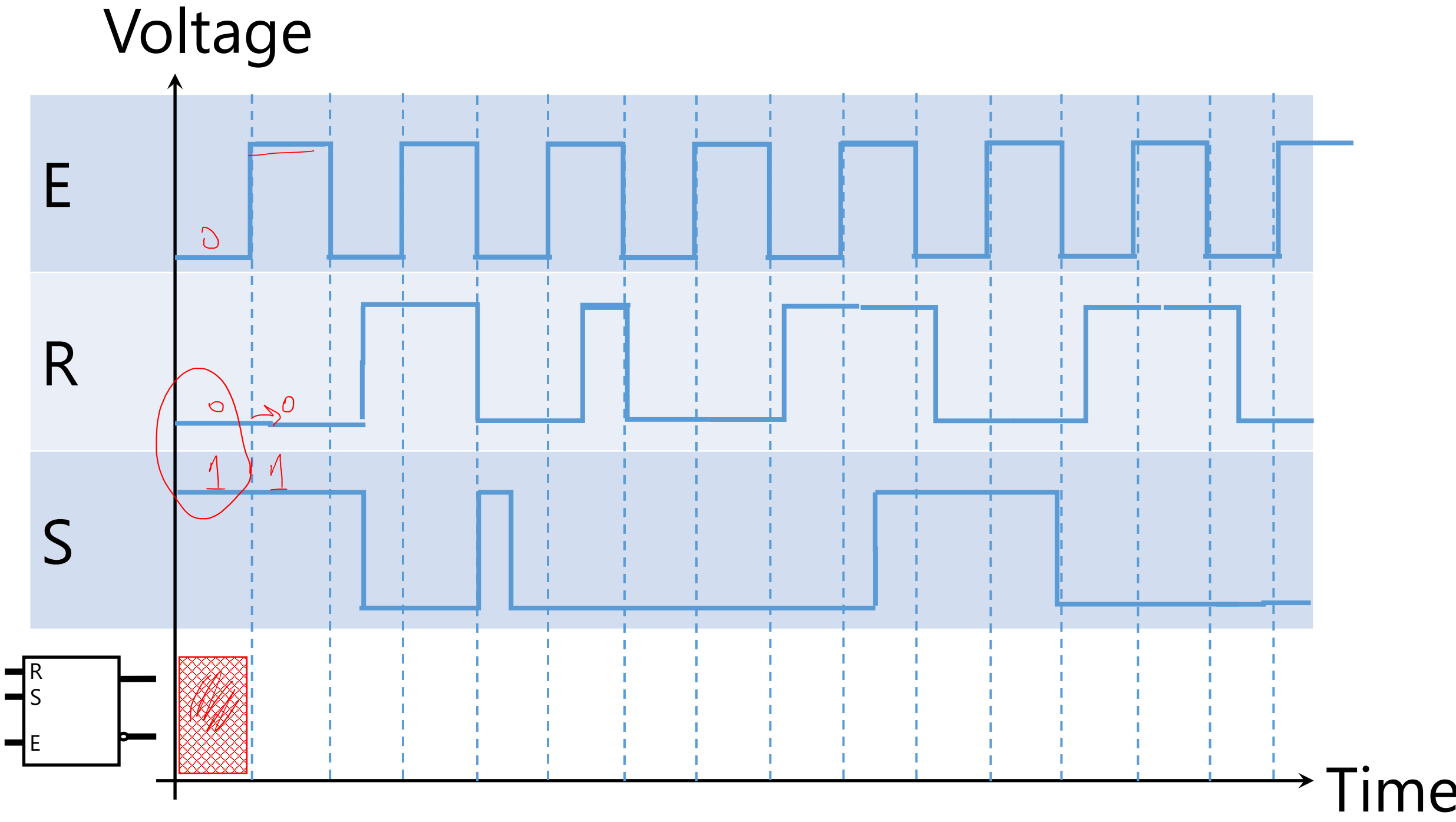


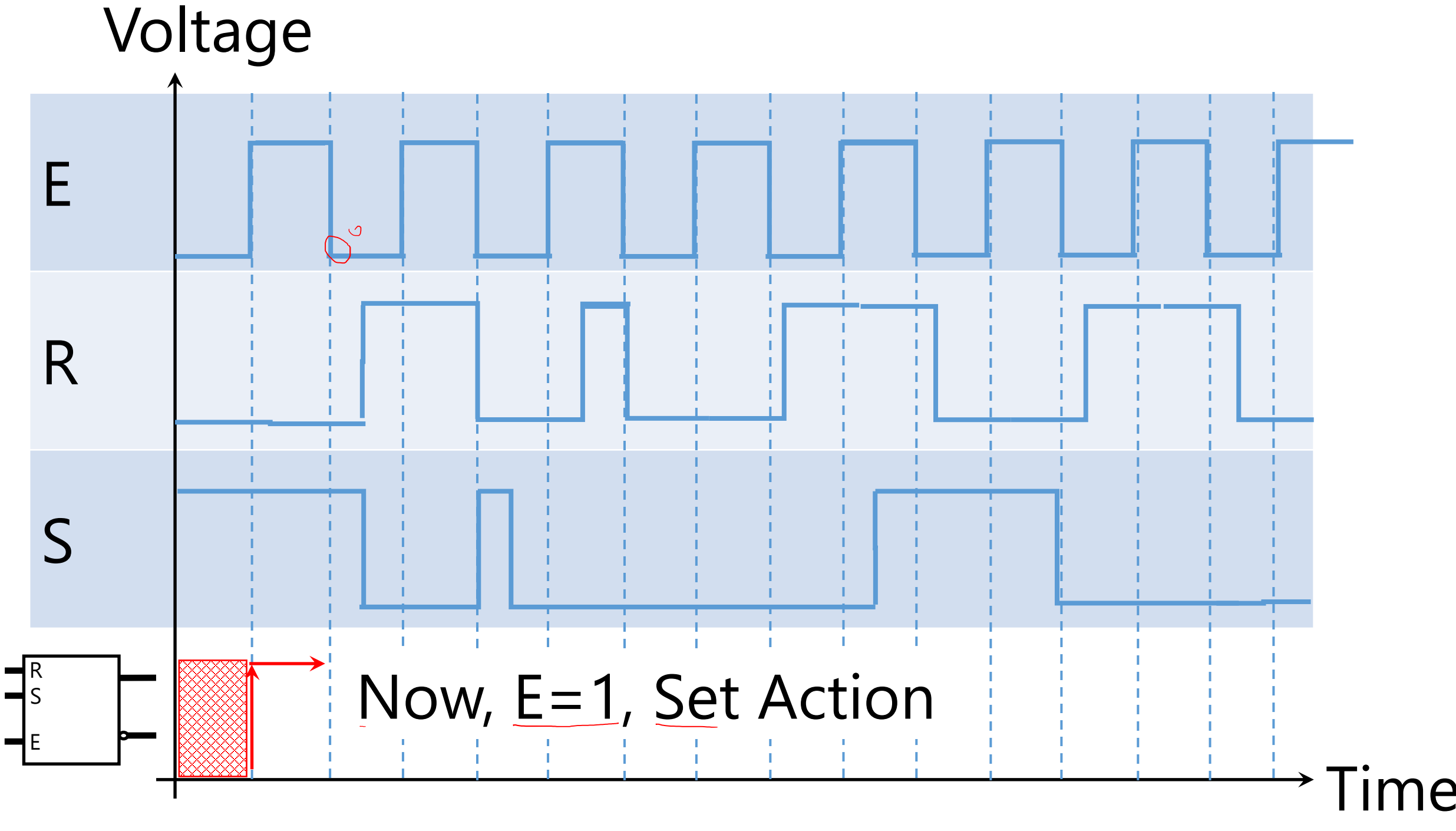
E	S	R	S	R	Q	Q′
0	X	X	0	0	Q_t	Q'_t
1	0	0	0	0	Q_t	Q'_t
1	0	1	0	1	0	1
1	1	0	1	0	1	0
1	1	1	1	1	X	X

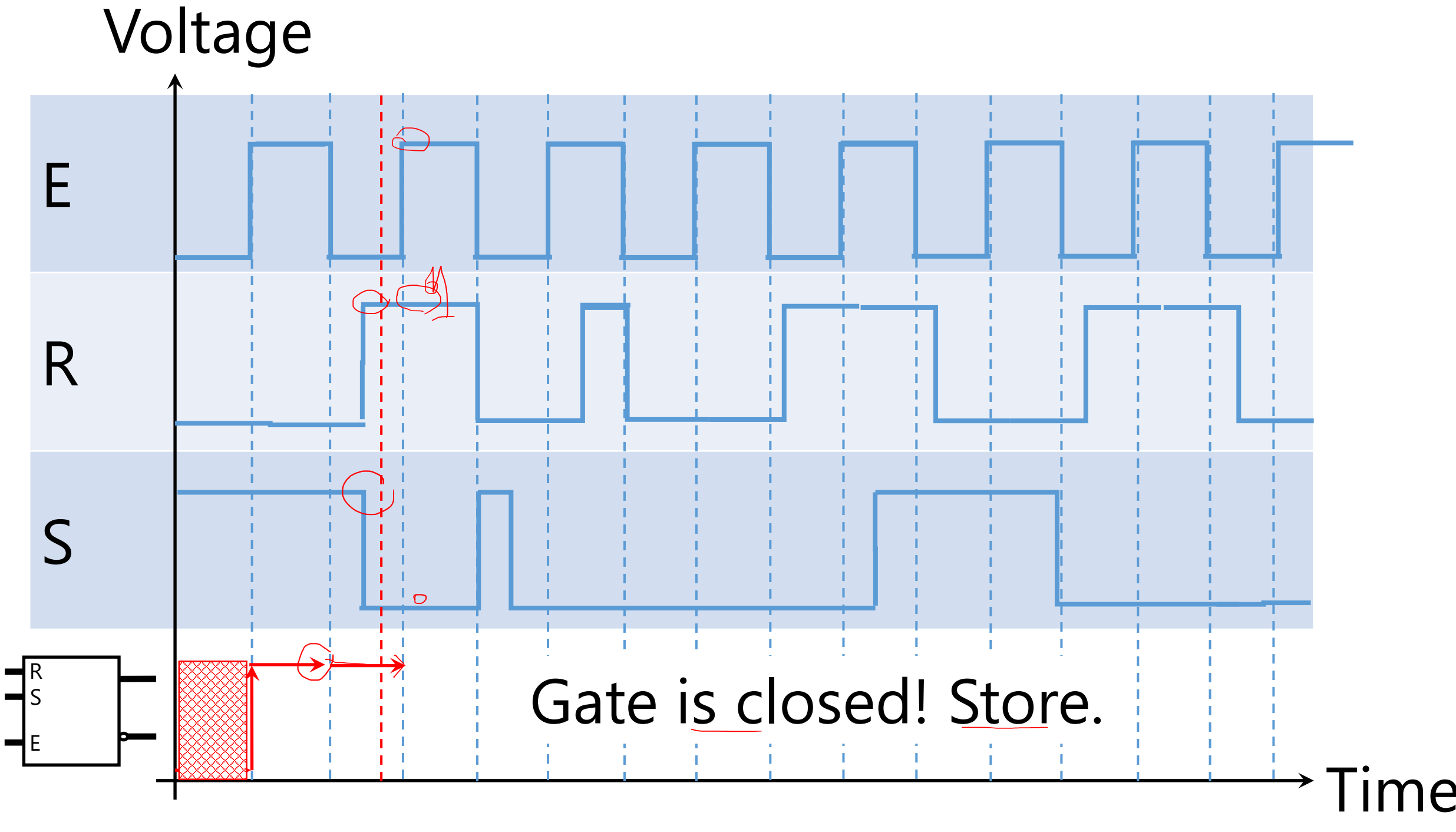

R
S
E
SR Latch
Gated SR

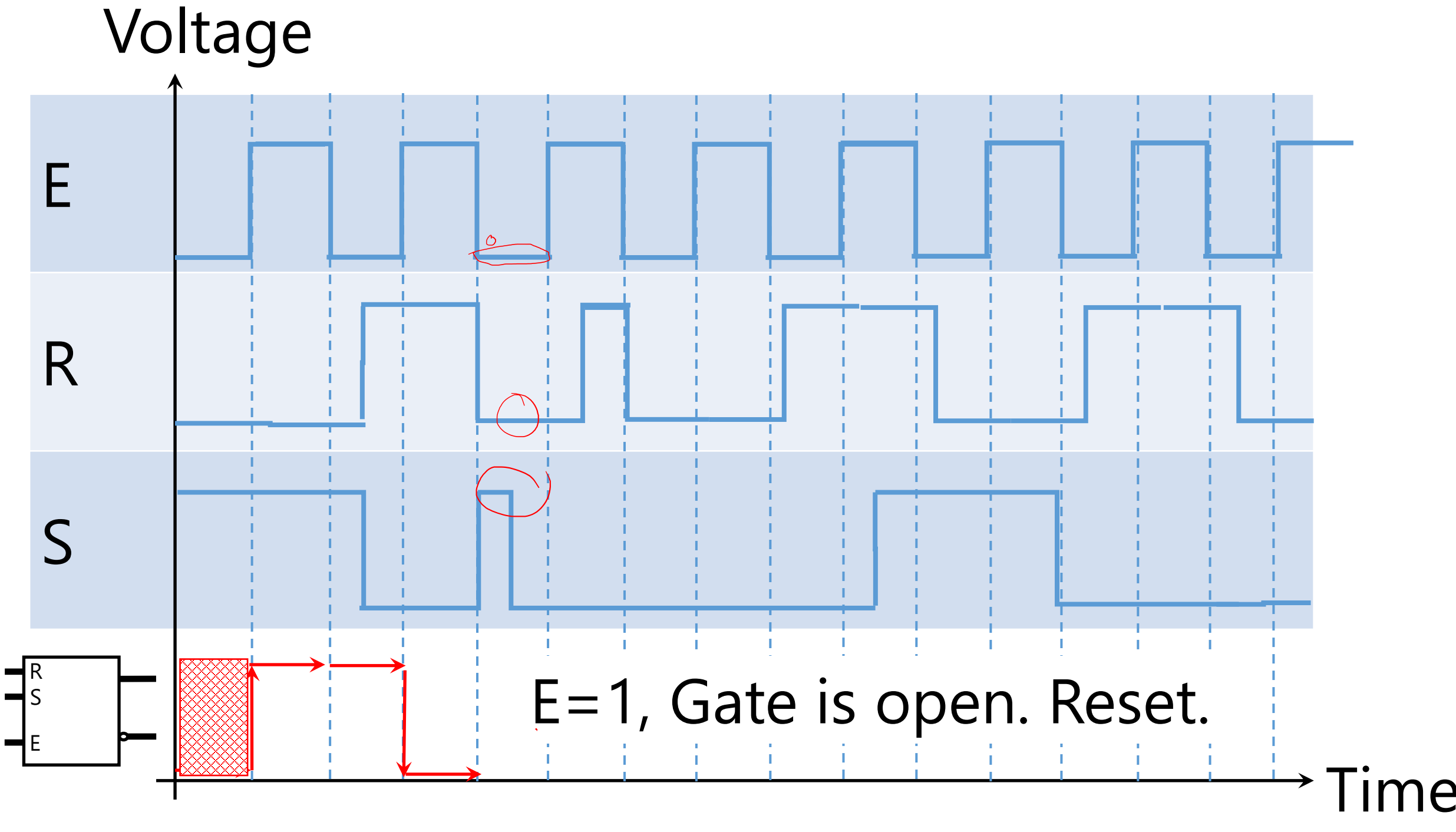

Voltage
E
R
S
R
S
E
Time

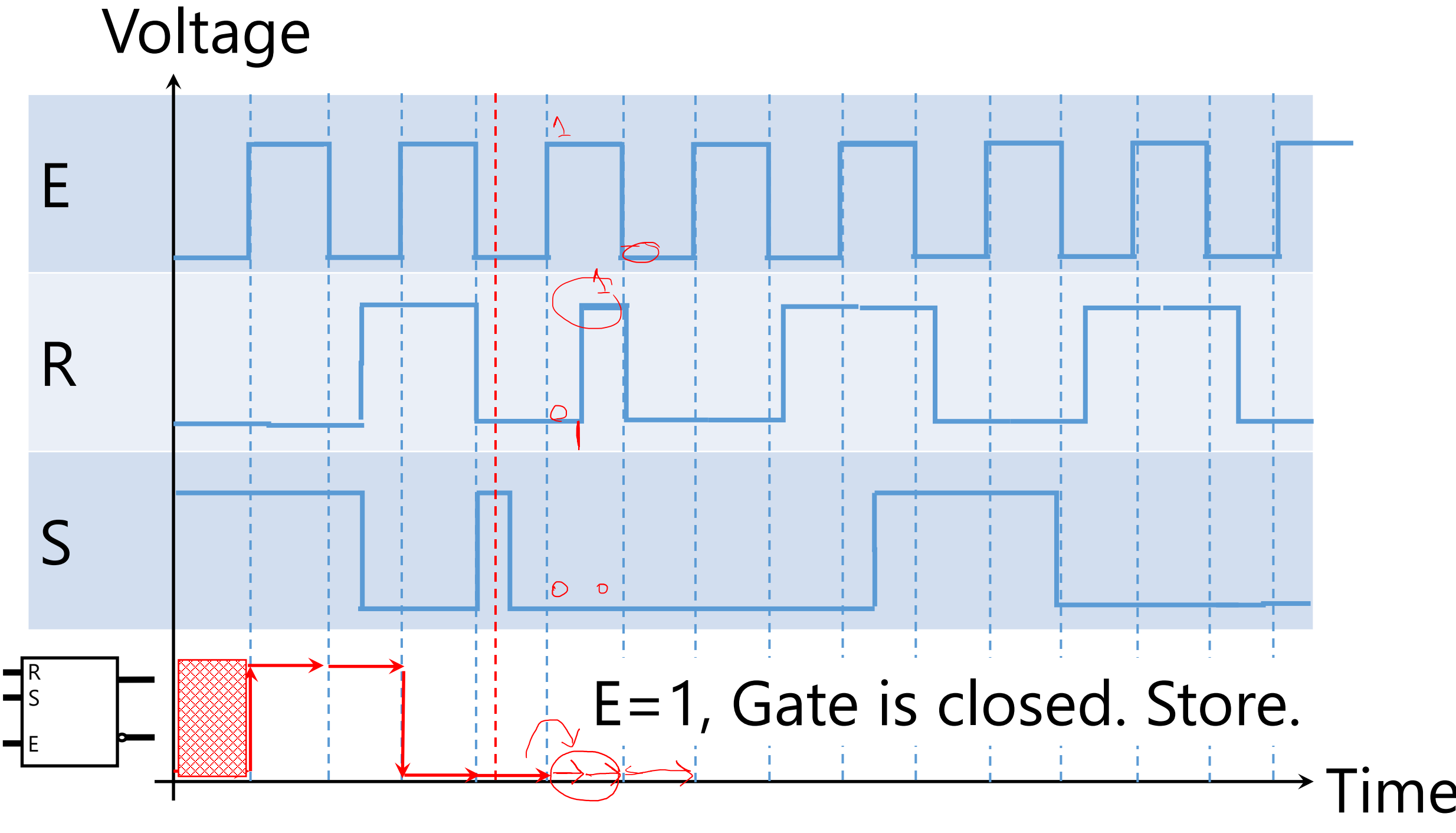

Voltage
E
R
S
R
S
E
Time


Voltage
E
R
S
0
0
0
1
1
R
S
E
Time


Voltage
E
R
S
R
S
E
Now, E=1, Set Action
Time

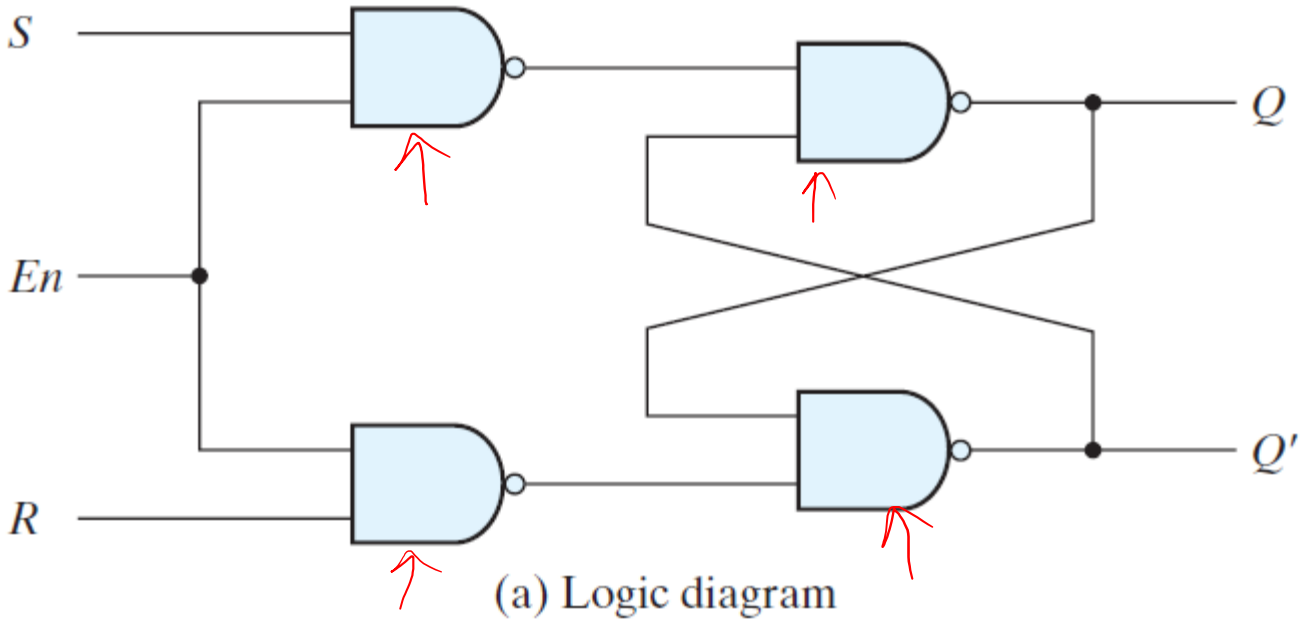

Voltage
E
R
S
R
S
E
Gate is closed! Store.
Time


Voltage
E
R
S
R
S
E
E=1, Gate is open. Reset.
Time


Voltage
E
R
S
R
S
E
E=1, Gate is closed. Store.
Time

SR *Latch*

NAND

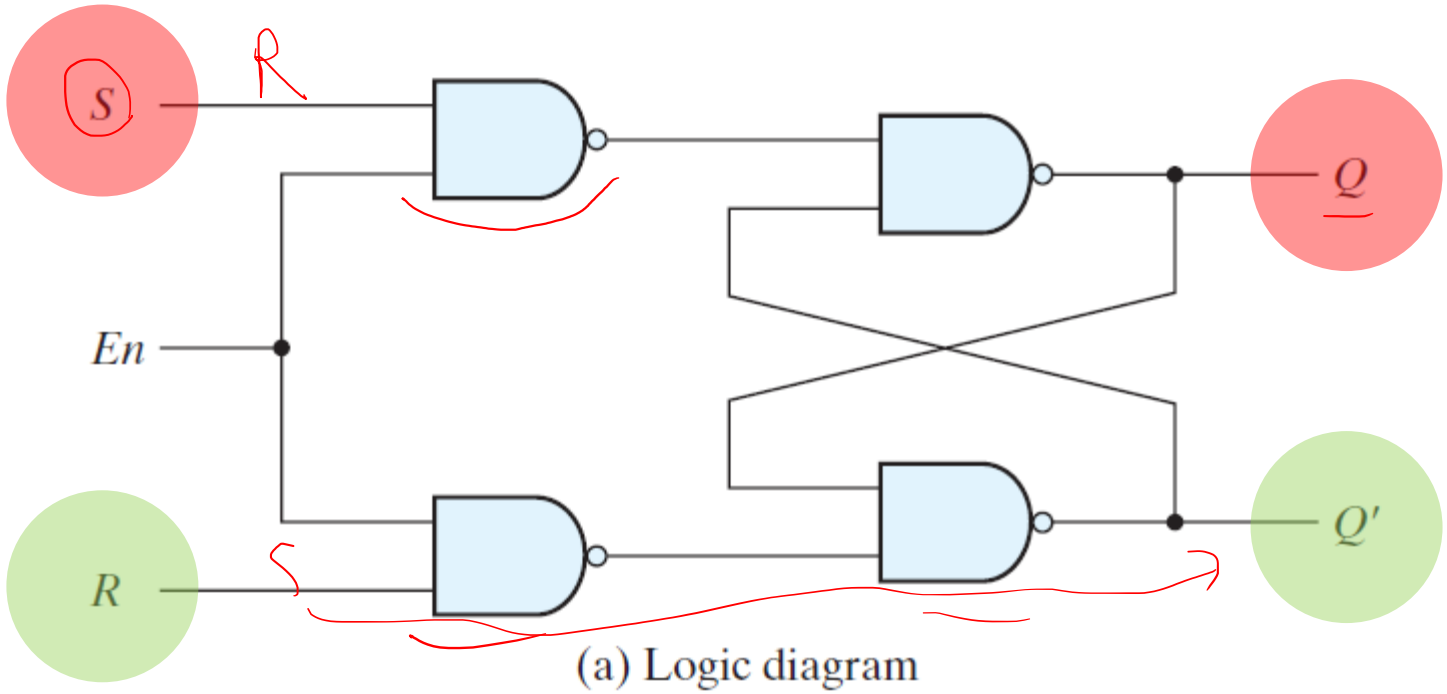


(a) Logic diagram

En	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

(b) Function table

FIGURE 5.5
SR latch with control input



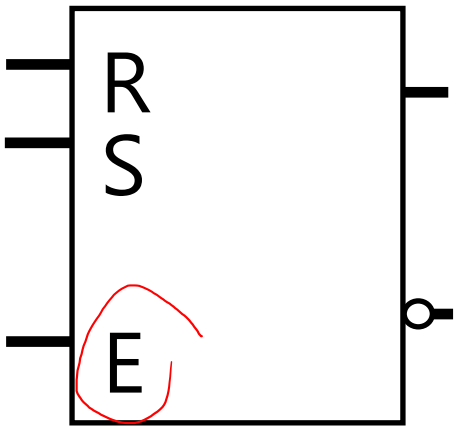
(a) Logic diagram

En	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

(b) Function table

FIGURE 5.5
SR latch with control input

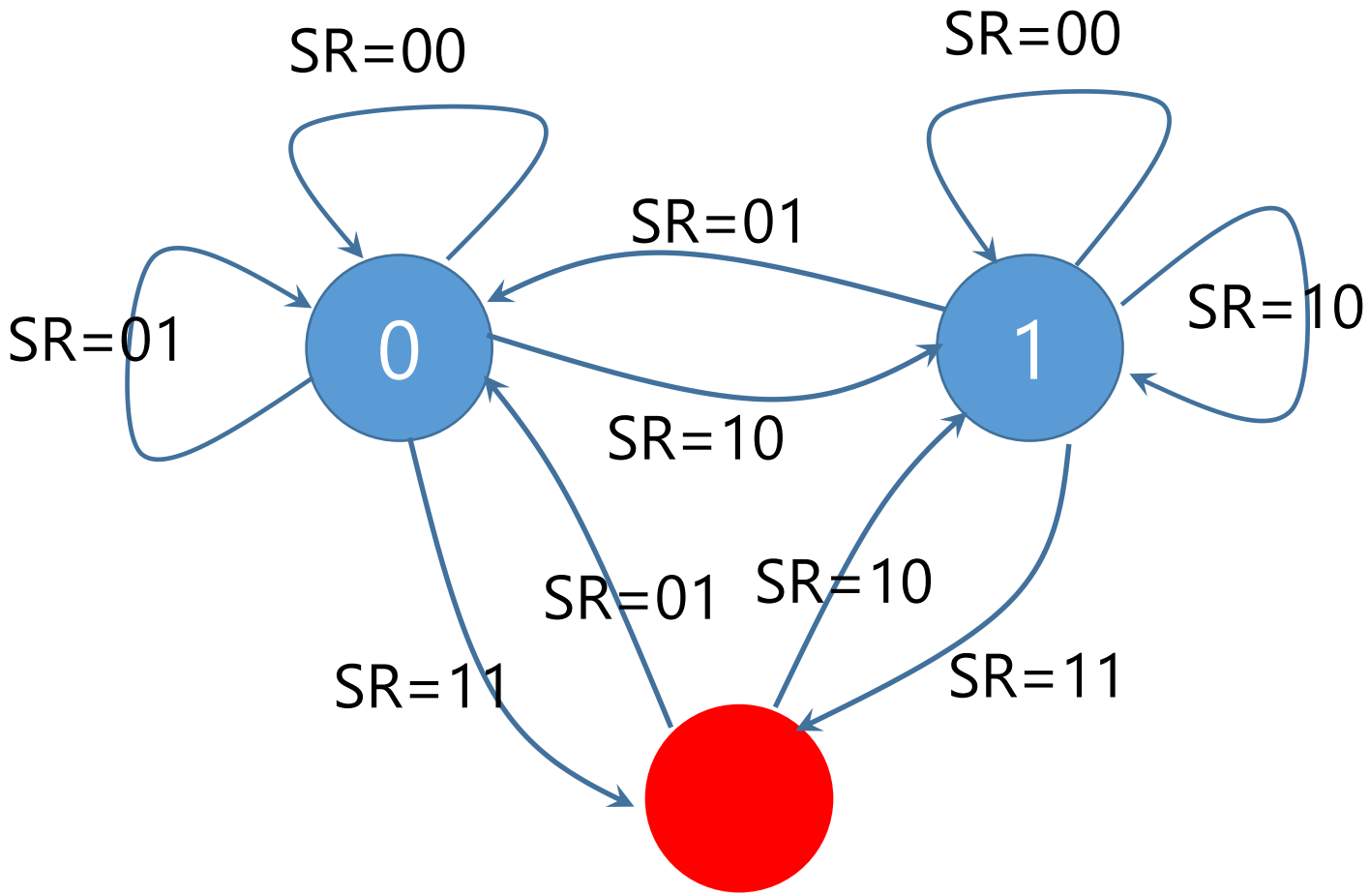
Block Diagram



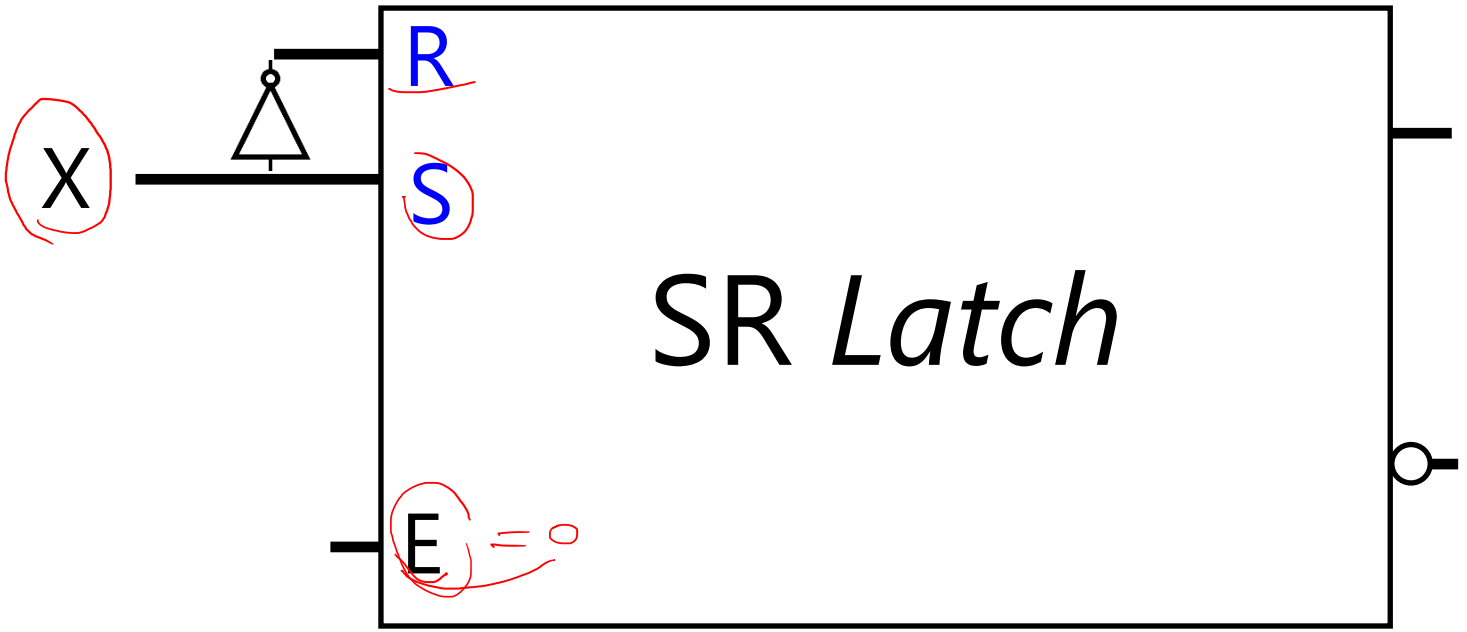
Characteristic Table

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

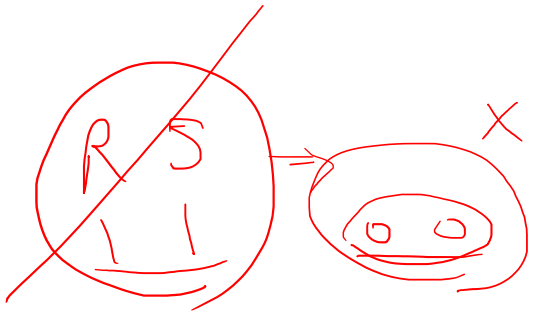
State Transition Diagram



Other *Latches*

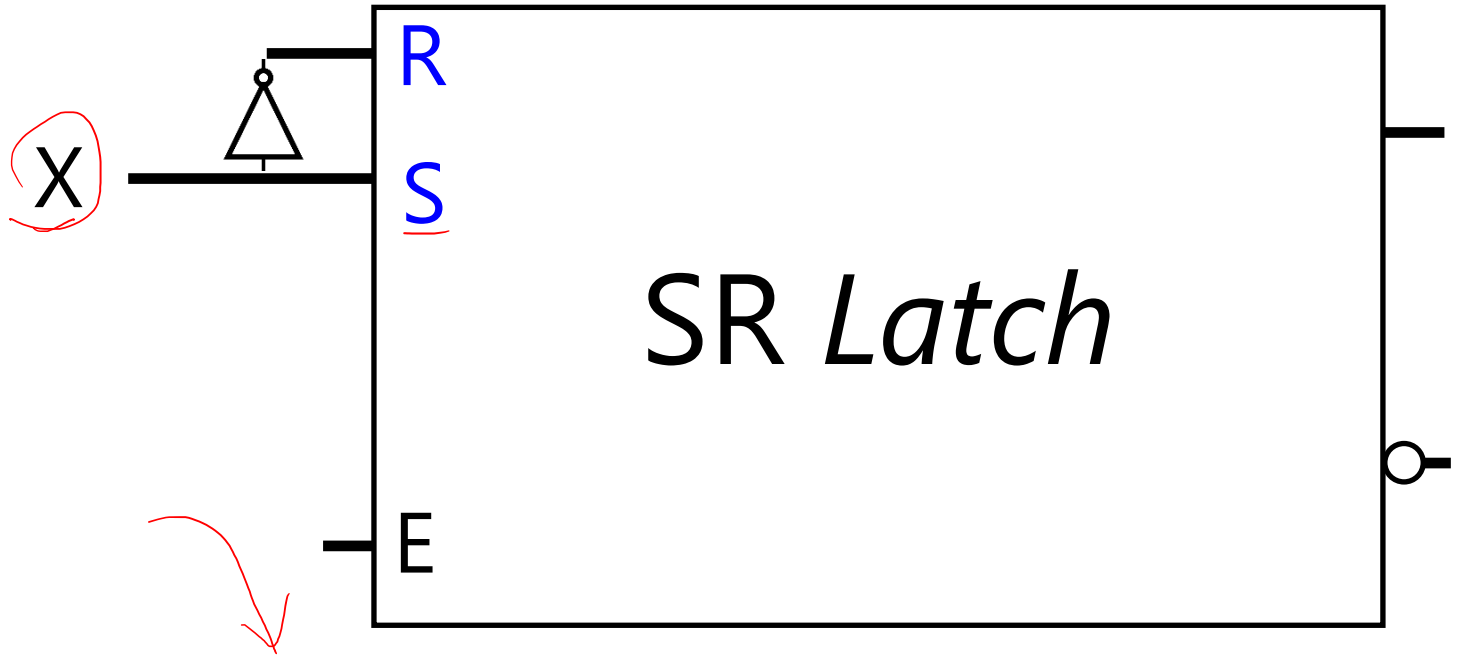


$R = S'$ $S = R'$

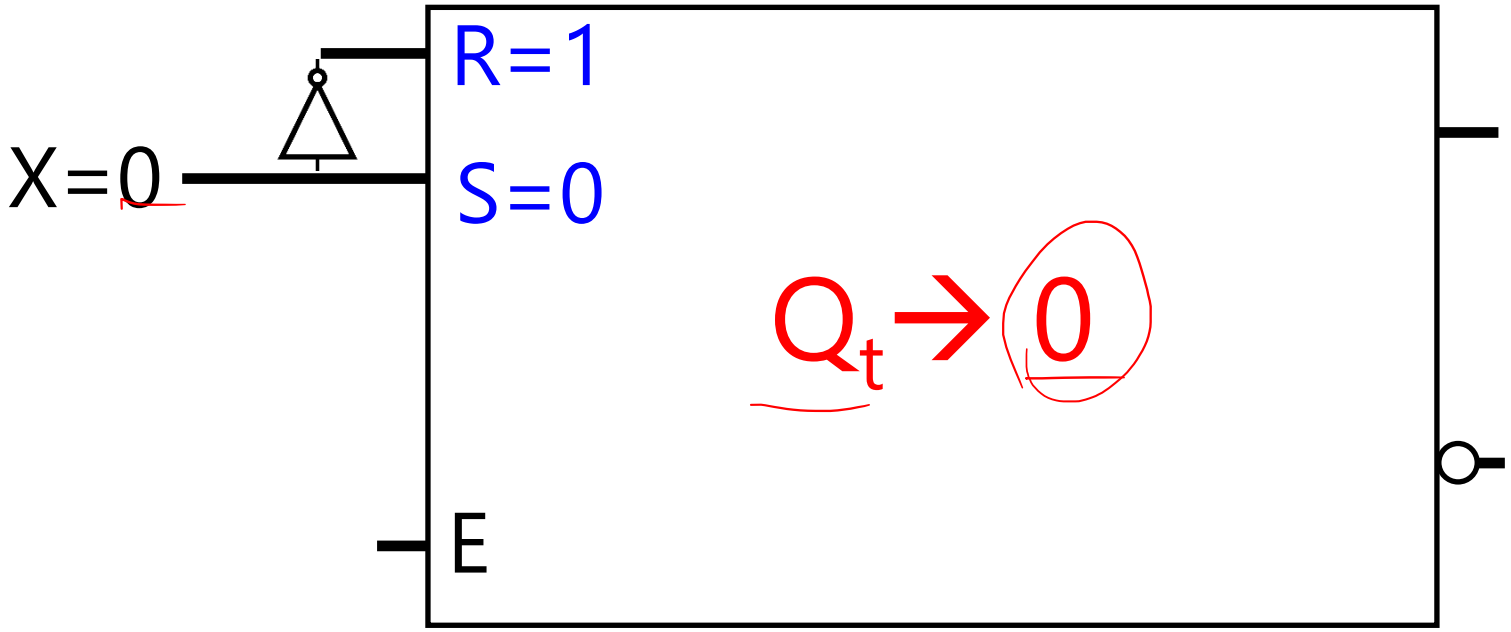


Let's avoid the forbidden action! $R \neq S$

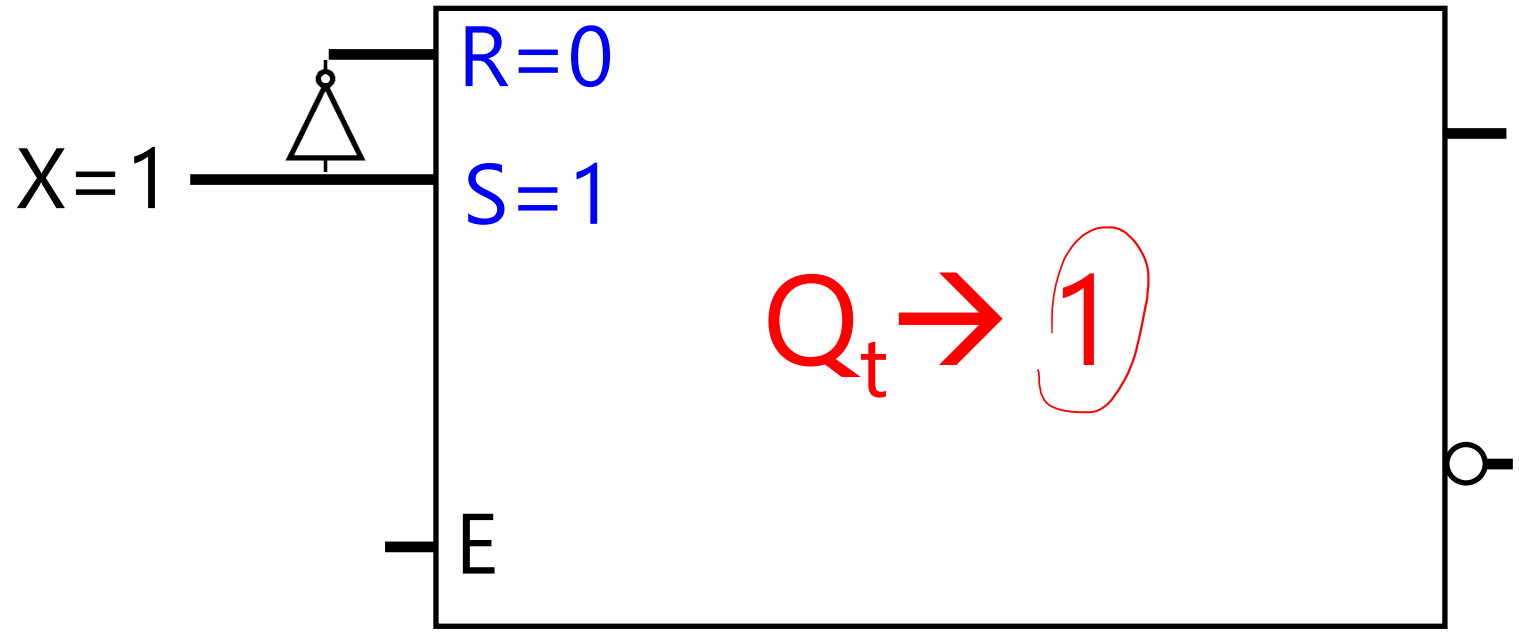
$R = S'$ and $S = R'$



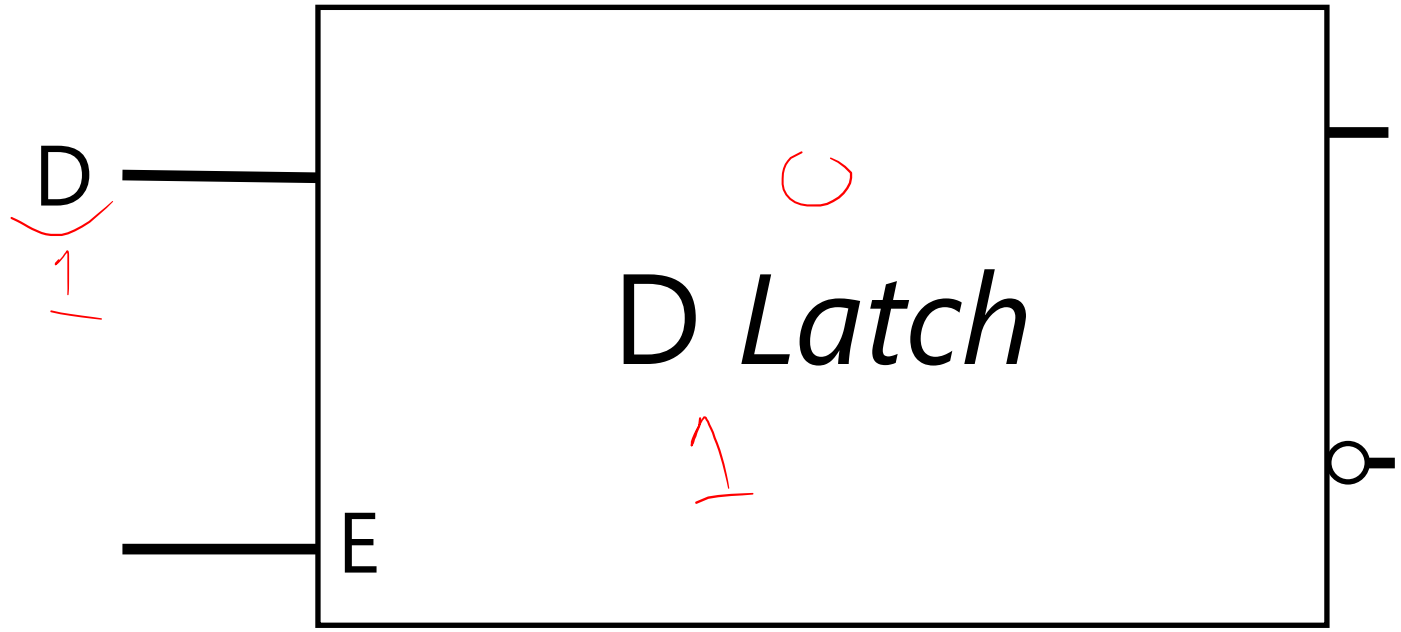
X	E	S	R	Q	Q′
0	0	x	x	Q_t	Q'_t
1	0	x	x	Q_t	Q'_t
0	1	0	1	0	1
1	1	1	0	1	0
Never happens		1	1	x	x



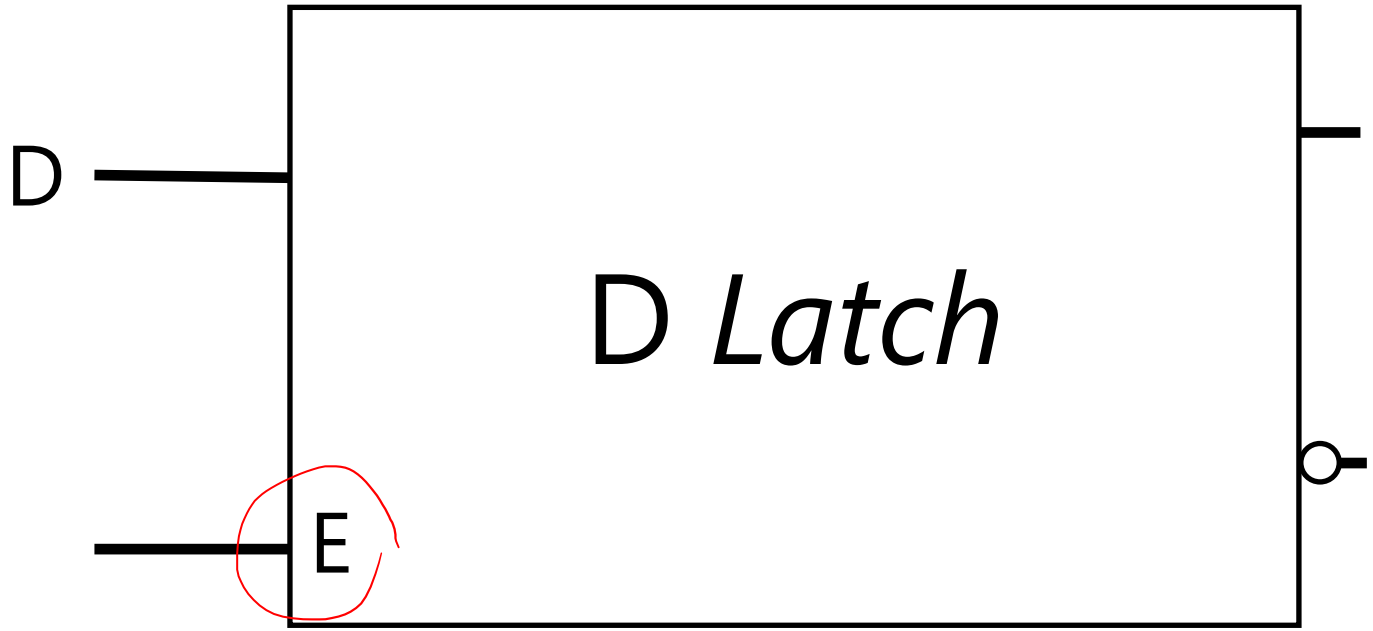
X	E	S	R	Q	Q′
0	0	X	X	Q_t	Q'_t
1	0	X	X	Q_t	Q'_t
0	1	0	1	0	1
1	1	1	0	1	0
Never happens		1	1	X	X



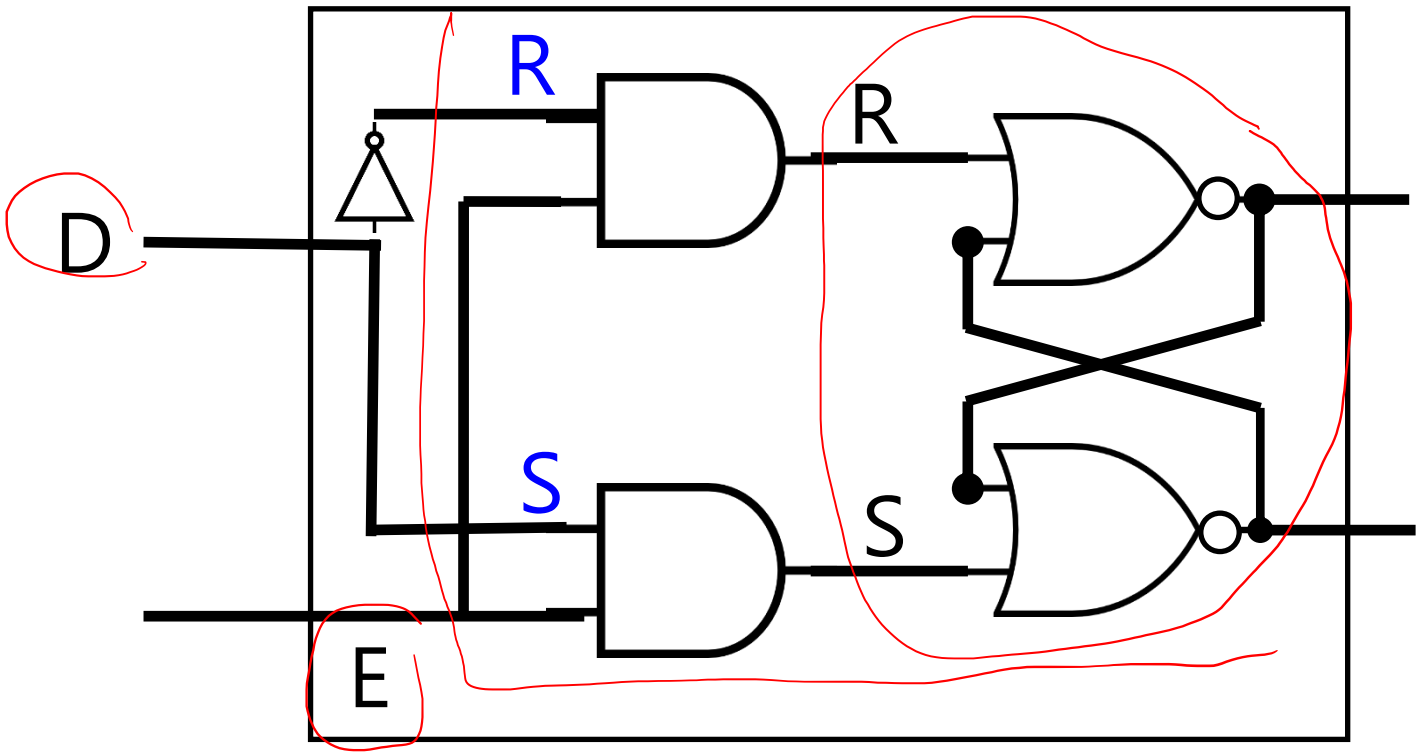
X	E	S	R	Q	Q′
0	0	x	x	Q_t	Q'_t
1	0	x	x	Q_t	Q'_t
0	1	0	1	0	1
1	1	1	0	1	0
Never happens		1	1	x	x



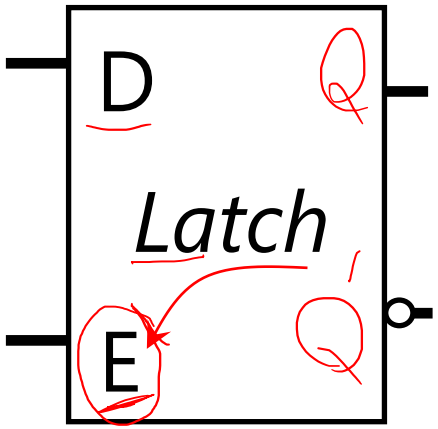
D	E	Q
0	0	Q_t
1	0	Q_t
0	1	0
1	1	1



D	Q
0	0
1	1


R
R
D
S
S
E

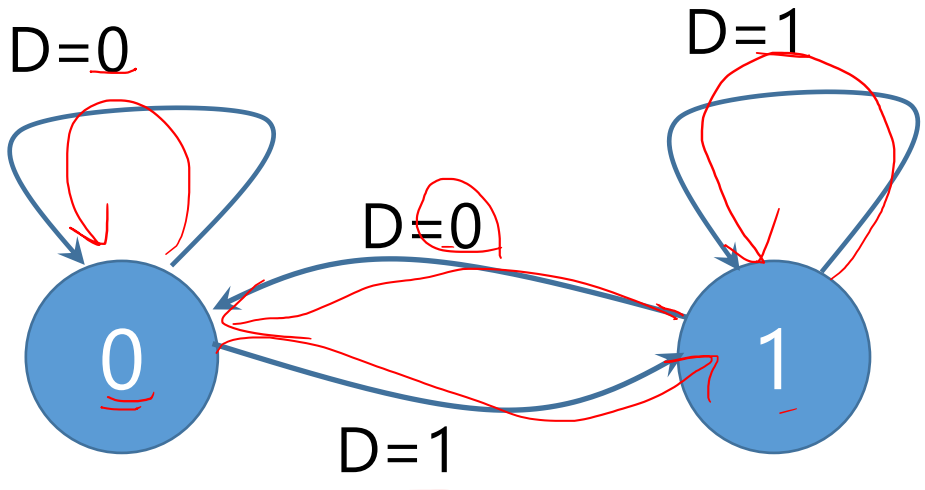
Block Diagram



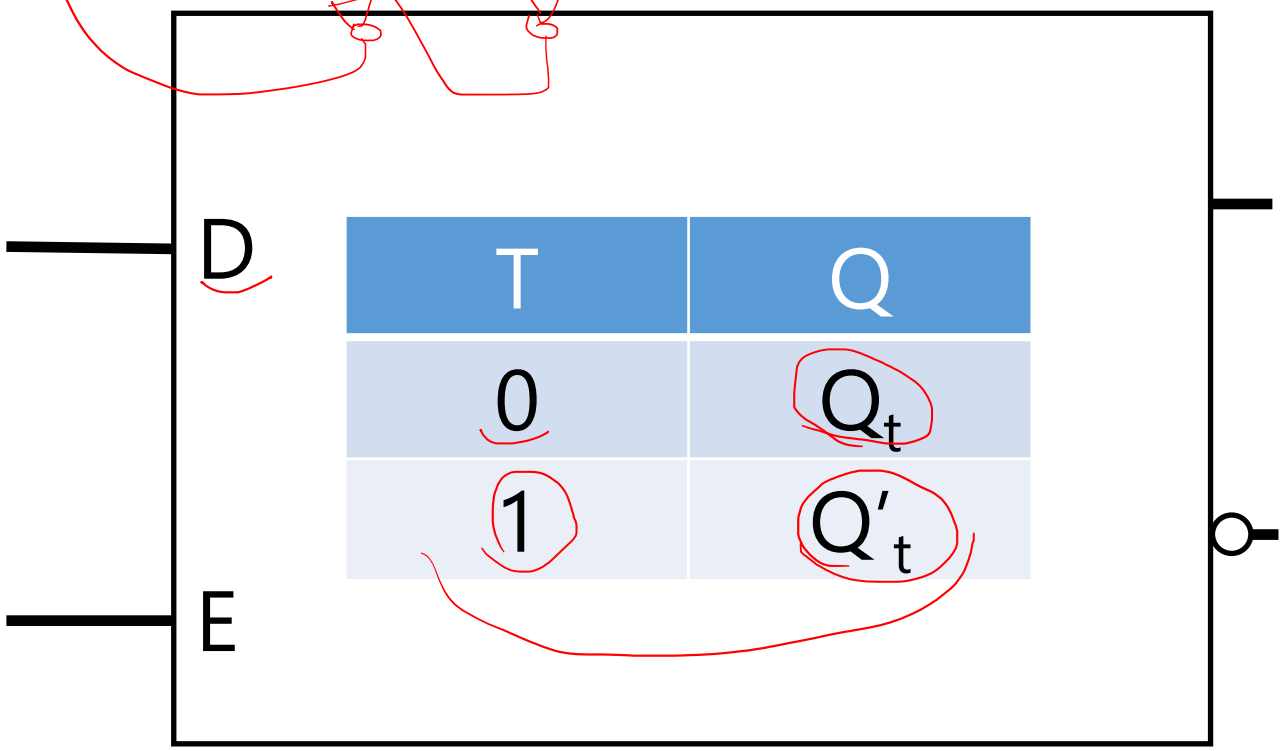
Characteristic Table

D	Q
0	0
1	1

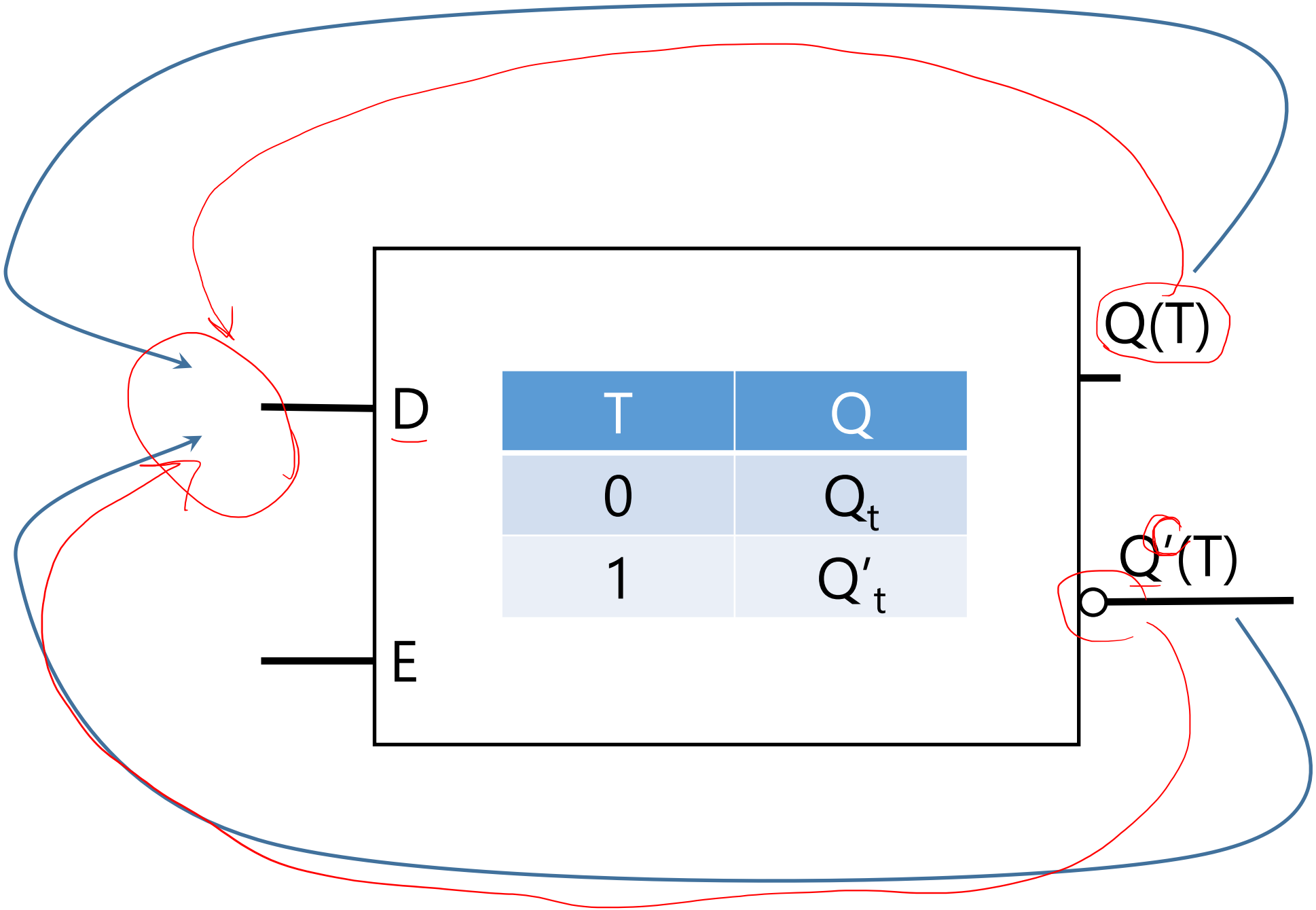
State Transition Diagram

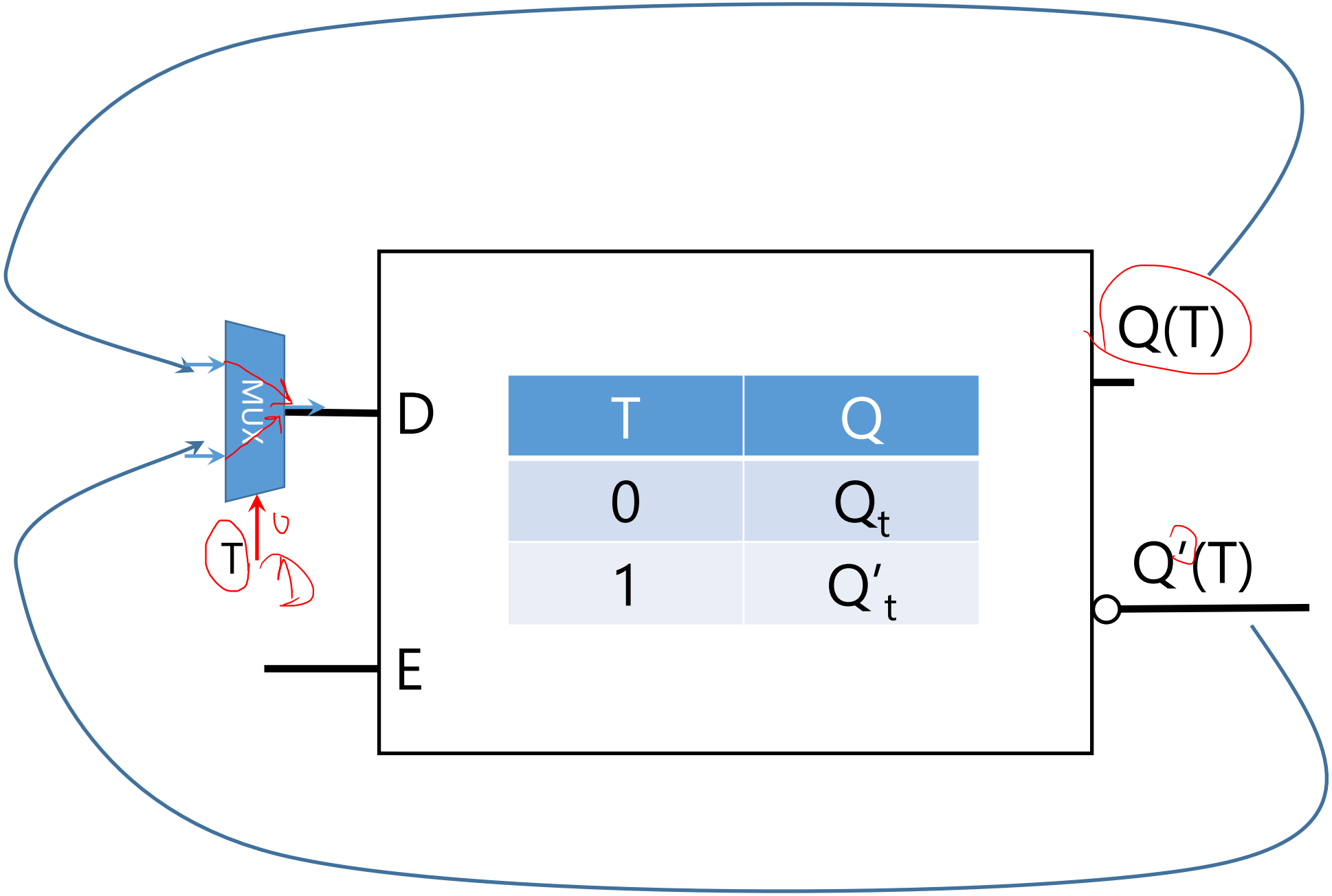


T *Latch*

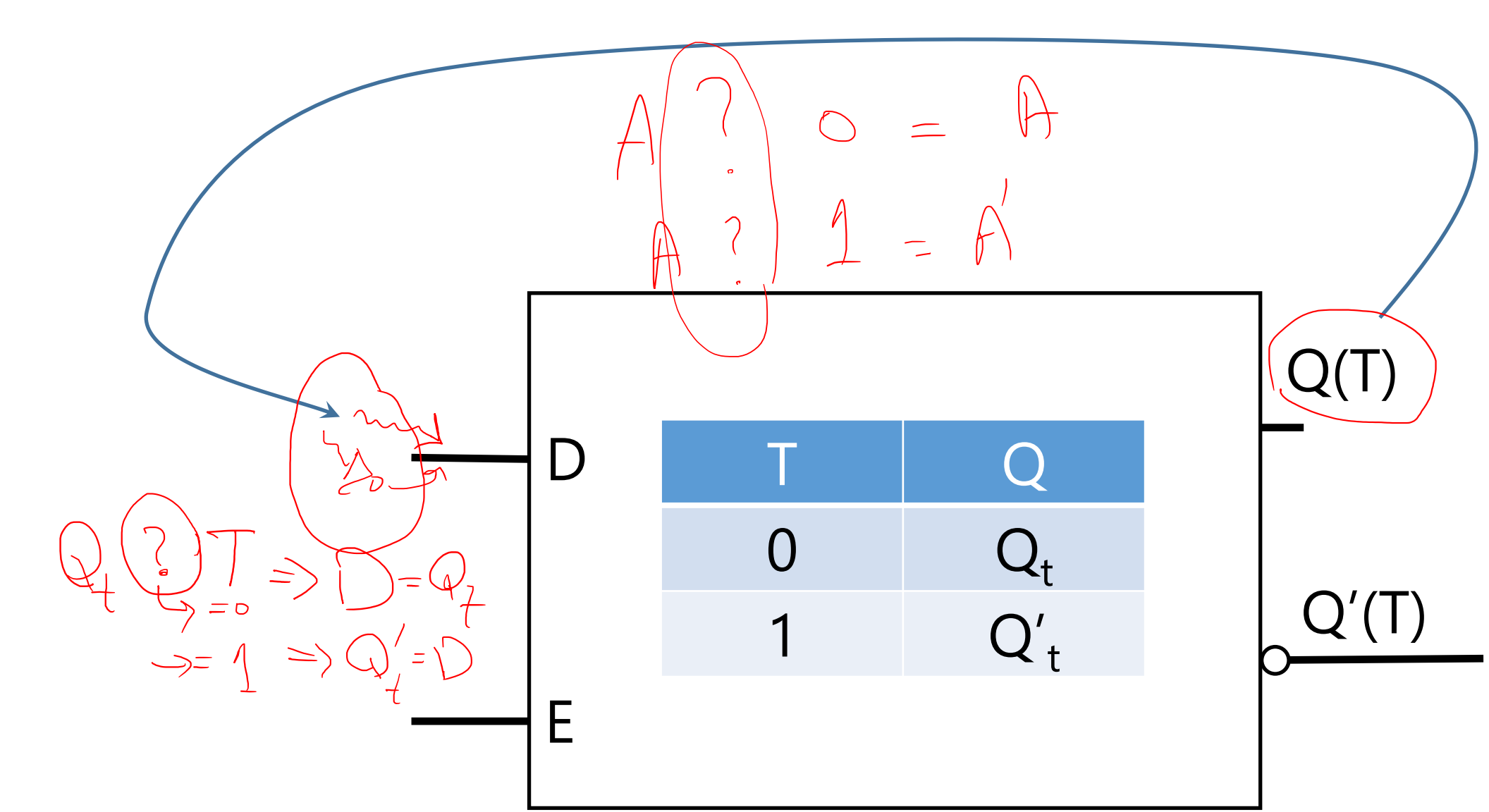


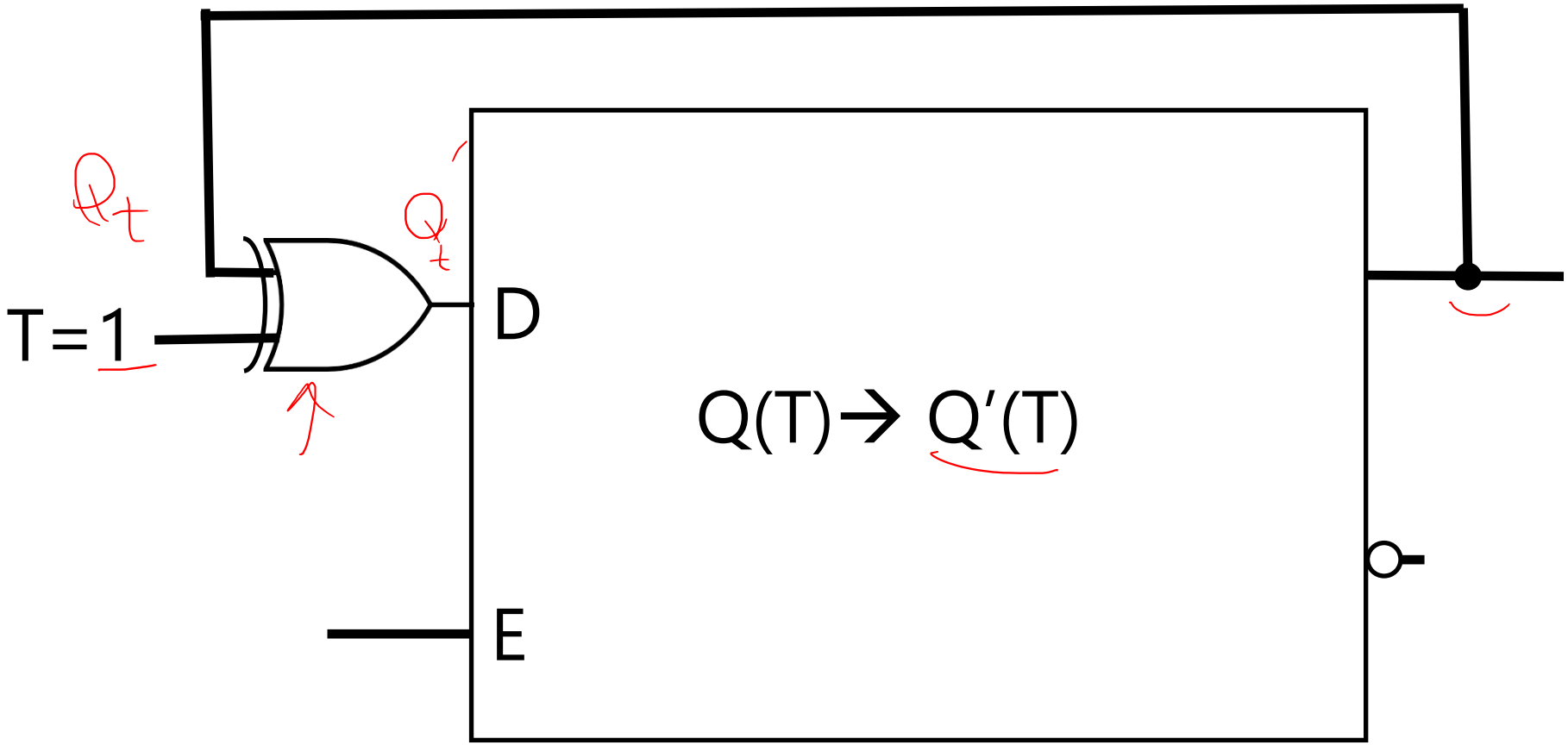
T	Q
0	Q_t
1	Q'_t

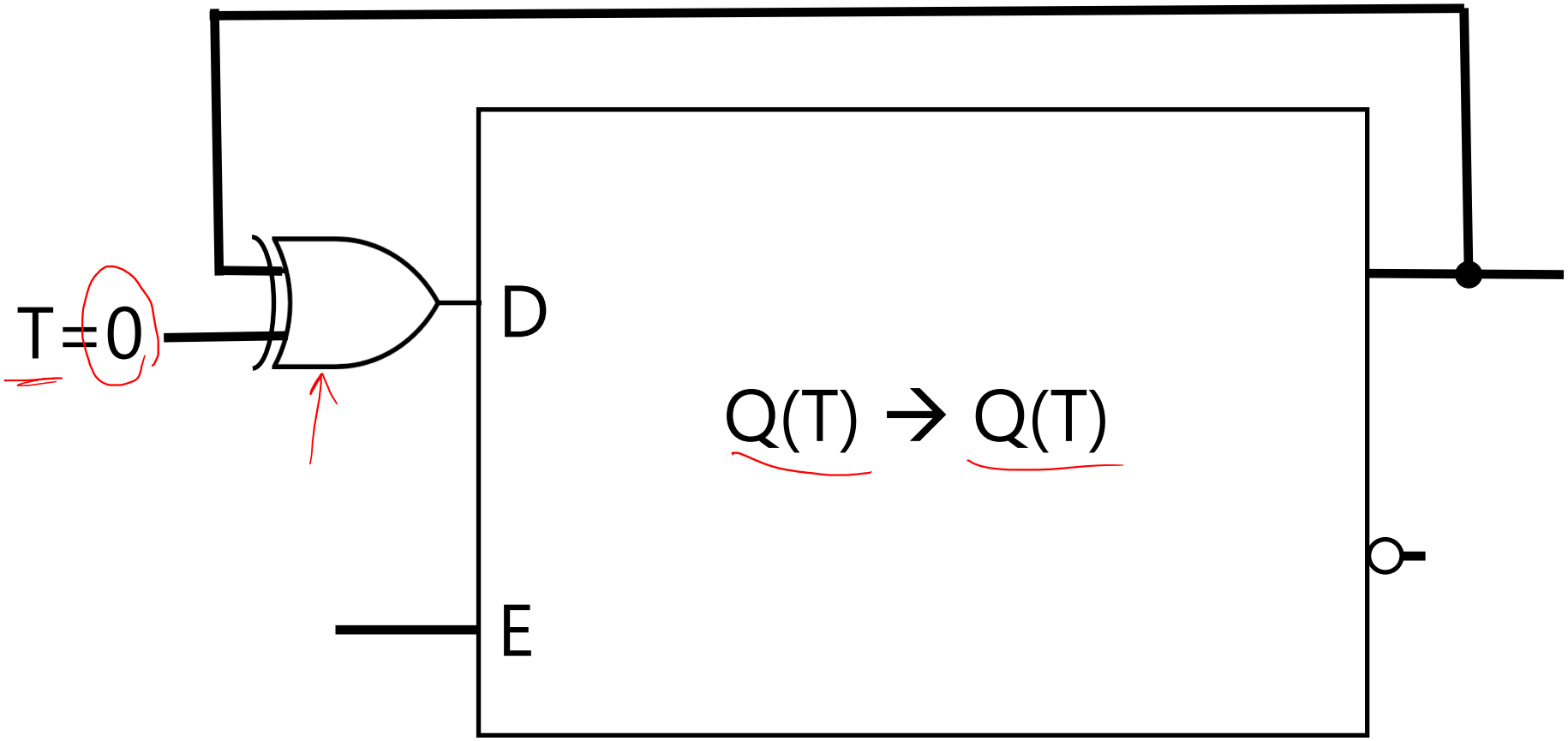

D
E
Q(T)
Q'(T)
T
Q
0
Q_t
1
Q'_t

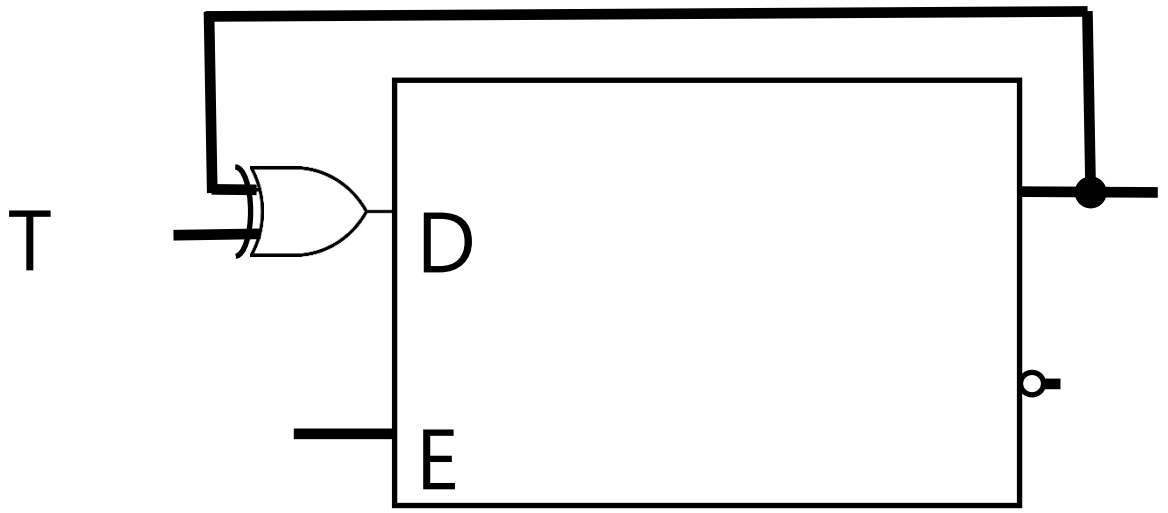


T	Q
0	Q_t
1	Q'_t


A ? 0 = A
A ? 1 = A'
Q(T)
D
E
T
Q
0
Q_t
1
Q'_t
Q_t ? T => D = Q_t
=0
=> 1 => Q'_t = D
Q'(T)

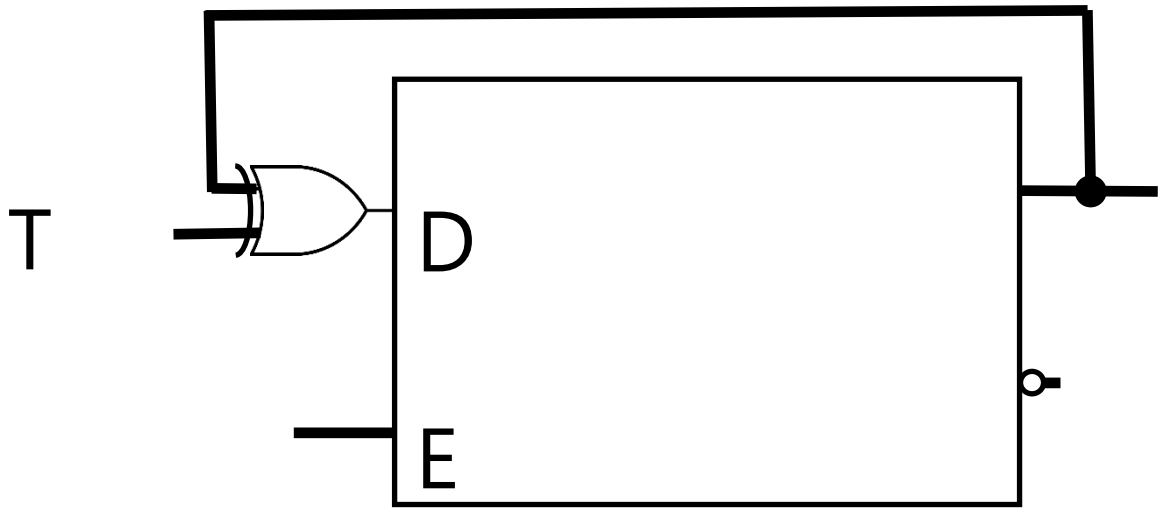

T=1
D
E
Q(T)→ Q'(T)


T=0
D
Q(T) → Q(T)
E



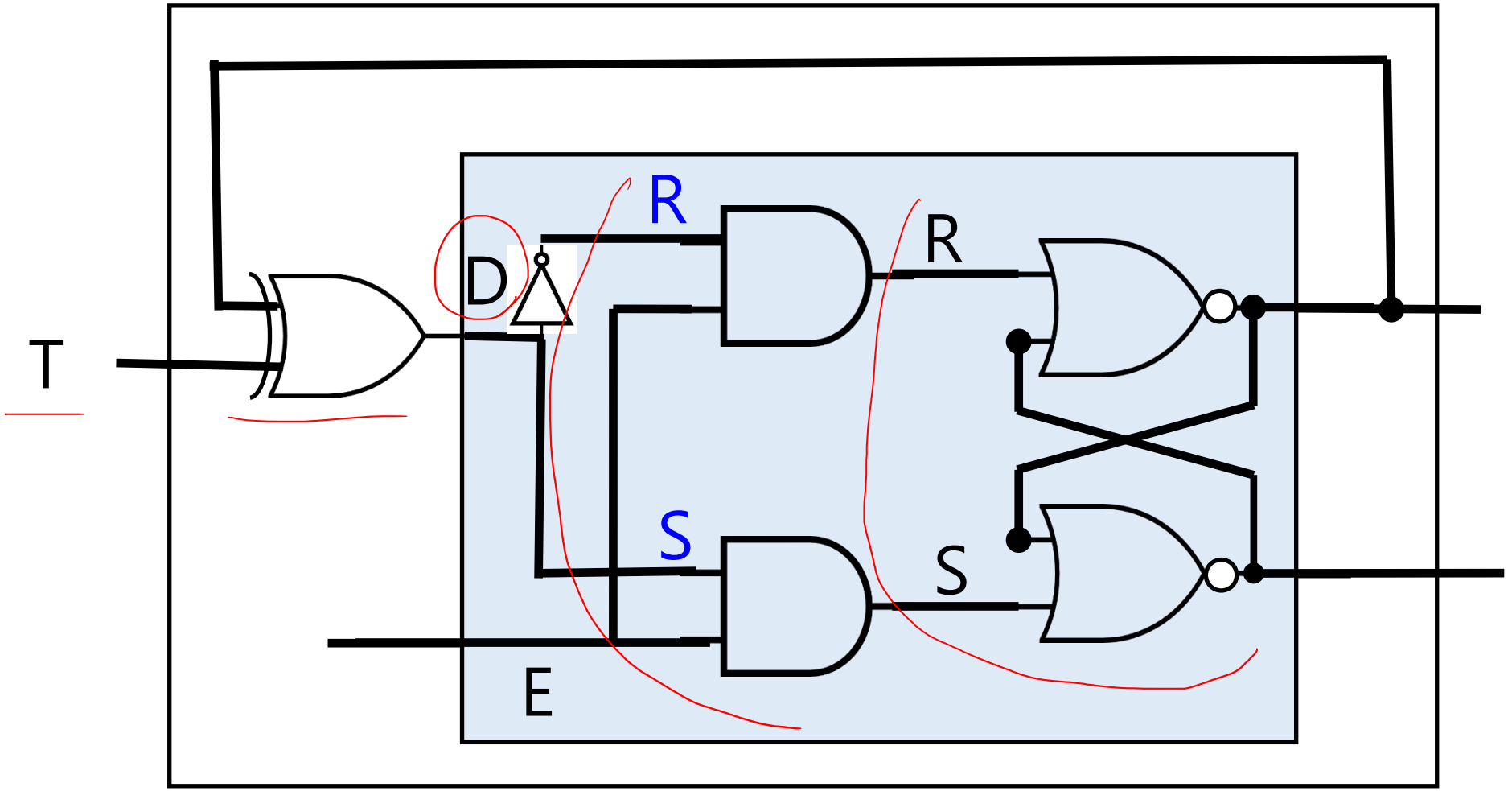
Store

T	D	Q
0	Q⊕0	Q
1	Q⊕1	Q’

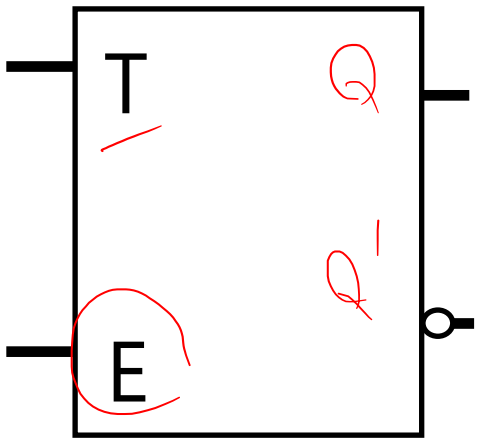


Complement

T	D	Q
0	Q⊕0	Q
1	Q⊕1	Q′


T
D
R
R
S
S
E

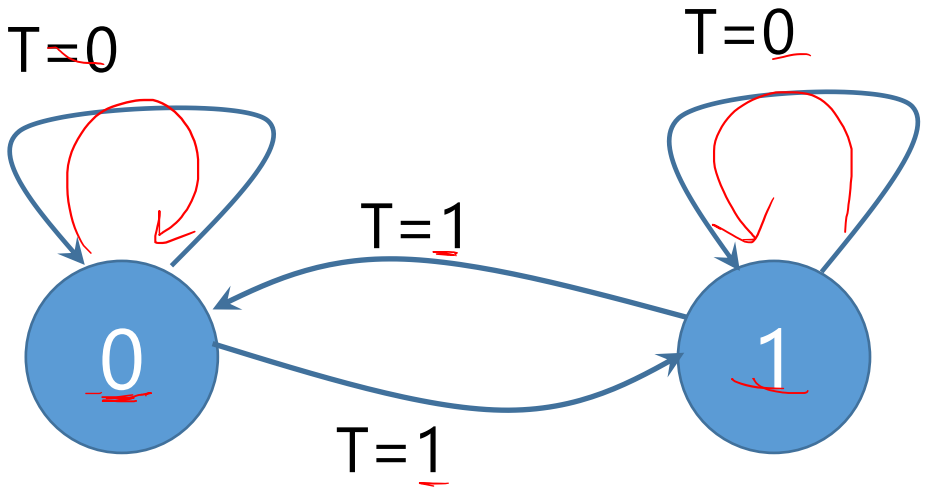
Block Diagram



Characteristic Table

T	Q
0	Q_t
1	Q'_t

State Transition Diagram



JK *Latch*

Jack St. Clair **Kilby**

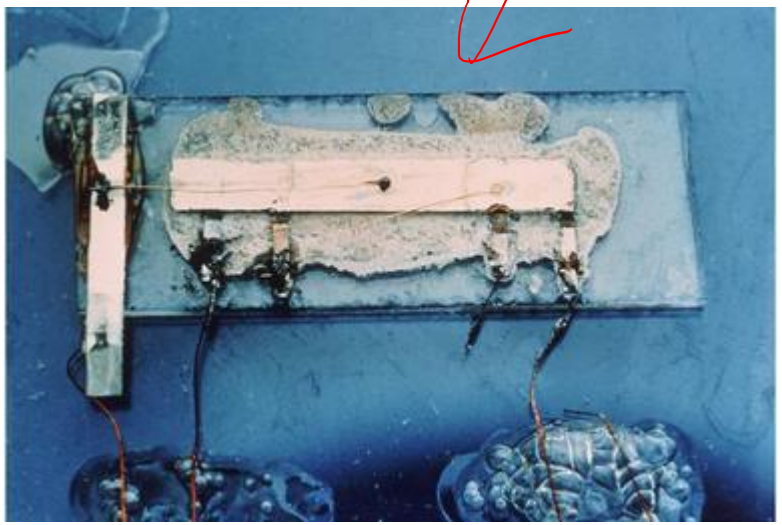
(Nov. 8, 1923 – June 20, 2005)

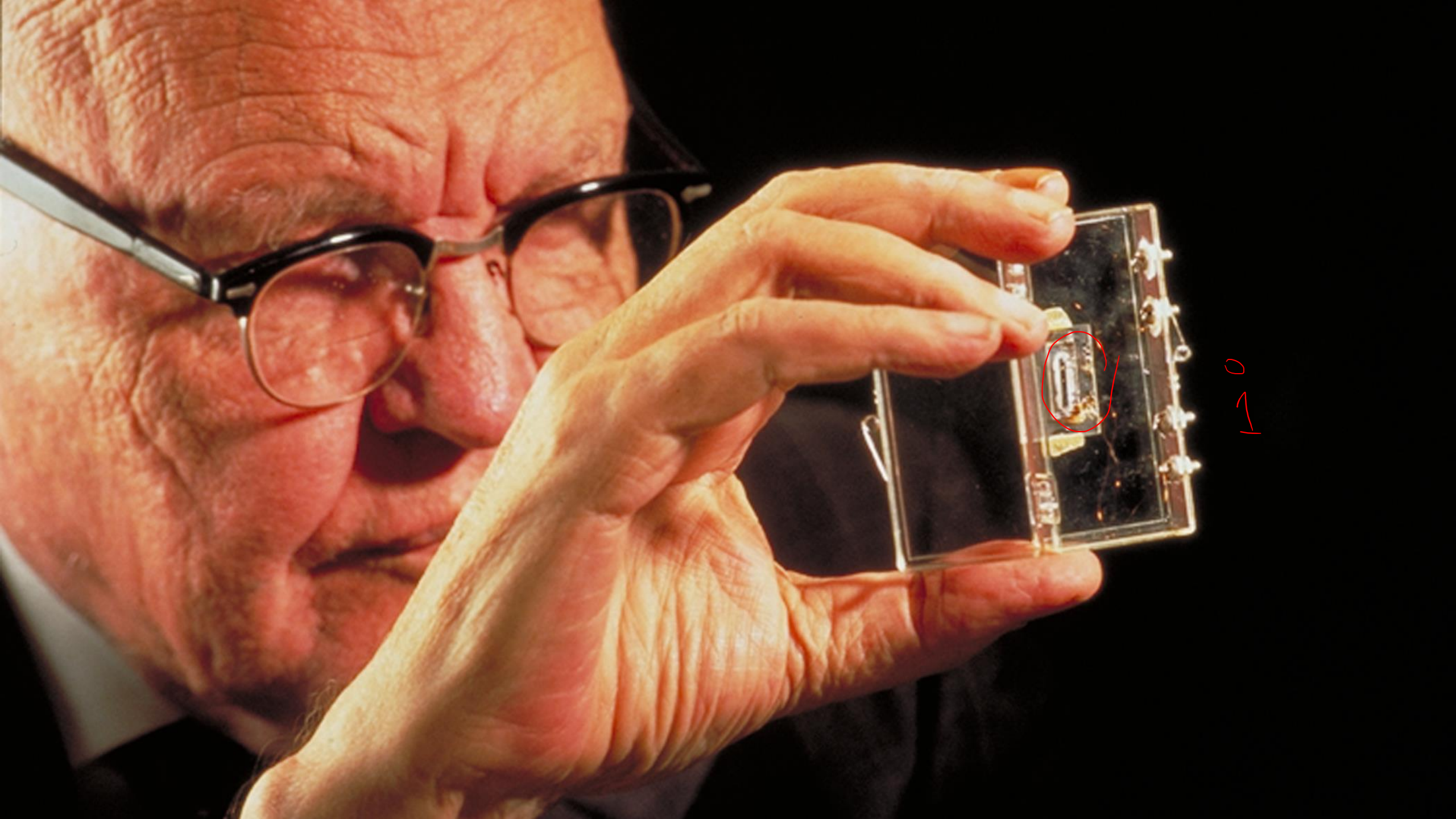
Electrical Engineer

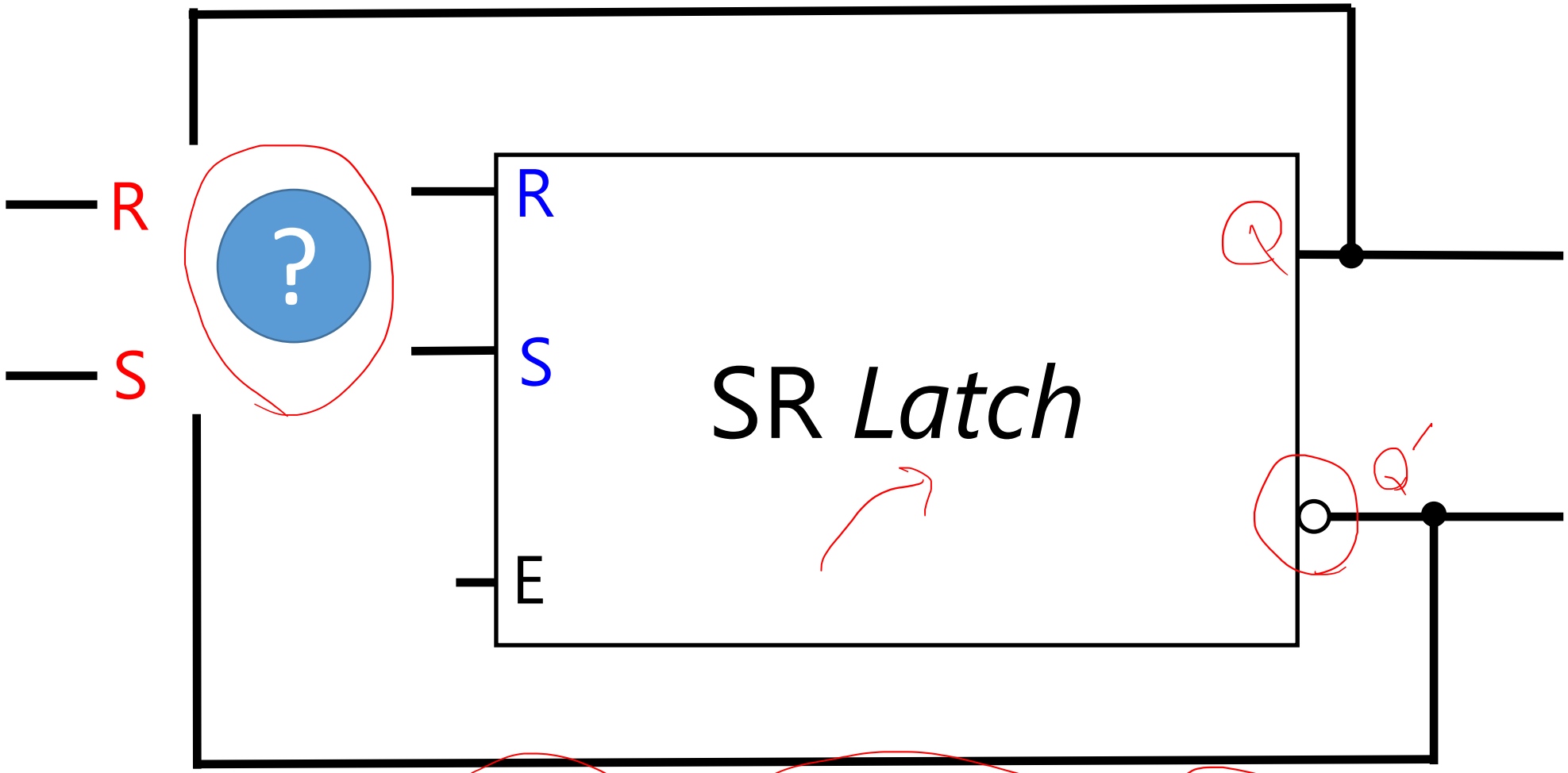
The 1st integrated circuit

1958

Nobel Prize in Physics, 2000

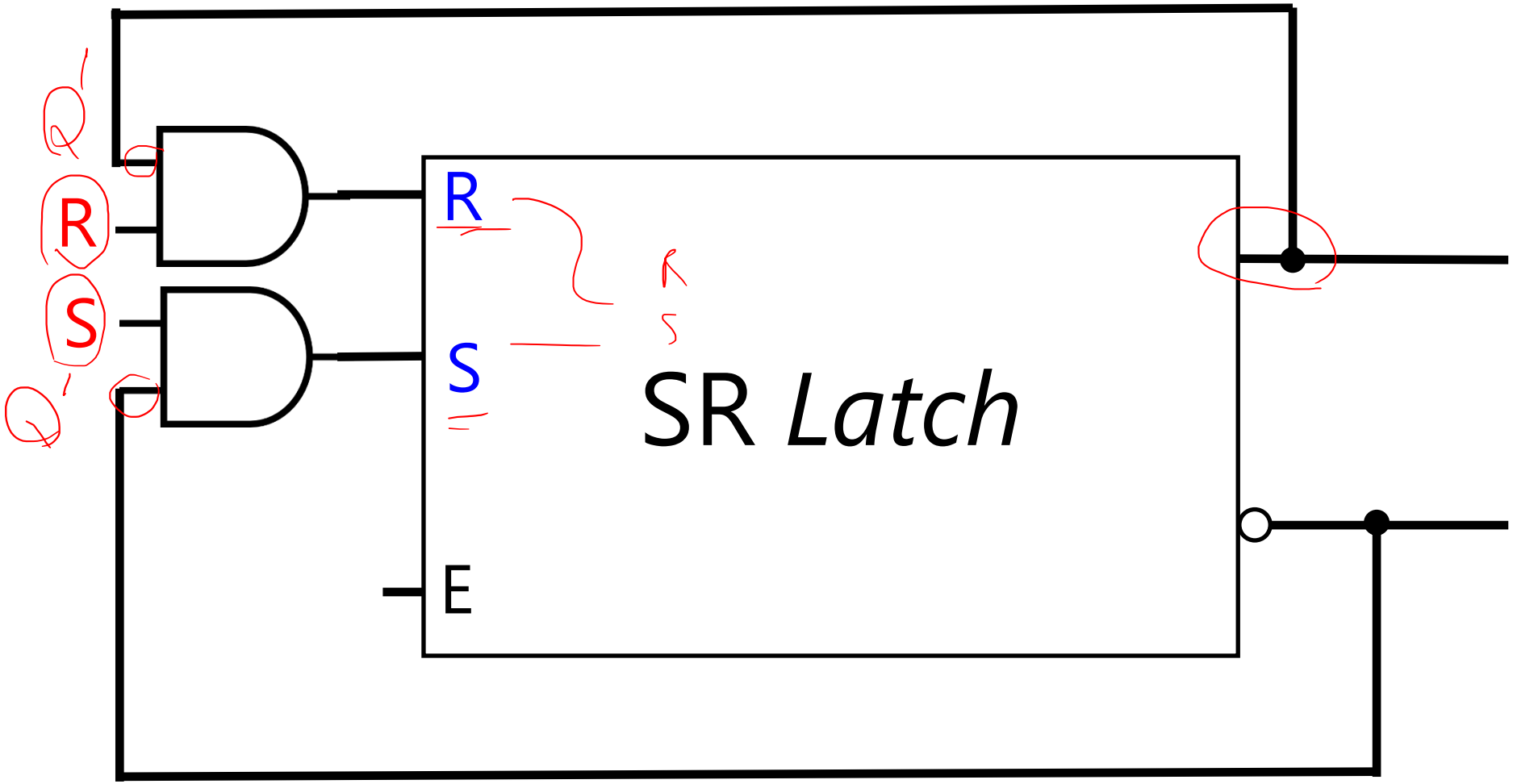


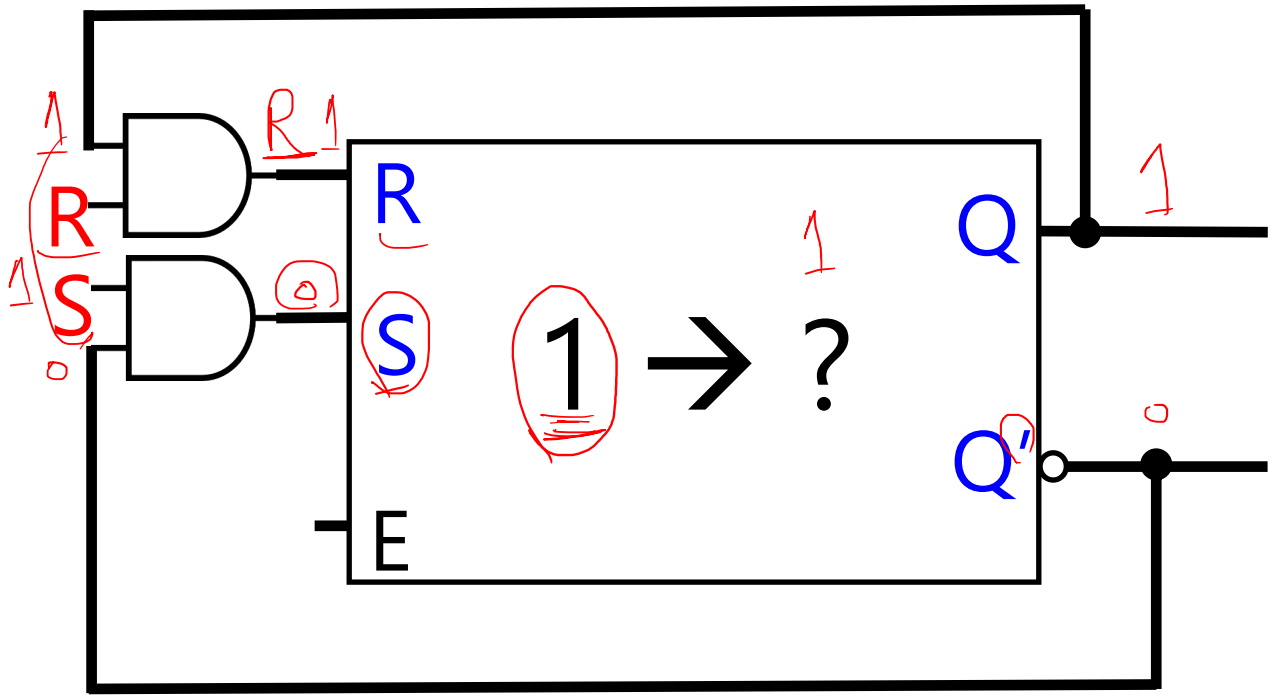




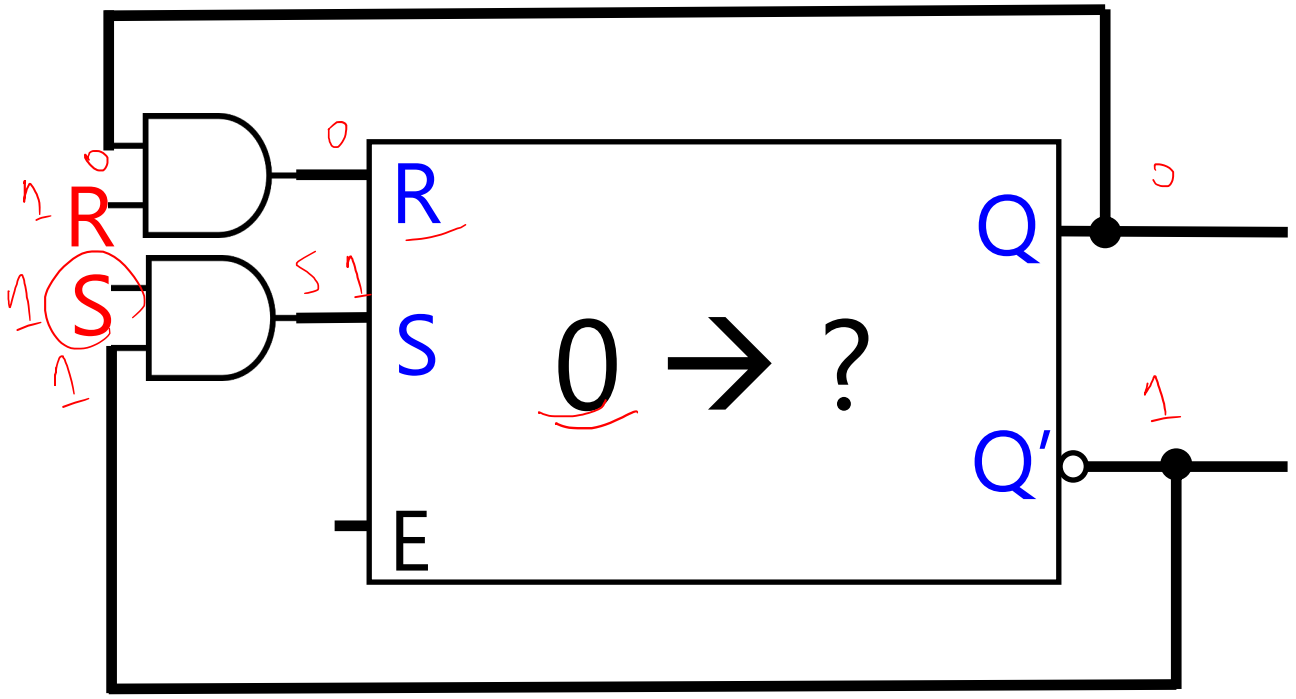
Let's mix T and SR: Store, Set, Reset, Complement

Although you have to guess, we'll see a design algorithm for it ☺

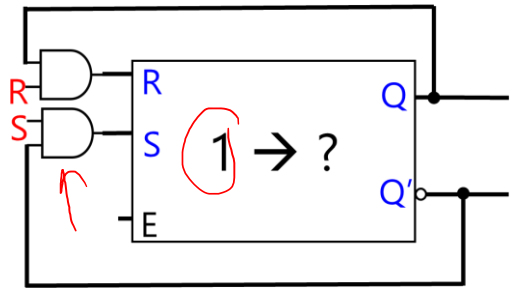

R
S
Q
R
S
E
SR Latch



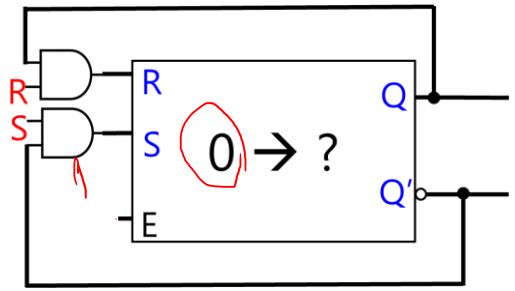
S	R	S	R	Q
0	0	0	0	Store=Q_t=1
0	1	0	1	Reset=0
1	0	0	0	Store=Q_t=1
1	1	0	1	Reset=0



S	R	S	R	Q
0	0	0	0	Store=Q_t=0
0	1	0	0	Store=Q_t=0
1	0	1	0	Set=1
1	1	1	0	Set=1



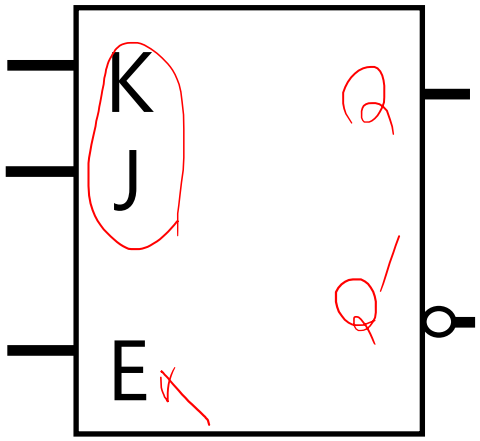
S	R	S	R	Q
0	0	0	0	Store=Q_t=1
0	1	0	1	Reset=0
1	0	0	0	Store=Q_t=1
1	1	0	1	Reset=0



S	R	S	R	Q
0	0	0	0	Store=Q_t=0
0	1	0	0	Store=Q_t=0
1	0	1	0	Set=1
1	1	1	0	Set=1

S=J	R=K	Q
0	0	Store=Q_t
0	1	Reset = 0
1	0	Set = 1
1	1	Comp. =Q'_t

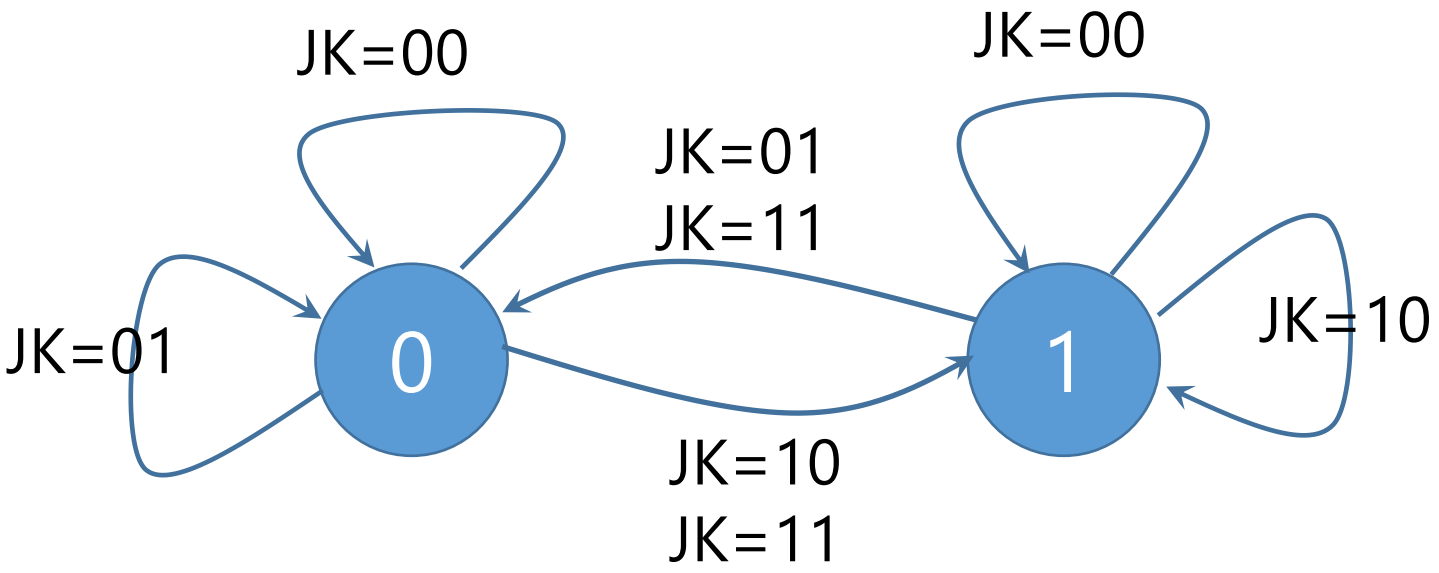
Block Diagram



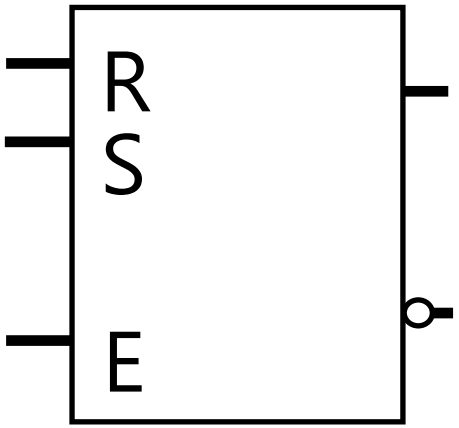
Characteristic Table

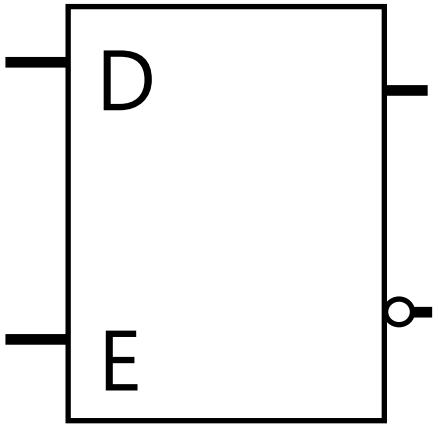
J	K	Q
0	0	Q_t
0	1	0
1	0	1
1	1	Q'_t

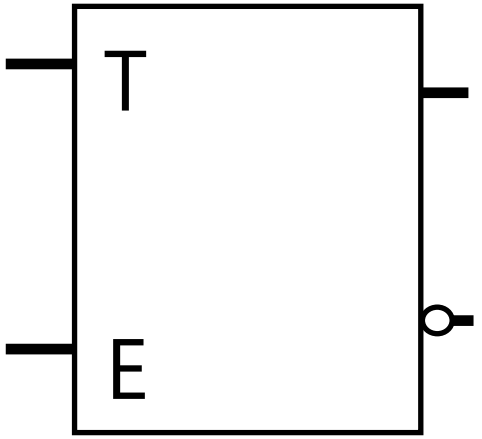
State Transition Diagram

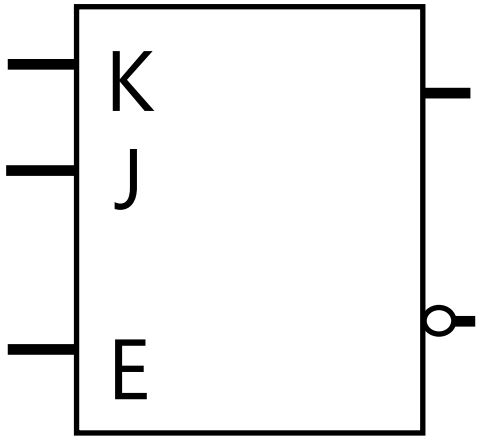


Recap









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

Clock

shortened as *clk*

timing device that generates a train of pulses