

Sequential Logic

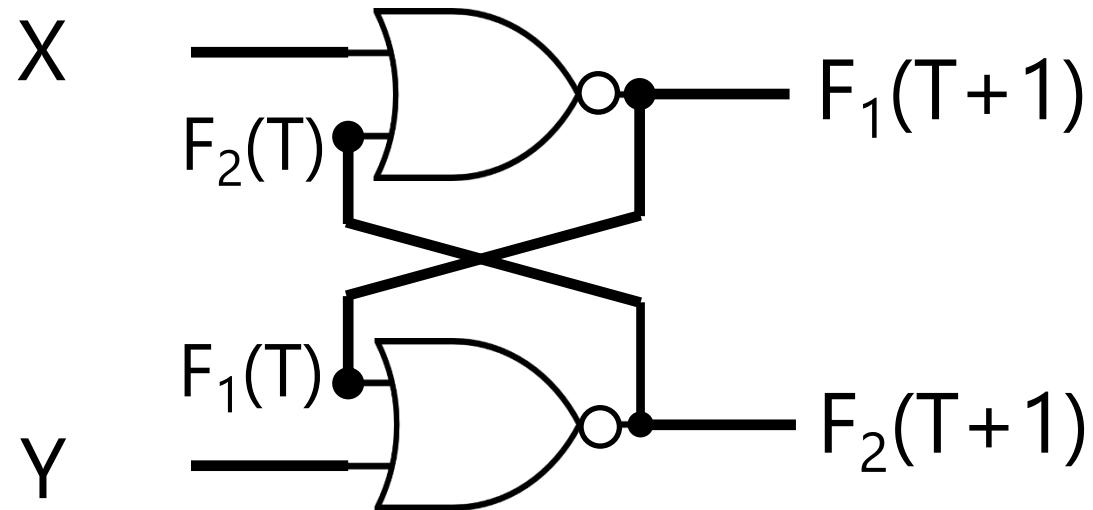


Student Evaluations of Teaching (SET)

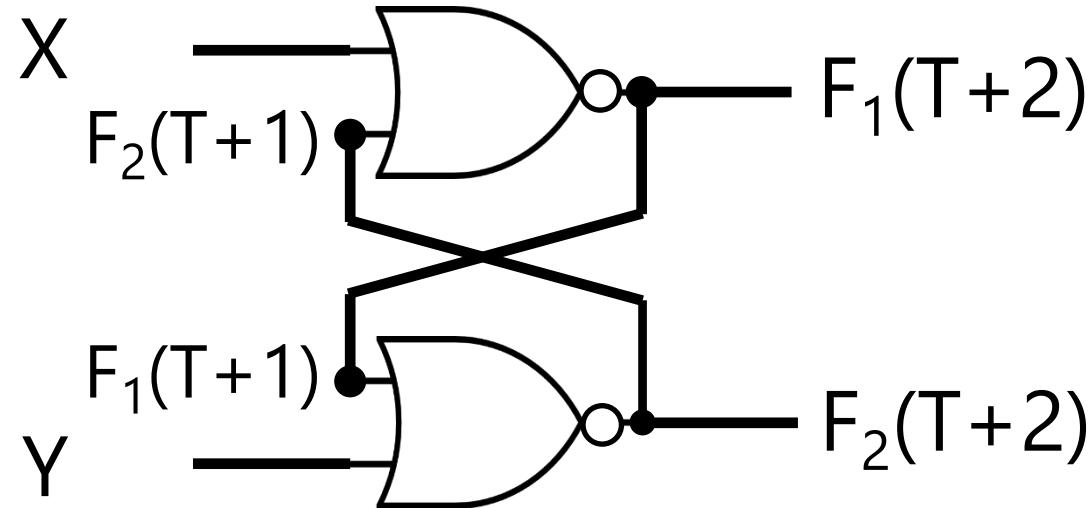
UWinsite Student

Thursday, November 26 – Wednesday, December 9, 2020

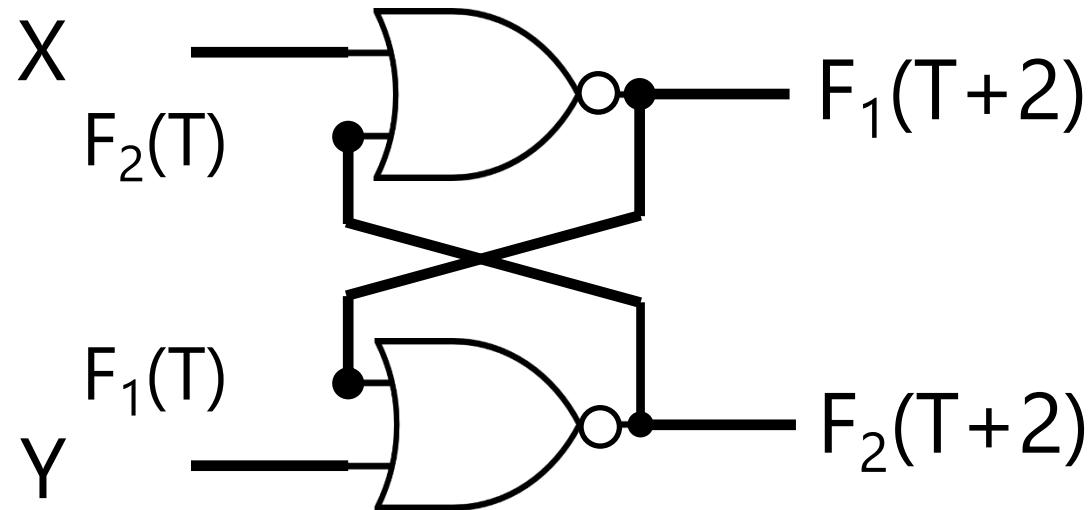
http://ask.uwindsor.ca/app/answers/detail/a_id/176



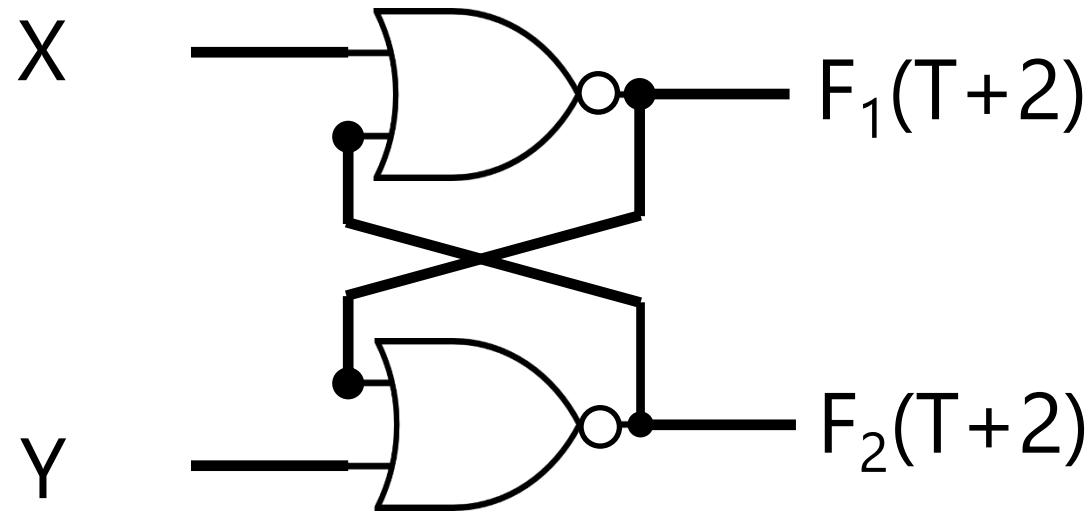
Y	X	$F_1(T+1)$	$F_2(T+1)$
0	0	$F'_2(T)$	$F'_1(T)$
0	1	0	$F'_1(T)$
1	0	$F'_2(T)$	0
1	1	0	0



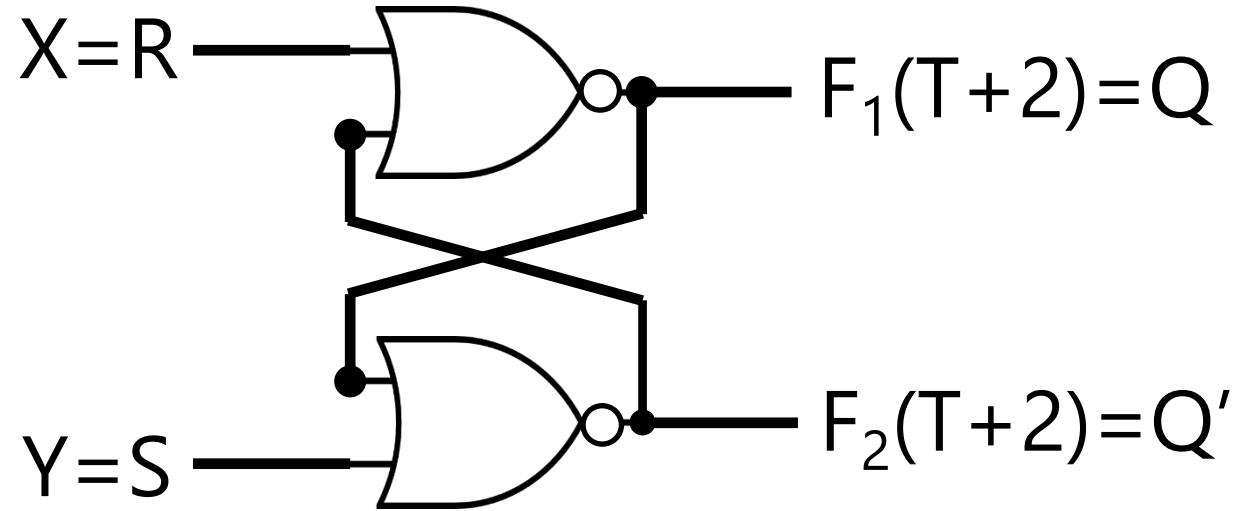
Y	X	$F_1(T+2)$	$F_2(T+2)$
0	0	$F'_2(T+1)$	$F'_1(T+1)$
0	1	0	$F'_1(T+1)$
1	0	$F'_2(T+1)$	0
1	1	0	0



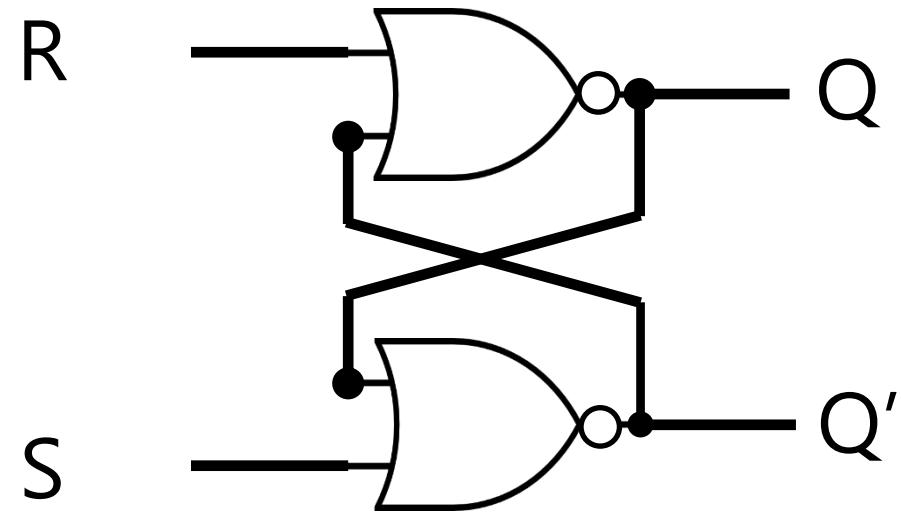
Y	X	$F_1(T+1)$	$F_2(T+1)$	$F_1(T+2)$	$F_2(T+2)$
0	0	$F'_2(T)$	$F'_1(T)$	$F'_2(T+1) = F_1(T)$	$F'_1(T+1)=F_2(T)$
0	1	0	$F'_1(T)$	0	$F'_1(T+1)=1$
1	0	$F'_2(T)$	0	$F'_2(T+1)=1$	0
1	1	0	0	0	0



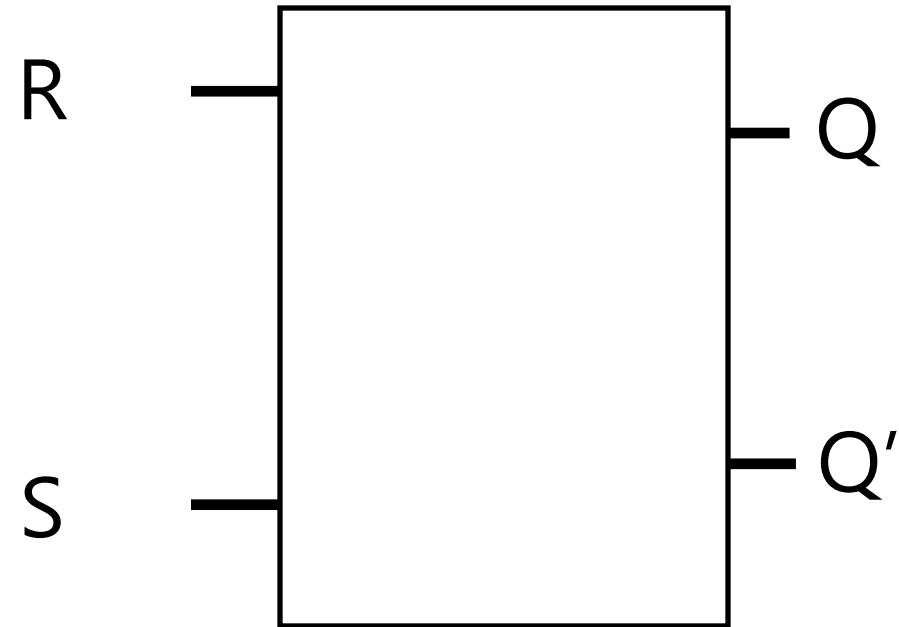
Y	X	$F_1(T+2)$	$F_2(T+2)$
0	0	$F_1(T)$	$F_2(T)$
0	1	0	1
1	0	1	0
1	1	0	0



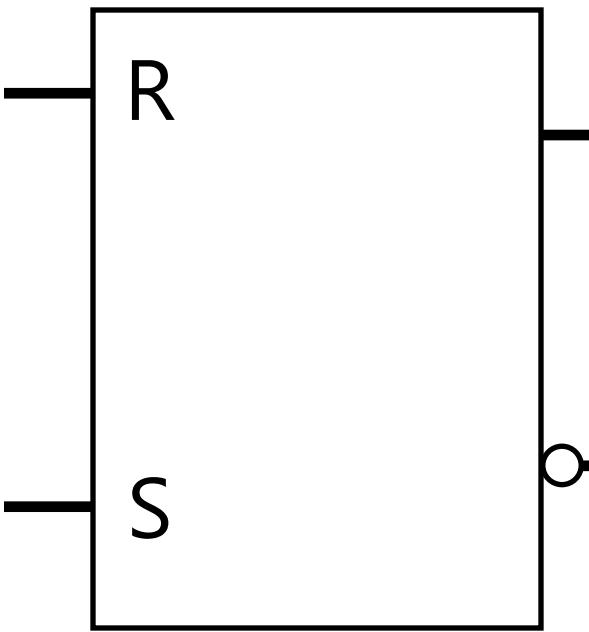
$Y=S$	$X=R$	$F_1(T+2)=Q$	$F_2(T+2)=Q'$
0	0	Q_t	Q'_t
0	1	0	1
1	0	1	0
1	1	\times	\times



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

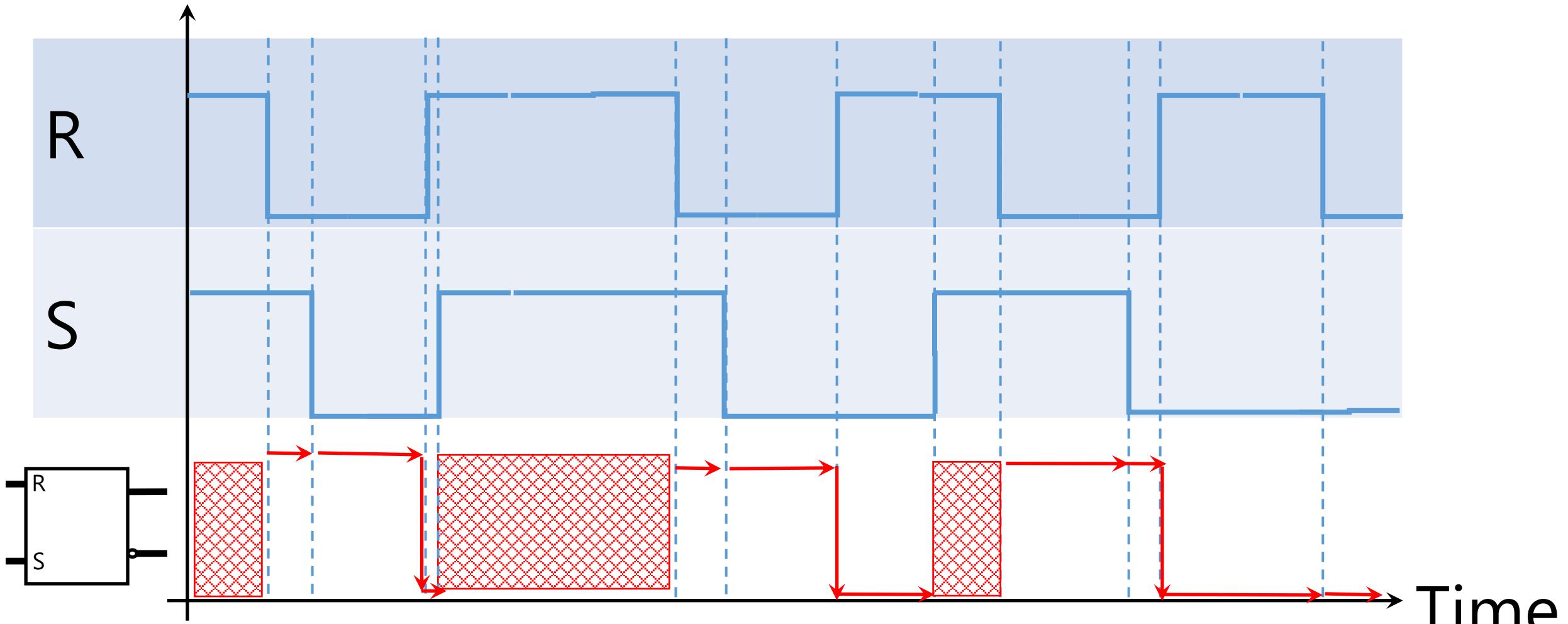


S	R	Q	Q'
0	0	Q_t	Q'_t
0	1	0	1
1	0	1	0
1	1	X	X



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

Voltage

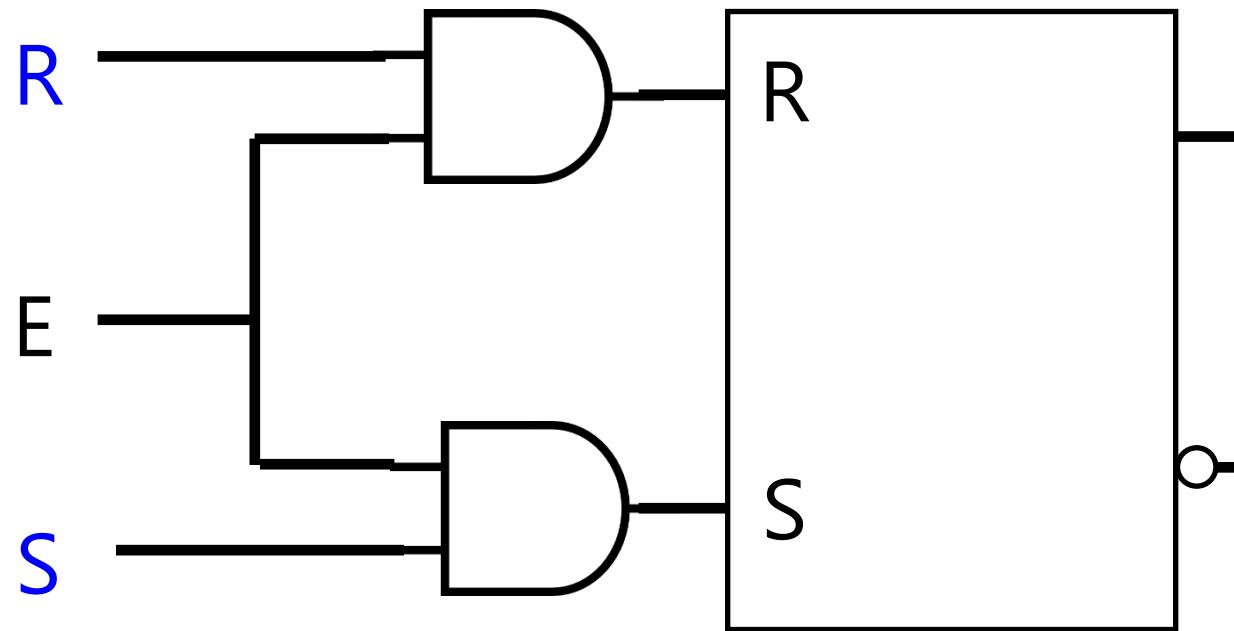


S and R control **how** the
state changes.

SR Latch

Put a control on **when**
change applies

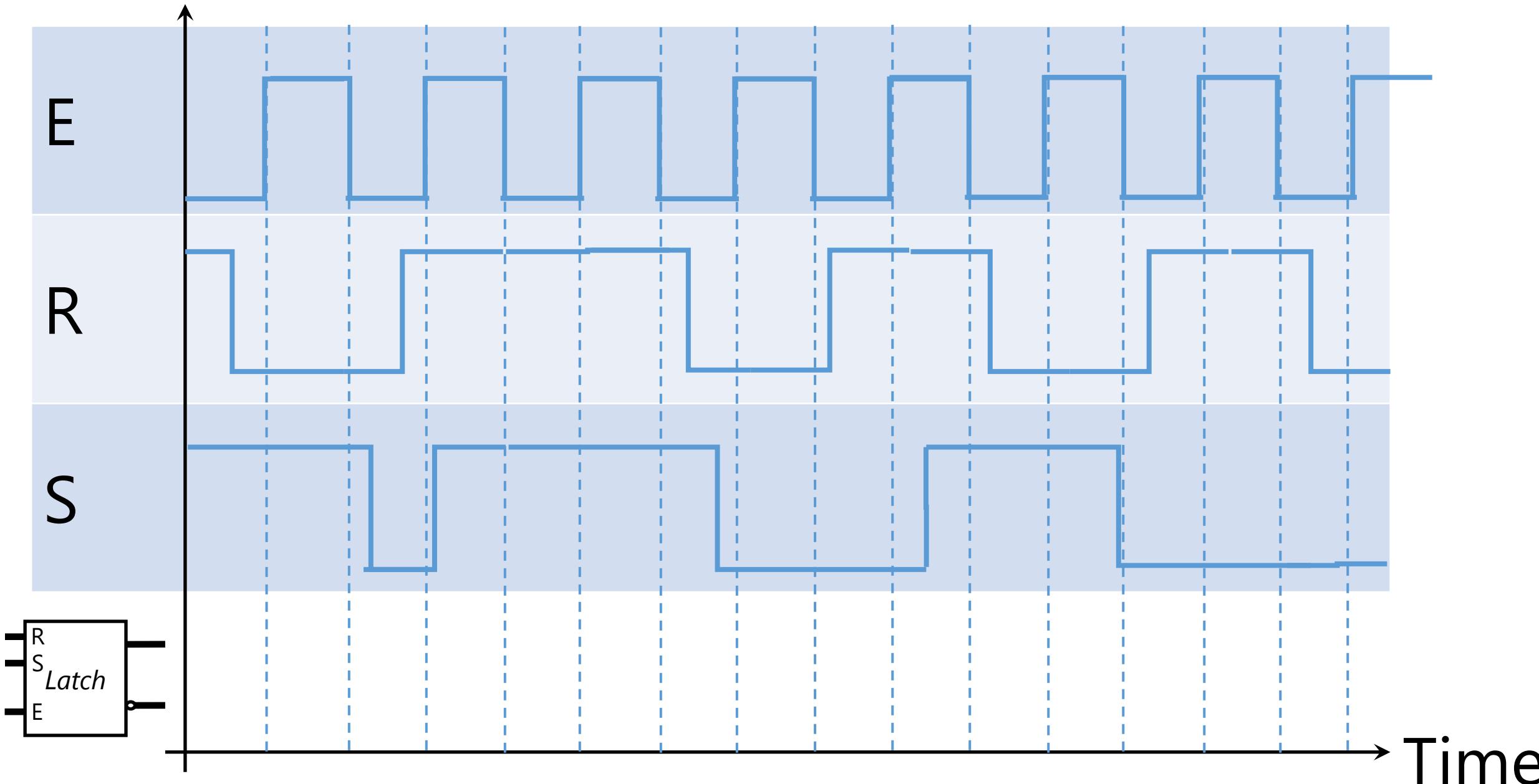
SR w/ enable input



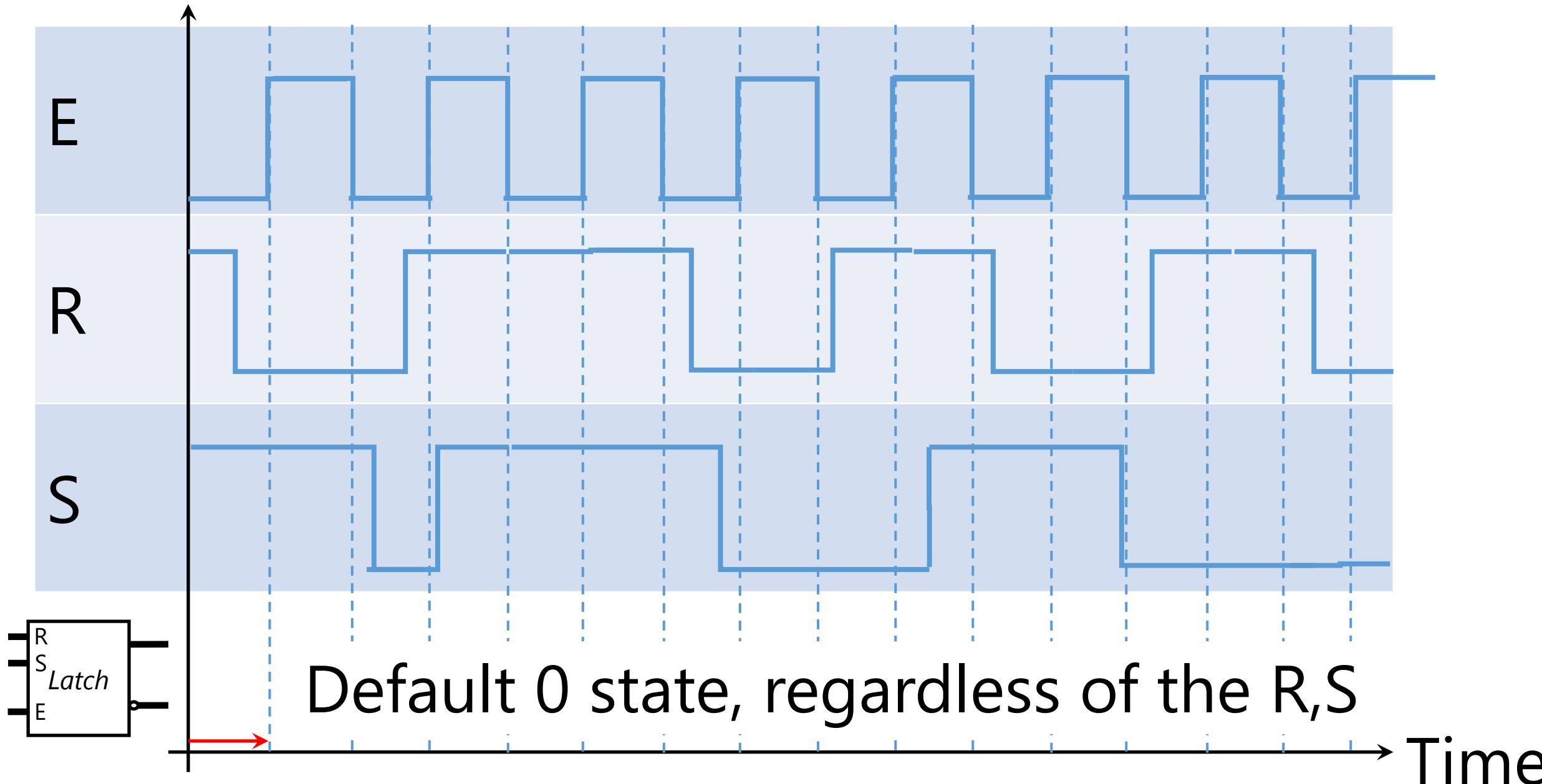
E	S	R	S	R	Q	Q'
0	x	x	0	0	Q_t	Q'_t
1	0	0	0	0	Q_t	Q'_t
1	0	1	0	1	0	1
1	1	0	1	0	1	0
1	1	1	1	1	x	x



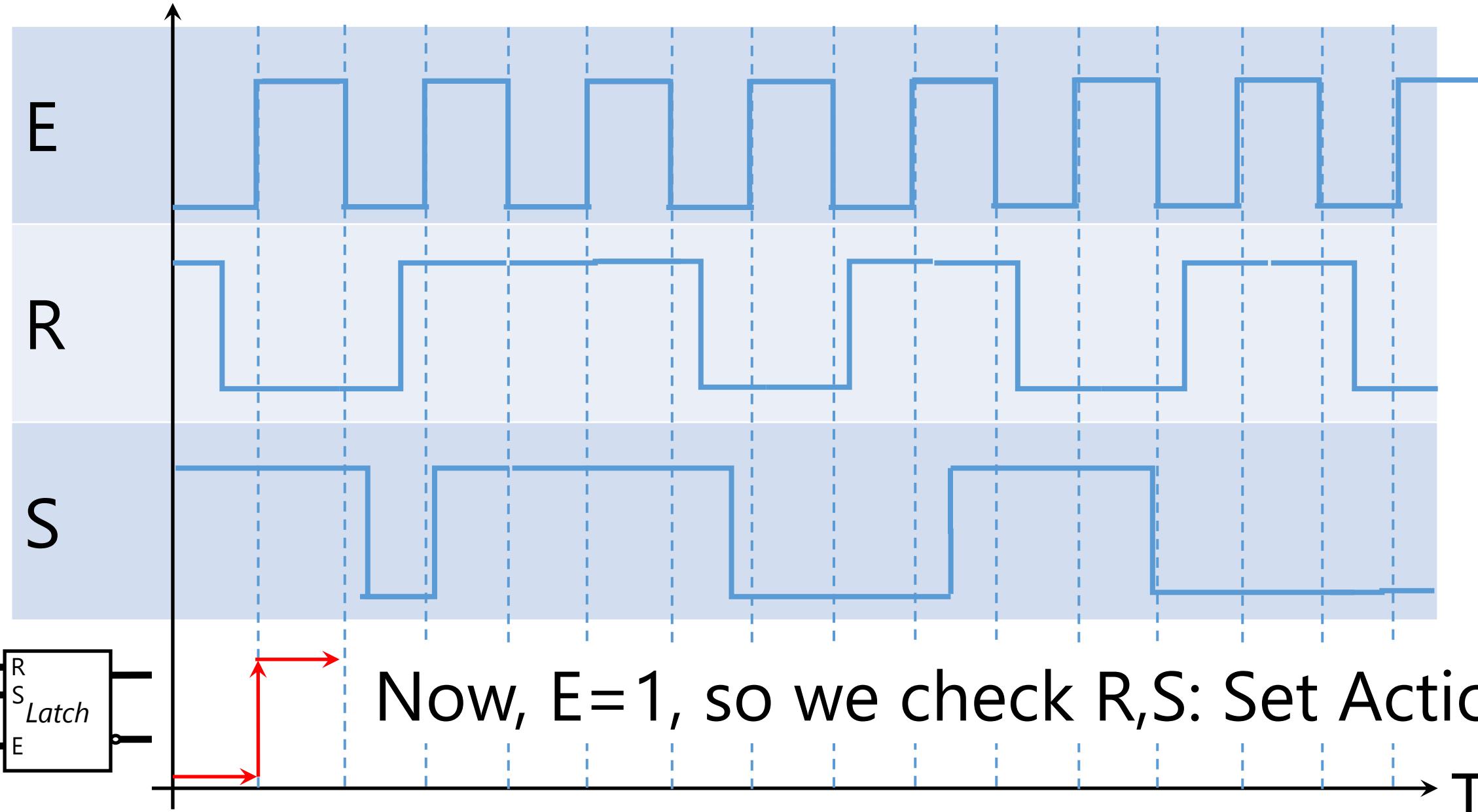
Voltage



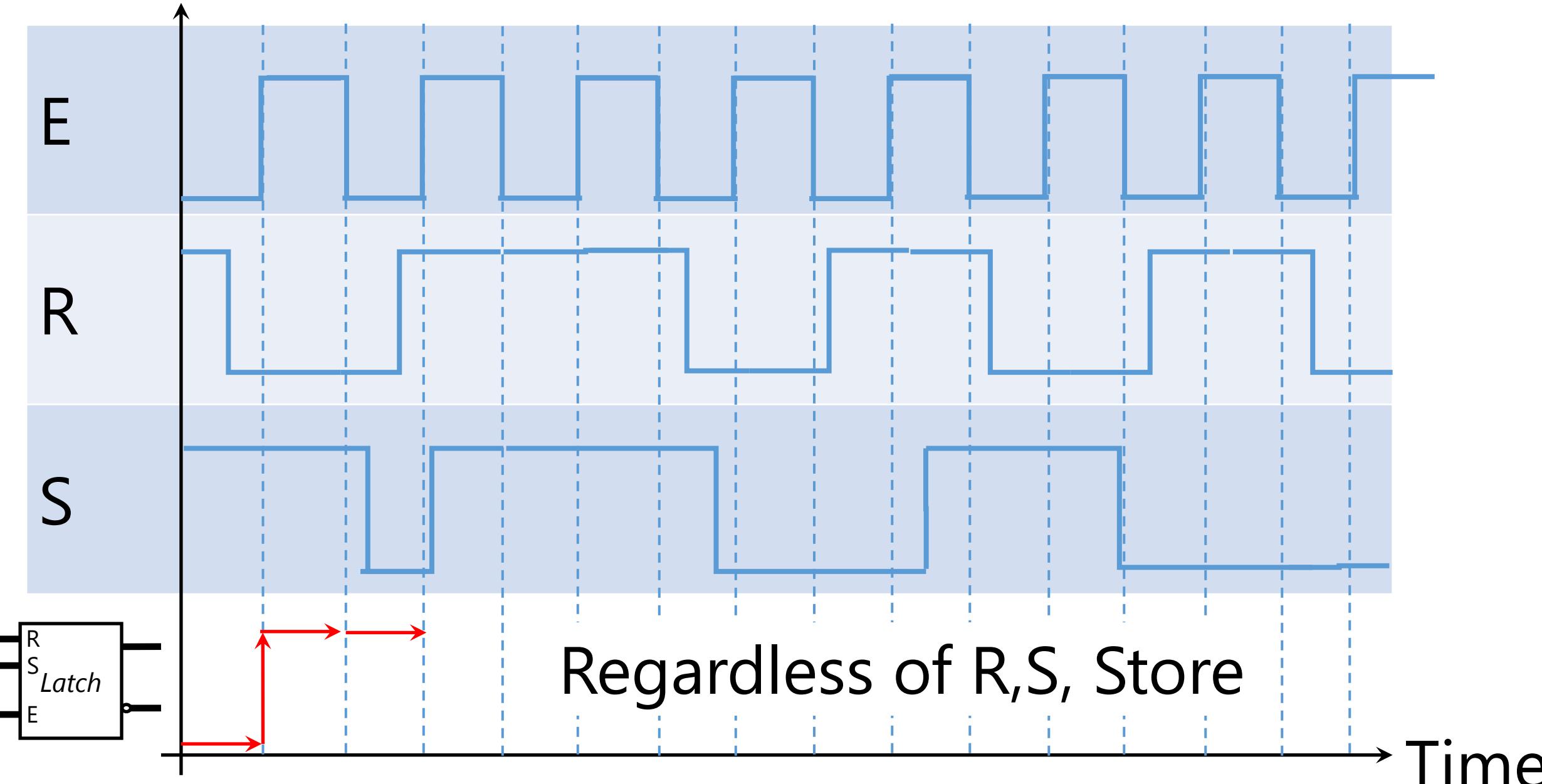
Voltage



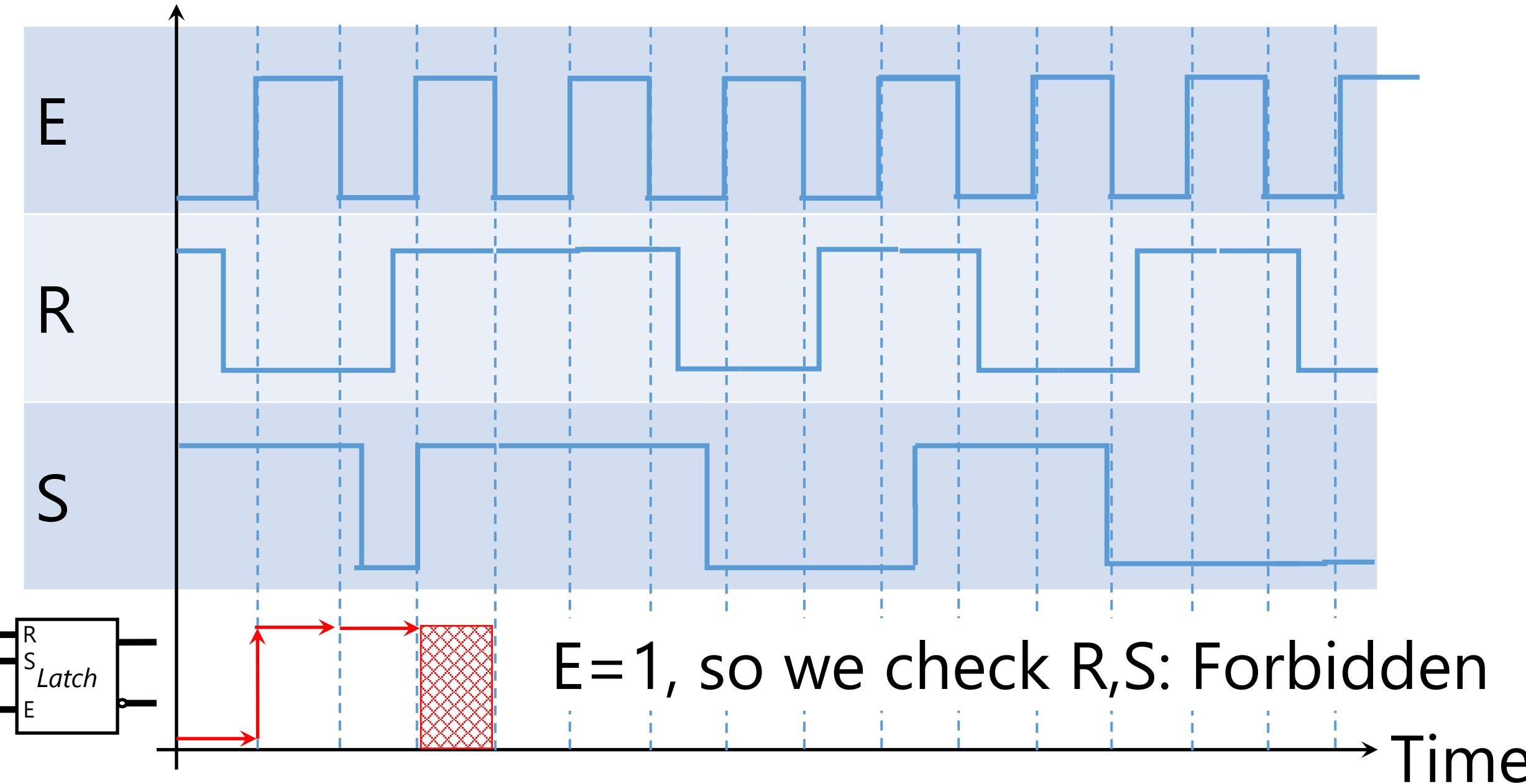
Voltage



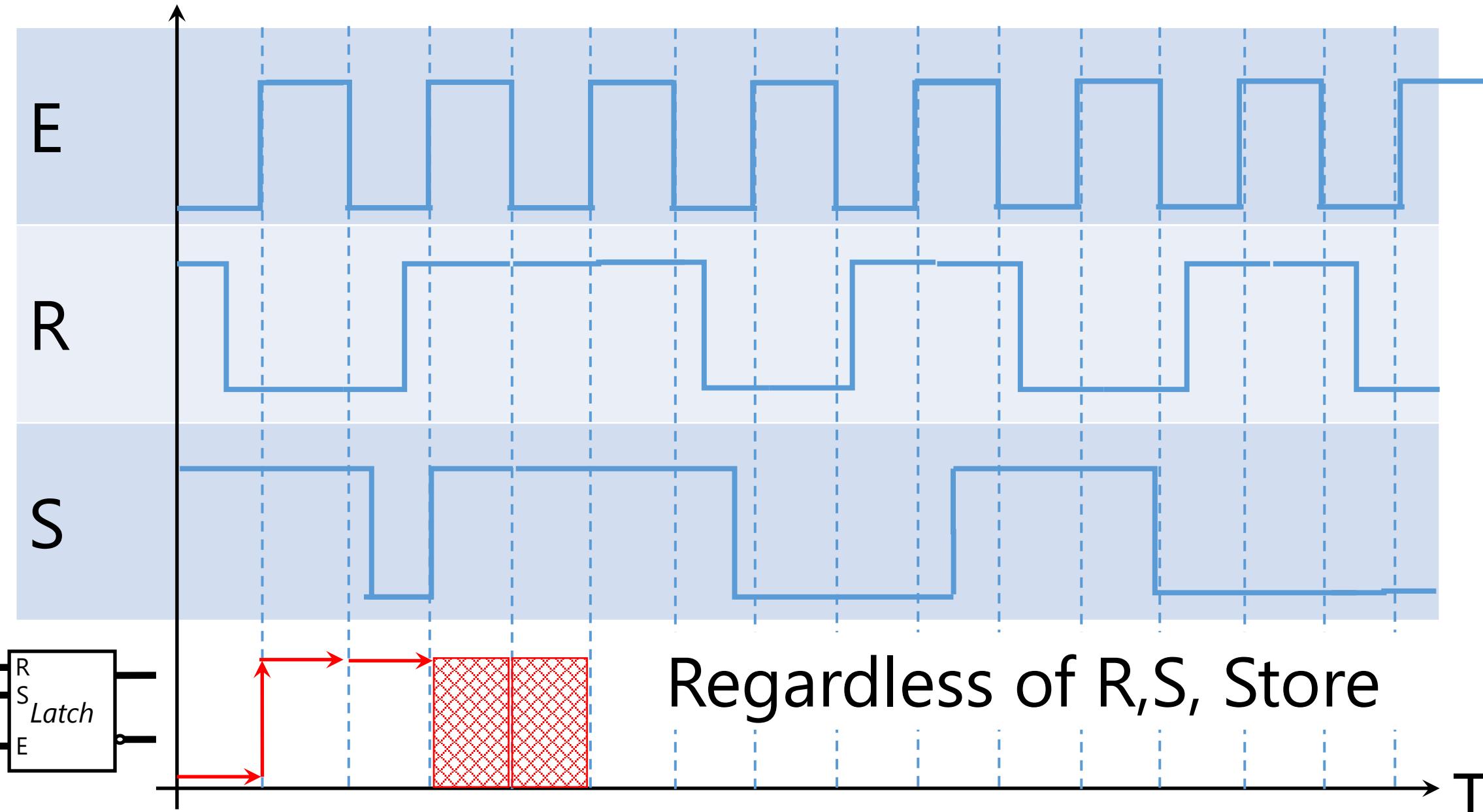
Voltage



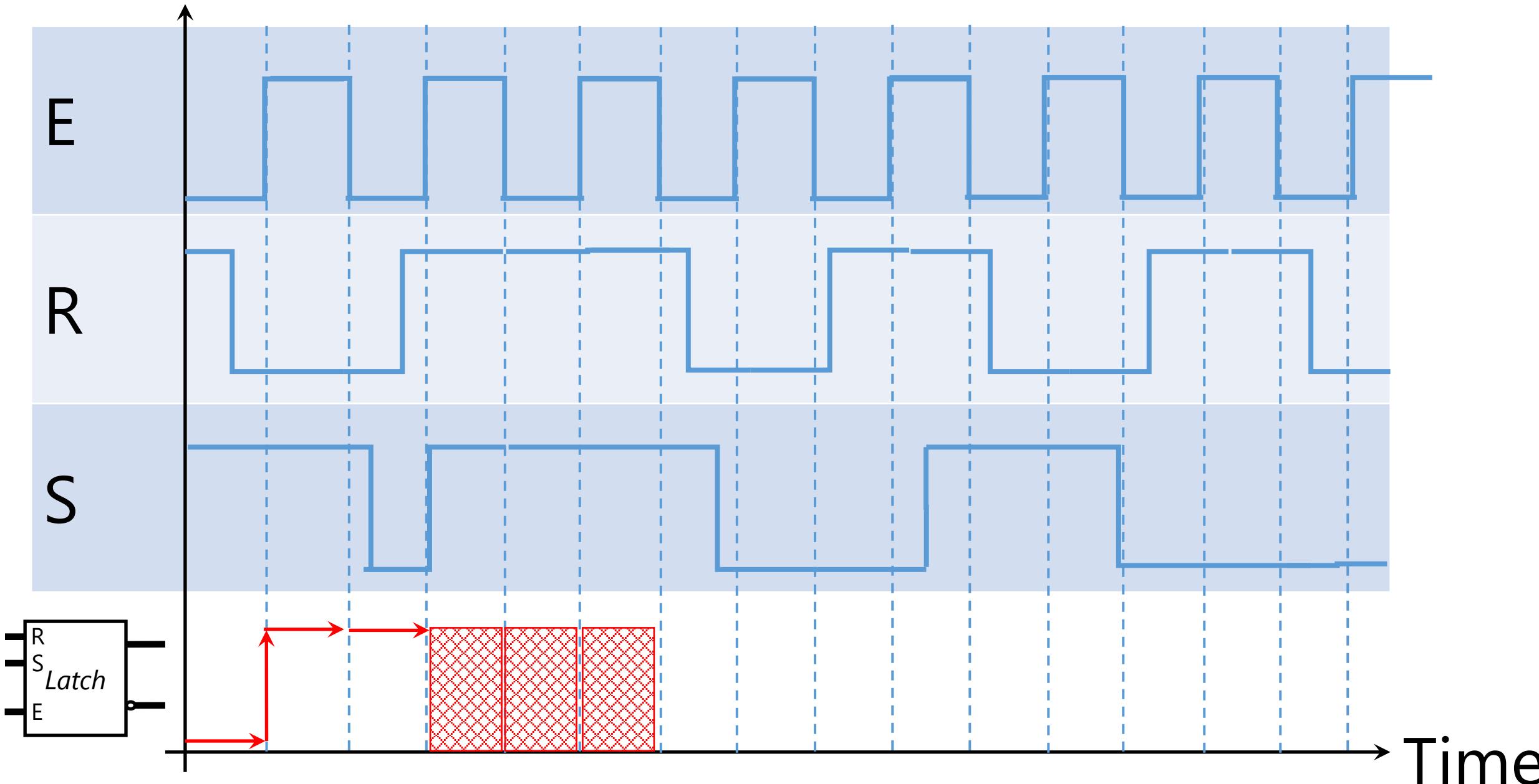
Voltage



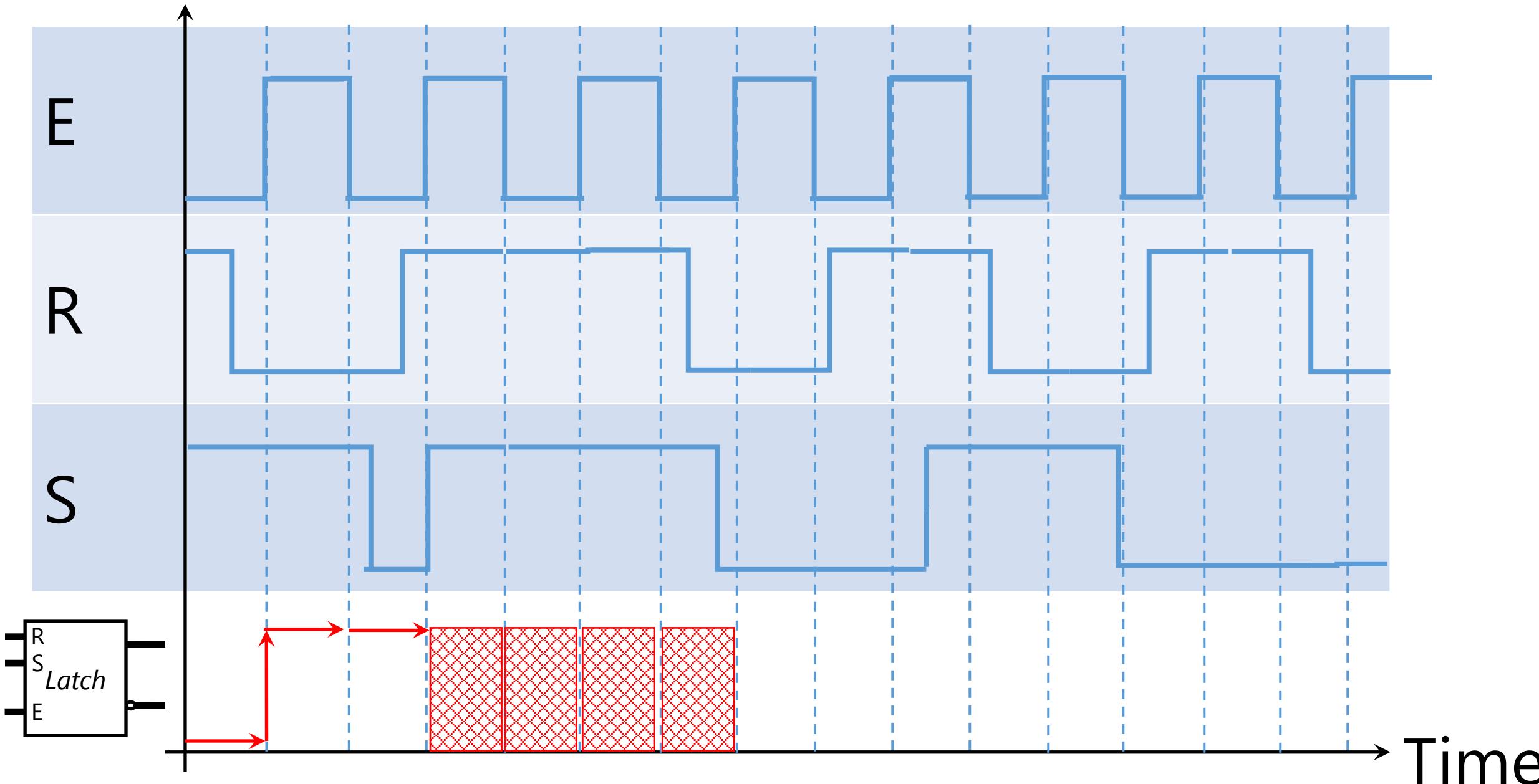
Voltage



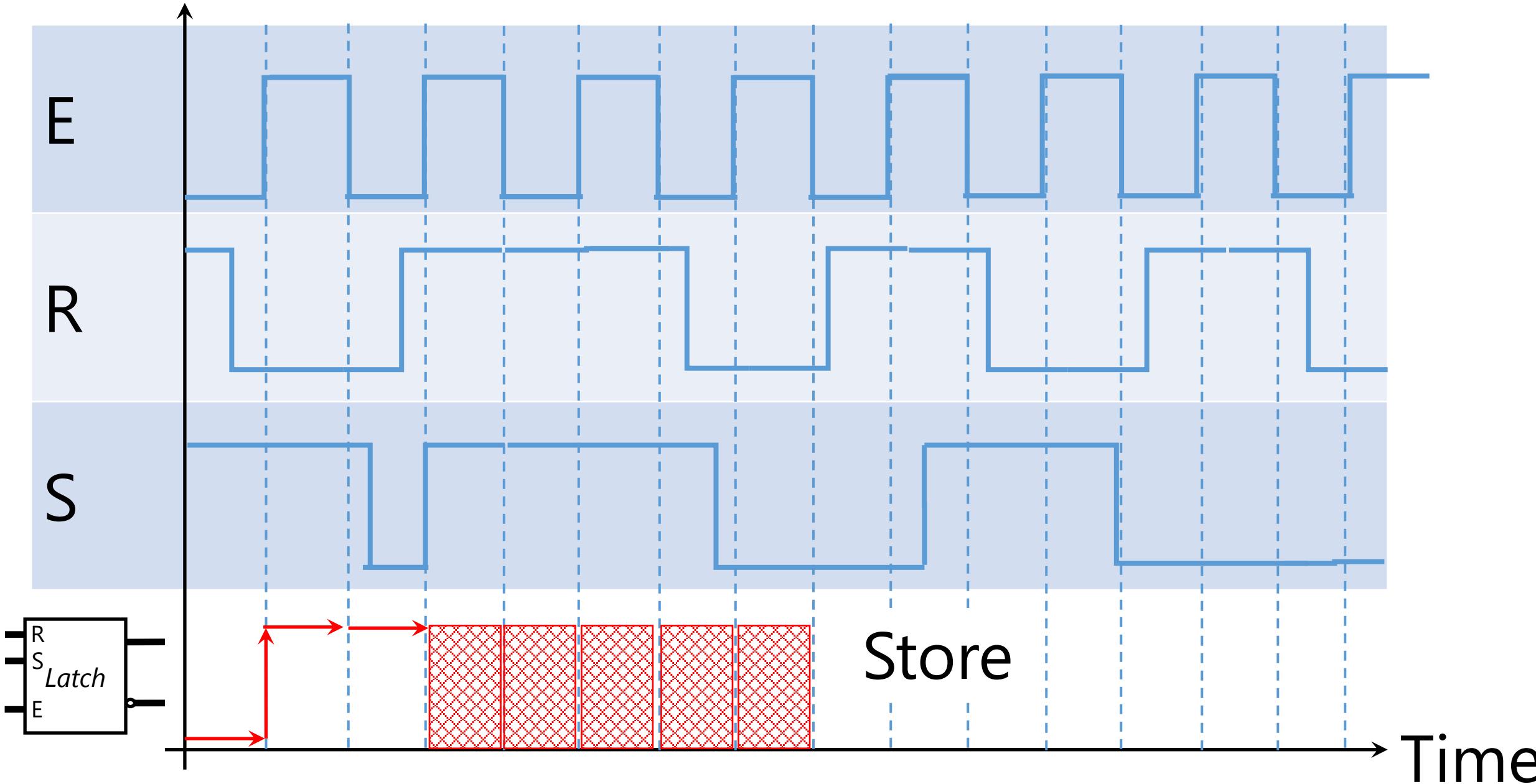
Voltage



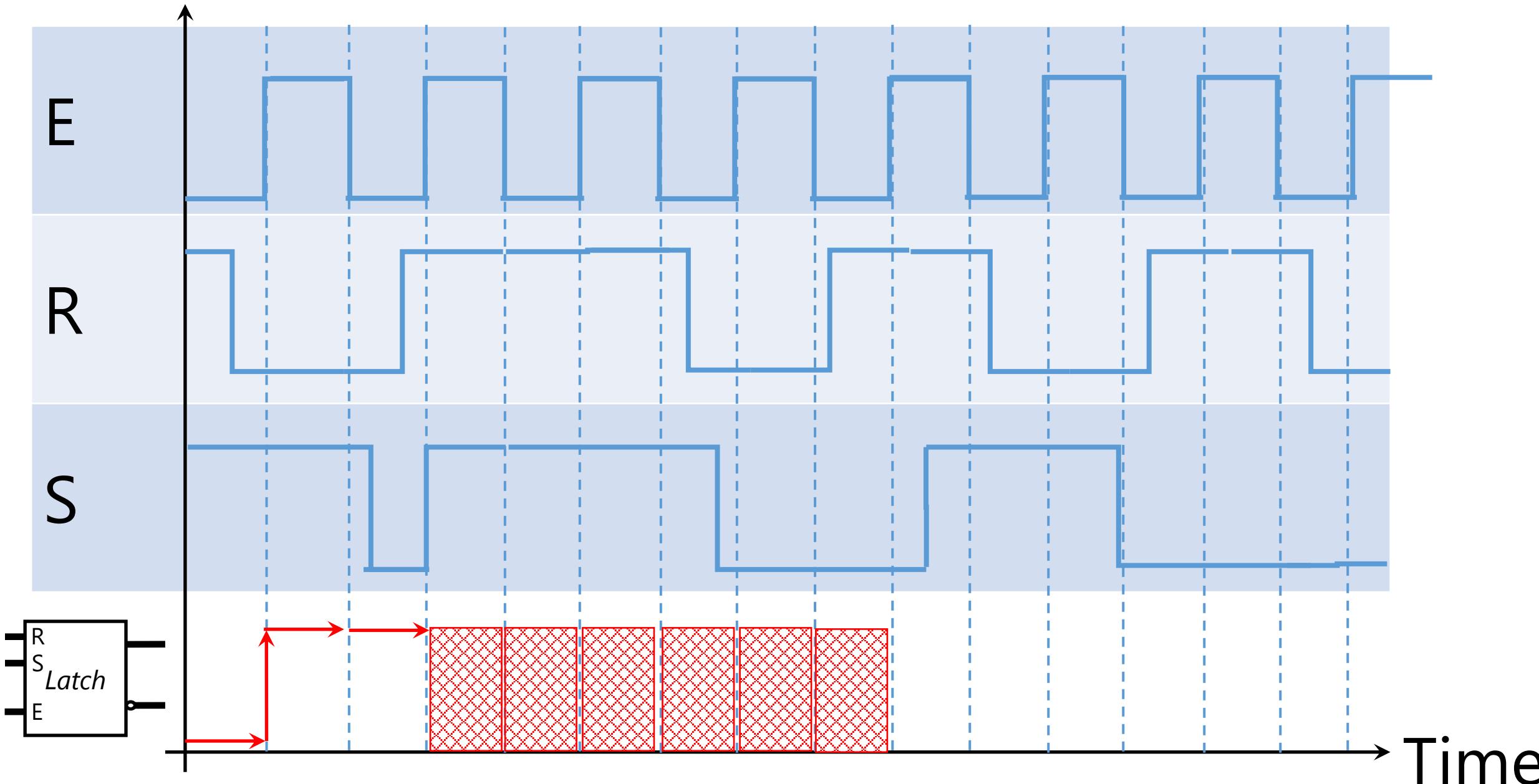
Voltage



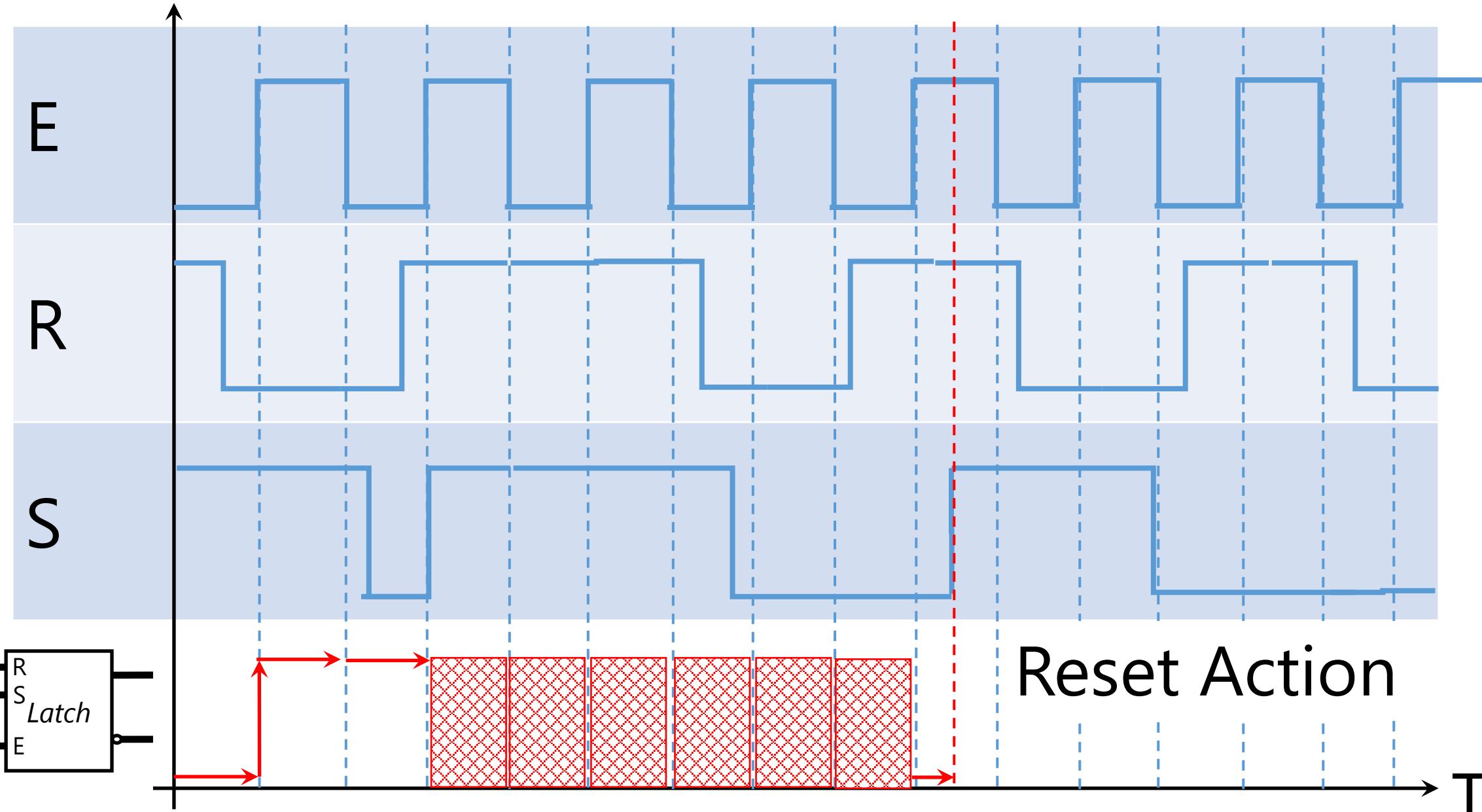
Voltage



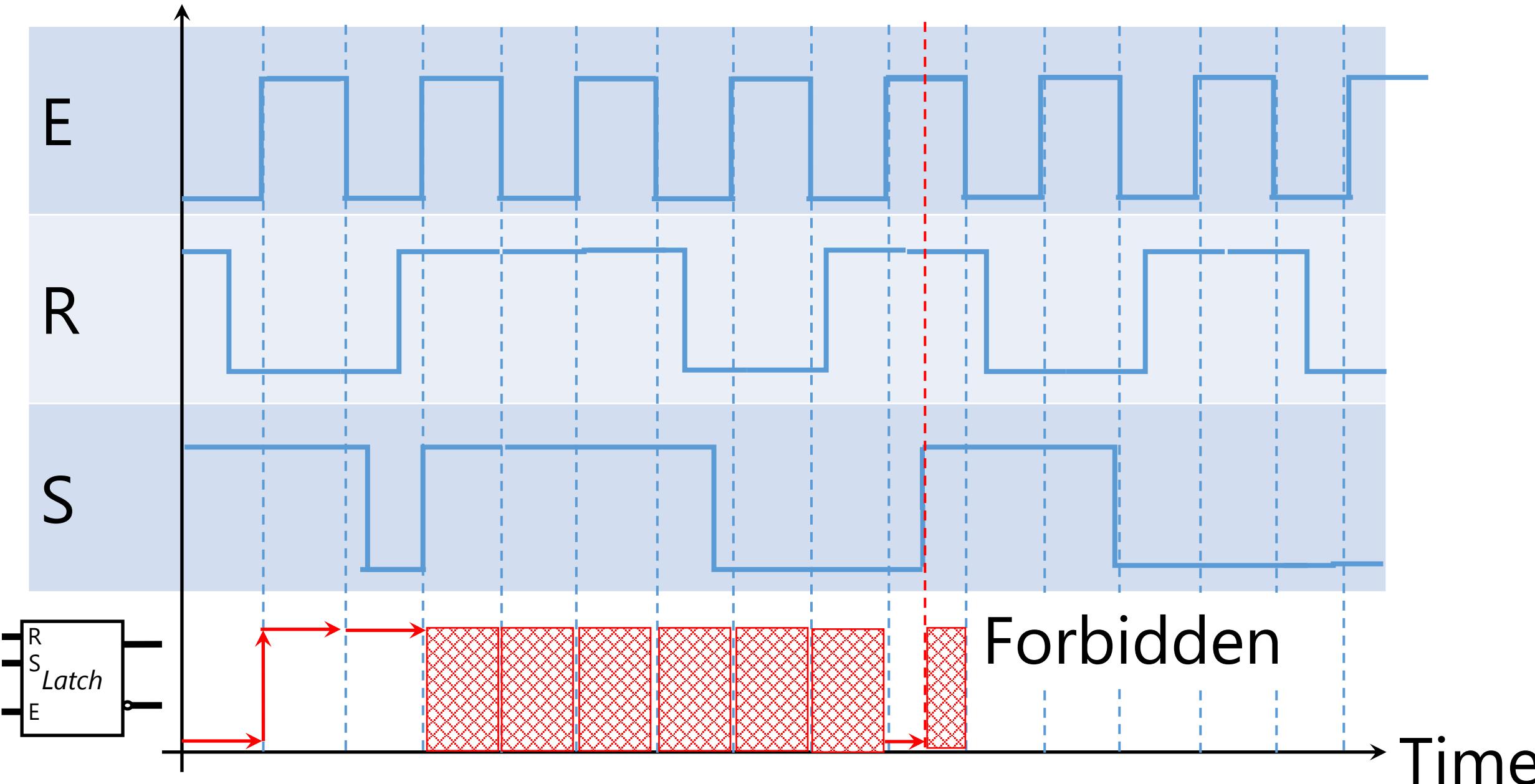
Voltage



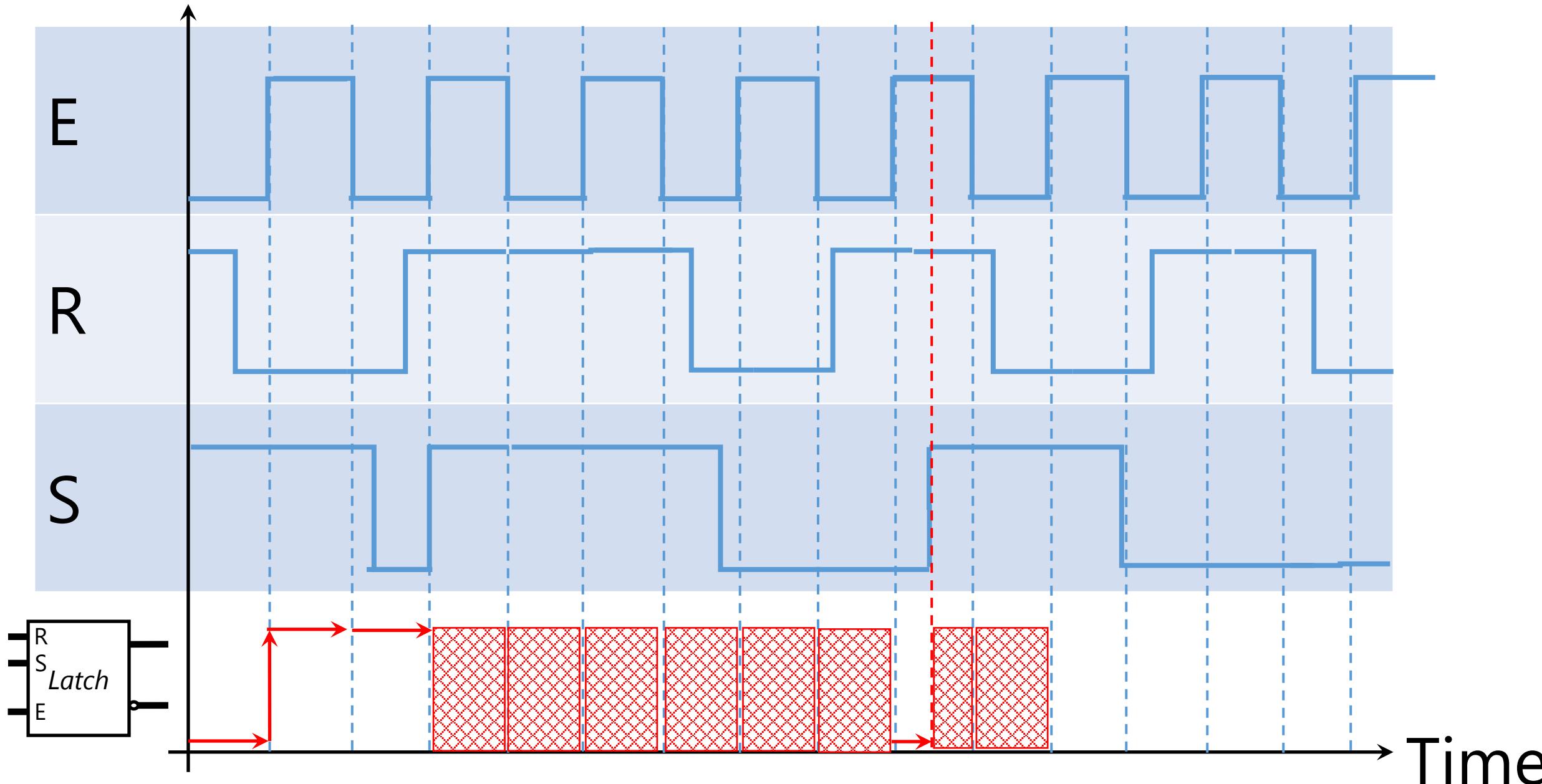
Voltage



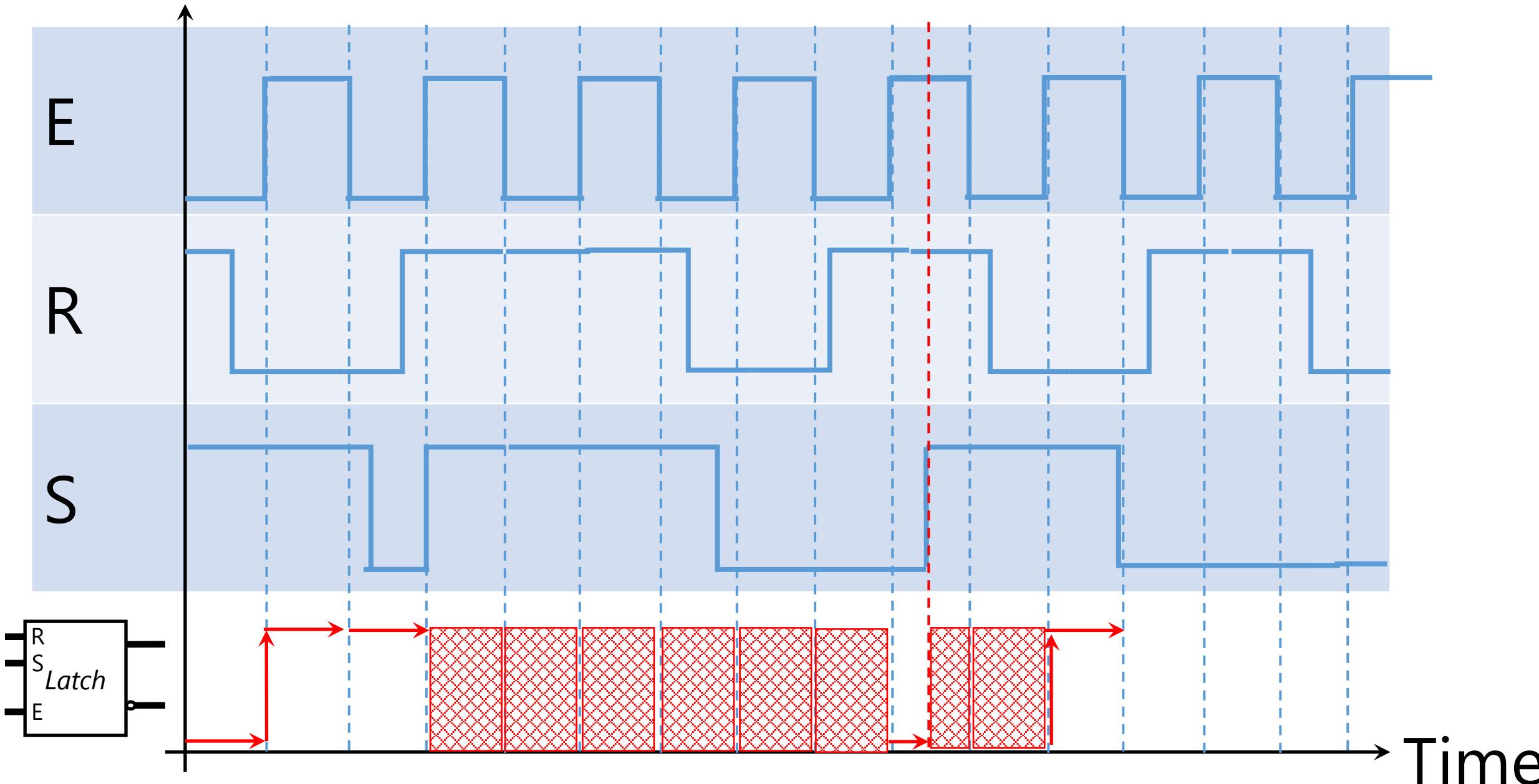
Voltage



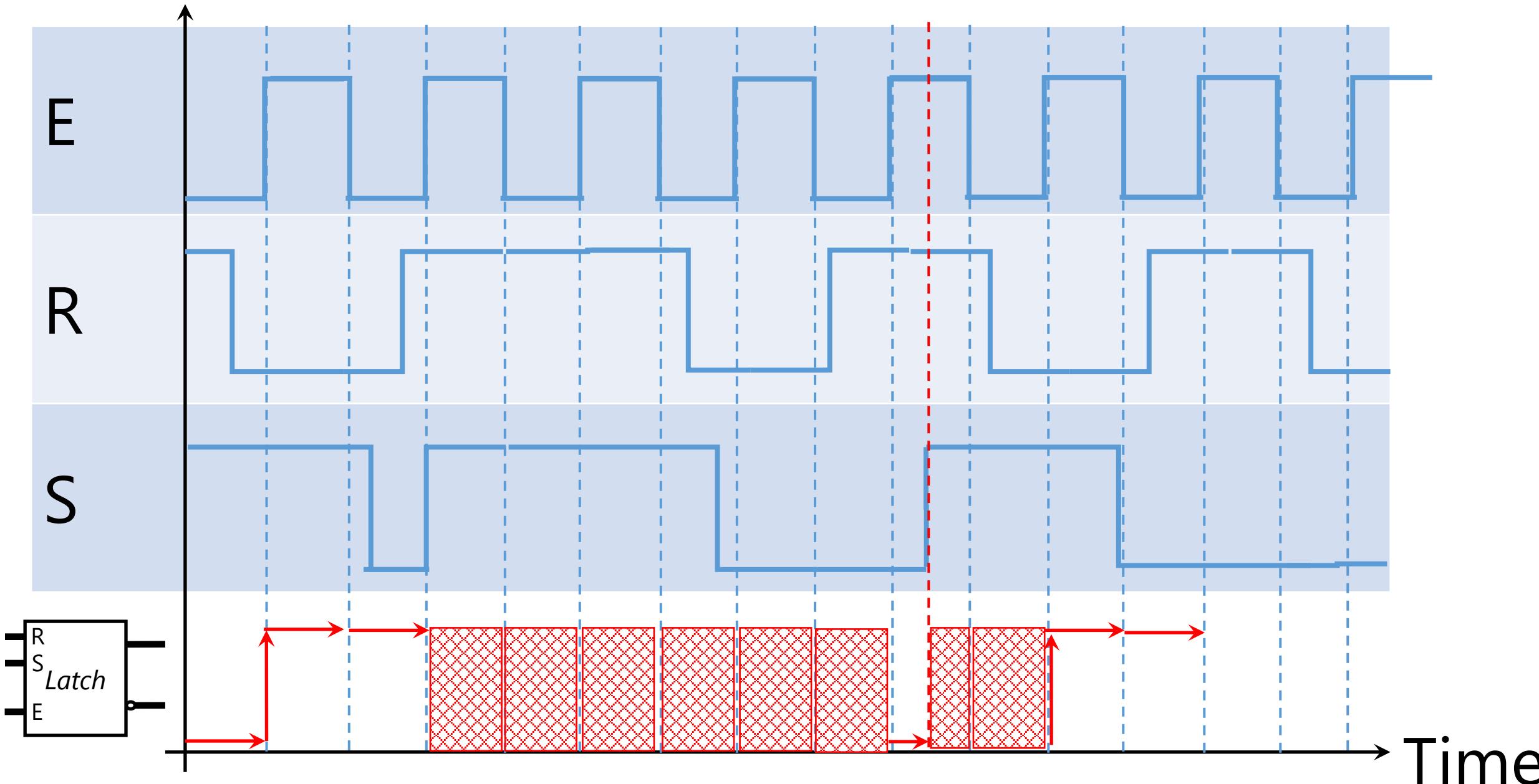
Voltage



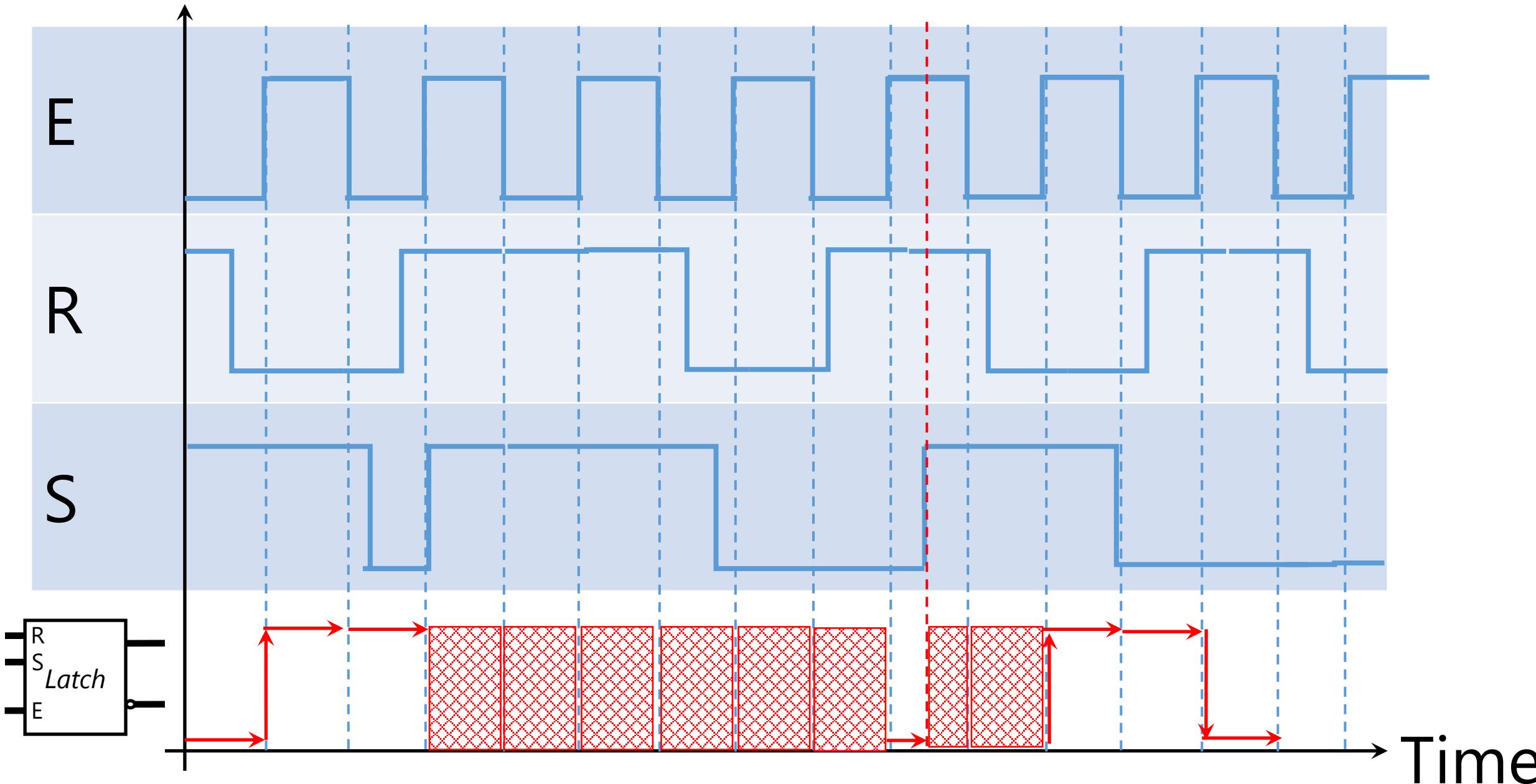
Voltage



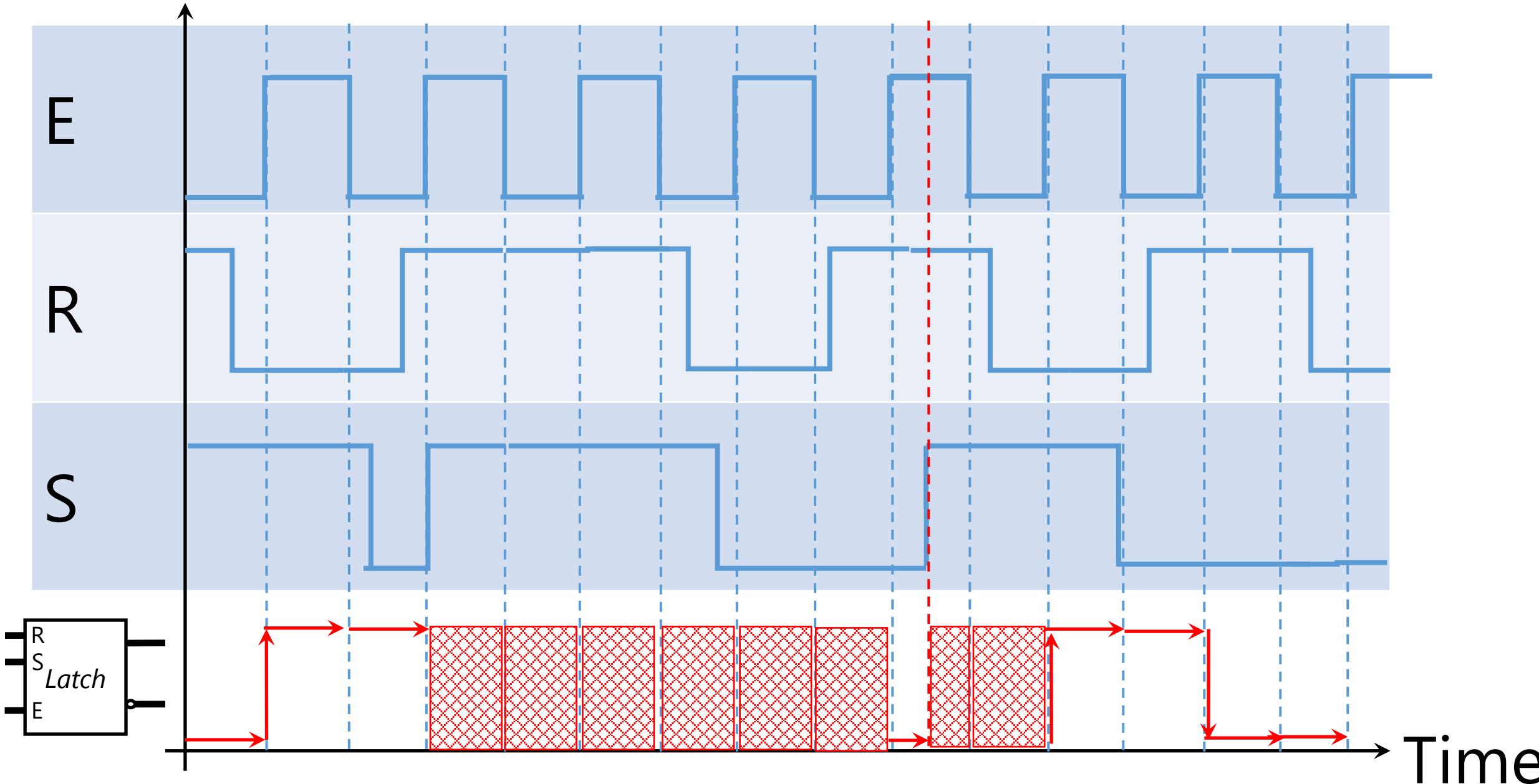
Voltage



Voltage



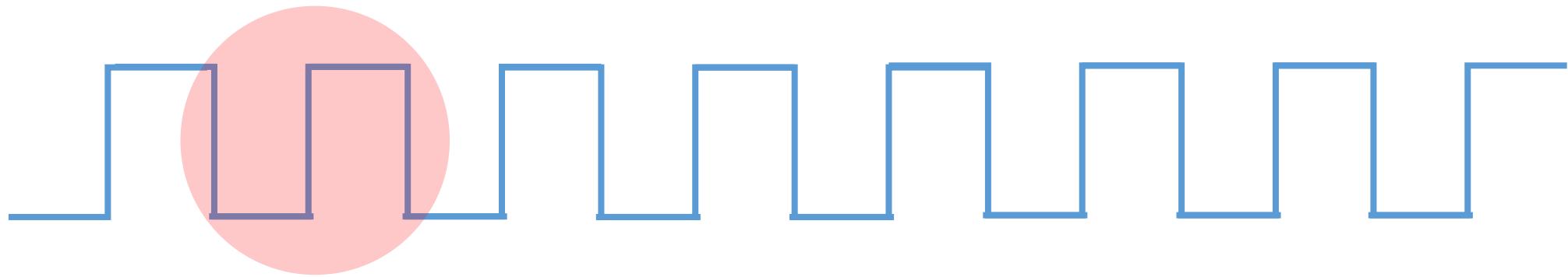
Voltage



Clock

shortened as *clk*

timing device that generates a train of pulses



One period is called **pulse!**

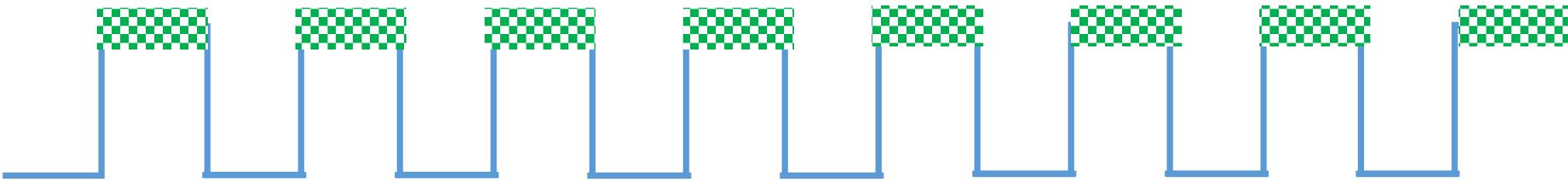
Clock

shortened as *clk*

Synchronize *all* the memory units
when to work

Clock

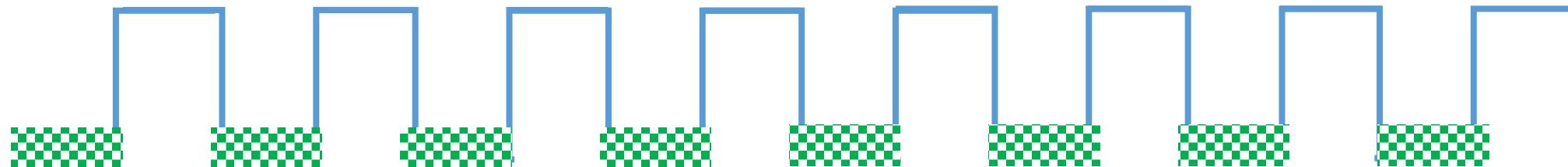
Positive Level (default)





Clock

Negative Level





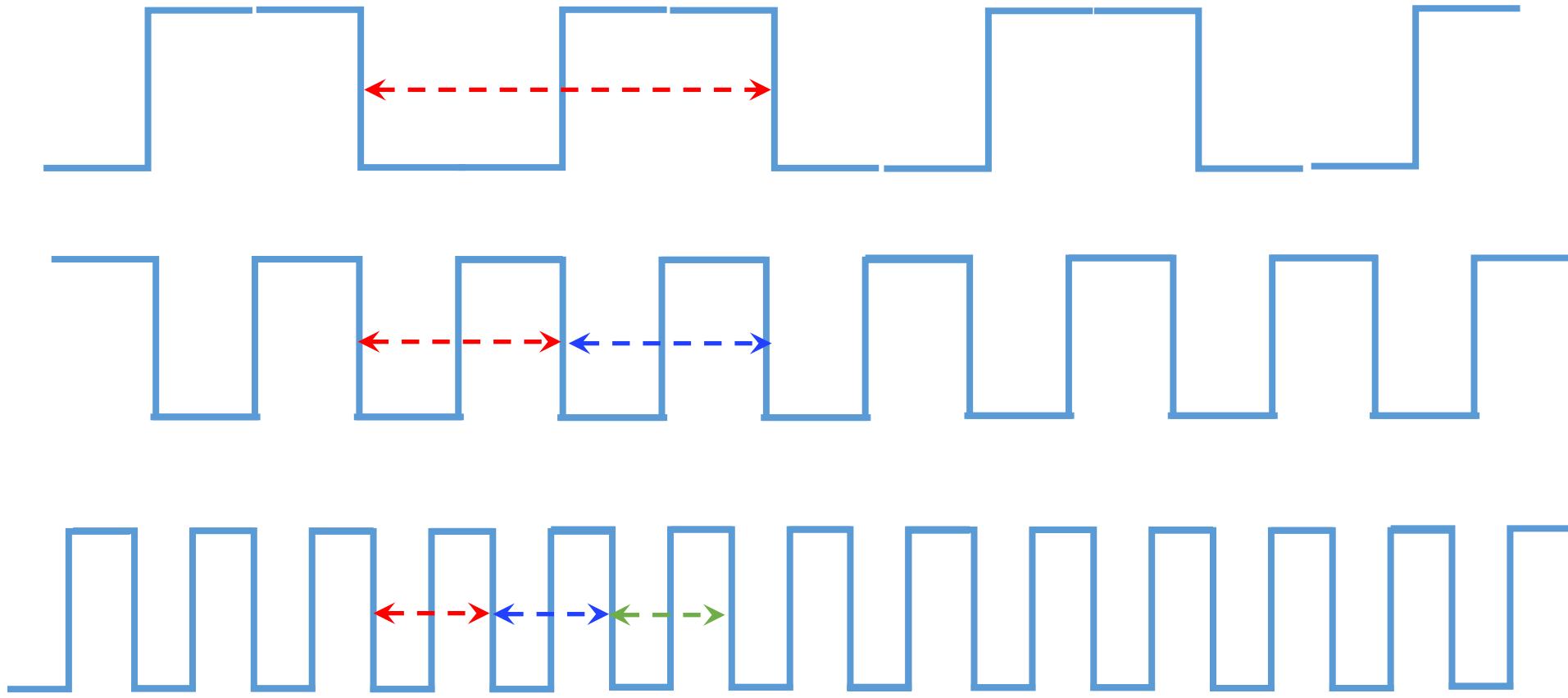
Clock

Frequency (Hz)

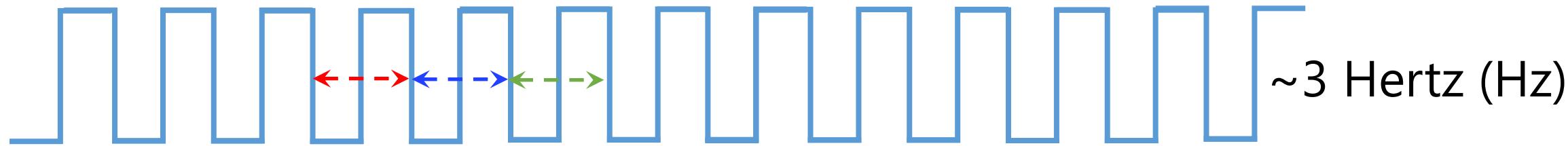
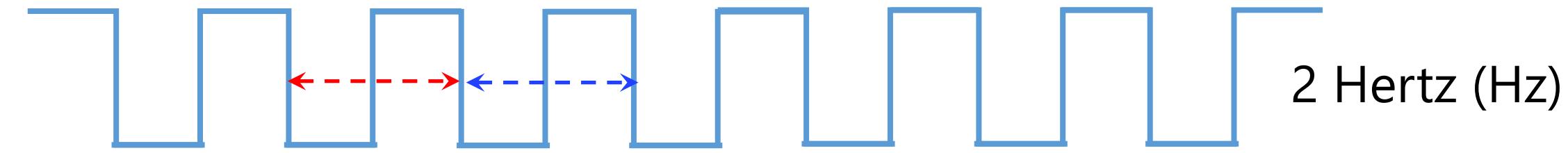
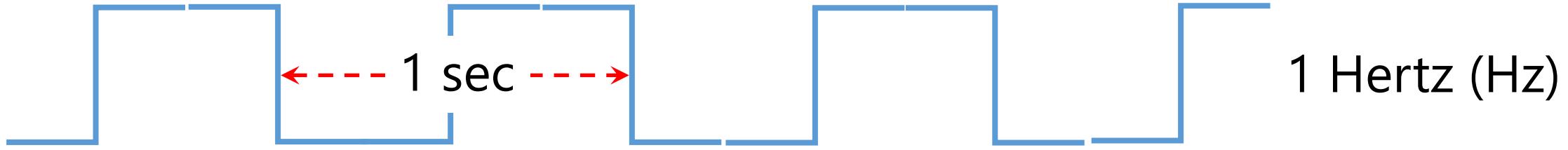


Heinrich Rudolf Hertz

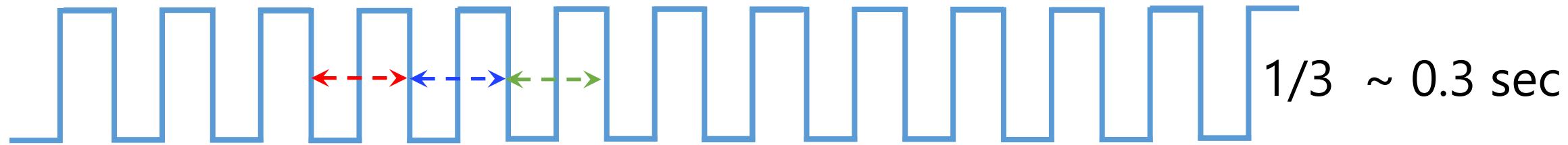
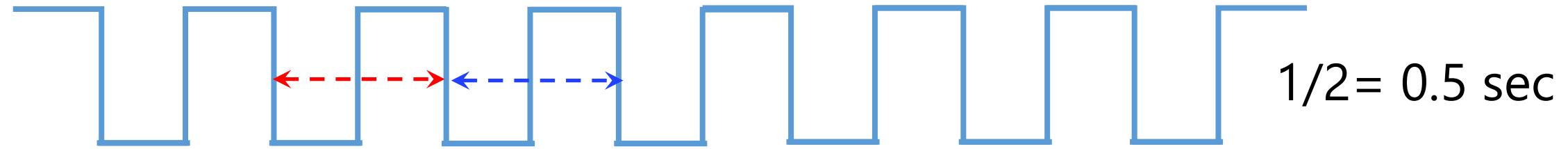
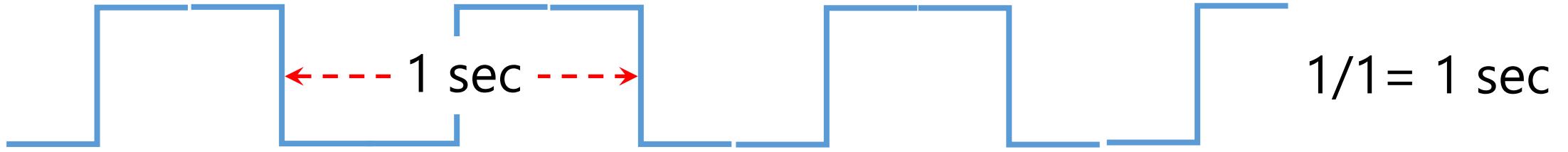
A Metric for Speed



How many pulse in **1** sec?



How many pulse in **1** sec?



How long is one pulse?
 $1/\text{freq. (Hz)}$

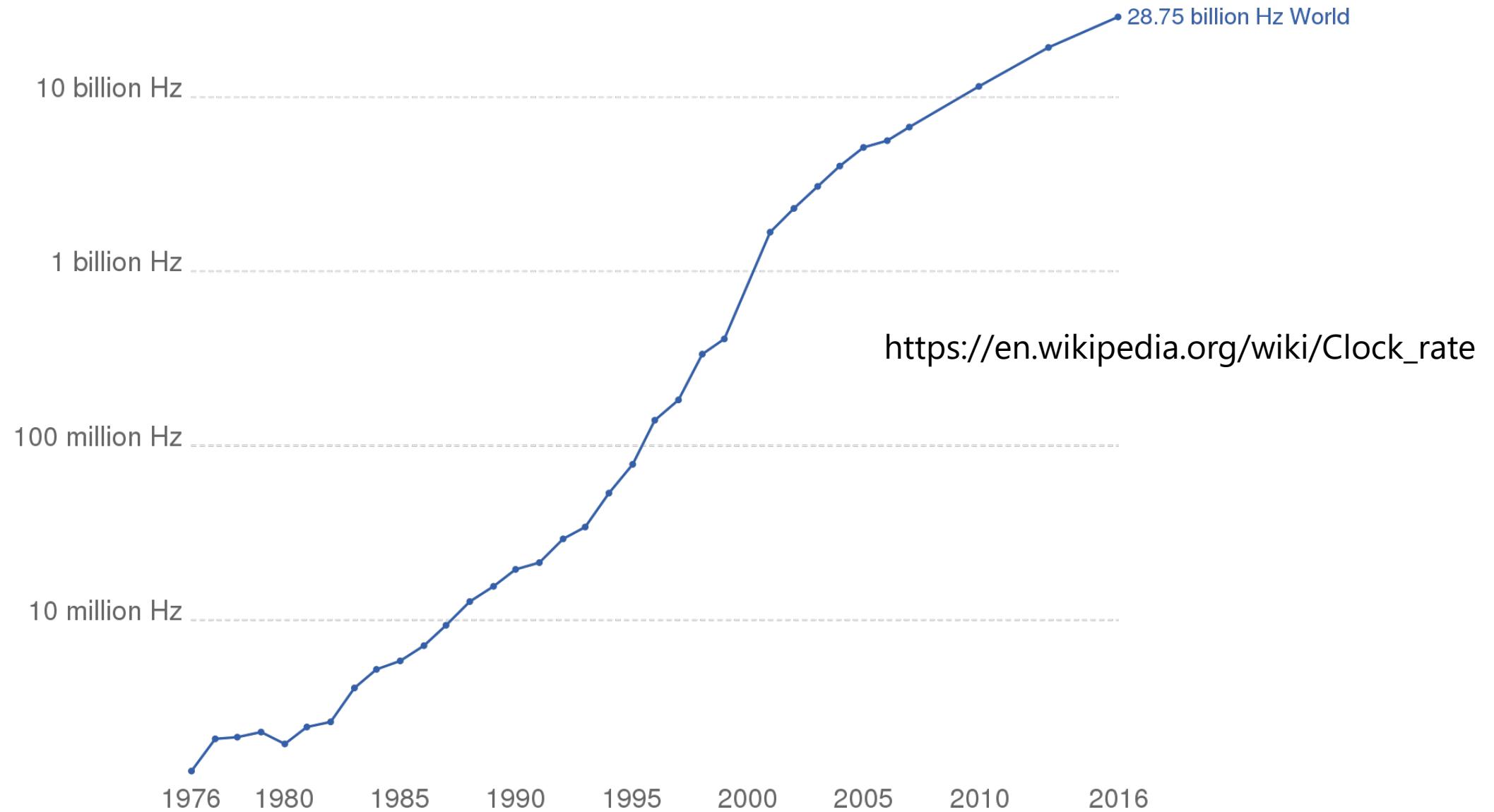


Intel® Xeon® Platinum 8380HL Processor (38.5M Cache, 2.90 GHz)

- 38.5 MB Cache
 - 28 Cores
 - 56 Threads
 - 4.30 GHz Max Turbo Frequency
- 1,000,000,000 (one billion) Hz (hertz)
1,000,000,000 (one billion) pulse per sec!

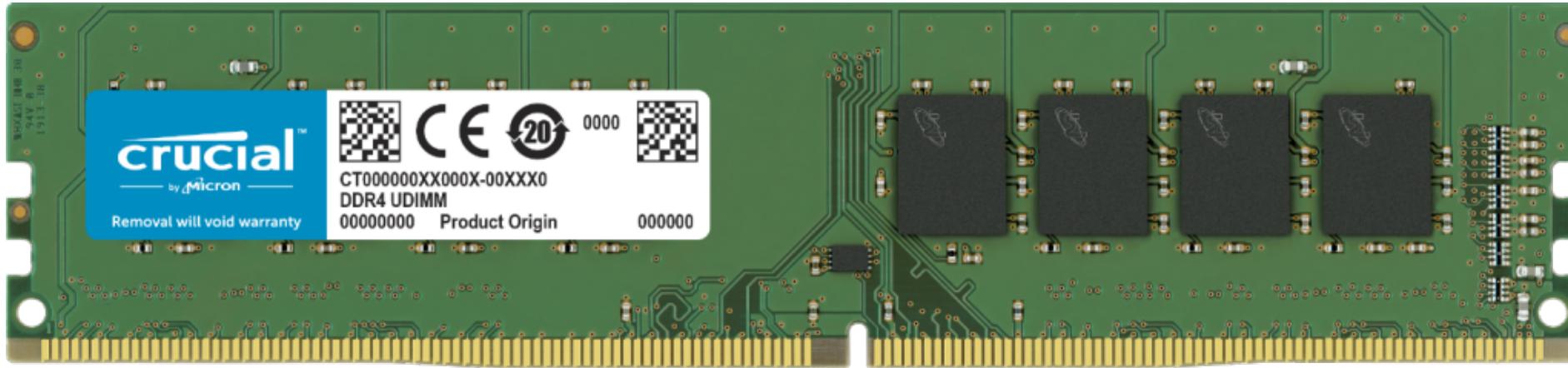
Microprocessor clock speed

Microprocessor clock speed measures the number of pulses per second generated by an oscillator that sets the tempo for the processor. It is measured in hertz (pulses per second).



Source: Ray Kurzweil (2005, updated to 2016). *The Singularity Is Near: When Humans Transcend Biology*.

https://en.wikipedia.org/wiki/Clock_rate



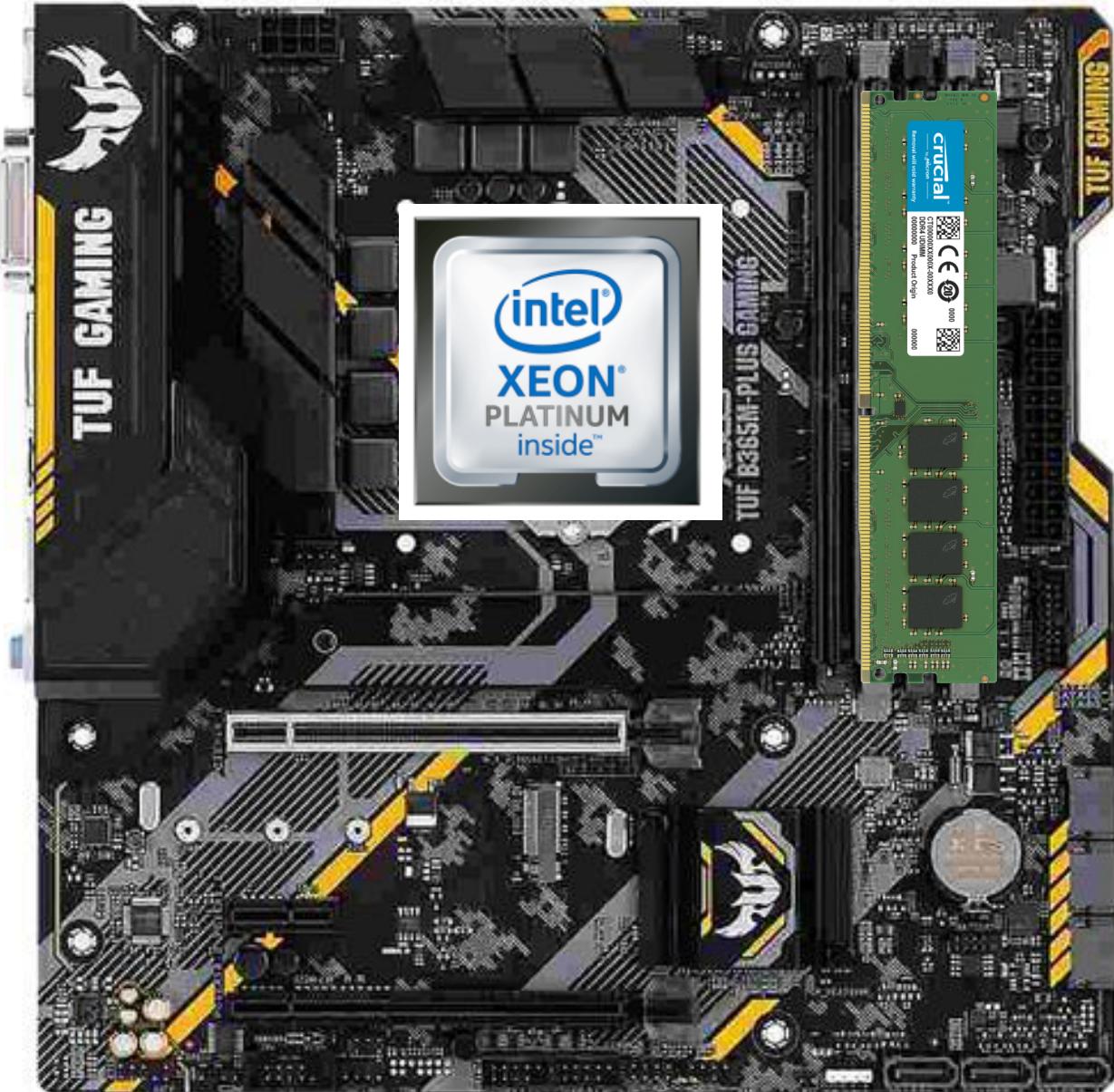
Speed - DDR4-2666



Warranty - Manufact

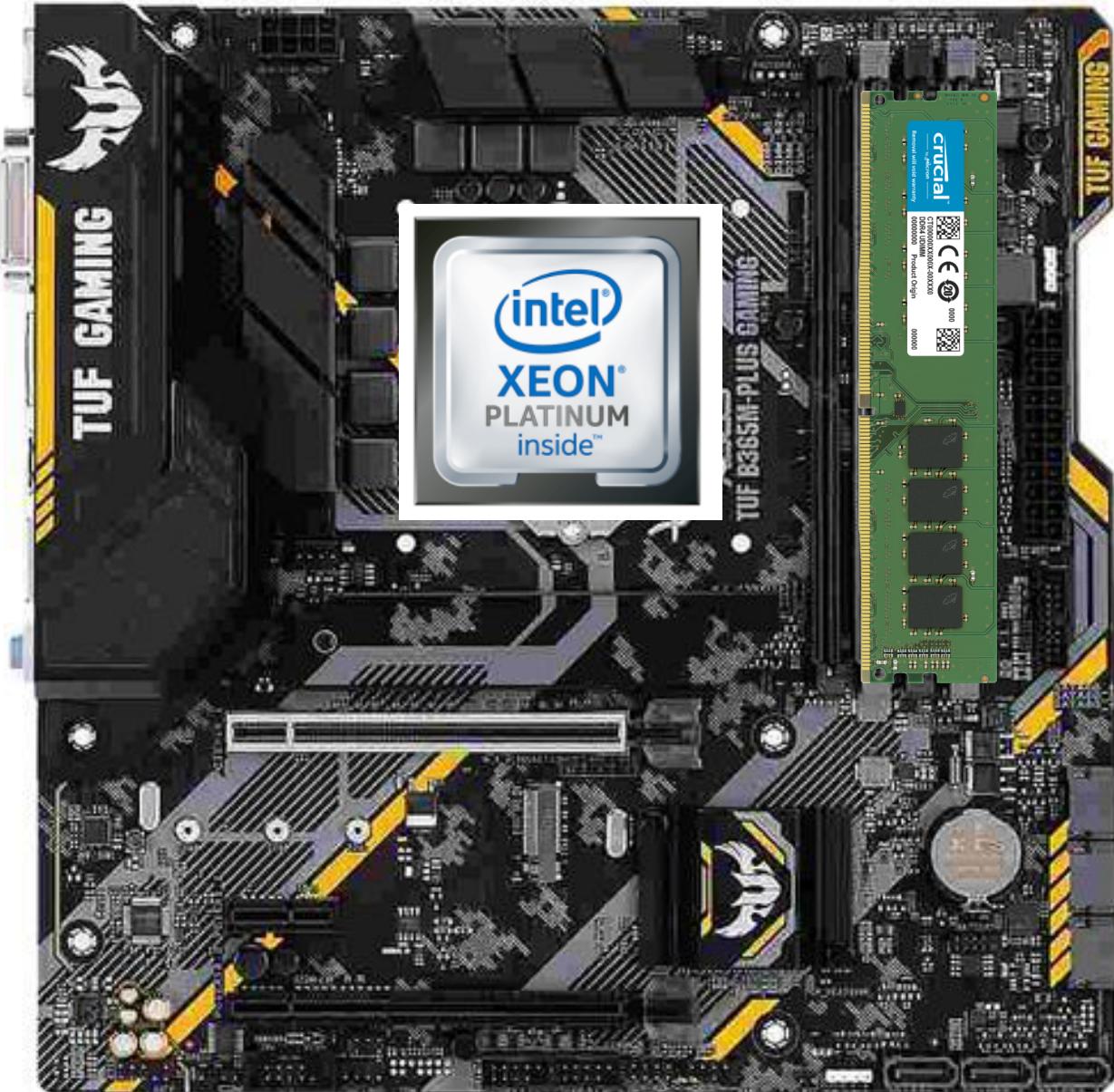
CAS latency - 19

The number of transfers per clock cycle times the clock frequency,
expressed as MegaTransfers per second.



CPU: X Hz
Memory: Y Hz
Mainboard (BUS): Z Hz

Final Speed?



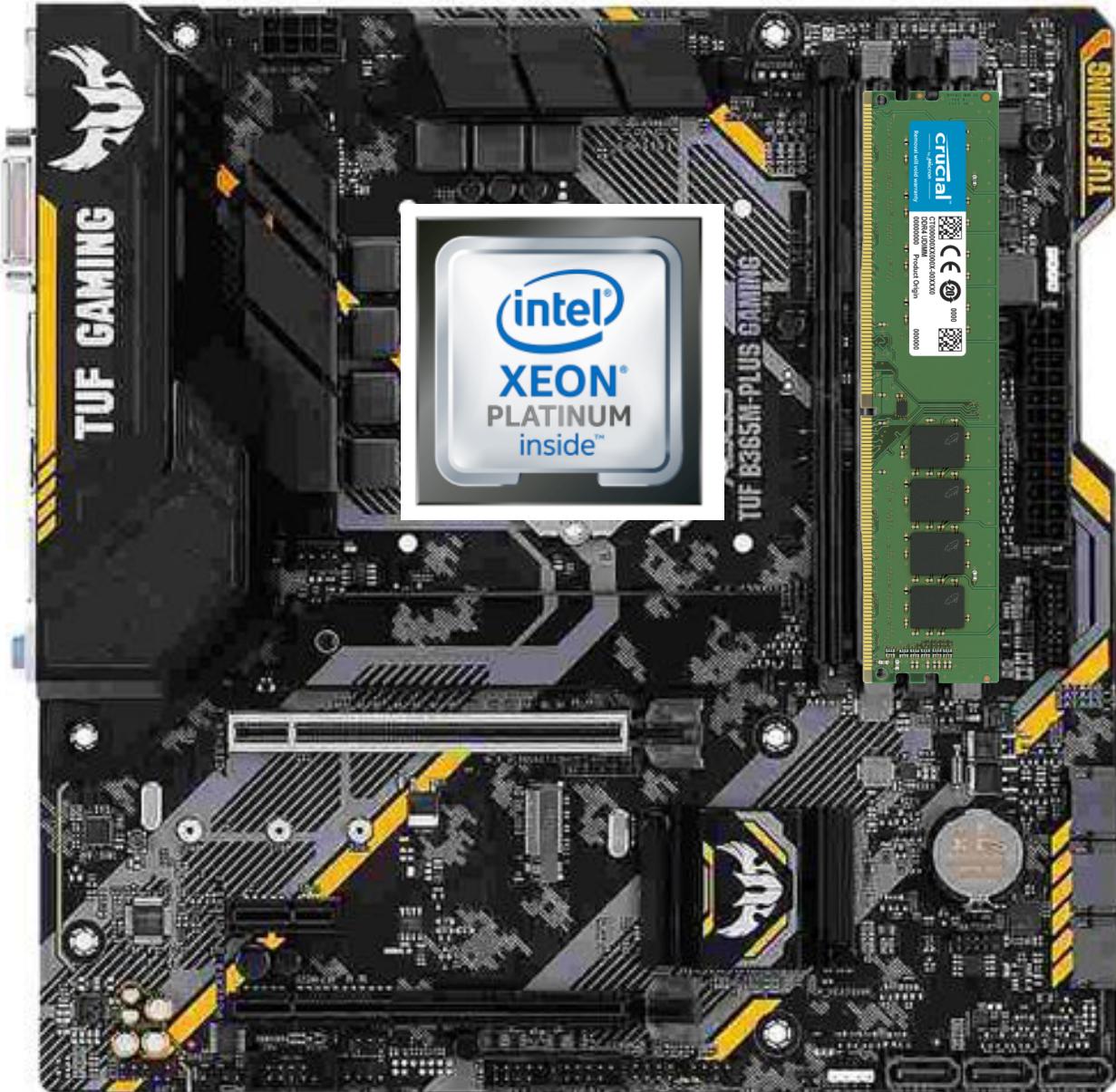
CPU: X Hz
Memory: Y Hz
Mainboard (BUS): Z Hz

At market:

$$X > Y = Z$$

$$X = 2.9 \text{ GHz}$$

$$Y = Z = 2.6 \text{ GHz}$$



CPU: X Hz
Memory: Y Hz
Mainboard (BUS): Z Hz

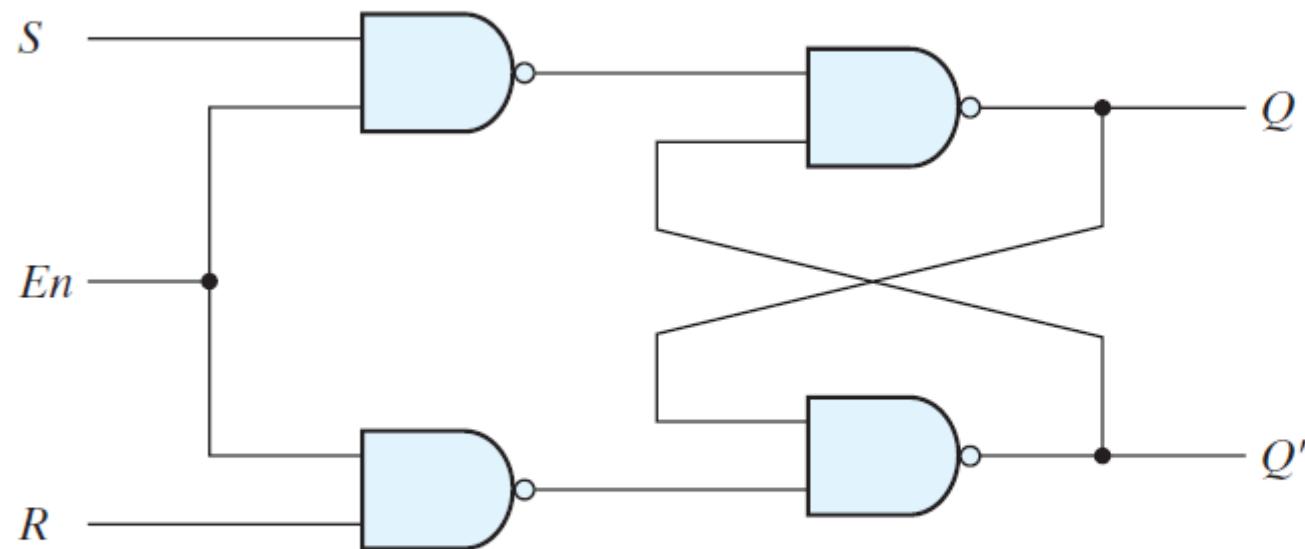
Final Speed:

CPU internal: X
CPU external \leftrightarrow Memory
 $Y=Z= 2.6\text{GHz}$

Overclock?

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

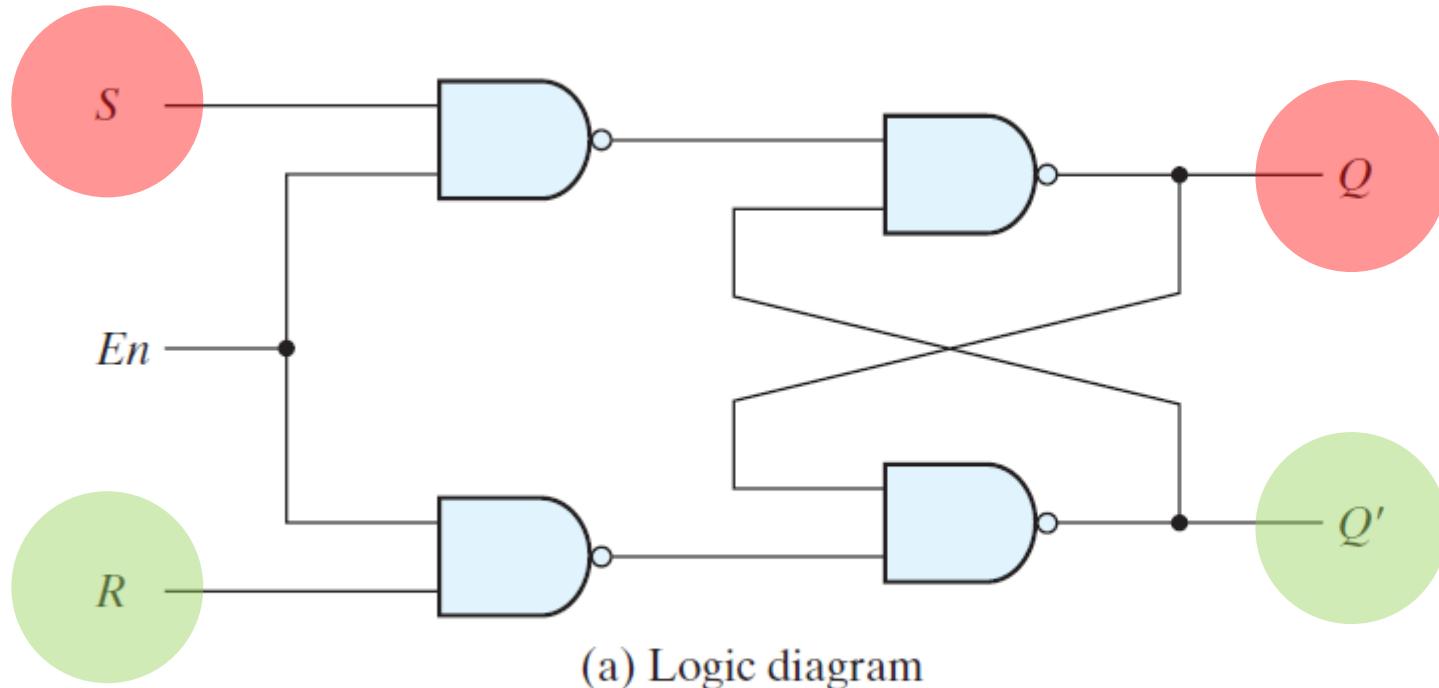
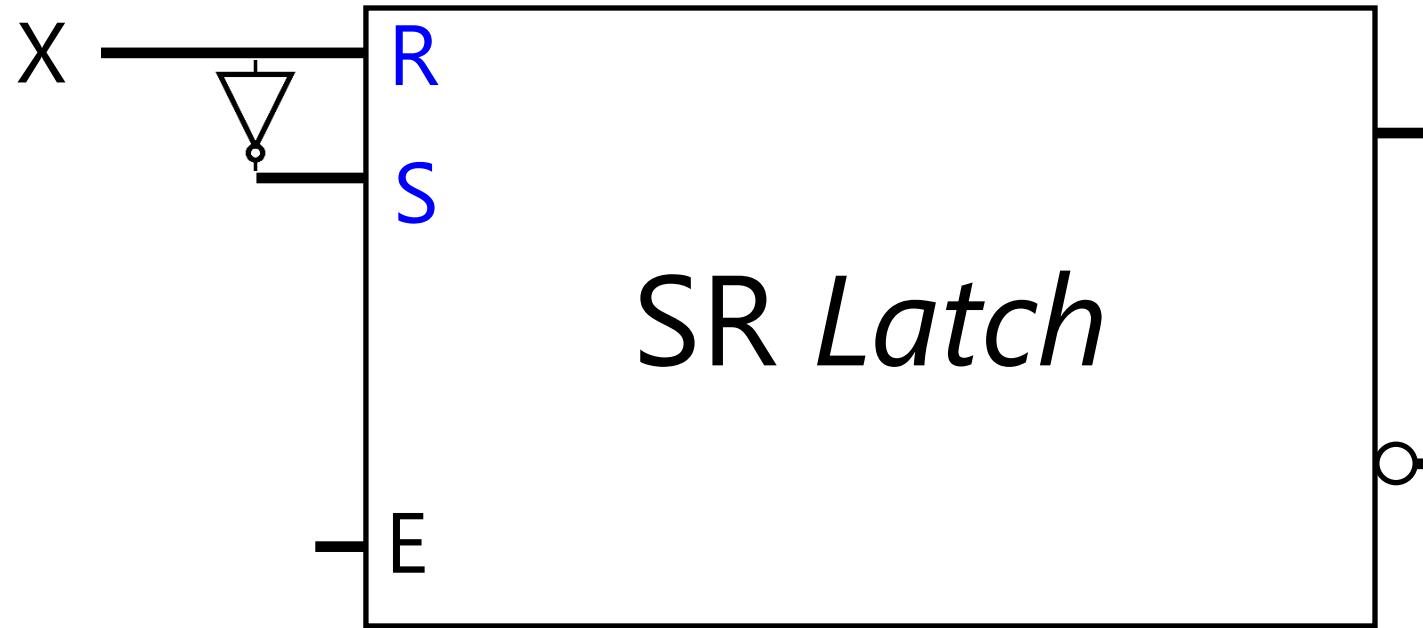
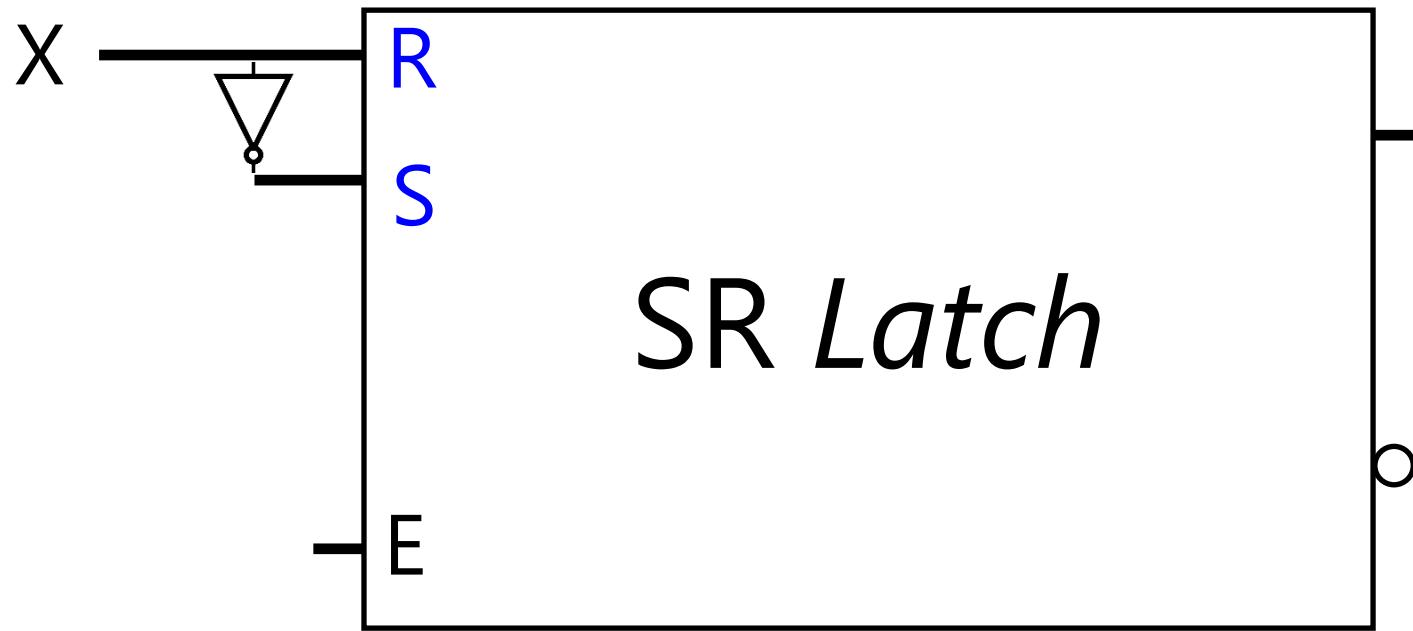


FIGURE 5.5
SR latch with control input

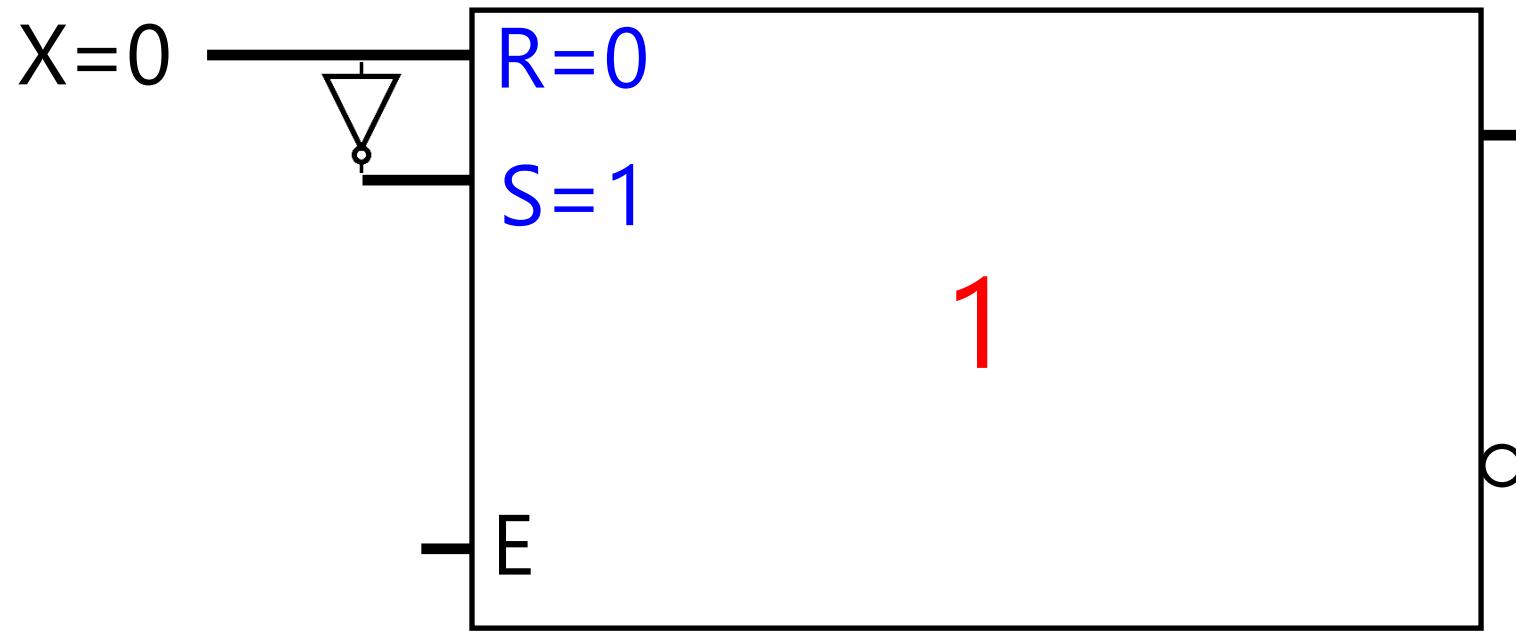
<i>En</i>	<i>S</i>	<i>R</i>	Next state of <i>Q</i>
0	X	X	No change
1	0	0	No change
1	0	1	<i>Q</i> = 0; reset state
1	1	0	<i>Q</i> = 1; set state
1	1	1	Indeterminate

(b) Function table

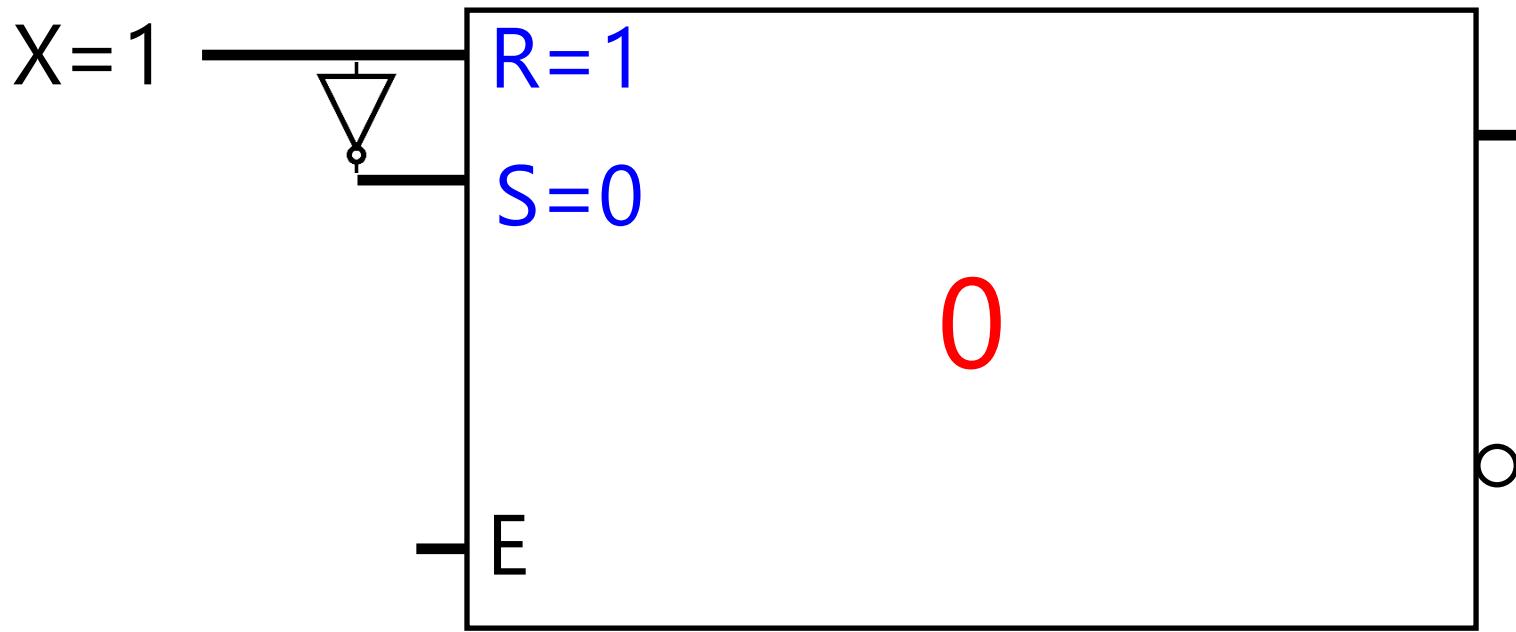




X	E	S	R	Q	Q'
0	0	X	X	Q_t	Q'_t
1	0	X	X	Q_t	Q'_t
0	1	1	0	1	0
1	1	0	1	0	1
Never happens		1	1	*	*



X	E	S	R	Q	Q'
0	0	\times	\times	Q_t	Q'_t
1	0	\times	\times	Q_t	Q'_t
0	1	1	0	1	0
1	1	0	1	0	1
Never happens		1	1	*	*

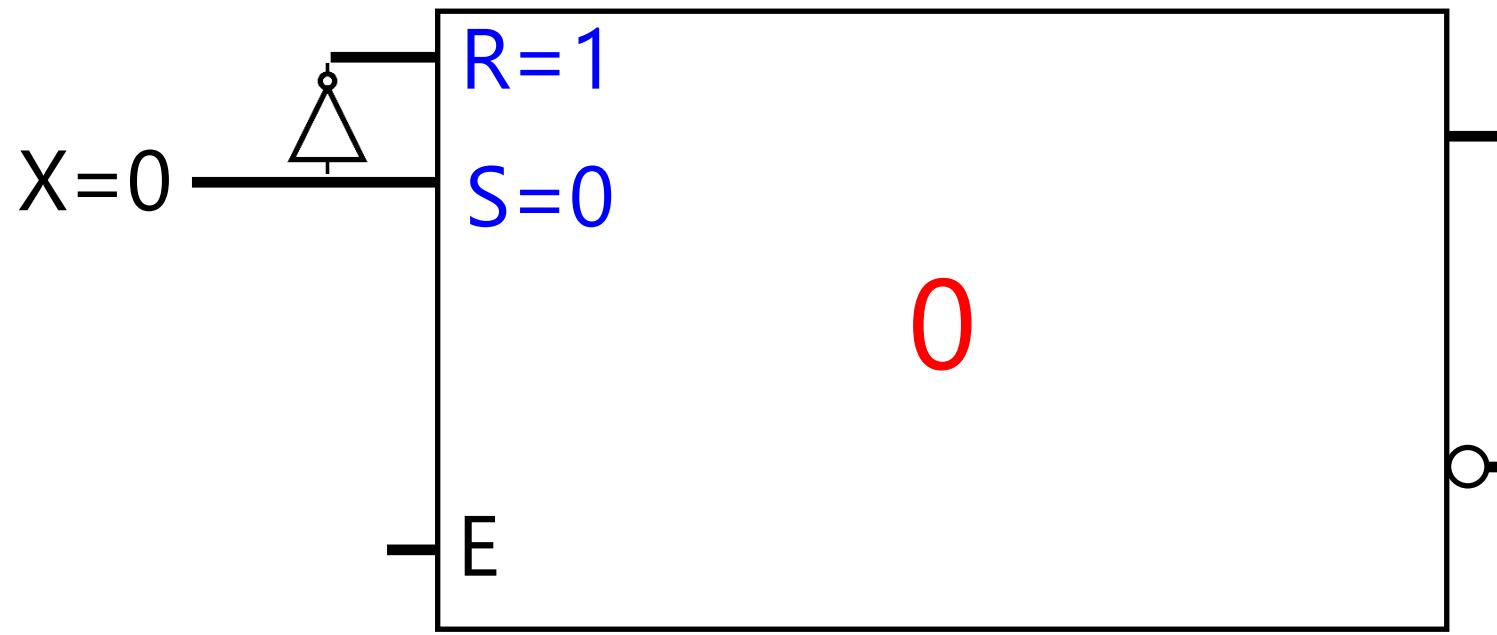


X	E	S	R	Q	Q'
0	0	\times	\times	Q_t	Q'_t
1	0	\times	\times	Q_t	Q'_t
0	1	1	0	1	0
1	1	0	1	0	1
Never happens		1	1	*	*

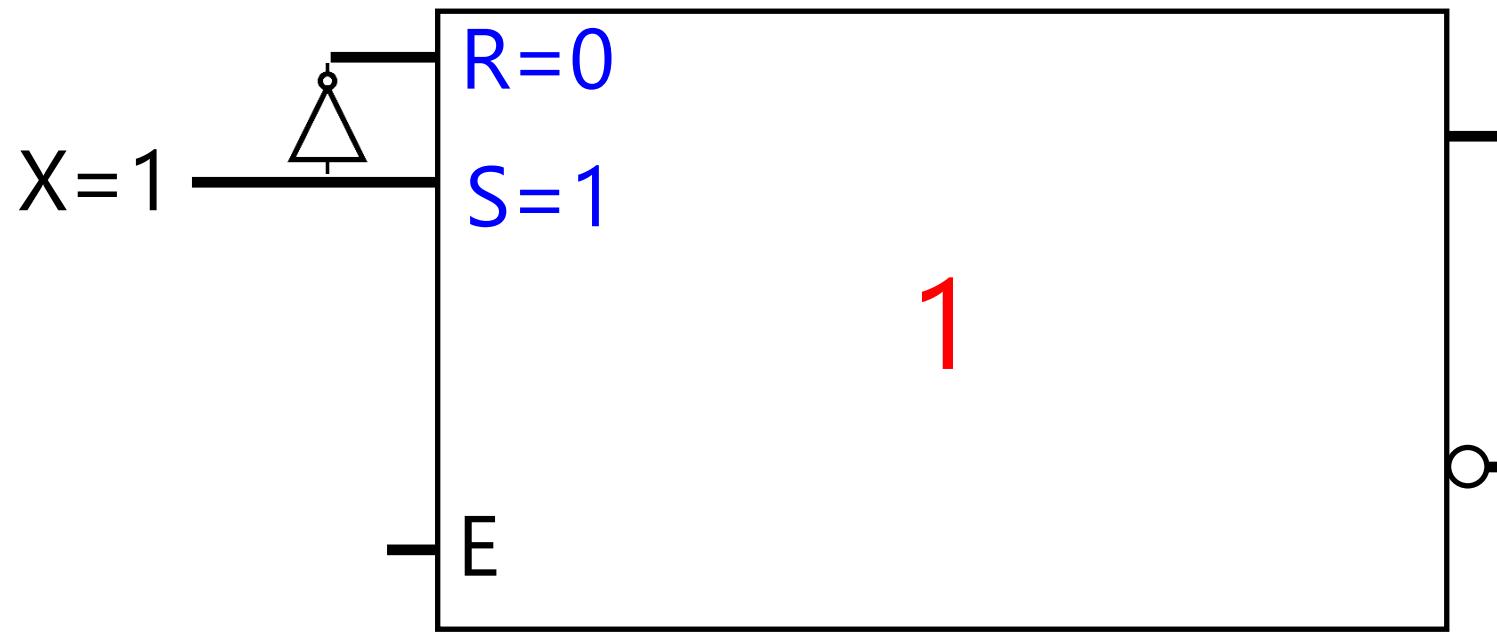




X	E	S	R	Q	Q'
0	0	x	x	Q_t	Q'_t
1	0	x	x	Q_t	Q'_t
0	1	0	1	0	1
1	1	1	0	1	0
Never happens		1	1	*	*



X	E	S	R	Q	Q'
0	0	\times	\times	Q_t	Q'_t
1	0	\times	\times	Q_t	Q'_t
0	1	0	1	0	1
1	1	1	0	1	0
Never happens		1	1	*	*



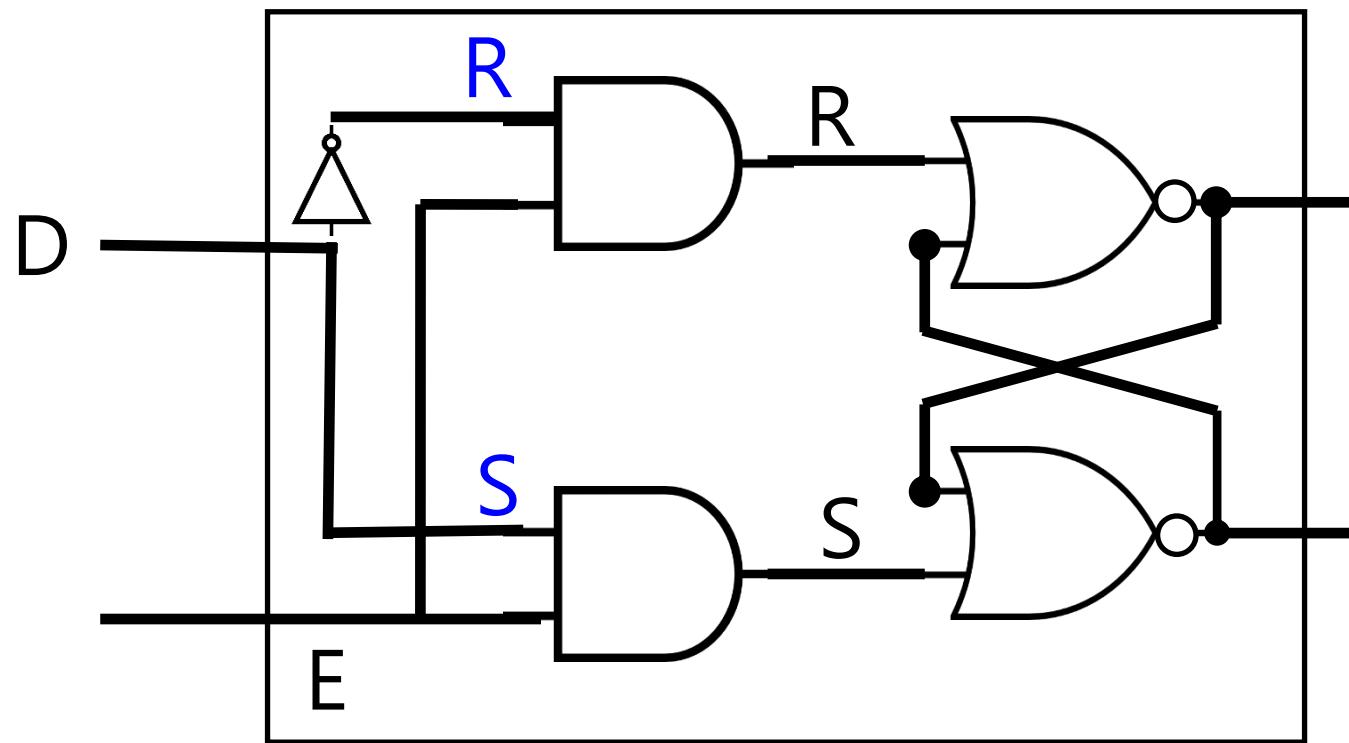
X	E	S	R	Q	Q'
0	0	\times	\times	Q_t	Q'_t
1	0	\times	\times	Q_t	Q'_t
0	1	0	1	0	1
1	1	1	0	1	0
Never happens		1	1	*	*

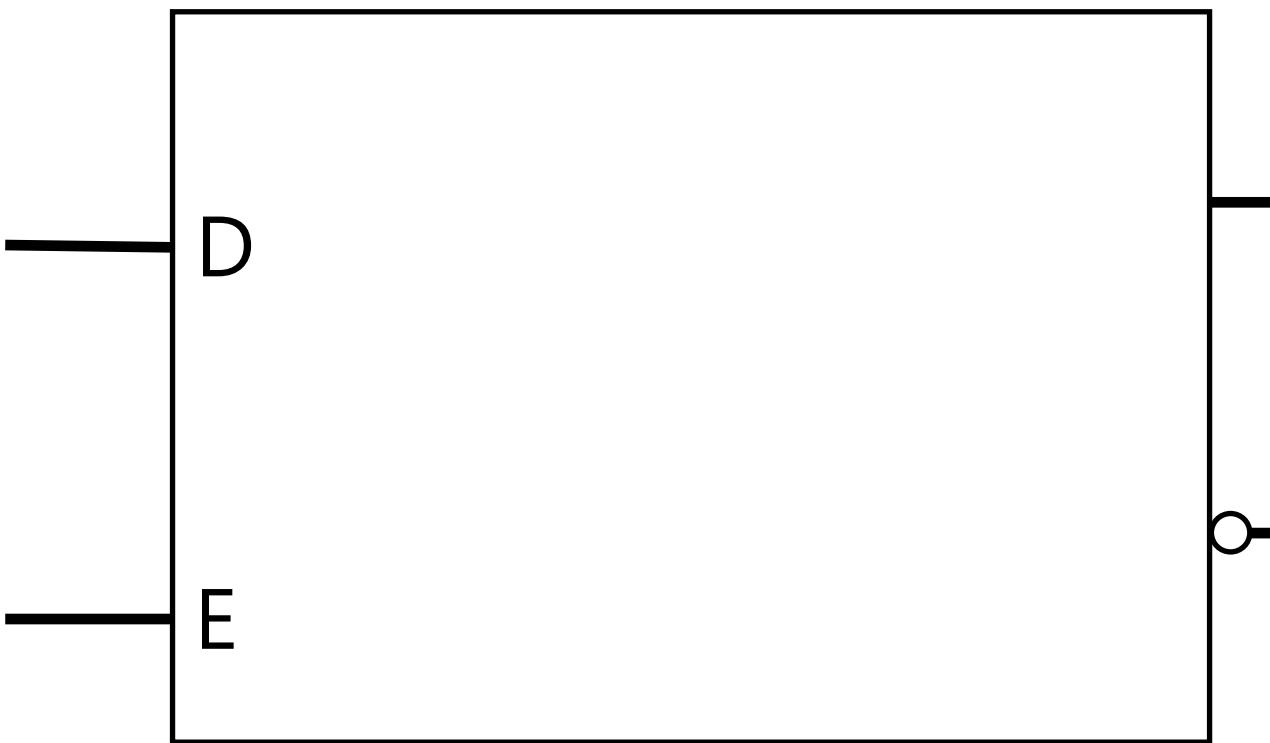


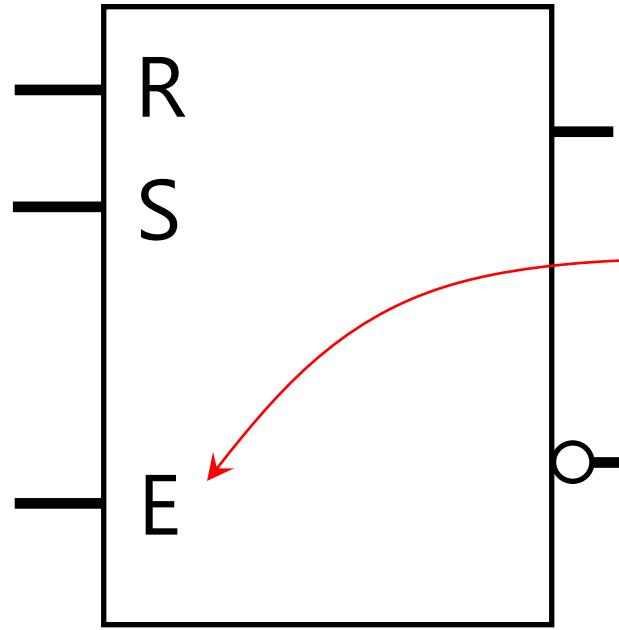
D	E	Q
0	0	Q_t
1	0	Q_t
0	1	0
1	1	1



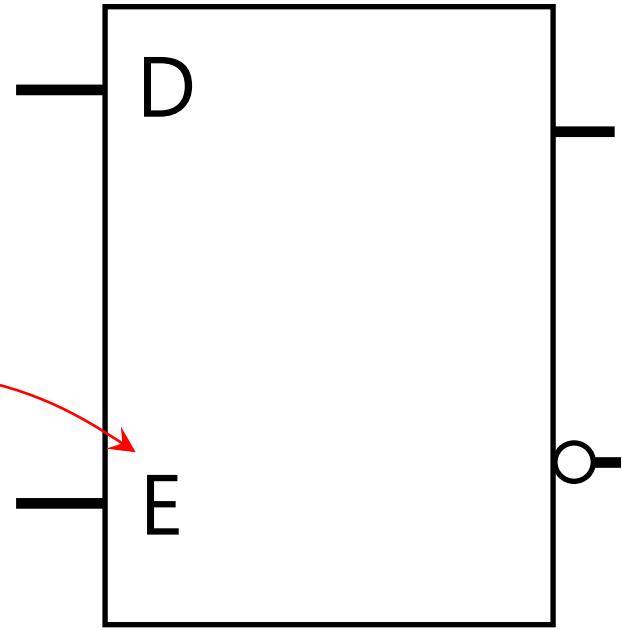
D	Q
0	0
1	1





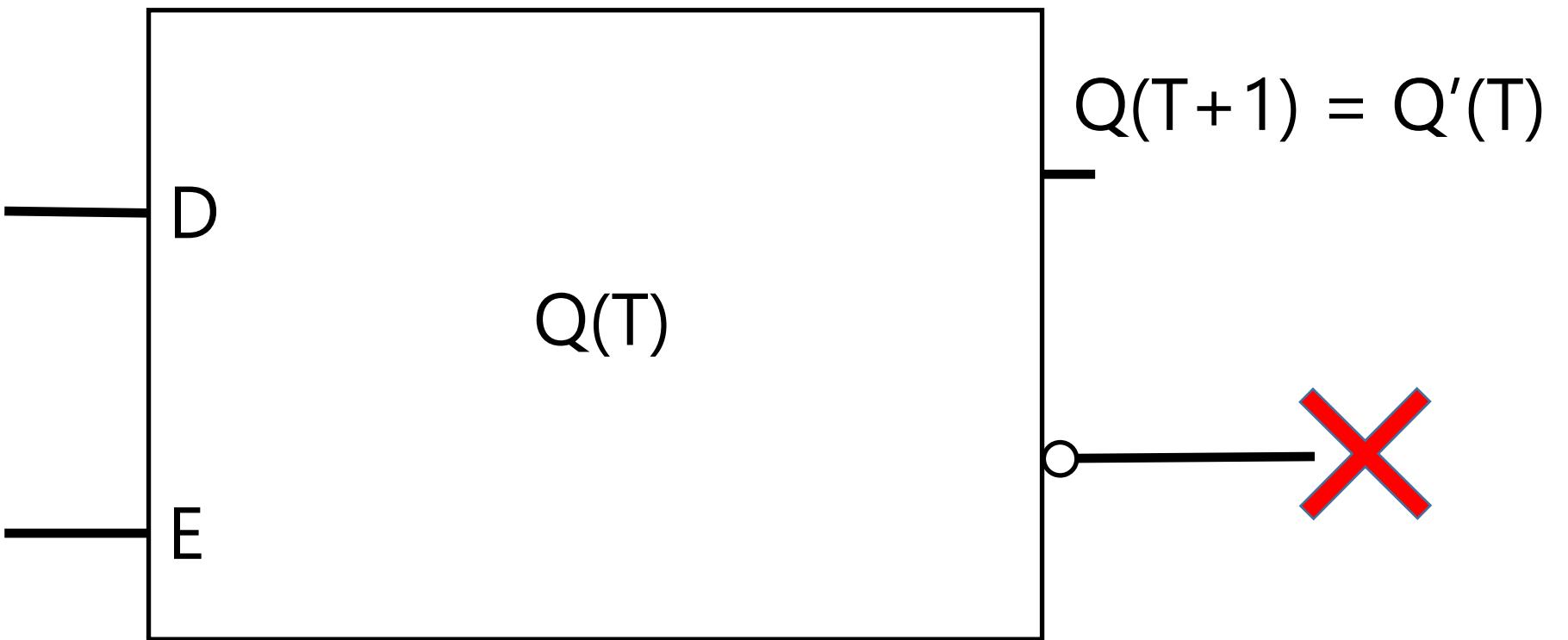


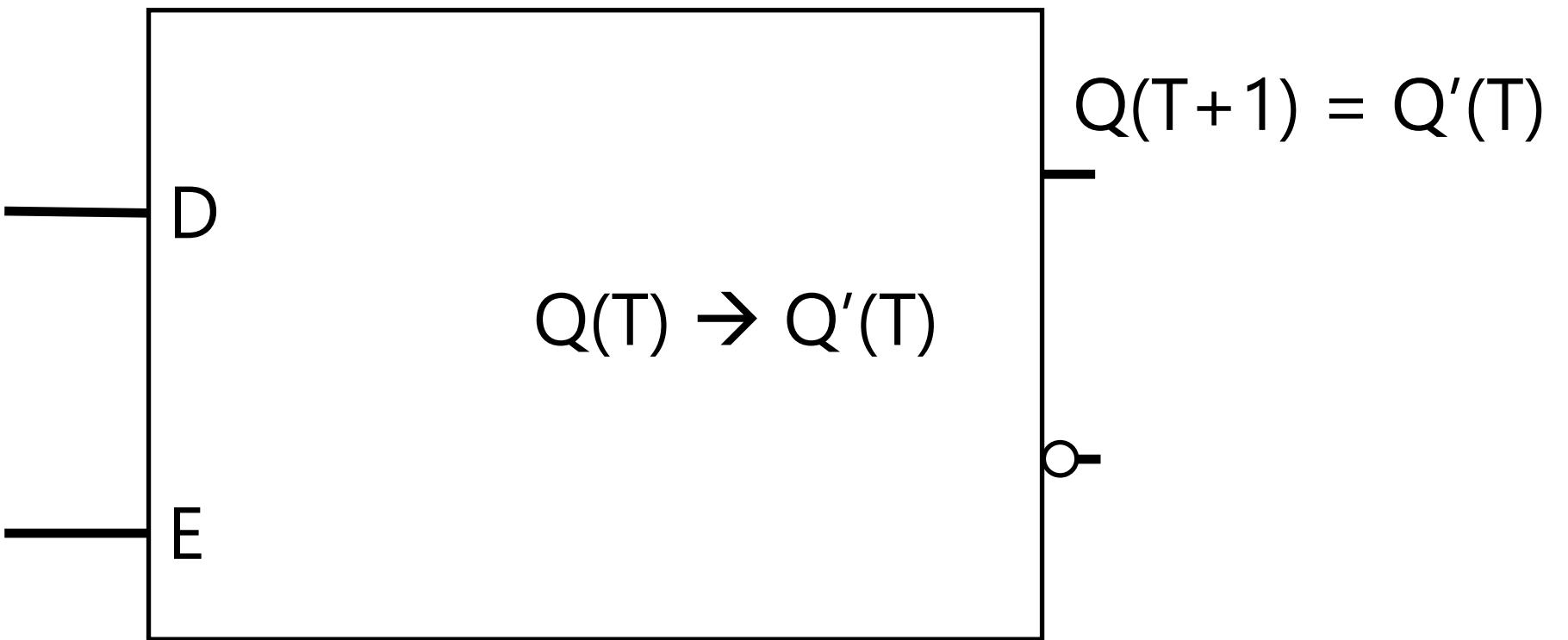
Latch

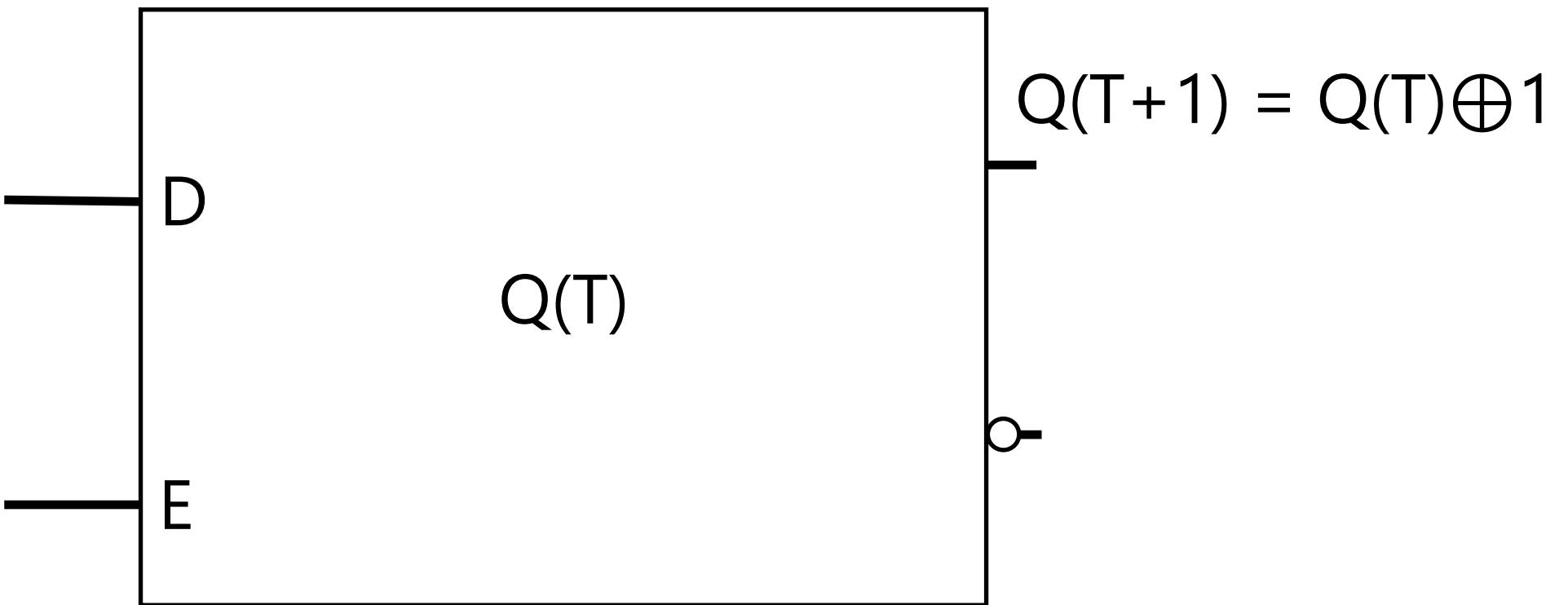


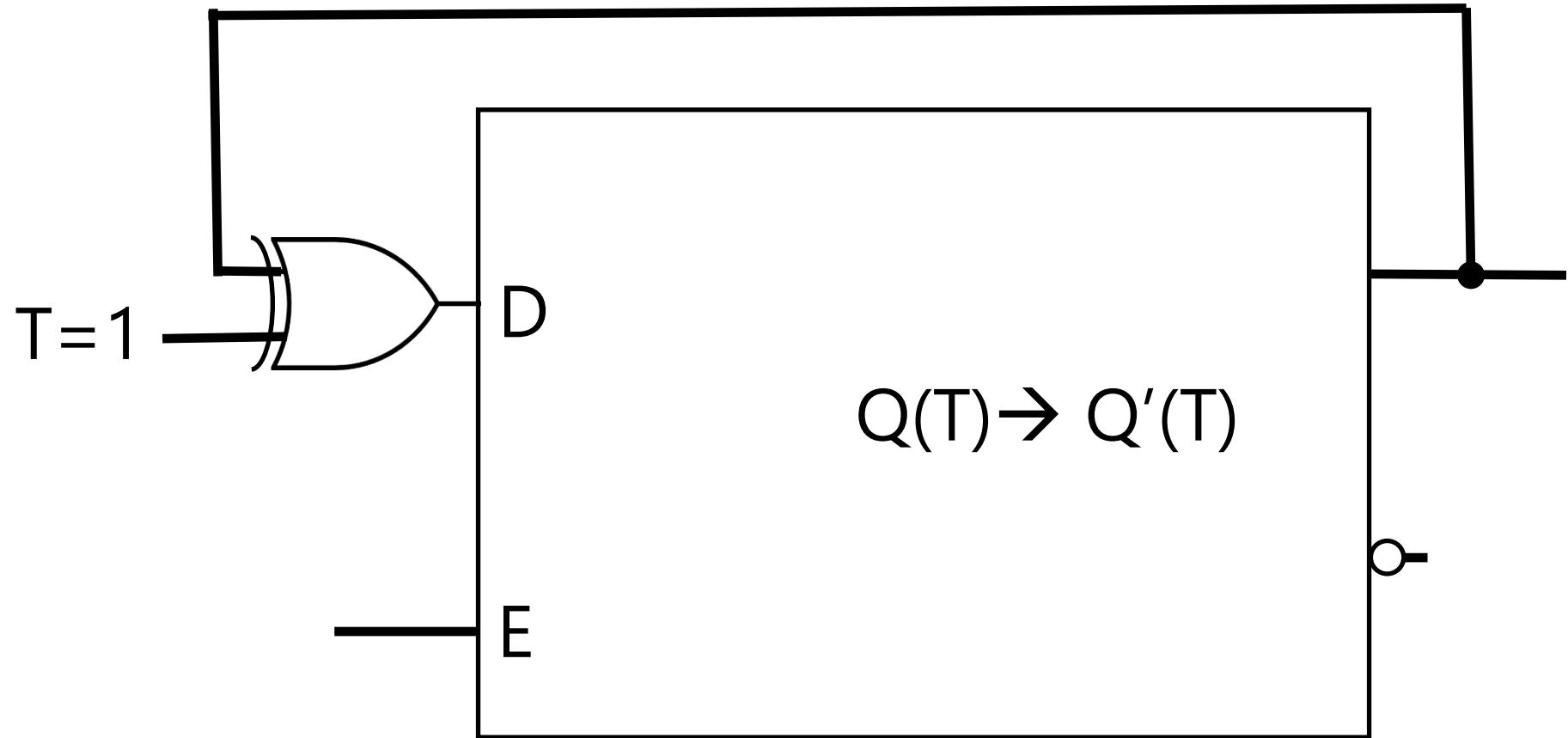
S	R	Q
0	0	Q_t
0	1	0
1	0	1
1	1	X

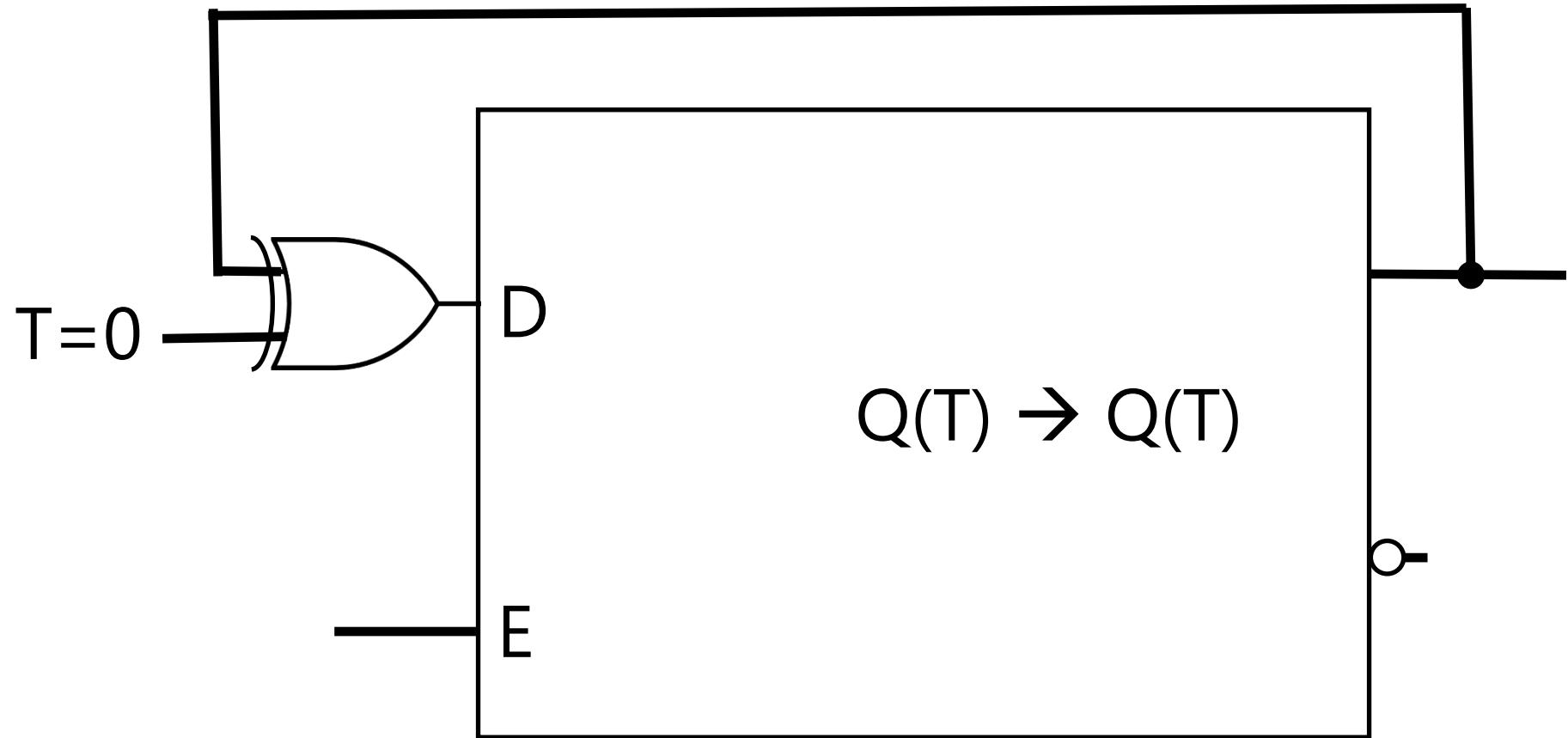
D	Q
0	0
1	1

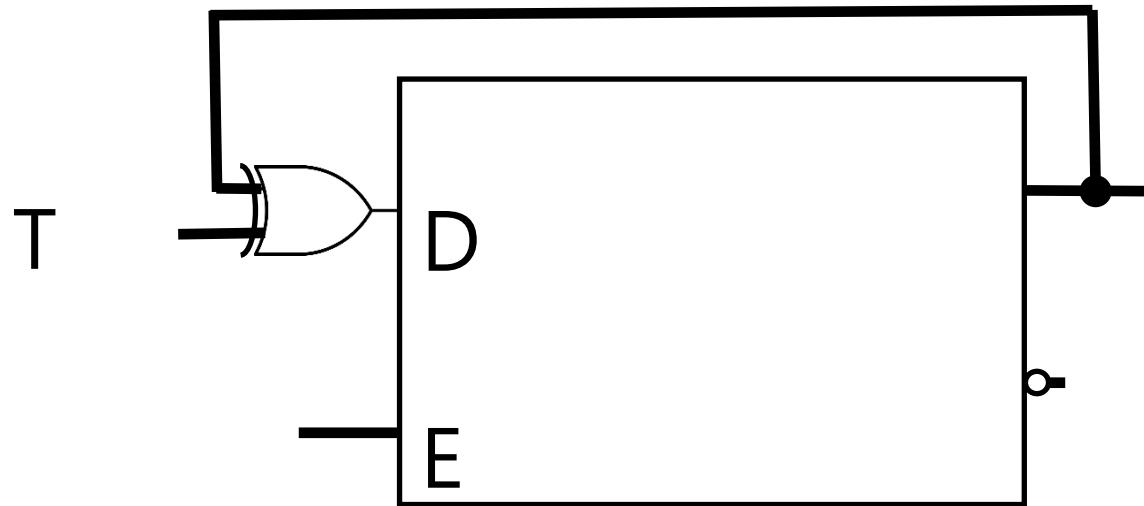




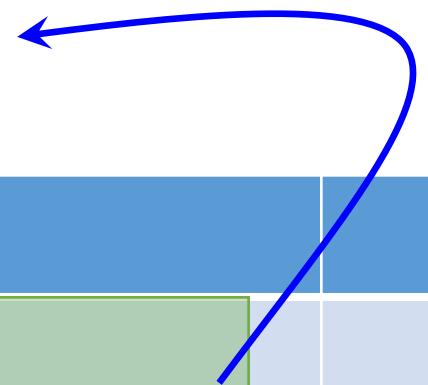




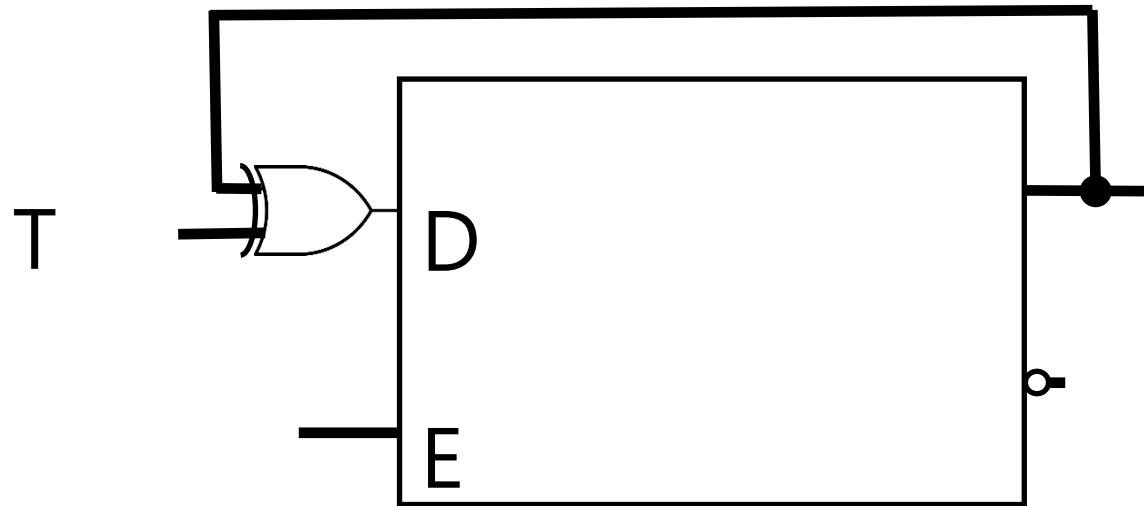




Store

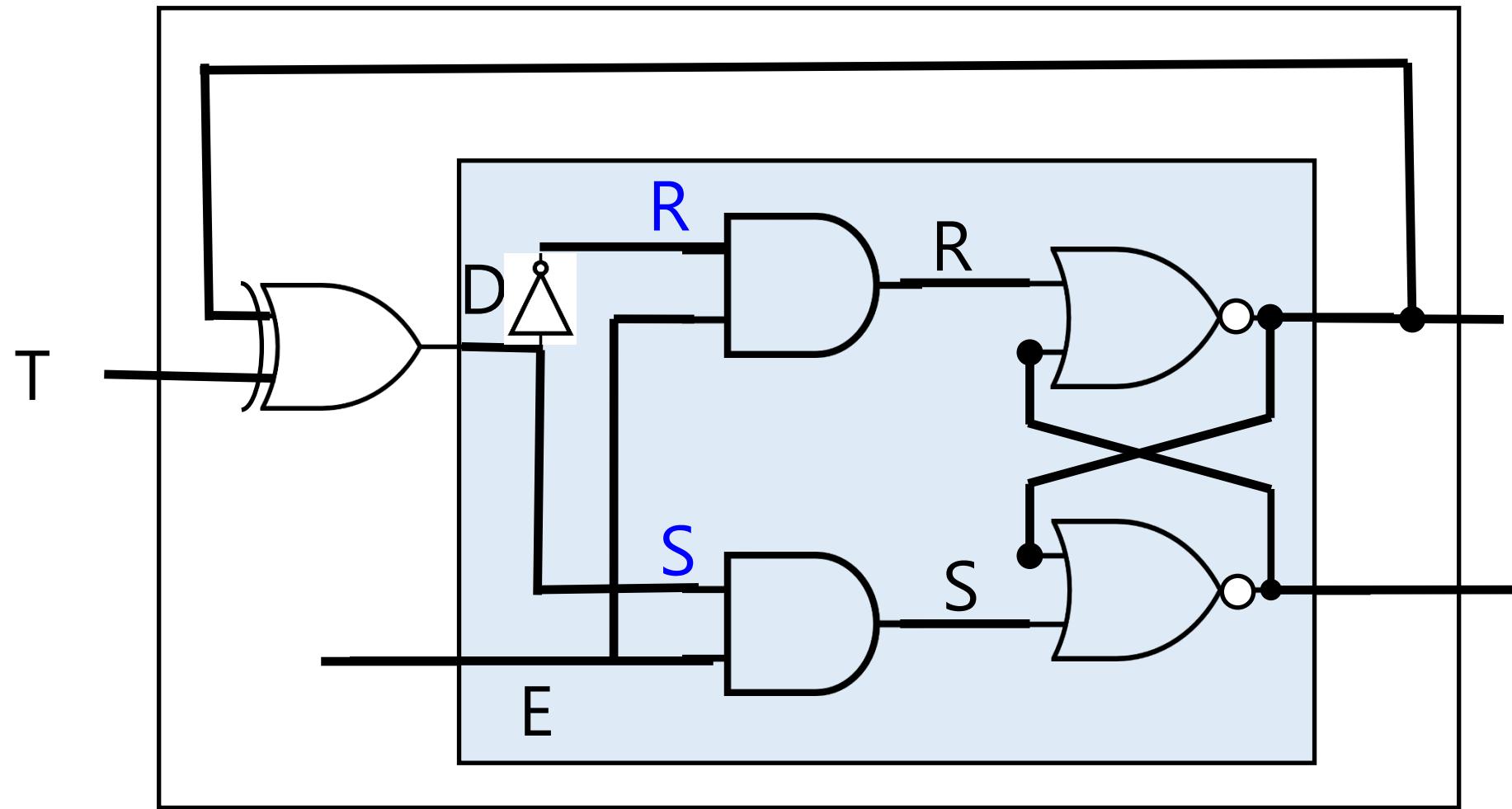


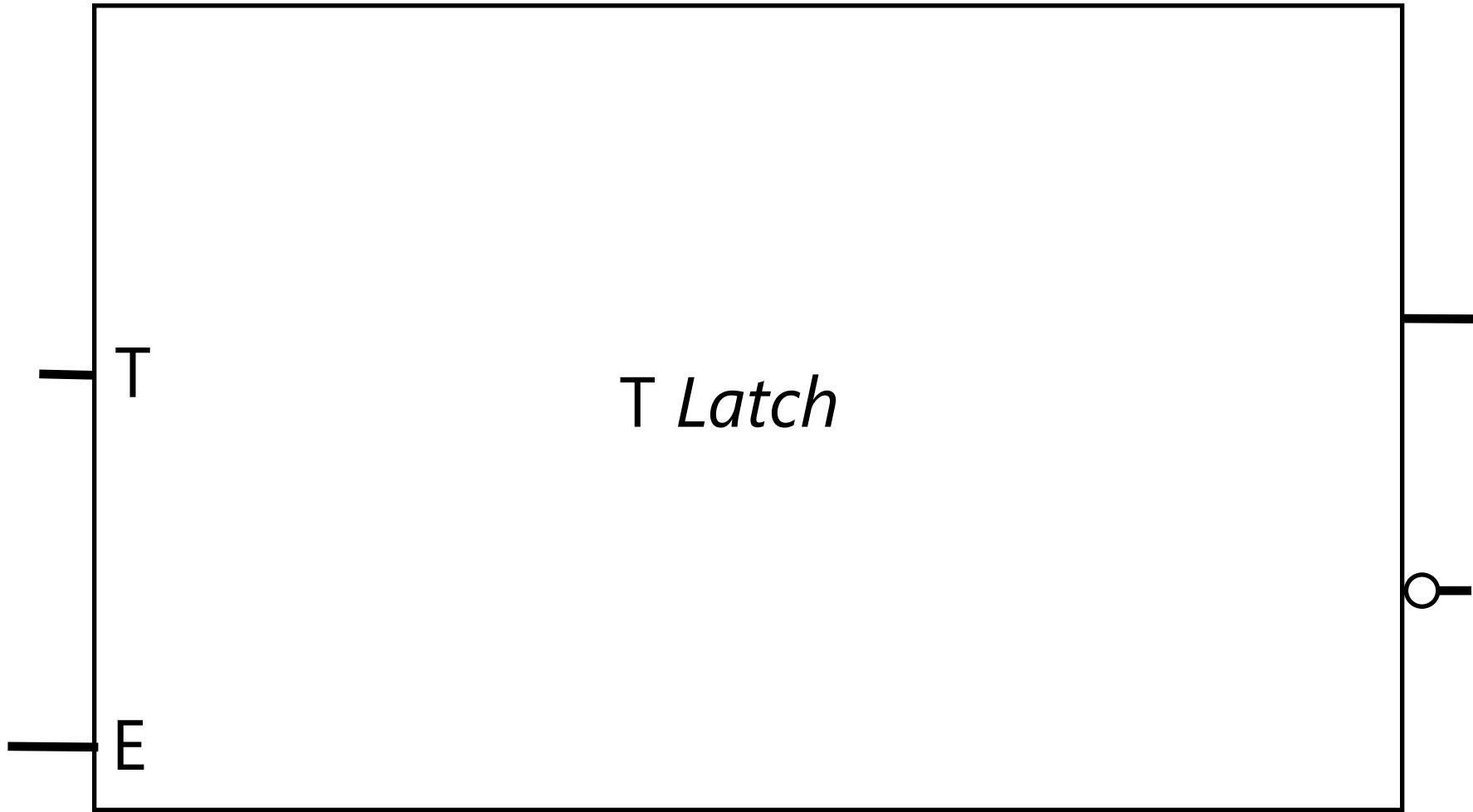
T	D	Q
0	$Q \oplus 0$	Q
1	$Q \oplus 1$	Q'

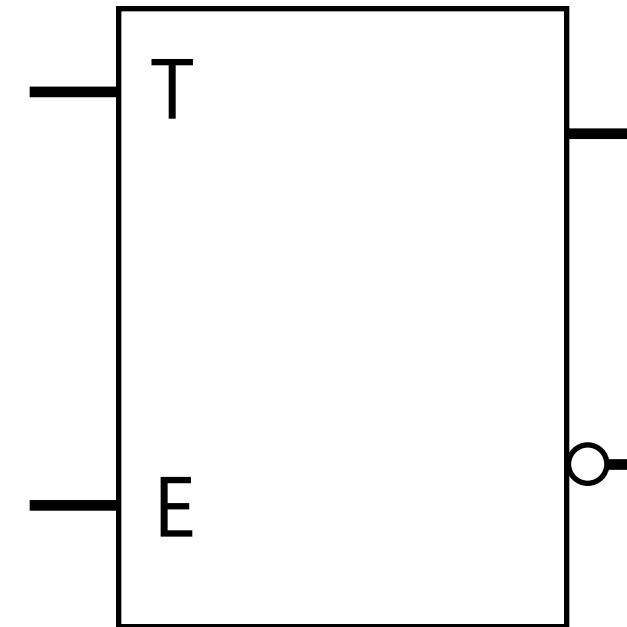
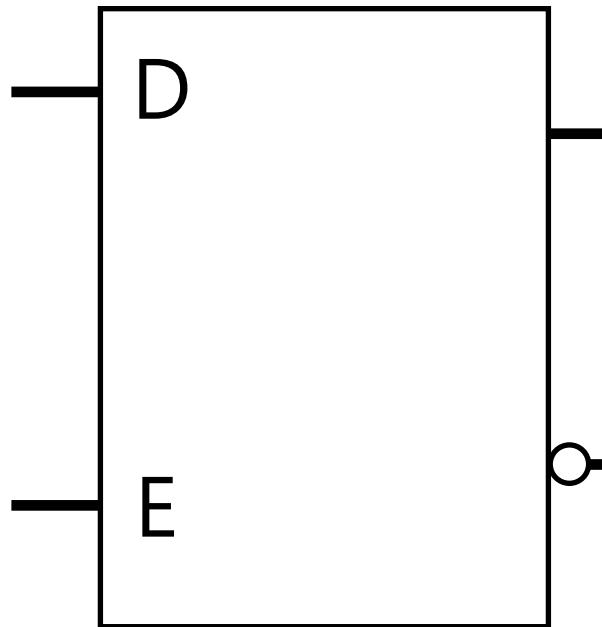
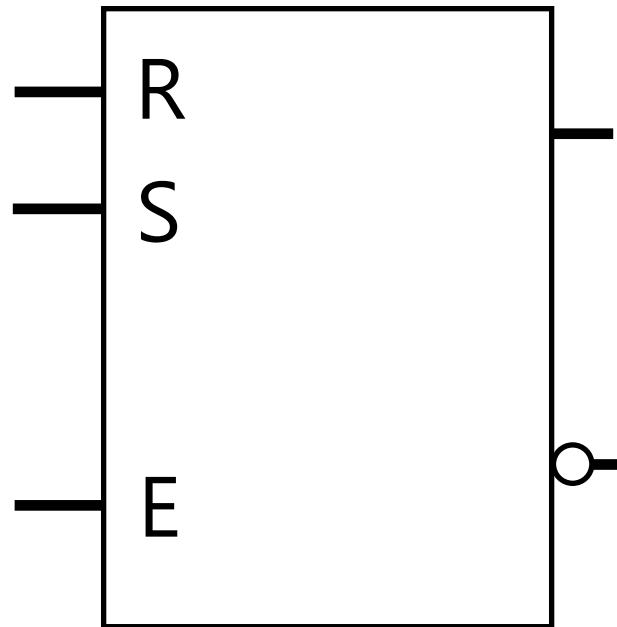


Complement

T	D	Q
0	$Q \oplus 0$	Q
1	$Q \oplus 1$	Q'



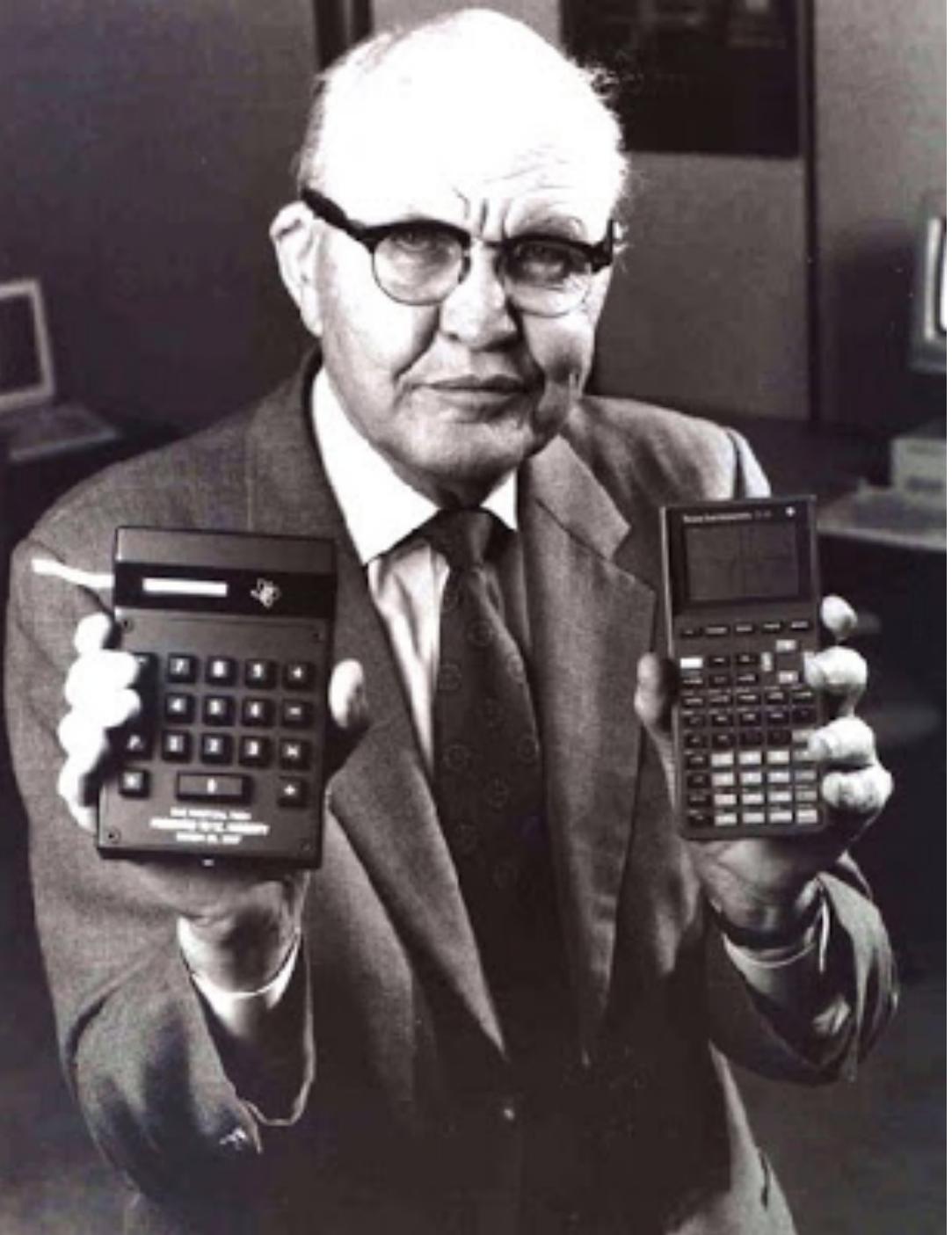




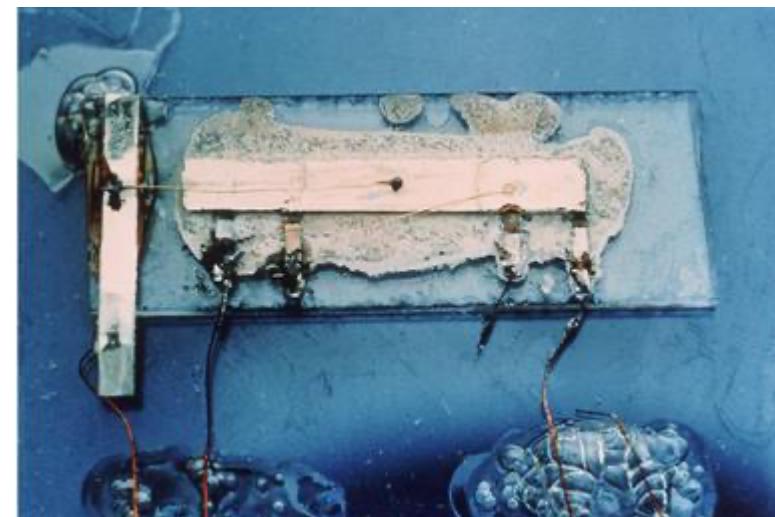
S	R	Q
0	0	Q_t
0	1	0
1	0	1
1	1	X

D	Q
0	0
1	1

T	Q
0	Q_t
1	Q'_t

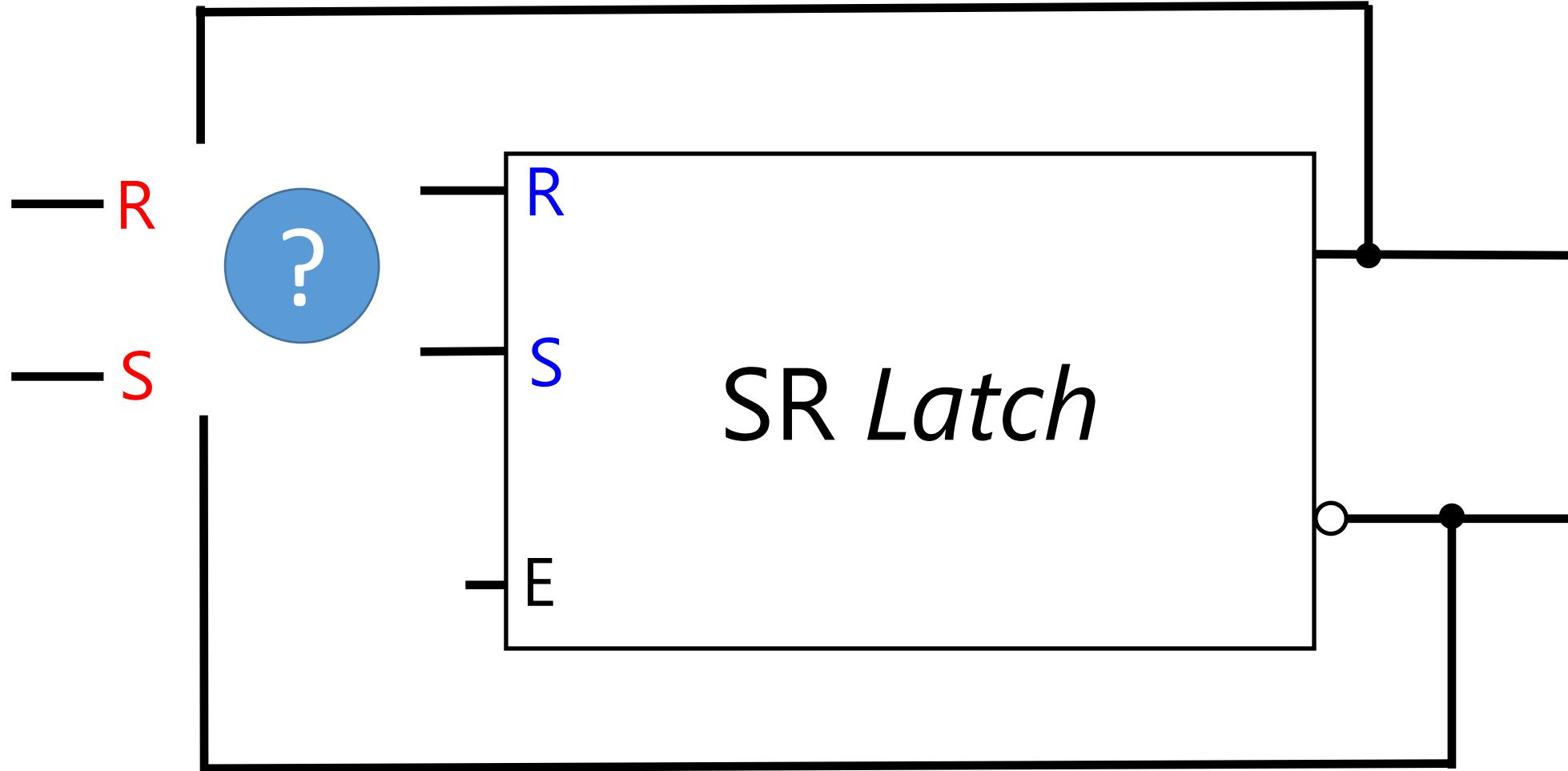


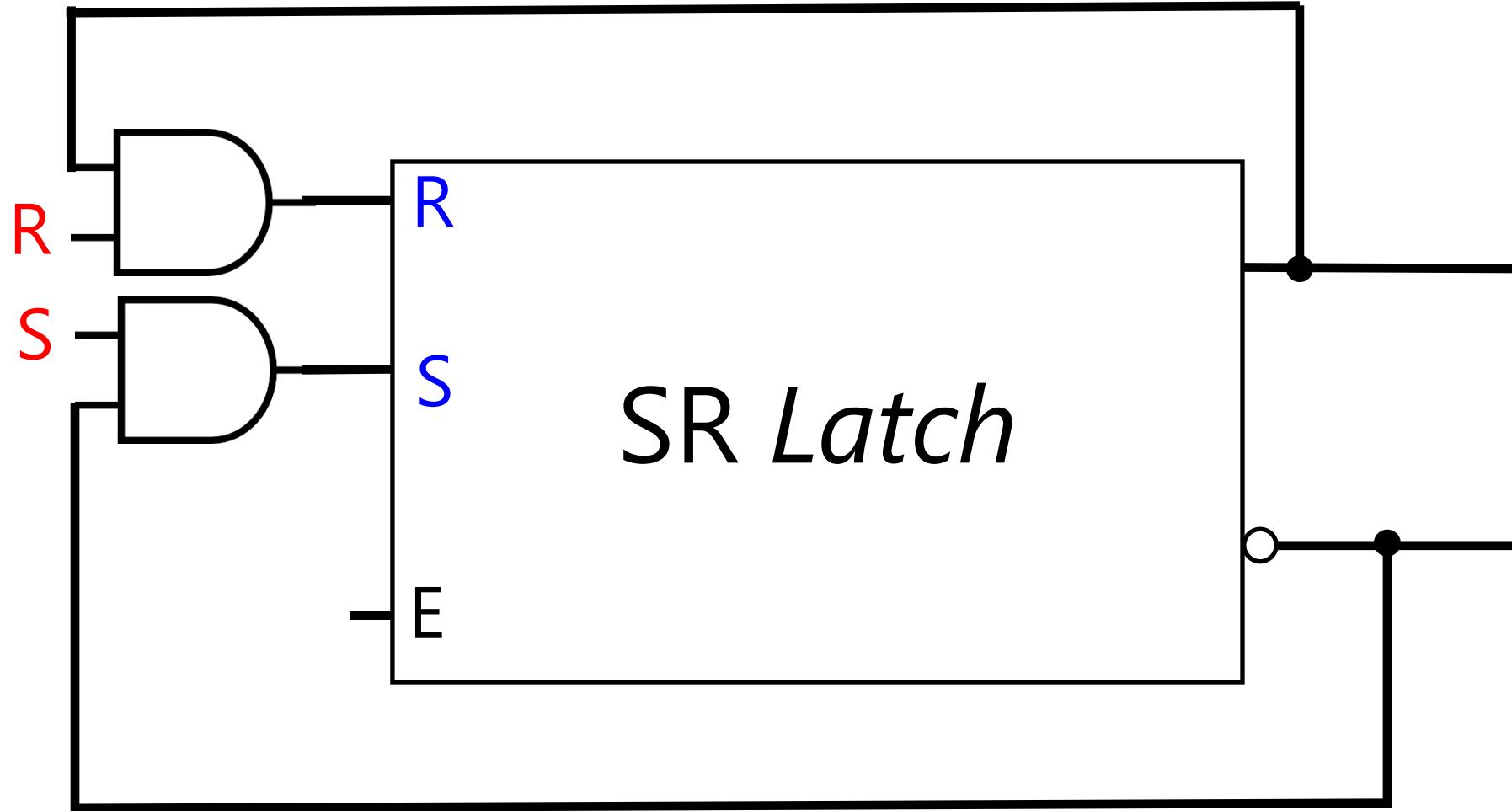
Jack St. Clair Kilby
(Nov. 8, 1923 – June 20, 2005)
Electrical Engineer
The 1st integrated circuit
1958
Nobel Prize in Physics, 2000

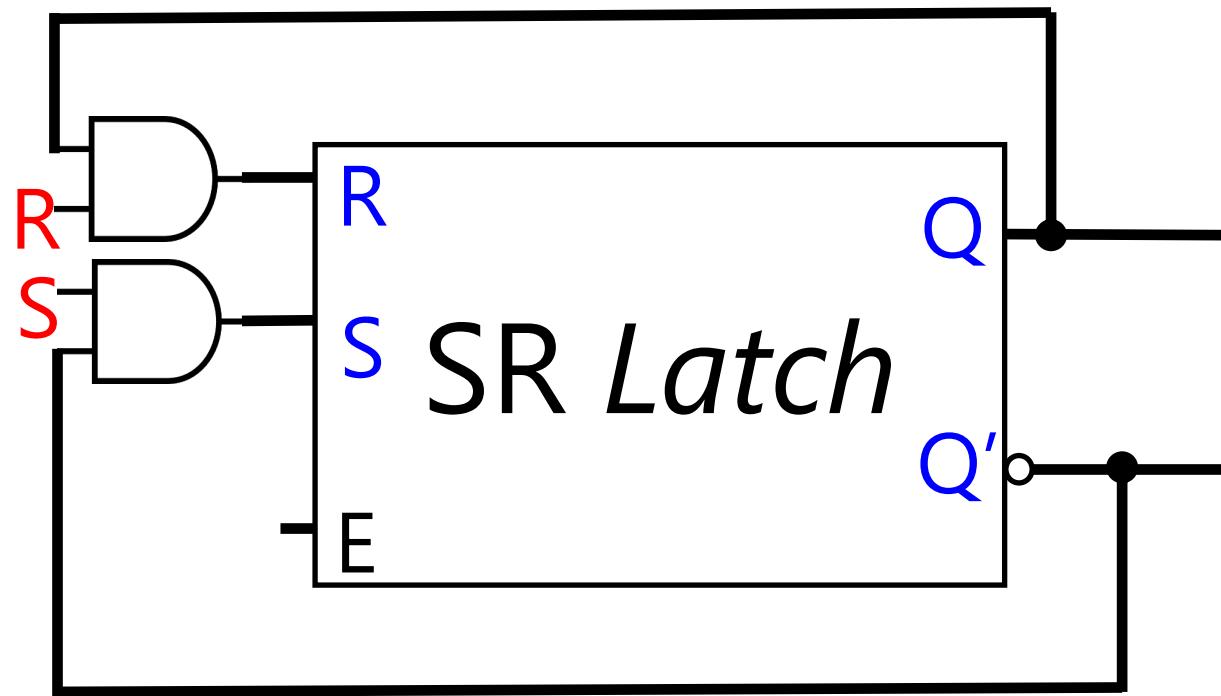




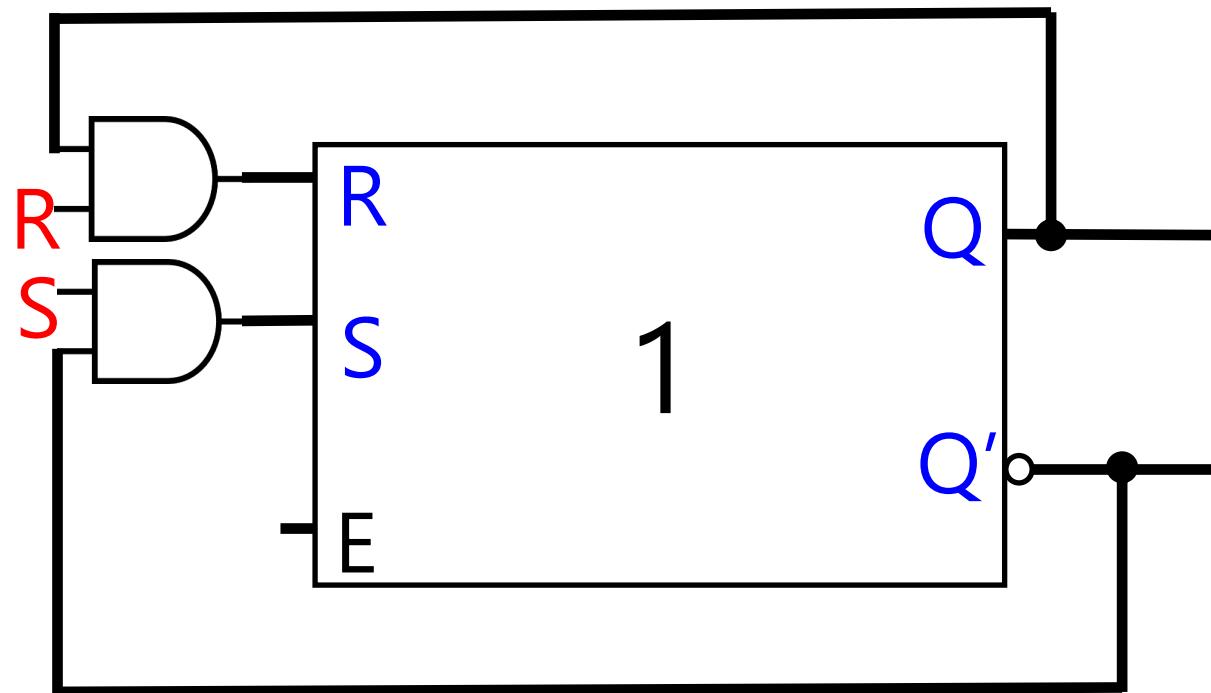
JK Latch



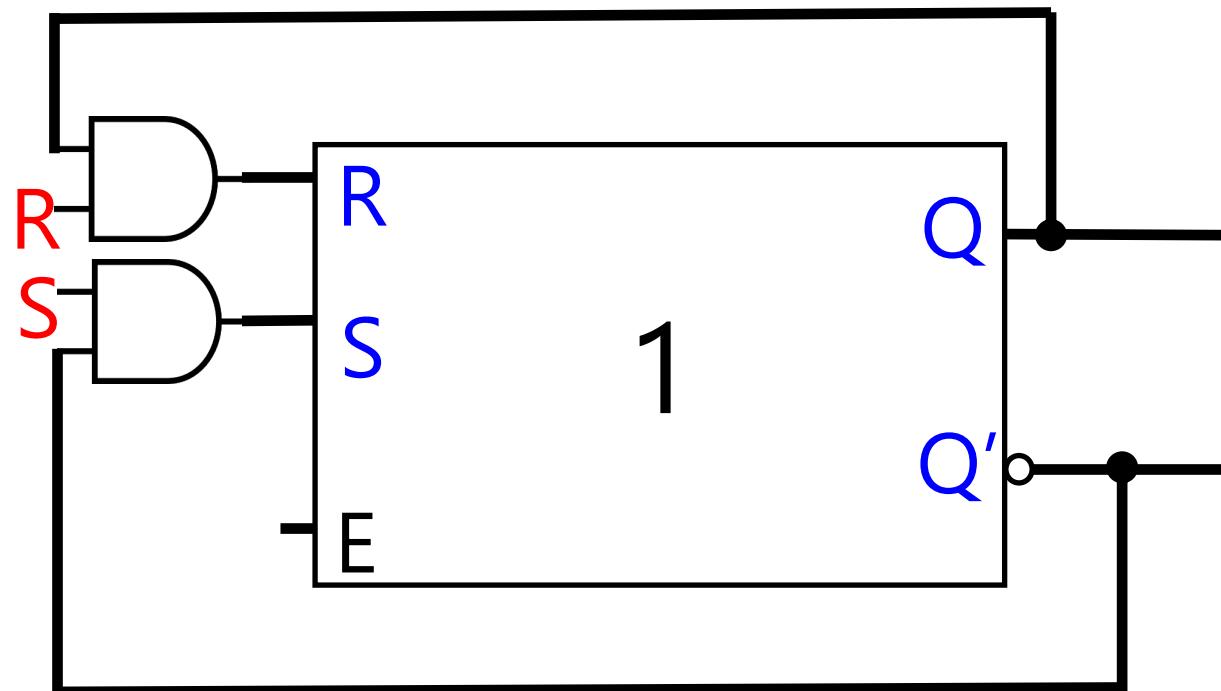




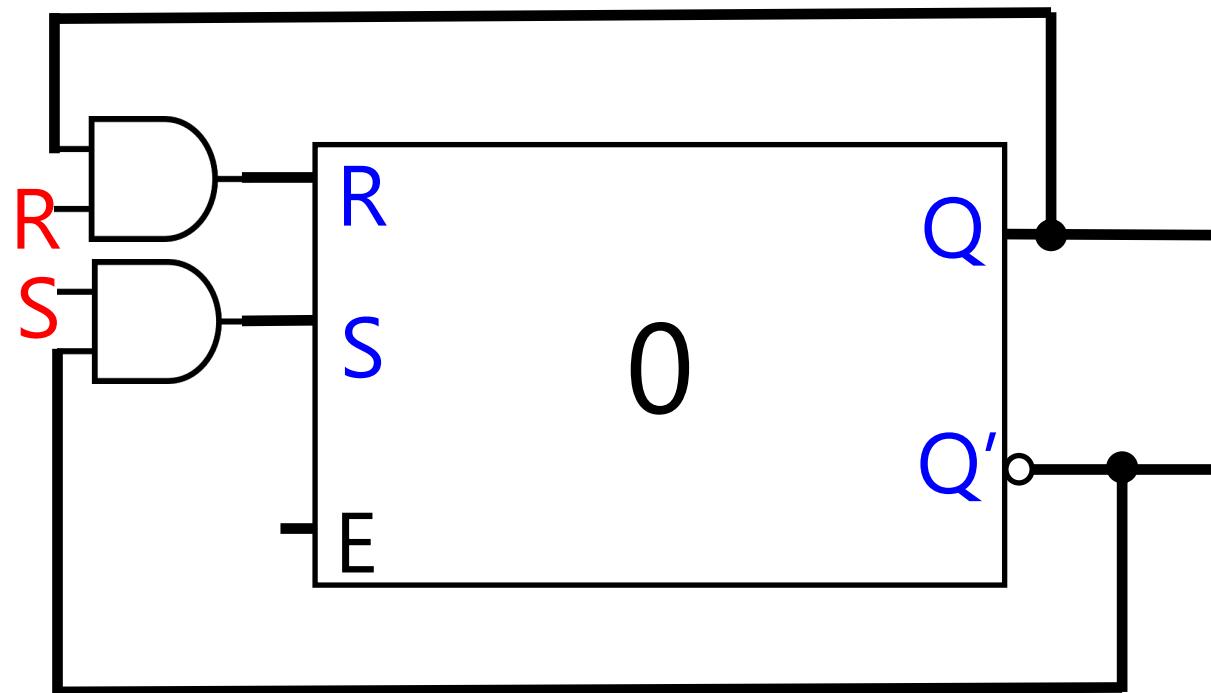
S	R	S	R
0	0	0	0
0	1	0	Q_t
1	0	Q'_t	0
1	1	Q'_t	Q_t



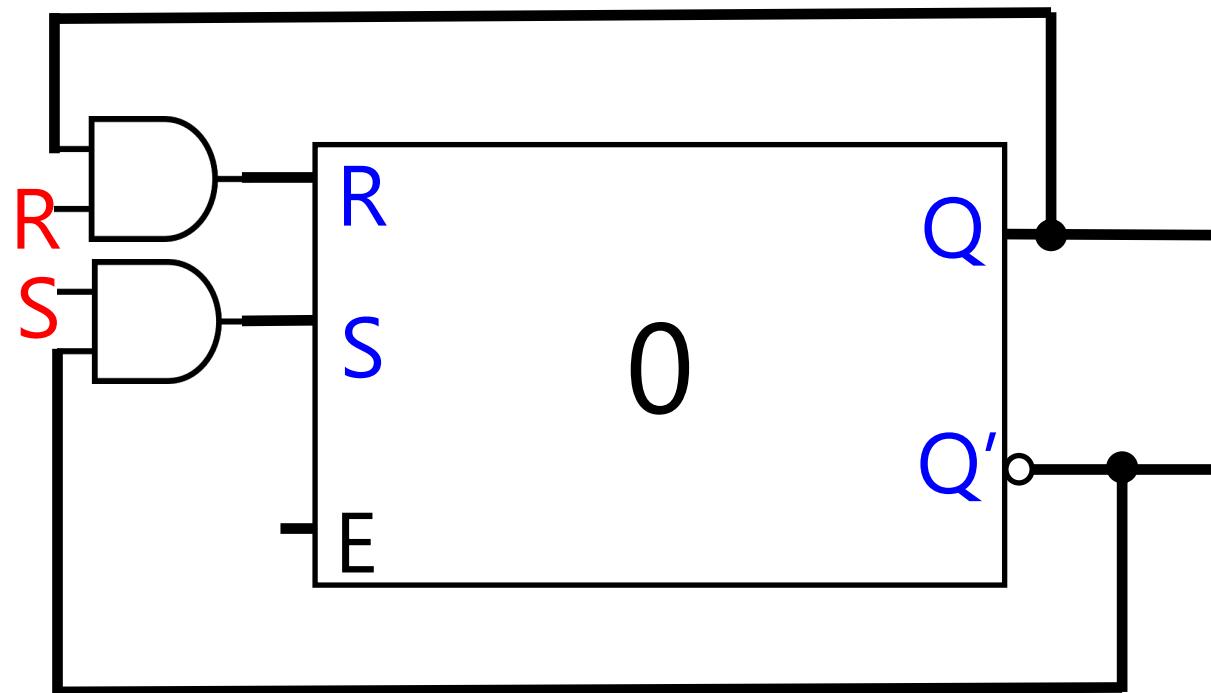
S	R	S	R
0	0	0	0
0	1	0	$Q_t = 1$
1	0	$Q'_t = 0$	0
1	1	$Q'_t = 0$	$Q_t = 1$



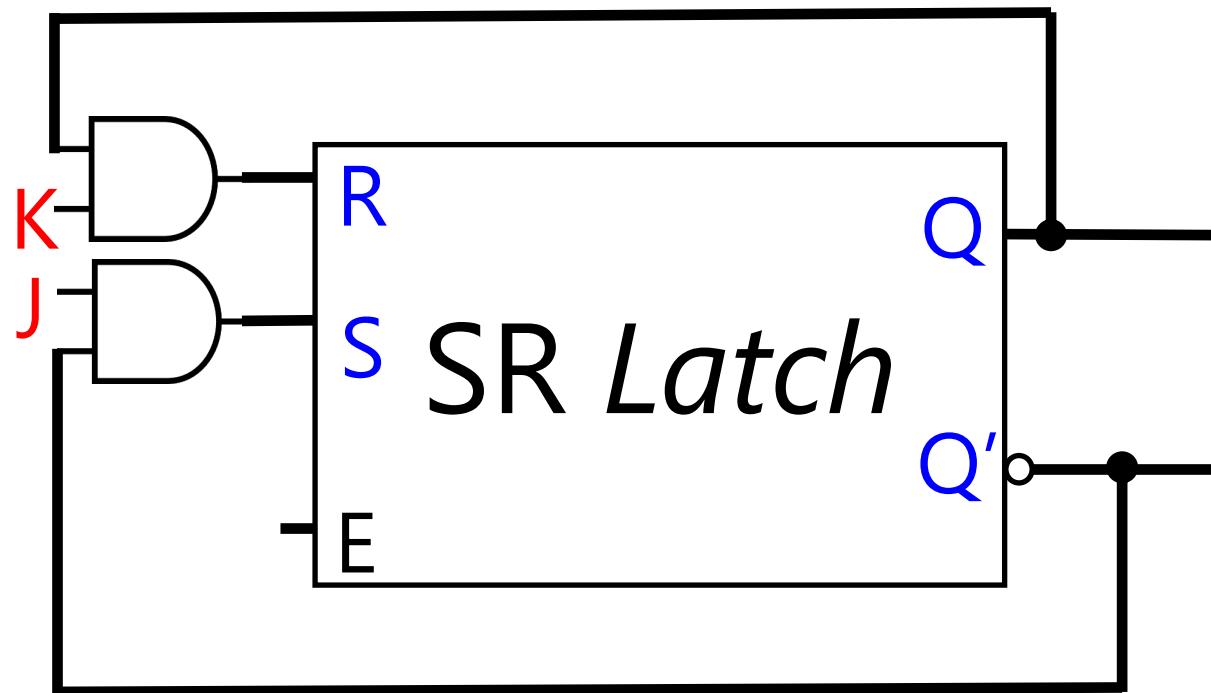
S	R	S	R	Q
0	0	0	0	Store= $Q_t=1$
0	1	0	$Q_t=1$	Reset=0
1	0	$Q'_t=0$	0	Store= $Q_t=1$
1	1	$Q'_t=0$	$Q_t=1$	Reset=0



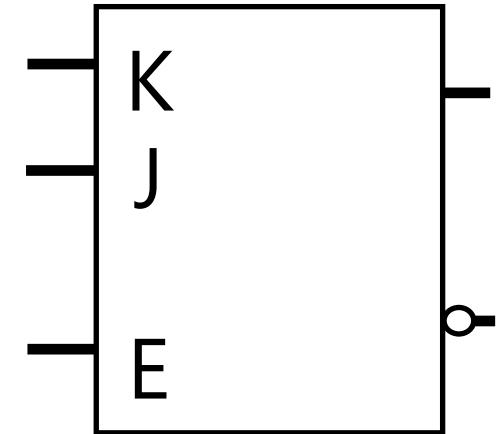
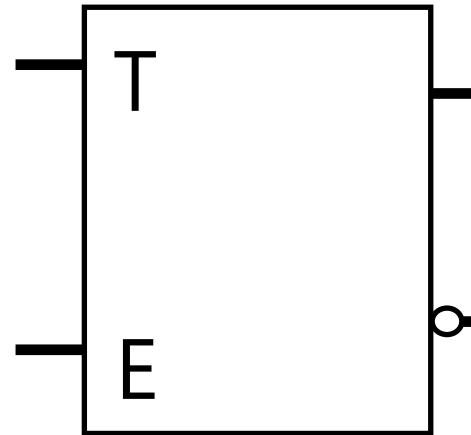
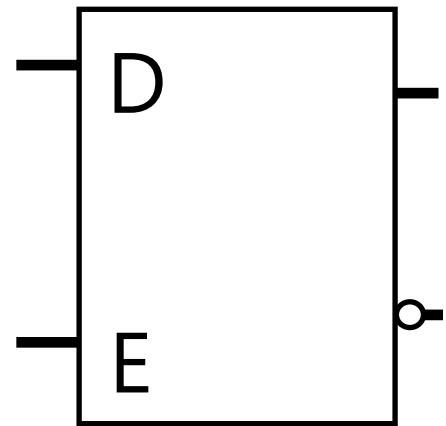
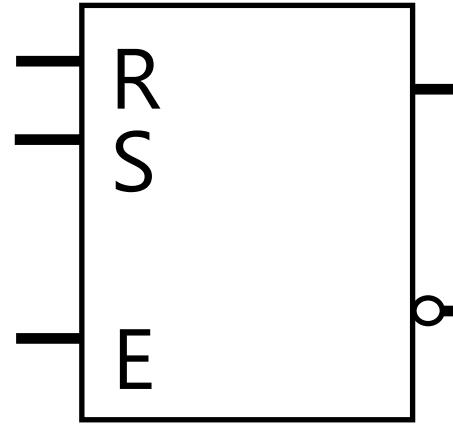
S	R	S	R
0	0	0	0
0	1	0	$Q_t = 0$
1	0	$Q'_t = 1$	0
1	1	$Q'_t = 1$	$Q_t = 0$



S	R	S	R	Q
0	0	0	0	Store = $Q_t = 0$
0	1	0	$Q_t = 0$	Store = $Q_t = 0$
1	0	$Q'_t = 1$	0	Set = 1
1	1	$Q'_t = 1$	$Q_t = 0$	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



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

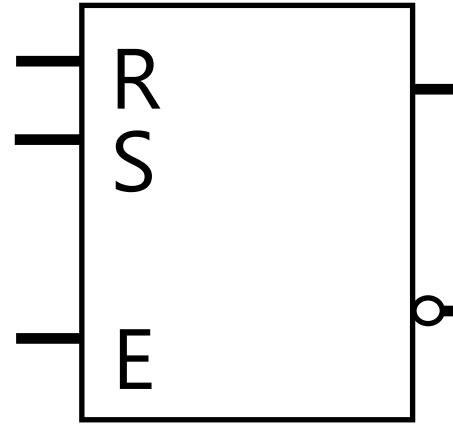
D	Q
0	0
1	1

T	Q
0	Q_t
1	Q'_t

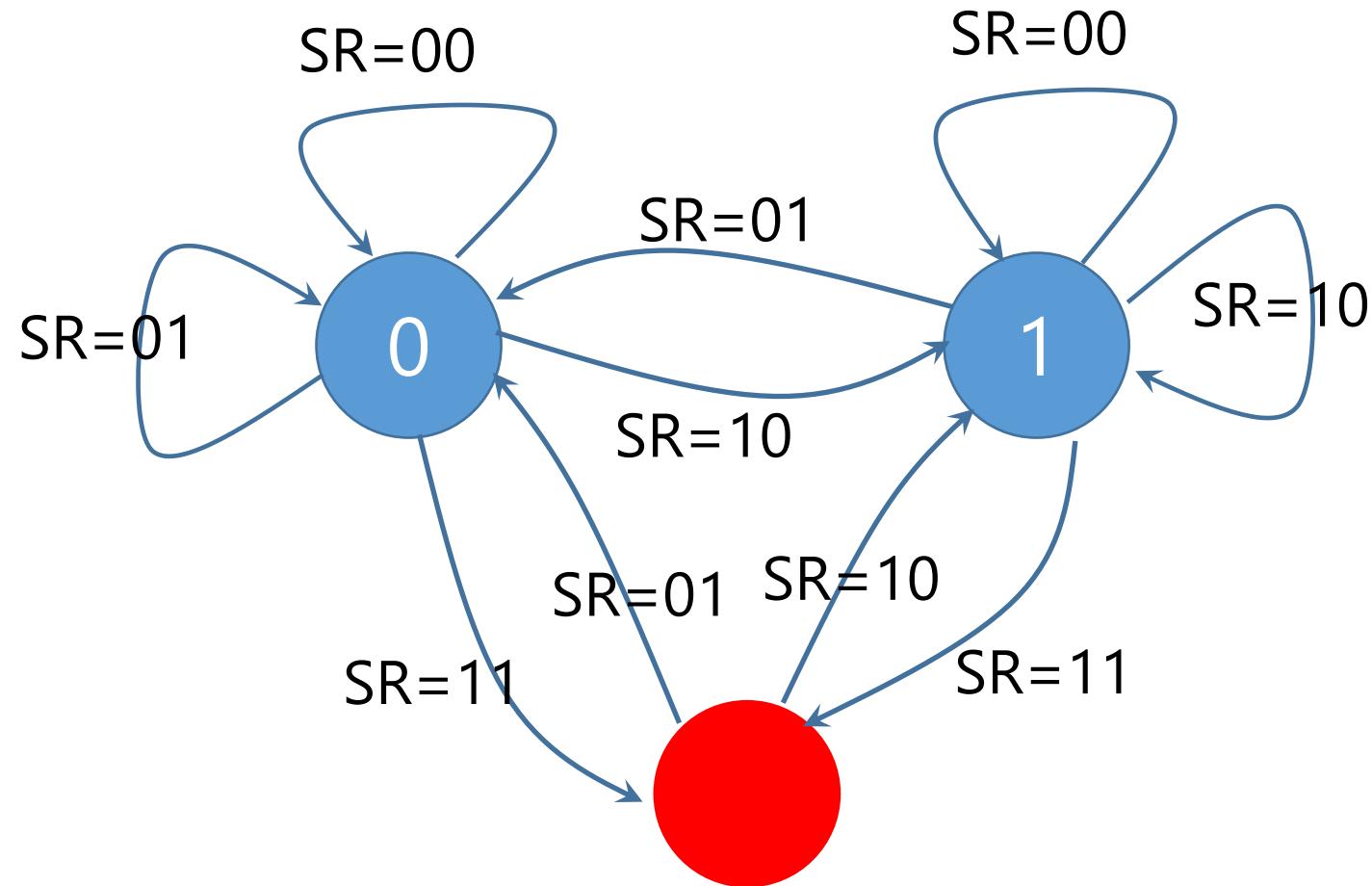
J	K	Q
0	0	Q_t
0	1	0
1	0	1
1	1	Q'_t

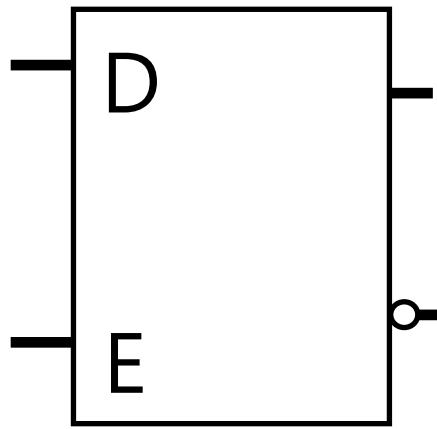
State Transition Diagram *Life Cycle*

Theory of Automata
COMP-2140: Computer Languages, Grammars, and Translators

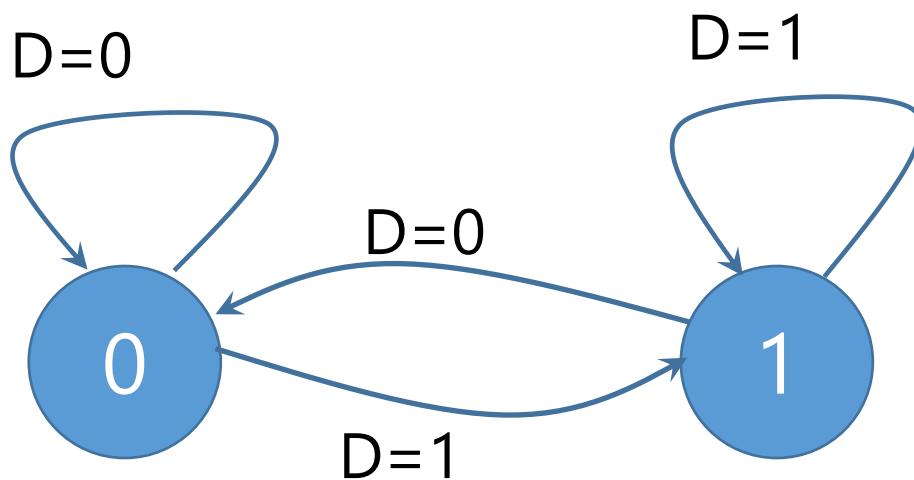


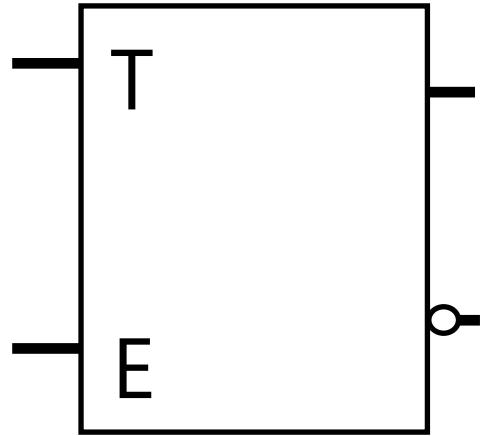
S	R	Q
0	0	Q_t
0	1	0
1	0	1
1	1	\times



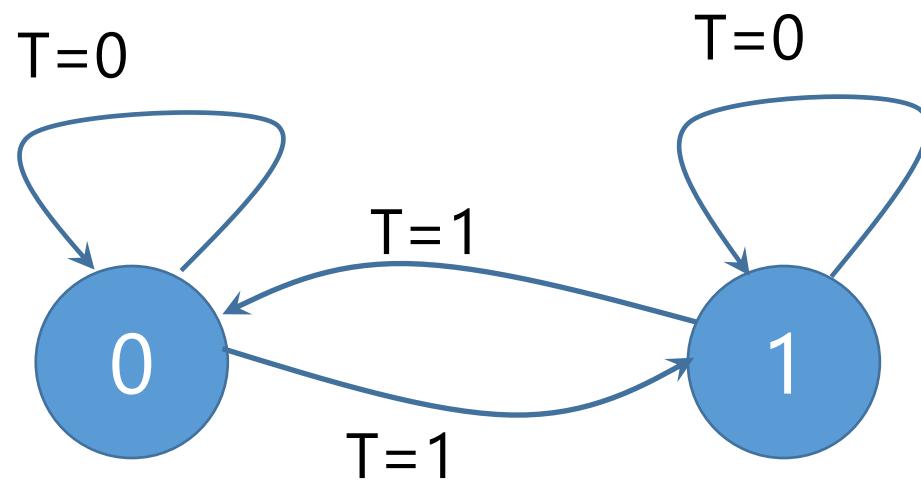


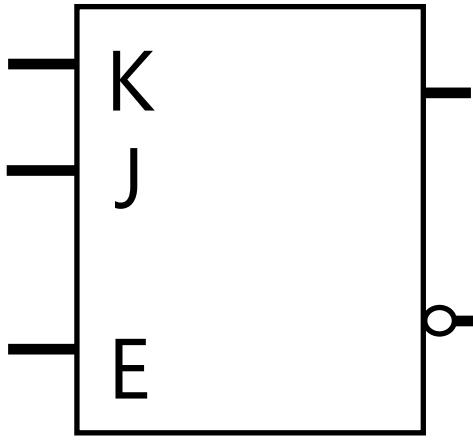
D	Q
0	0
1	1



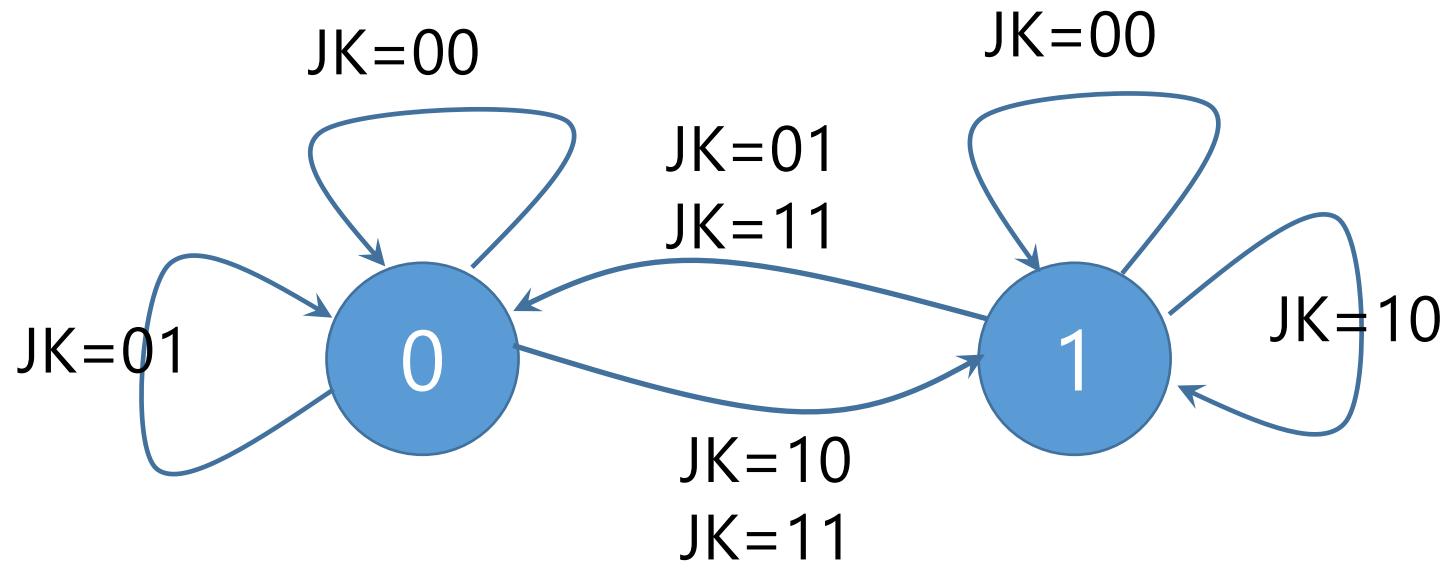


T	Q
0	Q_t
1	Q'_t





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



Flip-Flop
