

Flip-flops

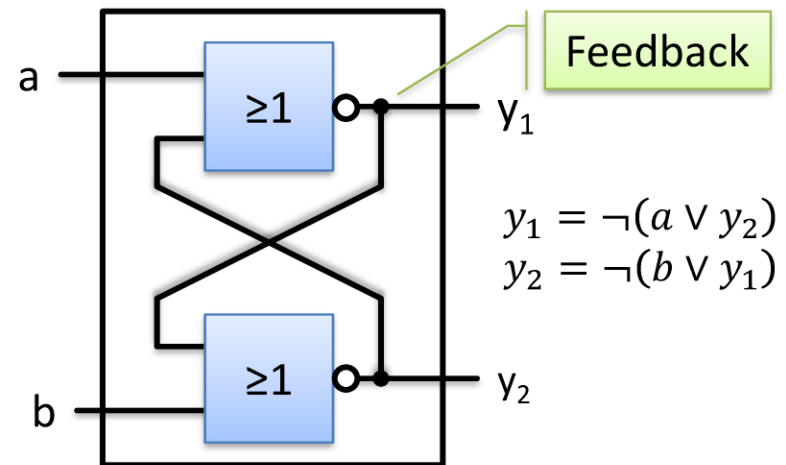
Digital Electronics

Wolfgang Neff

Latches (1)

- SR Latch

- Stores a bit
- Two input lines (a, b)
- Two output lines (y_1, y_2)
- Implementation
 - Two NOR gates
 - Four input lines
 - Feedback
 - Two inputs come from feedbacks



Latches (2)

- SR Latch (continued)
 - Truth table

a	b	y_1	y_2	y_1^+	y_2^+
0	0	0	0	1	1
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	0	0

continues on the right side


		Current state		Next state	
a	b	y_1	y_2	y_1^+	y_2^+
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

Latches (3)

- SR Latch (continued)
 - Truth table


a	b	y_1	y_2	y_1^+	y_2^+
0	0	0	0	1	1
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	0	0

instable



continues on the right side

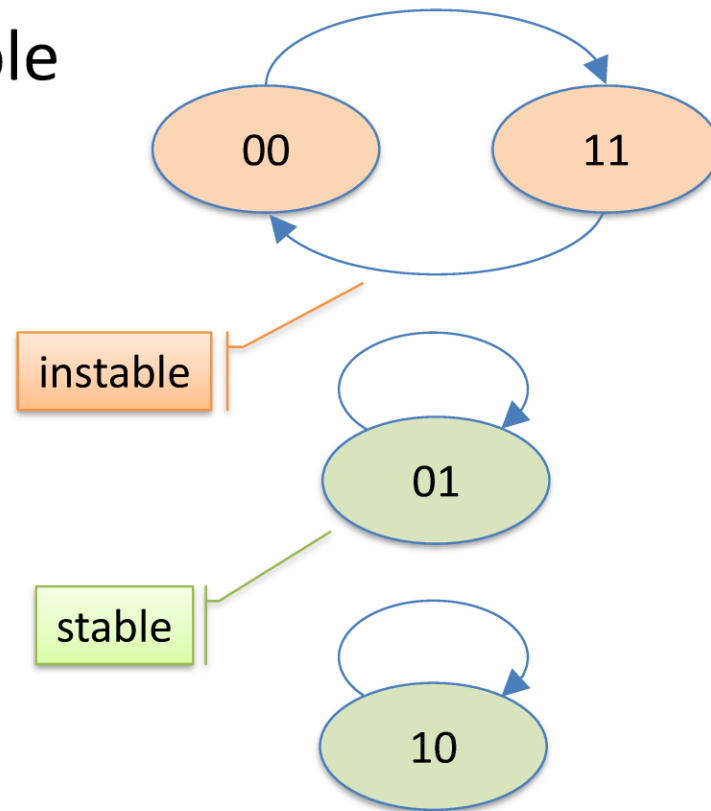
a	b	y_1	y_2	y_1^+	y_2^+
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0



Latches (4)

- SR Latch (continued)
 - First block of truth table

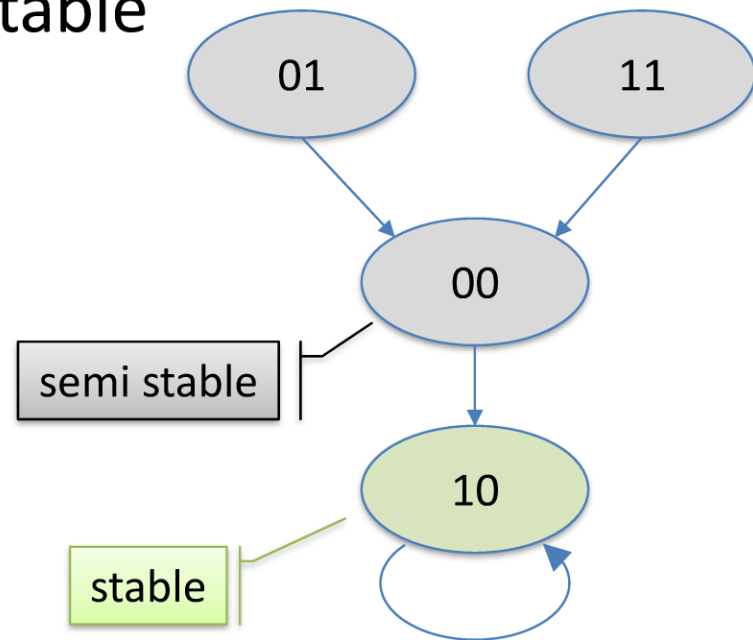
a	b	y_1	y_2	y_1^+	y_2^+
0	0	0	0	1	1
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0



Latches (5)

- SR Latch (continued)
 - Second block of truth table

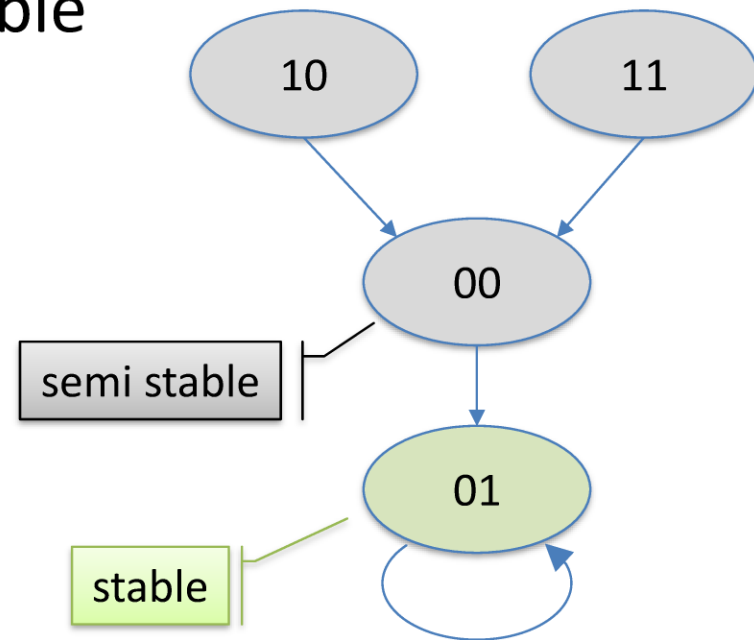
a	b	y_1	y_2	y_1^+	y_2^+
0	1	0	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	0	0



Latches (6)

- SR Latch (continued)
 - Third block of truth table

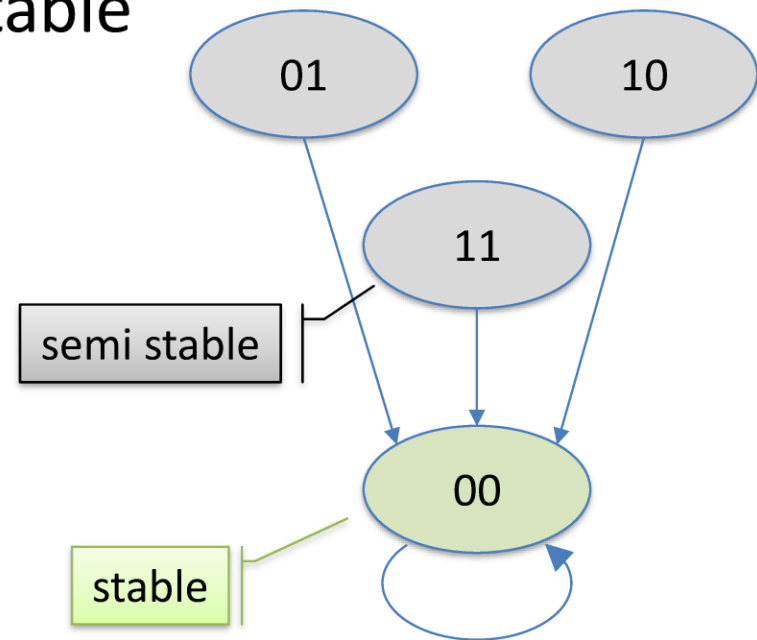
a	b	y_1	y_2	y_1^+	y_2^+
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	0
1	0	1	1	0	0



Latches (7)

- SR Latch (continued)
 - Fourth block of truth table

a	b	y_1	y_2	y_1^+	y_2^+
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0



Latches (8)

- SR Latch (continued)

- Analysis of block $a = 0$ and $b = 0$

a	b	y_1	y_2	y_1^+	y_2^+	Criterion
0	0	0	0	1	1	$y_1 = y_2$
0	0	0	1	0	1	$y_1 \neq y_2$
0	0	1	0	1	0	$y_1 \neq y_2$
0	0	1	1	0	0	$y_1 = y_2$

- There are stable and instable states
- The states are instable if $y_1 = y_2$
- The states are stable if $y_1 \neq y_2$ or $y_1 = \neg y_2$
- We must prohibit $y_1 = y_2$

Latches (9)

- SR Latch (continued)
 - Analysis of block $a = 1$ and $b = 1$

a	b	y_1	y_2	y_1^+	y_2^+
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

- All states end up in a stable state
- At the stable state there is $y_1 = y_2$
- $y_1 = y_2$ is prohibited
- We must prohibit $a = 1$ and $b = 1$

Latches (10)

- SR Latch (continued)

- Analysis of block $a = 0$ and $b = 1$

a	b	y_1	y_2	y_1^+	y_2^+	Action
0	1	0	0	1	0	
0	1	0	1	0	0	
0	1	1	0	1	0	Set
0	1	1	1	0	0	

- All states end up in a stable state
- At the stable state there is $y_1 = \neg y_2$
- The stable state is okay
- At the stable state there is $y_1^+ = 1$ for sure

Latches (11)

- SR Latch (continued)

- Analysis of block $a = 1$ and $b = 0$

a	b	y_1	y_2	y_1^+	y_2^+	Action
1	0	0	0	0	1	
1	0	0	1	0	1	Reset
1	0	1	0	0	0	
1	0	1	1	0	0	

- All states end up in a stable state
- At the stable state there is $y_1 = \neg y_2$
- The stable state is okay
- At the stable state there is $y_1^+ = 0$ for sure

Latches (12)

- SR Latch (continued)
 - Analysis of block $a = 0$ and $b = 0$ (continued)

a	b	y_1	y_2	y_1^+	y_2^+	Action
0	0	0	0	1	1	
0	0	0	1	0	1	Store
0	0	1	0	1	0	Store
0	0	1	1	0	0	

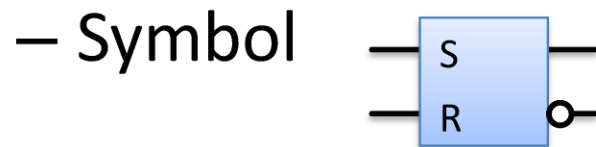
- The instable states are prohibited
- There are two stable states
- At the stable states there is $y_1^+ = y_1$ for sure

Latches (13)

- SR Latch (continued)
 - Analysis
 - Renaming the lines
 - $a = 0, b = 0$
stable only if $y_1 \neq y_2$ or $y_1 = \neg y_2$ (new names: $y_1 = Q, y_2 = \neg Q$)
if $y_1 \neq y_2$ then $Q^+ = Q$ (Q gets stored)
 - $a = 0, b = 1$
always stable and $Q^+ = 1$ (Q gets set, new name: $b = S$)
 - $a = 1, S = 0$
always stable and $Q^+ = 0$ (Q gets reset, new name: $a = R$)
 - $R = 1, S = 1$
invalid because $y_1 = y_2$ and circuit might become instable

Latches (14)

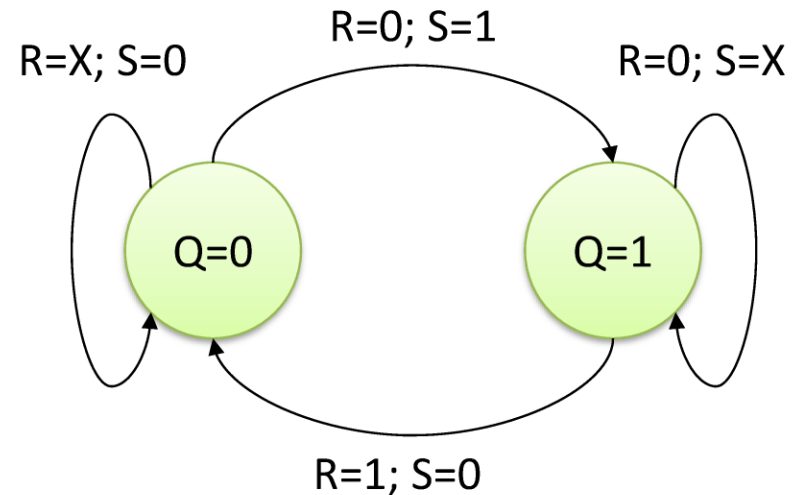
- SR Latch (continued)



– State table

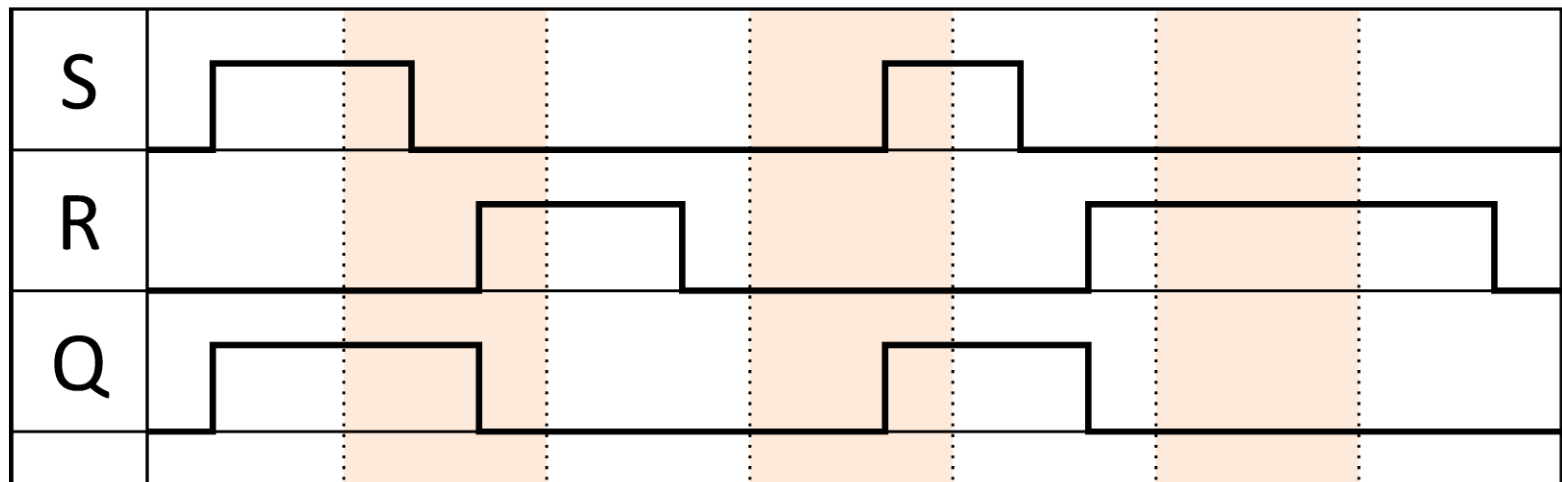
S	R	Q ⁺	Action
0	0	Q	Store
0	1	0	Reset
1	0	1	Set
1	1	X	Invalid

– State diagram



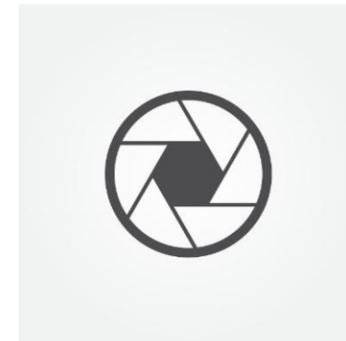
Latches (15)

- SR Latch (finished)
 - Time diagram



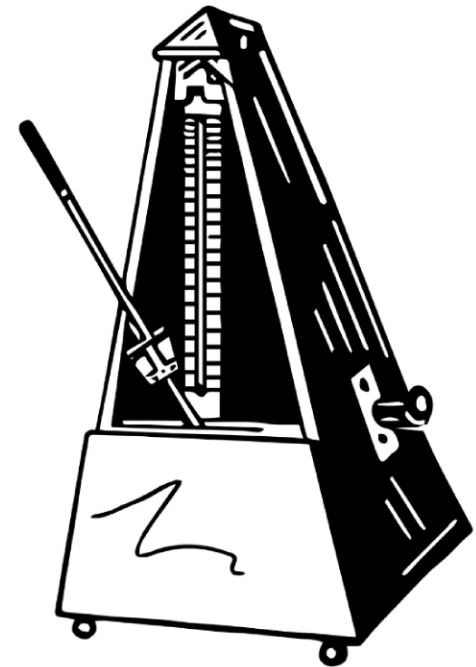
Triggers (1)

- Active and Inactive Inputs
 - Inputs are not always active
 - Active state
 - Input is considered
 - Output depends on input
 - Inactive state
 - Input is ignored
 - Input has no effect on output
 - Inputs can be triggered
 - Captured at certain times



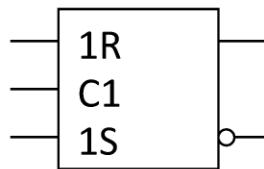
Triggers (2)

- Types of Triggers
 - Level triggers
 - State depends on level
 - 0: inactive, 1: active
 - Or vice versa
 - Edge triggers
 - State depends on change of level
 - $0 \rightarrow 1$: active, $1 \rightarrow 0$: inactive
 - \uparrow : active, \downarrow : inactive
 - Or vice versa

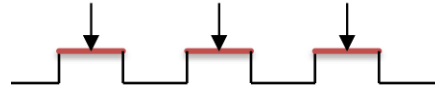


Triggers (3)

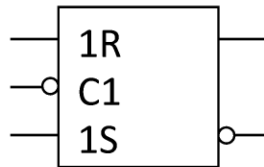
- Graphical Symbols of Triggers



Active Level is $C=1$



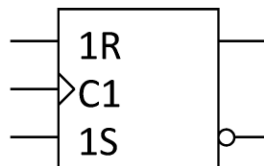
active-high state-triggered



Active Level is $C=0$



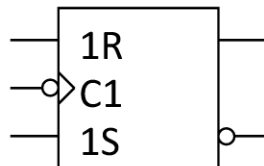
active-low state-triggered



Active Edge is $C=0 \rightarrow 1$



positive-edge triggered



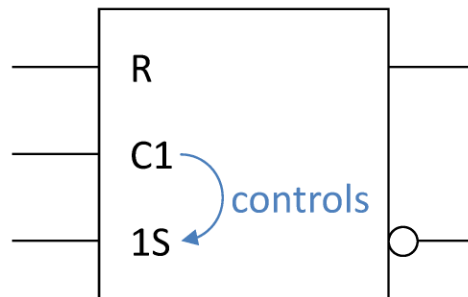
Active Edge is $C=1 \rightarrow 0$



negative-edge triggered

Triggers (4)

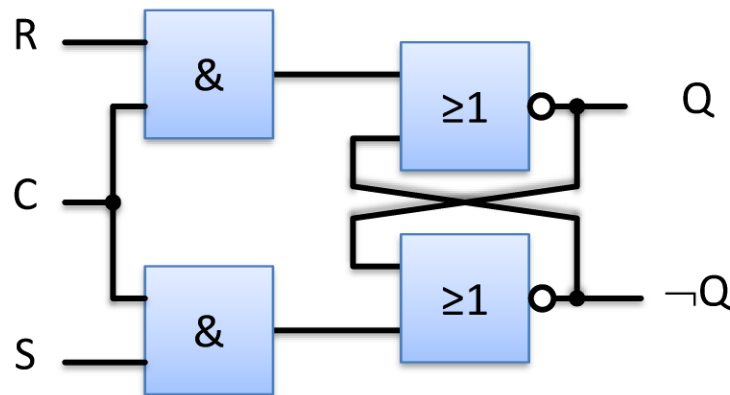
- Synchronous and Asynchronous Lines
 - Not every line is controlled by a clock
 - Synchronous lines are
 - Asynchronous lines are not



State-triggered SR flip-flop
with **synchronous** set
and **asynchronous** reset

Flip-flops (1)

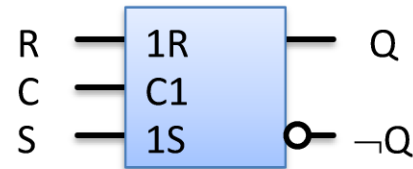
- Synchronous SR Flip-flop
 - Level triggered, active state: 1
 - $C=0$: R und S may change, Q is stable
 - $C=1$: R und S must be stable, Q may change



Flip-flops (2)

- Synchronous SR Flip-flop (level triggered, continued)

- Symbol

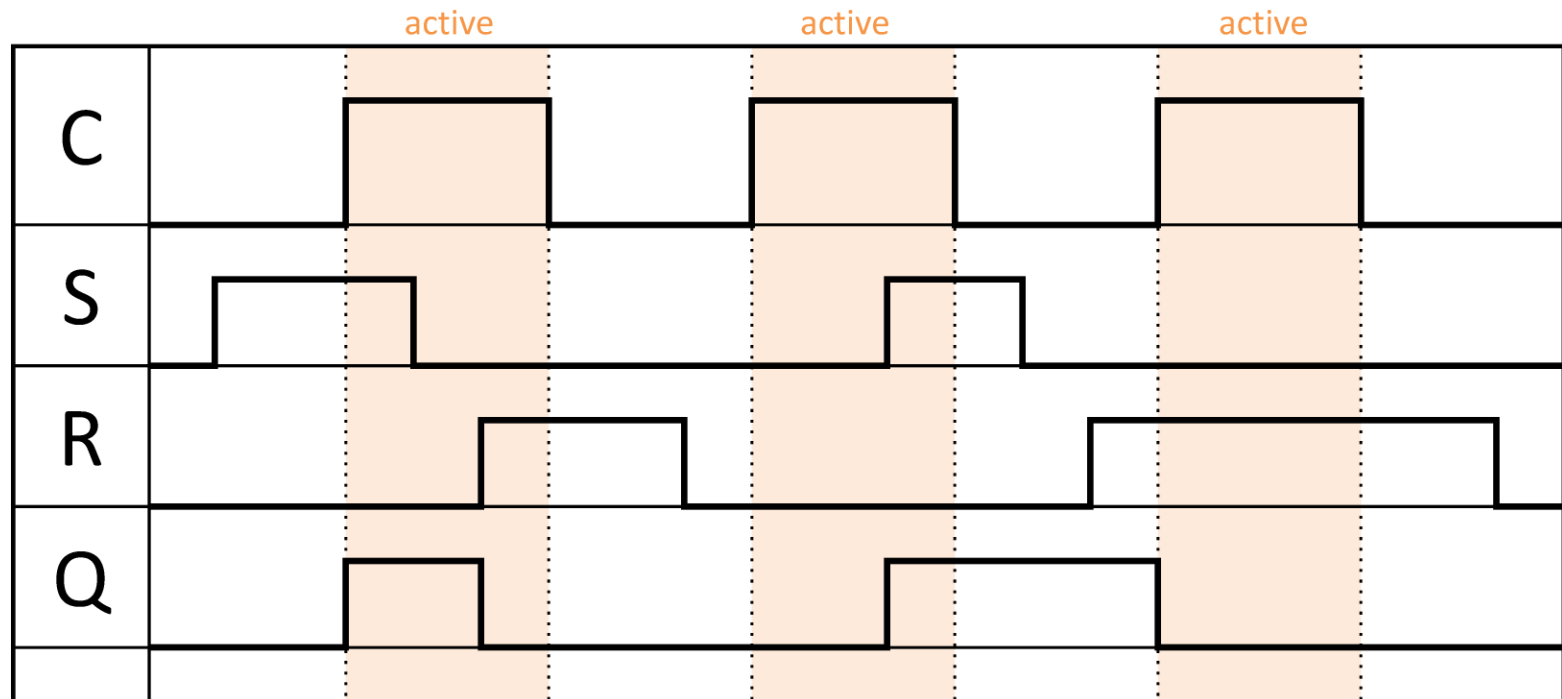


- State table

C	S	R	Q ⁺	Action
0	X	X	Q	Store
1	0	0	Q	Store
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	X	Invalid

Flip-flops (3)

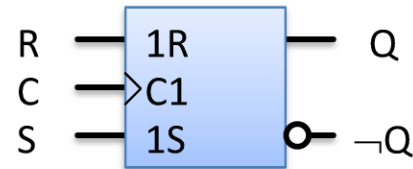
- Synchronous SR Flip-flop (level triggered, finished)
 - Time diagram (active state: 1)



Flip-flops (4)

- Synchronous SR Flip-flop (edge triggered)

- Symbol

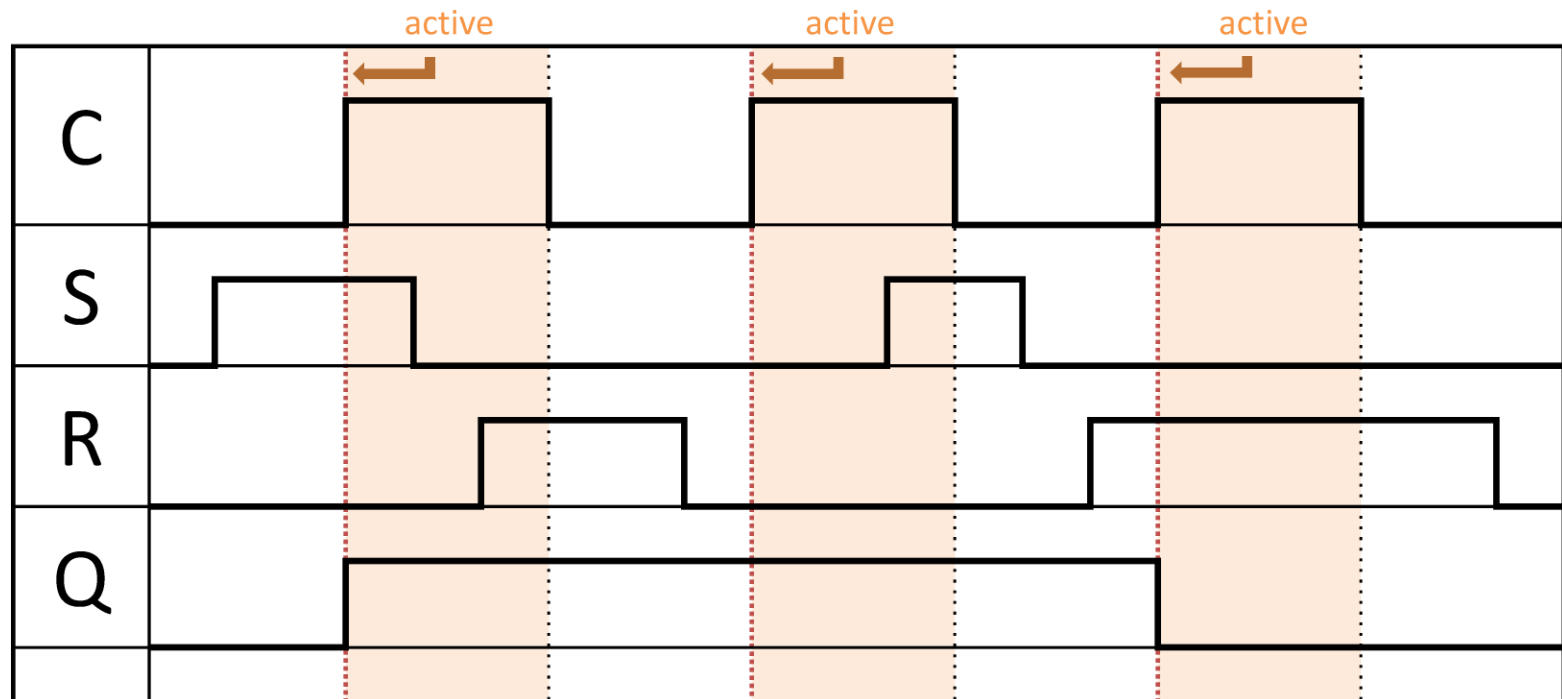


- State table

C	S	R	Q ⁺	Action
X	X	X	Q	Store
↑	0	0	Q	Store
↑	0	1	0	Reset
↑	1	0	1	Set
↑	1	1	X	Invalid

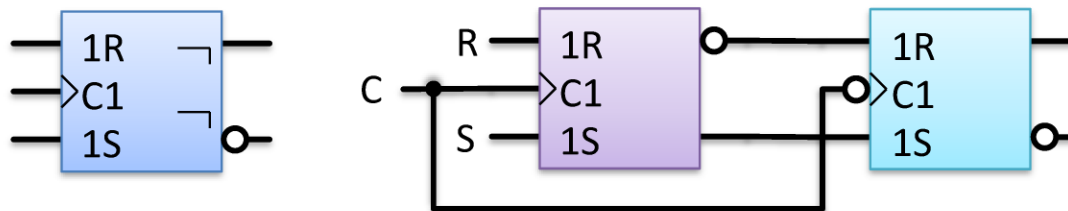
Flip-flops (5)

- Synchronous SR Flip-flop (edge triggered, continued)
 - Time diagram (positive edge triggered)



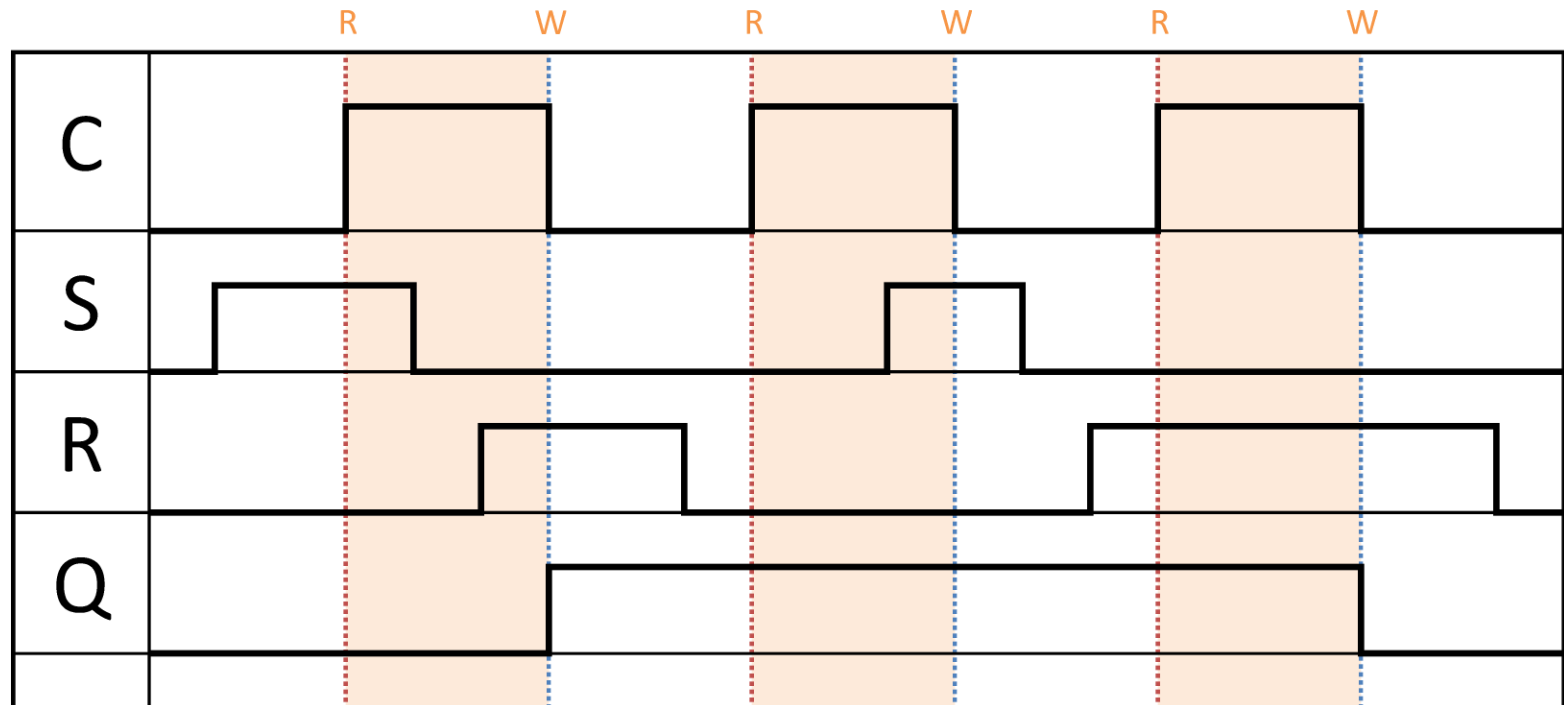
Flip-flops (6)

- Master-slave Flip-flop
 - Input and output are decoupled
 - Output is delayed
 - Positive edge: input is read
 - Negative edge: output is written
 - Build with two edge triggered flip-flops



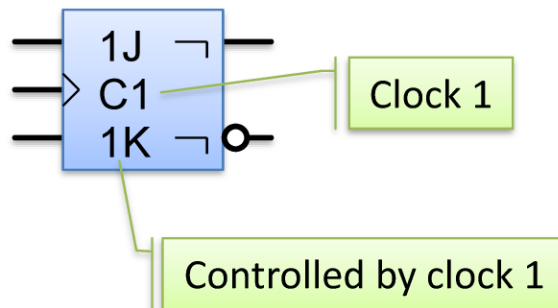
Flip-flops (7)

- Synchronous Master-slave SR Flip-flop
 - Time diagram (positive edge triggered)



Flip-flops (8)

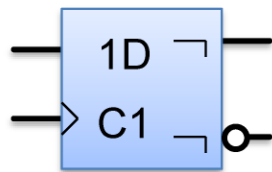
- JK Flip-flop
 - Similar to SR flip-flop
 - No invalid state
 - State gets toggled
 - Toggle: $Q^+ = \neg Q$ ($0 \rightarrow 1, 1 \rightarrow 0$)



J	K	Q^+	Action
0	0	Q	Store
0	1	0	Reset
1	0	1	Set
1	1	$\neg Q$	Toggle

Flip-flops (9)

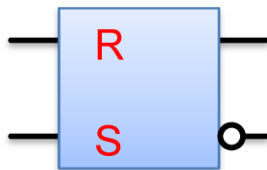
- D Flip-flop
 - Just one input
 - Controlled by clock
 - Input and output are decoupled
 - Input read on positive edge
 - Output written on negative edge
 - Output changes only at certain times



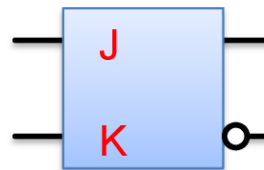
D	Q ⁺	Action
0	0	Reset
1	1	Set

Flip-flops (10)

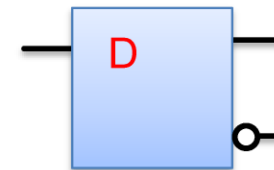
- Naming conventions



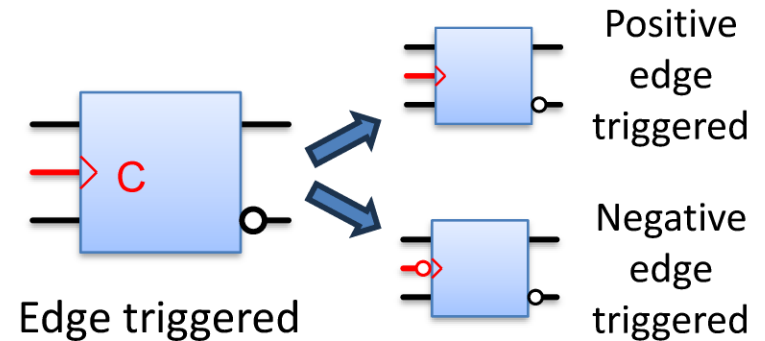
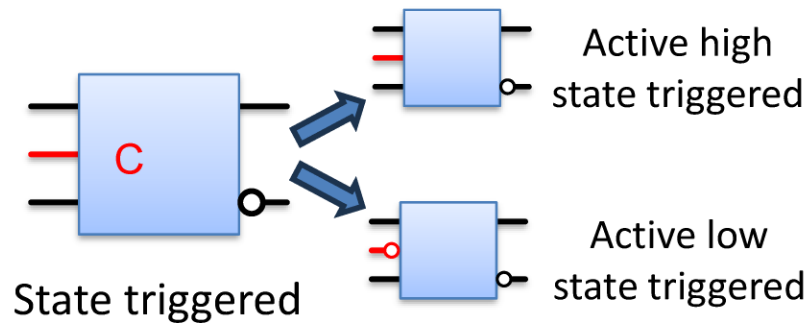
SR flip-flop



JK flip-flop



D flip-flop

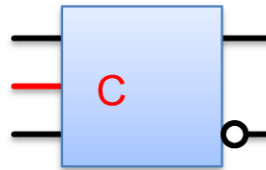


Flip-flops (11)

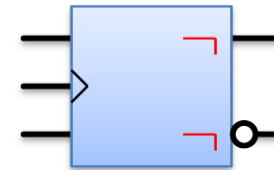
- Naming conventions (continued)



Asynchronous flip-flop



Synchronous flip-flop



Master-slave flip-flop

