Flip-flops

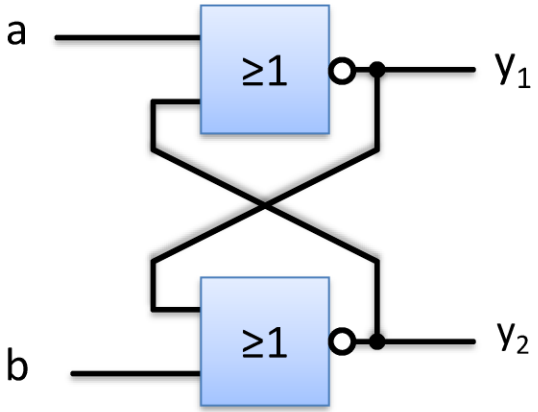
Networks and Embedded Software

Module 3.3.4 (optional)

by Wolfgang Neff

Latches (1)

- Implementation of a SR latch
  - Based on two NOR gates with feedback



Latches (2)

- 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

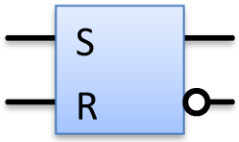
a	b	y_1	y_2	y_1^+	y_2^+
1	*0*	*0*	*0*	*0*	*1*
1	0	0	1	0	1
1	*0*	*1*	*0*	*0*	*0*
1	*0*	*1*	*1*	*0*	*0*
1	1	0	0	0	0
1	*1*	*0*	*1*	*0*	*0*
1	*1*	*1*	*0*	*0*	*0*
1	*1*	*1*	*1*	*0*	*0*

Latches (3)

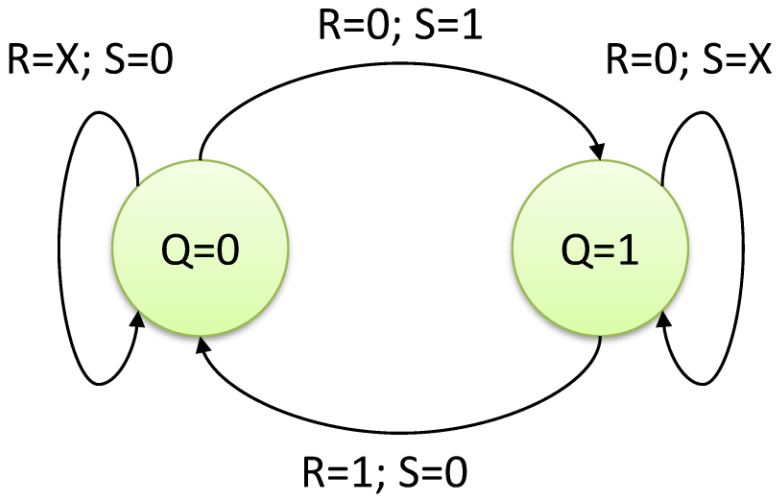
- Characteristics of a stable SR latch
  - a=0, b=0
    stable only if $y_1 \neq y_2$ or $y_1 = \neg y_2$
  - a=0, b=1
    always stable and $y_1 = 1$ (set)
  - a=1, b=0
    always stable and $y_1 = 0$ (reset)
  - a=1, b=1
    invalid since $y_1 = y_2$

Latches (4)

- Symbol, state table and state diagram

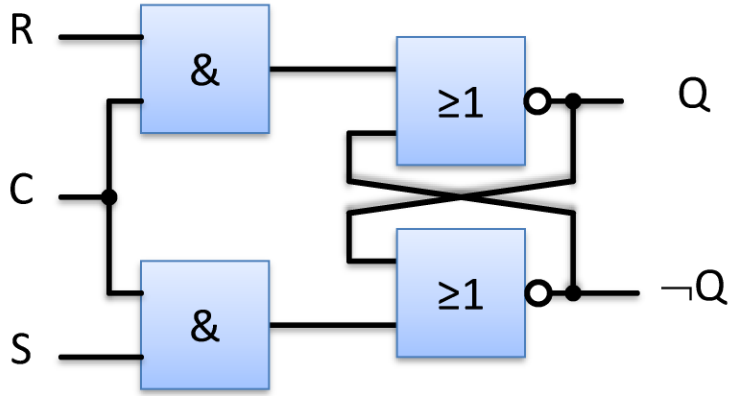


S	R	Q^+	Action
0	0	Q	Store
0	1	0	Reset
1	0	1	Set
1	1	X	Invalid



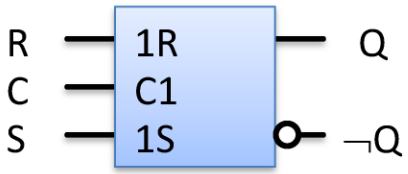
Latches (5)

- Synchronous SR latch
  - C=0: R und S may change, Q is stable
  - C=1: R und S must be stable, Q may change



Latches (6)

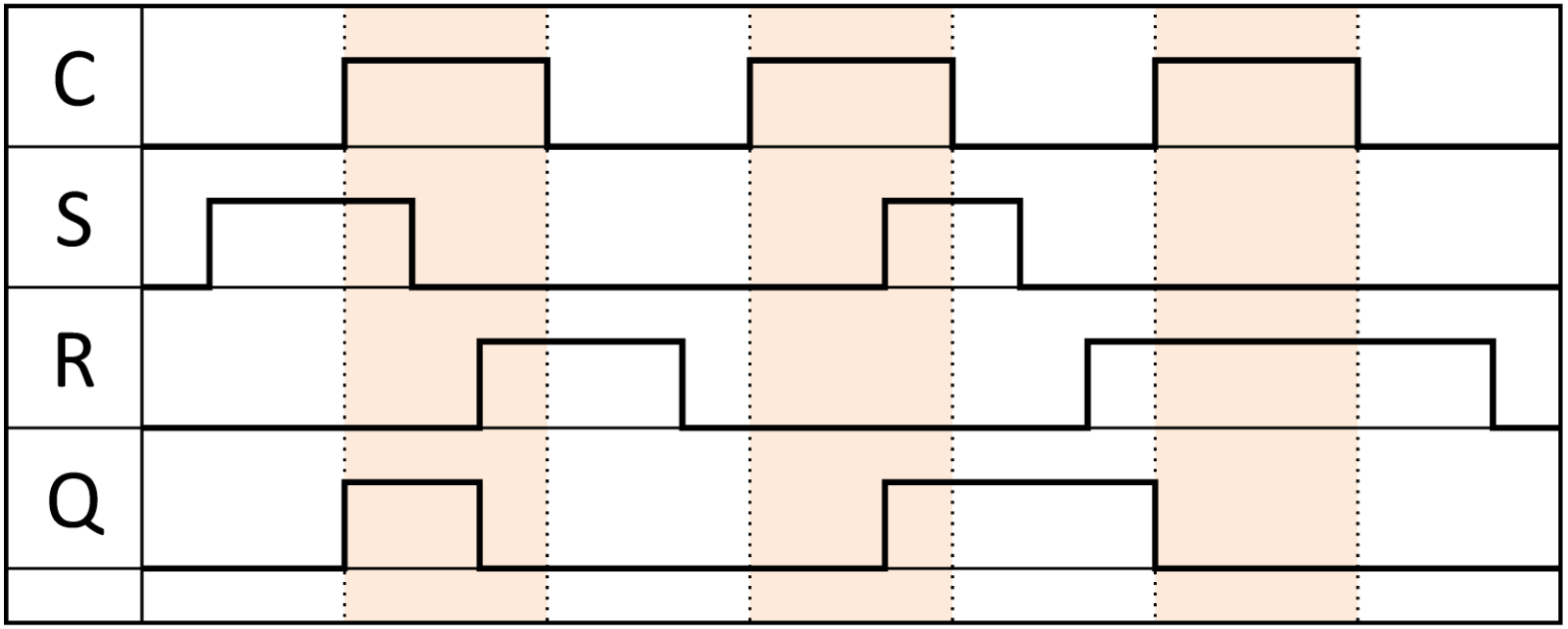
- Symbol and state table of a sync SR latch



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

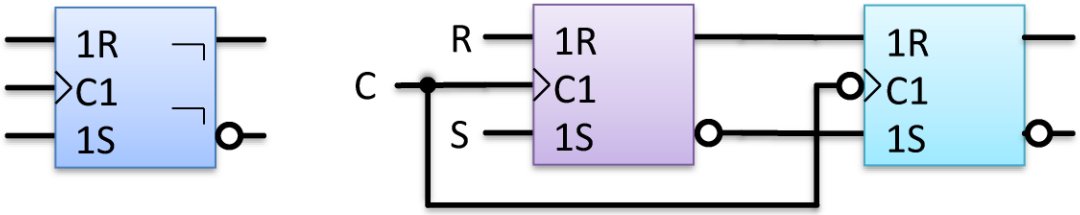
Latches (7)

- Time diagram of a synchronous SR latch
  - Active state: level 1



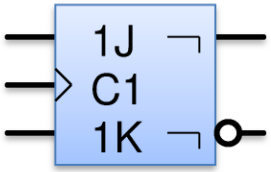
Flip-flop (1)

- Input and output are decupled
  - Edge triggered flip-flops
  - Master-slave flip-flop



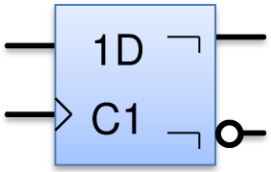
Flip-flop (2)

- The JK flip-flop has no invalid state



J	K	Q^+	Action
0	0	Q	Store
0	1	0	Reset
1	0	1	Set
1	1	¬Q	Toggle

- The D flip-flop is used to store data



D	Q^+	Action
0	0	Reset
1	1	Set