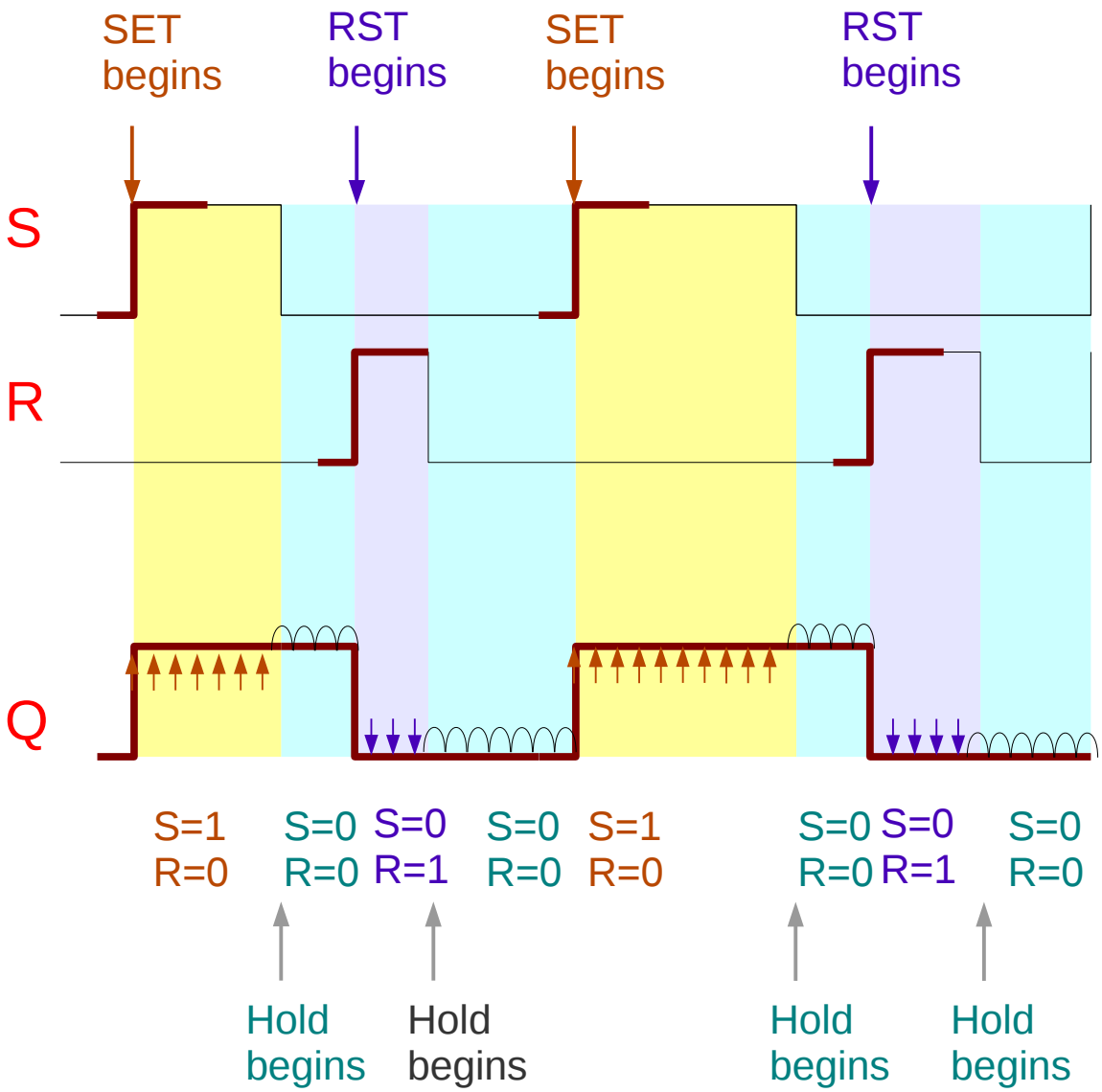
Sequential Gates

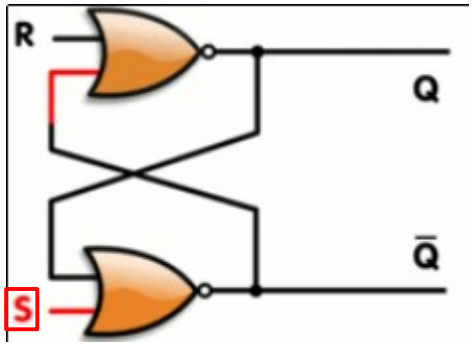
Gate Level Design

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

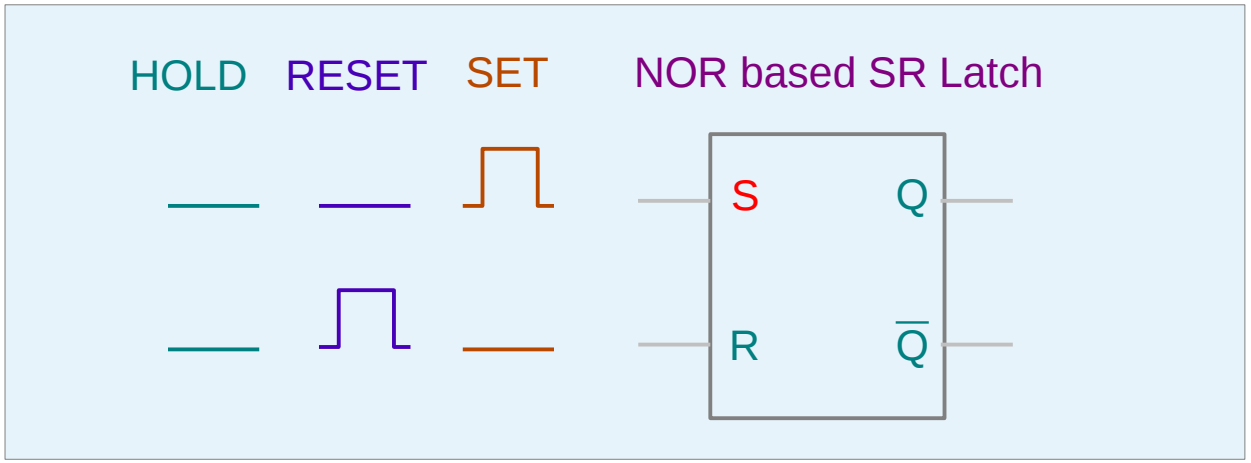
NOR-based SR Latch

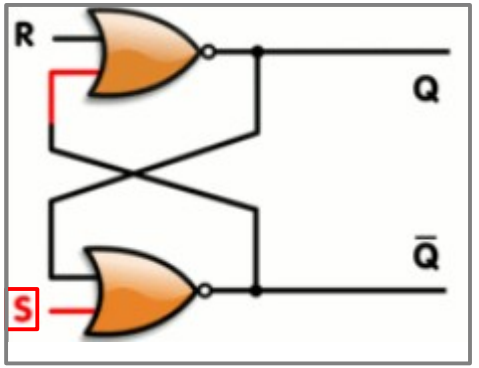


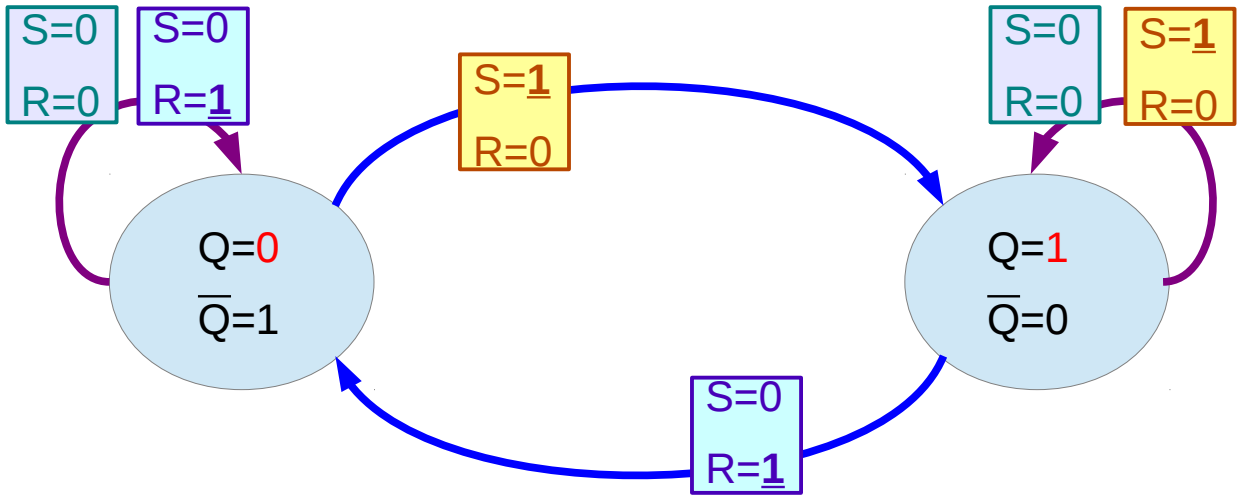


SET	S=1, R=0	Q=1, $\overline{Q}$=0
RESET	S=0, R=1	Q=0, $\overline{Q}$=1
HOLD	S=0, R=0	Q=old Q, $\overline{Q}$=old $\overline{Q}$

NOR-based SR Latch States

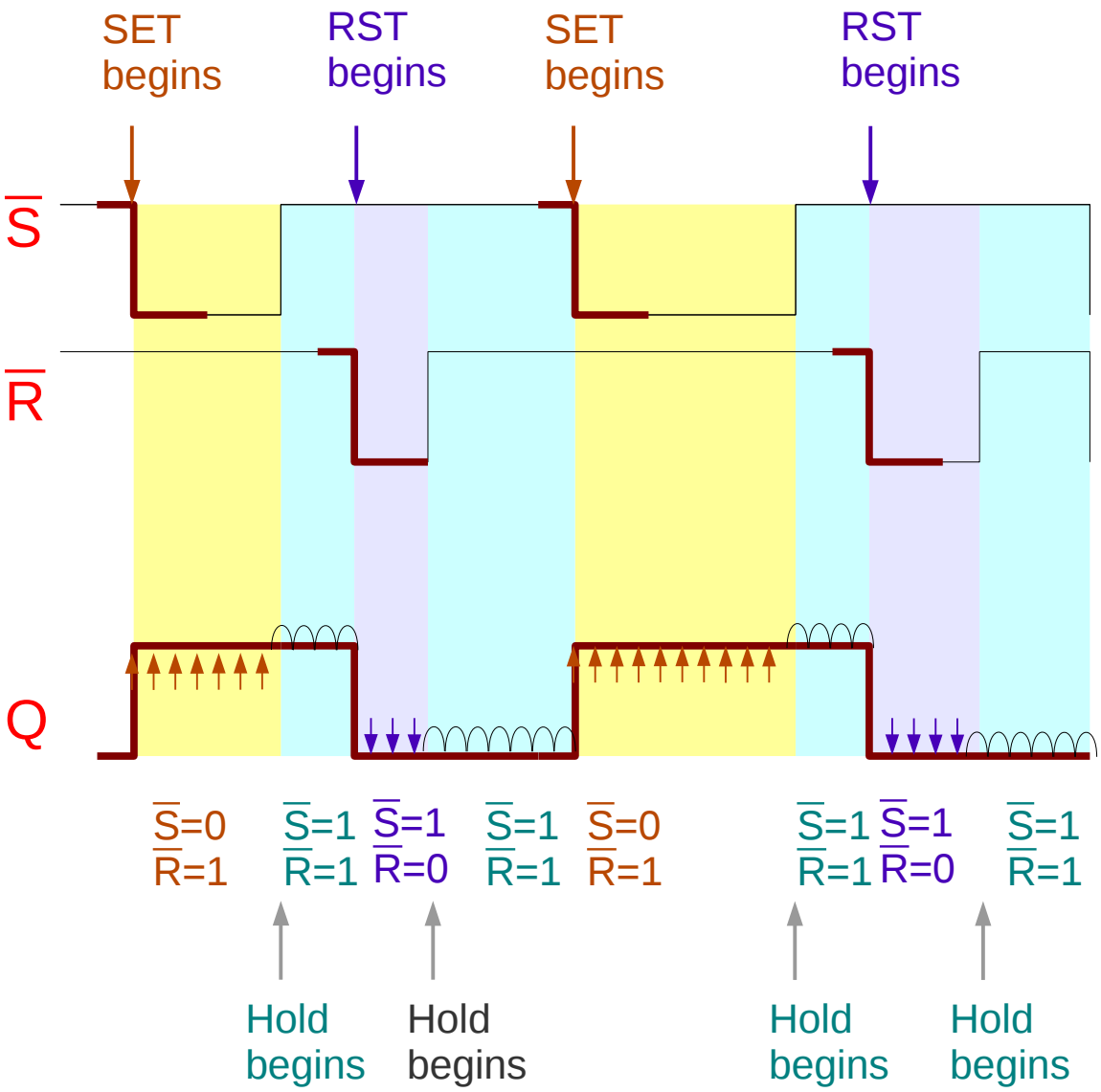


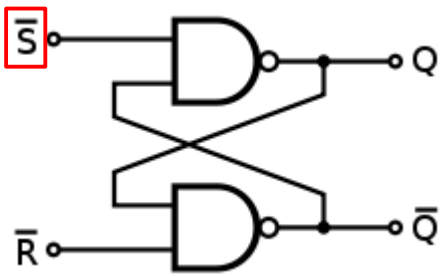




SET	S=1 R=0	Q=1 $\overline{Q}$=0
RESET	S=0 R=1	Q=0 $\overline{Q}$=1
HOLD	S=0 R=0	Q=old Q $\overline{Q}$=old $\overline{Q}$

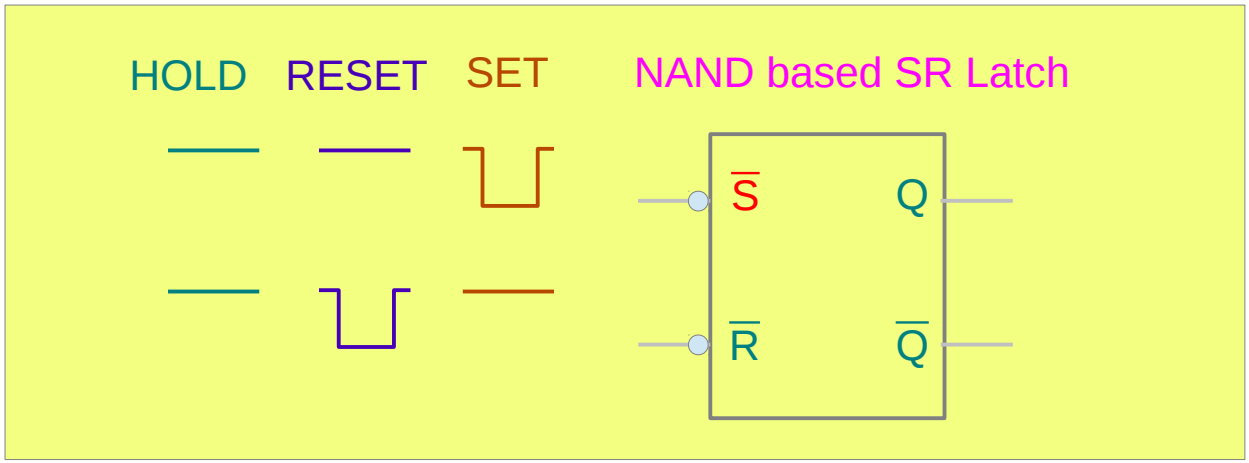
NAND-based SR Latch

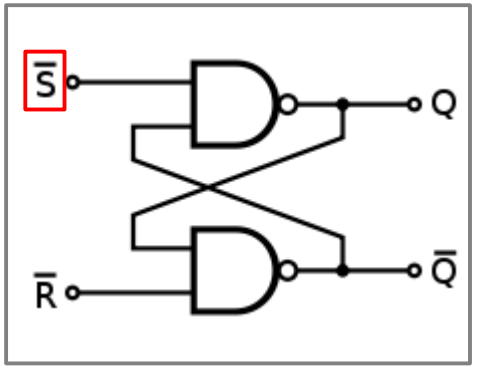


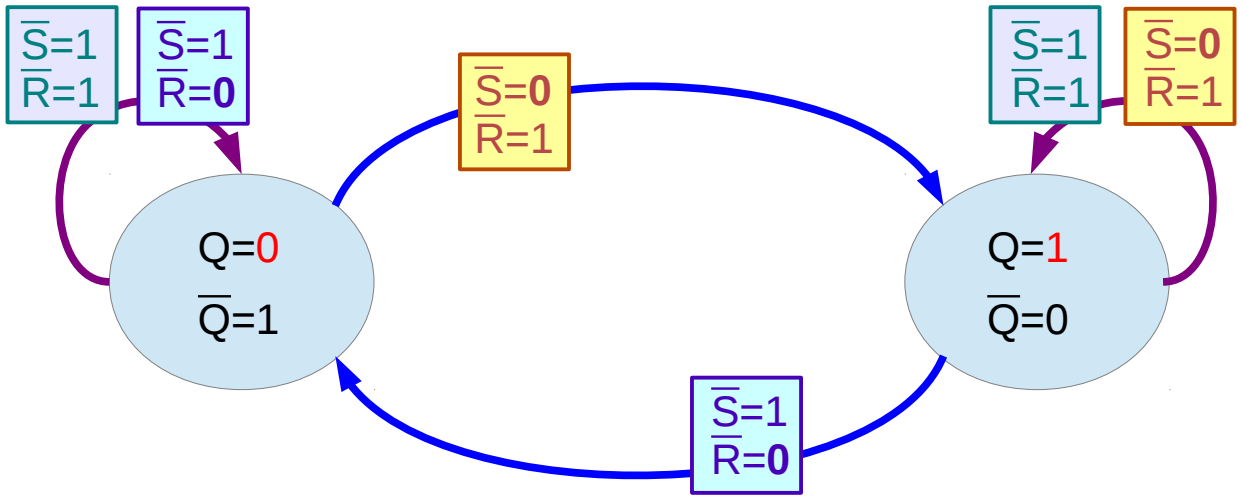


SET	$\overline{S}=0$ $\overline{R}=1$	Q=1 $\overline{Q}=0$
RESET	$\overline{S}=1$ $\overline{R}=0$	Q=0 $\overline{Q}=1$
HOLD	$\overline{S}=1$ $\overline{R}=1$	Q=old Q $\overline{Q}$=old $\overline{Q}$

NAND-based SR Latch States

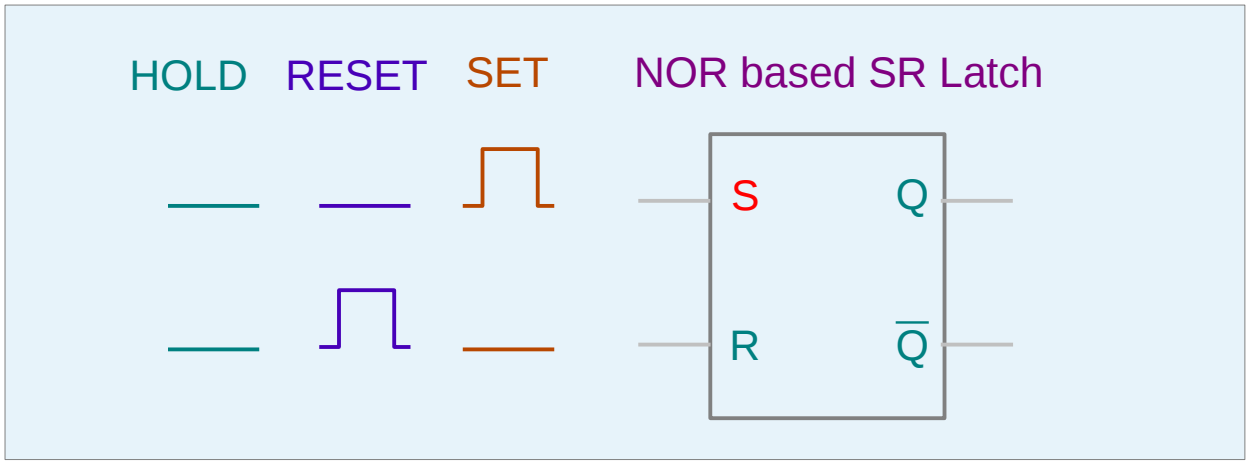


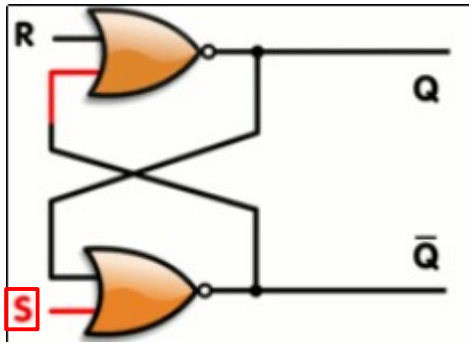


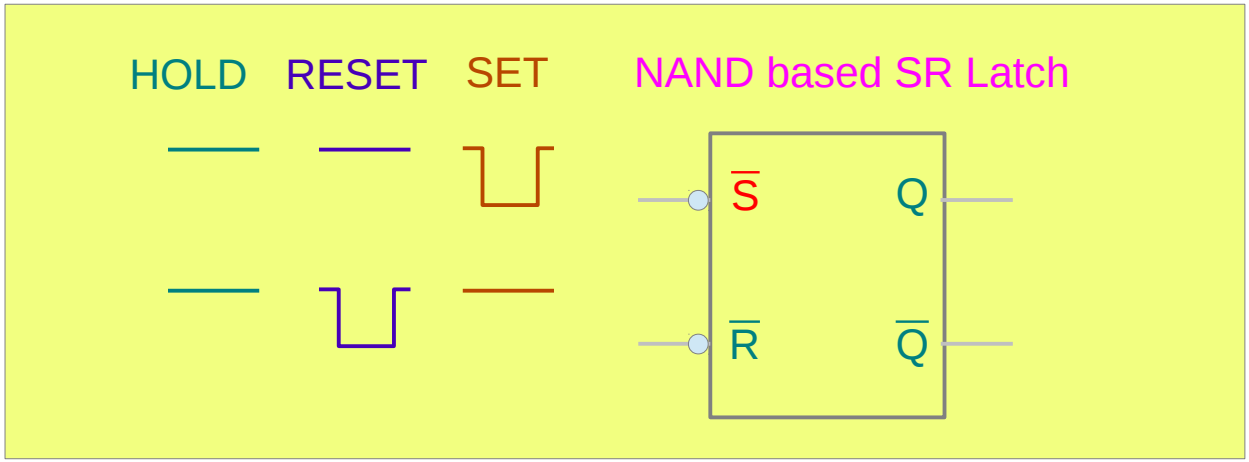


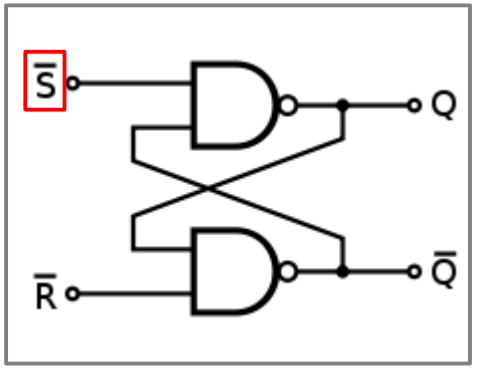
SET	$\overline{S}=0$ $\overline{R}=1$	Q=1 $\overline{Q}=0$
RESET	$\overline{S}=1$ $\overline{R}=0$	Q=0 $\overline{Q}=1$
HOLD	$\overline{S}=1$ $\overline{R}=1$	Q=old Q $\overline{Q}$=old $\overline{Q}$

SR Latch Symbols



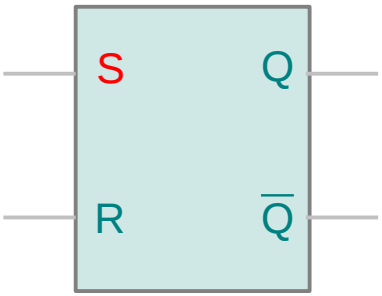




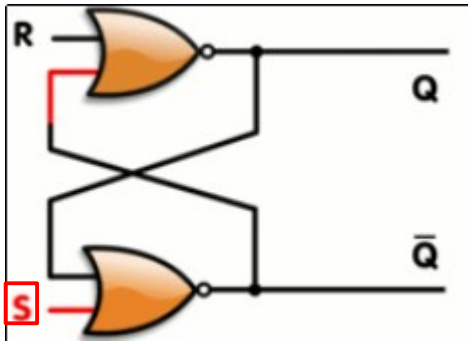


Active High and Low Inputs

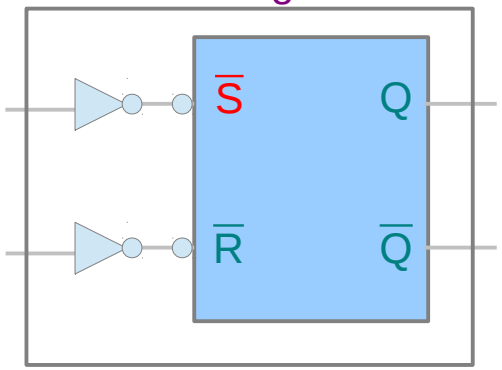
Active High



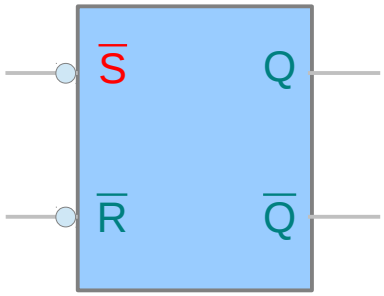
Active High



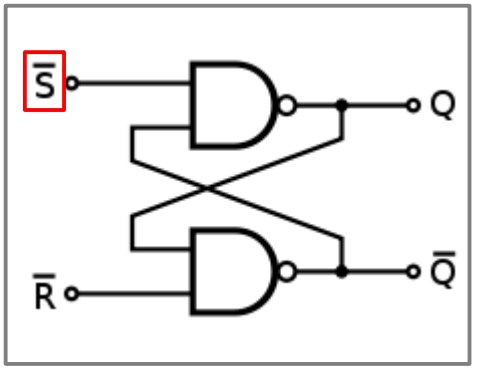
Active High



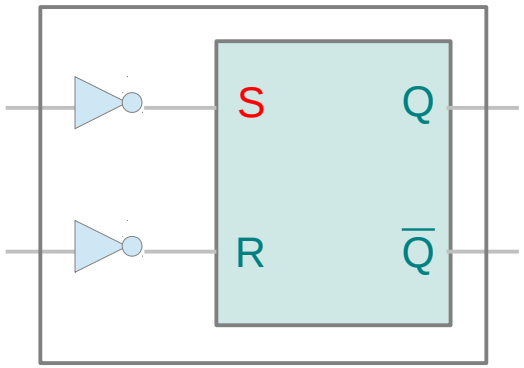
Active Low



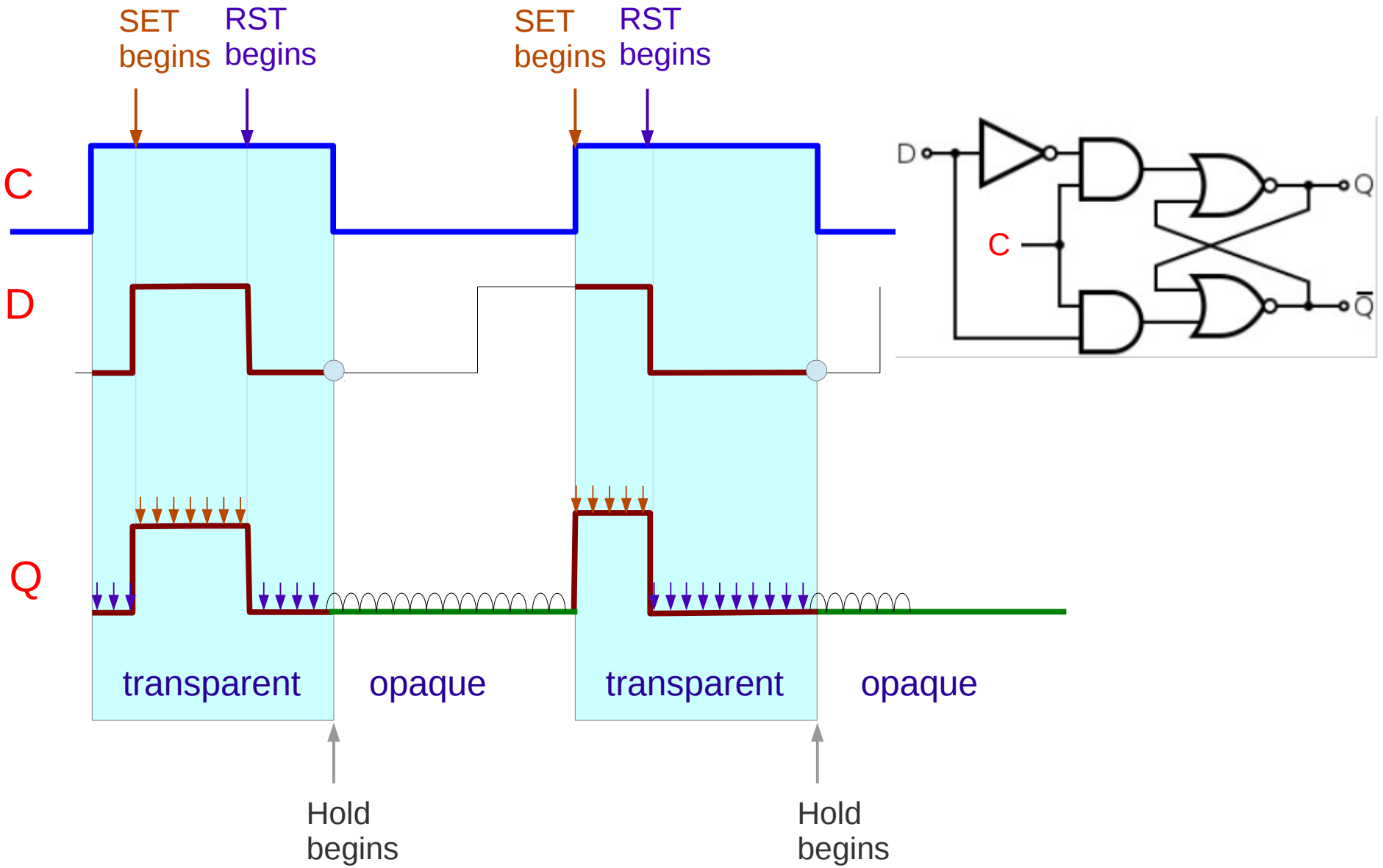
Active Low



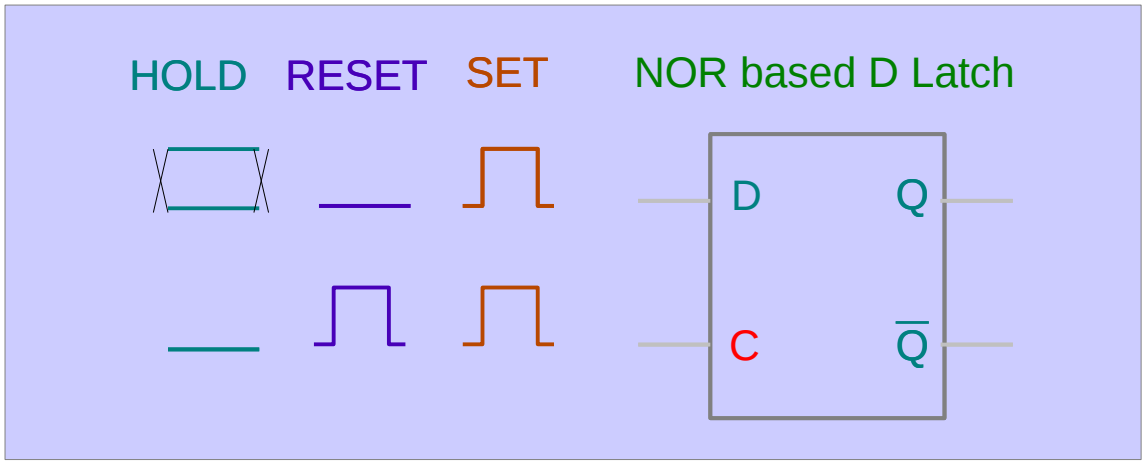
Active Low

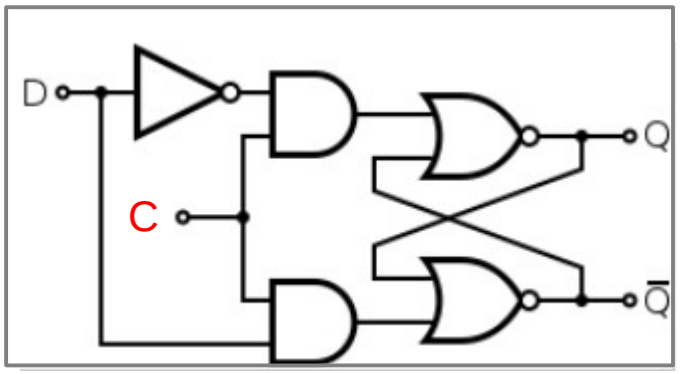


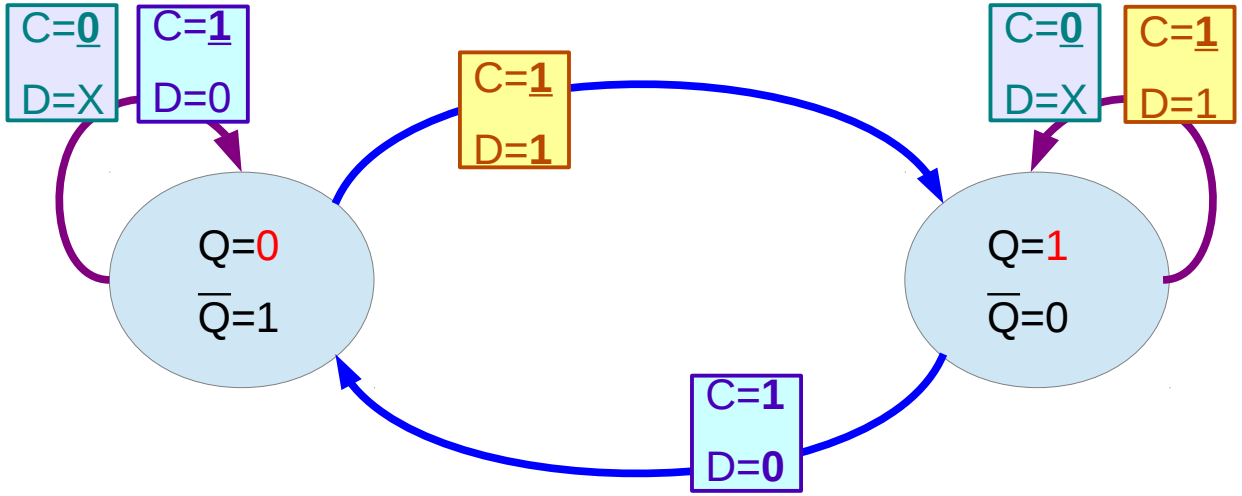
NOR-based D Latch



NOR-based D Latch

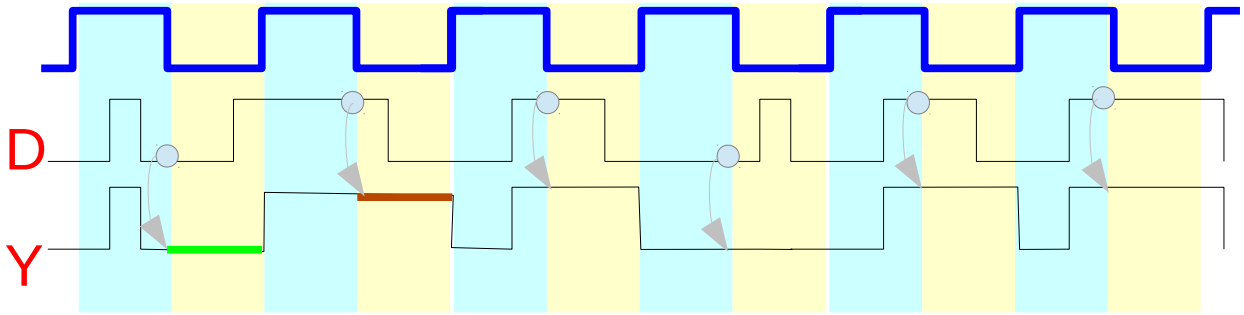


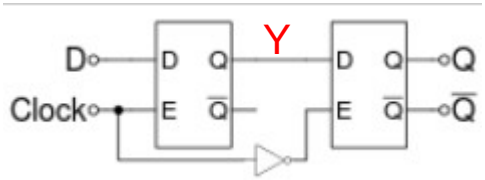




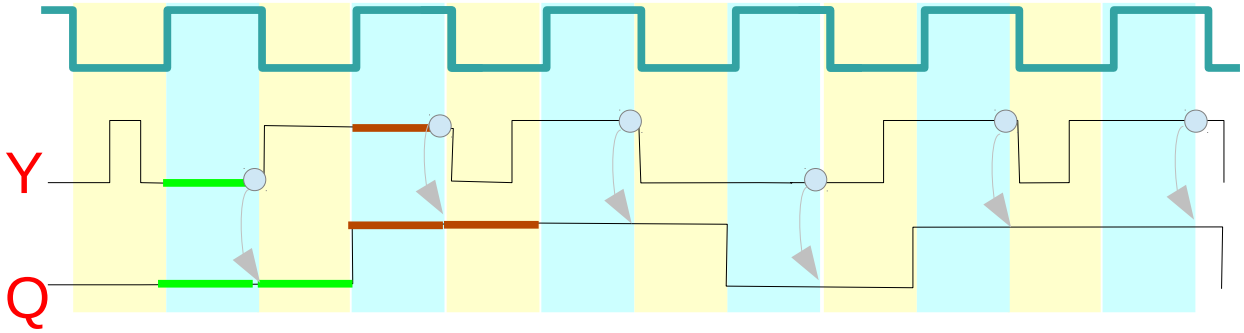
Master-Slave D FlipFlop

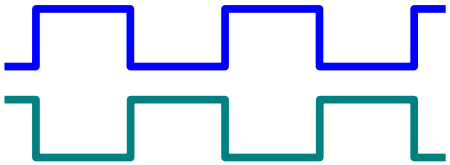
Master D Latch

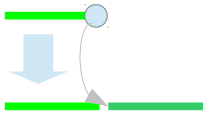




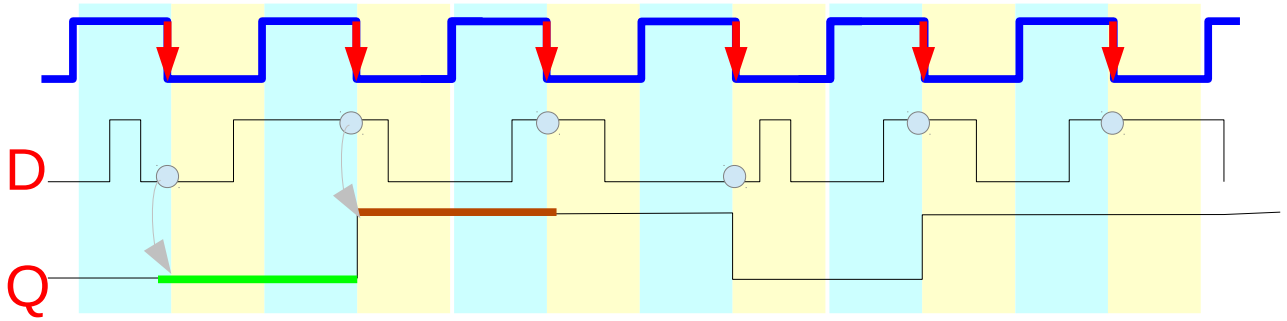
Slave D Latch







Master-Slave D F/F

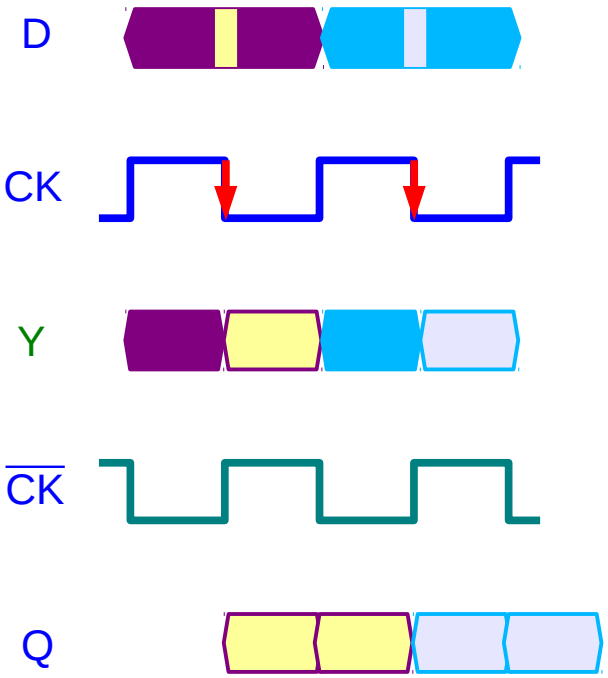


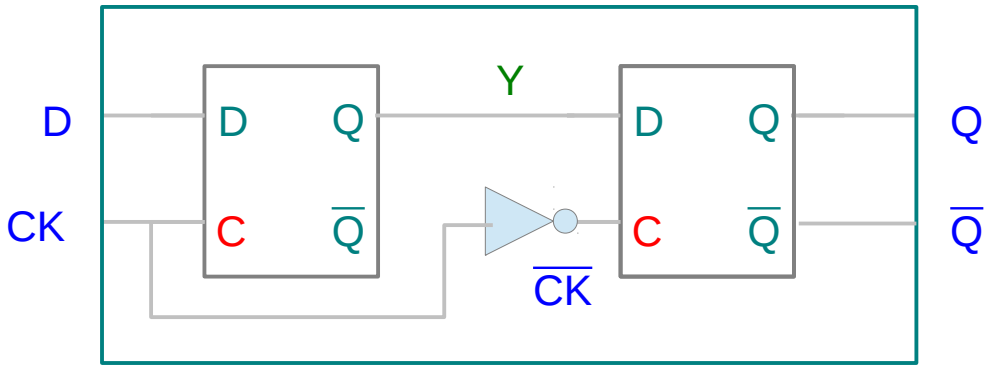
the hold output of the master is transparently reaches the output of the slave

this value is held for another half period

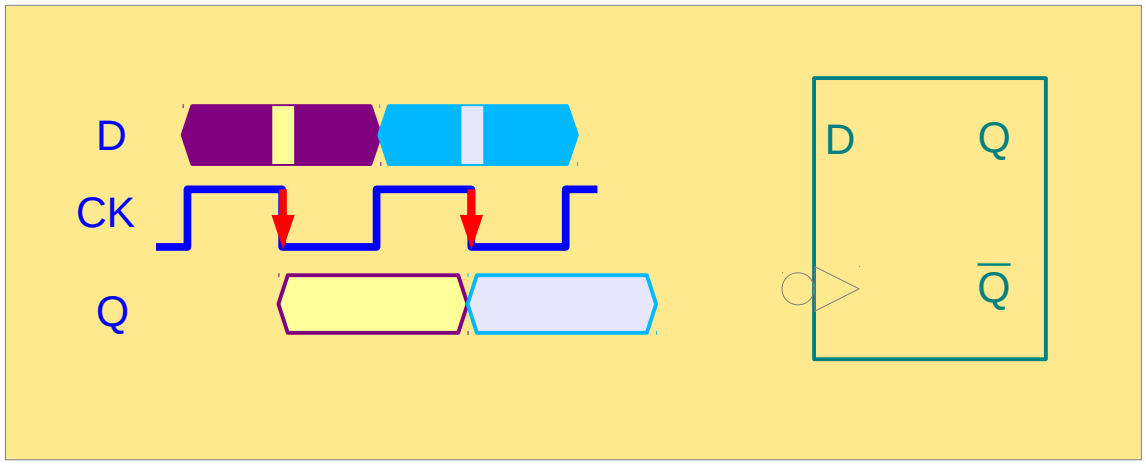
Master-Slave D FlipFlop – Falling Edge

Master D Latch



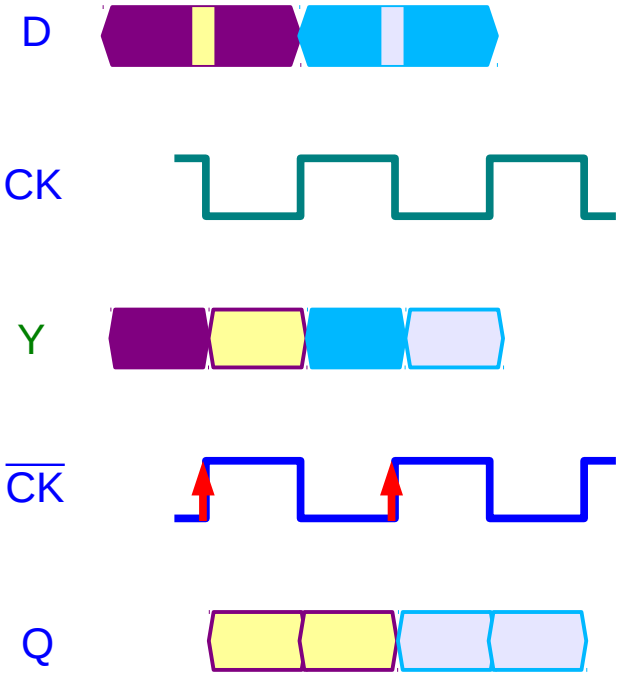


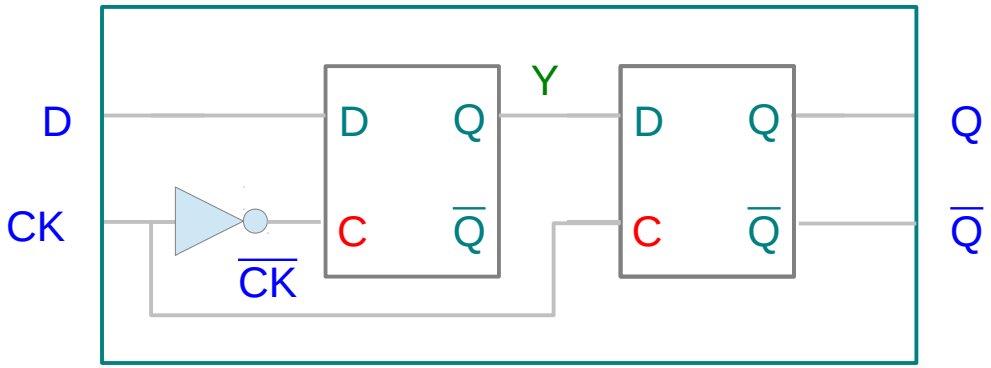
Slave D Latch



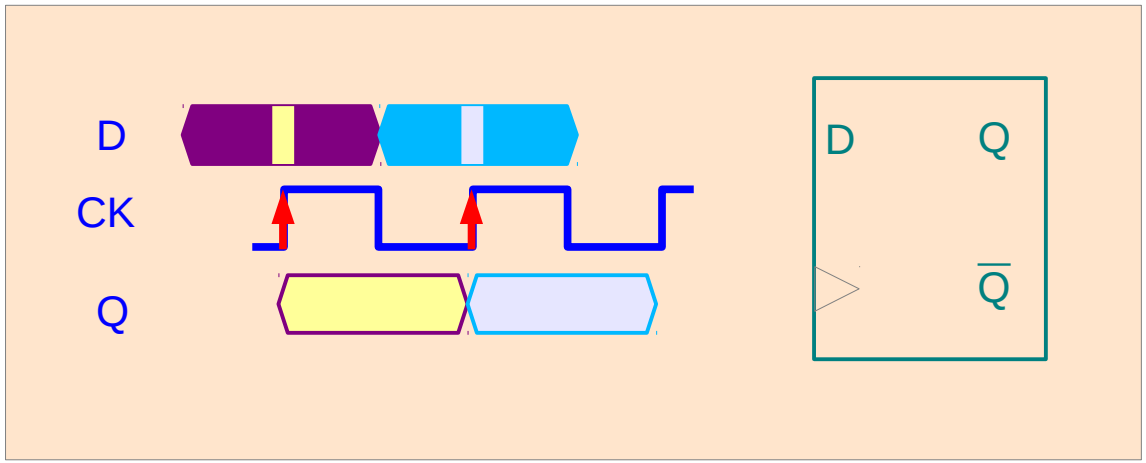
Master-Slave D FlipFlop – Rising Edge

Master D Latch





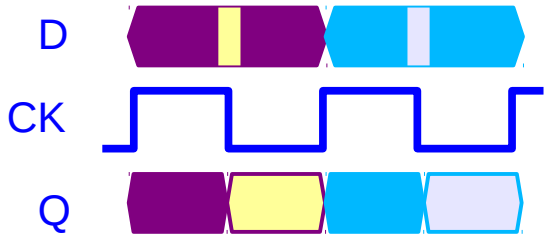
Slave D Latch

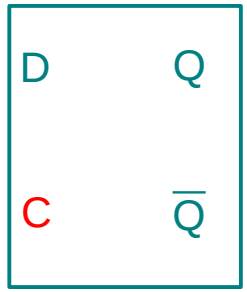


D Latch & D FlipFlop

Level Sensitive D Latch

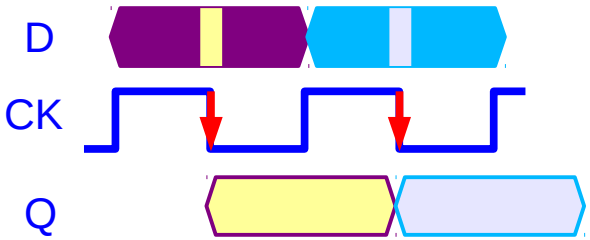
CK=1 transparent
CK=0 opaque

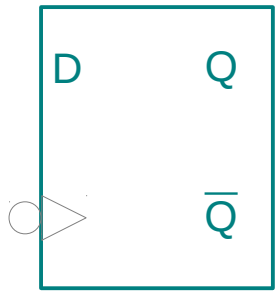




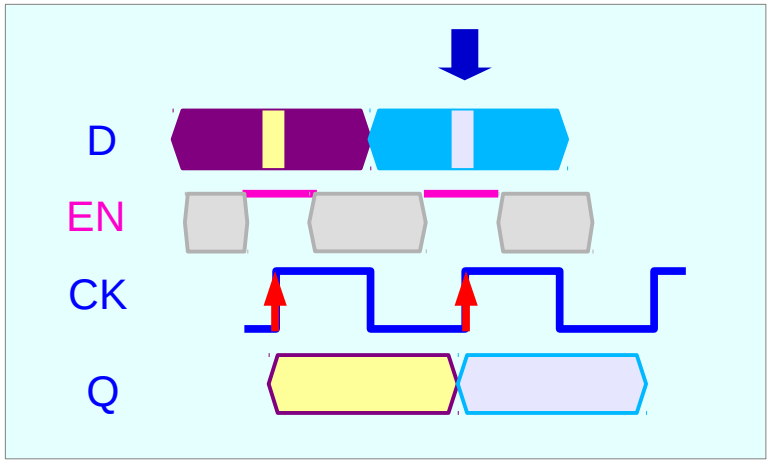
Edge Sensitive D FlipFlop

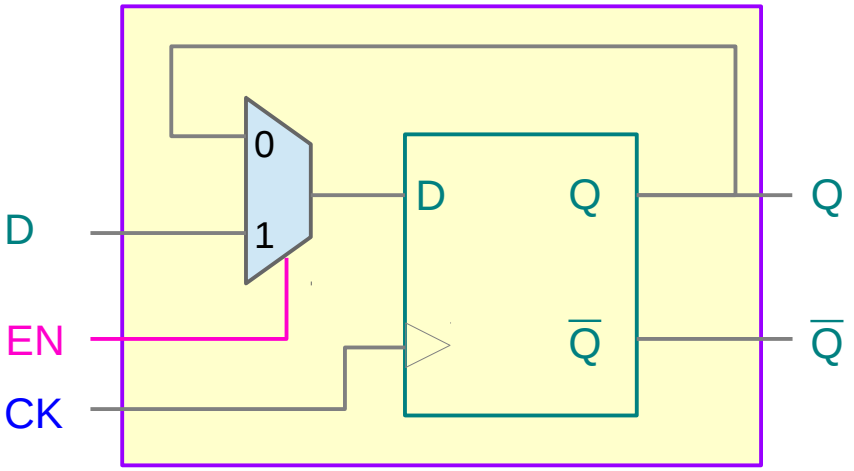
CK=1→0 transparent
else opaque

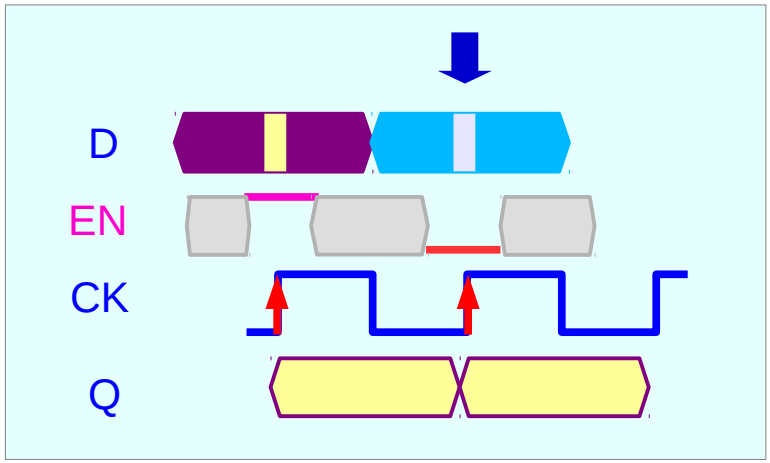


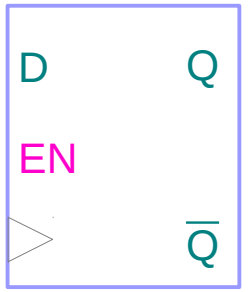


D FlipFlop with Enable









Advantages of Latches over FFs

Flipflop designs are very **easy to verify timing**

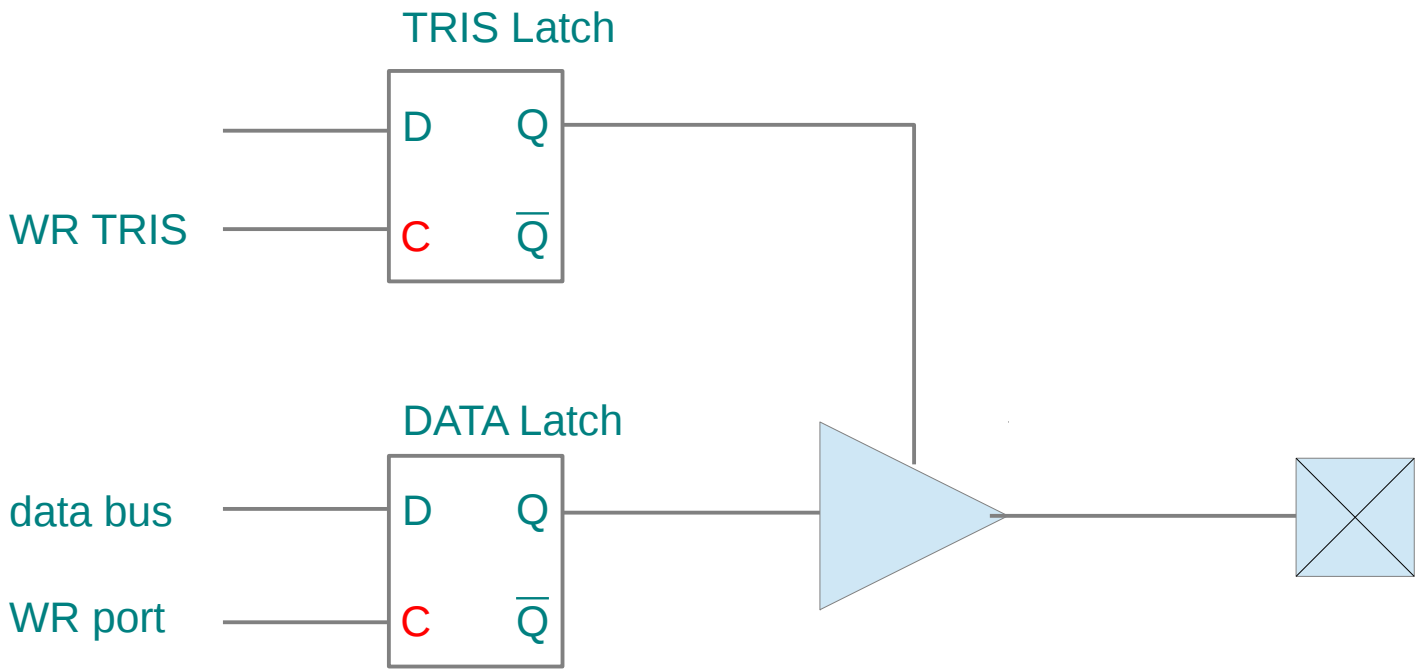
- Each path between flip-flops must be less than the clock period
- Tools check for skew, setup, and hold time violations
- Short paths are padded
  (buffers are added to slow down the signals)
- Skew in flip-flop based systems affects the critical path

Most designs in industry are based on flip-flops

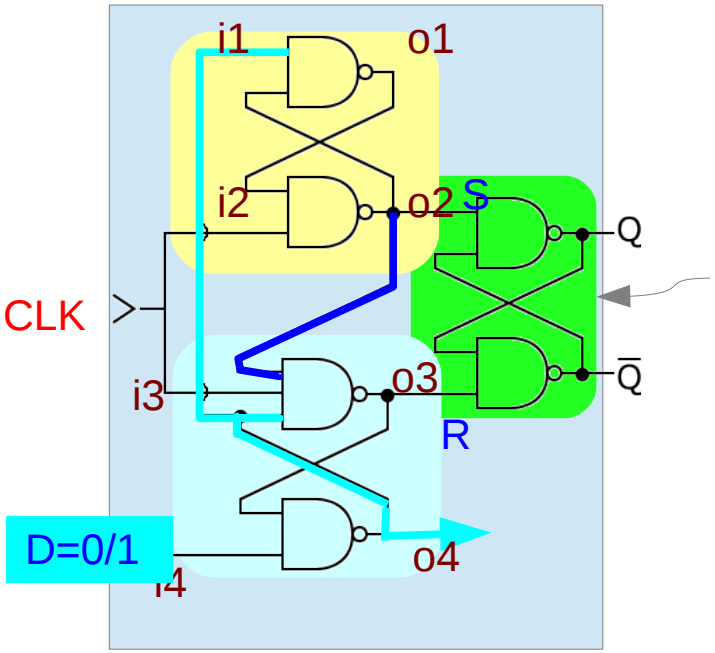
Latch designs are **more flexible** than a flip-flop design

- Need to CAD tools to make sure it works
- Can borrow time to allow a path to be longer than clock period
- Can tolerate clock skew
  -- skew does not directly add to cycle time
- Less silicon area

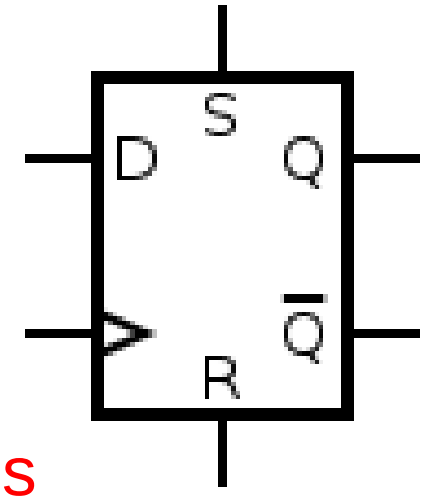
Latches at the output ports



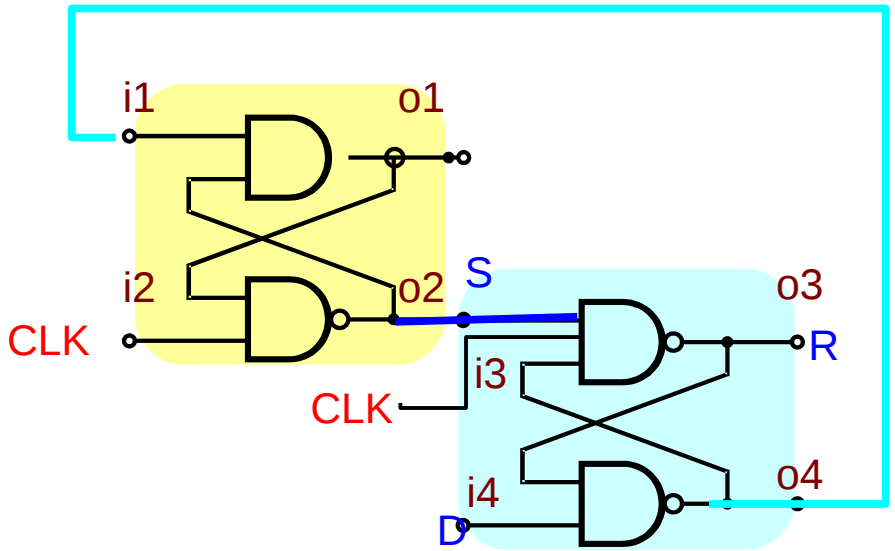
Classical Edge Triggered FF



Output Stage Latch

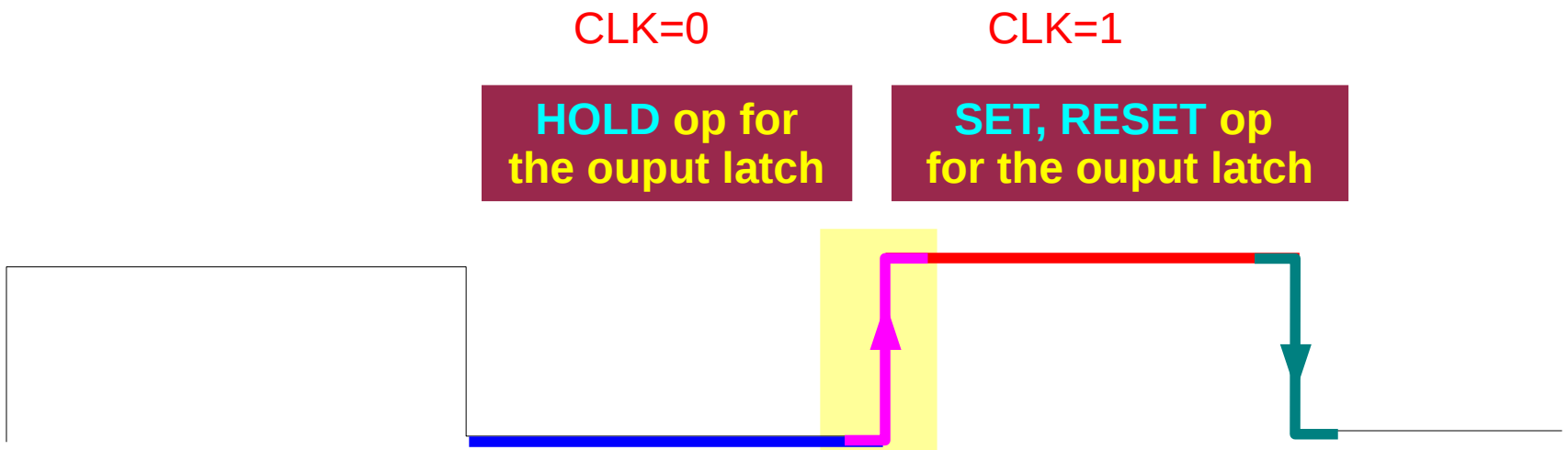


Input Stage Latches

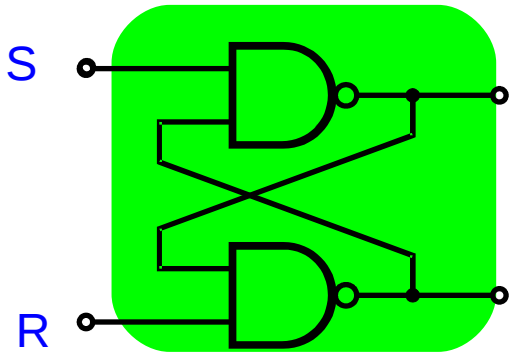


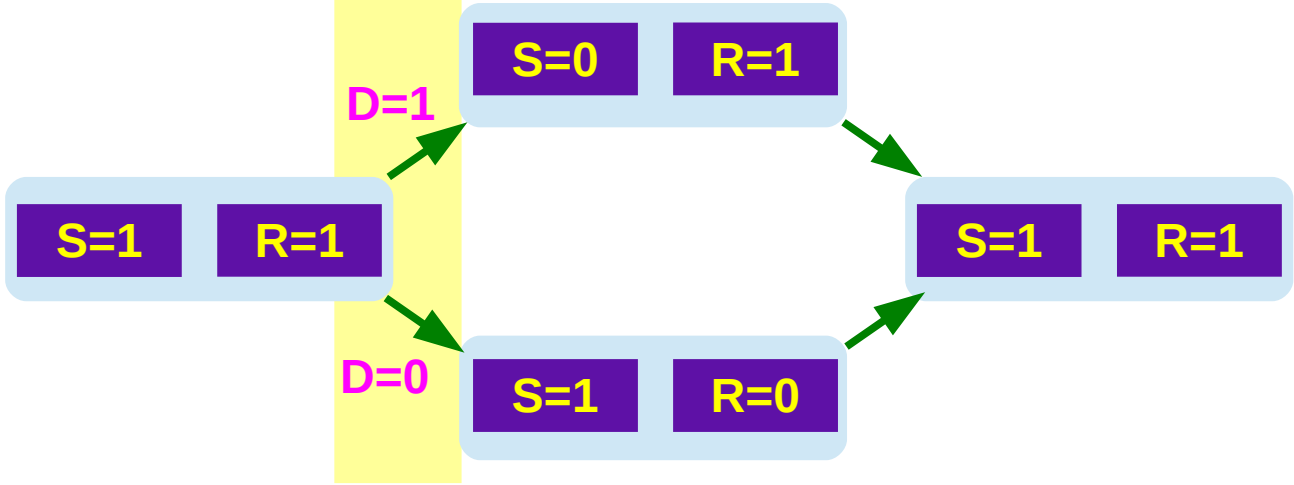
* figures from
wikipedia.org

Classical Edge Triggered FF

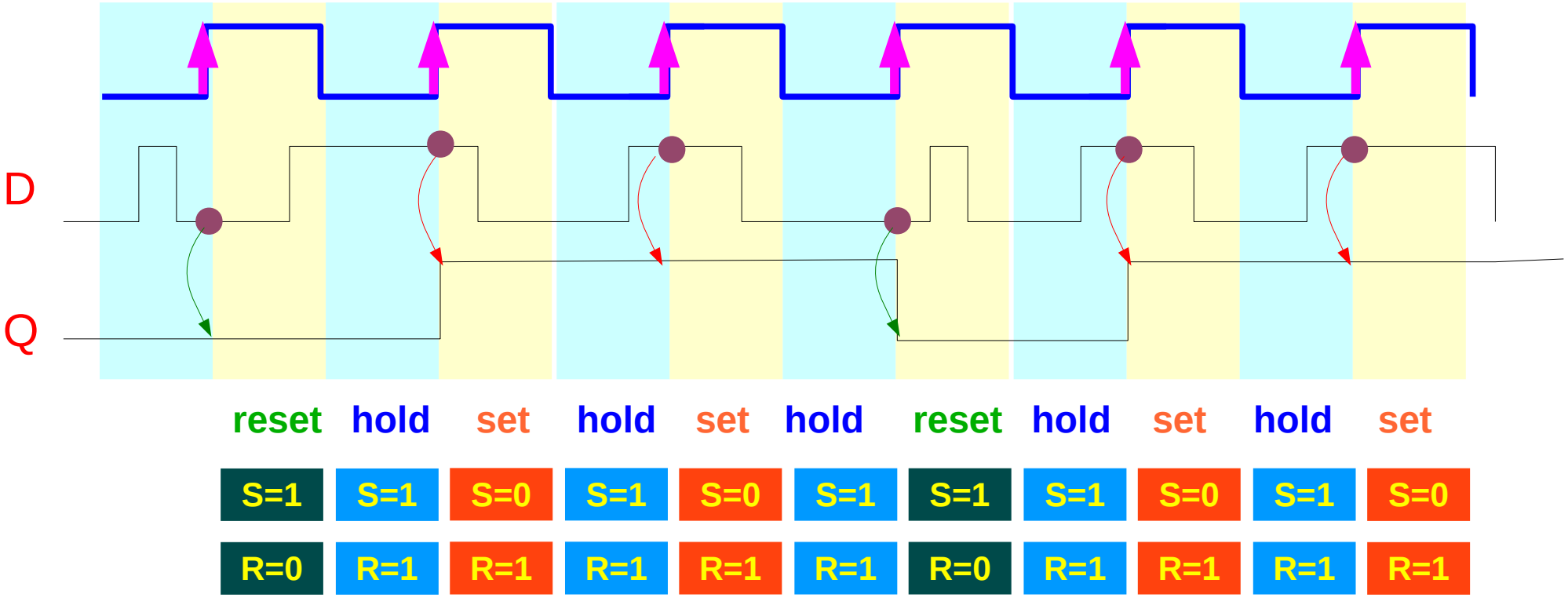


Output Stage Latch

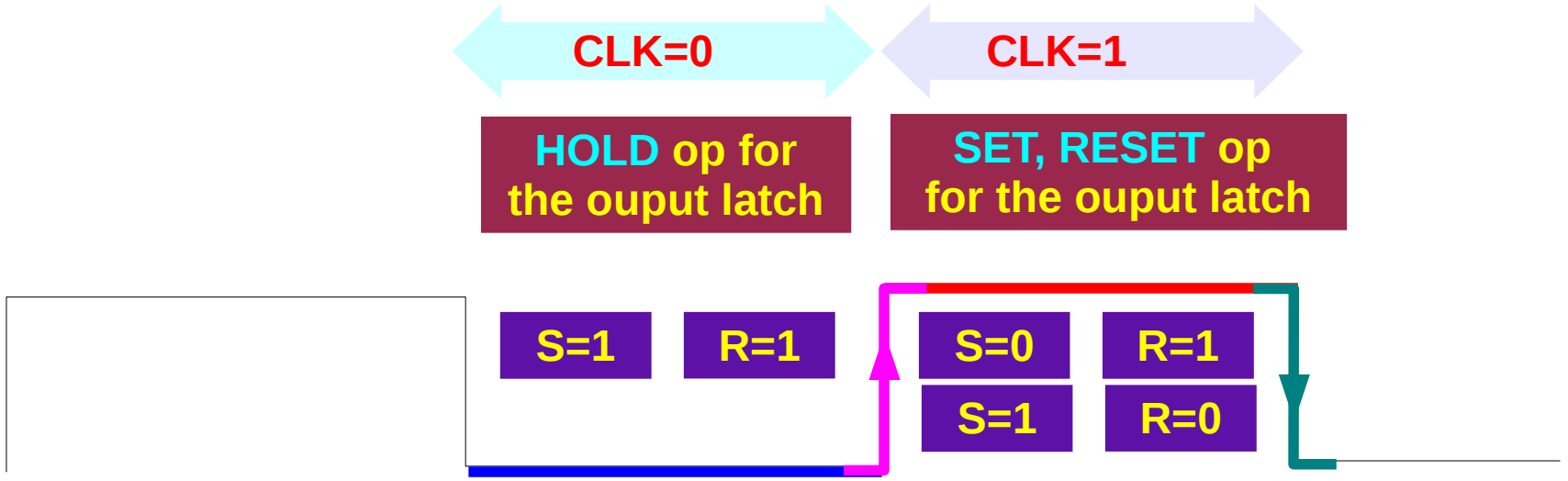




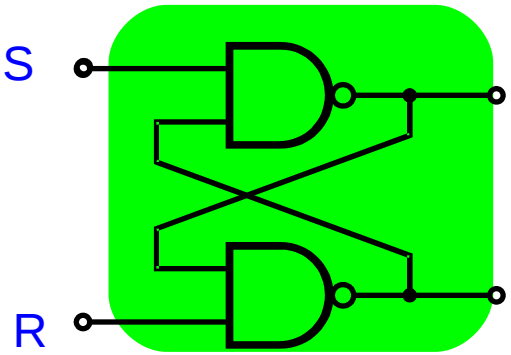
Classical Edge Triggered FF



Classical Edge Triggered FF



Output Stage Latch



CLK=0 → 1

When D=0
HOLD → RESET

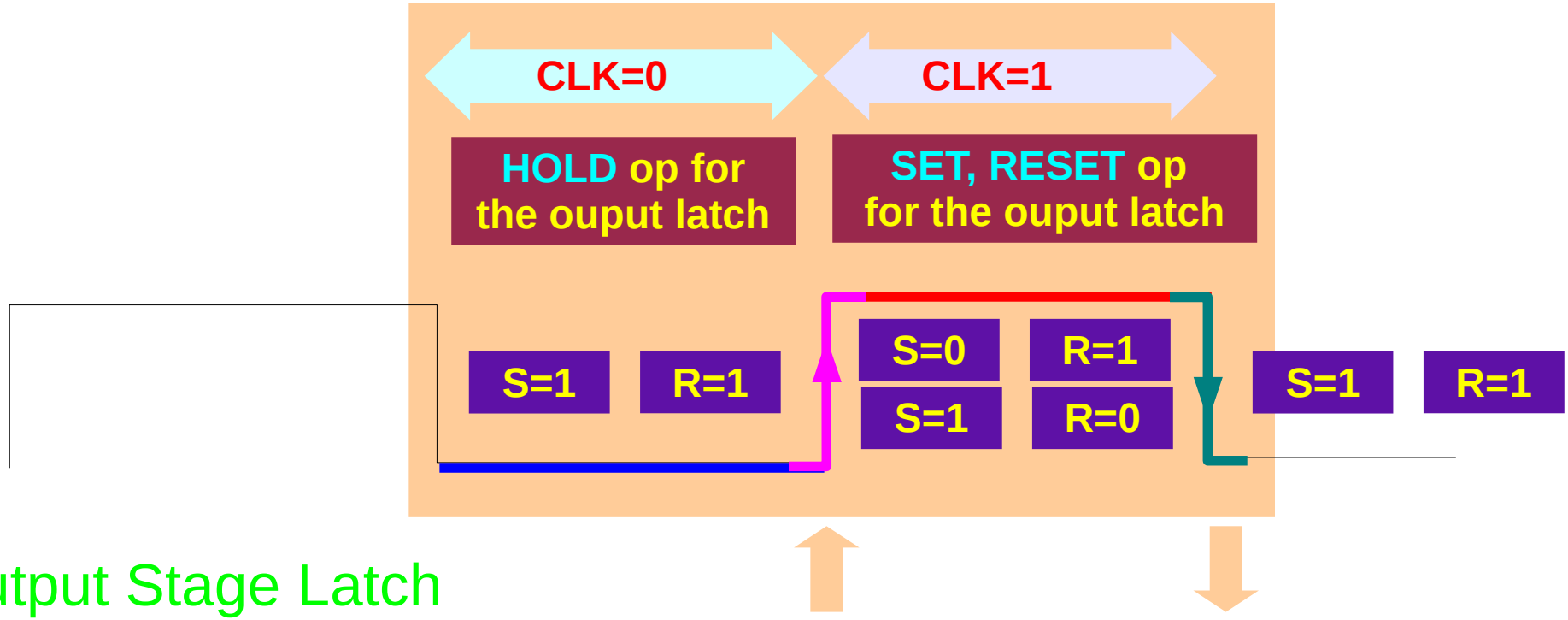
When D=1
HOLD → SET

CLK=1 → 0

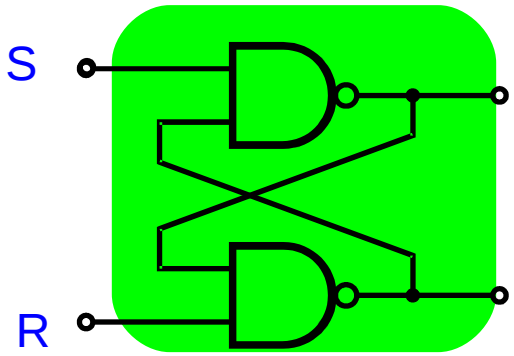
RESET → HOLD

SET → HOLD

Output Latch Operation (Classical Edge Triggered FF)



Output Stage Latch



CLK=0 → 1

When D=0
HOLD → RESET

When D=1
HOLD → SET

CLK=1 → 0

RESET → HOLD

SET → HOLD

Sequential Element Types

CMOS Latches
CMOS FlipFlops
Pulsed Latches
Latches and FlipFlops with a Reset
Latches and FlipFlops with an Enable

References

[1] http://en.wikipedia.org/
[2] http://www.allaboutcircuits.com/
[3] W. Wolf, “Modern VLSI Design : Systems on Silicon
[4] N. Weste, D. Harris, “CMOS VLSI Design: A Circuits and Systems Perspective”
[5] J. P. Uyemura, “Introduction to VLSI Circuits and Systems”
[6] https://en.wikiversity.org/wiki/The_necessities_in_SOC_Design
[7] https://en.wikiversity.org/wiki/The_necessities_in_Digital_Design
[8] https://en.wikiversity.org/wiki/The_necessities_in_Computer_Design
[9] https://en.wikiversity.org/wiki/The_necessities_in_Computer_Architecture
[10] https://en.wikiversity.org/wiki/The_necessities_in_Computer_Organization