Faculty of Engineering

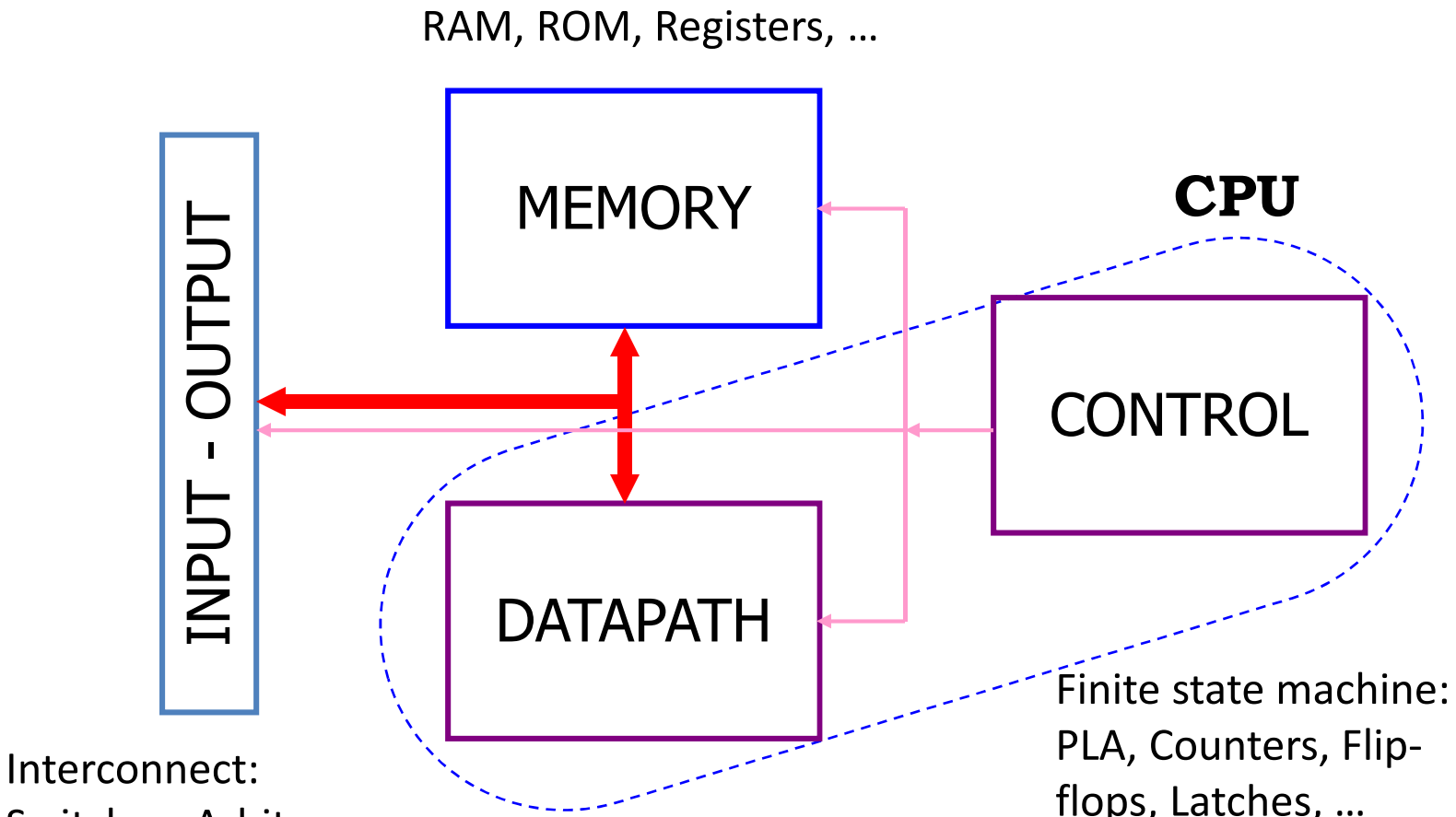
CSE115: Digital Design

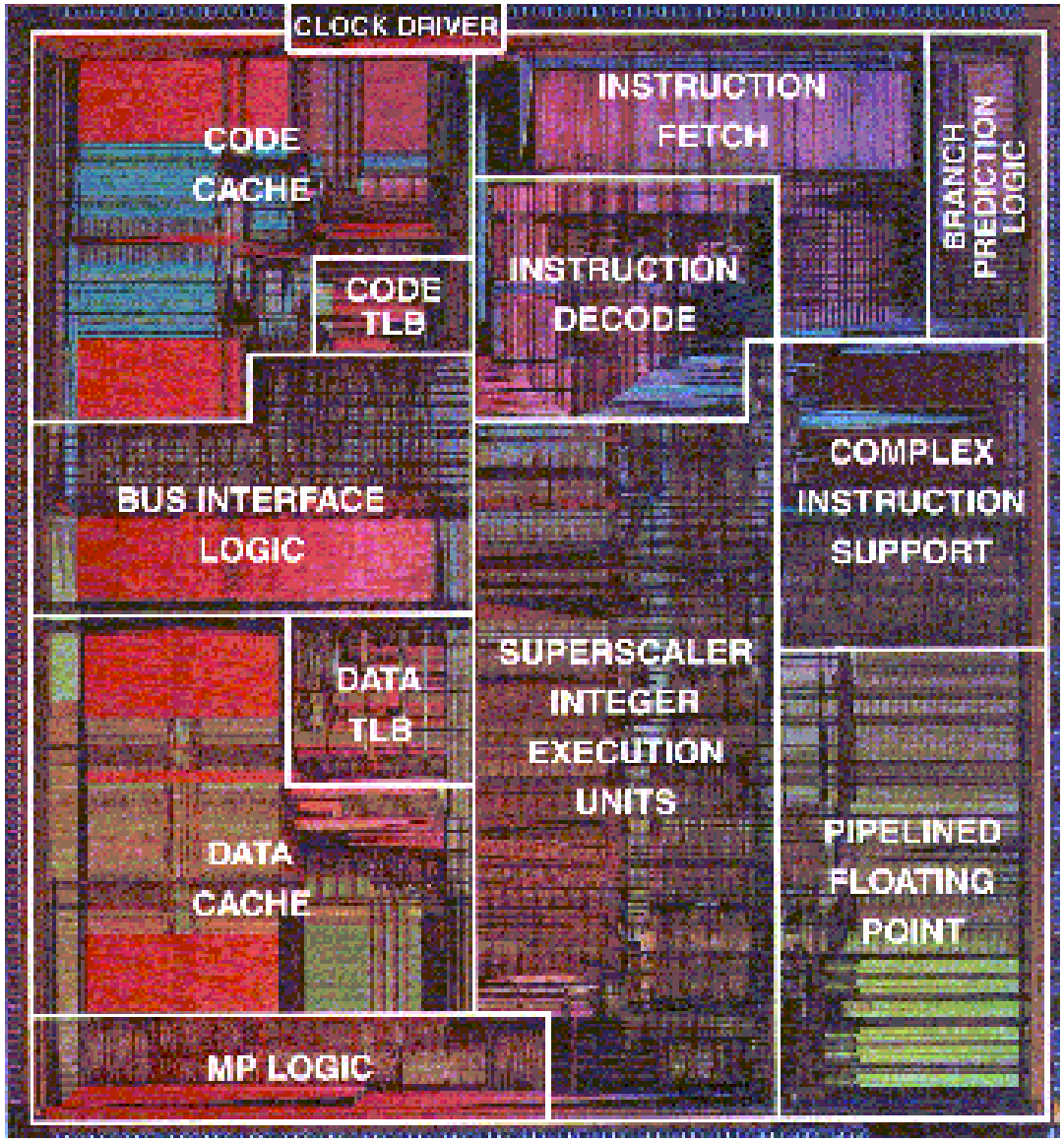
Lecture 23:
Latches & Flip-Flops

Suggested Reading

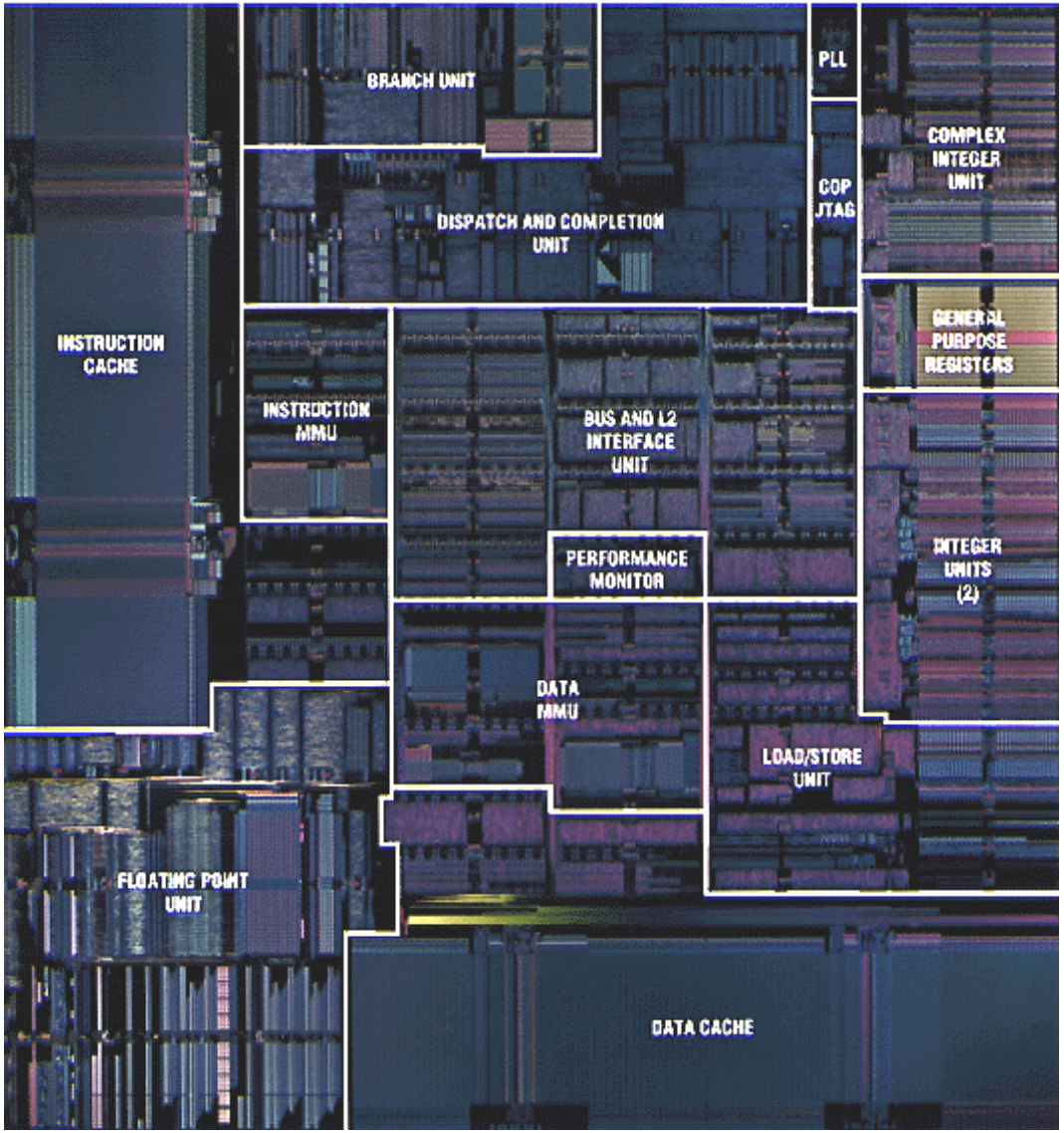
- Sections 7.1-7.2

A Generic Digital Processor




CLOCK DRIVER
CODE CACHE
INSTRUCTION FETCH
BRANCH PREDICTION LOGIC
CODE TLB
INSTRUCTION DECODE
BUS INTERFACE LOGIC
COMPLEX INSTRUCTION SUPPORT
DATA TLB
SUPERSCALER INTEGER EXECUTION UNITS
DATA CACHE
PIPELINED FLOATING POINT
MP LOGIC

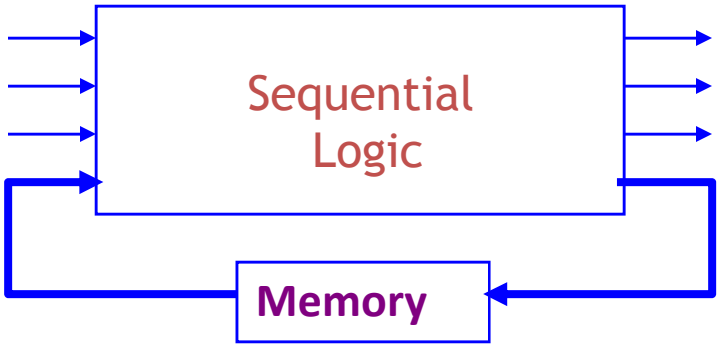
Motorola's PowerPC™ 620 32/64-Bit RISC Microprocessor



Logic Devices

- **Combinational Logic:** Current output depends on current input only
  - ⇒ Gates
  - ⇒ Decoders
  - ⇒ Multiplexers
  - ⇒ ALUs

- **Sequential Logic:** Current output depends on past inputs as well as current input; thus has a memory (**state**).
  - ⇒ Latches and Flip-Flops
  - ⇒ State Machines
  - ⇒ Counters
  - ⇒ Shift Registers



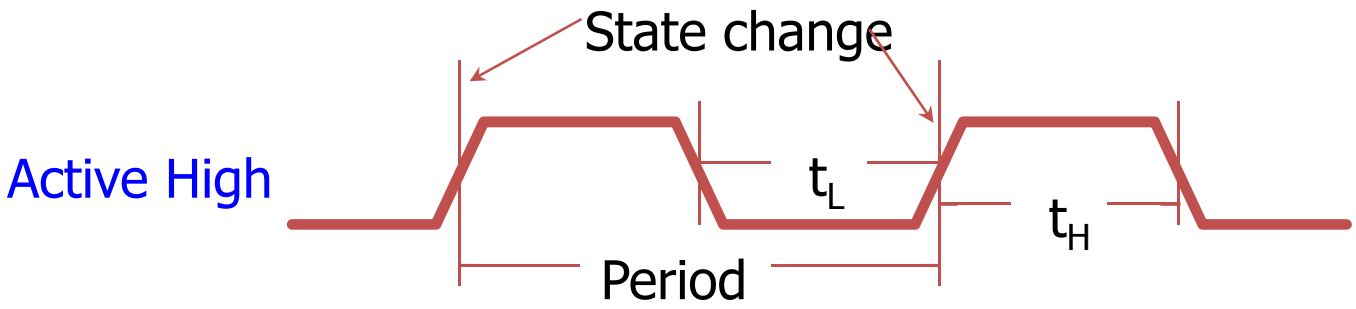
Sequential Logic Definitions

- **STATE**:
  - A collection of state variables whose values contain all the information about the past values necessary to account for future behavior. (e.g. a TV tuner with up/down button)
  - Circuit with *n* binary state variables has 2^n possible states
  - Changes usually synchronized with a system clock

- **Digital sequential logic**
  - Also known as a ***finite state machine (FSM)***.

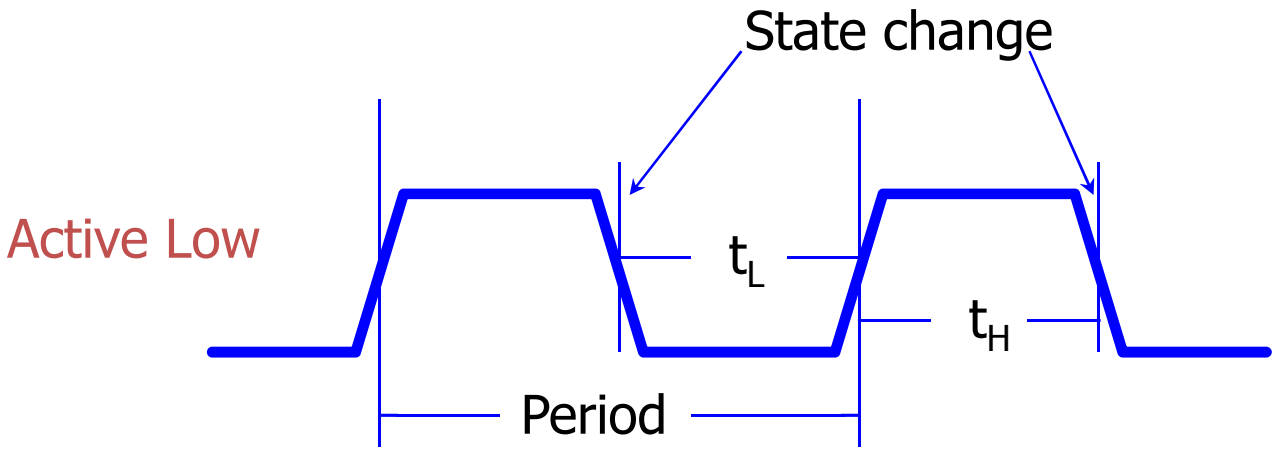
Clock Characteristics

Clock - the master timing element behind the state changes.



Frequency
= 1/Period

Period: time between successive transitions in the same direction



Duty Cycle
= t_H/Period

Duty Cycle: the percentage of time that a clock is at its assertion level. = t_L/Period

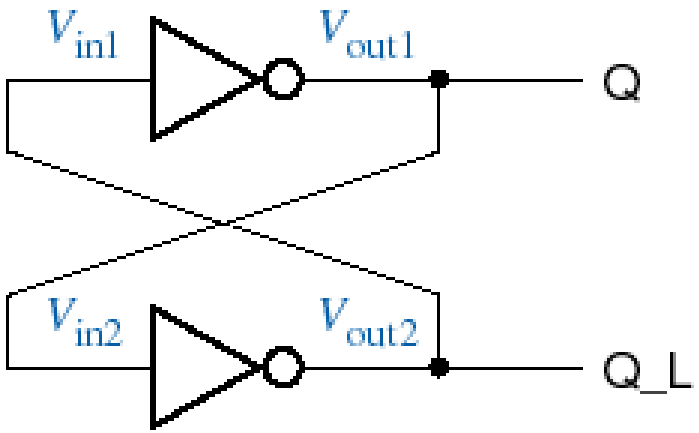
Types of Sequential Logic

- A **Feedback Sequential Circuit** uses gates with feedback to form memory elements (latches and flip/flops) used in state machines.

- A **Clocked Synchronous State Machine** uses clocked flip-flops to form useful sequential logic functions or application.

Bistable Element

- The simplest possible feedback sequential logic circuit:

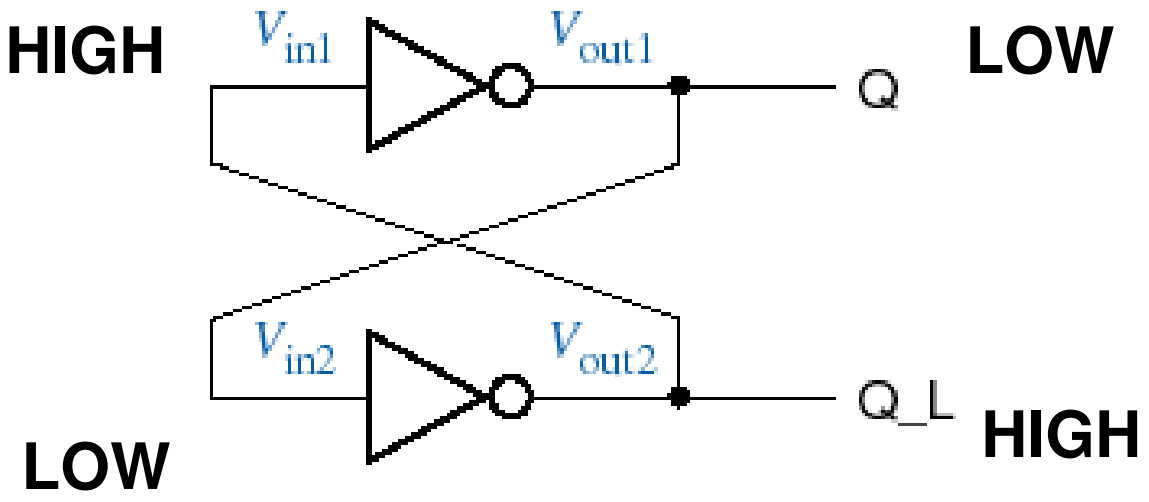


It is **bistable** because it has two stable states:

- **State 1:** If Q ($Q=V_{out1}=V_{in2}$) is high, the bottom inverter output ($/Q = V_{out2}=V_{in1}$) is low, which keeps the top inverter output Q high.
- **State 2:** If Q is low, the bottom inverter output /Q is high, which keeps the top inverter output Q low.

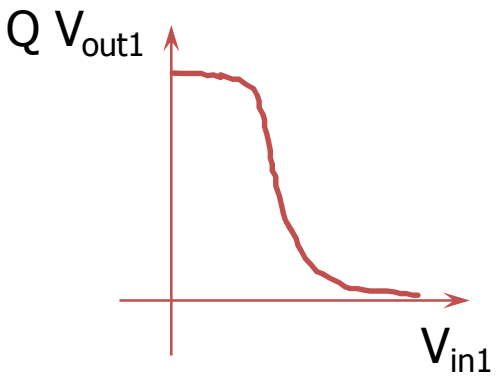
Bistable Element

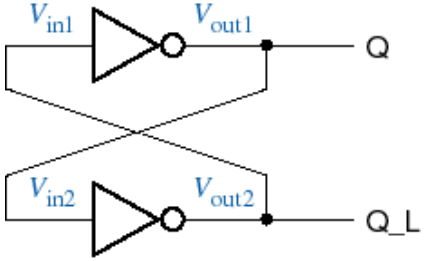
- The simplest sequential circuit
- Two states
  - One state variable, Q



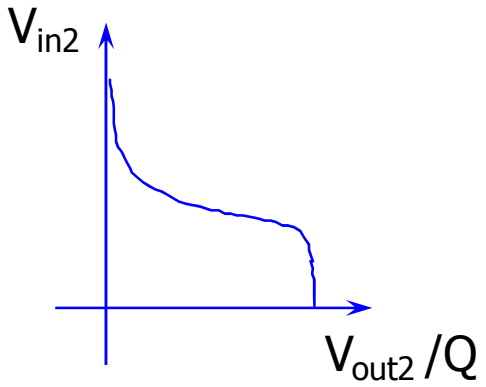
Analog Analysis of Bistable

Top Inverter alone

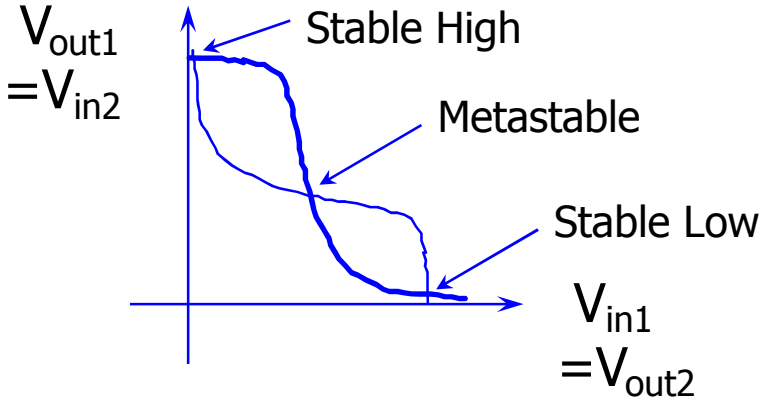




Bottom Inverter alone



Complete Bistable



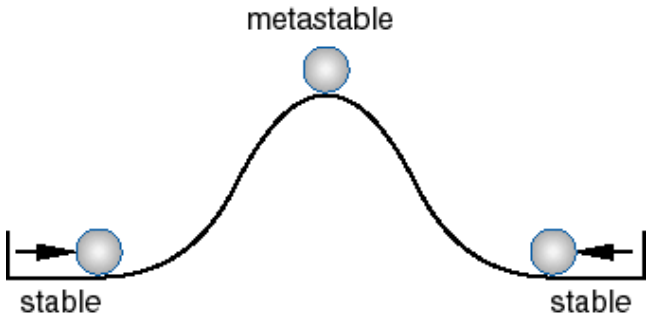
Bistable and Metastability

There are not two stable states, but 3 states (**a problem!**)

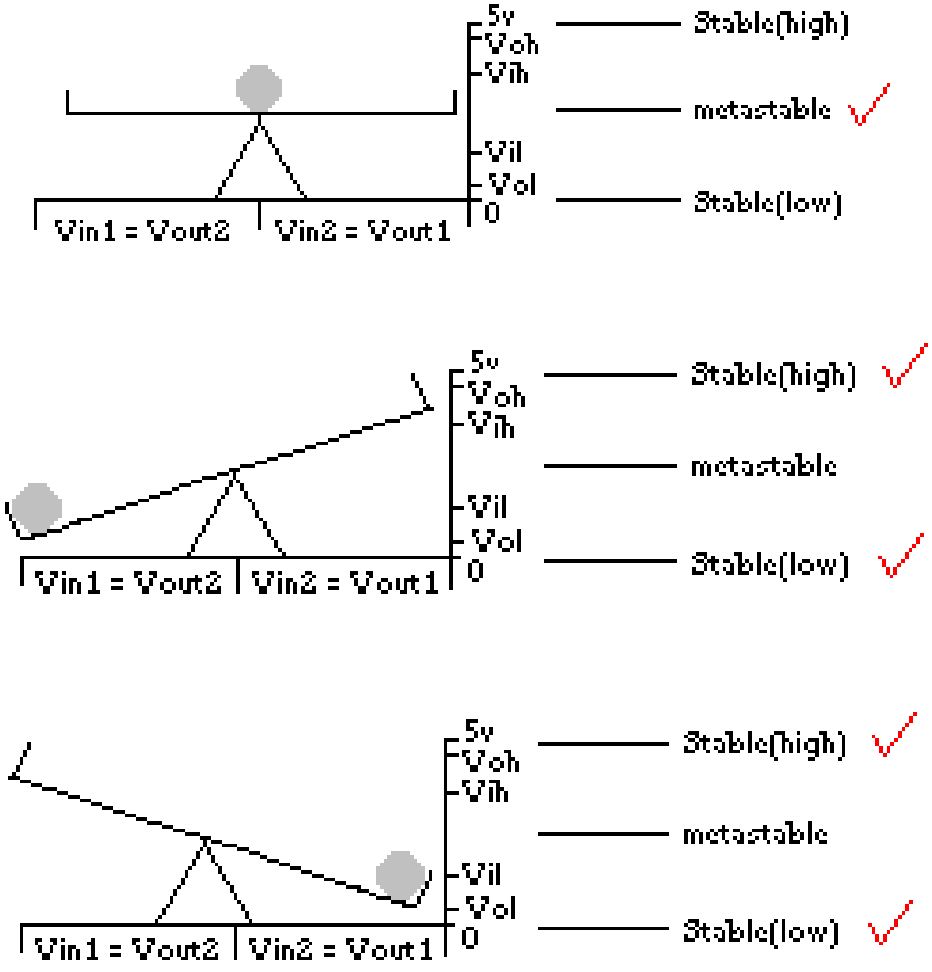
Metastable point occurs when both outputs are halfway between high and low

- not a valid logic level!!!

Could last forever, but noise pushes towards a stable state.



Metastability



Latches and Flip-flops

- Common feedback sequential circuits

- **Latch**
  - Single-bit storage (memory)
  - Changes state at **any time** due to input change

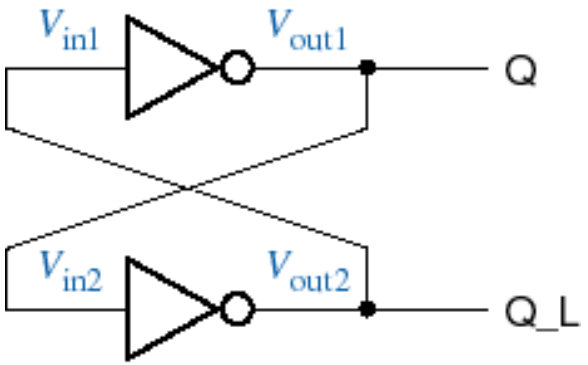
- **Flip-flop**
  - Also single-bit storage
  - Changes state ONLY when a **clock** edge or pulse is applied

Types of Latches and Flip-flops

- **Latches**
  - S-R Latch
  - /S-/R Latch
  - S-R Latch with Enable
  - D Latch

- **Flip-flops**
  - Edge-Triggered D Flip-Flop
  - Master/Slave S-R Flip-Flop
  - Master/Slave J-K Flip-Flop
  - Edge-Triggered J-K Flip-Flop
  - T Flip-Flop

Back to the Bistable….

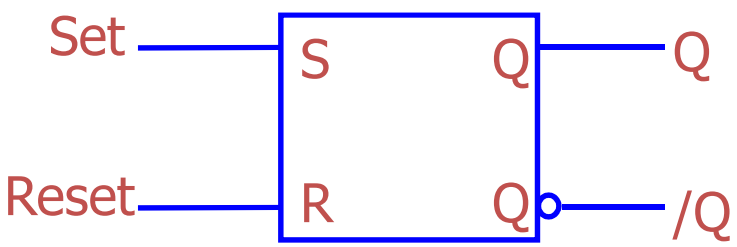


- How to control it?
  - Screwdriver
  - Control inputs

- **S-R latch (set-reset)**

S-R Latch

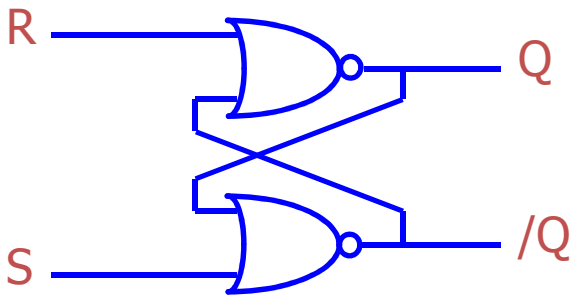
Symbol



Function Table

	S	R	Q	/Q
Hold	0	0	Last Q	Last /Q
Reset	0	1	0	1
Set	1	0	1	0
ILLEGAL	1	1	0	0

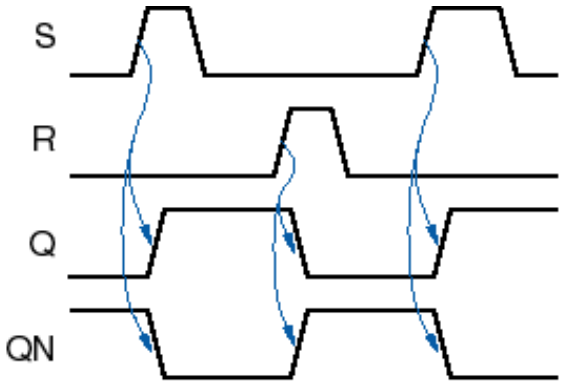
Schematic

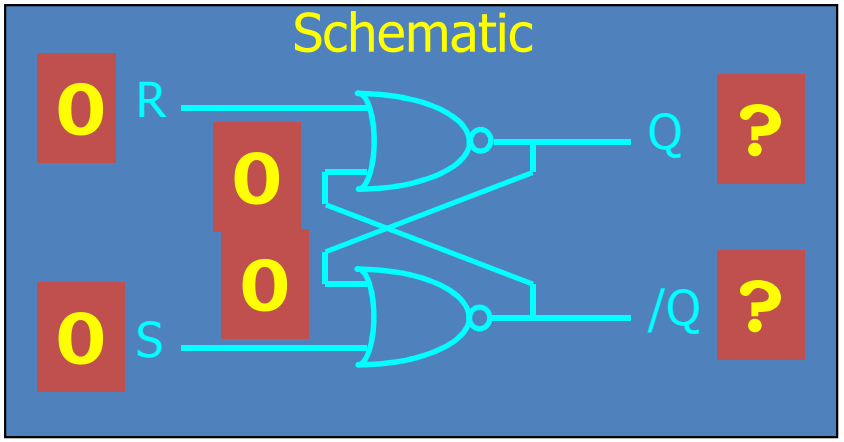


Consider:

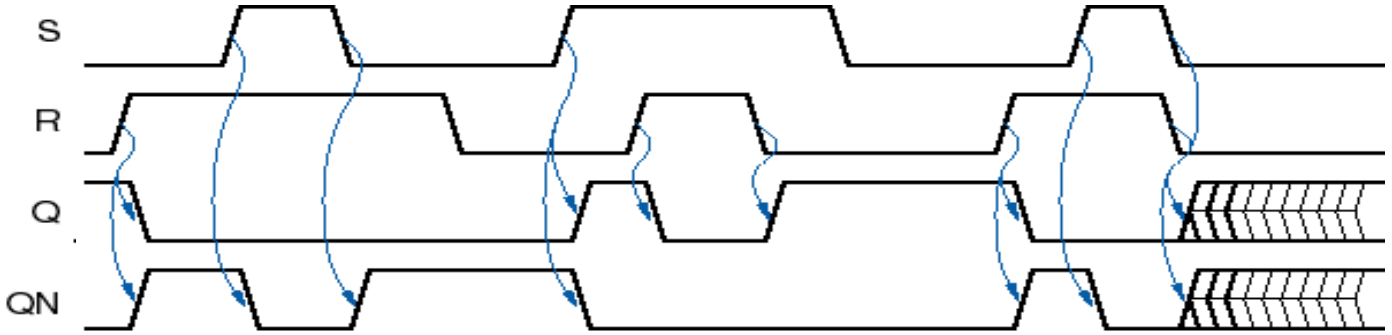
1. Timing Diagram
2. Propagation delay
3. Minimum pulse width
4. Oscillation

S-R latch Operation





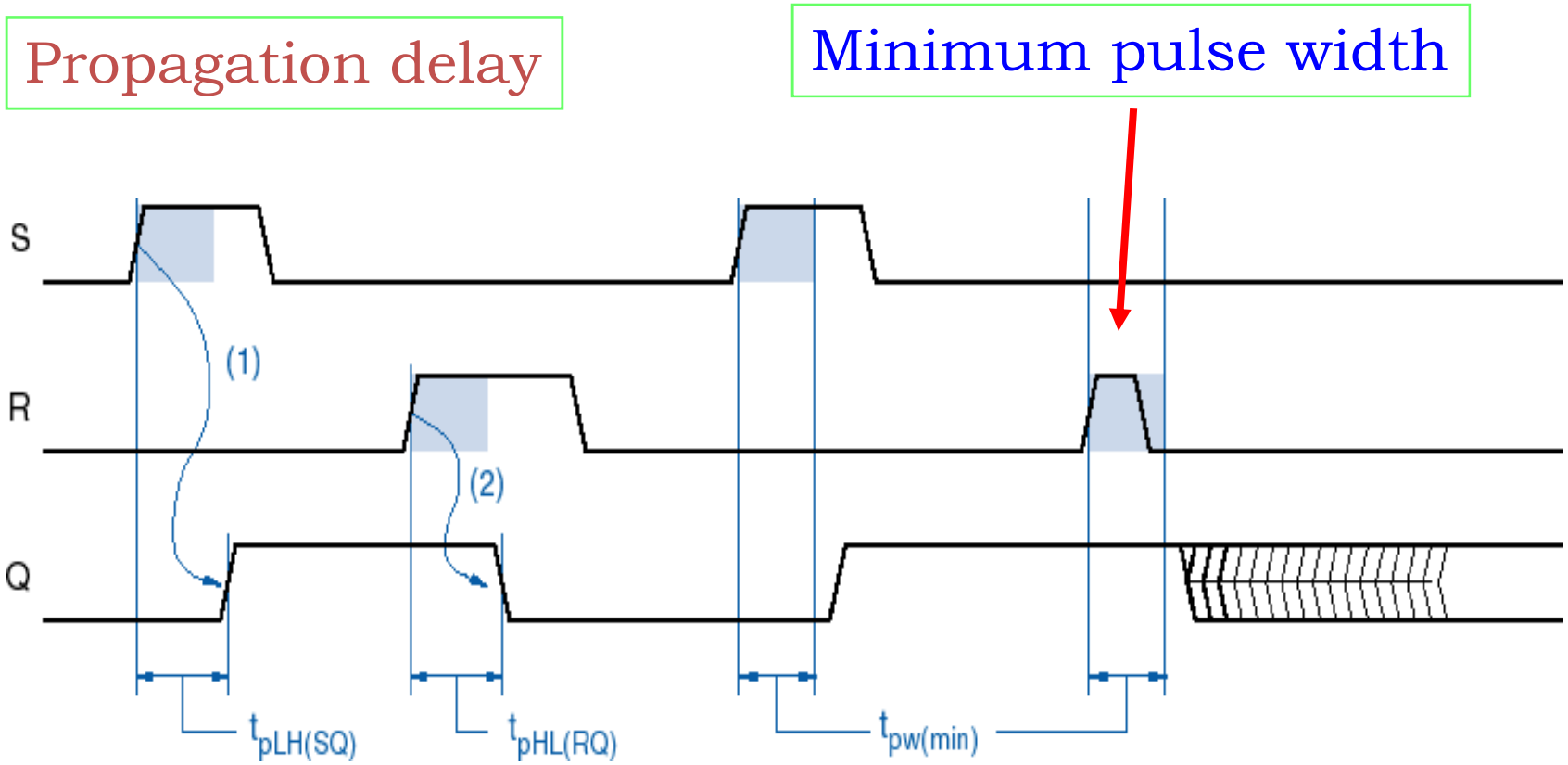
Metastability is possible
if S and R are negated simultaneously.



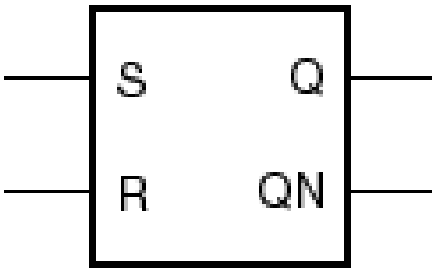
S-R latch Timing Parameters

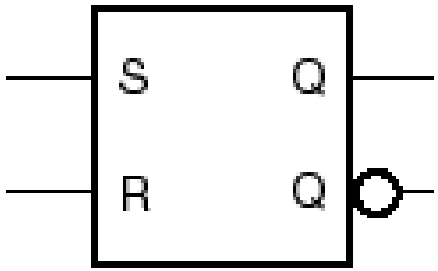
Propagation delay

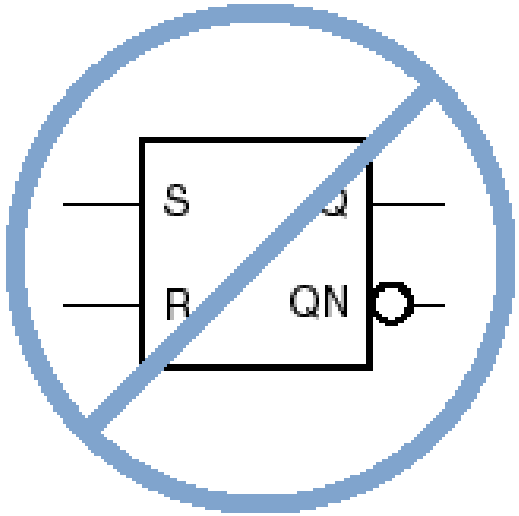
Minimum pulse width



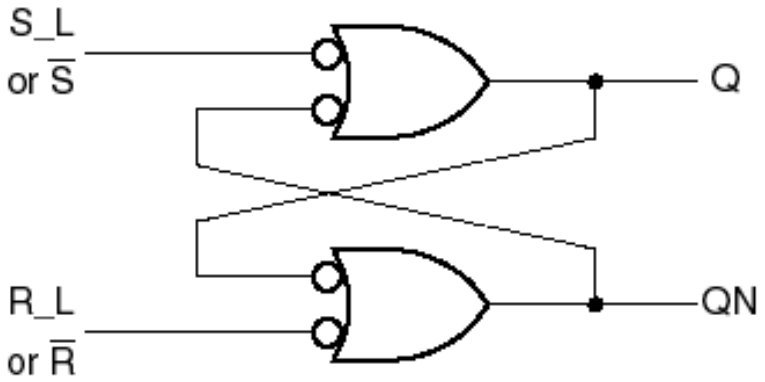
S-R latch Symbols

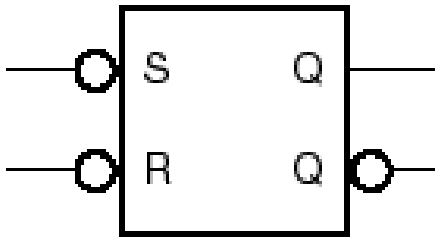






S-R latch using NAND gates:/S-/R Latch



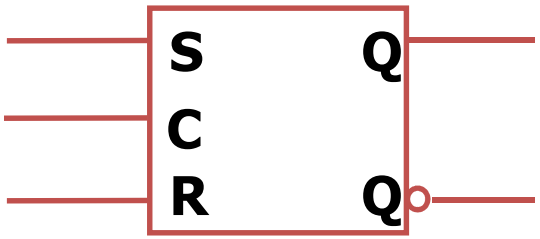


S_L	R_L	Q	QN
0	0	1	1
0	1	1	0
1	0	0	1
1	1	last Q	last QN

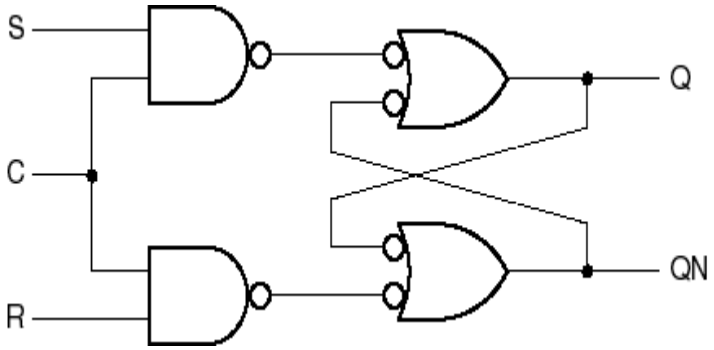
Function Table

	/S	/R	Q	/Q
Illegal	0	0	1	1
Set	0	1	1	0
Reset	1	0	0	1
Hold	1	1	Last Q	Last /Q

S-R Latch with Enable



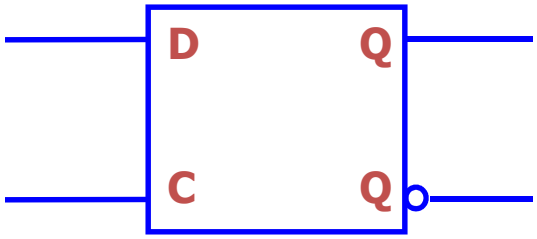
/S	/R	C	Q	/Q
0	0	1	Last Q	Last /Q
0	1	1	0	1
1	0	1	1	0
1	1	1	1	1
x	x	0	Last Q	Last /Q



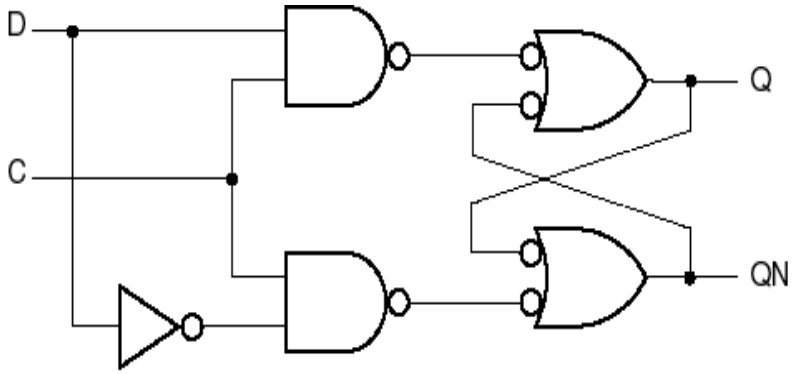
Only sensitive to S and R when enabled (C=1)

Same oscillation problem

D Latch



C	D	Q	QN
1	0	0	1
1	1	1	0
0	x	last Q	last QN



Store a data bit, not set/reset

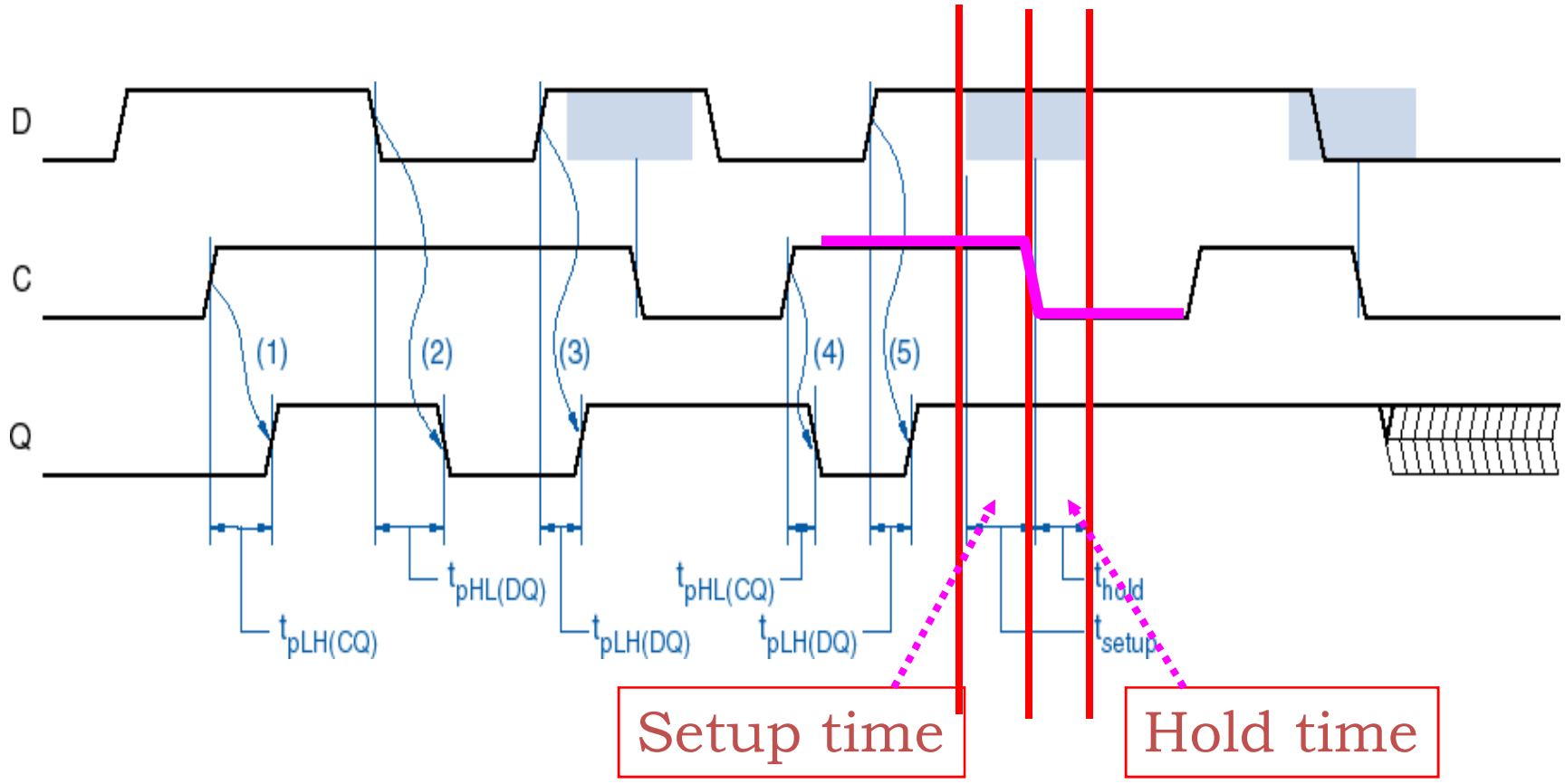
'Transparent latch'

Setup and Hold time

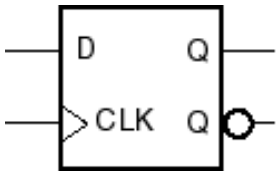
D-latch Timing Parameters

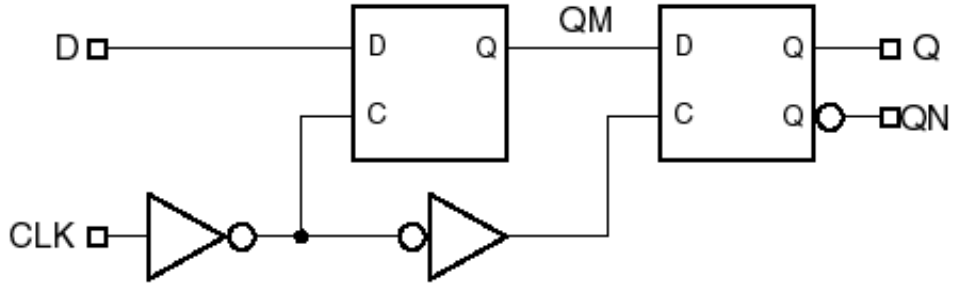
Propagation delay
from C or D

D should not change during:
$t_{setup\ (before\ C\ edge)} + t_{hold\ (after\ C\ edge)}$

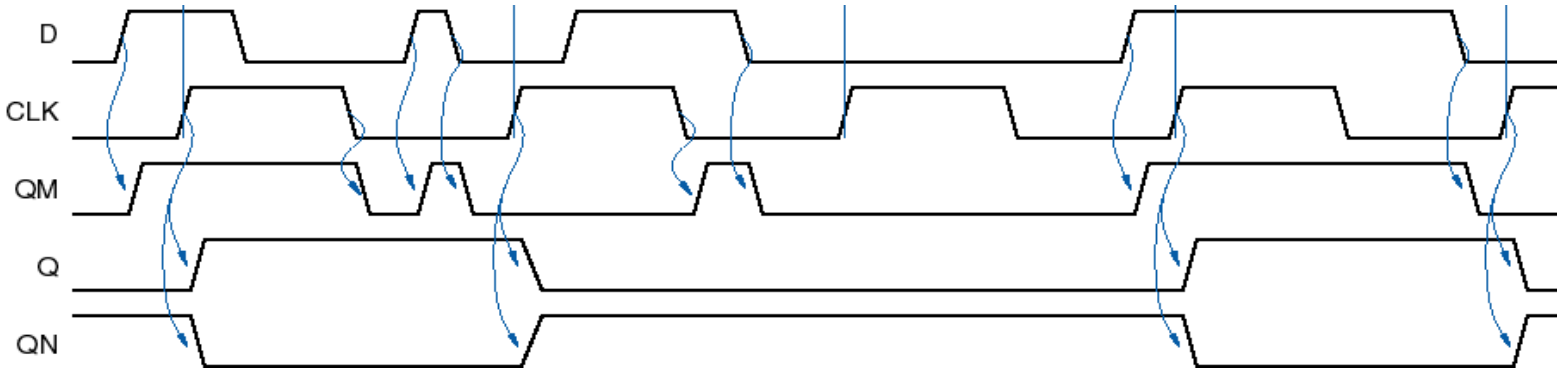


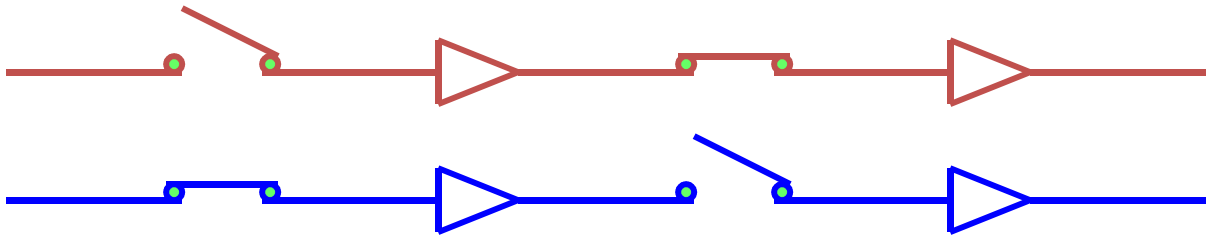
Positive-Edge-Triggered D Flip-Flop





D	CLK	Q	QN
0	↑	0	1
1	↑	1	0
x	0	last Q	last QN
x	1	last Q	last QN

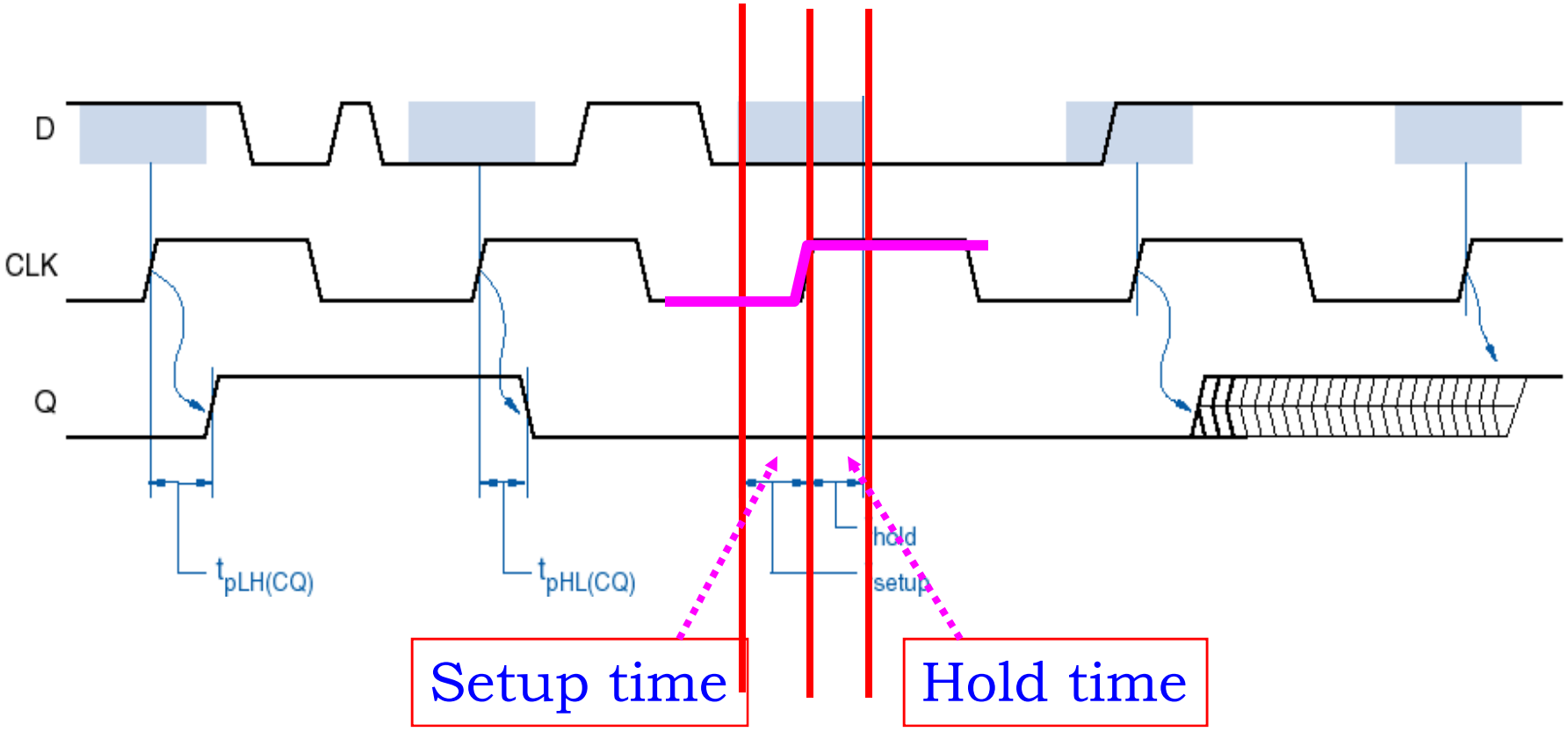




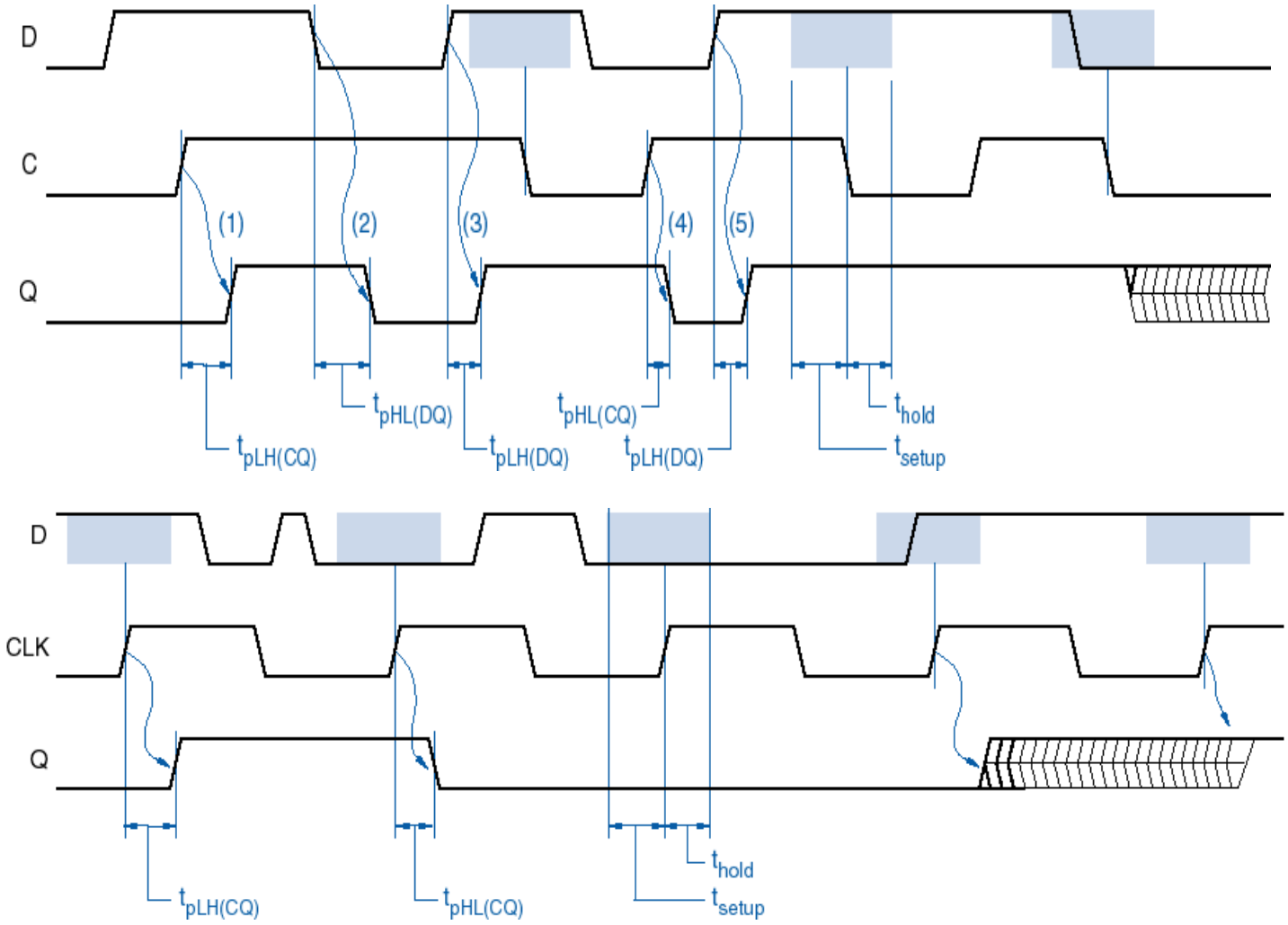
D Flip-Flop Timing Parameters

Propagation delay
from CLK

D should not change during:
$t_{setup\ (before\ C\ edge)} + t_{hold\ (after\ C\ edge)}$

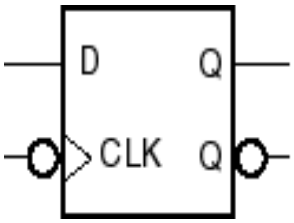


D flip-flop Versus Latch

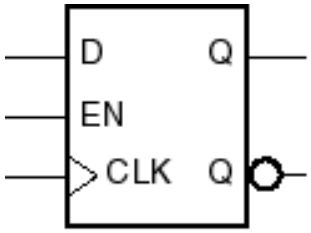


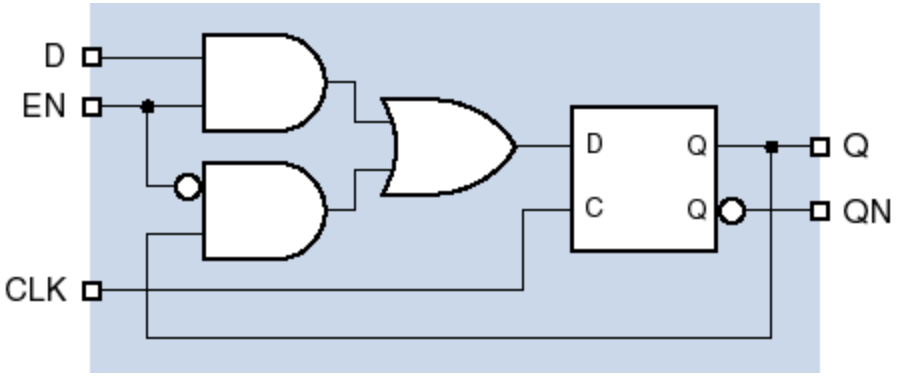
Other D flip-flop Variations

Negative-edge triggered

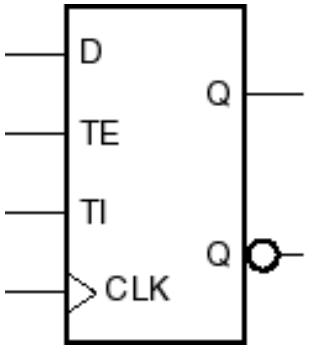


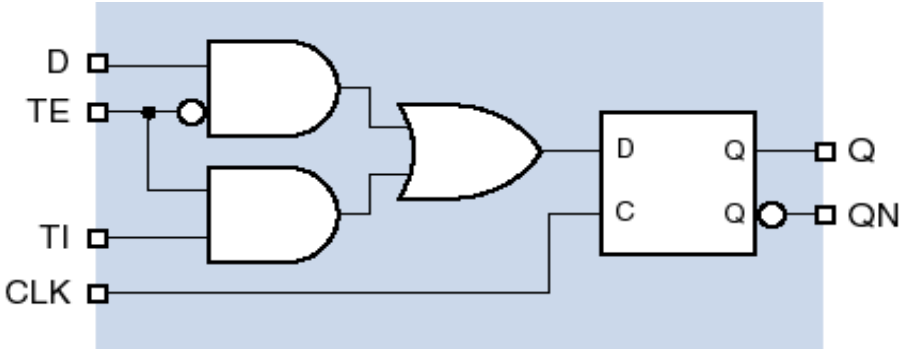
Clock Enable



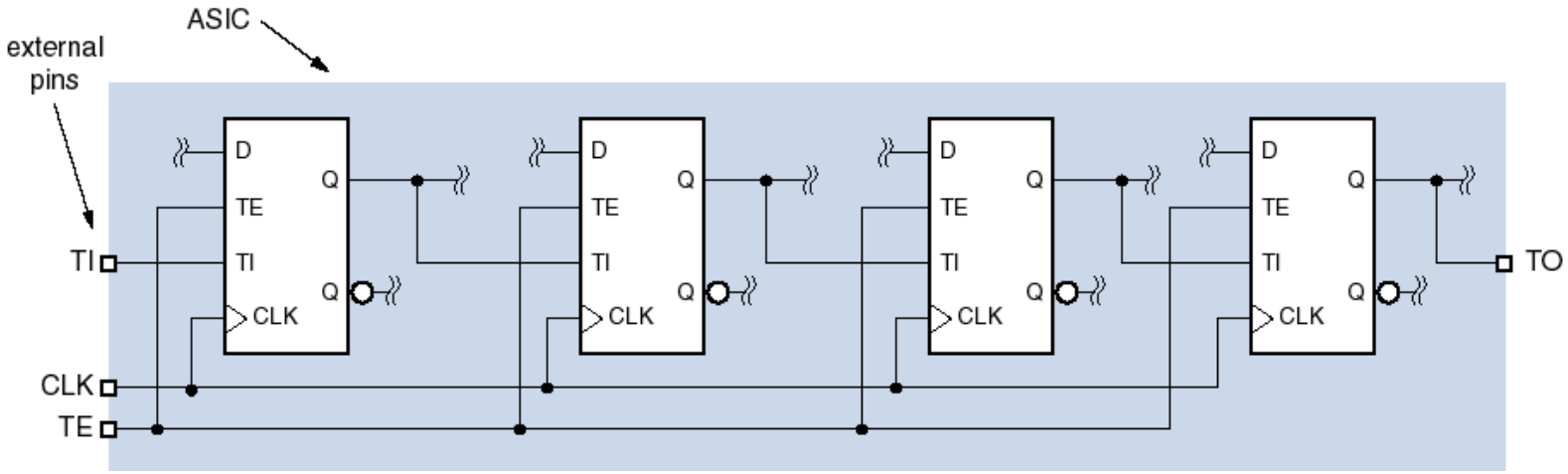


Scan





Scan Flip-Flops -- for testing



TE = 0 → normal operation

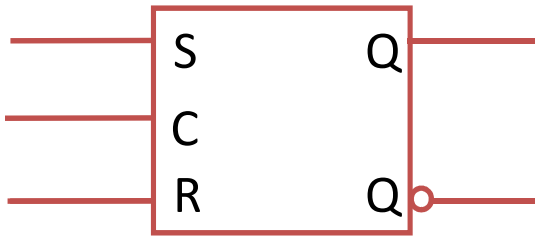
TE = 1 → test operation

- All of the flip-flops are hooked together in a daisy chain from external test input TI.
- Load up ("scan in") a test pattern, do one normal operation, shift out ("scan out") result on TO.

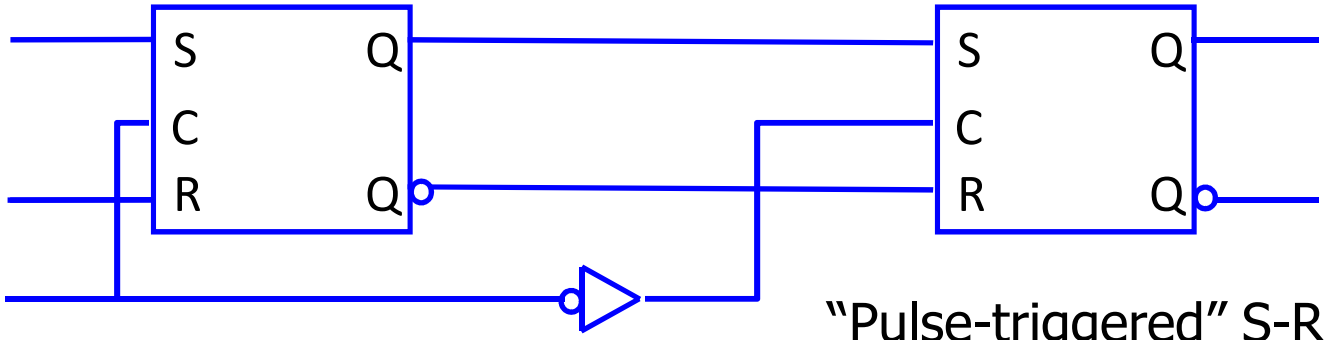
Asynchronous Inputs

- Most flip-flops have two asynchronous inputs
- **Preset** and **Reset** (or Clear)
- Directly set or reset the /S-/R latches
- Operate independent of clock
- USE asynchronous inputs for logic functions ONLY for system INITIALIZATION to a known state

Master/Slave S-R Flip-Flop

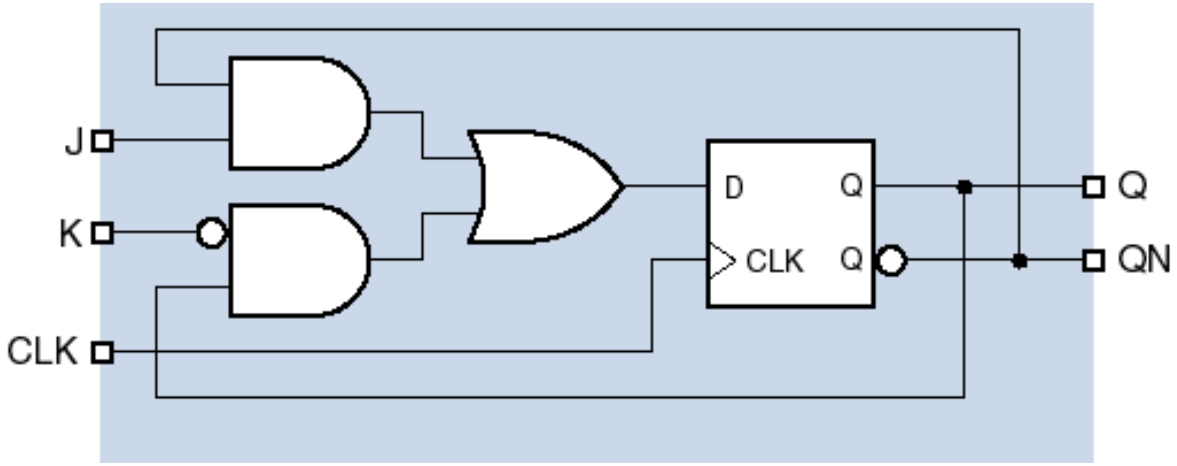


S	R	C	Q	/Q
x	x	0	Last Q	Last /Q
0	0	⊓	Last Q	Last /Q
0	1	⊓	0	1
1	0	⊓	1	0
1	1	⊓	Undef	undef

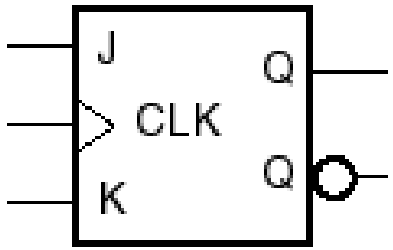


"Pulse-triggered" S-R flip-flop
Pulse-catching behavior

J-K Flip-flops

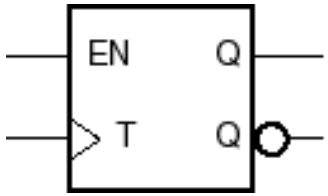


J	K	CLK	Q	QN
x	x	0	last Q	last QN
x	x	1	last Q	last QN
0	0	↓	last Q	last QN
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	last QN	last Q

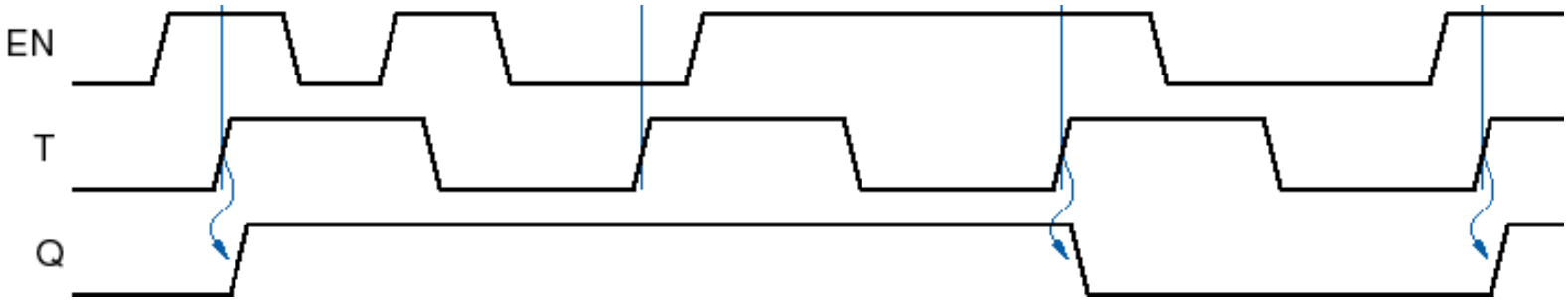


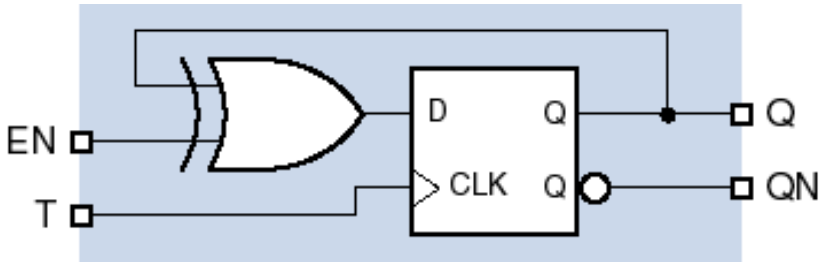
- Not used much anymore

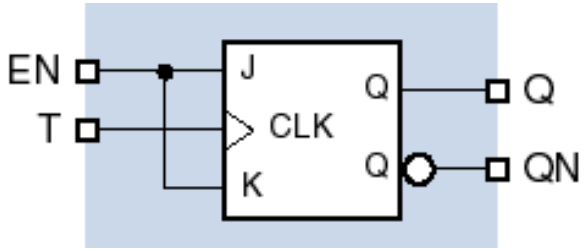
T (toggle) Flip-Flop



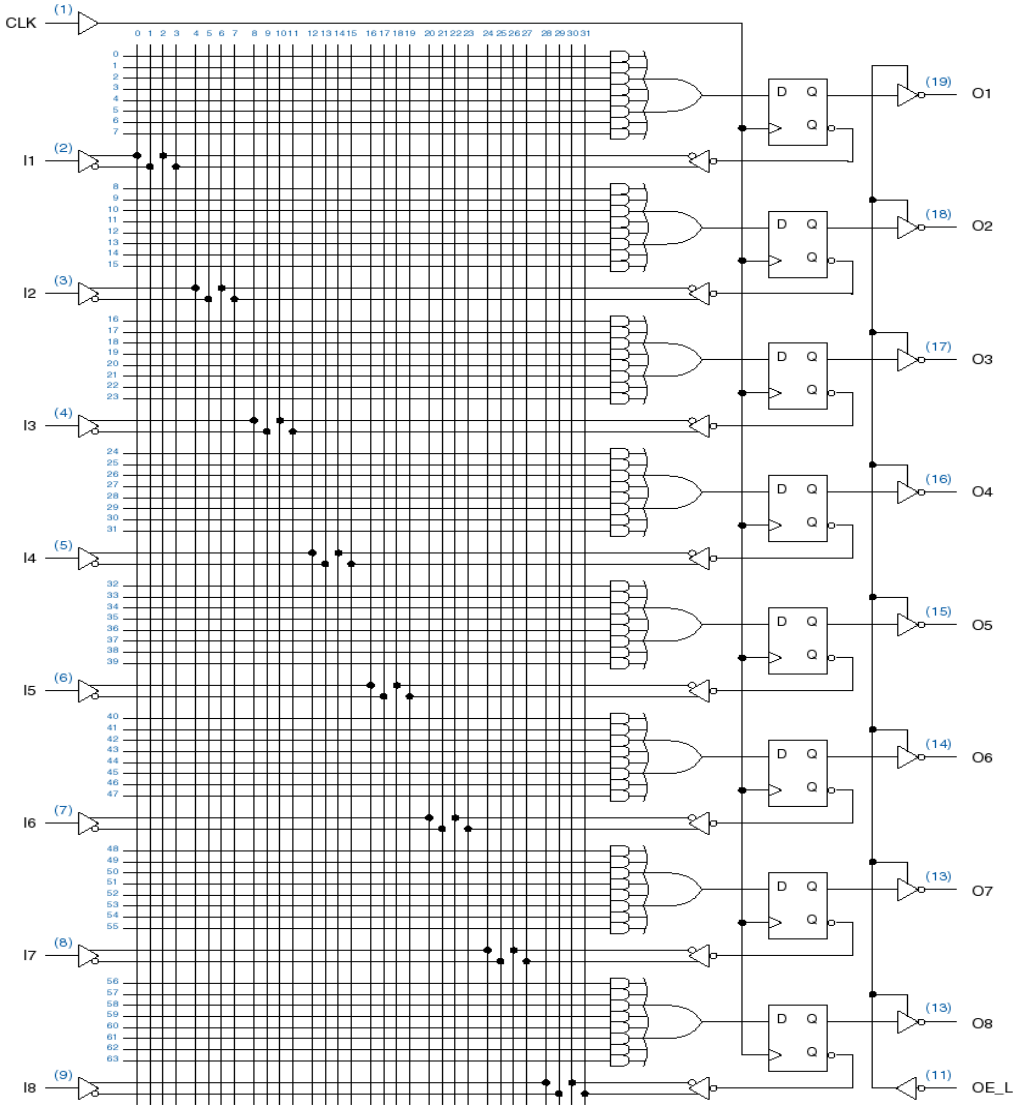
- T flip-flop changes state on every clock tick.
- Important for counters







Sequential PALs



Analysis of State Machines

Characteristic Equations

Describe the next state (QN) of a flip-flop as function of current state (Q) and inputs:

QN = f (Q, inputs) [or Q^* = f (Q, inputs)]

Derived from basic function table for a given flip-flop type

Characteristic Equations

Input	Present state	Next state
D	**Q**	Q*
0	0	0
0	1	0
1	0	1
1	1	1

$Q^* = D$

Input		Present state	Next state
S	**R**	**Q**	Q*
0	0	0	0
0	0	1	1
0	1	x	0
1	0	x	1
1	1	x	x

$Q^* = S + R' Q$

Characteristic Equations

J	K	Q	Q*	
0	0	0	0	hold
0	0	1	1	
0	1	0	0	reset
0	1	1	0	
1	0	0	1	set
1	0	1	1	
1	1	0	1	toggle
1	1	1	0	

$Q^* = J\,Q' + K'\,Q$

EN	Q	Q^*
0	0	0
0	1	1
1	0	1
1	1	0

$Q^* = EN\,Q' + EN'\,Q$

Characteristic Equations Summary

Device Type	Characteristic Eq.
• S-R latch	$Q^* = S + R' Q$
• D latch	$Q^* = D$
• Edge-triggered D flip-flop	$Q^* = D$
• Master/slave S-R flip-flop	$Q^* = S + R' Q$
• Master/slave J-K flip-flop	$Q^* = J Q' + K' Q$
• Edge-triggered J-K flip-flop	$Q^* = J Q' + K' Q$
• T flip-flop	$Q^* = Q'$
• T flip-flop with enable	$Q^* = EN Q' + EN' Q$