Principles Of
Digital Design

Flip-Flops

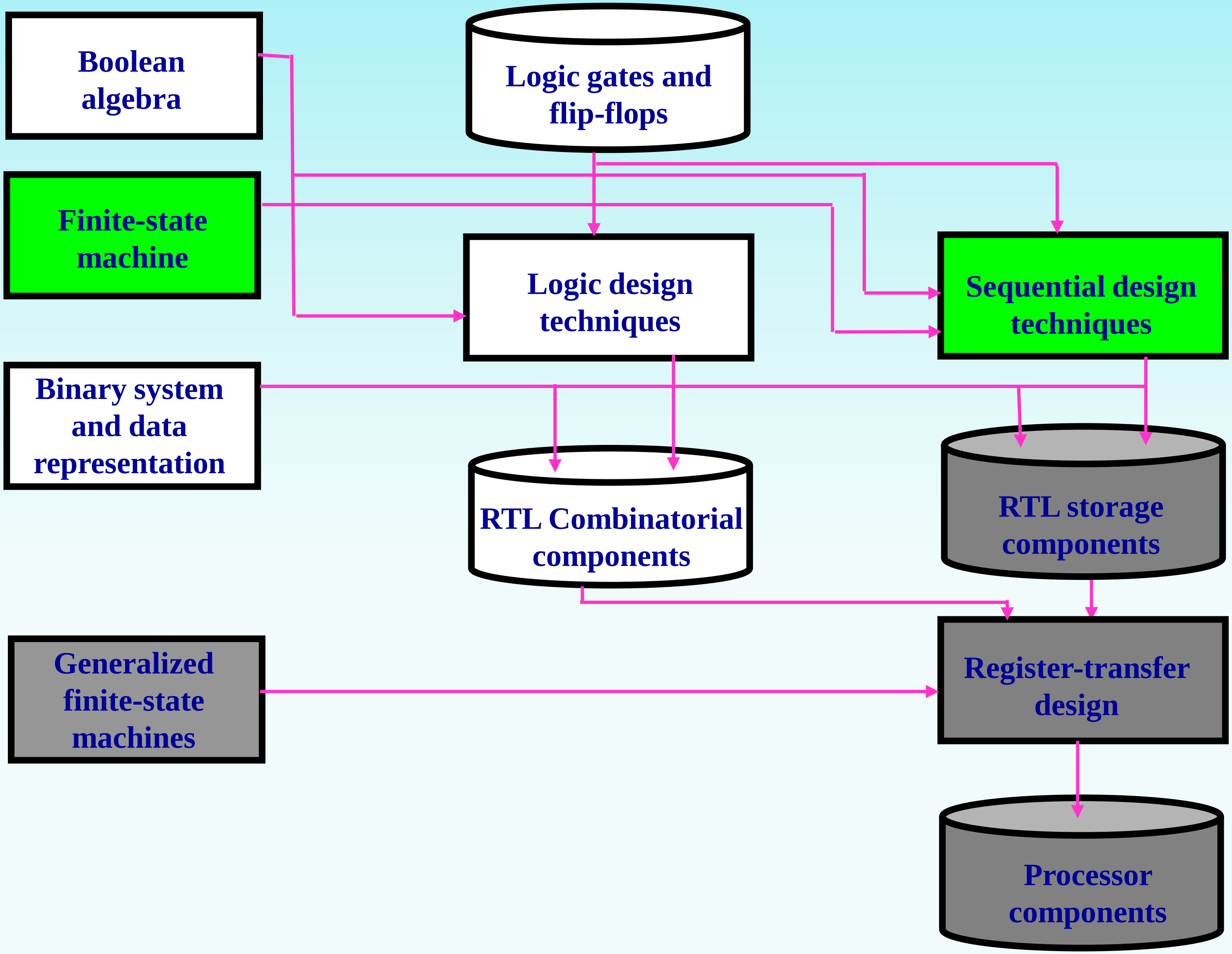
Clocks

Latches

Flip-flops

State diagrams

Topic preview

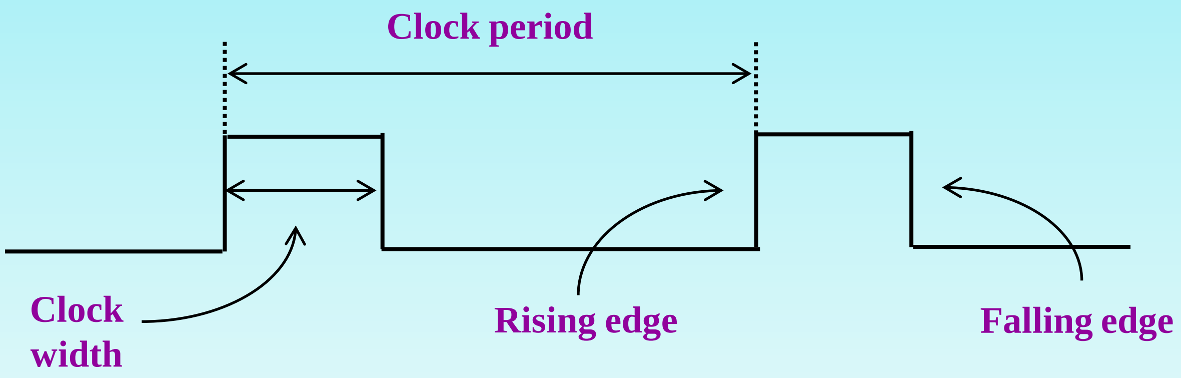

Boolean algebra
Logic gates and flip-flops
Finite-state machine
Logic design techniques
Sequential design techniques
Binary system and data representation
RTL Combinatorial components
RTL storage components
Generalized finite-state machines
Register-transfer design
Processor components

Sequential components

- Sequential components contain memory elements
- The output values of sequential components depend on the input values and the values stored in the memory elements
- The values in the memory elements define the state of sequential components
- Example : Ring counter that starts the answering machine after 4 rings
- Sequential components can be

  (1) asynchronous or (2) synchronous
- Asynchronous sequential components change their state and output values as a response to change in input values
- Synchronous sequential components change their state and output values at fixed points of time defined by the clock signal

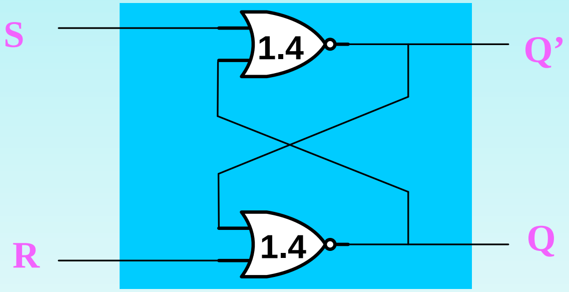
Clock signal



- Clock period (measured in micro, nano seconds) is the time between successive transitions in the same direction
- Clock frequency (measured in MHz, GHz) is the reciprocal of clock period
- Clock width is the time interval during which clock is equal to 1
- Duty cycle is the ratio of the clock width and clock period
- Clock signal is active high if the changes occur at the rising edge or during the clock width
- Clock signal is active low otherwise

SR-latch (NOR implementation)

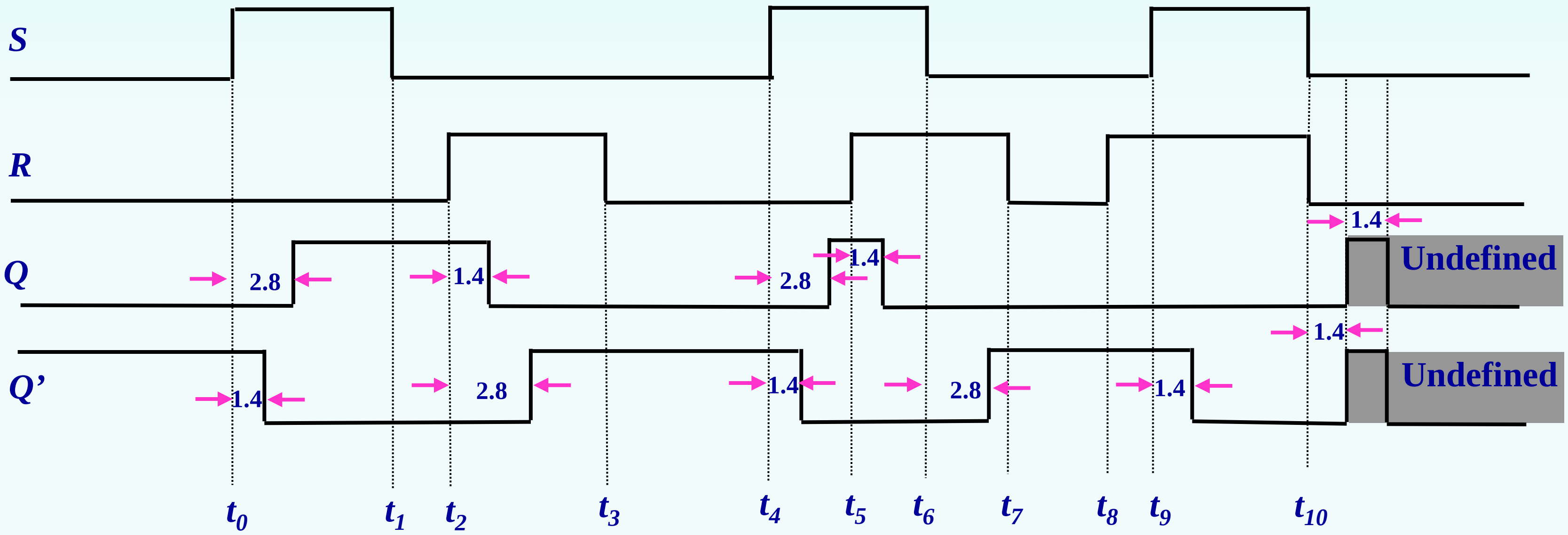
- SR-latch has two states: (1) set state (Q=1) and (2) reset state (Q=0)



Logic schematic

S	*R*	*Q*	*Q (next)*	*Q' (next)*	
0	0	0	0	1	(hold)
0	0	1	1	0	(hold)
0	1	X	0	1	(reset)
1	0	X	1	0	(set)
1	1	X	0	0	(?)

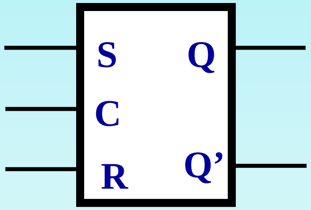
Truth table



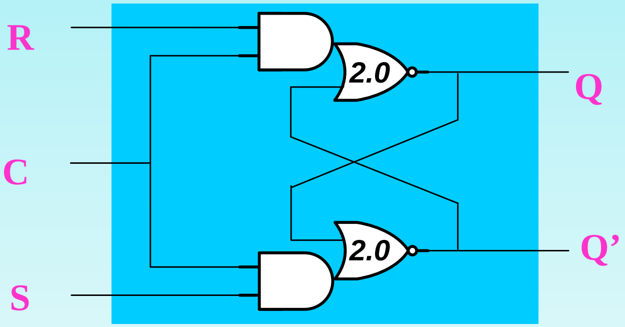
Timing diagram

Gated SR-latch

- Control signal C activates the latch



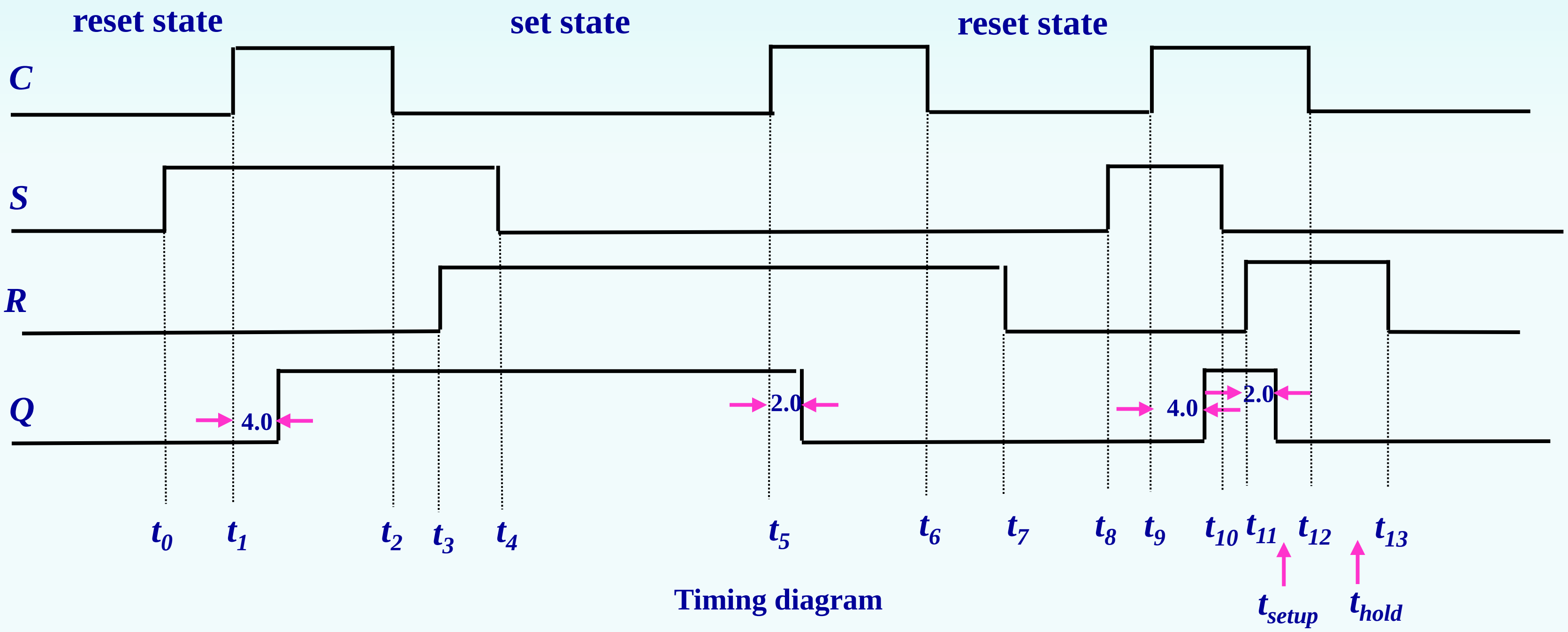
Graphic symbol



Logic schematic

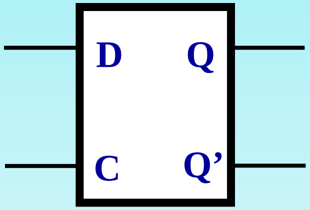
C	*S*	*R*	*Q*	*Q(next)*	
0	X	X	0	0	(inactive)
0	X	X	1	1	(inactive)
1	0	0	0	0	(hold)
1	0	0	1	1	(hold)
1	0	1	X	0	(reset)
1	1	0	X	1	(set)
1	1	1	X	NA	(?)

Truth table

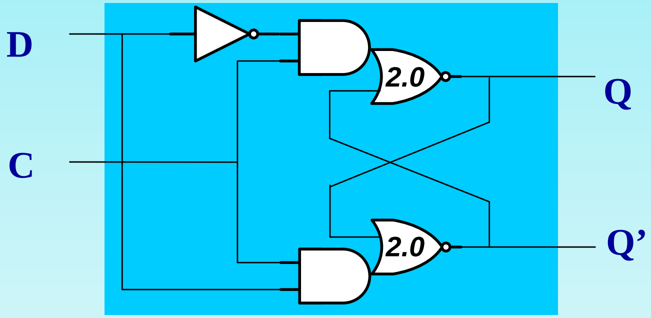


Timing diagram

Gated D-latch



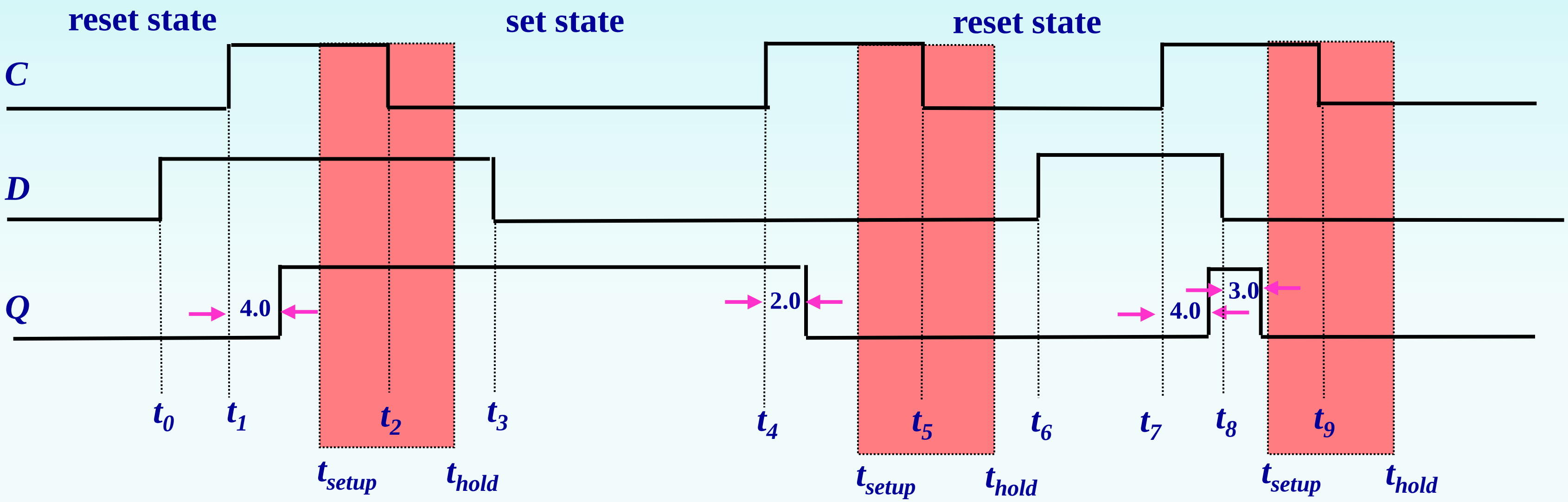
Graphic symbol



Logic schematic

C	*D*	*Q*	*Q(next)*
0	X	0	0
0	X	1	1
1	0	X	0
1	1	X	1

Truth table



Timing diagram

- •Setup time is minimum time inputs must be stable before C ↓
- •Hold time is minimum time inputs must be stable after C ↓
- •Q follows D while C is asserted as long as D satisfies setup and hold time restrictions

Flip Flops

- Latches are level-sensitive since they respond to input changes during clock width.
- Latches are difficult to work with for this reason.
- Flip-Flops respond to input changes only during the change in clock signal.
- They are easy to work with though more expensive than latches.
- Several styles of flip-flops are available.

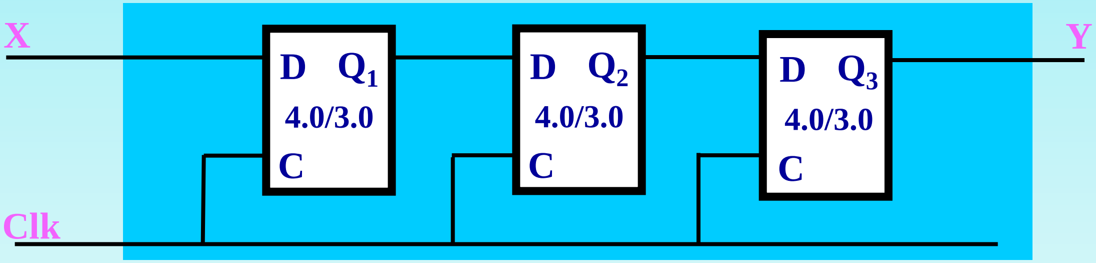
  (1) master-slave (MS)

  (2) edge-triggered (ET)

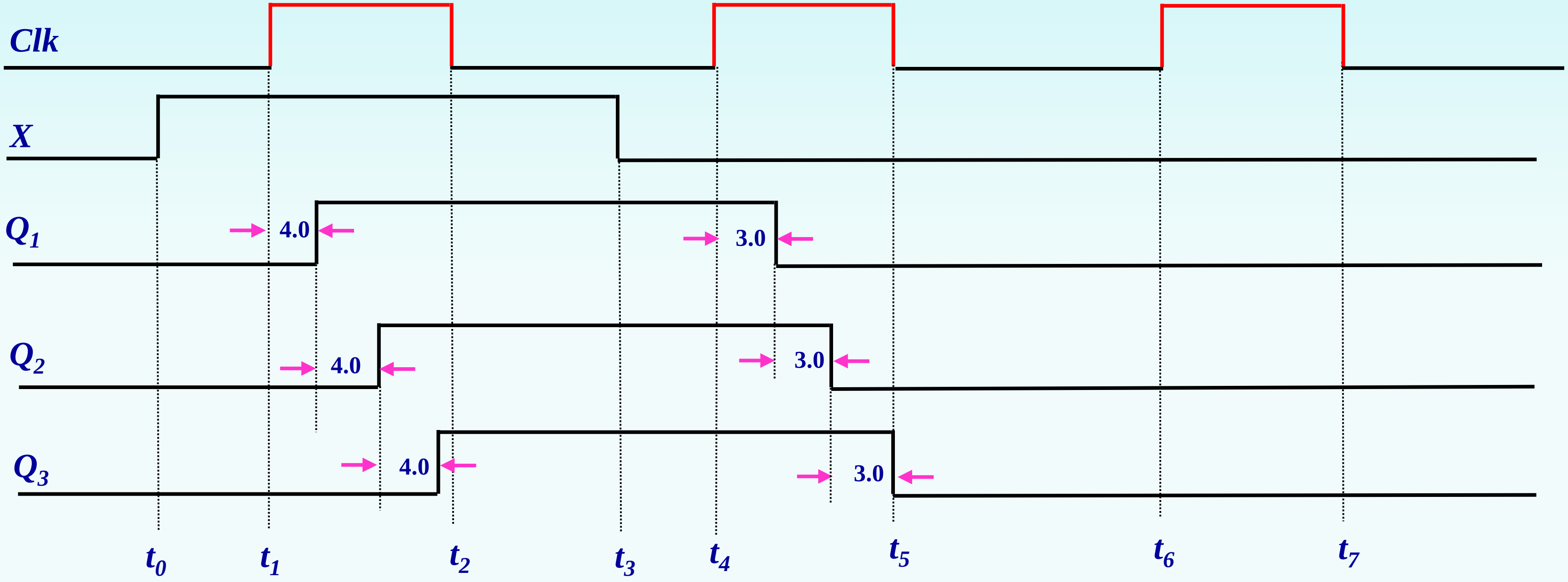
  (3) …

Erroneous shifting with D-latches

- Erroneous operation is possible with level-sensitive latches



Logic schematic

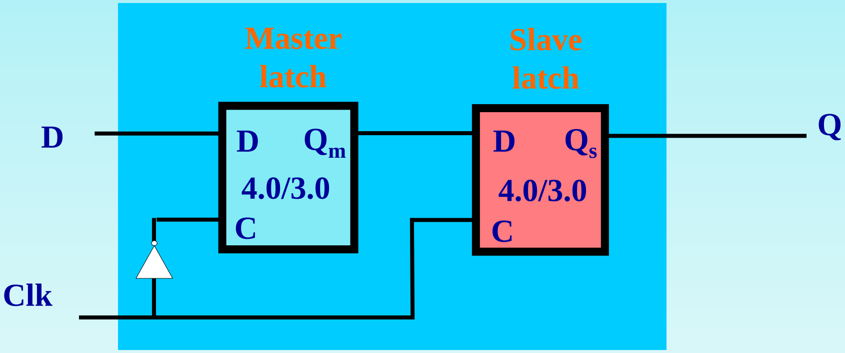


Timing diagram

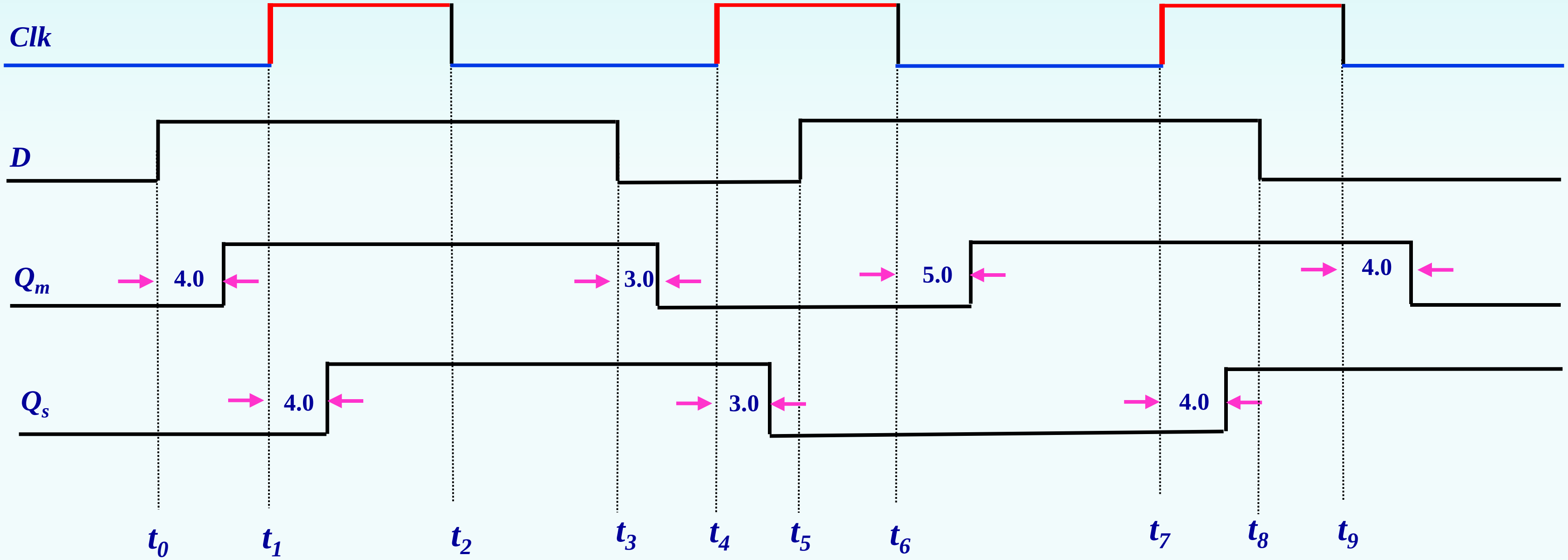
Note: Low-to-high delay is 4.0ns. High-to-low delay is 3.0ns.

Master-slave flip-flop

- In a MS flip-flop D is sampled and stored at the rising edge (low-to-high) of the Clk signal

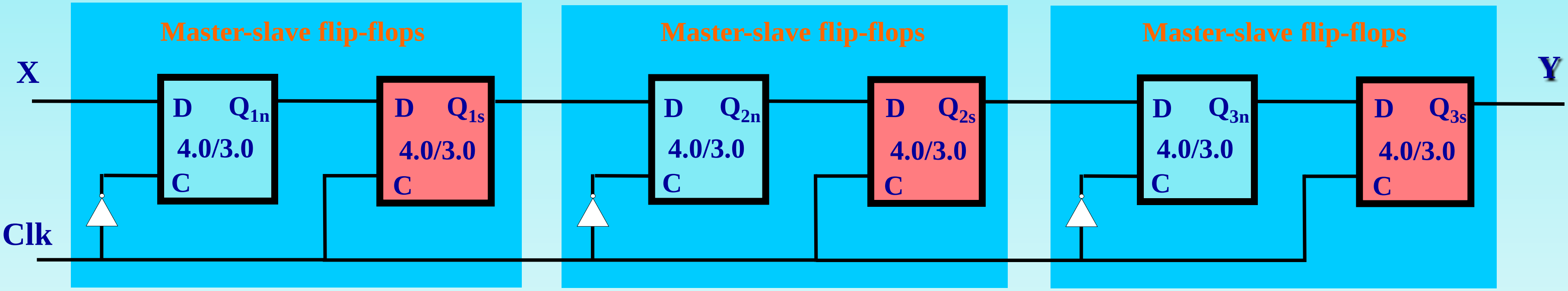


Logic schematic

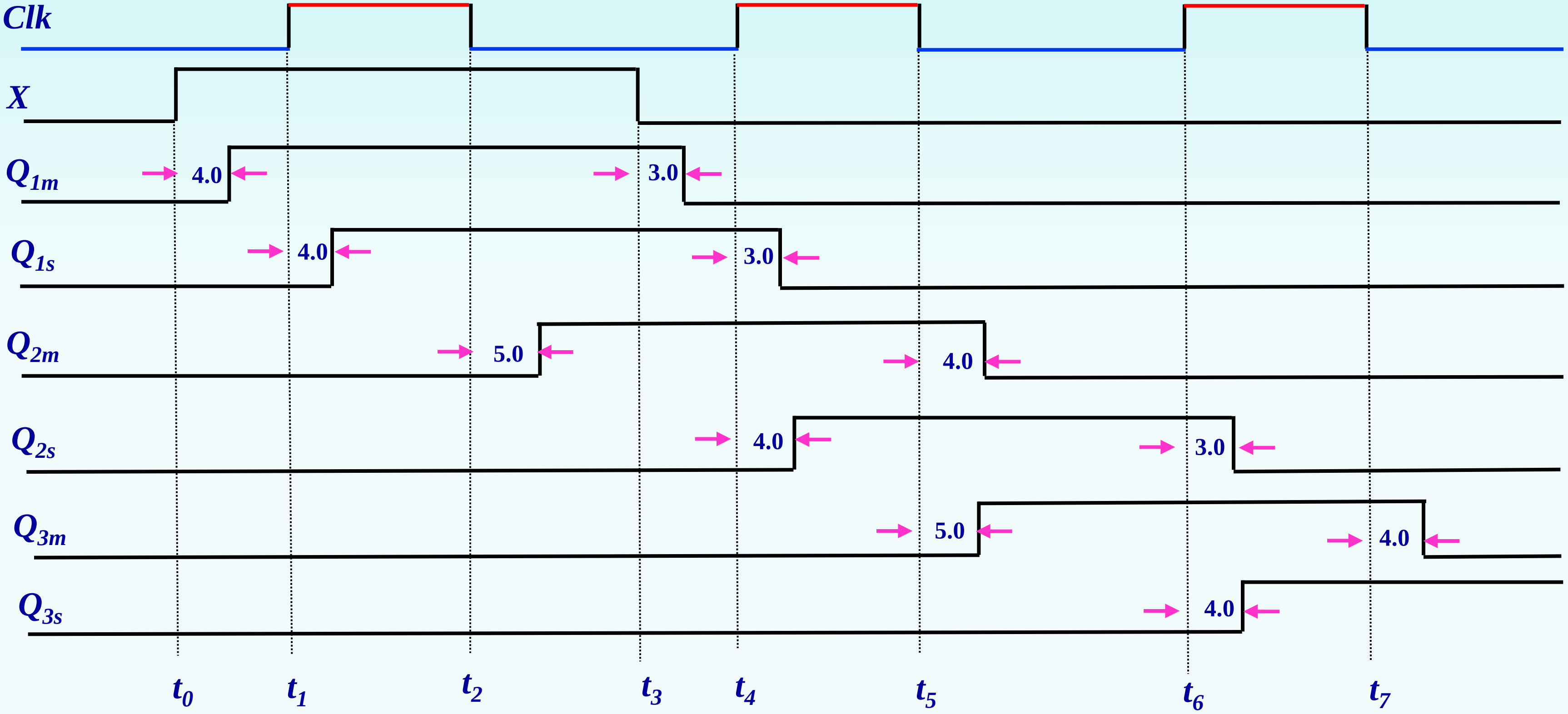


Timing diagram

Shifting with master-slave flip-flops



Logic schematic



Timing diagram

Flip-flop types

Flip-flop name

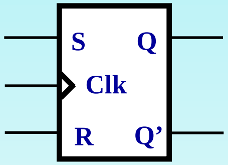
Flip-flop symbol

Characteristic table

Characteristic equation

Excitation table

SR



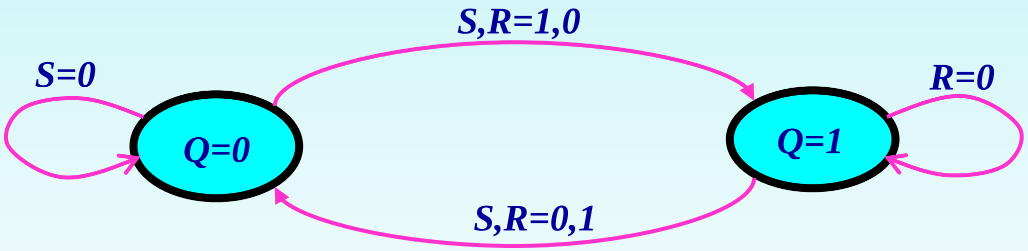
S	*R*	*Q(next)*
0	0	Q
0	1	0
1	0	1
1	1	NA

Q(next)=S+R'Q

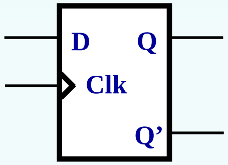
SR=0

Q	*Q(next)*	*S*	*R*
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

State diagram



D

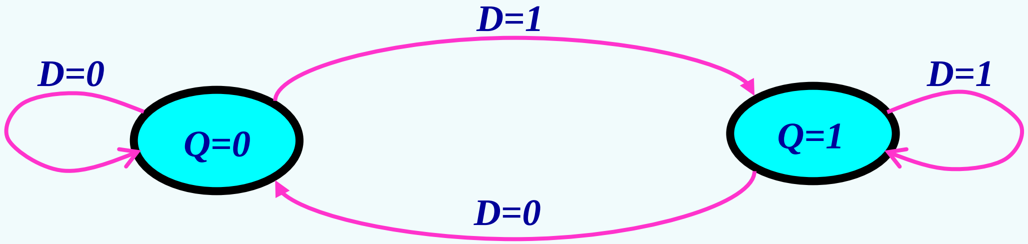


D	*Q(next)*
0	0
1	1

Q(next)=D

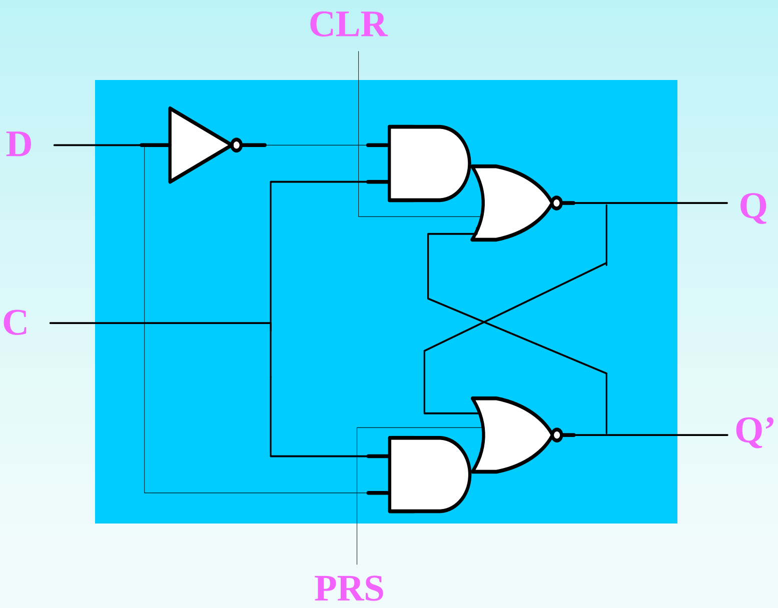
Q	*Q(next)*	*D*
0	0	0
0	1	1
1	0	0
1	1	1

State diagram

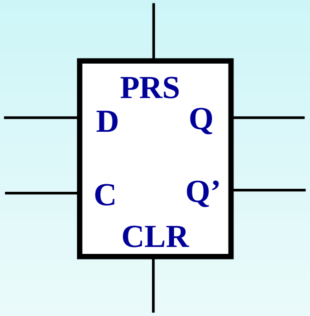


Note: For master-slave flip-flops data inputs must satisfy set-up and hold time constraints.

A latch / flip-flop with asynchronous inputs



D latch



Graphic symbol

Summary

- We introduced memory elements
  - Latches (asynchronous)
  - Flip-flops (synchronous)

- We presented several ways to describe memory elements
  - Characteristic tables
  - Characteristic equations
  - State diagrams
  - Timing diagrams

- We introduced the concept of a state diagram >> FSM