Design of Digital Systems II
Sequential Logic Design Principles (1)

Moslem Amiri, Václav Přenosil

Masaryk University

Resource: "Digital Design: Principles & Practices"

by John F. Wakerly

Introduction

- Logic circuits are classified into two types, "**combinational**" and "**sequential**".

- A **combinational** logic circuit is one whose outputs depend only on its current inputs.

- A **sequential** logic circuit is one whose outputs depend not only on its current inputs, but also on past sequence of inputs, possibly arbitrarily far back in time.

- So it is **inconvenient**, and often impossible, to describe behavior of a sequential circuit by means of a **table** that lists outputs as a function of input sequence that has been received up until current time.

- **State** of a sequential circuit is a collection of **state variables** whose values at any one time contain **all information about past necessary** to account for circuit's future behavior.

- In a digital logic circuit, **state variables are binary values**, corresponding to certain logic signals in circuit.

- A circuit with n binary state variables has 2^n possible states.

Introduction

- State changes of most sequential circuits occur at times specified by a free-running **clock** signal.

- Clock signals: (a) active high (b) active low:
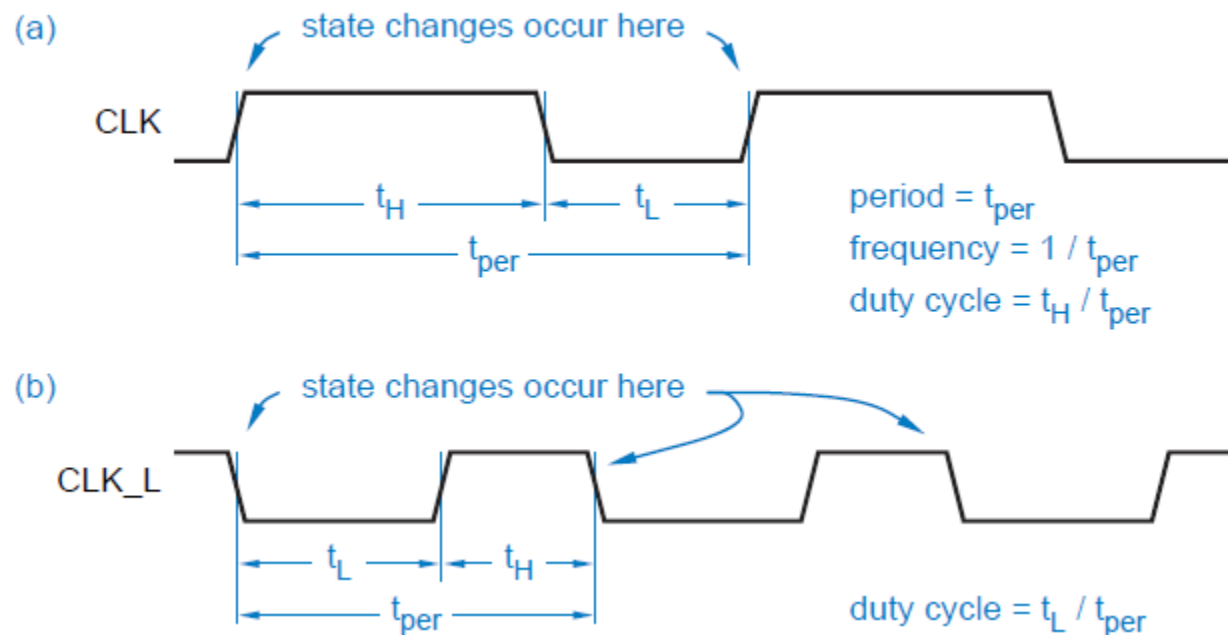
- By convention, a clock signal is **active high** if state changes occur at **clock's rising edge** or when clock is **HIGH**, and active low in complementary case.

- First edge or pulse in a clock period or sometimes period itself is called a **clock tick**.

Introduction

- There are generally **two types of sequential circuits**:

    1) **feedback sequential circuit** which uses ordinary gates and feedback loops to obtain memory in a logic circuit, thereby creating sequential circuit building blocks such as latches and flip-flops that are used in higher-level designs.

    2) **clocked synchronous state machine** which uses these building blocks, in particular edge-triggered D flip-flops, to create circuits whose inputs are examined and whose outputs change in accordance with a controlling clock signal.

Bistable Elements: Digital Analysis

- Simplest sequential circuit (it has **no inputs**):
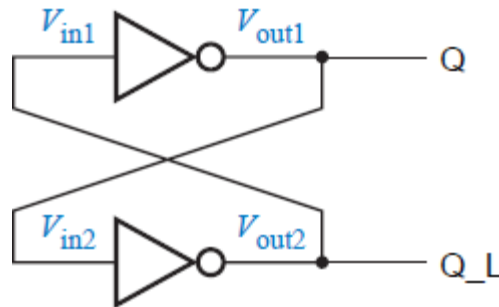
- This circuit is often called a **bistable**, since a strictly digital analysis shows that it has **two stable states**.

- If Q is HIGH, bottom inverter has a HIGH input and a LOW output, which forces the top inverter's output HIGH as we assumed in first place.

- If Q is LOW, bottom inverter has a LOW input and a HIGH output, which forces Q LOW, another stable situation.

- There are **two possible states**, Q = 0 and Q = 1.

Bistable Elements: Analog Analysis

Transfer function:

$$V_{out1} = T(V_{in1})$$

$$V_{out2} = T(V_{in2})$$
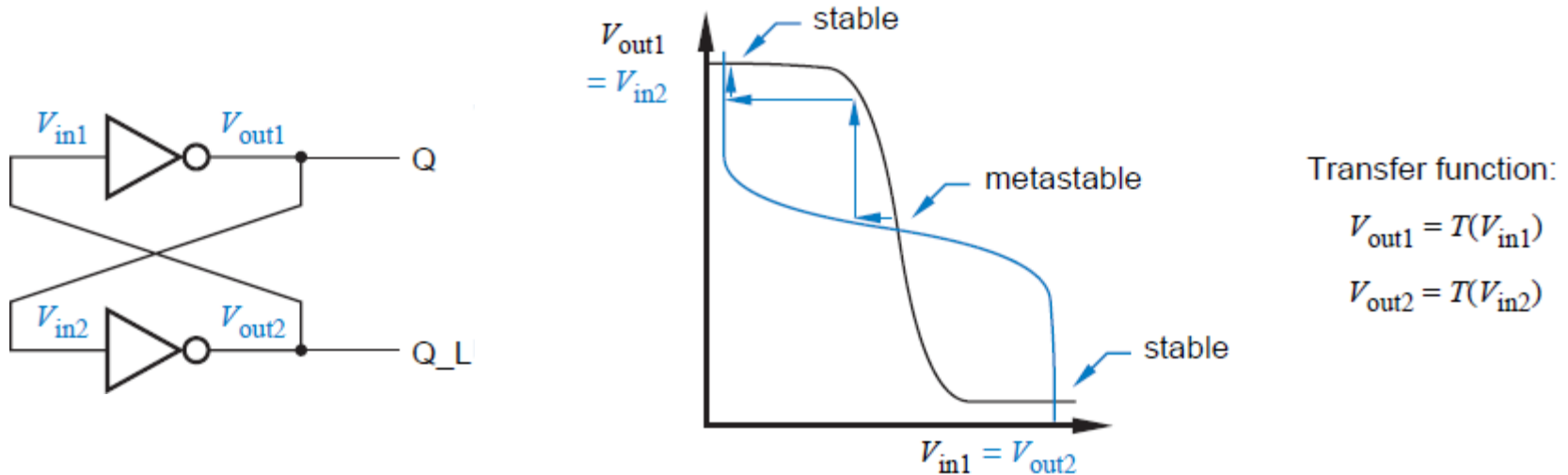
- Lines in figure above (right) show steady-state (**DC**) **transfer functions** T for inverters.

- Feedback loop is in **equilibrium** if **input and output** voltages of both inverters are **constant** DC values consistent with loop connection and inverters' DC transfer function. That is, we must have:

$$V_{in1} = V_{out2} = T(V_{in2}) = T(V_{out1}) = T(T(V_{in1}))$$

- Likewise we must have: $V_{in2} = T(T(V_{in2}))$

Bistable Elements: Analog Analysis

- These **equilibrium points** are ones in figure at which two transfer **curves meet**. There are **three** equilibrium **points**, two stable and one metastable.

- **Metastable point** occurs with V_{out1} and V_{out2} about **halfway** between a valid **logic 1** voltage and a valid **logic 0** voltage. Q and Q_L are **not valid logic signals** at this point. Yet loop equations are satisfied; if we can get circuit to operate at metastable point, it could **theoretically stay** there **indefinitely**.

- Metastable point is **not truly stable**, because **random noise** will tend to drive a circuit that is operating at metastable point toward one of stable operating points (consecutive arrows on right figure).
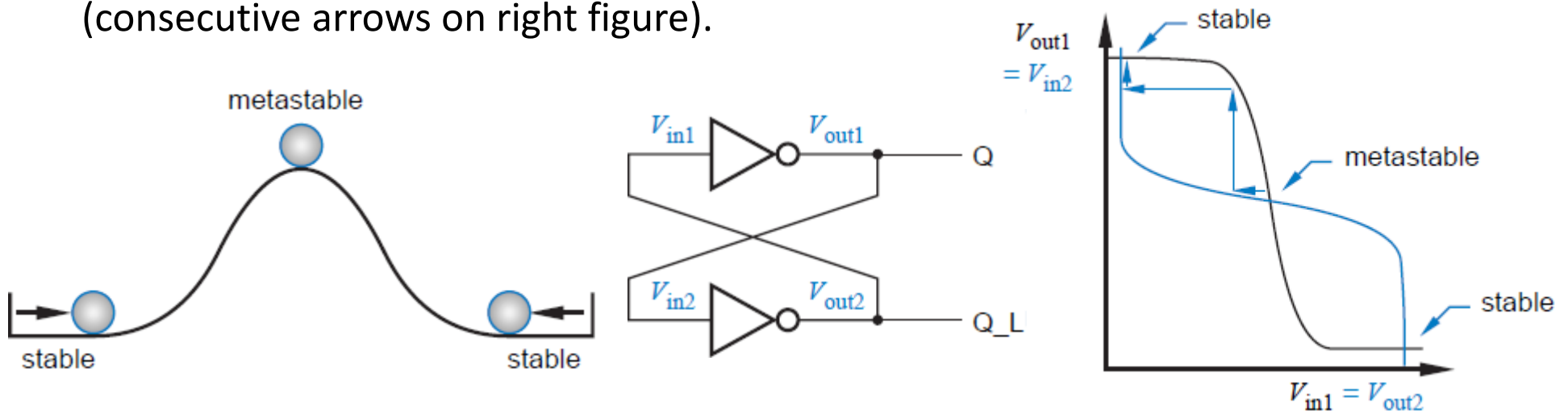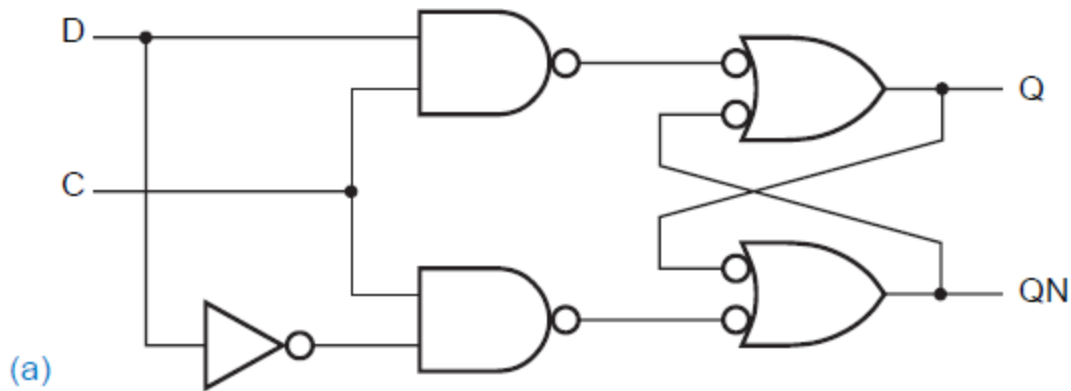
Bistable Elements

- **Bistable may stay** in **metastable** state for an **unpredictable length of time** before nondeterministically settling into one stable state or the other.

- If **simplest** sequential circuit is **susceptible to metastable** behavior, then **all sequential circuits are susceptible** and this behavior is not something that only occurs at power-up.

- Example: in **S-R flip-flops**, a pulse on *S* input forces flip-flop from 0 state to 1 state. A minimum pulse width is specified for *S* input.

  1) Apply a pulse of this width or longer, and flip-flop immediately goes to 1 state.

  2) Apply a very short pulse, and flip-flop stays in 0 state.

  3) Apply a pulse just under minimum width, and flip-flop may go into metastable state.

- Once flip-flop is in metastable state, its operation depends on "**shape of its hill**".

Latches and Flip-Flops

- Latches and flip-flops are basic building blocks of most sequential circuits.

- **Flip-flop** is a sequential device that normally **samples its inputs** and changes **its outputs** only at times determined by a **clocking signal**.

- **Latch** is a sequential device that **watches all of its inputs continuously** and **changes its outputs at any time**, independent of a clocking signal.
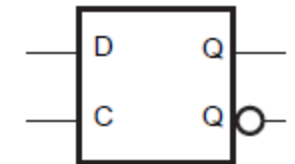
D Latch

- We often need latches simply to **store bits of information**. A **D latch** may be used in such an application:
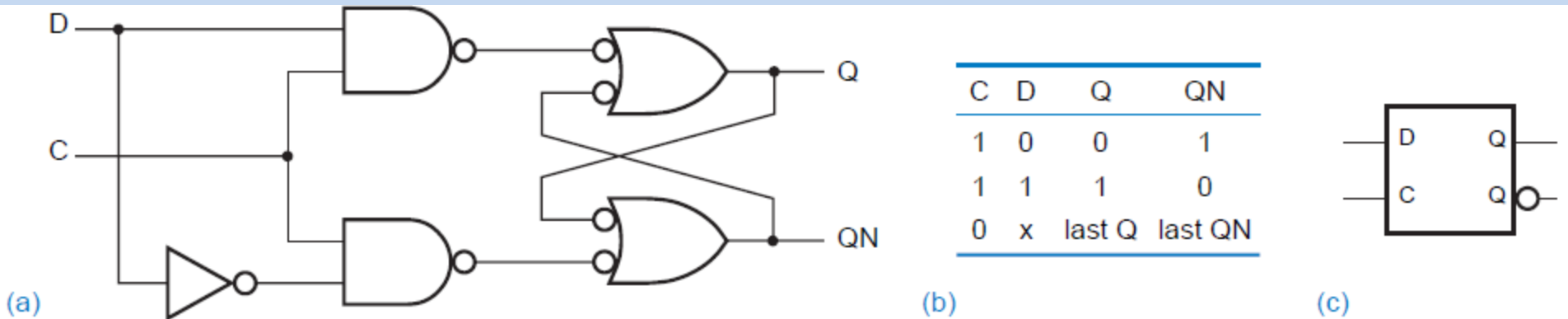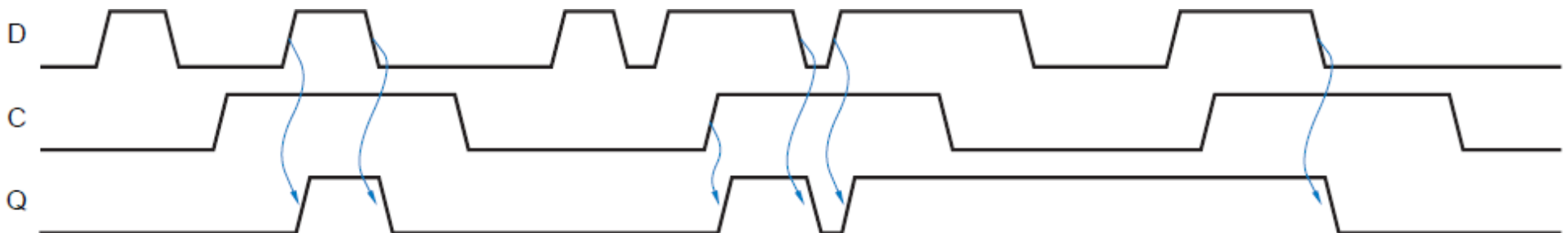

C	D	Q	QN
1	0	0	1
1	1	1	0
0	x	last Q	last QN

(a) (b) (c)

D Latch

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

(a) (b) (c)

- Functional behavior of a D latch for various inputs:

- When **C input is asserted**, Q output follows D input. In this situation, latch is said to be "**open**" and path from D input to Q output is "**transparent**"; circuit is called a "**transparent latch**" for this reason.

- When **C input is negated**, latch "**closes**"; Q output retains its **last value** and no longer changes in response to D, as long as C remains negated.

11

D Latch: Timing Parameters

(a)

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

(b)

(c)
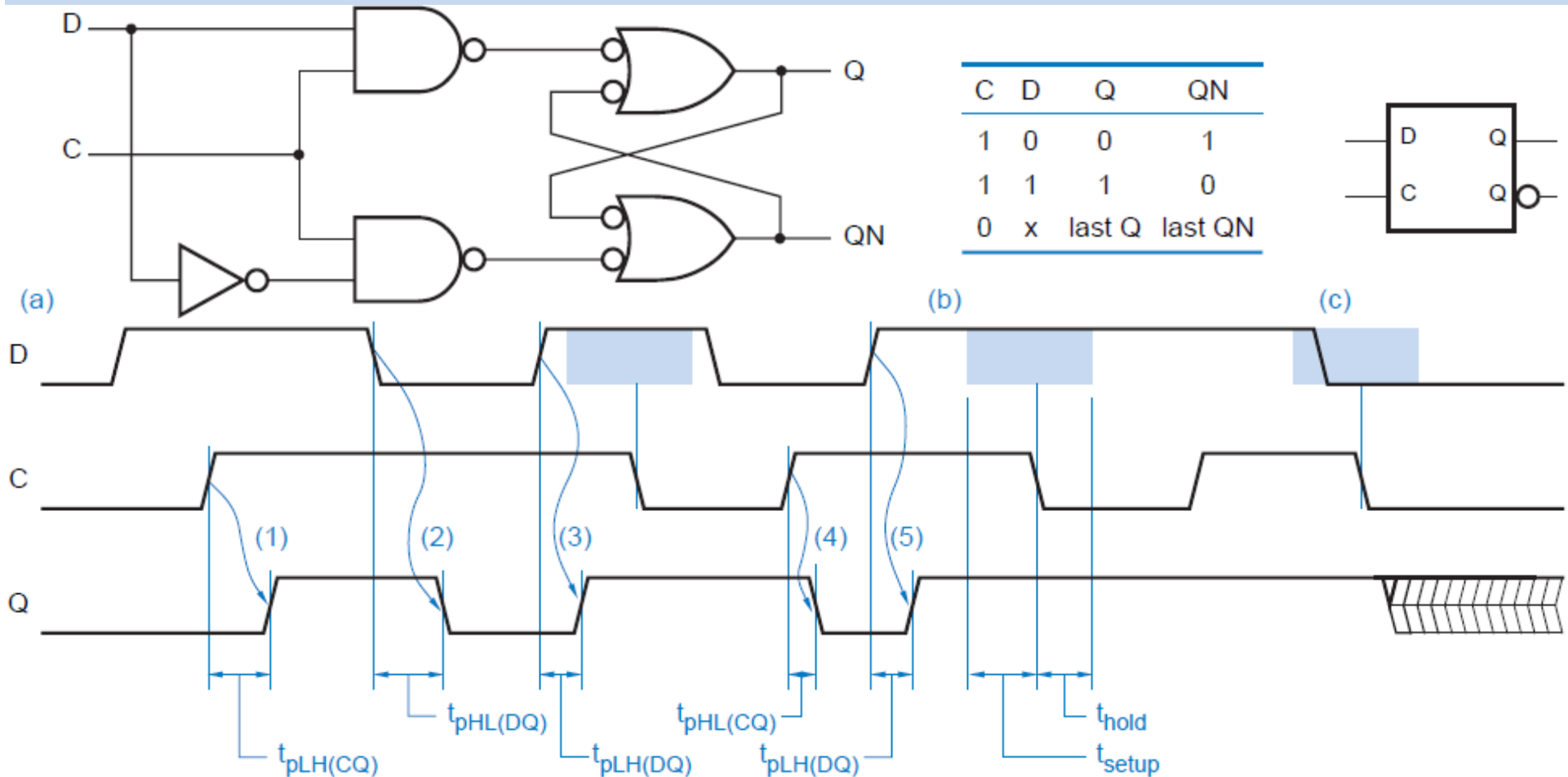
- **Setup time** (**Hold time**) is minimum amount of time the data signal should be held steady **before** (**after**) clock event so that data are reliably sampled by clock.

  D latch does not eliminate metastability problem (unpredictable or metastable).

12

Edge-Triggered D Flip-Flop

- A **positive-edge-triggered D flip-flop** combines **a pair of D latches** to create a circuit that samples its D input and changes its Q and QN outputs only at rising edge of a controlling CLK signal:

(a)

(b)

D	CLK	Q	QN
0	�
0	1
1	⌐	1	0
x	0	last Q	last QN
x	1	last Q	last QN

(c)
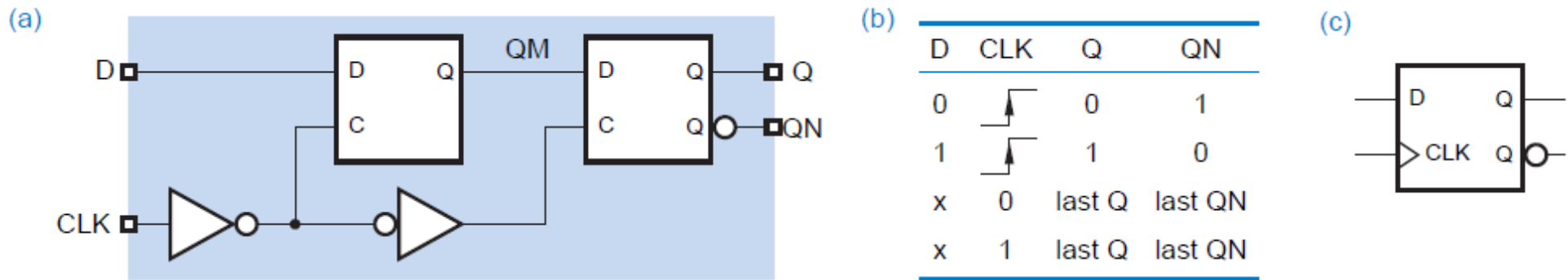
- First latch is called **master**; it is **open** and follows input **when CLK is 0**. When **CLK goes to 1**, **master** latch is **closed** and its output is transferred to second latch, called **slave**.

- **Slave** latch is **open** all the while that **CLK is 1**, but **changes only at beginning of this interval**, because master is closed and unchanging during rest of interval.

- **Triangle** on D flip-flop's CLK input indicates **edge-triggered** behavior, and is called a "dynamic-input indicator".
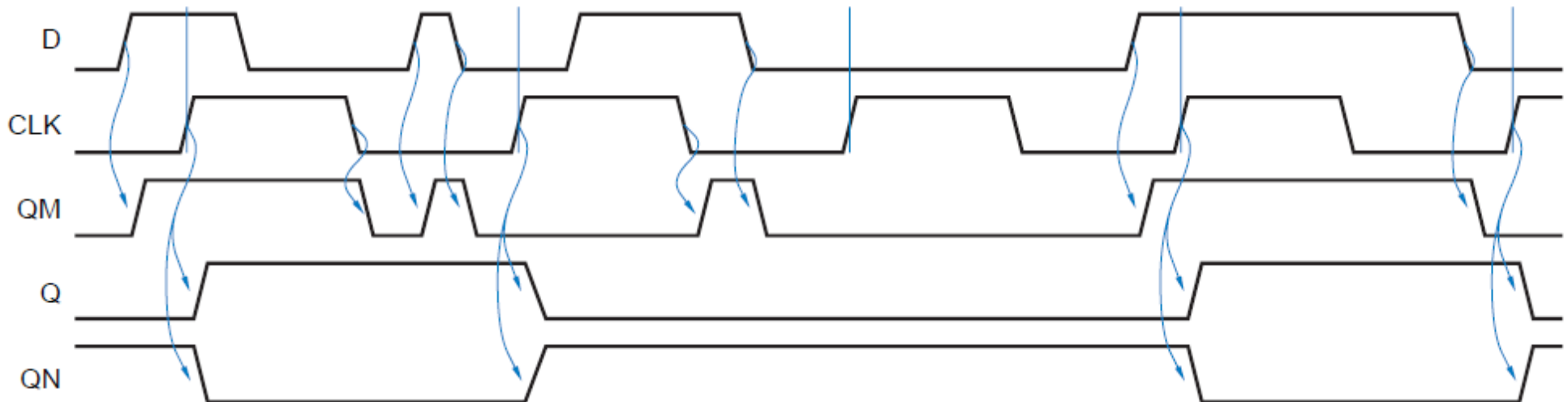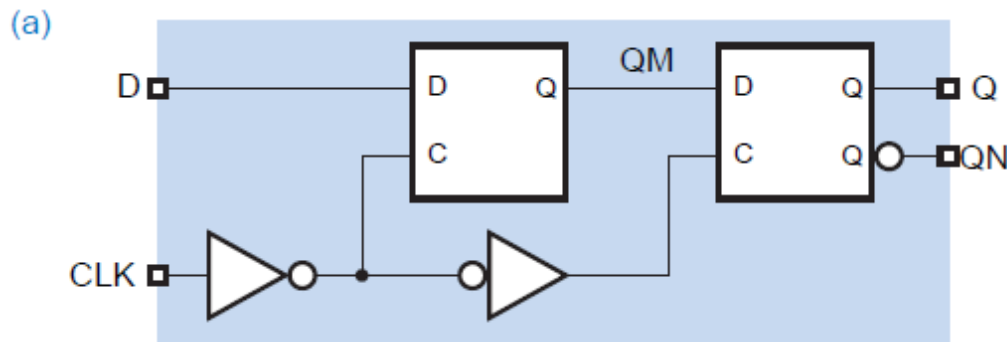
13

Edge-Triggered D Flip-Flop

(a)

(b)

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

(c)
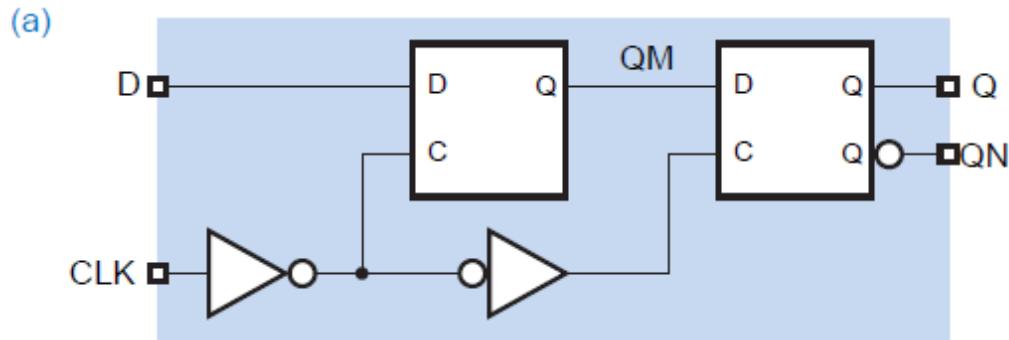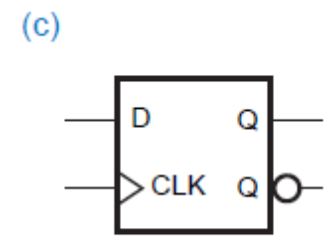
- Functional behavior of a positive-edge-triggered D flip-flop:

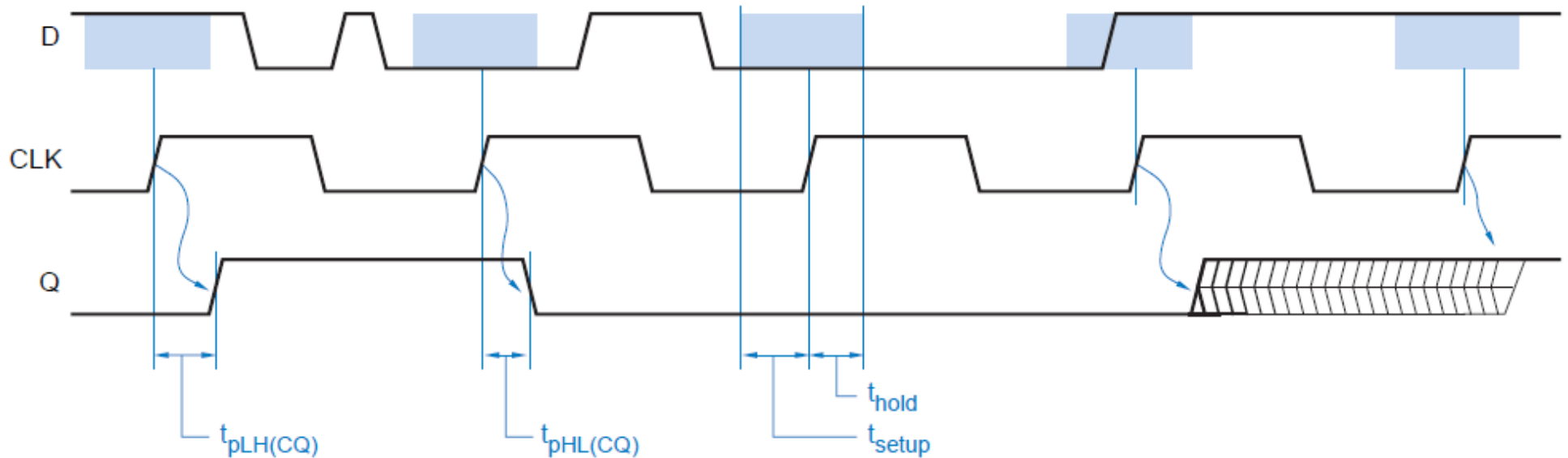# Edge-Triggered D Flip-Flop

(a)

(b)

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

(c)
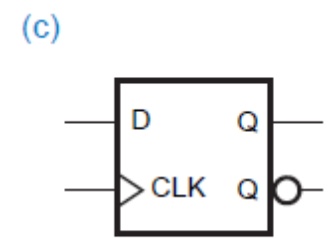
- Timing behavior of a positive-edge-triggered D flip-flop:

- **Different delays** may be specified for **LOW-to-HIGH** and **HIGH-to-LOW** output changes.

Edge-Triggered D Flip-Flop

- Like a D latch, edge-triggered D flip-flop has a **setup and hold time window** during which D inputs must not change.

- This window occurs around triggering edge of clock.

- If **setup and hold times are not met**, flip-flop output will usually go to a **stable**, though **unpredictable, 0 or 1 state**.

- In some cases, output will **oscillate** or go to a **metastable** state halfway between 0 and 1.

- If flip-flop goes into **metastable state**, it will **return to a stable** state on its own only after a **probabilistic delay**.

- It can also be **forced into a stable state** by **applying another triggering clock edge** with a D input that **meets setup- and hold-time** requirements.
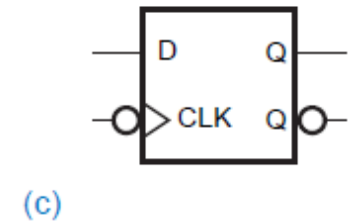
Edge-Triggered D Flip-Flop

- A **negative-edge-triggered D flip-flop** simply inverts clock input, so that all action takes place on falling edge of CLK_L; by convention a falling-edge trigger is considered to be active low:
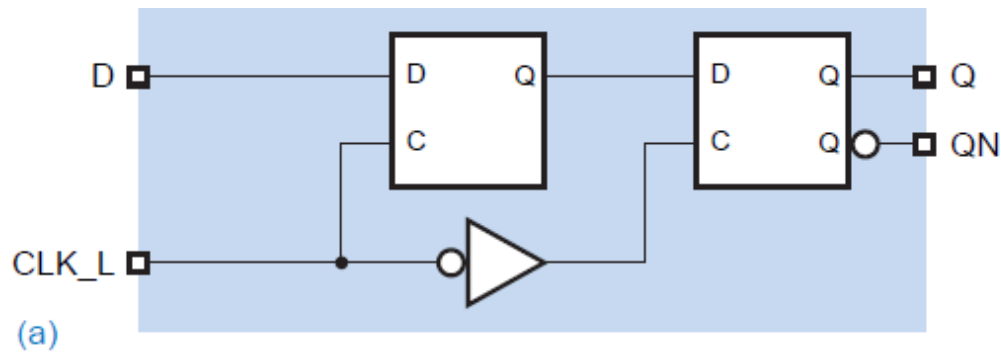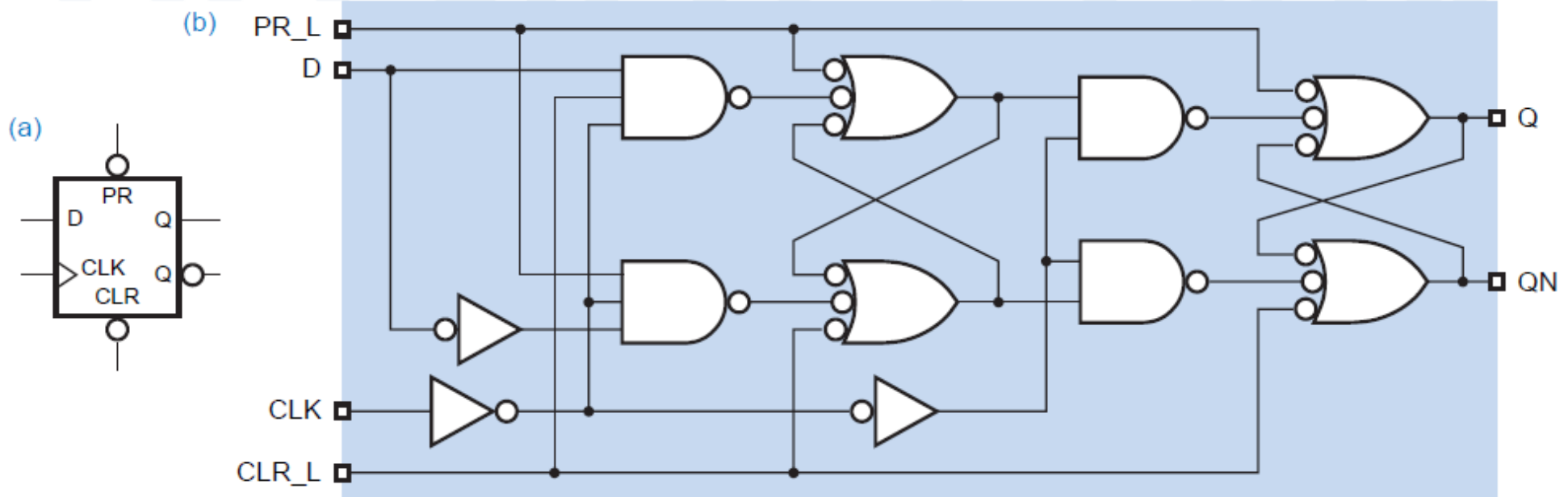

D	CLK_L	Q	QN
0	↓	0	1
1	↓	1	0
x	0	last Q	last QN
x	1	last Q	last QN

(a)

(b)

(c)

Edge-Triggered D Flip-Flop

- Some D flip-flops have **asynchronous inputs** that may be used to force flip-flop to a particular state independent of CLK and D inputs.

- These inputs are typically labeled **PR** (**preset**) and **CLR** (**clear**).

- Positive-edge-triggered D flip-flop with preset and clear:

- Asynchronous inputs are usually used for **initialization** and **testing purposes**, to force a sequential circuit into a known starting state.
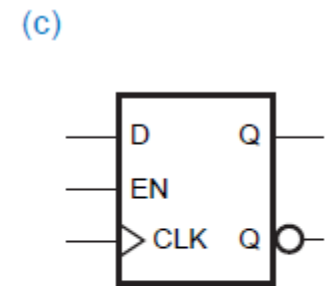
Edge-Triggered D Flip-Flop with Enable

- A commonly desired function in D flip-flops is **ability to hold last value stored**, rather than load a new value, at clock edge.

- This is accomplished by adding an *enable* input.

- Positive-edge-triggered D flip-flop with enable:

(a)
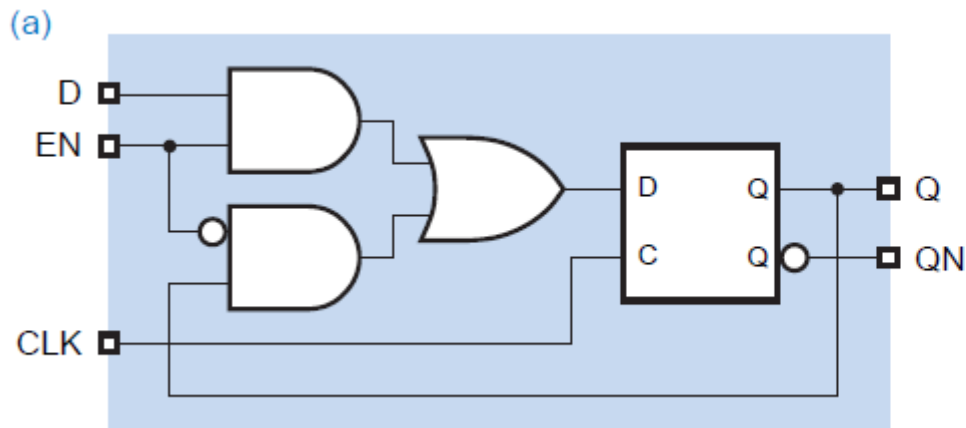
(b)

| D	EN	CLK	Q	QN
0	1	⌐	0	1
1	1	⌐	1	0
x	0	⌐	last Q	last QN
x	x	0	last Q	last QN
x	x	1	last Q	last QN

(c)

- A **2-input multiplexer** controls value applied to internal flip-flop's D input.

Scan Flip-Flop

- An important flip-flop function for **ASIC testing** is **scan capability**.

- Idea is to be able to **drive** flip-flop's **D input with an alternate source of data** during device testing.

- When all of flip-flops are put into testing mode, a **test pattern** can be "**scanned in**" to ASIC using flip-flops' alternate data inputs.

- After test pattern is loaded, flip-flops are put **back into "normal" mode**, and all of flip-flops are clocked normally.

- After one or more clock ticks, flip-flops are put back into test mode, and **test results are "scanned out"**.
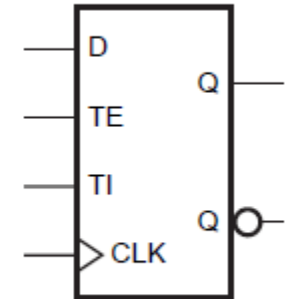
Scan Flip-Flop

- Positive-edge-triggered D flip-flop with scan:
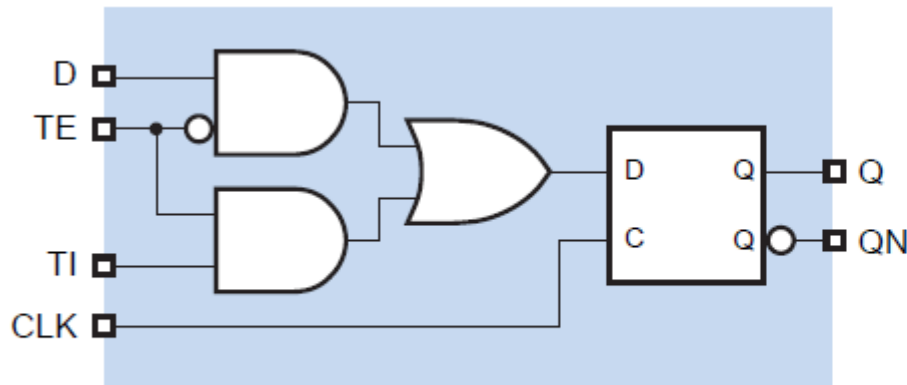
| TE	TI	D	CLK	Q	QN
0	x	0	↑	0	1
0	x	1	↑	1	0
1	0	x	↑	0	1
1	1	x	↑	1	0
x	x	x	0	last Q	last QN
x	x	x	1	last Q	last QN

(a) (b) (c)

- It is a D flip-flop with a **2-input multiplexer** on the D input.

- When TE (test enable) input is negated, circuit behaves like an ordinary D flip-flop. When TE is asserted, it takes its data from TI (test input) instead of from D.

Scan Flip-Flop

- Extra inputs are used to connect all of an ASIC's flip-flops in a **scan chain** for testing purposes.

**Figure 7-23** A scan chain with four flip-flops.
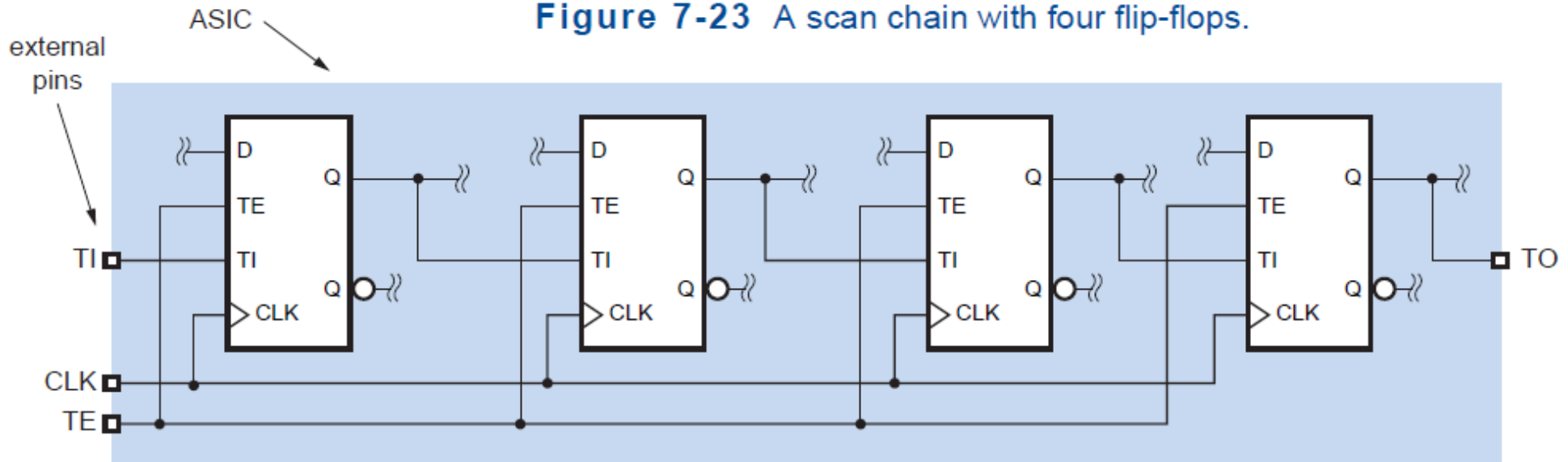
- TE inputs of all flip-flops are connected to a **global TE** input, while each flip-flop's Q output is connected to another's TI input in serial fashion.

- **TI, TE, and TO** (test output) connections are strictly for **testing purposes**.

- To test circuit, global TE input is asserted while *n* clock ticks occur and *n* **test-vector bits** are applied to global TI input and are thereby scanned (shifted) into *n* flip-flops (*n* = 4 in figure above).
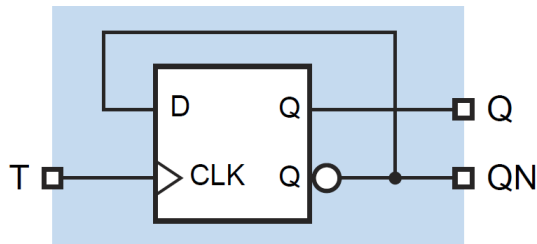
Scan Flip-Flop

- Then TE is negated, and circuit is allowed to run for **one or more additional clock ticks**.

- New state of circuit, represented by new values in *n* flip-flops, can be observed (**scanned out)** at TO by asserting TE while *n* **more clock ticks** occur.

- To make testing process more efficient, another test vector can be scanned in while previous result is being scanned out.

- Scan capability can also be added to other flip-flop types.

T Flip-Flop

- A T (toggle) flip-flop **changes state on every tick** of clock.

- Positive-edge-triggered T flip-flop and its functional behavior:

(a)   (b)
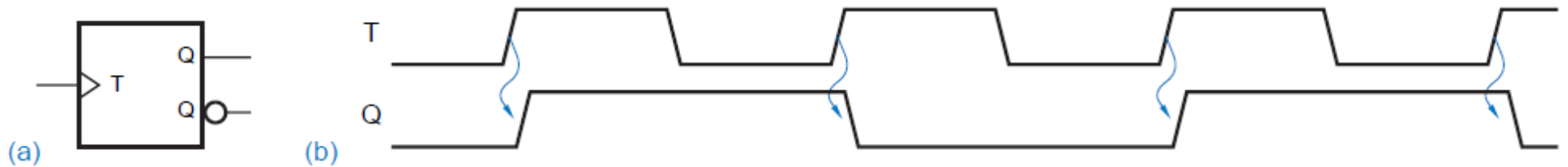
- Signal on flip-flop's Q output has precisely **half frequency** of T input.

- Following figure shows how to obtain a T flip-flop from a D flip-flop:

- T flip-flops are most often used in **counters** and **frequency dividers**.

T Flip-Flop

- In many applications of T flip-flops, **flip-flop need not be toggled** on every clock tick. Such applications can use a **T flip-flop with enable**.

- Positive-edge-triggered T flip-flop with enable:
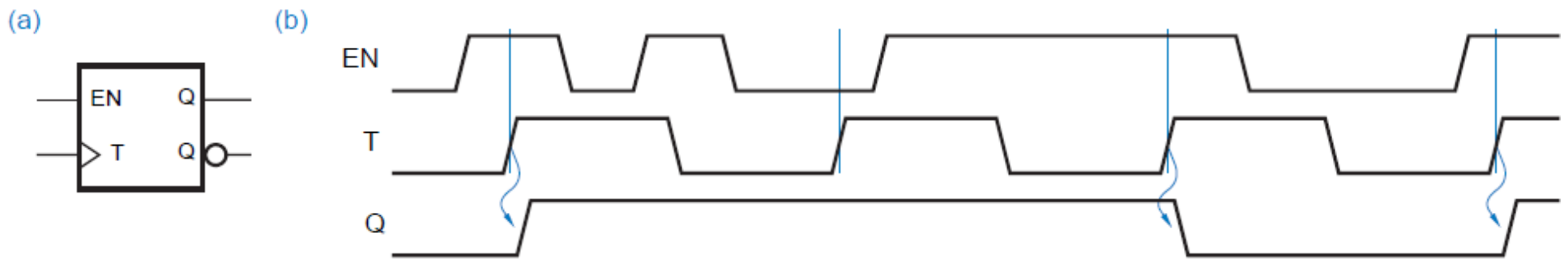
- Flip-flop changes state at triggering edge of clock only if enable signal EN is asserted.

- Like D input on edge-triggered D flip-flop, **EN input must meet specified setup and hold times** with respect to triggering clock edge.

T Flip-Flop

- Possible circuit for a T flip-flop with enable: