DIGITAL LOGIC

Chapter 5: Flip-Flops, Registers, and Counters I

Basic Storage Element

Ru Han
Fall

OUTLINE

Flip-Flops, Registers, and Counters

- Logic circuits that can store information
- Flip-flops, which store a single bit
- Registers, which store multiple bits
- Shift registers, which shift the contents of the register
- Counters of various types
- Verilog constructs used to implement storage elements

COMBINATIONAL VS. SEQUENTIAL LOGICS

- **Combinational Logic Circuits:**
  - The output at any time depends only on the inputs at the same time.
  - Output does not depend on input values at previous time, or some internal states of the device.
- **Sequential Logic Circuits:**
  - The output depends on both the present inputs and the present states.
  - The state represents the stored information that is determined by what the inputs were at some time in the past.

SEQUENTIAL LOGICS

- In this chapter we will introduce circuits that can be used as storage elements.
  - Why do we need such circuits ?
  - Example: Alarm devices.

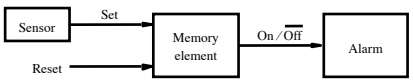


Figure 5.1. Control of an alarm system.

STORAGE ELEMENTS

- A rudimentary memory element, consisting of a loop that has two inverters.

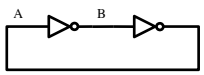
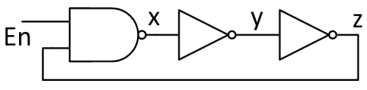


Figure 5.2. A simple memory element.

SEQUENTIAL LOGIC

- Ring Oscillator



En = 0 En = 1

- This circuit is not combinational:

If En = 1, we cannot specify if z = 0 or z = 1 without knowing the internal state.

- We have built a sequential circuit using combinational components (gates) by feeding the output back to one of the inputs.

5.1 BASIC LATCH

- A memory element built with NOR gates.
- Its inputs, *Set* and *Reset*, provide the means for changing the state, Q, of the circuit.

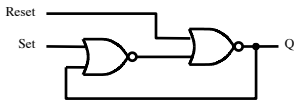
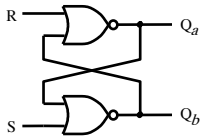


Figure 5.3. A memory element with NOR gates.

7

BASIC LATCH

SR Latch



(a) Circuit

Figure 5.4 A basic latch built with NOR gates.

- In this circuit, the two NOR gates are said to be connected in **cross-coupled** style.
- Q_a and Q_b are both inputs and outputs of the NOR gates.

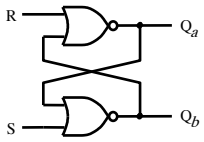
  Q(t): Value of Q at present time (input)

  Q(t+1): Value of Q after 1 NOR gate delay (output)

8

BASIC LATCH

SR latch: Characteristic table



(a) Circuit

S	R	Q_a	Q_b	
0	0	0/1	1/0	(no change)
0	1	0	1	
1	0	1	0	
1	1	0	0	

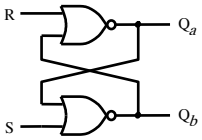
(b) Characteristic table

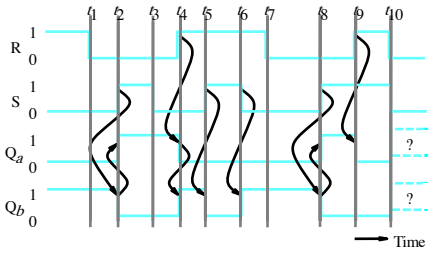
Figure 5.4 A basic latch built with NOR gates.

9

BASIC LATCH

- ***SR latch***: Timing diagram





(c) Timing diagram

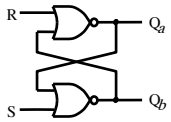
Figure 5.4 A basic latch built with NOR gates.

10

BASIC LATCH

Drawback?

SR latch: Redraw Characteristic table– stable states only



(a) Circuit

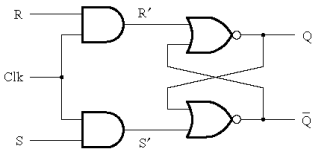
S	R	Q_a	Q_b	Name of State
0	0	$Q_a(t)$	$Q_b(t)$	HOLD
0	1	0	1	RESET
1	0	1	0	SET
1	1	0	0	ILLEGAL

- ***Hold***: Q_a and Q_b retain previous values
- ***Reset***: Stabilize at $Q_a = 0$ and $Q_b = 1$ regardless of prev. states
- ***Set***: Stabilize at $Q_a = 1$ and $Q_b = 0$
- ***Illegal***: illegal because $Q_a = Q_b (= 0)$
- This circuit is intended to maintain $Q_a \neq Q_b$, but the illegal state violates this.

11

5.2 GATED SR LATCH

- ***Gated SR latch***



(a) Circuit

- Characteristic table

Clk	S	R	$Q(t+1)$
0	x	x	$Q(t)$ (no change)
1	0	0	$Q(t)$ (no change)
1	0	1	0
1	1	0	1
1	1	1	x

(b) Characteristic table

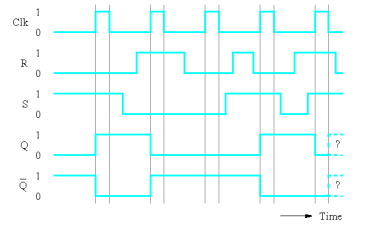
Figure 5.5 Gated SR latch.

- Timing diagram

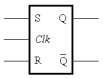
12

GATED SR LATCH

- Timing diagram



(c) Timing diagram



(d) Graphical symbol

Figure 5.5 Gated SR latch.

GATED SR LATCH WITH NAND GATES

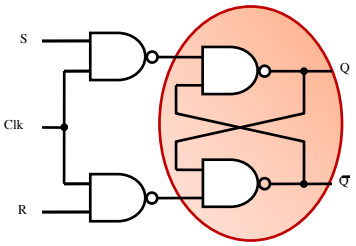
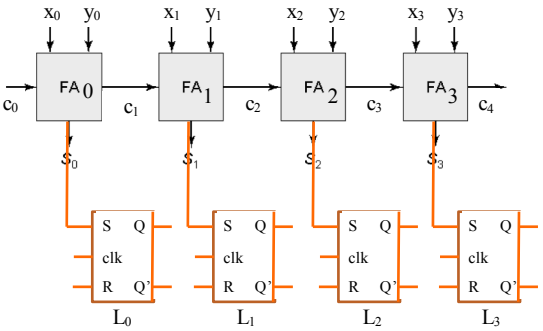


Figure 5.6. Gated SR latch with NAND gates.

APPLICATION OF SR LATCH



If there is no clk signal, SR Latch will make a mistake!!!

5.3 GATED D LATCH

- We can eliminate the illegal state from the SR latch by using additional circuitry to make sure S = R =1 never occurs.
- ***D-Latch***

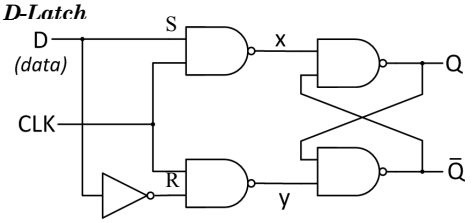
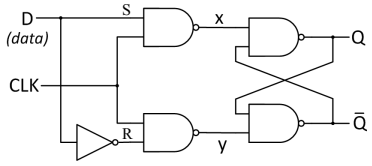


Figure 5.7. Gated D latch.

GATED D LATCH

D-Latch



Characteristic table

CLK	D	x	y	Q (t+1)	$\bar{Q}$(t+1)	Name of State
0	0	1	1	Q(t)	$\bar{Q}$(t)	HOLD
0	1	1	1	Q(t)	$\bar{Q}$(t)	HOLD
1	0	1	0	0	1	RESET
1	1	0	1	1	0	SET

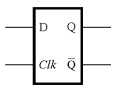
Clk	D	$Q(t+1)$
0	x	$Q(t)$
1	0	0
1	1	1

(b) Characteristic table

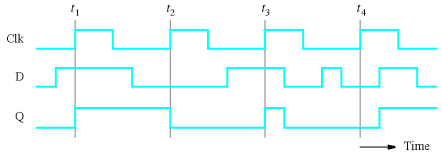
GATED D LATCH

Clk	D	$Q(t+1)$
0	x	$Q(t)$
1	0	0
1	1	1

(b) Characteristic table



(c) Graphical symbol



(d) Timing diagram

Figure 5.7. Gated D latch.

Since the output of the gated D latch is controlled by the level of the clock input, the latch is said to be ***level sensitive***.

GATED D LATCH

D-Latch

- Compact Characteristic table

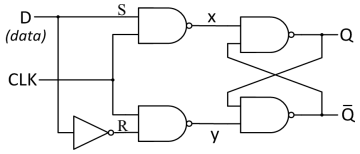
CLK	Q (t+1)	$\overline{Q}$(t+1)
0	Q(t)	$\overline{Q}$(t)
1	D	$\overline{D}$

- This D Latch is a "**level-sensitive**" device:
  - Q follows D when CLK = 1
  - Q remains stable when CLK = 0
- Potential problems:
  - If CLK = 1 and the D input contains "noise", the noise will appear on Q and $\overline{Q}$.
  - Unpredictable state may occur if D changes at the same time that CLK goes from 1 → 0.

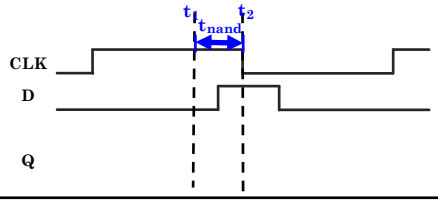
19

EFFECTS OF PROPAGATION DELAYS

D-Latch



t_{nand} is the delay of 2 NAND gate



20

EFFECTS OF PROPAGATION DELAYS

D-Latch's timing parameter

Setup Time: t_{su} **= 0.3ns**
Time before CLK:1→0 that D must remain stable

Hold Time: t_h **= 0.2ns**
Time after CLK:1→0 that D must remain stable

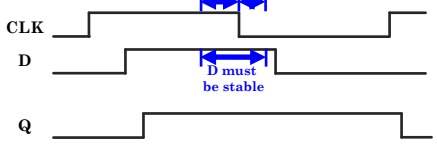


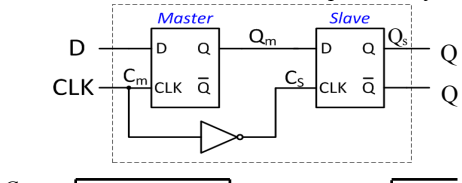
Figure 5.8 Setup and hold times.

21

5.4 EDGE-TRIGGERED D FLIP-FLOPS

Master-Slave D Flip-Flops

- We can create an edge-triggered D flip-flop using two level-sensitive D-Latches with complementary clocks.



•Master Latch responds to input (Q_m changes)
•Slave latch is hold:
•Q unchanged

•Master Latch is hold
•Slave latch responds to inputs Q_m;
•Output Q changes

22

MASTER-SLAVE D FLIP-FLOP

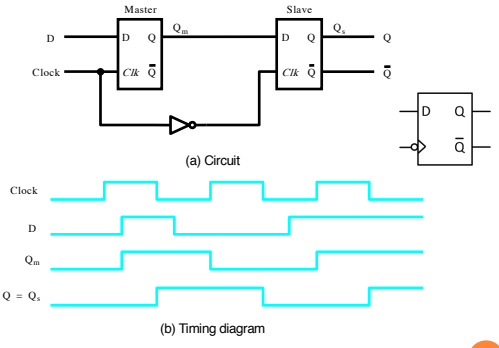
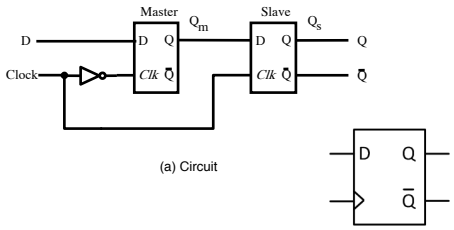


Figure 5.9 Master-slave D flip-flop.

23

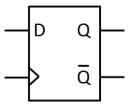
MASTER-SLAVE D FLIP-FLOP

- We can alter the master-slave to be rising-edge triggered by driving the master with **CLK and the slave with CLK.**



24

MASTER-SLAVE D FLIP-FLOP



D	CLK	Q (t+1)	$\bar{Q}$(t+1)
0	↑	0	1
1	↑	1	0
X	↛	Q(t)	$\bar{Q}$(t)

NOTE:
- ↑ : Rising edge
- ↛ : anything except a rising edge (0, 1, ↓)
- X : don't care
- D value must be stable for one set-up time prior to clock transition

25

MASTER-SLAVE D FLIP-FLOP

- The master-slave D flip-flop passes the value on D to Q on the falling edge (or rising edge) of the clock signal. ⇒ A falling (or rising) edge triggered D Flip-Flop.

▷ denotes rising-edge-triggered clock input

○▷ denotes falling-edge-triggered clock input

Figure 5.9 Master-slave D flip-flop.

26

LEVEL-SENSITIVE VERSUS EDGE-TRIGGERED STORAGE ELEMENTS

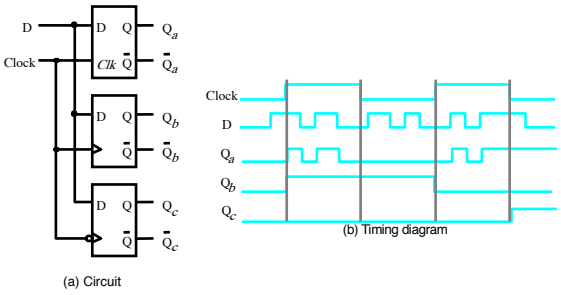


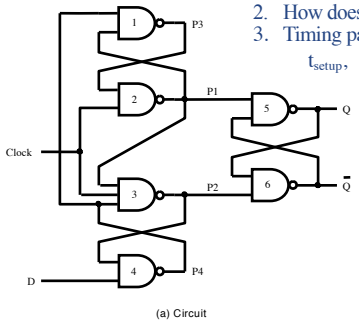
Figure 5.10. Comparison of level-sensitive and edge-triggered D storage elements.

27

OTHER TYPES OF EDGE-TRIGGERED D FLIP-FLOPS

1. Area?
2. How does it works?
3. Timing parameters?

t_{setup}, t_{hold}



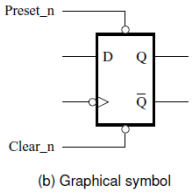
1. A similar circuit, constructed with *NOR gates*, can be used as a *negative-edge-triggered* flip-flop.
2. This type is more efficient!

Figure 5.11. A positive-edge-triggered D flip-flop.

28

D FLIP-FLOPS WITH CLEAR AND PRESET

- We can further alter the flip-flop by proving low-true inputs for Preset_n and Clear_n.
  - Preset_n=0: Q = 1, regardless of other inputs, including clock (asynchronous SET)
  - Clear_n = 0: Q = 0, regardless of other inputs, including clock (asynchronous RESET)
  - Preset_n = Clear_n = 1: it functions as an edge-triggered D flip-flop.
  - Preset_n = Clear_n = 0: an illegal input combination.



29

D FLIP-FLOPS WITH CLEAR AND PRESET

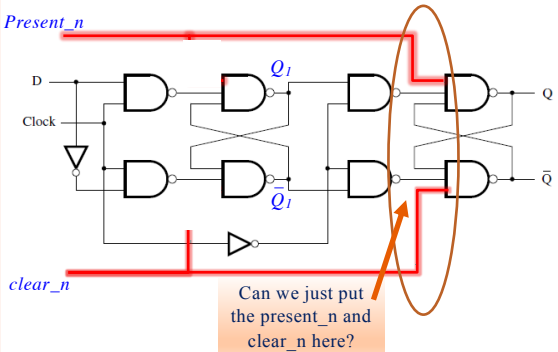


Figure 5.12. Master-slave D flip-flop with *Clear* and *Preset*

30

D Flip-Flops with Clear and Preset

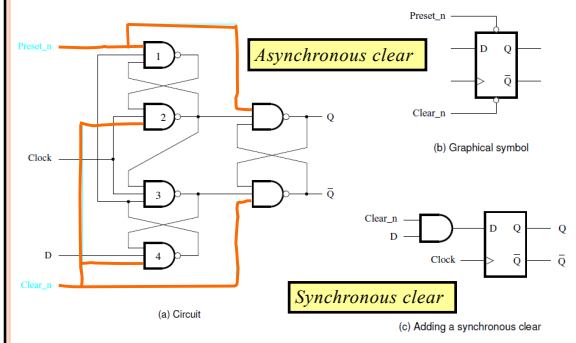
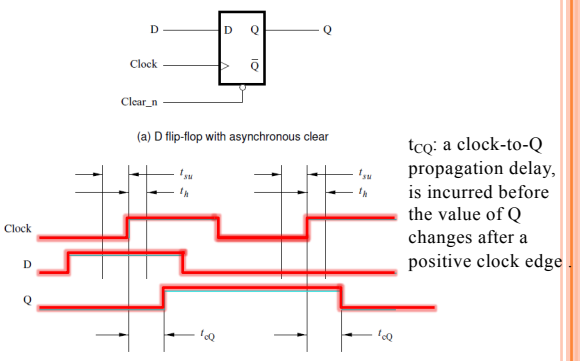


Figure 5.13. Positive-edge-triggered D flip-flop with *Clear* and *Preset*.

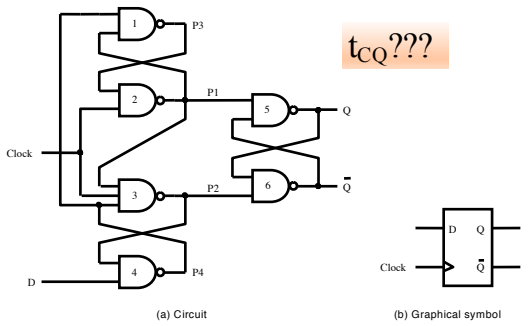
Flip-Flop Timing Parameters



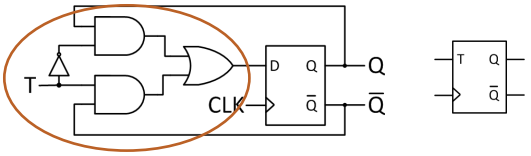
t_{CQ}: a clock-to-Q propagation delay, is incurred before the value of Q changes after a positive clock edge.

Figure 5.14. Timing for a flip-flop.

Flip-Flop Timing Parameters



5.5 T Flip-Flops



$D = Q\overline{T} + \overline{Q}T$

- If T = 0, then D = Q
- If T = 1, then D = $\overline{Q}$

CLK	T	D	Q (t+1)	$\overline{Q}$(t+1)	Status
↓	X	X	Q(t)	$\overline{Q}$(t)	HOLD
↑	0	Q(t)	Q(t)	$\overline{Q}$(t)	HOLD
↑	1	$\overline{Q}$(t)	$\overline{Q}$(t)	Q(t)	TOGGLES

Very useful in counters

T	Q(t+1)
0	Q(t)
1	$\overline{Q}$(t)

T Flip-Flops

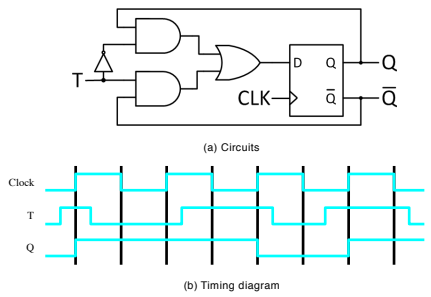
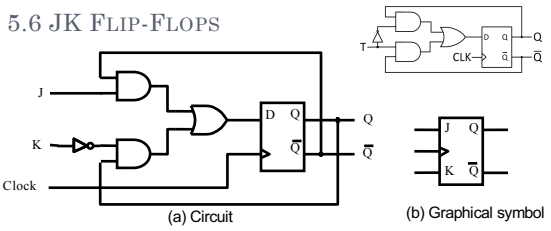


Figure 5.15. T flip-flop.

5.6 JK Flip-Flops



$D = J\overline{Q} + \overline{K}Q$

- J = K = 0 $\Rightarrow$ D = Q
- J = 0, K = 1 $\Rightarrow$ D = 0
- J = 1, K = 0 $\Rightarrow$ D = 1
- J = K = 1 $\Rightarrow$ D = $\overline{Q}$

J	K	Q (t+1)	Status
0	0	Q(t)	HOLD
0	1	0	RESET
1	0	1	SET
1	1	$\overline{Q}$(t)	TOGGLE

(c) Characteristic table

Figure 5.16 JK flip-flop

The JK Flip-Flop can mimic the behavior of SR, D, or T Flip-Flops by appropriate connection of the J, K inputs.

EDGE-TRIGGERED FLIP-FLOPS

Characteristic Equation

- Boolean equation describing edge-triggered flip-flops. Gives Q(t+1) when next clock edge occurs.

- D Flip-Flop (No Preset or Clear)

  $Q(t+1) = D$

- T Flip-Flop

  $Q(t+1) = \overline{T} \cdot Q(t) + T \cdot \overline{Q}(t)$

- JK Flip-Flop

  $Q(t+1) = \overline{K} \cdot Q(t) + J \cdot \overline{Q}(t)$

37

5.7 SUMMARY OF TERMINOLOGY

Basic latch

- Basic latch is a feedback connection of two NOR gates or two NAND gates, which can store one bit of information.
- It can be set to 1 using the *S* input and reset to 0 using the *R* input.

Gated latch

- Gated latch is a basic latch that includes input gating and a control input signal.
- The latch retains its existing state when the control input is equal to 0. Its state may be changed when the control signal is equal to 1. (positive latch)

Gated SR latch *Gated D latch*

38

SUMMARY OF TERMINOLOGY

Flip Flop

- A *flip-flop* is a storage element that can have its output state changed only on the edge of the controlling clock signal.

- If the state changes when the clock signal goes from 0 to 1, we say that the flip-flop is *positive-edge triggered*.
- If the state changes when the clock signal goes from 1 to 0, we say that the flip-flop is *negative-edge triggered*.

39

CONCLUSION

- Understand the basic store element architecture
- Understand the t_{su}, t_{hold}, and t_{CQ}

- Assignment:
- Pages: 321-322
- 5.1, 5.2, 5.3

40