Lecture 8: Latches and Flip-flops

CS207: Digital Logic

Jialin Liu

Department of Computer Science and Engineering (CSE)
Southern University of Science and Technology (SUSTech)

28 October 2022

Acknowledgement

These slides were prepared based on the slides by Dr. Jianqiao Yu and the ones by Prof. Georgios Theodoropoulos of the Department of CSE at the SUSTech, as well as the contents of the following book:
M. M. Mano and M. Ciletti, *Digital design: with an introduction to the Verilog HDL*. Pearson, 2013

Recap: Arithmetic Circuits

- Standard components: combinational circuits that are employed extensively in the design of digital systems.
  - Adders, subtractors, multipliers, comparators, decoders, encoders, and multiplexers.
- Adders can be used to implement subtractors and multipliers.
- Encoders and multiplexers can be used to implement any combinatorial circuits.

Recap: Half Adder and Full Adder

- A combinational circuit that performs the addition of two bits is called a **half adder** (半加器).
- A combinational circuit that performs the addition of three bits (two significant bits and a previous carry) is a **full adder** (全加器).
- Two half adders can be employed to implement a full adder.

Recap: Subtraction (减法)

- **Subtraction** is another basic function of arithmetic operations of information-processing tasks of digital computers.
- A combinational circuit that performs the subtraction of 2 bits is called a **half-subtractor** (半减器).
- The subtraction operation involves 3 bits — the **minuend** (被减数) bit, **subtrahend** (减数) bit, and the **borrow** (借位数) bit, and produces a different result as well as a borrow. The combinational circuit that performs this type of subtraction operation is called a **full-subtractor** (全减器) .
- Adders can be used to implement subtractors.

Recap: Binary Multiplication (乘法器)

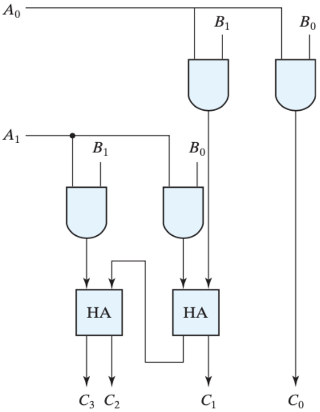
- Same as multiplication of decimal numbers.
  - Starting from the least significant bit (right → left),
    - the multiplicand is multiplied by each bit of the multiplier;
    - each such multiplication forms a partial product;
    - successive partial products are shifted one position to the left.
  - The final product is obtained from the sum of the partial products.
- Example: $B_1 B_0 \times A_1 A_0$

$$\begin{array}{cccc}
 & & B_1 & B_0 \\
 & & A_1 & A_0 \\ \cline{3-4}
 & & A_0 B_1 & A_0 B_0 \\
 & A_1 B_1 & A_1 B_0 & \\ \hline
C_3 & C_2 & C_1 & C_0
\end{array}$$

Binary Multiplier Implemented with A Combinational Circuit

- Example: $B_1 B_0 \times A_1 A_0$
  - The two partial products are added with two half-adder (HA) circuits.

		B_1	B_0
		A_1	A_0
		A_0B_1	A_0B_0
	A_1B_1	A_1B_0	
C_3	C_2	C_1	C_0



Recap: Binary Multiplier

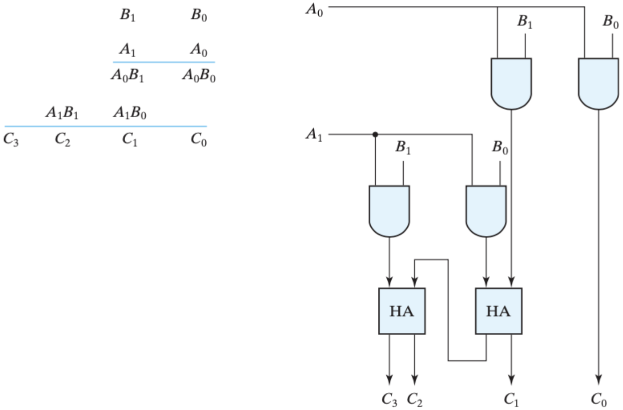
- A combinational circuit binary multiplier with more bits can be constructed in a similar fashion.
  - A bit of the multiplier is `AND` with each bit of the multiplicand in as many levels as there are bits in the multiplier. → $(J \times K)$ `AND` gates
  - The binary output in each level of `AND` gates is added with the partial product of the previous level to form a new partial product. → $(J-1)K$-bit adders
  - The last level produces the product.

- For J multiplier bits and K multiplicand bits, we need $(J \times K)$ `AND` gates and $(J-1)K$-bit adders to produce a product of $(J+K)$ bits.

Exercise

Design a binary multiplier that multiplies a 4-bit number, $B_3B_2B_1B_0$, by a 3-bit number $A_2A_1A_0$. The circuit is to be implemented using `AND` gates and `full adders`.

- Example: $B_1B_0 \times A_1A_0$
  - The two partial products are added with two half-adder (HA) circuits.



Exercise

Design a binary multiplier that multiplies a 4-bit number, $B_3B_2B_1B_0$, by a 3-bit number $A_2A_1A_0$. The circuit is to be implemented using `AND` gates and `full adders`.

Solution:

- "For J multiplier bits and K multiplicand bits, we need $J \times K$ `AND` gates and $J-1$ `K-bit adders` to produce a product of $(J+K)$ bits." [1]
- The are 3 multiplier bits and 4 multiplicand bits, hence $K=4$ and $J=3$. → 12 `AND` gates and two `4-bit adders` are needed to produce a product of 7 bits.

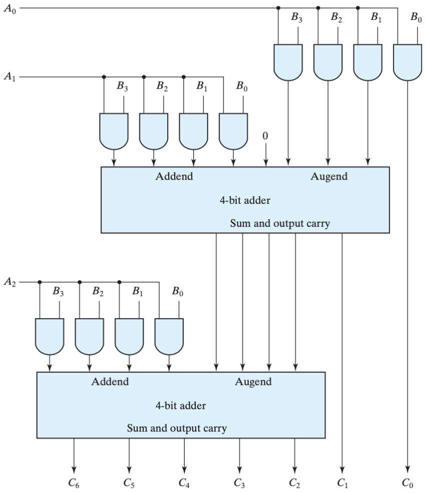


FIGURE 4.16
Four-bit by three-bit binary multiplier

Figure: Screenshot of Figure 4.16 in [1].

Tentative Course Plan

- Week 1 (Sept. 9): Lecture 1 - Binary Numbers
- Week 2 (Sept. 16): Lecture 2 - Boolean Algebra and Logic Gates
- Week 3 (Sept. 23): Lecture 3 - Gate-Level Minimisation
- Week 4 (Sept. 30): Lecture 4 - Two-Level Implementation
- ———— (Oct. 9): Assignment 1 Analysis & Lecture 5 - Combinational Logic
- Week 5 (Oct. 14): Lecture 6 - Standard Components
- Week 6 (Oct. 21): Lecture 7 - Arithmetic Circuits
- Week 7 (Oct. 28): Lecture 8 - Latches and Flip-flops

Tentative Course Plan (continued)

- Week 8 (Nov. 4): Lecture 8 - Latches and Flip-flops (cont'd) & Lecture 9 - Sequential Logic
- Offline (video/PDF): Assignment 2 Analysis
- **Week 9 (Nov. 11): No course (University Sports Day)**
- Week 9: Release project on *Sakai*
- **Week 10 (Nov. 18): No course according to University's calendar**
- Week 11 (Nov. 25): Open-book in-class test (Lectures 1–8, including FFs and latches)
- Week 11 (Nov. 25): Lecture 10 - Sequential Logic (cont'd)
- Week 12 (Dec. 2): Assignment 3 Analysis & Lecture 11 - Registers
- Week 13 (Dec. 9): Lecture 12 - Registers (cont'd) and Counters
- Week 14 (Dec. 16): Lecture 13 - Counters (cont'd)
- Week 15 (Dec. 23): Assignment 4 Analysis & Lecture 14 - Programmable Logic Devices & Project Inspection #1
- Week 16 (Dec. 30): Project Inspection #2

Outline of This Lecture

Introduction to Sequential Logic

Memory Elements

- Memory Element: Latches
- Memory Element: Flip-Flop

Characteristic Table & Characteristic Equation

Summary

Vocabulary

- Synchronous Sequential Circuits (同步时序电路).
- Asynchronous sequential circuit (异步时序电路).
- clock pulse (时钟脉冲), denoted by `clock` or `clk`.
- Level sensitive device. (时钟信号电平敏感的存储单元电路).
- Edge-sensitive devices (时钟信号边沿敏感的存储单元电路).
- Memory elements (记忆元件):
  - Flip-flop (触发器) (FF).
  - Latch (锁存器).
- Set: S (置位), usually means “set the state set to 1”.
- Reset: R (复位), usually means “set the state to 0”.
- Present state: Q_t or Q (现态).
- Next State: Q_{t+1} or Q' (次态).
- Ways to describe a sequential circuit:
  K-map (卡诺图), logic diagram (逻辑电路图), function table (功能表), state table (状态表), state diagram (状态图), characteristic equations (特性方程) (next state equation, 次态方程), excitation table (激励方程).

Outline of This Lecture

Introduction to Sequential Logic

Memory Elements

Characteristic Table & Characteristic Equation

Summary

Sequential Circuits

- Almost all electronic consumer products
  - send, receive, store, retrieve, and process information
  - depends on the ability to store binary — has memory.
- The logic circuits whose outputs at any instant of time depend on the present inputs as well as on the past outputs are called **sequential circuits**.
- **Sequential circuits** act as storage elements.

Sequential Circuits

- A **sequential circuit** is specified by
  - a time sequence of inputs,
  - outputs, and
  - internal states.
- A combinational circuit, but with **memory elements** (记忆元件) connected in a feedback path (反馈回路).
  - A *memory element* is a medium in which one bit of information (0 or 1) can be stored or retained until necessary, and thereafter its contents can be replaced by a new value.
- Outputs are binary functions of not only inputs, but also the present state of the circuit.

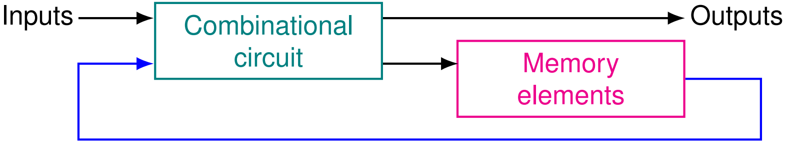


Figure: Block diagram of sequential circuit.

Types of Sequential Circuits

- Sequential circuits are broadly classified into two main categories depending on the timing of their signals.
  - **Synchronous sequential circuit** (同步时序电路)
  - **Asynchronous sequential circuit** (异步时序电路)

Synchronous Sequential Circuits (同步时序电路)

- **Synchronous or clocked sequential circuit**: A system whose behaviour can be defined from the knowledge of its signals at discrete instants of time.
  - The synchronisation is achieved by a timing device known as a **system clock** or **clock generator**.
  - Synchronous sequential circuits use clock pulses to control storage elements.
  - “Synchronous” because the circuit activity and the updating of storage are synchronised.
  - The outputs are affected only with the application of a **clock pulse** (时钟脉冲), commonly denoted by `clock` or `clk`.
    - Clock pulse: when changes will happen.
    - Other signals: what changes will happen.

Asynchronous Sequential Circuit (异步时序电路)

- Asynchronous sequential circuit: there is no clock pulse.
- In other words, it is not driven by a periodic clock signal to synchronise its internal states.
- A system whose behaviour depends upon input signals at any instant of time and the order in which the inputs change.

Types of Sequential Circuits

- Sequential circuits are broadly classified into two main categories depending on the **timing of their signals.**
  - **Synchronous sequential circuit** (同步时序电路): A system whose behaviour can be defined from the knowledge of its signals at discrete instants of time.
    - The synchronisation is achieved by a timing device known as a **system clock** or **clock generator**.
    - The outputs are affected only with the application of a **clock pulse**.
  - **Asynchronous sequential circuit** (异步时序电路): A system whose behaviour depends upon input signals at any instant of time and the order in which the inputs change.

Outline of This Lecture

Introduction to Sequential Logic

Memory Elements

- Memory Element: Latches
- Memory Element: Flip-Flop

Characteristic Table & Characteristic Equation

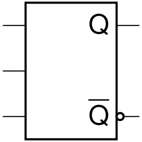
Summary

Memory Elements (记忆元件)

- A **memory element** is a medium in which one bit of information (0 or 1) can be stored or retained until necessary, and thereafter its contents can be replaced by a new value.
- Common memory elements: flip-flop (触发器) (FF) and latch (锁存器)
  - **In common**: two stable states (0 and 1). The state is set as 0 or 1 according to the input signal
  - **Key difference**: level sensitive devices (时钟信号电平敏感的存储单元电路) vs. edge-sensitive devices (时钟信号边沿敏感的存储单元电路)
  - **Various types**:
    - SR latch
    - Controlled SR latch
    - Controlled D latch
    - DFF
    - JKFF
    - TFF

Flip-flops

- The storage elements (memory) used in clocked sequential circuits are called **flip-flops** (触发器).
  - It can have only two states, either the `1` state or the `0` state.
  - The general block diagram representation of a flip-flop is shown below:
    - It has one or more inputs and two outputs.
    - The two outputs are complementary to each other.



  - Normally, the state of Q is called the **state** of the flip-flop, whereas the state of $\bar{Q}$ (or Q') is called the **complementary state** of the flip-flop.

Flip-flops

- The storage elements (memory) used in clocked sequential circuits are called **flip-flops** (触发器).
  - A binary storage device storing one bit of information. Many flip-flops can be used to store as many bits as necessary.
  - In a stable state, the output of a flip-flop is either 0 or 1.
  - Change in state happens only at predetermined intervals dictated by the clock pulses.
  - Loop cut when clock is inactive, no update.
- Flip-flops are **edge-sensitive devices** (时钟信号边沿敏感的存储单元电路).

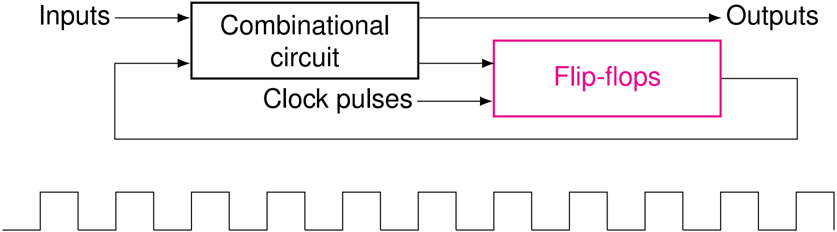
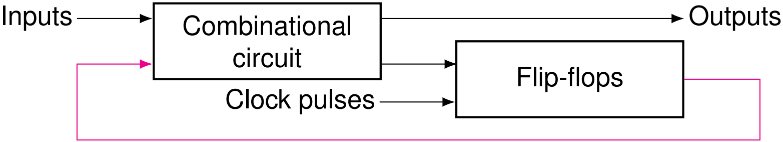


Figure: Block diagram and timing diagram of clock pulses.

Memory Elements

- Memory elements: flip-flop (触发器) and latch (锁存器).
- There has always been considerable confusion over the use of the terms latch and flip-flop.
  - A **flip-flop** is a device which changes its state at times when a change is taking place in the clock signal.
  - A **synchronous latch** is continuously monitoring the input signals, but only can changes its state when a control signal is active.
  - An **asynchronous latch** is continuously monitoring the input signals and changes its state at times when an input signal is changing.



Outline

Introduction to Sequential Logic

Memory Elements

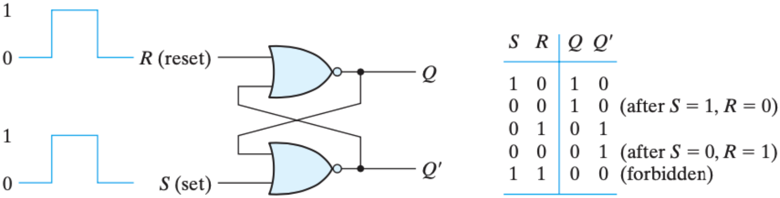
- Memory Element: Latches
- Memory Element: Flip-Flop

Characteristic Table & Characteristic Equation

Summary

SR Latches

- **Set-Reset-Latch (SR latch)** (SR锁存器): a circuit with
  - two cross-coupled `NOR` gates or two cross-coupled `NAND` gates, and
  - two inputs labeled S for set and R for reset.
- Latches are **level sensitive devices** (时钟信号电平敏感的存储单元电路).

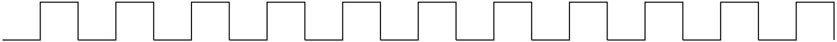


S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	(forbidden)

(a) Logic diagram

(b) Function table

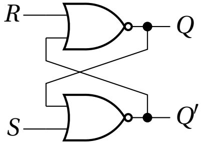
Figure: A SR latch with `NOR` gates. Figure 5.3 in [1].



SR Latch

Stable Table

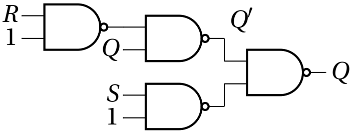
- The **state table** of SR latch is shown below



Present			Next
Q_t	S_t	R_t	Q_{t+1}
0	0	0	0
1	0	0	1
0	0	1	0
1	0	1	0
0	1	0	1
1	1	0	1
0	1	1	X
1	1	1	X

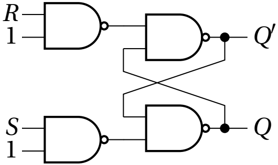
SR Latches

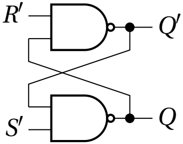
- From the K-map, the **characteristic equations** (特性方程):
  - $Q_{t+1} = S + R'Q_t$ (次态方程): .
  - $SR = 0$ (约束方程).



$S \backslash RQ$	00	01	11	10
0		1		
1	1	1	X	X

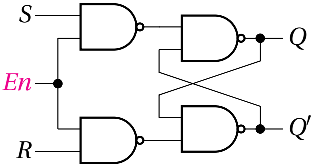
- A more conventional form:





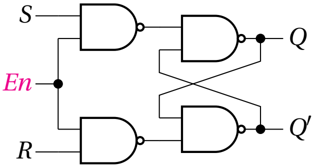
Controlled SR Latch (钟控SR锁存器)

- The transparency of SR latches can be controlled by an additional signal En:
  - If $En = 0$, the outputs of first-level `NAND` gates are always `1`, disabling any changes in the second level gates.
  - If En makes a transition from `0` to `1`, the first-level `NAND` gates are enabled, making the latch active.



Controlled SR Latch (钟控SR锁存器)

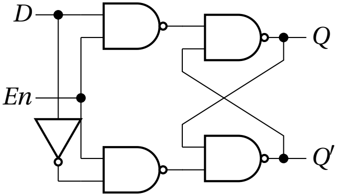
- The transparency of SR latches can be controlled by an additional signal En:
  - If $En = 0$, the outputs of first-level NAND gates are always 1, disabling any changes in the second level gates.
  - If En makes a transition from 0 to 1, the first-level NAND gates are enabled, making the latch active.



- A problem with SR latches: state is indeterminate when S and R are both 1.
  → D latch is designed to handle this problem.

Controlled D Latch (钟控D锁存器)

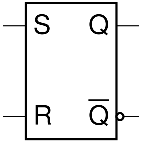
- **D latch**:
  - Also called **transparent latch**, “D” for data.
  - Ensure the previous S and R are never equal to 1 at the same time.



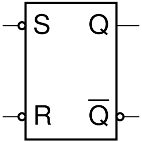
En	D	Next state of Q
0	X	No change
1	0	$Q=0$; reset state
1	1	$Q=1$; set state

Controlled D Latch

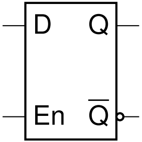
- The controlled D latch has the advantage that it only requires one data input and there is no input condition that has to be avoided.
- Also called transparent latch:
  - Data input is transferred to Q output when enable is asserted. The output follows the input.
  - When the enable input En is at 0, the circuit cannot change state regardless of the value of D. The previous information is stored.



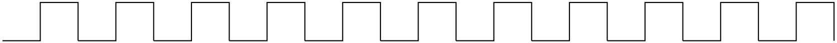
高电平有效SR latch



低电平有效SR latch



高电平有效D latch



Outline

Introduction to Sequential Logic

Memory Elements

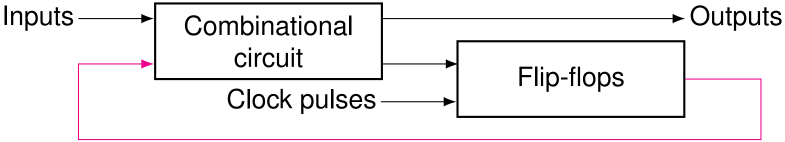
- Memory Element: Latches
- Memory Element: Flip-Flop

Characteristic Table & Characteristic Equation

Summary

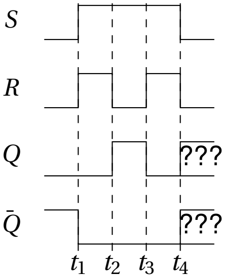
Issue of Latches

- The state of a latch or flip-flop is switched by a change in the control input.
  - This momentary change is called a **trigger** (触发).
  - The transition it causes is said to trigger the flip-flop.
- When latches are used for the storage elements, difficulty arises:
  - State changes as soon as the clock pulse switches to logic-1 level.
    → Unpredictable situation, since the state of the latches may keep changing for as long as the clock pulse stays at the active level.
    → The output of a latch cannot be applied directly or through combinational logic to the input of the same or another latch when all the latches are triggered by a common clock source.



When Latches are Used for The Storage Elements

- State changes as soon as the clock pulse switches to logic-1 level.
  → Unpredictable situation, since the state of the latches may keep changing for as long as the clock pulse stays at the active level.
  → The output of a latch cannot be applied directly or through combinational logic to the input of the same or another latch when all the latches are triggered by a common clock source. ← Cause of the issue.



S	R	Q	$\bar{Q}$	
0	0	No change		
0	1	0	1	
1	0	1	0	
1	1	0	0	(forbidden)

Solutions

- Form a flip-flop
  - Key to the proper operation: trigger it only during a signal transition.
  - How: eliminate the feedback path that is inherent in the operation of the circuit using latches.
  - Two ways of forming a flip-flop:
    1. Produce a flip-flop that triggers only during a signal transition (from 0 to 1 or from 1 to 0) of the synchronising signal (clock) and is disabled during the rest of the clock pulse.
    2. Using two latches: Employ two latches in a special configuration that isolates the output of the flip-flop and prevents it from being affected while the input to the flip-flop is changing.

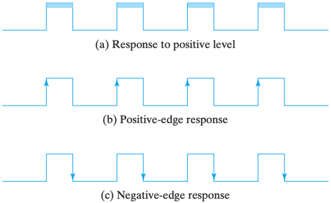
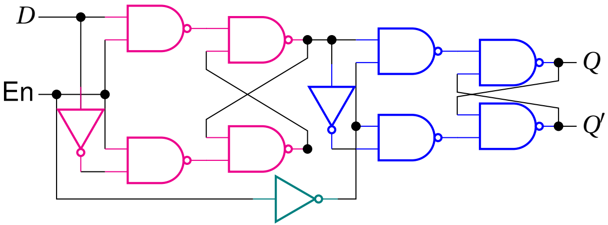

(a) Response to positive level
(b) Positive-edge response
(c) Negative-edge response

Figure: Clock response in latch and flip-flop. Figure 5.8 in [1].

Edge Triggered D Flip-flop

- Negative edge triggered (下降沿触发) **D type master-slave FF** (主从D触发器): a change in the output of the flip-flop can be triggered only by and during the transition of the clock from 1 to 0.
- It consists of a pair of D latches:



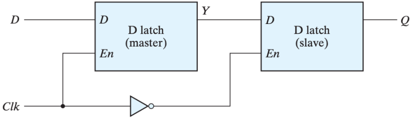
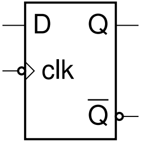


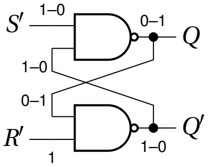
Figure: Master-slave DFF .Figure 5.9 in [1].

- Right is the symbolic diagram of D type master-slave FF (DFF).
- The triangle is termed dynamic input indicator.



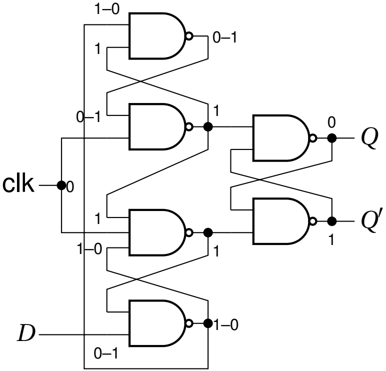
Alternative Configuration of DFF

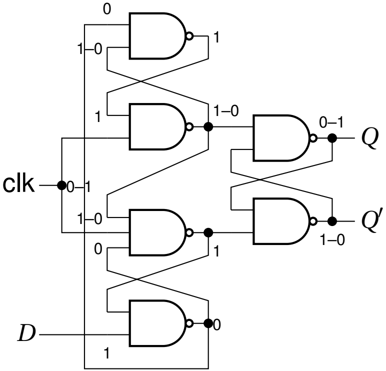
- An alternative configuration of a DFF consists of three pairs of cross-coupled `NAND` gates, each pair constituting a basic S’R’ latch:



  - The latch is stable when $S' = R' = 1, Q = 0$.
  - To change the state, S' must make a `1`→`0` transition.

Alternative Configuration of DFF



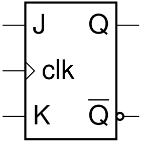


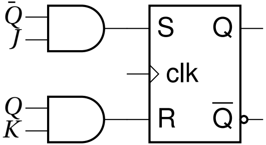
Summary of Edge-triggered DFF

- Edge-triggered DFF are the most economical and efficient FF constructed by interconnecting the various gates, because they require the smallest number of gates.
- Other types of flip-flops can be constructed by using the DFF and external logic.
- Two flip-flops less widely used in the design of digital systems are the JK flip-flops and T flip-flops.

JK Flip-flop

- Latch circuits are not suitable for operation in synchronous sequential circuits because of their transparency.
    - Flip-flops are used as the basic memory elements, which only respond to a transition on a clock input.
    - A typical example is called **JK flip-flop** (JKFF).

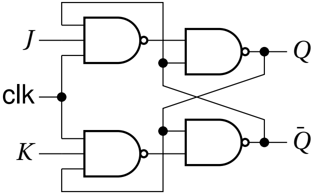




Present			Next
Q_t	J_t	K_t	Q_{t+1}
0	0	0	0
1	0	0	1
0	0	1	0
1	0	1	0
0	1	0	1
1	1	0	1
0	1	1	1
1	1	1	0

JK Flip-flop

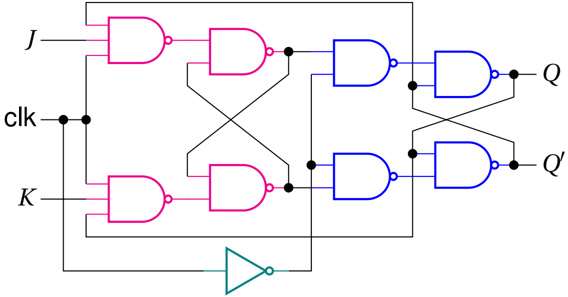
- J is S, and K is R.
- When $J = K = 1$, the flip-flop toggles.
- Combining the two `AND` gates with the SR latch circuit, we have the following reduced circuit:



- However, the above circuit alone still cannot resolve the problem with latches.
  - A master-slave JKFF can be used.

Master-slave JK Flip-flop

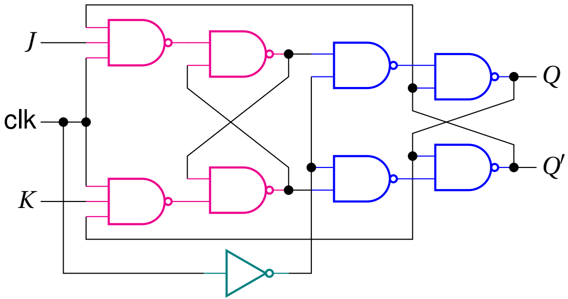
- The **master-slave JKFF** consists of two SR latches, the master and the slave.
  - The master is clocked in the normal way, while the slave clock is inverted:



- When the clock is `1`, the master latch works transparently.
  - However, as the slave latch is disabled at the same time, no changes will be reflected on the output.

Master-slave JK Flip-flop

- The master-slave JKFF consists of two SR latches, the master and the slave.
  - The master is clocked in the normal way, while the slave clock is inverted:



- Upon the clock is changed to 0, the slave is activated to reflect the data from master to the output.
  - However, the master latch is disabled: no further changes on input will be reflected.

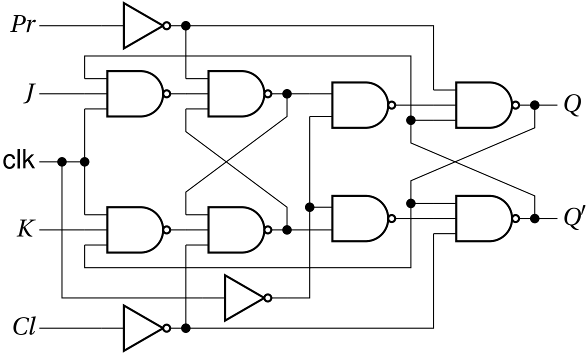
Asynchronous Control

- As well as the J, K, and clock inputs, a master-slave JKFF may also have one or two additional controls to set the state of flip-flop irrespective of clock:
    - These asynchronous controls are usually called **preset** and **clear**.

Cl	Pr	Q
1	1	Forbidden
1	0	0
0	1	1
0	0	X

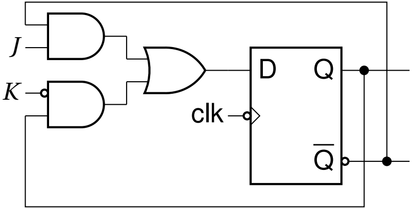
- If $Cl = 1$ and $Pr = 0$ both master and slave are cleared to 0.
- If $Cl = 0$ and $Pr = 1$ the flip-flop is preset to 1.
- Active high on Cl and Pr will override J and K.

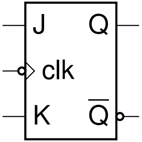
Asynchronous Control



JK Flip-flop Revisit

- The previous DFF can be modified to provide the function of a JKFF as follows:

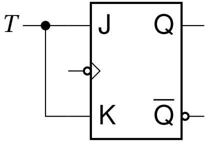


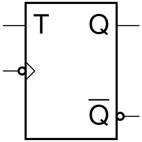


- If $J = K = 1$ and $Q' = 1$, the input to DFF is 1, the Q outputs 1.
- Think what happens when $J = 1$ and $K = 0$, and vice versa.

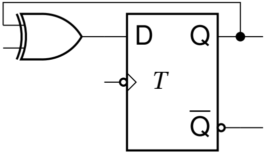
T Flip-flop

- Finally, a **T flip-flop** (TFF) toggles the state when input $T = 1$ upon clock signal.
- It is simple to construct a TFF from JKFF:





- or DFF:



Outline of This Lecture

Introduction to Sequential Logic

Memory Elements

Characteristic Table & Characteristic Equation

Summary

Characteristic Table

- A **characteristic table** describes the logical properties of a flip-flop by describing its operation in tabular form.

J	K	Q_{t+1}	
0	0	Q_t	No change
0	1	0	Reset
1	0	1	Set
1	1	Q'_t	Complement

D	Q_{t+1}	
0	0	Reset
1	1	Set

T	Q_{t+1}	
0	Q_t	No change
1	Q'_t	Complement

Characteristic Equation

- A **characteristic equation** describes the logical properties of a flip-flop by describing its Boolean function.
- DFF: $Q_{t+1} = D$.
- JKFF: $Q_{t+1} = JQ'_t + K'Q_t$.
- TFF: $Q_{t+1} = T \oplus Q_t$.

J	K	Q_{t+1}	
0	0	Q_t	No change
0	1	0	Reset
1	0	1	Set
1	1	Q'_t	Complement

D	Q_{t+1}	
0	0	Reset
1	1	Set

T	Q_{t+1}	
0	Q_t	No change
1	Q'_t	Complement

Outline of This Lecture

Introduction to Sequential Logic

Memory Elements

Characteristic Table & Characteristic Equation

Summary

Summary

- A sequential circuit is specified by
  - a time sequence of inputs,
  - outputs, and
  - internal states (stored by memory elements).
- Memory elements: flip-flops and latches (SR latch, Controlled SR latch, Controlled D latch, DFF, JKFF, TFF).
- Ways to describe a sequential circuit:
  K-map (卡诺图), logic diagram (逻辑电路图), function table (功能表), state table (状态表), state diagram (状态图), characteristic equations (特性方程) (next state equation, 次态方程), excitation table (激励方程).

Essential Reading

- Essential reading for this lecture and next lecture: pages 190-236 of the textbook.

[1] M. M. Mano and M. Ciletti, *Digital design: with an introduction to the Verilog HDL*. Pearson, 2013

Today's lab: 7-seg tube