Latches and Flip-flops

CS207 Chapter 6

James YU
yujq3@sustech.edu.cn

Department of Computer Science and Engineering
Southern University of Science and Technology

Oct. 26, 2022

Sequential logic

- Logic circuits for digital systems:
  - combinational logic (previous lectures),
  - sequential logic.
- Combinational?
  - Output determined by the combination of inputs.
  - Perform an operation specified by a set (combination) of Boolean functions.

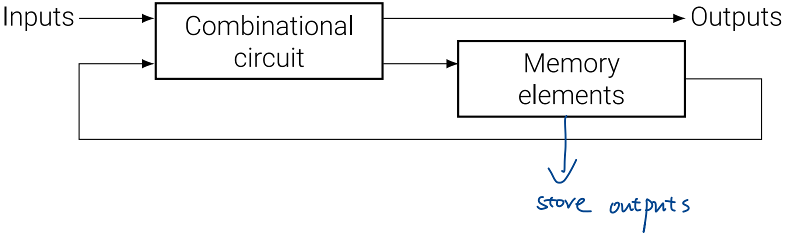
Sequential logic

- Almost all electronic consumer products
  - send, receive, store, retrieve, and process information
  - depends on the ability to store binary — has memory.
- Sequential circuits
  - act as storage elements.
- The logic circuits whose 2 outputs at any instant of time depend on the 1 **present inputs** as well as on the **past outputs** are called *sequential circuits*.

stored

Sequential logic

- The logic circuits whose 2 outputs at any instant of time depend on the 1 **present inputs** as well as on the **past outputs** are called *sequential circuits*.
- A combinational circuit, but with
  - memory elements connected in a feedback path.
  - Outputs are binary functions of not only inputs, but also the present state of the circuit.
  - 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.



Sequential logic

time - related

fundamental

- Sequential circuits are broadly classified into two main categories, known as
  - ① *synchronous* or *clocked sequential circuits*, and
  - ② *asynchronous* or *unclocked sequential circuits*,
  - depending on the timing of their signals.

change at the SAME time

①
- A sequential circuit whose behavior can be defined from **the knowledge of its signal at discrete instants of time** is referred to as a synchronous sequential circuit.
  - The synchronization is achieved by a timing device known as a system clock.
  - The outputs are affected only with the application of a clock pulse.

②
- A sequential circuit whose behavior **depends upon the sequence in which the input signals change** is referred to as an asynchronous sequential circuit.

Sequential logic

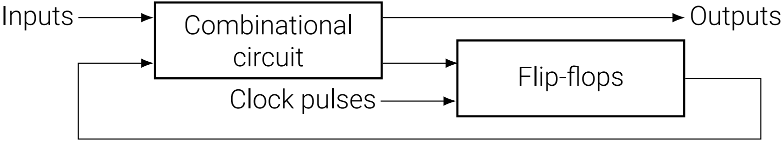
- A sequential circuit whose behavior can be defined from **the knowledge of its signal at discrete instants of time** is referred to as a synchronous sequential circuit.
  - The synchronization is achieved by a timing device known as a system clock.
  - The outputs are affected only with the application of a clock pulse.
- Clock pulse: when changes will happen; Other signals: what changes will happen.
- Clocked sequential circuits:
  - Synchronous sequential circuits that use clock to control.
  - Synchronous because the circuit activity and the updating of storage are synchronized.

  cause to occur or operate
  at the same time or rate

Sequential logic

physical equipment

- Memory element: *flip-flop*.
  - A binary storage device storing one bit;
  - Change in state happens only during clock pulse transition;
  - Loop cut when clock is inactive, no update.


Inputs

Combinational circuit

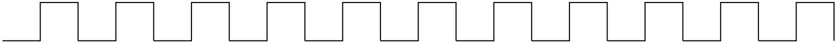
Outputs

Flip-flops

Clock pulses

clock signal

frequency ⇕



Flip-flops

store 1 bit information

- 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.

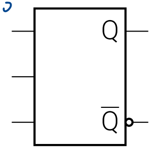
$Q \leftrightarrow f$

output of mux

internal stage

$Q=1.\ Q'=0$

$Q=0.\ Q'=1$



- Normally, the state of Q is called the *state* of the flip-flop, whereas the state of Q' is called the *complementary state* of the flip-flop.

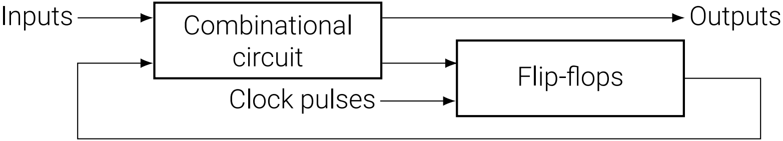
combining in such a way
as to enhance or
emphasize the qualities of
each other or another

Flip-flops apply to latch

- There has always been considerable confusion over the use of the terms *latch* and *flip-flop*.

a circuit that retains whatever output state results from a momentary input signal until reset by another signal.

  - A *flip-flop* is a device which changes its state at times when a change is taking place in the clock signal.
  - An *asynchronous latch* is continuously monitoring the input signals and changes its state at times when an input signal is changing.
  - A *synchronous latch* is continuously monitoring the input signals, but only can changes its state when a control signal is active.



flip-flop is an edge-triggered type of memory circuit while the latch is a level-triggered type. It means that the output of a latch changes whenever the input changes. On the other hand, the latch only changes its state whenever the control signal goes from low to high and high to low.

A flip-flop is a digital memory circuit that stores one bit of data. They are the primary blocks of the most sequential circuits. It is also called one-bit memory, binary, or a bistable multivibrator. Flip-flops act as memory elements in a sequential circuit. You can obtain the output in the sequential circuits using a flip-flop, a combinational circuit, or both.

A flip-flop state repeatedly changes at an active state of the clock pulses. They remain unaffected even when the clock pulse does not stay active. The clocked flip-flops particularly act as the memory elements of the synchronous sequential circuit- while the un-clocked ones (latches) function as the memory elements of asynchronous sequential circuits.

There are mainly four types of flip-flops:

- JK Flip-Flop
- SR (Set-Reset) Flip-Flop
- Data or Delay (D) Flip-Flop
- Toggle (T) Flip-Flop

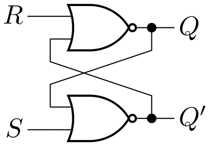
A latch is an electronic device that changes its output immediately on the basis of the applied input. One can use it to store either 0 or 1 at a specified time. A latch contains two inputs- SET and RESET, and it also has two outputs. They complement each other. One can use a latch for storing one bit of data. It is a memory device- just like the flip-flop. But it is not synchronous, and it does not work on the edges of the clock like the flip-flop.

SR latches

$\begin{cases} 1 & I_1 = I_2 = 0 \\ 0 & \text{otherwise} \end{cases}$

- By cross coupling a pair of NOR gates, we have a first latch: *SR latch*.
  - The set and reset inputs are labelled S and R respectively.

set reset



start from left to right to go through all lines

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

↕ truth table

reset (to 0)

set (to 1)

we use subtext t to represent current time

→ head row

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

S=R=0 ⇒ NOT to change Q

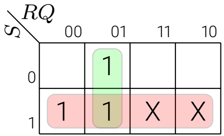
Q prime

Sometimes $Q_t = Q_t' = 1$

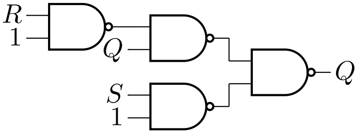
SR latches

NAND gates is MOSTly used in manufacting

- From the k-map: $Q_{t+1} = S + R'Q$.

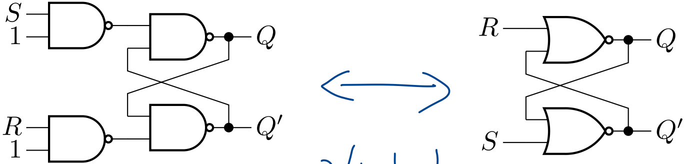


S \ RQ	00	01	11	10
0		1		
1	1	1	X	X



based on or in accordance with what is generally done or believed:

- A more conventional form:



2/1 level

S/R on the top

NAND/NOR gate

Controlled SR latch

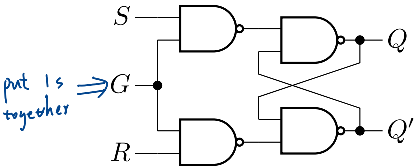
$$\text{NAND}\begin{cases}0 & I_1=I_2=1\\ 1 & \text{otherwise}\end{cases}$$

G

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

$(0\cdot 1)' = 1$

$(1\cdot 1)' = 0$



- A problem with SR latches: **state is indeterminate when S and R are both 1.**

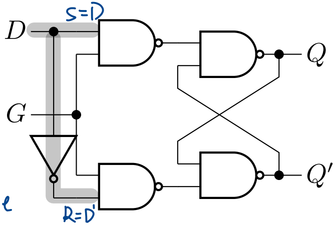
⇒ controlled D latch

Controlled D latch

the state when S=R=1
can't determines
(controlled) SR gates

- D latch is data designed to handle this problem.
  - Also called *transparent latch*, D for data.
  - Ensure the previous S and R are never equal to 1 at the same time.

D=0 ⇒ Q=0
D=1 ⇒ Q=1



enable line

G=0 ⇒ disable
Q will NOT
response
D will NOT
be reflected

Present		Next
D_t	Q_t	Q_{t+1}
0	0	0
0	1	0
1	0	1
1	1	1

S=0
R=1

S=1
R=0

complementary
⇓
S=R is IMpossible

Controlled D latch

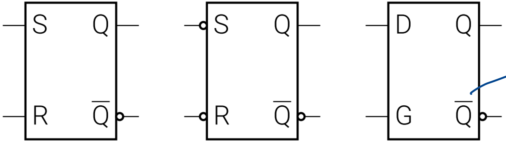
called transparent latch $\Leftarrow \begin{cases} 1 \to 1 \\ 0 \to 1 \end{cases}$

- 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. SR latch needs to forbidden 11
- Transparent?
  - Data input is transferred to Q output when enable is asserted. The output follows the input.

    state a fact or belief confidently and forcefully
  - When enable is de-asserted, the information is stored.

Latches

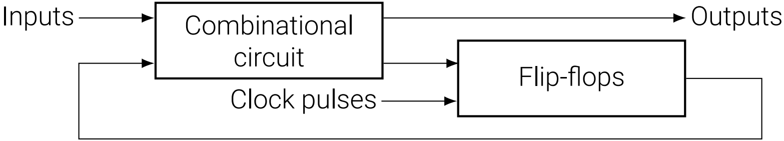
block diagram

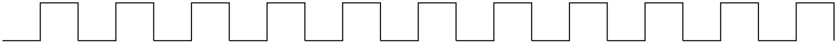


Q prime or Q bar is both okay

Flip-flops invited based on latches

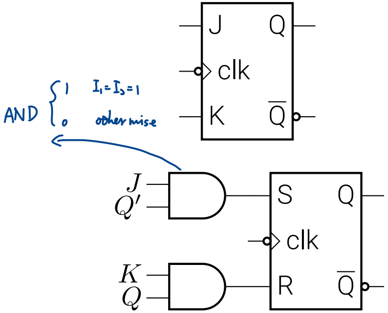
- Latch or flip-flop state change by a control input.
  - The event is called trigger.
- When latches are memories, difficulty arises:
  - State changes as soon as clock switches to 1.
  - Infinite loop during clock-1.





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).



Negative-edge-triggered.

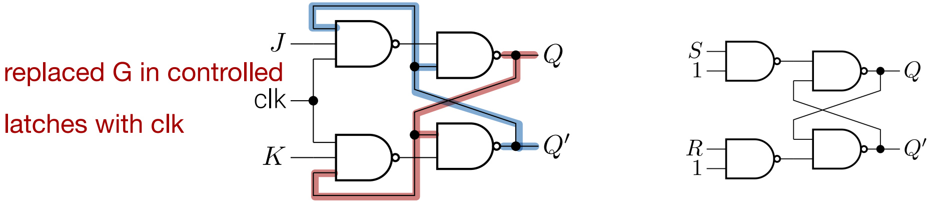
J= K = 0
Q retain

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

JK flip-flop

switch from one effect, feature, or state
to another by using a toggle

- 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:

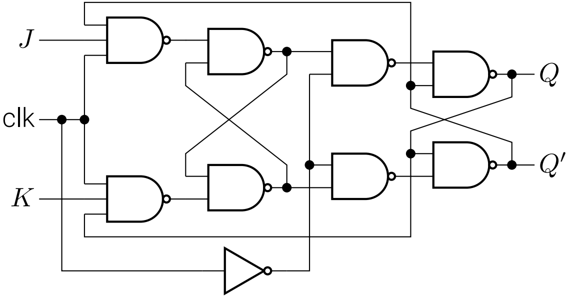


replaced G in controlled latches with clk

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

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:

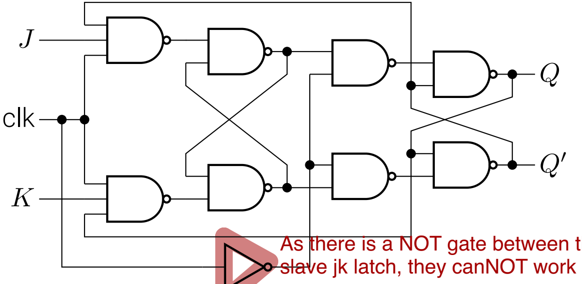


- When the clock is 1, the master latch works transparently.
  - However, as the slave latch is disabled at the same time, no changes will be reflected on the output.

JK flip-flop

The D ff is consist of 2 D latches.

- 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:



As there is a NOT gate between the clk of master ff and slave jk latch, they canNOT work at the same time.

- 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

unlocked sequential

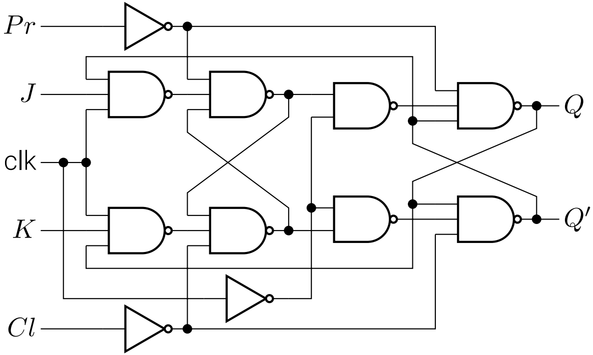
- As well as the J, K, and clock inputs, a master-slave JKFF may also have one or two additional controls to set the state of flip-flop irrespective of clock:
  - These asynchronous controls are usually called *preset* and *clear*.

(of two or more objects or events) not existing or happening at the same time.

clear	preset	
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.

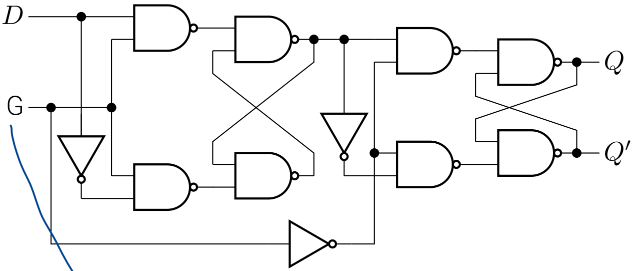
Asynchronous control

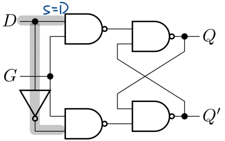


D flip-flop

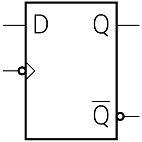
data

- A negative edge triggered D type master-slave FF consists of a pair of D latches:



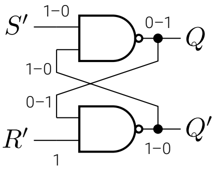


- Right is the symbolic diagram of DFF.
- The triangle is termed *dynamic input indicator*.



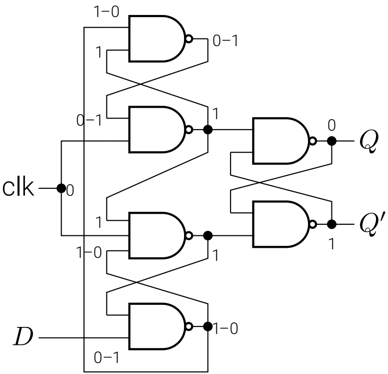
D flip-flop

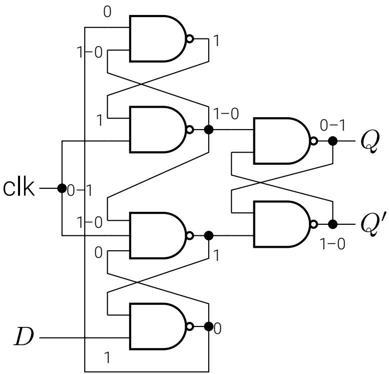
- An alternative configuration of a DFF consists of three pairs of cross-coupled NAND gates, each pair constituting a basic S’R’ latch:



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

D flip-flop

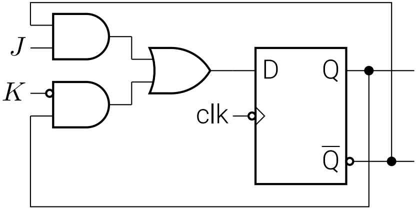

1−0
0−1
1
0−1
1
0
Q
clk
0
1
Q'
1−0
1
1
1−0
D
0−1

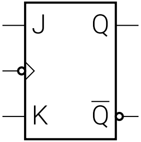

0
1
1−0
1
1−0
0−1
Q
clk
0−1
1−0
Q'
0
1
1−0
0
D
1

JK flip-flop revisit

D flip-flop

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

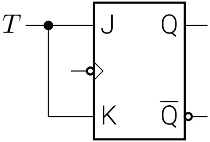


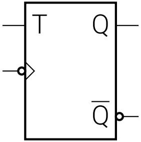


- If $J = K = 1$ and $Q' = 1$, the input to DFF is 1, the Q outputs 1.
- Think what happens when $J = 1$ and $K = 0$, and vice versa.

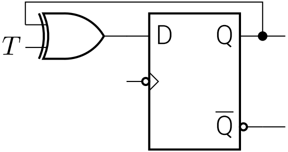
T flip-flop

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





- or DFF:



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.

In some characteristic equations, the j and k and other input may be represent by other variables, but when calculate the equations, do not forget to remove the parenthesis

- DFF: $Q_{t+1} = D$.
- JKFF: $Q_{t+1} = JQ'_t + K'Q_t$.
- TFF: $Q_{t+1} = T \oplus Q_t$.

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

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

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