Lecture 7: Sequential Logic I – Latches and Flip-flops

CS207: Digital Logic

Jialin Liu

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

21 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: Types of Logic Circuits

- "A **combinational circuit** consists of logic gates whose outputs *at any time* are determined from only the present combination of inputs."
  → Combinational logic (Week 5 and Week 6).

- "**Sequential circuits** employ storage elements in addition to logic gates. Their outputs are a function of the inputs and the state of the storage elements."
  - State of the storage elements: a function of previous inputs.
  - Outputs of a sequential circuit depend on: values of present inputs and past inputs.
  - Its behaviour must be specified by a time sequence of inputs and internal states.

  → Sequential logic (Week 7 and Week 8).

Lecture 7: Sequential Logic

Outline of This Lecture

Sequential Circuits

- 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 outputs at any instant of time depend on the present inputs as well as on the past outputs are called **sequential circuits**.

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.

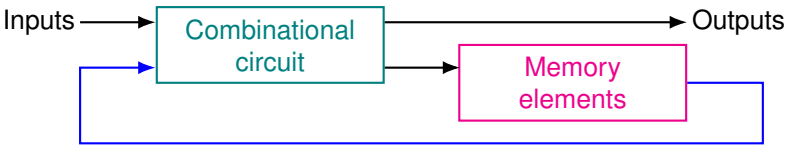


Figure: Block diagram of sequential circuit.

Types of Sequential Circuits

- Sequential circuits are broadly classified into two main categories depending on the timing of their signals.
  - **Synchronous** (同步的) 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**.
    - The outputs are affected only with the application of a **clock pulse**.
  - **Asynchronous** (异步的) sequential circuit: A system whose behaviour depends upon input signals at any instant of time and the order in which the inputs change.

Synchronous Sequential Circuits

- **Synchronous** (同步的) 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**.
  - The outputs are affected only with the application of a **clock pulse** (时钟脉冲).
  - The clock signal is commonly denoted by the identifiers `clock` and `clk`.
- Clock pulse: when changes will happen; other signals: what changes will happen.
- Clocked sequential circuits:
  - Synchronous sequential circuits that use clock pulses to control storage elements.
  - Synchronous because the circuit activity and the updating of storage are synchronised.

Outline of This Lecture

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.
  - 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-flops

- The storage elements (memory) used in clocked sequential circuits are called **flip-flop**s (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 (对时钟信号边沿敏感的存储单元电路).

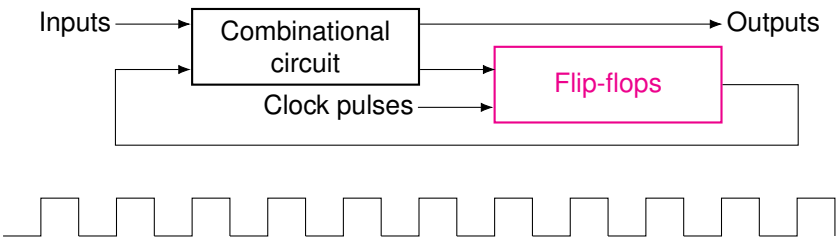
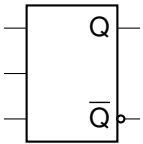


Figure: Block diagram and timing diagram of clock pulses.

Flip-flops

- 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.

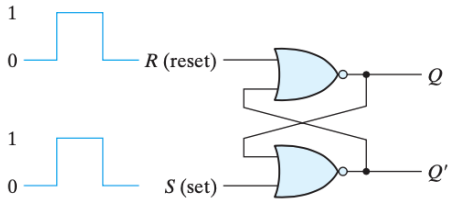


- Normally, the state of Q is called the **state** of the flip-flop, whereas the state of $\bar{Q}$ (or Q') is called the **complementary state** of the flip-flop.

Outline

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 (对时钟信号电平敏感的存储单元电路).



(a) Logic diagram

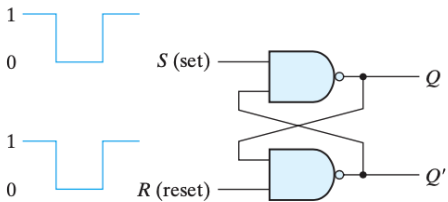
S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	(forbidden)

(b) Function table

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

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 (对时钟信号电平敏感的存储单元电路).



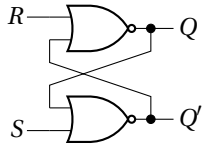
(a) Logic diagram

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

(b) Function table

Figure: A SR latch with `NAND` gates. Figure 5.4 in [1].

SR Latch

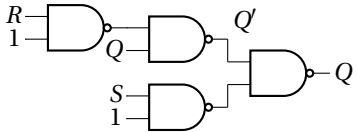


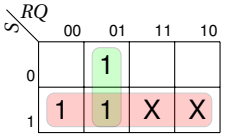
- The **state table** of SR latch is shown below

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	0	X
1	1	1	X

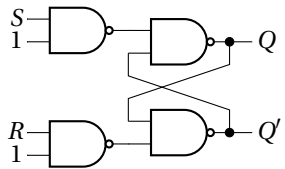
SR Latches

- 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).



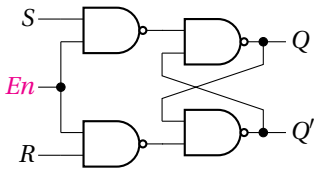


- A more conventional form:



Controlled SR Latch (钟控SR锁存器)

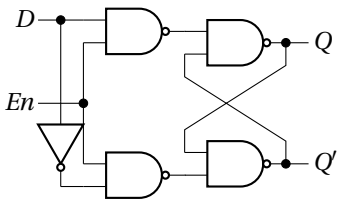
- The transparency of SR latches can be controlled by an additional signal G:
  - If $En = 0$, the outputs of first-level `NAND` gates are always `1`, disabling any changes in the second level gates.
  - If En makes a transition from `0` to `1`, the first-level `NAND` gates are enabled, making the latch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

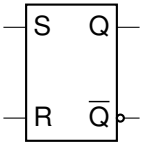


En	D	Next state of Q
0	X	No change
1	0	$Q = 0$; reset state
1	1	$Q = 1$; set state

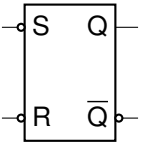
Controlled D Latch

- 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.

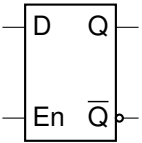
Latches



高电平有效SR latch



低电平有效SR latch

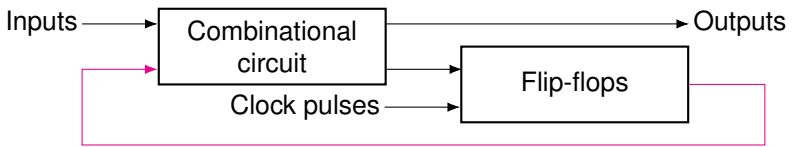


高电平有效D latch

Outline

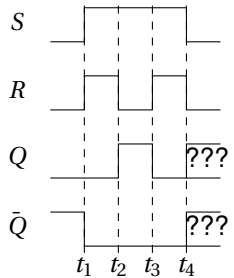
Issue of Flip-flops

- 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.



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

- Flip-flops are constructed as part of a sequential circuit that employs a common clock.
- A latch responds to a change in the level of a clock pulse.
- Solutions:
  - Key to the proper operation of a flip-flop: 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:
    1. 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.
    2. 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.

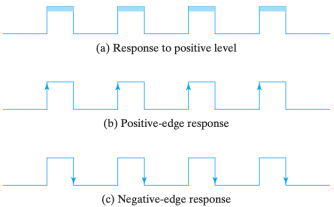
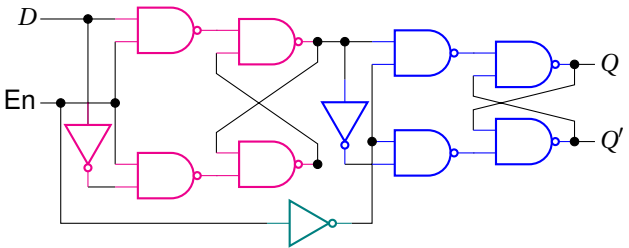


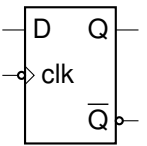
Figure: Clock response in latch and flip-flop. Figure 5.8 in [1].

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:

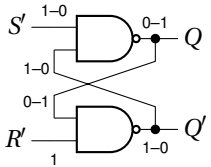


- Right is the symbolic diagram of D type master-slave FF (DFF).
- The triangle is termed **dynamic input indicator**.



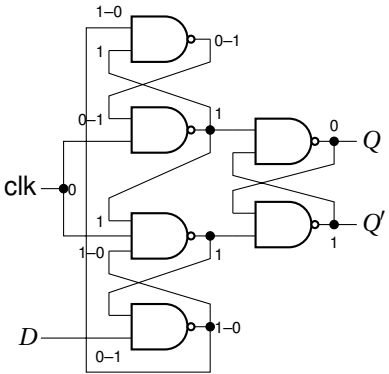
D Flip-flop

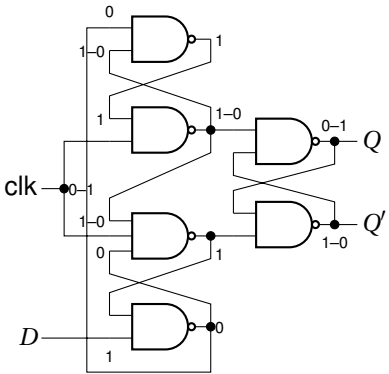
- An alternative configuration of a DFF consists of three pairs of cross-coupled `NAND` gates, each pair constituting a basic S'R' latch:



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

D Flip-flop

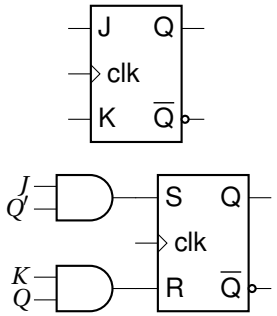

1–0
0–1
1
0–1
1
0
Q
clk
0
1
Q′
1
1–0
1
D
1–0
0–1


0
1
1–0
1
1–0
0–1
Q
clk
0–1
1–0
Q′
1–0
0
1
D
0
1

- Edge-triggered D flip-flops are the most economical and efficient flip-flop 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 D flip-flop 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

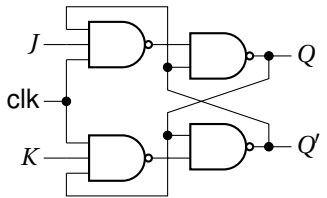
- Latch circuits are not suitable for operation in synchronous sequential circuits because of their transparency.
  - Flip-flops are used as the basic memory elements, which only respond to a transition on a clock input.
  - A typical example is called **JK flip-flop** (JKFF).



Present			Next
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

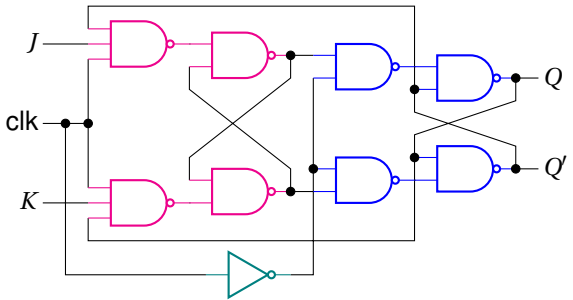
- J is S, and K is R.
- When $J = K = 1$, the flip-flop toggles.
- Combining the two `AND` gates with the SR latch circuit, we have the following reduced circuit:



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

Master-slave JK Flip-flop

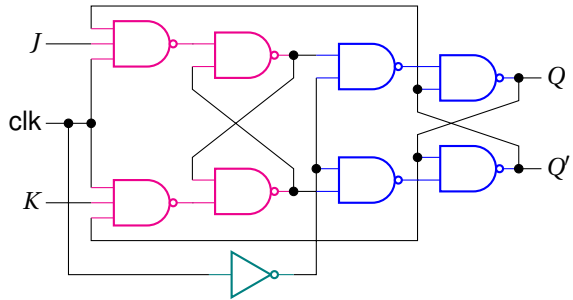
- The **master-slave JKFF** consists of two SR latches, the master and the slave.
  - The master is clocked in the normal way, while the slave clock is inverted:



- 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.

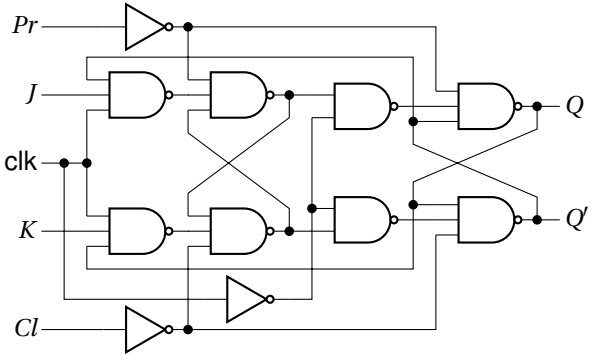
Asynchronous Control

- 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

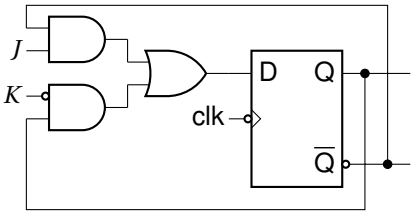
- If $Cl = 1$ and $Pr = 0$ both master and slave are cleared to 0.
- If $Cl = 0$ and $Pr = 1$ the flip-flop is preset to 1.
- Active high on Cl and Pr will override J and K.

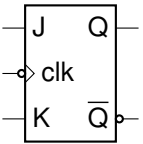
Asynchronous Control



JK Flip-flop Revisit

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

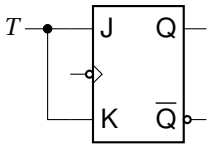


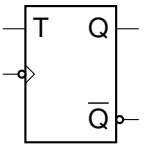


- If $J = K = 1$ and $Q' = 1$, the input to DFF is 1, the Q outputs 1.
- Think what happens when $J = 1$ and $K = 0$, and vice versa.

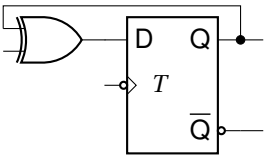
T Flip-flop

- 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:



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

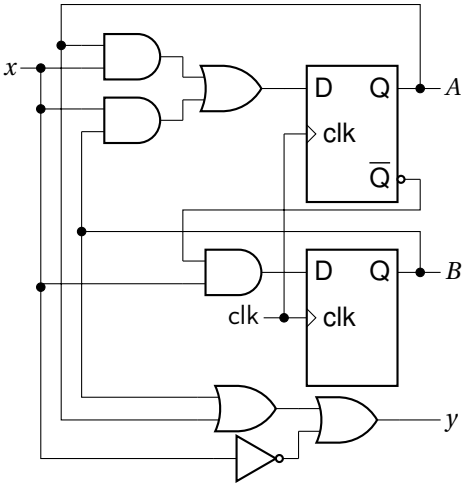
Outline of This Lecture

Analysis of Clocked Sequential Circuits

- Describes what a given circuit will do under certain operating conditions.
  - Obtaining a table or a diagram for the time sequence of inputs, outputs, and internal states.
  - Or obtain the Boolean expressions that describes the behaviour of the circuit.
    → Ways to describe a sequential circuit: Function table, state table, state diagram, next state equation, logic diagram, excitation table, K-map.
- A diagram is a clock sequential circuit, if it includes flip-flops and clock inputs.

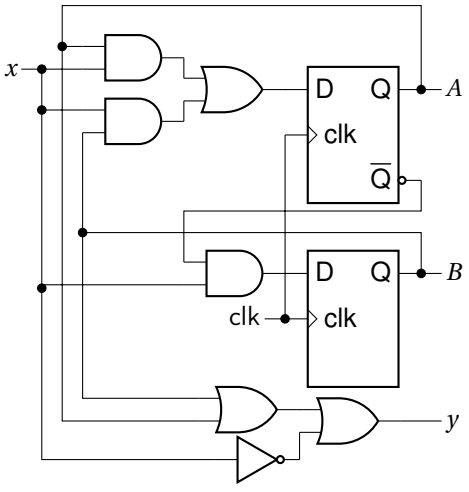
Analysis of Clocked Sequential Circuits

- The behaviour can be described with state equations, or transition equations.



Analysis of Clocked Sequential Circuits

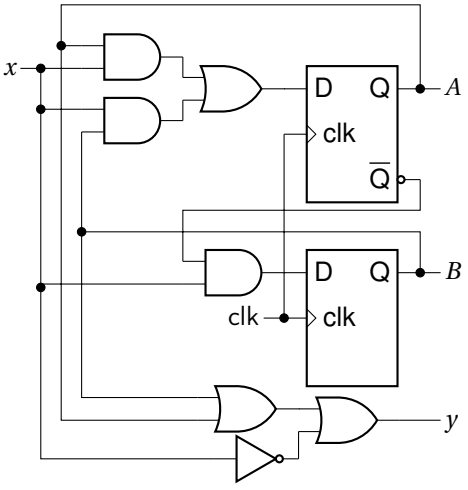
- Two D flip-flops:
  - $A(t+1) = A(t)x(t) + B(t)x(t)$,
  - $B(t+1) = A'(t)x(t)$.
- The output:
  - $y(t) = [A(t) + B(t)]x(t)'$.
  - Since all signals are labeled by t, we can also write $y = (A+B)x'$.



Analysis of Clocked Sequential Circuits

- Time-sequence of inputs, outputs, FFs can be enumerated in a **state table**.

Present		Input	Next		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0



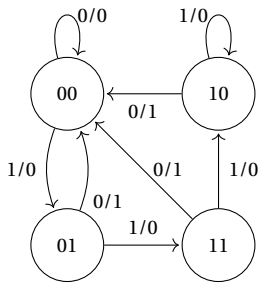
Analysis of Clocked Sequential Circuits

- State table has 2^{m+n} rows for m flip-flops and n inputs, which is very long.
- Or with three sections, with input in the next state and output column.

Present		Next				Output	
		$x=0$		$x=1$		$x=0$	$x=1$
A	B	A	B	A	B	y	y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

Analysis of Clocked Sequential Circuits

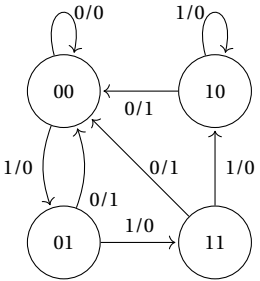
- **State diagram**:
  - Each state as a circle.
  - Transitions between states are directed lines connecting the circles.



Present		Next				Output	
		$x=0$		$x=1$		$x=0$	$x=1$
A	B	A	B	A	B	y	y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

Analysis of Clocked Sequential Circuits

- Circuit diagram → Equations → State table → State diagram.
  - State table is easier to derived from circuit diagram and state equations.
  - State diagram gives a pictorial view of state transitions.
- The shown diagram is a 0-detector.



Analysis of Clocked Sequential Circuits

- The logic diagram of a sequential circuit consists of flip-flops and gates.
- The interconnections among the gates form a combinational circuit and may be specified algebraically with Boolean expressions.
- The part of the combinational circuit that generates external outputs is described algebraically by a set of Boolean functions called **output equations**.
- The part of the circuit that generates the inputs to flip-flops is described algebraically by a set of Boolean functions called **flip-flop input equations** (or, sometimes, **excitation equations**).

Analysis of Clocked Sequential Circuits

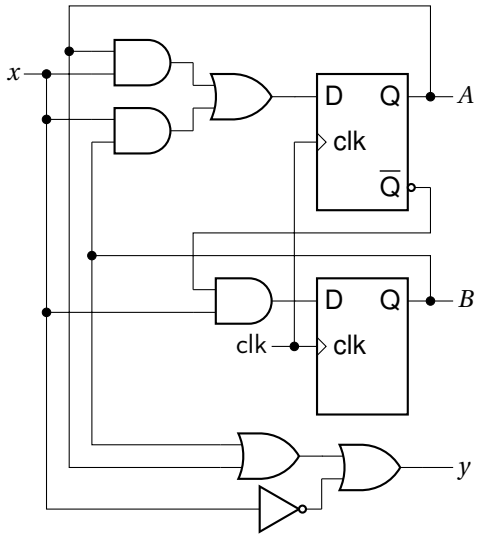
- We will adopt the convention of using the flip-flop input symbol to denote the input equation variable and a subscript to designate the name of the flip-flop output.
  - For example, the following input equation specifies an OR gate with inputs x and y connected to the D input of a flip-flop whose output is labeled with the symbol Q:

$$D_Q = x + y$$

Analysis of Clocked Sequential Circuits

- The sequential circuit consists of two D flip-flops A and B, an input x, and an output y.
- The logic diagram of the circuit can be expressed algebraically with two flip-flop input equations and an output equation:

$$D_A = Ax + Bx,$$
$$D_B = A'x,$$
$$y = (A + B)x'.$$



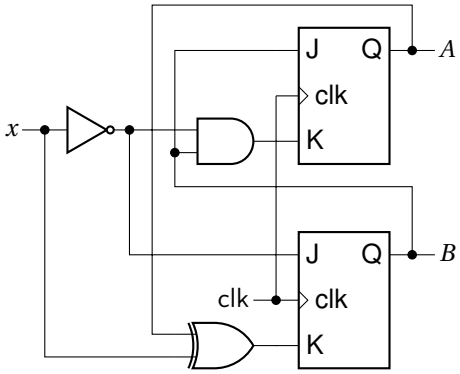
Analysis with JK/T Flip-flops

- JK flip-flops and T flip-flops are different from D flip-flops whose state equation is the same as the input equation.
  - Refer to the corresponding characteristic equation.
- Next-state values can be derived by
  - Determining input equations.
  - Listing binary values for each input equation.
  - Using the corresponding flip-flop characteristic table to determine next state values.

An Example

- Determine input equations:

$$J_A = B,\ K_A = Bx',$$
$$J_B = x'\ K_B = A \oplus x.$$



An Example

- List binary values for each input equation.
- Use the corresponding flip-flop characteristic table to determine next state values.
  - $Q(t+1) = JQ' + K'Q$.
- $A(t+1) = J_A A' + K'_A A = A'B + A(Bx')'$.
- $B(t+1) = J_B B' + K'_B B = x'B + B(A \oplus x)'$.

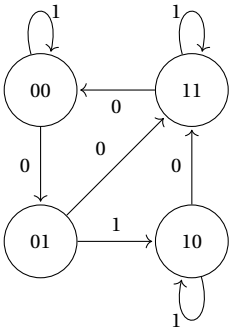
An Example

- $A(t+1) = J_A A' + K_A' A = A'B + A(Bx')'$.
- $B(t+1) = J_B B' + K_B' B = x'B + B(A \oplus x)'$.

Present		Input	Next		FF Inputs			
A	B	x	A	B	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

An Example

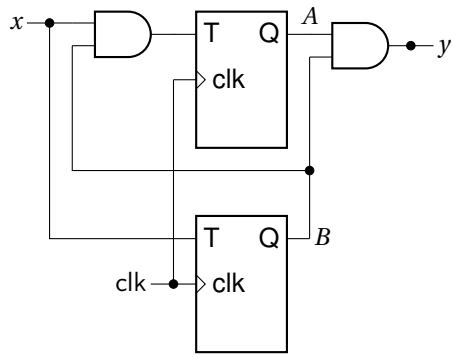
Present		Input	Next		FF Inputs			
A	B	x	A	B	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0



Another Example

- Determine input equations:

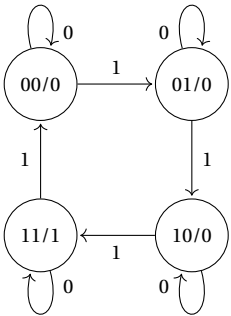
$$T_A = Bx, T_B = x$$
$$y = AB.$$



- List binary values for each input equation.
- Use the corresponding flip-flop characteristic table to determine next state values.
  - $Q(t+1) = T \oplus Q = T'Q + TQ'$.
- $A(t+1) = (Bx)'A + (Bx)A' = AB' + Ax' + A'Bx$.
- $B(t+1) = x \oplus B$.

Another Example

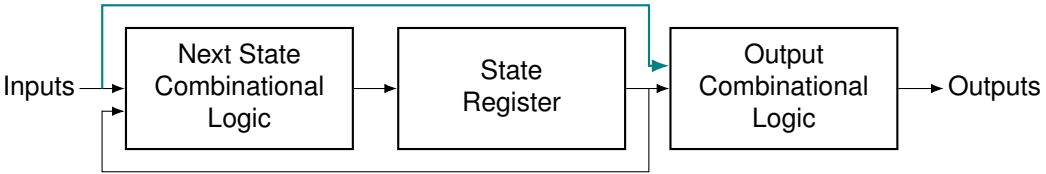
Present		Input	Next		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1



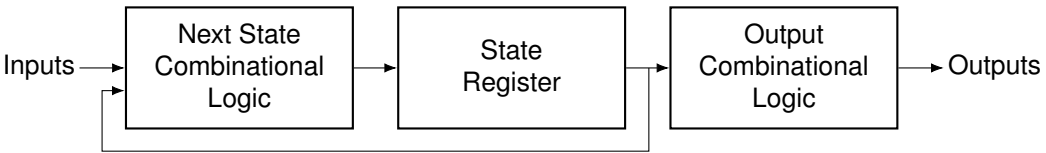
Outline of This Lecture

Finite State Machine

- The most general model of a sequential circuit has inputs, outputs, and internal states.
- It is customary to distinguish between two models of sequential circuits:
- **Mealy model** (米里型电路).



- **Moore model** (摩尔型电路).



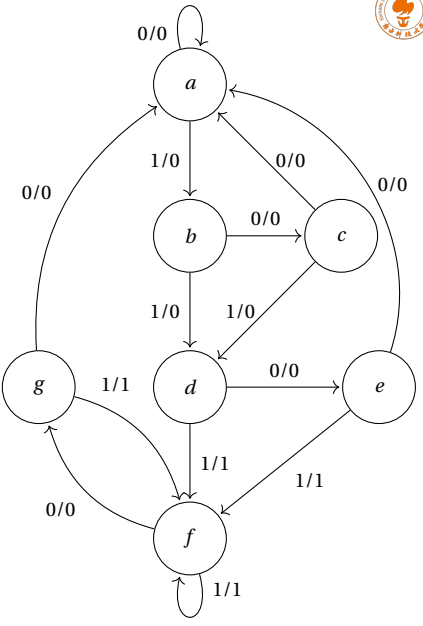
Design of Clocked Sequential Circuits

- The **analysis** of sequential circuits starts from a circuit diagram and culminates in a state table or diagram.
- The **design** (synthesis) of a sequential circuit starts from a set of specifications and culminates in a logic diagram.
- Two sequential circuits may exhibit the same input–output behavior, but have a different number of internal states in their state diagram.
  - In general, reducing the number of flipflops reduces the cost of a circuit.
  - **State reduction** and **state assignment**.

Outline

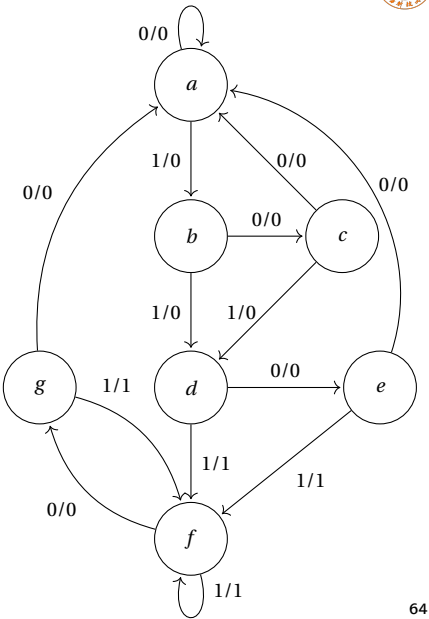
State Reduction

- The reduction in the number of flip-flops in a sequential circuit is referred to as the **state-reduction** problem.
  - Reducing the number of states in a state table, while keeping the external input–output requirements unchanged.
- We illustrate the procedure with an example.



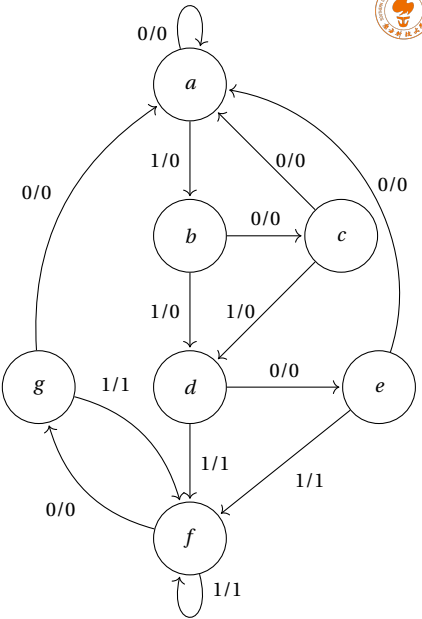
State Reduction

- In our example, only the input–output sequences are important.
  - The internal states are used merely to provide the required sequences.
- There are an infinite number of input sequences that may be applied to the circuit.
  - Each results in a unique output sequence.
  - Example, consider the input sequence `01010110100` starting from state a.



State Reduction

State	Input	Output
a	0	0
a	1	0
b	0	0
c	1	0
d	0	0
e	1	1
f	1	1
f	0	0
g	1	1
f	0	0
g	0	0



State Reduction

- Let us assume that we have found a sequential circuit whose state diagram has fewer than seven states.
  - If identical input sequences are applied to the two circuits and identical outputs occur for all input sequences, then the two circuits are said to be equivalent.
  - One may be replaced by the other.
- The problem of state reduction is to find ways of reducing the number of states in a sequential circuit without altering the input–output relationships.

State Reduction

- First, we need the state table:

Present	Next		Output	
	$x=0$	$x=1$	$x=0$	$x=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

State Reduction

- Two states are said to be **equivalent** if, for each member of the set of inputs, they give exactly the same output and send the circuit either to the same state or to an equivalent state.
  - States g and e are equivalent, and one of these states can be removed.

Present	Next		Output	
	$x=0$	$x=1$	$x=0$	$x=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

State Reduction

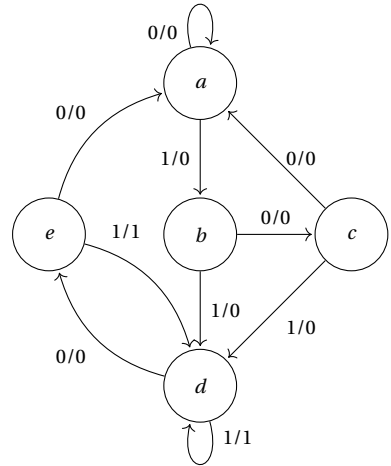
- Now states f and d are equivalent, and state f can be removed and replaced by d.

Present	Next		Output	
	$x=0$	$x=1$	$x=0$	$x=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e (was g)	f	0	1

State Reduction

- Finally we have

Present	Next		Output	
	$x=0$	$x=1$	$x=0$	$x=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1



- The sequential circuit of this example was reduced from seven to five states.

Outline

State Assignment

- In order to design a sequential circuit with physical components, it is necessary to assign unique coded binary values to the states.
- For a circuit with m states, the codes must contain n bits, where $2^n \geq m$.
  - Before the state reduction, we must assign binary values to seven states; the remaining state is unused.
  - If the state table after reduction is used, only five states need binary assignment, and we are left with three unused states.
  - Unused states are treated as don't-care conditions during the design.
- Since don't-care conditions usually help in obtaining a simpler circuit, it is more likely but not certain that the circuit with five states will require fewer combinational gates than the one with seven states.

State Assignment

State	Binary	Gray Code	One-Hot
a	000	000	00001
b	001	001	00010
c	010	011	00100
d	011	010	01000
e	100	110	10000

- **One-hot encoding** usually leads to simpler decoding logic for the next state and output.

Outline of This Lecture

Outline

Design Procedure

- The design of a clocked sequential circuit starts from a set of specifications.
  - Different from combinational circuits, a sequential circuit requires a state table.
- It consists of choosing the flip-flops and combinational gates.
  - Number of flip-flops determined from the number of states.
  - Combinational circuits derived from state table.
- The procedure for designing synchronous sequential circuits can be summarized by a list of recommended steps:
  1. From the word description and specifications of the desired operation, derive a state diagram for the circuit.
  2. Reduce the number of states if necessary.
  3. Assign binary values to the states.
  4. Obtain the binary-coded state table.
  5. Choose the type of flip-flops to be used.
  6. Derive the simplified flip-flop input equations and output equations.
  7. Draw the logic diagram.

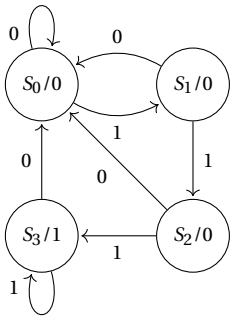
Design Procedure

- The first step is a critical part of the process, because succeeding steps depend on it.
  - We will give one simple example to demonstrate how a state diagram is obtained from a word specification.

Design Procedure

- Suppose we wish to design a circuit that detects a sequence of three or more consecutive 1's in a string of bits coming through an input line.

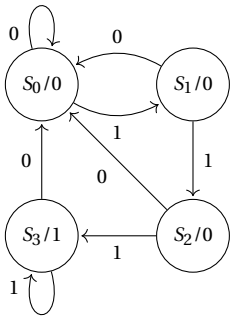
- Starting with state S_0, the reset state.
  - If the input is 0, the circuit stays in S_0.
  - If the input is 1, it goes to state S_1 to indicate that a 1 was detected.
- Starting with state S_1, if the next input is 1, the change is to state S_2 to indicate the arrival of two consecutive 1's, but if the input is 0, the state goes back to $S0$.
- The third consecutive 1 (thus already at state S_2) sends the circuit to state S_3. If more 1's are detected, the circuit stays in $S3$. Any 0 input sends the circuit back to S_0.



Design Procedure

- Once the state diagram has been derived, the rest of the design follows a straightforward synthesis procedure.

Present		Input	Next		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1



Design Procedure

- Once the state diagram has been derived, the rest of the design follows a straightforward synthesis procedure.

Present		Input	Next		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

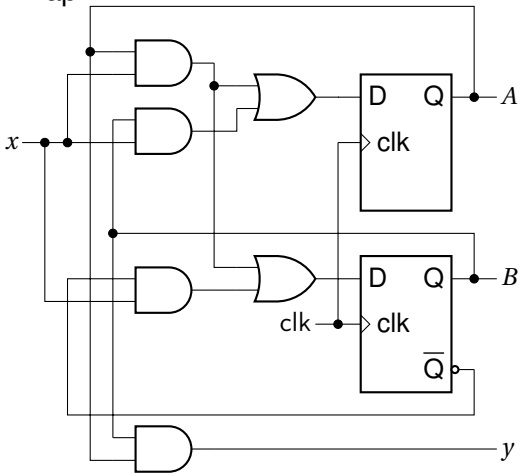
- We use D flip-flop.
- The flip-flop input equations can be obtained:

$$\begin{aligned} A(t+1) &= D_A(A,B,x) \\ &= \sum(3,5,7), \\ B(t+1) &= D_B(A,B,x) \\ &= \sum(1,5,7), \\ y(A,B,x) &= \sum(6,7). \end{aligned}$$

Design Procedure

- The equations are simplified by means of K-map:

$$D_A = Ax + Bx,$$
$$D_B = Ax + B'x,$$
$$y = AB.$$



Outline

Excitation Tables

- The design of a sequential circuit with flip-flops other than the D type is complicated by the fact that the input equations for the circuit must be derived indirectly from the state table.
  - When D-type flip-flops are employed, the input equations are obtained directly from the next state.
  - This is not the case for the JK and T types of flip-flops.
- In order to determine the input equations for these flip-flops, it is necessary to derive a functional relationship between the state table and the input equations.
- During the design process, we usually know the transition from the present state to the next state and wish to find the flip-flop input conditions that will cause the required transition.
- For this reason, we need a table that lists the required inputs for a given change of state: **excitation table** (激励表).

Excitation Tables

- JK flip-flop:

$Q(t)$	$Q(t=1)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

- T flip-flop:

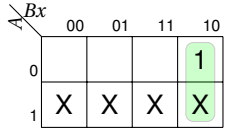
$Q(t)$	$Q(t=1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

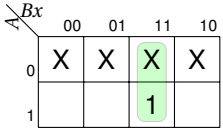
Synthesis using JK Flip-flops

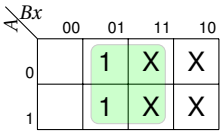
- The manual synthesis procedure for sequential circuits with JK flip-flops is the same as with D flip-flops.
  - Except that the input equations must be evaluated from the present state to the next-state transition derived from the excitation table.

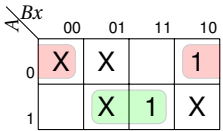
Present		Input	Next		Flip-flip Inputs			
A	B	x	A	B	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

Synthesis using JK Flip-flops



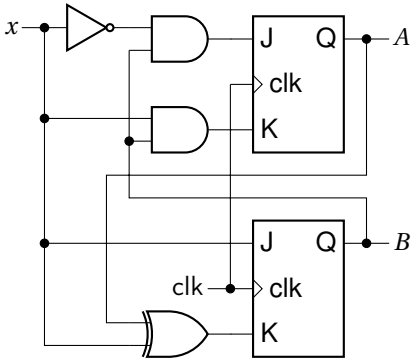






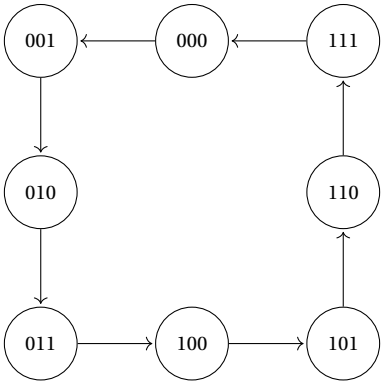
Synthesis using JK flip-flops

$$J_A = Bx',$$
$$J_B = x,$$
$$K_A = Bx,$$
$$K_B = (A \oplus x)'.$$



Synthesis using T Flip-flops

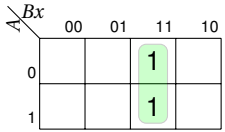
- The procedure for synthesising circuits using T flip-flops will be demonstrated by designing a binary counter.
- An n-bit binary counter consists of n flip-flops that can count in binary from 0 to $2^n - 1$.

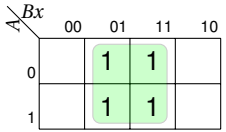


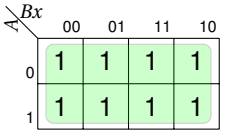
Synthesis using T flip-flops

Present			Next			Flip-flip Inputs		
A_2	A_1	A_0	A_2	A_1	A_0	T_{A2}	T_{A1}	T_{A0}
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

Synthesis using T Flip-flops







$$T_{A2} = A_1 A_0,$$
$$T_{A1} = A_0,$$
$$T_{A0} = 1.$$

Outline of This Lecture

Summary

- Memory elements: flip-flop and latch.
- Finite State Machine.
- Analysis and design of clocked sequential circuits.
- Ways to describe a sequential circuit:
  - Function table, state table, state diagram, next state equation, logic diagram, excitation table, K-map.

Essential Reading

- Essential reading for this lecture: pages 190-236 of the textbook.
- Essential reading for next lecture: pages 255-283 of the textbook.

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