Queen Mary
University of London
Science and Engineering

# EBU4202: Digital Circuit Design
# Sequential Circuits Analysis

Dr. Md Hasanuzzaman Sagor (Hasan)
Dr. Chao Shu (Chao)
Dr. Farha Lakhani (Farha)

School of Electronic Engineering and Computer Science,
Queen Mary University of London,
London, United Kingdom.

# Overview: Sequential Logic Design Principles

* Introduction
* Bistable Elements
* Latches & Flip-Flops
* Analysis Procedure
* Design Procedure

**Chapter 7** – "Digital Design: Principles and Practices" book

# Analysis and Synthesis

**In the context of electronic circuits:**

- **Analysis** means that we have a circuit, and we want to know what is does
- **Synthesis** means that we know what we want a circuit to do, and we then need to *design and realise* (put a design into practice) a circuit to do it.
- In this set of lectures, we will look at the *analysis* process of Synchronous Sequential Circuits and the *design* of Sequential Circuits (limited to Synchronous Sequential Circuits)

# Analysis Procedure: What this is about

- **Behavior of sequential circuits**:
  - Determined from *inputs*, *outputs*, and the *state of flip-flops*.
- **Outputs** and **next state** are both a function of the *inputs* and the *present state*.
- **Analysis consists of**:
  - Obtaining a suitable description that demonstrates the time sequence of *inputs*, *outputs*, and *flip-flop states*.

# Clocked Synchronoues State Machines: Operation

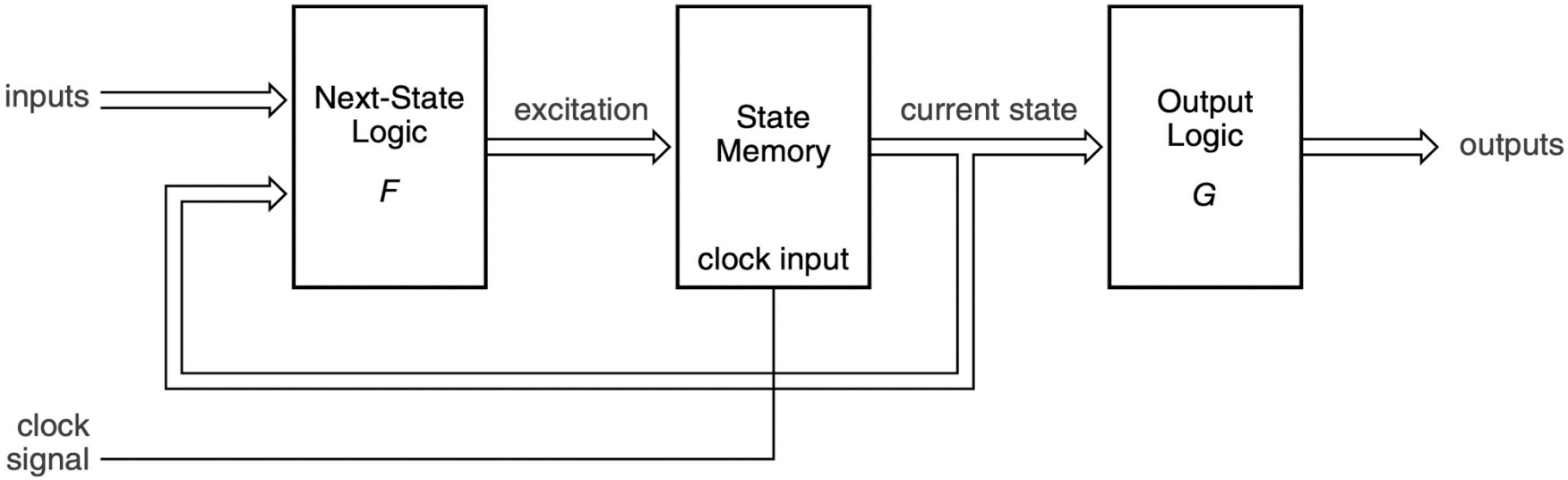
- **Latches & Flip-Flops**: basic building blocks of sequential circuits and can be formally analysed.
- Usually, sequential circuits may not just include latches & flip-flops; they may also include combinational circuits.
- **Clocked Synchronous State Machines**:
  - *State Machine*: general sequential circuit.
  - *Clocked*: their storage elements (i.e., flip-flops) use a clock input.
  - *Synchronous*: all the flip-flops use the same clock signal.
  - Its state changes only when a change occurs on the clock signal.

# Moore Machines

- Moore model for a finite state machine (FSM)
- Outputs only depend on current state

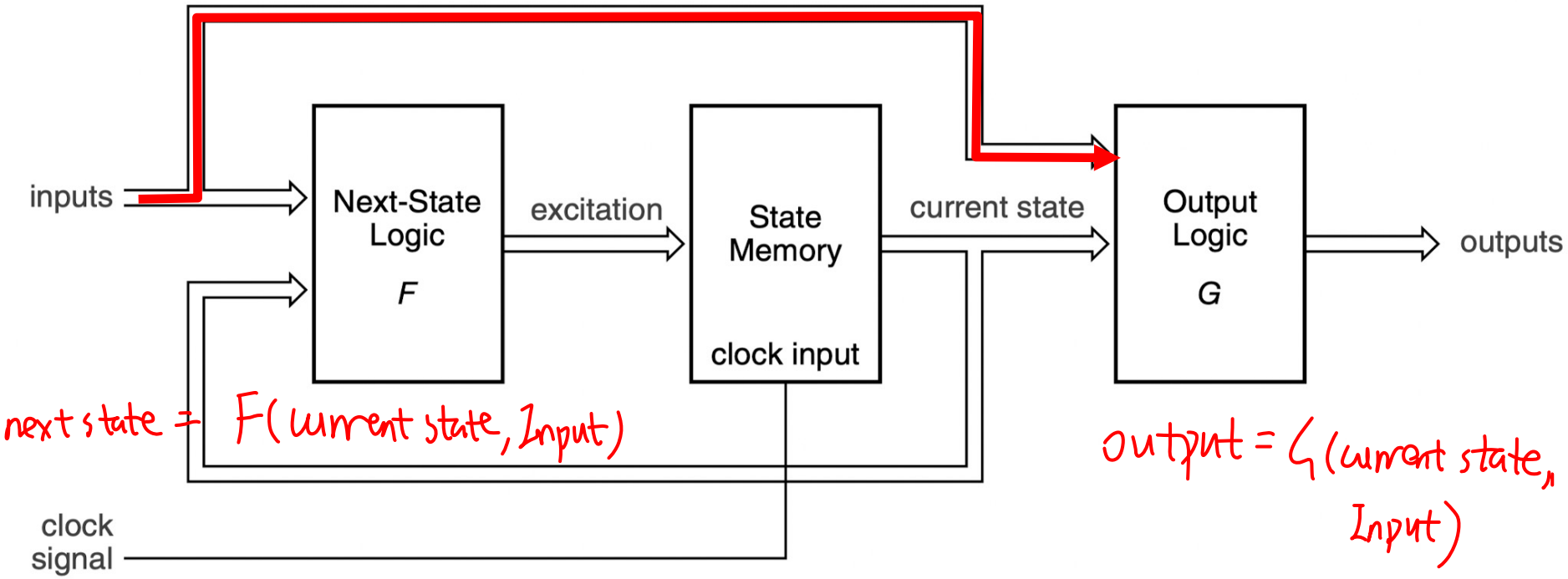
input
output
state

next input 存入 state memory, state memory 输出当前 state



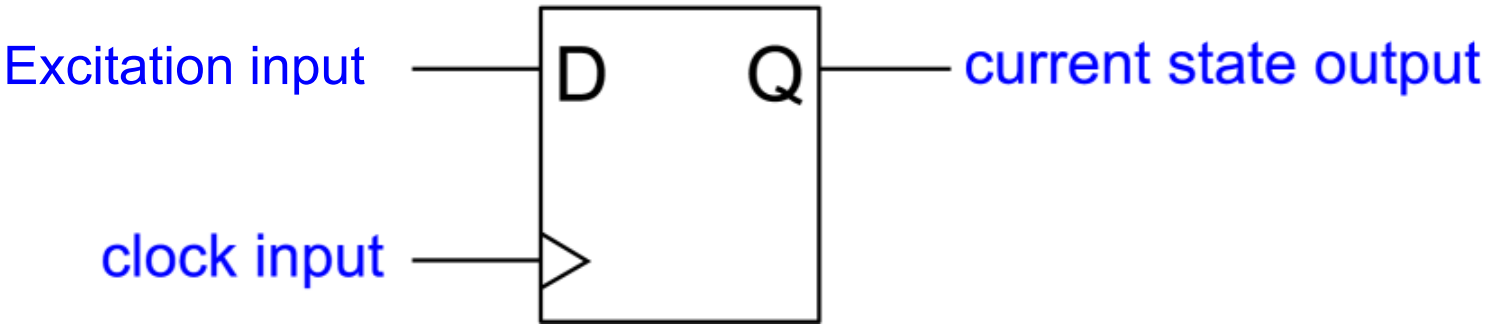
# Mealy Machines

- Mealy model for a finite state machine (FSM)
- Outputs depend on inputs and current state



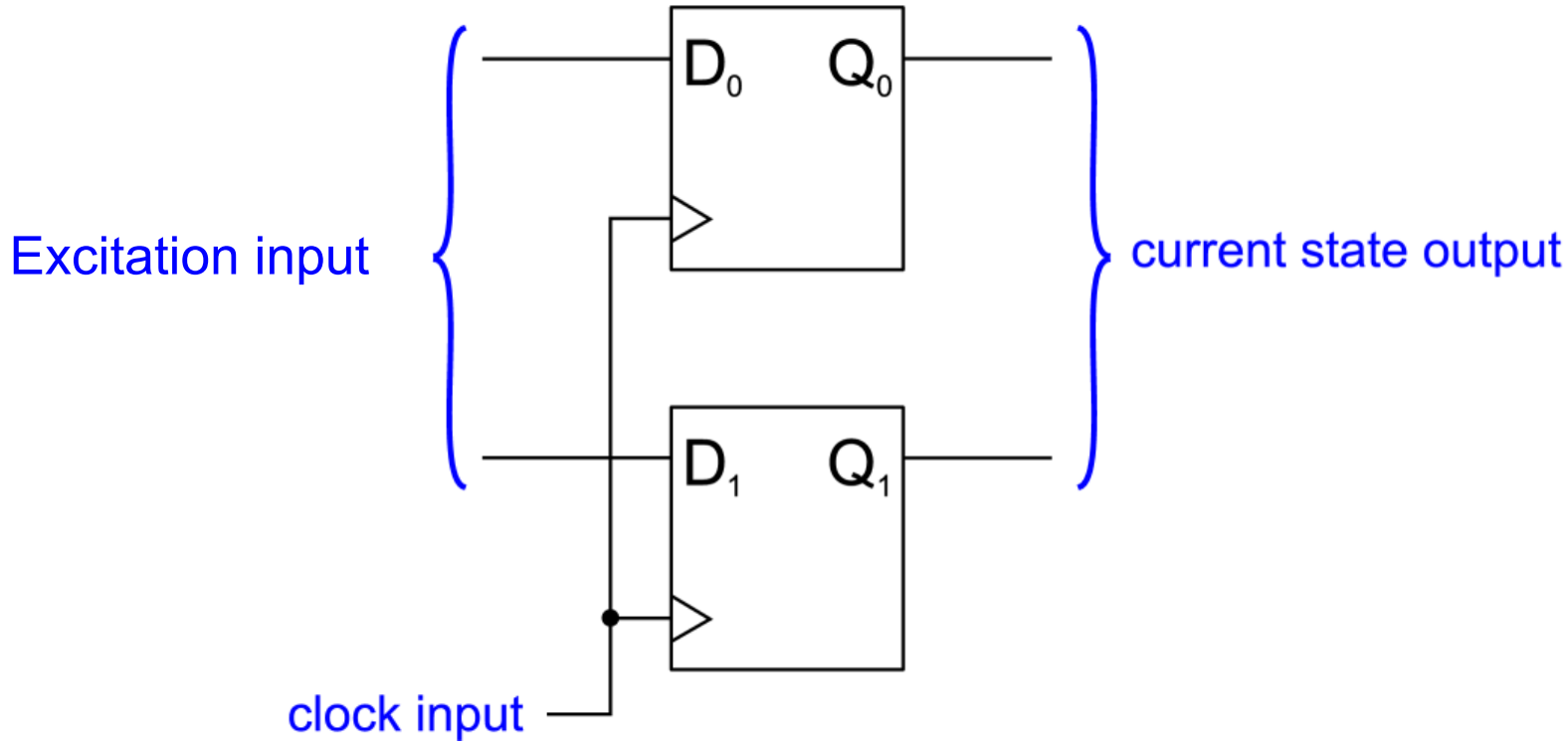
# State Memory/Register

- One D-type flip-flop has how many states?



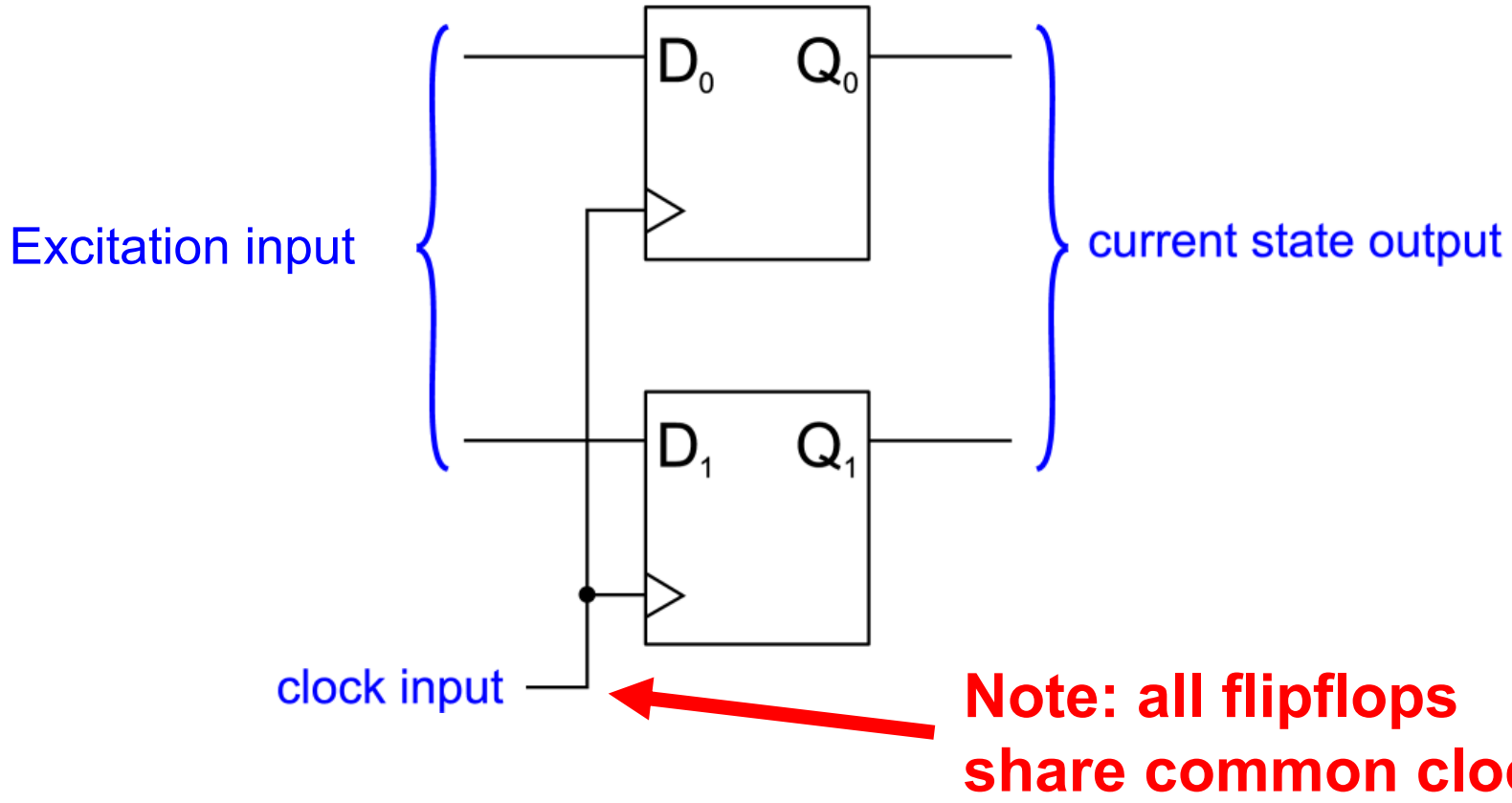
# State Memory/Register

- Two D-type flip-flops have how many states?



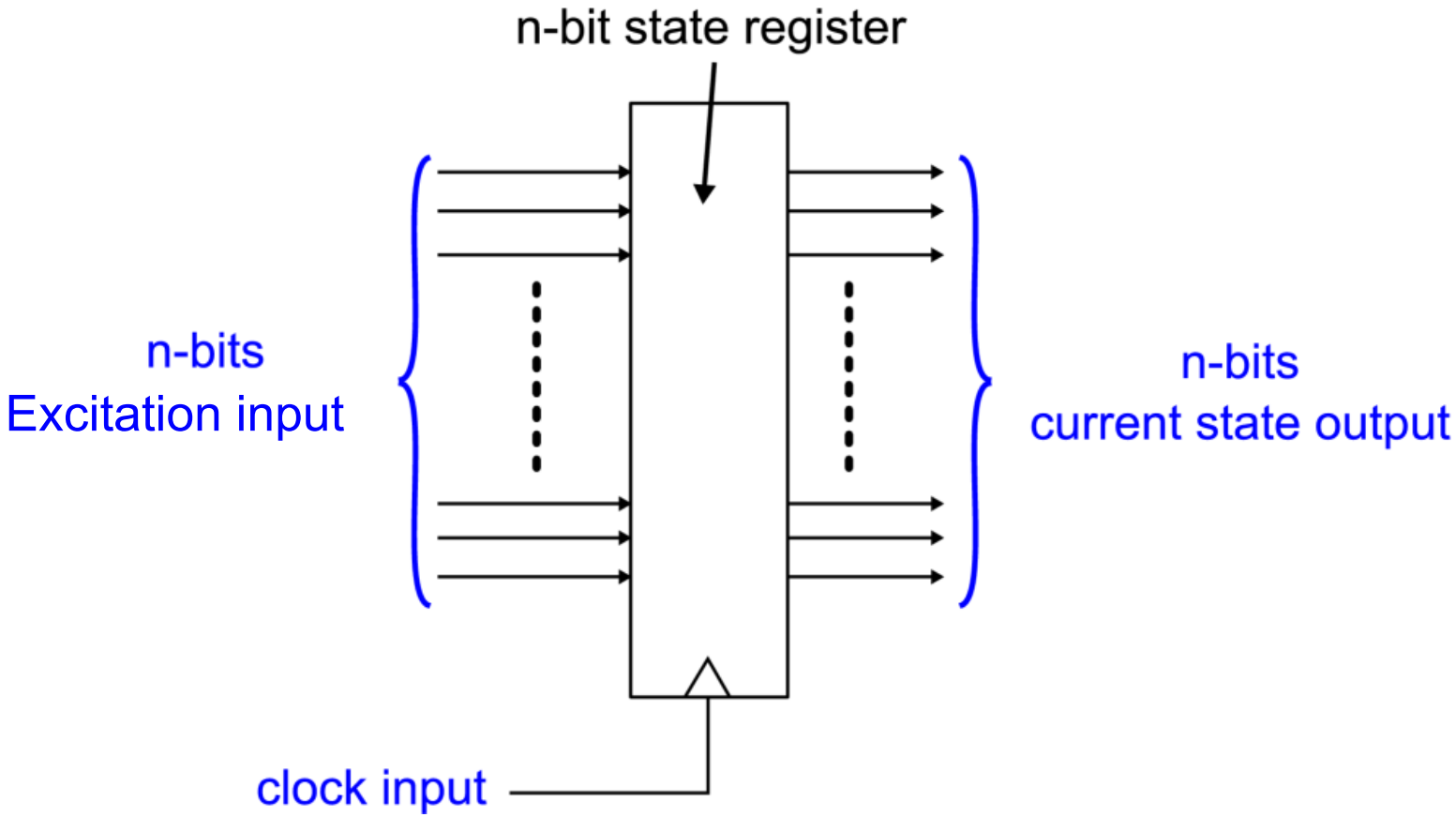
# State Memory/Register

- Two D-type flip-flops have how many states?



# State Memory/Register

- A state register with *n* flipflops has $2^n$ possible states

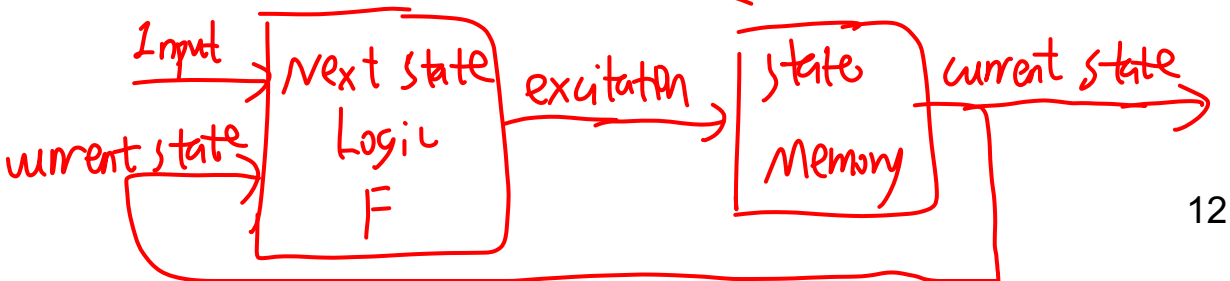



Queen Mary
University of London

# State Machines (1/2)

- A ***state machine*** (or a *clocked synchronous state machine*) consists of three components:

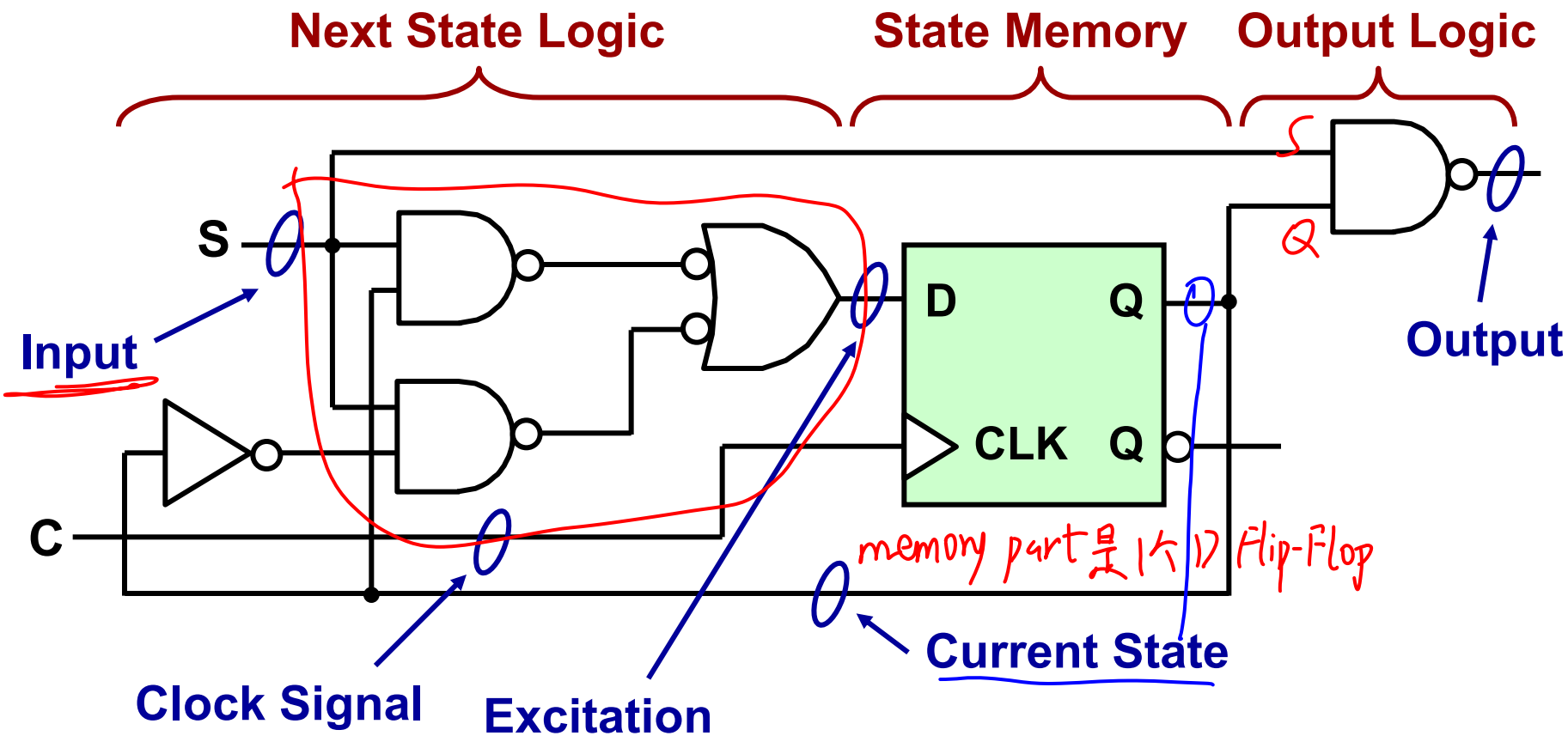
1. **Next-State Logic**:
   - Combinational logic applied to the inputs before being put into the "state memory" (i.e., flip-flops).
   - Expresses the input to the state memory as a boolean equation called either ***Input Equation*** or ***Excitation Equation.***
     - It is a function of the input and the current state i.e., Next State = F(Current State, Input).



# State Machines (2/2)

2. **State Memory**:
   - A set of flip-flops that store the current state.
   - *n* flip-flops can store $2^n$ distinct states.
   - Output of the flip-flop is determined by the characteristic equation for the flip-flop type; also called the ***Characteristic Equation***.
3. **Output Logic**:
   - Determines the output of the state machine.
   - ***Output Equation*** is a function of the input & current state.
     - Output = G(Current State, Input).

# Example: State Machine Diagram




这是1个mealy machine，因为output depend on input and current state

# Characteristic Equations: How they are Derived

- Can derive the characteristic equations by considering the behaviour of the flip-flop/latch, **but these equations**:
  - do not describe in detail the flip-flop/latch timing behaviour.
  - only describe the functional behaviour of the flip-flop/latch upon changes to the control inputs.
- **Examples**:
  - *D (Data) flip-flop* – what goes in comes out. Thus, Q* = D.
  - *T (Trigger) flip-flop* – the next ***Q*** is always the complement of the current ***Q***. Thus, Q* = Q′.

# Characteristic Equations: Definition

- Describe the functional behaviour of a latch or flip-flop.
- Specify the flip-flop's next state as a function of its current state and inputs.
- **Q*** means "*the next value of Q*".

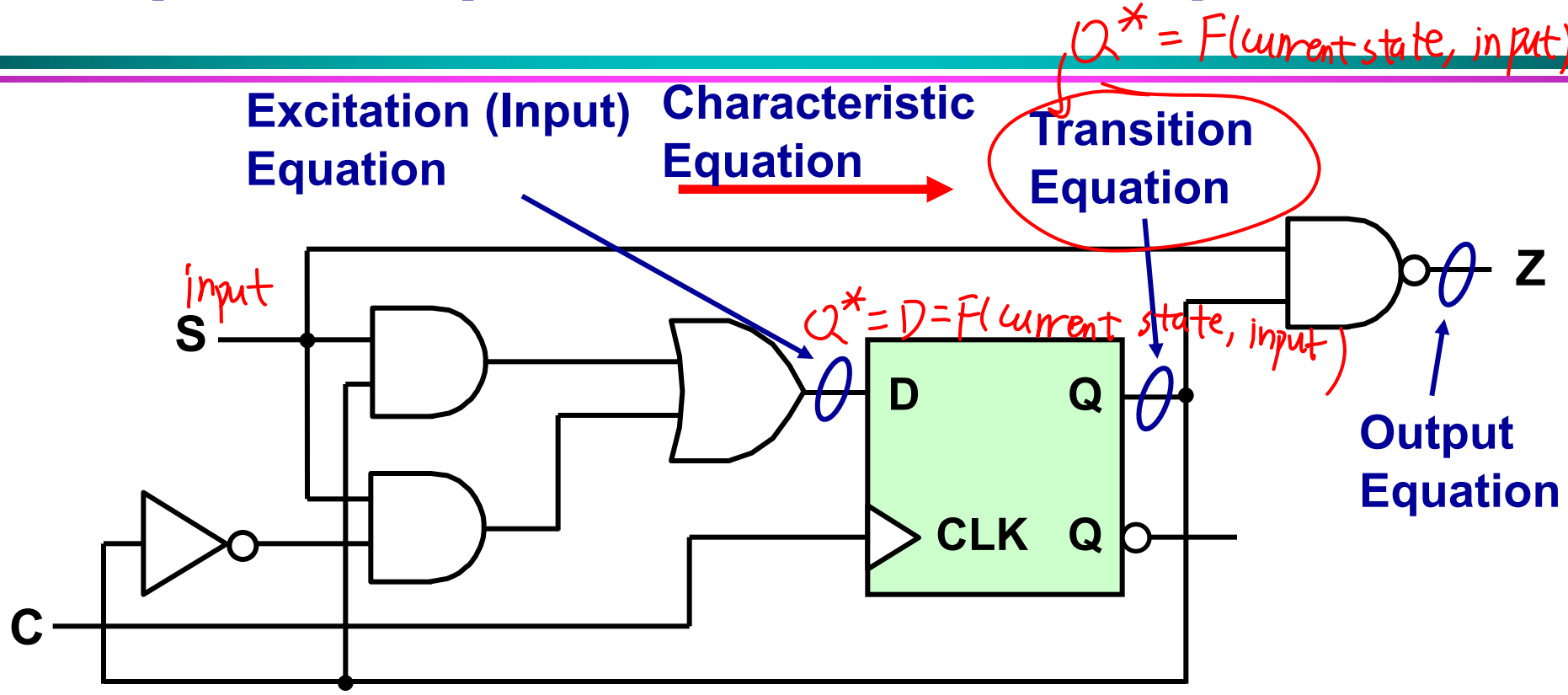
MS = Master/Slave
ET = Edge-Triggered

| *Device Type* | *Characteristic Equation* |
|---|---|
| S-R Latch<br>MS S-R Flip-Flop | $Q^* = S + R'Q$ |
| D Latch | $Q^* = D$ |
| JK Latch<br>MS J-K Flip-Flop<br>**ET J-K Flip-Flop** | $Q^* = JQ' + K'Q$ |
| **ET D Flip-Flop** | $Q^* = D$ |
| D Flip-Flop w/ Enable | $Q^* = E_N D + E_N'Q$ |
| T Flip-Flop | $Q^* = Q'$ |
| T Flip-Flop w/ Enable | $Q^* = E_N Q' + E_N'Q$ |

# Analysis Procedure: Steps

1. Obtain the ***input*** *(*or ***excitation****)* ***equations****.*
2. Obtain the ***output equations***.
3. Obtain the ***next state*** *(*or ***characteristic****)* ***equations****.*
4. Substitute the *input equations* into the *next state equations* to obtain ***transition equations***.
5. Develop a ***transition table*** from the *transition equations*.
6. Develop a ***state table*** that relates the possible states in terms of the *present* and *next state*.
7. Develop a ***state/output table*** that relates the possible states in terms of the *present* and *next state, together with the outputs.*
8. Draw the ***state diagram***.

# *Input, Output & Transition* Equations




What kind of flip-flop is used here?

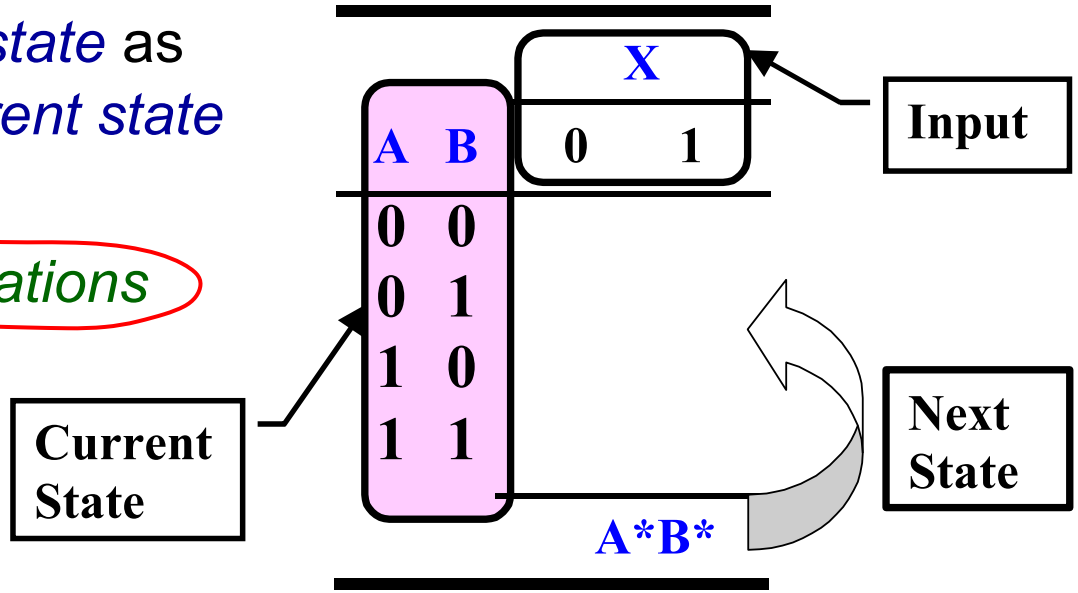
# Table Development Process in CSSMs

- ***Tables of Clocked Synchronous State Machines*** (CSSMs) have up to four sections:
  - **Current state**: shows state of the flip-flops at any given time *t*.
  - **Input**: gives the value of the inputs for each possible state.
  - **Next state**: shows the state of the flip-flops one clock period later, i.e., at *(t+1)*.
  - **Output**: gives the system output values for each present state.

# Table Development Process: Transition Table

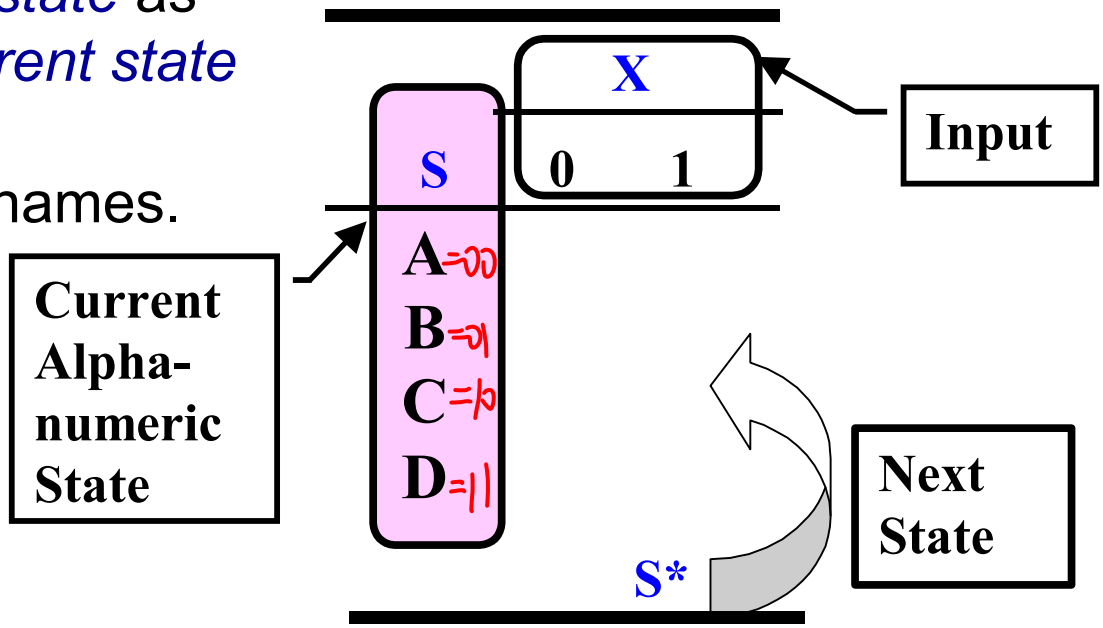
- There are three types of tables to develop:
  - ***Transition Table***
    - Expresses the *next state* as a function of the *current state* and the *input*.
    - Uses *Transition Equations* to determine entries.

$Q^* = D = F(\text{input}, \text{current state})$



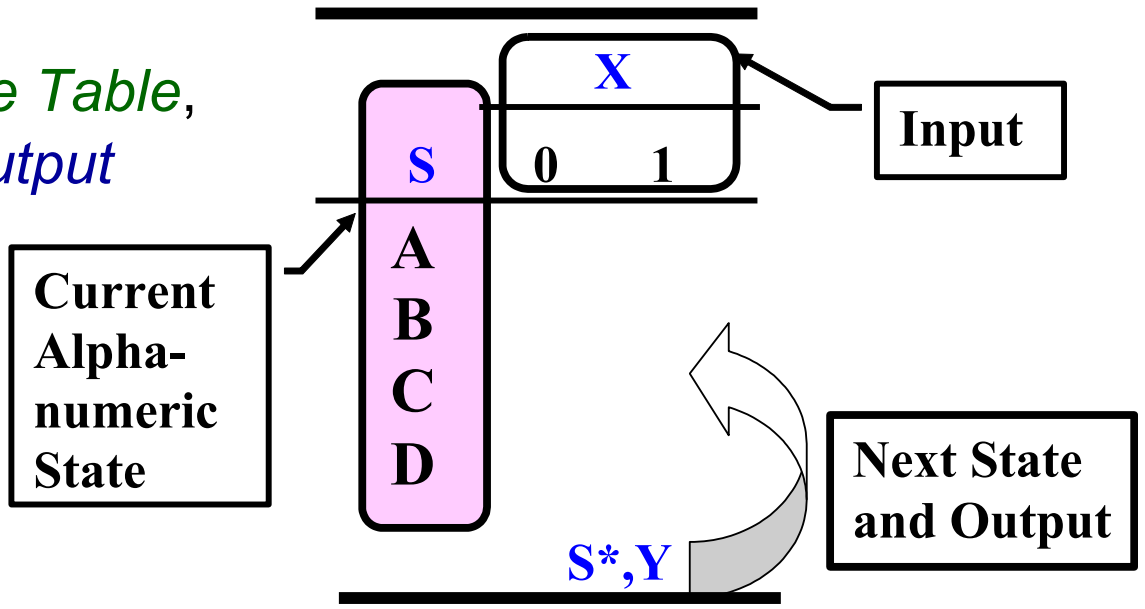
# Table Development Process: State Table

- There are three types of tables to develop (cont.):
  - ***State Table***
    - Expresses the *next state* as a function of the *current state* and the *input* using alphanumeric state names.



# Table Development Process: State/Output Table

- There are three types of tables to develop (cont.):
  - ***State/Output Table***
    - Similar to the *State Table*, but includes the *output* as well.



# Things to remember about tables …

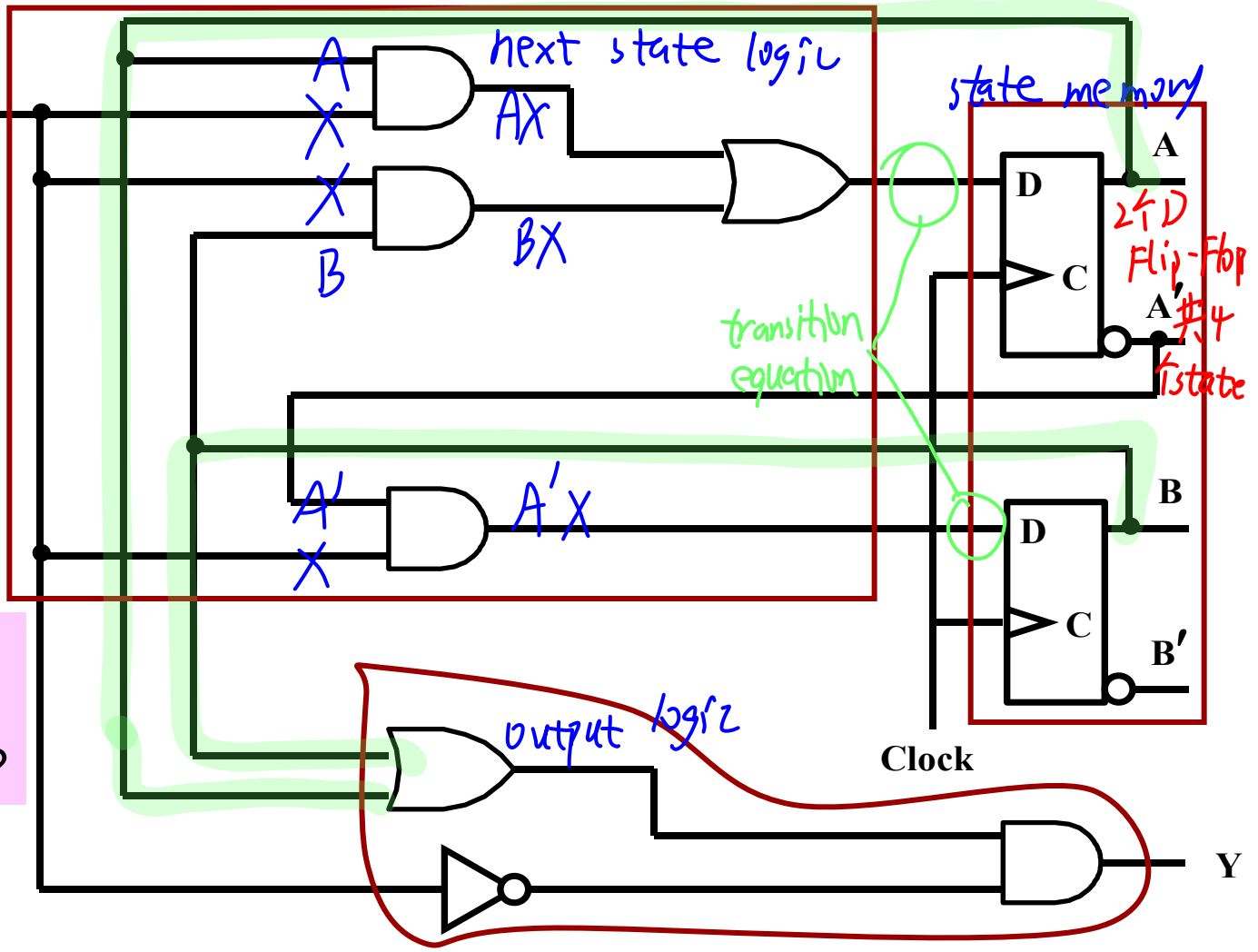
- Always **correctly label your tables** – you will lose marks on your coursework/exam if this is not done!
- Always provide a key to your **alphanumberic states** e.g.:
  - If you have *four states*, each described by *2 bits*, then your states could be labelled as, A = 00; B = 01; C = 10; D = 11. (Try to avoid 'I' and 'O' as labels as they could look like 1 and 0.)
- **State/Output tables** for *Moore Machines* (i.e., where the output depends only on the current state) are simpler than those of *Mealy Machines* (i.e., where the output depends both on the current state and the inputs):
  - Simpler *output equation* as it is not a function of the input(s).

# Analysis Example: State Machine with D Flip-Flops

step 1: identify 3 main part: next state logic, state memory, output logic

What do the areas in **red** represent?

What kind of state machine is this and why?




Queen Mary University of London

# Analysis Example: Table Development Process

step 2: find 3 equation excitation equation, character equation, output equation
对应 next state logic state memory output logic

- *First*, determine the **Excitation (input)** and **output equations**:

**Excitation (Input) Equations**:

$D_A = AX + BX$

$D_B = A'X$

**Output Equation**:

$Y = (A + B)X'$

What is another name for the "Input Equations"?

step 3: calculate transition equation

- How do we determine what the *output* of the D flip-flops is or the *Next State Equations*?
  - Use the **Characteristic Equation** for a D flip-flop: Q* = D
  - So what goes into a D flip-flop comes out again! Thus,

**Transition Equation**:

$A^* = D_A = AX + BX$ (substituting in for $D_A$)

$B^* = D_B = A'X$ (substituting in for $D_B$)

characteristic equation

将 next state 与 current state 相联系

# Analysis Example: Transition Table

step 4: 列每种情况的表

*Transition Equations*:

$A^* = AX + BX$

$B^* = A'X$

*Output Equation*: $Y = (A + B)X'$

| B | A | X=0 | X=1 |
|---|---|---|---|
| 0 | 0 | 00 | 10 |
| 0 | 1 | 00 | 01 |
| 1 | 0 | 00 | 11 |
| 1 | 1 | 00 | 01 |
| | | $B^*A^*$ | |

**Determine B*A* using the Transition Equations for the appropriate flip-flops.**
**For A=0, B=0, X=0:**

$A^* = AX + BX = 0 \bullet 0 + 0 \bullet 0 = 0$

$B^* = A'X = 1 \bullet 0 = 0$

**Thus entry is 00**

**1. List all possible states**

**2. Determine Next State of BA for X=0 and X=1**

以上2列是$B^*A^*$的值

$B^* = A'X$ 对于 B=0, A=1, X=1 ⇒ $B^*=0$

$A^* = AX+BX$ 对于 B=0, A=1, X=1 ⇒ $A^*=1$

Queen Mary University of London

# Analysis Example: State Table

Give each numerical state an alphanumeric name:

**00 = M** **01 = N** **10 = O** **11 = P**

Substitute into the **Transition Table** to form the **State Table**.

M=00
N=01
O=10
P=11

| B | A | X=0 | X=1 |
|---|---|---|---|
| 0 | 0 | 00 | 10 |
| 0 | 1 | 00 | 01 |
| 1 | 0 | 00 | 11 |
| 1 | 1 | 00 | 01 |

B*A*

| S | X=0 | X=1 |
|---|---|---|
| M | M | O |
| N | M | N |
| O | M | P |
| P | M | N |

S*

# Analysis Example: State/Output Table

step 3: 算 output, 画 output table.

– Expand your *State Table* to include the output **Y**, to form the *State/Output Table*:

| | X | |
|---|---|---|
| S | 0 | 1 |
| M | M,0 | O,0 |
| N | M,1 | N,0 |
| O | M,1 | P,0 |
| P | M,1 | N,0 |
| | S*,Y | |

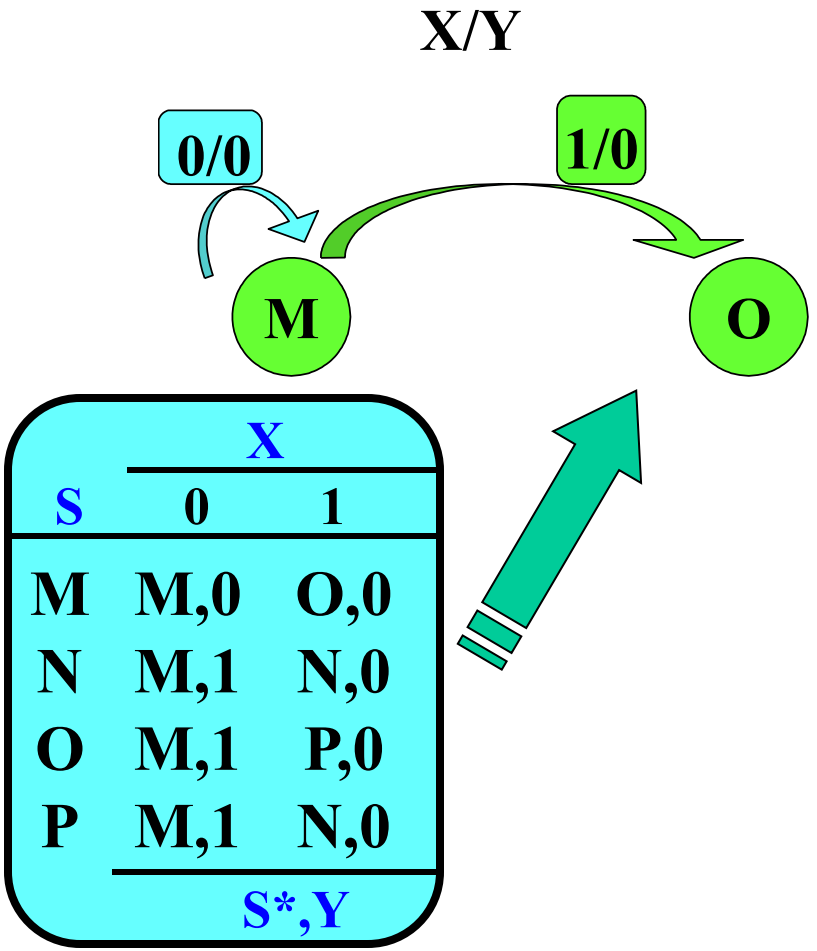
(3.) N=0, X=0 ←

Y=(A+B)X'=1

**Determine Y using the Output Equation.**
**For A=0, B=0, X=0:**
**Y =(A + B)X′**
**=(0 + 0) 1**
**= 0**
**Thus entry is 0.**

# Analysis Example: State Diagram Development Process (1/2)

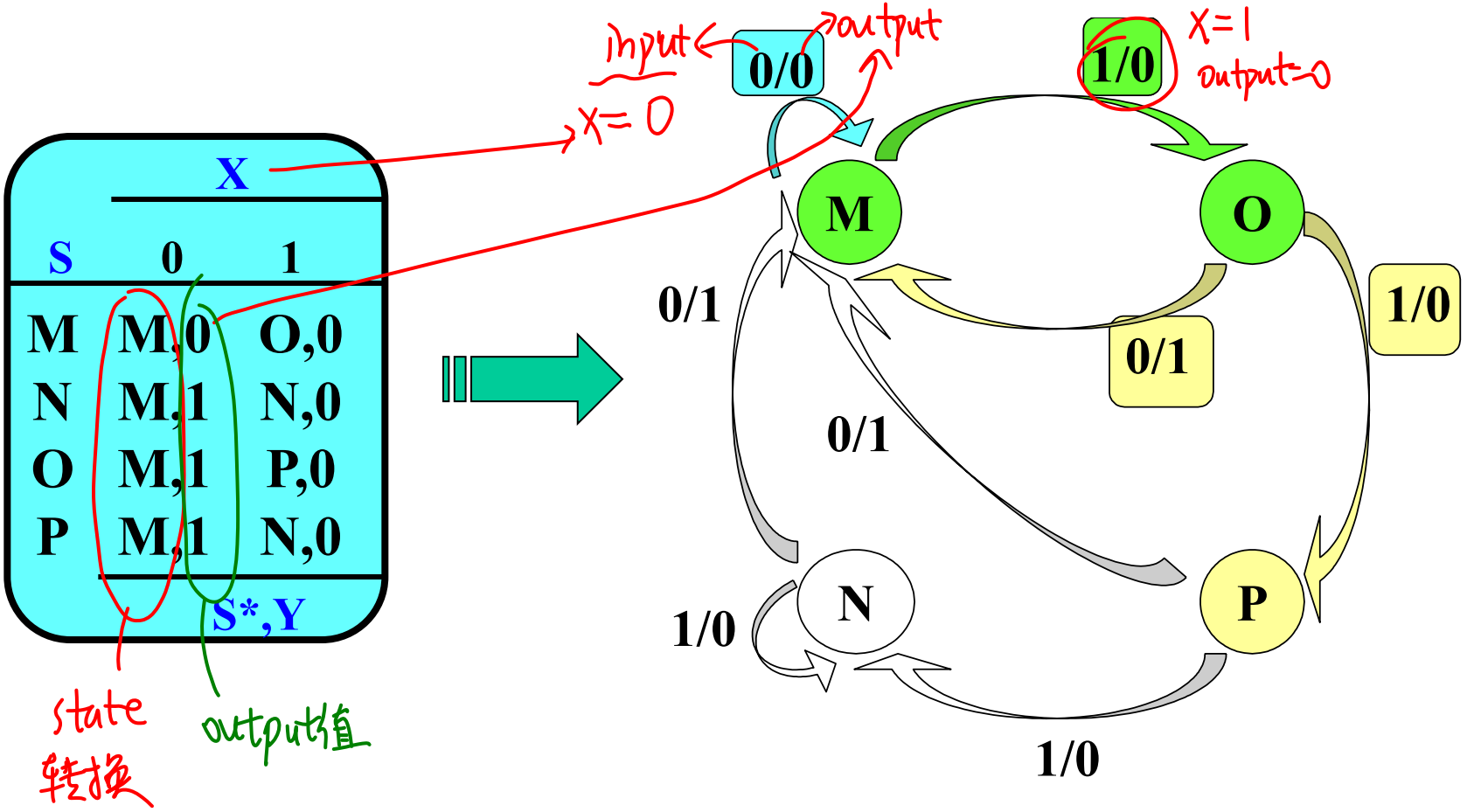
- Development Process:
  - Develop the *State Diagram* from the *State/Output Table*.
  - Represent the *Present States* by circles.
  - Represent the transition between states by *directed lines*.
  - Label the *directed lines* with *input/output values*.




| S | X = 0 | X = 1 |
|---|---|---|
| M | M,0 | O,0 |
| N | M,1 | N,0 |
| O | M,1 | P,0 |
| P | M,1 | N,0 |

S*,Y

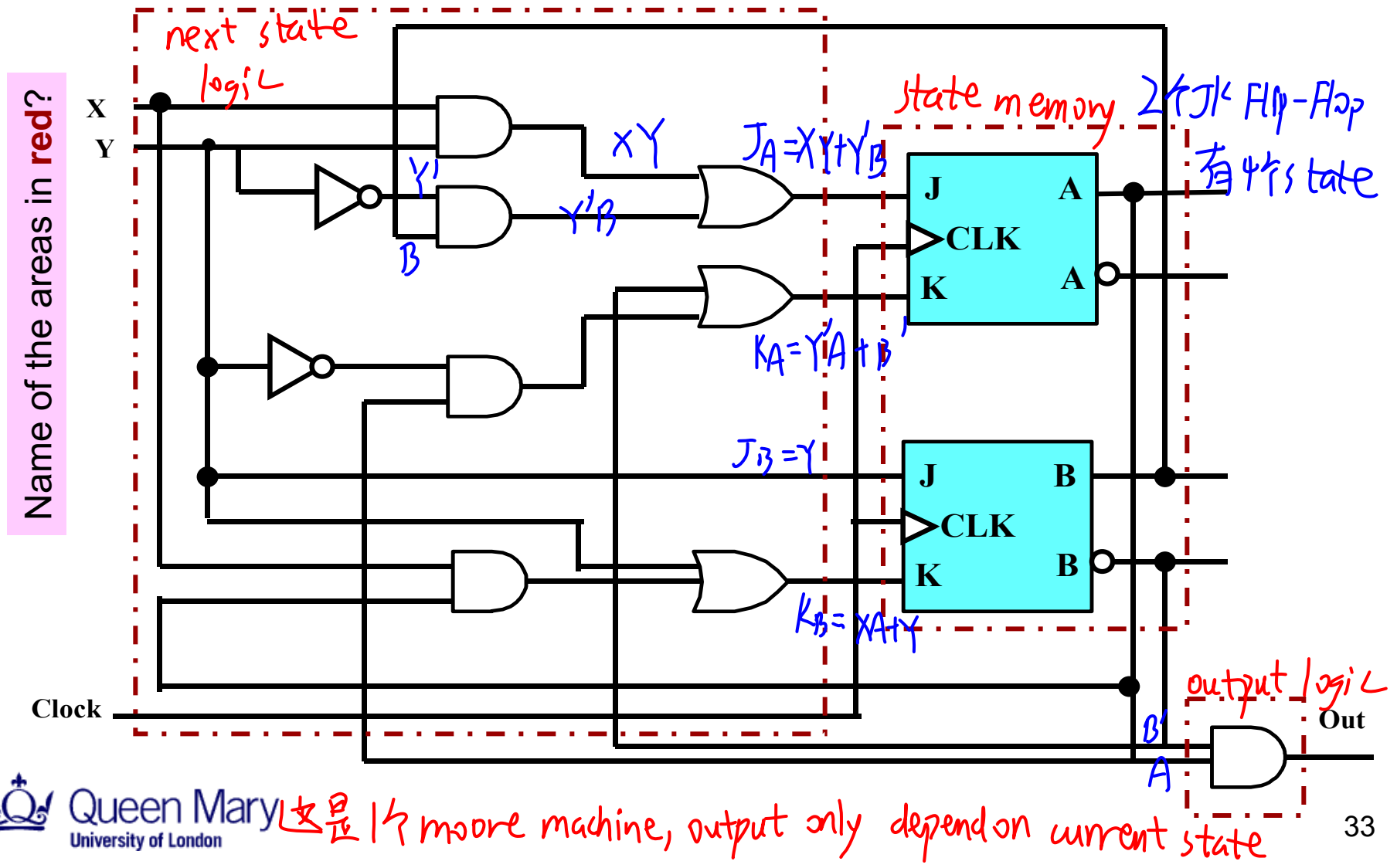
# Analysis Example: State Diagram Development Process (2/2)



# Analysis with JK Flip-Flops

- The analysis example we just went through was only for **D Flip-Flops** with only one data input.
  - The *Next State* values could be obtained directly from the excitation (input) equations!
- What about if there are 2 inputs, i.e., with **JK Flip-Flops**?
  - We need to use the appropriate *Characteristic Equations*!

# Analysis Example: JK Flip-Flop



# JK Flip-Flops Analysis Example: Obtain Input & Output Equations

Reading directly from the diagram:

**Excitation (Input) Equations:**

$J_A = XY + Y'B$ $\quad K_A = Y'A + B'$

$J_B = Y$ $\quad K_B = XA + Y$

**Output Equation:** $OUT = AB'$

---

For *Next State*, use the ***Characteristic Equation*** for a JK flip-flop!

– $Q^* = JQ' + K'Q$ 这是 JK Flip-Flop 公式

**Transition Equations:**

$$A^* = J_A A' + K_A' A$$
$$= (XY + Y'B)A' + (Y'A + B')'A = XYA' + Y'BA' + ABY$$

$$B^* = J_B B' + K_B' B$$
$$= YB' + (XA + Y)'B = YB' + X'Y'B + Y'A'B$$

# JK Flip-Flops Analysis Example: Transition Table

Build transition table

- *Transition* equations:

$A^* = XYA' + Y'A'B + ABY$

$B^* = YB' + X'Y'B + Y'A'B$

2 input

| B | A | XY = 00 | XY = 01 | XY = 10 | XY = 11 |
|---|---|---|---|---|---|
| 0 | 0 | 00 | 10 | 00 | 11 |
| 0 | 1 | 00 | 10 | 00 | 10 |
| 1 | 0 | 11 | 00 | 11 | 01 |
| 1 | 1 | 10 | 01 | 00 | 01 |
| | | B*A* | | | |

1. List all possible states

2. Determine Next State of BA for all permutations of the inputs

For B*A*, plug in appropriate equations for each entry.
For A=0, B=1, X=0, Y=0:

$A^* = XYA' + Y'A'B + ABY$
$= 0 \bullet 0 \bullet 1 + 1 \bullet 1 \bullet 1 + 0 \bullet 1 \bullet 0$
$= 1$

$B^* = YB' + X'Y'B + Y'A'B$
$= 0 \bullet 0 + 1 \bullet 1 \bullet 1 + 1 \bullet 1 \bullet 1$
$= 1$

Thus entry is 11.

用 transition equation 算 $B^*A^*$

# JK Flip-Flops Analysis Example: State Table

Give each numerical state an alphanumeric name:

**00 = H** **01 = I**

**10 = J** **11 = K**

Substitute into **Transition Table** to form the **State Table**:

| | XY | | | |
|---|---|---|---|---|
| **S** | **00** | **01** | **10** | **11** |
| **H** | **H** | **J** | **H** | **K** |
| **I** | **H** | **J** | **H** | **J** |
| **J** | **K** | **H** | **K** | **I** |
| **K** | **J** | **I** | **H** | **I** |
| | **S*** | | | |

# JK Flip-Flops Analysis Example: State/Output Table

Expand your **State Table** to include the output **Out** to form the **State/Output Table**.

This is a Moore Machine – the outputs are *only* a function of the state. Thus, they *do not change* per input and can be listed in their own column.

| | XY | | | | |
|---|---|---|---|---|---|
| S | 00 | 01 | 10 | 11 | Out |
| H | H | J | H | K | 0 |
| I | H | J | H | J | 1 |
| J | K | H | K | I | 0 |
| K | J | I | H | I | 0 |
| | S* | | | | |

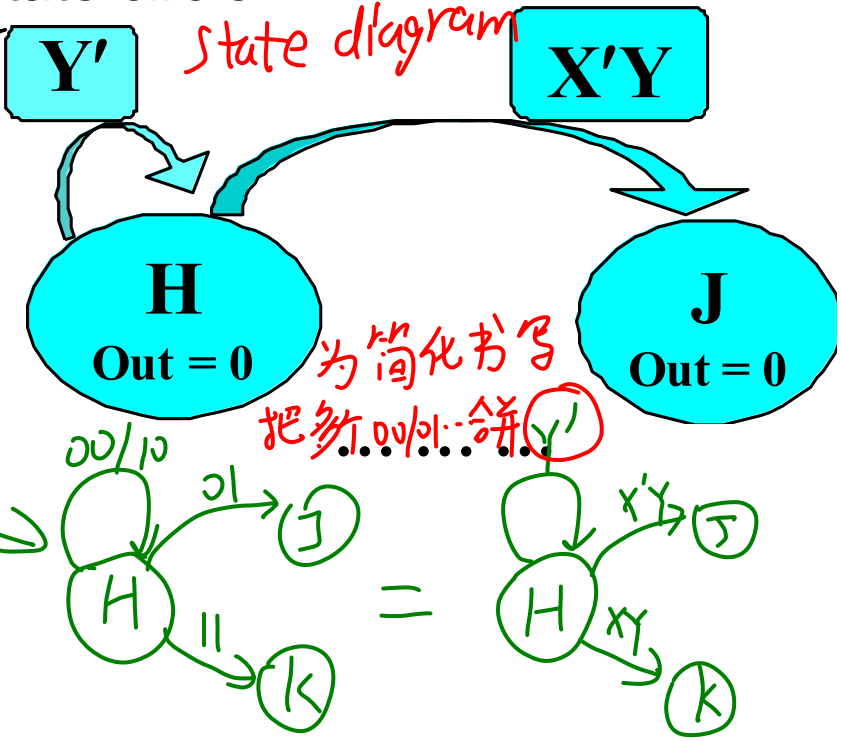
# JK Flip-Flops Analysis Example: State Diagram Development Process (1/2)

- Development Process: Same as before with a Mealy Machine, but can put the *output* within the actual state circle.

画state转换图

Could have labelled this as *D0*, where D=don't care.

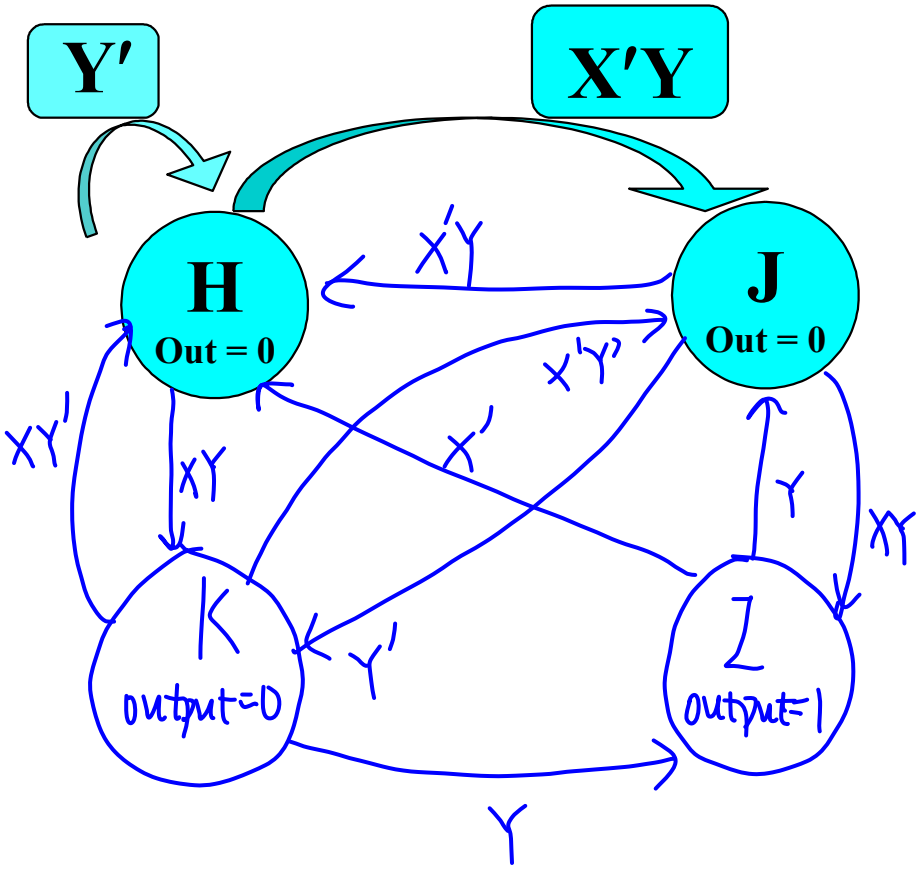
| S | XY 00 | 01 | 10 | 11 | Out |
|---|---|---|---|---|---|
| H | H | J | H | K | 0 |
| I | H | J | H | J | 1 |
| J | K | H | K | I | 0 |
| K | J | I | H | I | 0 |
| | | | S* | | |



# JK Flip-Flops Analysis Example: State Diagram Development Process (2/2)

- Complete the State Diagram, based on the information in the State/Output Table.

*To be completed in class …*



# Time Delay & Maximum Clock Speed (1/3)

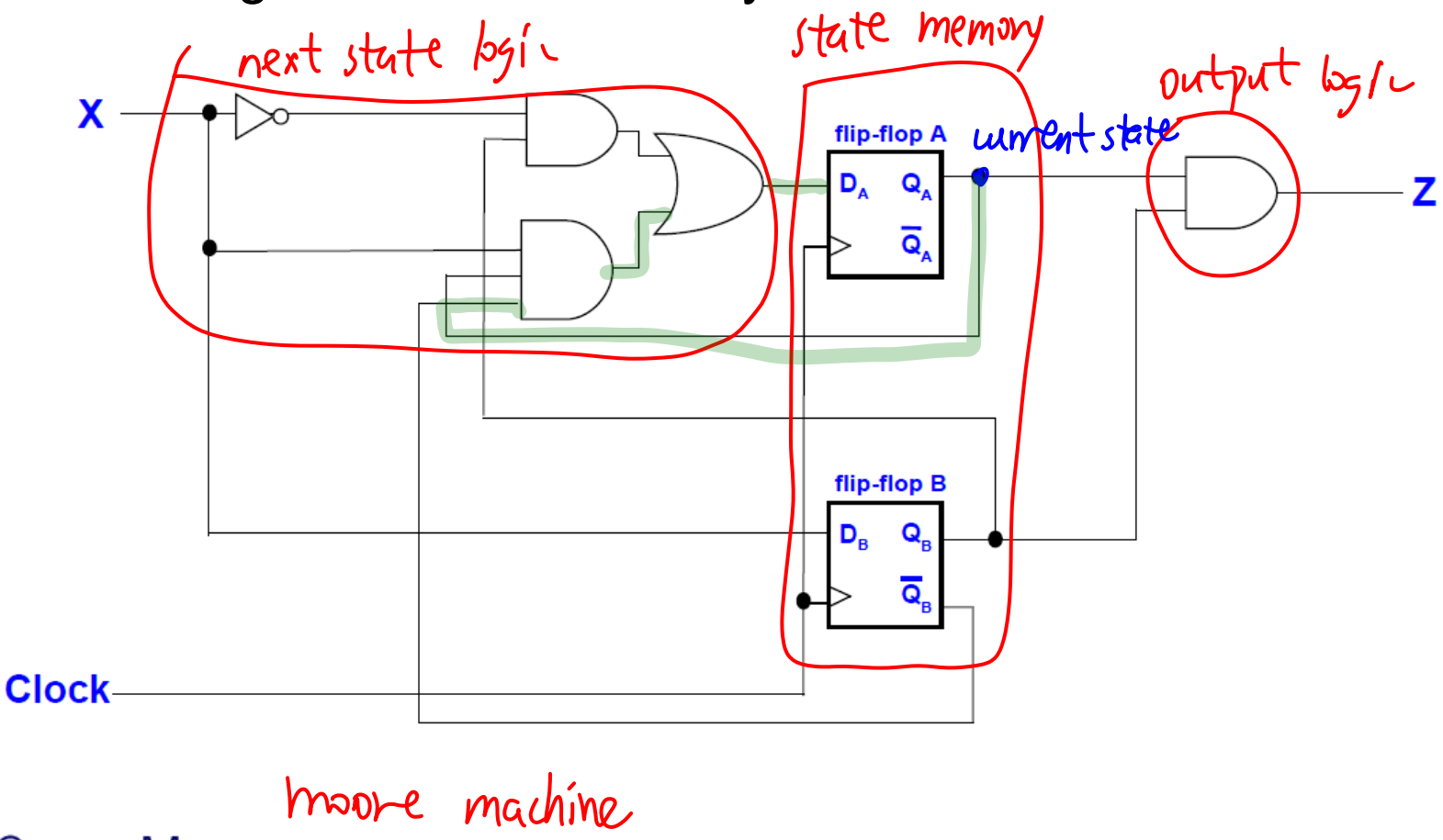
- Gate delay limits the speed of state machines...
- Feedback logic delay dictates **highest clock frequency**
- For circuit with **max delay $T_D$**, **max clock frequency** is:

$$F_{max} < \frac{1}{T_D}$$

- Attempt to **clock > $F_{max}$** and circuit will be **unstable**

# Time Delay & Maximum Clock Speed (2/3)

- Assume all gates have 7ns delay ...



# Time Delay & Maximum Clock Speed (3/3)

- Assume all gates have 7ns delay ...