Week 4 - T1 2020

Sequential Circuits Elements and Analysis

ELEC2141: Digital Circuit Design

UNSW SYDNEY

Summary

Circuit design

Hierarchical design

Decoders

Encoders

Multiplexers

Demultiplexers

UNSW SYDNEY

Overview

Sequential circuits

Latches

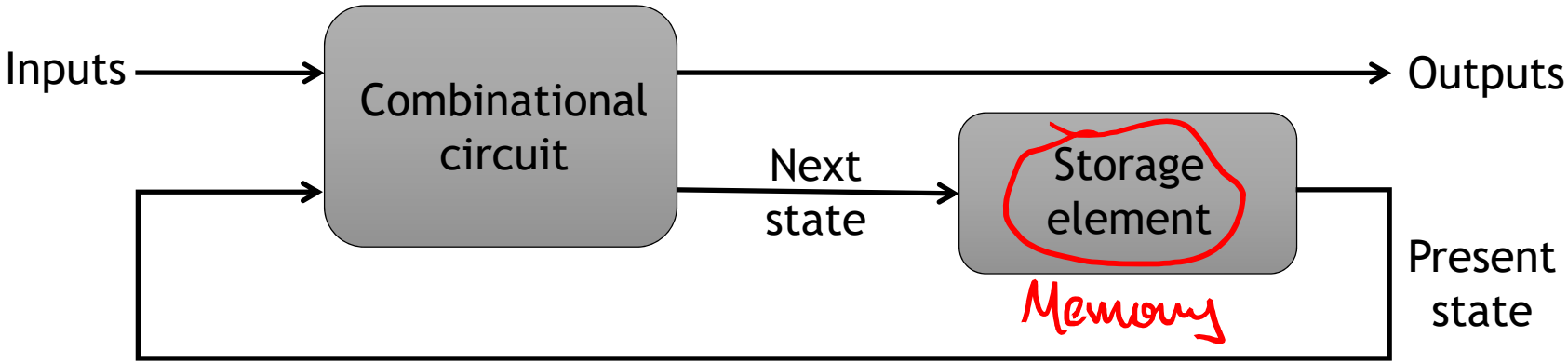
Flip- flops

Sequential circuit analysis

Reading: Mano - Chapter 4, 4.1-4.4

Sequential circuits

Digital circuits that include storage elements are described as *sequential circuits*

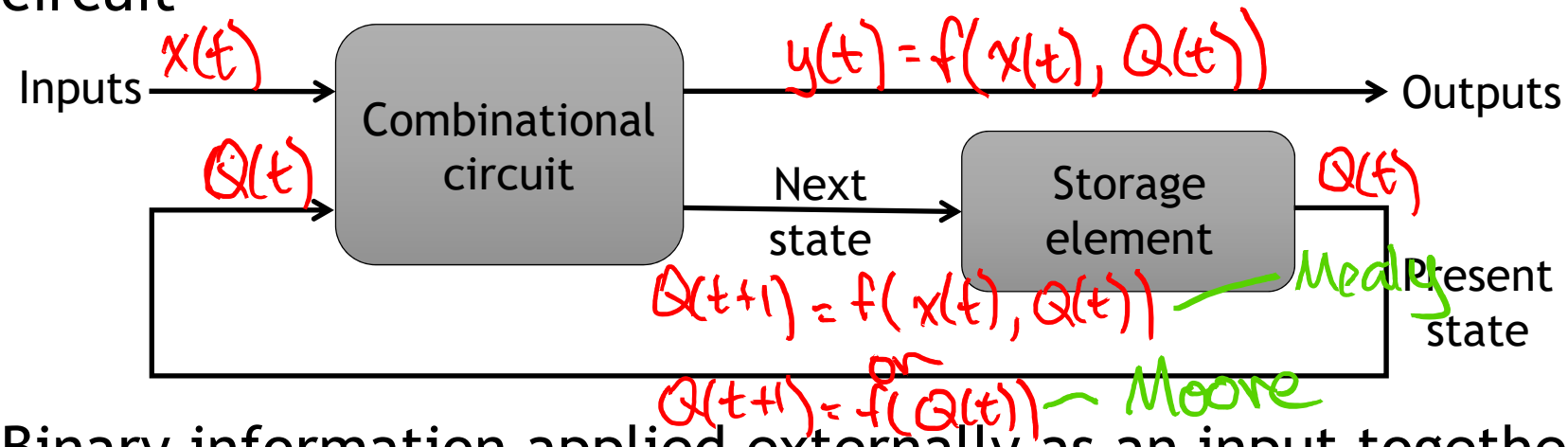


Sequential circuits consist of a combinational circuit connected to storage elements to form a feedback path

Storage elements are devices capable of storing binary data

Sequential circuits

Binary information stored in the storage element at any given time defines the *present state* of the sequential circuit



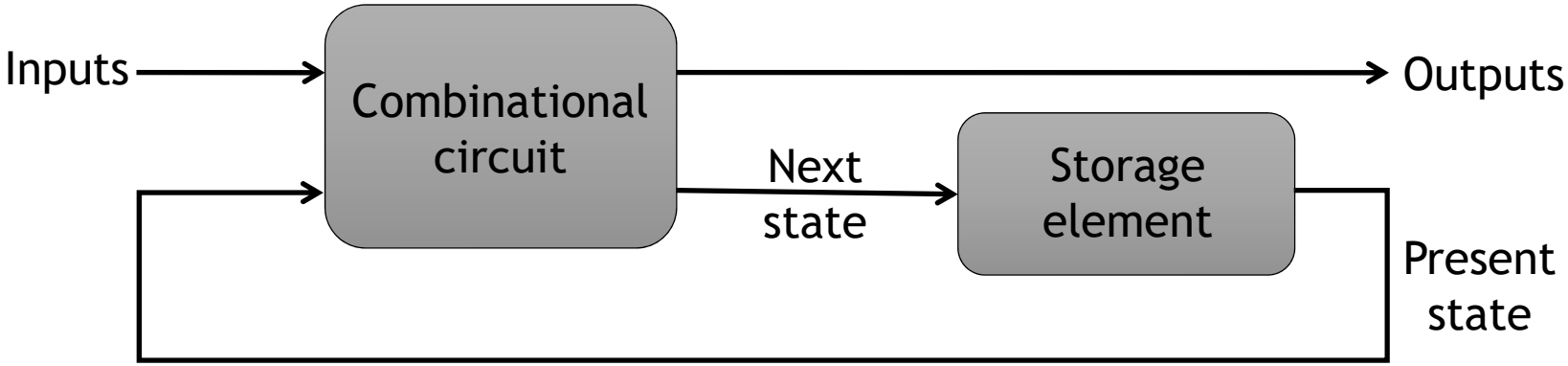
Binary information applied externally as an input together with the *present state* determines the *next state* of the sequential circuit

The output is also determined by the input and the present state of the sequential circuit

UNSW SYDNEY

Sequential circuits

The behavior of the sequential circuit is specified by a time sequence of its inputs, internal states and outputs



Sequential circuits are divided into *synchronous* and *asynchronous* sequential circuits.

dependant on clock

In synchronous sequential circuits, the outputs and states are modified (or changed) only at a discrete instants of time.

UNSW SYDNEY

Asynchronous sequential circuits

Dependant on input event

In asynchronous sequential circuits, the outputs and states are modified (or changed) only at any instant of time depending upon the inputs

The storage elements commonly used in asynchronous sequential circuits are time delay devices

Propagation delay in a gate or through interconnected gates can constitute a time delay device

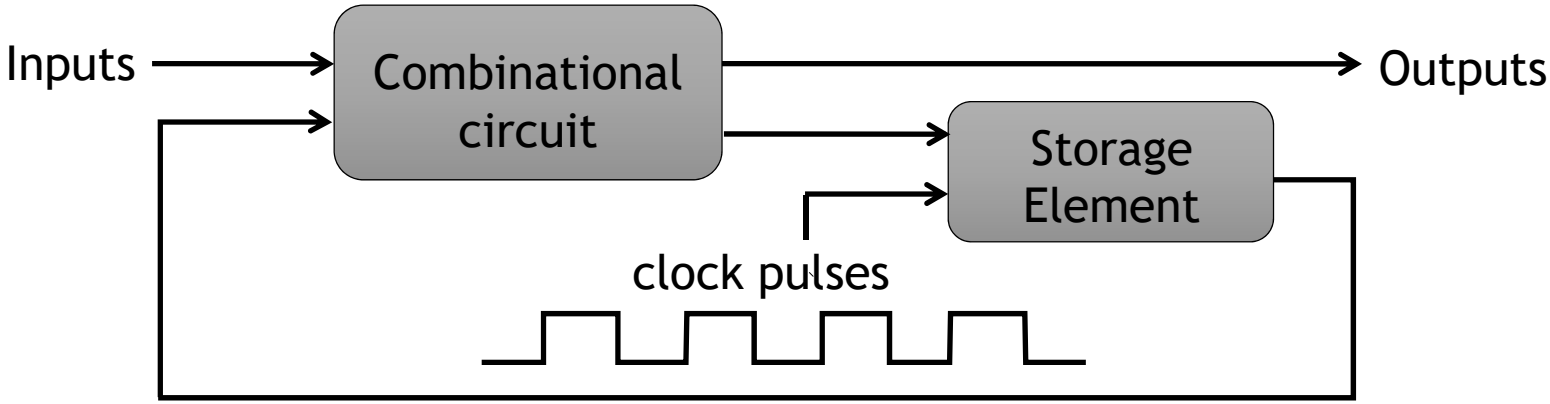
Therefore, asynchronous sequential circuits can be regarded as combinational circuits with feedback

The feedback may introduce instabilities at times. The instability problem imposes many difficulties on the designer

UNSW SYDNEY

Synchronous sequential circuits

In synchronous sequential circuits, synchronization is achieved by a timing device called a clock generator.



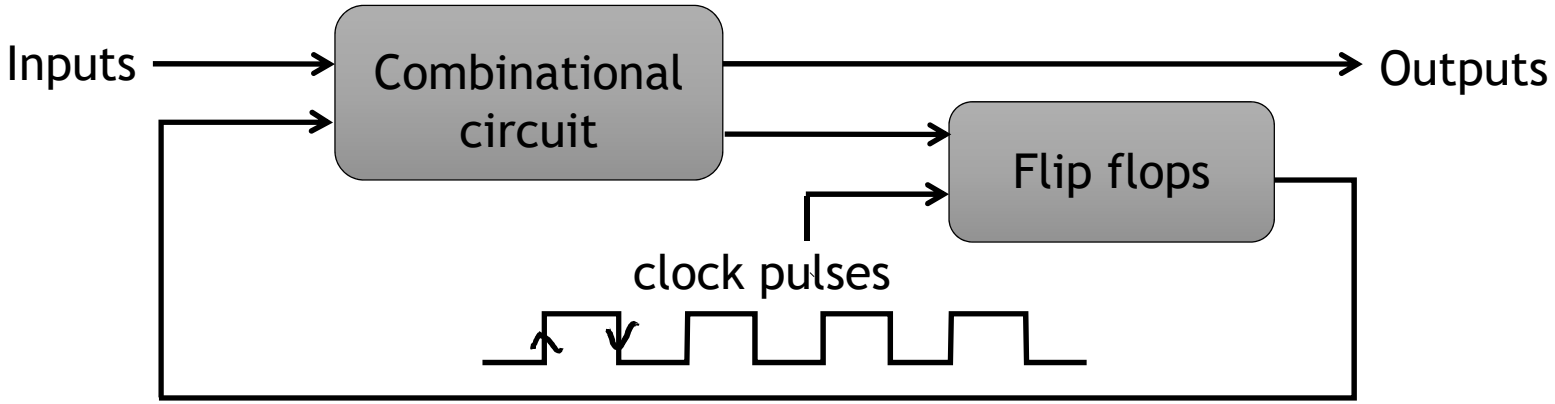
A clock generator provides a clock signal in the form of periodic train of pulses called clock or clk

Synchronous sequential circuits are also called clocked sequential circuits

They are commonly used as they do not manifest instability and easy to design

Synchronous sequential circuits

The storage elements used in clocked sequential circuits are called flip-flops



A flip-flop is a binary storage device that is capable of storing one bit of information

Flip-flops change states only in response to a clock pulse

A flip-flop has two outputs: normal and its complement

Q, $\overline{Q}$

UNSW SYDNEY

Storage elements

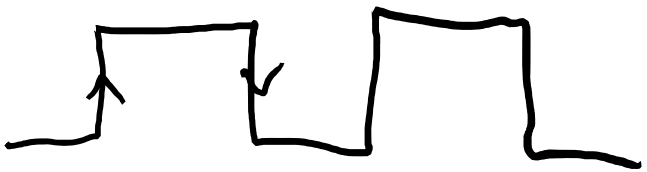
Storage elements are characterized by

- the number of inputs they possess
- the manner in which the inputs affect the binary state

Latches are storage elements that operate based on signal levels and are thus signal level sensitive devices

async.

Flip-flops are storage elements controlled by a clock transition and thus are edge sensitive devices

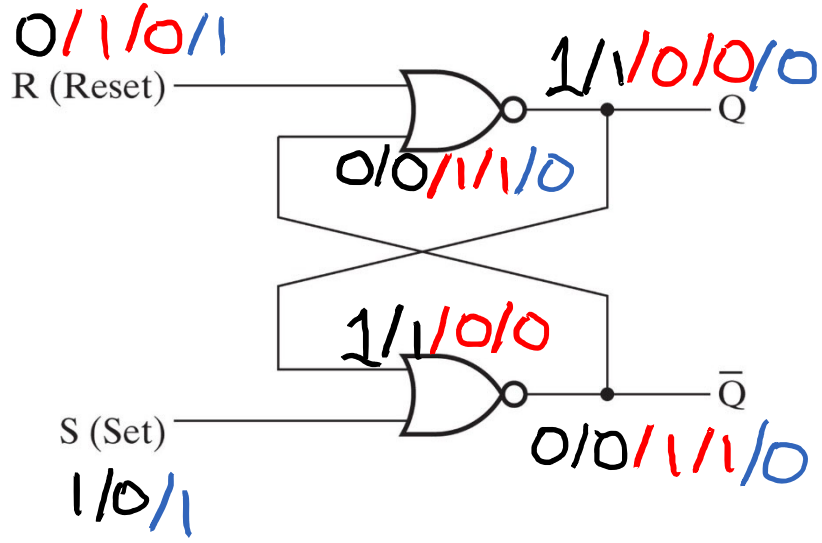


sync.

set / Reset NOR

SR latch

X	Y	NOR
0	0	1
0	1	0
1	0	0
1	1	0



S	R	Q	$\bar{Q}$	
1	0	1	0	Set
0	0	1	0	
0	1	0	1	Reset
0	0	0	1	
1	1	0	0	Undefined

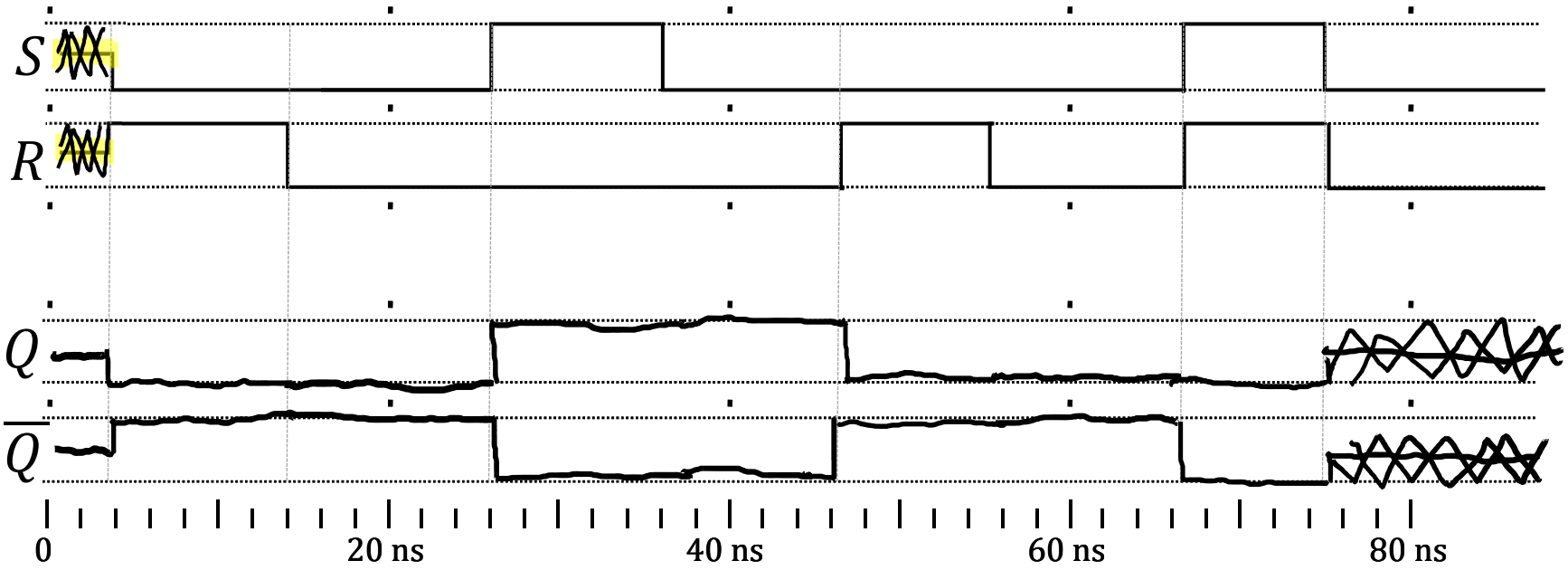
The *SR latch* is constructed with two cross-coupled NOR gates and two inputs labeled as $\boldsymbol{S}$ (set) and $\boldsymbol{R}$ (reset)

The latch has two states. When $\boldsymbol{Q} = \mathbf{1}$ and $\overline{\boldsymbol{Q}} = \mathbf{0}$, the latch is said to be in the set state

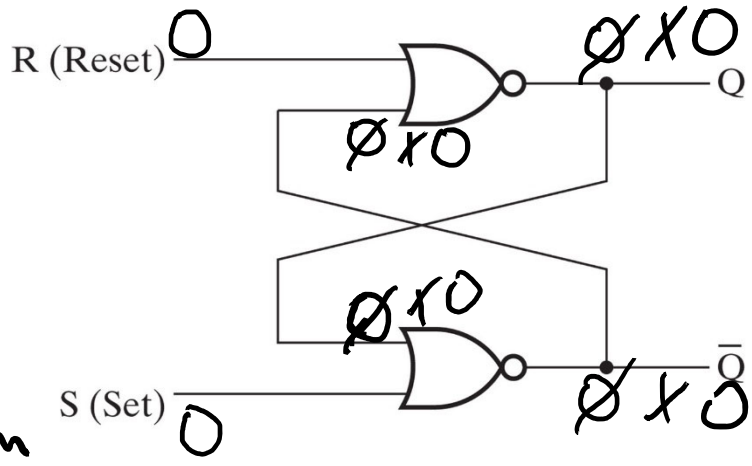
When $\boldsymbol{Q} = \mathbf{0}$ and $\overline{\boldsymbol{Q}} = \mathbf{1}$, the latch is said to be in the reset state

UNSW SYDNEY

SR latch



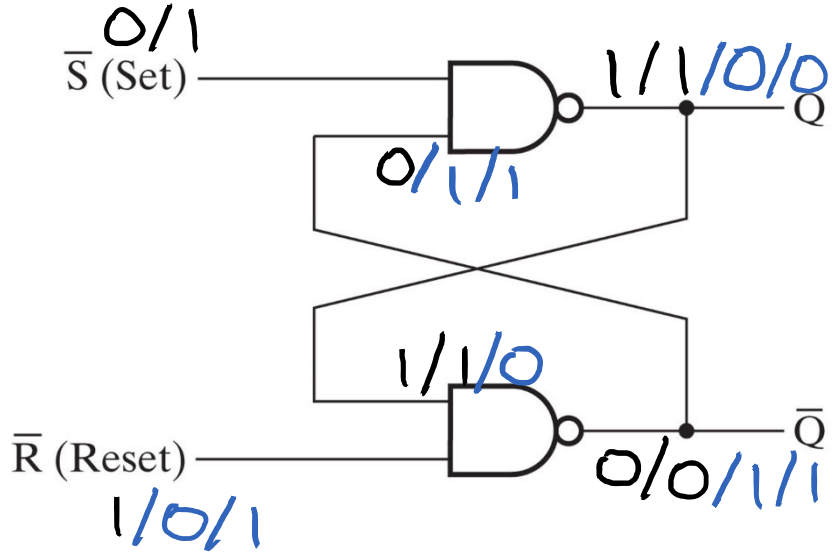
	S	R	Q	$\overline{Q}$	
→	1	0	1	0	Set
→	0	0	1	0	
→	0	1	0	1	Reset
→	0	0	0	1	
	1	1	0	0	Undefined
	0	0	X	X	Race Condition



$\overline{S}\overline{R}$ latch

NAND

active low $\overline{S}\overline{R}$ latch



$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$	
0	1	1	0	Set
1	1	1	0	
1	0	0	1	Reset
1	1	0	1	
0	0	1	1	Undefined
1	1	X	X	Race Condition

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

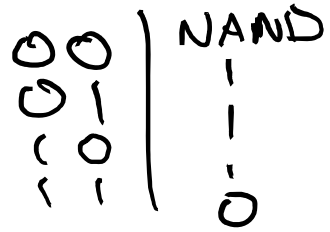
It can also be constructed with two cross-coupled NAND gates and two inputs labeled $\overline{S}$ (set) and $\overline{R}$ (reset).

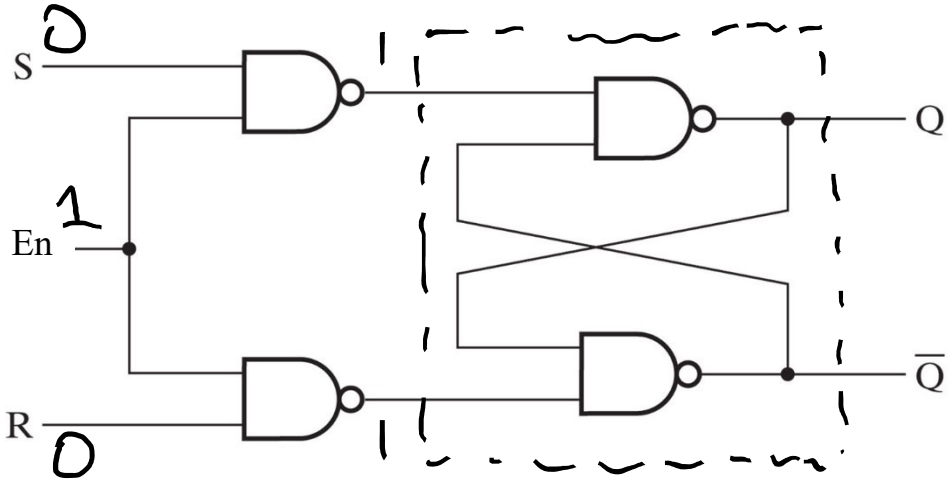
The latch has two states. When $Q = 1$ and $\overline{Q}$ = 0, the latch is said to be in the set state.

When $Q = 0$ and $\overline{Q}$ = 1 , the latch is said to be in the reset state.

UNSW SYDNEY

SR latch with control





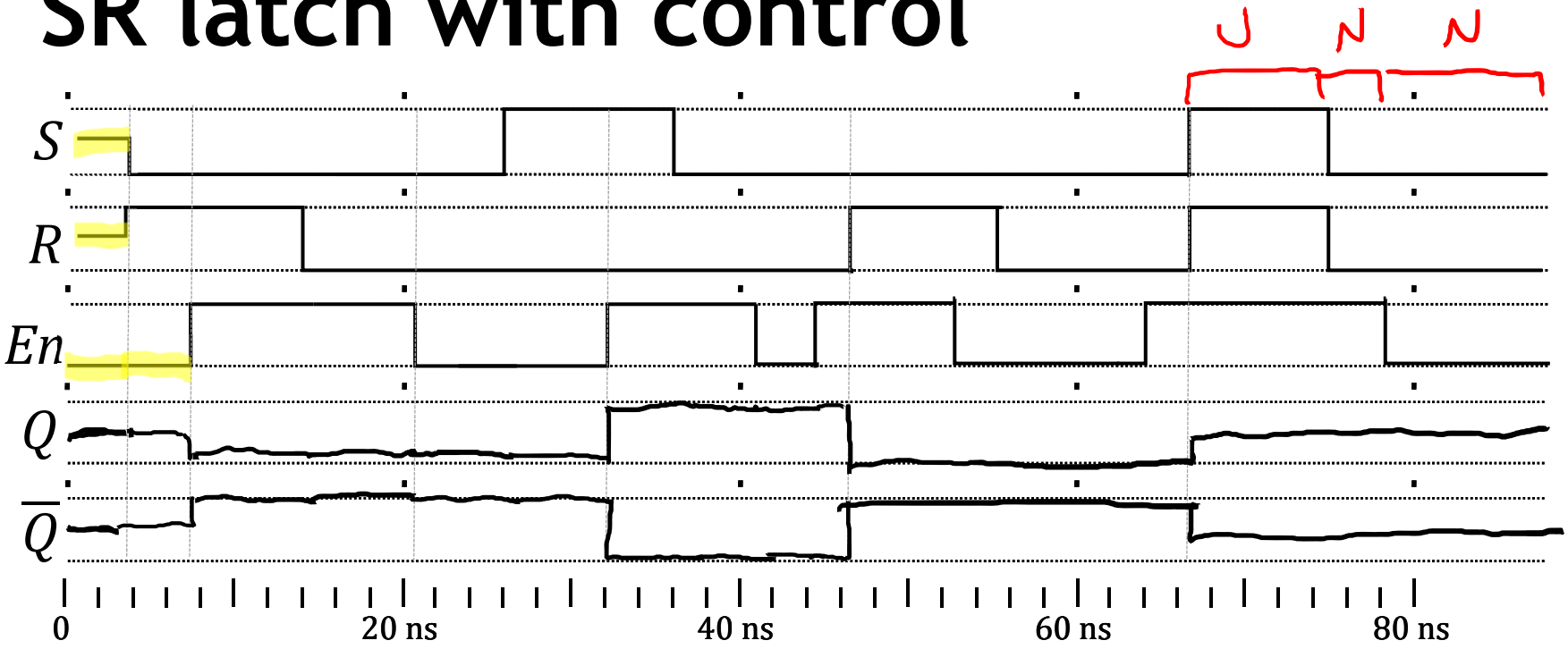
En	S	R	Next state of Q
0	X	X	Q unchanged
1	0	0	Q unchanged
1	0	1	Q=0 Reset
1	1	0	Q=1 Set
1	1	1	Undefined

The basic operation of SR latch can be modified by adding an enable input signal that determines when the state of the latch can be changed

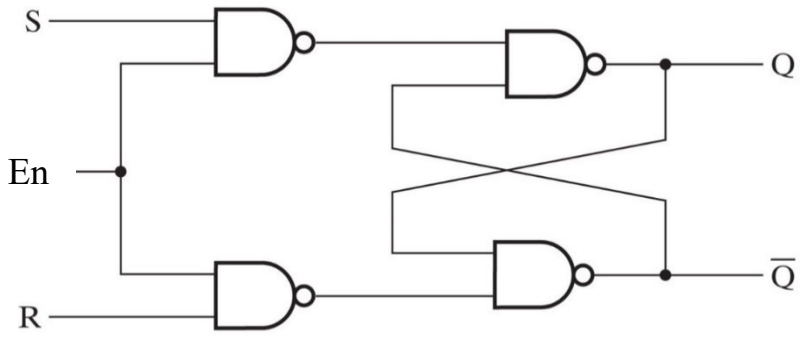
It consists of the basic SR latch and two additional NAND gates

With $\boldsymbol{En = 1}$, $\boldsymbol{S = 1}$, and $\boldsymbol{R = 1}$ the next state of indeterminate condition occurs

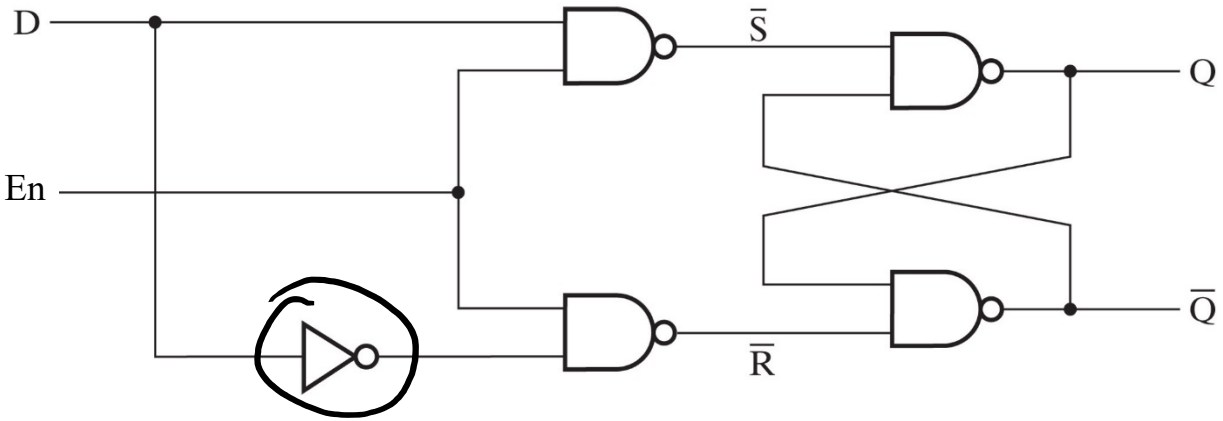
SR latch with control



En	*S*	*R*	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	Q=0 Reset
1	1	0	Q=1 Set
1	1	1	Undefined



D latch



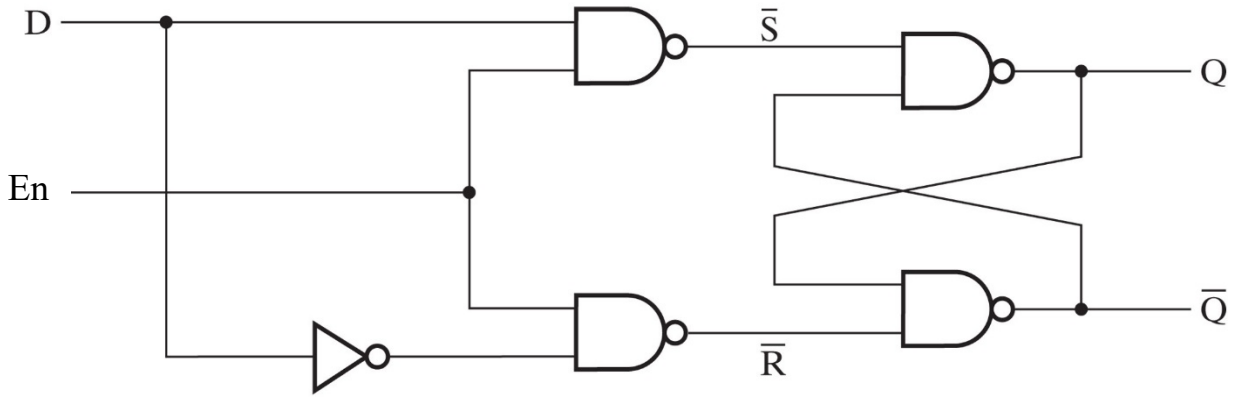
En	*D*	*Q*
0	X	No change
1	0	Q=0 Reset
1	1	Q=1 Set

If $\boldsymbol{D = 1}$, the $\boldsymbol{Q}$ output goes to 1, placing the circuit in the set state. If $\boldsymbol{D = 0}$, the $\boldsymbol{Q}$ output goes to 0, placing the circuit in the reset state

The binary information present at the data input of the *D latch* is transferred to the $\boldsymbol{Q}$ output when the enable input is asserted

UNSW SYDNEY

D latch



En	D	Q
0	X	No change
1	0	Q=0 Reset
1	1	Q=1 Set

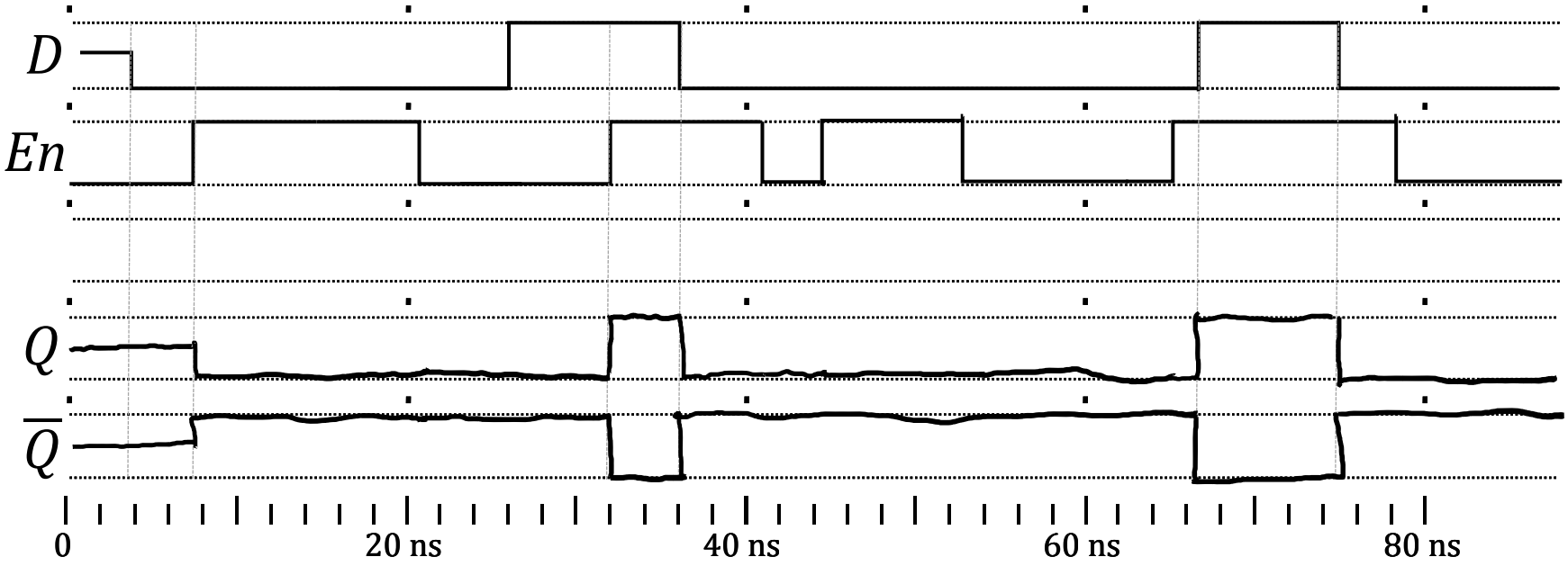
The indeterminate condition in the SR latch when $\boldsymbol{En} = \boldsymbol{1}$, $\boldsymbol{S} = \boldsymbol{1}$, and $\boldsymbol{R} = \boldsymbol{1}$ can be eliminated by ensuring that inputs $\boldsymbol{S}$ and $\boldsymbol{R}$ are never equal to 1 at the same time

This is done with a D latch

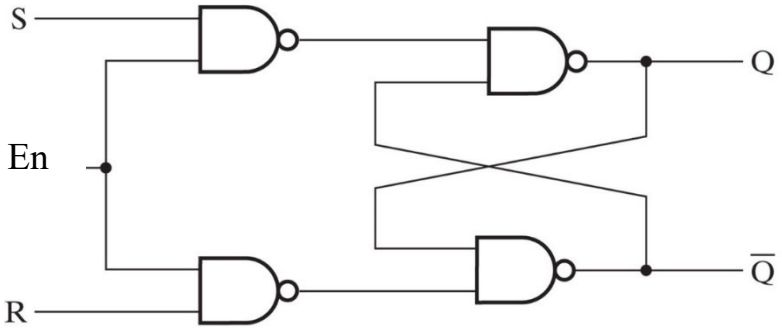
$\boldsymbol{En} = \boldsymbol{0}$ produces $\overline{\boldsymbol{S}} = \boldsymbol{1}$ and $\overline{\boldsymbol{R}} = \boldsymbol{1}$, which drives the basic SR latch to remain in same state

The D-input is sampled when $\boldsymbol{En} = \boldsymbol{1}$

D latch



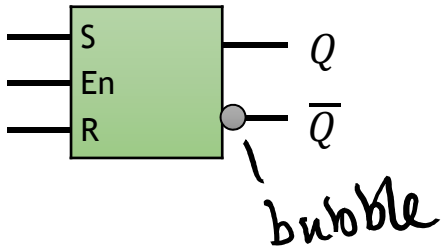
En	*D*	*Q*
0	X	No change
1	0	Q=0 Reset
1	1	Q=1 Set



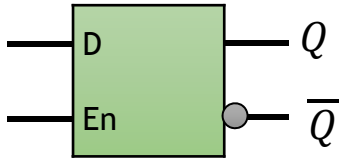
Latch-designations

A latch is designated by a rectangular block with inputs on the left and outputs on the right

One output designates the normal output and the other designates the complement output



SR latch with enable



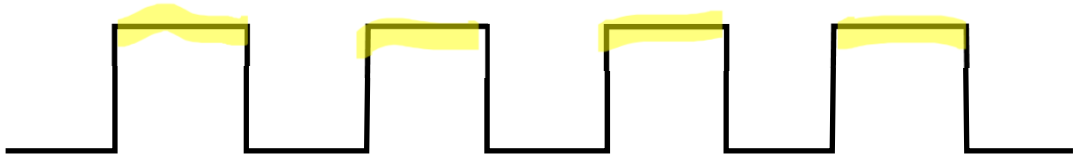
D latch

NAND implementation

Clocked D-Latch

In most sequential circuits, the En (control) input is connected to a clock generator

For a clocked D-latch, the output *Q* follows the input D whenever the clock is high

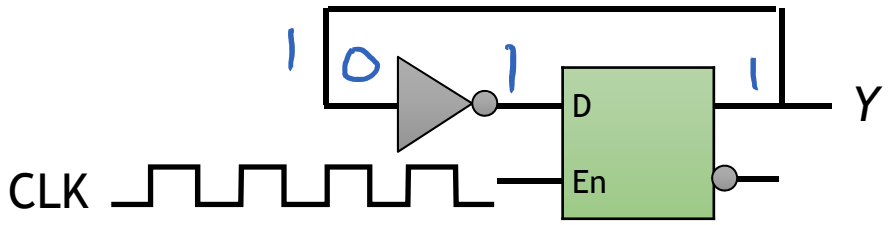


There may be connecting paths between one storage elements to another, or even back to the same element

Can be a problem when using latches

The latch timing problem

Consider the following circuit:



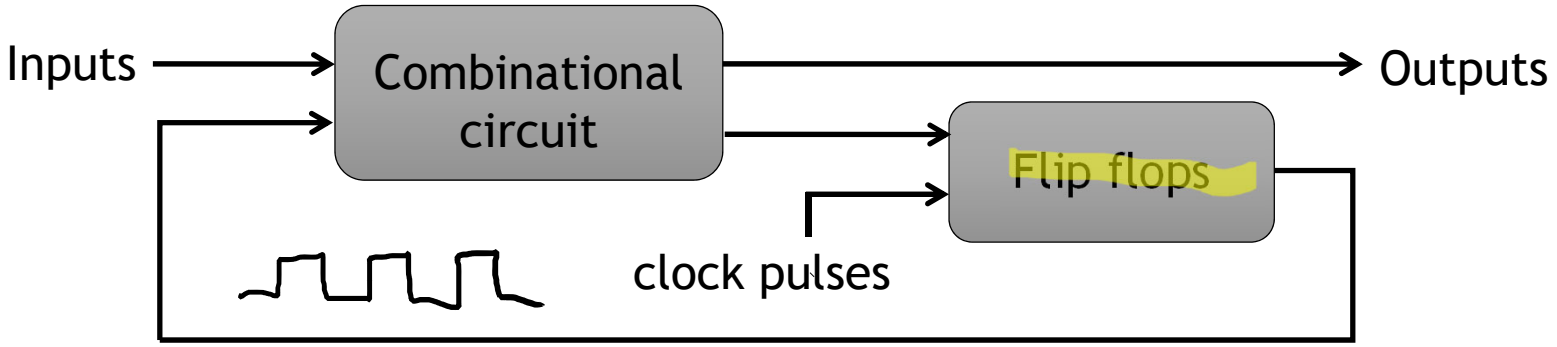
The output *Y* will oscillate whenever the clock is high, and will be constant when the clock is low

Desired behaviour: *Y* changes only once per clock pulse

The latch timing problem

Using latches as storage elements in a synchronous logic circuit will lead to a serious malfunction as the state of the storage element will continue to change as long as the input changes while the clk is high

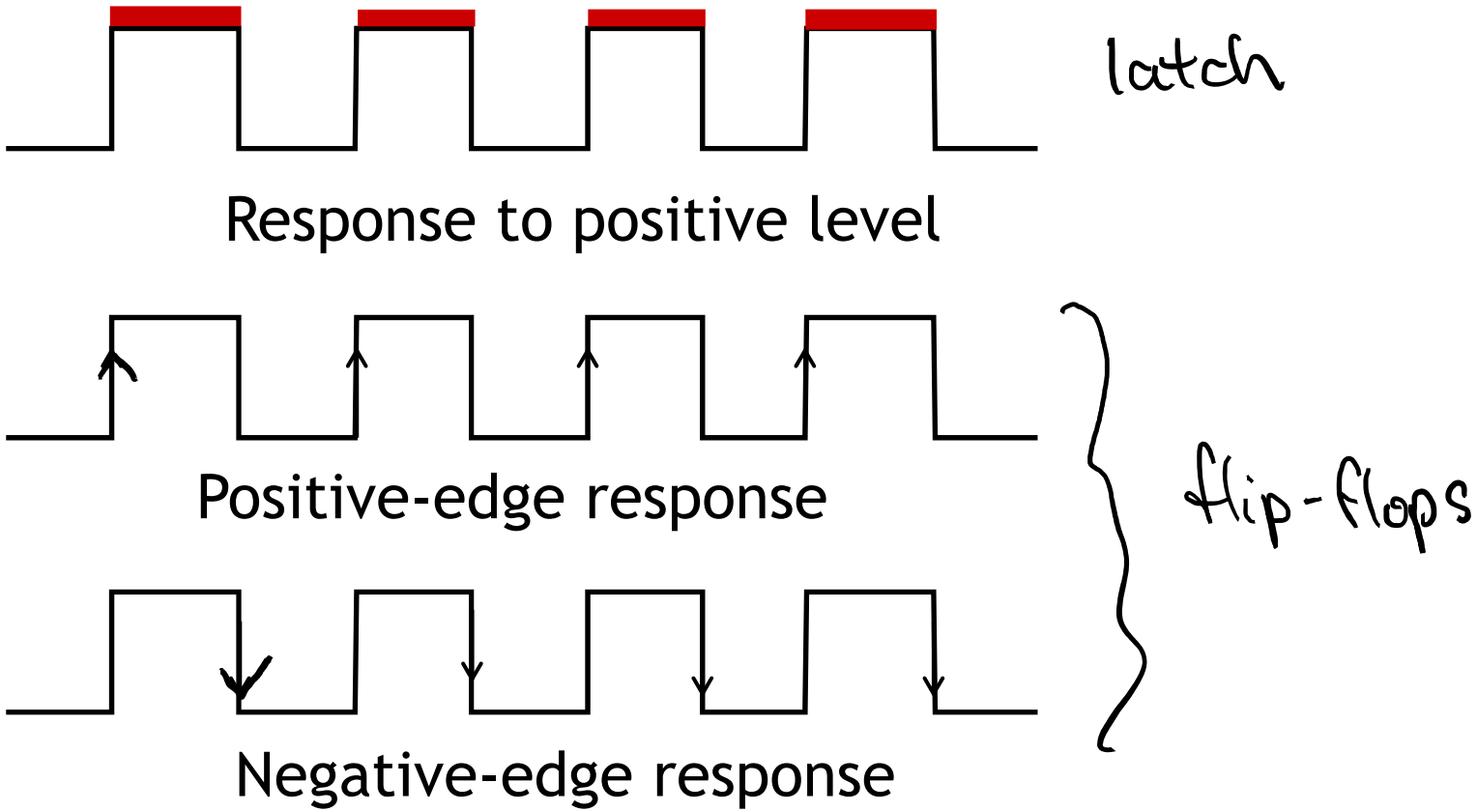
It is thus difficult to get a stable output when latches are used as storage elements



This can be resolved with *flip-flops* whose states can only change in response to signal transition rather than signal level, i.e. edge sensitive devices

UNSW SYDNEY

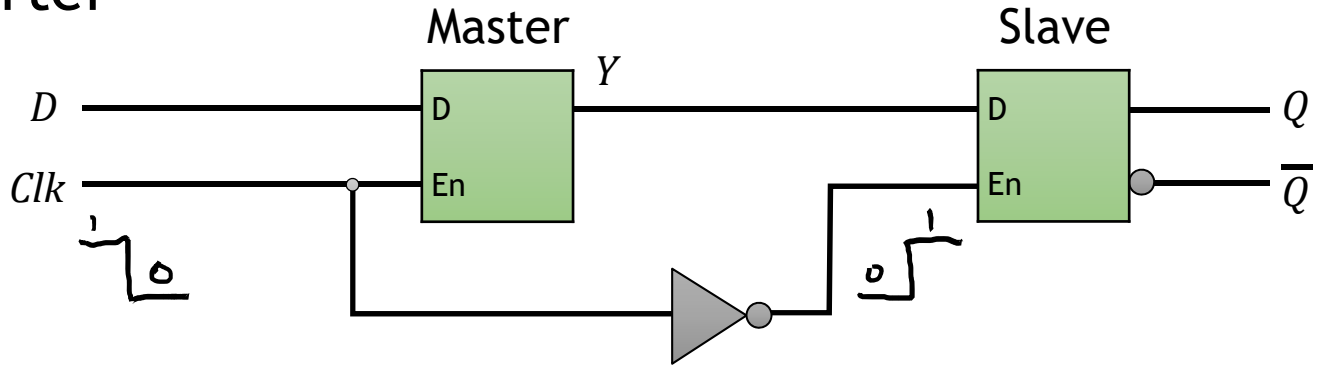
Solving the timing problem



UNSW SYDNEY

Edge-triggered D flip-flop

A D flip-flop is constructed from two D latches and an inverter



When clk = 1, master D latch will be enabled while slave D latch is disabled, $\boldsymbol{Y = D}$

When clk = 0, master D latch is disabled while slave D latch is enabled, $\boldsymbol{Q = D}$

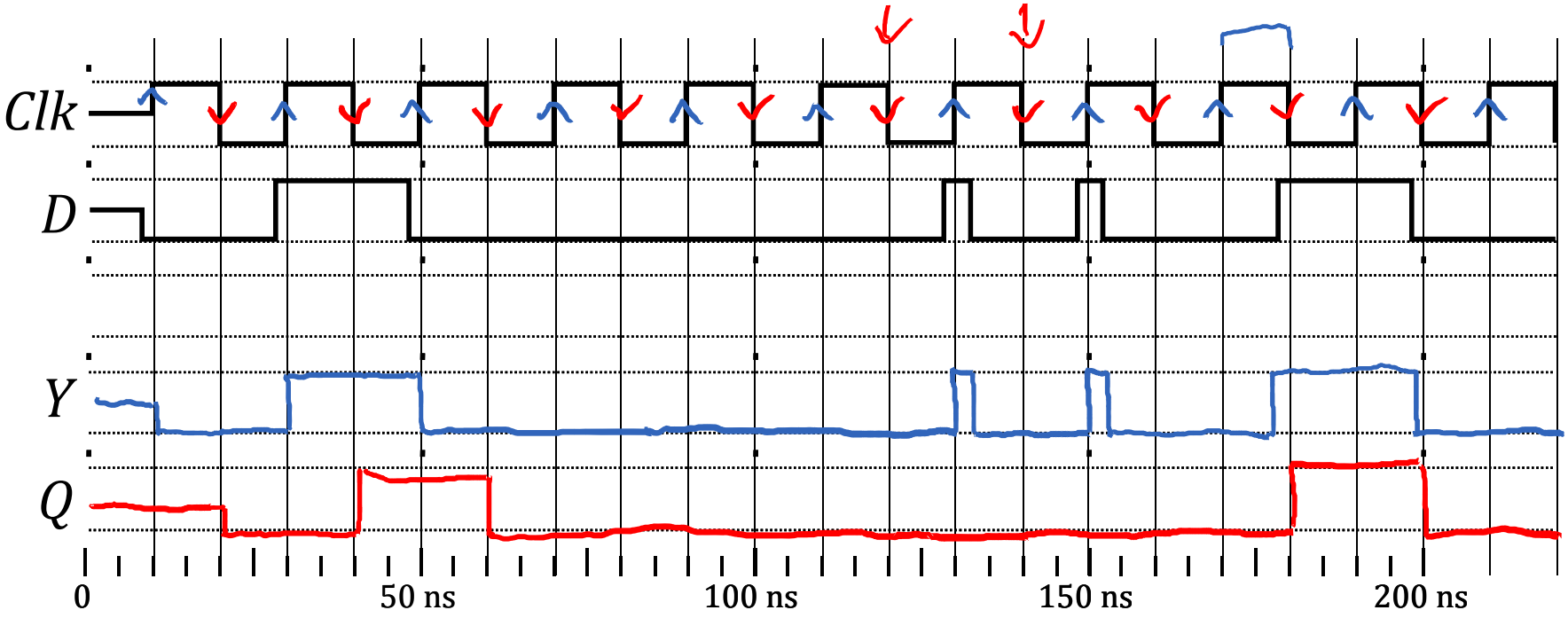
This output remains unchanged until the clock transitions from 1 to 0, i.e. it is negative edge triggered

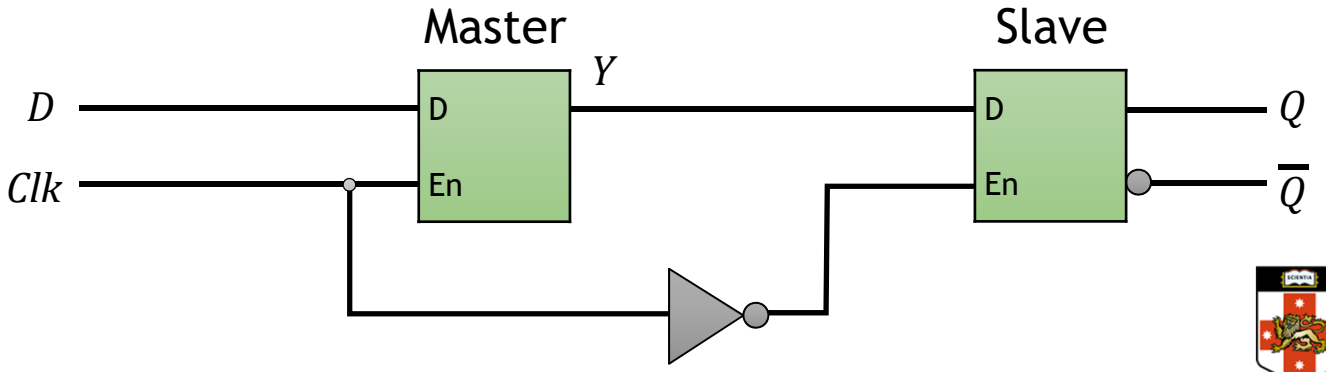
UNSW SYDNEY

Negative

Edge-triggered D flip-flop

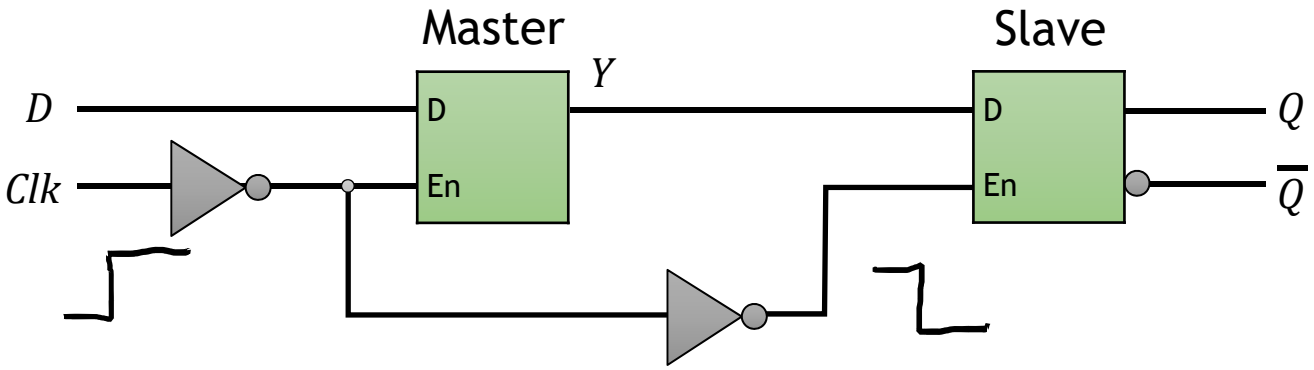
Clk	*D*	*Q*
0	X	No change
1	0	Q=0 Reset
1	1	Q=1 Set





Edge-triggered D flip-flop

In this D flip-flop



When clk = 0, the master D latch will be enabled while the slave D latch is disabled, $\boldsymbol{Y = D}$

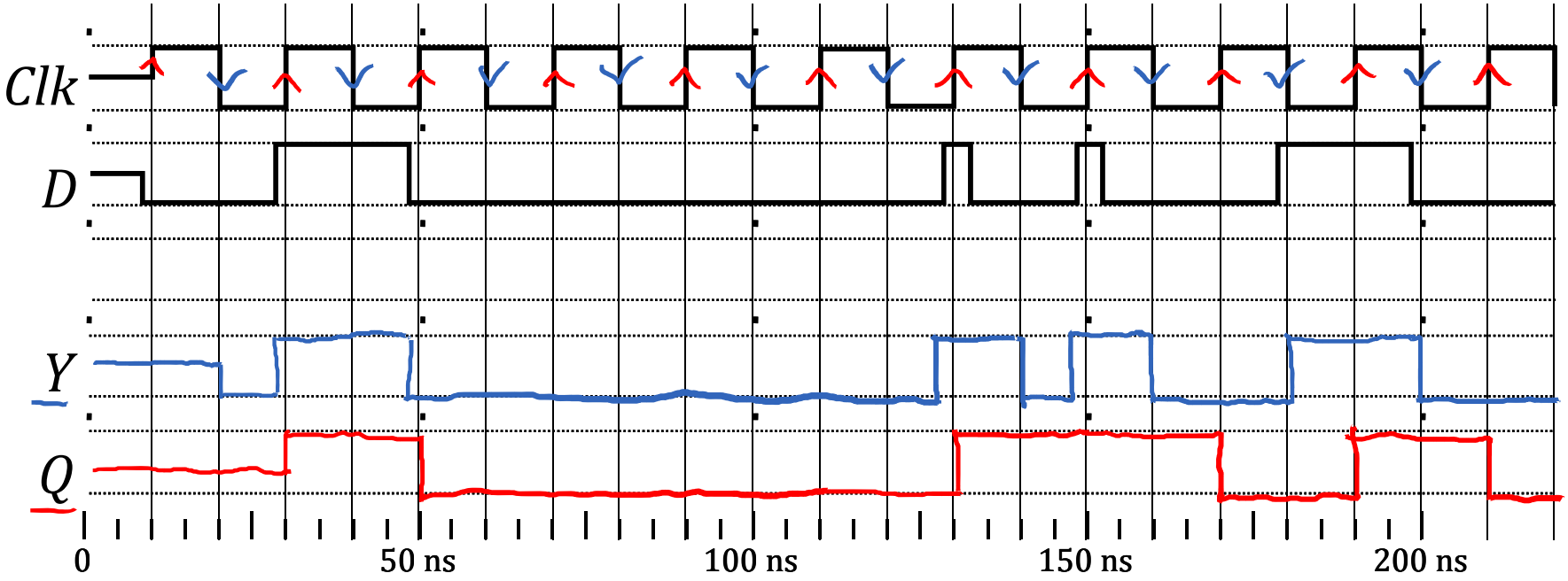
When clk = 1, the master D latch is disabled while the slave D latch is enabled, $\boldsymbol{Q = D}$

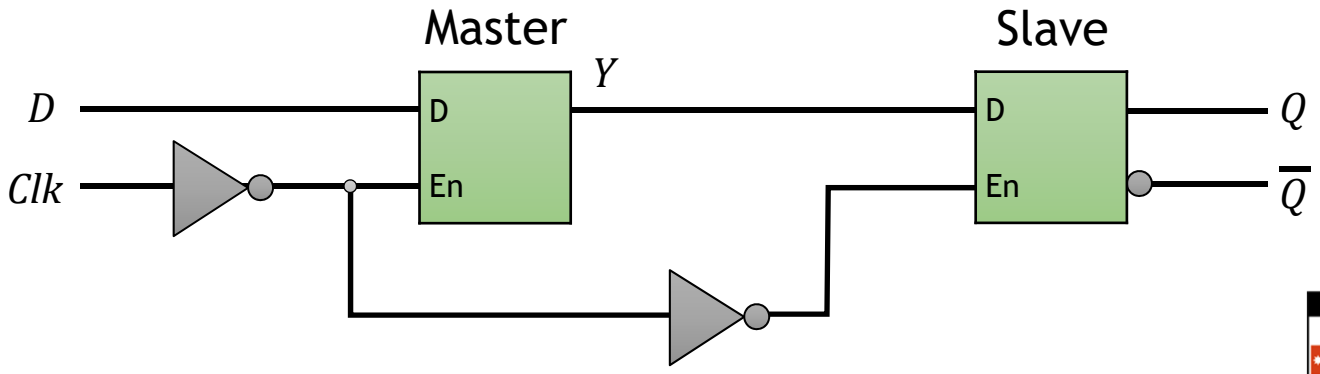
This output remains unchanged until the clock transitions from 0 to 1, i.e. it is positive-edge triggered

positive

Edge-triggered D flip-flop

Clk	*D*	*Q*
0	X	No change
1	0	Q=0 Reset
1	1	Q=1 Set

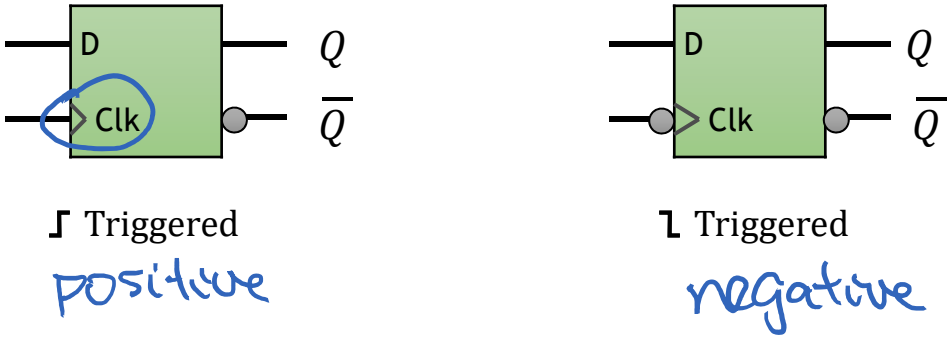




UNSW SYDNEY

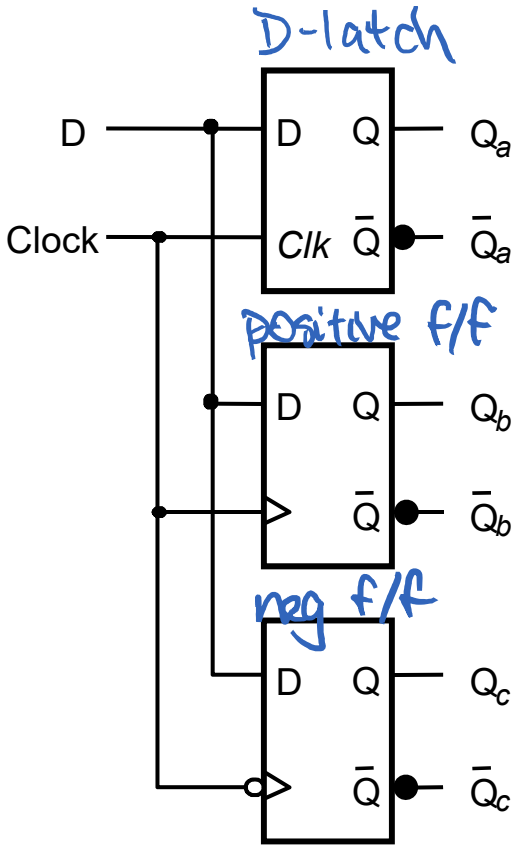
Flip-flop symbols

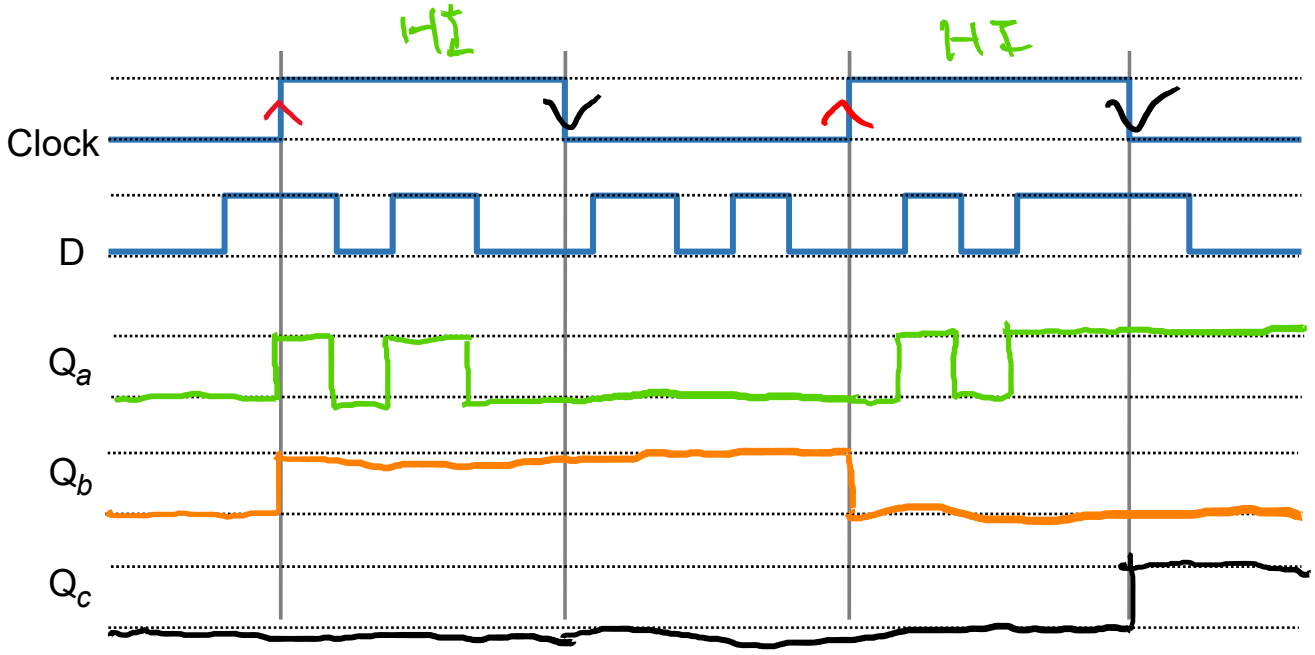
Positive and negative edge-triggered D flip-flops are represented by



The symbols ⌜ and ⌞ are used to denote positive and negative edge triggering, respectively

Level vs Edge Triggered Devices





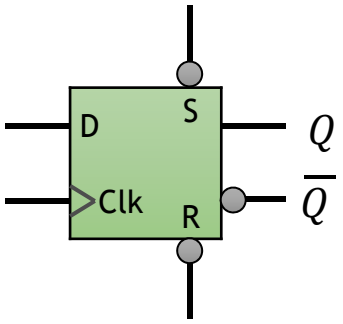
UNSW SYDNEY

Direct inputs

Asynchronous *Set* and *Reset* input signals

Used to force a value to the output of the flip-flop, independent of the clock

Usually used to initialize the circuit to known values at system start-up



Active-low set and reset

S	R	Clk	D	Q	$\overline{Q}$
0	1	X	X	1	0
1	0	X	X	0	1
0	0	X	X	Undefined	
1	1	↑	0	0	1
1	1	↑	1	1	0

Other flip-flops

The edge triggered D flip-flop requires the smallest number of gates and hence is the most economical and efficient flip-flop

Other types of flip-flops can be constructed by using D flip-flop and external logic

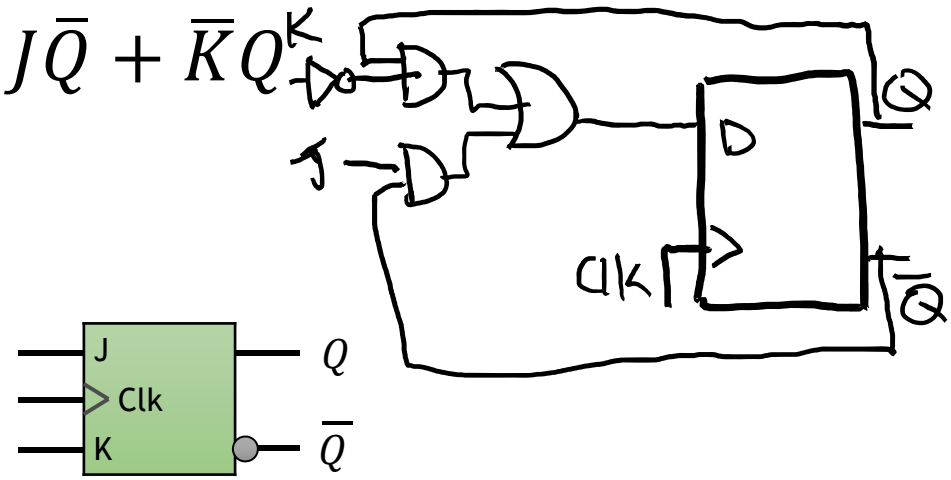
The other two flip-flops less widely used in the design of digital systems are JK and T flip-flops

JK flip-flops

The JK flip-flop has a similar behavior to the SR latch with the exception that 1-1 input is allowed.

It is built from a D flip-flop, where

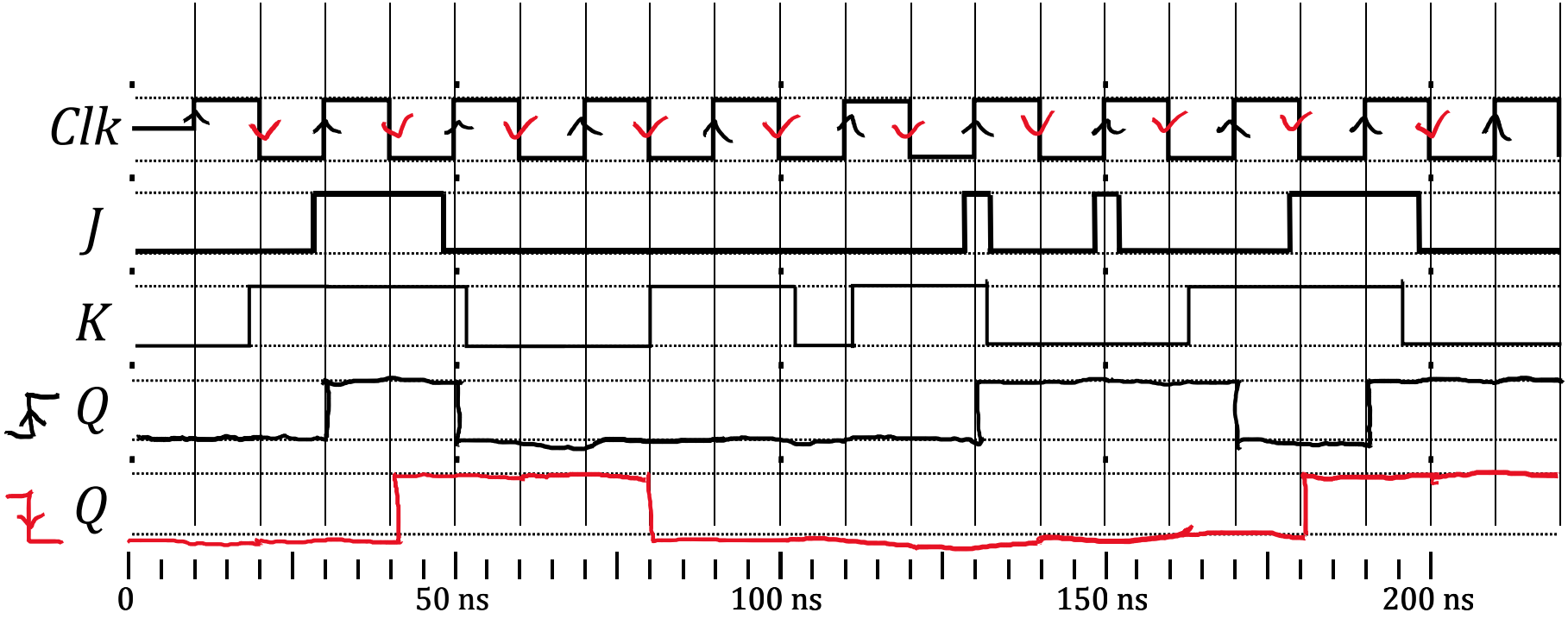
$$D = J\bar{Q} + \bar{K}Q$$



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

UNSW SYDNEY

JK flip-flop



J	*K*	*Q(t+1)*
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\overline{Q}(t)$

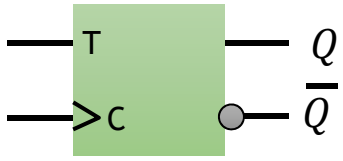
T flip-flops

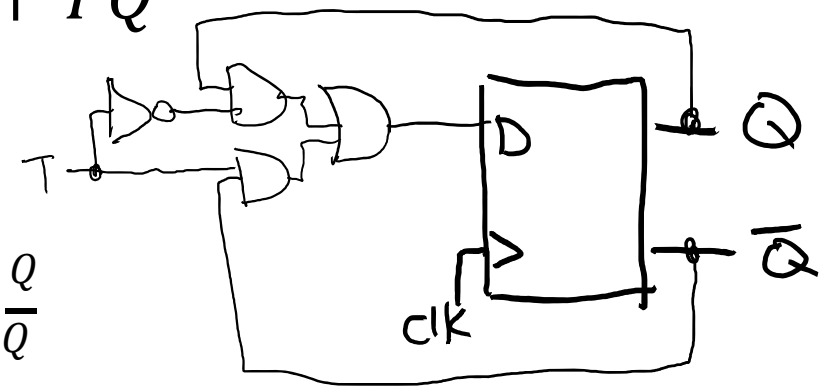
The T (Toggle) flip-flop maintains its stored value when T = 0, and complements it when T = 1.

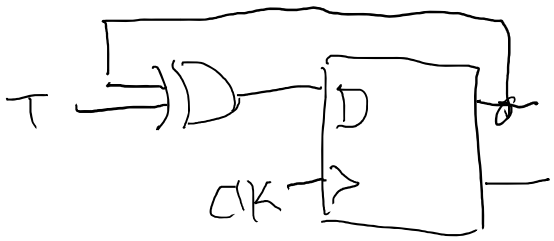
It is built from a D flip-flop, where

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

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

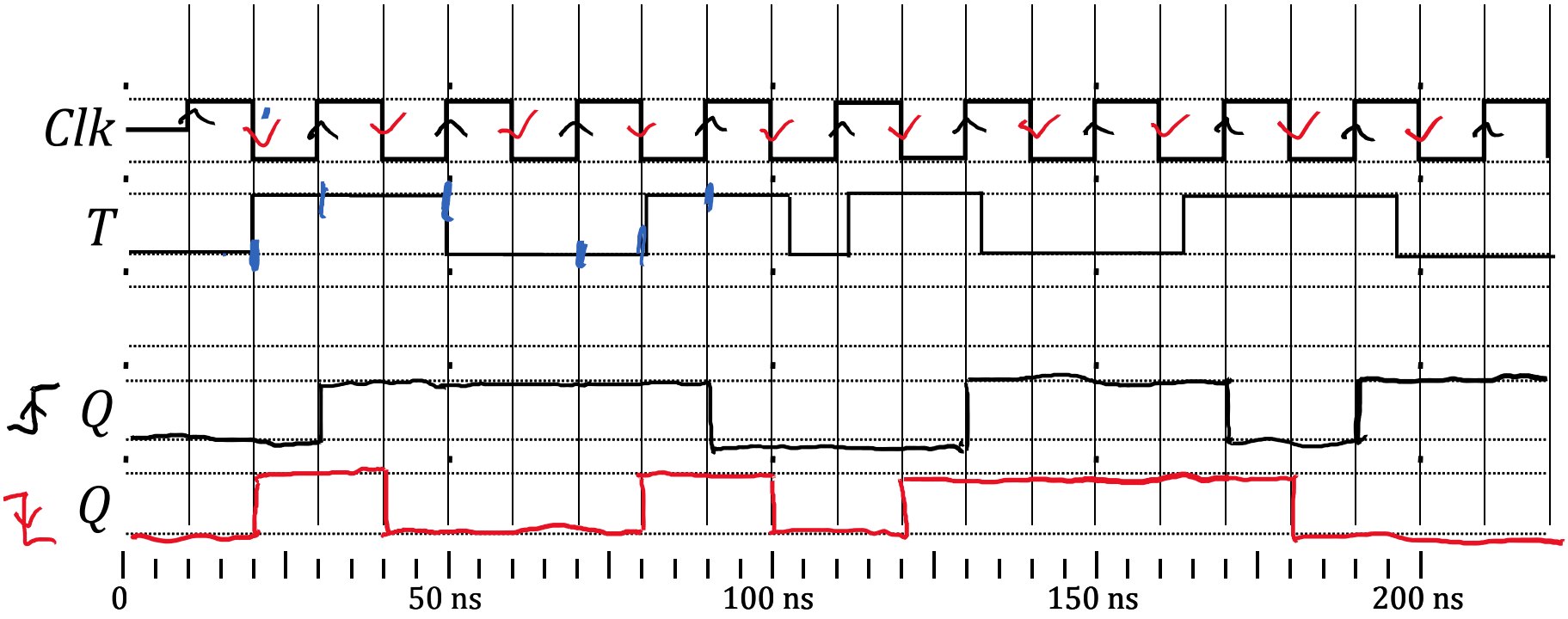






UNSW SYDNEY

T flip-flop



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

UNSW SYDNEY

Characteristic tables

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

It defines the next state as a function of the inputs and the presents state. $Q(t+1) = f(x(t), Q(t))$

The next state is the state of the flip-flop after the clock transition and denoted by Q(t+1).

input

present state

The present state is the state of the flip-flop immediately before the clock edge and denoted by Q(t).

Characteristic tables

J	K	$Q(t+1)$	
0	0	$Q(t)$	unchanged
0	1	0	Reset
1	0	1	Set
1	1	$\overline{Q}(t)$	Complement

D	$Q(t+1)$	
0	0	Reset
1	1	Set

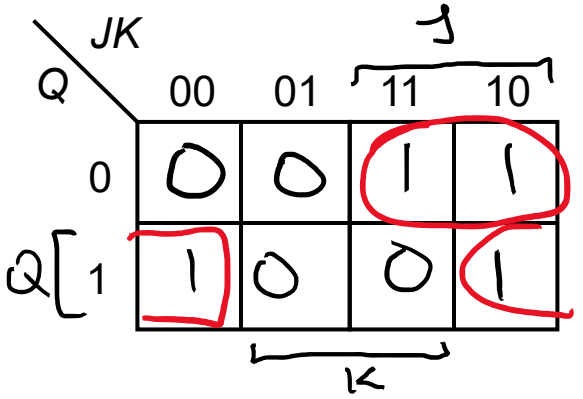
T	$Q(t+1)$	
0	$Q(t)$	unchanged
1	$\overline{Q}(t)$	toggle / complement.

Characteristic equations

Present / Next

Q	J	K	$Q(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$Q(t+1)$:



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

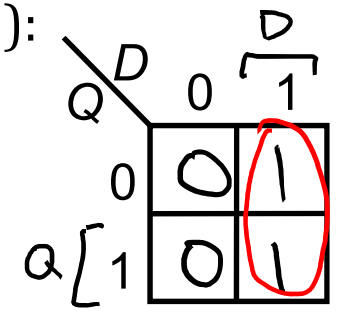
J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

UNSW SYDNEY

Characteristic equations

Q	D	$Q\,(t+1)$	
0	0	0	Reset
0	1	1	Set
1	0	0	Reset
1	1	1	Set

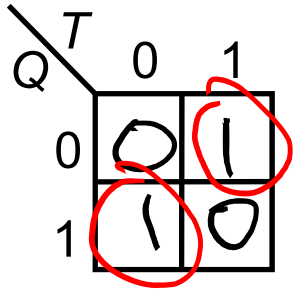
$Q(t+1)$:



$Q(t+1) = D$

Q	T	$Q\,(t+1)$	
0	0	0	unchanged
0	1	1	toggle
1	0	1	unchanged
1	1	0	toggle

$Q(t+1)$:



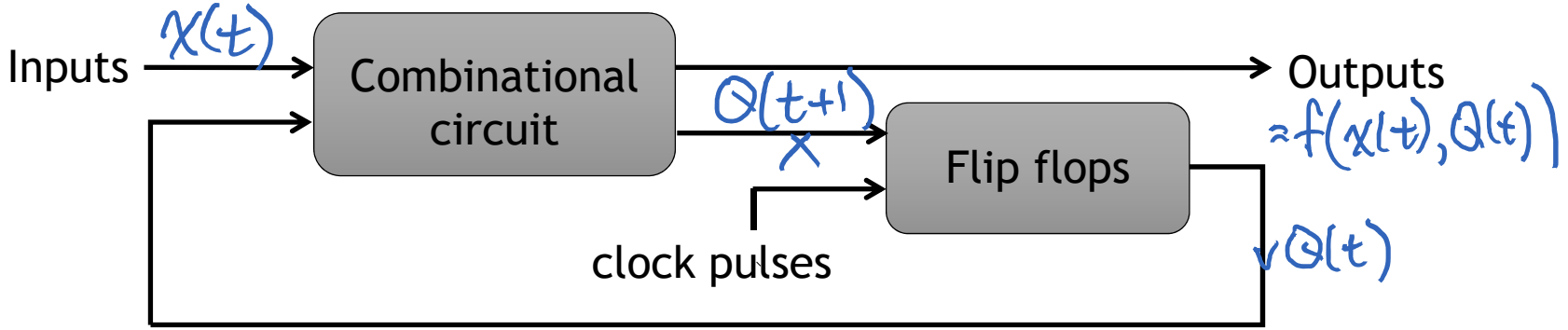
$Q(t+1) = \bar{T}Q + T\bar{Q}$

$= T \oplus Q$

UNSW SYDNEY

Analysis of clocked sequential circuits

The behavior of a clocked sequential circuit is determined from the inputs, internal states and outputs.



The outputs and the next state are both a function of the inputs and the present state

Sequential circuit are analysed by obtaining a table or a diagram for the inputs, internal states and outputs

Boolean expressions can then be used to describe the behaviour of the sequential circuit

UNSW SYDNEY

Analysis of clocked sequential circuits

State equations, a state table and a state diagram are used to describe the behavior of any clocked sequential circuit

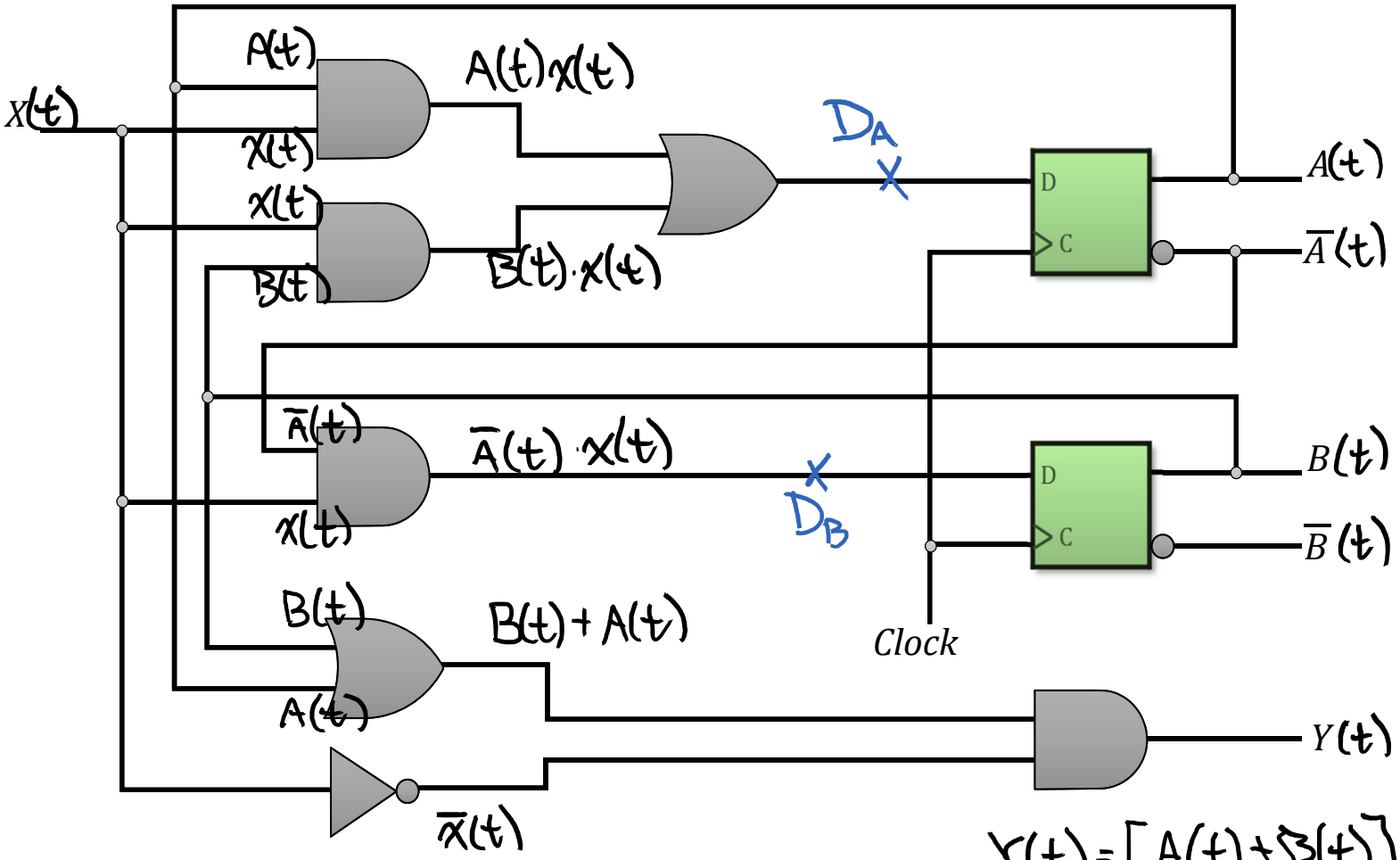
A *state equation* (*transition equation*) specifies the next state as a Boolean function of the present state and inputs

State eqn. $\boldsymbol{Q(t+1) = Q(t)x(t) + \bar{Q}(t)\big(y(t) + x(t)\big)}$

logic func. $\boldsymbol{W(t) = x(t) + Q(t)y(t)}$ - purely comb. cir.

State equations

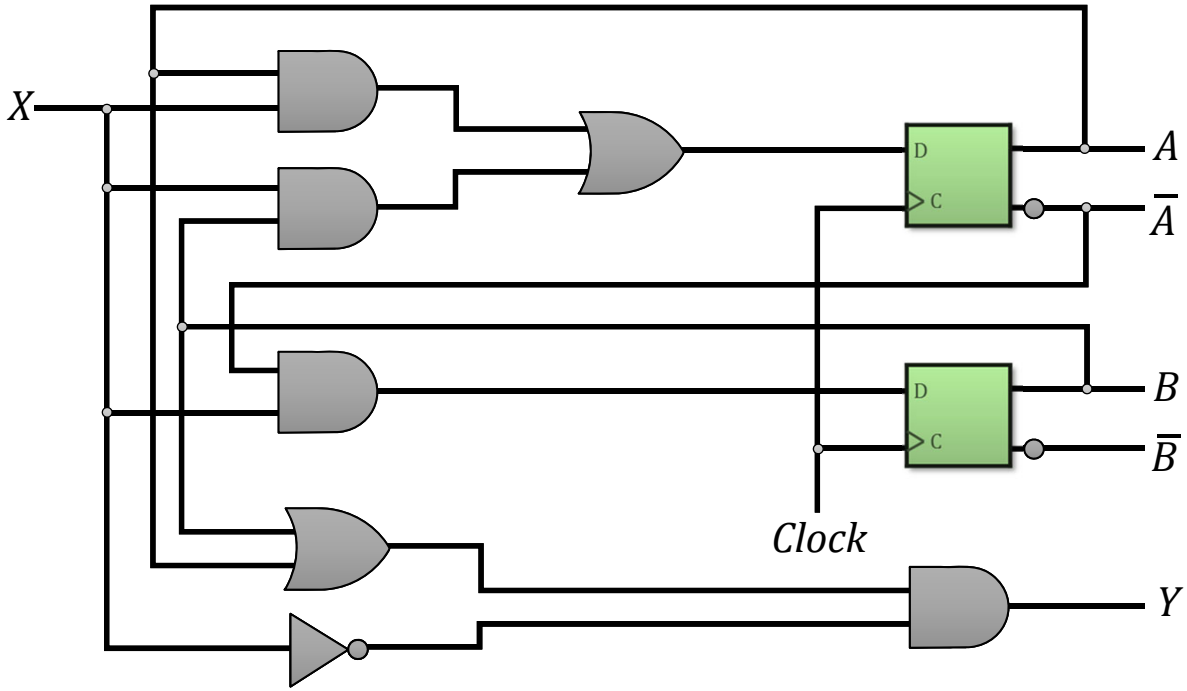
$D_A = A(t+1) = [A(t) + B(t)]\,x(t)$



$D_B = B(t+1) = \bar{A}(t)\cdot x(t)$

$Y(t) = [A(t) + B(t)]\,\bar{x}(t)$

State equations



The D input of a flip-flop determines the value of the next state (i.e. the state after the clock transition)

$$\boldsymbol{A(t+1) = A(t)x(t) + B(t)x(t)}$$

$$\boldsymbol{B(t+1) = \overline{A(t)}x(t)}$$

The present-state of the output is given by

$$\boldsymbol{Y(t) = [A(t) + B(t)]\overline{x(t)}}$$

State Equations

Since all variables in the Boolean expressions are a function of the present state, the designation of (t) after each variable can be omitted

$$\boldsymbol{A(t+1) = Ax + Bx}$$

$$\boldsymbol{B(t+1) = \overline{A}x}$$

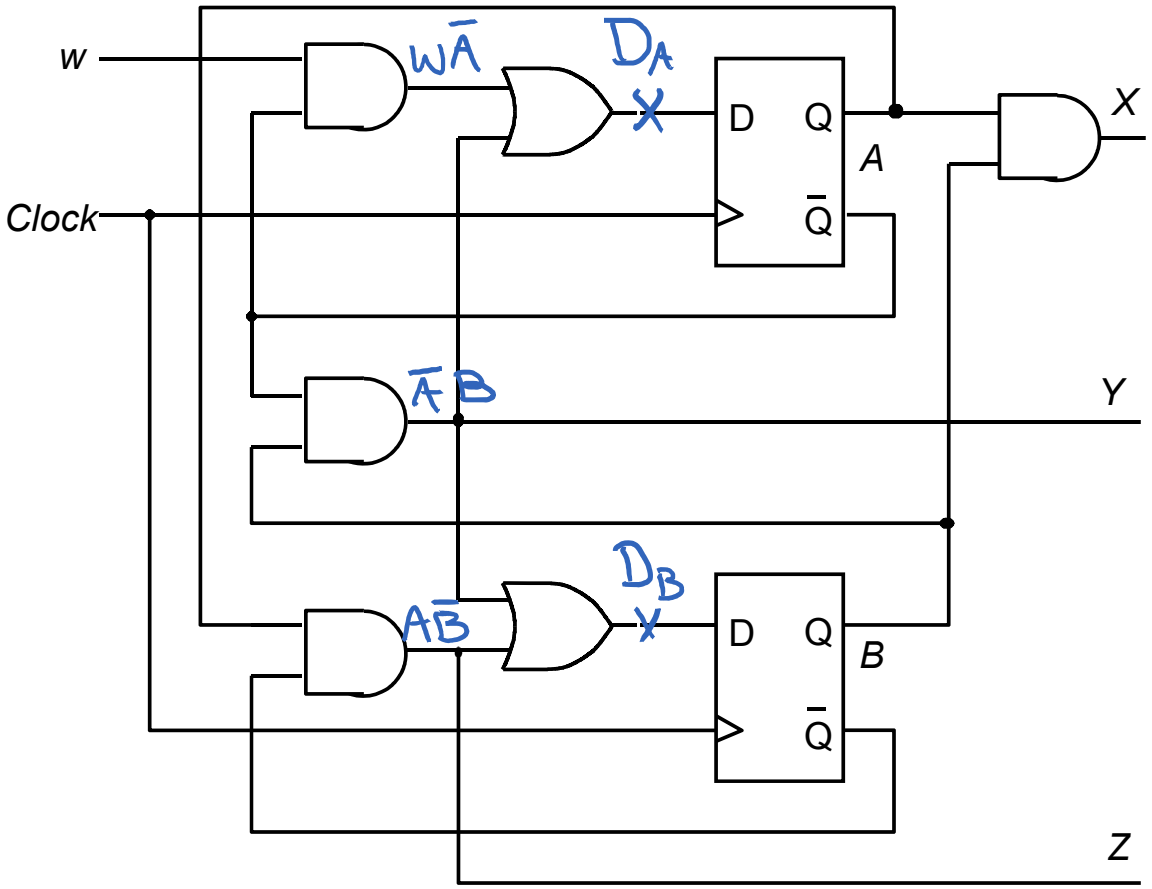
The present-state value of the output can be expressed algebraically as

$$\boldsymbol{Y(t) = [A + B]\overline{x}}$$

Problem

$A(t+1) = D_A = w\bar{A} + \bar{A}B$

$B(t+1) = D_B = \bar{A}B + A\bar{B}$



$X(t) = AB$

$Y(t) = \bar{A}B$

$Z(t) = A\bar{B}$

UNSW SYDNEY

State table

It is possible to enumerate the time sequences of inputs, outputs and states of the sequential circuit using a *state table* (*transition table*)

The table consists of four sections labelled present state, input, next state and output

The derivation of a state table requires listing all possible binary combinations of present states and inputs

The next-states and the output values are then determined from the logic diagram or state equations for each possible binary combination

State table

Present state		Input	Next state		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

$$A(t+1) = Ax + Bx$$
$$B(t+1) = \overline{A}x$$
$$Y(t) = [A + B]\overline{x}$$

State table

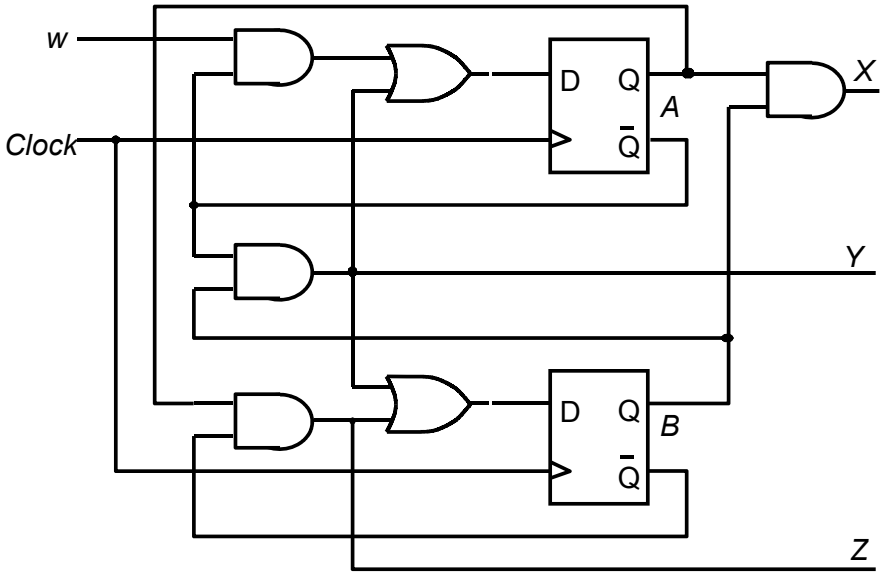
The state transition can be conveniently expressed in a slightly different form having only three sections: present state, next state and output

Present state		Next state				Output	
		X=0		X=1		X=0	X=1
A	B	A	B	A	B		
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

$$A(t+1) = Ax + Bx$$
$$B(t+1) = \overline{A}x$$
$$Y(t) = [A + B]\overline{x}$$

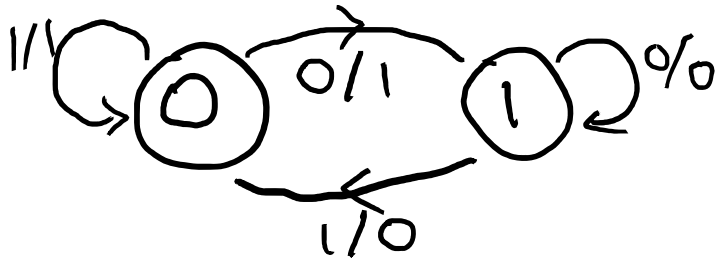
Problem

$$A(t+1) = \bar{A}w + \bar{A}B$$
$$B(t+1) = A\bar{B} + \bar{A}B$$
$$X(t) = AB$$
$$Y(t) = \bar{A}B$$
$$Z(t) = A\bar{B}$$



Present state		Next state				Output					
						X		Y		Z	
		w=0		w=1							
A	B	A	B	A	B						
0	0	0	0	1	0	0		0		0	
0	1	1	1	1	1	0		1		0	
1	0	0	1	0	1	0		0		1	
1	1	0	0	0	0	1		0		0	

State diagram

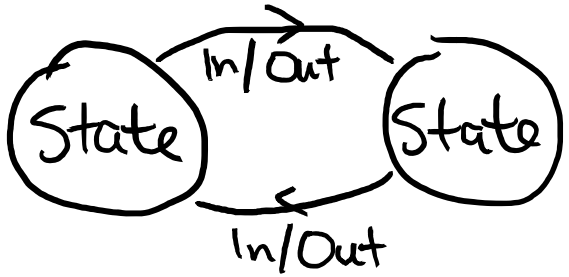


The information available in state tables can be represented graphically in the form of a state diagram

In a state diagram, a state is indicated by a circle and the transitions between the states are represented by directed lines connecting the circles

The directed lines are labeled with two binary numbers separated by a slash

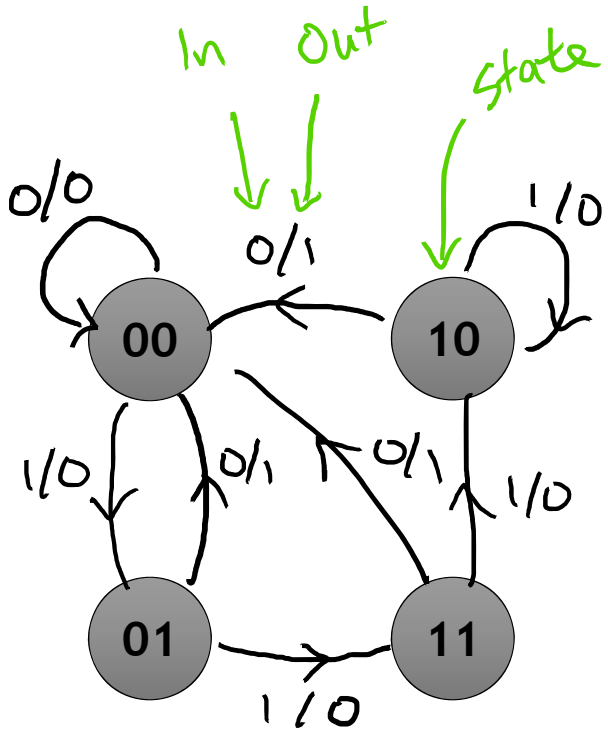
The first binary number is for the input value and the second number shows the output value for the given input value during the present state which is represented by a circle from which the directed line emanates



UNSW SYDNEY

State diagram

Present state		Next state				Output	
		$X=0$		$X=1$		$X=0$	$X=1$
A	B	A	B	A	B		
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



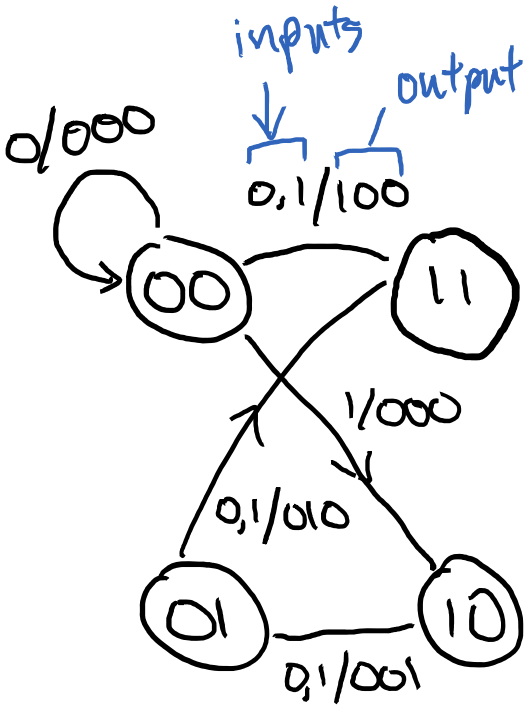
The same information is conveyed by both state diagram and state table.

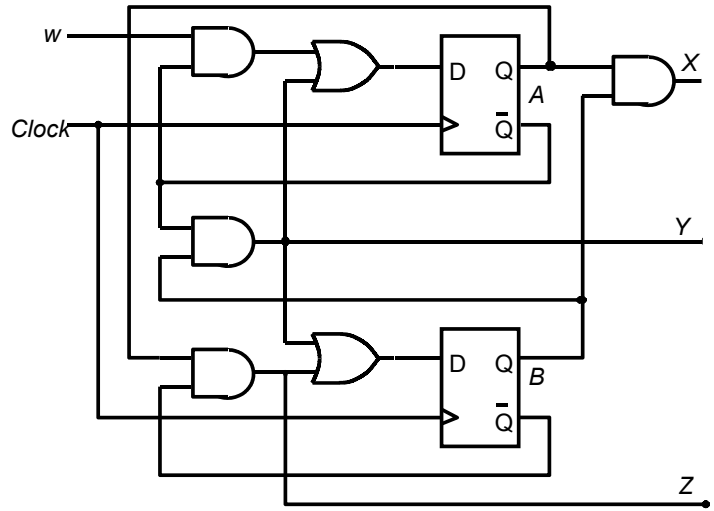
State diagrams provides pictorial representation that is easier for human representation.

UNSW SYDNEY

Problem

Present state		Next state				Output		
		w=0		w=1				
A	B	A	B	A	B	X	Y	Z
0	0	0	0	1	0	0	0	0
0	1	1	1	1	1	0	1	0
1	0	0	1	0	1	0	0	1
1	1	0	0	0	0	1	0	0



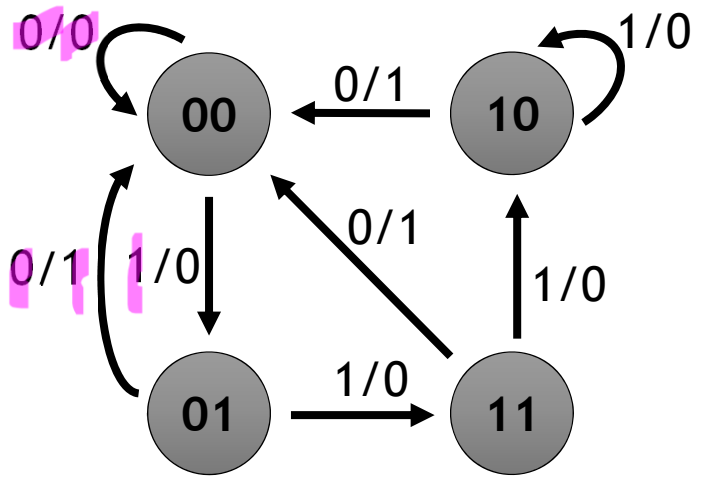


UNSW SYDNEY

State and output time sequence

The state table and state diagram can be used to find state and output time sequences

Here for an input sequence of 010100110 assuming the sequential is initially in state 00, the state and outputs are



Present state		Next state				Output	
		$X=0$		$X=1$		$X=0$	$X=1$
A	B	A	B	A	B		
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

X	0	1	0	1	0	0	1	1	0	0
A	0	0	0	0	0	0	0	1	0	0
B	0	1	0	1	0	0	1	1	0	0
Y	0	0	1	0	1	0	0	0	1	0

UNSW SYDNEY

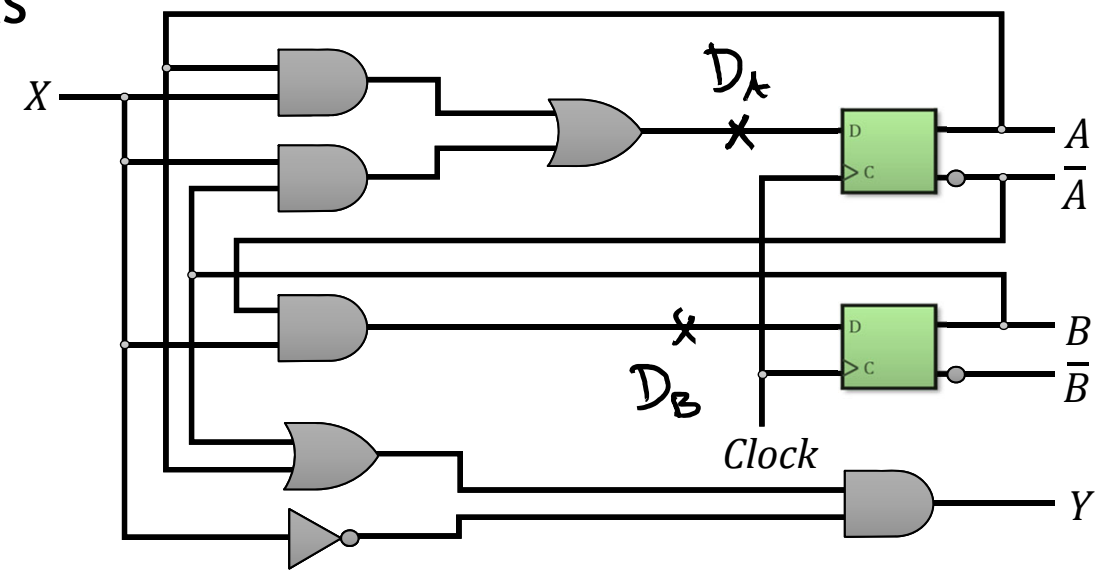
Flip-flop input equations

Flip-flop equations are the Boolean functions that describes the part of the sequential circuit that generate the flip flop inputs

$$\boldsymbol{D_A = Ax + Bx}$$

$$\boldsymbol{D_B = \overline{A}x}$$

$$\boldsymbol{Y = [A + B]\overline{x}}$$



The three equations provide the necessary information for drawing the logic diagram of the sequential circuit

$\boldsymbol{D_A}$ specifies a D flip-flop labeled A and $\boldsymbol{D_B}$ the D flip-flop labeled B

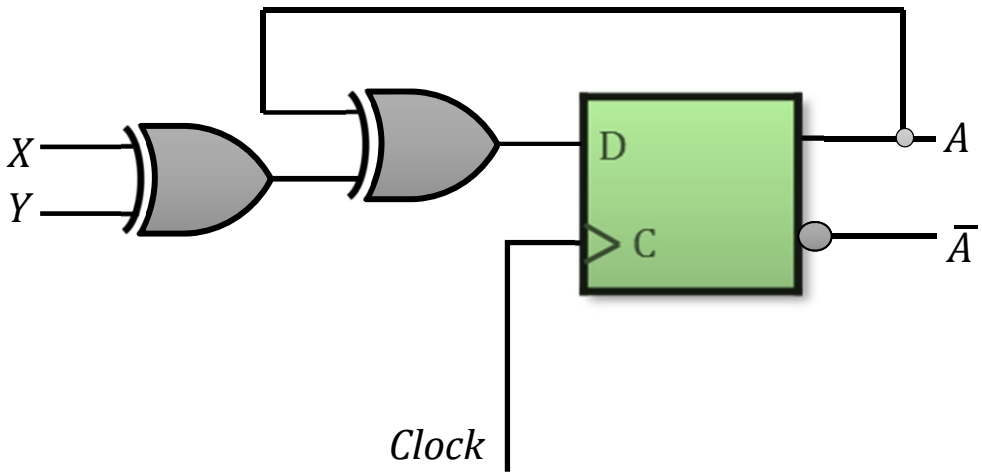
UNSW SYDNEY

Analysis with D flip-flops

Steps in analyzing sequential circuits with D flip-flops:

1. Find the flip-flop input equations
2. The state equations are the same as the flip-flop input equations for D flip-flops
3. Derive the state table from the state equations
4. Draw the state diagram from the state table

Example:



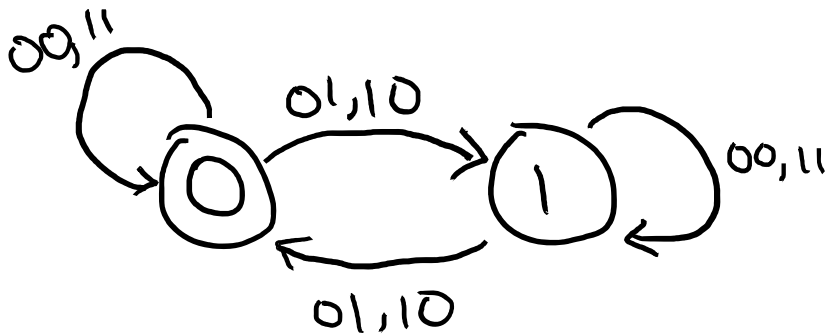
UNSW SYDNEY

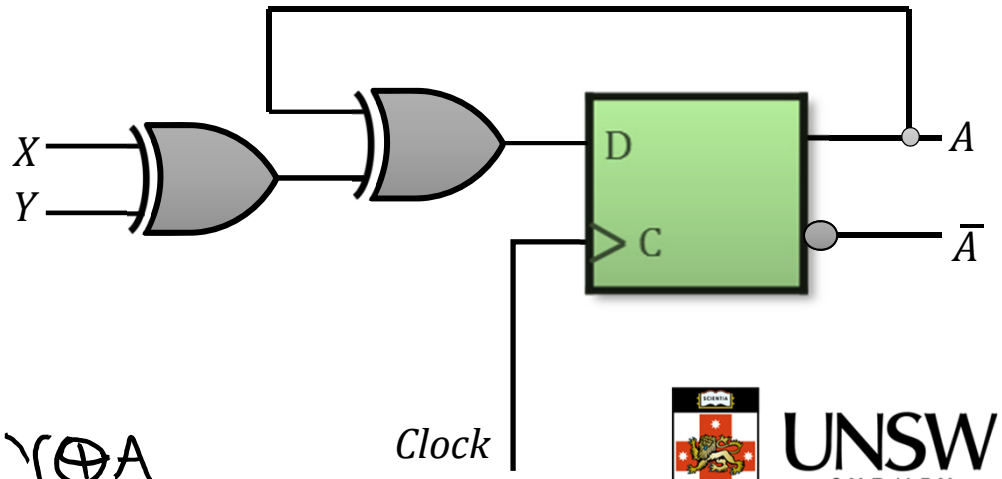
Analysis with D flip-flops

State table:

Present state	Input		Next state
A	X	Y	A
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

State Diagram:





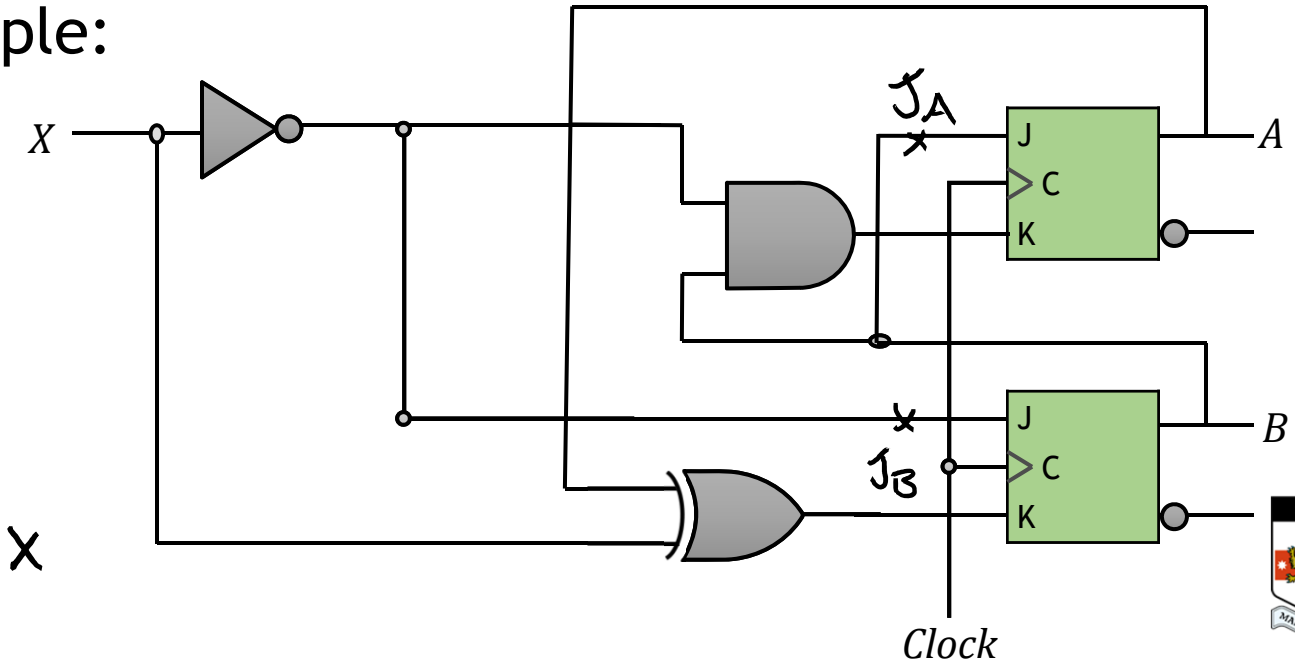
$A(t+1) = X \oplus Y \oplus A$

Analysis with JK flip-flops

Steps in analyzing sequential circuits with JK flip-flops:

1. Find the flip-flop input equations in terms of present state and input
2. List the binary values of each input equation
3. Use the JK flip-flop characteristic table to determine the next states in the state table
4. Use K-map to obtain minimized state equations

Example:



$J_A = B$

$J_B = \bar{X}$

$K_A = B\bar{X}$

$K_B = A \oplus X$

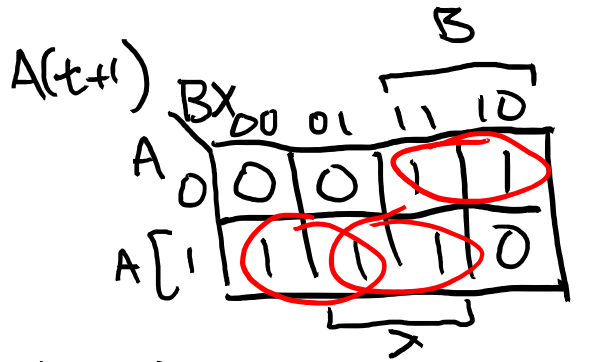
UNSW SYDNEY

Analysis with JK flip-flops

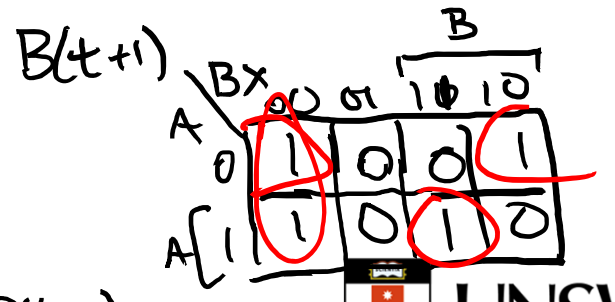
Flip-flop input equations:

$$J_A = B \qquad K_A = B\bar{X}$$
$$J_B = \bar{X} \qquad K_B = A \oplus X$$

Present state		Input	Next state		Flip-Flop 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	0
1	0	1	1	0	0	0	0	1
1	1	0	0	0	1	1	1	0
1	1	1	1	1	1	0	0	1



$$A(t+1) = \bar{A}B + A\bar{B} + AX$$



$$B(t+1) = \bar{A}\bar{X} + ABX + \bar{B}\bar{X}$$

UNSW SYDNEY

Analysis with JK flip-flops

The steps can also be stated as:

1. Find the flip-flop input equations in terms of present state and inputs
2. Substitute the input equations into the flip-flop characteristic equation to obtain the state equations
3. Use the corresponding state equations to determine the next-values in the state table and state diagrams

In the previous example:

$$J_A = B \qquad K_A = B\bar{X}$$

$$J_B = \bar{X} \qquad K_B = A \oplus x$$

The characteristic equations for the JK flip-flop:

$$A(t+1) = J_A\bar{A} + \overline{K_A}A = B\bar{A} + A(\overline{B\bar{x}}) = \bar{A}B + A\bar{B} + Ax$$

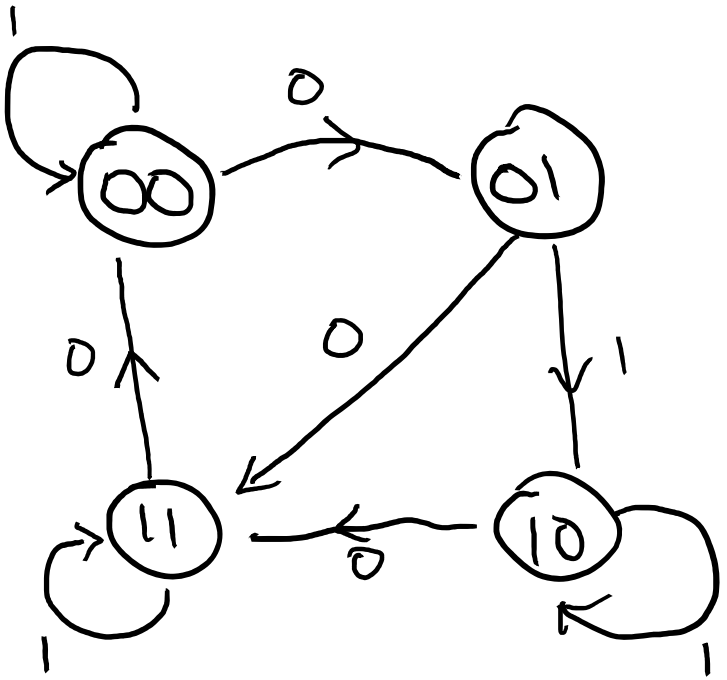
$$B(t+1) = J_B\bar{B} + \overline{K_B}B = \bar{x}\bar{B} + \overline{A \oplus x}B$$

$$= \bar{x}\bar{B} + \bar{A}\bar{x} + ABx$$

Analysis with JK flip-flops

Present state			Next state			
			$X=0$		$X=1$	
A	B		A	B	A	B
0	0		0	1	0	0
0	1		1	1	1	0
1	0		1	1	1	0
1	1		0	0	1	1

State Diagram:



$$A(t+1) = J_A\overline{A} + \overline{K_A}A$$
$$B(t+1) = J_B\overline{B} + \overline{K_B}B$$

UNSW SYDNEY

Analysis with T flip-flops

The same procedure as with JK flip-flops is followed:

1. Find the flip-flop input equations in terms of present state and inputs
2. List the binary values of each input equation
3. Use the T flip-flop characteristic table to determine the next states in the state table
4. Use K-map to obtain minimized state equations

OR

1. Find the flip-flop input equations in terms of present state and inputs
2. Substitute the input equations into the flip-flop characteristic equation to obtain the state equations
3. Use the corresponding state equations to determine the next-values in the state table and state diagrams

UNSW SYDNEY

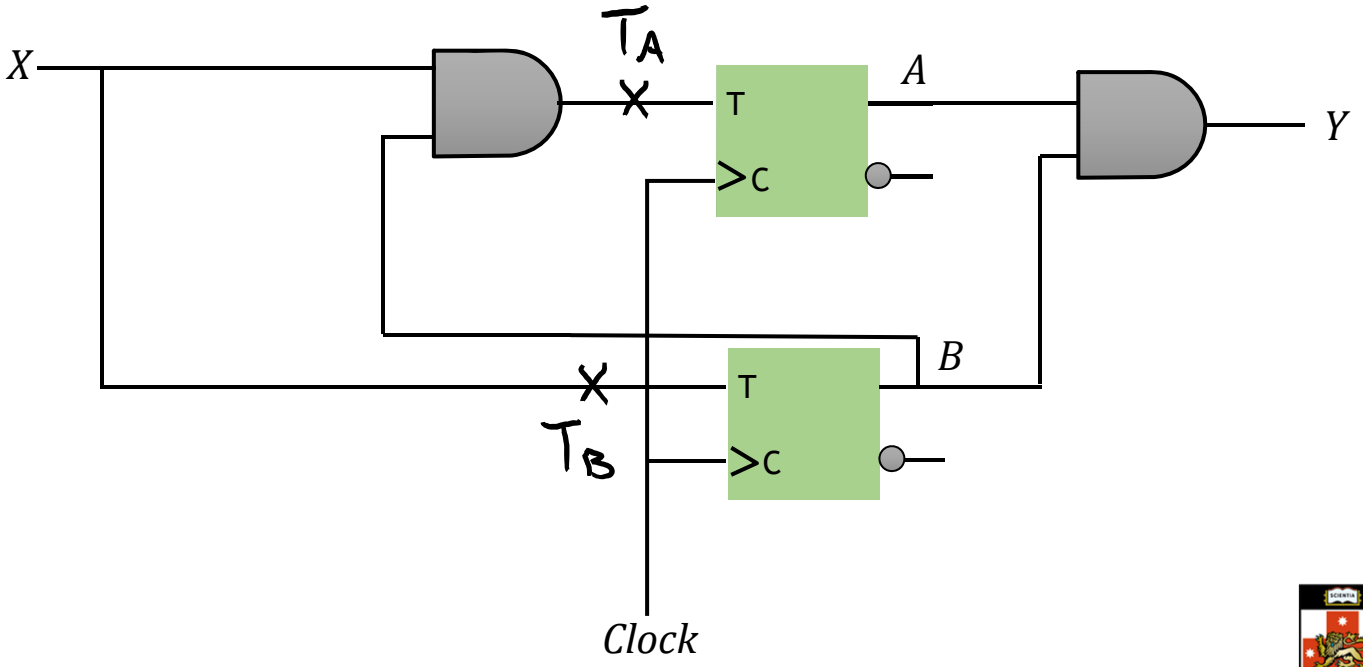
Analysis with T flip-flops

Flip-flop input equations:

$T_A = Bx$ $\quad\quad$ $T_B = x$

Output equation:

$Y = AB$

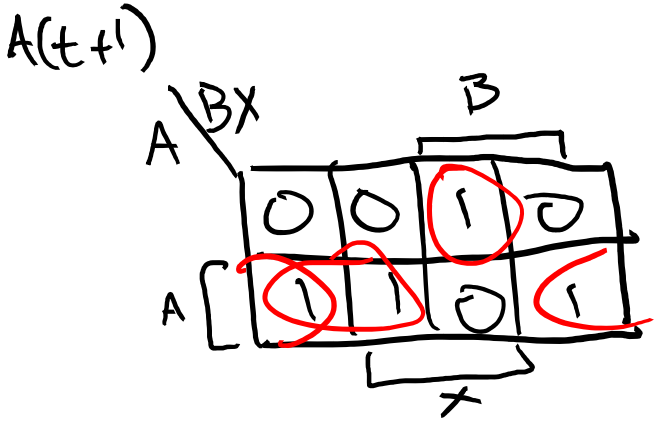


UNSW SYDNEY

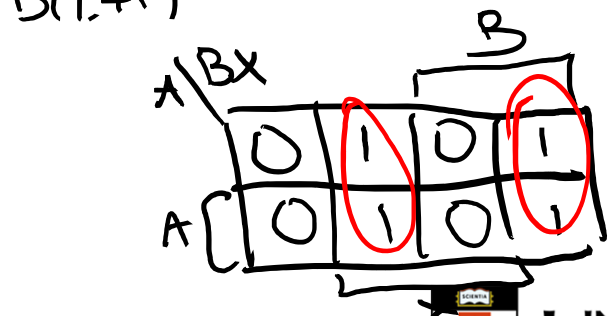
Analysis with T flip-flops

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

Present state		Input	Next state		Flip-Flop inputs	
A	B	X	A	B	T_A	T_B
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	0	1	0	0
0	1	1	1	0	1	1
1	0	0	1	0	0	0
1	0	1	1	1	0	1
1	1	0	1	1	0	0
1	1	1	0	0	1	1



$$A(t+1) = \overline{A}BX + A\overline{B} + A\overline{X}$$



$$B(t+1) = \overline{B}X + B\overline{X} = B \oplus X$$

$$T_A = BX \qquad T_B = X$$

$$Y = AB$$

UNSW SYDNEY

Analysis with T flip-flops

Here

$$T_A = BX$$

$$T_B = X$$

Characteristic equation for T flip-flop:

$$A(t+1) = T_A \oplus A$$

$$B(t+1) = T_B \oplus B$$

Analysis with T flip-flops

Present state	Next state				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	1	1	0	0	0
1 0	1	0	1	1	0	0
1 1	1	1	0	0	1	1

$$A(t+1) = T_A \oplus A$$
$$B(t+1) = T_B \oplus B$$
$$Y = AB$$

State Diagram: