[handwritten: combinational logic vs]

Sequential logic design

Vikas Dhiman for ECE275

October 26, 2022

1 Objectives

1. Building blocks of sequential circuits

2. Analyze a sequential circuit and derive a state-table and a state-graph

3. Derive a state graph or state table from a word description of the problem

4. Understanding the structure of an FPGA

[handwritten: next state table   state table]

2 Why do we need sequential circuits?

Example 1. *Think about this problem: Design an occupancy counter that depends on a sensor S at the class door. The sensor is triggered every time a person passes through the door. The counter can be reset to zero with a reset button. Assume we only need up to two bit counter C_1C_0. Draw a truth table for this circuit. Do you have requisite knowledge for designing this circuit? Can this circuit be designed without a memory element?*

[handwritten table:]

S	C_1	C_0	C_1^+	C_0^+
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

[handwritten: Memory / State of the system]

[handwritten: State/transition table]

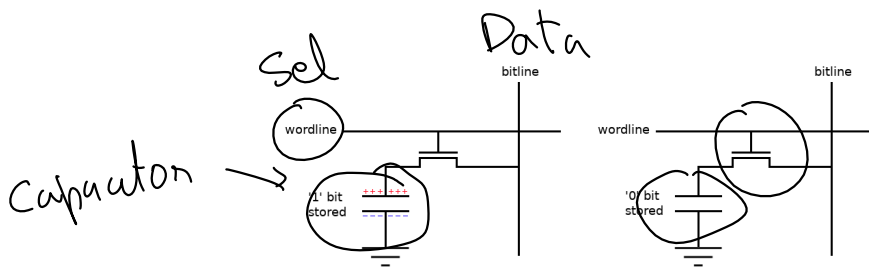## 3 How to create memory element from circuits

Two types of memory

1. Volatile memory. For example, RAM, CPU registers.

2. Non-volatile memory. For example, SSD, Flash drives. (Not covered in this course)

[handwritten: Volatile memory]   *[handwritten: Slowest]*

    (a) Memories that require periodic refreshing. For example, DRAM: Dynamic Random Access memory (Not covered in this course)

(handwritten) Memory refreshing

(handwritten) DRAM

(handwritten) slightly faster

(handwritten, at left) Sel   Data

(figure labels) bitline, wordline, '1' bit stored, ++++++, bitline, wordline, '0' bit stored

(handwritten) Capacitor

1

(b) Memories that are always refreshing. For example, SRAM: Static Random Access memory [2, Appendix B.64]

Sel

Data   *A*   *B* $\overline{Data}$
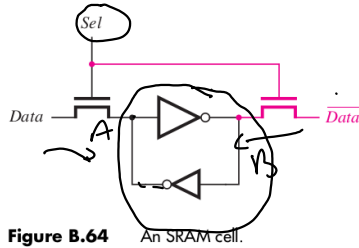
Figure B.64   An SRAM cell.

(handwritten, right side) CPUs
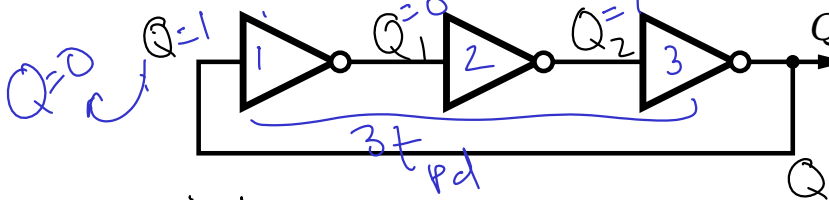
L1-cache

register

fastest (power hungry)

4   Latches and Flip-Flops [1, Sec 3.2]

Example 2 (Ring oscillator). *[1, Sec 3.31] How many stable states does the following circuit have?*

(handwritten annotations on circuit) Q=0   Q=1   1   Q_1=0   2   Q_2=?   3   Q = 0

Q=0

$3t_{pd}$

Q
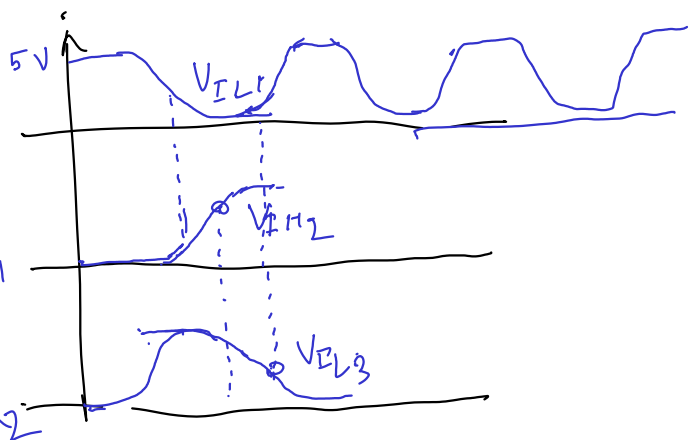
(handwritten) non-ideal

Odd number of NOT gates in a loop you get an oscillator

(handwritten, timing diagram) 5V, V_{IL1}, V_{IH2}, V_{IL3}, Q_1, Q_2

Definition 1 (Astable circuits).

(handwritten) Unstable = no stable states

Example 3. *Analyze the timing diagram of the following circuit.*

$A = 0$

$A = 0$ at inverter 1, $A \geq 0$ ... $A := 1 \equiv$ ... $B = 1$ ... $B = 0 \equiv$ ... $A^+ = 0$, $A^+ = 1$

$A = 0$, $5V$ ... V_{IHI} ... $1 = 5V$ ... 0

A

$B = \bar{A}$ ... $H = 5V$ ... 0

schmitt trigger oscillate

1-bit

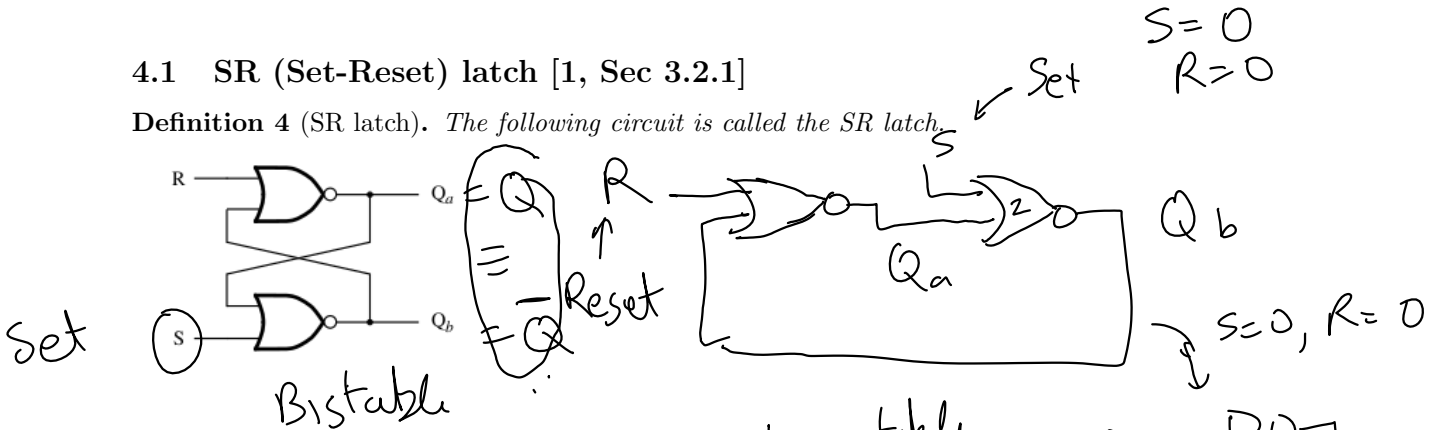**Definition 2** (Bistable circuits).

Two stable state

$A = \bar{B}$

A 0

B $5V$
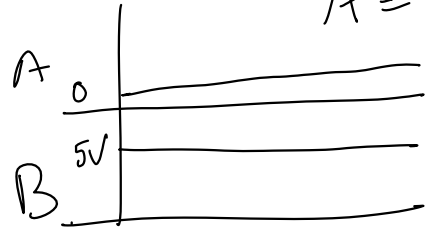
Definition 3 (Characteristic or state table). *Draw the characteristic or state table of the above circuit.*

A	B	A^+
0	1	0
1	0	1

$$A^+ = A(t + t_{pd})$$
$$A = A(t)$$

4.1 SR (Set-Reset) latch [1, Sec 3.2.1]

Definition 4 (SR latch). *The following circuit is called the SR latch.*

$S = 0$
$R = 0$

Set

S

Q_a

R
↑
Reset

$S = 0, R = 0$

Q_a Q_b

Set

Bistable

two stable state

$DO \frac{}{Q_a} DO$ Q_b

1. *How many stable states does this circuit have?* two stable state

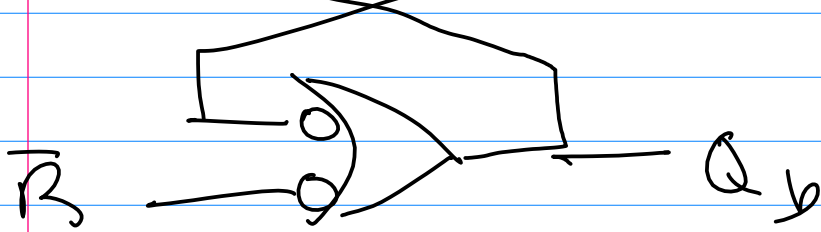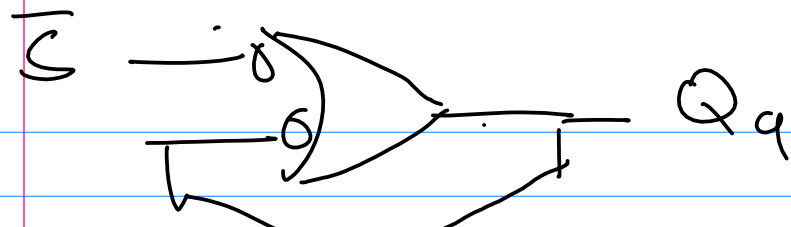2. *Draw its characteristic or state table.*

3. *Draw SR latch symbol*

State table

	S	R	Q_a	Q_b	Q_a^+	Q_b^+
Hold →	0	0	Q_a	$\overline{Q_a}$	Q_a	$\overline{Q_a}$ [3]
Reset →	0	1	d	d	0	1
Set →	1	0	d	d	1	0
Invalid →	1	1	d	d	0	0

NOR_2

S	Q_a	Q_b
0	0	1
0	1	0
1	0	0

NOR_1

Q_b	R	Q_a
0	0	1
1	0	0
0	1	0
1	1	0

$\overline{S}$ ─○

$\overline{R}$ ─

Q_a

Q_b

SR Latch

S ─

R ─

$\overline{Q}_a = \overline{Q}$
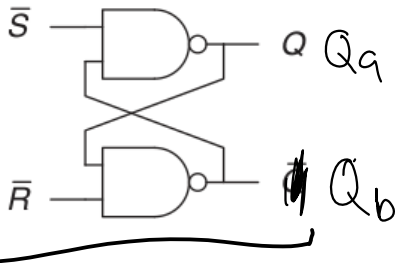
$\overline{Q}_b = Q$

Synchronous (avoid glitches at clocked intervals)

vs

Asynchronous

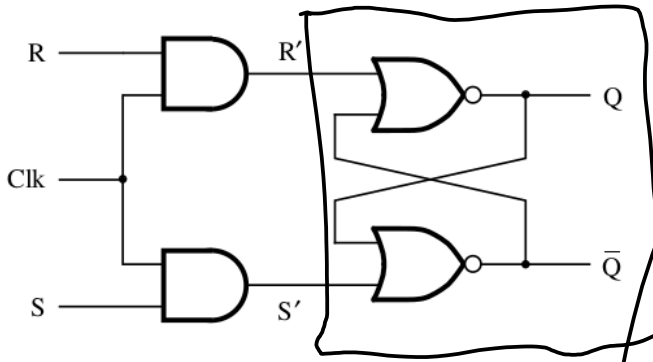**Problem 1** (SR latch using NAND gates). *Draw the characteristic or state table for the following circuit*

S	R	Qa	Qb	Q_a^+	$\bar{Q}_b$
0	0				
0	1				
1	0				
1	1				

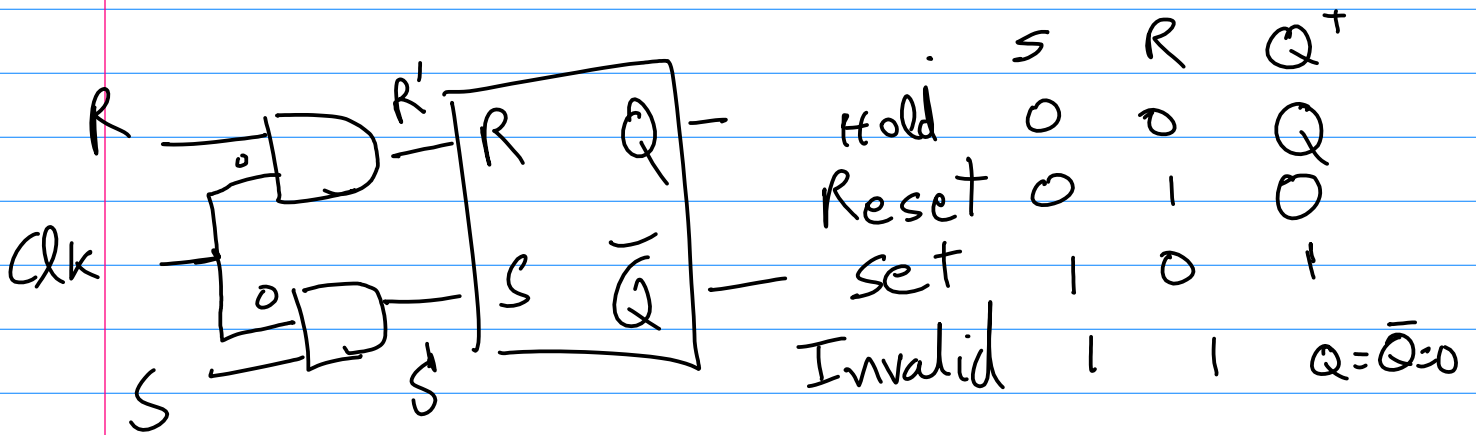## 4.2 Gated SR latch [2, Sec 5.2]

Definition 5 (Gated SR latch). *The following circuit is called the Gated SR latch.*

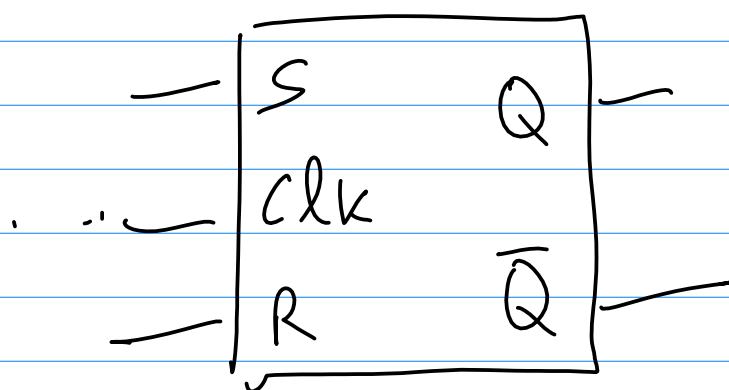
Level triggered

Clock

Later : edge triggered clock

1. *Draw its characteristic table.*

2. *Draw the Gated SR latch symbol*

	S	R	Q^+
Hold	0	0	Q
Reset	0	1	0
Set	1	0	1
Invalid	1	1	$Q=\bar{Q}=0$

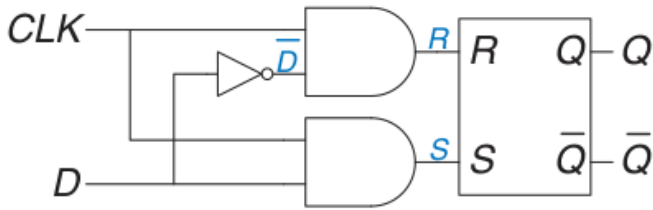| S	R	Clk	S'	R'	Q^+
X	X	0	0	0	Q
0	0	1	0	0	Q
0	1	1	0	1	0
1	0	1	1	0	1
1	1	1	1	1	$Q=\bar{Q}=0$

Hold
Hold
Reset
Set
Invalid

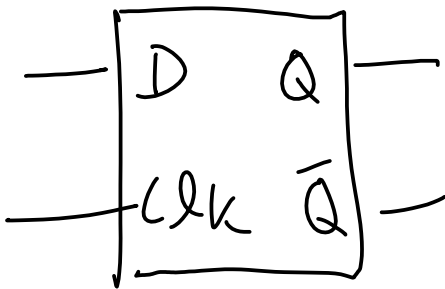Gated SR latch

. Transparent latch → Flip flop

4.3 D (Data) latch [1, Sec 3.2.2]

Definition 6 (D latch). *The following circuit is called the D latch.*

1. Draw its characteristic table.

2. Draw the D latch symbol

Hold
Reset
Set

D	$\bar{D}$	Clk	S	R	Q^+
x	x	0	0	0	Q
0	1	1	0	1	0
1	0	1	1	0	1

D	Clk	Q^+
x	0	Q
D	1	D

4.4 D flip-flop [1, Sec 3.2.2]

Definition 7 (D flip-flop). *The following circuit is called the D flip-flop.*
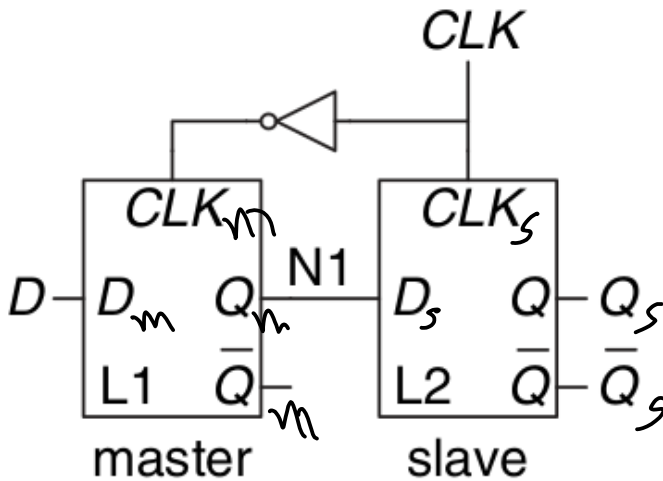
1. *Draw its timing diagram*

2. *Draw its characteristic table.*

3. *Draw the D flip-flop symbol*

Remark 1. *What is the difference between a latch and a flip-flop?*

latches vs flip flops

level triggered on clock | edge triggered

Example 4. *Add a* RESET *signal to the D flip-flop that resets the state of flip-flop to 0.*

Example 5. *The toggle (T) flip-flop has one input, CLK, and one output, Q. On each rising edge of CLK, Q toggles to the complement of its previous value. Draw a schematic for a T flip-flop using a D flip-flop and an inverter.*

Def 7 D- flip flop

CLK_S, CLK_m, D_m, D_S = Q_m, Q_S
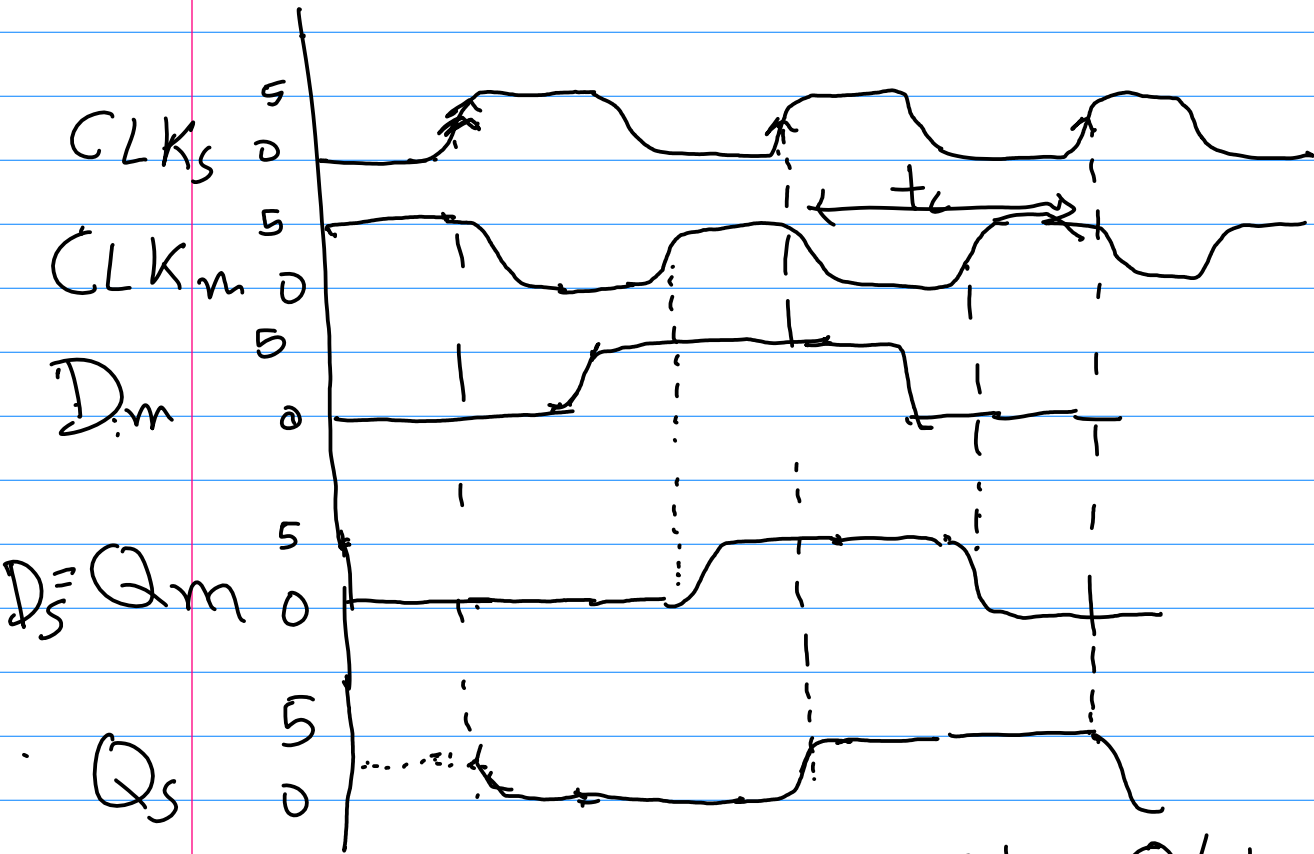
$$Q^+ = Q(t + t_{pd}).$$
$$Q = Q(t)$$

CLK = 0 $Q^+ = Q$

D	Clk ↑	Q^+
D	↑	D
		Q

State table

D flip flop

D	Q^+
0	0
1	1
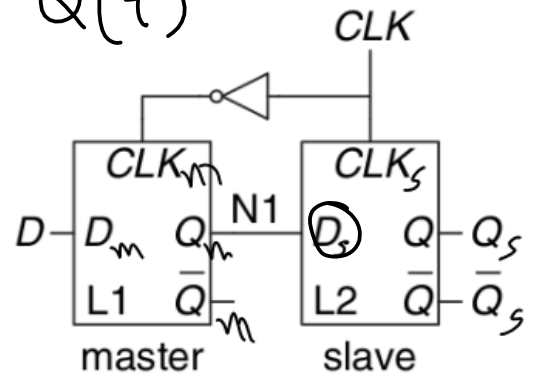
clock period

Synchronous circuit
$$Q^+ = Q(t + t_c)$$

T-flip flop

↑
Toggle

T	Q^+
0	Q
1	$\bar{Q}$

How can we construct T-flip flop from D-flip flop?

T ———⊐
      ⊐XOR⊐—→ D Q
Q ———⊐   >clk $\bar{Q}$

Combination Logic
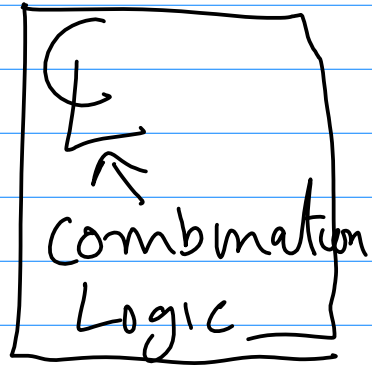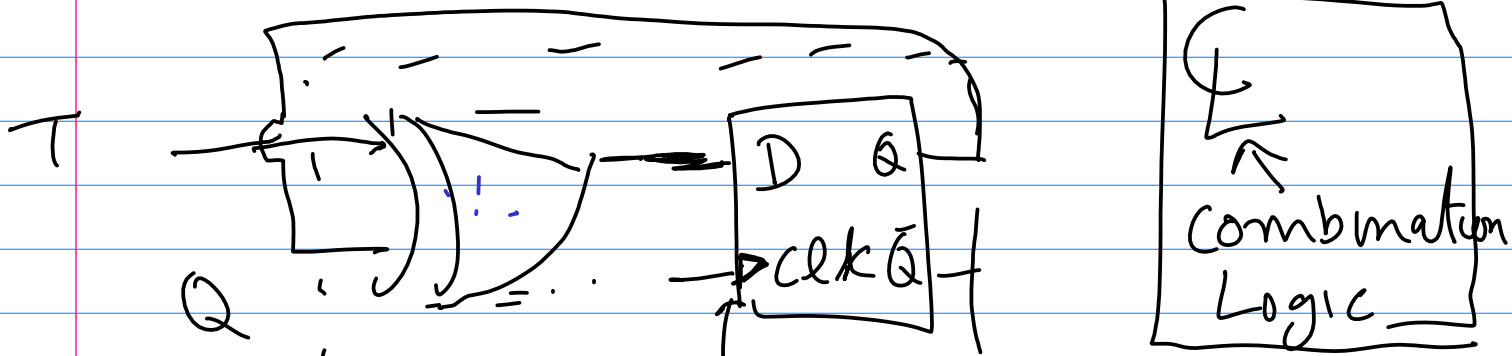
Triangle denotes edge triggered

T	Q		D		Q^+
0	0		0		0
0	1		1		1
1	0		1		1
1	1		0		0

XOR

Q Q⁺ | D

Q	Q⁺	D
* 0	0
✦	1	1

J K Q | D
?

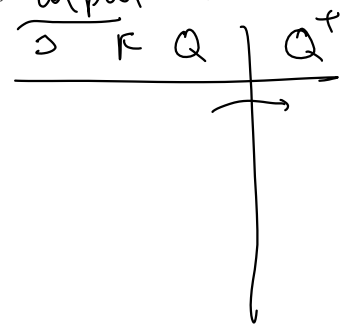states, next state

input

① J K Q | Q⁺

J	K	Q	Q⁺

Problem 2. *A JK flip-flop receives a clock and two inputs, J and K. On the rising edge of the clock, it updates the output, Q. If J and K are both 0, Q retains its old value. If only J is 1, Q becomes 1. If only K is 1, Q becomes 0. If both J and K are 1, Q becomes the opposite of its present state.*

1. *Construct a JK flip-flop using a D flip-flop and some combinational logic.*

2. *Construct a D flip-flop using a JK flip-flop and some combinational logic.*

3. *Construct a T flip-flop (see Exercise 3.9) using a JK flip-flop.*

5 Finite State Machines [1, Sec 3.4]

2

Example 6. *Design an occupancy counter that depends on a sensor S at the class door. The sensor is triggered every time a person passes through the door. Assume that the counter starts at zero. Assume we only need up to two bit counter $C_1 C_0$. Draw a state table for this circuit.*

Problem 3. *A divide-by-N counter has one output and no inputs. The output Y is HIGH for one clock cycle out of every N. In other words, the output divides the frequency of the clock by N. The waveform for a divide-by-3 counter is shown here:*

(a)
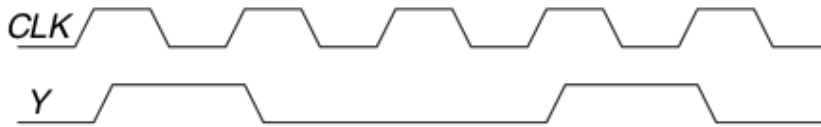
Sketch circuit designs for such a counter

Problem 4. *Design a 3-bit counter which counts in the sequence: 001, 011, 010, 110, 111, 100, (repeat) 001, ...*

Example 7. *Design an odd-even counter for an single bit input. The output of this circuit should be 1 if the number of 1s to the input have been odd so far and 0 otherwise.*

Example 8 (Sequence detectors)**.** *A sequential circuit has one input and one output. The output becomes 1 and remain 1 thereafter when at least two 0's and at least two 1's have occurred as inputs regardless of the order of*

Example 9. *Consider the problem of inventing a controller for a traffic light at a busy intersection on campus. There are two traffic sensors, T_A and T_B , on Academic Ave. and Bravado Blvd., respectively. Each sensor indicates TRUE if students are present and FALSE if the street is empty. There are two traffic lights, L_A and L_B, to control traffic. Each light receives digital inputs specifying whether it should be green, yellow, or red. When the system is reset, the lights are green on Academic Ave. and red on Bravado Blvd. As long as traffic is present on Academic Ave., the lights do not change. When there is no longer traffic on Academic Ave., the light on Academic Ave. becomes yellow for 5 seconds before it turns red and Bravado Blvd.'s light turns green. Similarly, the Bravado Blvd. light remains green as long as traffic is present on the boulevard, then turns*

2These notes will not fit on your note sheet.

yellow and eventually red.
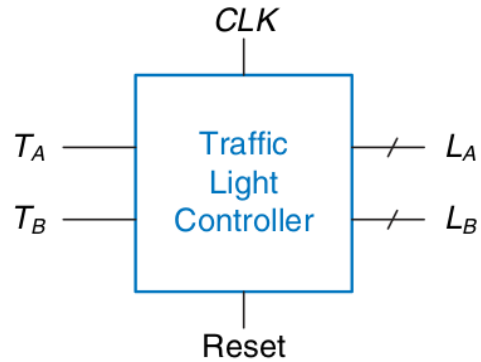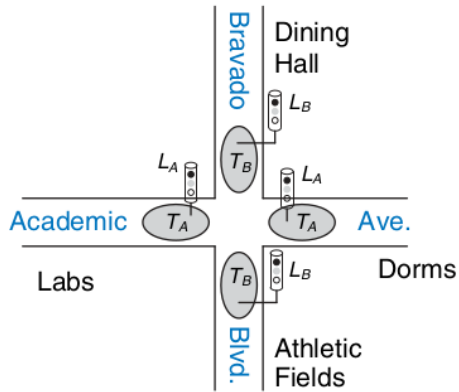
1. *Draw a state transition diagram*

2. *Draw a state table*

3. *Assign binary encodings to each of the states*

4. *Redraw the state table with binary encodings. Design a minimal SOP boolean expression.*

5. *Assign binary encodings to each of the output and redraw the output table. Design a minimal SOP boolean expression for the outputs.*

Problem 5. *Design a circuit for a 2x2 pixel resolution pong game, where the ball can only occupy 4 possible pixels and a single paddle occupies another 2 pixels. The ball bounces of the paddle when the paddle is in the correct row. To keep it interesting, the ball takes a different path from the source path. Track the score with a single bit counter.*

References

[1] Sarah L Harris and David Harris. *Digital design and computer architecture.* Morgan Kaufmann, 2022.

[2] Brown Stephen and Vranesic Zvonko. *Fundamentals of digital Logic with Verilog design.* McGraw Hill, 2022.

Example 6 : occupancy counter

D- flip flop

+ve edge triggered
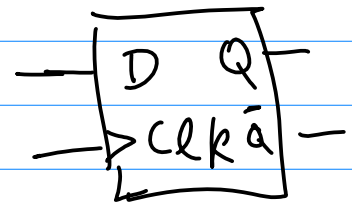
-ve edge triggered

Characteristic table of D flip flop?

Clk	D	Q	Q⁺
↑	0	d	0
	1	-d	1
	d	◎	0
		1	1

$Q^+ = Q(t + t_c)$

$Q = Q(t)$

D	Q⁺
0	0
1	1

characteristic table

Sensor

State transition table
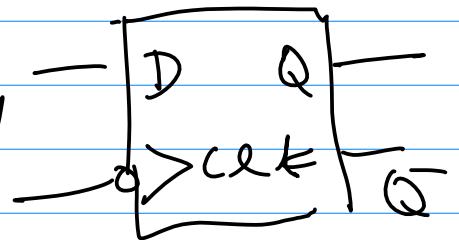
Sensor	State		Next State	
S	C_1	C_0	C_1^+	C_0^+
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

counter stay same

Increment the counter

State transition diagram
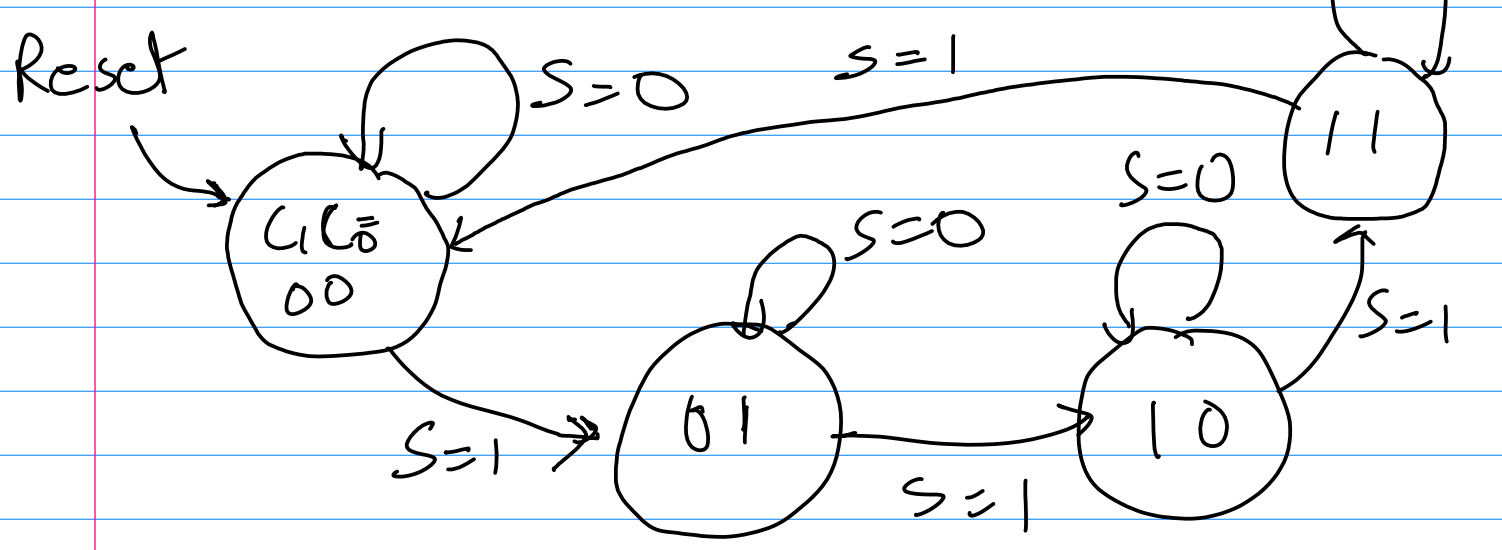
① Identify what states will you need

② How are you transitioning b/w the states

How to design a circuit from state transition table?

Synchronous sequentic circuit template

2-bit state

input

CL

CLK

Register

input to the circuit

CL

input to FF

State of the circuit

─── = bus

S
Sensor

P_0

D_0 Q_0
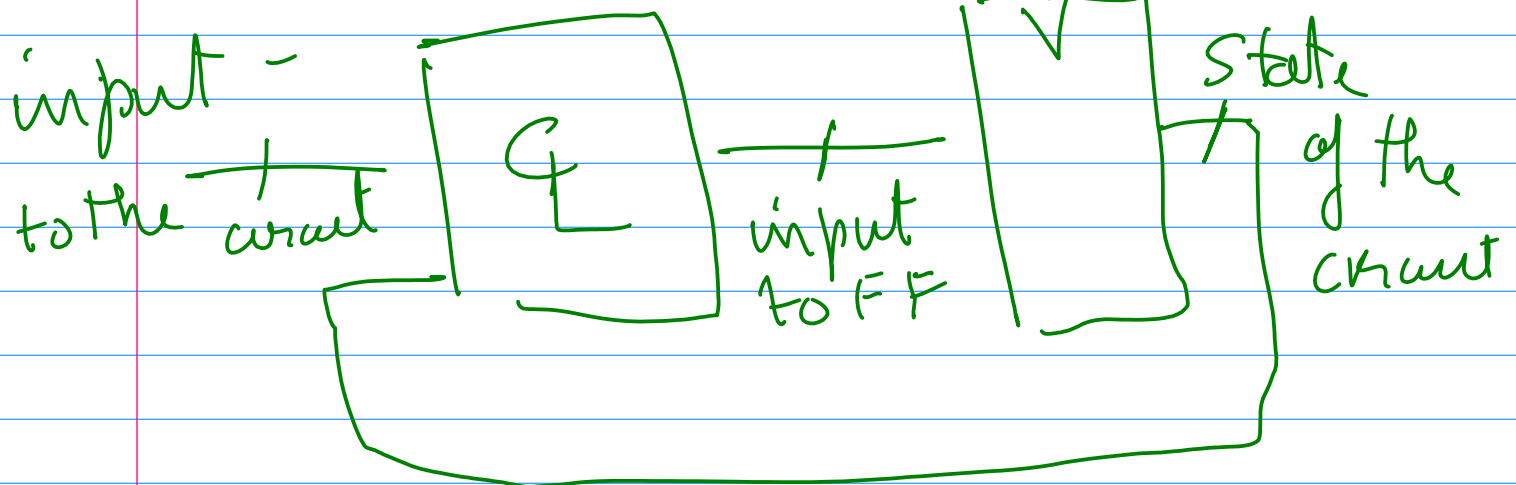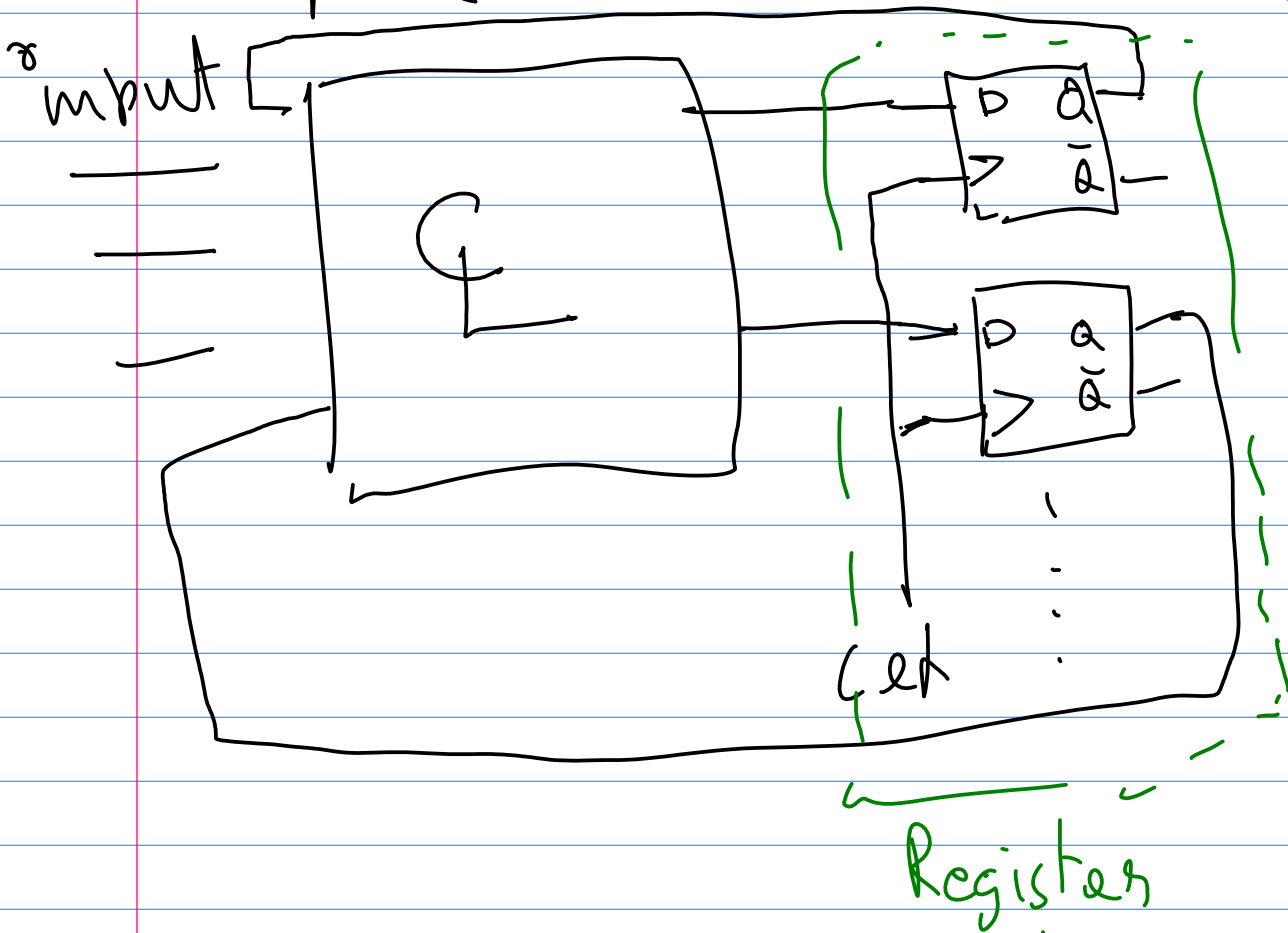
C_0

$\overline{Q_0}$

C_0

G

P_1

D_1 Q_1

C_1

C_1

Q_1

Clk
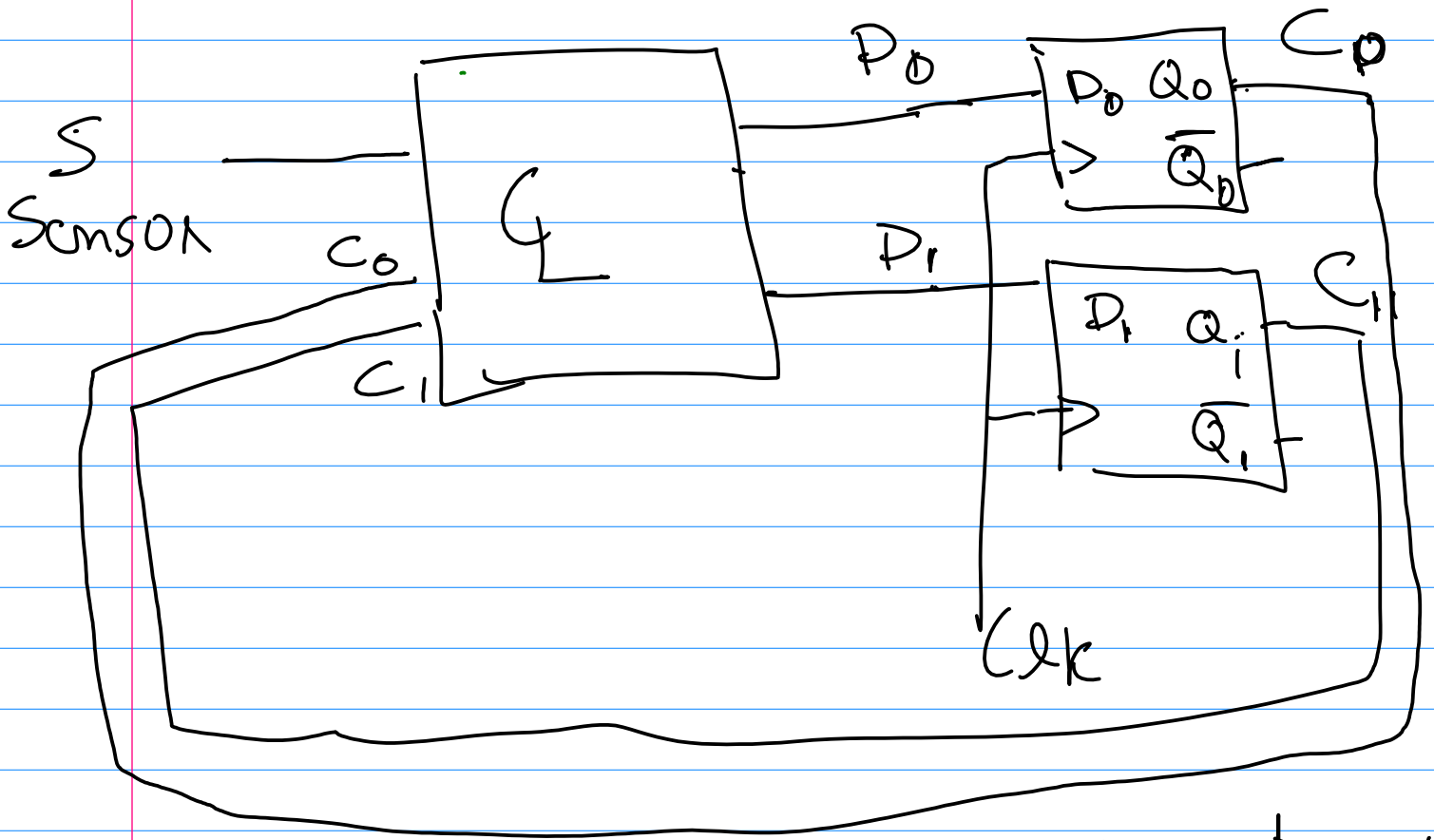
characteristic table

S	C_1	C_0		D_1	D_0
				?	?

D_0	Q_0	Q_0^+
0	d	0
1	d	1

C_1	C_0	C_1^+	C_0^+
0	0	0	1

↳ Excitation table

Q	Q^+		D
0	0		0
0	1		1
1	0		0
1	1		1

cursor → State   ex! --- e

S	C_1	C_0	C_1^+	C_0^+

0

0	0	0	0	0
0	0	1	1	0
0	0	2	1	0

4

| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 |

D_0

Truth table

S. Q_1	$= C_1$	Q_0 $= C_0$	D_1	D_0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	1	0

		S		
D_1 \	0	2	6	4
	0	1	1	0
	1	3	7	5
C_0 [	0	1	0	1

C_1

$$D_1 = \overline{C_1} S$$
$$+ C_1 \overline{C_0}$$
$$+ S C_0 \overline{C_1}$$

D- flip flop
(Data)

D-flip flop block diagram with inputs D, Clk and outputs Q, Q̄

Timing diagram showing Clk, D, and Q waveforms

Ex 7   Odd - even counter

x ———→ | Seq Circuit | ——→ f   1 if odd
clk —————→   ⟳ Reset   0 if even

clk ⎍⎍⎍⎍⎍⎍⎍

x _____⎍_⎍_⎍___   0

Increment
the counter!

f _____⎍___⎍____
        odd even odd
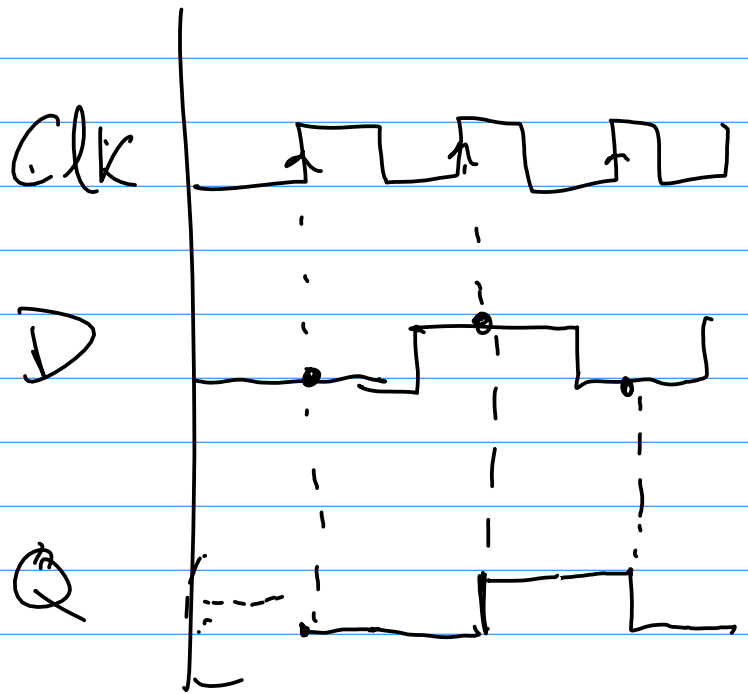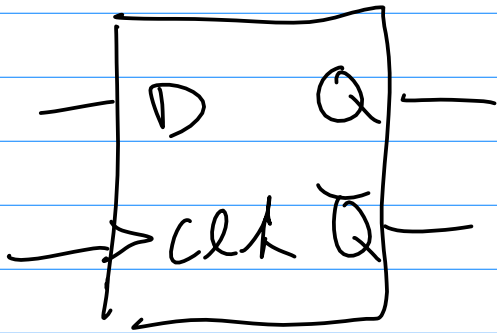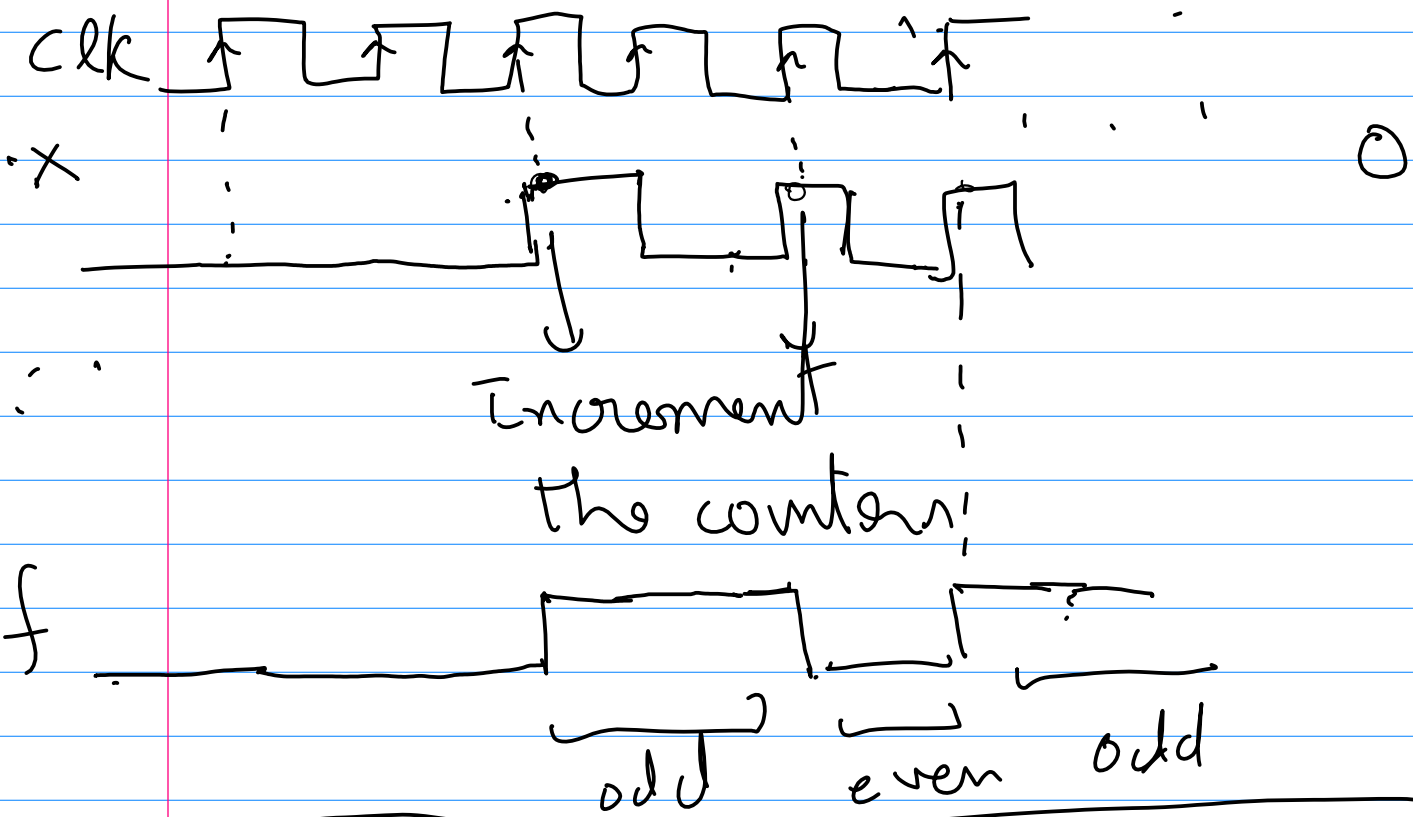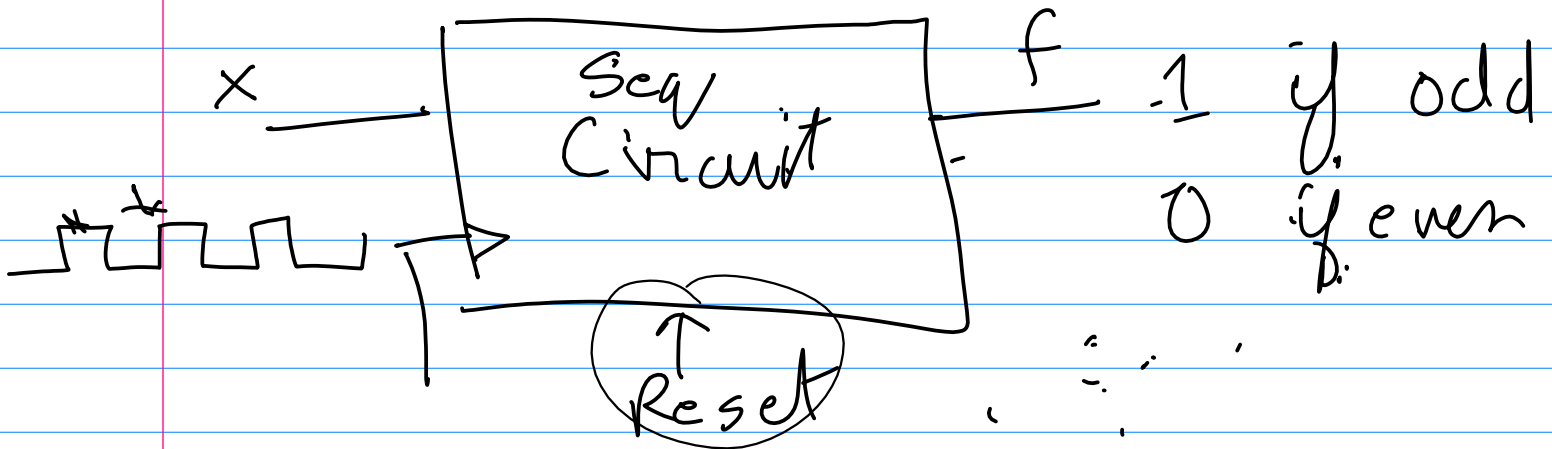
① Does it even need state/memory

② How many bits of memory? = 1

Reset

$S_0 = 0$ inputs
$x=0$ $S_1 = 1$ input
$S_2 = 2$ input

$S_0 = 0$

even

$S_0 / f=0$ $x=0$

$x=1$ $S_1 / f=1$ odd
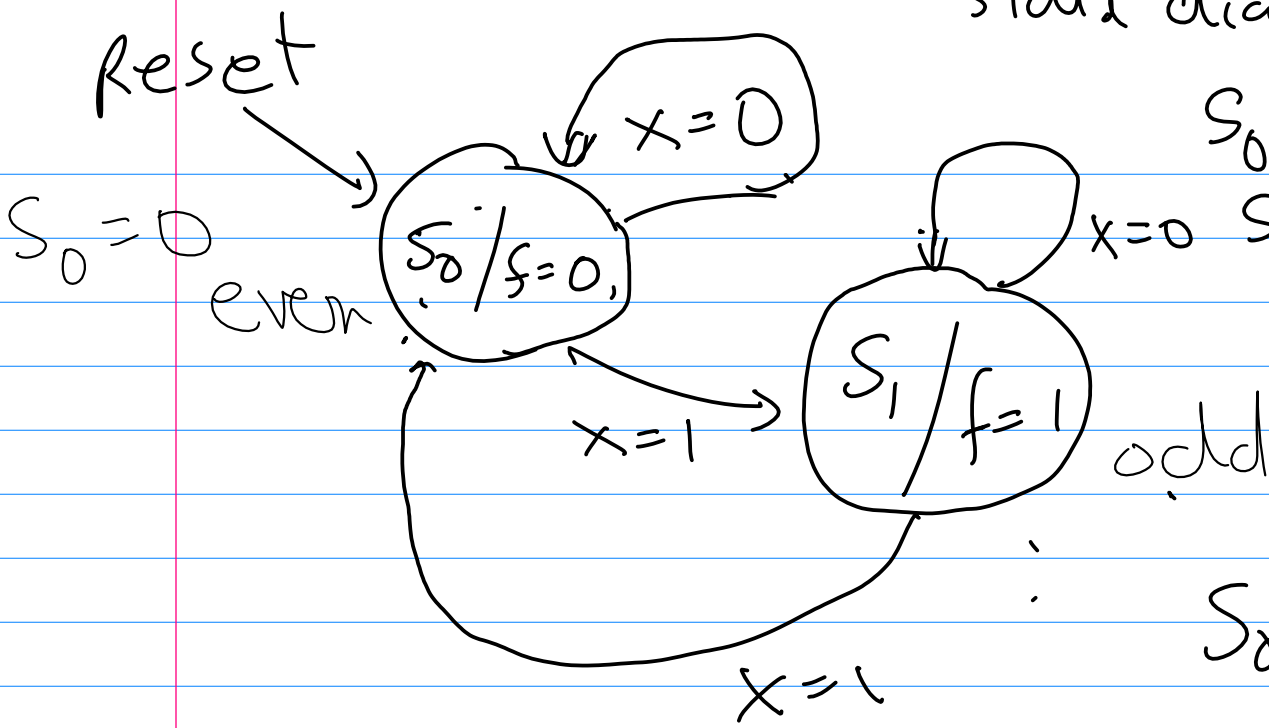
$x=1$

$S_0 =$ even
input

$S_1 = $ odd
inputs

#

next table;
state transition
→ state transition

$2 \to 1$ bit

$4 \to 2$ bits

$8 \to 3$ bits

$0 = S_0$
$1 = S_1$

$f = S^+$

Input to
D flip flop

X	S.	S$^+$	
0	0	0	
0	1	1	
1	0	1	
1	1	0	

D- flip flop

D	Q	Q̄
0	⭐	0
1	⭐	1

inversion

Excitatation table

Q	Q^+	D
	0	0
	1	1

X	S'	S^+	D
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

S.

X —— [L] —— D —— [D Q / ⊳clk Q̄] —— S=Q
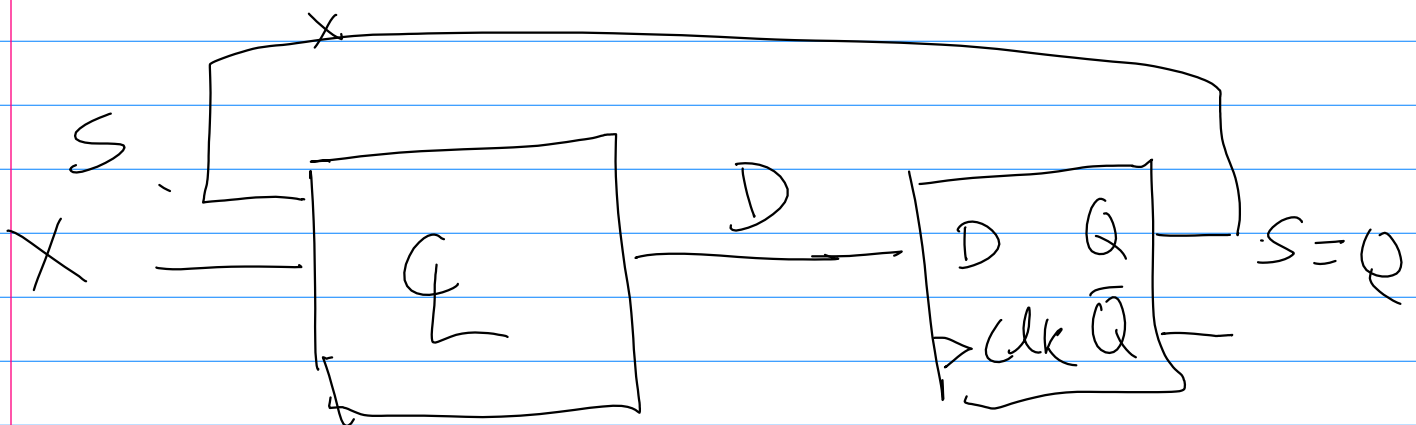
X	S	D
0	0	0
0	1	1
1	0	1
1	1	0

Named
gate

XOR gate

Reset

S

X

D   Q

CLK Q̄

Given

Clk

X

XOR { S

D } D FF
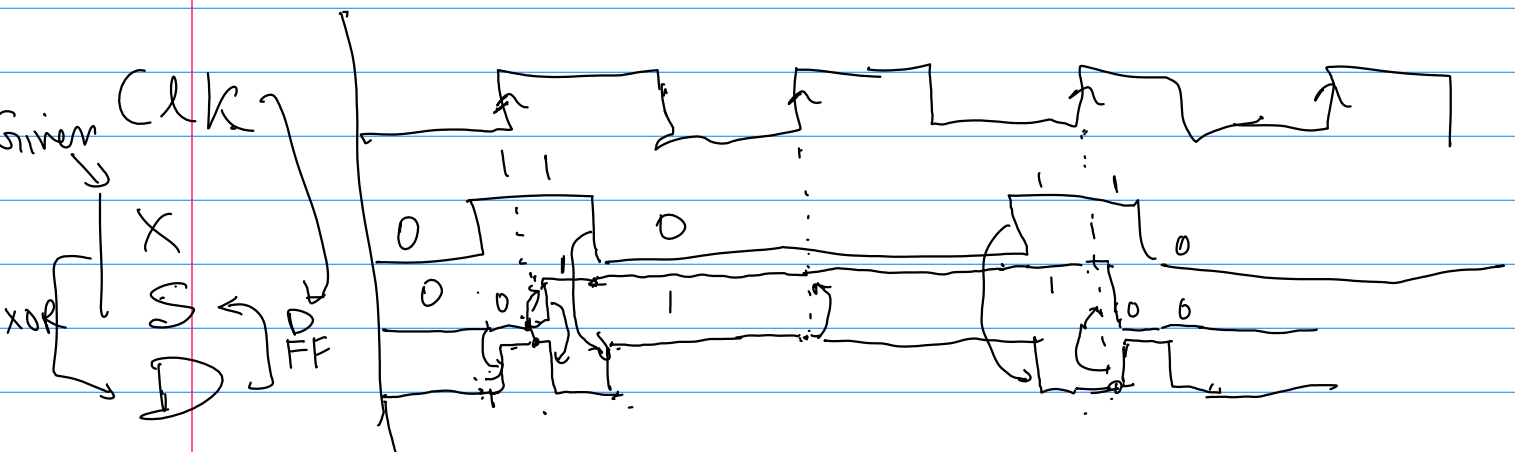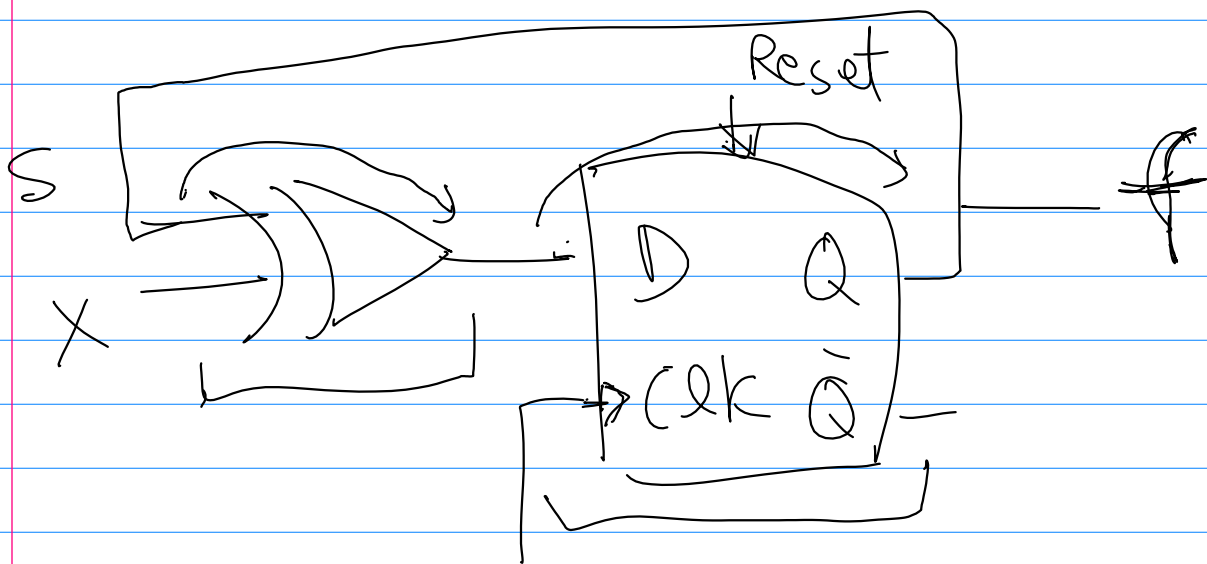
① next state table

J	K		Q	Q⁺

② Use excitation table of the
      ?   desired ff

③ Find the truth-table for inputs to the ff

① Next state table :-
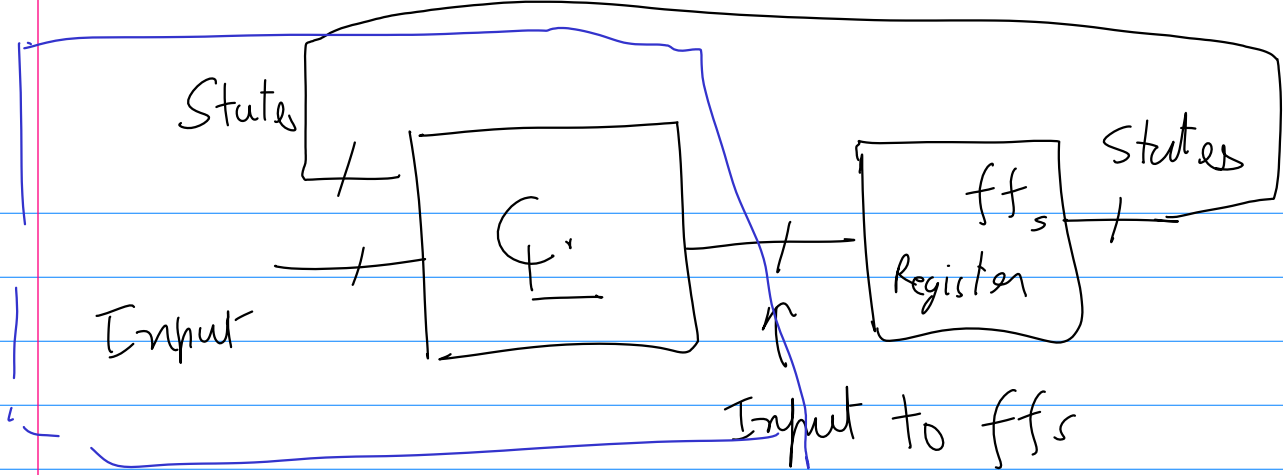   Set U Reset

J	K	Q	Q⁺	
0	0	0	0	] Hold
0	0	1	1	
0	1	0	0	] Reset
0	1	1	0	
1	0	0	1	] Set
1	0	1	1	
1	1	0	1	] Toggle
1	1	1	0	

toggle {

② D-ff

Q	Q⁺	D
d	0	0
d	1	1

③ Truth table

J	K	Q	D
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

States

Input

C_r

ff_s

Register

States

Input to ffs

D - ff
T - ff
Jk - ff

Problem 2. *A JK flip-flop receives a clock and two inputs, J and K. On the rising edge of the clock, it updates the output, Q. If J and K are both 0, Q retains its old value. If only J is 1, Q becomes 1. If only K is 1, Q becomes 0. If both J and K are 1, Q becomes the opposite of its present state.*

1. *Construct a JK flip-flop using a D flip-flop and some combinational logic.*
2. *Construct a D flip-flop using a JK flip-flop and some combinational logic.*
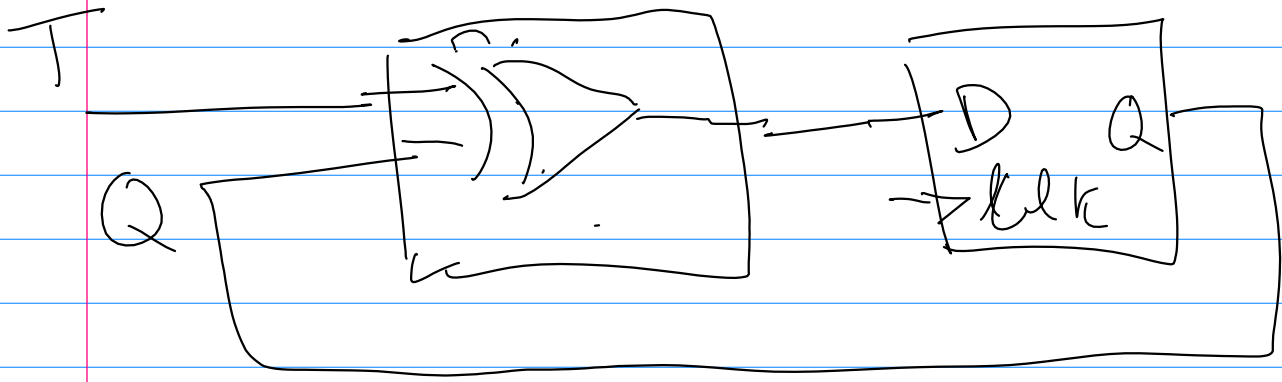3. *Construct a T flip-flop (see Exercise 3.9) using a JK flip-flop.*

T-ff

Toggle -ff

state table

T	Q	Q^+
0	0	0
0	0	1
1	0	1
1	1	0
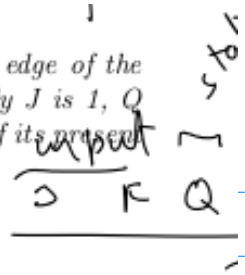
hold

toggle

Design a T-ff using D-ff!

Truth table for ⊕

Excitation table for D-ff

T	Q	D
0	0	0
0	1	1
1	0	1
1	1	0

Q	Q^+	D
d	0	0
d	1	1

$$D = T \oplus Q$$

Design a D-ff using T-ff

State table for D-ff

①

D	Q	Q$^+$
0	0	0
0	1	0
1	0	1
1	1	1

Using T-ff

Excitation table for T-ff

T = 0

Toggle

Q	Q$^+$
0 → 0	
1 → 1	

T = 1

Q + Q$^+$
0 → 1
1 → 0

② All possible transition

Q → Q$^+$	T
0 0	0
0 1	1
1 0	1
1 1	0

③

T	Q	Q$^+$
0	0	0
0	1	1
1	0	1
1	1	0

D

Q

T   Q

>Clk $\bar{Q}$
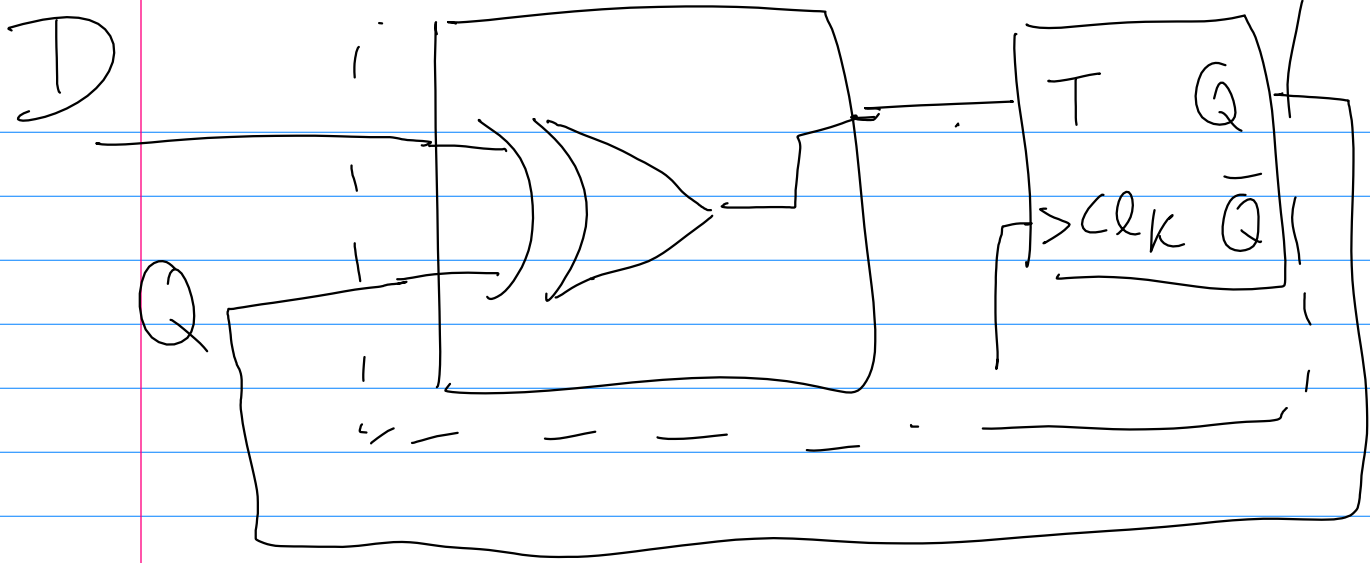
Truth table for the CL

D	Q	T
0	0	0
0	1	1
1	0	1
1	1	0

$$T = D \oplus Q$$

Design J-k ff using T-ff

	Set	Reset		
	J	K	Q	Q^+
hold	0	0	0	0
	0	0	1	1
Reset	0	1	0	0
	0	1	1	0
Set	1	0	0	1
	1	0	1	1
Toggl	1	1	0	1
	1	1	1	0

Excitation Table for T-ff

Q	→	Q^+	T
0		0	0
0		1	1
1		0	1
1		1	0
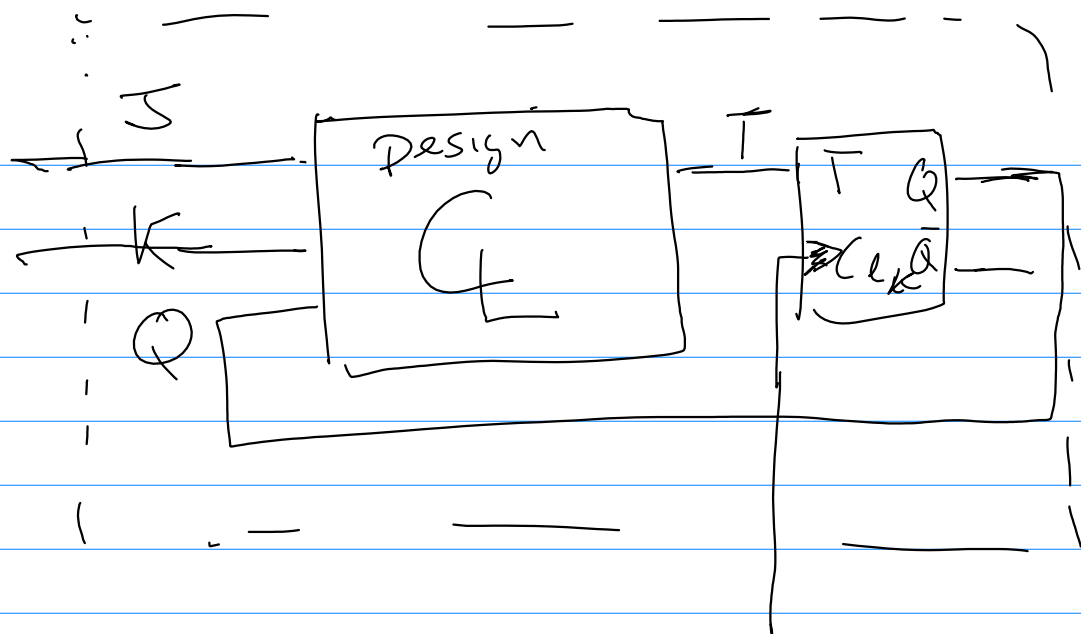
Truth table for CL

J	K	Q	T
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$$T = f(J, K, Q) = J\bar{Q} + KQ$$

D using J-K   Practice

T using J-K

J-K using D

J-K using T

D using T
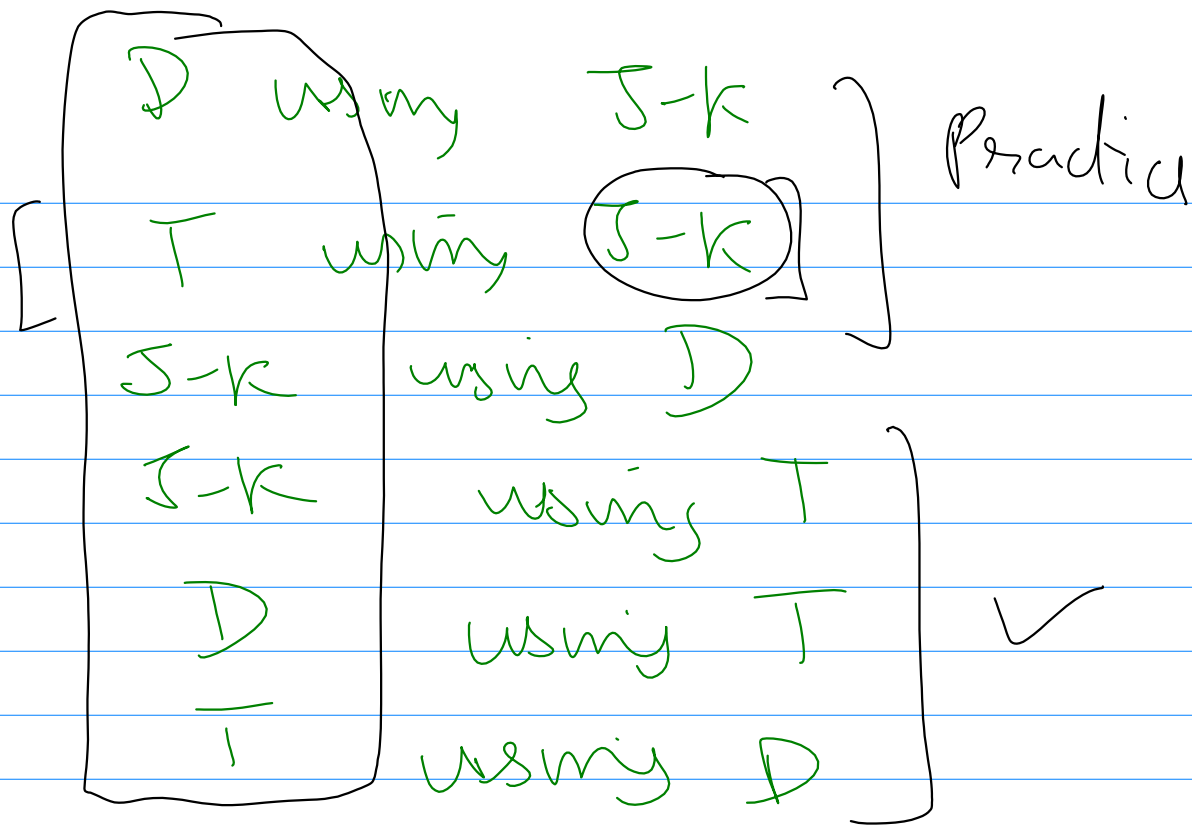
T using D   ✓

✗ flip flop using X-ff

✓ any sequential using X-ff
circuit

Design a T-ff using J-K ff

State table for the desired sequential circuit
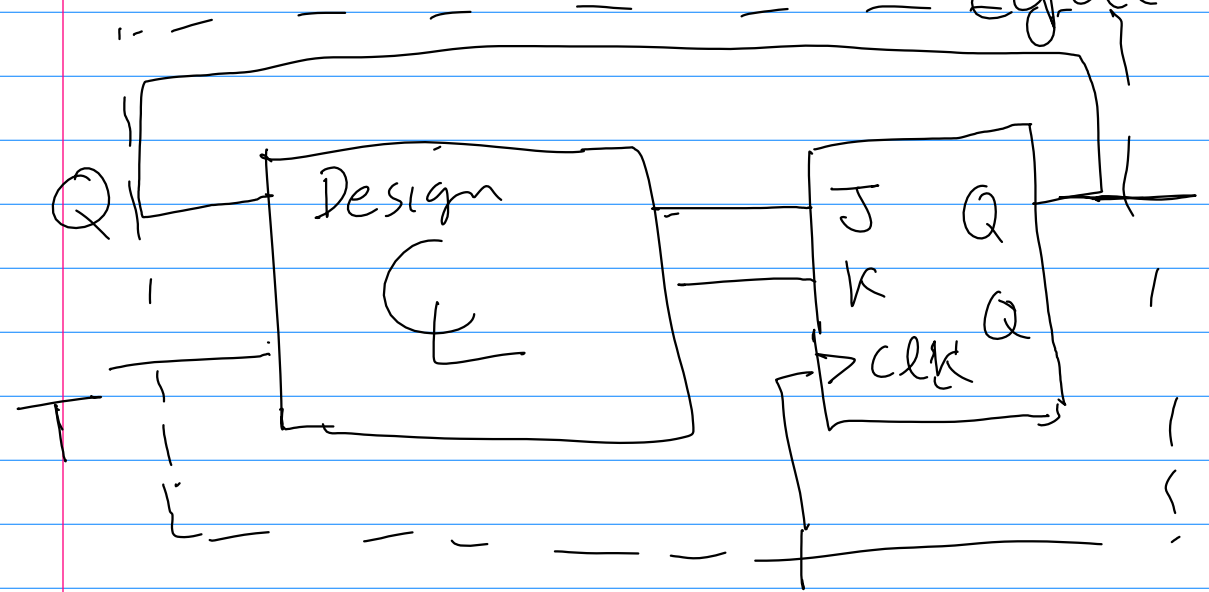
State transition table ←

T	Q (start)	Q^+ (next)
0	0	0
0	1	1
1	0	1
1	1	0

Desired

row 1

Q^+ state at the next clock cycle

Q

Design CL

T

J   Q
K
▷clk Q̄

[Excitation table for J-K ff]
State trans table

	J	K		Q^+
hold	0	0		Q
Reset	0	1	✓	0
Set	1	0		1
toggle	1	1	✓	$\overline{Q}$

→

	$Q \to Q^+$		J	K
hold reset	$0 \to 0$		0	d
set toggle	$0 \to 1$		1	d
reset toggle	$1 \to 0$		d	1
hold set	$1 \to 1$		d	0

Truth table for the ff

→

	T	Q		J	K
row 1	0	0		0	d
row 2	0	1		d	0
row 3	1	0		d	d
row 4	1	1		d	1

$$J = T \qquad ?$$
$$K = T \qquad ?$$

Problem 3. *A divide-by-N counter has one output and no inputs. The output Y is HIGH for one clock cycle out of every N. In other words, the output divides the frequency of the clock by N. The waveform for a divide-by-3 counter is shown here:*

CLK

Y

(a)
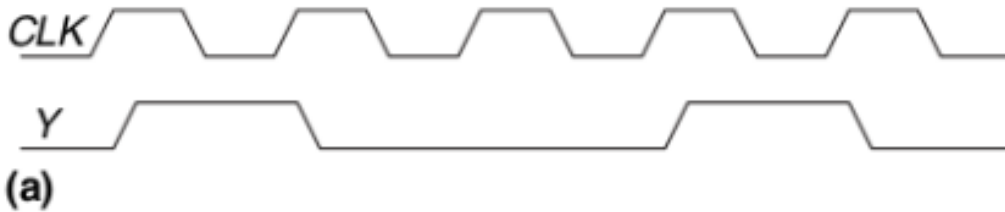
Sketch circuit designs for such a counter

10 ns ← For example

2 clock cycle

Divide by 2
counter

Divide by 3
counter

Divide by 10^6
counter

Pong game

FPGA

$$\frac{10^9}{10} = 10^8 = 10^2 \times 10^6$$

= 100 MHz

frequency

50 MHz
↓
20 ns

50 MHz
↳ 60 Hz
↳ 10 Hz
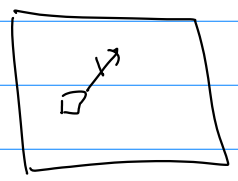
Divide by 3 counter

* New   outputs $\neq$ state

So far   outputs $=$ state

① states → outputs
Moore configuration

② inputs
+
states → outputs
Mealy conf.

'inputs   inputs

Clk

output

Design CL ②

input
to reg

Design CL ③

states

states

n-ff
n-bit
register
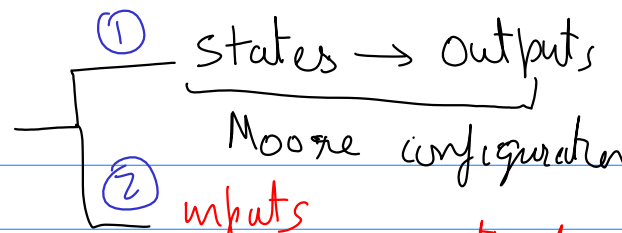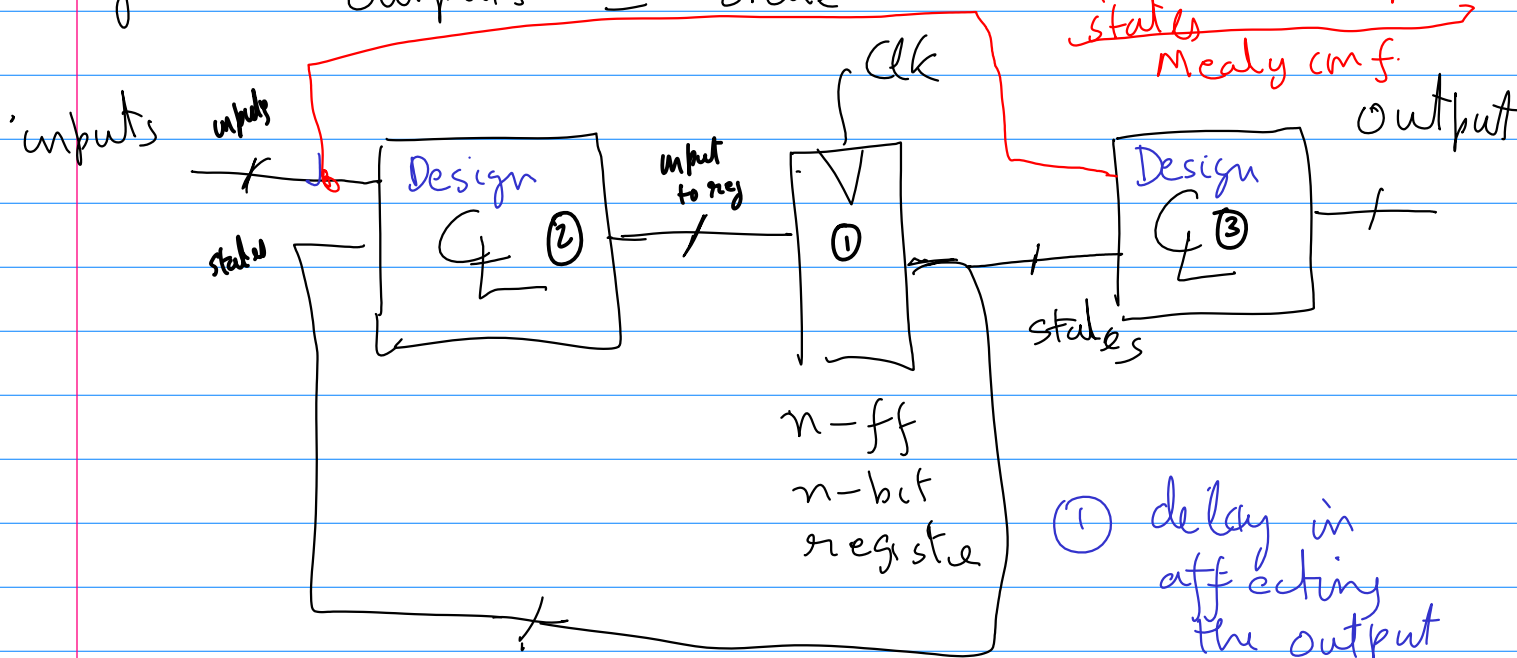
① delay in
affecting
the output

② input need to
by synchronous

2^n = states

Reset

State transition diagram

no inputs

3-states

$S_0/F=1$ → $S_1/F=0$ → $S_2/F=0$ →(back to S_0)

Clk

F =

S_0 S_1 S_2 S_0
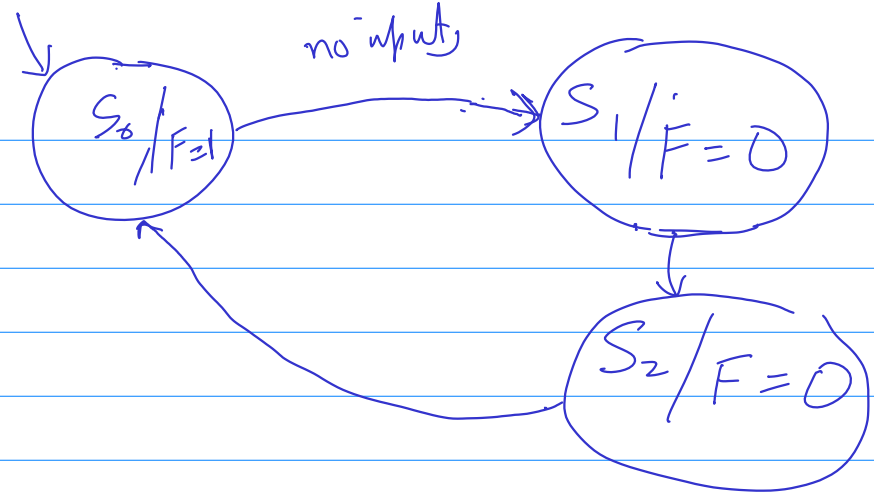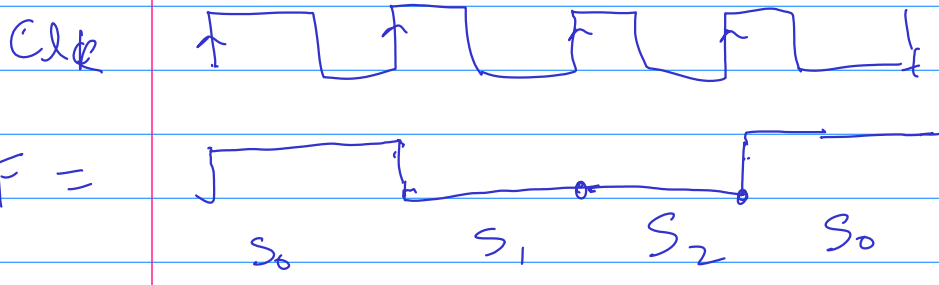
Divide by 5 counter. How many states?   5-states

states of possible $= 2^{(\# \text{ of } ff)}$

3-states   with   minimum # of ff

How many ff do we need?

We nec   2 ff $3 \leq 2^{\leftarrow \# \text{ of } ff}$

If we needed 5 states, How many ff?

$c_1 c_0$
$S_0 = 00$
$S_1 = 01$
$S_2 = 10$

$2^2 < 5 \leq 2^{3 \leftarrow \# \text{ of } ff}$

$\longrightarrow$ State assignment
(assigning binary number to each unique state)

Desired state transition table

State	next state
S_0	S_1
S_1	S_2
S_2	S_0

state $C_1 C_0$	next state $C_1^+ C_0^+$
0 0	0 1
0 1	1 0
1 0	0 0

	C_1 C_0	f ?
S_0	0 0	1
S_1	0 1	0
S_2	1 0	0

Homework
for next class