Sequential Circuit Analysis

next state → A^+ or $A(t) = A(t+1) = D_A$

next cycle

Example on Analysis Sticle (44)

Input : x, y — moore sync

Z : output

state : A next : A^+

equations :

$Z = A$ $A^+ = x \oplus y \oplus A$

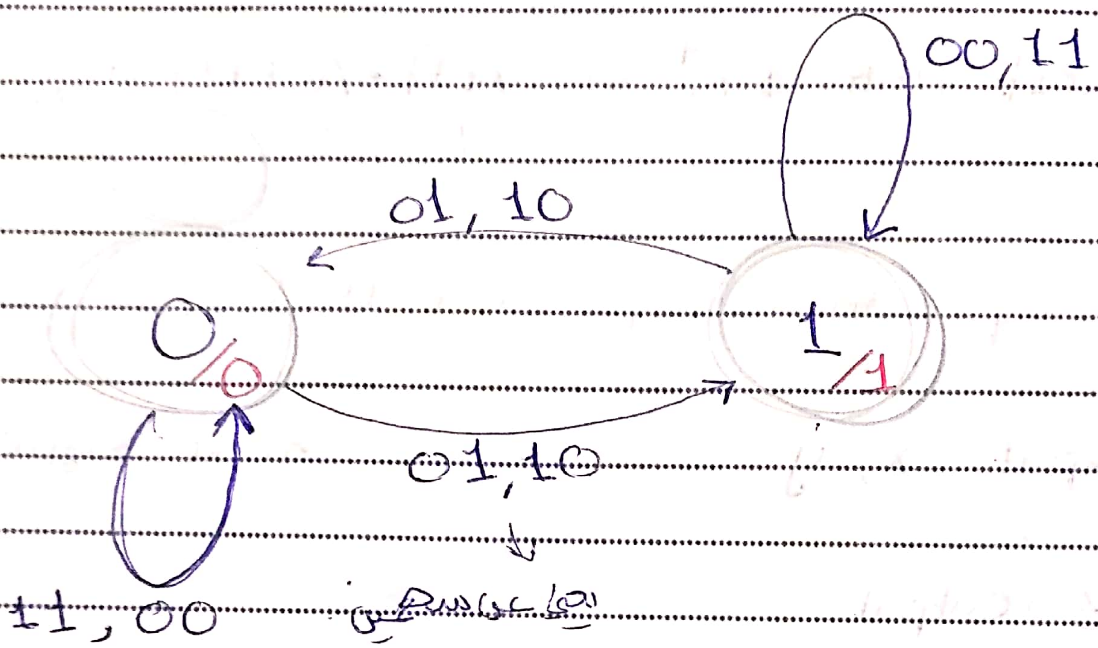
State	Inputs		Next state	output
A	x	y	A^+	Z
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

STARS NOTE BOOK

SUBJECT: .... / /

$\# \text{ of states} = 2^{\# \text{ flip flops}}$

State diagram $\qquad \# \text{ of transition} = 2^{\# \text{ Inputs}}$



← لما يكون moore ← بنحط ال Z جوا الدائرة لأنها ما بتتغير (هو نفسه ال A)

← لما يكون mealy ← بنحط ال Z ع السهم لأنه عكس تتغير (Z تتغير عن A)

Exercise slide (45)

(mealy)

input: x
output: y
states: A, B

equations:

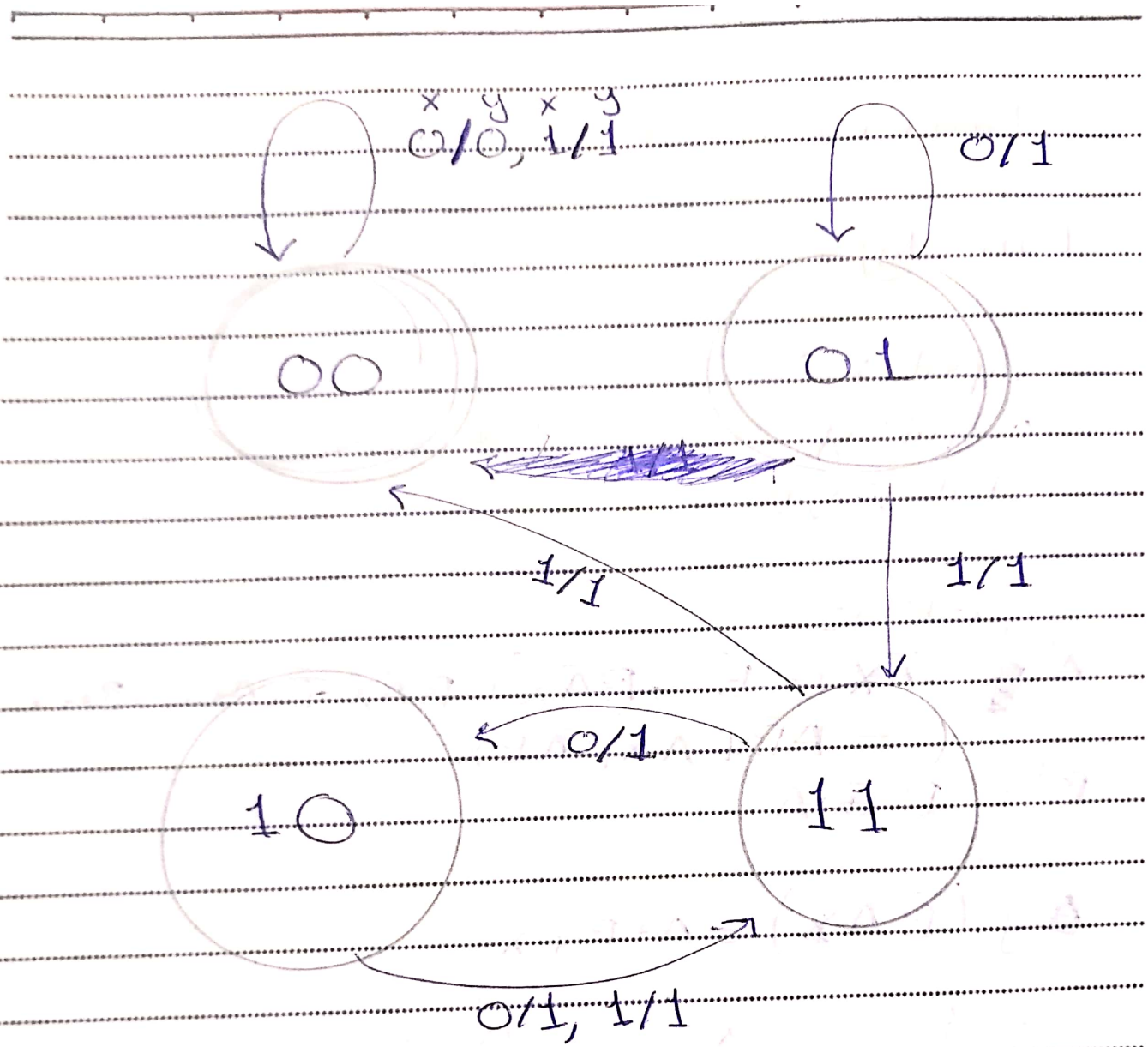
$A^+ = \bar{A}X \oplus \bar{B} = \bar{B}\bar{A}X + B\bar{A}X = \bar{B}(A+\bar{X}) + B\bar{A}X$

$= A\bar{B} + A\bar{X} + \bar{A}BX$

$B^+ = B \oplus A$

$y = (\bar{B}\bar{A}\bar{X}) = A + B + X$

states A	states B	inputs X	next states A^+	next states B^+	output y
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	0	0	1



Alternate State Table

المقصة إنه هاي بالعربي نفس ال state table بالزبط بس ^^

Exercise slide (51)

when no input is available:
transition without condition

Reset ← Direct input ملاحظة
(active low)

وبيعتمد بالأصل ع ال clock ، بيكبس تشتغل ذلك
وراجعون لحد ما يرجع ال user يكبس ال reset

111 → 110 → 101
unused states
مارح نوصلهم

Reset
→ high voltage → مش مكبوسة
→ low voltage → مكبوسة

Sequential Circuit diagram.

Word description → State diagram

→ State table (State encoding)

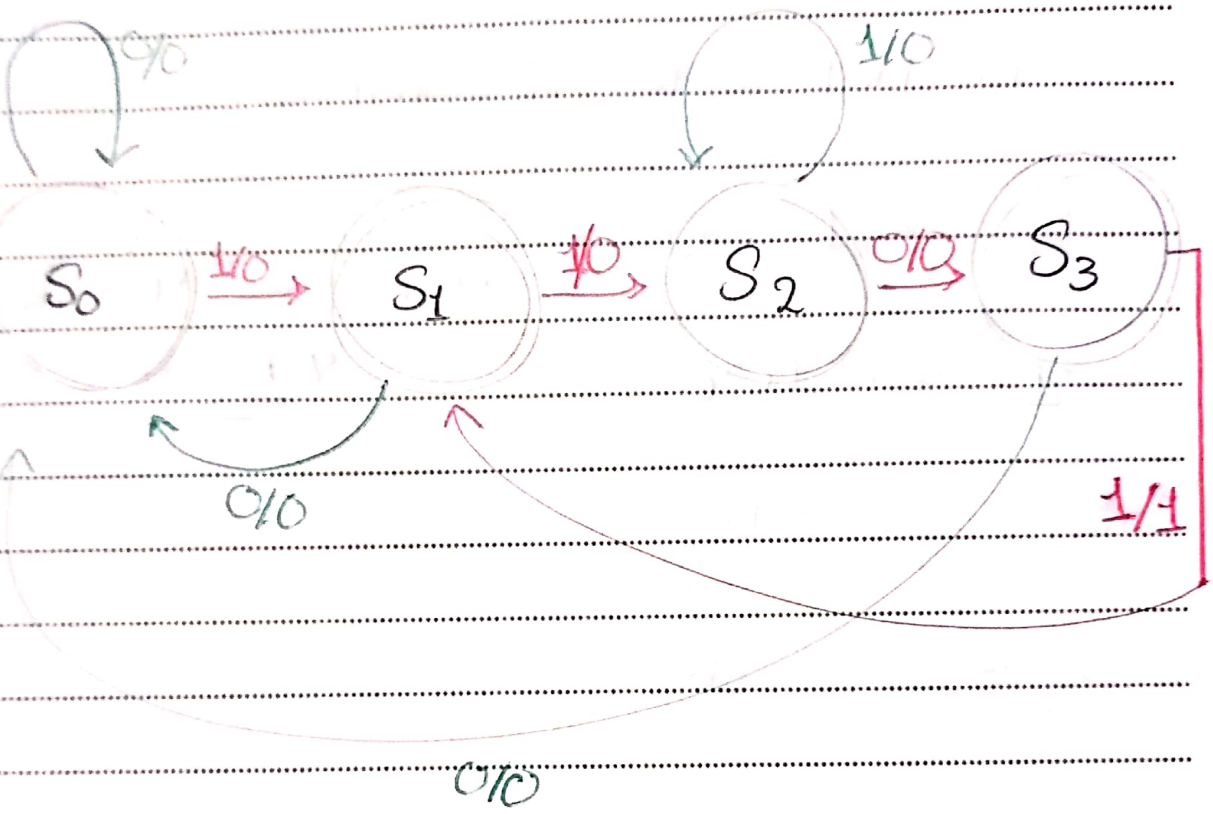
→ Selecting flip-flop type

→ Input equ. to F.F, output equ.

مهم اكون عارف إنه ال design محتاج ال memorization

Example slide 58:
Sequence detector example (1101)

Follow the pattern 1101 — states of 2 bits (2 D flip-flops)



→ then assign the counting order or the grey code or one-hot-state to (states)

→ create the k-maps and derive simplified equations

→ draw the diagram

بستخدم بعدين طبعاً الـ k-map - مشان احجم الـ circuit

وبضيف الـ Reset

One-Hot state Assignment-

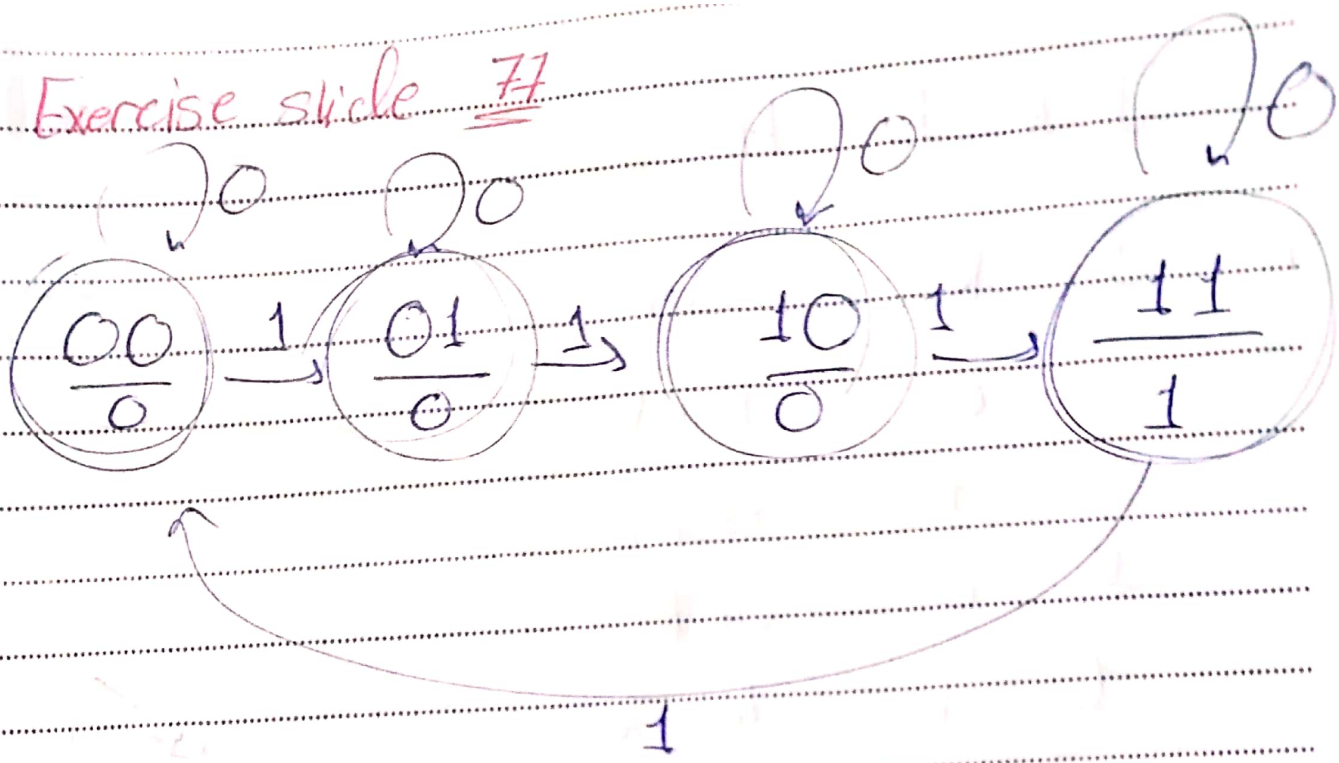
$S_0 \rightarrow 1000$ → 4 bits → 4 states

$S_1 \rightarrow 0100$ 4 Flip flops

$S_2 \rightarrow 0010$

$S_3 \rightarrow 0001$

Exercise slide 71

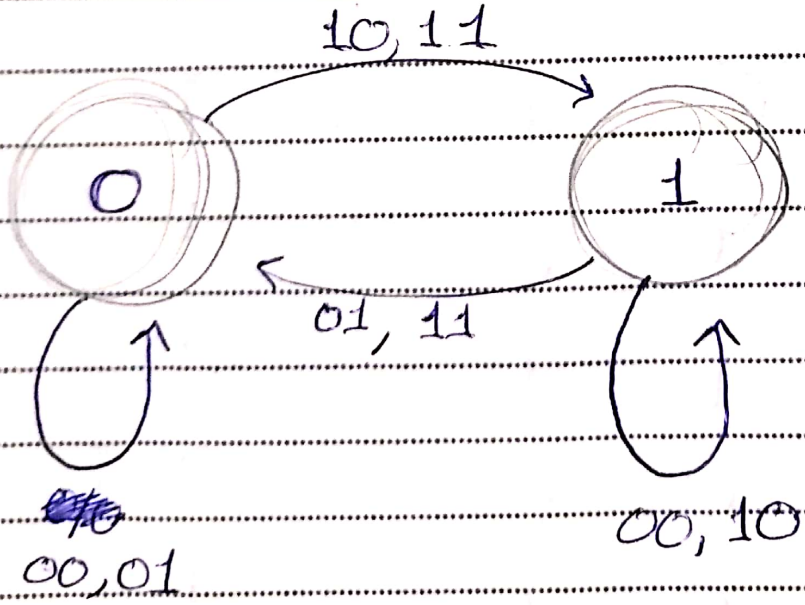


A	B	X	A^+	B^+	Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1

J-K Flip-Flops

J	K	out
0	0	hold
0	1	Reset
1	0	Set
1	1	toggle → العكس

Inner implementation



Q	J	K	Q^+	
0	0	0	0	Hold
0	0	1	0	
0	1	0	1	
0	1	1	1	toggle
1	0	0	1	hold
1	0	1	0	
1	1	0	1	
1	1	1	0	toggle

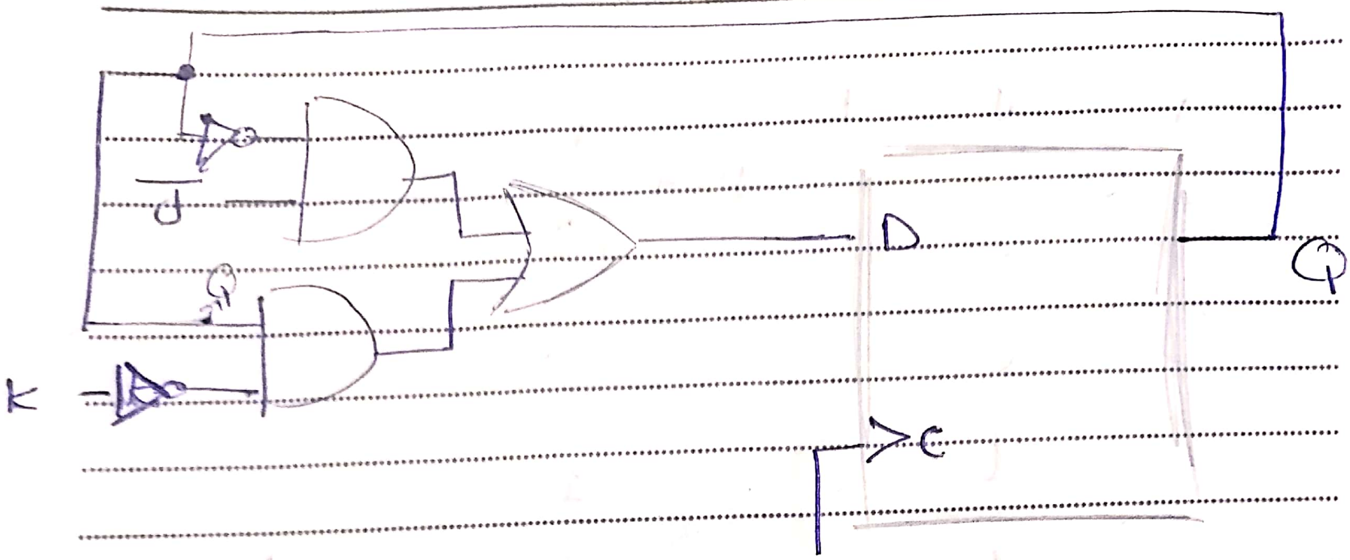
اعتبره كأنه ديزاين خارجي ، اعتبر ال state

هي ال Q و الـ new state هي الـ Q^+

لأنه وجود أي تغيير في ال hold combination

(1 1) ← toggle combination او الـ (0 0)

$Q^+ = \bar{Q}J + \bar{K}Q$ ← حسب ال k-map



sequential circuits design ← تصميم بطريقة ال

→ state diagram

→ state table → J K inputs ← اعتبر ال Q old state

↓

بيعطيك ال Q^+ الي بيطلع معك من هذا الجدول

→ design.

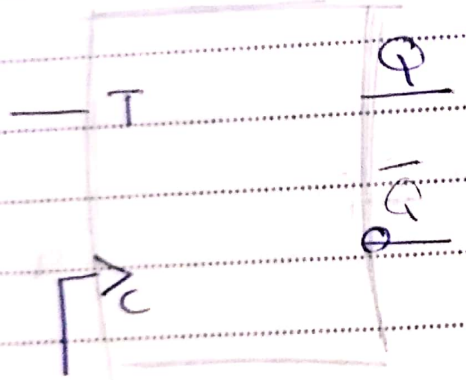
T Flip-flop

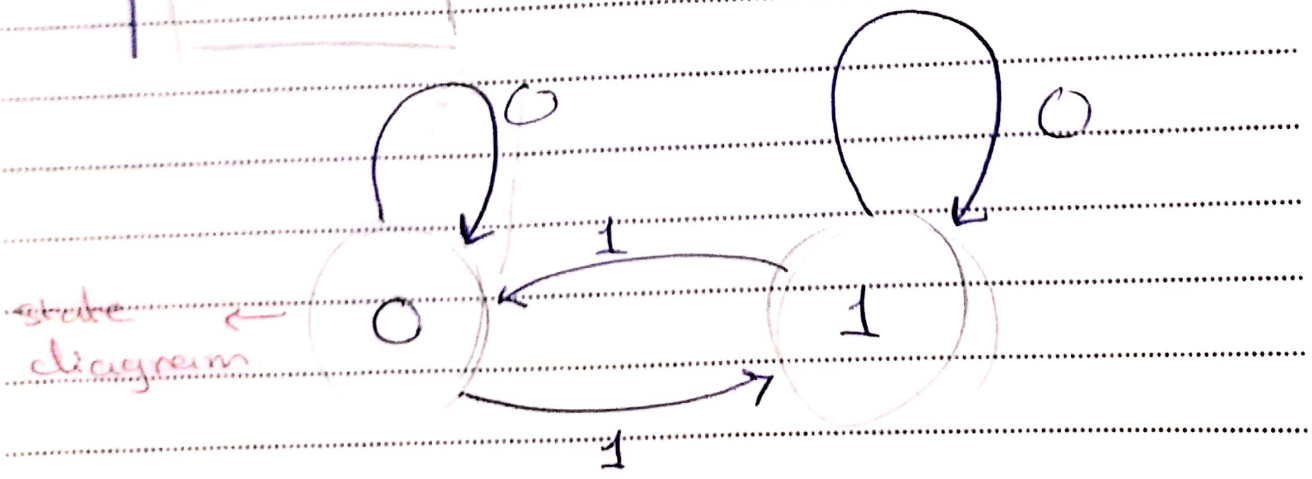
T	
0	hold
1	toggle

يشبه ال D من ناحية flip-flop.

→ 1 input اله

بس مختلف التنفيذ تماماً.



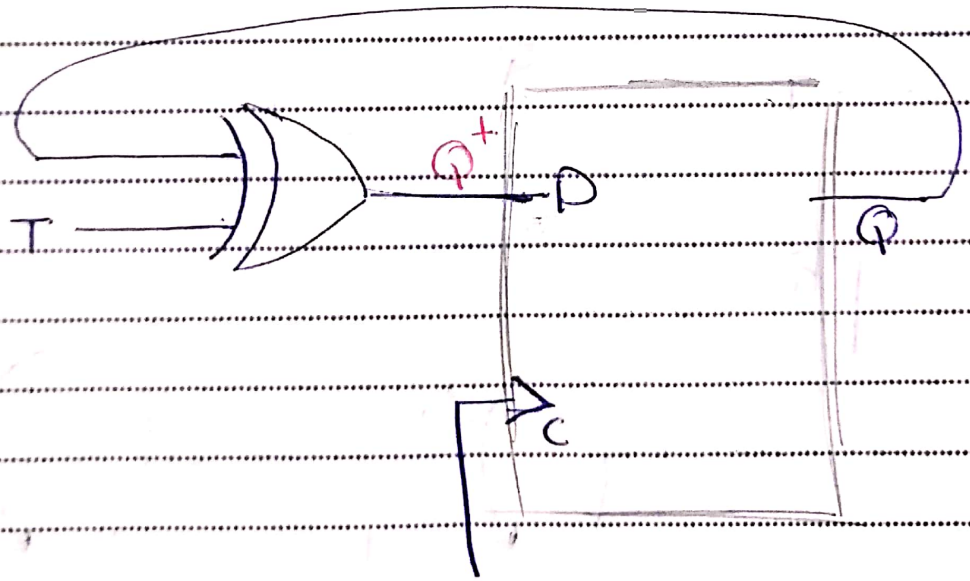


state diagram ←

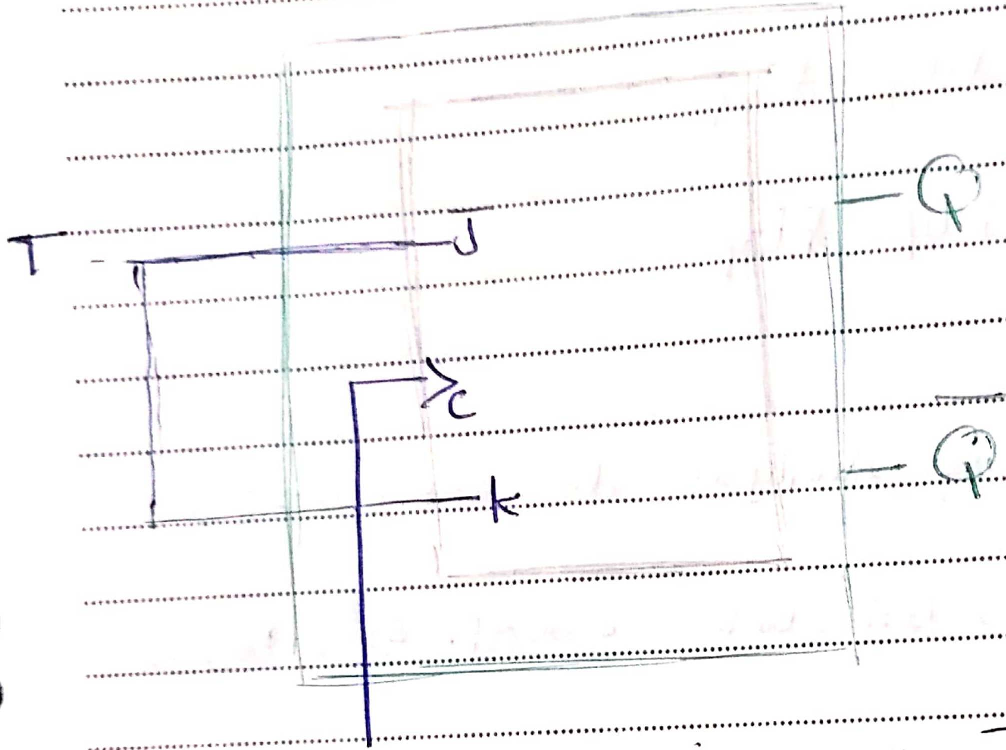
Q	T	Q^+
0	0	0
0	1	1
1	0	1
1	1	0

XOR

$Q^+ = Q \oplus T$



أو ممكن ؟! أستخدم ال JK Flip-Flop



ليش زابطة، لأنه هيك مستحيل ال J و K يكونوا مختلفين

T

	J	K	
T=0 ←	0	0	hold
	0	1	Reset
	1	0	set
T=1 ←	1	1	toggle

لكن الطريقة الأولى أرخص

SR → Flip-Flop

D → Flip-Flop

JK → Flip-Flop

T → Flip-Flop

راح نحتاج عشان نحل Analysis لـ سيركت T أو JK خطوة [illegible] ، اضافية ، ما كنا نحلها بال D Flip-Flop لأنه كنا نعتمد على قيمة ال old state بدون أي اضافات.

لكن مثلاً مع ال T ← بحتاج اعرف شو هي ال T مع ال old state شو رح تكون.

Analysis procedure for any sequential circuit ✓

① derive equations of
✓ new states + define inputs and old states.
✓ output

from equations find state table ✓

② derive state diagram from the table. ✓

← تذكر انه الدوائر بتمثل ال states بالحالية لكم حسب combinations

عدد ال states ← اذا 3 ← 8 دوائر مثلاً

← تحديد كمان اذا هو moore او mealy.

← تذكر اذا في input بتحلي conditional او لا transition

Slide 86 Exercise

① A^+, B^+, C^+ — But there is first T_A, T_B, T_C — they control which values A^+, B^+, C^+ will store.

and $y = A \oplus B \oplus C$

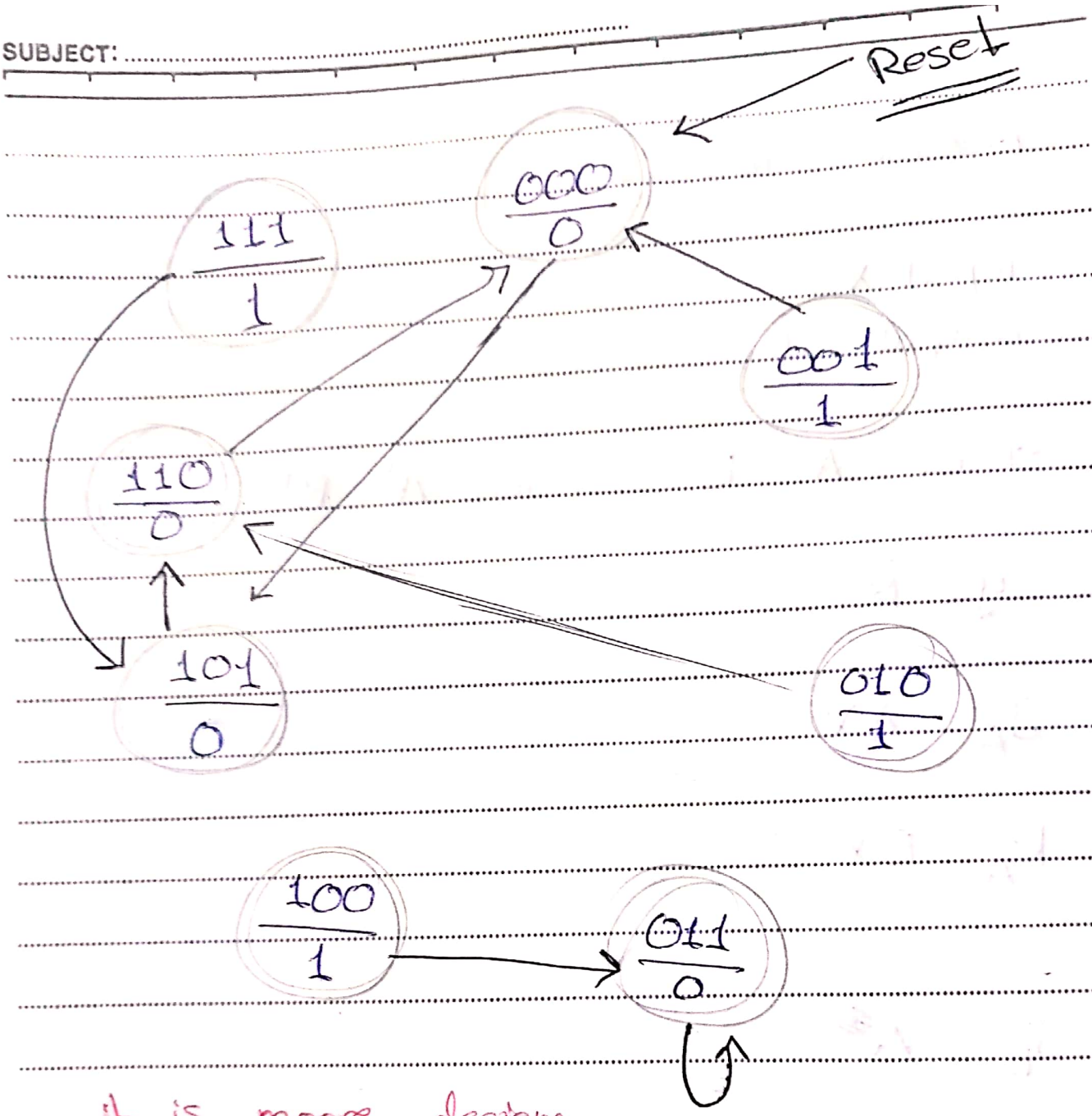
$T_A = \bar{C}$

$T_B = A$ ← لازم نعرف قيمهم اولا

$T_C = \bar{B}$ — حتى نحط ال new states.

A	B	C	T_A	T_B	T_C	A^+	B^+	C^+	y
0	0	0	1	0	1	1	0	1	0
0	0	1	0	0	1	0	0	0	1
0	1	0	1	0	0	1	1	0	1
0	1	1	0	0	0	0	1	1	0
1	0	0	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	0	0
1	1	0	1	1	0	0	0	0	0
1	1	1	0	1	0	1	0	1	1

SUBJECT: .....



it is moore design

SUBJECT:

Exercise: Slide 87

input x

output y

states A, B → new A^+ B^+

$y = \bar{B}$

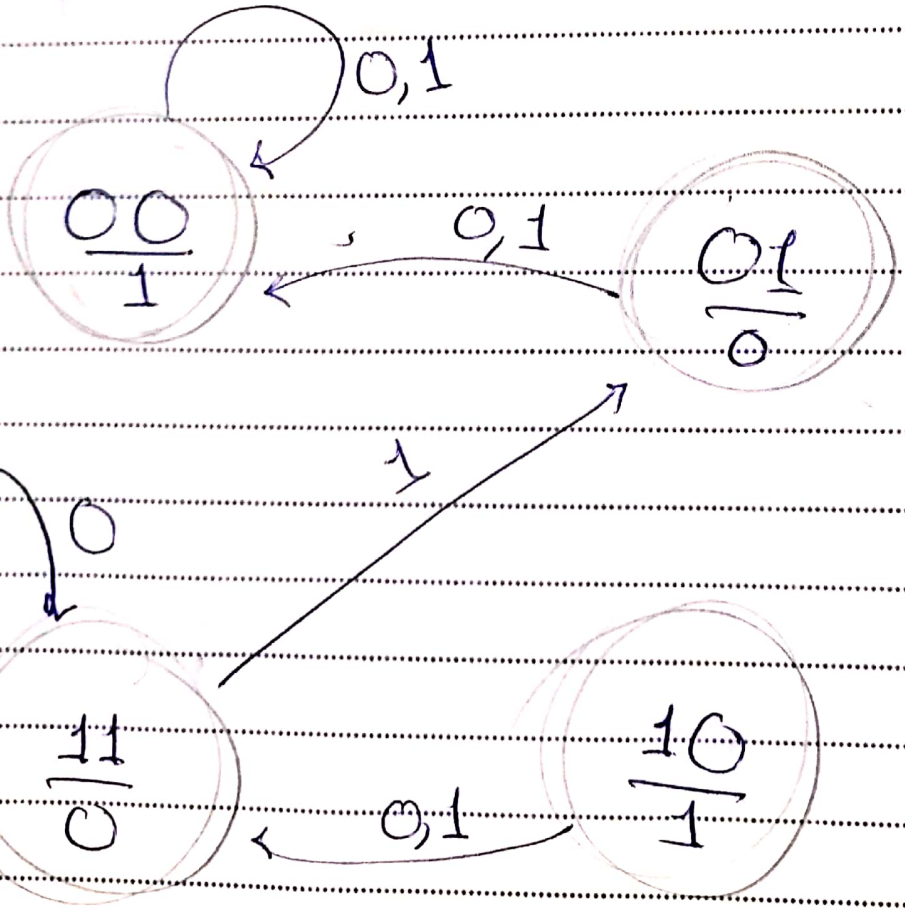
$J_A = AX$

$K_A = BX$

$J_B = A$

$K_B = \bar{A}$

A	B	X	J_A	K_A	J_B	K_B	A^+	B^+	Y
0	0	0	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	0	1
0	1	0	0	0	0	1	0	0	0
0	1	1	0	1	0	1	0	0	0
1	0	0	0	0	1	0	1	1	1
1	0	1	1	0	1	0	1	1	1
1	1	0	0	0	1	0	1	1	0
1	1	1	1	1	1	0	0	1	0



moore ✓✓

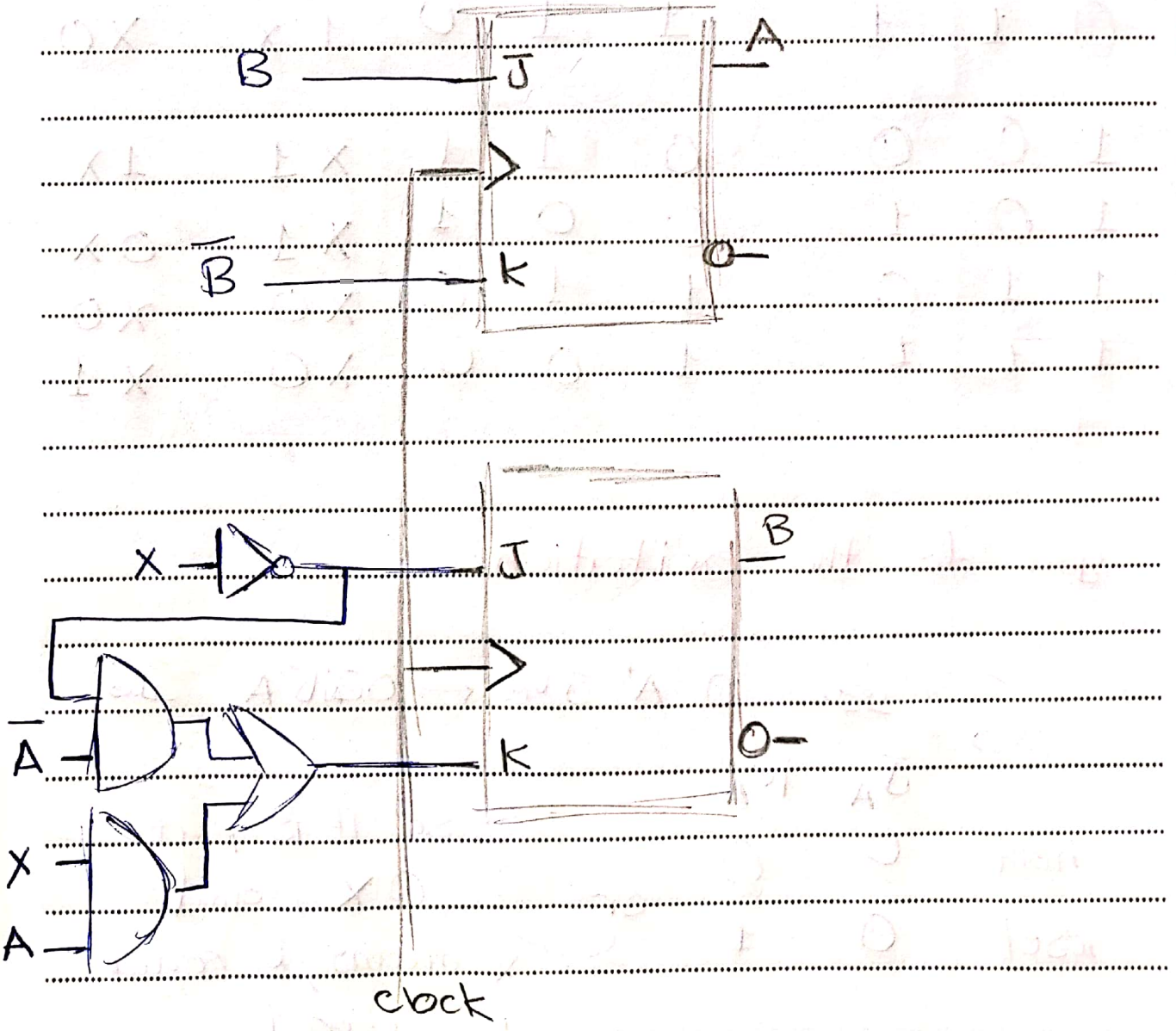
SUBJECT: .....

Exercise slide 89.

① Using D Flip-Flops

doing the k-maps.
as the previous example.

② J-k Flip-Flop ← عدد الـ Flip-Flops ما بتغير.



لازم اجيب 5 معادلات (5 k-maps)

J_B J_A لـ

y K_B K_A

A	B	x	A^+	B^+	y	J_A	K_A	J_B	K_B
0	0	0	0	1	1	0	X	1	X
0	0	1	0	0	1	0	X	0	X
0	1	0	1	0	0	1	X	X	1
0	1	1	1	1	0	1	X	X	0
1	0	0	0	1	1	X	1	1	X
1	0	1	0	0	1	X	1	0	X
1	1	0	1	1	0	X	0	X	0
1	1	1	1	0	0	X	0	X	1

you do the excitation:

فلانَ A كانتَ 0 ← حابب A^+ 0 يعني

J_A K_A

hold 0 0

reset 0 1

or

so it is put 0X, and X means, it could be 0 or 1

هناك اربع طرق لحل الجدول حتى تحرق شو لازم نعمل ؟!

Q	Q^+	J	k
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

excitation table

لي بساعدنا حسب ال Q وال k

اذا انا عارف Q و Q^+

→ After (y) k-map

$y = \overline{B}$

→ J_A k-map → بتعامل مع ال X كانه don't care

$J_A = B$

→ k_A k-map

$k_A = \overline{B}$

→ J_B k-map

$J_B = \bar{X}$

→ K_B k-map

$K_B = \bar{A}\bar{X} + AX$ → this is an XNOR

(3) Now implementing it with T Flip-Flop

A	B	X	A^+	B^+	y	T_A	T_B
0	0	0	0	1	1	0	1
0	0	1	0	0	1	0	0
0	1	0	1	0	0	1	1
0	1	1	1	1	0	1	0
1	0	0	0	1	1	1	1
1	0	1	0	0	1	1	0
1	1	0	1	1	0	0	0
1	1	1	1	0	0	0	1

Two Flip-Flops

→ y k-map

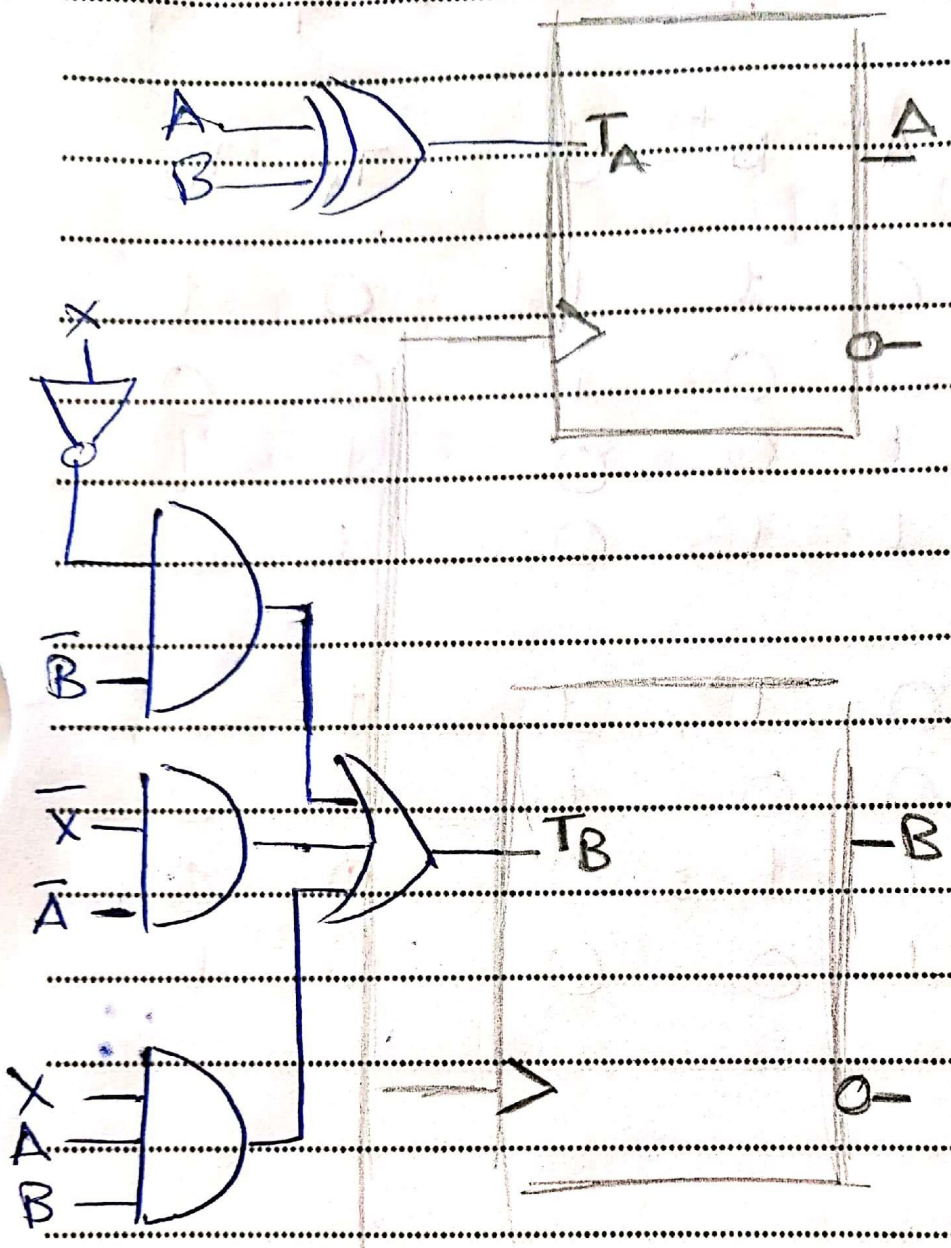
$y = \bar{B}$

→ T_B k-map

$T_B = \bar{X}\bar{B} + \bar{A}\bar{X} + ABX$

→ T_A k-map

$T_A = \bar{A}B + A\bar{B}$ → XOR between A and B



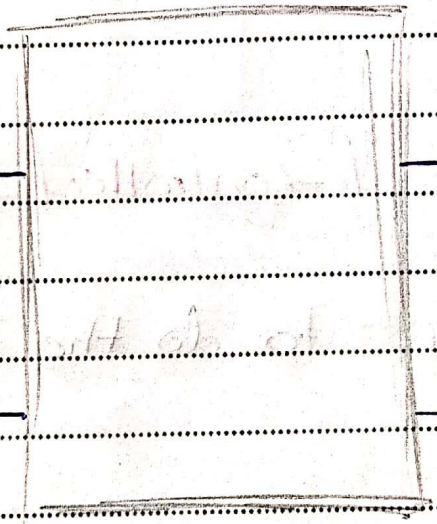
SUBJECT:

chapter 7

New chapter: Registers and Register Transfers

Registers → Storing data

(Array of flip-flops)



to write or load

to read the output

[clock Gating]

→ using an OR gate between inverted value of [local] enel the clock

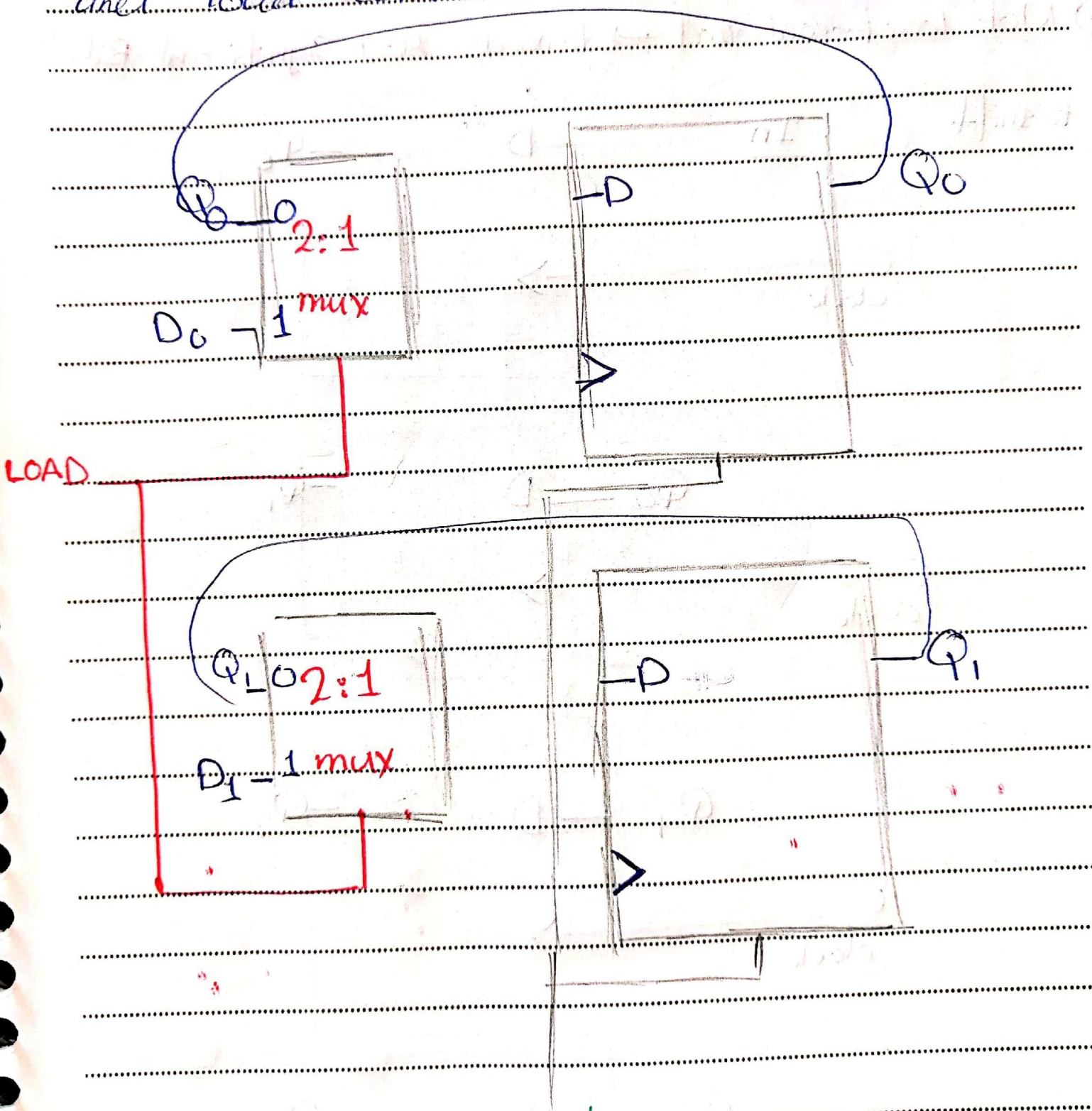
!! it causes a clock skew problem

[Registers with Local-Controlled Feedback]

Using multiplexers to do the selection

SUBJECT: .....

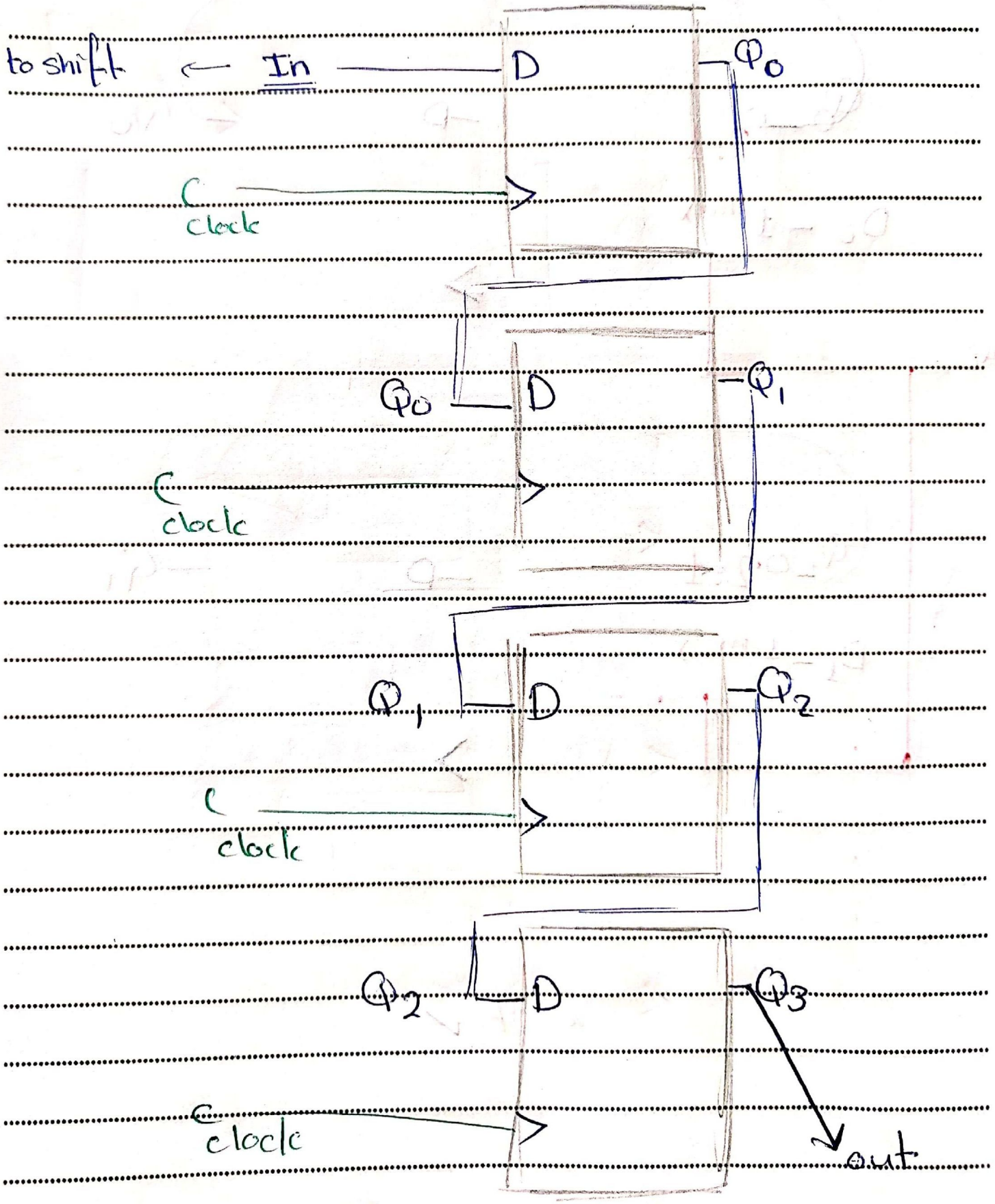
Register to read and write only ← اول نوع معلم؟
and load-controlled.



Shift Registers → it can be ... في نوع مهم ؟!

local-controlled or not

① Not local-controlled → towards Most Significant Bit



② Parallel load shift Register

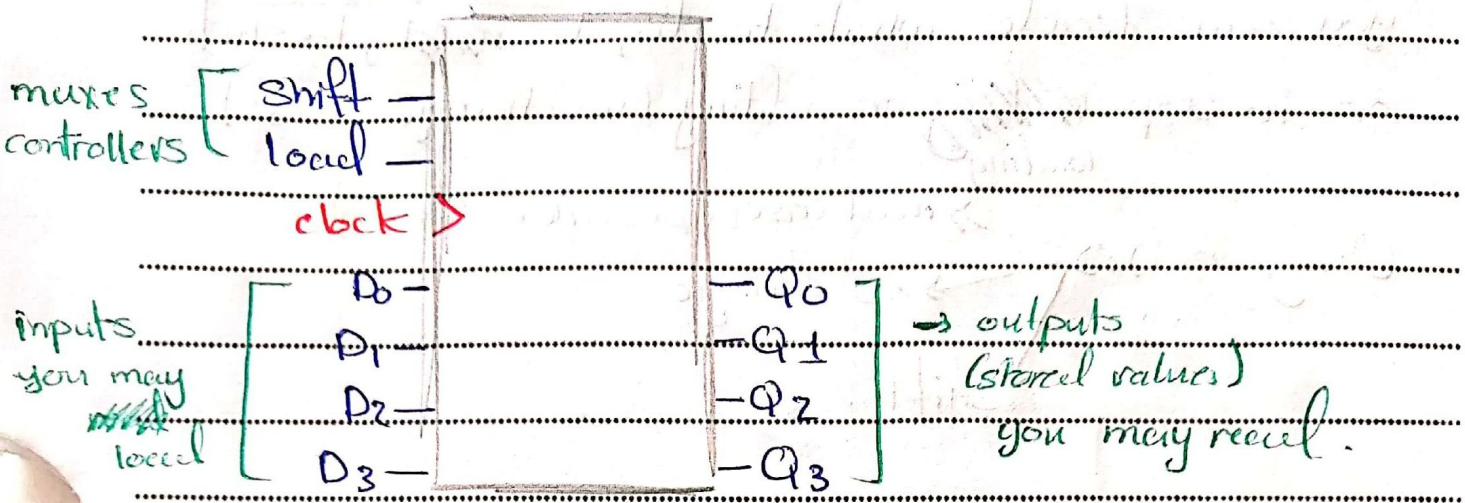
you can decide what to do, to ~~read~~ write, to shift or to stop ~~reading~~ loading or shifting (no change case)

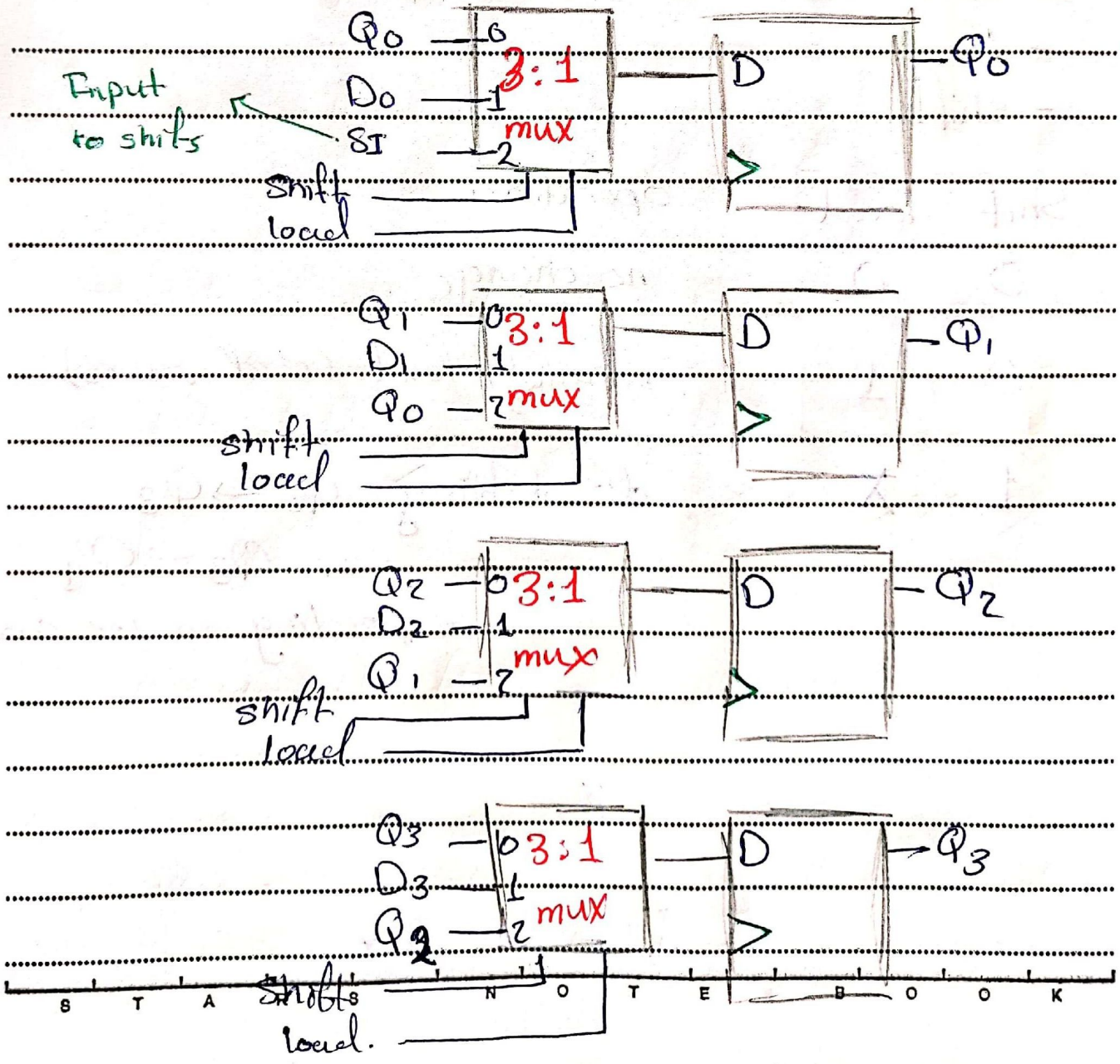
Using muxes → read case (parallel load)
→ no change
→ shift

2 controllers on selection line of the mux
- Load
- shift

Shift	load	operation
0	0	no change
0	1	parallel load (read values)
1	X	do shifting $Q_0 \rightarrow Q_3$ or $Q_3 \rightarrow Q_1$ depending on the case

High-level





③ Bidirectional shift Register

you can
→ write ✓
→ shift ✓ in both sides based on mux condition
→ you can stop all operations.

truth table:

Shift	load	
0	0	no change.
0	1	shift down (Q0 → Q3) مثلاً
1	0	Shift up (Q3 → Q0) مثلاً
1	1	

it takes left serial input to start shifting

it takes right serial input to start shifting

you'll use 4:1 mux

and the design is then clone as the previous example.

Counters

→ it can count up
→ " " " down
→ " " " through other fixed sequences

We study Synchronous Counters

you will design any counter the way you made designs in chapter 5

→ state diagram [if it starts with englis]

→ state table → 

→ k-maps → equations

put in flip-flops, with number defined by number of states.

you can do one more thing if mentioned [parallel load controlling]

SUBJECT:

→ 4 Bit counter with parallel load

(→ 4 states → 4 flip-flops

use 2:1 mux for each flip flop

load 0 → stop moving

load 1 → keep moving

→ Synchronous Counter with an Arbitrary Sequence

don't cares الي مش موجودين بخط بداخل combinations ال