EICoM
ELECTRICAL COMPUTER MECHATRONICS

مختـبر منطـق رقمي

للطالبـة المبدعـة

رزان قرمش

إرادة - ثقـة - تغيـير

* Digital logic lab

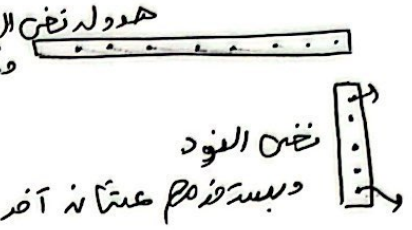
EXP 1+2 :-

* بالنسبة لل output

high = 1 ← احمر ←

low = 0 ← اخضر ←

Short cct ← طافي هو ←



* AND $x \cdot y = z$

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

* OR $x + y = z$

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

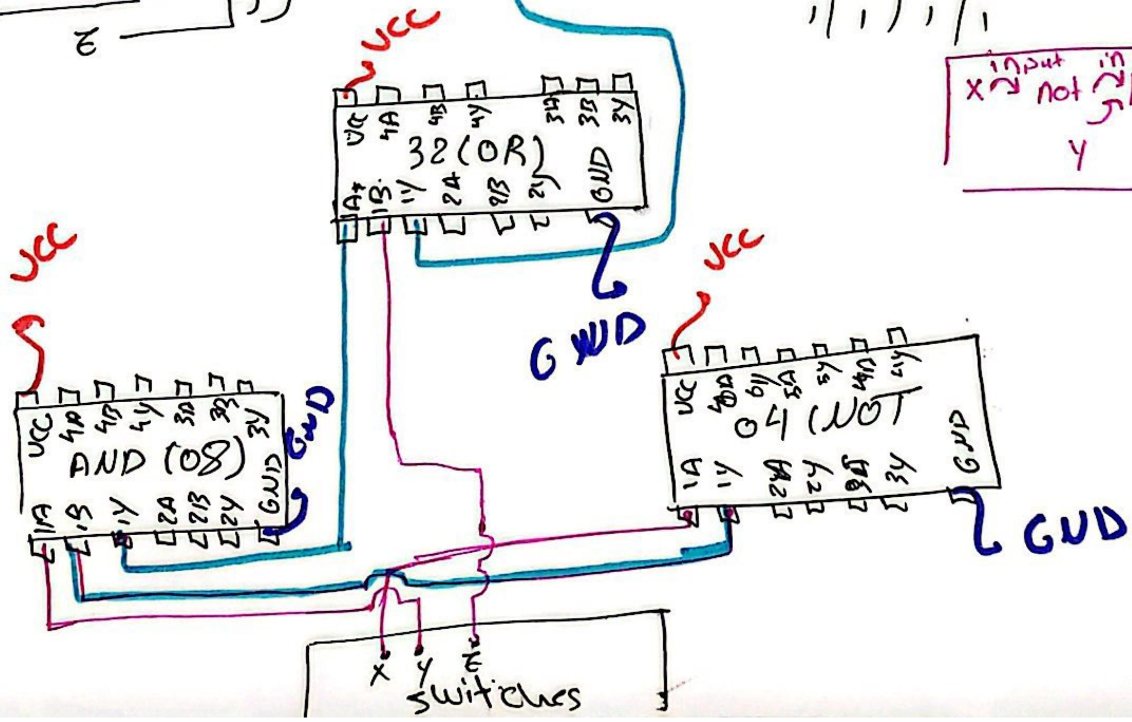
* NOT $\bar{x} = z$

x	z
0	1
1	0

→ $F = \bar{X} \cdot Y + Z$

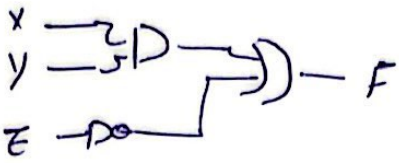
truth table

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

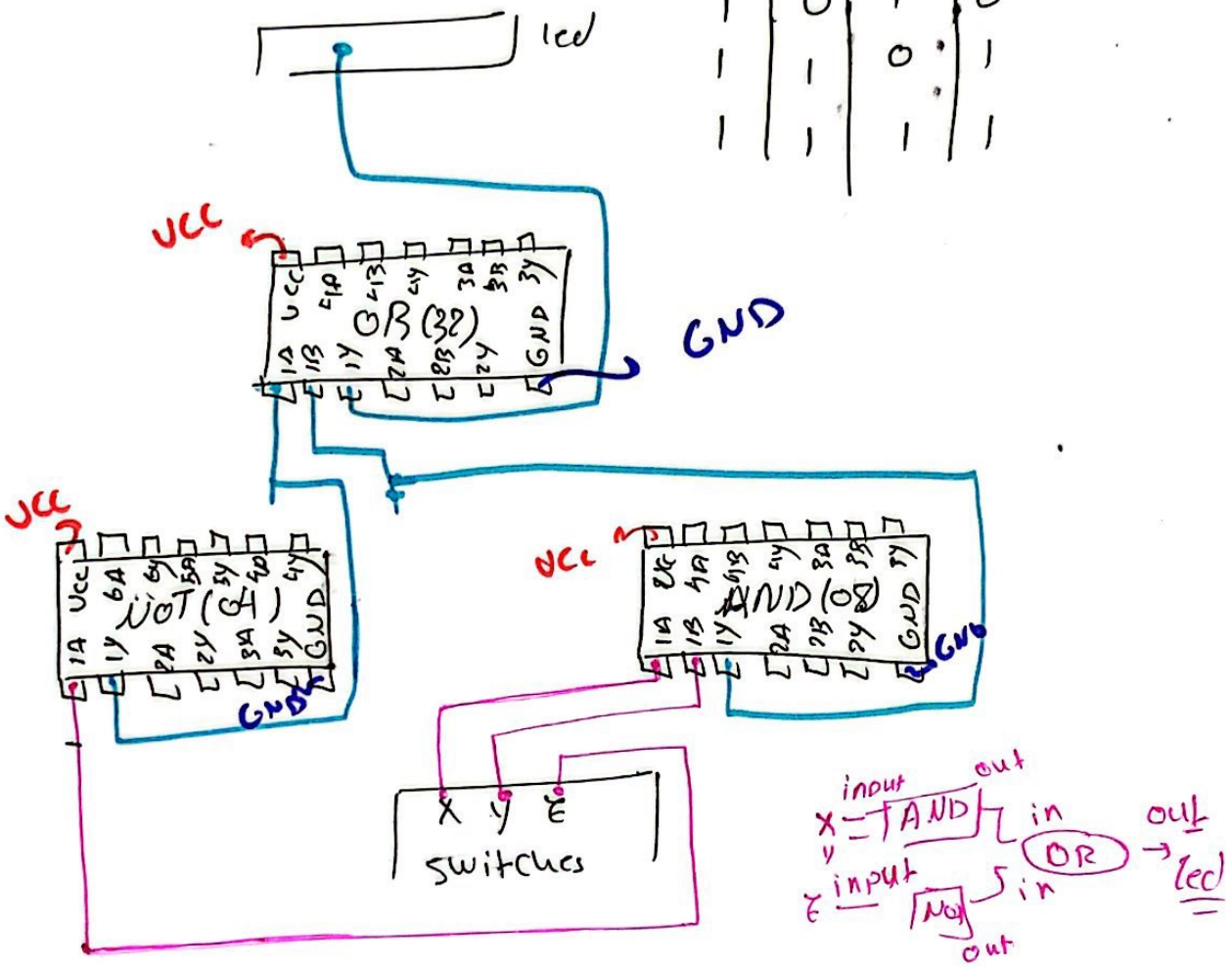


X input → not → in AND → in OR
Y Z

$F = x.y + \bar{z}$



x	y	z	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



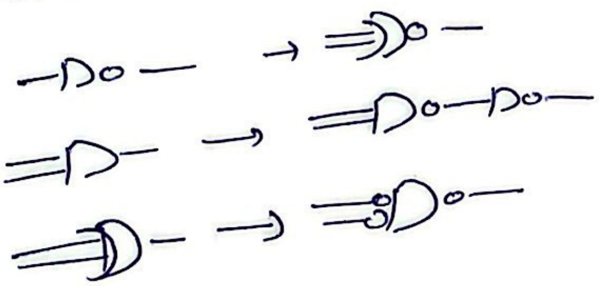
→ $X \oplus Y = X\bar{Y} + \bar{X}Y$

→ $\overline{X \oplus Y} = XY + \bar{Y}\bar{X}$

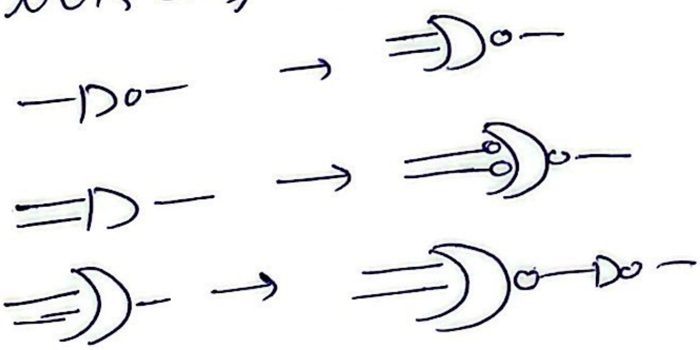
→ $\overline{XY} = \bar{X} + \bar{Y}$

→ $\overline{X+Y} = \bar{X} \cdot \bar{Y}$

NAND only:



NOR only:



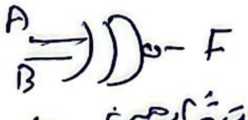
XOR $A \oplus B = F$

X	B	F
0	0	0
0	1	1
1	0	1
1	1	0

المتشابهين ← zero

المختلفين ← one

XNOR $\overline{A \oplus B} = F$



A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

المتشابهين ← one

المختلفين ← zero

* simplification

$XY + \bar{X}Y = Y$

$X + XY = X + Y$

$\bar{X} + XY = \bar{X} + Y$

$X \oplus 0 = X$

$X \oplus 1 = \bar{X}$

$\overline{(X \oplus Y)} = XY + \bar{X}\bar{Y}$

K-maps.

$\Sigma(-, -, -)$ → هوية واحدات

$\pi(-, -, -)$ → هوية أصفار

يعني $\Sigma(m_0, m_2)$

$m_0 \to 1$

$m_2 \to 1$

2 variable

x \ y	0	1
0	m_0	m_1
1	m_2	m_3

3-Variable

x \ yz	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

4-Variable

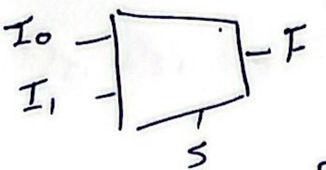
xy \ zw	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

بنحاول ناخذ أكبر عدد من الجيران
والمجموعة فيها أكبر عدد من الخلايا

Exp 3 : Multiplexer. 2x1 , 4x1 , 8x1, 16x1

بيختار واحد من ال input ويطلعه output

2x1

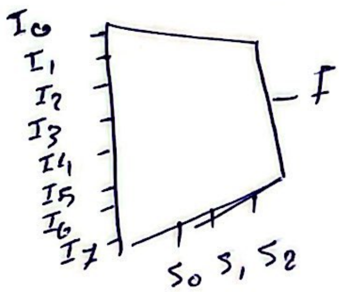


$S=0 \rightarrow F=I_0$

$S=1 \rightarrow F=I_1$

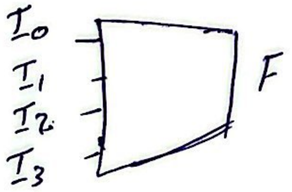
$F = I_0\bar{S} + I_1 \cdot S$

8x1



S_0	S_1	S_2	F
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

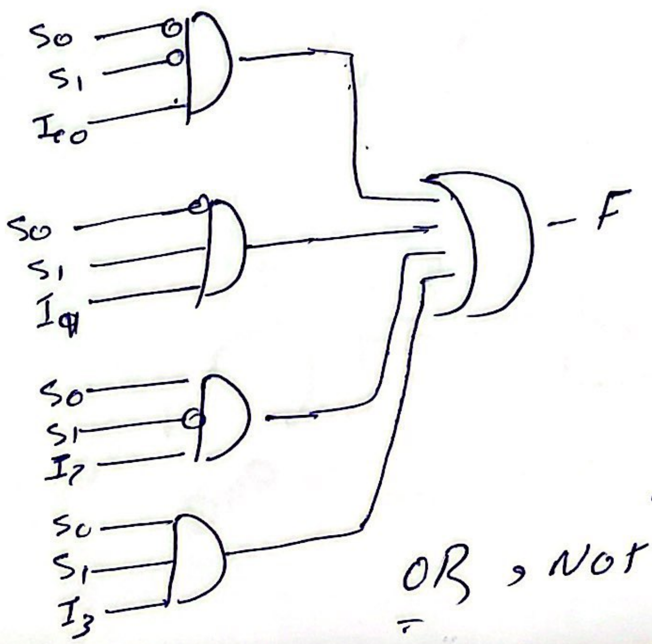
Design 4x1 MUX



S_0	S_1	F
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$[F = I_0 m_0 + I_1 m_1 + I_2 m_2 + I_3 m_3]$

$F = I_0 \bar{S_0} \bar{S_1} + I_1 \bar{S_0} S_1 + I_2 S_0 \bar{S_1} + I_3 S_0 S_1$

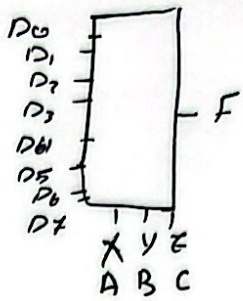


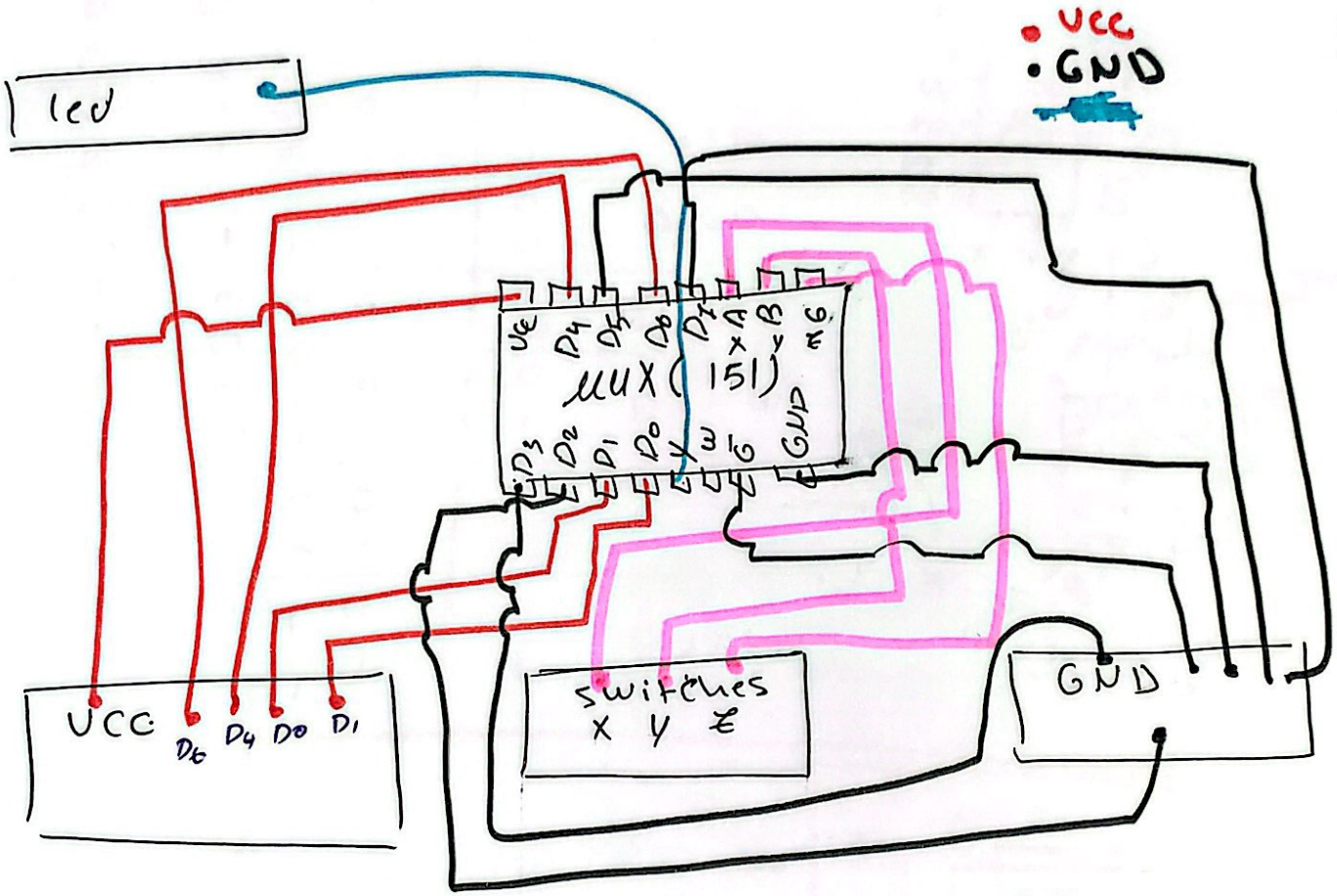
* بالعملي عنا IC بتكون MUX . طالما داخلي الشبك AND و NOT و OR

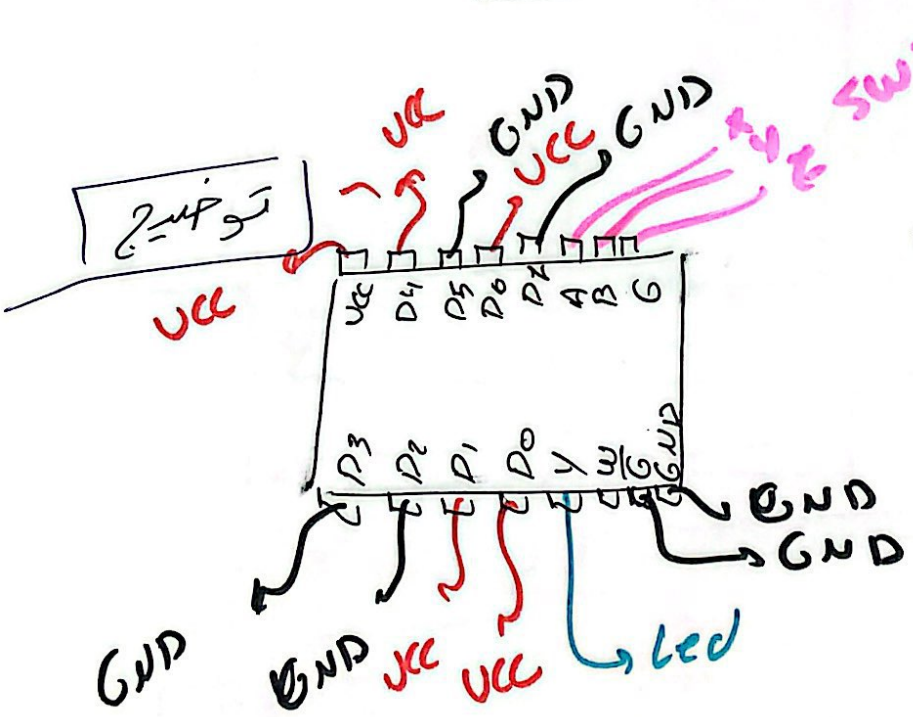
$F(x,y,z) = \Sigma(0,1,4,6)$

$D_0, D_1, D_4, D_6 \rightarrow VCC\ (1)$

$D_2, D_3, D_5, D_7 \rightarrow GND\ (0)$

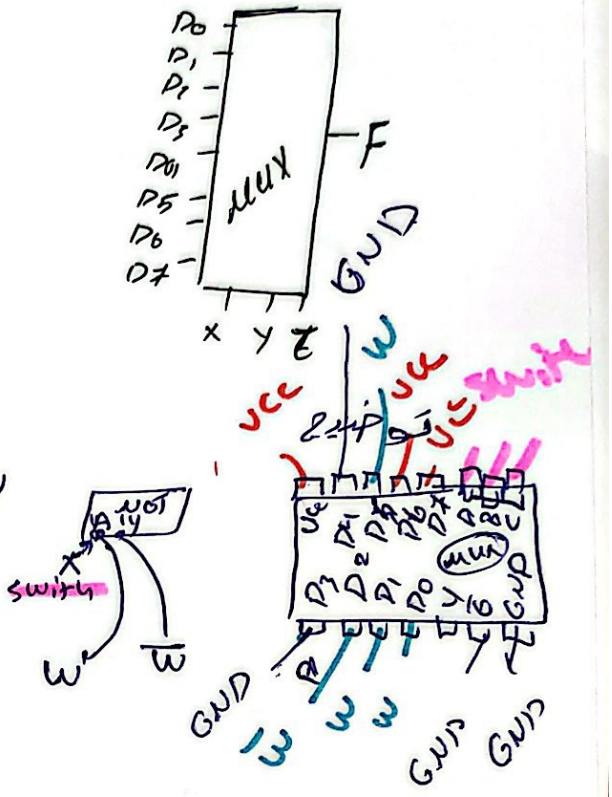


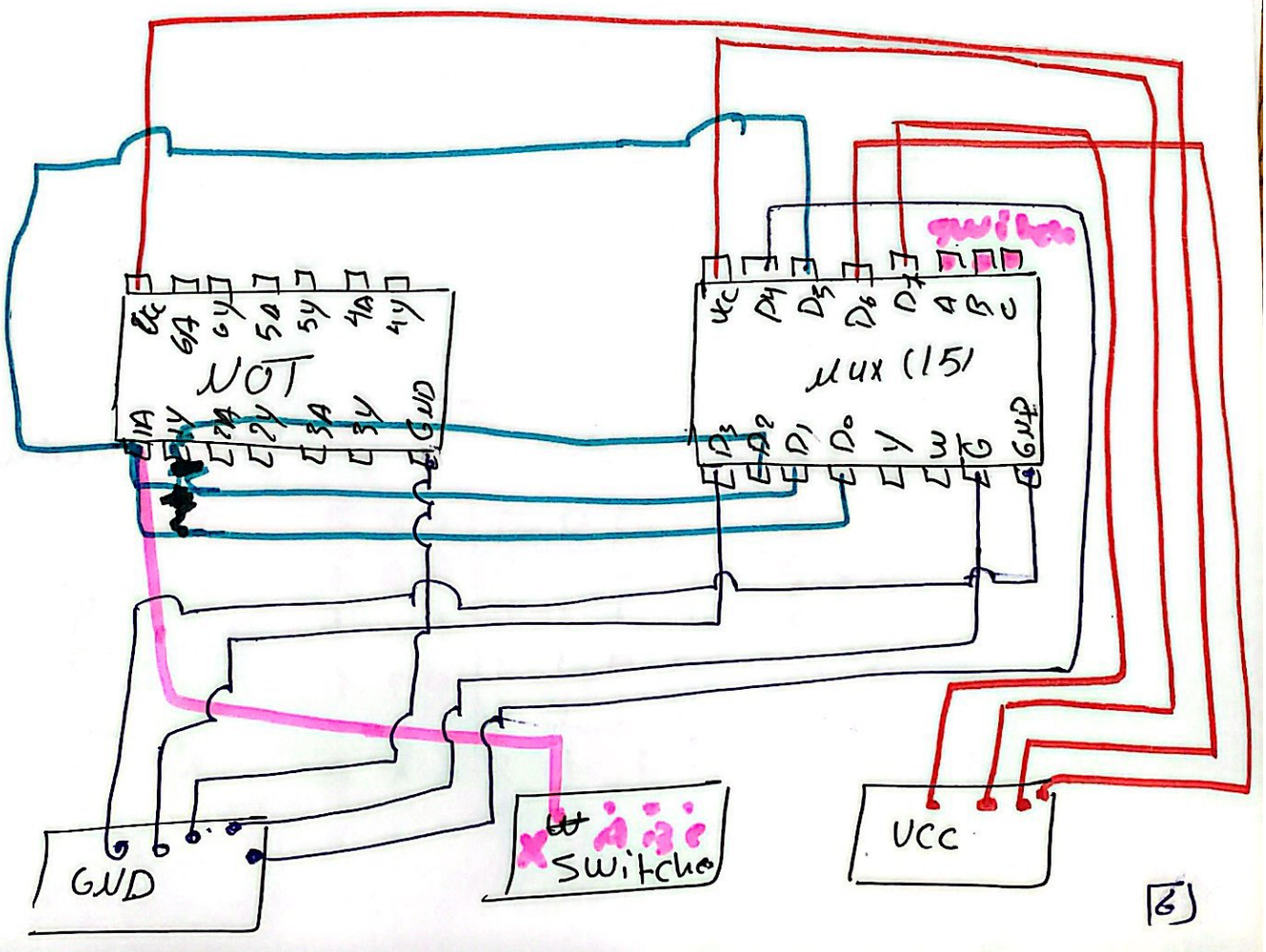




$F(x, y, z, w) = \Sigma(1, 3, 4, 11, 12, 13, 14, 15)$

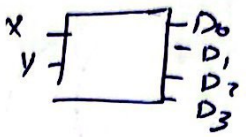
x	y	z	w	F	
0	0	0	0	0	$D_0 = w$
0	0	0	1	1	
0	0	1	0	0	$D_1 = w$
0	0	1	1	1	
0	1	0	0	1	$D_2 = \bar{w}$
0	1	0	1	0	
0	1	1	0	0	$D_3 = 0$
0	1	1	1	0	
1	0	0	0	0	$D_4 = 0$
1	0	0	1	0	
1	0	1	0	0	$D_5 = w$
1	0	1	1	1	
1	1	0	0	1	$D_6 = 1$
1	1	0	1	1	
1	1	1	0	1	$D_7 = 1$
1	1	1	1	1	





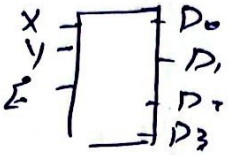
EXP 4 :- Decoder

* Active high decoder without enable.



X	Y	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

* Active high decoder with active high enable

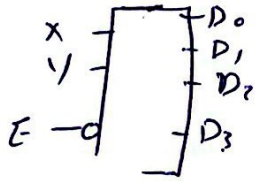


0 ← E ينشط الدائرة

E	X	Y	D_0	D_1	D_2	D_3
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1
0	X	X	0	0	0	0

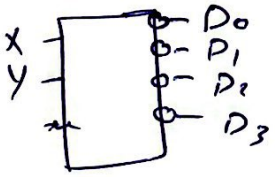
* Active high decoder with active low enable.

E=1 ينشط الدائرة



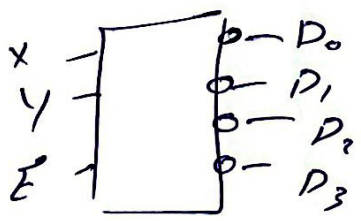
E	X	Y	D_0	D_1	D_2	D_3
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	1	0	0	0	1
1	X	X	0	0	0	0

* Active low decoder without enable.

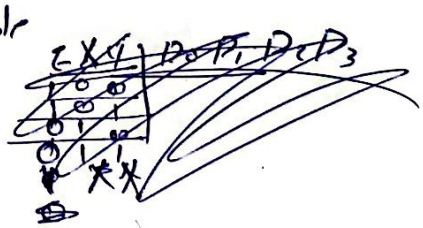


X	Y	D_0	D_1	D_2	D_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

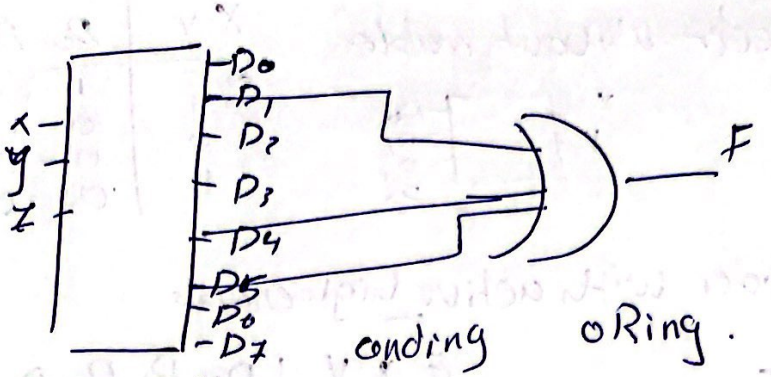
* Active low decoder with active high enable



E	X	Y	D_0	D_1	D_2	D_3
1	0	0	0	1	1	1
1	0	1	1	0	1	1
1	1	0	1	1	0	1
1	1	1	1	1	1	0
0	X	X	1	1	1	1

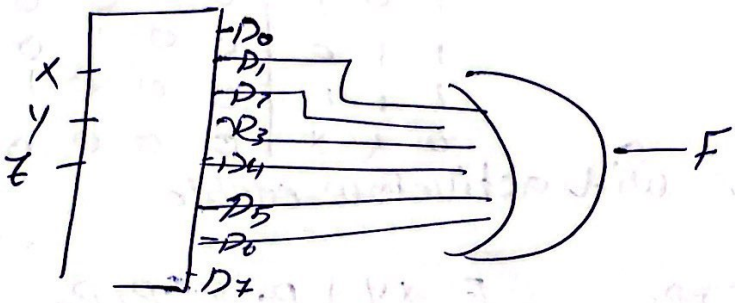


$F = \Sigma(0, 1, 4, 5)$



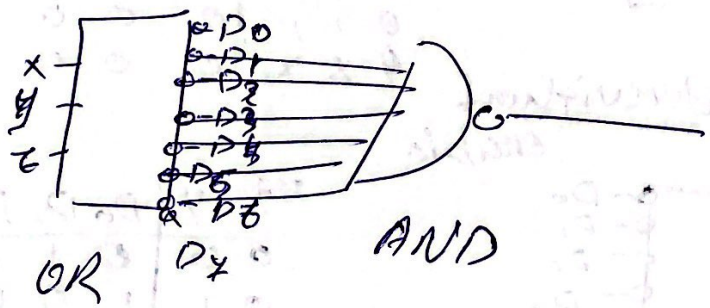
anding oRing.

$F = \Sigma(1, 2, 3, 4, 5, 6)$



$\overline{F} = \Pi(1, 2, 3, 4, 5, 6)$

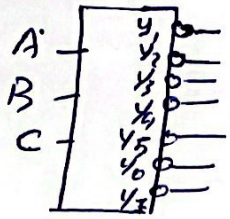
oRing then anding



OR AND

Decoder (Active low)

* الرمز العام



active low بال

input (101) → $Y_5 = 0$

$Y_0 - Y_7 \rightarrow 1$

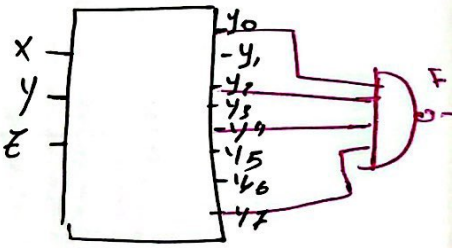
Y_5 صفر

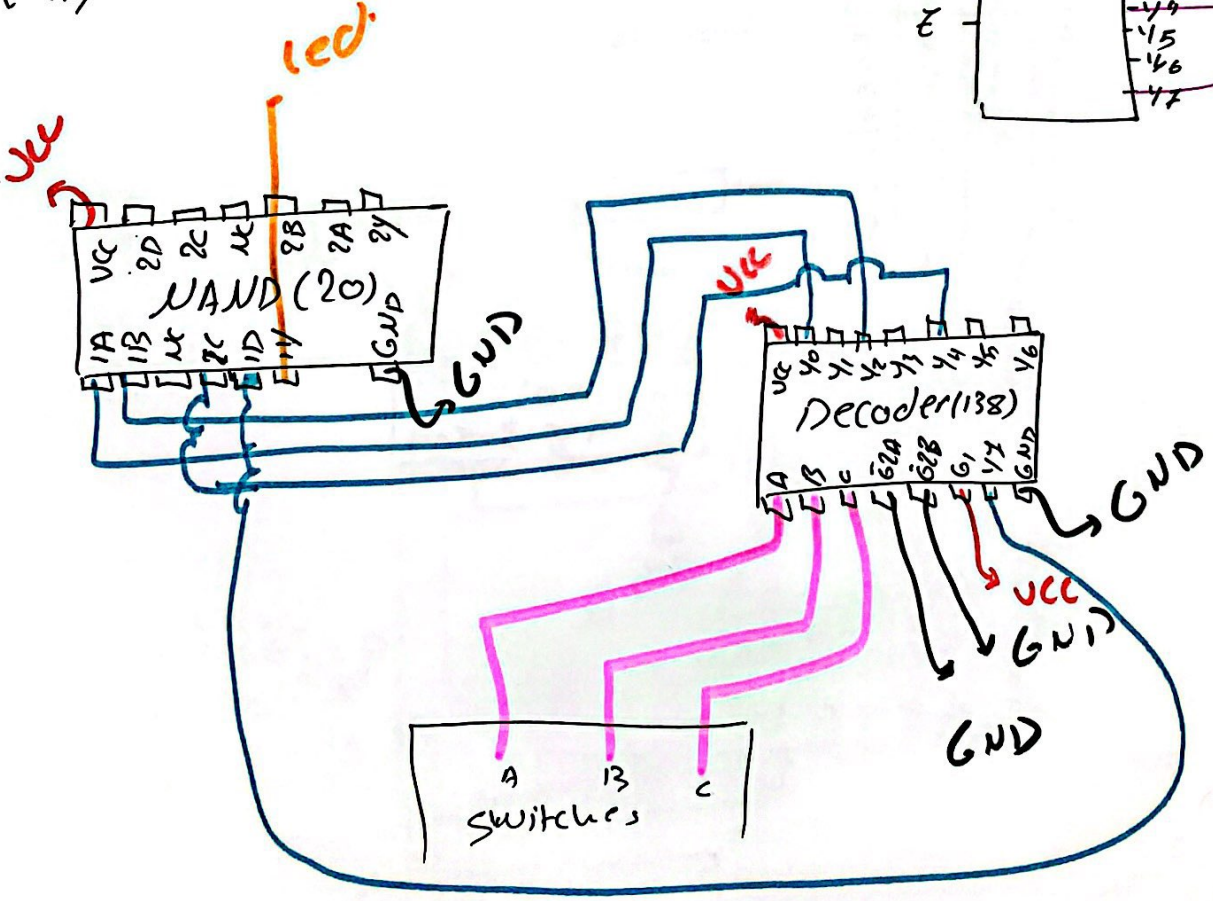
وال Active بالعكس high

* بال Active low يقدر ياعمل NAND

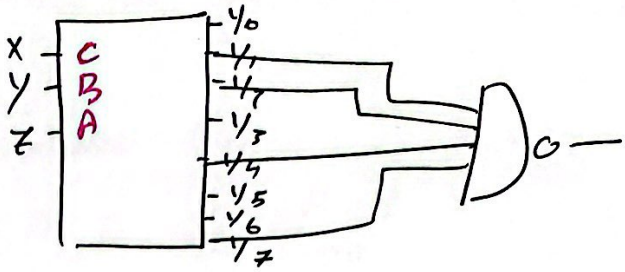
* بال Active high يقدر ياعمل OR

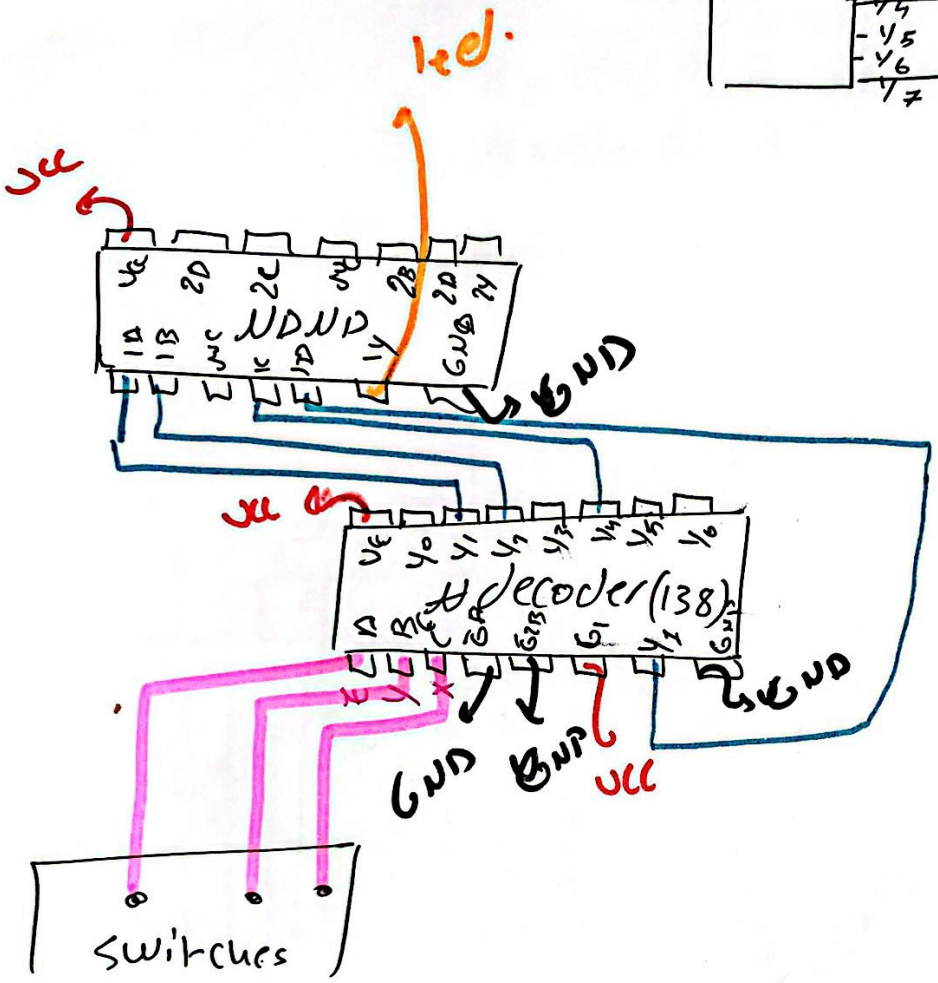
$F(X,Y,Z) = \Sigma(0, 2, 4, 7)$.





$F(x, y, z) = \Sigma(1, 2, 4, 7)$





EXP 4

→ Comparator.

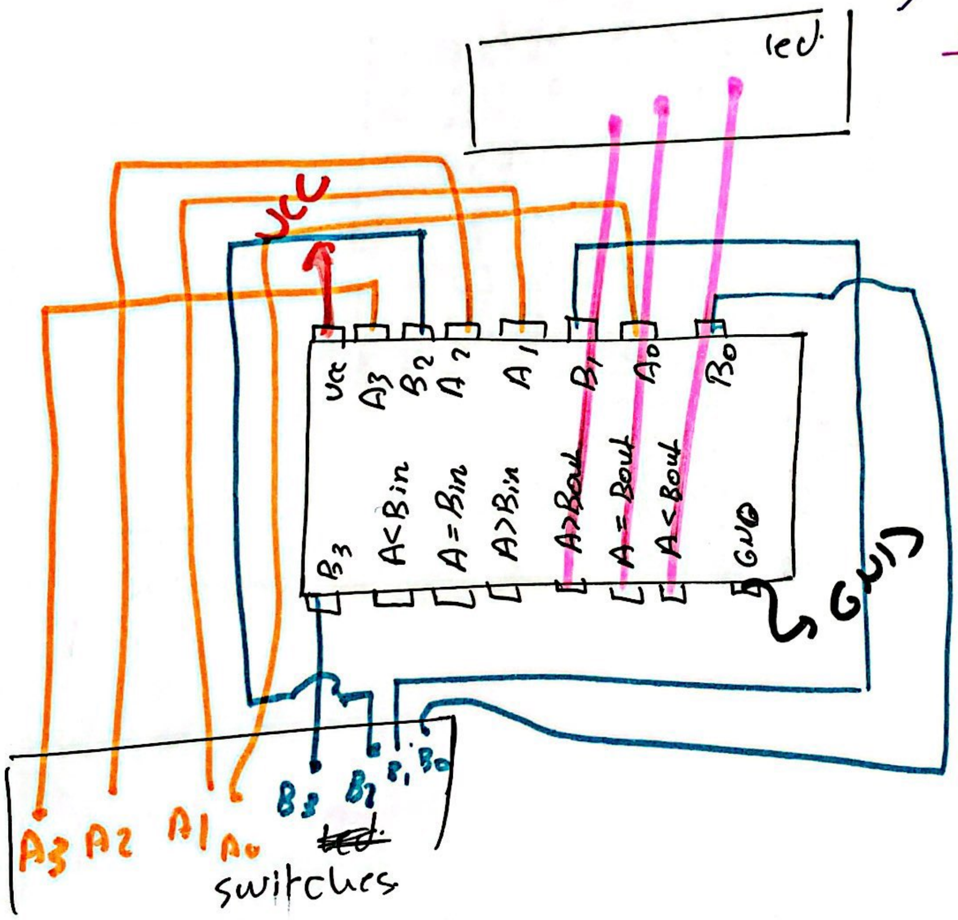
A & B each one 4 Bits.

A3 A2 A1 A0

B3 B2 B1 B0

A > B (1)
OR
A < B (0)
OR
A = B (0)

A > B اذا امثلا



* جابولنا ب Quiz نقارن ال A ب (3)

بحلولها 0011

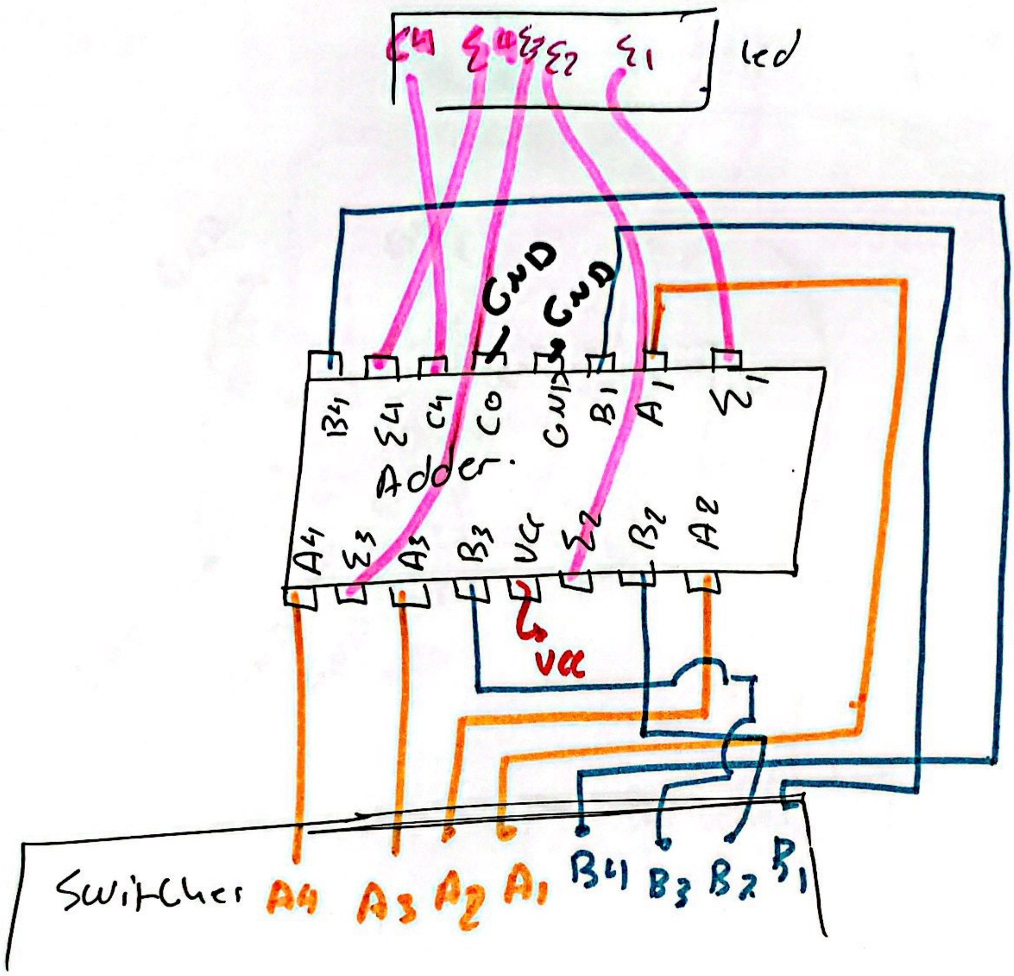
B3 B2 B1 B0

GND ← GND VCC VCC

دال A على ال switches و خلصنا بس حبيت اله الفكرة

*adder

A+B = Cout + sum.



* ركز معي ♡

(+) Adder
(-) subetracter } عايز استخدم ال Adder على انه

ازاي ؟؟

① طريقة واحدة [Co = 0 (GND → adder

[Co = 1 (Vcc → sub
+ $\overline{B}$ (ونشبك ال inverter على B
swithes → invertor → B)

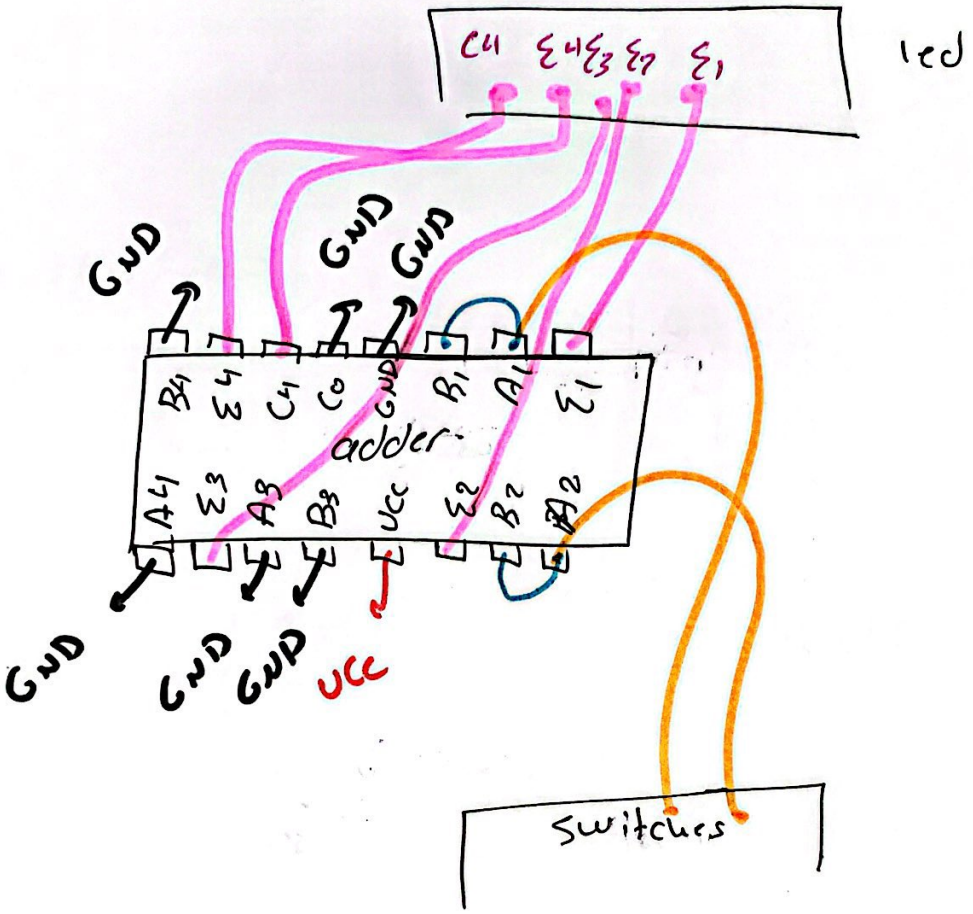
طريقة التانيه

sub ← 1 } Co
add ← 0 } Co واذا اعطيت

Co → XOR ←
B →

بنشبكها على XOR

F = A * B (using adder).



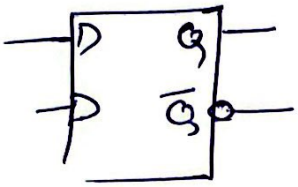
EXP5:- sequantial CCt

* بالباب بعملنا بس الـ JK Flip Flop

بس احتياط ادرس الـ D Flip Flop و T Flip Flop

عشان الامتحان النظري

→ D - Flip Flop

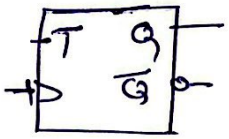


+Ve -Ve D	D	Q(t+1)
	0	0
	1	1

D	Q(t)	Q(t+1)
0	0	0
0	1	0
1	0	1
1	1	1

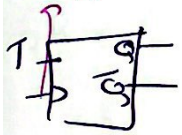
$D = Q(t+1)$

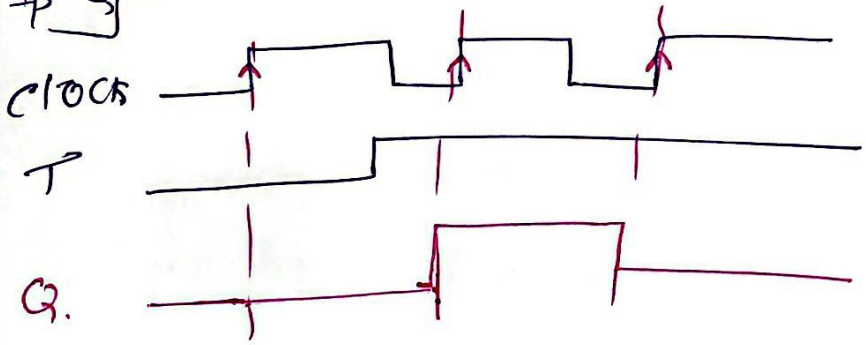
→ T - Flip Flop



(C) طالع -ve نازل +ve

T	Q(t+1)
0	No change
1	toggle



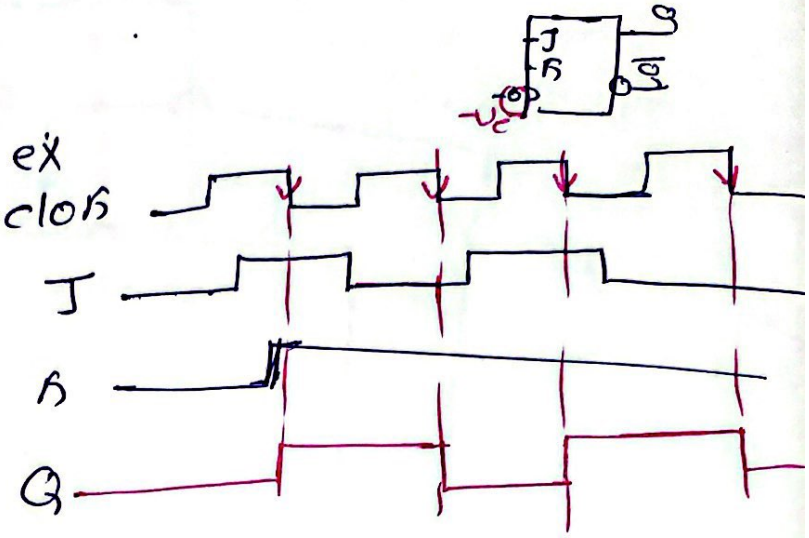


T	Q(t+1)	Q(t+1)
0	0	0
0	1	1
1	0	1
1	1	0

$Q(t+1) = T \oplus Q(t)$

→ JK Flip Flop

J	K	Q(t+1)	$\bar{Q}$(t+1)
0	0	No change	
0	1	Reset	
1	0	set	
1	1	toggle.	



Q	Q+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

A	B	X	A^*	B^*	JA	KA	JB	KB
0	0	0	1	1	1	X	1	X
0	0	1	1	1	1	X	1	X
0	1	0	0	0	0	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	1	X	0	1	X
1	0	1	0	1	X	1	1	X
1	1	0	1	0	X	0	X	1
1	1	1	0	1	X	1	X	0

يعطيني الجدول ضنه → B^* , A^* , A , B , X

KB , KA , JB , JA [1] يطلع

[2] بعمل map-K ويطلع معادلة كل

$JA = \bar{B}$

A \ BX	00	01	11	10
0	1	1	0	0
1	X	X	X	X

$JB = \bar{B}$

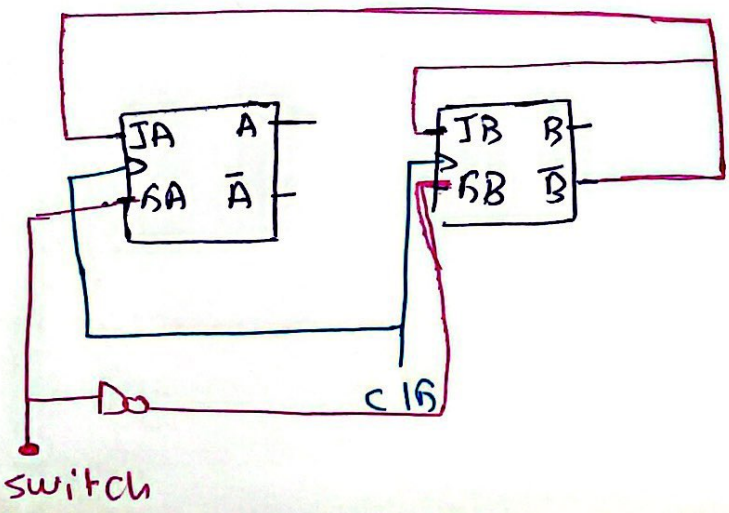
A \ BX	00	01	11	10
0	1	1	X	X
1	1	1	X	X

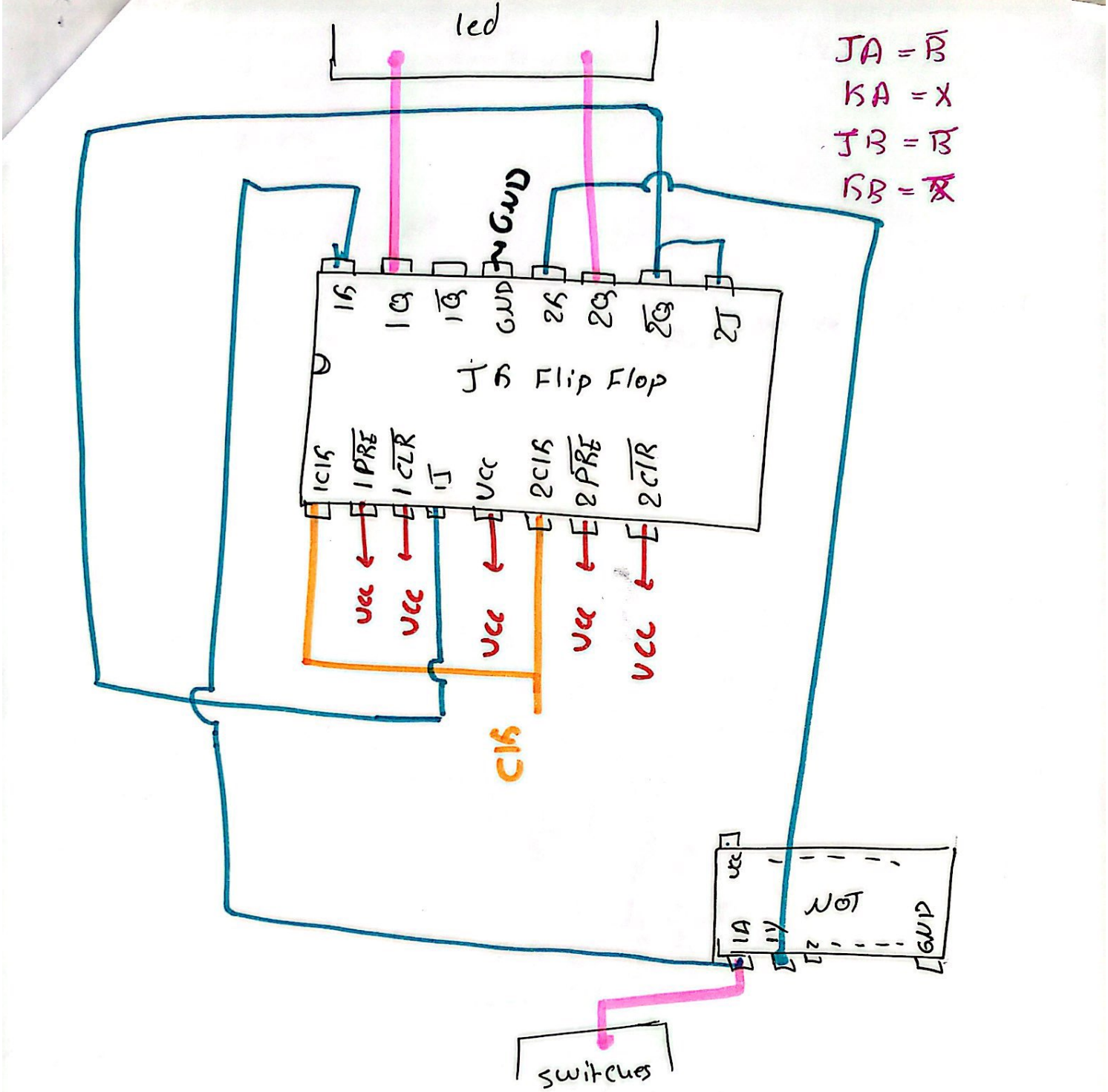
$KA = X$

A \ BX	00	01	11	10
0	X	X	X	X
1	0	1	1	0

$KB = \bar{X}$

A \ BX	00	01	11	10
0	X	X	0	1
1	X	X	0	1





EXP6

counters

161 → up Counter (0-F.)

MSB(D) LSB(A)

DC BA

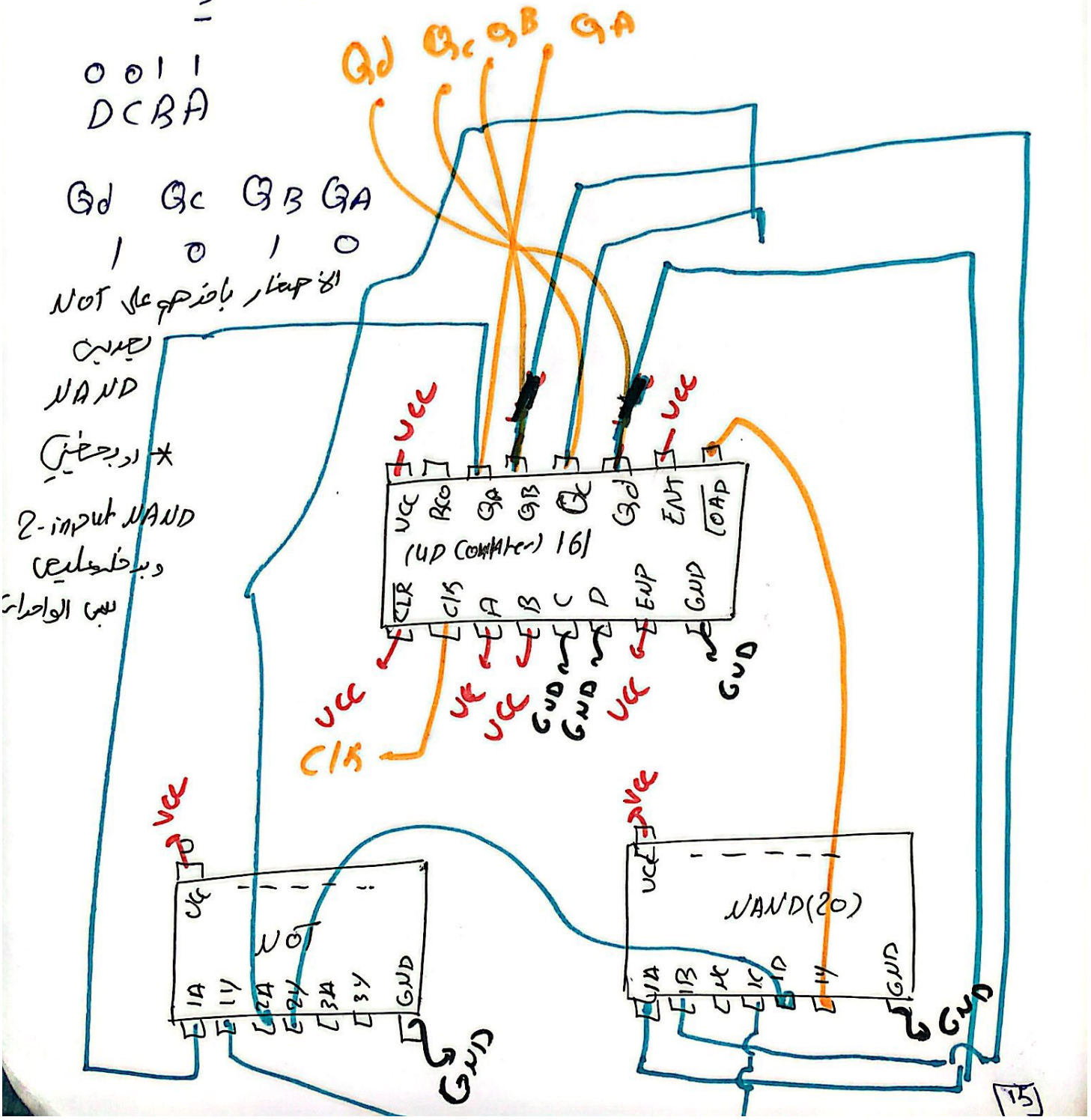
3 - 10

0 0 1 1

D C B A

Qd Qc QB QA

1 0 1 0



Scanned with CamScanner

193 up dow counter

up → clk down → 1 counting up (0 - F)

down → clk up → 1 counting down. (F - 0

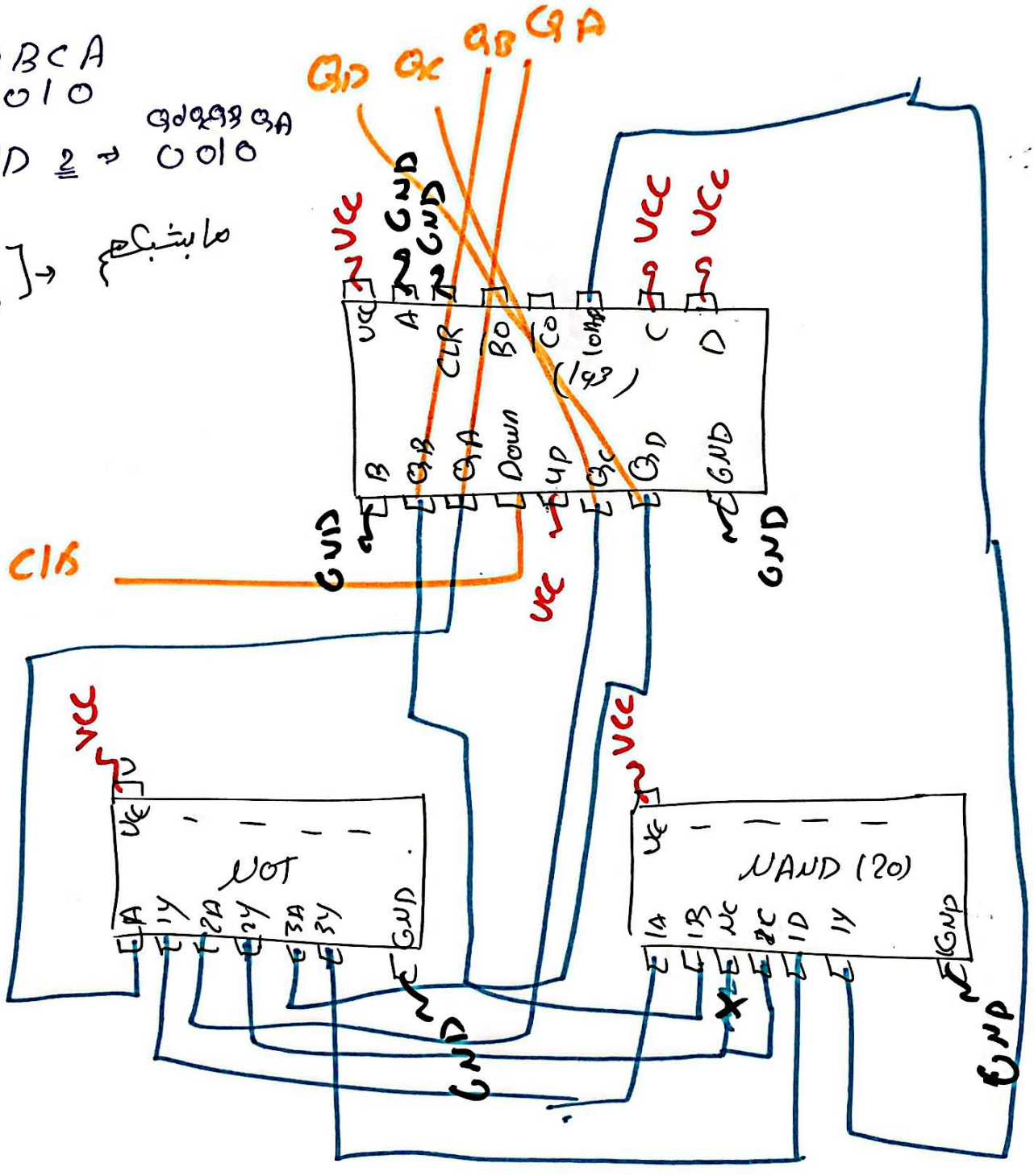
Next last value بتاخد NAND ال

* 10 - 3

D B C A
1 0 1 0

NAND 2 → 0 0 1 0

* $\overline{Co}$ / $\overline{Bo}$ → ما بتشتغل

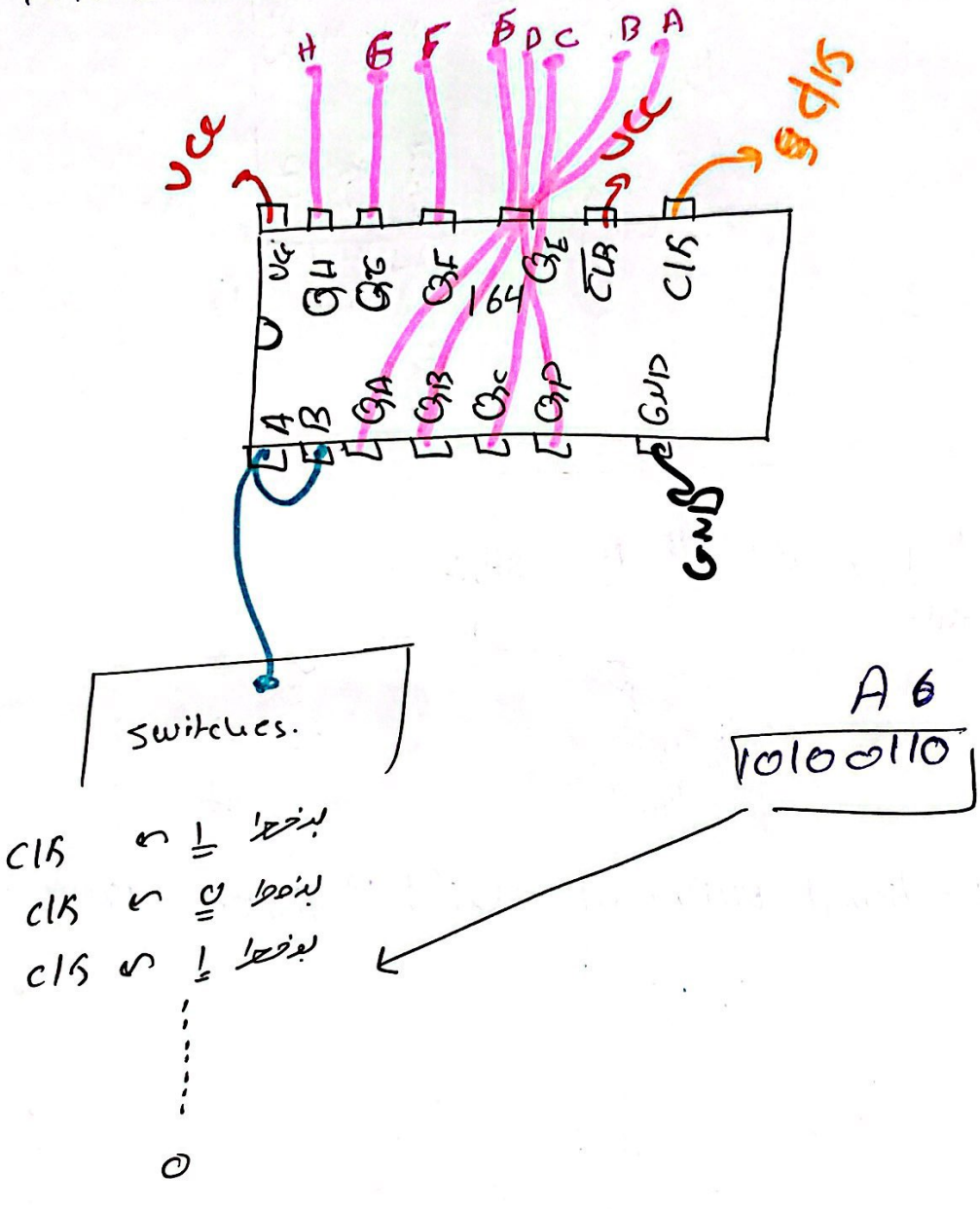


Exp:-8 Registors

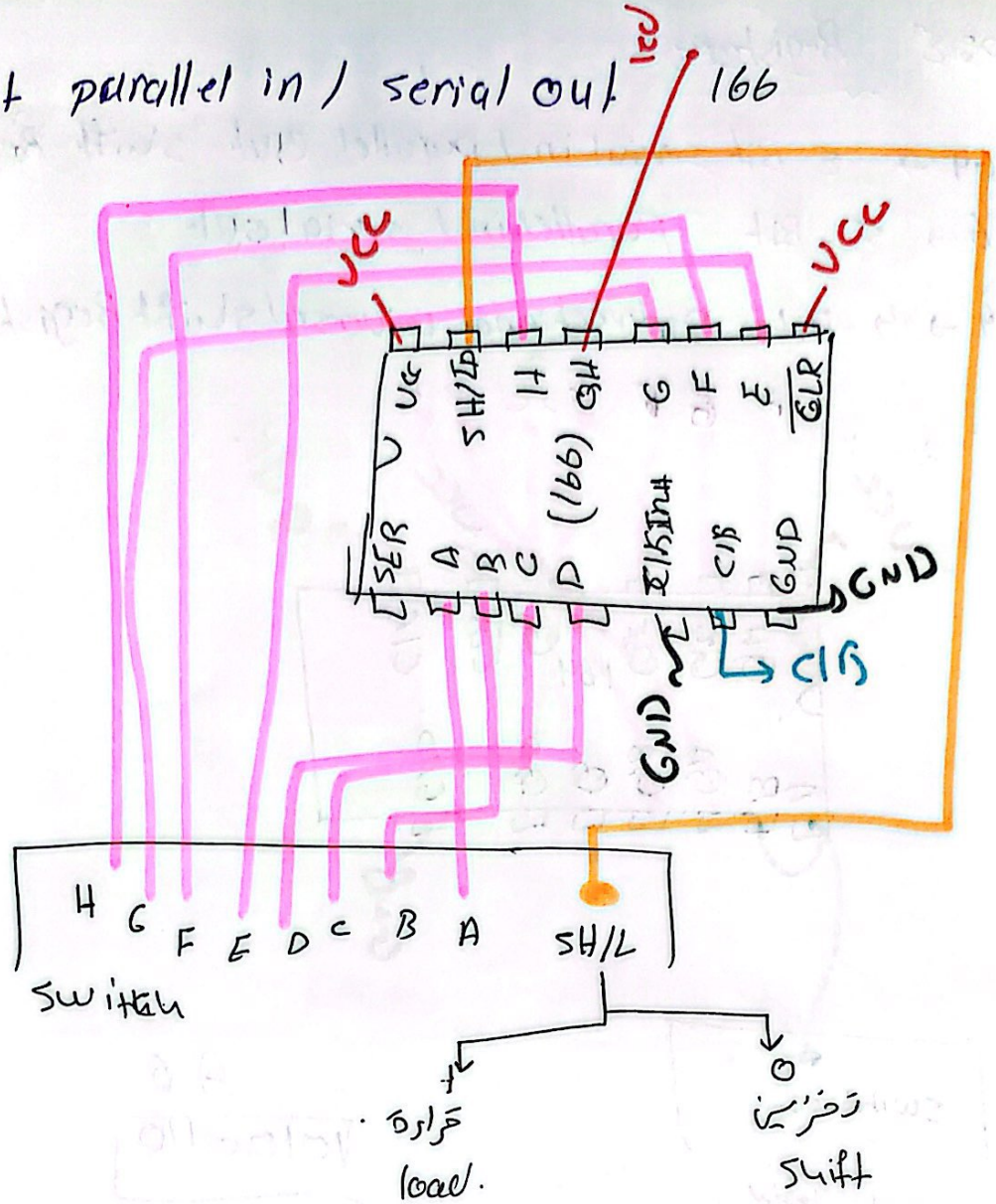
164 → 8-Bit serial in / parallel Out shift Register

166 → 8-Bit Parallel in / serial out

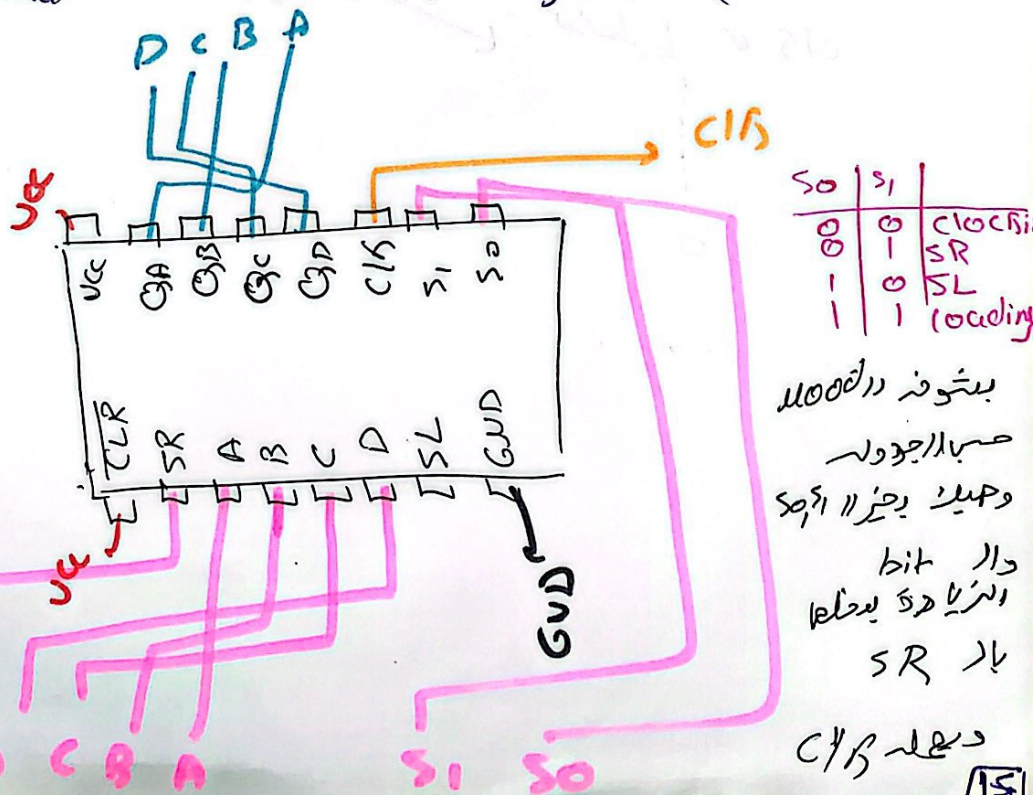
1944 → 4-Bit Bidirectional universal shift Register



8- Bit parallel in / serial out 166



4- Bit Bidirectional universal shift Register. (194)



بشوف الـmode
حسب الجدول
وهيك بخزن الـ bit
الزيادة بدخلها
بال SR

دخله CLK