ElCoM
ELECTRICAL COMPUTER MECHATRONICS

تقدم لجنة ElCoM الاكاديمية

دفتر لمادة:

منطق رقمي

من شرح:

فيديوهات نتالي الكايد

مضاف إليه حل اسئلة اضافية

جزيل الشكر للطالب:

صالح الغول

بخط:

هبة كتانة

Logic

chapter 1:-

* Digital systems of Binary numbers.

* Nember systems :-

ثنائي	عشري	ثماني
1) Binary	2) Decimal	3) octal
0, 1	0, 1, 2, ..., 9	0, 1, 2, ..., 7
2- digits	10 - digits	8- digits
$(100110)_2$	(753)	$(375)_8$
100110 B		375Q
% 100110		

السادس عشر

4) Hexa Decimal

0-9, A-F

16 - digits

$(3F)_{16}$

3FH

$ 3F

A = 10

B = 11

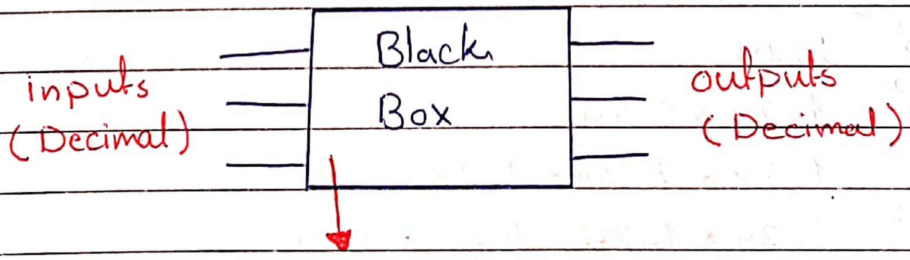
C = 12

D = 13

E = 14

F = 15

*



Black Box operations are in (Binary system) ??

Because the black Box consist of cct,s

of any cct has only 2-cases: (ON, oFF)

of the Best sys to represent that is the Binary.

* The Base:-

- Binary → base 2
- octal → base 8
- Decimal → base 10
- hexa Decimal → base 16

* The Basic logic gats :-

1) AND 2) OR 3) Not

* LSB & MSB:-

1) LSB → Least significant Bit .
2) MSB → Most significant Bit .

1010011

MSB LSB

*Bit = 0 or 1

Nibble = 4 bits

Byte = 8 bits

word = 2 bytes = 16 bits

1 kilo = 2^{10} = 1024

1 Mega = 2^{20} = 1024 * 1024

1 Giga = 2^{30} = 1024 * 1024 * 1024

1 Tera = 2^{40} = 1024 * 1024 * 1024 * 1024

⚠ Ex:- find the exact # of bits of 64 Mega bytes.

Sol:- $64 * 2^{20} * 8 = 64 * 1024 * 1024 * 8$

*** The weight :-** weight = $(\text{Base})^{(\text{status number})}$

(status #) .... 2 1 0 -1 -2

$(183\ 56)_{10} \rightarrow 1 * 10^{2} + 8 * 10^{1} + 3 * 10^{0} + 5 * 10^{-1} + 6 * 10^{-2}$

(Factors) (Base)

$= 183,56$

*** Digital sys.s table :-**

Decimal	octal	hexa Decimal	Binary	
0	0	0	0	صفر
1	1	1	1	واحد
2	2	2	10	عشرة
3	3	3	11	إحدى عشر
4	4	4	100	مية
5	5	5	101	مية وواحد
6	6	6	110	مية وعشرة
7	7	7	111	مية وحداش
8	10	8	1000	ألف
9	11	9	1001	ألف وواحد
10	12	A	1010	ألف وعشرة
11	13	B	1011	ألف وحداش
12	14	C	1100	ألف ومية
13	15	D	1101	ألف ومية وواحد
14	16	E	1110	ألف ومية وعشرة
15	17	F	1111	ألف ومية وحداش
16	20	10	10000	عشرة آلاف

* Conversion :-

1) from any sys to Decimal :- [Sum of weights]

2) from Decimal to any sys :- a) integer → [Division]

b) fraction → [Multibly]

⚠ Ex:- 1) $(157,4)_8 = (\quad)_{10}$?

$(\overset{2}{1}\overset{1}{5}\overset{0}{7},\overset{-1}{4})_8 = 1*8^2+5*8^1+7*8^0+4*8^{-1}$

$= 64+40+7+4/8$

$= (111,5)_{10}$

2) $(2A,F) = (\quad)_{10}$?

$(\overset{1}{2}\underset{10}{\overset{0}{A}},\underset{16}{\overset{-1}{F}})_{10} = 2*16^1+10*16^0+15*16^{-1}$

$= 32+10+15/16$

$= (42,9375)_{10}$

3) $(11011,10)_2 = (\quad)_{10}$?

$(\overset{4}{1}\overset{3}{1}\overset{2}{0}\overset{1}{1}\overset{0}{1},\overset{-1}{1}\overset{-2}{0})_2 = 1*2^4+1*2^3+0+1*2^1+1*2^0+1*2^{-1}+0$

$= 16+8+2+1+1/2$

$= (27,5)_{10}$

4) $(123)_4 = (\quad)_{10}$?

$123 = 1*4^2+2*4^1+3*4^0 = 16+8+3 = (27)_{10}$

5) $(8)_{10} = (\ \ )_2$?

integer → Devision on the Base of needed sys.

$\frac{8}{2} = 4$ Remain 0 LSB

$\frac{4}{2} = 2$ Remain 0

$\frac{2}{2} = 1$ Remain 0

$\frac{1}{2} = 0$ Remain 1 ↑ MSB → (البسط)

So = $(1000)_2$

6) $(100)_{10} = (\ \ )_2$?

$\frac{100}{2} = 50$ 0 LSB

$\frac{50}{2} = 25$ 0

$\frac{25}{2} = 12$ 1 → (1) بزيد $(12 = \frac{24}{2})$ أقرب اشي إلها

$\frac{12}{2} = 6$ 0

$\frac{6}{2} = 3$ 0

$\frac{3}{2} = 1$ 1 → (1) ويزيد $(1 = \frac{2}{2})$

$\frac{1}{2} = 0$ 1 ↑ MSB

So = $(1100100)_2$

7) $(100)_{10} = (\quad)_8$?

$\frac{100}{8} = 12$ 4 LSB → (4) بنضل $(12 = \frac{96}{8})$

$\frac{12}{8} = 1$ 4 → (4) بنضل $(1 = \frac{8}{8})$

$\frac{1}{8} = 0$ 1 ↑ MSB So $= (144)_8$

8) $(29)_{10} = (\quad)_{16}$?

$\frac{29}{16} = 1$ 13 LSB → (13) بنضل $(1 = \frac{16}{16})$

$\frac{1}{16} = 0$ 1 ↑ MSB So $= 113 = (1D)_{16}$

9) $(64)_{10} = (\quad)_4$?

$\frac{64}{4} = 16$ 0

$\frac{16}{4} = 4$ 0

$\frac{4}{4} = 1$ 0

$\frac{1}{4} = 0$ 1 ↑ So $= (1000)_4$

10) $(5)_{10} = (\quad)_8$?

$\frac{5}{8} = 0$ 5 So $= (5)_8$

11) $(57)_{10} = (\quad)_2$?

~~57/2~~

57/2 = 28 1 → LSB بجمل (1) $(28 = \frac{56}{2})$

24/2 = 14 0

14/2 = 7 0

7/2 = 3 1 So = $(111001)_2$

3/2 = 1 1

1/2 = 0 1 → MSB

12) $(0,3)_{10} = (\quad)_2$? precision = 5

Fraction → Multibly by the base of needed sys.

MSB

0,3 * 2 = 0,6 ①

0,6 * 2 = 1,2 ②

0,2 * 2 = 0,4 ③

0,4 * 2 = 0,8 ④

0,8 * 2 = 1,6 ⑤ Stop So = $(0,0100\overline{1})_2$

LSB

هون العدد الي بين القوسين
دوري (متكرر للالانهاية) بحيث
يكرر نفسه بعد ما أوصل للمرة
الخامسة من الضرب

13) $(0,625)_{10} = (\quad)_2$?

0,625 * 2 = 1 , 250 MSB

, 250 * 2 = 0 , 500

, 500 * 2 = 1 , 000 LSB stop [Fractions = zeros]

So = $(0,101)_2$

تنساش الفاصلة أبدًا ما

14) $(21,75)_{10} = (\quad)_2$ ؟

integer of fraction solve each part in alone.

[1]
$21/2 = 10$ 1 LSB
$10/2 = 5$ 0
$5/2 = 2$ 1
$2/2 = 1$ 0 ↑
$1/2 = 0$ 1 MSB

[2]
MSB
$,75 * 2 = 1,50$
$,50 * 2 = 1,00$
LSB

So $= (10101,11)_2$

* Convertion between Binary ↔ octal :-

Base = 2	Base = 8 = $(2)^3$ → [Group of 3]
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

ex:- $(10\ 110\ 111,\ 1011)_2$
$= (\quad)_8$ ؟

$$\underbrace{010}_{\cancel{4}2\cancel{1}}\ \underbrace{110}_{42\cancel{1}}\ \underbrace{111}_{421}\ ,\ \underbrace{101}_{4\cancel{2}1}\ \underbrace{100}_{4\cancel{2}\cancel{1}}$$

← →

Take groub of 3

powers of (2): بجمع كل مجموعة لحال

So $= (267,54)_8$

△* Ex:- $(1001011010111, 1011111)_2 = (\quad)_8$?

$\underbrace{010}_{\cancel{4}2\cancel{1}}\underbrace{010}_{\cancel{4}2\cancel{1}}\underbrace{110}_{42\cancel{1}}\underbrace{101}_{4\cancel{2}1}\underbrace{111}_{421},\underbrace{101}_{4\cancel{2}1}\underbrace{111}_{421}\underbrace{100}_{4\cancel{2}\cancel{1}}$

$= (22657, 574)_8$

* Conversion Between Binary $\xleftrightarrow{8}$ hexadecimal :-

Same previous way but take groups of 4 - Bits :
because Base = 16 = 2^4

*Example :- $(00100101101 0111, 1011 1110)_2 = (\quad)_{16}$?

Sol:- $\underbrace{0010}_{\cancel{8}\cancel{4}2\cancel{1}}\ \underbrace{0101}_{\cancel{8}4\cancel{2}1}\ \underbrace{1010}_{8\cancel{4}2\cancel{1}}\ \underbrace{1111}_{8421},\underbrace{1011}_{8\cancel{4}21}\ \underbrace{1110}_{842\cancel{1}}$

$\therefore = (25AF, BE)_{16}$

② $\underbrace{000(1}_{\cancel{8}\cancel{4}\cancel{2}1}\ \underbrace{1110}_{842\cancel{1}}\ \underbrace{1001}_{8\cancel{4}\cancel{2}1},\underbrace{0100}_{\cancel{8}4\cancel{2}\cancel{1}}\ \underbrace{11)_2 00}_{84\cancel{2}\cancel{1}} = (\quad)_{16}$?

$\therefore = (1E9, 4C)_{16}$

* Conversion from (octal to Binary):

→ Convert each octal # to 3 Binary Bits.

* conversion from (hexa to Binary):

→ Convert each hexa # to 4 Binary Bits.

△ EX:-

1) $(751,23)_8 = (\quad)_2$?

751 , 23

421 421 421 421 421

111 101 001 101 011

So = $(111\,101\,001, 010\,011)_2$

2) $(A2F , 1C)_{16} = (\quad)_2$?

A 2 F , 1 C

8421 8421 8421 8421 8421

1010 0010 1111 0001 1100

So = $(1010\,0010\,1111, 0001\,1100)_2$

* on the fly :- to convert Decimal ~~#~~ to Binary.

$(57)_{10}$ = 32 16 8 4 2 1

$(1\ 1\ 1\ 0\ 0\ 1)_2$

25 9

$(100)_{10}$ = 64 32 16 8 4 2 1

$(1\ 1\ 0\ 0\ 1\ 0\ 0)_2$

36 4

* unsigned number :- [+ve]

* signed number :- a) signed magnitude

b) complement : 1's or 2's

☒ Unsigned number :-

1) Base = B = r = 10 , [N: Number of digits]

n	2	3	Rule
Min	0	0	0
Max	99	999	$r^n - 1 = 10^n - 1$
number of values	100	1000	$r^n = 10^n$
number of Zeros	1	1	1

$\underline{0}\ \underline{0}$

$\underline{9}\ \underline{9}$ Decimal

$\underline{0}\ \underline{0}\ \underline{0}$

$\underline{9}\ \underline{9}\ \underline{9}$

2) Base = B = r = 2

n	2	3	Rule
Min	00 = 0	000 = 0	0
Max	$(11)_2 = 3$	$(111)_2 = 7$	$r^n - 1 = 2^n - 1$
Number of Values	4	8	$r^n = 2^n$
Nember of Zeros	1	1	1

$\underline{0}\,\underline{0}$

$\underline{1}\,\underline{1}$ Binary

$\underline{0}\ \underline{0}\ \underline{0}$

$\underline{1}\ \underline{1}\ \underline{1}$

⚠ Ex:- Find the Min number of bits required to represent :-

1) 1201 ?

$1201 = 2^n - 1 \longrightarrow 1202 = 2^n$, $2^{10} = 1024 <$ الرقم

$2^{11} = 2048 >$ الرقم ✓

so $n = 11$

2) 1024 ?

$1024 = 2^n - 1 \longrightarrow 1025 = 2^n$, $2^{10} = 1024 <$ الرقم ✗

So $n = 11$ $2^{11} = 2048 >$ الرقم ✓

3) 1000 ?

$1000 = 2^n - 1 \longrightarrow 1001 = 2^n$, $2^9 = 512$ ✗

So $n = 10$ $2^{10} = 1024$ ✓

⚠ Ex:- unsigned number n=8 , find the Range ?

$$\begin{aligned} \text{Range} &= (0 - 2^n - 1) \\ &= (0 - 2^8 - 1) \\ &= (0 - 256 - 1) \\ &= (0 - 255) \end{aligned}$$

✓ $(2)^1 \nearrow (1024)$ قوى الرقم 2 لازم تحفظهم منيح من ←

① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

* 2 - powers :- 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024

* Binary code : sequance of 0,1 that represent a certain value.

* Signed magnitude , [SM] :-

IF MSB = 0 , then (+ve) , ex:- $\underline{0}1011$ = +ve

IF MSB = 1, then (−ve) , ex:- $\underline{1}0100$ = −ve

⚠ Ex:- 1) n=4 , SM , write -3 ?

→ set 0 to fill the empty bits.

1 0 1 1

↓ Signe (-ve)

مهمنا ابدأ الكتابة
(بلش عكسي من الـ LSB
يعني)

powers:- 8 4 2 1 (on the fly)

0 0 (1 1) = 3

2) n = 5 , SM , write -27 ?

!! 1 1 0 1 1 !! ∝

↳ = (27) not (-27)

16	8	4	2	1
1	1	0	1	1

Still need another bit for signe

So -27 with n=5 is out of Range & can't Represent.

3) Find the Min # of bits required to represent -27 in SM ? (n = 6) at least.

4) $(00\,00)_{SM}$? [(00 00) what equal in SM sys]

= +0

5) $(1000)_{SM}$? = -0

6) $(0101101)_{SM}$?

0101101 = +32 + 8 + 4 + 1 = +45

↓ $32\ \cancel{16}\ 8\ 4\ \cancel{2}\ 1$

Signe

7) Fill the table :-

n	2	3	Rule
Min	$(11)_2 = -1$	$(111)_2 = -3$	$-(2^{n-1}-1)$
Max	$(01)_2 = +1$	$(011)_2 = +3$	$2^{n-1}-1$
# of values	4	8	2^n
# of Zeros	2	2	2

1 1 = −1

0 1 = +1

1 1 1 = −3

0 1 1 = +3,

, Values :- −1 & +1

−0 & +0

So from −1 to +1 = $\boxed{4}$

8) find the Range if n=8, SM ?

Range = Min ⟶ Max

$-(2^{8-1}-1) \longrightarrow 2^{8-1}-1$

$-(2^{7}-1) \longrightarrow 2^{7}-1$

$[-127 \longrightarrow 127]$

⊛ Diminshed Radix complement, [1's complement] :-

[D.R.C]

منقوص واحد

- * Binary ⟶ 1's Compl
- * Decimal ⟶ 9's Compl
- * Hexa ⟶ 15's Compl
- * octal ⟶ 7's Compl

if MSB = 0 ⟶ +ve

if MSB = 1 ⟶ −ve

Scanned with CamScanner

*Ex:- 1) $(101010)_{1's}$?

- 1st way:- $\underset{-32\ \cancel{16}\ 8\ \cancel{4}\ 2\ \cancel{1}}{101010} = -32 + 8 + \cancel{8}2 = -22 = \overset{\text{Add }(+1)}{-21}$ ✓

- 2nd way:-

$$\begin{array}{r} 101010 \\ -1\ * \\ \hline \underset{\cancel{32}\ 16\ \cancel{8}\ 4\ \cancel{2}\ 1}{010101} \end{array} = 16 + 4 + 1 = 21$$

$= -21 \leftarrow$ Add -ve (signe)

- في الطريقة الأولى لما تكون الـ MSB سالبة بضيف واحد.

- " " الثانية بضرب بسالب وبجمع الأوزان وبالنهاية بعكس الإشارة.

2) $(11101)_{1's}$?

[1] $\underset{-16\ 8\ 4\ \cancel{2}\ 1}{11101} = -3 \overset{\text{MSB = -ve}}{(+1)} = \boxed{-2}$ ✓

[2]

$$\begin{array}{r} 11101 \\ -1\ * \\ \hline \underset{\cancel{16}\ \cancel{8}\ \cancel{4}\ 2\ 1}{00010} \end{array} = 2 * -1 = \boxed{-2}$$ ✓

3) $(1111)_{1's}$?

[1] $\underset{-8\ 4\ 2\ 1}{1111} = -1 \overset{\text{MSB = -ve}}{(+1)} = \boxed{-0}$

[2]

$$\begin{array}{r} 1111 \\ -1\ * \\ \hline 0000 \end{array} = 0 * -1 = \boxed{-0}$$

* for $(0000)_{1's} = \boxed{+0}$

4) find the Range if n = 10 , 1's complement ?

☒ from this Table :

n	2	3	Rule
Min	$(10)_{1's} = -1$	$(100)_{1's} = -3$	$-(2^{(n-1)} - 1)$
Max	$(01)_{1's} = +1$	$(011)_{1's} = +3$	$2^{n-1} - 1$
# of values	4	8	$r^n = 2^n$
# of zeros	2	2	2

للتوضيح

1) 0 1 1 $= \boxed{+3}$ ما بضيف واحد لأنه

4 2 1 ← MSB +ve

2) 0 1 1

$\quad\;\; -1 \quad *$

$= -4 \; (+1) = -3 * -1 = \boxed{+3}$

MSB = -ve

1 0 0

-4 ~~2~~ ~~1~~

لكي استخدم

طريقة (2)

ضربنا بـ (-1)

* Solu :- Range = Min → Max $= [-(2^9 - 1) \rightarrow (2^9 - 1)]$

$= [-511 \rightarrow +511]$ ✓

* Ex :- 1) r = 10 , n = 2 , B = 55 , find -B ?

Rule : $[B + (-B) = r^n - 1]$

$55 + -B = 10^2 - 1$

$-B = 10^2 - 1 - 55 = 100 - 56 = \boxed{+44}$ ✓

2) r = 2 , n = 4 , B = 9 , -B ?

$9 + -B = 2^4 - 1$

$-B = 15 - 9 = \boxed{+6}$ ✓

3) Find (-77) using 1's Compl & Min # of Bits ?

Solve for +77 : 64 32 16 8 4 2 1

For +ve ← 0 1 0 0 1 1 0 1

Signe ⑬ ⑤

+77 = 010011011

-1 *

(10110010) = -77

*check : 10110010

-128 64 32 16 8 4 2 1

= -128 +32 +16 +2

= -78 ⇒ Add 1:- -78 (+1) = $\boxed{-77}$

(because MSB -ve)

⊛ Radix Complement, [R.c], [2's Comp] :-

Binary → 2's comp

Decimal → 10's comp

octal → 8's comp

Hexa → 16's comp

مش منقوص

if MSB = 0 → +ve

if MSB = 1 → -ve

⚠ Ex:- 1) $(111011)_{2's}$?

1 st way : 111011 = $\boxed{-5}$ ← ما بزيد واحد

-32 16 8 $\cancel{4}$ 2 1

2 nd way : 111011 *

-1

000100

+1 Add one

000101 = 5 * -1 = $\boxed{-5}$

$\cancel{32}$ $\cancel{16}$ 8 4 $\cancel{2}$ 1

☒ حسب ما سبق :-

- في الطريقة الأولى فقط نجمع الأوزان دون إضافة واحد.

- في الطريقة الثانية بضرب بسالب وبضيف واحد وبجمع الأوزان ثم برجع بضرب في سالب واحد.

2) $(011101)_{2's}$?

$\boxed{1}$ $011101 = +29$ ✓

$\cancel{32}\ 16\ 8\ 4\ \cancel{2}\ 1$

$\boxed{2}$ $011101 \ * \ -1$

100010

$+\ 1$

$100011 = -29 * -1 = +29$ ✓

$-32\ \cancel{16}\ \cancel{8}\ \cancel{4}\ 2\ 1$

3) $(111011)_{2's}$?

$\boxed{1}$ $111011 = \boxed{-5}$

$-32\ 16\ 8\ \cancel{4}\ 2\ 1$

$\boxed{2}$ $111011 \ * \ -1$

000100

$+\ 1$

$000101 = +5 * -1 = \boxed{-5}$

$4\ \cancel{2}\ 1$

$\boxed{3}$ 3rd way [on the fly] :-

← إمشي من اليمين إلى اليسار ونزّل كل الأرقام مثل ما هية بدون تعديل، لحتى تشوف أول (1)، نزله مثل ما هو وإقلب كل الأرقام الي بعده،

Sol:-

هي أول 1 شفته

$(1\ 1\ 1\ 0\ 1\ \textcircled{1})_{2's}$

$= (0\ 0\ 0\ 1\ 0\ 1) \qquad = 4+1 = 5$

$\cancel{32}\ \cancel{16}\ \cancel{8}\ 4\ \cancel{2}\ 1$

$5 * -1 = \boxed{-5}$ ✓

(لا تنسى بالنهاية تضرب في سالب واحد)

4) $(111011)_{2's}$?

on the fly : $(000101) = +5 * -1 = \boxed{-5}$ ✓

4 2 1

5) $(101100)_{2's}$?

$010100 = +20 * -1 = -20$

32 16 8 4 2 1

6) $(0000)_{2's}$?

$0000 = \boxed{0}$

7) $(1000)_{2's}$?

$\boxed{3}$ $1000 = -8 * -1 = \boxed{8}$

-8 4 2 1

$\boxed{2}$ 1000

-1

0111

+1

$1000 = -8 * -1 = \boxed{8}$

-8 4 2 1

$\boxed{1}$ $1000 = -8$ ∝ error !!!

-8 4 2 1

n	2	3	Rule
min	$(10)_{2's} = -2$	$(100)_{2's} = -4$	$-(2^{n-1})$
max	$(01)_{2's} = +1$	$(011)_{2's} = +3$	$(2^{n-1}-1)$
# of values	4	8	2^n
# of zeros	1	1	1

Table :- [2's complement]

← في مشكلة في الطريقة الأولى تبعت ال (2's) بنشوفها لما تكون قيمة الرقم بال (Decimal) أحد قوى ال (2):

[... + 32 + 16 + 8 + 4 + 2 + 1]

لأن ال max عند هذه الأرقام يكون أقل منهم بواحد حسب القانون $(2^{n-1} - 1)$

* For 1000 → n = 4 → $2^{(4)-1} - 1 = \boxed{+7}$

So +8 is out of Range

So 1st way not Valid. #

* بشكل عام لمن تشوف الرقم بال Binary ع شكل واحد وع يمينه أصفار:

[..., 10000, 1000, 100, 10]

ع طول ما بتنفع أول طريقة

⚠*Ex:- if n = 4, Range = ? , 2's ?

Range = min → max = $[-(2^{4-1}) \rightarrow (2^{4-1} - 1)]$

$= [-8 \rightarrow +7]$ ✓

⚠*Ex:- Find the equivalent of $(1110101)_{1's}$?

$\boxed{1}$ $(111\ 111\ 11\ 0101)_{1's}$

$\boxed{2}$ $(111011)_{2's}$

$\boxed{3}$ $(-11)_{10}$

$\boxed{4}$ $(-A)_{16}$

$\boxed{5}$ $(100\ 00\ 1010)_{SM}$

Solve each part for alone:

1) $(111\ 111\ 11\ 0101)_{1's}$

-1 *

$\overline{000\ 000\ 00\ 1010} = 10 * -1 = -10$

8 4 2 1

2) $(111\ 011)_{2's}$

000 101 = 5 * -1 = $\boxed{-5}$

421

3) -11

4) $(-A)_{16}$ = -10

5) $(100\ 001010)_{SM}$ = -10

-ve 8421

- from Qustion: $(111010l)_{1's}$ = -10

-1 *

0001010 = 10 * -1

~~42~~

8421

∴ Ams is ① & ④ & ⑤

Ex:- 1) r = 10 , n = 2 , B = 55 , -B = ?

Rule : [$B + (-B) = r^n$] , for 2's

$55 + -B = 10^2$

$-B = 100 - 55 = 45$

2) r = 2, n = 4, B = +9 , -B = ?

$-9 + -B = 2^n$

$-B = 16 - 9 = +7$

⊛ Summary :- For n = 3 :

Binary code	unsigned	SM	1's comp	2's comp
000	0	+0	+0	0
001	1	+1	+1	+1
010	2	+2	+2	+2
011	3	+3	+3	+3
100	4	-0	-3	-4
101	5	-1	-2	-3
110	6	-2	-1	-2
111	7	-3	-0	-1

⚠ Ex:- find -33, 2's, n=8 ?

+33　　32 16 8 4 2 1

(1 0 0 0 0 1) = 33

+33, n=8, 00100001 = +33

(11011111) = -33 ✓

☒ Addition & Subtraction: [Finite precision]

1) Fixing Number:

1 unsigned: add zeros after MSB (in left side).

ex: n=8, $5 = (101)_2 \rightarrow (00000101)$

2 SM: add zeros befor MSB.

ex: n=8, $10011 \rightarrow (1000\ 0011)_{SM}$

3 1's & 2's: extend (repeat) MSB

ex: $(1101)_{1's}$, $n=8 \rightarrow (1111\ 101)_{1's}$

$(00011)_{2's}$, $n=8 \rightarrow (00000011)_{2's}$

* unsigned, addition:

IF Cout = 1, ov is true, result is wrong.

IF Cout = 0, ov is False, result is true.

Cin →

ex:- ① 1 1 1 1

$$\begin{array}{r} 1101 \\ + \\ \hline \end{array}$$

Cout → ① $\underbrace{1100}$ → الناتج ده

⊛ 1's complement addition subtraction :-

IF Cin ≠ Cout → OV is true → result is wrong.
IF Cin = Cout = 1 → Discard ([illegible]) Cout & add (1).
IF Cin = Cout = 0 → result is correct.

⊛ Ex:-

① $(25)_{10} - (25)_{10} = ?$, 1's , n=8

16 8 4 2 1
25 + (-25)   000(1 1 0 0 1) = 25

Cin
9 1
⓪0 0 0 1 1 0 0 1 → +25
+
1 1 1 0 0 1 1 0 → -25

⓪ (1 1 1 1 1 1 1 1)$_{1's}$ ✓ [result in 1's]
Cout
check (*-1) → (000 000 00)
(-0) ← (*-1) +0 ↵

② $(115)_{10} - (39)_{10} = ?$, 1's , n=8

115 + (-39)
64 32 16 8 4 2 1
+115 → 0 1 1 1 0 0 1 1

51 19 3 1 0
+39 → 0 0 1 0 0 1 1 1

Cin ←
①
0 1 1 1 0 0 1 1 → +115
+
1 1 0 1 1 0 0 0 → -32 (by 1's)
1 Discard
1 1
~~①~~ 0 1 0 0 1 0 1 1
Cout
1 + add 1

(0 1 0 0 1 1 0 0)$_{1's}$ ✓ = 76 ✓

* 2's complement addition :

IF Cin ≠ Cout → ov is true → result wrong.
IF Cin = Cout = 1 → Discard Cout.
IF Cin = Cout = 0 → result is correct.

△ Ex:- ① 2's , n=5 , 8 + 12 = ؟

Cin ① 8 4 2 1
+8 → 0 1 0 0 0
+12 → 0 1 1 0 0 +
Cout ⓪ 1 0 1 0 0 ∝ [result in 2's]

So result is wrong [out of Range]

② 2's , n = 5, 12 - 8 ؟

12 + (-8) , +12 → 01100 (sign)
+8 → 01000

Cin ①
0 1 1 0 0
-8 → 0 1 0 0 0 + → by (2's) check 2-8 = 4
~~①~~ 0 0 1 0 0 → correct result ✓ = ④
Cout

③ 2's , n = 4, 7 -(-8) ؟

7 + 8 , Range = min → max
$= -(2^{4-1}) \rightarrow 2^{4-1} - 1 = -2^3 \rightarrow 2^3 - 1$
* لما تشوف رقم من قوى العدد 2 ع طول
اتأكد من الـ Range
= (-8 → 7) , But 7 + 8 = 15
(لازم كمان Bit لأمثل +8)

So out of Range [can't Represent]

④ 14-6 , n=8 , 2's ?

+14 → 00001110

+6 → 00000110 2's

11111010

~~1~~ 00001000 = 8 ✓

check : 11111000 = -8 * -1 = $\boxed{8}$ ✓

⊛ unsigned using 1's OR 2's :

[1] using 1's compl :

IF Cout =1 → Discard Cout & add (1).

IF Cout =0 → find 1's compl (* -1), Then place (-) in Result.

⚠Ex:- ① unsigned , n = 4, 13-5 using 1's ?

8 4 2 1

13 → 1 1 0 1

5 → 0 1 0 1

→ unsigned.

13 → 1 1 0 1

+

5 → 1 0 1 0 → 1's

Discard ~~1~~ Cout 0 1 1 1 Add 1

+1

(1 0 0 0) → final result in unsigned.

② unsinged, n=4, 5-13, using 1's?

5 → 0101

13 → 1101

5 → 0101 +

-13 → 0010 → 1's

⓪ 0111

Cout

*(-1) find 1's

(1000) → -(1000) ✓

place (-)

[2] using 2's compl:

IF Cout = 1 → Discard Cout.

IF Cout = 0 → find 2's compl (* -1), then place (-) in the result.

⚠Ex: ① n=4, 13-5, unsigned, 2's?

8 4 2 1

13 → 1101 → unsinged

5 → 0101

13 → 1101

5 → 1011 + → 2's (using on the fly)

×① 1000 → final result in unsigned.

Cout

② n=4, 5-13, unsigned, 2's?

8 4 2 1

5 → 0101 → unsigned.

13 → 1101

5 → 0101 +

-13 → 0011 → 2's

⓪ 1000 → -(1000) ✓

Cout 2's

☒ Binary Codes types:

1) Character coding [ASCII]
2) Gray Code
3) Decimal coding, [DMC]

*ASCII : American standerd code for information interchange.

عدد الـ Bits في نظام ascii هو 7 بينما الـ Bayt هو 8 Bits لذا يتبقى Digit فارغة؛

ex:

	1	0	0	0	0	0	1
8	7	6	5	4	3	2	1

*الحل الأول للمشكلة: Set MSB = 0

*الحل الثاني: Use it in even or odd parity error detection cooding

i.e: for odd detection as example

Sender ⟶ reciver

11000001   11000001

number of 1s n=3   n=3

∴ odd   ∴ odd

* في الـ even: نملأ الـ Bit الفارغة بحيث نجعل عدد الـ (1) زوجي.

* في الـ odd: نملأ الـ Bit الفارغة بحيث نجعل عدد الـ (1) فردي.

مشكلة أخرى وهي عدم القدرة على معرفة مكان الخطأ.

[Error recovery can not occure and can only detect odd number of errors]

*الحل: ١) إعادة إرسال الرسالة

٢) استخدام خوارزميات لتصحيح الخطأ لاستعادة الرسالة الأصلية.

☒ letters value in Ascii:

A = $(65)_{10}$ = $(41)_{16}$	a = $(97)_{10}$ = $(61)_{16}$
B = $(66)_{10}$	b = (98)
C = $(67)_{10}$	c = (99)
⋮ ⋮	⋮ ⋮
Z = = $(90)_{10}$	z = (122) #

Gray code:

	Binary	Gray
0	0 0 0	0 0 0
1	0 0 1	0 0 1
2	0 1 0	0 1 1
3	0 1 1	0 1 0
4	1 0 0	
5	1 0 1	
6	1 1 0	
7	1 1 1	

الفكرة أخلي الاختلاف بين أي سطرين متتالين فقط بـ Bit واحدة

* Convert:

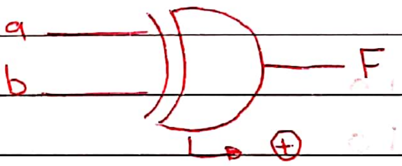
* Binary → Gray

① Copy MSB

② $G(i) = B(i) \oplus B(i+1)$

where ⊕ : exclusiv OR

[XOR].



	a	b	F
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

متشابهات = 0 → XoR gate

مختلفات = 1

⚠Ex:-

B = 11011 , G = ?

i = 4 3 2 1 0

B = 1 1 0 1 1

G = 1 0 1 1 0 = (10110) ✓

→ $G(4) = B(4) = 1$ (MSB)

→ $G(3) = B(3) \oplus B(4) = 0$

→ $G(2) = B(2) \oplus B(3) = 1$

→ $G(1) = B(1) \oplus B(2) = 1$

→ $G(0) = B(0) \oplus B(1) = 0$

* Gray → Binary :

① Copy MSB

② $B(i) = G(i) \oplus B(i+1)$

⚠Ex:-

G = 10110

B = ?

i = 4 3 2 1 0

G = 1 0 1 1 0

B = 1-1-0-1-1 = (11011)

$B(3) = G(3) \oplus B(4) = 1$

⊛ Binary Coded Decimal [BCD] :

Converting Between BCD and Decimal .

* أكبر رقم BCD بقدر أحوله إلى Decimal هو (1001) ويساوي 9 (لأنّ ال 9
أكبر رقم عشري مكون من منزلة واحدة) .

⚠ From Decimal to BCD :

Devide each Decimal Digit to 4 Binary Bits .

⚠ From BCD to Decimal :

each 4 Binary Bits must equal to 1 Decimal Digit .

⚠ EX:- ① $(128)_{10} \rightarrow$ BCD ?

8421 8421 8421

(0001 0010 1000) ✓

② $(9102)_{10} \rightarrow$ BCD ?

8421 8421 8421 8421

(1001 0001 0000 0010) ✓

③ $(\underbrace{1101}_{8421}\ 1000)_{BCD} \rightarrow$ Decimal ?

α (11) = Can't represent [out of Range , 11>9]

④ $(\underbrace{1001}_{8421}\ \underbrace{0011}_{8421}\ \underbrace{0111}_{8421})_{BCD} \rightarrow$ Decimal ?

9 3 7 = $(937)_{10}$ ✓

* يتم التخزين بنظام الـ BCD في جهاز الحاسوب تحديدًا في الـ (Memory / Ram) بطريقتين :

① packed : مضغوط

كل سطر في الذاكرة يحتوي على 8 Bits ويخزن الرقم على شكل (two of 4 Binary Bits)

② unpacked : غير مضغوط

كل سطر في الذاكرة يحتوي على 8 Bits وتخزن الرقم على شكل (only one of 4 Binary Bits)

Ex:- $(1\ 2\ 8)_{10} \rightarrow$ BCD

8421 8421 8421

$(0001\ 0010\ 1000)_{BCD}$ ✓

MSB LSB

1) packed method:

طريقة كتابة الرقم داخل الـ Memory :
من الـ LSB إلى الـ MSB ومن السطر الأسفل إلى السطر الأعلى

Memory

0000 0001
0010 1000

2) unpacked method:

0000 0001
0000 0010
0000 1000

The Hashemite University
Computer Engineering Department
Digital Logic (110408220)
Dr. Khalil Ahmad Yousef
HW1

Q1-2) What is the exact number of bytes in a system that contains (a) 32K byte, (b) 64M bytes, and (c) 6.4G byte ?

Solution:

a) $= 32 * 2^{10} = 32 * 1024 = 32{,}768$

b) $= 64 * 2^{20} = 64 * 1024 * 1024 = 67{,}108{,}864$

c) $= 6.4 * 2^{30} = 6871{,}947{,}674$

Q1-3) What is the largest binary number that can be expressed with 12 bits? What is the equivalent decimal and hexadecimal ?

Solution:

1,024 256

12-Bit Binary : $(1111\ 1111\ 1111)_2 = (4095)_{10}$

2,048 512 /64 32 16 8 4 2 1

128

$\rightarrow \quad \underset{F}{\underset{8421}{\underline{1111}}} \quad \underset{F}{\underset{8421}{\underline{1111}}} \quad \underset{F}{\underset{8421}{\underline{1111}}} = (FFF)_{16}$

Q1-4) Convert the following numbers with the indicated bases to decimal: $(4310)_5$, and $(198)_{12}$?

Solution:

a) $(\overset{3\,2\,1\,0}{4310})_5 = 4*5^3 + 3*5^2 + 1*5^1 = (580)_{10}$

b) $(\overset{2\,1\,0}{198})_{12} = 1*12^2 + 9*12^1 + 8*12^0 = (260)_{10}$

Q1-7) Express The following numbers in decimal: $(10110.0101)_2$, $(16.5)_{16}$, and $(26.24)_8$.

Solution:

a) 10110.0101 $= 16+4+2+\frac{1}{4}+\frac{1}{16} = (22.3125)_{10}$

16 8 4 2 1 $\frac{1}{2}\,\frac{1}{4}\,\frac{1}{8}\,\frac{1}{16}$

b) $(\overset{1}{1}\overset{0}{6}.\overset{-1}{5})_{16} = 1*16^1 + 6*16^0 + 5*16^{-1} = (22.3125)_{10}$

c) $(\overset{1}{2}\overset{0}{6}.\overset{-1}{2}\overset{-2}{4})_8 = 2*8^1 + 6*8^0 + 2*8^{-1} + 4*8^{-2} = (22.3125)_{10}$

Q1-8) Convert the following binary numbers to hexadecimal and to decimal:
(a) 1.11010, (b) 1110.10 , Explain Why the decimal answer in (b) is 8 times that of (a).

Solution:

a) $\underbrace{0001}_{8421}.\underbrace{1101}_{8421}\underbrace{0000}_{8421}$

1 , D 0

$= (\overset{0}{1}.\overset{-1}{D})_{16}$

$= (1.8125)_{10}$

b) $\underbrace{1110}_{8421}.\underbrace{1000}_{8}$

$= (\overset{0}{E}.\overset{-1}{8})_{16}$

$= 14*16^0 + \frac{8}{16}$

$= (14.5)_{10}$

Reason :

(b) 1110,10 is the same as (a) 1,11010 but shifted to the left by 3-places.

Q1-9) Convert the hexadecimal number (68BE) to binary and then from binary convert it to octal?

Solution:

6 8 B E

8421 8421 8421 8421

$(\underbrace{000}_{421}\underbrace{110}_{421}\underbrace{100}_{421}\underbrace{010}_{421}\underbrace{111}_{421}\underbrace{110}_{421})_2$

0 6 4 2 7 6

$= (64276)_8$

Q1-10) Convert the decimal number 345 to binary in two ways: (a) convert directly to binary; (b) convert first to hexadecimal , then from hexadecimal to binary. Which method is faster?

Solution:

		256	128	64	32	16	8	4	2	1	
a) $(345)_{10}$	=	(1	0	1	0	1	1	0	0	1$)_2$	by on the fly

b) $(\overset{2}{3}\overset{1}{4}\overset{0}{5})_{10} \Rightarrow$

$\frac{345}{16} = 21$ 9 LSB

$\frac{21}{16} = 1$ 5

$\frac{1}{16} = 0$ 1 MSB

$= (159)_{16}$

8421 8421 8421

0001 0101 1001

$= (101011001)_2$

159

If we use repeated division for both methods, Then the 2nd method is faster; لأنه تكرار القسمة في ال hexa أقل بكثير منها في ال Binra .

Q1-11) Do the following conversion problems:

(a) Convert decimal 34.4375 to binary.

(b) Calculate the binary equivalent of 1/3 out to 8 places. Then convert from binary to decimal. How close is the result to 1/3?

(c) Convert the binary result in (b) into hexadecimal. Then convert the result to decimal. Is the answer the same?

Solution:

a) 34,4375

34 /2 = 17 | 0 LSB

17/2 = 8 | 1

8 /2 = 4 | 0

4 /2 = 2 | 0

2 /2 = 1 | 0

1 /2 = 0 | 1 , MSB

[redacted]

,4375 * 2 = 0, 875 (MSB)

,875 * 2 = 1, 75

,75 * 2 = 1, 50

,5 * 2 = 1, 00 (LSB)

So = $(100010,0111)_2$

[redacted]

b & c

Behind the Page

Q1-16) Obtain the 1's and 2's complements of the following binary numbers:
(a) 11101010 (b) 01111110 (c) 00000001 (d) 10000000 (e) 00000000

start from the left & copy the Bits after 1st (1), inverte all the rest.

Solution:

*(-1)

Binary numbers	1's complement	2's complement
(a) 11101010	00010101	00010110
(b) 01111110	10000001	10000010
(c) 00000001	11111110	11111111
(d) 10000000	01111111	10000000
(e) 00000000	11111111	00000000

Q1-21) Convert the decimal 9126 to BCD code?

Solution:

9 1 2 6

8421 8421 8421 8421

1001 0001 0010 0110

→ BCD = (1001 0001 0010 0110)

*Q 1-11 :

b) $1/3 \simeq 0.333\ 333\ 3333.$

MSB

0,333 333 3333 * 2 = 0	,666 666 6667	1
0,666 666 6667 * 2 = 1	,333 333 333	2
0,333 333 333 * 2 = 0	,666 666 666	3
0,666 666 666 * 2 = 1	,333 333 332	4
0,333 333 332 * 2 = 0	,666 666 664	5
0,666 666 664 * 2 = 1	,333 333 328	6
0,333 333 328 * 2 = 0	,666 666 656	7
0,666 666 656 * 2 = 1	,333 333 312	8 - stop ✓

LSB

So $= (00101010 1)_2 = \frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \frac{1}{256} = (0.33203125)_{10}$

Very close to 1/3 ✓

c) $(0,0101\,0101)_2 = (0,\overset{-1}{5}\overset{-2}{5})_{16} = \frac{5}{16} + \frac{5}{16^2} = (0.33203125)_{10}$

the Same resut As (b) #

*Q (1-16):

a) $(11101010)_2$

→ $(11101010)_{1's} = (00010101)$

→ 2's comp:

$= (00010110)$ & so on....

*Q (1-21):

Already Done.

The Hashemite University
Computer Engineering Department
Digital Logic (110408220)
HW2

Q1.18.a) Perform subtraction on the given unsigned binary numbers using the 2's complement of the subtrahend. Where the result should be negative, find its 2's complement and affix a minus sign.

(a) 10011 - 10010
(b) 100010 - 100110
(c) 1001 - 110101
(d) 101000 - 10101

a) 10011
~~10010~~) 2's
01110
Cout ~~1~~ 00001 ✓
= 1

b) 100010
~~100110~~) 2's
011010
Cout 0 111100) 2's
-(000100)
= -4

c) 001001
~~110101~~) 2's
001011
Cout 0 010100) 2's
-(101100)
= -44

d) 101000
~~010101~~) 2's
101011
Cout ~~1~~ 010011 ✓
= 19

خلف **Q1.18.b)** Repeat the previous question by considering the given binary numbers as signed complement using the 2's complement.

خلف **Q1.20)** Convert decimal +49 and +29 to binary, using the signed-2's complement representation and enough digits to accommodate the numbers. Then perform the binary equivalent of (+29) + (-49), (-29) + (+49), and (-29) + (-49). Convert the answers back to decimal and verify that they are correct.

خلف **Q1.27)** Assign a binary code in some orderly manner to the 52 playing cards. Use the minimum number of bits.

(1- 18,6): , 2's Comp add / Sub

cin | check

① 1 1
a) 10011
10010 2's
Cout
⓪ 01110
00001
result ✓

10011 → 01101 = -13

10010 → 01110 = -14

-13 -(-14) = 1

cin ⓪
b) 100010
100110) 2's
cout 011010
⓪ 11111001 = -4

→ 100010 → 011110
= -30

→ 100110 → 011010
= -26
-30 - (-26) = -4

cin ①
c) 111001
cout 110101
⓪) 2's
001011
000100 = +4

111001 → 000111 = -7
110101 → 001011 = -11
-7 - (-11) = 4

d) ⓪ 101000
110101) 2's
⓪ 001011
110011 = -13

101000 → 011000 = -24
110101 → 001011 = - 11
-24 -(-11) = -13

(1-20):

+49 = 0110001 $\xrightarrow{2's}$ 1001111 , n=7
+29 = 0011101 $\xrightarrow{2's}$ 1100011 , n=7

① (+29)-(+49):
cin ⓪ 0011101
1001111 ✓
cout ⓪ 1101100 = -20

check
29 - 49 = -20

② -29 +49 :

①
1100001
0110011
0 0010100 ✓ =20

-29 +49 = 20 ✓

③ -29 -49 :

Cin ⓪
1100011
1001111
Cout ①0110010
Cout ≠ Cin

-29 -49 = -78

$\#^{max} = 63$

$\#^{Min} = -64$

(1-27):

الشدة فيها 13 ورقة كل وحدة إلها 4 أنواع

∴ 52 = 4*13 عددها كلها ، ببي أمثل كل بطاقة منال 52 جالBinary ، بأقل

n ممكنة

$\Rightarrow 2^n = \#$ of values $= 52$

$\therefore \log_2(52) = n = 5.7 \simeq 6$ التقريب لأكبر عدد صحيح

الآن نبدأ جالترتيب هكذا :

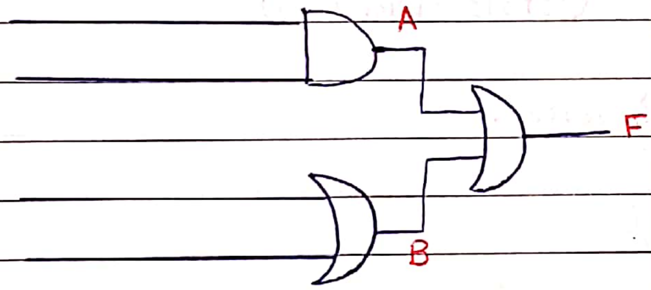
000 000 for ♡
000 001 for ◇
000 010 for ♧
111 111 52 حتى أصل إلى الرقم
لـ[من 0 إلى 51 = 52]

⊛ Chapter 2:

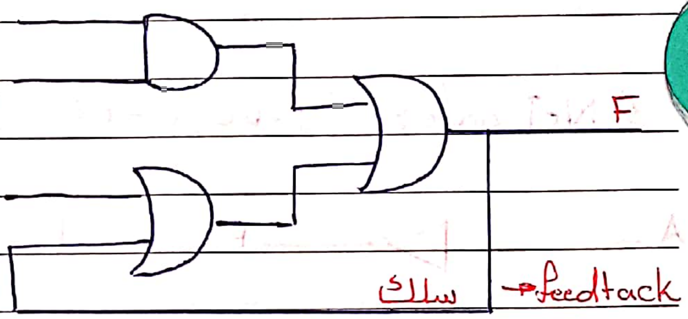
* Boolean Algebra & Logic gates

△* Combinational Logic: outputs depend on inputs only.

△* Sequential Logic: outputs depend on inputs and prevuos value.



Combinational



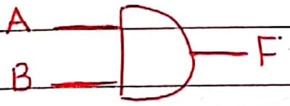
Sequntial

⊡ Logic gates:

[AND / OR / NOT / NAND / NOR / XOR / XNOR]

→ (3 - Basic gates)

1) AND gate:

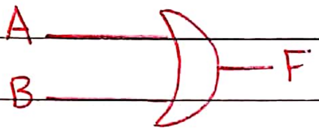


$F = A \circ B = AB$

A	B	F(A∘B)
0	0	0
0	1	0
1	0	0
1	1	1

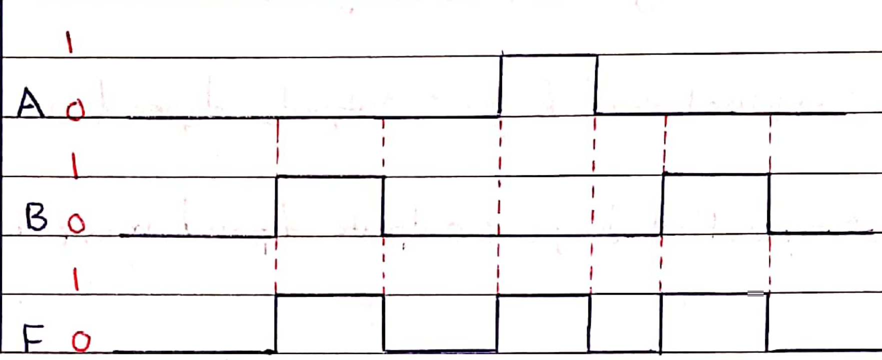
(State - Diagram)

2) OR gate:



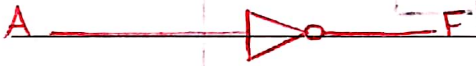
F = A+B

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



(State-Diagram)

3) NoT gate: (1→0, 0→1) = INverter.

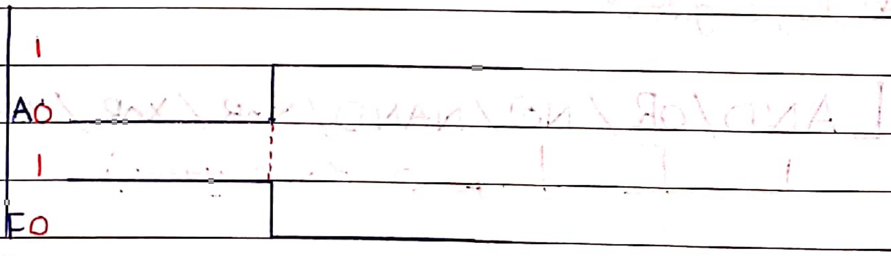


, F = $\bar{A}$

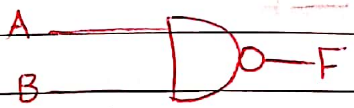

: Buffer (يقوي الإشارة)


: Buble (لا تأتي لوحدها)

A	F
0	1
1	0

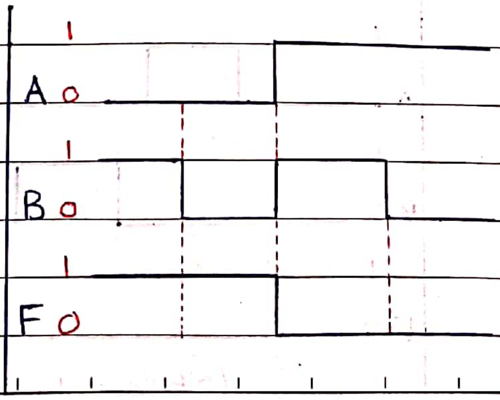


4) NAND: (Not AND)



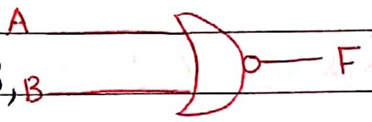
$F = \overline{(A \cdot B)} = \overline{AB}$

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

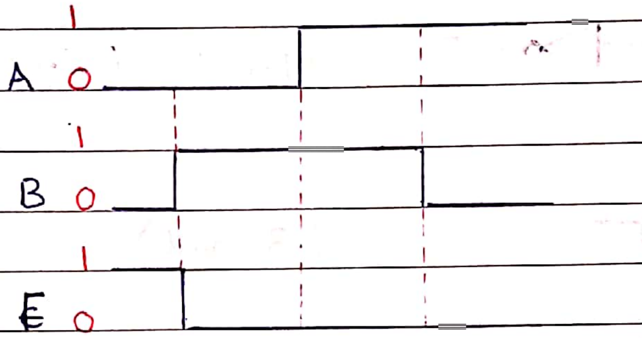


5) NoR :- (NoT oR)

$F = (\overline{A + B}),\ A \downarrow B,$

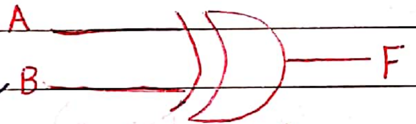


A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

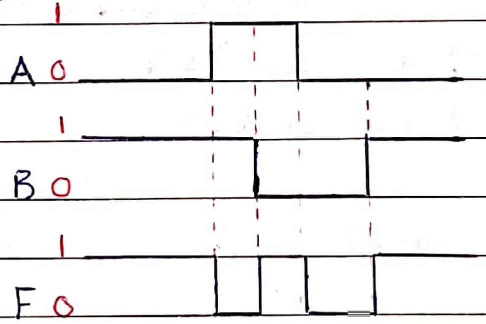


6) XoR :- (odd function)

$F = A \oplus B$,

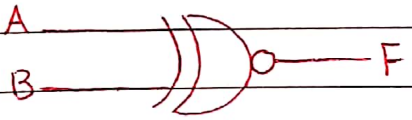


A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

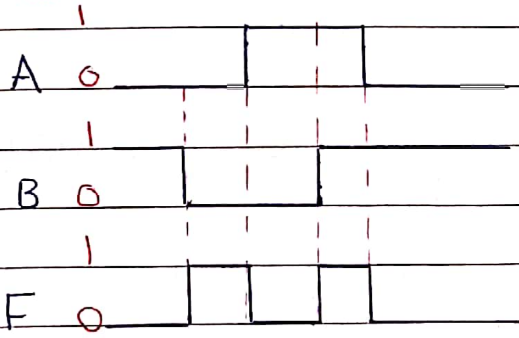


7) X NoR :- (even function)

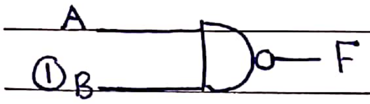
$F = (\overline{A \oplus B})$,



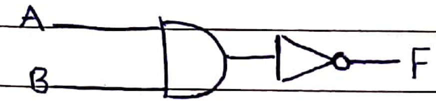
A	B	F
0	0	1
0	1	0
1	0	0
1	1	1



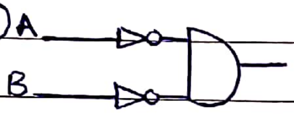
☒ Nots :

①


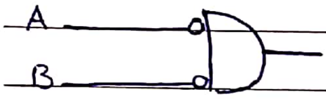
⇒ NAND, (1-gate)



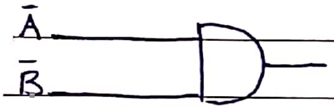
⇒ NAND, (2-gate)

②


, (3-gate)



, (3-gate)



, (1-gate)

ال inverter, (—▷o—) بعتبره بوابة كامل.

لكن ال pubble (○—) إذا كانت و * أمام ال gate (⊐D○—) ما بعتبرها بوابة

* خلف ال gate (⊐○D—) هون بعتبر كل فقاعة بوابة كامل.

⊛ Describing Logic Circuts (تمثيل الدوائر الكهربائية) :-

- Given : cct's
- Required : Boolean expresion.

⊛ priorities of performing :

① Braces (الأقواس)

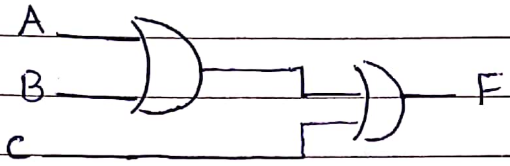
② Not (إبدأ من اليسار إلى اليمين)

③ AND (left to right)

④ OR (left to right)

*Ex:-

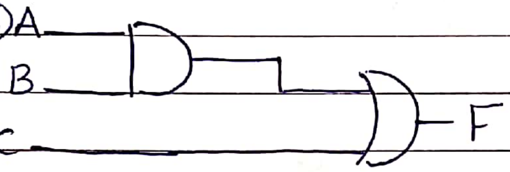
①



؟

$(A+B)+C = F$

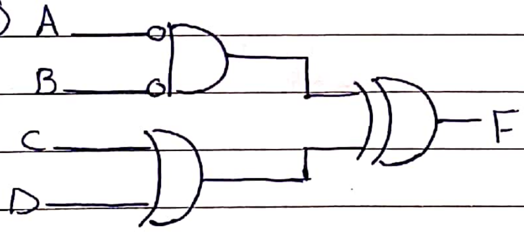
②



؟

$(A\cdot B)+C = F$

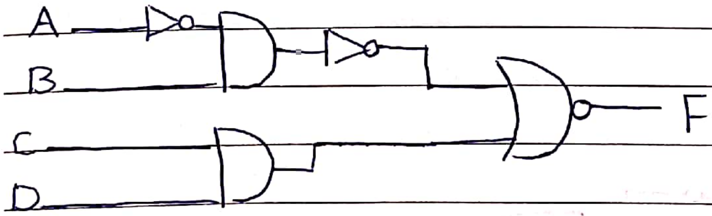
③



؟

$(\bar{A}\cdot\bar{B}) \oplus (C\cdot D) = F$

④



$\overline{\left[(\bar{A}\cdot B)+(C\cdot D)\right]} = F$

⊛ Evaluating Logic cct's :-

→ Given : Boolean function.

→ Required : cct's , value of (F).

⊛ priorities of performing :-

① Braces (الأقواس)

② Not → (بتكون أولاً الـ Single ~~Signal~~ ثم الأقواس)

③ AND

④ OR ماشي من اليسار لليمين

⚠ Ex:-

① $F = \bar{A} \cdot B \cdot C \cdot \overline{(\bar{A}+B)}$, $A=0$, $B=1$, $C=0$?

$F = \bar{0} \cdot 1 \cdot 0 \cdot \overline{(\bar{0}+1)}$ SingleNot ~~Single Not~~ و أقواس

left → right

$= 1 \cdot 1 \cdot 0 \cdot \overline{(1+1)}$ أقواس

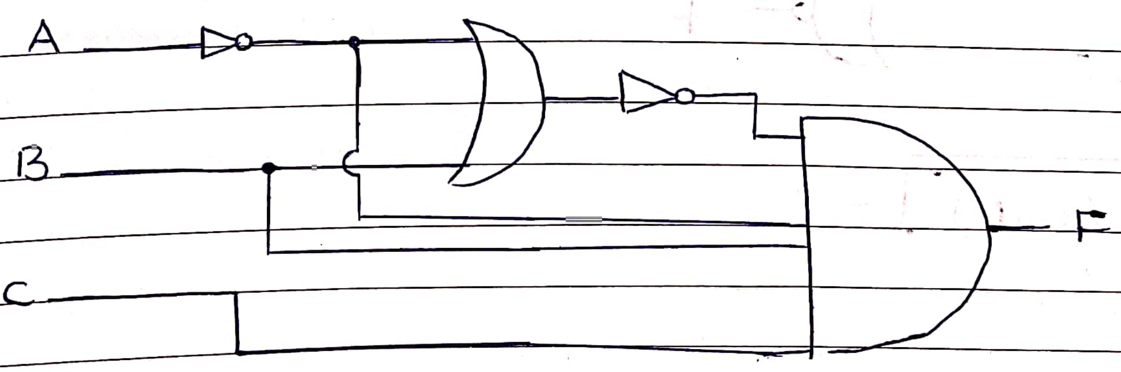
$= 1 \cdot 1 \cdot 0 \cdot (\bar{1})$ Single Not

$= 1 \cdot 1 \cdot 0 \cdot 0$ (left to right) AND كله

$= 1 \cdot 0 \cdot 0$

$= 0 \cdot 0 = \boxed{0}$ #

⇒ ~~plotting~~ plotting : $\bar{A}BC\,\overline{(\bar{A}+B)} = F$



② $F = AB + CD + \bar{A}B$, $(A=1, B=1, C=1)$?

$L \rightarrow R$

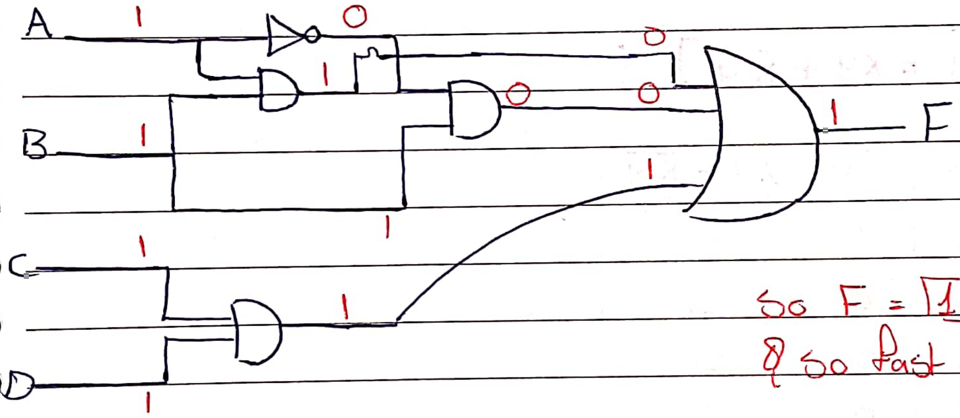
$F = 1\cdot1 + 1\cdot1 + \bar{1}\cdot1 = 1\cdot1 + 1\cdot1 + 0\cdot1$

$= 1 + 1\cdot1 + 0\cdot1 = 1 + 1 + 0$

$= 1 + 0 = \boxed{1}$

* والطريقة الأسهل والأسرع كل هاي الأسئلة هي الرسم أولاً ثم الحل من خلال الرسم

* plotting : $AB + CD + \bar{A}B = F$



So $F = \boxed{1}$ #

& so fast method.

$\boxed{*}$ provings :-

1) prove that $\bar{A}\bar{B} \neq \overline{A\cdot B}$?

using truth table :

A	B	$\bar{A}$	$\bar{B}$	$\bar{A}\bar{B}$	$\overline{AB}$
0	0	1	1	1	1
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	0

so $\bar{A}\bar{B} \neq \overline{AB}$ #

② prove that $[x + y = x\bar{y} + \bar{x}y]$?

x	y	$x \oplus y$	$x\bar{y}$	$\bar{x}y$	$x\bar{y} + \bar{x}y$
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	0	0	0

↳ So Done # correct.

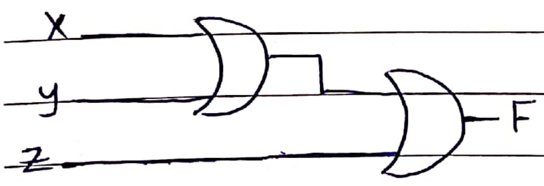
③ prove that $[\overline{x \oplus y} = xy + \bar{x}\bar{y}]$?

x	y	$x \oplus y$	xy	$\bar{x}\bar{y}$	$xy + \bar{x}\bar{y}$
0	0	1	0	1	1
0	1	0	0	0	0
1	0	0	0	0	0
1	1	1	1	0	1

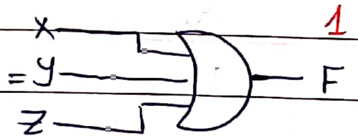
↳ So Done # correct.

⊛ Expansion of logic gates: (التشبيك)

⊛ Ex:- ① $F = x + y + z$?

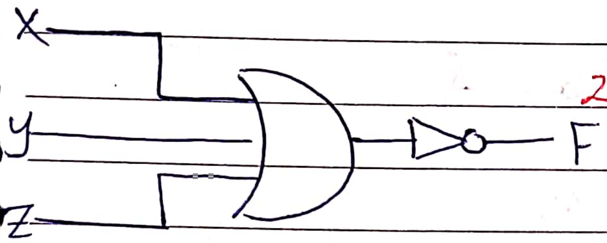


=



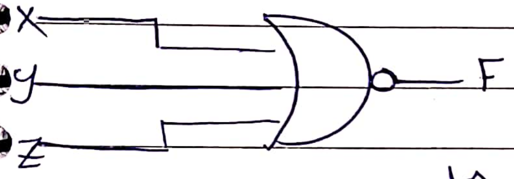
1 gate.

② $F = \overline{x+y+z}$ ؟



2-gate (OR & NOT)

=



1-gate (NoR)

③ $F = xyz \overset{?}{\underset{}{=}} \overline{\overline{xy}\ z}$ ؟ هل

xyz	x·y·z	$\overline{xy}$	$\overline{\overline{xy}\cdot z}$
0 000	0	1	1
1 001	0	1	0
2 010	0	1	1
3 011	0	1	0
4 100	0	1	1
5 101	0	1	0
6 110	0	0	1
7 111	1	0	0

3 - Variables

So

$2^{(\# \text{ of variables})} = \#$ of Cases

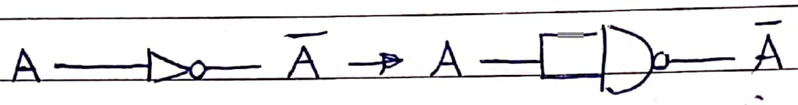
→ $2^3 = 8$ Cases.

→ so $xyz \neq \overline{xy}\cdot z$

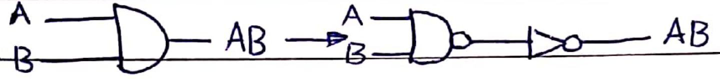
incorrect ✗.

1) NaND only:-

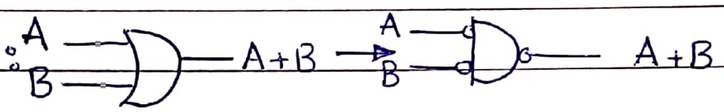
[1] Not:



[2] AND:



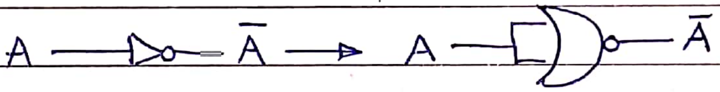
[3] oR:



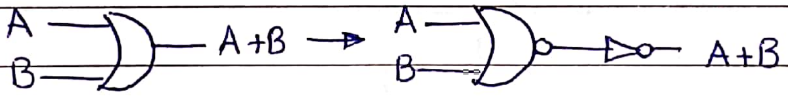
⇒ $\overline{\bar{A}\cdot\bar{B}} = \bar{\bar{A}}+\bar{\bar{B}} = A+B$

2) NoR only:-

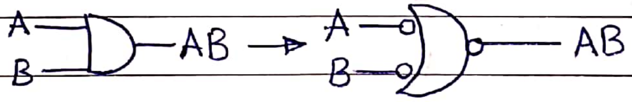
[1] Not:



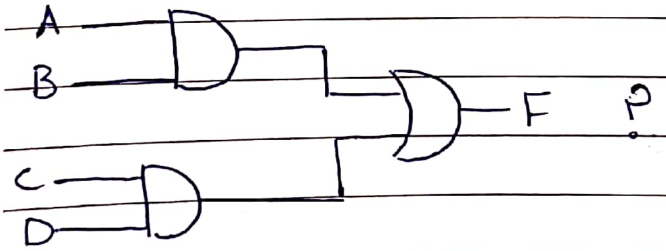
[2] oR:



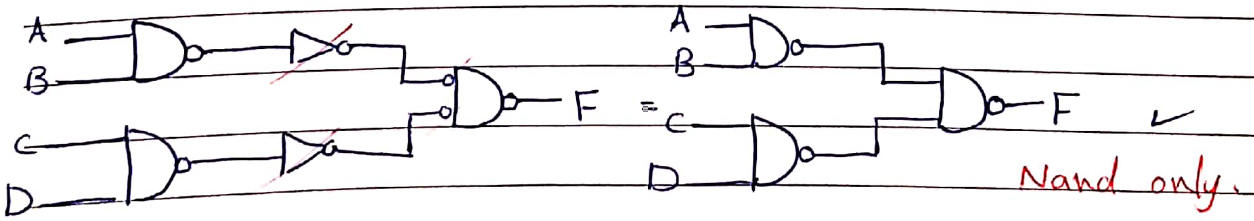
[3] ~~N~~ AND:



Ex:- ① invert to NaND only

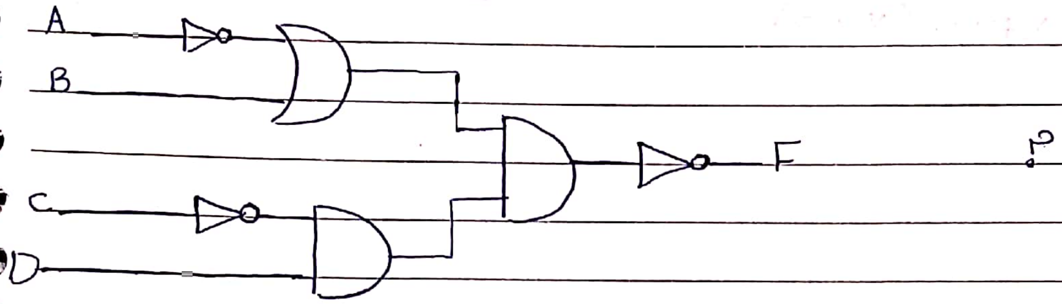


solu: → start from left to right step by step:



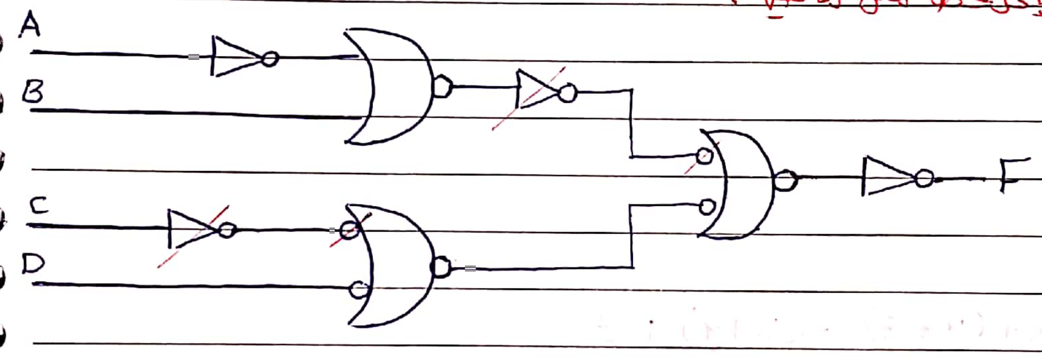
Nand only.

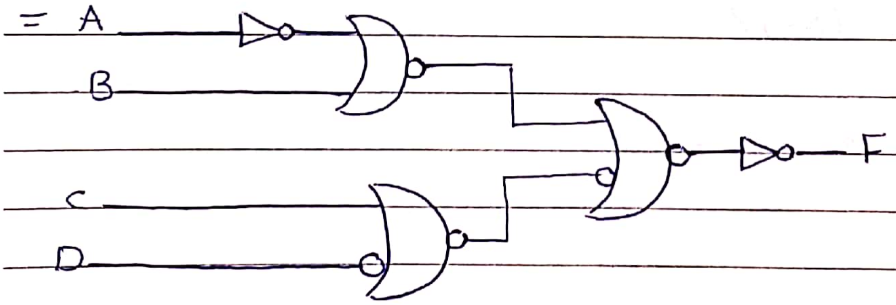
② Invert to NOR only



* Solu:

← لما تشوف (NOT) إتركها مثل ماهيا .





✓ NOR only.

* Rules *

→ $x + y \in$ Binary

→ $xy \in$ Binary

① $X + 0 = 0 + X = X$

② $X \cdot 1 = 1 \cdot X = X$

③ $X + y = y + X$

④ $xy = yX$

⑤ $X.(y+Z) = Xy + XZ$

⑥ $X + (yZ) = (X+y).(X+Z)$

⑦ $X + \overline{X} = 1$

⑧ $X\overline{X} = 0$

⑨ $X + X = X$

⑩ $XX = X$

⑪ $X + 1 = 1$

⑫ $X.0 = 0$

⑬ $\overline{\overline{X}} = X$

⑭ $(X+y+Z) = X+(y+Z) = (X+y)+Z$

⑮ $(X.y.Z) = X(yZ) = (xy)Z$

⑯ $\overline{X+y} = \overline{X}.\overline{y}$

⑰ $\overline{xy} = \overline{X} + \overline{y}$

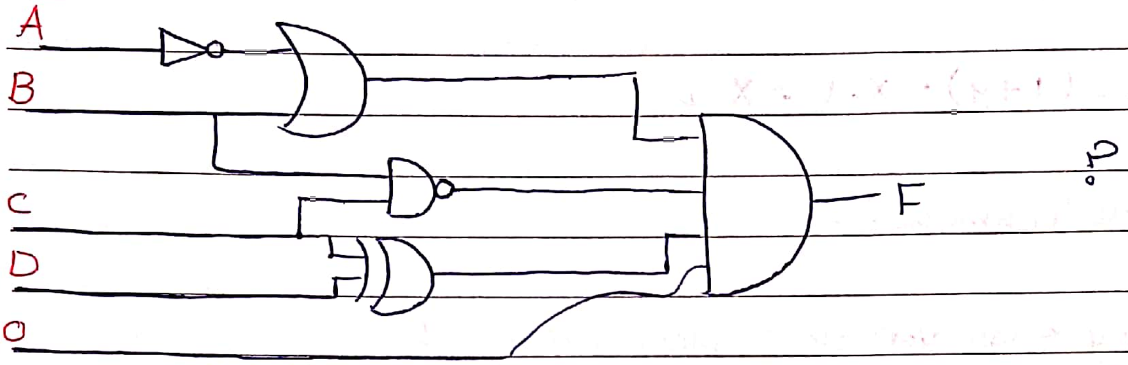
$\Big\}$ Demorgan's law.

⑱ $x + xy = x(1+y) = x.(1) = x$

⑲ $ABC + 0 = ABC$

⑳ $AB\overline{C} + \overline{AB\overline{C}} = 1$

⚠ Ex:-



*Solu:

(0 · any things) = Zero

So F = Zero ✓

⚠ Ex:- ① $F = \overline{ABCD}$, find Simplefy F ?

*Solu:

$F = \bar{A} + \bar{B} + \bar{C} + \bar{D}$ ✓

② $F = A + B\bar{C} + \bar{D}$, $\bar{F} = ?$

*Solu:

$\bar{F} = \overline{A + B\bar{C} + \bar{D}} = \bar{A} \cdot \overline{B\bar{C}} \cdot \bar{\bar{D}} = \bar{A}(\bar{B} + C)D$ ✓

③ prove, $X + X = X$? "using Rules"

سحب عامل مشترك

$= (X+X)\cdot 1 = (X+X)\cdot(X+\bar{X}) = X + (X\bar{X}) = X + 0 = X$ ✓

④ prove, $X + 1 = 1$?

$= (X+1)\cdot 1 = (X+1)\cdot(X+\bar{X}) = X + (1\cdot\bar{X}) = X + \bar{X} = ①$ ✓

⑤ prove, $x + xy = x$?

$x + xy = x \cdot (1+y) = x \cdot 1 = x$ ✓

⊛ Simplification techniques:-

- ▷ Literal: any single variable complemented or not
  $F(x,y,z) = xy + \bar{x}y + \bar{z}$ = 5 literals.
- ▷ Term: collection of literal that are inp to a single logic gate.

$F(x,y,z) = xy + \bar{x}y + z$ = 3 terms.

$F(x,y,z) = (x+y)(x+z)$ = 2 terms.

▷ Combining terms: $(xy + x\bar{y}) = x$;

*) $xy + x\bar{y} = x(y+\bar{y}) = x \cdot 1 = x$ ✓

*) $abcd + ab\overline{cd} = ab(cd + \overline{cd}) = ab$ ✓

*) $abcd + ab\bar{c}\bar{d} = ab(cd + \bar{c}\bar{d}) \neq ab$

▷ Eliminating terms: $(x + xy = x)$;

*) $x + xy = x(1+y) = x \cdot 1 = x$ ✓

*) $\bar{a}b + a\bar{b}\bar{c}d = \bar{a}b(1 + \bar{c}d) = \bar{a}b$ ✓

▷ Adding Redundant term: كرر ال (term) التاني لكل

$(a\bar{b}c + abc + \bar{a}bc) = (ac + bc)$;

*) $a\bar{b}c + abc + abc + \bar{a}bc$

$= ac(\bar{b}+b) + bc(a+\bar{a}) = ac + bc$ ✓

$= c(a+b)$ ✓

→ Eliminating Literal:

$(X + \bar{X}y = x+y), (\bar{X}+Xy = \bar{X}+y);$

$*)\ x+(\bar{x}y) = \cancel{(x+\bar{x})}\cdot(x+y) = (x+y)$ ✓

$*)\ \bar{x}+(xy) = \cancel{(\bar{x}+x)}\cdot(\bar{x}+y) = (\bar{x}+y)$ ✓

$*)\ \bar{A}B + \bar{A}\bar{B}\bar{C}\bar{D} + ABC\bar{D}$

$= \bar{A}(B+\cancel{\bar{B}}\bar{C}\bar{D}) + ABC\bar{D}$

$= \bar{A}(B+\bar{C}\bar{D}) + ABC\bar{D}$

$= \bar{A}B + \bar{A}\bar{C}\bar{D} + ABC\bar{D}$

$= B(\bar{A}+\cancel{A}C\bar{D}) + \bar{A}\bar{C}\bar{D}$

$= B(\bar{A}+C\bar{D}) + \bar{A}\bar{C}\bar{D}$

$= B\bar{A} + BC\bar{D} + \bar{A}\bar{C}\bar{D}$ ✓ as simplefy as posible.

Ex:- ① $XZ + Z(\bar{X}+Xy)$, simplify ?

$= XZ + Z(\bar{X}+y) = Z(\overset{①}{\cancel{X+\bar{X}}}+y)$

$= Z(1+y) = Z(1) = Z$ ✓

② $xy+\bar{x}Z+yZ$, simplify ?

$= xy+\bar{x}Z+yZ\cdot 1 = xy+\bar{x}Z+yZ(x+\bar{x})$

$= xy+\bar{x}Z+xyZ+\bar{x}yZ$

$= xy(1+Z)+\bar{x}Z(1+y) = xy+\bar{x}Z$ ✓

③ $\overline{(\bar{a}b+\bar{a}\bar{b}+\bar{b})}$, simplify ?

$= \overline{(\bar{a}(b+\bar{b})+\bar{b})} = \overline{\bar{a}+\bar{b}} = \bar{\bar{a}}\bar{\bar{b}} = ab$ ✓

④ prove that, $(x+y)(\bar{x}+Z) = xZ+\bar{x}y$?

$= \overset{0}{\cancel{x\bar{x}}} + xZ + y\bar{x} + \underline{yZ\cdot 1}$

لما تشوف term زيادة ، علطول فكر بال eliminat كله .

يتبع →

$= xz + \bar{x}y + yz(x + \bar{x})$

$= xz + \bar{x}y + xyz + \bar{x}yz$

$= xz(1+y) + \bar{x}y(1+z) = xz + \bar{x}y$ ✓

[*] Rules :- [XoR], odd

① $x \oplus y = x\bar{y} + \bar{x}y$

② $x \oplus 0 = x$

③ $x \oplus 1 = \bar{x}$

④ $x \oplus y = y \oplus x$

⑤ $(x \oplus y) \oplus z = x \oplus (y \oplus z)$

⑥ $x(y \oplus z) = xy \oplus xz$

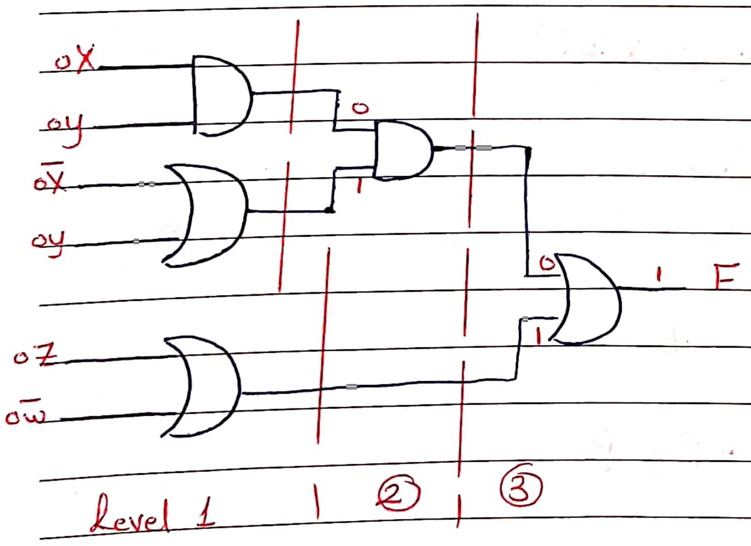
⑦ $\overline{(x \oplus y)} = xy + \bar{x}\bar{y}$

[*] Canonical representation of Bolean function:

1) non-standard: (mix of AND & OR) gates

without arrangement

$f = (xy)(\bar{x} + y) + (z + \bar{w})$

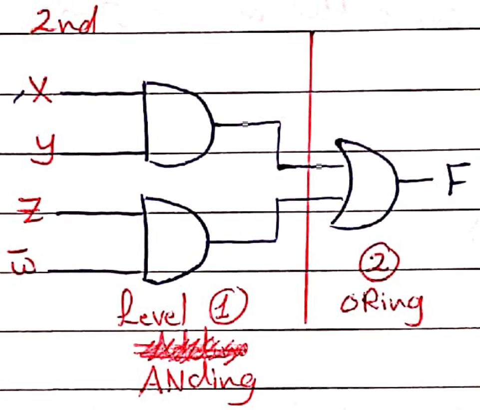


2) Standard : [2 Level]

① Sum of products (Sop) :

→ (Adding → oRing)
1st 2nd

F = xy + z$\bar{w}$



(*) Canonical : is special case from std sop, where each term should Contain all variables so called: Sum of Min term .

→ term : not Contained all variables .

→ Minterm : Contained all variables .

(*) Ex :- ① F = xyz + x$\bar{y}\bar{z}$, check std or not ?
& Canonical or not ?

$$F = \underbrace{xyz}_{AND} \underbrace{+}_{OR} \underbrace{x\bar{y}\bar{z}}_{AND} \equiv (sop)$$

So, yes std & since 1st & 2d term Containe all variables (x, y, z) So called Minterm

∴ yes Canonical

② $F = xy\bar{z} + \bar{x}y$, std ? , Canonical ?

yes std, but not Canonical.

Q: what is the number of possible Minterm for 2-variables ?

2-variables = xy
Base of Binary = 2
# of cases = $2^{(\#\text{ of variables})}$
$= 2^2 = 4$ so (4 minterm) ✓

Decimal # of Case	xy	Minterms	F(given)
m_0	00	$\bar{x}\bar{y}$	1
m_1	01	$\bar{x}y$	0
m_2	10	$x\bar{y}$	1
m_3	11	xy	1

* Minterm (Sop):-
[ANDing → oRing] ;

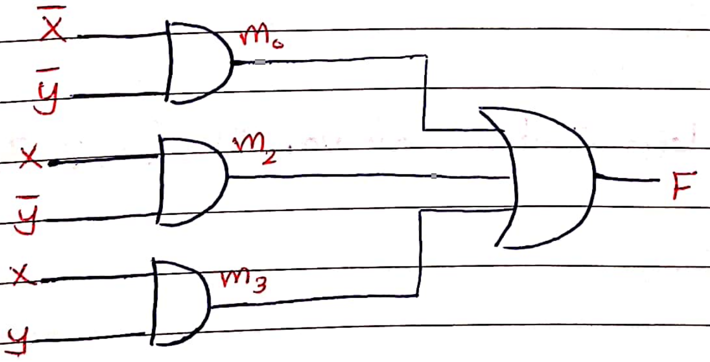
الصفر عليه Bar
الواحد ما عليه #

↓ minterm numbers

↳ For (Sop) we chose 1(s) only.

so $F = m_0 + m_2 + m_3 = \sum(0,2,3)$
$= \bar{x}\bar{y} + x\bar{y} + xy$

↳ 3-ways to represent F as a (Sop).



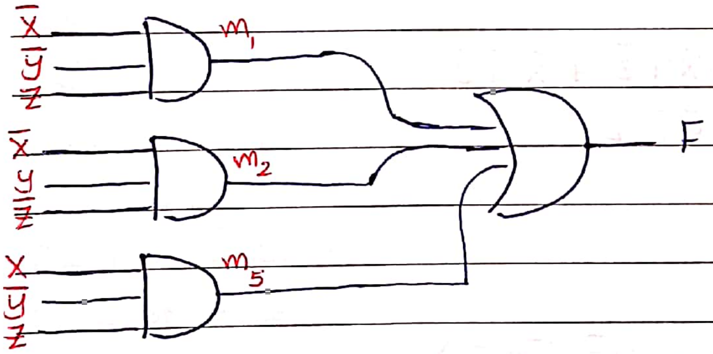
Q: Find F as a function of x, y, F ?

x y z	F (Given)	
0 0 0	0	m_0
0 0 1	1	$m_1 = \bar{x}\bar{y}z$
0 1 0	1	$m_2 = \bar{x}y\bar{z}$
0 1 1	0	m_3
1 0 0	0	m_4
1 0 1	1	$m_5 = x\bar{y}z$
1 1 0	0	m_6
1 1 1	0	m_7

$F = m_1 + m_2 + m_5$

$F = \Sigma(1, 2, 5)$

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



Q: $F = \Sigma(1, 2, 9, 10)$, Find F in term of variables ?

→ Look to the largest # within parentheses & find the equivnt in Binary:

⑧ 4 ② 1

$n=$

$(1010) \rightarrow$ so 4-B, Bits,

so 4-variables (x, y, z, w)

→ determine # of terms := 4 ; (1, 2, 9, 10)
$\quad$ 1 2 3 4

→ determine if sop or not?

$F = \Sigma(\ldots)$, so (sop) ✓

So ANDing then oRing.

$\overset{m_1}{\bar x \bar y \bar z w} + \overset{m_2}{\bar x \bar y z \bar w} + \overset{m_9}{x \bar y \bar z w} + \overset{m_{10}}{x \bar y z \bar w}$

$\cancel{8}\,\cancel{4}\,\cancel{2}\,1 \qquad \cancel{8}\,\cancel{4}\,2\,\cancel{1} \qquad \cancel{8}\,\cancel{4}\,\cancel{2}\,1 \qquad 8\,\cancel{4}\,2\,\cancel{1}$

0001 $\quad$ 0010 $\quad$ 1001 $\quad$ 1010

Q: $F = xy + \bar z$, convert to Canonical?

term to minterm

$xy \rightarrow xyz \rightarrow xy\cdot 1 = xy(z+\bar z)$

$\bar z \xrightarrow{to} xyz \rightarrow \bar z\cdot 1 = \bar z\,(x+\bar x)$

→ $xyz + xy\bar z$ ✓

→ $1\cdot(x\bar z) + (\bar x \bar z)\cdot 1 = (x\bar z)(y+\bar y) + (\bar x \bar z)(y+\bar y)$

$= xy\bar z + x\bar y\bar z + \bar x y \bar z + \bar x \bar y \bar z$ ✓

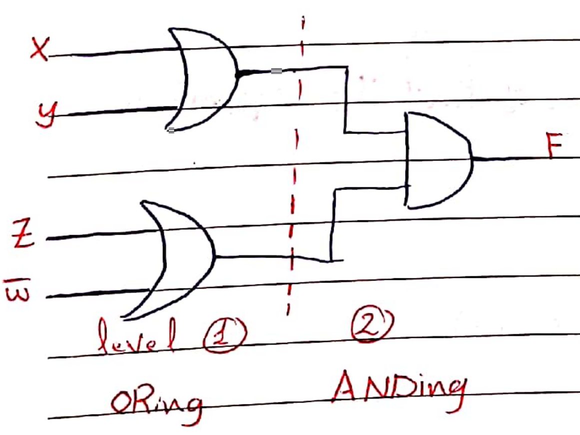
$\therefore F = xyz + xy\bar z + xy\bar z + x\bar y\bar z + \bar x y\bar z + \bar x\bar y\bar z$

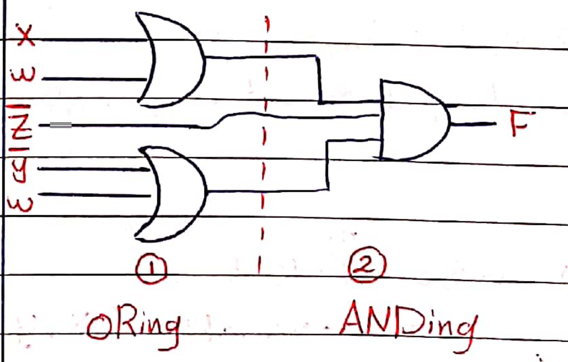
$= xyz + xy\bar z + x\bar y\bar z \ldots\ \bar x\bar y\bar z$ ✓

② product of sum (pos):

→ (oRing → ANDing),

$F = (x+y).(z+\bar w)$ $\qquad$ $F = (x+w)(\bar z)(\bar y+\bar w)$





→ Canonical : Convert term $\xrightarrow{to}$ Maxterm

⚠ *Ex :- Canonical , std ?

① $F = (x+\bar{y})(\bar{z}+\bar{w})$?

ORing → ANDing So yes std ✓ (pos).

But not canonical ✗ (Maxterm = $x+y+z+w$)

② $F = (x+y+\bar{z})(\bar{x}+\bar{y}+\bar{z})$?

OR → AND , yes std (pos) ✓

Maxterm ⇒ $M_1 = (x+y+z)$, $M_2 = (\bar{x}+\bar{y}+\bar{z})$ ✓

So yes canonical ✓ (M_1 , M_2 : رقمين مختلفة للتوضيح)

⊛ Maxterm (pos) :-

	x y	Maxterm	F(given)
M_0	0 0	$x+y$	1
M_1	0 1	$x+\bar{y}$	1
M_2	1 0	$\bar{x}+y$	0
M_3	1 1	$\bar{x}+\bar{y}$	1

← فوق الواحد Bar

[ORing → ANDing]

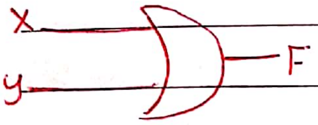
← بنختار ال Zero's .

$F = M_2$

$F = \Pi(2)$

$F = \bar{x}+y$

} 3-way to represent F in (pos).



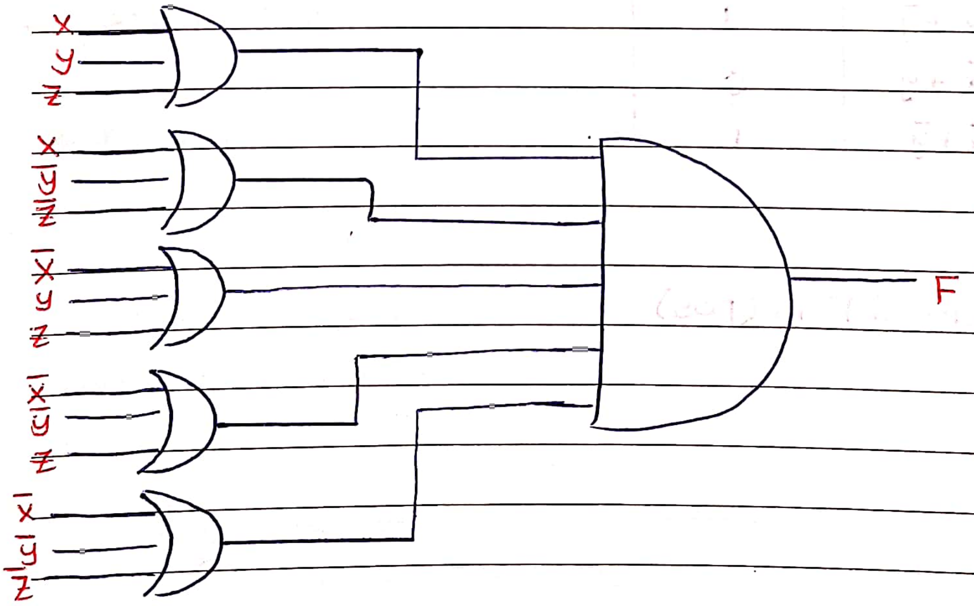
Q: Find F as pos of Maxterm?

X y Z	F	
0 0 0	0	$M_0 = X + y + Z$
0 0 1	1	
0 1 0	1	
0 1 1	0	$M_3 = X + \bar{y} + \bar{Z}$
1 0 0	0	$M_4 = \bar{X} + y + Z$
1 0 1	1	
1 1 0	0	$M_6 = \bar{X} + \bar{y} + Z$
1 1 1	0	$M_7 = \bar{X} + \bar{y} + \bar{Z}$

$$F = (M_0) \cdot (M_3) \cdot (M_4) \cdot (M_6) \cdot (M_7)$$

$$= \Pi(0,3,4,6,7)$$

$$= (X + y + Z)(X + \bar{y} + \bar{Z})(\bar{X} + y + Z)(\bar{X} + \bar{y} + Z)(\bar{X} + \bar{y} + \bar{Z})$$



*Ex:- ① Convert to canonical, $F=(x+y+\bar{z})(x+\bar{y})$?

هون حال (pos)

بعمل إضافة (+0) لل term الناقص

→ pos ✓

$F=(x+y+\bar{z})(x+\bar{y})+0$

$(x+y+\bar{z})(x+\bar{y})+(z\cdot\bar{z})$

$(x+y+\bar{z})(x+\bar{y}+z)\cdot(x+\bar{y}+\bar{z})$

→ canonical ✓

② $F=\Sigma(0,1,\cancel{2},4,7)$, find $\bar{F}$ as a (pos) ?

لما يطلب $\bar{F}$ ويعطيك ال F بال (sop) بنعملها حلين، الأول: خذ نفس الأرقام داخل القوس لكن بدل Σ حط Π يعني (pos).

الثاني: خذ متممة الأرقام داخل القوس وحافظ على شكل ال Σ ← (sop) ✓

① $\bar{F}=\Pi(0,1,4,7)$

② $\bar{F}=\Sigma(2,3,5,6)$

ضبطا لما تاخذ المتممة تأكد إنك غطيت كل الاحتمالات من خلال إنك تشوف أكبر رقم كم عدد ال Bits لإله؟ وعبيهم كلهم

$7 \rightarrow 111 \rightarrow 3\text{ Bits} \rightarrow$ (from (0) to (7))

$\Rightarrow$ as a (pos) so $\bar{F}=\Pi(0,1,4,7)$

③ $(B\bar{C}+\bar{A}D)(A\bar{B}+C\bar{D})=$? "evaluate"

$=A\bar{B}B\bar{C}+B\bar{C}C\bar{D}+\bar{A}A\bar{B}D+\bar{A}CD\bar{D}$

= Zero

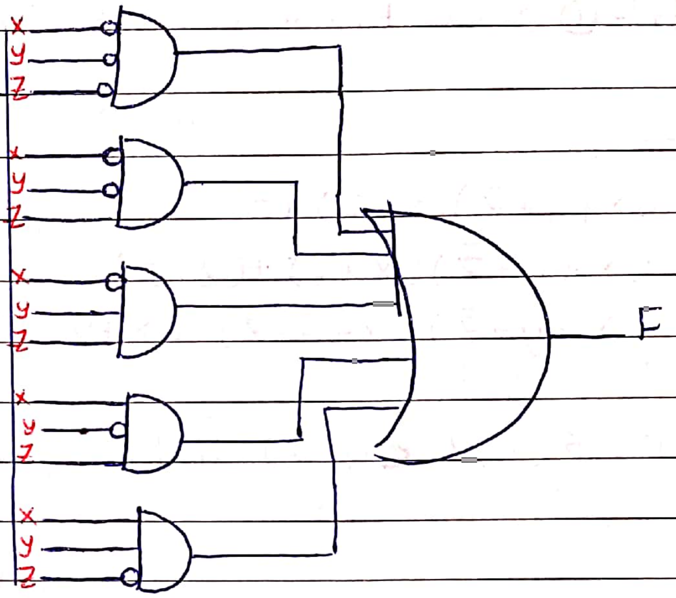
④ $F_1=\bar{x}y+\bar{z}$

$F_2=\Pi(0,1,5,7)$

$F_3=\Sigma(0,1,2,4,5)$

$F_4=(F_1\cdot F_2)\oplus F_3$, find F_4 as a (sop) ?

	x y z	F_1	F_2	F_3	F_4
0	0 0 0	1	0	1	1
1	0 0 1	0	0	1	1
2	0 1 0	1	1	1	0
3	0 1 1	1	1	0	1
4	1 0 0	1	1	1	0
5	1 0 1	0	0	1	1
6	1 1 0	1	1	0	1
7	1 1 1	0	0	0	0

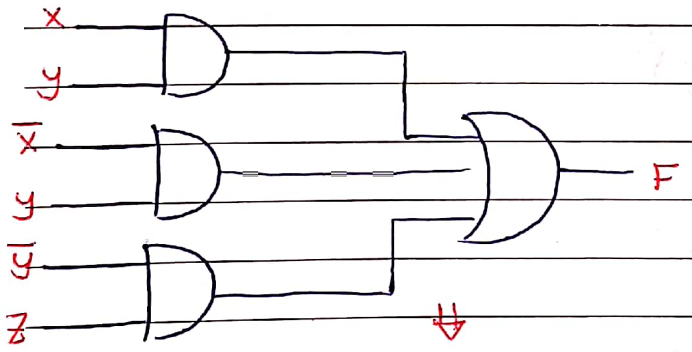


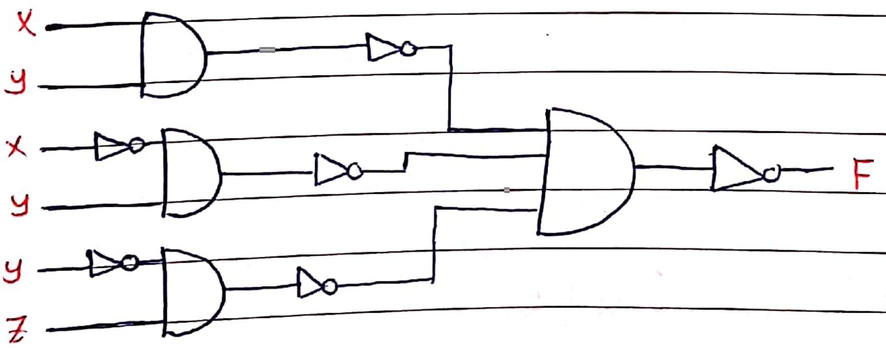
So $F = m'_0 + m_1 + m_3 + m_5 + m_6$

$= \Sigma(0,1,3,5,6)$

$= (\bar{x}\bar{y}\bar{z}) + (\bar{x}\bar{y}z) + (\bar{x}yz) + (x\bar{y}z) + (xy\bar{z})$

Q : $F = xy + \bar{x}y + \bar{y}z$, use AND gate & inverter only ?

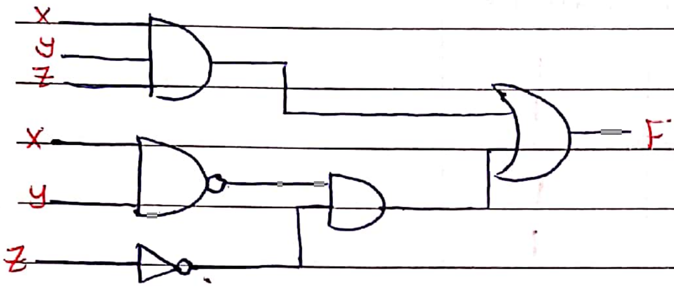


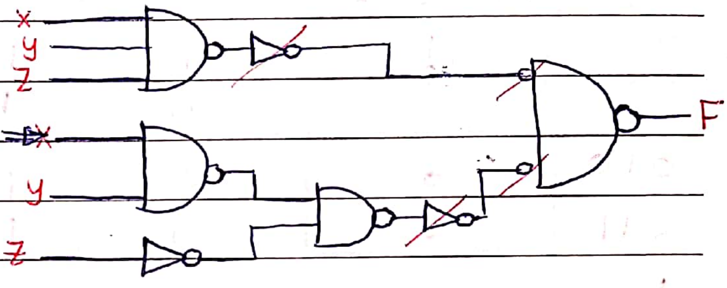


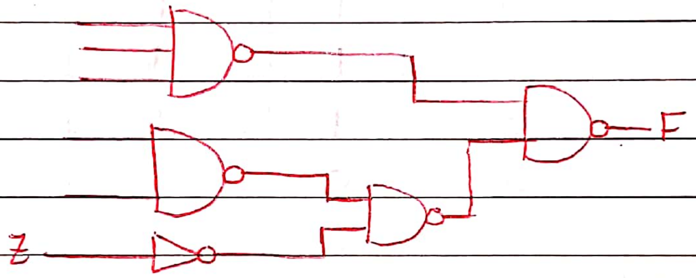
Q: $F = xy \oplus \bar{Z}$, Build cct using NAND ?

→ $X \oplus y = X\bar{y} + \bar{X}y$

$Xy \oplus \bar{Z} = \overline{xy}\bar{Z} + xyZ$

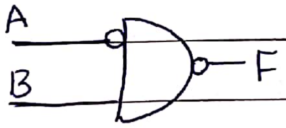






*Active high [AH]:
→ Empty of (Not):
[—o or —▷o—]
pupple inverter
→ if value = 1, active
→ if value = 0, inactive.

*Active low [AL]:
→ Contain [Not]:
[—o or —▷o]
→ if value = 1 → inactive
→ if value = 0 → active.



السلك الذى تكون فيه دائرة
هو [AL]

∴ A & F are [AL]
B is [AH]

((End of Ch.2))

→ $\overline{xy} = \bar{x} + \bar{y}$
→ $\overline{x+y} = \bar{x}\,\bar{y}$
} Demorgen

provied can being by truth table #

The Hashemite University
Computer Engineering Department
Digital Logic (110408220)
HW2

خلف ← **Q2-1)** Demonstrate by means of truth tables the validity of the following identities:

(a) DeMorgan's theorem for three variables: $(x + y + z)' = x'y'z'$ and $(xyz)' = x' + y' + z'$

(b) The distributive law: $x + yz = (x + y)(x + z)$.

درجة خارجية ← **Q2-2)** Simplify the following Boolean expressions to a minimum number of literals:

(a) $xy + xy'$
(b) $(x + y)(x + y')$
(c) $xyz + x'y + xyz'$
(d) $(A + B)'(A' + B')'$

درجة خارجية ← **Q2-3)** Simplify the following Boolean expressions to a minimum number of literals:

(a) $ABC + A'B + ABC$
(b) $x'yz + xz$
(c) $(x + y)'(x' + y')$
(d) $xy + x(wz + wz')$
(e) $(BC' + A'D)(AB' + CD')$

درجة خارجية ← **Q2-4)** Reduce the following Boolean expressions to the indicated number of literals:

(a) $A'C' + ABC + AC'$ to three literals ✓
(b) $(x'y' + z)' + z + xy + wz$ to three literals ✓
(c) $A'B(D' + C'D) + B(A + A'CD)$ to one literal ✓
(d) $(A' + C)(A' + C')(A + B + C'D)$ to four literal ✓

(2-1):- Demorgans.

a)

xyz	$\overline{(x+y+z)}$	$x'y'z'$	$\overline{(xyz)}$	$x'+y'+z'$
000	1	1	1	1
001	0	0	1	1
010	0	0	1	1
011	0	0	1	1
100	0	0	1	1
101	0	0	1	1
110	0	0	1	1
111	0	0	0	0

equal ✓ equal ✓

b)

xyz	$x + yz$	$(x+y)(x+z)$
000	0	0
001	0	0
010	0	0
011	1	1
100	1	1
101	1	1
110	1	1
111	1	1

Distributive.

equal

SUBJECT: HW 2 3-

Q 2-2)

a) $xy + x\bar{y} = x(y+\bar{y}) = x \cdot 1 = \textcircled{x}$ ←

b) $(x+y)(x+\bar{y}) = x + (y \cdot \bar{y}) = \textcircled{x}$ ←

c) $xyz + \bar{x}y + xy\bar{z} = xy(z+\bar{z}) + \bar{x}y$

$= xy + \bar{x}y$

$= y(x+\bar{x}) = \textcircled{y}$ ←

d) $\overline{(A+B)}(\overline{\bar{A}+\bar{B}}) = \bar{A}\bar{B} \cdot AB = \textcircled{0}$ ←

Q 2-3)

a) $ABC + \bar{A}B + ABC = ABC + \bar{A}B$

$= B(AC + \bar{A}) = \boxed{B(\bar{A}+C)}$ ←

$= B\bar{A} + BC$ α as min # of literals

b) $\bar{x}yz + xz = z(\bar{x}y + x) = \boxed{z(y+x)}$ ←

c) $\overline{(x+y)}(\bar{x}+\bar{y}) = \bar{x}\bar{y}(\bar{x}+\bar{y}) = \bar{x}\bar{y}\bar{x} + \bar{x}\bar{y}\bar{y}$

$= \bar{x}\bar{y} + \bar{x}\bar{y} = \textcircled{\bar{x}\bar{y}}$ ←

d) $xy + x(wz + w\bar{z}) = xy + xw(z+\bar{z})$

$= \boxed{x(y+w)}$ ←

e) $(B\bar{C} + \bar{A}D)(A\bar{B} + C\bar{D}) = A\cancel{\bar{B}B}\bar{C} + B\cancel{\bar{C}C}\bar{D}$

$+ \cancel{\bar{A}A}\bar{B}D + \bar{A}C\cancel{D\bar{D}}$

$= \textcircled{0}$ ←

(P 2-4)

a) $\bar{A}\bar{C} + ABC + A\bar{C} = \bar{C}(\bar{A}+A) + ABC$

$= \bar{C} + AB\cancel{C}$

$= \boxed{\bar{C} + AB}$ ← 3 literals

b) $\overline{(\bar{x}\bar{y} + z)} + z + xy + wz$

$= \overline{\bar{x}\bar{y}} \cdot \bar{z} + z + xy + wz$

$= (x + y)\cancel{\bar{z}} + z + xy + wz$

$= \underline{x} + y + z + \underline{xy} + wz$

$= x(1+y) + y + z + wz$

$= x + y + z(1+w)$

$= \boxed{x + y + z}$ ← 3 literals

c) $\bar{A}B(\bar{D} + \bar{C}D) + B(A + \bar{A}CD)$

$= \bar{A}B\bar{D} + \underline{\bar{A}B\bar{C}D} + AB + \underline{\bar{A}BCD}$

$= \bar{A}BD(\bar{C}+C) + \bar{A}B\bar{D} + AB$

$= \bar{A}BD + \bar{A}B\bar{D} + AB$

$= \bar{A}B(D+\bar{D}) + AB$

$= \bar{A}B + AB = B(\bar{A}+A) = \textcircled{B}$ ←

d) $(\bar{A}+C)(\bar{A}+\bar{C})(A+B+\bar{C}D)$

~~$= \bar{A}A + A\bar{C}$ $\quad \bar{A} + (C\bar{C})$~~

$= [\bar{A} + (C\bar{C})](A + B + \bar{C}D) = \bar{A}(A+B+\bar{C}D)$

$= \bar{A}A + \bar{A}B + \bar{A}\bar{C}D = \boxed{\bar{A}(B+\bar{C}D)}$ ← 4-literals

دراسة خارجية ١٥ — **Q2-5)** Find the complement of $F = x + yz$; the show that $FF' = 0$ and $F + F' = 1$.

٩ — **Q2-6)** Find the complement of the following expressions:
(a) $xy' + x'y$
(b) $(AB' + C)D' + E$
(c) $(x + y' + z)(x' + z')(x + y)$

Q2-8) List the truth table of the function: $F = xy + xy' + y'z$

Q2-9) Logical operations can be performed on string of bits by considering each pair of corresponding bits separately (this is called bitwise operation). Given two 8-bit strings $A = 10101101$ and $B = 10001110$, evaluate the 8-bit result after the following logical operations: **(a) AND, (b) OR, (c) XOR , (d)NOT A, (e) NOT B**

Q2-10) Draw the logic diagrams for the following Boolean expressions:
(a) $Y = A'B' + B(A + C)$ (b) $Y = BC + AC'$
(c) $Y = A + CD$ (d) $Y = (A + B)(C' + D)$

Q2-11) Given the Boolean function:

$$F = xy + x'y + y'z$$

(a) implement it with **AND, OR**, and inverter gates,
(b) implement it with **OR** and inverter gates,
(c) implement it with **AND** and inverter gates.
(d) implement it with **NAND** and inverter gates.
(e) implement it with **NOR** and inverter gates.

Q2-12) Simplify the Boolean function T_1 and T_2 to a minimum number of literals. Typo

	A	B	C	T_1	T_2
0	0	0	0	1	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	0	1
5	1	0	1	0	1
6	1	0	0	0	1
7	1	1	1	0	1

1st exam

Q2-14) Obtain the truth table of the following functions and express each function in sum of minterms and product of maxterms:
(a) $(xy + z)(y + xz)$. (b) $(A' + B)(B' + C)$ (c) $y'z + wxy' + wxz' + w'x'z$.

دراسة خارجية

Q2-15) Given the Boolean function $F = xy'z + x'y'z + w'xy + wx'y + wxy$.
(a) Obtain the truth table of the function.
(b) Draw the logic diagram using the original Boolean expression.
(c) Simplify the function to a Minterm number of literals using Boolean algebra.

Typo (minimum)

(d) Obtain the truth table of the function from simplified expression and show that it is the same as one in part (a).

(e) Draw the logic diagram from the simplified expression and compare the total number of gates with the diagram of part (b).

[illegible]

Q2-16) Express the following function in sum of minterms and product of maxterms:
F(A,B,C,D) = B'D +A'D +BD.

sop pos

Q2-19) convert the following expressions into sum of products and product of sums :

(a) (AB +C) (B +C'D) (b) x' + x(x + y')(y + z').

(yx + u) (x + u'v)

Q2-20) draw the logic diagram corresponding to the following Boolean expression without simplifying them:

(a) BC' +AB +ACD. (b) (A + B) (C+ D)(A' +B +D).

(C) (AB + A'B')(CD' +C'D).

Q 2-5)

→ $\bar{F} = \overline{x+yz} = \bar{x}\,\overline{yz} = \bar{x}(\bar{y}+\bar{z}) = (\bar{x}\bar{y}+\bar{x}\bar{z})$

→ $F\bar{F} = (x+yz)(\bar{x}\bar{y}+\bar{x}\bar{z})$

$= \cancel{x\bar{x}}\bar{y} + \cancel{x\bar{x}}\bar{z} + \bar{x}\cancel{\bar{y}y}z + \bar{x}y\cancel{z\bar{z}}$

$= \textcircled{0}$ ←

→ $F+\bar{F} = (x+yz) + \bar{x}\bar{y} + \bar{x}\bar{z}$

$= \underline{\underline{x}} + yz + \underline{\underline{\cancel{\bar{x}}\bar{y}}} + \underline{\underline{\cancel{\bar{x}}\bar{z}}}$

$= x + \underline{\underline{y\cancel{z}}} + \bar{y} + \underline{\underline{\bar{z}}}$

$= x + (y+\bar{y}) + \bar{z}$

$= x + (1) + \bar{z} = \textcircled{1}$ ←

Q 2-6)

a) $\bar{F} = \overline{x\bar{y}+\bar{x}y} = \overline{x\bar{y}}\cdot\overline{\bar{x}y} = (\bar{x}+y)(x+\bar{y})$

$= \bar{x}x + \bar{x}\bar{y} + xy + y\bar{y} = \boxed{xy + \bar{x}\bar{y}}$ ←

b) $\overline{(A\bar{B}+C)\bar{D}+E} = \overline{(A\bar{B}+C)\bar{D}}\cdot\bar{E}$

$= (\overline{(A\bar{B}+C)}+D)\,\bar{E}$

$= ((\overline{A\bar{B}}\cdot\bar{C})+D)\,\bar{E}$

$= ((\bar{A}+B)\bar{C}+D)\,\bar{E}$

$= (\bar{A}\bar{C}+B\bar{C})\bar{E} + D\bar{E} = \bar{A}\bar{C}\bar{E} + B\bar{C}\bar{E} + D\bar{E}$ ←

c) $\overline{(x+\bar{y}+z)(\bar{x}+\bar{z})(x+y)} = \overline{(x+\bar{y}+z)} + \overline{(\bar{x}+\bar{z})} + \overline{(x+y)}$

$= \underline{\bar{x}y\bar{z}} + xz + \underline{\bar{x}\bar{y}}$ ←

$= \bar{x}(\cancel{y}\bar{z}+\bar{y}) + xz = \bar{x}\bar{z} + \bar{x}\bar{y} + xz$ ←

equal

Q(2-8) :- $F = xy + x\bar{y} + \bar{y}z$

x y z	F
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	1

→ Convert to Canonical:

$$F = xy(z+\bar{z}) + x\bar{y}(z+\bar{z}) + \bar{y}z(x+\bar{x})$$

$$= xyz + xy\bar{z} + x\bar{y}z + x\bar{y}\bar{z} + x\bar{y}z + \bar{x}\bar{y}z$$

$$= \underset{111}{xyz} + \underset{110}{xy\bar{z}} + \underset{101}{x\bar{y}z} + \underset{100}{x\bar{y}\bar{z}} + \underset{001}{\bar{x}\bar{y}z}$$

as a (Sop) ✓

Q(2-9):- A = 10101101

B = 10001110

Using betwise operation (consider each pair of corresponding bits) are separately

a) AND: 1 0 1 0 1 1 0 1

and 1 0 0 0 1 1 1 0

(1 0 0 0 1 1 0 0) ✓

b) 10101101 → A

! 10001110 → B

(e) (10101111) OR

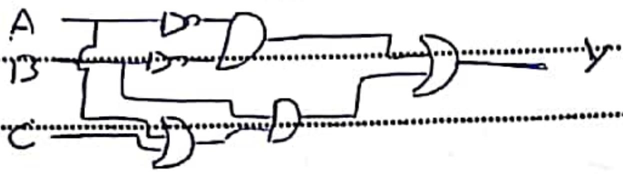
(00100011) XoR

(01010010) NoT A

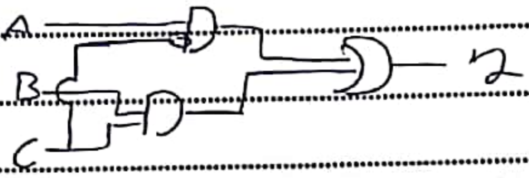
(01110001) NoT B ✓

Q 2-12)

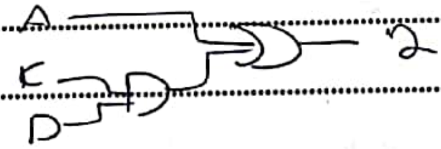
a) $Y = \bar{A}\bar{B} + B(A+C)$



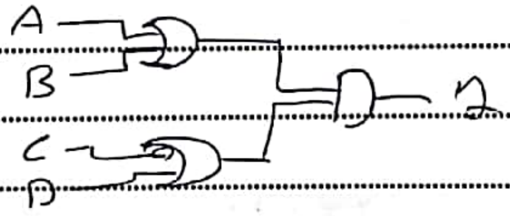
b) $y = BC + A\bar{C}$



c) $y = A+CD$



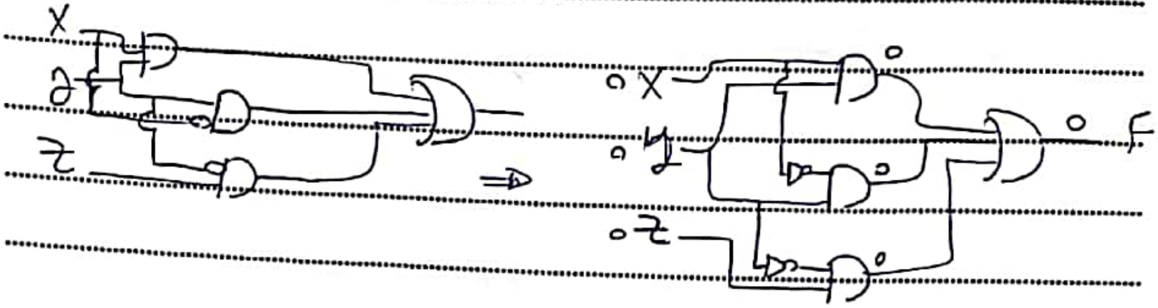
d) $y = (A+B)(\bar{C}+D)$



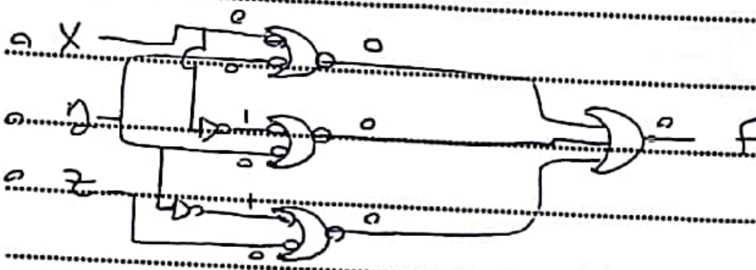
P 2-11) $F = xy + \bar{x}y + \bar{y}z$

0 0 $\bar{0}$ 0 $\bar{0}$ 0 = 0

a) imp. (AND, OR, inverter) ;



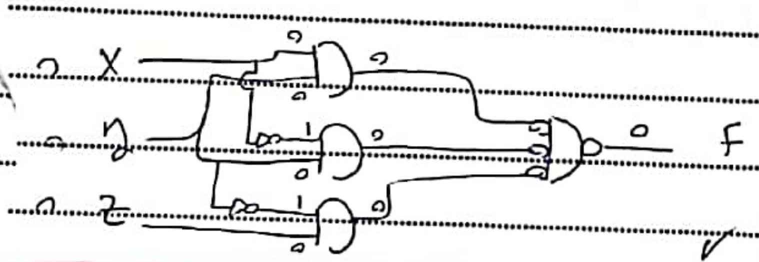
b) (OR, inverter) ; $xy = \overline{\bar{x}+\bar{y}}$



(غلط)

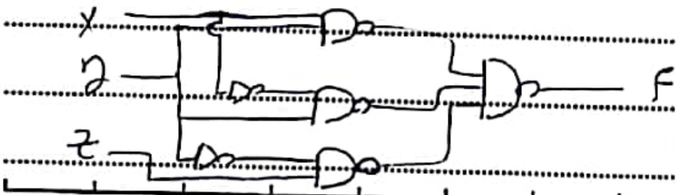
ليش ؟

c) (AND, inv.) ; $x+y = \overline{\bar{x}\bar{y}}$

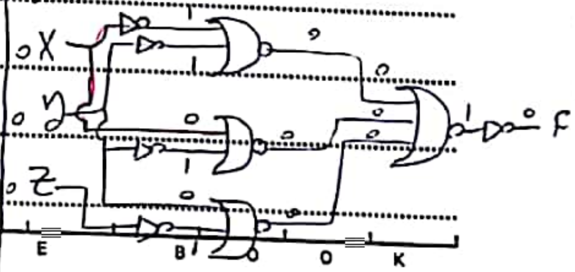


نحول بالأذكر (a, b, c) حط مكان كل (فقاعة) (—▷—) هيك بتغير صح

d) (NAND & inverter)



e) (NOR, inv) ;



Q 2-12)

$\Rightarrow T_2 = \overline{T_1}$, as (SOP) ;

$T_1 = m_0 + m_1 + m_2$

$$T_1 = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC$$
$$= \bar{A}\bar{B}(\bar{C}+C) + \bar{A}BC$$
$$= \bar{A}\bar{B} + \bar{A}BC$$
$$= \bar{A}(\bar{B} + \cancel{B}C) = \boxed{\bar{A}(\bar{B}+C)} \rightarrow \text{3-literals}$$

$$\Rightarrow T_2 = \overline{T_1} = \overline{(\bar{A}\bar{B} + \bar{A}C)} = (\overline{\bar{A}\bar{B}})(\overline{\bar{A}C})$$
$$= (A+B)(A+\bar{C}) = \boxed{A+(B\bar{C})} \rightarrow \text{3-literals}$$

Q 2-14) (c) $F = \bar{y}z + x\bar{y}w + x\bar{z}w + \bar{x}z\bar{w}$

x y z w	$\bar{y}z$	$x\bar{y}w$	$x\bar{z}w$	$\bar{x}z\bar{w}$	F	
0 0 0 0	0	0	0	0	0	0
0 0 0 1	0	0	0	0	0	1
0 0 1 0	1	0	0	1	1	2
0 0 1 1	1	0	0	0	1	3
0 1 0 0	0	0	0	0	0	4
0 1 0 1	0	0	0	0	0	5
0 1 1 0	0	0	0	1	1	6
0 1 1 1	0	0	0	0	0	7
1 0 0 0	0	0	0	0	0	8
1 0 0 1	0	1	1	0	1	9
1 0 1 0	1	0	0	0	1	10
1 0 1 1	1	1	0	0	1	11
1 1 0 0	0	0	0	0	0	12
1 1 0 1	0	0	1	0	1	13
1 1 1 0	0	0	0	0	0	14
1 1 1 1	0	0	0	0	0	15

$$F = \Sigma(2,3,6,9,10,11,13) = \Pi(0,1,4,5,7,8,12,14,15)$$

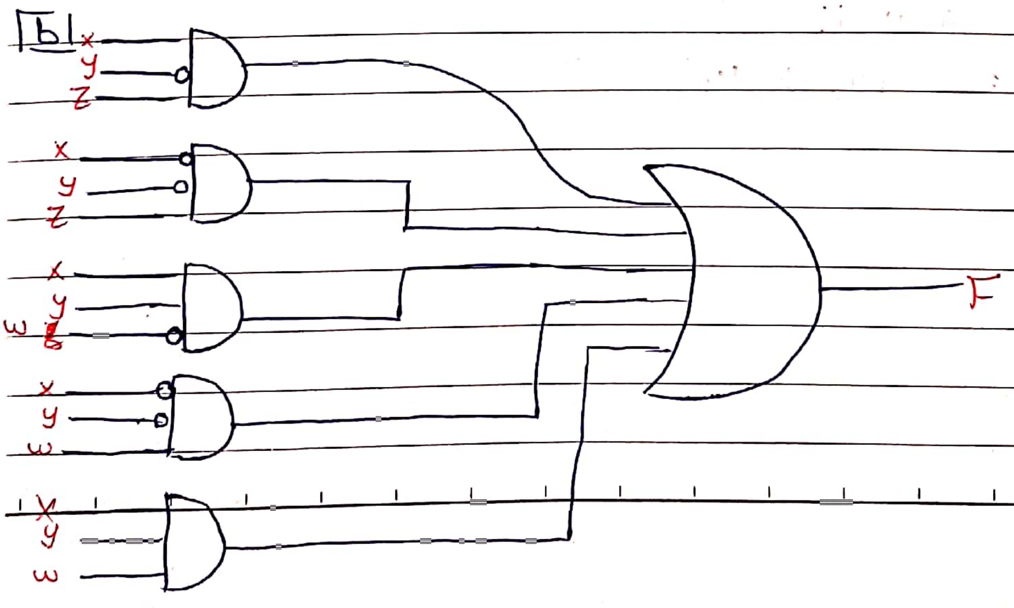
Q(2-15):- $F = x\bar{y}z + \bar{x}\bar{y}z + xy\bar{w} + \bar{x}yw + xy\bar{w}$

a

x y z w	$x\bar{y}z$	$\bar{x}\bar{y}z$	$xy\bar{w}$	$\bar{x}yw$	xyw	F	
0 0 0 0	0	0	0	0	0	0	0
0 0 0 1	0	0	0	0	0	0	1
0 0 1 0	0	1	0	0	0	0	2
0 0 1 1	0	1	0	0	0	0	3
0 1 0 0	0	0	0	0	0	0	4
0 1 0 1	0	0	0	1	0	1	5
0 1 1 0	0	0	0	0	0	0	6
0 1 1 1	0	0	0	1	0	1	7
1 0 0 0	0	0	0	0	0	0	8
1 0 0 1	0	0	0	0	0	0	9
1 0 1 0	1	0	0	0	0	1	10
1 0 1 1	1	0	0	0	0	1	11
1 1 0 0	0	0	1	0	0	1	12
1 1 0 1	0	0	0	0	1	1	13
1 1 1 0	0	0	1	0	0	1	14
1 1 1 1	0	0	0	0	1	1	15

$F = \Sigma(2,3,5,7,10,11,12,13,14,15)$, (Sop)

b



11-gates.

$\boxed{C}$ $x\bar{y}z \cdot (w+\bar{w}) \rightarrow x\bar{y}zw + x\bar{y}z\bar{w}$

$\bar{x}\bar{y}z \cdot (w+\bar{w}) \rightarrow \bar{x}\bar{y}zw + \bar{x}\bar{y}z\bar{w} +$

$xy\bar{w} \cdot (z+\bar{z}) \rightarrow xyz\bar{w} + xy\bar{z}\bar{w} +$

$\bar{x}yw \cdot (z+\bar{z}) \rightarrow \bar{x}yzw + \bar{x}y\bar{z}w +$

$xyw \cdot (z+\bar{z}) \rightarrow xyzw + xy\bar{z}w +$ ✓

d) $F = \bar{y}z(x + \bar{x}) + xy(\bar{w} + w) + \bar{x}yw$

$= \bar{y}z + xy + \bar{x}yw$

$= \bar{y}z + y(x + \bar{x}w)$

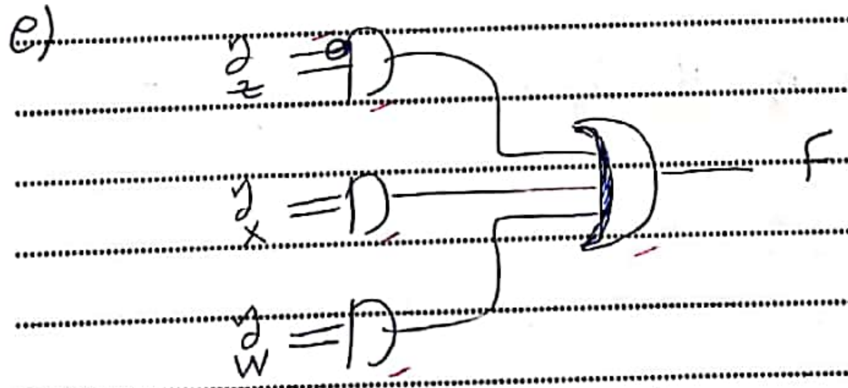
$= \boxed{\bar{y}z + yx + yw} \leftarrow$

x y z w	$\bar{y}z$	yx	yw	F	
0 0 0 0	0	0	0	0	0
0 0 0 1	0	0	0	0	1
0 0 1 0	1	0	0	①	②
0 0 1 1	1	0	0	①	③
0 1 0 0	0	0	0	0	4
0 1 0 1	0	0	1	①	⑤
0 1 1 0	0	0	0	0	6
0 1 1 1	0	0	1	①	⑦
1 0 0 0	0	0	0	0	8
1 0 0 1	0	0	0	0	9
1 0 1 0	1	0	0	①	⑩
1 0 1 1	0	0	0	①	⑪
1 1 0 0	0	1	0	①	⑫
1 1 0 1	0	1	1	①	⑬
1 1 1 0	0	1	0	①	⑭
1 1 1 1	0	1	1	①	⑮

$F = \Sigma(2, 3, 5, 7, 10, 11, 12, 13, 14, 15)$

the Same # ←

e)



5 - gates

less gates by: (11-5)

= 6 gate

Q(2-16):-

→ $\bar{B}D\cdot(A+\bar{A}) = (A\bar{B}D + \bar{A}\bar{B}D)\cdot(C+\bar{C})$

$= [A\bar{B}CD + A\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}\bar{B}\bar{C}D]$ ①

→ $\bar{A}D\cdot(B+\bar{B}) = (\bar{A}BD + \bar{A}\bar{B}D)\cdot(C+\bar{C})$

$= [\bar{A}BCD + \bar{A}B\bar{C}D + \bar{A}\bar{B}CD + \bar{A}\bar{B}\bar{C}D]$ ②

→ $BD\cdot(A+\bar{A}) = (ABD + \bar{A}BD)\cdot(C+\bar{C})$

$= [ABCD + AB\bar{C}D + \bar{A}BCD + \bar{A}B\bar{C}D]$ ③

$\cancel{11}$ 1011 $\cancel{9}$ 1001 $\cancel{3}$ 0011 $\cancel{1}$ 0001 $\cancel{7}$ 0111 $\cancel{5}$ 0101

∴ $F = A\bar{B}CD + A\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}\bar{B}\bar{C}D + \bar{A}BCD + \bar{A}B\bar{C}D$

$+ ABCD + AB\bar{C}D$

1111 1101

$\cancel{15}$ 13

$F = \Sigma(1,3,5,7,9,11,13,15)$

$F = \Pi(0,2,4,6,8,10,12,14)$.

Q(2-19):- Convert from non-std.

[a] $(AB+C)(B+\bar{C}D)$;

$= ABB + AB\bar{C}D + BC + \cancel{C\bar{C}D}^{0}$

$= AB + AB\bar{C}D + BC$ Sop:

$= AB(1+\bar{C}D) + BC = \boxed{AB + BC}$

pos:- $B.(A+C)$

[b] $\bar{x} + x(x+\bar{y})(y+\bar{z})$

$= \bar{x} + xy + x\bar{z} + \bar{y}y + \bar{y}\bar{z}$

$= \bar{x} + y + (\bar{z} + \bar{y}\bar{z}) = (\bar{x} + y + \bar{z})$

Sop:- $\bar{x} + xy + x\bar{z} + \bar{y}\bar{z}$

⊛ Chapter 3 / Gate level minimization.

☒ Karnough maps :-

* 2 - variables :- $\left[2^{(\# \text{ of variables})} = \# \text{ of cells} \right]$

$\hookrightarrow 2^2 = 4$

x y	
0 0	m_0
0 1	m_1
1 0	m_2
1 1	m_3

→

x \ y	0	1
0	m_0	m_1
1	m_2	m_3

→

x \ y	0	1
0	0) 00	1) 01
1	2) 10	3) 11

↳ m_0

{ متجاورة مع m_1 & m_2 } / التجاور يكون إما أفقيا / أو عامودي

يجب أن يكون الإختلاف بينهم بمتغير واحد فقط

1) Select the Suitable K-map.
2) map the Function to the K-map.
3) Grouping based on adjacent cells.

تختلف مع ما فوقها أو تحتها أو جانبها بمتغير واحد فقط

* 3 - variables :- $2^3 = 8$

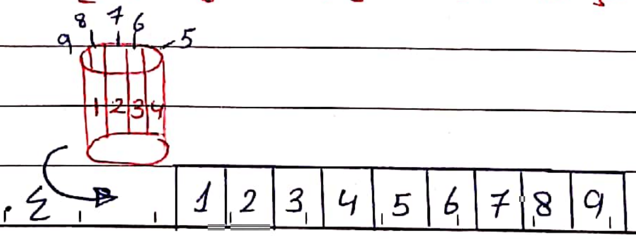
بدلنا ترتيبهم كي احقق شرط التجاور

X Y Z	
0 0 0	0
0 0 1	1
0 1 0	2
0 1 1	3
1 0 0	4
1 0 1	5
1 1 0	6
1 1 1	7

X \ YZ	00	01	11	10
0	0) m_0	1) m_1	3) m_3	2) m_2
1	4) m_4	5) m_5	7) m_7	6) m_6

كون m_2 متجاورة مع m_0 & m_1 & m_3

[الخريطة كأنها عبارة عن إفراد لشكل اسطواني



OR

yz \ x	0	1
00	0) m_0	4) m_4
01	1) m_1	5) m_5
11	3) m_3	7) m_7
10	2) m_2	6) m_6

* 4 -variables :- $2^4 = 16$

xy \ zw	00	01	11	10
00	0)	1)	3)	2)
01	4)	5)	7)	6)
11	12)	13)	15)	14)
10	8)	9)	11)	10)

m_0 & m_3 & m_6 & m_{10} : تجاور m_2

إبجسم الترتيب

⚠ Ex:- ① $F = \Sigma(0,1,3)$, map(F) ?

Large # as(3) → 11 → 2-variables ✓

(Sop) → حط (واحد) عند كل خلية مختارة ؛

واملى الباقي (أصفار) .

x \ y	0	1
0	0) 1	1) 1
1	2) 0	3) 1

② $F = x \oplus y$, map(F) ؟

x \ y	0	1
0	(0) 0	(1) 1
1	(2) 1	(3) 0

xy	F	
0 0	0	m_0
0 1	1	m_1
1 0	1	m_2
1 1	0	m_3

as(sop) ← 1 1

③ $F = \Pi(0,1,5,7)$ ؟

7 → 111 → 3-variables ✓

(pos) → حط (صفر) عند كل خلية مختارة ؛ والباقي ← إملى (واحدات).

x \ yz	00	01	11	10
0	(0) 0	(1) 0	(3) 1	(2) 1
1	(4) 1	(5) 0	(7) 0	(6) 1

④ $F = xyz + \bar{x}\bar{y} + \bar{x}$ ؟ (map F).

xyz → 3-variables;

(ANDing → ORing) → sop ✓ → (Bar = 0)

↳ $(x \cdot y \cdot z) + (\bar{x} \cdot \bar{y}) + (\bar{x})$ (otherwise = 1)

x \ yz	00	01	11	10
0	(0) 1	(1) 1	(3) 1	(2) 1
1	(4) 0	(5) 0	(7) 1	(6) 0

, $xyz + \bar{x}\bar{y} + \bar{x}$

1 1 1 → خلية (7)

0 0 → ثبت ال ($\bar{x}\bar{y}$) 0 0 وعبيهم د(1) بغض النظر عن قيمة (Z)

0 → ثبت (0) عند ال (x) وعبي (1) بغض النظر عن (Z) ، (y)

⇐ الباقي صفار.

⑤ $F = x\bar{y}\bar{z}w + x\bar{y} + \bar{w}$, map(F) ?

xyzw → 4-variables;
(ANDing → oRing) → sop ✓

xy \ zw	00	01	11	10
00	0 1	1 0	3 0	2 1
01	4 1	5 0	7 0	6 1
11	12 1	13 0	15 0	14 1
10	8 1	9 1	11 1	10 1

$$, \underset{\underset{⑨}{\downarrow}}{\underset{1001}{x\bar{y}\bar{z}w}} + \underset{\underset{\text{نثبتهم ونعبي واحدات عند كل الخلايا المشتركة معهم}}{\downarrow}}{\underset{10}{x\bar{y}}} + \underset{\underset{\text{نثبته ونعبي}}{\downarrow}}{\underset{0}{\bar{w}}}$$

⑥ $F = (x+y) \cdot (x+\bar{z})$, map(F) ?

xyz → 3-variables.
(oRing → ANDing) → pos → $\left(\begin{matrix}\text{Bar} = 1\\ 0.w = 0\end{matrix}\right)$

x \ yz	00	01	11	10
0	0 0	1 0	3 0	2 1
1	4 1	5 1	7 1	6 1

$$, (x+y)(x+\bar{z})$$

$$\underbrace{\underbrace{10\ \ 0,}\ \ \underbrace{10\ \ 1,}}_{\downarrow}$$

نثبتهم ونعبي كل المشترك (أصفار).

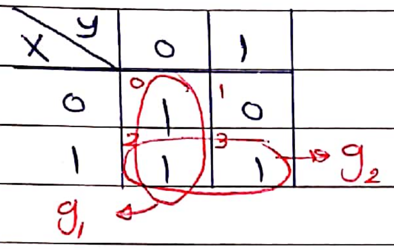
* note :- IF (F) is unstd; Then Convert it to std either (sop) or (pos) #

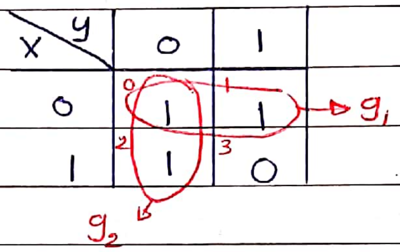
☒ Grouping :-

(طلع (F) من ال k-map)
→ get simplified F.

1) Sop :-

[a]





☒ شروط ال Grouping :-

1) عدد خلايا ال Group لازم تكون من مضاعفات ال (2) : [1, 2, 4, 8, 16, 32, ...]

2) هل أخذت أقل عدد من ال Groups ؟

3) هل أخذت أكبر عدد من الخلايا بكل Group ؟

4) قبل كل الي فوق لازم أحدد sop ولا pos ; ← إذا sop (دور على الواحدات)

← إذا pos (دور على الأصفار)

[a] $F = ?$

[indicat ANDing]

$F = g_1 + g_2$

→ ORing

$F = \bar{y} + x$

g_1 :

⊛ x بتغير ∝

y ثابت ع (٥)

اذاً $(\bar{y})$

g_2 :

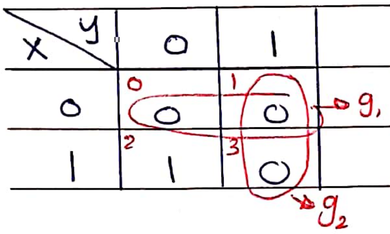
⊛ x ثابت ع (١)

اذاً (x)

y بتغير ∝

[b] $F = g_1 + g_2 = \bar{x} + \bar{y}$ ✓

2) pos:-



x \ y	0	1
0	0	0
1	1	0

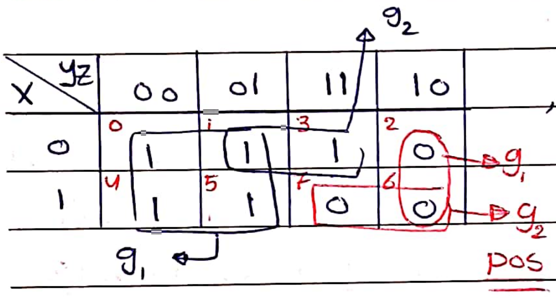
[indicate ORing]

, $F = g_1 \cdot g_2$

ANDing

$F = x \cdot \bar{y}$

Ex:- ①



x \ yz	00	01	11	10
0	1	1	1	0
1	1	1	0	0

, F ?

Sop: $g_1 + g_2 = F$

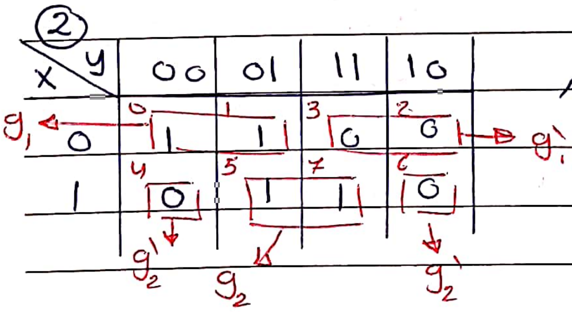
only y - Fixed = 0 ∴ $\bar{y}$

x - Fixed = 0
z - Fixed = 1
∴ $(\bar{x} \cdot z)$

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

pos: $F = g_1 \cdot g_2 = (\bar{y} . z) . (\bar{x} + \bar{y})$

②



x \ y	00	01	11	10
0	1	1	0	0
1	0	1	1	0

, F = ?

Sop: $F = g_1 + g_2 = \bar{x}\bar{y} + xz$

pos: $F = (g_1') \cdot (g_2') = (x + \bar{y}) \cdot (\bar{x} + z)$

③

x \ yz				
	$_{0}$ 0	$_{1}$ 1	$_{3}$ 1	$_{2}$ 1
	$_{4}$ 1	$_{5}$ 1	$_{7}$ 1	$_{6}$ 1

, F = ?

$g_1' \leftarrow$ $\quad g_3 \quad g_2 \quad g_1$

Sop: $F = g_1 + g_2 + g_3$

↳ $F = y + z + x$

pos: $F = (g_1') = (x + y + z)$

④

x \ yz	00	01	11	10
0	1	0	1	1
1	1	0	1	1

, F ?

$g_1 \quad g_1' \quad g_2$

Sop: $F = g_1 + g_2$

$= \bar{z} + y$

pos: $F = (g_1') = (y + \bar{z})$

⑤

x \ yz				
	1	0	1	0
	1	1	1	0

, F = ?

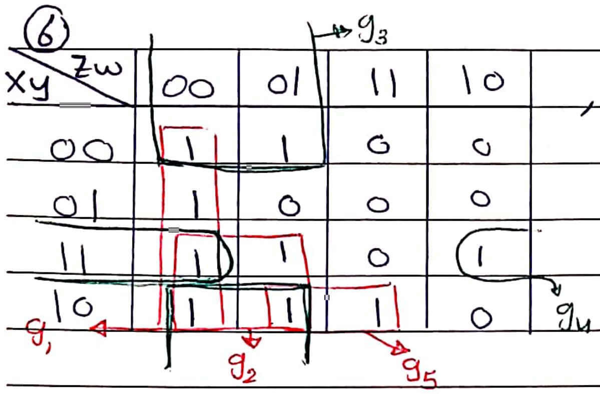
$g_1 \leftarrow \quad \rightarrow g_2'$

* إذا الصفر عدده أقل خذ pos
إذا الواحد " " " sop
} ← للتسهيل

, pos: $F = (g_1').(g_2')$

$= (x + y + \bar{z}).(\bar{y} + z)$

⑥



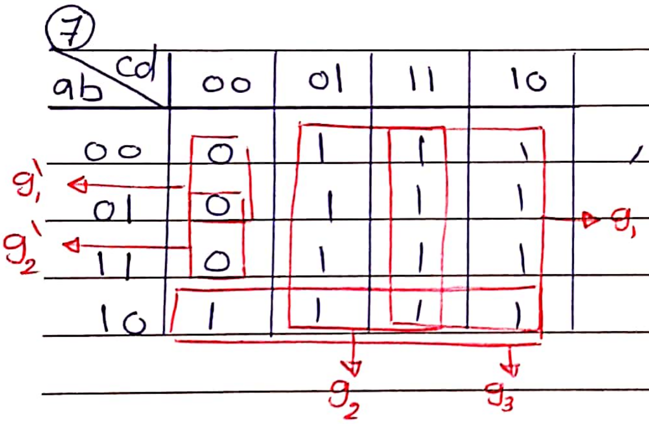
xy \ zw	00	01	11	10
00	1	1	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	1	0

, F = ?

, Sop: $F = g_1 + g_2 + g_3 + g_4 + g_5$

$F = \bar{z}\bar{w} + x\bar{z} + \bar{y}\bar{z} + xy\bar{w} + x\bar{y}w$

⑦



ab \ cd	00	01	11	10
00	0	1	1	1
01	0	1	1	1
11	0	1	1	1
10	1	1	1	1

, F = ?

Sop: $F = g_1 + g_2 + g_3$
$= c + d + a\bar{b}$

pos: $F = (a + c + d)(\bar{b} + c + d)$

[*] Don't care :- (X or x)

بتمدها حسب حاجتي إلها ، فممكن أخبيها صفر أو واحد

EX:- ①

xy \ zw	00	01	11	10
00	1	X	0	0
01	1	X	0	0
11	1	0	X	X
10	1	1	0	0

, F = ؟

g_1

g_2

Sop : $F = \bar{z}\bar{w} + \bar{y}\bar{z}$ باخذ بعين الإعتبار اني آخذ أقل عدد من الـ Groups وأكبر عدد من الـ Celles داخل كل Group .
وإني أشمل كل الواحدات . حالة (sop) و كل الأصفار . حالة (pos) .

②

a \ bc	00	01	11	10
0	X	0	0	1
1	1	0	0	0

, F = ؟

g_2

g_1

Sop : $F = \bar{b}\bar{c} + \bar{a}\bar{c}$ ✓

③ $F = \Sigma(0,1,5,7) + d\Sigma(14,15)$ map(F) ؟

d = Don't Care (x)

15 → 1111 → 4-variables

→ يتبع

xy \ zw	00	01	11	10
00	0 1	1 1	3	2
01	4 0	5 1	7 1	6 0
11	12 0	13 0	15 X	14 X
10	8 0	9 0	11 0	10 0

✓

④ "فكرة مهمة"

xy \ zw	00	01	11	10
00	0 1	1 1	3 1	2 1
01	0	1	1	1
11	0	1	1	1
10	1	1	1	1

g_1 g_3 g_2

, F = ?

حكيت سابقاً إنه الخريطة إفراد
بإسطوانة لكن فعلياً هي (كرة)
لإنه أعلاها ملتصق بأسفلها وأقصى
اليمين ملتصق بأقصى اليسار.

Sop : $F = g_1 + g_2 + g_3$

$= \bar{y}\bar{w} + z + w$

الفكرة لمن تكون الزوايا (1) كلها خدهم Group

((End of ch.3))

The Hashemite University
Computer Engineering Department
Digital Logic (110408220)
HW3

✓ **Q3-1)** Simplify the following Boolean functions, using three-variable maps:

(a) $F(x,y,z)=\sum(0,2,6,7)$
(b) $F(A,B,C)=\sum(0,2,3,4,6)$
(c) $F(a,b,c)=\sum(0,1,2,3,7)$
(d) $F(x,y,z)=\sum(3,5,6,7)$

Solution:

(a)

x\ yz	00	01	11	10
0				
1				

(b)

A\ BC	00	01	11	10
0				
1				

(C)

a\bc	00	01	11	10
0				
1				

(d)

x\ yz	00	01	11	10
0				
1				

✓ **Q3-2)** Simplify the following Boolean functions, using three-variable maps:

(a) $F(x,y,z)=\sum(0,1,5,7)$
(b) $F(x,y,z)=\sum(1,2,3,6,7)$

Solution:

(a)

x\ yz	00	01	11	10
0				
1				

(b)

x\yz	00	01	11	10
0				
1				

✓ **Q3-3)** Simplify the following Boolean expressions, using three-variable maps:

(a) $xy+x'y'z'+x'yz'$
(b) $x'y'+yz+x'yz'$
(c) $A'B+BC'+B'C'$

Solution:

(a)

x\yz	00	01	11	10
0				
1				

(b)

x\yz	00	01	11	10
0				
1				

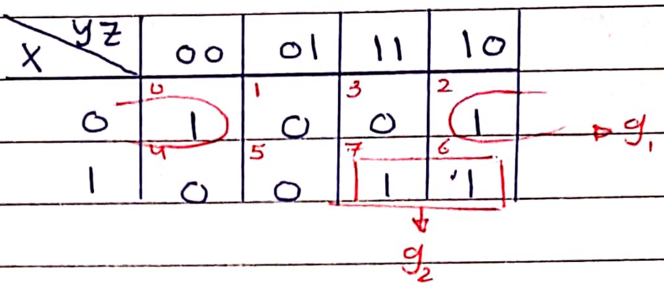
(c)

A\ BC	00	01	11	10
0				
1				

Q(3-1): $\boxed{a}$ $F = \Sigma(0, 2, 6, 7)$

7 → 111 → 3V

$F = \bar{x}\bar{z} + xy$

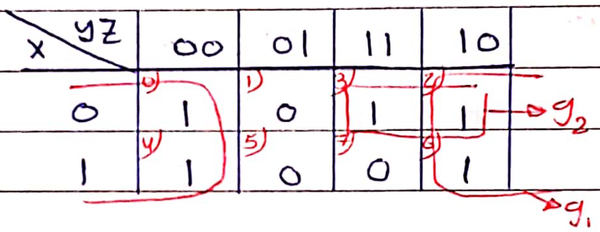


x \ yz	00	01	11	10
0	1	0	0	1
1	0	0	1	1

$\boxed{b}$ $F = \Sigma(0, 2, 3, 4, 6)$

6 → 110 → 3V

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

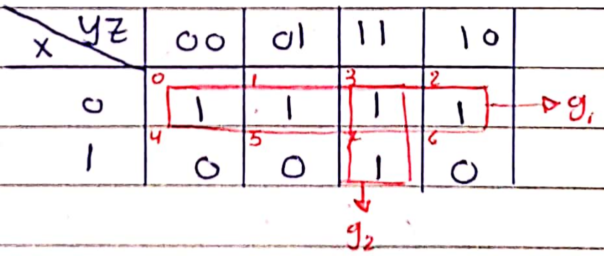


x \ yz	00	01	11	10
0	1	0	1	1
1	1	0	0	1

$\boxed{c}$ $F = \Sigma(0, 1, 2, 3, 7)$

7 → 3V

$F = \bar{x} + yz$

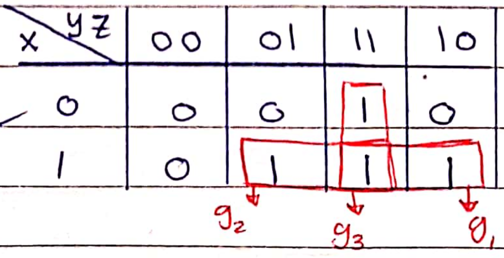


x \ yz	00	01	11	10
0	1	1	1	1
1	0	0	1	0

$\boxed{d}$ $F = \Sigma(3, 5, 6, 7)$

7 → 3V

$F = xy + xz + yz$ ✓



x \ yz	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Q (3-2):- [a] $F = \Sigma(0,1,5,7)$

$7 \rightarrow 3v$

$F = \bar{x}\bar{y} + xz$

x \ yz	00	01	11	10
0	1	1	0	0
1	0	1	1	0

g_1

g_2

[b] $F = \Sigma(1,2,3,6,7)$

g_2

$7 \rightarrow 3v$

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

x \ yz	00	01	11	10
0	0	1	1	1
1	0	0	1	1

g_1

Q(3-3):-

[a] $F = \underset{11}{xy} + \underset{000}{\bar{x}\bar{y}\bar{z}} + \underset{010}{\bar{x}y\bar{z}}$

$F = \bar{x}\bar{z} + xy$

x \ yz	00	01	11	10
0	1	0	0	1
1	0	0	1	1

g_1

g_2

[b] $\underset{00}{\bar{x}\bar{y}} + \underset{11}{yz} + \underset{010}{\bar{x}y\bar{z}}$

$F = \bar{x} + yz$

x \ y	00	01	11	10
0	1	1	1	1
1	0	0	1	0

g_1

g_2

[c] $F = \underset{01}{\bar{A}B} + \underset{10}{B\bar{C}} + \underset{00}{\bar{B}\bar{C}}$

$F = \bar{A}B + \bar{C}$

g_2

A \ Bc	00	01	11	10
0	1	0	1	1
1	1	0	0	1

g_1

!!! (!-_-)

✓ **Q3-4)** Simplify the following Boolean functions, using x maps:

(a) $F(x,y,z)=\sum(2,3,6,7)$ (b) $F(A,B,C,D)=\sum(4,6,7,15)$

(c) $F(A,B,C,D)=\sum(3,7,11,13,14,15)$ (d) $F(w,x,y,z)=\sum(2,3,12,13,14,15)$

Solution:

(a)

x\yz	00	01	11	10
0				
1				

(b)

AB\CD	00	01	11	10
00				
01				
11				
10				

(c)

AB\CD	00	01	11	10
00				
01				
11				
10				

(d)

wx\yz	00	01	11	10
00				
01				
11				
10				

✓ **Q3-5)** Simplify the following Boolean functions, using four –variable maps:

(a) $F(w,x,y,z)=\sum(1,4,5,6,12,14,15)$ *(b) $F(A,B,C,D)=\sum(0,1,2,4,5,7,11,15)$

* (c) $F(w,x,y,z)=\sum(2,3,10,11,12,13,14,15)$ (d) $F(A,B,C,D)=\sum(0,2,4,5,6,7,8,10,13,15)$

Solution:

(a)

wx\yz	00	01	11	10
00				
01				
11				
10				

***(b)**

AB\CD	00	01	11	10
00				
01				
11				
10				

***(c)**

wx\yz	00	01	11	10
00				
01				
11				
10				

(d)

AB\CD	00	01	11	10
00				
01				
11				
10				

✓ **Q3-6)** Simplify the following Boolean functions, using four –variable maps:

(a) $A'B'C'D' + AC'D' + B'CD' + A'BCD + BC'D$.

(b) $x'z+w'xy'+w(x'y+xy')$

Solution:

(a)

AB\CD	00	01	11	10
00				
01				
11				
10				

(b) مهم

wx\yz	00	01	11	10
00				
01				
11				
10				

Q(3-4):- [a] $F = \Sigma(2,3,6,7)$

$7 \rightarrow 3V$

$F = y$

$x \backslash yz$	00	01	11	10
0	0	0	1	1
1	0	0	1	1

$\rightarrow g_1$

[b] $F_{ABCD} = \Sigma(4,6,7,15)$

$15 \rightarrow 4V$

$F = \bar{A}B\bar{D} + BCD$

$AB \backslash CD$	00	01	11	10
00	0	0	0	0
01	1	0	1	1
11	0	0	1	0
10	0	0	0	0

g_1

g_2

[c] $F_{ABCD} = \Sigma(3,7,11,13,14,15)$

$15 \rightarrow 4V$

$F = CD + ABD + ABC$

$AB \backslash CD$	00	01	11	10
00			1	
01			1	
11		1	1	1
10			1	

$g_2 \leftarrow$ $\rightarrow g_1$

g_3

[d] $F_{wxyz} = \Sigma(2,3,12,13,14,15)$

$15 \rightarrow 4V$

$F = wx + \bar{w}\bar{x}y$

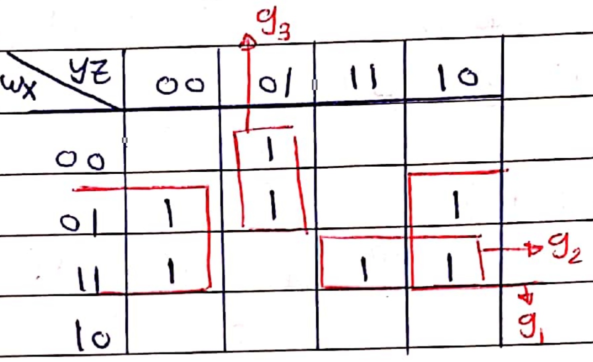
$wx \backslash yz$	00	01	11	10
00			1	1
01				
11	1	1	1	1
10				

$\rightarrow g_1$

$\rightarrow g_2$

Q(3-5):- [a] $F_{wxyz} = \Sigma(1,4,5,6,12,14,15)$

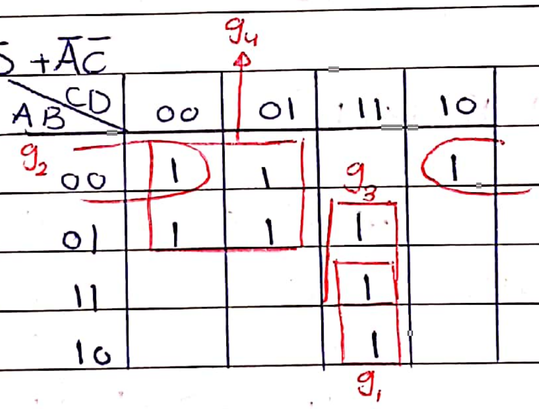
$F = x\bar{z} + wxy + \bar{w}\bar{y}z$



wx \ yz	00	01	11	10
00		1		
01	1	1		1
11	1		1	1
10				

[b] $F_{ABCD} = \Sigma(0,1,2,4,5,7,11,15)$

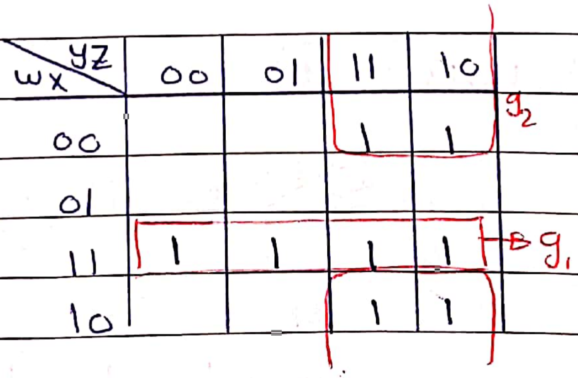
$F = ACD + BCD + \bar{A}\bar{B}\bar{D} + \bar{A}\bar{C}$



AB \ CD	00	01	11	10
00	1	1		1
01	1	1	1	
11			1	
10			1	

[c] $F_{wxyz} = \Sigma(2,3,10,11,12,13,14,15)$

$F = wx + \bar{x}y$



wx \ yz	00	01	11	10
00			1	1
01				
11	1	1	1	1
10			1	1

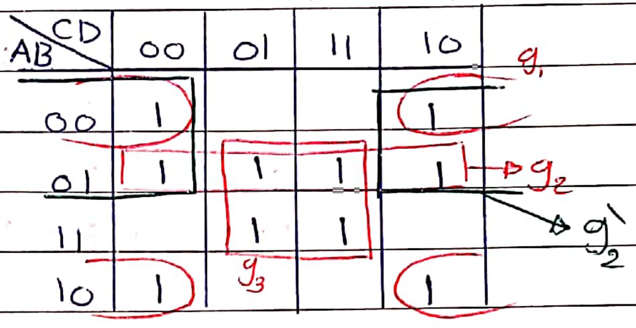
$\boxed{d}$ $F_{ABCD} = \Sigma(0,2,4,5,6,7,8,10,13,15)$

$F = \bar{B}\bar{D} + \bar{A}B + BD$

or $= \bar{B}\bar{D} + \bar{A}\bar{D} + BD$

*بدل g_2 بـ g_2' فيطلع $\bar{A}\bar{D}$

بَدَل $\bar{A}B$ (حل آخر)



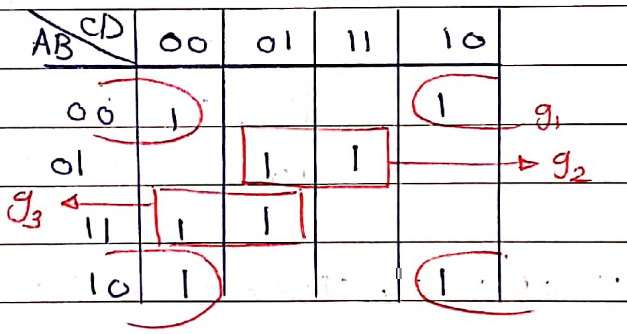
AB \ CD	00	01	11	10
00	1			1
01	1	1	1	1
11		1	1	
10	1			1

Q(3-6):-

$\boxed{a}$ $\bar{A}\bar{B}\bar{C}\bar{D} + A\bar{C}\bar{D} + \bar{B}C\bar{D} + \bar{A}BCD + B\bar{C}D$

0000 100 010 0111 101

$F = \bar{B}\bar{D} + \bar{A}BD + AB\bar{C}$



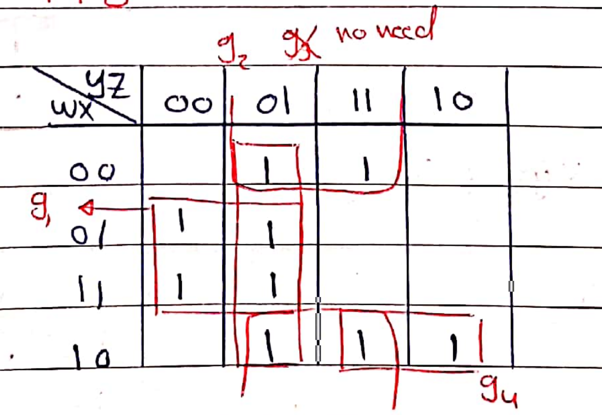
AB \ CD	00	01	11	10
00	1			1
01		1	1	
11	1	1		
10	1			1

$\boxed{b}$ $\bar{x}z + \bar{w}x\bar{y} + w(\bar{x}y + x\bar{y})$

$\bar{x}z + \bar{w}x\bar{y} + w\cancel{y}\bar{x}y + wx\bar{y}$ (ANDing → ORing)

01 010 101 110

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



wx \ yz	00	01	11	10
00		1	1	
01	1	1		
11	1	1		
10		1	1	1

✓ **Q3-7)** Simplify the following Boolean functions, using four –variable maps:

(a) $w'z + xz + x'y + wx'z$ * (b) $B'D + A'BC' + AB'C + ABC'$

(c) $AB'C + BCD + B'C'D' + ACD' + A'B'C + A'BC'D$

* (d) $wxy + yz + xy'z + x'y$.

Solution:

(a)

wx\yz	00	01	11	10
00				
01				
11				
10				

* (b)

AB\CD	00	01	11	10
00				
01				
11				
10				

(c)

AB\CD	00	01	11	10
00				
01				
11				
10				

* (d)

wx\yz	00	01	11	10
00				
01				
11				
10				

✓ **Q3-8)** find the minterms of the following Boolean expression by first plotting each function in a map:

(a) $xy + yz + xy'z$ (b) $C'D + ABC' + ABD' + A'B'D$

* (c) $wxy + x'z' + w'xz$

Solution:

(a)

x\yz	00	01	11	10
0				
1				

(b)

AB\CD	00	01	11	10
00				
01				
11				
10				

* (c)

wx\yz	00	01	11	10
00				
01				
11				
10				

!!! **Q3-11)** Simplify the following Boolean functions, using five –variable maps:

(a) $F(A,B,C,D,E) = \sum(0,1,4,5,16,17,21,25,29)$.

(b) $F = A'B'CE' + A'B'C'D' + B'D'E' + B'CD' + CDE' + BDE'$

Solution:

(a) $F(A,B,C,D,E) = \sum(0,1,4,5,16,17,21,25,29)$

A=0

BC\DE	00	01	11	10
00				
01				
11				
10				

A=1

BC\DE	00	01	11	10
00				
01				
11				
10				

✓ **Q3-7)** Simplify the following Boolean functions, using four –variable maps:

(a) $w'z + xz + x'y + wx'z$ ✱ (b) $B'D + A'BC' + AB'C + ABC'$

(c) $AB'C + BCD + B'C'D' + ACD' + A'B'C + A'BC'D$

✱ (d) $wxy + yz + xy'z + x'y$.

Solution:

(a)

wx\yz	00	01	11	10
00				
01				
11				
10				

✱ (b)

AB\CD	00	01	11	10
00				
01				
11				
10				

(c)

AB\CD	00	01	11	10
00				
01				
11				
10				

✱ (d)

wx\yz	00	01	11	10
00				
01				
11				
10				

✓ **Q3-8)** find the minterms of the following Boolean expression by first plotting each function in a map:

(a) $xy + yz + xy'z$ (b) $C'D + ABC' + ABD' + A'B'D$

✱ (c) $wxy + x'z' + w'xz$

Solution:

(a)

x\yz	00	01	11	10
0				
1				

(b)

AB\CD	00	01	11	10
00				
01				
11				
10				

✱ (c)

wx\yz	00	01	11	10
00				
01				
11				
10				

!!! **Q3-11)** Simplify the following Boolean functions, using five –variable maps:

(a) $F(A,B,C,D,E) = \sum(0,1,4,5,16,17,21,25,29)$.

(b) $F = A'B'CE' + A'B'C'D' + B'D'E' + B'CD' + CDE' + BDE'$

Solution:

(a) $F(A,B,C,D,E) = \sum(0,1,4,5,16,17,21,25,29)$

A=0

BC\DE	00	01	11	10
00				
01				
11				
10				

A=1

BC\DE	00	01	11	10
00				
01				
11				
10				

مش مطلوب

(b) $F = A'B'CE' + A'B'C'D' + B'D'E' + B'CD' + CDE' + BDE'$

A=0

BC\DE	00	01	11	10
00				
01				
11				
10				

A=1

BC\DE	00	01	11	10
00				
01				
11				
10				

√ **Q3-15)** Simplify the following Boolean functions F, together with the don't-care conditions d, and then express the simplified function in sum of minterms:

*(a) $F(x,y,z) = \sum(0,1,2,4,5) \quad ,d(x,y,z) = \sum(3,6,7)$.

(b) $F(A,B,C,D) = \sum(0,6,8,13,14) \quad ,d(A,B,C,D) = \sum(2,4,10)$.

*(c) $F(A,B,C,D) = \sum(1,3,5,7,9,15) \quad ,d(A,B,C,D) = \sum(4,6,12,13)$.

Solution:

*(a)

x\yz	00	01	11	10
0				
1				

(b)

AB\CD	00	01	11	10
00				
01				
11				
10				

*(c)

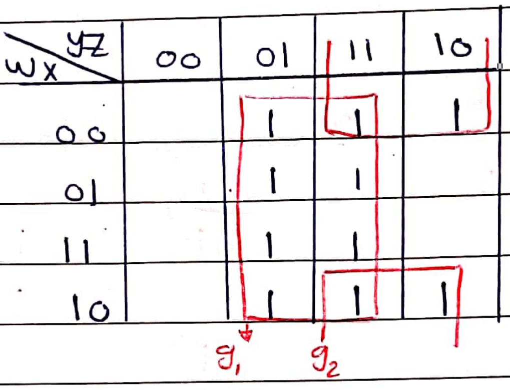
AB\CD	00	01	11	10
00				
01				
11				
10				

Q(3-7):-

[a] $\bar{w}z + xz + \bar{x}y + w\bar{x}z$

01 11 01 101

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

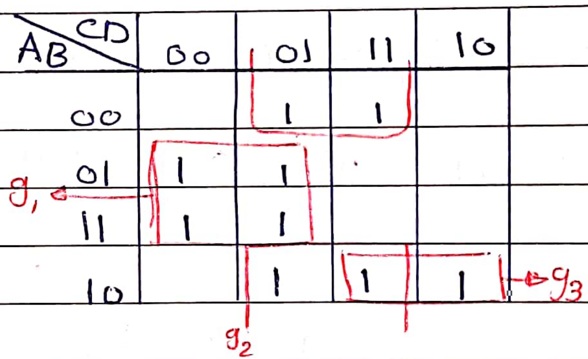


wx \ yz	00	01	11	10
00		1	1	1
01		1	1	
11		1	1	
10		1	1	1

[b] $\bar{B}D + \bar{A}B\bar{C} + A\bar{B}C + AB\bar{C}$

01 010 101 110

$F = B\bar{C} + \bar{B}D + A\bar{B}C$

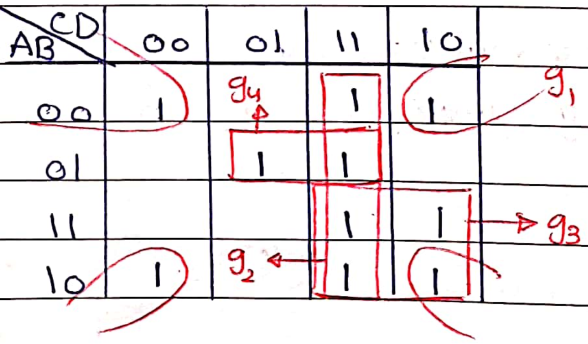


AB \ CD	00	01	11	10
00		1	1	
01	1	1		
11	1	1		
10		1	1	1

[c] $A\bar{B}C + BCD + \bar{B}\bar{C}\bar{D} + AC\bar{D} + \bar{A}\bar{B}C + \bar{A}B\bar{C}D$

101 111 000 110 001 0101

$F = \bar{B}\bar{D} + CD + AC + \bar{A}BD$

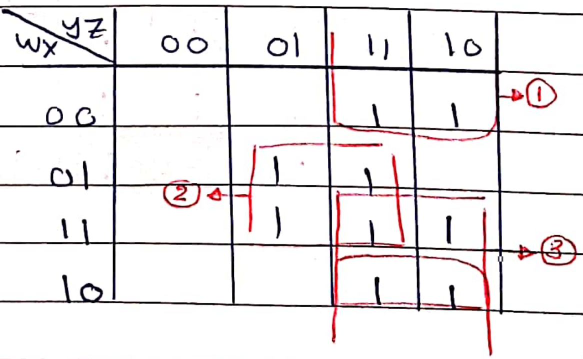


AB \ CD	00	01	11	10
00	1		1	1
01		1	1	
11			1	1
10	1		1	1

[d] $wxy + yz + x\bar{y}z + \bar{x}y$

111 11 101 01

$F = xz + \bar{x}y + wy$



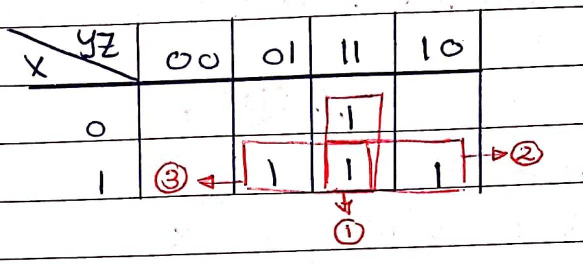
wx \ yz	00	01	11	10
00			1	1
01		1	1	
11		1	1	1
10			1	1

Q(3-8):-

[a] $xy + yz + x\bar{y}z$

11 11 101

$F = yz + xz + xy$



x \ yz	00	01	11	10
0			1	
1		1	1	1

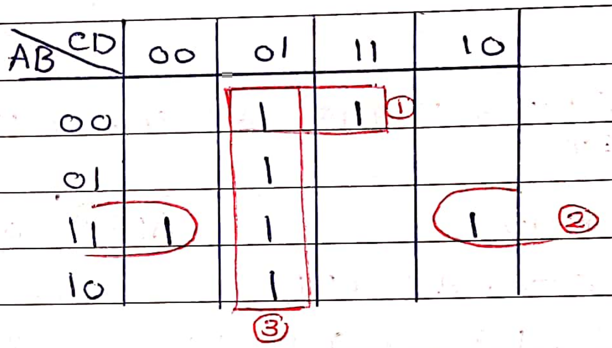
$F = \Sigma(3,5,6,7)$

[b] $\bar{C}D + AB\bar{C} + AB\bar{D} + \bar{A}\bar{B}D$

01 110 110 001

$F = \bar{C}D + \bar{A}\bar{B}D + AB\bar{D}$

$F = \Sigma(1,3,5,9,12,13,14)$

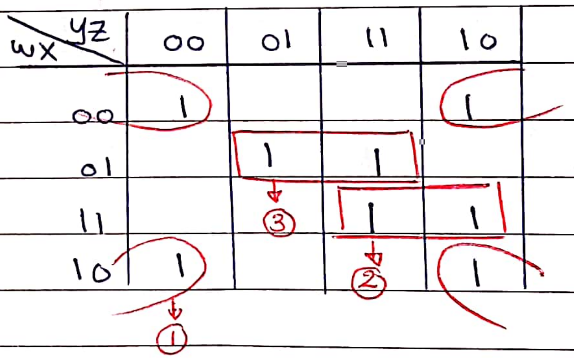


AB \ CD	00	01	11	10
00		1	1	
01		1		
11	1	1		1
10		1		

[c] $wxy + \bar{x}\bar{z} + \bar{w}xz$

111 00 010

$F = \Sigma(0,2,5,7,8,10,14,15)$



wx \ yz	00	01	11	10
00	1			1
01		1	1	
11			1	1
10	1			1

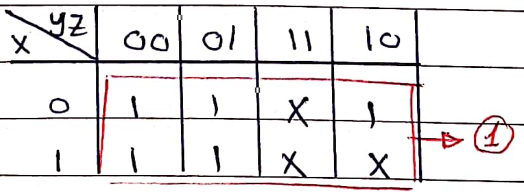
Q(3-15):-

[a] $\Sigma(0,1,2,4,5)$, $d(x,y,z) = \Sigma(3,6,7)$

$7 \rightarrow 3V$

$F = 1$

$= \Sigma(0,1,2,3,4,5,6,7)$



x \ yz	00	01	11	10
0	1	1	X	1
1	1	1	X	X

[b] Σ(0,6,8,13,14), d(A,B,C,D) = Σ(2,4,10)

14 → 4V

$F = \bar{B}\bar{D} + AB\bar{C}D + C\bar{D}$

F = Σ(0,2,6,8,10,13,14)

AB \ CD	00	01	11	10
00	1			X
01	X			1
11		1		1
10	1			X

[C] Σ(1,3,5,7,9,15), dΣ(4,6,12,13)
ABCD

15 → 4V

$F = \bar{C}D + \bar{A}D + BD$

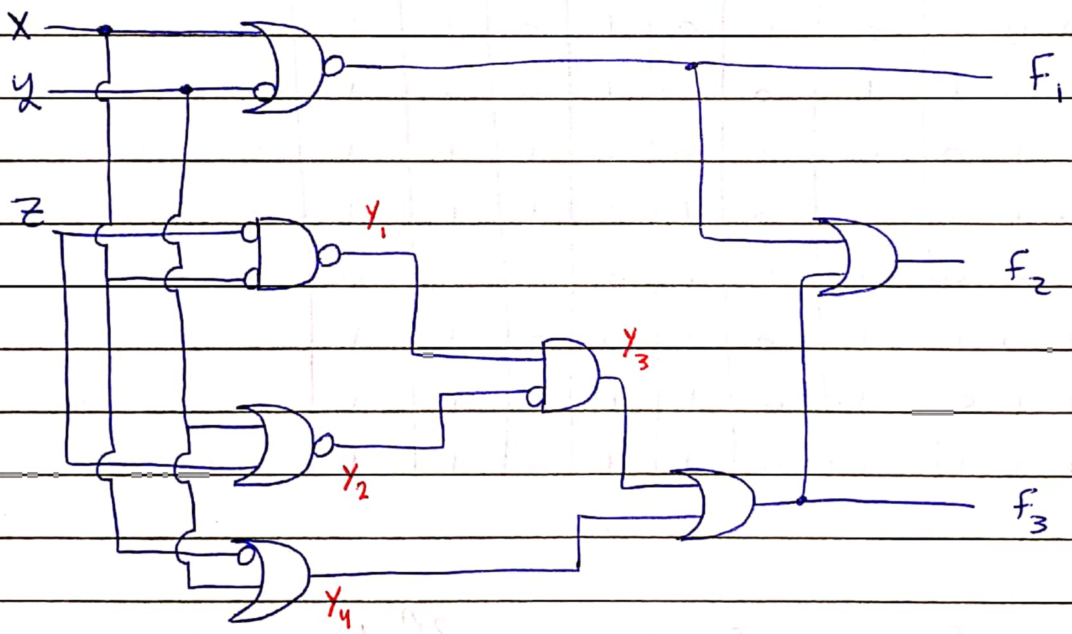
F = Σ(1,3,5,7,9,13,15)

AB \ CD	00	01	11	10
00		1	1	
01	X	1	1	X
11	X	X	1	
10		1		

Chapter 4 (Combinational Logic)

* Analysis of Combinational cct :
  - Given : cct
  - Required : Bolean expression of output
  - steps : 1) Truth Table.
    2) Verify.

* Example : analyse the following cct, then find # of unique function.



?

Solu :-

* لمن تشوف cct صعبة افرض اقترانات زيادة منك عند كل
مجموعة gates صعبة للتسهيل.
* ثم طلع الـ B-expression لكل الإقترانات.
* ثم تحقق باستخدام الـ Truth-T وأوجد المطلوب.
* قبل لا تطلع الـ B-expression شوف المدخلات والمخرجات
وارجع رجوع من الـ output للـ input وإفهم #

1) $F_1 = \overline{(X + 5)}$

$Y_1 = \overline{(\bar{X} . \bar{Z})}$

$Y_2 = \overline{(Y + Z)}$

$Y_3 = (Y_1 . \bar{Y_2})$

$Y_4 = (\bar{X} + Y)$

$F_3 = (Y_3 + Y_4)$

$F_2 = (F_1 + F_3)$

2)

x	y	z	Y_1	Y_2	Y_3	Y_4	f_1	F_3	F_2
0	0	0	0	1	0	1	0	1	1
0	0	1	1	0	1	1	0	1	1
0	1	0	0	0	0	1	1	1	1
0	1	1	1	0	1	1	1	1	1
1	0	0	1	1	0	0	0	0	0
1	0	1	1	0	1	0	0	1	1
1	1	0	1	0	1	1	0	1	1
1	1	1	1	0	1	1	0	1	1

* Rule : # of unique Function = $(2)^{(2)^n}$ / n = # of variables (X, Y, Z)

$n = 3$

$\therefore$ # of u.f $= 2^{2^3} = 2^8 = 256$

↓

عدد ال cct's المكافئة لهذه ال cct .

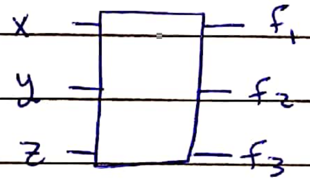
* Design of Combinational cct :

- Given : Description.
- Required : cct.
- Steps : 1) Define input's and output's
  2) Define Truth-T based on Description.
  3) Obtain Boolean-expression of output by K-map
  4) Draw the cct.

* Example : Design a 3-bits binary to gray-code Converter?

Solu :

1)



3)

x \ yz	00	01	11	10
0	0	0	0	0
1	1	1	1	1

$F_1 = x$

2)

				MSB		LSB
	x	y	z	F1	F2	F3
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	1	1	0	0

Xor

MSB

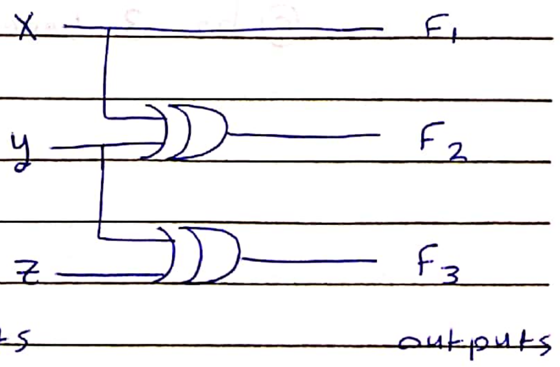
x \ yz	00	01	11	10
0	0	1	0	1
1	0	1	0	1

$F_3 = y \oplus z$

x \ yz	00	01	11	10
0	0	0	1	1
1	1	1	0	0

$F_2 = x \oplus y$

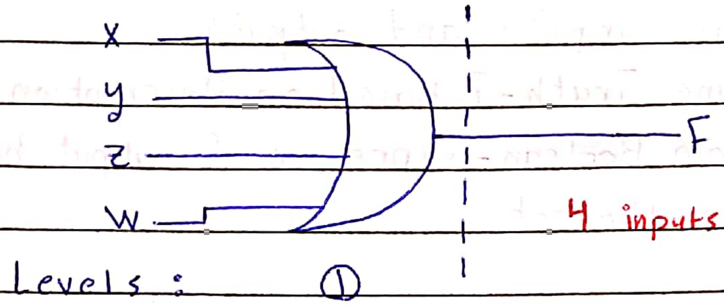
4)



* Example : F = X + Y + W + Z , Design the cct if only you have 2 - inputs (OR) gates ?

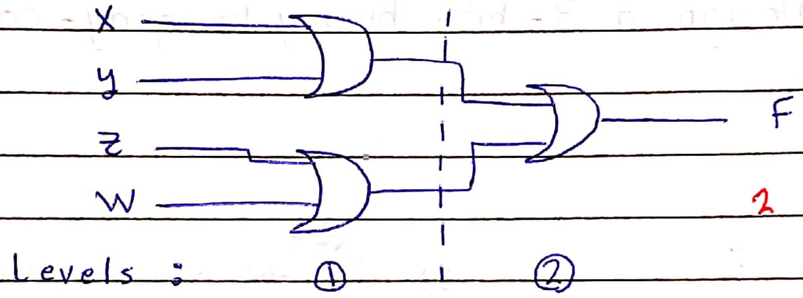
Soln :

a)



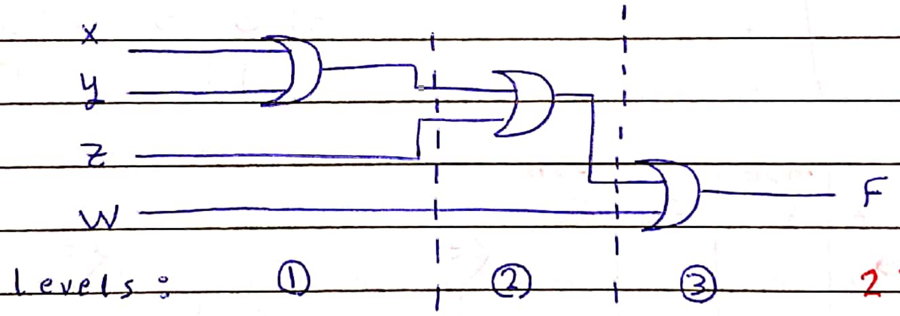
4 inputs oR gate

b)



2 inputs oR gate

c)



2 inputs oR gate

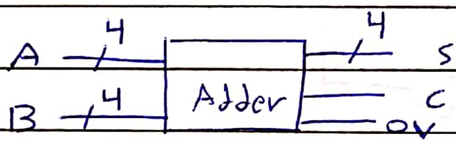
But : ⓐ has 1-Level so 1 delay (best one)
ⓑ has 2-level so 2 delay
ⓒ has 3-Level so 3 delay (boor one)
↓
من ناحية السرعة

▸ Subject : .... / /

* Adder / Subtracter :

* Example : Design a 4 Bits Adder using 2's complement ?

Solu :



	c_3	c_2	c_1		
	a_3	a_2	a_1	a_0	
	b_3	b_2	b_1	b_0	+
C_4	S_3	S_2	S_1	S_0	
الناتج الباقي	4rth	3rd	2nd	1st stage	

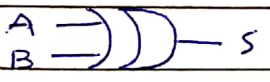
* 1st Stage : $a_0 + b_0 = S_0 , C_1$ [2 inp & 2 outp]
Called Half Adder (HF) .

* All rest stages : $a_n + b_n + C_{in} = S_n , C_{out}$ [3 in & 2 out]
called Full Adder (FA) For each one .

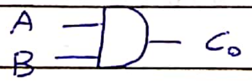
1) HA :

A	B	S	C_o
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

∴ $S = A \oplus B$



$C_o = A . B$



2) FA :

	A	B	C_i	S	C_o
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

A \ BC_i	00	01	11	10
0	0	①	0	①
1	①	0	①	0

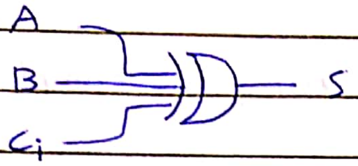
A \ BC_i	00	01	11	10
0	0	0	1	0
1	0	1	1	1

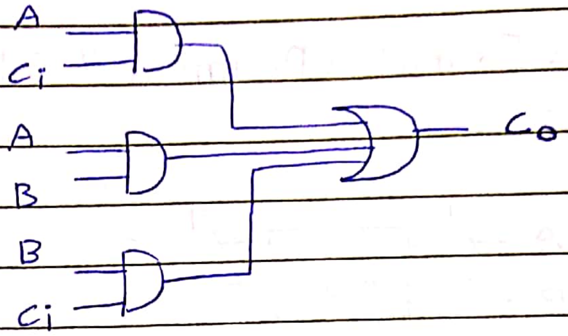
$S = ABC_i + \bar{A}\bar{B}C_i$
$+ A\bar{B}\bar{C}_i + \bar{A}B\bar{C}_i$
$(S = A \oplus B \oplus C_i)$

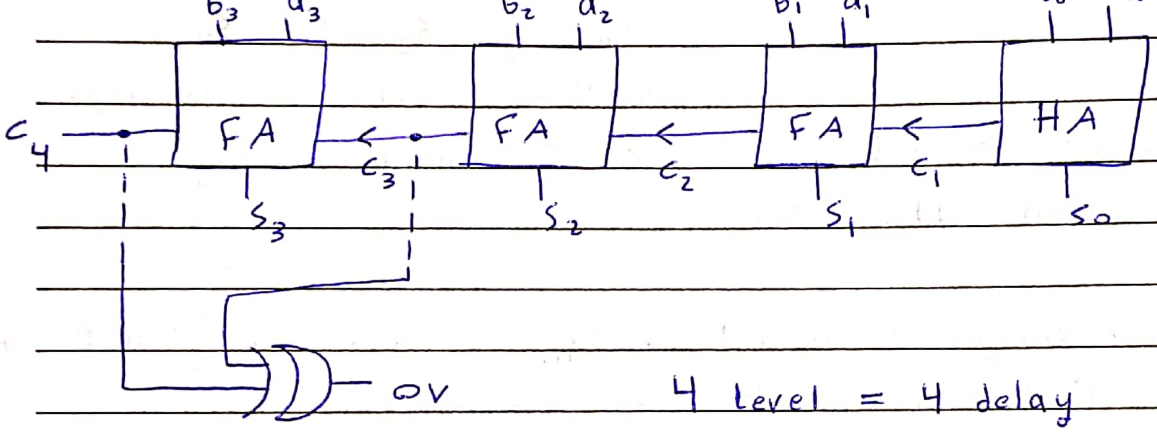
$\left(C_o = AC_i + AB + BC_i \right)$

S T A R S N O T E B O O K

$S = \Sigma(1,2,4,7)$ $C_o = \Sigma(3,5,6,7)$



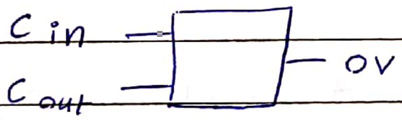




4 level = 4 delay

This cct calld : 4 Bit Binary Ripple Carry Adder .

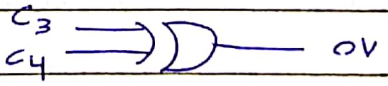
3) ov :



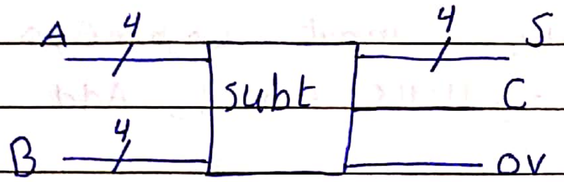
, Cin and Cout from least stage . so C_3 & C_4 .

C_3	C_4	ov
0	0	0
0	1	1
1	0	1
1	1	0

$\therefore (ov = C_3 \oplus C_4)$



* Example : Design a 4-bits subtracter using 2's complement ?



c_3 c_2 c_1

a_3 a_2 a_1 a_0

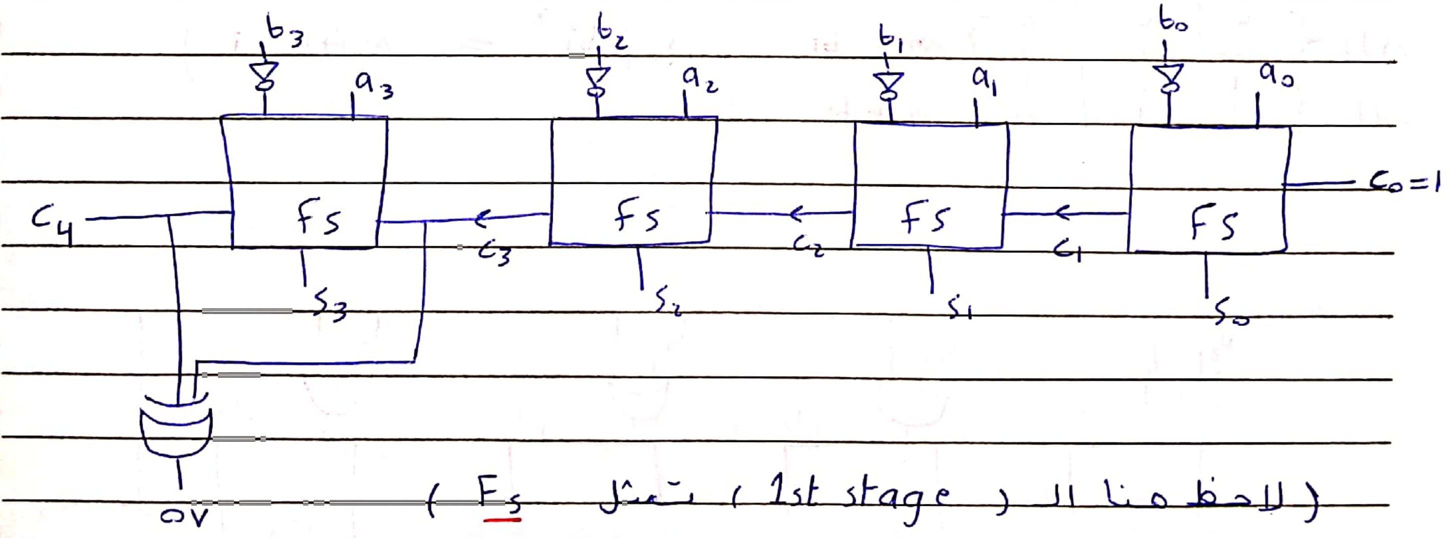
~~b_3 b_2 b_1 b_0~~ 1's

$\bar{b}_3$ $\bar{b}_2$ $\bar{b}_1$ $\bar{b}_0$

1 +

S_3 S_2 S_1 S_0

إضافة الواحد لتحويل
الرقم إلى 2's
تمثل إضافة مدخل
جديد وهو C_0
وقيمته هي = (1)



(لاحظنا ان ال (1st stage) تمثل Fs)

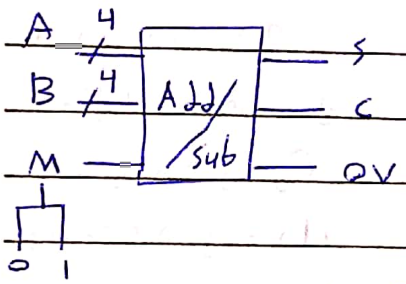
(Full subtracter)

*Example : Design a 4-bits Add / Sub using 2's complement ?

هون بفرض input جديد اسمه (M) لتحديد العملية Add ولا sub كالتالي:

$$\begin{pmatrix} M = 0 \rightarrow \text{Add} \\ M = 1 \rightarrow \text{sub} \end{pmatrix}$$

لأنه بال Add مابلزمني input جديد يعني (M=0)، أما بال Sub احتجت ال (1).

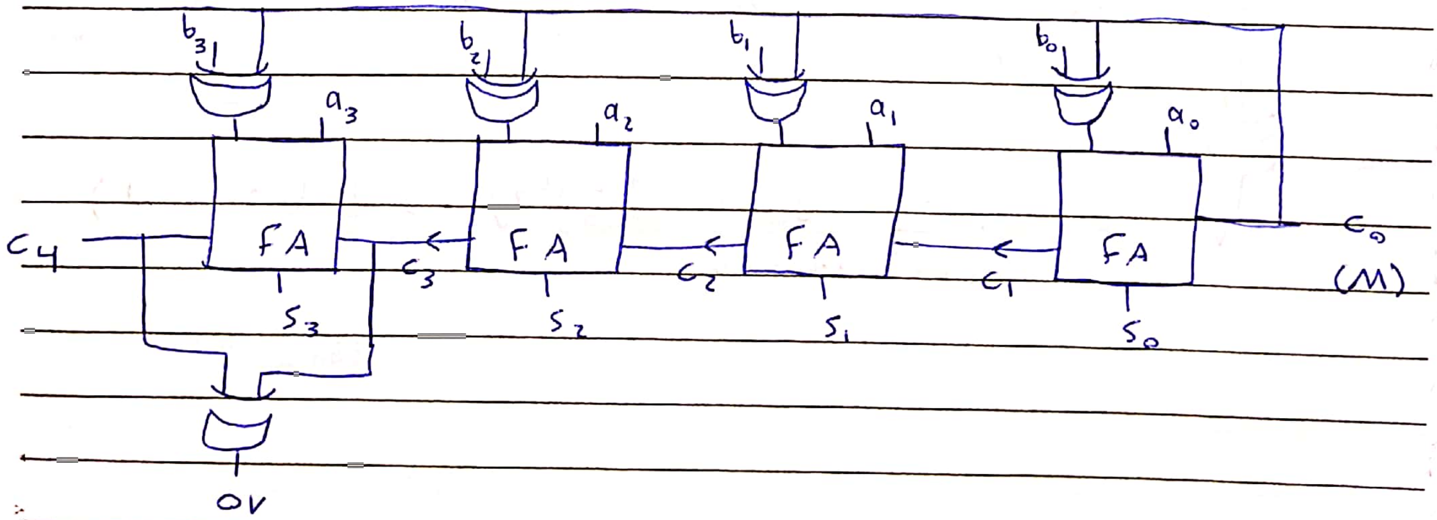


	M	Co
add	0	0
sub	1	1

$\therefore (M = C_0)$

	M	b_i	b_i^*	
Add {	0	0	0	} Same b_i
	0	1	1	
sub {	1	0	1	} Not b_i
	1	1	0	

$\therefore \left(b_i^* = M \oplus b_i \right)$



* Example :

A = 1001

B = 0101 Find S, OV , Cout ?

M = 1

Solu :

M = 1 ∴ sub by 2's

⓪ ' '

1 0 0 1

~~0 1 0 1~~ } 2's

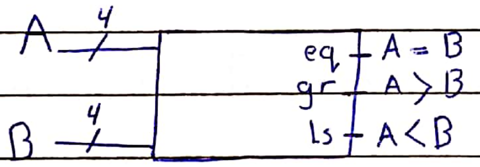
1 0 1 1

$①\underbrace{0 1 0 0}_{s}$ ∴ S = 0100

Cout = 1

ov = 1 [Cout ≠ cin]

* Example : Design a cct that Compares between 2 - unsigned numbers (4 - bits) ?



بنوخد كل وحدة ونعملها الحال

if eq → out put = 1 , other wise = 0

if gr → out put = 1 , other wise = 0

if ls → out put = 1 , other wise = 0

1) eq : A=B

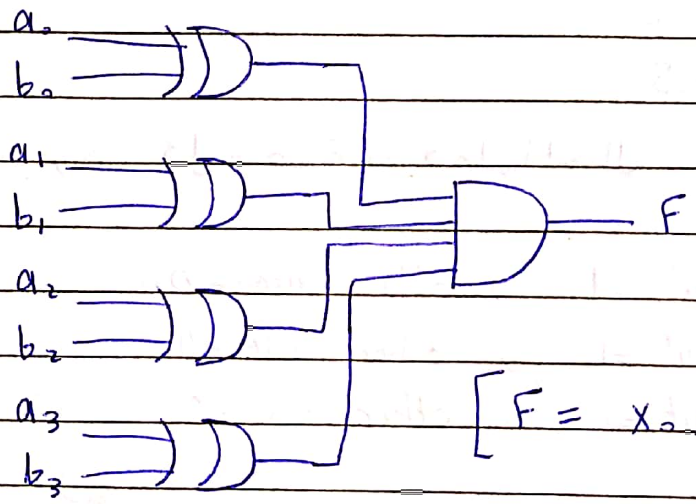
let output = Xi
(Xi : detector)
if eq true , Xi = 1
other wise , Xi = 0

a_3 a_2 a_1 a_0 → A
b_3 b_2 b_1 b_0 → B

a_i	b_i	X_i
0	0	1
0	1	0
1	0	0
1	1	1

$$\therefore X_i = (\overline{a_i + b_i})$$

to get A=B , it must be : $(a_0 = b_0)$ and $(a_1 = b_1)$
and $(a_2 = b_2)$ and $(a_3 - b_3)$



$$[F = X_0 X_1 X_2 X_3 = eq]$$

if F = 1 → A = B
other wise → A ≠ B

2) gr : A > B

$A \rightarrow a_3 \quad a_2 \quad a_1 \quad a_0$

$B \rightarrow b_3 \quad b_2 \quad b_1 \quad b_0$

ai	bi	Xi
0	0	0
0	1	0
1	0	1
1	1	0

if gr , $X_i = 1$

other wise, $X_i = 0$

to get $a_0 > b_0$, let $a_0 = 1$ Then b_0 must be $= 0$

$\therefore \overline{b_0}$ as sop so get :

$$\left(gr = a_3 \cdot \overline{b_3} + X_3\, a_2 \cdot \overline{b_2} + X_3 X_2\, a_1 \cdot \overline{b_1} + X_3 X_2 X_1\, a_0 \cdot \overline{b_0} \right)$$

Start Comparing From left to Right

if $X_3 = 1$; ($a_3 > b_3$) then stop because you get $A < B$.

3) ls : A < B

$$\left(ls = \bar{a}_3\, b_3 + X_3\, \bar{a}_2\, b_2 + X_3 X_2\, \bar{a}_1\, b_1 + X_3 X_2 X_1\, \bar{a}_0\, b_0 \right)$$

Same as gr but it should be ($a_3 < b_3$) $\rightarrow X_3 = 1$ to stop at first term.

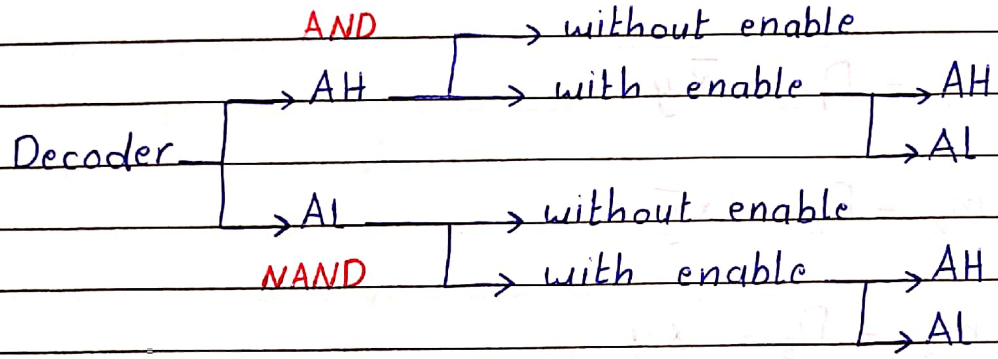
Finally :

gr	eq	ls	
0	0	1	→ less than
0	1	0	→ equal
1	0	0	→ greater
1	1	X	→ don't care

↓

eq and gr at same time مستحيل

* Decoders :

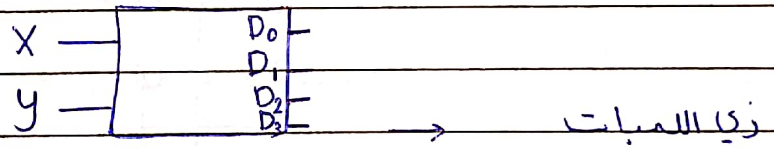


1) AH Decoder without enable :

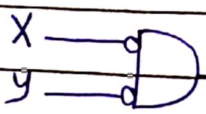
$(2^{(input)} = output)$

$(2 \rightarrow 4)$, $(3 \rightarrow 8)$, $(4 \rightarrow 16)$ ...

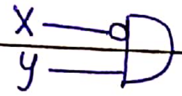
$(2 \rightarrow 4)$:



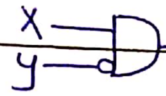
x	y	D_0	D_1	D_2	D_3	
0	0	1	0	0	0	m_0 (only D_0 on)
0	1	0	1	0	0	m_1 (only D_1 on)
1	0	0	0	1	0	m_2 (only D_2 on)
1	1	0	0	0	1	m_3 (only D_3 on)



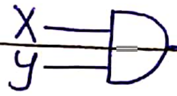
$m_0 = D_0 = \bar{x}\,\bar{y}$



$m_1 = D_1 = \bar{x}\,y$



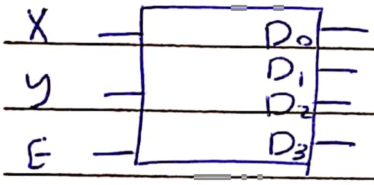
$m_2 = D_2 = x\,\bar{y}$

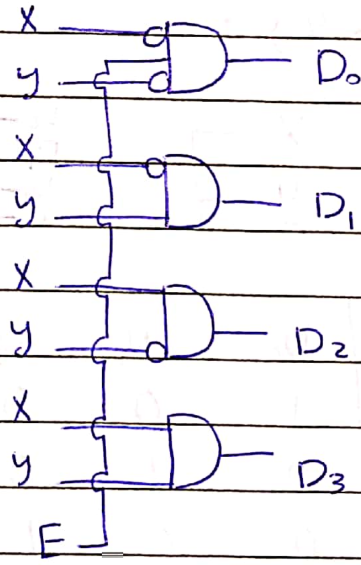


$m_3 = D_3 = x\,y$

2) Active high decoder with active high enable :

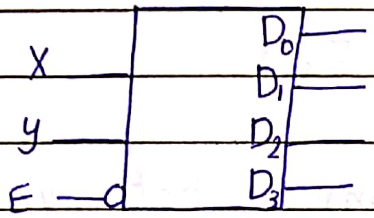
(2 → 4) :



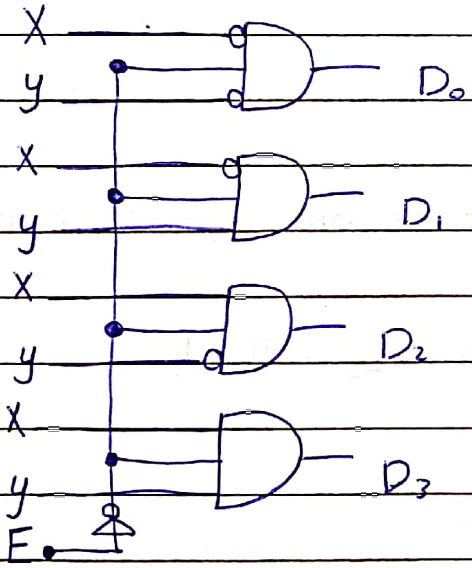


	E	x	y	D0	D1	D2	D3
ON	1	0	0	1	0	0	0
	1	0	1	0	1	0	0
	1	1	0	0	0	1	0
	1	1	1	0	0	0	1
OFF	0	x	x	0	0	0	0

3) Active high decoder with active low enable :



	E	X	y	D_0	D_1	D_2	D_3
ON	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
off	1	X	X	0	0	0	0

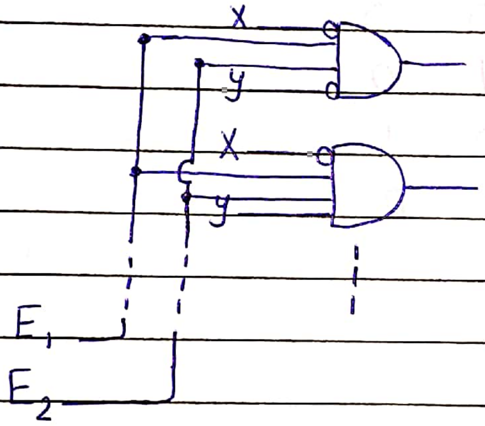


* معلومة مهمة : في حال بدك تحط أكثر من E وحدة في الـ cct عادي بتقدر وبعدة خيارات مثل :

For All (Decoder) and All (E's)

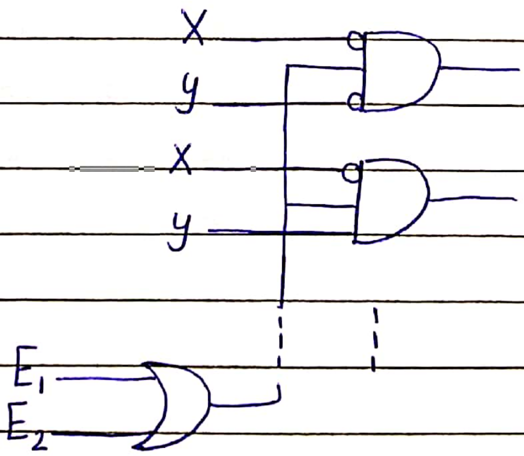
① only if E_1 and E_2 Both (on) → cct run :

فقط بشبك E_1 و E_2 مع كل الـ ANDs كالتالي :



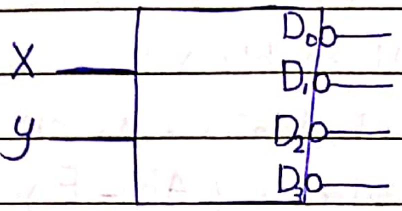
② at least on of Both must be on → cct run :

فقط اشبك التنين على OR ثم على الـ ANDs كلهم

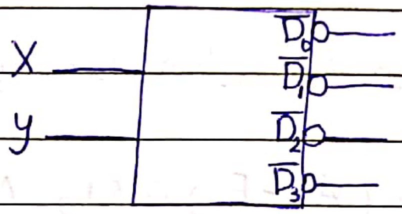


Subject : .... / /

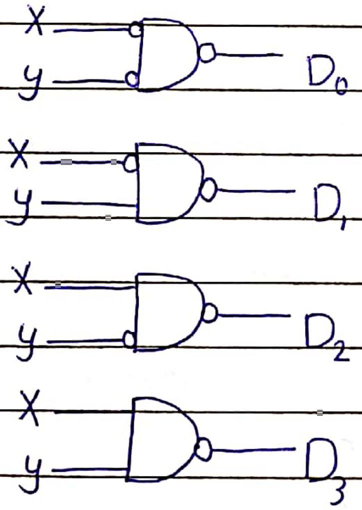
4) Active low decoder without enable :



OR



X	y	D_0	D_1	D_2	D_3	
0	0	0	1	1	1	(only D_0 off)
0	1	1	0	1	1	(only D_1 off)
1	0	1	1	0	1	(only D_2 off)
1	1	1	1	1	0	(only D_3 off)



والباقي نفس الشرح السابق...

* ملاحظات:

1) ال (AH decoder) شغال ب ANDs بينما ال (AL decoder) شغال ب NANDs

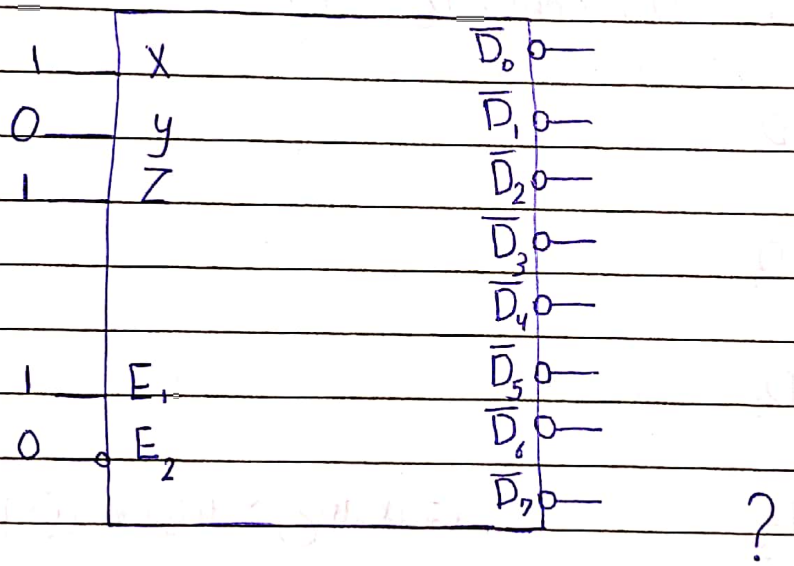
2) ال (AH decoder) مفش على ال Di فقاعات "(Pupples)" بينما ال (AL decoders) في على ال Di فقاعات.

3) ال (AH enable) شغالة طبيعي لمن تعطيها (1) ← on
ولمن تعطيها (صفر) بتكون off
بينما ال (Al enable) شغالة طبيعي لمن تعطيها (صفر) وبتكون on ولمن تعطيها (1) بتكون off .

4) (AH - E) ما عليها فقاعة أو Not ولكن (Al - E) عليها

X Examples :

1) Find outputs for cct



Solu :

input 101 = 5 output

E_1 is AH with (1) so on
E_2 is AL with (0) so on
Decoder is AL
So only $\overline{D}_5$ off :

$\overline{D}_0$ o— 1
$\overline{D}_1$ o— 1
$\overline{D}_2$ o— 1
$\overline{D}_3$ o— 1
$\overline{D}_4$ o— 1
$\overline{D}_5$ o— 0
$\overline{D}_6$ o— 1
$\overline{D}_7$ o— 1

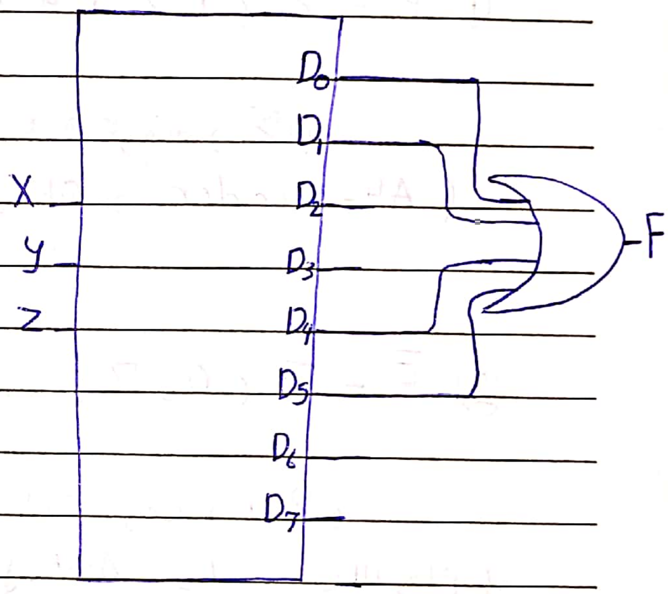
2) Implement the following function using the appropriate decoder $F = \Sigma(0,1,4,5)$?

Solu :

5 → 101 ∴ 3 variables

3 input , $2^3 = 8$ output

X	y	Z	F	
0	0	0	1	$m_0 = D_0$
0	0	1	1	$m_1 = D_1$
0	1	0	0	m_2
0	1	1	0	m_3
1	0	0	1	$m_4 = D_4$
1	0	1	1	$m_5 = D_5$
1	1	0	0	m_6
1	1	1	0	m_7



Since (SOP) is (ANDing) Then (ORing) and decoder contain ANDing , so it remains to do ORing only.

ومش داعي للجدول ع طول من $\Sigma(0,1,4,5)$

$(D_0 \; D_1 \; D_4 \; D_5) \rightarrow$ on

3) Implement $F = \Sigma(1,2,3,4,5,6)$ using appropriate decoder ?

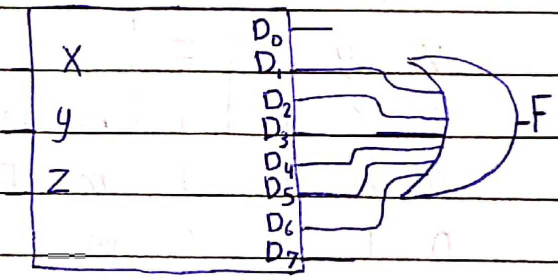
Solu :

6 = 110 , 3 input , 8 output

① $F = \Sigma(1,2,3,4,5,6)$

* لما تشوف (Σ)

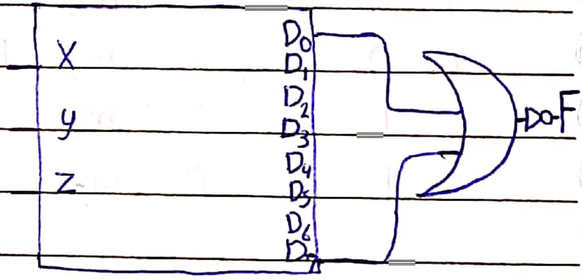
(استعمل OR و AH-decoder)



② $\overline{F} = \Sigma(0,7)$, $F = \overline{(\overline{F})}$

* لما تشوف (Bar)

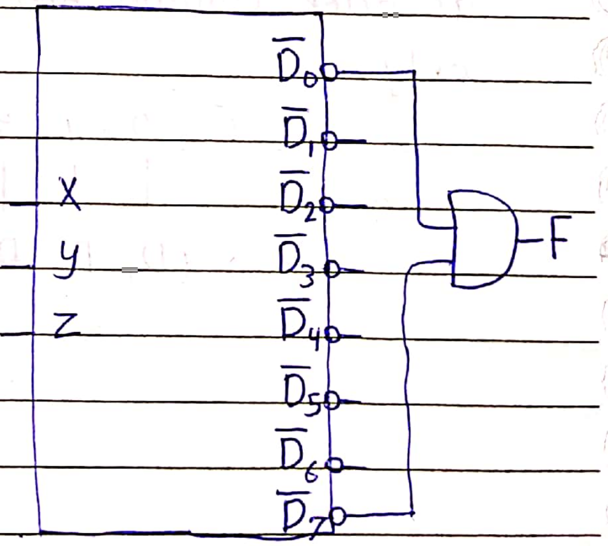
(استعمل Not واعكس الأرقام)



③ $F = \Pi(0,7)$

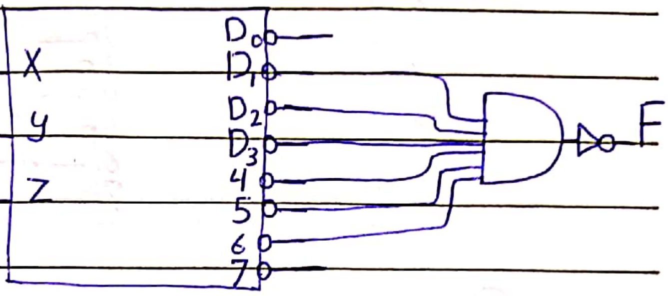
* لما تشوف (Π)

(استعمل AL-decoder واعكس الأرقام
واستعمل AND)



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

* لما تشوف (π) مع (Bar)
(استعمل Al-decoder واستعمل AND
واستعمل Not)

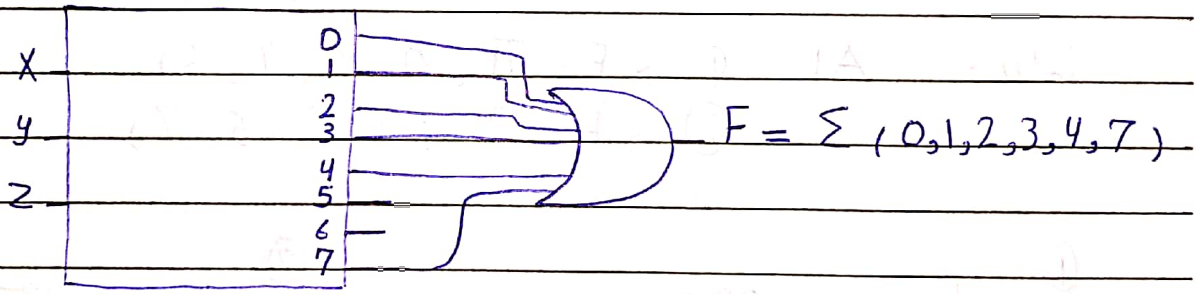


4) $F = \Sigma(0,1,2,3,4,7)$, Implement this function?

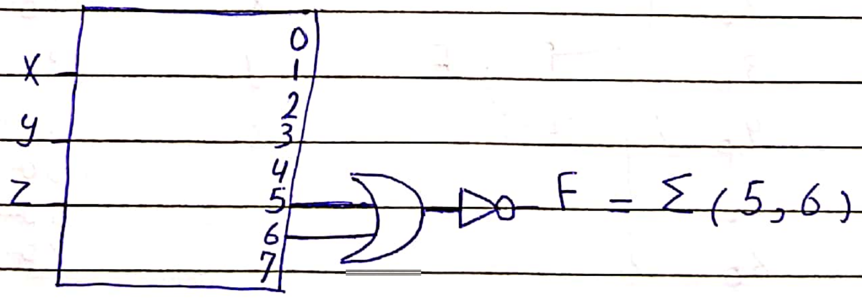
Solu:

7 = 111 , 3 input , 8 output

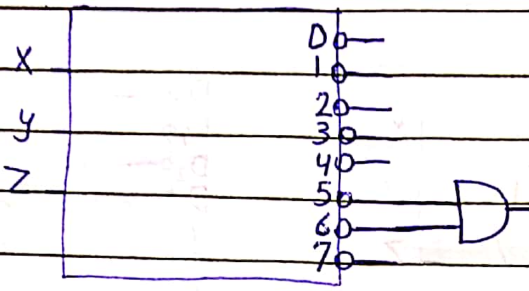
①



②

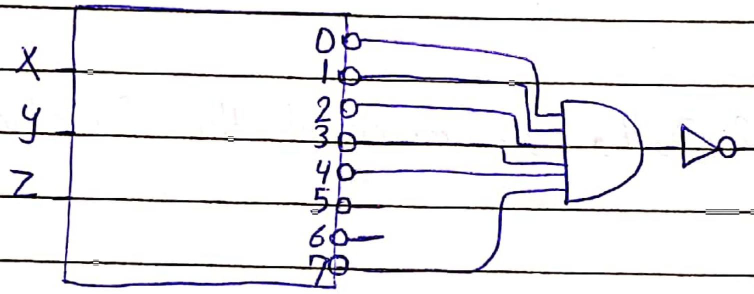


③



$F = \Pi(5,6)$

④



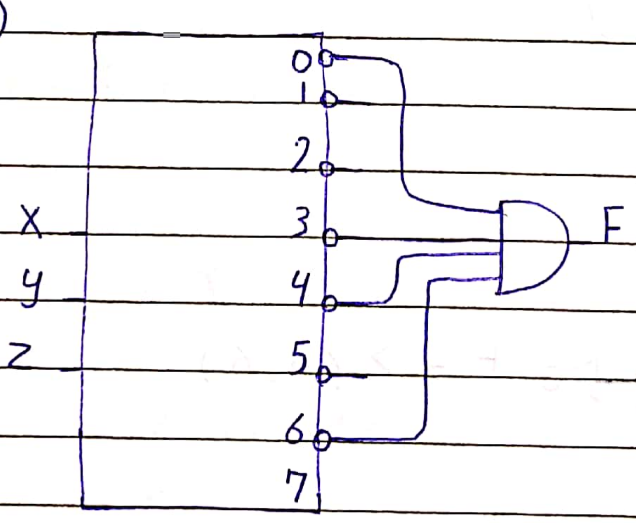
$F = \Pi(0,1,2,3,4,7)$

5) $F = \Sigma(1,2,5,7)$, using AL-decoder ?

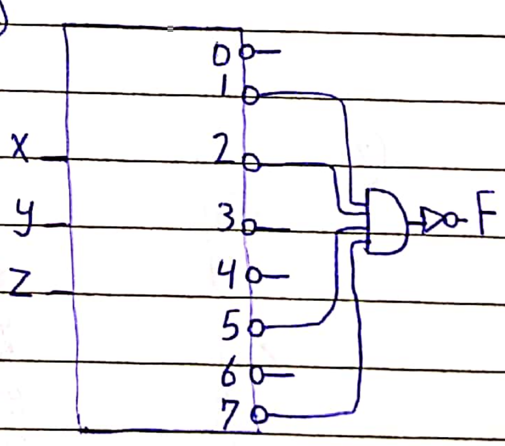
Solu: AL ① $\rightarrow F = \Pi(0,3,4,6)$

② $\rightarrow \overline{F} = \Pi(1,2,5,7)$

①



②

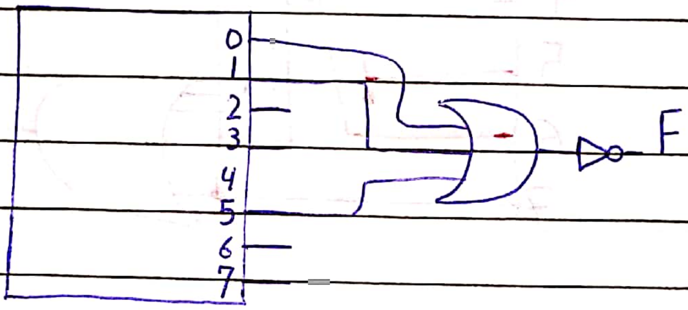


6) $F = \Pi(0,1,5)$, AH - decoder ?

Solu : ① $F = \Sigma(2,3,4,6,7)$ هنالك حلين

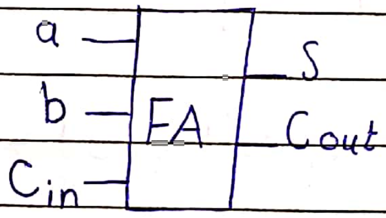
② $\bar{F} = \Sigma(0,1,5)$

②



7) Designe a Full adder using OR gates and 3x8 active high decoder ?

Solu :



a	b	Cin	S	Cout	
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	2
0	1	1	0	1	3
1	0	0	1	0	4
1	0	1	0	1	5
1	1	0	0	1	6
1	1	1	1	1	7

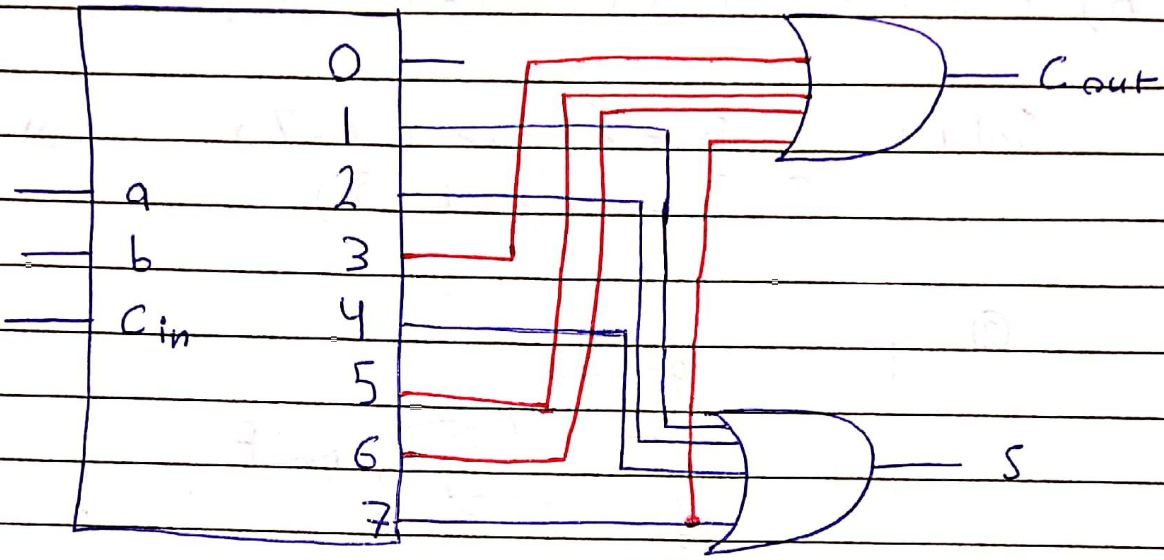
$S = \Sigma(1,2,4,7)$, $C_{out} = \Sigma(3,5,6,7)$

SOP : ANDing Then ORing

$\downarrow$

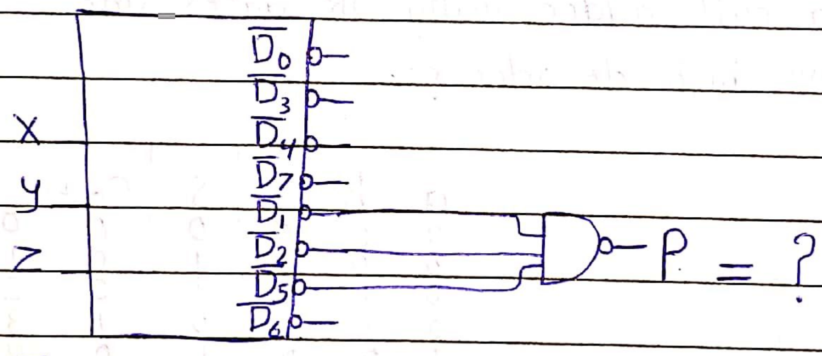
مباشرة

... يتبع



* Examples :

1)



$p = \overline{F} = \Pi(1, 2, 5)$ (وانتبه لترتيب الأرقام)

$F = \Sigma(1, 2, 5)$

* بهيك أسئلة امشي عكسي من P للمدخلات

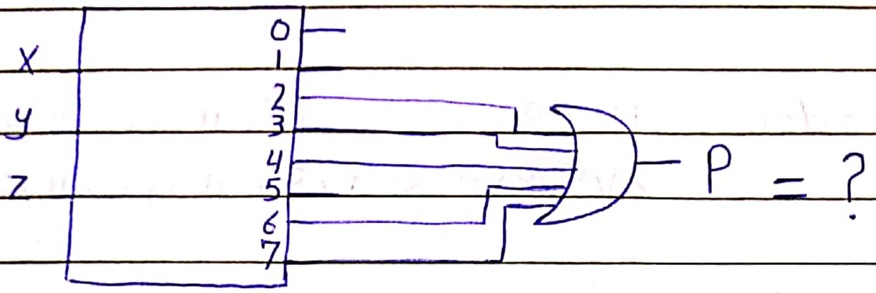
إذا شفت (Not) حط Bar وإذا شفت (AND) حط Π ، ونفس الملاحظات السابقة لكن بشكل عكسي .

* لما يكون عندك (Bar مع Π) وبدك F الحل :

يا بتثبت العملية وبتعكس الأرقام $[\Pi(0, 3, 4, 6, 7)]$

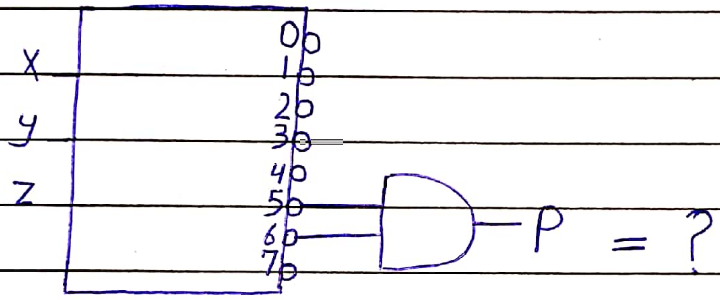
يا بتغير العملية وبتثبت الأرقام $[(1, 2, 5)]$

2)



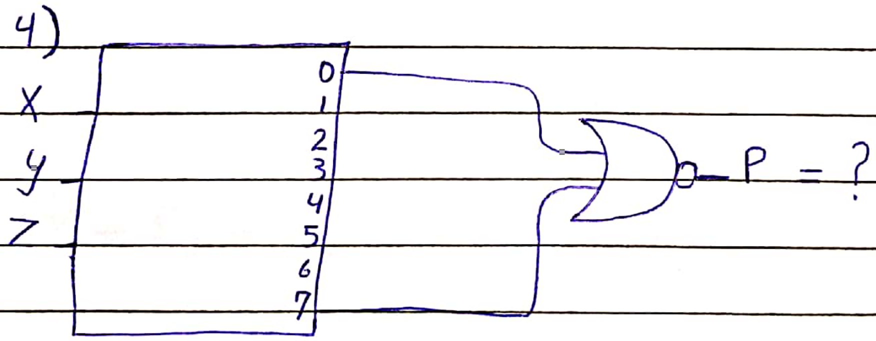
$P = \Sigma(2, 3, 4, 6, 7)$

3)



$P = \Pi(5, 6) = \Sigma(0, 1, 2, 3, 4, 7)$

4)



$P = \overline{F} = \Sigma(0, 7)$

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

* Notes :

1) AH-decoder : 1) اجمع الحدود للـ (F) باستخدام (OR)
2) اجمع الحدود للـ ($\bar{F}$) باستخدام (NOR)

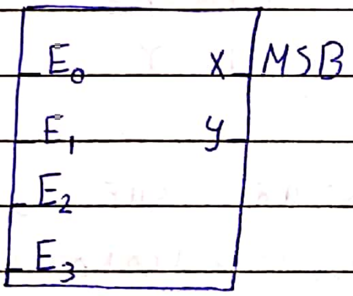
2) AL-decoder : 1) اجمع الحدود للـ (F) باستخدام (NAND)
2) اجمع الحدود للـ ($\bar{F}$) باستخدام (AND)

* Encoder :

(2 → 1) , (4 → 2) , (8 → 3) , (16 → 4)

inputs = 2^n , n = outputs

(4 → 2) :



E_0	E_1	E_2	E_3	x	y	
1	0	0	0	0	0	Pressing on (E_0) only
0	1	0	0	0	1	Pressing on (E_1) only
0	0	1	0	1	0	Pressing on (E_2) only
0	0	0	1	1	1	Pressing on (E_3) only

$$X = E_0 + E_2 \quad , \quad y = E_1 + E_3$$

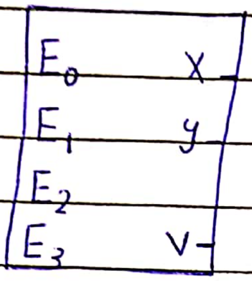
* عندنا مشكلتين هون :

١) لو ما بدي أكبس ولا كبسة ، شو قيمة X, y ؟

٢) لو بدي أكبس كبستين معاً ، شو قيمة X, y ؟

* Priority encoder :

هاي لما تكون شغالة بتعطي الأولوية للأكبر والباقي مش مهم .



	E_0	E_1	E_2	E_3	V	x	y
on	1	0	0	0	1	0	0
	X	1	0	0	1	0	1
	X	X	1	0	1	1	0
	X	X	X	1	1	1	1
off	0	0	0	0	0	X	X

الأولوية للكبسة الشغالة والأكبر منها أما الأصغر منها Don't care .

هاض لما تكون V لحالها بدون فقاعة .

$E_0E_1 \backslash E_2E_3$	00	01	11	10
00	0	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$V = E_0 + E_1 + E_2 + E_3$$

$E_0E_1 \backslash E_2E_3$	00	01	11	10
00	X	1	1	1
01	0	1	1	1
11	0	1	1	1
10	0	1	1	1

$$X = E_2 + E_3$$

E_0E_1 \ E_2E_3	00	01	11	10
00	x	1	1	0
01	1	1	1	0
11	1	1	1	0
10	0	1	1	0

$$y = (\overline{E}_2 + E_3) \cdot (E_1 + E_3)$$

$$y = E_3 + E_1 \overline{E}_2$$

* لما تكون الـ V عليها فقاعة بتكون الأولوية للقيمة الشغالة والأصفر منها أما الأكبر منها Don't care .

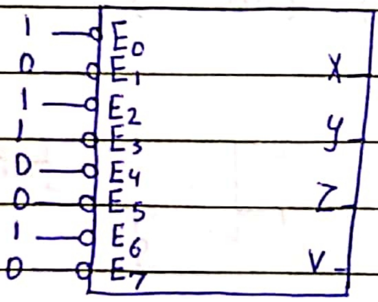
يعني فقط اعكس جهة (الأصفار) والـ (Don't care) بالجدول وعبي الـ K-map و Done

* AH (V) : دور على أكبر قيمة شغالة ، AL (V) دور على أصغر قيمة شغالة

AH-encoder : أكبر أو أصغر (1) ، AL-encoder : أكبر أو أصغر (0) .

* Examples :

1)



Find X, y, Z, V ?

Solu : AL-encoder , AH priority

∴ Largest (0) : 7 = 111 (MSB)

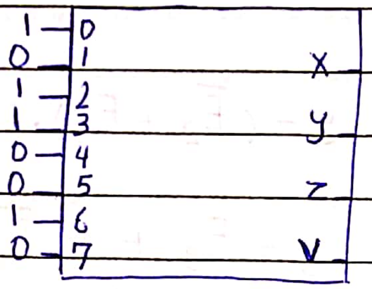
X = 1

y = 1

Z = 1

V = 1

2)



X, y, z, v ?

AH-encoder and AH priority:

Largest (1) : 6 = $\overset{MSB}{1}10$

X = 1

y = 1

Z = 1

V = 1

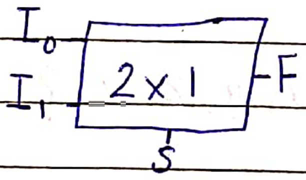
* Multiplexer :

combinational cct that select one of the inputs.

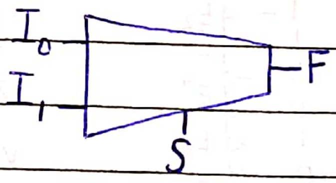
input output

(2 x 1) , (4 x 1) , (8 x 1) , (16 x 1) , ...

(2 x 1) :



or



if $S=0 \longrightarrow F = I_0$

if $S=1 \longrightarrow F = I_1$

(S: selecter)

S	F
0	I_0
1	I_1

$\longrightarrow (F = I_0\bar{S} + I_1 S)$

check:

$$F\Big|_{S=0} = I_0\bar{0} + \cancel{I_1 0}$$

$$= I_0 1 = I_0$$

$$F\Big|_{S=1} = \cancel{I_0\bar{1}} + I_1 1$$

$$= I_1$$

* $2^{(\# \text{ of } S_i)} = (\# \text{ of } I_i)$

4x1 Mux : $2^n = 4 \rightarrow n = 2 = (\# \text{ of } s)$

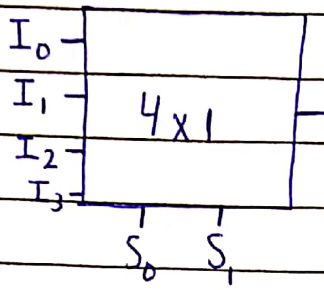
8x1 Mux : $2^n = 8 \rightarrow n = 3 = (\# \text{ of } s)$

	S	I_0	I_1	F	
I_0	0	0	0	0	0
	0	0	1	0	1
	0	1	0	1	2
	0	1	1	1	3
I_1	1	0	0	0	4
	1	0	1	1	5
	1	1	0	0	6
	1	1	1	1	7

S \ I_0I_1	00	01	11	10
0			1	1
1		1	1	

$(F = I_0\bar{S} + I_1 S)$

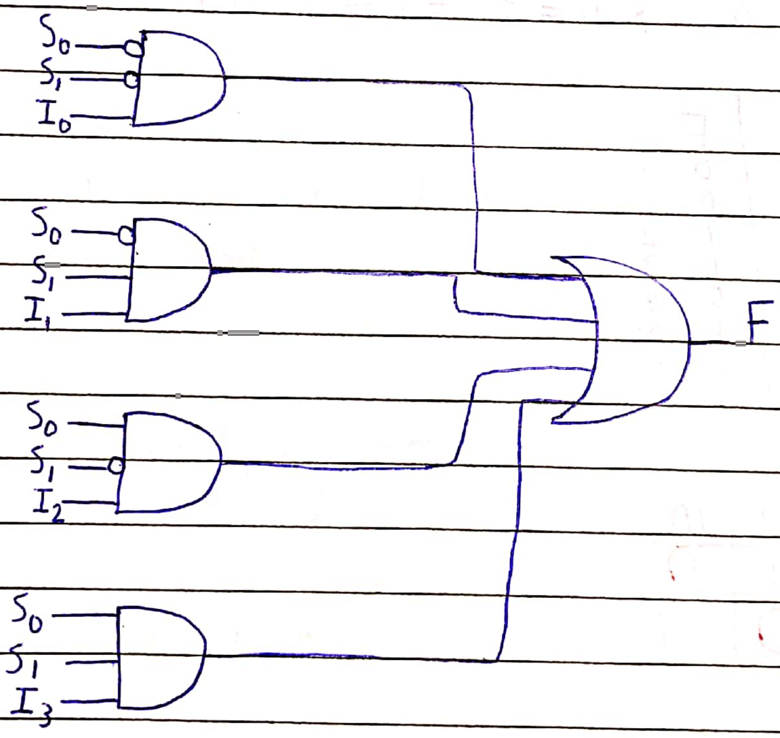
* Example : Design 4x1 Mux ?



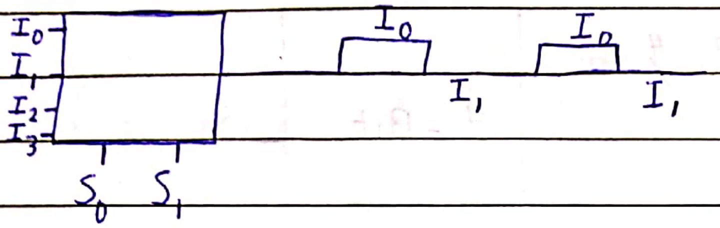
S_0	S_1	F	
0	0	I_0	0
0	1	I_1	1
1	0	I_2	2
1	1	I_3	3

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

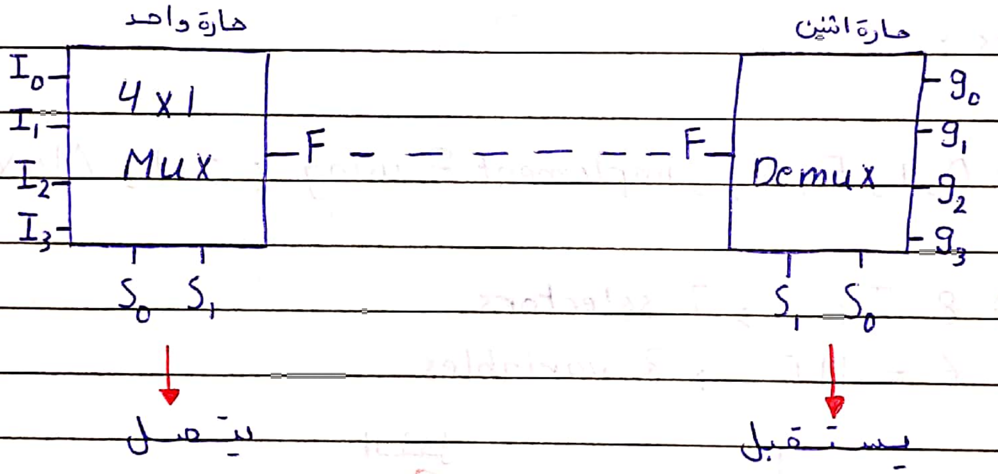
$$(F = I_0 \bar{S}_0 \bar{S}_1 + I_1 \bar{S}_0 S_1 + I_2 S_0 \bar{S}_1 + I_3 S_0 S_1)$$



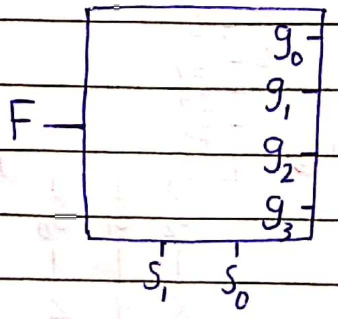
* تطبيق على الـ Mux هو التلفون الأرضي :



في حال الشخصين في نفس الوقت عملوا اتصال، يعني انشبك عندي خطين مثلاً I_0 و I_1 فعلياً الـ Selectors بصيروا يبدلوا بين هدول الشخصين بشكل سريع جداً بحيث لا نلاحظ الانقطاع في الصوت أثناء المكالمة.



* Demux :



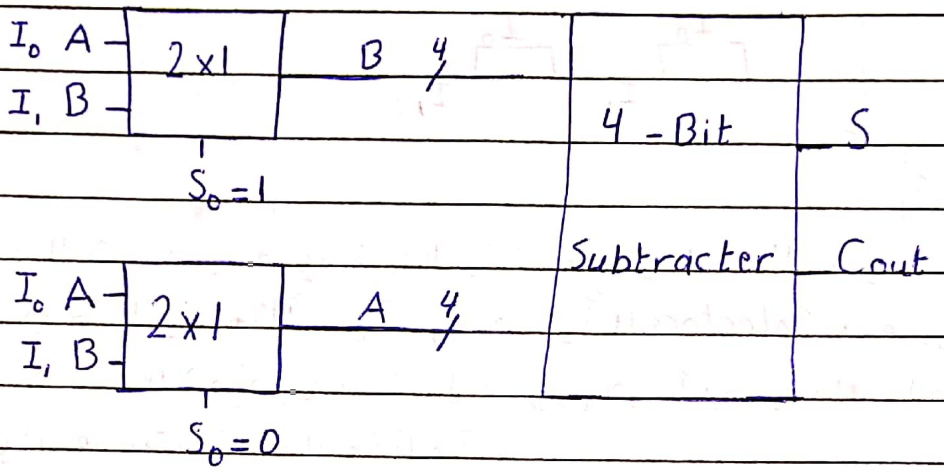
$g_0 = \bar{S}_1 \bar{S}_0 F$

$g_1 = \bar{S}_1 S_0 F$

$g_2 = S_1 \bar{S}_0 F$

$g_3 = S_1 S_0 F$

* Doing (B − A) where : B < A , by mux :



* Examples :

1) $F = \Sigma(0, 1, 5, 6)$, implement F using (8x1) Mux ?

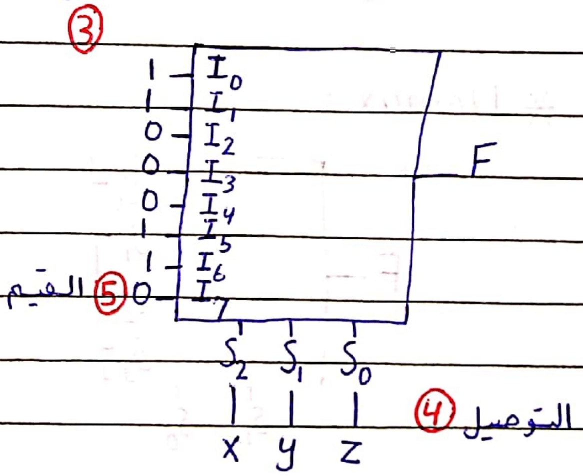
Solu : ① 8 I's → 3 selectors

6 = 110 → 3 variables

②

	X	y	z	F
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

الشكل ③

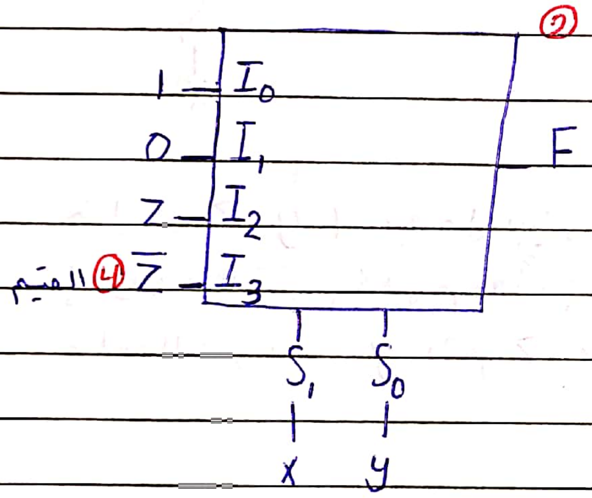


* when # of selectors = # of variables , connect Variables Directly with selectors .

Subject : .... / /

2) same previous F with (4x1) Mux ?

①	X	y	Z	F	③ العلاقة
I_0	0	0	0	1	$I_0 = 1$
	0	0	1	1	
I_1	0	1	0	0	$I_1 = 0$
	0	1	1	0	
I_2	1	0	0	0	$I_2 = Z$
	1	0	1	1	
I_3	1	1	0	1	$I_3 = \bar{Z}$
	1	1	1	0	



فبل (Z) بعمل بينه وبين (F) علاقة

* when # of selectors ≠ # of Variables :

- بقسم عدد السطور على عدد ال I's حتى أعرف كل I حتوخذ كم سطر: (2 for each I) ∴ → 8 ÷ 4 = 2

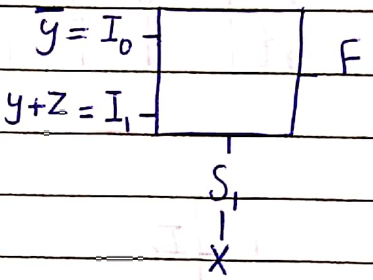
- ثم بشبك ال Variables مباشرة بال selectors على الترتيب بداية من X حتى ينتهي عدد ال selector ، ال Variable المتبقي أقوم بإيجاد علاقة بينه وبين ال (F) لكل I (سطرين).

Subject : .... / /

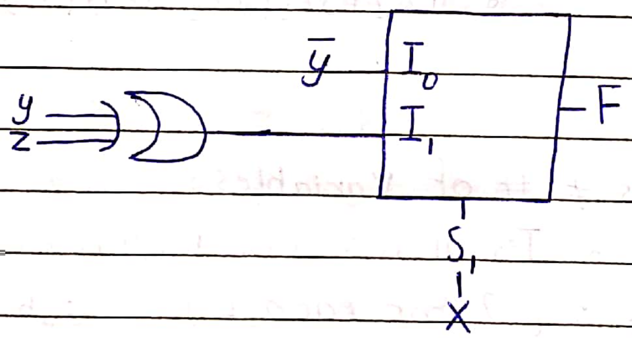
3) Same F with (2 x 1) Mux ?

8 raw / 2 I = 4 for each I

	x	y	z	F	
I_0	0	0	0	1	$I_0 = \overline{y}$
	0	0	1	1	
	0	1	0	0	
	0	1	1	0	
I_1	1	0	0	0	$I_1 = y + z$
	1	0	1	1	
	1	1	0	1	
	1	1	1	0	



خل (Z و y) بعمل بينهم وبين F علاقة
إدرس علاقتهم كلهم معاً بنفس الوقت ثم علاقة كل واحد لحال
على الترتيب y ثم Z ، وأول علاقة بتشوفها حطها.



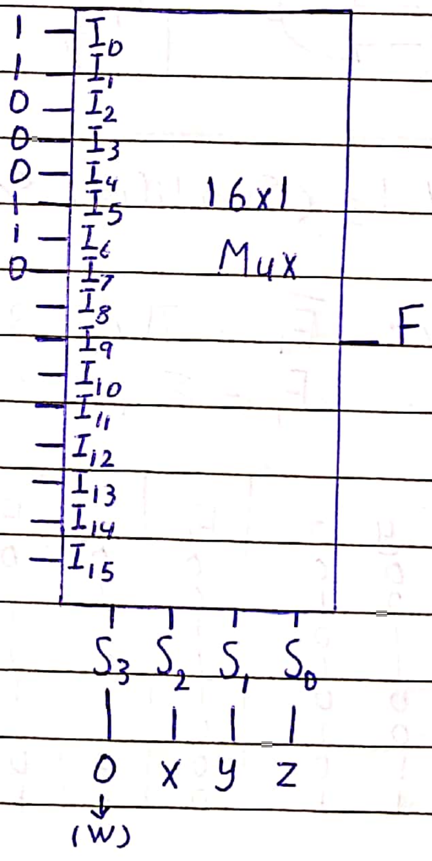
4) Find same F by (16 x 1) Mux ?

16 I , 8 raw

*لازم أعلى عدد الـ raws يساوي أو أكبر من عدد الـ I دائماً"
لذا هون بزيد متغير جديد على اليسار من المتغيرات الأصلية
(لأن زيادة صفر عالشمال ما بأثر على القيمة) وليكن كالتالي:
طبعاً دائماً قيمته = صفر

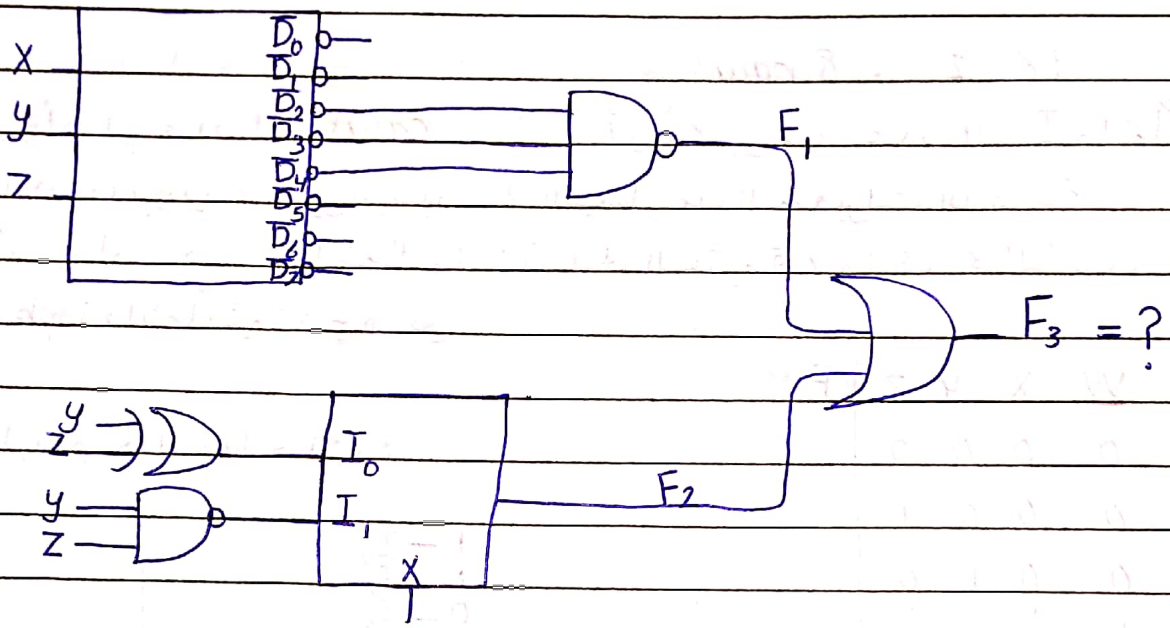
W	X	y	Z	F
0	0	0	0	
0	0	0	1	
0	0	1	0	
⋮	⋮	⋮	⋮	

هسا الرسم خلص انحلت:



بعبي الـ I اللي الهم سطور فقط
والباقي فاضي

* Example :



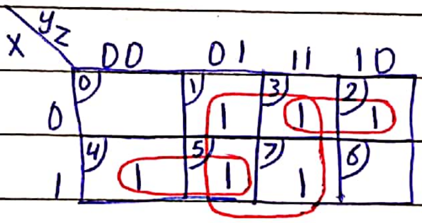
Find F_3 (Simplified Sop) ?

Solu : $\overline{F_1} = \Pi(2,3,4)$

$F_1 = \Sigma(2,3,4)$

X	F_2
0	$I_0 = y+z$
1	$I_1 = \overline{y.z}$

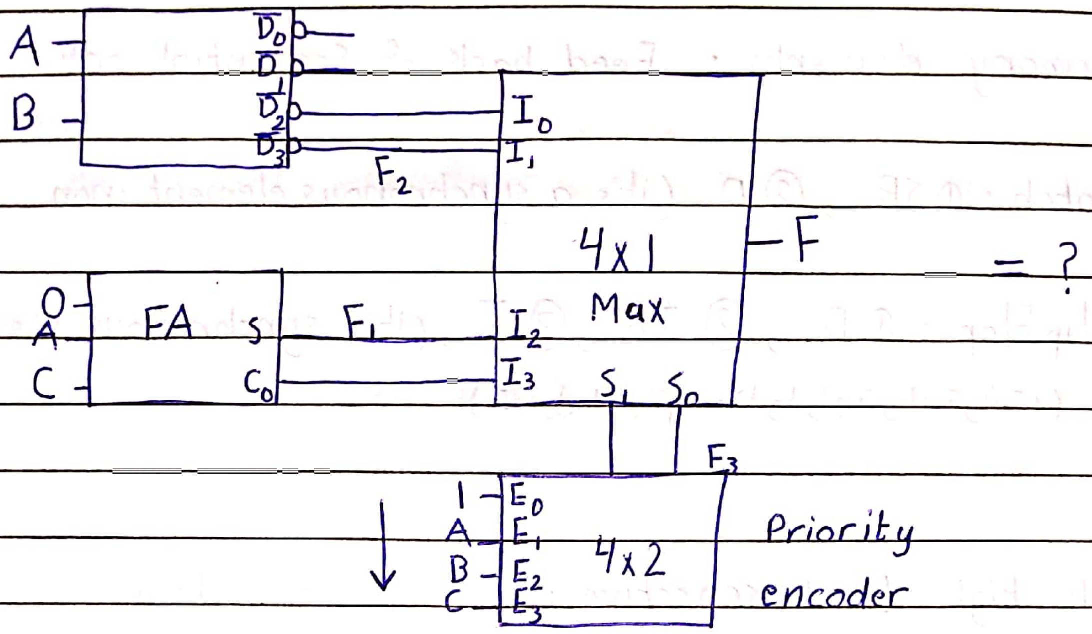
	X	y	z	F_1	F_2	F_3
0	0	0	0	0	0	0
1	0	0	1	0	1	1
2	0	1	0	1	1	1
3	0	1	1	1	0	1
4	1	0	0	1	1	1
5	1	0	1	0	1	1
6	1	1	0	0	0	0
7	1	1	1	0	1	1



$F_3 = \overline{X}y + Z + x\overline{y}$

$F_3 = F_1 + F_2$

* Example :



Solu :

	A	B	C	F_1: S	F_1: C_0	F_2	F_3: S_1	F_3: S_0	F
0	0	0	0	0	0	1	0	0	$I_0=0$
1	0	0	1	1	0	1	1	1	$I_3=0$
2	0	1	0	0	0	0	1	0	$I_2=0$
3	0	1	1	1	0	0	1	1	$I_3=0$
4	1	0	0	1	0	1	0	1	$I_1=0$
5	1	0	1	0	1	1	1	1	$I_3=1$
6	1	1	0	1	0	1	1	0	$I_2=1$
7	1	1	1	0	1	1	1	1	$I_3=1$

A \ BC	00	01	11	10
0	(0)	(1)	(3)	(2)
1	(4)	(5) 1	(7) 1	(6) 1

$(F = AC + AB)$

*شرح : أول شي قسم الرسمة لعدة أقسام

ثم حدد المدخلات والمخرجات $(F_1, F_2, ...)$ واعمل

جدول ثم بلش طلع واحد واحد :

F_1 : هو مخرجات الـ $FA = S, C_0$

F_2 : $F_2 = \Pi(2,3)$ معناها صفار

F_3 : امشي على الترتيب وخذ أكبر قيمة شغالة ، بلش من E_0 وانتهي بـ E_3 أكبر

وحدة شغالة بتلاقاها بطريقك خذ قيمتها وحطها على المخرجات اللي هم (S_1, S_0)

MSB

آخر اشي شوف الـ select وين بأشر وخذ القيمة .

The Hashemite University
Computer Engineering Department
Digital Logic (110408220)
HW4

Q4-1) Consider the combinational circuit shown in Fig.P4-1.

a) Derive the Boolean expression for T_1 Through T_4 . Evaluate the outputs F_1 and F_2 as a function of the four inputs.

b) List the truth table with 16 binary combinations of the four input variables, then list the binary values for T_1 through T_4 and outputs F_1 and F_2 in the table.

c) Plot the output Boolean functions obtained in part (b) on maps and show that the simplified Boolean expression are equivalent to the ones obtained in part (a).

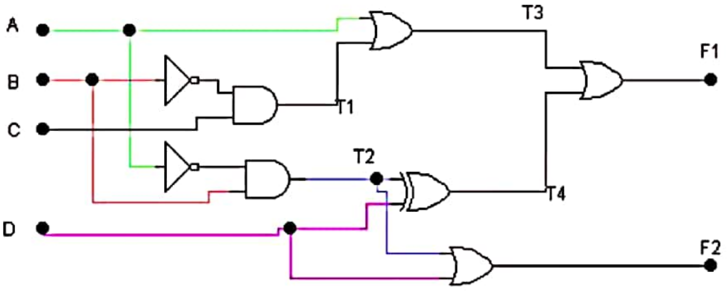


Fig.P4-1

Q4-2) Obtain the simplified Boolean expressions for output F and G in terms of the input variables in the circuit of Fig.P4-2.

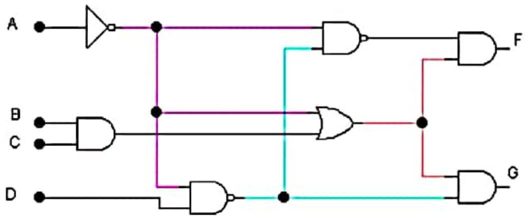


Fig.P4-2

Q4-3) For the circuit shown in Fig.4-26(section 4-10),

a) write the Boolean functions for the four outputs in terms of the input variables.

b) If the circuit is listed in a truth table , how many rows and columns would there be in the table?

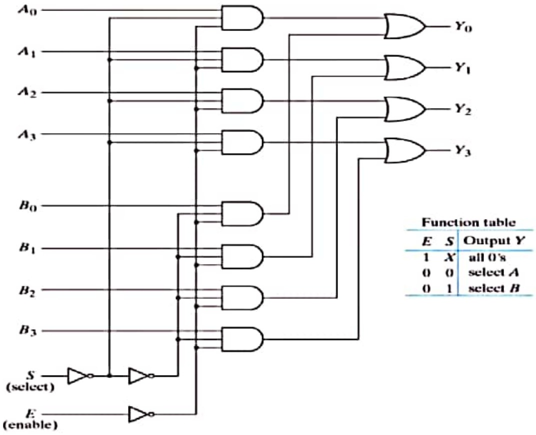


Fig. 4-26 Quadruple 2-to-1-Line Multiplexer

Q4-4) Design a combinational circuit with three inputs and one output. The output is 1 when the binary value of the input is less than 3 . The output is 0 otherwise.

Q4-5) Design a combinational circuit with three inputs , *x*, *y*, and *z*, and the three outputs, *A*, *B*, and *C*. when the binary input is 0,1,2, or 3, the binary output is one greater than the input. When the binary input is 4,5,6, or 7, the binary output is one less than the input.

Q4-6) A majority circuit is a combinational circuit whose output is equal to 1 if the input variables have more 1's than 0's . The output is 0 otherwise . Design a 3-input majority circuit.

Q4-12)

a) Design a half – subtractor circuit with inputs *x* and *y* and the outputs *D* and *B* . The circuit subtracts the bits $x - y$ and places the difference in *D* and the borrow in *B*.

b)Design a Full–subtractor circuit with the three inputs *x*, *y* , *z* and two outputs *D* and *B* . The circuit subtracts $x - y - z$, where *z* is the input borrow , *B* is the output borrow , and *D* is the difference.

Q4-13) The adder–subtractor circuit of Fig.4-13 has the following values for mode input *M* and data inputs *A* and *B* . In each case , determine the values of the four SUM outputs , The carry *C* , and overflow *V*.

	M	*A*	*B*
(a)	0	0111	0110
(b)	0	1000	1001
(c)	1	1100	1000
(d)	1	0101	1010
(e)	1	0000	0001

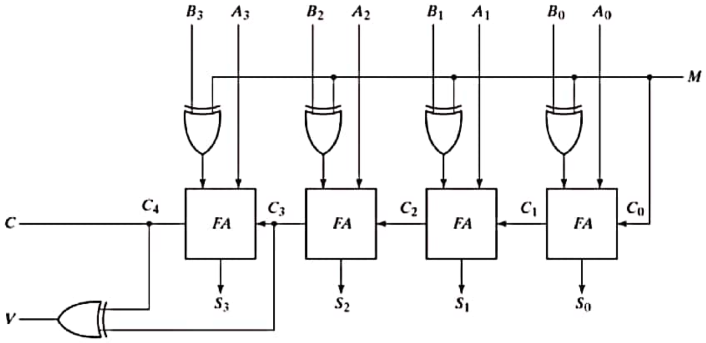


Fig. 4-13 4-Bit Adder Subtractor

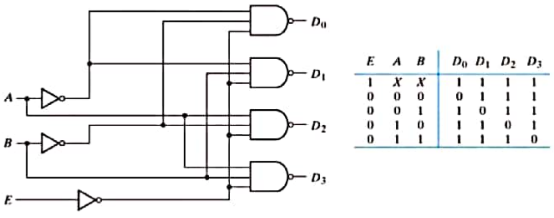
Q4-25) Construct a 5–to-32 line decoder with four 3-to-8 line decoders with enable and a 2-to-4-line decoder. Use block diagrams for the components .

Q4-26) Construct a 4-to-16-line decoder with five 2-to-4-line decoders with enable.

Q4-27) A combinational circuit is specified by the following three Boolean functions:

$F_1(A,B,C)=\sum(2,4,7)$, $F_2(A,B,C)=\sum(0,3)$, $F_3(A,B,C)=\sum(0,2,3,4,7)$.

Implement the circuit with a decoder constructed with NAND gates (similar to Fig.4-19) and NAND or AND gates connected to the decoder outputs . Use a block diagram for the decoder . Minimize the number of inputs in the external gates.



E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

(a) Logic diagram (b) Truth table

Fig. 4-19 2-to-4-Line Decoder with Enable Input

Q4-28) A combinational circuit is defined by the following three Boolean functions:

$F_1 = x'y'z' + xz$, $F_2 = xy'z' + x'y$, $F_3 = x'y'z + xy$. Design the circuit with a decoder and external gates.

Q4-29) Design a 4-input priority encoder with inputs as in table 4-8 (See textbook), but with input D_0 having the highest priority and input D_3 the lowest priority.

D_0 : highest priority. D_3 : lowest priority

Inputs				**Outputs**		
D_0	D_1	D_2	D_3	x	Y	V

Q4-31) Construct a 16×1 multiplexer with two 8×1 and one 2×1 multiplexers. Use block diagrams.

Q4-32) Implement the following Boolean function with a multiplexer:
$F(A,B,C,D)=\sum(0,1,3,4,8,9,15)$.

Q4-33) Implement a full adder with two 4×1 multiplexers.

Q4-34) An 8×1 multiplexer has inputs A, B, and C connected to the selection inputs S_2 , S_1 , and S_0 , respectively . The data inputs I_0 through I_7 , are as follows: $I_1=I_2=I_7=0$; $I_3=I_5=1$; $I_0=I_4=D$; and $I_6=D'$, Determine the Boolean function that the multiplexer implements.

Q4-35) Implement the following Boolean function with a 4×1 multiplexer and external gates. Connect inputs A and B to the selection lines . Te input requirements for the four data lines will be a function of variables C and D. These values are obtained by expressing F as a function of C and D for each of the four cases when AB =00, 01, 10, and 11 . These functions may have to be implemented with external gates $F(A,B,C,D)=\sum(1,3,4,11,12,13,14,15)$.

Subject : .... / /

* HW/4 :

Q 4-1)

a) $T_1 = \overline{B}C$

$T_2 = \overline{A}B$

$T_3 = A + T_1 = A + \overline{B}C$

$T_4 = \underline{T_2 + D} = \overline{T_2}D + T_2\overline{D}$

$= \overline{(\overline{A}B)}D + \overline{A}B\overline{D} = (A+\overline{B})D + \overline{A}B\overline{D} = AD + \overline{B}D + \overline{A}B\overline{D}$

$F_1 = T_3 + T_4 = \underline{A} + \overline{B}C + \underline{AD} + \overline{B}D + \overline{A}B\overline{D}$

$= \underline{A} + \overline{B}C + \overline{B}D + \underline{\overline{A}B\overline{D}}$

$= A + \overline{B}C + \overline{B}D + B\overline{D}$

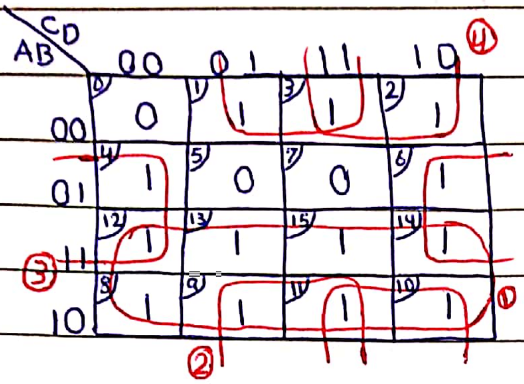
$F_2 = T_2 + D = \overline{A}B + D$

b)

	A	B	C	D	T_1	T_2	T_3	T_4	F_1	F_2
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	1	1
2	0	0	1	0	1	0	1	0	1	0
3	0	0	1	1	1	0	1	1	1	1
4	0	1	0	0	0	1	0	1	1	1
5	0	1	0	1	0	1	0	0	0	1
6	0	1	1	0	0	1	0	1	1	1
7	0	1	1	1	0	1	0	0	0	1
8	1	0	0	0	0	0	1	0	1	0
9	1	0	0	1	0	0	1	1	1	1
10	1	0	1	0	1	0	1	0	1	0
11	1	0	1	1	1	0	1	1	1	1
12	1	1	0	0	0	0	1	0	1	0
13	1	1	0	1	0	0	1	1	1	1
14	1	1	1	0	0	0	1	0	1	0
15	1	1	1	1	0	0	1	1	1	1

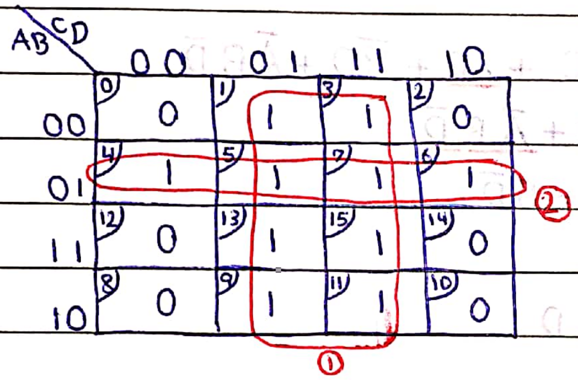
$F_1 = \Sigma(1,2,3,4,6,8,9,10,11,12,13,14,15)$

$F_2 = \Sigma(1,3,4,5,6,7,9,11,13,15)$



AB \ CD	00	01	11	10
00	0	1	1	1
01	1	0	0	1
11	1	1	1	1
10	1	1	1	1

$F_1 = A + \bar{B}D + B\bar{D} + \bar{B}C$



AB \ CD	00	01	11	10
00	0	1	1	0
01	1	1	1	1
11	0	1	1	0
10	0	1	1	0

$F_2 = D + \bar{A}B$ (same)

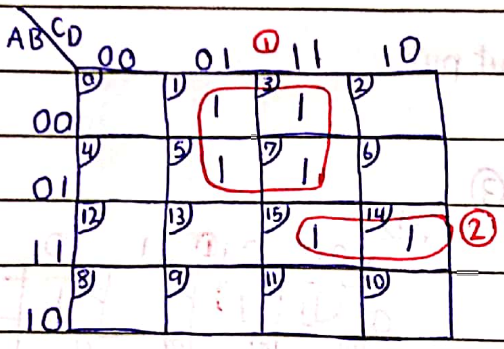
*Q 4-2) $Z = \overline{(\bar{A}D)} = (A + \bar{D})$, $w = BC$

$X = \overline{(\bar{A}Z)} = A + \bar{Z} = \overline{A + \bar{D}} = A + \bar{A}D = A + D$

$y = \bar{A} + w = \bar{A} + BC$

$F = X \cdot y = (A+D)\cdot(\bar{A} + BC) = ABC + \bar{A}D + BCD$ (Sop)

111 01 111



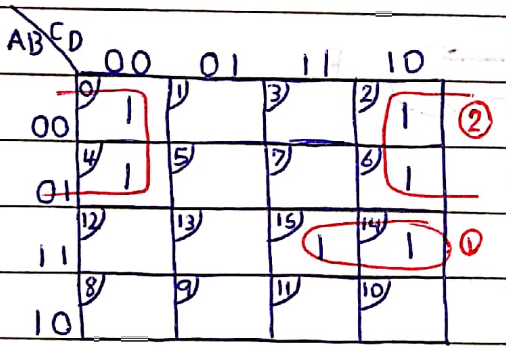
AB \ CD	00	01	11	10
00	^{0}	^{1} 1	^{3} 1	^{2}
01	^{4}	^{5} 1	^{7} 1	^{6}
11	^{12}	^{13}	^{15} 1	^{14} 1
10	^{8}	^{9}	^{11}	^{10}

$F = \bar{A}D + ABC$

$G = y.Z = (\bar{A} + BC)(A + \bar{D})$

$= \bar{A}\bar{D} + ABC + BC\bar{D}$ (SOP)

00 111 110



AB \ CD	00	01	11	10
00	^{0} 1	^{1}	^{3}	^{2} 1
01	^{4} 1	^{5}	^{7}	^{6} 1
11	^{12}	^{13}	^{15} 1	^{14} 1
10	^{8}	^{9}	^{11}	^{10}

$G = \bar{A}\bar{D} + ABC$

* Q 4-3)

a) $Y_i = X_i + W_i$

$X_i = A_i\bar{S}\bar{E}$, $W_i = B_i S\bar{E}$

$Y_i = A_i\bar{S}\bar{E} + B_i S\bar{E}$, $i = 0,1,2,3$

b) [# of rows = $2^{(\# \text{ of inputs})}$]

\# of inputs = $4+4+2 = 10$

$\therefore$ rows = $2^{10} = 1024$ rows

[# of columns = # of inputs and outputs]

$\therefore$ columns = $10 + 4 = 14$ columns

* Q 4-4)

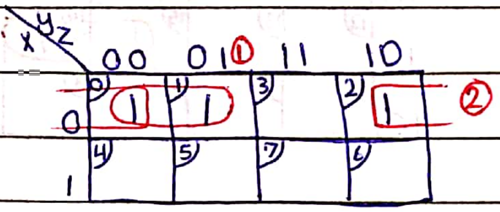
steps:

① (3) input ⟶ (1) output

②

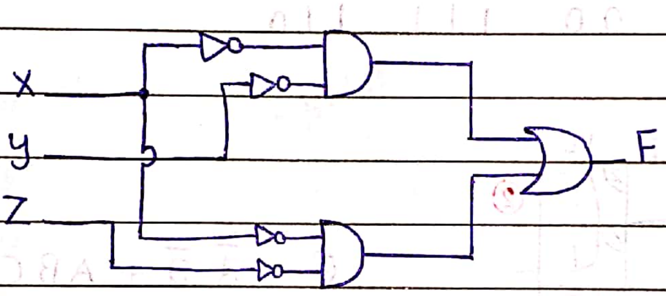
	x	y	z	F
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

③



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

④



* Q 4-5)

① 3 input ⟶ 3 output

②

x	y	z	A	B	C
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	1	0

output = (input + 1)

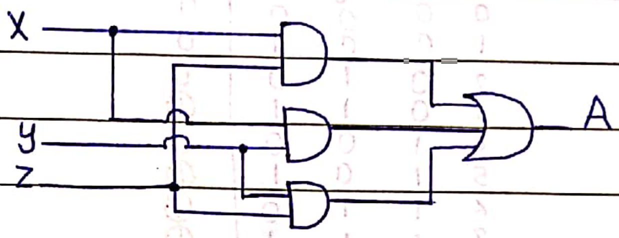
output = (input − 1)

③

x \ yz	00	01	③ 11	10
0	(0)	(1)	(3) 1	(2)
1	(4)	(5) 1	(7) 1	(6) 1

② ①

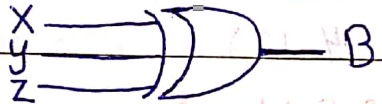
$A = xz + xy + yz$



x \ yz	00	01	11	10
0	(0)	(1) 1	(3)	(2) 1
1	(4) 1	(5)	(7) 1	(6)

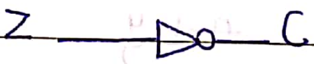
$(B = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz)$

$\therefore B = X + y + z$



x \ yz	00	01	11	10
0	(0) 1	(1)	(3)	(2) 1
1	(4) 1	(5)	(7)	(6) 1

$C = \bar{z}$



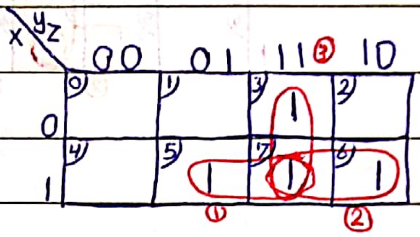
* Q 4-6)

① 3 input ⟶ 1 output

②

	X	y	Z	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	①
4	1	0	0	0
5	1	0	1	①
6	1	1	0	①
7	1	1	1	①

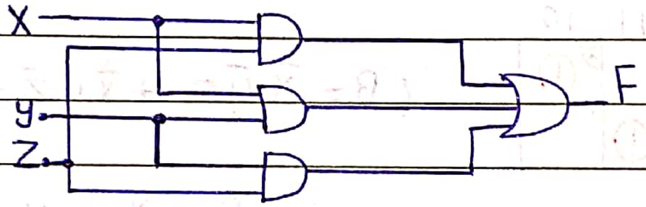
③



x \ yz	00	01	11 ③	10
0	⁰	¹	³ 1	²
1	⁴	⁵ 1	⁷ 1	⁶ 1

$F = XZ + Xy + yZ$

④



* Q 4-12) (we use subtracter with 2's complement)

مهم Half Subtracter

a) ① 2 input ⟶ 2 output

(# of Bits = # of output)

0) $\overset{①}{0}$ 0 x
~~0 0~~ y
1 1
1 +
0 0
B D

1) $\overset{⓪}{0}$ 0 x
~~0 1~~ y
1 0
1 +
⓪ 1 1
B D

2) $\overset{①}{0}$ 1 x
~~0 0~~ y
1 1
1 +
① 0 1

3) $\overset{①}{0}$ 1
~~0 1~~
1 0
1 +
① 0 0

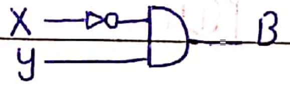
②

	x	y	B	D
0	0	0	0	0
1	0	1	①	①
2	1	0	0	①
3	1	1	0	0

③ and ④

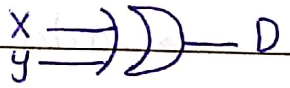
x \ y	0	1
0	^{0}	^{1} ①
1	^{2}	^{3}

$B = \bar{x}\,y$



x \ y	0	1
0	^{0}	^{1} ①
1	^{2} ①	^{3}

$$D = \bar{x}y + x\bar{y}$$
$$= x \oplus y$$



b) ① 3 input → 2 output

مهم Full Subtracter

0)
$$\begin{array}{r} \overset{①}{0}\;0 \;\; x \\ \cancel{0}\;\cancel{0} \;\; y \\ \hline 1\;\;1 \;\;\; \\ 1 + \\ \hline 1\,\underbrace{0\;0} \end{array}$$

$$\begin{array}{r} \overset{①}{0}\;0 \;\; x-y \\ \cancel{0}\;\cancel{0} \;\; z \\ \hline 1\;\;1 \;\;\; \\ 1 + \\ \hline 1\,\underbrace{0\;0}_{B\;\;D} \end{array}$$

7)
$$\begin{array}{r} \overset{①}{0}\;1 \;\; x \\ \cancel{0}\;\cancel{1} \;\; y \\ \hline 1\;\;0 \;\;\; \\ 1 + \\ \hline 1\,\underbrace{0\;0} \end{array}$$

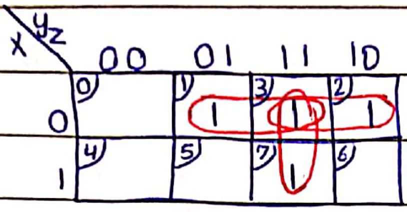
$$\begin{array}{r} \overset{⓪}{0}\;0 \;\; x-y \\ \cancel{0}\;\cancel{1} \;\; z \\ \hline 1\;\;0 \;\;\; \\ 1 + \\ \hline \underbrace{1\;\;1}_{B\;\;D} \end{array}$$

and so on...

②

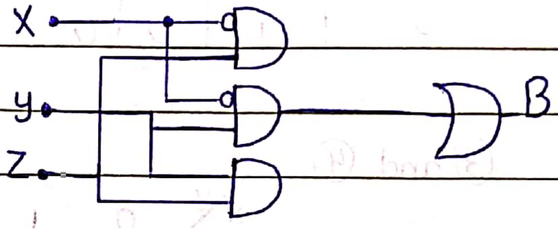
	x	y	z	B	D
0	0	0	0	0	0
	0	0	1	①	①
	0	1	0	①	①
	0	1	1	①	0
	1	0	0	0	①
	1	0	1	0	0
	1	1	0	0	0
7	1	1	1	①	①

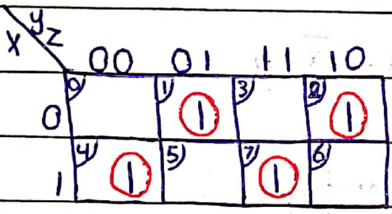
③ and ④



x \ yz	00	01	11	10
0		1	1	1
1			1	

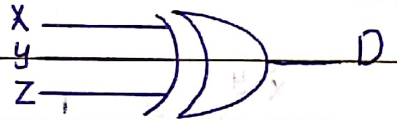
$B = \bar{X}Z + \bar{X}y + yZ$





x \ yz	00	01	11	10
0		1		1
1	1		1	

$D = X + y + Z$



* Q 4-13)

a) M = 0 ⟶ add :

Cin
① 1
0111
0110
Cout ⓪ 1101

S = 1101
$C = C_o = 0$
V = 1

b) M = 0 ⟶ add :

⓪
1000
1001
① 0001

S = 0001
$C_o = 1$
V = 1

c) $M=1 \longrightarrow$ sub :

~~1100~~
~~1000~~
~~1000~~
~~①0100~~

on fly method
not valied

$S = 0100$
$C_o = 1$
$V = 0$

① 1 1
1100
~~1000~~
0111
1 +
①0100

d) $M=1 \longrightarrow$ sub :

①
0101
~~1010~~
0101
1 +
⓪1011

$S = 1011$
$C_o = 0$
$V = 1$

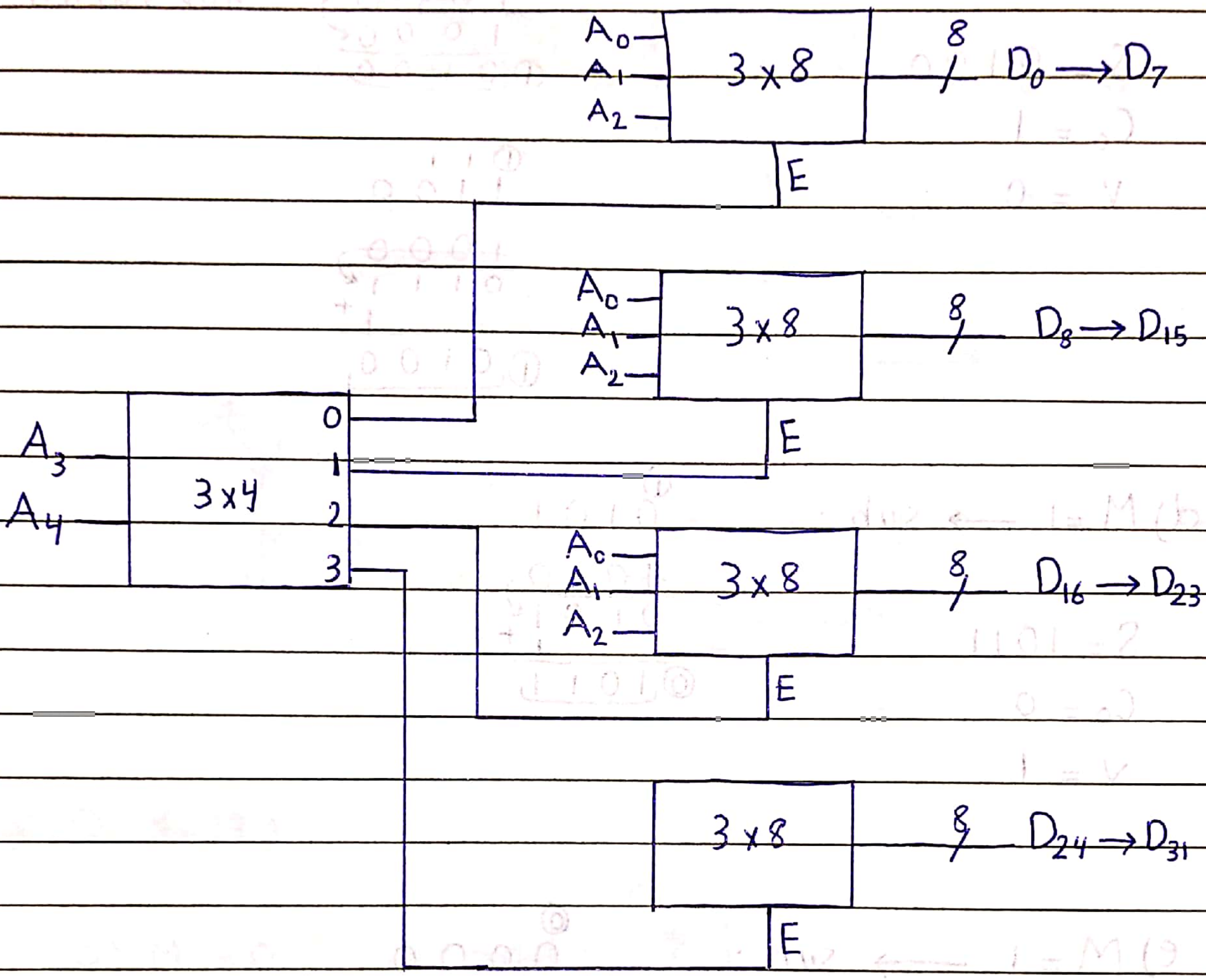
e) $M=1 \longrightarrow$ sub :

⓪
0000
0001
1110
1 +
⓪1111

$S = 1111$
$C_o = 0$
$V = 0$

▸ Subject : .... / /

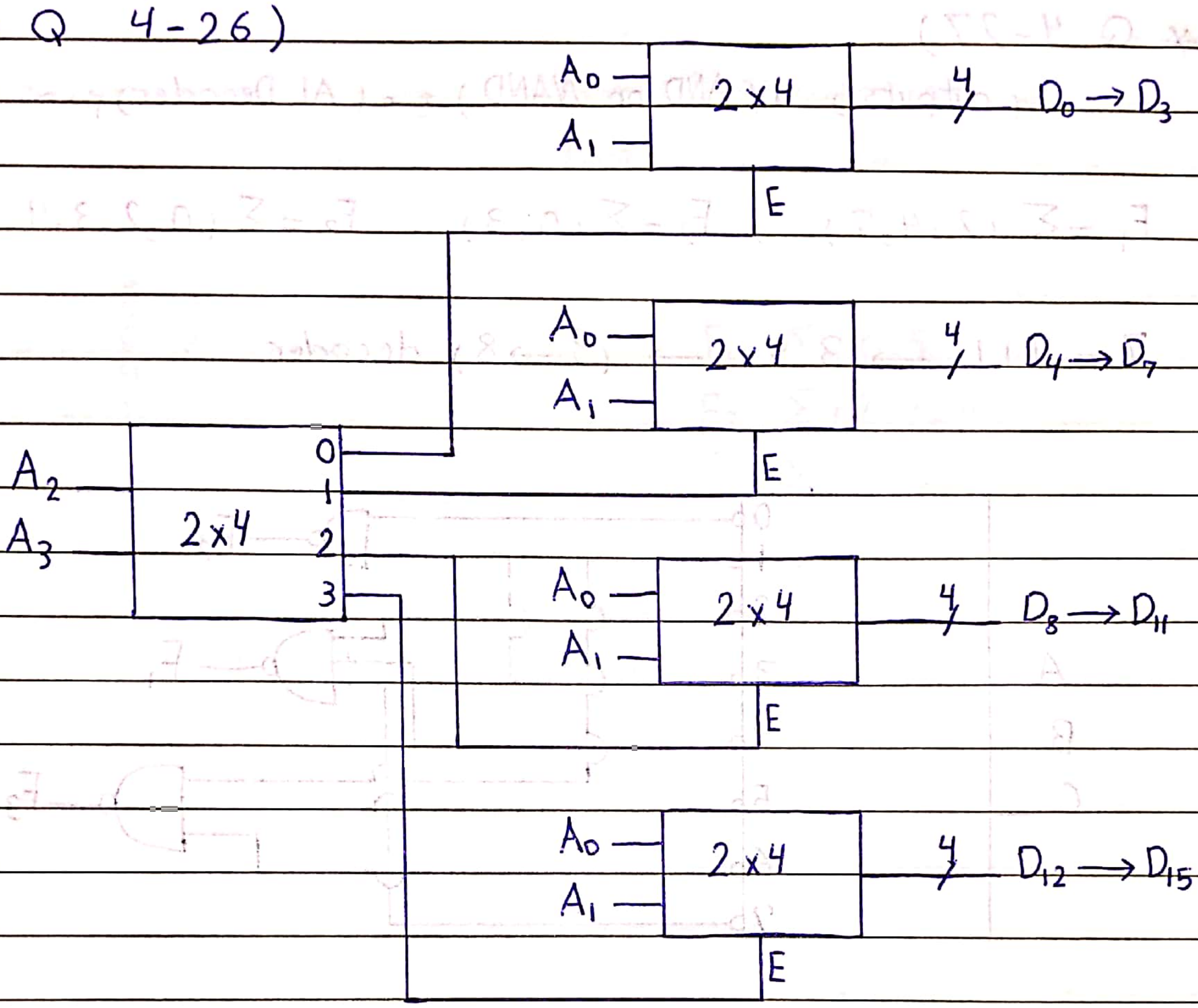
* Q 4-25)



(5 → 32) line decoder

▸ Subject : .... / /

* Q 4-26)

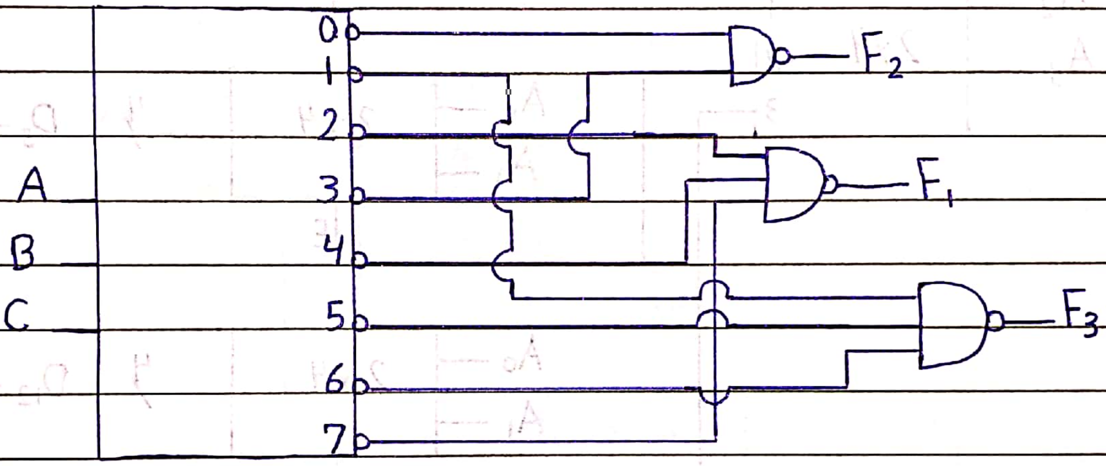


* Q 4-27)

استخدم (Al Decoder) مع (AND or NAND) لـ (outputs) .

$F_1 = \Sigma(2,4,7)$, $F_2 = \Sigma(0,3)$, $F_3 = \Sigma(0,2,3,4,7)$

7 = 111 → 3 V → (3 → 8) decoder



$\overline{F_1} = \Pi(2,4,7)$

$F_1 = \overline{(\overline{F_1})}$

with NAND

$\overline{F_2} = \Pi(0,3)$

F_2 with NAND

$\overline{F_3} = \Sigma(1,5,6)$

F_3 with NAND

* Q 4-28)

$F_1 = \overline{x}\,\overline{y}\,\overline{z} + xz$

$F_2 = x\overline{y}\,\overline{z} + \overline{x}y$

$F_3 = \overline{x}\,\overline{y}z + xy$

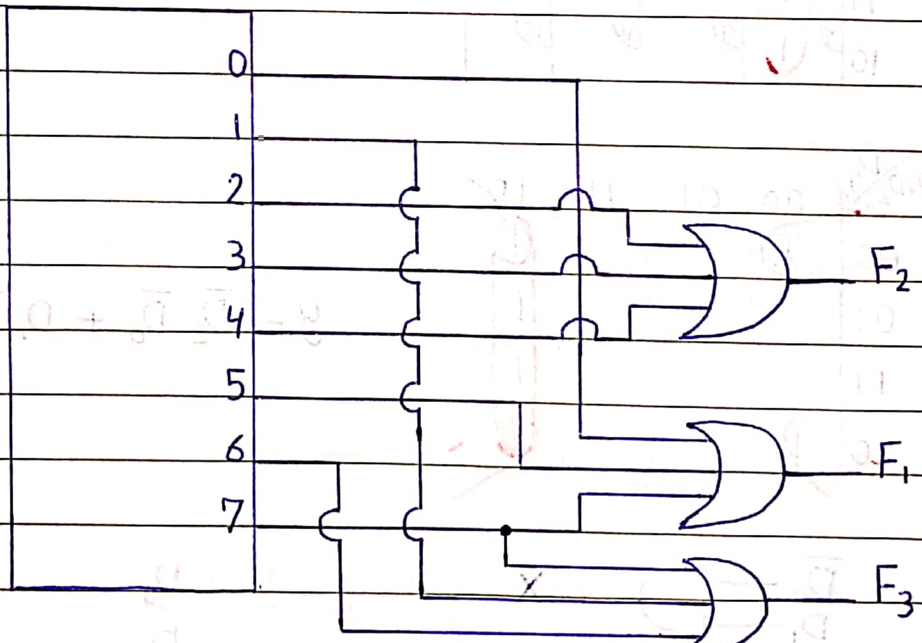
x	y	z	F_1	F_2	F_3
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	1

$F_1 = \Sigma(0,5,7)$

$F_2 = \Sigma(2,3,4)$

$F_3 = \Sigma(1,6,7)$

7 = 111 → (3 → 8) decoder



* Q 4-29)

E_0 ↖ Low D_3	E_1 ↑ D_2	E_2 ↑ D_1	E_3 ↖ high D_0	x	y	V
0	0	0	0	X	X	0
X	X	X	1	0	0	1
X	X	1	0	0	1	1
X	1	0	0	1	0	1
1	0	0	0	1	1	1

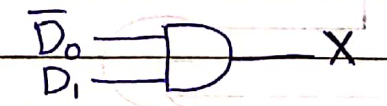
$V = D_3 + D_2 + D_1 + D_0$

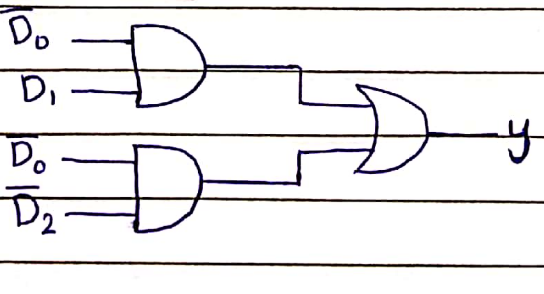
$D_3D_2 \backslash D_1D_0$	00	01	11	10
00	(0) X	(1)	(3)	(2)
01	(4) 1	(5)	(7)	(6)
11	(12) 1	(13)	(15)	(14)
10	(8) 1	(9)	(11)	(10)

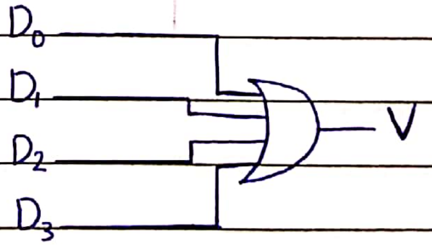
$X = \overline{D}_1 \overline{D}_0$

$D_3D_2 \backslash D_1D_0$	00	01	11	10
00	X			1
01				1
11				1
10	1			1

$y = \overline{D}_2 \overline{D}_0 + D_1 \overline{D}_0$

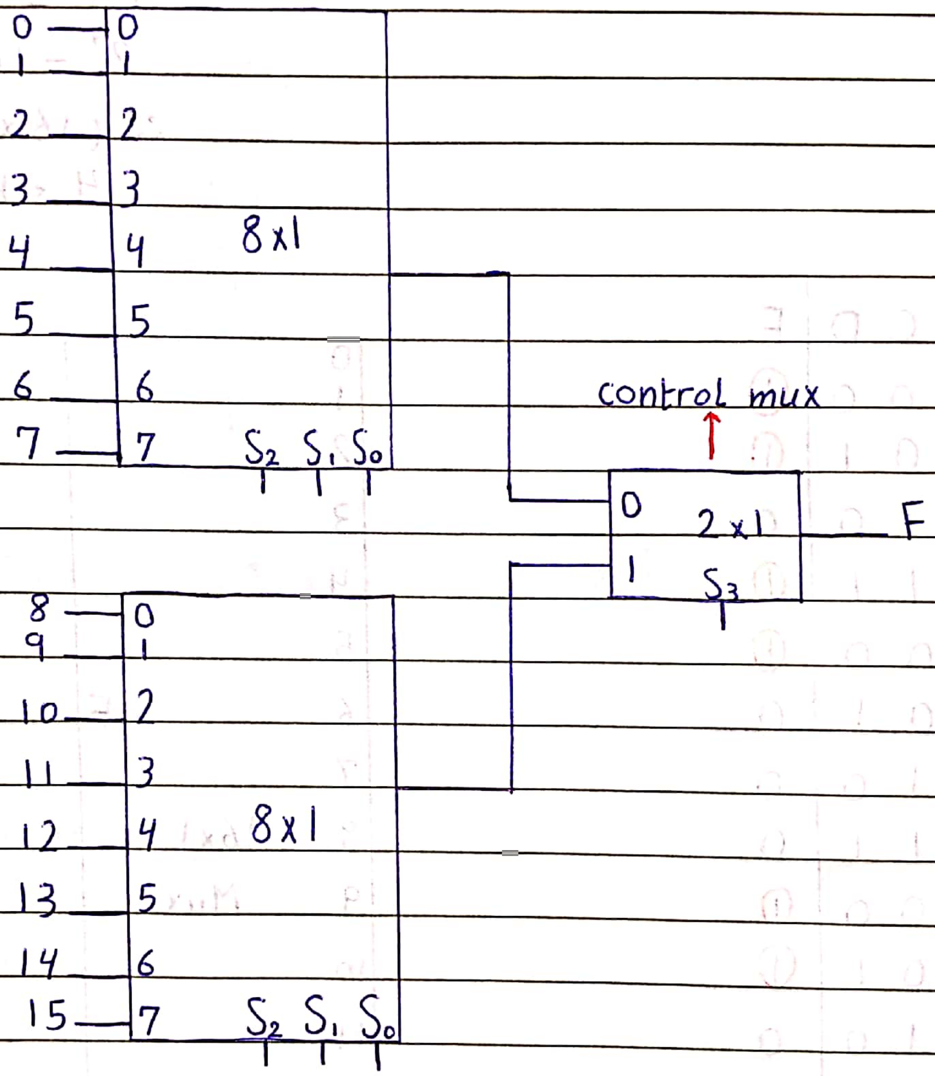






* Q 4-31)

(16 x1) by 2 (8x1) and 1 (2x1):



ال mux عبارة عن مدخلات Decimel و selector
ومخرجات (output Decimel) 1

حط ببالك: بدي غير عدد كل المدخلات كمان selectors (control) بختارلي
ال mux اللي فيو القيمة المرادة.

* Q 4-32)

$F = \Sigma(0, 1, 3, 4, 8, 9, 15)$

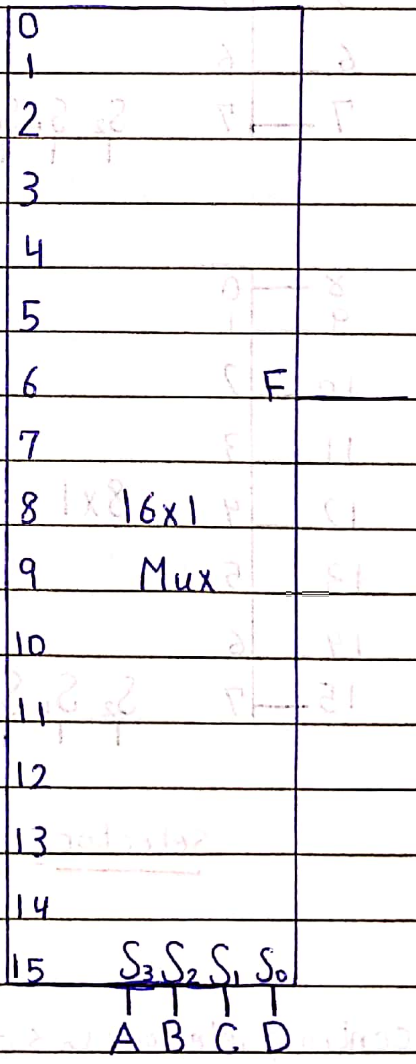
$1111 = 4$ V

$2^4 = 16$

∴ (16x1) mux

4 selectors

A B C D	F
0 0 0 0	1
0 0 0 1	1
0 0 1 0	0
0 0 1 1	1
0 1 0 0	1
0 1 0 1	0
0 1 1 0	0
0 1 1 1	0
1 0 0 0	1
1 0 0 1	1
1 0 1 0	0
1 0 1 1	0
1 1 0 0	0
1 1 0 1	0
1 1 1 0	0
1 1 1 1	1



* Q 4-33)

Full Added by 2 (4x1) mux:

$S = \Sigma(1,2,4,7) \longrightarrow 8\times1$

$C_o = \Sigma(3,5,6,7) \longrightarrow 8\times1$

	x	y	z	S	C_o
I_0	0	0	0	0	0
	0	0	1	1	0
I_1	0	1	0	1	0
	0	1	1	0	1
I_2	1	0	0	1	0
	1	0	1	0	1
I_3	1	1	0	0	1
	1	1	1	1	1

Z	$I_{0(S)}$
0	0
1	1

$\longrightarrow I_{0(S)} = Z$

Z	$I_{2(S)}$
0	1
1	0

$\longrightarrow I_{2(S)} = \overline{Z}$

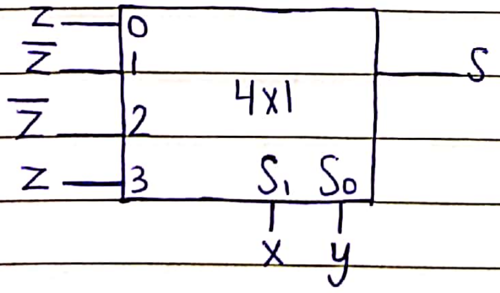
Z	$I_{1(S)}$
0	1
1	0

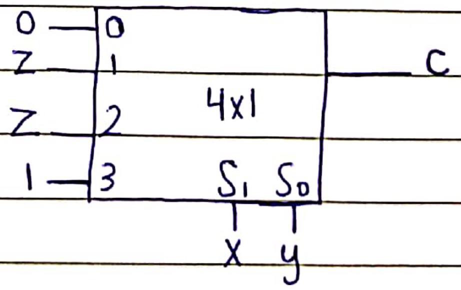
$\longrightarrow I_{1(S)} = \overline{Z}$

Z	$I_{3(S)}$
0	0
1	1

$\longrightarrow I_{3(S)} = Z$

$I_{0(C)} = 0 \qquad I_{1(C)} = Z \qquad I_{2(C)} = Z \qquad I_{3(C)} = 1$





* Q 4-34)

$I_1 = I_2 = I_7 = 0$

$I_3 = I_5 = 1$

$I_0 = I_4 = D$

$I_6 = \overline{D}$

	A	B	C	D	F
I_0	0	0	0	0	0
	0	0	0	1	1
I_1	0	0	1	0	0
	0	0	1	1	0
I_2	0	1	0	0	0
	0	1	0	1	0
I_3	0	1	1	0	1
	0	1	1	1	1
I_4	1	0	0	0	0
	1	0	0	1	1
I_5	1	0	1	0	1
	1	0	1	1	1
I_6	1	1	0	0	1
	1	1	0	1	0
I_7	1	1	1	0	0
	1	1	1	1	0

$\therefore F = \sum(1, 6, 7, 9, 10, 11, 12)$

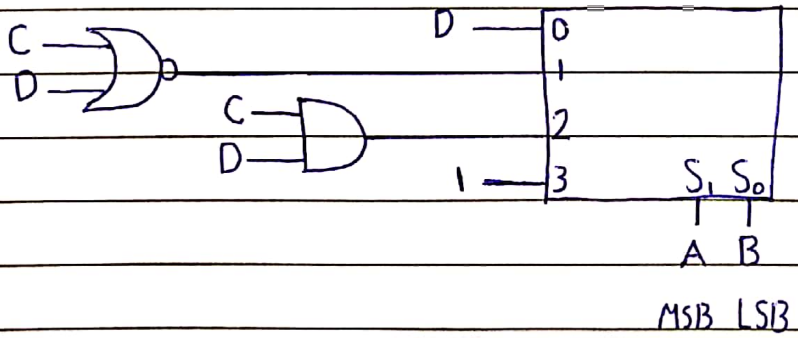
* Q 4-35)

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

by (4x1) Mux

	S_1	S_0				
	A	B	C	D	F	
I_0	0	0	0	0	0	$I_0 = D$
	0	0	0	1	1	
	0	0	1	0	0	
	0	0	1	1	1	
I_1	0	1	0	0	1	$I_1 = \overline{(C+D)}$
	0	1	0	1	0	
	0	1	1	0	0	
	0	1	1	1	0	
I_2	1	0	0	0	0	$I_2 = CD$
	1	0	0	1	0	
	1	0	1	0	0	
	1	0	1	1	1	
I_3	1	1	0	0	1	$I_3 = 1$
	1	1	0	1	1	
	1	1	1	0	1	
	1	1	1	1	1	

C D	I_1
0 0	1
0 1	0
1 0	0
1 1	0



Chapter 5 (Synchronous Sequential logic)

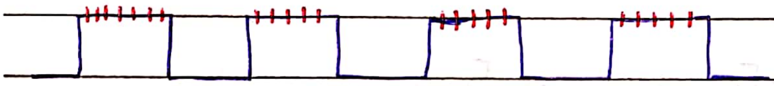
* memory elements : Feed back of Sequential cct

1) Latch : ① SR , ② D , (it's a synchronous element (non

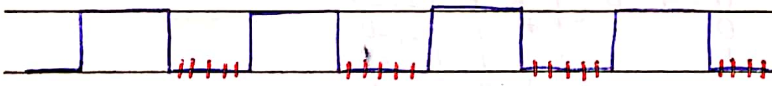
2) Flip Flop : ① D , ② JK , ③ T , (it's synchronous element).

(تخيل انهم بمثلوا راوتر انترنت)

latch high level sensetive :

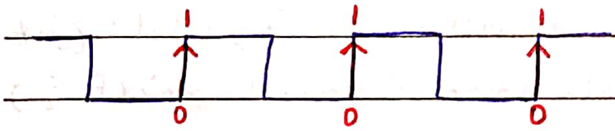


Latch low level sensetive :



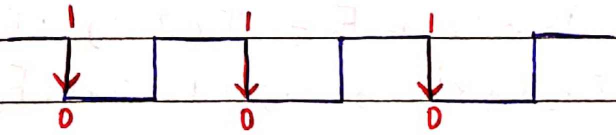
التغير
على
فترة

Flip Flop +ve edg trigged (0→1) :

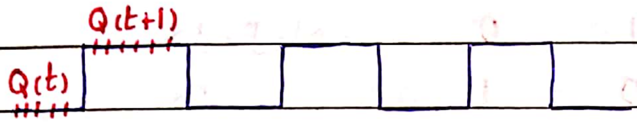


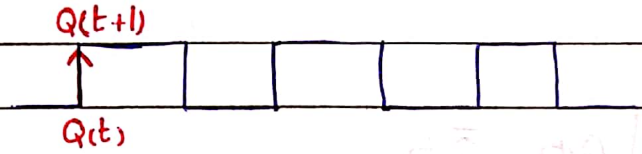
التغير
لحظي

Flip Flop -ve edg trigged (1→0):



Q (t) : Current state (0 or 1).
Q (t+1) : Future / next state.

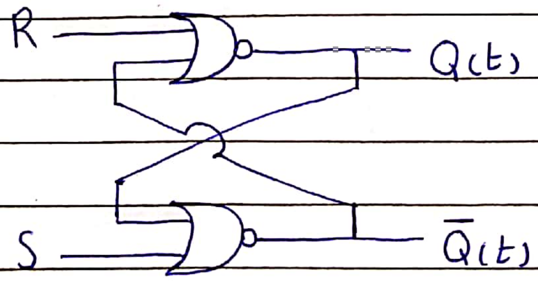




* Active high SR latch :
S: ON / set / 1 / high
R: off / reset / 0 / low

	S R	Q(t) (current state)	$\bar{Q}$(t)
Set	1 0	1	0
No change	0 0	1	0
Reset	0 1	0	1
Forbiddin	1 1	0	0

Set: شغّل وتُمثل ب $(\overset{Q}{1}\ \overset{\bar{Q}}{0})$.
no change: أبقي القيمة السابقة كما هي $(\overset{Q}{0}\ \overset{\bar{Q}}{0})$
Reset: طفّي وتمثل ب $(\overset{Q}{0}\ \overset{\bar{Q}}{1})$.
Forbiddin: مفرّكل القيم وتمثل $(\overset{Q}{1}\ \overset{\bar{Q}}{1})$

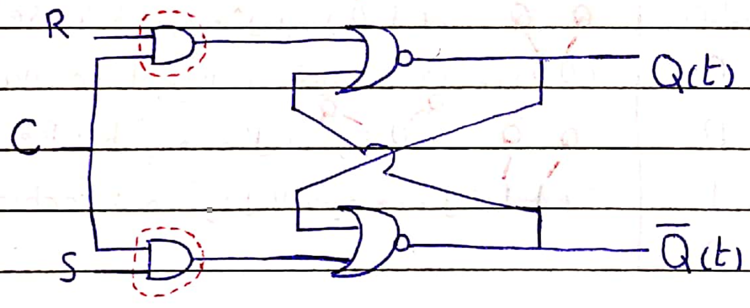


*

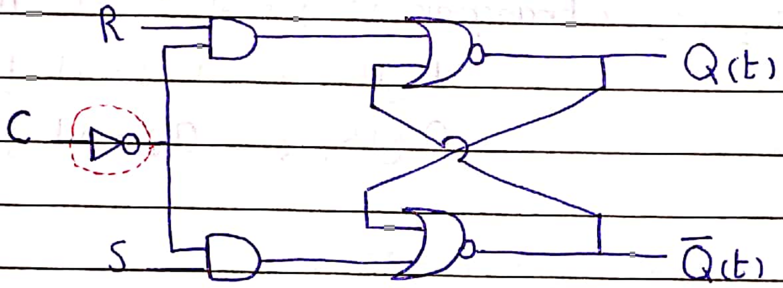
		Current		Future		
S	R	Q(t)	$\bar{Q}$(t)	Q(t+1)	$\bar{Q}$(t+1)	
1	0	0	1	1	0	اشتغل
0	0	(1	0)	1	0	ضل مثل ماهو
0	1	(1	0)	0	1	طفي
1	1	(0	1)	0	0	صفّر

*

	C(clock),AH	S	R	Q(t) $\bar{Q}$(t)
اشتغل طبيعي {	1	0	0	No change
	1	1	0	set
	1	0	1	Reset
	1	1	1	Forbidden
(no change) دائماً {	0	X	X	No change

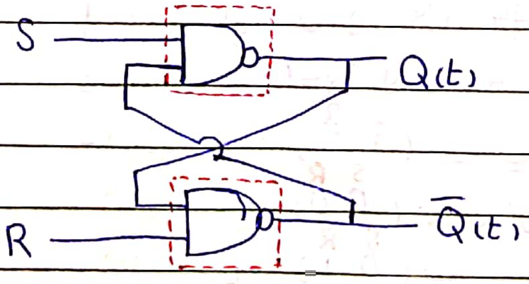


	C, AL	S	R	Q(t)	$\overline{Q}(t)$
اشتغل طبيعي	0	0	0	No change	
	0	1	0	set	
	0	0	1	Reset	
	0	1	1	Forbidden	
(no change) دائماً	1	X	X	No change	



* Active low SR Latch :

S	R	Q(t)	$\overline{Q}(t)$	
0	1	set		
1	1	No change		(العكس تماماً)
1	0	reset		
1	1	No change		
0	0	Forbidden		→ (خلّي كل القيم واصلات = 11)



* D - Latch :

C, AH	D	Q(t)
1	0	0
1	1	1
0	X	No change

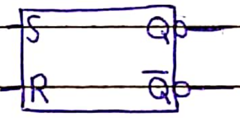
- كلمة AH و AL بتكون لل clock مش لل D
- ال D شفافة (transparent) يعني:

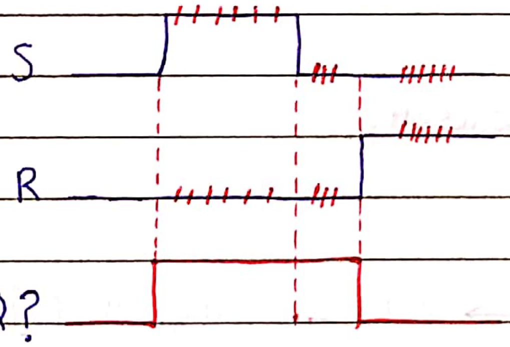
إذا دخل 1 ← بطلع 1

وإذا دخل 0 ← بطلع 0

* Examples :

1) $Q = 0$,





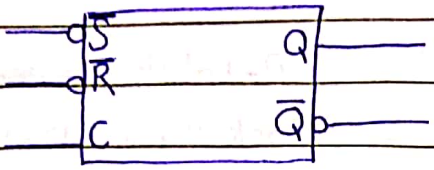
Solu: أولاً ال R , S ماعليهم Pubbles

إذاً شغلي على (AH SR)

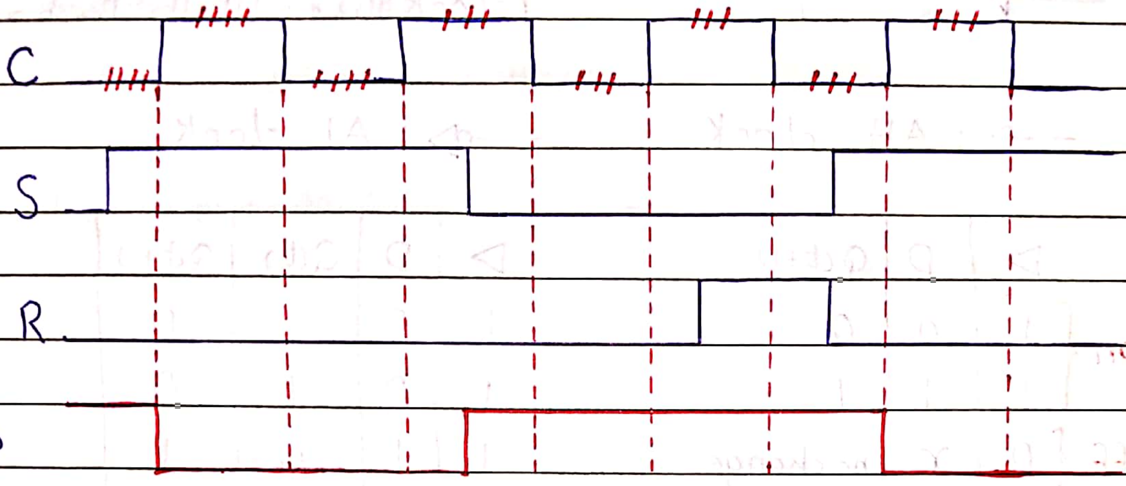
وبلش ب $Q=0$ لأنه معطيني بالسؤال

- أول تغير: ($\overset{S}{1}\ \overset{R}{0}$) يعني set (شغّل)
- ثاني تغير: ($\overset{S}{0}\ \overset{R}{0}$) يعني no change (ضل زي ما إنت)
- ثالث تغير: ($\overset{S}{0}\ \overset{R}{1}$) يعني Reset (طفّي)

2)



, Q = 1



Solu:

مشان ما نخربط ننزل خطوط تغير الـclock فقط.

وإشتغل بالعكس لأنه عندك $\bar{S}$, $\bar{R}$ معناها: AL, SR

($\overset{S}{0}$ $\overset{R}{1}$) → set

($\overset{S}{1}$ $\overset{R}{0}$) → reset

($\overset{S}{0}$ $\overset{R}{0}$) → forbidden = 11

($\overset{S}{1}$ $\overset{R}{1}$) → no change

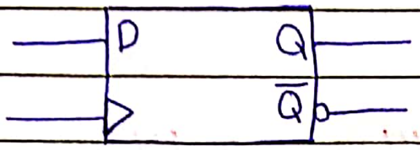
AH , C. و

1 → طبيعي

0 → no change

Given , Q=1 وبلش بـ

* D - Flip Flop :



* بميزه عن ال D-Latch
من خلال المثلث بتاع ال clock
(بحط مثلث للدلالة على ال clock)

—▷ : AH clock , —○▷ : Al clock

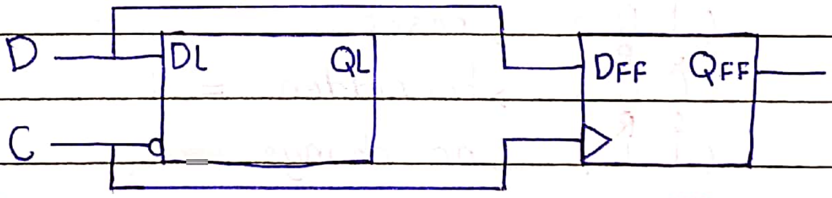
	▷	D	Q(t+1)
on	1	0	0
	1	1	1
off	0	x	no change

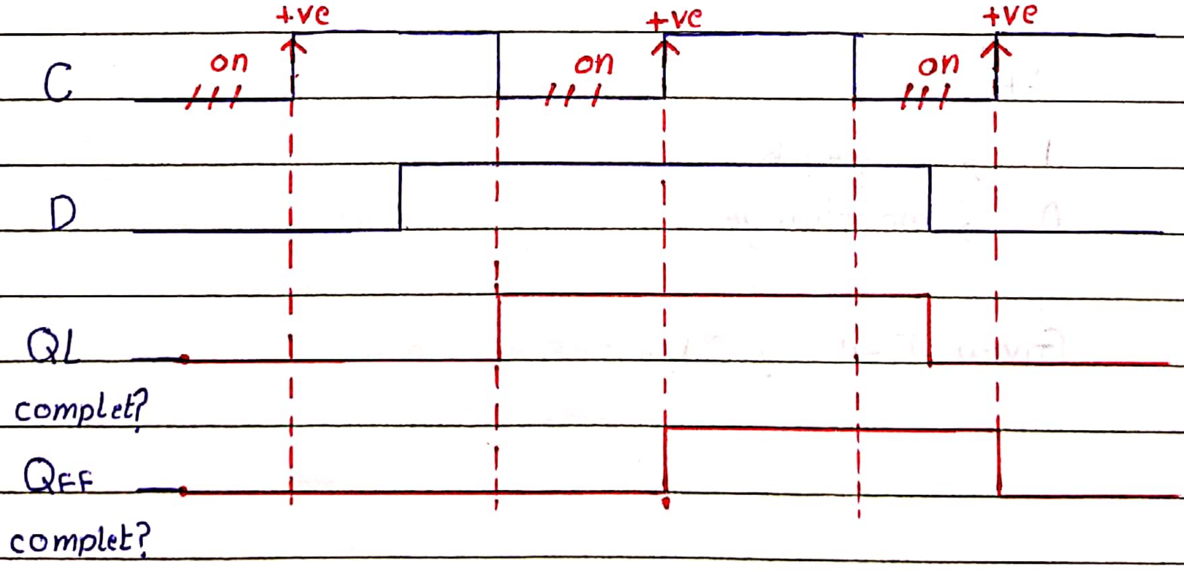
▷	D	Q(t)	Q(t+1)
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

ال D شفافة تذكر

∴ $(D = Q(t+1))$

* Example :





▶ Subject : .... / /

* ملاحظات :

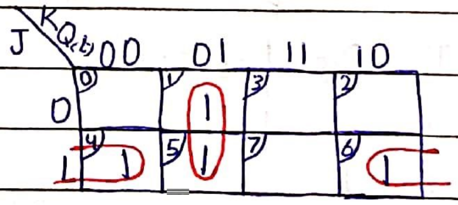
- بال Flip Flop التغير لحظي عند (+ve) أو (-ve edg)
حسب (AH) وال (ALFF) ، يعني ما بتهمني الفترة أو التغير الحاصل
بالفترة مش مهم ، لذا ما دون ال edg فهو no change .
- ال latch فترات بكون فيها التغير مستمر ومهم .

* JK Flip Flop :

J	K	$Q(t+1)$	$\bar{Q}(t+1)$	
0	0	no change		ضل مثل ما انت
0	1	reset		طفّي = $\overset{Q}{0}$ $\overset{\bar{Q}}{1}$
1	0	set		شغّل = $\overset{Q}{1}$ $\overset{\bar{Q}}{0}$
1	1	toggle		إعكس = العكس

ex : $01 \xrightarrow[\text{إعكس}]{} 10$

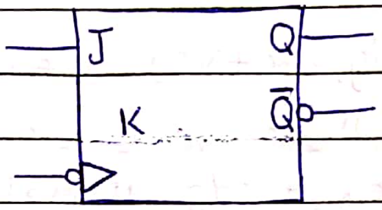
	J	K	$Q(t)$	$Q(t+1)$	
no change	0	0	0	0	ضل مثل ما كان
	0	0	1	1	
reset	0	1	0	0	طفي
	0	1	1	0	
set	1	0	0	1	اشتغل
	1	0	1	1	
toggle	1	1	0	1	انعكس
	1	1	1	0	

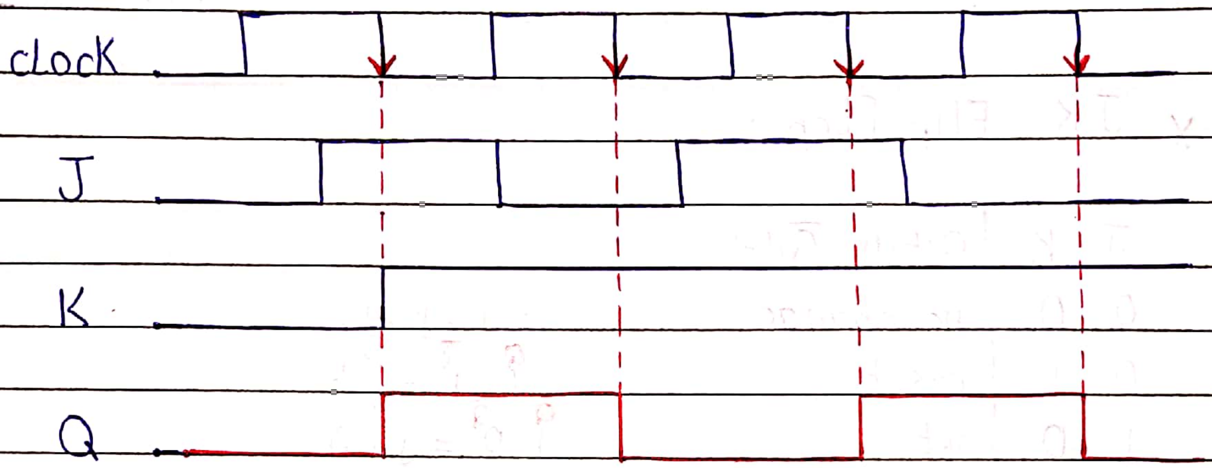


J \ K$Q(t)$	00	01	11	10
0	⓪	① 1	③	②
1	④ 1	⑤ 1	⑦	⑥ 1

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

* Example :





Draw Q ?

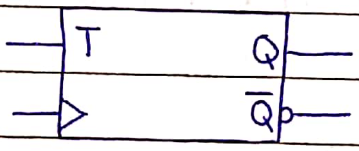
Solu:

—○▷ so (-ve edg)

J, K without Pubble so AH JK FF

* T- Flip Flop :

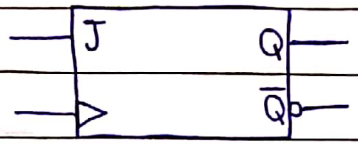
T	Q(t+1)
0	no change
1	Toggle

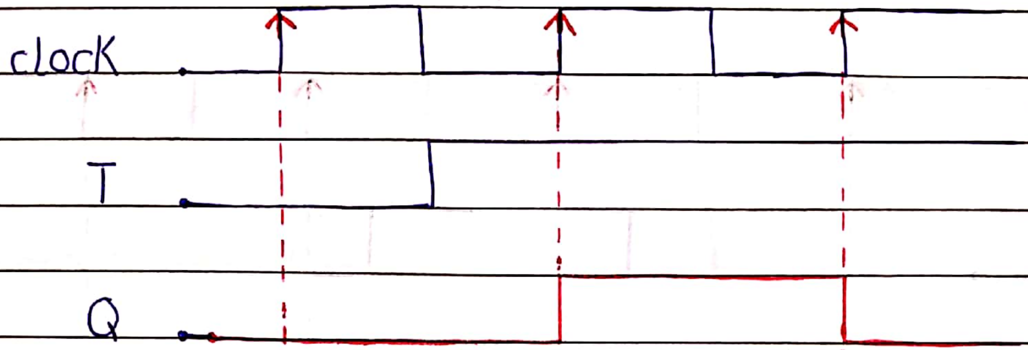


	T	Q(t)	Q(t+1)
no change	0	0	0
	0	1	1
Toggle	1	0	1
	1	1	0

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

* Example :





Draw Q?

—▷ , so (+ve edg)

T : without pubble so AH T-FF

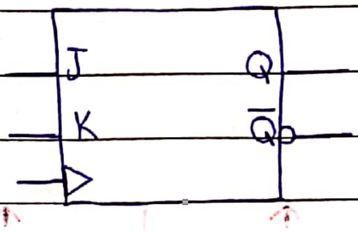
* Direct inputs :

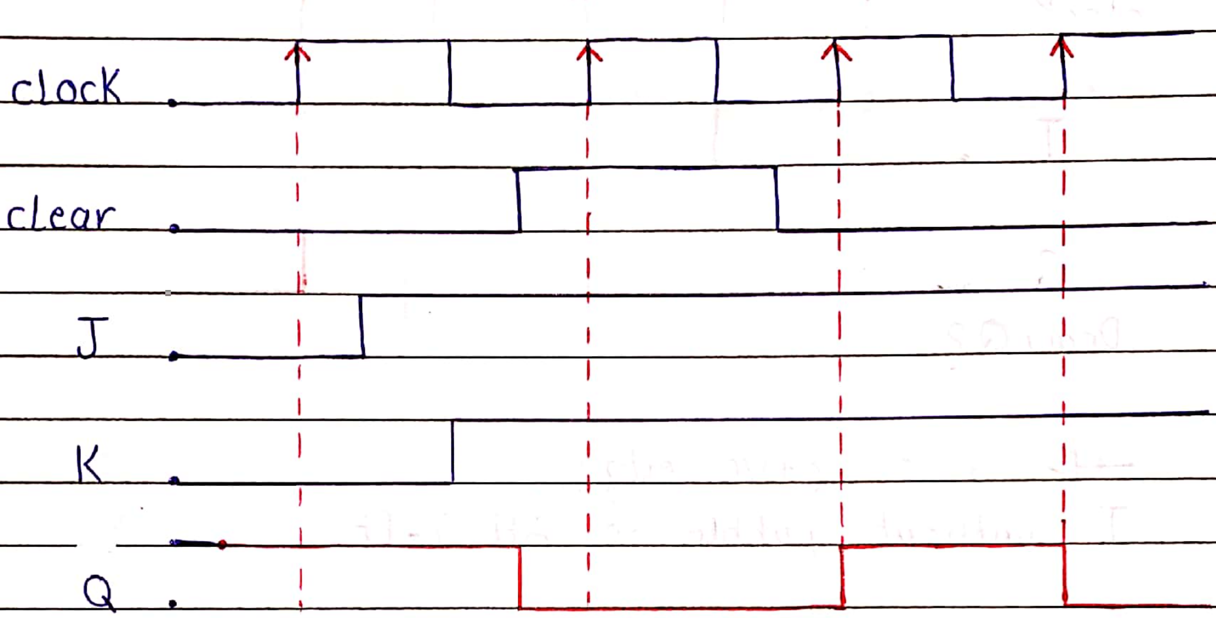
1) Direct reset → [clear] → دائماً صفر

2) Direct set → [Pre set] → دائماً واحد

يتم تنفيذهم في أي وقت (مابعتمدوا على ال clock)

* Example :





Draw Q?

$\rightarrow\triangleright$, so (+ve edg)

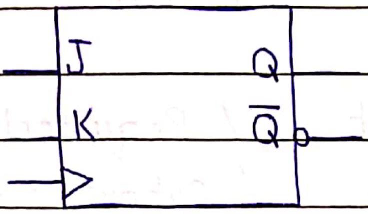
JK = AH

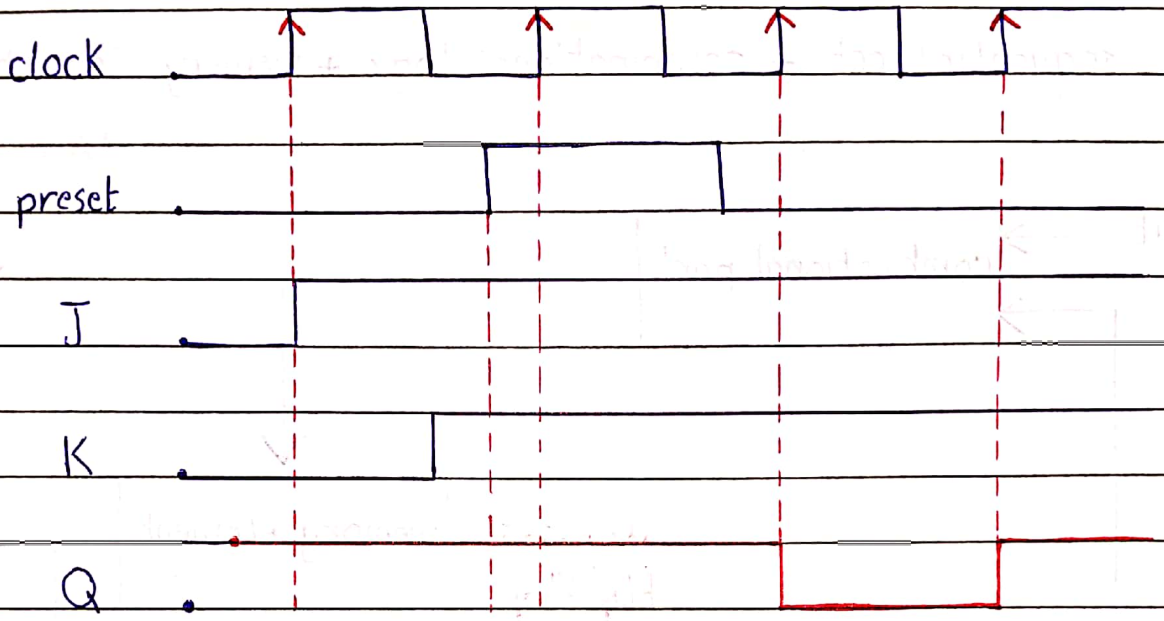
وانتبه لل clear

متى on = 1 ← صفّر كلشي طفّي

ومتى off = 0 ← طبيعي

* Example:





Draw Q ?

Solu:

+ve edg , AH JK , preset

شفل = 1

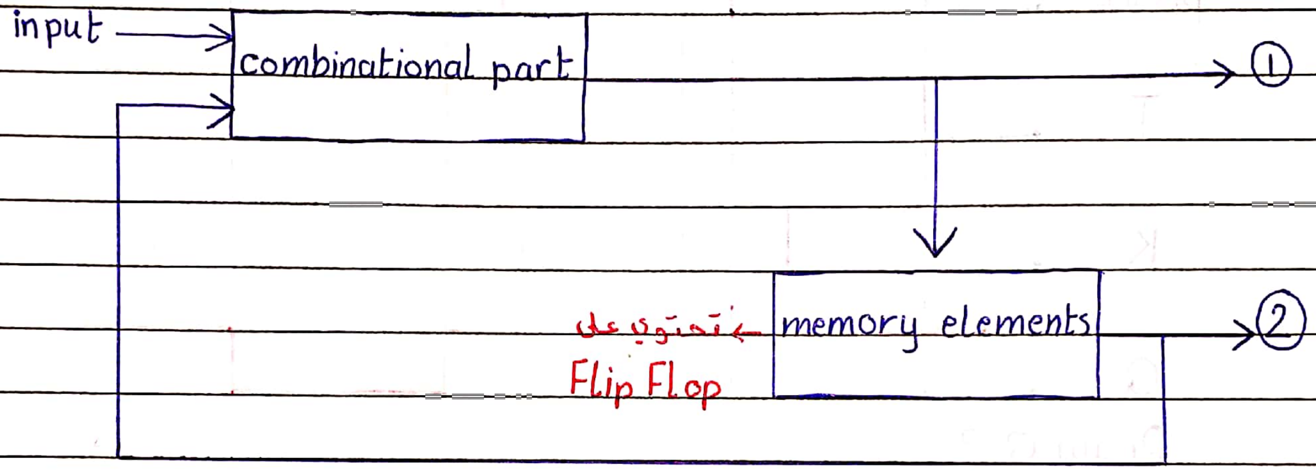
(لماتكون on)

(واشتغل طبيعي لماتكون off)

* Analysis of sequential ccts :

— Given : cct . / Required: state, Transition Table.
— input : Q(t) / Output : Q(t+1) .

— sequential cct = combinational logic + memory elements



① = meally cct : الناتج بيعتمد على ال input وعلى ال Q
② = moore cct : الناتج بيعتمد على Q فقط

* وال Q هي جاي من ال memory elements.
* ال state تبعت ال cct تعتمد على ال memory.

* [max # of states = 2^m
m : number of Flip Flops]

* Analyis steps :

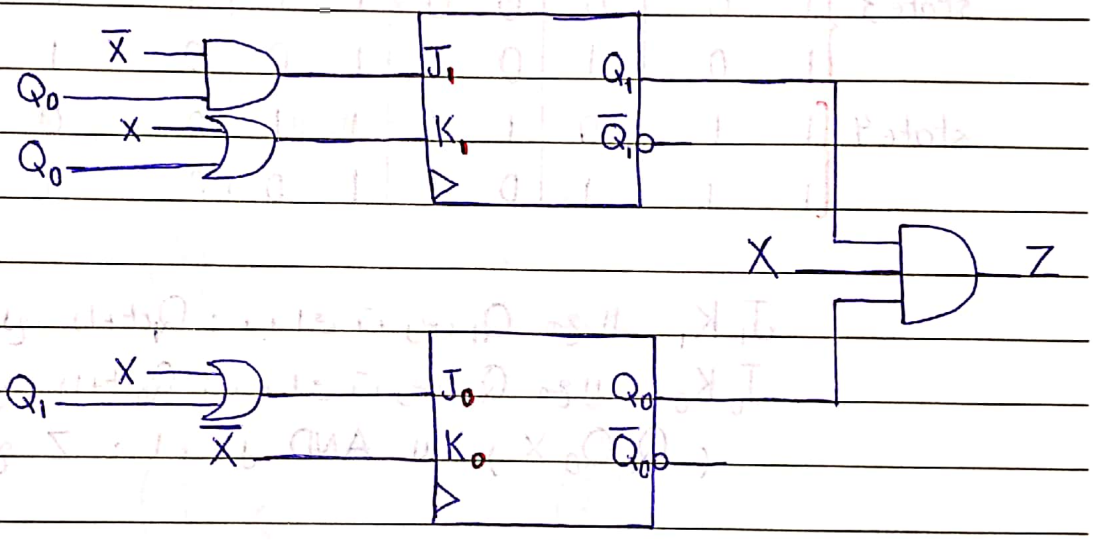
1) Find the boolean expression of the F.F inputs and outputs.

2) Find the values of the F.F , by (Truth table).

3) Find the next state (Q(t+1))

↳ رسمة

* Example :



Analyis this cct ?

Solu: سمّي J_1K_1 و J_0K_0 لحتى تميزهم عن بعض

① $J_1 = \overline{X} \cdot Q_0$

$K_1 = X + Q_0$

$J_0 = X + Q_1$

$K_0 = \overline{X}$

$Z = Q_1 \cdot X \cdot Q_0 \rightarrow Z = Q_1 Q_0 X$

② ترتيب الـ input والـ output بالـ Truth Table ثابت لكل الـ (analysis of JK F.F) ولازم تحفظه :

	current state		input					next state		output
	Q_1	Q_0	X	J_1	K_1	J_0	K_0	$Q_1(t+1)$	$Q_0(t+1)$	Z
state 1	0	0	0	0	0	0	1	0	0	0
	0	0	1	0	1	1	0	0	1	0
state 2	0	1	0	1	1	0	1	1	0	0
	0	1	1	0	1	1	0	0	1	0
state 3	1	0	0	0	0	1	1	1	1	0
	1	0	1	0	1	1	0	0	1	0
state 4	1	1	0	1	1	1	1	0	0	0
	1	1	1	0	1	1	0	0	1	1

لما نطلع $Q_1(t+1)$: بدك تدرس Q_1 مع الـ $J_1 K_1$

لما نطلع $Q_0(t+1)$: بدك تدرس Q_0 مع الـ $J_0 K_0$

لما نطلع Z : اعمل الـ AND للـ ($Q_1 Q_0 X$)

③ $m=2$, $\therefore 2^m = 2^2 = 4$ = # of states

in general:

* هيك بنمثل الـ state بشكل عام $\boxed{Q(t)} \xrightarrow{\text{input/output}} Q(t+1)$

* السهم بطلع من $Q(t)$ وبروح للـ $Q(t+1)$

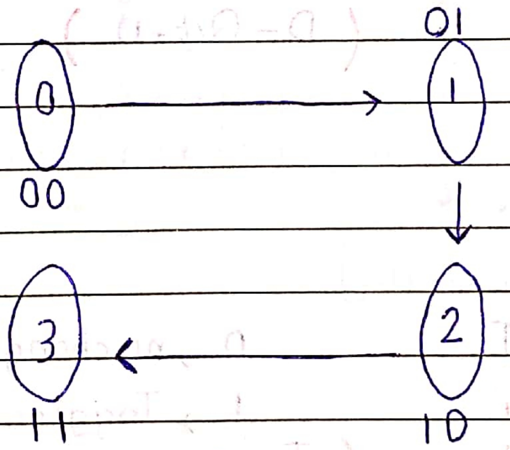
* الـ input / output فقط بكتبهم بكل حالة شو كانوا وبس مالهم تأثير على اتجاه السهم .

4 states and 2 Row for each state → لأنه بدك 2 current state

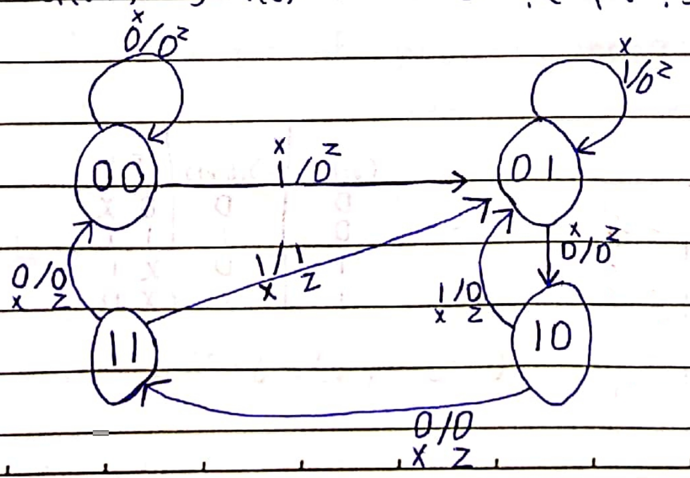
		State		Row1		Row2
State1	0	0 0	2 Rows :	X_1/Z_1	and	X_2/Z_2
state2	1	0 1	2 Rows :	X_1/Z_1	and	X_2/Z_2
state3	2	1 0	2 Rows :	X_1/Z_1	and	X_2/Z_2
State4	3	1 1	2 Rows :	X_1/Z_1	and	X_2/Z_2

* لازم تحفظ طريقة تمثيل أو (state Diagram)

أول إشي أكتب داخل دوائر الـ state الحالات من 0 إلى 3 كالتالي:



ثم ارسم الـ Row الي بتربطهم مع بعض الـ Q(t) والـ Q(t+1)
والـ input والـ output



∴ Rule # of rows for each case = 2^n where
n = # of inputs
$2^1 = 2$ لذا X واحد وهو input هون عندي

* Design of sequential cct :

*Excitation tables:

1) D-Flip Flop:

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

شفاف
(D = Q(t+1))

2) T-Flip Flop:

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

0 → no change
1 → Toggle
(T = Q(t) + Q(t+1))

3) JK-Flip Flop:

Q(t)	Q(t+1)	JK	
0	0	0 0 0 1	0 X
0	1	1 1 1 0	1 X
1	0	1 1 0 1	X 1
1	1	0 0 1 0	X 0

∴

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

(هدول بحفظهم مشان تشتغل عليهم بسرعة)

▸ Subject : .... / /

* How to Design :

1) state diagram.
2) assign Binary code for each state.
3) Find flip flop output valu. (Truth Table)
4) Find boolean expression. (K-map)
5) Draw.

* Example : Design a cct that detect if X (1-bit input) has the pattern "1001"? (use JK-F.F)

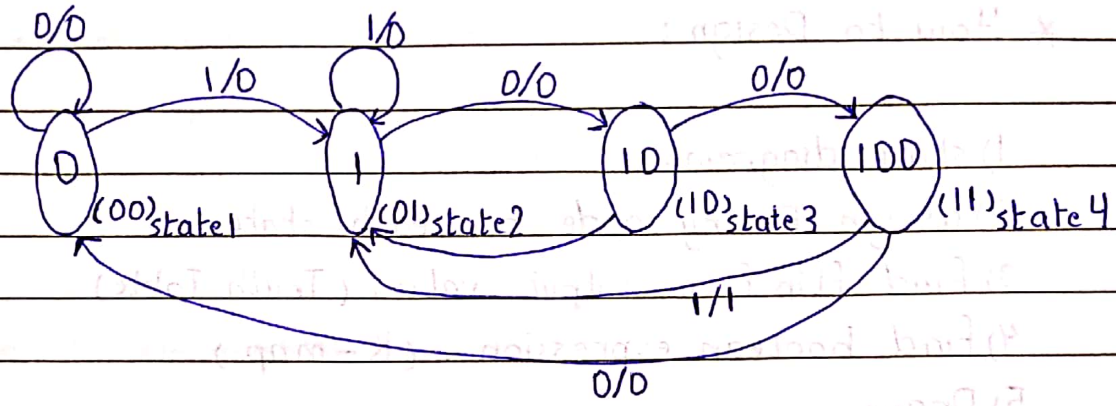
Solu:

				(بلش من جديد) ↑					(بلش من جديد) ↑				
ex :	$\overset{\times}{0}$	$\overset{\times}{0}$	$\overset{\times}{\overset{\checkmark}{1}}$	$\overset{\times}{\overset{\checkmark}{1}}$	$\overset{\times}{\overset{\checkmark}{0}}$	$\overset{\checkmark}{1}$	$\overset{\checkmark}{0}$	$\overset{\checkmark}{0}$	$\overset{\checkmark}{1}$	$\overset{\checkmark}{0}$	$\overset{\checkmark}{0}$	$\overset{\checkmark}{1}$	→ input
للتوضيح	0	0	0	0	0	0	0	0	1	0	0	1	→ output
			↓ (لساما اكتمل ال pattern)						↓ (اكتمل عندي ال pattern)			↓ اكتمل مرة ثانية	

1) 4-bit (pattern) = A B C D = 4 states
A: Pattern didn't start → 0
B: 1st bit of pattern appear → 1
C: 2nd bit of pattern appear → 1 0
D: 3rd bit of pattern appear → 1 0 0

$\underline{\underline{0}}$ 1 0 0 1
↓
لسا قبل ما نبلش
لازم تكون عكس أول Bit بالنمط

(كل input بفوت راح يكون يا صفر يا واحد)



* إذا شفت اللي بدك اياه، كمّل والـ out ما بصير 1 إلا لما
يكتمل الـ pattern .

بطلع الـ (Binary code) لكل state من الـ (Truth Table) 2) and 3)
نفسه وبرضه بطلع منه الـ output

	$Q_1(t)$	$Q_0(t)$	X	($Q_1(t+1)$	$Q_0(t+1)$)	(J_1 K_1	J_0 K_0)	Z
A	0	0	0	0	0	0 X	0 X	0
	0	0	1	0	1	0 X	1 X	0
B	0	1	0	1	0	1 X	X 1	0
	0	1	1	0	1	0 X	X 0	0
C	1	0	0	1	1	X 0	1 X	0
	1	0	1	0	1	X 1	1 X	0
D	1	1	0	0	0	X 1	X 1	0
	1	1	1	0	1	X 1	X 0	1

* لما تطلع الـ (next state) : بدك تدرس الـ (current state) مع الـ (input)
وتشوف السهم على أني (state) رايح .

* لما تطلع الـ J_1K_1 : بدك تدرس الـ $Q_1(t)$ مع الـ $Q_1(t+1)$.
ونفس الإشي للـ J_0K_0 من $Q_0(t)$ ، $Q_0(t+1)$.

Subject : .... / /

4) بعمل ال K-map ل J_1 , K_1 , J_0 , K_0 , Z فقط

3 input → 3 v K map

Q_1 Q_0 X

$Q_1 \backslash Q_0X$	00	01	11	10
0	^{0} 0	^{1} 0	^{3} 0	^{2} 1
1	^{4} 0	^{5} X	^{7} X	^{6} X

$J_1 = Q_0 \overline{X}$

$Q_1 \backslash Q_0X$	00	01	11	10
0	^{0} X	^{1} X	^{3} X	^{2} X
1	^{4} 0	^{5} 1	^{7} 1	^{6} 1

$K_1 = X + Q_0$

$Q_1 \backslash Q_0X$	00	01	11	10
0	^{0} 0	^{1} 1	^{3} X	^{2} X
1	^{4} 1	^{5} 1	^{7} X	^{6} X

$J_0 = Q_1 + X$

$Q_1 \backslash Q_0X$	00	01	11	10
0	X	X	0	1
1	X	X	0	1

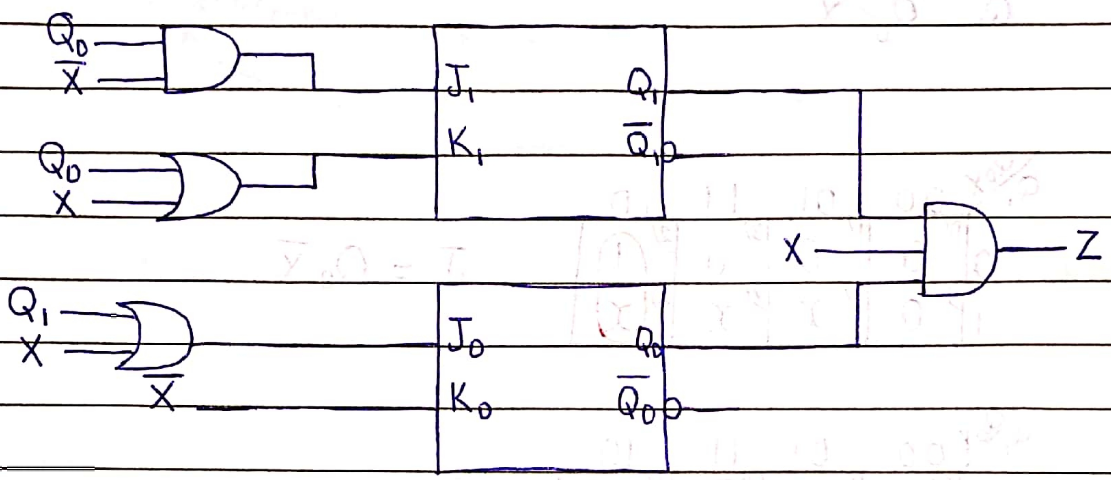
$K_0 = \overline{X}$

$Q_1 \backslash Q_0X$	00	01	11	10
0	0	0	0	0
1	0	0	1	0

$Z = Q_1 Q_0 X$

5) 4 states $= 2^m$

$\therefore m = 2$ (# of F.Fs)

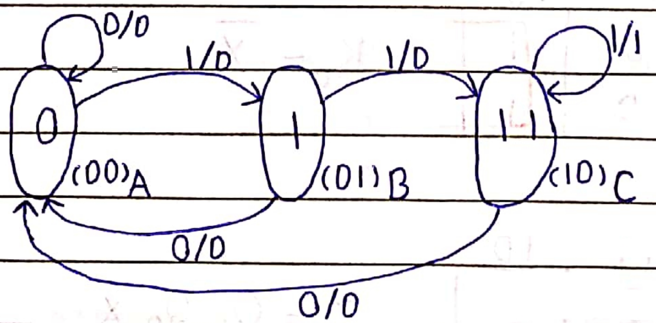


* Example: Designe a cct that detects 3 or more Cosequention 1's ?

Solu:

① $\underline{\underline{Q}}$ 1 1 1 .... (or more)

3 bit pattern at least = 3 state



لأنه قال or more (يعني ٣ أو ٤ عادي)
(1111) (111)

$2^m = 4$, $\therefore m = 2$

② and ③ :

	$Q_1(t)$	$Q_0(t)$	X	$Q_1(t+1)$	$Q_0(t+1)$	$J_1 K_1$	$J_0 K_0$	Z
A	0	0	0	0	0	0 X	0 X	0
	0	0	1	0	1	0 X	1 X	0
B	0	1	0	0	0	0 X	X 1	0
	0	1	1	1	0	1 X	X 1	0
C	1	0	0	0	0	X 1	0 X	0
	1	0	1	1	1	X 0	1 X	1
D	1	1	0	X	X	X X	X X	X
	1	1	1	X	X	X X	X X	X

④ : J_1 , K_1 , J_0 , K_0 , Z

Q_1 \ Q_0X	00	01	11	10
0	0	0	1	0
1	X	X	X	X

$J_1 = Q_0 X$

Q_1 \ Q_0X	00	01	11	10
0	X	X	X	X
1	1	0	X	X

$K_1 = \overline{X}$

Q_1 \ Q_0X	00	01	11	10
0	0	1	X	X
1	0	1	X	X

$J_0 = X$

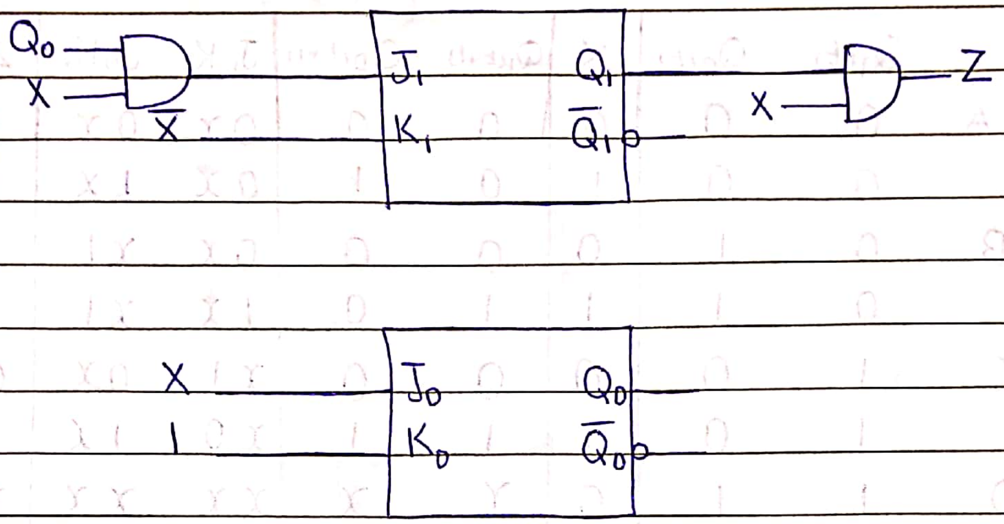
Q_1 \ Q_0X	00	01	11	10
0	X	X	1	1
1	X	X	X	X

$K_0 = 1$

Q_1 \ Q_0X	00	01	11	10
0	0	0	0	0
1	0	1	X	X

$Z = Q_1 X$

⑤



* Counters :

counters are synchronous

Regular (step width = 1):

Irregular

- up $0 \rightarrow 2^m - 1$ current +1
- down $2^m - 1 \rightarrow 0$ current -1
- up / down

ex : $(0 \rightarrow 7)$: 0, 1, 2, 3, 4, 5, 6, 7

$7 = 111 \rightarrow m = 3 \rightarrow 2^m - 1 = 7$

$(0 \rightarrow 2^m - 1) = (0 \rightarrow 7)$

∴ Regular

ex : $(0 \rightarrow 8)$:

$8 = 1000$, $m = 4$

$2^4 - 1 = 16 - 1 = 15$

$(0 \rightarrow 15) \neq (0 \rightarrow 8)$

∴ irregular

* Examples:

1) 0 , 1 , 2 , 3 , 4 , 0 , 1 , 2 , 3 , 4 , . . . ?

4 : 100 , m=3 $(2^3-1)=7$ irregular

2) 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 0 , . . . ?

7 : 111 , m=3 $2^3-1=7$ Regular

3) 0 , 2 , 5 , 3 , 2 , 5 , 3 , 2 , 0 ?

irregular

4) mod-8 counter ?

يعني باقي القسمة على 8 ، وهو كل الأعداد من 0 إلى 7

0 → 7 ∴ Regular

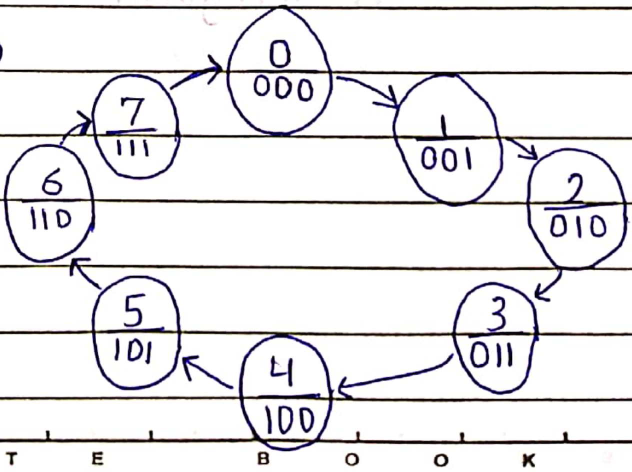
* Rule : # of Bits = m = # of flip flops .

*Example: Design (0 → 7) counter ?

Solu:

Regular up

Digram ①



② Truth Table

(*احفظ الترتيب)

(current state)			(next state)			(F.F)		
$Q_2(t)$	$Q_1(t)$	$Q_0(t)$	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$	T_2	T_1	T_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

③ K - map

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0			1	
1			1	

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0		1	1	
1		1	1	

$T_2 = Q_1 Q_0$

$T_1 = Q_0$

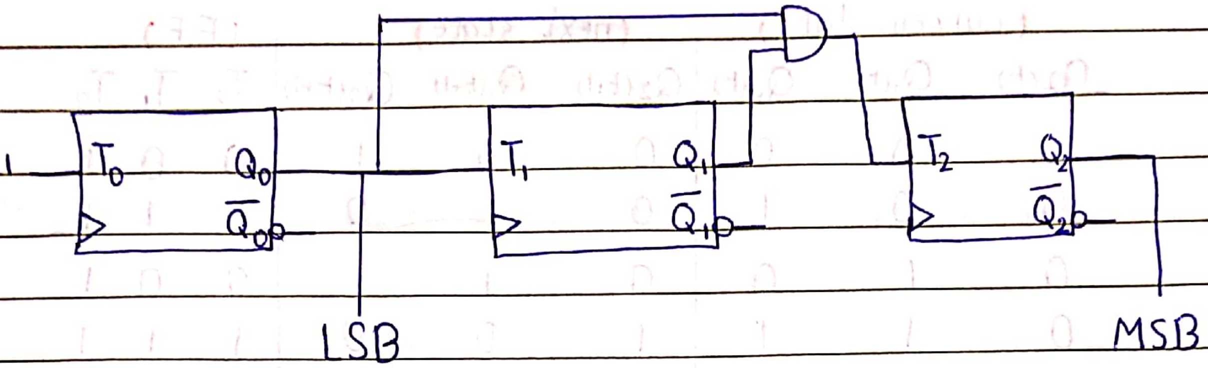
$T_0 = 1$

جدول العلاقات ثابتة دائماً لل Regular up counter ومثلاً لو عندك

كمان input [$T_3 = Q_2 Q_1 Q_0$] وهكذا .

Subject : .... / /

④ Draw cct.



خلص بتصير طول ترسمه بدون حاجة للخطوات السابقة لأنك حافظ العلاقات يعني حافظ الشكل.

او طلب تستخدم JK-F.F عادي نفس العلاقات:

$J_0 K_0 = 1$

$J_1 K_1 = Q_0$

$J_2 K_2 = Q_1 Q_0$

$J_3 K_3 = Q_2 Q_1 Q_0$

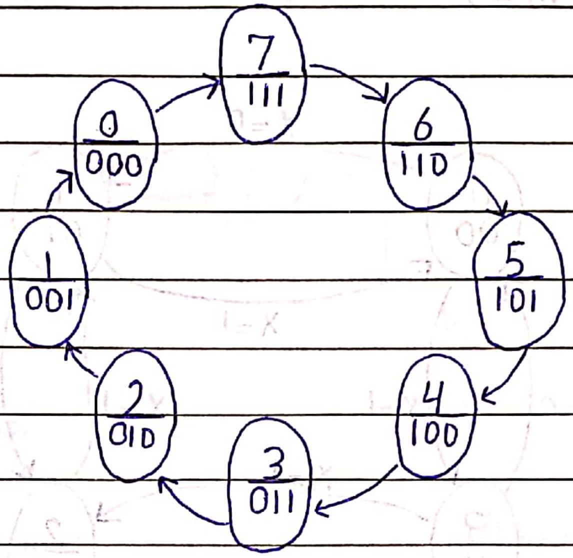
وهكذا...

S T A R S N O T E B O O K

▸ Subject : .... / /

* Example : Regular (7 → 0) Down counter (Design) ?

7 : 111   m = 3



Q_2	Q_1	Q_0	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$	T_2	T_1	T_0
0	0	0	1	1	1	1	1	1
0	0	1	0	0	0	0	0	1
0	1	0	0	0	1	0	1	1
0	1	1	0	1	0	0	0	1
1	0	0	0	1	1	1	1	1
1	0	1	1	0	0	0	0	1
1	1	0	1	0	1	0	1	1
1	1	1	1	1	0	0	0	1

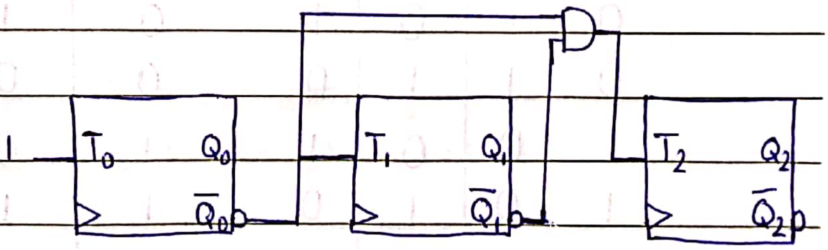
حفظ

($T_0 = 1$)

($T_1 = \overline{Q_0}$)

($T_2 = \overline{Q_1}\,\overline{Q_0}$)

وهكذا ...



نفس ال up لكن مع Bar

if JK - F.F : $J_0 K_0 = 1$

$J_1 K_1 = \overline{Q_0}$

$J_2 K_2 = \overline{Q_1}\,\overline{Q_0}$

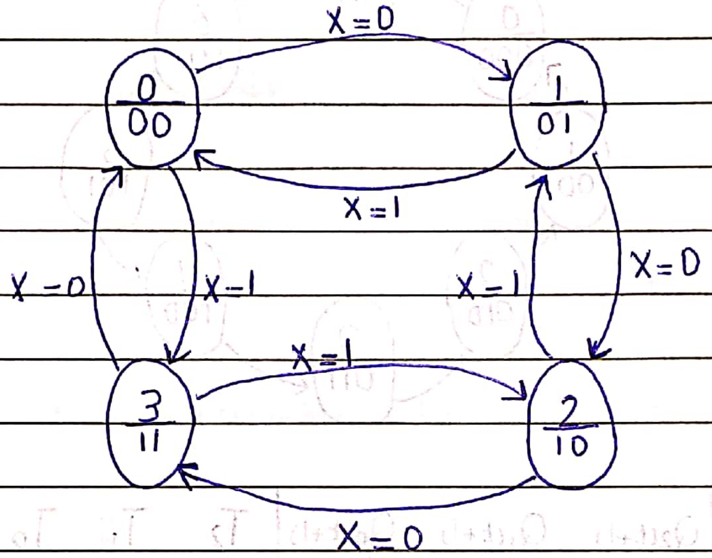
وهكذا ...

~~*~~ Example: Design Regular counter, (0→3) → X=0, up
(3→0) → X=1, Down ?

Solu:

3 : 11 m=2

①



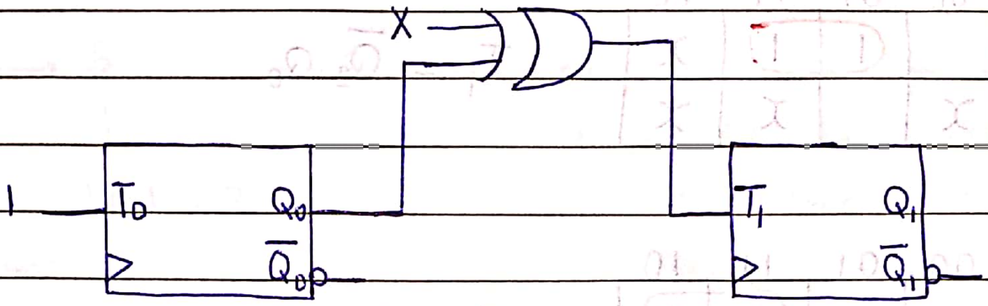
②

Q_1	Q_0	X	$Q_1(t+1)$	$Q_0(t+1)$	T_1	T_0
0	0	0	0	1	0	1
0	0	1	1	1	1	1
0	1	0	1	0	1	1
0	1	1	0	0	0	1
1	0	0	1	1	0	1
1	0	1	0	1	1	1
1	1	0	0	0	1	1
1	1	1	1	0	0	1

$Q_1 \backslash Q_0X$	00	01	11	10
0		1		1
1		1		1

$$T_1 = \bar{Q}_0 X + Q_0 \bar{X}$$

$$\therefore (T_1 = Q_0 + X) \quad , (T_0 = 1)$$



هون بال up/Down فش علاقة ثابتة لازم كل مرة تعمل analyis جديد.

*Example: Design a counter that counts 0, 1, 3, 5 ?

في طريقتين لحل الـ irrigular :

① Simplified : بشرطين ١) أن يكون (up) or (Down)

2) each step width = 1

② الطريقة الطبيعية والطويلة.

$$\begin{tikzcd} 0 \arrow[r, bend left] & 1 \arrow[r, bend left] & 3 \arrow[r, bend left] & 5 \arrow[lll, bend left] \end{tikzcd}$$

(step width ≠1 alwyas)

∴ بتنفعش الطريقة السهلة روح عالطويلة :

5 : 101 → m = 3

	Q_2	Q_1	Q_0	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$	T_2	T_1	T_0
0	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	1	0	1	0
2	0	1	0	X	X	X	X	X	X
3	0	1	1	1	0	1	1	1	0
4	1	0	0	X	X	X	X	X	X
5	1	0	1	0	0	0	1	0	1
6	1	1	0	X	X	X	X	X	X
7	1	1	1	X	X	X	X	X	X

Q_2 \ Q_1Q_0	00	01	11	10
0	1			X
1	X	1	X	X

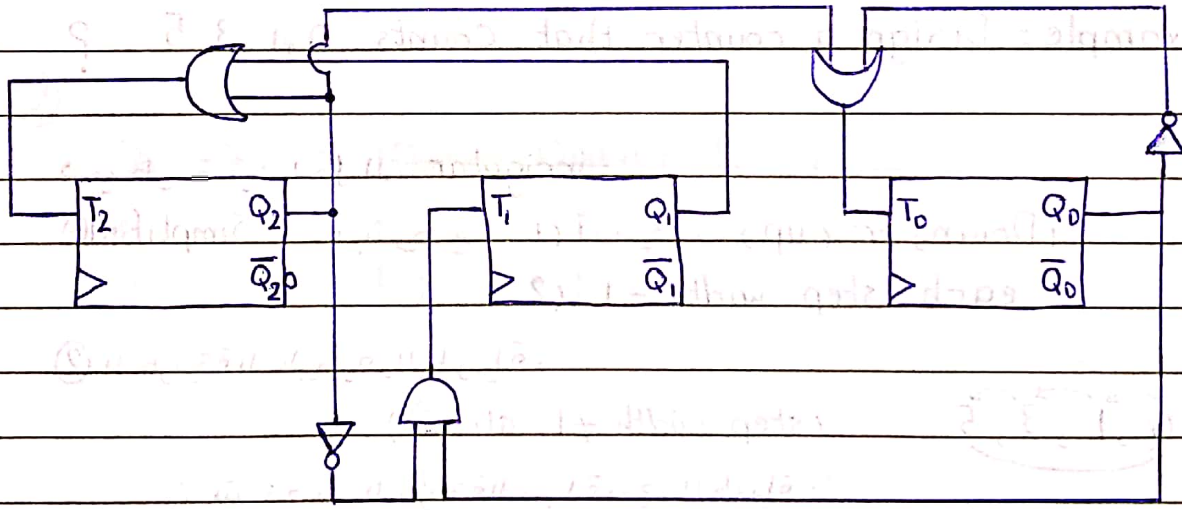
$T_0 = Q_2 + \overline{Q}_0$

Q_2 \ Q_1Q_0	00	01	11	10
0		1	1	X
1	X		X	X

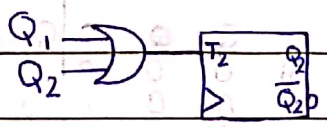
$T_1 = \overline{Q}_2 \, Q_0$

Q_2 \ Q_1Q_0	00	01	11	10
			1	X
	X	1	X	X

$T_2 = Q_2 + Q_1$



أو حطها هيك أسهل :



وهكذا ..

* Example : Design 0 → 5 counter ?

irregular

$(2^3 - 1 = 7 \neq 5)$, 5 : 101 , m = 3

0 → 5 , up and step = 1 , ∴ on simplified method

0 → 5

000   101 → 3 F.F

clear clear

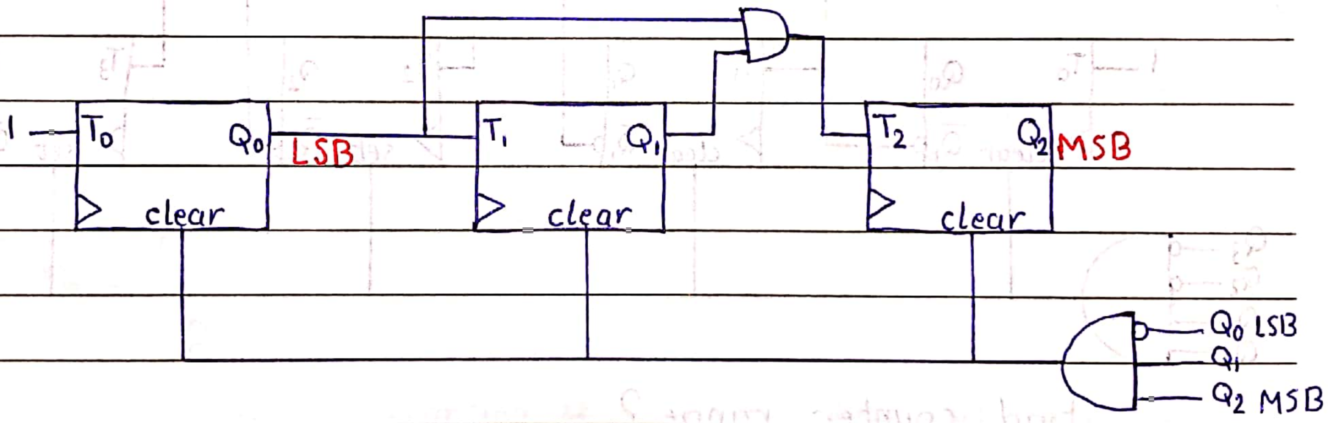
clear

بشوف الرقم اللي بلشت عنده وبمثله

بال Binary حسب عدد ال Bits المهم :

بستعمل clear لل 0

وبستعمل set لل 1



5 then 6 بشوف الرقم الأكبر مباشرةً من أعلى رقم بالعداد

1 1 0

MSB | LSB

$Q_2 \; Q_1 \; \bar{Q}_0$

(طبعاً ال 6 تظهر بشكل لحظي ثم بيبلش يعد من أول وجديد)

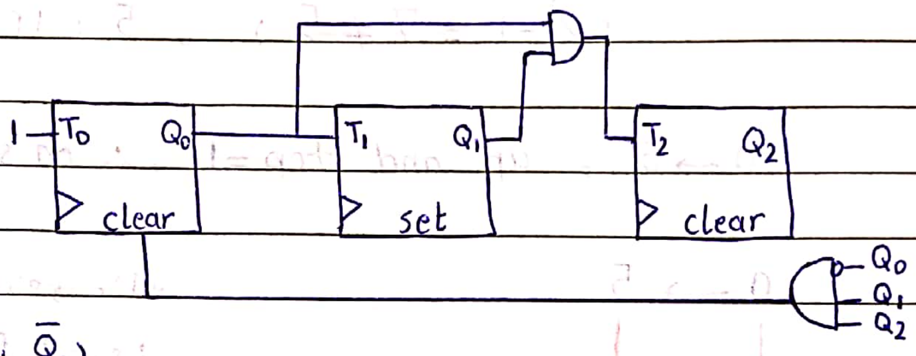
* Example : (2 → 5) counter ?

2 → 5

010 101

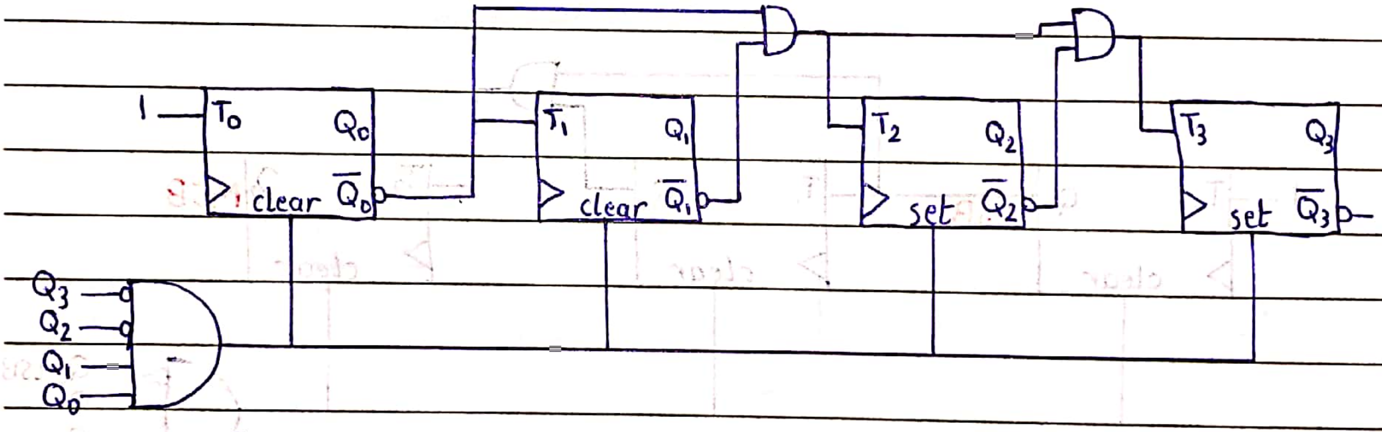
clear | clear

set



$6 \rightarrow 110 \rightarrow (Q_2\ Q_1\ \bar{Q}_0)$

* Example :



Find counter range ?

Down , inital # from clear and set :

$\left(\begin{array}{l} MSB \rightarrow Q_3 \\ LSB \rightarrow Q_0 \end{array} \right)$ بشكل عام

set set clear clear

∴ 1 1 0 0 = 12 Starting

8 4 ~~2~~ ~~1~~

⟹

MSB LSB

Final # from ($\overline{Q_3}$ $\overline{Q_2}$ Q_1 Q_0 on AND gate)

(0 0 1 1)

0 0 1 1

8 4 2 1

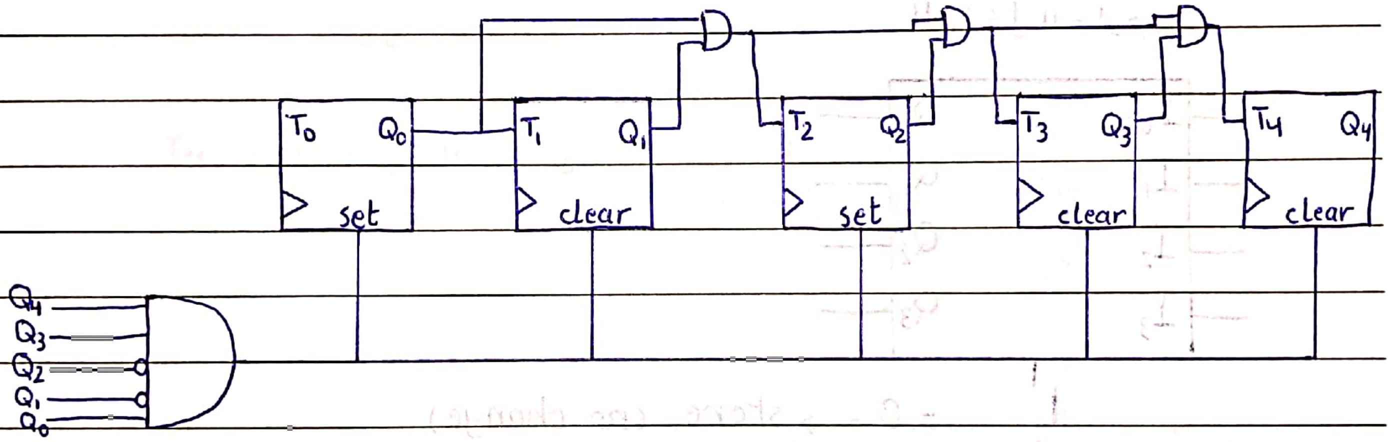
3 + 1 = 4

∴ Final # is (4)

So counter (12 ⟶ 4)

بطلع قيمتهم وبزيد عليها 1 لأنه
قيمتهم بتكون الرقم اللحظي اللي هو
أصغر ب 1 من أصغر رقم بالعداد.

* Example :



?

up , starting # = clear clear set clear set

(0 0 1 0 1)

4 2 1

= 5

Final # → Q_4 Q_3 $\overline{Q_2}$ $\overline{Q_1}$ Q_0

1 1 0 0 1

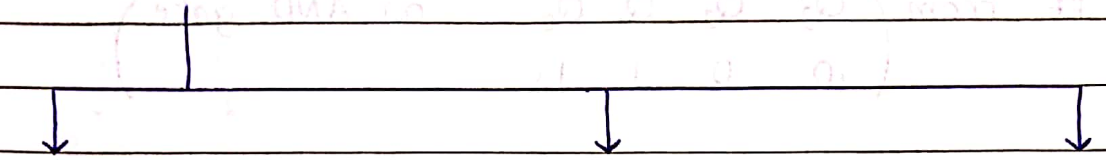
16 8 4 2 1 = 25

∴ 25 - 1 = 24

(5 ⟶ 24) counter.

الرقم اللحظي أكبر
بواحد من الرقم الفعلي.

* Registers : المسجلات

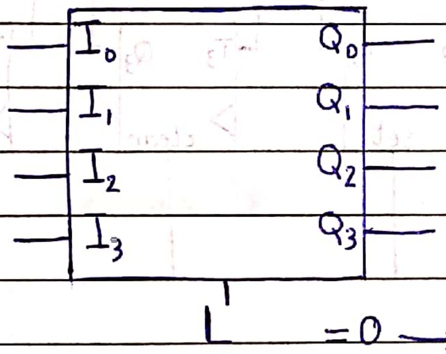


Parallel in / Parallel out   series in / series out   universal

* تعاملنا فقط مع (4-Bit Register).

* Parallel in / Parallel out : (المدخلات بدخلوا معاً وبطلعوا معاً بنفس الوقت)

الشكل الخارجي



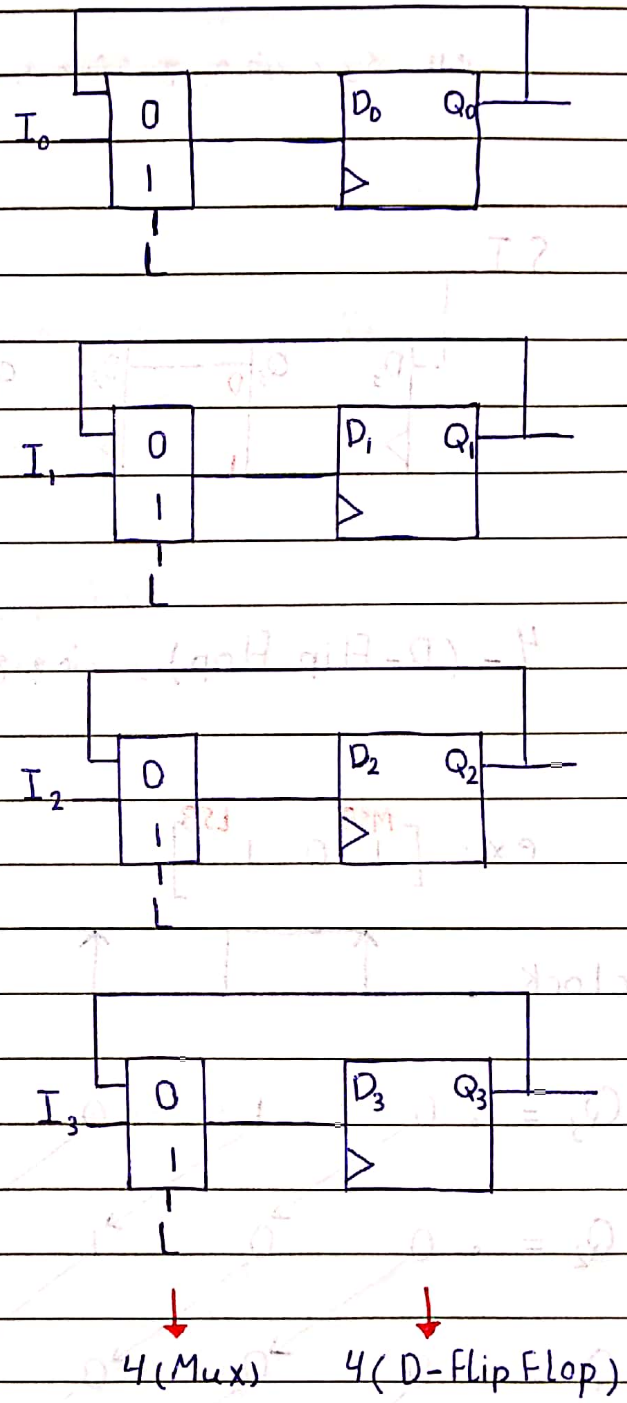
L = 0 → store (no change)

= 1 → loud (write)

بشتغل زي شغل الـ selector

* بحتاج 1 cycle مشان أكتب، وبعدين الـ output بطلع لحاله بدون أي cycle (وبنفس الوقت).

* الشكل الداخلي :



* لو كانت الـ (L = 0)
يعني Q_0 = no change
فبرجع زي مهو من
الـ feed back .

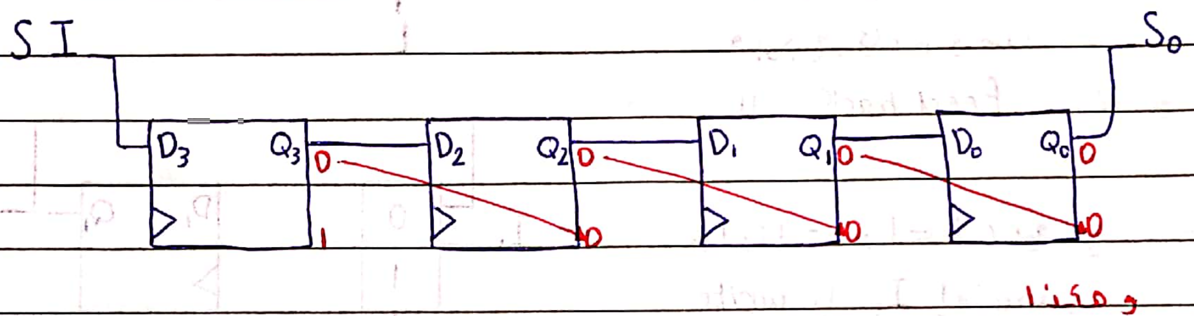
* لو كانت (L = 1) يعني
write يا I_0 إشي جديد
مشان أطلع إشي جديد للـ Q_0

ونفس الشي لكل الـ I's .

▸ Subject : .... / /

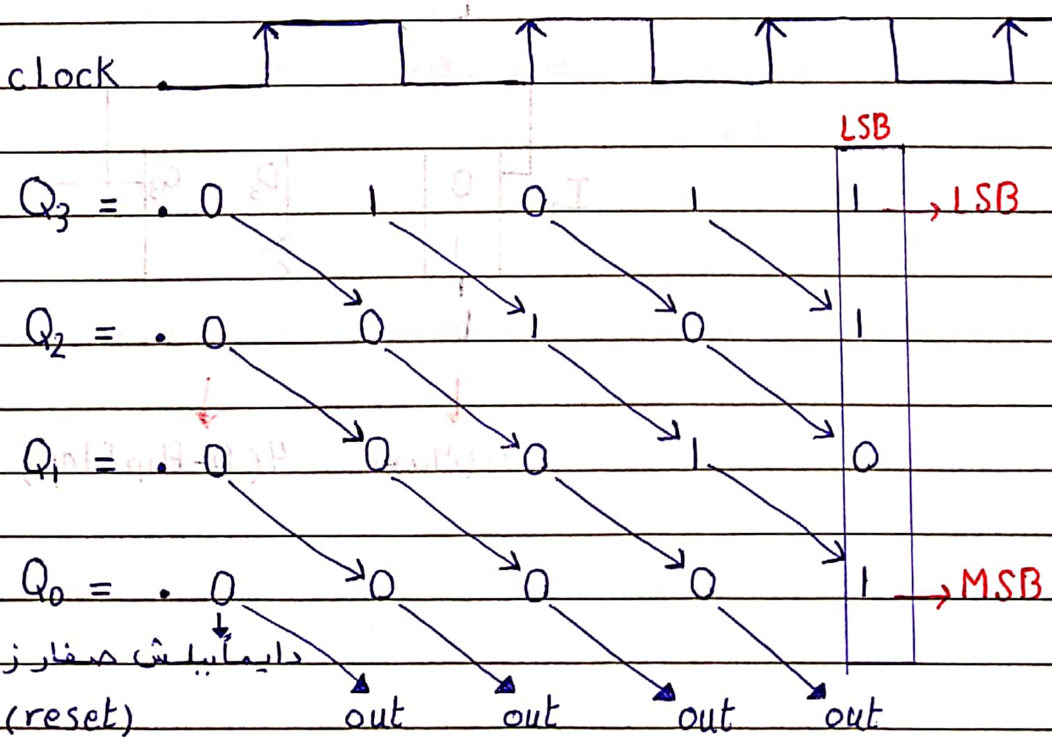
* series in / series out :

ال input بيدخل Bit و Bit بالترتيب مش مع بعض ، وكذلك
ال output بطلع Bit و Bit .



والـ clock بتضرب على كل ال D-F.Fs مع بعض ، 4 - (D-Flip Flop)
يعني :
ميك

ex : [1 0 1 1]
MSB LSB



Q_3 = . 0 1 0 1 1 → LSB
Q_2 = . 0 0 1 0 1
Q_1 = . 0 0 0 1 0
Q_0 = . 0 0 0 0 1 → MSB

دايماً بيبلش صفار زي
(reset)

شوف نفس الشغل
على نفس ال cct
كيف

* بالمثال السابق : * احتجت 4 cycle للكتابة :

→ 1 0 1 1 = 4 Bit = (4 cycle للكتابة)

* واحتجت 3 cycle للقراءة :

لأن الـ Bit الأولى (MSB) ما

بتحتاج لـ cycle هي already ظهرت عطول .

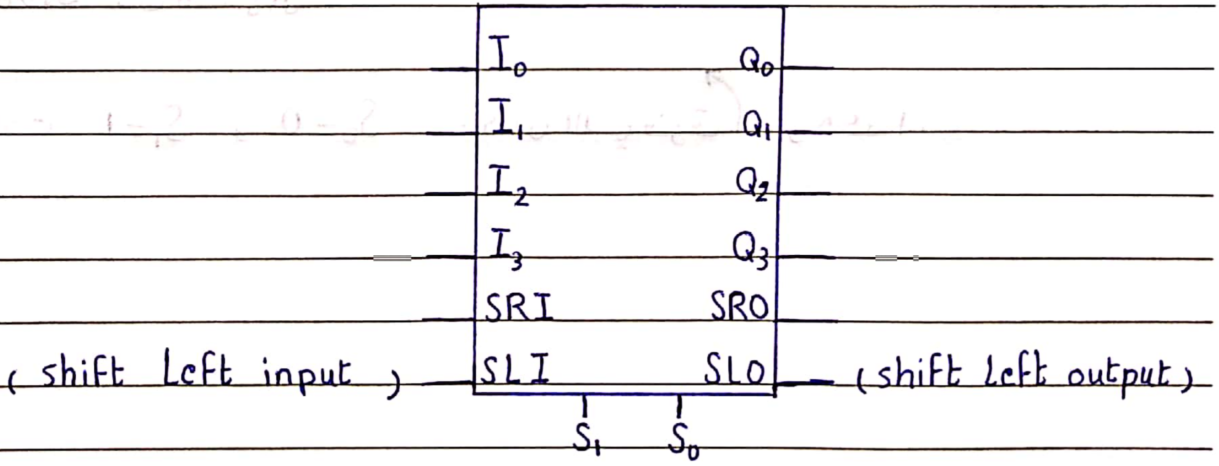
∴ 4 cycle for write and 3 cycle for Read

so 7 cycle we have .

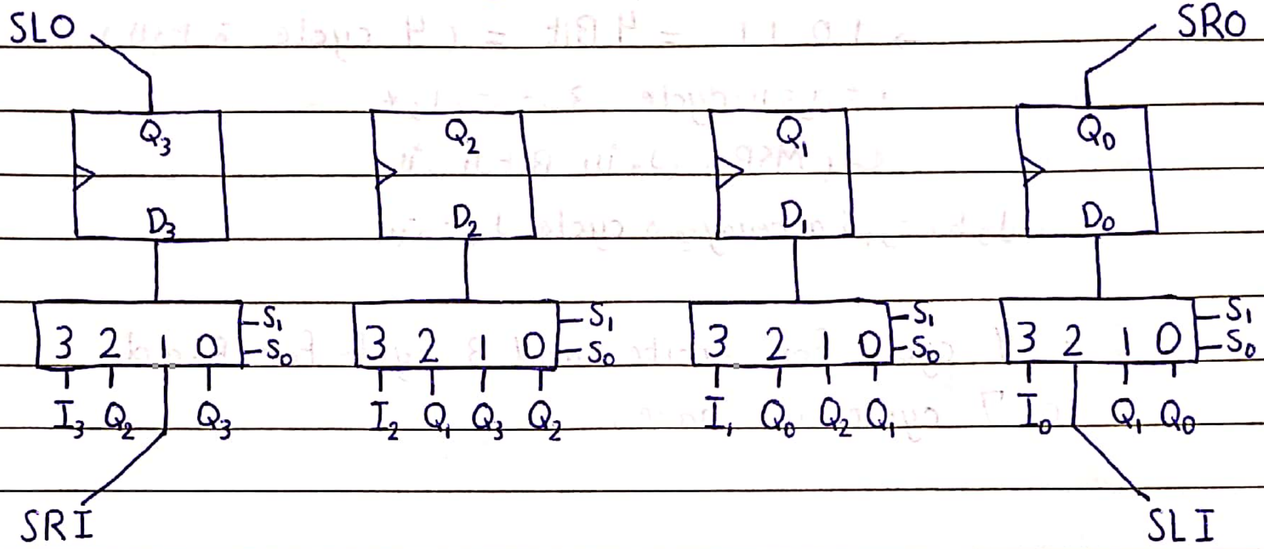
* universal Register : (أم منوع)

S_1	S_0	Type
0	0	store / no change
0	1	shift right (SR)
1	0	Shift Left (SL)
1	1	Parallel

الشكل الخارجي :



التصميم الداخلي :



* لو كانت $S_1=0$ و $S_0=0$ يعني (no change)
بتنزل Q_0 مكانها و Q_1 مكانها و Q_2 مكانها و Q_3 مكانها .

* لو كانت $S_1=0$ و $S_0=1$ يعني (shift Right)
بتنزل Q_3 على الـ mux الثاني
و Q_2 على الـ mux الأول
و Q_1 على الـ mux صفر
و على الـ mux الثالث SRI بتكون
Given من السؤال .

* لو كانت $S_1=1$ و $S_0=0$ عكس اللي فوق وهكذا ....

* Example :

$Q_3\ Q_2\ Q_1\ Q_0 = 1\,1\,0\,0$

SRI = 0

SLI = 1

$I_3\ I_2\ I_1\ I_0 = 11\ 11$

$S_1\ S_0 = 0\ 0$ cycle 1

1 0 cycle 2

0 1 cycle 3

Find $Q_3\ Q_2\ Q_1\ Q_0$ after 3 cycle ?

Solu :

	Q_3	Q_2	Q_1	Q_0	
After starte	1	1	0	0	
cycle 1	1	1	0	0	no change
cycle 2	1	0	0	$\boxed{1}$	shift Left
cycle 3	$\boxed{0}$	1	0	0	shift Right

$\therefore\ Q_3\ Q_2\ Q_1\ Q_0 = 0\,1\,0\,0$ after 3 cycle .

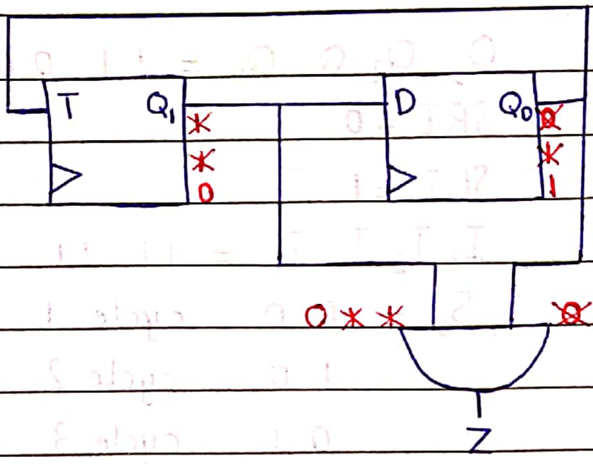
Subject : .... / /

* Example :

Q_1 Q_0

1 0

Find Z and Q's after 2 cycles ?



Solu :

$Q_1 = 0$

$Q_0 = 1$

$Z = 0\ 1\ 0$

0 → initial Bit

1 → after cycle 1

0 → after cycle 2

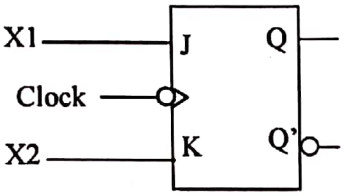
* شرح :

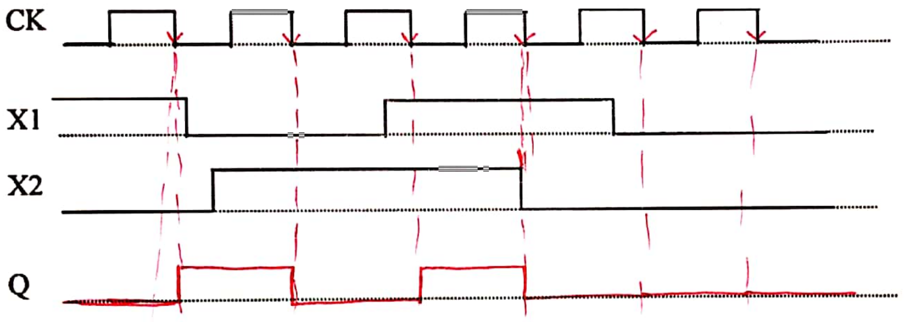
أعطاني قيم Q_1 Q_0 initial
(1 0)

- بمشيهم على السلك الأول (السلك الي بعطيك Z بعد AND (سلك ال output)
وبطلع ال Onitial Bit لل (Z) .
- ثم بمشيهم على السلك الثاني (السلك الداخل على ال memory element)
وبطلع القيم الجديدة لل Q_1 Q_0 وبشطب القيم القديمة.
- ثم بمشيهم على سلك ال output وبطلع Z بعد 1 cycle وهكذا
بكرر العملية حسب عدد ال cycles المطلوب .

The Hashemite University
Computer Engineering Department
Digital Logic (110408220)
HW5

Problem 1) Given the JK flip-flop circuit below, complete the following timing diagram by determining the waveform of the output Q (ignore setup and hold time requirements and assume propagation delays to be negligible). Assume Q is initially 0

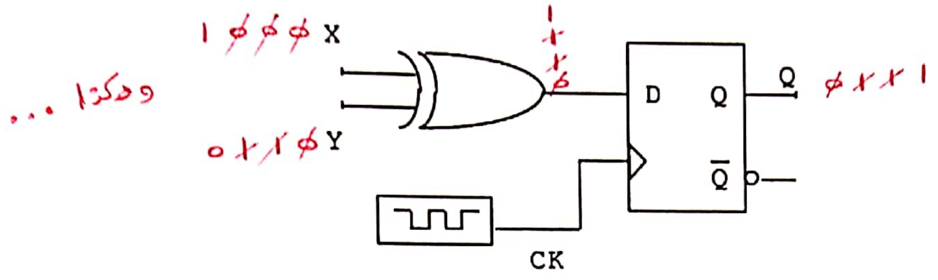


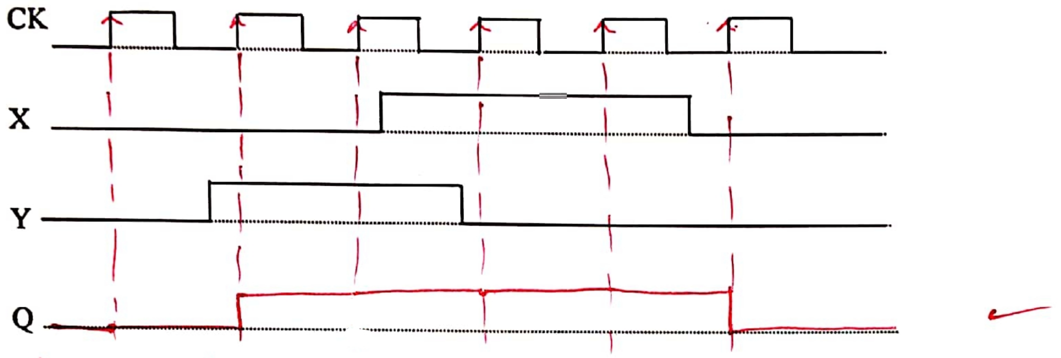


✓

Solu: -ve edg, since −o▷ ✓

Problem 2) : Given the D flip-flop circuit below, complete the following timing diagram by determining the waveform of the output Q (ignore setup and hold time requirements and assume propagation delays to be negligible). Assume Q is initially 0.





[illegible] +ve edg

The following questions are from the text book (5th Edition)

Problem 5.2) Construct a *JK* flip-flop using a *D* flip-flop, a two-to-one-line multiplexer, and an inverter.

Problem 5.3) Show that the characteristic equation for the complement output of a *JK* flip-flop is:
$Q'(t+1) = J'Q' + KQ$

Problem 5.4) A *PN* flip-flop has four operations: clear to 0, no change, complement, and set to 1, when inputs *P* and *N* are 00, 01, 10, and 11, respectively.
(a) Tabulate the characteristic table. (b) Derive the characteristic equation.
(c) Tabulate the excitation table. (d) Show how the *PN* flip-flop can be converted to a *D* flip-flop.

* HW.5 :

5.2) JK by Mux (2x1) and D

	J	K	Q(t)	D Q(t+1)
I_0	0	0	0	0
	0	0	1	1
	0	1	0	0
	0	1	1	0
I_1	1	0	0	1
	1	0	1	1
	1	1	0	1
	1	1	1	0

J \ KD	00	01	11	10
0		1		
1	1	1		1

$$\left(D = J\bar{Q} + \bar{K}Q \right)$$

J	K	I_0
0	0	X
0	1	0
1	0	1
1	1	X

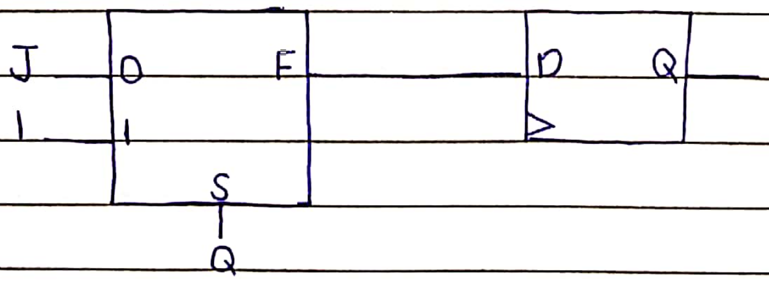
J \ K	0	1
0	X	
1	1	X

$I_0 = J$

J	K	I_1
0	0	X
0	1	X
1	0	1
1	1	X

J \ K	0	1
0	X	X
1	1	X

$I_1 = 1$



5.3)

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

$$\overline{Q}_{(t+1)} = \overline{\overline{J}\,\overline{Q} + \overline{K}\,Q}$$

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

J \ KQ	00	01	11	10
0	1		1	1
1			1	

$$\overline{Q}_{(t+1)} = \overline{J}\,\overline{Q} + KQ$$

5.4)

a)

P	N	
0	0	clear = 0
0	1	no change
1	0	complement (not)
1	1	set = 1

b)

P	N	Q	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

P \ NQ	00	01	11	10
0			1	
1	1		1	1

$$\left(Q_{(t+1)} = P\overline{Q} + NQ \right)$$

c) and d)

Q(t)	Q(t+1)	P N	
0	0	0 X	→ clear or no change
0	1	1 X	→ not or set
1	0	X 0	→ not or clear
1	1	X 1	→ set or no change

Connect P and N together to get (D-F.F).