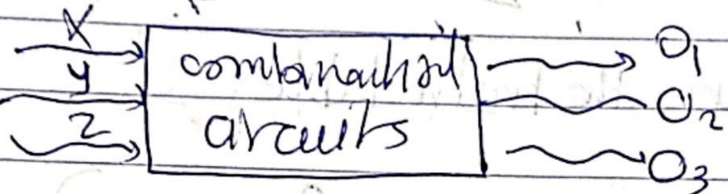
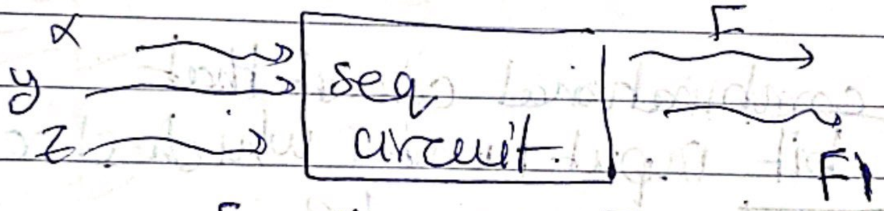
Ch 4 combinational circuits (Design)

Two types of digital circuits:

① combinational circuits (Ch 4): output only depends on input

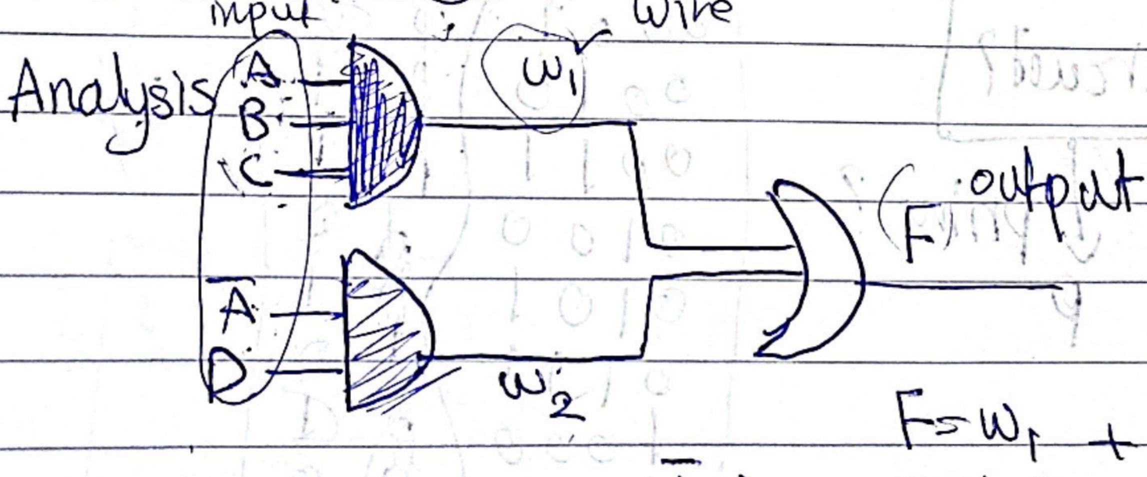


② sequential circuit (ch5 + ch6): output depends on inputs and previous outputs.



$F = x, y, z, F_1$

combinational circuit



$F = w_1 + w_2$

$w_1 = A.B.C$ $\quad w_2 = \bar{A}.D$ $\quad F = A.B.C + \bar{A}.D$

output input

Design procedure ⇒

① you need to read the problem carefully.

② Determine the number of inputs & outputs.

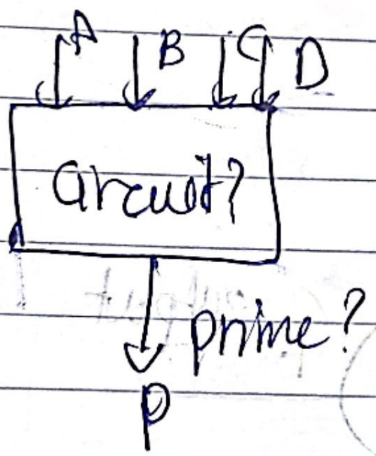
③ Assigns letter symbols to inputs & outputs.

④ determine the truth table.

⑤ Simplify the boolean function.

⑥ Design the circuits.

Ex. Design a combinational circuit that takes 4 bit input number which check if the number is prime or not?



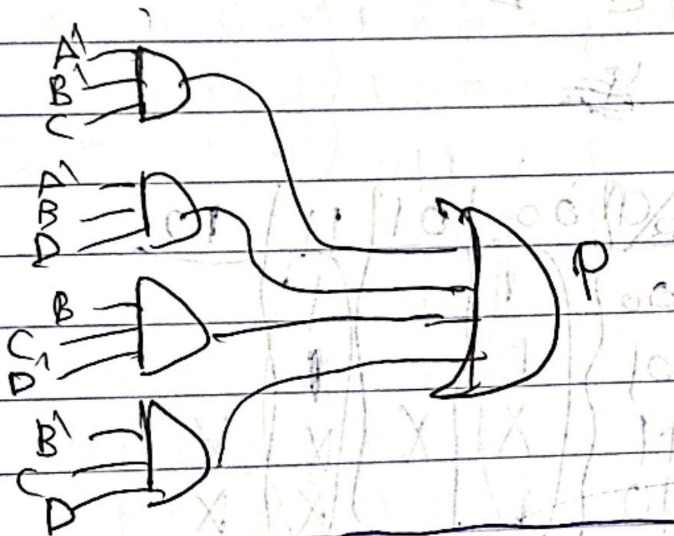
A	B	C	D	P
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0

⋮

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

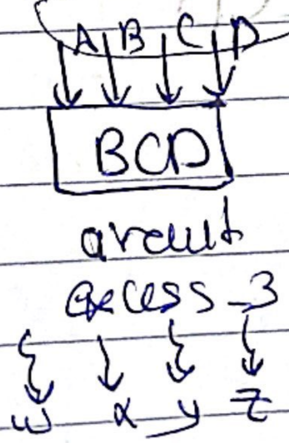
→ A'B'C
→ A'BD
→ B'CD
BC'D

$P = A'B'C + A'BD + BC'D + B'CD$



Design BCD to excess-3 code conversion?

BCD	excess
0000	0011



input (BCD) A B C D	output (excess 3) w x y z
0 0 0 0	0 0 1 1
0 0 0 1	0 1 0 0
0 0 1 0	0 1 0 1
0 0 1 1	0 1 1 0
0 1 0 0	0 1 1 1
0 1 0 1	1 0 0 0
0 1 1 0	1 0 0 1
0 1 1 1	1 0 1 0
1 0 0 0	1 0 1 1
1 0 0 1	1 1 0 0
x x x x	x x x x
x x x x	x x x x
x x x x	x x x x
x x x x	x x x x
x x x x	x x x x
x x x x	x x x x

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

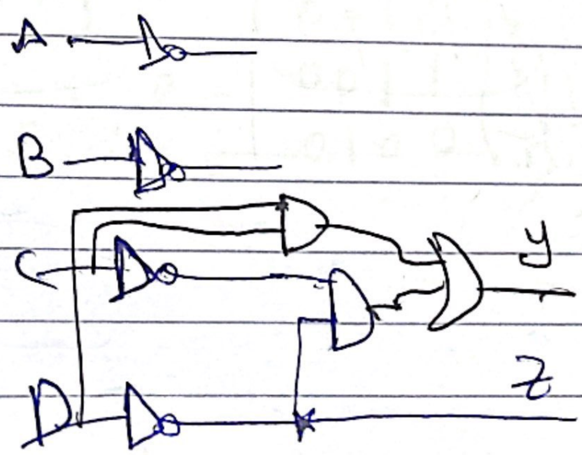
$Z = D'$

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

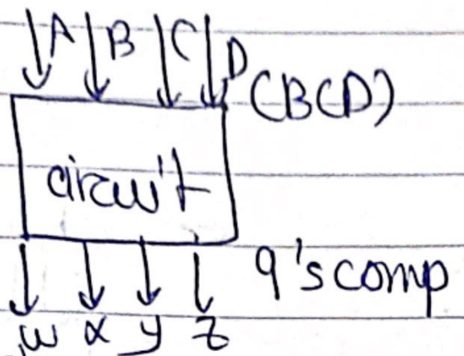
10
12
13 11
14
15

$y = C'D' + CD$

يمكن نعمل لـ X ولـ س



design a combinational circuit that generate the 9's comp of a BCD digit?



9' complement BCD

1001
1000
0001

(1000)

input(BCD)

A	B	C	D	w	x	y	z
0	0	0	0	1	0	0	1
0	0	0	1	1	0	0	0
0	0	1	0	0	1	1	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	1
0	1	1	1	0	0	1	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x

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

$z = D'$

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

$y = C$

x , w ⇐ بنحلها

Digital

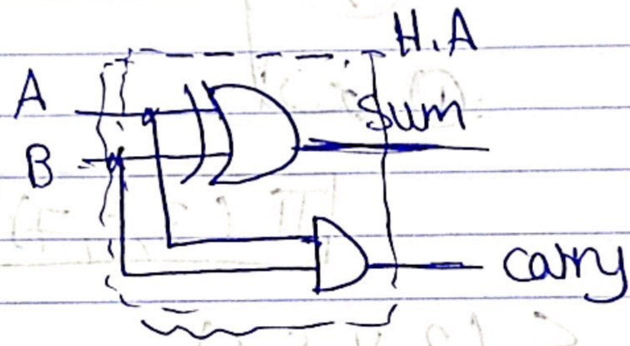
1) Half Adder

$$\begin{array}{r} A \\ +B \\ \hline \end{array} \quad \begin{array}{r} 0 \\ +0 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ +0 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array}$$

A	B	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Sum = $A \oplus B$

Carry = AB



2) Full adder

$$\begin{array}{r} A \\ +B \\ \hline \end{array} \quad \begin{array}{r} 1 \\ 11 \\ 11 \\ \hline 0 \end{array}$$

A	B	C_{in}	C_{out}	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

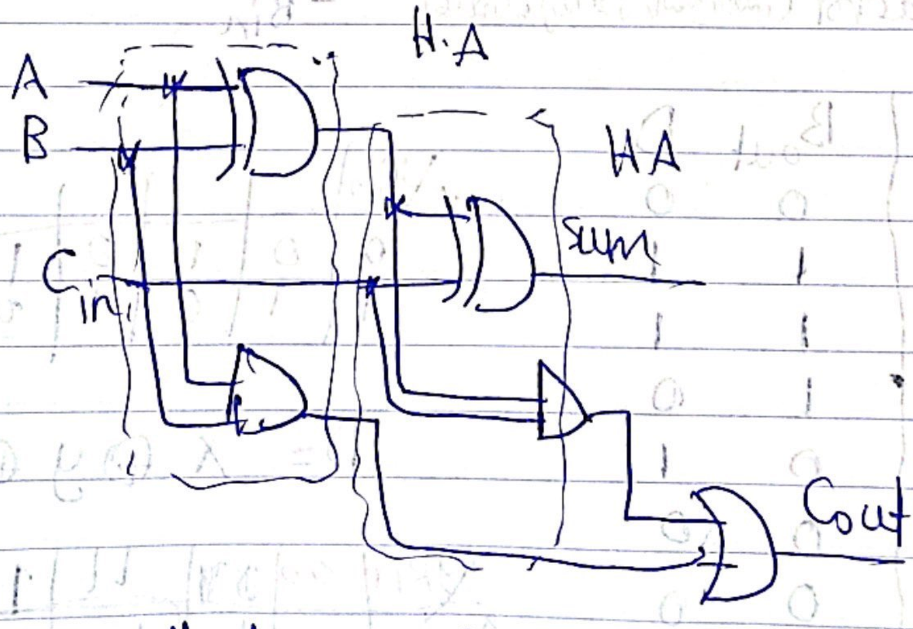
A \ BC_{in}	00	01	11	10
0	0	1	0	1
1	1	0	1	0

حل بطريقة ⊕

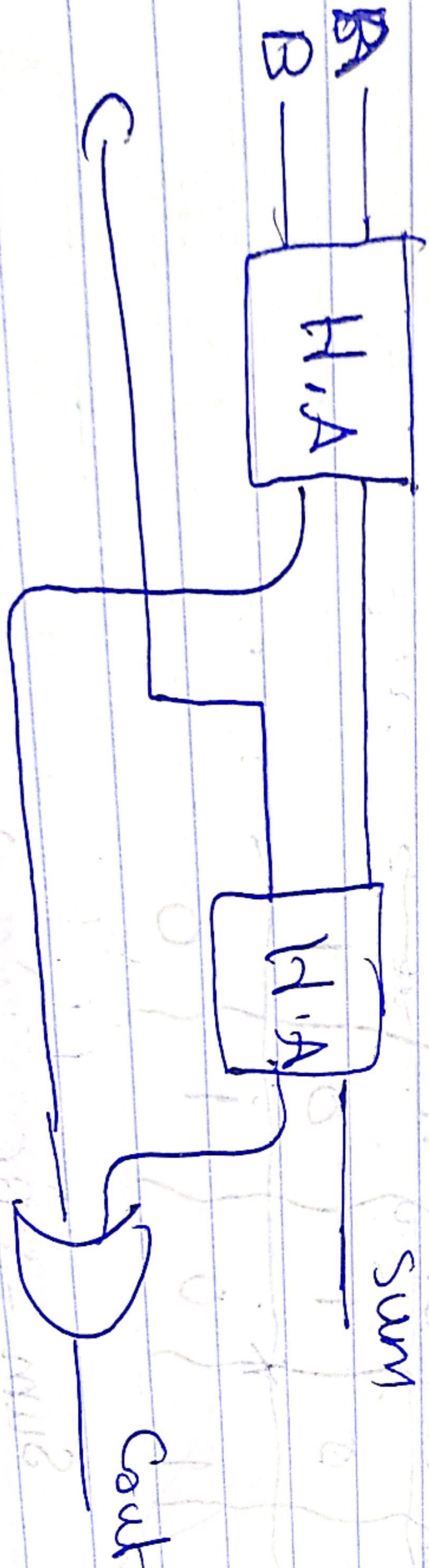
$$Sum = A'B'C + A'BC' + AB'C' + ABC$$
$$= A'(B'C + BC') + A(B'C' + BC)$$
$$A'(B \oplus C) + A(B \odot C)$$
$$A'(B \oplus C) + A(B \oplus C)'$$
$$= A \oplus (B \oplus C)$$

A \ BC_{in}	00	01	11	10
0	0	0	1	0
1	0	1	1	1

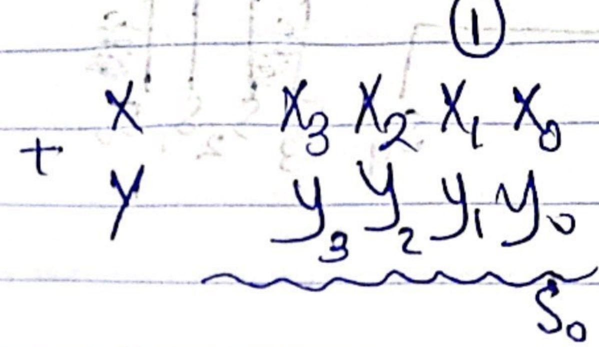
$$C_{out} = AB + A'BC + AB'C$$
$$= AB + C(A'B + AB')$$
$$= AB + C(A \oplus B)$$



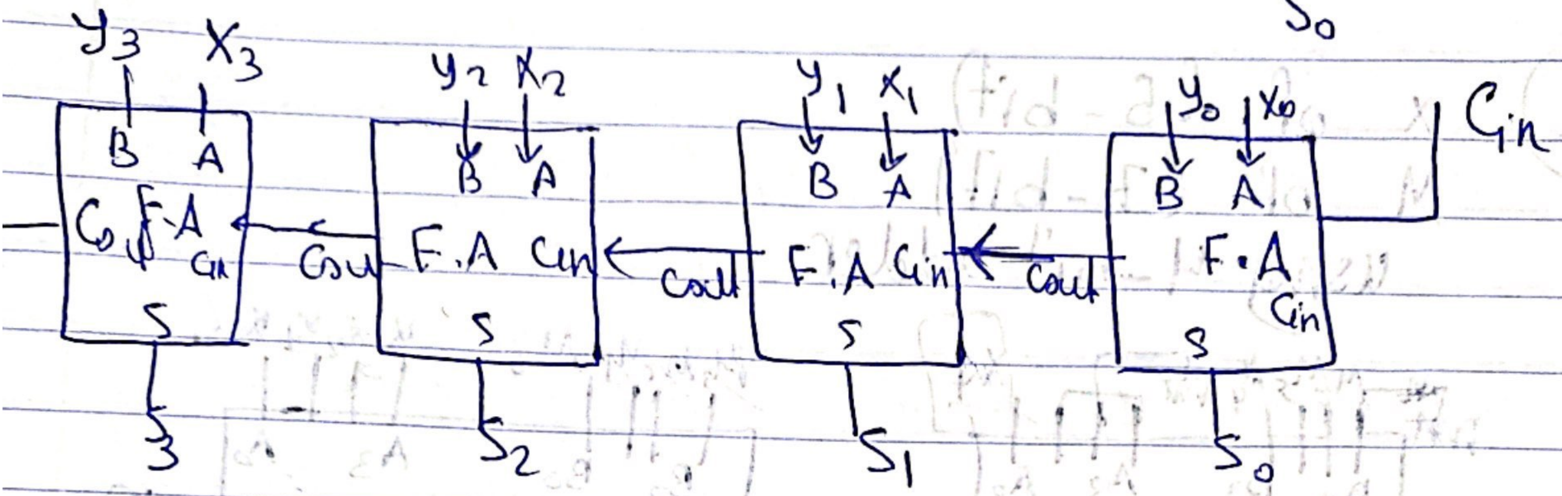
FA = 2H.A + OR


A
B
H.A
H.A
Sum
Cout

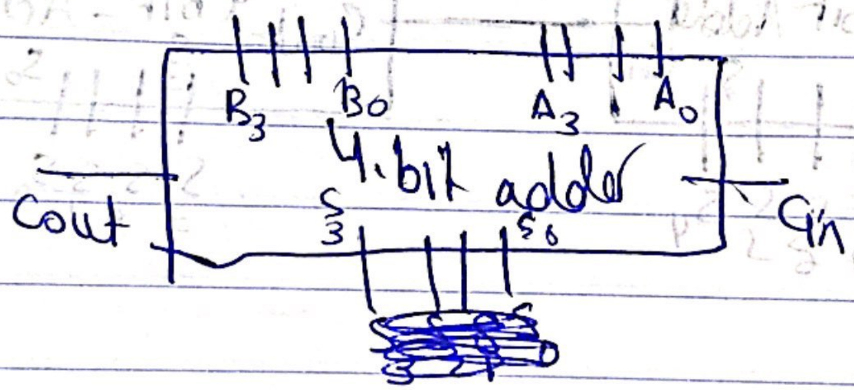
4 - Bit Adder



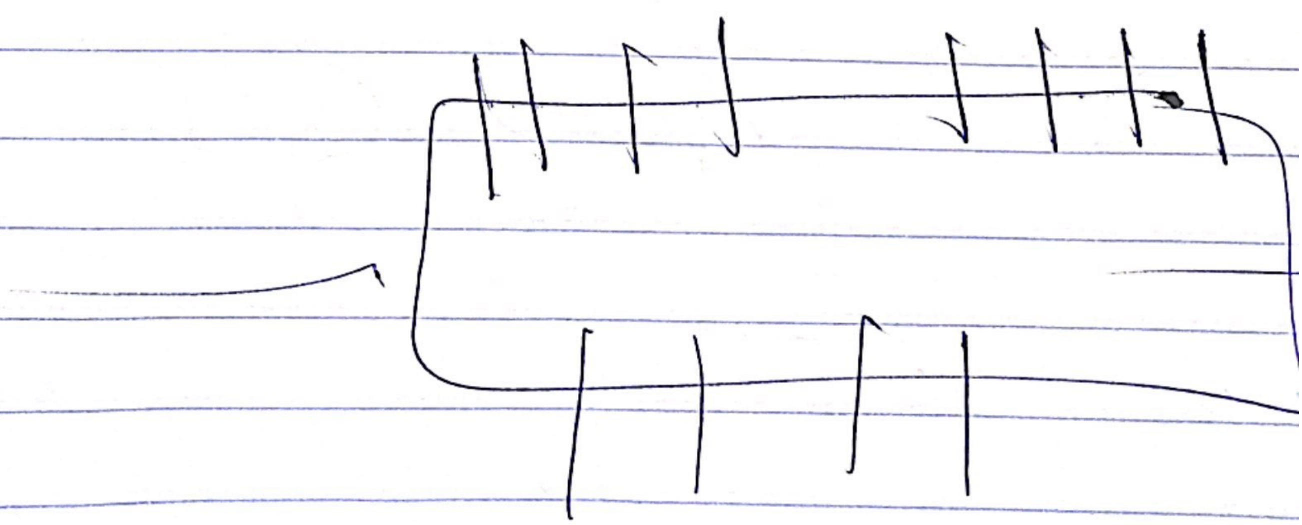
①
X
+ Y
$X_3 X_2 X_1 X_0$
$Y_3 Y_2 Y_1 Y_0$
S_0



Y_3 X_3
B A
F.A
Cin
S
S_3
Y_2 X_2
B A
F.A Cin
Cout
S
S_2
Y_1 X_1
B A
F.A Cin
Cout
S
S_1
Y_0 X_0
B A
F.A
Cin
Cout
S
S_0
Cin

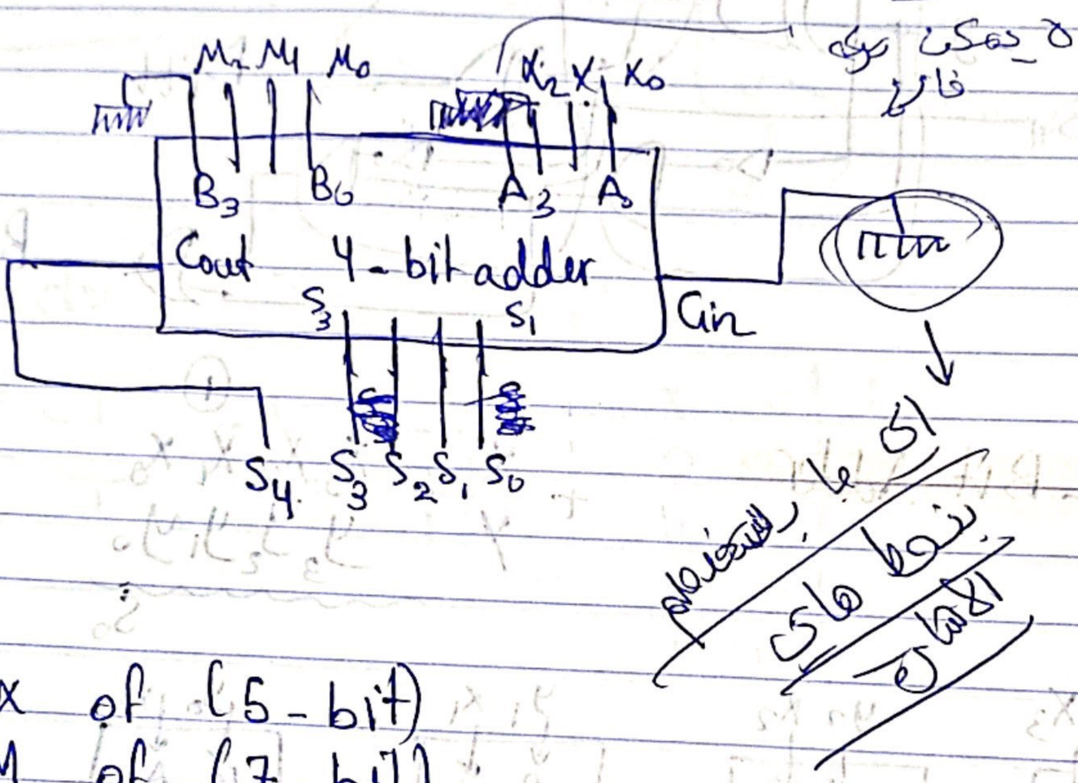


B_3 B_0 A_3 A_0
4-bit adder
Cout
Cin
S_3 S_0

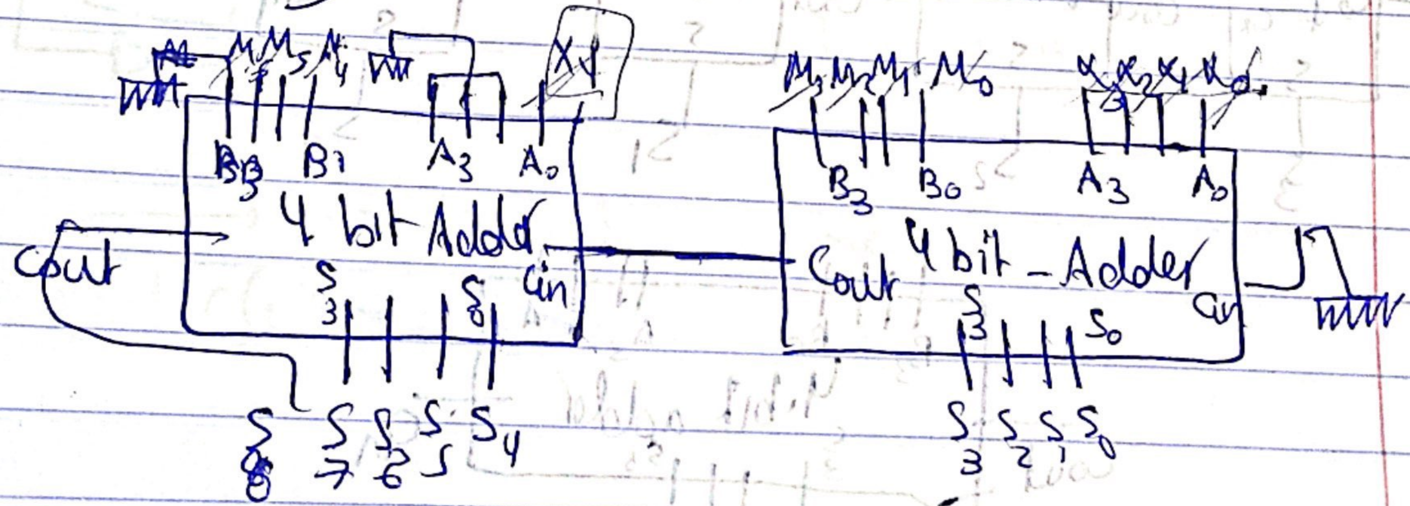


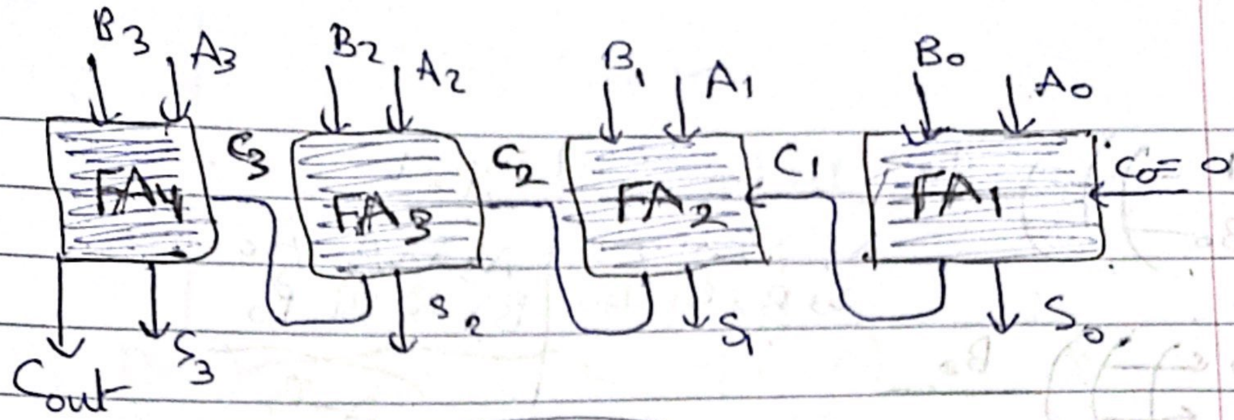
Design a logical circuit to add X of (2-bit) to M of (8 bit) using 4-bit adder in your design

2020



Ex: X of (5-bit)
M of (7-bit)
using 4-bit adder





Binary Subtractor

A + 2's comp B

$A - B = A + 2\text{'s comp } B$

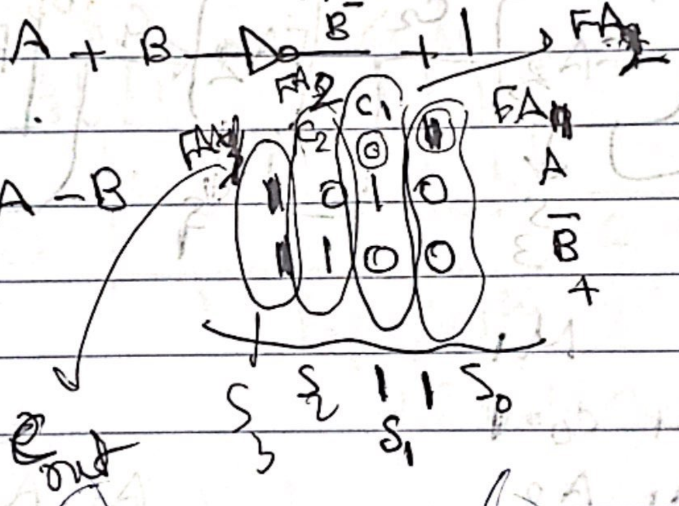
$A + 1^{st}\text{ comp} + 1\,(B)$

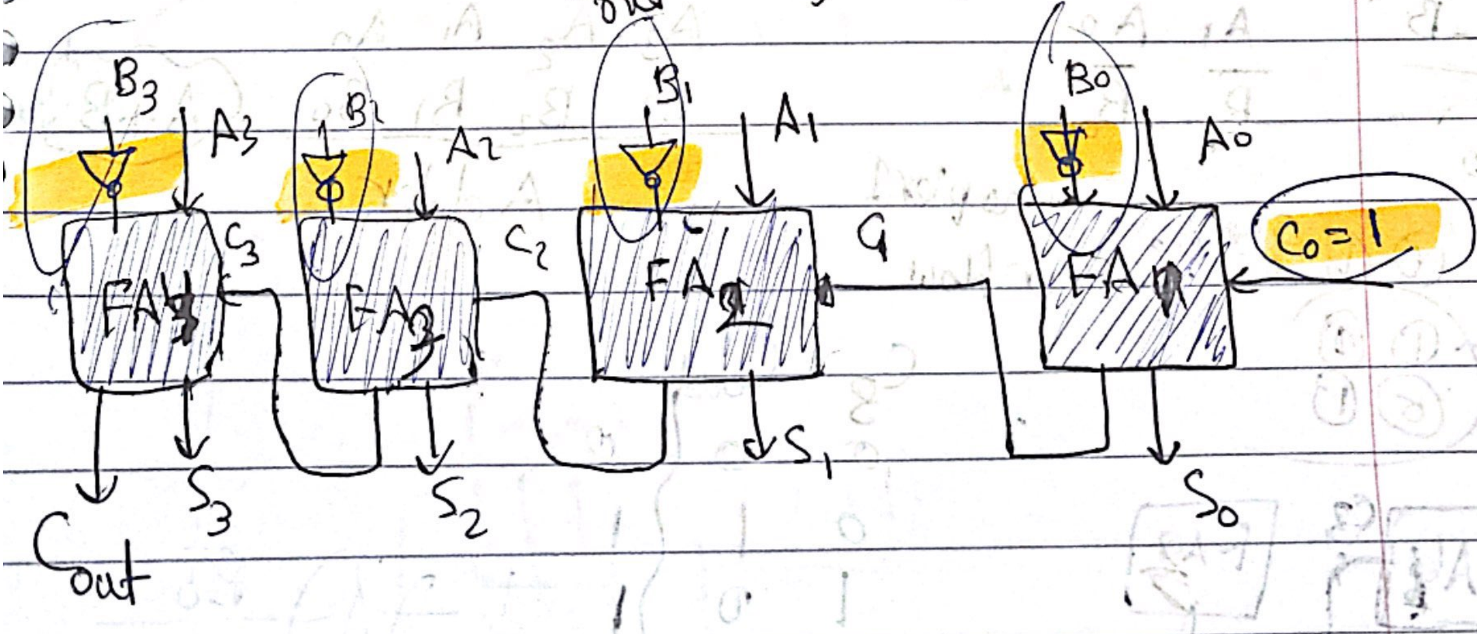
$A + B \to \bar{B} + 1 \to FA_1$

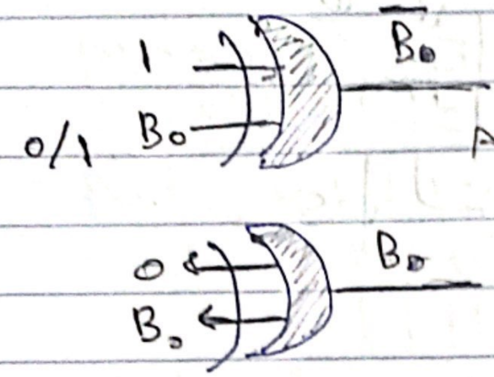
A = 1010

B = 0011

A − B





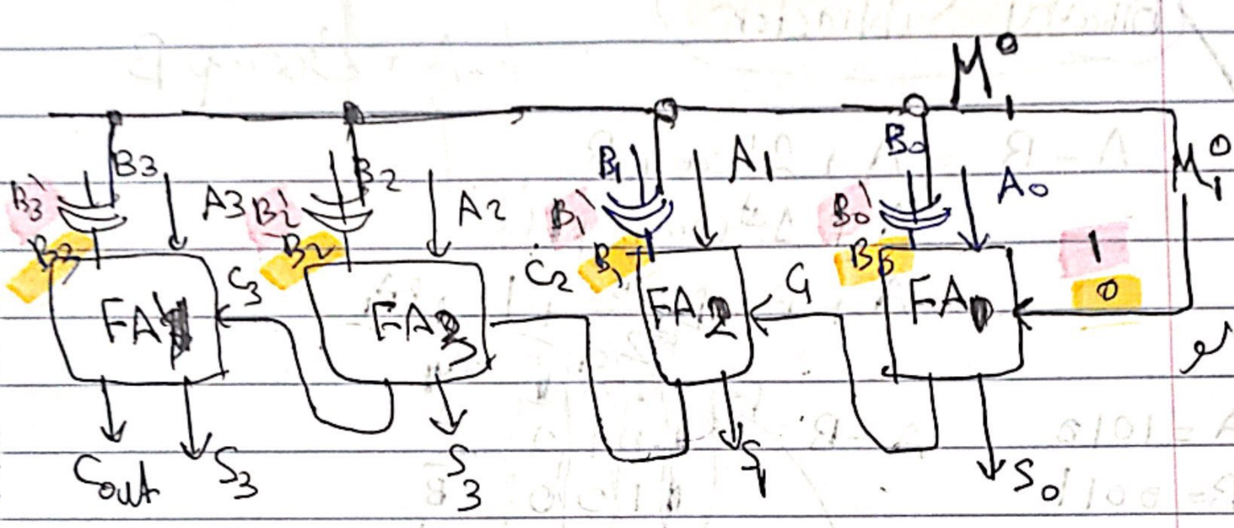


A3 A2 A1 A0
B3 B2 B1 B0

M = 1

A3 A2 A1 A0
B̄3 B̄2 B̄1 B̄0

(A − B)



M = 1

$C_1 \quad C_0 = 1$

$A_1 \quad A_0$

$\bar{B}_1 \quad \bar{B}_0$

(A − B)

(sub)

.chapter1

M = 0

$C_2 \quad C_1 \quad C_0 = 0$

$A_3 \quad A_2 \quad A_1 \quad A_0$

$B_3 \quad B_2 \quad B_1 \quad B_0$

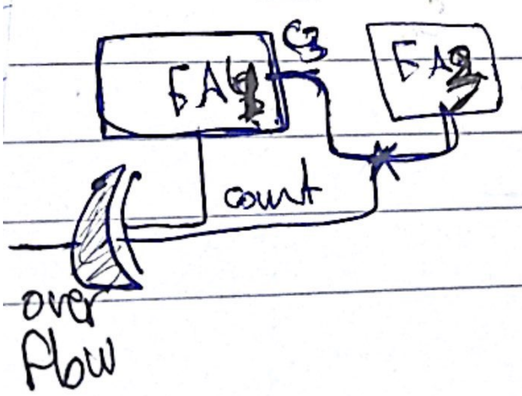
Adder

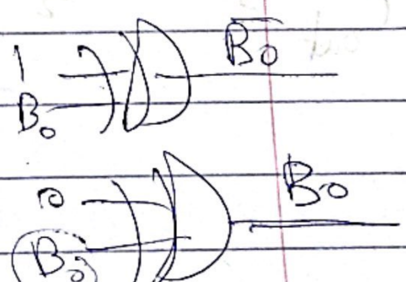
(A + B)

no over ① ① overflow ← over flow

over flow ① ① / ⓪ ①

C_3	Cout	
0	0	0
0	1	1
1	0	1
1	1	0





M!
C = 0

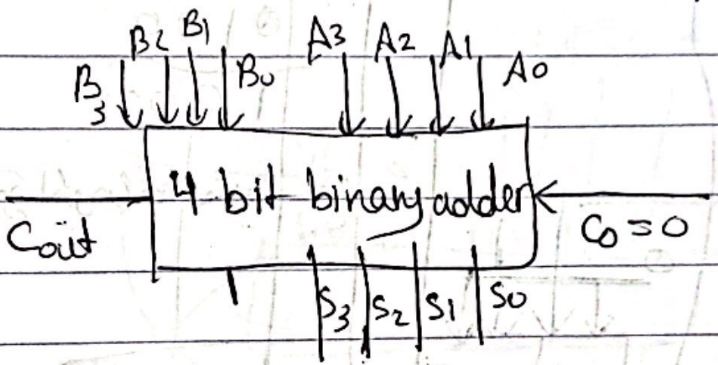
5 full adder

Design 5 bits binary adder ?] X

BCD Adder (Decimal Adder)

```
      1 0 0 0
 5      0 1 0 1
+4      0 1 0 0
      ---------
      0 1 0 0 1   valid
      [yes]
```

```
 5    0 1 0 1
+5    0 1 0 1
    ---------
    0 1 0 1 0
      0 1 1 0
```



addition Binary 4 bits.

* يوجد 4 fulladder

1 0 0 1

① اذا صار واحد

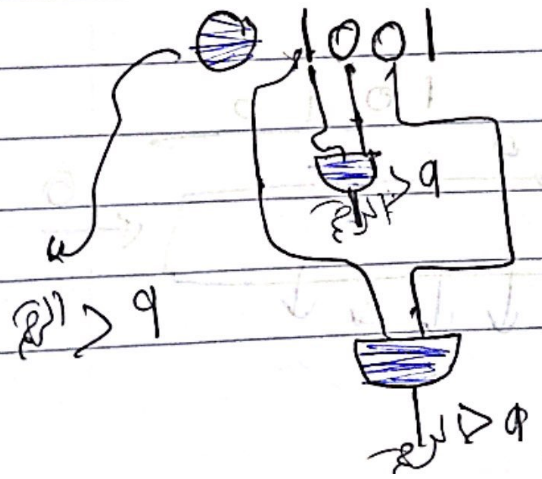
② أو اذا صار هاد واحد.

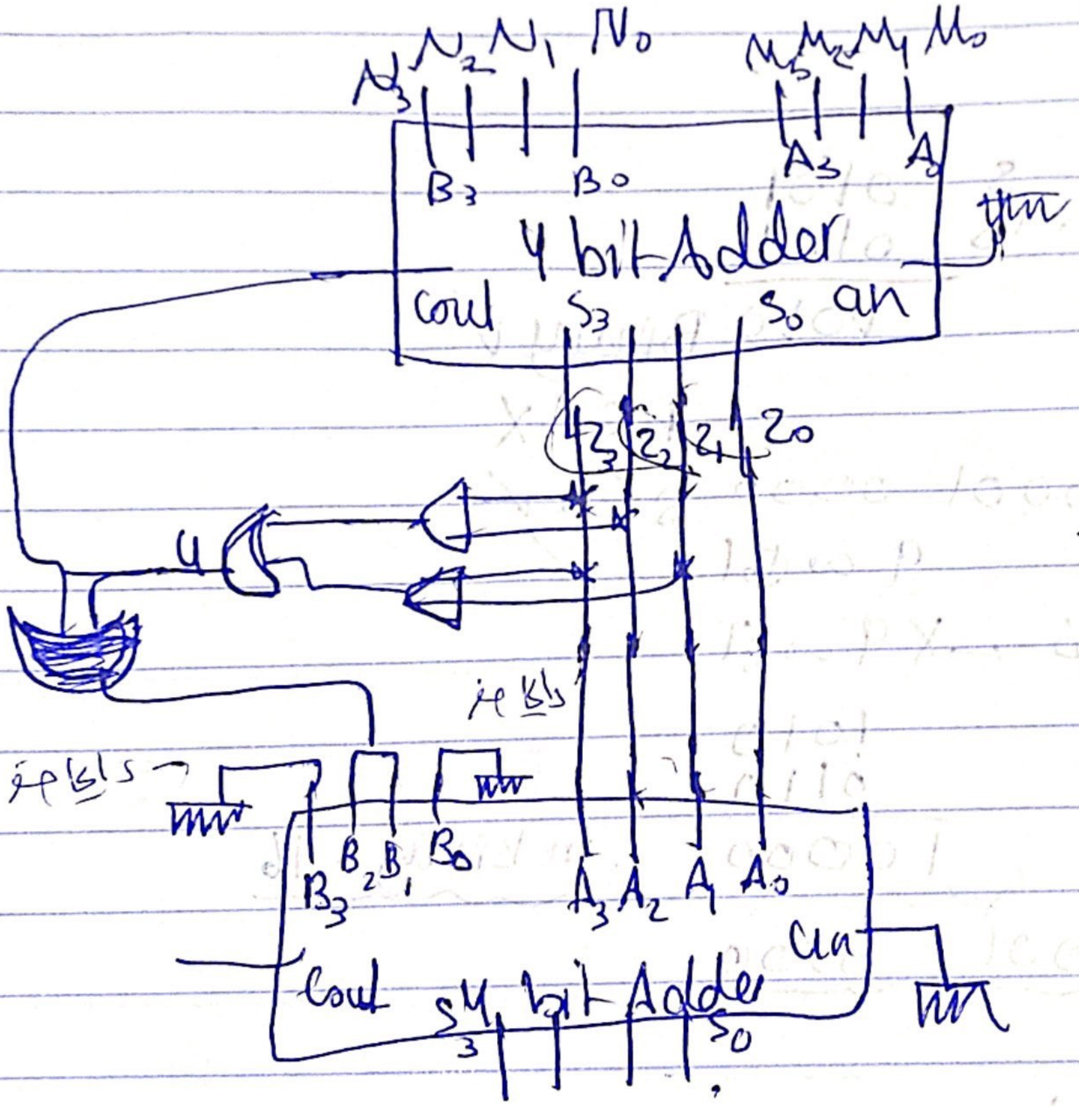
```
 5    0101
+5    0101
    ------
     01010
```

```
 9    1001
+9    1001
    ------
    10010
```

carry = 1

٣ حالات يكون اكبر من ٩





0	0	0	0
0	1	1	0

9 1001
9 + 1001

(1) 0010
0110 +

0001 1000

Subtraction

من الكتاب

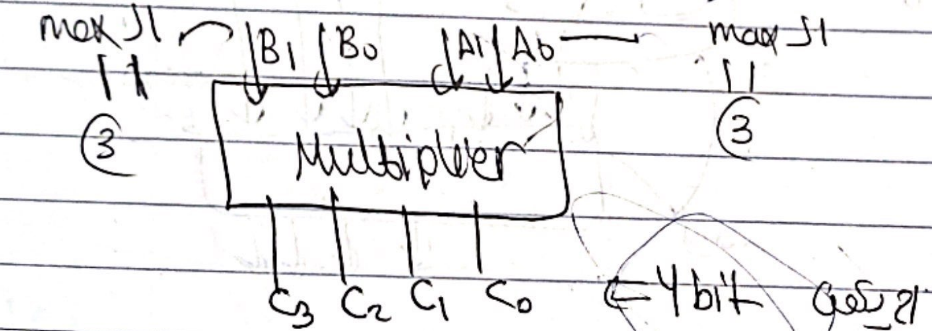
Binary Multiplier

42
16
———
252
420
———
672

1×0 = 0
0×1 = 0
0×0 = 0
1×1 = 1

AND

Design a 2bit × 2bit binary Multiplier

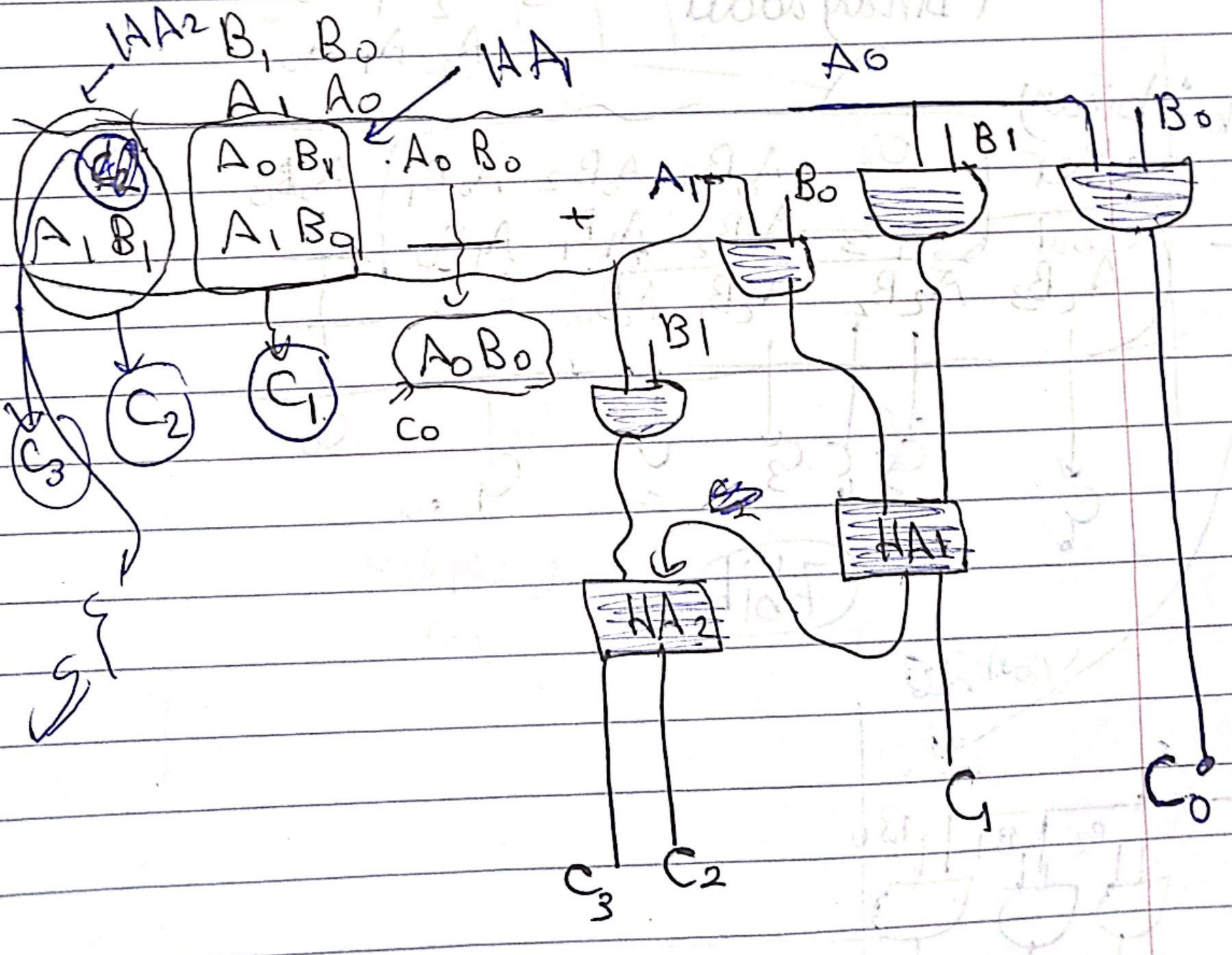


A1	A0	B1	B0	C3	C2	C1	C0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0				
⋮							
1	1	1	1	1	0	0	0

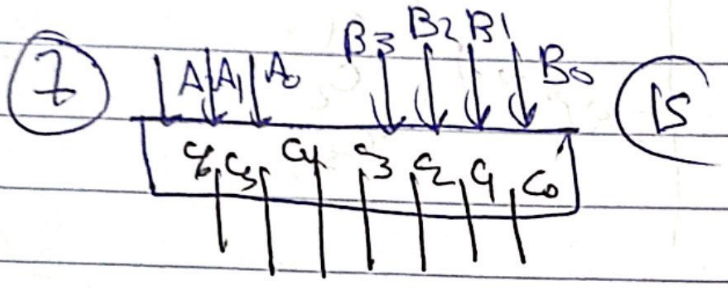
A_1A_0 \ B_1B_0	00	01	11	10
00				
01		1	1	
11			1	
10				

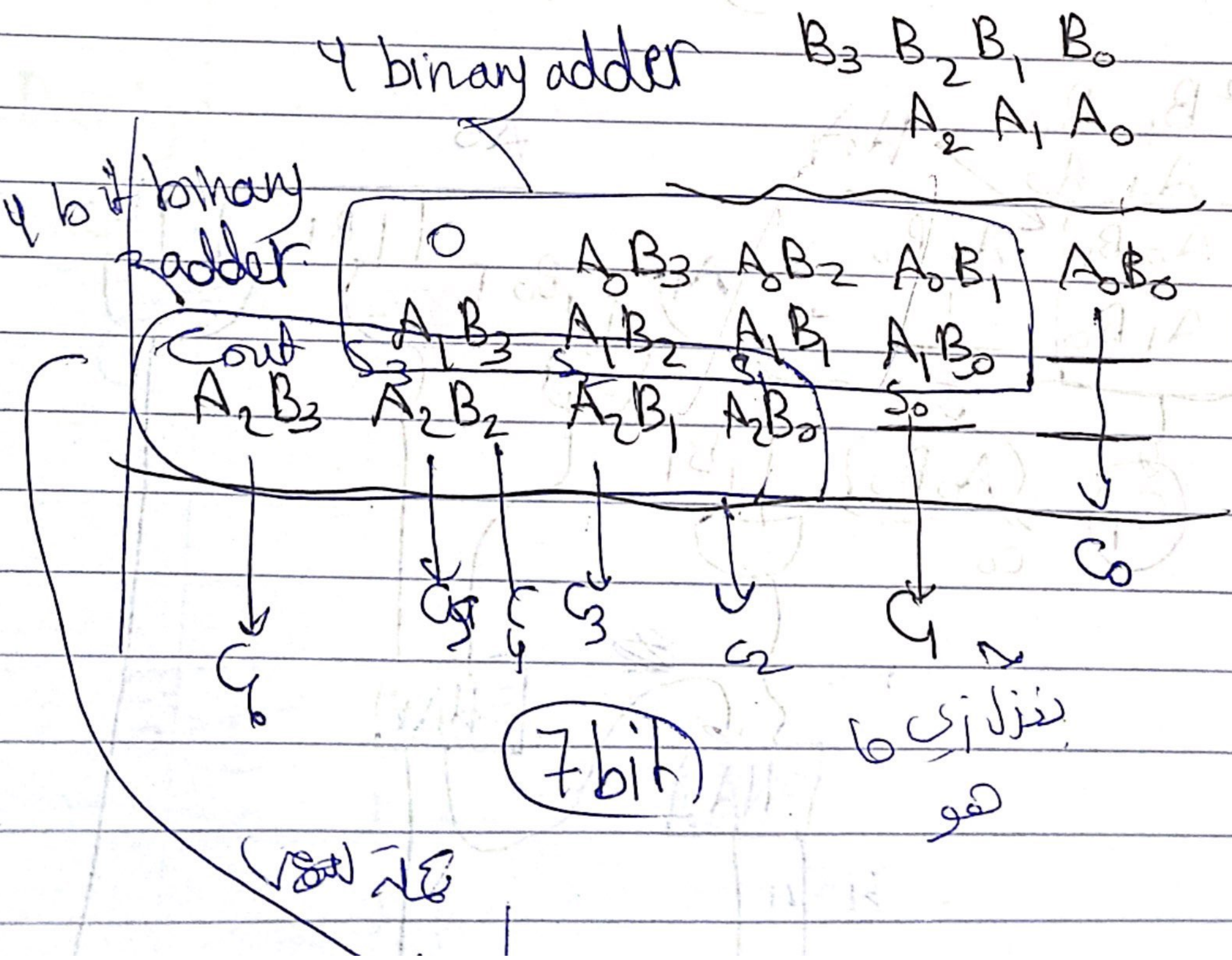
C_0 يمثل
اربع مربعات
C_1
C_2
C_3

2 bit x 2 bit binary multiplier ⇐ truth

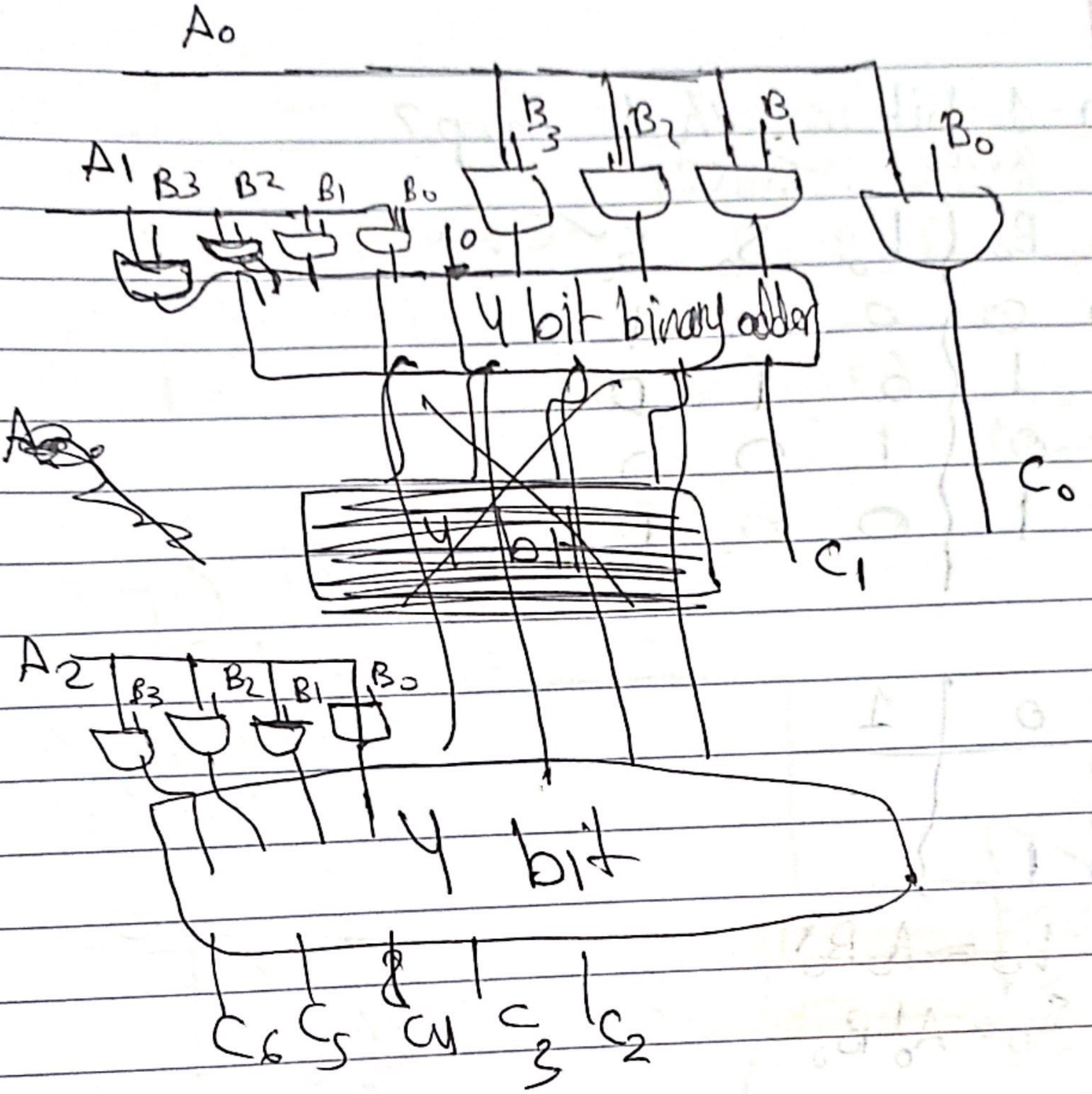


Design 4 bits x 3bits multiple





نفس الشي



magnitude computer

Design 1 bit magnitude comp?

A_0	B_0	L_0 ($A_0>B_0$)	S_0 ($A_0<B_0$)	E_0 (Equal)
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

A_0 \ B_0	0	1
0		
1	1	

$L_0 = A_0 B_0'$

K-map لكل ⇒ $S_0 = A_0' B_0$

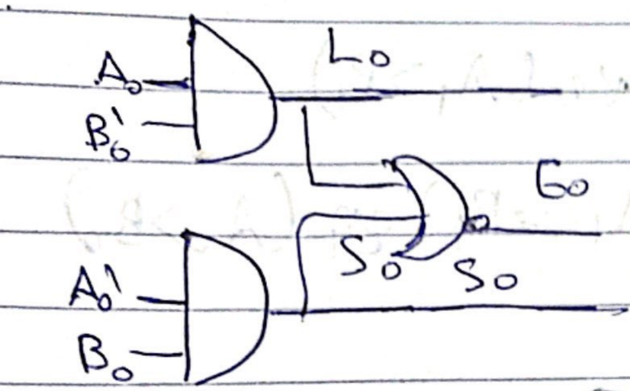
A_0 \ B_0	0	1
0	1	
1		1

$E_0 = A_0' B_0' + A_0 B_0$

$L_0 + S_0 = A_0 B_0' + A_0' B_0 = A_0 \oplus B_0$

$E_0 = A_0' B_0' + A_0 B_0 = (A_0 + B_0)'$

$E_0 = (L_0 + S_0)'$



$A = A_2 \; A_1 \; A_0$

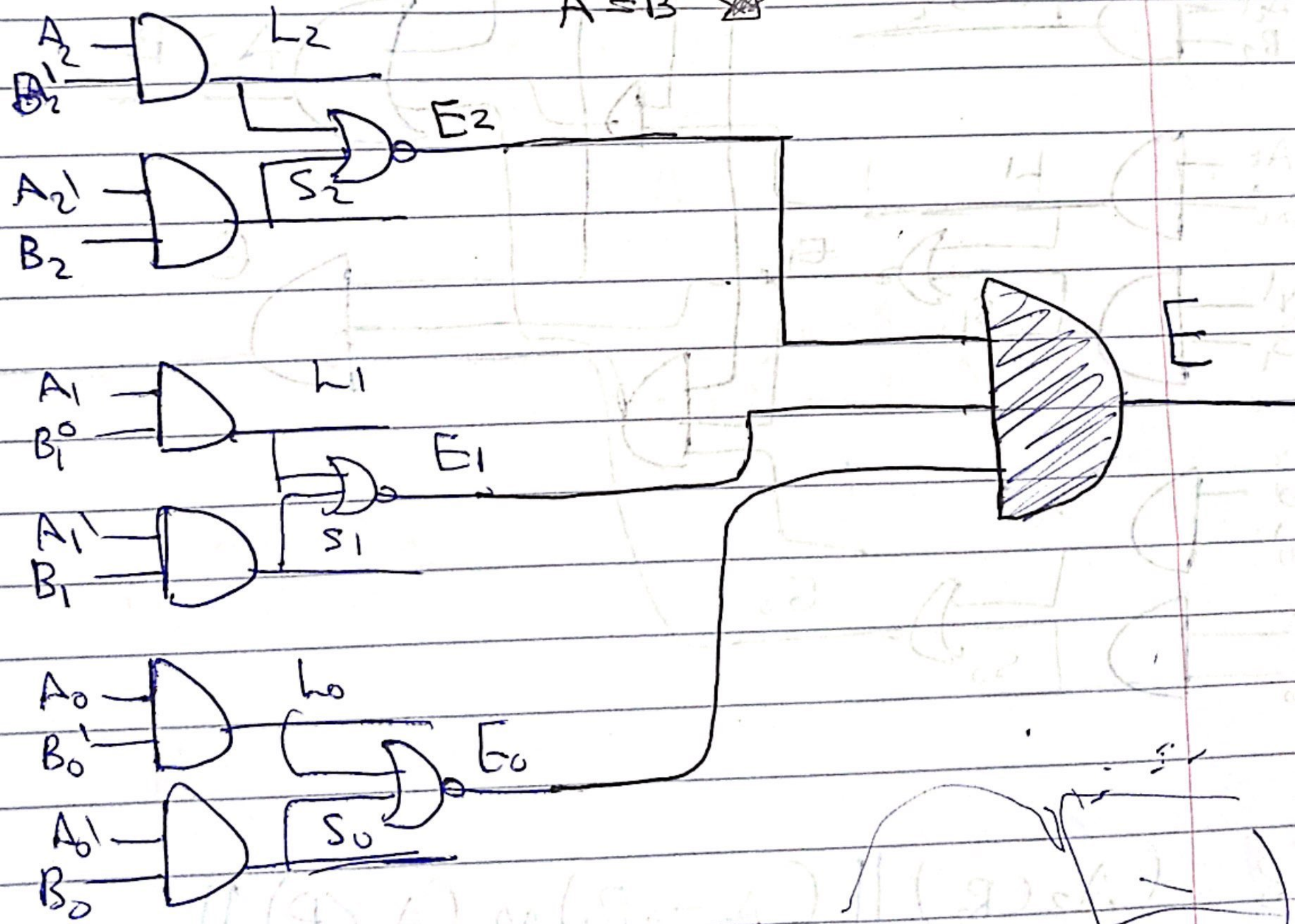
$B = B_2 \; B_1 \; B_0$

$A = B \Rightarrow (A_2 == B_2) \;\&\&\; (A_1 == B_1) \;\&\&\; (A_0 == B_0)$

$\Downarrow$

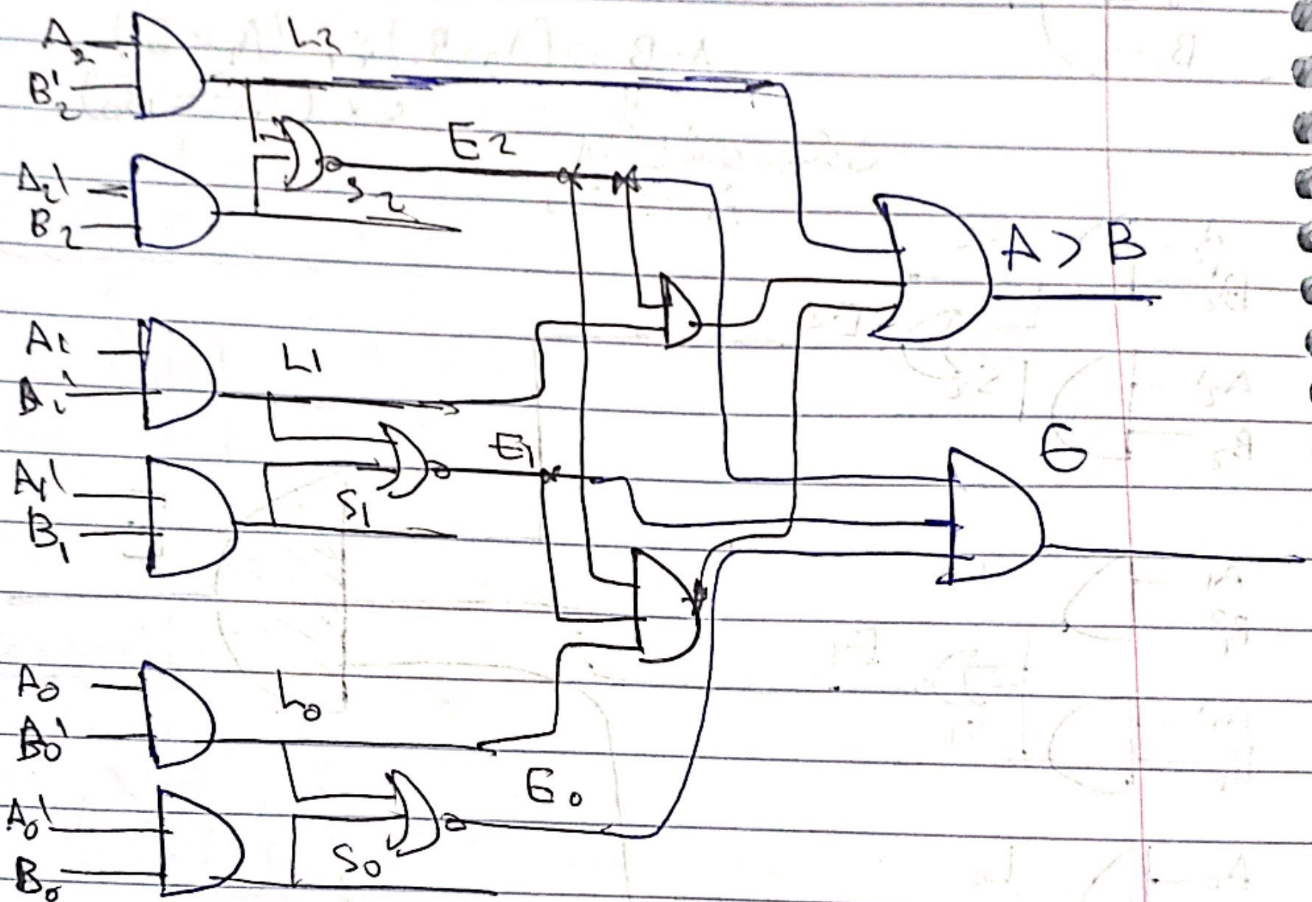
اذا جوا اللمبات بتكون

A=B



✓ 10000
01111

$$A > B = (A_2 > B_2) \,||\, (A_2 == B_2) \,\&\&\, (A_1 > B_1)$$
$$||\, (A_2 == B_2) \,\&\&\, (A_1 == B_1) \,\&\&\, (A_0 > B_0)$$



$$A < B = (A_2 < B_2) \,||\, (A_2 == B_2) \,\&\&\, (A_1 < B_1) \,||$$
$$(A_2 == B_2) \,\&\&\, (A_1 == B_1) \,\&\&\, (A_0 < B_0).$$

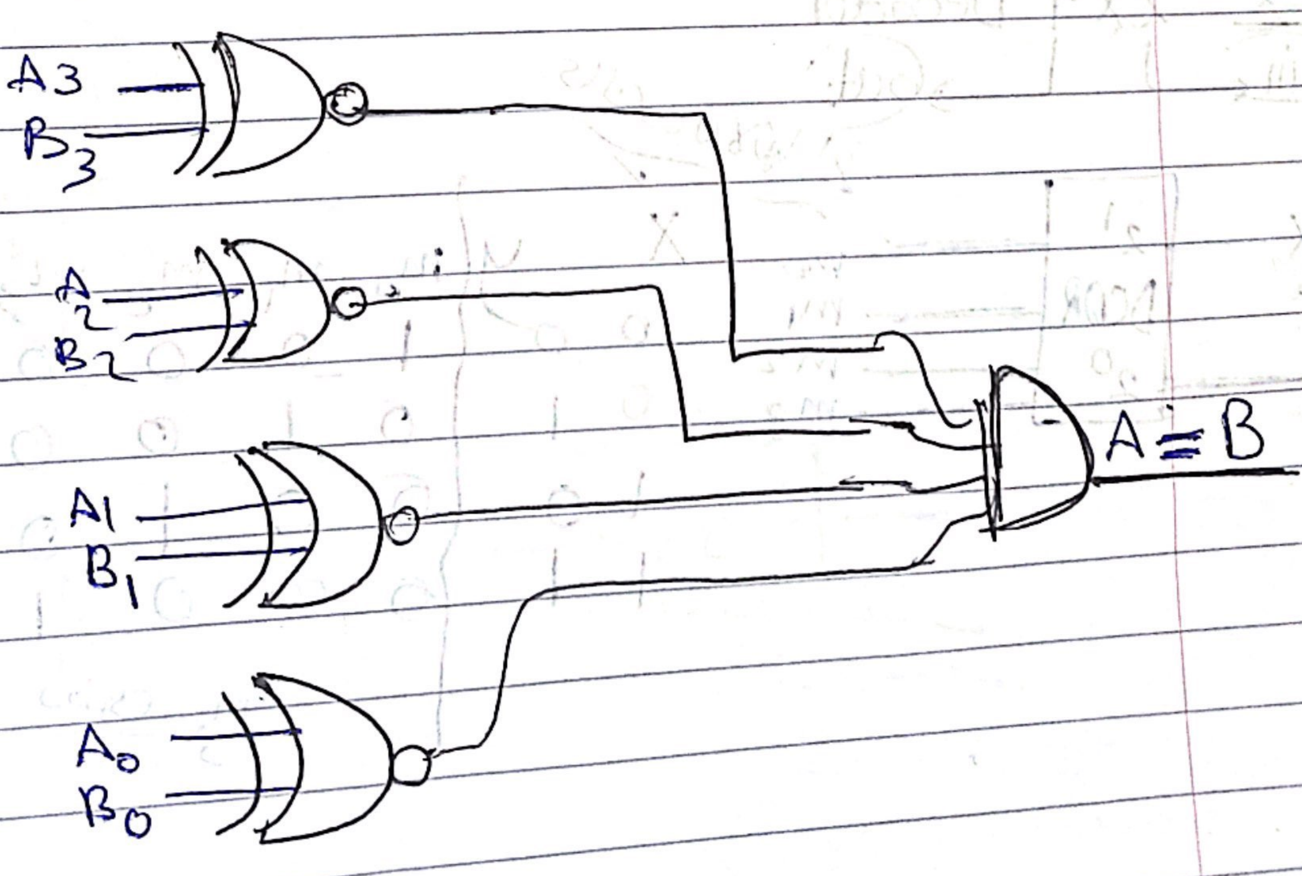
Design 4 bit x 4 bit equality computer?

A = $A_3 A_2 A_1 A_0$
B = $B_3 B_2 B_1 B_0$

A=B

A	B	$(A \oplus B)'$
0	0	1
0	1	0
1	0	0
1	1	1

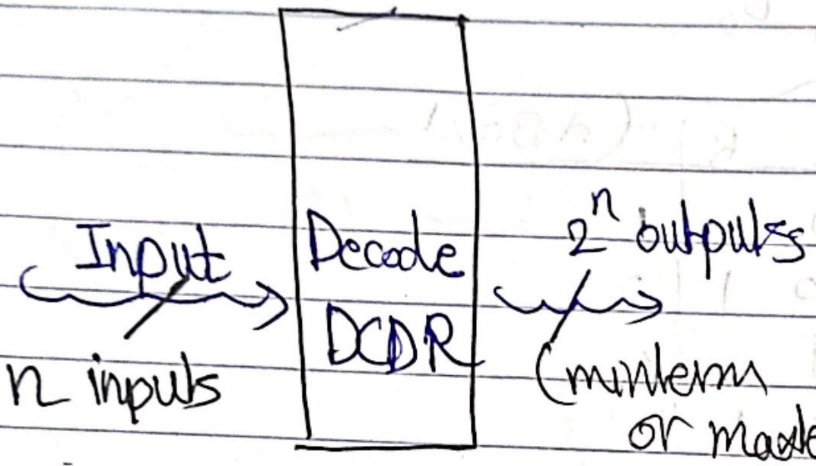
A=B $(A_3 == B_3)$ && $(A_2 == B_2)$ && $(A_1 == B_1)$ && $(A_0 == B_0)$



Decoder

i/p: 2 → 4, 3 → 8, 4 → 16 (2^n)

combinational circuit



2 → 4
3 → 8
4 → 16
5 → 32

only one output is on other is off.

Ex 2X4 Decoder
(in) (out)

X	y	m_0	m_1	m_2	m_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

x\y	0	1
0	1	
1		

$m_0 = x'y'$

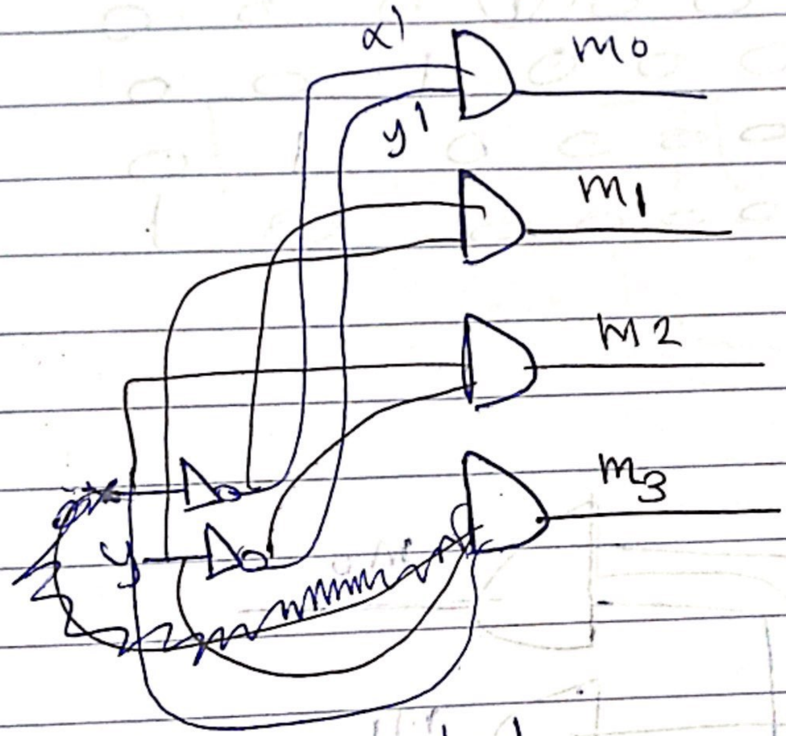
Decoder

D_0 — $m_0 = x'y'$

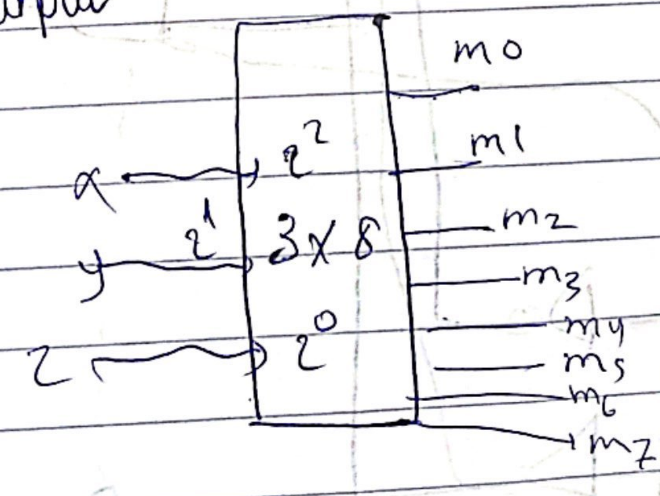
D_1 — $m_1 = x'y$

D_2 — $m_2 = xy'$

D_3 — $m_3 = xy$



input → Decoder 3 x 8 ← output



x	y	z	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

$m_0 = x'.y'.z'$

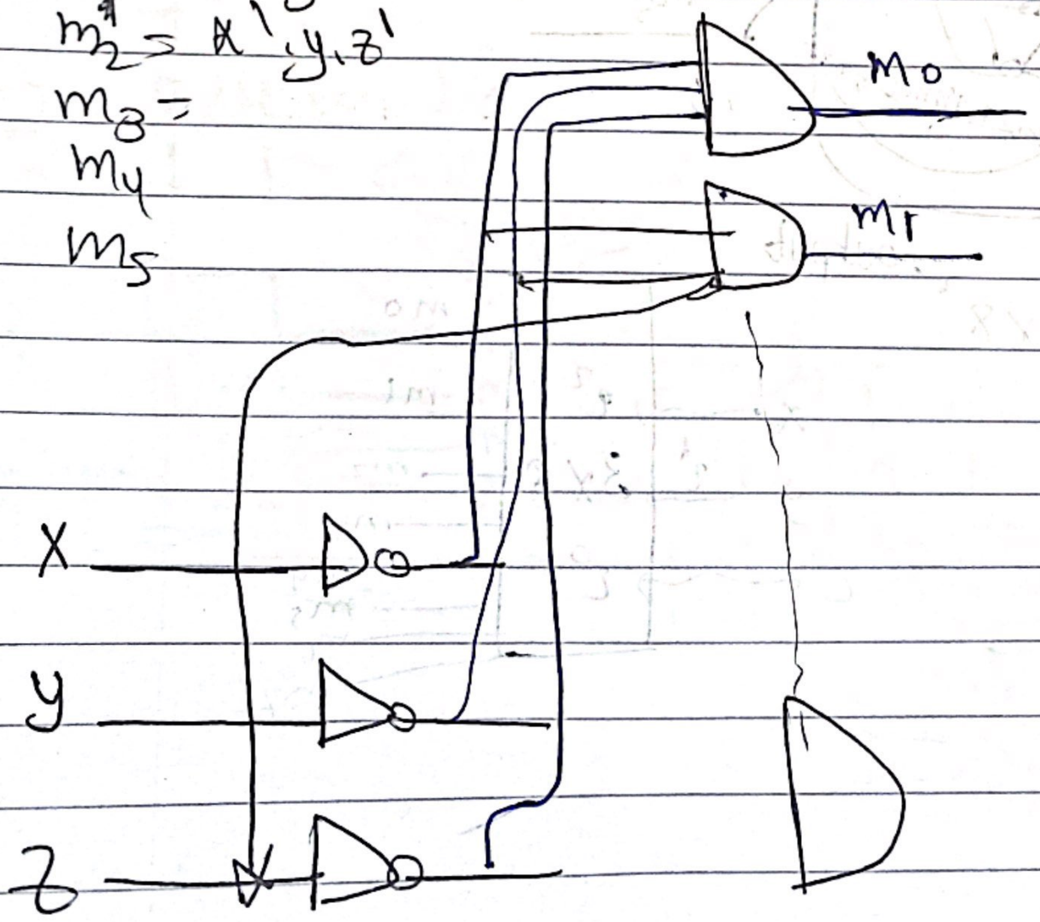
$m_1 = x'.y'.z$

$m_2 = x'.y.z'$

$m_3 =$

m_4

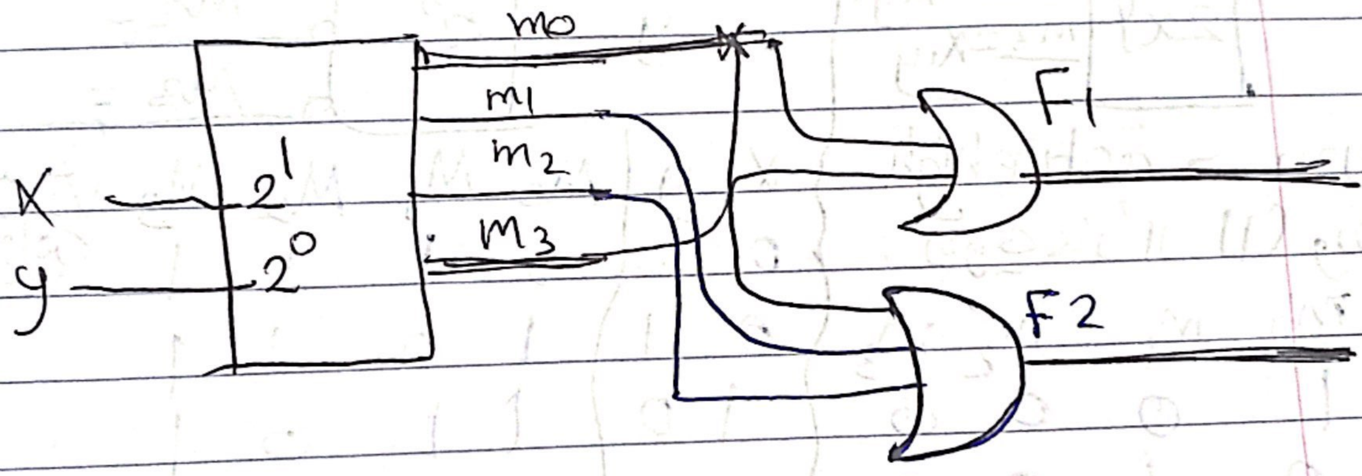
m_5



Implement the function, using decoder ?

$F_{1(x,y)} = \Sigma(0,3) = m_0 + m_3$

$F_{2(x,y)} = \Sigma(0,1,2) = m_0 + m_1 + m_2$



شرط يكونوا نفس عدد ال Inputs.

هوناها لانها اوفر

Active high decoder } Active low decoder

2 x 4 active high decoder

x — 2^1
y — 2^0
2x4

$m_0 = x' \cdot y'$
$m_1 = x' \cdot y$
$m_2 = x \cdot y'$
$m_3 = x \cdot y$

minterm = active high

بتوي عند ال (1) وبطلع شعع

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

⟸ الرحة زي قبل

2x4 active low decoder

x — 2^1
y — 2^0

$M_0 = (x+y) = (x' \cdot y')'$
$M_1 =$
$M_2 =$
$M_3 =$

x	y	M_0	M_1	M_2	M_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

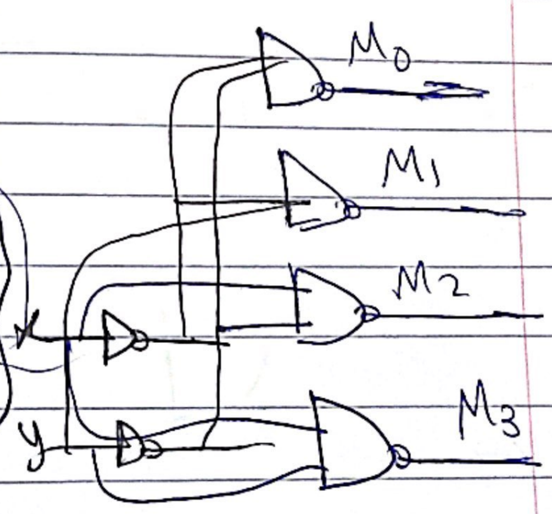
بتوي شد عع

بتوي عند واحد

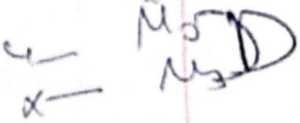
$M_1 = x + y' = (x' \cdot y)'$
$M_2 = x' + y = (x \cdot y')'$
$M_3 = x' + y' = (x \cdot y)'$

عكس الرحة السابقة

الرياضة (0)

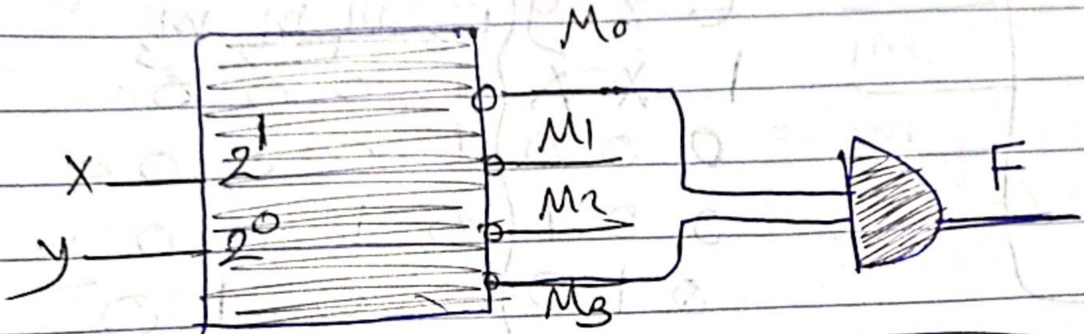


$M_0 = (m_0)'$



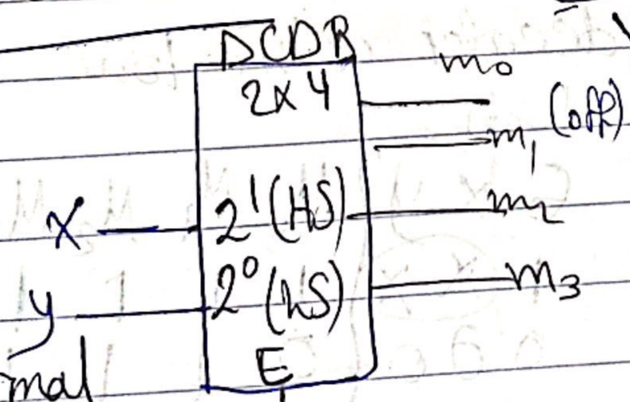
implement the following function using Decode $F(x, y) = \Pi(0,3)$

$= M_0 \cdot M_3$



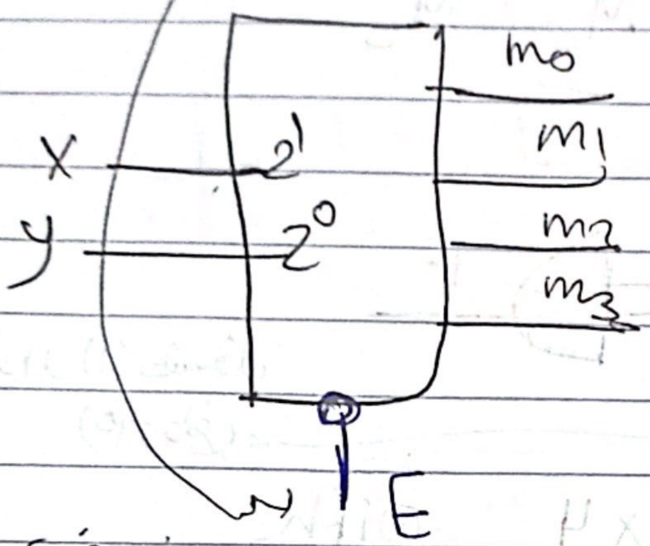
Ex: active high decoder 2 x 4 with high enable [E]

imp output control



E	x	y	m_0	m_1	m_2	m_3
0 (off)	X	X	0	0	0	0
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

2x4 active high decoder with active low enable.

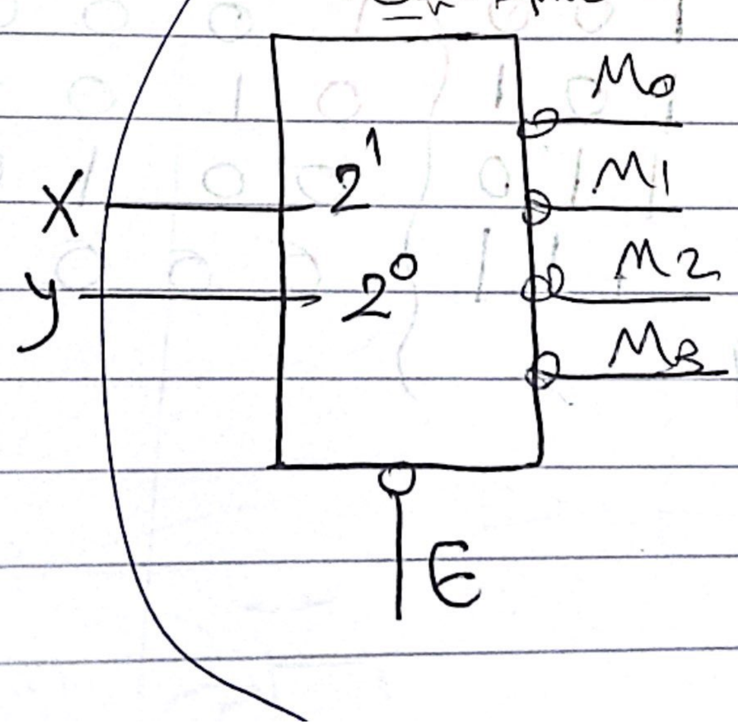


E	x	y	m_0	m_1	m_2	m_3
1	X	X	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	1	0	0	0	1

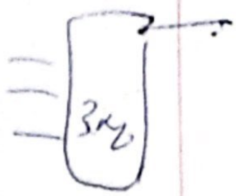
عند الصفر يضوي
عند الواحد يطفي.

2x4 active low decoder active low enable.

بعمل الشطرين (x x)
كلهم طافيين.



E	X	Y	M_0	M_1	M_2	M_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



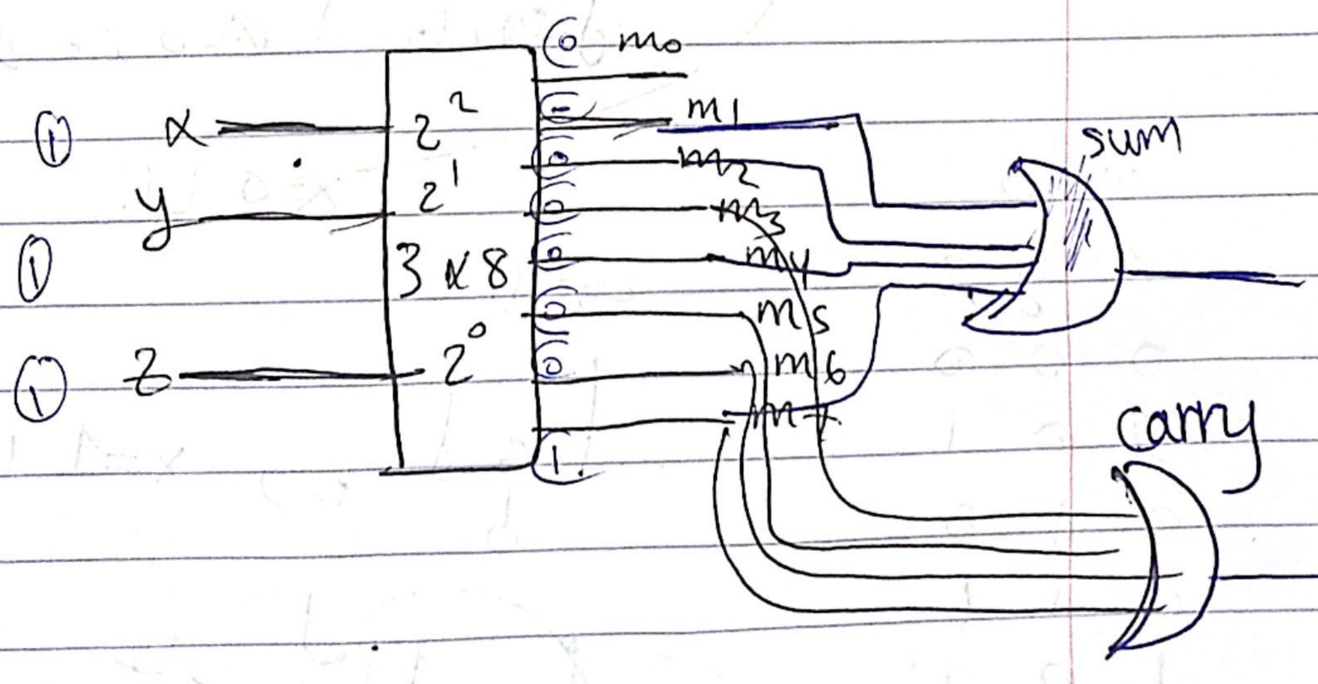
Implement Full adder using decoder
① 3 x 8 ② decoder 2x4.

x	y	z	sum	carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

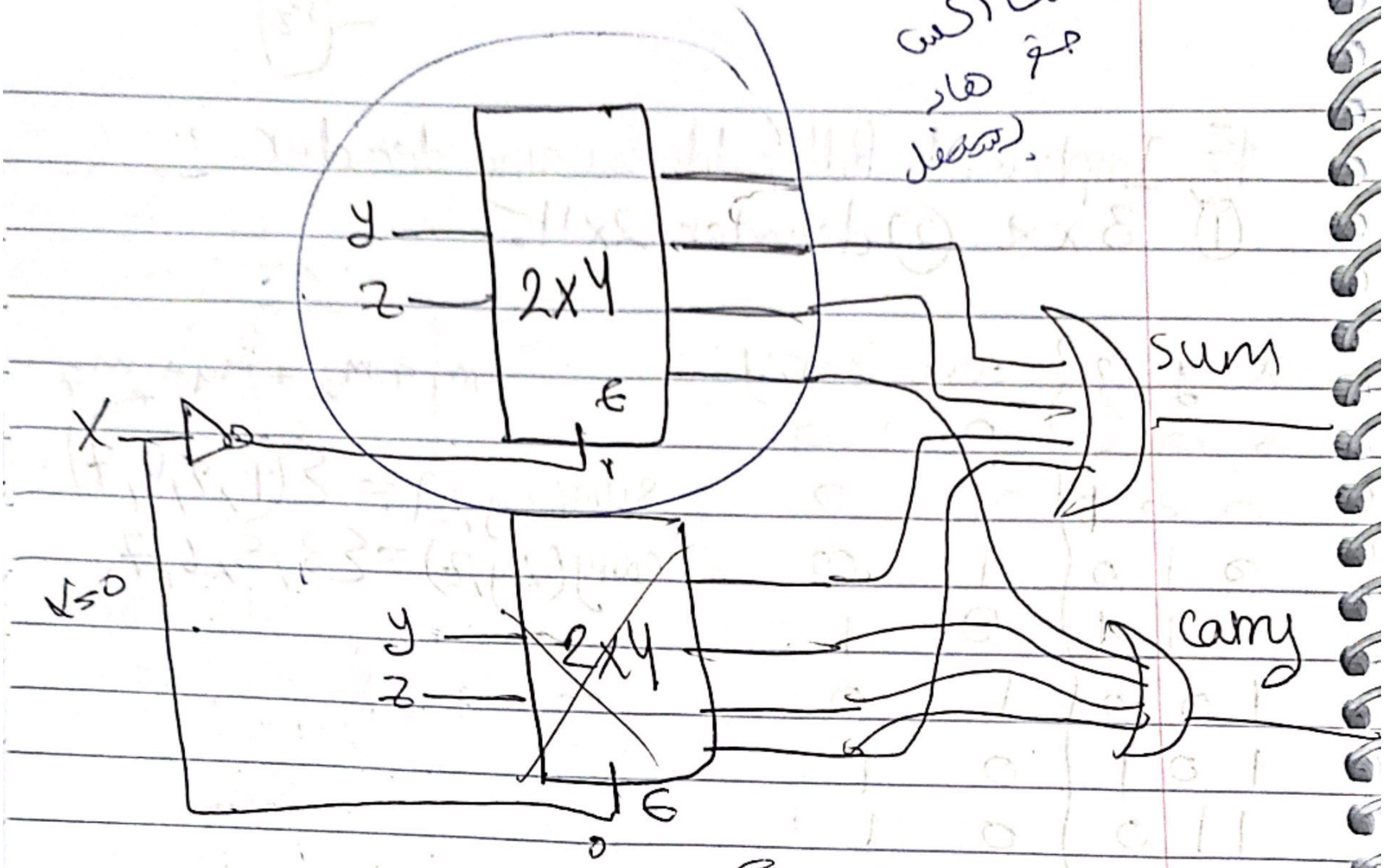
$m_1 + m_2 + m_4 + m_7$

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

$carry(x,y,z) = \sum 3,5,6,7$



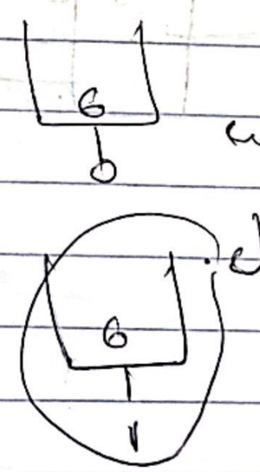
لما اكس
جو هاد
بيشتغل



لا عند E=0 ما بيشتغل

لما x=0
ال فوق

لما x=1 ...

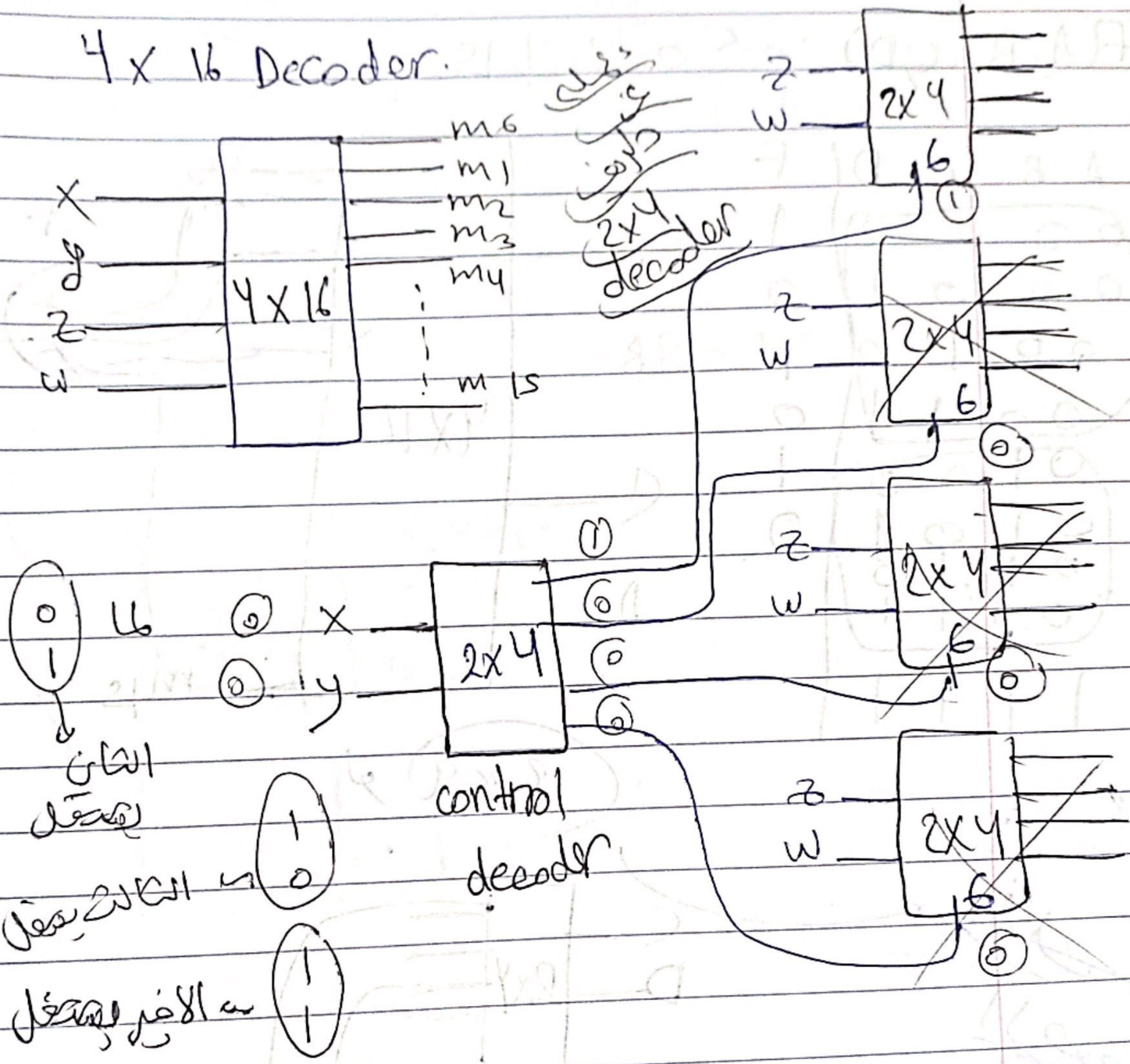


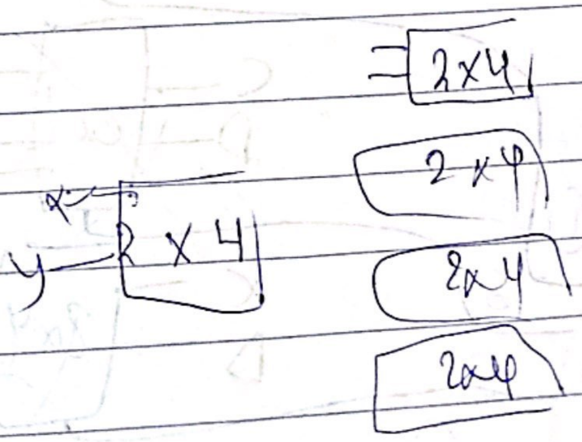
لا هاد
بيشتغل

x=0

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

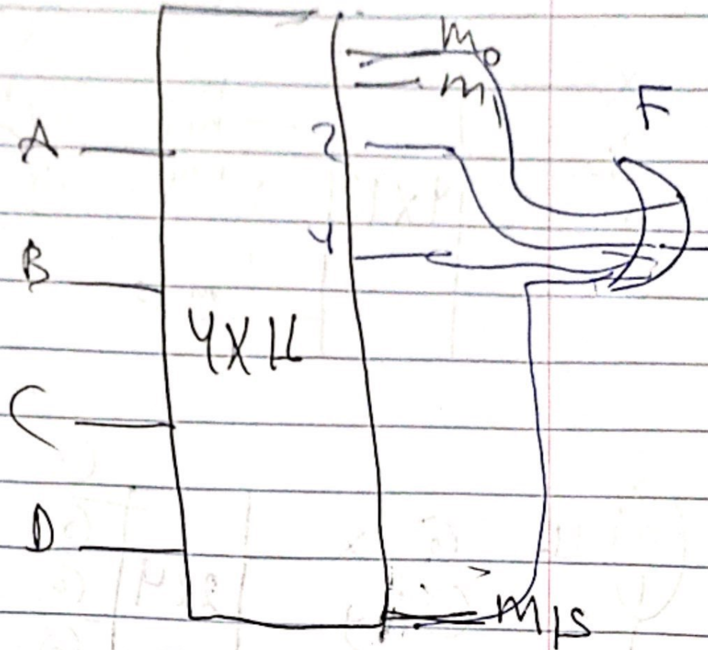
4 x 16 Decoder.

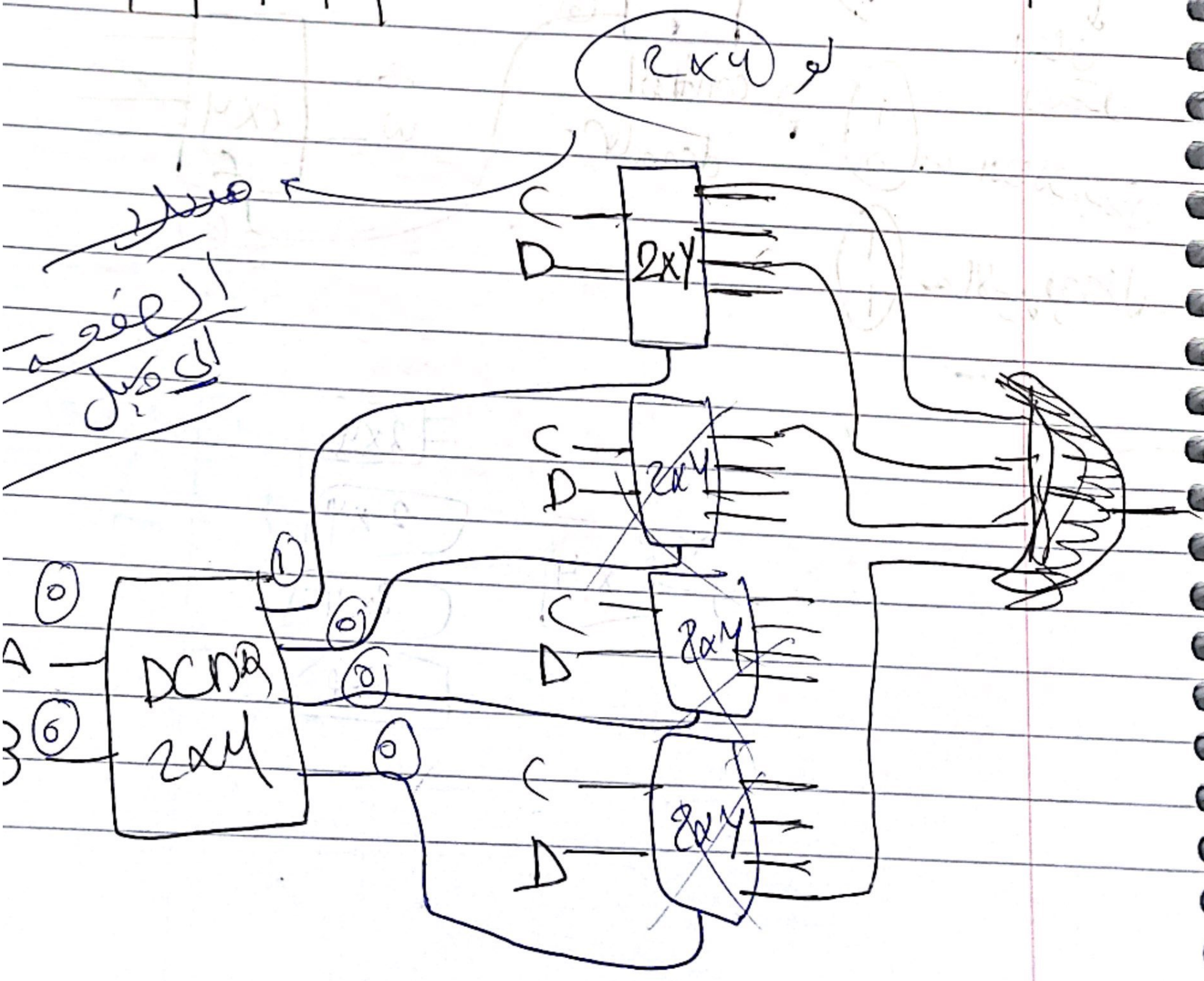


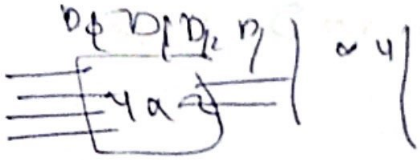


$F(A,B,C,D) = \Sigma(0, 2, 4, 15$

A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	
0	1	1	1	
1	1	1	1	





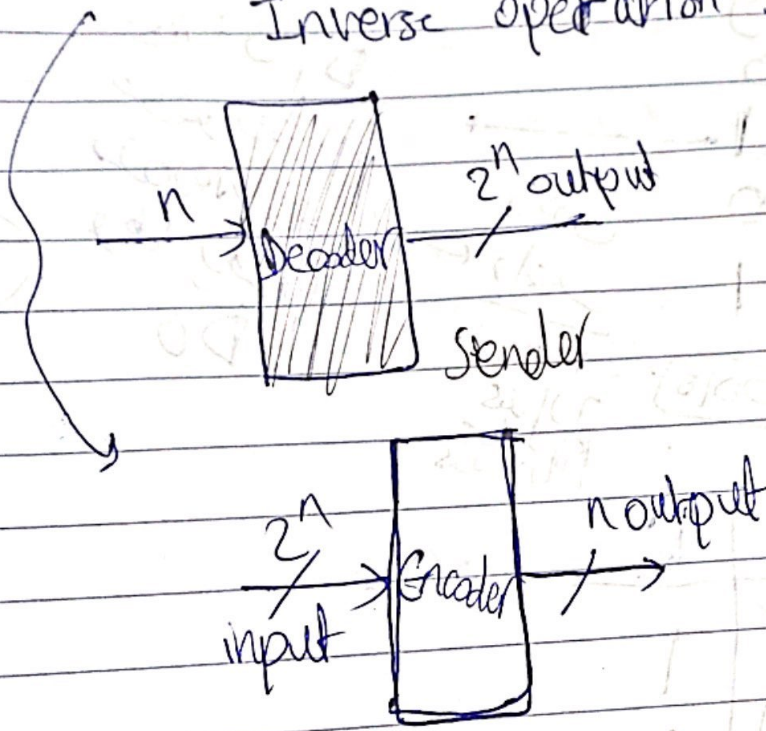


0 1 0 0 ← العادي بيستقبل

[(0) A
(1) B] لـ

Encoder

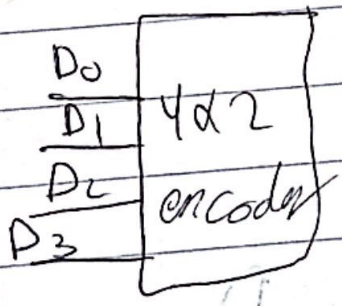
Inverse operation Decoder



($2^n \times n$)

(4) x 2 encoder

input output



D_0	D_1	D_2	D_3	x	y
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

$x = D_2 + D_3$

$y = D_1 + D_3$

اولوية

- Priority encoder. (Input priority on other input).

انا موجود

انا 4×2 priority encoder (D_3, D_2, D_1, D_0)

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

لما D_1 موجودة بتجاهل X وما يهتم D_0

X ← رح تكون واحد

0010 ← عند ال ٢، عند ال ٦، عند ال ١٠، عند ال ١٤

00, 01, 10, 11

D_0D_1 \ D_2D_3	00	01	11	10
00		1	1	1
01		1	1	1
11		1	1	1
10		1	1	1

X =

0 0 0 ٦

1 1 1 1

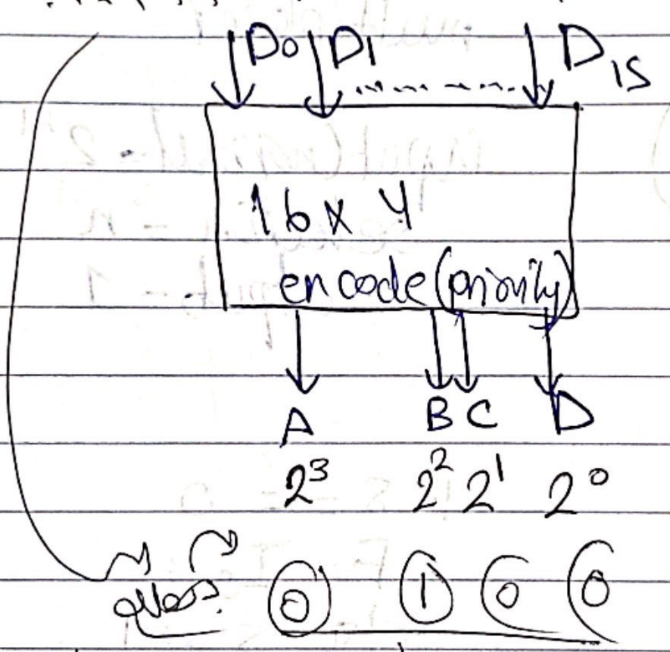
$X = D_2 + D_3$

D_0D_1 \ D_2D_3	00	01	11	10
00		1	1	
01	1	1	1	
11	1	1	1	
10		1	1	

0001
0011
0101
0111
1001
101
110
111

$$y = D_3 + D_1 D_2'$$

1) اذا كان 1 ... 1 1 1 1 1 1 1 1 1 ...



اعلى priority
D4
D3
D2
D0
.
.
D14

priority لانه اعلى
فلو بنشتغل اول

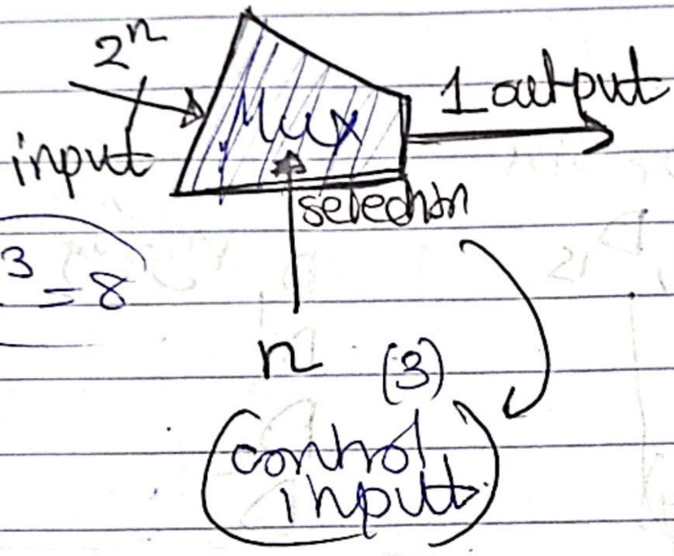
لو D_4 ⟸ (0) 1 بيعطي 0 0 1 1

1 1 1 1 1 1 1 1
D_0 — — D_{15}

A B C D ⟸ 1 ⟸ D_4
0 1 0 0
0 0 1 1 ⟸ لو كان ⟸ D_3 لو (مع D_4)

Multiplexer (Mux)
↓
متعدد
الاختيار من متعدد

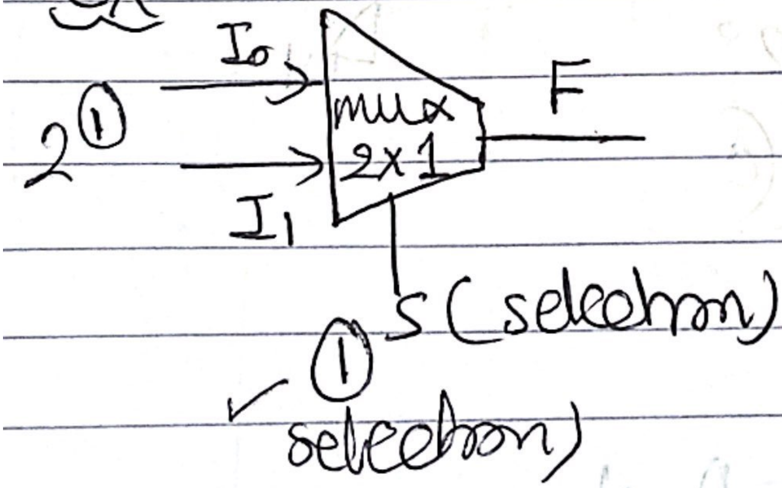
Mux



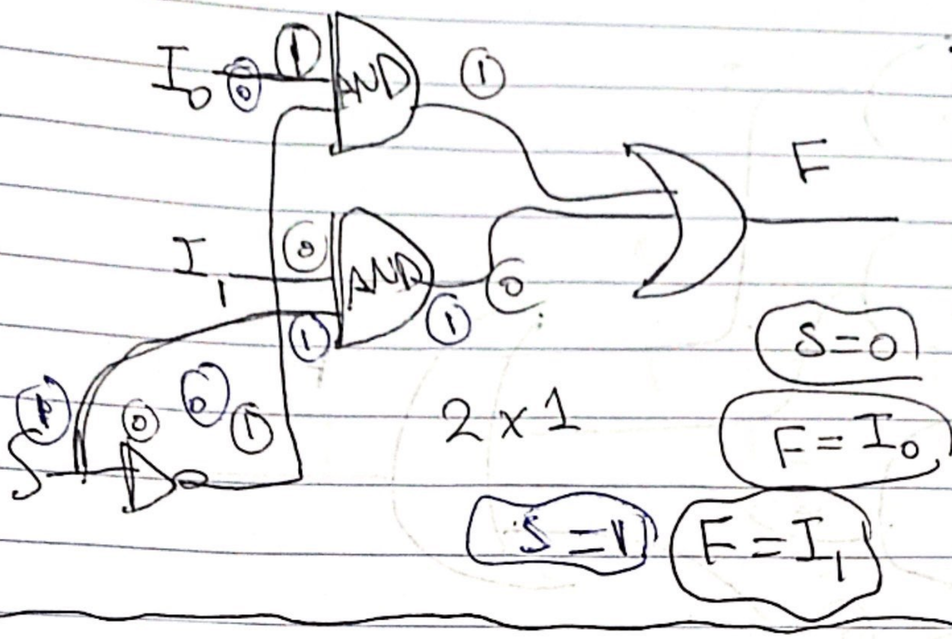
multiplexer

input (normal) = 2^n
selection = n
output = 1

Ex

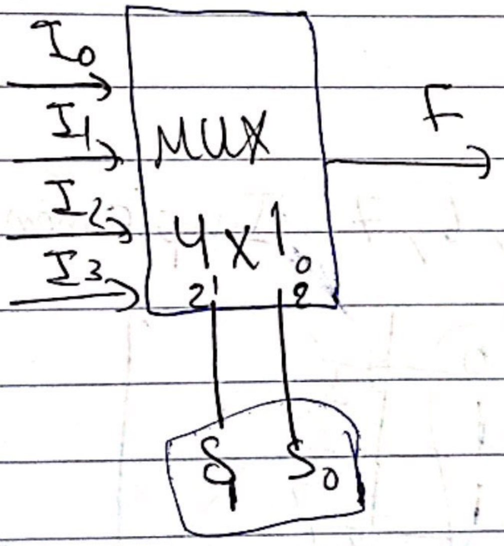


if S == 0
F = I_0 ;
else S == 1
F = I_1

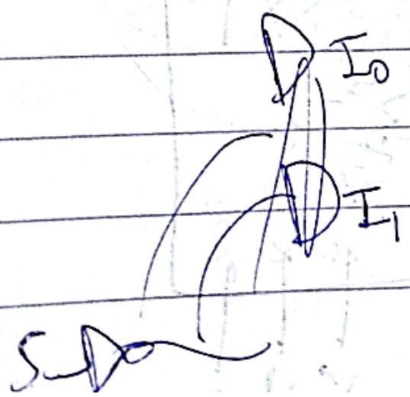


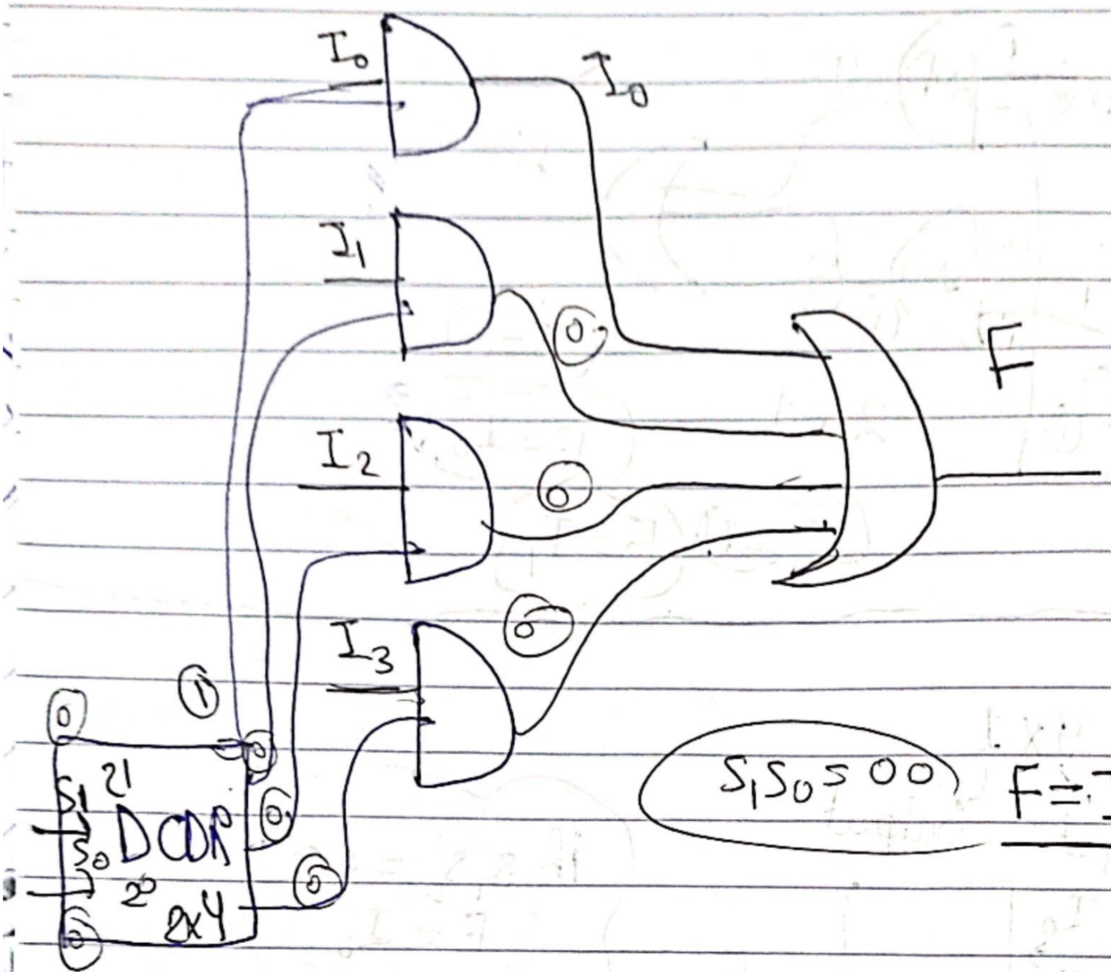
Mux 4x1

input output



if $S_1S_0 = 00$
$F = I_0$
$S_1S_0 = 01$
$F = I_1$
$S_1S_0 = 10$
$F = I_2$
$S_1S_0 = 11$
$F = I_3$



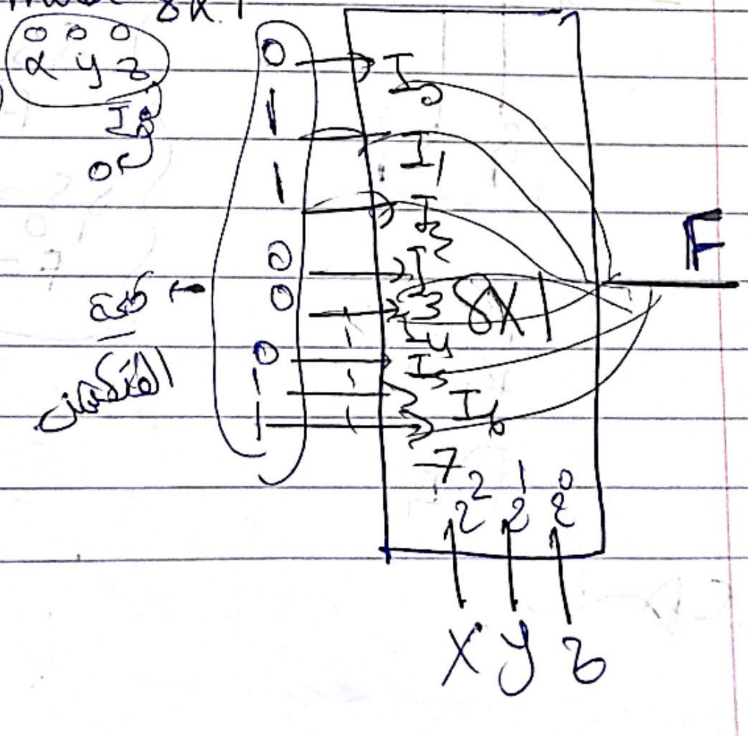


$S_1S_0 = 00$ $F = I_0$

(6) $F(x, y, z) = \Sigma\, 1, 2, 6, 7$ Implementation

mux 8x1

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



$F(A,B,C) = \Sigma\ 0, 3, 6, 7$

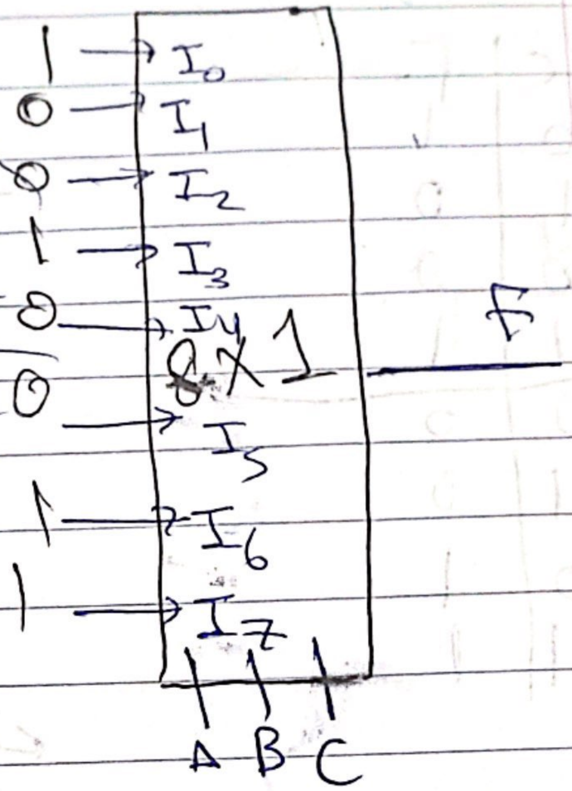
A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$F = \bar{C}$

$F = C$

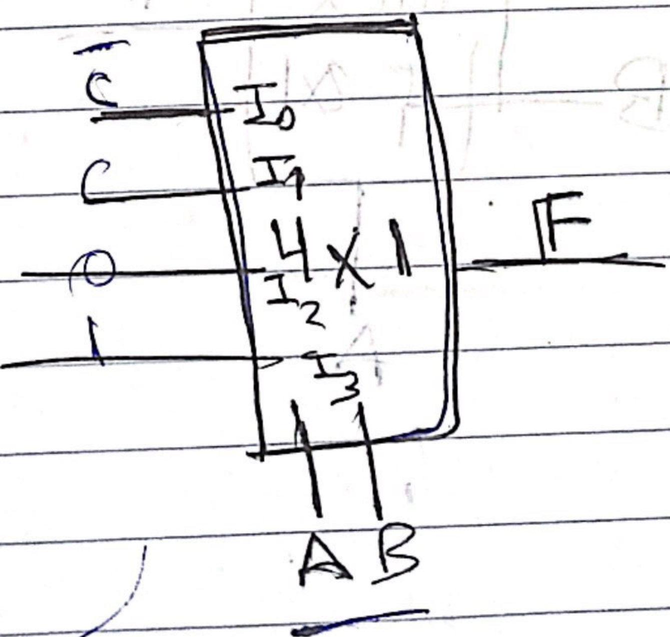
$F = 0$

$F = 1$



لـ ABC 000 → F → I_0 → 1

4X1



AB → يقسم الى اربع اقسام

12

$F_{(A,B,C)} = \Sigma\ 0,3,6,7$

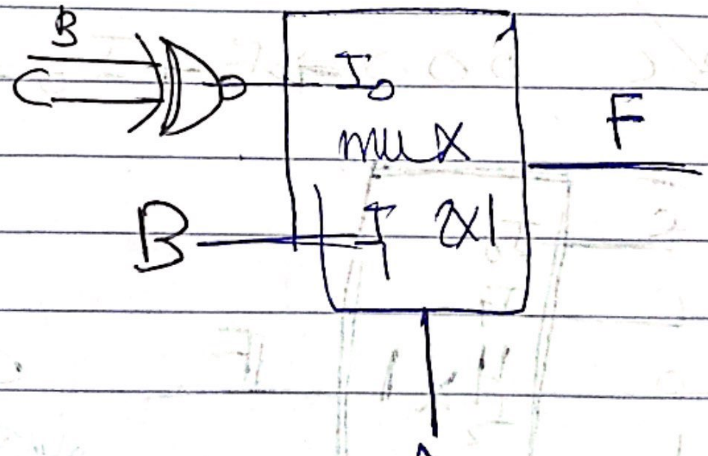
(2X1)

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

B\C	0	1
0	1	
1		1

$F = B'C' + BC$

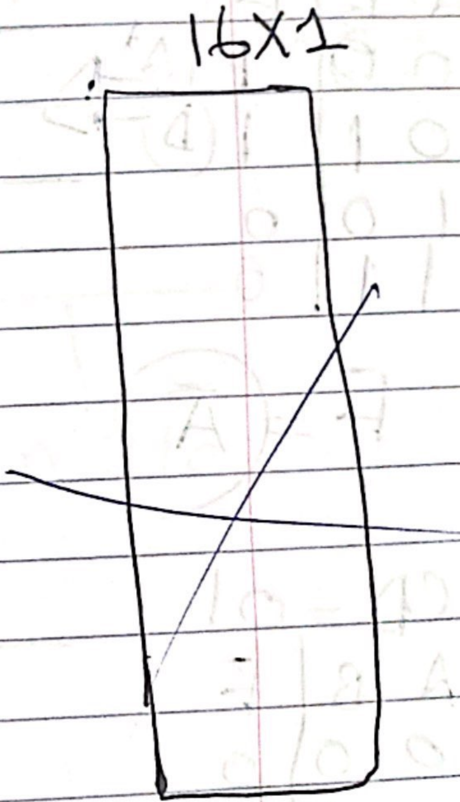
$(B \oplus C)'$

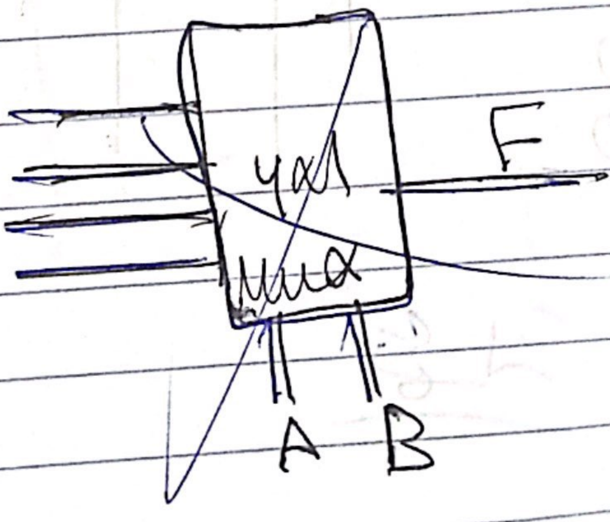


F(A,B,C,D) = Σ(0,2,4,5,6,10,14,15)

4x1 Mux

A	B	C	D	F	
0	0	0	0		1
0	0	0	1		0
0	0	1	0		1
0	0	1	1		0
0	1	0	0		1
0	1	0	1		1
0	1	1	0	0111	0
1	0	0	0		0
1	0	0	1		0
1	0	1	0		1
1	0	1	1		0
1	1	0	0		0
1	1	0	1		0
1	1	1	1		1



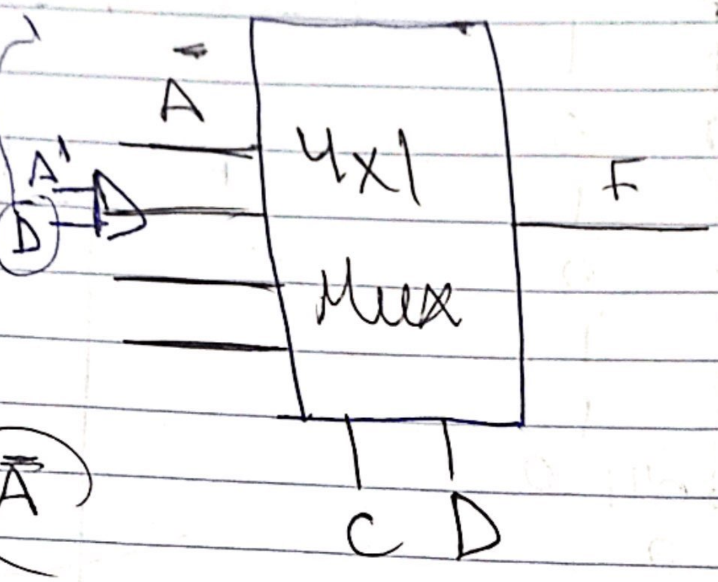


CD selection

(بدك) B,1

CD = 00

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



$F = \bar{A}$

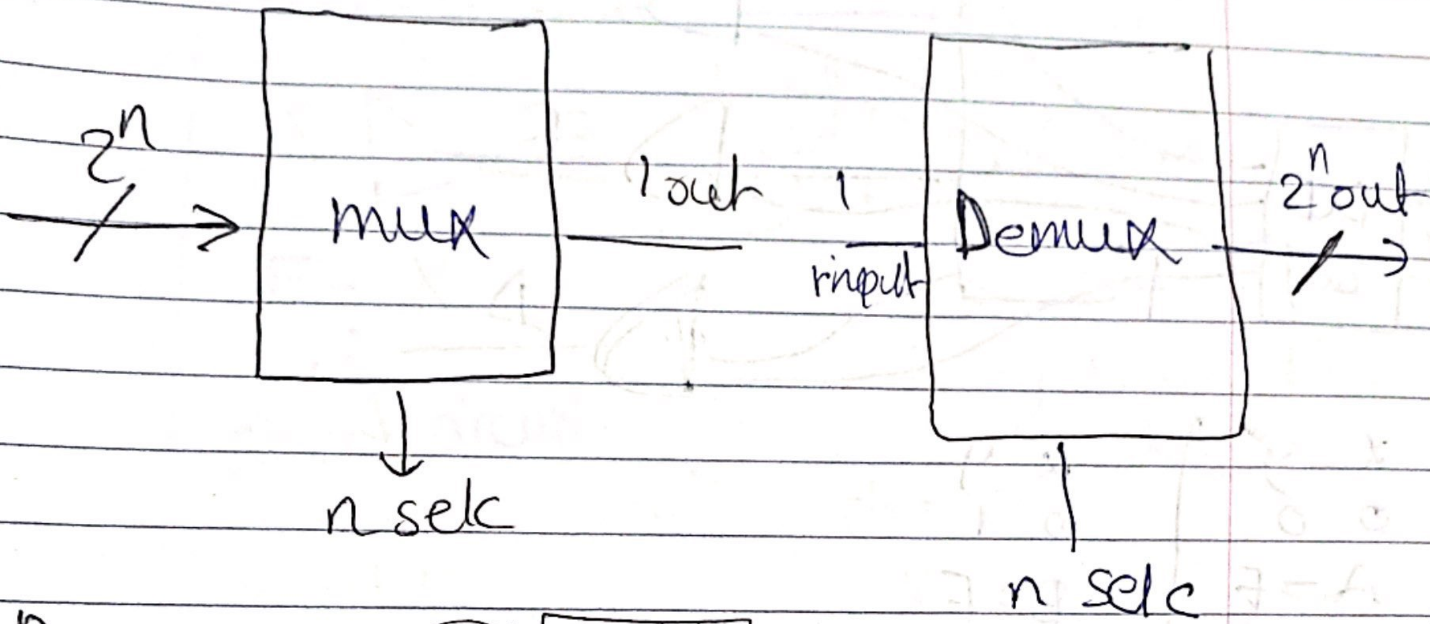
CD = 01

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

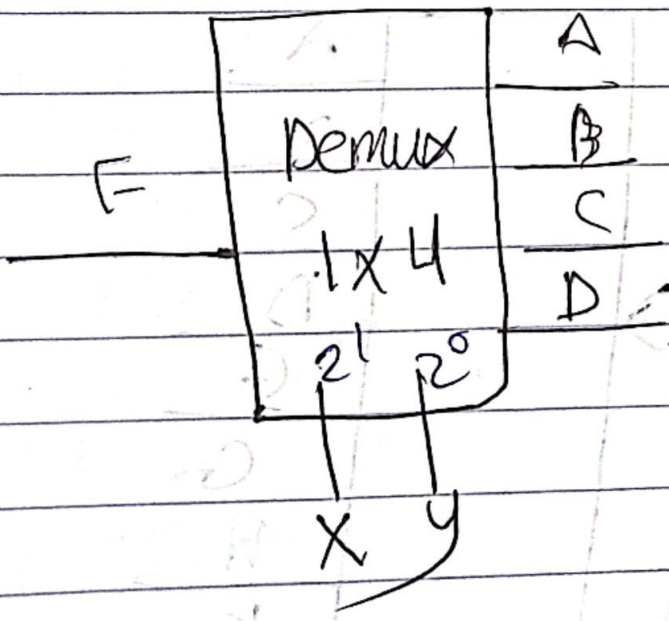
A\B	0	1
0		1
1		

$F = A'B$

Demultiplexer (De mux)

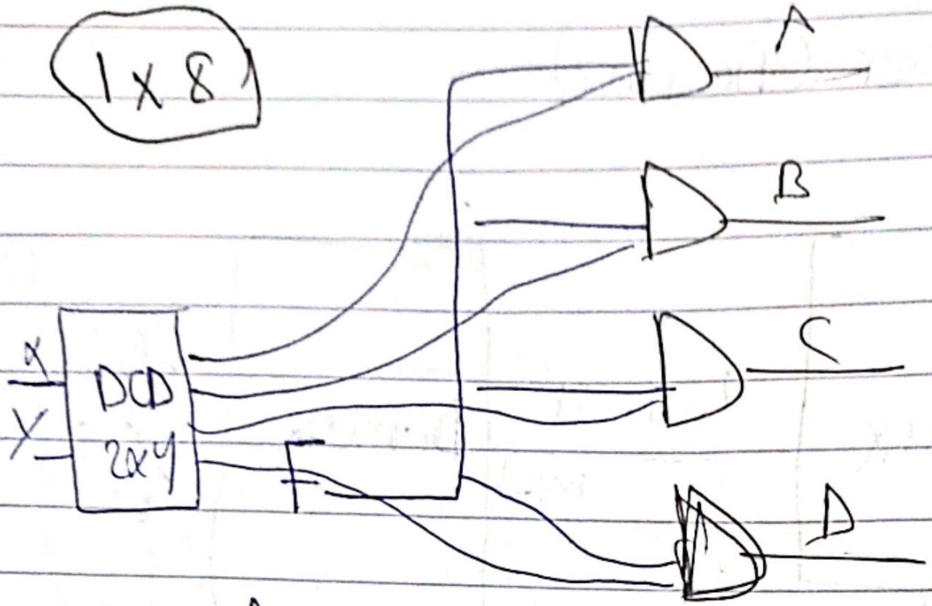


Demux 1 x 4



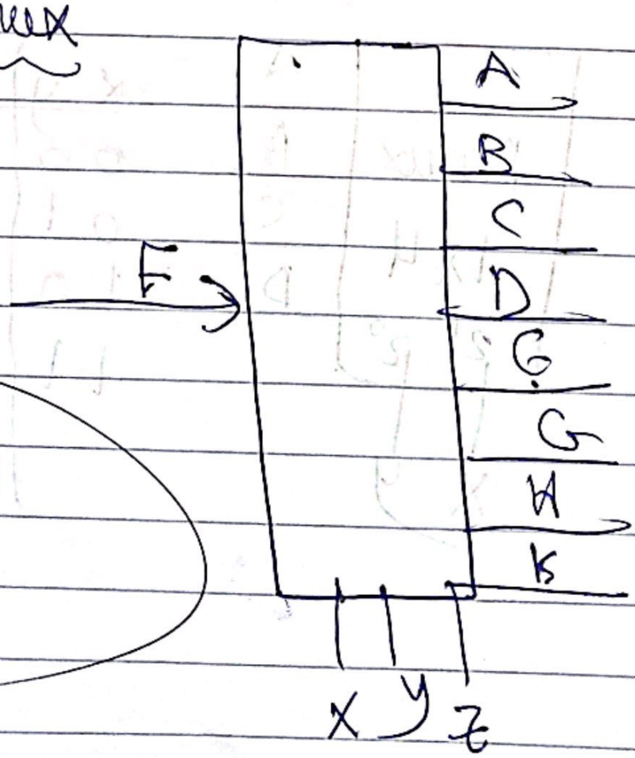
x	y		
0	0	A=F	B, C, D=0
0	1	B=F	A, C, D=0
1	0	C=F	
1	1	D=F	

1x8



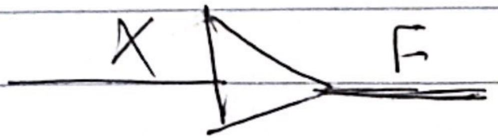
x	y	x	y
0	0	0	1
A = F		B = F	

1x8 Demux

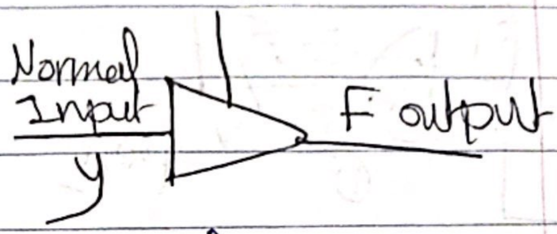


Three (Tri) state buffer

buffer (normal)



control (X)

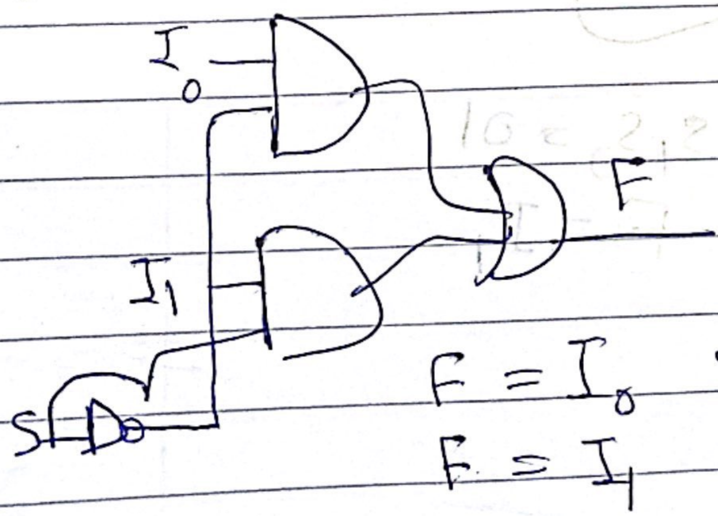


F = X
short circuit

if X == 1
F = y (normal) buffer
else
X == 0
(open circuit)
high impedence

x	y	F
0	0	X
0	1	X
1	0	0
1	1	1

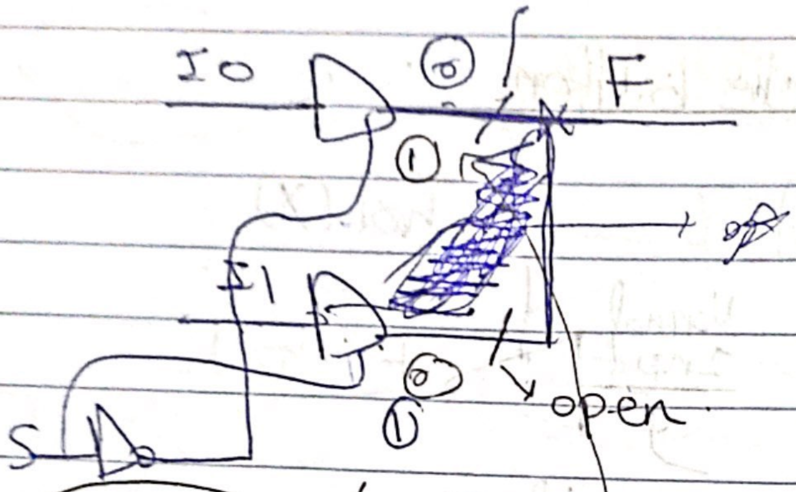
2x1 mux



$F = I_0 \quad S = 0$

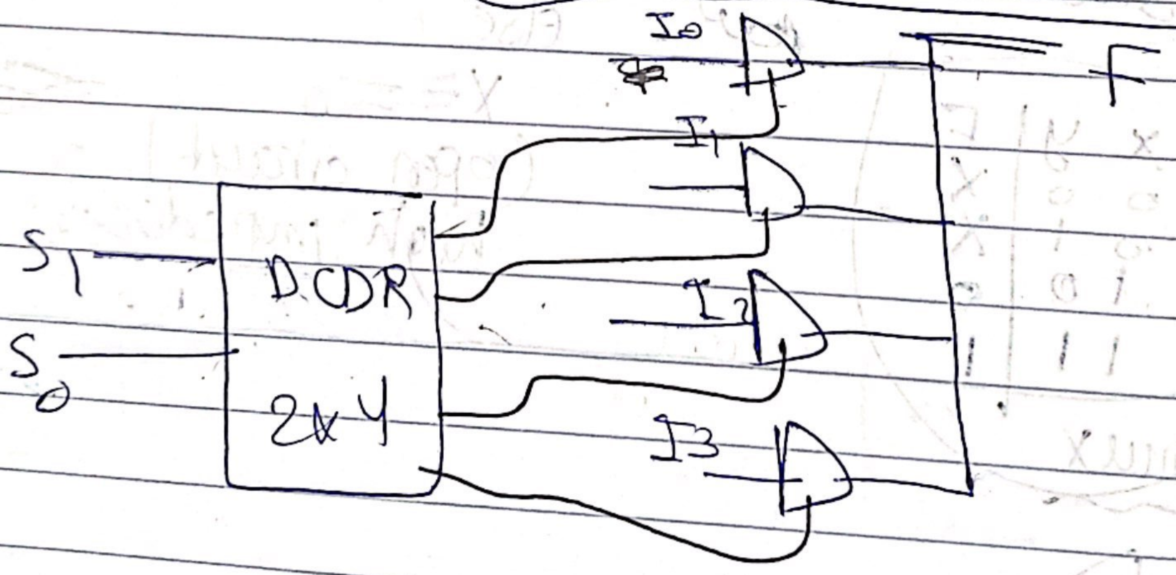
$F = I_1 \quad S = 1$

عمر تحكم



S=0 / $F = I_0$

S=1 / $F = I_1$.



$S_1 S_0 = 00$
$F = I_0$

$S_1 S_0 = 01$
$F = I_1$