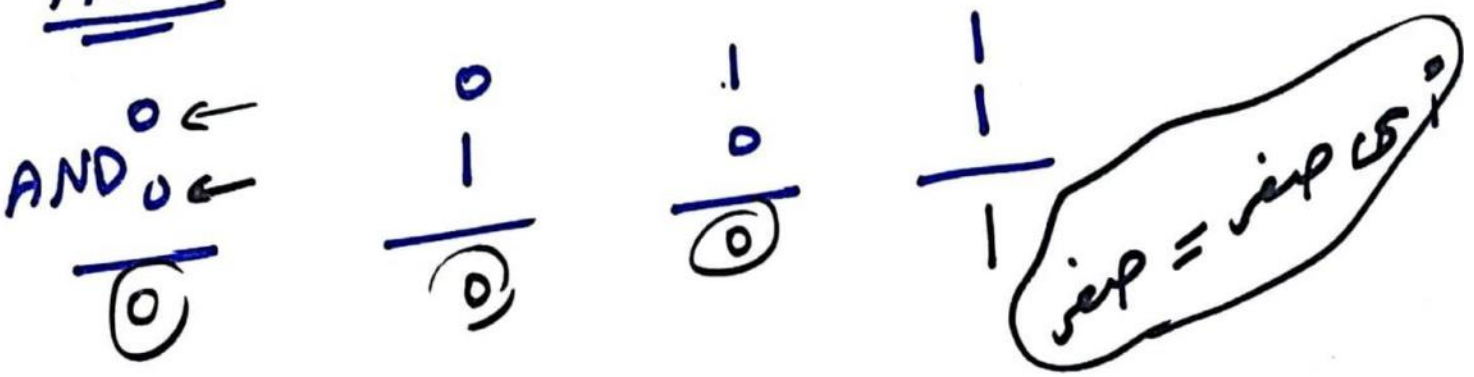
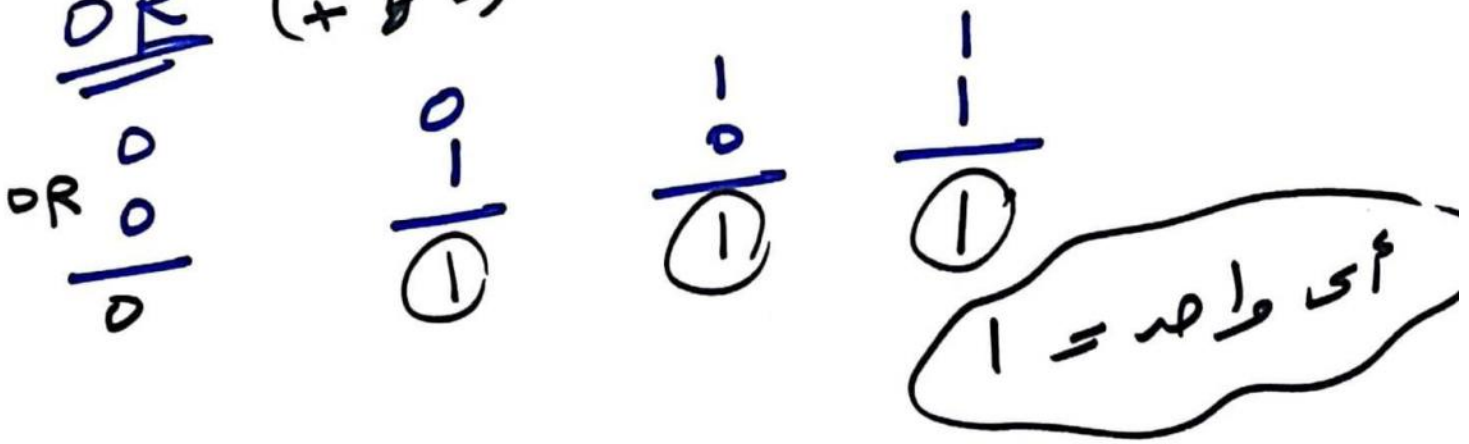
Logical operations — 3 operations.

AND OR XOR

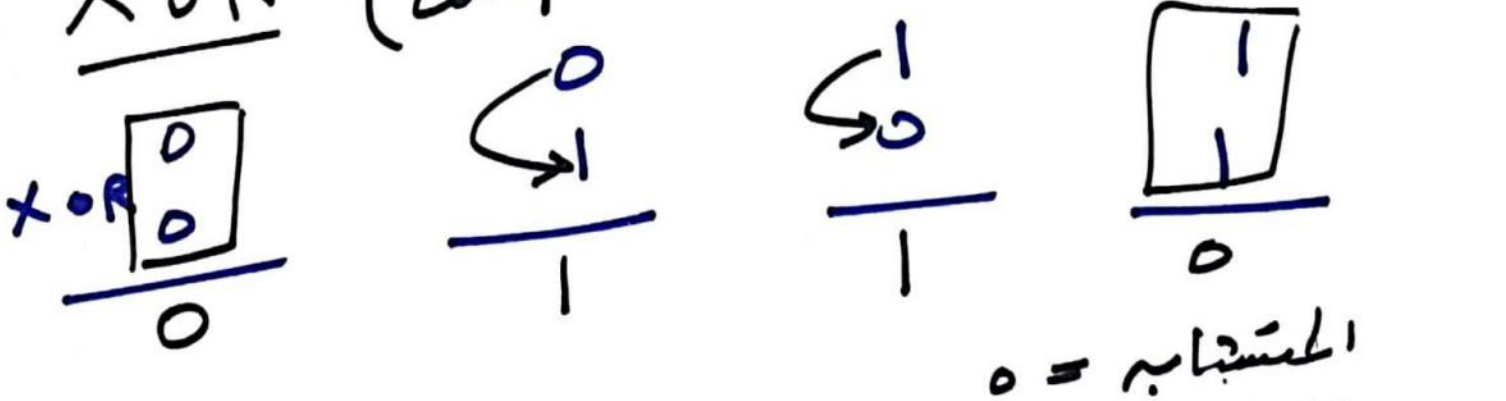
AND (ضرب)



OR (+ بالكلام)



XOR (complement)



~~XOR~~

AND

	0 0	0	
AND	1 0	1 1 0	1 0
	1 0 0 0 0	0	0

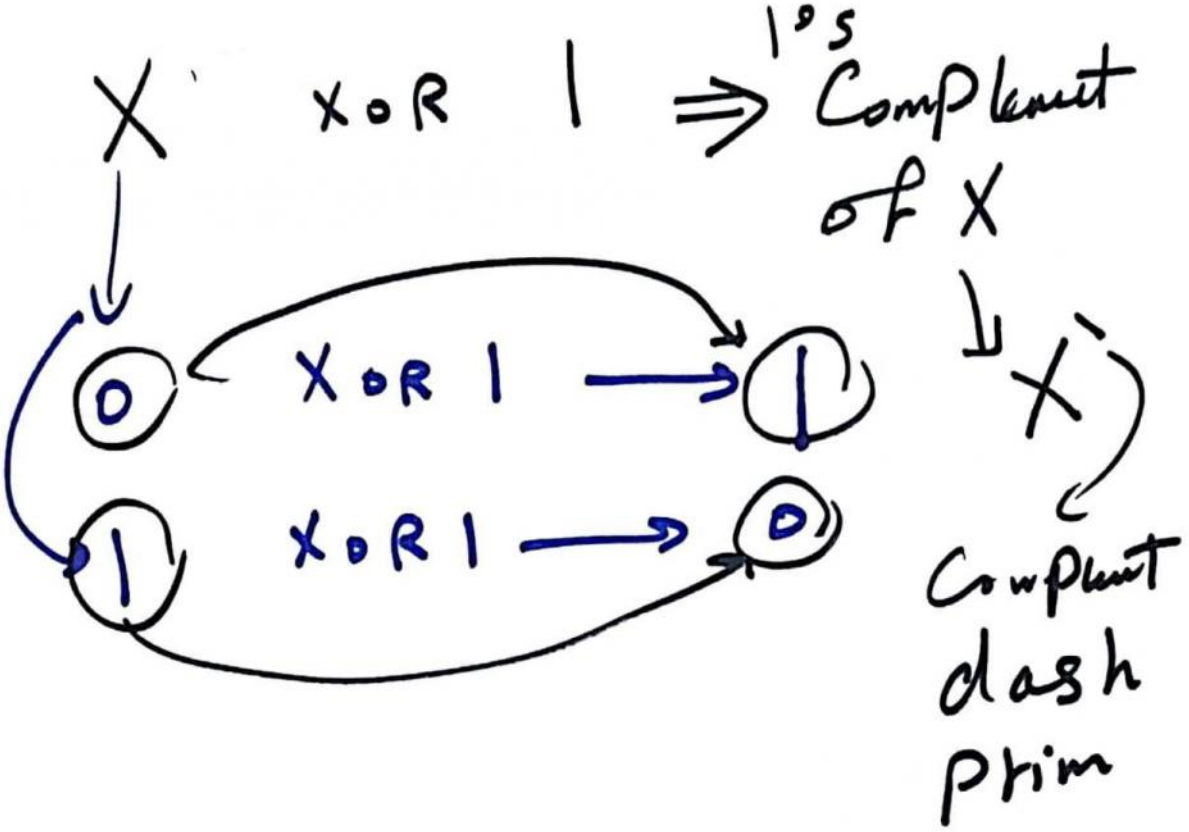
OR

	1 0 0	0	1	0
OR	1 0	1 1 0	1	0 0
	1 1 1 1 1	1	1	1 0

XOR

	1 0 0	0	1	●
XOR	0 1 1	0	1	0 ●
	1 0 1	0 0	1	0

المتشابه = 0
المختلف = 1



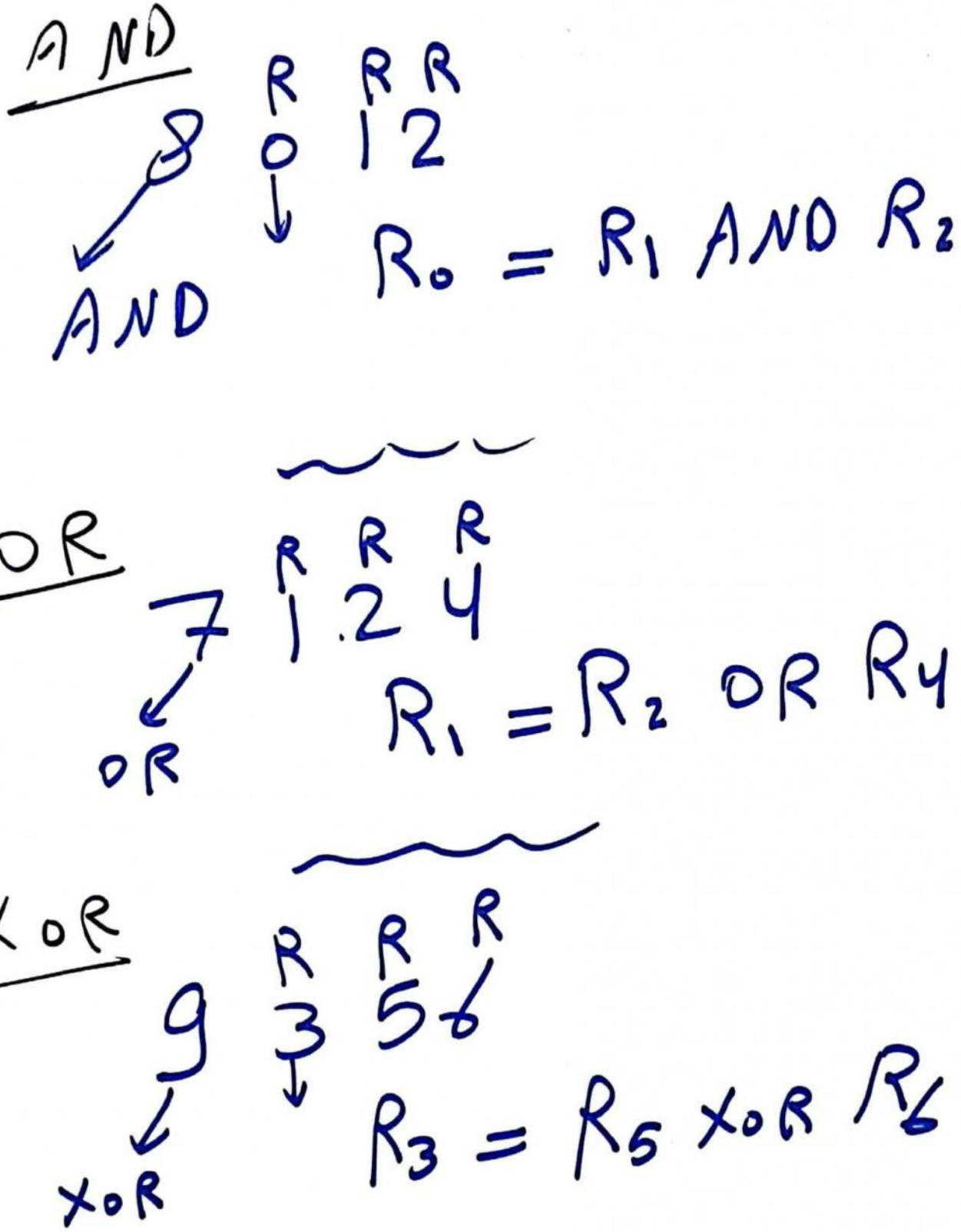
y XOR 1 → y'

z XOR 1 → z'

X XOR 0

1 XOR 0 → 1

0 XOR 0 → 0


AND
8
AND
R R R
0 1 2
$R_0 = R_1$ AND R_2
OR
7
OR
R R R
1 2 4
$R_1 = R_2$ OR R_4
XOR
9
XOR
R R R
3 5 6
$R_3 = R_5$ XOR R_6

logical masks. / filters.

8bits

X X X X y y y y

F 0

AND 1 1 1 1 0 0 0 0

operation mask

Result

X X X X 0 0 0 0

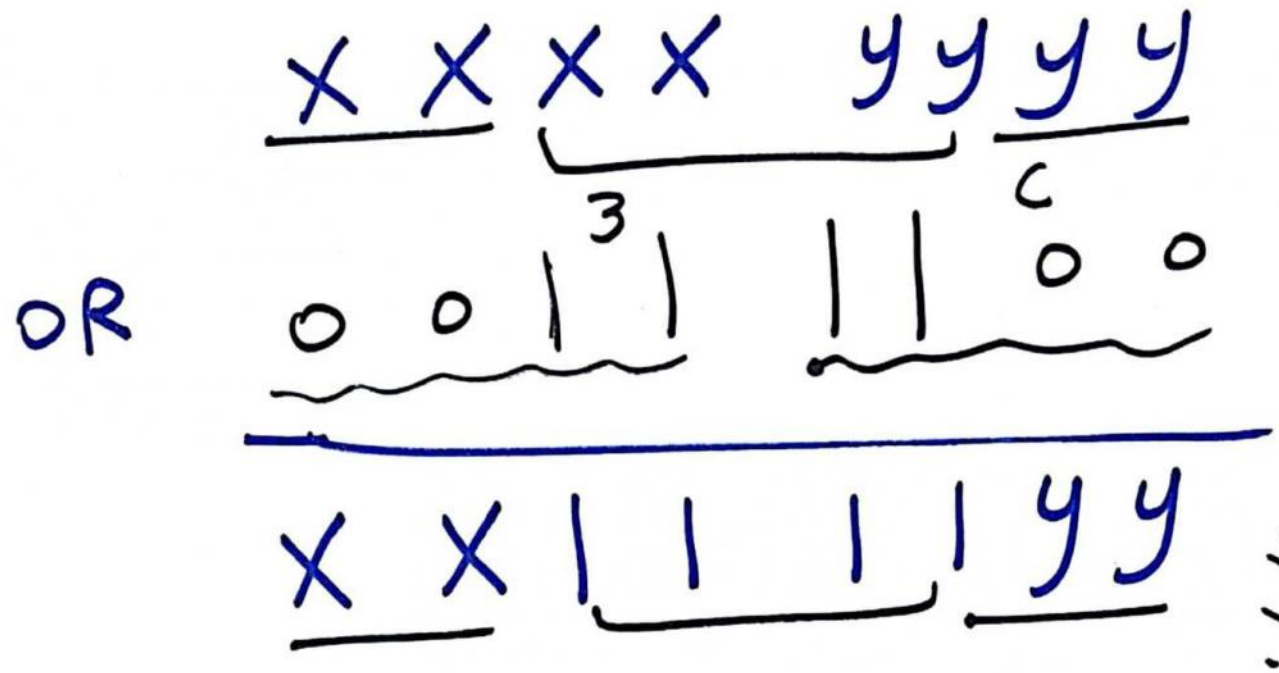
The operation will be AND with the mask (11110000) = $(F0)_{16}$

X X X X y y y y

XOR 0 0 0 0 y y y y

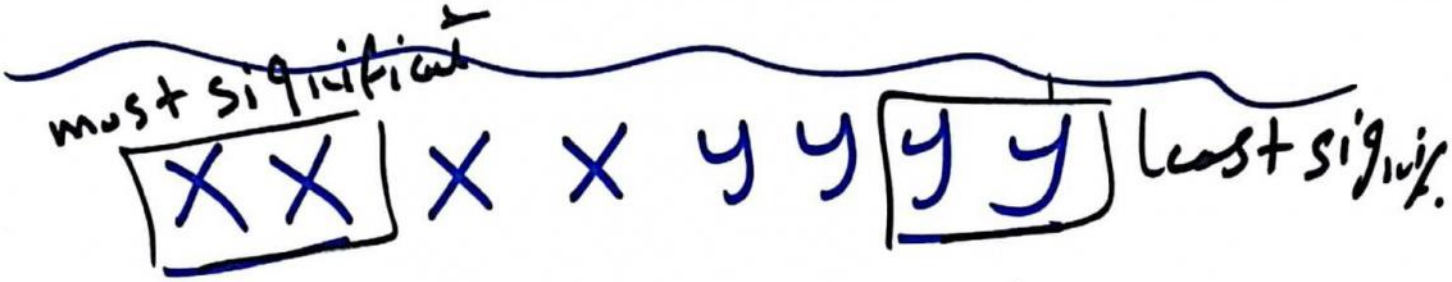
The first 4 bits = 0

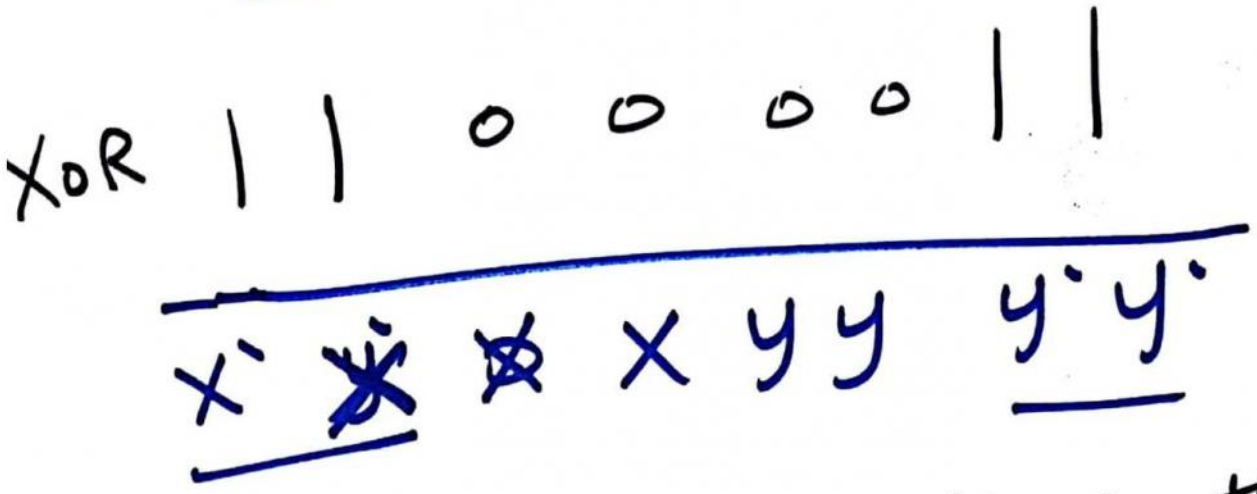
X X X X 0 0 0 0



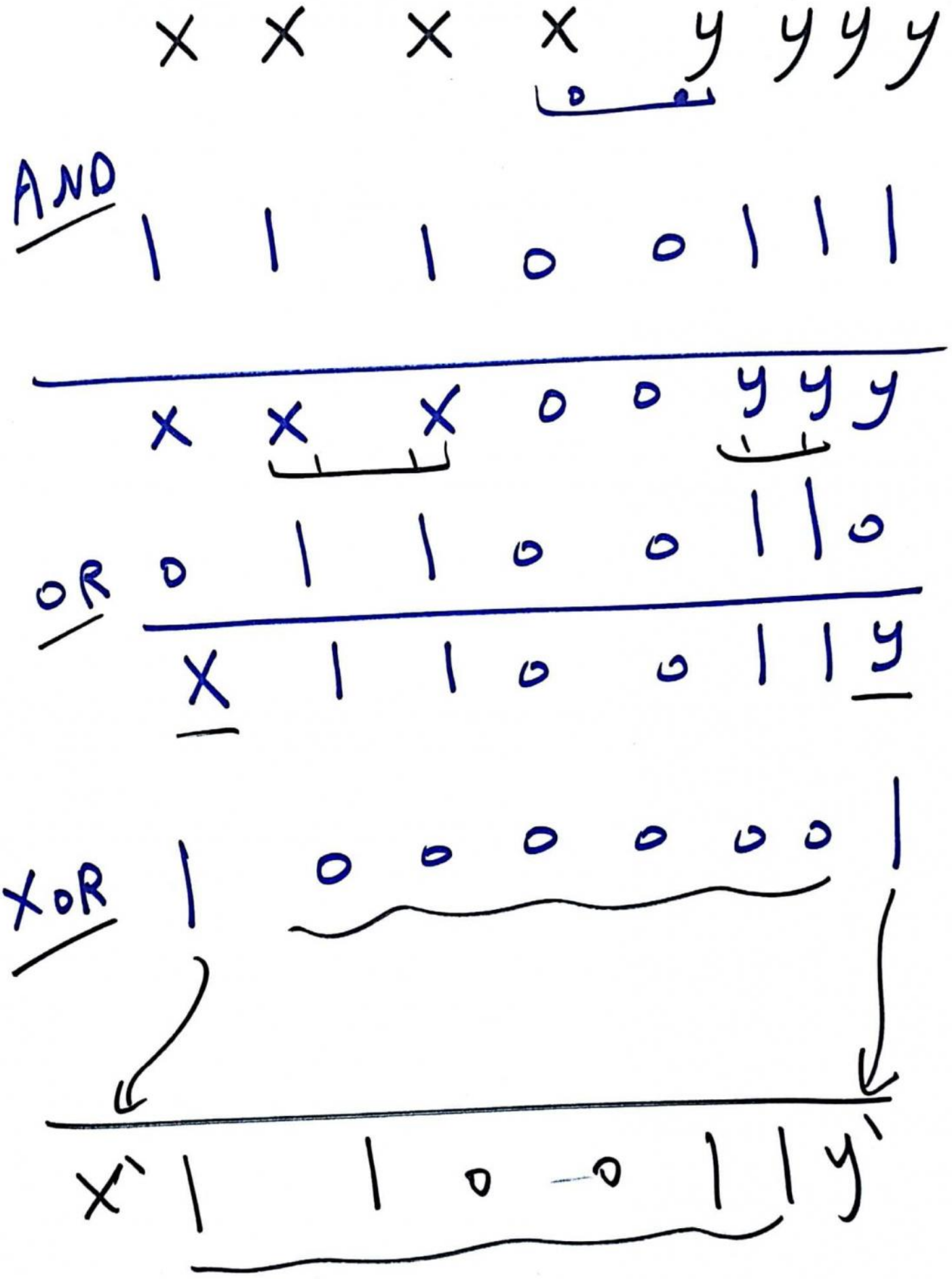
operation ⇒ OR with the mask $(3C)_{16}$

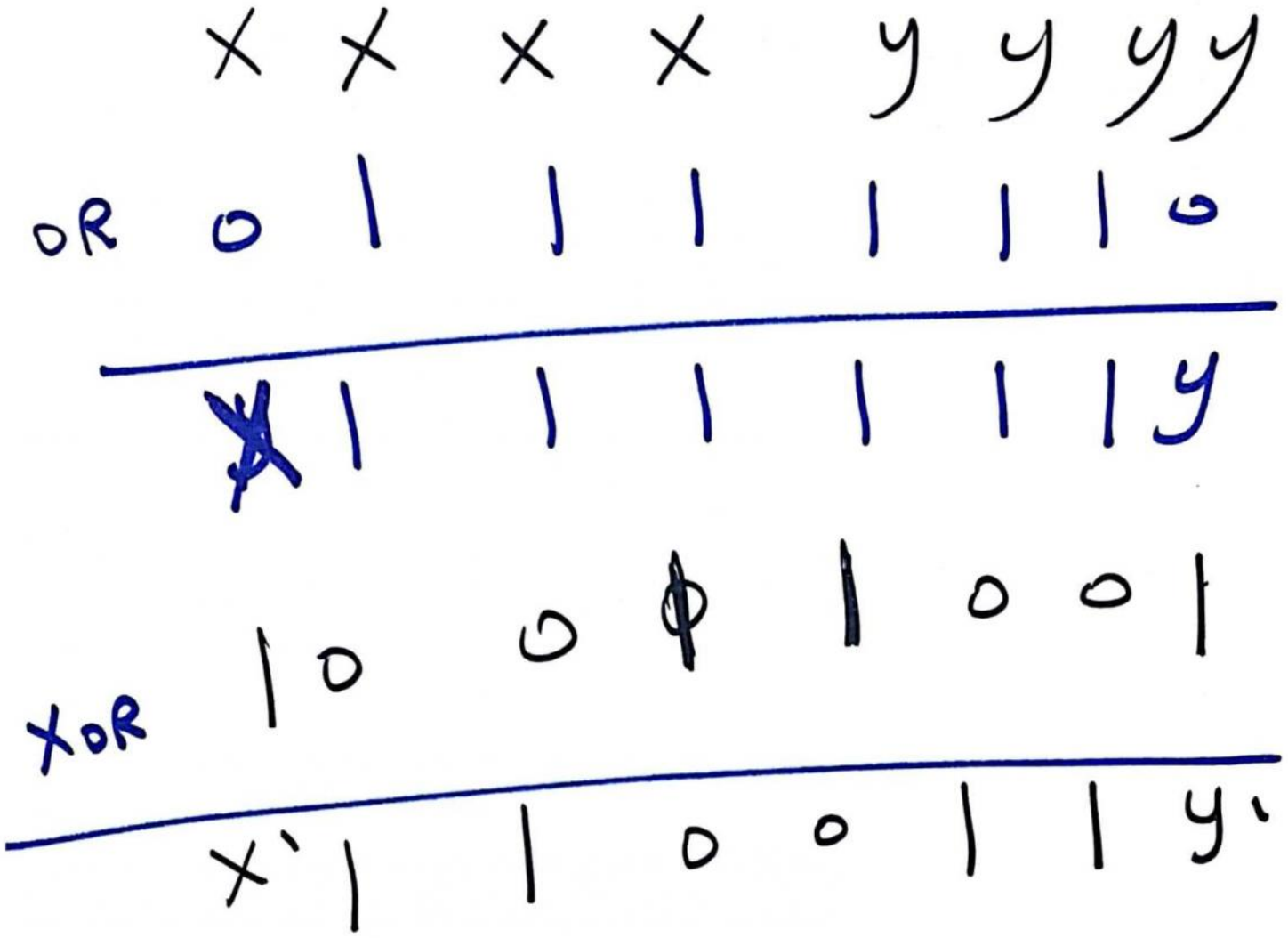
the middle four bits = 1

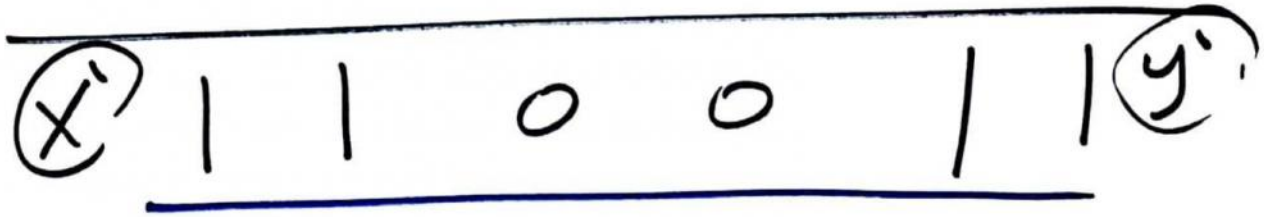




The most significant 2 bits & the least significant 2 bits are complemented.


x x x x y y y y
AND
1 1 1 0 0 1 1 1
x x x 0 0 y y y
OR
0 1 1 0 0 1 1 0
X 1 1 0 0 1 1 y
XOR
1 0 0 0 0 0 0 1
X' 1 1 0 0 1 1 y'

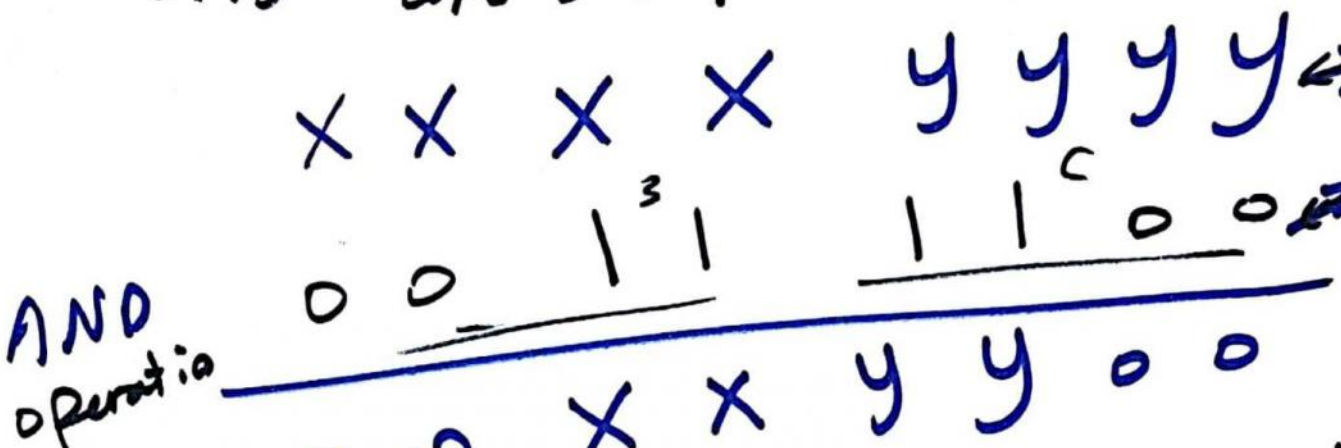




هنشتغل بـ OR ← 1

Ex:

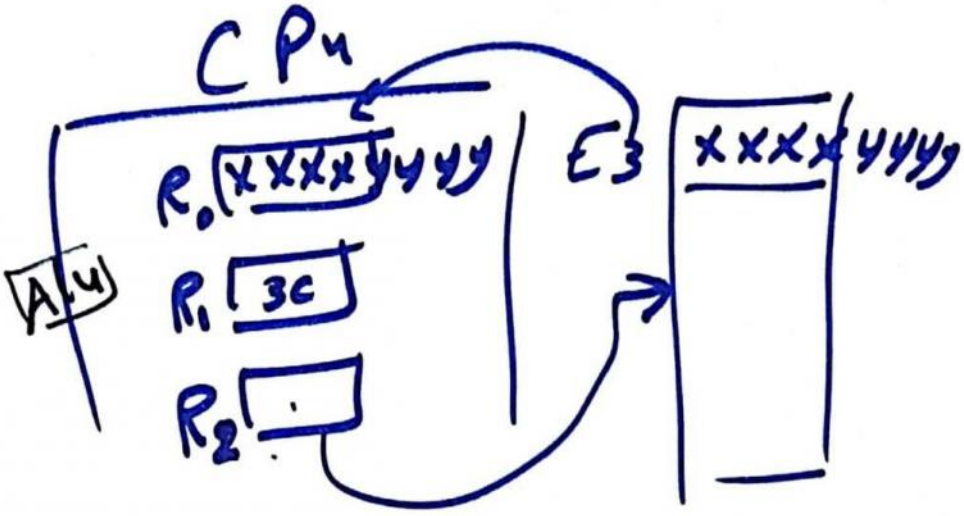
a) What is the mask with what operation that must be used to make the middle 4 bits of eight bits string as it & the other bits are 0's?



→ We will use "AND" operation with the mask → $(3C)_{16}$

b) * Write the program that make this if the number is in

M[E3]



step 1: 1 0 E3 → load $R_0 \leftarrow M[E3]$

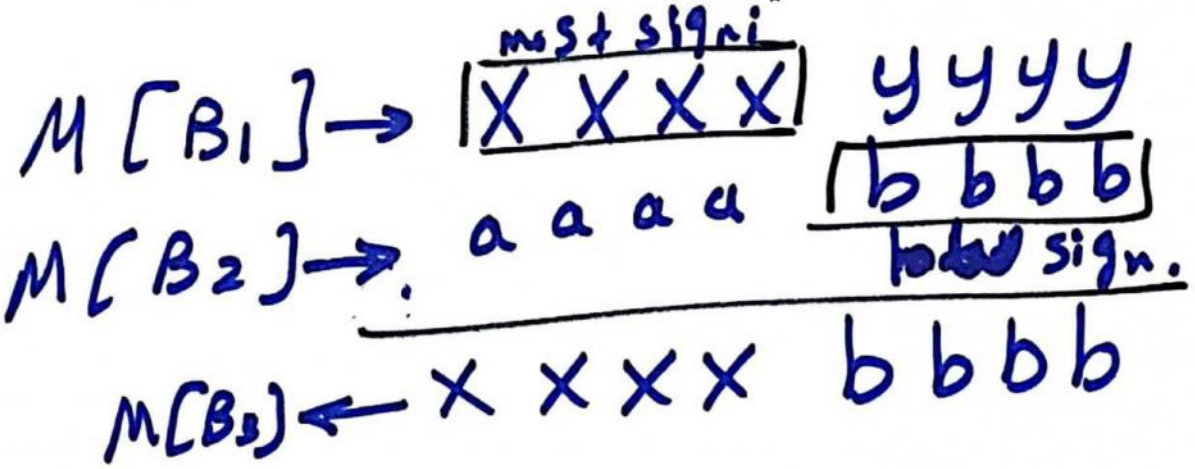
step 2: 2 1 3C → load $R_1 \leftarrow 3C$

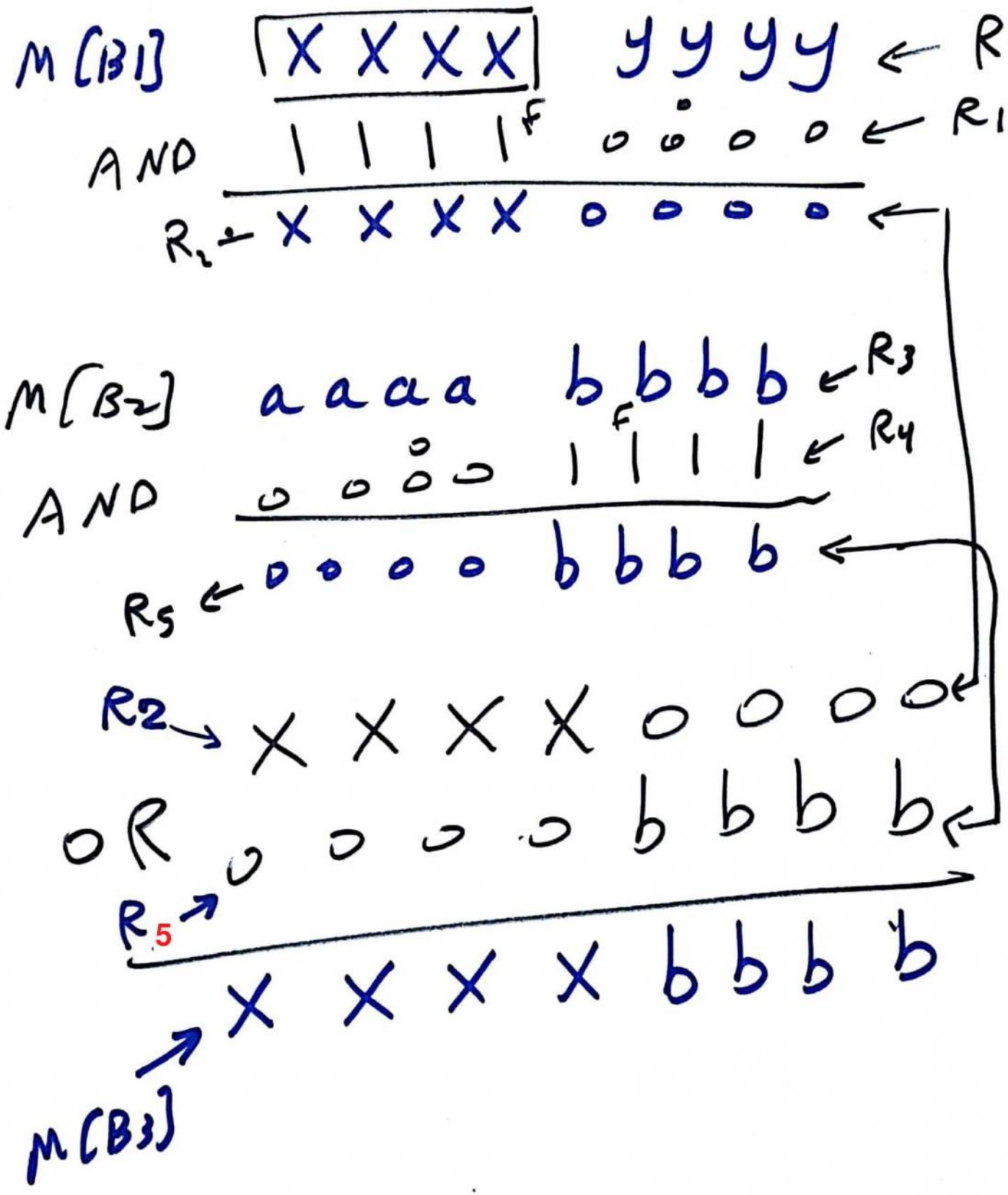
step 3: 8 2 0 1 → AND $R_2 \leftarrow R_0 \overset{AND}{\&} R_1$

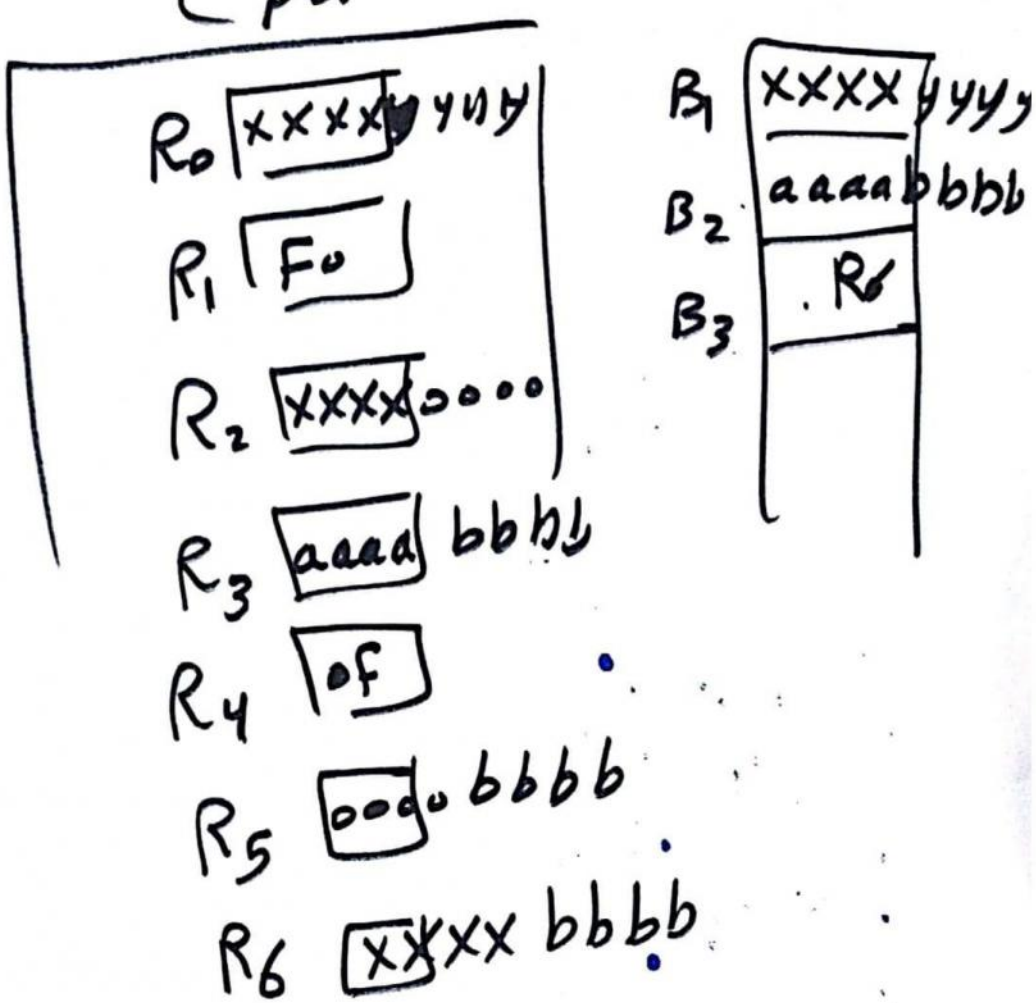
step 4: 3 2 E3 → store $R_2 \rightarrow M[E3]$

step 5: C0 0 0 → halt

*Write a machine language Program That: Combines the most Significant four bits of the memory cell Whose address is B_1 with the low significant 4bits of the memory Cell B_2 and Put the Result in memory Cell Whose Address is B_3?




M[B1]
X X X X
y y y y
R0
AND
1 1 1 1
F
0 0 0 0
R1
R2
X X X X 0 0 0 0
M[B2]
a a a a
b b b b
F
R3
AND
0 0 0 0
1 1 1 1
R4
R5
0 0 0 0 b b b b
R2
X X X X 0 0 0 0
OR
0 0 0 0 b b b b
R5
X X X X b b b b
M[B3]


CPU
R0 xxxx yyyy
R1 F0
R2 xxxx 0000
R3 aaaa bbbb
R4 0f
R5 00 bbbb
R6 xxxx bbbb
B1 xxxx yyyy
B2 aaaa bbbb
B3 R6

1) load $R_0 \leftarrow M[B_1]$ → 1 0 B 1

2) load $R_1 \leftarrow F_0$ → 2 1 F 0

3) AND $R_2 \leftarrow R_0, R_1$ → 8 2 0 1

4) load $R_3 \leftarrow M[B_2]$ → 1 3 B 2

5) load $R_4 \leftarrow$ 0F → 2 4 0F

6) AND $R_5 \leftarrow R_3, R_4$ → 8 5 3 4

7) OR $R_6 \leftarrow R_2, R_5$ → 7 6 2 5

8) store $R_6 \rightarrow M[B_3]$ → 3 6 B 3

9) halt → C 0 0 0

بسم الله الرحمن الرحيم

Shifting

Shift to Right (÷)

Ex. Shift (100) to the Right

1. Location.

$100 \div 10$ (4 → 2)

$= 10 = (2)_{10}$

1- Location means $2^1 \Rightarrow 2$ ÷2

2- location means $2^2 \Rightarrow 4$ ÷4

3- locatin means $2^3 \Rightarrow 8$ ÷8

Examples:

EX1 shift to Right (3-location)

$10000 \div 1000$

$= 10$

EX2 shift to Right (2-location)

$10100 \div 100$

$= 101$

Shift to Left (*)

Ex. Shift (101) to the left

1. location.

$101 \rightarrow 10100$ (5 → 2)

$= 101 * 10$

$= 1010 = (10)_{10}$

1. location means 2^1 * 2

2. Location means 2^2 * 4

3. Location means 2^3 * 8

EX3 shift to left (2-locat

$100 * 100$

$= 10000$

EX4 shift to left (3 Location

$1110 * 1000$

$= 1110000$

EX5

shift to Right

$100.10 \div 10$

(1-location)

$= 10.0$

EX6

shift to Right (2-location)

$1100.100 \div 100$

$= 11.00$

EX7

shift to left (2-Location)

11.10×100

$= 1110.100$

Rotate Operations

Left Rotate

Called Circular Left shift

EX:

Perform a rotation operation 2-location to the Left for the number:

(10110110)

حيث أننا نحرك (2 bits) من اليسار وضعهم بنفس ترتيبهم في اليمين.

⇒ 11011010

Right Rotate

Called circular Right shift

EX:

Perform a rotation operation 3-Locations to the Right for (11101001)

(3 bits) حيث أننا نحرك من اليمين ونضعهم بنفس ترتيبهم في الشمال.

⇒ 00111101

The Rotate instruction ⇒ "A"

A 5 0 1

Right Rotate ← R_5 ← (1-Location)

↳ this instruction makes Right Rotate ⇒ (1-bit) (1-Location)
For the Contents in Register 5.
and Puts the Result in the same Register (R_5)

EX

↳ If $R_5 = E3$

↳ After execution of the instruction A501,
What is the value Content in R_5?

ANS.

$R_5 = (E3)_{16}$ ← Right Rotate (1-Location)

⇓

1 1 1 0 0 0 1 (1)

1 1 1 1 0 0 0 1

F 1

R_5 will have the value (F1)

*Suppose the memory cells at addresses (E4) to (F7) Contain the following program:-

Add.	Content
E4	20
E5	F2
E6	21
E7	CB
E8	22
E9	01
EA	23
EB	01
EC	B1
ED	F6

Add.	Content
EE	A1
EF	01
F0	53
F1	32
F2	33
F3	FE
F4	B0
F5	EC
F6	C0
F7	00

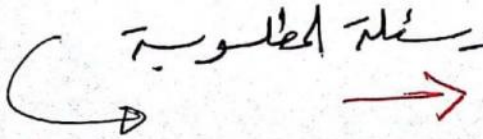
Instruction Table

Start Address	instruction	Meaning.
E4	20F2	LOAD $R_0 \leftarrow$ F2
E6	21CB	LOAD $R_1 \leftarrow$ CB
E8	2201	LOAD $R_2 \leftarrow$ 01
EA	2301	LOAD $R_3 \leftarrow$ 01
EC	B1F6	Jump to M[F6] if $R_1 = R_0$
EE	A101	Right Rotate For R_1 (1-location)
F0	5332	Add $R_3 \leftarrow R_3 + R_2$
F2	33FE	Store $R_3 \rightarrow$ M[FE]
F4	B0EC	Jump to M[EC] if $R_0 = R_0$ (unconditional Jum
F6	C000	halt/stop.

Trace Table

Start Address	Trace	M[FE]	R_0	R_1	R_2	R_3
		~~02~~ ~~03~~	F_2	~~CB~~ ~~E5~~ F_2	01	~~01~~ ~~02~~ 03
E4	$R_0 = F_2$					
E6	$R_1 = CB$					
E8	$R_2 = 01$					
EA	$R_3 = 01$					
EC	if $R_1 = R_0$ X					
EE (Rotate)	$R_1 = CB = 11\phi01011$ $= 11\overset{E}{1}\phi0\overset{5}{1}01$					
F0	$R_3 = R_3 + R_2 = 02$					
F2	$M[FE] = R_3 = 02$					
F4	$R_0 = R_0$ ✓					
EC	if $R_1 = R_0$ X					
EE (Rotate)	$R_1 = E5 = 1110010\phi1$ $= 11110010$ F 2					
F0	$R_3 = R_3 + R_2 = 03$					
F2	$M[FE] = R_3 = 03$					
F4	$R_0 = R_0$ ✓					
EC	if $R_1 = R_0$ ✓					
F6	halt / stop.					

بعد ذلك انجاوب ع اسئلة المطلوبة



1) What is the content of Register 1 after the first time the instruction EE is executed?

$R_1 = E5$

---

2) What is the content of R_1 after the second time the instruction EE is executed?

$R_1 = F2$

---

3) How many times the instruction at address EC is executed before the machine halts?

3 times

---

4) What is the content of R_3 after the Program Terminates?

$R_3 = 03$

---

5) What will happen if the Program starts execution with Replacing the contents at address EF to be 08?

لو بدل ما خزنا قيمة محتويات M[EF] الى (08) من الجدول اللي فوق

Add	content
EE	A1
EF	08

أمر الـ Rotate سيصبح A108

ومعنى إننا نعمل Rotate بمقدار (08)

يعني معيش أي تغير هيحصل يعني البرنامج هيكمل غير متوقف أبدًا

⇒ After the Rotate instruction the number at R_1 will be as it each time, and the Jump operation at (EC) won't be true

So, it will be (infinite Loop)

Ex. (From the book P.160)

⇒ Write a machine language Program that Copies the middle Four bits From memory Cell E_0 into the least significant Four bits of memory Cell E_1, while Placing 0's in the ~~t~~ most significant Four bits of the same cell.

Ans.

مثال توضيحي

↳ if M[E_0] has (a a b b b b a a)

↳ We need to change the number in M[E_0] to be

most significant least significant

0 0 0 0 b b b b

Then, Put the new number ↑ in M[E_1]

① We will make "Rotate" "operation" (2-location) For (a a b b b b a a) to be

(a a a a b b b b)

② We will make "AND" operation with (0F) Mask

a a a a b b b b

AND 0 0 0 0 1 1 1 1 (F)

0 0 0 0 b b b b

↳ Then store this number in M[E_1]

→

* لحل البرنامج ← لابد من تحميل كل القيم التي سوف نستخدمها داخل (R)

Steps	Symbolic form	Instruction form
step 1 →	LOAD $R_0 \leftarrow M[E_0]$	1 0 E 0
step 2 →	Rotate R_0, 2-Location	A 0 0 2
step 3 →	LOAD R_1, 0F "AND Mask"	2 1 0 F
step 4 →	AND $R_2 \leftarrow R_0$ AND R_1	8 2 0 1
step 5 →	STORE $R_2 \rightarrow M[E_1]$	3 2 E 1
step 6 →	halt / stop.	C 0 0 0

EX:

* It is Required to Change the Content of Memory Cell C_2 from (b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0) to be (b_3 o b_1' | | b_6' o b_4)

* Determine the Required masks and logical operations to make such a Change, Then Write a machine Program to Perform the above operations?

ANS.

→

للتحويل من الرقم $(b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0)$

إلى $(b_3\ 0\ b_1\ 1\ 1\ b_6\ 0\ b_4)$

① نحتاج أولاً لعمل (Rotate) بمقدار أربع حركات لكي يصبح الرقم

$b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$

$b_3\ b_2\ b_1\ b_0\ b_7\ b_6\ b_5\ b_4$

② نستخدم (OR mask) وذلك لجعل الأرقام التي نحتاجها في الناتج صفر أو ١ كما في (١) كالآتي.

$$\begin{array}{cccccccc} b_3 & b_2 & b_1 & b_0 & b_7 & b_6 & b_5 & b_4 \\ \text{OR} & & & 5 & & & A & \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ \hline b_3 & 1 & b_1 & 1 & 1 & b_6 & 1 & b_4 \end{array}$$

ثم نستخدم

$$\begin{array}{cccccccc} & & & 6 & & & 6 & \\ \text{XOR} & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ \hline b_3' & 0 & b_1 & 1 & 1 & b_6' & 0 & b_4 \end{array}$$

ولفتح (١) فالأرقام التي نحتاجها صفر نستخدم Complement

خطوات عمل البرنامج

⟶

خطوات عمل البرنامج :

① تحمل الرقم المراد تحويله والموجود في M[C2] داخل R_0

② تحمل Rotate لهذا الرقم بمقدار (4 bits) وتخزنه الناتج في R_0

~~③ تحمل (OR Mask) الرقم الموجود في R_0~~

③ تحمل الـ (OR Mask) والذي قيمته (5A) داخل R_1

④ تعمل (OR operation) بين R_1 S R_0 وتضع الناتج في R_2

⑤ تحمل (XOR mask) والذي قيمته (66) داخل R_3

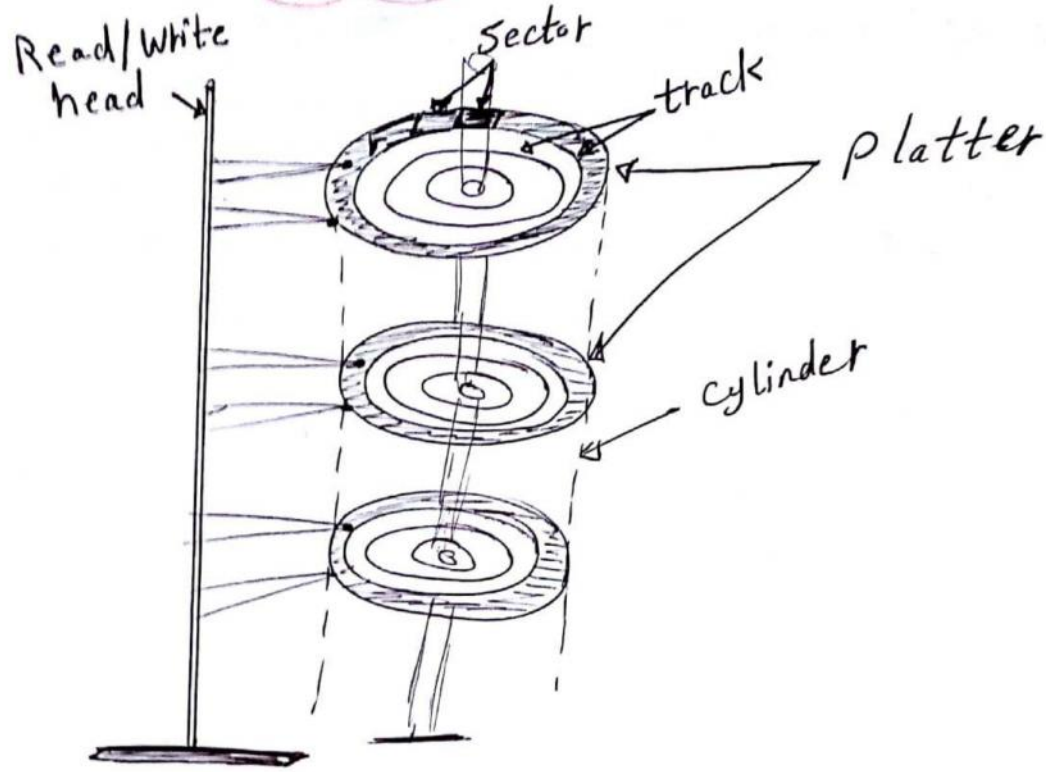
⑥ تعمل (XOR operation) بين R_3 S R_2 وتضع الناتج داخل R_4

⑦ تخزنه (R_4) داخل M[C2] .

⑧ تعلم البرنامج .

steps	symbolic form	Instruction Form
step1:	LOAD $R_0 \leftarrow M[C2]$	1 0 C2
step2:	Rotate R_0, 4-Location.	A 0 0 4
step3:	LOAD $R_1 \leftarrow 5A$	2 1 5A
step4:	OR $R_2 \leftarrow R_1$ OR R_0	7 2 1 0
step5:	LOAD $R_3 \leftarrow 66$	2 3 6 6
step6:	XOR $R_4 \leftarrow R_2, R_3$	9 4 2 3
step7:	STORE $R_4 \rightarrow M[C2]$	3 4 C2
step8:	halt/stop.	C 0 0 0

Hard disk



* الـ (hard disk) بتتكون من عدد من الـ (Platters)

* كل (platter) له وجهين لتخزين البيانات (2 sides)

* كل (Side) عليه عدد من الـ (tracks)

* كل (track) متقسم لمجموعة من الـ (sectors) كلهم متساوية

حجم الـ track = عدد الـ sectors × حجم الـ sector

* الـ (Cylinder) ← هو تجميع لنفس الـ (track) من كل الـ (platters) الموجودة

حجم الـ Cylinder = عدد الـ platters × 2 (وجهين sides) × حجم الـ track

* حجم الـ hard يمكن حسبه بطريقتين حسب المعطيات:

① حجم الـ hard = عدد الـ tracks × عدد الـ sectors × حجم الـ sector ÷ حجم الـ platter × عدد الـ platters × 2 (وجهين sides)

② حجم الـ hard = عدد الـ cylinders × عدد الـ sectors × حجم الـ sector

Ex:

* If the size of sector = 512 B / sector (per)
  * No. of sector/track = 63 sector/track.
* No. of platters = 8

a) Find the size of cylinder.

b) If the hard disk has 100 cylinders find the size of the hard disk.

Ans.

a) The size of cylinder = size of track X 2 X No. of Platters

size of track = No. of sector X size of one sector

= 63 X 512

= 32256 Bytes

The size of cylinder = size of track X 2 X No. of Platter.

= 32256 X 2 X 8

= 516096 Bytes.

b) If the hard disk has 100 cylinder ⇒

The hard disk size = Cylinder size X No. of Cylin

= 516096 X 100

= 516096 00 B ≈ 49 MB

المواضيع التي تم دراستها بالتفصيل

① Number systems.

← التحويل من وإلى (decimal)

← التحويل المباشر !!

$(\ )_4$ $(\ )_{16}$ $(\ )_8$

2bits 4bits 3bits.

② Addition & subtraction.

← وللطرح ثلاثة أنواع :-

- Direct → verified by decimal
- 1's Comp. → " " $(?)_{9's\ comp}$
- 2's Comp → " " 10's comp.

③ Signed Numbers:

From $(-)_{10} \rightarrow (?)_2$ عن طريق

From $(-)_2$ to $(?)_{10}$

محتاجين نجيب الموجب الأول عشان نعرف هل الرقم الـ decimal كبير ولا صغير.

- → unsign method
- → Signed magnitude
- → 1's Complement
- → 2's Complement
- → Excess notation.

④ Signed 2's complement subtraction

في نوع معين من المسائل بيقل الرقم لعدد الـ bits المعطاة في رأس السؤال ثم نحولها لرقم سالب بالـ 2's comp.

⑤ Floating point Representation

→ IEEE (طريقة خاصة)

← نوعية الرسالة التي سيتم تبادلها :-

- Code
- Decode.

⑥ Memory Capacity

لها قانون ثابت

Memory Capacity = no. of cell X Cell size

For Address line

$= 2^n$ X

⑦ Machine manipulating.

* How to Write machine language Program.
* Jump.
* Masks (AND, OR, XOR)
* Rotate, Shift (÷, *)

⑧ Hard disk.