

تقدم لجنة EICoM الاكاديمية

دفتر فاينل لمادة:

مختبر منطق رقمي

من شرح:

م. نسرين زيادنة

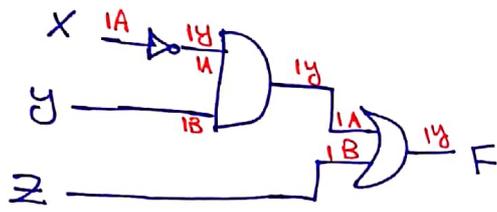
جزيل الشكر للطالبة:

نتالبي الكايد



EXP (1) :- Basic Logic gates + Boolean algebra, demorgan's theorem and Kmaps.

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



* $A, B \Rightarrow$ تجر عن الأبتوت

* $\bar{y} \Rightarrow$ تجر عن الأوتوت

* أرقام البوابات المنطقية: ICs

- \rightarrow and \rightarrow 08
- \rightarrow or \rightarrow 32
- \rightarrow not \rightarrow 04

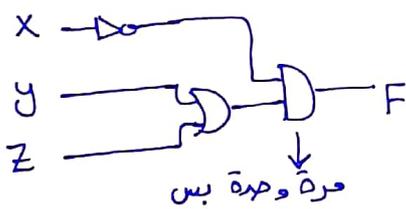
* input = x, y, z

* output = F (تشبيه على الليد)

* لو كان الاقتران كالتالي :-

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

ما بكر and مرتين ، بجعلها مرة وحدة بس



صا شكل and, or :-

VCC = 5V
GND = 0V

1	A ₁	VCC	14
2	B ₁	A ₃	13
3	y ₁	B ₃	12
4	A ₂	y ₃	11
5	B ₁	A ₄	10
6	y ₂	B ₄	9
7	GND	y ₁	8

كل A و B تمثل ابوت وكل y هي أوتوت .

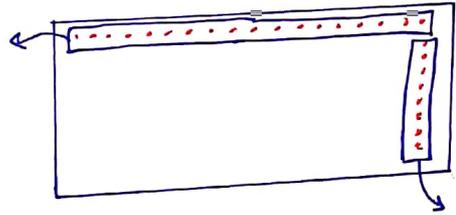
* الأوتوت حالتين يا مفر يا واحد يعني الليد
يا بصوي يا بغير ، لذلك اذا كان لون الليد

أحمر ← يعني high = 1

أخضر ← يعني low = 0

ماضوا ← short cct

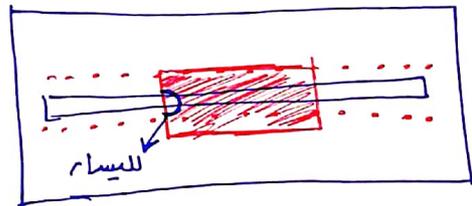
كلهم نفس النود
على الكيت .



كلهم نفس النود على الكيت.

* لا يجي تركيب and أو or أو not بتركبهم

بيت السرير على الكيت زي هيكل :-



أول اشي بالهمل بنوصل سلك مع (VCC) مع البين 14

وسلك (GND) مع البين (7).

تاني اشي بنوصل قيم الأبتوت A و B مع البينات

يلو على الرسمة ←

تالت اشي بنوصل الأوتوت y مع واحد من الليدات ،

وينحلهم بالمفاتيح بتجت A و B وبنواقيب إضاءة المصابيح.

انتهت ...

Exp(2):- multiplexers , Decoders

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

IC = 74151

X	y	z	F
0	0	0	1 = I ₀
0	0	1	1 = I ₁
0	1	0	0 = I ₂
0	1	1	0 = I ₃
1	0	0	1 = I ₄
1	0	1	0 = I ₅
1	1	0	1 = I ₆
1	1	1	0 = I ₇

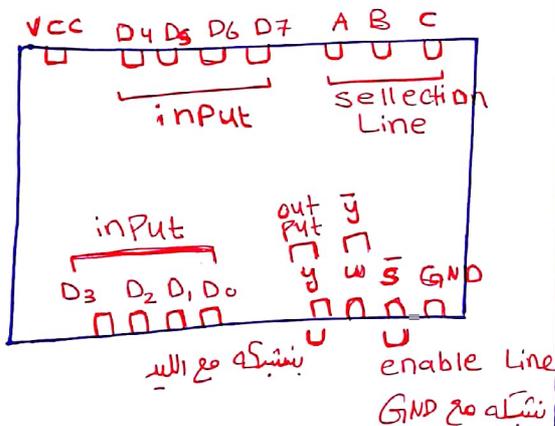
التي قيمته (1) ينشئ على (Vcc)

التي قيمته (0) ينشئ على (GND)

Selection : كل من x, y, z اسمهم اسمهم Line

لأنه حسب قيمته يتم اختيار الأوتبوت (F)

x → MSB
z → LSB



* VCC → 5V ينشئ مع

* GND → 0V ينشئ مع

* A, B, C → s, y, w على x, y, z ينشئ مع

* D₀, D₁, ... → التي قيمته واحد ينشئ على Vcc والتي قيمته صفر ينشئ على GND

* y → ينشئ على أحد الليرات

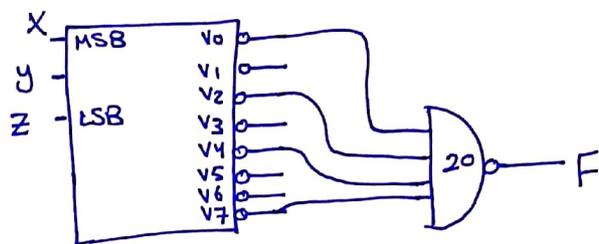
* s → ينشئ على GND

* Decoder :-

IC = 74138

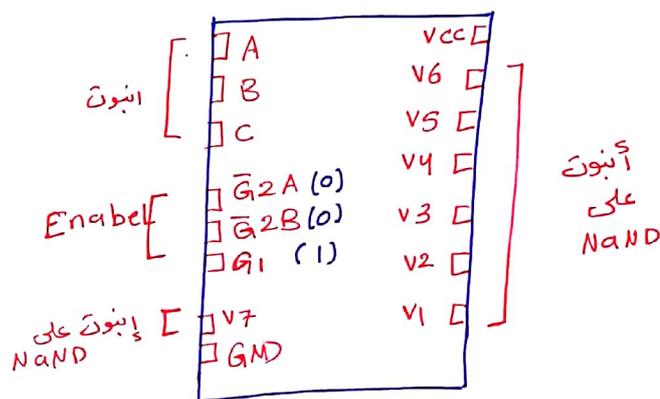
التي x, y, z تعامل معه بعدد الالان هو (Activ Low)

* $F(x, y, z) = \sum(0, 2, 4, 7)$



* خديماً x, y, z في ديكودر وفعه (Nand)

* بالنسبة للديكودور هيلك شكله :-



* A, B, C → ينشئهم على x, y, z

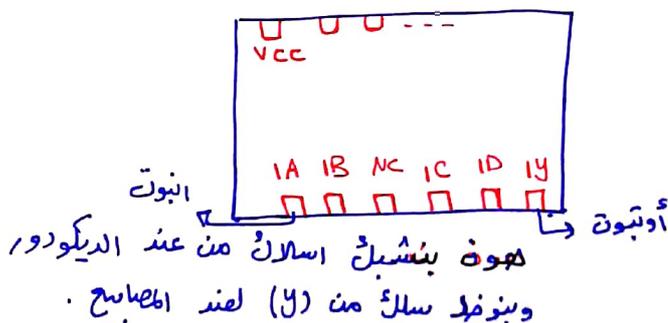
* G_{2A} → ينشئها على GND

* G_{2B} → " " "

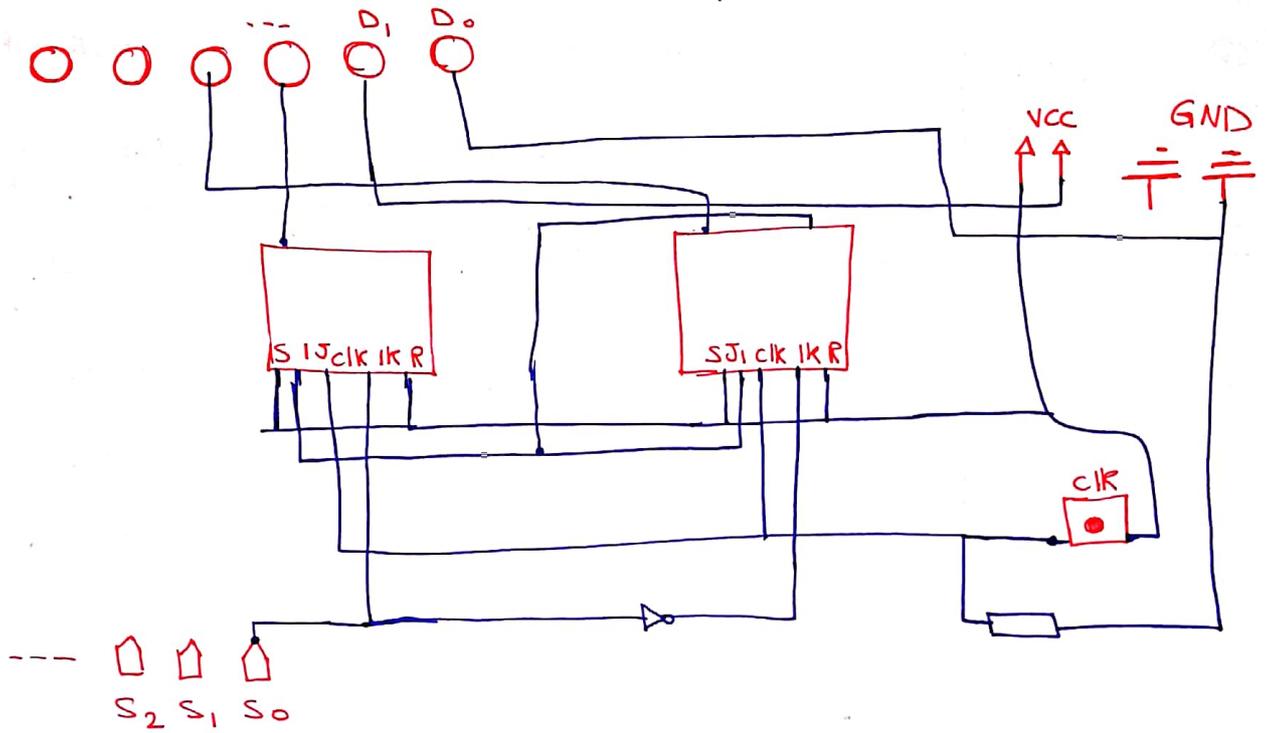
* G₁ → على Vcc

* v₀, v₁, ... → Nand على انبوت

* بالنسبة لل Nand هيلك شكلها :-

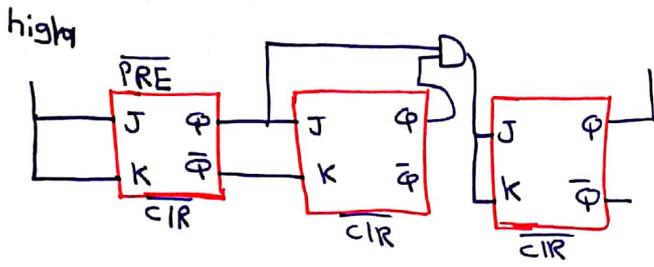


والرسمه على البروتوس هيكل يتكون :-



* Ex(6) :- counter

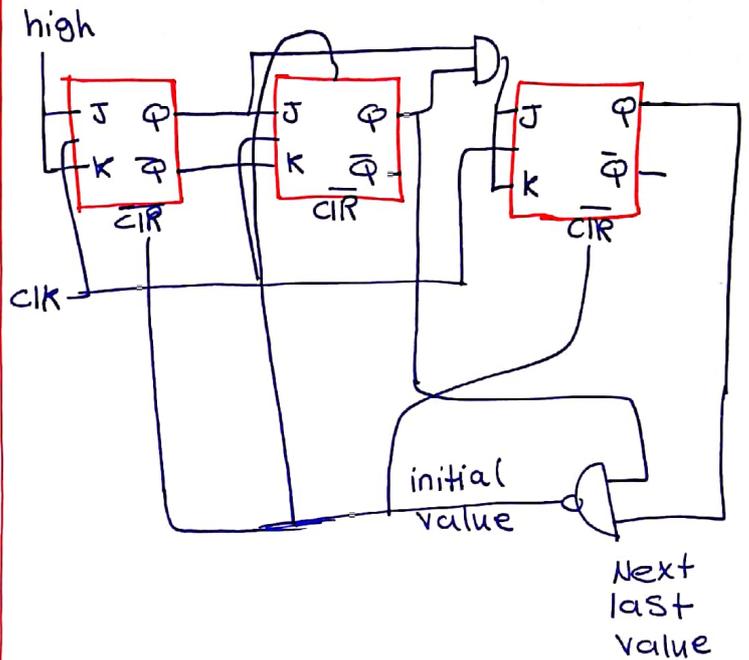
مثال :- counter (0-7)



J	K	CIR	Q	\bar{Q}
0	0	↑	Q	\bar{Q}
1	0	↑	1	0
0	1	↑	0	1
1	1	↑	\bar{Q}	Q

clock pulse	Q ₂	Q ₁	Q ₀
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

مثال :- counter (2-5)



Ic (NAND) = 74LS00

* د هيل يكون الشكل على البرنامج 8 -

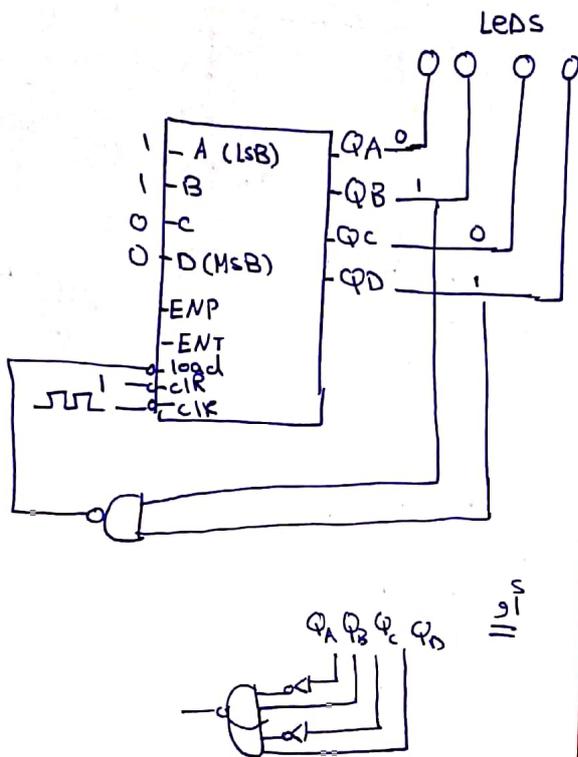
نفس الرسمة اللي فوق تماما ويخط كيسة

مع مقاومة مخرجة عنان نيلهم Push button

مع الكلوك .

Examples :-

* use 74LS161 as a counter seq (3, 4, ..., 10, 3, ...).



MSB		LSB	
D	C	B	A
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0

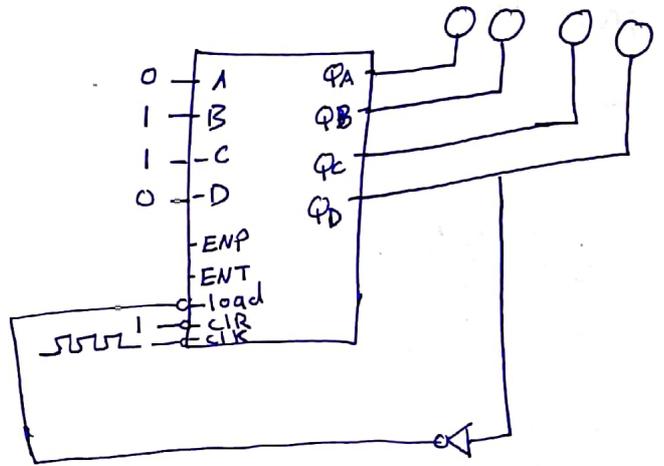
↑ ↑

unique (in this state, the counter must return back to 3).

* load is active low, so when the counter reach to (10) = 1010, it must restart from (3).

- * NAND ⇒ D with 1's input.
- OR ⇒ with 0's input.
- NOT ⇒ with 1's input.

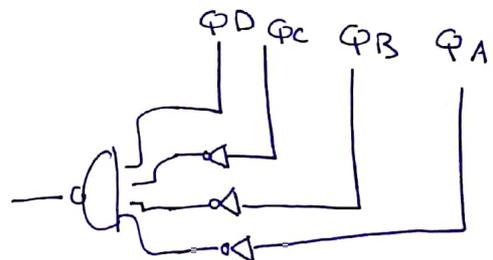
* use 74LS161 as a counter seq (4, 5, ..., 8, 4, 5, ...).



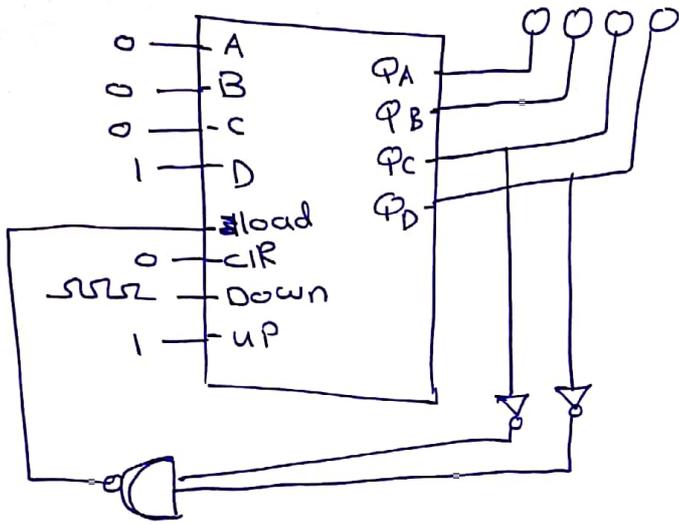
D	C	B	A
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0

↳ unique when QD = 1

طريقة ثانية لكل



* use 74LS193 counter
 seq (8, 7, ..., 4, ...)

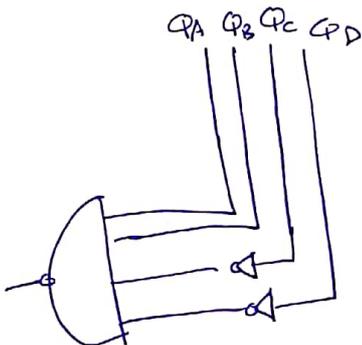


D	C	B	A
1	0	0	0
0	1	1	1
0	1	1	0
0	1	0	1
0	1	0	0
0	0	1	1

unique

test the next state.

other solution :-



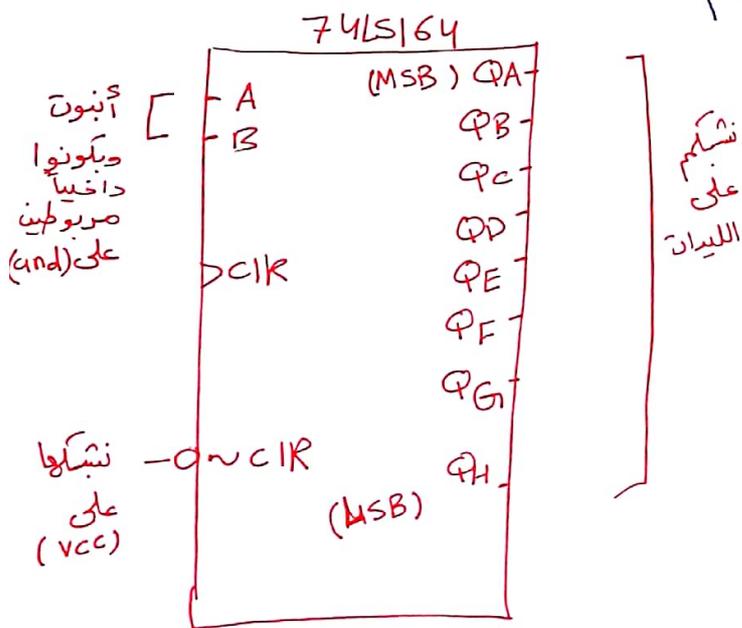
exp (7) :- Registers .

1- 674LS164 → 8-bit serial in/
Parallel out shift
reg.

2- 74LS166 → 8-bit Parallel in/
Serial out shift reg.

3- 74LS194 → 4-bit Bidirectional
universal shift reg.

1



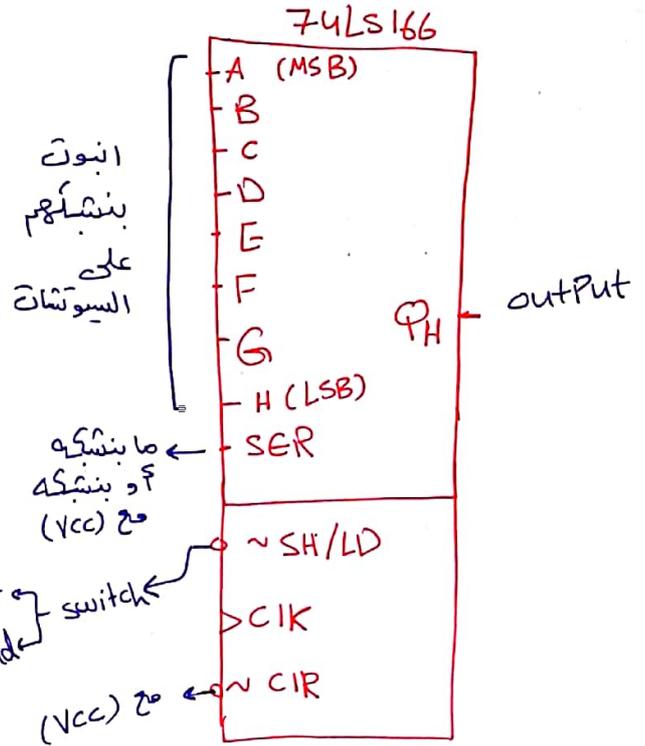
مثال :- write the value 0xA6
in the output of this
register .

$$A6 = 10100110$$

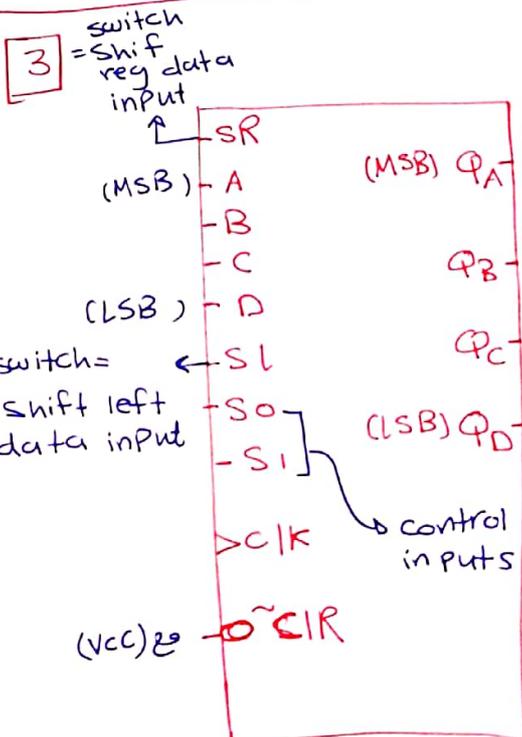
ملاحظة :- تدخيل الـ data يكون من اليسار
لليمين .

وآدول اثنى راج يفتري هو (QH) .

2



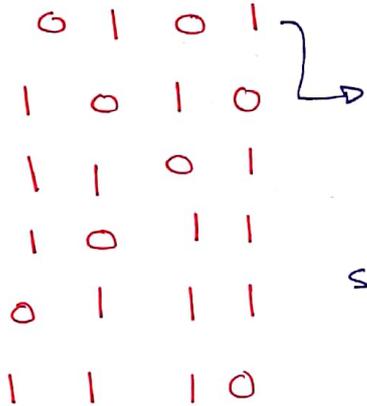
مثال :- writ the value 0xB2
in the outputs of this reg.



S ₁	S ₀	
0	0	clocking
0	1	shift right
1	0	shift left
1	1	loading



بجملتهم يعني
 S₀ = 1
 S₁ = 1
 ديفين كلوك



البيانات تمررت لليمين
 مضاعفة S₀ = 1
 S₁ = 0
 والحانة الجديدة
 بتنسلك على SR
 وديفين كلوك

Exp (8) :- ALU .

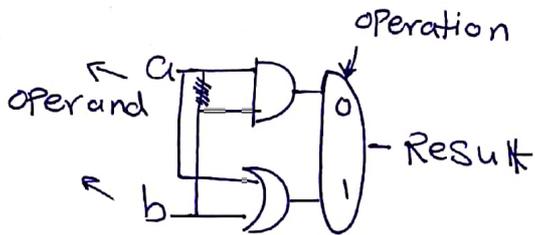
* ALU is responsible for instructions execution, the execution of an instruction is called operation.

* the operation can be either :-

- ① arithmetic operations (addition, subtraction, multiplication, division)
- ② logic operations (AND, OR, NOT, XOR).
- ③ Bit-shifting operations (shifting right or left, with or without sign extension)

* there are 2 types of signals (input to the ALU) which are the data signals and control signals.

* the 1-bit logical unit for (and) and (OR) looks like figure \rightarrow [multiplexer].



* كيف نستخدم size مع mux
 (operation) في الـ ALU

* a (32) bit ALU can be implemented by combining 32 of the above (1) bit (ALU) sequentially.

