NO. CH 3

* Design

1 - ترجمة السؤال

2- Truth table دعمل

3- K-map بعمل

4- Logic diagram ارسم

* Mapping to NoR, NAND

NoR

AND → OR + Inputs, outputs بتكس

OR → NoR + Inverter

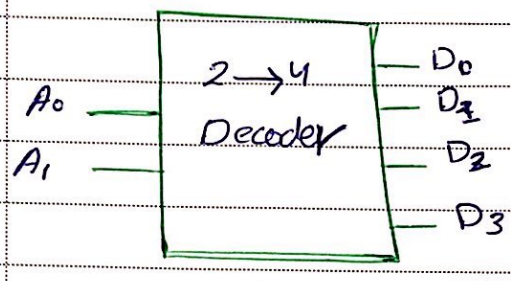
NAND

AND → NAND + Inverter

OR → AND + Input, outputs بزيادة

END of part 1

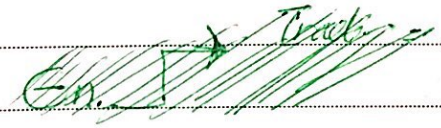
1) Decoder



n: Inputs
2^n: outputs

* Decoder is a minterm generator
* يعطي 1 على الـ Line اللي رقمه نفس رقم الـ Input
* (1 → 2) decoder : عبارة عن سلك Inverter +

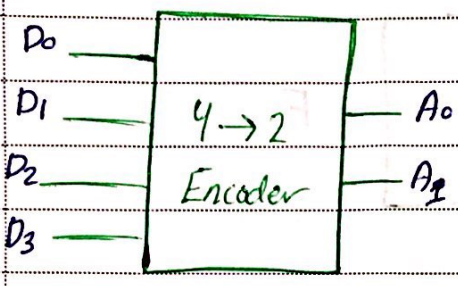
Decoder expansion: 1 → no. of (1 → 2) line decoders
n → no. of 2-input AND gates $= 2^n$



Gn → Traditional ⇒ ((no. AND * No. Inputs) + No. Inverters)
Gn → Decoder exp. ⇒ ((No. AND * 2) + No. (1 → 2) line decoders)

NO.

2) Encoder



2^n : Inputs

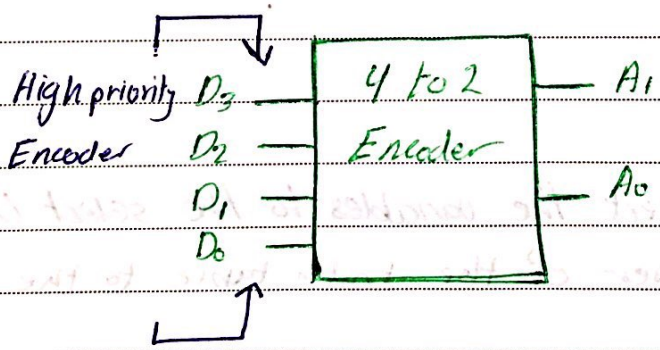
n: outputs

* Opposite of Decoder.

* Input على واحد (1) يكون من

* High priority Encoder ← بياخد من الـ MSD أول 1 باخده وغيره (X).

* Low priority Encoder ← = = الـ LSD أول 1 باخده وغيره (X)

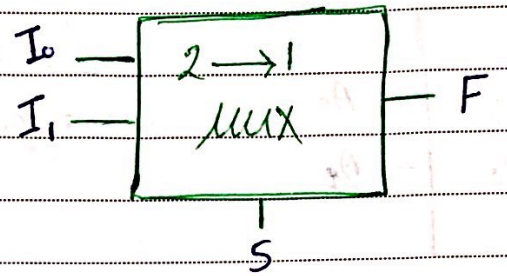


Low priority Encoder.

NO.

3) Multiplexer

2^n: Input
n: select lines.
one output



* To Build $2^n \rightarrow 1$ MUX, you need:

1) $n \rightarrow 2^n$ decoder
2) 2^n 2-input AND gates.
3) one 2^n- input OR gate

* Function Implementation using muxes:

~~Approach~~

Approach 1: connect the variables to the select lines and copy the answer of the truth table to the inputs.

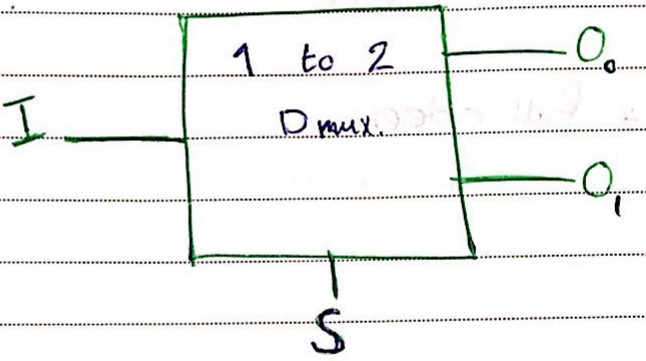
Approach 2: 1) use $2^{(n-1)} \rightarrow 1$ mux where n: is the No. of variables
2) find the truth table
3) Seperate the truth table into pairs without the least sig. digit.
4) make a relation between the truth table & the LSD and copy it to the inputs.

NO.

4) Dmux.

* Opposite of mux.

* 1 input.
  n select lines.
  2^n outputs.



Dmux ≡ Decoder with enable

* إذا كان S=0 بيعبّي قيمة الـ I على Line 0 , والباقي Zeros.

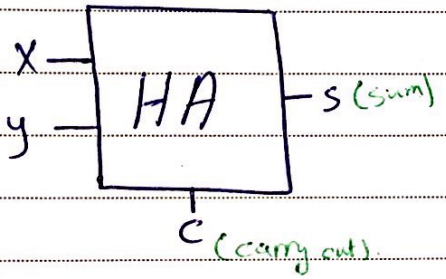
NO. CH 4

* Half adder : 2-bits adder

* Full adder : 3 bits adder

* Ripple carry adder : a group of bits adder

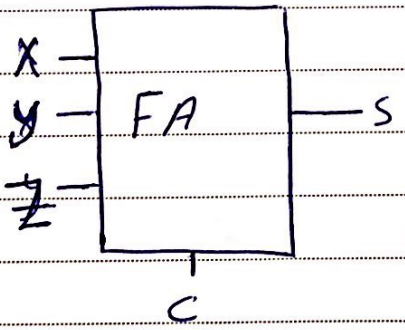
Half adder



$S = X \oplus y$

$C = Xy$

Full adder



$S = X \oplus y \oplus Z$

$C = Xy + XZ + yZ \equiv Xy + (X \oplus y)Z$

NO.

Sub.

$M-N \rightarrow B=0 \rightarrow M>N$

$M-N \rightarrow B=1 \rightarrow M<N$

The right answer is to subtract

the answer from 2^n and put the negative sign.

where n: is th No. of bits.

Complements

1) one's complement $= (r^n-1)$'s complement.

* 1's complement
* 7's complement.
* 15's complemt
* 9's complemat.

$(r^n-1) = (r-1)(r-1)(r-1)\ldots$

حسب عدد الـ bits (n).

* find 1's complemat for N? $\Rightarrow (r^n-1)-N$.

بعملك الرقم وبعكس كل اشي فيه.

NO.

2) r's complement.

* 2's complement
* 10's complement
* 16's complement
* 8's complement

Find r's complement for N? → $(r^n - N)$

* بعمل بالرقم كل (0) باضده زي ماهو لما أشوف (1) باضده ويعكس كل اللي بعده.

* 2's complement ⇒ سالب الرقم

---

sub. with 2's complement.

$M - N \Rightarrow M + (-N) \equiv \boxed{M + (2^n - N)}$

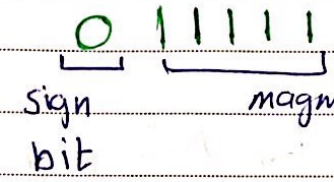
$B = 0,\ C = 1,\ M \geq N$

$B = 1,\ C = 0,\ M < N$

NO.

* Signed Integers.

* Sign Magnitude :-



* sign bit (0) ⇒ positive number
" " (1) ⇒ negative number

* Signed 1's complement

← الأرقام الموجبة زي الـ sign magnitude

← السالبة باخد 1's comp للرقم عدا الـ sign-bit =

* إذا بدي أعرف ش قيمة الرقم
بشوف الـ sign bit بحدد باخد
2's comp. للرقم

* Signed 2's complement

← الأرقام الموجبة زي الـ sign-magnitude

← السالبة باخد 2's comp. للرقم عدا sign-bit =

* Over flow

* Unsigned numbers

- addition:

  Cout = 1 ⇒ overflow ✓

- subtraction:

  No way for overflow.

* Signed numbers

- addition:

Two numbers with the same sign (إهمال!)

نهمل ال carryout

$+ \oplus + = -$

$- \oplus - = +$

overflow ✓

$$OV = C_{n-1} \oplus C_n$$

- subtraction:

Two numbers with different signs.

NO.

* Multiplication & Division

$X * 2^n$ ⇒ shift left by n-positions.

$X / 2^n$ ⇒ Shift Right by n-positions.

* Zero fill: Two different numbers with different width.

⟹ fill Zeros from the left.
for (unsigned).

* Sign Extension. for (signed).

copy the sign bit to the left

ex: 1111001 ⇒ -7
extend to 10 bits
1111111001 ⇒ still -7

1101 ⇒ -3
extend to 8 bits
11111101 ⇒ -3