2024/08/26

pMOS cannot pass logic 0 properly
nMOS cannot pass logic 1 properly
→ which is why we use NAND and NOR gates

All Boolean expressions can be implemented using NAND / NOR / NOT gates. (May not be optimised)

F(A, B, C) → expression
F(A, B, C) → truth table

→ circuit
SoP realisation
PoS realisation

SoP → minterms

$m_0 \leftarrow (\bar{A} \cdot \bar{B} \cdot C) + \cdots$

To get 1 in the o/p → only 1 of the terms need to be 1

PoS → maxterms

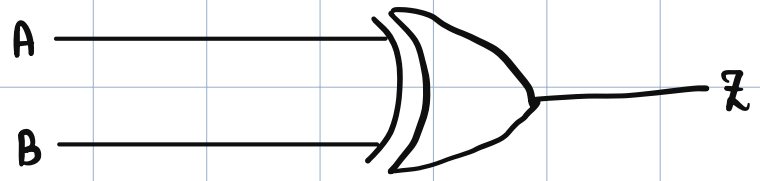
$(A + B + \bar{C}) \cdot (\cdots) \cdot (\cdots)$ → M_0

all terms need to be 1

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

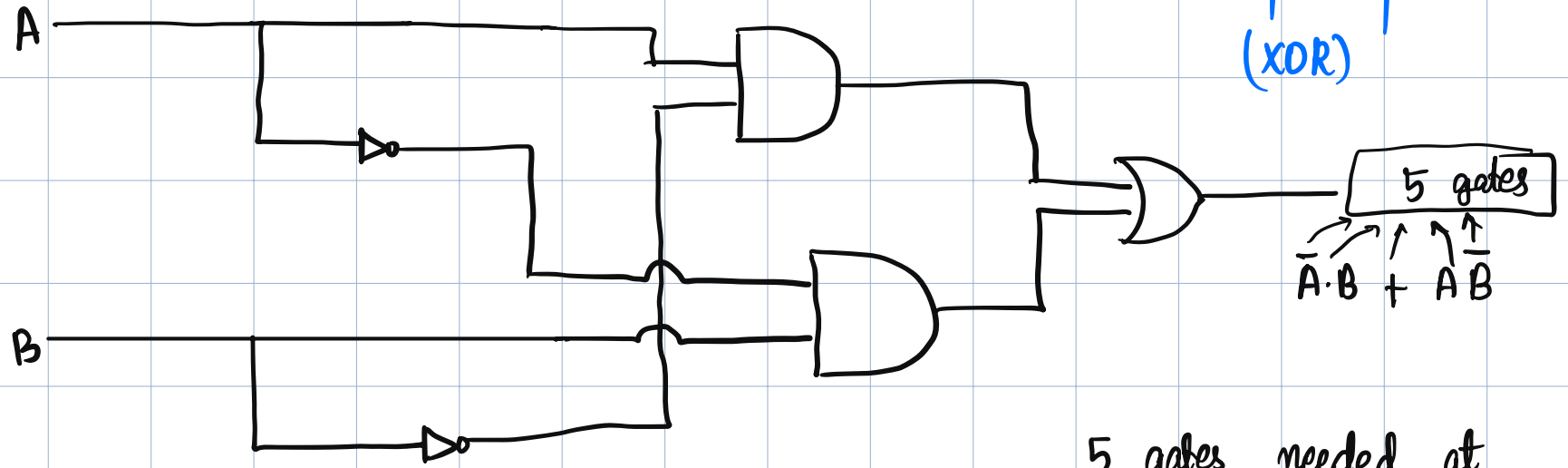
$$= (A + B) \cdot (\bar{A} + \bar{B})$$

$$= A \oplus B$$



	A	B	Z	
m_0	0	0	0	M_0
m_1	0	1	1	M_1
m_2	1	0	1	M_2
m_3	1	1	0	M_3

NOT - equal operator (XOR)



5 gates needed at minimum

$$Z = \bar{A}B + A\bar{B}$$

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

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

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

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

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

$$= A \cdot \overline{A \cdot B}$$

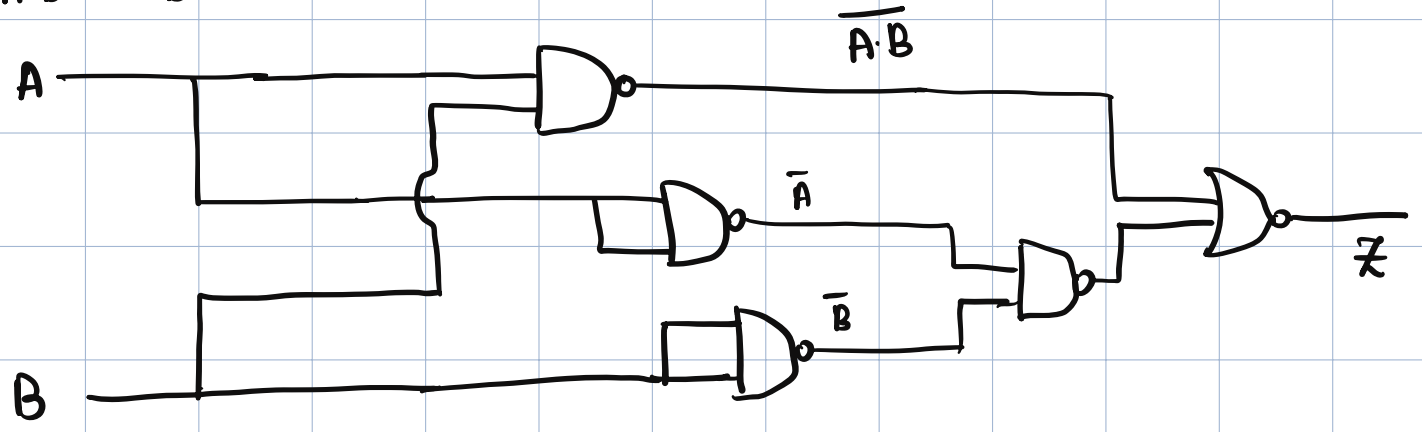
$$\overline{A+B} = \bar{A} \cdot \bar{B}$$

$$Z = (A+B) \cdot (\overline{A \cdot B})$$

$$\overline{\bar{A} \cdot \bar{B}} \cdot \overline{A \cdot B}$$

Implement

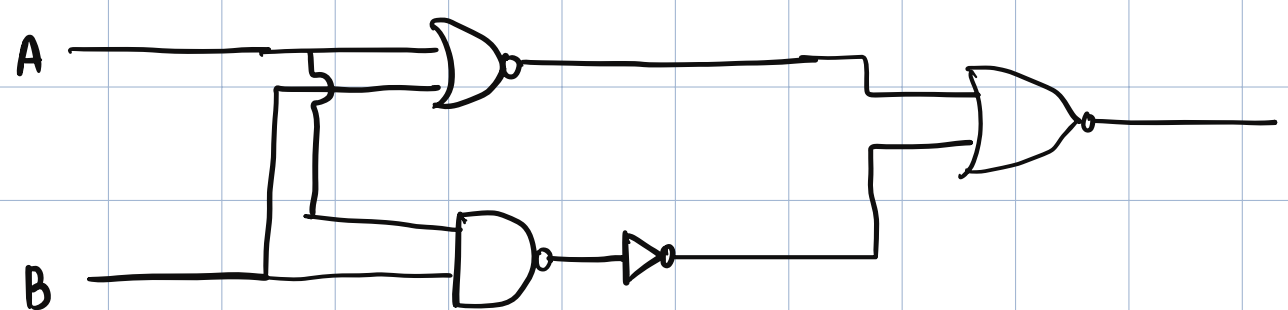
NAND
NOR



What is the minimum number of gates required to realise a given boolean expression?

$$(A+B)\cdot(\overline{A\cdot B})$$

$$\overline{\left(\overline{A+B} + \overline{\overline{A\cdot B}}\right)}$$



Trick : Minimum number of NAND or NOR gates required

	NAND	NOR
NAND	1	4
NOR	4	1
AND	2	3
OR	3	2
XOR	4	5
XNOR	5	4

Simple :

	NAND	NOR
$Y = ABCD \ldots$	$2n - 2 + k$	$3n - 3 - k$
$Y = A + B + \ldots$	$3n - 3 - k$	$2n - 2 + k$

Compound

(i) $A + \bar{B}C + AC$

$= A + \bar{B}C$

$= A \cdot A + \bar{B}C \longrightarrow ③ + ①$

(ii) $A\bar{B} + \bar{C}D \longrightarrow ③ + ② = ⑤$

(iii) $A\bar{B} + BC$

$③ + ① \rightarrow ④$

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

$XC + XD$

$③ + ①$

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

$A + B = x \rightarrow 3 \times 2 - 3 - 0 = (3)$

$xC + x\bar{D} \longrightarrow (3) + (1)$

(7)

NOR

$(A + B) \cdot (C + D) \rightarrow (3)$

$(A + \bar{B}) \cdot (C + \bar{D}) \rightarrow (3) + (2)$

$\bar{A} + \bar{B}$

$A\bar{B} + \bar{A}B$

$(3) + (2)$

$$A \oplus B = \bar{A} \cdot B + A \cdot \bar{B}$$

$$B \cdot \bar{B} + A \cdot \bar{B}$$

$$\bar{A} \cdot (A + B) \qquad = \bar{B}\,(A + B)$$

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

$$= \overline{A \cdot B} \cdot (A + B)$$

pMOS device 3 times bigger than nMOS

NAND

* NAND gates are preferred for CMOS circuits

2024|08|28

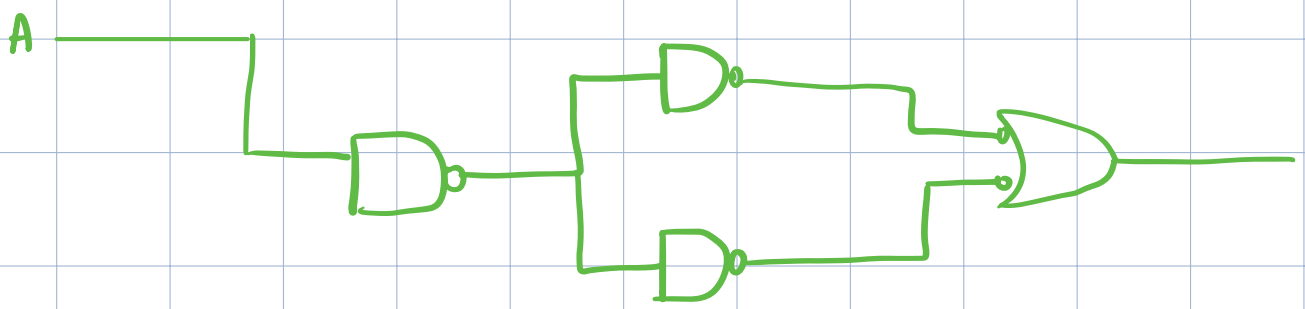
$A \oplus B = \bar{A}B + A\bar{B}$

$B \cdot (\overline{A \cdot B})$

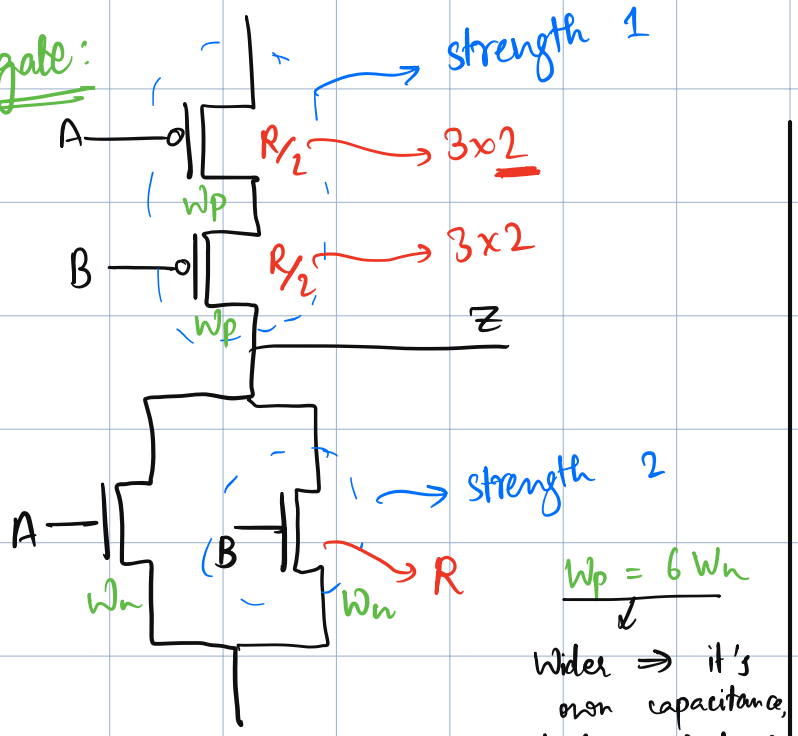
$A\bar{A} + A\bar{B}$

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

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



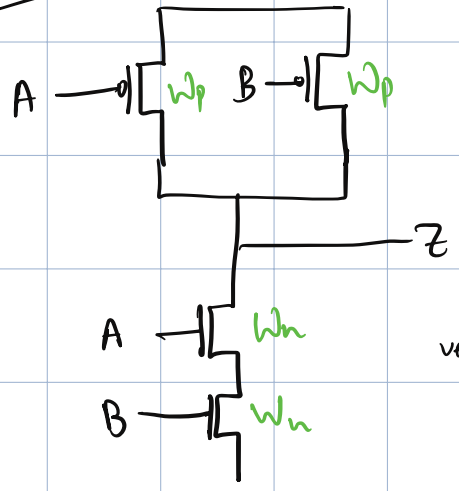
NOR gate:



$W_p = 6\,W_n$

Wider ⇒ it's more capacitance, discharge ⇒ slower

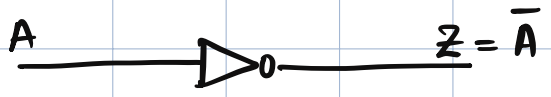
NAND gate:



$W_p = 1.5\,W_n$

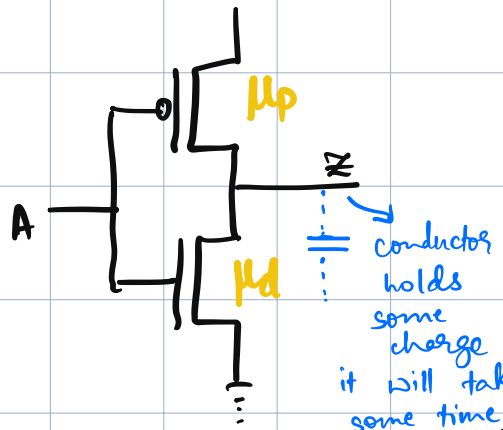
faster and smaller

very specific to CMOS



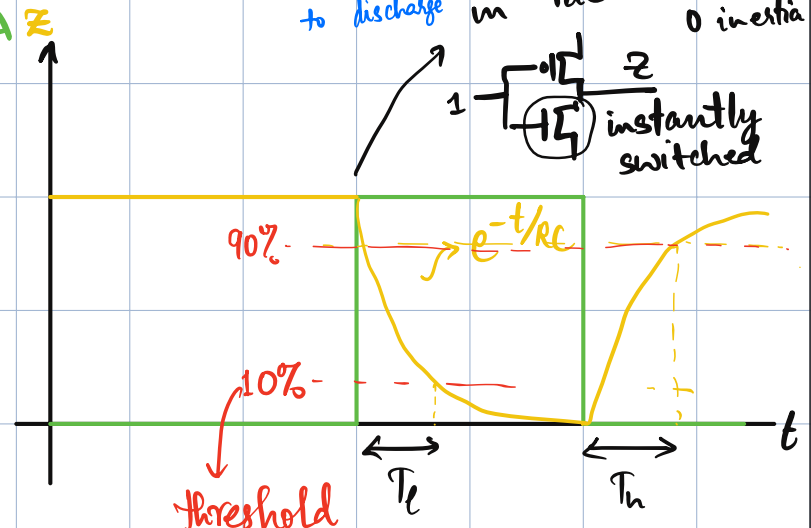
In steady state

$A \rightarrow \bar{A}$

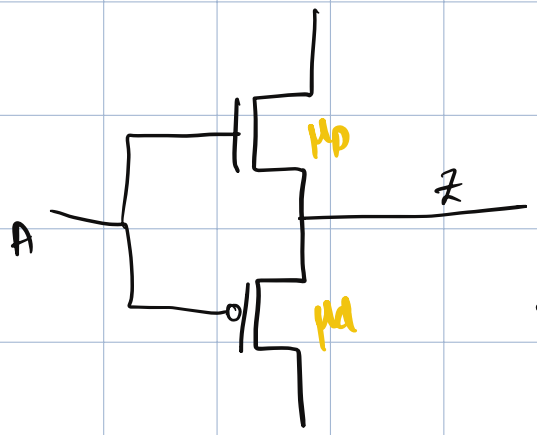


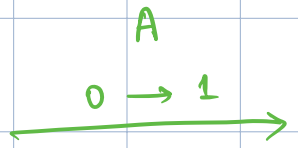
steady state – not always useful to study

in ideal scenario 0 inertia



T_f and T_r may not be same

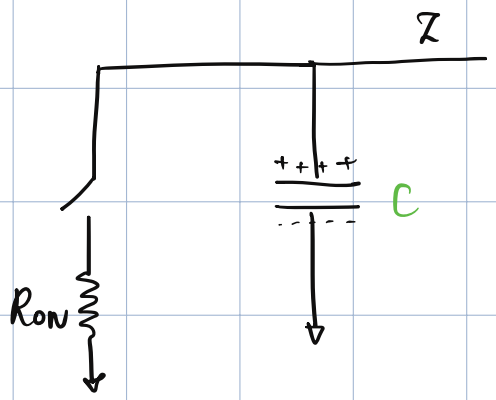




In steady state

- conductor discharged
- no energy stored

$= Q = 0$



RC - circuit

$$\frac{dQ}{dt} = I$$

$$\Rightarrow I = C \frac{dV}{dt}$$

$V \propto e^{-t/RC = \tau}$

delays T_H and T_e are functions of RC

to reduce delays ⟶ RC values must be low.

put two MOSFETs in parallel

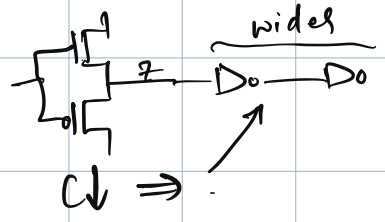
nMOS
pMOS

⟶ length ↓ ⇒ R ↓
width ↑ ⇒ R ↓

~ resistance of conductor

T_e and T_h cannot be zero.

Why should the next switch be wider than the prev?



* nMOS and pMOS → materials different.

~~avg~~ drift velocity → depends on electric field

E↑ v_d↑

↓

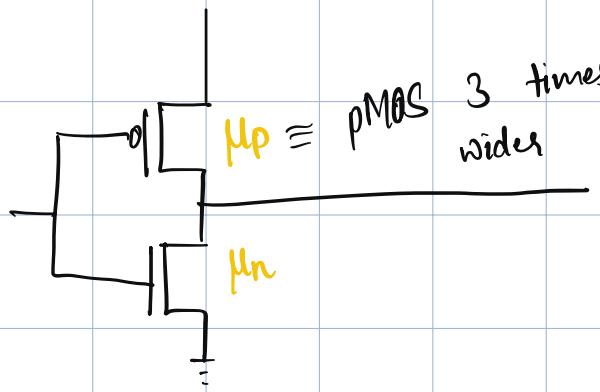
determined by a parameter

called mobility

Mobility ↑ v_d ↑

speed of ~~electricity~~

mobility



$3\mu_p = \mu_n$

↓

to keep delays roughly equal

1. **Mobility of Charge Carriers:** The mobility of holes (the charge carriers in PMOS transistors) is actually about 2-3 times lower than that of electrons (the charge carriers in NMOS transistors). Lower mobility means that for the same electric field, holes move more slowly than electrons.
2. **Conductivity and Sizing:** Because the PMOS transistors have lower mobility, they naturally have lower conductivity compared to NMOS transistors when they are the same size. To compensate for this and ensure that the inverter switches symmetrically (i.e., with equal rise and fall times), the PMOS transistor is made wider. By increasing the width of the PMOS transistor, its drive strength (current-carrying ability) is increased, which helps to balance the switching characteristics of the inverter.
3. **Width Ratio:** The common design practice is to make the width of the PMOS transistor about 2-3 times that of the NMOS transistor. This compensates for the lower mobility of the holes and ensures that the inverter has similar pull-up and pull-down strengths, leading to balanced delays.

Ex:

$$F(A,B) = \text{sum of minterms}$$

$$= \sum (m_0, m_1, m_2, m_3)$$

$$= 1$$

systematic approach but not always the best

no dependence on input

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	1

$$F(A,B) = \sum (m_0, m_2, m_3)$$

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

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

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

irrespective of what B is, when A is 1, Z=1

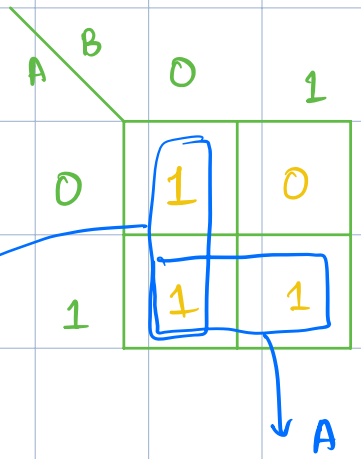
A	B	Z
0	0	1
0	1	0
1	0	1
1	1	1

when A is 0, $Z = \bar{B}$

as long as only 1 changes

More systematic than finding SoP

– always optimized



grouping

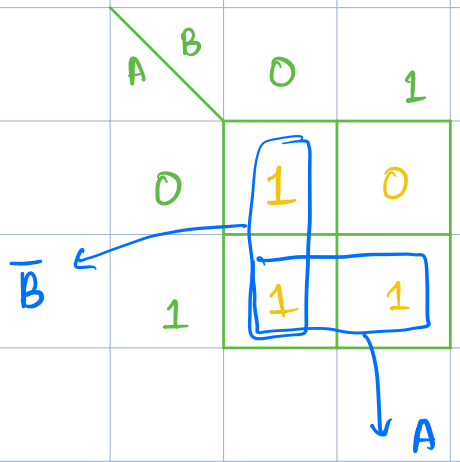
Combining theorem in a graphical manner

2024/08/29

Minimization of Boolean logic:-

A	B	Z
0	0	1
0	1	0
1	0	1
1	1	1

* Using k-Map
  * Minterms are drawn or filled in a 2-D table
  * Only one literal changes when moving horizontally / vertically



For n-bits, no. of functions = 2^{2^n}

2^n rows → each can be 0 or 1

2 Variable Boolean function

A	B	Z
0	0	
0	1	
1	0	
1	1	

$Z = 0$, $Z = 1$

$Z = A$, $Z = \bar{A}$

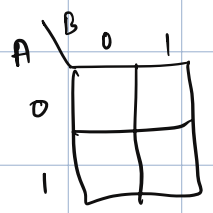
$Z = B$, $Z = \bar{B}$

$Z = A \oplus B$, $Z = \overline{A \oplus B}$

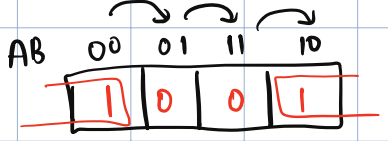
XNOR → equivalence operator

Three input function minimization

Ex:



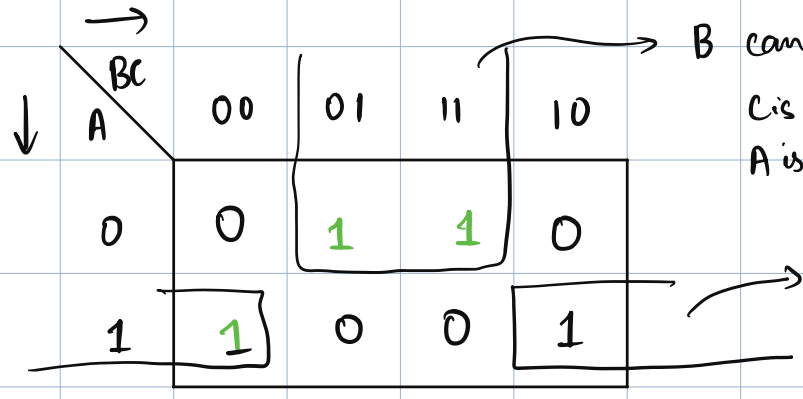
only 1 literal should change in adjacent cell



$\equiv$ circular connection

$\bar{B}$

- grouping of first and last cells allowed



B can be removed
C is not changing
A is not changing

A is not changing
C is not changing
B can be removed

$$\bar{A} \cdot C + \bar{C} \cdot A$$

[Prime implicant] [Prime implicant]

A	B	C	Z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Literal

expression with variables as it is or as complement

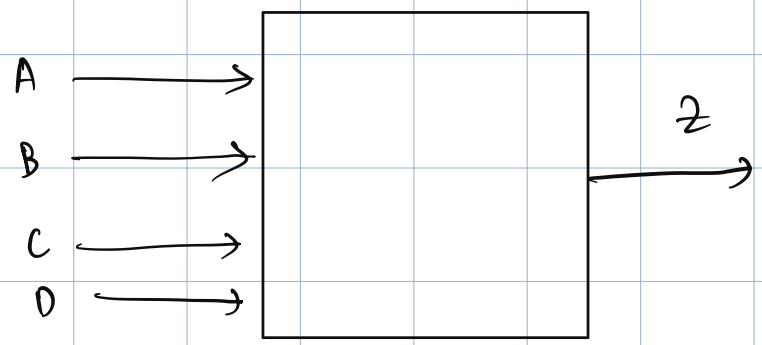
count no. of literals to get the complexity in logic

"Essential prime implicant"

Design a digital logic which can identify if an integer 0 – 9 is a prime numbers. (no calculation here)

① Convert to a boolean form

4 inputs are needed



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

don't care

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0	0	1	1
$\bar{A}B$ 01	0	1	1	0
11	X	X	X	X
10	0	0	X	X

→ 3 prime implicants

→ assume 0

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