ECN-104

Digital Logic Design

04-03-23

→ Morris. Mano

LOGIC GATES

- **AND:**  AB
- **OR:**  $A+B$
- **NOT:**  $\bar{A}$
- **NOR:**  $\overline{A+B}$
- **NAND:**  $\overline{A\cdot B}$

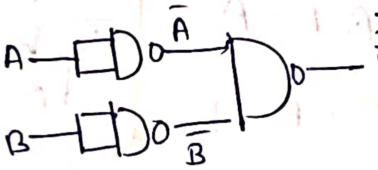
$$\boxed{\begin{aligned}\overline{A\cdot B} &= \bar{A}+\bar{B}\\ \overline{A+B} &= \bar{A}\cdot\bar{B}\end{aligned}}$$

- **XOR:** $\boxed{\bar{A}\cdot B+\bar{B}\cdot A}$

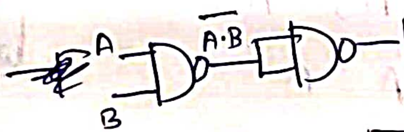
- **Universal Gates:** NAND and NOR are known as Universal gates because they can be used to obtain AND, OR, NOT gates.

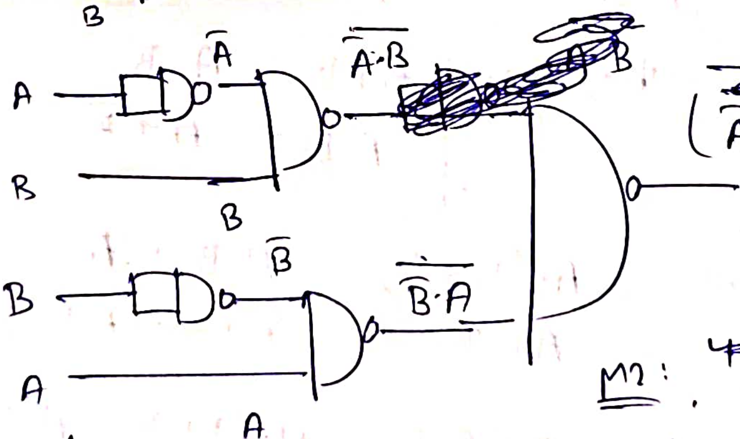
▲ **Using NAND gate:**

① **NOT:**  $\overline{A\cdot A} = \bar{A}$

② **OR:**  $\overline{\bar{A}\cdot\bar{B}} = A+B$

4 NAND → XOR

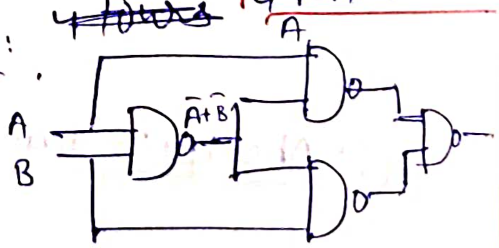
③ **AND:**  $A\cdot B$

④ **XOR:** 

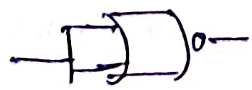
$\overline{(\overline{\bar{A}\cdot B})\cdot(\overline{\bar{B}\cdot A})}$

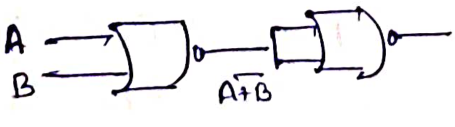
$= \bar{A}\cdot B + \bar{B}\cdot A$

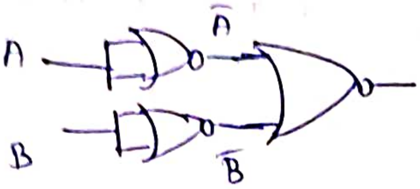
~~4 NAND~~ 4 NAND

M2: 

Using NOR gate:

① **NOT:**  $\overline{A+A} = \bar{A}$

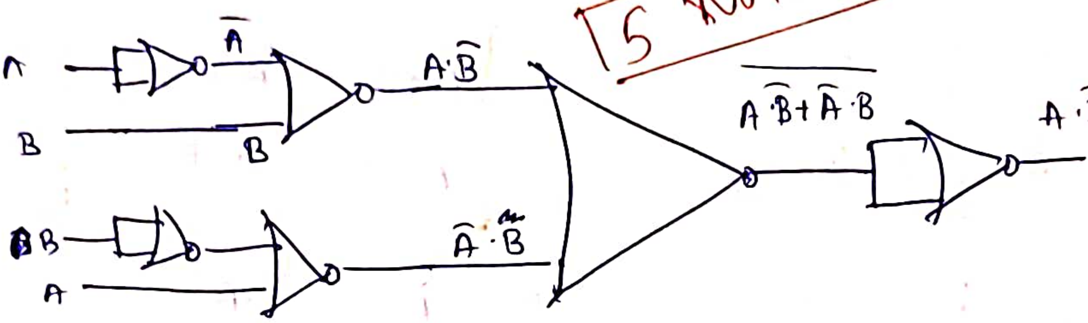
② **OR:**  $A+B$

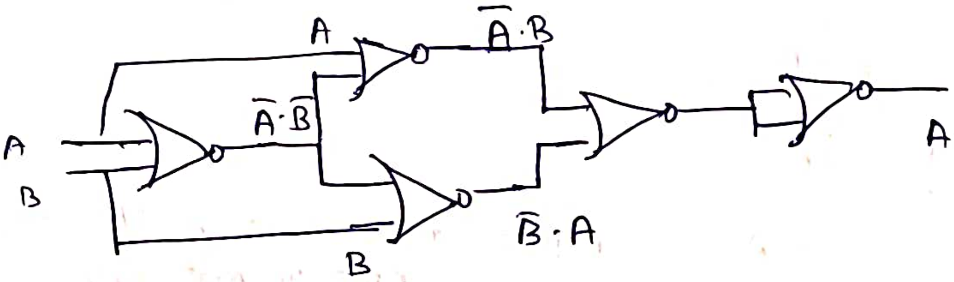
③ AND: 

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

④ XOR:

$\boxed{5 \text{ XNOR} \rightarrow \text{XOR}}$

M1: 

M2: 

- Laws of Boolean Algebra:

① Annulment law: $A \cdot 0 = 0$, $A+1 = 1$, ~~$A+0=A$, $A \cdot 1=A$~~

② Identity law: $A+0 = A$, $A \cdot 1 = A$

③ Idempotent law: $A+A = A$, $A \cdot A = A$.

④ Complement law: $A \cdot \bar{A} = 0$, $A + \bar{A} = 1$

⑤ Commutative law: $A \cdot B = B \cdot A$, $A + B = B + A$

⑥ Double negation law: $\bar{\bar{A}} = A$.

⑦ De-Morgan's Theorem:
$$\boxed{\begin{aligned} \overline{A+B} &= \bar{A} \cdot \bar{B} \\ \overline{A \cdot B} &= \bar{A} + \bar{B} \end{aligned}}$$

- Duality Principle: Any algebraic equality derived ~~from~~ using laws of Boolean Algebra will still be valid if all $\cdot \rightarrow +$, $+ \rightarrow \cdot$ and $0 \rightarrow 1$, $1 \rightarrow 0$.

⑧ Distributive: $A \cdot (B+C) = (A \cdot B) + (A \cdot C)$ (OR law)
$A + (B \cdot C) = (A+B) \cdot (A+C)$ (AND law)

⑨ Absorptive law: $A + A \cdot B = A \cdot (1+B) = A \cdot 1 = A$ (OR law)
$A.(A+B) = A \cdot A + A \cdot B = 1A + A \cdot B = A$
$\hookrightarrow = (A+0) \cdot (A+B) = A$ (AND Law)

⑩ Associative: $A + (B+C) = (A+B) + C$ (OR Law)
$A(B \cdot C) = (A \cdot B) C$ (AND Law)

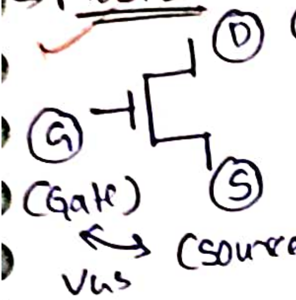
07/03/23

Transistors

MOSFETs (Uni-polar) — PMOS (Holes), NMOS (electrons)

BJTs (Bi-polar Junction Transistors) (NPN/PNP)

→ **MOSFETS:** Metal-Oxide Semi-cond. Field-Effect-Transistor



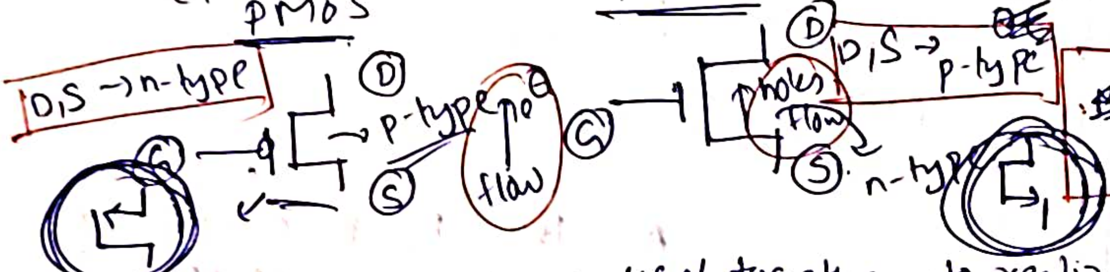
- $V_{GS} \geq V_{threshold}$ ⇒ Connection is made and current flows

$V_{GS} < V_{threshold}$ ⇒ Connection is not made.

(p-channel-MOS) PMOS

(n-channel MOS) NMOS

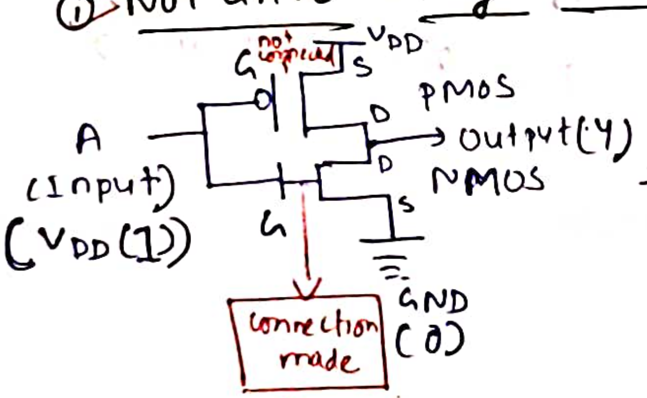
~~Voltage NMOS PMOS~~



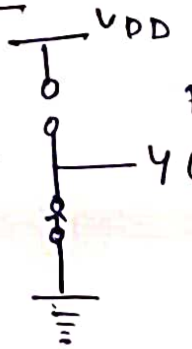
VDD (1) → High
GND (0) → Ground (Low voltage)

- When PMOS & NMOS are used together to realize LOGIC GATES, it is called CMOS (Complementary - Metal Oxide. Semi-conductor)

① NOT GATE using MOSFET:

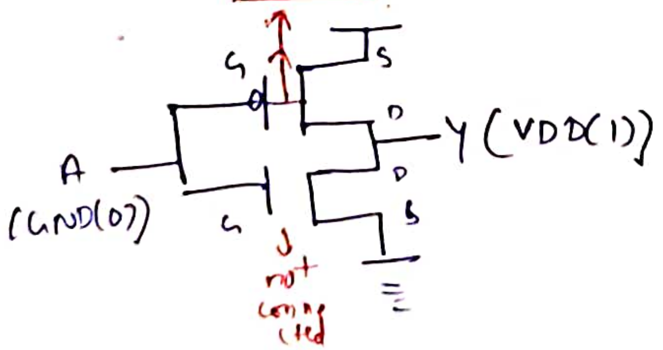


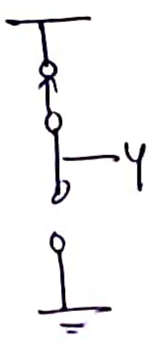
≡



$V_{GSn} = V_{DD}$
$V_{GSp} = 0$

[∵ Connected to Ground]



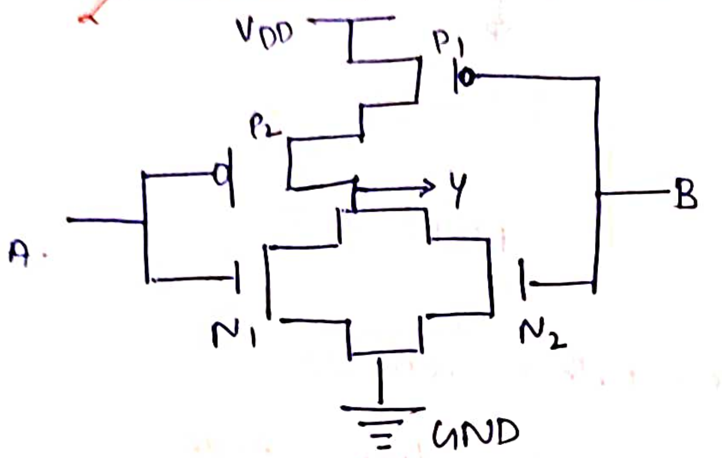


Truth Table

A	Y
1	0
0	1

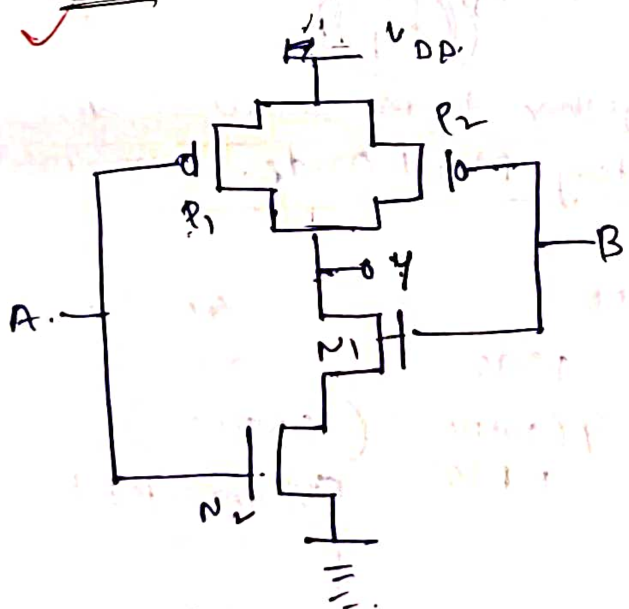
$V_{GSn} = 0$
$V_{GSp} = -V_{DD}$

② NOR Gate using MOSFET/CMOS.



A	B	P_1	P_2	N_1	N_2	Y
0	0	ON	ON	OFF	OFF	1
1	0	ON	OFF	ON	OFF	0
0	1	OFF	ON	OFF	ON	0
1	1	OFF	OFF	ON	ON	0

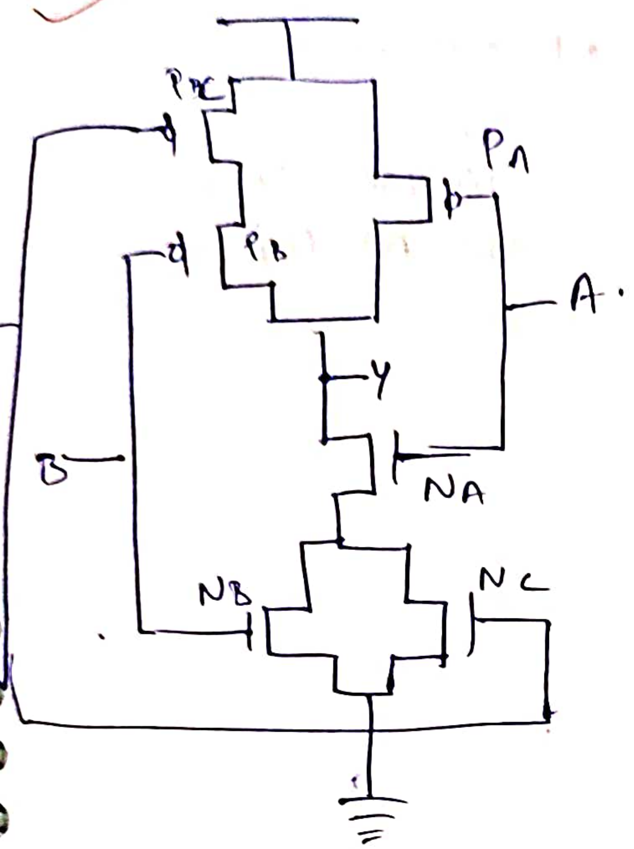
③ NAND Gate using CMOS:



A	B	P_1	P_2	N_1	N_2	Y
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	ON	OFF	1
1	0	OFF	ON	OFF	ON	1
1	1	OFF	OFF	ON	ON	0

• ~~$Y = A\cdot(B+C)$~~

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



For boolean exp under [Combination of · & +]

Ex: $\overline{A \cdot (B + C) \cdot D \cdot E \cdot (F + G + H)}$

(i) [· (AND)] → Series

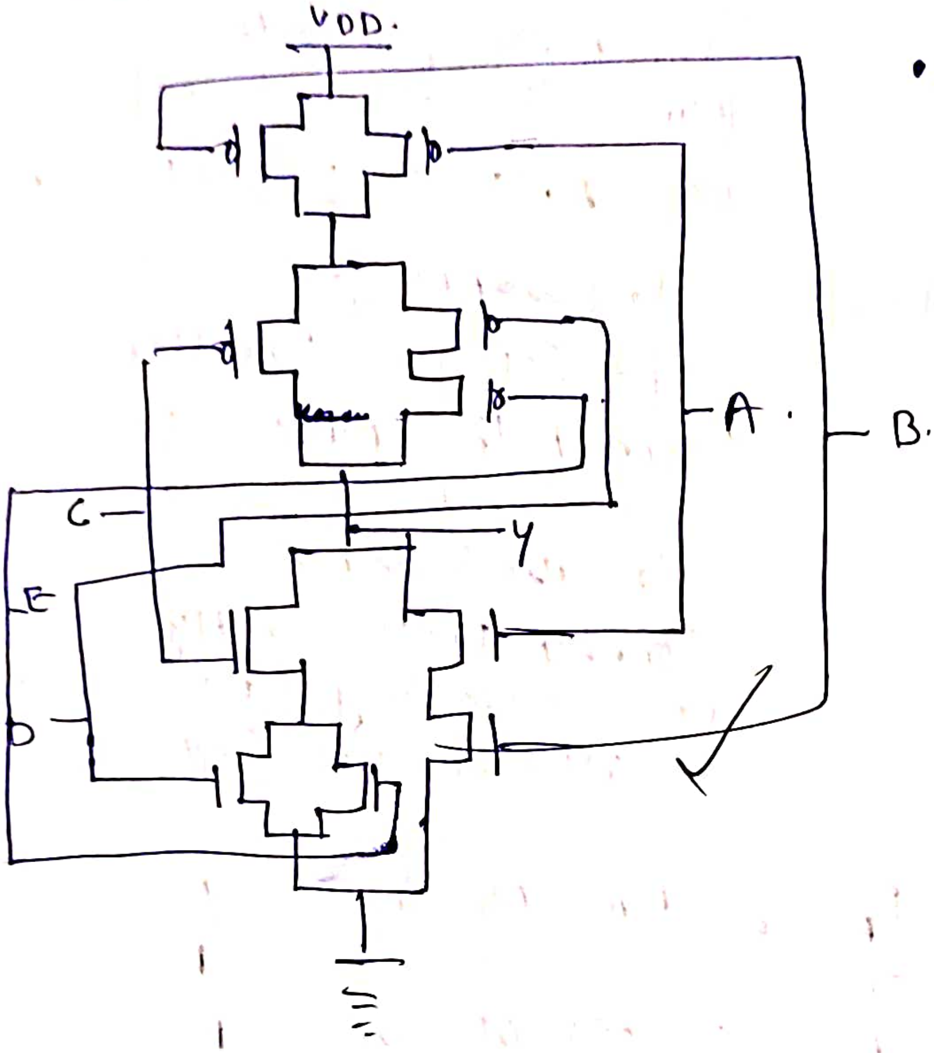
(ii) [+ (OR)] → Parallel

connect the NMOS in series and parallel acc: to rules (i) and (ii).

Connect PMOS exactly in reverse way (i.e parallel in place of series) (and vice-versa)

A	B	C	P_A	P_B	P_C	N_A	N_B	N_C	Y
0	0	0	ON	ON	ON	OFF	OFF	OFF	1
0	0	1	ON	ON	OFF	OFF	OFF	ON	1
0	1	0	ON	OFF	ON	OFF	ON	OFF	1
0	1	1	ON	OFF	OFF	OFF	ON	ON	1
1	0	0	OFF	ON	ON	ON	OFF	OFF	1
1	0	1	OFF	ON	OFF	ON	OFF	ON	0
1	1	0	OFF	OFF	ON	ON	ON	OFF	0
1	1	1	OFF	OFF	OFF	ON	ON	ON	0

• $Y = \overline{A \cdot B + C \cdot (D+E)}$



• Minimum
5×2
$= 10$ transistors
are needed.

11/03/22:

→ VHDL:

VHSIC – Very High Speed Integrated Circuit
H → Hardware
D → Description
L → Language.

• Specification, Simulation, Synthesis

13/03/23

- **POS:** Product of Sums Ex: $\overset{\uparrow}{(A+\bar{B})}\,\overset{\uparrow}{(C+\bar{D}+E)}\,\overset{\uparrow}{(G+\bar{H})}$. ← Max-terms
- **SOP:** Sum of Products. Ex: $\underset{\downarrow}{A\bar{B}} + \underset{\downarrow}{CD\bar{E}} + \underset{\downarrow}{\bar{F}G}$ → Min-Terms

Canonical Forms:

- **Min-terms:** (Standard Product) (for 3 variable case)

x	y	z	min term	
0	0	0	$x'y'z'$	(m_0)
0	0	1	$x'y'z$	(m_1)
0	1	0	$x'yz'$	(m_2)
0	1	1	$x'yz$	(m_3)
1	0	0	$xy'z'$	(m_4)
1	0	1	$xy'z$	(m_5)
1	1	0	xyz'	(m_6)
1	1	1	xyz	(m_7)

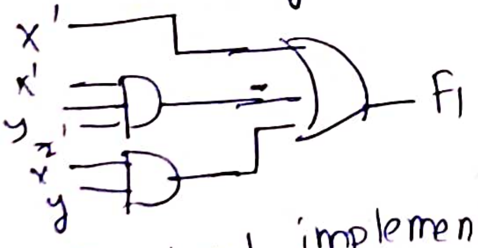
(m_7) ↓ General Notation.

- **Max-Terms:** (Standard Sum) (for 3 variable case)

x	y	z	maxterm	
0	0	0	$x+y+z$	(M_0)
0	0	1	$x+y+z'$	(M_1)
0	1	0	$x+y'+z$	(M_2)
0	1	1	$x+y'+z'$	(M_3)
1	0	0	$x'+y+z$	(M_4)
1	0	1	$x'+y+z'$	(M_5)
1	1	0	$x'+y'+z$	(M_6)
1	1	1	$x'+y'+z'$	(M_7)

↙ General Notation

- $m_i = M_i'$
- $f = m_0 + m_1 + m_3 \Rightarrow f' = \Sigma m(2,4,5,6,7) \Rightarrow \cancel{f' = \Pi M'(2,4,5,6,7)}$
  $\Rightarrow f = \Pi M(2,4,5,6,7)$.
- For n-variables → 2^{2^n} boolean functions.
- **Standard Form:** → SOP, → POS
  - Each Product term / sum term may contain any no. of literals Ex: $x + y'z$



Two-level implementation
(∵ 2 levels are there)

- # K-Map: (Karnaugh-map).

2-variable case:

A \ B	$\bar{B}$ 0	B 1
$\bar{A}$ 0	m_0	m_1
B. 1	m_2	m_3.

3-variable case:

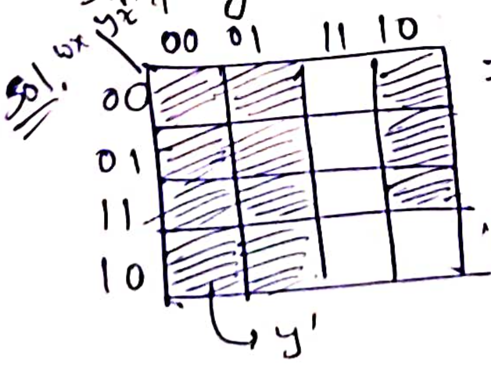
A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	m_0	m_1	m_3	m_2
A 1.	m_4	m_5	m_7	m_6

4-variable case:

wx \ yz	$y'z'$ 00	$y'z$ 01	yz 11	yz' 10
$w'x'$ 00	m_0	m_1	m_3	m_2
$w'x$ 01	m_4	m_5	m_7	m_6
wx 11	m_{12}	m_{13}	m_{15}	m_{14}
wx' 10	m_8	m_9	m_{11}	m_{10}

- one square ⟹ one minterm with four literals
- two adj. squares ⟹ one minterm with. three literals
- four adj. squares ⟹ one minterm with ~~eight~~ two literals
- eight adj. squares ⟹ one min-terms with one literal
- sixteen adj. squares ⟹ 1 is the result
- $F(w, x, y, z) = \sum(0,1,2,4,5,6,8,9,12,13,14)$. simplify

Sol.



$\Rightarrow F = y' + w'x'yz' + w'xyz' + wxyz'$

$F = y' + w'yz' + wxyz'$

$= y' + (w' + wx)yz'$

$= y' + w'yz' + w'yz' = \boxed{y' + w'z' + wxz'}$

Ex:

$Y = ABC + \bar{A}B + C$

$\Rightarrow Y = ABC + \bar{A}BC + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}C + ABC + \bar{A}\bar{B}C$

$\Rightarrow Y = ABC + \bar{A}\bar{B}C + \bar{A}BC + \bar{A}B\bar{C} + A\bar{B}C.$

From prev page

$Y = \sum m(1,2,3,5,7)$

From K-Map:

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

Possible pairing:

$(1,3,5,7) + (2)$ /

$(3,2) + (1) + (5,7)$