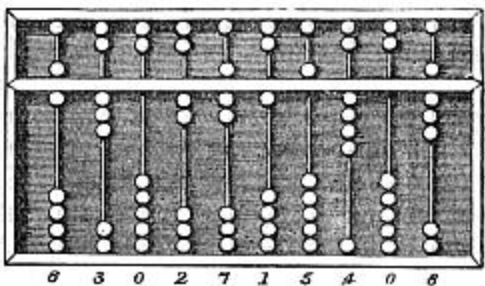




Елементи схемотехніки

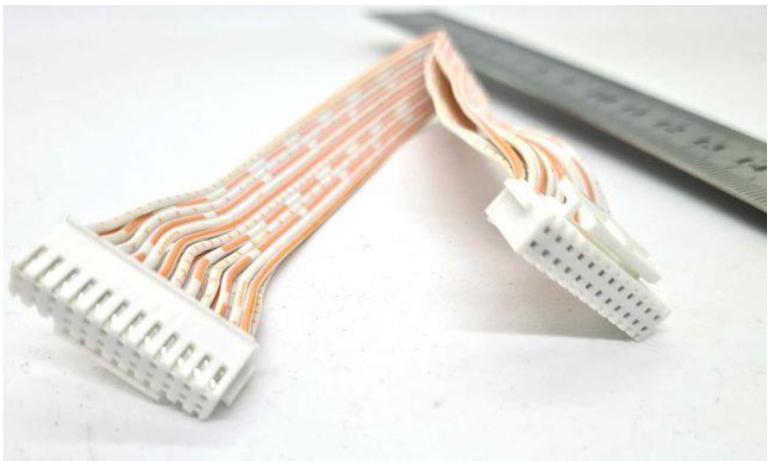
Історія обчислювальної техніки

- 1 етап

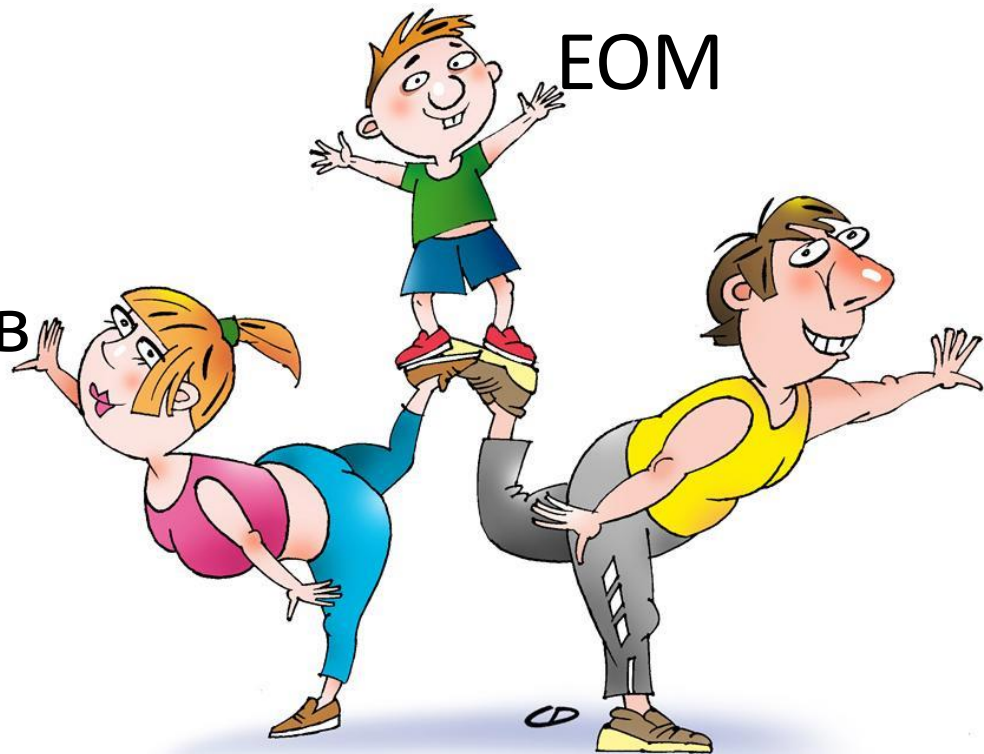


- 2 етап





Теорія
алгоритмів



ЕОМ

Наука
логіка

Математична логіка

- Одним із розділів Математичної логіки є **Алгебра логіки**.
- **Об'єкти алгебри логіки** - окремі **твердження** (речення), про які можна судити істинні вони або хибні.
- Ці твердження називаються **висловлюваннями**.

Висловлювання

Прості

Складені

$A, B, \dots X, Y, F1, \dots$

$\overline{A \& B}$

$A \vee B$

$\bar{A}$

$\overline{A \& B \vee C}$

$\overline{A \vee B} \& (A \vee B \vee C) \& A$

Операція	Позначення	Мовний зворот
Заперечення, інверсія, лог. НЕ)	$\neg A$, $\bar{A}$, not A, не A	«Не», «не вірно, що»
Кон'юнкція (лог. множення, лог. І)	$A \wedge B$, $A \& B$, $A \cdot B$, AB , $A \text{ і } B$, $A \text{ and } B$	«І», «як ..., так і», «разом з», «але», «хоча»
Диз'юнкція (лог. додавання, лог. АБО)	$A \vee B$, $A + B$, $A B$, $A \text{ АБО } B$, $A \text{ or } B$	«Або», «або ..., або ...», «або обидва разом»
Строга диз'юнкція (викл. диз'юнкція, викл. АБО)	$A \oplus B$, $A \text{ xor } B$	«Або ..., або», «лише ... або лише»
Імплікація (лог. слідування)	$A \rightarrow B$, $A \Rightarrow B$	«Якщо ..., то», «з ... впливає, що», «спричиняє»
Еквіваленція (еквівалентність, рівнозначність)	$A \leftrightarrow B$, $A \Leftrightarrow B$, $A \equiv B$	«Еквівалентно», «необхідно і достатньо»

Логічні елементи

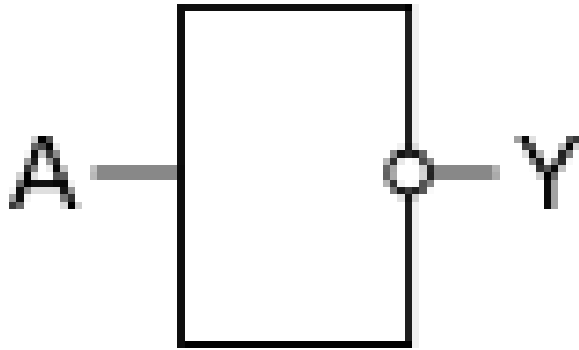
- **Логічний елемент** — [пристрій](#), призначений для обробки [інформації](#) в [цифровій формі](#) (послідовності [сигналів](#) високого — «1» і низького - «0» рівнів у [двійковій логіці](#))
- Фізично логічні [елементи](#) можуть бути виконані механічними, електромеханічними (на [електромагнітних реле](#)), електронними (на [діодах](#) і [транзисторах](#)), пневматичними, гідравлічними, оптичними та ін. способами.
- Із розвитком [електротехніки](#) від механічних логічних елементів поступово перейшли до електромеханічних логічних елементів (на електромагнітних реле), а потім до електронних логічних елементів на [електронних лампах](#), пізніше - на транзисторах. Після підтвердження в [1946](#) р. [теореми Джона фон Неймана](#) про економічність показникових [позиційних систем числення](#) стало відомо про переваги [двійкової](#) та [трійкової систем числення](#) в порівнянні з [десятьковою системою числення](#). Від десятикових логічних елементів перейшли до двійкових логічних елементів.



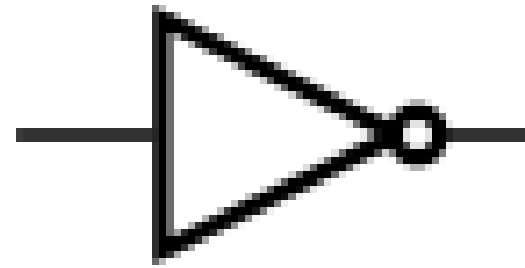
Логічні елементи

- 
- Логічний елемент – це електронний пристрій, що реалізує одну з логічних операцій над вхідними сигналами.

Заперечення. Операція "НЕ"



Інвертор, НЕ (IEC)



Інвертор, НЕ (ANSI)

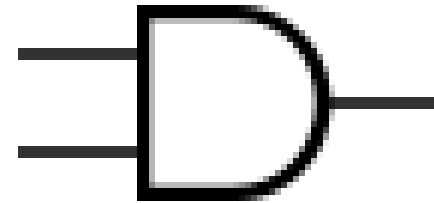
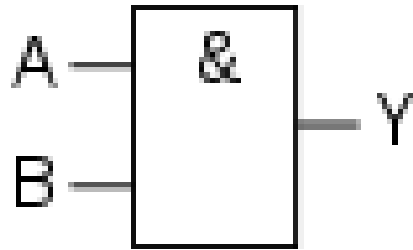
A	$\neg A$
0	1
1	0

Мнемонічне правило для НЕ звучить так:

На виході буде:

- «1» тоді і лише тоді, коли на вході «0»,
- «0» тоді і лише тоді, коли на вході «1»

Кон'юнкція. Операція "І"

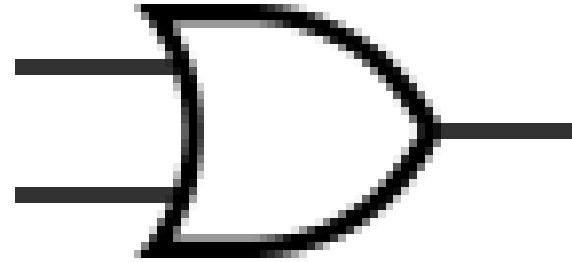
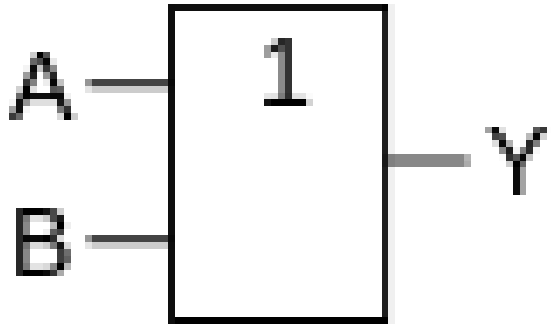


A	B	$A \wedge B$
0	0	0
1	0	0
0	1	0
1	1	1

Мнемонічне правило для І з будь-якою кількістю входів звучить так: На виході буде:

- «1» тоді і тільки тоді, коли на **всіх** входах діють «1»,
- «0» тоді і тільки тоді, коли **хоча б на одному** вході діє «0»

Диз'юнкція. Операція "АБО"



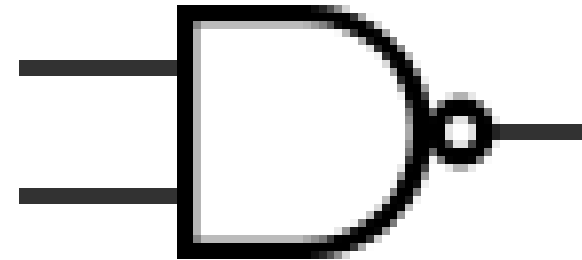
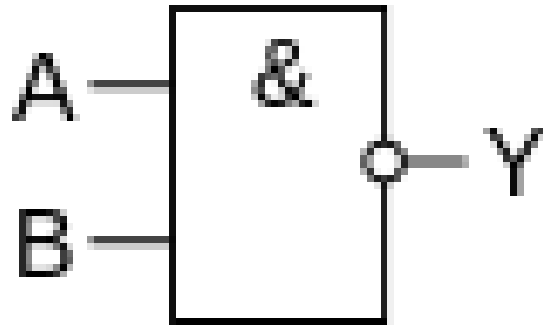
A	B	$B \vee A$
0	0	0
1	0	1
0	1	1
1	1	1

Мнемонічне правило для АБО з будь-якою кількістю входів звучить так: На виході буде:

- «1» тоді і тільки тоді, коли **хоча б на одному** вході діє «1»,
- «0» тоді і тільки тоді, коли на **всіх** входах діють «0»

Заперечення кон'юнкції. Операція "І-НЕ"

([штрих Шефера](#))

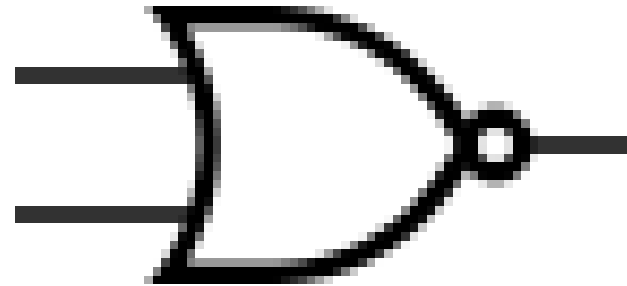
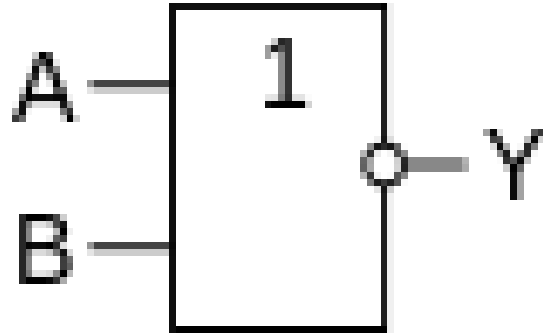


A	B	$A B$
0	0	1
0	1	1
1	0	1
1	1	0

Мнемонічне правило для І-НЕ з будь-якою кількістю входів звучить так: На виході буде:

- «1» тоді і тільки тоді, коли **хоча б на одному** вході діє «0»,
- «0» тоді і тільки тоді, коли на **всіх** входах діють «1»

Заперечення диз'юнкції. Операція "АБО-НЕ" ([стрілка Пірса](#))



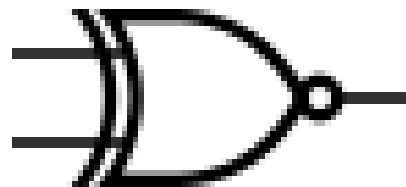
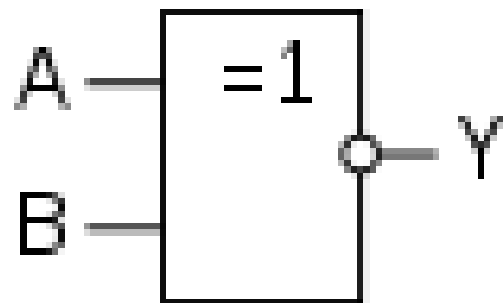
A	B	$A \downarrow B$
0	0	1
0	1	0
1	0	0
1	1	0

Мнемонічне правило для АБО-НЕ з будь-якою кількістю входів звучить так: На виході буде:

- «1» тоді і тільки тоді, коли на **всіх** входах діють «0»,
- «0» тоді і тільки тоді, коли **хоча б на одному** вході діє «1»

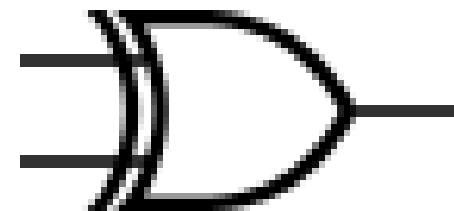
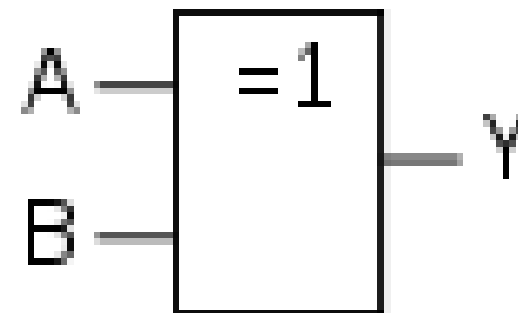
Еквіваленція. Операція
"ВИКЛЮЧНЕ_АБО-НЕ"

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

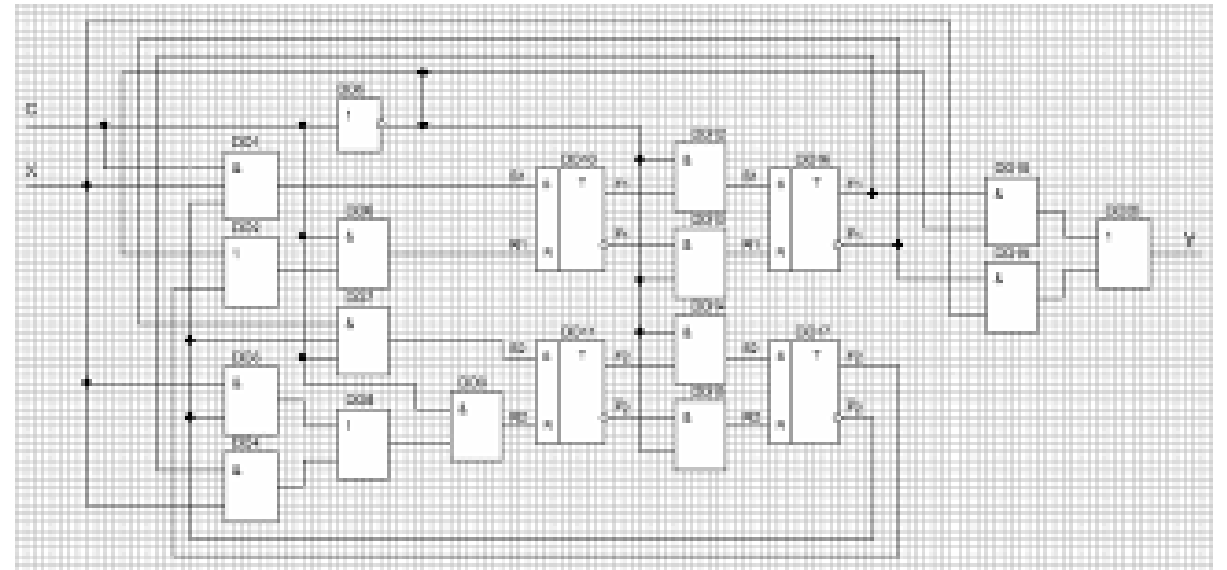
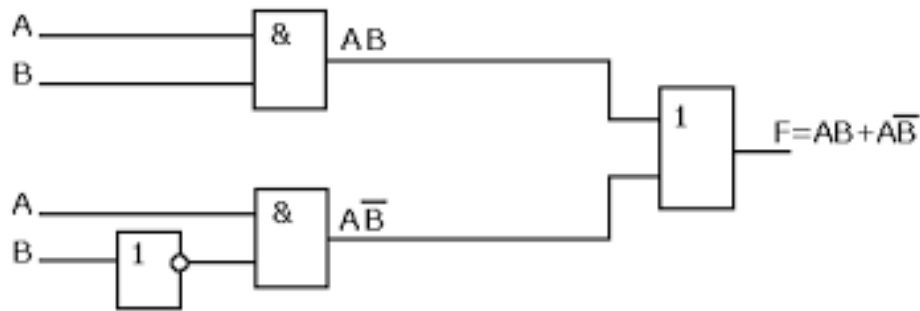
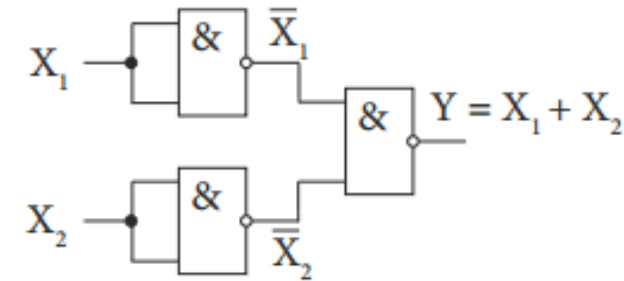
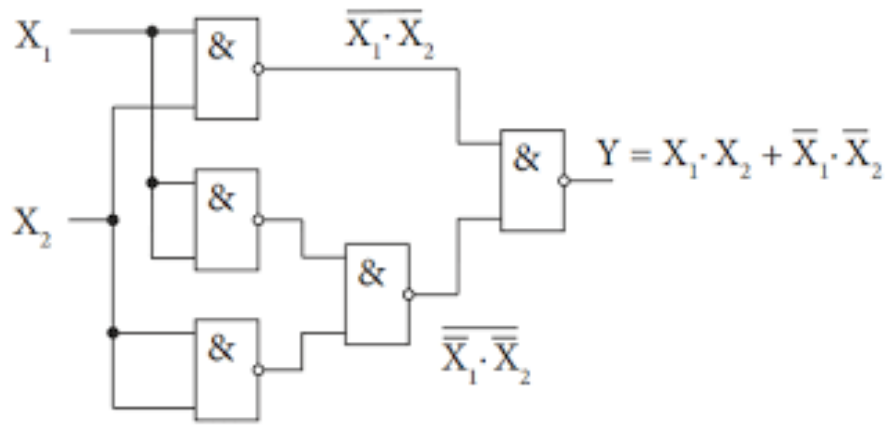


Виключна диз'юнкція.
Операція ВИКЛЮЧНЕ_АБО

A	B	$f(AB)$
0	0	0
0	1	1
1	0	1
1	1	0



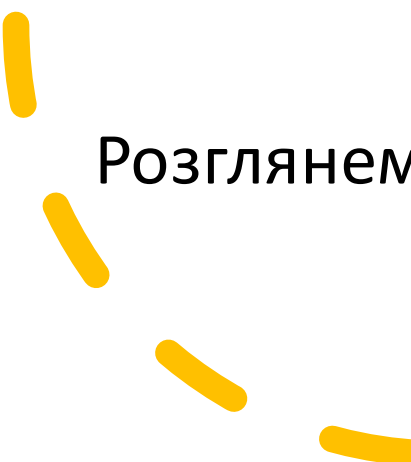
Сигнал, що виходить з виходу одного логічного елемента можна подавати на вхід другого логічного елемента. Це дає можливість утворювати ланцюг із окремих логічних елементів і утворюється функціональна (логічна) схема.





З окремих логічних елементів можна скласти, наприклад, пристрої, які здійснюють арифметичні операції над двійковими числами.

Визначення. Електронна логічна схема, що виконує підсумовування двійкових кодів, називається **суматором**.



Розглянемо схему складання двох n -розрядних двійкових чисел.

$$\begin{array}{r}
 a_n \dots a_i \dots a_1 a_0 \\
 + \quad b_n \dots b_i \dots b_1 b_0 \\
 \hline
 s_{n+1} s_n \dots s_i \dots s_1 s_0
 \end{array}$$

- При додаванні цифр i -го розряду складаються a_i і b_i до них додається p_i - ознака перенесення з $(i - 1)$ -го розряду. Результатом додавання буде s_i та p_{i+1} — ознака перенесення у наступний розряд.

Таким чином, **однорозрядний двійковий суматор** - це пристрій із трьома входами та двома виходами. Його робота описується такою таблицею істинності:

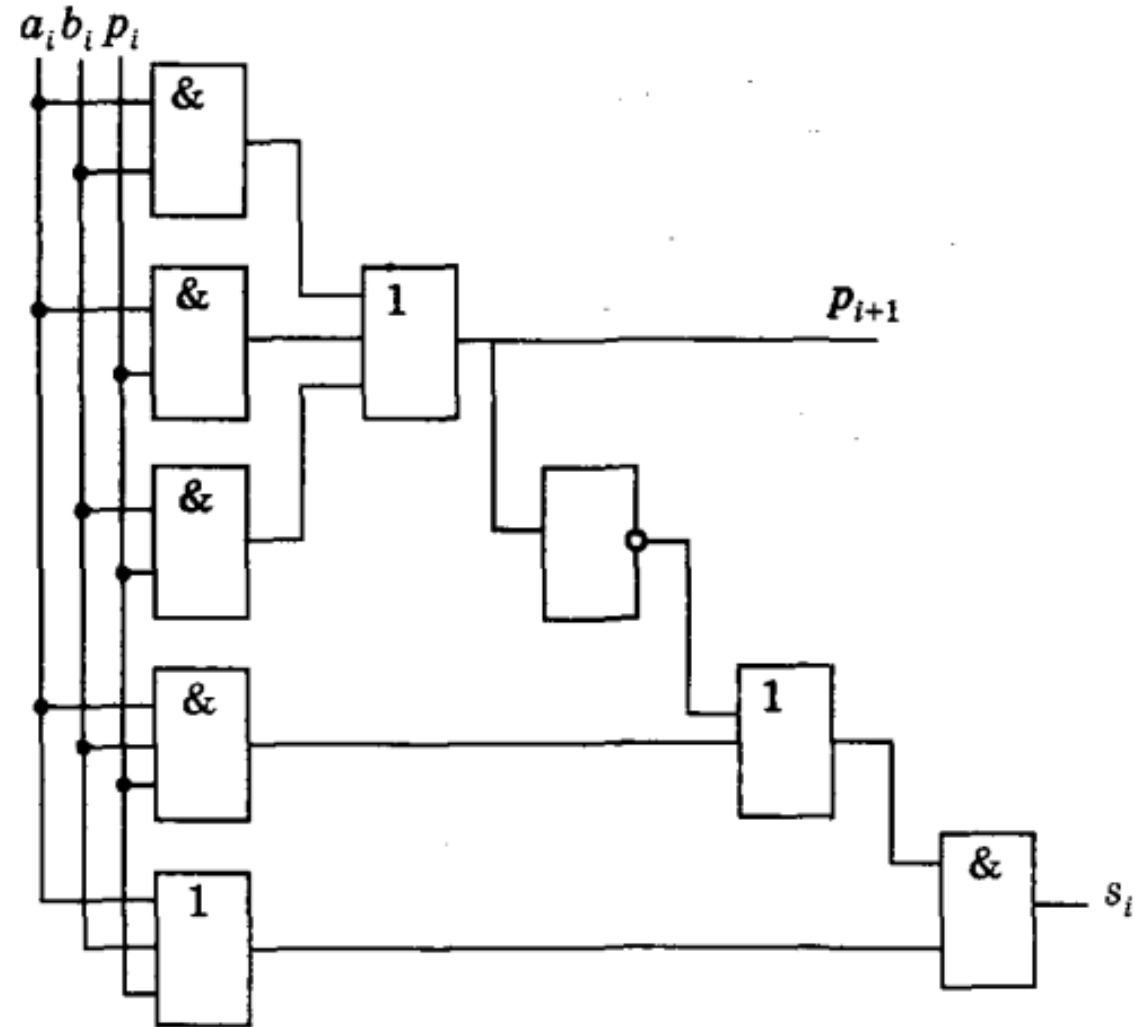
Входи			Виходи	
a_i	b_i	p_i	s_i	p_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Вихідні функції можна відновити за таблицею та спростити за допомогою тотожних перетворень. В результаті перетворень шукані функції набувають, наприклад, наступного вигляду:

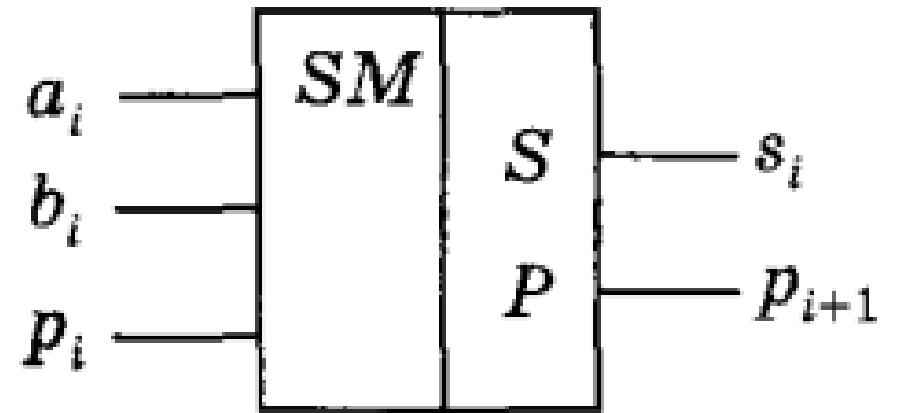
$$p_{i+1} = a_i \& b_i \vee a_i \& p_i \vee b_i \& p_i;$$

$$s_i = (\bar{p}_{i+1} \vee a_i \& b_i \& p_i) \& (a_i \vee b_i \vee p_i).$$

Однорозрядний двійковий суматор можна реалізувати наступною схемою:



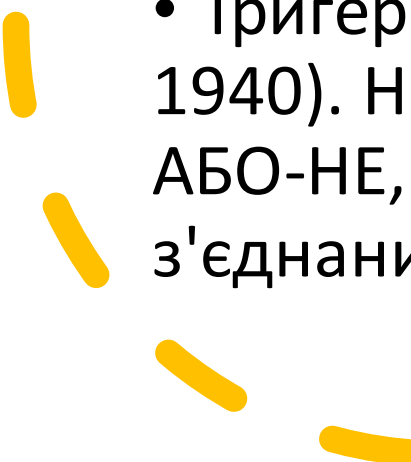
- Додавання n -розрядних двійкових чисел здійснюється за допомогою комбінації однорозрядних суматорів (умовне позначення одноразрядного суматора наведено на малюнку нижче).





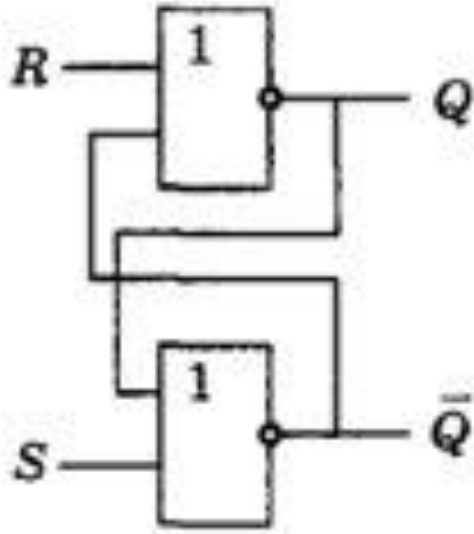
- У цифрових автоматах з пам'яттю набір вихідних сигналів залежить не тільки від набору вхідних сигналів, а й від внутрішнього стану пристрою. Такі пристрої мають пам'ять.

- **Визначення.** Логічний елемент, здатний зберігати один розряд двійкового числа називають **тригером**.



- Тригер був винайдений в 1918 р. М. А. Бонч-Бруєвічем (1888-1940). Найпростіший тригер - RS. Він складається з двох елементів АБО-НЕ, входи та виходи яких з'єднані кільцем: вихід першого з'єднаний з входом другого, вихід другого - з входом першого.

Схема RS-триггера:



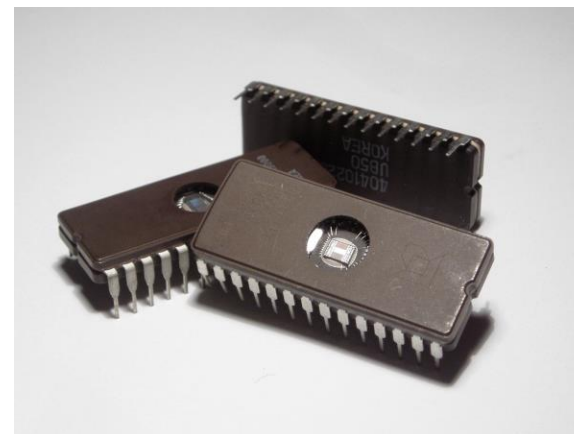
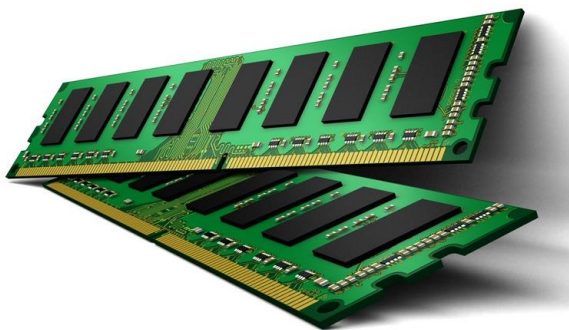
Тут: вхід S (set) – установка триггера в 1,
вхід R (reset) – установка триггера в 0.

Зазвичай на входи
надходять сигнали $R = 0$ і $S = 0$, і тригер зберігає старий
стан.

Якщо на вхід S надходить
на короткий час сигнал 1,
то тригер переходить у стан
1, і після того, як сигнал S
дорівнюватиме 0, тригер
буде зберігати стан 1.

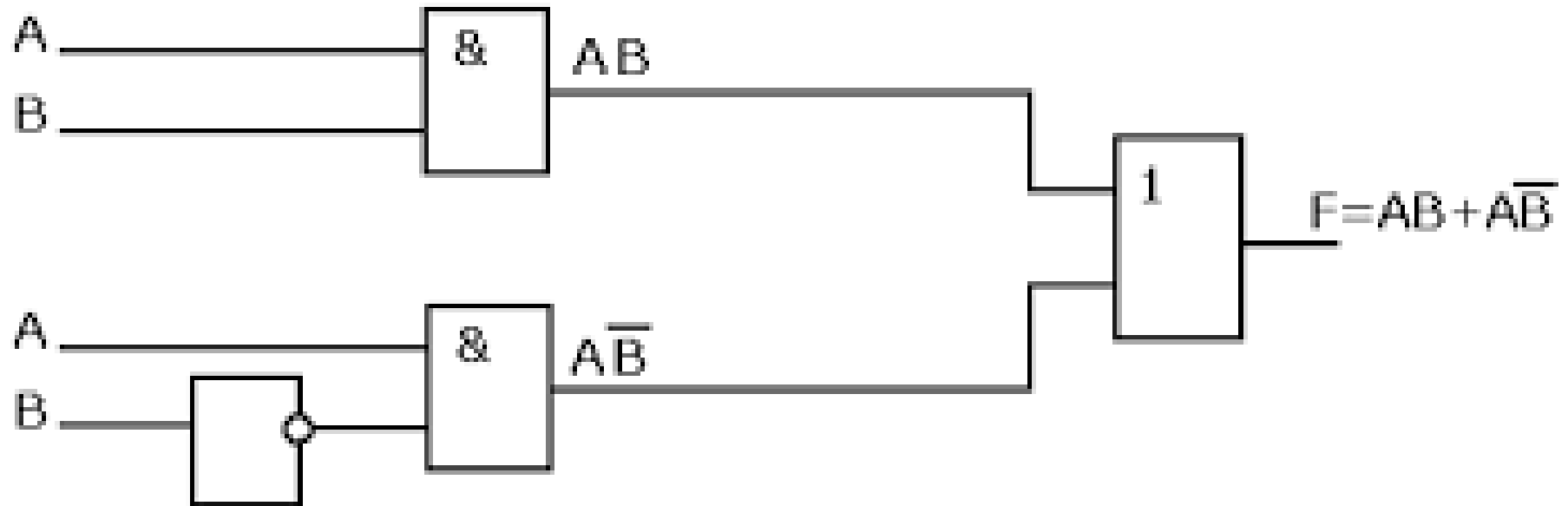
При подачі 1 на вхід R
тригер перейде в стан 0.

- Для зберігання **1 байта** інформації необхідно **8 тригерів**, для 1 кілобайта - 8×1024 тригери.
- Отже, оперативна пам'ять сучасних комп'ютерів містить мільйони тригерів. У цілому ж комп'ютер складається з величезної кількості логічних елементів, що утворюють всі його вузли та пам'ять.



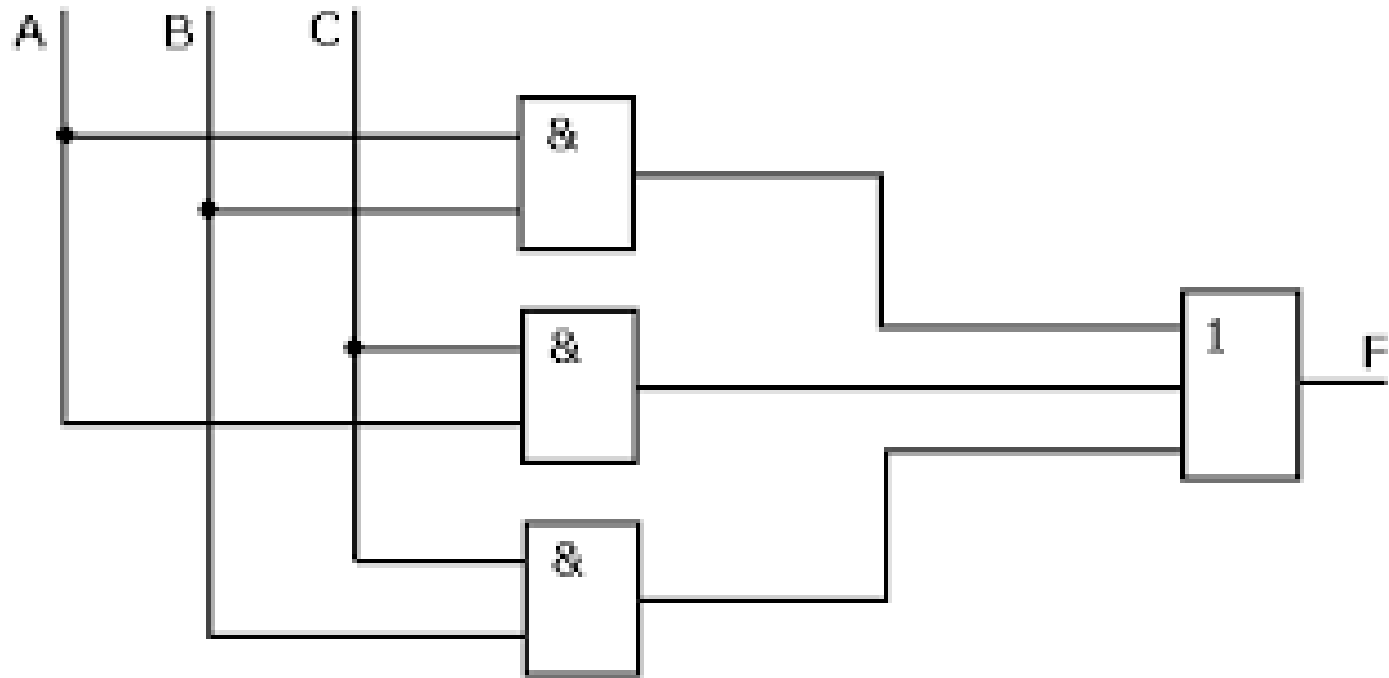
Завдання 1

Проаналізуйте схему, наведену на наступному малюнку, та випишіть формулу для функції F:



Завдання 2

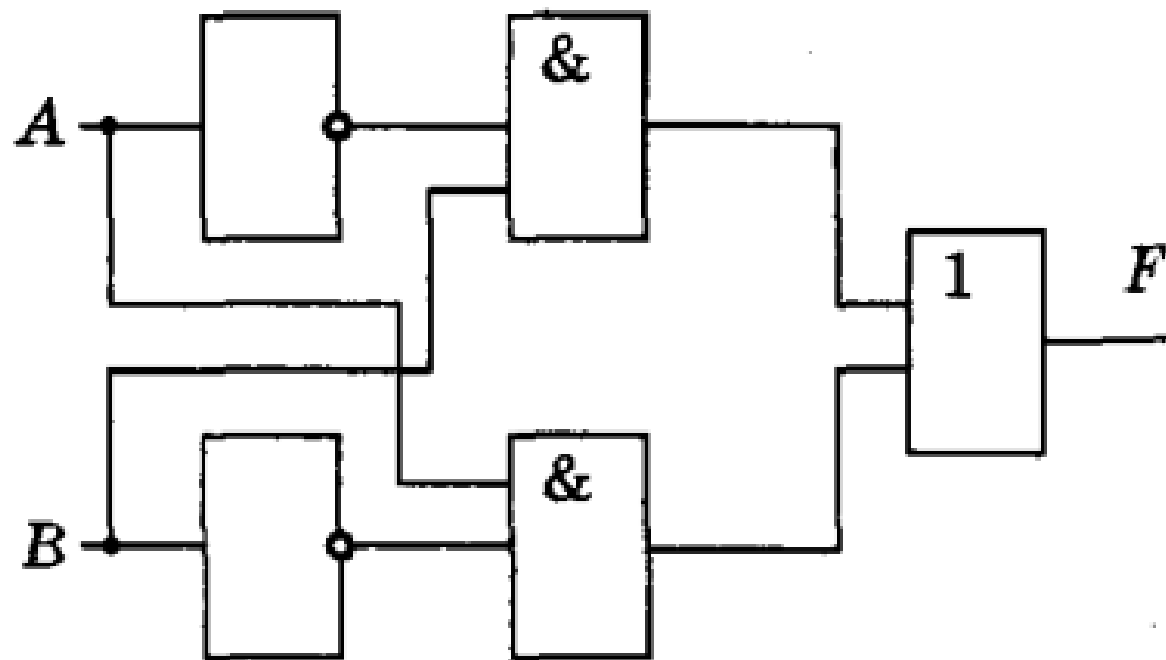
Проаналізуйте схему, наведену на наступному малюнку, та випишіть формулу для функції F:



$$F = AB + AC + BC$$

Завдання 3

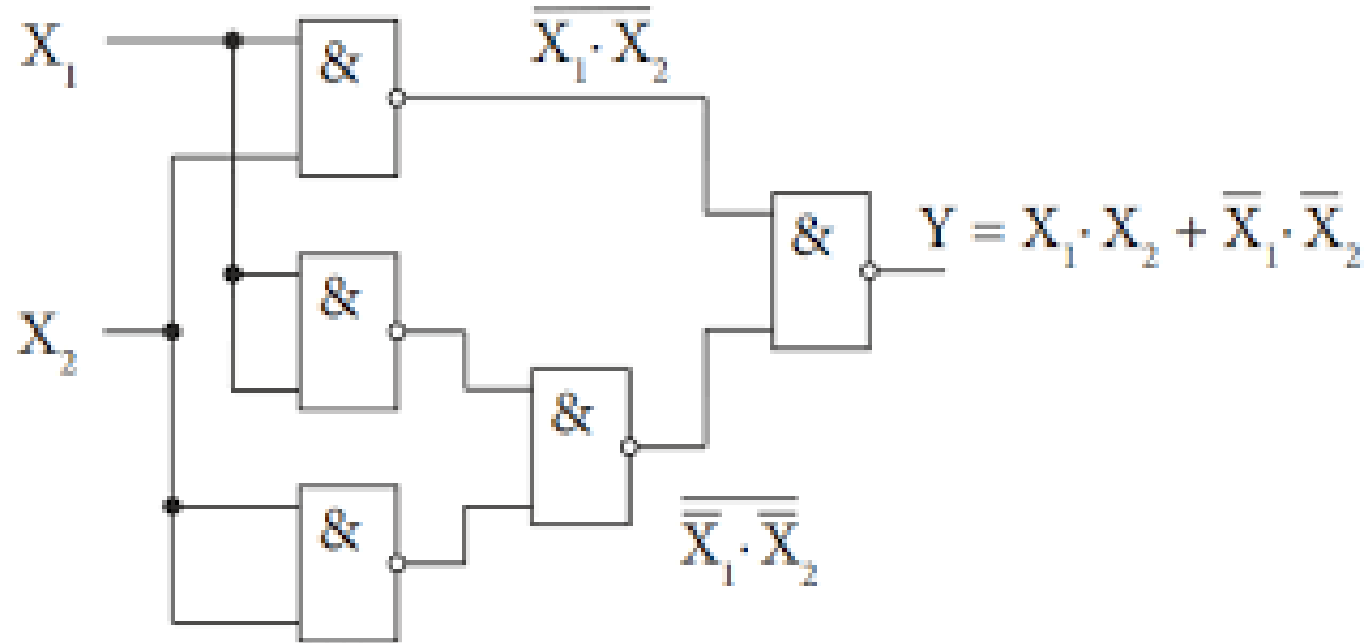
Проаналізуйте схему, наведену на наступному малюнку, та випишіть формулу для функції F :



$$F = \overline{A}B + \overline{B}A$$

Завдання 4

Проаналізуйте схему, наведену на наступному малюнку, та випишіть формулу для функції Y:

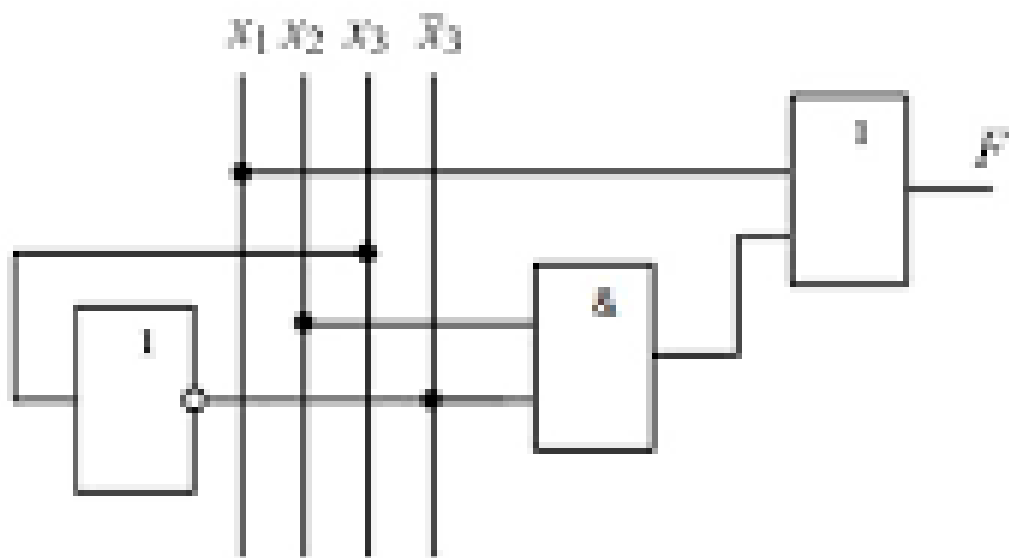


$$Y = \overline{X_1 \cdot X_2} \cdot \overline{\overline{X_1} \cdot \overline{X_2}} = X_1 \cdot X_2 + \overline{X_1} \cdot \overline{X_2}$$

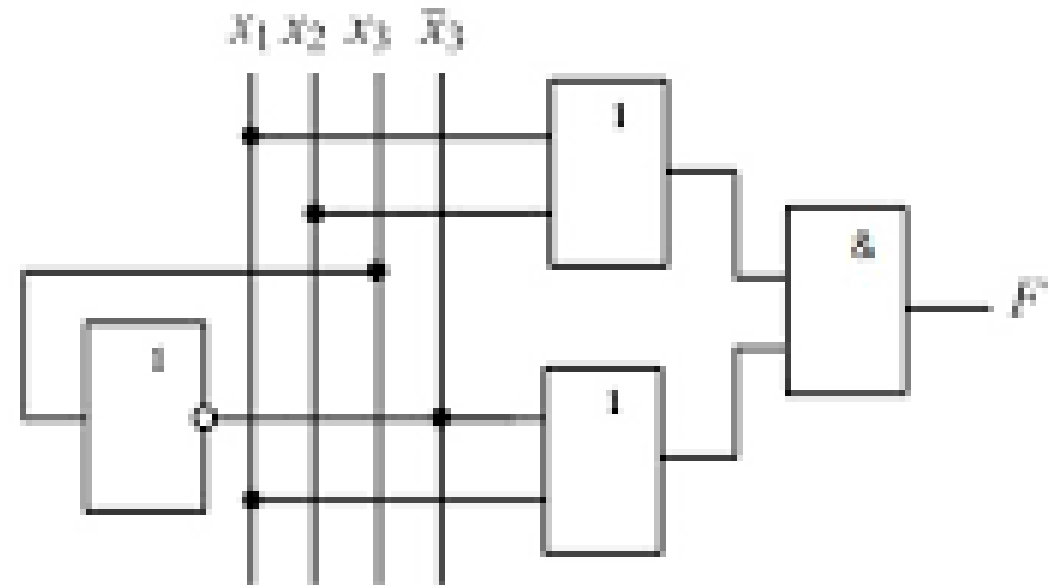
За законом де Моргана

Домашнє завдання

1. Проаналізуйте схему, наведену на наступному малюнку, та випишіть формулу для функції F :



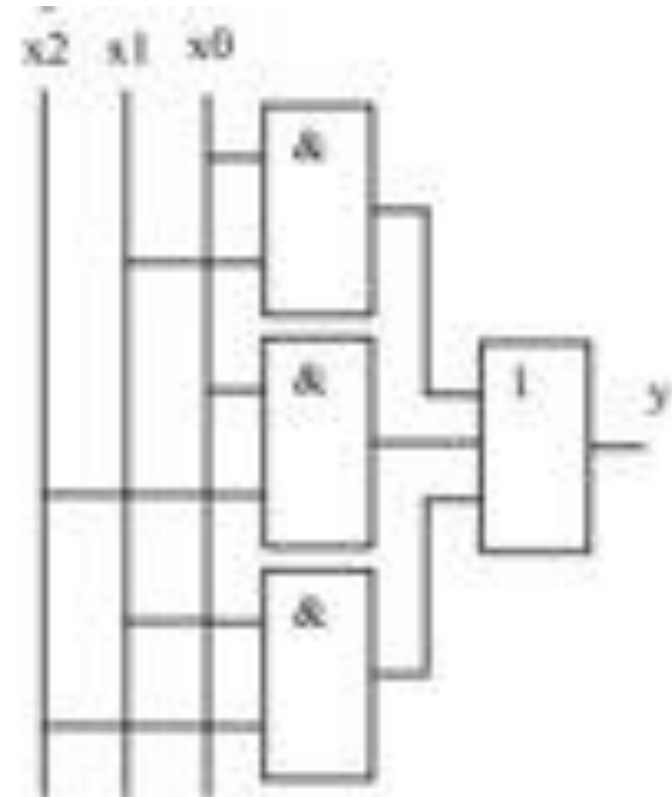
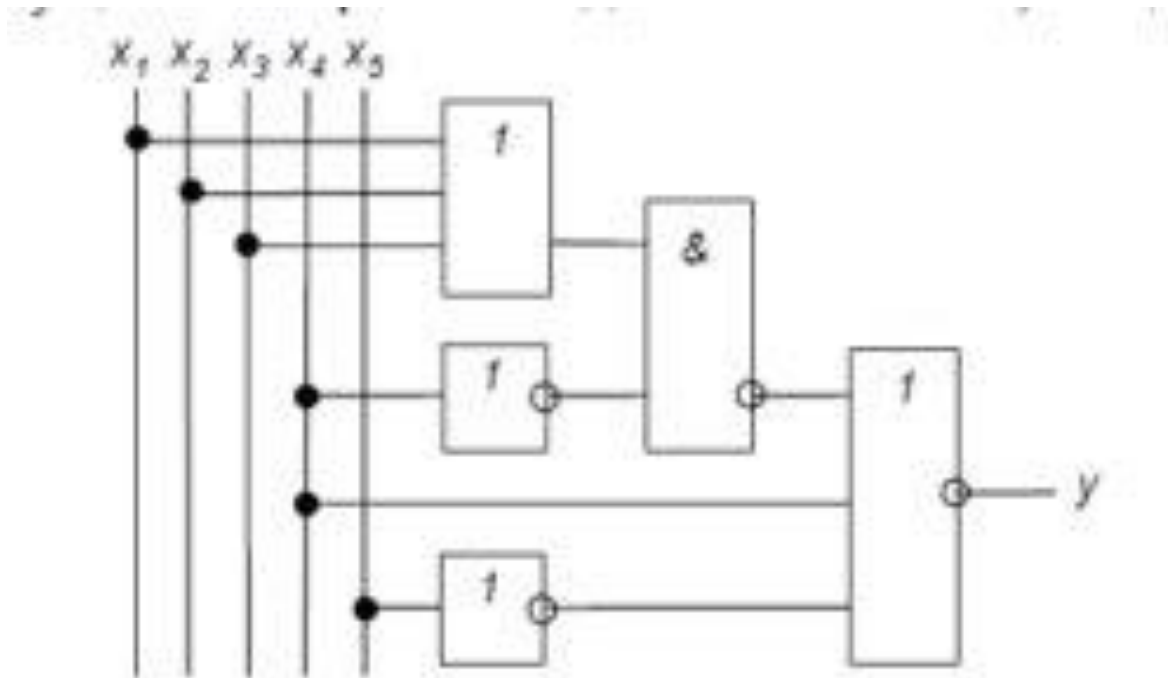
а)



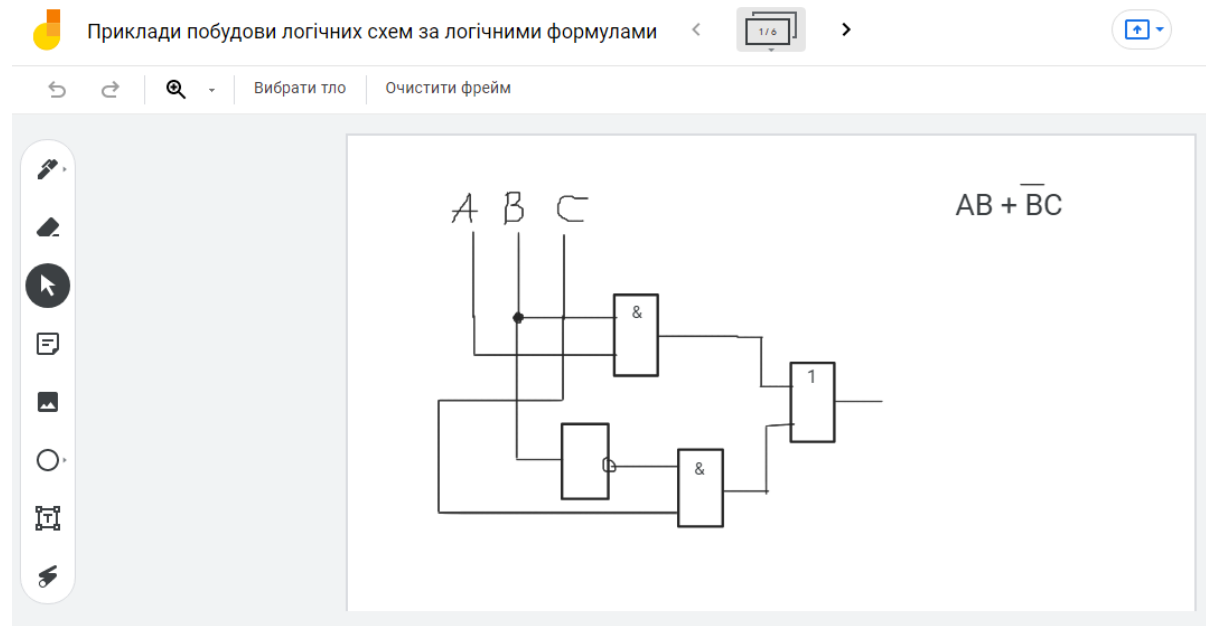
б)

Домашнє завдання

2. Проаналізуйте схему, наведену на наступному малюнку, та впишіть формулу для функції Y :



Приклади побудови логічних схем за логічними формулами



<https://bit.ly/3JjdAlg>