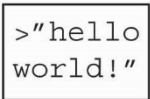
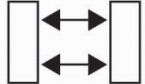
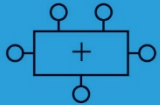
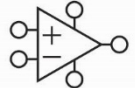
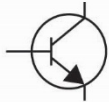


Розділ 2. Комбінаційна логіка

Теми

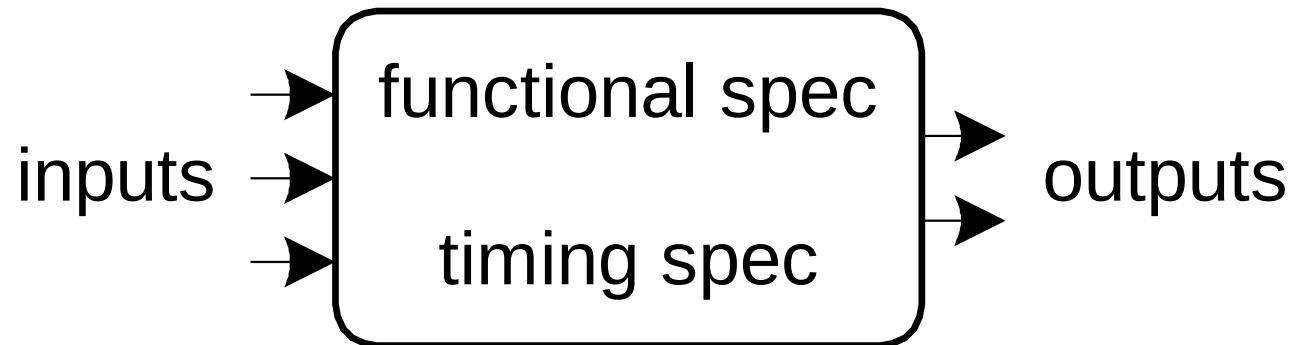
- Вступ
- Булеві вирази
- Булева алгебра
- Від логіки до логічних елементів
- Багаторівнева комбінаційна логіка
- Що за X і Z ?
- Карти Карно
- Базові комбінаційні блоки
- Часові характеристики

Application Software	
Operating Systems	
Architecture	
Micro-architecture	
Logic	
Digital Circuits	
Analog Circuits	
Devices	
Physics	

Вступ

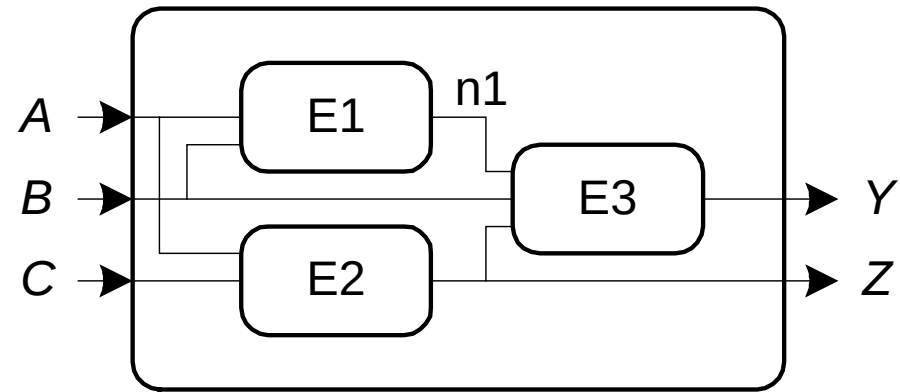
Логічна схема складається з:

- Вхідів
- Виходів
- Функціональної специфікації
- Часової специфікації



Схеми

- Вузли
 - Входи: A, B, C
 - Виходи: Y, Z
 - Внутрішній вузол: $n1$
- Елементи схеми
 - $E1, E2, E3$
 - Кожний з них, в свою чергу, є схемою



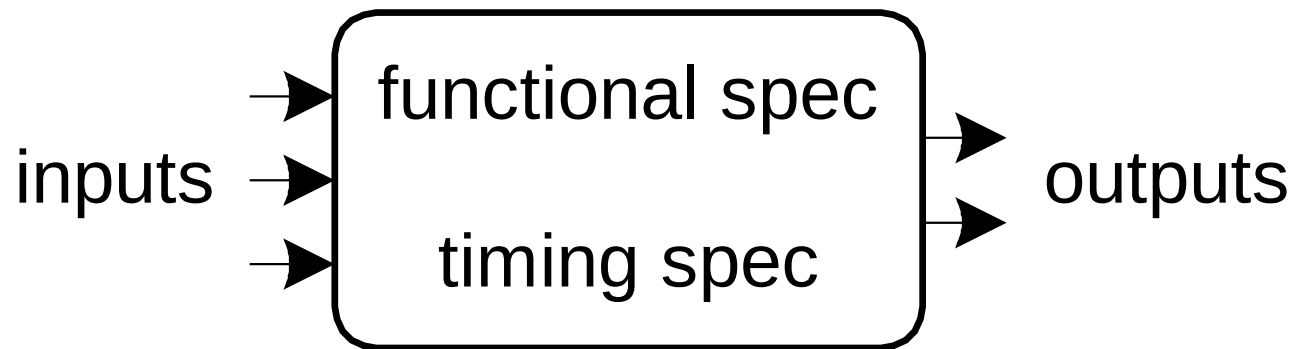
Типи цифрових схем

- **Комбінаційні цифрові схеми**

- Не мають пам'яті
- Вихід визначається поточним станом входів

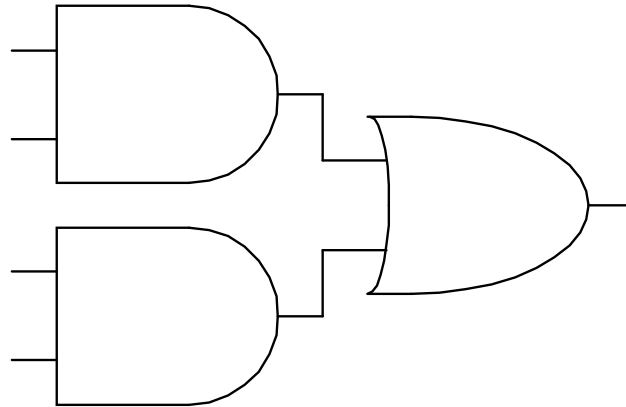
- **Послідовнісні цифрові схеми**

- Мають пам'ять
- Вихід визначається поточним і попередніми станами входів



Правила комбінаційної композиції

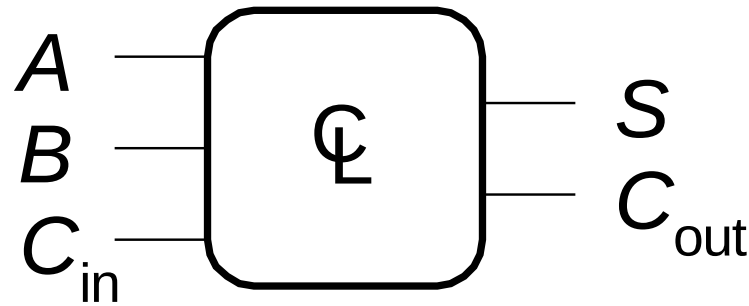
- Кожний елемент сам є комбінаційним
- Кожний вузел схеми є або входом, або приєднаний до одного єдиного виходу другого елементу
- Схема не містить циклічних шляхів
- **Приклад:**



Булеві вирази

- Функціональна специфікація виходів за значеннями входів
- Приклад: $S = F(A, B, C_{in})$

$$C_{out} = F(A, B, C_{in})$$



$$S = A \oplus B \oplus C_{in}$$
$$C_{out} = AB + AC_{in} + BC_{in}$$

Деякі визначення

- Доповнення: змінна з рискою над іменем

$$\overline{A}, \overline{B}, \overline{C}$$

- Літерал: змінна або її доповнення

$$A, \overline{A}, B, \overline{B}, C, \overline{C}$$

- Імпліканта: добуток літералів

$$AB\overline{C}, A\overline{C}, B\overline{C}$$

- Мінтерм: добуток, у який входять літерали усіх вхідних змінних

$$AB\overline{C}, A\overline{B}\overline{C}, ABC$$

- Макстерм: сума, у яку входять літерали всіх вхідних змінних

$$(A + \overline{B} + C), (\overline{A} + B + \overline{C}), (A + \overline{B} + \overline{C})$$

Диз'юнктивна форма

- Всі вирази можуть бути записані в диз'юнктивній формі
- Кожному рядку таблиці істиності відповідає **мінтерм**, який має значення **ІСТИНА** для цього рядка
- Мінтерм є добутком (I, AND) літералів
- Диз'юнктивна функція є сумою мінтермів тих рядків, для яких вихід Y дорівнює ІСТИНІ
- Таким чином, формується канонічна форма сум (АБО, OR) добутків (І, AND)

A	B	Y	minterm	minterm name
0	0	0	$\bar{A} \bar{B}$	m_0
0	1	1	$\bar{A} B$	m_1
1	0	0	$A \bar{B}$	m_2
1	1	1	$A B$	m_3

$$Y = F(A, B) = m_1 + m_3 = \bar{A}B + AB = \Sigma(1, 3)$$

Кон'юнктивна форма

- Всі булеві вирази можуть бути записані в кон'юнктивній формі
- Кожному рядку відповідає **макстерм**, який має значення **ХИБА** для цього рядка
- Макстерм є сумою (АБО, OR) літералів
- Кон'юнктивна функція є добутком макстермів тих рядків, для яких вихід Y дорівнює ХИБІ
- Таким чином, формується добуток (І, AND) сумм (АБО, OR)

A	B	Y	maxterm	maxterm name
0	0	0	$A + B$	M_0
0	1	1	$A + \overline{B}$	M_1
1	0	0	$\overline{A} + B$	M_2
1	1	1	$\overline{A} + \overline{B}$	M_3

$$Y = F(A, B) = M_0 + M_2 = (A + B)(\overline{A} + B) = Д(0, 2)$$

Диз'юнктивна і кон'юнктивна форми

- Диз'юнктивна форма (SOP, sum-of-products) сума (АБО) добутків (І)

O	C	E	minterm
0	0	0	$\overline{O} \overline{C}$
0	1	0	$\overline{O} C$
1	0	1	$O \overline{C}$
1	1	0	$O C$

$$Y = O\overline{C} = \Sigma(2)$$

- Кон'юнктивна форма (POS, product-of-sums) – добуток (І) сум (АБО)

O	C	E	maxterm
0	0	0	$O + C$
0	1	0	$O + \overline{C}$
1	0	1	$\overline{O} + C$
1	1	0	$\overline{O} + \overline{C}$

$$Y = (O + C)(O + \overline{C})(\overline{O} + \overline{C}) = \Pi(0, 1, 3)$$

Булева алгебра

- Аксиоми і теореми дозволяють **спростити** булеві вирази
- Подібно до звичайної алгебри, але простіше: змінні приймають тільки два значення (0 або 1)
- **Двоякість** аксіом і теорем:
 - Можна взаємно замінити І або АБО, 0 і 1

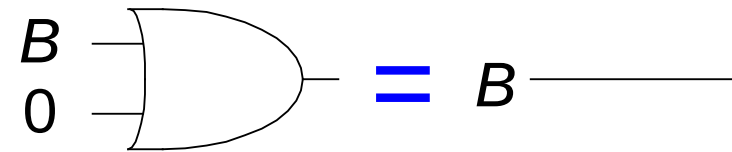
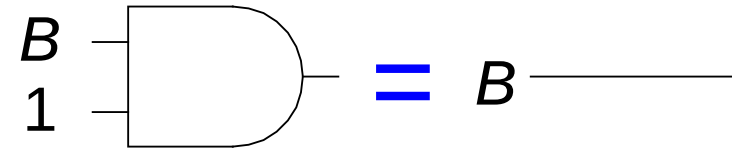
Булєві аксіоми

Axiom		Dual		Name
A1	$B = 0 \text{ if } B \neq 1$	A1'	$B = 1 \text{ if } B \neq 0$	Binary field
A2	$\overline{0} = 1$	A2'	$\overline{1} = 0$	NOT
A3	$0 \bullet 0 = 0$	A3'	$1 + 1 = 1$	AND/OR
A4	$1 \bullet 1 = 1$	A4'	$0 + 0 = 0$	AND/OR
A5	$0 \bullet 1 = 1 \bullet 0 = 0$	A5'	$1 + 0 = 0 + 1 = 1$	AND/OR

Theorem		Dual		Name
T1	$B \bullet 1 = B$	T1'	$B + 0 = B$	Identity
T2	$B \bullet 0 = 0$	T2'	$B + 1 = 1$	Null Element
T3	$B \bullet B = B$	T3'	$B + B = B$	Idempotency
T4		$\overline{\overline{B}} = B$		Involution
T5	$B \bullet \overline{B} = 0$	T5'	$B + \overline{B} = 1$	Complements

Теорема ідентичності

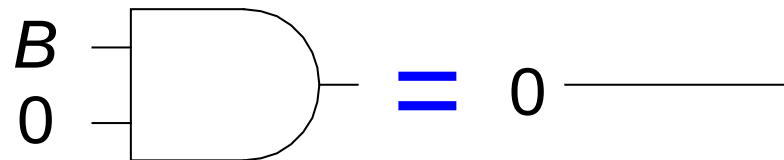
- $B \cdot 1 = B$
- $B + 0 = B$



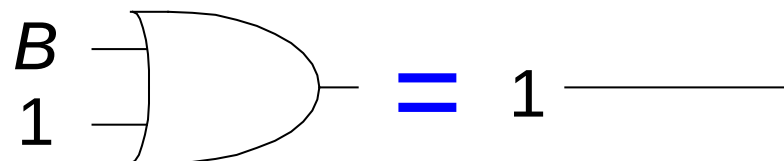
- Заміна 2-входового AND з'єднанням входу B з дротом
- Заміна 2-входового OR з'єднанням входу B з дротом

Теорема про нульовий елемент

- $B \cdot 0 = 0$



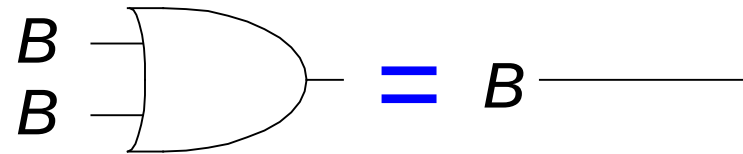
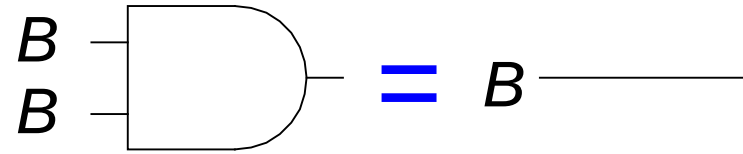
- $B + 1 = 1$



- Заміна 2-входового AND дротом під'єднаним до логічного рівня 0
- Заміна 2-входового OR дротом під'єднаним до логічного рівня 1

Теорема про подібність (ідемпотентність)

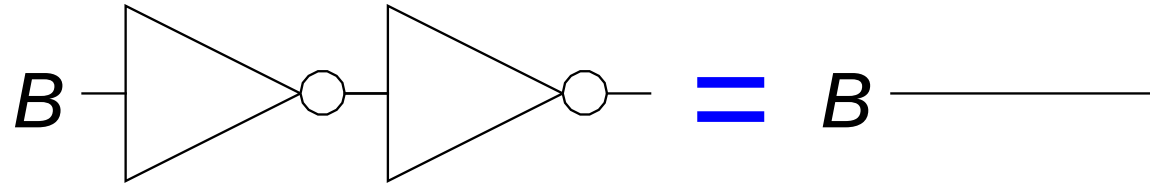
- $B \cdot B = B$
- $B + B = B$



- Заміна 2-входового AND дротом
- Заміна 2-входового OR дротом

Теорема ідентичності

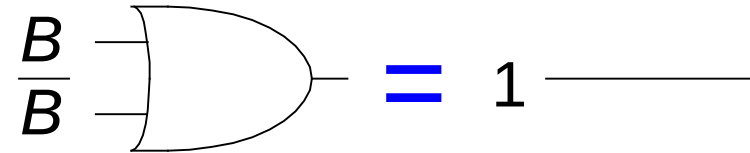
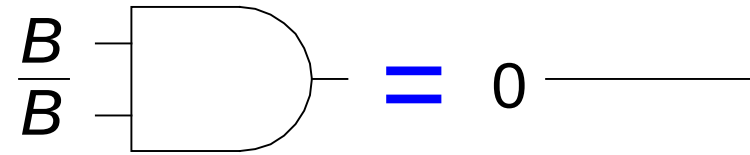
- $\overline{\overline{B}} = B$



- Заміна подвійної інверсії дротом

Теорема доповнювальності

- $B \cdot \overline{B} = 0$
- $B + \overline{B} = 1$



- Заміна 2-входового AND з інверсними входами дротом під'єднаним до логічного рівня 0
- Заміна 2-входового OR з інверсними входами дротом під'єднаним до логічного рівня 1

Булеві теореми декількох змінних

Theorem		Dual		Name
T6	$B \bullet C = C \bullet B$	T6'	$B + C = C + B$	Commutativity
T7	$(B \bullet C) \bullet D = B \bullet (C \bullet D)$	T7'	$(B + C) + D = B + (C + D)$	Associativity
T8	$(B \bullet C) + B \bullet D = B \bullet (C + D)$	T8'	$(B + C) \bullet (B + D) = B + (C \bullet D)$	Distributivity
T9	$B \bullet (B + C) = B$	T9'	$B + (B \bullet C) = B$	Covering
T10	$(B \bullet C) + (B \bullet \overline{C}) = B$	T10'	$(B + C) \bullet (B + \overline{C}) = B$	Combining
T11	$(B \bullet C) + (\overline{B} \bullet D) + (C \bullet D)$ $= B \bullet C + \overline{B} \bullet D$	T11'	$(B + C) \bullet (\overline{B} + D) \bullet (C + D)$ $= (B + C) \bullet (\overline{B} + D)$	Consensus
T12	$\overline{B_0 \bullet B_1 \bullet B_2 \dots}$ $= (\overline{B_0} + \overline{B_1} + \overline{B_2} \dots)$	T12'	$\overline{B_0 + B_1 + B_2 \dots}$ $= (\overline{B_0} \bullet \overline{B_1} \bullet \overline{B_2})$	De Morgan's Theorem

Спрощення булевих виразів

Приклад 1:

- $Y = AB + AB$

$$= B(A + A) \quad T8$$

$$= B(1) \quad T5'$$

$$= B \quad T1$$

Приклад 2:

- $Y = A(AB + ABC)$

$$= A(AB(1 + C)) \quad T8$$

$$= A(AB(1)) \quad T2'$$

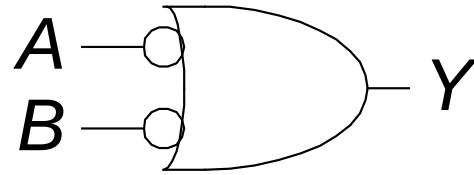
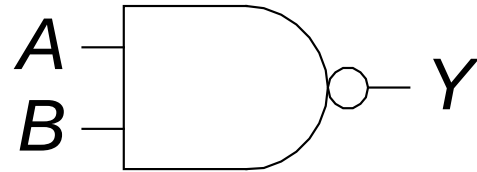
$$= A(AB) \quad T1$$

$$= (AA)B \quad T7$$

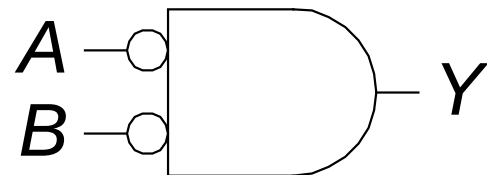
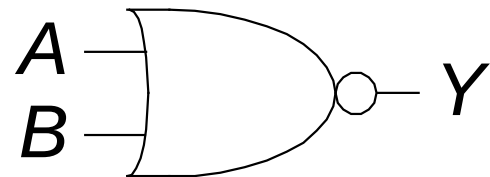
$$= AB \quad T3$$

Теорема де Моргана

- $Y = AB = A + B$



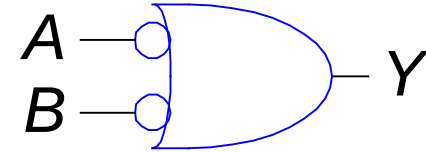
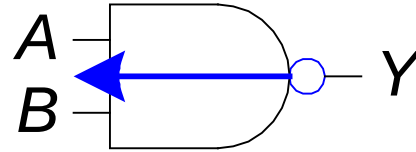
- $Y = A + B = A \cdot B$



Переміщення інверсії

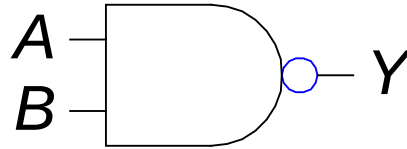
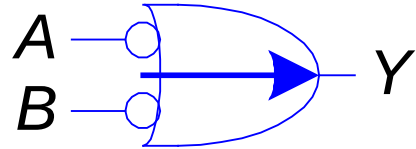
- **Назад:**

- Змінити тип елемента
- Додати інверсію на входи



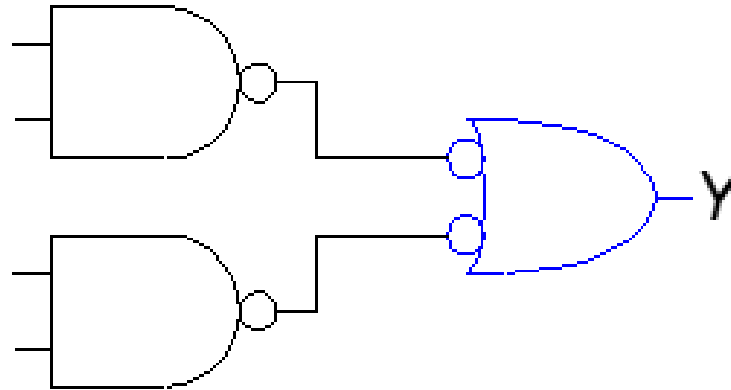
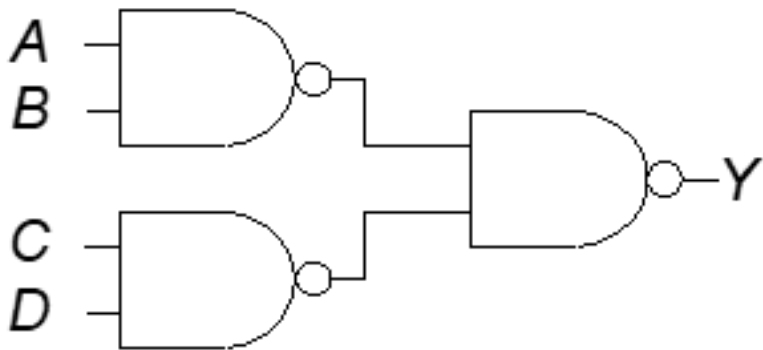
- **Вперед:**

- Змінити тип елемента
- Додати інверсію на вихід



Переміщення інверсії

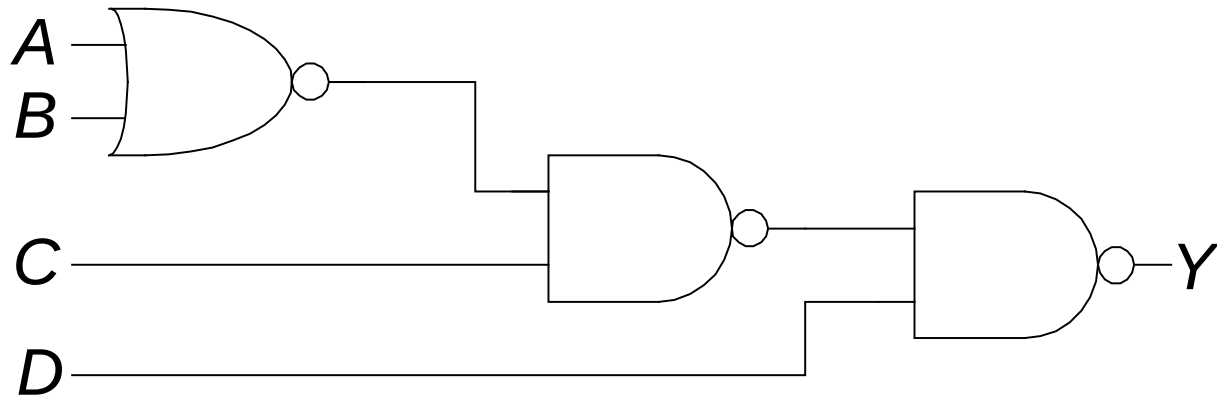
- Булевий вираз схеми після переміщення інверсії



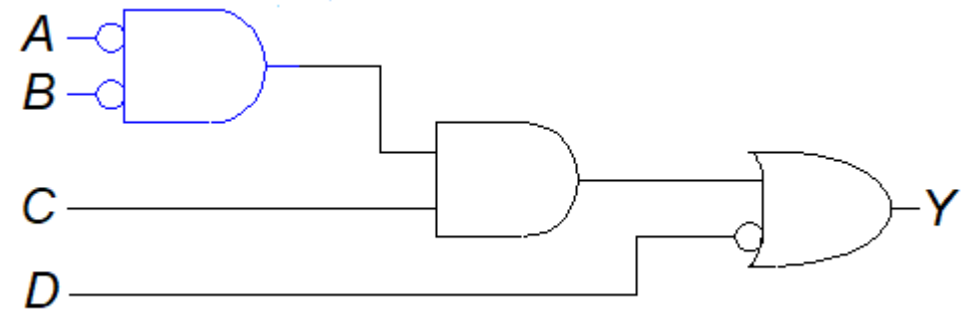
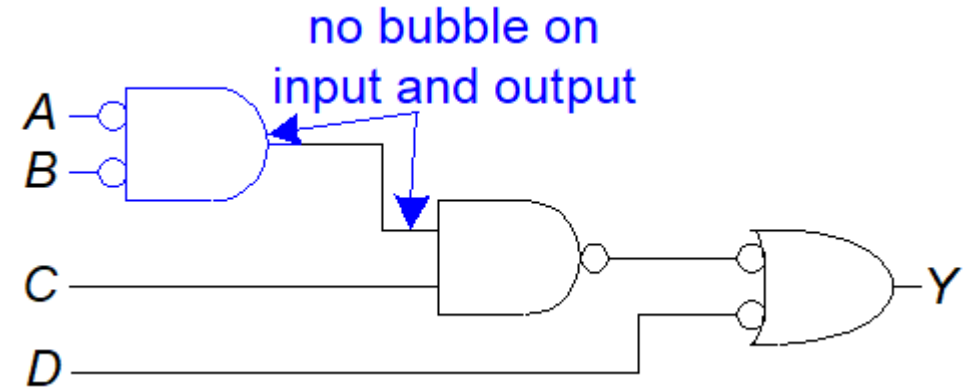
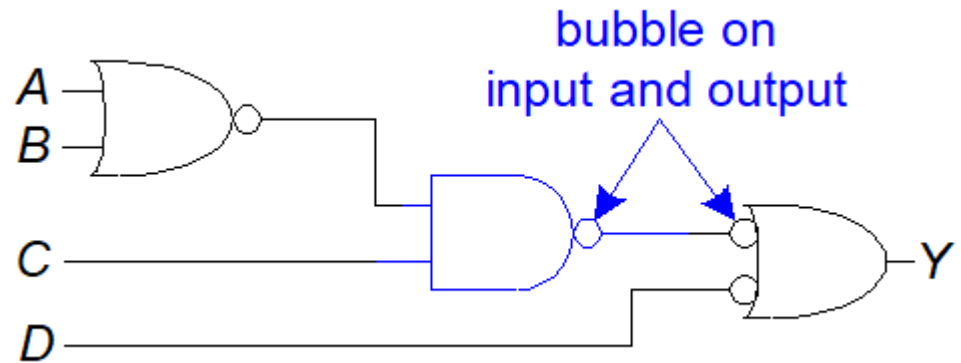
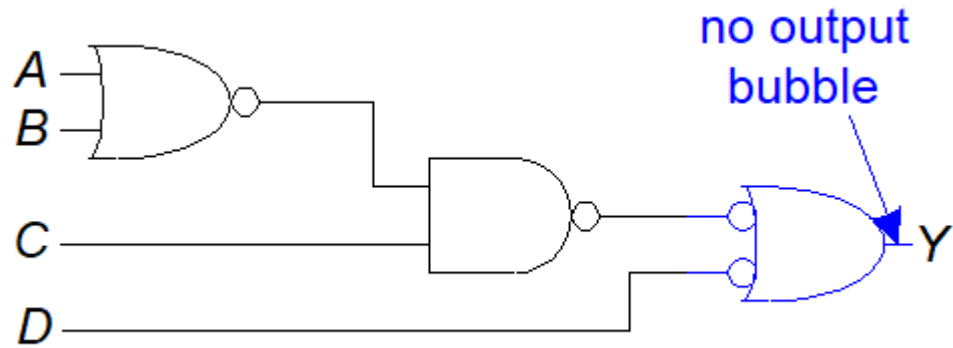
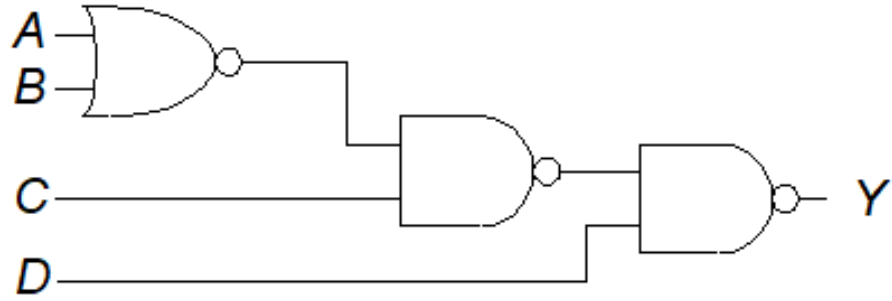
$$Y = AB + CD$$

Правила переміщення інверсії

- Почати з виходу і рухатися в напрямку входів
- Перемещати інверсію від виходів до входів
- Намалювати елементи так, щоб побачити, що інверсії взаємно знищуються



Приклад переміщення інверсії

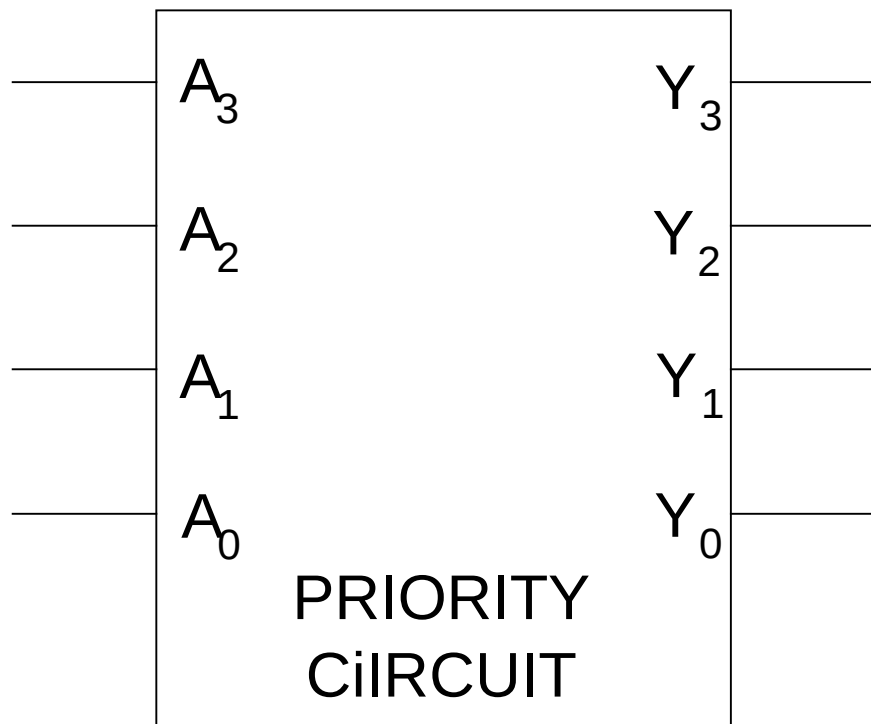


$$Y = \overline{A}BC + \overline{D}$$

Схеми з декількома виходами

- Приклад: Схеми пріоритету

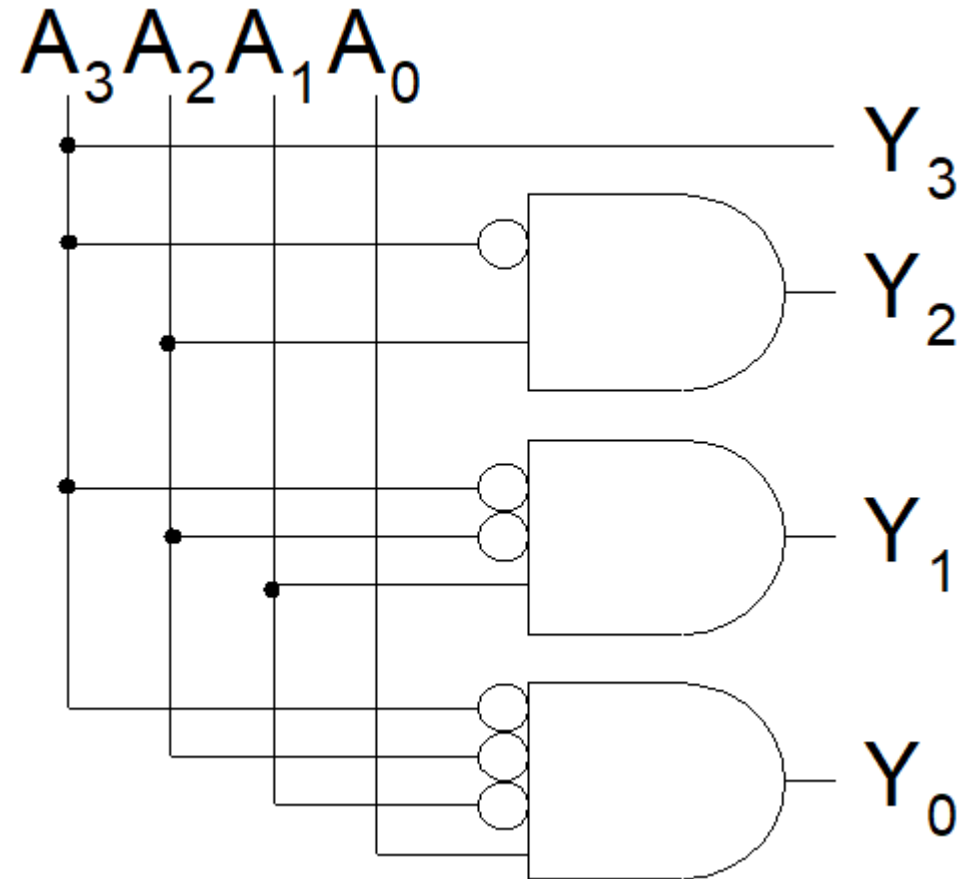
Вихід встановлюється у відповідності із старшим разрядом який дорівнює ІСТИНІ



A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	0

Реалізація схеми пріоритету

A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	0

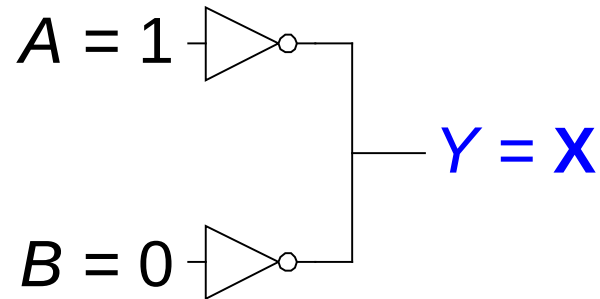


Довільне значення (конфлікт)

A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0	A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	1	0	0	1	0	0	0	1	X	0	0	1	0
0	1	0	0	0	1	0	0	0	1	X	X	0	1	0	0
0	1	0	1	0	1	0	0	1	X	X	X	1	0	0	0
0	1	1	0	0	1	0	0								
0	1	1	1	0	1	0	0								
1	0	0	0	1	0	0	0								
1	0	0	1	1	0	0	0								
1	0	1	0	1	0	0	0								
1	0	1	1	1	0	0	0								
1	1	0	0	1	0	0	0								
1	1	0	1	1	0	0	0								
1	1	1	0	1	0	0	0								
1	1	1	1	1	0	0	0								

Конфлікт: X

- Конфлікт: схема пробує встановити на виході одночасно 0 і 1
 - Дійсне значення знаходиться між межевими значеннями
 - Може бути 0, або 1, або у забороненому діапазоні
 - Може залежати від напруги, температури, часу, шуму
 - Часто спричиняє велике енергоспоживання

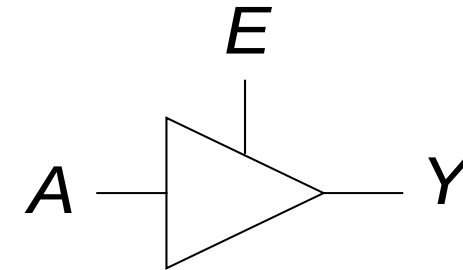


- **Попередження:**
 - Конфлікт звичайно є **помилкою**
 - В залежності від контексту **X** позначає довільне значення або конфлікт

Третій стан: Z

Буфер із трьома станами

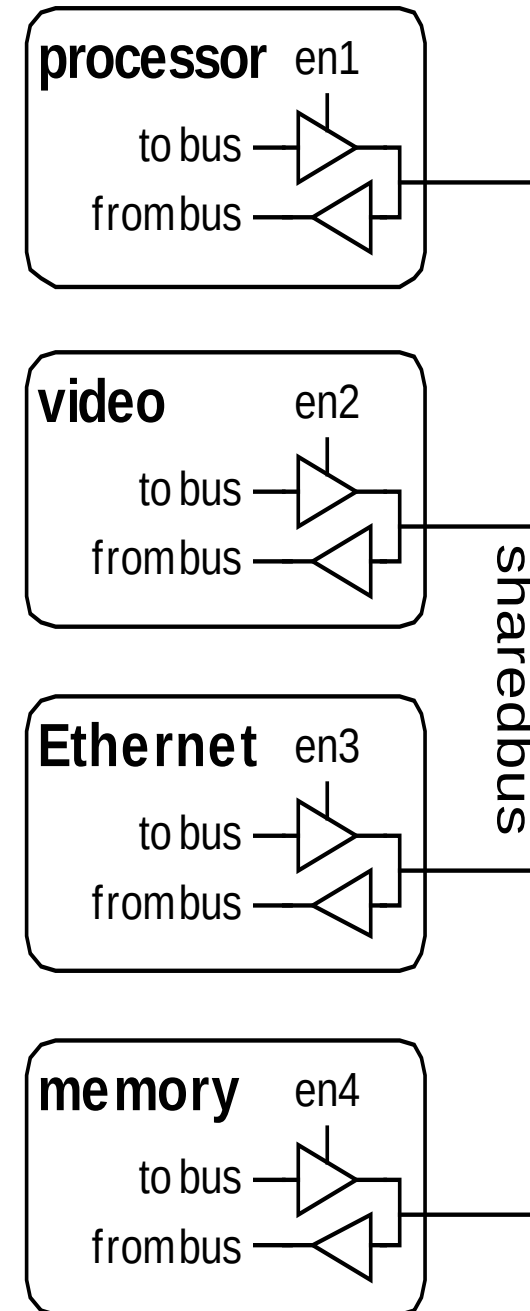
- Третій стан, стан високого імпедансу, непідключений або плаваючий ланцюг
- Стан непідключеного виходу может бути 0, або 1, або проміжним
- Вольтметр не показує, що вузол знаходиться в плаваючому стані



E	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1

Шини з трьома станами

- Стан високого імпедансу використовується для створення шин
 - Декілька мікросхем підключені до шини
 - Однак активною в деякий момент часу може бути одна і тільки одна шина



Карти Карно

- Булеві вирази можна спростити шляхом комбінування термів
- Карти Карно дозволяють наглядно мінімізувати вирази $PA + P\bar{A} = P$

Таблиця істинності

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Карта Карно

Y \ AB \ C		AB			
		00	01	11	10
C	0	1	0	0	0
	1	1	0	0	0

Карта Карно з мінтермами

Y \ AB \ C		AB			
		00	01	11	10
C	0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$AB\bar{C}$	$A\bar{B}\bar{C}$
	1	$\bar{A}\bar{B}C$	$\bar{A}BC$	ABC	$A\bar{B}C$

З використанням булевої алгебри отримаємо $Y = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C = \bar{A}\bar{B}(\bar{C} + C) = \bar{A}\bar{B}$

Мінімізація за допомогою карти Карно

- Обвести овалом 1 в сусідніх квадратах, які утворюють квадрат або прямокутник
- В булевий вираз включаються тільки ті літерали, пряма і інверсна форма яких не попадає в овал

<i>A</i>	<i>B</i>	<i>C</i>	<i>Y</i>
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

<i>Y</i> <i>AB</i>		00	01	11	10
<i>C</i>	0	1	0	0	0
1	1	0	0	0	0

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

Побудова карти Карно

- **Доповнення:** змінна з рискою над іменем

$\overline{A}, \overline{B}, \overline{C}$

- **Літерал:** Літерал: змінна або її доповнення

$A, \overline{A}, B, \overline{B}, C, \overline{C}$

- **Імпліканта:** добуток літералів

$\overline{A}\overline{B}C, A\overline{C}, BC$

- **Первинна імпліканта:** імпліканта, яка відповідає найбільшому овалу на карті Карно

Правила карт Карно

- Кожна 1 має входити хоча б в один овал
- Кожний овал має охоплювати блок, число кліток якого в кожному напрямку дорівнює степені двійки (тобто 1, 2 або 4)
- Кожний овал має бути настільки великим, наскільки це можливо
- Овал може зв'язувати краї карти Карно
- Довільні значення (X) можуть входити в овал, якщо це допомагає мінімізувати вираз

Карта Карно – три входи

Y \ C \ AB		00	01	11	10
C	0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$AB\bar{C}$	$A\bar{B}\bar{C}$
	1	$\bar{A}\bar{B}C$	$\bar{A}BC$	ABC	$A\bar{B}C$

Truth Table

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

K-Map

Y \ C \ AB		00	01	11	10
C	0	0	1	1	0
	1	0	1	0	0

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

Карти Карно – чотири входи

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

Y		AB			
CD		00	01	11	10
		00	01	11	10
00		1	0	0	1
01		0	1	0	1
11		1	1	0	0
10		1	1	0	1

$$Y = \bar{A}C + \bar{A}BD + A\bar{B}\bar{C} + \bar{B}\bar{D}$$

Карти Карно і довільні значення

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>Y</i>
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	X
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Y CD		AB			
		00	01	11	10
CD	00	1	0	X	1
	01	0	X	X	1
	11	1	1	X	X
	10	1	1	X	X

$$Y = A + \overline{B}\overline{D} + C$$

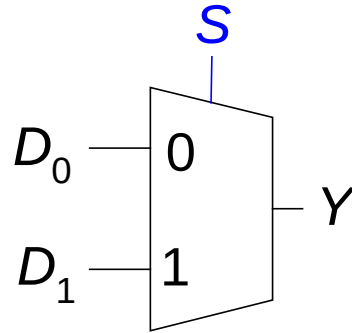
Базові комбінаційні блоки

- Мультиплексори
- Дешифратори

Мультиплексор

- Вибирає один з N входів і з'єднує його з виходом
- $\log_2 N$ -біт для вибору входу – **S**-вхід керування (вибору)
- **Приклад:**

2:1 Mux



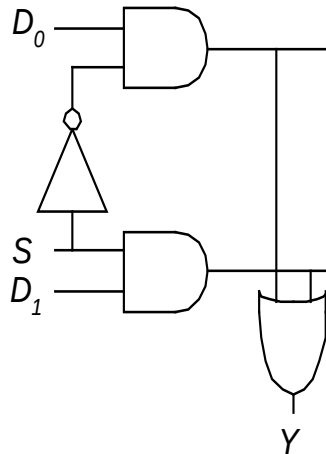
S	D_1	D_0	Y	S	Y
0	0	0	0	0	D_0
0	0	1	1	1	D_1
0	1	0	0		
0	1	1	1		
1	0	0	0		
1	0	1	0		
1	1	0	1		
1	1	1	1		

2-входовий мультиплексор

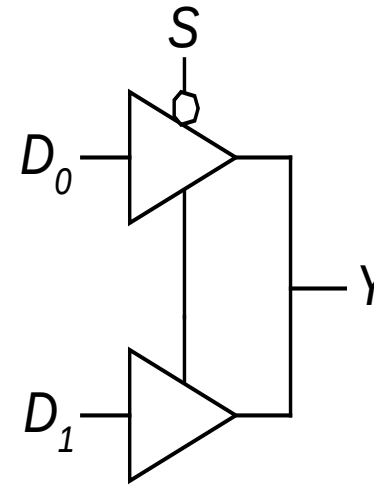
- 3 логічних елементів
 - Диз'юнктивна форма

Y S	D ₀ D ₁		00	01	11	10
	0	1	0	0	1	1
1	0	1	0	1	1	0

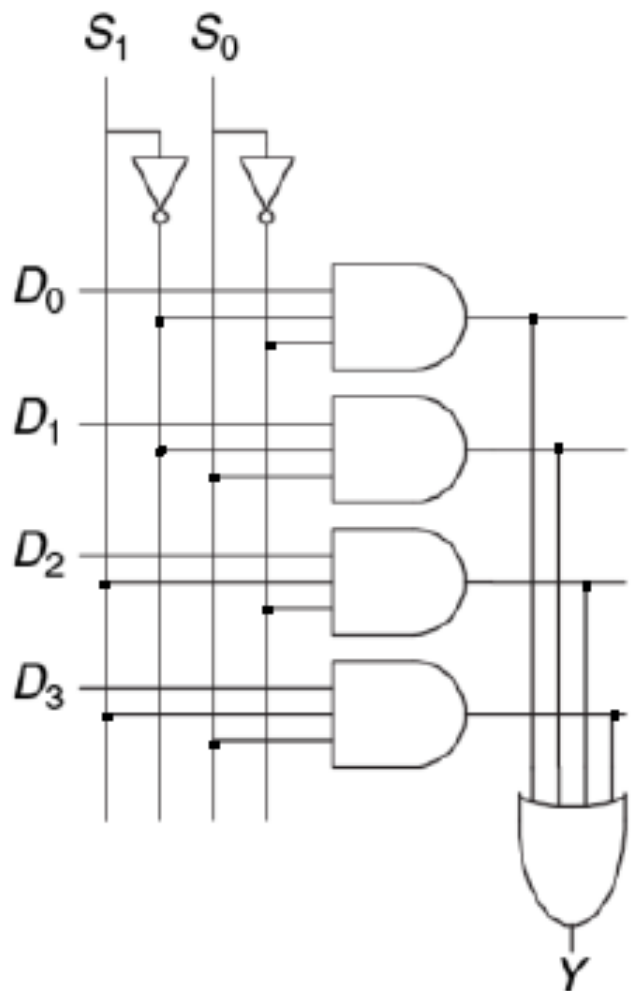
$$Y = D_0 \bar{S} + D_1 S$$



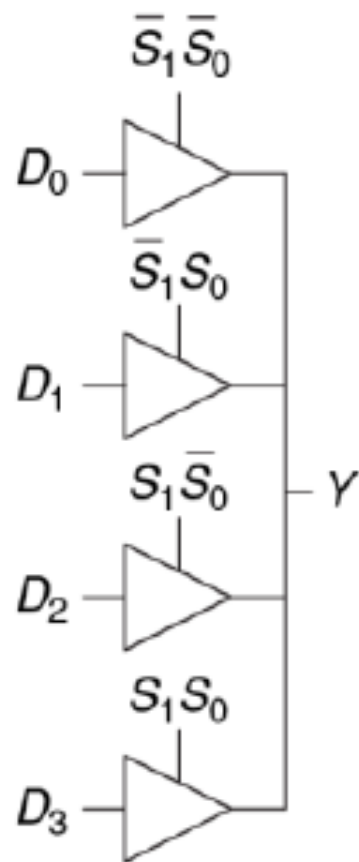
- 3 буферів з третім станом
 - Для 2-входового мультиплексора використовується 2 буфери з третім станом
 - Один і тільки один з них включається для вибору відповідного входу



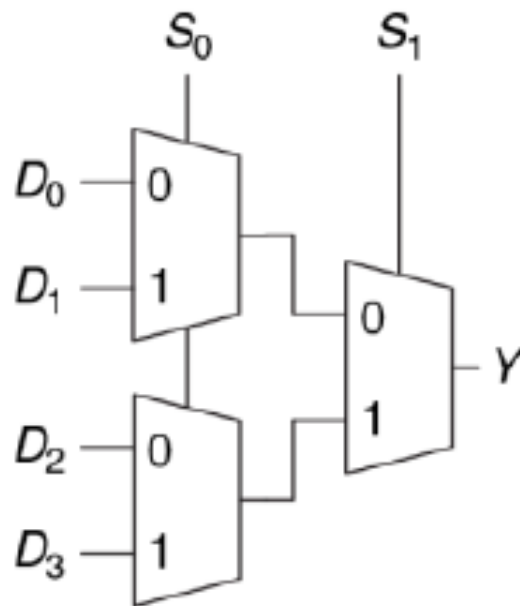
4-входовий мультиплексор



a



b



c

Реалізація мультиплексора:

а – дворівнева логіка

б – буфери з третім станом

с – ієрархічна

Використання мультиплексора як таблиці перетворень (lookup table)

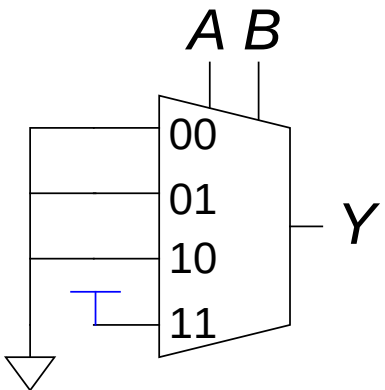
a – 2-входовий елемент AND з 4-входового мультиплексора

b – 3-входовий логічний елемент з 8-входового мультиплексора

a

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

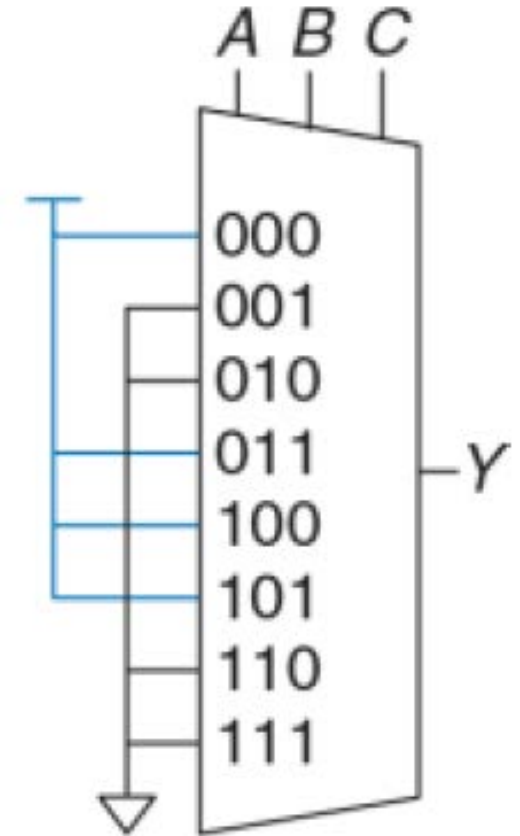
$$Y = AB$$



b

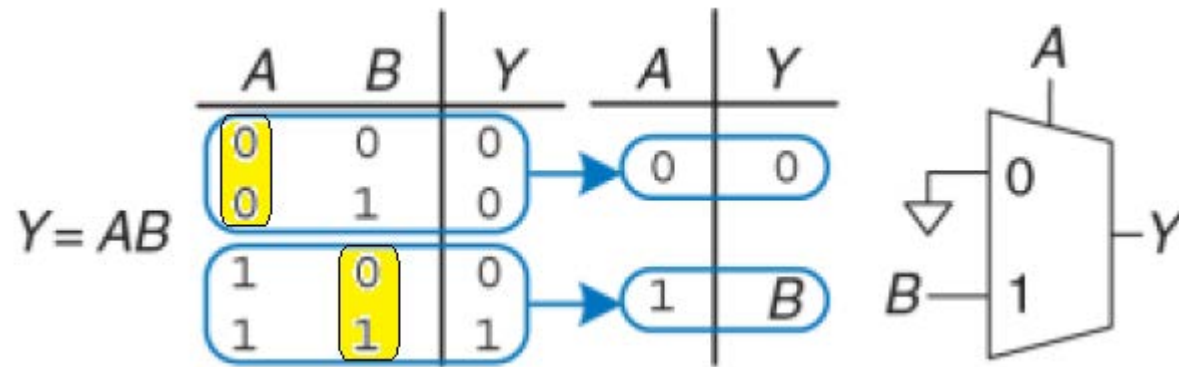
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

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

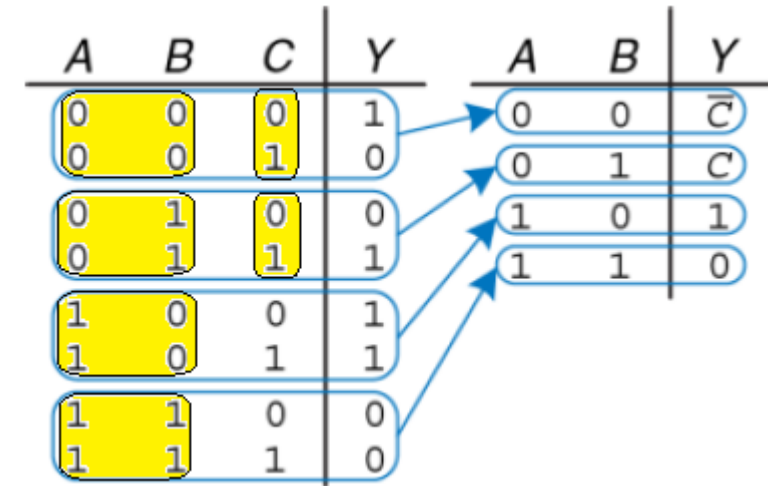


Зменшення розміру мультиплексора при реалізації таблиці перетворень

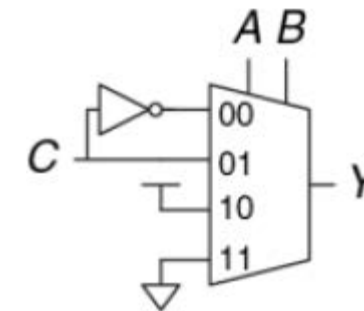
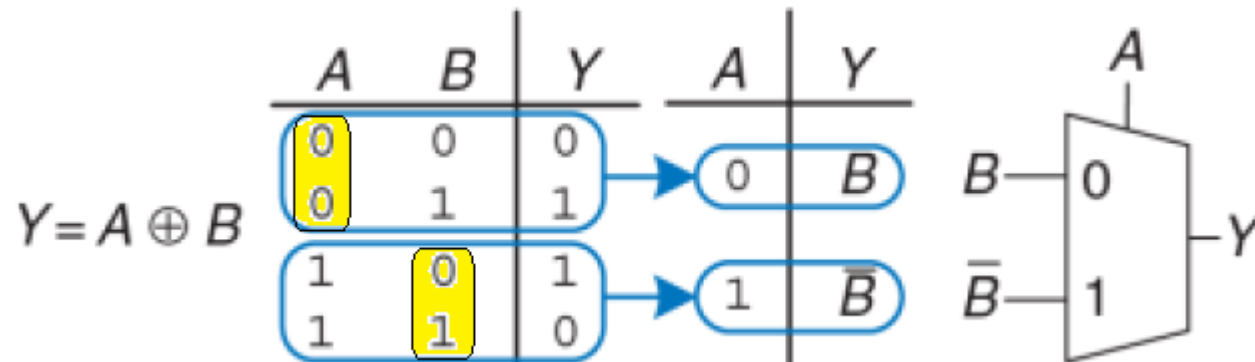
2-входовий AND



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

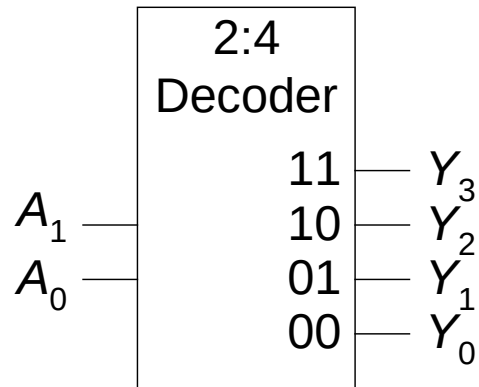


2-входовий XOR



Дешифратор (decoder)

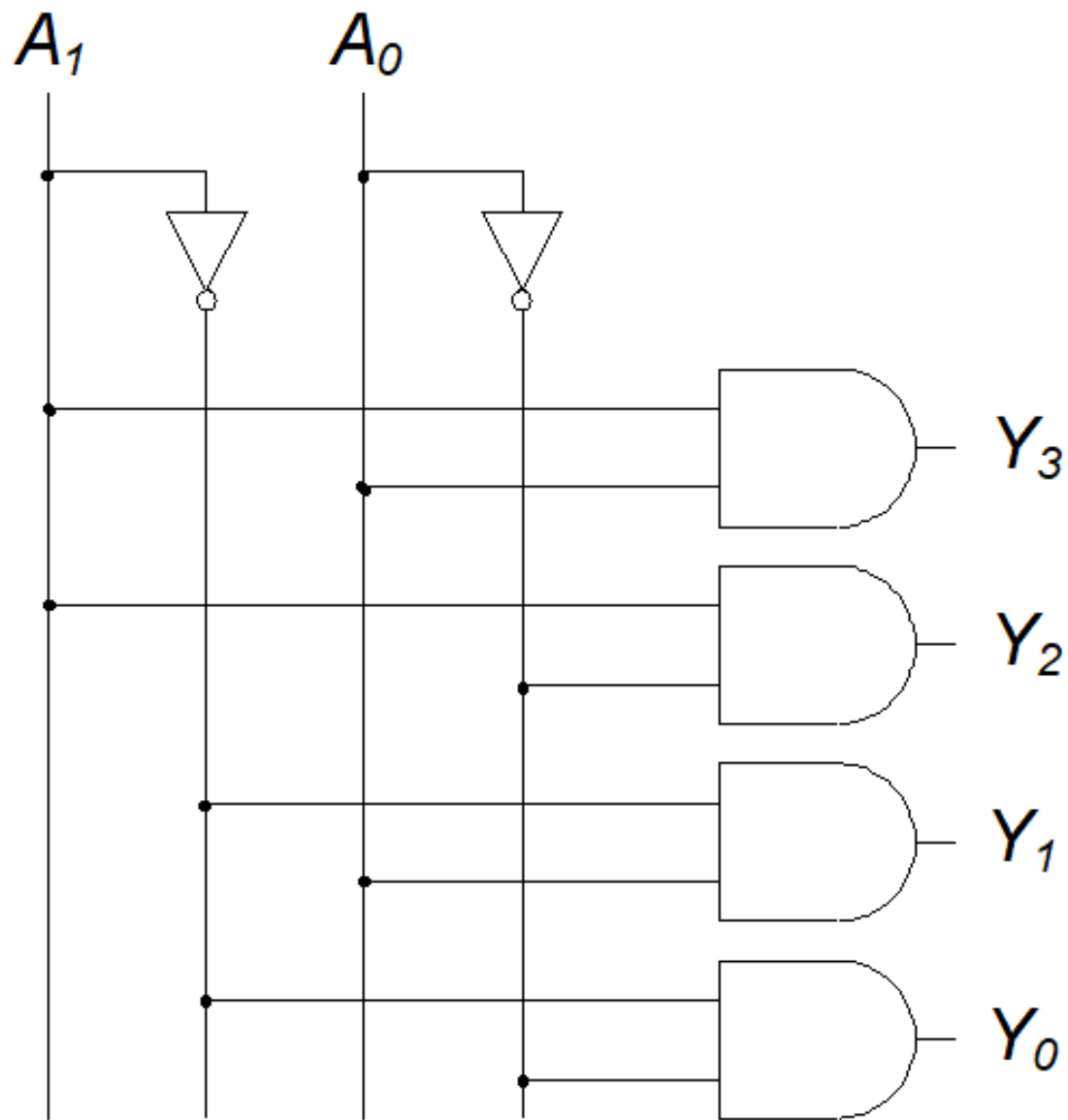
- N входів, 2^N виходів
- Видає 1 на один з виходів в залежності від набору входних значень
- Прямий унітарний код (one-hot code): тільки один вихід приймає значення ІСТИНА



A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

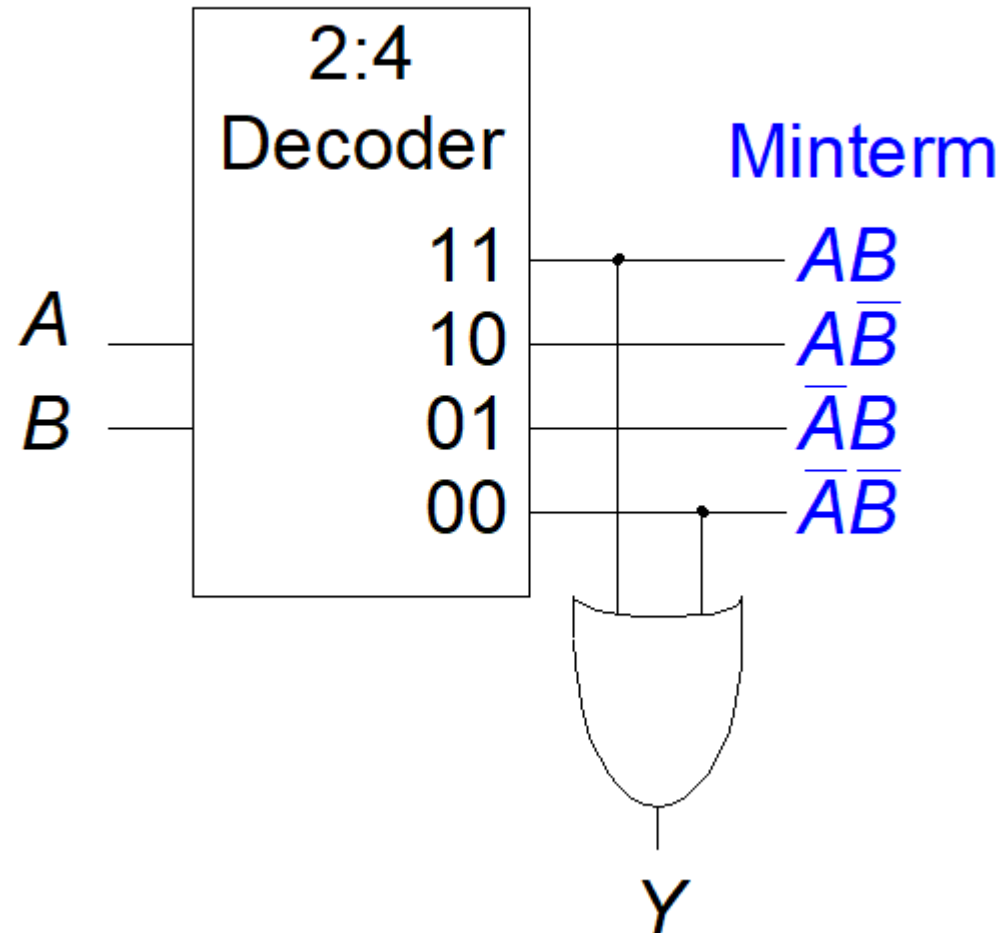
Реалізація дешифратора 2:4

A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



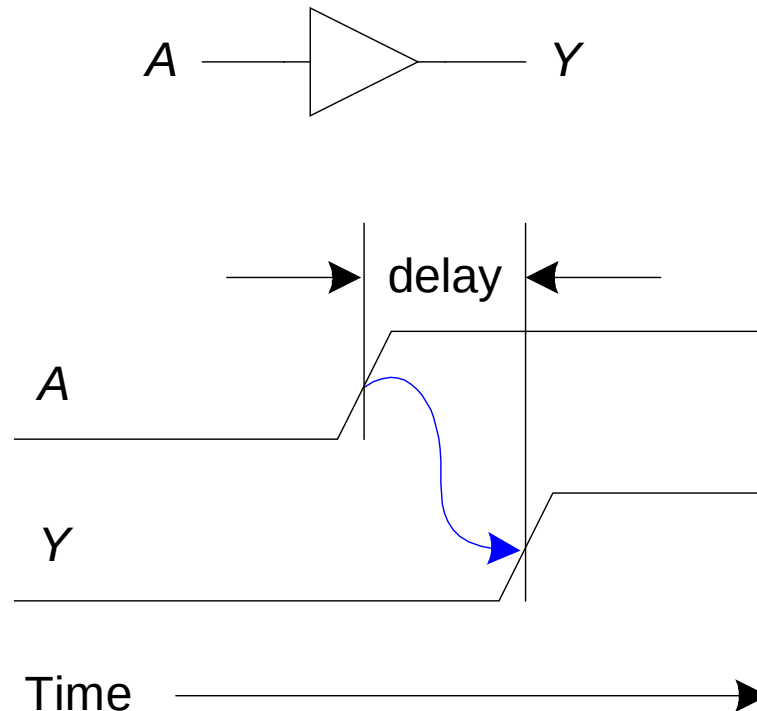
Логічні функції на основі дешифраторів

$$Y = AB + \overline{A}\overline{B} = \overline{A \oplus B}$$



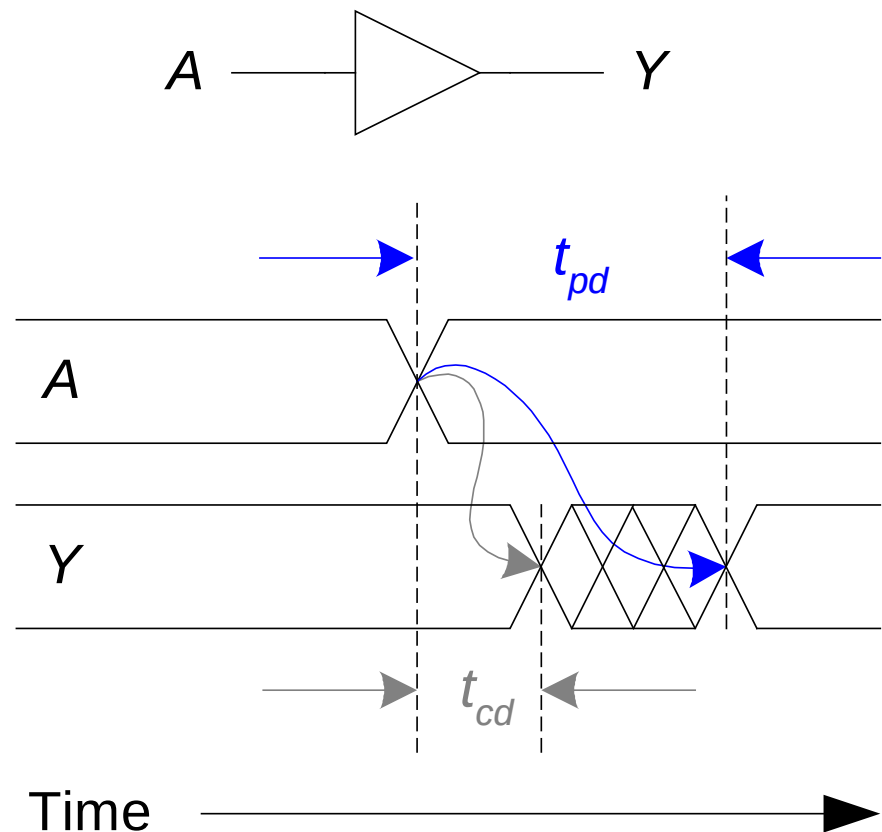
Часові характеристики

- Затримка – різниця між часом досягнення рівня 50% вхідним і вихідним сигналами
- Затримка поширення – це найбільша затримка в схемі



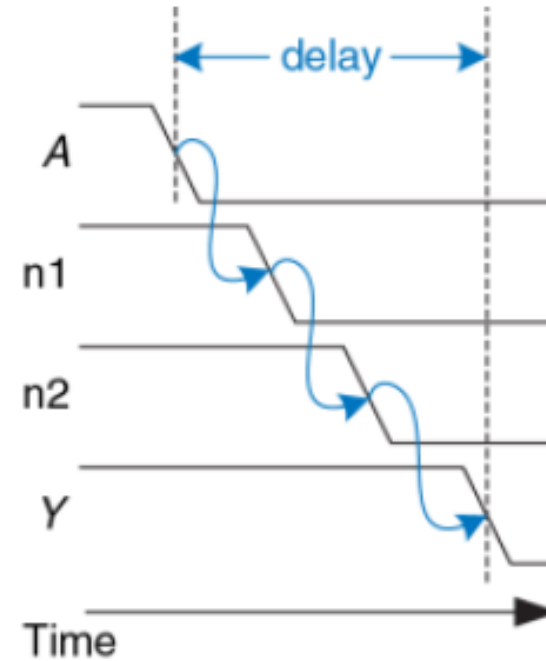
Затримки поширення і реакції

- **Затримка поширення** (propagation delay) t_{pd} – це максимальний час від початку зміни входу до моменту коли всі виходи досягнуть встановлених станів
- **Затримка реакції** (contamination delay) t_{cd} - це мінімальний час від початку зміни входу до моменту коли будь-який із виходів почне змінювати своє значення

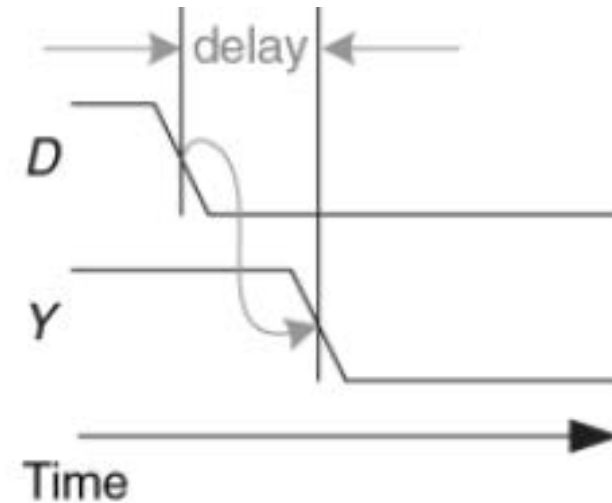
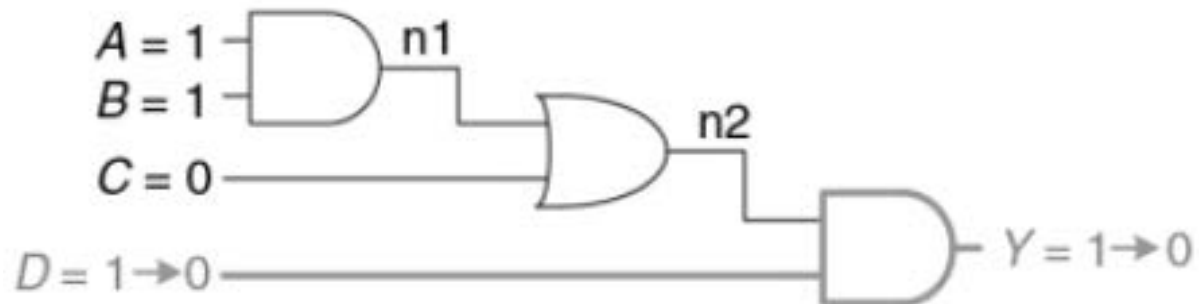


Критичні і найкоротші шляхи

Critical Path $t_{pd} = 2t_{pd_AND} + t_{pd_OR}$

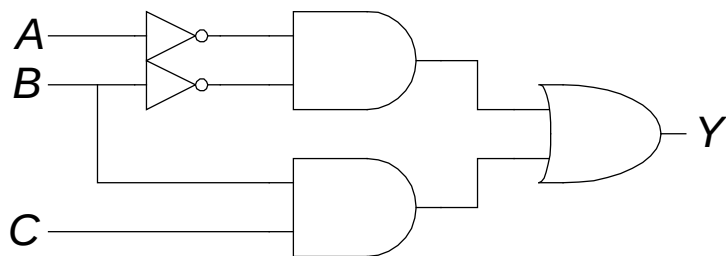


Short Path $t_{cd} = t_{cd_AND}$



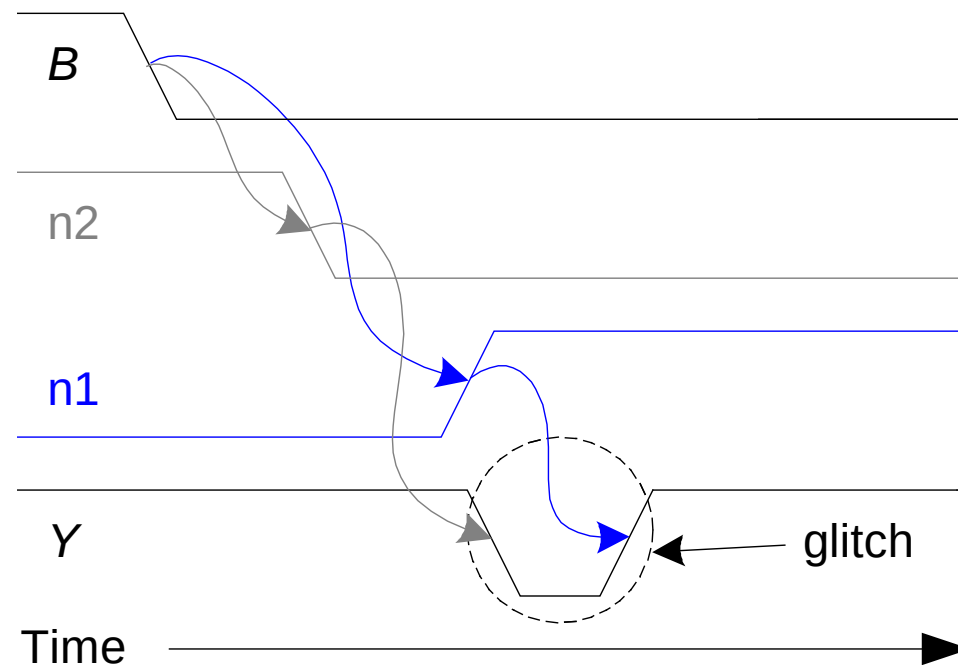
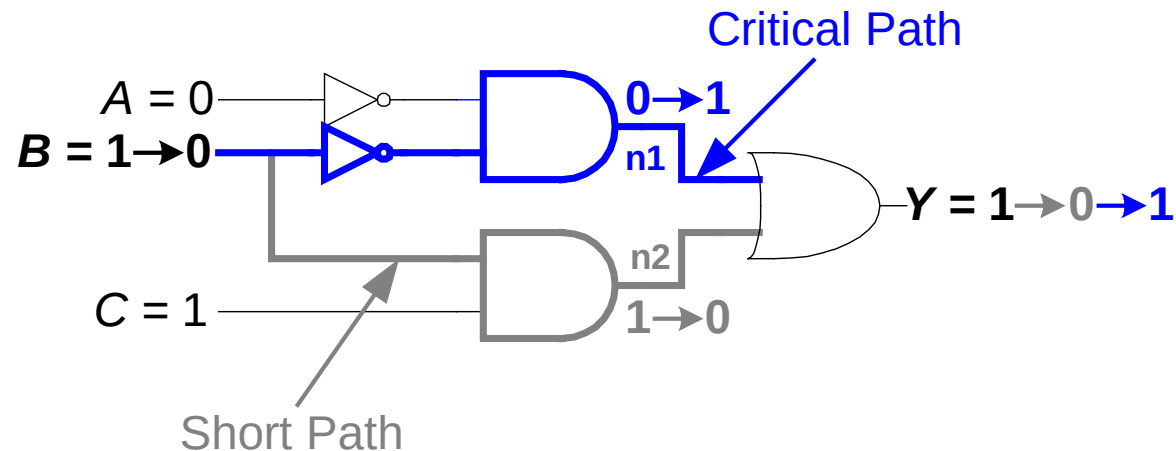
Паразитний імпульс

- Що відбувається коли $A = 0$, $C = 1$, а B змінюється з 1 до 0? Одиночна зміна B спричиняє декілька змін на виході

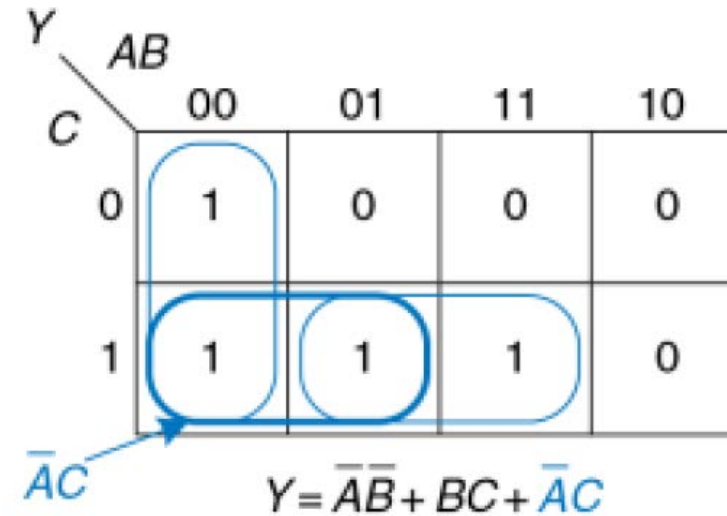
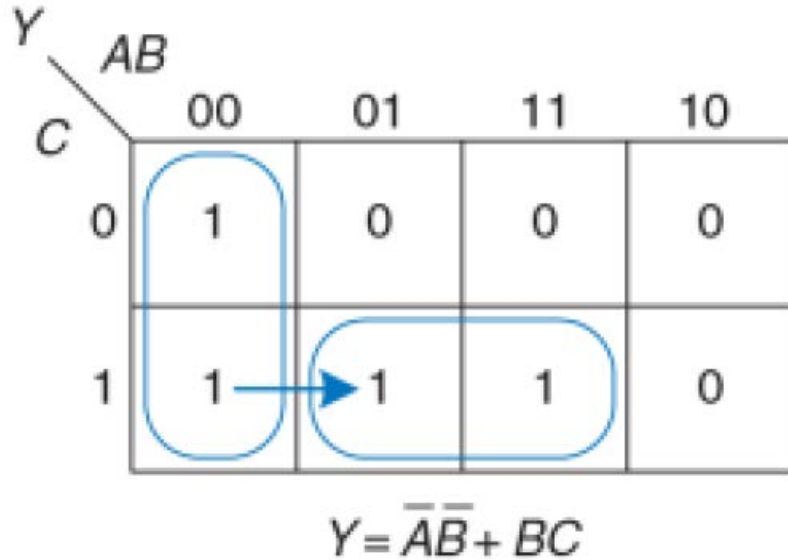


		AB			
		00	01	11	10
C	0	1	0	0	0
	1	1	1	1	0

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



Усунення паразитного імпульсу



- Паразитний імпульс виникає, коли одна змінна перетинає границю між двома первинними імплікантами на карті Карно
- Цей імпульс можна усунути додаванням додаткових імплікант у карту Карно так, щоб перекрити ці границі

