



О. В. Разживін, О.Ф. Єнікєєв

КОМП'ЮТЕРНА ЛОГІКА

Краматорськ 2020

Міністерство освіти і науки України
Донбаська державна машинобудівна академія

О. В. Разживін, О.Ф. Єнікєєв

КОМП'ЮТЕРНА ЛОГІКА

НАВЧАЛЬНИЙ ПОСІБНИК

для студентів денної та заочної форм навчання
спеціальності 123 "Комп'ютерна інженерія"

Краматорськ 2020

ББК 32.844
УДК 658.512.011.56
С - 32

Разживін О.В. Комп'ютерна логіка. Навчальний посібник для студентів денної та заочної форм навчання спеціальності 123 "Комп'ютерна інженерія"/ О.В. Разживін, Єнікєєв О.Ф.: Краматорськ: ДДМА, - 2020.-116с.

Дисципліна "Комп'ютерна логіка" є однією зі спеціальних профільюючих дисциплін і тому займає провідне місце у підготовці бакалаврів комп'ютерної інженерії.

Метою дисципліни "Комп'ютерна логіка" є: 1) формування компетентностей, необхідних для розроблення пристроїв, які містять комбінаційні схеми, цифрові автомати з пам'яттю та операційні автомати; 2) розвиток у студентів фахового стилю мислення; 3) надання глибоких та міцних знань з комп'ютерної логіки, необхідних для подальшого вивчення спеціальних дисциплін та для практичної інженерної діяльності; 4) вироблення у студентів вміння використовувати набуті знання при розробці пристроїв, які містять комбінаційні схеми та цифрові автомати з пам'яттю; 5) ознайомлення студентів з комп'ютерною арифметикою, необхідною для синтезу операційних автоматів.

ББК 32.844

© О.В. Разживін, 2020

© ДДМА, 2020

ЗМІСТ

1 ОСНОВНІ СВІДЧЕННЯ З ТЕОРІЇ АЛГЕБРИ ЛОГІКИ

- 1.1 Аксіоми, основні теореми та тотожності алгебри логіки
- 1.2 Властивості перемикальних функцій
- 1.3 Форми запису логічних функцій
- 1.4 Мінімізація структурних формул за допомогою методу Квайна та карт Карно
- 1.5 Методи мінімізації системи логічних функцій

2 МЕТОДИ СИНТЕЗУ ЦИФРОВИХ ПРИСТРОЇВ

- 2.1 Універсальні логічні елементи
- 2.2 Приклади побудови комбінаційних пристроїв
- 2.3 Загальна задача синтезу комбінаційних пристроїв
- 2.4 Проектування комбінаційних пристроїв, вільних від змагань сигналів
- 2.5 Засоби формального опису послідовнісних пристроїв
- 2.6 Канонічний метод проектування послідовнісних пристроїв

3 ТИПОВІ ПРИСТРОЇ СИСТЕМ КЕРУВАННЯ

- 3.1 Дешифратори, шифратори та перетворювачі кодів
- 3.2 Схеми порівняння кодів та контролю парності
- 3.3 Пристрої для підсумовування
- 3.4 Мультиплексори та демультіплексори
- 3.5 Тригери. Методи синтезу схем тригерів
- 3.6 Регістри
- 3.7 Лічильники імпульсів
- 3.8 Допоміжні цифрові пристрої

4 ЦИФРОВІ ПРИСТРОЇ ОБРОБКИ ДИСКРЕТНИХ СИГНАЛІВ СИСТЕМ

- 4.1 Цифрові пристрої формування сигналів
- 4.2 Цифрові корелятори
- 4.3 Цифрові погодженні фільтри (ЦПФ)
- 4.4 Синтез логічних схем
- 4.5 Синтез комбінаційних схем на базі комутаторів

СПИСОК СКОРОЧЕНЬ

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1 ОСНОВНІ СВДЧЕННЯ З ТЕОРІЇ АЛГЕБРИ ЛОГІКИ

1.1 Аксиоми, основні теореми та тотожності алгебри логіки

Теорія роботи пристроїв цифрової обробки сигналів базується на рівняннях алгебри логіки. Змінні, якими оперує алгебра логіки, приймають тільки два значення: 0 та 1. Усі змінні звичайно позначають латинськими буквами x, y, z, \dots

В алгебрі логіки є такі основні визначення для опису роботи цифрового пристрою: відношення еквівалентності ($=$); операція диз'юнкції (АБО), що позначається знаком \vee або $+$; операція кон'юнкції (І), що позначається знаком \wedge або точкою, яку можна опустити (наприклад, xu); операція заперечення (НІ), що позначається рискою над змінною (наприклад, \bar{x}); а також операція логічної нерівнозначності (виключаюче АБО), що позначається знаком \oplus .

Відношення еквівалентності задовольняє наступним властивостям:

- 1) $x = x$ – рефлексивність;
- 2) якщо $x = y$, то $y = x$ – симетричність;
- 3) якщо $x = y$ та $y = z$, то $x = z$ – транзитивність.

Алгебра логіки при мінімізації структурної формули цифрового пристрою оперує такими аксіомами:

$$\left. \begin{array}{l} x = 0, \quad \text{якщо } x \neq 1 \\ x = 1, \quad \text{якщо } x \neq 0 \end{array} \right\}, \quad (1.1)$$

$$\left. \begin{array}{l} 1 + 1 = 1 \\ 0 \cdot 0 = 0 \end{array} \right\}, \quad (1.2)$$

$$\left. \begin{array}{l} 0 + 0 = 0 \\ 1 \cdot 1 = 1 \end{array} \right\}, \quad (1.3)$$

$$\left. \begin{array}{l} 0 + 1 = 1 + 0 = 1 \\ 1 \cdot 0 = 0 \cdot 1 = 0 \end{array} \right\}, \quad (1.4)$$

$$\left. \begin{array}{l} \bar{0} = 1 \\ \bar{1} = 0 \end{array} \right\}. \quad (1.5)$$

Аксиома (1.1) стверджує, що змінні приймають тільки два значення. Аксиоми (1.2) ... (1.4) визначають операції диз'юнкції і кон'юнкції, аксіома (1.5) визначає операцію заперечення.

За допомогою аксіом алгебри логіки можна довести цілий ряд теорем (законів):

- ідемпотентні закони

$$x + x = x, \quad x \cdot x = x;$$

- комутативні закони

$$x + y = y + x, \quad xy = yx;$$

- асоціативні закони

$$(x + y) + z = x + (y + z), \quad (xy)z = x(yz);$$

- дистрибутивні закони

$$x(y + z) = xy + xz, \quad x + yz = (x + y)(x + z);$$

- закони заперечення

$$x + \bar{x} = 1, \quad x\bar{x} = 0;$$

$$0 + x = x, \quad 1x = x;$$

$$1 + x = 1, \quad 0x = 0;$$

- закон подвійності (теорема де Моргана)

$$\overline{x + y} = \bar{x}\bar{y}, \quad \overline{xy} = \bar{x} + \bar{y};$$

- закон подвійного заперечення

$$\overline{\bar{x}} = x;$$

- закон поглинання

$$x + x\bar{0} = x, \quad x(x + 1) = x;$$

- операція склеювання

$$xy + \bar{x}y = y, \quad (x + y)(\bar{x} + \bar{y}) = \bar{x} + y;$$

- операція узагальненого склеювання

$$xy + \bar{x}z + yz = x\bar{y} + \bar{x}z,$$

$$(x + y)(\bar{x} + z)(y + z) = (x + y)(\bar{x} + z),$$

$$x + \bar{x}y = x + y, \quad x(\bar{x} + y) = xy.$$

Теорема дозволяють спростити логічні вирази, а також перетворити їх в таку форму, яка буде більш сприятлива для реалізації в логічних пристроях.

1.2 Властивості перемикальних функцій

Сукупність змінних $x_n \dots x_1$ можна розглянути як n – мірний вектор, який описує роботу цифрового пристрою, а також як область визначення функції. Для того, щоб задати функцію, треба вказати її значення (0,1) у всіх точках області визначення. При цьому кількість різних станів функції для n змінних дорівнюватиме 2^n . Якщо функція залежить не від усіх змінних, то вона зветься виродженою. Найбільше поширення знайшли не вироджені функції двох змінних $f(x_2, x_1)$ за такими назвами операції алгебри логіки

$$f(x_2, x_1) = x_2 + x_1 \quad (\text{АБО});$$

$$f(x_2, x_1) = x_2 x_1 \quad (\text{І});$$

$$f(x_2, x_1) = \overline{x_2 x_1} \quad (\text{І – НІ});$$

$$f(x_2, x_1) = \overline{x_2 + x_1} \quad (\text{АБО} - \text{НІ});$$

$$f(x_2, x_1) = x_2 \oplus x_1 \quad (\text{сума за модулем два}).$$

Область визначення цих функцій становить чотири точки, ($2^2 = 4$), значення функцій наведені в табл. 1.1.

Таблиця 1.1 – Область визначення функцій

i	x_2	x_1	$x_2 + x_1$	$x_2 \cdot x_1$	$\overline{x_2 \cdot x_1}$	$\overline{x_2 + x_1}$	$x_2 \oplus x_1$
0	0	0	0	0	1	1	0
1	0	1	1	0	1	1	1
2	1	0	1	0	1	1	1
3	1	1	1	1	0	0	0

Використовуючи тільки функції двох змінних, можна побудувати функції більшого числа змінних шляхом композиції, тобто підстановкою одних функцій замість змінних в інші функції. Така підстановка можлива, тому що області значень функцій та змінних співпадають (0 або 1).

1.3 Форми запису логічних функцій

Логічні функції (ЛФ), які у алгебрі логіки описують роботу цифрового пристрою, звичайно задаються або в табличній формі (таблицями істинності), або логічними виразами. *Таблиця істинності* – це перерахування усіх можливих станів вхідних та вихідних сигналів, які розташовано у порядку зростання вхідних (наприклад, табл. 1.2).

Таблиця 1.2 – Таблиця істинності функції

x_2	x_1	x_0	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Логічна функція роботи цифрового пристрою виражається за допомогою структурної формули, яка містить логічні операції над незалежними змінними. Ці змінні розташовано в порядку зростання станів вхідного сигналу. Завдання логічної функції за допомогою структурної формули дозволяє побудувати схему пристрою, оцінити складність її реалізації та мінімізувати кількість елементів заданого типу. Мінімізація

кількості елементів спрощує електричну принципову схему цифрового пристрою.

Існують дві форми запису структурної формули, які дають однозначне її уявлення:

- у вигляді звершеної диз'юнктивної нормальної форми (ЗДНФ);
- у вигляді звершеної кон'юнктивної нормальної форми (ЗКНФ).

Запис структурної формули у вигляді ЗДНФ. Розглянемо на прикладі правило переходу від таблиці істинності ЛФ до її запису у вигляді ЗДНФ. Для $y = f(x_1, x_2, x_3)$ введемо в таблицю істинності три стовпці допоміжних функцій y_1, y_2, y_3 (табл. 1.3)

Таблиця 1.3 – Таблиця істинності функції

x_3	x_2	x_1	y_1	y_2	y_3	y
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	1	1
1	0	1	0	0	0	0
1	1	0	0	1	0	1
1	1	1	1	0	0	1

Кількість допоміжних функцій, які утворюють вихідну функцію y , відповідає числу одиниць у таблиці істинності. Структура допоміжних функцій у цьому випадку буде наступна: кожному значенню $y = 1$ в стовпці y_i відповідає одне значення допоміжної функції. При цьому сама допоміжна функція записується як кон'юнкція одиничних станів вхідного сигналу. Якщо стан вхідного сигналу дорівнює 0, то y у виразі допоміжної функції записується його інверсне значення. Функції такого виду називають повною елементарною кон'юнкцією, мінтермом чи континуентою одиниці. За даними табл. 1.3 можна скласти формули для $y_i = f_i(x_1, x_2, x_3)$, $y = f(y_1, y_2, y_3)$ а також електричну принципову схему, яка реалізує задану логічну функцію.

Запишемо значення основної та допоміжних функцій

$$y_1 = x_1 \cdot x_2 \cdot x_3, \quad y_2 = \bar{x}_1 \cdot x_2 \cdot x_3, \quad y_3 = \bar{x}_1 \cdot \bar{x}_2 \cdot x_3;$$

$$y = y_1 + y_2 + y_3 = x_1 \cdot x_2 \cdot x_3 + \bar{x}_1 \cdot x_2 \cdot x_3 + \bar{x}_1 \cdot \bar{x}_2 \cdot x_3.$$

Отримана структурна формула є ЗДНФ для функції y .

Звершеною називають функцію, у якій добутки (суми) містять всі аргументи, причому кожен аргумент входить тільки один раз у прямому чи інверсному вигляді. Отже, до властивостей ЗДНФ варто віднести наступне: у ЗДНФ немає двох однакових добутків, жоден добуток не містить двох однакових множників і жоден добуток не містить разом з незалежною змінною її інверсію.

Для запису структурної формули у вигляді ЗДНФ за таблицею

істинності необхідно:

- визначити рядки, де $y = 1$;
- для кожного визначеного рядка скласти логічний добуток усіх незалежних змінних, причому, якщо у виділеному рядку незалежна змінна дорівнює 1, то вона входить у відповідний добуток без інверсії, а якщо дорівнює 0 – з інверсією;

- скласти логічну суму отриманих добутоків.

Сформульоване правило дозволяє записати структурну формулу без використання допоміжних функцій.

Запис структурної формули у вигляді ЗКНФ. Інший спосіб запису структурної формули за таблицею істинності може бути поданий у вигляді звершеної кон'юнктивної нормальної форми (ЗКНФ). Різниця полягає в тому, що число введених допоміжних функцій в табл. 1.4 відповідає числу нулів для функції y .

Допоміжні функції y_i будуються таким чином, щоб значенню $y = 0$ відповідало значення $y_i = 0$ при наявності в кожному стовпці семи одиниць. Такі функції називають макстермом або континуентою нуля.

Для запису структурної формули у вигляді ЗКНФ в таблиці істинності потрібно:

- визначити всі рядки, де $y = 0$;
- для кожного визначеного рядка скласти логічну суму всіх незалежних змінних, причому, якщо у визначеному рядку незалежна змінна дорівнює 0, то вона входить у вказану суму без інверсії, якщо дорівнює 1 – з інверсією;
- скласти логічний добуток всіх отриманих раніше сум.

Таблиця 1.4 – Таблиця істинності функції

x_3	x_2	x_1	y_1	y_2	y_3	y
0	0	0	0	1	1	0
0	0	1	1	0	1	0
0	1	0	1	1	1	1
0	1	1	1	1	0	0
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

З аналізу станів вхідних сигналів табл. 1.4 одержимо такі вирази для рівнянь алгебри логіки

$$y_1 = x_1 + x_2 + x_3, \quad y_2 = \bar{x}_1 + x_2 + x_3, \quad y_3 = \bar{x}_1 + \bar{x}_2 + x_3;$$
$$y = y_1 \cdot y_2 \cdot y_3 = (x_1 + x_2 + x_3) \cdot (\bar{x}_1 + x_2 + x_3) \cdot (\bar{x}_1 + \bar{x}_2 + x_3).$$

Структурні формули у вигляді ЗДНФ та ЗКНФ еквівалентні та можуть бути перетворені одна в іншу за допомогою законів та тотожностей алгебри логіки [1... 3].

Застосування законів та тотожностей дає можливість виконувати еквівалентні перетворення та спрощувати структурні формули ЛФ, тобто знаходити для них більш прості вирази. Прикладами еквівалентних перетворень є операції склеювання та поглинання. Так, для ЗДНФ, виконавши еквівалентні перетворення структурних формул, одержимо наступні вирази

$$\begin{aligned} A \cdot B + A \cdot \bar{B} &= A \cdot (B + \bar{B}) = A \cdot 1 = A; \\ A \cdot B \cdot D + A \cdot \overline{B \cdot D} &= A \cdot (B \cdot D + \overline{B \cdot D}) = A; \\ A \cdot B \cdot D + A \cdot \bar{B} \cdot \bar{D} &= A \cdot (B \cdot D + \bar{B} \cdot \bar{D}) \neq A. \end{aligned}$$

Операція склеювання двох додатків виконується тоді, коли додатки мають загальний множник. Інший співмножник в один додаток входить в інверсному вигляді, в другий – у прямому. Результатом операції склеювання є загальний множник. У випадку, коли перший додаток входить як співмножник у другий додаток, то виконується операція поглинання. У результаті цієї операції перший додаток

$$A + A \cdot B = A \cdot (1 + B) = A \cdot 1 = A.$$

Для ЗКНФ після еквівалентних перетворень виразу алгебри логіки одержимо таке

$$(A + B) \cdot (A + \bar{B}) = A \cdot A + B \cdot A + A \cdot \bar{B} + B \cdot \bar{B} = A \cdot (1 + B + \bar{B}) = A.$$

Якщо маємо два співмножники, кожний з яких являє собою логічну суму, причому в цих сумах є однакові складові, а неоднакові складові інверсні, то здійснюється операція склеювання. Результатом цієї операції є загальна частина. Якщо один із двох співмножників входить в іншій як додаток, то виконується операція поглинання. У результаті чого залишається загальна частина співмножників

$$A \cdot (A + B) = A + A \cdot B = A.$$

Використовуючи аналогічні перетворення, можна записати результат поглинання заперечення

$$\begin{aligned} A + \bar{A} \cdot B &= A + B, & \bar{A} + A \cdot B &= \bar{A} + B; \\ A \cdot (\bar{A} + B) &= A \cdot B, & \bar{A} \cdot (A + B) &= \bar{A} \cdot B. \end{aligned}$$

1.4 Мінімізація структурних формул за допомогою методу Квайна та карт Карно

Метою мінімізації є одержання еквівалентної структурної формули, що відповідає найменшій кількості логічних елементів у схемі. Досягнення мінімізованої форми ЛФ ґрунтується на еквівалентних (тотожних) перетвореннях структурних формул для ЛФ.

При мінімізації на основі карт Карно використовуються ті ж операції, що і при мінімізації за методом Квайна, але формою подання вихідних даних є карта з числом клітинок 2^n , де n – кількість аргументів функції. Кожна карта відповідає одній функції, число n якої визначає розміри карти. Ознакою виконання операції склеювання є розташування в сусідніх клітинках одиниць. Відзначимо, що сусідніми вважаються також і крайні за вертикаллю і горизонталлю клітки, оскільки карта може бути згорнута в циліндр.

Кarti для трьох функцій $y_1 = \bar{x}_2 \cdot \bar{x}_3$; $y_2 = \bar{x}_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot x_3$; $y_3 = x_1 \cdot \bar{x}_2 + x_2 \cdot x_3$ наведені на рис. 1.1.

	00	01	11	10	
\bar{x}_3 0	1	1			<i>a</i>
x_3 1					

	00	01	11	10	
\bar{x}_3 0	1	1		1	<i>б</i>
x_3 1			1		

	00	01	11	10	
\bar{x}_3 0		1			<i>в</i>
x_3 1		1	1	1	

Рисунок 1.1 – Кarti Карно:
a – для функції y_1 ; *б* – для функції y_2 ; *в* – для функції y_3

Операції склеювання показані замкненим контуром, який охоплює сусідні одиниці. Відповідно до карти одержимо мінімізовану структурну формулу, яка еквівалентна вихідній. Виконання операції склеювання приводить до поглинання змінних із різними значеннями.

Правила склеювання за допомогою карт Карно:

- при склеюванні двох сусідніх одиниць у виразі допоміжної функції випадає одна вхідна змінна, яка змінює свій стан з 1 на 0 або з 0 на 1;
- при склеюванні чотирьох сусідніх одиниць, які утворюють строку, стовпець або квадрат, у виразі допоміжної функції випадає дві вхідні змінні;
- при склеюванні восьми сусідніх одиниць у виразі допоміжної функції випадає три вхідні змінні.

При мінімізації за методом Квайна використовуються математичні перетворення логічної функції та операції склеювання й поглинання. Розглянемо три функції, які задано таблицею істинності (табл. 1.5).

Таблиця 1.5 – Таблиця істинності функції

x_3	x_2	x_1	y_1	y_2	y_3
0	0	0	1	1	0
0	0	1	1	1	1
0	1	0	0	1	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	0	1	1

Запишемо їх у вигляді ЗДНФ та математичними перетвореннями спростимо формули, використовуючи також операції склеювання та поглинання. Для першої функції одержимо

$$y_1 = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 = \bar{x}_2 \cdot \bar{x}_3 \cdot (\bar{x}_1 + x_1) = \bar{x}_2 \cdot \bar{x}_3.$$

Функція y_2 у виді ЗДНФ після математичних перетворень набуває наступного вигляду

$$y_2 = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot x_3.$$

Виконуючи можливі операції склеювання, подамо y_2 у формі

$$y_2 = \bar{x}_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot x_3.$$

Аналогічно, для y_3 маємо

$$\begin{aligned} y_3 &= x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + x_1 \cdot \bar{x}_2 \cdot x_3 + \bar{x}_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot x_3 = \\ &= x_1 \cdot \bar{x}_2 + x_1 \cdot x_3 + x_2 \cdot x_3 = x_1 \cdot \bar{x}_2 + x_2 \cdot x_3. \end{aligned}$$

На цьому прикладі очевидні недоліки методу Квайна. По-перше, при складній функції зростає кількість можливих варіантів склеювань та поглинань, а по-друге, можуть з'являтися зайві операції склеювання. З цього можна зробити висновок, що для функцій трьох-чотирьох змінних варто використовувати карти Карно, які забезпечують просту мінімізацію. У випадку ж більшого числа змінних (п'ять-шість) варто застосовувати метод Квайна.

Для одержання мінімізованої функції у кон'юнктивній формі використовують такі операції та прийоми, які описано вище для диз'юнктивних форм. Проілюструємо їхнє застосування на прикладах. За допомогою методу Квайна проведемо мінімізацію структурних формул логічних функцій, які задано у табл. 1.6. Приймаючи до уваги ту обставину, що $(A + B)(A + \bar{B}) = A$, запишемо таке

$$\begin{aligned} y_1 &= (x_1 + x_2 + x_3) \cdot (\bar{x}_1 + x_2 + x_3) = x_2 + x_3; \\ y_2 &= (\bar{x}_1 + \bar{x}_2 + \bar{x}_3) \cdot (\bar{x}_1 + x_2 + \bar{x}_3) \cdot (x_1 + x_2 + \bar{x}_3) \cdot (\bar{x}_1 + \bar{x}_2 + x_3) = \\ &= (\bar{x}_1 + x_2) \cdot (\bar{x}_2 + \bar{x}_3). \end{aligned}$$

Ті ж функції y_1 та y_2 можна мінімізувати за допомогою карт Карно (див. рис. 1.2).

Таблиця 1.6 – Таблиця істинності функції

x_3	x_2	x_1	y_1	y_2
0	0	0	0	1
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	0
1	1	0	1	0
1	1	1	1	0

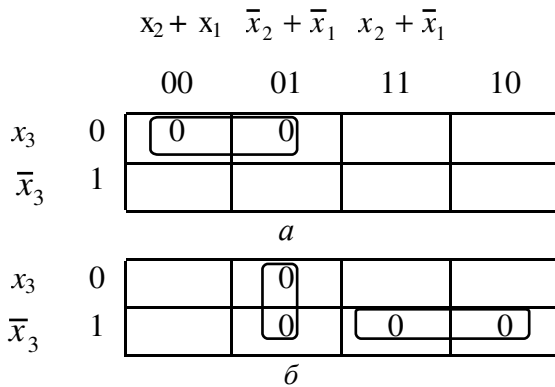


Рисунок 1.2 – Мінімізація функцій y_1 та y_2 за допомогою карт Карно:
a – логічна функція y_1 ; *б* – логічна функція y_2

Додаткові прийоми мінімізації структурної формули дозволяють у ряді випадків істотно скоротити структурну формулу. Ці прийоми припускають:

- застосування закону заперечення

$$\overline{x_1 \cdot x_2} = \bar{x}_1 + \bar{x}_2;$$

- використання розподільного закону

$$x_1 \cdot x_2 + x_1 \cdot x_3 = x_1 \cdot (x_2 + x_3);$$

- додавання у вираз для структурної формули додатків, тотожно рівних нулю

$$\begin{aligned} y &= x_1 \cdot \bar{x}_2 + \bar{x}_1 \cdot x_2 = x_1 \cdot \bar{x}_2 + \bar{x}_1 \cdot x_2 + x_1 \cdot \bar{x}_1 + x_2 \cdot \bar{x}_2 = \\ &= x_1 \cdot (\bar{x}_2 + \bar{x}_1) + x_2 \cdot (\bar{x}_2 + \bar{x}_1) = (x_1 + x_2) \cdot (\bar{x}_1 + \bar{x}_2) = \overline{x_1 \cdot x_2} \cdot (x_1 + x_2). \end{aligned}$$

Після додавання додатків, які тотожно рівні нулю, потім застосування двічі розподільного закону та закону заперечення, одержали структурну формулу, яка містить на один елемент ІІ менше, ніж вихідна формула.

Не цілком (частково) визначеними логічними функціями називають функції, значення яких задано лише для частини можливих наборів їхніх вхідних сигналів. Розглянемо приклад.

Приклад. У логічному пристрої (рис. 1.3) вихідний сигнал y_1 приймає значення $y_1=1$, якщо три розрядний двійковий код K на вході змінюючись від 1 до 7, буде мати значення $K=1$, а якщо $K \neq 1$, то $y_1 = 0$.

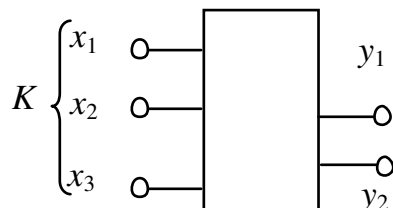


Рисунок 1.3 – Структура логічного пристрою

Заповнимо таблицю істинності для y_1 (табл. 1.7). У першому рядку можна поставити значення y_1 як 0 так і 1, і це не суперечить словесному опису. Отже, у першому рядку має місце невизначеність, що відзначається знаком Φ у таблиці. Таким чином, y_1 не цілком визначена на наборі аргументів x_1, x_2, x_3 . Вихідний сигнал y_2 приймає значення $y_2 = 1$, K приймає значення при $K = 1$ і $K = 2$, змінюючись від 1 до 5. В інших припустимих випадках $y_2 = 0$. При заповненні таблиці істинності (табл. 1.7), невизначеність y_2 відзначимо знаками Φ_1, Φ_2, Φ_3 .

Таблиця 1.7 – Таблиця істинності функції

x_3	x_2	x_1	y_1	y_2
0	0	0	Φ	Φ_1
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	Φ_2
1	1	1	0	Φ_3

Проведемо мінімізацію не цілком визначених логічних функцій, які задано у прикладі. При мінімізації функції y_1 за методом Квайна приймаємо $\Phi = 1$ та подаємо ЗДНФ у вигляді

$$y_1 = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 = \bar{x}_2 \cdot \bar{x}_3.$$

Відзначимо, що перший додаток є необов'язковим та відповідає Φ . Використання цього додатка дозволяє спростити функцію y_1 . Для y_2 припустимо, що усі $\Phi_i = 1$, тоді одержимо наступне

$$y_2 = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot x_3.$$

Тут перший, четвертий та п'ятий додатки є необов'язковими. Використовуючи операції склеювання стосовно до першого та другого, до першого та третього, третього та четвертого, а також до четвертого та

П'ятого, маємо

$$y_2 = \bar{x}_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot \bar{x}_3 + \bar{x}_1 \cdot x_2 + x_2 \cdot x_3.$$

Якщо прийняти $\Phi_1 = 1$, а $\Phi_2 = \Phi_3 = 0$, то мінімізована функція y_2 буде мати простий вигляд

$$y_2 = \bar{x}_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot \bar{x}_3.$$

Мінімізацію розглянутих вище функцій y_1 та y_2 можна провести також з використанням карт Карно (рис. 1.4а, б відповідно). Причому в цьому випадку вибір значень Φ_i визначається розташуванням одиниць у клітинках карт.

		x_1x_2	00	01	11	10	
x_3	0	0	Φ_1	1			<i>a</i>
	1						

		x_1x_2	00	01	11	10	
x_3	0	Φ_1	1			1	<i>b</i>
	1			Φ_3	Φ_2		

Рисунок 1.4 – Мінімізація функцій за допомогою карт Карно:
a – логічна функція y_1 ; *b* – логічна функція y_2

1.5 Методи мінімізації системи логічних функцій

Системою логічних функцій звичайно описується пристрій з декількома виходами. Спосіб мінімізації такої системи розглянемо на прикладах.

Приклад 1. На рис. 1.5а наведено умовне зображення напівсуматора, який призначений для додавання двох двійкових цифр із формуванням суми (S) та цифри переносу в наступний розряд (p). Процес підсумовування двох двійкових змінних описується двома логічними функціями наступного вигляду

$$S = f_1(x_1, x_2), \quad p = f_2(x_1, x_2).$$

Відповідно до правил додавання складемо таблицю істинності та запишемо структурні формули роботи цифрового пристрою. Безпосереднє використання методу Квайна або карт Карно не дозволяє провести подальше спрощення логічних виразів. Тому використовуємо наступне подання результату підсумовування

$$S = S' = f_3(x_1, x_2, p).$$

Складемо таблицю істинності для вихідної величини S' як функції трьох змінних. Використаємо для цього вихідну таблицю істинності (див. рис. 1.5б) та правила роботи пристрою (табл. 1.8), що проектуємо.

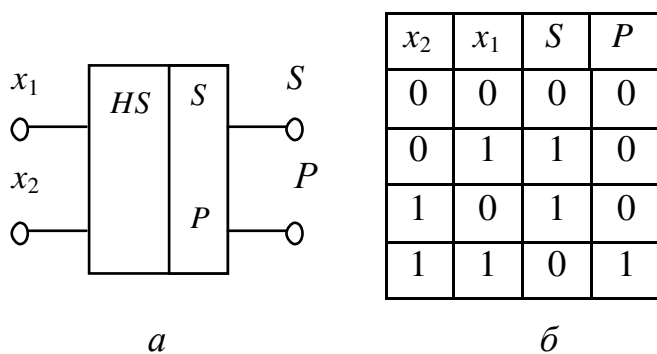


Рисунок 1.5 – Умовне зображення та таблиця істинності напівсуматора:
a – умовне зображення; *б* – таблиця істинності

Таблиця 1.8 – Таблиця істинності функції

p	x_2	x_1	S'
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	Φ_1
1	0	0	Φ_2
1	0	1	Φ_3
1	1	0	Φ_4
1	1	1	0

Відповідно до цих правил у випадку $x_1 = x_2 = 1, p = 1$. Це означає, що не повинна виникати комбінація вхідних сигналів $p x_1 x_2 = 011$. У таблиці істинності така комбінація відзначається знаком Φ_1 . Аналогічним способом відзначаються знаками Φ_2, Φ_3, Φ_4 комбінації вхідних сигналів 100, 101, 110. Карта Карно для отриманої не цілком визначеної логічної функції $S' = f_3(x_1, x_2, p)$ наведена на рис. 1.6. За її допомогою отримуємо спрощену структурну формулу

$$S' = x_1 \cdot \bar{p} + x_2 \cdot \bar{p} = (x_1 + x_2) \cdot \bar{p},$$

де $p = x_1 \cdot x_2$.

		00	01	11	10
p	0		1	Φ_1	1
	1	Φ_2	Φ_3	0	Φ_4

Рисунок 1.6 – Карта Карно напівсуматора

Порівнюючи отриману формулу з вихідною, бачимо, що для реалізації потрібно зменшити кількість логічних операцій.

Приклад 2. Потрібно побудувати одно розрядний комбінаційний суматор. Цей пристрій, призначено для реалізації двох функцій: додавання

двох двійкових цифр з урахуванням переносу з попереднього розряду та з формуванням цифр суми та переносу для наступного розряду.

Проведемо мінімізацію кожної функції окремо за допомогою карт Карно для ЗДНФ (рис. 1.7а)

		00	01	11	10
$P_i P_{i-1}$	0	0	1	0	1
	1	1	0	1	0
		0	0	1	0
		0	1	1	1

а

		$x_2 x_1$					
		00	01	11	10		
$P_i P_{i-1}$	00	0	1	Φ_1	1		
	01	1	Φ_2	Φ_4	Φ_3		
	11	Φ_8	0	1	0		
	10	Φ_5	Φ_6	0	Φ_7		

б

Рисунок 1.7 – Карты Карно:

а – функції додавання, б – формування результату

Карты Карно дозволяють отримати мінімізовану структурну формулу комбінаційного суматора у такому вигляді

$$S = \bar{p}_{i-1} \cdot \bar{x}_2 \cdot x_1 + \bar{p}_{i-1} \cdot x_2 \cdot \bar{x}_1 + p_{i-1} \cdot \bar{x}_2 \cdot \bar{x}_1 + p_{i-1} \cdot x_2 \cdot x_1,$$

де $p_i = x_2 \cdot x_1 + p_{i-1} \cdot x_2 + p_{i-1} \cdot x_1$.

Складемо таблицю істинності для функції $S' = f(p_{i-1}, x_1, x_2, p_i)$, виходячи з правил роботи цифрового пристрою (табл. 1.9). За таблицею 1.9 та картою Карно (рис. 1.7б) запишемо структурну формулу для не цілком визначеної функції S' у такому вигляді

$$S' = x_1 \cdot \bar{p}_i + x_2 \cdot \bar{p}_i + p_{i-1} \cdot \bar{p}_i + x_1 \cdot x_2 \cdot p_{i-1}.$$

У результаті порівняння виразів для S та S' бачимо, що число логічних операцій у формулі S' зменшилось. Тому структурна формула S' має простіший вигляд та реалізується меншою кількістю логічних елементів.

Контрольні запитання

1. Яким чином організовано двійкову систему числення?
2. Які ви знаєте закони алгебри логіки?
3. Сформулюйте основні тотожності алгебри логіки.
4. У чому полягає табличний засіб подання логічної функції?
5. Що таке ЗДНФ та ЗКНФ?
6. Сформулюйте правила склеювання у картах Карно.
7. На використанні яких математичних законів побудовано метод Квайна?
8. Які методи мінімізації систем логічних функцій вам відомі?

Таблиця 1.9 – Таблиця істинності функції

p_i	p_{i-1}	x_2	x_1	S'
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	Φ_1
0	1	0	0	1
0	1	0	1	Φ_2
0	1	1	0	Φ_3
0	1	1	1	Φ_4
1	0	0	0	Φ_5
1	0	0	1	Φ_6
1	0	1	0	Φ_7
1	0	1	1	0
1	1	0	0	Φ_8
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

2 МЕТОДИ СИНТЕЗУ ЦИФРОВИХ ПРИСТРОЇВ

2.1 Універсальні логічні елементи

Для задоволення вимогам однорідності функціональної схеми застосовуються універсальні логічні елементи (ЛЕ). До таких елементів відносяться: І-НІ, АБО-НІ, І-АБО-НІ.

І-НІ (елемент Шеффера) зображений на рис. 2.1а. Реалізована логічна функція може бути записана у вигляді $y = x_1 \cdot x_2 \cdot \dots \cdot x_n$. За допомогою такого ЛЕ можна побудувати логічні функції НІ, І, АБО (рис. 2.1б, в, г відповідно).

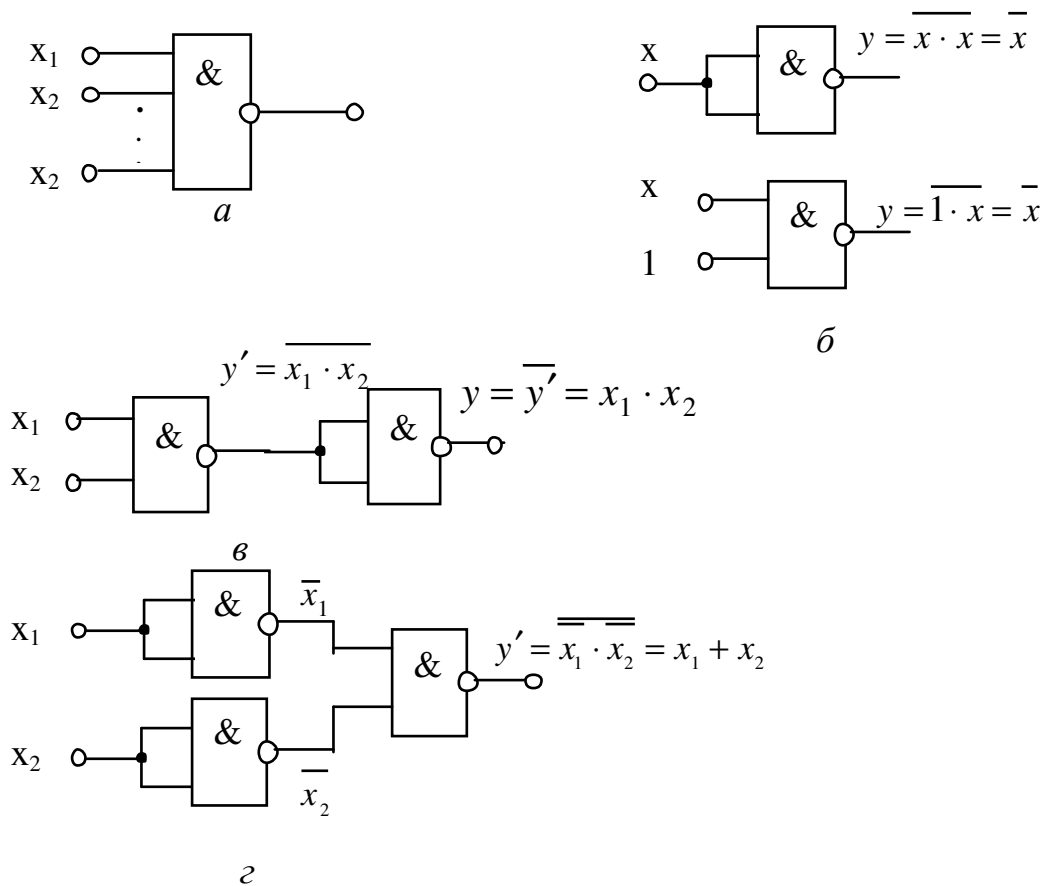


Рисунок 2.1 – Варіант використання ЛЕ І-НІ:

а – умовне зображення логічного елемента І-НІ; б – застосування для отримання інверсії; в – кон'юнкції; г – диз'юнкції

Використовуючи елемент АБО-НІ, також можна реалізувати логічні функції НІ, І, АБО. Логічні елементи І-НІ чи АБО-НІ мають властивість подвійності. Дійсно, замінюючи в таблиці істинності ЛЕ І-НІ (табл. 2.1) символи 0 на 1 і 1 на 0, одержимо табл. 2.2.

Таблиця 2.1

Таблиця 2.2

Таблиця 2.3

x_2	x_1	y
0	0	1
0	1	1
1	0	1
1	1	0

x_2	x_1	y''
1	1	0
1	0	0
0	1	0
0	0	1

x_2	x_1	y'
0	0	1
0	1	0
1	0	0
1	1	0

Порівнюючи цю таблицю з таблицею істинності ЛЕ АБО-НІ (табл. 2.3) бачимо, що з точністю до перестановки рядків вони збігаються. ЛЕ І-АБО-НІ подано на рис. 2.2. Реалізована ним логічна функція має такий вигляд

$$y = x_1 \cdot x_2 + x_3 \cdot x_4.$$

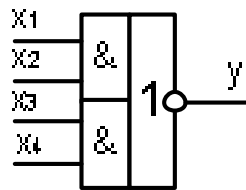


Рисунок 2.2 – Умовне зображення елемента 2І-2АБО-НІ

Так, наприклад, якщо $x_1 = x_3 = 1$, то $y = \overline{x_2 + x_4}$. При $x_1 = 0, x_2 = 1$ маємо $y = x_3 \cdot x_4$.

2.2 Приклади побудови комбінаційних пристроїв

Розглянемо, насамперед, приклади побудови комбінаційних пристроїв на логічних елементах І-НІ.

Приклад 1. Потрібно побудувати пристрій, який реалізує логічну функцію $y_1 = x_1 + x_2 \cdot \bar{x}_3$. Перетворимо логічну функцію за допомогою правила де Моргана, виключивши з неї операції АБО

$$y = x_1 + x_2 \cdot \bar{x}_3 = x_1 + x_2 \cdot x_3 = \overline{\bar{x}_1 \cdot x_2 \cdot \bar{x}_3}.$$

Схема, яка реалізує цю структурну формулу, наведена на рис. 2.3.

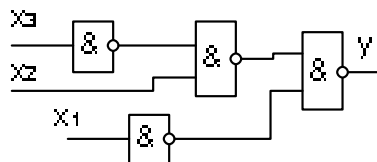


Рисунок 2.3 – Схема пристрою на ЛЕ І-НІ

З розглянутого приклада видно, що при реалізації логічної функції на ЛЕ І-НІ доцільно використовувати подання цієї функції в ЗДНФ. Ця рекомендація підтверджується наступним прикладом.

Приклад 2. Для реалізації логічної функції

$$y = x_1 \cdot x_2 + x_1 \cdot x_3 = \overline{x_1 \cdot x_2 + x_1 \cdot x_3} = \overline{\overline{x_1 \cdot x_2} \cdot \overline{x_1 \cdot x_3}}$$

потрібно три ЛЕ І-НІ. Переходячи до подання функції в ЗКНФ, маємо

$$y = x_1(x_2 + x_3) = x_1(\overline{\overline{x_2 + x_3}}) = \overline{\overline{x_1 \cdot \overline{x_2} \cdot \overline{x_3}}}$$

Для реалізації цієї логічної функції буде потрібно вже п'ять логічних елементів І-НІ.

Приклад 3. Синтезувати мажоритарний елемент (рис. 2.4, а), який являє собою пристрій з непарним числом m входів. У нього на виході $y = 1$, якщо на більшості входів сигнали рівні 1, та $y = 0$, якщо на більшості входів сигнали рівні 0. Склавши відповідно до словесного опису роботи пристрою таблицю істинності (табл. 2.4), одержимо за допомогою карти Карно (див. рис. 2.4, б) мінімізовану функцію для $m=3$

$$y = x_1 \cdot x_3 + x_1 \cdot x_2 + x_2 \cdot x_3.$$

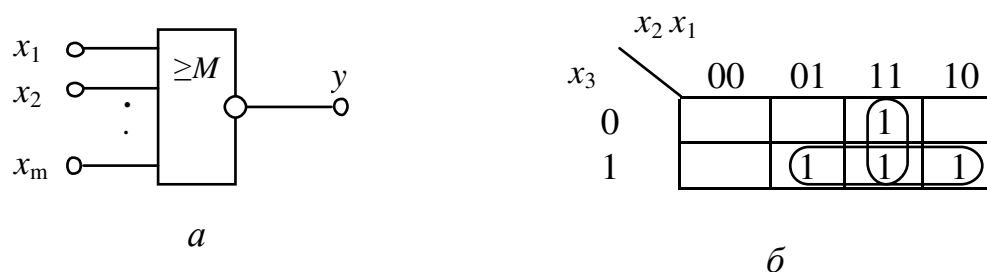


Рисунок 2.4 – Умовне зображення елемента та його карта Карно:
а – зображення мажоритарного елемента; б – карта Карно

Таблиця 2.4 – Таблиця істинності

x_3	x_2	x_1	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Виключивши за допомогою правила де Моргана операції АБО, одержимо

$$y = x_1 \cdot x_3 + x_1 \cdot x_2 + x_2 \cdot x_3 = \overline{\overline{x_1 \cdot x_3} \cdot \overline{x_1 \cdot x_2} \cdot \overline{x_2 \cdot x_3}}$$

Реалізація цього виразу можлива на ЛЕ І-НІ.

Приклад 4. Побудувати одно розрядний комбінаційний суматор на елементах І-НІ. Робота суматора описується функціями

$$p_i = x_1 x_2 + x_1 p_{i-1} + x_2 p_{i-1},$$

$$S = x_1 x_2 p_{i-1} + x_1 x_2 p_{i-1} + x_1 x_2 p_{i-1} + x_1 x_2 p_{i-1}.$$

Після відповідних еквівалентних перетворень та застосувавши правило де Моргана, одержимо

$$S = x_1 x_2 p_{i-1} x_1 x_2 p_{i-1} x_1 x_2 p_{i-1} x_1 x_2 p_{i-1},$$

$$p = x_1 x_2 x_1 p_{i-1} x_2 p_{i-1}.$$

Схема пристрою на елементах І-НІ наведена на рис. 2.5.

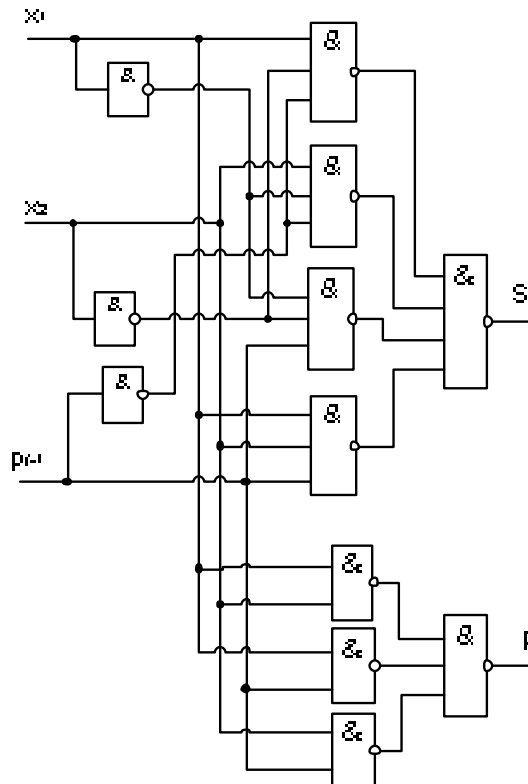


Рисунок 2.5 – Схема однорозрядного суматора на елементах І-НІ

При побудові комбінаційних пристроїв на елементах АБО-НІ логічну формулу необхідно перетворити таким чином, щоб виключити операцію І.

Приклад 5. Розглянемо логічну функцію вигляду

$$y = x_1 \cdot x_2 + x_1 \cdot x_3 = x_1 \cdot x_2 + x_1 \cdot x_3 = \bar{x}_1 + \bar{x}_2 + \bar{x}_1 + \bar{x}_3.$$

Для її реалізації потрібно сім ЛЕ АБО-НІ. Переводячи цю функцію в ЗКНФ, отримаємо

$$y = x_1 \cdot (x_2 + x_3) = x_1 \cdot (x_2 + x_3) = \bar{x}_1 + x_2 + x_3 = \bar{x}_1 + x_2 + x_3.$$

Для реалізації такої функції буде потрібно три ЛЕ АБО-НІ. З розглянутого приклада видно, що при синтезі електронних пристроїв на ЛЕ АБО-НІ доцільно використовувати представлення логічної функції в ЗКНФ.

Приклад 6. Розглянемо синтез мажоритарного елемента (приклад 3). По табл. 2.4 за допомогою карт Карно одержимо мінімізовану функцію та

запишемо її, використовуючи операції АБО-НІ

$$y = (x_2 + x_3)(x_1 + x_2)(x_1 + x_3) = \overline{(x_2 + x_3)(x_1 + x_2)(x_1 + x_3)} = \overline{x_2 + x_3 + x_1 + x_2 + x_1 + x_3}.$$

Подаючи функцію в ЗДНФ, маємо таку формулу

$$y = \overline{x_1 x_2 + x_1 x_3 + x_2 x_3} = \overline{x_1 x_2 + x_1 x_3 + x_2 x_3} = \overline{x_1} + \overline{x_2} + \overline{x_1} + \overline{x_3} + \overline{x_2} + \overline{x_3}.$$

Очевидно, що в другому випадку при реалізації пристрою буде потрібна більша кількість логічних елементів.

2.3 Загальна задача синтезу комбінаційних пристроїв

Вихідними даними для проектування комбінаційних пристроїв (КП) є їхній словесний опис та вимоги до основних електричних параметрів. У задачу синтезу КП входить одержання структурної схеми мінімальної складності, яку реалізовано на заданому чи обраному типі логічних елементів. Першим етапом синтезу КП з одним виходом є складання таблиці істинності по словесному опису.

Призначенням другого етапу є одержання структурної формули логічної функції, причому на цьому етапі прагнуть одержати найбільш простий (мінімізований) логічний вираз заданої функції. У більшості випадків процес спрощення виразу ЛФ зводиться до застосування операцій склеювань й поглинань. При використанні алгебраїчних методів мінімізації (методів Квайна та Квайна – Мак-Класкі) по таблиці істинності записується ЗДНФ (чи ЗКНФ) та проводиться склеювання наявних добутоків (чи сум) [1...3]. При графічному методі, який використовує карти Карно [1], по таблиці істинності за допомогою карти Карно виявляють усі можливі склеювання й поглинання, а потім записують мінімізовану структурну формулу.

Після одержання мінімізованої функції за допомогою зазначених методів можна спробувати досягти подальшого спрощення структурної формули шляхом використання додаткових прийомів мінімізації. До них відносяться: застосування закону заперечення, використання розподільного закону та додавання складових тотожно рівних 0.

На третьому етапі синтезу роблять запис отриманої мінімізованої структурної формули в заданому базисі у вигляді комбінації операцій, які виконуються заданим (обраним) типом логічних елементів. Цей запис виконується за допомогою відповідних прийомів. Так, при реалізації на елементах І-НІ над отриманою диз'юнктивною формою ставлять два знаки інверсії та за допомогою закону заперечення перетворюють інверсію диз'юнкції в кон'юнкцію інверсій. У результаті виходить логічний вираз, який містить тільки операції І-НІ.

Четвертим етапом синтезу є складання структурної схеми пристрою. На цьому етапі кожній логічній операції перетвореної структурної формули ставлять у відповідність визначений логічний елемент заданого типу

(базису). На основі структурної формули здійснюють необхідні з'єднання між елементами.

У загальному випадку реалізація отриманої мінімізованої функції має багато варіантів, які обумовлено типом використовуваних елементів та запропонованих вимог до пристрою (швидкодія, простота реалізації, вартість та ін.). Проілюструємо можливе різноманіття рішень задачі побудови комбінаційного пристрою.

Приклад. Побудувати схему, яка забезпечує мінімальні апаратні витрати і затримки сигналів при реалізації функції $y = \bar{a} \cdot \bar{c} + \bar{b} \cdot \bar{c} + \bar{d}$.

Схемна реалізація заданої ЛФ приведена на рис. 2.6а.

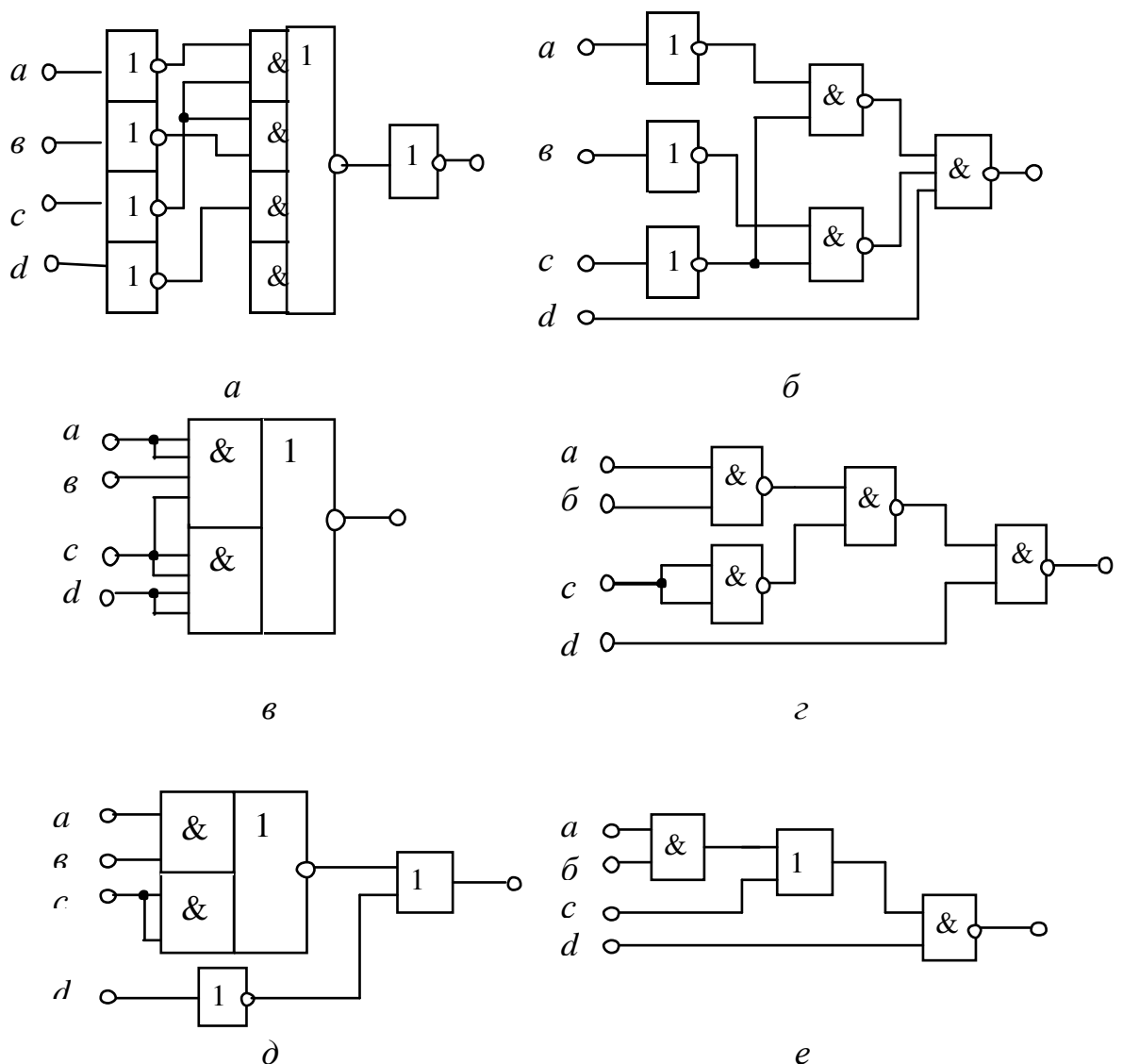


Рисунок 2.6 – Варіанти схемної реалізації заданої функції

Апаратні витрати будемо оцінювати по числу застосовуваних корпусів мікросхем. Доцільно при використанні ЛЕ однієї серії покласти, що затримки сигналів ЛЕ різного типу однакові на рівні деякої усередненої для даної серії величини τ . Оскільки затримки поширення сигналу в інверторі та

елементі І-АБО-НІ однакові, то затримка сигналів усієї схеми дорівнює $T_1=3\tau$. Схема складається з п'яти інверторів, кожний з яких займає 1/6 корпусу мікросхеми ЛН1, а також логічного елемента І-АБО-НІ, який займає корпус мікросхеми ЛРЗ. Таким чином, апаратні витрати W можна оцінити за допомогою такого виразу

$$W_1 = 5 \cdot 1/6 + 1 = 22/12.$$

Невикористані елементи частково зайнятого корпусу (у даному випадку шостий інвертор у ІС ЛН1) не враховуються, тому що вони можуть бути застосовані в інших пристроях. Оскільки в ІС ЛРЗ частина входів не використовується, то доцільно будувати схему за допомогою елементів І-НІ, які мають меншу кількість входів. Застосовуючи закони заперечення до заданої формули, одержимо таке

$$y_1 = \overline{\overline{a \cdot \bar{c}} + \overline{b \cdot \bar{c}} + \bar{d}} = \overline{\overline{a \cdot \bar{c}} + \overline{b \cdot \bar{c}} + d}.$$

При реалізації функції цього виду (рис. 2.6б) затримка сигналу та апаратні витрати дорівнюють

$$t_2 = 3\tau \quad ; \quad W_2 = 3 \cdot 1/6 + 2 \cdot 1/4 + 1 \cdot 1/3 = 16/12.$$

Продовжуючи перетворення заданої логічної функції, одержимо такий вираз (рис. 2.6в)

$$y_2 = \overline{a \cdot \bar{c}} + \overline{b \cdot \bar{c}} + \bar{d} = (\overline{a + b}) \cdot \bar{c} + \bar{d} = \overline{\overline{a + b} + c} + \bar{d} = \overline{a \cdot b + c} + \bar{d}.$$

У цьому випадку $T_3=2\tau$, $W_3 = 11/12$.

Можна одержати ще один вид запису структурної формули

$$y_3 = \overline{a \cdot b + c} + \bar{d} = \overline{(a \cdot b + c) \cdot d},$$

при реалізації якої (рис. 2.6г), потрібно менше корпусів ІС $W_4 = 9/12$ при $T_4=3\tau$. При подальшому перетворенні формули одержимо

$$y_4 = a \cdot b \cdot d + c \cdot d.$$

Структурна схема для y_4 представлена на рис. 2.6д

$$W_5 = 12/12, \quad T_5 = \tau.$$

При прагненні реалізувати задану функцію за допомогою одного типу ЛЕ її можна записати у вигляді

$$y_5 = \overline{a \cdot b \cdot \bar{c}} + \bar{d} = \overline{\overline{a \cdot b \cdot \bar{c}} \cdot d}.$$

У структурній схемі (рис. 2.6е) потрібен один корпус двовходових елементів І-НІ (К555ЛА3), тобто $W_6 = 12/12$ при $T_6=3\tau$. Таким чином, як видно з розглянутого вище прикладу, можливі різні схемні реалізації однієї і тієї ж структурної формули, яка отримана в результаті використання процедури мінімізації. У тому випадку, коли логічна функція є не цілком визначеною, на другому етапі синтезу функцію довизначають таким чином, щоб максимально спростити структурну формулу. При синтезі комбінаційних пристроїв з декількома виходами потрібно мінімізувати системи логічних функцій, які описують роботу такого пристрою. У цьому випадку прагнуть так виділити у функціях загальні складові, щоб з

відповідних їм вузлів пристрою можна було здійснити розгалуження сигналу на кілька напрямків.

2.4 Проектування комбінаційних пристроїв, вільних від змагань сигналів

При синтезі комбінаційних пристроїв на ІС однією з основних задач є забезпечення їхньої функціональної надійності. Під функціональною надійністю КП мають на увазі його властивість точно реалізовувати заданий алгоритм, тобто значення сигналу на виході КП повинне точно відповідати значенню синтезованої функції цього пристрою. Алгеброю логіки не розглядаються процеси переходу КП з одного стану в інший, тому що апарат алгебри логіки не враховує тимчасові (інерційні) параметри ЛЕ (затримку сигналів у фізичних елементах реальних пристроїв) і структуру реальних сигналів (кінцеву тривалість фронту і спаду імпульсів).

Наявність затримки сигналів у ЛЕ, а також розкид величини цієї затримки і кінцеві тривалості перепадів рівнів вхідних сигналів приводять до того, що під час перехідних процесів пристрій функціонує з порушенням законів алгебри логіки, в ньому спостерігається змагання сигналів.

Змагання сигналів може бути безпечним та небезпечним. *Безпечними* є змагання, які не призводять до виникнення на виході пристрою зміни рівня сигналу, не передбаченого алгоритмом роботи пристрою. Небезпечні змагання розрізняють за місцем їх виникнення в умовах переходу КП з одного стану в інший. Розрізняють змагання сигналів на входах і логічних елементах пристрою.

Змагання сигналів на входах мають місце, якщо під час перехідних процесів можуть бути моменти часу, коли сигнали на обох входах набувають однакового логічного рівня. Виникнення змагань сигналів у логічних елементах КП потребує пояснення. Логічний елемент у більшості випадків подається у вигляді двох частин: одна безінерційно виконує логічну функцію, інша включена слідом за першою і є елементом затримки на час, рівний середній затримці сигналів $t_{зсер}$.

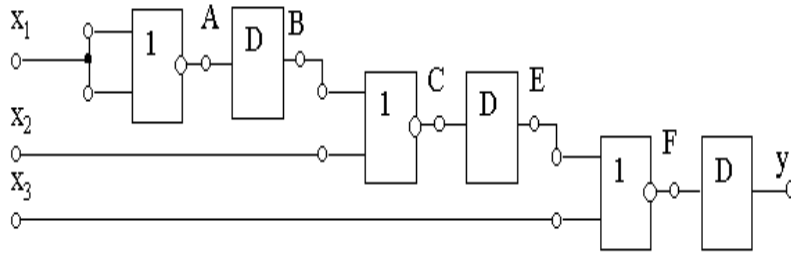
Використовуючи таке подання логічного елемента, розглянемо приклад послідовного з'єднання декількох логічних елементів, які реалізують логічну функцію (рис. 2.7а).

$$y = x_1 + x_2 + x_3.$$

Припустимо, що при $t < t_0$ значення $x_1 = 0$, $x_2 = x_3 = 1$, а при $t = t_0$ вхідні сигнали інвертуються. Без урахування затримок значення вихідного сигналу $y = 0$. Простежимо динаміку роботи пристрою, використовуючи для цього модель та часові діаграми вхідних і вихідних сигналів. Ці сигнали подані на рис. 2.7б.

Елементи затримки D включені послідовно з ЛЕ, затримуючи проходження сигналів на $t_{зсер}$ в точках B , E і на виході y . Результати аналізу

роботи КП приведені в таблиці (рис. 2.7б). Із таблиці видно, що на виході у виникає перешкода тривалістю $2 t_{3 \text{ сеп}}$ у вигляді логічної 1.



a

Рисунок 2.7 – Приклад появи завади при виникненні запізньень:
a – модель КП; б – часові діаграми

Залежно від умов переходу КП від одного стану до іншого розрізняють статичні й динамічні змагання. Якщо для двох послідовних у часі сусідніх станів сигналів на входах стан виходу повинен залишатися незмінним, то змагання в пристрої називають *статичними*. Якщо два послідовних у часі сусідні стани входів повинні мати перехід стану на виході, то змагання, які мають місце в пристрої, називають *динамічними*.

Статичні змагання підрозділяють на одиничні і нульові. Одиничні змагання з'являються у випадку, коли при зміні входніх сигналів на виході КП повинно зберегтися значення логічної 1, а в результаті змагань з'являється 0. При нульових змаганнях під час перехідного процесу на виходах з'являється 1, коли повинен зберігатись нуль.

Приведемо приклади утворення неалгоритмічних переходів при статичних змаганнях. Для ілюстрації розглянемо пристрій (рис. 2.8), який описується логічною функцією $y = x_1 \cdot x_2 \cdot x_3 \cdot x_1$. Часові діаграми, котрі пояснюють виникнення змагань сигналів, приведені на рис. 2.8б. Причому в ЛЕ ІІ затримка сигналу дорівнює t_{31} , а в ЛЕ І - ІІ - t_{32} . З аналізу часових діаграм видно, що в КП виникли два неалгоритмічних переходи (одиничного змагання). Процес утворення нульових змагань покажемо на прикладі КП (рис.2.8а), який реалізує логічну функцію $y = \bar{x}_1 \cdot x_2 + x_3 \cdot x_1$. Як неважко переконатися (рис. 2.9б), у такому пристрої можливе виникнення нульових змагань. Складаємо таблицю істинності (табл. 2.5).

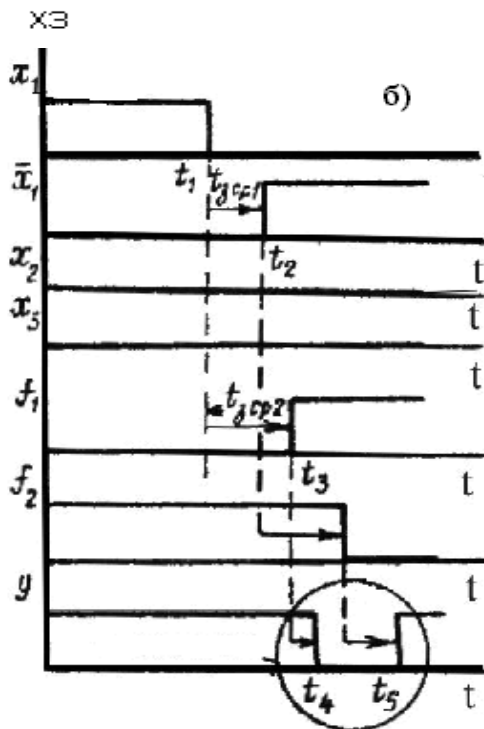
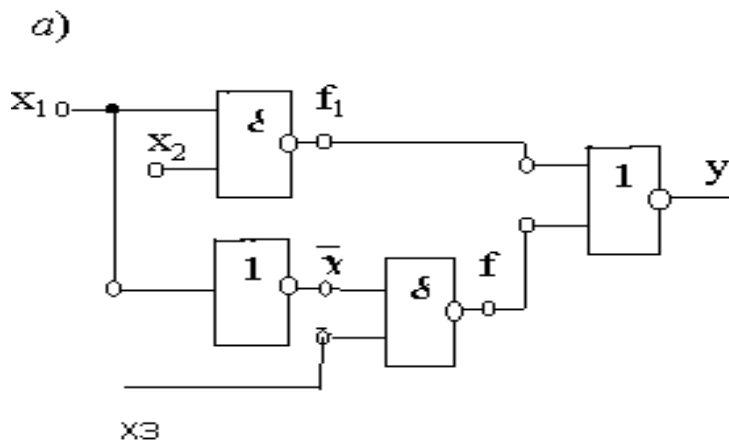


Рисунок 2.8 – Приклад появи
одиничних змагань:
а – схема пристрою;
б – часові діаграми

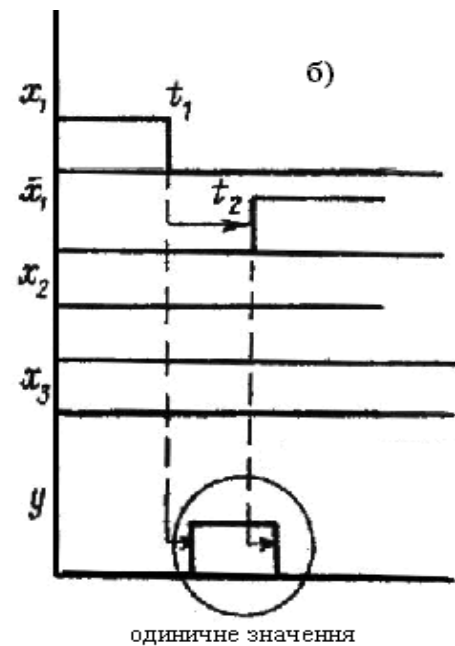
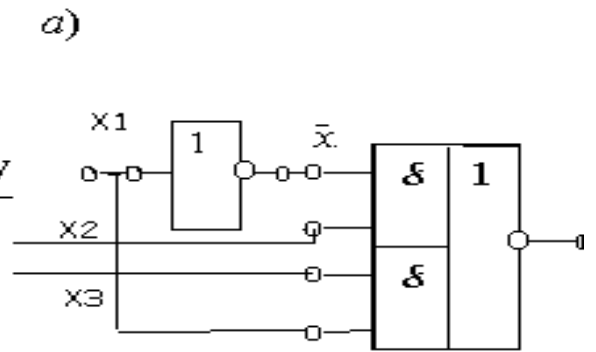


Рисунок 2.9 – Приклад появи
нульових змагань:
а – схема пристрою;
б – часові діаграми

Таблиця 2.5 – Таблиця істинності

Час	x_3	x_2	x_1	A	B	C	E	F	Y
$t < t_0$	0	1	1	1	1	0	0	0	0
t_0	1	0	0	0	1	0	0	1	0
$t_0 + t_{3\text{cp}}$	1	0	0	0	0	1	0	1	1
$t_0 + 2t_{3\text{cp}}$	1	0	0	0	0	1	1	0	1
$t_0 + 3t_{3\text{cp}}$	1	0	0	0	0	1	1	0	0
$t_0 + 4t_{3\text{cp}}$	1	0	0	0	0	1	1	0	0

Аналіз роботи КП з метою виявлення небезпечних змагань по часових діаграмах при великій кількості ЛЕ виявляється досить трудомістким. Тому,

як правило, використовуються формальні методи аналізу. Розглянемо докладніше *аналітичний метод* чи метод Мак-Класкі [1]. Логічну функцію у перетворюють або в диз'юнктивну нормальну форму (ДНФ) при аналізі на статичні нульові змагання, або в кон'юнктиву нормальну форму (КНФ) при аналізі на статичні одиночні змагання. При цьому не допускаються такі логічні перетворення, при яких відбувається втрата вхідних змінних x_i та \bar{x}_i . З цією метою x_i та \bar{x}_i розглядають як незалежні змінні, оскільки в динамічному режимі (тобто під час перехідних процесів) у деяких точках ланки ЛЕ x_i та \bar{x}_i можуть виявитися не взаємно інверсними, а одночасно рівними або логічному 0, або логічній 1.

За методом Мак-Класкі статичні нульові змагання мають місце, якщо:

а) логічна функція у в ДНФ містить хоча б один з додатків, у який одна із незалежних змінних входить у прямому x_i та інверсному \bar{x}_i вигляді (наприклад, $\dots + x_i \cdot \bar{x}_i + \dots$);

б) виконується умова

$$y \left| \begin{array}{l} x_1 = x_2 = \dots = 1 \\ \text{окрім } x_i = 1 \end{array} \right. + y \left| \begin{array}{l} x_1 = x_2 = \dots = 1 \\ \text{окрім } x_i = 0 \end{array} \right. \neq 1$$

Перший додаток у цьому виразі виходить шляхом підстановки в логічну функцію у замість x_i логічної одиниці, а замість \bar{x}_i – логічного нуля. Другий додаток виходить шляхом підстановки замість x_i значення логічного "0", а замість \bar{x}_i – логічної одиниці. Усі інші вхідні змінні $x_1, x_2 \dots$ при цьому залишаються записаними в загальному вигляді. Після відповідних спрощень ці змінні замінюються на логічні "1" для того, щоб з'ясувати, чи виконується друга умова. Якщо умова виконується, то у КП присутні нульові змагання.

Статичні одиночні змагання мають місце, якщо:

а) логічна функція у виді КНФ містить хоча б один співмножник, у який одна з незалежних змінних входить у прямому x_i та інверсному \bar{x}_i вигляді, наприклад, $(x_i + \bar{x}_i)$;

б) виконується така умова

$$y \left| \begin{array}{l} x_1 = x_2 = \dots = 0 \\ \text{окрім } x_i = 1 \end{array} \right. + y \left| \begin{array}{l} x_1 = x_2 = \dots = 0 \\ \text{окрім } x_i = 0 \end{array} \right. \neq 0$$

Розглянемо практичне застосування методу Мак-Класкі на прикладі функції

$$y = \overline{\bar{x}_1 \cdot x_2} + x_3 \cdot x_1.$$

Із структурної формули видно, що змагання сигналів можуть виникнути в ЛЕ АБО-ІІ. Припустимо, що в такому КП можливі статичні нульові змагання. Для перевірки цього припущення представимо вираз для у в ДНФ

$$y = \overline{\bar{x}_1 x_2} + \overline{x_1 x_3} = (x_1 + \bar{x}_2)(\bar{x}_1 + \bar{x}_3) = x_1 \bar{x}_1 + \bar{x}_2 \bar{x}_1 + x_1 \bar{x}_3 + \bar{x}_2 x_3.$$

У цьому виразі виконується перша умова, а саме: тут є доданок вигляду $x_i \cdot \bar{x}_i$. Перевіримо другу умову

$$(1 \cdot 0 + \bar{x}_2 \cdot 0 + 1 \cdot \bar{x}_3 + \bar{x}_2 \cdot \bar{x}_3) + (0 \cdot 1 + \bar{x}_2 \cdot 1 + 0 \cdot x_3 + \bar{x}_2 \cdot \bar{x}_3) = \\ = \bar{x}_3 + \bar{x}_2 \cdot \bar{x}_3 + \bar{x}_2 + \bar{x}_2 \cdot \bar{x}_3 = \bar{x}_2 + \bar{x}_3 + \bar{x}_2 \cdot \bar{x}_3 = \bar{x}_2 + \bar{x}_3.$$

Після підстановки замість x_2 і x_3 логічної 1 одержимо $0 + 0 \neq 1$. Отже, у такому КП виникають статичні нульові змагання.

Перевіримо роботу КП на наявність одиничних змагань. Для цього перетворимо вираз для y в КНФ

$$y = \overline{\bar{x}_1 \cdot x_2 \cdot x_1 \cdot x_3} = (x_1 + \bar{x}_2) \cdot (\bar{x}_1 + \bar{x}_3).$$

Умова не виконується, отже, одиничні змагання сигналів відсутні.

Іншим методом аналізу КП на наявність небезпечних змагань сигналів є метод з використанням карт Карно. Застосування цього методу зручно розглянути на конкретному прикладі. Для цього проведемо аналіз КП, логічна функція y якого визначається таким виразом:

$$y = x_1 \cdot \bar{x}_1 + \bar{x}_2 \cdot \bar{x}_1 + x_1 \cdot \bar{x}_3 + \bar{x}_2 \cdot x_3.$$

Карта Карно для y будується на основі її таблиці істинності (табл. 2.6) причому в даному випадку зручно взяти замість функції y інверсну функцію.

Для інверсної функції побудуємо карту Карно (рис 2.10). Виконавши лінеаризацію, одержимо такий вираз логічної функції

$$y = \bar{x}_1 \cdot x_2 + x_1 \cdot x_3.$$

Таблиця 2.6 – Таблиця істинності

x_3	x_2	x_1	y	\bar{y}
0	0	0	1	0
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

На картах Карно сусідні одиниці та сусідні нулі охоплюються загальним контуром (контуром склеювання). Якщо при зміні однієї з вхідних змінних зміни функції y такі, що на карті відбувається перехід з одного контуру склеювання до іншого, з ним не зв'язаного, то в КП мають місце змагання. Так, на рис. 2.10 видно, що при переході функції y від значення логічної 1 до 1 у випадку зміни однієї з вхідних змінних x_1 чи x_2 (сусідні переходи) вихід за межі одного контуру склеювання не буде відбуватися. З іншого боку, при переході функції y від значення логічного 0 до 0 здійснюється перехід з одного до іншого контуру. Це означає наявність у КП нульових змагань. Отже, умовою відсутності статичних змагань буде

зв'язаність усіх контурів склеювання на карті Карно, що досягається введенням у структурну формулу додаткових співмножників в інверсну функцію y , які забезпечують зв'язаність контурів склеювання.

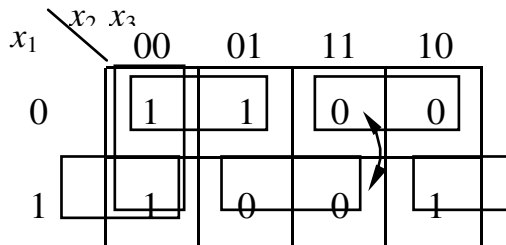


Рисунок 2.10 – Карта Карно для інверсного значення функції y

При проектуванні КП, вільних від небезпечних статичних змагань сигналів, використовуються наступні правила побудови пристроїв.

Застосування структурних методів. Найбільш ефективним є метод проектування з використанням карт Карно, відповідно до якого для кожної пари станів сигналів на входах КП необхідно мати на картах Карно для функцій y та \bar{y} контури склеювання, причому контури відповідають одній із цих складних функцій.

Корекція небезпечних змагань. КП у вигляді узагальненої структурної схеми (рис. 2.11), де A_1 та A_2 – ланцюги, за якими відбувається передача вхідних сигналів. Ці сигнали створюють на виході елемента D_n неалгоритмічні переходи, які поширюються далі по ланцюгу A_3 . Можна показати [1, 2], що виникаючий через ці переходи помилковий сигнал при виконанні визначених умов буде поступово зменшуватися за тривалістю в ланцюгу A_3 . На цьому і ґрунтується метод корекції небезпечних змагань.

Розкид значень затримок сигналів у ланцюгах A_1 і A_2 визначає тривалість складного сигналу, який утвориться на виході $D7$ (рис. 2.11). Задача корекції небезпечних змагань ланцюгів A_1 і A_2 у всіх відомих випадках складається у визначенні припустимої різниці затримок $\tau_{\text{доп}}$ цих ланцюгів. Ця затримка формує помилковий сигнал, який може бути подавлений ланцюгом A_3 у силу того, що послідовний ланцюг ЛЕ має формуючі властивості [1, 3]. Таке правило застосовується при розробці структур інтегральних схем середнього і більшого ступеня інтеграції (СІС і БІС).

Введення синхронізації. Роботою цифрових пристроїв у даному випадку управляють тактові (синхронізуючі) послідовності так, що запис та зчитування інформації здійснюються тільки протягом тривалості тактових імпульсів. Переключення КП повинне відбуватися за інтервал часу Δt між тактовими імпульсами. При цьому тривалість Δt вибирається такою, щоб протягом цього інтервалу часу всі перехідні процеси, які пов'язані з переключенням КП, закінчилися і на виходах КП установилися стаціонарні значення сигналів. Відзначимо, що введення синхронізації істотно зменшує

швидкодію пристрою.

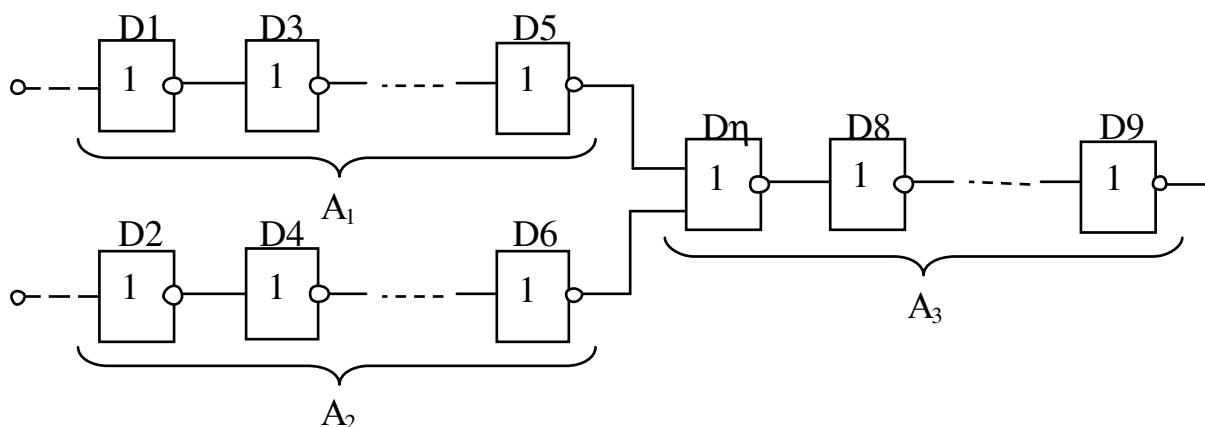


Рисунок 2.11 – Узагальнена структурна схема КП

Селекція імпульсів за тривалістю. Якщо тривалість помилкового сигналу (перешкоди), обумовленого змаганнями, менша, ніж час між сусідніми змінами рівнів сигналів, то можна використовувати для його придушення селектори імпульсів по тривалості.

Застосування визначеного порядку зміни сигналів і станів КП. Оскільки небезпечні змагання виявляються у вигляді перешкод тільки при зміні визначених комбінацій вхідних сигналів, то іноді можна передбачити певний порядок чергування цих комбінацій, при якому небезпечні змагання будуть відсутні.

Слід зазначити, що при використанні в радіотехнічних пристроях швидкодіючих ЛЕ і у випадках передачі сигналів між блоками пристрою на великі відстані необхідно при аналізі змагань враховувати затримки поширення сигналів по ланцюгах зв'язку між елементами і від блоку до блоку. Необхідно ще раз підкреслити, що проблема змагань у цифрових пристроях є дуже серйозною, тому що з ними пов'язані відмови у роботі схем.

2.5 Засоби формального опису послідовнісних пристроїв

На відміну від комбінаційних пристроїв (КП) вихідні сигнали послідовнісних пристроїв (ПП) визначаються не тільки комбінацією вхідних сигналів, але залежать також від внутрішнього стану пристрою (тобто набору внутрішніх перемінних) при надходженні вхідних сигналів.

Внутрішній стан ПП відбиває передісторію (послідовність) його роботи, тобто пристрій має пам'ять. Наявність пам'яті є характерною властивістю будь-якого ПП, тому структурна схема ПП (рис. 2.12) може бути представлена складеною з двох блоків:

- логічного перетворювача (ЛП), виконаного на основі КП;
- блока пам'яті (БП), як сукупності елементів пам'яті EP_1, EP_2, \dots, EP_r .

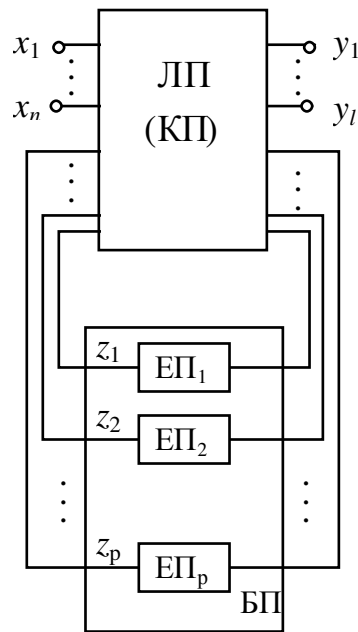


Рисунок 2.12 – Структурна схема ПП

Послідовнісні пристрої можна описати наступним чином:

- кінцевою безліччю станів входу $X = \{x_1, x_2, \dots, x_n\}$,
- кінцевою безліччю станів виходу $Y = \{y_1, y_2, \dots, y_l\}$,
- кінцевою безліччю внутрішніх станів пристрою $Z = \{z_1, z_2, \dots, z_p\}$, де x_i, y_i, z_i – вхідні, вихідні і внутрішні змінні,
 - а також двома функціями:
 - 1) функцією переходів, яка визначає порядок зміни внутрішніх станів;
 - 2) функцією виходів, яка задає стан виходів залежно від стану входів та внутрішнього стану.

При роботі цифрових пристроїв можуть бути виділені часові інтервали, на яких усі змінні (вхідні, вихідні та внутрішні) зберігають постійні значення. Ці інтервали часу Δt називають тактами роботи цифрового пристрою. Надалі будемо розглядати роботу ПП лише в моменти часу t_i ($i = 1, 2$) як рівновіддалені один від одного на величину Δt . Ці такти можуть бути довільно розташованими. Вважається, що саме в ці моменти відбувається зміна тієї чи іншої вхідної змінної. Позначимо x_i^n величину вхідної змінної x_i у момент часу t_n . Запис тієї ж вхідної змінної у формі x_i^{n-1} та x_i^{n+1} буде означати, що ця змінна розглядається в моменти t_{n-1} і t_{n+1} .

Розрізняють два типи ПП: асинхронні та синхронні. В асинхронних ПП усі входи рівноправні, і зміна сигналу на будь-якому вході може викликати зміну сигналів на яких-небудь виходах. У синхронних ПП мається мінімум один вхід синхронізації (С) і вхідні сигнали можуть впливати на роботу ПП лише при наявності визначеного сигналу на цьому вході.

Для того, щоб одержати алгоритм функціонування ПП за словесним описом його роботи, необхідно задати функції переходів та виходів. Для функції переходів це буде означати завдання визначеного переходу з одного внутрішнього стану Z_i в інший Z_j (при цьому не виключається випадок $i = j$)

при стані входу X_k (або при зміні послідовностей на вході ПП). Завдання функції виходів складається для кожної пари X_i і Z_i стану виходу $Y_i \in Y$. Обидві функції можна подати у вигляді таблиць (відповідно переходів і виходів) чи за допомогою графа.

Розглянемо правила побудови таблиць переходів. Рядки таблиці переходів відповідають поточним внутрішнім станам ПУ, стовпці – станам входів пристрою. Елементи таблиці переходів відповідають внутрішнім станам, у які повинен перейти ПП під впливом вхідних сигналів. Таблиця переходів у звичайних випадках містить 2^n стовпців, де n – число вхідних змінних X . Число рядків дорівнює числу станів схеми.

У табл. 2.7 приведено приклад переходів синхронного ПП, функції якого визначені для всіх наборів X і Z . Як бачимо, пристрій має чотири стани входу (X_1, X_2, X_3, X_4) і чотири внутрішніх стани (Z_1, Z_2, Z_3, Z_4). У кожній клітинці таблиці переходів зазначений номер внутрішнього стану, у який пристрій повинен перейти в наступний момент часу. Наприклад, при станах входу X_1 і X_4 ПП не змінює свого внутрішнього стану, при X_2 – пристрій переходить у наступний внутрішній стан, при X_3 – ПП переходить у попередній стан.

Таблиця 2.7 – Таблиця переходів

	X_1	X_2	X_3	X_4
Z_1	z_1	z_2	z_4	z_1
Z_2	z_2	z_3	z_1	z_2
Z_3	z_3	z_4	z_2	z_3
Z_4	z_4	z_1	z_3	z_4

Таблиця 2.8 – Таблиця переходів

	X_1	X_2
Z_1	1	2
Z_2	3	–
Z_3	–	4
Z_4	1	–

Таблиця 2.9 – Таблиця переходів

	X_1	X_2	X_3	X_4
Z_1	Y_1	Y_2	Y_4	Y_1
Z_2	Y_2	Y_3	Y_1	Y_2
Z_3	Y_3	Y_4	Y_2	Y_3
Z_4	Y_4	Y_1	Y_3	Y_4

Таблиця 2.10 – Таблиця переходів

	X_1	X_2	X_3	X_4
Z_1	$Z_1; Y_1$	$Z_2; Y_2$	$Z_4; Y_4$	$Z_1; Y_1$
Z_2	$Z_2; Y_2$	$Z_3; Y_3$	$Z_1; Y_1$	$Z_2; Y_2$
Z_3	$Z_3; Y_3$	$Z_4; Y_4$	$Z_2; Y_2$	$Z_3; Y_3$
Z_4	$Z_4; Y_4$	$Z_1; Y_1$	$Z_3; Y_3$	$Z_4; Y_4$

Надалі з метою спрощення замість Z_i у клітинках таблиць будемо вказувати лише номер стану. Якщо в ПП який-небудь стан не визначений, тобто він є невикористаним чи забороненим, то у відповідній клітинці таблиці ставлять прочерк (табл. 2.8).

У таблиці виходів (табл. 2.9) задається відповідність між станом виходу пристрою та його внутрішнім станом. Якщо обидві таблиці поєднати, одержимо так звану таблицю станів. Вона задає одночасно як функцію переходів, так і функцію виходів. У кожній клітинці такої таблиці (табл. 2.10) записані значення Z_i і Y_j . Помітимо, що для асинхронних ПП, якщо наступний внутрішній стан при зміні X залишається таким же, як і попередній, тобто $f(X_i, Z_j) = Z_j$, він є стійким, і в таблиці переходів номер цього стану беруть у дужки. У протилежному випадку внутрішній стан є нестійким і його позначають цифрою без дужок. Розглянемо табл. 2.11.

Таблиця 2.11 – Сійкі та несійкі стани

	X_1	X_2	X_3	X_4
1	(1)	2	3	-
2	1	(2)	(2)	3
3	4	-	(3)	(3)

Якщо ПП знаходиться в сійкому стані (2) і стан його входу змійується з X_2 на X_3 , то внутрїшній стан при цьому не змійується. У випадку, коли вїдбувається змїна X_2 на X_1 пристрїй спочатку прийме несійкий стан 1, а потїм сійкий (1), при цьому його внутрїшній стан змійується з Z_2 на Z_1 .

Таким чином, змїна внутрїшнього стану ПП завжди пов'язана з переходом його через несійкий стан. Перехїд з одного внутрїшнього стану ПП в їнший через сійкий стан неможливий.

Мїнїмїзація числа станїв. При побудовї таблиць переходїв може виявитися бїльше станїв, нїж необхідно для нормального функцїонування ПП. У зв'язку з цим виникає задача мїнїмїзацїї числа станїв. При скороченнї числа станїв надлишковими станами вважають такї, котрим у таблицї переходїв вїдповїдають рядки з несуперечливим розмїщенням цифр. Такї рядки в таблицї переходїв мають однаковї цифри в якому-небудь стовпцї, або ж в одному рядку цього стовпця стоїть цифра, а в їншому – прочерк. Цї рядки називають сумїсними. Об'єднання сумїсних рядкїв приводить до зменшення числа внутрїшнїх змїнних, необхідних для опису всїх переходїв. При об'єднаннї рядкїв кожнїй групї сумїсних рядкїв присвоюють нову цифру й роблять пере позначення станїв ПП. Зазначимо, що при виконаннї операцїї об'єднання рядкїв значення вихїдних функцїй до уваги не беруться. Отриману пїсля об'єднання рядкїв нову таблицю називають скороченою таблицю переходїв. Для оптимїзацїї операцїї об'єднання рядкїв будують вїдповїдну дїаграму.

Для прикладу вїзьмемо табл. 2.12. При побудовї дїаграми розмїстимо номери рядкїв по деякому колу (рис. 2.13).

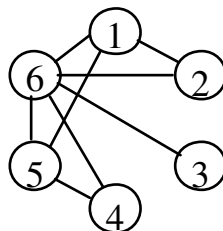


Рисунок 2.13 – Дїаграма об'єднання рядкїв

Можливїсть попарного об'єднання окремих рядкїв умовно позначимо лїнїями мїж вїдповїдними номерами. Кожен рядок необхідно порївняти з усїма їншими для виявлення можливостї їхнього об'єднання. Аналїз дїаграми

дозволяє виявити оптимальну схему об'єднання рядків, при якій мінімізується загальне число рядків у таблиці. Наприклад, поєднуючи рядки 1 та 2, 3 та 6, а також 4 та 5, одержимо скорочену таблицю переходів (див. табл. 2.13).

Таблиця 2.12 – Таблиця переходів

x_1x_2	00	01	11	10
1	(1)	–	–	2
2	–	–	3	(2)
3	–	–	(3)	4
4	–	–	5	(4)
5	–	6	(5)	–
6	1	(6)	–	–

Таблиця 2.13 – Таблиця переходів

x_1x_2	00	01	11	10
(1)	–	3	(1)	
1	(3)	(3)	4	
–	3	(4)	(4)	

Оскільки p внутрішніх змінних можуть утворити 2^p наборів їхніх значень і в ПП з таким числом внутрішніх змінних можна реалізувати 2^p станів, то виникає задача кодування значень внутрішніх змінних.

Кодування значень внутрішніх змінних полягає в присвоєнні двійкового коду кожному стану ПП, тобто кожному рядку скороченої таблиці, або кожному набору значень змінних Z_i у даний момент часу. Так, для реалізації чотирьох станів ПП будуть потрібні дві внутрішні змінні (позначимо їх Z_1 та Z_2). Задачу кодування внутрішніх змінних зручно вирішувати за допомогою спеціальної карти кодування (табл. 2.14), де код стану збігається з двійковим номером відповідної клітинки. Розташовуючи, наприклад, стан 1 послідовно у кожній клітинці і кожного разу записуючи послідовність станів на карті, дотримуючись їхнього розміщення в напрямку за годинною стрілкою, одержимо чотири варіанти кодування. При запису послідовності станів на карті в напрямку проти годинникової стрілки будемо мати ще чотири варіанти кодування.

Доповнюючи скорочену таблицю переходів пристрою кодом станів відповідно до обраного варіанта кодування, запишемо кодовану таблицю переходів (табл. 2.15). ПП можуть бути подані діаграмою чи графом, що складається з вузлів, які з'єднані галузями. Позначивши стан ПП вузлами, а переходи ПП, які одержані під впливом вхідних сигналів X_i – галузями, отримаємо діаграму станів. Так, діаграма станів, що відповідає таблиці станів (табл. 3.8), наведена на рис. 2.14. На галузях графа можуть бути зазначені значення вихідних сигналів.

Діаграма станів ПП може бути використана для визначення виду вихідних змінних при довільних вхідних для будь-якого початкового стану пристрою. Наприклад, для ПП, який подано графом на рис. 2.14, якщо має місце перший початковий стан, а вхідна послідовність $X = 00101$, то утвориться послідовність його станів такого виду 22434 при послідовності на виході $Y = 11010$. Та ж вхідна послідовність, яка впливає на роботу ПП при

третьому початковому стані, приводить до вихідної послідовності виду 01010.

Таблиця 2.14 – Таблиця переходів

z_1		0	1
z_2	0	2	3
	1	1	4

Таблиця 2.15 – Таблиця переходів

x	0	1	$z_1 z_2$
1	(1)	2	10
2	3	(2)	00
3	(3)	4	01
4	1	(4)	11

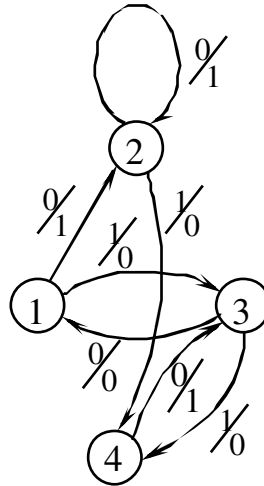


Рисунок 2.14 – Діаграма станів ПП

Порівнюючи табличний та графічний способи завдання ПП, можна відзначити наступне:

- табличний спосіб є найбільш повним, але неприйнятним для великого числа станів. У той же час даний спосіб дозволяє здійснити порівняно простий перехід до формульного запису функцій Y та Z ;
- графічний спосіб більш наочний. Особливо його переваги відчуються при розв'язанні задач по виявленню можливих циклічних режимів роботи ПП.

У процесі проектування ПП може бути виконаний перехід від графа до таблиці і навпаки.

Приклад формального опису ПП. Нехай потрібно синтезувати послідовний двійковий суматор. Числа вводяться в суматор послідовно розряд за розрядом, синхронно з тактовим сигналом. У кожному такті обчислюється значення суми з урахуванням одиниць переносу. При цьому в даному суматорі повинна запам'ятовуватися на один такт одиниця переносу в старший розряд.

Відповідно до цього опису ПП повинен мати два входи, на які подаються сигнали (числа) x_1 і x_2 , один вихід Y і один елемент пам'яті для запису сигналу переносу. Отже, ПП може мати:

- чотири стани входу $X = (x_1, x_2)$: $X_1 = (0, 0)$, $X_2 = (0, 1)$, $X_3 = (1, 0)$,

$X_4 = (1,1)$ чи $X = \{ 00, 01, 10, 11 \}$;

- два стани виходу $y_1 = 0, y_2 = 1$, тобто $Y = \{0,1\}$;
- два внутрішніх стани, $Z_1 = 0, Z_2 = 1$, тобто $Z = \{0,1\}$.

Оскільки сигнал переносу є водночас і внутрішньою і вихідною змінною, замість Z використовується позначення Q . Функції переходів та виходів даного ПП мають вигляд

$$Q^n = f(x_1^n, x_2^n, Q^{n-1}),$$

$$Y^n = \varphi(x_1^n, x_2^n, Q^{n-1}).$$

Суматор виконує операцію складання x_1 і x_2 з урахуванням переносу Q . Тому $Q^n = 1$ у тих випадках, коли два сигнали x_1^n, x_2^n рівні 1, а $Y^n = 1$ тільки при непарному числі одиниць в сигналах x_1^n, x_2^n, Q^{n-1} . Складемо таблицю станів ПП (табл. 2.16). Кожен рядок таблиці відповідає одному з можливих станів Z у n -такті, а кожен стовпець – одному з можливих станів входу X у такті n . У клітках, що відповідають рядкам Z_i і стовпцям X_j , вказують стан ПП $Q^n = Z^n$, у який він переходить, а також і стан виходу Y^n .

Таблиця 2.16 – Таблиця переходів

$Z_i^n \backslash x_j^n$	x_1^n	x_2^n	x_3^n	x_4^n
Z_1^n	Z_1, y_1	Z_1, y_2	Z_1, y_2	Z_2, y_1
Z_2^n	Z_1, y_2	Z_2, y_1	Z_2, y_1	Z_2, y_2

Таблиця 2.17 – Таблиця переходів

$i \backslash x_1, x_2$	00	01	10	11	Q
1	1,0	1,1	1,1	2,0	0
2	1,1	2,0	2,0	2,1	1

Таблиця 2.18 – Таблиця переходів

x_1^n	x_2^n	Q^n	Q^{n+1}	y^n
0	0	1	0	0
0	0	0	0	1
0	1	1	0	1
0	1	0	1	0
1	0	1	0	1
1	0	0	1	0
1	1	1	1	0
1	1	0	1	1

У більшості практичних задач кодування станів входу і виходу за допомогою наборів значень вхідних і вихідних сигналів впливає із самої постановки задачі. Тому в таблиці станів часто зручно вказувати набори значень цих сигналів (див. табл. 2.15). Доповнюючи табл. 2.16 стовпцем Q , у якому записується прийняте кодування станів ПП, перейдемо до табл. 2.17. Графи переходів такого пристрою, побудовані відповідно до таблиць 2.16 та 2.17, наведені на рис. 2.15 (рис. 2.15а – за табл. 2.16, а рис. 2.15б – за табл.

2.17). За допомогою цих таблиць можна також скласти таблицю істинності суматора (табл. 2.18).

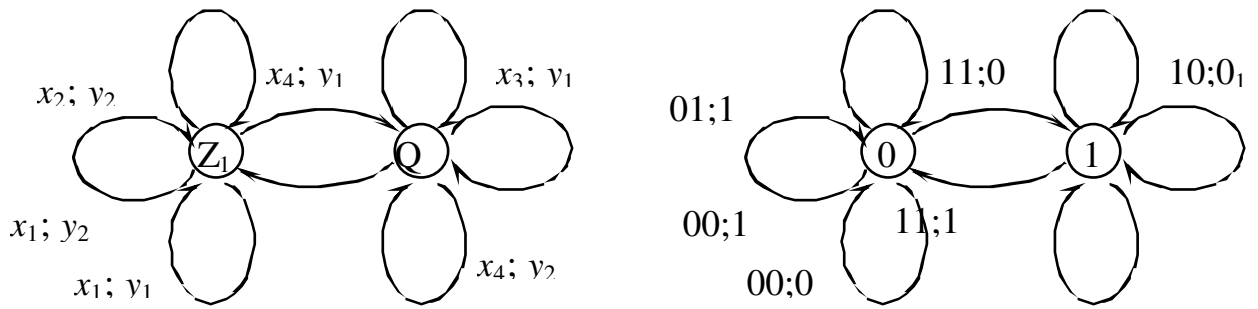


Рисунок 2.15 – Графи переходів суматора:

а – з позначенням станів; *б* – з кодуванням станів

2.6 Канонічний метод проектування послідовнісних пристроїв

Побудова первинного графа переходів, чи складання первинної таблиці переходів на підставі словесного опису є першим етапом проектування ПП. При наявності словесної постановки задачі, тобто словесного опису роботи ПП первинний граф переходів ПП можна побудувати, використовуючи для цього наступний порядок дій.

1. Визначити вхідні та вихідні сигнали проєктованого ПП і побудувати його структурну схему, розміщаючи входи ліворуч, а виходи праворуч (проєктовані ПП можуть являти собою спеціально виділені модулі – вузли) так, що вихідні сигнали одного модуля пристрою будуть вхідними сигналами іншого.

2. Виявити причинно-наслідкові співвідношення між вхідними та вихідними сигналами.

3. Визначити стан, з якого починається послідовність подій. Для цього можна, наприклад, виявити стан, при якому схема знаходиться в "очікуванні", чи стан, бажаний для переходу схеми за командою початкової установки. На етапах усім внутрішнім змінним цього стану у звичайних випадках привласнюють нульові значення.

4. Визначити черговість подій (станів) у схемі, починаючи з моменту подачі вхідних сигналів.

5. Знайти число переходів (ліній) з кожного стану (вузла). Якщо при перебуванні схеми в певному стані на неї можуть надходити залежно від зовнішніх умов різні вхідні сигнали, то у відповідний вузол графа повинні входити декілька ліній. Аналіз переходів варто продовжувати до того моменту, поки послідовність станів не утворить цикл (замкнутий граф). Поява ж тупикового стану, як правило, свідчить про помилку при складанні графа переходів.

6. Якщо на першому етапі проектування було отримано первинний граф переходів, то за цим графом необхідно скласти таблицю переходів.

7. Далі проводять мінімізацію числа станів та будують скорочену таблицю переходів і скорочений граф переходів. Після цього здійснюють кодування внутрішніх змінних й одержують кодовану таблицю.

8. На наступному етапі проектування будують карти станів і знаходять функції порушення внутрішніх змінних. Для цього привласнюють рядкам кодованої таблиці переходів замість десяткових цифр двійкові (рис. 2.16).

9. Складаючи карти вихідних змінних, якщо вони не збігаються із внутрішніми змінними, знаходимо функції вихідних змінних.

10. Заключним етапом проектування ПП є побудова структурної схеми або структурний синтез ПП. Розглянемо словниковий метод структурного синтезу [1], суть якого полягає в наступному.

		Z_1Z_2			
		00	01	11	10
x	0	3	(3)	1	(1)
	1	(2)	4	(4)	2

		Z_1Z_2			
		01	11	10	00
x	0	01	01	10	10
	1	00	11	11	00

Рисунок 2.16 – Карти Карно с кодованими переходами:
a – десятковий код; *b* – двійковий код

Карта станів Q тригера (рис. 2.16) перетвориться в таблицю функцій переходів F_Q (табл. 2.19) внутрішніх змінних, де $F_Q \in Q^n \rightarrow Q^{n+1}$, причому умовні позначки переходу внутрішніх змінних з $0 \rightarrow 0$; $1 \rightarrow 1$; $1 \rightarrow 0$ і $0 \rightarrow 1$, відповідно: 0, 1, Δ .

Таблиця 2.19 – Таблиця переключень

F_Q	S	R
0	1	X
1	X	1
Δ	0	1
Δ	1	0

Потім складаємо карту функцій зрушення кожного тригера. Для цього символи функцій переходів у карті функцій переходів заміняють значеннями вхідних сигналів тригера (наприклад, \bar{R} та \bar{S}), які необхідні для реалізації даного переходу і визначаються по словнику переходів обраного тригера (рис. 2.17). Отримані карти представлені в табл. 2.20.

Далі, визначивши на підставі карт функції збудження, треба провести їхнє перетворення для реалізації в заданому базисі, наприклад, у базисі І-НІ.

$$\begin{aligned} \bar{S}_1 &= \bar{x} + \bar{Q}_2 = \overline{x \cdot Q_2}; & \bar{R}_1 &= \bar{x} + Q_2 = \overline{x \cdot \bar{Q}_2}; \\ \bar{S}_2 &= x + Q_1 = \overline{x \cdot \bar{Q}_1}; & \bar{R}_2 &= x + \bar{Q}_1 = \overline{x \cdot Q_1}. \end{aligned}$$

Таблиця 2.20 – Таблиця переключень

Внутрішній стан	Стан в ходів CD				Стан виходу Z
	00	01	11	10	
1	(1)				0
2		(2)			0
3			(3)		0
4				(4)	0
5	(5)				1
6		(6)			1
7			(7)		1
8				(8)	1

Отримані формули дозволяють побудувати таку структурну схему пористою (рис. 2.19, а). Оскільки карти функцій порушень (табл. 2.21) містять невизначені значення цих функцій, то є можливість скласти ряд інших варіантів структурної реалізації синтезованого ПП. Потім провести їхній аналіз і вибрати кращий з огляду на функціональну надійність, простоту реалізації і т.д. У цьому полягає перевага словникового методу структурного синтезу.

Проілюструємо сказане одним із прикладів. Довизначаємо карти функцій \bar{S}_2 і \bar{R}_2 (рис. 2.18), прийнявши значення функції \bar{S}_2 в клітинках із двійковими номерами 001 й 001 та значення функції \bar{R}_2 в клітинках 010 й 110, які дорівнюють нулю. Тоді маємо таке

$$\bar{S}_2 = Q_1 + x \cdot \bar{Q}_2 = Q_1 + R_1 = \overline{\bar{Q}_1 \cdot \bar{R}_1};$$

$$\bar{R}_2 = \bar{Q}_1 + x \cdot Q_2 = \bar{Q}_1 + S_1 = \overline{Q_1 \cdot \bar{S}_1}.$$

$Q_1 Q_2$

x	00	01	11	10
0	0Δ	01	1	10
Δ	00	Δ1	11	0

Рисунок 2.17 – Карта станів тригера

	00	01	11	10	
0	1	1	X	X	\bar{S}_1
1	1	0	X	1	
0	X	X	1	1	
1	X	1	1	0	

	00	01	11	10	
0	0	X	1	1	\bar{S}_2
1	1	X	X	1	
0	1	1	0	X	
1	X	1	1	X	

	00	01	11	10	
0	0	X	1	1	\bar{R}_1
1	1	X	X	1	
0	1	1	0	X	
1	X	1	1	X	

	00	01	11	10	
0	0	X	1	1	\bar{R}_2
1	1	X	X	1	
0	1	1	0	X	
1	X	1	1	X	

Рисунок 2.18 – Карти функцій тригера

У ПП, карта якого наведена на рис. 2.19а, можуть виникнути небезпечні змагання сигналів по таким рівнобіжним ланцюгам: входи $D9 - \bar{X}$, $D1 - D5 - Q_1$ та $D2 - D6 - \bar{Q}_1$. Так, наприклад, у випадку, коли вхідна змінна X змінює своє значення з 0 на 1, сигнал \bar{X} може затримати зміни сигналу Q_1 на

виходах $D5$ і $D6$. Це може викликати неалгоритмічний перехід та зміну вихідних змінних Q_2 на \bar{Q}_2 . У другому варіанті ПП (рис. 2.19б) такі небезпечні змагання відсутні. Крім того, цей ПП містить на один логічний елемент менше, ніж аналогічний ПП, який показано на рис. 2.19а.

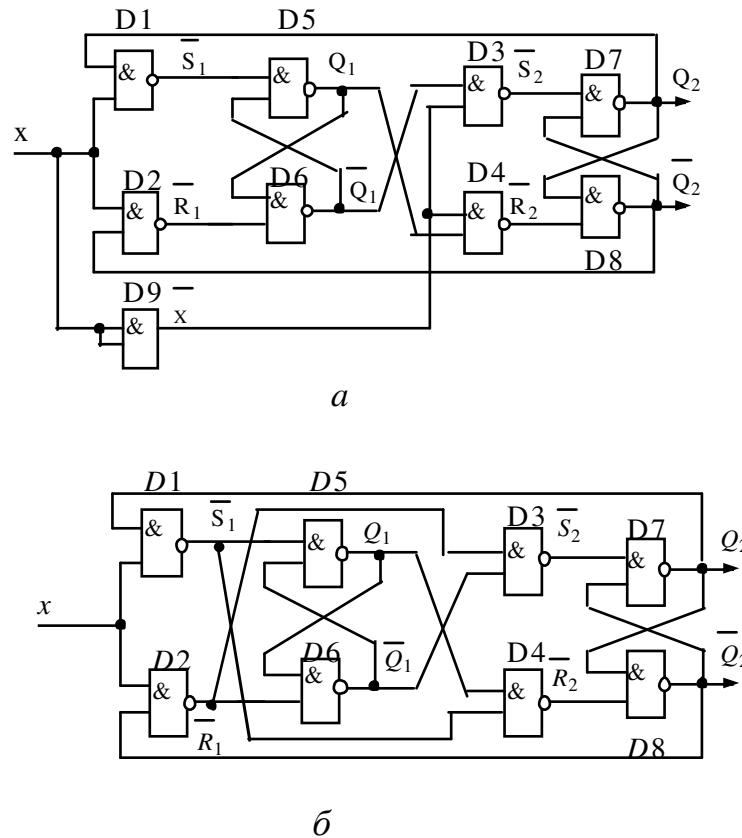


Рисунок 2.19 – Структурні варіанти тригера і варіант усунення змагань:
а – з можливістю змагань; *б* – з усуненням змагань

Контрольні запитання

1. Перерахуйте універсальні логічні елементи.
2. У чому полягає табличний метод проектування комбінаційних пристроїв?
3. Що таке графічний метод синтезу цифрового пристрою?
4. Які види змагань сигналів можливі в комбінаційних пристроях?
5. Назвіть методи боротьби зі змаганням сигналів.
6. Які ви знаєте методи формального опису послідовнісних пристроїв?
7. Особливості канонічного методу синтезу цифрових пристроїв.

3 ТИПОВІ ПРИСТРОЇ СИСТЕМ КЕРУВАННЯ

3.1 Дешифратори, шифратори та перетворювачі кодів

Дешифратором називають пристрій, що перетворює m – розрядний двійковий код в однозначно відповідний йому унітарний. У дешифраторі (рис. 3.1) кожному значенню вхідного m – розрядного двійкового коду відповідає сигнал 1, яка з'являється тільки на тому виході, номер якого дорівнює значенню вхідного числа. При цьому на всіх інших виходах дешифратора сигнал 0. Дешифратор має $K = 2^m$ виходів. Дешифратори знайшли застосування в системах керування електричним транспортом при поданні вимірювальної інформації у цифровому вигляді. Розглянемо приклади побудови дешифратора на елементах базису І, ІІ, АБО.

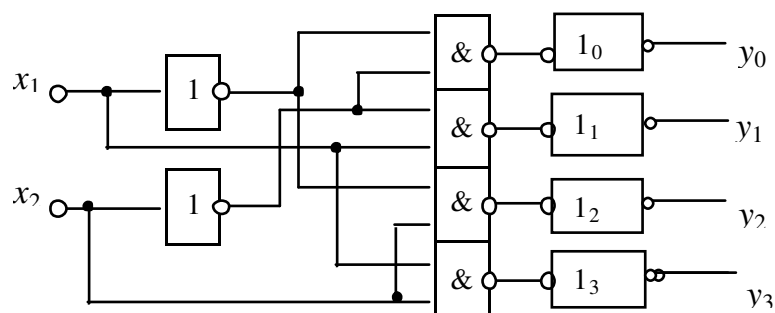


Рисунок 3.1 – Схема дешифратора на чотири виходи

Приклад 1. Прийнемо $m = 2$, $K = 2^m = 4$. За табл. 3.1 одержимо такі вирази $y_0 = \bar{x}_1 \cdot \bar{x}_2$, $y_1 = x_1 \cdot \bar{x}_2$, $y_2 = \bar{x}_1 \cdot x_2$, $y_3 = x_1 \cdot x_2$.

Таблиця 2.1 – Таблиця істинності дешифратора

x_2	x_1	y_0	y_1	y_2	y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Схема дешифратора, яка приведена на рис. 3.1, реалізує такі логічні функції

$$y_0 = \bar{x}_1 \cdot \bar{x}_2 = \overline{\overline{\bar{x}_1 \cdot \bar{x}_2}}, \quad y_1 = x_1 \cdot \bar{x}_2 = \overline{\overline{x_1 \cdot \bar{x}_2}},$$

$$y_2 = \bar{x}_1 \cdot x_2 = \overline{\overline{\bar{x}_1 \cdot x_2}}, \quad y_3 = x_1 \cdot x_2 = \overline{\overline{x_1 \cdot x_2}}.$$

Приклад 2. Розглянемо проектування дешифратора, що має шість вхідів і 64 виходи. Таблиця істинності дешифратора (табл. 3.2) дозволяє записати такі вирази алгебри логіки

$$y_0 = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot \bar{x}_5 \cdot \bar{x}_6 = \overline{x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6},$$

$$y_1 = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot \bar{x}_5 \cdot \bar{x}_6 = x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot \bar{x}_5 \cdot \bar{x}_6,$$

.....

$$y_{63} = x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6 = \overline{\overline{x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6}}.$$

Таблиця 3.2 – Таблиця істинності дешифратора

x_6	x_5	x_4	x_3	x_2	x_1	y_0	y_1	$y_2 \dots y_{62}$	y_{63}
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	1	1	0
–	–	–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–	–	–
1	1	1	1	1	1	0	0	0	1

Для реалізації такого дешифратора потрібно 64 корпусів ІС 8-ми входових елементів І-НІ та 16 корпусів ІС, які містять по 4 двовходових елемента І-НІ. Зменшити кількість корпусів ІС можливо шляхом переходу до двохступінчастої схеми, у якій на першій та другій ступені використовуються дешифратори ДС1 та ДС2 з $m = 3$. В табл. 3.3 дано опис дешифратора ДС 1.

Таблиця 3.3 – Таблиця істинності

x_3	x_2	x_1	y'_0	y'_1	y'_2	y'_7
0	0	0	1	0	0	0
0	0	1	0	1	0	0
–	–	–	–	–	–	–
1	1	1	0	0	0	1

Структурні формули ДС1 мають наступний вигляд

$$y'_0 = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 = \overline{x_1 \cdot x_2 \cdot x_3},$$

$$y'_1 = x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 = \overline{x_1 \cdot x_2 \cdot x_3},$$

.....

$$y'_7 = x_1 \cdot x_2 \cdot x_3 = \overline{\overline{x_1 \cdot x_2 \cdot x_3}}.$$

Аналогічно для другої ступені дешифратора ДС2 маємо такі вирази для функцій алгебри логіки

$$y_0'' = \bar{x}_4 \cdot \bar{x}_5 \cdot \bar{x}_6 = \overline{\overline{\bar{x}_4 \cdot \bar{x}_5 \cdot \bar{x}_6}},$$

$$y_1'' = x_4 \cdot \bar{x}_5 \cdot \bar{x}_6 = \overline{\overline{x_4 \cdot \bar{x}_5 \cdot \bar{x}_6}},$$

.....

$$y_7'' = x_4 \cdot x_5 \cdot x_6 = \overline{\overline{x_4 \cdot x_5 \cdot x_6}}.$$

Поєднання ступенів дешифратора дає такі вирази

$$y_0 = y_0' \cdot y_0'' = \overline{\overline{y_0' \cdot y_0''}},$$

$$y_1 = y_1' \cdot y_0'' = \overline{\overline{y_1' \cdot y_0''}},$$

.....

$$y_{63} = y_7' \cdot y_7'' = \overline{\overline{y_7' \cdot y_7''}}$$

Побудова такого двохступінчастого дешифратора наведена на рис. 3.2.

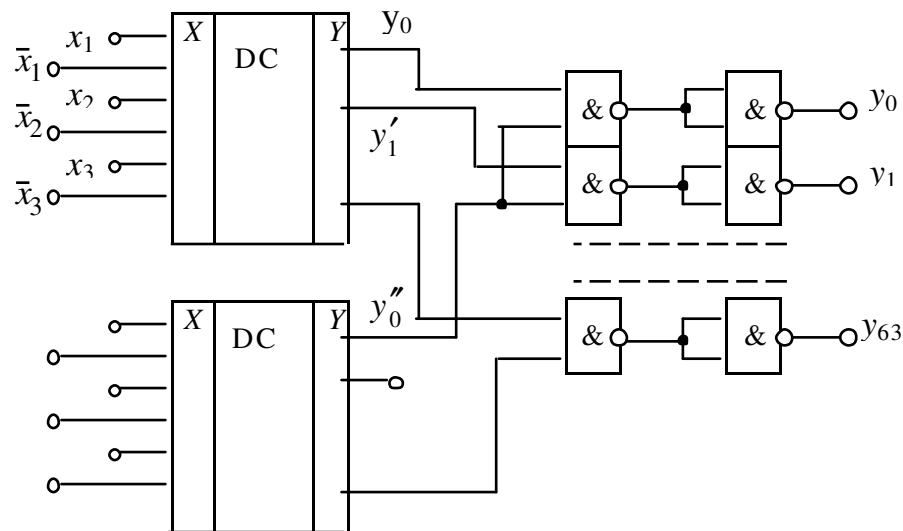


Рисунок 3.2 – Двоступінчастий дешифратор на 64 виходи

Шифратор перетворює унітарний код на вході в l -розрядний двійковий код на виході. Тобто, виконує функцію зворотню дешифратору. При наявності сигналу 1 на одному з входів на l виходах шифратора з'являється l -розрядний код, значення якого дорівнює номеру входу, де присутній сигнал 1. Число входів $m+1$ та виходів l шифратора пов'язано зі співвідношенням $2^l = m+1$. Наприклад, при $m = 3$, $l = 2$. Функціонування шифратора описується таблицею істинності (табл. 3.4).

Запишемо рівняння для цього прикладу

$$y_0 = x_1 + x_3, \quad y_1 = x_2 + x_3.$$

Реалізація такого шифратора на елементах АБО приведена на рис. 3.3. Шифратори у системах керування електричним транспортом найбільш поширені як кодоперетворювачі двійкового коду в двійково-десятичний та

навпаки, а також двійкового коду в код Грея та навпаки. Такі перетворювачі випускають серійно у вигляді постійних запам'ятовуючих пристроїв (К155 ПР6, К155 ПР7).

Таблиця 3.4 – Таблиця істинності

x_3	x_2	x_1	x_0	y_1	y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

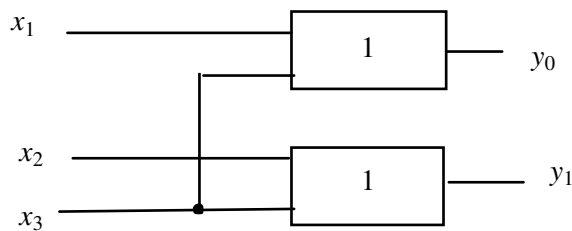


Рисунок 3.3 – Схемна реалізація шифратора

Перетворювачі кодів організуються послідовним включенням дешифратора та шифратора. Схемну реалізацію цих пристроїв розглянемо на прикладі 2-розрядного перетворювача коду Грея у двійковий. Для побудови пристрою складаємо таблицю істинності (табл. 3.5).

Таблиця 3.5 – Таблиця істинності перетворювача кодів

Код Грея		N	Двійковий	
x_1	x_2		y_1	y_2
0	0	0	0	0
1	0	1	1	0
1	1	2	0	1
0	1	3	1	1

З аналізу табл. 3.5 видно, що для реалізації перетворювача кодів варто виконати відповідні з'єднання виходів дешифратора та входів шифратора. Схема пристрою наведена на рис. 3.4.

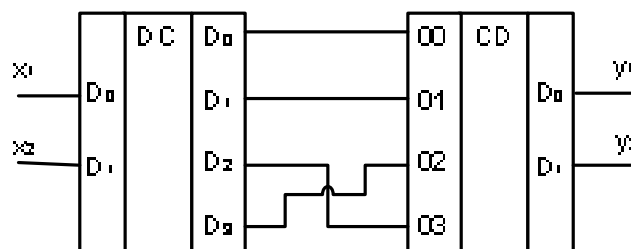


Рисунок 3.4 – Схемна реалізація перетворювача

3.2 Схеми порівняння кодів та контролю парності

Схеми порівняння кодів будуються на логічних елементах рівнозначності та нерівнозначності. Розглянемо метод синтезу схеми однорозрядного порівняння кодів на прикладі елементів І-НІ. Будемо таблицю істинності роботи пристрою (табл. 3.6).

Таблиця 3.6 – Таблиця істинності схеми порівняння кодів

x_1	x_2	y
0	0	1
1	0	0
0	1	0
1	1	1

На підставі цієї таблиці записуємо рівняння алгебри логіки роботи пристрою, перетворення якого до потрібного базису виконуємо за допомогою правила де Моргана

$$y = \overline{x_1 x_2} + x_1 x_2 = \overline{x_1} \overline{x_2} + x_1 x_2.$$

На основі отриманого виразу будемо електричну принципову схему порівняння кодів (рис. 3.5).

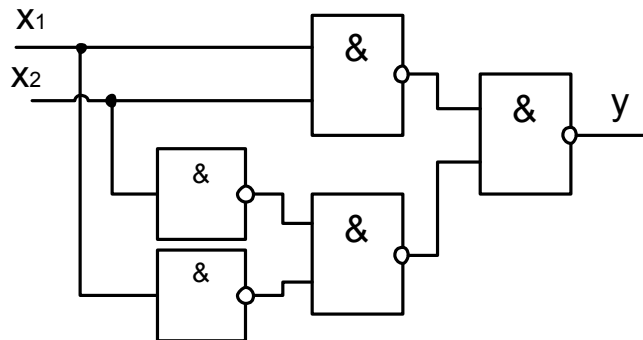


Рисунок 3.5 – Схемна реалізація пристрою порівняння кодів

Побудова схем порівняння багаторозрядних кодів залежить від способу кодування, від організації вхідних та вихідних сигналів, а також методу контролю. При проектуванні таких пристроїв можливо безліч варіантів їхньої схемної реалізації. Загальним підходом до розв'язування цієї задачі є приведення вхідних кодів до єдиного вигляду, вибір методу контролю та складання рівняння алгебри логіки для вихідного сигналу.

Схеми контролю парності використовуються у системах управління електричним транспортом для виявлення похибок при прийомі та обробці цифрової інформації. Розглянемо метод синтезу цих пристроїв на прикладі контролю парності двох біт. Будемо таблицю істинності роботи пристрою (табл. 3.7). У якості базового елемента для побудови принципової схеми пристрою застосуємо І-НІ.

Таблиця 3.7 – Таблиця істинності схеми контролю парності

x_1	x_2	ПАР	НЕПАР
0	0	1	0
1	0	0	1
0	1	0	1
1	1	1	0

На підставі цієї таблиці записуємо рівняння алгебри логіки роботи пристрою, перетворення якого до потрібного базису виконуємо за допомогою правила де Моргана

$$ПАР = x_1x_2 + x_1\bar{x}_2 = x_1x_2 \bar{x}_1\bar{x}_2.$$

На основі отриманого виразу будуємо електричну принципову схему порівняння кодів (рис. 3.6).

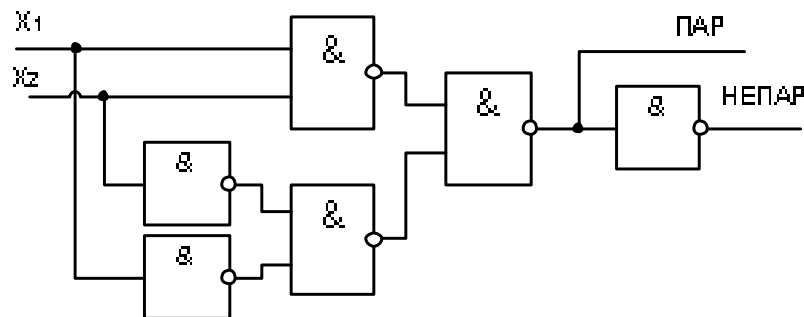


Рисунок 3.6 – Схемна реалізація пристрою контролю парності

3.3 Пристрої для підсумовування

Базовим елементом, на основі якого в системах управління електричним транспортом будуються схеми для підсумовування, є однорозрядний суматор. Ці логічні елементи бувають двох типів: повні та неповні.

Неповний суматор реалізує функцію підсумовування двох біт інформації з видачею сигналів суми S та переносу p , який виникає при підсумовуванні. Будуємо таблицю істинності роботи суматора (табл. 3.8). У якості базового елементу для побудови принципової схеми пристрою застосуємо І-ІІ.

Таблиця 3.8 – Таблиця істинності суматора

x_1	x_2	S	p
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

На підставі цієї таблиці записуємо рівняння алгебри логіки роботи пристрою, перетворення якого до потрібного базису виконуємо за допомогою правила де Моргана

$$S = \overline{x_1 x_2} + \overline{x_1} x_2 = \overline{x_1 x_2 x_1 x_2}, \quad p = x_1 x_2.$$

На основі отриманих виразів будуємо електричну принципову схему неповного суматора (рис. 3.7).

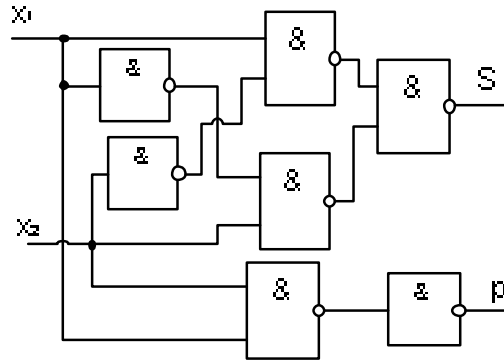


Рисунок 3.7 – Схемна реалізація неповного суматора

Повний суматор реалізує функцію підсумовування двох біт інформації та сигналу переносу p_{i-1} від попереднього підсумовування з видачею сигналів суми S та переносу p , який виникає при цьому. Будуємо таблицю істинності роботи суматора (табл. 3.9). У якості базового елемента для побудови принципової схеми пристрою застосуємо І-НІ.

Таблиця 3.9 – Таблиця істинності повного суматора

x_1	x_2	p_{i-1}	S	p
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

На підставі цієї таблиці записуємо рівняння алгебри логіки роботи пристрою, перетворення якого до потрібного базису виконуємо за допомогою правила де Моргана. Мінімізацію виразу для сигналу переносу виконуємо за допомогою методу Квайна

$$S = \overline{x_1 x_2 p_{i-1}} + \overline{x_1} x_2 p_{i-1} + x_1 \overline{x_2} p_{i-1} + x_1 x_2 \overline{p_{i-1}} =$$

$$= \overline{x_1 x_2 p_{i-1} x_1 x_2 p_{i-1} x_1 x_2 p_{i-1} x_1 x_2 p_{i-1}},$$

$$p = x_1 x_2 + x_1 p_{i-1} + x_2 p_{i-1} = \overline{x_1 x_2 x_1 p_{i-1} x_2 p_{i-1}}.$$

На основі отриманих виразів будемо електричну принципову схему неповного суматора (рис. 3.8).

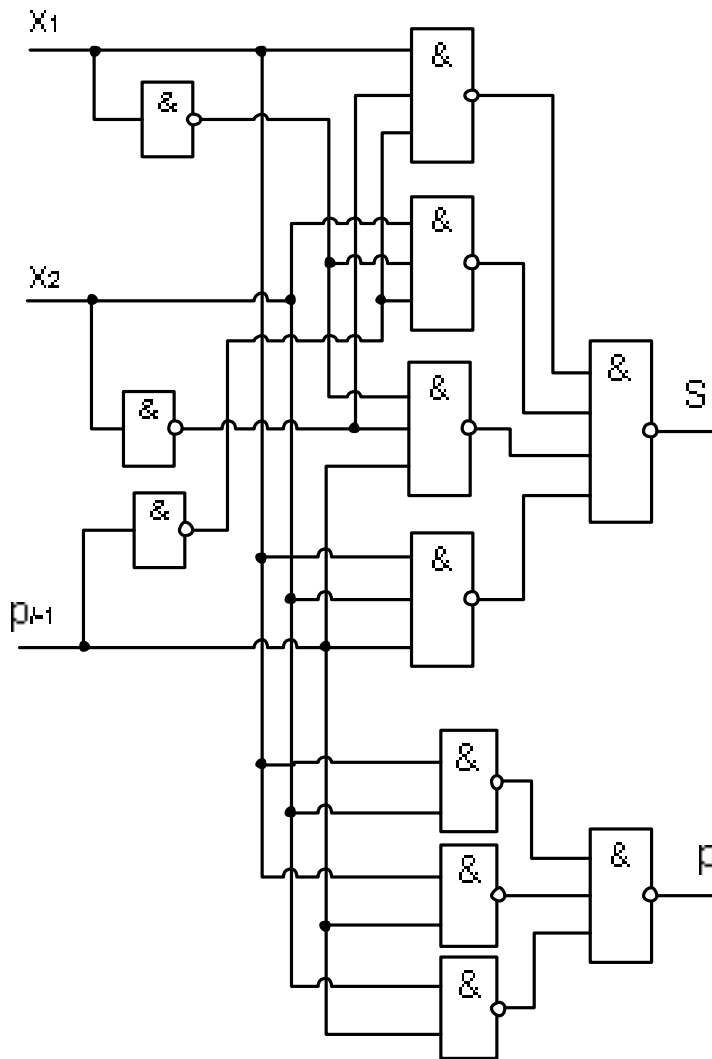


Рисунок 3.8 – Схемна реалізація повного суматора

3.4 Мультиплексори та демультимплексори

Особливу групу КП у системах керування електричним транспортом складають логічні комутатори. Це такі схеми, які призначені для вибіркового переключення вхідних каналів [2...6]. До них відносяться мультиплексори.

Мультиплексор – це цифровий пристрій, що залежно від значення коду на адресних входах (A_0, A_1, \dots, A_M) здійснює вибір одного з інформаційних входів (D_0, D_1, \dots, D_N) та забезпечує спрямовану передачу його сигналу на вихід Q . Мультиплексор можна уявити як безконтактний багатопозиційний перемикач.

На рис. 3.9а вказано умовне позначення мультиплексору на ІМС, яка має вхід управління C . Кожному інформаційному входу мультиплексору привласнюється номер, який називається його адресою. При появі на управляючому вході C сигналу логічної одиниці, мультиплексор підключає

інформаційний вхід до виходу. Адреса цього входу відповідає двійковому коду, який подано на адресні входи. Змінюючи сигнали на адресних входах мультиплексу при наявності на управляючому вході дозволяючого сигналу, можна забезпечити передачу сигналів із різних інформаційних входів на його вихід. Число інформаційних входів N мультиплексу та кількість його адресних входів M пов'язані таким співвідношенням

$$N = 2^{M+1}.$$

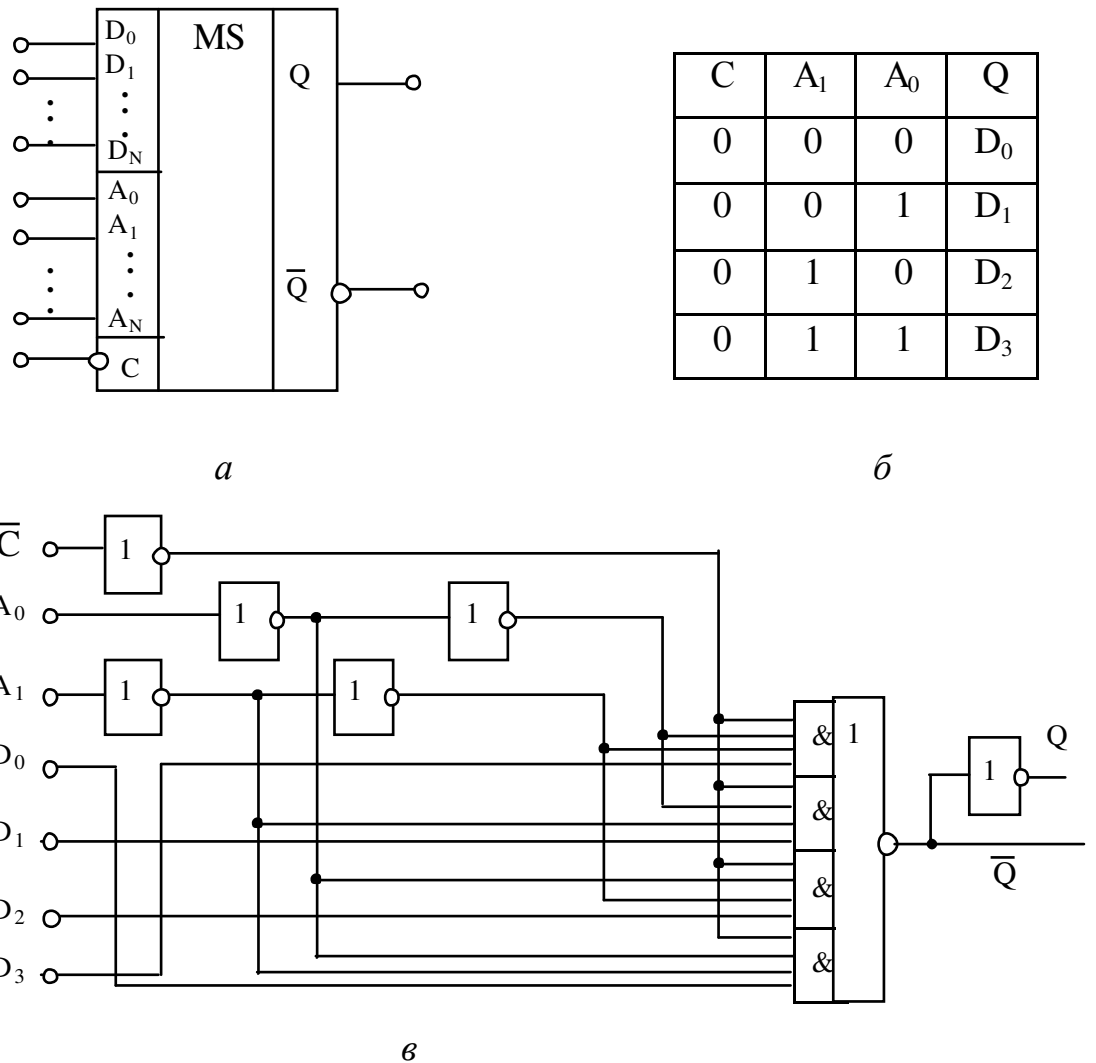


Рисунок 3.9 – Мультиплексор:

а – схематичне зображення; б – опис роботи; в – структурна побудова

Роботу мультиплексу, наприклад, із двома адресними і чотирма інформаційними входами, можна описати таблицею (рис. 3.9б). При відсутності дозволяючого сигналу ($\bar{C} = 1$) зв'язок між інформаційними входами та виходом відсутній (при цьому $Q = 0$). При наявності дозволяючого сигналу ($\bar{C} = 0$) на вихід передається логічний рівень того з інформаційних входів D_i , номер якого у двійковій формі заданий сигналами на його адресних входах. Так, якщо $A_1\bar{A}_0 = 10_2 = 2_{10}$, на вихід Q буде

передаватися сигнал із входу D_2 . За таблицю можна записати логічну функцію виходу мультиплексу

$$Q = (D_0 \cdot \bar{A}_1 \cdot \bar{A}_0 + D_1 \cdot \bar{A}_1 \cdot A_0 + D_2 \cdot A_1 \cdot \bar{A}_0 + D_3 \cdot A_1 \cdot A_0) \cdot C.$$

Структурна схема такого мультиплексу наведена на рис. 3.9в. Мультиплекси можуть бути використані для синтезу КП, які реалізують логічні функції багатьох змінних. При цьому мультиплексор працює як універсальний ЛЕ. Він реалізує будь-яку логічну функцію, яка містить у собі до $M + 1$ змінних, де M – число адресних входів мультиплексу [2, 3]. Один мультиплексор у цьому випадку заміняє кілька корпусів з логічними елементами І, АБО, НІ.

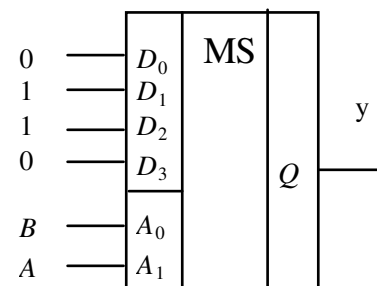
При синтезі КП на мультиплексах треба скласти карту Карно заданої функції. Число інформаційних входів мультиплексу вибирають рівним числу клітинок карти Карно. Якщо двійкові коди на адресних входах мультиплексу та коди клітинок карт Карно збігаються, а інформаційні сигнали визначаються значеннями 0 чи 1, то такий мультиплексор цілком відтворює карту Карно, тобто заданий комбінаційний пристрій. Розглянемо приклади застосування мультиплексу при синтезі КП і деякі прийоми підвищення ефективності їхнього використання.

Приклад 1. Задана функція нерівнозначності, яка описується виразом $y = A \cdot \bar{B} + \bar{A} \cdot B$ та картою Карно (рис. 3.10а). У правому нижньому куті кожної клітинки карти позначимо відповідний інформаційний вхід мультиплексу. Логічні рівні 0 та 1, записані в клітинках карти, привласнимо значенням вхідних сигналів D_i ($i = 0, \dots, 3$) мультиплексу. Якщо на входи мультиплексу подати код, який вказано на рис. 3.9б, то при подачі змінних A і B на входи адреси $A_0 A_1$ маємо таке:

- при $AB = 00$ значення $y = 0$;
- при $AB = 01$ $y = 1$;
- при $AB = 10$ $y = 1$;
- при $AB = 11$ $y = 0$.

	B	
A	0	1
0	0 D_0	1 D_1
1	1 D_2	0 D_3

а



б

Рисунок 3.10 – Карта Карно для мультиплексу:

а – карта Карно; б – схематичне зображення мультиплексу

Приклад 2. Задана логічна функція

$$y = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot C.$$

Карта Карно для цієї функції подана на рис. 3.11а. Для реалізації логічної функції використовуємо 8-входовий мультиплексор (рис. 3.11б). Однак ту ж функцію можна реалізувати за допомогою мультиплексору з меншим числом входів. Наприклад, вхідні сигнали B і C , як і раніше, будуть присутні на адресних входах мультиплексору, а вхідний сигнал A використаємо для формування сигналів на інформаційних входах. При цьому кожній комбінації сигналів BC на карті Карно буде відповідати не клітинка, а стовець. Зіставимо кожному стовпцю карти один із входів 4-входового мультиплексору, що еквівалентно переходу до 4-клітинної карти Карно (див. рис. 3.12а). У новій 4-клітинній карті Карно значення функції y виражені через вхідний сигнал A за допомогою первісної (8-клітинної) карти, де значення y збігаються з A у клітинках першого стовпця ($y = A$), значення y дорівнюють інверсним значенням y у клітках другого стовпця ($y = \bar{A}$), і т.д. У загальному випадку можливі чотири варіанти визначення функції y за новою картою: $y=A$; $y = \bar{A}$; $y = 0$ і $y = 1$. Два останніх варіанти будуть мати місце, коли у вихідній карті (рис. 3.5а) в обох клітинках стовпця $y = 1$ або $y = 0$.

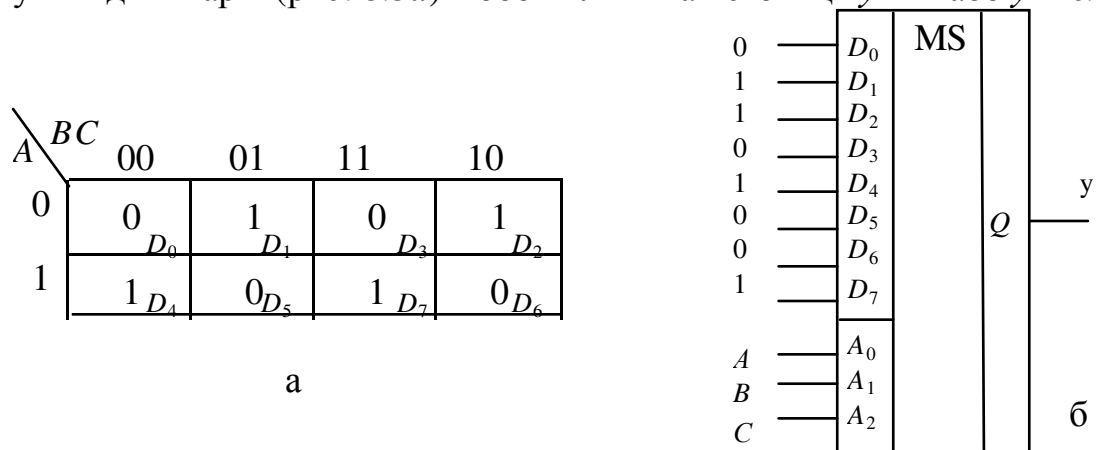


Рисунок 3.11 – Побудова карти Карно і схема підключення мультиплексору: а – карта Карно; б – схема подачі сигналів

Реалізація функції y за допомогою 4-входового мультиплексору показана на рис. 3.12б.

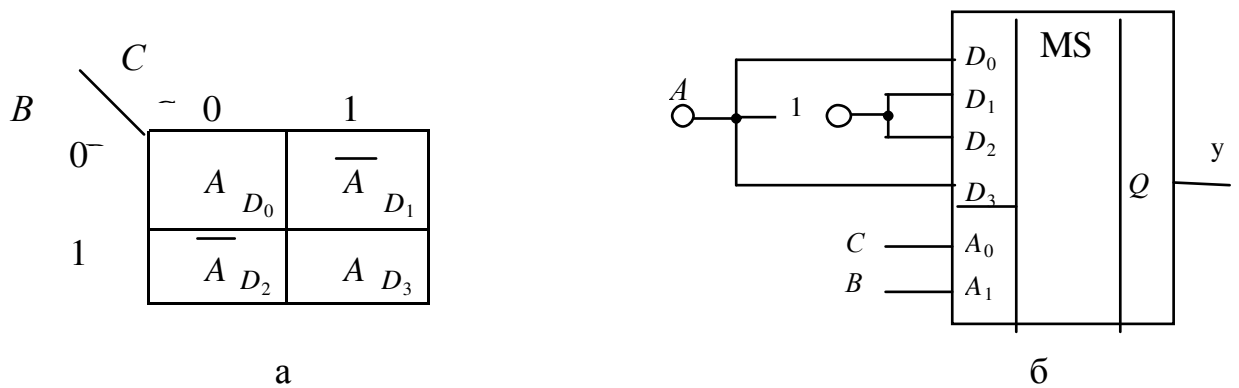


Рисунок 3.12 – Варіанти реалізації функції: а – карта Карно; б – схема подачі сигналів

Приклад 3. Більш складним прикладом синтезу КП на основі мультиплексу є 4-входовий цифровий компаратор для сигналів $ABCD$. Нехай: при $y = 1$ у тих випадках, коли три і більше вхідних сигналів дорівнюють 1. Відповідно до цього словесного опису карта Карно подана на рис. 3.13а. Нехай вхідні сигнали A , C і D будуть надходити на адресні входи, а сигнали B використовуємо для формування сигналів на інформаційних входах. У загальному випадку доцільно подавати на адресні (селекторні) входи мультиплексу змінні, котрі входять у більше число додатків структурної формули в ЗДНФ.

Для прикладу, що розглядаємо, в кожному стовпці вихідної карти попарно об'єднаємо клітинки при $A = 0$ і $A = 1$, а значення вихідної функції у для кожної пари клітинок виразимо через значення змінної B . Цей прийом дозволяє перейти до 8-клітинної карти Карно (рис. 3.13б), за допомогою якої встановлюємо стани інформаційних входів: $D_0 = 0$ (код клітинки карти Карно $ACD = 000$); $D_1 = 0$ ($ACD = 001$); $D_2 = 0$ ($ACD = 010$); $D_3 = B$ ($ACD = 011$) і т.п. Реалізація такого компаратора подана на рис. 3.13в.

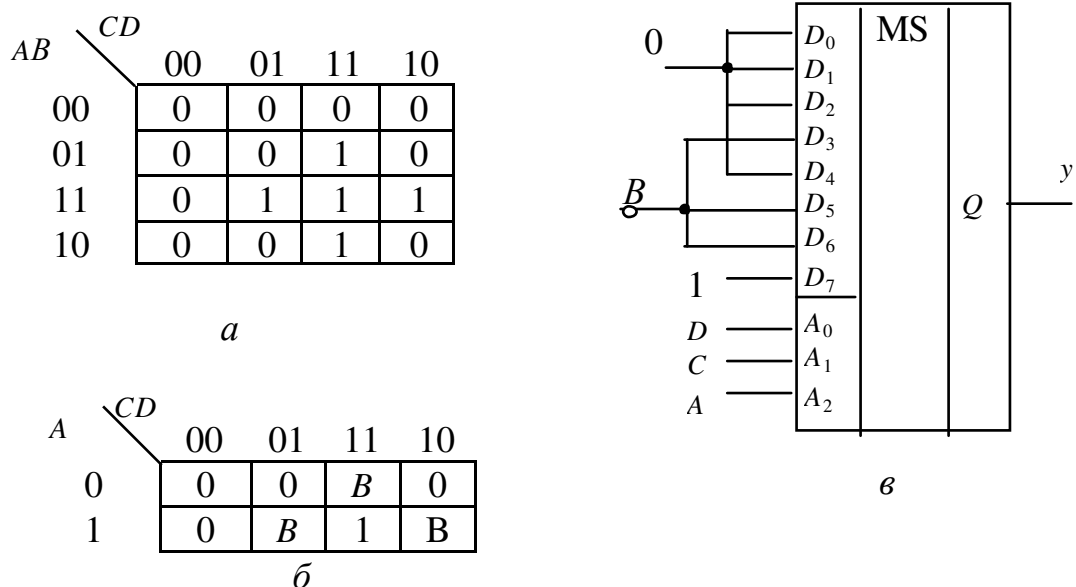


Рисунок 3.13 – Формування карт Карно, схеми подачі сигналів:

а – початкова карта, *б* – 8-клітинна карта, *в* – схема подачі сигналів

Якщо для формування сигналів на інформаційних входах мультиплексу використати ЛЕ, можна скоротити число його входів. Нехай, наприклад, потрібно синтезувати логічну функцію чотирьох змінних $f(x_1, x_2, x_3, x_4)$ з використанням 4-входового мультиплексу. Якщо адресними змінними вибрати x_1 та x_2 , то на інформаційні входи мультиплексу повинні надходити змінні x_3 та x_4 , які обумовлено, як вказано на рис. 3.14а, областями діаграм Вейча [2]. У середині кожної чергової області діаграми Вейча проводиться процедура мінімізації. Якщо логічна функція чотирьох змінних задана картою (рис. 3.14б), то на інформаційні входи $D_0 \dots D_3$ мультиплексу повинні надходити сигнали 1,

0, $(x_3 \cdot \bar{x}_4)$ та \bar{x}_4 відповідно. Реалізація заданої функції приведена на рис. 3.14в.

У мультиплексорів, які випускаються промисловістю у вигляді інтегральних мікросхем [2...5], число інформаційних входів не перевищує шістнадцяти. Збільшити кількість входів можна або шляхом об'єднання декількох мультиплексорів у пірамідальну (деревоподібну) систему, або шляхом послідовного з'єднання входів, управляючих сигналів та зовнішніх логічних елементів.

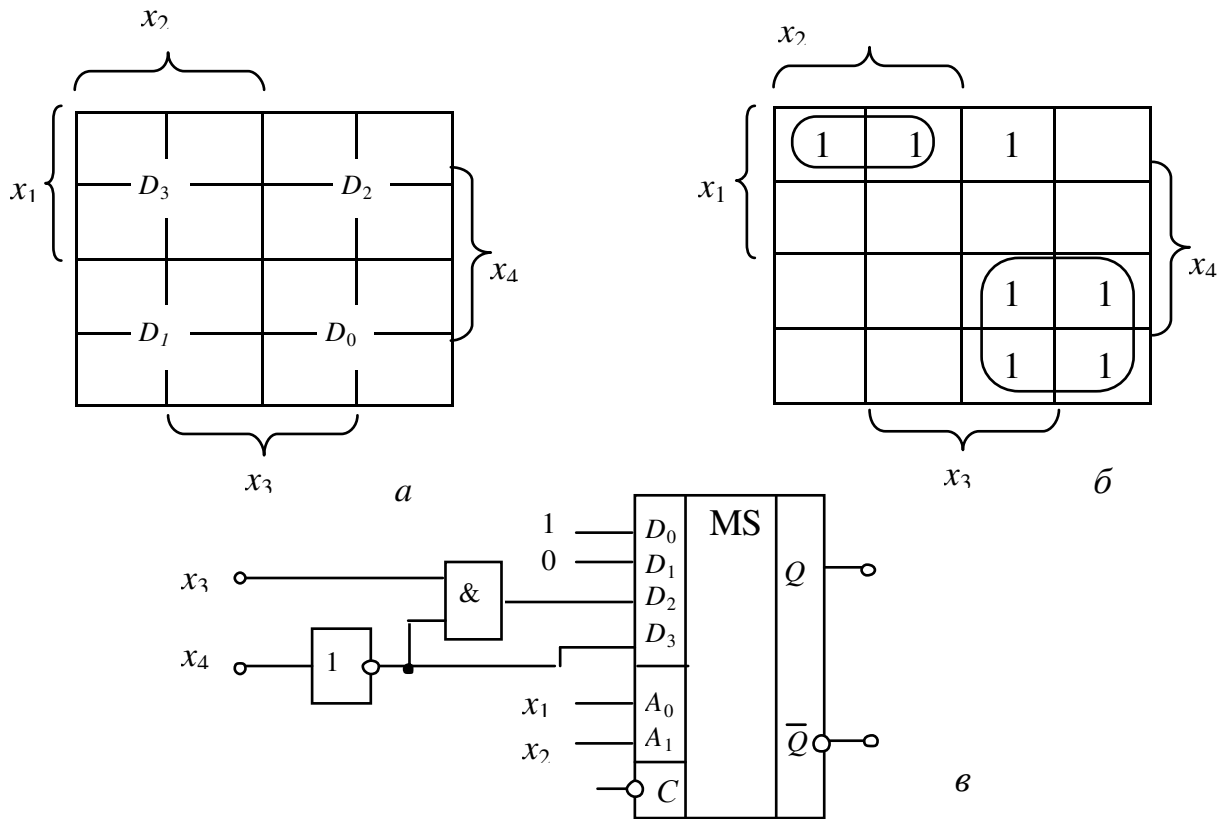


Рисунок 3.14 – Синтез КП на мультиплексорі з використанням ЛЕ:
а – діаграма Вейча; *б* – карта Карно; *в* – схемна реалізація

При пірамідальній побудові принципової схеми мультиплексору кожна наступна ступінь починається з першої. Вона має меншу кількість входів, чим попередня. Молодші розряди коду підключено до адресних входів першої ступені. Ступеням більш високого рангу відповідають старші розряди коду. На рис. 3.15а приведена схема мультиплексору з 32-ма інформаційними входами, яку побудовано з використанням чотирьох 8-входових мультиплексорів та одного 4-входового. Велика кількість ІС є недоліком пірамідальної побудови схеми мультиплексору. Зменшити кількість ІС можливо шляхом побудови мультиплексорів із використанням входів управляючих сигналів (рис. 3.15б).

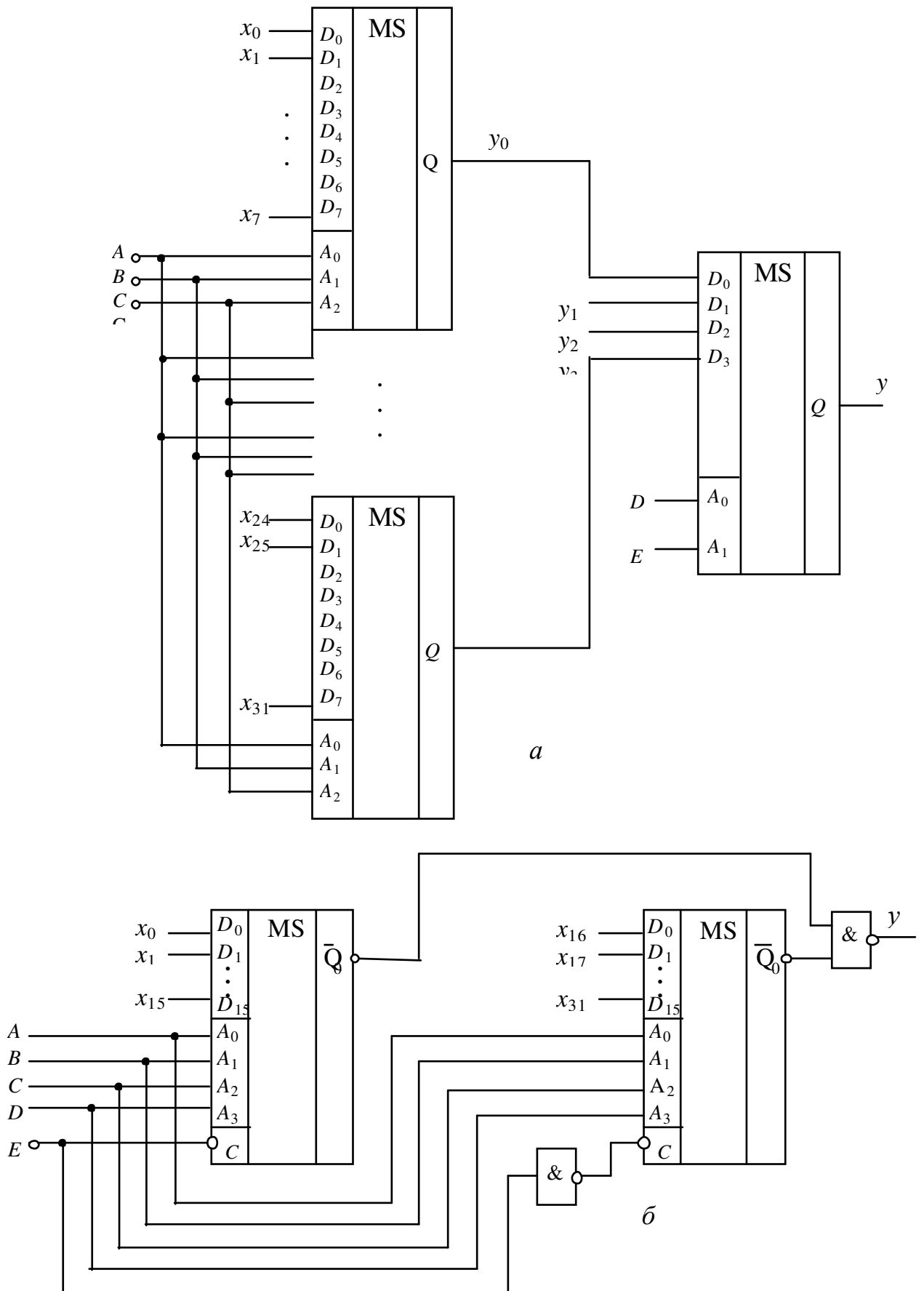


Рисунок 3.15 – Схема розширення числа входів мультиплексуру:
а – пірамідальна побудова; *б* – з використанням управляючих сигналів

На відміну від мультиплексорів, які виконано по ТТЛ-технології, мультиплексори на основі КМОП-технології по даним роботи [3] будуються з використанням дешифраторів та двоспрямованих ключів (рис. 3.16). Мікросхема містить у собі такі пристрої:

- перетворювач логічних рівнів, який забезпечує узгодження потенціалів цифрових вхідних сигналів і внутрішніх потенціалів мікросхеми;
- дешифратор, що здійснює перетворення вхідного коду на адресних входах A, B, C у сигнал на одному з його виходів;
- двоспрямовані ключі, які управляються вихідними сигналами дешифратора.

Сигнал від входу X до виходу Y проходить без додаткових перетворень у проміжних елементах ІС. Це дає можливість здійснювати комутацію як цифрових (імпульсних), так і аналогових сигналів. При цьому неспотворена передача аналогових сигналів забезпечується відповідним вибором величини живлячої напруги та схемою підключення двоспрямованих ключів.

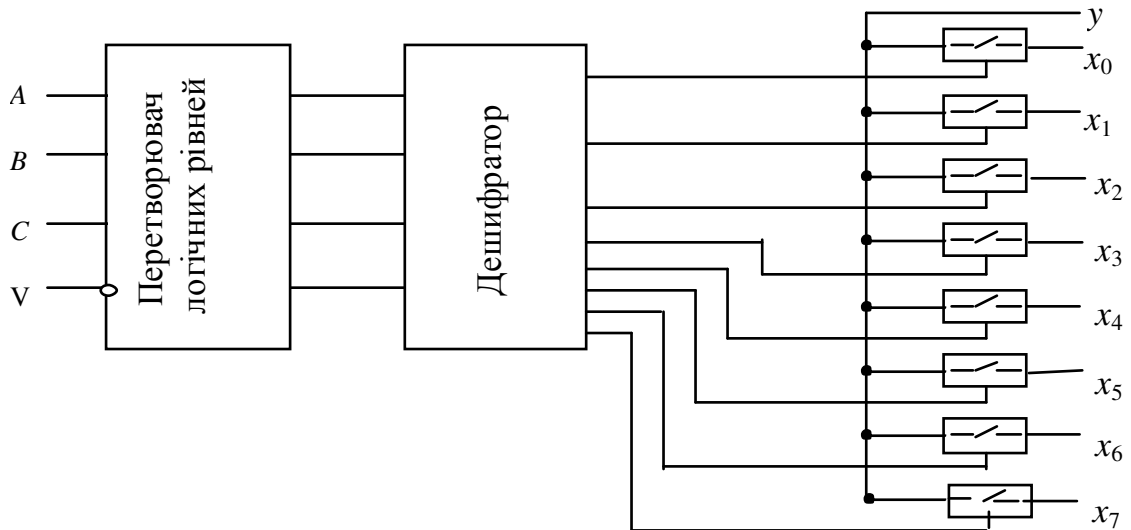


Рисунок 3.16 – Структура мультиплексору КМОП-технології

Демультимплексори виконують функції, які протилежні функціям мультиплексору. Вхідний сигнал x надходить на один з N виходів залежно від значення коду адреси $A_1...A_M$ (рис. 3.17). При цьому демультимплексори мають $N = 2^M$ виходів (прямих чи інверсних). Приклад таблиці істинності для $N = 4$ наведено в табл. 3.10.

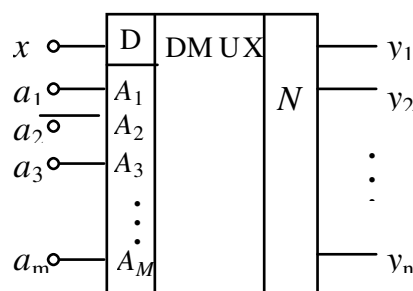


Рисунок 3.17 – Схема зображення демультимплексору

Таблиця 3.10 – Таблиця істинності

a_2	a_1	y_1	y_2	y_3	y_4
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

Якщо на інформаційному вході D присутня логічна одиниця, то на обраному згідно з адресою виході також формується логічна одиниця (на інших виходах логічний нуль). Демультіплексор у даному випадку є дешифратором. Якщо на вході D встановлено логічний нуль, то мікросхема виконує функцію демультіплексору. Такі ІС називаються дешифраторами-демультіплексорами. Звичайно дешифратори-демультіплексори мають чотири, вісім або шістнадцять виходів.

Дешифратор-демультіплексор (рис. 3.18) має два входи дозволу ($\overline{E0}$ і $\overline{E1}$) та чотири адресних входи. Для демультіплексору на одному з входів дозволу повинен бути логічний нуль, а інший вхід використовується як інформаційний. Якщо логічний нуль присутній на обох входах $\overline{E0}$ і $\overline{E1}$, то ІС працює як дешифратор на чотири входи і шістнадцять виходів. При наявності логічної одиниці на кожному із входів $\overline{E0}$ і $\overline{E1}$ на усіх виходах встановлюється логічна одиниця.

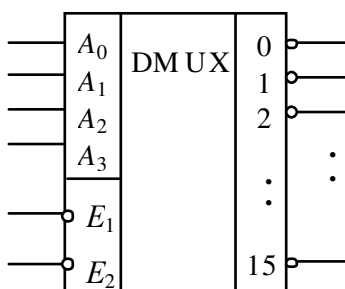


Рисунок 3.18 – Дешифратор-демультіплексор на шістнадцять виходів

Демультіплексори можуть поєднуватися у пірамідальну структуру з використанням входів дозволу (рис. 3.19а, б). За допомогою демультіплексорів-дешифраторів здійснюється реалізація логічних функцій, заданих таблицями істинності (табл. 3.11).

Шукана функція y може бути подано у вигляді

$$y = m_1 + m_2 + m_4 + m_5 + m_7,$$

де m_i – відповідний мінтерм чи кон'юнкція.

Перепишемо цей вираз в інверсній формі

$$\overline{y} = m_0 + m_3 + m_6.$$

Перетворимо отриману функцію з використанням інверсії

$$y = \overline{m_0 + m_3 + m_6} = \overline{m_0} \cdot \overline{m_3} \cdot \overline{m_6}.$$

Таблиця 3.11 – Таблиця істинності

x_3	x_2	x_1	y_0
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

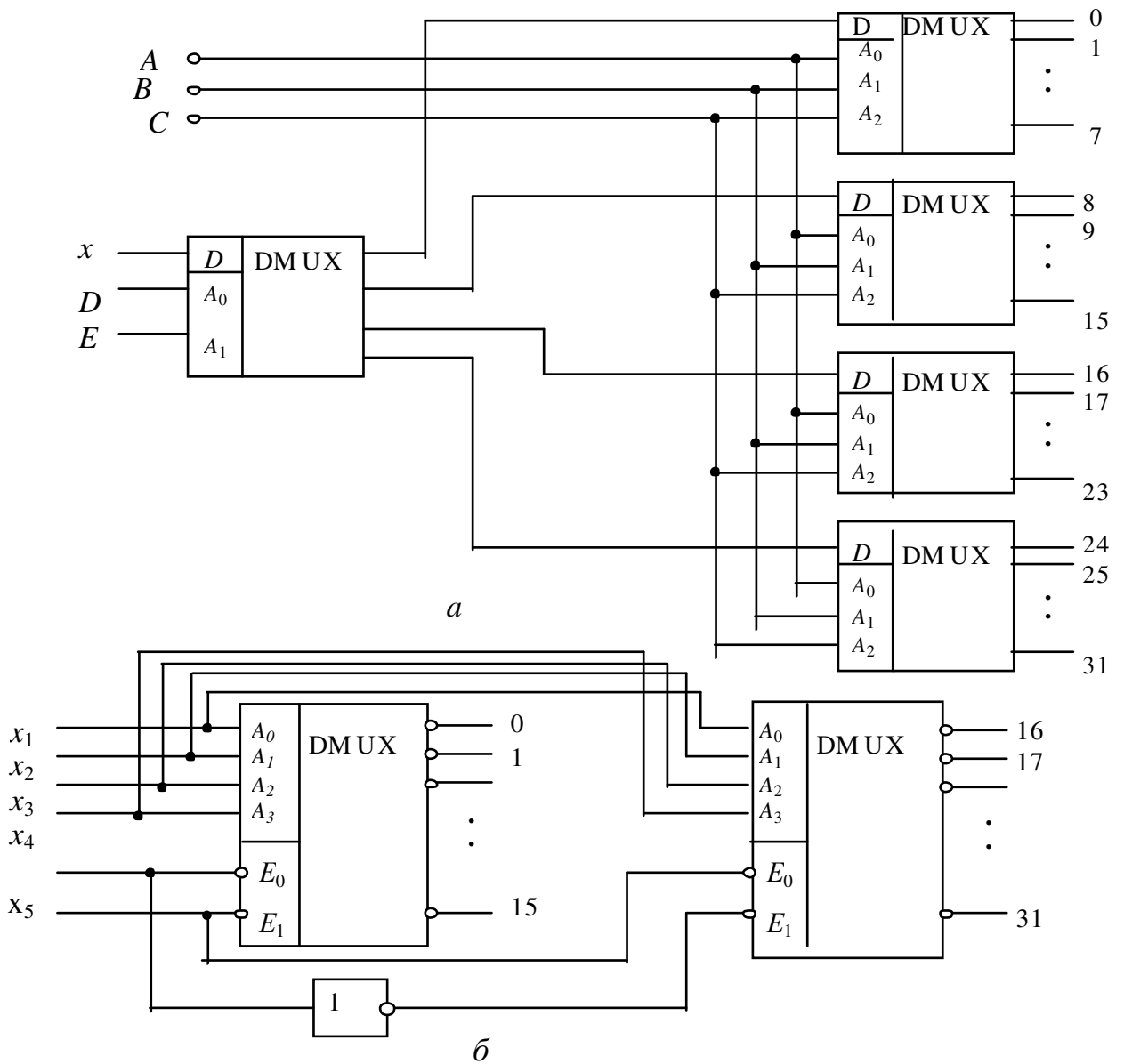


Рисунок 3.19 – Схеми розширення числа виходів:
a – демультиплексори; *б* – дешифраторів-демультиплексорів

Схемні реалізації двох останніх функцій наведено на рис. 3.20а, б відповідно. Найбільш ефективним є використання демультимплексорів у комбінаційних пристроях, які мають значну кількість вихідних сигналів:

- розподільники імпульсних послідовностей [4];
- пристрої формування сигналів [7];
- формування декількох логічних функцій тих самих змінних [3] та ін.

При спільному використанні мультиплексору та демультимплексору можна побудувати пристрої, у яких входи-виходи підключаються по заданим адресам [3, 7].

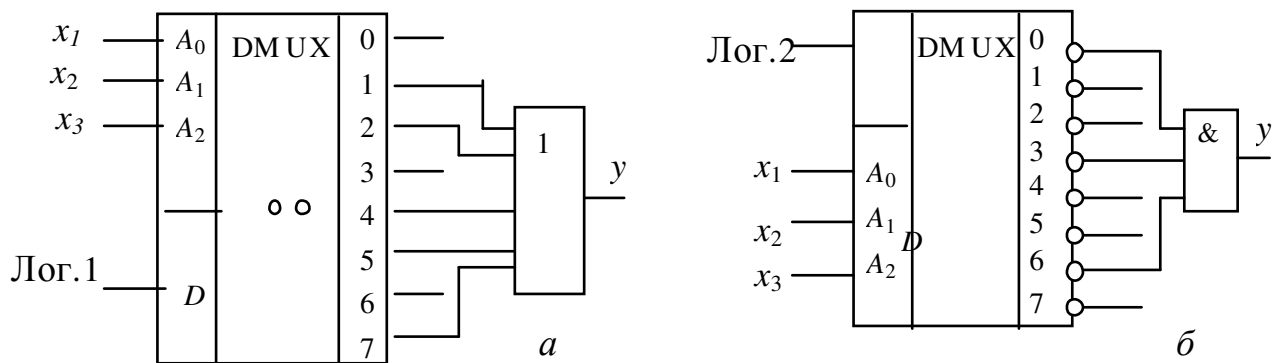


Рисунок 3.20 – Схемні реалізації логічної функції:
 а – на демультимплексорі; б – на дешифраторі

При реалізації комбінаційних пристроїв на мультиплексорах та демультимплексорах, які представляють собою ІС середнього ступеня інтеграції, скорочується необхідна кількість корпусів ІС і кількість зв'язків (сполучень) між ними. В наслідок цього підвищується надійність роботи комбінаційних пристроїв.

3.5 Тригери. Методи синтезу схем тригерів

Основною властивістю тригерів, які являють собою найпростіші ПП, є збереження одного з двох стійких станів після припинення дії вхідних сигналів. Тригер має два виходи: прямий Q і інверсний \bar{Q} . Якщо напруга на виході Q відповідає рівню логічного 0 ($Q = 0$), то тригер знаходиться в стані логічного 0; при $Q = 1$ тригер знаходиться в стані логічної 1. Розглянемо властивості RS , T , D , DV та JK -тригерів, які широко використовуються в системах керування електричним транспортом при побудові ключів, пристроїв для переривань та інше.

Асинхронні RS-тригери. Асинхронний RS -тригер, умовне позначення якого показано на рис. 3.21а, функціонує таким чином:

- якщо сигнали на його входах $R^n = S^n = 0$, то тригер не змінює свого стану ($Q^n = Q^{n-1}$);

- якщо $R^n = 0$, а $S^n = 1$ (назва входу S походить від англійського слова "SET" - установка), то тригер переключається в стан 1 $Q^n = 1$;
- якщо $R = 1$ (R відповідає слову "RESET" - скидання), $S^n = 0$, тригер переключається в стан 0 $Q^n = 0$;
- одночасна поява 1 на обох входах ($R^n = S^n = 1$) заборонена.

Правила роботи можна подати графом (рис. 3.21б), але слід зазначити, що граф не містить заборонених комбінацій вхідних сигналів. Маючи формальний опис тригера, можна синтезувати його схему так само, як схему комбінаційного пристрою [2]. Для цього, використовуючи, наприклад, елементи АБО-НІ, заповнюють таблицю істинності для Q^n , складають карту Карно (рис. 3.21в), з якої одержують структурну формулу для Q^n

$$Q^n = \overline{R^n} \cdot (S^n + Q^{n-1}) = \overline{R^n} \cdot (S^n + Q^{n-1}) = R^n + \overline{(S^n + Q^{n-1})}$$

Схема тригера, який функціонує за цією структурною формулою, подана на рис. 3.15г. Там враховано, що Q^{n-1} і Q^n є сигналами в одній точці схеми в різні моменти часу (штрихова лінія на рис. 3.21г). При наявності на обох входах пристрою однакових перемикаючих сигналів R^n і S^n стан виходів з рівною ймовірністю може стати як одиничним, так і нульовим. Оскільки ця невизначеність неприйнятна для тригера, то така комбінація є забороненою.

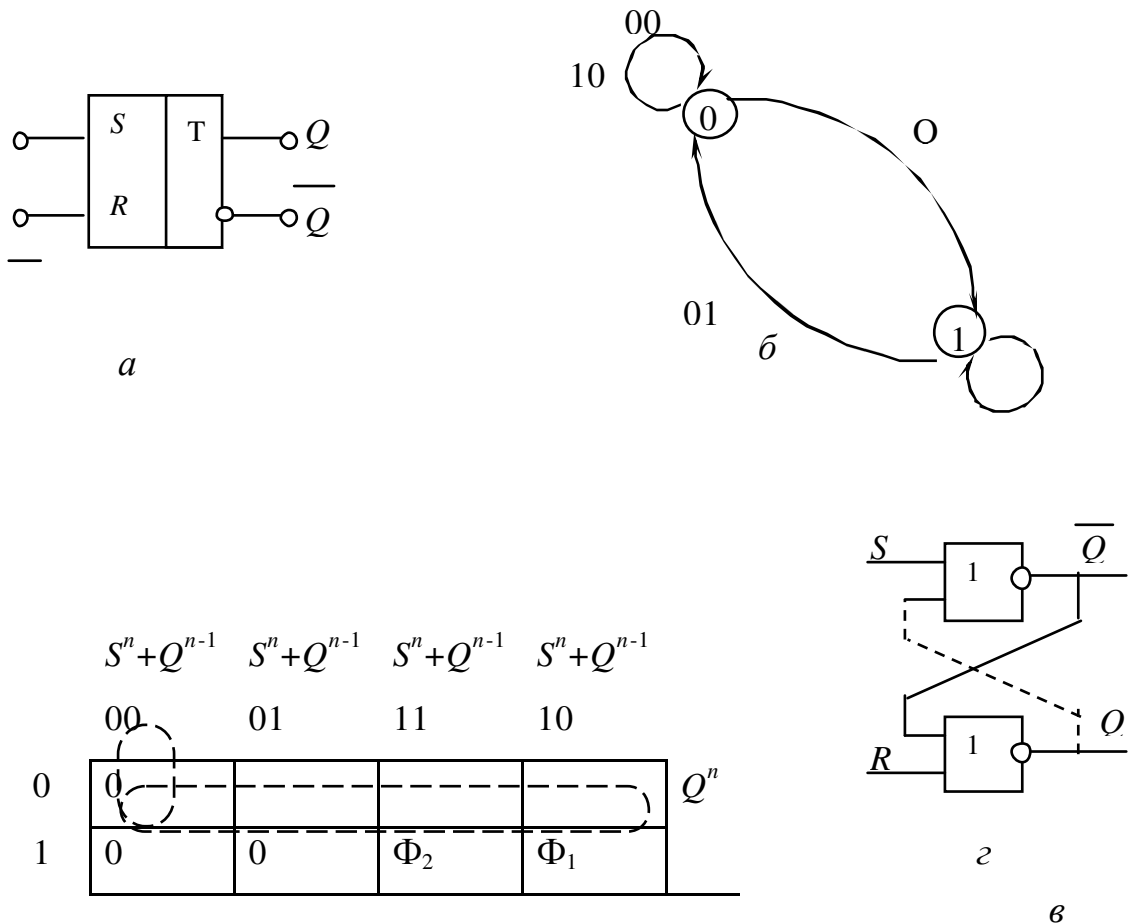


Рисунок 3.21 – RS-тригер:

а – умовне зображення; б – граф роботи; в – карта Карно; г – схема

Використовуючи елементи І-НІ, одержимо схему, яку зображено на рис. 3.22а. Виключивши зайві елементи, перекреслені пунктирними лініями, прийдемо до RS -тригера (рис. 3.22б) з тією ж таблицею перемикання. Усунувши із схеми входні інвертори, одержимо RS -тригер з нульовими перемикаючими сигналами (рис. 3.23в). Переходи цього RS -тригера показані в табл. 3.12.

Таблиця 3.12 – Таблиця істинності RS -тригера

R^n	S^n	Q^n
0	0	Q^{n-1}
0	1	1
1	0	0
1	1	–

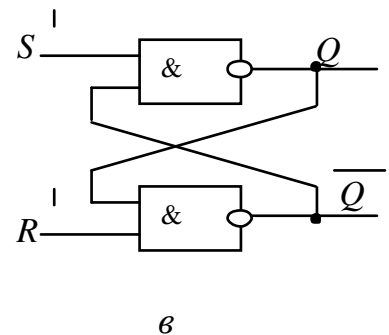
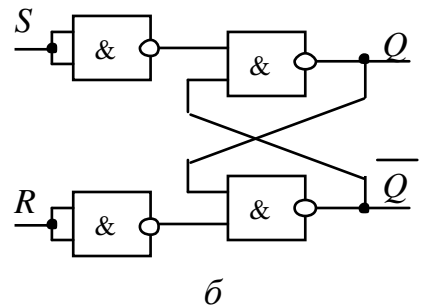
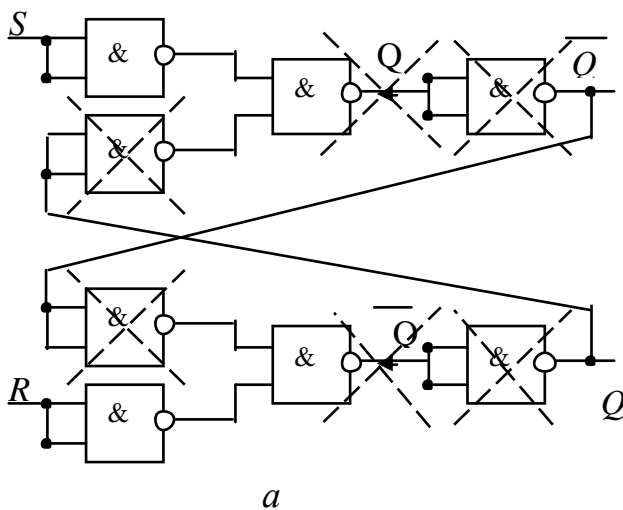


Рисунок 3.22 – Синтез RS -тригера на ІС І-НІ:

а – схема складена по формулі; *б* – схема з одиничними сигналами перемикання; *в* – схема з нульовими сигналами перемикання

Синхронні RS -тригери. Синхронний RS -тригер змінює свій стан, при зміні управляючих сигналів лише при наявності сигналу потрібного рівня на вході синхронізації C . При $C = 0$ стан тригера не змінюється. Функціональне позначення RS -тригера та граф його роботи показані на рис. 3.23. За принципом роботи цей тригер відповідає асинхронному RS -тригеру з одиничними перемикаючими сигналами при $C^n = 1$. З огляду на це, можна подати структуру синхронного тригера у вигляді послідовно з'єднаних

пристроїв управління (ПК) та елемента пам'яті – RS -тригера (рис. 3.24а).

Таблиці істинності для входів елемента пам'яті R' та S' (табл. 3.13, табл. 3.14 та табл. 3.15) складено з урахуванням словесного опису роботи синхронного тригера. Оскільки таблиці містять значне число одиниць для функцій R' і S' , то карти Карно можна скласти для зворотних функцій. Одержана завдяки цьому схема наведена на рис. 3.24б.

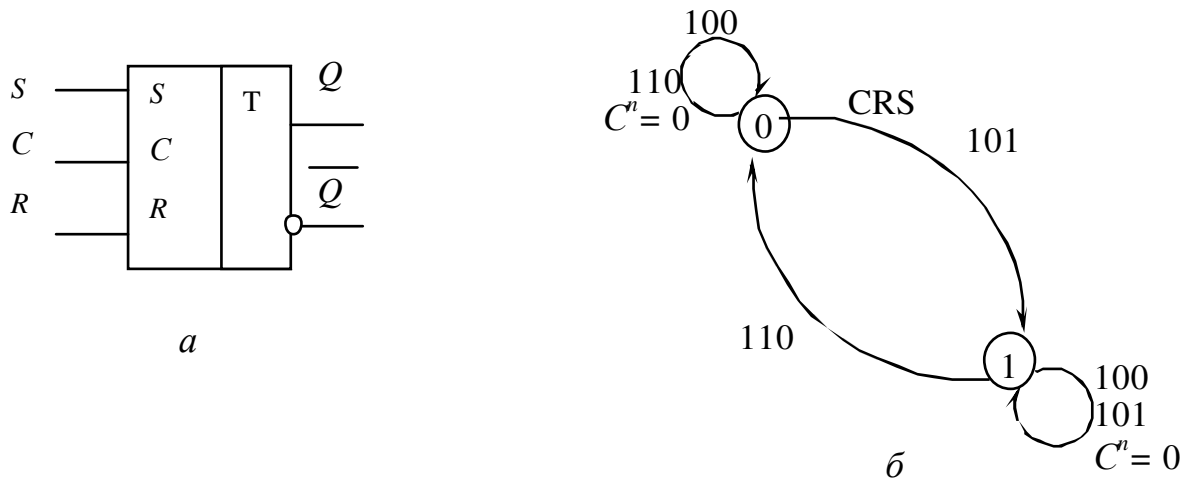


Рисунок 3.23 – Синхронний RS - тригер:
а – умовне позначення; б – граф роботи

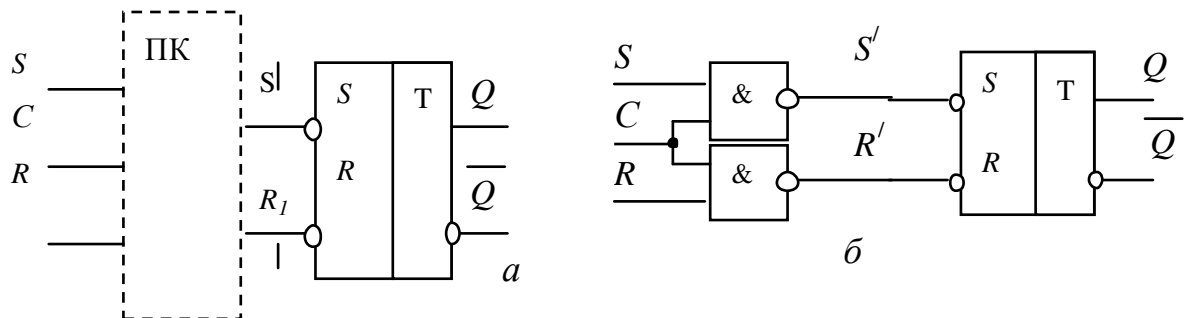


Рисунок 3.24 – Побудова синхронного тригера:
а – блок-схема; б – структурна схема

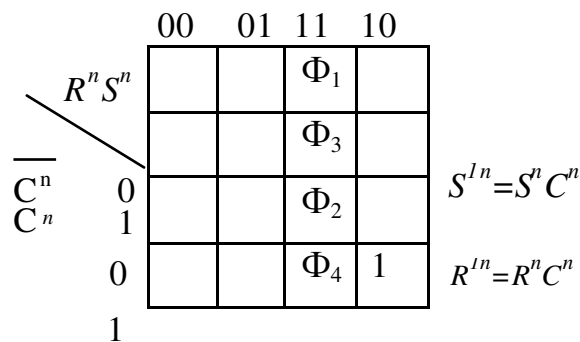


Рисунок 3.25 – Карта Карно синхронного RS - тригера

T -тригери. Рахунковий тригер (рис 3.26а) має один інформаційний вхід

(від слова "TOGGLE" – перемикач). T -тригер переключастся в протилежний стан із приходом кожного перемикаючого сигналу на вхід T . Таблиця переключень такого пристрою (табл. 3.16) містить два рядки.

Таблиця 3.13 – Таблиця істинності

R^n	S^n	Q^{n-1}	Q^n
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	Φ_1
1	1	1	Φ_2

Таблиця 3.14 – Таблиця істинності

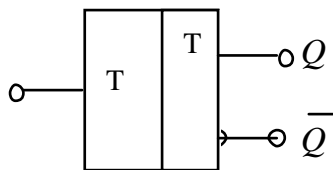
R'	S'	Q^n
1	1	Q^{n-1}
1	0	1
0	1	0
0	0	-

Таблиця 3.15 – Таблиця істинності

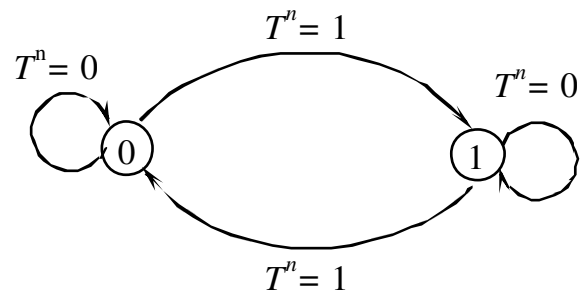
C^n	R^n	S^n	R^n	S^n
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	Φ_1	Φ_2
1	0	0	1	1
1	0	1	1	0
1	1	0	0	1
1	1	1	Φ_3	Φ_4

Таблиця 3.16 – Таблиця переключень T -тригеру

T^n	Q^n
0	$\frac{Q^{n-1}}{Q^{n-1}}$
1	$\frac{Q^{n-1}}{Q^{n-1}}$



а



б

Рисунок 3.26 – Одноступеневий T - тригер:
а – умовне позначення; б – граф роботи

Якщо розглянути граф переходів T -тригера на одному елементі пам'яті (рис. 3.27б), то очевидно, що він буде нестійким. При $T^n = 1$ буде відбуватися переключення тригера з одного стану в інший і назад (режим генерації). Забезпечення стійкості роботи T -тригера здійснюється одним із двох

способів: побудовою двоступеневих тригерів, або організацією динамічного управління. У двоступеневому T -тригері (рис. 3.27) один елемент пам'яті формує вихідні сигнали Q і \bar{Q} , а другий забезпечує стійкість роботи. Ці тригери часто називають тригерами типу MS від англійських слів “*MASTER*” – хазяїн та “*SLAVE*” – раб (у російській інтерпретації: система “ведучий-ведений”).

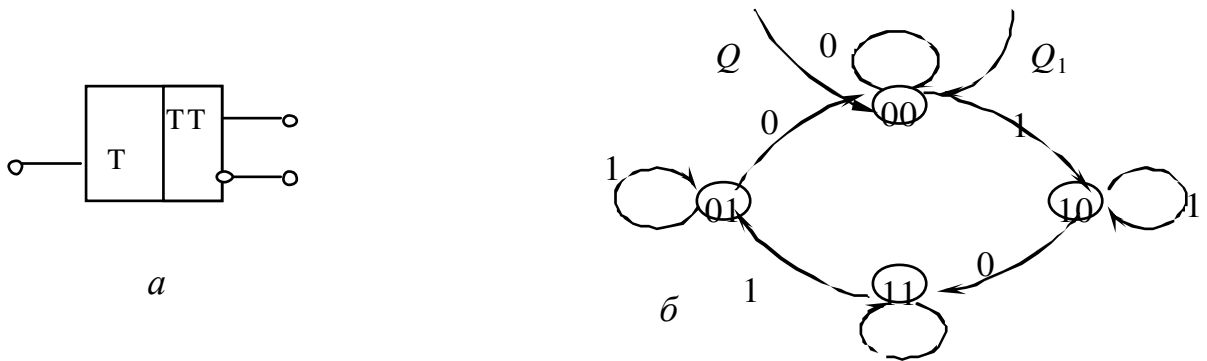


Рисунок 3.27 – Двоступеневий T -тригер:
a – умовне позначення; *б* – граф роботи

Проведемо синтез такої схеми T -тригера, вважаючи, що в його структуру входять два RS -тригери: основний Q та допоміжний Q_1 . Задача синтезу зводиться до визначення функцій управління елементами пам'яті (RS -тригерів). За допомогою графа переключень складається таблиця істинності (табл. 3.12) для вхідних сигналів RS -тригерів. При цьому там, де можливо, ставиться знак Φ , щоб повніше використовувати можливості процедури мінімізації. Потім за картами Карно (рис. 3.28), отриманими для інверсних функцій, визначаються структурні формули, що дозволяють побудувати схему двоступеневого T -тригера. Вона включає два RS -тригери та додаткові елементи, що відповідають отриманим структурним формулам.

Q^{n-1}	Q_1^{n-1}	00	01	11	10
\bar{T}^n	0			Φ_5	Φ_4
T^n	1				Φ_{10}

$$\bar{S}_1^n = \bar{T}^n \cdot Q_1^{n-1}$$

0	Φ_1	Φ_3		
1		Φ_8	1	

$$\bar{R}_1^n = \bar{T}^n \cdot Q_1^{n-1}$$

a

Q^{n-1}	Q_1^{n-1}	00	01	11	10
\bar{T}^n	0			Φ_6	1
T^n	1		Φ_3	Φ_{12}	

$$\bar{S}_1^n = \bar{T}^n \cdot Q^{n-1}$$

0	Φ_2	1		
1	Φ_1			Φ_{11}

$$\bar{R}_1^n = \bar{T}^n \cdot Q^{n-1}$$

б

Рисунок 3.28 – Карти Карно двоступеневого T -тригера

У тригерах з динамічним управлінням стійкість роботи забезпечується тим, що процес зміни вхідного сигналу T запам'ятовується на кожному вході основного RS -тригера. Граф такого T -тригера та варіант симетричної схеми наведено на рис. 3.29.

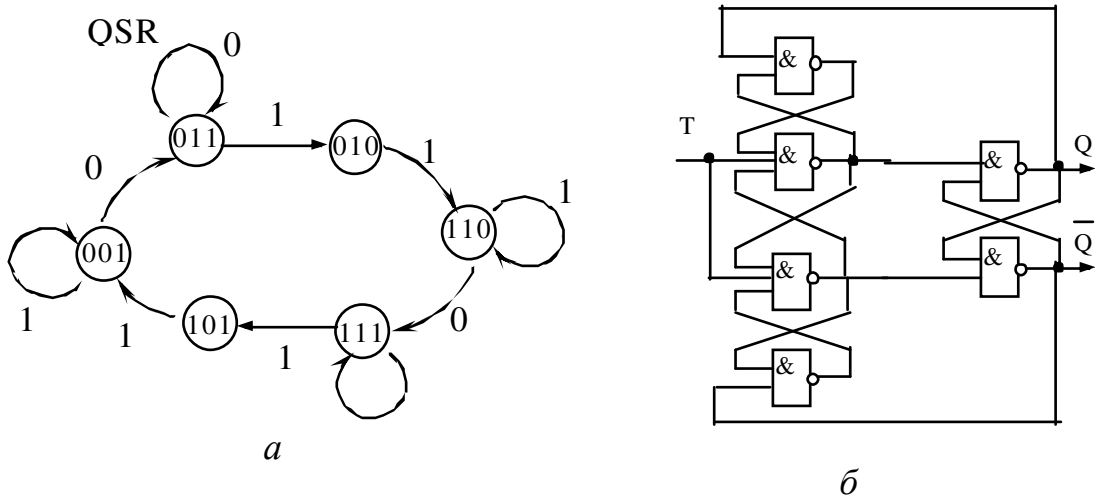


Рисунок 3.29 – Синтез двоступеневого T -тригера:
а – граф роботи; *б* – симетрична схема

D-тригери. Синхронний *D*-тригер (рис. 3.30) має два входи:

- даних D ;
- синхронізації C .

Такий тригер переходить у стан, що вказується сигналом на вході D , тільки з появою перемикаючого сигналу на вході C .

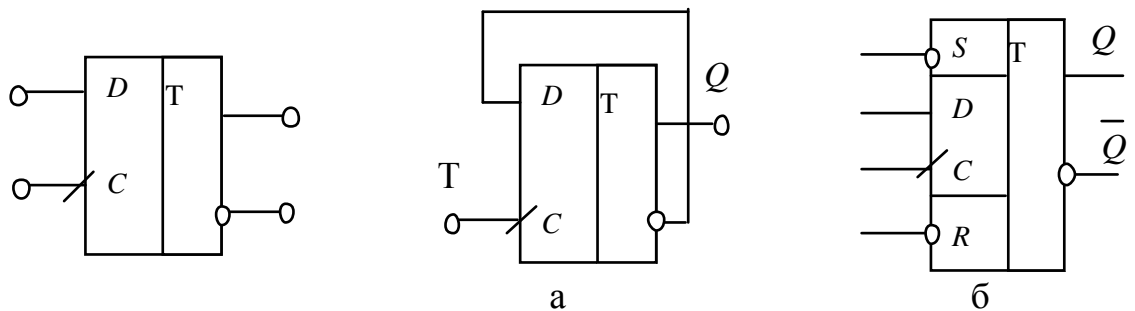


Рисунок 3.30 – Умовне зображення D -тригера

Рисунок 3.31 – Варіанти D -тригерів:
а – T -тригер; *б* – DRS -тригер

D -тригер здійснює затримку сигналу на вході D до появи сигналу на вході C , оскільки його вихідний сигнал Q^n повторює значення сигналу D на попередньому такті (D^{n-1}) та зберігає цей стан до наступного такту. Таким чином, D -тригер здійснює затримку на один такт сигналу. D -тригер часто називають тригером затримки (від англійського слова “*DELAY*” – затримка). Порівнюючи таблиці переключення D -тригера (див. табл. 3.17) та T -тригера

(див. табл. 3.16), можна зробити такий висновок: якщо стан $T = C$ та $D = Q$ (рис. 3.31а), то D -тригер буде функціонувати як T -тригер.

Якщо доповнити D -тригер установчими R та S – входами, то маємо DRS -тригер (рис. 3.31б). Стан його визначається сигналами на установчих входах та при $R^n = S^n = 1$ залежить від сигналів на входах D та C . Кодова комбінація $S^n = R^n = 0$, як і для RS -тригера, є забороненою.

Модифікацією D -тригерів є DV -тригери із додатковим входом V (від слова “VALVE” – клапан), який грає роль дозволяючого по відношенню до входу. При $V = 1$ тригер функціонує як D -тригер, а при $V = 0$ він зберігає свій стан при будь-яких значеннях сигналів на входах D та C . Тим самим наявність входу V дозволяє в необхідні моменти часу зберігати інформацію на виходах протягом заданого числа тактів, що розширює функціональні можливості пристрою.

Таблиця 3.17 – Таблиця переключень D -тригеру

Такт n		Такт $n+1$	
C	D^n	Q^n	Q^{n+1}
0	0	0	0
0	0	1	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

JK-тригери. Ці тригери (рис. 3.32а) є універсальними синхронними тригерами, в яких входи J та K відповідають входам S та R як у RS -тригері. Однак, на відміну від RS -тригеру набір $J^n = K^n = 1$ вважається припустимим й приводить до зміни стану тригера ($Q^n = Q^{n-1}$).

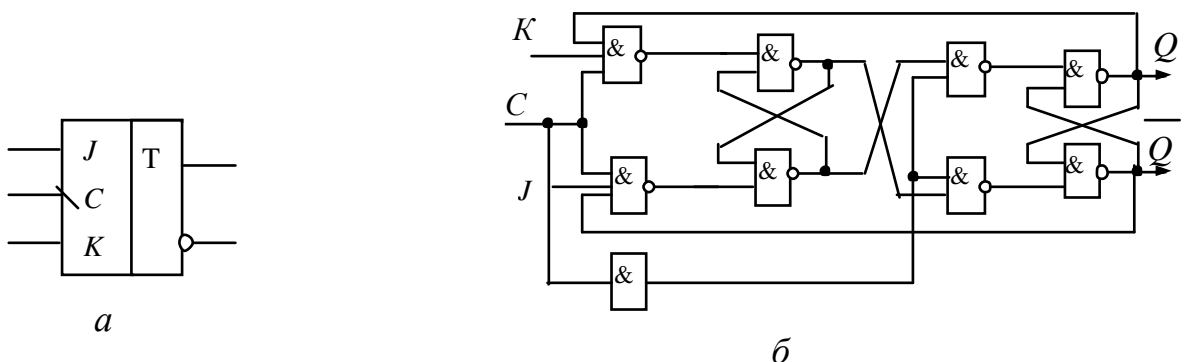


Рисунок 3.32 – JK -тригер:

а – умовне зображення; *б* – структурна схема

Оскільки JK -тригери виконуються синхронними, їхні вихідні рівні встановлюються тільки при надходженні на вхід перемикаючих сигналів. Стани JK -тригера наведені в табл. 3.18. Складаючи карту Карно для Q^{n+1}

(рис. 3.33), одержимо з її допомогою структурну формулу для вихідного сигналу тригера

$$Q^{n+1} = J^n \overline{Q^n} + \overline{K^n} Q^n.$$

Це рівняння називається характеристичним рівнянням тригера та цілком описує його роботу. Приклад схеми JK -тригера, який має структуру T -тригера та побудований за двоступеневою системою, наведено на рис. 3.32б.

Таблиця 3.18

J^n	K^n	Q^{n+1}
0	0	Q^n
0	1	0
1	0	1
1	1	$\overline{Q^n}$

	00	01	11	10	
$\overline{K}J$		1	1		Q^{n+1}
1	1	1			
Q^n					

Рисунок 3.33 – Карта Карно JK -тригера

Особливим з'єднанням виводів JK -тригера може бути переведений у режим роботи інших типів тригерів. Так, при $J = K = 1$ та при наявності відповідного сигналу на вході C він виконує функції T -тригера, а при $J = 0$ функції D -тригера. У деяких JK -тригерів для розширення їхніх функціональних можливостей введені асинхронні установчі входи (R, S), що діють незалежно від входу C .

Найпростіші тригерні схеми, які реалізовані у вигляді мікросхем [3, 4], дозволяють проектувати різноманітні цифрові пристрої систем керування електричним транспортом. Зокрема, на їхній основі будуються лічильники та регістри. Зростаюча складність цифрових схем робить практично неможливим інтуїтивний підхід при синтезі та змушує використовувати формалізовані узагальнені методи [1].

Розглянемо синтез тригерних схем як найпростіших цифрових ПП на прикладі D -тригера. Вихідними даними для синтезу є таблиця зовнішніх переходів (залежність вихідних сигналів від вхідних) та необхідні властивості спроектованої тригерної схеми. Метою синтезу є пошук рівняння виходу та системи рівнянь переходів. Таблиця зовнішніх переходів не цілком описує роботу тригера. Необхідно додатково визначити переходи тригера з одного внутрішнього стану в інший.

Оскільки на першому етапі проектування число внутрішніх змінних (Q) невідоме, то спочатку складають так звану первісну таблицю переходів та виходів, яка містить можливе число внутрішніх стійких станів. Кількість стовпців первісної таблиці обов'язково дорівнює числу різних станів входів тригерів. Для D -тригера, який має два входи C та D , потрібно чотири стовпця. Число рядків первісної таблиці можна визначити, якщо кожному

можливому стану входів та виходів тригера зіставити один стійкий внутрішній стан. Для *D*-тригера, таблиця буде містити 8 рядків.

Кожен стійкий внутрішній стан нумерують. Значення виходу *Z* записують в окремому стовпці табл. 3.19. Таким чином, кожен стовець табл. 3.19 визначає стан вхідних змінних, а кожен рядок – стан внутрішніх змінних. Повний стійкий стан тригера буде умовно визначатися перетинанням рядка і стовпця таблиці. Цей стан записується цифрою в круглих дужках. У кожному рядку первісної таблиці вказується лише один стійкий стан.

Таблиця 3.19 – Таблиця переключень

Внутрішній стан	Стан входів				Стан виходу
	00	01	11	10	
1	(1)	2	<i>x</i>	4	0
2	1	(2)	3	<i>x</i>	0
3	<i>x</i>	6	(3)	4	0
4	1	<i>x</i>	3	(4)	0
5	(5)	6	<i>x</i>	8	1
6	5	(6)	7	<i>x</i>	1
7	<i>x</i>	6	(7)	8	1
8	1	<i>x</i>	7	(8)	1

Наступним кроком заповнення первісної таблиці є запис переходів тригера з одного стійкого стану в інший на підставі таблиці зовнішніх переходів та необхідних властивостей тригера. Помітимо, що проєктований *D*-тригер переключається за спадом логічної 1 на синхронізуючому вході *C*.

Перехід з одного стійкого стану в інший можливий тільки при зміні значень вхідних змінних, яка виражається зміною стовпця в таблиці переходів. Для того щоб в останній відобразити перехід тригера з одного стійкого стану в інший, необхідно на перетинанні рядка, який визначає попередній стан тригера, зі стовпцем таблиці, який визначає нове значення вхідних змінних. Потім записати номер того внутрішнього стійкого стану, у який повинен перейти тригер відповідно до вихідних даних. Цей номер не беруть у дужки, тому що він відповідає нестійкому стану тригера. Тут можливі тільки горизонтальне й вертикальне переміщення. Горизонтальне переміщення визначається зміною значень вхідних змінних тригерної схеми. Вертикальне переміщення відбувається мимоволі внаслідок зміни значень внутрішніх змінних.

Через нестійкий стан тригер проходить у тому випадку, коли змінився стан вхідних сигналів, а виходи блоків пам'яті через наявні в них затримки ще не змінили свого стану. Після закінчення часу затримки пристрій мимоволі переходить у новий стійкий стан.

Розглянемо запис переходів. Припустимо, що $C = D = 1$, $Z = 0$. Ця комбінація відповідає повному стану (3) у табл. 3.19. Допустимо, що вхідна

змінна C змінила своє значення на логічний 0, тобто $CD = 01$. Оскільки перемикаючим сигналом C є зміна $1 \rightarrow 0$, то відповідно до таблиці зовнішніх переходів D -триггер повинен змінити свій вихідний стан. Новим стійким станом повинен бути стан (6), який відповідає $C = 0, D = 1, Z = 1$. Щоб позначити цей перехід у табл. 3.19, необхідно на перетинанні третього рядка й стовпця $CD = 01$ записати цифру 6 без дужок. Якщо ж перехід з внутрішнього стану (3) здійснюється заміною змінної D , тобто $CD = 10$, то новим стійким станом повинен бути стан (4), оскільки на вході C стан не змінився. Так на етапі заповнення первісної таблиці переходів враховують асинхронне поводження проєктованого синхронного тригера, тобто його тип [1]. Нарешті, перехід від стійкого стану (3), при одночасній зміні змінних C та D , повинен бути забороненим, оскільки в цьому випадку фізично неможливо забезпечити однозначність операцій.

Таким чином, залежно від порядку зміни входних сигналів тригер може виявитися в стані 1 чи 0. У таблиці переходів заборонені переходи будемо позначати символом (x). Аналогічно заповнюють усю табл. 3.19, переходять до табл. 3.20. Очевидно, що при складанні таблиць усі основні властивості проєктованого пристрою, які сформульовані словесно, виражаються у вигляді первісної таблиці переходів.

Таблиця 3.20 – Таблиця переключень

Внутрішній стан	Стан входів CD				Стан виходу
	00	01	11	10	
1,2,4	(1)	(2)	3	(4)	0
3		6	(3)	4	0
5,6,7	(5)	(6)	(7)	8	1
8	1		7	(8)	1

Складність логічної структури проєктованого тригера залежить від числа внутрішніх змінних, тому доцільно мінімізувати число внутрішніх станів, яке дорівнює числу рядків первісної таблиці переходів. Під мінімізацією числа внутрішніх станів ПП мають на увазі процес, метою якого є одержання ПП. У цьому випадку ПП має мінімальне число внутрішніх станів серед усіх, які реалізують задані умови його роботи. Процес сполучення рядків таблиці переходів називають об'єднанням та виконують його відповідно до правил [1]:

- 1) два рядки (чи більше) можуть бути об'єднані, якщо у відповідних стовпцях номери станів однакові чи на одному з рядків є заборонений стан;
- 2) результуючий стан треба взяти у дужки, якщо один із поєднаних номерів у дужках, а інший без них. Якщо ж деякі рядки містять цифри, а інші символ (x), то в об'єднаному рядку повинна стояти цифра.

Розглянемо перший та четвертий рядки табл. 3.19. Їхнє об'єднання відповідно до даних правил дає новий рядок вигляду: (1); (2); (4). Перехід від стану (4) до (1) у новому об'єднаному рядку здійснюється зміною тільки

вхідної змінної C без зміни внутрішнього стану.

У другому варіанті (табл. 3.21) на відміну від першого переходу з одного рядка в інший здійснюються при заміні змінної C разом зі зміною логічної змінної D . При проектуванні двоступеневих тригерів доцільно використовувати об'єднання рядків первісної таблиці, що приводить до табл. 3.20, а при побудові тригерів з динамічним управлінням – до табл. 3.21.

Таблиця 3.21 – Таблиця переключень

Внутрішній стан	Стан входів CD				Стан виходу
	00	01	11	10	
1,2	(1)	(2)	3	4	0
3,4	1	6	(3)	(4)	0
6,6	(5)	(6)	7	8	1
7,8	1	6	(7)	(8)	1

Слід зазначити, що хоча число внутрішніх станів ПП визначає число елементів пам'яті, скорочення числа внутрішніх станів ПП не завжди приводить до зменшення числа елементів пам'яті. Але навіть і в цьому випадку доцільно робити мінімізацію числа внутрішніх станів, щоб не збільшувати число не використовуваних станів та не ускладнювати структуру логічного перетворювача.

3.6 Регістри

Основна функція регістра полягає в збереженні багаторозрядного двійкового числа. У звичайних випадках регістри мають регулярну структуру й складаються з однакових частин. Кожна з таких частин призначена для збереження одного розряду двійкового числа. Розряд регістра містить елемент пам'яті, який виконано на тригері.

Регістр може працювати в режимах запису, збереження та зчитування чисел. У режимі запису в регістр вводиться двійкове число. У режимі збереження записане число залишається без зміни, а при зчитуванні - число передається з регістра в інші пристрої.

За способами організації запису й зчитування чисел розрізняють наступні типи регістрів:

- паралельні: запис та зчитування у всіх розрядів здійснюються одночасно;
- послідовні (регістри зрушення): запис та зчитування даних здійснюється розрядами послідовно;
- паралельно-послідовні: запис здійснюється паралельно, а зчитування послідовно;
- послідовно-паралельні: на відміну від попереднього типу запис робиться послідовно, а зчитування паралельно;

- реверсивні регістри зсуву: напрямком зсуву чисел у них може бути змінений залежно від значень управляючих сигналів.

За допомогою регістрів реалізують перетворення послідовного двійкового коду в паралельний та навпаки, одержують часову затримку цифрових сигналів і виконують арифметичні та логічні операції з багаторозрядними числами. Розглянемо основні принципи побудови перерахованих вище типів регістрів.

Структура паралельного регістру (рис. 3.34а) являє собою сукупність однотипних осередків (рис. 3.34б). Ці осередки містять тригер та схеми управління записом W_1 й зчитуванням W_2 . Кожен осередок має інформаційний вхід (X_i). Входи управління записом (C_1) та зчитуванням (C_2) є загальними для всіх осередків. *Правило роботи для тригера в j-му осередку формулюється наступним чином:*

- якщо $C_1^n = 1, C_2^n = 0$, то $Q_i^n = X_j^n; Y_j^n = 0$;
- при $C_1^n = C_2^n = 0, Q_j^n = Q_j^{n-1}; Y_j^n = 0$, тобто здійснюється режим збереження числа;
- у випадку, коли $C_1^n = 0, C_2^n = 1$, тригер ($Q_j^n = Q_j^{n-1}$) не переключається і вхідний сигнал проходить на вихід $Y_j^n = Q_j^n$;
- комбінація $C_1^n = C_2^n = 1$ вважається забороненою.

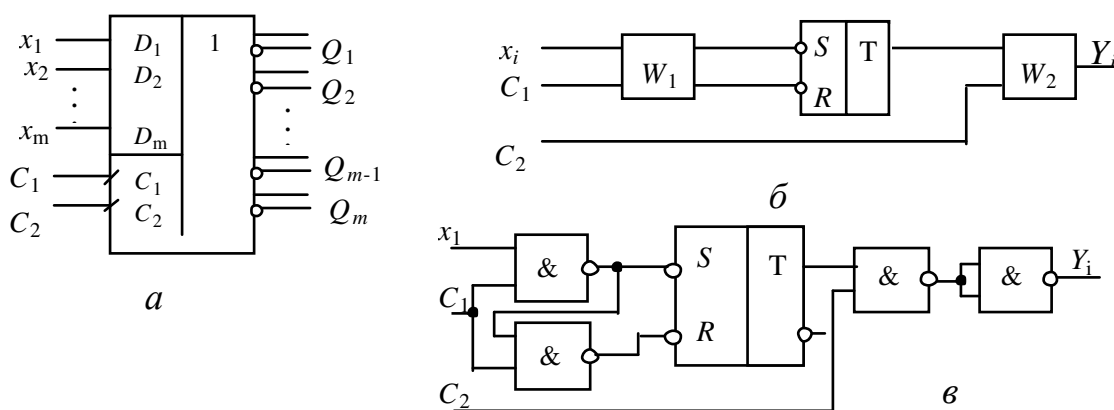


Рисунок 3.34 – Регістр паралельного типу:
а – умовне позначення; *б* – структура осередку; *в* – принципова схема осередку

Вибравши елементом пам'яті RS -тригер та склавши таблицю істинності для вихідної функції Y_j^n й функцій зрушення тригера (табл. 3.22), одержимо структурні формули роботи осередка

$$y_i^n = C_2^n \cdot Q_i^n = C_2^n \cdot Q_i^{n-1};$$

$$S = x_i \cdot C_1;$$

$$R = x_i \cdot C_1 + C_1 \cdot \overline{C_1} = C_1 \cdot (x_i + \overline{C_1}) = C_1 \cdot x_i \cdot C_1 = C_1 \cdot S.$$

Відповідно до цих формул схема одного розряду регістра має вигляд, який зображено на рис. 3.34в.

Таблиця 3.22 – Таблиця істинності для вихідної функції Y_j^n й функцій зрушення тригера

C_2	Q_i	y_i	C_1	x_i	R	S
0	0	0	0	0	1	1
0	1	0	0	1	1	1
1	0	0	1	0	0	1
1	1	1	1	1	1	0

Послідовний регістр зсуву (рис. 3.35а) має інформаційний та тактовий входи, які забезпечують запис числа та зсув його по регістрі. Структура регістра (рис. 3.35б) являє собою послідовне з'єднання однотипних осередків. *Правило роботи для j-го розряду можна записати у такому вигляді:*

- якщо $C^n = 0$, то $Q_{j-1}^n = Q_{j-1}^{n-1}$;
- при $C^n = 1$ $Q_j^n = Q_{j-1}^{n-1}$.

Такий запис збігається з правилом роботи D -тригера, на вхід якого надходить сигнал Q_{j-1} . Отже, для побудови регістра зсуву можна використати D -тригери або JK -тригери, які працюють у режимі D -тригера. Відповідна схема регістра дана на рис. 3.35в.

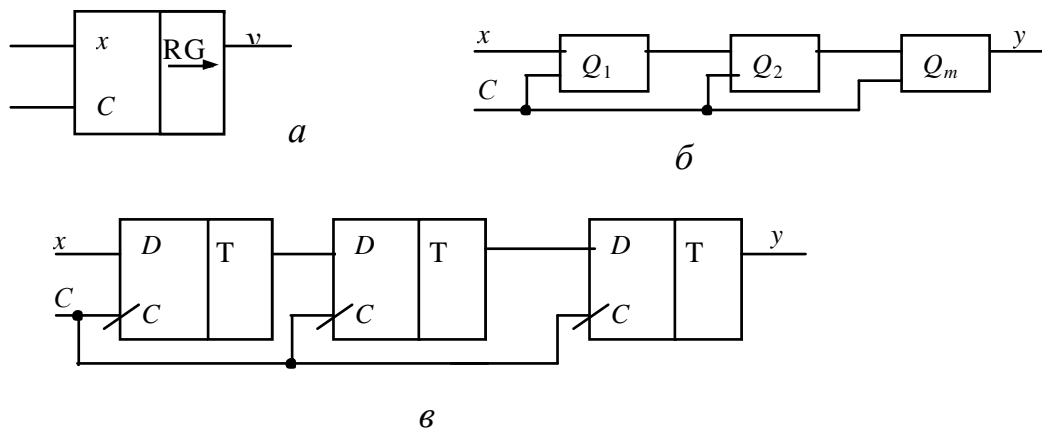


Рисунок 3.35 – Регістр послідовного зсуву:

а – умовне позначення; б – структурна схема; в – принципова схема

Правила роботи для паралельно-послідовного регістру такі:

- при $C_1=1$ та $C_2=1$ виконується запис інформації у всі розряди регістру;
- при $C_1=0$ та $C_2=1$ виконуються зсуви інформації.

Застосуємо у якості елемента пам'яті D -тригер. Входи R та S використано для паралельного запису інформації, входи D та C використано для організації послідовних зсувів чисел. Схема паралельно-послідовного регістру наведена на рис. 3.36.

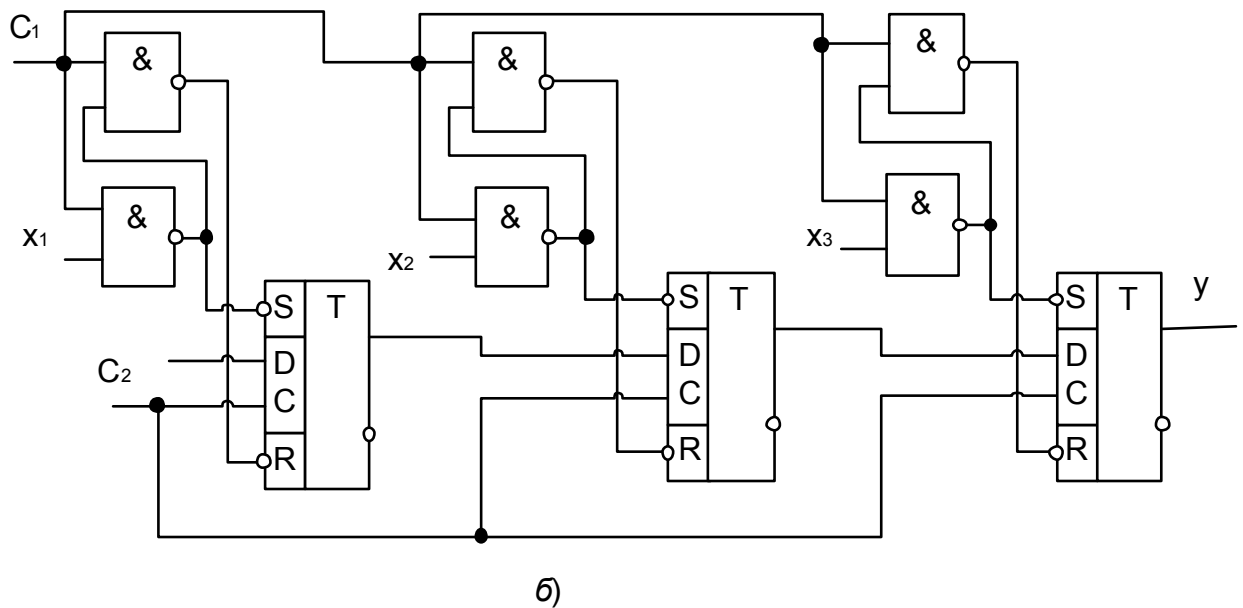
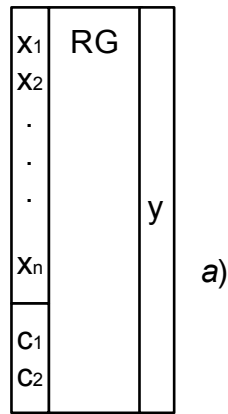


Рисунок 3.36 – Регістр паралельно-послідовний:
а – умовне позначення; б – принципова схема

Послідовно-паралельний регістр має вхід для послідовного вводу даних по управляючому сигналу C_1 та m -виходів, на яких з'являються двійкові числа при наявності управляючого сигналу C_2 . *Правила роботи послідовно-паралельного регістру такі:*

- при $C_1=1$ та $C_2=0$ виконуються зсуви даних;
- при $C_1=0$ та $C_2=1$ виконуються вивід інформації з усіх розрядів регістру.

Якщо послідовний регістр на D -тригерах забезпечити схемами управління зчитуванням інформації, то отримаємо принципову схему послідовно-паралельного регістру (рис. 3.37). Його схема управління зчитуванням інформації аналогічна як у паралельного регістру. Основна функція послідовно-паралельного регістру у системах керування електричним транспортом це перетворення послідовного коду у паралельний.

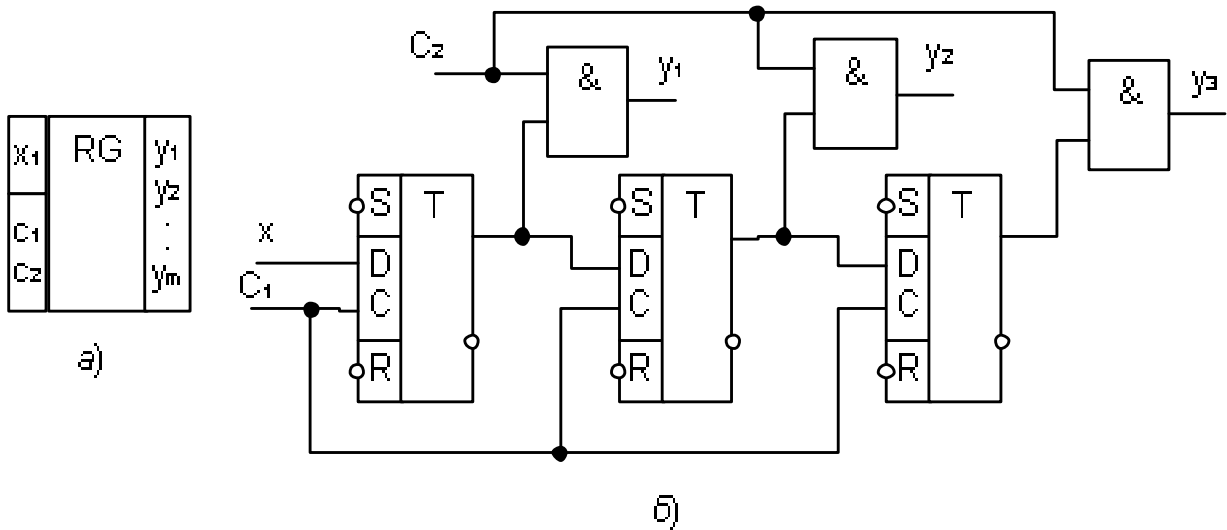


Рисунок 3.37 – Регістр послідовно-паралельний:
а – умовне позначення; *б* – принципова схема

Реверсивний послідовний регістр являє собою регістр зсуву, у якого напрям зсувів інформації визначається управляючим сигналом на вході C_1 . Зсуви інформації у регістрі виконуються по сигналу C_2 . Реверсивний регістр можна організувати на D -тригерах. При зсуві інформації праворуч схеми комутації забезпечують передачу стану попереднього тригера на вхід послідуєчого. При зсуві інформації ліворуч схеми комутації забезпечують передачу стану послідуєчого тригера на вхід попереднього. *Правила роботи реверсивного регістру такі:*

- при $C_1=1$ та $C_2=1$ виконуються зсуви даних праворуч;
- при $C_1=0$ та $C_2=1$ виконуються зсуви даних ліворуч;

На підставі словесного опису роботи реверсивного регістру будемо таблицю істинності (табл. 3.23).

Таблиця 3.23 – Таблиця істинності реверсивного регістру

C_1	Q_{i-1}	Q_{i+1}	D_i
0	0	0	0
0	0	1	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

Рівняння алгебри логіки схеми управління реверсивним регістром має такий вигляд

$$D_i = \overline{C_1} \overline{Q_{i-1}} Q_{i+1} + \overline{C_1} Q_{i+1} \overline{Q_{i-1}} + C_1 \overline{Q_{i+1}} \overline{Q_{i-1}} + C_1 Q_{i-1} Q_{i+1} = \overline{C_1} Q_{i+1} + C_1 Q_{i-1}.$$

Мінімізація рівняння виконано за допомогою методу Квайна. На підставі цього рівняння будемо принципову схему реверсивного регістру. Ця схема наведена на рис. 3.38.

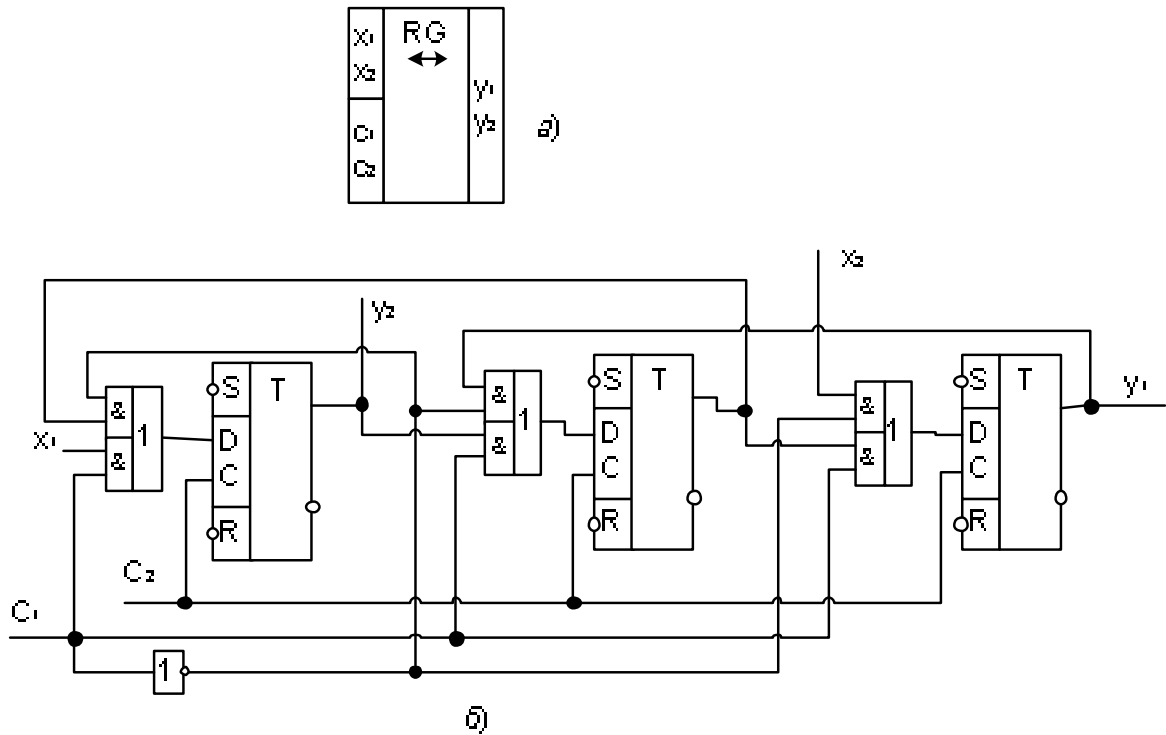


Рисунок 3.38 – Реверсивний регістр:
а – умовне позначення; б – принципова схема

Приклад 1. Спроекувати трьохрозрядний кільцевий регістр зсуву на D -тригерах. Складаємо таблицю станів та переходів тригерів (табл. 3.24).

Таблиця 3.24 – Таблиця станів та переходів тригерів

Q_n			Q_{n+1}		
3	2	1	3	2	1
ф	ф	ф	ф	ф	ф
0	0	1	0	1	0
0	1	0	1	0	0
ф	ф	ф	ф	ф	ф
1	0	0	0	0	1
ф	ф	ф	ф	ф	ф
ф	ф	ф	ф	ф	ф
ф	ф	ф	ф	ф	ф

На підставі таблиці переходів складаємо карти Карно для D -входів кожного тригера (рис. 3.39).

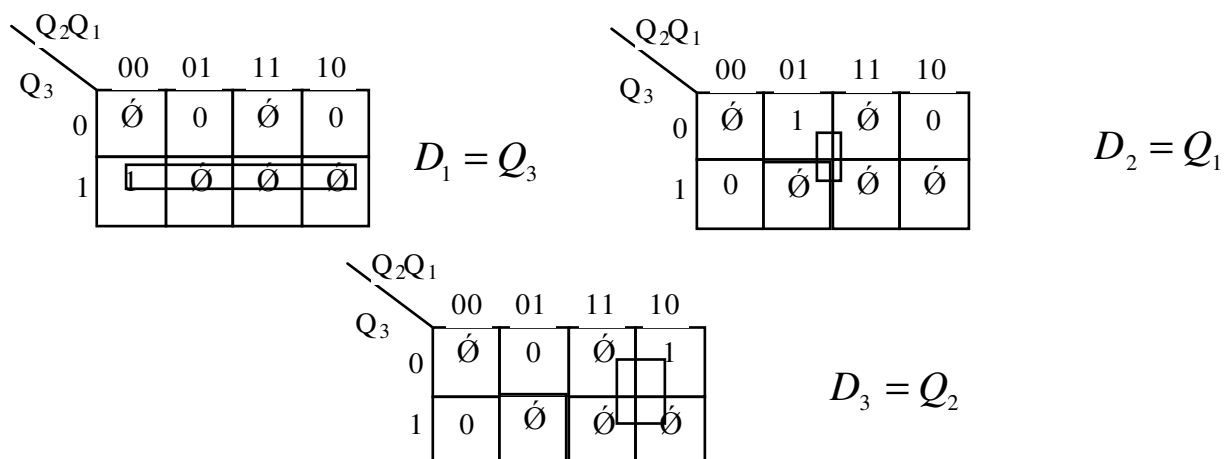


Рисунок 3.39 – Карти Карно

Відповідно до отриманих рівнянь проектуємо електричну принципову схему кільцевого регістру зсуву (рис. 3.40).

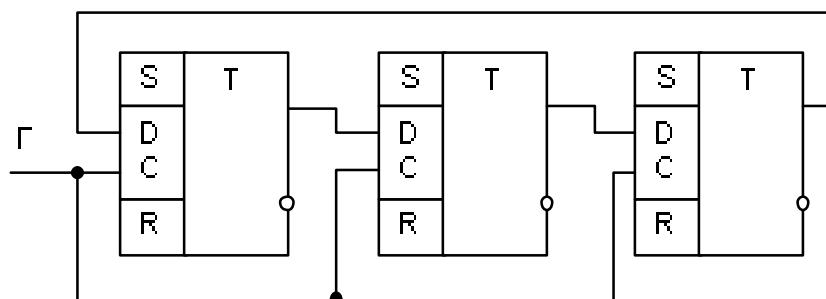


Рисунок 3.40 – Принципова схема кільцевого регістру зсуву на *D*-тригерах

3.7 Лічильники імпульсів

Лічильником називається ПП, який послідовно змінює свій стан у визначеному для даної схеми порядку, повертаючись до початку циклу після кожних K вхідних сигналів. Значення K , яке відповідає числу станів лічильника, є модулем чи коефіцієнтом перерахування лічильника. Розглянемо лічильники на основі тригерів. Код числа, записаного в тригери лічильника, може бути представлений у наступному вигляді:

$$Q = Q_m Q_{m-1} \dots Q_2 \cdot Q_1,$$

де Q_i – стан i -го тригера, $i = 1, 2, \dots, m$.

Один з можливих станів лічильника приймається за початковий – Q^0 . Запис Q^m відповідає стану лічильника після надходження на його вхід m -го вхідного сигналу. Якщо порядок зміни станів тригерів відповідає послідовності двійкових чисел, то кількість станів лічильника відповідно дорівнює $K=2^m$. Такі лічильники називають *двійковими*. У загальному випадку ($K=a^m$) пристрій називають лічильником за модулем числа a .

Розглянемо *двійкові лічильники з послідовним переносом*. У таких лічильниках вхідний сигнал впливає тільки на перший тригер, який виробляє перемикаючий сигнал для наступного. Перевагою цих лічильників є простота реалізації схеми й можливість нарощування їхньої розрядності. У підсумовуючому лічильнику з послідовним переносом кожен вхідний імпульс збільшує значення двійкового числа, яке записане в лічильник, на одиницю. Правило роботи підсумовуючого, наприклад, трирозрядного лічильника може бути подано у вигляді таблиці переключень лічильника (табл. 3.19). Використовуючи таблицю перемикань, можна визначити необхідний тип першого тригера, орієнтуючись на Q_1 , після чого варто виявити спосіб з'єднання тригерів. Очевидно, що першим у схемі повинен бути *T*-тригер, тому що він перемикається кожним вхідним сигналом. Оскільки з табл. 3.25 випливає, що другий тригер змінює свій стан, коли перший тригер переходить у рівень 0, то необхідно подати сигнал перемикання на вхід другого тригера з інверсного виходу першого. Аналогічні умови роботи будуть у третього тригера. Схема такого лічильника наведена на рис. 3.41.

Таблиця 3.25 – Таблиця переключень

Q^n			n
Q_3^n	Q_2^n	Q_1^n	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

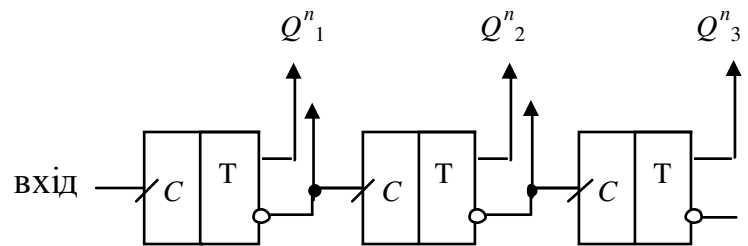


Рисунок 3.41 – Схема організації трирозрядного лічильника

У *віднімаючого лічильника* з кожним вхідним імпульсом значення двійкового числа, яке записано в ньому, зменшується. Правила роботи віднімаючого лічильника мають наступний вигляд:

$$Q^n = Q^{n-1} - 1, \text{ якщо } Q^{n-1} \neq 0;$$

$$Q^n = 2^m - 1, \text{ якщо } Q^{n-1} = 0.$$

Склавши таблицю перемикань (табл. 3.26), можна тим же способом визначити принципову схему віднімаючого лічильника (рис. 3.42).

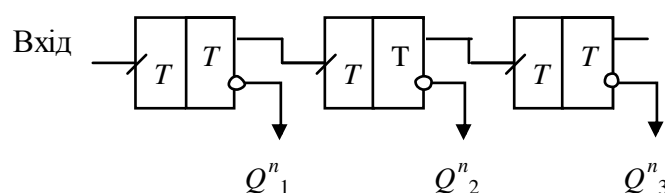


Рисунок 3.42 – Схема організації віднімаючого лічильника

Таблиця 3.26–
Таблиця перемикачів

Q^n			n
Q_3^n	Q_2^n	Q_1^n	
0	0	0	0
1	1	1	1
1	1	0	2
1	0	1	3
1	0	0	4
0	1	1	5
0	1	0	6
0	0	1	7

Таблиця 3.27–
Таблиця істинності

x	F	Q	T
0	0	0	0
0	0	1	0
0	0	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Реверсивний лічильник може працювати як підсумовуючий так і віднімаючий. Вибір режиму роботи здійснюється за допомогою управляючого сигналу, який надходить на вхід F лічильника. Для організації реверсивного лічильника скористаємось послідовним з'єднанням T -тригерів, доповнивши елементами комутації. Елемент комутації повинен сформувати такі сигнали:

- у режимі підсумовування $T_i = \overline{Q_{i-1}}$;
- у режимі віднімання $T_i = Q_{i-1}$.

Варто врахувати, що сигнал на вході F може змінюватися тільки при відноному сигналі лічильника $X = 0$. Складена по словесному опису таблиця істинності для вхідного сигналу i -го тригера (табл. 3.27) дозволяє записати для T_i таку структурну формулу

$$T_i = X \overline{F} Q_{i-1} + X F \overline{Q_{i-1}} = X \overline{\overline{F} Q_{i-1}} + X F Q_{i-1}$$

і розробити структурну схему елемента комутації, наведену на (рис. 3.43).

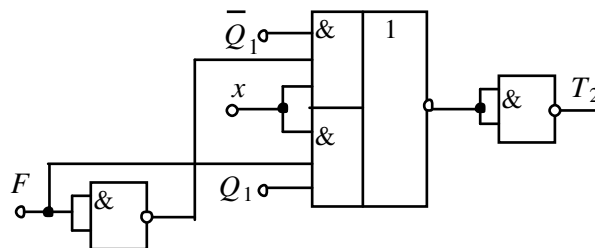


Рисунок 3.43 – Схема організації управління реверсивним лічильником

Недоліком лічильників з послідовним переносом є мала швидкодія, що обумовлено послідовним спрацьовуванням тригерів. Означеного недоліку не має у лічильників з паралельним переносом. У таких лічильниках вхідний сигнал впливає на всі тригера одночасно та кожен попередній тригер виробляє управляючі сигнали для всіх послідуочих.

Розглянемо принцип побудови підсумовуючого лічильника з паралельним переносом на JK -тригерах. Для переводу JK -тригера до

рахункового режиму роботи потрібно виконати такі умови: $j=k=1$. Тому рівняння алгебри логіки першого тригеру наступні: $j_1=k_1=1$. Оскільки, кожен попередній тригер формує управляючі сигнали на послідуєчих, то рівняння алгебри логіки для другого, третього та четвертого тригерів такі

$$j_2 = k_2 = Q_1, \quad j_3 = k_3 = Q_1Q_2, \quad j_4 = k_4 = Q_1Q_2Q_3.$$

На підставі цих рівнянь будемо електричну принципову схему лічильника з паралельним переносом (рис. 3.44).

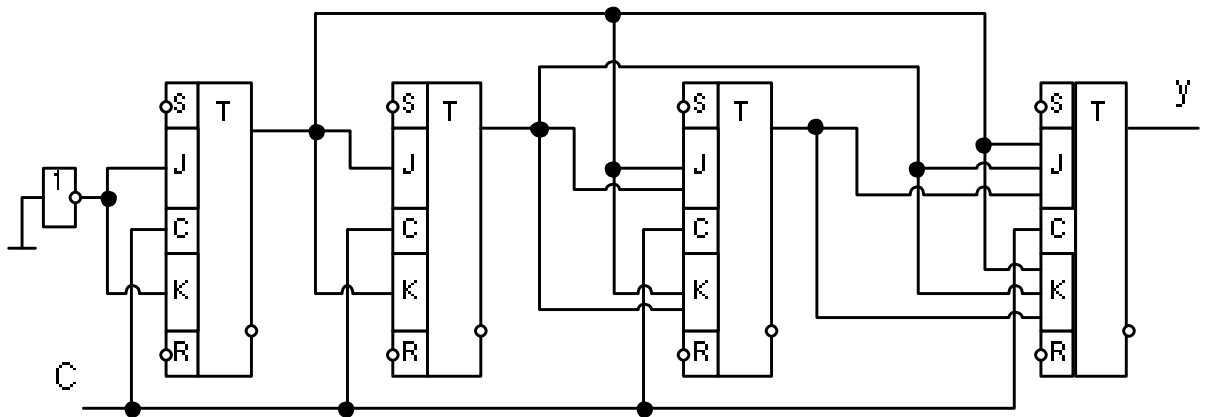


Рисунок 3.44 – Лічильник з паралельним переносом

При побудові лічильників з довільним коефіцієнтом переліку застосовується метод виключення зайвих станів у двійкових лічильниках. Для побудови лічильника з коефіцієнтом переліку $k < 2^n$ використовується лічильник з коефіцієнтом переліку 2^n . Визначаємо кількість лишніх станів таким чином: $l=2^n-k$. Виключення зайвих станів лічильника виконуємо за допомогою схем управління на логічних елементах. Ці схеми доповнюють принципову схему лічильника. В залежності від обраного способу організації лічильника можна запропонувати п'ять варіантів реалізації схеми. Чотири варіанти реалізуються двійковим лічильником зі зворотнім зв'язком.

1. Логічна схема зворотного зв'язку фіксує стан лічильника $(k-1)$, який визначає кінець рахування. Вихідний сигнал цієї схеми переводить лічильник у початковий стан.

Розробку лічильника з довільним коефіцієнтом переліку розглянемо на конкретному прикладі $k=6$. Визначаємо кількість тригерів, яка потрібна для побудови принципової схеми лічильника

$$n = \log_2 k \approx 3.$$

Складаємо таблицю (табл. 3.28) переключень підсумовуючого лічильника, який будемо на JK-тригерах. В цій таблиці комбінацію переходу лічильника у початковий стан виділено. На підставі цієї комбінації складаємо рівняння алгебри логіки схеми початкової установки лічильника

$$CB = \overline{Q_1}Q_2Q_3.$$

Електрична принципова схема лічильника з коефіцієнтом переліку 6 наведена на рис. 3.45.

Таблиця 3.28 – Таблиця переключень лічильника

k	Q_1	Q_2	Q_3
0	0	0	0
1	1	0	0
2	0	1	0
3	1	1	0
4	0	0	1
5	1	0	1
6	0	1	1
7	1	1	1

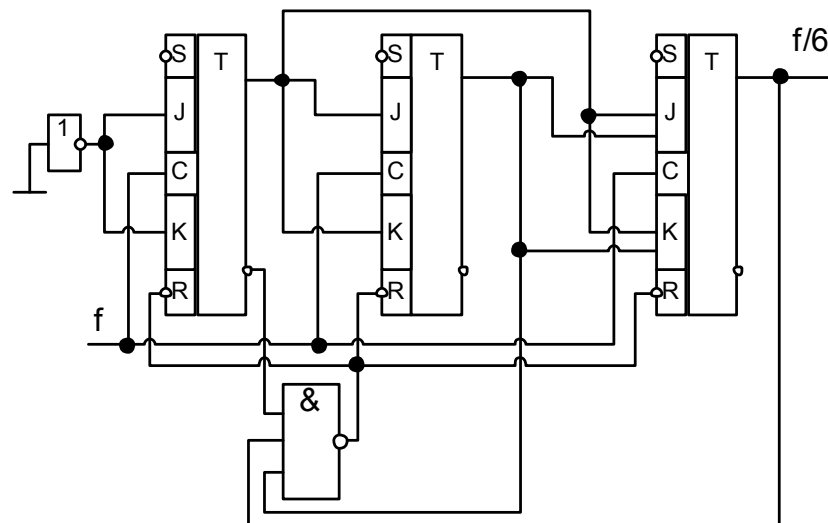


Рисунок 3.45 – Лічильник з коефіцієнтом переліку 6

2. У тригери двійкового лічильнику завдяки зворотному зв'язку записується код, який відповідає кількості зайвих станів. Розробимо лічильник коефіцієнтом переліку $k=6$. Складаємо таблицю (табл. 3.29) переключень підсумовуючого лічильника, який будемо на JK -тригерах. В цій таблиці комбінація початкового стану лічильника виділено. На підставі цієї комбінації складаємо рівняння алгебри логіки початкової установки

$$ПУ = \overline{Q_1} Q_2 \overline{Q_3}.$$

Таблиця 3.29 – Таблиця переключень лічильника

k	Q_1	Q_2	Q_3
0	0	0	0
1	1	0	0
2	0	1	0
3	1	1	0
4	0	0	1
5	1	0	1
6	0	1	1
7	1	1	1

Для реалізації рівняння алгебри логіки достатньо вихід схеми початкової установки підключити до входу S_2 . Електрична принципова схема лічильника з коефіцієнтом переліку 6 наведена на рис. 3.46.

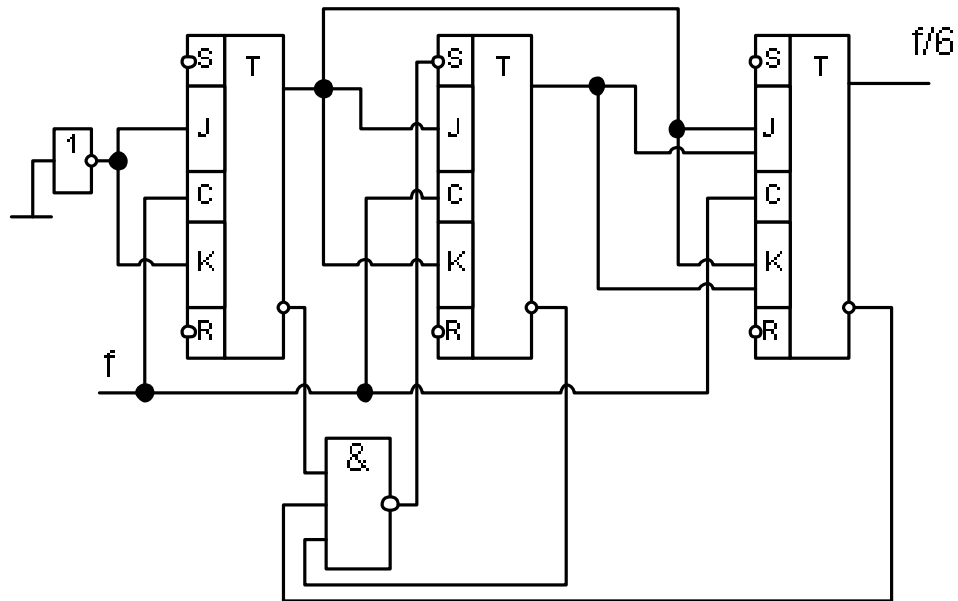


Рисунок 3.46 – Лічильник з коефіцієнтом переліку 6

3. Передбачає використання у якості сигналу для початкової установки тригерів деякої довільної кодової комбінації. Розробимо лічильник коефіцієнтом переліку $k=6$. Складаємо таблицю (табл. 3.30) переключень підсумовуючого лічильника, який будемо на JK -тригерах. В цій таблиці комбінації початкового стану лічильника та формування сигналу зворотного зв'язку виділено. Складаємо рівняння алгебри логіки для початкової установки лічильника та коду, на підставі якого формується сигнал зворотного зв'язку

$$ПУ = \overline{Q_1 Q_2 Q_3},$$

$$СБ = Q_1 Q_2 Q_3.$$

Для реалізації рівняння алгебри логіки достатньо вихід схеми початкової установки підключити до входів R_2 та R_3 .

Таблиця 3.30 – Таблиця переключень лічильника

k	Q_1	Q_2	Q_3
0	0	0	0
1	1	0	0
2	0	1	0
3	1	1	0
4	0	0	1
5	1	0	1
6	0	1	1
7	1	1	1

Електрична принципова схема лічильника з коефіцієнтом переліку 6 наведена на рис. 3.47.

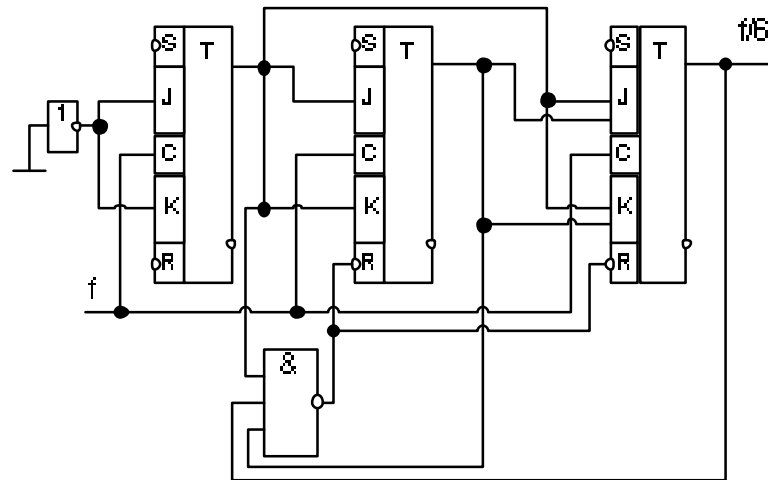


Рисунок 3.47 – Лічильник з коефіцієнтом переліку 6

4. Використання стандартної ІС К155ІЕ7, яка являє собою реверсивний лічильник зі схемою попередньої установки тригерів. Код l записується у тригери лічильника сигналом переносу. Цей сигнал подається на вхід попередньої установки V . Задати код l можна шляхом підключення деяких інформаційних входів лічильника D_0, D_1, D_2 та D_3 до нульової шини джерела живлення або за допомогою схеми на логічних елементах. Це дозволяє отримати коефіцієнт переліку лічильника у межах від 1 до 16, причому $k=16-l$ у режимі підсумовування та $k=l+1$ у режимі віднімання.

Спроекуємо лічильник з коефіцієнтом переліку 10 на базі ІС К155ІЕ7. Визначимо кількість станів лічильника, яку слід виключити, $k=16-l=6$. Числу 6 відповідає такий двійковий код 0110. У відповідності з цим стан інформаційних входів лічильника такий: $D_0=0, D_1=1, D_2=1$ та $D_3=0$. Оскільки мікросхеми серії К155 мають таку властивість – на обірваному вході завжди присутня 1. Тому для завдання потрібного коду потрібно підключити інформаційні входи лічильника D_0 та D_3 до нульової шини джерела живлення. Електрична принципова схема лічильника з коефіцієнтом переліку 10 наведена на рис. 3.48.

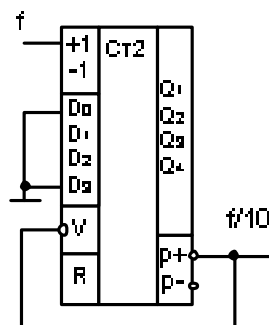


Рисунок 3.48 – Лічильник з коефіцієнтом переліку 10

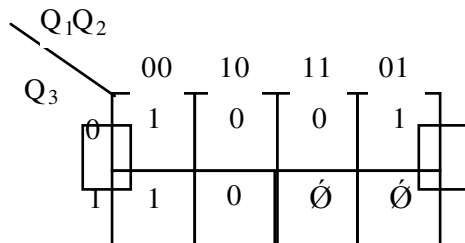
5. Синтез лічильника імпульсів з довільним коефіцієнтом переліку. У результаті синтезу отримуємо рівняння алгебри логіки для схем на логічних елементах, які утворюють зворотні зв'язки між тригерами. Ці зворотні зв'язки обмежують кількість станів двійкового лічильника.

Розробимо лічильник коефіцієнтом переліку $k=6$. Складаємо таблицю переключень підсумовуючого лічильника, який побудуємо на D -тригерах (табл. 3.25).

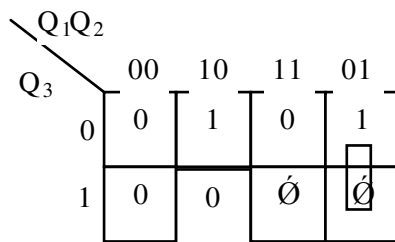
Таблиця 3.25 – Таблиця переключень лічильника

Q_1	Q_2	Q_3	D_1	D_2	D_3
0	0	0	1	0	0
1	0	0	0	1	0
0	1	0	1	1	0
1	1	0	0	0	1
0	0	1	1	0	1
1	0	1	0	0	0
ф	ф	ф	ф	ф	ф
ф	ф	ф	ф	ф	ф

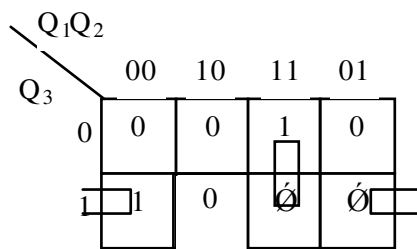
Складаємо карти Карно для станів та переключень тригерів лічильника (рис. 3.49). На підставі отриманих виразів будемо електричну принципову схему лічильника (рис. 3.50).



$$D_1 = \overline{Q_1}$$



$$D_2 = Q_1 \overline{Q_2} \overline{Q_3} + \overline{Q_1} Q_2$$



$$D_3 = Q_1 Q_2 + \overline{Q_1} Q_3$$

Рисунок 3.49 – Карти Карно для станів та переключень лічильника

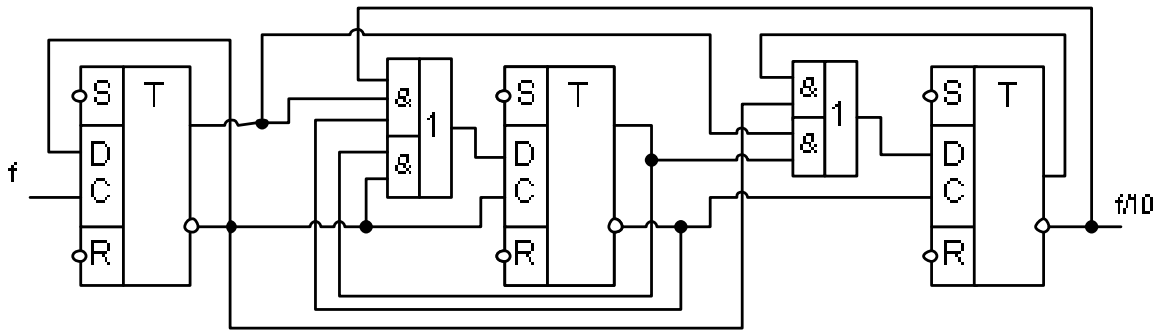


Рисунок 3.50 – Лічильник з коефіцієнтом переліку 10

Лічильники зі змінним коефіцієнтом переліку будуються на базі стандартних ІС. Можливі три варіанти реалізації таких лічильників.

1. Використання стандартної ІС К155ІЕ7, схема управління початковою установкою якої реалізовано на базі двійкового лічильника. Сигнал з виходу переносу реверсивного лічильника подається на вхід двійкового лічильника та на вхід початкової установки V . Виходи двійкового лічильника підключено до інформаційних входів реверсивного лічильника D_0, D_1, D_2 та D_3 . Коефіцієнт переліку утвореного пристрою $k=16-l$. Оскільки двійковий код підсумовуючого двійкового лічильника у процесі роботи пристрою збільшується, то коефіцієнт його переліку зменшується циклічно у межах від 16 до 1. Схему пристрою наведено на рис. 3.51.

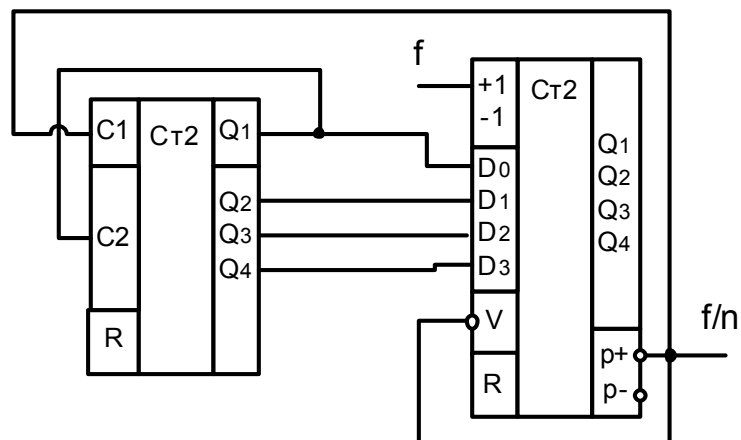


Рисунок 3.51 – Лічильник зі змінним коефіцієнтом переліку

2. Використання стандартної ІС К155ІЕ8, яка являє собою 6-розрядний лічильник зі змінним коефіцієнтом переліку. Умовне позначення цієї мікросхеми наведено на рис. 3.52. На виході y_2 мікросхеми частота вихідного сигналу дорівнює $f/64$, а на виході y_1 – $(Nf)/64$, де N – код на інформаційних входах $D_0, D_1, D_2, \dots, D_5$. Таким чином, код інформаційних входів визначає кількість імпульсів, яка з'явиться на виході y_1 за проміжок часу, який формується на виході y_2 . Недоліком цього лічильника зі змінним коефіцієнтом переліку є нерівномірність розміщення у часі вихідних імпульсів.

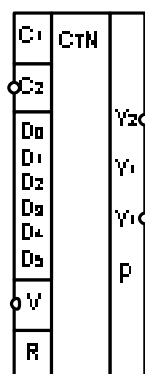


Рисунок 3.52 – Умовне позначення мікросхеми K155IE8

3. Використання стандартної ІС K564IE15, яка являє собою програмований подільувач частоти. Коефіцієнт поділу частоти цієї мікросхеми визначається наступним чином

$$N = M(1000p_m + 100p_c + 10p_d + p_{od}) + p_o,$$

де $M = 2, 4, 5, 8, 10$ – модуль числа; p_m, p_c, p_d, p_{od} – відповідно множники тисячі, сотні, десятку та одиниць; p_o – залишок. Коефіцієнт поділу частоти цією мікросхемою змінюється у межах від 3 до 21327.

Лічильники імпульсів можливо побудувати на базі регістрів зсуву. При замиканні регістра зсуву у кільце шляхом підключення виходу останнього триггеру зі входом першого отримаємо лічильник, який працює таким чином (табл. 3.26).

Таблиця 3.26 – Таблиця переключень лічильника

n	Q_1	Q_2	Q_3
0	1	0	0
1	0	1	0
2	0	0	1
3	1	0	0

Початкова установка цього лічильника наступна: у перший тригер записується 1, а в інші – 0. Коефіцієнт переліку такого лічильника дорівнює кількості тригерів, які його утворюють. Принципова схема 3-розрядного лічильника наведена на рис. 3.53. Недоліком такої схеми побудови лічильника є необхідність захисту від випадкових впливів.

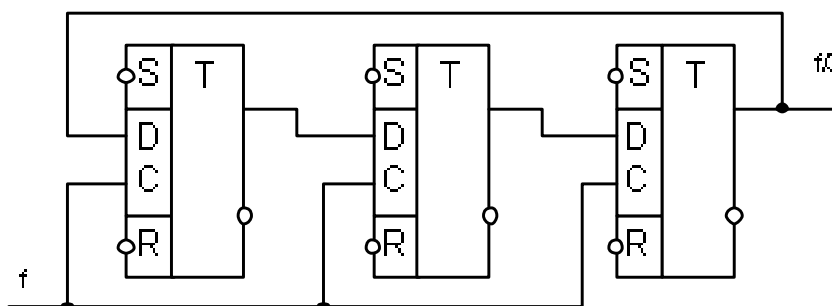


Рисунок 3.53 – Лічильник з коефіцієнтом переліку 3

Інша різновидність лічильників на основі регістрів зсуву отримала назву лічильників Джонсона. *Особливості роботи лічильників Джонсона такі:*

- з приходом на його вхід кожного імпульсу змінює свій стан тільки один тригер;
- кожен тригер змінює свій стан як можна ріже (наприклад, два рази за один цикл роботи лічильника).

Таблиця переключень тригерів 3-розрядного лічильника Джонсона має шість строк та, відповідно, $k=2n=6$, де n – кількість розрядів (табл. 3.27).

Таблиця 3.27 – Таблиця переключень лічильника

n	Q_1	Q_2	Q_3
0	0	0	0
1	1	0	0
2	1	1	0
3	1	1	1
4	0	1	1
5	0	0	1
6	0	0	0

Для побудови лічильника Джонсона використаємо D -тригери. Складемо таблицю переключень (табл. 3.28) та за допомогою карт Карно (рис. 3.54) одержимо рівняння алгебри логіки для побудови лічильника.

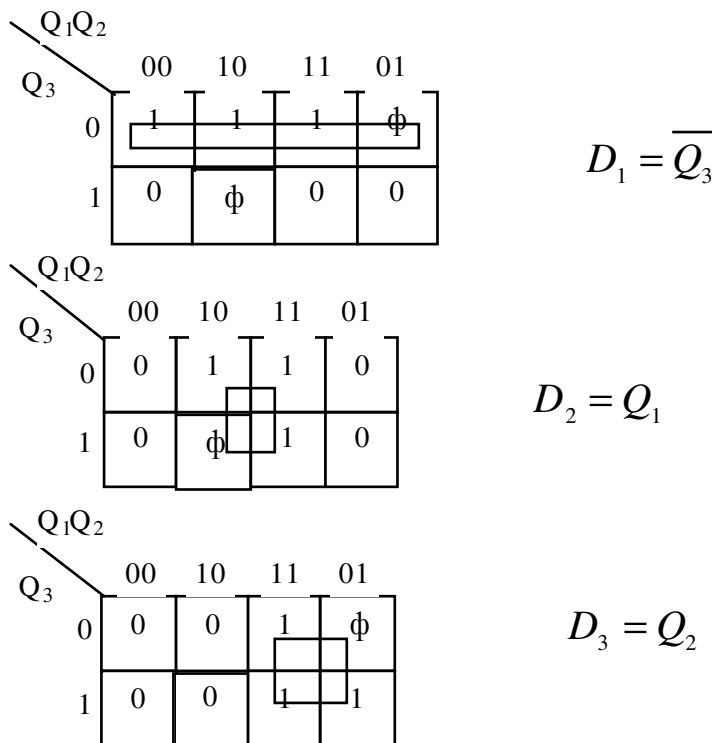


Рисунок 3.54 – Карти Карно лічильника Джонсона

Таблиця 3.28 – Таблиця переключень лічильника Джонсона

Q_1	Q_2	Q_3	D_1	D_2	D_3
0	0	0	1	0	0
1	0	0	1	1	0
0	1	0	ф	ф	ф
1	1	0	1	1	1
0	0	1	0	0	0
1	0	1	ф	ф	ф
0	1	1	0	0	1
1	1	1	0	1	1

На підставі отриманих рівнянь будемо принципову схему лічильника Джонсона на D -тригерах (рис. 3.55).

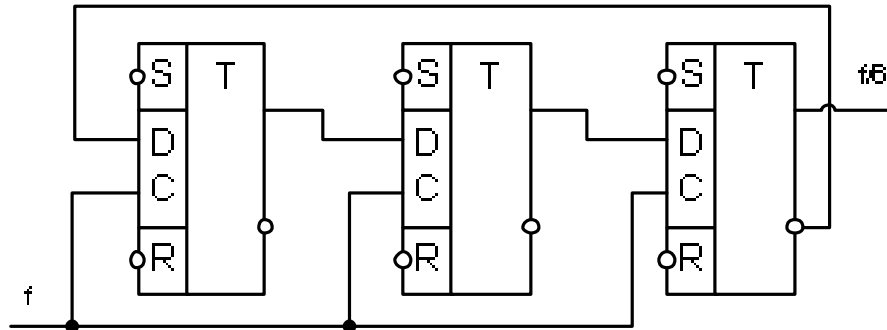


Рисунок 3.55 – Лічильник Джонсона з коефіцієнтом переліку 6

Поліноміальні лічильники імпульсів будуються на основі регістрів зсуву з використанням підсумовувача по модулю 2 у якості пристрою для формування зворотних зв'язків. На рис. 3.56 наведена одна із можливих схем побудови 3-розрядного лічильника. У табл. 3.29 наведено послідовність зміни станів тригерів поліноміального лічильника. При використанні другої схеми реалізації зворотного зв'язку послідовність станів тригерів буде іншою. У загальному випадку довжина та вид послідовності імпульсів, яку генерує лічильник, залежить від кількості розрядів та виду зворотних зв'язків.

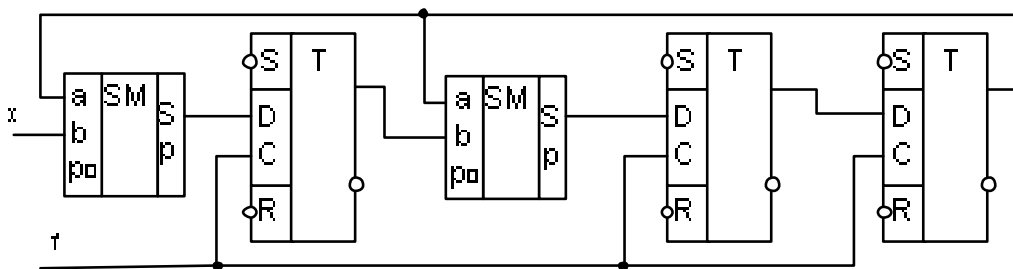


Рисунок 3.56 – Поліноміальний лічильник

Статистичні характеристики послідовності одиниць та нулів, яку

отримуємо на виходах тригерів при достатньо великій розрядів регістру, наближаються к характеристикам випадкової послідовності. У зв'язку з цим поліноміальні лічильники використовуються у системах керування електричним транспортом в якості генераторів псевдовипадкових послідовностей при перевірці апаратури. Також ці лічильники використовуються для формування циклічних кодів, які забезпечують виявлення та виправлення похибок при передачі дискретних повідомлень.

Таблиця 3.29 – Таблиця переключень поліноміального лічильника

N	x	Q_1	Q_2	Q_3
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	0	1	1	0
5	0	0	1	1
6	0	1	1	1
7	0	1	0	1
8	0	1	0	0

3.8 Допоміжні цифрові пристрої

Робота цифрового пристрою у складі системи керування електричним транспортом неможлива без застосування допоміжних пристроїв таких як: генератори та формувачі імпульсів.

При побудові генератора типу мультівібратор на основі логічних елементів використовуються підсилювальні властивості інверторів. Для забезпечення стійких коливань необхідно забезпечити умови роботи інвертора по постійному струму на лінійної ділянці його передаточної характеристики. Після цього необхідно до складу пристрою ввести позитивний зворотній зв'язок за допомогою одного чи декілька конденсаторів. Приклади побудови генераторів подано на рис. 3.57.

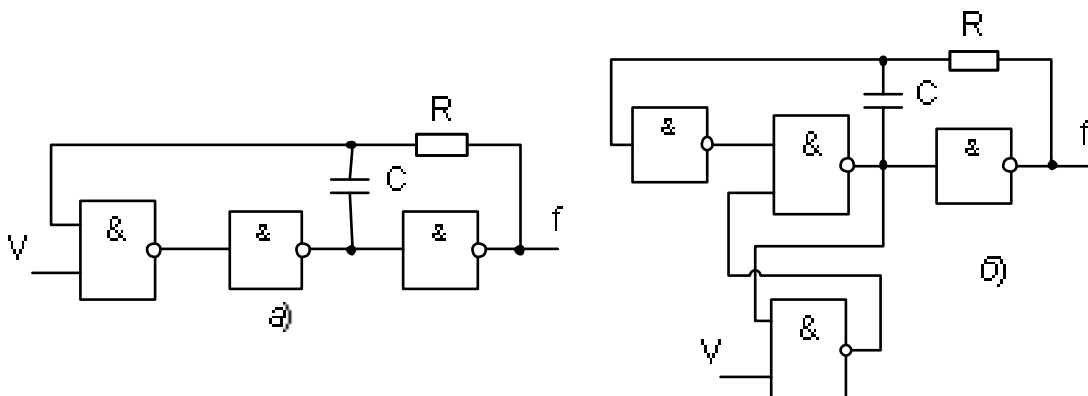


Рисунок 3.57 – Схеми генераторів зі зривом коливань

Стабілізація режимів інверторів по постійному струму забезпечується за рахунок ланцюга зворотного зв'язку через резистор. Позитивний зворотній зв'язок реалізується за рахунок конденсатору. Вхід V у цих схемах використовується для того, щоб припинити генерацію імпульсів. Причому у схемі рис. 3.57а для припинення коливань потрібно, щоб $V=1$, а для схеми рис. 3.57б – $V=0$. Особливістю схеми рис. 3.57б є те, що при подачі забороняючого сигналу мультивібратор закінчує роботу не миттєво, а після формування імпульсу на виході.

Генератори імпульсів можна побудувати також на двох або одному інверторі (3.58).

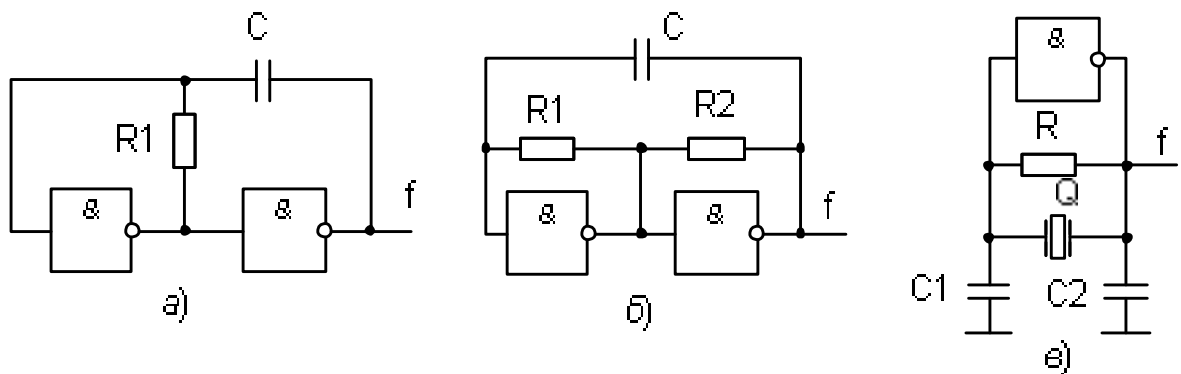


Рисунок 3.58 – Схеми генераторів

Можна також будувати генератори на основі триггеру Шмітта. Приклад реалізації принципової схеми подано на рис. 3.59.

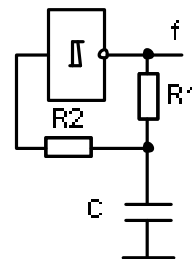


Рисунок 3.59 – Схема генератора на базі триггеру Шмітта

Формувачі напруги прямокутної форми будуються на базі триггеру Шмітта. Вони використовуються для перетворень сигналів довільної форми у цифровий сигнал. Приклади побудови цих пристроїв подано на рис. 3.60.

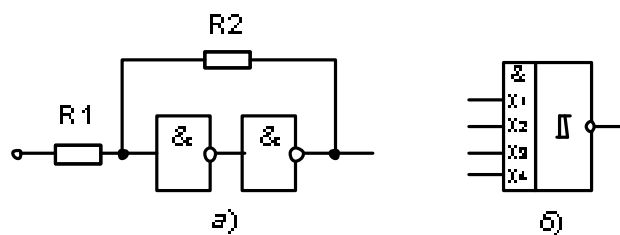


Рисунок 3.60 – Схема формувача прямокутної напруги

На рис. 3.60а показана схема триггеру Шмітта на базі інверторів. Позитивний зворотній зв'язок між інверторами утворює резистор R_1 , опір якого складає 22 Ом. Вхідна напруга подається через резистор R_2 , опір якого складає 470 Ом. Збільшення цього опору впливає на коефіцієнт позитивного зворотного зв'язку та зменшує чутливість схеми до вхідної напруги. Приклад триггеру Шмітта на базі стандартної ІС наведено на рис. 3.60б.

Формувачі імпульсів заданої тривалості (одновібратори) формують вихідний імпульс при подачі на їх вхід перепаду сигналу. Деякі приклади побудови цих пристроїв наведено на рис. 3.61.

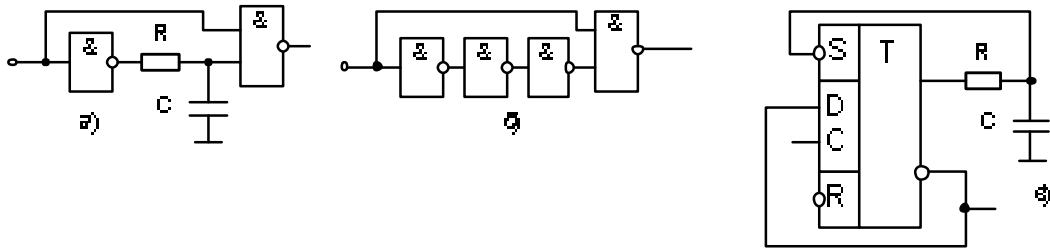


Рисунок 3.61 – Схеми формувачів імпульсів заданої тривалості

Початковий стан одновібраторів (рис. 3.61а, б) визначається сигналом логічного нуля на їхніх входах. При цьому на виході інвертора сигнал логічної одиниці. Конденсатор заряджено й на відповідному вході схеми І-НІ присутня логічна 1. Подача на вхід одновібратору сигналу логічної 1 призводить до того, що другому вході схеми І-НІ з'являється логічна 1. На виході одновібратору з'являється сигнал логічного 0. На виході інвертору з'являється сигнал логічного 0, що призводить до розряду конденсатора. Коли напруга на конденсаторі стане меншою ніж рівень логічної 1 спрацьовує схема І-НІ, на виході пристрою з'являється сигнал логічної 1. Тривалість сформованого імпульсу дорівнюється часу розряду конденсатора до рівня спрацьовування схеми І-НІ. Тривалість вихідного сигналу схеми одновібратору (рис. 3.61б) визначається запізненням, яке дорівнюється часу спрацьовування трьох послідовно включених інверторів. У одновібраторі (рис. 3.61в) D-тригер знаходиться у одиничному стані. На його прямому виході знаходиться сигнал логічного 0, тому конденсатор заряджено. Поява сигналу логічної 1 на С-вході D-тригеру переводить його у нульовий стан. На його прямому виході з'являється сигнал логічної 0, тому конденсатор починає розряджатися. Коли напруга на конденсаторі стане меншою ніж рівень логічної 1, D-тригер переходить у одиничний стан. Тривалість сформованого імпульсу дорівнюється часу розряду конденсатора до рівня спрацьовування D-тригеру. Одновібратори випускаються також у вигляді ІС (К155АГ1, К155АГ3).

Формувачі імпульсів від механічних контактів широко використовуються у складі систем керування електричним транспортом для формування чіткого переходу від 0 до 1, або навпаки. Специфіка роботи цього пристрою полягає в усуненні ефекту «дребезг контактів». Приклади

схемної побудови формувачів імпульсів від механічних контактів подано на рис. 3.62.

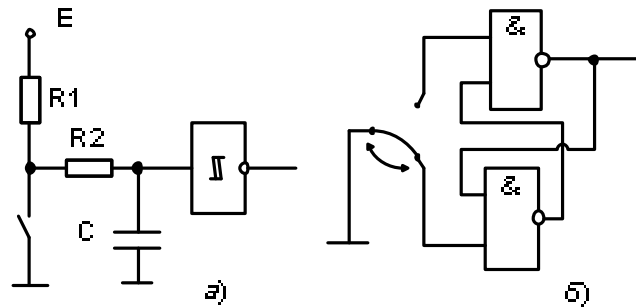


Рисунок 3.62 – Схеми формувачів імпульсів від механічних контактів

Формувач (рис. 3.62а) утворюють тригер Шмітта та фільтр нижніх частот, який усуває ефект «дребезг контактів». При замиканні перемикача напруга на вході фільтру подає до нуля. Змінна складова вхідної напруги згладжується фільтром. Постійна часу фільтру нижніх частот підбирається такою, щоб амплітуда пульсацій напруги на його виході була меншою ніж ширина петлі гістерезису тригеру Шмітта. Вхідний опір останнього повинен бути значно більшим, ніж вихідний опір фільтру. Формувач (рис. 3.62б) являє собою RC-тригер. При переключенні перемикача тригер реагує на перше замикання контактної пари.

Контрольні запитання

1. Поясніть принцип дії T -тригера.
2. Що потрібно зробити для того, щоб D -тригер працював у цифрових схемах як T -тригер?
3. Як працює JK -тригер?
4. Які методи синтезу тригерних пристроїв вам відомі?
5. Сформулюйте правило роботи регістра.
6. Види регістрів.
7. Назвіть відомі вам методи побудови лічильників.
8. Порядок синтезу лічильника з довільним коефіцієнтом переліку.
9. Принцип роботи генераторів на логічних елементах.
10. У чому полягає принцип дії формувачів імпульсів заданої тривалості?

4 ЦИФРОВІ ПРИСТРОЇ ФОРМУВАННЯ ТА ОБРОБКИ ДИСКРЕТНИХ СИГНАЛІВ

Цифрові пристрої формування та обробки дискретних сигналів використовуються для формування й перетворення різного класу сигналів. Розглянемо приклад реалізації цифрового формування гармонічного сигналу коливання в збудниках радіопередавача, у кореляторах та погоджених фільтрах, у системах фазового автопідстроювання частоти.

4.1 Цифрові пристрої формування сигналів

Такі пристрої можуть бути виконані на основі запам'ятовуючих (ЗП) та цифроаналогових (ЦАП) пристроїв. Дійсно, оскільки робота ЗП звичайно описується M -розрядною (M – число входів ЗП) логічною функцією, то за його допомогою можна сформувати таблично задані складні функції. Відтворення необхідної функції, вибіркові значення якої записані в осередках постійних запам'ятовуючих пристроїв (ПЗП), звичайно відбувається шляхом формування номерів адрес осередків ПЗП за допомогою, наприклад, двійкових лічильників СТ2 (рис. 4.1). Включення на виході ПЗП (ROM) цифроаналогового перетворювача (DAC) та фільтра нижніх частот дозволяє побудувати функціональний генератор або формувач аналогових сигналів.

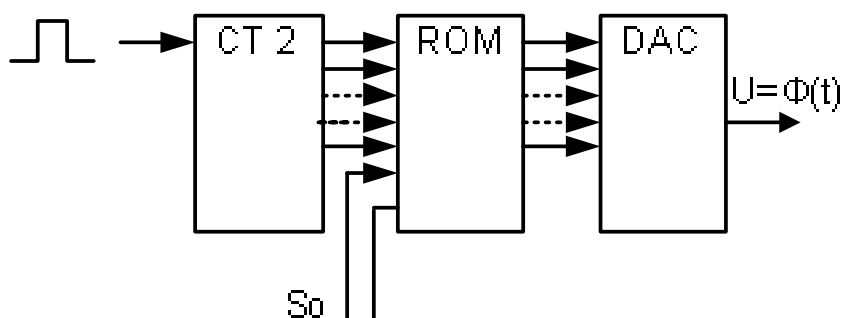


Рисунок 4.1 – Структурна схема функціонального генератора

Розглянемо принципову схему функціонального генератора (рис. 4.2). У ПЗП D3...D5 записуються в додатковому коді з інвертованим знаковим розрядом 12-розрядні кодові слова, які відповідають вибірковим значенням вихідної функції. З виходів ПЗП коди надходять на входи ЦАП (D6). Зчитуванням інформації з ПЗП управляють двійкові лічильники D1 та D2. Вхідна послідовність U_3 надходить безпосередньо на вхід лічильника D1. Обсяг ПЗП, необхідний для формування 256 вибірових значень сигналу $S(t)$ при $x = 8$, складає 256x12 біт. Максимальна частота вихідного сигналу $S(t)$ обмежується швидкодією застосованого цифроаналогового перетворювача D6.

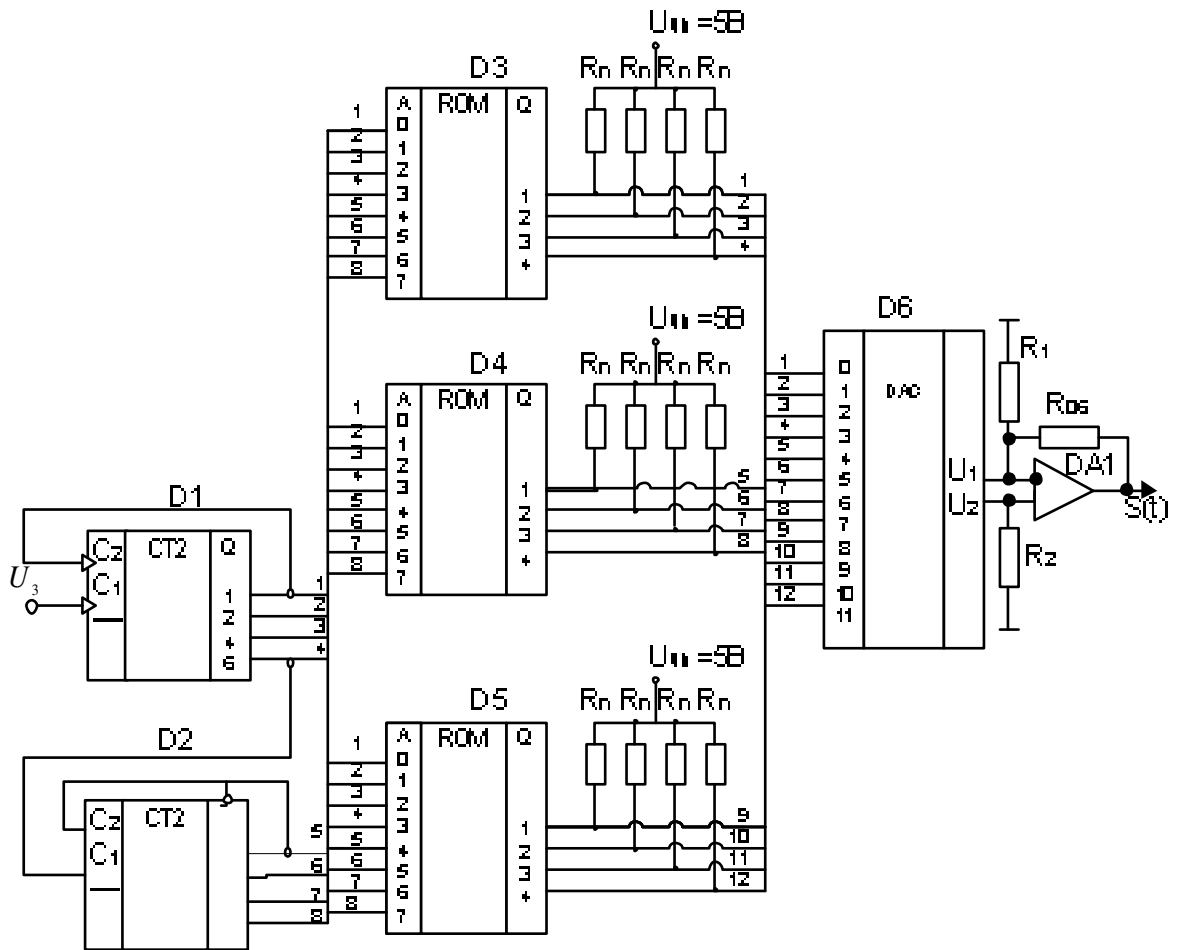


Рисунок 4.2 – Принципова схема формувача гармонічного сигналу

4.2 Цифрові корелятори

Розглянемо такий алгоритм когерентної обробки сигналів цифровими кореляторами при обчисленні U_0 у наступній формі

$$U_0 \cong U_0^{(q)} = \frac{\Delta t}{2} \sum_{K=0}^{N-1} A_x^{(n)}[k] A^{(p)}[k],$$

де $A_x^{(n)}[k]$ та $A^{(p)}[k]$ – послідовності двійкових кодів, які еквівалентні для оброблюваного коливання $x(t)$ й прийнятого сигналу $s_0(t)$; n, p – розрядності розглянутих кодів; q – кількість розрядів числа на виході цифрового корелятора.

Структурна схема цифрового корелятора наведена на рис. 4.3. На цьому рисунку позначено: ФНЧ – фільтр нижчих частот; ЦП – цифровий перемножник n -розрядних та p -розрядних чисел у деякі фіксовані моменти часу $K \cdot \Delta t$; ЦУ – цифровий пристрій формування послідовності чисел $A^{(p)}[k]$; СН – цифровий суматор-накопичувач $N(n+p)$ -розрядних кодів.

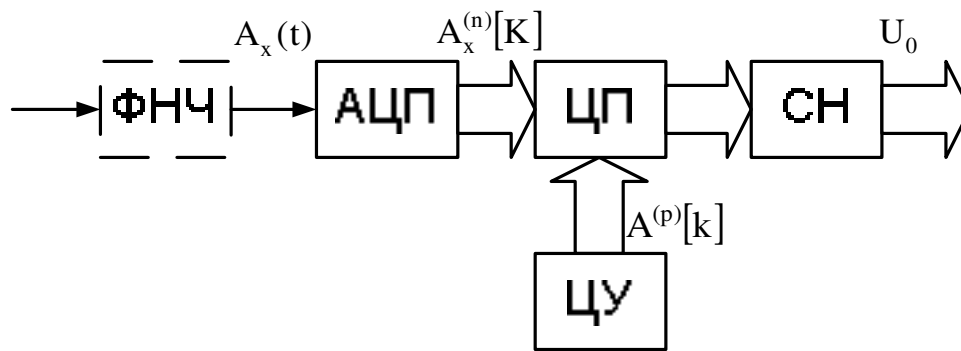


Рисунок 4.3 – Структурна схема цифрового корелятору

Зупинимося докладніше на принципах побудови ЦП. Найбільш швидкодіючими є цифрові перемножники рівнобіжного типу. Такі ЦП поділяються на матричні перемножники та перемножники на основі ПЗП.

Матричні перемножники (рис. 4.4а) здійснюють операцію перемноження n -розрядних та p -розрядних чисел $X^{(n)}$ й $Y^{(p)}$ відповідно до наступних алгоритмів роботи

$$Z^{(n+p)} = X^{(n)} Y^{(p)} = \sum_{i=0}^{n-1} x_i 2^i \sum_{j=0}^{p-1} y_j 2^j,$$

де x_i , y_i та z_k – значення i -го, j -го та k -го розрядів чисел $X^{(n)}$, $Y^{(p)}$ та $Z^{(n+p)}$, записаних у двійковому коді. При цьому кожне число $X^{(n)}$ подається у вигляді n -розрядного числа $(x_{n-1}, x_{n-2}, \dots, x_0)$; число $Y^{(p)}$ – у вигляді p -розрядного коду $(y_{p-1}, y_{p-2}, \dots, y_0)$ й число $Z^{(n+p)}$ у вигляді $(n+p)$ -розрядного коду $(z_{n+p-1}, z_{n+p-2}, \dots, z_0)$, причому старшими є розряди x_{n-1} , y_{p-1} та z_{n+p-1} а молодшими – x_0 , y_0 , z_0 . На входи двійкових суматорів надходять результати логічного перемноження розрядів позитивних чисел $X^{(n)}$ та $Y^{(p)}$. Двійковий суматор має входи "а" й "б", вхід переносу числа "р₀", вихід суми чисел "s" та вихід переносу числа "р₁". Кожен рядок суматорів (рис. 4.3) формує часткову суму та передає її до наступного рядку. Молодший розряд числа $Z^{(n+p)}$ утвориться окремо як результат логічного перемноження розрядів x_0 та y_0 .

Принципова схема матричного перемножувача, яка призначена для множення 4-розрядних $X^{(4)}$ та 3-розрядних кодів $Y^{(3)}$ наведена на рис 4.4б. Для логічного перемноження розрядів цих кодів використовуються елементи D1...D3(ІС 564 ЛА7), а у якості суматорів кодів D4 та D5 застосовуються 4-розрядні повні суматори (ІС 564 ІМ1). Розряди результуючого коду $Z^{(7)}$ формуються в інвертованому вигляді. На входи "а" суматорів D4 та D5 подається напруга логічної "1".

Швидкодія такого матричного перемножувача визначається швидкодією елементів, які входять до складу пристрою, та для отримання її оцінки необхідно знати самий тривалий шлях проходження сигналу по перемножнику. При використанні зазначених типів ІС максимальна частота зчитування чисел з виходу перемножника не перевищує 1 МГц.

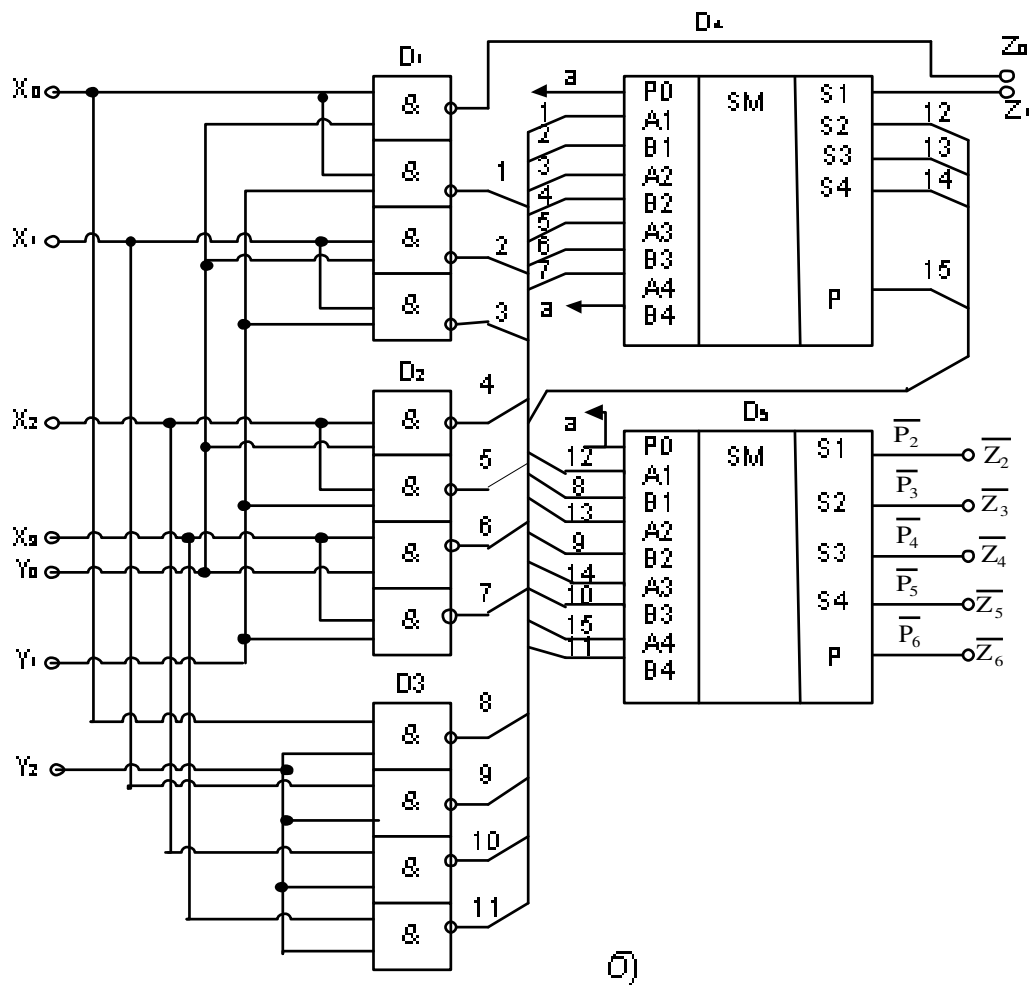
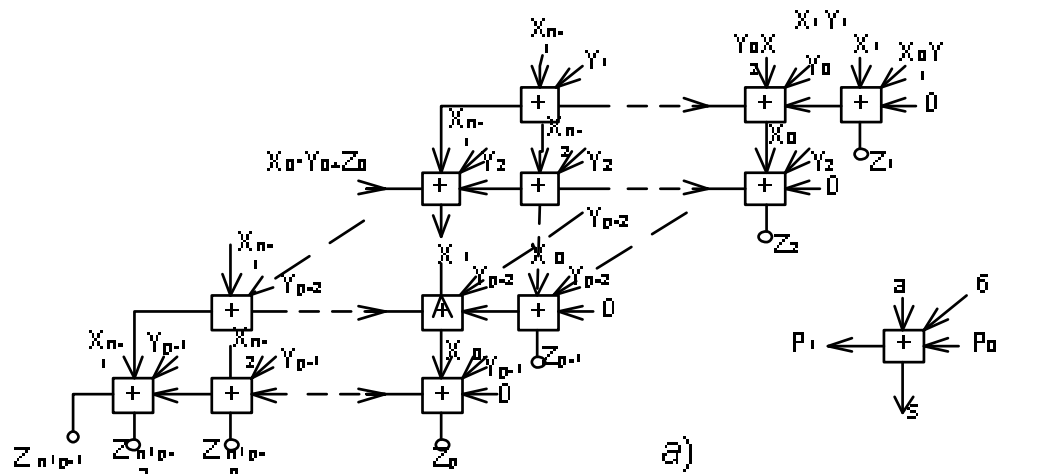


Рисунок 4.4 – Цифрові перемножники:
a – структурна схема; *б* – принципова схема

Перемножувачи цифрових кодів на основі постійних запам'ятовуючих пристроїв містять у собі ПЗП, у якому зберігаються таблиці добуток різних варіантів співмножників, причому загальне число добуток дорівнює 2^{n+p} $(n+p)$ -розрядних чисел. Основу ПЗП складають матриці комірок пам'яті з записаними в них $(n+p)$ -розрядними кодами, наприклад, шляхом

пропалювання відповідних перемичок у матриці (ІС 556РТ4, 556РТ5). Звертання до збереженої інформації виконується через адресні входи шляхом подачі на них відповідного коду адреси, а зчитування $(n+p)$ -розрядних чисел відбувається, як правило, у рівнобіжному коді з $(n+p)$ -виходів ПЗП. Швидкодія таких перемножників визначається мінімальним часом зчитування інформації з ПЗП та, як правило, вона менша, ніж у матричних перемножників.

4.3 Цифрові погодженні фільтри

Реалізація алгоритму погодженої фільтрації може бути виконана шляхом безпосереднього обчислення виразу дискретної згортки. Розглянемо один із можливих варіантів побудови ЦПФ. Алгоритм дискретної згортки в цьому випадку може бути записаний у такому вигляді

$$S_{\text{вих}}^{(q)}[k] = \frac{\Delta t}{2} \sum_{i=0}^{N-1} A_x^{(n)}[k-i] K^{(p)}[i],$$

де $K^{(p)}[i]$ – послідовності p -розрядних кодів, які є цифровими еквівалентами ґратчастої функції $K[i]$ вибіркових значень з імпульсної перехідної характеристики $K(t)$ погодженого фільтра.

Структурна схема ЦПФ наведена на рис. 4.5. На цьому рисунку показано ФВ – пристрій формування імпульсного відгуку (послідовності p -розрядних чисел $K^{(p)}[i]$). Дискретна згортка здійснюється в два етапи.

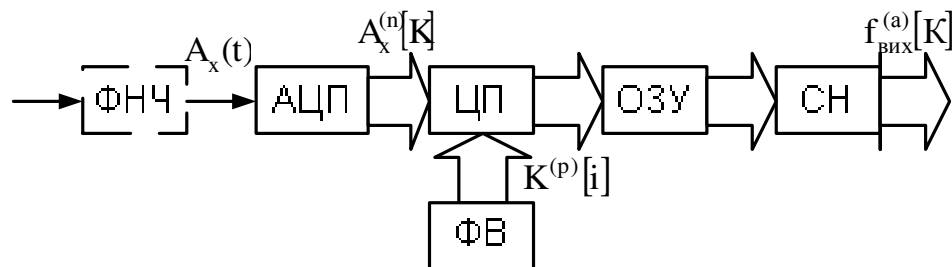


Рисунок 4.5 – Структурна схема цифрового погодженого фільтра

Перший етап роботи пристрою полягає в організації перемножування кожного n -розрядного коду $A_x^{(n)}[k-i]$ на всі N значень p -розрядних чисел $K^{(p)}[i]$ та запису отриманих $N(n+p)$ -розрядних добутків в ОЗП. Останній блок являє собою куб пам'яті (рис. 4.6) розмірами $N \times N \times (n+p)$, у який записуються добутки $A_x^{(n)}[k-i] K^{(p)}[i]$. Другий етап роботи пристрою полягає у зчитуванні інформації з ОЗП та підсумовуванні за допомогою цифрового суматора-накопичувача СН $N(n+p)$ -розрядних кодів для кожного значення імпульсної перехідної характеристики K функції $s_{\text{вих}}^{(q)}[k]$.

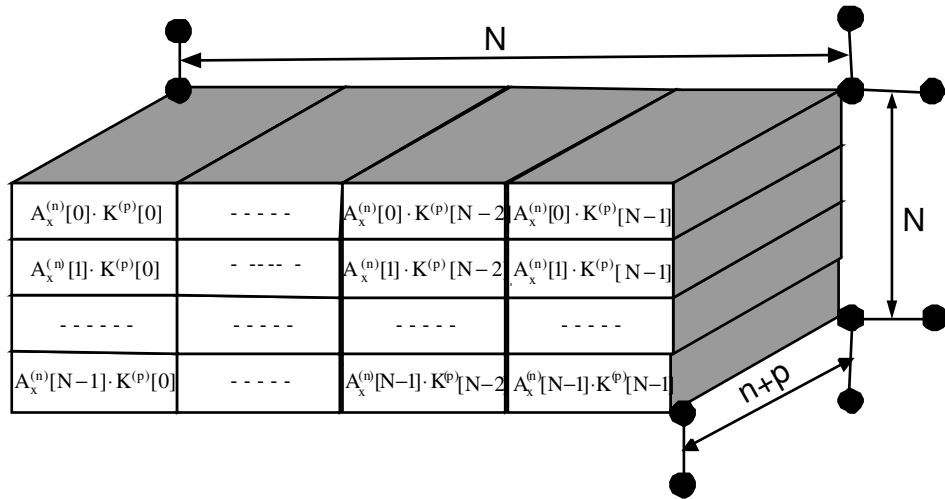


Рисунок 4.6 – Куб пам'яті ЦПФ

Усі елементи куба є додатками, які входять у розглянутий вище вираз згортки, та призначено для одержання $s_{\text{вих}}^{(q)}[k]$. Необхідно організувати діагональний режим зчитування чисел з ОЗП.

При $\kappa = 0$

$$s_{\text{вих}}^{(q)}[0] = A_x^{(n)}[0]K^{(p)}[0] + A_x^{(n)}[-1]K^{(p)}[1] + \dots + A_x^{(n)}[-(N-1)]K^{(p)}[N-1];$$

при $\kappa = 1$

$$s_{\text{вих}}^{(q)}[1] = A_x^{(n)}[1]K^{(p)}[0]K^{(p)}[1] + \dots + A_x^{(n)}[-(N-2)]K^{(p)}[N-1];$$

при $\kappa = N - 2$

$$s_{\text{вих}}^{(q)}[N-2] = A_x^{(n)}[N-2]K^{(p)}[0] + A_x^{(n)}[N-3]K^{(p)}[1] + \dots + A_x^{(n)}[-1]K^{(p)}[N-1];$$

при $\kappa = N - 1$

$$s_{\text{вих}}^{(q)}[N-1] = A_x^{(n)}[N-1]K^{(p)}[0] + A_x^{(n)}[N-2]K^{(p)}[1] + \dots + A_x^{(n)}[0]K^{(p)}[N-1].$$

Дійсно, при $\kappa = 0$ значення кодів

$$A_x^{(n)}[-1], A_x^{(n)}[-2], \dots, A_x^{(n)}[-(N-1)]$$

у випадку існування $A_x(t)$ на інтервалі аналізу дорівнюють нулю. У той же час, якщо $A_x(t)$ являє собою безперервний сигнал на інтервалі $-\infty < t < +\infty$, то значення кодів

$$A_x^{(n)}[-1], A_x^{(n)}[2], \dots, A_x^{(n)}[-(N-1)]$$

відповідають значенням функцій $A_x[-1]$, $A_x[-2]$, ..., $A_x[-(N-1)]$ вибірових значень $A_x[-\Delta t]$, $A_x[-2\Delta t]$, ..., $A_x[-(N-1)\Delta t]$ на інтервалі аналізу, так що

$$A_x^{(n)}[k] = A_x^{(n)}[k + N] \quad (k = 1, 2, 3, \dots, N-1).$$

Підставивши цей вираз у попередній, одержимо діагональний закон зчитування й підсумовування чисел з ОЗП. При цьому для організації

обробки $A_x(t)$ на інтервалі $-\infty < t < +\infty$ необхідно чергувати етапи зчитування та підсумовування з етапами запису чисел. Після k -го діагонального зчитування та підсумовування чисел з ОЗП необхідно зробити запис N кодів наступних добутків у k -й рядок куба пам'яті. Процес повторюється циклічно через N тактів (зчитування, підсумовування, записування). Час одного такту дорівнює інтервалу дискретизації Δt . Такий метод побудови цифрових погоджених фільтрів дозволяє нарощувати структуру пристрою.

4.4 Синтез логічних схем

Синтез комбінаційних та послідовних схем полягає у визначенні таких способів поєднання деяких найпростіших схем, при яких побудований пристрій реалізує поставлену перед ним задачу. Найпростіші схеми називають логічними елементами.

Синтез комбінаційних та послідовних схем на базі логічних елементів виконують таким чином:

- на підставі словесного опису принципу роботи пристрою складають часову діаграму;
- на підставі часової діаграми або словесного опису принципу роботи пристрою створюють таблицю істинності для функції алгебри логіки, яка описує роботи проектованої логічної схеми;
- на підставі таблиці істинності створюють математичну формулу синтезованої схеми у вигляді ЗДНФ або ЗКНФ;
- з використанням законів бульової алгебри проводять аналіз отриманої функції з метою мінімізації операцій;
- будують електричну принципову схему пристрою з логічних елементів, які складають вибраний базис.

Синтез деяких логічних пристроїв у різних базисах розглянемо на прикладах.

Приклад 1. Реалізувати електричну принципову схему пристрою в базисі І, АБО, НІ відповідно до часової діаграми показаної на рис. 4.7.

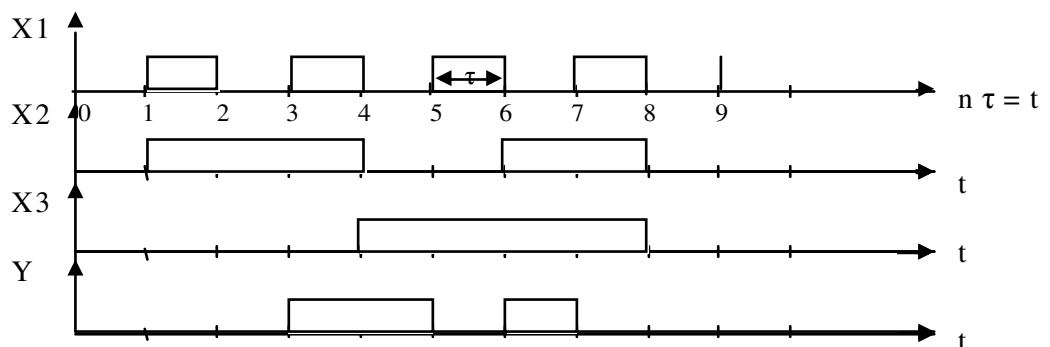


Рисунок 4.7 – Часова діаграма роботи пристрою

На підставі часової діаграми роботи пристрою складаємо таблицю істинності (табл. 4.1).

Таблиця 4.1 – Таблиця істинності

n	X_1	X_2	X_3	Y
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	1
4	0	0	1	1
5	1	0	1	0
6	0	1	1	1
7	1	1	1	0

Записуємо математичну формулу для функції алгебри логіки

$$Y = X_1 X_2 \bar{X}_3 + \bar{X}_1 \bar{X}_2 X_3 + \bar{X}_1 X_2 X_3.$$

За правилом склеювання мінімізуємо цей математичний вираз

$$Y = \bar{X}_1 X_3 + X_1 X_2 \bar{X}_3.$$

Далі утворюємо електричну принципову схему пристрою на логічних елементах І, АБО, НІ (рис. 4.8).

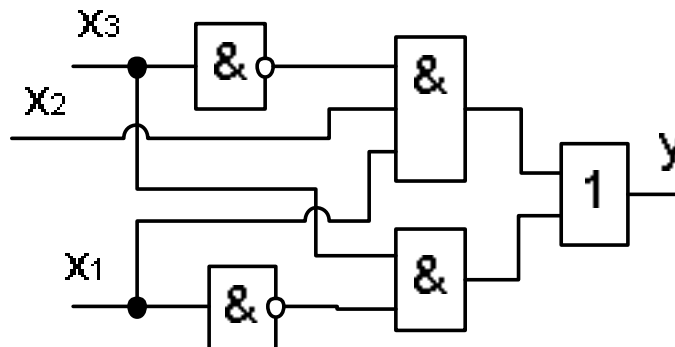


Рисунок 4.8 – Електрична принципова схема пристрою

Приклад 2. Синтезувати у базисі Шеффера електричну принципову схему пристрою, яка реалізує таку логічну функцію

$$Y = \bar{X}_3 X_4 + \bar{X}_1 \bar{X}_2 \bar{X}_4 + X_1 X_2 \bar{X}_4 + \bar{X}_1 X_2 \bar{X}_3 + X_1 \bar{X}_3 \bar{X}_4.$$

Перетворюємо задану формулу рівняння алгебри логіки у ДНФ шляхом введення додаткових комбінацій

$$Y = \bar{X}_3 X_4 (\bar{X}_1 + X_1) (\bar{X}_2 + X_2) + \bar{X}_1 \bar{X}_2 \bar{X}_4 (\bar{X}_3 + X_3) + X_1 X_2 \bar{X}_4 (\bar{X}_3 + X_3) + \bar{X}_1 X_2 \bar{X}_3 (\bar{X}_4 + X_4) + \\ + X_1 \bar{X}_3 \bar{X}_4 (\bar{X}_2 + X_2) = \bar{X}_1 \bar{X}_2 \bar{X}_3 X_4 + \bar{X}_1 X_2 \bar{X}_3 X_4 + X_1 \bar{X}_2 \bar{X}_3 X_4 + X_1 X_2 \bar{X}_3 X_4 + \bar{X}_1 \bar{X}_2 \bar{X}_3 \bar{X}_4 + \\ + \bar{X}_1 \bar{X}_2 X_3 \bar{X}_4 + X_1 X_2 \bar{X}_3 \bar{X}_4 + X_1 X_2 X_3 \bar{X}_4 + \bar{X}_1 X_2 \bar{X}_3 \bar{X}_4 + X_1 \bar{X}_2 \bar{X}_3 \bar{X}_4$$

Мінімізуємо отриманий математичний вираз за допомогою карти Карно (рис. 6.9).

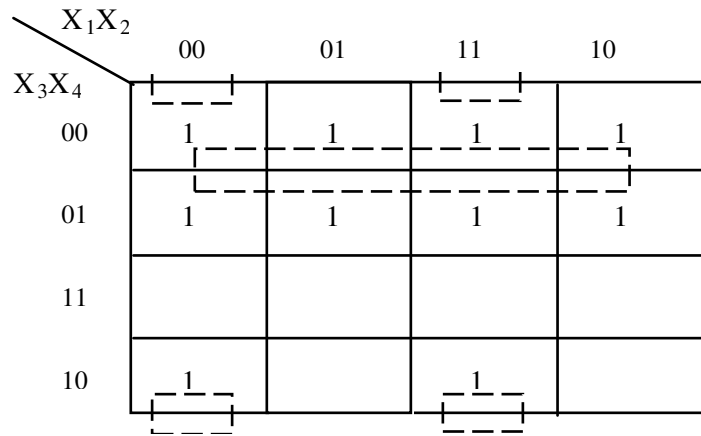


Рисунок 4.9 – Карта Карно

У результаті мінімізації маємо

$$Y = \overline{X_3} + \overline{X_1} \overline{X_2} \overline{X_4} + X_1 X_2 \overline{X_4}.$$

Використовуючи правило де Моргана, маємо

$$Y = \overline{\overline{\overline{X_1} \overline{X_2} \overline{X_4}} \cdot \overline{\overline{X_1} X_2 \overline{X_4}} \cdot \overline{X_3}}.$$

Відповідно до цього виразу будемо електричну принципову схему пристрою (рис. 4.10).

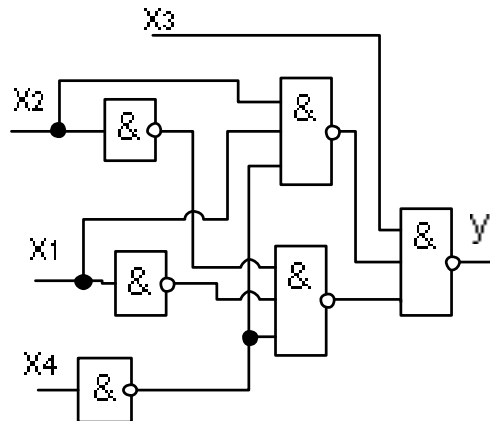


Рисунок 4.10 – Електрична принципова схема

Із карти Карно (див. рис. 4.9) може бути знайдена мінімальна КНФ даної функції алгебри логіки

$$Y = (\overline{X_3} + \overline{X_4})(X_3 + \overline{X_2} + \overline{X_3})(\overline{X_1} + X_2 + \overline{X_3}).$$

Використовуючи правило де Моргана, маємо

$$Y = \overline{\overline{\overline{X_3} X_4} \cdot \overline{\overline{X_1} X_2 X_3} \cdot \overline{X_1 \overline{X_2} X_3}}.$$

Відповідно до цього виразу маємо схему (рис. 4.11).

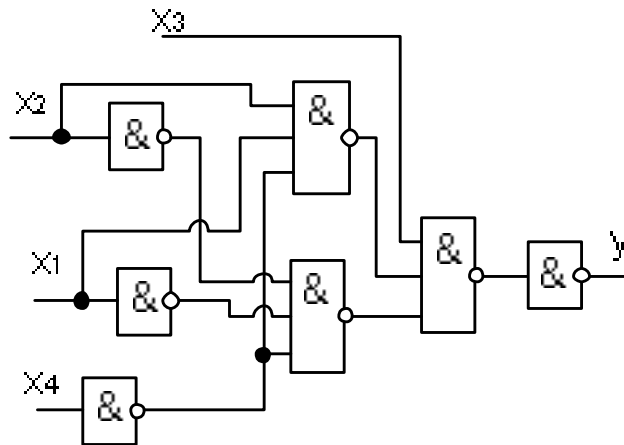


Рисунок 4.11 – Електрична принципова схема

Задану функцію алгебри логіки можна також реалізувати в базисі Пірса, тобто на логічних елементах АБО–НІ. Застосовуючи до мінімальної КНФ правило де Моргана, маємо

$$Y = \overline{\overline{X_3 + X_4 + X_1 + X_2 + X_3 + X_1 + X_2 + X_3}}$$

Відповідно до цього виразу утворюємо електричну принципову схему пристрою (рис. 4.12).

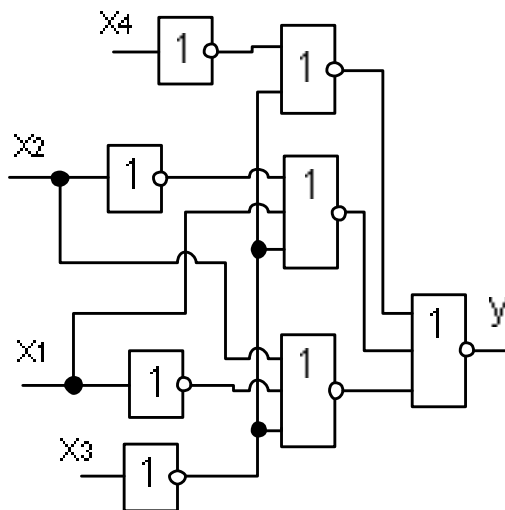


Рисунок 4.12 – Електрична принципова схема

Приклад 3. Спроекувати систему управління промисловим роботом, яка працює згідно з такою часовою діаграмою (рис. 4.13). Згідно з наведеною часовою діаграмою можливі два варіанти побудови системи управління. Розглянемо кожний з варіантів.

Варіант 1. Технічну реалізацію системи управління промисловим роботом можна виконати на базі лічильника імпульсів з дешифратором. Кількість тригерів n лічильника визначається з аналізу часової діаграми за

допомогою співвідношення $[n] = \log_2 N$, де $[n]$ – ціле число; N – кількість імпульсів генератора, які складають цикли роботи промислового робота.

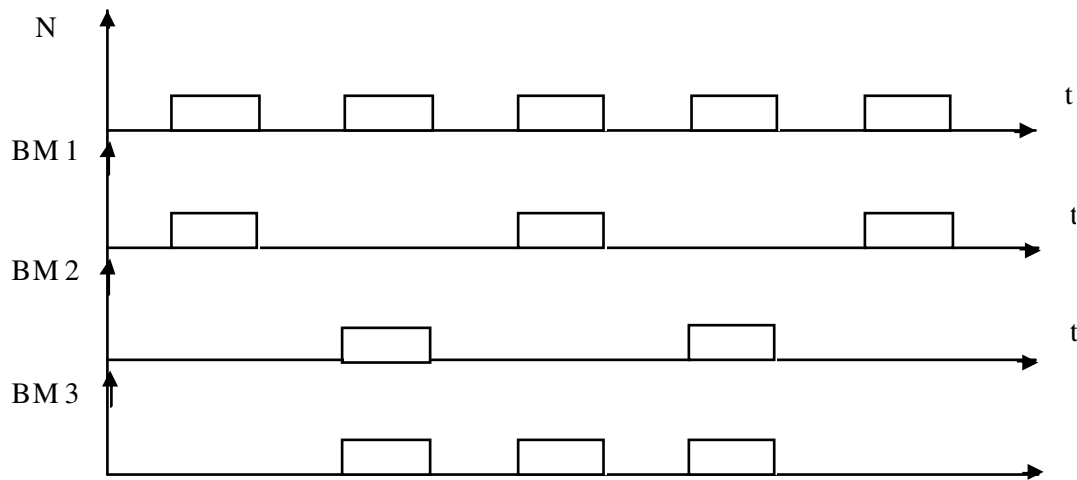


Рисунок 4.13 – Часова діаграма роботи пристрою

Так для розглядуваного прикладу маємо $[n] = \log_2 5 = 3$. Спроекуємо лічильник імпульсів на базі JK -тригерів. На підставі аналізу часової діаграми промислового робота складаємо таблицю станів та переходів (табл. 4.2).

Таблиця 4.2 – Таблиця станів та переходів

N	Q_3	Q_2	Q_1	J_3	J_2	J_1
0	0	0	0	0	0	1
1	0	0	1	0	1	0
2	0	1	0	0	1	1
3	0	1	1	1	0	0
4	1	0	0	1	0	1
5	1	0	1	∅	∅	∅
6	1	1	0	∅	∅	∅
7	1	1	1	∅	∅	∅

За таблицю 4.2 складаємо карти Карно для J -входів кожного тригеру (див. рис. 4.14). За допомогою карт Карно отримаємо мінімізовані вирази рівнянь алгебри логіки роботи тригерів у складі підсумовуючого лічильника імпульсів. За допомогою правила де Моргана виконуємо перетворення виразів алгебри логіки лічильника. Відповідно до отриманих математичних виразів складаємо схеми елементів управління тригерами, а потім будуємо електричну принципову схему лічильника (рис. 4.15).

Із часової діаграми роботи промислового робота будуємо таблицю істинності дешифратора (табл. 4.3). За таблицею істинності складаємо карти Карно для кожного виходу дешифратора (рис. 4.16). За допомогою карт Карно отримаємо мінімізовані вирази рівнянь алгебри логіки роботи дешифраторів. Відповідно до отриманих математичних виразів складаємо електричну принципову схему пристрою (рис. 4.17).

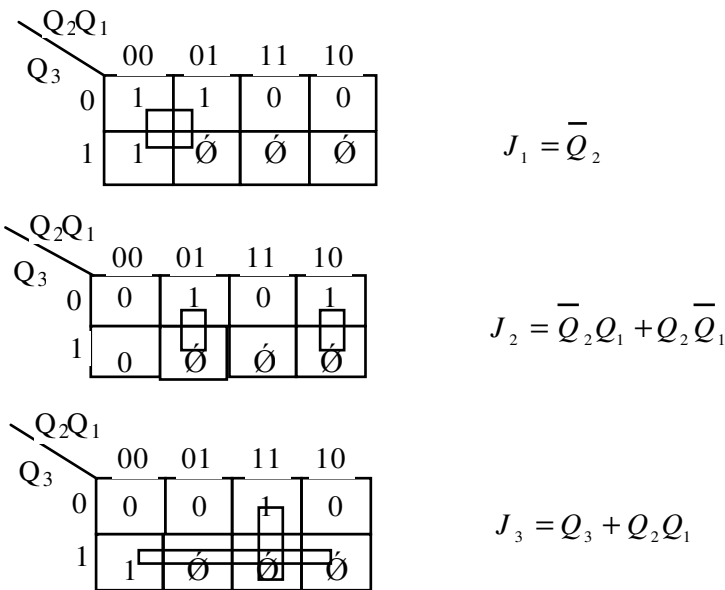


Рисунок 4.14 – Карти Карно для трьох тригерів

Вирази алгебри логіки переключення тригерів лічильника

$$J_1 = \bar{Q}_2, \quad J_2 = \bar{Q}_1Q_2 + Q_1\bar{Q}_2, \quad J_3 = \bar{Q}_2Q_3 + Q_2Q_1.$$

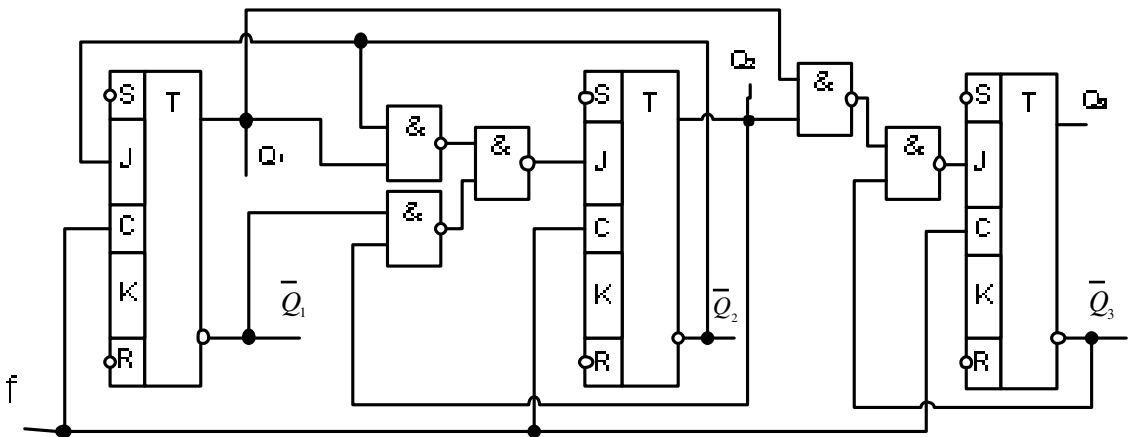


Рисунок 4.15 – Принципова схема лічильника

Таблиця 4.3 – Таблиця істинності дешифратора

N	Q_3	Q_2	Q_1	BM_3	BM_2	BM_1
0	0	0	0	0	0	1
1	0	0	1	1	1	0
2	0	1	0	1	0	1
3	0	1	1	1	1	0
4	1	0	0	0	0	1
5	∅	∅	∅	∅	∅	∅
6	∅	∅	∅	∅	∅	∅
7	∅	∅	∅	∅	∅	∅

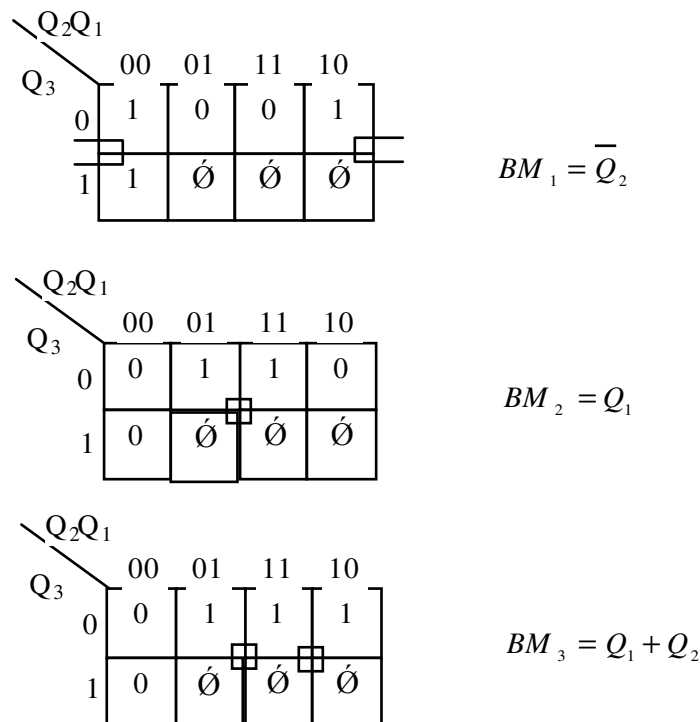


Рисунок 4.16 – Карти Карно для трьох виходів дешифратора

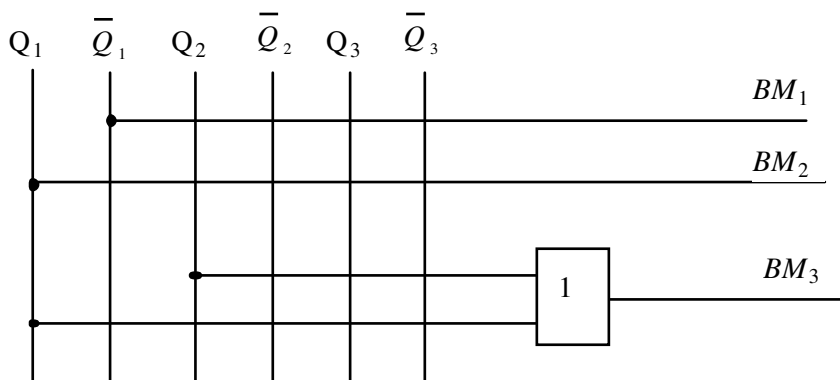


Рисунок 4.17 – Електрична принципова схема дешифратора

Варіант 2 Технічну реалізацію системи управління промисловим роботом можна виконати на базі кільцевого регістру зсуву та шифратора. Розрядність регістру зсуву N_p залежить від кількості імпульсів генератора, які складають цикл роботи промислового робота $N_p = N_c$.

При цьому вважається, що перед початком роботи системи в регістр записується 1000...0 або 0111...1. З приходом кожного імпульсу генератора інформація у регістрі зсувається на один розряд вправо. Для прикладу, що розглядається, з урахуванням часової діаграми маємо $N_p = 5$.

Спроекуємо кільцевий регістр зсуву на *JK*-тригерах. Для скорочення обсягу перетворень спроекуємо регістр зсуву на трьох тригерах. Складаємо таблицю станів та переходів тригерів (табл. 4.4).

Таблиця 4.4 – Таблиця станів та переходів тригерів

Q_n			Q_{n+1}		
3	2	1	3	2	1
0	0	1	0	1	0
0	1	0	1	0	0
1	0	0	0	0	1

На підставі таблиці переходів складаємо карти Карно для J -входів кожного тригера (рис. 4.18).

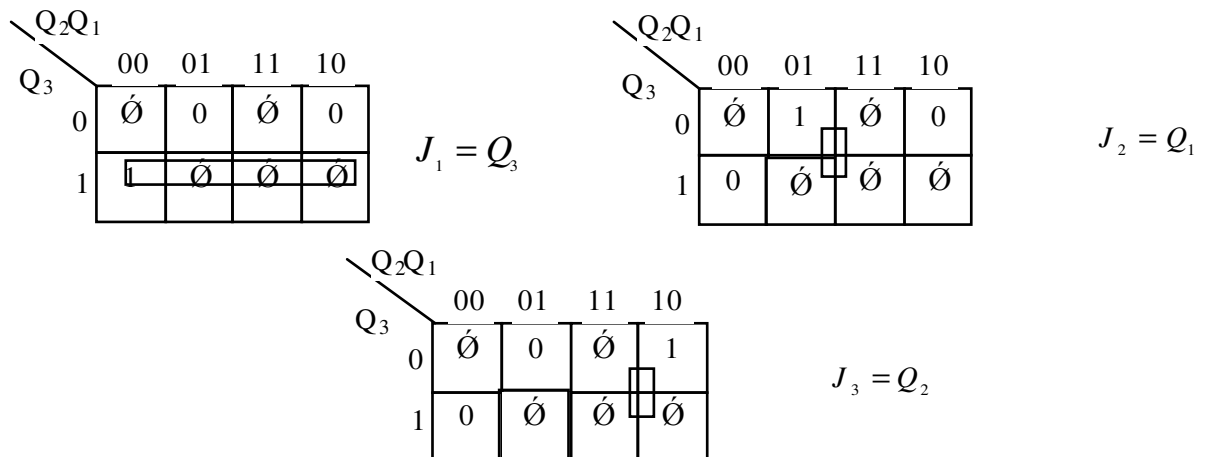


Рисунок 4.18 – Карти Карно для трьох тригерів

На підставі отриманих рівнянь маємо таке

$$J_1 = Q_5, \quad J_2 = Q_1, \quad J_3 = Q_2, \quad J_4 = Q_3, \quad J_5 = Q_4.$$

Відповідно до отриманих рівнянь проектуємо електричну принципову схему кільцевого регістру зсуву (рис. 4.19).

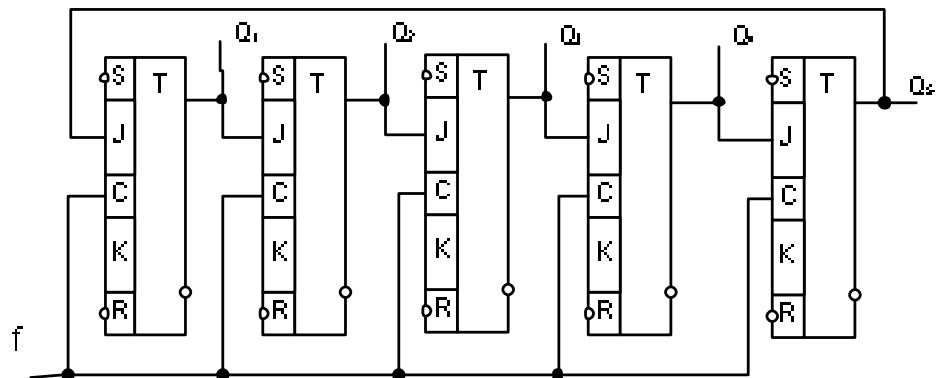


Рисунок 4.19 – Електрична принципова схема регістру зсуву

На підставі часової діаграми роботи промислового роботу складаємо таблицю істинності для шифратора (табл. 4.5).

Таблиця 4.5 – Таблиця істинності шифратора

Q_5	Q_4	Q_3	Q_2	Q_1	BM_3	BM_2	BM_1
0	0	0	0	1	0	0	1
0	0	0	1	0	1	1	0
0	0	1	0	0	1	0	1
0	1	0	0	0	1	1	0
1	0	0	0	0	0	0	1

На підставі таблиці істинності маємо такі рівняння алгебри логіки

$$BM_1 = Q_1 + Q_3 + Q_5, \quad BM_2 = Q_2 + Q_4, \quad BM_3 = Q_2 + Q_3 + Q_4.$$

Згідно з отриманими виразами будуюмо схему шифратора (рис. 4.20).

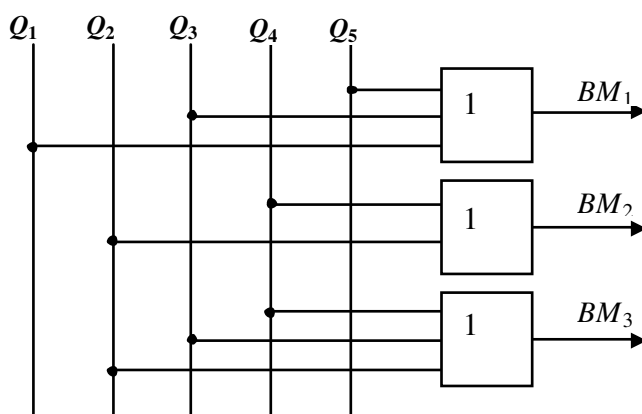


Рисунок 4.20 – Електрична принципова схема шифратора

Приклад 4. Розробити функціональну схему автоматичного пристрою з пам'яттю на D -тригерах, опис якого подано блок-схемою алгоритму (див. рис. 4.12).

Як видно зі схеми, пристрій повинен мати п'ять входів ($X_1 \dots X_5$) та три виходи (y_1, y_2, y_3). Він може знаходитися в одному із п'яти станів $a_1 \dots a_5$. Для фіксації та збереження станів необхідно визначити кількість тригерів за формулою

$$m = \log_2 N,$$

де N – число станів.

У нашому прикладі $N = 5$, тоді $m = \log_2 5 = 2,52$. Приймаємо $m = 3$.

Присвоюємо кожному стану пристрою 3-розрядний код

- a_1 – 000,
- a_2 – 001,
- a_3 – 010,
- a_4 – 011,
- a_5 – 100.

Визначені стани подано в унітарному коді довжиною $l = 5$, для чого використаємо дешифратор. Щоб запобігти появі гонок використаємо імпульсну синхронізацію D -тригерів. Для відображення станів пристрою будуюмо граф переходів (рис. 4.22).

Вихідні сигнали тригерів отримано на підставі аналізу їхніх станів та переходів

$$Y_1 = a_1 X_1 X_3 + a_1 \bar{X}_1 \bar{X}_2 + a_2 \bar{X}_3;$$

$$Y_2 = a_2 X_3 + a_3 \bar{X}_3;$$

$$Y_3 = a_1 X_1 X_3 + a_2 \bar{X}_3 + a_4 X_4 + a_5 X_5.$$

Сигнали збудження тригерів

$$D_1 = a_2 X_3;$$

$$D_2 = a_1 \bar{X}_1 X_2 + a_1 \bar{X}_1 \bar{X}_2 + a_3 \bar{X}_3 + a_3 X_3 + a_4 X_4 + a_5 \bar{X}_5;$$

$$D_3 = a_1 X_1 X_3 + a_2 \bar{X}_3 + a_3 X_3 + a_4 X_4 + a_5 \bar{X}_5.$$

Таблиця 4.6 – Таблиця переходів тригерів

Поточний стан		Наступний стан		Вхідні сигнали	Вихідні сигнали	Сигнали збудження тригерів
Позначення	Код	Позначення	Код			
a ₁	000	a ₁	000	X ₁ \bar{X}_3	-	-
		a ₂	001	X ₁ X ₃	Y ₁ \bar{Y}_3	D ₃
		a ₃	010	$\bar{X}_1 X_2$	-	D ₂
				$\bar{X}_1 \bar{X}_2$	Y ₁	D ₃
a ₂	001	a ₂	001	\bar{X}_3	Y ₁ \bar{Y}_3	D ₂
		a ₅	100	X ₃	Y ₂	D ₁
a ₃	010	a ₃	010	\bar{X}_3	Y ₂	D ₂
		a ₄	011	X ₃	-	D ₂ D ₃
a ₄	011	a ₁	000	\bar{X}_4	-	-
		a ₄	011	X ₄	Y ₃	D ₂ D ₃
a ₅	100	a ₁	000	X ₅	-	-
		a ₄	011	\bar{X}_5	Y ₃	D ₂ D ₃

Мінімізуємо функцію сигналу збудження D₂

$$D_2 = a_1 \bar{X}_1 + a_3 + a_4 X_4 + a_5 \bar{X}_5.$$

Інші функції записані в мінімальній формі ДНФ. Отримані функції приведемо до базису І-НІ

$$Y_1 = a_1 X_1 X_3 + a_1 \bar{X}_1 \bar{X}_2 + a_2 \bar{X}_3 = a_1 X_1 X_3 + a_1 \bar{X}_1 \bar{X}_2 + a_2 \bar{X}_3,$$

$$Y_2 = a_2 X_3 + a_3 \bar{X}_3 = a_2 X_3 + a_3 \bar{X}_3,$$

$$Y_3 = a_1 X_1 X_3 + a_2 \bar{X}_3 + a_4 X_4 + a_5 \bar{X}_5 = a_1 X_1 X_3 + a_2 \bar{X}_3 + a_4 X_4 + a_5 \bar{X}_5.$$

Використовуючи для цього D-тригери, дешифратор та логічні елементи ТА-НІ будуюмо електричну принципову схему автоматичного пристрою (див. рис. 4.23).

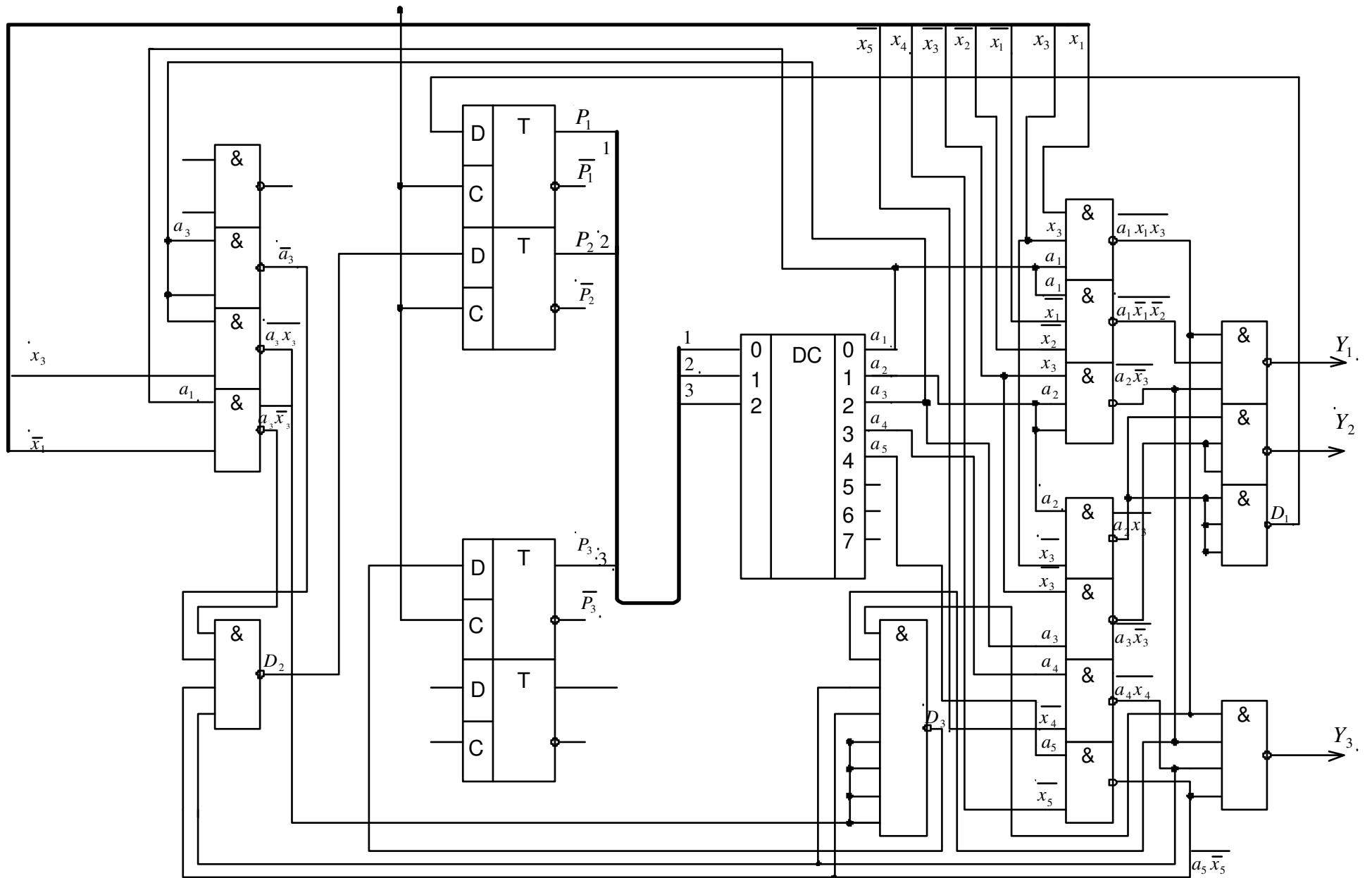


Рисунок 4.23 – Функціональна схема пристрою

4.5 Синтез комбінаційних схем на базі комутаторів

У складі різних серій мікросхем, які застосовуються у пристроях систем керування електричним транспортом, є елементи середнього ступеня інтеграції – комутатори (мультиплексори). Наприклад, К155КП1, К155КП5, К155КП7 та інші. Комутатор являє собою багато входний логічний елемент з одним виходом. Входи комутатора поділяють на інформаційні та управляючі. При подачі на управляючі входи відповідного сигналу у вигляді двійкового коду до виходу комутатора підключається один з його інформаційних входів. Промисловістю випускаються комутатори, які здійснюють вибір одного з 4-х, з 8-ми або з 16-ти інформаційних сигналів. Умовне позначення таких комутаторів наведено на рис. 4.24.

Рівняння комутаторів 1 з 4, 1 з 8 та 1 з 16 можна записати наступним чином

$$F_{1-4} = \overline{\overline{A_1 A_0}} B_0 + \overline{A_1 A_0} B_1 + A_1 \overline{A_0} B_2 + A_1 A_0 B_3. \quad (4.1)$$

$$F_{1-8} = \overline{\overline{\overline{A_2 A_1 A_0}}} B_0 + \overline{\overline{A_2 A_1 A_0}} B_1 + \overline{A_2 A_1 A_0} B_2 + \overline{A_2 A_1 A_0} B_3 + \overline{A_2 A_1 A_0} B_4 + A_2 \overline{\overline{A_1 A_0}} B_5 + A_2 \overline{A_1 A_0} B_6 + A_2 A_1 A_0 B_7. \quad (4.2)$$

$$F_{1-16} = \overline{\overline{\overline{\overline{A_3 A_2 A_1 A_0}}} B_0 + \overline{\overline{\overline{A_3 A_2 A_1 A_0}}} B_1 + \overline{\overline{A_3 A_2 A_1 A_0}} B_2 + \overline{A_3 A_2 A_1 A_0} B_3 + \overline{A_3 A_2 A_1 A_0} B_4 + \overline{A_3 A_2 A_1 A_0} B_5 + \overline{A_3 A_2 A_1 A_0} B_6 + \overline{A_3 A_2 A_1 A_0} B_7 + A_3 \overline{\overline{\overline{A_2 A_1 A_0}}} B_8 + A_3 \overline{\overline{A_2 A_1 A_0}} B_9 + A_3 \overline{A_2 A_1 A_0} B_{10} + A_3 \overline{A_2 A_1 A_0} B_{11} + A_3 A_2 \overline{\overline{A_1 A_0}} B_{12} + A_3 A_2 \overline{A_1 A_0} B_{13} + A_3 A_2 A_1 \overline{A_0} B_{14} + A_3 A_2 A_1 A_0 B_{15}. \quad (4.3)$$

У виразах (6.1) - (6.3) через B_i та A_j позначено сигнали, які подаються на інформаційні та управляючі входи комутаторів відповідно. Нехай ми маємо довільні функції алгебри логіки 3-х, 4-х та 5-ти змінних

$$F_3(x_3, x_2, x_1), \quad F_4(x_4, x_3, x_2, x_1), \quad F_5(x_5, x_4, x_3, x_2, x_1). \quad (4.4)$$

Позначимо змінні x_5, x_4, x_3, x_2 відповідно через A_3, A_2, A_1, A_0 , а змінну x_1 – через B .

$$F_3(A_1, A_0, B), \quad F_4(A_2, A_1, A_0, B), \quad F_5(A_3, A_2, A_1, A_0, B). \quad (4.5)$$

Використовуючи метод функціональної декомпозиції, подамо функції (4.5) у вигляді таких розкладів

$$F_3(A_1, A_0, B) = F_3(0, 0, B) \overline{\overline{A_1 A_0}} + F_3(0, 1, B) \overline{A_1 A_0} + F_3(1, 0, B) A_1 \overline{A_0} + F_3(1, 1, B) A_1 A_0. \quad (4.6)$$

$$F_4(A_2, A_1, A_0, B) = F_4(0, 0, 0, B) \overline{\overline{\overline{A_2 A_1 A_0}}} + F_4(0, 0, 1, B) \overline{\overline{A_2 A_1 A_0}} + F_4(0, 1, 0, B) \overline{A_2 A_1 A_0} + F_4(0, 1, 1, B) \overline{A_2 A_1 A_0} + F_4(1, 0, 0, B) A_2 \overline{\overline{A_1 A_0}} + F_4(1, 0, 1, B) A_2 \overline{A_1 A_0} + F_4(1, 1, 0, B) A_2 A_1 \overline{A_0} + F_4(1, 1, 1, B) A_2 A_1 A_0. \quad (4.7)$$

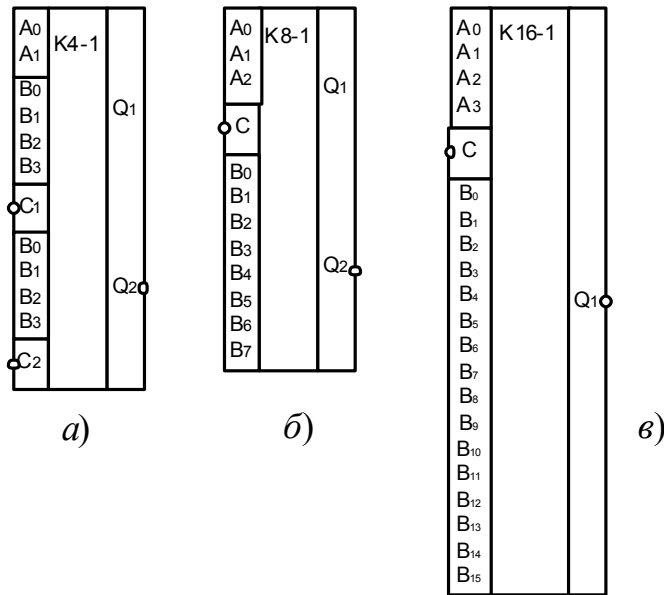


Рисунок 4.24 – Типи комутаторів серії K155:
a – ІМС K155KП2, *б* – ІМС 155KП7, *в* – ІМС K155KП11

$$\begin{aligned}
 F_5(A_3, A_2, A_1, A_0, B) = & F_5(0, 0, 0, 0, B) \overline{A_3} \overline{A_2} \overline{A_1} \overline{A_0} + F_5(0, 0, 0, 1, B) \overline{A_3} \overline{A_2} \overline{A_1} A_0 + \\
 & F_5(0, 0, 1, 0, B) \overline{A_3} \overline{A_2} A_1 \overline{A_0} + F_5(0, 0, 1, 1, B) \overline{A_3} \overline{A_2} A_1 A_0 + F_5(0, 1, 0, 0, B) \overline{A_3} A_2 \overline{A_1} \overline{A_0} + \\
 & F_5(0, 1, 0, 1, B) \overline{A_3} A_2 \overline{A_1} A_0 + F_5(0, 1, 1, 0, B) \overline{A_3} A_2 A_1 \overline{A_0} + F_5(0, 1, 1, 1, B) \overline{A_3} A_2 A_1 A_0 + \\
 & F_5(1, 0, 0, 0, B) A_3 \overline{A_2} \overline{A_1} \overline{A_0} + F_5(1, 0, 0, 1, B) A_3 \overline{A_2} \overline{A_1} A_0 + F_5(1, 0, 1, 0, B) A_3 \overline{A_2} A_1 \overline{A_0} + \\
 & F_5(1, 0, 1, 1, B) A_3 \overline{A_2} A_1 A_0 + F_5(1, 1, 0, 0, B) A_3 A_2 \overline{A_1} \overline{A_0} + F_5(1, 1, 0, 1, B) A_3 A_2 \overline{A_1} A_0 + \\
 & F_5(1, 1, 1, 0, B) A_3 A_2 A_1 \overline{A_0} + F_5(1, 1, 1, 1, B) A_3 A_2 A_1 A_0.
 \end{aligned} \tag{4.8}$$

Розглядаючи вирази (4.6) – (4.8), легко переконатися в тому, що всі значення функцій F_3, F_4, F_5 , які знаходяться в правій частині цих виразів, можуть приймати значення тільки з множини $\{0, 1, B \text{ або } \overline{B}\}$. Ці вирази також описують будь-яку функцію алгебри логіки відповідно 3-х, 4-х та 5-ти змінних.

Попарне зіставлення виразів (4.1) та (4.6), (4.2) та (4.7), (4.3) та (4.8) показує, що коли на управляючі входи комутаторів подавати такі незалежні змінні $A_1 A_0, A_2 A_1 A_0, A_3 A_2 A_1 A_0$, а на відповідні інформаційні входи – величини з множини $\{0, 1, B \text{ або } \overline{B}\}$ за таблицею істинності заданої функції, то можна записати такі рівності

$$F_{1-4} = F_3, \quad F_{1-8} = F_4, \quad F_{1-16} = F_5. \tag{4.9}$$

Ці вирази свідчать про те, що комутатори 1 з 4, 1 з 8 та 1 з 16 є універсальними логічними елементами, що реалізують будь-яку функцію алгебри логіки відповідно 3-х, 4-х та 5-ти змінних.

Приклад 1. Реалізувати функцію

$$F_4 = x_1 x_2 x_3 + \overline{x_1} x_2 x_3 + x_2 x_3 x_4 + \overline{x_2} x_3 x_4 + x_1 x_3 x_4 + \overline{x_1} x_2 x_4$$

за допомогою комутатора 1 з 8.

Аналіз таблиці істинності цієї функції (табл. 4.7) вказує, що вона буде реалізована за допомогою комутатора 1 з 8 у тому випадку, коли на управляючі входи $A_2A_1A_0$ подати змінні $x_1x_2x_3$, а на відповідні інформаційні входи - сигнали: $B_0 - 0, B_1 - \bar{x}_4, B_2 - 1, B_3 - 0, B_4 - x_4, B_5 - 1, B_6 - 1, B_7 - \bar{x}_4$. Сигнал логічної 1 на вході ІМС серії К155 реалізується його обривом. Для завдання логічного 0 підключаємо відповідний вхід до нульової шини джерела живлення. Приклад побудови пристрою показано на рис. 4.25.

Таблиця 4.7 – Таблиця істинності комутатора

x_1	x_2	x_3	x_4	F_4	Вихід, що підключається до входу
0	0	0	0	0	B_0
0	0	0	1	0	B_0
0	0	1	0	$1(\bar{x}_4)$	B_1
0	0	1	1	$0(\bar{x}_4)$	B_1
0	1	0	0	1	B_2
0	1	0	1	1	B_2
0	1	1	0	0	B_3
0	1	1	1	0	B_3
1	0	0	0	$0(x_4)$	B_4
1	0	0	1	$1(x_4)$	B_4
1	0	1	0	1	B_5
1	0	1	1	1	B_5
1	1	0	0	1	B_6
1	1	0	1	1	B_6
1	1	1	0	$1(\bar{x}_4)$	B_7
1	1	1	1	$0(\bar{x}_4)$	B_7

На підставі викладеного можна сформулювати правило.

Правило 1. Для реалізації функції алгебри логіки 3-х, 4-х або 5-ти змінних відповідно на комутаторі 1 з 4, 1 з 8, 1 з 16 необхідно:

- 1) скласти таблицю істинності функції алгебри логіки;
- 2) сигнали, відповідні $i-1$ змінним ($i = 3, 4$ або 5), подати на управляючі входи комутатора (i – індекс молодшої змінної);
- 3) на інформаційні входи комутатора відповідно з таблицею істинності функції алгебри логіки подати сигнали з множини $\{0, 1, x_i, \bar{x}_i\}$.

Наявність у комутатора стробуючого входу, який позначено літерою С, дозволяє значно поширити їх логічні можливості. Для ілюстрації цього положення перетворимо вираз (4.2) таким чином

$$F_{1-8} = (B_0\bar{A}_1\bar{A}_0 + B_2\bar{A}_1A_0 + B_4A_1\bar{A}_0 + B_6A_1A_0)\bar{A}_2 + (B_1\bar{A}_1\bar{A}_0 + B_3\bar{A}_1A_0 + B_5A_1\bar{A}_0 + B_7A_1A_0)A_2. \quad (4.10)$$

З цього можна побачити, що вирази у дужках є рівняннями комутатора 1 з 4, які еквівалентні виразу (4.1). Отже логічний вираз (4.10) можна записати у такому вигляді

$$F_{1-8} = F_{1-4}^1 \overline{A_2} + F_{1-4}^2 A_2. \quad (4.11)$$

Аналогічно для комутатора 1 з 16 одержимо

$$F_{1-16} = F_{1-8}^1 \overline{A_3} + F_{1-8}^2 A_3. \quad (4.12)$$

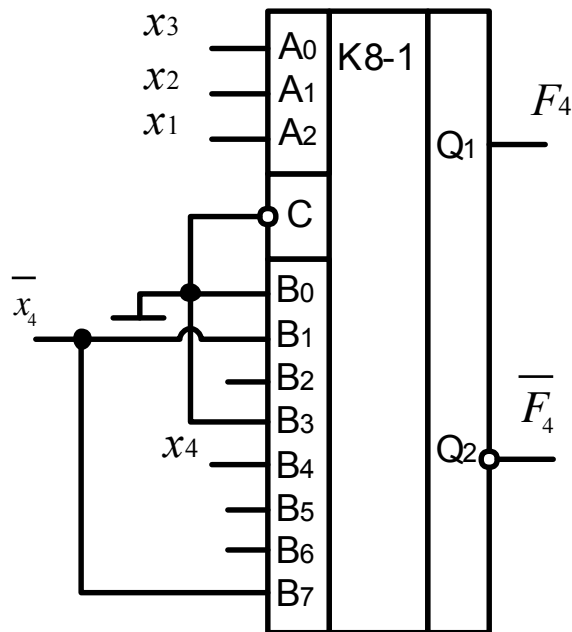


Рисунок 4.25 – Реалізація ФАЛ 4-х змінних на комутаторі 1 з 8 (приклад 1)

Таким чином, об'єднуючи рівняння комутаторів 1 з 16, можна отримати рівняння гранично універсального логічного елемента для функцій 6-ти змінних

$$F_{1-32} = F_{1-16}^1 \overline{A_4} + F_{1-16}^2 A_4. \quad (4.13)$$

На підставі викладеного сформулюємо друге правило.

Правило 2. Для реалізації функції алгебри логіки 4-х, 5-ти або 6-ти змінних відповідно на двох комутаторах 1 з 4, 1 з 8, 1 з 16, необхідно:

- 1) пронумерувати змінні від 1 до i , де $i = 4, 5$ або 6 (i – індекс молодшої змінної);
- 2) скласти таблицю істинності заданої функції;
- 3) сигнали, відповідні $i-2$ змінним, подати рівнобіжно на управляючі входи двох комутаторів;
- 4) сигнал, який відповідає змінній з індексом $i-1$, треба подати на стробуючий вхід першого комутатора та через інвертор – на стробуючий вхід другого комутатора;
- 5) на інформаційні входи комутаторів треба подати, відповідно із заданою таблицею істинності функції алгебри логіки, сигнали із такої множини $\{0, 1, x_i, \overline{x_i}\}$;

б) прямі виходи комутаторів об'єднати елементом АБО.

Приклад 2. Реалізувати функцію алгебри логіки, яку задано таким рівнянням

$$F = x_1x_2x_3 + x_1x_2x_4 + x_1x_2x_5 + x_1x_3x_4 + x_1x_3x_5 + x_1x_4x_5 + x_2x_3x_4 + x_2x_3x_5 + x_2x_4x_5 + x_3x_4x_5.$$

Функцію алгебри логіки побудувати на комутаторі 1 з 16 та окремо на двох комутаторах 1 з 8. За правилом 1 одержана схема, яка зображена на рис. 4.26, де на управляючі входи комутатора К16-1 подані старші змінні (x_1, x_2, x_3, x_4) заданої ФАЛ, а на інформаційні – сигнали з множини $\{0, 1, x_5, \overline{x_5}\}$.

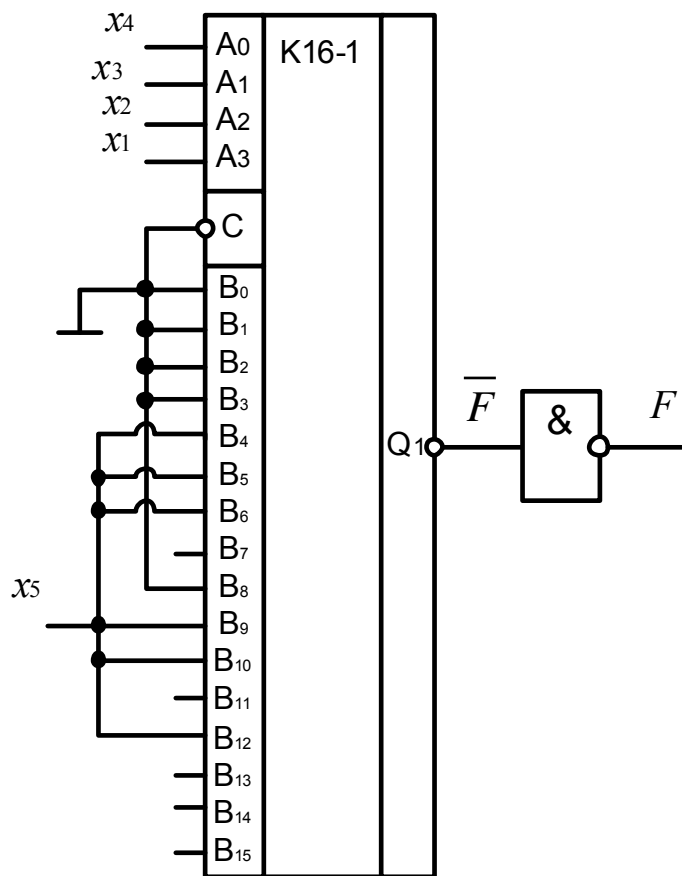


Рисунок 4.26 – Реалізація ФАЛ 5-х змінних на комутаторі 1 з 16 (приклад 2)

Для реалізації цієї ж ФАЛ на двох комутаторах К8-1 необхідно скористатися правилом 2. Згідно із побудованою таблицею істинності (див. табл. 4.8) старші змінні (x_1, x_2, x_3) подати рівнобіжно на управляючі входи двох комутаторів К8-1, сигнал x_4 поступає на стробуючий вхід С комутатора D1, та через інвертор D2 – на вхід С комутатора D3. Таким чином забезпечується почергова робота обох комутаторів. На інформаційні входи відповідно з табл. 4.8 подаються сигнали з множини $\{0, 1, x_5, \overline{x_5}\}$. Щоб отримати сумарне значення функції F , треба виходи комутаторів об'єднати елементом АБО (D4). Відповідна схема одержана на рис. 4.27.

Таблиця 4.8 – Таблиця істинності комутаторів

Позначення	x_4x_5	$x_1x_2x_3$							
		000	001	010	011	100	101	110	111
D_1	00	0	0	0	$0(x_5)$	0	$0(x_5)$	$0(x_5)$	1
	01	0	0	0	$1(x_5)$	0	$1(x_5)$	$1(x_5)$	1
D_3	10	0	$0(x_5)$	$0(x_5)$	1	$0(x_5)$	1	1	1
	11	0	$1(x_5)$	$1(x_5)$	1	$1(x_5)$	1	1	1
Входи комутатора		B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7

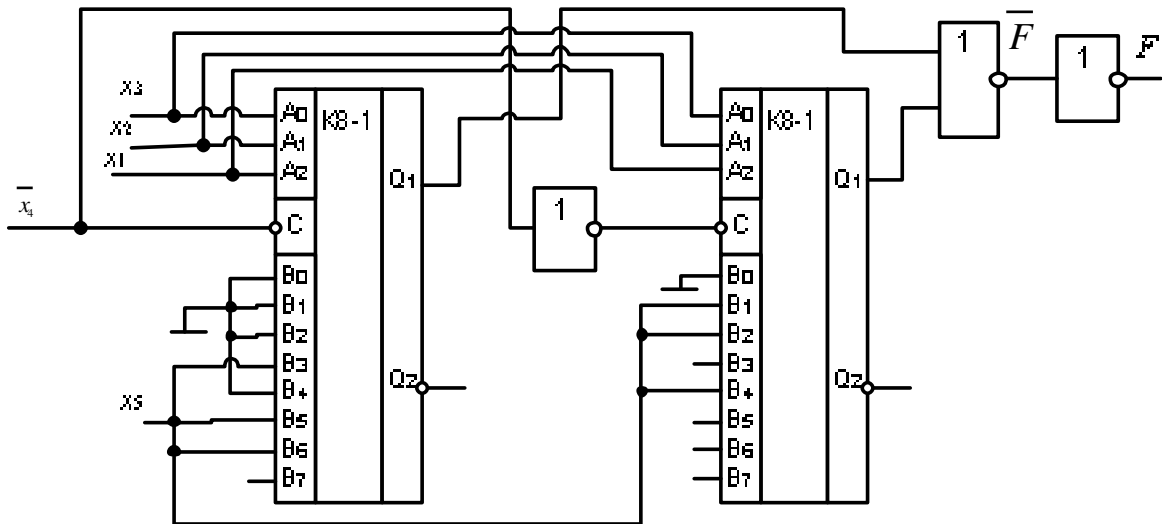


Рисунок 4.27 – Реалізація ФАЛ 5-х незалежних змінних за допомогою двох комутаторів 1 з 8 (приклад 2)

Контрольні запитання

1. Сформулюйте методи захисту інформації в МП системах.
2. Приведіть структурну схему багатofункціонального генератора на базі МП.
3. Приведіть структурну схему корелятора.
4. Розкрийте принцип дії ЦПФ.
5. З яких операцій складається синтез систем програмного управління електричним транспортом?
6. У чому полягає синтез систем управління електричним транспортом із застосуванням сигнальних графів?
7. Які особливості синтезу схем на базі комутаторів?