

О.Ф.Єнікєєв, О.О.Сердюк



ДДМА

ДДМА

ДДМА

ДДМА

ДДМА

ДДМА

ДДМА

ДДМА

ДДМА

ДДМА

ДДМА

2004

**ПРОЕКТУВАННЯ
ЦИФРОВИХ ПРИСТРОЇВ
НА ІНТЕГРАЛЬНИХ
МІКРОСХЕМАХ**

Міністерство освіти і науки України
Донбаська державна машинобудівна академія

621.382(075)

Є 63

472

О.Ф. Єнікєєв, О.О. Сердюк

**ПРОЕКТУВАННЯ ЦИФРОВИХ ПРИСТРОЇВ
НА ІНТЕГРАЛЬНИХ МІКРОСХЕМАХ**

Рекомендовано

Міністерством освіти і науки України
як навчальний посібник
для студентів денної заочної форм навчання
спеціальності 7.092501

НТБ ВНТУ

421198
621.382(075) Є 63 2004
Єнікєєв О.Ф. Проектування цифрових пристроїв

Краматорськ 2004

[631.382 + 681.32.00.02] (09)

ББК 32.844.1
УДК 621.382.3
Є-63

Рецензенти:

Дербунович Л.В., доктор технічних наук, професор кафедри „Автоматика і управління в технічних системах” Національного технічного університету „ХПІ”

Черняк Ю.В., кандидат технічних наук, доцент, завідувач кафедри „Управління рухом поїздів та рухомий склад” Донецького інституту залізничного транспорту

Габузов Г.Г., доктор технічних наук, професор кафедри “Автоматизація виробничих процесів”, ДДМА

Гриф надано Міністерством освіти і науки України
Лист № 14/18.2-1163 від 02.06.2004

О.Ф. Єнікєєв, О.О. Сердюк

Є-63 Проектування цифрових пристроїв на інтегральних мікросхемах:
Посібник/ О.Ф. Єнікєєв, О.О. Сердюк. - Краматорськ: ДДМА, 2004. – 108 с.

ISBN 966-7851-46-X

Викладаються основні принципи побудови цифрових пристроїв на інтегральних мікросхемах. Розглянуто методи проектування цифрових пристроїв для формування і обробки дискретних сигналів. Наведені приклади проектування структурних і електричних принципів схем пристроїв різного функціонального призначення.

Призначено для студентів спеціальності “Автоматизоване управління технологічними процесами”.

421198

ISBN 966-7851-46-X

НТБ ВНТУ
м. Вінниця

ББК 32.844.1

© О.Ф. Єнікєєв,
О.О. Сердюк, 2004
© ДДМА, 2004

ЗМІСТ

ВСТУП	4
1 ОСНОВИ ТЕОРІЇ ПЕРЕМІКАЛЬНИХ ФУНКЦІЙ	5
1.1 Двійкова система обчислення	5
1.2 Аксиоми, основні теореми і тотожності алгебри логіки	6
1.3 Властивості перемикальних функцій	8
1.4 Форми запису логічних функцій	8
1.5 Мінімізація структурних формул за допомогою методу Квайна і карт Карно	13
1.6 Мінімізація системи логічних функцій	18
2 ПРОЕКТУВАННЯ КОМБІНАЦІЙНИХ ПРИСТРОЇВ	21
2.1 Універсальні логічні елементи	21
2.2 Приклади побудови комбінаційних пристроїв	22
2.3 Загальна задача синтезу комбінаційних пристроїв	26
2.4 Типові комбінаційні пристрої	30
2.4.1 Дешифратори і шифратори	30
2.4.2 Мультиплексори і демультиплексори	33
2.5 Проектування комбінаційних пристроїв, вільних від змагань сигналів	45
3 ПОСЛІДОВНІСНІ ПРИСТРОЇ	53
3.1 Засоби формального опису послідовнісних пристроїв	53
3.2 Тригери	62
3.3 Канонічний метод проектування ПП	71
3.4 Синтез тригерних пристроїв	76
3.5 Регістри	79
3.6 Лічильники	82
4 ЦИФРОВІ ПРИСТРОЇ ФОРМУВАННЯ ТА ОБРОБКИ ДИСКРЕТНИХ СИГНАЛІВ	85
4.1 Цифрові пристрої формування сигналів	85
4.2 Цифрові корелятори	87
4.3 Цифрові погодженні фільтри (ЦПФ)	89
4.4 Синтез логічних схем	92
СПИСОК СКОРОЧЕНЬ	106
ЛІТЕРАТУРА	107

Цифрові пристрої широко використовуються для генерування, формування і перетворення дискретних сигналів малої потужності. Інтенсивний розвиток теорії і методів оптимального прийому сигналів на тлі перешкод викликали необхідність розробки досить складних пристроїв перетворення сигналів. Такими пристроями є: погоджені фільтри, корелятори, пристрої нелінійної фільтрації і т.п. Їхня технічна реалізація можлива тільки на основі сучасних цифрових інтегральних схем.

У залежності від призначення і параметрів цифрових пристроїв застосовуються різні методи їхнього проектування. Так, наприклад, у багатьох системах передачі й обробки зображень, в першу чергу, необхідно вирішувати задачу підвищення швидкодії. Найчастіше раціональне проектування таких пристроїв пов'язано з правильним вибором елементної бази і побудовою електричної принципової схеми на мінімальному числі логічних елементів. Також при проектуванні цифрових пристроїв варто виключати змагання сигналів.

Значна частина цифрових приладів будується на основі комбінаційних пристроїв. У цих пристроях логічний стан виходу однозначно визначається комбінацією вхідних сигналів, що присутні в даний момент часу. Комбінаційні пристрої реалізуються на окремих інтегральних схемах (ІС) малого ступеня інтеграції, або виробляються у вигляді окремої ІС середнього ступеня інтеграції.

У послідовнісних пристроях (ПП) значення вхідних сигналів у кожний момент часу залежать не тільки від комбінації вхідних сигналів. Їхній вихідний сигнал залежить також від свого значення в попередній момент часу. Послідовнісні пристрої реалізують функціональний зв'язок між послідовностями окремих комбінацій вхідних і вихідних сигналів. Тобто такі пристрої мають пам'ять, що зберігає інформацію про попередні стани вхідних сигналів.

Навчальний посібник присвячений питанням, пов'язаним із проектуванням цифрових пристроїв перетворення дискретних сигналів на сучасній елементній базі. У ньому розглянуті особливості проектування комбінаційних і послідовнісних схем, а також саме пристроїв формування й обробки складних сигналів. Даний навчальний посібник не заміняє існуючу технічну літературу, а тільки доповнює її.

1 ОСНОВИ ТЕОРІЇ ПЕРЕМИКАЛЬНИХ ФУНКЦІЙ

1.1 Двійкова система обчислення

У цифрових електронних пристроях використовуються сигнали, які набувають тільки два значення - низький та високий рівні. Одному із цих значень відповідає цифра «0», іншому - «1». Така двійкова форма інформації дозволяє побудувати позиційну систему обчислення, в якій можна подати будь-яке число. У позиційній системі число подається рядом цифр, вага яких збільшується справа наліво. При цьому будь-яке число A в системі обчислення з основою q можна записати у вигляді

$$A = \sum_{j=0}^{n-1} a_j (q^j),$$

де $a_j \in [0 \dots (q-1)]$ - цифра в j -му розряді.

Наприклад, число $(A)_{10} = (1987)_{10}$ в десятковій системі можна перевести в двійкова:

$$\begin{aligned} (A)_2 &= 1 \cdot 2^{10} + 1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = \\ &= (A)_2 = (11111000011)_2. \end{aligned}$$

Комбінація цифр позиційної двійкової системи дозволяє побудувати кодові вирази, причому кількість комбінацій буде залежати від кількості двійкових розрядів.

Цифрові пристрої формують сигнали кодових комбінацій шляхом логічної обробки вхідних сигналів. Логічна обробка сигналів виконується згідно з теоремами, аксіомами і тотожностями алгебри логіки. Любий логічний вираз, зведений із змінних x_n, x_{n-1}, \dots, x_1 за допомогою кінцевого числа операцій алгебри логіки, можна розглядати як деяку функцію, яка являється зручним інструментом для опису, аналізу і синтезу перемикальних схем, вихідні сигнали яких характеризуються лише двома рівнями напруги: високим (логічна «1») і низьким (логічний «0»). У зв'язку з цим логічні вирази часто називаються перемикальними функціями.

1.2 Аксіоми, основні теореми і тотожності алгебри логіки

Змінні, які розглядаються в алгебрі логіки, можуть набувати тільки два значення: 0 та 1. Усі змінні обозначають латинськими буквами $x, y, z \dots$.

В алгебрі логіки визначені: відношення еквівалентності ($=$), операція диз'юнкція (АБО), що позначається знаком \vee , операція кон'юнкції (І), що позначається точкою, яку можна опустити (наприклад, xy), операція заперечення (НІ), що позначається рисою над змінною (наприклад, \bar{x}), а також операція логічної нерівнозначності (виключаючи АБО), що позначається знаком \oplus .

Відношення еквівалентності задовольняє наступним властивостям:

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

Алгебра логіки визначається наступними аксіомами:

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

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

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

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

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

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

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

- ідемпотентні закони

$$x \vee x = x; \quad x \cdot x = x;$$

- комутативні закони

$$x \vee y = y \vee x; \quad x \cdot y = y \cdot x;$$

- асоціативні закони

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

- дистрибутивні закони

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

- закони заперечення

$$x \vee \bar{x} = 1; \quad x \cdot \bar{x} = 0;$$

$$0 \vee x = x; \quad 1 \cdot x = x;$$

$$1 \vee x = 1; \quad 0 \cdot x = 0;$$

- закон подвійності (теорема де Морганя)

$$\overline{x \vee y} = \bar{x} \cdot \bar{y}; \quad \overline{xy} = \bar{x} \vee \bar{y};$$

- закон подвійного заперечення

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

- закон поглинання

$$x \vee x \cdot \bar{y} = x; \quad x(x \vee y) = x;$$

- операція склеювання

$$x \cdot y \vee x \cdot y = x \cdot y; \quad (x \vee y) \cdot (x \vee y) = x \vee y;$$

- операція узагальненого склеювання

$$xy \vee \bar{x}z \vee yz = xy \vee \bar{x}z;$$

$$(x \vee y)(\bar{x} \vee z)(y \vee z) = (x \vee y)(\bar{x} \vee z);$$

$$x \vee \bar{x}y = x \vee y;$$

$$x(\bar{x} \vee y) = xy.$$

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

1.3 Властивості перемикальних функцій

Сукупність змінних x_1, \dots, x_n можна розглянути як n -мірний вектор і як область визначення функції.

Для задання функції треба вказати її значення (0,1) у всіх точках області визначення. При цьому число різних функцій для n змінних дорівнюватиме 2^n .

Якщо функція залежить не від усіх змінних, то вона називається виродженою.

Найбільше поширення знайшли не вироджені функції двох змінних $f(x_2, x_1)$ з назвами операції алгебри логіки:

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

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

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

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

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

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

Таблиця 1.1

i	x_2	x_1	$x_2 \vee x_1$	$x_2 \cdot x_1$	$\overline{x_2 \cdot x_1}$	$\overline{x_2 \vee x_1}$	$x_2 \oplus x_1$
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 другі функції. Така підстановка можлива, тому що області значень функцій і змінних співпадають (0 або 1).

1.4 Форми запису логічних функцій

Логічні функції задаються або в табличній формі (таблицями істинності), або структурними формулами.

Таблиця істинності – це перерахування усіх можливих значень вхідних і вихідних сигналів в порядку зростання вхідних значень, наприклад, табл.1.2

Таблиця 1.2

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

Логічну функцію можна виразити за допомогою структурної формули, що містить логічні операції над незалежними перемінними, розташовані в певному порядку. Завдання логічної функції за допомогою структурної формули дозволяє побудувати схему пристрою, оцінити складність її реалізації мінімізувати число елементів заданого типу.

Існують дві форми запису структурної формули, що дають однозначне її уявлення: у виді звершеної диз'юнктивної нормальної форми (ЗДНФ) і у виді звершеної кон'юнктивної нормальної форми (ЗКНФ).

Запис структурної формули у виді ЗДНФ

Розглянемо на прикладі правило переходу від таблиці істинності ЛФ до її запису у виді 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

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

Число допоміжних функцій відповідає числу одиниць у таблиці істинності шуканої функції y . Структура допоміжних функцій у цьому випадку буде наступна: кожному значенню $y = 1$ відповідає одне значення 1 в стовпці y_i , так щоб число нулів у даному стовпці завжди дорівнювало семи. Функції такого виду називають повною елементарною кон'юнкцією, мінтермом чи конституантою одиниці. За табл. 1.3 можна скласти формули для $y_i = f_i(x_1, x_2, x_3)$, $y = f(y_1, y_2, y_3)$ і схему, що реалізує логічну функцію.

Запишемо значення основної і допоміжних функцій:

$$y_1 = x_1 \cdot x_2 \cdot x_3;$$

$$y_2 = \bar{x}_1 \cdot x_2 \cdot x_3;$$

$$y_3 = \bar{x}_1 \cdot \bar{x}_2 \cdot x_3;$$

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

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 виді. Отже, до властивостей ЗДНФ варто віднести наступні: у ЗДНФ немає двох однакових добутків, жоден добуток не містить двох однакових множників і жоден добуток не містить разом з незалежною перемінною її інверсію.

Для запису структурної формули у виді ЗДНФ за таблицею істинності необхідно:

- визначити рядки, де $y = 1$;
- для кожного визначеного рядка скласти логічний добуток усіх незалежних перемінних, причому, якщо у виділеному рядку незалежна перемінна дорівнює 1, то вона входить у відповідний добуток без інверсії, а якщо дорівнює 0 - з інверсією;
- скласти логічну суму отриманих добутків.

Сформульоване правило дозволяє записати структурну формулу без використання допоміжних функцій.

Запис структурної формули у виді ЗКНФ

Інший спосіб запису структурної формули за таблицею істинності може бути представлений у виді звершеною кон'юнктивної нормальної форми (ЗКНФ). Різниця полягає в тому, що число введених допоміжних функцій до 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 макстермами чи конститuentами нуля.

Таблиця 1.4

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

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

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

З табл.1.4 одержимо:

$$y_1 = x_1 + x_2 + x_3;$$

$$y_2 = \bar{x}_1 + x_2 + x_3;$$

$$y_3 = \bar{x}_1 + \bar{x}_2 + x_3;$$

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

Структурні формули у виді ЗДНФ і ЗКНФ еквівалентні і можуть бути перетворені одна в іншу на основі законів і тотожностей алгебри логіки [1...3].

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

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

Як видно, операція склеювання двох додатків виконується, коли додатки мають загальний множник, а другий співмножник в один додаток входить в інверсному виді, в інший - у прямому. Результатом операції склеювання є загальний множник.

У випадку, коли перший додаток входить як співмножник у другий додаток, виконується операція поглинання, результат якої - перший додаток:

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

Для ЗКНФ після еквівалентних перетворень одержимо

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

Якщо маємо два співмножники, кожний з яких являє собою логічну суму, причому в цих сумах є однакові складові, а неоднакові складові інверсні один одному, то здійснюється операція склеювання, результатом якої є загальна частина. Якщо один із двох співмножників входить в інший як доданок, 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:

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

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

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

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

1.5 Мінімізація структурних формул за допомогою методу Квайна і карт Карно

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

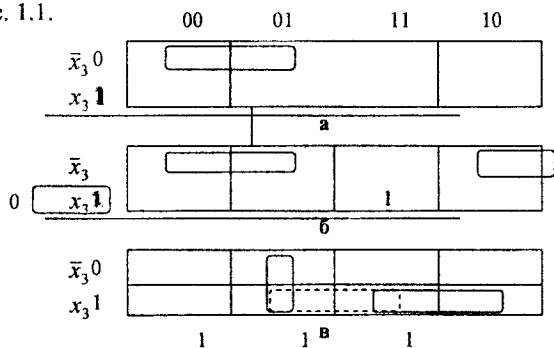
При мінімізації на основі карт Карно використовуються ті ж операції, що і при мінімізації за методом Квайна, але формою подання вихідних даних є карта з числом кліток 2^n , де n - число аргументів функції. Кожна карта відповідає одній функції, число n якої визначає розміри карти. Ознакою виконання операції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



а - для функції y_1 , б - для функції y_2 , в - для функції y_3

Рисунок 1.1 – Карты Карно

Операції склеювання показані замкнутим контуром, що охоплює сусідні одиниці. Відповідно до карти одержимо мінімізовану формулу, еквівалентну вихідній структурній. Виконання операції склеювання приводить до поглинання перемінних з різними значеннями. У мінімізованій формулі при наявності на карті чотирьох сусідніх кліток "випадають" дві, а при наявності двох сусідніх кліток - одна перемінна.

При мінімізації за методом Квайна використовуються операції склеювання і поглинання. Розглянемо три функції, задані таблицею істинності (табл.1.5).

Таблиця 1.5

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 ЗДНФ і спростимо формули, використовуючи операції склеювання і 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 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

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

Тут очевидні недоліки методу Квайна. По-перше, при складній функції зростає число аналізованих варіантів склеювань і поглинань, а по-друге, можуть

з'являтися зайві операції склеювання.

З цього можна зробити висновок, що для функцій трьох - чотирьох перемінних варто використовувати карти Карно, що забезпечують просту і наочну мінімізацію. У випадку ж більшого числа перемінних (п'ять-шість) варто застосовувати метод Квайна (чи Квайна - Мак-Класки)

Процедури одержання мінімізованої функції в кон'юнктивній формі використовують операції і прийоми, описані вище для диз'юнктивних форм. Пріоритетно їхнє застосування на прикладі. За допомогою методу Квайна проведемо мінімізацію структурних формул для функцій, заданих таблицею 1.6.

Таблиця 1.6

x_3	x_2	x_1	y_1	y_2
1	2	3	4	5
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

Враховуючи, що $(A+B)v(A+\bar{B})=A$, запишемо

$$y_1 = (x_1 + x_2 + x_3) \cdot (\bar{x}_1 + x_2 + x_3) = x_2 + x_3,$$

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

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

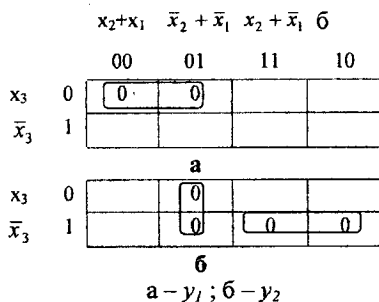


Рисунок 1.2 – Мінімізація функцій y_1 та y_2 за допомогою карт Карно

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

- застосування закону заперечення:

$$\overline{x_1 \cdot x_2} = \overline{x_1 + x_2} \quad ;$$

- використання розподільного закону

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

- додавання в структурну формулу складань, які тотожно дорівнюють нулю

$$\begin{aligned} y &= \underbrace{x_1 \cdot \overline{x_2} + \overline{x_1} \cdot x_2}_{=0} = x_1 \cdot \overline{x_2} + \overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_1} + x_2 \cdot \overline{x_2} = \\ &= x_1 \cdot (\overline{x_2} + \overline{x_1}) + x_2 \cdot (\overline{x_2} + \overline{x_1}) = (x_1 + x_2) \cdot (\overline{x_1} + \overline{x_2}) = \overline{x_1 \cdot x_2} \cdot (x_1 + x_2). \end{aligned}$$

Як впливає з останнього вираження, після додавання складань, тотожно рівних нулю, застосування двічі розподільного закону і закону заперечення, можна одержати структурну формулу, що містить на один елемент НІ менше, ніж у вихідній формулі.

Виникають випадки, коли логічні функції не цілком визначені.

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

Приклад. У логічному пристрої (рис.1.3) вихідний сигнал y_1 набуває значення $y_1=1$, якщо трирозрядний двійковий код K на вході,

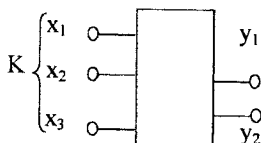


Рисунок 1.3 – Структура логічного пристрою

Змінюючись від 1 до 7, буде мати значення $K=1$, а якщо $K \neq 1$, то $y_1 = 0$. Заповнимо таблицю істинності для y_1 (табл.1.7). У першому рядку можна поста-

вити 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 в 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 (див.табл.1.7), невизначеність y_2 відзначимо знаками Φ_1, Φ_2, Φ_3 .

Таблиця 1.7

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, заданих у прикладі. При мінімізації функції y_1 за методом Квайна покладемо $\Phi = 1$ і подамо ЗДНФ у виді

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

Помітимо, що перший доданок є необов'язковим і відповідає Φ . Використання цього доданка дозволяє спростити функцію y_1 .

Для y_2 припустимо, що усі $\Phi_i = 1$, тоді одержимо

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

Тут перший, четвертий і п'ятий доданки є необов'язковими. Використовуючи операції склеювання стосовно до першого і другого, до першого і третього, третього і четвертого, а також до четвертого і 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 покласти $\Phi_1 = 1$, а $\Phi_2 = \Phi_3 = 0$, то мінімізована функція y_2 буде мати простий вигляд:

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

НТБ ВНТУ
М.Вінниця

Мінімізацію розглянутих вище функцій y_1 і y_2 можна провести і з використанням карт Карно (рис. 1.4, а, б відповідно), причому в цьому випадку вибір значень Φ , визначається розташуванням одиниць у клітках карт.

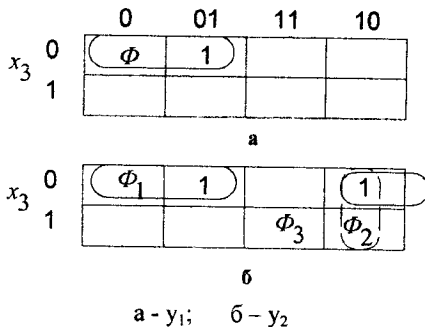


Рисунок 1.4 – Мінімізація функцій y_1, y_2 за допомогою карт Карно

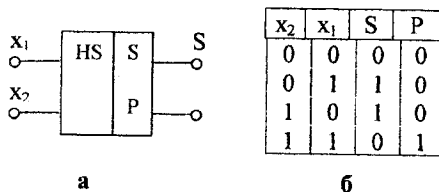
1.6 Мінімізація системи логічних функцій

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

Приклад 1. На рис. 1.5,а приведено умовне зображення півсуматора, який призначений для додавання двох двійкових цифр із формуванням суми (S) і цифри переносу в наступний розряд (P). Процес додавання двох двійкових цифр описується двома логічними функціями

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

Відповідно до правил додавання складемо таблицю істинності (рис 1.5 б), і запишемо структурні формули:



а – умовне зображення; б – таблиця істинності.

Рисунок 1.5 – Умове зображення і таблиця істинності півсуматора

Безпосереднє використання методу Квайна і карт Карно не дозволяє провести подальше спрощення функцій. У цьому зв'язку можна спробувати використовувати подання: $S = S' = f_3(x_1, x_2, p)$.

Складемо таблицю істинності для S' як функції трьох перемінних, використовуючи вихідну таблицю істинності (рис. 1.5,б) і правила роботи пристрою (табл. 1.8).

Таблиця 1.8

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 $px_1x_2 = 011$. У таблиці істинності така комбінація відзначається знаком Φ₁. Аналогічним способом відзначаються знаками Φ₂, Φ₃, Φ₄ 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 отримаємо 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		Φ ₄

Рисунок 1.6 – Карта Карно півсуматора

Порівнюючи отриману формулу з вихідною, бачимо, що для реалізації потрібна зменшена кількість логічних операцій

Приклад 2. Побудувати однорозрядний комбінаційний суматор, тобто пристрій, призначений для реалізації двох функцій: додавання двох двійкових цифр з урахуванням переносу з попереднього розряду і з формуванням цифр суми і переносу для наступного розряду.

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

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

0			1	
1		1	1	

а

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

б

а - функції додавання, б - формування результату

Рисунок 1.7 – Карти Карно

Карти Карно дозволяють отримати мінімізовану структурну формулу комбінаційного суматора

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

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 пристрою (табл.1.9).

Таблиця 1.9

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

За таблицею 1.9 і картою Карно (рис. 1.7,б) запишемо формулу для не цілком визначеної функції S' у виді

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

Порівнюючи число логічних операцій у формулах, S і S' , бачимо, що структурна формула S' має простіший вигляд і реалізується меншим числом елементів.

2 ПРОЕКТУВАННЯ КОМБІНАЦІЙНИХ ПРИСТРОЇВ

2.1 Універсальні логічні елементи

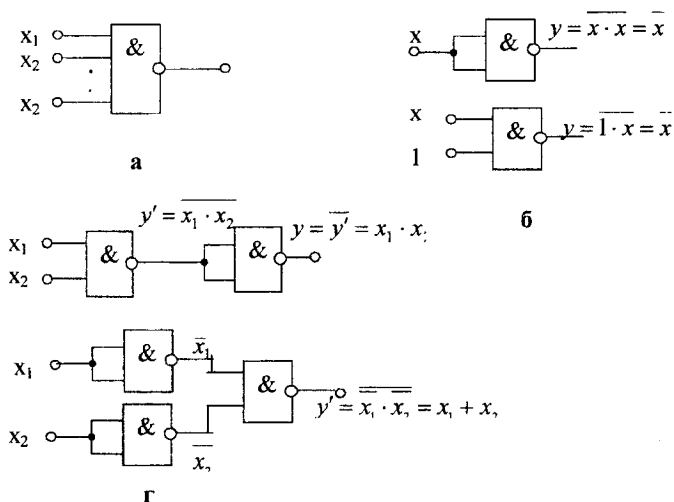
Для задоволення вимогам однорідності функціональної схеми застосовуються універсальні логічні елементи (ЛЕ). До таких елементів відносяться: І-НІ, АБО-НІ, І-АБО-НІ.

І-НІ (елемент Шеффера) (рис. 2.1, а). Реалізована логічна функція може бути записана у виді: $y = x_1 \cdot x_2 \cdot \dots \cdot x_n$.

За допомогою такого ЛЕ можна побудувати логічні функції НІ, І, АБО (рис. 2.1, б, в, г відповідно).

АБО-НІ (елемент Пірса). Логічна функція цього елемента може бути представлена в наступній формі

$$y = x_1 + x_2 + \dots + x_n.$$



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

Рисунок 2.1– Варіант використання ЛЕ І-НІ

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

Таблиця 2.1

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

Таблиця 2.2

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

Таблиця 2.3

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.

І-АБО-НІ (рис.2.2). Реалізована логічна функція має вид

$$y = \overline{x_1 \cdot x_2 + x_3 \cdot x_4}.$$

Так, наприклад, якщо $x_1 = x_3 = 1$, то $y = x_2 + x_4$. При $x_1 = 0$, $x_2 = 1$ маємо

$$y = \overline{x_3 \cdot x_4}.$$

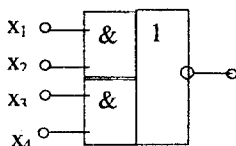


Рисунок 2.2 – Умовне зображення елемента 2І-АБО-НІ

2.2 Приклади побудови комбінаційних пристроїв

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

Приклад 1. Потрібно побудувати пристрій, що реалізує логічну функцію $y_1 = x_1 + x_2 \cdot \bar{x}_3$. Перетворимо логічну функцію, виключивши з неї операції

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

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

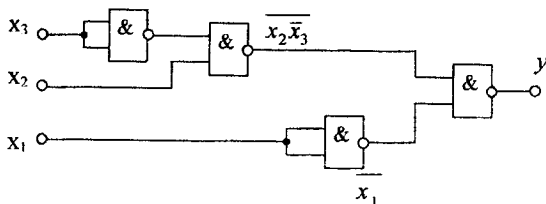


Рисунок 2.3 – Схема пристрою на ЛЕ І-НІ

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

Приклад 2. Для реалізації логічної функції

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

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

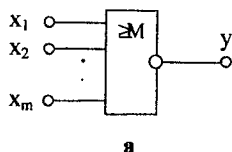
$$y = x_1 \cdot (x_2 + x_3) = \overline{\overline{x_1 \cdot (x_2 + x_3)}} = \overline{\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}}$$

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

Приклад 3. Синтезувати мажоритарний елемент (рис. 2.4 а), що уявляє

собой пристрій з непарним числом m входів, у якого $y = 1$, якщо на більшості входів сигнали рівні 1, і $y = 0$, якщо на більшості входів сигнали рівні 0. Склавши у відповідності зі словесним описом таблицю істинності (табл.2.4), одержимо за допомогою карти Карно (рис. 2.4,б) мінімізовану функцію для $m=3$:

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



а

	$x_2 x_1$	00	01	11	10
x_3	0			1	
	1	1	1		1

б

а - мажоритарного елементу; б - карта Карно

Рисунок 2.4- Умовне зображення елемента і його карта Карно

Таблиця 2.4

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 операції АБО, одержимо:

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

Реалізація цього виразу можлива на ЛЕ І-НІ.

Приклад 4. Побудувати однорозрядний комбінаційний суматор на елементах І-НІ. Робота суматора описується функціями:

$$P_i = x_2 \cdot x_1 + P_{i-1} \cdot x_1 + P_{i-1} \cdot x_2,$$

$$S = P_{i-1} \cdot \bar{P}_i + x_2 \cdot \bar{P}_i + x_1 \cdot \bar{P}_i + x_1 \cdot x_2 \cdot P_{i-1}.$$

Після відповідних перетворень, застосувавши правило де Моргана, одержимо:

$$P_i = x_2 \cdot x_1 \cdot P_{i-1} \cdot x_1 \cdot P_{i-1} \cdot x_2,$$

$$S = P_{i-1} \cdot \bar{P}_i \cdot x_2 \cdot \bar{P}_i \cdot x_1 \cdot \bar{P}_i \cdot x_1 \cdot x_2 \cdot P_{i-1}.$$

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

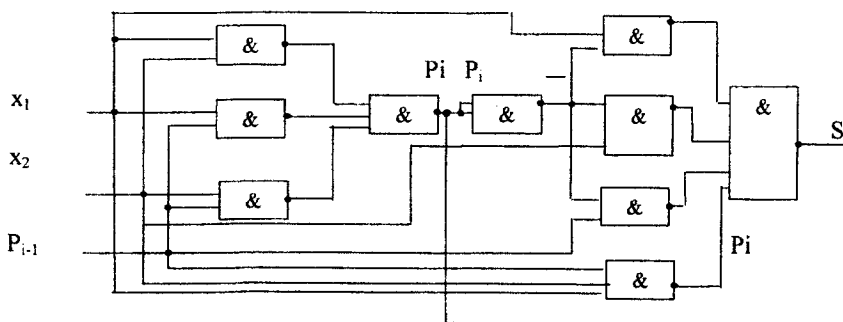


Рисунок 2.5 – Схема одно розрядного комбінаційного суматора на елементах І-НІ

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

Приклад 5. Розглянемо логічну функцію виду

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

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

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

Для реалізації такої функції буде потрібно три ЛЕ АБО-НІ.

З розглянутого приклада видно, що при синтезі пристроїв на ЛЕ АБО-НІ доцільно використовувати представлення логічної функції в ЗКНФ.

Приклад 6. Розглянемо синтез мажоритарного елемента (див. приклад 3).

За табл.10 за допомогою карт Карно одержимо мінімізовану функцію і запишемо її, використовуючи операції АБО-НІ:

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

Подаючи функцію в ЗДНФ, маємо:

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

Очевидно, що в другому випадку при реалізації буде потрібно більше число логічних елементів.

2.3 Загальна задача синтезу комбінаційних пристроїв

Вихідними даними для проектування комбінаційних пристроїв (КП) є їхній словесний опис і вимоги до основних електричних параметрів. У задачу синтезу КП входить одержання структурної схеми мінімальної складності, реалізованої 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 просте (мінімізоване) логічне вираження заданої функції. У більшості випадків процес спрощення (мінімізації) зводиться до застосування операцій склеювання і поглинання. При використанні алгебраїчних методів мінімізації - методів Квайна і Квайна - Мак-Класки за таблицею істинності записується ЗДФ (чи ЗКНФ) і проводиться склеювання наявних добутків (чи сум) [1 - 3]. При графічному методі, що використовує карти Карно [1], за таблицею істинності за допомогою карти Карно виявляють усі можливі склеювання і 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 і додавання складових тотожно рівних 0.

На третьому етапі синтезу роблять запис отриманої мінімізованої структурної формули в заданому базисі у виді комбінації операцій, виконуваних заданим (обраним) типом логічних елементів. Цей запис виробляється за допомогою відповідних прийомів. Так, при реалізації на елементах І-НІ над отриманою диз'юнктивною формою ставлять два знаки інверсії і за допомогою закону запе-

речення перетворюють інверсію диз'юнкції в кон'юнкцію інверсій. У результаті виходить логічне вираження, що містить тільки операції І-НІ.

Четвертим етапом синтезу є складання структурної схеми. 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, обумовлених типом використовуваних елементів і пропонованих вимог пристрою (швидкодія, простота реалізації, вартість та ін.). Проілюструємо можливе різноманіття рішень задачі побудови комбінаційного пристрою.

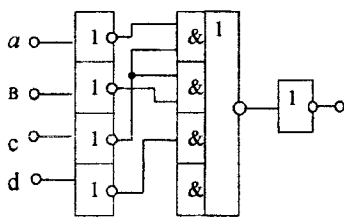
Приклад. Побудувати схему, що забезпечує мінімальні апаратні витрати і затримки сигналів при реалізації функції:

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

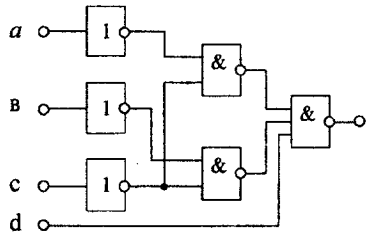
Апаратні витрати будемо оцінювати по числу застосовуваних корпусів мікросхем. Доцільно при використанні ЛЕ однієї серії покласти, що затримки сигналів ЛЕ різного типу однакові і рівні деякій усередненій для даної серії величині τ . Схемна реалізація заданої формули приведена на рисунку 2.6, а. Оскільки 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 елемента І-АБО-НІ, що займає корпус мікросхеми ЛРЗ. Таким чином, апаратні витрати W можна оцінити в такому виразі:

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

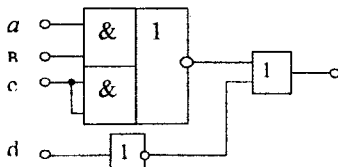
Невикористані елементи частково зайнятого корпусу (у даному випадку шостий інвертор у ІС ЛН1) не враховуються, тому що вони можуть бути застосовані в інших пристроях.



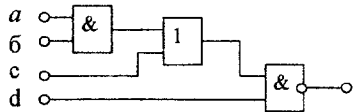
а



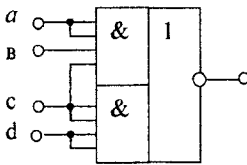
б



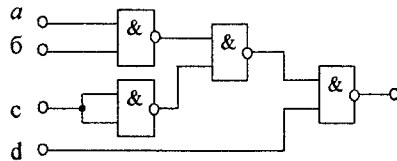
в



г



д



е

Рисунок 2.6 – Варіанти схемної реалізації заданої функції

Оскільки в ІС ЛРЗ частина входів не використовується, раціонально реалізувати схему за допомогою елементів І-НІ, що мають менше число входів. Застосовуючи закони заперечення до заданої формули, одержимо

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

При реалізації функції цього виду (рис. 2.6,б) затримка сигналу та апаратні витрати будуть рівні:

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

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

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

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

Можна одержати ще один вид запису структурної формули

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

при реалізації якої (рис. 2.6,г), потрібно менше корпусів ІС $W_4=9/12$

при $T_4=3\tau$. При подальшому перетворенні формули, одержимо

$$y_4 = \overline{a \cdot b \cdot d + c \cdot d}.$$

Структурна схема для y_4 представлена на рисунку 2.6, д. Тут

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

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

$$y_3 = \overline{a \cdot b \cdot \bar{c}} + \bar{d} = \overline{\overline{\overline{a \cdot b \cdot \bar{c}}}} + \bar{d}.$$

У структурній схемі (рис. 2.6,е) потрібно 1 корпус двовходових елементів І-НІ (К555ЛА3), тобто $W_6=12/12$ при $T_6=3\tau$.

Таким чином, як видно з розглянутого вище приклада; можливі різні схеми реалізації однієї і тієї ж структурної формули, отриманої 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, що 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 Типові комбінаційні пристрої

2.4.1 Дешифратори і шифратори

Дешифратором називають пристрій, що перетворює m - розрядний доічний код в однозначно відповідний йому унітарний. У дешифраторі (див.рис. 2.4) кожному значенню вхідного m - розрядного двоїчного коду відповідає сигнал 1, що з'являється тільки на тїм виході, номер якого дорівнює значенню вхідного числа. При цьому на всіх інших виходах дешифратора присутній сигнал 0. Дешифратор має $K=2^m$ виходів. Розглянемо приклади побудови дешифратора на елементах базису І, НІ, АБО.

Приклад 1. Припустимо, що $m = 2$, $K = 2^m = 4$. За таблицею істинності (табл.2.5) одержимо:

$$y_0 = \bar{x}_1 \cdot \bar{x}_2, \quad y_1 = x_1 \cdot \bar{x}_2, \quad y_2 = \bar{x}_1 \cdot x_2, \quad y_3 = x_1 \cdot x_2.$$

Таблиця 2.5

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, що реалізує зазначені логічні функції,

$$y_0 = \bar{x}_1 \cdot \bar{x}_2 = \overline{x_1 \cdot x_2}, \quad y_1 = x_1 \cdot \bar{x}_2 = \overline{x_1 \cdot x_2},$$
$$y_2 = \bar{x}_1 \cdot x_2 = \overline{x_1 \cdot x_2}, \quad y_3 = x_1 \cdot x_2 = \overline{x_1 \cdot x_2}.$$

наведена на рис. 2.7.

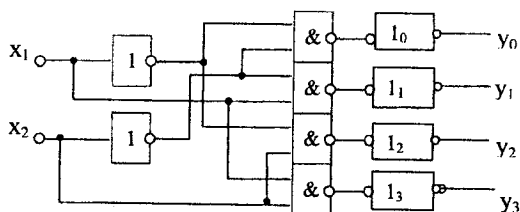


Рисунок 2.7 - Схема дешифратора на 4 виходи

Приклад 2. Розглянемо проектування дешифратора, що має шість входів і 64 виходи. Таблиця істинності дешифратора (табл.2.6.) дозволяє записати:

Таблиця 2.6

x_6	x_5	x_4	x_3	x_2	x_1	y_0	y_1	$y_2 - y_{62}$	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 дешифратора потрібно 64 корпусів ІС 8-ми входових елементів І-НІ і 16 корпусів ІС, що містять по 4 двохходових елемента І-НІ. Зменшити число корпусів ІС можливо шляхом переходу до двоступінчатої схеми, у якій на першій і другій ступені використовуються дешифратори ДС1 і ДС2 з $m = 3$. У таблиці 2.7 дано опис дешифратора ДС 1.

Таблиця 2.7

x_3	x_2	x_1	y'_0	y'_1	-	y'_7
0	0	0	1	0	-	0
0	0	1	0	1	-	0
-	-	-	-	-	-	-
1	1	1	0	0	-	1

Структурні формули ДС1 мають наступний вид:

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

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

.....

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

Аналогічно для ДС2 маємо:

$$y_0'' = \bar{x}_4 \cdot \bar{x}_5 \cdot \bar{x}_6 = \overline{x_4 \cdot x_5 \cdot x_6},$$

$$y_1'' = x_4 \cdot \bar{x}_5 \cdot \bar{x}_6 = \overline{x_4 \cdot x_5 \cdot x_6},$$

.....

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

Тоді

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

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

.....

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

Побудова такого двоступінчастого дешифратора наведена на рис.2.8.

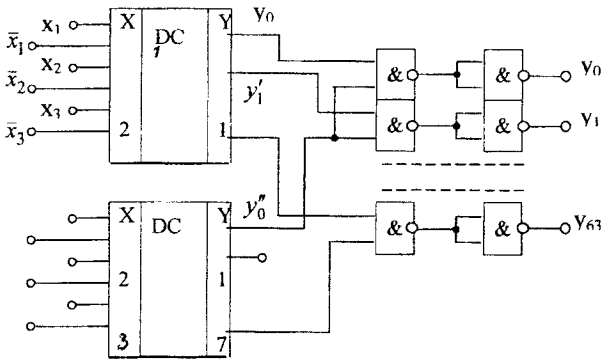


Рисунок 2.8 – Двоступінчастий дешифратора на 64 виходи

Шифратор виконує функцію, зворотну дешифратору, тобто перетворить унітарний код на вході в l - розрядний двійковий код на виході. При наявності сигналу 1 на одному з входів на l виходах шифратора з'являється l - розрядний код, значення якого дорівнює номеру входу, де є присутнім сигнал 1. Число входів $m+1$ і виходів l шифратора зв'язано співвідношенням $2^l = m+1$. Наприклад, при $m = 3, l = 2$. Функціонування описується таблицею істинності (табл. 2.8) шифратора.

Таблиця 2.8

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

Запишемо рівняння для цього прикладу:

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

Реалізація такого шифратора на елементах АБО наведена на рисунку 2.9 .

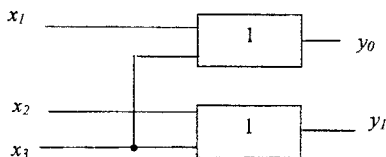


Рисунок 2.9 – Реалізація шифратора

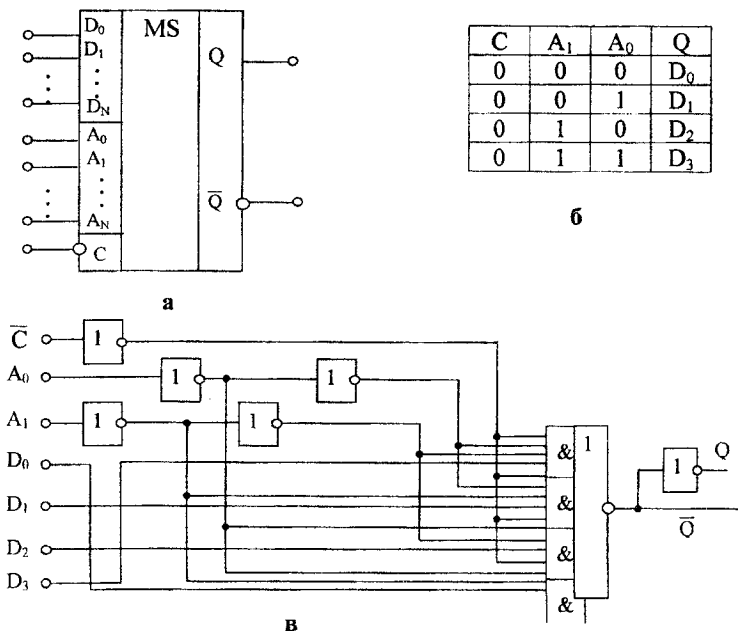
Найбільш поширені кодоперетворювачі двоїчного коду в двоїчно-десятичний і навпаки, а також двійковий коду в код Грея і навпаки. Такі перетворювачі створюють у вигляді постійних запам'ятовуючих пристроїв (К155 ПР 6, К155 ПР7).

2.4.2 Мультиплексори і демультиплексори

Особливу групу комбінаційних пристроїв складають логічні комутатори, тобто схеми, призначені для виборчого переключення вхідних каналів [2 ... 6]. До них відносяться мультиплексори.

Мультиплексор є пристроєм, що у залежності 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) і забезпечує спрямовану передачу його сигналу на вихід (Q). Мультиплексор можна уявити як безконтактний багатопозиційний перемикач. На рисунку 2.10, а показане позначення мультиплексора на функціональних схемах зі стробуванням входу S . Кожному інформаційному входу мультиплексора приписується номер, названий адресою. При стробуючому сигналі на вході S , мультиплексор підключає інформаційний вхід, адреса якого відповідає

двійковому коду на адресних входах, до виходу. Змінюючи сигнали на адресних входах, можна забезпечити передачу сигналів з різних інформаційних входів на вихід. Число інформаційних входів N і число адресних входів M зв'язані співвідношенням: $N = 2^{M+1}$.



а - схематичне зображення мультиплектора,
б - опис роботи, в - структурна побудова

Рисунок 2.10 – Схематичне зображення мультиплектора, опис його роботи та структурна побудова для варіанта

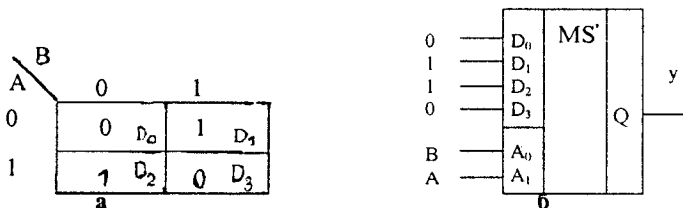
Роботу мультиплектора, наприклад, із двома адресними і 4 інформаційними входами, можна описати таблицею (рис. 2.10, б). При відсутності стробуючого сигналу ($\bar{C} = 1$) зв'язок між інформаційними входами і виходом відсутній (при цьому $Q = 0$). При наявності стробуючого сигналу ($\bar{C} = 0$) на вихід передається логічний рівень того з інформаційних входів D_i , номер якого у двоїчній формі заданий сигналами на адресних входах. Так, якщо $A_1\bar{A}_0 = 10_2 = 2_{10}$, на вихід Q буде передаватися сигнал із входу D_2 . За таблицею можна записати логічну функцію виходу мультиплектора:

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

Структурна схема такого мультиплектора наведена на рисунку 2.6,в.

Мультиплектори можуть бути використані для синтезу КП, що реалізують логічні функції багатьох перемінних. При цьому мультиплексор працює як універсальний ЛЕ, який реалізує будь-яку логічну функцію, що містить до $M + 1$ перемінних, де M - число адресних входів мультиплектора [2, 3]. Один мультиплексор у цьому випадку заміняє кілька корпусів з логічними елементами І, АБО, НІ. При синтезі КП на мультиплексорах треба скласти карту Карно заданої функції. Число інформаційних входів мультиплектора вибирають рівним числу кліток карти Карно. Якщо двійкові коди на адресних входах мультиплектора і коди кліток 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 мультиплексорів при синтезі КП і деякі прийоми підвищення ефективності використання мультиплексорів.

Приклад 1. Задана функція нерівнозначності, що описується вираженням $y = A \cdot \bar{B} + \bar{A} \cdot B$ і картою Карно (рис. 2.11,а). У правому нижньому куті кожної клітки карти позначимо відповідний інформаційний вхід мультиплектора. Логічні рівні 0 і 1, записані в клітках, привласнимо значенням вхідних сигналів D_i ($i = 0, \dots, 3$) мультиплектора. Якщо на входи мультиплектора подати код, вказаний на рис. 2.11,б, то при подачі перемінних А і В на входи адреси A_0 A_1 маємо: $AB = 00$, значення $y = 0$, при $AB = 01$ $y = 1$; $AB = 10$ $y = 1$, $AB = 11$ $y = 0$.



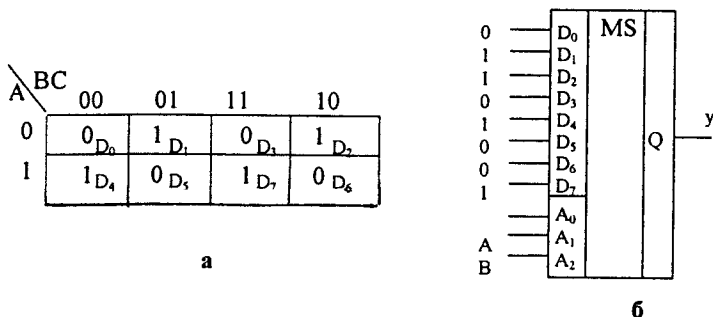
а - Карта Карно; б - схематичне зображення мультиплектора

Рисунок 2.11 – Карта Карно і схематичне зображення мультиплектора

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 рисунку 2.12, а, Для реалізації логічної функції використовуємо 8-входовий мультиплексор (рис. 2.12,б)

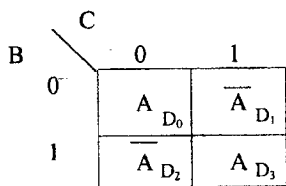


а – карта Карно, б – схема подачі сигналів

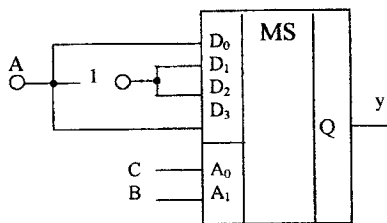
Рисунок 2.12 – Побудова карти Карно і схема підключення мультиплексора

Однак ту ж функцію можна реалізувати за допомогою мультиплексора з меншим числом входів. Нехай, наприклад, вхідні сигнали В і С, як і раніше, будуть присутні на адресних входах мультиплексора, а вхідний сигнал А використаємо для формування сигналів на інформаційних входах. При цьому кожній комбінації сигналів ВС на карті Карно буде відповідати не клітка, а стовпець. Зіставимо кожному стовпцю карти один із входів 4-входового мультиплексора, що еквівалентно переходу до 4-клітинної карти Карно (рис. 2.13,а). У новій 4-клітинній карті Карно значення функції y виражені через вхідний сигнал А за допомогою первісної (8-клітинної) карти, де значення y збігаються з А у клітках першого стовпця ($y = A$), значення y дорівнюють інверсним значенням 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. 2.13,а) в обох клітках стовпця $y = 1$ або $y = 0$.

Практична реалізація функції y за допомогою 4-входового мультиплексора показана на рис. 2.13,б.



а



б

а – карта Карно, б – схема подачі сигналів

Рисунок 2.13 – Варіанти реалізації функції

Приклад 3. Більш складним прикладом синтезу КП на основі мультиплектора є 4-входовий цифровий компаратор для сигналів ABCD. Нехай: $y = 1$ у тих випадках, коли три і більше вхідних сигналів дорівнюють 1. Відповідно до цього карта Карно подана на рисунку 2.14, а. Покладемо, що вхідні сигнали А, С и Д будуть надходити на адресні входи, а сигнали В використовуємо для формування сигналів на інформаційних входів. У загальному випадку доцільно подавати на адресні (селекторні) входи мультиплектора перемінні, котрі входять у більше число доданків структурної формули в ЗДНФ.

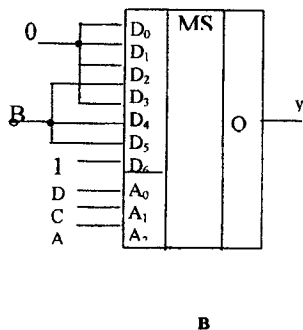
У розглянутому прикладі в кожному стовпці вихідної карти попарно об'єднаємо клітки при $A = 0$ і $A = 1$, а значення вихідної функції y для кожної пари кліток виразимо через значення перемінної В. Цей прийом дозволяє перейти до 8-клітинної карти Карно (рис 2.14,б) з допомогою якої встановлюємо стани 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.д. Практична реалізація такого компаратора подана на рисунку 2.14, в.

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

а

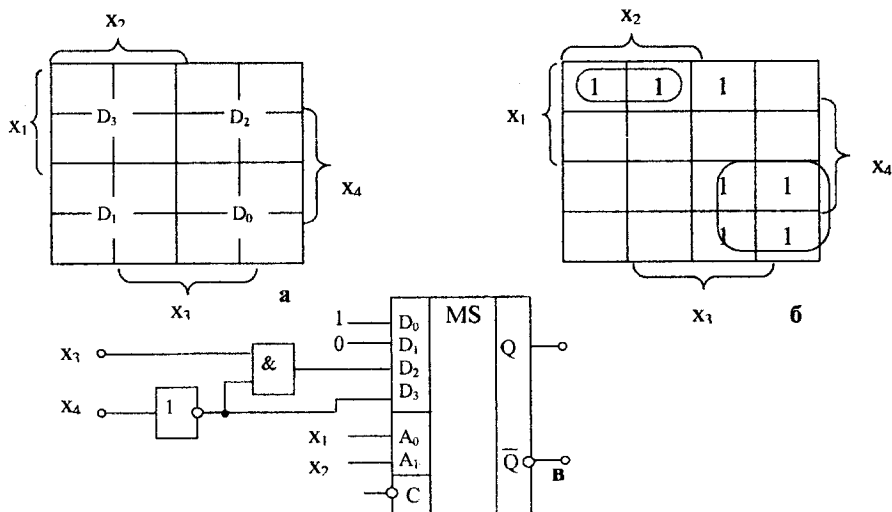
A \ CD		00	01	11	10
		0	0	B	0
0	0	0	B	1	B
	0	B	1	B	
1	0	B	1	B	

б



а – початкова карта Карно, б – 8-клітинна карта, в – схема подачі сигналів
Рисунок 2.14 – Формування карт Карно, схеми подачі сигналів

Якщо для формування сигналів на інформаційних входах мультиплексора використати ЛЕ, можна скоротити число його входів. Нехай, наприклад, потрібно синтезувати логічну функцію чотирьох перемінних $f(x_1, x_2, x_3, x_4)$ з використанням 4-входового мультиплексора. Якщо адресними перемінними вибрати x_1 і x_2 , то на інформаційні входи мультиплексора повинні надходити перемінні x_3 і x_4 , обумовлені, як показано на рисунку 2.15,а, областями діаграм Вейча [2]. Усередині кожної чергової області діаграми Вейча проводиться процедура мінімізації. Якщо логічна функція чотирьох перемінних задана картою (рис. 2.15,б), то на інформаційні входи $D_0 \dots D_3$ мультиплексора повинні надходити сигнали 1, 0, $(x_3 \cdot \bar{x}_4)$ і \bar{x}_4 , відповідно. Реалізація заданої функції приведена на рисунку 2.15,в.

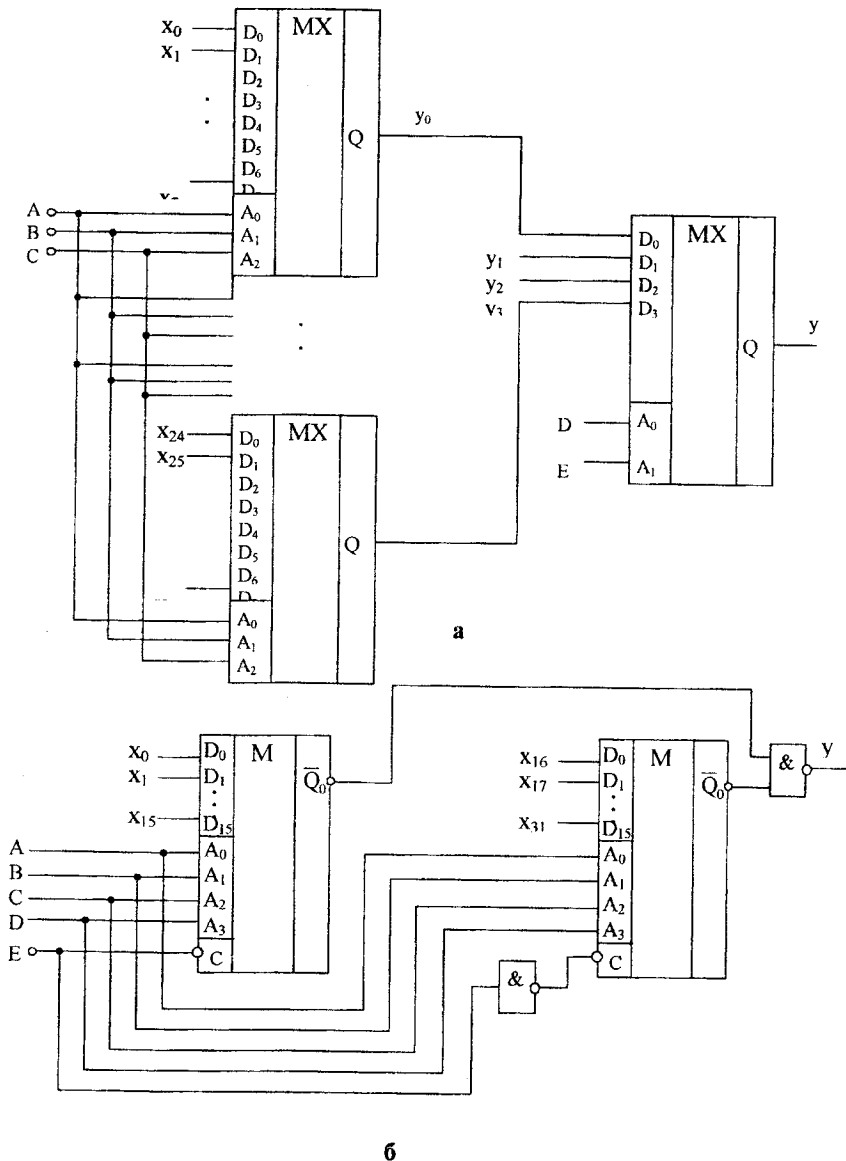


а – діаграма Вейча, б – карта Карно, в – схемна реалізація

Рисунок 2.15 – Синтез комбінаційного пристрою на мультіплексорі з використанням ЛЕ

У мультіплексорів, що випускаються у виді ІС [2...5], число інформаційних входів не перевищує шістнадцяти. Більше число входів можна забезпечити або шляхом об'єднання декількох мультіплексорів у пірамідальну (деревоподібну) систему, або шляхом використання послідовного з'єднання входів, строб-сигналів і зовнішніх логічних елементів.

При пірамідальній побудові схеми мультіплексора кожна наступна ступінь, починаючи з першої, має менше входів, чим попередня. Молодші розряди коду адреси підключені до адресних входів першої ступіні. Ступіням більш високого рангу відповідають старші розряди коду. На рисунку 2.16,а приведена схема мультіплексора з 32 інформаційними входами на основі чотирьох 8-входових і одного 4-входового мультіплексорів. Велике число ІС є недоліком пірамідальної побудови схеми мультіплексора. Зменшити кількість ІС можливо шляхом побудови мультіплексорів із використанням входів стробових сигналів (рис. 2.16,б).



б

а – пірамідальна побудова, б – з використанням строб-сигналів

Рисунок 2.16 –Схема розширення числа входів мультиплексора

На відміну від мультиплексорів, виконаних по ТТЛ-технології мультиплексори на основі КМОП технології будуються з використанням дешифраторів і двоспрямованих ключів (рис. 2.17) [3]. Мікросхема містить перетворювач логічних рівнів, який забезпечує узгодження потенціалів цифрових вхідних сигналів і внутрішніх потенціалів мікросхеми, дешифратор, що здійснює перетворення вхідного коду на адресних входах А, В, С в сигнал на одному з його виходів, а також двоспрямовані ключі, керовані 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 і схемою підключення двоспрямованих ключів.

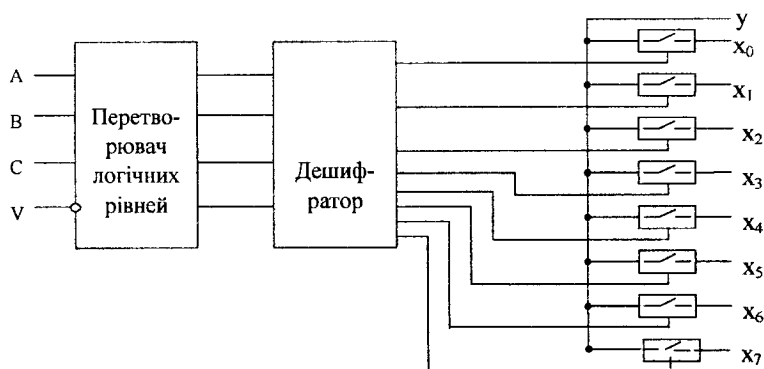


Рисунок 2.17 – Структура мультиплексора КМОП технології

Демультимплексори виконують функції, протилежні функціям мультиплексора. Вхідний сигнал x надходить на один з N виходів у залежності від значення коду $A_1 \dots A_M$ (рис. 2.18). При цьому демультимплексор має $N = 2^M$ виходів (прямих чи інверсних). Приклад таблиці істинності для $N = 4$ наведено в таблиці 2.9.

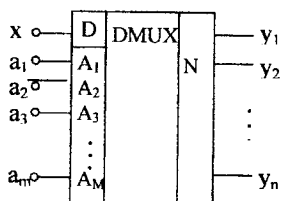


Рисунок 2.18 – Схематичне позначення демультиплексора

Таблиця 2.9

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 у відповідності із адресою виході також формується логічна одиниця (на інших виходах логічний нуль). Демультиплексор в даному випадку є дешифратором. Якщо на вході D встановлено логічний нуль, то мікросхема виконує функцію демультиплексора. Такі ІС називаються дешифраторами-демультиплексорами. У звичайних випадках дешифратори-демультиплексори мають чотири, вісім або шістнадцять виходів.

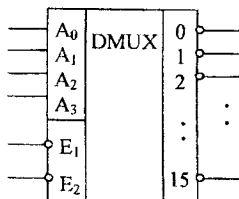
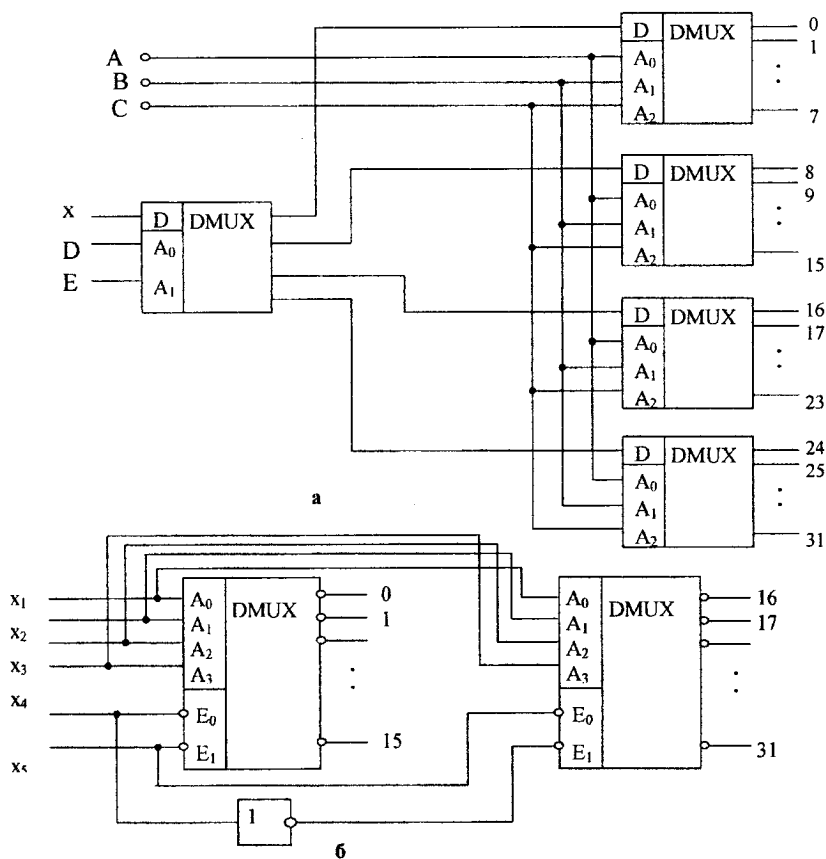


Рисунок 2.19 – Дешифратор-демультиплексор на 16 виходів

Він має два входи дозволу ($\overline{E0}$ и $\overline{E1}$) та чотири адресних входа. Для демультиплексора 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 $\overline{E1}$, IC 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. Демультіплексори можуть нарощуватися в систему, як і мультиплексори, шляхом побудови демультіплексорного дерева (пірамідальної структури) чи з використанням входів дозволу (рис. 2.20, а, б).

За допомогою демультіплексорів-дешифраторів здійснюється реалізація логічних функцій, заданих таблицями істинності, наприклад, таблицею 2.10.



а - демультіплексори; б - дешифратори-демультіплексори

Рисунок 2.20 – Схеми розширення числа виходів

Таблиця 2.10

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

Шукана функція y може бути подана у виді

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

де m_i - відповідний мінтерм чи кон'юнкція.

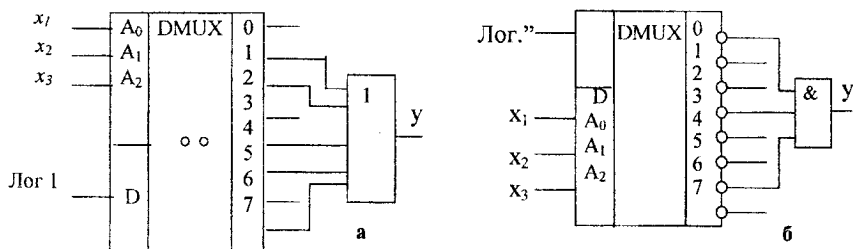
Перепишемо це вираження в інверсній формі

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

Перетворимо отриману функцію з використанням інверсії

$$y = \overline{m_0 + m_3 + m_6} = \bar{m}_0 \cdot \bar{m}_3 \cdot \bar{m}_6.$$

Схемні реалізації двох останніх функцій приведені на рисунку 2.21, а, б, відповідно.



а - на демультіплексорі; б - на дешифраторі

Рисунок 2.21 – Схемні реалізації логічної функції

Найбільш ефективним є використання демультіплексорів у комбінаційних пристроях, які мають значне число вихідних сигналів: розподільники імпульсних послідовностей [4], пристрої формування сигналів [7], формування декі-

льких логічних функцій тих самих перемінних [3] і ін. При спільному використанні мультиплексора і демультимплексора можна побудувати пристрої, у якому їх входи-виходи підключаються по заданим адресам [3, 7].

При реалізації комбінаційних пристроїв на мультиплексорах і демультимплексорах, що представляють собою ІС середнього ступеня інтеграції, скорочується необхідне число корпусів ІС і число зв'язків (сполучень) між ними і, як наслідок, підвищується надійність роботи комбінаційних пристроїв

2.5 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 розуміється 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 інше, 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 можуть бути безпечними і небезпечними. Безпечними є змагання, що 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 інше. 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 виді двох частин: одна безінерційно виконує логічну функцію, інша включена слідом за першою і є елементом затримки на час, рівний середній затримці сигналів $t_{сер}$.

Використовуючи таке подання логічного елемента, розглянемо приклад послідовного з'єднання декількох ЛЕ, що реалізують логічну функцію $y = x_1 + x_2 + x_3$ (рис. 2.22,а).

Припустимо, що при $t < t_0$ значення $x_1 = 0$, $x_2 = x_3 = 1$, а при $t = t_0$ вхідні сигнали інвертуються.

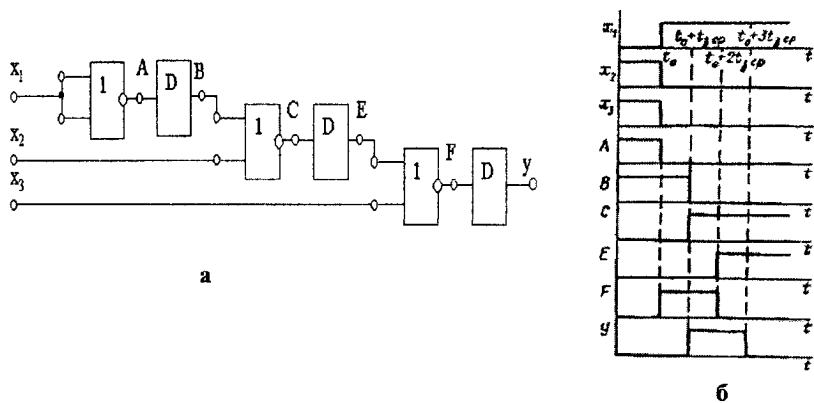
Без обліку затримок значення вихідного сигналу $y = 0$. Простежимо динаміку роботи пристрою, використовуючи модель і тимчасові діаграми вхідних і вихідних сигналів, подані на рис. 2.22,б.

Елементи затримки D включені послідовно з ЛЕ, затримуючи проходження сигналів на $t_{сер}$ в точках В, Е і на виході y . Результати аналізу роботи КП приведені в таблиці. Із таблиці видно, що на виході y виникає перешкода тривалістю

$2 t_{сер}$ у виді лог. "1"

У залежності від умов переходу комбінаційного пристрою від одного стану до іншого розрізняють статичні і 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, що містяться в пристрої, називають динамічними.

Статичні змагання підрозділяють на одиничні й нульові.



а – модель КП; б – тимчасові діаграми

Рисунок 2.22 – Приклад появи перешкоди при виникненні запізень

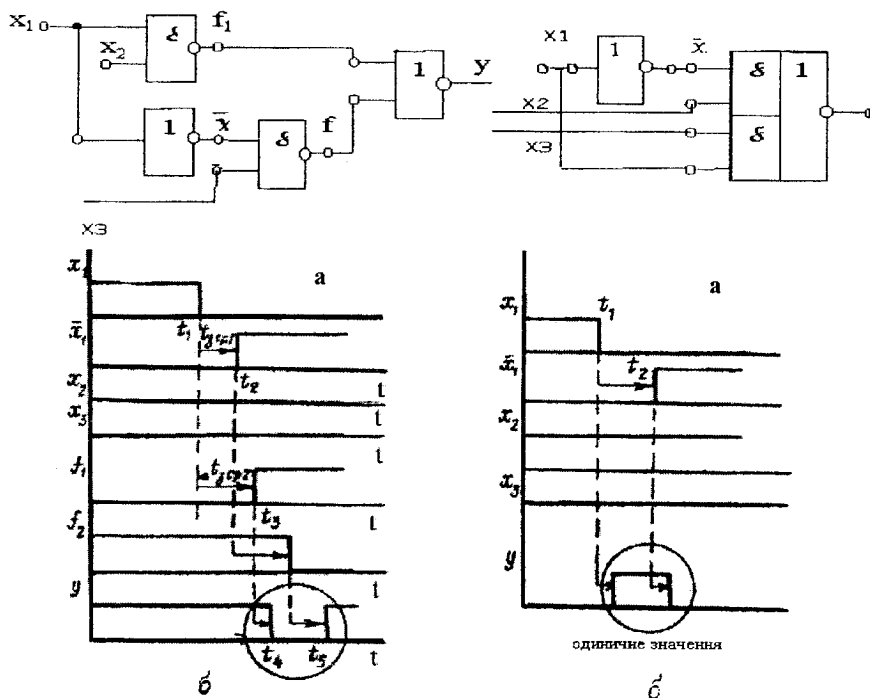
Таблиця 2.11

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
$t_0 + t_{3cp}$	1	0	0	0	0	1	0	1	1
$t_0 + 2t_{3cp}$	1	0	0	0	0	1	1	0	1
$t_0 + 3t_{3cp}$	1	0	0	0	0	1	1	0	0
$t_0 + 4t_{3cp}$	1	0	0	0	0	1	1	0	0

Одиничні змагання з'являються у випадку, коли при зміні вхідних 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.

Наведемо приклади утворення неалгоритмічних переходів при статичних змаганнях. Для ілюстрації розглянемо пристрій (рис. 2.23), що описується логічною функцією $y = x_1 \cdot x_2 \cdot x_3 \cdot x_1$. Тимчасові діаграми, що пояснюють виник-

нення змагань сигналів, приведені на рисунку 2.23,б, причому в ЛЕ НІ затримка сигналу дорівнює t_{31} , а в ЛЕ І - НІ - t_{32} . З аналізу тимчасових діаграм видно, що в КП виникли два неалгоритмічних переходи (одиночного змагання). Процес утворення нульових змагань покажемо на прикладі КП (рис.2.24,а), що реалізує логічну функцію $y = \overline{x_1} \cdot x_2 + x_3 \cdot x_1$. Як неважко переконатися (див. рис. 2.24,б), у такому пристрої можливе виникнення нульових змагань.



а - схема пристрою
б - тимчасові діаграми

Рисунок 2.23 – Приклад появи
одиночних змагань

а - схема пристрою
б - тимчасові діаграми

Рисунок 2.24 – Приклад появи
нульових змагань

Аналіз роботи КП з метою виявлення небезпечних змагань по тимчасових діаграмах при великій кількості ЛЕ виявляється досить трудомістким. Тому, як правило, використовуються формальні методи аналізу.

Розглянемо докладніше один з них, а саме аналітичний метод, чи метод Мак-Класки [1]. Логічну функцію y , що описує роботу КП, перетворюють або в диз'юнктивну нормальну форму (ДНФ) при аналізі на статичні нульові змагання, або в кон'юнктивну нормальну форму (КНФ) при аналізі на статичні одиночні змагання. При цьому не допускаються такі логічні перетворення, при яких відбувається втрата вхідних перемінних x_i і \bar{x}_i . З цією метою x_i і \bar{x}_i розглядають як незалежні перемінні, оскільки в динамічному режимі (тобто під час перехідних процесів) у деяких точках ланки ЛЕ x_i і \bar{x}_i можуть виявитися не взаємно інверсними, а одночасно рівними або логічному "0", або логічній "1".

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

а) логічна функція y в ДНФ містить хоча б один з доданків, у який одна з перемінних входить в прямому x_i і інверсному \bar{x}_i виді (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 доданок у цьому виразі виходить шляхом підстановки в логічну функцію y , записану в ДНФ, замість x_i логічної одиниці, а замість \bar{x}_i - логічного нуля. Другий доданок виходить шляхом підстановки замість x_i значення логічного "0", а замість \bar{x}_i - логічної одиниці.

Усі інші вхідні перемінні x_1, x_2, \dots при цьому залишаються записаними в загальному виді. Після відповідних спрощень ці перемінні 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 y у КНФ містить хоча б один співмножник, в який од-
на з перемінних входить у пряму x_i і інверсному \bar{x}_i виді, наприклад,
($x_i + \bar{x}_i$);

б) виконується 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-Класки на прикладі фун-
кції

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

Із структурної формули видно, що змагання сигналів можуть виникнути в
ЛЕ АБО-НІ. Припустимо, що в такому КП можливі статичні нульове змагання.
Для перевірки цього припущення представимо вираження для y в ДНФ:

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

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

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

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 пере-
творимо вираження для y в КНФ

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

Перша 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 розгля-
нути на конкретному прикладі. Для цього проведемо аналіз КП, логічна функція
 y , якого визначається 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.2.12) причому

в даному випадку зручно взяти замість функції y інверсну функцію, для якої побудуємо карту Карно (рис 2.25). Виконавши лінеаризацію одержимо наступний вираз логічної функції

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

Таблиця 2.12

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

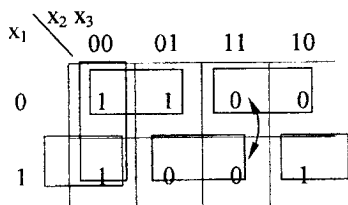


Рисунок 2.25 – Карта Карно для інверсного значення функції y

На картах Карно сусідні одиниці й сусідні нулі охоплюються загальним контуром (контуром склеювання). Якщо при зміні однієї з вхідних перемінних зміни функції y такі, що на карті відбувається перехід з одного контуру склеювання в інший, з ним не зв'язаний, то в КП мають місце змагання. Так, на рис. 2.25 видно, що при переході функції y від значення логічної 1 до 1 у випадку зміни однієї з вхідних перемінних 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 в інший контур. 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.

При проектуванні КП, вільних від небезпечних статичних змагань сигналів, як правило, 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 і \bar{y} контури склеювання, причому контури відповідають одній із цих складних функцій.

Корекція небезпечних змагань. Комбінаційний пристрій у виді узагальненої структурної схеми (рис. 2.26), де A_1 і A_2 - ланцюги, за якими відбувається передача вхідних сигналів, що створюють на виході елемента D_n неалгоритмічні переходи, які поширюються далі по ланцюзі 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 у ґрунтується метод корекції небезпечних змагань.

Розкид значень затримок сигналів у ланцюгах A_1 і A_2 визначає тривалість складного сигналу, який утвориться на виході D_n (див. рис. 2.26). Задача корекції небезпечних змагань ланцюгів A_1 і A_2 у всіх відомих випадках складатиметься у визначенні припустимої різниці затримок $\tau_{доп}$ 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).

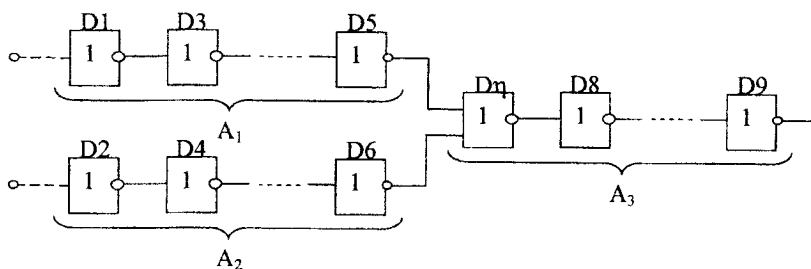


Рисунок 2.26 – Узагальнена структурна схема КП

Уведення синхронізації. Роботою цифрових пристроїв у даному випадку керують тактові (синхронізуючі) послідовності так, що запис і 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. Помітимо, що введення синхронізації істотно зменшує швидкодію пристрою.

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

Застосування визначеного порядку зміни сигналів і станів КП. Оскільки небезпечні змагання виявляються у виді 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.

На закінчення необхідно ще раз підкреслити, що проблема змагань у цифрових пристроях є дуже серйозною, тому що з ними пов'язані відмови у роботі схем.

3 ПОСЛІДОВНІСНІ ПРИСТРОЇ

3.1 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 приході вхідних сигналів.

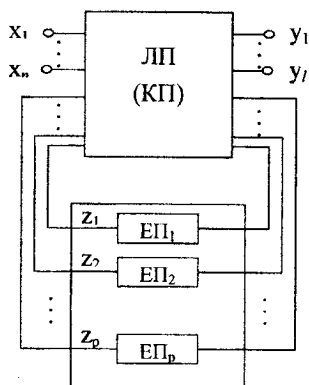


Рисунок 3.1 – Структурна схема ПП

Внутрішній стан ПП відбиває передісторію (послідовність) його роботи, тоб

то пам'ять пристрою. Наявність пам'яті є характерною властивістю будь-якого ПП, тому структурна схема ПП (рис. 3.1) може бути представлена складеною з двох блоків: логічного перетворювача (ЛП), виконаного на основі КП, і блоку пам'яті (БП), як сукупності елементів пам'яті $ЕП_1, ЕП_2, \dots, ЕП_p$.

Послідовнісні пристрої можна описати 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 $Z = \{z_1, z_2, \dots, z_p\}$, де x_i, y_i, z_i – вхідні, вихідні і внутрішні перемінні, а також двома функціями: функцією переходів, що визначає порядок зміни внутрішніх станів, і функцією виходів, що задає стан виходів у залежності від стану входів і внутрішнього стану.

При роботі цифрових пристроїв можуть бути виділені тимчасові інтервали, на яких усі перемінні (вхідні, вихідні і 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 , так і довільно розташовані, вважаючи, що саме в ці моменти відбувається зміна тієї чи іншої вхідної перемінної.

Позначимо x_i^n величину вхідної перемінної x_i у момент часу t_n . Запис тієї ж вхідної перемінної у формі x_i^{n-1} і x_i^{n+1} буде означати, що ця перемінна розглядається в моменти t_{n-1} і t_{n+1} .

Розрізняють два типи ПП: асинхронні і 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, принаймні, один виділений вхід синхронізації (С) і вхідні сигнали можуть впливати на ПП лише при наявності визначеного сигналу на цьому вході.

Для того, щоб одержати алгоритм функціонування ПП за словесним описом його роботи, необхідно задати функції його переходів і 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 в зіставленні кожній парі X_i і Z_i стану виходу $Y_i \in Y$. Обидві функції можна подати у виді 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 в звичайних випадках містить 2^n стовпців, де n – число вхідних перемінних X . Число рядків дорівнює числу станів схеми.

У табл. 3.1 наведений приклад переходів повного синхронного ПП, функції якого визначені для всіх наборів X и Z . Як видно, 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 клітці таблиці переходів зазначений номер внутрішнього стану, у яке пристрій повинний перейти у 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.

Надалі з метою спрощення замість Z_i у клітках таблиць будемо вказувати лише номер стану. Якщо в ПП який-небудь стан не визначений, тобто він є невикористаним чи забороненим, то у відповідній клітці таблиці ставлять прочерк (табл. 3.2).

У таблиці виходів (табл. 3.3) задається відповідність між станом виходу і внутрішнім станом. Якщо обидві таблиці сполучити, одержимо так називану таблицю станів, що задає одночасно як функцію переходів, так і функцію виходів. У кожній клітці такої таблиці (табл. 3.4) записані значення Z_i і Y_j .

Таблиця 3.1

	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 3.2

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

Таблиця 3.3

	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 3.4

	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	$Z_4; Y_4$	$Z_1; Y_1$	$Z_3; Y_3$	$Z_4; Y_4$

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 таблицю 3.5.

Таблиця 3.5

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

Якщо ПП знаходиться в стійкому стані (2) і стан його входу змінюється з X_2 на X_3 , то внутрішній стан при цьому не зміниться. У випадку, коли відбувається зміна X_2 на X_1 пристрій спочатку прийме нестійкий стан 1, а потім стійкий (1), при цьому його внутрішній стан зміниться з Z_2 на Z_1 .

Таким чином, зміна внутрішнього стану ПП завжди зв'язано з переходом його через нестійкий стан; перегін з одного внутрішнього стану в інший через стійкий стан неможливий.

Мінімізація числа станів. При побудові таблиць переходів може виявитися більше станів, чим необхідно для нормального функціонування ПП. У зв'язку з цим виникає задача мінімізації числа станів. При скороченні числа станів надлишковими станами вважають такі, котрим у таблиці переходів відповідають рядки з несуперечливим розміщенням цифр.

Такі рядки в таблиці переходів мають однакові цифри в якому-небудь стовпці, чи в одному рядку цього ж стовпця стоїть цифра, а в іншому маємо прочерк. Ці рядки називають сумісними. Об'єднання сумісних рядків приводить до зменшення числа внутрішніх перемінних, необхідних для опису всіх переходів. При об'єднанні рядків кожній групі сумісних рядків присвоюють нову цифру і роблять перепозначення станів ПП. Помітимо, що при виконанні операції об'єднання рядків значення вихідних функцій в увагу не приймаються. Отриману після об'єднання рядків нову таблицю називають скороченою таблицею переходів. Для оптимізації операції об'єднання рядків будують відповідну діаграму.

Для прикладу візьмемо таблицю 3.6. При побудові діаграми розмістимо номери рядків по деякому колу (рис.3.2).

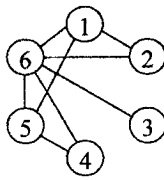


Рисунок 3.2 – Діаграма об'єднання рядків

Можливість попарного об'єднання окремих рядків умовно відзначимо лініями між відповідними номерами. Кожен рядок необхідно порівняти з усіма іншими для виявлення можливості їхнього об'єднання. Аналіз діаграми дозволяє виявити оптимальну схему об'єднання рядків, при якій мінімізується загальне число рядків у таблиці. Наприклад, поєднуючи рядки 1 і 2, 3 і 6, а також 4 і 5, одержимо скорочену таблицю переходів (табл. 3.7).

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

Кодування значень внутрішніх перемінних полягає в присвоєнні двійкового коду кожному стану ПП, тобто кожному рядку скороченої таблиці, або кожному набору значень перемінних Z_i у даний момент часу. Так, для реалізації чотирьох станів ПП будуть потрібні дві внутрішні перемінні (позначимо їх Z_1 і Z_2). Задачу кодування внутрішніх перемінних зручно вирішувати за допомогою спеціальної карти кодування (табл. 3.8), де код стану збігається з двійковим номером відповідної клітки. Розташовуючи, наприклад, стан 1 послідовно у кожній клітці і всякий раз 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 кодом станів відповідно до обраного варіанта кодування, запишемо кодовану таблицю переходів (табл. 3.9).

Послідовнісні пристрої можуть бути подані діаграмою чи графом, що складається з вузлів, з'єднаних галузями. Позначивши стан ПП вузлами, а переходи ПП, одержані під впливами вхідних сигналів X_i , галузями, будемо мати діаграма

му станів. Так, діаграма станів, що відповідає таблиці станів (див. табл. 3.8), наведена на рисунку 3.3. На галузях графа можуть бути зазначені значення вихідних сигналів.

Таблиця 3.6

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 3.7

X_1X_2	00	01	11	10
	(1)	-	3	(1)
	1	(3)	(3)	4
	-	3	(4)	(4)

Таблиця 3.8

z_1	0	1
0	2	3
1	1	4

Таблиця 3.9

x	0	1	z_1z_2
1	(1)	2	10
2	3	(2)	00
3	(3)	4	01
4	1	(4)	11

Діаграма станів ПП може бути використана для визначення виду вихідних перемінних при довільних входних перемінних для будь-якого початкового стану пристрою. Наприклад, для ПП, поданого графом на рисунку 3.3, якщо має місце перший початковий стан, а входна послідовність $X=00101$, то утвориться послідовність станів виду 22434 при послідовності на виході $Y=11010$. Та ж входна послідовність, що впливає на ПП при третьому початковому стані, приводить до вихідної послідовності виду 01010.

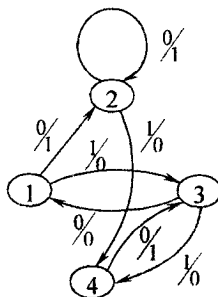


Рисунок 3.3 – Діаграма станів

Порівнюючи табличний і графічний способи завдання ПП, можна відзначити наступне.

Табличний спосіб завдання є найбільш повним, але не прийнятним для великого числа станів. У той же час даний спосіб дозволяє здійснити порівняно простий перехід до формульного запису функцій Y і Z .

Графічний спосіб більш наочний. Особливо його переваги відчуються при рішенні 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 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 являється і внутрішньою і вихідною перемінною, замість Z використовується позначення Q .

Функції переходів і виходів даного ПП мають вид:

$$\begin{aligned} Q^n &= f(x_1^n, x_2^n, Q^{n-1}), \\ Y^n &= \varphi(x_1^n, x_2^n, Q^{n-1}) \end{aligned}$$

Суматор виконує операцію складання x_1 і x_2 з урахуванням переносу Q , то-му $Q^n = 1$ 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. 3.10). 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 кліт-

ках, 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 .

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. 3.9). Доповнюючи табл. 3.10 стовпцем Q , у якому записується прийняте кодування станів ПП, перейдемо до табл. 3.11. Графи переходів такого пристрою, побудовані відповідно до таблиць 3.10 і 3.11, наведені на рисунку 3.4 (рис. 3.4,а - за табл. 3.10, а рис. 3.4,б - за табл. 3.11). За допомогою цих таблиць можна скласти і таблицю істинності суматора (табл. 3.12).

Таблиця 3.10

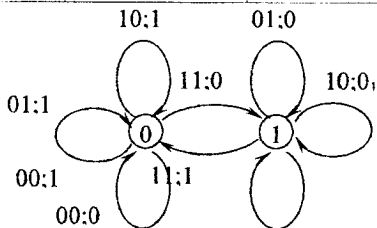
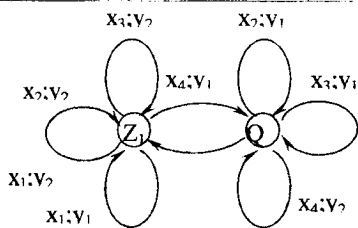
$Z_i^n \backslash X_j^n$	X_1^n	X_2^n	X_3^n	X_4^n
Z_1^n	Z_1, Y_1	Z_1, Y_2	Z_1, Y_2	Z_2, Y_1
Z_2^n	Z_1, Y_2	Z_2, Y_1	Z_2, Y_1	Z_2, Y_2

Таблиця 3.11

$i \backslash X_1, X_2$	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 3.12

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



а – з позначенням станів; б – з кодуванням станів

Рисунок 3.4 – Графи переходів суматора

3.2 Тригери

Основною властивістю тригерів, що уявляють 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 і JK -тригерів, які використовуються при побудові ключів, переривачів і т. ін.

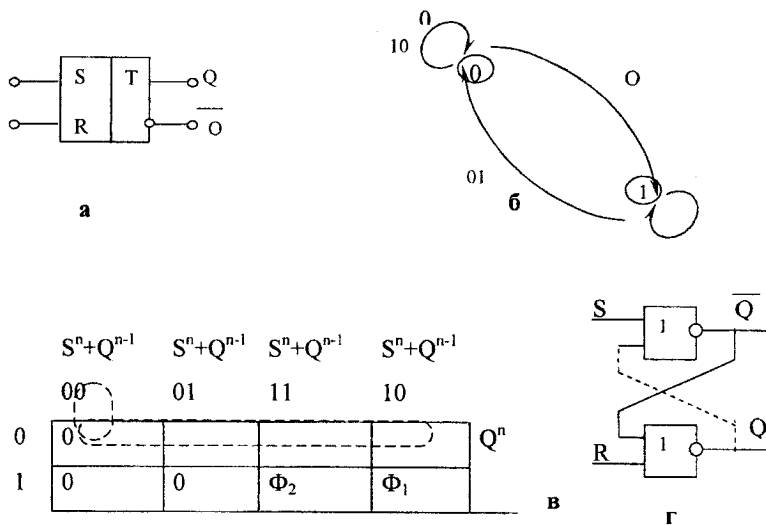
Асинхронні RS – тригери. Функціонування асинхронного RS – тригера, умовне позначення якого показано на рисунку 3.5, а, має такий словесний опис: якщо сигнали на входах $R^n = S^n = 0$, то тригер не змінює свого стану ($Q^n = Q^{n-1}$); якщо $R^n = 0$, а $S^n = 1$ (назва входу S походить від англійського слова "SET" - установка), то тригер переключиться в стан 1 $Q^n = 1$, якщо $R = 1$ (R відповідає слову "RESET" - скидання), $S^n=0$, тригер переключиться в стан 0. Одночасна поява "1" на обох входах, ($R^n = S^n=1$), повинна бути виключена. Правила роботи можна подати графом (рис. 3.5, б) 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 (табл. 3.13), складають карту Карно (рис.3.5,в), з якої одержують структурну формулу для Q^n :

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

Таблиця 3.13

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

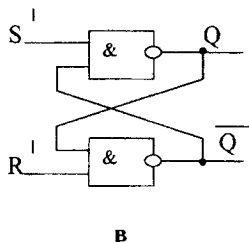
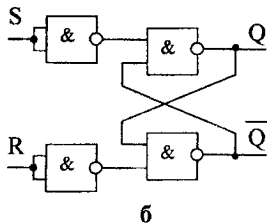
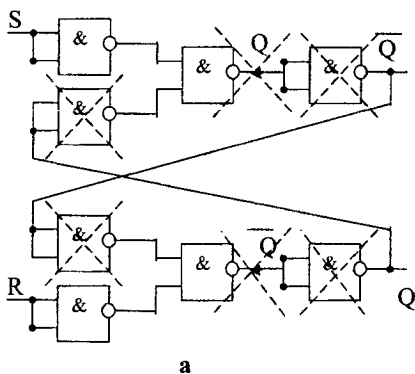


а – умовне зображення; б – граф роботи; в – карта Карно; г – структурна схема

Рисунок 3.5 - Умовне позначення RS-тригера, його граф, схема структурна

Схема тригера, який функціонує за цією формулою, подана на рис. 3.5, г, де враховане, що Q^{n-1} , і Q^n є сигналами в одній точці схеми в різні моменти часу (штрихова лінія на рис.3.5, г). При наявності на обох входах пристрою однакових перемикаючих сигналів R^n і S^n стан виходів з рівною імовірністю може стати як одиничним, так і нульовим. Оскільки ця невизначеність неприйнятна для тригера, то така комбінація є забороненою

Використовуючи елементи І-НІ, одержимо схему, зображену на рисунку 3.6, а. Виключивши зайві елементи, перекреслені пунктирними лініями, прийдемо до RS-тригера (рис. 3.6,б) з тією ж таблицею перемикування. Усунувши із схеми вхідні інвертори, одержимо RS-тригер з нульовими перемикаючими сигналами (рис. 3.6,в). Переходи цього RS-тригера показані в таблиці 3.14.



- а – схема, складена за формулою;
 б – схема з одиничними сигналами перемикання;
 в – схема з нульовими сигналами перемикання

Рисунок 3.6 – Синтез RS-тригера на IC I-II

Таблиця 3.14

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 3.15

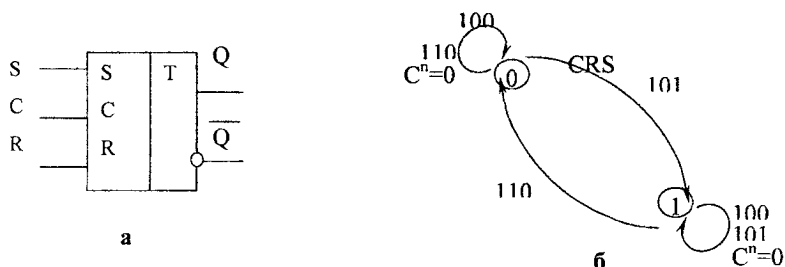
R'	S'	Q^n
1	1	Q^{n-1}
1	0	1
0	1	0
0	0	-

Таблиця 3.16

C^n	R^n	S^n	R^m	S^m
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

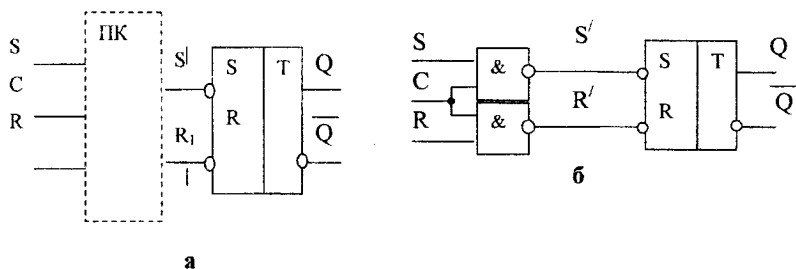
Синхронні RS-тригери. Синхронний RS-тригер змінює свій стан, при зміні керуючих сигналів лише при наявності сигналу на вході синхронізації C . При $C = 0$ стан тригера не змінюється. Функціональне позначення RS-тригера і граф його роботи показані на рисунку 3.7.

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 виді послідовно з'єднаних пристроїв керування (ПК) та елемента пам'яті - RS-тригера (рис. 3.8,а).



а – умовне позначення; б – граф роботи

Рисунок 3.7 – Синхронний RS – тригер



а – блок-схема; б – структурна схема

Рисунок 3.8 – Побудова синхронного тригера

Таблиця істинності для входів елемента пам'яті R' і S' (табл. 3.16) складена з урахуванням словесного опису роботи тригера. Оскільки таблиця містить значне число одиниць для функцій R' і S' , то карти Карно можна скласти для зворотних функцій

Одержана завдяки цьому схема наведена на рис. 3.9.

		$R^n S^n$				
		00	01	11	10	
C^n	0		Φ_1		$S^{1n} = S^n C^n$	
	1		Φ_3			
C^n	0		Φ_2		$R^{1n} = R^n C^n$	
	1		Φ_4	1		

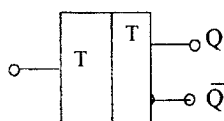
Рисунок 3.9 – Карта Карно синхронного RS – тригера

T-тригери

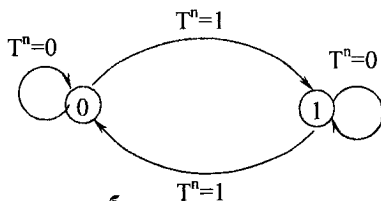
Рахунковий, або T - тригер (рис 3.10,а) має один інформаційний вхід ("Toqgle" - перемикач). Цей тригер переключється в протилежний стан із приходом кожного перемикаючого сигналу на вхід T . Таблиця переключень такого пристрою (табл. 3.17) містить два рядки.

Таблиця 3.17

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



а

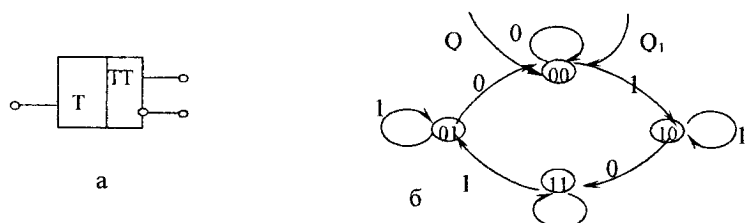


б

а - умовне позначення; б - граф роботи
Рисунок. 3.10 – Одноступеневий T – тригер

Якщо розглянути граф переходів T - тригера на одному елементі пам'яті (рис. 3.10,б), то стає очевидним, що він буде нестійким, тому що 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 двоступінчастих тригерів, або організацією динамічного керування. У двоступінчастому T - тригері (рис. 3.11) 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 і $SLAVE$ - раб (у російській інтерпретації: система "ведучий-ведений").



а - умовне позначення; б - граф роботи

Рисунок 3.11 – Двоступеневий T -тригер

Проведемо синтез такої схеми T -тригера, вважаючи, що в його структуру входять два RS -тригери: основний Q і допоміжний Q_1 . Задача синтезу зводиться до визначення функцій керування елементами пам'яті (RS -тригерів). За допомогою графа переключень складатиметься таблиця істинності (табл. 19) для вхідних сигналів RS - тригерів. При цьому там, де можливо, ставиться знак Φ , щоб по-можливіше використовувати можливості процедури мінімізації. Потім за картами Карно (рис. 3.12), отриманим для інверсних функцій, визначаються структурні формули, що дозволяють побудувати схему двоступінчастого T - тригера (рис. 3.13). Вона включає два RS - тригери і додаткові елементи, що відповідають отриманим структурним формулам.

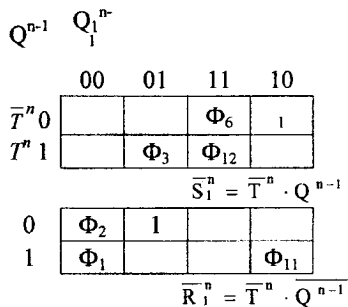
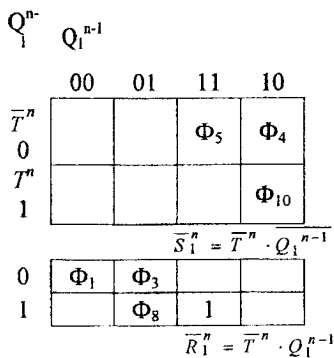
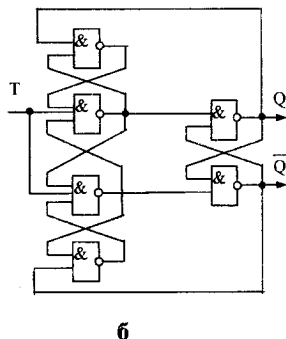
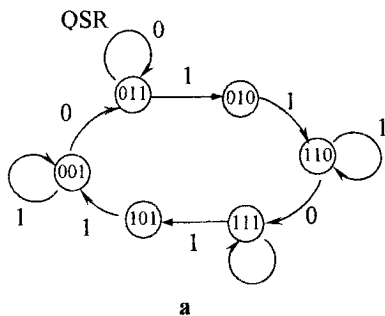


Рисунок 3.12 – Карти Карно двоступінчастого Т-тригера

У тригерах з динамічним керуванням стійкість роботи забезпечується тим, що процес зміни вхідного сигналу Т запам'ятовується на кожному вході основного RS – тригера. Граф такого Т - тригера і варіант симетричної схеми наведені на рисунку 3.13.



а - граф роботи; б - симетрична схема
Рисунок 3.13 – Синтез двоступеневого Т-тригера

D – тригери

Синхронний D - тригер (рис. 3.14) має два входи: даних D і синхронізації С. Такий тригер переходить у стан, що вказується сигналом на вході D, тільки з появою перемикаючого сигналу на вході С.

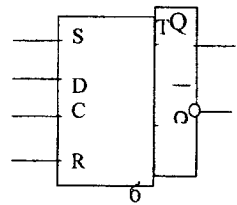
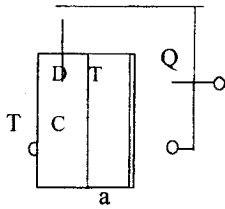
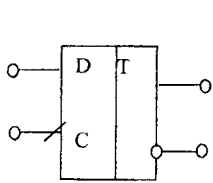


Рисунок 3.14 – Умовне зображення синхронного D-тригера

а - T-тригер; б - DRS-тригер на базі D-тригера
Рисунок. 3.15 – Варіанти D-тригерів

Такий тригер здійснює затримку сигналу за входом D до появи сигналу на вході C , оскільки його вихідний сигнал Q^n повторює значення сигналу D на попередньому такті (D^{n-1}) і зберігає цей стан до наступного такту. Таким чином, D -тригер здійснює затримку на один такт сигналу, що надходить до входу D . Такий D -тригер часто називають тригером затримки (від англійського слова *DELAY* - затримка).

Порівнюючи таблиці переключення D -тригера (табл. 3.18) і T-тригера (табл. 3.17), можна зробити висновок, що при $T = C$ і $D = Q$ (рис. 3.15,а) D -тригер буде функціонувати як T-тригер.

Доповнений установчими R і S - входами D тригер являє собою DRS-тригер (рис. 3.15,б), стан якого визначається, в першу чергу, сигналами на установчих входах і при $R^n = S^n = 1$ залежить від сигналів на входах D і C . Комбінація $S^n = R^n = 0$, як і для RS -тригера, є забороненою.

Таблиця 3.18

C	Такт n		Такт $n+1$
	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

Модифікацією D – тригерів є DV – тригери із додатковим входом V (від VALVE - клапан), який грає роль дозволяючого щодо відношення 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 і C. Тим самим наявність входу V дозволяє в необхідні моменти часу зберігати інформацію на виходах протягом заданого числа тактів, що розширює функціональні можливості пристрою.

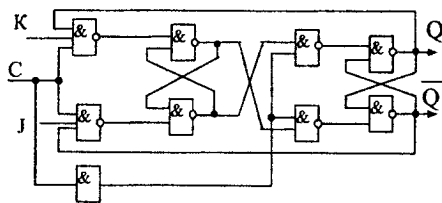
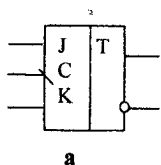
JK – тригери

Ці тригери (рис. 3.16,а) є універсальними синхронними тригерами, в яких входи J і K відповідають входам S і R у RS - тригері. Однак на відміну від RS - тригера набір $J^n = K^n = 1$ вважається припустимим і приводить до зміни стану тригера ($Q^n = Q^{n+1}$). Оскільки JK - тригери, реалізовані у виді IC, виконуються синхронними, їхні вихідні рівні встановлюються тільки при надходженні на вхід перемикаючих сигналів. Стани JK - тригера наведені в таблиці 3.19. Складаючи карту Карно для Q^{n+1} (рис. 3.17), 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 і цілком описує його роботу.

Приклад схеми JK - тригера, що має структуру T - тригера, побудованого за двоступінчастою системою, приведений на рис. 3.16,б. Шляхом визначеного з'єднання виводів JK - тригер може бути переведений у режим роботи інших типів тригерів. Так, при $J = K = 1$ і при наявності 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, і схеми I по входам J і K. Позначення такого JKRS - тригера показано на рис. 3.18.



а – умовне зображення; б – структурна схема

Рисунок 3.16 –JK – тригер

Таблиця 3.19

J^n	K^n	Q^{n+1}
0	0	Q^n
0	1	0
1	0	1
1	1	Q^n

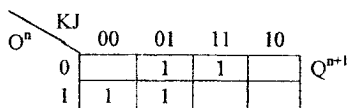


Рисунок 3.17 –Карта Карно JK – тригера

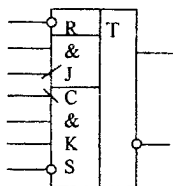


Рисунок 3.18 – Умовне зображення JKRS - тригера

3.3 Канонічний метод проектування ПП

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 наступний порядок дій.

1 Визначити входні і вихідні сигнали проектованого ПП і побудувати його структурну схему, розміщуючи входи ліворуч, а виходи праворуч (проектвані

1 Визначити вхідні і вихідні сигнали проєктованого ПП і побудувати його структурну схему, розміщуючи 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 – слідчі співвідношення між вхідними і вихідними сигналами.

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

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

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

6 Якщо на першому етапі проєктування був отриманий первинний граф переходів, то за цим графом необхідно скласти таблицю переходів.

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

8 На наступному етапі проєктування будують карти станів і знаходять функції порушення внутрішніх перемінних. Для цього привласнюють рядкам кодової таблиці переходів замість десяткових цифр двійковий (рис. 3.19).

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

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

а – десятичний код; б – двійковий код

Рисунок 3.19 – Карти Карно с кодованими переходами

9 Складаючи карти вихідних перемінних, якщо вони не збігаються з внутрішніми перемінними, знаходимо функції вихідних перемінних.

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

Карта станів Q тригера (рис. 3.20) перетвориться в таблицю функцій переходів F_Q (табл. 3.20) внутрішніх перемінних, де F_Q є перехід $Q^n \rightarrow Q^{n+1}$, причому умовні позначки переходу внутрішніх перемінних з $0 \rightarrow 0$; $1 \rightarrow 1$; $1 \rightarrow 0$ і $0 \rightarrow 1$, відповідно: 0, 1, Δ .

Таблиця 3.20

F_Q	S	R
0	1	X
1	X	1
Δ	0	1
Δ	1	0

Потім складаємо карту функцій порушення кожного тригера. Для цього символи функцій переходів у карті функцій переходів заміняють значеннями вхідних сигналів тригера, (наприклад \bar{R} і \bar{S}), які необхідні для реалізації даного переходу і визначаються по словнику переходів обраного тригера (рис. 3.21). Отримані карти подані в табл. 3.21.

Таблиця 3.21

Внутрішній стан	Стан входів 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

Далі, визначивши за картами функції збудження, треба провести їх перетворення для реалізації в заданому базисі, наприклад, у базисі І-НІ.

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

Отримані формули дозволяють побудувати структурну схему (рис 3.23, а); Оскільки карти функцій порушення (див.табл. 3.21) містять невизначені значення цих функцій, мається можливість скласти ряд інших варіантів структурної реалізації синтезованого ПП, провести їхній аналіз і вибрати кращий з огляду на функціональну надійність, простоту реалізації і т.і. У цьому полягає перевага словникового методу структурного синтезу.

Проілюструємо сказане одним із прикладів. Довизначимо карти функцій \bar{S}_2 і \bar{R}_2 (див.рис. 3.21), прийнявши значення функції \bar{S}_2 в клітках із двійковими номерами 001 і 001 і значення функції \bar{R}_2 в клітках 010 і 110, які дорівнюють нулю. Тоді:

$$\begin{aligned}\bar{S}_2 &= Q_1 + x \cdot \bar{Q}_2 = Q_1 + R_1 = \overline{\bar{Q}_1 \cdot \bar{R}_1}; \\ \bar{R}_2 &= \bar{Q}_1 + x \cdot Q_2 = \bar{Q}_1 + S_1 = \overline{Q_1 \cdot \bar{S}_1}.\end{aligned}$$

x $Q_1 Q_2$	00	01	11	10
0	0Δ	01	1	10
Δ	00	Δ1	11	0

Рисунок 3.20 –Карта станів тригера

$Q_1 Q_2$	00	01	11	10
x				
0	1	1	X	X
1	1	0	X	1

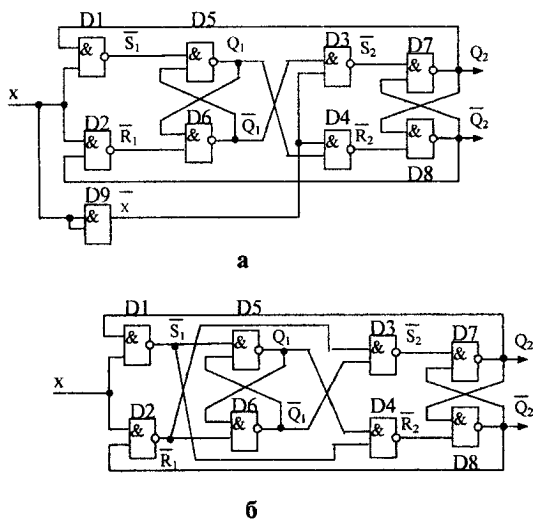
$Q_1 Q_2$	00	01	11	10
\bar{S}_1				
0	0	X	1	1
1	1	X	X	1

$Q_1 Q_2$	00	01	11	10
\bar{R}_1				
0	X	X	1	1
1	X	1	1	0

$Q_1 Q_2$	00	01	11	10
\bar{S}_2				
0	1	1	0	X
1	X	1	1	X

Рисунок 3.21 – Карти функцій тригера

У ПП, схема якого наведена на рисунку 3.22,а, можуть виникнути небезпечні змагання сигналів по рівнобіжних ланцюгах: входи $D9-\bar{X}$, $D1-D5-Q_1$ і $D2-D6-\bar{Q}_1$. Так, наприклад, у випадку, коли вхідна перемінна 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 і зміну вихідних перемінних Q_2 і \bar{Q}_2 . В другому варіанті ПП (рис. 3.23б) такі небезпечні змагання відсутні. Крім того, ПП містить на один логічний елемент менше, ніж аналогічний ПП, показаний на рисунку 3.22, а.



а – з можливістю змагань; б – з усуненням змагань

Рисунок 3.22 – Структурні варіанти тригера і варіант усунення значень

3.4 Синтез тригерних пристроїв

Найпростіші тригерні схеми, реалізовані у виді мікросхем [3, 4], дозволяють проектувати різноманітні цифрові пристрої. Зокрема, на їхній основі будуються лічильники і регістри. Зростаюча складність цифрових схем робить практично неможливим інтуїтивний підхід при синтезі і 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) і необхідні властивості проектованої тригерної схеми.

Метою синтезу є пошук рівняння виходу і 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 перемінних (Q), невідоме, то спочатку складають так названу первісну таблицю переходів і виходів, що містить можливе число внутрішніх стійких станів. Число стовпців первісної таблиці дорівнює числу різних станів входів тригерів. Для *D-тригера*, який має два входи C і D , треба чотири стовпця число рядків первісної таблиці можна визначити, якщо кожному можливому стану входів і 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 таблиці 3.22. Таким чином, кожен стовець таблиці 3.22 визначає стан вхідних перемінних, а кожен рядок – стан внутрішніх перемінних. Повний стійкий стан тригера буде умовно визначатися перетинанням рядка і стовпця таблиці. Цей стан записується цифрою в круглих дужках. У кожному рядку первісної таблиці мається лише один стійкий стан.

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

часться за спадом логічної "1" на синхронізуючому входу С.

Таблиця 3.22

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

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

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 мимовільно переходить у новий стійкий стан.

Розглянемо запис переходів. Припустимо, що $C = D = 1$, $Z = 0$. Ця комбінація відповідає повному стану (3) у таблиці 3.22. Допустимо, що вхідна перемінна C змінила своє значення на лог "0", тобто $CD = 01$. Оскільки перемикаючим сигналом C є зміна $1 \rightarrow 0$, то відповідно до таблиці зовнішніх переходів D -

тригер повинен змінити свій вихідний стан. Новим стійким станом повинен бути стан (6), що відповідає $C = 0, D = 1, Z = 1$. Щоб позначити цей перехід у таблиці 3.22, необхідно на перетинанні третього рядка і стовпця $CD = 01$ записати цифру 6 без дужок. Якщо ж перехід з внутрішнього стану (3) здійснюється зміною перемінної D , тобто $CD = 10$, то новим стійким станом повинен бути стан (4), оскільки на вході C стан не змінився. Так на етапі заповнення першої таблиці переходів враховують асинхронне поводження проектованого синхронного тригера, тобто його тип [1].

Нарешті, перехід від стійкого стану (3), при одночасній зміні перемінних C і D , повинен бути забороненим, оскільки в цьому випадку фізично неможливо забезпечити однозначність операцій.

Таким чином, у залежності від порядку зміни вхідних сигналів тригер може виявитися в стані 1 чи 0. У таблиці переходів заборонені переходи будемо позначати символом (х). Аналогічно заповнюють усю таблицю 3.22, переходять до таблиці 3.23. Очевидно, що при складанні таблиці всі основні властивості проектованого пристрою, сформульовані словесно, виражаються у виді першої таблиці переходів.

Таблиця 3.23

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 проектованого тригера залежить від числа внутрішніх перемінних, тому доцільно мінімізувати число внутрішніх станів, рівне числу рядків першої таблиці переходів. Під мінімізацією числа внутрішніх станів ПП розуміється процес, метою якого є одержання ПП, що має мінімальне число внутрішніх станів серед усіх ПП, що реалізують задані умови його роботи. Процес сполучення рядків таблиці переходів називають об'єднанням і виконують його у відповідності з наступними правилами [1]:

1) два рядки (чи більш) можуть бути об'єднані, якщо у відповідних стовпцях номери станів однакові чи на одному з рядків мається заборонений стан;

2) результуючий стан треба взяти у дужки, якщо один із поєднаних номерів в 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 і четвертий рядки таблиці 3.22. Їхнє об'єднання відповідно до даних правил дає новий рядок виду: (1); 2; 4.

Перехід від стану (4) до (1) у новому об'єднаному рядку здійснюється зміною тільки вхідної перемінної C без зміни внутрішнього стану.

У другому варіанті (табл.3.24) на відміну від першого переходи з одного рядка в інший здійснюються при зміні перемінної C разом зі зміною логічної перемінної D .

При проектуванні двоступінчастих тригерів доцільно використовувати об'єднання рядків первісної таблиці, що приводить до таблиці 3.23, а при побудові тригерів з динамічним керуванням – до таблиці 3.24.

Таблиця 3.24

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, і не ускладнити структуру логічного перетворювача.

3.5 Регістри

Основна функція регістра полягає в збереженні багаторозрядного двійкового числа. У звичайних випадках регістри мають регулярну структуру і складаються з однакових частин. Кожна з таких частин призначена для збереження

одного розряду двійкового числа. Розряд регістра містить елемент пам'яті, виконаний на тригері.

Регістр може працювати в режимах запису, збереження і 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 і зчитування чисел розрізняють наступні типи регістрів:

паралельні: запис і зчитування у всіх розрядів здійснюються одночасно;

послідовні (регістри зрушення): запис і зчитування даних здійснюється розрядами послідовно;

паралельно - послідовні: у регістрах цього типу запис здійснюється паралельно, а зчитування послідовно;

послідовно - паралельні: на відміну від попереднього типу запис робиться послідовно, а зчитування паралельно;

реверсивні регістри зрушення: напрямок зрушення чисел у них може бути змінений в залежності від значень керуючих сигналів.

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

Структура паралельного регістру (рис. 3.23,а) являє собою сукупність однотипних осередків (рис. 3.23,б), що містять тригер а також і схеми керування записом W_1 і зчитуванням W_2 . Кожен осередок має інформаційний вхід (X_i). Входи керування записом (C_1) і зчитуванням (C_2) є загальними для всіх осередків. Правило роботи для тригера в j -му осередку формулюється наступним образом. Якщо $C_1^n = 1$, $C_2^n = 0$, то $Q_j^n = X_j^n$; $Y_j^n = 0$. 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 не переключається ($Q_j^n = Q_j^{n-1}$) і вхідний сигнал проходить на вихід Y_j^n

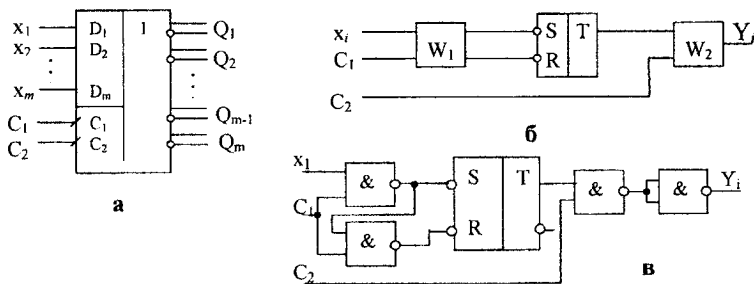
$= Q_j^n$. Комбінація $C_1^n = C_2^n = 1$ вважається забороненою. Вибравши елементом пам'яті RS -тригер і склавши таблиці істинності для вихідної функції Y_j^n і функцій порушення тригера, одержимо структурні формули:

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

$$S = \overline{x_i \cdot C_1};$$

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

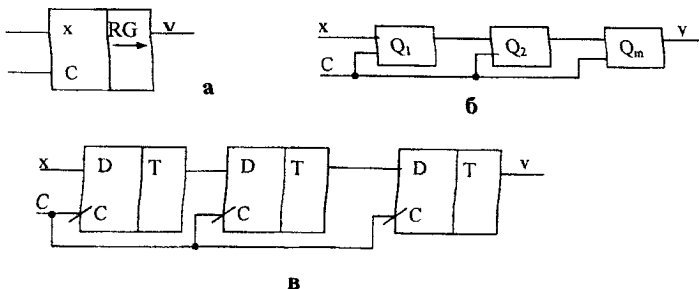
Відповідно до цих формул схема одного розряду регістра буде мати вид, показаний на рисунку 3.23,в.



а - умовне позначення паралельного регістру; б - структура типового осередку;
в - принципова схема

Рисунок 3.23 –Регістр паралельного типу

Послідовний регістр зрушення (рис. 3.24,а) має інформаційний і тактовий входи, що забезпечують запис числа і зрушення його по регістру. Структура регістра (рис. 3.24,б) уявляє собою послідовне з'єднання однотипних осередків. Правило роботи для j - го розряду можна записати в наступному виді: якщо $C^n = 0$, то $Q_{j-1}^n = Q_{j-1}^{n-1}$, а 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} .



а - умовне позначення послідовного регістру;
 б - структурна схема;
 в - функціональна схема

Рисунок. 3.24– Регістр послідовного зрушення

Отже, для побудови регістра зрушення можна вибрати D -тригери або JK -тригери, що працюють у режимі D -тригера. Відповідна схема регістра дана на рис. 3.24,в.

3.6 Лічильники

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 , що 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 виді:

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

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 і число станів лічильника відповідно дорівнює $K=2^m$, лічильник називають двійковим. 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 і можливість нарощування їхньої розрядності. У підсумовуючому лічильнику з послідовним переносом кожен вхідний імпульс збільшує значення двійкового числа, записаного в лічильник, на одиницю. Правило роботи підсумовуючого, наприклад трирозрядного лічильника може бути подано у виді таблиці переключень лічильника (табл. 3.25). Використовуючи таблицю переключень, можна визначити необхідний тип першого тригера, орієнтуючись на Q_1 , після чого виявити спосіб з'єднання тригерів. Очевидно, що першим повинен бути Т-тригер, тому що він перемикається кожним вхідним сигналом. Оскільки з таблиці 3.25 випливає, що другий тригер змінює свій стан, коли перший тригер переходить у рівень 0, то необхідно подати сигнал переключення на вхід другого тригера з інверсного виходу першого. Аналогічні умови роботи будуть у третього тригера. Схема лічильника наведена на рис. 3.25.

Таблиця 3.25

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
0	0	0	8

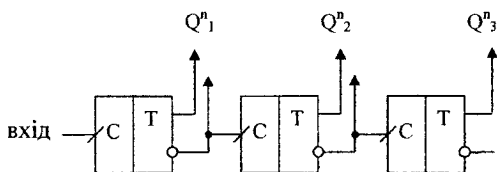


Рисунок 3.25 – Схема організації трирозрядного лічильника

У піднімаючого лічильника, з кожним вхідним імпульсом значення двоїчного числа, записаного в лічильник, зменшується. Правила його роботи 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 відповідно до правила роботи таблицю переключень лічильника (табл. 3.26), можна тим же способом визначити його схему (рис. 3.26).

Реверсивний лічильник може працювати як підсумовуючий і піднімаючий. Вибір режиму роботи здійснюється за допомогою керуючого сигналу, що надходить на вхід F лічильника (рис.3.27). Для побудови реверсивного лічильника можна скористатися, як і раніше, послідовним з'єднанням T тригерів, доповнивши його елементами комутації. Елемент комутації повинен сформувати сигнали: в режимі підсумовування $T_i = \overline{Q_{i-1}}$, а в режимі віднімання $T_i = Q_{i-1}$. Варто врахувати, що сигнал на вході F може змінюватися тільки при вхідному сигналі лічильника $X=0$. Складена по словесному опису таблиця істинності для вхідного сигналу i -го тригера (табл. 3.27) дозволяє записати для T_i структурну формулу

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

і розробити структурну схему елемента комутації (рис. 3.27).

Таблиця 3.26

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

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

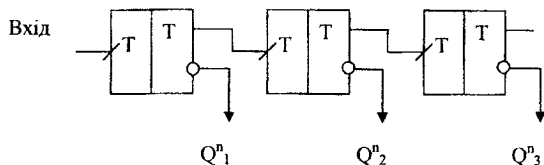


Рисунок 3.26 – Схема організації віднімаючого лічильника

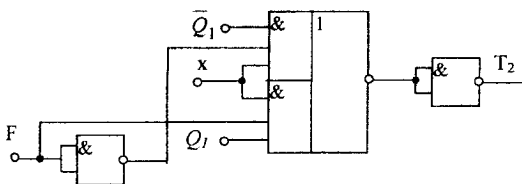


Рисунок 3.27 – Схема організації реверсивного лічильника

4 ЦИФРОВІ ПРИСТРОЇ ФОРМУВАННЯ ТА ОБРОБКИ ДИСКРЕТНИХ СИГНАЛІВ

Цифрові пристрої формування та обробки дискретних сигналів використовуються для формування і перетворення різного класу сигналів на малому рівні потужності, зокрема, для синтезу форм огинаючої. Розглянемо приклад реалізації цифрового формування гармонічного сигналу коливання в збудниках радіопередавача, у кореляторах і погоджених фільтрах, у системах фазового автопідстроювання частоти.

4.1 Цифрові пристрої формування сигналів

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

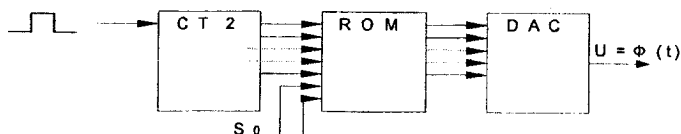


Рисунок 4.1 - Структурна схема функціонального генератора

Розглянемо принципову схему такого генератора (рис.4.2). У ПЗП D3... D5 записуються в додатковому коді з інвертованим знаковим розрядом 12-розрядні кодові слова, що відповідають вибіркоким значенням функції. З виходів ПЗП коди надходять на входи ЦАП (D6). Зчитуванням інформації з ПЗП керують двоїчні лічильники D1 D2. Вхідна послідовність u_3 надходить безпосередньо на вхід лічильника D1. Обсяг пам'яті ПЗП, необхідний для формування, наприклад 256 вибіркоких значень сигналу $s(t)$ при $x=8$ складає 256×12 біт.

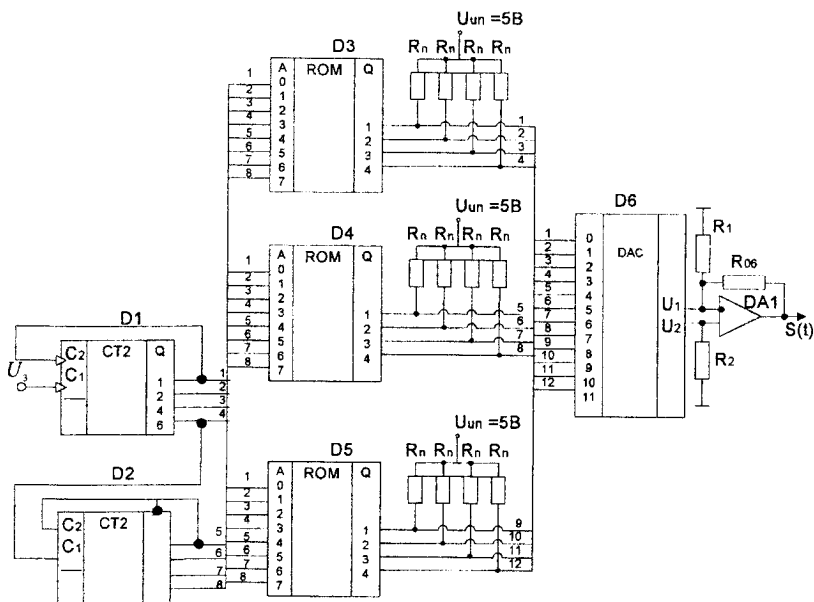


Рисунок 4.2 - Принципова схема формувача гармонічного сигналу

Максимальна частота вихідного сигналу $S(t)$ обмежується швидкодією застосованого цифроаналогового перетворювача Д6.

4.2 Цифрові корелятори

Розглянемо алгоритм когерентної обробки сигналів цифровими кореляторами обчислення U_0 у формі:

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

де $A_x^{(n)}[k]$ і $A^{(p)}[k]$ - послідовності чисел, які еквівалентні для оброблюваного коливання $x(t)$ і прийнятого сигналу $s_0(t)$; n, p - розрядності розглянутих чисел; q - кількість розрядів числа на виході цифрового корелятора. Структурна схема цифрового корелятора наведена на рисунку 4.3, де ФНЦ - фільтр нижчих частот ЦП - цифровий перемножник n - розрядних і p - розрядних чисел у деякі фіксовані моменти часу $K \cdot \Delta t$; ЦУ - цифровий пристрій формування послідовності чисел $A^{(p)}[k]$; СН - цифровий сумматор-накопичувач $N(n+p)$ - розрядних чисел.

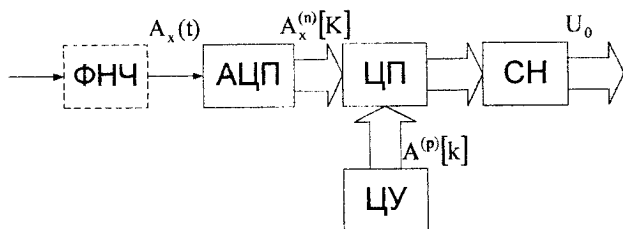


Рисунок 4.3 – Структурна схема цифрового корелятора

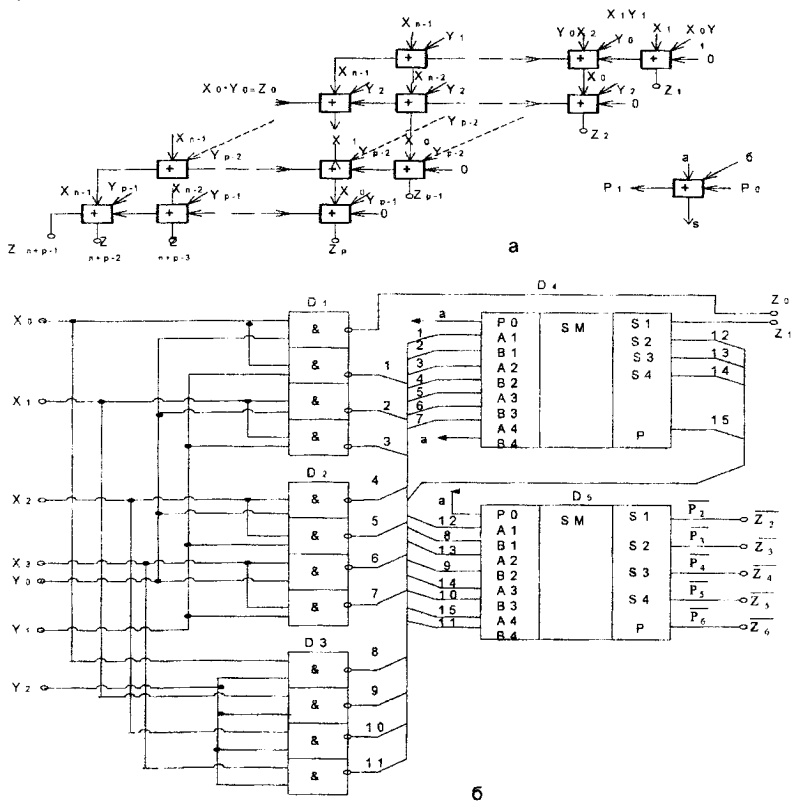
Зупинимося докладніше на принципах побудови ЦП. Найбільш швидкодіючими є цифрові перемножники рівнобіжного типу. Такі ЦП розділяються на матричні перемножники і перемножники на основі ПЗП.

Матричні перемножники (рис.4.4, а) здійснюють операцію перемноження n - розрядних і p - розрядних чисел $X^{(n)}$ і $Y^{(p)}$ у відповідності з наступним алгоритмом 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_j і z_k - значення i -го, j -го і k -го розрядів чисел $X^{(n)}, Y^{(p)}$ і $Z^{(n+p)}$, записаних у двоїчному коді. При цьому кожне число $X^{(n)}$ подається у виді n -

розрядного числа $(x_{n-1}, x_{n-2}, \dots, x_0)$; число $Y^{(p)}$ - у виді p - розрядного числа $(y_{p-1}, y_{p-2}, \dots, y_0)$ і число $Z^{(n+p)}$ у виді $(n+p)$ - розрядного числа; $(z_{n+p-1}, z_{n+p-2}, \dots, z_0)$ причому старшими є розряди x_{n-1}, y_{p-1} і z_{n+p-1} молодшими - x_0, y_0, z_0 . На входи двійкових суматорів надходять результати логічного перемножування розрядів позитивних чисел $X^{(n)}$ і $Y^{(p)}$. Двійковий суматор має входи "а" і "б", вхід переносу числа "р₀", вихід суми чисел "s" і вихід переносу числа "р₁". Кожен рядок суматорів (рис. 4.3) формує часткову суму і передає її наступному рядку. Молодший розряд числа $Z^{(n+p)}$ утвориться окремо, як результат логічного перемножування розрядів x_0 і y_0



а – структурна схема;
б – принципова схема

Рисунок 4.4 – Цифрові перемножники

Принципова схема матричного перемножування, призначена для перемножування 4-розрядних $X^{(4)}$ і 3-розрядних чисел $Y^{(3)}$ приведена на рис.4.4,б. Для логічного перемножування розрядів використовуються елементи D1...D3 (IC 564 LA7), а в якості суматорів D4 і D5 застосовуються 4-розрядні повні суматори (IC 564 IM1). Розряди результуючого числа $Z^{(7)}$ формуються в інвертованому виді. На входи "а" сумматорів D4 і D5 подається напруга логічної "1".

Швидкодія такого матричного перемножника визначається швидкодією елементів, що входять у пристрій, і для його оцінки необхідно знати самий тривалий шлях проходження сигналу по перемножнику. При використанні зазначених типів IC максимальна частота зчитування чисел з виходу перемножника не перевищує 1 МГц.

Перемножники на основі постійних запам'ятовуючих пристроїв містять ПЗП, у якому зберігаються таблиці добутків різних варіантів співмножників, причому загальне число добутків дорівнює $2^{n+p}(n+p)$ - розрядних чисел. Основу ПЗП складають матриці комірок пам'яті з записаними в них $(n+p)$ - розрядними числами, наприклад, шляхом пропалювання відповідних перемичок у матриці (IC 556PT4, 556PT5). Звертання до збереженої інформації виробляється 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 і, як правило, вона менше, ніж у матричних перемножників.

4.3 Цифрові погодженні фільтри (ЦПФ)

Реалізація алгоритму погодженої фільтрації може бути виконана шляхом безпосереднього обчислення вираження дискретної згортки. Розглянемо один з можливих варіантів побудови ЦПФ. Алгоритм дискретної згортки в цьому випадку може бути записаний у виді

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

де $K^{(p)}[i]$ - послідовності p -розрядних чисел, що є цифровими еквівалентами гратчастої функції $K[i]$ вибіркових значень з імпульсного відклику $K(t)$ погодженого фільтра.

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

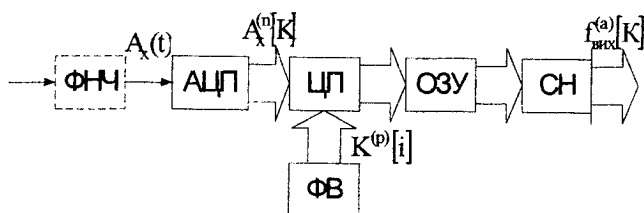


Рисунок 4.5 – Структурна схема цифрового погодженого фільтра

Перший етап полягає в організації перемножування кожного p - розрядного числа $A_x^{(n)}[k-i]$ на всі N значень p -розрядних чисел $K^{(p)}[i]$ і запису отриманих $N(p+p)$ -розрядних чисел добутків в ОЗП, що уявляє собою куб пам'яті (рис. 4.6) розмірами $N \times N \times (n+p)$, у який порядково (по N чисел) записуються добутки $A_x^{(n)}[k-i]K^{(p)}[i]$. Другий етап складається в зчитуванні інформації з ОЗП і підсумовуванні за допомогою цифрового сумматора-накопичувача СН $N(p+p)$ -розрядних чисел для кожного значення K функції $S_{\text{вих}}^{(q)}[k]$.

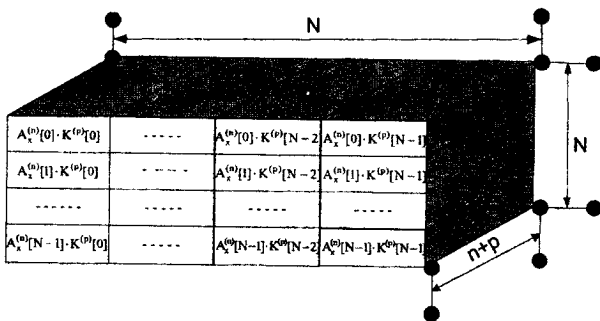


Рисунок 4.6 -- Куб пам'яті ЦПФ

Усі елементи куба являють собою доданки, що входять у розглянуте вище вираження згортки, і для одержання $s_{\text{вих}}^{(q)}[k]$. необхідно організувати діагональний режим зчитування чисел з ОЗП.

При $k=0$:

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

при $k=1$:

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

при $k=N-2$:

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

при $k=N-1$:

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

Дійсно, при $k=0$ значення чисел $A_x^{(n)}[-1], A_x^{(n)}[-2], \dots, A_x^{(n)}[-(N-1)]$ у випадку існування $A_x(t)$ на інтервалі аналізу дорівнюють нулю. У той же час, якщо $A_x(t)$ являє собою безупинний сигнал на інтервалі $-\infty < t < +\infty$, то значення чисел $A_x^{(n)}[-1], A_x^{(n)}[2], \dots, A_x^{(n)}[-(N-1)]$ відповідають значенням функцій $A_x[-1], A_x[-2], \dots, A_x[-(N-1)]$ вибірових значень $A_x[-\Delta t], A_x[-2\Delta t], \dots, A_x[-(N-1)\Delta t]$ на інтервалі аналізу, так що

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

Підставивши це вираження в попереднє, одержимо діагональний закон зчитування і підсумовування чисел з ОЗП, причому очевидно, що для організації обробки $A_x(t)$ на інтервалі $-\infty < t < +\infty$ необхідно чергувати етапи зчитування і підсумовування з етапами запису чисел. При цьому, після k -го діагонального зчитування і підсумовування чисел з ОЗП необхідно зробити запис N чисел наступних добутоків у k -й рядок куба пам'яті. Процес повторюється циклічно через N тактів (зчитування, підсумовування, запис). 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 утворюють таблицю істинності для функції алгебри логіки, яка описує роботи проєктованої логічної схеми;
- на підставі таблиці істинності утворюють математичну формулу синтезуючої 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 у відповідності до такої часової діаграми (рис. 4.7).

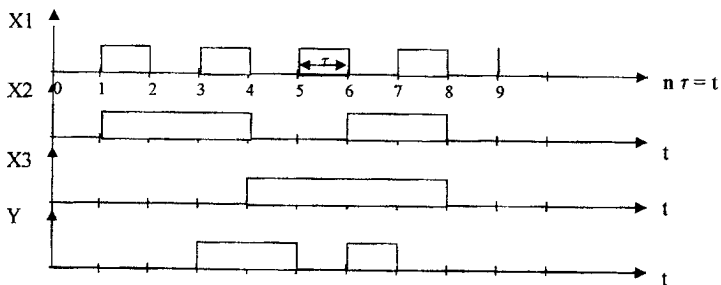


Рисунок 4.7 – Часова діаграма роботи пристрою

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

Таблиця 4.1

n	X1	X2	X3	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

Запишемо 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).

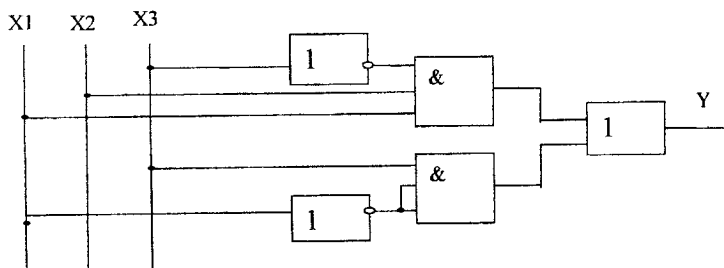


Рисунок 4.8 – Електрична принципова схема пристрою до прикладу 1

Приклад 2. Синтезувати у базисі Шефера схему, яка реалізує таку 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 алгебри логіки у ДНФ.

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

Мінімізуємо отриманий математичний вираз за допомогою карти Карно (рис. 4.9).

		$X_1 X_2$			
		00	01	11	10
$X_3 X_4$	00	1	1	1	1
	01	1	1	1	1
	11				
	10	1		1	

Рисунок 4.9 – Карта Карно

У результаті мінімізації отримаємо

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

Використовуючи правила де Моргана маємо

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

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

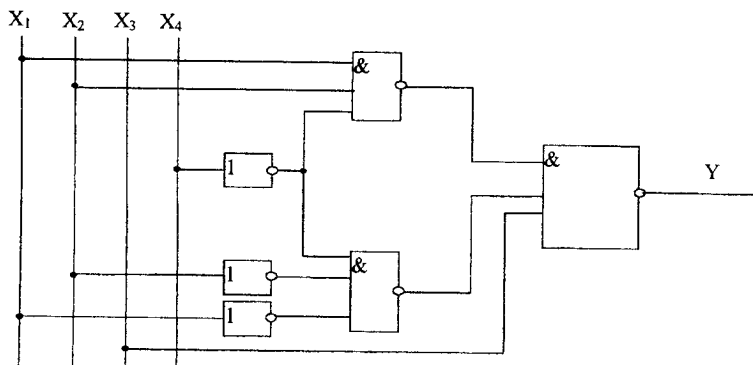


Рисунок 4.10 – Електрична принципова схема до прикладу 2 у ДНФ

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

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

Відповідно до цього виразу утворюємо електричну схему пристрою (рис. 4.11).

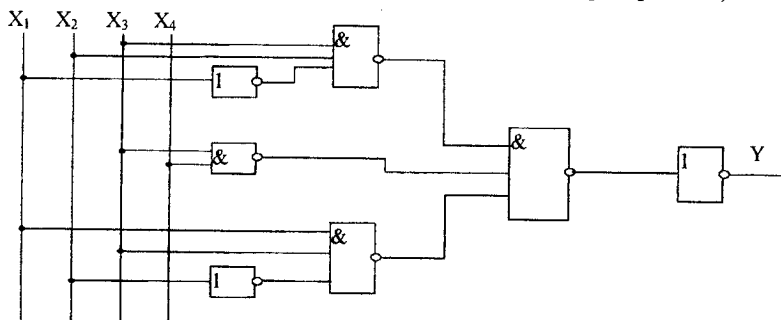


Рисунок 4.11 – Електрична принципова схема до прикладу у КНФ

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

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

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

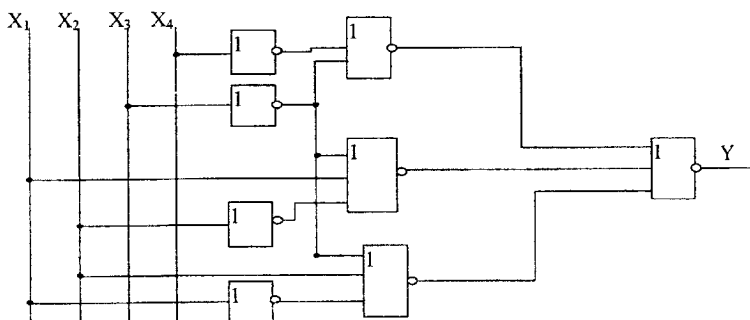


Рисунок 4.12 – Електрична принципова схема до прикладу 2 в базисі Пірса

Приклад 3. Спроекувати систему управління промисловим роботом, яка працює згідно з часовою діаграмою (рис. 4.13).

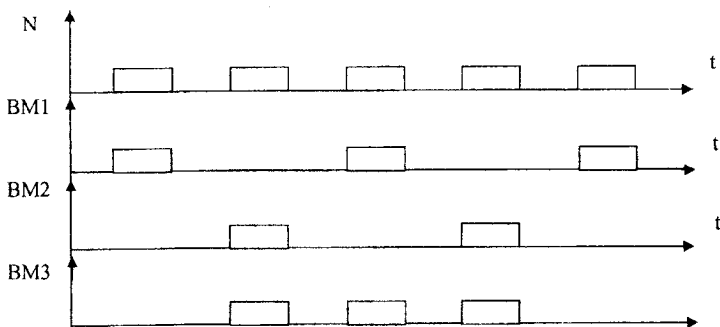


Рисунок 4.13 – Часова діаграма роботи пристрою

Згідно з наведеною часовою діаграмою роботи промислового робота 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 дешифрактором. Кількість триге-

рів n лічильника визначається з аналізу часової діаграми за допомогою такого співвідношення

$$[n] = \log_2 N,$$

де - $[n]$ ціле число;

N – Кількість імпульсів генератора, які складають цикли роботи промислового робота. Так, для розглядаючого прикладу маємо

$$[n] = \log_2 5 = 3.$$

Спроекуємо лічильник імпульсів на базі J_k – тригерів. На підставі часової діаграми промислового робота складаємо таблицю станів та переходів (табл. 4.2).

Таблиця 4.2

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	\emptyset	\emptyset	\emptyset
6	1	1	0	\emptyset	\emptyset	\emptyset
7	1	1	1	\emptyset	\emptyset	\emptyset

За таблицею 4.2 складаємо карту Карно для J – входів кожного тригеру (рис. 4.14).

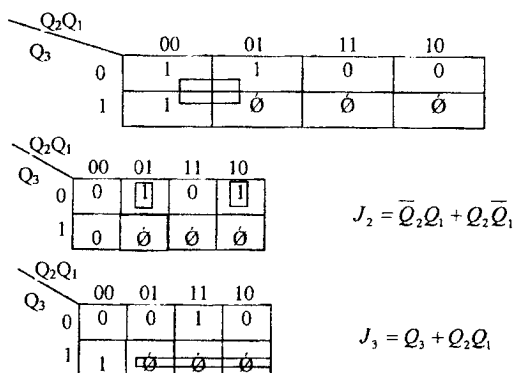


Рисунок 4.14 – Карти Карно для трьох тригерів

Відповідно до отриманих математичних виразів складемо електричну принципову схему лічильника (рис. 4.15).

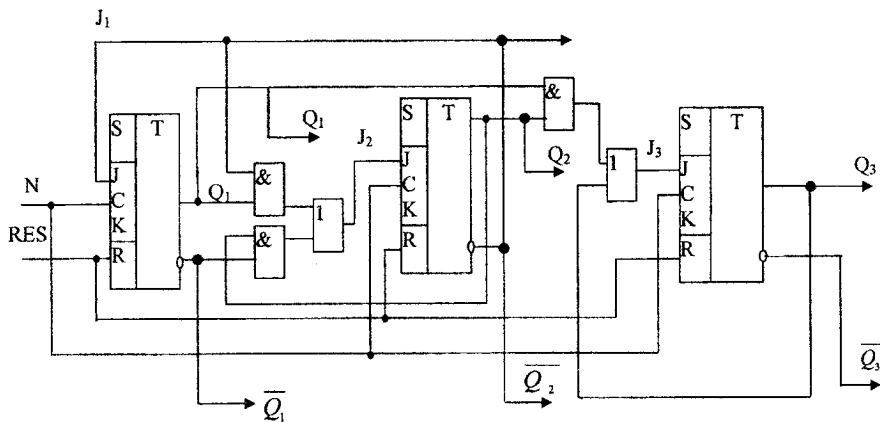


Рисунок 4.15 – Електрична принципова схема лічильників імпульсів

Із часової діаграми роботи промислового роботу будуюмо таблицю істинності дешифратора (табл. 4.3).

Таблиця 4.3

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	∅	∅	∅	∅	∅	∅

За таблицею істинності складемо карти Карно для кожного виходу дешифратора (рис. 4.16).

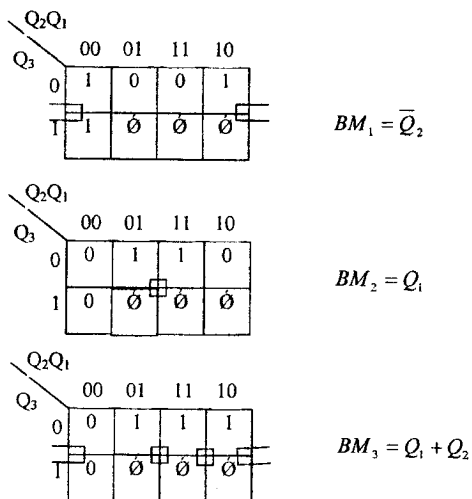


Рисунок 4.16 – Карти Карно для трьох виходів дешифратора

Відповідно до отриманих математичних виразів проектуємо електричну принципову схему дешифратора (рис. 4.17).

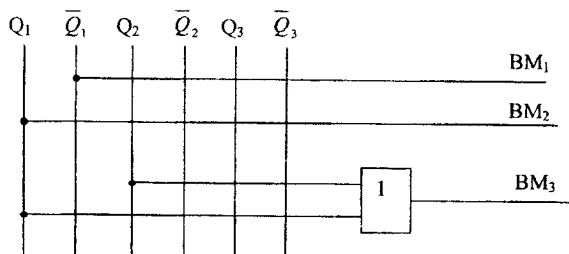


Рисунок 4.17 – Електрична принципова схема дешифратора

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

$$N_p = N_r.$$

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 3-х тригерах. Складаємо таблицю станів та переходів тригерів (табл. 4.4).

Таблиця 4.4

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).

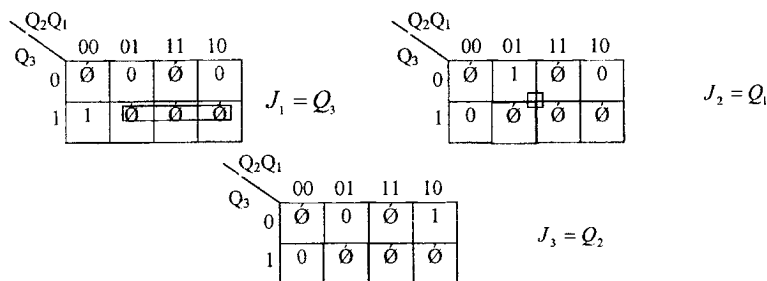


Рисунок 4.18 – Карти Карно для трьох тригерів

На підставі отриманих рівнянь для розглядаючого прикладу маємо

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

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

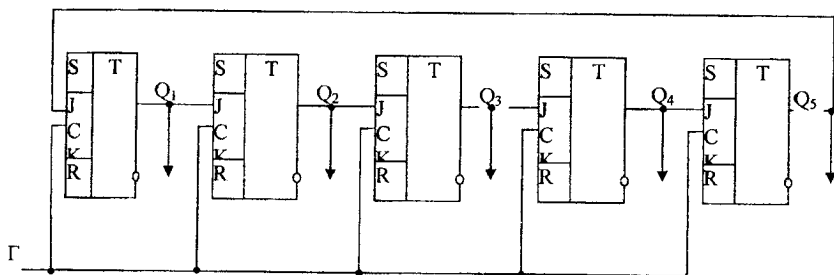


Рисунок 4.19 – Електрична принципова схема кільцевого регістру зсуву

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

Таблиця 4.5

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:

$$BM_1 = Q_1 + Q_3 + Q_5,$$

$$BM_2 = Q_2 + Q_4,$$

$$BM_3 = Q_2 + Q_3 + Q_4.$$

Згідно з отриманим виразами проектуємо електричну схему шифратора (рис. 4.20).

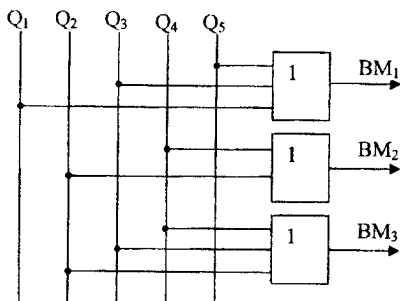


Рисунок 4.20 – Електрична принципова схема шифратора

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

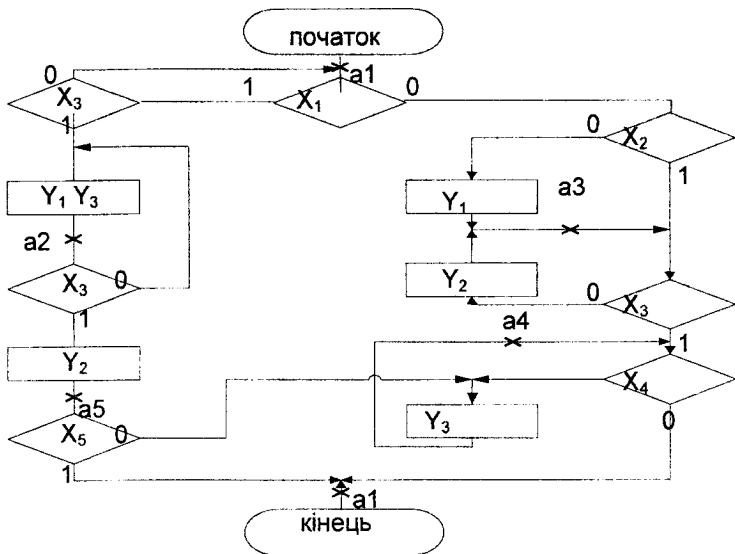


Рисунок 4.21 – Блок-схема алгоритму автоматичного пристрою

Як видно із схеми, пристрій повинен мати п'ять входів ($X_1 \dots X_5$) і три виходи (Y_1, Y_2, Y_3). Він може знаходитися в одному із п'яти станів $a_1 \dots a_5$. Для фіксації і зберігання станів необхідно визначити кількість тригерів за формулою

$$m = \log_2 N,$$

де N – число станів.

У нашому прикладі $N=5$, тоді:

$$m = \log_2 5 = 2,52.$$

Припустимо, що $m = 3$.

Присвоїмо кожному стану пристрою трирозрядний 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.

З метою перешкодження появи гонок використаємо імпульсну синхронізацію D-тригерів.

Для відображення станів пристрою будемо граф переходів (рис 4.22)

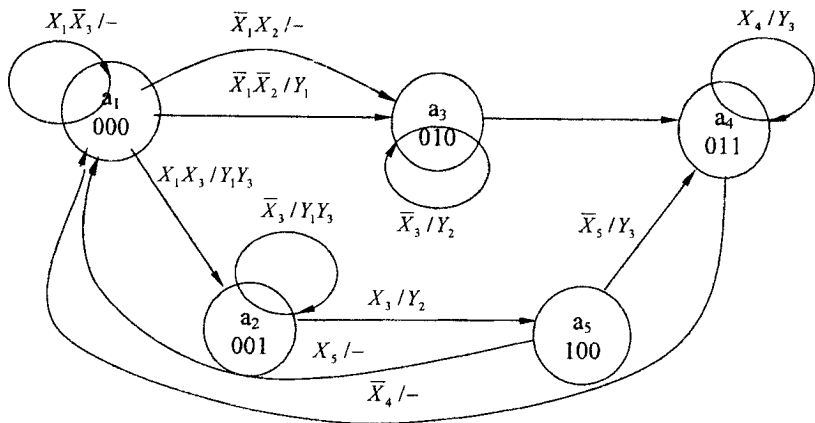


Рисунок 4.22 – Граф переходів (мікропрограми)

Складемо таблицю переходів тригерів (таб. 4.6)

Таблиця 4.6

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_1X_3	$Y_1\bar{Y}_3$	D_3
		a ₃	010	\bar{X}_1X_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_2D_3
a ₄	011	a ₁	000	\bar{X}_4	-	-
		a ₄	011	X_4	Y_3	D_2D_3
a ₅	100	a ₁	000	X_5	-	-
		a ₄	011	\bar{X}_5	Y_3	D_2D_3

Використовуючи канонічний метод структурного аналізу, запишемо Бульові функції вихідних сигналів і сигналів збудження тригерів в ЗДНФ.

Вихідні сигнали:

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

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

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

Сигнали збудження тригерів: $D_1 = a_2 X_3$;

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

$$D_3 = a_1 X_1 X_3 \cup a_2 \bar{X}_3 \cup a_3 X_3 \cup a_4 X_4 \cup a_5 X_5.$$

Мінімізуємо Бульову функцію сигналу збудження D_2 :

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

Інші функції записані в мінімальній формі ДНФ.

Отримані функції приведемо до базису І-НІ:

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

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

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

Далі будемо функціональну схему автоматичного пристрою (рис. 4.23)

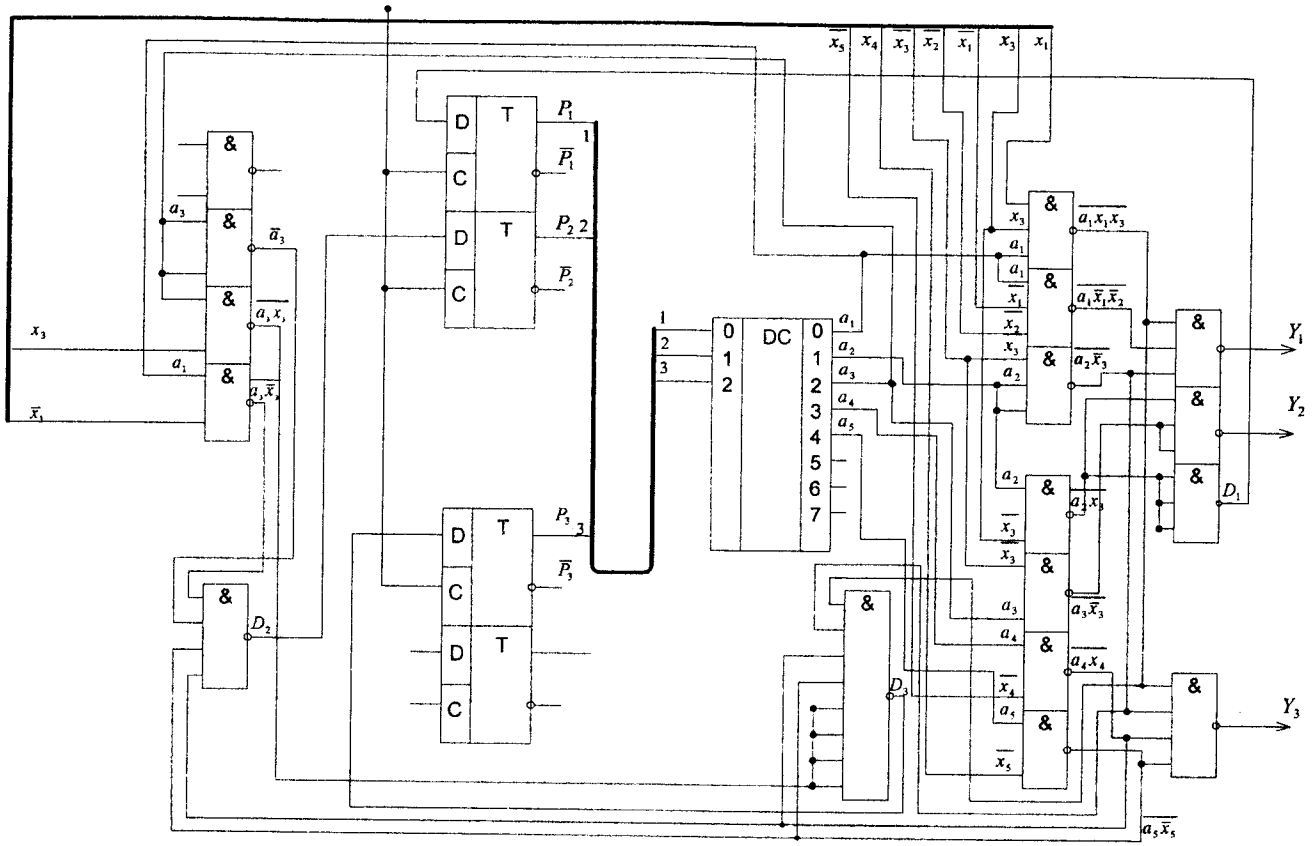


Рисунок 4.23 – Функціональна схема пристрою

СПИСОК СКОРОЧЕНЬ

РТП – радіотехнічні пристрої.

ІС – інтегральні схеми.

ЛФ – логічна функція.

ЗДНФ – звершена диз'юнктивна нормальна форма.

ЗКНФ – звершена кон'юнктивна нормальна форма.

ДОН – джерело опорної напруги.

ЛЕ – логічний елемент.

КП – комбінаційний пристрій.

ДС – дешифратор.

СД – шифратор.

ТТЛ – транзисторна логіка.

КМОП – комплементарний метал-окисел-напівпровідник.

ЗП – запам'ятовуючий пристрій.

ЦАП – цифро-аналоговий перетворювач.

ПЗП – постійний запам'ятовуючий пристрій.

БІП – блок формування імпульсної послідовності.

АІМ – амплітудно-імпульсна модуляція.

ЦСН – цифровий суматор-накопичувач.

ЦП – цифровий перемножувач.

ЦПФ – цифровий погоджений фільтр.

ОЗП – оперативний запам'ятовуючий пристрій.

ПП – послідовнісні пристрої.

ЛП – логічний перетворювач

ЕК – елементи комутації.

ОП – операційний підсилювач.

ЛІТЕРАТУРА

- 1 Микросхемотехника: Учебное пособие для вузов / А.Г.Алексеевко, И.И.Шагурин; Под ред. И. П. Степаненко. – М.: Радио и связь, 1982. – 416 с.
- 2 Проектирование дискретных устройств на интегральных микросхемах: Справочник/ Г.Н.Пухальский, Т.Я.Новосельцева. – М.: Радио и связь, 1990. – 304 с.
- 3 Зельдин Э. А. Цифровые интегральные микросхемы в информационно-измерительной аппаратуре. – Л.: Энергоатомиздат, 1986. – 250 с.
- 4 Цифровые и аналоговые интегральные микросхемы: Справочник /С.В.Якубовский, Л.И.Ниссельсон и др.; Под ред. С.В.Якубовского. – М.: Радио и связь, 1989. – 496 с.
- 5 Гутников В. С. Интегральная электроника в измерительных устройствах. – Л.: Энергоатомиздат, 1988. – 304 с.
- 6 Микросхемы ЦАП и АЦП: функционирование, параметры, применение/ Б. Г.Федорков, В.А.Телец. – М.: Энергоатомиздат, 1990. – 320 с.
- 7 Микроэлектронные устройства формирования и обработки сложных сигналов/ А.А.Сикарев, О.Н.Лебедев. – М.: Радио и связь, 1983. – 216с.
- 8 Устройства дискретной обработки радиосигналов: Учебное пособие/
С.Б.Макаров, И.А.Цикин. – Л.: ЛПИ, 1984. – 72 с.

Навчальне видання

Олександр Фанілович Єнікєєв

Олександр Олександрович Сердюк

ПРОЕКТУВАННЯ ЦИФРОВИХ ПРИСТРОЇВ
НА ІНТЕГРАЛЬНИХ МІКРОСХЕМАХ

Редактор

Ірина Іванівна Дья

Підпис. до друку.

Формат 60×8

Ризограф. друк.

Ум.-друк. арк.. 6,75

Обл. -вид. арк. 4,00

Тираж 60 прим.

Зам. № 58

ДДМА. 84313, м. Краматорськ, вул. Шкадінова, 72