

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

МЕТОДИЧНІ ВКАЗІВКИ

до практичних занять, самостійної роботи
та підготовки до модульного контролю
з дисципліни

**“ПРИСТРОЇ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНОЇ
ТЕХНІКИ”**

для студентів спеціальності
152 „Метрологія та інформаційно-вимірвальна техніка“
(освітня програма: „Якість, стандартизація та сертифікація“)
денної й заочної форм навчання

Методичні вказівки до практичних занять, самостійної роботи та підготовки до модульного контролю з дисципліни “Пристрої інформаційно-вимірювальної техніки” для студентів спеціальності для студентів спеціальності 152 „Метрологія та інформаційно-вимірювальна техніка“ (освітня програма: „Якість, стандартизація та сертифікація“) денної й заочної форм навчання /Укл.: Н.А. Смирнова, О.В. Василенко, Н.І. Павлице. – Запоріжжя: НУ «Запорізька політехніка», 2021. – 72 с.

Укладачі: Н.А. Смирнова, ст. викл.,
О.В. Василенко, доц., канд. техн. наук,
Н.І.Павлице, аспірант.

Рецензент: А.В. Коротун, доц., канд. фіз.-мат. наук

Відповідальний за випуск: А.В. Коротун, доц., канд. фіз.-мат. наук

Затверджено
на засіданні кафедри
“Мікро- та наноелектроніка”
Протокол №7
від “26” листопада 2020 р.

Рекомендовано до видання
НМК ФРЕТ
Протокол №4
від “10” грудня 2020 р.

ЗМІСТ

ВСТУП	5
1 ОСНОВИ АЛГЕБРИ ЛОГІКИ.	6
1.1 Основні поняття теорії дискретних автоматизованих пристроїв.	6
1.2 Приклади розв'язання задач.	16
1.3 Задачі для самостійного розв'язання	17
2 ЗАСТОСУВАННЯ АЛГЕБРИ ЛОГІКИ ДО РЕЛЕЙНО-КОНТАКТНИХ СХЕМ.	20
2.1 Основні поняття.	20
2.2 Приклади розв'язання задач.	24
2.3 Задачі для самостійного розв'язання	30
3 СИНТЕЗ КОМБІНАЦІЙНИХ АВТОМАТІВ. ДОСКОНАЛІ НОРМАЛЬНІ ФОРМИ	35
3.1 Основні поняття	35
3.2 Правила побудови ДДНФ і ДКНФ за таблицею істинності	37
3.3 Приклади розв'язання задач	38
3.4 Задачі для самостійного розв'язання	41
4 ПИТАННЯ ДО МОДУЛЬНОГО КОНТРОЛЮ І	44
5 СИНТЕЗ КОМБІНАЦІЙНИХ АВТОМАТІВ. МЕТОД КВАЙНА	45
5.1 Основні поняття	45
5.2 Приклад розв'язання задач	47
5.3 Задачі для самостійного розв'язання	51
6 СИНТЕЗ КОМБІНАЦІЙНИХ АВТОМАТІВ. МЕТОД КАРТ КАРНО	52
6.1 Основні поняття	52
6.2 Приклад розв'язання задач	53
6.3 Задачі для самостійного розв'язання	55
7 СИНТЕЗ ПОСЛІДОВНОСНИХ АВТОМАТІВ	57
7.1 Основні поняття	57
7.2 Приклади розв'язання задач	60

7.3 Задача для самостійного розв'язання	64
8 ПИТАННЯ ДО МОДУЛЬНОГО КОНТРОЛЮ II	66
РЕКОМЕНДОВАНА ЛІТЕРАТУРА	67
ДОДАТОК А Позначення логічних елементів за найбільш розповсюдженими стандартами	69
ДОДАТОК Б Тригери	70

ВСТУП

Засоби автоматизації обладнання є одним з основних шляхів підвищення продуктивності робочих машин, що використовуються в промисловості, зниження енергоспоживання, а також оперативного контролю за станом технологічного процесу і введення резерву. Ці засоби не тільки підвищують ефективність виробництва, але також полегшують побут людини в його повсякденному житті. В останні роки значний прогрес в цій області пов'язаний із застосуванням мікропроцесорів і мікроконтролерів.

Програмовані реле залишаються незамінними помічниками для реалізації широкого спектру інженерних і побутових завдань в якості засобів автоматизації локальних контурів, окремих агрегатів машин і механізмів.

На основі інтелектуальних реле інтуїтивно і зрозуміло будуються різні системи автоматичного керування. Компактні розміри і простота програмування дозволяють розробляти на базі програмованих реле елементи системи «розумний дім».

Серед технічних засобів автоматизації значне місце займають пристрої релейно-контактної дії. Ці пристрої (їх в загальному випадку називають перемикальними схемами) містять сотні реле, напівпровідників і електромагнітних елементів. Описувати та конструювати такі схеми в силу їх громіздкості доволі важко.

Ще у 1910 році фізик П. С. Еренфест вказав на можливість використання апарата алгебри логіки при дослідженні релейно-контактних схем (РКС). Однак його ідеї почали реалізовуватися значно пізніше, коли створення загальної теорії конструювання РКС стало гостро необхідним.

Використання алгебри логіки в конструюванні РКС виявилось можливим в зв'язку з тим, що кожній схемі можна поставити у відповідність деяку формулу алгебри логіки, і кожна формула алгебри логіки реалізується за допомогою деякої схеми.

Ця обставина дозволяє виявити можливості заданої схеми, вивчаючи відповідну формулу, а спрощення схеми звести до спрощення формули.

З іншого боку, до побудови схеми можна завчасно описати за допомогою формули ті функції, які схема повинна виконувати.

1 ОСНОВИ АЛГЕБРИ ЛОГІКИ

1.1 Основні поняття теорії дискретних автоматизованих пристроїв

Під дискретним автоматизованим пристроєм розуміють керуючий пристрій, що здійснює переробку апріорної і поточної інформації в керуючу, причому носіями всіх перерахованих складових інформації є дискретні за рівнем і в часі сигнали. Це означає, що стан сигналу кожного входу (виходу) автоматизованого пристрою характеризується двома рівнями: мінімальним, умовно позначається 0, і максимальним, що позначається 1. При наявності декількох входів (виходів) комбінацію мінімальних і максимальних (нульових і одиничних) рівнів можна сприймати двояко: як наявність або відсутність на кожному вході (виході) задаючого або командного сигналу, що надходить на будь-якої виконавчий елемент; як кодову комбінацію, що позначає, наприклад, двійкове число, кількість розрядів n якого дорівнює кількості входів (виходів). Отже, можна говорити про багаторівневий вихідний сигнал з числом рівнів $M = 2^n$.

У першому випадку дискретний логічний автоматизований пристрій вирішує завдання логічного аналізу ситуації, яка визначається низкою діючих або відсутніх факторів, у другому – арифметичний пристрій здійснює перетворення інформації шляхом виконання арифметичних операцій над числами. Як логічні, так і арифметичні пристрої реалізуються застосуванням обмеженого числа типових елементарних операцій над дискретними сигналами і східних методів синтезу та аналізу.

Дискретні керуючі пристрої бувають *комбінаційними* і *послідовносними*. Перші характерні тим, що комбінація станів вихідних сигналів в даному такті однозначно визначається комбінацією входних сигналів в цьому ж такті часу. Комбінаційні автомати називають також автоматами з нульовою пам'яттю. У послідовносних автоматів комбінація станів виходів залежить не тільки від комбінації станів входів в даному такті, а й від того, якими були комбінації станів входів в попередніх тактах. Їх називають також автоматами з ненульовою пам'яттю, так як для вироблення своїх вихідних (керуючих) сигналів вони повинні не тільки «знати», що зараз (в даному такті) відбувається на вході, а й «пам'ятати», що було

раніше. Послідовнісні автомати відрізняються від комбінаційних також наявністю зворотних зв'язків, за якими на входи автомата подаються сигнали, що відповідають стану виходів на попередньому такті.

За характером реакції на вхідний вплив розрізняють *асинхронні* автомати, у яких зміна стану виходів відбувається відразу після зміни вхідних, і *синхронні*, у яких зміна виходів відбувається лише в тому випадку, якщо після зміни вхідних сигналів надійде синхронізуючий сигнал, що дозволяє перетворення.

Багато механізмів промислового і побутового призначення працюють циклічно. *Циклом роботи обладнання* називають сукупність елементарних робочих і допоміжних операцій, які називаються етапами циклу, здійснюваних у певній послідовності, необхідній для виконання обладнанням робочих функцій. *Етап циклу* – це найпростіша, нероздільна частина циклу. Протягом відпрацювання етапу циклу не відбувається ніяких змін (включень або відключень) в дії або стані робітників і допоміжних органів обладнання. Етап циклу також часто називають *тактом роботи схеми* (термін «такт роботи схеми» повністю еквівалентний терміну «етап циклу», але його використовують в теорії релейних схем). Етапи циклу:

- елементарні рухи робочих та допоміжних органів обладнання,
- включення або відключення нагріву (охолодження) до досягнення певної робочої температури,
- витримки часу і т.п.

Етапи циклу характеризуються параметрами, до яких відносяться напрямок і швидкість переміщення, включення або відсутність включення елементів електрообладнання.

При синтезі принципових схем керування електроавтоматикою цих механізмів потрібно привести задані умови роботи механізму до вигляду, зручного для аналізу. При вирішенні цього завдання застосовують циклограми і діаграми роботи, оператори станів і подій, карти і таблиці станів, таблиці включень. Кожен з цих способів має переваги і недоліки.

На рис. 1.1 наведено спосіб графічної формалізації умов роботи наступного циклу:

- завдання на запуск циклу (кнопка «Цикл»);
- швидке переміщення механізму вправо до датчика Д1;

- повільне переміщення вправо до датчика Д2;
- швидке переміщення вліво в вихідну точку до датчика Д3.

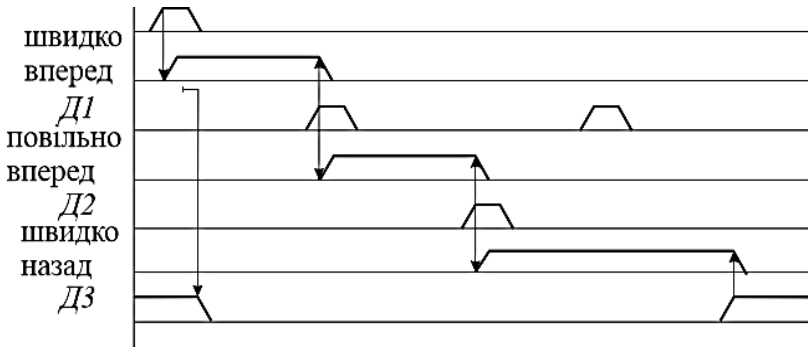


Рисунок 1.1 – Циклограма

Вибір способу графічної формалізації часто залежить від суб'єктивних причин. Спосіб накреслення циклограми повинен бути простим і наочним, при необхідності враховувати часові затримки, вплив фронтів включення і виключення і бути зручним для аналізу.

Головним для побудови циклограм є послідовність роботи в часі командних (кнопок, тумблерів і т. д.), виконавчих (керуючих електродвигунами, електромагнітами, муфтами і т. д.) і контролюючих (датчиків, реле тиску і т. д.) елементів. При необхідності в вихідну циклограму додають проміжні і часові сигнали. На осі ординат циклограми показують сигнали, що надходять від зазначених елементів механізму, в порядку їх появи при роботі механізму з початку циклу, а на осі абсцис – час в безрозмірному масштабі. На циклограмі зображується факт наявності сигналу (високий рівень з довільною амплітудою) або його відсутності (нульовий рівень).

Вплив сигналів один на одного зображують стрілкою, спрямованою від переднього або заднього фронту сигналу до керованого. Час виконання будь-якої операції, а також часові затримки показують горизонтальною лінією довільної довжини, що починається від командного сигналу, з переходом в вертикальну стрілку в кінці операції. Нахил фронтів сигналів може бути довільним, проте для наочності читання циклограми рекомендується нахил 60° .

Циклограма є основою для опису умов роботи механізму або типової схеми, тому для спрощення синтезу електроавтоматики

вхідним, проміжним і вихідним сигналам привласнюють позначення, що легко запам'ятовуються. Сигнали, зображені на циклограмі, можна розглядати як логічні змінні і описати вихідні змінні через вхідні рівняннями алгебри логіки.

Подібно до того, як виконуються різні операції між числами і літерними символами в звичайній алгебрі, можна здійснювати математичні операції над словесними висловлюваннями. Наука, що займається обчисленням висловлювань, називається *математичною логікою*. З математичної логіки можна виділити частину, яка займається вивченням математичних зв'язків тільки між двома висловлюваннями: «істинно» (1, включено, так) і «хибно» (0, відключено, немає). Цю частину прийнято називати *алгеброю логіки* (розділ дискретної математики). В алгебрі логіки з змінними можна виконувати операції додавання (АБО / диз'юнкція), множення (І / кон'юнкція), інверсії (НІ / заперечення / інверсія).

Можна скласти наступні *логічні рівності та правила*:

$$\bar{1} = 0; \bar{0} = 1; 0 + 0 = 0; 1 + 1 = 1; 1 \cdot 0 = 0; 1 \cdot 1 = 1.$$

Закони логіки записують у формі алгебраїчних виразів, де змінні позначають буквами, а види дій відповідними знаками. *Основні закони алгебри логіки* наведені у Табл. 1.1.

Таблиця 1.1 – Основні закони дискретної алгебри

Асоціативність операції $\vee$	$x \vee (y \vee z) = (x \vee y) \vee z$
Асоціативність операції $\wedge$	$x \wedge (y \wedge z) = (x \wedge y) \wedge z$
Комутативність операції $\vee$	$x \vee y = y \vee x$
Комутативність операції $\wedge$	$x \wedge y = y \wedge x$
Дистрибутивність операції $\wedge$ над $\vee$	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
Дистрибутивність операції $\vee$ над $\wedge$	$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$
Ідентичність для $\vee$	$x \vee 0 = x$
Ідентичність для $\wedge$	$x \wedge 1 = x$
Анігілятор для $\wedge$	$x \wedge 0 = 0$

Наступні закони (Табл. 1.2) є в алгебрі логіки, але відсутні у звичайній.

Таблиця 1.2 – Специфічні закони дискретної алгебри

Анігілятор для $\vee$	$x \vee 1 = 1$
Ідемпотентність $\vee$	$x \vee x = x$
Ідемпотентність $\wedge$	$x \wedge x = x$
Адсорбція або поглинання	$x \wedge (x \vee y) = x$ $x \vee (x \wedge y) = x$
Склеювання / розщеплення	$(x \wedge y) \vee (\bar{x} \wedge y) = y$ $(x \vee y) \wedge (\bar{x} \vee y) = y$
Узагальнене склеювання	$(x \wedge y) \vee (\bar{x} \wedge z) = (x \wedge y) \vee (\bar{x} \wedge z) \vee (y \wedge z)$
Закони де Моргана	$\overline{x \wedge y} = \bar{x} \vee \bar{y}$ $\overline{(x \vee y)} = \bar{x} \wedge \bar{y}$

Розглянемо закони більш детально (із альтернативними назвами):

➤ переміщувальний – на логічну суму і перемноження не впливає порядок розташування змінних, тобто

$$a + b + c = c + b + a; \quad abc = cba;$$

та
$$(a + b) + c = (c + b) + a; \quad (ab)c = (cb)a;$$

➤ розподільний – загальний множник можна винести за дужки, тобто

$$b \cdot a + b \cdot c = b(a + c);$$

➤ інверсії – при інвертуванні суми декількох змінних знак інверсії переноситься на кожен змінну і складання замінюється добутком, тобто

$$\overline{a + b + c} = \bar{a} \bar{b} \bar{c};$$

і при інвертуванні твору кількох змінних знак інверсії переноситься на кожен змінну і добуток замінюється сумою,

$$\overline{abc} = \bar{a} + \bar{b} + \bar{c};$$

- додатковості:

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

- універсальної множини:

$$a + 1 = 1, \quad a \cdot 1 = a;$$

- нульової множини:

$$a + 0 = a, \quad a \cdot 0 = 0;$$

- тавтології:

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

- поглинання:

$$a + ab = a(1 + b) = a \cdot 1 = a; \quad a(a + b) = aa + ab = a + ab + a;$$

- склеювання:

$$ab + a\bar{b} = a, \quad (a + b)(a + \bar{b}) = a.$$

Для подвійної операції інверсії: $\overline{\overline{a}} = a.$

Наслідками основних законів є *правила*:

$$a + \bar{a}b = (a + \bar{a})(a + b) = 1(a + b) = a + b;$$

$$\bar{a} + a\bar{b} = (\bar{a} + a)(\bar{a} + \bar{b}) = 1(\bar{a} + \bar{b}) = \bar{a} + \bar{b};$$

$$\bar{a}(a + b) = \bar{a}a + \bar{a}b = 0 + \bar{a}b = \bar{a}b;$$

$$a(\bar{a} + b) = a\bar{a} + ab = 0 + ab = ab.$$

Використовуючи закони алгебри логіки, можна легко довести наступні *теореми*:

$$a + cb = (a + c)(a + b);$$

$$a(a + b)(a + c) \dots = a;$$

$$a + ab + ac + \dots = a;$$

$$ab + \bar{a}c + bc = ab + \bar{a}c;$$

$$(a+b)(\bar{a}+c)=ac+\bar{a}b;$$

$$(a+b)(\bar{a}+c)(b+c)=(a+b)(\bar{a}+c).$$

Основні закони алгебри логіки дозволяють виконувати *перетворення логічних функцій*. Функція двох змінних є основою в алгебрі логіки. Так як кожна вхідна логічна змінна (x_1, x_2) може приймати два значення, то можливі 16 різних функцій двох змінних (Табл. 1.3).

Таблиця 1.3 – Логічні функції двох змінних

x_1	0	0	1	1	Позначення	Назва
x_2	0	1	0	1		
$f_0(x_1, x_2)$	0	0	0	0	0	константа 0
$f_1(x_1, x_2)$	0	0	0	1	$x_1 \& x_2$	кон'юнкція
$f_2(x_1, x_2)$	0	0	1	0	$(x_1 \rightarrow x_2)$	інверсія $f_{13}(x_1, x_2)$
$f_3(x_1, x_2)$	0	0	1	1	x_1	змінна x_1
$f_4(x_1, x_2)$	0	1	0	0	$(x_2 \rightarrow x_1)$	інверсія $f_{11}(x_1, x_2)$
$f_5(x_1, x_2)$	0	1	0	1	x_2	змінна x_2
$f_6(x_1, x_2)$	0	1	1	0	$x_1 \oplus x_2$	нерівнозначність
$f_7(x_1, x_2)$	0	1	1	1	$x_1 \vee x_2$	диз'юнкція
$f_8(x_1, x_2)$	1	0	0	0	$x_1 \downarrow x_2$	стрілка Пірса
$f_9(x_1, x_2)$	1	0	0	1	$x_1 \sim x_2$	еквівалентність
$f_{10}(x_1, x_2)$	1	0	1	0	$\overline{x_2}$	заперечення x_2
$f_{11}(x_1, x_2)$	1	0	1	1	$x_2 \rightarrow x_1$	імплікація x_2 в x_1
$f_{12}(x_1, x_2)$	1	1	0	0	$\overline{x_1}$	заперечення x_1
$f_{13}(x_1, x_2)$	1	1	0	1	$x_1 \rightarrow x_2$	імплікація x_1 в x_2
$f_{14}(x_1, x_2)$	1	1	1	0	$x_1 x_2$	штрих Шеффера
$f_{15}(x_1, x_2)$	1	1	1	1	1	константа 1

Функцією алгебри логіки $f(x_1, x_2, \dots, x_n)$ від n змінних $x_1, x_2, \dots, x_n$ називається функція, що приймає значення 1 або 0, аргументи якої також приймають значення 1 або 0.

При спрощення булевих формул або висловлювань, пов'язаних дужками і логічними операціями, використовують такі *правила пріоритету* (або *старшинства*) логічних операцій – від найбільш сильної – до слабкої:

$$\neg \quad \wedge (\cdot) (i) \quad \vee (+) \text{ (або)} \quad \rightarrow \quad \leftrightarrow.$$

Словесно: заперечення, кон'юнкція, диз'юнкція, імплікація, еквівалентність.

Будь-яка формула алгебри логіки є функція алгебри логіки, причому тотожно істинні і тотожно хибні формули являють собою постійні функції.

Будь-яку функцію алгебри логіки можна представити у вигляді формули алгебри логіки:

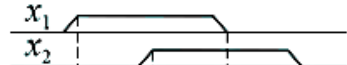
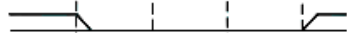
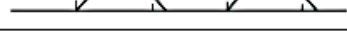
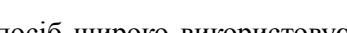
$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= f(1, 1, \dots, 1) \& x_1 \& x_2 \& \dots \& x_n \vee \\ &\vee f(1, 1, \dots, 0) \& x_1 \& x_2 \& \dots \& x_{n-1} \& \overline{x_n} \vee \dots \vee \\ &\vee f(0, 0, \dots, 0) \& \overline{x_1} \& \overline{x_2} \& \dots \& \overline{x_{n-1}} \& \overline{x_n}. \end{aligned}$$

Відповідну функції $f(x_1, x_2, \dots, x_n)$ формулу алгебри логіки можна отримати за допомогою таблиці істинності цієї функції. Для цього для всіх наборів значень змінних, на яких функція $f(x_1, x_2, \dots, x_n)$ приймає значення 1, записується кон'юнкція змінних висловлювань, причому за член кон'юнкції береться x_i , якщо на зазначеному наборі значень змінних x_i дорівнює 1, і – $\overline{x_i}$, якщо на зазначеному наборі значень змінних x_i дорівнює 0. Диз'юнкція всіх отриманих таким чином кон'юнкцій і буде шуканої формулою алгебри логіки.

Табл. 1.4 можуть служити першим кроком до розуміння інженерного методу синтезу логічних схем за допомогою циклограм роботи механізмів і апарату алгебри логіки. Наприклад, для функції «І-НІ» видно, що вихідна логічна змінна дорівнює одиниці у всіх випадках, крім одночасної присутності вхідних змінних.

Складання структурної схеми керування за заданими умовами її роботи називають *синтезом*. Визначення умов роботи схеми або її окремих елементів за наявною структурою називають *аналізом схем керування*. Схеми на релейних і безконтактних елементах можна складати двома способами.

Таблиця 1.4 – Логічні функції двох змінних з циклограмою

Функція	Циклограма	Позначення	Рівняння
		—	—
Нульова		0	$f_1 = 0$
Одинична		1	$f_2 = 1$
Повторення x_1		x_1	$f_3 = x_1$
Інверсії x_1		$\bar{x}_1$	$f_4 = \bar{x}_1$
Повторення x_2		x_2	$f_5 = x_2$
Інверсії x_2		$\bar{x}_2$	$f_6 = \bar{x}_2$
I		$x_1 x_2$	$f_7 = x_1 x_2$
I-НЕ		$\bar{x}_1 \bar{x}_2$	$f_8 = \bar{x}_1 \bar{x}_2$
Заборони x_2		$x_1 \leftarrow x_2$	$f_9 = x_1 \bar{x}_2$
Заборони x_1		$x_2 \leftarrow x_1$	$f_{10} = \bar{x}_1 x_2$
АБО		$x_2 + x_1$	$f_{11} = x_1 + x_2$
АБО-НЕ		$x_2 \downarrow x_1$	$f_{12} = \bar{x}_1 + \bar{x}_2$
Імплікації x_1		$x_2 \rightarrow x_1$	$f_{13} = x_1 + \bar{x}_2$
Імплікації x_2		$x_1 \rightarrow x_2$	$f_{14} = \bar{x}_1 + x_2$
Рівнозначності		$x_2 \equiv x_1$	$f_{15} = x_1 x_2 + \bar{x}_1 \bar{x}_2$
Нерівнозначності		$x_2 \not\equiv x_1$	$f_{16} = x_1 \bar{x}_2 + \bar{x}_1 x_2$

Перший спосіб широко використовується в практиці складання релейно-контактних схем. Виходячи із заданих умов роботи окремих частин робочої машини, складають принципову схему системи автоматики, раціональність і правильність розробки якої залежить від індивідуальних знань та досвіду конструктора. Аналогічно складають безконтактні аналоги релейно-контактних схем, в яких задані умови роботи схеми виражаються у вигляді математичних символів. На підставі їх аналізу і складають схему з'єднань безконтактних елементів. При цьому доцільно провести мінімізацію будь-якої

контактної або безконтактної схеми, побудованої таким досвідченим способом.

Другий спосіб побудови (синтезу) схем заснований на більш повному використанні теорії алгебри логіки і принципів формалізації реальних умов роботи схеми автоматики. В цьому випадку виходять із заданих умов роботи, складаючи відповідні таблиці станів (карти функцій), де мають місце комбінації аргументів і значень функцій (вихідних сигналів) у вигляді символів 1 і 0. Основна задача синтезу полягає у визначенні такої форми вираження шуканої логічної функції, яку можна реалізувати з застосуванням мінімального числа максимально простих елементів. Синтез релейних схем керування зводиться до складання структурної формули (аналітичного виразу), яка описує логічні функції, які повинні виконуватися цим пристроєм. Потім аналізують отриману алгебраїчну формулу і складають графічне зображення схеми.

Аналітичний вираз, що відображає логічну функцію, для реалізації якої призначена дана схема, називають *структурною формулою схеми*.

Залежно від характеру роботи схеми логічні функції поділяють на одно- і багатотактні, а відповідні їм пристрої на *комбінаційні* та *послідовні*. У одноктактних функціях вихідна змінна залежить від комбінації вхідних змінних в даний момент часу, а в багатотактних ще й від попередніх комбінацій вхідних змінних, тобто від їхньої послідовності (або від стану виходів, тобто пам'яті). Простіше здійснити синтез одноктактної функції.

Змінну, значення якої залежать від значень інших довічних змінних – аргументів називають *перемикальною функцією* (ПФ). ПФ задається у вигляді відповідності її значення кожному набору (комбінації) значень її аргументів. Перемикальні функції вважаються різними, якщо відрізняються значеннями хоча б для одного набору.

При n аргументах число наборів $N = 2^n$, а число комутаційних функцій $M = 2^N$. Запис наборів і ПФ легко виконується у вигляді натурального ряду двійкових чисел. Набори нумеруються за значеннями двійкових чисел від 0 до $2^n - 1$.

Формула, справжня на деякому наборі змінних, називається *здійсненою*. Формула, справжня на всіх можливих наборах змінних, називається *тотожно істинною* (або *тавтологією*). Формула

помилкова на всіх можливих наборах змінних, називається *тотожно хибною* (або *протиріччям*).

Формули, які беруть однакові значення при однакових наборах значень змінних, що входять в них, називаються *рівнозначними* або *еквівалентними*. При доказі рівносильності формул використовуються основні рівнозначності (позначка $\equiv$).

1.2 Приклади розв'язання задач

Приклад 1.2.1 Знайти формулу, яка визначає функцію $f(x, y)$ за наступною таблицею істинності:

x	y	$f(x, y)$
1	1	0
1	1	1
1	0	1
1	0	0

Розв'язок

Скористаємося правилом отримання формули алгебри логіки з таблиці істинності для функції $f(x, y)$. Отримаємо:

$$f(x, y) = x \& y \vee x \& \bar{y}.$$

Спростимо отриману формулу:

$$x \& y \vee x \& \bar{y} \equiv (y \vee \bar{y}) \& x \equiv 1 \& x \equiv x.$$

Відповідь: $x \& y \vee x \& \bar{y} \equiv x$.

Приклад 1.2.2 Побудувати таблицю істинності для формули $f = ((\bar{x} \rightarrow y) \& \bar{z})$.

Розв'язок

Таблиця будується послідовно, відповідно до кроків побудови формули f :

x	y	z	$\bar{x}$	$(\bar{x} \rightarrow y)$	$\bar{z}$	f
0	0	0	1	0	1	0
0	0	1	1	0	0	0
0	1	0	1	1	1	1
0	1	1	1	1	0	0
1	0	0	0	1	1	1
1	0	1	0	1	0	0
1	1	0	0	1	1	1
1	1	1	0	1	0	0

1.3 Задачі для самостійного розв'язання

1.3.1 Визначити логічні значення наступних складних висловлювань:

- 1) $x \wedge (y \wedge z)$ при $x=0, y=1, z=1$;
- 2) $(x \wedge y) \wedge y$ при $x=1, y=0$;
- 3) $x \rightarrow (y \rightarrow z)$ при $x=0, y=1, z=1$;
- 4) $x \wedge y \rightarrow z$ при $x=1, y=0, z=1$;
- 5) $(x \wedge y) \leftrightarrow (z \vee \bar{y})$ при $x=0, y=1, z=1$;
- 6) $((x \vee y) \wedge z) \leftrightarrow ((x \wedge z) \vee (y \wedge z))$ при $x=0, y=1, z=1$;
- 7) $(\overline{x \vee y \rightarrow z}) \wedge x$ при $x=1, y=0, z=0$.

1.3.2 Знайти логічні значення x і y , при яких виконуються рівності:

- 1) $(1 \rightarrow x) \rightarrow y = 0$;
- 2) $x \wedge y = \bar{x}$;
- 3) $y \rightarrow \bar{x} = 0$;
- 4) $x \wedge \bar{y} = y \rightarrow x$.

1.3.3 Скласти таблиці істинності для формул:

- 1) $\bar{x} \vee \bar{y}$;
- 2) $(x \wedge y) \vee z$;
- 3) $(x \rightarrow y) \vee (x \rightarrow (y \wedge x))$;

$$4) (x \rightarrow (\overline{y \wedge x})) \rightarrow (x \vee z);$$

$$5) (x \vee y \rightarrow y \wedge x) \wedge \bar{x};$$

$$6) (x \rightarrow y) \wedge (y \rightarrow z) \rightarrow (x \rightarrow z);$$

$$7) (x \rightarrow y \wedge y \rightarrow z) \rightarrow (x \rightarrow z);$$

$$8) (\bar{x} \vee \bar{y}) \wedge z \rightarrow (x \leftrightarrow y);$$

$$9) (x \wedge \bar{y} \rightarrow \bar{y}) \leftrightarrow x;$$

$$10) (x \wedge (y \vee \bar{x})) \wedge ((\bar{y} \rightarrow x) \vee y);$$

$$11) (x \wedge \bar{y} \rightarrow y) \rightarrow (x \rightarrow y).$$

1.3.4 Знайти формулу, яка визначає функцію $f(x, y, z)$ за наступною таблицею істинності:

x	y	z	$f(x, y, z)$
1	1	1	0
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	0
0	1	0	1
0	0	1	0
0	0	0	1

1.3.5 За таблицями істинності знайти формули, що визначають функції f_1, f_2, f_3, f_4, f_5 ($f_i \equiv f(x, y, z)$), і надати їм більш простий вигляд:

x	y	z	f_1	f_2	f_3	f_4	f_5
1	1	1	1	1	1	0	1
1	1	0	1	0	1	1	0
1	0	1	0	1	0	1	0
1	0	0	0	1	0	1	0
0	1	1	0	0	0	0	1
0	1	0	1	0	1	0	1
0	0	1	0	1	1	1	1
0	0	0	0	0	0	0	1

1.3.6 Виписати всі функції алгебри логіки однієї змінної.

1.3.7 Виписати всі функції алгебри логіки двох змінних.

1.3.8 Нехай функція алгебри логіки:

1) $f_1(x, y, z)$ приймає значення 1 тоді і тільки тоді, коли тільки одна з її змінних приймає значення 1;

2) $f_2(x, y, z)$ приймає значення 1 тоді і тільки тоді, коли одночасно дві її змінних приймають значення 1;

3) $f_3(x, y, z)$ приймає значення, що збігається зі значенням, яке приймає більшість її змінних.

Скласти таблиці істинності для цих функцій, знайти формули алгебри логіки, що визначають їх, і надати цим формулам простіший вид.

1.3.9 Скласти таблицю істинності для формули $(x \rightarrow y) \& (y \rightarrow x)$ і переконатися в справедливості тотожності $x \leftrightarrow y \equiv (x \rightarrow y) \& (y \rightarrow x)$.

1.3.10 Довести тотожність:

1) $x \& 1 \equiv x$;

2) $\overline{(x \vee y)} \equiv \bar{x} \& \bar{y}$;

3) $x \vee x \equiv x$;

4) $x \& (x \vee y) \equiv x$.

1.3.11 Скласти таблицю істинності для формули $(x \rightarrow y) \& (y \rightarrow z) \rightarrow (x \rightarrow z)$ і довести, що $(x \rightarrow y) \& (y \rightarrow z) \rightarrow (x \rightarrow z) \equiv 1$.

1.3.12 Спростити логічну формулу:

1) $\overline{(x \vee y) \& z} \rightarrow x \vee z$;

2) $(\bar{x} \& \bar{y}) \vee (x \rightarrow y) \& x$.

1.3.13 Висловити еквіваленцію $x \leftrightarrow y$ через заперечення, кон'юнкцію, диз'юнкцію і розкрити дужки.

2 ЗАСТОСУВАННЯ АЛГЕБРИ ЛОГІКИ ДО РЕЛЕЙНО-КОНТАКТНИХ СХЕМ

2.1 Основні поняття

Релейно-контактна схема являє собою схематичне зображення деякого пристрою, що складається з *перемикачів, провідників*, що з'єднують їх, *входів* в схему і *виходів* із неї. В релейно-контактній схемі будь-який перемикач має тільки два стани: *замкнутий* і *розімкнутий*.

Кожен пристрій, що складається з з'єднаних між собою котушок реле і контактів (контактів реле, кнопок, датчиків), можна описати за допомогою булевих функцій і, навпаки, будь-яку логічну функцію можна реалізувати за допомогою схеми, що складається зі з'єднаних між собою котушок і контактів реле. Зазвичай при записі функцій котушки реле позначаються великими буквами, а контакти - маленькими. Включений стан реле (контакти реле) записується як $P=1$ ($p=1$), відключене стан реле (контакти реле) $P=0$ ($p=1$). Для найпростішої схеми, зображеної на рис. 2.1, а, можна записати:

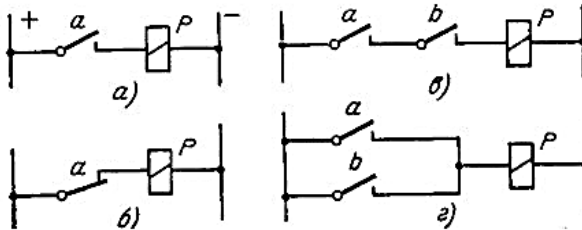


Рисунок 2.1 – Реалізація основних логічних функцій

$P=1$, якщо $a=1$, і $P=0$, якщо $a=0$, або $P=a$.

У схемі на рис. 2.1, б реле P включено, коли відключено реле a , або $P=\bar{a}$.

На схемі на рис. 2.1, в реле P включено, якщо обидва реле a і b : $P=ab$, а на схемі 2.1, г реле P включено, якщо включено хоча б одне реле a чи b : $P=a \vee b$.

Звідси випливають загальні правила: послідовному з'єднанню контактів відповідає кон'юнкція, паралельному – диз'юнкція, а нормально замкнутий контакт в схемі відповідає запереченню. Дотримую-

чись цих правил, легко виразити функцію, реалізовану схемою на рис. 2.2, а:

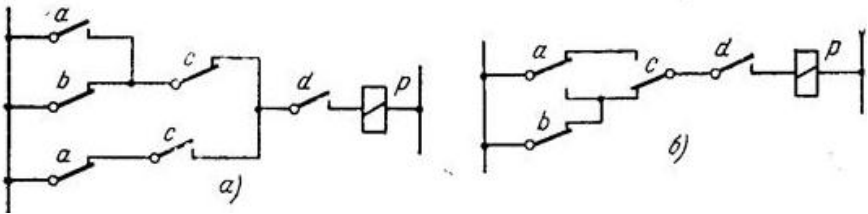


Рисунок 2.2 – Схеми, що реалізують функцію $P = [(a \vee \bar{b})\bar{c} \vee \bar{a}c]d$

Побудова контактної схеми, що відповідає аналітичному виразу логічної функції, також не представляє труднощів.

З метою зменшення числа контактних пружин можна з'єднувати контакти одного і того ж реле, що мають спільну точку: контакти, що замикає і розмикає, замінити одним, який переключає (рис. 2.2, б).

Основні задачі теорії релейно-контактних схем:

1. Задача *синтезу* – побудувати схему, яка реалізує задану функцію провідності. Для цього достатньо побудувати формулу алгебри логіки типу ДДНФ и ДКНФ (див.п. 3).

2. Задача *спрощення* – за даною схемою побудувати більш просту, що має таку ж функцію провідності (тобто рівнозначну схему).

3. Задача *аналізу схеми* – не вмикаючи схему в роботу, проаналізувавши з'єднання контактних груп, знайти функцію провідності схеми (задачі такого типу одна зі складових промислового шпигунства).

Логічні схеми реалізуються на логічних елементах: "НІ", "І", "АБО", "І-НІ", "АБО-НІ", "АБО, що виключає" і "Еквівалентність" (Табл. 2.1, ДОДАТОК А). Перші три логічних елемента дозволяють реалізувати будь-яку, як завгодно складну логічну функцію в булевому базисі.

Таблиця 2.1 – Позначення логічних елементів за стандартом

Логика	«НЕ»	«И»	«ИЛИ»	«И-НЕ»	«ИЛИ-НЕ»
Полож.					
Отриц.					

Логічними елементами називаються мікросхеми малого ступеня інтеграції, що реалізують найпростіші логічні функції двох – чотирьох аргументів. Найбільш поширені логічні елементи, що реалізують логічні функції І (рис. 2.3, а), АБО (рис. 2.3, б), І–НІ (рис. 2.3, в) і АБО–НІ (рис. 2.3, г). До логічних елементів відносяться також мікросхеми, що реалізують найпростіші послідовні алгоритми (наприклад, тригери).

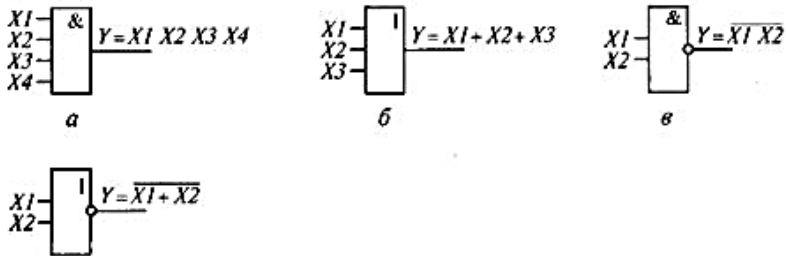


Рисунок 2.3 – Схемні позначення логічних елементів І (а), АБО (б), І–НІ (в), АБО–НІ (г)

Логічні елементи І реалізують функцію логічного множення (кон'юнкцію). Це означає, що вихідний сигнал схеми І дорівнює одиниці тільки в тому випадку, коли всі її вхідні сигнали рівні одиниці. Логічний елемент І називається також *схемою збігу*.

Логічні елементи АБО реалізують функцію логічного додавання (диз'юнкцію), тобто сигнал на виході схеми АБО дорівнює нулю тільки тоді, коли всі вхідні сигнали дорівнюють нулю.

Логічні елементи І–НІ реалізують функцію інверсії логічного добутку (функцію Шеффера), а елементи АБО–НІ – функцію інверсії

логічної суми (функцію Пірса). Отже, якщо логічний добуток дорівнює одиниці, то елемент І–НІ видає нульовий сигнал на своєму виході; якщо логічна сума дорівнює одиниці, то елемент АБО–НІ також видає нульовий сигнал. В іншому випадку на виходах елементів даного типу формується одиничний сигнал.

В одному корпусі мікросхеми зазвичай є чотири логічних схеми – на два входи кожна, або три схеми – на три входи кожна, або дві схеми – на чотири входи кожна незалежно від виду елементарних логічних функцій, які дані мікросхеми реалізують. Якщо не всі входи логічної схеми використовуються в проектуваному пристрої, то невикористовувані входи слід об'єднувати з використовуваними. Так, для реалізації функції інверсії необхідно об'єднати всі входи схеми І–НІ (рис. 2.4, а), тоді отримаємо $P = \overline{XX} = \overline{X}$.

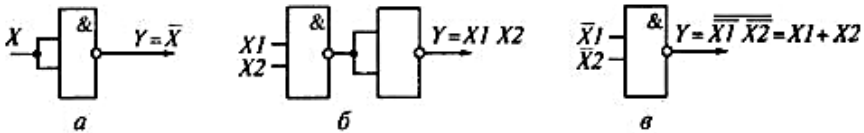


Рисунок 2.4 – Реалізація базових логічних функцій на елементах І–НІ: а - інверсія; б - логічне добуток; в - логічна сума

Якщо дві схеми І–НІ з'єднати послідовно, як показано на рис. 2.4, б, то друга схема інвертує інверсію логічного добутку, отриманого на першій схемі, так що на виході другої схеми отримаємо саме логічний добуток.

Якщо ж на вхід схеми І–НІ подати інверсії сигналів, що нас цікавлять (рис. 2.4, в), отриманих попередньо за допомогою схеми І–НІ, то на виході отримаємо логічну суму вихідних сигналів відповідно до закону Де Моргана. Таким чином, за допомогою елементів І–НІ можна реалізувати всі базові функції булевої алгебри, а отже, будь-які логічні функції. Так само універсальні і елементи АБО–НІ. Елементи інших типів, які при наявності елементів І–НІ або АБО–НІ не є обов'язковими для реалізації алгоритмів керування, мають, як правило, спеціальне призначення. Так, елементи І (див. рис. 2.3, а) зазвичай є підсилювальними елементами. Їх допустимий вихідний струм досягає 100 мА, в той час як звичайні логічні елементи мають допустимий вихідний струм до 5 мА.

Завдання аналізу полягає у визначенні функції f , яка реалізується заданою логічною схемою. При вирішенні такого завдання зручно дотримуватися такої послідовності дій.

1. Логічна схема розбивається на яруси. Ярусам присвоюються послідовні номери.

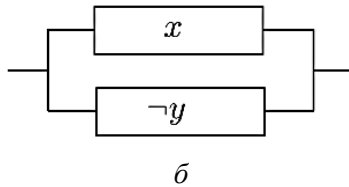
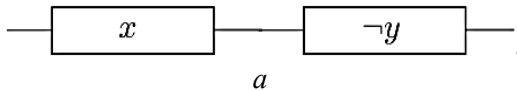
2. Виводи кожного логічного елемента позначаються назвою шуканої функції, забезпеченим цифровим індексом, де перша цифра - номер ярусу, а решта цифри - порядковий номер елемента в ярусі.

3. Для кожного елемента записується аналітичний вираз, що зв'яже його вихідну функцію з входніми змінними. Вираз визначається логічною функцією, яка реалізується даним логічним елементом.

4. Проводиться підстановка одних вихідних функцій через інші, поки не вийде булева функція, виражена через входні змінні.

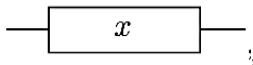
2.2 Приклади розв'язання задач

Приклад 2.2.1 Знайдіть функцію провідності для схем *a* і *б*:

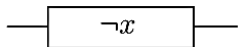


Розв'язок

a) Якщо керуючий сигнал позначити через x , то нормально розімкнуте реле позначається



а нормально замкнуте



Функцією провідності схеми називається здатність проводити або не проводити струм через схему з'єднання контактних груп реле в залежності від комбінації керуючих сигналів, поданих на обмотки всіх реле, що утворюють схему.

Отже, в даному випадку функція провідності схеми – $x \cdot \bar{y}$, тобто послідовне з'єднання реле реалізую кон'юнкцію.

б) Вочевидь, функція провідності д даної схеми – $x \vee \bar{y}$, тобто паралельне з'єднання реле реалізую диз'юнкцію.

Приклад 2.2.2 За таблицею істинності синтезувати та спростити релейно-контактну схему:

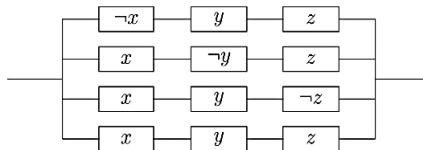
x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Розв'язок

Винишемо за таблицею ДДНФ:

$$f = \bar{x}yz \vee x\bar{y}z \vee xy\bar{z} \vee xyz .$$

Задача синтезу вирішена. Будуємо схему для отриманої формули:



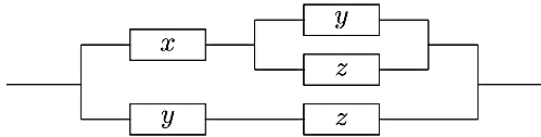
Переходими до задачі спрощення:

$$\begin{aligned}
& \bar{x}yz \vee x\bar{y}z \vee xy\bar{z} \vee xyz \equiv \\
& \equiv \bar{x}yz \vee xyz \vee x\bar{y}z \vee xyz \vee xy\bar{z} \vee xyz \equiv \\
& \equiv (\bar{x} \vee x)yz \vee x(\bar{y} \vee y)z \vee xy(\bar{z} \vee z) \equiv xy \vee xz \vee yz.
\end{aligned}$$

Тепер стало на 6 реле менше. Спростимо далі:

$$xy \vee xz \vee yz \equiv x(y \vee z) \vee yz.$$

Зэкономили ще одне реле, ала схема стала менш технологічною (встратила симетричність). Побудуємо схему для отриманої схеми:



Приклад 2.2.3 Знайдіть булеву функцію логічної схеми і складіть таблицю істинності для логічної схеми на рис. 2.5.

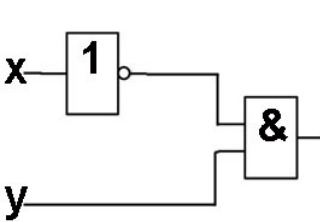


Рисунок 2.5 – Логічна схема до прикладу 2.2.1

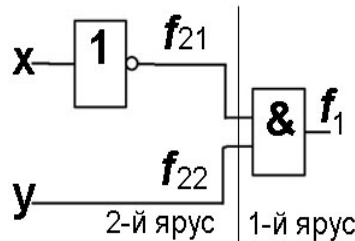


Рисунок 2.6 – Логічна схема до прикладу 2.2.1, розбита на яруси

Розв'язок

Розбиваємо логічну схему на яруси (рис. 2.6). Запишемо всі функції, починаючи з 1-го ярусу:

$$f_1 = f_{21} \wedge f_{22}; \quad f_{21} = \bar{x}; \quad f_{22} = y.$$

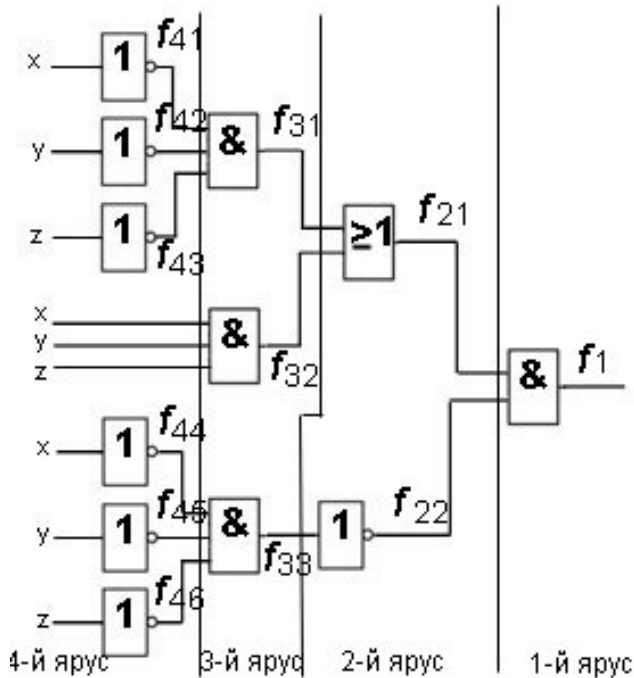
Отримуємо функцію, яку реалізує на виході логічна схема:

$$f_1 = \bar{x} \wedge y.$$

Таблиця істинності для даної логічної схеми:

x	y	f_{21}	f_{22}	f
1	1	0	1	0
1	0	0	0	0
0	1	1	1	1
0	0	1	0	0

Приклад 2.2.4 Знайдіть булеву функцію логічної схеми і складіть таблицю істинності для логічної схеми на рисунку:



Розв'язок

Розбиваємо логічну схему на яруси, що вже показано на рис. 2.7. Запишемо всі функції, починаючи з 1-го ярусу:

$$f_1 = f_{21} \wedge f_{22};$$

$$f_{21} = f_{31} \vee f_{32}; \quad f_{22} = \overline{f_{33}};$$

$$f_{31} = \overline{f_{41}} \wedge \overline{f_{42}} \wedge \overline{f_{43}};$$

$$f_{32} = x \wedge y \wedge z;$$

$$f_{41} = \overline{f_{44}} \wedge \overline{f_{45}} \wedge \overline{f_{46}};$$

$$f_{44} = \overline{x};$$

$$f_{45} = \overline{y};$$

$$f_{46} = \overline{z}.$$

Тепер запишемо всі функції, підставляючи вхідні змінні x , y , z :

$$f_{21} = (\overline{x} \wedge \overline{y} \wedge \overline{z}) \vee (x \wedge y \wedge z);$$

$$f_{22} = \overline{\overline{x} \wedge \overline{y} \wedge \overline{z}}.$$

В результаті отримаємо функцію, яку реалізує на виході логічна схема:

$$f = f_1 = \left((\overline{x} \wedge \overline{y} \wedge \overline{z}) \vee (x \wedge y \wedge z) \right) \wedge \overline{\overline{\overline{x} \wedge \overline{y} \wedge \overline{z}}}.$$

Таблиця істинності для даної логічної схеми:

x	y	z	f_{31}	f_{32}	f_{21}	f_{22}	f
1	1	1	0	1	1	1	1
1	1	0	0	0	0	1	0
1	0	1	0	0	0	1	0
1	0	0	0	0	0	1	0
0	1	1	0	0	0	1	0
0	1	0	0	0	0	1	0
0	0	1	0	0	0	1	0
0	0	0	1	0	1	0	0

Приклад 2.2.5 Побудувати логічну схему, що реалізує функцію за цією таблицею істинності:

x	y	f
1	1	0
1	0	0
0	1	1
0	0	0

Розв'язок

Розробка логічної схеми по її аналітичному опису має назву завдання *синтезу логічної схеми*.

Кожній диз'юнкції (логічній сумі) відповідає елемент "АБО", число входів якого визначається кількістю змінних в диз'юнкції. Кожній кон'юнкції (логічному добутку) відповідає елемент "І", число входів якого визначається кількістю змінних в кон'юнкції. Кожному запереченню (інверсії) відповідає елемент "НІ".

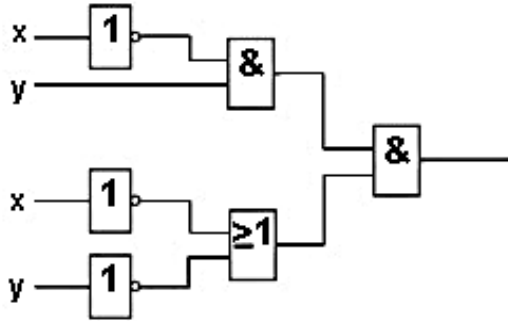
Часто розробка логічної схеми починається з визначення логічної функції, яку повинна реалізувати логічна схема. У цьому випадку дана тільки таблиця істинності логічної схеми.

Розбираємо таблицю істинності для логічної схеми. Визначаємо функцію, яка вийде на виході схеми і проміжні функції, які на вході приймають аргументи x і y . У першому рядку результатом реалізації вихідної функції при тому, що значення вхідних змінних рівні одиницям, повинен бути логічний "0", у другому рядку – при різних значеннях вхідних змінних на виході теж повинен бути логічний "0". Тому потрібно, щоб вихідна функція була кон'юнкцією (логічним добутком).

Тепер підбираємо проміжні функції. Отримуємо наступну таблицю для проміжних функцій і вихідної функції - кон'юнкції проміжних функцій:

$\bar{x} \wedge y$	$\bar{x} \vee \bar{y}$	$(\bar{x} \wedge y) \wedge (\bar{x} \vee \bar{y})$
0	0	0
0	1	0
1	1	1
0	1	0

Для побудови логічної схеми необхідно елементи, що реалізують логічні операції, зазначені в вихідній функції, розташовувати в порядку, заданої цією функцією. З виразу видно, що знадобляться 3 схеми "НІ", дві двухвходових схеми "І" та одна двухвходова схема "АБО". Відповідно до вихідної функції $(\bar{x} \wedge y) \wedge (\bar{x} \vee \bar{y})$ отримуємо наступну логічну схему:



2.3 Задачі для самостійного розв'язання

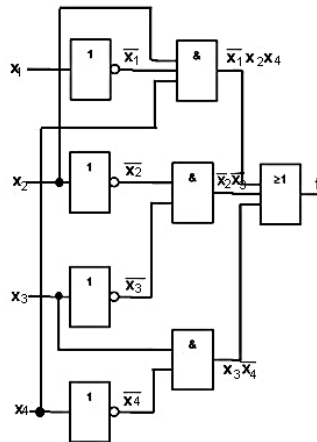
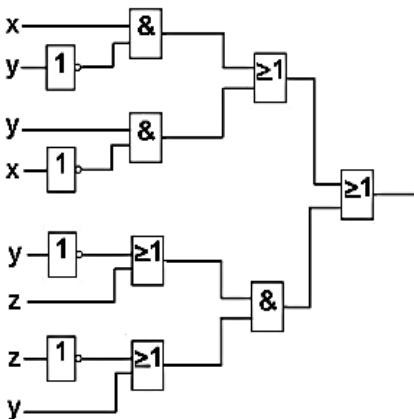
2.3.1 Скласти релейно-контактну схему, відповідну формулі $\bar{y} \& z \rightarrow x \vee y$.

2.3.2 Скласти релейно-контактні схеми, відповідні:

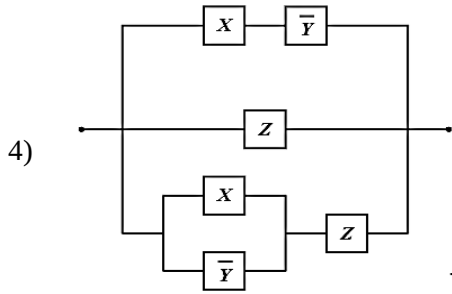
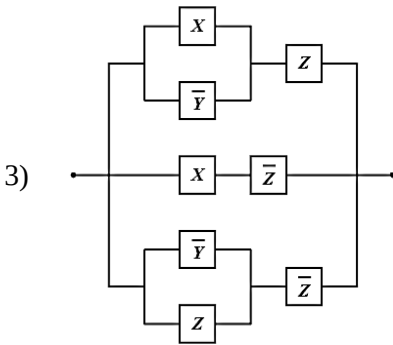
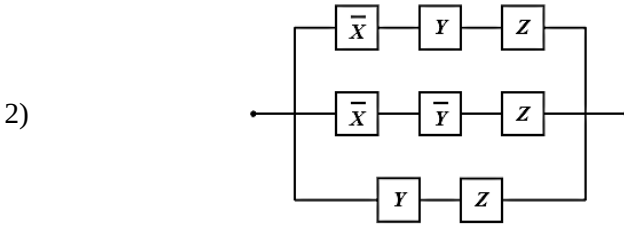
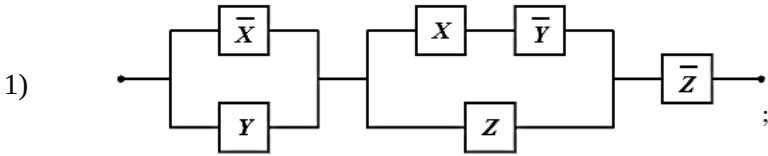
- 1) імплікації $x \rightarrow y$;
- 2) еквівалентності $x \leftrightarrow y$.

2.3.3 Побудувати в булевому базисі логічну схему, що реалізує функцію алгебри логіки $f = \bar{x}_1 x_2 x_4 \vee x_2 \bar{x}_3 \vee x_3 x_4$.

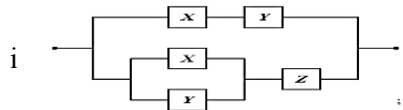
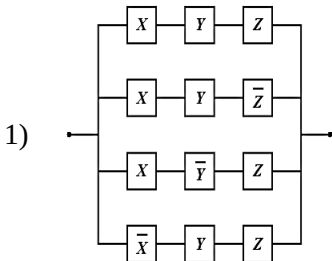
2.3.4 Знайдіть булеву функцію логічної схеми і складіть таблицю істинності для логічної схеми на рисунках:

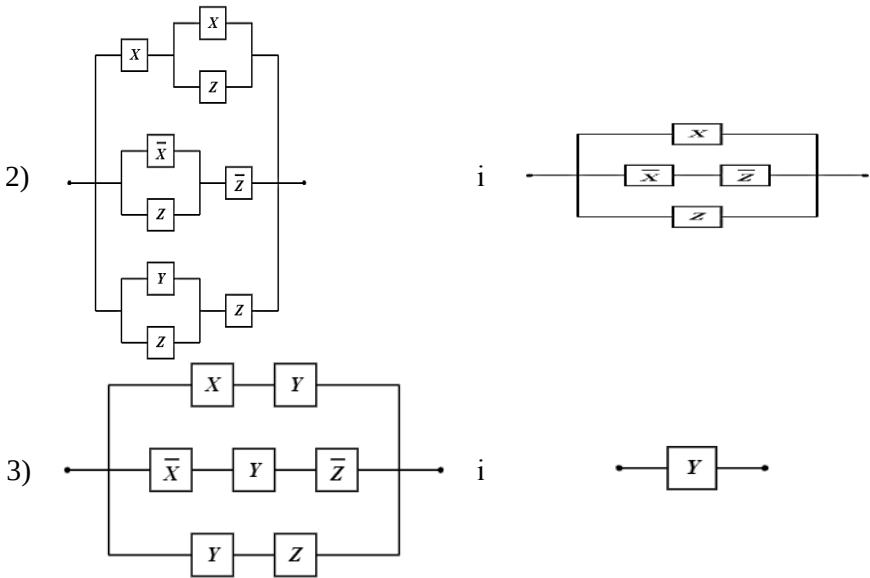


2.3.5 Спростити такі релейно-контактні схеми:

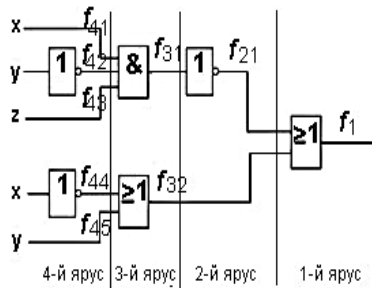


2.3.6 Довести рівнозначність наступних релейно-контактних схем:

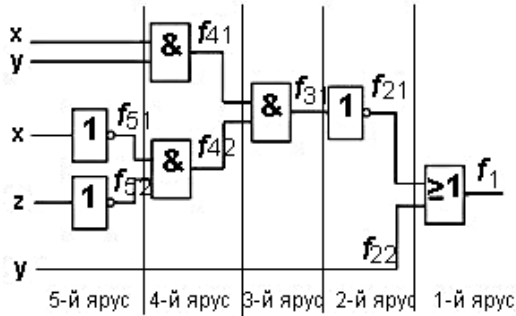




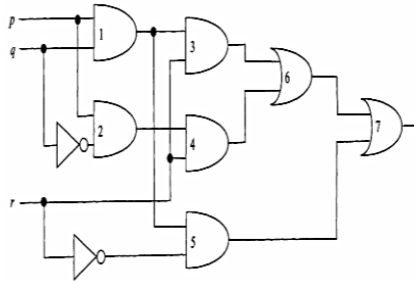
2.3.7 Знайдіть булеву функцію логічної схеми і складіть таблицю істинності для логічної схеми на рисунку:



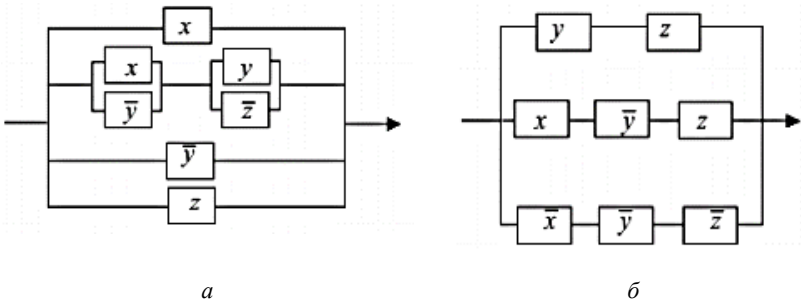
2.3.8 Знайдіть булеву функцію логічної схеми і складіть таблицю істинності для логічної схеми на рисунку:

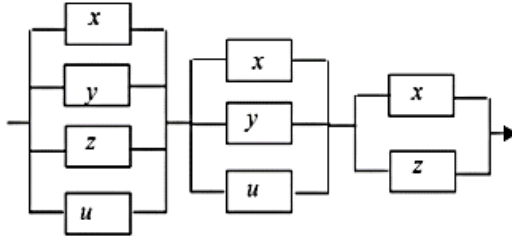


2.3.9 Знайдіть булеву функцію логічної схеми і складіть таблицю істинності для логічної схеми на рисунку (див. Додаток А):



2.3.10 Дана релейно-контактна схема. Запишіть функцію провідності і спростіть схему:





в

2.3.11 Скласти релейно-контактні схеми для формул:

- 1) $\bar{x}y \vee xyz$;
- 2) $(xy \vee \bar{z}y)(x \vee y)$;
- 3) $(x \vee y \vee \bar{y})(t \vee z)$;
- 4) $x(z \vee xt)\bar{y} \vee \bar{x}(\bar{y}t \vee zy)$;
- 5) $x \vee y \wedge z$;
- 6) $(x \vee y) \wedge z$;
- 7) $(x \vee y) \wedge (z \vee t)$;
- 8) $(x \vee y \wedge z) \wedge (a \vee b \wedge c)$;
- 9) $x \wedge (\bar{y} \wedge z \vee x \vee y)$;
- 10) $x \wedge y \wedge z \vee \overline{x \wedge y \wedge z} \vee \bar{x} \wedge y$;
- 11) $x \wedge (y \wedge z \vee \overline{y \wedge z}) \vee \bar{x} \wedge (\bar{y} \wedge z \vee y \wedge \bar{z})$;
- 12) $(x \rightarrow y) \wedge (y \rightarrow z)$;
- 13) $(x \rightarrow y) \rightarrow (\bar{x} \wedge (y \vee z))$;
- 14) $((x \rightarrow y) \rightarrow (y \rightarrow z)) \rightarrow (x \rightarrow z)$;
- 15) $(x \rightarrow (y \rightarrow z)) \rightarrow (y \rightarrow x)$;
- 16) $x \rightarrow \bar{x} \rightarrow x$;
- 17) $(x \rightarrow z) \& (z \rightarrow y)$;
- 18) $(z \rightarrow x) \& (y \rightarrow z) \rightarrow (y \rightarrow x)$.

ЗСИНТЕЗ КОМБІНАЦІЙНИХ АВТОМАТІВ. ДОСКОНАЛІ НОРМАЛЬНІ ФОРМИ

3.1 Основні поняття

Основні етапи синтезу дискретного автомата наступні:

- складання завдання на розробку у вигляді змістовного опису, де формуються вимоги керованого об'єкта до автомата, тобто опису, на який виконавчий елемент об'єкта і в якому такті повинен бути поданий мінімальний (0) або максимальний (1) рівень сигналу даного виходу автомата, щоб з урахуванням стану задаючих входів і виходів, що сприймають сигнали зворотних зв'язків, забезпечити необхідну функціонування керованого об'єкта;
- отримання формальної моделі автомата на основі змістовного опису у вигляді таблиць, графів, матриць і т. п.;
- отримання на основі моделі об'єкта вихідної або мінімальної форми аналітичного представлення оператора перетворення, що реалізується автоматом і забезпечує необхідний закон функціонування керованого об'єкта. Ця частина синтезу є основною і базується на аксіомах і постулатах Булевої (двійкової) алгебри;
- розробка принципової схеми автомата на основі отриманої його формальної моделі і використовуваної конкретної елементної бази;
- розробка і узгодження вузлів зв'язку автомата з оператором і керованим об'єктом, захист від перешкод і т. п.

Оператори, які реалізуються *комбінаційними автоматами*, встановлюють однозначний зв'язок комбінацій входних і вихідних змінних для кожного значення автоматного часу. Для створення складних комбінаційних автоматів використовують елементарні комбінаційні автомати, якими реалізується функціонально повний набір двійкових функцій. *Елементарним комбінаційним автоматом* називають автомат, який реалізує перемикальну функцію (ПФ) двох змінних. Синтез комбінаційних автоматів проводиться для статичної частини такту, тобто інерційність логічних елементів не враховується. Синтез комбінаційних автоматів починається зі змістовного опису функціонування об'єкта і складання таблиці істинності, яка відображає однозначну відповідність дискретних станів входів і виходів об'єкта керування.

Наприклад, для циклограми, наведеної на рис. 3.1, може бути складена таблиця істинності (Табл. 3.1). З цієї таблиці вихідний сигнал y виражається через вхідні за допомогою рівняння алгебри логіки (Булевого виразу).

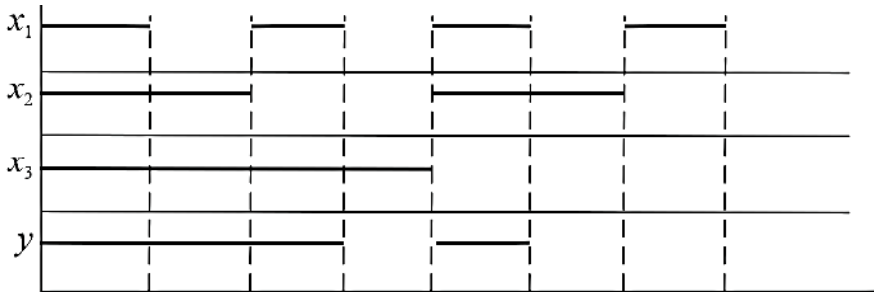


Рисунок 3.1– Циклограма

Таблиця 3.1 – Таблиця істинності

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

Задача *аналізу* комбінаційної схеми – за заданою схемою визначити її оператор.

Задача *синтезу* – за заданим оператором, тобто системою Булевих функцій, і заданим елементним базисом, тобто набіром функціональних елементів побудувати схему в заданому елементному базисі, яку реалізовує заданий оператор.

Для отримання аналітичного виразу, що визначає необхідну структуру автомата, записуються вихідні форми перемикальної функції y у вигляді *досконалої диз'юнктивної* (ДДНФ) або *кон'юнктивної* (ДКНФ) нормальної форми.

Нормальна форма логічної формули не містить знаків імплікації, еквіваленції і заперечення неелементарних формул.

Існує два види нормальної форми: *кон'юнктивна нормальна форма*, тобто кон'юнкція декількох диз'юнкцій (КНФ) і *диз'юнктивна*

нормальна форма, тобто диз'юнкція кількох кон'юнкція (ДНФ), наприклад:

$$\text{КНФ: } (x \vee \bar{y} \vee z) \wedge (y \vee z);$$

$$\text{ДНФ: } (x \wedge \bar{y} \wedge z) \vee (y \wedge z).$$

Досконала дон'юнктивна НФ - кон'юнкція диз'юнкцій, причому в кожній диз'юнкції (в кожній дужки) присутні всі змінні, що входять в формулу, або їх заперечення, немає однакових диз'юнкцій, в кожній диз'юнкції немає однакових доданків, наприклад:

$$\text{ДКНФ: } (x \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z).$$

Досконала диз'юнктивна НФ - диз'юнкція кон'юнкцій, причому в кожній кон'юнкції (в кожній дужці) присутні всі змінні, що входять в формулу, або їх заперечення, немає однакових кон'юнкцій, в кожній кон'юнкції немає однакових доданків, наприклад:

$$\text{ДДНФ: } (x \wedge y \wedge \bar{z}) \vee (x \wedge \bar{y} \wedge z).$$

Диз'юнктивна нормальна форма являє собою диз'юнкцію (суму) мінтермів, а СКНФ – кон'юнкцію (добуток) макстермів.

Мінтермом або *конституентною одиницею* називається логічна функція, що приймає значення 1 тільки на одному наборі змінних. Утворюється як кон'юнкція всіх вхідних змінних з запереченням тих, які в даному наборі дорівнюють нулю. Число мінтермів дорівнює числу наборів.

Макстермома або *конституентною нуля* називається логічна функція, що приймає значення 0 тільки на одному наборі змінних. Утворюється як диз'юнкція вхідних змінних, де змінні даного набору, рівні 1, взяті з запереченням. Число макстермів дорівнює числу наборів.

3.2 Правила побудови ДДНФ і ДКНФ за таблицею істинності

При записи ДДНФ складають суму добутків аргументів (x_1, x_2, x_3) для тих наборів, на яких функція y приймає значення 1.

Якщо аргумент в наборі дорівнює 1, то він записується без інверсії, якщо аргумент дорівнює 0 – то з інверсією. Так для умов, представлених в Табл. 1.2, отримаємо ДДНФ:

$$y = x_1 x_2 x_3 + \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3. \quad (3.1)$$

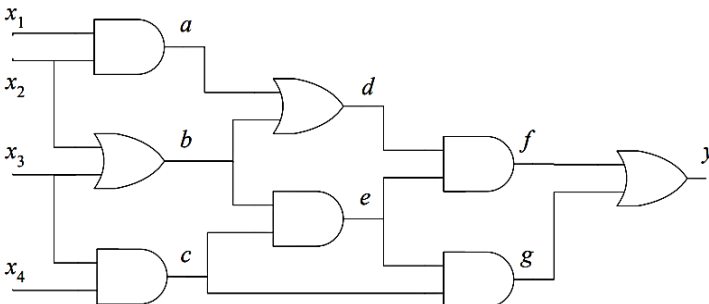
Запис в ДДНФ застосовують у випадку, коли функція має більше одиниць, ніж нулів. Коли ж більше нулів, доцільно застосувати ДКНФ, склавши добуток сум аргументів для тих наборів, на яких функція приймає значення 0. Якщо аргумент в наборі дорівнює 1, то він записується з інверсією, якщо аргумент дорівнює 0 – то без інверсії. Для умов, представлених в Табл. 1.2, отримаємо СКНФ:

$$y = (x_1 + x_2 + \bar{x}_3) \cdot (x_1 + \bar{x}_2 + x_3) \cdot (\bar{x}_1 + x_2 + x_3) \cdot (x_1 + x_2 + x_3). \quad (3.2)$$

На підставі отриманих рівнянь можна побудувати схеми керування на контактних або безконтактних елементах. Однак, з огляду на складність рівнянь, схеми будуть містити велику кількість елементів. Тому необхідно застосувати методи синтезу автоматів з мінімальною кількістю елементів. До них відносяться методи Квайна і Карно.

3.3 Приклади розв'язання задач

Приклад 3.3.1 Проаналізуємо комбінаційну схему:



Розв'язок

Припишемо виходам елементів символи функцій і, починаючи від виходу, будемо будувати суперпозицію цих функцій до тих пір,

поки не залишаться тільки символи вхідних змінних; одночасно будемо спрощувати вираз, використовуючи закони Булевої алгебри:

$$\begin{array}{llll} a = x_1 x_2; & d = a \vee b; & f = de; & \\ b = x_1 \vee x_2; & e = bc; & g = ce; & y = f \vee g. \\ c = x_2 x_3; & & & \end{array}$$

Отже, дану схему можна замінити одним кон'юнктором.

Функція, приписана полюсу схеми, називається *функцією, яка реалізується схемою на цьому полюсі*. Функція, приписана виходу схеми, називається *функцією, яка реалізується схемою*.

Система функцій, реалізованих схемою, називається *оператором схеми*. Цей оператор описує поведінку схеми.

Приклад 3.3.2 Відновіть логічну функцію за її таблицею істинності:

x	y	z	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Розв'язок

ДДНФ складається на основі таблиці істинності за таким *правилом*:

для кожного набору змінних, при якому функція дорівнює 1, записується добуток, в якому з запереченням беруться змінні, що мають значення «0».

Отримуємо ДДНФ: $f(x, y, z) = \overline{x}\overline{y}z + \overline{x}y\overline{z} + x\overline{y}\overline{z} + x\overline{y}z + xyz$.

ДКНФ складається на основі таблиці істинності за *правилом*:

для кожного набору змінних, при якому функція дорівнює 0, записується сума, в якій з запереченням беруться змінні, що мають

значення 1.

Отримуємо ДКНФ: $f(x, y, z) = (x + \bar{y} + z) \cdot (x + \bar{y} + \bar{z}) \cdot (\bar{x} + \bar{y} + z)$.

Приклад 3.3.3 Для формули $f(x, y) \equiv x \& (x \rightarrow y)$ знайти ДДНФ шляхом використання рівнозначних перетворень і таблиці істинності.

Розв'язок

Виконаємо рівнозначні перетворення. В результаті отримаємо:

$$f(x, y) \equiv x \& (x \rightarrow y) \equiv x \& (\bar{x} \vee y) \equiv x \& \bar{x} \vee x \& y \equiv x \& y \equiv \text{ДДНФ } f.$$

Складемо таблицю істинності:

x	y	$x \rightarrow y$	$x \& (x \rightarrow y)$
1	1	1	1
1	0	0	0
0	1	1	0
0	0	1	0

З таблиці істинності слідує: ДДНФ $f \equiv x \& y$.

Приклад 3.3.4 Для формули $f(x, y) \equiv x \& (x \rightarrow y)$ знайти ДКНФ шляхом використання рівнозначних перетворень і таблиці істинності.

Розв'язок

Виконаємо рівнозначні перетворення. В результаті отримаємо:

$$\begin{aligned} f(x, y) \equiv x \& (x \rightarrow y) &\equiv x \& (\bar{x} \vee y) \equiv (x \vee y \& \bar{y}) \& (\bar{x} \vee y) \equiv \\ &\equiv (x \vee y) \& (\bar{x} \vee y) \equiv \text{ДКНФ } f. \end{aligned}$$

Складемо таблицю істинності:

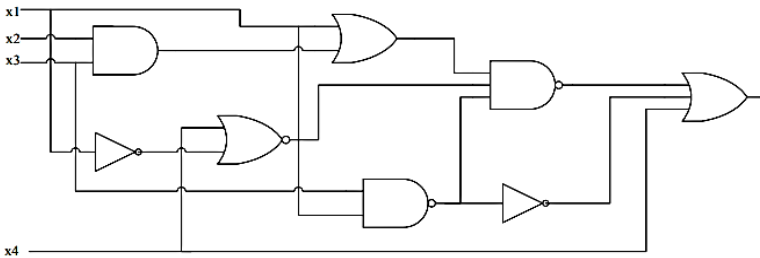
x	y	$x \& (x \rightarrow y)$	$\overline{x \& (x \rightarrow y)}$
1	1	1	0
1	0	0	1
0	1	0	1
0	0	0	1

З таблиці істинності слідує:

$$\begin{aligned} \text{ДКНФ } f &\equiv \overline{\overline{\text{ДДНФ } f}} = \overline{x \& \bar{y} \vee x \& y \vee \bar{x} \& y} = \overline{x \& \bar{y}} \& \overline{x \& y} \& \overline{\bar{x} \& y} = \\ &\equiv (\bar{x} \vee y) \& (x \vee \bar{y}) \& (x \vee y). \end{aligned}$$

3.4 Задачі для самостійного розв'язання

3.4.1 Провести аналіз комбінаційної схеми. Синтезувати якомога простішу комбінаційну схему з тим же оператором.



3.4.2 Знайти ДДНФ для тотожно істинної формули, що містить: 1) одну змінну; 2) дві змінних.

3.4.3 Найді ДКНФ для тотожно помилкової формули, що містить: 1) одну змінну; 2) дві змінних.

3.4.4 Для основних логічних операцій знайти: 1) ДДНФ; 2) ДКНФ.

3.4.5 Побудувати ДДНФ і ДКНФ для функцій f , для яких задана таблиця істинності:

x	y	z	f_1	f_2	f_3	f_4	f_5
0	0	0	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	0	0	0
0	1	1	0	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	0	0	0
1	1	0	0	0	0	0	0
1	1	1	1	0	0	1	1

3.4.6 Надати простіший вид формулами, які мають такі досконалі

нормальні форми:

$$1) x \& y \vee x \& \bar{y} \& \bar{x} \vee y;$$

$$2) (x \vee \bar{y}) \& (\bar{x} \vee y) \& (\bar{x} \vee \bar{y});$$

$$3) x \& y \& z \vee x \& \bar{y} \& z \vee \bar{x} \& y \& z;$$

$$4) x \& y \& z \vee x \& \bar{y} \& z \vee x \& y \& \bar{z} \vee x \& \bar{y} \& \bar{z} \vee \bar{x} \& y \& z.$$

3.4.7 Знайдіть ДДНФ функцій за допомогою рівнозначних перетворень:

$$1) f(x, y) = \bar{x} \vee \bar{y};$$

$$2) f(x, y, z) = (\bar{x} \rightarrow \bar{y}) \rightarrow (yz \rightarrow xz);$$

$$3) f(x, y, z) = (\overline{x \oplus y}) \sim z;$$

$$4) f(x, y, z) = x \oplus \overline{xyz};$$

$$5) f(x, y, z) = \overline{(\bar{x} \vee \bar{y}z) \oplus (z \rightarrow x)};$$

$$6) f(x, y, z, t) = \bar{x} \vee z \bar{y}t.$$

3.4.8 Знайдіть СКНФ функцій за допомогою рівнозначних перетворень:

$$1) f(x, y) = \bar{x} \rightarrow y;$$

$$2) f(x, y, z) = xy \vee yz \vee \bar{z};$$

$$3) f(x, y, z) = x \sim yz.$$

3.4.9 Для наступних формул знайти ДДНФ і ДКНФ шляхом використання рівнозначних перетворень і таблиць істинності:

$$1) f \equiv x \vee \bar{z} \rightarrow y \& z;$$

$$2) f \equiv \bar{x} \rightarrow z \rightarrow \overline{\bar{y} \rightarrow x};$$

$$3) f \equiv x \rightarrow y \rightarrow (y \rightarrow x);$$

$$4) f \equiv x \& (y \& z \rightarrow x \& y);$$

$$5) f \equiv x \rightarrow y \rightarrow \bar{x} \rightarrow \bar{y} \rightarrow \bar{z} \rightarrow z;$$

$$6) f \equiv x \rightarrow y \rightarrow (z \rightarrow \bar{x}) \rightarrow (\bar{y} \rightarrow \bar{z});$$

$$7) f \equiv x(yz \rightarrow xy);$$

$$8) f \equiv x(x \rightarrow y);$$

$$9) f \equiv (x \rightarrow y) \rightarrow (y \rightarrow x);$$

$$10) f \equiv (x \vee \bar{z}) \rightarrow y \wedge z;$$

$$11) f \equiv (x \vee \bar{y} \rightarrow x \wedge z) \rightarrow (\overline{x \rightarrow \bar{x}}) \vee y \wedge \bar{z};$$

$$12) f \equiv (a \cdot b \rightarrow b \cdot c) \rightarrow ((a \rightarrow b) \rightarrow (c \rightarrow b)).$$

3.4.10 Використовуючи критерії тотожної істинності і тотожної хибності, встановити, до якого класу відносяться такі формули:

$$1) f \equiv x \vee y \rightarrow z;$$

$$2) f \equiv x \rightarrow y \rightarrow (y \rightarrow x);$$

$$3) f \equiv x \rightarrow y \rightarrow \bar{x} \& y \vee \bar{y};$$

$$4) f \equiv (x \rightarrow y \& z) \& \overline{y \vee z \rightarrow x}.$$

3.4.11 Знайдіть ДДНФ функцію f , спростіть її, за спрощеною ДНФ побудуйте релейно-контактну схему:

$$1) f(x, y) = (x \vee y)(\bar{x} \vee y);$$

$$2) f(x, y, z) = (x \oplus \bar{z}) \vee (xy \rightarrow z).$$

4 ПИТАННЯ ДО МОДУЛЬНОГО КОНТРОЛЮ І

1. Що називають дискретним автоматизованим пристроєм?
2. Що представляє собою циклограма?
3. Які ви знаєте базові логічні функції та їх таблиці істинності?
4. Назвати основні закони дискретної математики (Булевої алгебри).
5. В чому полягає синтез схеми керування?
6. В чому полягає аналіз схеми керування?
7. Що представляє собою релейно-контактна схема?
8. Назвати основні задачі теорії релейно-контактних схем?
9. Що таке логічний елемент?
10. В чому полягає завдання аналізу логічної схеми?
11. Що представляє собою функція провідності?
12. Що представляють собою комбінаційні пристрої / автомати?
13. Які основні етапи синтезу дискретного автомата?
14. Що таке диз'юнктивна (ДДНФ) та кон'юнктивна (ДКНФ) форми?
15. Назвіть правила побудови ДДНФ і ДКНФ за таблицею істинності.

5 СИНТЕЗ КОМБІНАЦІЙНИХ АВТОМАТІВ. МЕТОД КВАЙНА

5.1 Основні поняття

Цей метод базується на поетапному спрощенні досконалої диз'юнктивній нормальній форми за допомогою операцій неповного склеювання і поглинання.

Метод Квайна складається з наступних етапів:

- за таблицею істинності записується ДДНФ;
- в отриманій ДДНФ відшуковуються суміжні мінтерми (відрізняються знаком тільки одного аргументу). Склеюючись за цим аргументом, суміжні мінтерми утворюють *імпліканти* – кон'юнкції з числом співмножників, меншим на один. Імпліканти в свою чергу можуть склеюватися між собою. Імпліканти, що не склеюються, називаються *простими*. Операція (повного) склеювання визначається співвідношенням:

$$XY \vee X\bar{Y} = X, \quad (5.1)$$

де X – однакова частина виразу в обох кон'юнкціях формули;

Y – змінна, яка в різних кон'юнкціях має протилежні значення.

- отримана скорочена ДДНФ приводиться до тупикової форми шляхом:

- усунення *надлишкових* імплікант (надлишкової називається імпліканта, яка «покриває» мінтерми, вже покриті іншими імплікантами);

- застосування закону поглинання для простих імплікант і мінтермів, що залишилися:

$$XZ \vee X = X, \quad (5.2)$$

де Z – частина кон'юнкції, яка в одній кон'юнкції присутня, а в іншій – відсутня.

Кажуть, що член XY поглинається членом X .

Операція неповного склеювання, що застосована в методі Квайна, визначається формулою:

$$XY \vee X\bar{Y} = X \vee XY \vee X\bar{Y}. \quad (5.3)$$

- проведення випробування окремих членів. Для цього

відкидається одна з складових, а в решту виразу підставляються значення аргументів, що звертають випробовуваний член в 1. Якщо решта при цьому тотожно дорівнює 1, то випробовуваний член зайвий, і навпаки.

Якщо в досконалої диз'юнктивній нормальній формі логічної функції провести всі операції неповного склеювання, а потім всі операції поглинання, то вийде скорочена диз'юнктивна нормальна форма цієї функції, тобто диз'юнкція всіх її простих імплікант.

З теореми Квайна слідує: *якщо функція задана в довільній формі, то її слід перетворити в досконалий ДНФ, застосувати функцію розгортання, і тільки після цього проводити операції склеювання і поглинання.*

Метод Квайна застосовується до числа аргументів не більше 6.

Наприклад, застосовуючи закон тавтології, запишемо вираз (3.1) у вигляді

$$y = x_1 x_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 \bar{x}_3 + x_1 x_2 \bar{x}_3. \quad (5.4)$$

Застосовуючи закон склеювання першого мінтерма з четвертим, другого з п'ятим, третього з шостим, і застосовуючи розподільний закон, отримаємо

$$y = x_2 x_3 + x_1 x_3 + x_1 x_2 = x_2 x_3 + x_1 (x_3 + x_2). \quad (5.5)$$

За отриманим рівнянням синтезуємо релейну схему керування (рис. 5.1).

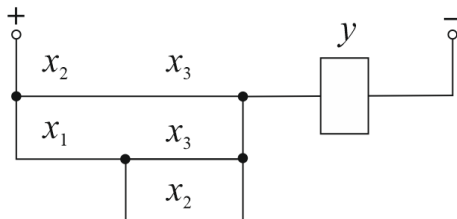


Рисунок 5.1 – Релейна схема керування

При цьому враховуємо, що прямому значенню аргументу відповідає замикаючий контакт, а інверсному – розмикаючий. Логічному добутку відповідає послідовне з'єднання контактів, а логічній сумі – паралельне.

5.2 Приклад розв'язання задач

Розглянемо мінімізацію логічної функції за методом Квайна, заданої у вигляді:

$$\begin{aligned} f(x_1, x_2, x_3, x_4) = & \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot x_4 + \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot \overline{x_4} + \\ & + \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot x_4 + \overline{x_1} \cdot x_2 \cdot x_3 \cdot x_4 + \\ & + x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} + x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4 + \\ & + x_1 \cdot x_2 \cdot \overline{x_3} \cdot \overline{x_4} + x_1 \cdot x_2 \cdot \overline{x_3} \cdot x_4. \end{aligned}$$

Розв'язок

Завдання мінімізації функції відповідно до алгоритму Квайна вирішуємо в кілька етапів.

Етап 1. Знаходження первинних імплікант.

В формулі кожний терм пронумеруємо для відстежування попарного перебору всіх термів на предмет застосування правила склеювання:

$$\begin{aligned} f = & \underbrace{\overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot x_4}_0 + \underbrace{\overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot \overline{x_4}}_1 + \underbrace{\overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot x_4}_2 + \underbrace{\overline{x_1} \cdot x_2 \cdot x_3 \cdot x_4}_3 + \\ & + \underbrace{x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4}}_4 + \underbrace{x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4}_5 + \underbrace{x_1 \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4}}_6 + \underbrace{x_1 \cdot \overline{x_2} \cdot x_3 \cdot x_4}_7. \end{aligned}$$

Виконуємо склеювання (можна у вигляді таблиці). Терм0 склеюється тільки з термом 3 (по змінній x_2) і з термом 5 (по змінній x_1), терм1 з термом2 і т. д. В результаті отримуємо:

$$\begin{array}{ll} 0-3: & \overline{x_1} \cdot x_3 \cdot x_4 \\ 0-5: & \overline{x_2} \cdot x_3 \cdot x_4 \\ 1-2: & \overline{x_1} \cdot x_2 \cdot \overline{x_3} \\ 1-6: & x_2 \cdot \overline{x_3} \cdot \overline{x_4} \\ 2-3: & \overline{x_1} \cdot x_2 \cdot x_4 \\ 2-7: & x_2 \cdot \overline{x_3} \cdot x_4 \\ 4-5: & x_1 \cdot \overline{x_2} \cdot x_4 \\ 4-7: & x_1 \cdot x_3 \cdot x_4 \\ 6-7: & x_1 \cdot x_2 \cdot \overline{x_3} \end{array}$$

Це знижує ранг термів, що входять в ДДНФ.

Зауважимо, що результатом склеювання є завжди

елементарний добуток, що представляє собою загальну частину термів, що склеюються.

Після першої ітерації склеювання отримана скорочена формула з термів третього рангу.

$$f = \underbrace{\overline{x_1} \cdot \overline{x_3} \cdot \overline{x_4}}_{0-3} + \underbrace{\overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4}}_{0-5} + \underbrace{\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}}_{1-2} + \underbrace{\overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4}}_{1-6} + \\ + \underbrace{\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_4}}_{2-3} + \underbrace{\overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4}}_{2-7} + \underbrace{\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_4}}_{4-5} + \underbrace{\overline{x_1} \cdot \overline{x_3} \cdot \overline{x_4}}_{4-7} + \underbrace{\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}}_{6-7}.$$

Якщо номер якого-небудь вихідного терма відсутній в цій формулі, тоді він повинен бути введений в цю формулу через знак диз'юнкції (він обов'язково входить в кінцевий результат!).

На практиці при виконанні операцій неповного склеювання на кожному етапі можна не писати члени, які беруть участь в цих операціях, а писати тільки результати всіляких повних склеювань і терми, які не приймають участь в жодному склеюванні.

Ця процедура знижує ранг термів і триває до тих пір, поки не залишиться жодного члена, який допускає поглинання з будь-яким іншим термом. Терми, які піддалися поглинанню, відзначаються. Невідмічені терми є *первинними імплікантами*.

Далі проводимо склеювання одержаних термів, які мають однаковий ранг і які містять однакові змінні. Тобто, 1-2 і 2-7, 1-6 і 2-7 по змінній x_4 . Отже:

$$(1-2) \vee (2-7): \overline{x_2} \cdot \overline{x_3} \\ (1-6) \vee (2-7): \overline{x_2} \cdot \overline{x_3}.$$

Склеювання триває доти, поки буде неможливо застосувати правило поглинання. У нашій задачі можна дійти до первинної імпліканти другого рангу $\overline{x_2} \cdot \overline{x_3}$. Тобто

$$f = \underbrace{\overline{x_2} \cdot \overline{x_3}}_{(1-3)/(2-7)} + \underbrace{\overline{x_2} \cdot \overline{x_3}}_{(1-6)/(2-7)}.$$

Мінтерми, що повторюються, можуть бути видалені з формули згідно з законом повторювання.

Якщо більше немає чого склеювати, то на цьому перший етап спрощення закінчується.

Етап 2. Встановлення позначок.

Складаємо таблицю, кількість рядків якої дорівнює кількості отриманих первинних імплікант, а кількість стовпців збігається з кількістю мінтермів ДДНФ. Якщо в деякий мінтерм ДДНФ входить будь-яка з первинних імплікант, то на перетині відповідного стовпця і рядка ставиться відмітка (див. Табл. 5.1).

Таблиця 5.1 – Таблиця для етапу 2

Вихідні терми	$\bar{x}_1 \bar{x}_2 \bar{x}_3 x_4$	$\bar{x}_1 x_2 \bar{x}_3 \bar{x}_4$	$\bar{x}_1 x_2 \bar{x}_3 x_4$	$\bar{x}_1 x_2 x_3 x_4$	$x_1 \bar{x}_2 \bar{x}_3 x_4$	$x_1 \bar{x}_2 x_3 x_4$	$x_1 x_2 \bar{x}_3 \bar{x}_4$	$x_1 x_2 \bar{x}_3 x_4$
первинні імплікант и								
$\bar{x}_1 \bar{x}_3 x_4$	✓			✓				
$\bar{x}_2 \bar{x}_3 x_4$	✓					✓		
$\bar{x}_1 x_2 x_4$			✓	✓				
$x_1 \bar{x}_2 \bar{x}_4$					✓	✓		
$x_1 \bar{x}_3 x_4$					✓			✓
$x_2 \bar{x}_3$		✓	✓				✓	✓

Етап 3. Знаходження істотних імплікант.

Якщо в будь-якому з стовпців таблиці є тільки одна відмітка, то первинна імпліканта у відповідному рядку є суттєвою, оскільки без неї не буде отримана вся множина заданих мінтермів. У таблиці суттєвої імплікантою є терм $x_2 \cdot \bar{x}_3$. Стовпці, які відповідають суттєвим імплікантам, з таблиці викреслюються.

Етап 4. Викреслювання зайвих стовпців.

Після третього етапу в результаті викреслювання стовпців 2, 3, 7 і 8 отримуємо Таблицю 5.2. Якщо в таблиці є два стовпці, в яких позначки в однакових рядках, то один з них викреслюється. Покриття стовпчика, що залишився, здійснюватиме відкинутий мінтерм. У прикладі такого випадку немає.

Таблиця 5.2 – Таблиця для етапу 4

Первинні імпліканти	Вихідні терми			
	$\overline{x_1} \overline{x_2} x_3 x_4$	$\overline{x_1} x_2 \overline{x_3} x_4$	$x_1 \overline{x_2} \overline{x_3} x_4$	$x_1 \overline{x_2} x_3 \overline{x_4}$
$\overline{x_1} x_3 x_4$	✓	✓		
$\overline{x_2} x_3 x_4$	✓			✓
$\overline{x_1} x_2 x_4$		✓		
$x_1 \overline{x_2} x_4$			✓	✓
$x_1 \overline{x_3} x_4$			✓	

Етап 5. Викреслювання зайвих первинних імплікант.

Якщо після викреслювання декількох стовпців на етапі 4 в таблиці з'являються рядки, в яких немає жодної позначки, то первинні імпліканти, які відповідають цим рядкам, виключаються з подальшого розгляду, оскільки вони не покривають мінтерм тих, що залишилися.

Етап 6. Вибір мінімального покриття.

Вибирається в останній таблиці така сукупність первинних імплікант, яка містить позначки у всіх стовпцях. При декількох можливих варіантах такого вибору віддається перевага варіанту покриття з мінімальним сумарним числом букв в імплікантах, що створюють покриття. Цю вимогу задовольняють первинні імпліканти $\overline{x_1} \cdot x_3 \cdot x_4$ і $x_1 \cdot \overline{x_2} \cdot x_4$.

Отже, мінімальна форма заданої функції буде складатися з суми істотних імплікант і первинних імплікант, які покривають мінтерм, що залишилися:

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

Як мінімальна ДНФ вибирається *тупикова*, яка має найменше число входжень змінних.

5.3 Задачі для самостійного розв'язання

5.3.1 Мінімізувати функцію методом Квайна:

1) $f = xyz + yz + \overline{xyt} + \overline{yzt}$;

2) $f = \overline{\overline{xyz}} + \overline{\overline{xyz}} + \overline{\overline{xyz}} + \overline{xyz}$;

3) $f = \overline{\overline{x(y \vee z) \wedge (x \vee yz)}}$;

4) $f = \overline{\overline{xy}} + \overline{\overline{xy}} + \overline{xy} + \overline{x} + \overline{y}$;

5) $f = \overline{\overline{\overline{xyz}}} + \overline{\overline{\overline{xyz}}} + \overline{\overline{\overline{xyz}}} + \overline{xyz}$;

6) $f = x_1 \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot \overline{x_3}$;

7) $f = \overline{xyz} + \overline{xyz} + \overline{xyz} + \overline{xyz}$.

5.3.2 Мінімізувати функцію, яка задана таблицею істинності методом Квайна:

№	x_1	x_2	x_3	x_4	f_1	f_2	f_3	f_4
0	0	0	0	0	1	0	1	0
1	0	0	0	1	1	1	1	0
2	0	0	1	0	1	0	0	0
3	0	0	1	1	1	1	0	1
4	0	1	0	0	0	0	1	1
5	0	1	0	1	0	1	0	0
6	0	1	1	0	1	0	1	1
7	0	1	1	1	1	1	1	1
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	0	1	0
10	1	0	1	0	0	0	0	0
11	1	0	1	1	1	0	0	0
12	1	1	0	0	0	0	0	1
13	1	1	0	1	0	0	0	0
14	1	1	1	0	0	1	1	0
15	1	1	1	1	1	1	1	0

6 СИНТЕЗ КОМБІНАЦІЙНИХ АВТОМАТІВ. МЕТОД КАРТ КАРНО

6.1 Основні поняття

Найбільшого поширення для синтезу дискретних автоматів отримав метод карт Карно. Суть його полягає в табличному поданні мінтермів ДДНФ і графічній інтерпретації їх склеювання.

Карта Карно представляє собою плоску розгортку булева куба, при цьому клітини, які знаходяться в крайніх стовпчиках і рядках таблиці, сусідять між собою. Карта Карно – це таблиця з числом осередків, що дорівнює кількості наборів аргументів. Загальна кількість клітин карти Карно для логічної функції n змінних становить 2^n . У кожен клітинку можна вписати один набір (мінтерм), причому суміжні мінтерми повинні бути в сусідніх клітках. Це забезпечується спеціальним підбором мінтермів для кожного осередку. Щоб спростити цю задачу, застосовують різні способи маркування сторін карт Карно. Нижче описані два найпоширеніші способи.

На кожній зі сторін карти половина клітин відповідає одиничному значенню одного аргументу, а половина нульового. На протилежних гранях аналогічне маркування для іншого аргументу робиться зі зрушенням на одну клітку. У клітинах тих наборів, де функція приймає одиничне значення, пишеться 1, де нульове – 0.

При другому способі аргументи розбиваються на дві групи з рівним або нерівним числом аргументів. Для однієї групи x_1x_2 (перші номери) відводяться стовпці, для іншої x_3x_4 (останні номери) – рядки. Пряме або інверсне значення аргументу визначається двійковим цифровим маркуванням стовпців.

Охоплюючи поодинокі клітини *контурами*, проводимо склеювання суміжних мінтермів. Контуру з двох клітин відповідає одноступеневе склеювання, з чотирьох – двоступеневе і т. д. Число клітин в контурі може бути *тільки парне*. В утворених імплікантах (простих) залишаються змінні, які в контурі не змінювали знак. Їх число дорівнює $n-k$, де n – число аргументів. Щоб функція була мінімальною, необхідно накрити все одиниці мінімальним числом контурів максимальної площі (почати з виділення контурів найбільш

можливого розміру, потім виділяти всі допустимі контури з меншим розміром, не покриті повністю будь-яким уже виділеним інтервалом більшого розміру). Контури можуть перетинатися. У тупиковій алгебраїчній формі, отриманій з карти Карно, це означає:

- число контурів дорівнює числу складових;
- площа контуру визначає число співмножників в складовому;
- перетин контурів означає тавтологію мінтермів.

Для мінімізації використовуються контури, що містять одиниці (для отримання ДНФ) або нулі (для отримання КНФ).

Контури не повинні містити неоднакові значення.

Кожен контур дає одну елементарну кон'юнкцію (для ДНФ) або диз'юнкцію (для КНФ).

За постійними змінними інтервалів записують елементарні кон'юнкції (диз'юнкції).

6.2 Приклад розв'язання задач

За заданою таблицею істинності логічної функції чотирьох аргументів (Табл. 5.3) мінімізувати функцію, застосовуючи метод карт Карно та синтезувати РКС.

x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
y	0	0	1	0	1	1	1	0	1	0	1	1	1	1	1	0

Розв'язок

Складаємо карту Карно у вигляді таблиці, стовпці і рядки якої відповідають всім можливим наборам значень аргументів x_1, x_2, x_3, x_4 . Для групи x_1, x_2 відводимо стовпці, для групи x_3, x_4 – рядки. Набори розташовуємо в такому порядку, щоб в сусідніх клітках комбінації змінних x_1, x_2, x_3, x_4 відрізнялися значенням тільки однієї змінної.

Кожному набору значень змінних відповідає одна клітинка таблиці Карно (рис. 6.1). Осередки (клітини) заповнюються відповідно до таблиці істинності. Клітини, що містять одиниці, потрібно обвести

контурами. У контур можна включати 2^k клітин ($k = 1, 2, 3 \dots$).

	00	01	11	10
00	0	0	0	$\bar{1}$
01	1	1	0	1
11	1	1	0	1
10	1	0	1	1

Рисунок 6.1 – Карта Карно

Мінімальна діз'юнктивна форма записується як сума добутків. Число доданків дорівнює числу контурів, а число співмножників в кожному доданку дорівнює числу змінних, які не змінюють свого значення в даному контурі. Причому значенням змінної 1 в добутку відповідає сама змінна, а значенням 0 – її заперечення (інверсія). У відповідності зі сказаним, отримаємо для даної карти Карно вираз:

$$y = x_2 \bar{x}_3 + \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 x_3 + x_3 \bar{x}_4$$

За отриманим рівнянням синтезуємо релейну схему керування (рис. 6.2).

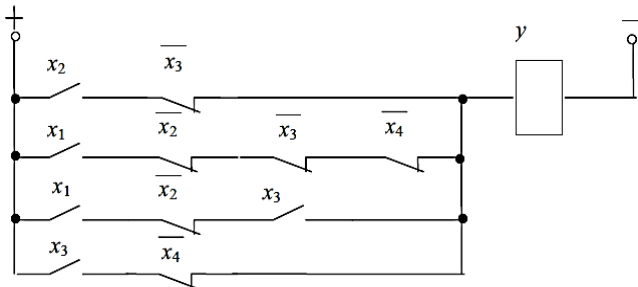


Рисунок 6.2 – Релейна схема керування

При синтезі ряду дискретних автоматів мають місце частково

визначена функції. Частково *визначена функція* – це така функція, значення якої задано не для всіх наборів аргументів. Це означає, що деякі набори аргументів неможливі або неприпустимі в реальних умовах. У таблиці істинності і карті Карно такі набори позначають знаками невизначеності. Якщо ці набори нереальні, значить, ПФ можна визначити з міркувань її найкращої мінімізації, тобто проставити в клітинах 0 або 1 замість знаку невизначеності.

6.3 Задачі для самостійного розв'язання

6.3.1 Мінімізувати функцію, застосовуючи метод карт Карно:

$$1) f(x_1, x_2, x_3) = x_2 \cdot \overline{x_3} + \overline{x_1} \cdot x_2 + \overline{x_1} \cdot x_3 + x_1 \cdot \overline{x_2};$$

$$2) f(x_1, x_2, x_3) = x_1 \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot \overline{x_3} + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3;$$

$$3) f(x_1, x_2, x_3) = x_1 \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot \overline{x_3} + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3;$$

$$4) f(x_1, x_2, x_3) = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot \overline{x_3} + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3;$$

$$5) f(x_1, x_2, x_3, x_4) = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4 + \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4} + \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot x_4 + \\ + \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot \overline{x_4} + \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot x_4 + \overline{x_1} \cdot x_2 \cdot x_3 \cdot \overline{x_4} + \overline{x_1} \cdot x_2 \cdot x_3 \cdot x_4 + \\ + x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} + x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4 + x_1 \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4} + \\ + x_1 \cdot \overline{x_2} \cdot x_3 \cdot x_4 + x_1 \cdot x_2 \cdot \overline{x_3} \cdot \overline{x_4} + x_1 \cdot x_2 \cdot \overline{x_3} \cdot x_4 + \\ + x_1 \cdot x_2 \cdot x_3 \cdot \overline{x_4} + x_1 \cdot x_2 \cdot x_3 \cdot x_4.$$

6.3.2 Мінімізувати функцію, представлену таблицею істинності методом карт Карно:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

6.3.3 Мінімізувати функцію, представлену таблицею істинності, двома способами (за допомогою карти Карно і методом Квайна)

та побудувати РКС:

№	x_1	x_2	x_3	x_4	f_1	f_2	f_3	f_4
0	0	0	0	0	1	0	0	1
1	0	0	0	1	1	1	0	1
2	0	0	1	0	0	0	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	0	0
5	0	1	0	1	1	0	1	1
6	0	1	1	0	0	1	1	0
7	0	1	1	1	1	0	1	1
8	1	0	0	0	0	0	0	0
9	1	0	0	1	0	1	1	1
10	1	0	1	0	1	0	1	1
11	1	0	1	1	1	0	1	1
12	1	1	0	0	0	1	1	0
13	1	1	0	1	0	1	1	0
14	1	1	1	0	1	1	1	1
15	1	1	1	1	1	0	1	0

7 СИНТЕЗ ПОСЛІДОВНОСНИХ АВТОМАТІВ

7.1 Основні поняття

Як зазначалося вище, *послідовнісні* автомати здійснюють перетворення двійкових вхідних змінних не тільки з урахуванням поточної комбінації на вході в даному такті автоматного часу, але і з урахуванням того, що було в попередніх тактах (стану пам'яті).

Цифровий пристрій називається *послідовносним*, якщо його вихідні сигнали залежать не тільки від поточних значень вхідних сигналів, але й від послідовності значень вхідних сигналів, що надійшли на вході в попередні моменти часу. Тому кажуть, що такі функціональні вузли "володіють пам'яттю".

Створення послідовносного автомата, здатного запам'ятовувати попередні даному такту комбінації сигналів на вході, забезпечується наявністю в комбінаційному автоматі не тільки зовнішніх (робочих) входів x і виходів y , але і допоміжних (внутрішніх) змінних z , які повинні мати можливість давати різні комбінації для кожного такту, що підлягає запам'ятовуванню, і реалізованих у вигляді зворотних зв'язків. Ці внутрішні змінні, що подаються на вхід комбінаційної схеми, як би коректують результат від впливу зовнішніх входів, з огляду на попередні ситуації.

Отже, завдання зовнішніх входів – задати поточну комбінацію, а внутрішніх – зберегти і поставити на вході комбінацію, однозначно відповідну тій, що сформувалася на виході в попередньому такті (що кодує її). В цьому випадку вихідна комбінація формується з урахуванням попереднього такту. У наступному такті внутрішні змінні внесуть чергову корективу (якщо їх комбінація буде відмінна від попередньої) і, отже, нові значення виходів вже несуть в собі сліди двох попередніх тактів і т. д. Кількість таких станів M внутрішніх змінних z , називається *вагою послідовносного автомата*.

Число внутрішніх змінних m_z , що забезпечує можливість кодування всіх M станів, називається *об'ємом пам'яті автомата* і дорівнює $m_z = \log_2 M$. Зовнішніми вхідними змінними автомата визначається кількість його можливих вхідних комбінацій, а внутрішніми – через скільки комбінацій зовнішніх починає

повторюватися корекція внутрішніми. Так, якщо число входних змінних $t_x = 3$, а внутрішніх $m_z = 2$, то автомат матиме вісімстанів з корекцією, що повторюється через чотири такти.

Наслідком цього є характерна особливість послідовносних автоматів: у них внаслідок корекції z можливі різні комбінації виходів при однакових комбінаціях на зовнішніх входах. Технічних варіантів реалізації навіть простих послідовносних автоматів може бути дуже багато. Досить наочною є схема послідовносного автомата, наведена на рис. 7.1 і представляє собою найбільш загальний випадок.



Рисунок 7.1 – Функціональна схема послідовносного автомата

Входи X^t логікою комбінаційної схеми формують виходи Y^t в даному такті. У формуванні Y^t при цьому беруть участь внутрішні змінні Z^{t-1} , комбінація яких сформувалася в попередньому такті $t-1$. Змінні Z^{t-1} сформувалися вузлом керування в колі зворотного зв'язку (33), який може включати: комбінаційну логіку, часові затримки і синхронізуючий вхід. Як видно зі схеми, саме вузол 33 забезпечує «запам'ятовування» шляхом прийому на вхід вихідної комбінації автомата в даному такті і передачу її з часовою затримкою на входи автомата в наступному такті.

Розрив ланцюга 33 перетворює послідовносні автомати в комбінаційні. Важливу роль відіграє характер затримки: якщо блок ЗС

має синхронізуючий вхід, який «відмикає» і «закриває» ланцюг зворотного зв'язку в кожному такті, – це синхронний автомат. У нього рівність $Z^{t-1} = Z^t$ настає тільки після «відмикання» синхроімпульсом ланцюга зворотного зв'язку (рис. 7.1).

Завдяки зворотному зв'язку, послідовнісні автомати мають специфічну властивість, що відрізняє їх від комбінаційних і характерну для замкнутих систем керування – вони можуть бути стійкими і нестійкими. *Стійким* будемо називати стан, коли комбінація виходів змінюється лише внаслідок зміни входу. *Нестійкі* автомати можуть після встановлення комбінації входів кілька разів міняти комбінацію на виході, проходячи ряд нестійких станів. Такий процес може завершитися переходом в стійкий стан (*загасання коливань*) або тривати необмежений час (*автоколивання*).

Комбінаційні автомати (розімкнуті) завжди стійкі, як і розімкнуті аналогові ланки. Формальною ознакою стійкого стану є рівність $Z^{t-1} = Z^t$, а нестійкого – нерівність $Z^{t-1} \neq Z^t$. Період автоколивань дорівнює подвійному часу затримки сигналу в ЗЗ.

Для послідовносних автоматів, як і для комбінаційних, застосовуються також алгебраїчні і табличні форми завдання оператора перетворення. На відміну від комбінаційних автоматів, де цей оператор задається у вигляді зв'язку робочих входів X^t і виходів Y^t , тобто для завдання оператора послідовносного автомата потрібно визначити зв'язок не тільки між робочими (зовнішніми) входами і виходами, а й між допоміжними (внутрішніми).

При алгебраїчній формі завдання записуються рівняння виходів, що зв'язують в загальному випадку робочі виходи Y^t з робочими X^t і допоміжними Z^{t-1} входами і рівняння переходів, що зв'язують виходи внутрішніх змінних Z^{t-1} з робочими X^t і допоміжними Z^t входами.

При табличній формі завдання складаються таблиці виходів і переходів, при об'єднанні яких виходить *таблиця станів*.

У реальних послідовносних автоматах комбінаційні і послідовнісні частини не розділяються, а з метою спрощення і скорочення числа елементів об'єднуються в одне ціле. Це підвищує використання елементів. Для побудови складних послідовносних автоматів, якими є більшість цифрових пристроїв, широко застосовуються елементарні послідовнісні автомати – тригери (Додаток Б).

7.2 Приклади розв'язання задач

Приклад 7.2.1 Розглянемо на конкретному прикладі синтез послідовного автомата. Насос H повинен автоматично вмикатися при досягненні робочою рідиною нижнього рівня, контрольованого поплавковим датчиком d_n , і вимикатися при досягненні верхнього рівня, контрольованого датчиком d_e .

Розв'язок

У циклі роботи насоса виділяємо чотири такти:

1 – рівень нижче d_n (насос працює, резервуар наповнюється);

2 – рівень вище d_n , але нижче d_e (насос працює, резервуар наповнюється);

3 – рівень досяг d_e (насос відключився, рідина витрачається, рівень знижується);

4 – рівень нижче d_e , але вище d_n (насос відключений, рівень знижується).

Зазначена послідовність відображена в Табл. 7.1.

Таблиця 7.1 – Послідовність роботи насоса

Елементи системи	Такти попереднього циклу	I		II		Такти наступного циклу
		1	2	3	4	
d_n		0	1	1	1	
d_e		0	0	1	0	
H		1	1	0	0	

Передбачається, що відбувається безперервне повторення однакових циклів (наповнення – витрата).

Для синтезу розглянуто один з циклів.

Вихідна змінна H приймає різні значення при однакових значеннях вхідних змінних (такти 2 і 4), але при різних їх значеннях в попередніх тактах 1 і 3. Це є характерним для послідовного автомата, тобто для відпрацювання необхідної послідовності необхідно «запам'ятати» попередній такт.

Введемо внутрішній зворотний зв'язок

$$y = c(x_i) \left(c(x_j) + x_e \right), \quad (7.1)$$

де $c(x_i)$ – кон'юнкція вхідних змінних, що не змінюються в сусідніх тактах;

$c(x_j)$ – кон'юнкція змінних, що змінюються в сусідніх тактах;

x_e – внутрішня змінна.

Відповідно до (7.1) запишемо:

для тактів 1,2

$$H_I = \overline{d_e} \left(\overline{d_n} + H'_e \right);$$

для тактів 3,4

$$\overline{H_{II}} = d_n \left(d_e + \overline{H''_e} \right),$$

де $\overline{H_e}$ – внутрішня змінна у вигляді зворотного зв'язку по виходу.

Перетворюючи останній вираз, отримуємо

$$H_{II} = \overline{d_n} + \overline{d_e} H''_e.$$

Контактні схеми дискретних автоматів представлені на рис. 7.2.

Схеми працездатні в нормальних умовах, проте при аварійному (залипання нижнього датчика) ($d_n = 0, d_e = 1$) – схема, зображена на рис. 7.2, а не допустить включення, а схема рис. 7.2, б допустить включення насоса, що викличе перелив.

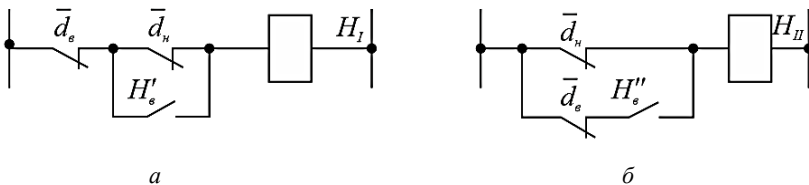


Рисунок 7.2 – Принципові схеми дискретних автоматів

Приклад 7.2.2 Розглянемо послідовнісну схему, полюсам якої приписані відповідні функції (рис. 7.3). За заданою схемою визначити її оператор (кінцевий автомат).

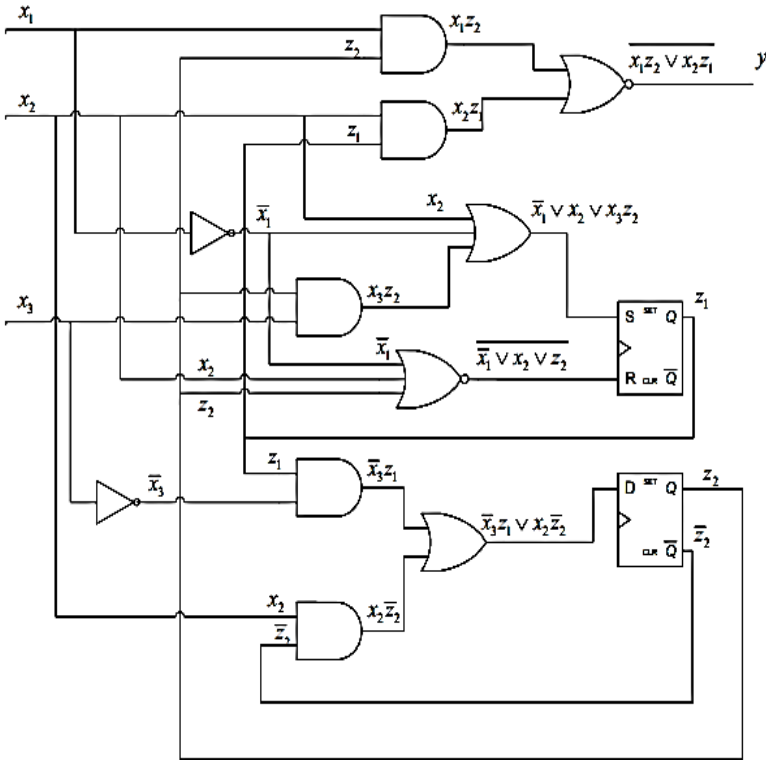


Рисунок 7.3 – Послідовнісна схема до прикладу 7.2.2

Розв'язок

Завдача має єдине рішення:

$$\begin{cases} y = \overline{x_1 z_2 \vee x_2 z_1}; \\ S = \bar{x}_1 \vee x_2 \vee x_3 z_2; \\ R = \bar{x}_1 \vee x_2 \vee \bar{z}_2; \\ D = \bar{x}_3 z_1 \vee x_2 \bar{z}_2. \end{cases}$$

Для виходу D -тригера маємо $z_2' = D$ (Додаток Б). Отримаємо функцію виходу RS -тригера (Додаток Б). Розглянемо для цього його таблицю істинності:

R	S	q	q'
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	—
1	1	1	—

Отримаємо найкоротшу ДНФ для даної функції:

$$q' = \overline{R}q \vee S.$$

Звідси

$$\begin{aligned} z' &= \overline{R}z_1 \vee S = \overline{\overline{x_1 \vee x_2 \vee z_2} \vee \overline{z_1 \vee \overline{x_1} \vee x_2 \vee x_3 z_2}} = \\ &= \overline{x_1} \vee x_2 \vee z_2 \vee \overline{z_1 \vee \overline{x_1} \vee x_2 \vee x_3 z_2} = z_2 z_1 \vee \overline{x_1} \vee x_2 \vee x_3 z_2. \end{aligned}$$

Остаточно маємо:

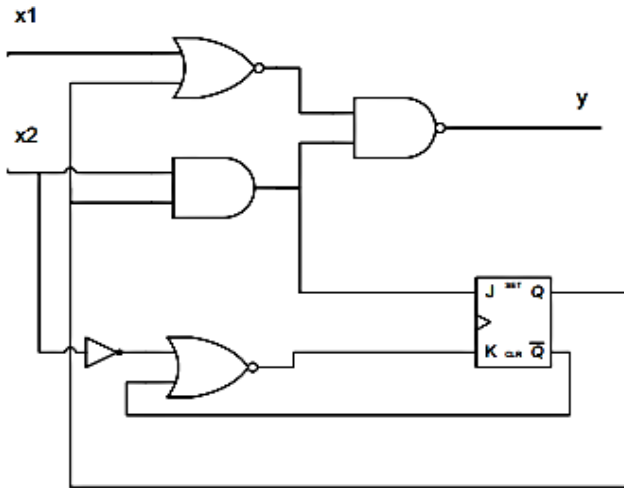
$$\begin{cases} y = \overline{x_1 z_2} \vee x_2 z_1; \\ z'_1 = z_2 z_1 \vee \overline{x_1} \vee x_2 \vee x_3 z_2; \\ z'_2 = \overline{x_3 z_1} \vee x_2 \overline{z_2}. \end{cases}$$

Це і є оператор схеми.

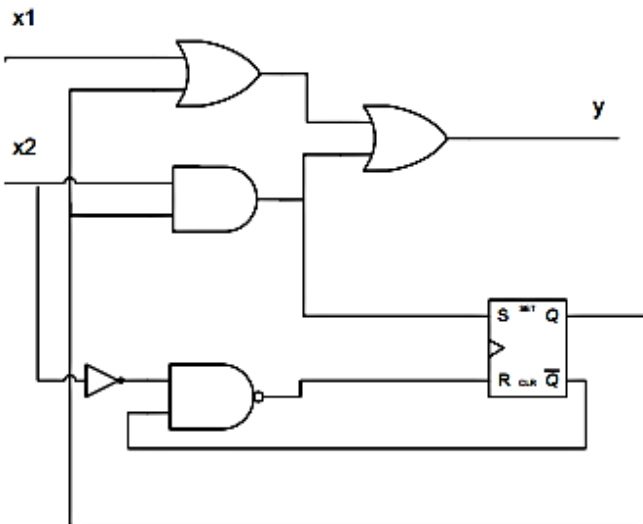
7.3 Задача для самостійного розв'язання

За заданою послідовнісною схемою визначити її оператор:

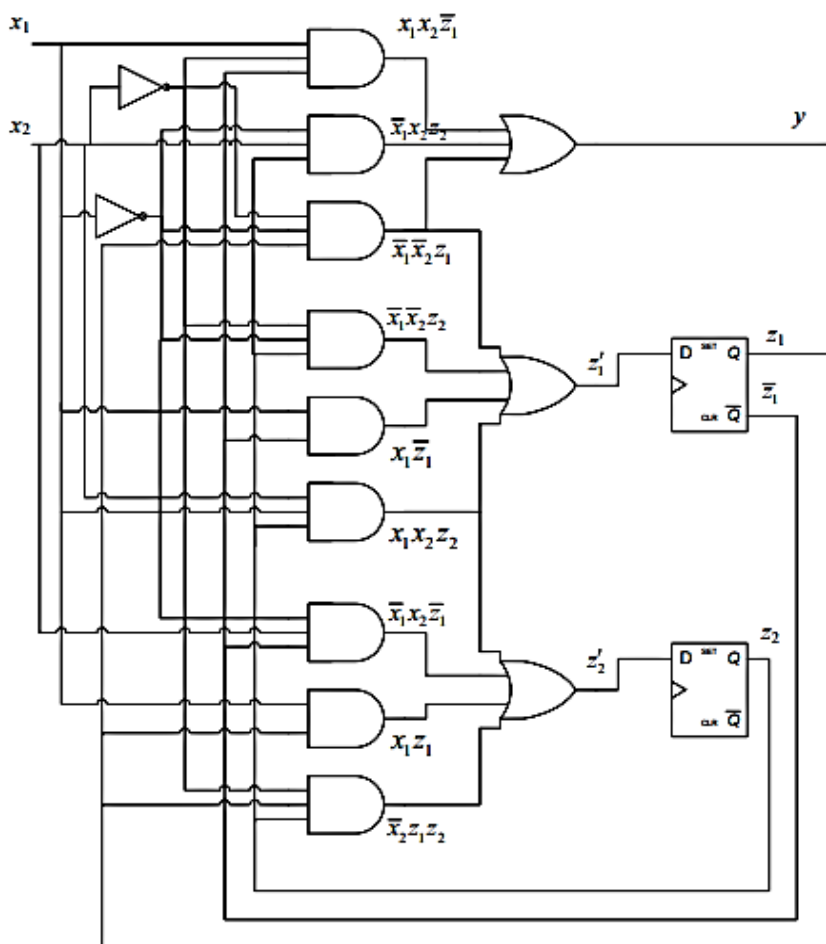
1)



2)



3)



8 ПИТАННЯ ДО МОДУЛЬНОГО КОНТРОЛЮ II

1. Назвати способи мінімізації логічних функцій.
2. В чому полягає синтез дискретних автоматів?
3. В чому полягає метод Квайна, його основні етапи?
4. Що представляє собою карта Карно?
5. Назвіть способи маркування сторін карт Карно.
6. Синтез послідовнісних автоматів.
7. Що називають вагою послідовного автомата?
8. Призначення зворотного зв'язку в послідовнісних автоматах.
9. В чому різниця між комбінаційними та послідовнісними автоматами?
10. Стійкий та нестійкий стани автомату.
11. Що таке таблиця станів?
12. Застосування програмованих реле для реалізації дискретних автоматизованих пристроїв.
13. Назвіть головні риси програмованих реле.
14. Якими засобами, крім описаних, можна реалізовувати логічні схеми?
15. Що таке тригери?

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

- 1 Математическая логика. Курс лекций. Задачник-практикум и решения [Текст]: учебное пособие / Л.М. Лихтарников, Т.Г. Сукачева. – 4-е изд. – СПб.: Изд-во «Лань», 2009. – 288 с.
- 2 Практикум по основам алгебры логики Часть I. Логические операции над высказываниями, формулы и функции алгебры логики [Текст]: учебное электронное текстовое издание / Д.В. Опарин. – 26 с.
- 3 Практикум по основам алгебры логики Часть II. Совершенные нормальные формы, приложения алгебры логики к релейно-контактным схемам, решения логических задач [Текст]: Учебное электронное текстовое издание / Опарин Д.В. – 33 с.
- 4 Практикум по дискретной математике [Текст]: учебное пособие. Часть I / В.П. Яковлев, Н.Л. Леонова. – ВШТЭ СПбГУПТД. – СПб., 2019. – 50 с.
- 5 Синтез комбинационных схем [Текст]: учебное пособие к курсовой работе по дисциплине "Дискретная математика" / П.С. Довгий, В.И. Поляков. – СПб: СПбГУ ИТМО, 2009. – 64 с.
- 6 Функции алгебры логики: задание и минимизация [Текст]: метод. указания и задание к контрольной работе по дисциплине «Теория дискретных устройств» / В.А. Алексеенко, М.В. Копанев. – Иркутск: ИрГУПС, 2015. – 36 с.
- 7 Дискретная математика. Теория и практикум [Текст]: учебник / Я.М. Ерусалимский. – СПб.: Изд-во «Лань», 2018. – 476 с.
- 8 Формулы и законы логики [Электронный ресурс] режим доступа до ресурсу: http://www.mathprofi.ru/formuly_i_zakony_logiki.html.
- 9 Логические схемы и таблицы истинности [Электронный ресурс] режим доступа до ресурсу: https://function-x.ru/logicheskie_shemy_i_tablici_istinnosti.html.
- 10 Схемная реализация комбинационных схем на логических элементах [Электронный ресурс] режим доступа до ресурсу: https://studopedia.su/12_44227_shemnaya-realizatsiya-kombinatsionnih-shem-na-logicheskikh-elementah.html.
- 11 СКНФ и СДНФ [Электронный ресурс] режим доступа до ресурсу: http://www.reshim.su/blog/sknf_i_sdnf/2013-08-27-407.
- 12 Булевы функции и формулы [Электронный ресурс] режим доступа до ресурсу: https://www.matburo.ru/ex_dm.php?p1=dmbul.

- 13 Глава 3 Метод минимизации логических функций [Электронный ресурс] режим доступа до ресурсу: https://web.posibnyky.vntu.edu.ua/fitki/6kondratenko_komp_praktikum_matlog/3.htm.
- 14 Булевы Функции [Электронный ресурс] режим доступа до ресурсу: http://ptca.narod.ru/lec/lec4_1.html.

Додаток А

Позначення логічних елементів за найбільш розповсюдженими стандартами

Американський (ANSI), європейський (DIN), міжнародний (IEC) і російський (ГОСТ)

Логика	«НЕ»	«И»	«ИЛИ»	«И-НЕ»	«ИЛИ-НЕ»
ГОСТ и IEC	Полож. 				
	Отриц. 				
ANSI	Полож. 				
	Отриц. 				
DIN	Полож. 				
	Отриц. 				

Додаток Б

Тригери

Це дискретні пристрої з булевими входними та вихідними змінними, які мають не більше двох входів, один вихід і рівно два внутрішніх стани 0 і 1.

Розглянемо основні види тригерів.

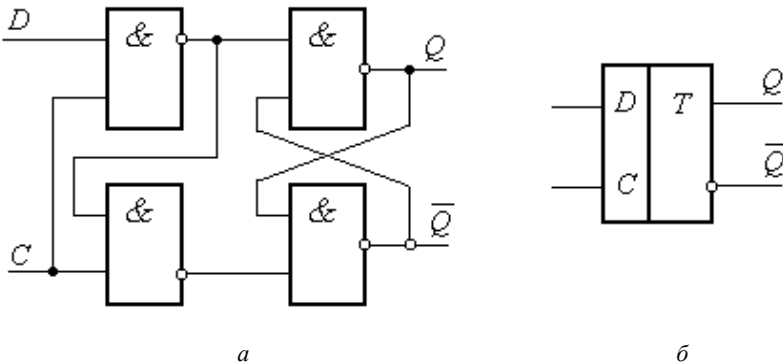
D -тригер (delay - затримати) затримує вхід на один такт. Він має один вхід і описується рівняннями

$$\begin{cases} y(t) = q(t); \\ q(t+1) = x(t), \end{cases}$$

або співвідношенням $y(t) = x(t-1)$. Очевидно, що D-тригер є автомат Мура з таблицею переходів:

D / q		0	1
0		0	1
1		1	0

D-тригер – базовий елемент для зберігання бітових даних. Цей тригер має один інформаційний вхід D і тактовий вхід C . Сигнал з входу D буде записаний в тригер і з'явиться на виході Q тільки при наявності активного рівня сигналу на тактовому вході C (рис. Б.1).



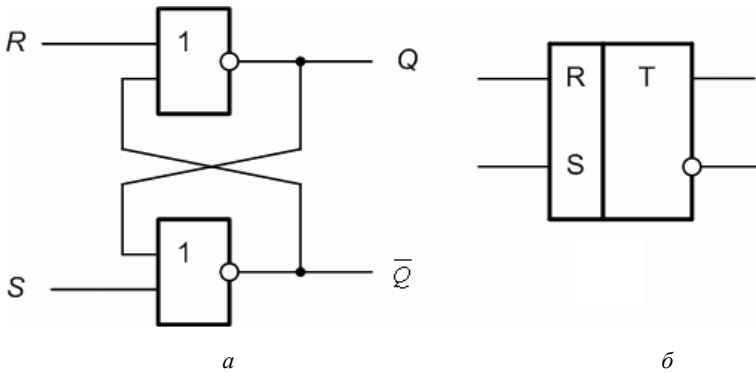
а) принципова схема; б) умовне графічне позначення

Рисунок Б.1 – Схема побудови D-тригера

Основна перевага D-тригера в порівнянні з RS-тригером в тому,

що він більш зручний для запису і зберігання цифрових даних за рахунок єдиного інформаційного входу, через який може бути встановлений в "1" або "0".

RS-тригер (reset - скинути, set - встановити) має два входи, під час вступу на входи нулів стан не змінюється, при надходженні на вхід R одиниці стан скидається в нуль, при надходженні на вхід S одиниці стан встановлюється в одиницю (рис. Б.2). При надходженні одиниць на обидва входи стан не визначено, тому цю комбінацію подавати на входи не рекомендується. Вихід збігається з поточним станом.



а) принципова схема; б) умовне графічне позначення

Рисунок Б.2 – Схема побудови RS-тригера

RS -тригер розглядається як частковий автомат Мура з таблицею переходів:

RS / q	0	1
00	0	1
01	1	1

JK-тригером називається цифровий автомат з двома інформаційними входами J та K , що володіє розширеними функціональними можливостями: може працювати як RS, D і T -тригери (рис. Б.3).

Інформаційні входи названі за першими літерами англійських слів:

K (*Kill*) - швидке вимкнення;

J (*Jark*) - швидке включення.

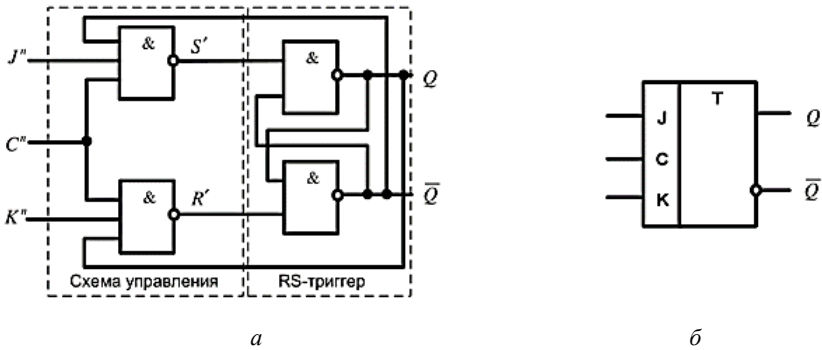


Рисунок Б.3– Схема побудови JK -тригера

Особливістю JK -тригера є відсутність заборонених комбінацій. При $J=K=1$ JK -тригер змінює свій стан на протилежний, тобто спрацьовує як T -тригер. В інших же випадках по входах J і K поводитья як RS -тригер (див. таблицю переходів).

номер набора	J^n	K^n	Q^n	Q^{n+1}	
0	0	0	0	0	} = Q^n Хранение информации
1	0	0	1	1	
2	0	1	0	0	} Установка 0
3	0	1	1	0	
4	1	0	0	1	} Установка 1
5	1	0	1	1	
6	1	1	0	1	} = $\overline{Q^n}$ Режим Т-триггера
7	1	1	1	0	

У інтегральній широко застосовуються синхронні JK -тригери. У одноступенчатому варіанті вони реалізуються на основі синхронного RS -тригера, а за рахунок перехресного ЗЗ (як у T -тригера на базі RS -тригера) реалізується рахунковий режим роботи.