МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний автомобільно-дорожній університет

МЕТОДИЧНІ ВКАЗІВКИ І ЗАВДАННЯ

до курсової роботи «Синтез логічних пристроїв»
з дисципліни «Електроніка та мікросхемотехніка»

Харків 2017

1. ПОРЯДОК ВИКОНАННЯ І ОФОРМЛЕННЯ КУРСОВОЇ РОБОТИ

Курсова робота присвячена синтезу генератора електричних сигналів заданої форми. В процесі виконання роботи студенти розробляють функціональні і принципові електричні схеми комбінаційних логічних пристроїв, які є складовими частинами генератора. Розроблені логічні пристрої забезпечують реалізацію сигналу заданої форми. При розробці принципових електричних схем передбачається використання сучасної елементної бази.

Виконання курсової роботи пов'язано з використанням теоретичного матеріалу за темами: «Логічні функції», «Дослідження логічних елементів», «Синтез комбінаційних логічних схем», «Мінімізація логічних пристроїв».

Курсова робота виконується відповідно до завдання. В завданні до курсової роботи передбачено 50 варіантів, кожному з яких відповідає своя форма вихідного сигналу генератора.

Пояснювальна записка виконується в обсязі 10–15 сторінок і включає розділи:

1. Теоретичні відомості, необхідні для виконання розрахунків;
2. Завдання, відповідне варіанту;
3. Розрахунки з поясненнями кожного етапу роботи;
4. Функціональна схема логічного пристрою в загальному базисі: «И», «ИЛИ», «НЕ»;
5. Схеми електричні принципові логічного пристрою в базисах «И-НЕ», «ИЛИ-НЕ»;
6. Переліки елементів для розроблених схем;
7. Заключення із зазначенням базису, в якому доцільно виконувати логічний пристрій;
8. Перелік використаної літератури.

2. ТЕОРЕТИЧНІ ВІДОМОСТІ

2.1. Логічні функції

В сучасних пристроях обробки інформації використовуються два класи змінних: числа і логічні змінні.

Числа несуть інформацію про кількісні характеристики системи. Над ними можна проводити арифметичні дії.

Логічні змінні визначають стан системи або приналежність її до певного класу станів. Логічна змінна може приймати одне з двох значень: істинно або хибно. Для стислості ці значення зазвичай позначають символами 1 або 0, не вкладаючи в них сенсу кількості.

Багато задач управління зводяться до аналізу логічних умов і видачі логічних команд. Для того щоб вирішувати такі завдання, необхідний спеціальний математичний апарат. Такий апарат розроблено в середині IXX століття ірландським математиком Джорджом Булем. За його імені математичний апарат і отримав назву булевої алгебри або алгебри логіки.

Логічні залежності записують у вигляді:

$$y = f(x_1, x_2, x_3, ..., x_n).$$

де, x_1-x_n – логічні змінні – аргументи функції;

y – логічна змінна – функція аргументів x_i;

y і x_1-x_n можуть приймати тільки два значення 0 або 1.

Задати функцію алгебри логіки означає визначити значення y для всіх можливих комбінацій $x_1...x_n$. Очевидно, що всього таких комбінацій 2_n. Функція називається повністю визначеною, якщо задані всі 2_n її значень. Якщо частина значень функції не задана, то вона називається частково визначеною або недовизначеною.

Пристрої, поведінки яких описуються за допомогою функцій алгебри логіки, називають логічними.

Для завдання функції алгебри логіки можуть бути використані: словесні описи, таблиці істинності, алгебраїчні вирази.

1) Словесний опис функції алгебри логіки.

Приклад: Логічна функція трьох змінних дорівнює одиниці, якщо хоча б дві її вхідні змінні дорівнюють одиниці.

2) Таблиця істинності.

Таблицею істинності називається таблиця, яка містить всі можливі комбінації значень вхідних змінних $x_1...x_n$ і відповідні їм значення вихідних змінних y_i.

Приклад: таблиця істинності для функції з попереднього прикладу:

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

3) Алгебраїчні вирази.

Основними операціями булевої алгебри є операції логічного додавання, множення, заперечення.

Логічне додавання (операція «ИЛИ», диз'юнкція). Постулати логічного додавання для функції двох змінних ілюструє таблиця істинності:

x_2	x_1	$x_1 \vee x_2$ $x_1 + x_2$
0	0	0
0	1	1
1	0	1
1	1	1

Дана операція справедлива для довільного числа змінних. Операція «ИЛИ» відповідає математичній операції об'єднання множин.

Логічне множення (операція «И», кон'юнкція).

Постулати логічного множення двох змінних ілюструє таблиця істинності:

x_2	x_1	$x_1 \wedge x_2$ $x_1 x_2$
0	0	0
0	1	0
1	0	0
1	1	1

Дана операція також справедлива для довільного числа змінних. Операція «И» відповідає математичній операції перетину множин.

Логічне заперечення (інверсія). Для позначення логічного заперечення використовують риску над відповідним виразом. Операція визначається наступними постулатами:

якщо $x = 1$, то $\bar{x} = 0$

якщо $x = 0$, то $\bar{x} = 1$.

Операції логічного додавання, множення і заперечення утворюють функціонально повну систему логічних функцій. Це означає, що яку завгодно складну логічну функцію можна виразити через ці три операції.

При описі функції алгебри логіки алгебраїчним виразом використовують дві стандартні форми її подання:

1) ***диз'юнктивною нормальною формою (ДНФ)*** називається логічна сума елементарних логічних добутків, в кожному з яких аргумент або його інверсія входить один раз.

Отримана ДНФ може бути з таблиці істинності з використанням наступного алгоритму:

а) для кожного набору змінних, на якому функція алгебри логіки (ФАЛ) дорівнює одиниці, записують елементарні логічні добутки вхідних змінних. Причому змінні, рівні нулю, записують з інверсією. Отримані добутки називають конституєнтами одиниці.

б) логічно підсумовують всі конституєнти одиниці. ДНФ, отриману внаслідок підсумовування конституєнтів одиниці, називають досконалою (ДДНФ).

Приклад: записати ДДНФ для ФАЛ, заданої в попередньому прикладі:

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

2) ***кон'юнктивною нормальною формою (КНФ)*** називається логічний добуток елементарних логічних сум, в кожному з яких аргумент або його інверсія входять один раз.

КНФ може бути знайдена з таблиці істинності з використанням наступного алгоритму:

а) для кожного набору змінних, на якому ФАЛ дорівнює нулю, записують елементарні логічні суми вхідних змінних. Причому змінні, значення яких дорівнюють 1, записують з інверсією. Отримані суми називають конституєнтамі нуля.

б) логічно перемножують всі отримані конституєнти нуля. КНФ, отриману перемноженням конституєнтів нуля, називають досконалою (ДКНФ).

Приклад: записати ДКНФ для ФАЛ, заданої в попередньому прикладі:

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

2.2. Дослідження логічних елементів схеми

Використовуючи ФАЛ, можна однозначно визначити внутрішню структуру логічного пристрою. Для апаратної реалізації такого пристрою необхідний набір елементарних вузлів, які реалізують основні логічні операції. Ці вузли називають логічними елементами (ЛЕ). За допомогою таких елементарних вузлів можна побудувати логічну схему, яка буде виконувати необхідні перетворення вхідних змінних у вихідні.

Умовні графічні позначення логічних елементів, що реалізують три основні операції булевої алгебри, наведені на рис.2.1

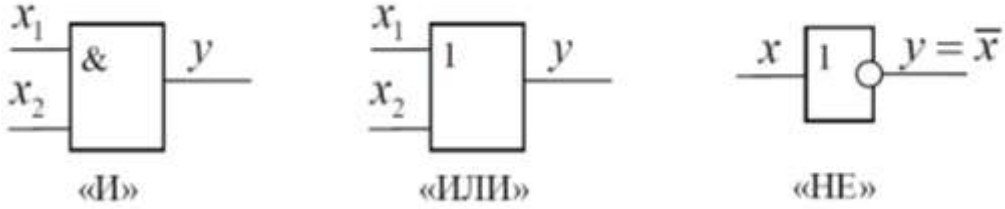


Рисунок 2.1 – Умовні графічні позначення логічних елементів

Як було сказано раніше, число входів елементів "И" і "ИЛИ" може бути довільним. Часто число входів позначають цифрою, що стоїть перед назвою елемента ("2И", "3ИЛИ"). Елемент "НЕ" має завжди тільки один вхід. Для побудови логічної схеми необхідно логічні елементи, призначені для виконання логічних операцій, зазначених в ФАЛ, розташовувати, починаючи від входу в порядку, визначеному булевим виразом.

Приклад: побудувати структурну схему логічного пристрою за ФАЛ з попереднього прикладу (див. рис. 2.2):

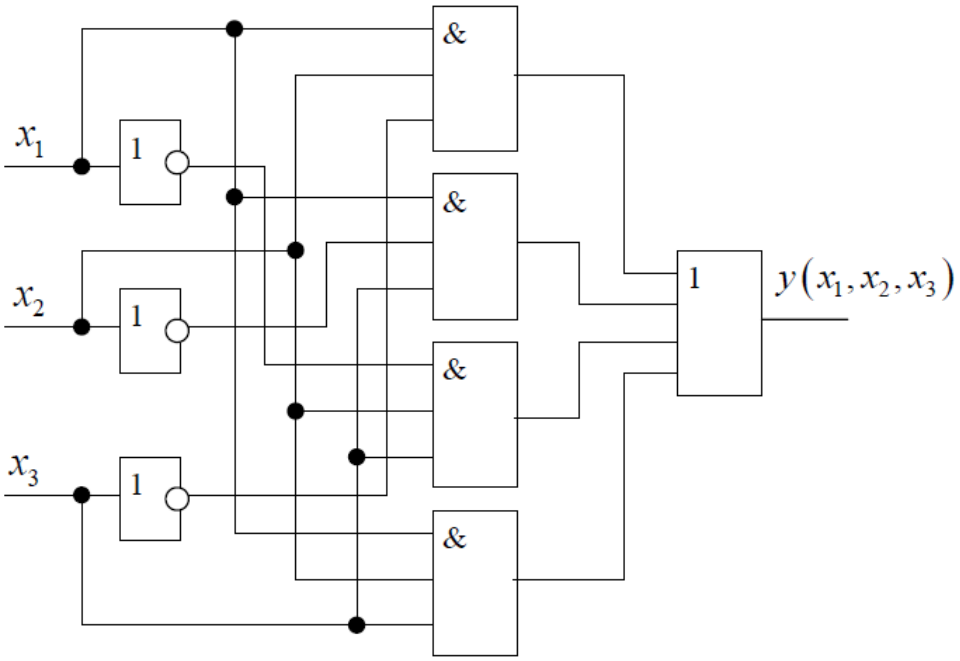


Рисунок 2.2 – Структурна схема логічного пристрою

2.3. Принцип подвійності. Функціональна повнота системи ЛЕ

Порівнюючи таблиці істинності для операцій "И" і "ИЛИ" можна помітити, що одну таблицю можна отримати з іншої, замінивши всі змінні їх інверсіями, а знак логічного множення – знаком логічного складання. Інакше кажучи:

якщо

$$x_1 \cdot x_2 = y, \quad \text{то } \bar{x}_1 + \bar{x}_2 = \bar{y};$$

якщо

$$x_1 + x_2 = y, \text{ то } \bar{x}_1 \cdot \bar{x}_2 = \bar{y};$$

Це властивість взаємного перетворення постулатів логічного додавання і множення носить назву ***принципу подвійності***. Важливим практичним наслідком принципу подвійності є той факт, що під час запису логічних виразів і, отже, побудові логічних схем, можна обійтися тільки двома типами операцій: операціями "И" та "НЕ" або операціями "ИЛИ" і "НЕ". Таким чином, кожна з цих систем операцій ("И" та "НЕ", а також "ИЛИ" і "НЕ") є функціонально повною системою нарівні з системою з трьох елементів "И", "ИЛИ", "НЕ".

На практиці широке застосування знаходять дві функціонально-повні системи елементів які містять лише по одному елементу: "И-НЕ" і "ИЛИ-НЕ". Вони носять назву "штрих Шеффера" і "стрілка Пірса" відповідно.

Таблиці істинності логічних елементів "2И-НЕ" і "2ИЛИ-НЕ" мають такий вигляд:

x_2	x_1	"И-НЕ" $\overline{x_1 \cdot x_2}$ $x_1 \mid x_2$	"ИЛИ-НЕ" $\overline{x_1 + x_2}$ $x_1 \downarrow x_2$
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	0

На рис.2.3 умовні графічні позначення логічних елементів "2И-НЕ" і "2ИЛИ-НЕ".

Рисунок 2.3 – Умовні графічні позначення логічних елементів "2И-НЕ" і "2ИЛИ-НЕ"

Для прикладу розглянемо виконання операцій "И", "ИЛИ" і "НЕ" на елементах "ИЛИ-НЕ" і "И-НЕ" (див. рис.2.4.).

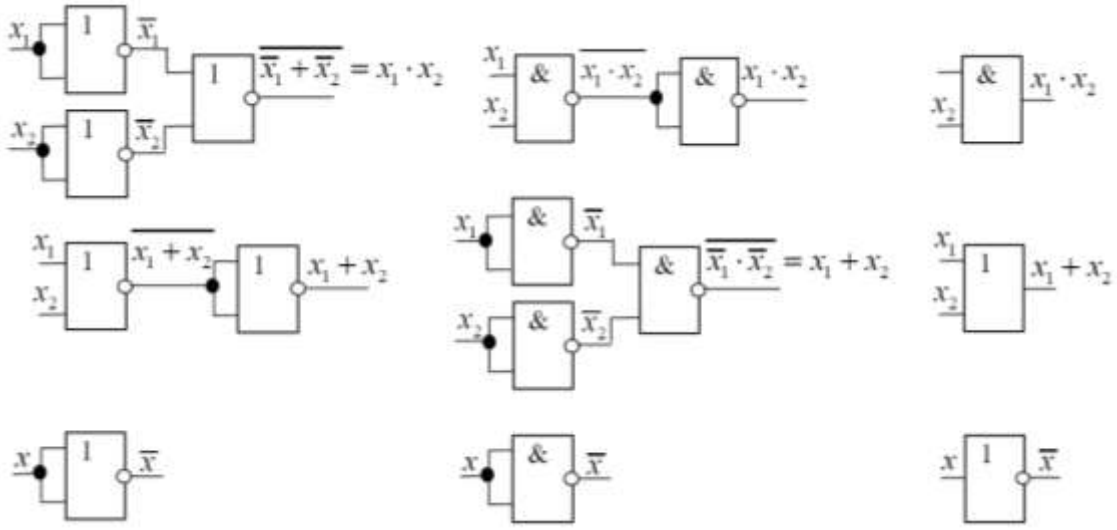


Рисунок 2.4 – Виконання операцій "И", "ИЛИ" і "НЕ" на елементах "ИЛИ-НЕ" і "И-НЕ"

2.4. Булева алгебра

Теореми булевої алгебри відображають зв'язки, що існують між операціями, що здійснюються над логічними змінними.

1.	$x+0=x$	$x\cdot 1=x$	
2.	$x+1=1$	$x+0=0$	
3.	$x+x=x$	$x\cdot x=x$	
4.	$\overline{x}+x=1$	$\overline{x}\cdot x=0$	
5.	$\overline{\overline{x}}=x$		
6.	$x_1+x_2=x_2+x_1$	$x_1\cdot x_2=x_2\cdot x_1$	
7.	$(x_1+x_2)+x_3=x_1+(x_2+x_3)$	$(x_1\cdot x_2)\cdot x_3=x_1\cdot(x_2\cdot x_3)$	теорема Де-Моргана
8.	$\overline{x_1+x_2}=\overline{x}_2\cdot\overline{x}_1$	$\overline{x_1\cdot x_2}=\overline{x}_2+\overline{x}_1$	теорема поглинання
9.	$x_1\cdot x_2+x_2=x_2$	$(x_1+x_2)\cdot x_2=x_2$	
10.	$x_1\cdot x_2+x_3=(x_1+x_3)\cdot(x_2+x_3)$	$(x_1+x_2)\cdot x_3=x_1\cdot x_3+x_2\cdot x_3$	
11.	$x_1\cdot\overline{x}_2+x_2=x_1+x_2$	$(x_1+\overline{x}_2)\cdot x_2=x_1\cdot x_2$	теорема склеювання
12.	$x_1\cdot x_2+\overline{x}_1\cdot x_2=x_2$	$(x_1+x_2)\cdot(\overline{x}_1+x_2)=x_2$	

2.5. Мінімізація логічних пристроїв

Раніше було сказано, що логічну схему, яка реалізує заданий алгоритм перетворення сигналів, можна синтезувати безпосередньо за виразом, представленим у вигляді ДДНФ або ДКНФ. Однак отримана при цьому схема, як правило, не оптимальна з точки зору її практичної реалізації. Тому вихідні ФАЛ зазвичай мінімізують. Метою мінімізації логічної функції зазвичай є зменшення вартості її технічної реалізації.

Критеріями мінімізації можуть бути:

– зменшення числа логічних елементів;

– збільшення регулярності внутрішньої структури логічних пристроїв;

– зменшення числа зовнішніх з'єднань в інтегральних схемах.

Все більш широке використання при проектуванні логічних пристроїв програмованих логічних НВІС на основі базових матричних кристалів визначає важливість мінімізації за критерієм зменшення числа логічних елементів.

Ці НВІС містять, як правило, окремі елементарні ЛЕ, що нез'єднані між собою, наприклад "2И-НЕ" або "2ИЛИ-НЕ". Оскільки число елементів в одній НВІС обмежено і задано з технологічних міркувань, то мінімізація ФАЛ дозволяє на одному кристалі вирішити більш складні завдання логічної обробки сигналів.

Для мінімізації ФАЛ можна застосовувати основні закони булевої алгебри.

Приклад:

$$y(x_1,x_2,x_3)=\overline{x}_1x_2x_3+x_1\overline{x}_2x_3+x_1x_2\overline{x}_3+x_1x_2x_3=$$

$$=(\overline{x}_1x_2x_3+x_1x_2x_3)+(x_1\overline{x}_2x_3+x_1x_2x_3)+(x_1x_2\overline{x}_3+x_1x_2x_3)= \quad \textit{// теореми 3, 6, 7}$$

$$=x_2x_3+x_1x_3+x_1x_2 \quad \textit{// теорема 12}$$

Застосування для мінімізації ФАЛ алгебраїчних перетворень – це універсальний шлях. Однак кінцевий результат при цьому залежить від кваліфікації фахівця, який проводить мінімізацію. Для функцій, які містять не більше 5 – 6 аргументів, зручно проводити мінімізацію за допомогою діаграм Вейча або карт Карно.

2.6. Мінімізація ФАЛ з використанням діаграм Вейча

Діаграма Вейча – це своєрідний вид записи таблиць істинності. Вона являє собою прямокутну таблицю, число клітин в якій для ФАЛ n змінних дорівнює 2^n. Кожній з клітин поставлено у відповідність деякий набір вхідних змінних, причому поруч розташованим клітинам відповідають сусідні набори вхідних змінних (які розрізняються значенням тільки однієї змінної).

У самих клітинах записують значення ФАЛ, відповідні для даного набору змінних.

Карта Вейча функції двох змінних:

	x_2	$\overline{x}_2$
x_1	$f(x_1, x_2)$	$f(x_1, \overline{x}_2)$
$\overline{x}_1$	$f(\overline{x}_1, x_2)$	$f(\overline{x}_1, \overline{x}_2)$

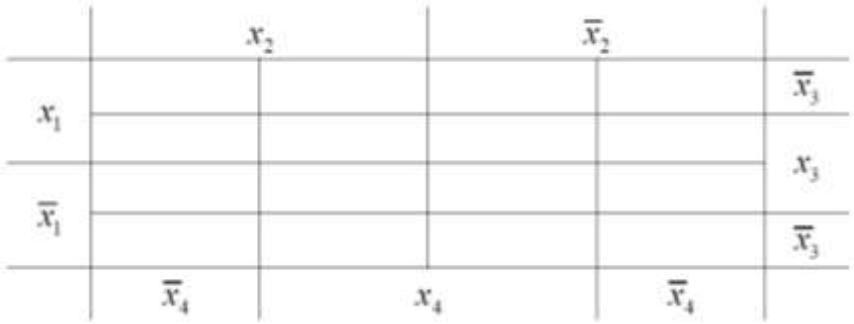
По краях карти вказано значення вхідних змінних для відповідних рядків і стовпців. Для двох змінних карта Вейча є плоскою фігурою.

Карта Вейча функції трьох змінних:

	x_2		$\overline{x}_2$	
x_1	$f(x_1, x_2, \overline{x}_3)$	$f(x_1, x_2, x_3)$	$f(x_1, \overline{x}_2, x_3)$	$f(x_1, \overline{x}_2, \overline{x}_3)$
$\overline{x}_1$	$f(\overline{x}_1, x_2, \overline{x}_3)$	$f(\overline{x}_1, x_2, x_3)$	$f(\overline{x}_1, \overline{x}_2, x_3)$	$f(\overline{x}_1, \overline{x}_2, \overline{x}_3)$
	$\overline{x}_3$	x_3		$\overline{x}_3$

Набори вхідних змінних, відповідні крайньому лівому і правому стовпцям, є сусідніми. Тому в даному випадку карту Вейча зручно представляти як поверхню циліндра, і вона є об'ємною фігурою.

Карта Вейча функції чотирьох змінних:



Набори вхідних змінних, відповідні крайнім лівому і правому стовпцям, а також нижньому і верхньому рядку карти, є сусідніми, тобто дана карта може бути представлена як поверхня тора.

Карта Вейча для функції п'яти змінних може бути представлена як дві карти Вейча функції 4 змінних, розташованих одна над другою і відрізняються лише значенням однієї змінної. При цьому відповідні клітини двох таблиць будуть сусідніми.

При заповненні карти Вейча в відповідну клітинку ставиться 1, якщо функція, що мінімізується при цьому наборі аргументів дорівнює 1. В інші клітинки таблиці вписуються нулі.

2.7. Процес мінімізації

В заповненій таблиці обводять прямокутними контурами всі одиниці, а потім записують мінімізовану функцію у вигляді суми логічних добутків, що описують ці контури.

При проведенні контурів дотримуються наступних правил:

– контур повинен бути прямокутним;

– всередині контуру повинні бути тільки клітини, які заповнені одиницями;

– число клітин, що знаходяться всередині контуру, має бути цілим ступенем числа 2 (1, 2, 4, 8);

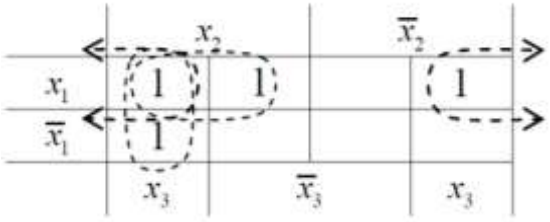
– одні й ті ж клітини заповнені одиницями, можуть входити в кілька контурів;

– число контурів має бути якомога меншим, самі контури повинні бути якомога більшими.

Для того щоб знайти логічний вираз (просту кон'юнкцію), який описує в діаграмі Вейча кожен контур, що охоплює одиниці, потрібно розглянути позначення рядків і стовпців, що входять в цей контур і виключити з цих позначень ті аргументи, які змінюють своє значення всередині контуру.

Приклад:

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



Тобто число доданків результуючого виразу дорівнює кількості контурів. Доданок, що описує кожен контур, містить тільки ті змінні, які в цьому контурі не змінюються.

При збільшенні числа аргументів функції понад чотирьох поняття про сусідні рядки і стовпці діаграми Вейча ускладнюється, і наочність процесу мінімізації втрачається.

2.8. Мінімізація недовизначених ФАЛ

Недовизначеною називається ФАЛ, значення якої задано не на всіх наборах вхідних змінних.

При мінімізації недовизначеної ФАЛ її необумовлені значення довизначають довільним чином з умови отримання на карті Вейча найменшого числа найбільш великих областей, охоплених контурами.

Виконання цієї вимоги призводить до найпростішої технічної реалізації ФАЛ.

2.9. Мінімізація логічних функцій за допомогою карт Карно

Карти Карно були винайдені в 1952 Едвардом В. Вейчем і вдосконалені в 1953 Морісом Карно, фізиком з «Bell Labs», і були покликані допомогти спростити цифрові електронні схеми.

Карта Карно – це спеціального виду таблиця, яка дозволяє спростити процес пошуку мінімальних форм і успішно застосовується, коли число змінних не перевищує шести. Карти Карно для функцій, що залежать від n змінних, являє собою прямокутник, розділений на 2^n клітин. Кожній клітці діаграми ставиться у відповідність двійковий n-мірний набір. Значення заданої функції f з таблиці істинності вносяться в потрібні квадрати, однак якщо клітці відповідає 0, то зазвичай вона залишається порожньою. Являє собою операції попарного неповного склеювання і елементарного поглинання.

У таблиці наведено розмітка карти Карно для $n = 4$ змінних.

X_3X_4 / X_1X_2	00	01	11	10
00	0000	0001	0011	0010
01	0100	0101	0011	0010
11	1100	1101	1111	1110
10	1000	1001	1011	1010

В карту Карно булеві змінні передаються з таблиці істинності і упорядковуються за допомогою коду Грея* в якому кожне наступне число відрізняється від попереднього тільки одним розрядом.

Основним методом мінімізації логічних функцій, представлених у вигляді ДДНФ або ДКНФ є операція попарного неповного склеювання і елементарного поглинання. Операція попарного склеювання здійснюється між двома термами (членами), що містять однакові змінні, входження яких (прямі і інверсні) збігаються для всіх змінних, крім однієї. В цьому випадку всі змінні, крім однієї, можна винести за дужки, а що залишилися в дужках пряме і інверсне входження однієї змінної піддати склейці.

Наприклад:

$$\overline{X}_1X_2X_3X_4 \vee \overline{X}_1X_2\overline{X}_3X_4 = \overline{X}_1X_2X_4(X_3 \vee \overline{X}_3) = \overline{X}_1X_2X_4.$$

Можливість поглинання випливає з очевидних рівностей:

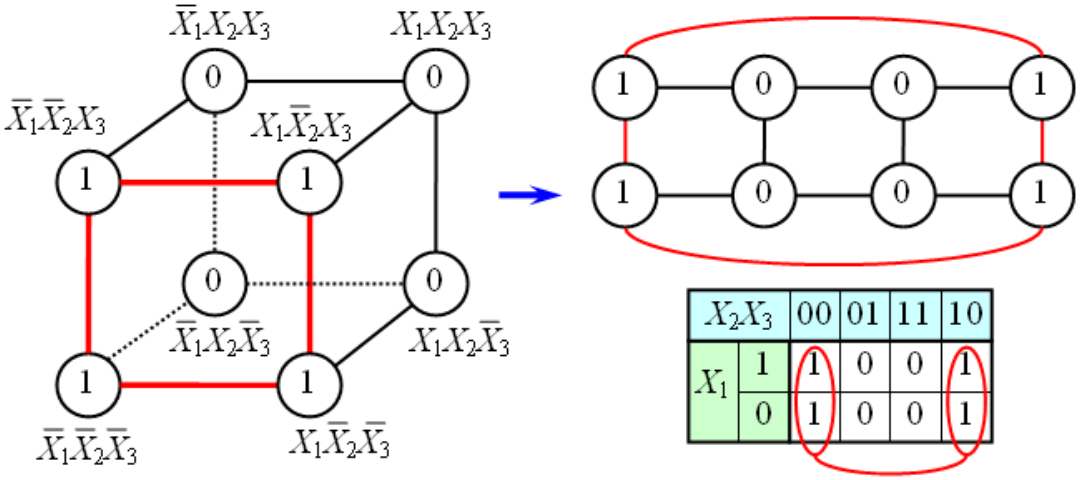
$$A \vee \overline{A} = 1; A\overline{A} = 0.$$

---

* Прим.: *(код Грея - двійковий код (0 і 1), в якому дві «сусідні» кодові комбінації відрізняються тільки цифрою в одному двійковому розряді.*

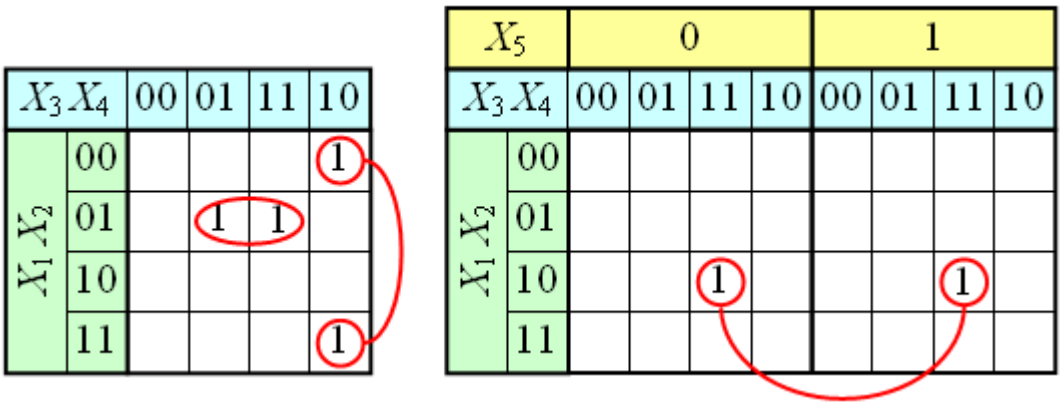
Таким чином, головним завданням при мінімізації ДДНФ і ДКНФ є пошук термів, придатних до склейки з подальшим поглинанням, що для великих форм може виявитися досить складним завданням. Карти Карно надають наочний спосіб відшукання таких термів.

Для спрощення роботи з булевими функціями великого числа змінних був запропонований наступний зручний прийом. Куб, що представляє собою структуру термів, розгортається на площину як показано на малюнку. Таким чином з'являється можливість представляти булеві функції з числом змінних більше двох у вигляді плоскої таблиці. При цьому слід пам'ятати, що порядок кодів термів в таблиці (00 01 11 10) не відповідає порядку проходження двійкових чисел, а клітини, що знаходяться в крайніх стовпчиках таблиці, сусідять між собою.



X_2X_3		00	01	11	10
X_1	1	1	0	0	1
	0	1	0	0	1

Аналогічним чином можна працювати з функціями чотирьох, п'яти і більше змінних. Приклад таблиці для N=4 наведено на рисунку. Для таких таблиць слід пам'ятати, що сусідніми є клітини, які знаходяться у відповідних клітинах крайніх стовпців і відповідних клітинах верхнього і нижнього рядка.



X_3X_4		00	01	11	10
X_1X_2	00				1
	01		1	1	
	10				
	11				1

X_5		0				1			
X_3X_4		00	01	11	10	00	01	11	10
X_1X_2	00								
	01								
	10			1				1	
	11								

Карта Карно може бути складена для будь-якої кількості змінних, однак зручно працювати при кількості змінних не більше п'яти. По суті Карта Карно – це таблиця істинності, складена в 2-х вимірному вигляді. Завдяки використанню коду Грея в ній верхній рядок є сусіднім з нижнім, а правий стовпець сусідній з лівим, тобто вся Карта Карно згортається в фігуру тор. На перетині рядка і стовпця проставляється відповідне

значення з таблиці істинності. Після того як Карта заповнена, можна приступати до мінімізації.

Якщо необхідно отримати мінімальну ДНФ, то в Карті розглядаємо тільки ті клітини які містять одиниці, якщо потрібна КНФ, то розглядаємо ті клітини які містять нулі. Сама мінімізація проводиться за такими правилами (на прикладі ДНФ):

1. Об'єднуємо суміжні клітини, які містять одиниці, в область, так щоб одна область містила 2^n (n ціле число = 0 ...) клітин (пам'ятаємо про те, що крайні рядки і стовпці є сусідніми між собою), в області не повинно знаходитися клітин, що містять нулі;
2. Область повинна розташовуватися симетрично осі (осі розташовуються через кожні чотири клітини);
3. Не суміжні області розташовані симетрично осі можуть об'єднуватися в одну;
4. Область повинна бути якомога більшою, а кількість областей якомога меншою;
5. Області можуть перетинатися;
6. Можливо кілька варіантів накриття.

Далі беремо першу область і дивимося які змінні не змінюються в межах цієї області, виписуємо кон'юнкцію цих змінних, якщо незмінна змінна нульова, проставляємо над нею інверсію. Беремо наступну область, виконуємо те ж саме що і для першої, і т.д. для всіх областей. Кон'юнкції областей об'єднуємо диз'юнкцією.

2.10. Синтез логічних пристроїв в заданому базисі ЛЕ

При побудові логічних пристроїв зазвичай не користуються функціонально повною системою ЛЕ, що реалізує всі три основні логічні операції: "И", "ИЛИ" і "НЕ".

На практиці, з метою скорочення номенклатури елементів, часто користуються функціонально повними системами елементів, що включають тільки по одному елементу, що виконує операцію "И-НЕ" або "ИЛИ-НЕ". Причому число входів цих елементів, як правило, задано.

У таблиці наведено можливі форми представлення вихідного сигналу елементів "2И-НЕ" і "2ИЛИ-НЕ".

Елемент	Умовне позначення операції	Форма представлення вихідного сигналу
«2И-НЕ» (штрих Шеффера)	$x_1 \mid x_2$	$\overline{x_1 \cdot x_2}$; $\overline{x}_1 + \overline{x}_2$
«2ИЛИ-НЕ» (стрілка Пірса)	$x_1 \downarrow x_2$	$\overline{x_1 + x_2}$; $\overline{x}_1 \cdot \overline{x}_2$

Для того щоб уявити будь-яку ФАЛ в необхідному базисі ЛЕ, використовують два технічних прийоми:

– подвійне інвертування вихідного виразу або його частини;

– застосування теорем Де-Моргана: $\overline{x_1 + x_2} = \overline{x}_2 \cdot \overline{x}_1$; $\overline{x_1 \cdot x_2} = \overline{x}_2 + \overline{x}_1$.

Зазначеними прийомами ФАЛ перетвориться до виду, який містить тільки операції логічного множення (складання) і інверсії. Далі вона переписується через умовні позначення операції "И-НЕ" ("ИЛИ-НЕ").

Приклад: Задана ФАЛ

$$y(\vec{x}) = x_0 x_3 + \left(\overline{\bar{x}_0 x_2 \bar{x}_3}\right) \cdot (x_1 + \bar{x}_2).$$

Перетворити її до базисів ЛЕ "И-НЕ" і "ИЛИ-НЕ".

$$y(\vec{x}) = \overline{\overline{x_0 x_3 + \left(\overline{\bar{x}_0 x_2 \bar{x}_3}\right) \cdot (x_1 + \bar{x}_2)}} = \overline{\overline{x_0 x_3} \cdot \left(\overline{\bar{x}_0 x_2 \bar{x}_3}\right) \cdot \left(\overline{\overline{x_1 + \bar{x}_2}}\right)} =$$

$$= \overline{\overline{x_0 x_3} \cdot \left(\overline{\bar{x}_0 x_2 \bar{x}_3}\right) \cdot \left(\overline{\bar{x}_1 \cdot x_2}\right)} = (x_0 \mid x_3) \mid \left((\bar{x}_0 \mid x_2 \mid \bar{x}_3) \mid (\bar{x}_1 \mid x_2)\right)$$

$$y(\vec{x}) = \overline{\overline{x_0 x_3}} + \overline{\left(\overline{\bar{x}_0 x_2 \bar{x}_3}\right) \cdot (x_1 + \bar{x}_2)} = \overline{\bar{x}_0 + \bar{x}_3} + \overline{\overline{\bar{x}_0 x_2 \bar{x}_3}} + \overline{x_1 + \bar{x}_2} =$$

$$= \overline{\bar{x}_0 + \bar{x}_3} + \overline{\overline{x_0 + \bar{x}_2 + x_3}} + \overline{x_1 + \bar{x}_2} = \overline{\bar{x}_0 \downarrow \bar{x}_3 \downarrow \left((x_0 \downarrow \bar{x}_2 \downarrow x_3) \downarrow (x_1 \downarrow \bar{x}_2)\right)}$$

У тому випадку, коли число входів ЛЕ більше числа змінних, що входять в реалізовану ФАЛ, зменшити фактичне число входів ЛЕ можна, подаючи на невикористовувані входи сигнал логічного "0" для елементів "ИЛИ-НЕ" і логічної "1" для елементів "И-НЕ".

Інший прийом зменшення фактичного числа входів ЛЕ базується на використанні тотожності: $x+x=x$; $x \cdot x = x$. Тому на кілька входів ЛЕ можна подавати одну й ту ж логічну змінну.

У тому випадку, якщо число входів ЛЕ менше необхідного, можна скористатися однією з наступних тотожностей:

$$x_2 \mid x_1 \mid x_0 = x_2 \mid \overline{x_1 \mid x_0}; \qquad x_2 \downarrow x_1 \downarrow x_0 = x_2 \downarrow \overline{x_1 \downarrow x_0}.$$

Доведемо перше з них:

$$x_2 \mid x_1 \mid x_0 = \overline{x_2 \cdot x_1 \cdot x_0} = \overline{\overline{x_2 \cdot x_1 \cdot x_0}} = x_2 \mid \overline{x_1 \mid x_0}.$$

Наведені тотожності справедливі для будь-якого числа вхідних змінних:

$$x_3 \mid x_2 \mid x_1 \mid x_0 = x_3 \mid \left(\overline{x_2 \mid x_1 \mid x_0}\right).$$

3. ЗАВДАННЯ ДО КУРСОВОЇ РОБОТИ

На рис.3.1 приведена структурна схема генератора електричних сигналів заданої форми.

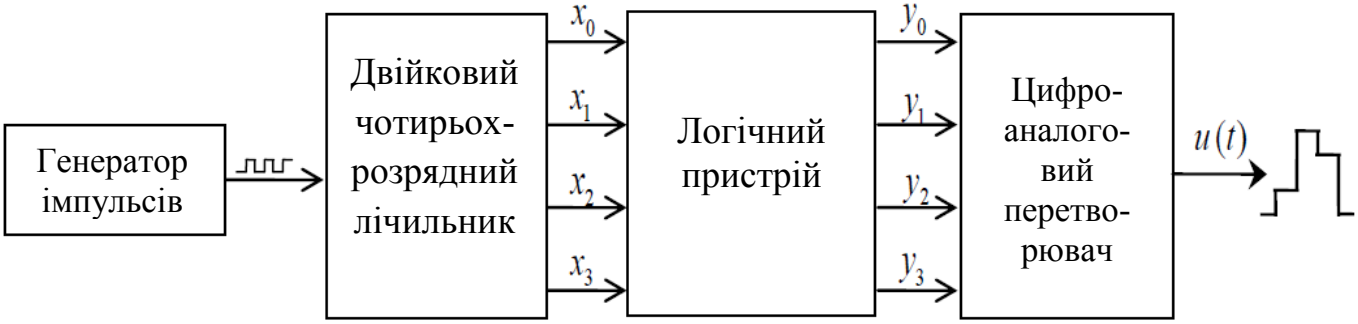


Рисунок – 3.1. Структурна схема генератора електричних сигналів

Від генератора імпульсів на двійковий чотирьохрозрядний лічильник надходять імпульси певної частоти. З приходом кожного наступного імпульсу значення двійкового чотирьохрозрядного числа на виході лічильника збільшується на одиницю, тобто вихідний сигнал лічильника змінюється від 0000 до 1111 (від 0 до 15 в десятковій системі числення), потім його значення знову стає рівним 0000 і процес повторюється. Чотири вихідні розряди лічильника імпульсів – x_0, x_1, x_2, x_3, надходять на вхід розроблюваного логічного пристрою, який здійснює перетворення вхідного 4-розрядного числа в вихідне 4-розрядне число.

До виходу логічного пристрою підключено цифроаналоговий перетворювач (ЦАП), напруга на виході якого пропорційна 4-разрядному двійковому числу, поданому на його вхід. В результаті на виході ЦАПа формується сигнал заданої форми.

У процесі виконання курсової роботи необхідно розробити функціональні та принципові електричні схеми комбінаційних логічних пристроїв, які забезпечують реалізацію сигналу заданої форми.

В завданні до курсової роботи передбачено 50 варіантів (див. додаток А), кожному з яких відповідає своя форма вихідного сигналу генератора.

При розробці принципових електричних схем передбачається використання сучасної елементної бази (див. додаток Б).

Для того щоб розробити логічний пристрій, відповідний завданням, необхідно виконати наступні дії:

1) Скласти таблицю істинності, яка відображає необхідні значення вихідних змінних логічного пристрою y_0, y_1, y_2 і y_3. Моменти часу, що задаються кожною комбінацією вхідних змінних x_0, x_1, x_2 і x_3. При заповненні таблиці істинності необхідно орієнтуватися на задану форму вихідного сигналу генератора;

2) З таблиці істинності отримати ДДНФ для кожної вихідної змінної логічного пристрою;

3) Мінімізувати отримані ФАЛ за допомогою діаграм Вейча;

4) Побудувати функціональну схему мінімізованого логічного пристрою в загальному логічному базисі ("И", "ИЛИ", "НЕ");

5) Привести мінімізовані ФАЛ до базисів "И – НЕ" і "ИЛИ – НЕ ";

6) Розробити функціональні і принципові електричні схеми в базисах "И–НЕ" і "ИЛИ–НЕ". При розробці принципових електричних схем необхідно користуватися

довідниками по цифровим інтегральним схемам або довідковим матеріалом, наведеним в додатку Б;

7) Підрахувати кількість логічних елементів і мікросхем, використаних для реалізації логічного пристрою в кожному розглянутому логічному базисі. Зробити висновок про доцільність використання того чи іншого логічного базису.

4. ПРИКЛАД ВИКОНАННЯ РОБОТИ

Для генератора електричних сигналів (рис 3.1) розробити логічний пристрій, який забезпечить на виході генератора сигнал, наведений на рис. 4.1.

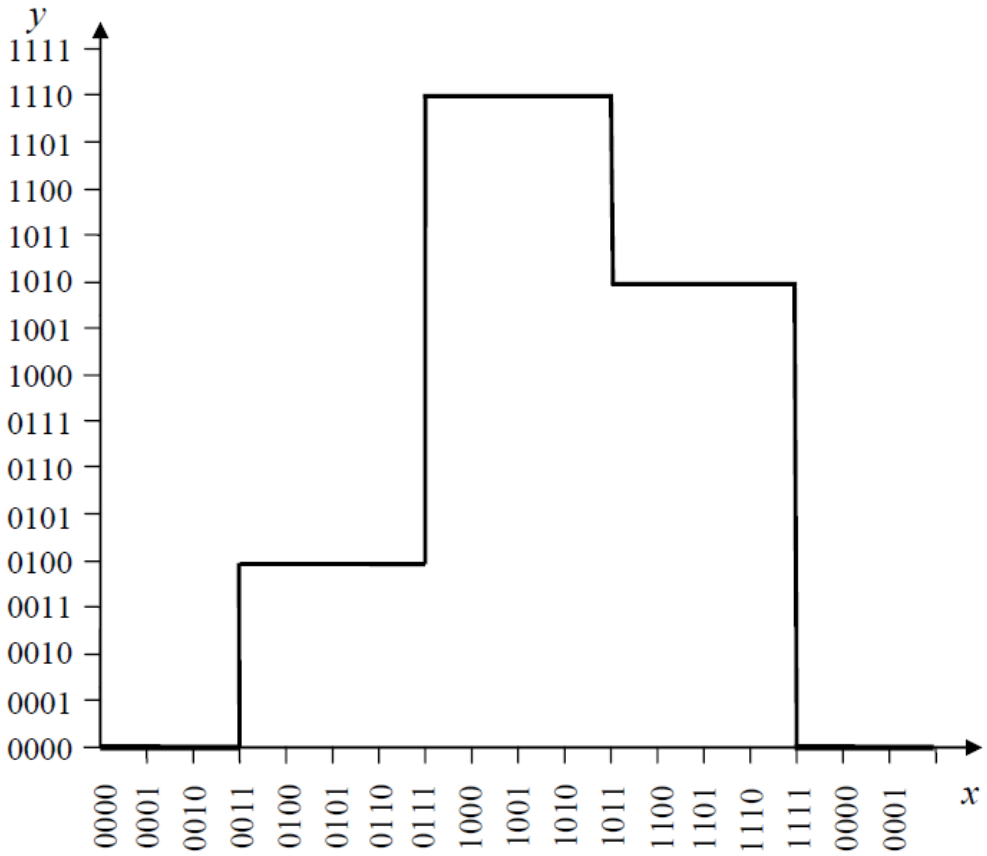


Рисунок 4.1 – Форма заданого сигналу

По осі абсцис кожний наступний момент часу відзначений 4-розрядним двійковим числом x_3, x_2, x_1, x_0 яке відповідає вихідному сигналу лічильника імпульсів (див. рис 3.1) і фазі вихідного сигналу генератора.

По осі ординат записані 4-розрядні двійкові числа y_3, y_2, y_1, y_0 , що представляють собою вхідний сигнал цифро-аналогового перетворювача. Таким чином, кожному вхідному сигналу проектованого логічного пристрою ми можемо поставити у відповідність вихідні сигнали, що забезпечують задану форму сигналу на виході генератора.

1) Складемо таблицю істинності логічного пристрою.

Ліва частина таблиці відповідає виходу лічильника і входу логічного пристрою, а права частина – виходу логічного пристрою і входу ЦАПа.

x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	1	1	1	0
1	0	0	0	1	1	1	0
1	0	0	1	1	1	1	0
1	0	1	0	1	1	1	0
1	0	1	1	1	0	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	0
1	1	1	0	1	0	1	0
1	1	1	1	0	0	0	0

2) По таблиці істинності складемо ДДНФ для кожної вихідної змінної логічного пристрою

$$y_0 = 0;$$

$$y_1 = \overline{x}_3 x_2 x_1 x_0 + x_3 \overline{x}_2 \overline{x}_1 \overline{x}_0 + x_3 \overline{x}_2 \overline{x}_1 x_0 + x_3 \overline{x}_2 x_1 \overline{x}_0 + \\ + x_3 \overline{x}_2 x_1 x_0 + x_3 x_2 \overline{x}_1 \overline{x}_0 + x_3 x_2 \overline{x}_1 x_0 + x_3 x_2 x_1 \overline{x}_0;$$

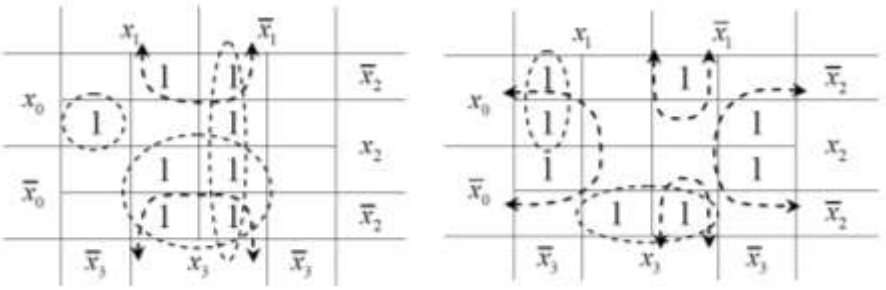
$$y_2 = \overline{x}_3 \overline{x}_2 x_1 x_0 + \overline{x}_3 x_2 \overline{x}_1 \overline{x}_0 + \overline{x}_3 x_2 \overline{x}_1 x_0 + \overline{x}_3 x_2 x_1 \overline{x}_0 + \\ + \overline{x}_3 x_2 x_1 x_0 + x_3 \overline{x}_2 \overline{x}_1 \overline{x}_0 + x_3 \overline{x}_2 \overline{x}_1 x_0 + x_3 \overline{x}_2 x_1 \overline{x}_0;$$

$$y_3 = y_1.$$

З таблиці істинності видно, що в даному прикладі значення функцій y_1 і y_3 приймають однакове значення для кожного моменту часу, тому для y_3 ДДНФ складати не потрібно.

3) За допомогою діаграм Вейча виконаємо мінімізацію отриманих ФАЛ.

$y_1 = f(x_0, x_1, x_2, x_3)$ $\qquad$ $y_2 = f(x_0, x_1, x_2, x_3)$



Отримаємо наступні мінімізовані функції:

$$y_1 = y_3 = x_3\overline{x}_0 + x_3\overline{x}_2 + x_3\overline{x}_1 + \overline{x}_3x_2x_1x_0;$$

$$y_2 = \overline{x}_3x_1x_0 + \overline{x}_3x_2 + x_3\overline{x}_2\overline{x}_0 + x_3\overline{x}_2\overline{x}_1.$$

4) Побудуємо функціональну схему мінімізованого логічного пристрою в загальному логічному базисі (див. рис.4.2).

5) Приведемо мінімізовані ФАЛ до базису "И-НЕ":

$$y_1 = y_3 = \overline{\overline{x_3\overline{x}_0 + x_3\overline{x}_2 + x_3\overline{x}_1 + \overline{x}_3x_2x_1x_0}} = \overline{\overline{x_3\overline{x}_0}\bullet\overline{x_3\overline{x}_2}\bullet\overline{x_3\overline{x}_1}\bullet\overline{\overline{x}_3x_2x_1x_0}} =$$
$$= \overline{(x_3 \mid \overline{x}_0)\bullet(x_3 \mid \overline{x}_2)\bullet(x_3 \mid \overline{x}_1)\bullet(\overline{x}_3 \mid x_2 \mid x_1 \mid x_0)} =$$
$$= (x_3 \mid \overline{x}_0) \mid (x_3 \mid \overline{x}_2) \mid (x_3 \mid \overline{x}_1) \mid (\overline{x}_3 \mid x_2 \mid x_1 \mid x_0)$$

$$y_2 = \overline{\overline{\overline{x}_3x_1x_0 + \overline{x}_3x_2 + x_3\overline{x}_2\overline{x}_0 + x_3\overline{x}_2\overline{x}_1}} = \overline{\overline{\overline{x}_3x_1x_0}\bullet\overline{\overline{x}_3x_2}\bullet\overline{x_3\overline{x}_2\overline{x}_0}\bullet\overline{x_3\overline{x}_2\overline{x}_1}} =$$
$$= \overline{(\overline{x}_3 \mid x_1 \mid x_0)\bullet(\overline{x}_3 \mid x_2)\bullet(x_3 \mid \overline{x}_2 \mid \overline{x}_0)\bullet(x_3 \mid \overline{x}_2 \mid \overline{x}_1)} =$$
$$= (\overline{x}_3 \mid x_1 \mid x_0) \mid (\overline{x}_3 \mid x_2) \mid (x_3 \mid \overline{x}_2 \mid \overline{x}_0) \mid (x_3 \mid \overline{x}_2 \mid \overline{x}_1)$$

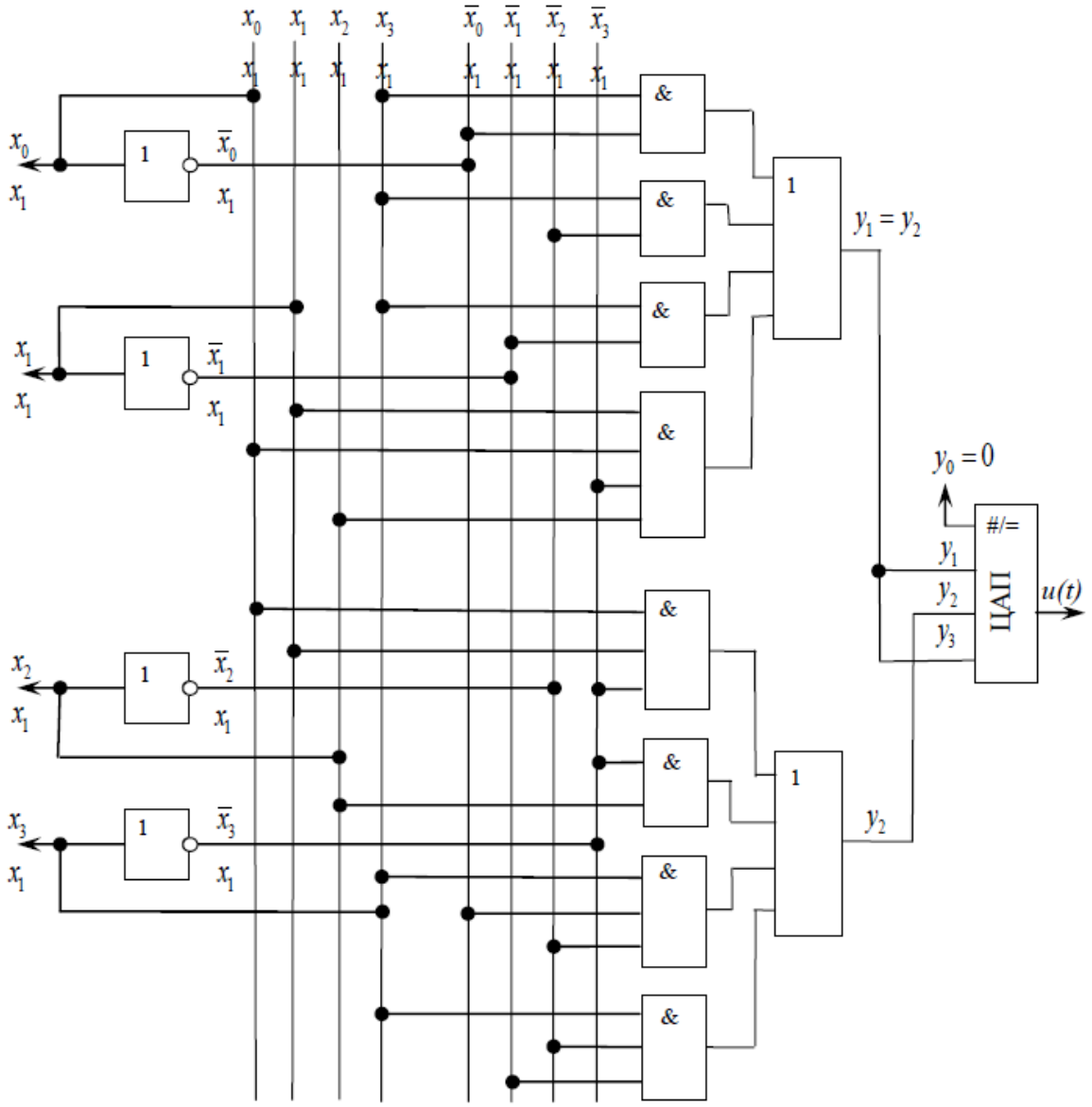


Рисунок 4.2 – Функціональна схема в загальному логічному базисі

Приведемо ФАЛ синтезованого логічного пристрою до базису "ИЛИ-НЕ":

$$y_1 = y_3 = \overline{\overline{x_3\bar{x}_0}} + \overline{\overline{x_3\bar{x}_2}} + \overline{\overline{x_3\bar{x}_1}} + \overline{\overline{\bar{x}_3 x_2 x_1 x_0}} =$$
$$= \overline{\bar{x}_3 + x_0} + \overline{\bar{x}_3 + x_2} + \overline{\bar{x}_3 + x_1} + \overline{x_3 + \bar{x}_2 + \bar{x}_1 + \bar{x}_0} =$$
$$= (\bar{x}_3 \downarrow x_0) + (\bar{x}_3 \downarrow x_2) + (\bar{x}_3 \downarrow x_1) + (x_3 \downarrow \bar{x}_2 \downarrow \bar{x}_1 \downarrow \bar{x}_0) =$$
$$= \overline{(\bar{x}_3 \downarrow x_0) \downarrow (\bar{x}_3 \downarrow x_2) \downarrow (\bar{x}_3 \downarrow x_1) \downarrow (x_3 \downarrow \bar{x}_2 \downarrow \bar{x}_1 \downarrow \bar{x}_0)}$$

$$y_2 = \overline{\overline{\bar{x}_3 x_1 x_0}} + \overline{\overline{\bar{x}_3 x_2}} + \overline{\overline{x_3 \bar{x}_2 \bar{x}_0}} + \overline{\overline{x_3 \bar{x}_2 \bar{x}_1}} =$$
$$= \overline{x_3 + \bar{x}_1 + \bar{x}_0} + \overline{x_3 + \bar{x}_2} + \overline{\bar{x}_3 + x_2 + x_0} + \overline{\bar{x}_3 + x_2 + x_1} =$$
$$= (x_3 \downarrow \bar{x}_1 \downarrow \bar{x}_0) + (x_3 \downarrow \bar{x}_2) + (\bar{x}_3 \downarrow x_2 \downarrow x_0) + (\bar{x}_3 \downarrow x_2 \downarrow x_1) =$$
$$= \overline{(x_3 \downarrow \bar{x}_1 \downarrow \bar{x}_0) \downarrow (x_3 \downarrow \bar{x}_2) \downarrow (\bar{x}_3 \downarrow x_2 \downarrow x_0) \downarrow (\bar{x}_3 \downarrow x_2 \downarrow x_1)}$$

6) При розробці принципових електричних схем логічного пристрою в базисах "І-НЕ" і "ИЛИ-НЕ" будемо використовувати цифрові мікросхеми серії КР1554 (див. додаток Б). Аналіз функціонального ряду мікросхем даної серії показує, що в ньому відсутні логічні елементи, що реалізують операцію "4ИЛИ-НЕ". Тому, використовуючи формулу:

$$x_2 \downarrow x_1 \downarrow x_0 = x_2 \downarrow \overline{x_1 \downarrow x_0},$$

виконаємо алгебраїчні перетворення, що виключають операцію "4ИЛИ-НЕ":

$$y_1 = y_3 = \overline{(\overline{x}_3 \downarrow x_0) \downarrow (\overline{x}_3 \downarrow x_2) \downarrow (\overline{x}_3 \downarrow x_1) \downarrow (x_3 \downarrow \overline{x}_2 \downarrow \overline{x}_1 \downarrow \overline{x}_0)} =$$
$$= \overline{(\overline{x}_3 \downarrow x_0) \downarrow (\overline{x}_3 \downarrow x_2) \downarrow (\overline{x}_3 \downarrow x_1) \downarrow (x_3 \downarrow \overline{(\overline{x}_2 \downarrow \overline{x}_1 \downarrow \overline{x}_0)})} =$$
$$= \overline{\overline{(\overline{x}_3 \downarrow x_0) \downarrow (\overline{x}_3 \downarrow x_2) \downarrow (\overline{x}_3 \downarrow x_1)} \downarrow (x_3 \downarrow \overline{(\overline{x}_2 \downarrow \overline{x}_1 \downarrow \overline{x}_0)})}$$

$$y_2 = \overline{(x_3 \downarrow \overline{x}_1 \downarrow \overline{x}_0) \downarrow (x_3 \downarrow \overline{x}_2) \downarrow (\overline{x}_3 \downarrow x_2 \downarrow x_0) \downarrow (\overline{x}_3 \downarrow x_2 \downarrow x_1)} =$$
$$= \overline{(x_3 \downarrow \overline{x}_1 \downarrow \overline{x}_0) \downarrow \overline{(x_3 \downarrow \overline{x}_2) \downarrow (\overline{x}_3 \downarrow x_2 \downarrow x_0) \downarrow (\overline{x}_3 \downarrow x_2 \downarrow x_1)}}$$

Принципову електричну схему розробленого логічного пристрою в логічному базисі "И-НЕ" наведено на рис.4.3, а в базисі "ИЛИ-НЕ" – на рис.4.4. Необхідно відзначити, що на даних схемах не показано кола живлення, а також не використано вентилі мікросхем.

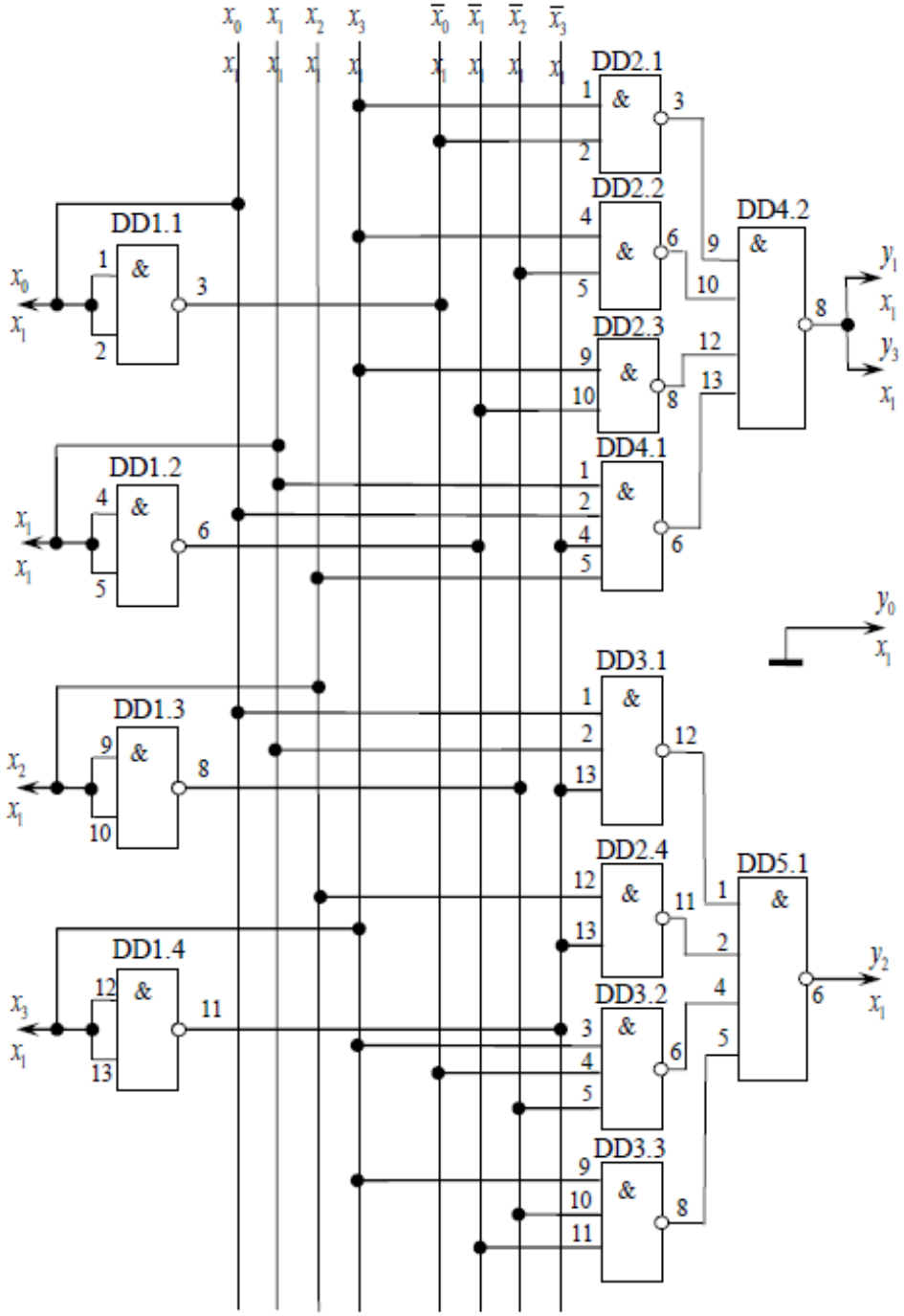


Рисунок 4.3 – Логічний пристрій. Схема електрична принципова в базисі "И-НЕ"

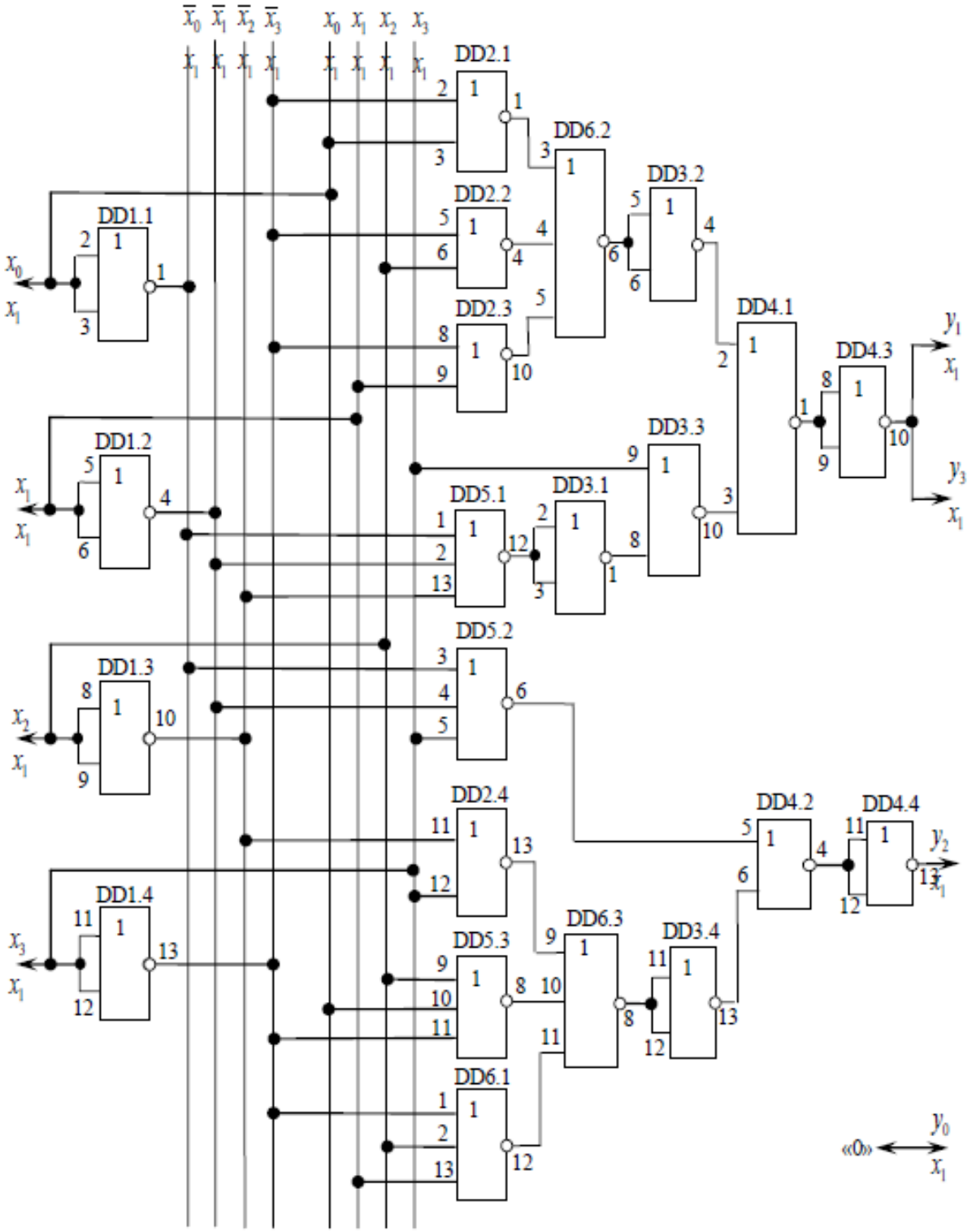


Рисунок 4.4 – Логічний пристрій. Схема електрична принципова в базисі "ИЛИ-НЕ"

7) При синтезі логічного пристрою в логічному базисі "ИНЕ" треба було в цілому 14 логічних елементів, які містяться в 5 мікросхемах. При цьому незадіяним залишився один елемент мікросхеми КР1554ЛА1 (DD5.2).

Позначення елементів	Тип мікросхеми	Функціональне наповнення мікросхем	Кількість мікросхем
DD1…DD2	КР1554ЛА3	“2И-НЕ”×4	2
DD3	КР1554ЛА4	“3И-НЕ”×3	1
DD4…DD5	КР1554ЛА1	“4И-НЕ”×2	2

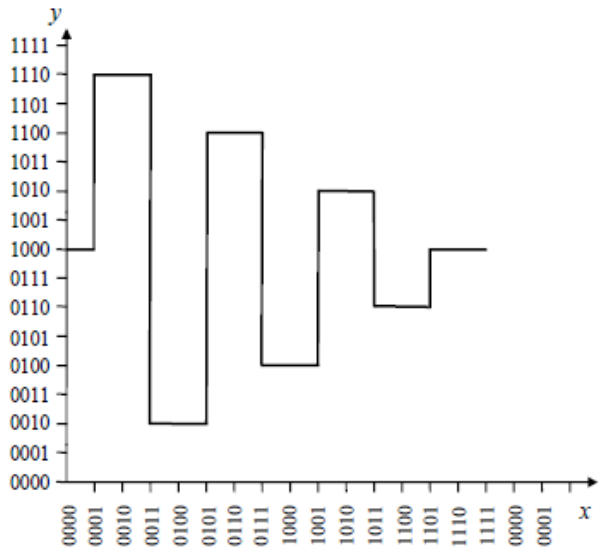
При застосуванні базису "ИЛИ-НЕ" було використано 22 логічних елементи, які містяться в 6 мікросхемах.

Позначення елементів	Тип мікросхеми	Функціональне наповнення мікросхем	Кількість мікросхем
DD1…DD4	КР1554 ЛЕ1	“2ИЛИ-НЕ”×4	4
DD5…DD6	КР1554ЛЕ4	“3ИЛИ-НЕ”×3	2

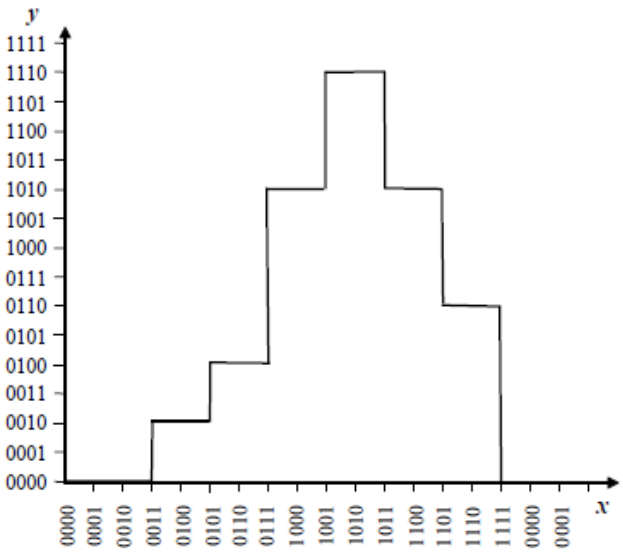
Таким чином, з точки зору матеріальних витрат і трудомісткості для реалізації даного логічного пристрою краще використовувати логічний базис "И-НЕ", тому що при цьому потрібна менша кількість мікросхем і відсутня необхідність алгебраїчних перетворень по виключенню операцій "4ИЛИ-НЕ".

ДОДАТОК А

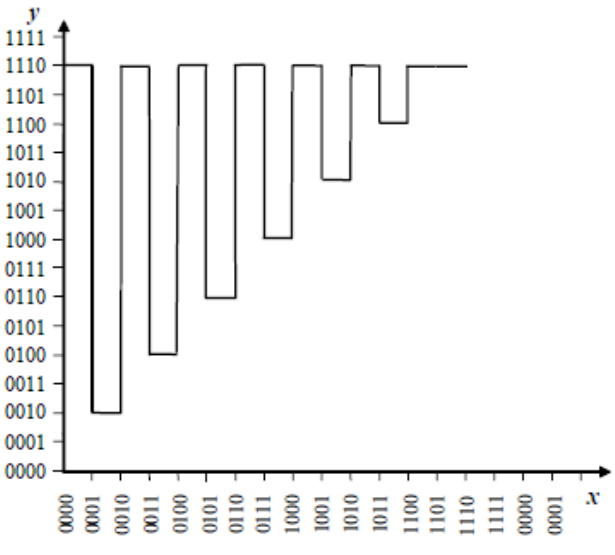
ВАРІАНТИ ЗАВДАНЬ



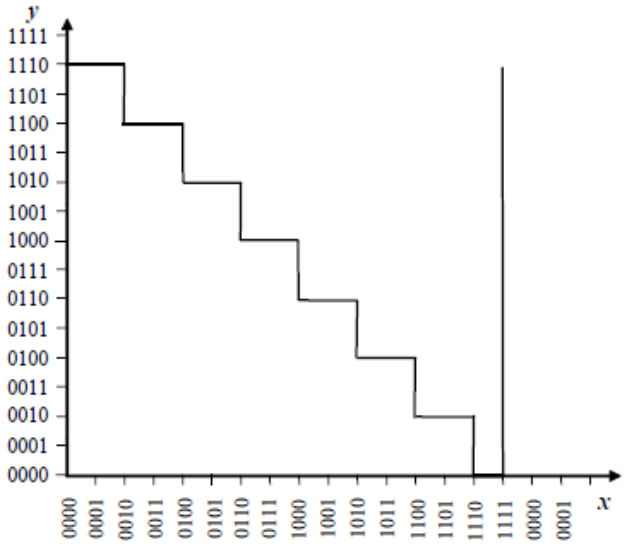
Вариант 1



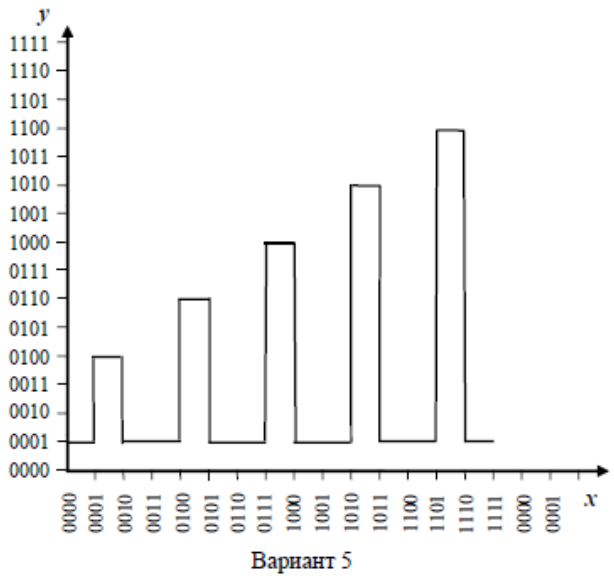
Вариант 2



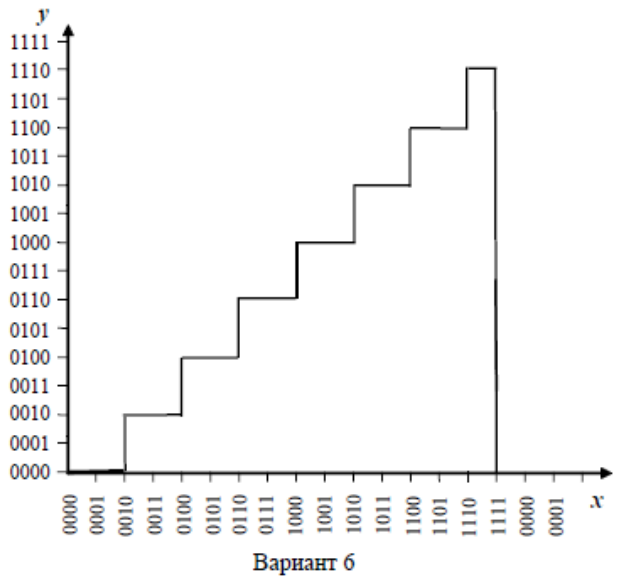
Вариант 3



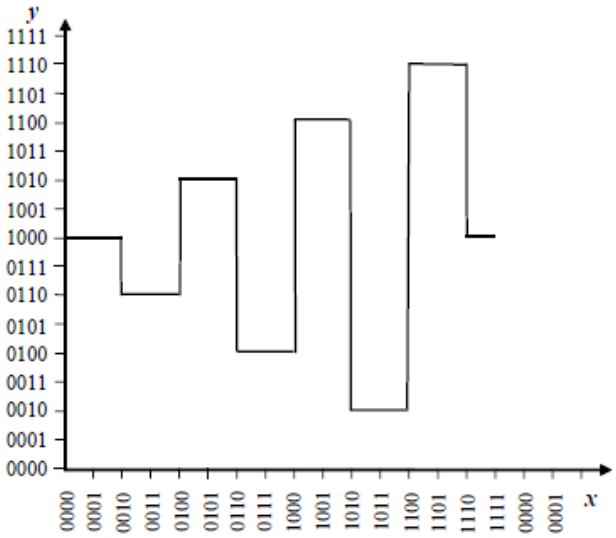
Вариант 4



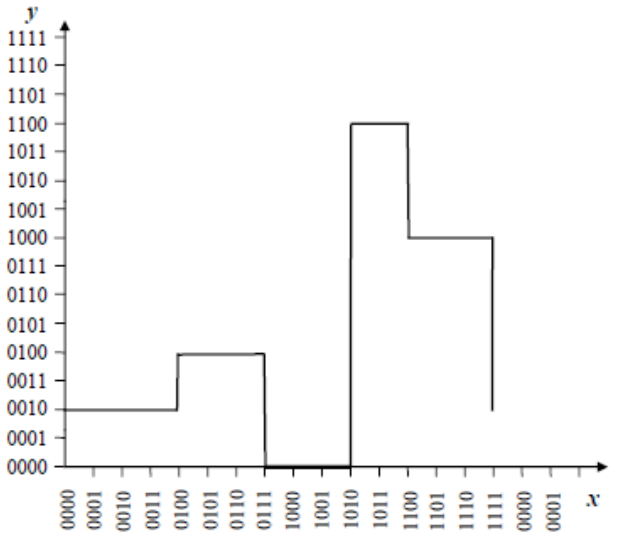
Вариант 5



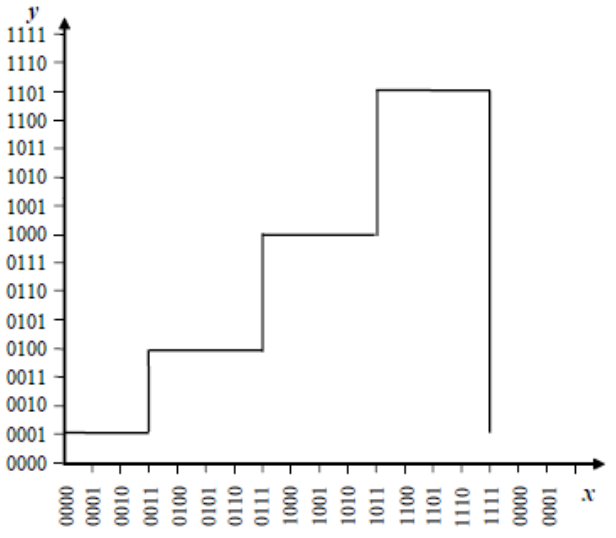
Вариант 6



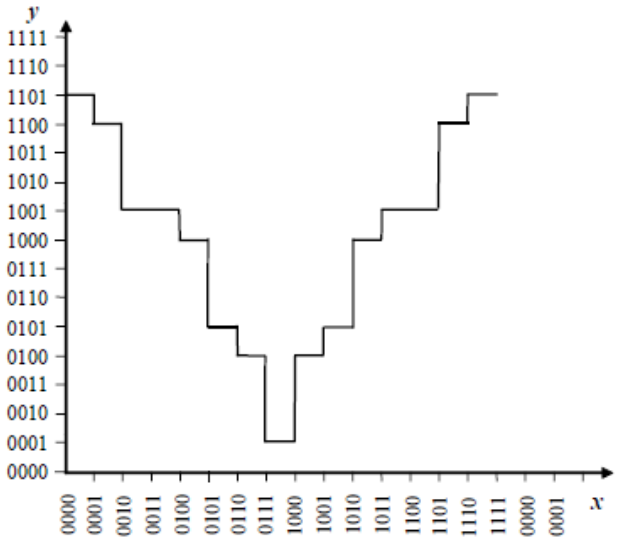
Вариант 7



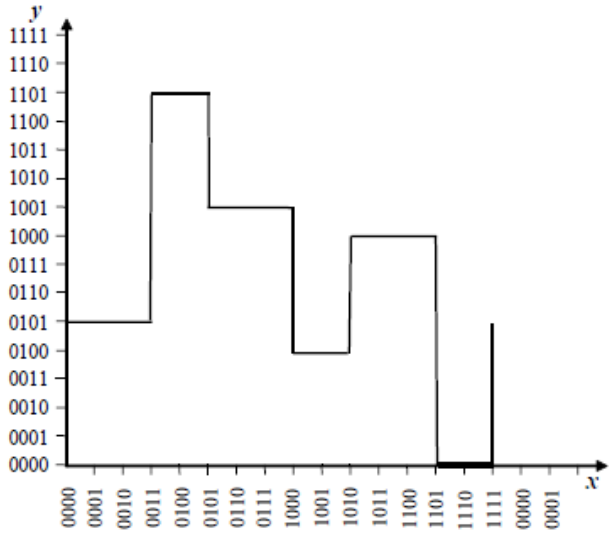
Вариант 8



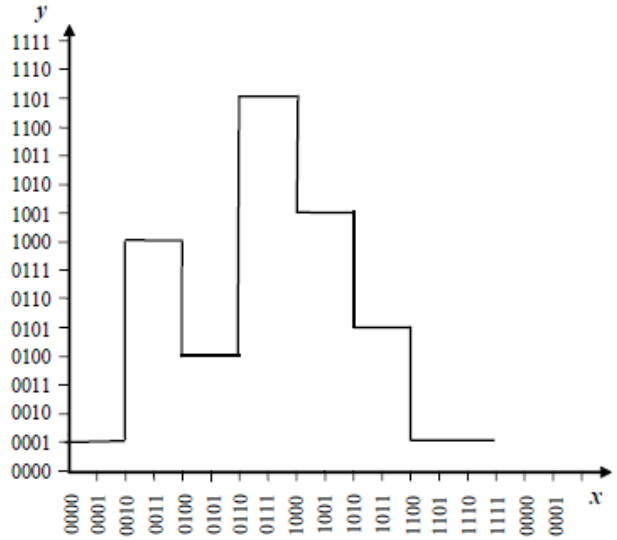
Вариант 9



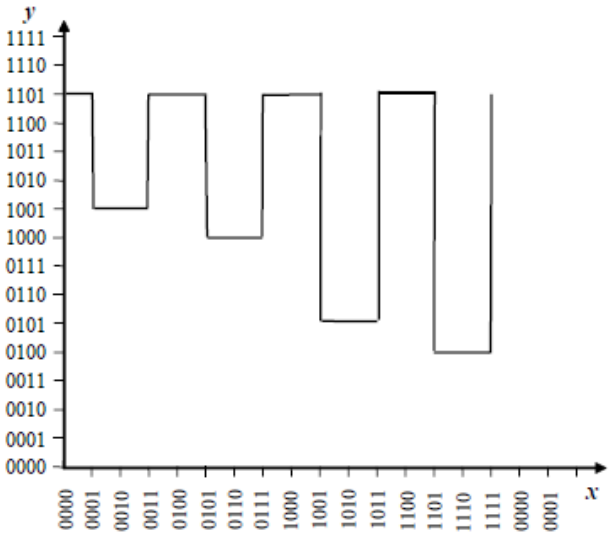
Вариант 10



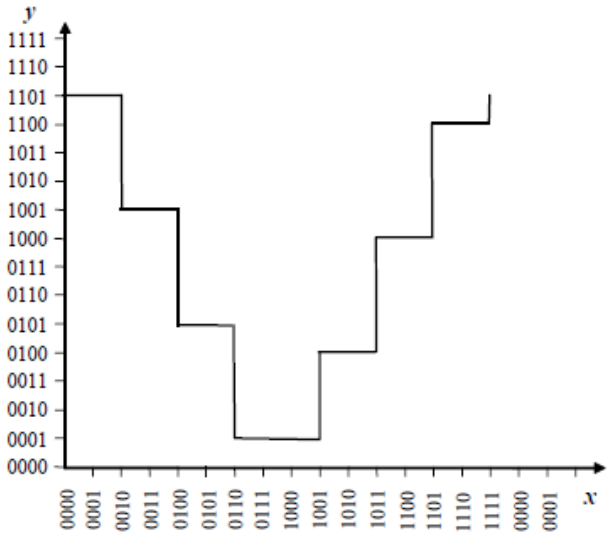
Вариант 11



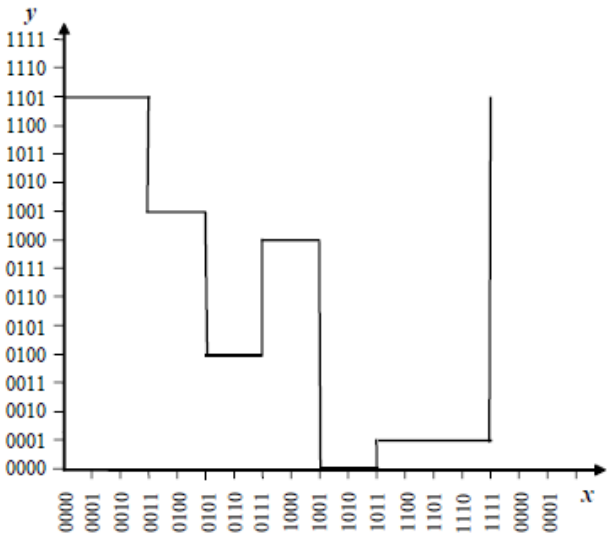
Вариант 12



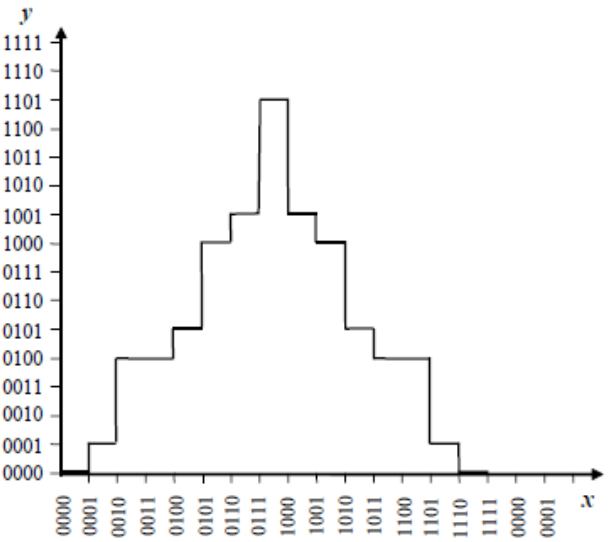
Вариант 13



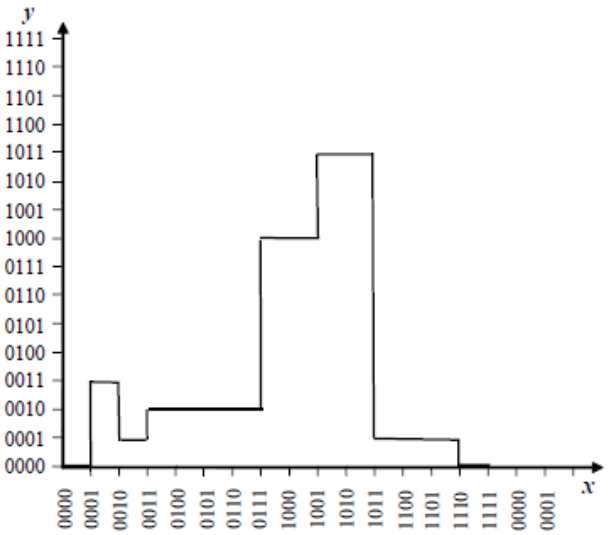
Вариант 14



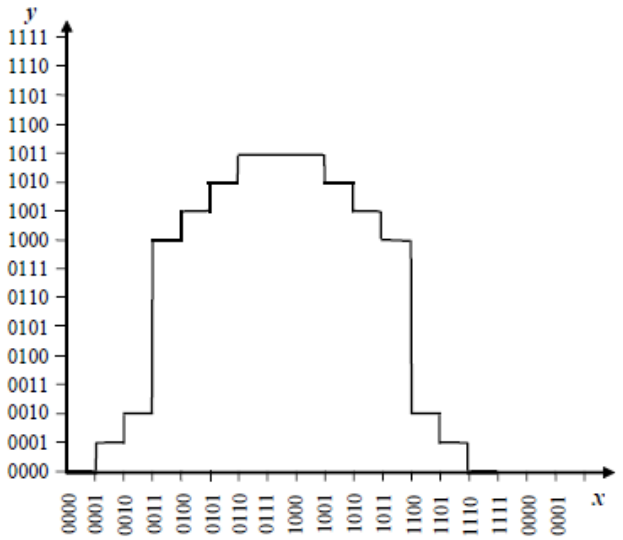
Вариант 15



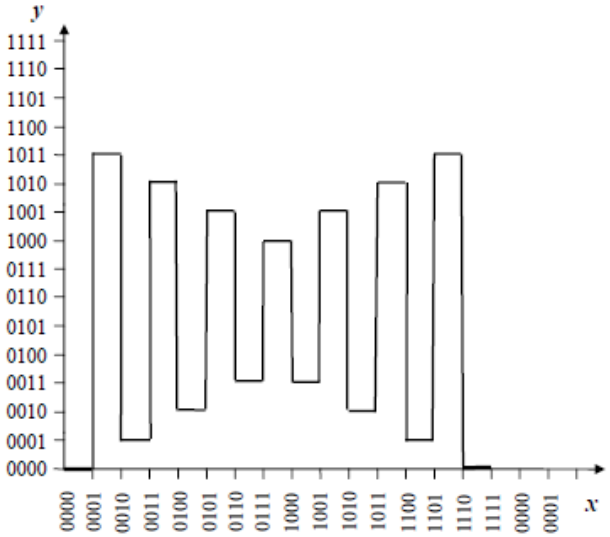
Вариант 16



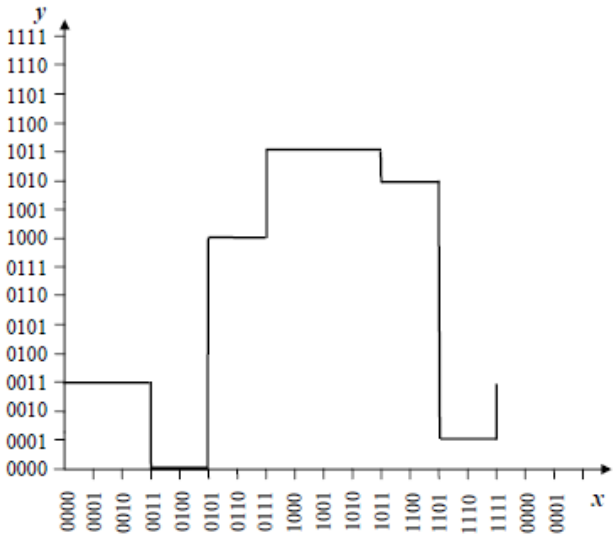
Вариант 17



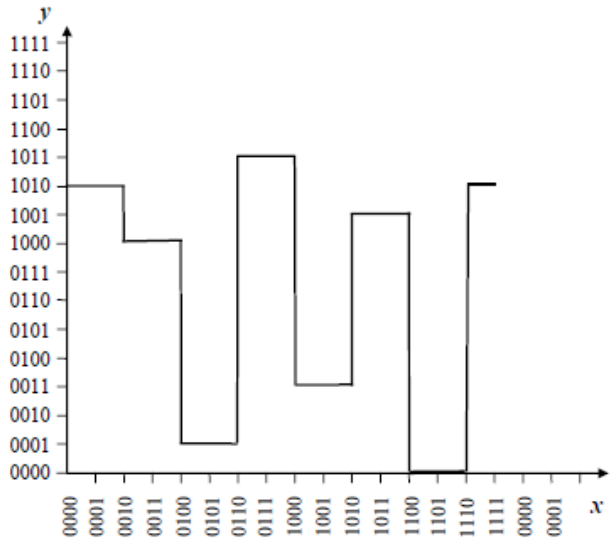
Вариант 18



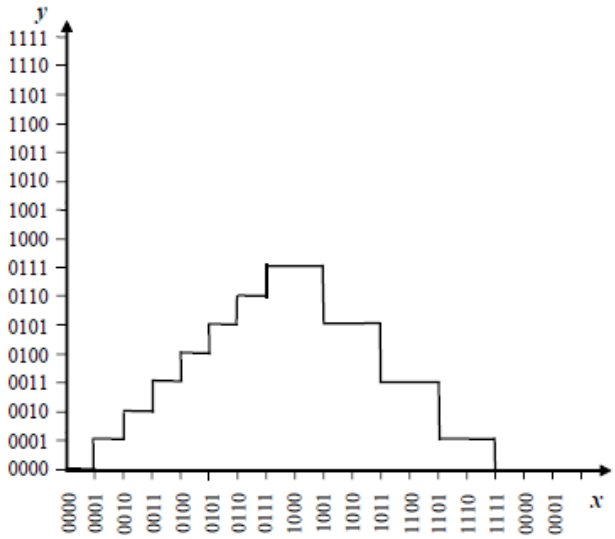
Вариант 19



Вариант 20



Вариант 21



Вариант 22