

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»
ІНЖЕНЕРНО-ТЕХНІЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ

КУРСОВА РОБОТА

з дисципліни

«Комп'ютерна логіка»

Напрямок підготовки 6.050102 – «Комп'ютерна інженерія»

«Синтез комбінаційних схем. Синтез автоматів»

Варіант № 2.3

Виконав

Студент 2 курсу

Лялько Руслан Михайлович

Науковий керівник

Ст. в. Тютюнникова Г. С.

Робота допущена до захисту

«.....».....2011р.

Науковий керівник.....

Робота захищена

«.....».....2011р.

З оцінкою.....

Науковий керівник.....

УЖГОРОД–2011

ЗМІСТ

ВСТУП.....	3
ТЕОРЕТИЧНА ЧАСТИНА	4
I. СИНТЕЗ КОМБІНАЦІЙНИХ СХЕМ	4
1. Представлення булевих функцій за допомогою таблиць істинності.....	4
2. Представлення булевих функцій за допомогою канонічних форм	5
3. Розвинення логічних функцій за змінними. Розклад Шеннона	6
4. Повнота системи логічних функцій	8
5. Мінімізація булевих функцій за допомогою мінімізуючих карт	9
6. Синтез комбінаційних схем	17
7. Синтез комбінаційних схем за допомогою дешифраторів та мультимплексорів	18
II. СИНТЕЗ АВТОМАТІВ.....	20
ПРАКТИЧНА ЧАСТИНА	23
I. СИНТЕЗ КОМБІНАЦІЙНИХ СХЕМ	23
1. Перевірка булевих функцій на наявність фіктивних аргументів.....	23
2. Представлення булевих функцій за допомогою канонічних форм.	28
3. Розклад Шеннона, представлення функції f_4 з допомогою функцій від двох змінних.	31
4. Визначення належності логічних функцій до п'яти предповних класів...	32
5. Мінімізація булевих функцій.....	36
6. Реалізація логічних функцій у заданому базисі.....	51
7. Синтез комбінаційних схем, за допомогою дешифраторів та мультимплексорів.	53
II. СИНТЕЗ АВТОМАТІВ.....	58
1. Синтез цифрового автомата Мілі	58
ВИСНОВКИ.....	62
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	63

ВСТУП

У першій частині курсової роботи розглядається синтез комбінаційних схем. Задача синтезу полягає в побудові із заданого набору логічних елементів комбінаційної схеми, що реалізує задану систему булевих функцій.

Розв'язок задачі синтезу не є однозначним, можна запропонувати різні варіанти комбінаційних схем, що реалізують одну й ту ж систему булевих функцій. Вибір конкретного варіанту здійснюється виходячи з додаткових критеріїв: типу базису, мінімальної кількості логічних елементів, потрібних для реалізації схеми, максимальної швидкодії і т.д. Існують різні методи синтезу комбінаційних схем, серед яких найбільш повно розробленим є канонічний метод.

Друга частина роботи присвячена синтезу цифрових абстрактних автоматів, тобто схем з пам'яттю. Найбільш поширеними на практиці є автомати Мілі та Мура, також застосовуються С-автомати, які об'єднують властивості автоматів Мілі і Мура. Скінчені автомати Мілі і Мура є основою для побудови керуючих автоматів із схемною логікою.

ТЕОРЕТИЧНА ЧАСТИНА

I. СИНТЕЗ КОМБІНАЦІЙНИХ СХЕМ

1. Представлення булевих функцій за допомогою таблиць істинності

Булева (логічна) змінна – це така змінна, яка може приймати лише два значення: $x = \{0, 1\}$.

Логічні змінні, як і змінні звичайної алгебри, позначають буквами латинського алфавіту або якою-небудь буквою з різними індексами, наприклад: x, y, z, x_0, x_1, x_2 .

Булевою (логічною або перемикальною) функцією називається функція, яка, як і її n аргументів, може приймати лише два значення: 0 і 1.

Функція, яка залежить від n аргументів, називається n -місною і є повністю заданою (повністю визначеною), якщо вказані її значення для всіх булевих (двійкових) наборів значень аргументів.

Довільна булева функція може бути задана одним із трьох способів: *геометричним, аналітичним та табличним*.

При *геометричному способі* булева функція задається за допомогою n -вимірного кубу. У цьому випадку, набори значень аргументів називають точками, оскільки кожний із них може бути ототожнений із певною вершиною одиничного n -вимірного кубу.

При *аналітичному способі* булева функція задається формулами, тобто аналітичними виразами, побудованими на основі операцій булевої алгебри. Аналітичний спосіб задання булевих функцій займає особливе місце в проектуванні цифрових пристроїв, оскільки усі перетворення булевих функцій, які необхідні для побудови цифрових схем, виконуються саме в аналітичному вигляді.

Скінченність області визначення та області значень булевих функцій дає можливість задавати їх за допомогою таблиць, які називаються *таблицями*

істинності. Номери наборів n -місної булевої функції змінюються від 0 до $2^n - 1$. Розташувавши набори стовпчиком у порядку зростання їх номерів і поклавши значення функції проти кожного набору, одержимо таблицю істинності булевої функції. Табличний спосіб задання функції прийнято для функції невеликої кількості змінних, оскільки зі збільшенням кількості змінних (n) кількість наборів збільшиться як 2^n і таблиця істинності стає громіздкою.

2. Представлення булевих функцій за допомогою канонічних форм

2.1 Представлення функцій у ДДНФ

Нехай маємо функцію $f(x_1, x_2, \dots, x_n)$, яку будемо записувати як $f_k^1(x_1, x_2, \dots, x_n)$.

Досконала диз'юнктивна нормальна форма (ДДНФ) має вигляд:

$$f(x_1, x_2, \dots, x_n) = f_k^1(x_1, x_2, \dots, x_n) = \bigvee_{\alpha_1, \dots, \alpha_n} x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot \dots \cdot x_n^{\alpha_n} = K_1^1 + K_2^1 + \dots + K_m^1, \text{ де } K_i^1$$

– мінтерми, на яких функція дорівнює 1.

Мінтермом називається функція від n змінних, яка дорівнює одиниці тільки на одному наборі.

2.2 Представлення функцій у ДКНФ

Логічна сума будь-якої кількості різних змінних, що входять із запереченням або без нього, називається *елементарною диз'юнкцією*.

Кон'юнкція елементарних диз'юнкцій вигляду:

$$F_D = D_1 + D_2 + \dots + D_k,$$

– називається кон'юнктивною нормальною формою (КНФ).

Досконала кон'юнктивна нормальна форма (ДКНФ) має вигляд:

$$f_d^0(x_1, x_2, \dots, x_n) = D_1^0 \cdot D_2^0 \cdot \dots \cdot D_3^0$$

де D_i^0 – макстерми, на яких функція дорівнює 0.

Макстермом називається функція n змінних, яка дорівнює нулю тільки на одному наборі.

2.3 Представлення функцій у вигляді поліному Жегалкіна

Алгебра $(B, \wedge, \oplus, 0, 1)$, що утворена множиною $B = \{0, 1\}$ разом з операціями $\wedge$ (кон'юнкції), $\oplus$ (XOR, сума за модулем 2) і константами 0, 1, називається *алгеброю Жегалкіна*.

Поліномом Жегалкіна називається скінчена сума за модулем 2 попарно різних елементарних кон'юнкцій над множиною змінних $(x_1, x_2, \dots, x_n)$. Кількість змінних, що входять до елементарної кон'юнкції, називається *рангом елементарної кон'юнкції*.

Кількість попарно різних елементарних кон'юнкцій у поліномі називається *довжиною полінома*.

Канонічний поліном Жегалкіна – це поліном вигляду:

$$f(x_{n-1}, \dots, x_1, x_0) = a_0 \oplus a_1 x_0 \oplus a_2 x_1 \oplus \dots \oplus a_n x_{n-1} \oplus a_{12} x_0 x_1 \oplus a_{13} x_0 x_2 \oplus \dots \oplus a_{12\dots n} x_0 x_1 \dots x_{n-1},$$
 де $a_i = \{0, 1\}$.

Теорема Жегалкіна. Будь-яка логічна функція може бути представлена у вигляді канонічного поліному Жегалкіна.

3. Розвинення логічних функцій за змінними. Розклад Шеннона

Нехай маємо деякі двійкові змінні x та α : $x = \{0, 1\}$, $\alpha = \{0, 1\}$. Введемо позначення $x^0 = \bar{x}$, $x^1 = x$. Тоді змінну x у прямому або інверсному вигляді можна задати як деякий вираз вигляду:

$$x^\alpha = \begin{cases} \bar{x}, & \text{якщо } \alpha = 0, \\ x, & \text{якщо } \alpha = 1, \end{cases}$$

При цьому буде справедлива наступна рівність

$$x^\alpha = \alpha \wedge x \vee \bar{\alpha} \wedge \bar{x} \quad (x^\alpha = \alpha x + \bar{\alpha} \bar{x}). \quad (3.1)$$

Вираз x^α дорівнює одиниці лише тоді, коли $x = \alpha$.

Теорема. Будь-яку функцію алгебри логіки можна подати у вигляді

$$\begin{aligned} f(x_1, x_2, \dots, x_k, x_{k+1}, \dots, x_n) = \\ = \bigvee_{\alpha_1, \alpha_2, \dots, \alpha_k} f(\alpha_1, \alpha_2, \dots, \alpha_k, x_{k+1}, \dots, x_n) \wedge x_1^{\alpha_1} \wedge x_2^{\alpha_2} \wedge \dots \wedge x_k^{\alpha_k} \end{aligned} \quad (3.2)$$

де $\vee$ – символ узагальненої диз'юнкції на множині двійкових наборів $\alpha'_1, \alpha'_2, \dots, \alpha'_k$, кількість яких дорівнює 2^k :

$$\left. \begin{array}{cccc} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \end{array} \right\} 2^k$$

$\underbrace{\hspace{1.5cm}}_k$

Формула (3.2) називається диз'юнктивним розвиненням функції алгебри логіки за k змінними.

Наслідок 1. Якщо $k=1$, то функцію алгебри логіки можна подати у вигляді

$$f(x_1, x_2, \dots, x_n) = f(0, x_2, \dots, x_n) \wedge \bar{x}_1 \vee f(1, x_2, \dots, x_n) \wedge x_1. \quad (3.3)$$

Перевіримо справедливість, одержаної формули безпосередньою перевіркою. Дійсно, при $x_1 = 0$

$$\begin{aligned} f(0, x_2, \dots, x_n) &= \bar{0} \wedge f(0, x_2, \dots, x_n) \vee 0 \wedge f(1, x_2, \dots, x_n) = \\ &= 1 \wedge f(0, x_2, \dots, x_n) = f(0, x_2, \dots, x_n); \end{aligned}$$

при $x_1 = 1$

$$\begin{aligned} f(1, x_2, \dots, x_n) &= \bar{1} \wedge f(0, x_2, \dots, x_n) \vee 1 \wedge f(1, x_2, \dots, x_n) = \\ &= 0 \wedge f(0, x_2, \dots, x_n) \vee 1 \wedge f(1, x_2, \dots, x_n) = f(1, x_2, \dots, x_n). \end{aligned}$$

Таким чином, ми одержали, що ліва частина рівності дорівнює правій частині при всіх значеннях змінної x_1 .

Наслідок 2. Якщо $k=2$, то функцію алгебри логіки можна подати у вигляді

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= f(0, 0, x_3, \dots, x_n) \wedge \bar{x}_1 \wedge \bar{x}_2 \vee f(0, 1, x_3, \dots, x_n) \wedge \bar{x}_1 \wedge x_2 \vee \\ &\vee f(1, 0, x_3, \dots, x_n) \wedge x_1 \wedge \bar{x}_2 \vee f(1, 1, x_3, \dots, x_n) \wedge x_1 \wedge x_2. \end{aligned} \quad (3.4)$$

Наслідок 3. Якщо $k=n$, то функцію алгебри логіки можна подати у вигляді

$$f(x_1, x_2, \dots, x_n) = \bigvee_{\alpha_1, \alpha_2, \dots, \alpha_n} x_1^{\alpha_1} \wedge x_2^{\alpha_2} \wedge \dots \wedge x_n^{\alpha_n}. \quad (3.5)$$

Оскільки в формулі (3.5) логічне сумування здійснюється за всіма наборами, на яких функція дорівнює одиниці, то вона є не що інше як ДДНФ функції алгебри логіки, яку, як відомо, можна подати у вигляді

$$f(x_1, x_2, \dots, x_n) = f_K^1 = K_1^1 + K_2^1 + \dots + K_m^1, \quad (3.6)$$

де K_i^1 ($i = \overline{1, m}$) – мінтерми функції f .

4. Повнота системи логічних функцій

Функціонально повною системою булевих функцій (базисом) називається сукупність таких булевих функцій $\{f_1, f_2, \dots, f_m\}$, що будь-яка логічна функція може бути виражена через функції цієї сукупності за допомогою принципу суперпозиції.

Множина всіх функцій, що є суперпозиціями функцій деякого класу F , і лише їх, називається *замиканням* цього класу й позначається $[F]$.

Множина функцій S називається *передповним класом* алгебри A , якщо $[S] \neq F$ і для будь-якої функції f з множини $F \setminus S$ набір $S \cup \{f\}$ є повним: $[S \cup \{f\}] = F$.

Коротко розглянемо кожний із п'яти передповних класів.

Клас функцій, які зберігають нуль (K_0). Якщо функція на нульовому наборі дорівнює нулю, то кажуть, що функція зберігає нуль: $f(0, 0, \dots, 0) = 0$.

Клас функцій, які зберігають одиницю (K_1). Якщо функція на одиничному наборі дорівнює одиниці, то кажуть, що функція зберігає одиницю: $f(1, 1, \dots, 1) = 1$.

Клас самодвоїстих функцій (K_c). Логічна функція є самодвоїстою, якщо на кожній парі протилежних наборів вона приймає протилежні значення, тобто $f(x_1, x_2, \dots, x_n) = \overline{f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)}$.

Клас лінійних функцій (K_L). Логічна функція називається лінійною, якщо вона може бути подана у вигляді полінома Жегалкіна першого степеня:

$$f(x_{n-1}, \dots, x_1, x_0) = k_0 \oplus k_1 x_0 \oplus k_2 x_1 \oplus \dots \oplus k_n x_{n-1}.$$

Клас монотонних функцій (K_M). Логічна функція називається монотонною, якщо для будь-яких двох двійкових наборів α і β , що задовольняють умову $\alpha \leq \beta$, справджується нерівність $f(\alpha) \leq f(\beta)$.

Теорема. Булева функція, відмінна від константи 0 і 1, є монотонною, якщо і тільки якщо вона може бути зображена формулою булевої алгебри без заперечень.

Теореми Поста:

Теорема 1. Для того, щоб система логічних функцій була функціонально повною, необхідно і достатньо, щоб ця система містила хоча би одну функцію, що: не зберігає нуль, не зберігає одиницю, не є самодвоїстою, не є лінійною, не є монотонною.

Теорема 2. Система булевих функцій є функціонально повною тоді і тільки тоді, коли вона повністю не міститься ні в одному з передповних класів.

5. Мінімізація булевих функцій за допомогою мінімізуючих карт

У загальному випадку *мінімізацією* логічної функції прийнято називати процес відшукування найменшого за складністю аналітичного зображення даної функції у вигляді суперпозиції функцій, належних деякій функціонально повній системі.

Методи мінімізації розробляються стосовно кожної функціонально повної системи елементарних логічних функцій. Найдокладніше такі методи розроблено для випадку, коли функціонально повна система елементарних логічних функцій складається з диз'юнкції, кон'юнкції та заперечення. При цьому мінімізація логічної функції зводиться до визначення такої її форми,

яка має найменший індекс (коефіцієнт) простоти, що характеризує складність ДНФ (КНФ).

Індексом простоти $L(f)$ зображення функції $f(x_1, x_2, \dots, x_n)$ у вигляді ДНФ (ДКФ) називається таке найменше число L , для якого існує формула, що зображає дану функцію і має точно L відповідних елементів, які покладено в основу мінімізації. Найчастіше вживаються такі типи індексів простоти:

$L_1(f)$ – сумарна кількість букв, (а не змінних), що до неї входять;

$L_2(f)$ – число елементів диз'юнкції (кон'юнкції), що входять в формулу;

$L_3(f)$ – число символів інверсії, які зустрічаються в записі ДНФ.

З наведених індексів простоти на практиці найчастіше використовується індекс простоти $L_1(f)$, тобто найменша кількість букв.

При розв'язанні задачі мінімізації істотну роль відіграють поняття імпліканти, простої імпліканти, імпліценти і простої імпліценти.

Розглянемо три повністю визначені функції f , g , h . Функція f визначена на множині одиничних наборів $M_1[f]$ і множині нульових наборів $M_0[f]$. Функція g визначена на множині одиничних наборів $M_1[g]$, а функція h – на множині нульових наборів $M_0[h]$.

Логічна функція g називається *імплікантою* логічної функції f , якщо множина одиничних наборів функції g збігається або є підмножиною множини одиничних наборів функції f , тобто $M_1[g] \subseteq M_1[f]$.

Логічна функція h називається *імпліцентою* логічної функції f , якщо множина нульових наборів функції h збігається або є підмножиною множини нульових наборів функції f , тобто $M_0[h] \subseteq M_0[f]$.

Простою імплікантою функції f називається будь-яка елементарна кон'юнкція g , яка є імплікантою цієї функції та має ту властивість, що ніяка її власна частина (тобто кон'юнкція, одержана з елементарної шляхом

вилучення однієї або декількох змінних в прямій або інверсній формі) уже не є імплікантою даної функції.

Простою імпліцентовою функції f називається будь-яка елементарна диз'юнкція h , яка є імпліцентовою цієї функції та має ту властивість, що ніяка її власна частина (тобто диз'юнкція, одержана з елементарної шляхом вилучення однієї або декількох змінних в прямій або інверсній формі) уже не є імпліцентовою даної функції.

Диз'юнкція (кон'юнкція) всіх простих імплікант (імпліцент) логічної функції називається *скороченою диз'юнктивною (кон'юнктивною) нормальною формою* СДНФ (СКНФ).

Мінімальною диз'юнктивною (кон'юнктивною) нормальною формою МДНФ (МКНФ) функції f називається така ДНФ (КНФ), яка зображає дану функцію і містить найменшу кількість букв порівняно з усіма іншими еквівалентними їй ДНФ (КНФ).

5.1 Мінімізація з допомогою К-карт

К-карта для функції від чотирьох змінних має вигляд

	$\bar{x}_0$	x_0	x_0	$\bar{x}_0$	
$\bar{x}_1$	0000	0001	0101	0100	$\bar{x}_3$
x_1	0010	0011	0111	0110	$\bar{x}_3$
x_1	1010	1011	1111	1110	x_3
$\bar{x}_1$	1000	1001	1101	1100	x_3
	$\bar{x}_2$	$\bar{x}_2$	x_2	x_2	

В основу мінімізації за допомогою К-карт покладено операцію склеювання вмісту *сусідніх* або *відповідних* клітин.

Сусідніми називаються клітини, які геометрично сусідні (тобто мають спільну сторону). До сусідніх відносяться і клітки крайніх лівого і правого стовпців, а також крайніх нижнього і верхнього рядків будь-якої К-карти для

$n \leq 4$ або будь-якого блоку 4×4 клітин – для $n > 4$ (вони стають сусідніми при згортанні К-карти або блоку в горизонтальний або вертикальний циліндр).

За даною логічною функцією f_1 складемо К-карту (рис. 5.1), в якій відображено двійкові набори, на яких функція приймає одиничні значення.

На побудованій К-карті для функцією f_1 виділимо прямокутні області, які об'єднують вибрані одиничні значення функції. Кожна область повинна містити 2^k , $k = 0, 1, 2, \dots$ клітинок. Вибрані області можуть перетинатися, тобто одна або декілька клітинок можуть належати різним областям. Чим менше прямокутників і чим більше клітин у прямокутниках, тим краще покриття.

Після цього записуємо двійкові набори, які належать кожній прямокутній області в таблицю (таблиця 5.1). Склеювання можна проводити у тих стовпчиках двійкових наборів, в яких кількість нулів і одиниць однакова. Склейки відмічено символом "*". Провівши склеювання, потрібно логічно провести логічне сумування імплікант, які відповідають вибраним областям. Одержана сума утворює мінімальну диз'юнктивну нормальну форму (МДНФ).

5.2 Мінімізація методом Квайна

При мінімізації за методом Квайна в базисі $\{ \neg, \vee, \wedge \}$ вважають, що функція задана в ДНФ і використовується поняття первинної або простої імпліканти.

Метод Квайна полягає в застосуванні до імплікант, які входять в ДНФ функції f закону поглинання $Fx + F\bar{x} = F$. Використовуючи операцію поглинання вдається знизити ранг термів. Ця процедура продовжується до тих пір, поки не залишиться жодного терму, який допускає поглинання з яким-небудь іншим термом. Терм над яким було здійснено поглинання відмічаються. Не відмічені терми представляють собою первинні імпліканти.

Алгоритм мінімізації методом Квайна зводиться до наступних етапів:

Етап 1. Знаходження первинних імплікант. Перш за все складається таблиця і знаходяться імпліканти третього і другого

рангу, тобто знижується ранг членів, які входять в ДДНФ. Складаємо наступну таблицю, яка включає усі терми, які не підлягають поглинанню на попередніх кроках, а також імпліканти 1-го рангу.

Етап 2. Розставлення міток. Складається таблиця, число рядків, якої дорівнює числу первинних імплікант, а число стовпців співпадає з числом мінтермів (конституент одиниць) ДДНФ функції f . Якщо в деякий мінтерм ДДНФ входить яка-небудь із первинних імплікант, то на перетині відповідного стовпця і рядка ставиться мітка (символ “*”).

Етап 3. Знаходження істотних імплікант. Якщо в деякому стовпці таблиці знаходиться лише одна мітка, то первинна імпліканта у відповідному рядку є істотною, оскільки без неї не буде можливості одержати усю множину заданих мінтермів.

Етап 4. Викреслювання лишніх стовпців. Після третього етапу в результаті викреслювання стовпців може одержатись таблиця, в якій є два стовпці, в яких є мітки в однакових рядках. Тоді один з них викреслюється. Покриття конституенти одиниці стовпця, який залишився буде здійснювати відкинутий мінтерм. У випадку, якщо таких рядів більше, то викреслюються всі, крім одного.

Етап 5. Викреслювання лишніх первинних імплікант. Якщо після викреслювання стовпців на етапі 4 в таблиці появляються рядки, в яких немає жодної мітки, то первинні імпліканти, які відповідають цим рядкам, вилучаються із подальшого розгляду, оскільки вони не покривають мінтерми, які залишились для розгляду.

Етап 6. Вибір мінімального покриття. В одержаній після попередніх етапів таблиці вибирається така сукупність первинних імплікант, яка включає мітки у всіх стовпцях принаймні по одній мітці в кожному стовпці. При декількох можливих варіантах

такого вибору віддається перевага варіанту покриття з мінімальним сумарним числом букв в імплікантах, які здійснюють покриття.

5.3 Мінімізація методом Квайна – Мак-Класкі

Метод Квайна – Мак-Класкі представляє собою формалізацію етапу пошуку первинних імплікант методу Квайна. Формалізація зводиться до наступного:

Етап 1. Усі мінтерми (конституенти одиниці) із ДДНФ булевої функції f записуються відповідними двійковими номерами (наборами).

Етап 2. На першому кроці усі номери розбиваються на групи по кількості одиниці у наборі.

Етап 3. Оскільки сусідні групи відрізняються між собою по кількості одиниць на одиницю, то склеювання здійснюється тільки між номерами сусідніх груп. Номери, які брали участь у склеювання якимось чином відмічаються, наприклад, закреслюються.

Етап 4. Кожний із наборів може брати участь у декількох склеюваннях, як і в методі Квайна. Невідмічені після склеювання номери являються первинними імплікантами.

Етап 5. На другому кроці результати склеювання, одержані на першому кроці, розбиваються на групи по розташуванню символу (наприклад, ”*”), яким відмічені змінні, які брали участь у склеюванні і проводиться склеювання у межах виділених груп. Знову набори, які брали участь у склеюванні відмічаються і підлягають перегрупованні по розташуванні символів ”*” для подальшого склеювання, а набори, які не брали участь в склеювання відносяться до первинних імплікант і т.д.

Етап 6. Знаходження мінімальних ДНФ здійснюється за допомогою імплікантної таблиці, як і в методі Квайна.

5.4 Мінімізація за допомогою карт Карно (карт Вейча)

Карта Вейча – це прямокутна або квадратна таблиця, число клітин в якій дорівнює 2^n – по кількості мінтермів (конституент одиниці) для логічної функції від n -змінних. Кожній клітині приписується номер певного набору значень аргументів (конституент одиниці). У самих клітинах записуються значення функції, визначені на цих наборах.

Карта Вейча для логічної функції від чотирьох змінних:

Таблиця 1.3.

	x_3		$\bar{x}_3$		
x_2	12	14	6	4	$\bar{x}_0$
x_2	13	15	7	5	x_0
$\bar{x}_2$	9	11	3	1	x_0
$\bar{x}_2$	8	10	2	0	$\bar{x}_0$
	$\bar{x}_1$	x_1		$\bar{x}_1$	

Алгоритм мінімізації за допомогою карт Вейча зводиться до наступного:

1. На карті Вейча для логічної функції від n змінних виділяють прямокутні області, які об'єднують вибрані одиничні значення функції. Кожна область повинна містити 2^k , $k = 0, 1, 2, \dots$ клітинок. Вибрані області можуть перетинатися, тобто одна або декілька клітинок можуть належати різним областям.

2. Кожній із виділених областей, яка складається із 2^k клітинок, відповідає 2^k мінтермів, які можна склеїти за k змінними, в результаті чого дістаємо елементарну кон'юнкцію рангу $n - k$, яку часто називають імплікантою.

3. Із виділеної множини областей, вибирають мінімальну кількість максимально великих областей (по кількості клітин), які охоплюють усі клітини з одиничними значеннями логічної функції.

4. Логічно сумують імпліканти, які відповідають вибраним областям. Одержана сума утворює мінімальну диз'юнктивну нормальну форму (МДНФ), тобто є мінімальним покриттям логічної функції за індексом простоти $L_1(f)$.

5.5 Мінімізація неповністю визначених функцій

Частково-визначена логічна функція від n змінних – функція, яка містить набори, на яких вона не визначена. Зустрічаються такі функції досить часто і тому питання їх мінімізації є досить актуальним. Якщо кількість невизначених наборів дорівнює m ($m < n$), то шляхом різних довизначень можна одержати 2^m різних функцій. Надалі набори, на яких функція невизначена будемо позначати символом “*”. На таких наборах значення логічної функції може бути довільним. Усе залежить від деяких інших умов. Наприклад, якщо взяти який-небудь десятковий код, то початкова функція визначена на 10 із 16 можливих наборів. Інші набори – заборонені. Разом з тим, їх треба довизначити, оскільки інакше буде неможливо використати аналітичне представлення у вигляді досконалої диз'юнктивної нормальної форми (ДДНФ). Вибір наборів, на яких треба довизначити функцію є дуже важливим, так як від нього залежить результат розв'язання поставленої задачі.

При довизначенні функції можна керуватись наступним правилом: мінімальна диз'юнктивна нормальна форма (МДНФ) не повністю визначеної функції $f(x_{n-1}, \dots, x_1, x_0)$ одержується як диз'юнкція найбільш коротких по кількості букв імплікант функції $f(x_{n-1}, \dots, x_1, x_0)$, яка приймає значення рівне одиниці на всіх наборах, де функція не визначена і які в сукупності

накривають усі одиниці (мінтерми) функції $f(x_{n-1}, \dots, x_1, x_0)$, яка приймає значення рівні нулю на всіх наборах, де функція $f(x_{n-1}, \dots, x_1, x_0)$ не визначена.

Таким чином, потрібно вибрати ті імпліканти із f_1 , які оптимально покривають мінтерми функції f_0 .

6. Синтез комбінаційних схем

Комбінаційною схемою називається логічна схема, яка реалізовує однозначну відповідність між значеннями вхідних і вихідних сигналів. Для реалізації комбінаційних схем використовуються логічні елементи, які випускаються у вигляді інтегральних схем (ІС) дешифраторів, шифраторів, мультиплексорів, демультиплексорів, суматорів та інше.

Форму запису логічних функцій у ДДНФ (ДНФ) будемо називати записом логічної функції у *канонічній нормальній формі* або форма *I/АБО*.

Форму запису логічних функцій у ДКНФ (КНФ) будемо називати записом логічної функції у *диз'юнктивній нормальній формі* або форма *АБО/I*.

Форми I/АБО і АБО/I не є одними способами задання функції алгебри логіки. Використовуючи закони подвійного заперечення $\overline{\overline{X}} = X$ і закони де Моргана $\overline{X \vee Y} = \overline{X} \wedge \overline{Y}$, або $\overline{X \wedge Y} = \overline{X} \vee \overline{Y}$, можна одержати ще шість канонічних форм. Справді, нехай маємо функцію, задану в ДНФ.

$$f(x_1, x_2, \dots, x_n) = K_1 + K_2 + \dots + K_m \quad \text{— (форма I/АБО).}$$

Використовуючи вищеназвані закони та формули, за допомогою послідовних перетворень одержимо:

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= \overline{\overline{f(x_1, x_2, \dots, x_n)}} = \overline{K_1 + K_2 + \dots + K_m} = \\ &= \overline{K_1} \cdot \overline{K_2} \cdot \dots \cdot \overline{K_m} = \quad \text{— (форма I-НЕ/I-НЕ)} \\ &= \overline{D_1} \cdot \overline{D_2} \cdot \dots \cdot \overline{D_m} = \quad \text{— (форма АБО/I-НЕ)} \\ &= \overline{D_1} + \overline{D_2} + \dots + \overline{D_m} \quad \text{— (форма АБО-НЕ/АБО).} \end{aligned}$$

У формі І/АБО можна подати як функцію f так і її заперечення $\bar{f}$, що дасть ще чотири форми:

$$\begin{aligned}
 f(x_1, x_2, \dots, x_n) &= \overline{\overline{f(x_1, x_2, \dots, x_n)}} = \overline{\bar{f}_K(x_1, x_2, \dots, x_n)} = \\
 &= \overline{K_1 + K_2 + \dots + K_s} = \quad \quad \quad - \text{(форма І/АБО-НЕ)} \\
 &= \overline{K_1} \cdot \overline{K_2} \cdot \dots \cdot \overline{K_s} = \quad \quad \quad - \text{(форма І-НЕ/І)} \\
 &= D_1 \cdot D_2 \cdot \dots \cdot D_m = \quad \quad \quad - \text{(форма АБО/І).} \\
 &= \overline{\overline{D_1 + D_2 + \dots + D_m}} = \quad \quad \quad - \text{(форма АБО-НЕ/АБО-НЕ)}
 \end{aligned}$$

7. Синтез комбінаційних схем за допомогою дешифраторів та мультиплексорів

Шифратор (кодер) – комбінаційна схема, яка здійснює перетворення сигналу на одній із вхідних шин у код номера цієї шини. Прикладом шифратора є логічна схема, яка перетворює сигнали з десяткової клавіатури $D_0, D_1, \dots, D_9$ (наприклад, мікрокалькулятора) у двійково-десятковий код $A_3 A_2 A_1 A_0$ (двійкову тетраду). Оскільки є можливість натиснення зразу декількох клавiш, то це може привести до збудження декількох вхідних шин, яке в свою чергу приведе до видачі коду, який у загальному випадку, не буде співпадати з кодом жодної із натиснутих шин. Наприклад, якщо $D_5 = D_6 = 1$, то одержимо $A_3 A_2 A_1 A_0 = 0111$, що відповідає сигналу $D_7 = 1$. Усунути вказаний недолік можна шляхом застосування пріоритетних шифраторів, які при декількох збуджених вхідних шинах формують код однієї із них, наприклад, старшої. У цьому випадку

$$\begin{aligned}
 A_3 &= D_9 \vee D_8, \\
 A_2 &= \overline{(D_9 \vee D_8)} \wedge (D_7 \vee D_6 \vee D_5 \vee D_4), \\
 A_1 &= \overline{(D_9 \vee D_8)} \wedge \overline{(D_7 \vee D_6)} \vee \overline{(D_5 \vee D_4)} \wedge (D_3 \vee D_2), \\
 A_0 &= D_9 \vee \overline{D_8} \wedge (D_7 \vee \overline{D_6}) \wedge (D_5 \vee \overline{D_4}) \wedge (D_3 \vee \overline{D_2}) \wedge D_1.
 \end{aligned}$$

Дешифратор (декодер) – логічна комбінаційна схема з n інформаційними входами і 2^n виходами, у якої певній комбінації вхідних сигналів відповідає активний стан одного із виходів. Таким чином дешифратор є деякою протилежністю демультиплексора, у якого адресні входи стали інформаційними, а інформаційний вхід став входом дозволу. Тому часто дешифратори називають дешифраторами-мультиплексорами. Як правило n дорівнює 2, 3 або 4. Якщо потрібно більше число входів, то дешифратори сполучаються в систему.

Крім інформаційних входів A , B , C дешифратори мають додаткові входи керування G . Сигнали на цих входах, наприклад, дозволяють функціонувати дешифратору або переводять його у пасивний стан, при якому, незалежно від сигналів на інформаційних входах, на всіх виходах встановлюється рівень логічної одиниці.

Вхід дозволу дешифратора може бути прямим або інверсним. У дешифраторів з прямим входом дозволу активним рівнем є рівень логічної одиниці, а у дешифраторів з інверсним входом – рівень логічного нуля.

Мультиплексор – комбінаційна логічна схема, яка являє собою керований перемикач, який під'єднує до виходу один із інформаційних входів даних. Номер під'єданого входу дорівнює числу (адресі), яке визначається комбінацією логічних значень на входах керування. Крім інформаційних входів і входів керування, схема мультиплексора містить вхід дозволу, при подачі на який активного рівня, мультиплексор переходить в пасивний стан, при якому сигнал на виході зберігає постійне значення незалежно від значення інформаційних і керуючих сигналів. Число інформаційних входів у мультиплексора, як правило, є 2, 4, 8 або 16. Якщо, наприклад, у мультиплексора 16 входів і до кожного із них під'єднано 16 джерел числових сигналів – генераторів послідовних цифрових слів, то байти із будь-якого із них можна передавати на єдиний вихід. Для вибору будь-якого із 16 каналів необхідно мати 4 входи вибору ($2^4=16$), на які подається

двійкова адреса каналу. Так, для передачі даних з каналу номер 10 на входи вибору необхідно подати код 1010.

Мультиплексори застосовуються, наприклад, у мікропроцесорах для видачі на одні і ті ж виводи адреси і даних, що дає можливість істотно скоротити загальну кількість виводів мікросхеми; у мікропроцесорних системах керування мультиплексори встановлюються на віддалених об'єктах для можливості передачі інформації по одній лінії від декількох встановлених на них датчиків.

Демультимплексори у функціональному відношенні протилежні мультиплексорам. З їх допомогою сигнали з одного інформаційного входу розподіляються у потрібній послідовності на декілька виходів. Вибір потрібного виходу, як і в мультиплексорі, забезпечується встановленням відповідного коду на адресних входах. При m адресних входах мультиплексор може мати до 2^m виходів.

II. СИНТЕЗ АВТОМАТІВ

Цифровий автомат – це пристрій, який здійснює приймання, зберігання і перетворення дискретної інформації за деяким алгоритмом.

Абстрактний цифровий автомат A визначається сукупністю п'яти об'єктів $\{X, S, Y, \varphi, \lambda\}$,

де $X = \{X_i\}, i \in \overline{1, m}$ – множина вхідних сигналів автомата A (вхідний алфавіт автомата A);

$S = \{S_j\}, j \in \overline{1, n}$ – множина станів автомата A (алфавіт станів автомата A);

$Y = \{Y_k\}, k \in \overline{1, l}$ – множина вихідних сигналів автомата A (вихідний алфавіт автомата A);

φ – функція переходів автомата A , яка відображає $(X \times S) \rightarrow S$, тобто ставить у відповідність будь-якій парі елементів добутку множин $(X \times S)$ елемент множини S ;

λ – функція виходів автомата A , яка задає відображення $(X \times S) \rightarrow Y$ або

$$S \rightarrow Y.$$

За способом формування функції виходів розрізняють наступні типи автоматів: автомат Мілі, автомат Мура (рис.1).

В абстрактному автоматі Мілі функція виходів λ задає відображення $(X \times S) \rightarrow Y$.

Автомат Мілі характеризується системою рівнянь:

$$y(t) = \lambda[s(t), x(t)];$$

$$s(t + 1) = \phi[s(t), x(t)].$$

Автомат Мура – системою рівнянь:

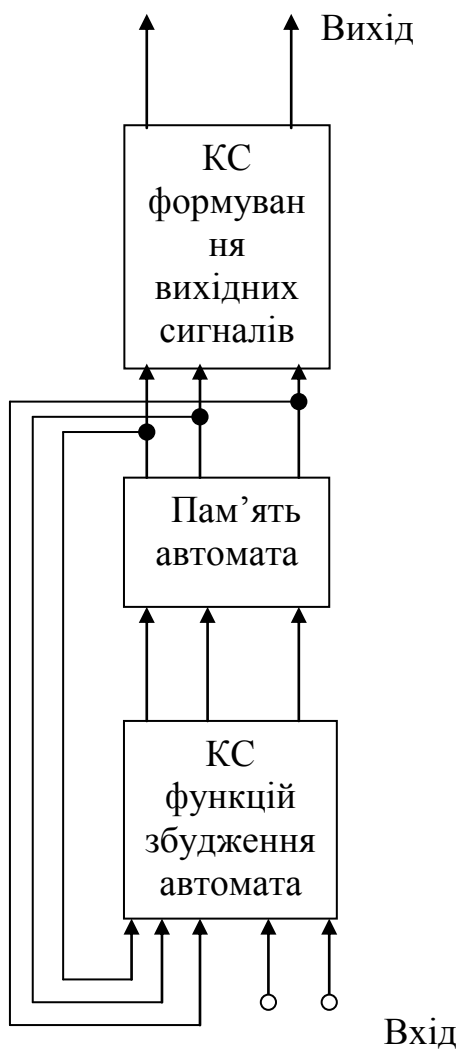
$$y(t) = \lambda[s(t)];$$

$$s(t + 1) = \phi[s(t), x(t)].$$

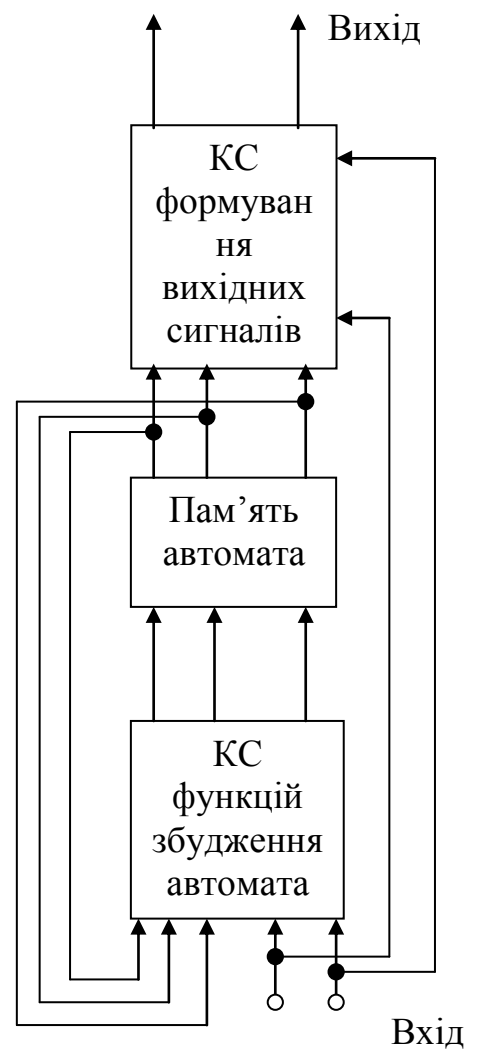
Синтез цифрових автоматів з пам'яттю можна розділити на наступні етапи:

- 1) кодування;
- 2) вибір елементів пам'яті автомата;
- 3) вибір структурно - повної системи елементів (типу автомату);
- 4) побудова рівнянь булевих функцій виходів і збудження автомата;
- 5) побудова функціональної схеми автомата.

Структурну схему автоматів зображено на рис. 1.1



Автомат Мілі



Автомат Мура

Рис. 1.1 – Структурні схеми автоматів з пам'яттю

ПРАКТИЧНА ЧАСТИНА

I. СИНТЕЗ КОМБІНАЦІЙНИХ СХЕМ

Таблиця 2.1.1 Таблиця істинності для функцій f_1, f_2, f_3, f_4

№	X_3	X_2	X_1	X_0	f_1	f_2	f_3	f_4
0	0	0	0	0	0	0	1	0
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	1	1
3	0	0	1	1	1	1	1	1
4	0	1	0	0	0	0	1	0
5	0	1	0	1	0	0	0	0
6	0	1	1	0	1	0	1	0
7	0	1	1	1	1	1	1	0
8	1	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	1
12	1	1	0	0	0	0	0	0
13	1	1	0	1	1	0	0	0
14	1	1	1	0	1	1	1	1
15	1	1	1	1	1	1	1	1

Таблиця 2.1.2 Таблиця істинності для неповністю визначеної функції

19	N	0	2	3	5	6	7	8	12	13	15
	F	0	1	1	0	0	1	0	0	1	1

1. Перевірка булевих функцій на наявність фіктивних аргументів

Перевіримо наявність фіктивних змінних у функції f_3 .

Таблиця 1.3

	x_3	x_2	x_1	x_0	f_3
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

Розглянемо змінну x_0 .

$$f_3(0,0,0,0)=1, \quad f_3(0,0,0,1)=0$$

x_0 є істотною для функції f_3 , тому що вона на протилежних наборах приймає різні значення.

Розглянемо змінну x_1 .

$$f_3(0,0,0,0)=f_3(0,0,1,0)=1$$

$$f_3(0,0,0,1)=0, \quad f_3(0,0,1,1)=1$$

x_1 є істотною для функції f_3 , тому що вона на протилежних наборах приймає різні значення.

Розглянемо змінну x_2 :

$$f_3(0, x_2, 0, 0) = f_3(0, 0, 0, 0) = f_3(0, 1, 0, 0) = 1$$

$$f_3(0, x_2, 0, 1) = f_3(0, 0, 0, 1) = f_3(0, 1, 0, 1) = 0$$

$$f_3(0, x_2, 1, 0) = f_3(0, 0, 1, 0) = f_3(0, 1, 1, 0) = 1$$

$$f_3(0, x_2, 1, 1) = f_3(0, 0, 1, 1) = f_3(0, 1, 1, 1) = 1$$

$$f_3(1, x_2, 0, 0) = f_3(1, 0, 0, 0) = f_3(1, 1, 0, 0) = 0$$

$$f_3(1, x_2, 0, 1) = f_3(1, 0, 0, 1) = f_3(1, 1, 0, 1) = 0$$

$$f_3(1, x_2, 1, 0) = f_3(1, 0, 1, 0) = f_3(1, 1, 1, 0) = 1$$

$$f_3(1, x_2, 1, 1) = f_3(1, 0, 1, 1) = f_3(1, 1, 1, 1) = 1$$

x_2 є фіктивною, тому що функція на протилежних наборах приймає однакові значення.

Розглянемо змінну x_3 :

$$f_3(0, 0, 0, 0) = 1, \quad f_3(1, 0, 0, 0) = 0$$

x_3 є істотною для функції f_3 , тому що вона на протилежних наборах приймає різні значення.

Отже, функція f_3 є виродженою, оскільки вона має один фіктивний аргумент – змінну x_2 .

Перевіримо наявність фіктивних змінних у функції f_4 .

Таблиця 2.1.4

	x_3	x_2	x_1	x_0	f_4
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

Розглянемо змінну x_0 .

$$f_4(0,0,0,x_0) = f_4(0,0,0,0) = f_4(0,0,0,1) = 0$$

$$f_4(0,0,1,x_0) = f_4(0,0,1,0) = f_4(0,0,1,1) = 1$$

$$f_4(0,1,0,x_0) = f_4(0,1,0,0) = f_4(0,1,0,1) = 0$$

$$f_4(0,1,1,x_0) = f_4(0,1,1,0) = f_4(0,1,1,1) = 0$$

$$f_4(1,0,0,x_0) = f_4(1,0,0,0) = f_4(1,0,0,1) = 0$$

$$f_4(1,0,1,x_0) = f_4(1,0,1,0) = f_4(1,0,1,1) = 1$$

$$f_4(1,1,0,x_0) = f_4(1,1,0,0) = f_4(1,1,0,1) = 0$$

$$f_4(1,1,1,x_0) = f_4(1,1,1,0) = f_4(1,1,1,1) = 1$$

x_0 є фіктивною f_4 , тому що функція на протилежних наборах приймає однакові значення.

Розглянемо змінну x_1 :

$$f_3(0,0,0,0) = 0, \quad f_3(0,0,0,1) = 1$$

x_1 є істотною для функції, тому що вона на протилежних наборах приймає різні значення.

Розглянемо змінну x_2 .

$$\begin{aligned} f_4(0, x_2, 0, 0) &= f_4(0, 0, 0, 0) = f_4(0, 1, 0, 0) = 0 \\ f_4(0, x_2, 0, 1) &= f_4(0, 0, 0, 1) = f_4(0, 1, 0, 1) = 0 \\ f_4(0, x_2, 1, 0) &= f_4(0, 0, 1, 0) = 1, \quad f_4(0, 1, 1, 0) = 0 \end{aligned}$$

x_2 є істотною для функції f_4 , тому що вона на протилежних наборах приймає різні значення.

Розглянемо змінну x_3 .

$$\begin{aligned} f_4(x_3, 0, 0, 0) &= f_4(0, 0, 0, 0) = f_4(1, 0, 0, 0) = 0 \\ f_4(x_3, 0, 0, 1) &= f_4(0, 0, 0, 1) = f_4(1, 0, 0, 1) = 0 \\ f_4(x_3, 0, 1, 0) &= f_4(0, 0, 1, 0) = f_4(1, 0, 1, 0) = 1 \\ f_4(x_3, 0, 1, 1) &= f_4(0, 0, 1, 1) = f_4(1, 0, 1, 1) = 1 \\ f_4(x_3, 1, 0, 0) &= f_4(0, 1, 0, 0) = f_4(1, 1, 0, 0) = 0 \\ f_4(x_3, 1, 0, 1) &= f_4(0, 1, 0, 1) = f_4(1, 1, 0, 1) = 0 \\ f_4(x_3, 1, 1, 0) &= f_4(0, 1, 1, 0) = 0, \quad f_4(1, 1, 1, 0) = 1 \end{aligned}$$

x_3 є істотною для функції f_4 , тому що вона на протилежних наборах приймає різні значення.

Отже, функція f_4 є виродженою, оскільки вона має один фіктивний аргумент – змінну x_0 .

2. Представлення булевих функцій за допомогою канонічних форм.

2.1 Представити функцію f_1 у ДДНФ.

Таблиця 2.2.1

	x_3	x_2	x_1	x_0	f_1
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

Запишемо ДДНФ у вигляді логічної суми логічних добутків значень функції на відповідні мінтерми, які записуються у вигляді елементарних кон'юнкцій, утворених змінними у прямій або інверсній формі: якщо значення змінної 1, то пряма форма, а якщо 0, то – інверсна.

$$f_1 = \bar{x}_3 \bar{x}_2 x_1 x_0 + \bar{x}_3 x_2 x_1 \bar{x}_0 + \bar{x}_3 x_2 x_1 x_0 + x_3 \bar{x}_2 x_1 \bar{x}_0 + x_3 \bar{x}_2 x_1 x_0 + x_3 x_2 \bar{x}_1 x_0 + x_3 x_2 x_1 \bar{x}_0 + x_3 x_2 x_1 x_0$$

2.2 Представити функцію f_2 у ДКНФ.

Таблиця 2.2.2

	x_3	x_2	x_1	x_0	f_2
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

Записавши для заданої функції f_2 диз'юнкцію всіх конститuent нуля, отримаємо ДКНФ функції f_2 :

$$f_2 = D_0^0 \cdot D_1^0 \cdot D_3^0 \cdot D_4^0 \cdot D_5^0 \cdot D_8^0 \cdot D_9^0 \cdot D_{13}^0 = (\bar{x}_3 + \bar{x}_2 + \bar{x}_1 + \bar{x}_0) \cdot (\bar{x}_3 + x_2 + \bar{x}_1 + \bar{x}_0) \cdot (\bar{x}_3 + x_2 + \bar{x}_1 + x_0) \cdot (\bar{x}_3 + x_2 + x_1 + \bar{x}_0) \cdot (x_3 + \bar{x}_2 + \bar{x}_1 + \bar{x}_0) \cdot (x_3 + \bar{x}_2 + \bar{x}_1 + x_0) \cdot (x_3 + x_2 + \bar{x}_1 + \bar{x}_0) \cdot (x_3 + x_2 + \bar{x}_1 + x_0).$$

2.3 Представити функцію f_3 у вигляді поліному Жегалкіна.

Канонічний поліном Жегалкіна – це поліном вигляду

$$f(x_{n-1}, \dots, x_1, x_0) = a_0 \oplus a_1 x_0 \oplus a_2 x_1 \oplus \dots \oplus a_n x_{n-1} \oplus a_{12} x_0 x_1 \oplus a_{13} x_0 x_1 \oplus \dots \oplus a_{01 \dots n-1} x_0 x_1 \dots x_{n-1},$$

де $a_i = \{0,1\}$.

Для функції від чотирьох змінних в загальному випадку поліном Жегалкіна має такий вигляд:

$$f = a_0 \oplus a_1 x_0 \oplus a_2 x_1 \oplus a_3 x_2 \oplus a_4 x_3 \oplus a_5 x_0 x_1 \oplus a_6 x_0 x_2 \oplus a_7 x_0 x_3 \oplus a_8 x_1 x_2 \oplus a_9 x_1 x_3 \oplus a_{10} x_2 x_3 \oplus a_{11} x_0 x_1 x_2 \oplus a_{12} x_0 x_1 x_3 \oplus a_{13} x_0 x_2 x_3 \oplus a_{14} x_1 x_2 x_3 \oplus a_{15} x_0 x_1 x_2 x_3.$$

Таблиця 2.2.3

	x_3	x_2	x_1	x_0	f_3
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

Щоб записати функцію f_3 у вигляді поліному Жегалкіна, знайдемо коефіцієнти a_i :

при $x_0 = x_1 = x_2 = x_3 = 0$ маємо $a_0 = 1$;

при $x_0 = 1; x_1 = x_2 = x_3 = 0$ $0 = 1 \oplus a_1 \Rightarrow a_1 = 1$;

при $x_1 = 1; x_0 = x_2 = x_3 = 0$ $1 = 1 \oplus 1 \oplus a_2 \Rightarrow a_2 = 1$;

при $x_2 = 1; x_0 = x_1 = x_3 = 0$ $1 = 1 \oplus 1 \oplus 1 \oplus a_3 \Rightarrow a_3 = 0$;

при $x_3 = 1; x_0 = x_1 = x_2 = 0$ $1 = 1 \oplus 1 \oplus 1 \oplus a_4 \Rightarrow a_4 = 0$;

при $x_0 = x_1 = 1; x_2 = x_3 = 0$ $0 = 1 \oplus 1 \oplus 1 \oplus a_5 \Rightarrow a_5 = 1$;

при $x_0 = x_2 = 1; x_1 = x_3 = 0$ $1 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus a_6 \Rightarrow a_6 = 1$;

при $x_0 = x_3 = 1; x_1 = x_2 = 0$ $1 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus a_7 \Rightarrow a_7 = 0$;

при $x_1 = x_2 = 1; x_0 = x_3 = 0$ $0 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus a_8 \Rightarrow a_8 = 1$;

при $x_1 = x_3 = 1; x_0 = x_2 = 0$ $0 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus a_9 \Rightarrow a_9 = 0$;

при $x_2 = x_3 = 1; x_0 = x_1 = 0$ $1 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus a_{10} \Rightarrow a_{10} = 1$;

при $x_0 = x_1 = x_2 = 1; x_3 = 0$ $1 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus a_{11} \Rightarrow a_{11} = 0$;

при $x_0 = x_1 = x_3 = 1; x_2 = 0$ $0 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus a_{12} \Rightarrow a_{12} = 1$;

при $x_0 = x_2 = x_3 = 1; x_1 = 0$ $0 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus a_{13} \Rightarrow a_{13} = 0$;

при $x_1 = x_2 = x_3 = 1; x_0 = 0$ $1 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus a_{14} \Rightarrow a_{14} = 1$;

при $x_0 = x_1 = x_2 = x_3 = 1$ $1 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus a_{15} \Rightarrow a_{15} = 0$.

Підставивши отримані коефіцієнти у вираз поліному Жегалкіна для функції від чотирьох змінних, отримаємо вираз для функції f_3

$$f_3 = 1 \oplus x_0 \oplus x_1 \oplus x_0 x_1 \oplus x_0 x_2 \oplus x_1 x_2 \oplus x_2 x_3 \oplus x_0 x_1 x_3 \oplus x_1 x_2 x_3$$

3. Розклад Шеннона, представлення функції f_4 з допомогою функцій від двох змінних.

Використовуючи розклад Шеннона, представити функцію f_4 за допомогою функцію від двох змінних.

$$\begin{aligned} f(x_3, x_2, x_1, x_0) &= f(0,0,0,0)\bar{x}_3\bar{x}_2\bar{x}_1\bar{x}_0 + f(0,0,0,1)\bar{x}_3\bar{x}_2\bar{x}_1x_0 + f(0,0,1,0)\bar{x}_3\bar{x}_2x_1\bar{x}_0 + f(0,0,1,1)\bar{x}_3\bar{x}_2x_1x_0 + \\ &+ f(0,1,0,0)\bar{x}_3x_2\bar{x}_1\bar{x}_0 + f(0,1,0,1)\bar{x}_3x_2\bar{x}_1x_0 + f(0,1,1,0)\bar{x}_3x_2x_1\bar{x}_0 + f(0,1,1,1)\bar{x}_3x_2x_1x_0 + \\ &+ f(1,0,0,0)x_3\bar{x}_2\bar{x}_1\bar{x}_0 + f(1,0,0,1)x_3\bar{x}_2\bar{x}_1x_0 + f(1,0,1,0)x_3\bar{x}_2x_1\bar{x}_0 + f(1,0,1,1)x_3\bar{x}_2x_1x_0 + \\ &+ f(1,1,0,0)x_3x_2\bar{x}_1\bar{x}_0 + f(1,1,0,1)x_3x_2\bar{x}_1x_0 + f(1,1,1,0)x_3x_2x_1\bar{x}_0 + f(1,1,1,1)x_3x_2x_1x_0 = \\ &= \bar{x}_3(\bar{x}_2(f(0,0,0,0)\bar{x}_1\bar{x}_0 + f(0,0,0,1)\bar{x}_1x_0 + f(0,0,1,0)x_1\bar{x}_0 + f(0,0,1,1)x_1x_0) + \\ &+ x_2(f(0,1,0,0)\bar{x}_1\bar{x}_0 + f(0,1,0,1)\bar{x}_1x_0 + f(0,1,1,0)x_1\bar{x}_0 + f(0,1,1,1)x_1x_0)) + \\ &+ x_3(\bar{x}_2(f(1,0,0,0)\bar{x}_1\bar{x}_0 + f(1,0,0,1)\bar{x}_1x_0 + f(1,0,1,0)x_1\bar{x}_0 + f(1,0,1,1)x_1x_0) + \\ &+ x_2(f(1,1,0,0)\bar{x}_1\bar{x}_0 + f(1,1,0,1)\bar{x}_1x_0 + f(1,1,1,0)x_1\bar{x}_0 + f(1,1,1,1)x_1x_0)). \end{aligned}$$

Підставивши значення функції на наборах в одержану формулу, отримаємо:

$$f(x_3, x_2, x_1, x_0) = \bar{x}_3(\bar{x}_2(f(0,0,1,0)x_1\bar{x}_0 + f(0,0,1,1)x_1x_0)) + x_3(\bar{x}_2(f(1,0,1,0)x_1\bar{x}_0 + f(1,0,1,1)x_1x_0) + x_2(f(1,1,1,0)x_1\bar{x}_0 + f(1,1,1,1)x_1x_0)).$$

4. Визначення належності логічних функцій до п'яти предповних класів.

Функціонально повною системою булевих функцій (базисом) називається сукупність таких булевих функцій $\{f_1, f_2, \dots, f_m\}$, що будь-яка логічна функція може бути виражена через функції цієї сукупності за допомогою принципу суперпозиції.

Таблиця 2.4.1

	зберігає 0	зберігає 1	самодвоїста	лінійна	монотонна
f_1	+	+	+	-	-
f_2	+	+	-	-	-
f_3	-	+	-	-	-
f_4	+	+	-	-	-

Проаналізуємо належність функцій f_1, f_2, f_3, f_4 до п'яти предповних класів.

Перевіримо чи зберігають функції 0:

$$f_1(0,0,0,0) = 0$$

$$f_2(0,0,0,0) = 0$$

$$f_3(0,0,0,0) = 1$$

$$f_4(0,0,0,0) = 0$$

Функції, які на нульовому наборі дорівнюють нулю «зберігають нуль».

Перевіримо чи зберігають функції 1:

$$f_1(1,1,1,1) = 1$$

$$f_2(1,1,1,1) = 1$$

$$f_3(1,1,1,1) = 1$$

$$f_4(1,1,1,1) = 1$$

Функція, які на одиничному наборі дорівнюють одиниці «зберігає одиницю».

Перевірка функції на самодвоїстість:

Логічна функція є самодвоїстою, якщо на кожній парі протилежних наборів вона приймає протилежні значення, тобто

$$f(x_{n-1}, \dots, x_2, x_1, x_0) = \overline{f(\bar{x}_{n-1}, \dots, \bar{x}_1, \bar{x}_0)}.$$

Для f_1 :

$$\begin{aligned} f_1(0,0,0,0) &= 0, a & f_1(1,1,1,1) &= 1 \\ f_1(0,0,0,1) &= 0, a & f_1(1,1,1,0) &= 1 \\ f_1(0,0,1,0) &= 0, a & f_1(1,1,0,1) &= 1 \\ f_1(0,0,1,1) &= 1, a & f_1(1,1,0,0) &= 0 \\ f_1(0,1,0,0) &= 0, a & f_1(1,0,1,1) &= 1 \\ f_1(0,1,0,1) &= 0, a & f_1(1,0,1,0) &= 1 \\ f_1(0,1,1,0) &= 1, a & f_1(1,0,0,1) &= 0 \\ f_1(0,1,1,1) &= 1, a & f_1(1,0,0,0) &= 0 \end{aligned}$$

Отже функція f_1 самодвоїста.

Для f_2 :

$$\begin{aligned} f_2(0,0,0,0) &= 0, a & f_2(1,1,1,1) &= 1 \\ f_2(0,0,0,1) &= f_2(1,1,1,0) &= 1 \end{aligned}$$

Отже функція f_2 не є самодвоїстою.

Для f_3 :

$$f_3(0,0,0,0) = f_3(1,1,1,1) = 1$$

Отже функція f_3 не є самодвоїстою.

Для f_4 :

$$\begin{aligned} f_4(0,0,0,0) &= 0, a & f_4(1,1,1,1) &= 1 \\ f_4(0,0,0,1) &= 0, a & f_4(1,1,1,0) &= 1 \\ f_4(0,0,1,0) &= 0, a & f_4(1,1,0,1) &= 1 \\ f_4(0,0,1,1) &= 0, a & f_4(1,1,0,0) &= 1 \\ f_4(0,1,0,0) &= 0, a & f_4(1,0,1,1) &= 1 \\ f_4(0,1,0,1) &= 1, a & f_4(1,0,1,0) &= 0 \\ f_4(0,1,1,0) &= f_4(1,0,0,1) &= 0 \end{aligned}$$

Отже функція f_4 не є самодвоїста.

Перевірка на лінійність:

Логічна функція називається лінійною, якщо її можна подати у вигляді полінома Жегалкіна першого степеня.

Для f_1 :

при $x_0 = x_1 = x_2 = x_3 = 0$ маємо $a_0 = 0$;

при $x_0 = 1; x_1 = x_2 = x_3 = 0$, $a_1 = 0$;

при $x_1 = 1; x_0 = x_2 = x_3 = 0$, $a_2 = 0$;

при $x_2 = 1; x_0 = x_1 = x_3 = 0$, $a_3 = 1$;

при $x_3 = 1; x_0 = x_1 = x_2 = 0$, $0 = 1 \oplus a_4 \Rightarrow a_4 = 1$;

при $x_0 = x_1 = 1; x_2 = x_3 = 0$ $0 = 1 \oplus 1 \oplus a_5 \Rightarrow a_5 = 0$;

при $x_0 = x_2 = 1; x_1 = x_3 = 0$ $1 = 1 \oplus 1 \oplus a_6 \Rightarrow a_6 = 1$;

$f_1 = x_2 \oplus x_3 \oplus x_0 x_2 \dots$ функція f_1 не є лінійною.

Для f_2 :

при $x_0 = x_1 = x_2 = x_3 = 0$ маємо $a_0 = 0$;

при $x_0 = 1; x_1 = x_2 = x_3 = 0$, $a_1 = 1$;

при $x_1 = 1; x_0 = x_2 = x_3 = 0$, $1 = 1 \oplus a_2 \Rightarrow a_2 = 0$;

при $x_2 = 1; x_0 = x_1 = x_3 = 0$, $1 = 1 \oplus a_3 \Rightarrow a_3 = 0$;

при $x_3 = 1; x_0 = x_1 = x_2 = 0$ $0 = 1 \oplus a_4 \Rightarrow a_4 = 1$;

при $x_0 = x_1 = 1; x_2 = x_3 = 0$ $0 = 1 \oplus 1 \oplus a_5 \Rightarrow a_5 = 0$;

при $x_0 = x_2 = 1; x_1 = x_3 = 0$ $0 = 1 \oplus 1 \oplus a_6 \Rightarrow a_6 = 0$;

при $x_0 = x_3 = 1; x_1 = x_2 = 0$ $1 = 1 \oplus 1 \oplus a_7 \Rightarrow a_7 = 1$;

$f_2 = x_0 \oplus x_3 \oplus x_0 x_3 \dots$ функція f_2 не є лінійною.

Для f_3 :

при $x_0 = x_1 = x_2 = x_3 = 0$ маємо $a_0 = 1$;

при $x_0 = 1; x_1 = x_2 = x_3 = 0$ $0 = 1 \oplus a_1 \Rightarrow a_1 = 1$;

при $x_1 = 1; x_0 = x_2 = x_3 = 0$ $1 = 1 \oplus 1 \oplus a_2 \Rightarrow a_2 = 1$;

при $x_2 = 1; x_0 = x_1 = x_3 = 0$ $1 = 1 \oplus 1 \oplus 1 \oplus a_3 \Rightarrow a_3 = 0$;

при $x_3 = 1; x_0 = x_1 = x_2 = 0$ $1 = 1 \oplus 1 \oplus 1 \oplus a_4 \Rightarrow a_4 = 0$;

при $x_0 = x_1 = 1; x_2 = x_3 = 0$ $0 = 1 \oplus 1 \oplus 1 \oplus a_5 \Rightarrow a_5 = 1$;

$f_3 = 1 \oplus x_0 \oplus x_1 \oplus x_0 x_1$ функція f_3 не є лінійною.

Для f_4 :

при $x_0 = x_1 = x_2 = x_3 = 0$ маємо $a_0 = 0$;

при $x_0 = 1; x_1 = x_2 = x_3 = 0$, $a_1 = 0$;

при $x_1 = 1; x_0 = x_2 = x_3 = 0$, $a_2 = 1$;

при $x_2 = 1; x_0 = x_1 = x_3 = 0$, $1 = 1 \oplus a_3 \Rightarrow a_3 = 0$;

при $x_3 = 1; x_0 = x_1 = x_2 = 0$, $0 = 1 \oplus a_4 \Rightarrow a_4 = 1$;

при $x_0 = x_1 = 1; x_2 = x_3 = 0$, $0 = 1 \oplus 1 \oplus a_5 \Rightarrow a_5 = 0$;

при $x_0 = x_1 = 1; x_2 = x_3 = 0$, $0 = 1 \oplus 1 \oplus a_6 \Rightarrow a_6 = 0$;

при $x_0 = x_1 = 1; x_2 = x_3 = 0$, $0 = 1 \oplus 1 \oplus a_7 \Rightarrow a_7 = 0$;

при $x_0 = x_1 = 1; x_2 = x_3 = 0$, $0 = 1 \oplus 1 \oplus a_8 \Rightarrow a_8 = 0$;

при $x_0 = x_1 = 1; x_2 = x_3 = 0$, $0 = 1 \oplus 1 \oplus a_9 \Rightarrow a_9 = 0$;

при $x_0 = x_1 = 1; x_2 = x_3 = 0$, $1 = 1 \oplus 1 \oplus a_{10} \Rightarrow a_{10} = 1$;

$f_4 = x_1 \oplus x_3 \oplus x_0 x_1 \dots$ функція f_4 не є лінійною.

Перевірка функції на монотонність:

Логічна функція називається монотонною, якщо для будь-яких двох двійкових наборів α і β , що задовольняють умову $\alpha \leq \beta$, справджується нерівність $f(\alpha) \leq f(\beta)$.

Для f_1 :

$$f_1(0,0,1,1) > f_1(0,1,0,0)$$

$$1 > 0$$

Для f_2 :

$$f_2(0,0,1,1) > f_2(0,1,0,0)$$

$$1 > 0$$

Для f_3 :

$$f_3(0,1,0,0) > f_3(0,1,0,1)$$

$$1 > 0$$

Для f_4 :

$$f_4(0,0,1,1) > f_4(0,1,0,0)$$

$$1 > 0$$

5. Мінімізація булевих функцій.

В основу мінімізації за допомогою К-карт покладено операцію склеювання вмісту *сусідніх* або *відповідних* клітин. К-карта для функції від чотирьох змінних має вигляд

Таблиця 2.5.1

0000	0001	0101	0100
0010	0011	0111	0110
1010	1011	1111	1110
1000	1001	1101	1100

5.1 Виконати мінімізацію функцій f_1, f_2, f_3, f_4 методом К-карт.

К-карта функції f_1 має наступний вигляд:

Таблиця 2.5.2

	0011	0111	0110
1010	1011	1111	1110
		1101	

Проведені склейки подані в наступній таблиці:

Таблиця 2.5.3

1111	1010	1111	1101
1110	1011	0111	1111
0111	1111	0011	
0110	1110	1011	
11	1*1*	**11	11*1

Мінімальна форма функції.

$$f_{1\min} = *11* + 1*1* + **11 + 11*1 = bc + ac + cd + abd$$

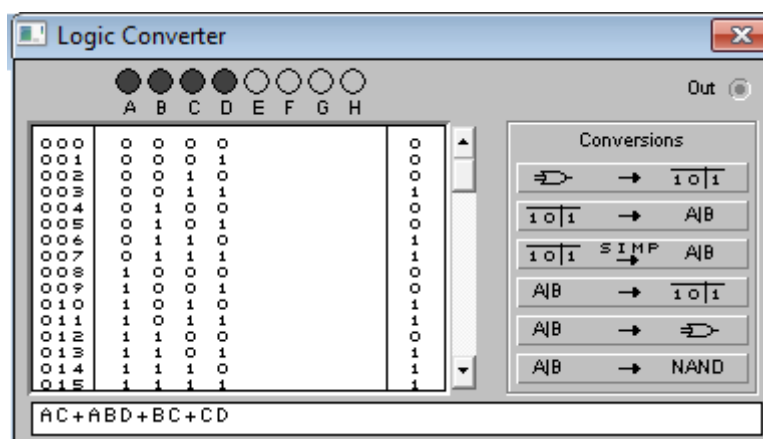


Рис. 2.5.1

К-карта функції f_2 має наступний вигляд:

Таблиця 2.5.4

	0001		
0010	0011	0111	
1010	1011	1111	1110

Проведені склейки подані в наступній таблиці:

Таблиця 2.5.5

0010	0011	1010	0001
1010	1011	1011	0011
1011	1111	1111	
0011	0111	1110	
01	**11	1*1*	00*1

Мінімальна форма функції.

$$F_{2\min} = *01* + **11 + 1*1* + 00*1 = \bar{b}c + cd + ac + \bar{a}\bar{b}d$$

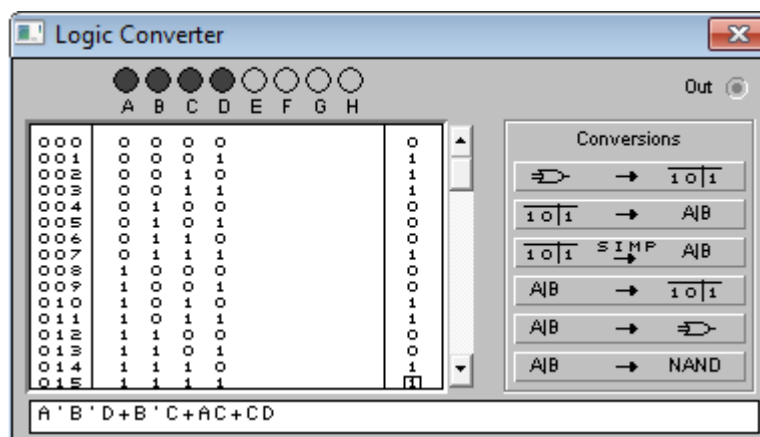


Рис. 2.5.2

К-карта функції f_3 має наступний вигляд:

Таблиця 2.5.6

0000			0100
0010	0011	0111	0110
1010	1011	1111	1110

Проведені склейки подані в наступній таблиці:

Таблиця 2.5.7

0000	0010
0010	1010
0110	1011
0100	1111
	1110
	0110
	0111
	0011
0**0	**1*

Мінімальна форма функції.

$$f_{3\min} = 0**0 + **1* = \overline{a}\overline{d} + c$$

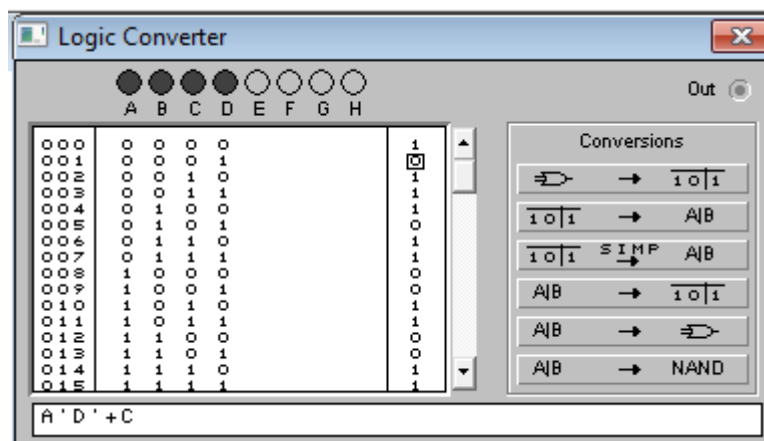


Рис. 2.5.3

К-карта функції f_4 має наступний вигляд:

Таблиця 2.5.8

0010	0011		
1010	1011	1111	1110

Проведені склейки подані в наступній таблиці:

Таблиця 2.5.9

0010	1010
0011	1011
1011	1111
1010	1110
01	1*1*

Мінімальна форма функції.

$$f_{4\min} = *01* + 1*1* = \bar{b}c + ac$$

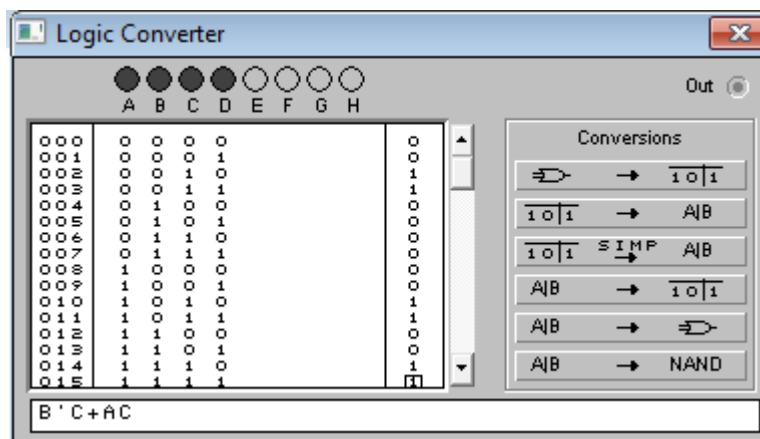


Рис. 2.5.4

5.2 Виконати мінімізацію функцій f_1 методом Квайна.

$$f_1(0011, 0110, 0111, 1010, 1011, 1101, 1110, 1111) = 1$$

Запишемо для функції f_1 імплікантну матрицю:

Таблиця 2.5.10

П.І.	$\bar{a}\bar{b}cd$	$\bar{a}b\bar{c}\bar{d}$	$\bar{a}bcd$	$a\bar{b}\bar{c}\bar{d}$	$a\bar{b}cd$	$ab\bar{c}\bar{d}$	$abcd$	$abcd$
$\bar{a}\bar{b}cd$	1		$\bar{a}cd$		$\bar{b}cd$			
$\bar{a}b\bar{c}\bar{d}$		1	$\bar{a}bc$				$b\bar{c}\bar{d}$	
$\bar{a}bcd$	*	*	1					bcd
$a\bar{b}\bar{c}\bar{d}$				1	$\bar{a}bc$		$a\bar{c}\bar{d}$	
$a\bar{b}cd$	*			*	1			acd
$ab\bar{c}\bar{d}$						1		abd
$abcd$		*		*			1	abc
$abcd$			*		*	*	*	1

Тоді функцію f_I можна представити у вигляді:

Таблиця 2.5.11

П.І.	$\bar{a}cd$	$\bar{b}cd$	$\bar{a}bc$	$b\bar{c}\bar{d}$	bcd	$\bar{a}\bar{b}c$	$a\bar{c}\bar{d}$	acd	abd	abc
$\bar{a}cd$	1							cd		
$\bar{b}cd$		1			cd					
$\bar{a}bc$			1							bc
$b\bar{c}\bar{d}$				1	bc					
bcd		*		*	1					
$\bar{a}\bar{b}c$						1				ac
$a\bar{c}\bar{d}$							1	ac		
acd	*						*	1		
abd									1	
abc			*			*				1

Таблиця 2.5.12

П.І.	$\bar{a}\bar{b}cd$	$\bar{a}b\bar{c}d$	$\bar{a}bcd$	$a\bar{b}\bar{c}d$	$a\bar{b}cd$	$ab\bar{c}d$	$abcd$	$abcd$
ac				*	*		*	*
bc		*	*				*	*
cd	*		*		*			*
abd						*		*

З імплікантної матриці можна бачити, що імпліканти $\bar{x}_2\bar{x}_1$, $\bar{x}_2x_0$, $\bar{x}_1x_0$ та $x_3\bar{x}_2$ накривають всі стовпці матриці.

Отже, функції f_1 можна записати як:

$$F_{1\min} = ac + bc + cd + abd$$

5.3 Виконати мінімізацію функцій f_2 методом Квайна – Мак-Класкі.

$$f_2 (0001, 0010, 0011, 0111, 1010, 1011, 1110, 1111) = 1$$

Запишемо номери наборів, на яких функція f_2 дорівнює одиниці у вигляді двійкових наборів (мінтермів).

Утворимо групи по кількості одиниць в наборі

Таблиця 2.5.13

Номер групи	Двійкові номери конститuent одиниці
0	-
1	0001, 0010
2	0011, 1010
3	0111, 1011, 1110
4	1111

Таблиця 2.5.14

Номери груп	Результати склеювання (двійкові номери імплікант)
0–1	
1–2	00*1, 001*, *010
2–3	0*11, *011, 101*, 1*10
3–4	*111, 1*11, 111*

Склеюємо набори із сусідніх груп табл. 2.5.13. Результати склеювання наведено в табл. 2.5.14. Закреслюємо набори в табл. 2.5.13, які брали участь в склеюванні.

Згрупуємо імпліканти по розташуванню зірочки в однакових позиціях (таблиця 2.5.15) і виконуємо склеювання в утворених групах. Результати склеювання зведено в табл. 2.5.16

Таблиця 2.5.15

Номери груп	Набори груп
1	*010 , *011 , *111
2	0*11 , 1*10 , 1*11
3	00*1
4	001* , 101* , 111*

Таблиця 2.5. 16

01, **11
**11, 1*1*
01, 1*1*

Таблиця 2.5.17

Двійкові набори простих імплікант	Двійкові набори конститuent одиниці							
	0001	0010	0011	0111	1010	1011	1110	1111
01		*	*		*	*		
**11			*	*		*		*
1*1*					*	*	*	*
00*1	*		*					

За імплікантою матрицею визначаємо сукупність двійкових наборів простих імплікант, які відповідають мінімальній ДНФ.

За допомогою двійкових наборів простих імплікант записуємо мінімальну ДНФ у буквенному вигляді:

$$F_{2\min} = *01* + **11 + 1*1* + 00*1 = \bar{b}c + cd + ac + \bar{a}\bar{b}d$$

5.4 Мінімізація за допомогою діаграм Вейча

Карта Вейча для логічної функції від чотирьох змінних:

Таблиця 2.5.18

	x_3		$\bar{x}_3$		
x_2	12	14	6	4	$\bar{x}_0$
x_2	13	15	7	5	x_0
$\bar{x}_2$	9	11	3	1	x_0
$\bar{x}_2$	8	10	2	0	$\bar{x}_0$
	$\bar{x}_1$	x_1		$\bar{x}_1$	

Функція f_1 :

Таблиця 2.5.19

	x_3		$\bar{x}_3$		
x_2	0	1	1	0	$\bar{x}_0$
x_2	1	1	1	0	x_0
$\bar{x}_2$	0	1	1	0	x_0
$\bar{x}_2$	0	1	0	0	$\bar{x}_0$
	$\bar{x}_1$	x_1		$\bar{x}_1$	

Проведені склейки подані в наступній таблиці:

Таблиця 2.5.20

1111	1010	1111	1101
1110	1011	0111	1111
0111	1111	0011	
0110	1110	1011	
11	1*1*	**11	11*1

$$f_{1\min} = *11* + 1*1* + **11 + 11*1 = bc + ac + cd + abd$$

Функція f_2 :

Таблиця 2.5.21

	x_3		$\bar{x}_3$		
x_2	0	1	0	0	$\bar{x}_0$
x_2	0	1	1	0	x_0
$\bar{x}_2$	0	1	1	1	x_0
$\bar{x}_2$	0	1	1	0	$\bar{x}_0$
	$\bar{x}_1$	x_1		$\bar{x}_1$	

Проведені склейки подані в наступній таблиці:

Таблиця 2.5.22

0010	0011	1010	0001
1010	1011	1011	0011
1011	1111	1111	
0011	0111	1110	
01	**11	1*1*	00*1

$$F_{2\min} = *01* + **11 + 1*1* + 00*1 = \bar{b}c + cd + ac + \bar{a}\bar{b}d$$

Функція f_3 :

Таблиця 2.5.23

	x_3		$\bar{x}_3$		
x_2	0	1	1	1	$\bar{x}_0$
x_2	0	1	1	0	x_0
$\bar{x}_2$	0	1	1	0	x_0
$\bar{x}_2$	0	1	1	1	$\bar{x}_0$
	$\bar{x}_1$	x_1		$\bar{x}_1$	

Проведені склейки подані в наступній таблиці:

Таблиця 2.5.24

0000	0010
0010	1010
0110	1011
0100	1111
	1110
	0110
	0111
	0011
0**0	**1*

$$f_{3\min} = 0**0 + **1* = \overline{a}\overline{d} + c$$

Функція f_4 :

Таблиця 2.5.25

	x_3		$\bar{x}_3$		
x_2	0	1	0	0	$\bar{x}_0$
x_2	0	1	0	0	x_0
$\bar{x}_2$	0	1	1	0	x_0
$\bar{x}_2$	0	1	1	0	$\bar{x}_0$
	$\bar{x}_1$	x_1		$\bar{x}_1$	

Проведені склейки подані в наступній таблиці:

Таблиця 2.5.26

0010	1010
0011	1011
1011	1111
1010	1110
01	1*1*

$$f_{4\min} = *01* + 1*1* = \bar{b}c + ac$$

5.5 Мінімізація неповністю визначених функцій

Розробити логічні схеми для реалізації частково визначених логічних функцій F від 4-х аргументів, заданих таблицями істинності. Кожна комбінація значень аргументів двійкових змінних $ABCD$ відображається числом N , яке дорівнює: $2^3A+2^2B+2^1C+2^0D$. Значення функцій при не вказаних комбінаціях значень аргументів необхідно довизначити для одержання схеми з мінімальним числом елементів. Мінімізацію логічної функції проводити за допомогою К-карт із перевіркою правильності мінімізації за допомогою логічного перетворювача пакету Electronics Workbench. Розробку схем провести на базі таких типів елементів і схем:

- Елементи І, АБО, НЕ;
- Елементи І-НЕ;
- Елементи АБО-НЕ.

Функція f_5 задана наступною таблицею (5.27):

Таблиця 2.5.27 Функція

19	N	0	2	3	5	6	7	8	12	13	15
	F	0	1	1	0	0	1	0	0	1	1

Таблиця 2.5.28 К-карта

0000	0001	0101	0100
0010	0011	0111	0110
1010	1011	1111	1110
1000	1001	1101	1100

Таблиця 2.5.29

N	A	B	C	D	F
0	0	0	0	0	0
2	0	0	1	0	1
3	0	0	1	1	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
12	1	1	0	0	0
13	1	1	0	1	1
15	1	1	1	1	1

К-карта для даної функції матиме наступний вигляд (таблиця 2.5.30).

Таблиця 2.5.30

0	?	0	?
0010	0011	0111	0
?	?	1111	?
0	?	1101	0

Після довизначення отримаємо наступну карту (функція на підкреслених наборах невизначена)

Таблиця 2.5.31

0	0001	0	0100
0010	0011	0111	0
1010	1011	1111	1110
0	1001	1101	0

Проведені склейки подані в наступній таблиці:

Таблиця 2.5.32 Склеювання

0010	1011	1011	1010	0001
1010	0011	1001	1011	0011
1011	0111	1101	1111	1011
0011	1111	1111	1110	1001
01	**11	1**1	1*1*	*0*1

Таблиця 2.5.33

П.І.	$\bar{a}\bar{b}\bar{c}d$	$\bar{a}\bar{b}c\bar{d}$	$\bar{a}b\bar{c}d$	$\bar{a}b\bar{c}\bar{d}$	$\bar{a}bcd$	$a\bar{b}\bar{c}d$	$a\bar{b}c\bar{d}$	$a\bar{b}cd$	$ab\bar{c}d$	$abc\bar{d}$	$abcd$
ac							*	*		*	*
ad						*		*	*		*
$\bar{b}c$					*					*	*
$\bar{b}d$	*		*			*		*			
cd			*		*			*			*
$\bar{a}b\bar{c}\bar{d}$				*							

$$f_{\min} = ac + ad + \bar{b}c + \bar{b}d + cd + \bar{a}b\bar{c}\bar{d}$$

Нижче наведено перші чотири базові мінімальні форми знайденої мінімальної функції:

$$1) f_{\min} = ac + ad + \bar{b}c + \bar{b}d + cd + \bar{a}b\bar{c}\bar{d} \quad - \text{I/АБО/НЕ}$$

$$2) f_{\min} = \overline{\overline{ac} \cdot \overline{ad} \cdot \overline{\bar{b}c} \cdot \overline{\bar{b}d} \cdot \overline{cd} \cdot \overline{\bar{a}b\bar{c}\bar{d}}} \quad - \text{I-НЕ / I-НЕ}$$

$$3) f_{\min} = \overline{\overline{\overline{a+c+a+d+b+c+c+d+b+d+a+b+c+d}}} \quad - \text{АБО-НЕ / АБО-НЕ}$$

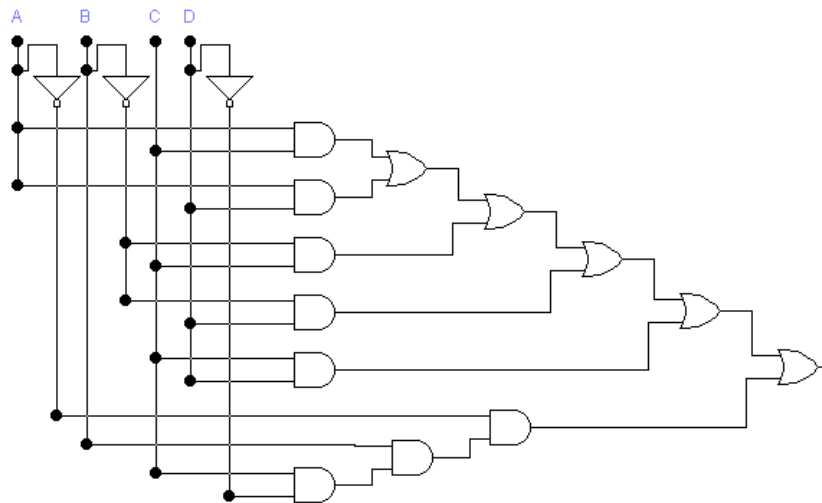


Рис. 2.5.5 Схема в базисі I, НЕ, АБО

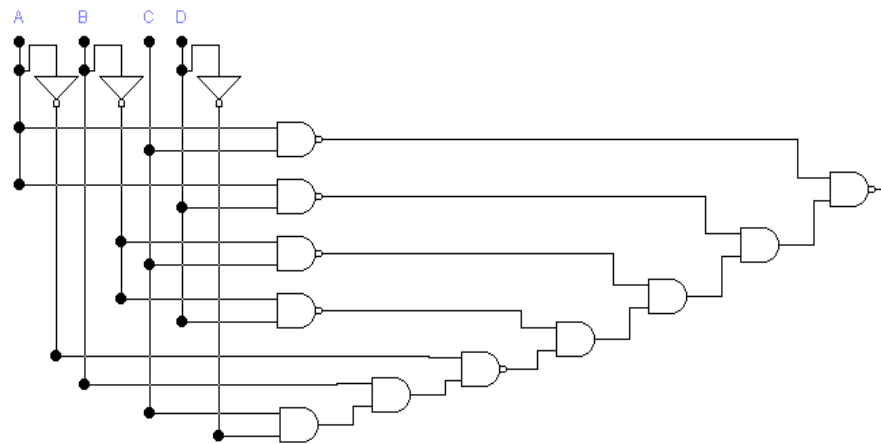


Рис. 2.5.6 Схема в базисі I-НЕ

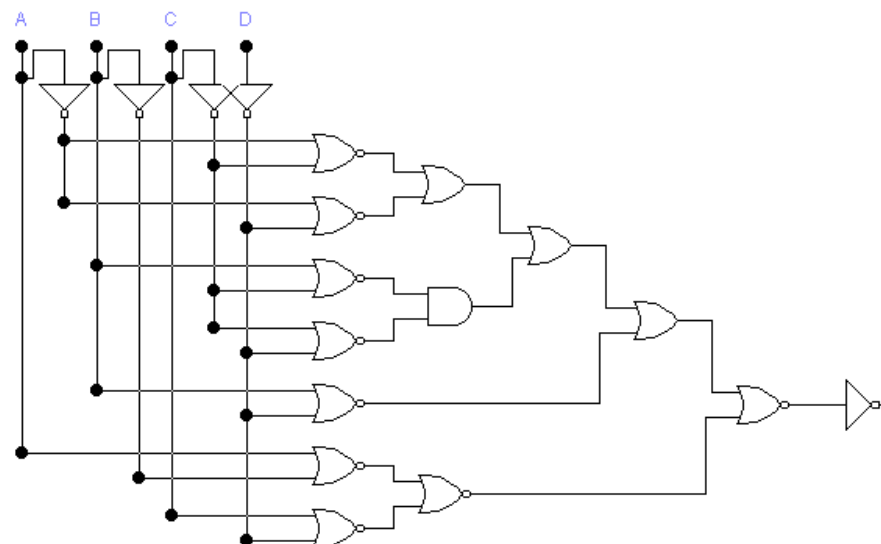


Рис. 2.5.7 Схема в базисі АБО-НЕ

6. Реалізація логічних функцій у заданому базисі.

6.1 Синтез функції f_1 у базисі I-АБО-НЕ

$$f_1 = \bar{x}_3 \bar{x}_2 x_1 x_0 + \bar{x}_3 x_2 x_1 \bar{x}_0 + \bar{x}_3 x_2 x_1 x_0 + x_3 \bar{x}_2 x_1 \bar{x}_0 + x_3 \bar{x}_2 x_1 x_0 + x_3 x_2 \bar{x}_1 x_0 + x_3 x_2 x_1 \bar{x}_0 + x_3 x_2 x_1 x_0$$

Відповідну логічну схему, виконану в пакеті Electronics Workbench на базі елементів I, АБО та НЕ наведено на рисунку (2.6.1).

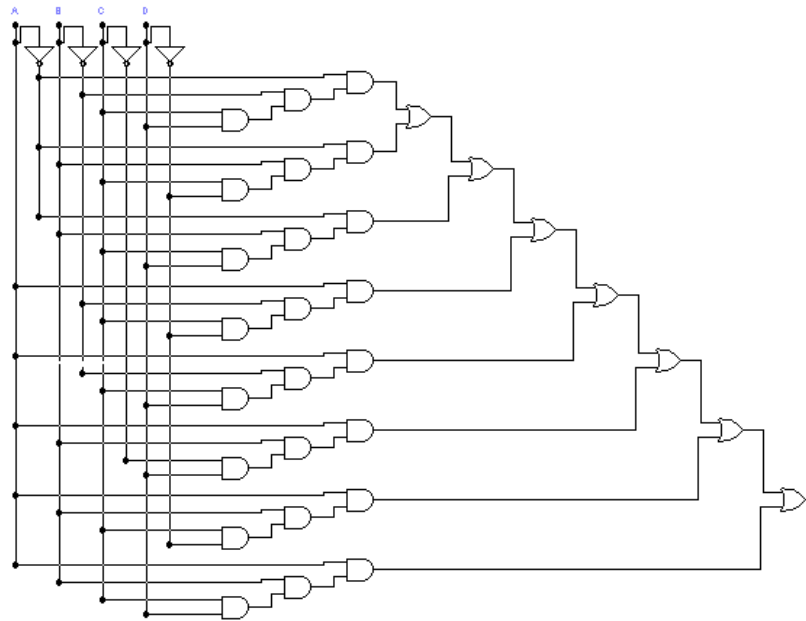


Рис. 2.6.1

6.2 Синтез функції f_2 у базисі I-НЕ

$$f(x_{n-1}, \dots, x_2, x_0) = \overline{\overline{f(x_{n-1}, \dots, x_2, x_0)}} = \overline{K_1 + K_2 + \dots + K_m} = \overline{K_1} \cdot \overline{K_2} \cdot \dots \cdot \overline{K_m}$$

$$\begin{aligned} f_2 &= \bar{x}_3 \bar{x}_2 \bar{x}_1 x_0 + \bar{x}_3 \bar{x}_2 x_1 \bar{x}_0 + \bar{x}_3 \bar{x}_2 x_1 x_0 + \bar{x}_3 x_2 x_1 x_0 + x_3 \bar{x}_2 x_1 \bar{x}_0 + x_3 \bar{x}_2 x_1 x_0 + x_3 x_2 x_1 \bar{x}_0 + x_3 x_2 x_1 x_0 = \\ &= \overline{\bar{x}_3 \bar{x}_2 \bar{x}_1 x_0 \cdot \bar{x}_3 \bar{x}_2 x_1 \bar{x}_0 \cdot \bar{x}_3 \bar{x}_2 x_1 x_0 \cdot \bar{x}_3 x_2 x_1 x_0 \cdot x_3 \bar{x}_2 x_1 \bar{x}_0 \cdot x_3 \bar{x}_2 x_1 x_0 \cdot x_3 x_2 x_1 \bar{x}_0 \cdot x_3 x_2 x_1 x_0} \end{aligned}$$

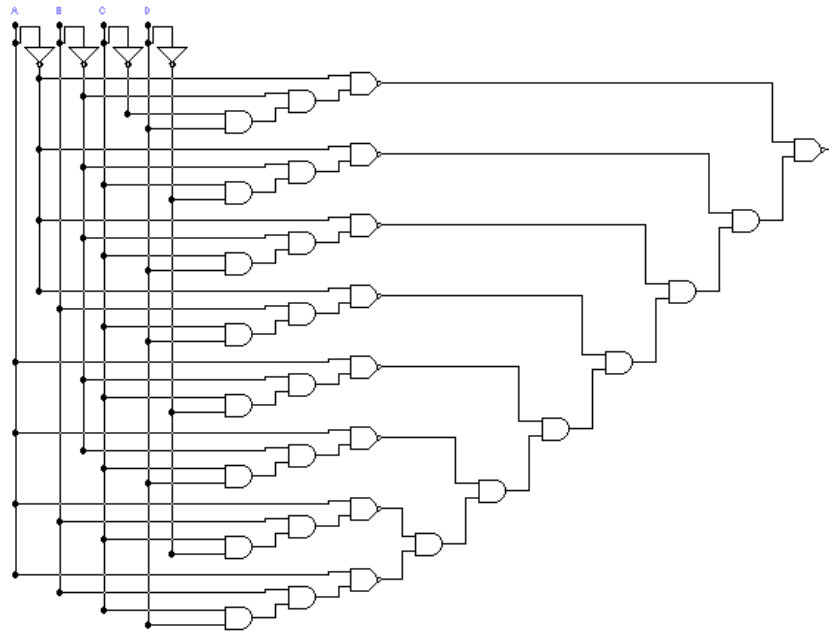


Рис. 2.6.2

6.3 Синтез функції f_3 у базисі АБО-НІ, при цьому використати тільки двовходові елементи;

$$\begin{aligned}
 f_3 = & \bar{x}_3 \bar{x}_2 \bar{x}_1 \bar{x}_0 + \bar{x}_3 \bar{x}_2 x_1 \bar{x}_0 + \bar{x}_3 \bar{x}_2 x_1 x_0 + \bar{x}_3 x_2 \bar{x}_1 \bar{x}_0 + \bar{x}_3 x_2 x_1 \bar{x}_0 + \bar{x}_3 x_2 x_1 x_0 + x_3 \bar{x}_2 \bar{x}_1 \bar{x}_0 + x_3 \bar{x}_2 x_1 \bar{x}_0 + \\
 & \underline{\underline{x_3 x_2 x_1 \bar{x}_0 + x_3 x_2 x_1 x_0}} = \underline{\underline{x_3 + x_2 + x_1 + x_0 + x_3 + x_2 + \bar{x}_1 + x_0 + x_3 + x_2 + \bar{x}_1 + \bar{x}_0 + x_3 + \bar{x}_2 + x_1 + x_0 +}} \\
 & \underline{\underline{x_3 + \bar{x}_2 + \bar{x}_1 + x_0 + x_3 + \bar{x}_2 + \bar{x}_1 + \bar{x}_0 + \bar{x}_3 + x_2 + \bar{x}_1 + x_0 + \bar{x}_3 + x_2 + \bar{x}_1 + \bar{x}_0 + \bar{x}_3 + \bar{x}_2 + \bar{x}_1 + x_0 +}} \\
 & \underline{\underline{+ \bar{x}_3 + \bar{x}_2 + \bar{x}_1 + \bar{x}_0}}
 \end{aligned}$$

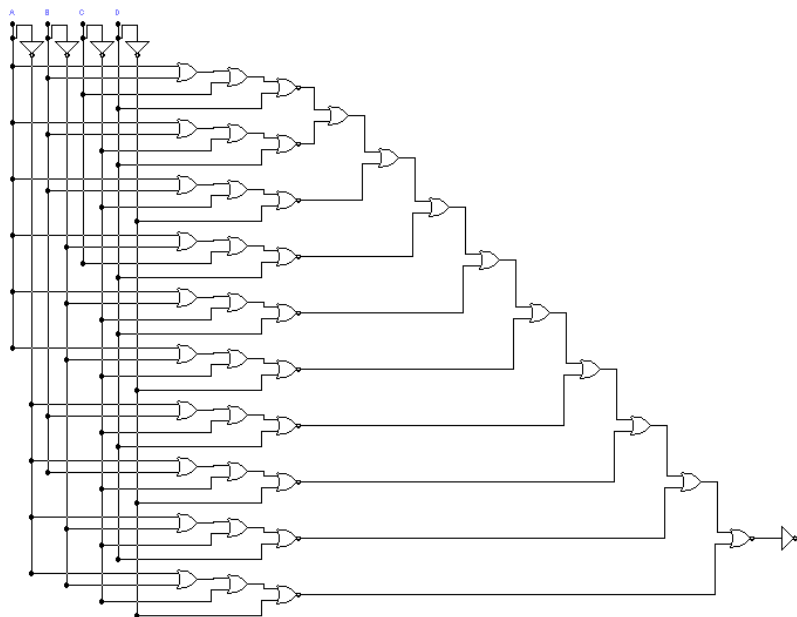


Рис. 2.6.3

7. Синтез комбінаційних схем, за допомогою дешифраторів та мультимплексорів.

7.1 Синтезувати функцію f_4 за допомогою повних дешифраторів із чотирма входами.

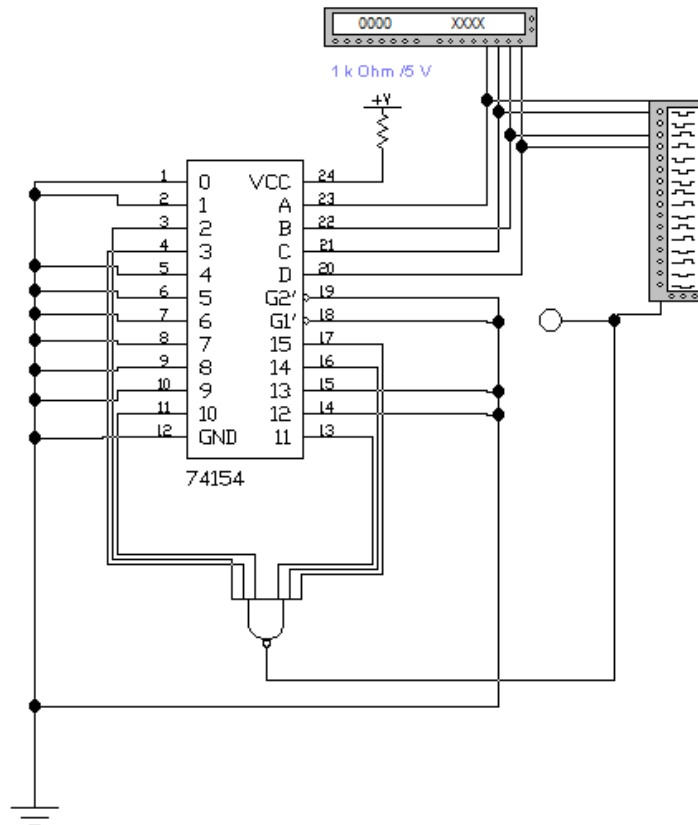


Рис. 2.7.1

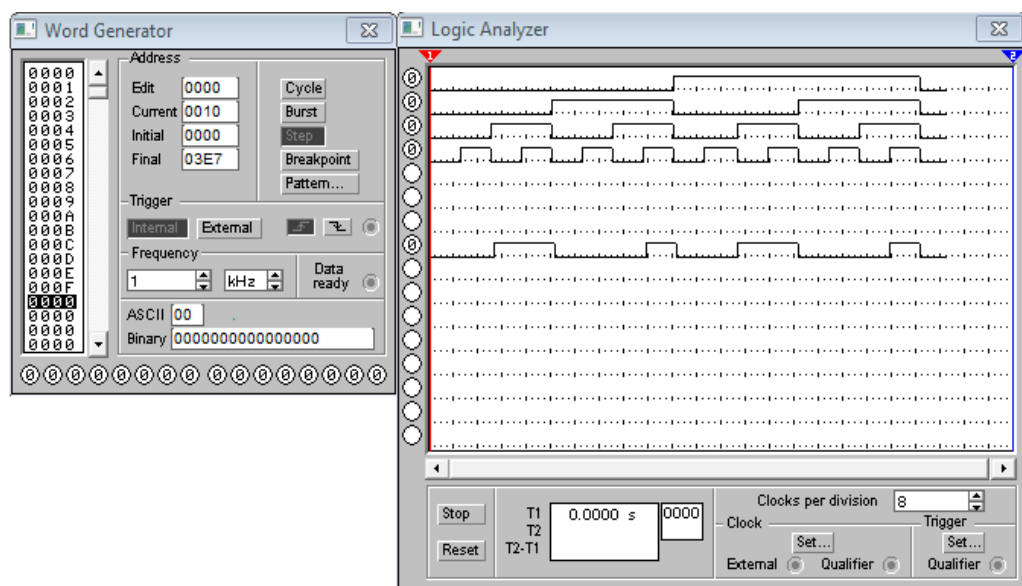


Рис. 2.7.2

7.2 Синтезувати функцію f_4 за допомогою повних дешифраторів із трьома входами.

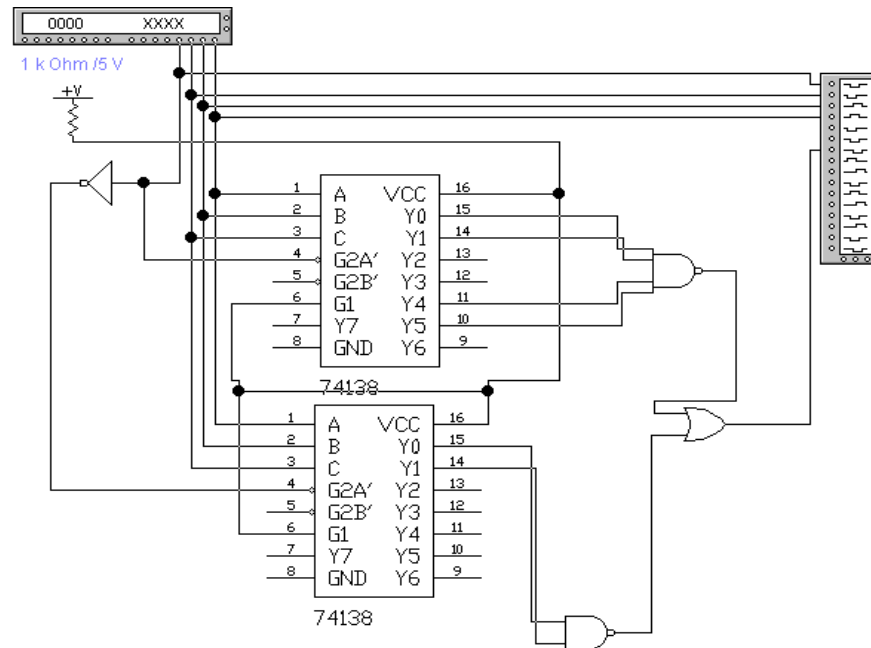


Рис. 2.7.3

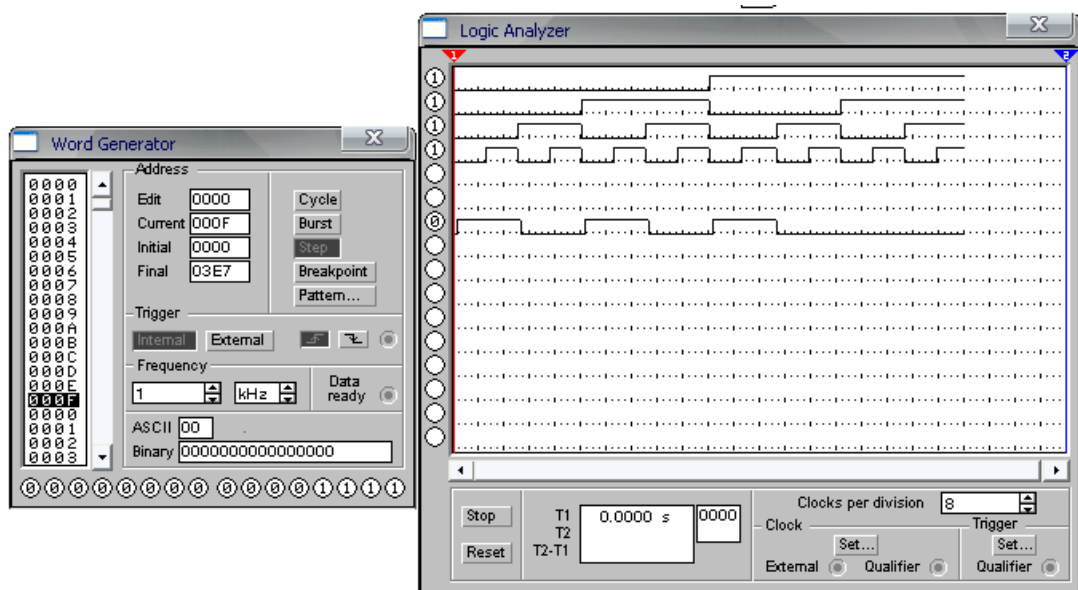


Рис. 2.7.4

7.3 Синтезувати функцію f_1 за допомогою мультиплексорів із чотирма адресними входами.

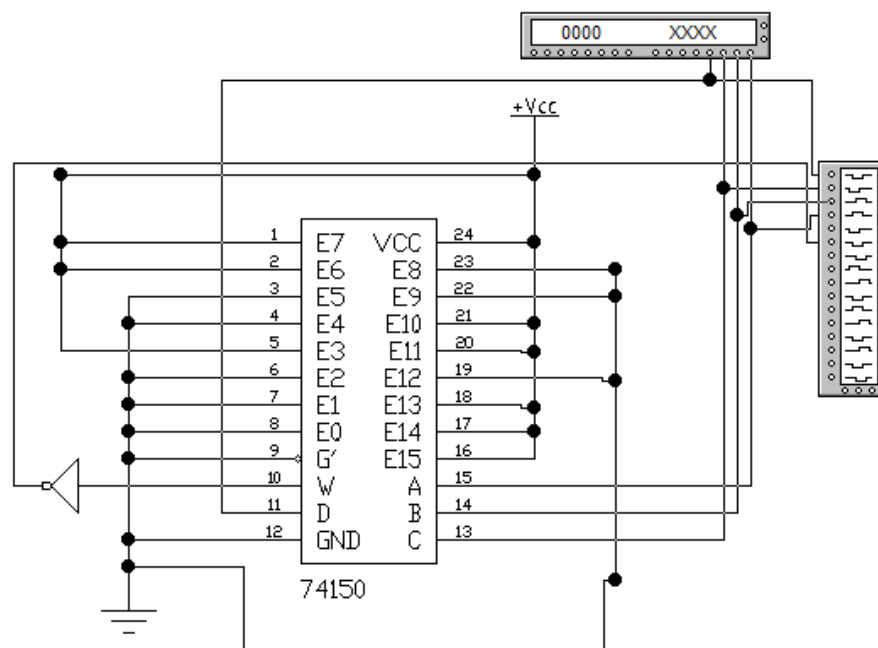


Рис. 2.7.5

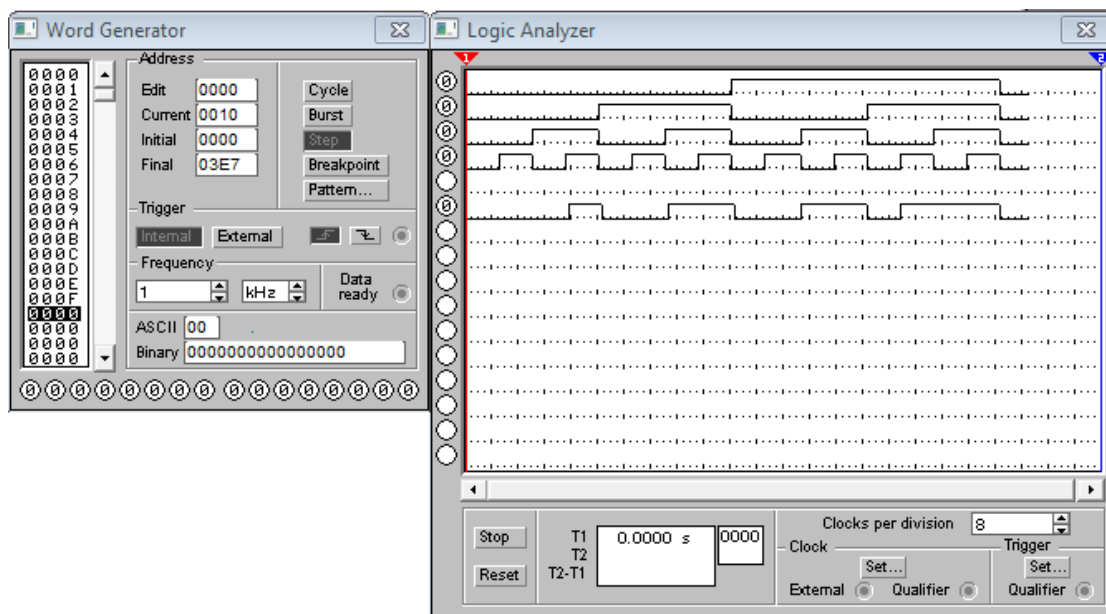


Рис. 2.7.6

7.4 Синтезувати функцію f_2 за допомогою мультимплексорів із трьома адресними входами.

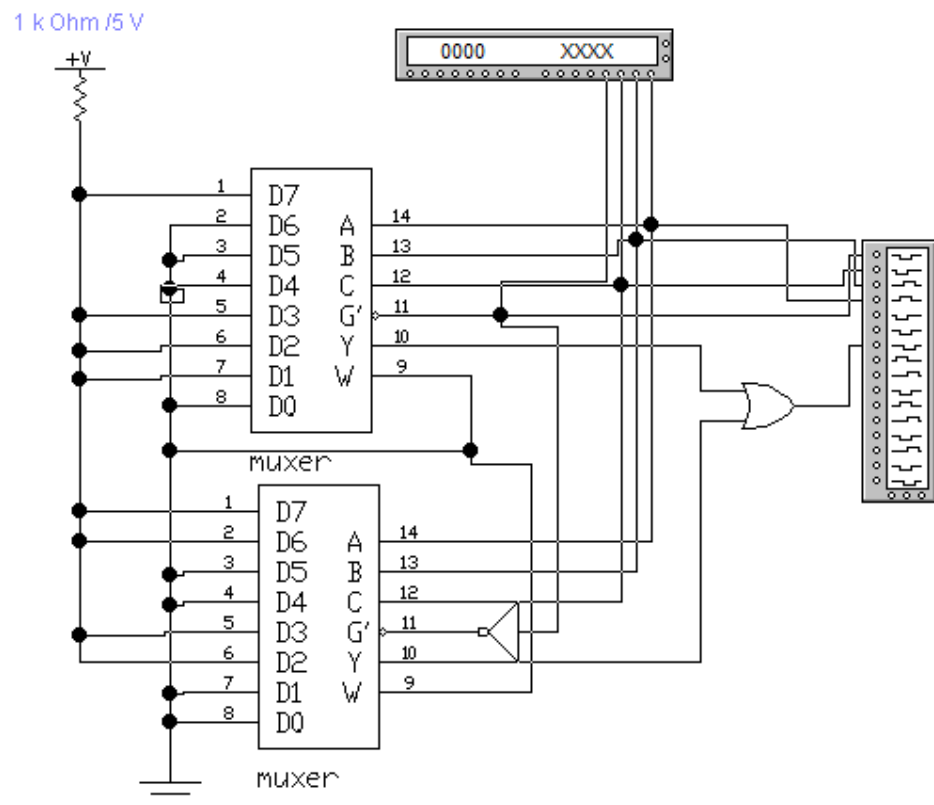


Рис. 2.7.7

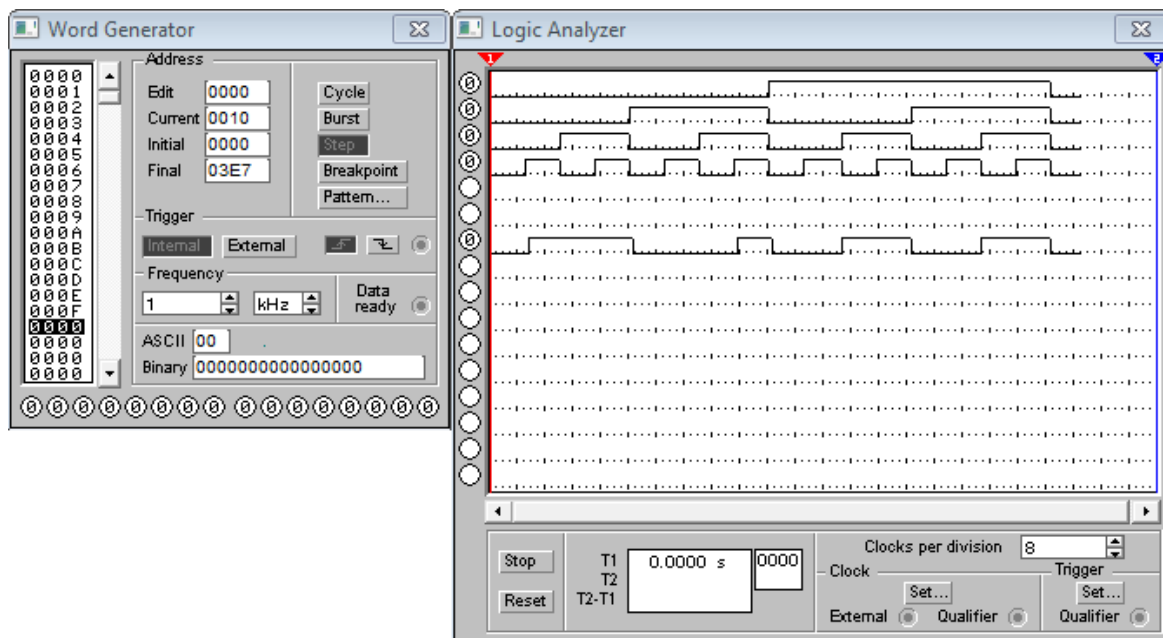


Рис. 2.7.8

7.5 Синтезувати функцію f_3 за допомогою мультиплексорів із двома адресними входами.

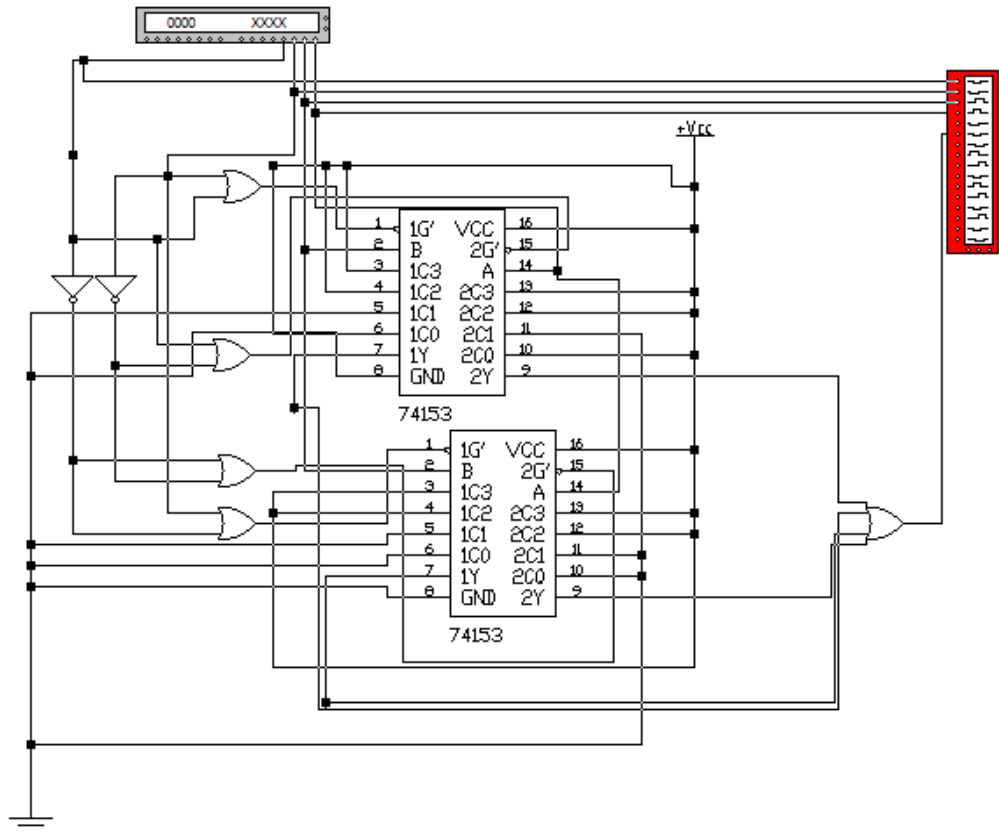


Рис. 2.7.9

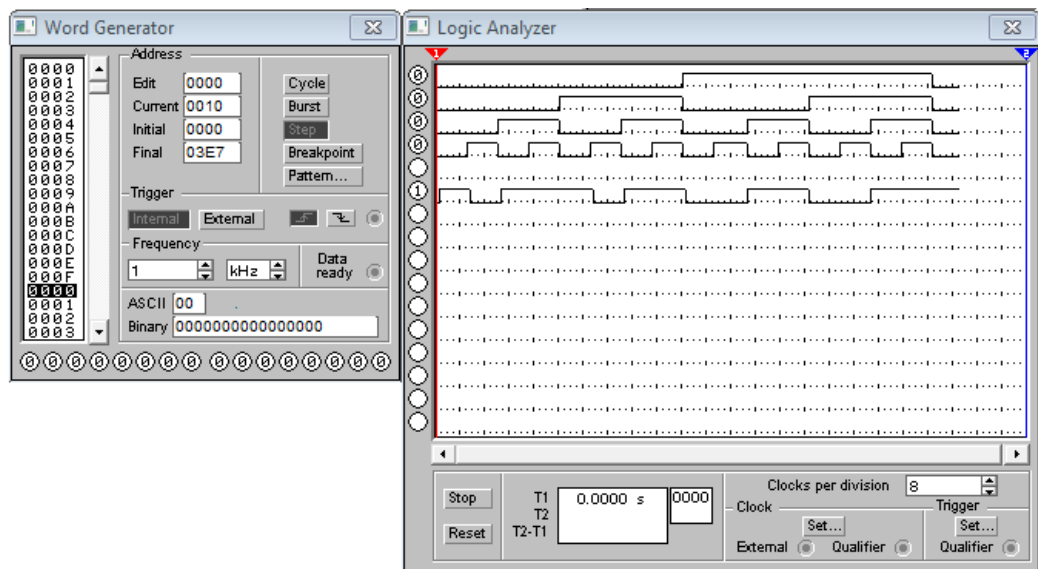


Рис. 2.7.10

II. СИНТЕЗ АВТОМАТІВ

1. Синтез цифрового автомата Мілі

Спроектувати цифровий автомат Мілі згідно заданих таблиць переходів і виходів. Комбінаційну схему збудження і комбінаційну схему формування вихідних сигналів реалізувати в заданому базисі.

Процес заміни букв алфавітів S, Y, X цифрового автомата двійковими векторами називається кодуванням і може бути описаний таблицею (табл. 8.3, табл. 8.4, табл. 8.5). В лівій частині таблиці перераховуються всі букви (наприклад вхідного алфавіту), а в правій – двійкові вектори, які ставляться у відповідність цим буквам.

Тип тригера – R-S. Базис – I-НЕ.

Таблиця переходів

Стан автомата	Вхідні сигнали	
	X ₁	X ₂
S ₁	S ₂	S ₁
S ₂	S ₁	S ₃
S ₃	S ₃	S ₂

Таблиця виходів

Стан автомата	Вхідні сигнали	
	X ₁	X ₂
S ₁	Y ₄	Y ₃
S ₂	Y ₂	Y ₄
S ₃	Y ₁	Y ₃

Виконую кодування вхідних сигналів, станів і вихідних сигналів:

Вхідні сигнали	Код
X ₁	0
X ₂	1

Стан автомата	Код
S ₁	00
S ₂	01
S ₃	10

Вихідні сигнали	Код
Y ₁	00
Y ₂	01
Y ₃	10
Y ₄	11

Таблиця переходів і виходів після кодування має наступний вигляд:

Таблиця переходів

Стан автомата	Вхідні сигнали	
	0	1
00	01	00
01	00	10
10	10	01

Таблиця виходів

Стан автомата	Вхідні сигнали	
	0	1
00	11	10
01	01	11
10	00	10

Таблиця переходів RS тригера має наступний вигляд:

Стан <i>RS-тригера</i>	Вхідний сигнал(<i>R,S</i>)		
	00	01	10
0	0	1	0
1	1	1	0

Складаємо таблицю функцій збудження автомата. Таблиця збудження елементів пам'яті будується на основі таблиці переходів

Таблиця виходу

Стан автомата		Вхідні сигнали							
		x=0				x=1			
0	0	*	0	0	1	*	0	*	0
0	1	*	0	1	0	0	1	1	0
1	0	0	*	*	0	1	0	0	1
a2	a1	R2	S2	R1	S1	R2	S2	R1	S1

$$R_2 = a_2 \bar{a}_1 x$$

$$S_2 = \bar{a}_2 a_1 \bar{x}$$

$$R_1 = \bar{a}_2 a_1 x + \bar{a}_2 a_1 x = \bar{a}_2 a_1$$

$$S_1 = \bar{a}_2 \bar{a}_1 \bar{x} + a_2 \bar{a}_1 x$$

Таблиця функції виходів будується на основі таблиці виходів:

Таблиця виходів

Стан автомата	Вхідні сигнали	
	x=0	X=1
0 0	1 1	1 0
0 1	0 1	1 1
1 0	0 0	1 0
a ₂ a ₁	y ₂ y ₁	y ₂ y ₁

Складаємо рівняння формування вихідних сигналів автомата:

$$y_1 = \bar{a}_2 \bar{a}_1 \bar{x} + \bar{a}_2 a_1 \bar{x} + \bar{a}_2 a_1 x = \bar{a}_2 \bar{a}_1 \bar{x} + \bar{a}_2 a_1$$

$$y_2 = \bar{a}_2 \bar{a}_1 \bar{x} + \bar{a}_2 \bar{a}_1 x + \bar{a}_2 a_1 x + a_2 \bar{a}_1 x = \bar{a}_2 \bar{a}_1 + \bar{a}_2 a_1 x + a_2 a_1 x$$

Перетворимо функції збудження і виходів в базис І-НЕ:

$$R_2 = \overline{a_2 \bar{a}_1 x}$$

$$S_2 = \overline{\bar{a}_2 a_1 x}$$

$$R_1 = \overline{\bar{a}_2 a_1}$$

$$S_1 = \overline{\bar{a}_2 \bar{a}_1 \bar{x} \cdot a_2 \bar{a}_1 x}$$

$$y_1 = \overline{\bar{a}_2 a_1 \cdot \bar{a}_2 \bar{a}_1 x}$$

$$y_2 = \overline{\bar{a}_2 \bar{a}_1 \cdot a_2 a_1 x \cdot a_2 a_1 x}$$

Побудуємо функціональну схему автомата Мілі (Рис 2.8.1).

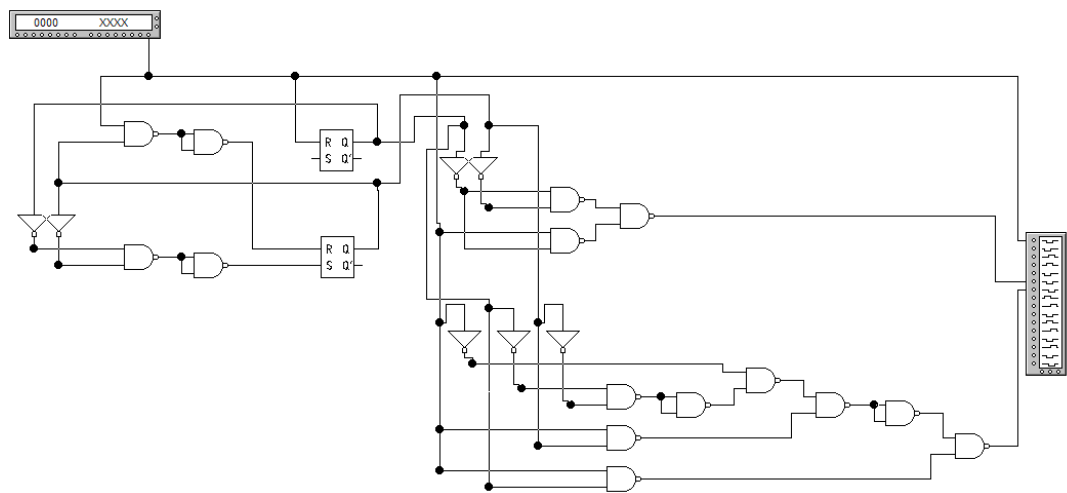


Рис. 2.8.1

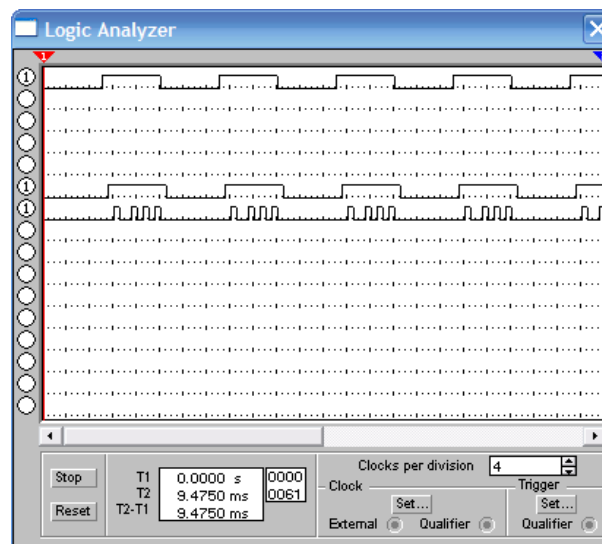


Рис. 2.8.2

ВИСНОВКИ

Головне завдання і мета курсової роботи – аналіз, синтез і структурне моделювання будь-яких дискретних скінченних систем. Розроблено універсальні (канонічні) форми представлення булевих функцій, які дають можливість одержати аналітичну форму довільної схеми безпосередньо з таблиць істинності. Ця форма надалі може бути мінімізована або спрощена. Оскільки множиною аналітичних представлень і множиною схем є взаємно однозначна відповідність, то пошук канонічної форми запису є початковим етапом синтезу логічних схем. Найбільше поширення одержала досконала диз'юнктивна нормальна форма (ДДНФ) і досконала кон'юнктивна нормальна форма (ДКНФ), а важливим етапом проектування комп'ютерних схем стала мінімізація булевих функцій, тобто знаходження їхніх виражень з мінімальним числом букв, що забезпечує побудову економічних схем комп'ютерів.

В результаті виконання курсової роботи було досліджено задані функції різними методами. Було виконано перевірку на наявність фіктивних аргументів; представлено функції у ДДНФ та ДКНФ; визначено належність функцій до п'яти предповних класів; проведено їх мінімізацію методами Квайна-Мак-Класкі, карт Вейча, Карно, К-карт; виконано синтез комбінаційних схем за допомогою повних дешифраторів із трьома, чотирма входами, мультиплексорів із двома, трьома, чотирма адресними входами; з допомогою програми Electronics Workbench зібрано функціональну схему цифрового автомата Мілі, яка відповідає заданим початковим параметрам.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Баранов С. И., Майоров С. А., Сахаров Ю. П. Автоматизация проектирования цифровых устройств. – Л.: Судостроение, 1979. –264с.
2. Шоломов Л. А. Основы теории дискретных логических и вычислительных устройств. – М.: Наука, 1980. –400с.
3. Бондаренко М.Ф, Білоус Н.В., Руткос А.Г. Комп'ютерна дискретна математика. – Харків «Компанія СМІТ». 2004. – 480 с.
4. Самофалов К.Г., Романкевич А.М. Прикладна теорія цифрових автоматів. – К.: Вища шк, 1987. – 375с.
5. Савельев А.Я., Прикладная теория цифровых автоматов: Учеб. Для вузов по спец. ЭВМ. – М.: Высш. Шк., 1987. – 272с.
6. Баранов С. И., Синтез микропрограмных автоматов. – Л.: Энергия, 1979. – 232с.
7. Каган Б.М. Электронные вычислительные машины и системы. Москва. “Энергоатомиздат ”, 1991. –592с.
8. Букреев И. Н. Микроэлектронные схемы цифровых устройств. – М.: Сов. радио 1975.–280с.
9. Лысиков Б.Г. Арифметические и логические основы цифровых автоматов.– Мн.:Высшая школа, 1980 г.
10. Майоров С. А. «Структура электронных вычислительных машин». Ленинград «Машиностроение» 1979г. с.384
11. Карцев М. А. Арифметика цифровых машин. М.:Наука. 1969. 575с.