

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до виконання курсової роботи з дисципліни
"Прикладна теорія цифрових автоматів"
для студентів за фахом
"Комп'ютерні системи та мережі",
"Спеціалізовані комп'ютерні системи"
всіх форм навчання

2010

Методичні вказівки до виконання курсової роботи з дисципліни "Прикладна теорія цифрових автоматів" для студентів за фахом "Комп'ютерні системи та мережі", "Спеціалізовані комп'ютерні системи" всіх форм навчання / Укл. А.М. Щербаков, М.П. Проскурін, С.С. Грушко - Запоріжжя: ЗНТУ, 2010. – 70 с.

Укладачі: А.М. Щербаков, к.т.н., доцент,
М.П. Проскурін, к.т.н., доцент,
С.С. Грушко, асистент

Рецензент: А.К.Тимовський, к.т.н., доцент

Відповідальний за випуск: С.С. Грушко, асистент

Затверджено на
засіданні кафедри "КСМ"

Протокол № 3
від 9 грудня 2010 р.

ЗМІСТ

ВСТУП	5
1 ЗАДАЧІ І ЗМІСТ КУРСОВОЇ РОБОТИ.....	6
1.1 Задачі курсової роботи	6
1.2 Зміст курсової роботи.....	7
2 ЗАВДАННЯ КУРСОВОЇ РОБОТИ.....	9
2.1 Завдання №1. Варіанти: A, B, C, D	9
2.2 Завдання №2. Варіанти: A, B, C, D	9
2.3 Завдання №3. Варіанти: A, B, C, D	10
2.4 Завдання №4. Варіанти: A, B, C, D	11
2.5 Завдання №5. Варіанти: A, B, C, D	12
2.6 Завдання №6. Варіанти: A, B, C, D	13
2.7 Завдання №7. Варіанти: A, B, C, D	14
2.8 Завдання №8. Варіанти: A, B, C, D	14
3 ЗАГАЛЬНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ.....	16
3.1 Поняття цифрових автоматів	16
3.1.1 Засоби опису кінцевих автоматів.....	18
3.2 Основні аксіоми, визначення та закони алгебри Буля.....	21
3.2.1 Основні аксіоми	21
3.2.2 Основні закони та співвідношення.....	22
4 СИНТЕЗ КОМБІНАЦІЙНИХ СХЕМ	27
4.1 Етапи синтезу	27
4.2 Побудова функцій алгебри логіки	28
5 МІНІМІЗАЦІЯ ФУНКЦІЙ АЛГЕБРИ ЛОГІКИ.....	32
5.1 Аналітичні методи	32
5.2 Метод невизначених коефіцієнтів	33
5.3 Метод Квайна – Мак-Класкі	35
5.4 Метод Блека – Порєцького.....	37
5.5 Мінімізація ФАЛ за допомогою карт Карно.....	37
5.6 Перехід із базису Буля до базисів Пірса та Шеффера	41
5.7 Побудова логічних схем по функціям	42
6 МЕТОДИЧНІ ВКАЗІВКИ ЩОДО РОЗРОБКИ ЦИФРОВОГО АВТОМАТА	49
6.1 Формат представлення чисел	50
6.2 Вибір зовнішніх зв'язків	51
6.3 Побудова алгоритму арифметичної дії.....	52
6.4 Побудова операційної схеми.....	59

6.5 Аналіз і синтез операційних вузлів.....	62
6.5.1 Регістр загального призначення.....	63
6.5.2 Накопичувальний суматор паралельної дії.....	67
ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	70

ВСТУП

Сучасний розвиток інформаційних технологій, обчислювальної техніки дає можливість прискорення загального наукового прогресу за рахунок використання формальних засобів синтезу цифрових автоматів, які дозволяють застосовувати автоматизацію процесів проектування. Але чим більше науковий розвиток, тим гостріше перед вищою школою стає проблема опанування студентами практичною інженерною діяльністю.

За останні роки досягнуті успіхи в виробництві різноманітних інтегральних логічних мікросхем, які використовуються в різних галузях із своїми функціональними можливостями, якістю та надійністю в роботі. Опанувати цим багажем і вміло використовувати його при практичних розробках – це ще одна із важливих задач навчання фахівців. Тому специфічна особливість підготовки фахівців цифрової техніки полягає в тому, що поступово зникає необхідність навчання студентів методам трудомістких розрахунків окремих унікальних цифрових пристроїв, а центр ваги переноситься на стадії операційного та логічного проектування на основі формальних методів синтезу. Ось чому в методичних вказівках особлива увага приділена загальним методам операційного та логічного синтезу комбінаційних логічних схем. Математичною базою проектування та дослідження цифрових автоматів є алгебра логіки. Основи двійкової алгебри логіки були закладені в середині 19 сторіччя ірландським математиком Дж. Булем. В той же час шотландський логік О. Де Морган сформулював один із основних законів алгебри логіки, названий його іменем. Далі двійкова алгебра логіки розроблялась російськими математиками П.С. Порецьким, И.И. Жегалкіним, американцями К. Шенноном, Ч. Пірсом, В. Квайном, М. Карно та іншими. Один із найважливіх елементів цифрових автоматів - елемент пам'яті типу тригер, був винайдений в 1918 році вченим М.А. Бонч-Бруєвичем. Українську школу технічної кібернетики очолював В.І. Глушков.

В методичних вказівках в стислій формі приведені основні аксіоми, закони та правила булевої алгебри, показані засоби їх використання при практичних розробках цифрових автоматів.

1 ЗАДАЧІ І ЗМІСТ КУРСОВОЇ РОБОТИ

1.1 Задачі курсової роботи

В процесі курсової роботи з дисципліни "Прикладна теорія цифрових автоматів" необхідно отримати навички самостійного проектування щодо визначених кінцевих цифрових автоматів, виконуючих задані функції дії та побудованих на логічних елементах мікросхемного базису. Ця робота охоплює лише частину загального процесу схемного проектування комплексу документації цифрового технічного обладнання.

Складність структурної та функціональної організації цифрових автоматів потребує розподілення процесу розробки на ряд послідовних етапів. Одним із перших є структурний (операційний) та логічний синтез ЦА.

На етапі операційного проектування визначають систему команд, формати та засоби керування при визначенні даних, алгоритми виконання дій та структуру ЦА на рівні операційних пристроїв.

Операційне проектування базується на технічному завданні (ТЗ) та виборі традиційних рішень, випробуваних на практиці, та оцінених в конкретних обставинах.

На етапі логічного проектування наповнюють визначену структуру пристроїв рівнем логічних схем, адекватних щодо виконання прийнятих алгоритмів дії ЦА та ефективних щодо критерію мінімальної кількості апаратних витрат на обладнання та часу виконання операцій.

Запропоновані методичні вказівки розглядають питання, які стосуються процесів операційного та логічного проектування цифрової апаратури для рішення арифметичних дій в двійковій системі числення. Арифметичні та логічні основи цифрових автоматів ЕОМ включені до

ПТЦА. Тому завдання курсових робіт орієнтоване на закріплення знання цих основ, а також засвоєння питань вибору оптимальних структурних рішень на підставі огляду, аналізу або розрахунків варіантів операційних логічних схем та алгоритмів їх дії.

Таким чином, курсова робота спрямована на закріплення знань та отримання практичних фахових навичок з питань застосування двійкових систем числення, виконання всіх арифметичних дій над

двійковими числами, побудови операційних, структурних та логічних схем окремих пристроїв, аналітичному опису логічних систем, логічних функцій в алгебрах Буля, Пірса та Шеффера, мінімізації аналітичних функцій із застосуванням основних методів: невизначених коефіцієнтів, метода Квайна – Мак-Класкі, метода карт Карно, побудови алгоритмів дії, комбінаційних та послідовностних схем цифрових автоматів на визначених логічних базисах.

Під час виконання курсової роботи студенти додатково здобувають навички користування науково-технічною та довідковою літературою, розробки технічної документації, її випуску

1.2 Зміст курсової роботи

Титульний лист і лист завдання на курсову роботу виконується на стандартних аркушах, які можна отримати у секретаря кафедри.

Пояснювальна записка виконується з урахуванням вимог ЄСКД. Рекомендований зміст:

Титульна сторінка.

Лист завдання.

Реферат (оформлення дивись СТП 15-91).

Вступ

1 Розділ 1

1.1 Підрозділ

1.2 Підрозділ

2 Розділ 2

2.1 Підрозділ

2.2 Підрозділ

3 Розділ 3

3.1 Підрозділ

3.2 Підрозділ

Висновки

Перелік посилань

Примітка. Перелік і назва розділів і підрозділів залежать від завдання і обговорюються із викладачем в індивідуальному порядку.

Пояснювальна записка обов'язково повинна містити в собі блок-схему алгоритму, операційну схему, мікропрограму дій, логічні схеми всіх операційних вузлів, їх аналітичний опис, мінімізацію в канонічних формах булевої алгебри за всіма методами, приведеними в методиці.

Схема загальна логічна розробляється на основі прийнятих рішень і розрахунків ПЗ та виконується на окремому аркуші формату А3 згідно з вимогами ЄСКД, підшивається в альбом складеним або зменшеним вдвічі.

2 ЗАВДАННЯ КУРСОВОЇ РОБОТИ

2.1 Завдання №1. Варіанти: А, В, С, D

2.1.1 Розробити цифровий автомат виконання арифметичних операцій додавання (віднімання) чисел, представлених у формі з рухомою комою, на двійковому суматорі оберненого коду з модифікованим знаком.

Розрядність ЦА1 для варіантів: А – 16, В – 24, С – 32, D – 18.

Розрядність мантиси зі знаком, порядку зі знаком обирається самостійно, оптимально.

2.1.2 Розробити та задати функції вузлів ЦА в зручних формах (таблиці істинності, таблиці переходів та виходів, графу переходів, діаграми та ін.). Визначити систему ФАЛ аналітичного опису цих вузлів, провести їх мінімізацію (всіма методами) та синтез логічних схем вузлів у базисах за варіантами:

А – базис алгебри Буля (заперечення, кон'юнкція, диз'юнкція);

В – скорочений базис алгебри Буля (заперечення, кон'юнкція, або заперечення, диз'юнкція);

С – базис алгебри Шеффера;

D – базис алгебри Пірса.

2.1.3 Синтезувати комбінаційну схему ЦА за варіантами таблиці:

Таблиця 2.1 - Варіанти синтезу комбінаційної схеми ЦА

Варіант А	Варіант В
$Y_1 = V(1,3,9,11,13,15)$ $Y_2 = V(4,5,6,8,10,11)$ $Y_3 = V(0,2,8,10,12,13,14,15)$	$Y_1 = V(0,2,5,7,8,10,13,15)$ $Y_2 = V(4,5,6,11,12,13,15)$ $Y_3 = V(8,9,10,12,13,14,15)$
Варіант С	Варіант D
$Y_1 = V(1,2,9,10,13,14)$ $Y_2 = V(4,6,7,12,14,15)$ $Y_3 = V(5,6,7,13,14,15)$	$Y_1 = V(1,2,8,11,14,15)$ $Y_2 = V(4,5,7,8,10,11,13,14)$ $Y_3 = V(0,2,8,10,12,13,14,15)$

2.2 Завдання №2. Варіанти: А, В, С, D

2.2.1 Розробити цифровий автомат (ЦА2) виконання арифметичної операції множення чисел, представлених у формі з фіксованою комою. Примініти суматор оберненого коду з

модифікованим представленням знака. Аналіз цифр множника починати з молодших розрядів.

Розрядність машини за варіантами: А – 16, В – 24, С – 32, D – 18.

2.2.2 Розробити та задати функції вузлів ЦА2 в зручних формах (таблиці істинності, таблиці переходів та виходів, граф-схеми переходів, кругових діаграм та ін.). Визначити систему ФАЛ аналітичного опису цих вузлів, провести їх мінімізацію (всіма методами) та синтез логічних схем вузлів у базисах за варіантами:

А – базис алгебри Буля (заперечення, кон'юнкція, диз'юнкція);

В – скорочений базис алгебри Буля (заперечення, кон'юнкція, або заперечення, диз'юнкція);

С – базис алгебри Шеффера;

D – базис алгебри Пірса.

2.2.3 Синтезувати комбінаційну схему ЦА за варіантами таблиці:

Таблиця 2.2 - Варіанти синтезу комбінаційної схеми ЦА

Варіант А	Варіант В
$Y1 = V(1,2,3,9,11,13,15)$ $Y2 = V(4,5,7,6,8,10,11)$ $Y3 = V(0,2,8,10,12,13,14,15)$	$Y1 = V(0,2,4,5,7,8,10,13,15)$ $Y2 = V(4,5,6,11,12,13,14,15)$ $Y3 = V(8,9,12,13,14,15)$
Варіант С	Варіант D
$Y1 = V(1,2,9,10,13,14)$ $Y2 = V(4,5,6,7,12,14,15)$ $Y3 = V(5,6,7,13,14,15)$	$Y1 = V(1,2,8,11,13,15)$ $Y2 = V(4,5,7,8,10,12,13,14)$ $Y3 = V(0,1,2,8,10,12,13,14,15)$

2.3 Завдання №3. Варіанти: А, В, С, D

2.3.1 Розробити цифровий автомат (ЦА3) виконання арифметичної операції ділення двійкових чисел, представлених у формі з рухомою комою, на суматорі додаткового модифікованого коду за методом відновлення залишку.

Розрядність ЦА3 за варіантами: А – 16, В – 24, С – 32, D – 18.

2.3.2 Розробити та задати функції вузлів ЦА3 в зручних формах (таблиці істинності, таблиці переходів та виходів, граф-схеми переходів, кругових діаграм та ін.). Визначити систему ФАЛ аналітичного опису цих вузлів, провести їх мінімізацію (всіма методами) та синтез логічних схем вузлів у базисах за варіантами:

А – базис алгебри Буля (заперечення, кон'юнкція, диз'юнкція);
 В – скорочений базис алгебри Буля (заперечення, кон'юнкція,
 або заперечення, диз'юнкція);

С – базис алгебри Шеффера;

Д – базис алгебри Пірса.

2.3.3 Синтезувати комбінаційну схему ЦА за варіантами таблиці:

Таблиця 2.3 - Варіанти синтезу комбінаційної схеми ЦА

Варіант А	Варіант В
$Y1 = V(2,3,6,7,9,11,13,15)$ $Y2 = V(0,1,4,5,6,7,8,10,11)$ $Y3 = V(0,2,8,9,10,12,13,14,15)$	$Y1 = V(0,2,4,6,7,8,10,13,15)$ $Y2 = V(3,4,5,6,11,12,13,15)$ $Y3 = V(1,3,8,9,10,12,13,14,15)$
Варіант С	Варіант Д
$Y1 = V(0,1,2,9,10,13,14)$ $Y2 = V(4,5,6,7,12,14,15)$ $Y3 = V(3,5,6,7,13,14,15)$	$Y1 = V(0,1,2,4,5,6,10,13,14)$ $Y2 = V(4,5,7,8,9,11,12,13,15)$ $Y3 = V(0,2,8,10,11,12,13,14,15)$

2.4 Завдання №4. Варіанти: А, В, С, D

2.4.1 Розробити цифровий автомат (ЦА4) виконання арифметичної операції додавання (віднімання) чисел, представлених у форматі з фіксованою комою, на двійковому суматорі додаткового коду з модифікованим знаком.

Розрядність машини за варіантами: А – 16, В – 24, С – 32, D – 18.

2.4.2 Розробити та задати функції вузлів ЦА4 в зручних формах (таблиці істинності, таблиці переходів та виходів, граф-схеми, діаграми та ін.). Визначити систему ФАЛ аналітичного опису цих вузлів, провести їх мінімізацію (всіма методами) та синтез логічних схем вузлів у базисах за варіантами:

А – базис алгебри Буля (заперечення, кон'юнкція, диз'юнкція);

В – скорочений базис алгебри Буля (заперечення, кон'юнкція,
 або заперечення, диз'юнкція);

С – базис алгебри Шеффера;

Д – базис алгебри Пірса.

2.4.3 Синтезувати комбінаційну схему ЦА за варіантами таблиці:

Таблиця 2.4 - Варіанти синтезу комбінаційної схеми ЦА

Варіант А	Варіант В
$Y1=V(1,3,8,9,10,11,13,15)$ $Y2=V(4,5,6,8,10,11)$ $Y3=V(0,2,8,10,12,13,14,15)$	$Y1=V(2,3,5,6,7,8,10,12,14)$ $Y2=V(4,5,6,11,12,14,15)$ $Y3=V(8,9,10,11,12,13,14,15)$
Варіант С	Варіант D
$Y1=V(1,3,5,7,9,10,13,14)$ $Y2=V(3,4,6,7,11,12,14,15)$ $Y3=V(4,5,6,7,12,13,14,15)$	$Y1=V(0,4,5,8,9,12,13)$ $Y2=V(8,9,10,11,12,13,14)$ $Y3=V(0,2,8,10,12,14,15)$

2.5 Завдання №5. Варіанти: А, В, С, D

2.5.1 Розробити цифровий автомат ЦА5 виконання арифметичної операції ділення двійкових чисел, представлених у формі з рухомою комою, на суматорі оберненого коду, використовуючи метод відновлення залишку.

Розрядність ЦА5 за варіантами: А – 16, В – 24, С – 32, D – 18.

2.5.2 Розробити ТЗ та задати функції вузлів ЦА5 в зручних формах (таблиці істинності, таблиці переходів та виходів, граф-схеми станів, діаграми та ін.). Визначити систему ФАЛ аналітичного опису цих вузлів, провести їх мінімізацію (всіма методами) та синтез логічних схем вузлів у базисах за варіантами:

А – базис алгебри Буля (заперечення, кон'юнкція, диз'юнкція);

В – скорочений базис алгебри Буля (заперечення, кон'юнкція, або заперечення, диз'юнкція);

С – базис алгебри Шеффера;

D – базис алгебри Пірса.

2.5.3 Синтезувати комбінаційну схему ЦА за варіантами таблиці:

Таблиця 2.5 - Варіанти синтезу комбінаційної схеми ЦА

Варіант А	Варіант В
$Y1=V(0,1,2,3,9,11)$ $Y2=V(4,5,6,7,8,10,11)$ $Y3=V(0,2,8,10,12,13,14,)$	$Y1=V(0,2,5,6,7,8,10,13,15)$ $Y2=V(4,5,6,11,12,13,15)$ $Y3=V(2,3,8,9,10,12,13,14,15)$

Таблиця 2.5 - Варіанти синтезу комбінаційної схеми ЦА

Варіант С	Варіант D
$Y1=V(0,1,2,8,9,10,13,14)$ $Y2=V(4,5,6,7,12,14,15)$ $Y3=V(5,6,7,8,9,13,14,15)$	$Y1=V(1,2,8,11,12,13,15)$ $Y2=V(0,1,4,5,7,8,10,11,13,14)$ $Y3=V(0,2,6,7,8,10,12,13,14,15)$

2.6 Завдання №6. Варіанти: А, В, С, D

2.6.1 Розробити цифровий автомат ЦА6 виконання арифметичної операції ділення двійкових чисел, представлених у формі з фіксованою комою, на суматорі оберненого коду, використовуючи метод відновлення залишку, з модифікованими кодами.

Розрядність ЦА6 за варіантами: А – 16, В – 24, С – 32, D – 18.

Коефіцієнт формату обґрунтувати оптимальним.

2.6.2 Розробити опис функцій та задати їх на головні вузли ЦА6 в зручних формах (таблиці істинності, таблиці переходів та виходів, граф-схеми, діаграми та ін.). Визначити систему ФАЛ аналітичного опису цих вузлів, провести їх мінімізацію (всіма методами) та синтез логічних схем вузлів у базисах за варіантами: А – базис алгебри Буля (заперечення, кон'юнкція, диз'юнкція);

В – скорочений базис алгебри Буля (заперечення, кон'юнкція, або заперечення, диз'юнкція);

С – базис алгебри Шеффера;

D – базис алгебри Пірса.

2.6.3 Синтезувати комбінаційну схему ЦА за варіантами таблиці:

Таблиця 2.6 - Варіанти синтезу комбінаційної схеми ЦА

Варіант А	Варіант В
$Y1=V(1,3,9,11,12,13,15)$ $Y2=V(4,5,6,8,9,10,11)$ $Y3=V(0,2,8,10,11,12,13,14,15)$	$Y1=V(0,2,4,5,7,8,10,13,15)$ $Y2=V(3,4,5,6,11,12,13,15)$ $Y3=V(8,9,10,11,12,13,14,15)$
Варіант С	Варіант D
$Y1=V(1,2,4,5,9,10,13,14)$ $Y2=V(4,6,7,8,12,14,15)$ $Y3=V(5,6,7,9,13,15)$	$Y1=V(1,2,8,11,12,13,15)$ $Y2=V(0,4,5,7,8,10,11,13,14)$ $Y3=V(0,2,3,8,10,12,13,14,15)$

2.7 Завдання №7. Варіанти: А, В, С, D

2.7.1 Розробити цифровий автомат ЦА7 виконання арифметичної операції ділення двійкових чисел, представлених у формі з фіксованою комою, на суматорі оберненого коду, використовуючи метод без відновлення залишку, з модифікованими кодами.

Розрядність ЦА7 за варіантами: А – 16, В – 24, С – 32, D – 18.

Коефіцієнт формату обрати оптимальним.

2.7.2 Розробити опис функцій та задати їх на головні вузли ЦА7 в зручних формах (таблиці істинності, таблиці переходів та виходів, граф-схеми, діаграми та ін.). Визначити систему ФАЛ аналітичного опису цих вузлів, провести їх мінімізацію (всіма методами) та синтез логічних схем вузлів у базисах за варіантами: А – базис алгебри Буля (заперечення, кон'юнкція, диз'юнкція);

В – скорочений базис алгебри Буля (заперечення, кон'юнкція, або заперечення, диз'юнкція);

С – базис алгебри Шеффера;

D – базис алгебри Пірса.

2.7.3 Синтезувати комбінаційну схему ЦА за варіантами таблиці:

Таблиця 2.7 – Варіанти синтезу комбінаційної схеми ЦА

Варіант А	Варіант В
$Y1 = V(1,3,9,11,12,13,15)$ $Y2 = V(4,5,6,8,10,12,11)$ $Y3 = V(0,2,8,10,12,14,15)$	$Y1 = V(0,2,3,5,7,8,10,13,15)$ $Y2 = V(4,5,6,9,11,12,13,15)$ $Y3 = V(0,1,8,9,10,12,13,14,15)$
Варіант С	Варіант D
$Y1 = V(1,2,7,8,9,10,13,14)$ $Y2 = V(4,6,7,11,12,14,15)$ $Y3 = V(5,6,7,9,13,14,15)$	$Y1 = V(1,2,3,8,11,13,15)$ $Y2 = V(4,5,7,8,9,10,11,13,14)$ $Y3 = V(0,2,8,10,11,12,13,14,15)$

2.8 Завдання №8. Варіанти: А, В, С, D

2.8.1 Розробити цифровий автомат ЦА8 виконання арифметичної операції ділення двійкових чисел, представлених у формі з рухомою комою, на суматорі додаткового коду, використовуючи метод без відновлення залишку, з модифікованими кодами.

Розрядність ЦА7 за варіантами: А – 16, В – 24, С – 32, D – 18.

Розрядність порядку числа обрати оптимальним.

2.8.2 Розробити опис функцій та задати їх на головні вузли ЦА8 в зручних формах (таблиці істинності, таблиці переходів та виходів, граф-схеми, діаграми та ін.). Визначити систему ФАЛ аналітичного опису цих вузлів, провести їх мінімізацію (всіма методами) та синтез логічних схем вузлів у базисах за варіантами: А – базис алгебри Буля (заперечення, кон'юнкція, диз'юнкція);

В - скорочений базис алгебри Буля (заперечення, кон'юнкція, або заперечення, диз'юнкція);

С - базис алгебри Шеффера;

D - базис алгебри Пірса.

2.8.3 Синтезувати комбінаційну схему ЦА за варіантами таблиці:

Таблиця 2.8 - Варіанти синтезу комбінаційної схеми ЦА

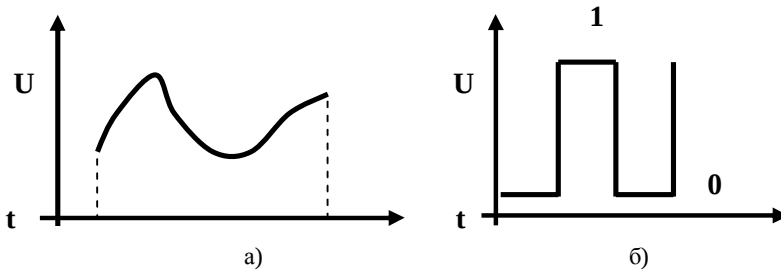
Варіант А	Варіант В
$Y1=V(1,3,4,6,9,11,13,15)$ $Y2=V(4,5,6,8,9,10,11)$ $Y3=V(0,2,7,8,9,10,12,13,14,15)$	$Y1=V(0,2,5,6,7,8,10,13,15)$ $Y2=V(4,5,6,9,11,12,13,15)$ $Y3=V(0,8,9,10,12,13,14,15)$
Варіант С	Варіант D
$Y1=V(1,2,3,9,10,13,14)$ $Y2=V(4,6,7,8,12,14,15)$ $Y3=V(5,6,7,9,13,15)$	$Y1=V(1,2,8,10,11,13,15)$ $Y2=V(4,7,8,10,11,13,14)$ $Y3=V(0,1,2,3,8,10,12,14,15)$

3 ЗАГАЛЬНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

3.1 Поняття цифрових автоматів

В електронно-обчислювальній техніці в якості сигналів управління та носіїв інформації використовують два основні види сигналів – аналоговий (неперервний), який приймає в будь-якому діапазоні часу необмежену кількість значень, або дискретний (цифровий), який приймає обмежену кількість значень.

В окремому випадку, найбільш розповсюдженому на практиці, використовують дискретний сигнал, який приймає два значення – високий рівень напруження, що за звичаєм позначається цифрою 1, та низький рівень, який позначається цифрою 0. Такі сигнали зветься двійковими або дискретними (цифровими).



а) аналоговий, б) цифровий

Рисунок 3.1 – Види сигналів

Визначення: Пристрої, які здійснюють перетворення та обробку дискретної інформації, називаються дискретними або цифровими автоматами (ЦА).

Цифровий автомат називають абстрактним, якщо не досліджується внутрішня структура, а розглядаються лише зовнішні відношення на загальному рівні. Абстрактний цифровий автомат має n входів, на які подаються дискретні сигнали (змінні) $x_1, x_2, \dots, x_n$, має кілька m виходів, на яких з'являються цифрові сигнали (функції) $y_1, y_2, \dots, y_m$, та декілька k внутрішніх станів $z_1, z_2, \dots, z_k$.

Тому що множина вхідних сигналів X , вихідних сигналів Y та вхідних станів Z кінцева, цифровий автомат ще зветь кінцевим ЦА.

ЦА функціонує в дискретному часі в послідовні моменти $t=0,1,2,..$

Перехід ЦА із одного стану Z_t в інший Z_{t+1} визначається окрім множини вхідних змінних X ще й функцією переходів δ , яка відповідає конкретним схемним рішенням, алгоритмам дії ЦА та ін. і визначає стан ЦА для часу $X(t)$ та попереднім його станом Z_{t-1} :

$$Z_t = \delta(Z_{t-1}, X_t).$$

Поява на виході сигналів із множини Y залежить від функції виходів λ , яка теж визначається конкретним схемним рішенням та протоколом його формування:

$$Y_t = \lambda(Z_{t-1}, X_t).$$

Абстрактно цифровий автомат (рисунок 3.2) можливо представити в загальній формі математичним кортежем:

$$A = \langle X, Y, Z, \delta, \lambda \rangle.$$

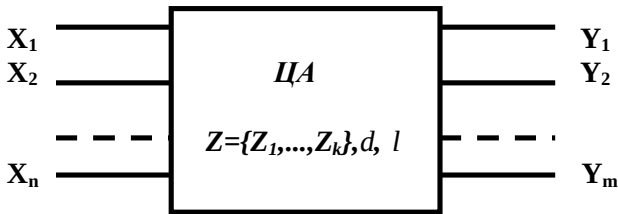


Рисунок 3.2 – Абстрактне зображення ЦА

Взагалі, кінцеві цифрові автомати розподіляють на два типи:

а) цифрові автомати без пам'яті (синоніми: комбінаційні схеми, логічні пристрої);

б) цифрові автомати із пам'яттю (синоніми: послідовні схеми, цифрові пристрої, цифрові автомати).

Цифрові автомати, які задані приведеними вище функціями переходів та виходів, звуть автоматами Мілі.

На практиці часто зустрічаються такі автомати, у яких функція виходу не залежить від вхідного слова X , а залежить лише від стану автомата Z_t : $Y_t = f(Z_t)$. Такі автомати одержали назву автоматів Мура.

Визначення: Комбінаційною схемою (КС) називається дискретний автомат, вихідна координата Y якого залежить тільки від вхідної координати змінних X , діючих в поточний момент часу t .

$$Y_t = f_{КС}(X_t). \quad (3.1)$$

Визначення: Послідовностною схемою (ПС) називається дискретний автомат, вихідна координата Y якого в майбутній момент часу $t+1$ залежить як від вхідної координати змінних X , так і від стану автомата в поточний момент часу t .

$$Y_{t+1} = f_{\text{ПС}}(X_{t+1}, Z_t). \quad (3.2)$$

Останній запис адекватний виразу:

$$Y_t = f_{\text{ПС}}(X_t, Z_{t-1}). \quad (3.3)$$

Таким чином, на відміну від комбінаційних схем (КС) вихідна координата послідовностної схеми (ПС) є функцією не тільки поточних значень вхідних змінних, але залежить і від їх попередньої послідовності, якими визначались попередні стани схеми, елементів пам'яті (тригерів).

3.1.1 Засоби опису кінцевих автоматів

Кінцеві комбінаційні цифрові автомати задаються функцією або системою функцій алгебри логіки, таблицею істинності, які визначають залежність вихідних сигналів від вхідних:

$$y_1 = f_1(x_1, x_2, \dots, x_n),$$

$$y_2 = f_2(x_1, x_2, \dots, x_n),$$

.....

$$y_m = f_m(x_1, x_2, \dots, x_n).$$

Кінцеві послідовностні цифрові автомати задаються таблицями переходів та виходів або за допомогою графа автомата, можливе й аналітичне завдання за допомогою функцій або системи функцій алгебри логіки.

В таблиці 3.1 та 3.2 дається приклад завдання автомата Мілі у формі таблиць переходів та виходів.

Таблиця 3.1 - переходів

Вхід	Із стану В стан	
	Z_0	Z_1
X_1	Z_1	Z_1
X_2	Z_0	Z_0

Таблиця 3.2 - виходів

Вхід	Стан Вихід	
	Z_0	Z_1
X_1	Y_1	Y_1
X_2	Y_2	Y_2

Стовпці визначають окремі стани (виходи) автомата, а рядки – вхідні сигнали (змінні). На їх пересіченні в таблиці переходів вказані

стан автомата, в який він перейде під дією вхідного сигналу, а в таблиці виходів – вихідний сигнал.

Граф цього автомата (рис. 3.3) побудований із вузлів окремих станів автомата Z_0, Z_1 . Дуги між вузлами відображають переходи автомата із одного стану в інший під дією вхідних сигналів. На кожній дузі вказані сформовані сигнали на виході автомата.

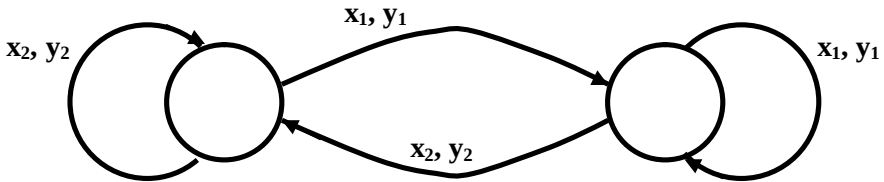


Рисунок 3.3 – Граф автомата Мілі

В таблиці 3.3 наведений приклад завдання автомата Мура у вигляді відміченої таблиці переходів.

Таблиця 3.3 – Відмічена таблиця переходів

Вхід	Y_1	Y_2
	Z_0	Z_1
X_1	Z_1	Z_1
X_2	Z_0	Z_0

В таких таблицях над станом позначається вихідний сигнал, який відповідає цьому стану.

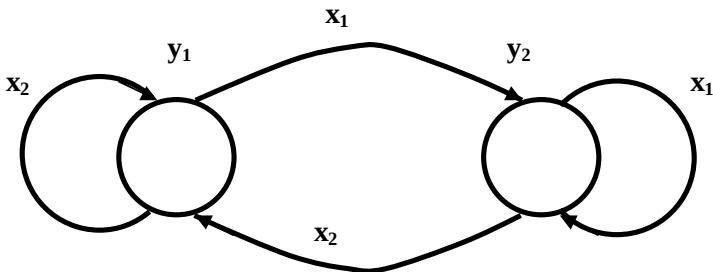


Рисунок 3.4 - Граф автомата Мура

Для завдання ЦА в аналітичній формі структуру автомата з пам'яттю ділять на дві частини: комбінаційну та пам'ять (рис. 3.5).

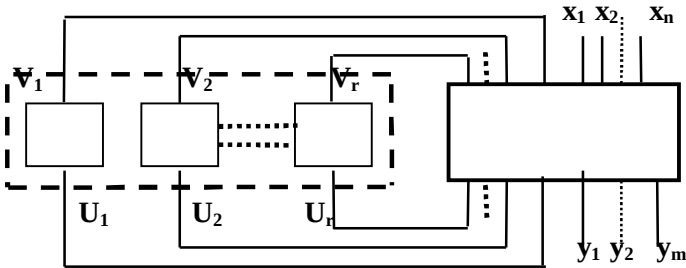


Рисунок 3.5 - Структура автомата з пам'яттю

В цьому випадку вважають, що до комбінаційної частини (схеми) надходять не тільки сигнали змінних $X_1, X_2, \dots, X_n$, а й сигнали з виходу кожного елемента пам'яті $V_1, V_2, \dots, V_r$. Та навпаки, з виходу КС знімають не тільки вихідні сигнали $Y_1, Y_2, \dots, Y_m$, а й сигнали переключення (збудження) пам'яті $U_1, U_2, \dots, U_r$.

Згідно з цим будують таблиці істинності та відшуковують систему ФАЛ у вигляді:

$$Y_1 = f_1(X_1, X_2, \dots, X_n; V_1, V_2, \dots, V_r),$$

$$Y_m = f_m(X_1, X_2, \dots, X_n; V_1, V_2, \dots, V_r),$$

$$U_1 = f_{m+1}(X_1, X_2, \dots, X_n; V_1, V_2, \dots, V_r),$$

$$U_r = f_{m+r}(X_1, X_2, \dots, X_n; V_1, V_2, \dots, V_r).$$

Приклад варіанту таблиці істинності ЦА для трьох змінних, чотирьох елементів пам'яті та чотирьох вихідних функцій заданий в таблиці 3.4.

Таблиця 3.4 - Таблиця істинності ЦА

Стан	V ₁	V ₂	V ₃	V ₄	X ₁	X ₂	X ₃	U ₁	U ₂	U ₃	U ₄	Y ₁	Y ₂	Y ₃	Y ₄
Z ₀	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Z ₁	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0
Z ₂	0	0	1	0	0	0	1	0	0	1	0	1	0	1	0
Z ₃	0	0	1	1	0	1	0	0	0	1	1	1	0	0	1

Продовження таблиці 3.4

Стан	V ₁	V ₂	V ₃	V ₄	X ₁	X ₂	X ₃	U ₁	U ₂	U ₃	U ₄	Y ₁	Y ₂	Y ₃	Y ₄
Z ₄	0	1	0	0	0	1	0	0	1	0	0	0	1	0	1
Z ₅	0	1	0	1	0	1	1	0	1	0	1	0	0	0	1
Z ₆	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0
Z ₇	0	1	1	1	1	0	0	0	1	1	1	0	0	0	0
Z ₈	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0
Z ₉	1	0	0	1	1	0	1	1	0	0	1	0	1	0	1
Z ₁₀	1	0	1	0	1	0	1	1	0	1	0	0	0	0	1
Z ₁₁	1	0	1	1	1	0	1	1	0	1	1	0	0	0	0
Z ₁₂	1	1	0	0	1	1	0	1	1	0	0	1	0	1	0
Z ₁₃	1	1	0	1	1	1	0	1	1	0	1	1	0	1	1
Z ₁₄	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1
Z ₁₅	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

Проблема більш детального математичного опису поведінки ЦА до рівня систем рівнянь $y=f(x)$, які дають змогу побудови логічних та електричних схем, може бути вирішена за допомогою математичного апарату алгебри логіки (булевої алгебри), основи якої заклав ірландський математик Дж. Буль.

Взагалі, аксіоматичним шляхом може бути побудовано декілька двозначних алгебр, які базуються на основах алгебри Буля або близькі до нього. До них можливо віднести алгебру Пірса, алгебру Шеффера, а також двійкову лінійну алгебру Жегалкіна.

3.2 Основні аксіоми, визначення та закони алгебри Буля

В алгебрі логіки вводяться відношення між змінними (висловлюваннями), такі як відношення рівності, які позначаються знаком $=$, а також три операції:

- логічне заперечення – інверсія, позначаємо рисочкою над змінною – $\bar{x}_1$;
- логічна сума – диз'юнкція, позначаємо символами $+$ або $\vee$;
- логічний добуток – кон'юнкція, позначаємо крапкою або символами $\&$ або $\wedge$.

3.2.1 Основні аксіоми

Відношення рівності характеризується трьома аксіомами: *рефлексивності* ($x = x$), *симетричності* (якщо $x_1 = x_2$, то $x_2 = x_1$) та

транзитивності (якщо $x_1 = x_2$ та $x_2 = x_3$, то $x_1 = x_3$). Якщо рівність порушена, то це позначається знаком $\neq$.

Двозначність алгебри (використання всього двох станів: 0 або 1, третього немає) витікає із аксіом двозначності:

а) $x = 0$ тільки в тому випадку, якщо $x \neq 1$;

б) $x = 1$ тільки тоді, коли $x \neq 0$.

Операція інверсії визначається аксіомами: $\overline{0} = 1$, $\overline{1} = 0$, або якщо $x = 1$, то $\overline{x} = 0$.

Операції диз'юнкції та кон'юнкції визначаються трьома аксіомами:

$$\begin{array}{ll} 0+0=0 & 1 \cdot 1=1 \\ 0+1=1+0=1 & 1 \cdot 0=0 \cdot 1=0 \\ 1+1=1 & 0 \cdot 0=0 \end{array}$$

Сума по mod2 визначається аксіомами:

$$0 \oplus 0=0 \quad 0 \oplus 1=1 \oplus 0=1 \quad 1 \oplus 1=0$$

Інверсія є унарна операція, тому що виконується над однією змінною – $\overline{x}$; сума по mod2, диз'юнкція та кон'юнкція – бінарні операції, тому що вони виконуються над двома або більше змінними ($x_1+x_2+x_3$) $x_1 \cdot x_3$. Константа 0, константа 1 – нульові операції.

Визначення: Множина $\{0,1\}$, яка виражена змінними $x_1, x_2, \dots, x_n$, над якими можна задати операції логічного заперечення, диз'юнкції та кон'юнкції, називається булевою алгеброю або алгеброю логіки.

3.2.2 Основні закони та співвідношення

Особистим законом алгебри логіки є закон ідемпотентності (у перекладі з латинського – зберігаючий ступінь незмінним), згідно якого ступінь та коефіцієнт змінних завжди дорівнює одиниці, тобто лишається не змінним:

$$x+x+\dots+x = x, \quad x \cdot x \cdot \dots \cdot x = x. \quad (3.4)$$

Закон абсорбції (поглинання):

$$\text{а) } x_1+x_1 \cdot x_2 = x_1(1+x_2) = x_1; \quad \text{б) } x_1(x_1+x_2) = x_1. \quad (3.5)$$

Закон обернення:

$$\text{Якщо } x_1=x_2, \quad \text{то } \overline{x}_1 = \overline{x}_2. \quad (3.6)$$

Закон зняття подвійного заперечення: $\overline{\overline{x}} = x$.

Закон Де Моргана:

$$\text{а) } \overline{x_1 + x_2} = \overline{x_1} \cdot \overline{x_2}; \quad \text{б) } \overline{x_1 \cdot x_2} = \overline{x_1} + \overline{x_2}. \quad (3.7)$$

Узагальнений закон дуальності (закон Де Моргана – Шеннона):

$$\text{а) } \overline{\sum x_i} = \prod \overline{x_i}; \quad \text{б) } \overline{\prod x_i} = \sum \overline{x_i}. \quad (3.8)$$

Правило Де Моргана – Шеннона: рівнозначність не порушиться, якщо взаємно замінити в ФАЛ операції диз'юнкції та кон'юнкції, проінвертувавши кожну змінну і всю ФАЛ:

$$\text{а) } x_1 + x_2 = \overline{\overline{x_1} \cdot \overline{x_2}}; \quad \text{б) } x_1 \cdot x_2 = \overline{\overline{x_1} + \overline{x_2}}. \quad (3.9)$$

або в загальній формі:

$$\text{а) } \sum x_i = \overline{\prod \overline{x_i}}; \quad \text{б) } \prod x_i = \overline{\sum \overline{x_i}}. \quad (3.10)$$

Закон комутативності (переміщувальний):

$$x_1 + x_2 = x_2 + x_1; \quad x_1 \cdot x_2 = x_2 \cdot x_1 \quad (3.11)$$

Закон асоціативності (сполучний):

$$x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3 = x_1 + x_2 + x_3; \\ x_1(x_2 \cdot x_3) = (x_1 \cdot x_2)x_3 = x_1 \cdot x_2 \cdot x_3. \quad (3.12)$$

Закон дистрибутивності (розподільний):

$$x_1 + (x_2 \cdot x_3) = (x_1 + x_2)(x_1 + x_3); \\ x_1(x_2 + x_3) = x_1 \cdot x_2 + x_1 \cdot x_3. \quad (3.13)$$

Закони алгебри логіки поширюються на будь-яку кількість змінних.

В алгебрі логіки діє той самий порядок та послідовність виконання операцій, як і в звичайній алгебрі: операція логічного множення випереджає операцію логічного додавання; можна брати вирази в дужки, розкривати їх, виносити загальний множник та інше.

Особливе значення має знак інверсії, який виконує також і роль дужок, наприклад:

- інвертування окремих змінних $\overline{x_2}$, $\overline{x_4}$, $\overline{x_6}$;

- множення $x_1 \times \overline{x_2}$, $x_3 \times \overline{x_4}$, $x_5 \times \overline{x_6}$;

- додавання під загальним знаком інверсії;
- інвертування виразу під інверсією;
- додавання $x_5 + x_6$.

В аналітичних перетворюваннях логічних функцій важливе значення мають співвідношення:

$$\begin{array}{ll} x + 0 = x & x \cdot 1 = x \\ x + 1 = 1 & x \cdot 0 = 0 \\ x + \bar{x} = 1 & x \cdot \bar{x} = 0 \end{array}$$

$$x_1 \cdot x_2 + x_1 \cdot \bar{x}_2 = x_1 \quad (x_1 + x_2)(x_1 + \bar{x}_2) = x_1 \quad (3.14)$$

Використовуючи закони алгебри логіки, можна визначати функції або системи функцій математичного відображення дії ЦА.

Введемо допоміжні поняття, які необхідні для формалізації дій із функціями алгебри логіки.

Визначення: Будь-яка змінна або її інверсія називається первинним термом (або просто термом), при цьому терми x_i та $\bar{x}_i$ є родинними.

Визначення: Логічний добуток кількох змінних, взятих в прямій x_i або інверсній формі $\bar{x}_i$, називається **елементарним добутком (контермом)** та позначається $K(x_i)$.

Аналогічно вводиться поняття **елементарної суми (дизтерма)** $D(x_i)$.

Наприклад, $x_1 \cdot \bar{x}_2 \cdot x_3$ – контерм, $x_1 + x_2 + \bar{x}_3$ – дизтерм, але $\overline{x_1 \cdot \bar{x}_2 \cdot x_3}$ – не контерм, а $\overline{x_1 + x_2 + \bar{x}_3}$ – не дизтерм, бо $\bar{x}_2 \cdot x_3$ та $x_1 + x_2$ не є первинними термами.

Визначення: Контерм (дизтерм), який містить в собі усі n змінних, називається **мінтермом (мактермом)** і позначається $m(M)$.

Іноді замість “мінтерм (мактерм)” говорять “конституента одиниці (нуля)”. При трьох змінних: $\bar{x}_1 \cdot x_2 \cdot x_3$ – мінтерм, $\bar{x}_1 + x_2 + x_3$ – мактерм.

Визначення: Основна (та дуальна) теорема алгебри логіки говорить: Будь-яка логічна функція, за винятком константи 0 (або 1),

може бути представлена у вигляді суми мінтермів (добутка макстермів). Ці форми представлення мають назву **досконалих нормальних (канонічних) форм**: досконала диз'юнктивна нормальна форма (ДДНФ), досконала кон'юнктивна нормальна форма (ДКНФ).

Згідно з основною (та дуальною) теоремою, будь-яку логічну функцію у можливо представити в таких восьми досконалих формах:

$$y = \sum m_i = \prod M_j = \overline{\sum m_j} = \overline{\prod M_i} = \sum \bar{m}_j = \prod \bar{M}_i = \overline{\sum \bar{M}_j} = \overline{\prod \bar{m}_i} \quad (3.15)$$

де $i (j)$ – ті набори змінних, на яких функція дорівнює 1 (0).

Аналогічно, функція $\bar{y}$ може бути записана в досконалий формі восьми засобами, для цього достатньо в (3.15) поміняти місцями i та j .

$$\bar{y} = \sum m_j = \prod M_i = \overline{\sum m_i} = \overline{\prod M_j} = \sum \bar{m}_i = \prod \bar{M}_j = \overline{\sum \bar{M}_i} = \overline{\prod \bar{m}_j} \quad (3.16)$$

Але такий запис не завжди є найпростішою функцією, і якщо існує така сама спрощена логічна функція, то і вона може бути представлена у вигляді восьми мінімізованих форм.

Визначення: Кількість змінних в контермі (тобто літер) визначає ранг контерма.

У попередньому прикладі ранг контермів дорівнює 2.

Визначення: Логічна скорочена сума контермів мінімального рангу називається мінімальною диз'юнктивною нормальною формою (МДНФ), а скорочений добуток дизтермів мінімального рангу називається мінімальною кон'юнктивною нормальною формою (МКНФ).

Визначення: Контерм (в окремому випадку – мінтерм), рівний нулю на тих самих наборах, що й задана логічна функція, називається імплікантою цієї функції. Ця імпліканта накриває всі нулі функції.

Визначення: Імпліканта, із якої неможливо виключити жодного терму (без порушення її властивості накривати всі нулі функції) називається простою імплікантою.

Наприклад, $y = x_1x_2 + x_1x_3 + x_2x_3$. Всі контерми цієї функції є простими імплікантами, тому що ні однієї з них не можна виключити або скоротити.

Визначення: Логічна сума простих імплікант, ні одну з яких не можливо виключити без порушення істинності. ФАЛ, називається тупиковою (скороченою)ТДНФ.

Пошук мінімальних або тупикових форм МДНФ (МКНФ) складає так звану канонічну задачу мінімізації ФАЛ, яка розпадається на дві самостійні задачі.

Першою задачею мінімізації є пошук скороченої ДНФ, тобто пошук простих імплікант. Але навіть серед простих імплікант можуть бути зайві, виключення яких не порушить істинності ФАЛ. Це і є вирішенням другої задачі ФАЛ.

4 СИНТЕЗ КОМБІНАЦІЙНИХ СХЕМ

4.1 Етапи синтезу

Задачу синтезу комбінаційної схеми можна поставити так. Описати аналітично закон функціонування КС, за визначеним законом функціонування КС необхідно побудувати схему, яка реалізує цей закон, при мінімальних витратах на обладнання.

Закон функціонування комбінаційної схеми можна задати:

- за допомогою словесного опису через прості та складні висловлювання;
- за допомогою таблиць істинності;
- у вигляді рівнянь або систем рівнянь аналітичних функцій алгебри логіки;
- за допомогою рівнянь ненульових коефіцієнтів ФАЛ;
- за допомогою карт - Карно;
- за допомогою часової діаграми.

Словесний опис функціонування КС простими висловлюваннями, об'єднаних потім за допомогою сполучників НІ, АБО, І в складні, є основою для побудови таблиці істинності, а інколи й логічних функцій. Пошук рівнянь з ненульовими коефіцієнтами, об'єднання сусідніх мінтермів карт Карно використовується при мінімізації (спрощенні) логічних виразів. Часові діаграми використовують при аналізі явища “запізень” при “гонках” в елементах схеми.

Процес синтезу комбінаційних схем містить в собі такі етапи та операції:

Опис закону функціонування – постановка задачі, виявлення умов функціонування та їх залежність від вхідних сигналів, зв'язок з іншими пристроями, побудова словесного опису роботи КС за допомогою сполучників НІ, І, АБО, побудова опису функціонування за допомогою таблиць істинності, визначення та позначення вхідних та вихідних змінних.

Аналітичний аналіз – побудова логічних функцій булевої алгебри, а якщо вихідних функцій більше однієї, то побудова системи логічних функцій булевої алгебри, використовуючи таблиці істинності, в формі досконалих ДНФ (КНФ); знаходження

мінімізованих ДНФ (КНФ), простих імплікант, тупикових форм за допомогою засобів мінімізації, наприклад, карт Карно, ненульових коефіцієнтів, Квайна та інших; перетворення мінімізованих булевих функцій, якщо потрібно, до вибраного або заданого базису логічних функцій (Пірса, Шеффера).

Побудова логічних схем – визначення рівнів логіки КС за результатом аналізу послідовності виконання логічних операцій, розподілення змінних за кількістю входів вибраних логічних елементів, побудова логічної схеми, перевірка адекватності виконання функцій схемою та аналітичними ФАЛ.

Далі йдуть етапи систем автоматизованого проектування (САПР), які до цієї програми навчання не входять.

4.2 Побудова функцій алгебри логіки

Для розгляду цієї теми та кращого її засвоєння, наведемо приклад синтезу комбінаційного суматора двійкових чисел без урахування знаків.

На таких суматорах додаються тільки модулі чисел, а знаки визначаються іншими пристроями.

Суматор може бути багато розрядним, але кожен розряд діє за однаковими арифметичними правилами додавання: $0+0=0$, $0+1=1$, $1+0=1$, $1+1=10$. Тому розглянемо синтез i -го розряду.

Опис за допомогою висловлювань. На вхід схеми надходять три змінних: a_i – розряд числа А, b_i – розряд числа В, p_i – перенос в цей розряд переповнення із молодшого розряду.

На виході схеми маємо дві функції: суму c_i розрядів $a_i + b_i$, та переповнення для переносу в старший розряд p_{i+1} . Позначимо вхідні змінні через $p_i = x_1$, $a_i = x_2$, $b_i = x_3$, а вихідні через $c_i = y_1$, $p_{i+1} = y_2$ (рис. 4.1).

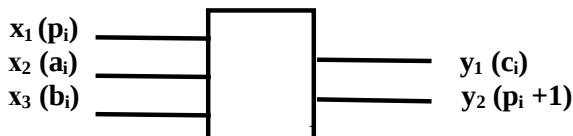


Рисунок 4.1 – Операційна схема комбінаційного суматора

Якщо $x_1 = 0$, то на виході y_2 з'явиться 1 тільки тоді, коли $x_2 = 1$ і $x_3 = 1$, в інших випадках $y_2 = 0$. На виході y_1 з'явиться 1 тільки тоді, коли $x_2 = 1$, $x_3 = 0$ або $x_2 = 0$, $x_3 = 1$, в інших випадках $y_1 = 0$.

Якщо $x_1 = 1$, то на виході y_1 з'явиться 1 у випадках, коли $x_2 = 1$, $x_3 = 1$, або $x_2 = 0$, $x_3 = 0$. На y_2 з'явиться 1, коли $x_2 = 0$, $x_3 = 1$, або $x_2 = 1$, $x_3 = 0$, або $x_2 = x_3 = 1$.

На основі цього опису будуємо таблицю істинності. табл.4.1.

Таблиця 4.1– Таблиця істинності

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

Таблиця істинності визначає значення 0 або 1 для функцій y_1 , y_2 на всіх можливих наборах змінних x_1 , x_2 , x_3 .

Із основної та дуальної теорем алгебри логіки витікає два правила запису(побудови) логічних рівнянь за таблицею істинності.

Правило 1: Використання наборів змінних, на яких функція має значення 1. Логічне рівняння в ДДНФ визначається так: в лівій частині рівняння записується вихідна функція y_i (або $\bar{y}_i$), а в правій – логічна сума мінтермів усіх наборів, де функція рівна 1; при цьому, якщо змінна дорівнює 0, то вона входить до мінтерму, як $\bar{x}_i$, і навпаки, якщо 1, то x_i . Для $\bar{y}_i$ записуються всі мінтерми, на яких функція дорівнює 0.

Правило 2: Використання наборів змінних, на яких функція має значення 0. Логічне рівняння в ДКНФ визначається так: в лівій частині рівняння записується вихідна функція y_i (або $\bar{y}_i$), а в правій – добуток макстермів усіх наборів, де функція рівна 0; при цьому, якщо змінна $x_i = 1$, то вона входить до макстерму, як $\bar{x}_i$, а якщо $x_i = 0$, то

входить в прямій формі x_i . Для $\bar{y}_i$ записуються всі макстерми, на яких функція дорівнює 1.

Правила 1,2 можливо визначити скорочено у вигляді:

$$a) y = \sum m_i = \prod M_j; \quad б) \bar{y} = \sum m_j = \prod M_i; \quad (4.1)$$

де $i(j)$ – номери тих наборів, де функція дорівнює 1 (0).

Побудуємо систему ФАЛ в ДДНФ для суматора, використовуючи таблицю істинності.

$$\begin{aligned} y_1 &= \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 \bar{x}_3 + x_1 \bar{x}_2 \bar{x}_3 + x_1 x_2 x_3; \\ y_2 &= \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3 \end{aligned} \quad (4.2)$$

Взагалі, цього вже досить для подальшої мінімізації ФАЛ та побудови логічної схеми, але з повчальною метою запишемо всі види ФАЛ.

$$\begin{aligned} \bar{y}_1 &= \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3; \\ \bar{y}_2 &= \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 \bar{x}_3 + x_1 \bar{x}_2 \bar{x}_3. \end{aligned} \quad (4.3)$$

Побудуємо систему ФАЛ у формі ДКНФ, користуючись правилом 2:

$$\begin{aligned} y_1 &= (x_1 + x_2 + x_3) \cdot (x_1 + \bar{x}_2 + \bar{x}_3) \cdot (\bar{x}_1 + x_2 + \bar{x}_3) \cdot (\bar{x}_1 + \bar{x}_2 + x_3); \\ y_2 &= (x_1 + x_2 + x_3) \cdot (x_1 + x_2 + \bar{x}_3) \cdot (x_1 + \bar{x}_2 + x_3) \cdot (\bar{x}_1 + x_2 + x_3). \end{aligned} \quad (4.4)$$

Для інверсної функції:

$$\begin{aligned} \bar{y}_1 &= (x_1 + x_2 + \bar{x}_3) \cdot (x_1 + \bar{x}_2 + x_3) \cdot (\bar{x}_1 + x_2 + x_3) \cdot (\bar{x}_1 + \bar{x}_2 + \bar{x}_3); \\ \bar{y}_2 &= (x_1 + \bar{x}_2 + \bar{x}_3) \cdot (\bar{x}_1 + x_2 + \bar{x}_3) \cdot (\bar{x}_1 + \bar{x}_2 + x_3) \cdot (\bar{x}_1 + \bar{x}_2 + \bar{x}_3). \end{aligned} \quad (4.5)$$

На прикладі функції y_2 в ДДНФ та ДКНФ покажемо перехід із канонічних форм алгебри Буля до канонічних форм алгебри Пірса та алгебри Шеффера, інакше, від елементного базису Π , I , АБО до елементного базису АБО- Π Пірса та I - Π Шеффера.

До канонічної форми алгебри Пірса зручніше перейти від ДКНФ 4.4 за допомогою правила Де Моргана.

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

$$=(x_1 \downarrow x_2 \downarrow x_3) \downarrow (x_1 \downarrow x_2 \downarrow \bar{x}_3) \downarrow (x_1 \downarrow \bar{x}_2 \downarrow x_3) \downarrow (\bar{x}_1 \downarrow x_2 \downarrow x_3) \quad (4.6)$$

До канонічної форми (елементного базису) алгебри Шеффера зручніше перейти від ДДНФ 4.2 теж за допомогою правила Де Моргана.

$$\begin{aligned} y_2 &= \bar{x}_1 \cdot x_2 \cdot x_3 + x_1 \cdot \bar{x}_2 \cdot x_3 + x_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot x_3 = \\ &= \overline{(\bar{x}_1 \cdot x_2 \cdot x_3)} \cdot \overline{(x_1 \cdot \bar{x}_2 \cdot x_3)} \cdot \overline{(x_1 \cdot x_2 \cdot \bar{x}_3)} \cdot \overline{(x_1 \cdot x_2 \cdot x_3)} = \\ &= (\bar{x}_1 / x_2 / x_3) / (x_1 / \bar{x}_2 / x_3) / (x_1 / x_2 / \bar{x}_3) / (x_1 / x_2 / x_3). \end{aligned} \quad (4.7)$$

Ці переходи від одного базису до іншого передбачені в деяких варіантах завдань курсової роботи, але цей перехід зручніше робити після мінімізації функцій.

5 МІНІМІЗАЦІЯ ФУНКЦІЙ АЛГЕБРИ ЛОГІКИ

5.1 Аналітичні методи

Мінімізацію ФАЛ краще проводити в формі алгебри Буля, тому що суттєві результати рішення цієї задачі отримані лише для базису, який складається із заперечення, диз'юнкції та кон'юнкції.

Аналітичні методи мінімізації застосовують усі закони та правила алгебри логіки, операції узагальненого склеювання, правило

Де Моргана та інші, які дозволяють значно скоротити досконалі ДНФ або КНФ.

Покажемо це на прикладі ДДНФ функції y_1 (4.2). Мінімізуємо ДДНФ функції y_1 .

$$\begin{aligned}
 y_1 &= \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 \bar{x}_3 + x_1 \bar{x}_2 \bar{x}_3 + x_1 x_2 x_3 = \\
 &= \bar{x}_2 (\bar{x}_1 x_3 + x_1 \bar{x}_3) + x_2 (\bar{x}_1 \bar{x}_3 + x_1 x_3) = \\
 &= \bar{x}_2 (\bar{x}_1 x_3 + x_1 \bar{x}_3) + x_2 \left(\overline{\bar{x}_1 \bar{x}_3 \cdot x_1 x_3} \right) = \\
 &= \bar{x}_2 (\bar{x}_1 x_3 + x_1 \bar{x}_3) + x_2 \left(\overline{(x_1 + x_3)(\bar{x}_1 + \bar{x}_3)} \right) = \\
 &= \bar{x}_2 (\bar{x}_1 x_3 + x_1 \bar{x}_3) + x_2 \left(0 + x_1 \bar{x}_3 + \bar{x}_1 x_3 + 0 \right) = \\
 &= \bar{x}_2 (\bar{x}_1 \bar{x}_3 + \bar{x}_1 x_3) + x_2 \left(\overline{x_1 \bar{x}_3 + \bar{x}_1 x_3} \right).
 \end{aligned}$$

Позначив $x_1 \bar{x}_3 + \bar{x}_1 x_3 = z$, одержимо:

$$y_1 = \bar{x}_2 z + x_2 \bar{z}. \quad (5.1)$$

Мінімізуємо ДДНФ функції y_2 :

$$\begin{aligned}
 y_2 &= \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 \bar{x}_2 \bar{x}_3 = \\
 &= x_2 x_3 (\bar{x}_1 + \bar{x}_1) + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 = x_2 x_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 = \\
 &= \overline{x_2 x_3 \cdot x_1 \bar{x}_2 x_3 \cdot x_1 x_2 \bar{x}_3} = \overline{(x_2 + x_3) \cdot (\bar{x}_1 + x_2 + \bar{x}_3) \cdot (\bar{x}_1 + \bar{x}_2 + x_3)} = \\
 &= \overline{(x_1 \bar{x}_2 + 0 + \bar{x}_2 \bar{x}_3 + \bar{x}_1 x_3 + x_2 x_3 + \bar{x}_3 \bar{x}_3) \cdot (\bar{x}_1 + \bar{x}_2 + x_3)} =
 \end{aligned}$$

$$\begin{aligned}
&= \overline{(\bar{x}_1 \bar{x}_2 + \bar{x}_3 (\bar{x}_2 + \bar{x}_1 + x_2 + 1)) \cdot (\bar{x}_1 + \bar{x}_2 + x_3)} = \overline{(\bar{x}_1 \bar{x}_2 + \bar{x}_3) \cdot (\bar{x}_1 + \bar{x}_2 + x_3)} = \\
&= \overline{\bar{x}_1 \bar{x}_1 \bar{x}_2 + \bar{x}_1 \bar{x}_2 \bar{x}_2 + \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 \bar{x}_3 + \bar{x}_2 \bar{x}_3 + 0} = \\
&= \overline{\bar{x}_1 \bar{x}_2 (1 + 1 + x_3) + \bar{x}_1 \bar{x}_3 + \bar{x}_2 \bar{x}_3} = \overline{\bar{x}_1 \bar{x}_2 + \bar{x}_1 \bar{x}_3 + \bar{x}_2 \bar{x}_3} = \\
&= \overline{\bar{x}_1 \bar{x}_2} \cdot \overline{\bar{x}_1 \bar{x}_3} \cdot \overline{\bar{x}_2 \bar{x}_3} = (x_1 + x_2) \cdot (x_1 + x_3) \cdot (x_2 + x_3) = \\
&= (x_1 x_1 + x_1 x_3 + x_1 x_2 + x_2 x_3) \cdot (x_2 + x_3) = (x_1 + x_2 \cdot x_3) \cdot (x_2 + x_3) = \\
&= x_1 x_2 + x_1 x_3 + x_2 x_3 + x_2 x_3 = x_1 x_2 + x_1 x_3 + x_2 x_3. \tag{5.2}
\end{aligned}$$

Отримана МДНФ значно скоротилася, але аналітичний метод потребує знань та навичок щодо алгебраїчних перетворень і є не зовсім формальним, що визначає особистість його застосування.

Розглянемо найбільш поширені формальні методи мінімізації функцій алгебри Буля.

5.2 Метод невизначених коефіцієнтів

Метод невизначених коефіцієнтів полягає в знаходженні ненульових коефіцієнтів для логічних функцій, представлених в диз'юнктивній нормальній формі (ДНФ).

Визначення: Будь-яку ДНФ можливо представити у вигляді диз'юнкції кон'юнкцій з коефіцієнтом k , який приймає значення 0 або 1. Так, для функції трьох змінних загальна форма ДНФ з коефіцієнтами має вигляд:

$$\begin{aligned}
f(x_1, x_2, x_3) = & k_1^1 x_1 + k_1^0 \bar{x}_1 + k_2^1 x_2 + k_2^0 \bar{x}_2 + k_3^1 x_3 + k_3^0 \bar{x}_3 + \\
& + k_{12}^{11} x_1 x_2 + k_{12}^{10} x_1 \bar{x}_2 + k_{12}^{01} \bar{x}_1 x_2 + k_{12}^{00} \bar{x}_1 \bar{x}_2 + k_{13}^{11} x_1 x_3 + \\
& + k_{13}^{10} x_1 \bar{x}_3 + k_{13}^{01} \bar{x}_1 x_3 + k_{13}^{00} \bar{x}_1 \bar{x}_3 + k_{23}^{11} x_2 x_3 + k_{23}^{10} x_2 \bar{x}_3 + \\
& + k_{23}^{01} \bar{x}_2 x_3 + k_{23}^{00} \bar{x}_2 \bar{x}_3 + k_{123}^{111} x_1 x_2 x_3 + k_{123}^{110} x_1 x_2 \bar{x}_3 + \\
& + k_{123}^{101} x_1 \bar{x}_2 x_3 + k_{123}^{100} x_1 \bar{x}_2 \bar{x}_3 + k_{123}^{011} \bar{x}_1 x_2 x_3 + k_{123}^{010} \bar{x}_1 x_2 \bar{x}_3 + \\
& + k_{123}^{001} \bar{x}_1 \bar{x}_2 x_3 + k_{123}^{000} \bar{x}_1 \bar{x}_2 \bar{x}_3. \tag{5.3}
\end{aligned}$$

Коефіцієнти k з різними індексами є невизначеними. Якщо k приймає значення 0, то в нуль перетворюється і мінтерм, якщо $k=1$, то цей мінтерм може увійти у МДНФ.

Для прикладу візьмемо функцію y_2 суматора:

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

Її значення на наборах змінних приведені в таблиці істинності (табл. 4.1)

Загальну форму ДНФ функції та її значення на наборах можна представити у вигляді таблиці 5.1.

Таблиця 5.1 – Пошук ненульових коефіцієнтів

№ функції	Набори	Значення	Невизначені коефіцієнти Мінтерми ДНФ						
	$x_1 x_2 x_3$		k_1	k_2	k_3	k_{12}	k_{13}	k_{23}	k_{123}
f_0	0 0 0	0	0	0	0	00	00	00	000
f_1	0 0 1	0	0	0	1	00	01	01	001
f_2	0 1 0	0	0	1	0	01	00	10	010
f_3	0 1 1	1	0	1	1	01	01	11	011
f_4	1 0 0	0	1	0	0	10	10	00	100
f_5	1 0 1	1	1	0	1	10	11	01	101
f_6	1 1 0	1	1	1	0	11	10	10	110
f_7	1 1 1	1	1	1	1	11	11	11	111

Спочатку викреслюємо всі мінтерми, для яких функція має значення нуль. Це означає, що невизначені коефіцієнти при них теж повинні бути рівними нулю.

Потім викреслюємо мінтерми, для яких функція рівна одиниці, але коефіцієнти раніше були визначені нульовими.

Після викреслення нульових коефіцієнтів залишиться система мінтермів з ненульовими коефіцієнтами у вигляді:

$$k_{23}^{11} x_2 x_3 + \cancel{k_{123}^{011} x_1 x_2 x_3} = 1$$

$$k_{13}^{11} x_1 x_3 + \cancel{k_{123}^{101} x_1 x_2 x_3} = 1$$

$$k_{12}^{11} x_1 x_2 + \cancel{k_{123}^{110} x_1 x_2 x_3} = 1$$

$$k_{12}^{11} x_1 x_2 + \cancel{k_{13}^{11} x_1 x_3} + \cancel{k_{23}^{11} x_2 x_3} + \cancel{k_{123}^{111} x_1 x_2 x_3} = 1$$

Виберемо із коефіцієнтів кожного рівняння ті, які мають найменший ранг (менша кількість змінних). Інші викреслимо. Це робиться на підставі того, що диз'юнкція буде дорівнювати одиниці, якщо хоча б один мінтерм був рівен одиниці.

Тоді одержимо систему:

$$k_{23}^{11} x_2 x_3 = 1$$

$$k_{13}^{11} x_1 x_3 = 1$$

$$k_{12}^{11} x_1 x_2 = 1$$

$$k_{12}^{11} x_1 x_2 = 1$$

Після склеювання однакових коефіцієнтів одержимо МДНФ функції y_2 суматора:

$$k_{12}^{11} x_1 x_2 + k_{12}^{11} x_1 x_2 = k_{12}^{11} x_1 x_2$$

$$y_2 = x_1 x_2 + x_1 x_3 + x_2 x_3 . \quad (5.4)$$

5.3 Метод Квайна – Мак-Класкі

Із геометричної точки зору будь-який мінтерм ДДНФ являє собою координату вершини n-куба, побудованого на n одиничних відрізках по кожній змінній з довжиною рівною 0 або 1. Тому такі мінтерми зручніше представляти в числовій двійковій формі. Наприклад, мінтерм $x_1 \bar{x}_2 x_3$ представляють як 101. Метод Квайна – Мак-Класкі базується на методі Квайна, в якому задача мінімізації полягає в попарному порівнянні всіх імплікант (мінтермів), які входять до складу ДДНФ, з метою виявлення змоги поглинення змінної:

$$F \cdot x_i + F \cdot \bar{x}_i = F .$$

Таким чином, можливо понизити ранг термів. Терми, які зазнали поглинення, помічаються. Непомічені терми являють собою первинні імпліканти, які теж відносять до складу простих імплікант МДНФ. Місце поглиненої змінної позначають літерою X. За методом Квайна – Мак-Класкі спочатку всі мінтерми розбивають на підгрупи за кількістю одиниць, а потім порівнюють підгрупи 1 і 2, 2 і 3 і так далі. Умовою запису простої імпліканти до нового r-куба побудованого із склеєних імплікант є розходження в r-1 кубі тільки по одній координаті (змінній), а інші координати повинні збігатися.

Виконаємо мінімізацію функції y_2 суматора методом Квайна – Мак-Класкі.

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

Запишемо функцію в числовій формі:

$$y_2 = \bigvee_1 (011, 101, 110, 111) = \bigvee_1 (3, 5, 6, 7).$$

Складемо 0-куби у двійковій формі, а потім розіб'ємо його на підгрупи за кількістю одиниць з метою спрощення порівняння.

$$k^0 = \left\{ \begin{matrix} 011 \\ 101 \\ 110 \\ 111 \end{matrix} \right\}, \quad k_1^0 = \left\{ \begin{matrix} 011 \\ 101 \\ 110 \end{matrix} \right\}^*, \quad k_2^0 = \{111\}.$$

Порівнюючи підгрупи k_1^0 та k_2^0 , відмічаємо * прості імпліканти, які відрізняються однією координатою та будуюмо 1-куби.

$$k^1 = \left\{ \begin{matrix} x11 \\ 1x1 \\ 11x \end{matrix} \right\}$$

Побудова 2-куба неможлива, тому що імпліканти 1-куба відрізняються один від одного більш ніж однією координатою. Таким чином, будуюмо покриття z , відповідне простим імплікантам 1-куба,

$$\text{тому що непомічених суттєвих імплікант не було} \quad z = \left\{ \begin{matrix} x11 \\ 1x1 \\ 11x \end{matrix} \right\}.$$

Складаємо таблицю покриття (табл.5.2), із якої визначаємо МДНФ шляхом порівнянь вихідних імплікант (мінтермів) із простими. Клітинки таблиці помічаються, якщо проста імпліканта входить до складу якогось вихідного мінтерма. Якщо в стовбці лише одна мітка, то ця імпліканта суттєва. Якщо в рядку є дві і більше міток, то один із цих стовбців викреслюється.

Таблиця 5.2– Таблиця покриття

Первинні імпліканти	1	2	3	4
	Вихідні імпліканти (мінтерми)			
	011	101	110	111
x11	*			*
1x1		*		*
11x			*	*

Таким чином, мінімальна диз'юнктивна нормальна форма включає в себе тільки суттєві імпліканти (стовбці 1, 2, 3), тому що 4 стовбець можна викреслити. Тоді:

$$y_2 = x_1 x_2 + x_1 x_3 + x_2 x_3 .$$

5.4 Метод Блека – Порецького

Метод Блека – Порецького дозволяє побудову мінімальної ДНФ не за ДДНФ, а за довільною ДНФ заданої функції. Метод базується на застосуванні рівняння узагальненого склеювання

$$A \cdot x + B \cdot \bar{x} = A \cdot x + B \cdot \bar{x} + A \cdot B .$$

Застосуємо це до мінімізації функції y_2 .

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

Кон'юнкція перших двох термінів обертається в 0, тому беремо тільки 3 та 4 терми. Продовжимо:

$$\begin{aligned} &= \bar{x}_1 \cdot x_2 \cdot x_3 + x_1 \cdot \bar{x}_2 \cdot x_3 + x_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 = \\ &= \bar{x}_1 \cdot x_2 \cdot x_3 + x_1 \cdot \bar{x}_2 \cdot x_3 + x_1 \cdot x_2 (\bar{x}_3 + x_3 + 1) = \\ &= \underline{\bar{x}_1 \cdot x_2 \cdot x_3} + x_1 \cdot \bar{x}_2 \cdot x_3 + \underline{x_1 \cdot x_2} = \\ &= \bar{x}_1 \cdot x_2 \cdot x_3 + x_1 \cdot \bar{x}_2 \cdot x_3 + x_2 \cdot x_3 \cdot x_2 + x_1 \cdot x_2 = \\ &= \underline{\bar{x}_1 \cdot x_2 \cdot x_3} + \underline{x_2 \cdot x_3} + x_1 \cdot \bar{x}_2 \cdot x_3 + x_1 \cdot x_2 = \\ &= \underline{x_2 \cdot x_3} + \underline{x_1 \cdot \bar{x}_2 \cdot x_3} + x_1 \cdot x_2 = \\ &= x_2 \cdot x_3 + \underline{x_1 \cdot \bar{x}_2 \cdot x_3} + \underline{x_1 \cdot x_3} + x_1 \cdot x_2 = \\ &= x_2 \cdot x_3 + x_1 \cdot x_3 (\bar{x}_2 + 1) + x_1 \cdot x_2 = x_1 \cdot x_2 + x_1 \cdot x_3 + x_2 \cdot x_3 . \end{aligned}$$

Як бачимо, цей метод хоча й потребує математичного мишлення, але досить формальний підхід робить його привабливим.

5.5 Мінімізація ФАЛ за допомогою карт Карно

Карту Карно представляють прямокутником (квадратом), на якому розміщено 2^n клітин, де n – число змінних функції.

Вигляд карт Карно для двох, трьох, чотирьох і п'яти змінних приведений на рис. 5.1.

		x_2	
x_1		0	1
	0	$\bar{x}_1 \cdot \bar{x}_2$	$\bar{x}_1 \cdot x_2$
	1	$x_1 \cdot \bar{x}_2$	$x_1 \cdot x_2$

а)

		$x_2 x_3$			
x_1		00	01	11	10
	0				
	1				

б)

		$x_3 x_4$			
$x_1 x_2$		00	01	11	10
	00				
	01				
	11				
	10				

в)

		$\bar{x}_5$				x_5			
		$x_3 x_4$				$x_3 x_4$			
$x_1 x_2$		00	01	11	10	00	01	11	10
	00								
	01								
	11								
	10								

г)

а) дві; б) три; в) чотири; г) п'ять змінних

Рисунок 5.1 – Карты Карно

Засіб мінімізації ФАЛ за допомогою карт Карно (їх різновид – карти Вейча) набув широкого застосування серед фахівців. Цьому сприяли достатня формальність дій, наочність графіки, досить легкі правила. Карты будують як розвернення кубів на площині, де вершинами кубів є клітини карти, координати яких збігаються з координатами мінтермів, що позначають вершини кубів.

Карта заповнюється як таблиця істинності: значення 1 ставлять в клітині, яка відповідає набору мінтерма, на якому функція має

значення 1. Значення 0 на картах звичайно не позначають. На Рис. 5.1 а) умовно розміщені змінні x_1, x_2 . Координати клітин вказують з лівого боку та зверху в кодї Грея. Якщо порушити місця координат, то мінімізація не буде відповідати істині. Кожній клітині відповідає свій мінтерм, утворений на перетині змінних.

Поняття сусідніх клітин. Сусідніми клітинами при $n \leq 4$ вважають не тільки клітини, які мають спільну сторону, а й клітини, розташовані в крайніх рядках або стовпцях, тому що карту можливо згорнути в прямокутник (куб), і тоді сторони крайніх клітин з'єднуються. Всі кутові клітини теж є сусідніми. При $n > 4$ сусідніми є не тільки клітини, відповідні попередній умові, але й ті, що знаходяться на однакових лініях (горизонталях, вертикалях) в сусідніх картах, тому що, наприклад, карта для p яти змінних складається з двох карт для чотирьох змінних і т. д.

Для мінімізації функцій з кількістю змінних більше 7-8 карти Карно практично не використовують.

Правила мінімізації. Після заповнення клітин карти одиницями виконують об'єднання контуром сусідніх клітин.

Визначення. Контур об'єднання повинен бути лише прямокутним (квадратним). Об'єднанню підлягають лише клітини, заповнені одиницями для МДНФ (або нулями для МКНФ). Кількість одиниць, включених до контуру, повинна дорівнювати одному з чисел 2, 4, 8, 16, 32, 64,... Будь-який контур може перетинатись з іншими, тобто одна, дві та більше одиниць можуть включатись в різні контури.

Загальна кількість контурів повинна бути як можна меншою, бо це зменшує кількість мінтермів в МДНФ (МКНФ), а кількість об'єднаних одиниць повинна бути як можна більшою, бо це зменшує кількість змінних в мінтермах. Взагалі, об'єднання 2^k клітин (одиниць) виключає з мінтермів k змінних, де $k=1, 2, 3, 4, \dots$

Побудова МДНФ. Карти Карно дозволяють визначити мінімальні ДНФ (КНФ) без проміжних математичних перетворювань. Після нанесення на карту контурів кожний окремий контур відповідає значенню простої імпліканти, яку потрібно визначити і включити до МДНФ.

Визначення. Для визначення та запису простої імпліканти з карти Карно треба аналізувати рядки та стовбці, які перетинають контур по кожній змінній окремо, починаючи з x_1 . Якщо значення цієї змінної змінюється в межах контуру з 1 (0) на 0 (1), то ця змінна в

мінтерм не включається і позначається невизначеною (наприклад, літерою X). Якщо значення змінної (0 або 1) зберігається в межах контуру, то вона входить до мінтерму (простої імпліканти) і записується своїм значенням ($\bar{x}_i$, якщо 0; x_i , якщо 1).

Мінтерми всіх контурів об'єднують операцією логічного додавання і записують МДНФ.

При побудові мінімальної форми для ДКНФ карта Карно заповнюється 0, тобто мінтермами наборів, для яких функція дорівнює 0. Об'єднання нулів виконують за тими ж правилами, а МКНФ записують в кон'юнктивній формі.

Для прикладу, виконаємо мінімізацію функції:

$$Y_2 = \bar{X}_1 X_2 X_3 + X_1 \bar{X}_2 X_3 + X_1 X_2 \bar{X}_3 + X_1 X_2 X_3;$$

Рисуємо карту Карно для трьох перемінних, наносимо на карту значення 1, об'єднуємо контурами та записуємо МДНФ.

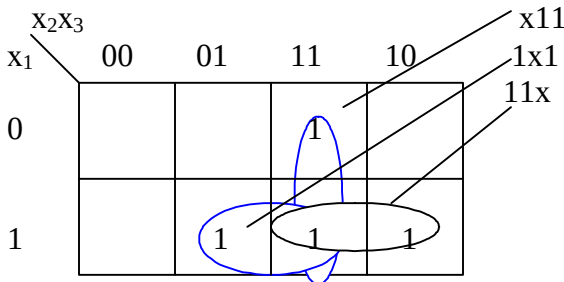


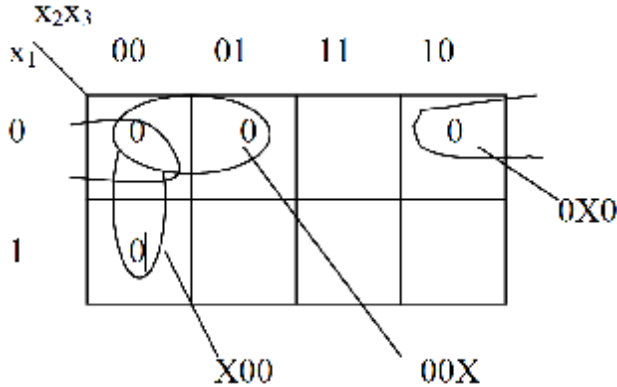
Рисунок 5.2 - Карта Карно для y_2 .

$$y_2 = x_1 x_2 + x_1 x_3 + x_2 x_3;$$

Проведемо мінімізацію для ДКНФ. З таблиці 4.1 напишемо рівняння для наборів на яких функція рівна 0.

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

Рисуємо карту Карно для трьох перемінних, визначаємо клітки із значенням 0, координати яких визначаються термами диз'юнкції, об'єднуємо контурами та записуємо інверсну мінімальну функцію $\bar{y}_2$.

Рисунок 5.3. – Карта Карно для $\bar{y}_2$.

Тоді МДНФ має вигляд:

$$\bar{y}_2 = \bar{x}_1\bar{x}_2 + \bar{x}_1\bar{x}_3 + \bar{x}_2\bar{x}_3;$$

Застосувавши закон Де Моргана, перейдемо до функції МКНФ.

$$\bar{y}_2 = \bar{x}_1\bar{x}_2 + \bar{x}_1\bar{x}_3 + \bar{x}_2\bar{x}_3;$$

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

5.6 Перехід із базису Буля до базисів Пірса та Шеффера

Перехід із базису Буля до базису Пірса виконують за допомогою рівнянь, які для двох перемінних виглядають як:

$$f(x_1, x_2) = x_1 \downarrow x_2 = \overline{x_1 + x_2} = \bar{x}_1 \cdot \bar{x}_2 \quad (5.5)$$

На основі (5.5) здійснюємо перехід від диз'юнкції до функції Пірса (Верба):

$$\left. \begin{aligned} x_1 + x_2 + \mathbf{K} + x_n &= \overline{\overline{x_1 \downarrow x_2 \downarrow \mathbf{K} \downarrow x_n}} \\ x_1 + x_2 + \mathbf{K} + x_n &= \overline{\bar{x}_1 \downarrow \bar{x}_2 \downarrow \mathbf{K} \downarrow \bar{x}_n} \end{aligned} \right\} \quad (5.6)$$

Перехід із базису Буля до базису Шиффера виконують за допомогою рівнянь:

$$x_1 / x_2 = \overline{x_1 \cdot x_2}; \quad (5.7)$$

$$x_1 \cdot x_2 = \overline{x_1 / x_2} = x_1 / x_2 / x_1 / x_2;$$

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

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

Непотрібні коди: Іноді в устроях використовують не всі набори перемінних. Тоді заповнюючи на їх місцях 1 або 0 в картах Карно, на свій погляд, і включаючи їх до контурів об'єднання 1 або 0 відповідно, значно поліпшуємо процес мінімізації за рахунок зменшення перемінних в термах об'єднання.

5.7 Побудова логічних схем по функціям

Побудову логічних схем починають з вибору логічного базису і перетворення мінімальних ФАЛ в цей базис. В курсовій роботі базис задається для кожного варіанту свій, а рівняння переходу до потрібного базису приведені в 5.6 та в таблиці 5.3.

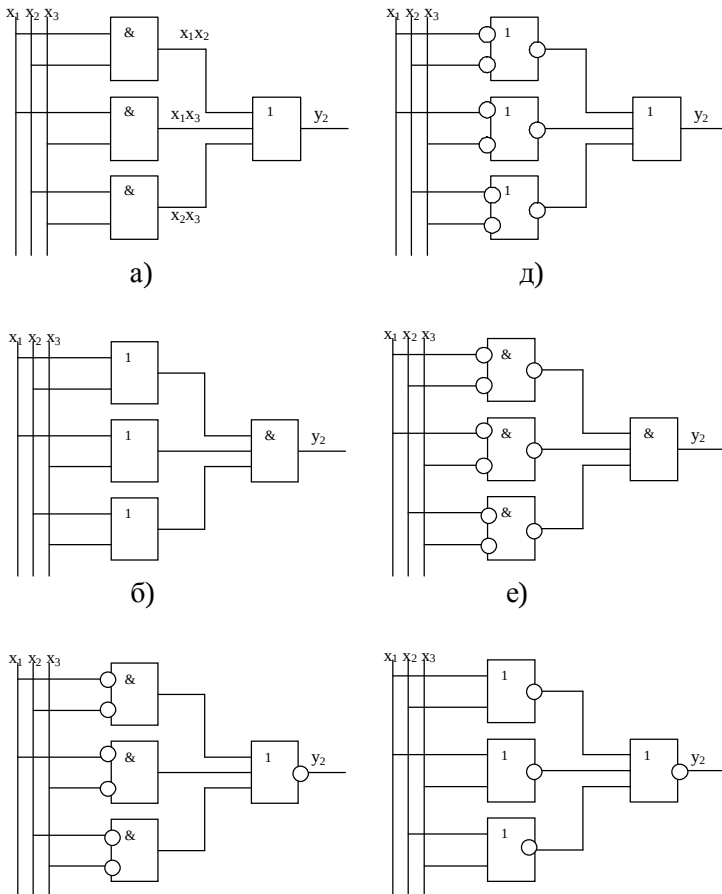
Таблиця 5.3 - Загальна форма, рівні та базиси ФАЛ суматора

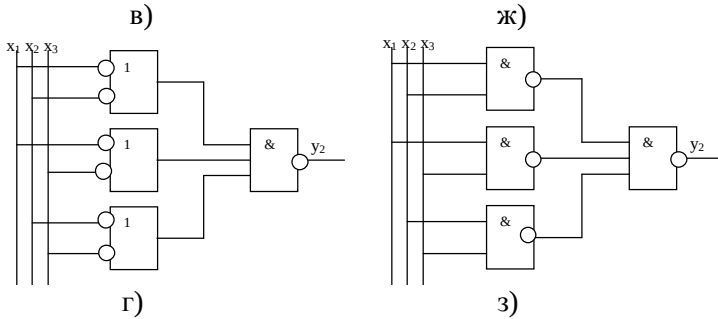
Мінімальна ФАЛ	Загальна форма	Тип логічних елементів		Базис	Схема на рис. 5.4
		1 рівень	2 рівень		
$y_2 = x_1x_2 + x_1x_3 + x_2x_3$	$\sum K(x_i)$	I	АБО	Буля	а
$y_2 = (x_1 + x_2)(x_1 + x_3)(x_2 + x_3)$	$\prod D(x_i)$	АБО	I	Буля	б
$y_2 = \overline{x_1} \cdot \overline{x_2} + \overline{x_1} \cdot \overline{x_3} + \overline{x_2} \cdot \overline{x_3}$	$\sum \overline{K(\overline{x_i})}$	I	АБО-Ні	Буля	в
$y_2 = (\overline{x_1} + \overline{x_2})(\overline{x_1} + \overline{x_3})(\overline{x_2} + \overline{x_3})$	$\prod \overline{D(\overline{x_i})}$	АБО	I-НІ	Буля	г
$y_2 = \overline{x_1} + \overline{x_2} + \overline{x_1} + \overline{x_3} + \overline{x_2} + \overline{x_3}$	$\sum \overline{D(\overline{x_i})}$	АБО-НІ	АБО	Буля	д
$y_2 = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_1} \cdot \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_3}$	$\prod \overline{K(\overline{x_i})}$	I-НІ	I	Буля	е
$y_2 = \overline{x_1} + \overline{x_2} + \overline{x_1} + \overline{x_3} + \overline{x_2} + \overline{x_3}$	$\sum \overline{D(\overline{x_i})}$	АБО-НІ	АБО-Ні	Пірса	ж
$y_2 = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_1} \cdot \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_3}$	$\prod \overline{K(\overline{x_i})}$	I-НІ	I-НІ	Шеффера	з

Звичайно, комбінаційні схеми будують по рівням логіки 1,2,...,n. До першого рівня відносять логічні операції як і в звичайній алгебрі, логічне множення, скобки, загальне заперечення (НІ) першого (нижчого) рівня. До другого рівня відносять логічне додавання, дії між скобками, дії під загальним запереченням другого рівня. При необхідності будують 3,4 рівні, але як правило, будь-яку комбінаційну

схему можливо представити у вигляді дворівневої структури, що значно зменшує “гонки” сигналів.

Приклади, варіанти побудови частини комбінаційного суматора наведено на рис. 5.4, які відповідають 8-ми варіантам представлення мінімальної ФАЛ - y_2 згідно табл. 5.3. Вхідні сигнали на схемах завжди подають на логічний елемент зліва, а виходи знімають з права. Операції зазначають на прямокутнику, зверху, зліва. Операцію АБО зазначають – 1, операцію І зазначають &, операцію НІ зазначають маленьким (діаметр 1,5-2 мм) кружечком на вході, або виході. Прямокутники мають розміри $H \times 2H$, де $H=5,10,15 \dots$ кратне 5мм.



Рисунок 5.4 – Варіанти логічних схем функцій y_2

Побудуємо, для прикладу, логічну схему комбінаційного суматора, мінімальну диз'юнктивну форму якої знайшли в 5.1.

$$y_1 = \bar{x}_2 \cdot Z + x_2 \cdot \bar{Z};$$

$$y_2 = x_1 \cdot x_2 + x_1 \cdot x_3 + x_2 \cdot x_3; \quad (5.8)$$

$$\partial e_Z = x_1 \cdot \bar{x}_3 + \bar{x}_1 \cdot x_3;$$

Для побудови логічної схеми візьмемо базис булевої алгебри НІ, І, АБО. Мінімізації система функцій y_1, y_2 не підлягає, тому що між функціями однакових мінтермів нема. Схема є на рис. 5.5.

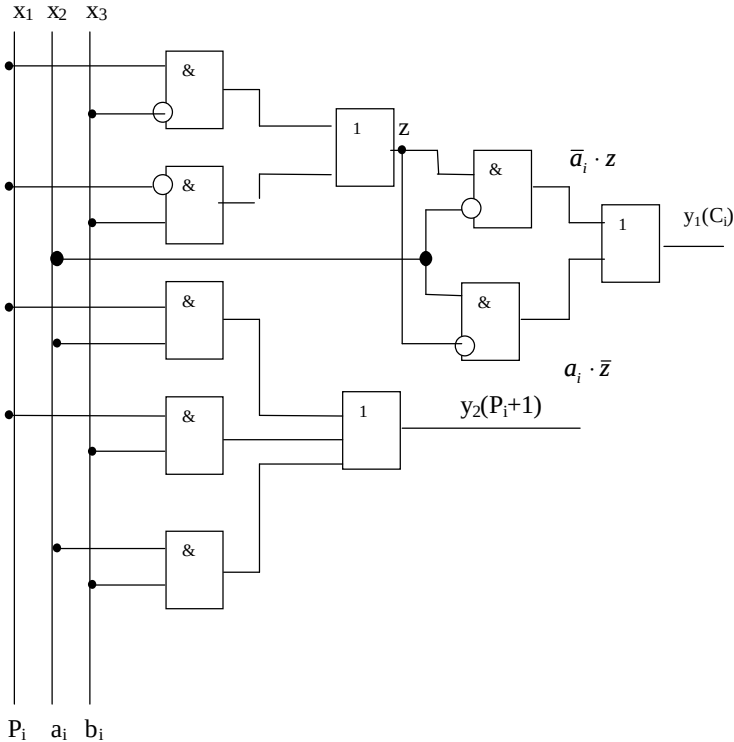


Рисунок 5.5– Однорозрядний комбінаційний суматор в базисі НІ, І, АБО

За допомогою правила Де Моргана перетворимо ФАЛ системи 5.8 із булевої алгебри до базису Шеффера (І-НІ) і побудуємо логічну схему Рис. 5.6.

$$y_1 = \bar{x}_2 z + x_2 \bar{z} = (\bar{x}_2 + \bar{z})(x_2 + z) = \bar{x}_2 z (x_2 + z) = \bar{x}_2 x_2 z + x_2 \bar{z} z = \bar{x}_2 \cdot x_2 z + x_2 \bar{z} \cdot z = [(x_2 / z) / x_2] / [(x_2 / z) / z];$$

$$y_2 = x_1 x_2 + x_1 x_3 + x_2 x_3 = \overline{\overline{x_1 x_2} \cdot \overline{x_1 x_3} \cdot \overline{x_2 x_3}} = (x_1 / x_2) / (x_1 / x_3) / (x_2 / x_3),$$

$$\begin{aligned} de \quad z &= \bar{x}_1 x_3 + x_1 \bar{x}_3 = (\bar{x}_1 + x_3)(x_1 + \bar{x}_3) = \bar{x}_1 x_3 (x_1 + \bar{x}_3) = \bar{x}_1 x_3 \cdot x_1 + \bar{x}_1 x_3 \cdot \bar{x}_3 \\ &= \overline{\overline{\bar{x}_1 x_3} \cdot \overline{x_1} \cdot \overline{x_3}} = [(x_1 / x_3) / x_1] / [(x_1 / x_3) / x_3]. \end{aligned}$$

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

Логічну схему комбінаційного суматора на елементах Пірса наведено на рис. 5.7.

На вхід суматора приходять додатки a_i та b_i по i -му розряду. Їх сума c_i представлена функцією y_1 . Перенос переповнення p_{i+1} до старшого розряду формується функцією y_2 . Всі модулі схеми є елементи алгебри Пірса.

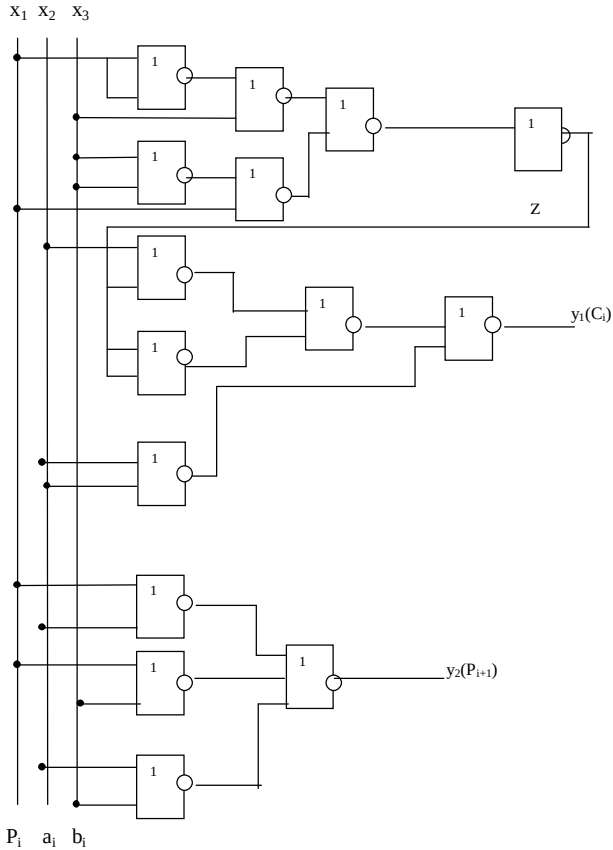


Рисунок 5.7 – Логічна схема одnorозрядного комбінаційного суматора на базі АБО-НІ (Пірса)

Найпростішою вийшла логічна схема на елементній базі Буля, але інколи варіант однотипності елементів бере перевагу.

На цьому методичні вказівки з логічних основ кінцевих цифрових автоматів закінчуються. Далі йдуть вказівки по проектуванню цифрових автоматів.

6 МЕТОДИЧНІ ВКАЗІВКИ ЩОДО РОЗРОБКИ ЦИФРОВОГО АВТОМАТА

Проектування – це процес розробки комплексу технічних документів, на основі яких може бути виготовлено обладнання цифрового автомата, який виконує задані функції з задовільною якістю функціонування.

Проектування виконується згідно з технічним завданням (ТЗ), яке, як технічний документ, розробляється студентом на основі заданого варіанта на курсову роботу. В ТЗ визначають призначення та умови роботи ЦА, функціональні дії, формат чисел, логічну базу, обмеження та інше.

Взагалі процес розробки ЦА розподіляють на етапи операційного та логічного проектування. Інші етапи (моделювання, розробка електричних схем, конструювання і т.д.) в курсовій роботі не розглядаються.

Мета операційного проектування – визначити формати та засоби кодування чисел (команд, даних), побудувати алгоритми виконання арифметичних дій, визначити необхідну систему мікро команд, мікро операцій, мікропрограми, структуру ЦА на рівні операційних схем, зовнішні зв'язки.

Мета логічного проектування – визначити мінімальну структуру та побудувати логічну схему як окремих операційних структур так і загального ЦА.

Процес розробки логічних схем розподіляють на 2 етапи: абстрактний та структурний синтез.

На етапі **абстрактного синтезу** визначаються аналітичні закони опису функціонування окремих операційних вузлів із використанням математичного апарата булевої алгебри та теорії ЦА.

На етапі **структурного синтезу** використовуються **канонічні методи** мінімізації функцій алгебри логіки, які допомагають побудувати рівносильні, але мінімальні форми ФАЛ, на базі яких проектується логічна схема з мінімальною кількістю елементів.

6.1 Формат представлення чисел

Припустимо, що згідно із завданням, числа в ЦА представляються в двійковому модифікованому коді в формі з рухомою крапкою. Розрядність машинного слова – 16.

В двійковій системі числення число N представляється адитивно:

$$N_{(2)} = \sum_{i=-p}^{+} a_i 2^{p-i}$$

де 2 – основа двійкової системи числення;
 i – номер розряду машини запису мантиси числа;
 a_i – будь-яке число із множини $\{0,1\}$;
 p – порядок цілої частини мантиси числа.

Значення p знаходиться в границях $0 \leq p \leq n$. Якщо p не фіксовано і змінюється від числа до числа то йдеться розмова про представлення числа з рухомою комою (крапкою), а значення p записують в окремі розряди машинного слова. Оптимальний вибір кількості розрядів для запису максимального цілого числа $N_{(2)} = 2^n - 1$.

З урахуванням, що знак мантиси та знак порядку потребують 4 розряди, то для мантиси залишиться $16 - 4 = 12$ розрядів. Щоб записати число 12 в двійковому коді, необхідно для p виділити чотири розряди, тоді для мантиси залишиться всього 8 розрядів.

Тоді, формат записі числа з рухомою крапкою можливо представити Рис. 6.1.

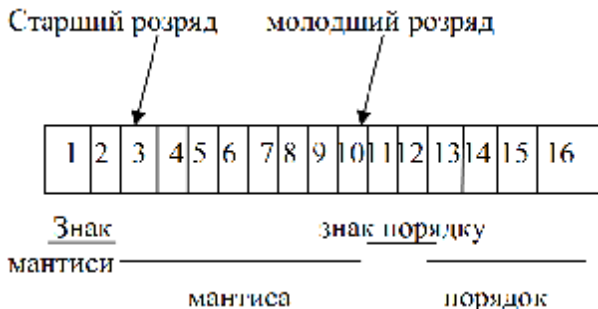


Рисунок 6.1 – Формат представлення чисел ЦА

В такому форматі повинні представлятись усі числа операційного автомата ЦА.

6.2 Вибір зовнішніх зв'язків

Процес функціонування автомата, виконуючого операцію алгебраїчного додавання двох чисел, складається із послідовної подачі на його входи чисел А та В, порозрядне додавання чисел a_i та b_i із урахуванням переносів в старші розряди та системи правил додавання, зняття та передачі результатів додавання С на вихід ЦА. Всі ці перетворення виконуються під дією сигналів управління ЦА, які подаються від пристрою керування.

Структурну схему зовнішніх зв'язків ЦА представлена рис. 6.2.

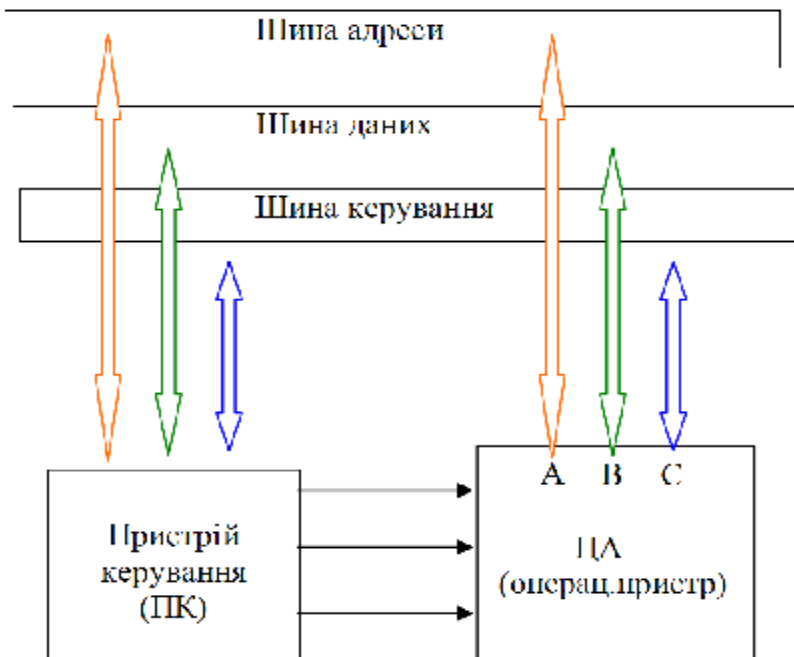


Рисунок 6.2 – Структура зв'язку ЦА

Числа А та В подаються на вхід ЦА з інтерфейсу – шини даних, результат виконання арифметичної дії – число С знімається з виходу ЦА, як це зазначено на рис. 6.2. Сигнали керування елементарними

діями, які визначають функціонування операційних схем, подаються на ЦА з пристрою керування, або по шинам інтерфейсу.

Послідовністю виконання дії ЦА, його операціями, можна керувати на основі прийнятих мікрооперацій керування, формування яких забезпечує апаратура пристрою керування.

6.3 Побудова алгоритму арифметичної дії

Окрема увага приділяється вибору алгоритму виконання арифметичної дії ЦА. Труднощі побудови ефективного алгоритму зв'язані з їх великою чисельністю, яка залежить від різноманітності структурно-схемних рішень, технічного обладнання, протоколів послідовності виконання дій та інше. Наприклад, при проектуванні ЦА для виконання арифметичних дій може використовуватися засіб паралельного порозрядного складання чисел, або послідовного; суматор прямого або оберненого додаткового коду та інше.

Спочатку алгоритм визначають змістовним описом з пошаговим виконанням операцій, а потім будується більш формалізована блок-схема алгоритму дії.

Для зображення алгоритму використовуються спеціальні графічні позначення функціональних, умовних та інших блоків, описуючих послідовні по шагові дії. Кожній дії надається значення мікро операції. Тому алгоритм має орієнтуватися на вибрані операційні блоки яким надають умовні імена та порядкові номери. Наприклад: суматор оберненого коду (СОК), суматор (СМ), суматор мантис (СМ) та ін.

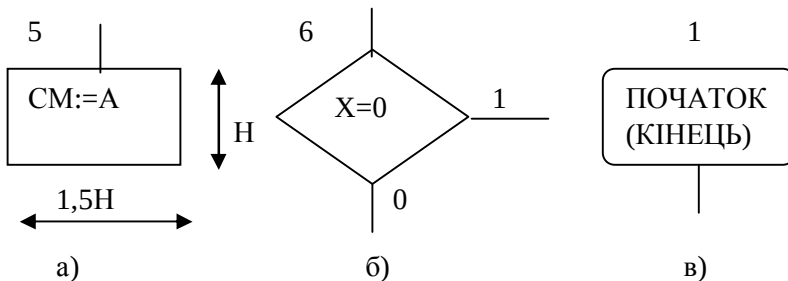


Рисунок 6.3 – Графічне позначення функціонального блока (а), аналізу умови (б) та коментарів (в)

Кожний засіб характеризується своєю швидкістю дії, потребує того чи іншого обладнання логічних мікросхем, послідовності перетворення інформації, тощо. Тому синтез операційних схем та алгоритму базується на методі послідовних приближень, а вибір чи поліпшення рішення носить евристичну вдачу.

Кожному блоку графічного позначення надається порядковий номер. Опис послідовності виконання дій алгоритму проводиться із використанням номерів блоків. Блоки поєднуються лініями зв'язку, або колом на кінцях ліній зв'язку в яких стоїть номер цього зв'язку. Лінія зв'язку входить в блок зверху, а виходить знизу без стрілок. В інших випадках проводять стрілки. Лінії зв'язку вдвоє менш товщини блоків, ведуть паралельно або перпендикулярно лініям рамки схеми. Розміри блоків вибираються із ряду $H=10\text{мм}$, далі кратно 5. З умовного блока виходять 2 умовні лінії. Над ними пишуть умову, яка очікується, а в середині блока – загальну умову.

Для опису дій алгоритму використовують оператори присвоювання $:=$, окремі знаки $;$, $\cdot$, $>$, $-$, $+$, виконання лівого зсуву числа A на p розрядів $L(p,A)$, вправо на K розрядів: $SmA:=R(2,SmA)$ означає, що число A в суматорі зсунуто на 2 розряди вправо. Операції перетворення прямого коду в обернений або додатковий визначається як $DK(PgA)$, або $OK(SmA)$. Операцію заперечення можна позначити як $-Pg\bar{A}$, що означає інверсію числа A по всім розрядам. Операція передачі числа з одного блока до іншого визначається як $Sm:=[PЗПА]ok$, що означає: “передати число A оберненим кодом з регістра загального призначення до суматора”.

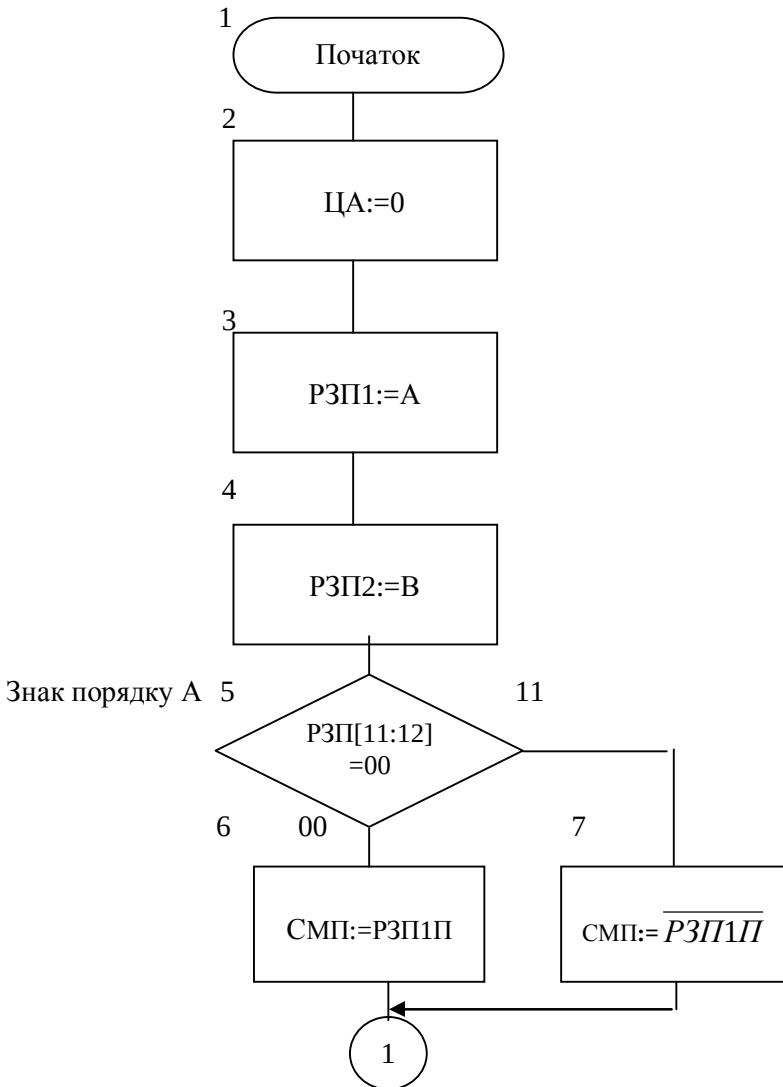
Алгоритм завжди починається з блоку “початок” і закінчується блоком “кінець”. Більш детальні положення про розробку алгоритмів можна знайти в довідковій літературі та державних стандартах. Наприклад: ДСТУ 2533-94 СОІ “Арифметичні та логічні операції” та інші.

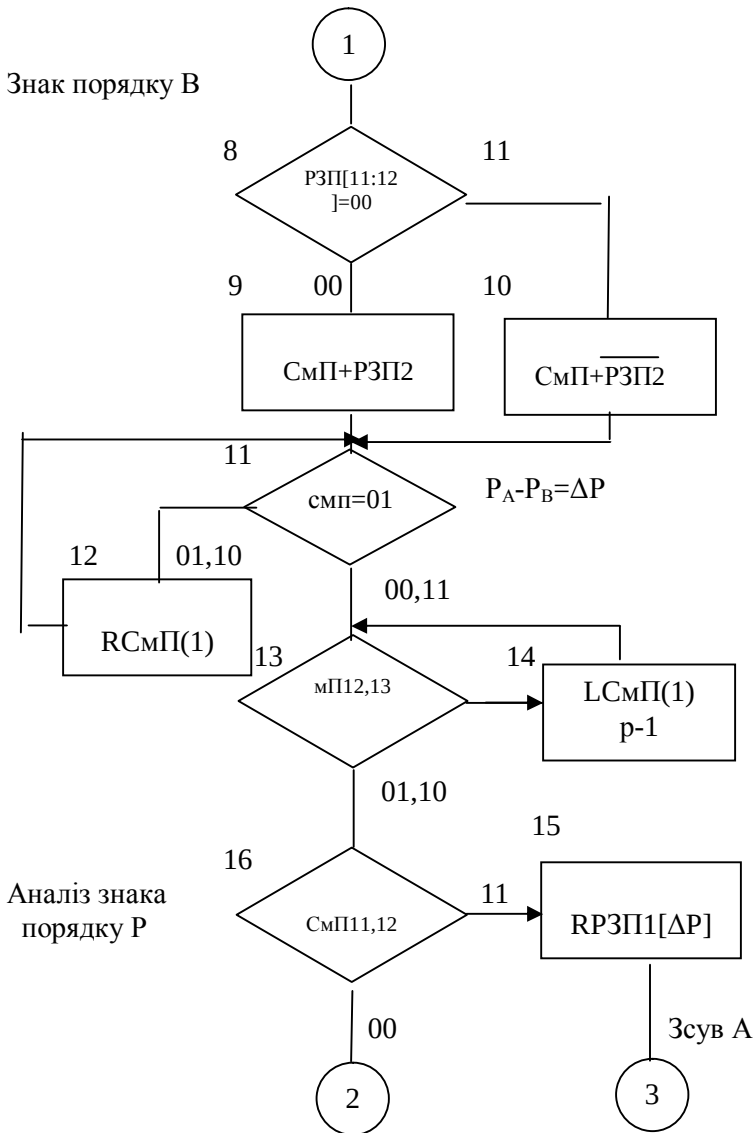
В якості приклада синтезу операційної схеми ЦА розглянемо покроковий опис алгоритму виконання алгебраїчної суми двійкових чисел на суматорі додаткового коду паралельним порозрядним засобом.

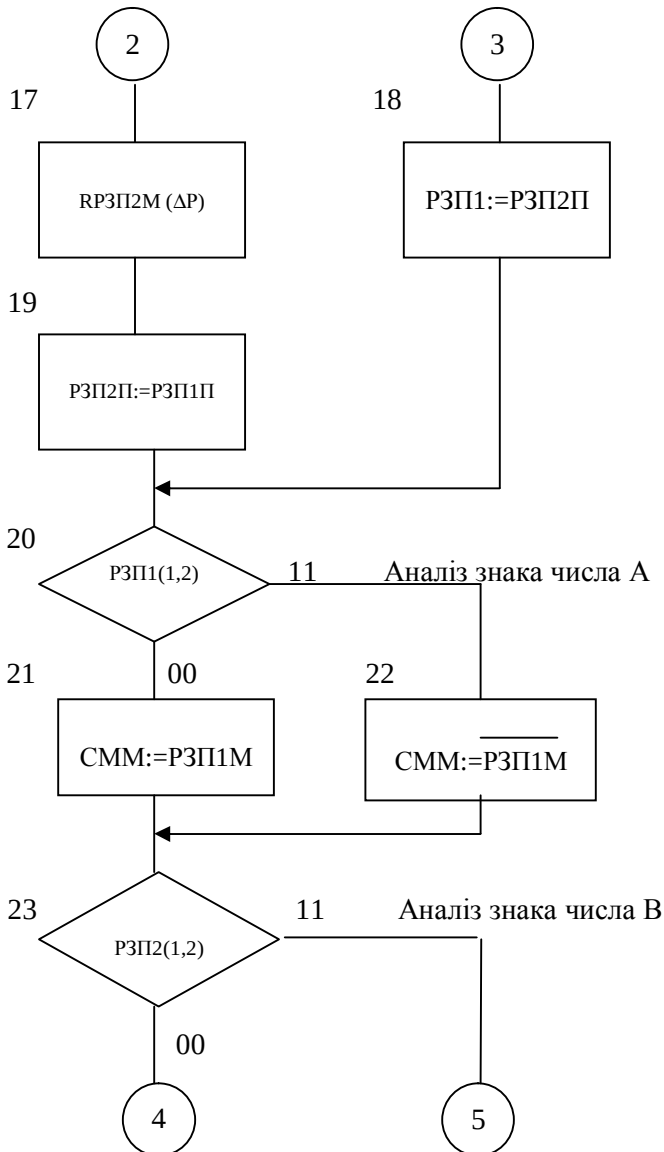
Особливістю суматорів додаткового коду є те, що одиниця переповнення старшого знакового розряду не береться до уваги і вона пропадає. Числа модифіковані, формат з рухомою крапкою.

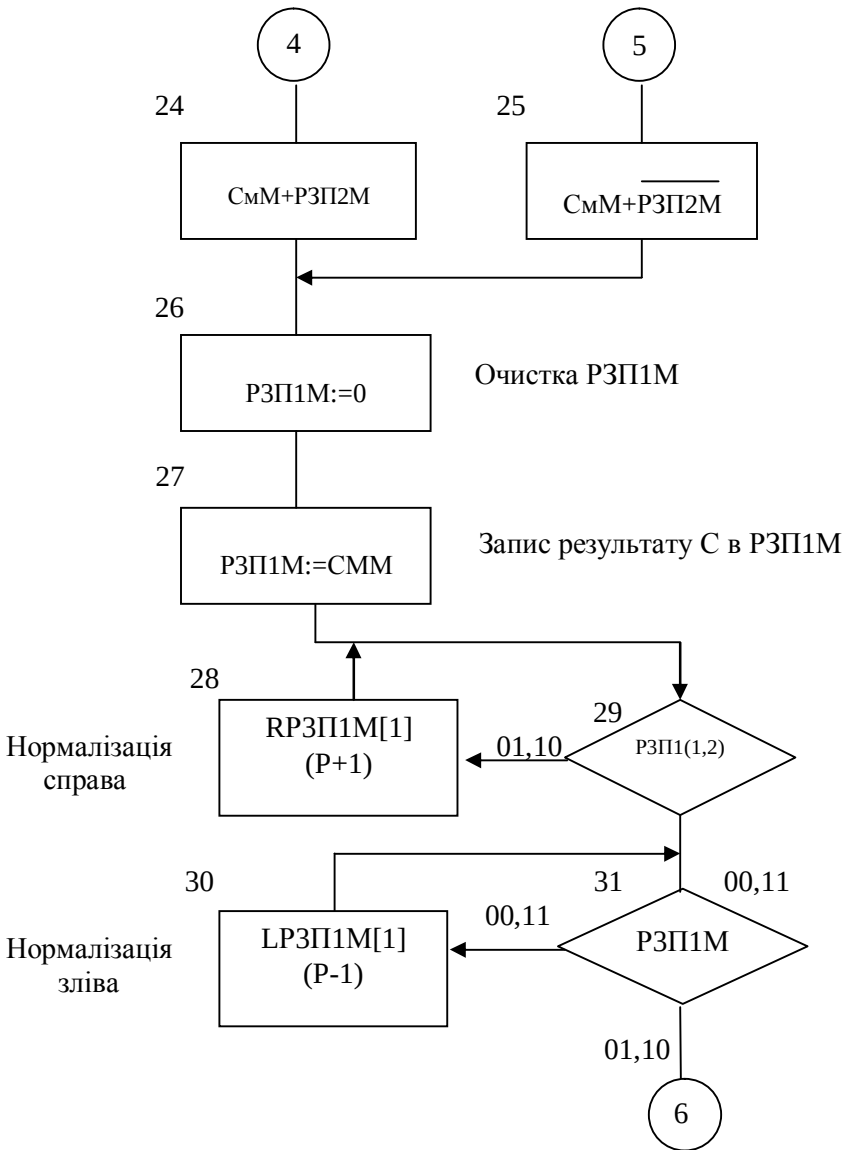
Опис алгоритму:

1. Встановити регістри збереження чисел А та В, суматор додаткового коду в стан 0;
 2. Занести в РЗП1 число А;
 3. Занести в РЗП2 число В;
 4. Провести вирівнювання порядків чисел А та В за допомогою зсуву вправо меншого;
 5. Провести додавання чисел А та В для чого:
 - визначити знак мантиси числа А :
 - якщо 00, то записати А в СМДК в прямому коді,
 - якщо 11, то в додатковому;
 - визначити знак мантиси числа В
 - якщо 00, то додати до СМДК в прямому коді,
 - якщо 11, то в додатковому;
 6. Визначити знак мантиси в суматорі СМДК;
 - якщо 00, то результат позитивний, код прямий, перейти до п.8;
 - якщо 11, то результат негативний, код додатковий, перейти до п.8;
 - якщо 01 або 10- було переповнення, перейти до п. 7;
 7. Зробити правий зсув числа на один розряд з корекцією значення порядку на плюс "1" в молодшому розряді, перейти до п.6;
 8. Перевірити відсутність порушення нормалізації числа. При 00.0х...х, або 11.1х...х зробити лівий зсув числа на один розряд з корекцією порядку на мінус "1" в молодшому розряді та знов перевірити нормалізацію числа. Якщо порушень немає, то перейти до п. 9;
 9. Зчитати результат суми, згідно з визначенням по п.6. до пам'яті ОЗУ;
 10. Кінець.
- Блок схема цього алгоритму приведена на рис. 6.4.









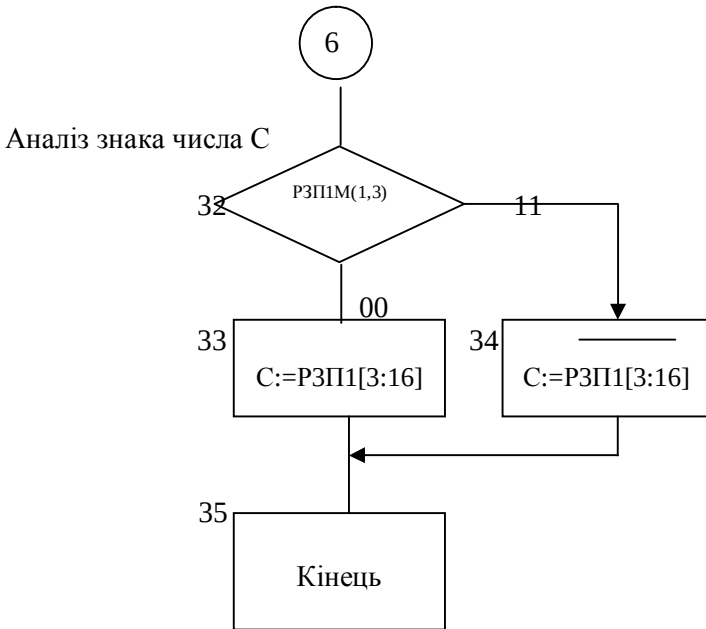


Рисунок 6.4 – Блок схема алгоритму

6.4 Побудова операційної схеми

Базуючись на подібних структурних рішеннях, визначаємо операційний устрій виконання цього алгоритму.

По-перше визначаємо необхідні вузли, блоки потрібні для тимчасового прийому, зберігання чисел А та В та їх додавання, визначаємо їх технічні характерні особливості та необхідні управляючі сигнали виконання мікрооперацій.

Для побудови мікропрограми визначається дія, яка повинна виконуватись в устрої згідно з алгоритмом операції.

Для реалізації цієї дії до схеми можуть вводитись додаткові устрої, аналізу розрядів, дешифратори, лічильники тактів, або інше. Ці додаткові, допоміжні устрої доповнюють операційну схему.

Синтез операційних схем проводять по критерію мінімальної кількості апаратних витрат при мінімальному часі виконання операцій.

Приклад побудови операційної схеми, виконання алгоритму наведено вище.

Усім блокам операційної схеми присвоюють умовні позначки (імена) порядкові номери: суматор зворотного коду (СЗК), суматор (См), суматор мантис (СмМ), суматор порядку (СмП), дешифратор адреси (ДшА), дешифратор команд (ДшК), регістри загального призначення (приймання, віддавання даних, зберігання, зсув) – РЗВ, лічильник (Л), генератор тактових імпульсів (ГТІ), синхронізуючий імпульс (СХІ), аналізатор знака мантиси (АЗМ), аналізатор знака порядку (АЗП).

Крім загальних визначень блоків операційної схеми, загально визначимо операнди (числа), які приймають участь в операціях та діях над ними: перший доданок А, другий доданок В, оператор присвоєння :=, (Наприклад, дію “записати в РЗП1 число А в оберненому коді запишеться у вигляді: $\text{РЗП1} := \text{Аоб}$. Запис $\text{РЗП} := 0$ означає установку регістра РЗП2 в нульове положення), алгебраїчну суму чисел А та В зазначимо С, зсув вправо R(k, PОН) , зсув вліво L(n, РЗП2) , додатковий та обернений код зазначимо ДК и ОК.

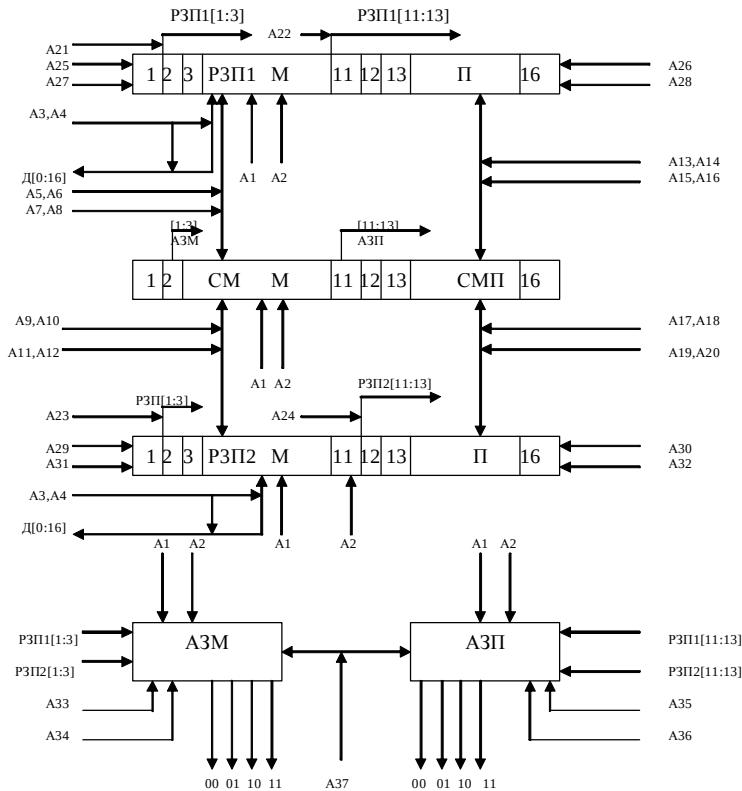


Рисунок 6.5 – Операційна схема ЦА

- A1 - сигнал встановлення в нуль;
- A2 - сигнал синхронізації;
- A3 - Запис даних у прямому коді в РЗП1 або РЗП2;
- A4 - зчитування даних у прямому коді з РЗП1 або РЗП2;
- A5 - запис до суматора мантиси з РЗП1 у прямому коді;
- A6 - зчитування з суматора мантиси в РЗП1 у прямому коді;
- A7 - запис до суматора мантиси з РЗП1 в оберненому коді;
- A8 - зчитування з суматора мантиси в РЗП1 в оберненому коді;
- A9 - запис до суматора мантиси з РЗП2 в прямому коді;
- A10 - зчитування з суматора мантиси в РЗП2 в прямому коді;
- A11 - запис до суматора мантиси з РЗП2 в оберненому коді;
- A12 - зчитування з суматора мантиси в РЗП2 в оберненому коді;

- A13 - запис до суматора порядку з РЗП1 в прямому коді;
- A14 - зчитування з суматора порядку до РЗП1 в прямому коді;
- A15 - запис до суматора порядку з РЗП1 в оберненому коді;
- A16 - зчитування з суматора порядку в РЗП1 в оберненому коді;
- A17 - запис до суматора порядку РЗП1 в прямому коді;
- A18 - зчитування з суматора порядку РЗП1 в прямому коді;
- A19 - запис до суматора порядку РЗП2 в оберненому коді;
- A20 - зчитування з суматора порядку РЗП2 в оберненому оберненому коді;
- A21 - зчитування з РЗП1 розрядів [1:3] до аналізатору знака;
- A22 - зчитування з РЗП1 розрядів [11:13] до аналізатору порядку;
- A23 - зчитування з РЗП2 розрядів [1:3] до аналізатору знака;
- A24 - зчитування з РЗП2 розрядів [11:13] до аналізатору порядку;
- A25 - лівий зсув мантиси РЗП1 з записом до 10 розряду 0 або 1;
- A26 - лівий зсув мантиси РЗП1 з записом до 16 розряду 0 або 1;
- A27 - правий зсув мантиси РЗП1 з записом до 1 розряду 0 або 1;
- A28 - правий зсув мантиси РЗП1 з записом до 11 розряду 0 або 1;
- A29 - лівий зсув мантиси РЗП2 з записом до 10 розряду 0 або 1;
- A30 - лівий зсув мантиси РЗП2 з записом до 16 розряду 0 або 1;
- A31 - правий зсув мантиси РЗП2 з записом до 1 розряду 0 або 1;
- A32 - правий зсув мантиси РЗП2 з записом до 11 розряду 0 або 1;
- A33 - аналіз розрядів знака мантиси РЗП1;
- A34 - аналіз розрядів знака мантиси РЗП2;
- A35 - аналіз розрядів знака порядку РЗП1;
- A36 - аналіз розрядів знака порядку РЗП2;
- A37 - аналіз переповнення.

Ідентифікатори:

- РЗП1[1:3] – регістр загального призначення №1, розряди – 1,2,3.
- СМ – суматор;
- АЗМ – аналізатор знака мантиси;
- АЗП аналізатор знака порядку.

6.5 Аналіз і синтез операційних вузлів

Розглянемо деякі основні операційні пристрої, необхідні для виконання арифметичних дій

6.5.1 Регістр загального призначення

Регістр – це упорядкована п-розрядна сукупність запам'ятовуючих елементів (звичайно тригер), яка призначена для прийому або передачі п-розрядного двійкового слова, його тимчасового зберігання, або виконання логічних операцій по розрядах, зсув слова вліво (вправо), перетворення послідовного коду в паралельний та інше.

Звичайно, для виконання арифметичних дій в АЛУ застосовується декілька регістрів загального призначення.

На операційній схемі регістр зображається прямокутником з обозначенням РЗП1 або Rg1, позначається його розрядність, за звичаєм ліворуч “0” – старший розряд, праворуч “31” – молодший розряд. Код регістра знімають по вихідній шині РЗП1 {0:31} за допомогою операційного сигналу “читання” або “зчитування”. Запис числа А {0:31} в регістр виконується мікрооперацією := “надати значення” і її можна зазначити так: РЗП1:=А, або РЗП1{0:31}:=А{0:31}. У регістр загального призначення записати число А.

Вихідні та вхідні шини бувають однофазними, або двофазними. Кожен розряд регістра має два виходи – вихід прямого коду та вихід оберненого коду. Кожна мікрооперація збуджується та виконується по сигналу мікрокоманди А. Наприклад, мікрокоманда А1 установлювання регістра РЗП1 в “0”, тобто вихідний стан, визначається як А1:РЗП1:=0. Взагалі, для регістра характерні такі мікрокоманди:

А1:=РЗП1:=0 – установити в “0” РЗП1;

А2:РЗП1{0:31}:=А{0:31} – записати число А в РЗП1;

А3:См{0:31}:=РЗП1{0:31} – переписати число, яке зберігається в См до РЗП1;

А4:РЗП1{0:31}:=См{0:31} – переписати число, яке зберігається в См до РЗП1;

А5:РЗП1{0:31}:=R(1,РЗП1) – зсув праворуч РЗП1 на 1 розряд;

А6:РЗП1{0:31}:=L(1,РЗП1) – зсув ліворуч РЗП1 на 1 розряд;

А7:См{0:31}:=РЗП1(1,РЗП1) – переписати число, яке зберігається в РЗП1, в оберненому коді до суматора; Послідовність таких мікрокоманд, виконання мікрооперацій, складає мікропрограму арифметичної дії операційного устрою і яка повинна забезпечуватись логічною схемою.

Для будовання РЗП наряду з базисними логічними елементами, потрібен простіший автомат Мура з двома стійкими станами Т, який має повну систему переходів та виходів.

Простіший автомат Т може бути здійснений RS-тригером на логічних елемента І-НІ (штрих Шеффера), АБО-НІ (стрілка Пірса). Такий тригер являє собою базовий елемент побудови ЦУ. Входи тригера R та S – інформаційні, виходи: Q – прямий, $\bar{Q}$ з рискою – інверсний.

RS – тригер на елементах Пірса.

Таблиця 6.1 – Таблиця істинності

S_{t+1}	R_{t+1}	Q_t	Q_{t+1}	Режим
0	0	0	0	Збереження інформації
		1	1	
0	1	0	0	Установка в "0"
		1		
1	0	0	1	Установка в "1"
		1		
1	1	0	-	Заборонено
		1		

Закон функціонування:

$$Q_{t+1} = S_t + \bar{R}_t \cdot Q_t = \overline{S_t \downarrow (\bar{R}_t \cdot Q_t)} = \overline{S_t \downarrow (R_t \downarrow \bar{Q}_t)};$$

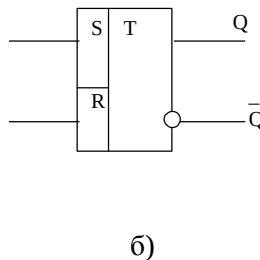
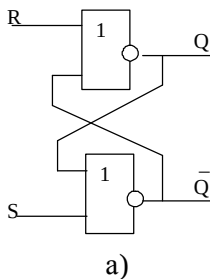


Рисунок 6.6 – а) логічна схема на АБО-НІ; б) графічне позначення RS-тригера;

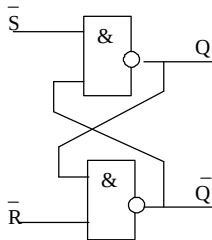
RS-тригер на елементах Шеффера.

Таблиця 6.2 – Таблиця істинності

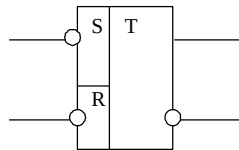
$\bar{S}_{t+1}$	$\bar{R}_{t+1}$	Q_t	Q_{t+1}	Режим
0	0	0	-	Заперечення
		1		
0	1	0	1	Установка в "1"
		1		
1	0	0	0	Установка в "0"
		1		
1	1	0	0	Збереження інформації
		1	1	

Закон функціонування:

$$Q_{t+1} = \overline{\bar{S}_t \cdot (R_t + \bar{Q}_t)} = \bar{S}_t / (R_t + \bar{Q}_t) = \bar{S}_t \cdot (\bar{\bar{R}_t} + Q_t) = \bar{S}_t \cdot (\bar{R}_t + Q_t);$$



а)

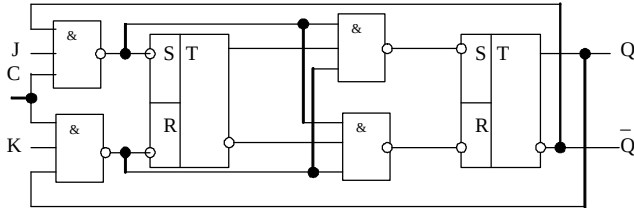


б)

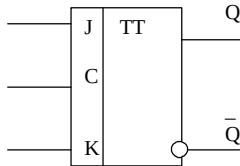
а) логічна схема на I-НІ; б) графічне позначення RS-тригера

Рисунок 6.7 – RS-тригер

На основі RS-тригера будемо універсальний двоступінчатий синхронний JK-тригер, який широко застосовується, при будівництві паралельних регістрів із зсувом та накопичувальних суматорів.



а)



б)

а) логічна схема; б) умовне позначення

Рисунок 6.8 – JK- тригер (а), умовне позначення (б)

JK- тригер працює в два етапи. При $C=1$ переключається перша ступень у відповідності з сигналами J та K. При $C=0$ переключається друга ступень.

Таблиця 6.3 - Таблиця істинності JK- тригера

J_{t+1}	K_{t+1}	C_{t+1}	Q_t	Q_{t+1}	Режим
-	-	0	Q_t	Q_t	Збереження інформації
0	0	1	Q_t	Q_t	Збереження інформації
0	1	1	Q_t	0	Установка в "0"
1	0	1	Q_t	1	Установка в "1"
1	1	1	Q_t	$\bar{Q}_t$	Невизначене

З'єднавши інформаційні входи J та K через інвертор отримаємо динамічний D-тригер.

Маючи базовий елемент пам'яті типа JK-тригер або D-тригер будемо реєстр загального призначення забезпечуємо здвиг вправо, вліво, передачу та отримання інформації за допомогою комбінаційних цифрових пристроїв.

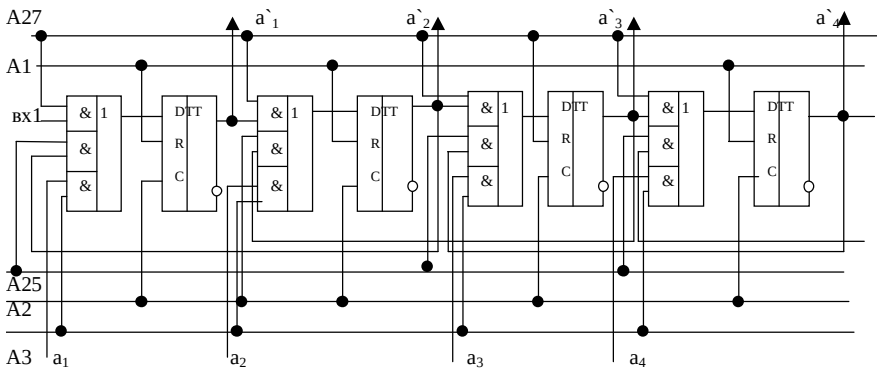


Рисунок 6.9 - Реєстр загального призначення

6.5.2 Накопичувальний суматор паралельної дії

Суматор – це упорядкований n -розрядний цифровий устрій виконуючий порозрядне додавання чисел з умовою перенесення переповнення з молодшого розряду до старшого.

Суматор може будуватись як комбінаційний устрій, який містить в собі логічні схеми для побудови окремих розрядів. Такі суматори мають назву комбінаційні. Синтез одного розряду такого комбінаційного суматора розглядався раніше.

За допомогою багаторозрядного комбінаційного суматора виконують лише мікрооперації типу $C := A + B$, де обидва числа позитивні або від'ємні.

Виконання операцій алгебраїчного додавання, множення та ділення полягає в послідовності мікрооперацій додавання, зсуву, інвертування. Для цих операцій необхідні елементи пам'яті – тригери з лічильним входом. Такі суматори іменують як накопичувальні. В залежності від засобу вводу кодів доданків, розрізняють суматори послідовної та паралельної дії. У суматорів паралельної дії швидкість

виконання арифметичних операцій значно вища. Для складання від'ємних чисел такі суматори використовують крім прямих ще спеціальні коди – обернені або додаткові. Тоді, операція віднімання замінюється операцією додавання оберненого або додаткового коду від'ємного числа:

$$A-B=A+(-B)=A+B_{\text{оберн}}=A+B_{\text{додатк}}$$

Як основний елемент кожного розряду накопичувального двійкового суматора оберненого коду (ДСОК) або додаткового коду (ДСДК) може використовуватись JK або D-тригер з лічильним входом.

Структурна схема такого суматора на n -розрядів приведена на рисунку 6.10.

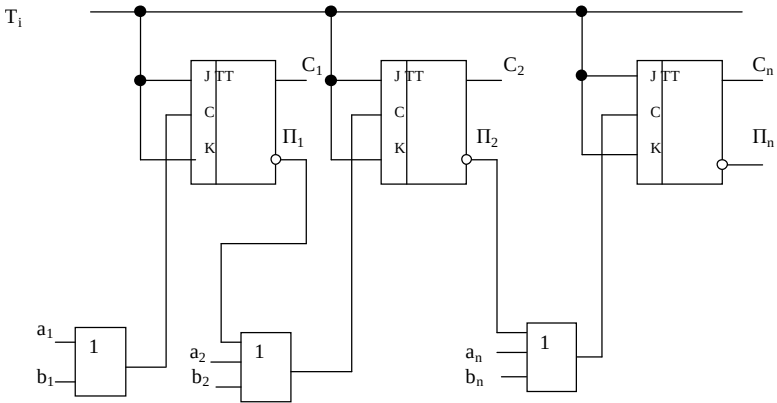


Рисунок 6.10 – Накопичувальний суматор паралельної дії

На рисунку 6.10 відсутні логічні елементи організації переповнення зі старшого до молодшого розряду для суматорів оберненого коду.

Додавання двійкових чисел $A=a_1a_2...a_n$ та $B=b_1b_2...b_n$ виконується за три такти Т. В першому такті на всіх синхронізованих входах С присутні нулі тому з приходом Т суматор стане в "0". В другому такті до суматору записується перший доданок $a_1a_2...a_n$. В третьому такті лічильний вхід тригерів С прийде код доданку $b_1b_2...b_n$. Якщо тригер і-го розряду знаходиться в стані 1, то на його вхід прийде сигнал $b_i=1$, то тригер перейде в стан 0, а на інверсному виході сформується по задньому фронту С сигнал переносу Π_i в старший розряд. Сигнал

переносу буде послідовно переходити по розрядам до тригера, який знаходиться в стані 0. Після закінчення переносів суматор установиться в стан суми чисел $S_m := A + B$.

Прямий код суми зчитувати з виходів $C_1 C_2 \dots C_n$, а обернений – з виходів $P_1 P_2 \dots P_n$.

Комбінаційні схеми синтезувати згідно з таблицями п.3 Завдання.

Послідовність синтезу КС приведена в розділі 4. Пошук МДНФ виконувати за всіма методами мінімізації, приведеними в цьому розділі.

ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Савельев А.Я. Арифметические и логические основы цифровых автоматов / А.Я. Савельев — М: Высшая школа, 1980. — 255 с.
2. Бабич Н.П. Компьютерная схемотехника. Учебное пособие / Н.П. Бабич, И.А. Жуков — К.: "МК-Пресс", 2004. — 576 с, ил.
3. Майоров С.А. Проектирование цифровых вычислительных машин / С.А. Майоров и др. — М: Высшая школа. - 1972 г.
4. Евреинов Э.В. Цифровая и вычислительная техника / Э.В. Евреинов, - М: Радио и связь. - 1991 г.
5. Муттер В.М. Электронные логические устройства автоматики и телемеханики / В.М. Муттер. - Ленинград: СЗПИ. - 1985.