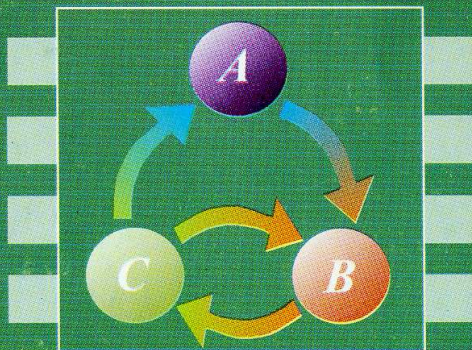


ВТ-145

В.І. Жабін, В.В. Ткаченко

# ЦИФРОВІ АВТОМАТИ



"ВЕК"

**ББК 84Р7**  
**Ж12**  
**УДК 681.3**

*Відповідальний за випуск Стіренко С.Г.*

Рецензент – Тарасенко В.П., доктор технічних наук, професор, завідувач кафедру спеціалізованих комп'ютерних систем Національного технічного університету України "КПІ";

Жабін В.І., Ткаченко В.В.

**Ж12** Цифрові автомати. Практикум / Жабін В.І. и др., – К.: ВЕК +, 2004. – 160 с., ил.

**ISBN 966–7140–11–3**

Розглядаються прикладні питання теорії цифрових автоматів, методи синтезу логічних схем в заданому елементному базисі, способи побудови типових вузлів та пристроїв ЕОМ.

Навчальний посібник призначено для проведення лабораторних занять і організації курсового проектування зі студентами спеціальності “Комп'ютерна інженерія”.

**ББК 84Р7**

**ISBN 966–7140–11–3**

© Жабін В.І., Ткаченко В.В., 2004

## ВВЕДЕННЯ

Дисципліни "Прикладна теорія цифрових автоматів" і "Комп'ютерна схемотехніка" є базовими для студентів спеціальності "Комп'ютерна інженерія". Виконання лабораторних робіт дозволяє розширити і закріпити теоретичні знання з курсу, опанувати навички проектування, налагодження і дослідження вузлів і пристроїв цифрових ЕОМ.

Кожній лабораторній роботі повинна передувати самостійна підготовка студентів, в процесі якої вони докладно вивчають опис лабораторної роботи, відповідний розділ конспекту лекцій і літературні джерела. В процесі підготовки складається звіт про лабораторну роботу, в якому повинні бути відображені всі пункти теоретичного завдання, а також заготовлені необхідні для експериментальної частини лабораторної роботи таблиці, осі для часових діаграм і т. ін. Схеми необхідно вичерчувати чітко, дотримуючи правил діючих стандартів.

Перед початком лабораторної роботи результати підготовки перевіряються викладачем. При цьому студент повинний сформулювати ціль і порядок виконання лабораторної роботи, представити заготовлений звіт і відповісти на контрольні питання. Непідготовлений студент до лабораторної роботи не допускається.

Перед початком наступного заняття в лабораторії студент повинний представити викладачеві цілком оформлений звіт по попередній роботі. Звіт повинний містити короткі теоретичні відомості, необхідні для виконання завдання, відповіді на контрольні питання, усі схеми, формули, таблиці, діаграми, графіки, отримані при виконанні завдання та в процесі експериментального дослідження схем, а також висновки по роботі. Студент, який не представив звіт, не допускається до виконання наступної роботи. Залік по лабораторній роботі студент одержує після сівбесіди по тематиці виконаної роботи.

Для проведення лабораторних робіт використовуються програмні засоби моделювання цифрових систем, які розроблені на кафедрі обчислювальної техніки НТУУ "КПІ" за участю студентів. В розробці програм моделювання ЕОМ приймали участь Р.Л.Антонов, О.Л.Брагинський, Д.В.Хабенко. Програми моделювання логічних схем розробили Ю.В.Нечаєв, Г.Г.Салтищак і В.П.Щурко.

Автори вдячні В.А.Доріченко за допомогу в підготовці рукопису до видання.

# 1. ПРОГРАМНИЙ КОМПЛЕКС ДЛЯ МОДЕЛЮВАННЯ ЛОГІЧНИХ СХЕМ

Програмний комплекс призначений для моделювання процесів у комбінаційних і послідовних схемах. Він дозволяє створювати і редагувати логічні схеми, здійснювати моделювання у синхронному (без урахування затримок сигналів в елементах схеми) і в асинхронному (з урахуванням затримок) режимах, а також зберігати отримані моделі у вигляді файлів на дисках.

Вигляд моделі при застосуванні ПРОГМОЛС 2.0 проілюстровано на рис. 1.1. Комплекс включає систему підказок, що полегшує роботу в різних режимах моделювання. Для роботи з програмою використовують систему ієрархічних меню.

Меню містить наступні розділи:

- файл;
- виправлення;
- проект;
- моделювання;
- інструменти;
- вікно;
- допомога.

Побудова та елементи керування для кожного розділу пояснюються нижче.

## Файл



**Створити** (сполучення клавіш **Ctrl+N**). Створює новий файл проекту.



**Відкрити** (**Ctrl+O**). Викликає діалогове вікно відкриття проекту.



**Зберегти** (**Ctrl+S**). Зберігає файл проекту на диск під поточним ім'ям.



**Зберегти як** (**Ctrl+A**). Зберігає файл проекту на диск і запитує ім'я та шлях файлу.

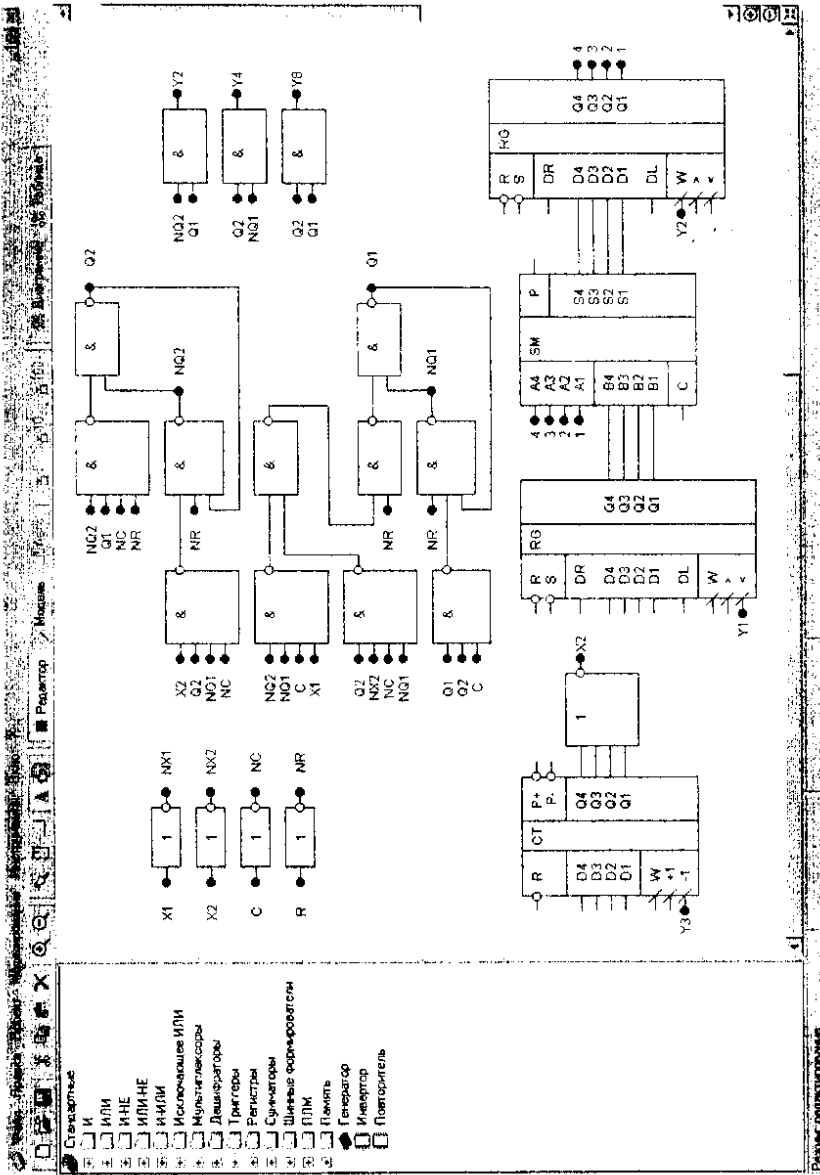


Рис. 1.1. Зображення моделі операційного пристрою



**Закрити (Ctrl+F4).** Закриває поточний файл проекту.



**Вихід (Alt+F4).** Закриває програму і всі вікна.

### Виправлення



**Вирізати (Ctrl+X).** Вирізає фрагмент схеми у внутрішній буфер обміну програми.



**Копіювати (Ctrl+C).** Копіює фрагмент схеми у внутрішній буфер обміну програми.



**Вставити (Ctrl+V).** Вставляє фрагмент схеми з внутрішнього буфера обміну програми.



**Видалити (Delete).** Видаляє фрагмент схеми.

### Проект



**Корпус мікросхеми.** Показати/сховати корпус мікросхеми поточного проекту.



**Редактор корпусу.** Викликає діалогове вікно редактора корпусу мікросхеми поточного проекту.



**Компіляція (Alt+C).** Компілює поточний проект. Дана команда використовується для відновлення списку змінних у діаграмі і таблиці проекту після зміни схеми без включення режиму моделювання. При включенні режиму моделювання компіляція робиться автоматично.



**Додати в бібліотеку.** Копіює поточний проект у буфер обміну редактора бібліотек і викликає редактор бібліотек. Для вставки компонента в бібліотеку необхідно викликати команду **Вставити** редактора бібліотек.



**Настроювання (Ctrl+F4).** Викликає діалогове вікно настроювання проекту.

## Моделювання



**Відробити інтервал.** Відпрацьовує заданий користувачем інтервал модельного часу.

**Відробити до.** Відпрацьовує до моменту модельного часу, заданого користувачем.

**Генератор (G).** У синхронному режимі викликає чергову зміну стану генераторів і відпрацьовує схему до закінчення перехідних процесів. В асинхронному режимі відпрацьовує 1 такт модельного часу.

## Інструменти



**Редактор бібліотек (Alt+L).** Активізує редактор бібліотек.



**Настроювання.** Викликає діалогове вікно настроювань програми.



**Діаграма (Alt+D).** Виводить на екран діаграму станів змінних поточного проекту.



**Таблиця (Alt+T).** Виводить на екран таблицю станів змінних поточного проекту.

## Вікно



**Каскадом.** Розташовує вікна відкритих проектів каскадом.



**Розділити по вертикалі.** Розташовує вікна відкритих проектів по вертикалі таким чином, щоб вони не перекривалися.



**Розділити по горизонталі.** Розташовує вікна відкритих проектів по горизонталі таким чином, щоб вони не перекривалися.



**Збільшити.** Збільшує масштаб у редакторі схеми поточного проекту.



**Зменшити.** Зменшує масштаб у редакторі схеми поточного проекту.

### Допомога

Про програму. Виводить відомості про програму і розроблювачів.



Допомога. Викликає файл довідки.

Вказані нижче елементи керування винесені на окремі панелі.



Створити (Ctrl+N). Створює новий файл проекту.



Відкрити (Ctrl+O). Викликає діалогове вікно відкриття проекту.



Зберегти (Ctrl+S). Зберігає файл проекту на диск під поточним ім'ям.



Вирізати (Ctrl+X). Вирізає фрагмент схеми у внутрішній буфер обміну програми.



Копіювати (Ctrl+C). Копіює фрагмент схеми у внутрішній буфер обміну програми.



Вставити (Ctrl+V). Вставляє фрагмент схеми з внутрішнього буфера обміну програми.



Видалити (Delete). Видаляє фрагмент схеми.



Збільшити. Збільшує масштаб у редакторі схеми поточного проекту.



Зменшити. Зменшує масштаб у редакторі схеми поточного проекту.












Виділити. Дозволяє виділити і перетягнути лівою кнопкою миші фрагмент схеми, а також інвертувати виходи та входи елементу подвійним натисканням.



Елемент. Дозволяє вставити новий елемент у схему.



-  **Лінія.** Дозволяє провести зв'язок у схемі.
-  **Перемінна.** Дозволяє визначити змінну в схемі.
-  **Захоплення.** Дозволяє перетаскувати лівою кнопкою миші весь зміст вікна редактора логічних схем.
-  **Редактор.** Переводить редактор логічних схем у режим редагування.
-  **Модель.** Переводить редактор логічних схем у режим моделювання.
-  **Авто.** У натиснутому стані вкочає режим автоматичного моделювання, а у віджатому – режим покрокового моделювання.
-  **1 такт.** Відпрацьовує 1 такт модельного часу.
-  **10 тактів.** Відпрацьовує 10 тактів модельного часу.
-  **100 тактів.** Відпрацьовує 100 тактів модельного часу.

### ***Панель компонентів***

Панель *компонентів* розташована в лівій частині робочого стола моделі і являє собою ієрархічний список, у якому відображені підключені до програми бібліотеки і їх зміст. Кожна бібліотека містить ієрархічну структуру компонентів, подібну до файлової системи. Панель компонентів призначена для вибору компонента для наступної вставки його в схему.

Для дослідження схем у загальному випадку необхідно виконати послідовність дій.

1. Створити за допомогою редактора логічну схему на екрані дисплея.
2. Позначити на схемі вхідні і вихідні змінні.
3. Створити заголовок (перелічити вхідні і вихідні змінні) і сформулювати послідовність вхідних наборів в таблиці істинності.

4. Задати необхідні величини затримок сигналів для елементів схеми.
5. Встановити початковий стан схеми.
6. Перейти до режиму моделювання схеми.

## 2. Лабораторна робота №1

### ПРОЕКТУВАННЯ КОМБІНАЦІЙНИХ СХЕМ

*Ціль роботи* – вивчити методи побудови комбінаційних схем в заданому елементному базисі, визначення складності і дослідження швидкодії комбінаційних схем.

#### Теоретичні відомості

*Логічний елемент* – це електронна схема, що реалізує певну перемикальну функцію.

Сукупність логічних елементів, призначена для перетворення двійкових змінних, називається *логічною схемою*.

Логічні схеми поділяються на послідовні і комбінаційні.

*Комбінаційною* називається логічна схема, в якій значення вихідних сигналів цілком визначаються значеннями вхідних сигналів, що діють в даний момент часу і не залежать від значень вхідних сигналів, що діяли в попередні моменти часу.

Вважають, що така схема має один стан. Поведінка комбінаційної схеми може бути описана системою перемикальних функцій.

Розрізняють задачі аналізу і синтезу комбінаційних схем.

*Задача аналізу* комбінаційної схеми зводиться до знаходження системи функцій, що відображають логіку роботи цієї схеми.

*Задача синтезу* зворотна задачі аналізу, тобто припускає побудову схеми, використовуючи заданий базис логічних елементів.

Синтез комбінаційної схеми з одним виходом можна розбити на три етапи.

На першому етапі виконують мінімізацію перемикальної функції.

На другому етапі функцію записують у так званій операторній формі, тобто у вигляді суперпозиції операторів заданих логічних елементів.

Оператором логічного елемента називають функцію, що реалізує цей елемент. Якщо у елементів достатньо входів, то одержання операторного запису функції зводиться до її представлення в одній з нормальних форм.

В базисі елементів I, АБО, НЕ, I-НЕ, АБО-НЕ таких форм вісім.

На прикладі функції  $F(X,Y,Z) = \bar{X} \cdot Y \vee X \cdot \bar{Y} \vee \bar{Y} \cdot Z$  і її заперечення  $\overline{F(X,Y,Z)} = X \cdot Y \vee \bar{X} \cdot \bar{Y} \cdot \bar{Z}$ , покажемо одержання всіх нормальних форм.

Позначати нормальні форми будемо з використанням внутрішньої і зовнішньої функцій. Наприклад, у диз'юнктивної нормальної форми (ДНФ)

внутрішньою є функція І, а зовнішньою – АБО, тобто ДНФ – форма типу І/АБО.

Взявши подвійне заперечення заданої функції і застосувавши кілька разів правило де Моргана, послідовно одержимо такі нормальні форми:

$$\begin{aligned}
 F(X,Y,Z) &= \overline{\overline{X \cdot Y \vee X \cdot \overline{Y} \vee \overline{Y} \cdot Z}} && \text{(форма І/АБО);} \\
 &= \overline{\overline{X \cdot Y} \vee \overline{X \cdot \overline{Y}} \vee \overline{Y} \cdot Z} \\
 &= \overline{\overline{X \cdot Y} \cdot \overline{X \cdot \overline{Y}} \cdot \overline{Y} \cdot Z} && \text{(форма І-НЕ/І-НЕ);} \\
 &= \overline{(X \vee \overline{Y}) \cdot (\overline{X} \vee Y) \cdot (Y \vee \overline{Z})} && \text{(форма АБО/І-НЕ);} \\
 &= \overline{(X \vee \overline{Y}) \vee (\overline{X} \vee Y) \vee (Y \vee \overline{Z})} && \text{(форма АБО-НЕ/АБО).}
 \end{aligned}$$

Виходячи з заперечення заданої функції, запишемо ще чотири нормальні форми:

$$\begin{aligned}
 F(X,Y,Z) &= \overline{X \cdot Y \vee \overline{X} \cdot \overline{Y} \cdot \overline{Z}} && \text{(форма І/АБО-НЕ);} \\
 &= \overline{X \cdot Y \cdot \overline{X} \cdot \overline{Y} \cdot \overline{Z}} && \text{(форма І-НЕ/І);} \\
 &= \overline{(\overline{X} \vee \overline{Y}) \cdot (X \vee Y \vee Z)} && \text{(форма АБО/І);} \\
 &= \overline{(\overline{X} \vee \overline{Y}) \cdot (X \vee Y \vee Z)} \\
 &= \overline{(\overline{X} \vee \overline{Y}) \vee (X \vee Y \vee Z)} && \text{(форма АБО-НЕ/АБО-НЕ).}
 \end{aligned}$$

Нормальні форми дозволяють одержати комбінаційну схему з двома рівнями (каскадами) логічних елементів, якщо елементи мають необхідне число входів, а аргументи представлені прямими та інверсними значеннями.

Якщо число входів  $p$  елементів менше, ніж потрібно для реалізації нормальної форми, то для одержання операторної форми змінні поєднують у групи, що містять не більше  $p$  елементів, і використовують співвідношення:

$$\begin{aligned}
 X_1 \cdot X_2 \cdot \dots \cdot X_m &= (X_1 \cdot \dots \cdot X_g) \cdot \dots \cdot (X_s \cdot \dots \cdot X_m); \\
 X_1 \vee X_2 \vee \dots \vee X_m &= (X_1 \vee \dots \vee X_g) \vee \dots \vee (X_s \vee \dots \vee X_m); \\
 \overline{X_1 \cdot X_2 \cdot \dots \cdot X_m} &= \overline{(X_1 \cdot \dots \cdot X_g)} \cdot \dots \cdot \overline{(X_s \cdot \dots \cdot X_m)};
 \end{aligned}$$

$$\overline{X_1 \vee X_2 \vee \dots \vee X_m} = \overline{(X_1 \vee \dots \vee X_g) \vee \dots \vee (X_s \vee \dots \vee X_m)},$$

де  $g \leq p$  і  $m-s+1 \leq p$ .

Число груп змінних також не повинне перевищувати  $p$ . В протилежному випадку зазначені перетворення виконують стосовно груп змінних. Такі перетворення дозволяють представити задану функцію в операторній формі з урахуванням числа входів елементів. Схема, отримана по операторній формі, може містити більше двох рівнів.

На третьому етапі по операторним представленням функцій складають комбінаційну схему. Задана система елементів може дозволити реалізувати кілька операторних представлень функції. Наприклад, при наявності елементів І, АБО та І-НЕ можна використовувати в якості вихідної одну з п'яти нормальних форм (І/АБО, І-НЕ/І-НЕ, АБО/І-НЕ, І-НЕ/І, АБО/І) для одержання відповідних операторних представлень з урахуванням числа входів елементів. Щоб вибрати одну схему з декількох можливих, необхідно порівнювати їх по заданим параметрам (найбільш часто – по складності і швидкодії).

Існує кілька способів оцінки складності схем. Часто використовують оцінку по Квайну ( $K$ ), яка визначається як сумарне число входів усіх логічних елементів. Складність можна також оцінити в числі логічних елементів чи в числі умовних логічних елементів, що визначається по формулі

$$N = \sum_{i=1}^r \frac{m_i \cdot n_i}{g},$$

де  $r$  – число типів елементів;  $m_i, n_i$  – кількість відповідно елементів  $i$ -го типу і входів такого елемента,  $g$  – число входів умовного елемента. В якості умовного елемента в даній роботі використовується елемент з двома входами.

Параметри  $K$  і  $N$  можуть використовуватись при проектуванні інтегральних схем (як порівняна оцінка складності варіантів проектування), тому що складність схем залежить від площі кристала, яка пропорційна числу логічних елементів і числу їх входів.

Швидкодія комбінаційних схем залежить від часових параметрів логічних елементів  $t_{01}$  і  $t_{10}$ , що характеризують затримку сигналів елементів (час переходу вихідного сигналу від одного логічного рівня до іншого). На практиці використовують звичайно усереднене значення часу затримки  $t^* = (t_{01} + t_{10})/2$  чи максимальне –  $t^* = \max(t_{01}, t_{10})$ .

Для комбінаційних схем на однотипних елементах середній час затримки сигналів  $T = Lt$ , де  $L$  – рівень схеми, дорівнює числу елементів, що вхо-

дять в максимальний по довжині ланцюжок елементів. Якщо використовуються елементи з різною затримкою, то в схемі визначається шлях, який вимагає максимального часу поширення сигналів.

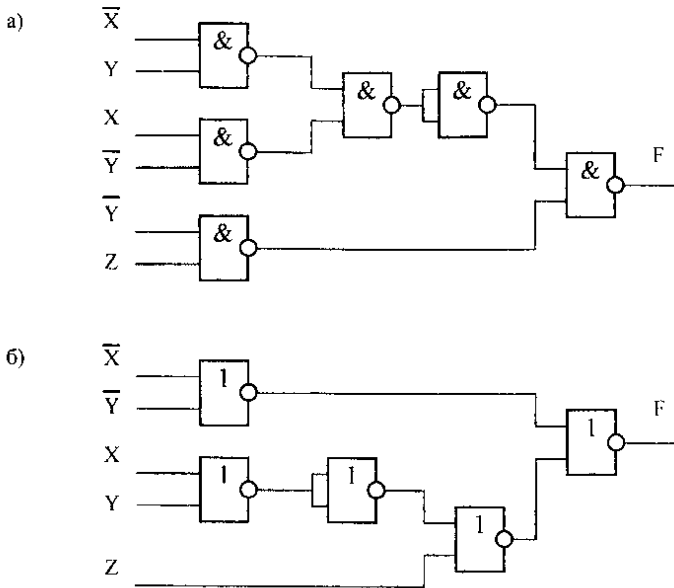


Рис. 2.1

З декількох можливих вибирають комбінаційну схему, що краще інших задовольняє заданим параметрам. Наприклад, при наявності елементів 2І-НЕ та 2АБО-НЕ розглянуту функцію можна представити в операторних формах І-НЕ/І-НЕ та АБО-НЕ/АБО-НЕ таким чином:

$$F(X, Y, Z) = \overline{\overline{\overline{X} \cdot Y} \cdot X \cdot \overline{Y}} \cdot \overline{Y} \cdot Z = \overline{\overline{\overline{X} \cdot Y} \cdot X \cdot \overline{Y} \cdot \overline{Y} \cdot Z};$$

$$F(X, Y, Z) = \overline{\overline{\overline{X} \vee \overline{Y}} \vee (X \vee Y \vee Z)} = \overline{(\overline{\overline{X} \vee \overline{Y}}) \vee (X \vee Y \vee Z)}.$$

Отриманим формам відповідають схеми на рис. 2.1.

Якщо елементи І-НЕ мають менший час затримки сигналів, ніж елементи АБО-НЕ, то схема на рис. 2.1-а більш швидкодіюча, але вона програє другій схемі (рис. 2.1-б) по складності (для першої схеми  $K=12$ , а для другої  $K=10$ ).

## Підготовка до роботи

1. Визначити свій варіант перемикальної функції. Для цього необхідно номер варіанта перевести в двійкову систему числення і записати шість його молодших розрядів у вигляді слова  $h_6 h_5 h_4 h_3 h_2 h_1$ . Значення  $h_i$  підставити в табл.2.1 та табл.2.2. Наприклад, якщо номер варіанта 19 (у двійковій системі 010 011), то  $h_6=0, h_5=1, h_4=0, h_3=0, h_2=1, h_1=1$ .

Таблиця 2.1

$x_3$	$x_2$	$x_1$	$y$
0	0	0	$h_6$
0	0	1	$h_5$
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	$h_4$

Таблиця 2.2

$h_3$	$h_2$	$h_1$	Характеристики елементів	
			Тип	$t$
0	0	0	3І-НЕ	10
			3І	14
0	0	1	4І-НЕ	10
			2АБО	12
0	1	0	4І	14
			2АБО	12
0	1	1	3І	14
			2АБО	12
1	0	0	2АБО-НЕ	12
			4І	14
1	0	1	2І-НЕ	10
			2АБО	12
1	1	0	2АБО-НЕ	10
			3І	14
1	1	1	2І-НЕ	10
			2АБО-НЕ	12

2. Знайти досконали ДНФ функції і її заперечення. Представити функцію у всіх восьми нормальних формах.

3. Одержати операторні представлення функції, що можуть бути реалізовані на елементах, заданих табл. 2.2.

4. Вибрати операторні форми, що забезпечують одержання комбінаційної схеми з максимальною швидкодією і комбінаційною схемою з мінімальним числом умовних елементів, тобто схему з кращим параметром  $T$  і схему з мінімальним значенням  $N$ . Вважати, що умовний елемент має два входи. Побудувати зазначені комбінаційні схеми.

## Порядок виконання роботи

1. Побудувати модель заданої комбінаційної схеми.

2. Перекопатися в правильності функціонування моделі, визначити часові параметри комбінаційної схеми за допомогою часової діаграми.

## Зміст звіту

Звіт повинний містити короткі теоретичні відомості, отримані формули, таблиці, малюнки і висновки за результатами роботи.

## Контрольні питання

1. Сформулювати визначення перемикальної функції, логічного елемента, комбінаційної схеми.
2. Основні властивості комбінаційних схем.
3. У чому сутність задач аналізу і синтезу комбінаційних схем?
4. Охарактеризувати основні етапи синтезу комбінаційних схем.
5. Що таке операторне представлення функції?
6. Як визначити складність і швидкодію комбінаційних схем?
7. Чим пояснюється можливість виникнення збоїв комбінаційних схем при збільшенні частоти подачі змінних на їх входи?

## Література

1. Поспелов Д.А. Логические методы анализа и синтеза схем. – М.: Энергия, 1974. – 367 с.
2. Самофалов К.Г., Романкевич А.М., Валуйский В.Н., Каневский Ю.С., Пиневиц М.М. Прикладная теория цифровых автоматов.– К.: Вища школа, 1987. – 375 с.
3. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум. – К.: Вища школа, 1990. – 215 с.



### 3. Лабораторна робота №2

## МІНІМІЗАЦІЯ ПЕРЕМИКАЛЬНИХ ФУНКЦІЙ

*Ціль роботи* – вивчення методів мінімізації перемикальних функцій, знаходження операторних форм перемикальних функцій, побудова та дослідження параметрів логічних схем.

### Теоретичні відомості

Функції  $f$  і  $\varphi$  називаються *еквівалентними*, якщо вони приймають однакові значення на всіх наборах аргументів.

Еквівалентні функції можуть відрізнятися формами представлення і ціною. Під *ціною* перемикальної функції розуміється кількість букв, що входять в її запис.

Проблема мінімізації зводиться до відшукування форми представлення функції з мінімальною ціною. Мінімізація дозволяє спростити схеми, що реалізують перемикальні функції.

В роботі методи мінімізації розглядаються щодо диз'юнктивних форм представлення функцій.

### Метод мінімізації Квайна

Вихідною формою представлення функції для мінімізації по методу Квайна є досконала диз'юнктивна нормальна форма (ДДНФ).

Метод забезпечує одержання скороченої ДНФ, тобто сукупності всіх простих імплікант.

Метод базується на використанні співвідношення неповного склеювання

$$Ax \vee \bar{A}x = Ax \vee \bar{A}x \vee A$$

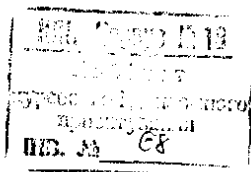
і співвідношення поглинання

$$BA \vee A = A,$$

де  $A$  і  $B$  – довільні кон'юнктивні терми,  $x$  – змінна.

### Етапи мінімізації:

1) запис функції у вихідній формі – ДДНФ;



2) застосування співвідношення склеювання послідовно до конститuent одиниці, потім до імплікант  $n-1$  рангу,  $n-2$  рангу і так далі, поки можливе формування нових імплікант;

3) виконання всіх можливих поглинань, в результаті чого визначаються всі прості імпліканти;

4) побудова таблиці покриття (імплікантної матриці) і знаходження тупикових ДНФ (ТДНФ);

5) вибір мінімальної ДНФ (МДНФ) з числа ТДНФ.

*Приклад.* Виконати мінімізацію функції, заданої табл. 3.1.

Таблиця 3.1

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

Представимо функцію в ДДНФ

$$y = \overline{x_3}x_2x_1 \vee \overline{x_3}x_2\overline{x_1} \vee \overline{x_3}x_2x_1 \vee \overline{x_3}x_2\overline{x_1} \vee \overline{x_3}x_2x_1.$$

Виконавши попарне склеювання конститuent одиниці, одержуємо множину імплікант 2-го рангу:

$$\overline{x_3}x_1, \overline{x_2}x_1, \overline{x_3}x_2, \overline{x_2}x_1.$$

Подальше склеювання імплікант неможливе.

Тоді функцію можна записати у вигляді

$$y = \overline{x_3}x_2x_1 \vee \overline{x_3}x_2\overline{x_1} \vee \overline{x_3}x_2x_1 \vee \overline{x_3}x_2\overline{x_1} \vee \overline{x_3}x_2x_1 \vee \overline{x_3}x_1 \vee \overline{x_2}x_1 \vee \overline{x_3}x_2 \vee \overline{x_2}x_1.$$

Виконавши поглинання, одержуємо скорочену ДНФ

$$y = \overline{x_3}x_1 \vee \overline{x_2}x_1 \vee \overline{x_3}x_2 \vee \overline{x_2}x_1$$

Будуємо таблицю покриття (табл. 3.2).

Таблиця 3.2

Імпліканти	Конститuentи				
	$\overline{x_3}x_2x_1$	$\overline{x_3}x_2\overline{x_1}$	$\overline{x_3}x_2x_1$	$\overline{x_3}x_2\overline{x_1}$	$\overline{x_3}x_2x_1$
$\overline{x_3}x_1$	⊙		⊙		
$\overline{x_2}x_1$	∨			∨	
$\overline{x_3}x_2$		∨	∨		
$\overline{x_2}x_1$		⊙			⊙

Знаходимо ядро функції – сукупність імплікант відповідних однократ-  
но покритим конститuentам. В даному випадку ядро складають імпліканти  
 $\overline{x_2 x_1}$  і  $x_2 x_1$ .

Як правило, ядро варто доповнити ще декількома імплікантами, для  
одержання повного покриття всіх конститuent вихідної функції. Різні варіан-  
ти покриття є туліковими ДНФ. Серед ТДНФ форма з мінімальною ціною  
буде мінімальної ДНФ (МДНФ).

Для розглянутої функції, існують дві рівноцінні ТДНФ:

$$y = \overline{x_3 x_1} \vee \overline{x_2 x_1} \vee x_2 x_1 ;$$

$$y = \overline{x_2 x_1} \vee \overline{x_3 x_2} \vee x_2 x_1 .$$

В якості МДНФ вибираємо, наприклад,

$$y = \overline{x_3 x_1} \vee \overline{x_2 x_1} \vee x_2 x_1 .$$

### Метод мінімізації Квайна-Мак-Класки

Метод Квайна-Мак-Класки є модифікацією методу Квайна. Він ґрун-  
тується на співвідношеннях неповного склеювання і поглинання, як і метод  
Квайна. Особливістю методу є використання цифрової форми запису пере-  
микальних функцій. В цьому випадку зменшується число символів для  
представлення термів і число операцій в процесі мінімізації, що робить  
метод зручним при програмній реалізації.

Якщо використовувати геометричну інтерпретацію представлення пе-  
ремикальних функцій, то кожен набір аргументів є  $n$ -мірним вектором ( $n$  –  
число аргументів) і визначає точку  $n$ -мірного простору. Сукупність усіх  
наборів представляє  $n$ -мірний куб. Конститuentам відповідають вершини  
куба, а імплікантам – ребра і грані. Кожній конкретній функції відповідає  
певне просторове представлення.

Наприклад, для функції 3-х змінних (рис. 3.1) конститuentам відпові-  
дають вершини 3-мірного куба, а імплікантам – ребра і грані. Терми  $n$ -рангу  
називають 0-кубами, терми  $(n-1)$ -го рангу – 1-кубами,  $(n-2)$ -рангу – 2-  
кубами і т.д.

Символом  $X$  в  $r$ -кубах відзначаються аргументи, по яким склеюються  
 $(r-1)$ -куби. Множина кубів  $i$ -го рангу утворюють комплекс  $K^i$ .

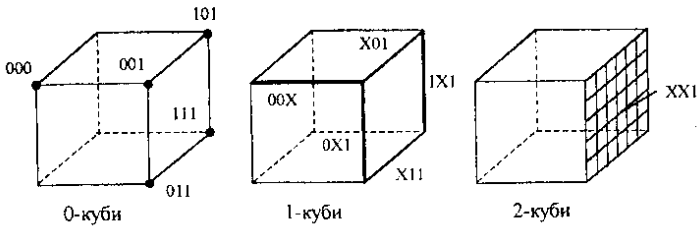


Рис. 3.1

Етапи мінімізації

1. Для функції виписують комплекс 0-кубів ( $K^0$ ). Набори упорядковуються по кількості одиниць. Одержують групи без одиниць, з однією одиницею, із двома і т.д. В цьому випадку склеювання можливе тільки між сусідніми групами кубів.

2. Шляхом склеювання формують 1-куби, 2-куби і т.д., поки можливе склеювання. Кожен куб упорядковується аналогічно 0-кубу. При цьому в одну групу повинні входити куби, що мають не тільки однакове число одиниць, але й залежать від тих самих змінних.

3 Шляхом поглинання формується покриття Z, що відповідає скороченій ДНФ.

4 Будується матриця покриття, з якої визначають усі ТДНФ.

5 Серед ТДНФ відшукується МДНФ.

Таблиця 3.3

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

*Приклад.* Методом Квайна-Мак-Класки виконати мінімізацію функції, заданої табл. 3.3.

Виходячи з таблиці істинності функції, запишемо конституенти, поєднуючи їх у групи по кількості одиниць ( $K^0$ ). Виконуючи склеювання, формуємо куби  $K^1$  і  $K^2$ . Після виконання поглинань, одержуємо Z-покриття.

$$K^0 = \begin{matrix} \frac{000}{-} \\ 010 \\ \frac{100}{-} \\ 011 \\ 101 \\ 110 \end{matrix}; \quad K^1 = \begin{matrix} \frac{X00}{-} \\ 0X0 \\ \frac{1X0}{-} \\ 01X \\ 10X \end{matrix}; \quad K^2 = \{XX0\}; \quad Z = \begin{matrix} 01X \\ 10X \\ XX0 \end{matrix}$$

Будуємо таблицю покриття (табл. 3.4).

Таблиця 3.4

Імпліканта	Конституента					
	000	010	100	011	101	110
01X		✓		✓		
10X			✓		✓	
XX0	✓	✓	✓			✓

В даному випадку всі імпліканти входять в ядро функції. Отже, МДНФ має вигляд

$$y = \overline{x_3}x_2 \vee x_3\overline{x_2} \vee \overline{x_1}.$$

### Метод невизначених коефіцієнтів

Будь-яку функцію можна представити у вигляді диз'юнкції всіх конституент і всіх можливих імплікант, помножених на відповідний коефіцієнт, що може приймати значення 0 або 1. (Метод може бути використаний у будь-якій алгебрі перемикальних функцій. Перетерплюють зміни тільки вихідні канонічні форми запису функцій і системи рівнянь для знаходження коефіцієнтів).

Наприклад, при  $n=2$  можна записати

$$y = k_2^1 x_2 \vee k_2^0 \overline{x_2} \vee k_1^1 x_1 \vee k_1^0 \overline{x_1} \vee k_{21}^{00} \overline{x_2} \overline{x_1} \vee k_{21}^{01} \overline{x_2} x_1 \vee k_{21}^{10} x_2 \overline{x_1} \vee k_{21}^{11} x_2 x_1.$$

Кожна функція визначається своїм набором значень коефіцієнтів. Для пошуку значень коефіцієнтів необхідно вирішити систему рівнянь:

$$\begin{cases} y(0,0) = k_2^0 \vee k_1^0 \vee k_{21}^{00}; \\ y(0,1) = k_2^0 \vee k_1^1 \vee k_{21}^{01}; \\ y(1,0) = k_2^1 \vee k_1^0 \vee k_{21}^{10}; \\ y(1,1) = k_2^1 \vee k_1^1 \vee k_{21}^{11}. \end{cases}$$

Всі ненульові коефіцієнти після процедури поглинання визначають сукупність простих імплікант, тобто дозволяють побудувати скорочену ДНФ. Мінімізацію зручно виконувати за допомогою спеціальної таблиці, що після знаходження простих імплікант розглядається як таблиця покриття. За допомогою таблиці знаходять ТДНФ, а потім визначають МДНФ.

### Етапи мінімізації

1. Складання таблиці коефіцієнтів.

2. Викреслювання нульових коефіцієнтів.
3. Виділення простих імплікант.
4. Знаходження покриття, що відповідає тупиковим ДНФ.
5. Вибір МДНФ.

Приклад. Виконати мінімізацію функції, заданої табл. 3.5.

Складаємо таблицю коефіцієнтів (табл. 3.6). Викреслюємо в таблиці коефіцієнти, що знаходяться в рядках з нульовим значенням функції. Викреслені коефіцієнти мають нульові значення. Далі викреслюємо вже знайдені нульові коефіцієнти в інших рядках таблиці. Коефіцієнти, що залишилися, поглинають у рядку праворуч від себе всі інші коефіцієнти, в індекси яких входять індекси даного коефіцієнта. Наприклад,  $k_{32}^{01}$  поглинає  $k_{321}^{010}$ . (Поглинені коефіцієнти в табл. 3.6 позначені зірочкою).

Таблиця 3.5

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

Таблиця 3.6

$x_3$	$x_2$	$x_1$	Коефіцієнти						$y$	
0	0	0	<del><math>k_3^0</math></del>	<del><math>k_2^0</math></del>	<del><math>k_1^0</math></del>	<del><math>k_{32}^{00}</math></del>	<del><math>k_{31}^{00}</math></del>	<del><math>k_{21}^{00}</math></del>	<del><math>k_{321}^{000}</math></del>	0
0	0	1	<del><math>k_3^0</math></del>	<del><math>k_2^0</math></del>	<del><math>k_1^1</math></del>	<del><math>k_{32}^{00}</math></del>	$k_{31}^{01}$	<del><math>k_{21}^{01}</math></del>	$k_{321}^{001*}$	1
0	1	0	<del><math>k_3^0</math></del>	<del><math>k_2^1</math></del>	<del><math>k_1^0</math></del>	$k_{32}^{01}$	<del><math>k_{31}^{00}</math></del>	$k_{21}^{10}$	$k_{321}^{010*}$	1
0	1	1	<del><math>k_3^0</math></del>	<del><math>k_2^1</math></del>	<del><math>k_1^1</math></del>	$k_{32}^{01}$	$k_{31}^{01}$	$k_{21}^{11}$	$k_{321}^{011*}$	1
1	0	0	<del><math>k_3^1</math></del>	<del><math>k_2^0</math></del>	<del><math>k_1^0</math></del>	$k_{32}^{10}$	$k_{31}^{10}$	<del><math>k_{21}^{00}</math></del>	$k_{321}^{100*}$	1
1	0	1	<del><math>k_3^1</math></del>	<del><math>k_2^0</math></del>	<del><math>k_1^1</math></del>	$k_{32}^{10}$	<del><math>k_{31}^{11}</math></del>	$k_{21}^{01}$	$k_{321}^{101*}$	1
1	1	0	<del><math>k_3^1</math></del>	<del><math>k_2^1</math></del>	<del><math>k_1^0</math></del>	$k_{32}^{11}$	$k_{31}^{10}$	$k_{21}^{10}$	$k_{321}^{110*}$	1
1	1	1	<del><math>k_3^1</math></del>	<del><math>k_2^1</math></del>	<del><math>k_1^1</math></del>	<del><math>k_{32}^{11}</math></del>	<del><math>k_{31}^{11}</math></del>	<del><math>k_{21}^{11}</math></del>	<del><math>k_{321}^{111}</math></del>	0

Коефіцієнти, що залишилися,  $\{k_{32}^{01}, k_{32}^{10}, k_{31}^{01}, k_{31}^{10}, k_{21}^{01}, k_{21}^{10}\}$  визначають скорочену ДНФ.

$$y = x_3 x_2 \vee x_3 x_2 \vee x_3 x_1 \vee x_3 x_1 \vee x_2 x_1 \vee x_2 x_1.$$

Враховуємо в таблиці тільки ненульові коефіцієнти і розглядаємо її як таблицю покриття функції. Знаходимо дві ТДНФ, що містять по три імплі-

канти. Вони відповідають множині коефіцієнтів  $\{k_{32}^{01}, k_{31}^{10}, k_{21}^{01}\}$  і  $\{k_{32}^{10}, k_{31}^{01}, k_{21}^{10}\}$ . Інші ТДНФ складаються з чотирьох імплікант.

В якості МДНФ, наприклад, виберемо

$$y = x_3 x_2 \vee x_3 x_1 \vee x_2 x_1.$$

### Графічний метод мінімізації функцій

Існують два різновиди таблиць, що забезпечують одержання МДНФ, минаючи етапи формування скороченої і тупікової ДНФ.

На рис. 3.2 представлені діаграми Вейча і Карно для функцій 2, 3 і 4-х аргументів. Номера наборів показані всередині кліток.

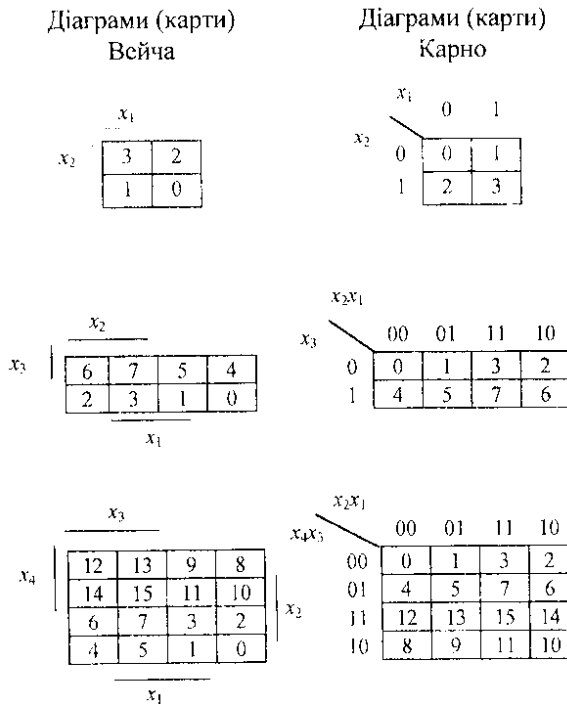


Рис. 3.2

Наочність методів зберігається при невеликій кількості аргументів.

Кожна клітинка відповідає конституенті. Прямокутник, що містить  $2^k$  клітинок ( $k=1, \dots, n-1$ ), відповідає імпліканті.

Обґрунтуванням графічного методу мінімізації є той факт, що поруч розташовані клітинки відповідають наборам аргументів, що відрізняється значенням однієї змінної і, таким чином, склеюються по Квайну. Чим більше клітинок містить прямокутник, тим менше букв входить у представлення імпліканти. Імпліканта містить тільки ті змінні, котрі приймають однакові значення для всіх клітинок прямокутника.

### Етапи мінімізації

1. Заповнення діаграми Вейча або карти Карно. Значення функцій записують в клітинки, що відповідають номерам наборів.

2. Об'єднання одиниць в прямокутники з максимально можливою кількістю клітинок (число клітинок повинне дорівнювати  $2^k$ ). При цьому кожна одиниця повинна входити як мінімум в один прямокутник. Прямокутник може містити й одну клітинку.

3. Визначення МДНФ. Сукупності простих імплікант, що входять у МДНФ, відповідає мінімальна множина прямокутників, що покривають всі одиниці.

*Приклад.* Одержати МДНФ функцій трьох аргументів (табл. 3.7).

Таблиця 3.7

$x_3$	$x_2$	$x_1$	$y_1$	$y_2$	$y_3$
0	0	0	0	1	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	1	1	0
1	1	1	0	1	1

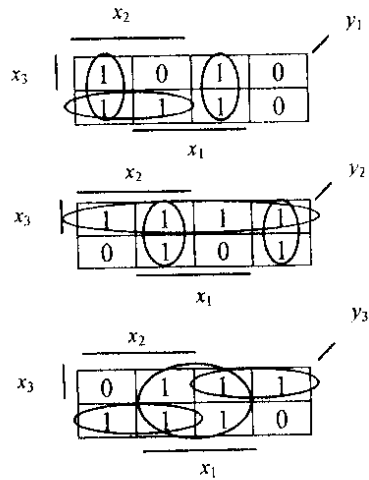


Рис. 3.5

Виходячи з діаграм Вейча на рис. 3.5, запишемо МДНФ функцій:

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

$$y_2 = x_3 \vee x_2 x_1 \vee x_2 x_1;$$

$$y_3 = x_1 \vee x_3 x_2 \vee x_3 x_2.$$



Графічний метод призначений для ручної мінімізації при невеликих  $n$ . При цьому відшукування імплікант не формалізоване, і успіх мінімізації цілком визначається кваліфікацією оператора.

### Підготовка до роботи

1. Визначити свій варіант перемикальних функцій, заданих табл. 3.8.

Для цього необхідно одержати дев'ять молодших розрядів номера залікової книжки студента, представленого в двійковій системі числення ( $h_9 h_8 h_7 \dots h_1$ ), а потім підставити  $h_i$  в табл. 3.8.

2. Виконати мінімізацію функцій або їх заперечень (в залежності від заданої елементної бази) наступними методами:

- методом Квайна (для  $f_1$ );
- методом Квайна – Макс-Класки ( $f_2$ );
- методом невизначених коефіцієнтів ( $f_3$ );
- методом діаграм Вейча ( $f_4$ ).

3. Одержати операторні представлення функцій. Для комбінаційних схем, що реалізують функції  $f_1$  і  $f_2$  використовувати елементи 3І-НЕ, а функцій  $f_3$  і  $f_4$  – елементи ЗАБО-НЕ.

4. Представити комбінаційні схеми, що відповідають отриманим операторним представленням функцій.

Таблиця 3.8

$x_4$	$x_3$	$x_2$	$x_1$	$f_1$	$f_2$	$f_3$	$f_4$
0	0	0	0	1	1	$h_1$	0
0	0	0	1	$h_1$	0	$h_2$	1
0	0	1	0	$h_2$	$h_1$	$h_3$	$h_3$
0	0	1	1	$h_3$	$h_2$	$h_4$	$h_4$
0	1	0	0	$h_4$	$h_3$	1	0
0	1	0	1	1	$h_4$	$h_5$	$h_5$
0	1	1	0	$h_5$	1	0	0
0	1	1	1	0	$h_5$	$h_6$	$h_6$
1	0	0	0	$h_6$	0	$h_7$	$h_7$
1	0	0	1	$h_7$	$h_6$	1	1
1	0	1	0	1	$h_7$	$h_8$	$h_8$
1	0	1	1	$h_8$	1	0	$h_2$
1	1	0	0	0	$h_8$	0	1
1	1	0	1	1	0	$h_9$	$h_9$
1	1	1	0	$h_9$	1	0	$h_1$
1	1	1	1	0	$h_9$	1	1

### Порядок виконання роботи

1. Побудувати моделі комбінаційних схем, вказаних викладачем.
2. Переконатися в правильності функціонування моделей, визначити параметри комбінаційних схем.

### Зміст звіту

Звіт повинний містити короткі теоретичні відомості, отримані формули, таблиці, малюнки і висновки за результатами роботи.

### Контрольні питання

1. Сформулювати визначення перемикальної функції, конституенти, імпліканти і простої імпліканти функції.
2. Що таке досконала, скорочена, тупікова і мінімальна ДНФ?
3. Дати визначення функціонально повної системи перемикальних функцій.
4. В чому сутність проблеми мінімізації перемикальних функцій?
5. Охарактеризувати основні етапи мінімізації перемикальних функцій різними методами.
6. Як побудувати операторні форми представлення функцій?
7. Дайте порівняльну оцінку методів мінімізації функцій.

### Література

1. Поспелов Д.А. Логические методы анализа и синтеза схем. – М.: Энергия, 1974. – 367 с.
2. Проектирование цифровых вычислительных машин/ Под ред. С.А. Майорова. Учебное пособие для студентов вузов. – М.: Высшая школа, 1972. – 344 с.
3. Савельев А.Я. Арифметические и логические основы цифровых автоматов: Учебник. – М.: Высшая школа, 1980. – 255 с.
4. Самофалов К.Г., Романкевич А.М., Валуйский В.Н., Каневский Ю.С., Пиневич М.М. Прикладная теория цифровых автоматов.– К.: Вища школа, 1987. – 375 с.
5. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум. – К.: Вища школа., 1990. – 215 с.

## 4. Лабораторна робота №3

### МІНІМІЗАЦІЯ СИСТЕМ ПЕРЕМИКАЛЬНИХ ФУНКЦІЙ

*Ціль роботи* – вивчення методу мінімізації систем перемикальних функцій, дослідження часових параметрів схем.

#### Теоретичні відомості

При спільній мінімізації декількох функцій з використанням диз'юнктивних нормальних форм (ДНФ) необхідно усунути дублювання однакових імплікант в представленні функцій, що приводить до спрощення комбінаційної схеми.

В даній роботі розглядається мінімізація систем функцій **методом Квайна (Квайна-Мак-Класки)**.

Вихідною формою для мінімізації функцій методом Квайна є досконала ДНФ (ДДНФ) системи функцій.

Перед мінімізацією кожен функцію необхідно представити в ДДНФ. В цьому випадку ДДНФ системи знаходиться як непересічна множина конститuent одиниці всіх функцій, причому кожній конститuentі приписується множина міток, що визначають її приналежність до функцій.

#### Етапи мінімізації

1. Склеювання термів.
2. Виконання поглинань.
3. Складання таблиці покриття.
4. Вибір форм представлення кожної функції.

Склеювання і поглинання виконуються за тими ж правилами, що і при мінімізації окремих функцій (див. розділ 2). Відмінності методу мінімізації систем полягають в наступному:

- склеювання здійснюються тільки для тих кон'юнктивних термів, в яких маються однакові мітки. Отриманому в результаті склеювання терму привласнюється множина міток, що обумовлюється як перетинання множин міток термів, що склеюються;
- поглинання здійснюється тільки в тому випадку, коли множини міток у термів повністю збігаються.

При виборі остаточної форми представлення функцій з використанням таблиці покриття необхідно прагнути до того, щоб загальне число букв у покритті було мінімальним.

При використанні методу Квайна-Мак-Класки всі операції виконуються аналогічно, але в цифровій формі.

*Приклад.* Виконати мінімізацію системи функцій, заданої табл. 4.1.

Скористаємося методом Квайна-Мак-Класки. Послідовно одержуємо комплекси кубів:

Таблиця 4.1

$x_3$	$x_2$	$x_1$	$y_1$	$y_2$	$y_3$
0	0	0	1	0	1
0	0	1	1	1	1
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	1	0
1	0	1	0	0	1
1	1	0	1	1	0
1	1	1	0	0	1

$$K^0 = \begin{cases} \overline{000(1,3)} \\ \overline{001(1,2,3)} \\ \overline{010(2)} \\ \overline{100(1,2)} \\ \overline{011(1,2,3)} \\ \overline{101(3)} \\ \overline{110(1,2)} \\ \overline{111(3)} \end{cases}; \quad K^1 = \begin{cases} \overline{X00(1)} \\ \overline{X01(3)} \\ \overline{X10(2)} \\ \overline{X11(3)} \\ \overline{0X1(1,2,3)} \\ \overline{1X0(1,2)} \\ \overline{1X1(3)} \\ \overline{00X(1,3)} \\ \overline{01X(2)} \end{cases}; \quad K^2 = \begin{cases} \overline{XX1(3)} \\ \overline{XX1(3)} \end{cases}$$

Визначаємо покриття, що відповідає скороченій ДНФ системи:

$$Z = \begin{cases} \overline{X00(1)} \\ \overline{X10(2)} \\ \overline{0X1(1,2)} \\ \overline{1X0(1,2)} \\ \overline{00X(1,3)} \\ \overline{01X(2)} \\ \overline{XX1(3)} \end{cases}$$

Складасмо таблицю покриття (табл. 4.2), на підставі якої знаходимо остаточної форми представлення функцій, що забезпечують їх спільну реалізацію з мінімальними апаратними витратами.

Таблиця 4.2

	$y_1$					$y_2$					$y_3$				
	000	001	011	100	110	000	010	011	100	110	000	001	011	101	111
00X(1,3)	✓	✓									✓	✓			
X00(1)	✓			✓											
0X1(1,2,3)		✓	✓			✓		✓				✓	✓		
01X(2)							✓	✓							
X10(2)							✓					✓			
1X0(1,2)				✓	✓					✓	✓				
XX1(3)												✓	✓	✓	✓

$$y_1 = \overline{x_3}x_2 \vee x_3\overline{x_1} \vee x_3x_1; \quad y_2 = x_3x_1 \vee x_3x_2 \vee x_3\overline{x_1}; \quad y_3 = x_3x_2 \vee x_1.$$

На рис. 4.1 приведена комбінаційна схема, що відповідає знайденим формам функцій.

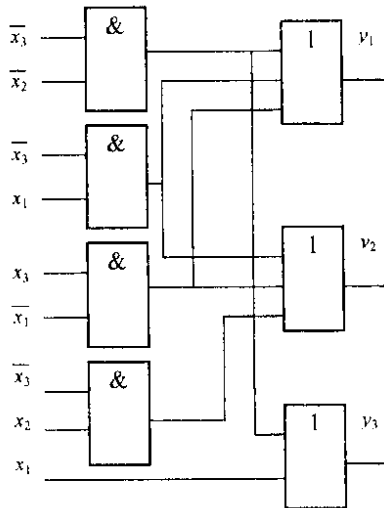


Рис. 4.1

При побудові комбінаційних схем, що відповідають системам булевих функцій, виникає проблема забезпечення заданого коефіцієнта розгалуження по виходу елементів, що реалізують загальні імпліканти.

Якщо число входів, до яких повинний бути підключений вихід елемента, перевищує коефіцієнт розгалуження, то застосовують способи дублювання елемента або посилення сигналу.

В першому випадку входи однакових елементів з недостатньою навантажувальною здатністю з'єднують паралельно, а в другому випадку до виходу елемента підключають повторювачі.

### Підготовка до роботи

1. Визначити свій варіант системи перемикальних функцій (табл. 4.3).

Для цього необхідно одержати дев'ять молодших розрядів номера залікової книжки студента, представленого в двійковій системі числення ( $h_9, h_8, h_7, \dots, h_1$ ), а потім підставити  $h_i$  в табл. 4.3.

2. Виконати спільну мінімізацію функцій і їх заперечень методом Квайна або Квайна-Мак-Класки.

3. Одержати операторні представлення функцій (два варіанти) для їх реалізації у формі І/АБО і формі І/АБО-НЕ. Можна використовувати елементи з будь-яким числом входів, не більш чотирьох.

4. Представити комбінаційні схеми, що відповідають отриманим операторним формам. При необхідності виконати розв'язку схем з урахуванням коефіцієнта розгалуження елементів по виходу, рівного двом.

### Порядок виконання роботи

1. Побудувати моделі комбінаційних схем, вказаних викладачем.
2. Переконатися в правильності функціонування моделей.
3. Визначити часові параметри схем за допомогою часових діаграм.

Таблиця 4.3

$x_4$	$x_3$	$x_2$	$x_1$	$f_1$	$f_2$	$f_3$
0	0	0	0	1	1	$h_1$
0	0	0	1	$h_1$	0	$h_2$
0	0	1	0	$h_2$	$h_1$	$h_3$
0	0	1	1	$h_3$	$h_2$	$h_4$
0	1	0	0	$h_4$	$h_3$	1
0	1	0	1	1	$h_4$	$h_5$
0	1	1	0	$h_5$	1	0
0	1	1	1	0	$h_5$	$h_6$
1	0	0	0	$h_6$	0	$h_7$
1	0	0	1	$h_7$	$h_6$	1
1	0	1	0	1	$h_7$	$h_8$
1	0	1	1	$h_8$	1	0
1	1	0	0	0	$h_8$	0
1	1	0	1	1	0	$h_9$
1	1	1	0	$h_9$	1	0
1	1	1	1	0	$h_9$	1

## Зміст звіту

Звіт повинний містити короткі теоретичні відомості, отримані формули, таблиці, малюнки і висновки за результатами роботи.

## Контрольні питання

1. Сформулюйте визначення досконалої ДНФ системи функцій.
2. Охарактеризуйте етапи мінімізації системи функцій.
3. Як виконуються операції склеювання і поглинання при мінімізації систем перемикальних функцій?
4. В чому сутність проблеми мінімізації систем перемикальних функцій?
5. Охарактеризуйте способи забезпечення заданого коефіцієнта розгалуження елементів по виходу. Вкажіть переваги і недоліки способів.
6. Як побудувати операторні форми представлення функцій?

## Література

1. Поспелов Д.А. Логические методы анализа и синтеза схем. – М.: Энергия, 1974. – 367 с.
2. Самофалов К.Г., Романкевич А.М., Валуїський В.Н., Каневський Ю.С., Пиневич М.М. Прикладная теория цифровых автоматов.– К.: Вища школа, 1987. – 375 с.
3. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум. -- К.: Вища школа., 1990. – 215 с.

## 5. Лабораторна робота №4

### МІНІМІЗАЦІЯ ЧАСТКОВО ВИЗНАЧЕНИХ ФУНКЦІЙ

*Ціль роботи* – вивчення методів мінімізації частково визначених функцій, дослідження параметрів схем.

#### Теоретичні відомості

В реальних системах можливі випадки, коли не всі набори змінних можуть подаватися на входи комбінаційної схеми, тобто існують заборонені відні комбінації змінних.

На заборонених наборах функція вважається невизначеною, що дає додаткові можливості для спрощення комбінаційної схеми. В таблиці істинності значення функції на таких наборах відзначаються символом, відмінним від 0 і 1, наприклад – прочерком. Довизначення функції на заборонених наборах необхідно робити таким чином, щоб забезпечити найбільш ефективну мінімізацію.

При використанні для мінімізації методу діаграм Вейча прочерки розглядають як одиниці в тих випадках, коли це приводить до збільшення розміру прямокутника, що відповідає імпліканті. В протилежному випадку вони розглядаються як нулі.

*Приклад.* Знайти МДНФ функції, заданої діаграмою Вейча (рис. 5.1).

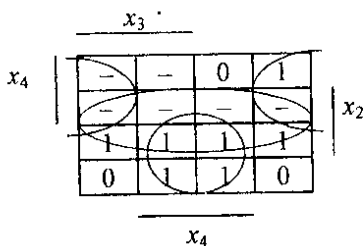


Рис. 5.1

Після відшукування простих імплікант, записуємо МДНФ

$$y = x_4 \overline{x_1} \vee x_2 \vee \overline{x_4} x_1.$$

При використанні аналітичних методів мінімізації функцій у її ДДНФ вводять всі конституенти заборонених наборів, але в таблицю покриття дані конституенти не включаються.



Приклад. Одержати МДНФ частково визначеної функції, заданої табл. 5.1 методом Квайна.

Таблиця 5.1

$x_3$	$x_2$	$x_1$	$y$
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	—
1	0	0	—
1	0	1	—
1	1	0	1
1	1	1	—

Записуємо досконалу ДНФ:

$$y = \overline{x_3 x_2 x_1} \vee \overline{x_3 x_2 x_1} \vee \overline{x_3 x_2 x_1} .$$

Після доповнення конститuentами одиниці, що відповідають забороненим наборам, функція приймає вигляд

$$y = \overline{x_3 x_2 x_1} \vee \overline{x_3 x_2 x_1} \vee \overline{x_3 x_2 x_1} \vee \overline{x_3 x_2 x_1} \vee \overline{x_3 x_2 x_1} \vee \overline{x_3 x_2 x_1} \vee \overline{x_3 x_2 x_1} .$$

Для зручності мінімізації конститuentи та імпліканти записуємо у стовпчик.

$$\begin{array}{l}
 \overline{x_3 x_2 x_1} \text{ (0)} \\
 \overline{x_3 x_2 x_1} \text{ (1)} \\
 \overline{x_3 x_2 x_1} \text{ (3)} \\
 \overline{x_3 x_2 x_1} \text{ (4)} \\
 \overline{x_3 x_2 x_1} \text{ (5)} \\
 \overline{x_3 x_2 x_1} \text{ (6)} \\
 \overline{x_3 x_2 x_1} \text{ (7)}
 \end{array}
 ;
 \begin{array}{l}
 \overline{x_3 x_2} \text{ (0,1)} \\
 \overline{x_3 x_1} \text{ (0,4)} \\
 \overline{x_3 x_1} \text{ (1,3)} \\
 \overline{x_3 x_1} \text{ (1,5)} \\
 \overline{x_2 x_1} \text{ (3,7)} \\
 \overline{x_3 x_2} \text{ (4,5)} \\
 \overline{x_3 x_1} \text{ (4,6)} \\
 \overline{x_3 x_1} \text{ (5,7)} \\
 \overline{x_3 x_2} \text{ (6,7)}
 \end{array}
 ;
 \begin{array}{l}
 \overline{x_2} \text{ (0,4,1,5)} \\
 \overline{x_1} \text{ (1,5,3,7)} \\
 \overline{x_3} \text{ (4,5,6,7)}
 \end{array}
 .$$

В результаті склеювання і поглинання одержуємо скорочену ДНФ функції, що довизначена, можливо, не оптимально.

$$y = \overline{x_2} \vee \overline{x_1} \vee \overline{x_3} .$$

Будуємо таблицю покриття, в яку включаємо тільки три вихідні конститuentи одиниці (табл. 5.2).

Таблиця 5.2

Імпліканти	Конституенти		
	$\overline{x_3 x_2 x_1}$	$x_3 \overline{x_2} x_1$	$x_3 x_2 \overline{x_1}$
$\overline{x_2}$	✓	✓	
$x_1$		✓	
$x_3$			✓

Знаходимо ТДНФ функції, що одночасно є її МДНФ:

$$y = \overline{x_2} \vee x_3.$$

При мінімізації системи частково визначених функцій в її ДДНФ також вводяться конституенти, що відповідають забороненим наборам. На етапі вибору мінімального покриття ці конституенти не включаються в таблицю покриття.

Таблиця 5.3

*Приклад.* Виконати мінімізацію методом Квайна-Мак-Класки системи функцій, заданої табл. 5.3.

Випишемо 0-куби, що відповідають забороненим наборам і наборам, на яких функції приймають одиничне значення. При цьому відзначасмо приналежність кубів до заданих функцій. Виконуємо склеювання і поглинання:

$x_3$	$x_2$	$x_1$	$y_1$	$y_2$	$y_3$
0	0	0	-	1	-
0	0	1	1	1	0
0	1	0	0	0	0
0	1	1	1	1	1
1	0	0	1	-	1
1	0	1	0	-	1
1	1	0	0	0	0
1	1	1	-	1	1

$$K^0 = \begin{cases} \overline{000(1,2,3)} \\ \overline{001(1,2)} \\ \overline{100(1,2,3)} \\ \overline{011(1,2,3)} \\ \overline{101(2,3)} \\ \overline{111(1,2,3)} \end{cases}; \quad K^1 = \begin{cases} \overline{x00(1,2,3)} \\ \overline{x01(2)} \\ \overline{x11(1,2,3)} \\ \overline{0x1(1,2)} \\ \overline{1x1(2,3)} \\ \overline{00x(1,2)} \\ \overline{10x(2,3)} \end{cases}; \quad K^2 = \begin{cases} \overline{x0x(2)} \\ \overline{x0x(2)} \\ \overline{xx1(2)} \\ \overline{xx1(2)} \end{cases}; \quad Z = \begin{cases} \overline{x00(1,2,3)} \\ \overline{x11(1,2,3)} \\ \overline{0x1(1,2)} \\ \overline{1x1(2,3)} \\ \overline{00x(1,2)} \\ \overline{10x(2,3)} \\ \overline{x0x(2)} \\ \overline{xx1(2)} \end{cases}$$

Таблиця 5.4

	$y_1$			$y_2$				$y_3$			
	001	011	100	000	001	011	111	011	100	101	111
$a=00X(1,2)$	⊙			⊙	⊙						
$b=X00(1,2,3)$			⊙	∨					⊙		
$c=0X1(1,2)$	∨	∨			∨	∨					
$d=X11(1,2,3)$		⊙				⊙	⊙	⊙			∨
$e=10X(2,3)$									∨	∨	
$f=1X1(2,3)$							∨			∨	∨
$g=X0X(2)$				∨	∨						
$h=XX1(2)$					∨	∨	∨				

Складаємо таблицю покриття (табл. 5.4), на підставі якої знаходимо форми функцій вихідної системи, що забезпечують їх спільну реалізацію.

Для визначення покриття (для однієї функції або системи функцій) як можливий варіант можна використовувати метод Петрика, що складається з виконання наступних етапів:

- визначення умов покриття імплікантами кожної конституенти одиниці окремо, використовуючи функцію АБО;
- складання умови одночасного покриття всіх конститuent одиниці з використанням функції І;
- розкриття дужок в отриманому логічному виразі за правилами булевої алгебри.

Кон'юнктивні терми, отримані в результаті виконання зазначених етапів, відповідають множинам імплікант, кожна з яких визначає можливе покриття. З отриманих варіантів покриття вибирають один відповідно до цільової функції проектування (мінімальні апаратурні витрати, максимальна швидкодія і т. ін.).

Позначимо для зручності виконання логічних перетворень імпліканти в табл. 5.4 буквами від  $a$  до  $h$ .

Умова покриття 001 має вигляд  $(a \vee c)$ , для 011 одержимо  $(c \vee d)$  і т. ін. Покриття всіх конститuent визначається узагальненою умовою

$$(a \vee c)(a \vee d)b(a \vee b \vee g)(a \vee c \vee g \vee h)(c \vee d \vee h)(d \vee f \vee h)d \& \\ \& (b \vee e)(e \vee f)(d \vee f).$$

Після поглинання диз'юнктивних термів одержуємо  $(a \vee c)bd(e \vee f)$ , а після розкриття дужок знаходимо  $abde \vee cbde \vee abdf \vee cbdf$ .

Отримані терми визначають чотири множини імплікант, що покривають функції. Вибравши множини  $\{a, b, d, f\}$ , можна одержати

$$y_1 = \overline{x_3} \overline{x_2} \vee x_2 x_1 \vee \overline{x_2} \overline{x_1};$$

$$y_2 = \overline{x_3} x_2 \vee x_2 x_1;$$

$$y_3 = x_2 x_1 \vee \overline{x_2} \overline{x_1} \vee x_3 x_1.$$

При перехідних процесах на виходах комбінаційних схем можуть формуватися помилкові (не передбачені таблицею істинності) короткочасні сигнали ("просічки"). Якщо такі сигнали неприпустимі (можуть привести до неправильного спрацьовування інших схем), то для їх усунення використовуються "пастки", варіанти яких представлені на рис.5.2.

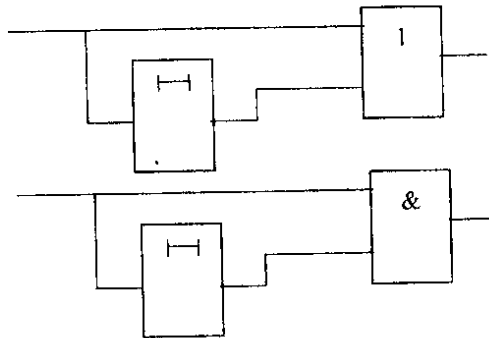


Рис. 5.2

### Підготовка до роботи

1. Визначити свій варіант системи перемикальних функцій (табл. 5.5). Для цього необхідно одержати дев'ять молодших розрядів номера залікової книжки студента, представленого в двійковій системі числення ( $h_8 h_7 \dots h_1$ ), а потім підставити  $h_i$  в табл. 5.5.
2. Виконати окремо мінімізацію кожної функції методом Вейча.
3. Виконати спільну мінімізацію функцій методом Квайна.
4. Виконати спільну мінімізацію заперечення функцій методом Квайна-Мак-Класки.

5. Представити комбінаційні схеми, що відповідають отриманим операторним формам. Оцінити можливість формування короточасних помилкових сигналів в отриманих схемах. Показати способи усунення ризику збою в комбінаційних схемах.

Таблиця 5.5

$x_4$	$x_3$	$x_2$	$x_1$	$f_1$	$f_2$	$f_3$
0	0	0	0	1	1	1
0	0	0	1	1	1	0
0	0	1	0	1	1	1
0	0	1	1	0	0	0
0	1	0	0	-	0	1
0	1	0	1	0	0	0
0	1	1	0	1	-	-
0	1	1	1	-	-	1
1	0	0	0	1	$h_4$	$h_7$
1	0	0	1	0	0	$h_8$
1	0	1	0	0	0	$h_9$
1	0	1	1	$h_1$	0	0
1	1	0	0	1	-	1
1	1	0	1	$h_2$	$h_5$	0
1	1	1	0	$h_3$	$h_6$	0
1	1	1	1	1	1	1

### Порядок виконання роботи

1. Побудувати моделі комбінаційних схем, вказаних викладачем.

2. Переконатися в правильності функціонування моделей, усунути при необхідності короточасні помилкові вихідні сигнали.

3. Визначити часові параметри схем за допомогою часових діаграм.

### Зміст звіту

Звіт повинний містити короткі теоретичні відомості, отримані формули, таблиці, малюнки і висновки за результатами роботи.

### Контрольні питання

1. В чому складається особливість мінімізації частково визначених функцій?
2. Охарактеризувати основні етапи спільної мінімізації систем частково визначених перемикальних функцій.
3. Як одержати операторні форми представлення функцій?
4. Чим пояснюється можливість виникнення збоїв комбінаційних схем?
5. Як оцінити апаратні витрати і швидкодію комбінаційних схем?
6. Як забезпечити заданий коефіцієнт розгалуження елементів по виходу при побудові комбінаційних схем з багатьма виходами?
7. Як усунути короточасні помилкові сигнали на виходах комбінаційної схеми при перехідних процесях?

8. Як вибрати тривалість затримки сигналу при використанні схеми “пастки”?

### Література

1. Поспелов Д.А. Логические методы анализа и синтеза схем. – М.: Энергия, 1974. – 367 с.
2. Самофалов К.Г., Романкевич А.М., Валуйский В.Н., Каневский Ю.С., Пиневич М.М. Прикладная теория цифровых автоматов.– К.: Вища школа, 1987. – 375 с.
3. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум. – К.: Вища школа., 1990. – 215 с.

## 6. Лабораторна робота №5

### ПРОЕКТУВАННЯ І ДОСЛІДЖЕННЯ ТРИГЕРІВ

*Ціль роботи* – вивчити особливості функціонування і схемні різновиди тригерів різних типів, описувати методи їх проектування в потенційній елементній базі, одержати навички експериментального дослідження основних характеристик тригерів.

#### Теоретичні відомості

У схемах цифрової обчислювальної техніки в якості запам'ятовуючих елементів широко використовуються тригерні пристрої (тригери). Тригерна схема являє собою пристрій із двома стійкими станами, що містить запам'ятовуючий елемент (ЗЕ) (власне тригер), і схему керування (СК).

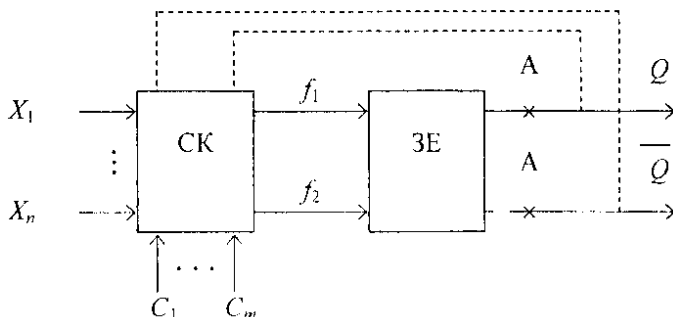


Рис. 6.1

На рис. 6.1 показана тригерна схема. На ній прийняті наступні позначення:

$X_1, \dots, X_n$  – інформаційні входи;

$C_1, \dots, C_m$  – тактуючі входи;

$f_1$  і  $f_2$  – функції порушення ЗЕ;

$Q$  і  $\bar{Q}$  – відповідно прямий і інверсний виходи.

Можливі і більш прості варіанти тригерних схем, коли, наприклад, відсутній СК і тригер не має тактуючих входів.

Тригери класифікують по функціональній ознаці і способі запису інформації.

*Функціональна класифікація* більш загальна і характеризує стани входів і виходів тригера в моменти часу до його спрацьовування і після нього.

Наприклад, якщо тригер має один інформаційний вхід (тобто можливі стани на вході  $X=0$  і  $X=1$  і на виході  $-0, 1, Q, \bar{Q}$ , а також один невизначений стан \*), то можна одержати 25 функціональних типів тригерів. У загальному випадку при наявності  $n$  інформаційних входів можна одержати  $5^{2^n}$  типів тригерних схем. На практиці застосовується велика кількість типів тригерів. До них відносяться RS-, D-, DV-, JK-, T-, E-, R-, S-тригери.

Спосіб функціонування тригерів може бути описаний таблицею переходів. З таблиці переходів **RS-тригера** (табл. 6.1) випливає, що тригер не змінює свого стану в момент  $t^{S+1}$  ( $Q^{S+1}=Q^S$ ), якщо в момент часу  $t$  має місце  $R^S=0$  і  $S^S=0$ . При наявності сигналів  $R^S=0$ ,  $S^S=1$  тригер встановлюється в одиничний стан  $Q^{S+1}=1$ , а при комбінації  $R^S=1$ ,  $S^S=0$  – в нульовий  $Q^{S+1}=0$ . При  $R^S=S^S=1$  стан тригера не визначене ( $Q^{S+1}=*$ ). Така комбінація сигналів для RS-тригера є забороненою.

**R-тригер** відрізняється від RS-тригера тим, що при комбінації вхідних сигналів  $S^S=R^S=1$  він переходить у нульовий стан ( $Q^{S+1}=0$ ) (табл. 6.2). **S-тригер** (табл. 6.3) у цьому випадку переходить в одиничний стан ( $Q^{S+1}=1$ ), а **E-тригер** (табл. 6.4) не змінює свого стану ( $Q^{S+1}=Q^S$ ).

Таблиця 6.1

Переходи RS-тригера		
$R^S$	$S^S$	$Q^{S+1}$
0	0	$Q^S$
0	1	1
1	0	0
1	1	*

Таблиця 6.2

Переходи R-тригера		
$R^S$	$S^S$	$Q^{S+1}$
0	0	$Q^S$
0	1	1
1	0	0
1	1	0

Таблиця 6.3

Переходи S-тригера		
$R^S$	$S^S$	$Q^{S+1}$
0	0	$\bar{Q}^S$
0	1	1
1	0	0
1	1	1

Таблиця 6.4

Переходи E-тригера		
$R^S$	$S^S$	$Q^{S+1}$
0	0	$Q^S$
0	1	1
1	0	0
1	1	$Q^S$

**D-тригер** називають також тригером затримки (табл. 6.5). Він затримує вхідний сигнал. Для такого тригера справедлива рівність ( $Q^{S+1}=D^S$ ). **DV-тригер** відрізняється від D-тригера тим, що має додатковий вхід V. При  $V=1$  DV-тригер працює як D-тригер (табл. 6.6), а при  $D=0$  – не змінює свого стану (як при  $D=0$ , так і при  $D=1$ ).

**T-тригер** називають також рахунковим. Він підраховує одиниці, що надходять на вхід T, по модулю 2, що видно з табл. 6.7.

Як впливає з таблиці переходів **JK-тригера** (табл. 6.8) при комбінаціях вхідних сигналів  $J=K=0$ ,  $J=0$  і  $K=1$ ,  $J=1$  і  $K=0$  він працює як RS-тригер



(вхід  $J$  відповідає входові  $S$ , а  $K$  – входові  $R$ ), а при  $J=K=1$  змінює свій стан на протилежний, тобто працює як рахунковий (Т-тригер).

Таблиця 6.5  
Переходи  
D-тригера

$C^S$	$D^S$	$Q^{S+1}$
0	0	$Q^S$
0	1	$Q^S$
1	0	0
1	1	1

Таблиця 6.6  
Переходи  
Т-тригера

$T$	$Q^{S+1}$
0	$Q^S$
1	$\overline{Q^S}$

Таблиця 6.7  
Переходи  
JK-тригера

$J^S$	$K^S$	$Q^{S+1}$
0	0	$Q^S$
0	1	0
1	0	1
1	1	$\overline{Q^S}$

Класифікація тригерів за способом запису інформації характеризує хід процесу переключення тригера. Відповідно до цієї класифікації тригери розділяють на *асинхронні* і *синхронні*. Запис інформації в *асинхронні тригери* здійснюється безпосередньо з надходженням інформаційних сигналів (такі тригери не мають тактуючих входів). *Синхронні тригери* мають тактуючі входи. Якщо число тактуючих входів дорівнює  $m$ , то формування нового стану тригера завершується після надходження  $m$ -го тактуючого сигналу. В даний час найбільш часто використовуються одноктактні тригери, що розглядаються далі.

Розрізняють синхронні тригери, *керовані рівнем* тактового сигналу, і з *внутрішньою затримкою*.

Тригери першого типу з появою тактуючого сигналу на вході  $C$  (тобто при  $C=1$ ) підключаються відповідно до таблиці переходів. Характерна риса таких тригерів полягає в тому, що при  $C=1$  вони можуть переключатися стільки разів, скільки разів змінюються інформаційні сигнали, тобто сигнали на виходах тригерів можуть неодноразово змінюватися, поки рівень тактуючого сигналу на вході  $C$  дорівнює рівневі логічної одиниці.

У тригерах другого типу вихідні сигнали, що відповідають новому станові тригера, з'являються тільки в момент переходу тактуючого сигналу з 0 у 1 або навпаки. У деяких реальних схемах тригерів із внутрішньою затримкою зміна інформаційних сигналів при сталому рівні тактуючого сигналу може викликати переключення ряду логічних елементів, однак на виходах тригера сигнали при цьому не змінять свого значення.

Якщо тригер змінює свій стан при переході тактуючого сигналу з 0 в 1, то вважають, що тригер спрацює по передньому позитивному фронту тактуючого сигналу, а при переході тактуючого сигналу з 1 в 0 – по задньому негативному фронту.

Проектування тригерних пристроїв складається у вибірці ЗЕ (див. рис. 6.1) і синтезі СК, що реалізує функції збудженні  $f_1$  і  $f_2$  для ЗЕ в заданому елементному базисі.

Якщо в стовпці  $Q^{S+1}$  таблиці переходів проєктованого тригера є значення  $\overline{Q^S}$  (див., наприклад, табл. 6.6 і 6.7), то сигнали на виходах  $Q$  і  $\overline{Q}$  тригера являють собою аргументи функцій  $f_1$  і  $f_2$ . Для забезпечення правильного переключення тригера в точках А (див. рис. 6.1) необхідно включити елементи затримки. Аналогічна ситуація виникає в тому випадку, коли аргументами функцій  $f_1$  і  $f_2$  будуть сигнали на виходах  $Q$  і  $\overline{Q}$  інших тригерів, що переключаються в процесі роботи одночасно з даним тригером. Наприклад, такими "залежними" є тригери в зсувних регістрах, лічильниках і т. ін.

Однак при побудові тригерів на потенційних елементах не можуть бути використані елементи затримки, що містять реактивні компоненти, через складність виготовлення зі стабільними характеристиками таких компонентів в інтегральному виконанні. У цьому випадку застосовують два основних способи побудови тригерів із внутрішньою затримкою: по *MS-схемі* і за *схемою трьох тригерів*.

Перший спосіб полягає у використанні для побудови тригера двох ЗЕ: основного (М-ЗЕ; М-тригера) і допоміжного (S-ЗЕ, S-тригера). Структурна схема однокантного М-тригера показана на рис. 6.2.

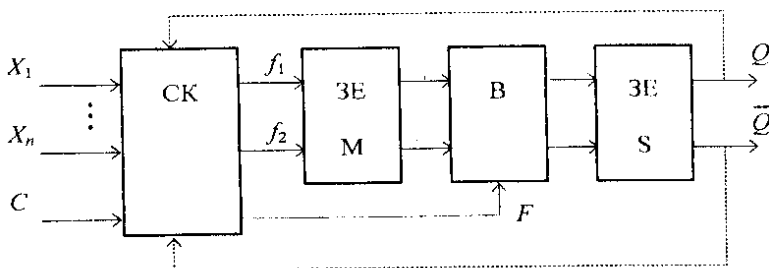


Рис. 6.2. Структура однокантного М-тригера

Запис інформації в М-тригер тактується сигналом  $C$ , а в S-тригер – сигналом  $F$ . Інформація з М-тригера в S-тригер передається через вентилі  $B$ .

Найбільше поширення одержали MS-тригери з інвертором у ланцюзі  $C$  та MS-тригери зі зв'язками, що забороняють переключення S-тригера. Схеми таких тригерів на елементах І-НЕ показані на рис. 6.3 і рис. 6.4. На даних схемах елементи 1, 2 утворюють S-тригер, а елементи 5, 6 – М-тригер. Вентилі 3, 4 у схемі на рис. 6.3 керуються сигналом виходу інвертора 7, а в схемі на рис. 6.4 – сигналами  $f_1$  і  $f_2$ .

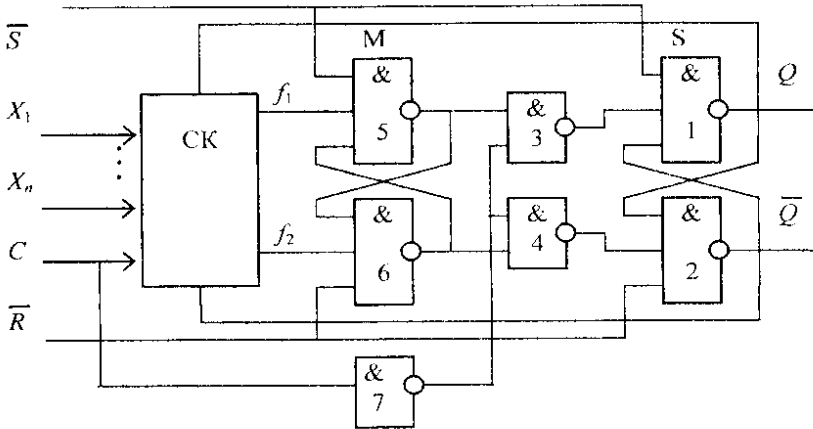


Рис. 6.3. Структура MS-тригера з інвертором в ланцюзі  $C$

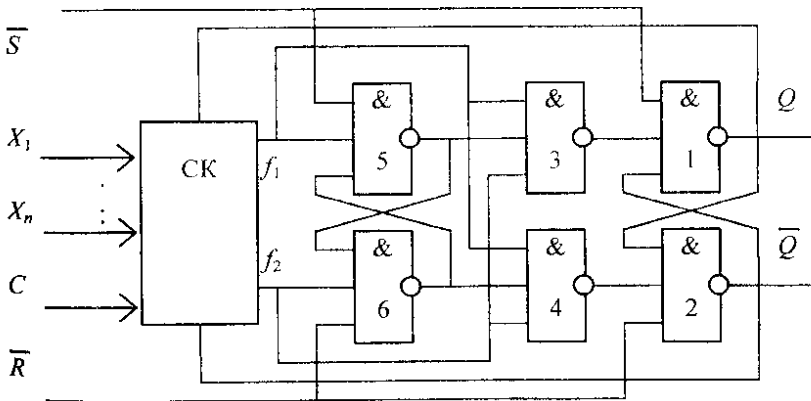


Рис. 6.4

У розглянутих схемах елементи І-НЕ можуть бути замінені на елементи АБО-НЕ.

Тригерний пристрій, побудований на елементах І-НЕ за схемою трьох тригерів, зображений на рис. 6.5.

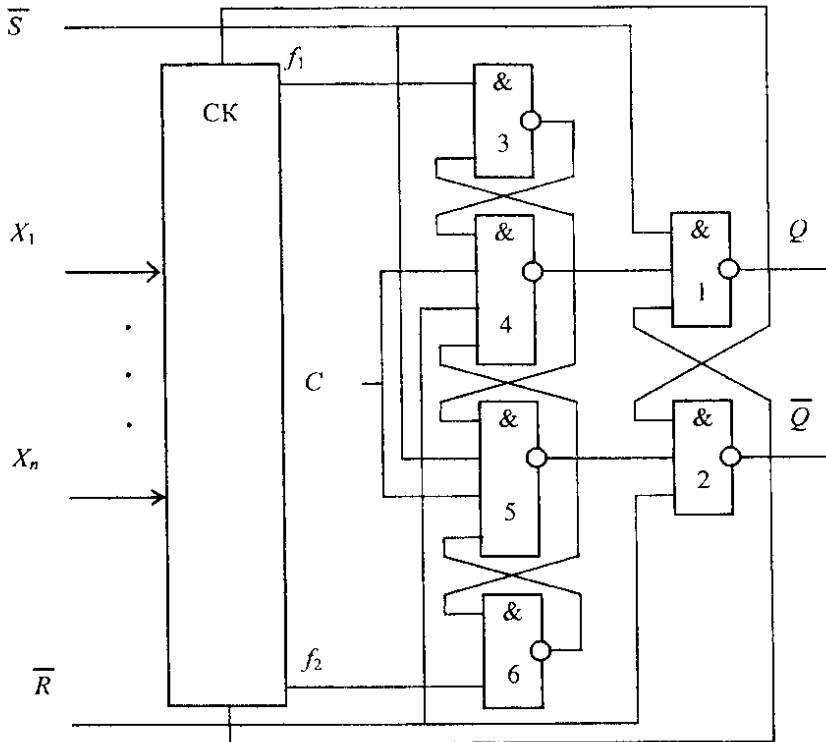


Рис. 6.5

У розглянутому тригері сигнали, що відповідають новому станові, установлюються при переході тактуючого сигналу з 0 у 1. При  $C=0$  на виходах вентилів 4 і 5 є присутнім сигнал логічної одиниці, тобто тригер на елементах 1 і 2 не змінює свого стану, а вентиля 3 і 6 виконують функцію інвертора.

Нехай, наприклад,  $f_1=0$  і  $f_2=1$ . Тоді при  $C=1$  на виході вентиля 4 з'являється сигнал з рівнем логічного нуля, що встановлює основний тригер (вентилі 1 і 2) в одиничний стан і підтверджує одиничний сигнал на виході вентиля 3. Після цього сигнали  $f_1$  і  $f_2$  можуть змінювати своє значення, що вплине на стан основного тригера (вентилі 1 і 2) доти, поки не здійсниться черговий перехід сигналу  $C$  з 0 у 1. Розглянутий тригер може бути побудований і на елементах АБО-НЕ.

Перед синтезом СК необхідно визначити, при яких значеннях  $f_1$  і  $f_2$  ЗЕ, виходи якого є виходами  $Q$  і  $\bar{Q}$  тригера, здійснює визначені переходи з одного стану в інший в момент спрацювання тригера.

При синтезі СК на підставі таблиці переходів тригера будується повна таблиця переходів, у якій відображають також значення  $Q^S$  у момент часу  $t$  і, при необхідності, - значення  $C$ . З повної таблиці переходів одержують вирази для  $f_1$  і  $f_2$ , мінімізують отримані функції і реалізують їх на заданих елементах.

Як приклад розглянемо процес проектування JK-тригера на елементах І-НЕ.

Оскільки в табл. 6.8 є значення  $Q^S$ , то тригер повинний мати внутрішню затримку. Вибираємо структуру тригера, показану на рис. 6.3.

Порядок переходів тригера на вентилях І-НЕ (див. елементи 1 і 2 на рис. 6.3) у залежності від значень  $f_1$  і  $f_2$  при переході сигналу на вході  $C$  з 1 у 0 можна відобразити у вигляді системи підграфів (рис. 6.6-а). У разі використання елементів АБО-НЕ застосовують систему рис. 6.6-б.

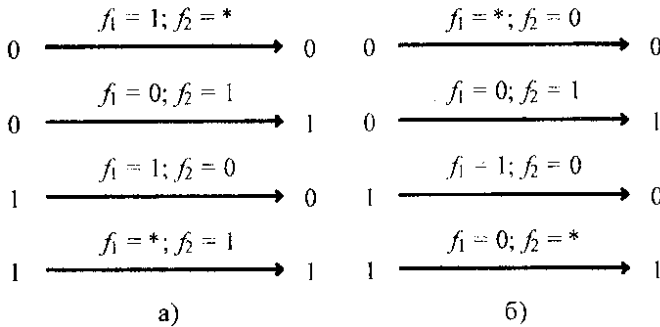


Рис. 6.6

Тут знаком \* відзначено довільні значення функцій  $f_1$  і  $f_2$ .

Для розглянутого тригера одержуємо повну таблицю переходів (табл. 6.8), побудовану відповідно до табл. 6.7.

На підставі визначеного раніше порядку переходів тригера на вентилях 1 і 2 (див. рис. 6.3) заповнюємо в табл. 6.8 графи для  $f_1$  і  $f_2$ , аналізуючи переходи  $Q^S \rightarrow Q^{S+1}$  у кожному рядку таблиці.

Таблиця 6.8

Повна таблиця переходів JK-тригера

C	$J^S$			$J^{S+1}$	$f_1$	$f_2$
	$J^S$	$K^S$	$Q^S$	$Q^{S+1}$		
0	0	0	0	0	1	*
0	0	0	1	1	*	1
0	0	1	0	0	1	*
0	0	1	1	1	*	1
0	1	0	0	0	1	*
0	1	0	1	1	*	1
0	1	1	0	0	1	*
0	1	1	1	1	*	1
1	0	0	0	0	1	*
1	0	0	1	1	*	1
1	0	1	0	0	1	*
1	0	1	1	0	1	0
1	1	0	0	1	0	1
1	1	0	1	1	*	1
1	1	1	0	1	*	1
1	1	1	1	0	1	0

$C^S$	$J^S$			$f_1$
	$J^S$	$K^S$	$Q^S$	
0	*	*	1	
0	1	1	1	
1	*	*	1	
1	*	*	1	

Рис. 6.7

$C^S$	$J^S$			$f_2$
	$J^S$	$K^S$	$Q^S$	
1	1	1	*	
1	0	0	*	
*	1	1	*	
*	1	1	*	

Рис. 6.8

За допомогою діаграми Вейча (рис. 6.7 і 6.8) знаходимо мінімальну диз'юнктивну нормальну форму функцій  $f_1$  і  $f_2$  (індекси  $S$  при цьому опускаємо):

$$f_1 = \bar{C} \vee Q \vee \bar{J};$$

$$f_2 = \bar{C} \vee \bar{Q} \vee \bar{K}.$$

Приводимо отримані функції до вигляду, зручному для реалізації на елементах І-НЕ:

$$f_1 = \overline{\overline{C} \vee \overline{Q} \vee \overline{J}} = C \cdot \overline{Q} \cdot J;$$

$$f_2 = \overline{\overline{C} \vee \overline{Q} \vee \overline{K}} = C \cdot \overline{Q} \cdot K.$$

Отримана схема тригера показана на рис. 6.9.

Синхронні тригери можуть мати асинхронні входи попередньої установки тригера в 0 (вхід  $R$ ) і в 1 (вхід  $S$ ). Сигнали, що надходять на ці входи, незалежно від стану інших входів тригера переключають його в новий стан, тобто мають пріоритет стосовно інших сигналів.

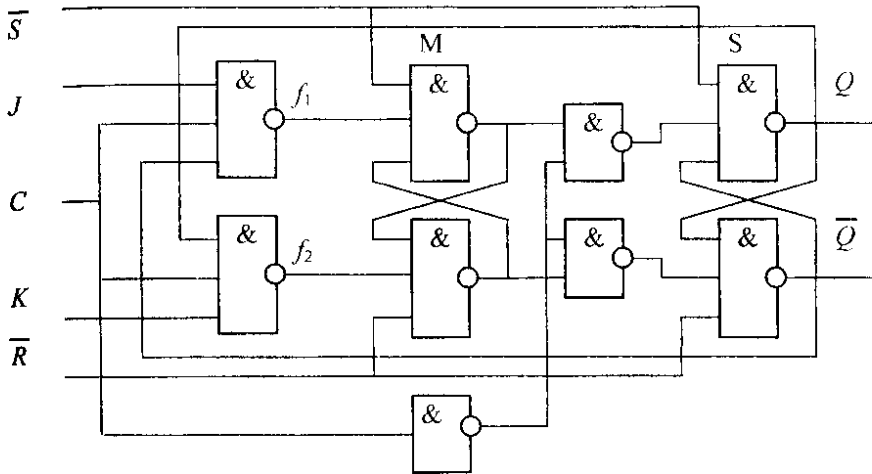


Рис. 6.9

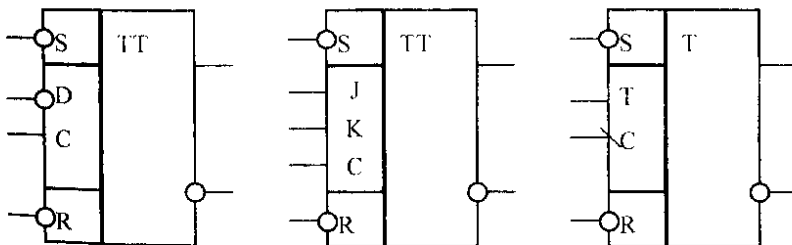


Рис. 6.10

Позначення асинхронних входів записуються за позначенням логічних входів тригера. Наприклад, T-тригер з асинхронними входами позначається

TRS. На рис. 6.3 – 6.5 асинхронні  $R$  і  $S$ -входи показані пунктиром. При графічному зображенні тригерів на функціональних схемах логічні сигнали синхронних тригерів показують в одному полі з тактуючим сигналом  $C$ , а асинхронні входи – на окремих полях. Як приклад на рис. 6.10 показані графічні зображення DRS-, JKRS-, TRS-тригерів.

Якщо установка тригера здійснюється нульовим сигналом, то відповідний вхід позначається кружком (на рис. 6.10 – це DRS і TRS-тригери). Трикутником на вході  $C$  вказується, по якому перепаді здійснюється переключення тригера. На рис. 6.10 DRS і TRS-тригери спрацьовують по позитивному перепаді керуючого сигналу, а JKRS-тригер – по негативному.

### Підготовка до роботи

1. Побудувати і замалювати в протоколі тригери, керовані рівнем тактуючого сигналу, на елементах І-НЕ по таблицях переходів (див. табл. 6.1 – 6.4). Для кожної з отриманих схем за допомогою часових діаграм визначити час переключення тригера, мінімальну тривалість і максимальну частоту тактуючих сигналів. Затримку сигналів елементом вважати рівною  $t$ . Замалювати умовне графічне зображення тригерів.

2. Побудувати і замалювати в протоколі тригери з внутрішньою затримкою на елементах І-НЕ по MS-схемі відповідно до таблиць переходів (див. табл. 6.5 – 6.8). Для кожної з отриманих схем визначити, по якому фронті тактуючого сигналу переключається тригер, час його переключення, мінімальну тривалість і максимальну частоту тактуючих сигналів. Пунктирними лініями на схемах показати, як організуються асинхронні входи тригерів  $R$  і  $S$ . Замалювати умовне графічне зображення отриманих тригерів.

3. Побудувати і замалювати тригер на елементах АБО-НЕ, виконаний за схемою трьох тригерів відповідно до таблиці варіантів (табл. 6.9). Для визначення номера варіанта необхідно одержати п'ять молодших розрядів номера залікової книжки в двійковій системі числення і вважати цілим числом (від 0 до 31). Для двох побудованих тригерів замалювати часові діаграми сигналів на виходах всіх елементів.

Визначити час переключення тригерів, мінімальну тривалість і максимальну частоту тактуючих сигналів.



### Порядок виконання роботи

1. Побудувати моделі декількох (за завданням викладача) схем тригерів, отриманих при виконанні пп. 1, 2 і 3 теоретичного завдання, використовуючи моделюючий комплекс.
2. Дослідити правильність функціонування схем у різних режимах.
3. Визначити часові параметри схем за допомогою часових діаграм.

Таблиця 6.9

Таблиця варіантів для побудови синхронних тригерів

$X_1^s$	$X_2^s$	$Q^{s+1}$															
		Варіант															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	$\bar{Q}^s$	$Q^s$	1	1	$\bar{Q}^s$	$Q^s$	0	0	$\bar{Q}^s$	1	0	$\bar{Q}^s$	1	0	1
0	1	1	0	$\bar{Q}^s$	$Q^s$	0	1	$\bar{Q}^s$	$Q^s$	1	$Q^s$	$\bar{Q}^s$	0	0	$\bar{Q}^s$	1	0
1	0	$Q^s$	1	0	$\bar{Q}^s$	$Q^s$	0	1	$\bar{Q}^s$	$\bar{Q}^s$	0	$Q^s$	1	0	0	$\bar{Q}^s$	0
1	1	$\bar{Q}^s$	$Q^s$	1	0	$\bar{Q}^s$	$\bar{Q}^s$	0	1	$\bar{Q}^s$	1	0	$\bar{Q}^s$	1	0	0	$\bar{Q}^s$

Продовження табл. 6.9

$X_1^s$	$X_2^s$	$Q^{s+1}$															
		Варіант															
		16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	$\bar{Q}^s$	0	1	0	$\bar{Q}^s$	0	1	1	$\bar{Q}^s$	*	0	1	$\bar{Q}^s$	0	*	$Q^s$
0	1	1	$\bar{Q}^s$	1	0	1	$\bar{Q}^s$	0	0	1	$\bar{Q}^s$	*	*	1	$\bar{Q}^s$	0	$\bar{Q}^s$
1	0	0	1	0	$\bar{Q}^s$	1	1	$\bar{Q}^s$	*	0	1	$\bar{Q}^s$	0	*	1	$\bar{Q}^s$	*
1	1	0	0	$\bar{Q}^s$	1	0	1	1	$\bar{Q}^s$	*	0	1	$\bar{Q}^s$	0	*	1	1

### Контрольні питання

1. Скласти таблиці переходів RS-, R-, S-, E-, D-, T-, DV- і JK-тригерів.
2. У чому різниця між синхронними й асинхронними тригерами?
3. У чому різниця між синхронними тригерами, керованими рівнем тактуючого сигналу, і синхронними тригерами з внутрішньою затримкою? У яких випадках використовуються зазначені типи тригерів?
4. Пояснити роботу синхронних тригерів, виконаних по MS-схемі і за схемою тригерів.

5. Указати різницю між синхронними й асинхронними входами асинхронного тригера.
6. Пояснити, як будується часова діаграма роботи тригера.
7. Охарактеризувати етапи проектування тригерних схем. Побудувати тригери по заданій таблиці переходів.
8. Назвати основні часові характеристики тригерів.
9. Порівняти по швидкодії тригери, виконані по MS-схемі і за схемою трьох тригерів. Які часові умови є основними для правильної роботи цих тригерів?
10. Як перейти до базису I-НЕ і АБО-НЕ, якщо функція представлена в МДНФ?
11. Указати, у яких випадках у тригерних схемах можливе явище гонок. У чому їх сутність? Які існують способи усунення гонок?
12. Як побудувати T-тригер на основі RS-, D- і JK-тригерів?
13. Які зміни необхідно внести в схему тригера з внутрішньою затримкою, щоб змінити фронт тактуючого сигналу, по якому переключається тригер?
14. Чи можна в цифровому пристрої замінити синхронний RS-тригер на синхронний JK-тригер (E-, R-, S-тригер), не порушуючи правильності роботи пристрою?

### Література

1. Алексенко А.Г. Основы микроэлектроники. – М.; Сов. радио, 1977. – 405 с.
2. Букреев И.Н., Мансуров В.М., Горячев В.И. Микроэлектронные схемы цифровых устройств. – М.; Сов. радио, 1975. – 368 с.
3. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум. – К.: Вища школа., 1990. – 215 с.

## 7. Лабораторна робота №6

### ПРОЕКТУВАННЯ І ДОСЛІДЖЕННЯ РЕГІСТРІВ

*Ціль роботи* – вивчити схемні різновиди регістрів, мікрооперацій, виконуваних на них, опанувати методи проектування і дослідження регістрів.

#### Теоретичні відомості

**Регістром** називається упорядкована послідовність запам'ятовуючих елементів (тригерів), призначена для збереження слів і виконання мікрооперацій над ними.

Число розрядів у регістрі називають його довжиною. У  $n$ -розрядному регістрі може бути записано  $2^n$  різних слів, тобто регістр може знаходитися в  $2^n$  різних станах. Наприклад, трирозрядний регістр може перебувати в таких станах: 000, 001, 010, 011, 100, 101, 110, 111.

Найбільше часто на регістрах виконують наступні мікрооперації:

$Y_1$  – встановлення початкового стану (наприклад, нульового);

$Y_2$  – прийом (запис) слова;

$Y_3$  – логічне множення двох слів (порозрядна кон'юнкція);

$Y_4$  – логічне додавання двох слів (порозрядна диз'юнкція);

$Y_5$  – порозрядна сума двох слів по модулю 2;

$Y_6$  – зсув слова на  $i$  розрядів;

$Y_7$  – інвертування розрядів слова;

$Y_8$  – видача слова в прямому коді;

$Y_9$  – видача слова в зворотному коді;

$Y_{10}$  – видача слова в парафазному коді.

Узагальнена логічна структурна схема регістра показана на рис. 7.1.

На схемі прийняті такі позначення: КС – комбінаційна схема;  $Y_1 - Y_m$  – сигнали мікрооперацій;  $X_i$  – комбінаційні входи регістра;  $Z_i$  – інформаційні входи регістра;  $A$  і  $B$  – інформаційні входи тригерів;  $C$  – тактуючі входи тригерів. Тригери утворюють власне регістр.

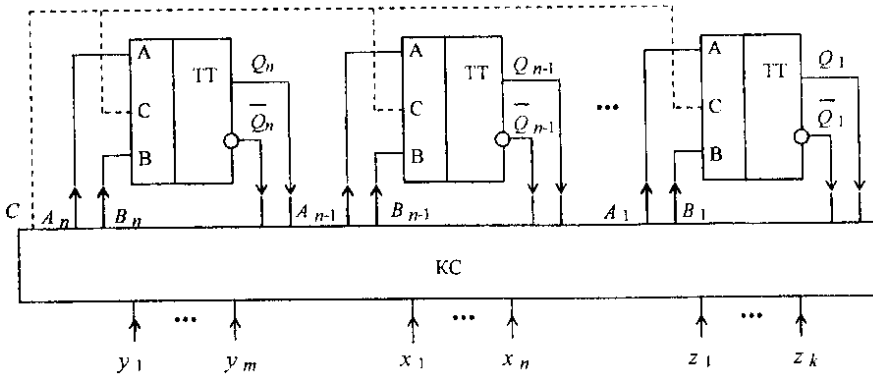


Рис. 7.1. Загальна структурній схема регістра

Складність КС залежить від виду виконуваних мікрооперацій і від типів тригерів. Іноді КС може бути відсутня.

Регістри можуть бути побудовані на тригерах різного типу, наприклад, на RS-, JK-, D- і T-тригерах. У залежності від виконуваних мікрооперацій для побудови регістрів можна використовувати тригери з різною внутрішньою організацією (синхронні і асинхронні, із внутрішньою затримкою і без неї і т. ін.).

Регістри, на яких виконуються мікрооперації зсуву, називаються *зсувними*. Зсув слова може бути здійснений вліво (вбік старших розрядів) або вправо (вбік молодших розрядів) на  $i$  розрядів одночасно, де  $i = \overline{1, n-1}$ . Регістри, що мають ланцюги як лівого, так і правого зсуву, називаються *реверсивними*.

Звичайно при виконанні мікрооперацій усі розряди регістра працюють однаково.

Мікрооперація  $Y_1$  складається в установці кожного розряду або в 0, або в 1. Для виконання цієї мікрооперації доцільно використовувати асинхронні входи R- і S-тригерів.

При виконанні мікрооперації  $Y_2$  у  $i$ -й розряд регістра записується цифра  $X_i$ , тобто (див. рис. 3.1)

$$Q_i^{S+1} = X_i,$$

де  $Q_i^{S+1}$  - стан  $i$ -го тригера в  $S+1$ -й момент часу (після виконання мікрооперації);  $X_i$  - значення сигналу на  $i$ -м вході регістра в  $S$ -й момент часу (перед виконанням мікрооперації).

Для мікрооперацій  $Y_3, Y_4$  і  $Y_5$  можна відповідно записати:

$$Q_i^{S+1} = Q_i^S \& X_i^S,$$

$$Q_i^{S+1} = Q_i^S \vee X_i^S,$$

$$Q_i^{S+1} = Q_i^S + X_i^S,$$

де  $Q_i$  – стан  $i$ -го регістра в  $S$ -й момент часу (перед виконанням мікрооперації).

Мікрооперація  $Y_6$  зсуву слова на  $j$  розрядів полягає в наступному:

$$Q_i^{S+1} = Q_{i-j}^S \quad \text{-- при зсуві вліво;}$$

$$Q_i^{S+1} = Q_{i+j}^S \quad \text{-- при зсуві вправо.}$$

При виконанні мікрооперації  $Y_7$  здійснюється таке перетворення інформації в  $i$ -м розряді регістра:

$$Q_i^{S+1} = \overline{Q_i^S}.$$

У процесі виконання мікрооперацій  $Y_8$ ,  $Y_9$  і  $Y_{10}$  стан регістра не змінюється. Для видачі слова в прямому коді до виходів регістра варто підключити прямі виходи тригерів  $Q_i$ , у зворотному – інверсні виходи  $\overline{Q_i^S}$ , а в парафазному – і прямі й інверсні виходи. КС для виконання мікрооперацій  $Y_8$ ,  $Y_9$  показана на рис. 7.2.

Проектування регістрів зводиться до вибору типу тригерів і синтезу КС, що формує функції збудження тригерів при виконанні заданих мікрооперацій. Синтез КС, що забезпечує виконання однієї мікрооперації, при використанні асинхронних тригерів для побудови регістра можна здійснити таким чином:

1) скласти таблицю переходів тригера з урахуванням рівнів сигналів, що забезпечують його переключення;

2) скласти таблицю переходів  $i$ -го розряду регістра при виконанні заданої мікрооперації, у якій для  $S$ -го моменту часу необхідно відобразити всі можливі комбінації значень аргументів функцій збудження тригера  $X_i$ ,  $Y_j$  і

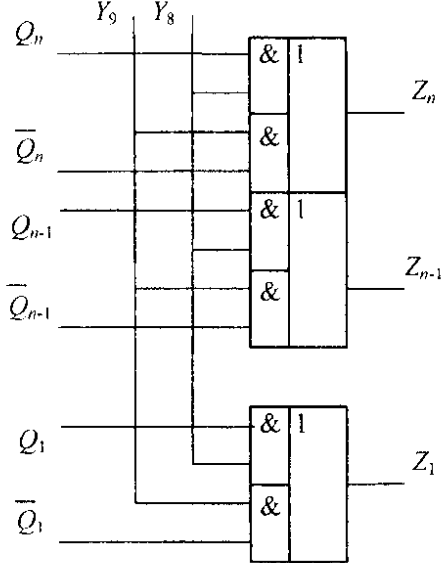


Рис. 7.2

г.д. і стану  $i$ -го розряду регістра  $Q_i^S$ , а для  $S+1$ -го моменту часу – тільки відповідні стани  $i$ -го розряду регістра  $Q_i$ ;

3) у кожнім рядку отриманої таблиці записати значення функції збудження тригера (відповідно до таблиці його переходів), що забезпечують необхідний перехід тригера зі стану  $Q_i^S$  у стан  $Q_i^{S+1}$  ( $Q_i^S \rightarrow Q_i^{S+1}$ );

4) записати СДНФ функцій збудження тригера;

5) знайти МДНФ функцій збудження тригера;

6) при необхідності перейти від МДНФ отриманих функцій до операторної форми їх представлення в заданому елементному базисі;

7) побудувати функціональну схему регістра.

Загальна логічна структура регістра в розглянутому випадку відповідає рис. 7.1, але тригери не мають тактуючих входів ( $C$ ).

Розглянемо процес синтезу КС для виконання мікрооперації  $Y_3$ , коли регістр побудований на асинхронних RS-тригерах. Графічно таблиця переходів RS-тригера показана на рис. 7.3 (тут знаком \* відзначено довільні значення  $R$  і  $S$ ).

Оскільки перетворення інформації в  $i$ -м розряді регістра при виконанні мікрооперації  $Y_3$  визначається виразом  $Q_i^{S+1} = Q_i^S \& X_i$ , аргументами функцій збудження тригера  $i$ -го розряду є значення  $Q_i^S$ ,  $X_i$ , а також значення сигналу мікрооперації  $Y_3$ .

Заносимо в табл. 7.1 усі комбінації значень  $Y_3$ ,  $X_i$  і  $Q_i^S$ . При  $Y_3=0$  у стовпці  $Q_i^{S+1}$  для кожного рядка повторюємо значення  $Q_i^S$ . Далі (при  $Y_3=1$ ) зазначений стовпець заповнюємо відповідно до виразу  $Q_i^{S+1} = Q_i^S \& X_i$ . З огляду на закон переходів RS-тригера (див. рис. 7.3), у стовпці  $R_i$  і  $S_i$  отриманої таблиці (табл. 7.1) для кожного рядка записуємо значення функцій збудження, що забезпечують необхідний перехід  $Q_i^S \rightarrow Q_i^{S+1}$ . Наприклад, для п'ятого рядка перехід  $Q_i^S \rightarrow Q_i^{S+1}$  має вигляд  $1 \rightarrow 0$ . Отже, відповідно до рис. 7.3 одержимо  $R_i=1$ ,  $S_i=0$ .

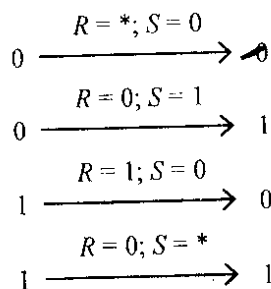


Рис. 7.3

Таблиця 7.1

Таблиця переходів  $i$ -го розряду регістра при виконанні мікрооперації  $Y_3$

№п/п	$Y_3$	$X_i^S$	$Q_i^S$	$Q_i^{S+1}$	$R$	$S$
0	0	0	0	0	*	0
1	0	0	1	1	0	*
2	0	1	0	0	*	0
3	0	1	1	1	0	*
4	1	0	0	0	*	0
5	1	0	1	0	1	0

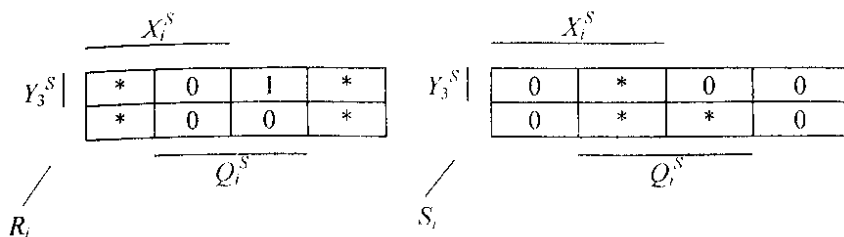


Рис. 7.4

Після оптимального до визначення функцій  $R_i$ ,  $(Y_i, X_i, Q_i^S)$  і  $S_i^S$  ( $Y_3, X_i, Q_i^S$ ) і їх мінімізації, виконаної з використанням діаграм Вейча (рис. 7.4), одержуємо (опускаючи індекси  $S$ ):

$$R_i = Y_3 \cdot \bar{X}_i; \quad S_i = 0.$$

Функціональна схема  $i$ -го розряду регістра показана на рис. 7.5.

Усі розряди регістра будуть однаково.

Якщо в МДНФ функції збудження тригера  $i$ -го розряду як аргумент входять стани  $Q_i^S$ , де  $i = 1, n$ , використовувати тригери повинні мати внутрішню затримку.

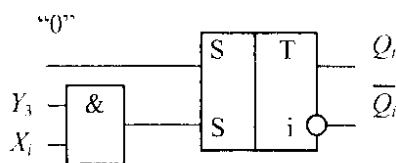


Рис. 7.5

Нехай для побудови регістра використовуються асинхронні тригери Т-типа. Виконаємо синтез КС для реалізації мікрооперації  $Y_2$ .

Порядок переключення Т-тригера показаний на рис. 7.6.

Мікрооперація  $Y_2$  складається в перетворенні виду  $Q_i^{S+1} = X_i^S$ , тобто аргументами функцій збудження тригерів є значення  $X_i^S$  і  $Y_2$ .

Складемо таблицю переходів  $i$ -го розряду регістра (табл. 7.2).

У даному випадку в стовпець  $Q_i^{S+1}$  переносимо значення  $Q_i^S$  при  $Y_2 = 0$  і значення  $X_i^S$  при  $Y_2 = 1$ .

СДНФ і МДНФ функції мають вигляд:

$$T_i = Y_2 \cdot \bar{X}_i \cdot Q_i \vee Y_2 \cdot X_i \cdot \bar{Q}_i.$$

Функціональна схема розряду регістра показана на рис. 7.7.

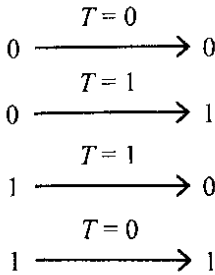


Рис. 7.6

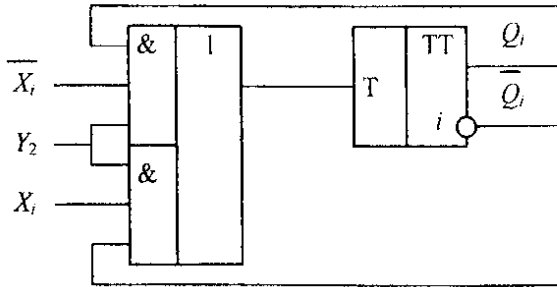


Рис. 7.7

Таблиця 7.2

Таблиця переходів  $i$ -го розряду регістра при виконанні мікрооперації  $Y_2$

№ <sub>nn</sub>	$Y_2$	$X_i^S$	$Q^S$	$Q^{S+1}$	$T_i$
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	0	0
3	0	1	1	1	0
4	1	0	0	0	0
5	1	0	1	0	1
6	1	1	0	1	1
7	1	1	1	1	0

Таблиця 7.3

Таблиця переходів  $i$ -го розряду регістра при виконанні мікрооперації  $Y_5$

№ <sub>nn</sub>	$Y_5$	$X_i^S$	$Q^S$	$Q^{S-1}$	$T_i$
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	0	0
3	0	1	1	1	0
4	1	0	0	0	0
5	1	0	1	1	0
6	1	1	0	1	1
7	1	1	1	0	1

При побудові КС для виконання мікрооперації  $Y_5$  на такому ж регістрі одержимо таблицю переходів  $i$ -го розряду (табл. 7.3), з якої випливає:

$$T_i = Y_5 \cdot X_i \cdot Q_i \vee Y_5 \cdot X_i \cdot \overline{Q_i} = Y_5 \cdot X_i$$

Функціональна схема розряду регістра, що виконує мікрооперацію  $Y_5$ , зображена на рис. 7.8.

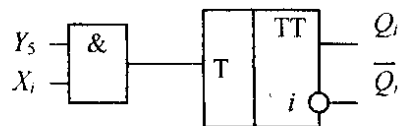


Рис. 7.8

Для регістра, на якому виконується кілька мікрооперацій з використанням однойменних входів тригерів, узагальнені функції збудження тригерів є диз'юнкцією однойменних функцій збудження, що відповідають окремим мікроопераціям. Наприклад, функціональна схема  $i$ -го розряду регіст-



ра, на якому можна виконувати мікрооперації  $Y_2$  і  $Y_5$ , має вигляд, показаний на рис. 7.9.

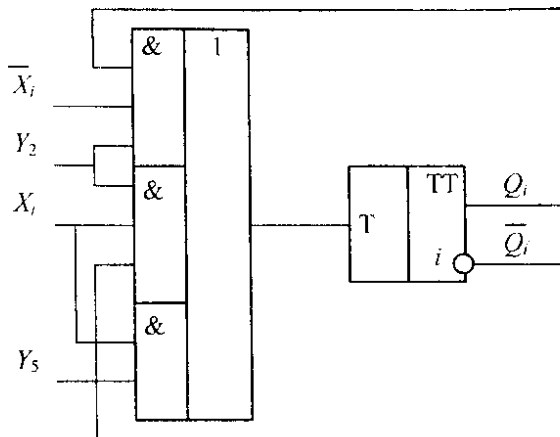


Рис. 7.9

Регістри на синхронних тригерах функціонують наступним чином. Спочатку на КС (див. рис. 7.1) подається сигнал (потенціал з рівнем логічної одиниці мікрооперації  $Y_i$ ). Потім, коли на інформаційних входах тригерів встановлюються значення функцій збудження, що забезпечують необхідне переключення тригерів, на їх тактуючі входи надходить сигнал синхронізації, під дією якого тригери переключаються.

Якщо на реєстрі виконується тільки одна мікрооперація з використанням інформаційних входів, які тактуються, змінюється тільки п.2 – синтез КС. У цьому випадку для спрощення КС у таблицю переходів  $i$ -го розряду реєстра не включають значення сигналу мікрооперації  $Y_i$ . Цей сигнал подається безпосередньо на тактуючі входи (С) тригерів, тобто виконує функції синхронізуючого сигналу.

Якщо на реєстрі виконуються дві і більш мікрооперації з використанням однойменних інформаційних входів, які тактуються, то синтез КС залишається таким самим, як і для використання асинхронних тригерів. Для кожної мікрооперації функції збудження тригерів знаходяться окремо, а диз'юнкція отриманих однойменних функцій є узагальненими функціями збудження тригера.

Нехай на регістрі з використанням входів  $J$  і  $K$  синхронних JK-тригерів повинна виконуватися тільки одна мікрооперація  $Y_4$ .

З урахуванням порядку переключення JK-тригера (рис. 7.10) складаємо таблицю переходів  $i$ -го розряду регістра (табл. 7.4). Після до визначення до мінімізації функцій  $J_i$  і  $K_i$ , одержуємо  $K_i=0$ ;  $J_i=X_i$ .

Один розряд регістра показаний на рис. 7.11.

Побудуємо регістр із використанням JK-тригера, на якому можна виконувати мікрооперації зсуву слова на один розряд вліво ( $Y_{6л}$ ) і на один розряд вправо ( $Y_{6п}$ ). Для мікрооперації  $Y_{6л}$  за допомогою табл. 7.5 і рис. 7.12 одержуємо:

$$J_i = Y_{6л} \cdot Q_{i-1}; \quad K_i = Y_{6л} \cdot \overline{Q_{i-1}}$$

Таблиця 7.4

Таблиця переходів  $i$ -го розряду регістра при виконанні мікрооперації  $Y_4$

№ <sub>пн</sub>	$X_i$	$Q_i$	$Q_{i-1}$	$J_i$	$K_i$
0	0	0	0	0	*
1	0	1	1	*	0
2	1	0	1	1	*
3	1	1	1	*	0

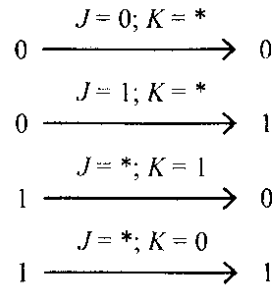


Рис. 7.10

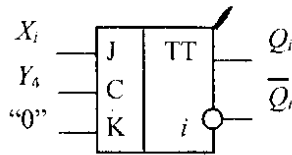


Рис. 7.11

Таблиця 7.5

Таблиця переходів  $i$ -го розряду регістра при виконанні мікрооперації  $Y_{6л}$

№ <sub>пн</sub>	$Y_{6л}^S$	$Q_{i-1}^S$	$Q_i^S$	$Q_{i+1}^S$	$J_i$	$K_i$
0	0	0	0	0	0	*
1	0	0	1	1	*	0
2	0	1	0	0	0	*
3	0	1	1	1	*	0
4	1	0	0	0	0	*
5	1	0	1	0	*	1
6	1	1	0	1	1	*
7	1	1	1	1	*	0

Очевидно, що для мікрооперації  $Y_{6л}$  у виразах для  $J_i$  і  $K_i$  значення  $Q_{i-1}$  і  $\overline{Q_{i-1}}$  варто замінити відповідно на  $Q_{i+1}$  і  $\overline{Q_{i+1}}$ . Диз'юнкція однойменних функцій збудження тригера для мікрооперацій  $Y_{6л}$  і  $Y_{6п}$  дає узагальнені функції збудження  $i$ -го розряду проєктованого регістра:

$$J_i = Y_{6л} \cdot Q_{i-1} \vee Y_{6п} \cdot \overline{Q_{i+1}}; \quad K_i = Y_{6л} \cdot \overline{Q_{i-1}} \vee Y_{6п} \cdot Q_{i+1}$$

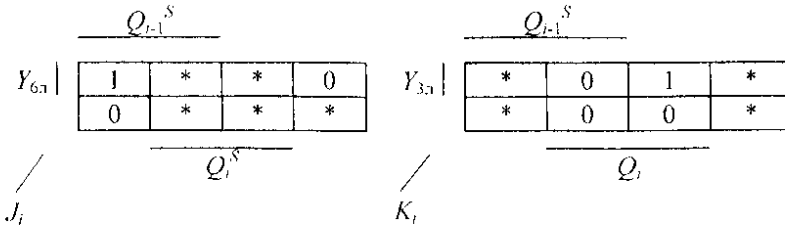


Рис. 7.12

Часовими характеристиками регістрів є:  $t_{mo}$  – час виконання мікрооперації;  $f_{kc}$  – максимальна частота надходження керуючих сигналів. Для регістрів на асинхронних тригерах ними є сигнали мікрооперацій, а для регістрів на синхронних тригерах – синхронізуючі сигнали.

### Підготовка до роботи

1. Побудувати і замалювати в протоколі один розряд регістрів на асинхронних T- і RS-тригерах, кожний з яких повинний забезпечувати виконання мікрооперацій:  $Y_1, Y_2, Y_3, Y_4, Y_5, Y_6$ .

Визначити складність  $L$  комбінаційних схем для одного розряду регістрів ( $L$  – сумарне число входів усіх логічних елементів).

Скласти часові діаграми роботи регістрів з урахуванням часу  $t_3$  затримки сигналів логічним елементом, часу  $t_T$  переключення тригера і визначити часові характеристики регістрів  $t_{mo}$  і  $f_{kc}$ .

2. На асинхронному тригері побудувати один розряд регістра, призначеного для виконання трьох мікрооперацій відповідно до табл. 7.6. Для отриманої схеми визначити  $L, t_{mo}$  і  $f_{kc}$ . Для визначення номера варіанта необхідно одержати п'ять молодших розрядів номера залікової книжки в двійковій системі числення і вважати цілим числом (від 0 до 31).

3. Побудувати трирозрядні регістри на синхронних D- і JK-тригерах, кожний з яких призначений для виконання мікрооперацій:  $Y_3, Y_6, Y_7$  і  $Y_{10}$ . Для отриманих схем визначити  $L, t_{mo}$  і  $f_{kc}$ .

4. Побудувати 4-розрядний регістр на синхронних тригерах, призначений для виконання мікрооперацій зсуву вліво на  $M$  розрядів і вправо на  $N$  розрядів за один такт відповідно до табл. 7.6.

Таблиця 7.6

Таблиця варіантів

Номер варіанта	Для п.2 завдання		Для п.4 завдання		
	Тип тригерів	Мікрооперація	Тип тригерів	M	N
1	RS	$Y_1, Y_2, Y_3$	JK	1	1
2	T	$Y_1, Y_2, Y_3$	D	1	2
3	RS	$Y_1, Y_2, Y_4$	JK	1	3
4	T	$Y_1, Y_2, Y_4$	D	1	4
5	RS	$Y_1, Y_2, Y_5$	JK	2	1
6	T	$Y_1, Y_2, Y_5$	D	2	2
7	RS	$Y_1, Y_2, Y_7$	JK	2	3
8	T	$Y_1, Y_2, Y_7$	D	2	4
9	RS	$Y_1, Y_2, Y_{10}$	JK	3	1
10	T	$Y_1, Y_2, Y_8$	D	3	2
11	RS	$Y_1, Y_2, Y_9$	JK	3	3
12	T	$Y_1, Y_2, Y_9$	D	3	4
13	RS	$Y_1, Y_3, Y_4$	JK	4	1
14	T	$Y_1, Y_3, Y_4$	D	4	2
15	RS	$Y_1, Y_3, Y_5$	JK	4	3
16	T	$Y_1, Y_3, Y_5$	D	4	4
17	RS	$Y_1, Y_3, Y_7$	JK	1	2
18	T	$Y_1, Y_3, Y_7$	D	1	3
19	RS	$Y_1, Y_3, Y_8$	JK	1	4
20	T	$Y_1, Y_3, Y_9$	D	2	1
21	RS	$Y_1, Y_3, Y_{10}$	JK	2	2
22	T	$Y_2, Y_3, Y_9$	D	2	3
23	RS	$Y_2, Y_3, Y_4$	JK	2	4
24	T	$Y_2, Y_3, Y_4$	D	3	2
25	RS	$Y_2, Y_3, Y_5$	JK	3	2
26	T	$Y_2, Y_3, Y_5$	D	3	3
27	RS	$Y_2, Y_3, Y_7$	JK	3	4
28	T	$Y_2, Y_3, Y_7$	D	4	1
29	RS	$Y_2, Y_3, Y_8$	JK	4	2
30	T	$Y_2, Y_3, Y_8$	D	4	3
31	RS	$Y_2, Y_3, Y_{10}$	JK	4	4

### Порядок виконання роботи

1. Побудувати моделі декількох регістрів (за вказівкою викладача), отриманих при виконанні теоретичного завдання, переконатися в правильності їх функціонування.
2. Використовуючи асинхронне моделювання, замалювати діаграми сигналів на виходах елементів у різних точках схем регістрів і порівняти їх з часовими діаграмами, отриманими раніше.

### Контрольні питання

1. Що таке довжина регістра?
2. Скільки різних слів можна записати в регістр довжиною  $n$ ?
3. Які мікрооперації можна виконувати на регістрах?
4. Намалювати узагальнену логічну структуру регістра на синхронних і асинхронних тригерах.
5. Від чого залежить складність комбінаційної схеми регістра?
6. Які регістри називаються зсувними?
7. Охарактеризувати, які перетворення інформації здійснюються при виконанні різних мікрооперацій.
8. У яких випадках можна здійснити прийом слова на регістр в однофазному коді і у яких – у паралельному?
9. Охарактеризувати етапи синтезу комбінаційної схеми регістра.
10. Чим відрізняється синтез комбінаційної схеми при використанні для побудови регістра синхронних і асинхронних тригерів?
11. Побудувати регістр на тригерах заданого типу для виконання визначеної мікрооперації.
12. Як одержати функції збудження тригерів для регістра, призначеного для виконання декількох мікрооперацій?
13. Як перейти від МДНФ функцій до операторної форми їх представлення в елементному базисі I-НЕ (АБО-НЕ)?
14. Скласти таблицю переходів для RS-, JK-, T- і D- тригерів.
15. У яких випадках для побудови регістра необхідно використовувати тригери з внутрішньою затримкою?
16. У чому відмінність тригерів із внутрішньою затримкою і без неї?
17. Як одержати кожний з чотирьох тригерів RS-, JK-, T- і D-типів на основі трьох інших?
18. Назвати основні часові характеристики регістрів.
19. Як побудувати часову діаграму роботи регістрів з урахуванням часу затримки сигналів логічними елементами і часу переключення тригерів?

20. Як визначити основні часові характеристики регістрів?

### Література

1. Алексенко А.Г. Основы микроэлектроники. – М.; Сов. радио, 1977. – 405 с.
2. Букреев И.Н., Мансуров В.М., Горячев В.И. Микроэлектронные схемы цифровых устройств. – М.; Сов. радио, 1975. – 368 с.
3. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум. – К.: Вища школа., 1990. – 215 с.

## 8. Лабораторна робота №7

### ПРОЕКТУВАННЯ І ДОСЛІДЖЕННЯ ЛІЧИЛЬНИКІВ

*Ціль роботи* – вивчити різні типи лічильників у потенційній елементній базі, опанувати методами їх проектування, налагодження й експериментального дослідження.

#### Теоретичні відомості

*Лічильником* називають послідовну схему, призначену для виконання мікрооперації рахунка одиниць і збереження слів.

Кількість дозволених станів лічильників називають його *періодом, модулем або коефіцієнтом перерахування K*.

Лічильники можуть бути побудовані на основі рахункових тригерів зі спеціальними міжрозрядними зв'язками, зсувних регістрів (кільцеві лічильники) і багатостійких елементів. У даній роботі розглядаються лічильники двох перших типів.

Основними часовими характеристиками лічильників є:  $f_c$  – максимальна частота надходження рахункових сигналів і  $t_n$  – час переходу лічильника з одного стану в інший.

Лічильники зі спеціальними міжрозрядними зв'язками класифікуються по різних ознаках.

По характеру *мікрооперації рахунка* лічильники підрозділяються на *підсумовуючі, віднімаючі, реверсивні*.

При надходженні чергового рахункового сигналу  $x$  вміст підсумовуючого лічильника збільшується на 1, а віднімаючого – зменшується на 1. Реверсивний лічильник може виконувати мікрооперації підсумовування і вирахування в залежності від значення сигналу на керуючому вході  $Y$  (наприклад, при  $Y=1$  виконується підсумовування, а при  $Y=0$  – вирахування).

У залежності від основи *системи числення*, у якій здійснюється мікрооперація рахунка, розрізняють, наприклад, двійкові, двійково-п'ятиричні і двійково-десяткові лічильники і т. ін.

Лічильники класифікуються і по *схемних ознаках*. Для побудови лічильників у потенційній елементній базі застосовуються переважно синхронні тригери з внутрішньою затримкою, що дозволяє використовувати на один розряд двійкового лічильника один тригер.

Лічильники бувають *асинхронні і синхронні*.

В *асинхронних лічильниках* на тактуючі входи синхронних тригерів або на інформаційні входи асинхронних надходять сигнали з виходів сусідніх тригерів (можливо через логічні елементи). Тригери в таких лічильниках спрацьовують не одночасно, оскільки переключення одних тригерів починається тільки після зміни стану інших. У *синхронних* же лічильниках усі тригери переключаються одночасно під дією загального синхронізуючого сигналу, що надходить на тактуючі входи всіх тригерів одночасно.

За способом організації ланцюгів переносу (позики) між розрядами лічильники підрозділяються на наступні типи: з *послідовним*, з *наскрізним*, з *паралельним і груповим переносом*.

У лічильниках першого типу перенос (позика) у сусідній старший розряд формується тільки після переключення тригера в попередньому розряді, тобто такі лічильники асинхронні. При проектуванні асинхронних лічильників виникають труднощі, пов'язані з необхідністю аналізу не тільки логічного рівня сигналів, що формуються в схемі, але і моментів зміни рівнів сигналів. На структуру таких лічильників впливає внутрішня організація тригерів. На рис. 8.1, а показана функціональна схема  $n$ -розрядного підсумовуючого лічильника з послідовним переносом, побудованого на синхронних Т-тригерах, що переключаються по задньому фронту тактуючого сигналу. Часова діаграма зміни сигналів на виходах  $Q_i$  ( $i=1, 2, 3$ ) (без врахування часу переключення тригерів) показана на рис. 8.1, б.

Лічильники з іншою організацією ланцюгів переносів (позики) будуються за синхронним принципом. У лічильниках з рівнобіжним переносом аргументами функцій переносів для кожного розряду є тільки сигнали на виходах тригерів відповідних розрядів. Переноси для всіх розрядів лічильника формуються одночасно (за умови, що всі логічні елементи в схемі мають однаковий час переключення).

Ланцюги наскрізного переносу організуються таким чином, щоб функція переносу  $i$ -го розряду лічильника була аргументом функції переносу  $i+1$ -го розряду. У цьому випадку сигнали переносів для кожного розряду лічильника формуються по черзі, починаючи з молодших. Лічильники з наскрізним переносом вимагають меншого числа логічних елементів організації ланцюгів переносу, але поступаються лічильникам з рівнобіжним переносом у швидкодії.



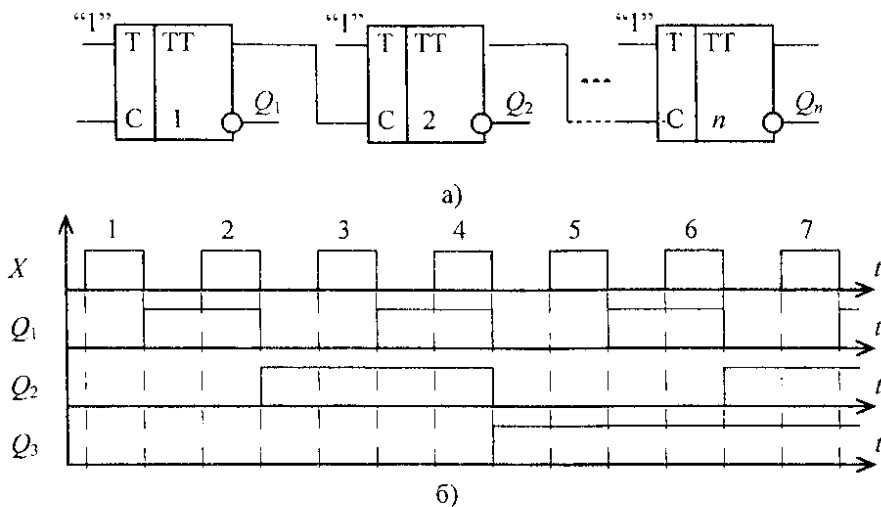


Рис. 8.1

У лічильниках із груповим переносом розряди розбиваються на групи (наприклад,  $n$  розрядів розбиваються на  $m$  груп). У межах однієї групи звичайно організується рівнобіжний перенос, а між групами – послідовний або наскрізний.

Якщо мікрооперація рахунка одиниць виконується в канонічній двійковій системі числення (в однорідній позиційній двійковій системі числення з природним порядком ваг) і лічильник має 2 дозволених стана, він називається лічильником із природним порядком рахунка по модулю 2.

Стани чотирирозрядного лічильника з природним порядком рахунка приведені в табл. 8.1.

Якщо мікрооперація рахунка виконується в неканонічних системах (наприклад, символічних, зі штучним порядком ваг і т. ін.), порядок рахунка вважається штучним. Стани чотирирозрядного лічильника зі штучним порядком рахунка по модулю 2, що виконує мікрооперацію рахунка в коді Грея, приведені в табл. 8.2.

З табл. 8.1 і 8.2 видно, що стани лічильників повторюються з періодом  $2n$ . Для скорочення таблиць станів у них заносяться тільки стани для одного періоду.

Таблиця 8.1

Таблиця станів лічильника з природним порядком рахунка

Кількість рахункових сигналів	Стан лічильника	
	підсумовуючого	віднімаючого
0	0000	0000
1	0001	1111
2	0010	1110
3	0011	1101
4	0100	1100
5	0101	1011
6	0110	1010
7	0111	1001
8	1000	1000
9	1001	0111
10	1010	0101
11	1011	0101
12	1100	0100
13	1101	0011
14	1110	0010
15	1111	0001
16	0000	0000
17	0001	1111
18	0010	1110
19	0011	1101
...	...	...

Таблиця 8.2

Таблиця станів лічильника зі штучним порядком рахунка

Кількість рахункових сигналів	Стан лічильника
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101
10	1111
11	1110
12	1010
13	1011
14	1001
15	1000
16	0000
17	0001
18	0011
19	0010
...	...

Лічильники з природним і штучним порядком рахунка можуть мати коефіцієнт перерахування  $K \neq 2^n$ . У цьому випадку природним вважається такий порядок рахунка, коли дозволеними є стани від 0 до  $K-1$  і мікрооперація рахунка виконується в канонічній двійковій системі числення.

Найбільш прості схеми лічильників із природним порядком рахунка, побудовані на основі тригерів з рахунковим входом (Т і JK-тригерів).

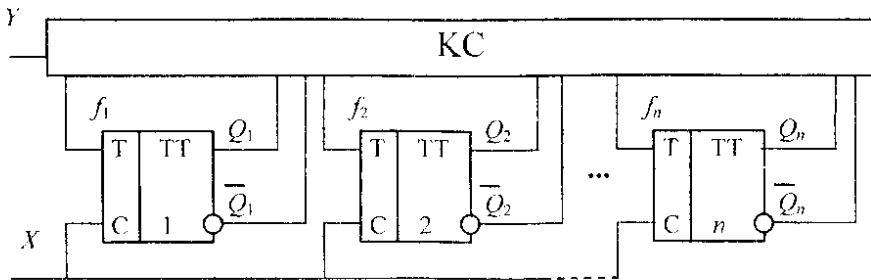


Рис. 8.2

На рис. 8.2 зображена узагальнена структура синхронного лічильника на Т-тригерах (КС – комбінаційна схема, що формує сигнали переносів  $f_i$ , що надходять на рахункові входи  $i$ -х тригерів).

У JK-тригерах рахунковий вхід організується з'єднанням входів  $J$  і  $K$ . З табл. 8.1 видно, що переключення тригера молодшого розряду здійснюється з приходом кожного рахункового сигналу, а інших тригерів – тільки в тому випадку, коли всі тригери молодших розрядів встановлені в 1 (підсумовуючий лічильник) або в 0 (віднімаючий лічильник).

Отже, для підсумовуючих лічильників із природним порядком рахунка по модулю  $2^n$ , що має ланцюги рівнобіжного переносу, перемикальні функції  $f_i$  мають вигляд:

$$f_i = Q_1 \cdot Q_2 \cdot \dots \cdot Q_{i-1}, \quad (i = \overline{2, n}) \quad (8.1)$$

для віднімаючих лічильників:

$$f_i = \overline{Q}_1 \cdot \overline{Q}_2 \cdot \dots \cdot \overline{Q}_{i-1}, \quad (i = \overline{2, n}) \quad (8.2)$$

а для реверсивних:

$$f_i = Q_1 \cdot Q_2 \cdot \dots \cdot Q_{i-1} \cdot Y \vee \overline{Q}_1 \cdot \overline{Q}_2 \cdot \dots \cdot \overline{Q}_{i-1} \cdot \overline{Y}, \quad (i = \overline{2, n}) \quad (8.3)$$

Для всіх типів лічильників  $f_1 = 1$ .

Функції (8.1 – 8.3) для лічильників з наскрізним переносом можуть бути представлені відповідно таким чином:

$$f_i = f_{i-1} \cdot Q_{i-1}, \quad (i = \overline{2, n}),$$

$$f_i = f_{i-1} \cdot \overline{Q}_{i-1}, \quad (i = \overline{2, n}),$$

$$f_i = f_i' \vee f_i'', \quad (i = \overline{2, n}),$$

де  $f_1 = 1$ ,  $f_2' = Q_1 \cdot Y$ ,  $f_2'' = \overline{Q}_1 \cdot \overline{Y}$ ,  $f_i' = f_{i-1} \cdot Q_{i-1}$ ,  $f_i'' = f_{i-1} \cdot \overline{Q}_{i-1}$ ,  $(i = \overline{3, n})$ .

Як приклад на рис. 8.3 показана функціональна схема підсумовуючого лічильника з рівнобіжним переносом на Т-тригерах, а на рис. 8.4 – реверсивного лічильника з наскрізним переносом на JK-тригерах (для  $n=4$ ).

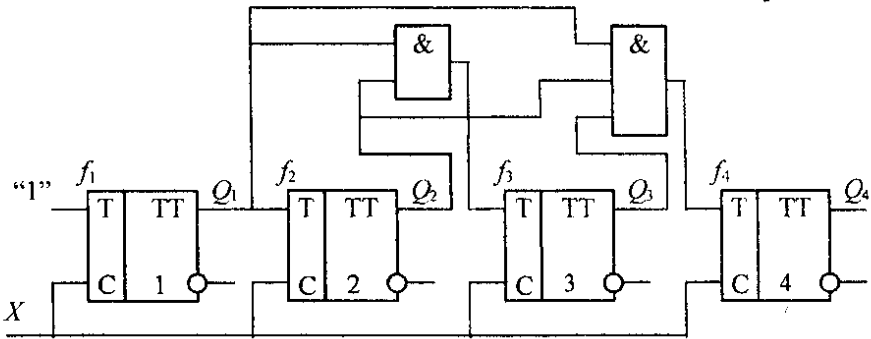


Рис. 8.3

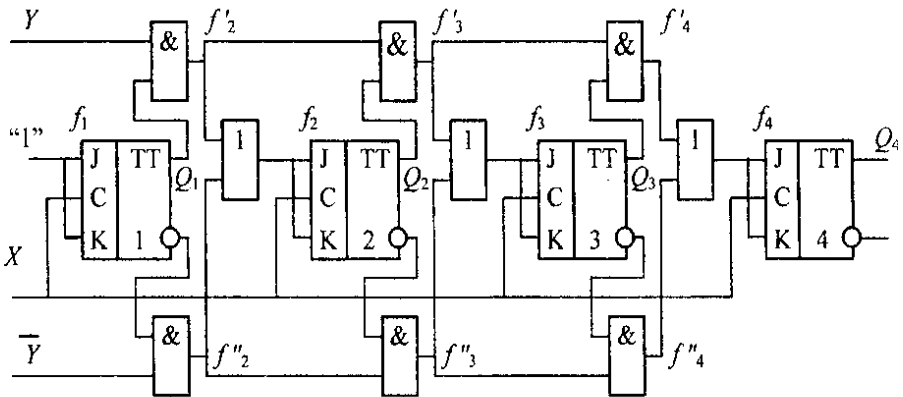


Рис. 8.4

Синхронні лічильники з довільним коефіцієнтом перерахування  $K$  і будь-яким порядком рахунка можна побудувати наступним чином:

- 1) визначити число розрядів  $n$  лічильника

$$n = \lceil \log_2 K \rceil,$$

де  $\lceil \log_2 K \rceil$  - ціле, не менше  $\log_2 K$ ;

- 2) вибрати тип тригерів і визначити комбінації інформаційних сигналів, що забезпечують переключення тригерів з одного стану в інший (графічно таблиця переходів RS-, D-, JK і T-тригерів показана на рис. 8.5, де зна-

ком \* відзначено довільне значення сигналу на інформаційному вході тригера);

JK- тригер	T- тригер	RS- тригер	D- тригер
$J = 0; K = *$	$T = 0$	$R = *; S = 0$	$D = 0$
$0 \longrightarrow 0$	$0 \longrightarrow 0$	$0 \longrightarrow 0$	$0 \longrightarrow 0$
$J = 1; K = *$	$T = 1$	$R = 0; S = 1$	$D = 1$
$0 \longrightarrow 1$	$0 \longrightarrow 1$	$0 \longrightarrow 1$	$0 \longrightarrow 1$
$J = *, K = 1$	$T = 1$	$R = 1; S = 0$	$D = 0$
$1 \longrightarrow 0$	$1 \longrightarrow 0$	$1 \longrightarrow 0$	$1 \longrightarrow 0$
$J = *, K = 0$	$T = 0$	$R = 0; S = *$	$D = 1$
$1 \longrightarrow 1$	$1 \longrightarrow 1$	$1 \longrightarrow 1$	$1 \longrightarrow 1$

Рис. 8.5

3) скласти таблицю переходів лічильника (табл. 8.3), записавши у відповідні стовпці в кожному рядку коди станів лічильника до надходження чергового рахункового сигналу ( $S$ -й момент часу); і після його надходження ( $S+1$ -й момент часу); наприклад, у 0-й рядкові для  $S$ -го моменту часу записати значення  $Q_i^S$  сигналів на виходах тригерів у вихідному стані, а для  $S+1$ -го -- значення  $Q_i^{S+1}$  після надходження першого рахункового сигналу; уміст стовпців  $Q_i^{S+1}$  0-й рядка перенести в стовпці  $Q_i^S$  1-го рядка, а в стовпці  $Q_i^{S+1}$  цього рядка записати значення виходів тригерів після надходження чергового (у даному випадку другого) рахункового сигналу і т. ін.;

4) для кожного  $i$ -го розряду лічильника записати в кожному  $j$ -му рядкові таблиці значення сигналів по інформаційних входах  $f_i$  тригера (функції збудження тригера), що забезпечують переключення тригера зі стану  $Q_i^S$  у стан  $Q_i^{S+1}$ , відповідно до таблиці його переходів;

Таблиця 8.3

Таблиця переходів лічильника

Номер набору	Стан лічильника	Функції збудження тригерів
$j$	$Q_n^S, \dots, Q_1^S$ $Q_n^{S+1}, \dots, Q_1^{S+1}$	$f_n f_{n-1} \dots f_1$
0		
1		
2		
3		
...		
$K-1$		

5) записати СДНФ функцій  $f_i$ , аргументами яких є значення  $Q_n^S, Q_{n-1}^S, \dots, Q_1^S$ ;

6) одержати МДНФ функцій  $f_i$ ;

7) при необхідності перейти до оперативної форми представлення функцій  $f_i$  у заданому елементному базисі;

8) побудувати функціональну схему лічильника.

Припустимо, що необхідно побудувати лічильник з коефіцієнтом перерахування  $K=6$ . Відповідно до п.1 знаходимо  $n=\lceil \log_2 6 \rceil = 3$ . Вибираємо тригери Т-типа. Нехай дозволеними для лічильника є стани 000, 001, 010, 011, 110, 111.

Таблиця 8.4

Таблиця переходів лічильника з коефіцієнтом перерахування  $K=6$ 

Номер набору $j$	Стан лічильника						Функції збудження тригерів		
	$Q_3^S$	$Q_2^S$	$Q_1^S$	$Q_3^{S+1}$	$Q_2^{S+1}$	$Q_1^{S+1}$	$T_3$	$T_2$	$T_1$
0	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0	1	1
2	0	1	0	0	1	1	0	0	1
3	0	1	1	1	1	0	1	0	1
4	1	1	0	1	1	1	0	0	1
5	1	1	1	0	0	0	1	1	1

Після виконання пп. 3 і 4 одержуємо таблицю переходів лічильника (табл. 8.4). Наприклад, для нульового рядка значення  $T_1$ ,  $T_2$  і  $T_3$  визначені таким чином. Переходи  $Q_1^S \rightarrow Q_1^{S+1}$ ,  $Q_2^S \rightarrow Q_2^{S+1}$ ,  $Q_3^S \rightarrow Q_3^{S+1}$  мають відповідно вид  $0 \rightarrow 1$ ,  $0 \rightarrow 0$  і  $0 \rightarrow 0$ . З рис. 8.5 для Т-тригера одержуємо  $T_1=1$ ,  $T_2=0$  і  $T_3=0$ . Записуємо СДНФ функцій  $T_i$ :

$$T_1 = 1;$$

$$T_2 = \overline{Q_3^S} \cdot \overline{Q_2^S} \cdot \overline{Q_1^S} \vee Q_3^S \cdot Q_2^S \cdot Q_1^S;$$

$$T_3 = \overline{Q_3^S} \cdot Q_2^S \cdot Q_1^S \vee Q_3^S \cdot Q_2^S \cdot Q_1^S.$$

Після мінімізації функцій, пропускаючи індекси S, одержуємо:

$$T_1 = 1;$$

$$T_2 = Q_3 \cdot Q_1 \vee \overline{Q_2} \cdot Q_1;$$

$$T_3 = Q_2 \cdot Q_1.$$

Функціональна схема лічильника зображена на рис. 8.6.

При великій розрядності лічильників розглянутий метод стає досить трудомістким, зокрема, при знаходженні МДНФ функції збудження тригерів.

Для побудови лічильників з коефіцієнтом перерахування  $K=2^n$  можна скористатися методом виключення надлишкових станів, що мало критичний до величини  $n$ . Для побудови лічильника необхідно:

1) визначити число розрядів  $n$  лічильника

$$n = \lceil \log_2 K \rceil,$$

де  $\lceil \log_2 K \rceil$  - ціле, не менше  $\log_2 K$ ;

2) вибрати тип лічильника (підсумовуючий або віднімаючий), спосіб організації переносів і записати функції  $f_i$  ( $i = \overline{1, n}$ ), для лічильників із природним порядком рахунка по модулю  $2^n$ ;

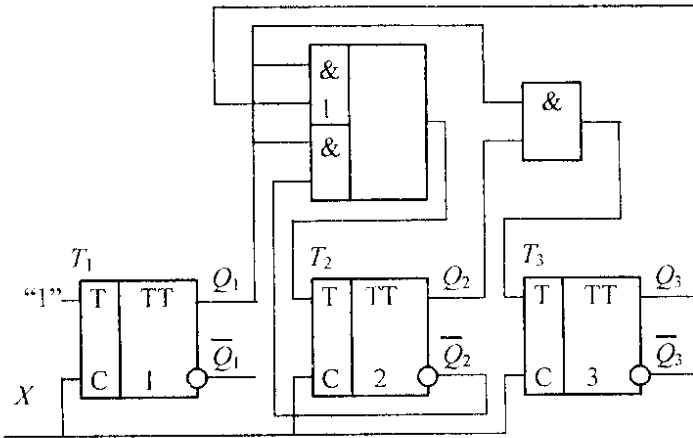


Рис. 8.6

3) скласти таблицю станів  $n$ -розрядного лічильника з природним порядком рахунка по модулю  $2^n$ ;

4) виключити з таблиці  $2^n - K$  станів, що слідують піряд;

5) стани, що передують штучному переходові, природного переходу і штучного переходу позначити відповідно А, В і С;

6) заповнити скорочену таблицю переходів лічильника (табл. 8.5), записавши в стовпці 3-5 відповідні значення  $Q_i$  (0 або 1) для станів А, В і С;

7) визначити і записати в скорочену таблицю функції переносів у  $i$ -ті розряди проектованого лічильника ( $f_i$ ) за наступним правилом:

а) якщо в  $i$ -му рядкові скороченої таблиці переходів значення  $Q_i$  збігаються для станів В і С, то

$$f_i^* = f_i,$$

де  $f_i$  – функція переносу в  $i$ -й розряд лічильника з природним порядком рахунка по модулю  $2^n$ ;

б) якщо в  $i$ -му рядковій таблиці значення  $Q_i$  не збігаються для станів В й С, але збігаються для станів А й В, то

$$f_i = f_i \vee f_a,$$

де  $f_a$  – конституента одиниці, що відповідає станів А (наприклад, при  $A=0101$  одержимо  $f_a = \bar{Q}_4 \cdot Q_3 \cdot \bar{Q}_2 \cdot Q_1$ );

в) якщо в  $i$ -му рядковій таблиці значення  $Q_i$  не збігаються як для станів В й С, так і для станів А й В, то

$$f_i^* = f_i \cdot \bar{f}_a,$$

8) перейти при необхідності до операторної форми представлення перемикальних функцій  $f_i$  і  $f_a$  у заданому елементному базисі (наприклад, перетворити МДНФ функцій до виду, зручному для реалізації на елементах І-НЕ з визначеним числом входів);

Таблиця 8.5

$i$	$Q_i$	Стан лічильника			$f_i^*$
		попереднє штучному переході А	природного переходу В	штучного переходу С	
1	2	3	4	5	6
1	$Q_1$				
2	$Q_2$				
⋮	⋮				
$n$	$Q_n$				

9) побудувати функціональну схему лічильника на рахункових тригерах заданого типу.

Якщо в лічильниках виключаються останні  $2^n - K$  станів (наприклад, у підсумовуючому лічильнику рахунок починається з нуля, закінчується числом  $K-1$ ), то вони є лічильниками з природним порядком рахунка по модулю  $K$ . При виключенні інших станів природний порядок рахунка порушується. У цьому випадку (наприклад, для підсумовуючого лічильника) число рахункових сигналів, що надходять, не відповідає числовому еквівалентові коду суми по модулю  $K$  даної кількості одиниць.

Нехай необхідно побудувати лічильник на Т-тригерах з коефіцієнтом перерахування  $K=6$ .



Відповідно до п.1 приведеної процедури  $n = \lceil \log_2 6 \rceil = 3$ .

Вибираємо паралельний спосіб формування переносів і записуємо функції  $f_i$  ( $i = 1, 3$ ) для підсумовуючого лічильника з природним порядком рахунка:

$$f_1 = 1, f_2 = Q, f_3 = Q_1 Q_2.$$

Складаємо таблицю станів трирозрядного лічильника (табл. 8.6) і виключаємо з неї два стани ( $2^3 - 6 = 2$ ), коди яких відзначені зірочкою.

Стан 011 позначаємо А, стан 100 – В і 110 – С.

Заповнюємо скорочену таблицю переходів лічильника (табл. 8.7) і знаходимо функції  $f_1, f_2$  і  $f_3$ . У 1-му та 3-му рядкові табл. 8.7 значення  $Q_i$  для станів В і С збігаються. Отже,  $f'_1 = f_1$  і  $f'_3 = f_3$ . У 2-му рядкові значення  $Q_i$  не збігаються ні для станів В та С, ні для станів А і В. Відповідно до правила (див. п.7) одержуємо  $f'_2 = f_2 \bar{f}_a$ , де  $f_a = \bar{Q}_3 Q_2 Q_1$ .

Функціональна схема лічильника показана на рис. 8.7.

Якщо мікрооперація рахунка виконується в двійково-кодованій системі числення (двійково-п'ятиричній, двійково-десяткової і т. ін.), то для побудови одного розряду лічильника необхідно не менш  $n = \lceil \log_2 K \rceil$  двійкових тригерів, де  $K$  – основа системи числення, цифри якого кодуються двійковим кодом. Наприклад, для побудови одного розряду десяткового лічильника потрібно не менш чотирьох двійкових тригерів. Таким чином, один розряд  $K$ -го лічильника являє собою двійковий лічильник з коефіцієнтом перерахування  $K$ , що виконує мікрооперацію рахунка у відповідному коді.

Для встановлення лічильників у вихідний стан перед виконанням мікрооперації рахунка використовуються звичайно асинхронні входи тригерів

Таблиця 8.6

Таблиця станів трирозрядного лічильника

Кількість рахункових сигналів	Стани лічильника		
	$Q_3$	$Q_2$	$Q_1$
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	*1	0	0
5	*1	0	1
6	1	0	0

А  
В  
С

Таблиця 8.7

Скорочена таблиця переходів лічильника з коефіцієнтом перерахування  $K = 6$

$i$	$Q_i$	Стани лічильника			$f'$
		А	В	С	
1	$Q_1$	1	0	0	$f'_1 = f_1$
2	$Q_2$	1	0	1	$f'_2 = f_2 \bar{f}_a$
3	$Q_3$	0	1	1	$f'_3 = f_3$

R- і S-. На рис. 8.7 ланцюг встановлення нульового стану лічильника показаний пунктирною лінією.

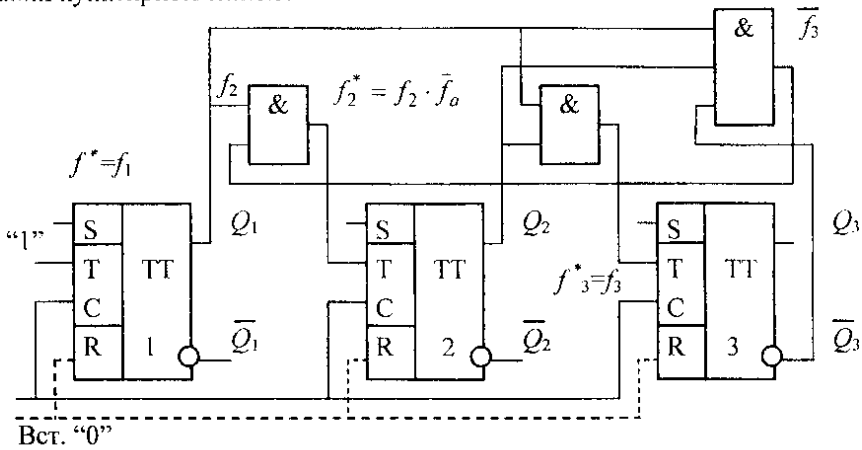


Рис. 8.7

### Підготовка до роботи

1. Побудувати і замалювати в протоколі наступні лічильники з природним порядком рахунка по модулю 32 :

- підсумовуючий лічильник з наскрізним переносом;
- віднімаючий лічильник з наскрізним переносом;
- підсумовуючий лічильник з паралельним переносом;
- віднімаючий лічильник з паралельним переносом;
- реверсивний лічильник з наскрізним переносом;
- реверсивний лічильник з паралельним переносом.

Побудувати підсумовуючий лічильник із груповим переносом (дві групи по чотири розряди).

Для кожного лічильника записати перемикальну функцію переносу (позики) у  $i$ -тий розряд.

Скласти часові діаграми роботи лічильників з урахуванням години затримки сигналів у логічних елементах  $t_3$  і години переключення тригерів  $t_T$ . Значення  $t_3$  і  $t_T$  вибрати довільно з урахуванням співвідношення  $t_3 < t_T$ .

На підставі часових діаграм визначити для кожного лічильника характеристики  $f_C$  і  $t_{об}$ , а також складність  $L$ -ланцюгів формування переносів (позики), де  $L$  – сумарне число входів усіх логічних елементів.

Таблиця 8.8

Таблиця станів лічильника.

Кількість рахункових сигналів	Стани лічильника			
	$Q_4$	$Q_3$	$Q_2$	$Q_1$
0	0	0	0	$h_1$
1	0	0	1	0
2	0	0	1	1
3	0	1	0	0
4	0	1	0	1
5	0	1	1	0
6	1	0	0	$h_2$
7	1	0	1	$h_3$
8	1	1	0	$h_4$
9	1	1	1	$h_5$

2. Побудувати лічильники на JK- і D-тригерах, які в процесі рахунка змінюють свої стани відповідно до табл. 8.8. Лічильники повинні мати ланцюга встановлення початкового стану  $000h_1$ .

Для одержання значень  $h_i$  (табл. 8.8) необхідно номер залікової книжки записати в двійковій системі числення і виділити п'ять молодших розрядів отриманого слова. Наприклад, якщо номер бригади - 5, то  $h_1=1$ ,  $h_2=0$ ,  $h_3=1$ ,  $h_4=0$ ,  $h_5=0$ .

Для отриманих схем визначити  $f_C$ ,  $t_n$ ,  $L$ .

3. Побудувати підсумовуючий лічильник на JK-тригерах з наскрізним переносом і природним порядком рахунка по модулю  $64-N$  ( $N$  – номер бригади), використовуючи метод виключення останніх надлишкових станів. Лічильник повинний мати ланцюг встановлення нульового стану. Для отриманої схеми визначити  $f_C$ ,  $t_n$ ,  $L$ .

### Порядок виконання роботи

1. Побудувати моделі і налагодити в режимі синхронного моделювання декілька (за завданням викладача) схем лічильників, отриманих при виконанні теоретичного завдання.

2. Досліджувати лічильники в режимі динамічного моделювання. Замалювати часові діаграми в протоколі, порівняти їх з отриманими раніше, зробити висновки по роботі.

### Контрольні питання

1. По яких ознаках можна класифікувати лічильники?
2. Написати вираз для функції формування переносів (позик) підсумовуючого, віднімаючого, реверсивного з наскрізним і паралельним переносом і природним порядком рахунка по модулю  $2^n$ .
3. Як одержати T-тригер на основі RS-, D- і JK- тригерів із внутрішньою затримкою?

4. Як перейти від МДНФ функцій до операторної форми при їх реалізації на елементах І-НЕ, АБО-НЕ?
5. Як враховуються набори, що відповідають забороненим станам лічильників, при мінімізації функцій збудження тригерів?
6. Скласти таблицю переходів RS-, JK-, D- і T- тригерів.
7. Як можна встановити лічильник у вихідний стан?
8. Охарактеризувати основні етапи проектування лічильників з довільною зміною станів і з виключенням надлишкових станів, що слідують підряд. Назвати переваги і недоліки зазначених методів.
9. Який порядок рахунка називається природним, а який - штучним?
10. Побудувати лічильник на RS-, JK-, D- і T- тригерах по заданій таблиці станів.
11. Назвати основні часові характеристики лічильників.
12. Від чого залежить коефіцієнт перерахування кільцевого лічильника?
13. Як побудувати часову діаграму роботи лічильника з урахуванням часу і затримки сигналів логічними елементами і часу переключення тригерів?
14. Як визначити час переходу лічильника з одного стану в інший і максимальну частоту надходження рахункових сигналів?
15. Яка внутрішня організація тригерів дозволяє використовувати на один розряд двійкового лічильника один тригер?
16. У яких випадках доцільна групова організація ланцюгів переносів у лічильниках?
17. Як побудувати  $n$ -розрядний  $K$ -й лічильник ( $K \neq 2^n$ )?
18. Як визначити кількість періодів і число станів, що відповідають кожному періодові, для кільцевого лічильника з заданою функцією зворотного зв'язку?
19. Вказати переваги і недоліки кільцевих лічильників.

### Література

1. Алексенко А.Г. Основы микроэлектроники. – М.; Сов. радио, 1977. – 405 с.
2. Букреев И.Н., Мансуров В.М., Горячев В.И. Микроэлектронные схемы цифровых устройств. – М.; Сов. радио, 1975. – 368 с.
3. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум. – К.: Вища школа., 1990. – 215 с.

## 9. Лабораторна робота №8

### ПРОЕКТУВАННЯ І ДОСЛІДЖЕННЯ СУМАТОРІВ

*Ціль роботи* – вивчити методи проектування суматорів, одержати навички в налагодженні та експериментальному дослідженні суматорів.

#### Теоретичні відомості

*Суматор* – операційний вузол, що виконує мікрооперацію арифметичного додавання (підсумовування) двох чисел (слів). Підсумовування  $n$ -розрядних чисел зводиться до виконання порозрядних операцій

$$\left. \begin{array}{l} S_i = X_i + Y_i + Z_i \\ P_i = 0 \end{array} \right\} \text{ при } X_i + Y_i + Z_i < k,$$
$$\left. \begin{array}{l} S_i = X_i + Y_i + Z_i \\ P_i = 1 \end{array} \right\} \text{ при } X_i + Y_i + Z_i \geq k.$$

де  $S_i$  – значення суми в  $i$ -м розряді;  $Z_i$  – перенос з молодшого розряду;  $P_i$  – перенос у старший розряд;  $k$  – основа системи числення;  $X_i, Y_i \in \{0, 1, \dots, k-1\}$  – порозрядні значення доданків

$$X = \sum_{i=1}^n X_i \cdot k^{i-1} \quad \text{і} \quad Y = \sum_{i=1}^n Y_i \cdot k^{i-1}.$$

У залежності від основи системи числення і прийнятого кодування  $k$ -ічних цифр розрізняють двійкові, троїчні, десяткові, двійково-десяткові та інші суматори.

За способом організації процесу додавання однорозрядних доданків розрізняють комбінаційні, нагромаджуючі та комбінаційно-нагромаджуючі суматори.

За способом обробки багаторозрядних доданків суматори розділяють на паралельні, послідовні та послідовно-паралельні.

Організація ланцюгів переносу між розрядами визначає структуру суматора з послідовним, наскрізним, груповим і одночасним (паралельним) переносом.

Схемне рішення суматора при його проектуванні залежить від використовуваної системи елементів (серії мікросхем), вимог до тривалості операції підсумовування і припустимих витрат.

*Комбінаційні однорозрядні суматори.* Однорозрядним суматором називають перемикальну схему, що за розрядним значенням  $X_i$  і  $Y_i$  доданків і за значенням переносу  $Z_i$  з молодшого розряду формує значення розрядної суми  $S_i$  і перенос у старший розряд  $P_i$ . Робота такого суматора може бути описана табл. 9.1 (при  $k=2$ ).

МДНФ функцій  $S_i$  і  $P_i$  мають вигляд

$$S_i = \bar{X}_i \cdot \bar{Y}_i \cdot Z_i \vee \bar{X}_i \cdot Y_i \cdot \bar{Z}_i \vee X_i \cdot \bar{Y}_i \cdot \bar{Z}_i \vee X_i \cdot Y_i \cdot Z_i; \quad (9.1)$$

$$P_i = X_i \cdot Y_i \vee X_i \cdot Z_i \vee Y_i \cdot Z_i, \quad (9.2)$$

а їх МКНФ –

$$S_i = (\bar{X}_i \vee \bar{Y}_i \vee Z_i) \cdot (\bar{X}_i \vee Y_i \vee \bar{Z}_i) \cdot (X_i \vee Y_i \vee Z_i); \quad (9.3)$$

$$P_i = (X_i \vee Y_i) \cdot (X_i \vee Z_i) \cdot (Y_i \vee Z_i). \quad (9.4)$$

Нормальні форми (9.1)-(9.4) функцій  $S_i$  і  $P_i$  можуть бути перетворені до операторного виду, зручного для реалізації суматора на елементах типу І-НЕ, І-АБО-НЕ і т.д. Наприклад, при дослідженні елементів І-АБО-НЕ однорозрядний комбінаційний суматор можна побудувати за виразами

$$S_i = \overline{X_i \cdot Y_i \cdot Z_i \vee X_i \cdot \bar{P}_i \vee Y_i \cdot \bar{P}_i \vee Z_i \cdot \bar{P}_i},$$

$$P_i = \overline{X_i \cdot Y_i \vee X_i \cdot Z_i \vee Y_i \cdot Z_i}.$$

Схема такого суматора зображена на рис. 9.1.

*Напівсуматором* називають комбінаційну схему, що реалізує функції  $q^*$  суми по mod  $k$  і переносу  $C^*$  при додаванні двох змінних, тобто ( $k=2$ ):

$$q = x \cdot \bar{y} \vee \bar{x} \cdot y = (x \vee y) \cdot (\bar{x} \vee \bar{y}) = (x \vee y) \cdot \bar{c}, \quad c = x \cdot y.$$

Повний однорозрядний суматор може бути побудований із двох напівсуматорів у відповідності зі схемою, зображеною на рис. 9.2.

Таблиця 9.1

Таблиця істинності комбінаційного суматора

$X_i$	$Y_i$	$Z_i$	$S_i$	$P_i$
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

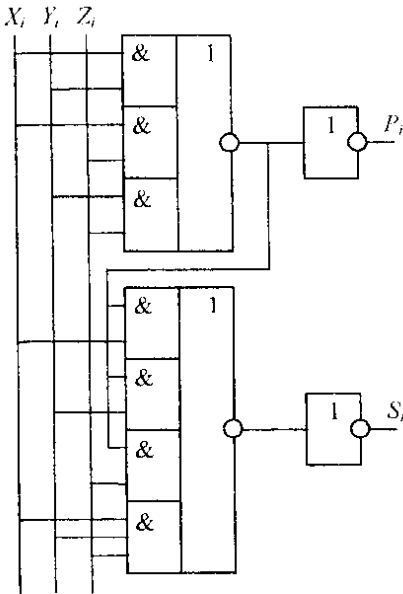


Рис. 9.1. Функціональна схема комбінаційного суматора

десятковий розряд  $P_{i+3}$ . Ці функції залежать від десятичних цифр ( $X_i, X_{i+1}, X_{i+2}, X_{i+3}$ ) і ( $Y_i, Y_{i+1}, Y_{i+2}, Y_{i+3}$ ) і переноси з молодшої тетради  $Z_i$ . Нормальні форми функцій, реалізованих ДДС, дуже громіздкі і важко мінімізуються. Тому додавання двійково-десятичних кодів (ДДК) виконують у відповідності зі схемою на рис. 9.3.

Тут на першому етапі складають ДДК десятичних цифр за правилами двійкової арифметики. Потім на другому етапі за допомогою суматорів роблять корекцію отриманого результату шляхом додавання або вирахування деякого виправлення, величина якої залежить від некорегованого результату і виду ДДК, а також виділяють десятичний перенос у старшу тетраду. Використовуваний ДДК повинний мати властивості адитивності, тобто ДДК суми десятичних цифр повинний дорівнювати сумі ДДК доданків. Такою властивістю володіють, наприклад, ДДК 8, 4, 2, 1 і 8, 4, 2,  $1+D$ , де  $D$  ціле число, назване надлишком. Якщо ж використовуваний ДДК не має властивості адитивності, то цифри доданків потрібно перед підсумовуванням перетворити в адитивний ДДК.

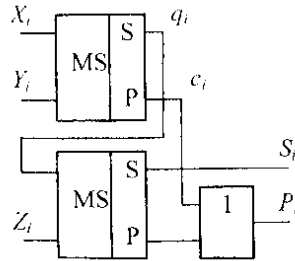


Рис. 9.2. Функціональна схема однорозрядного суматора з двох напівсуматорів

Комбінаційні двійково-десятикові суматори. При побудові двійково-десятикових суматорів (ДДС) десятикові цифри доданків звичайно кодують за допомогою чотирирозрядних чисел – тетрад.

У цьому випадку ДДС повинний реалізувати п'ять перемикальних функцій; чотири з них відповідають двійково-кодованій десятичній сумі  $S_i, S_{i+1}, S_{i+2}, S_{i+3}$  і одна – переносів в старший

Структура схеми корекції і виділення переносу може бути визначена шляхом порівняння слова  $P_{i+3} S_{i+3} S_{i+2} S_{i+1} S_i$ , отриманого при підсумовуванні цифр доданків, і необхідного результату  $P'_{i+3} S'_{i+3} S'_{i+2} S'_{i+1} S'_i$  на виходах схеми корекції (див. рис. 9.3). Нехай, наприклад, у якості ДДК використовується код 8, 4, 2, 1. Тоді стани виходів за схемою рис. 9.3 можна описати за допомогою табл. 9.2.

З таблиці 9.2 видно, що в залежності від суми, отриманої на першому етапі, корекція результату для ДДК 8, 4, 2, 1 складається в додатку 0 або 6. Підраховуючи функції  $S''_{i+3} S''_{i+2} S''_{i+1} S''_i$  і  $P''_{i+3}$  не цілком визначеними функціями аргументів  $S'_{i+3} S'_{i+2} S'_{i+1} S'_i$  і  $P'_{i+3}$ , неважко переко-

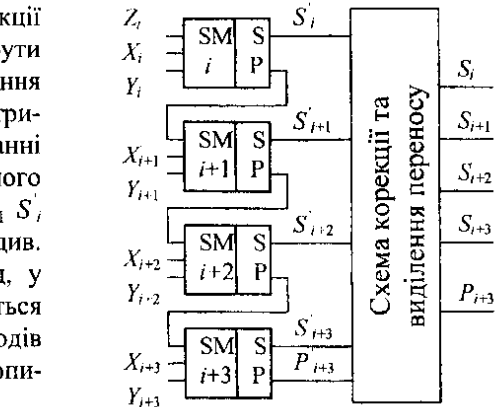


Рис. 9.3. Структурна схема комбінаційного двійково-десятькового суматора

натися, що

$$S''_{i+3} = S''_i = 0;$$

$$S''_{i+2} = S''_{i+1} = P_{i+3} = P'_{i+3} \vee S'_{i+3} \cdot S'_{i+1} \vee S'_{i+3} \cdot S'_{i+2}.$$

*Нагромаджуючі однорозрядні суматори.* Нагромаджуючим однорозрядним суматором називають схему з пам'яттю, що здійснює арифметичне додавання цифр доданків  $X_i$ ,  $Y_i$  і переносу  $Z_i$  при передачі їх на суматор послідовно в часі, незалежно від того, яким кодом (послідовним або паралельним) представлені слова, що додаються,  $X$  і  $Y$ , а також запам'ятовуючий результат додавання. Такі суматори будують на основі тригерів Т-типу.

Одна зі схем однорозрядного нагромаджуючого суматора, реалізованого на потенційних елементах, показана на рис. 9.5. Тут на тригері J спочатку формується сума  $Q_i = X_i Y_i \vee X_i Y_i$ , а потім  $S_i = Q_i Z_i \vee Q_i Z_i$ . Перенос при додаванні  $X_i$  і  $Y_i$  формується елементом 4 і запам'ятовується тригером 2 до подачі зовнішнього сигналу дозволу переносу ДП, синхронного із сигналом  $Z_i$  (відмінність між  $Z_i$  і ДП у тім, що коли ДП=1,  $Z_i$  може дорівнювати і 0, і 1). Перенос при додаванні  $Q_i$  і  $Z_i$  формується елементом 7. Час підсумовування на такому суматорі дорівнює трьом тактам. Вхід  $X_i$  може бути відсутнім. При цьому цифри доданків надходять послідовно на вхід  $Y_i$ .



Таблиця 9.2  
Таблиця істинності комбінаційного двійково-десятькового суматора

$\Sigma_n$	Сума до корекції					Сума після корекції					Корекція			
	$P'_{i+3}$	$S'_{i+3}$	$S'_{i+2}$	$S'_{i+1}$	$S'_i$	$P_{i+3}$	$S_{i+3}$	$S_{i+2}$	$S_{i+1}$	$S_i$	$S''_{i+3}$	$S''_{i+2}$	$S''_{i+1}$	$S''_i$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
2	0	0	0	1	0	0	0	0	1	0	0	0	0	0
3	0	0	0	1	1	0	0	0	1	1	0	0	0	0
4	0	0	1	0	0	0	0	1	0	0	0	0	0	0
5	0	0	1	0	1	0	0	1	0	1	0	0	0	0
6	0	0	1	1	0	0	0	1	1	0	0	0	0	0
7	0	0	1	1	1	0	0	1	1	1	0	0	0	0
8	0	1	0	0	0	0	1	0	0	0	0	0	0	0
9	0	1	0	0	1	0	1	0	0	1	0	0	0	0
10	0	1	0	1	0	1	0	0	0	0	0	1	1	0
11	0	1	0	1	1	1	0	0	0	1	0	1	1	0
12	0	1	1	0	0	1	0	0	1	0	0	1	1	0
13	0	1	1	0	1	1	0	0	1	1	0	1	1	0
14	0	1	1	1	0	1	0	1	0	0	0	1	1	0
15	0	1	1	1	1	1	0	1	0	1	0	1	1	0
16	1	0	0	0	0	1	0	1	1	0	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1	0	1	1	0
18	1	0	0	1	0	1	1	0	0	0	0	1	1	0
19	1	0	0	1	1	1	1	0	0	1	0	1	1	0

Схема виділення переносу і корекції показана на рис. 9.4.

У комбінаційно-нагромаджуючому суматорі (рис. 9.6) сигнал переносу  $P_i$  формується по МДНФ цієї функції, а сума  $S_i$  – на тригері Т-типу. На вхід цього тригера подають сигнал, що дорівнює сумі по mod 2  $Y_i$  і  $Z_i$ , і сформований комбінаційною схемою. Час підсумовування тут складає два такти.



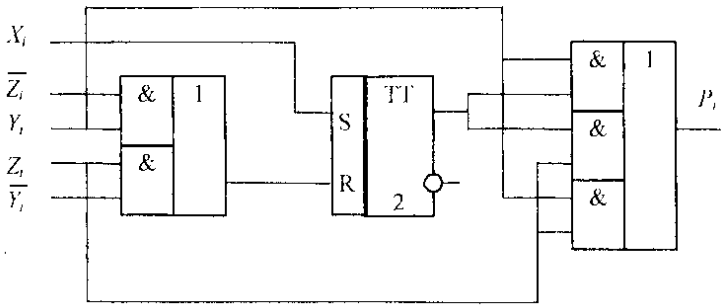


Рис. 9.6. Функціональна схема комбінаційно-нагромаджуючого суматора

*Виявлення несправностей у суматорах.* Для перевірки правильності функціонування суматора і пошуку несправностей у ньому служать тест-програми. Тест-програма передбачає виконання деякого набору операцій із заздалегідь відомими результатами. Порівняння цих результатів з одержуваними при виконанні тест-програми дозволяє локалізувати несправність з точністю до одного розряду. Найпростіша тест-програма приведена у табл. 9.3.

Розбіжність цифр отриманого і еталонного результатів у  $i$ -м розряді (рахуючи від молодших розрядів до старших) свідчить про несправності в ланцюгах формування сигналів  $S_i$  або про несправності в ланцюгах формування сигналів (тобто  $P_{i-1}$ ). Далі перевірки повинні бути піддані  $i$ -й і  $i-1$ -й розряди (по табл. 9.1).

Таблиця 9.3

Таблиця тест-програми

№ тестового набору	$X$	$Y$	$Z=X+Y$	$P_n$	$\overline{P_n}$
1	000...000	000...000	000...000	0	1
2	111...111	000...000	111...111	0	1
3	000...000	111...111	111...111	0	1
4	101...010	101...010	010...100	1	0
5	010...101	010...101	101...010	0	1
6	111...111	111...111	111...110	1	0
7	111...111	000...001	000...000	1	0
8	000...001	111...111	000...000	1	0

## Підготовка до роботи

1. Побудувати операторні форми функцій, реалізованих однорозрядним повним суматором на елементах І-НЕ, АБО-НЕ, І-АБО-НЕ, АБО-І-НЕ. Відповідні схеми замалювати в протоколі, а також оцінити швидкодію і складність (у числі елементів і по Квайну) отриманих схем.

2. Побудувати схему одного розряду двійково-десятькового суматора і цифрову діаграму станів його виходів (за аналогією з табл. 9.2). Використовуваний ДДК визначається номером варіанта і зазначений у табл. 9.4. (Якщо ДДК не має властивості адитивності, необхідно побудувати схеми перетворювачів заданого ДДК в адитивний код). Для визначення номера варіанта необхідно одержати п'ять молодших розрядів номера залікової книжки в двійковій системі числення і вважати цілим числом (від 0 до 31).

3. Побудувати схеми однорозрядних нагромаджуючого і комбінаційно-нагромаджуючого суматорів. Побудувати часові діаграми роботи цих суматорів при різних значеннях цифр доданків.

4. Знайти операторні форми функцій, реалізованих чотирирозрядними комбінаційними суматорами з наскрізним і паралельним переносами (на елементах І-НЕ). Оцінити швидкодію і складність цих суматорів. Схеми суматорів замалювати в протоколі.

Таблиця 9.4

Таблиця варіантів завдання

Номер варіанта	0,12,24	1,13,25	2,14,26	3,15,27	4,16,28	5,17,29
ДДК	7,4,2,1	6,4,2,1	5,4,2,1	4,4,2,1	4,3,2,1	2,4,2,1

Номер варіанта	6,18,30	7,19,31	8,20	9,21	10,22	11,23
ДДК	8,4,2,1 +1	8,4,2,1 +2	8,4,2,1 +3	8,4,2,1 +4	8,4,2,1 +5	8,4,2,1 +6

## Порядок виконання роботи

1. Побудувати і налагодити схеми, отриманим при виконанні п.1 теоретичного завдання.

2. Побудувати і налагодити заданий варіант двійково-десятькового суматора по схемах, отриманих при виконанні п.2.

3. Побудувати і налагодити однорозрядний нагромаджуючий і комбінаційно-нагромаджуючий суматори по схемах, отриманих при виконанні п.3.

4. Побудувати і налагодити чотирирозрядні суматори з наскрізним і одночасним переносами по схемах, отриманих при виконанні пункту 4.

5. Використовуючи тест-програму (табл. 9.3) і табл. 9.1, відшукати несправність у чотирирозрядному суматорі з паралельним переносом (несправність вводиться викладачем) з точністю до елемента.

### Зміст звіту

Звіт повинний містити короткі теоретичні відомості, необхідні для виконання завдання і відповідей на контрольні питання, усі схеми, формули, таблиці і діаграми, отримані при підготовці і виконанні лабораторної роботи, а також висновки по роботі.

### Контрольні питання

1. За якими ознаками класифікують суматори ?
2. Чому дорівнює мінімальна складність (у числі елементів і по Квайну) однорозрядного комбінаційного суматора і напівсуматора?
3. В скільки разів суматор з одночасним переносом є більш швидкодіючим у порівнянні із суматорами з наскрізним і послідовним переносами?
4. Яким вимогам повинні задовольняти ДДК, використовувані в ДДС?
5. У чому сутність властивості адитивності ДДК і до чого може привести відсутність такої властивості в ДДК ?
6. Приведіть приклади ДДК, що володіють і не володіють властивістю адитивності.
7. Чому при організації наскрізного, одночасного і частково-групового переносу однорозрядні суматори доцільно використовувати за схемою з двох напівсуматорів?
8. В скільки разів число заборонених вхідних наборів для ДДК більше числа дозволених наборів ?
9. Скільки розрядів повинний мати нагромаджуючий суматор, якщо на ньому послідовно підсумовують  $M$  чисел довжиною в  $n$  розрядів ?
10. Чи можна змінювати місцями позначення входів (тобто  $X_i$  і  $Y_i$ ,  $X_i$  і  $Z_i$  і т.д.) у комбінаційному суматорі? У нагромаджуючому? У комбінаційно-нагромаджуючому?

### Література

1. Алексенко А.Г. Основы микроэлектроники. – М.; Сов. радио, 1977. – 405 с.
2. Букреев И.Н., Мансуров В.М., Горячев В.И. Микроэлектронные схемы цифровых устройств. – М.; Сов. радио, 1975. – 368 с.
3. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум. – К.: Вища школа., 1990. – 215 с.

## 10. Лабораторна робота №9

### ПРОЕКТУВАННЯ І ДОСЛІДЖЕННЯ КЕРУЮЧИХ АВТОМАТІВ

*Ціль роботи* – вивчити методи структурного синтезу керуючих автоматів із твердою логікою, одержати навички в їх налагодженні та експериментальному дослідженні.

#### Теоретичні відомості

Послідовність мікрооперацій, виконання яких призводить до виконання операції, називають мікроалгоритмом даної операції.

Мікроалгоритм може бути реалізований за допомогою операційної схеми і керуючого автомата, узагальнена структура якого показана на рис. 10.1. В даному випадку автомат розглядається як декомпозиція елементарних автоматів (тригерів).

Автомат містить комбінаційну схему (КС) і пам'ять (П), що складається з тригерів  $T_i$ . Входами КС є виходи  $Q_1, \dots, Q_m$  тригерів і вхідні сигнали (логічні умови)  $x_1, \dots, x_k$ , що формуються в операційному пристрої. КС виробляє керуючі сигнали  $y_1, \dots, y_p$  для операційного пристрою і функції збудження тригерів  $q_1, \dots, q_m$ , що ви-

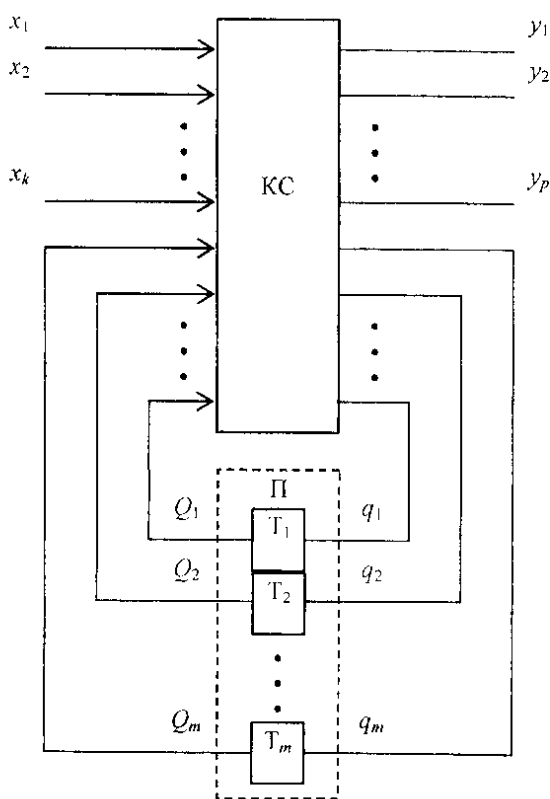


Рис. 10.1. Структурна схема автомата

значають перехід автомата з одного стану в інший. Кожному з множини станів  $\{a_1, \dots, a_m\}$  відповідає визначений набір значень  $Q$ .

Якщо вихідні сигнали залежать тільки від стану, в якому знаходиться автомат, його називають автоматом Мура. Закон функціонування такого автомата визначається виразами

$$\begin{aligned} a^{S+1} &= \delta(a^S, x^S), \\ y^{S+1} &= \lambda(a^S), \end{aligned}$$

де

$s=0, 1, 2, \dots$  – моменти автоматного (дискретного) часу;

$\delta$  – функція переходів;

$\lambda$  – функція виходів;

$a \in \{a_1, a_2, \dots, a_m\}$  – стан автомата;

$x = \{x_1, x_2, \dots, x_k\}$  – вектор значень вхідних сигналів;

$y = \{y_1, y_2, \dots, y_p\}$  – вектор вихідних сигналів автомата.

Автомат, вихідні сигнали якого залежать як від стану, так і від вхідних сигналів, називають автоматом Мілі. Його функціонування визначається виразами

$$\begin{aligned} a^{S+1} &= \delta(a^S, x^S), \\ y^{S+1} &= \lambda(a^S, x^S). \end{aligned}$$

Вихідними даними для синтезу автомата є схема операційного пристрою і змістовний мікроалгоритм операції, що включає опис мікрооперацій і логічних умов. Побудова схеми і розробка мікроалгоритму є взаємозалежними процесами.

Синтез автомата включає наступні етапи:

- 1) складання списку керуючих сигналів, що забезпечують виконання кожної мікрооперації;
- 2) визначення тривалості кожного керуючого сигналу (в числі тактів) і періоду тактуючих сигналів автомата;
- 3) одержання закодованого мікроалгоритму;
- 4) оцінка станів автомата;
- 5) складання графа автомата;
- 6) кодування станів автомата;
- 7) складання структурної таблиці автомата;
- 8) одержання МДНФ функцій збудження тригерів і керуючих сигналів;
- 9) представлення функцій збудження тригерів і керуючих сигналів в операторній формі;
- 10) побудова схеми керуючого автомата.



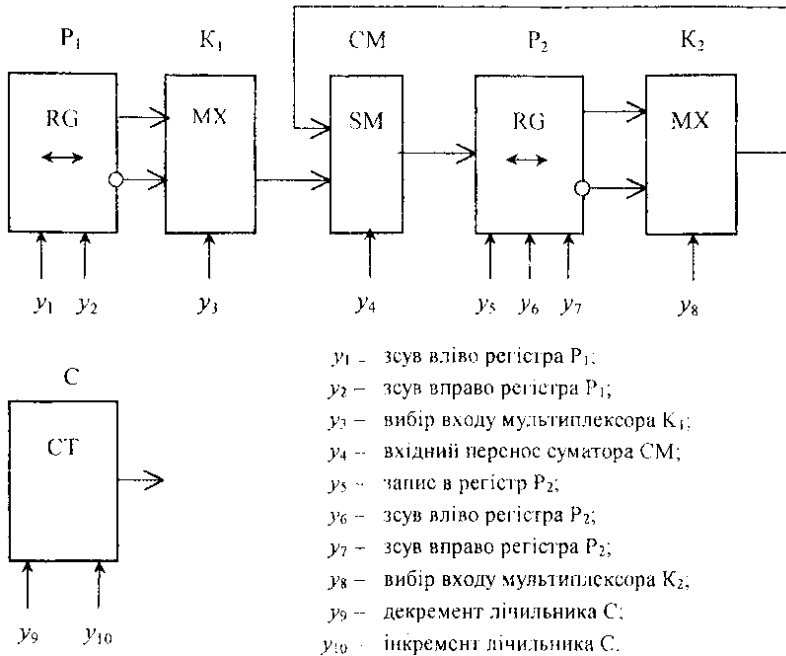


Рис. 10.2. Структурна схема операційного пристрою

Розглянемо приклад синтезу автомата Мілі для керування виконанням операції  $D=2A^2+0.5B$  на операційному пристрої (рис. 10.2).

Змістовний мікроалгоритм виконання операції показаний на рис. 10.3. У вихідному стані операнд  $B$  записаний в  $P_2$ , а операнд  $A$  – в  $P_1$  і  $C$ . В першому такті шляхом зсуву слів здійснюється подвоєння  $A$  в  $P_1$  і ділення  $B$  на 2 в  $P_2$ . Далі до вмісту  $P_2$   $A$  раз додається слово, записане в  $P_1$ . Після кожного додатка вміст  $C$  зменшується на 1. Обчислення закінчуються при виконанні умови  $C=0$ . Відповідний цьому сигнал можна одержати, наприклад, дешифруванням нульового стану  $C$ . Результат операції формується в  $P_2$ .

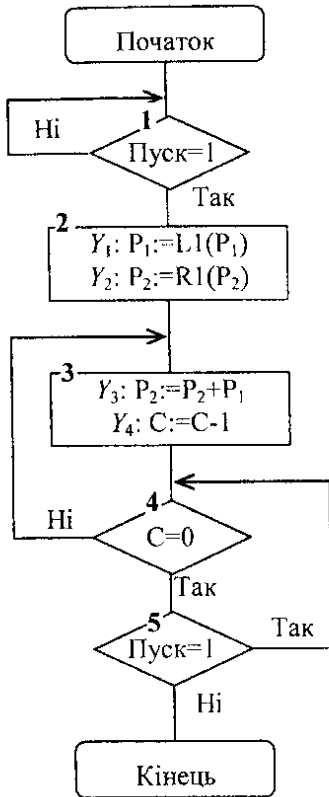


Рис. 10.3. Граф-схема змістовного мікроалгоритму виконання операції

ні  $t$ .

Асинхронний режим можна забезпечити, наприклад, введенням у мікроалгоритм додаткових операторних вершин з керуючими сигналами, тривалості яких перевищують  $t$ .

Будемо вважати, що з урахуванням швидкодії елементів для розглянутого приклада керуючі сигнали  $y_3$  і  $y_8$  повинні

Для виконання мікрооперації на регістрі необхідно подати одиничний сигнал на відповідний керуючий вхід. На всі інші керуючі входи цього регістра повинний подаватися нульовий сигнал.

Мікрооперації  $Y_1$ ,  $Y_2$ , і  $Y_4$  керуються відповідно сигналами  $y_1$ ,  $y_7$  і  $y_9$ . Для виконання мікрооперації  $Y_3$  необхідно подати три сигнали:  $y_3$ ,  $y_8$ ,  $y_5$ .

Необхідна тривалість керуючих сигналів визначається за допомогою діаграми, побудованої з урахуванням затримок в елементах операційного пристрою. Період  $t$  тактуючих сигналів звичайно вибирається або рівним максимальній тривалості керуючих сигналів, або мінімальній. При цьому величина  $t$  повинна бути не менше часу переключення автомата з одного стану в інший. У першому випадку всі мікрооперації виконуються в синхронному режимі (за однаковий проміжок часу), а в другому – в асинхронному, причому тривалості керуючих сигналів кратні величині  $t$ .

Таблиця 10.1

Таблиця тривалості керуючих сигналів

Мікрооперації	Керуючі сигнали	Тривалість керуючих сигналів
$Y_1: P_1 := L1(P_1)$	$y_1$	$t$
$Y_2: P_2 := R1(P_2)$	$y_7$	$t$
$Y_3: P_2 := P_2 + P_1$	$y_3, y_8, y_5$	$2t, 2t, t$
$Y_4: C := C - 1$	$y_9$	$t$

мати тривалість  $2t$ , а інші –  $t$ . Результати виконання перших двох етапів відбиті в таблиці 10.1.

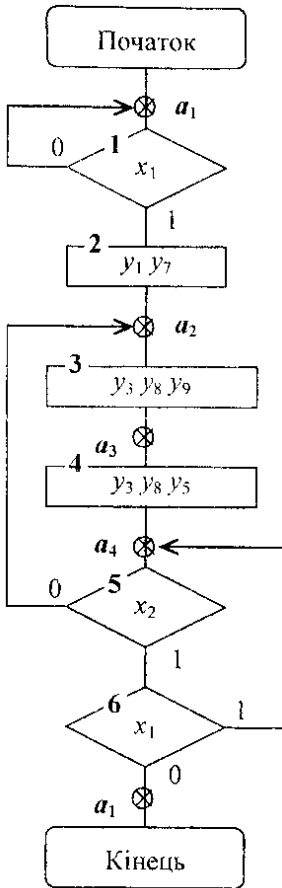


Рис. 10.4. Графа-схема за-  
кодованого мікроалгоритму  
автомата Мілі

Для одержання закодованого мікроалгоритму складаємо таблицю позначень логічних умов (табл. 10.2) і заміняємо в змістовному мікроалгоритмі (див. рис. 10.3) описи логічних умов їх позначеннями, описи мікрооперацій – відповідними керуючими сигналами (рис. 10.4). Оскільки керуючі сигнали  $y_3, y_8$ , що відповідають мікрооперації  $Y_3$ , записаній у вершині 3 на рис. 10.3, повинні мати тривалість  $2t$ , то на закодованому мікроалгоритмі (рис. 10.4) вводимо додаткову операторну вершину 4 з цими керуючими сигналами.

Таблиця 10.2

Таблиця логічних умов

Логічні умови	Позначення логічних умов
Пуск=1	$x_1$
$C=1$	$x_2$

Таблиця 10.3

Таблиця кодування станів автомата

Мілі

Стан	Код стану	
	$Q_1$	$Q_2$
$a_1$	0	0
$a_2$	0	1
$a_3$	1	1
$a_4$	1	0

Оцінка станів автомата Мілі здійснюється таким чином: символом  $a_i$  відзначається вхід вершини (логічної або операторної), наступної за початковою, а також вхід кінцевої вершини; входи всіх вершин, наступних за операторними, повинні бути відзначені різними символами. Крім станів, визначених таким чином, може виникнути необхідність введення додаткових станів. Наприклад, якщо мікрооперація

виконується по перепаду керуючого сигналу і мікроалгоритм має петлю, яка охоплює операторну вершину з таким сигналом, або кілька вершин, що слідують підряд, то для забезпечення перепадів керуючого сигналу при кожному черговому виконанні мікрооперації необхідно вводити додаткові стани. В цих станах автомат не повинний виробляти керуючі сигнали. Додаткові стани можуть знадобитися для забезпечення протигоночного кодування (див. далі).

Мікроалгоритм на рис. 10.4 відзначений чотирма різними станами ( $a_1, a_2, a_3, a_4$ ). Граф розглянутого автомата Мілі зображений на рис. 10.5.

Число вершин графа дорівнює числу станів  $a_i$  автомата. Кожному переходові автомата з одного стану в інший відповідає дуга графа. Дузі приписується набір логічних умов, при якому здійснюється перехід автомата з одного стану в інший, а також набір керуючих сигналів, що відповідають даному переходові.

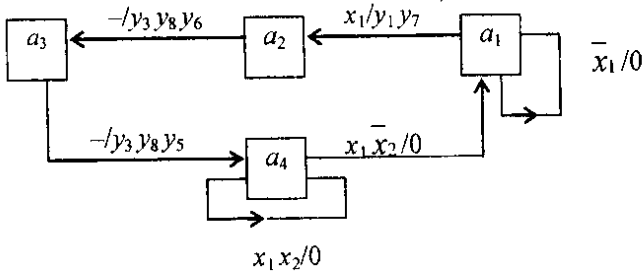


Рис. 10.5. Граф автомата Мілі

Кількість тригерів, необхідних для організації пам'яті автомата (див. рис. 10.4) визначається із співвідношення  $m > \lceil \log_2 M \rceil$ , де  $M$  – число станів автомата. Кожному станові  $a_i$  повинна відповідати одна визначена комбінація значень  $Q_1, \dots, Q_m$ . Для розглянутого приклада вибираємо коди станів відповідно до табл. 10.3. Для організації пам'яті будемо використовувати JK-тригери.

Відмітимо, що спосіб кодування впливає на правильність формування керуючих сигналів і складність автомата. Можливість формування сигналів, не передбачених графом автомата при неоптимальному кодуванні станів обумовлена появою “гонок”, що пов'язано з розкидом часу переключення окремих тригерів автомата. Наприклад, при переході автомата зі стану 10 у стан 01 під час переключення тригерів можлива поява станів 00 або 11 (в залежності від того, який із тригерів раніш переключається). Ці проміжні стани при використанні тригерів із внутрішньою затримкою не впливають

на правильність переключення автомата, однак можуть привести до появи короточасних помилкових керуючих сигналів.

Для усунення цього недоліку можна використовувати протигоночне сусіднє кодування. При сусіднім кодуванні перехід автомата з одного в будь-який інший припустимий для даного автомата стан здійснюється переключенням тільки одного тригера, внаслідок чого "гонки" не виникають. В автоматах, що не допускають сусіднього кодування, необхідно вводити додаткові стани.

Структурна таблиця автомата складається по його графу. Кожен рядок (табл. 10.4) відповідає визначеному переходу автомата з одного стану в інший. В ній записують початковий стан, стан переходу, коди цих станів, значення логічних умов, що забезпечують перехід, необхідні значення керуючих сигналів і функцій збудження тригерів. Значення функцій збудження визначаються відповідно до таблиці переходів тригера відповідного типу. В кожному рядку для  $i$ -го тригера розглядаються переходи  $Q_i^S \rightarrow Q_i^{S-1}$ . Довільні значення (0 або 1) сигналів позначаються в таблиці знаком \*.

Таблиця 10.4

Структурна таблиця автомата

ПС	Код ПС		СП	Код СП		Логічна умова		Керуючі сигнали						Функції збудження тригерів			
	$Q_1^S$	$Q_2^S$		$Q_1^{S+}$	$Q_2^{S+}$	$x_1$	$x_2$	$y_1$	$y_7$	$y_3$	$y_8$	$y_5$	$y_6$	$J_1$	$K_1$	$J_2$	$K_2$
$a_1$	0	0	$a_1$	0	0	0	*	0	0	*	*	0	0	0	*	0	*
$a_1$	0	0	$a_2$	0	1	1	*	1	1	*	*	0	0	0	*	1	*
$a_2$	0	1	$a_3$	1	1	*	*	0	0	1	1	*	1	1	*	*	0
$a_3$	1	1	$a_4$	1	0	*	*	0	0	1	1	1	*	*	0	*	1
$a_4$	1	0	$a_3$	1	1	1	1	*	0	*	*	0	0	*	0	0	*
$a_4$	1	0	$a_2$	0	1	*	0	0	0	*	*	0	0	*	1	1	*
$a_4$	1	0	$a_1$	0	0	0	1	*	0	*	*	0	*	*	1	0	*

ПС - початковий стан, СП - стан переходу.

На підставі структурної таблиці автомата визначаємо МДНФ функцій збудження і керуючих сигналів. Аргументами функцій переключення  $J, K$ , та  $Y$ , є значення  $Q_1, Q_2, X_1$  і  $X_2$ .

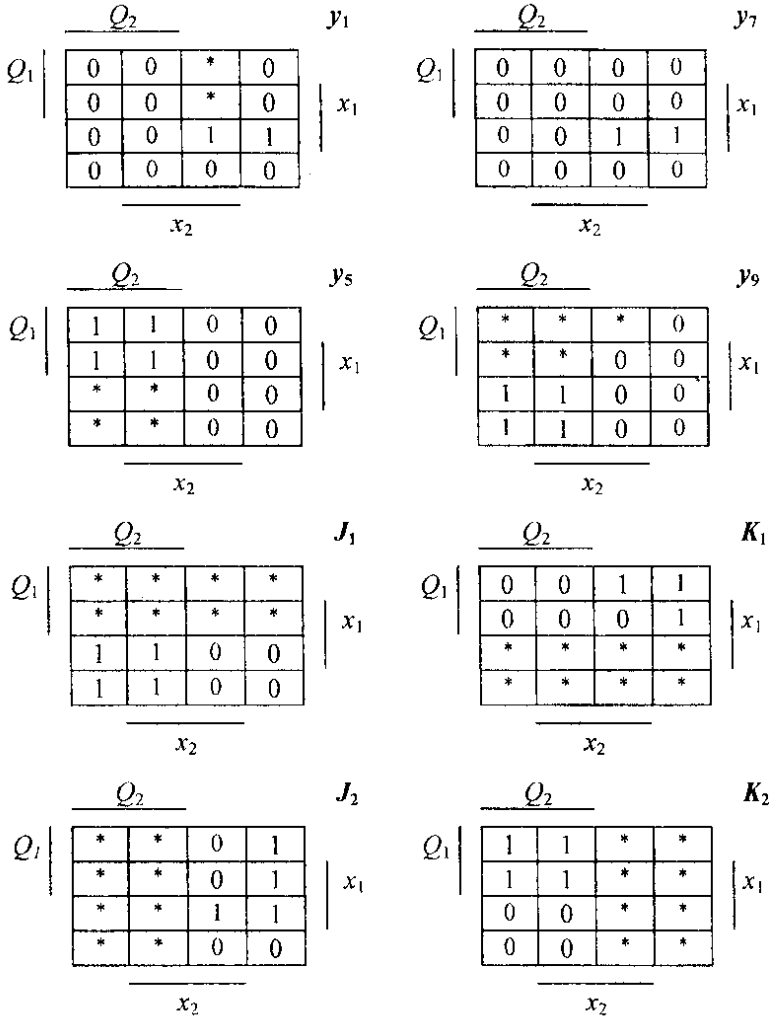


Рис. 10.6. Діаграми Вейча керуючих сигналів і функцій збудження тригерів керуючого автомата

Використовуючи діаграми Вейча (рис. 10.6), одержуємо:

$$y_1 = y_7 = \underline{Q_1} \underline{Q_2} x_1;$$

$$y_5 = y_9 = J_1 = Q_2;$$

$$K_1 = Q_2 x_2 \vee \overline{Q_2} x_1 = \overline{Q_2} (x_2 \vee x_1);$$

$$K_2 = Q_1;$$

$$J_2 = \overline{Q_1} x_1 \vee Q_1 x_2.$$

Крім того, з табл. 10.4 видно що  $Y_3 = Y_8 = 1$ .

Функціональна схема автомата зображена на рис. 10.7, де УПС – управління початкового стану, ТС – тактуючі сигнали.

Відмінність синтезу автомата Мура полягає в одержанні відзначеного мікроалгоритму і графа автомата.

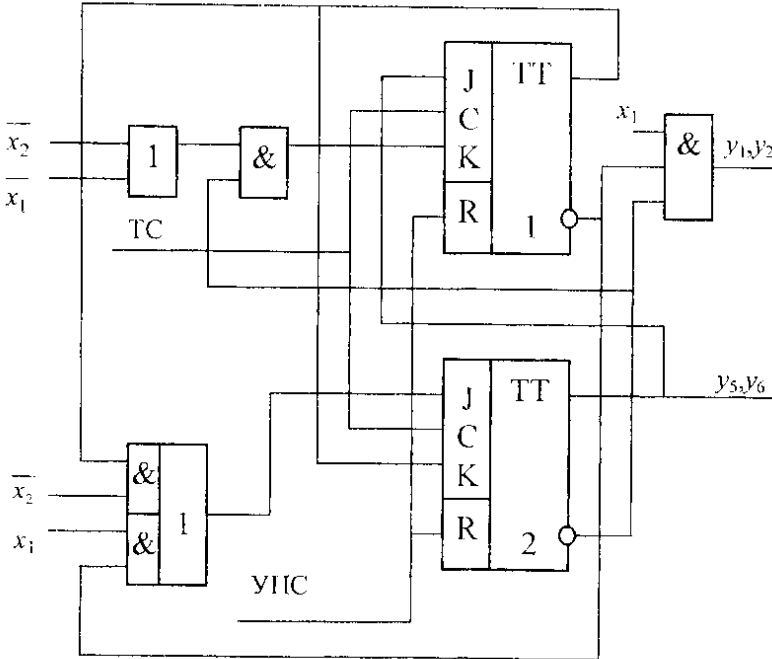


Рис. 10.7. Функціональна схема керуючого об'єкта

Оцінка станів автомата в цьому випадку здійснюється так: символом  $a_i$  відзначаються початкова і кінцева вершини; всі операторні вершини відзначаються різними символами  $a_j$ .

В даному випадку, як і при оцінці станів автомата Мілі, може виникнути необхідність введення додаткових станів.

На графі автомата Мура дугам приписують набори логічних умов, що забезпечують відповідний перехід автомата. Керуючі сигнали записують у вершинах графа, тому що вони не залежать від логічних умов.

Відзначений мікроалгоритм і граф автомата Мура для розглянутого приклада показані відповідно на рис. 10.8 і 10.9.

Кодування станів автомата Мура можна виконувати так само, як і для автомата Мілі. Однак при відповідному кодуванні керуючі сигнали можна знімати і безпосередньо з виходів тригерів автомата Мура (комбінаційна схема для формування функцій  $y_j$  не потрібна). При цьому число тригерів повинне бути не менше числа керуючих сигналів, які не повторюють один одного і не є константами.

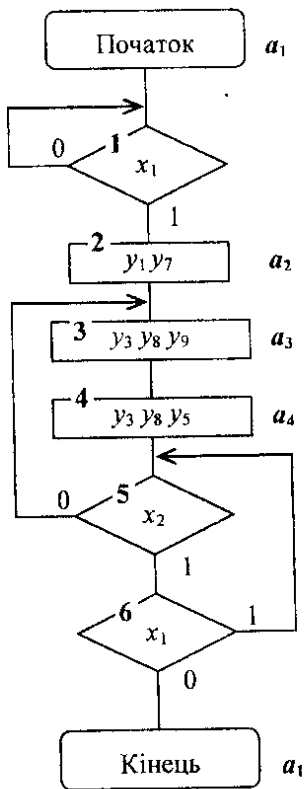


Рис. 10.8. Граф-схема закодованого мікроалгоритму автомата Мура

В процесі переключення автомата Мура в цьому випадку відповідний рівень керуючих сигналів установлюється швидше, ніж в автомата Мілі. Крім того, при будь-якому переході виключається можливість короткочасного формування помилкових керуючих сигналів. Приклад такого кодування станів автомата, що відповідає графові на рис. 10.9, приведений в табл. 10.5.

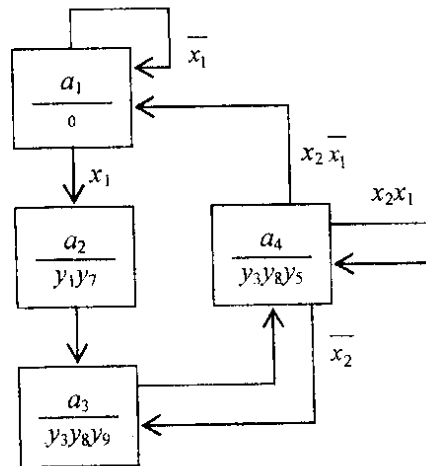


Рис. 10.9. Граф автомата Мура



В даному випадку автомат має три тригери, причому

$$y_1=Q_1, y_7=Q_1, y_5=Q_2, y_9=Q_3.$$

Для формування константних керуючих сигналів  $y_3=y_8=1$  тригери не потрібні. Співпадаючі сигнали  $y_1$  і  $y_7$  знімаються з одного тригера.

Таблиця 10.5

Таблиця кодування станів автомата Мура

Стан	Код станів		
	$Q_1$	$Q_2$	$Q_3$
$a_1$	0	0	0
$a_2$	1	0	0
$a_3$	1	0	1
$a_4$	1	1	0

### Підготовка до роботи

1. Скласти змістовний мікроалгоритм обчислення функції D на операційному пристрої (див. рис. 10.2). Вид функції D визначається по табл. 10.6 у відповідності зі значеннями  $h_5$  і  $h_4$  двох старших цифр номера бригади, записаного у вигляді п'ятизначного двійкового слова  $h_5 h_4 h_3 h_2 h_1$  (наприклад, якщо номер бригади дорівнює 12, то  $h_5=0, h_4=1, h_3=1, h_2=0, h_1=0$ ).

Вважати, що у вихідному стані операнди A, B та C записані відповідно в  $P_1, C$  та  $P_2$ .

2. Відповідно до отриманого мікроалгоритму і рис. 10.2 побудувати керуючий автомат. Тип тригерів визначається по табл. 10.6 у залежності від значень  $h_3$  і  $h_2$ , а тип автомата – за значенням  $h_1$ . Для побудови комбінаційної схеми використовувати елементи І-НЕ.

Таблиця 10.6

Таблиця варіантів завдання

$h_5$	$h_4$	Функція	$h_3$	$h_2$	Тип тригера	$h_1$	Тип автомата
0	0	$D=2C-4AB$	0	0	JK	0	Мілі
0	1	$D=A(B-1)+0,5C$	0	1	T	1	Мура
1	0	$D=2A(B+1)+0,5C$	1	0	RS		
1	1	$D=A(B+1)+2C$	1	1	D		

3. Побудувати часову діаграму роботи автомата для кожної комбінації значень логічних умов.

4. Проілюструвати прикладом обчислення результату D для одного довільного набору значень операндів A, B та C представлених трьома двійковими розрядами.

### Порядок виконання роботи

1. Викикати з відповідного каталогу схему операційного пристрою (рис. 10.2), перейменувати її і доповнити схемою автомата. Виходи автомата до входів операційного пристрою спочатку не підключати. Налагодити окремо схему автомата в синхронному режимі.

2. Підключити до керуючих входів операційного пристрою виходи автомата. Записати в  $P_1$ ,  $S$  та  $P_2$  операнди, що відповідають п.4 завдання. Зробити комплексне налагодження схеми в синхронному режимі і переконатися в правильності одержання результату  $D$  у  $P_2$ .

3. Перейти до асинхронного моделювання. Досліджувати зазначені викладачем часові параметри схеми.

### Зміст звіту

Звіт повинний включати короткі теоретичні відомості, необхідні для виконання лабораторної роботи, всі схеми, формули, таблиці і графіки, отримані при виконанні теоретичного завдання і у процесі моделювання схем, а також висновки по роботі.

### Контрольні питання

1. Що таке мікроалгоритм операції?
2. Намалюйте узагальнену структурну схему керуючого автомата.
3. Напишіть вирази, що визначають закон функціонування автоматів Мілі і Мура.
4. В чому відмінність автоматів Мілі і Мура?
5. Охарактеризуйте основні етапи проектування автомата.
6. Як перейти від змістовного мікроалгоритму до закодованого мікроалгоритму?
7. Як побудувати граф автомата?
8. Як здійснюється оцінка станів автомата?
9. Як визначити необхідну тривалість керуючих сигналів?
10. Від чого залежить кількість тригерів, необхідна для побудови автомата?
11. В чому сутність протигоночного кодування станів автомата?
12. Як скласти структурну таблицю автомата?
13. Складіть таблицю переходів для JK-, RS-, T- і D-тригерів.

14. Чи можливий перехід автомата в стан, не передбачений графом, при використанні тригерів із внутрішньою затримкою (тригерів, керованих рівнем сигналів)?
15. Коли можливе виникнення помилкових керуючих сигналів (не передбачених графом автомата) і чим визначається їх тривалість?
16. Як визначити час переходу автомата з одного стану в інший?
17. Як одержати T- і RS-тригери на основі JK-тригерів?
18. Як побудувати часову діаграму роботи автомата?

### Література

1. Букреев И.Н., Мансуров В.М., Горячев В.И. Микроэлектронные схемы цифровых устройств. – М.: Сов. радио, 1975. – 368 с.
2. Майоров С.А., Новиков Г.И. Принципы организации цифровых машин. Л.: Машиностроение, 1977. – 432 с.
3. Самофалов К.Г., Романкевич А.М., Валуйский В.Н., Каневский Ю.С., Пиневич М.М. Прикладная теория цифровых автоматов. – К.: Вища школа, 1987. – 375 с.
4. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум. – К.: Вища школа, 1990. – 215 с.

## 11. Лабораторна робота №10

### СИНТЕЗ АВТОМАТІВ З ВИКОРИСТАННЯМ ЧАСОВИХ ФУНКЦІЙ

*Ціль роботи* – вивчити метод структурного синтезу керуючих автоматів із твердою логікою з використанням апарата часових функцій, одержати навички в їх налагодженні та експериментальному дослідженні.

#### Теоретичні відомості

Керуючий автомат, що має два і більше станів, є послідовнісною схемою. Ознакою послідовної логічної схеми є наявність петель. Під петлю розуміється шлях з виходу логічного елемента на його вхід безпосередньо або через інші елементи.

Крім розглянутого в лабораторній роботі №9 методу синтезу автомата у вигляді декомпозиції тригерів (елементарних автоматів), існує метод прямого синтезу послідовних логічних схем з використанням апарата синтезу часових функцій. Застосування такого методу дозволяє побудувати керуючий автомат у будь-якому функціонально повному елементному базисі.

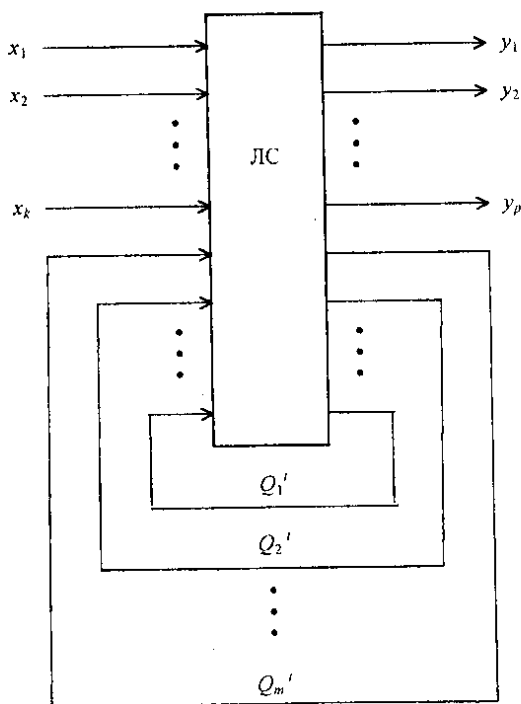


Рис. 11.1. Структура автомата

Часові перемикальні функції, на відміну від звичайних перемикальних функцій, в якості аргумента можуть використовувати власні значення, що обумовлено наявністю петлі з виходу елемента на його вхід.

Узагальнена структура автомата показана на рис. 11.1. Автомат містить логічну схему (ЛС), що складається з логічних елементів. Входами ЛС є зовнішні вхідні сигнали (логічні умови)  $x_1, \dots, x_k$ , а також деякі виходи цієї ЛС, сигнали на яких розглядаються як часові функції  $Q_1', \dots, Q_m'$  (верхній індекс визначає момент автоматного часу). ЛС виробляє зовнішні керуючі сигнали  $y_1, \dots, y_p$  для операційного пристрою (на рис. 11.1 операційний пристрій не показаний).

Стани автомата кодуються значеннями  $Q_i'$ . У зв'язку з цим справедливо співвідношення  $m \geq \lceil \log_2 s \rceil$ . У загальному виді значення  $i$ -ої часової функції можна записати як

$$Q_i'^{t+1} = f(x_1, \dots, x_k, Q_1', \dots, Q_m').$$

У процесі функціонування автомат переходить з одного стану в інший під дією зовнішніх керуючих сигналів. Кожному з множини станів  $\{a_1, \dots, a_s\}$  відповідає визначений набір значень часових функцій  $Q_i'$ .

Розглянутий метод може бути використаний для синтезу автоматів Мілі і Мура. Автомати можуть бути як синхронними, так і асинхронними. В даній лабораторній роботі розглядається синтез синхронних автоматів.

Для реалізації заданого мікроалгоритму перетворення інформації потрібен операційний пристрій, що дозволяє виконувати необхідні мікрооперації. Керуючий автомат забезпечує формування керуючих сигналів для операційного пристрою.

Синтезові автомата передують побудова операційного пристрою і розробка мікроалгоритму заданого перетворення інформації. Вихідними даними для синтезу автомата є схема операційного пристрою і змістовний мікроалгоритм операції, що включає описи мікрооперацій і логічних умов.

Синтез автомата включає наступні етапи:

1) складання для операційного пристрою списку керуючих сигналів, що забезпечують виконання кожної мікрооперації;

2) визначення тривалості керуючих сигналів (в числі тактів) і періоду тактуючих сигналів автомата;

3) одержання закодованого мікроалгоритму;

4) оцінка станів автомата;

5) складання графа автомата;

6) протигоничне кодування станів автомата значеннями  $Q_i'$ ;

7) знаходження логічних виражень для часових функцій;

8) знаходження МДНФ керуючих сигналів;

9) представлення керуючих сигналів і часових функцій в операторній формі з урахуванням заданого елементарного базису;

10) побудова і оптимізація схеми керуючого автомата.

Розглянемо приклад синтезу автомата Мура для керування виконанням операції  $D = 2AB$  на операційному пристрої (рис. 11.2). (Для розглянутого приклада набір мікрооперацій, які можна виконати на даному операційному пристрої, є надлишковим).

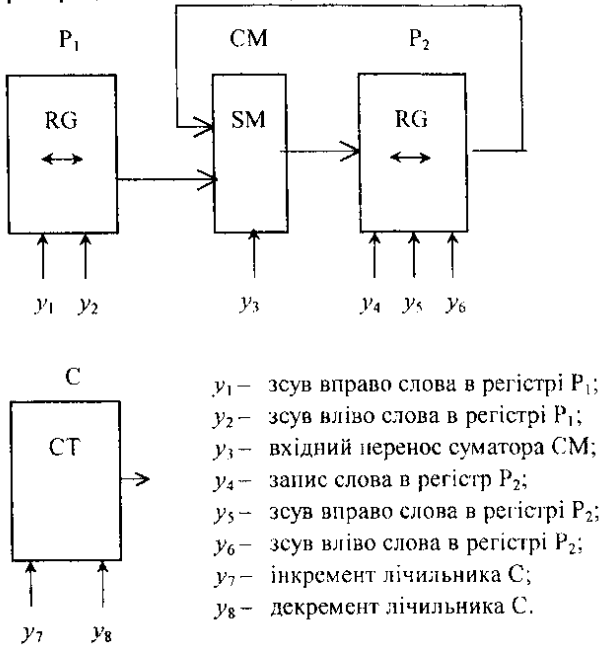


Рис. 11.2. Структурна схема операційного пристрою

Змістовний мікроалгоритм виконання операції показаний на рис. 11.3-а. У вихідному стані регістр  $P_2$  обнулений, операнд  $A$  записаний у  $P_1$ , а  $B$  - в лічильнику  $C$  (ланцюги становлення вихідного стану регістрів і лічильника на рис. 11.2 не показані).

У першому такті шляхом зсуву слова здійснюється подвоєння  $A$  в  $P_1$ . Далі до вмісту  $P_2$  додається  $B$  раз слово, записане в  $P_1$ . Після кожного додавання вміст  $C$  зменшується на 1. Обчислення закінчуються при виконанні умови  $C=0$ . Відповідний цьому сигнал можна одержати, наприклад, дешифруванням нульового стану  $C$ . Результат операції формується в  $P_2$ .

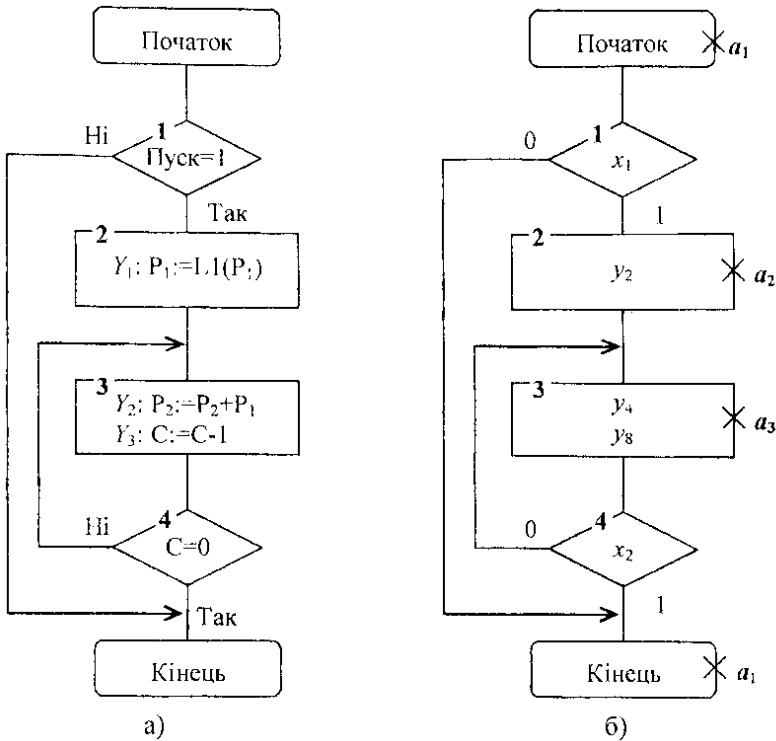


Рис. 11.3. Граф-схема змістовного (а) та закодованого (б) мікроалгоритмів виконання операції

Для виконання мікрооперації на відповідний керуючий вхід регістра (лічильника) необхідно подати одиничний сигнал. На всі інші керуючі входи цього вузла повинний подаватися нульовий сигнал. Стан сигналів на виходах регістрів і лічильника змінюється по негативному перепаді керуючого сигналу (після зняття одиничного значення сигналу).

Мікрооперації  $Y_1$ ,  $Y_2$ , і  $Y_4$  керуються відповідно сигналами  $y_2$ ,  $y_4$  і  $y_8$ . На вході  $y_3$  завжди повинний бути присутнім нульовий сигнал.

Необхідна тривалість керуючих сигналів визначається за допомогою діаграми, побудованої з урахуванням затримок в елементах операційного пристрою. Тривалість такту  $T$  звичайно вибирається або рівною максимальній тривалості керуючих сигналів, або мінімальній. При цьому величина  $T$  повинна бути більше часу переключення автомата з одного стану в інший. У першому випадку всі мікрооперації виконуються в синхронному режимі (за однаковий проміжок часу), а в другому – в асинхронному. Тривалість керуючих сигналів повинна бути кратна величині  $T$ . Асинхронний

режим можна забезпечити, наприклад, введенням у мікроалгоритм визначеного числа додаткових операторних вершин з керуючими сигналами, тривалості яких перевищують  $T$ .

Для розглянутого приклада виберемо синхронний режим виконання мікрооперацій. Результати виконання перших двох етапів представлені в табл. 11.1.

Табл. 11.1

Таблиця керуючих сигналів

Мікрооперації	Керуючі сигнали	Тривалість керуючих сигналів
$Y_1:P_1 := L1(P_1)$	$y_2$	$t$
$Y_3:P_2 := P_2+P_1$	$y_4$	$2t$
$Y_4:C := C-1$	$y_8$	$t$

При синхронному режимі керування для всіх мікрооперацій приділяється максимальний проміжок часу  $T > 2t$ .

Позначення логічних умов приведені в табл. 11.2.

Для одержання закодованого мікроалгоритму заміняємо в змістовному мікроалгоритмі описи логічних умов їх позначеннями, а описи мікрооперацій – відповідними керуючими сигналами. Після розмітки станів закодований мікроалгоритм автомата Мура приймає вид, показаний на рис.11.3-б.

На підставі мікроалгоритму будуюмо граф автомата (рис. 11.4).

Табл. 11.2

Таблиця логічних умов

Логічні умови	Позначення логічних умов
Пуск=1	$x_1$
$C=0$	$x_2$

Для розглянутого методу синтезу автоматів необхідною умовою є протигоничне (сусіднє) кодування станів автомата, з'єднаних дугами. Зв'язані дугами вершини повинні мати коди, що відрізняються значеннями тільки одного розряду. Граф на рис. 11.5 вимагає доробки, оскільки три зв'язані між собою вершини не можна забезпечити сусідніми кодами.



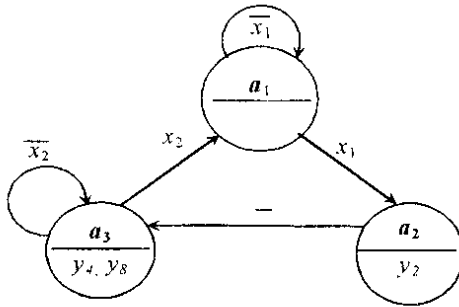


Рис. 11.4. Граф автомата

У загальному випадку для кодування станів автомата можна використовувати шаблони, показані на рис. 11.5. Граф накладається на шаблон відповідно до дуг, що з'єднують вершини. Якщо при накладенні графа на шаблон немає відповідних зв'язків, то вводяться додаткові вершини. Це дозволяє перемістити розглянуту вершину на позицію, що забезпечує необхідні зв'язки даної вершини з іншими вершинами. Необхідність введення додаткових вершин може бути також обумовлена необхідністю забезпечення перепадів керуючих сигналів.

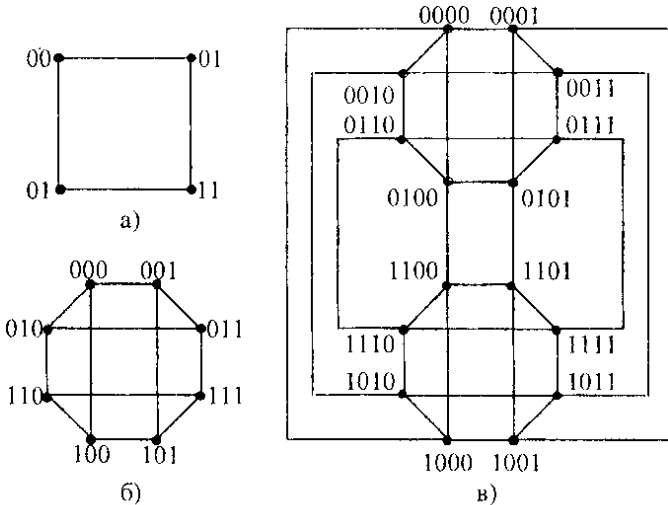


Рис. 11.5. Шаблони для кодування станів автомата: а – для 2-розрядних, б – для 3-розрядних и в – для 4-розрядних кодів станів

Скориставшись шаблоном для дворозрядних кодів станів, одержимо граф на рис. 11.6, закодований сусідніми кодами, а також забезпечує перепади для керуючих сигналів при зміні автоматом станів.

Коди станів зведені в табл. 11.3. Дуги, що замикаються на власні вершини, на графі не відображаються, тому що значення часових перемінних визначаються тільки дугами між різними вершинами графа.

Табл. 11.3

Таблиця кодування станів

Стан	Код стану	
	$Q_1'$	$Q_2'$
$a_1$	0	0
$a_2$	0	1
$a_3$	1	0
$\beta$	1	1

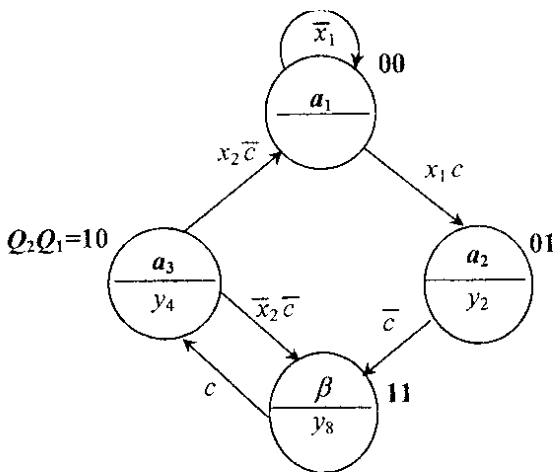


Рис. 11.6. Граф автомата із сусіднім кодуванням вершин

Для синхронного режиму роботи кожній дузі графа приписується синхросигнал  $c$  або його заперечення. Один такт відповідає напівперіодові синхросигналу. Для кожної вершини вхідні і вихідні дуги повинні бути відзначені різними значеннями синхросигналів. Наприклад, вхідні дуги

відзначені  $c$ , а вихідні -  $\bar{c}$  (або навпаки). Синхросигнал не приписують дузі, яка замикається сама на себе.

Для знаходження часових змінних використовують формулу

$$Q_i^{t+1} = F_i \vee Q_i^t \bar{G}_i, \quad (11.1)$$

або

$$Q_i^{t+1} = (F_i \vee Q_i^t) \bar{G}_i, \quad (11.2)$$

де  $F_i$  – функція встановлення часової змінної в одиничний стан;  $G_i$  – функція скидання часової змінної в нульовий стан.

При побудові схем у Булевому базисі можна використовувати кожен з приведених формул. Для елементного базису Шеффера використовується формула (11.1), а для базису Пірса – формула (11.2). У цьому випадку відповідне операторне представлення часової змінної можна одержати застосуванням правила де Моргана.

Для одержання функцій  $F_i$  записують через логічну операцію АБО всі умови переключення  $Q_i^t$  з 0 у 1. Наприклад, для змінної  $Q_1^t$  на графі рис. 11.6 маються два переходи, на яких змінна переключається з 0 у 1 ( $a_1 \rightarrow a_2$  та  $a_3 \rightarrow \beta$ ). Для функцій  $G_i$  записують умови переключення змінної з 1 у 0. Мінімізація функцій  $F_i$  і  $G_i$  виконується незалежно одне від одного. Якщо для кодування станів автомата використовуються не всі передбачені шаблоном коди, то невикористовувані коди можна розглядати як набори, на яких часові функції не визначені.

Часові функції можна визначити безпосередньо з графа автомата або для цього можна спочатку побудувати структурну таблицю автомата за формою табл. 9.4, в яку замість функцій збудження тригерів треба внести значення часових функцій.

Для графа на рис. 11.6 одержуємо систему функцій

$$\begin{cases} F_1 = a_1 x_1 c \vee a_3 x_2 \bar{c} = \bar{Q}_2 \bar{Q}_1 x_1 c \vee Q_2 \bar{Q}_1 x_2 \bar{c}; \\ G_1 = \beta c = Q_2 Q_1 c; \\ F_2 = a_2 \bar{c} = \bar{Q}_2 Q_1 \bar{c}; \\ G_2 = a_3 x_2 \bar{c} = Q_2 \bar{Q}_1 x_2 \bar{c}. \end{cases} \quad (11.3)$$

Нехай необхідно реалізувати часові функції на елементах I-III, а вихідні сигнали на елементах I. Відповідно до (11.1) і системи (11.3), застосовуючи правило де Моргана, одержимо операторну форму часових функцій

$$Q_1^{t+1} = \overline{\overline{Q_2 Q_1 x_1 c} \cdot \overline{Q_2 Q_1 x_2 c} \cdot \overline{R} \cdot \overline{Q_1^t R} \cdot \overline{Q_2 Q_1 c}}, \quad (11.4)$$

$$Q_2^{t+1} = \overline{\overline{Q_2 Q_1 c R} \cdot \overline{Q_2^t R} \cdot \overline{Q_2 Q_1 x_2 c}}. \quad (11.5)$$

Тут  $R$  – сигнал встановлення часових функцій у нульовий стан.

Вихідні сигнали в автоматі Мура цілком визначаються кодом стану.

Для графа на рис. 11.6 одержимо:  $y_2 = \overline{Q_2}Q_1$ ,  $y_4 = Q_2\overline{Q_1}$ ,  $y_8 = Q_2Q_1$ .

Схема, отримана по операторних формах (11.4) і (11.5), показана на рис. 1.1. У такому вигляді схема відображається на екрані дисплея при використанні моделюючої програми ПРОГМОЛС 2.0. Усі сигнали позначені заголовними буквами (наприклад,  $x_2 = X2$ ). Заперечення сигналів починаються з букви  $N$  (наприклад,  $\overline{Q_1} = NQ1$ ).

При синтезі автомата Мілі часові функції знаходяться аналогічно. Відмінність полягає в розмітці станів, побудові графа автомата і знаходженні функцій виходів. Зауважимо, що у даному випадку функції виходів залежать від станів автомата та вхідних сигналів (синхросигнал при цьому у якості аргументу не враховується).

### Підготовка до роботи

1. Визначити свій варіант побудови автомата. Для цього необхідно номер залікової книжки перевести в двійкову систему числення і записати шість його молодших розрядів у вигляді слова  $h_6 h_5 h_4 h_3 h_2 h_1$ . Значення  $h_i$  підставити в табл. 11.4. Наприклад, якщо десятковий номер 19 (у двійковій системі 010011), то  $h_6=0$ ,  $h_5=1$ ,  $h_4=0$ ,  $h_3=0$ ,  $h_2=1$ ,  $h_1=1$ .

Табл. 11.4

Таблиця варіантів завдання

$h_6$	Тип автомата	$h_5$	$h_4$	Функція	$h_3$	$h_2$	$h_1$	Логічні елементи
0	Мілі	0	0	$D=C+2AB$	0	0	0	I-НЕ
1	Мура	0	1	$D=AB+0,5C$	0	0	1	I, АБО, НЕ
		1	0	$D=2A(B+1)+C$	0	1	0	АБО-НЕ
		1	1	$D=A(B+1)+0,5C$	0	1	1	I, I-НЕ
		1	0		1	0	0	АБО, I-НЕ
		1	0		1	0	1	АБО-НЕ, I-НЕ
		1	1		1	1	0	I, АБО-НЕ
		1	1		1	1	1	АБО, АБО-НЕ

2. Скласти змістовний мікроалгоритм обчислення функції  $D$  на операційному пристрої (рис. 11.2). Вважати, що у вихідному стані операнди  $A$ ,  $B$ , і  $C$  записані в  $P_1$ , лічильнику  $C$  та  $P_2$ .

3. Відповідно до отриманого мікроалгоритму реалізації заданої функції на операційному пристрої рис. 11.2 побудувати синхронний керуючий автомат відповідно до заданого варіанта (табл. 11.4).

4. Побудувати часову діаграму роботи автомата для кожної комбінації значень логічних умов, задавши значеннями затримок сигналів на використовуваних елементах ( $t_1$ ,  $t_{1-HE}$  і т.д.).

5. Проілюструвати прикладом обчислення результату  $D$  для одного довільного набору значень операндів  $A$ ,  $B$ , і  $C$ , представлених трьома двійковими розрядами.

### Порядок виконання роботи

1. Викликати з відповідного каталогу схему операційного пристрою (рис. 11.2), перейменувати її і доповнити схемою автомата. Виходи автомата до входів операційного пристрою спочатку не підключати. Налогодиги окремо схему автомата.

2. Підключити до керуючих входів операційного пристрою виходи автомата. Записати в  $P_1$ ,  $C$  і  $P_2$  операнди, що відповідають п. 5 завдання. Зробити комплексне налагодження схеми в синхронному режимі і переконатися в правильності одержання результату  $D$  у  $P_2$ .

3. Перейти до асинхронного моделювання з урахуванням затримок сигналів на елементах. Досліджувати зазначені викладачем часові параметри схеми.

### Зміст звіту

Звіт повинний включати короткі теоретичні відомості, необхідні для виконання лабораторної роботи, усі схеми, формули, таблиці і графіки, отримані при виконанні теоретичного завдання та у процесі моделювання схем, а також висновки по роботі.

### Контрольні питання

1. Що таке мікроалгоритм операції?
2. Намалюйте узагальнену структурну схему керуючого автомата.
3. Напишіть вирази, що визначають закон функціонування автоматів

Мілі і Мура.

4. У чому відмінність автоматів Мілі і Мура?
5. Охарактеризуйте основні етапи проектування автомата.

6. Як перейти від змістовного мікроалгоритму до закодованого мікроалгоритму?
7. Як побудувати граф автомата?
8. Як здійснюється оцінка станів автомата?
9. Як визначити необхідну тривалість керуючих сигналів?
10. Від чого залежить кількість часових функцій, необхідних для кодування станів автомата?
11. У чому суть протигоночного кодування станів автомата?
12. Як одержати часові функції?
13. Як одержати операторну форму часових функцій?
14. Чи можливий перехід автомата в стан, не передбачений графом?
15. Коли можливе виникнення помилкових керуючих сигналів (не передбачених графом автомата) і чим визначається їх тривалість?
16. Як визначити час переходу автомата з одного стану в інший?
17. Як побудувати часову діаграму роботи автомата?

### Література

1. Зиссос Д. Проектирование систем на микропроцессорах/ Пер. с англ. под ред. А.И.Петренко. – К.: Техніка, 1982. – 176 с.
2. Майоров С.А., Новиков Г.И. Принципы организации цифровых машин. Л.: Машиностроение, 1977. – 432 с.
3. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум. – К.: Вища школа., 1990. – 215 с.

## 12. Лабораторна робота №11

### ПРОЕКТУВАННЯ І ДОСЛІДЖЕННЯ ПРИСТРОЇВ ДЛЯ МНОЖЕННЯ ЧИСЕЛ

*Ціль роботи* – вивчити методи реалізації операції множення, одержати навички в проектуванні і налагодженні операційних і керуючих пристроїв.

#### Теоретичні відомості

При множенні чисел у прямих кодах знакові та основні розряди обробляються роздільно. Для визначення знака добутку здійснюють підсумовування по модулю 2 цифр, записаних в знакових розрядах співмножників. Будемо вважати, що множене  $Y$  і множник  $X$  – правильні двійкові дроби виду  $X=0,x_1x_2\dots x_n$ ,  $Y=0,y_1y_2\dots y_m$ , де  $x_i, y_i \in \{0,1\}$ . Тоді добуток  $Z$  модулів чисел дорівнює

$$Z = YX = Yx_12^{-1} + Yx_22^{-2} + \dots + Yx_i2^{-i} + \dots + Yx_n2^{-n}. \quad (12.1)$$

Множення  $Y$  і  $X$  може бути реалізоване шляхом виконання визначеного циклічного процесу, характер якого залежить від конкретної форми виразу (12.1). Один цикл множення складається з додавання чергового часткового добутку, що представляє собою добуток множеного на одну цифру множника, до суми часткових добутків. Розрізняють чотири способи множення.

*Перший спосіб множення*

Вираз (12.1) можна представити у вигляді

$$Z = YX = (((...(0 + Yx_n)2^{-1} + Yx_{n-1})2^{-1} + \dots + Yx_i)2^{-1} + \dots + Yx_1)2^{-1},$$

Звідси випливає, що отримані суми і часткових добутків в  $i$ -м циклі ( $i=\overline{1,n}$ ) зводиться до обчислення

$$Z_i = (Z_{i-1} + Yx_{n-i+1})2^{-1}$$

з початковими значеннями  $i=1$ ,  $Z_0=0$ , причому  $Z_n=Z=YX$ .

Множення здійснюється з молодших розрядів множника, сума часткових добутоків зсувається вправо, а множене залишається нерухомим.

*Другий спосіб множення*

Запишемо (12.1) у вигляді

$$Z = (((...(0 + Y2^{-n} x_n) + Y2^{-n+1} x_{n-1}) + \dots + Y2^{-1} x_1.$$

Очевидно, що процес множення може бути зведений до  $n$ -кратного виконання циклу

$$Z_i = Z_{i-1} + Y_i x_{n-i+1}, \quad Y_i = 2Y_{i-1},$$

з початковими значеннями  $i=1$ ,  $Y_0=Y2^{-n}$ ,  $Z_0=0$ . Множення здійснюється з молодших розрядів, множене зсувається вліво, а сума часткових добутоків залишається нерухомою.

*Третій спосіб множення*

Представимо (12.1) у вигляді

$$Z = (((...(0 + Y2^{-n} x_1)2 + Y2^{-n} x_2)2 + \dots + Y2^{-n} x_i)2 + \dots + Y2^{-n} x_n.$$

Отже, суму часткових добутоків у  $i$ -м циклі ( $i=\overline{1,n}$ ) можна одержати по формулі

$$Z_i = 2Z_{i-1} + Y2^{-n} x_i.$$

Початковими значеннями є  $i=1$ ,  $Z_0=0$ . Множення здійснюється зі старших розрядів множника, сума часткових добутоків зсувається вліво, а множене нерухоме.

*Четвертий спосіб множення*

$$Z = (((...(0 + Y2^{-1} x_1) + Y2^{-2} x_2) + \dots + Y2^{-i} x_i) + \dots + Y2^{-n} x_n.$$

Процес множення може бути зведений до  $n$ -кратного виконання циклу

$$Z_i = Z_{i-1} + Y_{i-1} x_i, \quad Y_i = Y_{i-1} 2^{-1}$$



с початковими значеннями  $i=1$ ,  $Y_0=Y_2^{-1}$ ,  $Z_0=0$ .

Множення виконується зі старших розрядів множника, сума часткових добутоків залишається нерухомою, а множене зсувається вправо.

Для формування і накопичення суми часткових добутоків можна використовувати або комбінаційний суматор (СМ) і регістр добутку, або тільки накопичувальний суматор, який у функціональному відношенні можна розглядати як композицію комбінаційного суматора і регістра. Принцип побудови пристроїв, що реалізують різні способи множення, показаний на рис. 12.1, де  $P_1$  – регістр множеного,  $P_2$  – регістр добутку,  $P_3$  – регістр множника. Цифрами зазначені номери розрядів СМ і регістрів, а стрілками показаний напрямок зсуву кодів у регістрах. До входу старшого розряду регістра  $P_2$  на рис. 12.1-а, б, г підключений вихід переносу  $P$  старшого розряду СМ. Цифри, записані в молодших розрядах  $P_1$  і  $P_2$ , при реалізації першого способу мають вагу  $2^n$ , а при реалізації інших способів –  $2^{2n}$ . Перед початком множення будь-яким способом  $P_2$  встановлюється в нульовий стан.

При множенні першим способом (див. рис. 12.1-а) в першому такті  $i$ -го циклу аналізується значення  $x_n^*$  молодшого ( $n$ -го) розряду  $P_3$ , в якому знаходиться чергова цифра  $x_{n-i}+1$  множника. Вміст  $P_1$  додається до суми часткових добутоків, що знаходяться в  $P_2$ , якщо  $x_n^*=1$ , або не додається, якщо  $x_n^*=0$ . В другому такті здійснюється правий зсув в  $P_3$  і  $P_2$ , що еквівалентно множенню їх вмісту на  $2^{-1}$ . При зсуві цифра молодшого розряду  $P_2$  записується у вивільнюваний старший розряд  $P_3$ . Після виконання  $n$  циклів молодші розряди  $2n$ -розрядного добутку будуть записані в  $P_3$ , а старші – у  $P_2$ .

Час множення, якщо не застосовуються методи прискорення операції, визначаються виразами  $t_y=n(t_1+t_3)$ , де  $t_1$  і  $t_3$  – відповідно тривалості тактів підсумовування і зсуву.

Перед початком множення другим способом (див. рис. 12.1-б)  $X$  записують в  $P_3$ , а  $Y$  – в молодші розряди  $P_1$  (тобто в  $P_1$  установлюють  $Y_0=Y_2^{-1}$ ). В кожному  $i$ -м циклі множення додаванням кодів  $P_1$  і  $P_2$  керує цифра  $x_n^*$ , а в  $P_1$  виробляється зсув вліво на один розряд, в результаті чого формується величина  $Y_i=2Y_{i-1}$ . Оскільки сума часткових добутоків в процесі множення нерухома, зсув в  $P_1$  можна сполучити в часі з підсумовуванням (як правило,  $t_1 \geq t_3$ ). В цьому випадку  $t_y=nt_1$ . Множення можна закінчувати по нульовому вмісту  $P_3$ , що також приводить до збільшення швидкодії, якщо множник не нормалізований.

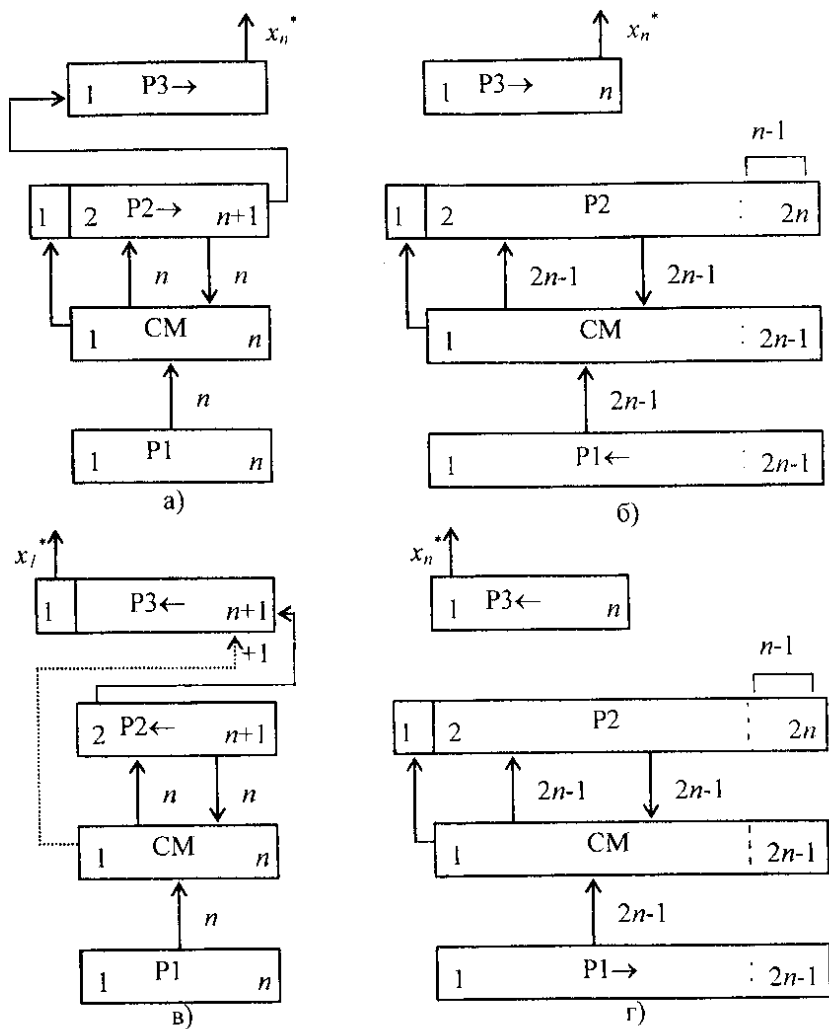


Рис. 12.1. Операційні схеми множення:

- а) 1-й спосіб множення; б) 2-й спосіб множення;  
в) 3-й спосіб множення; г) 4-й спосіб множення.

При множенні третім способом (див. рис. 12.1-в) вага молодшого розряду  $P_1$  дорівнює  $2^{-2^n}$ , тому код в  $P_1$  являє собою значення  $Y2^{-n}$ . На початку кожного циклу множення здійснюється лівий зсув в  $P_2$  і  $P_3$ , а потім

виконується додавання, яким керує  $x_1^*$ . В результаті підсумовування вмісту  $P_1$  і  $P_2$  може виникнути перенос в молодший розряд  $P_3$ , який повинний мати ланцюги для його додавання. Збільшення довжини  $P_3$  на один розряд усуває можливість поширення переносу в розряди множника. Після виконання  $n$  циклів молодші розряди добутку будуть знаходитися в  $P_2$ , а старші – в  $P_3$ . Час множення по третьому способі такий саме, як і по першому.

Перед множенням четвертим способом (див. рис. 12.1-г) множник записують в  $P_3$ , а множене – в старші розряди  $P_1$  (тобто в  $P_1$  установлюють  $Y_0 = Y2^{-1}$ ). В кожному циклі цифра  $x_1^*$ , що знаходиться в старшому розряді  $P_3$ , керує підсумовуванням, а в  $P_1$  виконується правий зсув на один розряд, що еквівалентно множенню вмісту цього регістра на  $2^{-1}$ . Час виконання множення четвертим способом складає  $t_y = nt_r$ . Як приклад на рис. 12.2 і рис. 12.3 показані мікроалгоритми множення чисел третім і четвертим способами. Для підрахунку циклів у пристроях використовуються лічильники С.

В ЦЕОМ при роботі з дробовими числами часто потрібно обчислювати не  $2n$ , а тільки  $n+1$  цифр добутку й округляти його до  $n$  цифр. В цьому випадку при реалізації другого способу можна зменшити довжину  $SM$  і  $P_2$ , а при реалізації четвертого – зменшити довжину  $SM$ ,  $P_2$  і  $P_1$ . Для того, щоб погрішність від відкидання молодших розрядів не перевищила половини ваги  $n$ -го розряду результату, в перерахованих вузлах досить мати тільки по  $l$  додаткових молодших розрядів, де  $l$  вибирається з умови

$$l \geq 1 + \log_2(n - l - 1).$$

Операція округлення здійснюється звичайно шляхом додавання одиниці до  $n+1$ -го розряду результату і відкидання всіх розрядів, розташованих правіше  $n$ -го. При цьому погрішність стає знаковмінною, а максимальне абсолютне її значення не перевищує половини ваги молодшого розряду. Додаткового такту підсумовування для округлення не потрібно. Досить записати одиницю перед початком множення в той розряд  $P_2$ , що після виконання множення залишається старшим розрядом, який відкидається.

В процесі формування суми часткових добутків код з  $P_2$  видається на  $SM$ , а з виходів  $SM$  знову записується в  $P_2$ . В зв'язку з цим при використанні потенційних елементів  $P_2$  будують на тригерах із внутрішньою затримкою. Характер керуючих сигналів і ланцюга, на який вони впливають, визначається конкретною теоретичною реалізацією вузлів і використовуваною елементною базою.

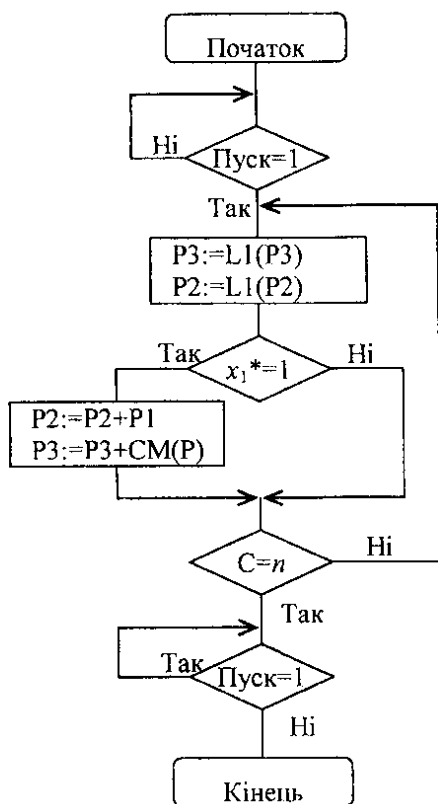


Рис. 12.2. Графа-схема  
змістовного мікроалгоритму  
операції множення 3-м  
способом

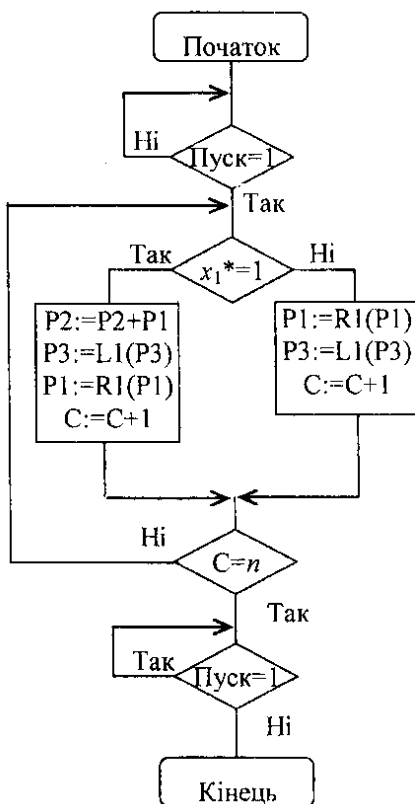


Рис. 12.3. Графа-схема  
змістовного мікроалгоритму  
операції множення 4-м  
способом

В пристроях, що реалізують другий і четвертий способи множення, можна без пересилань кодів між регістрами обчислювати вирази виду  $\sum X_i Y_i$  ( $i = \overline{1, n}$ ), для чого досить черговий результат операції залишати в  $P_2$ , який в цьому випадку повинний мати додаткові старші розряди.

В пристрої, що реалізує третій спосіб, можна без пересилань обчислювати, наприклад, функції виду  $X_i$ . Для цього  $X$  перед початком обчислення записується в  $P_1$  і в молодші розряди  $P_3$ , а потім  $i-1$  раз виконується операція множення з округленням проміжних результатів до  $n$

розрядів. Після кожної чергової операції  $P_2$  встановлюється в нульовий стан. Остаточний результат буде знаходитися в  $n$  молодших розрядах  $P_3$ . Найбільш простими є пристрої, що реалізують перший спосіб, а найбільш швидкодіючими – другий і четвертий. Однак другий спосіб не має особливих переваг у порівнянні з четвертим і, крім того, вимагає великих апаратних витрат при реалізації.

### Підготовка до роботи

1. Відповідно до завдання (табл. 12.2) розробити структурну схему і змістовний мікроалгоритм множення позитивних чисел. Для побудови схеми використовувати суматор, лічильник циклів і асинхронні регістри, що мають входи керування зсувом і занесенням інформації. На схемі повинна бути зазначена розрядність регістрів.

2. Здійснити синтез керуючого автомата (методика синтезу описана в лабораторній роботі №9). Враховувати, що мікрооперації на регістрах виконуються по негативному перепаду керуючих сигналів.

3. Побудувати часову діаграму роботи автомата для кожної комбінації значень логічних умов.

4. Виконати числовий приклад множення, вказавши стани регістрів в кожному такті. Даний приклад буде використовуватися при налагодженні пристрою.

Табл. 12.2

Таблиця варіантів завдання

$h_6$	$h_5$	$h_4$	Спосіб множення, розрядність операндів	$h_3$	$h_2$	Тип тригера	$h_1$	Тип автомата
0	0	0	1-й, 4	0	0	JK	1	Мілі
0	0	1	2-й, 4	0	1	T	0	Мура
0	1	0	3-й, 4	1	0	RS		
0	1	1	4-й, 4	1	1	D		
1	0	0	1-й, 8					
1	0	1	2-й, 3					
1	1	0	3-й, 6					
1	1	1	4-й, 3					

### Порядок виконання роботи

1. Викликати з відповідного каталогу схему операційного пристрою, перейменувати її і доповнити схемою автомата. Виходи автомата до входів

операційного пристрою спочатку не підключати. Налагодити окремо схему автомата в синхронному режимі.

2. Підключити до керуючих входів операційного пристрою виходи автомата. Зробити комплексне налагодження схеми в синхронному режимі і переконатися в правильності одержання результату.

3. Перейти до асинхронного моделювання. Досліджувати зазначені викладачем часові параметри схеми.

### Зміст звіту

Звіт повинний включати короткі теоретичні відомості, необхідні для виконання лабораторної роботи, всі схеми, формули, таблиці і графіки, отримані при виконанні теоретичного завдання і в процесі моделювання схем, а також висновки по роботі.

### Контрольні питання

1. Охарактеризуйте чотири основних методи множення чисел.
2. Як розрахувати розрядність вузлів операційного пристрою?
3. Що таке мікроалгоритм операції?
4. Намалюйте узагальнену структурну схему керуючого автомата.
5. Напишіть вирази, що визначають закон функціонування автоматів Мілі і Мура.
6. В чому відмінність автоматів Мілі і Мура?
7. Охарактеризуйте основні етапи проектування керуючого автомата.
8. Як перейти від змістовного мікроалгоритму до закодованого мікроалгоритму?
9. Як побудувати граф автомата?
10. Як здійснюється оцінка станів автомата?
11. Як визначити необхідну тривалість керуючих сигналів?
12. Від чого залежить кількість тригерів, необхідна для побудови автомата?
13. В чому суть протигоночного кодування станів автомата?
14. Як скласти структурну таблицю автомата?
15. Складіть таблицю переходів для JK-, RS-, T- і D-тригерів.
16. Чи можливий перехід автомата в стан, не передбачений графом, при використанні тригерів із внутрішньою затримкою (тригерів, керованих рівнем сигналів)?
17. Коли можливе виникнення помилкових керуючих сигналів (не передбачених графом автомата) і чим визначається їх тривалість?

18. Як визначити час переходу автомата з одного стану в інший?

### Література

1. Баранов С.И. Синтез микропрограммных автоматов.– Л.: Энергия, 1974. -- 216 с.
2. Карцев М.А. Арифметика цифровых машин. -- М.: Наука, 1971. -- 576 с.
3. Самофалов К.Г., Корнейчук В.І., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум – К.: Вища шк., 1990. -- 215 с.

## 13. Лабораторна робота №12

### ПРОЕКТУВАННЯ І ДОСЛІДЖЕННЯ ПРИСТРОЇВ ДЛЯ ДІЛЕННЯ ЧИСЕЛ

*Ціль роботи* – вивчити методи ділення чисел в прямих кодах і способи їх апаратної реалізації, придбати навички в налагодженні та дослідженні операційних і керуючих пристроїв.

#### Теоретичні відомості

Існують два основних методи ділення чисел: ділення з відновленням і без відновлення негативного залишку. Реалізація цих методів вимагає приблизно однакового обсягу устаткування, але при діленні першим методом потрібно більше часу для виконання операції. Тому метод ділення чисел без відновлення залишку є кращим.

Нехай ділене  $X$  и дільник  $Y$  є  $n$ -розрядними правильними дробами, представленими в прямому коді. В цьому випадку знакові й основні розряди операндів обробляються окремо. Знак результату визначається шляхом підсумовування по модулю 2 цифр, записаних в знакових розрядах.

Алгоритм ділення чисел без відновлення залишку зводиться до виконання наступних дій.

1. Одержати різницю  $R_0 = X - Y$ . Якщо  $R_0 \geq 0$ , то цифра  $Z_0$  частки, що має вагу 1, а при  $R_0 < 0$  – дорівнює 0. Різниця  $R_0$  є залишком.
2. Подвоїти залишок (одержати  $2R_i$ ).
3. Якщо  $2R_i < 0$ , то додати, а якщо  $2R_i \geq 0$ , то відняти  $Y$ . Якщо знову отриманий залишок  $R_{i+1} \geq 0$ , то  $Z_{i+1} = 1$ , інакше  $Z_{i+1} = 0$ .
4. Повторити пп. 2 і 3  $n-1$  раз.

Пункт 2 алгоритму можна замінити пунктом “зменшити в два рази дільник”. Наявність двох інтерпретацій другого пункту дає два основних варіанти реалізації ділення.

При реалізації ділення за першим варіантом здійснюється зсув вліво залишку при нерухомому дільнику. На рис. 13.1 показана можлива побудова ділильного пристрою. Черговий залишок формується в регістрі  $P_2$  (у вихідному стані в цьому регістрі записаний  $X$ ). Виходи  $P_2$  підключені до входів СМ безпосередньо, тобто ланцюги видачі коду з  $P_2$  не потрібні. Час для підключення  $n+1$  цифри частки визначається виразом  $t = (n+1)(t_1 + t_2)$ , де  $t_1$



– тривалість виконання мікрооперації підсумовування-вирахування;  $t_3$  – тривалість виконання мікрооперації зсуву.

При реалізації ділення відповідно до другого варіанта збільшується розрядність  $P_1$ ,  $P_2$  і  $CM$ . Рис. 13.2 пояснює принцип реалізації другого варіанта ділення. В даному випадку процеси підсумовування-вирахування і зсуву можуть бути сполучені в часі. Отже, для ділення за другим варіантом  $t=(n+1)t_i$ .

При діленні чисел з фіксованою комою повинна бути передбачена можливість фіксації переповнення розрядної сітки. Ознака переповнення виробляється, якщо в першому циклі обчислення отримане значення  $\bar{t}_0 = 1$ .

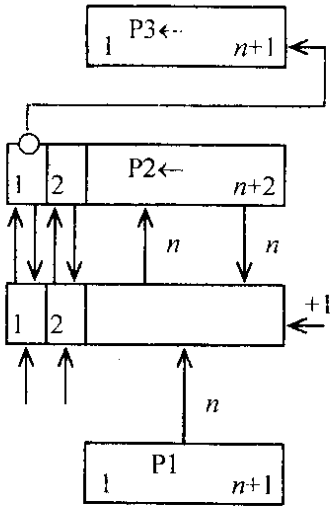


Рис. 13.1. Операційна схема пристрою ділення з відновленням залишку

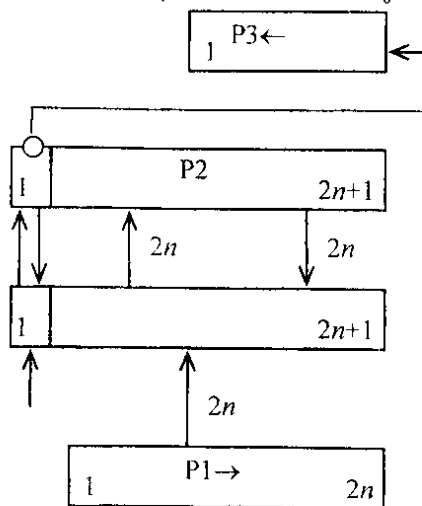


Рис. 13.2. Операційна схема пристрою ділення без відновлення залишку

### Підготовка до роботи

1. Побудувати функціональну схему операційного пристрою для реалізації варіанта ділення відповідно до табл. 13.1. Розробити змістовний мікроалгоритм розподілу.
2. Здійснити синтез керуючого автомата згідно варіанта (табл. 13.1).
3. Побудувати часову діаграму роботи автомата для кожної комбінації значень логічних умов.

4. Виконати числовий приклад ділення, вказавши стани регістрів в кожному такті. Даний приклад буде використовуватися при налагодженні пристрою.

Таблиця 13.1

Таблиця варіантів завдання

$h_5$	$h_4$	Спосіб ділення, розрядність операндів	$h_3$	$h_2$	Тип тригера	$h_1$	Тип автомата
0	0	1-й, 7	0	0	JK	1	Мілі
0	1	2-й, 6	0	1	T	0	Мура
1	0	1-й, 5	1	0	RS		
1	1	2-й, 4	1	1	D		

### Порядок виконання роботи

1. Викликати з відповідного каталогу схему операційного пристрою, перейменувати її і доповнити схемою автомата. Виходи автомата до входів операційного пристрою спочатку не підключати. Налаштувати окремо схему автомата в синхронному режимі.

2. Підключити до керуючих входів операційного пристрою виходи автомата. Зробити комплексне налагодження схеми в синхронному режимі і переконаватися в правильності одержання результату.

3. Перейти до асинхронного моделювання. Досліджувати зазначені викладачем часові параметри схеми.

### Зміст звіту

Звіт повинний включати короткі теоретичні відомості, необхідні для виконання лабораторної роботи, всі схеми, формули, таблиці і графіки, отримані при виконанні теоретичного завдання і в процесі моделювання схем, а також висновки по роботі.

### Контрольні питання

1. Описати алгоритм ділення чисел в прямих кодах.
2. Які керуючі сигнали необхідно подати на входи вузлів для виконання мікрооперацій, використовуваних в процесі ділення?
3. Вказати переваги і недоліки реалізації різних варіантів ділення.
4. Як визначити тривалість виконання мікрооперацій?

5. В яких ділительних пристроях можна сполучати мікрооперації підсумовування-вирахування і зсуву? Чому можливе сполучення цих мікрооперацій?
6. Чи можна зменшити довжину регістрів операційного пристрою при реалізації ділення чисел за другим варіантом, якщо результат повинний бути представлений  $q$  розрядами ( $q < n$ )?
7. Як виробляється округлення результату ділення? Чи потрібний додатковий такт для округлення результату?
8. В чому відмінність синтезу автоматів Мілі і Мура?
9. Охарактеризувати етапи синтезу керуючих автоматів.
10. Які тригери (з точки зору внутрішньої організації) можна використовувати для побудови кожного з регістрів ділительного пристрою?
11. Як одержати T- і RS-тригери на основі JK-тригерів?
12. Скласти таблиці переходів T-, D-, RS- і JK-тригерів.
13. В яких випадках керуючий автомат може виробляти сигнали, не передбачені його графом? Яка максимальна тривалість таких сигналів?
14. Намалювати структурні схеми ділительних пристроїв з накопичувальними суматорами. Як змінюється при цьому список виконуваних мікрооперацій?

### Література

1. Баранов С.И. Синтез микропрограмных автоматов.– Л.: Энергия, 1974. – 216 с.
2. Карцев М.А. Арифметика цифровых машин. – М.: Наука, 1971. – 576 с.
3. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум – К.: Вища шк., 1990. – 215 с.

## 14. КУРСОВА РОБОТА

Курсова робота з курсу "Прикладна теорія цифрових автоматів" виконується за індивідуальним завданням і є самостійною роботою студента. Вона призначена для розширення, закріплення, узагальнення і практичного застосування знань, умінь і навичок, отриманих студентом при вивченні курсу. У процесі виконання роботи студент повинен навчитися проектувати комбінаційні схеми, автомати з пам'яттю у заданому елементному базисі, а також навчитися користуватися довідковою літературою і вивчити процес створення проектно-конструкторської документації відповідно до діючих стандартів.

**Завдання на курсову роботу** включає наступне.

1. Виконати синтез і побудувати функціональну схему керуючого автомата по заданому алгоритму.

2. Виконати перетворення форм заданих перемикальних функцій і побудувати комбінаційні схеми на базі програмувальних логічних схем.

Варіант завдання визначається дев'ятьма молодшими розрядами номера залікової книжки студента, представленого в двійковій системі числення ( $h_9, h_8, h_7, \dots, h_1$ ).

Для одержання вихідного алгоритму керування необхідно з'єднати послідовно зверху вниз фрагменти блок-схеми алгоритму (рис. 14.1) в порядку, зазначеному в табл. 14.1. У кожному логічному вершині отриманої блок-схеми, починаючи з верхньої, переписати з табл. 14.2 в зазначеному порядку по одній логічній умові. Потім відповідно до табл. 14.3 в порядку зверху вниз і зліва направо записати в операторні вершини керуючі сигнали. Сигнали, зазначені в дужках, записуються в одну вершину. Отримана блок-схема алгоритму коректується з урахуванням подвоєної тривалості сигналу, зазначеного в табл. 14.4 (інші сигнали мають тривалість  $t$ ).

Тип тригерів і набір логічних елементів, які можна використовувати для побудови автомата, зазначені в табл. 14.5 і 14.6, а тип автомата визначений в табл. 14.7.

Система з чотирьох перемикальних функцій задана табл. 14.8.

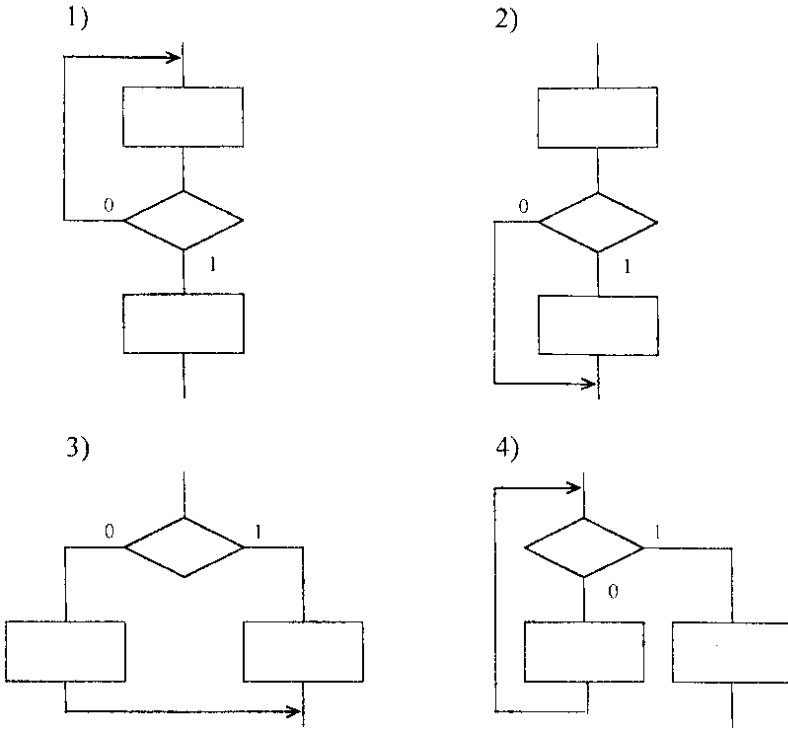


Рис. 14.1

Табл. 14.1

$h_8$	$h_4$	$h_2$	Порядок з'єднання фрагментів
0	0	0	1, 2, 3
0	0	1	1, 2, 4
0	1	0	2, 3, 4
0	1	1	2, 1, 3
1	0	0	3, 1, 2
1	0	1	3, 2, 4
1	1	0	4, 1, 2
1	1	1	4, 1, 3

Табл. 14.2

$h_8$	$h_7$	$h_3$	Логічні умови
0	0	0	X2, not X2, not X1
0	0	1	X2, not X2, X1
0	1	0	X2, X2, not X1
0	1	1	X2, X2, X1
1	0	0	not X2, not X2, X1
1	0	1	not X2, not X2, not X1
1	1	0	not X2, X2, X1
1	1	1	not X2, X2, not X1

Табл. 14.3

$h_9$	$h_4$	$h_1$	Послідовність керуючих сигналів
0	0	0	(Y1 Y2), Y3, (Y4 Y5), Y2, Y3, (Y1 Y3)
0	0	1	Y1, (Y1 Y2), Y3, (Y4 Y5), Y2, (Y1 Y3)
0	1	0	(Y1 Y2), (Y4 Y5), Y2, Y3, (Y1 Y3), Y3
0	1	1	(Y1 Y2), Y3, Y2, Y3, (Y1 Y3), (Y4 Y5)
1	0	0	(Y1 Y2), Y3, (Y4 Y5), Y3, (Y1 Y3), Y2
1	0	1	(Y1 Y2), (Y4 Y5), Y3, Y2, (Y1 Y3), Y3
1	1	0	Y3, (Y4 Y5), Y2, Y3, (Y1 Y3), (Y1 Y2)
1	1	1	Y3, (Y4 Y5), (Y1 Y2), Y2, Y3, (Y1 Y3)

Табл. 14.4

$h_6$	$h_2$	Сигнал, тривалістю $2t$
0	0	Y1
0	1	Y2
1	0	Y3
1	1	Y4

Табл. 14.5

$h_6$	$h_5$	Тригери
0	0	RS
0	1	D
1	0	JK
1	1	T

Табл. 14.6

$h_3$	$h_2$	$h_1$	Логічні елементи
0	0	0	3І-НЕ, 2І
0	0	1	3І, 4І-НЕ
0	1	0	3АБО, 4І, НЕ
0	1	1	3І, 2АБО, НЕ
1	0	0	2АБО-НЕ, 4І
1	0	1	2І-НЕ, 4АБО
1	1	0	3АБО-НЕ, 3І
1	1	1	3І-НЕ, 3АБО-НЕ

Табл. 14.7

$h_4$	Тип автомата
0	Мілі
1	Мура

Табл. 14.8

$x_4$	$x_3$	$x_2$	$x_1$	$f_1$	$f_2$	$f_3$	$f_4$
0	0	0	0	1	1	1	0
0	0	0	1	1	1	0	1
0	0	1	0	1	1	1	$h_3$
0	0	1	1	0	0	0	$h_4$
0	1	0	0	-	0	1	0
0	1	0	1	0	0	0	$h_5$
0	1	1	0	1	-	-	0
0	1	1	1	-	-	1	$h_6$
1	0	0	0	1	$h_4$	$h_7$	$h_7$
1	0	0	1	0	0	$h_8$	1
1	0	1	0	0	0	$h_9$	$h_8$
1	0	1	1	$h_1$	0	0	$h_2$
1	1	0	0	1	-	1	1
1	1	0	1	$h_2$	$h_5$	0	$h_9$
1	1	1	0	$h_3$	$h_6$	0	$h_1$
1	1	1	1	1	1	1	1

**Зміст курсової роботи.** Курсова робота повинна містити наступні документи (в порядку їх комплектування):

- титульний лист (додаток);

- опис альбому;
- сторінка з написом у середині листа "Технічне завдання".
- технічне завдання;
- сторінка з написом у середині листа "Керуючий автомат. Схема електрична функціональна";
- керуючий автомат, схема електрична функціональна;
- сторінка з написом у середині листа "Пояснювальна записка";
- пояснювальна записка.

Усі скомплектовані документи приводяться до формату А4 і скріплюються в одну папку або альбом.

Пояснювальна записка повинна містити наступні розділи.

1. Вступ.
2. Синтез автомата.
3. Синтез комбінаційних схем.
4. Висновок.
5. Список літератури.

У вступі вказується, на підставі яких документів (вихідних даних) здійснюється розробка.

В розділі 2 необхідно представити закодовану графічну схему алгоритму (ГСА), виконати розмітку станів автомата, зробити абстрактний і структурний синтез автомата.

Виконати спільну мінімізацію функцій збудження тригерів і вихідних сигналів автомата. Одержати операторні представлення функцій у заданому елементному базисі. Функціональна схема автомата представляється на окремому листі (формат А1 або А2) за правилами виконання електричних схем Е2.

В третьому розділі здійснюється синтез комбінаційних схем.

Функцію  $f_4$  необхідно представити в канонічних формах алгебр Буля, Жегалкіна, Пірса і Шеффера. Визначити належність даної функції до п'яти чудових класів. Виконати мінімізацію функції  $f_4$  методами:

- невизначених коефіцієнтів;
- Квайна (Квайна-Мак-Класкі);
- діаграм Вейча.

Виконати спільну мінімізацію функцій  $f_1, f_2, i f_3$ . Одержати операторні представлення для реалізації системи функцій на програмувальних логічних матрицях і програмувальних матрицях вентилів. В результаті синтезу повинні бути отримані мнемонічні схеми, карти програмування відповідних логічних схем, визначені мінімальні параметри логічних схем.

У висновку узагальнюються результати роботи.



**Література, що рекомендується**

1. Блейкли Т. Проектирование цифровых устройств с малыми и большими интегральными схемами. – К.: Вища школа, 1981. – 336 с.
2. Зиссос Д. Проектирование систем на микропроцессорах/ Пер. с англ. под ред. А.И.Петренко. – К.: Техніка, 1982. – 176 с.
3. Майоров С.А., Новиков Г.И. Принципы организации цифровых машин. Л.: Машиностроение, 1977. – 432 с.
4. Поспелов Д.А. Логические методы анализа и синтеза схем. – М.: Энергия, 1974. – 367 с.
5. Проектирование цифровых вычислительных машин/ Под ред. С.А. Майорова. Учебное пособие для студентов вузов. – М.: Высшая школа, 1972. – 344 с.
6. Савельев А.Я. Арифметические и логические основы цифровых автоматов. – М.: Высшая школа, 1980. – 255 с.
7. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум. – К.: Вища школа., 1990. – 215 с.
8. Самофалов К.Г., Романкевич А.М., Валуйский В.Н., Каневский Ю.С., Пиневич М.М. Прикладная теория цифровых автоматов. – К.: Вища школа, 1987. – 375 с.
9. Складов В.А. Синтез автоматов на матричных БИС/ Под ред. С.И.Баранова. – Минск: Наука и техника, 1984. – 287 с.

## 15. КУРСОВИЙ ПРОЕКТ

Курсовий проект за дисципліною "Комп'ютерна схемотехніка" виконується за індивідуальним завданням і є самостійною роботою студента. Він призначений для розширення, закріплення, узагальнення і практичного застосування знань, умінь і навичок, отриманих студентом при вивченні курсу. У процесі курсового проектування студент повинний ознайомитися з основними етапами проектування ЕОМ відповідно до технічного завдання, а також вивчити процес створення проектно-конструкторської документації відповідно до діючих стандартів.

**Завдання на курсове проектування.** Розробити проблемно орієнтовану ЕОМ з мікропрограмним керуванням, система команд якої забезпечує ефективну реалізацію заданого класу алгоритмів (наприклад, рішення різних систем рівнянь чи обчислення функцій визначеним чисельним методом і т. ін.). Виконати оцінку ефективності прийнятих технічних рішень.

До складу розроблювальної ЕОМ повинні входити процесор (П), основна пам'ять (ОП), що містить ОЗП і ПЗП, а також зовнішні пристрої (ЗП), контролери переривань і прямого доступу до пам'яті.

Дані для розробки системи команд видаються керівником індивідуально кожному студенту. Особливості структури ЕОМ визначаються табл. 15.1...15.7. (У таблицях через  $a_0$  –  $a_1$  позначені молодші розряди номера залікової книжки, представленого у двійковій системі числення).

Табл. 15.1.

$a_3$	$a_2$	$a_1$	Характеристика системи
0	0	0	TSS, індивідуальний захист комірок пам'яті
0	0	1	TSS, захист пам'яті по граничних адресах
0	1	0	MFT, захист пам'яті по масках
0	1	1	MFT, захист пам'яті по ключах
1	0	0	MVT, індивідуальний захист комірок пам'яті
1	0	1	MVT, захист пам'яті по граничних адресах
1	1	0	SVS, сторінкова організація пам'яті
1	1	1	SVS, сегментно-сторінкова організація пам'яті

Табл. 15.2.

a4	Система команд (можливий прототип)
0	Комплексна (Intel, VAX, Motorola)
1	Із даними, що визначаються самі (SWORD)

Табл. 15.3.

a10	a5	Обсяг оперативної пам'яті, Мбайт	Кількість ЗП
0	0	32	8
0	1	64	16
1	0	16	12
1	1	128	24

Табл. 15.4.

a7	Організація системної магістралі
0	З розділеними шинами адреси і даних
1	З об'єднаними шинами адреси і даних

Табл. 15.5.

a8	a6	Обов'язковий тип даних
0	0	Integer
0	1	Real
1	0	Char
1	1	Word

Табл. 15.6.

a9	a4	a10	Обов'язкова команда
0	0	0	Множення
0	0	1	Ділення
0	1	0	Обчислювання квадратного кореня
0	1	1	Перетворення 2 - 10
1	0	0	Перетворення 10 - 2
1	0	1	Додавання (real)
1	1	0	Віднімання (real)
1	1	1	Зведення в цілу ступінь

Табл. 15.7.

a1	Контролер переривань
0	Централізований
1	Децентралізований

**Зміст курсового проекту.** Курсовий проект повинний містити наступні документи (у порядку їхнього комплектування):

- титульний лист (додаток);
- технічне завдання;
- сторінка з написом у середині листа " Документи технічного проекту";
- відомість технічного проекту;
- сторінка з написом у середині листа "Графічна документація";
- схема електрична структурна;
- схема електрична функціональна;
- схема електрична принципова;
- сторінка з написом у середині листа "Пояснювальна записка ";
- пояснювальна записка і додатки до неї;
- сторінка з написом у середині листа "Робоча документація";
- специфікація виробу;
- відомість специфікацій;
- відомість покупних виробів;
- технічні умови;

- технічний опис виробу.

Усі скомплектовані документи приводяться до формату А4 і скріплюються в альбом.

**Пояснювальна записка** повинна містити наступні розділи.

Введення.

1. Розробка архітектури і формування системи команд ЕОМ.
  2. Розробка структури ЕОМ і її обґрунтування.
  3. Програма в командах розробленої ЕОМ реалізації заданого алгоритму.
  4. Функціонування ЕОМ у багатопрограмному режимі.
  5. Реалізація переривань.
  6. Реалізація обміну інформацією з зовнішніми пристроями в режимі прямого доступу до пам'яті.
  7. Розробка функціональних мікроалгоритмів реалізації заданих операцій.
  8. Розробка мікропрограм.
  9. Розробка заданого вузла ЕОМ на базі ПЛІС.
  10. Моделювання з використанням АРМ роботи вузла ЕОМ на ПЛІС з метою визначення основних характеристик.
  11. Розрахунок продуктивності і надійності заданого вузла ЕОМ.
  12. Навчальна заявка на винахід.
- Висновки.  
Список літератури.  
Додатки.  
Зміст.

У введенні вказується, на підставі яких документів здійснюється розробка ЕОМ.

У розділі 1 представляється обґрунтування вибору системи команд ЕОМ, формати команд і даних, модель програміста.

В другому розділі представляється опис структурної схеми. Указується склад і призначення пристроїв та вузлів ЕОМ, обґрунтовується вибір розрядності регістрів і шин, пояснюється організації модулів пам'яті, інтерфейсів і т. ін.

Розділ 3 повинний містити програму реалізації заданого алгоритму у командах розробленої ЕОМ. Програма розробляється в умовних адресах і представляється у вигляді послідовності мнемокодів чи кодів команд.

Четвертий розділ повинний містити опис роботи ЕОМ у багатопрограмному режимі, функцій операційної системи, взаємодії апаратних і програмних засобів у процесі функціонування ЕОМ.

У п'ятому розділі описується організація апаратних, програмних і мікропрограмних засобів реалізації переривань, алгоритми взаємодії процесора і контролера переривань.

Розділ 6 повинний містити опис апаратних, програмних і мікропрограмних засобів прямого доступу до пам'яті, алгоритми взаємодії процесора і контролера прямого доступу.

У розділі 7 повинні бути представлені функціональні мікроалгоритми (у термінах узагальнених мікрооперацій) виконання по одній команді для кожної з наступних груп команд:

- системні команди;
- команди вводу-виводу;
- команди перетворення інформації;
- команди передачі керування.

У розділі 8 представляються структурні мікроалгоритми і мікропрограми, задані керівником проекту (виконання команд, взаємодії процесора і контролерів і т. ін.).

Розділи 9 і 10 присвячені розробці і дослідженню вузла ЕОМ, побудованого на основі ПЛІС. Завдання на розробку вузла узгоджується з керівником проекту. Моделювання вузла здійснюється з використанням спеціалізованого АРМ.

У наступному розділі виконуються розрахунки.

У розділі 12 представляється навчальна заявка на винахід. Предмет передбачуваного винаходу узгоджується з керівником проекту.

**Навчальна заявка** повинна містити основні матеріали реальної заявки на винахід, передбачені діючими положеннями, а саме: довідку про патентне дослідження об'єкта, опис передбачуваного винаходу й анотацію з вказівкою індексу УДК.

Довідка про патентне дослідження складається на підставі патентного пошуку по джерелах країни, що займають ведуче положення в області обчислювальної техніки.

Опис передбачуваного винаходу повинний включати:

- область застосування і призначення об'єкта;
- опис структури і принципу дії пристроїв аналогів із вказівкою їхніх основних недоліків;
- опис пристрою-прототипу (аналога, що має найбільше число ознак, загальних з ознаками об'єкта, що заявляється,) із вказівкою їхніх недоліків;
- опис структури об'єкта, що заявляється;
- опис принципу дії;
- обґрунтування техніко-економічної ефективності;
- формулу передбачуваного винаходу;

- графічну частину (креслення і рисунки).

Якщо патентні дослідження дозволяють зробити висновок про те, що розроблений студентом пристрій має ознаки істотної новизни і корисності (дане технічне рішення може бути захищене патентом), то складається реальна заявка на винахід, що включає всю необхідну патентну документацію.

У випадку подачі реальної заявки на передбачуваний винахід, у результаті підготовки якої складається повний комплект патентної документації, обсяг розрахунково-графічного матеріалу курсового проекту може бути зменшений.

У висновках узагальнюються результати роботи.

У додатках до пояснювальної записки при необхідності приводять матеріали, що не є конструкторською документацією.

**Етапи проектування.** Проектування варто починати з аналізу алгоритму рішення заданої задачі і розробки архітектури ЕОМ. У якості індивідуального завдання, як правило, пропонується реалізація визначеного чисельного методу. При необхідності можуть задаватися додаткові дані (конкретна обчислювальна схема, форма представлення даних, погрішність обчислень і т. ін.).

Спочатку в систему команд включаються команди, що забезпечують ввід і вивід даних, роботу з підпрограмами, розгалуження програми, дозвіл і заборона переривань, а також команда, що зазначена в табл. 15.7. Потім система команд розширюється командами, що у сукупності з наявними забезпечують реалізацію заданого алгоритму. Розробляється формат команд, визначається їх мнемоніка.

Виходячи з заданої точності обчислень і інших факторів вибирається форма і розрядність представлення даних.

Складається програмістська модель процесора, уточнюється організація пам'яті і розробляються функціональні мікроалгоритми виконання команд. Визначається необхідне число регістрів загального призначення і робочих регістрів.

На підставі цього розробляється структурна схема процесора, уточнюється його архітектура.

Далі розглядаються питання обміну інформацією з зовнішніми пристроями (режим переривань, прямого доступу, організація вводу-виводу даних).

На основі аналізу розроблених мікроалгоритмів виконання команд і обраних способів взаємодії з зовнішніми пристроями (програмний, мікропрограмний, апаратний чи змішаний) уточнюється структурна схема ЕОМ.

Далі виконується індивідуальне завдання, розробляється функціональна схема заданого вузла ЕОМ і завантажується в структуру ПЛІС. На цьому

етапі використовуються спеціалізовані АРМ, обумовлені обраною серією ПЛІС.

При оцінці продуктивності підраховується кількість тактів, необхідна для виконання окремих команд, визначається тривалість тактів. Виходячи з частоти появи окремих команд і сполучення процесів визначається продуктивність ЕОМ.

Надійність компонентів ЕОМ характеризується інтенсивністю відмовлень і коефіцієнтом готовності, тому що ЕОМ є відновлюваною системою.

На заключному етапі проектування оформляються матеріали навчальної заявки на передбачуваний винахід.

### Література, що рекомендується

1. Байков В.Д., Смолів В.Б. Специализированные процессоры: интегральные алгоритмы и структуры. – М.: Радио и связь, 1985. – 288 с.
2. Балашов е.п., Пузанков Д.В. Микропроцессоры и микропроцессорные системы. – М.: Радио и связь, 1981. – 328 с.
3. Бородин В.Б., Калинин А.В. Системы на микроконтроллерах и БИС программируемой логики. – М.: Издательство ЭКОМ, 2002. – 400 с.
4. Брик Дж., Мик Дж. Проектирование микропроцессорных устройств с разрядно-модульной организацией: В 2-х частях. – М.: Мир, 1984. – 479 с.
5. Булгаков С.С. и др. Проектирование цифровых систем на комплектах микропрограммируемых БИС. – М.: Радио и связь, 1984. – 240 с.
6. Глушков В.М. и др. Программное обеспечение ЭВМ МИР-1 и МИР-2. Том 2. – К.: "Наукова думка", 1976. – 371 с.
7. Евреинов Э.В. Однородные вычислительные системы, структуры и среды. – М.: Радио и связь, 1981. – 208 с.
8. Жабин В.И. Архитектура вычислительных систем реального времени. – К.: ВЕК+, 2003. – 176 с.
9. Каган Б.М. Электронные вычислительные машины и системы. – М.: Энергоатомиздат, 1985. – 552 с.
10. Карцев М.А. Архитектура цифровых вычислительных машин. – М.: "Наука", 1978. – 295 с.
11. Карцев М.А., Брик В.А. Вычислительные системы и синхронная арифметика. – М.: Радио и связь, 1981. – 360 с.



12. Козлов Б.А., Ушаков И.А. Справочник по расчету надежности аппаратуры радиоэлектроники и автоматики. – М.: "Сов. радио", 1975. – 472 с.
13. Корнеев В.В., Киселев А.В. Современные микропроцессоры. – М.: НОЛИДЖ, 1998. – 240 с.
14. Ларионов А.М., Майоров С.А., Новиков Г.И. Вычислительные комплексы, системы и сети. – Л.: Энергоатомиздат, 1986. – 286 с.
15. Лю Ю-Чжен, Гибсон Г. Микропроцессоры семейства 8086/8088. – М. "Радио и связь", 1987. – 512 с.
16. Майерс Г. Архитектура современных ЭВМ: В 2-х частях. – М.: "Мир", 1985. – 676 с.
17. Микропроцессоры. Архитектура и проектирование микроЭВМ. Организация вычислительных процессов/ П.В.Нестеров, В.Ф.Шаньгин, В.Л.Горбунов. Под редакцией Л.Н.Преснухина. – М.: Высш. школа, 1986. – 495 с.
18. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П. Цифровые ЭВМ. Теория и проектирование. – К.: Вища школа, 1989. – 424 с.
19. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум. – К.: Вища школа, 1989. – 124 с.
20. Уокерли Дж. Архитектура и программирование микроЭВМ. – М.: "Мир", 1984. – 360 с.
21. Фролов А.В., Фролов Г.В. Защищенный режим процессоров Intel 80286/80386/80486. М.: "Диалог-МИФИ", 1993. – 234 с.

## 16. ПРОГРАМНИЙ КОМПЛЕКС МОДЕЛЮВАННЯ ЕОМ

При виконанні курсового проекту у процесі розробки архітектури, системи команд і мікропрограм може бути використаний програмний моделюючий комплекс, розроблений на кафедрі обчислювальної техніки НТУУ "КПІ".

Комплекс включає програмний емулятор ЕОМ з мікропрограмним керуванням, мікроасемблер, текстовий редактор та інші програми, об'єднані в інтегроване середовище. Комплекс дозволяє розробляти та відлагоджувати мікропрограми для будь-якої системи команд у рамках заданої структури ЕОМ (рис. 16.1), системи мікрооперацій, розрядності слів даних і адреси.

До складу ЕОМ входить процесор, основна пам'ять (ОП), пристрій вводу-виводу (ПВВ), блок пріоритетних переривань (БПП).

Процесор складається з пристрою обробки даних (ПОД), пристрою мікропрограмного керування (ПМК) і інтерфейсу системної магістралі (ІСМ). До складу ПОД входять арифметико-логічний пристрій (АЛП) і блок регістра стану (БРС). ПМК містить мультиплексор умов (МУ), схему формування адреси мікрокоманди (СФАМ), пам'ять мікрокоманд. Інтерфейс ІСМ містить регістр адреси (РАД) і буфер даних (БД).

Програмний емулятор настроєний на 16-розрядні регістри процесора. Шина адреси (ША) і шина даних (ШД) мають відповідно 20 і 16 розрядів. Мікропрограми можуть бути написані на мікроасемблері, а також в кодах мікрокоманд.

Для запису слова в ОП необхідно завантажити адресу комірки пам'яті в РАД, а потім подати сигнал R (Read). Через визначений проміжок часу на ШД встановлюється вміст комірки пам'яті, до якого здійснювалося звертання. При цьому ОП виробляє сигнал готовності RDM (ReaDy Memory). Через БД зчитане з пам'яті слово може бути записане у визначений регістр процесора. При записі інформації в ОП після завантаження адреси в РАД через БД виставляється слово, яке повинне бути записане в пам'ять, а потім видається керуючий сигнал W (Write). По завершенні запису в ОП формується сигнал RDM.

Локальна шина (ЛШ) процесора є 16-розрядною. Тому завантаження РАД здійснюється в два етапи. В старші розряди запис інформації супроводжується сигналом EWH, а в молодші - EWL.

Обмін інформацією між процесором та ВПП відбувається через регістр даних (РД), який входить до складу ВПП. Готовність ПВВ до обміну визначається за допомогою його регістру стану (РС). Розряд РС за номером 7 містить біт "Готовність". При зверненні до ВПП у програмному режимі спочатку необхідно перевірити його готовність. Регістри РД та РС мають адреси у адресному просторі процесора. Звертання до регістрів ПВВ здійснюється аналогічно звертанням до комірок ОП, але замість сигналів R, W і

RDM формуються відповідно сигнали I (Input), O (Output) і RDD (Read Device). Крім цього, ПБВ можуть формувати сигнали запиту переривань  $IRQ_i$  ( $i=0, \dots, 7$ ), якщо вони готові до обміну даними. Дозвіл на формування запиту на переривання забезпечується записом одиниці у 6-й розряд PC (розряд дозволу переривання).

БІП забезпечує аналіз пріоритетів сигналів  $IRQ_i$ , які потрапляють на входи INR, формування сигналу INT вимоги переривання і вектору переривання VEC. Вхід M дозволяє маскування сигналів переривань від ПБВ. Перетворювач вектору (ПВ) дозволяє формувати початкову адресу підпрограми обслуговування переривання. Адреса передається на ШД по сигналу EV.

Система може бути настроєна на реалізацію розподіленого контролера переривань. У цьому випадку у склад кожного ПБВ входить блок переривань, який формує сигнал  $IRQ_i$ . Всі блоки зв'язані пріоритетним ланцюжком (на рис. 16.1 не показаний). Виходи  $IRQ_i$  об'єднані через монтажну логічну функцію I. Загальний сигнал  $IRQ$  приймає активний рівень (нульовий), коли є хоча б одна вимога переривання. Цей сигнал перевіряється в ПМК. Якщо необхідно виконати переривання програми, то ПМК видає по черзі два сигнали: LOCK (блокування пріоритетного ланцюжка) та INTA (сигнал підтвердження переривання). Після цього ПБВ видає на ШД вектор переривань, який може бути прийнятим у процесор.

АЛП містить арифметико-логічний блок (АЛБ) і надоперативний запам'ятовуючий пристрій (НОЗП), до складу якого входять 16 регістрів ( $R0 \dots R15$ ) та допоміжний регістр RQ. Інформація із НОЗП може видаватися одночасно по двох каналах (A і B). Вибір регістрів здійснюється подачею адрес RA і RB на відповідні входи НОЗП. Адреси можуть видаватися з регістрів RA і RB або безпосередньо з ПМК, міняючи зазначені регістри (відповідні зв'язки на рис. 16.1 умовно не показані). Запис адрес у RA і RB відбувається відповідно сигналами EWA і EWB.

АЛБ виконує мікрооперації, зазначені в табл. 16.1, де через R і S позначені операнди, а через C1 - вхідний перенос у молодший розряд. В якості операндів R і S може використовуватися інформація на локальній шині (BUS\_D), а також з регістрів НОЗП по каналах A і B.

Результат мікрооперації Y через блок зсуву зсувач (БЗ) може бути записаний в регістр НОЗП за адресою RB. БЗ забезпечує передачу результату без зсуву, а також із зсувом на один розряд вліво або вправо. Якщо зсув не виконується, то результат може бути записаним у RQ. Крім того, через буфер Y (BY), що керується сигналом OEY, результат мікрооперації може бути виданий на локальну шину.

При виконанні мікрооперацій в АЛБ формуються ознаки:

CO - перенос зі старшого розряду;

ZO - нульовий результат;

NO - негативний результат;  
 VO - переповнення розрядної сітки.

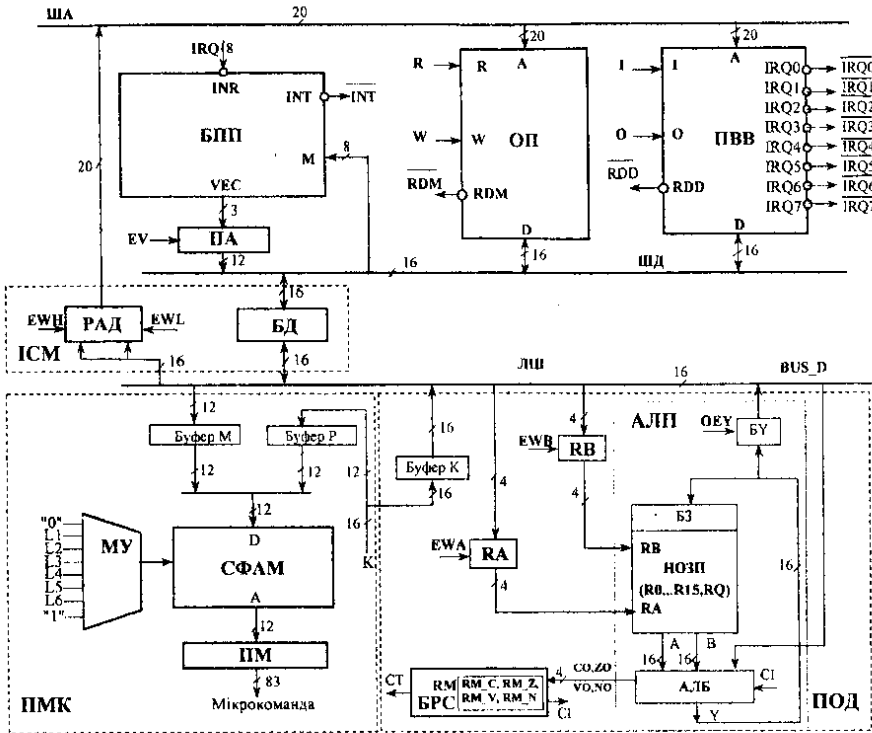


Рис. 16.1. Структура EOM

Табл. 16.1

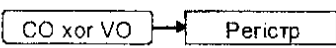
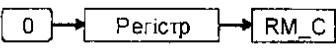
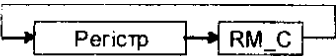
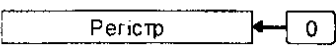
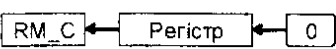
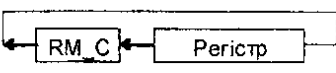
Мнемоніка	Мікрооперація в АЛБ
ADD	R+S+CI
SUB	R-S-I+CI
OR	R or S
AND	R and S
NAND	not(R and S)
XOR	R xor S
NXOR	not(R xor S)

Ознаки можуть бути записані у відповідні розряди регістра RM, що входить до складу БРС. Ознаки, записані в RM, позначаються як RM\_C, RM\_Z, RM\_N, RM\_V. Логічні ланцюги БРС забезпечують підключення до

входу С1 АЛБ прямого або інверсного значення ознаки  $RN\_C$ , а також сигналів логічного нуля або одиниці. Біт  $RM\_C$  бере участь також у деяких мікроопераціях зсуву. Різновиди зсувів зазначені в табл. 16.2.

ПМК забезпечує аналіз логічних умов, що надходять на входи  $L1, L2, \dots, L6$ . До входів логічних умов можуть бути підключені зовнішні відносно ПМК сигнали ( $RDM, RDD, CT$  та ін.). З виходу  $CT$  БРС можуть бути видані прямі та інверсні значення ознак, що формуються безпосередньо в АЛБ, а також зберігаються в  $RM$ . ПМК формує керуючі сигнали, що забезпечують роботу всіх компонентів системи. (На рис. 16.1 зазначені тільки ті сигнали, що використовуються при записі мікрокоманд на мікроасемблері). Виходи буфера  $K$  забезпечують видачу константи  $K$  на ЛШ. Поле  $K$  константи належить слову мікрокоманди. Константа  $K$  може прийматися в ПОД, а також через буфери  $P$  і  $M$  у ПМК. Виходи буфера  $M$  використовуються для завдання початкової адреси мікропрограми, а буфера  $P$  – для завдання адреси переходу в мікропрограмах.

Табл. 16.2

Мнемоніка	Найменування	Схема зсуву
SRA	Зсув вправо арифметичний	
SRL	Зсув вправо логічний	
SR.9	Зсув вправо з переносом	
SLA	Зсув вліво арифметичний	
SLL	Зсув вліво логічний	
SL.25	Зсув вліво з переносом	

## 17. ВВЕДЕННЯ В МІКРОАСЕМБЛЕР

**Загальні відомості про мікроасемблер.** Мнемонічний двохпрохідний мікроасемблер призначений для розробки мікропрограм.

Результатом роботи мікроасемблера є файл даних з розширенням ".pmk", який є вихідним для програмного емулятора системи на рис. 16.1.

Процес трансляції вихідного файлу здійснюється за два проходи. Під час першого проходу відбувається визначення обсягу вихідного файлу, формуються таблиці міток і відповідностей, а також проводиться попередній синтаксичний аналіз. Під час другого проходу безпосередньо формуються коди мікрокоманд. У випадку виявлення синтаксичної або семантичної помилки в вихідному тексті процес трансляції припиняється з видачею повідомлення про характер помилки і рядка тексту, на якому вона була виявлена.

Вихідним файлом для мікроасемблера є текстовий файл в кодах ASCII з розширенням ".asm". Розходження між заголовними і малими літерами мікроасемблером не сприймаються. Між окремими мнемоніками може бути будь-яке число службових символів (наприклад, пробіл, табуляція, повернення каретки, переклад рядка і т. ін.).

Строге дотримання правил написання мікропрограми, акуратність в наборі тексту прискорюють трансляцію і налагодження. Більшість помилок виникає, насамперед, через недбалий стиль написання і неточне знання самого об'єкта розробки.

**Коментарі, числові константи і мітки.** Коментарі використовуються для пояснень. Ознакою початку коментарю служить символ ";". Далі мікроасемблер ігнорує всі символи, які зустрічаються до наступного символу "\n" або до кінця рядка.

*Приклад.*

```
\ *****  
\          Приклад коментарів у мікропрограмі  
\ *****  
{ add R11, R11, R10,Z; } \ додати до R11 вміст R10
```

Числові константи застосовуються при завданні значень операндів і адрес. Ознакою числової константи є цифра на початку мнемоніки.

*Приклади.*

65535	\ десяткова константа
0FFFFh	\ шістнадцяткова константа
177777o	\ вісімкова константа
1111111111111111%	\ двійкова константа

Мітки можуть включати до 10 символів (букви, цифри і символ "\_"), причому, першим символом повинна бути буква. Ознакою кінця мітки служить кожний з наступних роздільників: пробіл, повернення каретки, пере-

ведення рядка, табуляція. Мітка не повинна збігатися з зарезервованою мнемонікою.

*Приклади.*

```
Loop
First_go
Мітка_1
LABEL1
```

**Мікрокоманди АЛП.** Арифметичні мікрокоманди, що виконуються в АЛП, записуються у вигляді

**<мнемоніка> [<оператор\_зсуву>,<]&nbsp; [<приймач\_результату>,<]&nbsp; <джерело\_1>,<джерело\_2>,<вхідний\_перенос>**

У квадратних дужках зазначені необов'язкові елементи конструкції.

Запис логічних мікрокоманд відрізняється від арифметичних відсутністю останнього операнда, тому що перенос не бере участь у логічних мікроопераціях.

Мнемоніка арифметичних і логічних мікрокоманд зазначена в табл. 17.1. Як приймач результату може бути зазначений кожний з регістрів R0 – R15. Крім того, в якості приймача може бути зазначений RQ (якщо зсуву не було), а також NH. В останньому випадку результат у НОЗП не записується, але може бути виданий на локальну шину через БУ. Якщо приймач результату не вказується, то результат записується на місце першого джерела операндів.

Джерелами операндів можуть бути регістри НОЗП, а також один регістр (він вказується як перше джерело операндів) у комбінації з константою, BUS\_D чи нулем. Нуль у полі джерела операнда позначається буквою Z. Регістри НОЗП можуть адресуватися непрямою адресацією. Якщо у якості джерела операндів зазначені RA і/або RB, то операнди вибираються з регістрів, коди яких записані в RA і RB. Вхідний перенос може приймати значення 0, 1 (записується відповідно через Z і NZ), а також RM\_C і not RM\_C (інвертоване значення розряду RM\_C). Мнемоніки операторів зсуву зазначені в табл. 17.2.

*Приклади мікрокоманд.*

add srl,r10,r10,r2,z	\відповідає	мікрооперації
	\ r10 := 0,r(r10 + r10)	
xor r5,r5	\ r5 := r5 xor r5 або r5 := 0	
SUB R7,R7,BUS_D,NZ	\ R7 := R7-BUS_D	
and rb,0Ch	\якщо в rb записаний код 011%,	
	\ то r3:=r3 and 0000000000001100%	
add r9,z,nz	\ r9:=r9+1	

Інформація в 4-розрядні регістри адреси RA і RB записується з ЛШ. Комутація ліній ЛШ і входів регістрів задається директивою LINK (див. нижче).

Для завантаження регістра RM в БРС можуть використовуватися наступні мікрокоманди.

Load RM, Z \ встановлення всіх розрядів в нуль  
 Load RM, NZ \ встановлення всіх розрядів в одиницю  
 Load RM, FLAGS \ завантаження всіх ознак, сформованих  
 \ при виконанні мікрооперації в АЛБ

Разом із зазначеними мікрокомандами можуть бути використані мікрокоманди заборони запису в розряди RM, а саме: SEM\_C, SEM\_Z, SEM\_N, SEM\_V.

*Приклад.*

load rm, flags; sem\_v; sem\_n; sem\_z \ забезпечить завантаження  
 \ тільки розряду RM\_C.

**Мікрокоманди ПМК.** У Табл. 17.3 приведені мнемоніки основних мікрокоманд передачі керування, використовуваних у ПМК. У таблиці символом <Y> позначені позиції запису умови, а символом <A> – адреса або мітка.

Табл. 17.3

Мнемоніка	Мікрооперація в ПМК
JZ	Безумовний перехід на нульову адресу
JMAP	Безумовний перехід за адресою на ЛШ
CONT	Безумовний перехід до наступної мікрокоманди
CJP <Y>, <A>	Умовний перехід по зазначеній адресі (мітки)
CJS <Y>, <A>	Умовний перехід до мікропідпрограми по зазначеній адресі (мітки)
CRTN <Y>	Умовне повернення з мікропідпрограми

Як умови можуть бути використані ознаки: RDM, RDD, INT, IRQ, RM\_C, RM\_Z, RM\_N, RM\_V, CO, ZO, NO, VO. Можна також зазначити заперечення умов (наприклад, not RM\_C). Крім того, можна вказати умову Z, що ніколи не виконується, а також умову NZ, що завжди виконується.

**Мікрокоманди керування пристроями та вузлами.** Мнемоніка мікрокоманд керування зовнішніми пристроями, пам'яттю, регістрами і буфером БУ збігається з найменуванням відповідного керуючого сигналу (табл. 17.4).



Табл. 17.4

Мнемоніка	Найменуванням мікрокоманди
R	Читання з ОП
W	Запис в ОП
I	Ввід із зовнішнього пристрою
O	Вивід у зовнішній пристрій
EWH	Запис в старші розряди РАД
EWL	Запис в молодші розряди РАД
OEY	Видача результату Y з БУ на ЛШ
EWA	Запис в RA
EWB	Запис в RB
EV	Видача вектора з ПА на ЛШ

**Мікрокоманди для БПП.** Для роботи з БПП передбачені спеціальні мікрокоманди. Для ініціалізації БПП виконується мікрокоманда `reset`. Сигнали вимоги переривань, що надходять з ПБВ, фіксуються в регістрі IR, що входить до складу БПП. Для читання вектора переривання з виходів VEC використовується мікрокоманда `READ VR`. Для видачі на ЛШ відповідної адреси необхідно в цьому ж такті сформувавши сигнал `EV`. Для переривання, вектор якого зчитаний, необхідно виконати мікрокоманду `CLR IR, VR`, яка забезпечує встановлення нуля у відповідному розряді регістру IR. При ініціалізації системи можна маскувати групу переривань від ПБВ. Для цього необхідно завантажити в БПП маску, використовуючи мікрокоманду `LOAD MR, <маска>`.

**Директиви.** Мікроасемблер підтримує директиву **ORG**, яка має вигляд

`ORG <мітка/адреса>`

Директива забезпечує розміщення виконавчого коду в пам'яті мікрокоманд по зазначеній адресі. Адреса задається у вигляді числової константи або раніше визначеної мітки.

*Приклади.*

`org 010h`  
`org start`

Для завдання відповідностей використовується директива **EQU**, що записується у формі

`EQU <ім'я>:<еквівалент>`

*Приклади.*

`equ start : 10`

```
equ скласти: add
equ операнд2:15
```

Директива **INCLUDE** вставки в трансльовану мікропрограму тексту з файлу записується так:

```
INCLUDE <ім'я файлу>
```

Файл, що задається в директиві повинний знаходитися в тій же директорії, що і трансльований файл.

*Приклади.*

```
include macro.lib
include routine
```

Директива макросу задається за допомогою ключового слова **MACRO** наступним чином

```
MACRO <ім'я> <формальні параметри>:<мікрокоманда>
```

Директива завдання макросів дозволяє конструювати власні мікрокоманди і користуватися ними надалі як стандартними.

*Приклади.*

```
MACRO inc reg : add reg, reg, z, nz
MACRO mov reg1,reg2 : or reg1,reg2,z
macro jmp label : cjp nz, label
```

Ім'я макросу надалі стає для транслятора звичайною стандартною мнемонікою з усіма правилами, що поширюються на неї. Імена формальних операндів макросу не можуть бути зарезервованими мнемоніками. В програмі макрос задається вказівкою свого ключового імені і реальних операндів перерахованих через кому в тій же порядку, що і при завданні макросу.

*Приклади.*

```
inc r4 \ r4:=r4+1
mov r3,r8 \ r3:=r8
jmp delta
```

Директива **DW** завдання значень комірок пам'яті має вигляд

```
DW <адреса>:<значення>
```

*Приклади.*

```
DW 12:0ffh
dw 03Fh:15
```

Директива установки регістрів **ACCEPT** дозволяє встановити початковий стан регістрів R0,...,R15, RA,RB і RM. Форма запису

```
ACCEPT <регістр>:<значення>
```

*Приклади.*

```
ACCEPT RA:12
ACCEPT R4:0fffh
ACCEPT RM: 0101%
```

Директива ACCEPT використовується також для установки стану зовнішніх пристроїв. Можна задати такі характеристики зовнішніх пристроїв (DEV):

- тип пристрою (I - вводу, O - виводу);
- адреса регістра стану (PC) в межах 64К (0000h – 0ffffh);
- адреса регістра даних (RD) в межах 64К;
- затримка в тактах формування сигналу RDD;
- затримка в тактах установки біта "Готовність" в регістрі стану.

*Приклад.*

```
accept dev[2]: i,          \ пристрій вводу
                      30h,  \ адреса PC
                      32h,  \ адреса RD
                      3,    \ затримка сигналу RDM в тактах
                      114   \ затримка установки біта готовності
                          \ в PC після звертання до RD
```

Для пристроїв вводу можна задавати внутрішній буфер даних DEV\_BUF, обсягом до 16 слів.

*Приклад.*

```
accept dev_buf[2]:1234h,
                  5678h,
                  89abh,
                  0eeeh
```

Значені після ":" дані вводяться в процесор по черзі при кожнім звертанні до RD даного пристрою вводу.

Директива ACCEPT дозволяє задати швидкодію пам'яті за допомогою перемінної RDM\_DELAY, наприклад,

```
ACCEPT RDM_DELAY : 3
```

В даному випадку сигнал RDM буде формуватися з затримкою на 3 такти після видачі сигналу R або W.

Для опису конфігурації зв'язків між компонентами системи використовується директива LINK.

Для установки відповідності входів I1,...,I6 ПМК і логічних умов використовується директива

```
LINK <ім'я_входу>:<умова>
```

*Приклади.*

```
link I1:rdm
link I2:RDD
link I3:ct
```

Підключення 20-розрядного РАД до 16-розрядної ЛШ описується директивою

```
LINK EWH : <номер_розряду>
```

Номер розряду РАД[19...0], зазначений у директиві, розділяє регістр на дві частин. Старша частина, включаючи зазначений розряд, керується сигналом EWH, а молодша - EWL.

*Приклад.*

```
LINK EWH : 16
```

Вказана директива забезпечує зв'язки між ЛШ і РАД так, що по сигналу EWH чотири молодших розрядів на ЛШ записуються в поле РАД[19...16]. По сигналу EWL всі 16 розрядів із ЛШ записуються в РАД[15...0], тобто в молодшу частину регістра.

Підключення вхідної 12-розрядної шини М ПМК до ЛШ задається директивою LINK, в якій перелічуються номери розрядів ЛШ, зв'язані з входами М.

*Приклад.*

```
LINK M: 15,14,13,12,11,10,9,8,7,6,5,4
```

Дана директива визначає підключення до входів М 12-ти старших розрядів ЛШ.

Аналогічно задається підключення до ЛШ входів RA і RB.

Директива LINK також визначає в ПА адресу для різних номерів ПВВ.

*Приклад.*

```
link vec[2]: 0CDEFh
```

У цьому випадку на виходи ПА по сигналу EV видається слово 0CDEFh, якщо на входах є присутнім код 2.

*Приклади.*

```
LINK RA : 11,12,13,14
```

```
LINK RB : 10,9,8,7
```

В один момент часу в різних вузлах системи можуть виконуватися різні мікрооперації. Всі мікрокоманди, що керують мікроопераціями, які виконуються в одному такті, записуються в операторних дужках "{" і "}", утворюючи повну мікрокоманду для даного такту роботи ЕОМ. Окремі мікрокоманди розділяються символом ";" і, крім того, можуть використовуватися роздільники типу "пробіл", "повернення каретки", "переведення рядка". Повна мікрокоманда може займати кілька рядків в тексті мікропрограми. При необхідності мітка записується перед операторною дужкою, яка відкривається.

Розглянемо приклади мікропрограм.

*Приклад.*

Встановити нулі в чотирьох старших розрядах РАД (нульова сторінка ОП) і записати в молодші розряди цього регістра адресу з R7. Прийняти

слово з ОП в регістр R15. Сигнал готовності RDM формується рівнем логічного нуля.

```

link l1:rdm          \підключення виходу RDM до l1
link ewh:16          \установка зв'язків між РАД і ЛПШ
accept rdm_delay:2   \затримка формування RDM
                    accept r7:1234h      \вихідна установка R7 (для
                                        \налагодження)
dw 1234h:070fh      \визначення даних в ОП за
                    \адресою 1234h
{cont;xor nil,r0,r0;oeu;ewh;} \РАД[19...16]:=0
{cont;or nil,r7,z;oeu;ewl;}  \РАД[15...0]:=r7
ll1 {cjp rdm,l1;r;or r15,bus_d,z;} \R15:=070fh (дані з ОП)
    {} \кінець мікропрограми

```

Зауважимо, що мікрокоманду cont записувати не обов'язково.

*Приклад.*

Підсумувати коди в регістрах R1,R2 і R15. Записати подвоєний результат в пам'ять за адресою, яка записана в регістрі, зазначеному в RA.

```

link l1:rdm
link ewh:16
accept rdm_delay:3
accept ra:3
accept r3:0004h
accept r1:4
accept r2:16
accept r15:32
{xor nil,r0,r0;oeu;ewh;} \РАД[19...16]:=0
{add r1,r1,r2,z;}        \R1:=R1+R2
{add sla,r1,r1,r15,z;}  \R1:=!(R1+R15).0 (результат в R1)
{or nil,ra,z;oeu;ewl;}  \запис адреси в РАД
ll2 {cjp rdm,ll2;w;or nil,r1,z;oeu;} \запис результату в пам'ять
    {} \кінець мікропрограми

```

*Приклад.*

Мікропрограма налаштування БПП і читання вектора переривання з ПА в АЛП.

```

link vec[2]: 0CDEFh      \ завдання в ПА адреси для переривання від
                        \ 2-го ПБВ
link L[4]:INT           \ підключення виходу INT до входу l4 ПМК
{reset;}               \ ініціалізація БПП

```

```

{load mr,11111011%;} \ дозволити обробку переривання IRQ2
< мікропрограма виконання команди, включаючи формування адреси на-
ступної команди>
{cjp not INT, MakeInt;} \ якщо INT=0, то перейти до читання вектора
< інакше почати обробку наступної команди >
MakeInt \ початок обробки переривання
{ev;read vr;or r9,bus_d,z;} \ читання вектора в R9
{clr ir,vr;} \ скидання розряду в регістрі IR БПП, що
\ відповідає даному перериванню
< продовження мікропрограми обслуговування переривань >

```

*Приклад.*

Для системи з розподіленим контролером переривань настроїти ВПП з номером 4 на ввід даних, ВПП з номером 6 на вивід даних.

```

ACCEPT DEV[4]: 1, \ пристрій вводу
                1234h, \ адреса PC
                1236h, \ адреса PD
                10, \ затримка в тактах формування RDD
                50, \ затримка в тактах формування сигналу
                \ готовності в PC
                4444h \ вектор переривання
ACCEPT DEV[6]: 0, \ пристрій виводу
                12a4h, \ адреса PC
                12a6h, \ адреса PD
                8, \ затримка в тактах формування RDD
                30, \ затримка в тактах формування сигналу
                \ готовності в PC
                444eh \ вектор переривання

```

Присутність в опису ВПП вектора переривань автоматично забезпечує роботу ВПП у складі пріоритетного ланцюжку. У протилежному випадку ВПП може працювати в режимі переривань тільки разом з централізованим БПП.

## 18. ВИМОГИ ДО ОФОРМЛЕННЯ ТЕХНІЧНОЇ ДОКУМЕНТАЦІЇ

**Технічне завдання** розробляється студентом на підставі вихідних даних відповідно до діючих стандартів. У технічному завданні повинні бути наступні розділи:

- *призначення* проектованого об'єкта, у якому розкриваються його області застосування (коло розв'язуваних задач);

- *склад* пристрою, у якому приводиться перелік основних складових частин проектованого пристрою відповідно до ієрархічного принципу його побудови;

- *технічні вимоги*, що розділяються на загальні і приватні. У загальних вимогах вказуються умови експлуатації, вимоги перешкодозахищеності й електричному режиму роботи, що комплектують вироби, умови збереження, безпеки, експлуатації, транспортування і т.п. У приватних – обумовлюються основні структурні й алгоритмічні особливості проектованого пристрою, як наприклад, система команд, формати команд, система числення, довжина машинних слів, форма представлення чисел, тривалість операцій, ємність пам'яті і будь-які інші технічні вимоги;

- *вимоги надійності*, у яких указують припустимі значення основних характеристик надійності;

- *конструктивні вимоги*, що визначають типи і номенклатуру використовуваних елементів, кріпильних і комунікаційних виробів, ступінь уніфікації і стандартизації окремих конструкторських рішень, особливості технологічного процесу виготовлення проектованого об'єкта;

- *етапи* проектування і терміни їхнього виконання;

- *перелік* текстової і графічної документації.

Технічне завдання повинне бути підписане виконавцем і керівником проекту.

**Порядок побудови розділів і підрозділів** пояснювальної записки, правила викладу тексту, розрахунків, а також побудови таблиць повинні цілком відповідати вимогам діючих стандартів.

Текст виконується авторським рукописом чорним чорнилом (заголовки, формули, цифри і таблиці повинні виконуватися креслярським шрифтом) або роздруковується на принтері.

Усі документи повинні мати позначення, яке має вигляд:

XXXX.XXXXXX.XX  
позначення виробу

XXXX  
код, (шифр) документа

У свою чергу, позначення виробу має наступну структуру:

XXXX  
код організації роз-  
роблювача

XXXXXX  
код класифікаційної  
характеристики

XXX  
реєстраційний номер  
документа

Реєстраційний номер звичайно привласнюється кожному документові в межах коду організації розроблювача. Код кваліфікаційної характеристики має структуру:

XX  
клас

X  
підклас

X  
група

X  
підгрупа

X  
вид

Даний код визначають відповідно з класифікатором виробів.

Перед комплектацією всі документи курсової роботи повинні бути підписані виконавцем і керівником на титульному листі і в основних написах відомостей і креслень.

Підпис керівника про допуск до захисту ставиться після остаточного оформлення альбому, що містить документацію по курсовому проекту.

**Функціональна схема** визначає основні функціональні частини виробу, їх призначення і взаємозв'язок, роз'ясняє визначені процеси, що протікають в окремих функціональних частинах або у виробі в цілому.

Схеми електричні функціональні можуть виконуватися як на весь виріб, так і на його окремі функціональні частини. Функціональні частини таких схем зображуються, як правило, у виді прямокутників.

Допускається використання у функціональних схемах умовних графічних позначень (УГП) деяких функціональних частин, наприклад, комбінаційних елементів, суматорів, дешифраторів, елементів пам'яті і т.ін.

УГП елемента має форму прямокутника, до якого підводять лінії виводів. УГП елемента в загальному випадку може містити три поля: основне і два додаткових, котрі розташовують ліворуч і праворуч від основного (рис. 18.1).



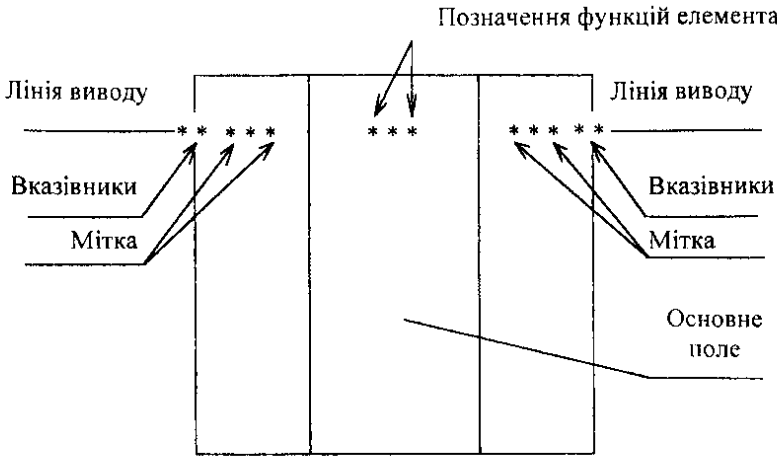


Рис. 18.1

В основному полі УГП поміщують позначення функції, яка реалізується елементом.

В додаткових полях поміщують інформацію про призначення виводів (мітки виводів, вказівники). Допускається проставляти вказівники на лініях виводів.

Крім виду, зазначеного на рис 18.1, УГП може також складатися:

- тільки з основного поля;
- з основного поля і одного додаткового (праворуч або ліворуч від основного);

Допускаються додаткові поля розділяти на зони, відокремлені горизонтальною рисою.

Основне і додаткове поля можуть бути не відділені лінією. При цьому відстань між буквеними, цифровими і буквено-цифровими позначеннями, поміщеними в основне і додаткові поля, визначається однозначністю розуміння кожного позначення.

Входи елемента зображують з лівої сторони УГП, виходи – з правої сторони УГП. Двонаправлені виводи і виводи, що не несуть логічної інформації, зображують з правої або лівої сторони УГП.

При підведенні ліній виводів до контуру УГП не допускається:

- проводити їх на рівні сторін прямокутника;
- проставляти на них у контурі УГП стрілки, що вказують напрямок передачі інформації.

Табл. 18.1

Найменування	Позначення
Тригер	T
Двоступінчастий тригер	ТТ
Логічне "І"	&
Логічне "АБО"	$\geq 1$ або 1
Виключне "АБО"	=1
Генератор імпульсів	G

Табл. 18.2


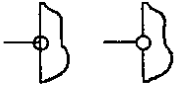
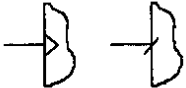
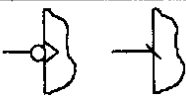
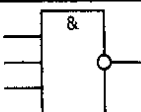
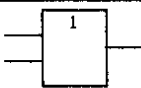
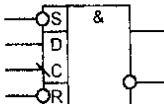
Найменування	Позначення
Прямий статичний вивід	
Інверсний статичний висновок	
Прямий динамічний вхід	
Інверсний динамічний вхід	

Табл. 18.3

Найменування	Позначення
Елемент 3І-НЕ	
Елемент 2АБО	
D-тригер	

Розміри сторін УГП повинні бути кратні  $M$ , а відстані між висновками –  $2M$ . Діаметр вказівника інверсного виводу повинний дорівнювати  $M$ . Величина  $M$  вибирається виходячи з вимог мікрофільмування. При виконанні курсової роботи прийняти  $M=2.5$  мм.

Позначення функцій, виконуваних елементом, утворюють із прописних букв латинського алфавіту, арабських цифр і спеціальних знаків, записаних без пробілів. Кількість знаків у позначенні функції не обмежена, однак варто прагнути до їх мінімального числа при збереженні однозначності розуміння кожного позначення.

В таблиці 18.1 приведені деякі стандартні позначення функцій, яких варто дотримуватись при виконанні курсової роботи.

Виводи елементів, які несуть логічну інформацію, підрозділяють на статичні і динамічні, а також на прямі та інверсні. Властивості виводів позначають за допомогою міток відповідно до табл. 18.2.

Приклади деяких УГП приведені в табл. 18.3.

*ДОДАТОК*

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра обчислювальної техніки

КУРСОВА РОБОТА (КУРСОВИЙ ПРОЕКТ)

по дисципліні "(Назва дисципліни)"

Виконав \_\_\_\_\_  
(прізвище, ім'я, по батькові)

Факультет \_\_\_\_\_

Група \_\_\_\_\_

Залікова книжка № \_\_\_\_\_

Допущений до захисту \_\_\_\_\_

Номер технічного завдання \_\_\_\_\_

\_\_\_\_\_  
(підпис керівника)

Київ -- 200\_ р.

## ЗМІСТ

Введення	3
1. Програмний комплекс для моделювання логічних схем	4
2. Лабораторна робота №1. Проектування комбінаційних схем	11
3. Лабораторна робота №2. Мінімізація перемикальних функцій	17
4. Лабораторна робота №3. Мінімізація систем перемикальних функцій	27
5. Лабораторна робота №4. Мінімізація частково визначених функцій	32
6. Лабораторна робота №5. Проектування і дослідження тригерів	39
7. Лабораторна робота №6. Проектування і дослідження регістрів	51
8. Лабораторна робота №7. Проектування і дослідження лічильників	63
9. Лабораторна робота №8. Проектування і дослідження суматорів	77
10. Лабораторна робота №9. Проектування і дослідження керуючих автоматів	87
11. Лабораторна робота №10. Синтез автоматів з використанням часових функцій	100
12. Лабораторна робота №11. Проектування і дослідження пристроїв для множення чисел	111
13. Лабораторна робота №12. Проектування і дослідження пристроїв для ділення чисел	120
14. Курсова робота	124
15. Курсовий проект	130
16. Програмний комплекс моделювання ЕОМ	138
17. Введення в мікроасемблер	142
18. Вимоги до оформлення технічної документації	151
Додаток	156