

Міністерство освіти і науки України
Національний авіаційний університет
Навчально-науковий інститут комп'ютерних інформаційних технологій
Кафедра комп'ютеризованих систем управління

Розрахунково-графічна робота
з дисципліни «Комп'ютерна схемотехніка»

Виконав:
студент ННІКІТ
групи СП-225
Клокун В. Д.
Перевірив:
Іскренко Ю. Ю.

Київ 2018

1 Завдання

Завданням розрахунково-графічної роботи є розробка алгоритму виконання вказаної в завданні операції та синтезу функціональної схеми керуючого автомата.

Табл. 1: Завдання на розрахунково-графічну роботу

Параметр	Значення
№	16
Тип операції	Додавання
Початковий код операндів	ДК
Розрядність операндів	16
КВМСМ	МДК
Структура ОБ	ЗМО
Тип автомата	Мілі
Пам'ять автомата	D
ОР	P
ЛО	$NAND$

З завдання на розрахунково-графічну роботу (табл. 1) видні такі характеристики цільового арифметико-логічного пристрою:

1. Тип арифметичної операції — додавання двійкових чисел.
2. Початковий код подання операндів — доповняльний.
3. Розрядність операндів — 16 біт.
4. Код виконання операції у суматорі — доповняльний модифікований.
5. Структура операційного блока — із закріпленими мікроопераціями.
6. Тип керуючого блока — автомат Мілі з пам'яттю на D -тригерах.
7. Схема логічної ознаки парності молодшого байту.

2 Хід роботи

2.1 Розробка алгоритму

Алгоритм додавання двійкових чисел можна словесно описати так:

1. У першому і другому машинних тактах із вхідної шини паралельним кодом записуються операнди A і B у відповідні регістри RGA і RGB . Зчитування операндів здійснюється ЦПК.

2. Протягом одного машинного такту виконується мікрооперація додавання.
3. Якщо розрядна сітка не переповнилась, результат записується у регістр *RGC*.
4. Якщо розрядна сітка переповнилась, результат не фіксується, і в ЦПК подається сигнал переповнення «ПП».

2.2 Розробка функціональної схеми для виконання додавання

Розроблюємо функціональну схему для виконання додавання двох двійкових чисел (рис. 1), яка містить:

1. Регістри *RGA*, *RGB* для приймання і подальшого зберігання першого і другого операндів із вхідної шини *Ш1*.
2. Паралельний комбінаційний суматор з додатковим старшим розрядом знака *П* для створення модифікованого доповняльного коду.
3. Регістр результату *RGC*, дані з якого пересилаються в оперативну пам'ять по вихідній шині *Ш2*.
4. Схеми електронних ключів *SW1* і *SW2*.
5. Схему вироблення ознак переповнення *OP*.
6. Схему диз'юнкторів *OR* для виконання операцій порозрядного логічного додавання кодів операндів *A* і *B*.

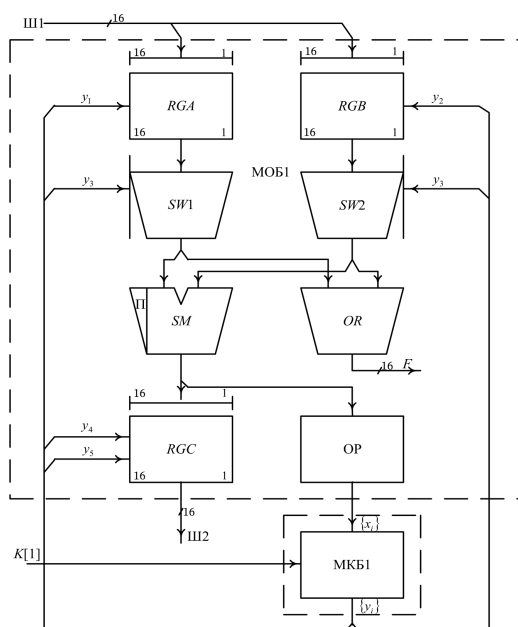


Рис. 1: Функціональна схема для додавання чисел

Після закінчення операції керуючий автомат аналізує ознаки результату і встановлює значення відповідних тригерів ознак. Ознаки результату обчислюються за допомогою булевих виразів:

$$\begin{aligned}\varphi_1 &= \Pi \wedge \overline{SM}[n] \text{ — додатний результат (знаки 00),} \\ \varphi_2 &= \Pi \wedge SM[n] \text{ — від'ємний результат (знаки 11),} \\ \varphi_3 &= \overline{\Pi} \wedge SM \vee \Pi \wedge \overline{SM}[n] \text{ — переповнення розрядної сітки (знаки 01 чи 10),} \\ \varphi_4 &= \bigwedge_{i=1}^{n+1} \overline{SM}[i] \text{ — нульовий результат.}\end{aligned}$$

Ознака переповнення перевіряється до закінчення операції і за її наявності виконання програми переривається. Перевірка ознаки *OR* реалізується за допомогою 16 логічних 2-входових елементів АБО за співвідношенням:

$$F_i = A_i \vee B_i, \quad i = \{1, \dots, 16\},$$

де F_i — i -й вихід вузла логічного додавання. Ця операція виконується автоматично незалежно від коду команди.

2.3 Розробка мікропрограми додавання

За словесним алгоритмом додавання двійкових чисел у доповняльних кодах запишемо мікропрограму (алг. 1). Отримана мікропрограма дозволяє скласти змістовний граф мікропрограми (рис. 2). В свою чергу отриманий змістовний граф мікропрограми кодується та розмічується. В результаті отримуємо закодований граф мікропрограми (рис. 3).

Алгоритм 1: Додавання двійкових чисел		
1	якщо $K[1]$ то	<i>/* $K[1]$ — однорозрядний код команди додавання */</i>
2	$RGA := A;$	<i>/* Приймання першого операнда */</i>
3	$RGB := B;$	<i>/* Приймання другого операнда */</i>
4	$SM := A + B;$	<i>/* Додавання */</i>
5	якщо φ_3 то	
6	$T_{\Pi} := \text{ПП};$	<i>/* Тригеру переповнення T_{Π} присвоюється ознака ПП */</i>
7	інакше	
8	$RGC := SM;$	<i>/* Присвоєння результату */</i>
9	$III2 := RGC;$	<i>/* Пересилання в пам'ять */</i>
10	кінець	
11	інакше	
12	Чекати;	
13	кінець	

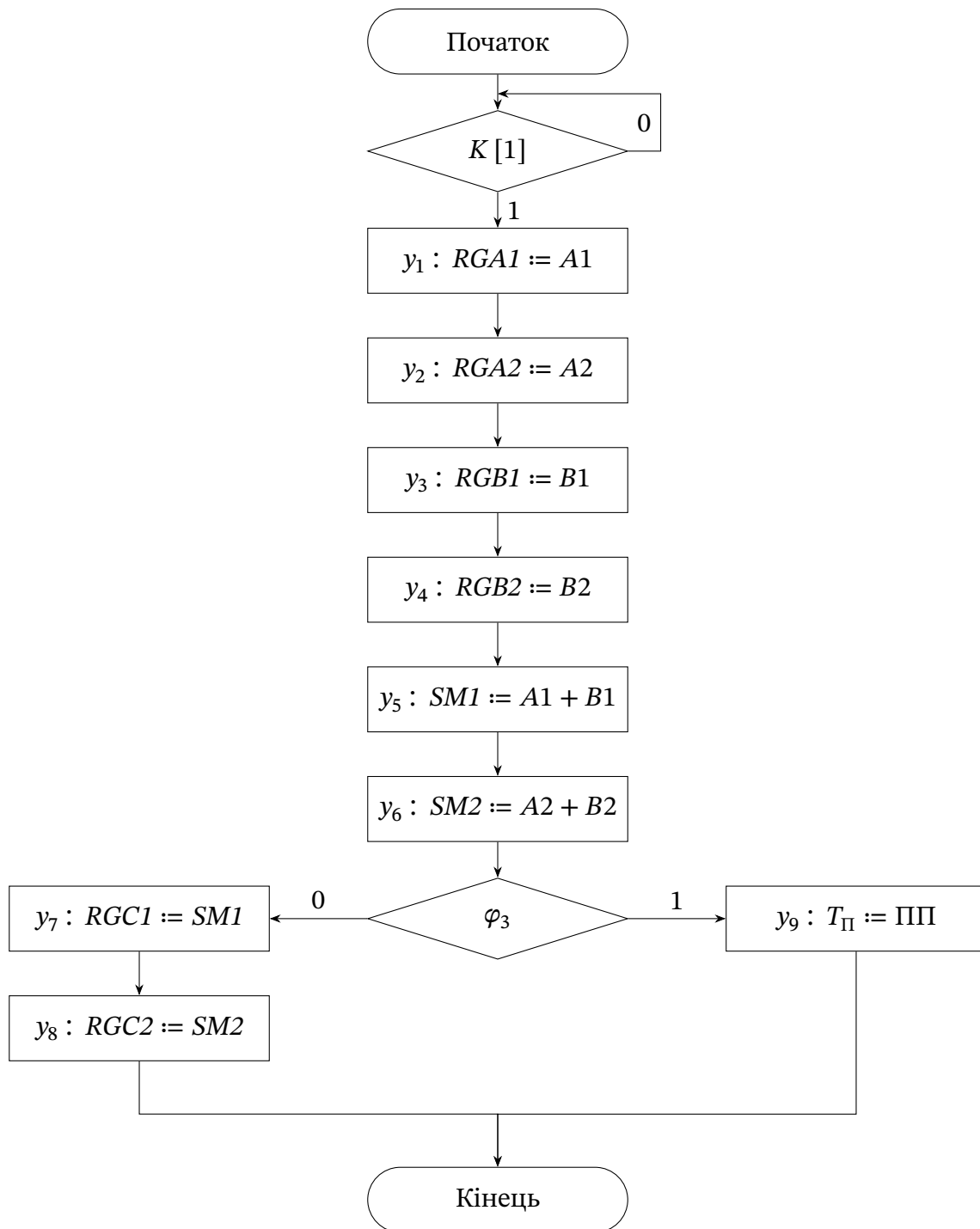


Рис. 2: Змістовний граф мікропрограми додавання

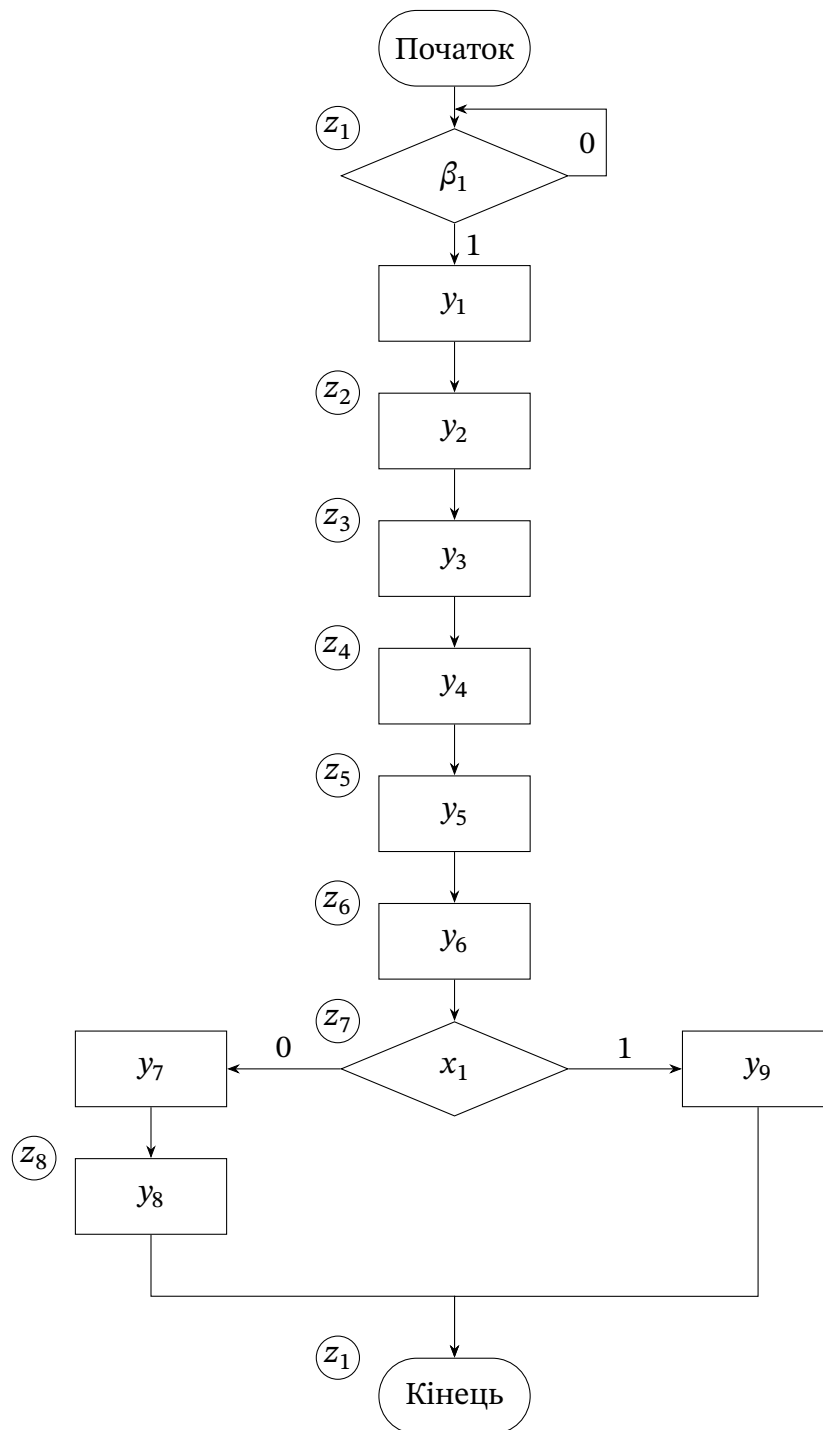


Рис. 3: Закодований граф мікропрограми додавання

2.4 Розробка схеми модуля операційного блока

Отримані дані дозволили розробити модуль операційного блока (рис. 4).

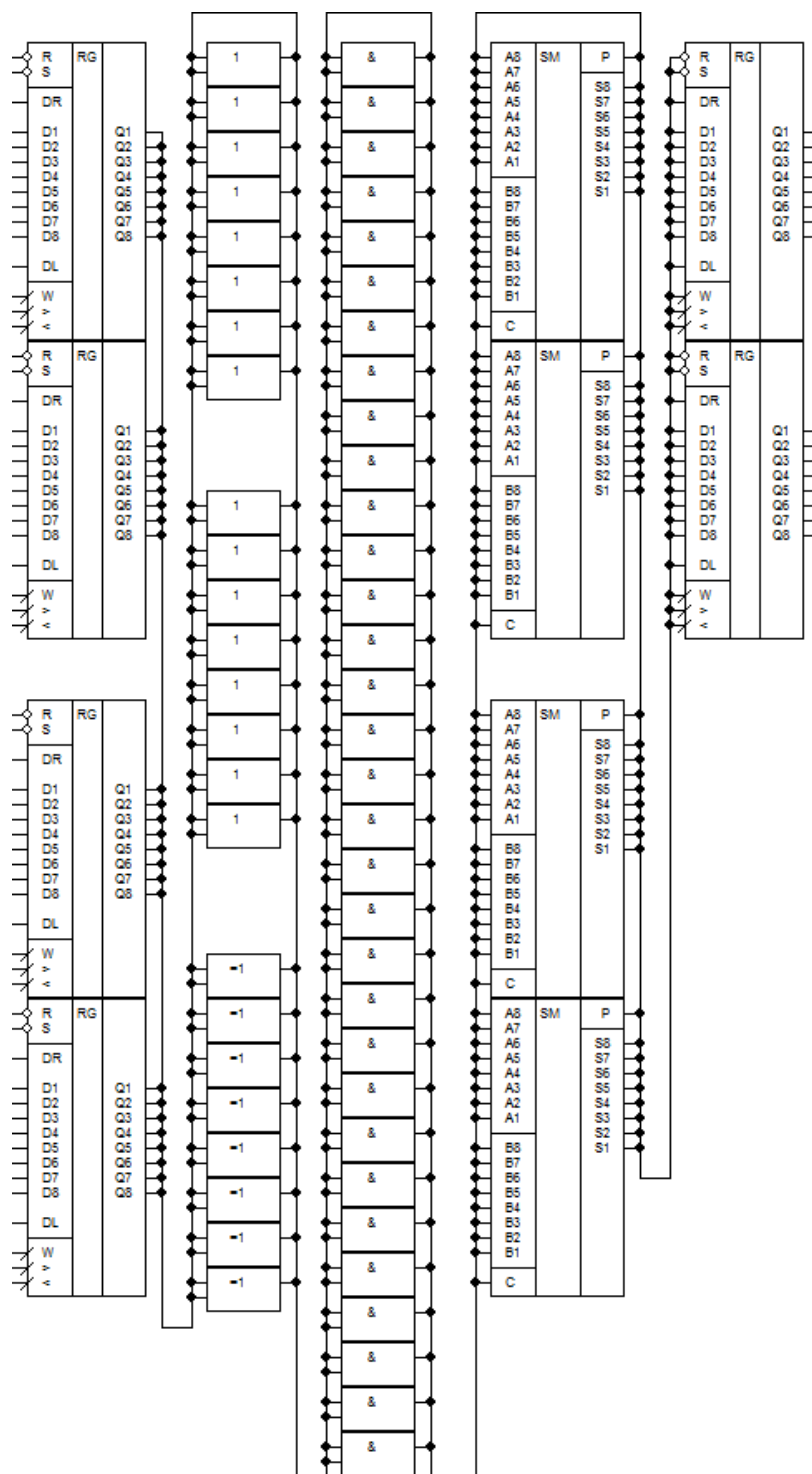


Рис. 4: Схема модуля операційного блока

Розроблений модуль операційного блока складається з таких елементів:

1. Регістри для зберігання операндів.
2. Мікросхеми логічного АБО для реалізації порозрядної диз'юнкції над кодами операндів A і B .
3. Мікросхеми логічного І для підключення виходів регістрів до входів суматора.
4. Мікросхеми логічного виключного АБО для створення старшого знакового розряду суматора і логічної ознаки $\varphi_3 = x_1$ та $\overline{x_1}$.
5. Вихідний регістр для приймання результату додавання та його передачі на вихідну шину.
6. Суматори для виконання операції додавання.

2.5 Проектування керуючого блоку

Із закодованого графа мікропрограми видно, що максимальна кількість станів автомата $L = 8$, тому для реалізації необхідно $n = \lceil \log_2 8 \rceil = 3$ D -тригери. Закодуємо стани автомата значеннями виходів D -тригерів за принципом кодування Грея:

$$\begin{aligned} z_1 &= \overline{Q_1} \overline{Q_2} \overline{Q_3}, & z_2 &= \overline{Q_1} \overline{Q_2} Q_3, & z_3 &= \overline{Q_1} Q_2 Q_3, & z_4 &= \overline{Q_1} Q_2 \overline{Q_1}, \\ z_5 &= Q_3 Q_2 \overline{Q_1}, & z_6 &= Q_3 Q_2 Q_1, & z_7 &= Q_3 \overline{Q_2} Q_1, & z_8 &= Q_3 \overline{Q_2} \overline{Q_1}. \end{aligned}$$

На основі отриманих даних складаємо граф автомата Мілі (рис. 5).

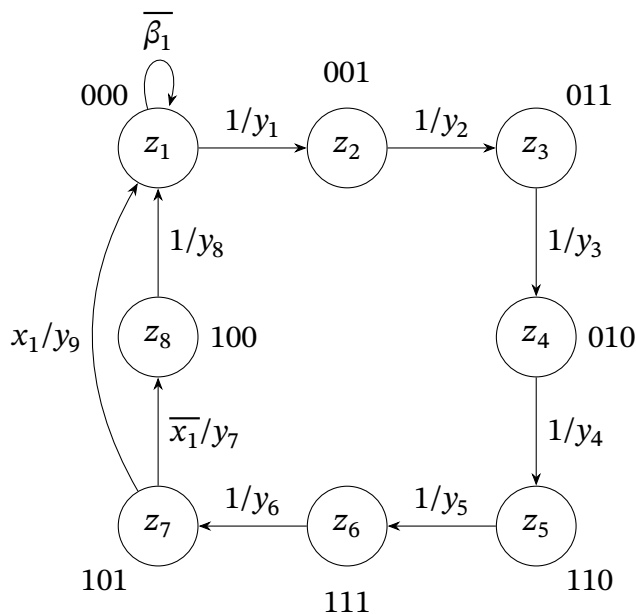


Рис. 5: Граф автомата Мілі для мікропрограми додавання

Отриманий граф автомата Мілі для мікропрограми додавання двох двійкових чисел у доповняльних кодах дозволяє скласти структурну таблицю переходів автомату Мілі (табл. 2), яка знадобиться для подальших обчислень і є більш наочною.

Табл. 2: Структурна таблиця переходів автомата Мілі

z_i	$k(z_i)$	z_j	$k(z_j)$	$\{x_i\}$	$\{y_i\}$	D_1	D_2	D_3
z_1	000	z_1	000	$\overline{\beta_1}$	—	0	0	0
z_1	000	z_2	001	1	y_1	0	0	1
z_2	001	z_3	011	1	y_2	0	1	1
z_3	011	z_4	010	1	y_3	0	1	0
z_4	010	z_5	110	1	y_4	1	1	0
z_5	110	z_6	111	1	y_5	1	1	1
z_6	111	z_7	101	1	y_6	1	0	1
z_7	101	z_1	000	x_1	y_9	0	0	0
z_7	101	z_8	100	$\overline{x_1}$	y_7	1	0	0
z_8	100	z_1	000	1	y_8	0	0	0

На підставі даних структурної таблиці переходів автомату Мілі для мікропрограми додавання записуємо системи логічних рівнянь. Для функцій збудження входів D -тригерів:

$$D_1 = z_4 \vee z_5 \vee z_6 \vee z_7 \overline{x_2}, \quad D_2 = z_2 \vee z_3 \vee z_4 \vee z_5, \quad D_3 = z_1 \vee z_2 \vee z_5 \vee z_6.$$

Перетворимо отримані функції до заданого елементного базису «І—НЕ»:

$$\begin{aligned} D_1 &= z_4 \vee z_5 \vee z_6 \vee z_7 \overline{x_2} = \overline{z_4} \wedge \overline{z_5} \wedge \overline{z_6} \wedge (z_7 \wedge x_2), \\ D_2 &= z_2 \vee z_3 \vee z_4 \vee z_5 = \overline{z_2} \wedge \overline{z_3} \wedge \overline{z_4} \wedge \overline{z_5}, \\ D_3 &= z_1 \vee z_2 \vee z_5 \vee z_6 = \overline{z_1} \wedge \overline{z_2} \wedge \overline{z_5} \wedge \overline{z_6}. \end{aligned}$$

Рівняння для вихідних сигналів:

$$\begin{aligned} y_1 &= z_1, & y_2 &= z_2, & y_3 &= z_3, \\ y_4 &= z_4, & y_5 &= z_5, & y_6 &= z_6, \\ y_7 &= z_7 \overline{x_1}, & y_8 &= z_8, & y_9 &= z_7 x_1. \end{aligned}$$

В результаті розробили функціональну схему керуючого автомату (рис. 6).

3 Висновок

Під час виконання даної розрахунково-графічної роботи ми навчились розробляти мікропрограми для виконання арифметично-логічних операцій, синтезувати за розробленим алгоритмом відповідні керуючі автомати та реалізовувати синтезовані автомати у вигляді функціональних схем.

