МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ВОЛОДИМИРА ДАЛЯ

МЕТОДИЧНІ ВКАЗІВКИ

**до практичних занять
з дисципліни**

«Комп'ютерна логіка та цифрові автомати»

для здобувачів першого (бакалаврського) рівня освіти
за спеціальністю
F3 – Комп'ютерні науки
F7 – Комп'ютерна інженерія
(Електронне видання)

Затверджено
на засіданні кафедри
"Комп'ютерних наук та інженерії"
Протокол № 17 від 03.06.2025 р.

Київ 2025

УДК 681.32 (075.8)

Методичні вказівки до практичних занять з дисципліни “Комп’ютерна логіка та цифрові автомати“ для здобувачів першого (бакалаврського) рівня освіти за спеціальністю F3 – Комп’ютерні науки, F7 – Комп’ютерна інженерія (Електронне видання) / Уклад.: О.К. Лифар. – Київ: вид-во СНУ ім. В. Даля, 2023. - 225 с.

Дані методичні вказівки є навчальним виданням для курсу “Комп’ютерна логіка та цифрові автомати“, який викладається для здобувачів вищої освіти галузі знань F – Інформаційні технології.

Методичні вказівки містять основні програмні положення курсу, спрямовані на формування у студентів знань і навичок в питаннях щодо вивчення основ спеціального математичного апарату, який дозволяє синтезувати цифрові обчислювальні пристрої на двійкових елементах із заданим законом функціонування.

Пропоновані методичні вказівки сприяють розвитку у здобувачів першого (бакалаврського) рівня освіти як загальних, так і професійних компетентностей. Розглянуто основні питання, які дають можливість розширення, поглиблення та деталізації теоретичних знань, отриманих на лекціях та в процесі самостійної роботи, прищеплення умінь і навичок з рішення задач комп’ютерної арифметики, синтезу комбінаційних схем та автоматів з пам’яттю.

Укладач: О. К. Лифар

Відповідальний за випуск:

Рецензент: д.т.н., професор О. І. Рязанцев,

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЦЕОМ - цифрова електронна обчислювальна машина

ПК – плаваюча кома

ФК – фіксована кома

ПК – прямий код

ОК – обернений код

ДК – додатковий код

МДК – модифікований додатковий код

ДДА – двійково-десяткова арифметика

ДДНФ – досконала диз'юнктивна нормальна форма

ДКНФ – досконала кон'юнктивна нормальна форма

КС – комбінаційна схема

1 ПЕРЕВЕДЕННЯ ЧИСЕЛ З ОДНІЄЇ ПОЗИЦІЙНОЇ СИСТЕМИ ЧИСЛЕННЯ В ІНШУ

1.1 Мета заняття

Вивчити форми представлення чисел у двійковій, вісімковій і шістнадцятковій системах числення і отримати практичні навички переведення десяткових чисел у вище перераховані системи числення і навпаки.

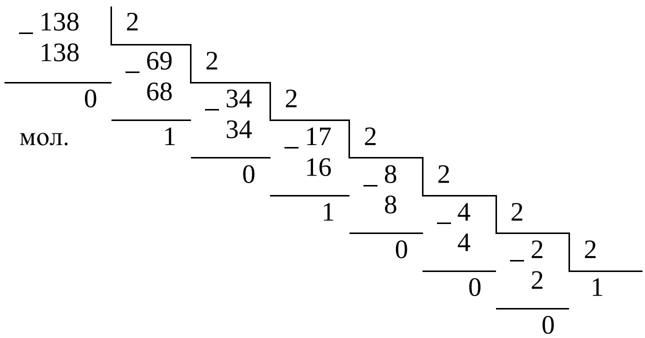
1.2 Короткі основні відомості з теорії та вирішення типових завдань

1.2.1 Переведення цілих чисел

Для переведення цілого числа в нову систему його треба послідовно ділити на основу нової системи числення, записану у вихідній системі числення доти, поки не вийде частка, у якій ціла частина дорівнює нулю (або менше основи нової системи числення). Число в новій системі числення записується із залишків від послідовного ділення, починаючи з останнього, при цьому останній залишок буде старшою цифрою нового числа [1,3,7,11].

Приклад: перевести число $138_{(10)}$ у двійкову, вісімкову і шістнадцяткову системи числення.

Переведення у двійкову систему:



– старший розряд

$138_{(10)} = 10001010_{(2)}$ → потрібно 8 розрядів.

Перевіримо правильність переведення:

$$10001010_{(2)} = \underset{\downarrow}{\underline{1\cdot 2^7}} + \underset{\downarrow}{\underline{0\cdot 2^6}} + \underset{\downarrow}{\underline{0\cdot 2^5}} + \underset{\downarrow}{\underline{0\cdot 2^4}} + \underset{\downarrow}{\underline{1\cdot 2^3}} + \underset{\downarrow}{\underline{0\cdot 2^2}} + \underset{\downarrow}{\underline{1\cdot 2^1}} + \underset{\downarrow}{\underline{0\cdot 2^0}}$$

$$128 + 0 + 0 + 0 + 8 + 0 + 2 + 0 = 138_{(10)}$$

Переведення у вісімкову систему:

$$\begin{array}{r|l}
_\,138 & 8 \\ \cline{2-2}
136 & \begin{array}{r|l} _\,17 & 8 \\ \cline{2-2} 16 & 2 \\ \hline 1 & \end{array} \\ \hline
2 &
\end{array}$$

– старший розряд

мол.

$138_{(10)} = 212_{(8)} \rightarrow$ потрібно 3 розряди.

Перевіримо правильність переведення:

$$212_{(8)} = \underset{\downarrow}{\underline{2\cdot 8^2}} + \underset{\downarrow}{\underline{1\cdot 8^1}} + \underset{\downarrow}{\underline{2\cdot 8^0}}$$

$$128 + 8 + 2 = 138_{(10)}$$

Переведення у шістнадцяткову систему:

$$\begin{array}{r|l}
_\,138 & 16 \\ \cline{2-2}
128 & 8 \\ \hline
10 & \\
\| & \\
A &
\end{array}$$

– старший розряд

мол. розр

$138_{(10)} = 8A_{(16)} \rightarrow$ потрібно 2 розряди.

Перевіримо правильність переведення:

$$8A_{(16)} = \underset{\downarrow}{\underline{8\cdot 16^1}} + \underset{\downarrow}{\underline{10\cdot 16^0}}$$

$$128 + 10 = 138_{(10)}$$

1.2.2 Переведення дробів

Для переведення правильного дробу з однієї позиційної системи числення в іншу його треба послідовно множити на основу нової системи числення доти, поки в новому дробу не буде потрібної кількості цифр, яка визначається необхідною точністю представлення дробу. Правильний дріб у новій системі числення записується із цілих частин добутків, що утворюються при послідовному множенні, причому перша ціла частина буде старшою цифрою нового дробу.

При переведенні чисел з однієї системи в іншу для збереження заданої точності представлення числа (Δ) число розрядів після коми (*k*) повинне вибиратися зі співвідношення [1, 11]:

$$k = \frac{\lg\left(\frac{1}{\Delta}\right)}{\lg p}, \qquad (1.1)$$

де *p* – основа системи числення.

Для $\Delta = 0{,}001 = 10^{-3}$ $\quad \frac{1}{\Delta} = 10^3 \quad \lg\left(\frac{1}{\Delta}\right) = 3 \quad k = \frac{3}{\lg p}$; Так, для

p=2; k=10	p=8; k=4	p=16; k=3

Для $\Delta = 0{,}01 = 10^{-2}$ $\quad \frac{1}{\Delta} = 10^2 \quad \lg\left(\frac{1}{\Delta}\right) = 2 \quad p=2$

$k = \frac{3}{\lg 2} = \frac{3}{0{,}30102} = 6{,}6438 \approx 7$ розрядів.

Приклад*:* перевести правильний дріб $0{,}536_{(10)}$ у двійкову, вісімкову і шістнадцяткову системи числення з точністю $\Delta = 0{,}001$ т.ч. при $p = 2$ $k = 10$; $p = 8$ $k = 4$; $p = 16$ $k = 3$.

Переведення в двійкову систему:

0,	536
	* 2
1,	072
	* 2
0,	144
	* 2
0,	288
	* 2
0,	576
	* 2
1,	152
	* 2
0,	304
	* 2
0,	608
	* 2
1,	216
	* 2
0,	432
	* 2
0,	864

$0{,}536_{(10)} = 0{,}1000100100_{(2)}$

Перевірка правильності переведення:

$0{,}1000100100_{(2)} = 1\cdot2^{-1} + 0\cdot2^{-2} + 0\cdot2^{-3} + 0\cdot2^{-4} + 1\cdot2^{-5} + 0\cdot2^{-6} + 0\cdot2^{-7} + 1\cdot2^{-8} + 0\cdot2^{-9} + 0\cdot2^{-10} = \frac{1}{2} + 0 + 0 + 0 + \frac{1}{32} + 0 + 0 + \frac{1}{256} + 0 + 0 =$

$= \frac{1}{2} + \frac{1}{32} + \frac{1}{256} = \frac{128+8+1}{256} = \frac{137}{256} = 0{,}535(1562)_{(10)}$ → точність до 0,001.

Переведення в вісімкову систему:

0,	536
	* 8
4,	288
	* 8
2,	304
	* 8
2,	432
	* 8
3,	456
	* 8
3,	648
	* 8
8,	184

$0{,}536_{(10)} = 0{,}4223_{(8)}$

Перевірка правильності переведення:

$0{,}4223_{(8)} = 4\cdot8^{-1} + 2\cdot8^{-2} + 2\cdot8^{-3} + 3\cdot8^{-4} =$

$= \frac{4}{8} + \frac{2}{64} + \frac{2}{512} + \frac{3}{4096} = \frac{2048+128+16+3}{4096} = \frac{2195}{4096} = 0{,}535(8886)_{(10)} \approx 0{,}536_{(10)}$

→ точність до 0,001.

Переведення в шістнадцяткову систему:

0,	536 * 16
8,	576 * 16
9,	216 * 16
3,	456 * 16
7,	296

$0{,}536_{(10)} = 0{,}893_{(16)}$

Перевірка правильності переведення:

$0{,}893_{(16)} = 8\cdot16^{-1} + 9\cdot16^{-2} + 3\cdot16^{-3} =$

$$= \frac{8}{16} + \frac{9}{256} + \frac{3}{4096} = \frac{2048+144+3}{4096} = \frac{2195}{4096} = 0{,}535(8886)_{(10)} \approx 0{,}536_{(10)}$$

$\rightarrow$ точність до 0,001.

При переведенні неправильних дробів окремо перетворюють цілу і дробову частини за відповідними правилами. Наприклад, неправильний дріб $138{,}536_{(10)}$, з урахуванням раніше розглянутих прикладів, запишеться в двійковій, вісімковій і шістнадцятковій системах так:

$10001010{,}1000100100_{(2)}$

$212{,}4223_{(8)}$

$8A{,}893_{(16).}$

1.3 Індивідуальні завдання

Дані числа $A_{(10)}$ у десятковій системі числення (конкретні значення чисел для кожного студента задаються по варіантах у табл. 1.1), які являються неправильними дробами. Необхідно виконати і представити в письмовому вигляді хід наступних операцій:

1) перевести задане число в двійкову, вісімкову і шістнадцяткову системи числення. При переведенні дробової частини числа виходити із заданої точності представлення числа Δ=0,001.

2) зробити перевірку правильності переведення шляхом обчислення десяткових еквівалентів чисел.

Таблиця 1.1 – Варіанти завдань

Вар.№	Число $A_{(10)}$	Вар.№	Число $A_{(10)}$	Вар.№	Число $A_{(10)}$
1	411,587	12	443,683	23	381,683
2	333,682	13	343,692	24	669,328
3	436,615	14	428,318	25	640,495
4	493,653	15	454,512	26	561,629
5	363,670	16	241,197	27	507,146
6	355,693	17	256,407	28	590,765
7	371,662	18	278,653	29	680,778
8	427,652	19	213,386	30	812,342
9	430,628	20	379,931	31	331,491
10	406,646	21	366,277	32	456,642
11	777,893	22	254,768	33	238,965

1.4 Контрольні запитання

1) Що називається системою числення?
2) Що називається кодом числа?
3) Що таке довжина числа?
4) Що таке кількісний еквівалент розряду і числа?
5) Які системи числення називаються позиційними?
6) Що така основа системи числення?
7) Системи з якими основами найпоширеніші на практиці?
8) Як у загальному виді можна представити запис n-розрядного числа A в позиційній системі з деякою основою p?
9) Що таке розрядна вага деякої цифри?
10) З яких цифр складається алфавіт двійкової, вісімкової і шістнадцяткової систем числення?
11) Як виглядає запис дробових чисел у позиційних системах числення?
12) Чому дорівнюють мінімальне і максимальне значення числа в позиційній системі числення з основою p?
13) Які два основні методи переведення чисел з однієї позиційної системи в іншу вам відомі?
14) Як здійснюється переведення з однієї системи числення в іншу:
    – цілих чисел,
    – дробових чисел?
15) Як визначити кількість розрядів дробової частини при переведенні чисел з однієї системи числення в іншу для збереження заданої точності?

2 ПЕРЕВЕДЕННЯ ЧИСЕЛ У СИСТЕМИ ЧИСЛЕННЯ ІЗ КРАТНОЮ ОСНОВОЮ

2.1 Мета заняття

Навчитися користуватися методами переведення чисел у системи числення із кратною основою, тобто з двійкової → вісімкову, з двійкової → шістнадцяткову, з вісімкової → двійкову, з шістнадцяткової → двійкову.

2.2 Короткі основні відомості з теорії та вирішення типових завдань

При переведенні цілих чисел з двійкової в вісімкову систему необхідно вихідне двійкове число розбити на тріади (три двійкові розряди) і представити кожну двійкову тріаду однієї цифрою в вісімковій системі. Для переведення в шістнадцяткову систему вихідне число розбивається на тетради (чотири двійкові розряди) [3,7,11].

При переведенні дробів розбивка дробової частини числа проводиться починаючи зі старшого розряду (з ліва на право).

Щоб користуватися викладеними вище правилами корисно знати напам'ять таблицю представлення чисел від 0 до 15 у різних системах числення.

Приклади переведення:

– з двійкової в вісімкову

$$\underline{010}\ \underline{001}\ \underline{010}_{(2)},\quad \underline{100}\ \underline{010}\ \underline{010}_{(2)}$$

$$\downarrow\ \ \downarrow\ \ \downarrow\qquad \downarrow\ \ \downarrow\ \ \downarrow$$

$$2\ \ 1\ \ 2_{(8)},\quad 4\ \ 2\ \ 2_{(8)}$$

– з двійкової в шістнадцяткову

$$\underline{1000}\ \underline{1010}_{(2)},\quad \underline{1000}\ \underline{1001}_{(2)}$$

$$\downarrow\ \ \downarrow\qquad \downarrow\ \ \downarrow$$

$$8\ \ A_{(16)},\quad 8\ \ 9_{(16)}$$

Аналогічно здійснюється зворотній перехід, тобто вісімкові числа заміняються відповідними тріадами, а шістнадцяткові – відповідними тетрадами.

Таблиця 2.1 - Системи числення

Системи числення			
десяткова p = 10	двійкова p = 2	вісімкова p = 8	шістнадцяткова p = 16
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

2.2.1 Переведення чисел в систему з кратною основою

Задача 1: Перевести число, яке задане в двійковій системи числення

1101011 , 0100110 (2)

в вісімкову і шістнадцяткову системи числення застосовуючи метод розбивки двійкового числа на тріади і тетради.

Рішення

1) Переведення в вісімкову систему. Розбиваємо вихідне число на тріади і кожній тріаді ставимо у відповідність вісімкову цифру

$$\begin{array}{ccccccc} \underline{001} & \underline{101} & \underline{011} & , & \underline{010} & \underline{011} & \underline{000} & _{(2)} \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow & \\ 1 & 5 & 3 & , & 2 & 3 & 0 & _{(8)} \end{array}$$

Остаточно маємо: 153 , 230 $_{(8)}$

2) Переведення в шістнадцяткову систему. Розбиваємо вихідне число на тетради і кожній тетраді ставимо у відповідність шістнадцяткову цифру:

0110	1011	,	0100	1100	(2)
↓	↓		↓	↓	
6	B	,	4	C	(16)

Остаточно маємо:

$$6B,4C_{(16)}$$

Задача 2: Перевести число, яке задане в вісімковій системі числення, в двійкову систему числення

$$371,265_{(8)}$$

Рішення

1) Переводимо задане число з вісімкової в двійкову систему числення, застосовуючи метод тріад:

3 .	7 .	1 .	,	2 .	6 .	5 .	(8)
↓	↓	↓		↓	↓	↓	
011	111	001	,	010	110	101	(2)

Остаточно маємо:

$$11\ 111\ 001,010\ 110\ 101_{(2)}$$

2) Переводимо отримане число з двійкової системи числення в шістнадцяткову, застосовуючи метод тетрад:

1111	1001	,	0101	1010	1000	(2)
↓	↓		↓	↓	↓	
F	9	,	5	A	B	(16)

Остаточно маємо:

$F9,5AB_{(16)}$

Задача 3: Перевести число, що задане в шістнадцятковій системі числення в двійкову, а потім в вісімкову систему числення, застосовуючи метод тетрад і тріад:

A6C , 4BE $_{(16)}$

Рішення

1) Переводимо задане число з шістнадцяткової системи числення в двійкову, застосовуючи метод тетрад:

A .	6 .	C .	,	4 .	B .	E .	(16)
↓	↓	↓		↓	↓	↓	
1010	0110	1100	,	0100	1011	1110	(2)

Остаточно маємо:

1010 0110 1100 , 0100 1011 1110 $_{(2)}$

2) Переводимо задане число з двійкової системи числення в вісімкову, застосовуючи метод тріад:

101	001	101	100	,	010	010	111	110	(2)
↓	↓	↓	↓		↓	↓	↓	↓	
5	1	5	4	,	2	2	7	6	(8)

Остаточно маємо:

5154 , 2276 $_{(8)}$

2.2.2 Деякі методи переведення чисел з однієї системи числення в іншу, що не вимагають застосування калькулятора

2.2.2.1 Використання схеми Горнера для обчислення поліномів

Зручно при переведенні цілих чисел з двійкової системи числення в десяткову.

Для цього використовується наступне представлення полінома:

$$A_{(p)} = \sum_{i=1}^{n-1} a_i p_i = a_{n-1} \cdot p^{n-1} + a_{n-2} \cdot p^{n-2} + \cdots + a_1 \cdot p^1 + a_0 \cdot p^0 = \\ ((\ldots(a_{n-1} \cdot p + a_{n-2}) \cdot p + a_{n-3}) \cdot p + \ldots + a_1) \cdot p + a_0). \qquad (2.1)$$

При p=2 вираз (2.1) записується так:

$$A_{(2)} = \left((\ldots(a_{n-1} \cdot 2 + a_{n-2}) \cdot 2 + a_{n-3}) \cdot 2 + a_{n-4}\right) \cdot 2 + \ldots + a_1) \cdot 2 + a_0. \qquad (2.2)$$

Тобто, при відомих значеннях розрядів a_i, переведення у десяткову систему числення зводиться до послідовного множення на 2, починаючи зі старших розрядів і додаванням з наступним молодшим розрядом.

Приклад застосування цього правила

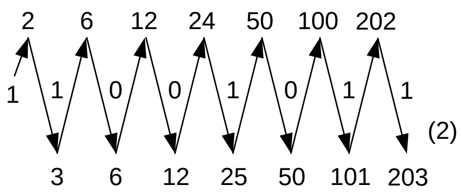
Задане число в двійковій системі числення:

$11001011_{(2)}$

Перевести його в десяткову систему числення, використовуючи схему Горнера для обчислення поліномів:

$$11001011_{(2)} = ((\ldots(1*2+1)*2+0)*2+0)*2+1)*2+0)*2+1)*2+1 = 203_{(10)}$$

Це правило переведення можна представити у вигляді "ялинки":



Перевіримо правильність переведення класичним методом:

$$1\,1\,0\,0\,1\,0\,1\,1 = 1*2^7 + 1*2^6 + 0*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 =$$
$$a_7a_6a_5a_4a_3a_2a_1a_0 = 128 + 64 + 0 + 0 + 8 + 0 + 2 + 1 = 203_{(10)}$$

Цю ж методику можна використовувати для переведення з вісімкової і шістнадцяткової систем числення в десяткову, але при цьому множення потрібно виконувати на 8 і на 16 відповідно.

2.2.2.2 Метод добору доданків

Для швидкого переведення з десяткової системи числення у двійкову можна використовувати метод добору доданків, що складаються із цілих ступенів двійки. Для цього бажано пам'ятати напам'ять цілі ступені числа 2 (див. табл. 2.2).

Таблиця 2.2 - Цілі ступені числа 2

2^i	Десятковий еквівалент
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512
2^{10}	1024
2^{11}	2048
2^{12}	4096

Тоді, наприклад, число $365_{(10)}$ можна представити, як

$$
\begin{array}{lr}
365_{(10)} = & 256 \\
+ & 64 \\
\hline
 & 320 \\
+ & 32 \\
\hline
 & 352 \\
+ & 8 \\
\hline
 & 360 \\
+ & 4 \\
\hline
 & 364 \\
+ & 1 \\
\hline
 & 365
\end{array}
$$

$$365(10) = \underset{2^8}{256} + \underset{2^6}{64} + \underset{2^5}{32} + \underset{2^3}{8} + \underset{2^2}{4} + \underset{2^0}{1} =$$

$$= 1{*}2^8 + 0{*}2^7 + 1{*}2^6 + 1{*}2^5 + 0{*}2^4 + 1{*}2^3 + 1{*}2^2 + 0{*}2^1 + 1{*}2^0 = 101101101_{(2)}$$

Класичне переведення дає наступний результат:

```
_ 365 | 2
  364 |_______
      _ 182 | 2
        182 |______
-----       _ 91 | 2
    1         90 |______
    ------       _ 45 | 2
         0         44 |______
         ------       _ 22 | 2
              1         22 |______
              ------       _ 11 | 2
                   1         10 |______
                   ------       _ 5 | 2
                        0         4 |______
                        ------      _ 2 | 2
                             1        2 |______
                             ------        1
                                  1
                                  ------
                                       0
```

$1011011101_{(2)}$ – Результат той же самий!

2.4 Індивідуальні завдання

Необхідно виконати і представити в письмовому вигляді хід наступних операцій.

1) Перевести задане число в вісімкову і шістнадцяткову системи числення, використовуючи правило розбивки двійкових чисел на тріади і тетради (табл.2.3).

2) Перевести число, задане в вісімковій системі числення в шістнадцяткову систему числення, використовуючи проміжну систему числення (табл.2.3).

3) Перевести число, задане в шістнадцятковій системі числення в вісімкову систему числення, використовуючи проміжну систему числення (табл.2.3).

4) Перевести цілу частину числа, що задане, у 10-у систему числення, використовуючи метод переведення, що не потребує застосування калькулятора (табл.2.3)

4.1) використовуючи схему Горнера для обчислення поліномів;

4.2) використовуючи метод добору доданків, що складаються із цілих ступенів двійки.

Таблиця 2.3 - Варіанти завдань

Вар.№	Число А(10)	Вар.№	Число А(10)	Вар.№	Число А(10)
1	411,587	12	443,683	23	381,683
2	333,682	13	343,692	24	669,328
3	436,615	14	428,318	25	640,495
4	493,653	15	454,512	26	561,629
5	363,670	16	241,197	27	507,146
6	355,693	17	256,407	28	590,765
7	371,662	18	278,653	29	680,778
8	427,652	19	213,386	30	812,342
9	430,628	20	379,931	31	331,491
10	406,646	21	366,277	32	456,642
11	777,893	22	254,768	33	238,965

2.5 Контрольні запитання

1) Які системи числення із кратною основою знайшли найбільше застосування в комп'ютерних системах?
2) Що називається "тріадою"?
3) Що називається "тетрадою"?
4) Як правильно розбити на тріади число, записане у двійковій системі числення?
5) Як правильно розбити на тетради число, записане у двійковій системі числення?
6) Як перевести число, записане у двійковій системі числення, у вісімкову і навпаки, використовуючи тріади?
7) Як перевести число, записане у двійковій системі числення, у шістнадцяткову і навпаки, використовуючи тетради?
8) Як перевести числа з вісімкової в шістнадцяткову і навпаки, використовуючи тріади і тетради?
9) Як швидко перевести двійкове ціле число в десяткову систему числення, використовуючи схему Горнера для обчислення поліномів?
10) Як швидко перевести ціле десяткове число у двійкову систему числення використовуючи метод добору доданків, що складаються із цілих ступенів двійки?

3 ПРЕДСТАВЛЕННЯ ЧИСЕЛ В ПРЯМОМУ, ОБЕРНЕНОМУ І ДОДАТКОВОМУ КОДАХ У ФОРМІ З ФІКСОВАНОЮ КОМОЮ

3.1 Мета заняття

Вивчити форми представлення чисел у ЦЕОМ, методи переведення десяткових чисел у двійкову систему числення та їх представлення у прямому (ПК), оберненому(ОК), додатковому (ДК) кодах з урахуванням знаків чисел у формі з фіксованою комою.

3.2 Короткі основні відомості з теорії та вирішення типових завдань

Машинне представлення числа – це представлення числа в розрядній сітці ЦЕОМ.

Під розрядною сіткою розуміється певна кількість розрядів, виділених для представлення числа, а також розбивка їх на впорядковані групи, для представлення окремих частин числа, таких як знак і порядок числа, знак і мантиса числа.

Машинне представлення числа умовно будемо позначати [A]. Тоді справедливе співвідношення [1,11]:

$$A = [A] \cdot K_A, \tag{3.1}$$

де K_A – масштабний коефіцієнт, величина якого залежить від форми представлення числа в ЦЕОМ.

Під формою представлення числа в ЦЕОМ розуміється множина правил, що дозволяє встановити взаємну відповідність між записом числа і його кількісним еквівалентом.

Дійсні числа в ЦЕОМ можуть бути представлені точно, приблизно або взагалі не можуть бути представлені.

У ЦЕОМ, що оперують із числами, представленими в двійковій системі числення, звичайно використовуються дві форми представлення чисел: з фіксованою комою (крапкою) і із плаваючою комою або в нормальній формі. Для відображення знака числа в розрядній сітці передбачається спеціальний розряд – знаковий. Звичайно «+» кодується «0», а «-» кодується «1».

При представленні чисел з фіксованою комою, кому, звичайно, фіксують або після молодшого розряду (тобто всі числа представляються у вигляді цілих чисел), або перед старшим розрядом (тобто всі числа представляються у вигляді правильних дробів). Тоді в першому випадку розрядна сітка для машинного представлення числа буде виглядати так (див. рис. 3.1):

	$\times 2^{n-1}$	$\times 2^{n-2}$	$\times 2^{n-3}$	...	...	$\times 2^{1}$	$\times 2^{0}$
знак	a_{n-1}	a_{n-2}	a_{n-3}	...	...	a_1	a_0
(n+1) розряд	$\underbrace{\qquad\qquad\qquad}_{\text{Поле числа } (n \text{ – розрядів})}$						

Рис.3.1

А у другому випадку розрядна сітка для машинного представлення числа буде виглядати так (див. рис. 3.2):

	$\times 2^{-1}$	$\times 2^{-2}$	$\times 2^{-3}$	...	...	$\times 2^{-(n-1)}$	$\times 2^{-n}$
знак	a_{-1}	a_{-2}	a_{-3}	...	...	$a_{-(n-1)}$	a_{-n}

(n+1) розряд

Поле числа (n – розрядів)

Рис. 3.2

Наприклад, числа A_1 = +10100111, і A_2 = - 0,11001101 в розрядній сітці ЦЕОМ будуть представлені як:

0	1	0	1	0	0	1	1	1	$[A_1]$ = 010100111,

1	1	1	0	0	1	1	0	1	$[A_2]$ = 1,11001101

Оскільки технічна реалізація операції додавання в ЦЕОМ значно простіше, чим операція віднімання, то операцію віднімання заміняють операцією додавання, використовуючи для представлення від'ємних чисел наступні коди: прямий, обернений, додатковий. Останні два коди ще бувають модифікованими.

3.2.1 Прямий код числа

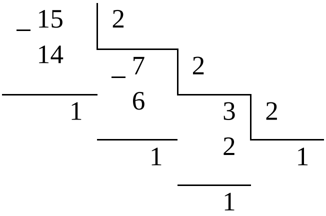
Число А в прямому коді позначається $[A]_{пр}$ або $[A]_{пк.}$ Прямий код двійкового числа збігається по представленню із двійковим записом самого числа, але в знаковому розряді ставиться 0 якщо число додатне та 1, якщо число від'ємне [1…13].

Приклад.

Нехай дано чотири числа

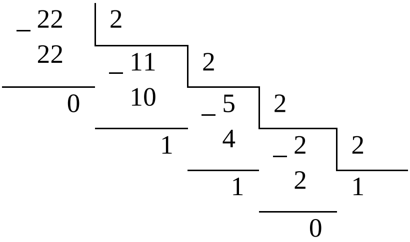
$A_{(10)}$ = +15; $B_{(10)}$ = -22; $C_{(10)}$ = +0,392; $D_{(10)}$ = -0,68

Перевести ці числа в двійкову систему числення і записати їх у прямому коді. Для представлення правильних дробів використовувати 8 розрядів (1 байт) у полі числа.



$A_{(2)} = +1111$

$$[A]_{ПК} = \underbrace{0}_{\text{знак}}\ \underbrace{1111}_{\text{поле числа}}$$



$B_{(2)} = -10110$

$$[B]_{ПК} = \underbrace{1}_{\text{знак}}\ \underbrace{10110}_{\text{поле числа}}$$

$C_{(2)} = +0{,}01100100$

$$[C]_{ПК} = \underbrace{0}_{\text{знак}}\ ,\ \underbrace{01100100}_{\text{поле числа}}$$

0,	392 × 2
0,	784 × 2
1,	568 × 2
1,	136 × 2
0,	272 × 2
0,	544 × 2
1,	088 × 2
0,	176 × 2
0,	352

$D_{(2)} = -\ 0{,}10101110$

$$[D]_{ПК} = \underbrace{1}_{\text{знак}}\ ,\ \underbrace{10101110}_{\text{поле числа}}$$

0,	68 × 2
1,	36 × 2
0,	72 × 2
1,	44 × 2
0,	88 × 2
1,	76 × 2
1,	52 × 2
1,	04 × 2
0,	08

3.2.2 Обернений код

Число А в оберненому коді позначається $[A]_{ОК}$.

Обернений код додатного числа збігається з його прямим кодом, тобто $[A]_{ОК} = [A]_{ПК}$

Обернений код від'ємного числа утворюється за наступним правилом: у знаковий розряд числа записується «1», а в цифрових розрядах усі цифри інвертуються, тобто замість «0» пишуться «1», а замість «1» – «0» [1,7,11].

Тоді числа, що відповідають, A, B, C, D з попереднього прикладу записуються в оберненому коді так:

$[A]_{ОК} = [A]_{ПК} = 01111$,

$[B]_{ОК} = 101001$,

$[C]_{ОК} = [C]_{ПК} = 0{,}01100100$

$[D]_{ОК} = 1{,}\ 01010001$

3.2.3 Додатковий код

Число А в додатковому коді позначається $[A]_{ДК}$.

Додатковий код додатного числа збігається з його прямим кодом, тобто $[A]_{ДК} = [A]_{ПК}$

Додатковий код від'ємного числа отримують за наступним правилом: у знаковому розряді числа записується «1», у всіх цифрових розрядах цифри інвертуються, після чого до молодшого розряду додається одиниця [1,2,11].

Тоді числа, що відповідають, A, B, C, D з попередніх прикладів запишуться в додатковому коді так:

$[A]_{ДК} = [A]_{ПК} = 01111$,

$[B]_{ДК} = 101010$,

$[C]_{ДК} = [C]_{ПК} = 0{,}01100100$

$[D]_{ДК} = 1{,}\ 01010010$

3.3 Індивідуальні завдання

Дані числа A, B, C, D у десятковій системі числення (конкретні значення чисел для кожного студента задаються по варіантах у табл. 3.1).

Необхідно виконати і представити в письмовому вигляді хід наступних операцій.

1) перевести числа у двійкову систему числення;
2) представити отримані двійкові числа в прямому (ПК), оберненому (ОК) і додатковому (ДК) кодах;

Для представлення правильних дробів використовувати 8 розрядів (1 байт) у полі числа.

Таблиця 3.1.

№	A	B	C	D	№	A	B	C	D	№	A	B	C	D
1	125	- 2	0,650	- 0,34	13	235	- 9	0,250	- 0,125	25	364	- 2	0,630	- 0,140
2	234	- 8	0,750	- 0,24	14	254	- 3	0,42	-0,34	26	376	- 8	0,380	- 0,120
3	342	- 7	0,150	- 0,125	15	244	- 6	0,34	-0,240	27	387	- 7	0,530	- 0,255
4	182	-4	0,860	- 0,42	16	342	- 5	0,560	- 0,16	28	396	-4	0,480	- 0,122
5	235	- 2	0,460	- 0,34	17	264	- 2	0,740	- 0,38	29	114	- 9	0,750	- 0,32
6	152	- 8	0,780	- 0,16	18	277	- 8	0,440	- 0,125	30	127	- 3	0,950	- 0,12
7	142	- 7	0,630	- 0,38	19	286	- 7	0,650	- 0,234	31	295	- 6	0,250	- 0,11
8	132	-4	0,380	- 0,125	20	298	-4	0,750	- 0,215	32	125	- 5	0,340	- 0,25
9	122	- 9	0,530	- 0,234	21	324	- 9	0,150	- 0,11	33	234	- 2	0,240	- 0,11
10	113	- 3	0,480	- 0,215	22	332	- 3	0,860	- 0,230	34	342	- 8	0,560	- 0,240
11	214	- 6	0,750	- 0,34	23	345	- 6	0,460	- 0,138	35	182	- 7	0,740	- 0,24
12	225	- 5	0,950	- 0,24	24	355	- 5	0,780	- 0,230					

3.4 Контрольні запитання

1) Що означає «машинне» представлення числа?
2) Що розуміється під розрядною сіткою ЦЕОМ?
3) Що розуміється під формою представлення числа? Які форми машинного представлення чисел вам відомі?
4) Якими символами кодуються додатні та від’ємні числа?
5) Як виглядає розрядна сітка для машинного представлення чисел у формі з фіксованою комою:
   – з фіксацією коми після молодшого розряду,

– з фіксацією коми перед старшим розрядом?

6) Які правила запису чисел:
   - у прямому коді (ПК),
   - у оберненому коді (ОК),
   - у додатковому коді (ДК)?

4 ПРЕДСТАВЛЕННЯ ЧИСЕЛ У ФОРМІ З ПЛАВАЮЧОЮ КОМОЮ. МОДИФІКОВАНІ КОДИ

4.1 Мета заняття

Вивчити форми представлення чисел у ЦЕОМ, методи переведення десяткових чисел у двійкову систему числення та їх представлення у прямому, оберненому, додатковому і модифікованих кодах з урахуванням знаків чисел у формі з плаваючою комою.

4.2 Короткі основні відомості з теорії та вирішення типових завдань

Числа з фіксованою комою мають порівняно вузький діапазон представлення чисел і вимагають попереднього розрахунку масштабного коефіцієнта K_A, як це було зазначено у практичній роботі №3

При представленні чисел у форматі із плаваючою комою числа в ЦЕОМ представляються у вигляді:

$$p^m \cdot a = A = p^m \sum_{i=-1}^{-n} a_i p^{i-m}, \tag{4.1}$$

де a – мантиса (звичайно правильний дріб),

m – порядок.

Формат числа, представленого у формі із плаваючою комою, має вигляд

знак	m_{s-1}	m_{s-2}	...	m_0	знак	a_{-1}	a_{-2}	...	a_{-n}
(s+1)-й розряд	поле порядка (s – розрядів)				(n+1)-й розряд	поле мантиси (n– розрядів)			

Наприклад, двійкове число $A_{(2)}$ = - $1101{,}1010_{(2)}$ в форматі із плаваючою комою може бути представлено як:

$A_{(2)} = -10^{+100} \cdot 0{,}11011010$

або в розрядній сітці ЦЕОМ

0	100	1	11011010
Зн. пор	пор	Зн. числа	мантиса

Числа із плаваючою комою мають набагато ширший діапазон представлення чисел у порівнянні з їхнім представленням у формі з фіксованою комою.

4.2.1 Модифіковані коди

Ці коди відрізняються від звичайних тим, що в них під знак числа виділяється не один, а два розряди, тобто знак додатного числа кодується як «00», а від'ємного – «11» [1, 5, 13].

Так, наприклад, числа С і D у оберненому модифікованому коді запишуться так:

$[C]_{OK}^{M}$=00,01100100,

$[D]_{OK}^{M}$=11,01010001

а в додатковому модифікованому коді:

$$[C]_{ДК}^{M} = 00{,}01100100\,,$$

$$[D]_{ДК}^{M} = 11{,}01010010\,;$$

При додаванні чисел у модифікованих кодах легко виявляється переповнення. Ознакою переповнення є різні цифри в знакових розрядах, тобто «01» або «10», відповідно додатне і від'ємне переповнення.

4.2.2 Нормалізований вигляд представлення чисел з плаваючою комою

Для однозначного представлення чисел з плаваючою комою мантису числа прийнято записувати в т.зв. нормалізованому вигляді.

Нормалізованим вважається число, у якого в старшому значащому розряді мантиси стоїть «1» тобто $a_{-1} = 1$. Якщо число не нормалізоване, то його нормалізують шляхом зсуву мантиси з відповідною зміною порядку числа.

Представити числа $A_{(2)}$, $B_{(2)}$, $C_{(2)}$, $D_{(2)}$ у форматі із плаваючою комою і нормалізованою мантисою в прямому коді.

$A_{(2)} = +1111.$

$$[A] = \underbrace{0}_{\text{зн. пор}}\ \underbrace{100}_{\text{пор}}\ \underbrace{0}_{\text{зн. числа}}\ \underbrace{11110000}_{\text{мантиса}}$$

$A_{(2)} = +10^{+100}\cdot 0{,}11110000$

ФПК s=3, n=8, m=+4

$B_{(2)} = -10110.$

$[B] = 0\ 101\ 1\ 10110000$

$B_{(2)} = -10^{+101}\cdot 0{,}10110000$

ФПК s=3, n=8, m=+5

$C_{(2)} = +0{,}01100100$

$[C] = 1\ 001\ 0\ 11001\ 000$

$C_{(2)} = +10^{-001}\cdot 0{,}11001000$

ФПК s=3, n=8, m=-1

$D_{(2)} = -0{,}10101110$

$$[D] = \underbrace{0}_{\text{зн. пор}}\ \underbrace{000}_{\text{пор}}\ \underbrace{1}_{\text{зн. числа}}\ \underbrace{10101110}_{\text{мантиса}}$$

$D_{(2)} = -10^{+000}\cdot 0{,}10101110$

ФПК s=3, n=8, m=+0

4.3 Індивідуальні завдання

Дані числа A, B, C, D у десятковій системі числення (конкретні значення чисел для кожного студента задаються по варіантах у табл. 4.1).

Необхідно виконати і представити в письмовій формі хід наступних операцій.

1) перевести числа у двійкову систему числення;
2) представити отримані двійкові числа модифікованому додатковому (МДК) коді;
3) представити числа $A_{(2)}$, $B_{(2)}$, $C_{(2)}$, $D_{(2)}$ у форматі із плаваючої комою і нормалізованою мантисою в ПК. Для представлення **правильних дробів** використовувати 8 розрядів (1 байт) у полі числа; для представлення цілих чисел – кількість розрядів не меншу, ніж необхідна для даного числа.
4) Привести до одного порядку числа A та B.

Таблица 4.1.

№	A	B	C	D	№	A	B	C	D	№	A	B	C	D
1	125	- 2	0,650	- 0,34	13	235	- 9	0,250	- 0,125	25	364	- 2	0,630	- 0,140
2	234	- 8	0,750	- 0,24	14	254	- 3	0,42	-0,34	26	376	- 8	0,380	- 0,120
3	342	- 7	0,150	- 0,125	15	244	- 6	0,34	-0,240	27	387	- 7	0,530	- 0,255
4	182	-4	0,860	- 0,42	16	342	- 5	0,560	- 0,16	28	396	-4	0,480	- 0,122
5	235	- 2	0,460	- 0,34	17	264	- 2	0,740	- 0,38	29	114	- 9	0,750	- 0,32
6	152	- 8	0,780	- 0,16	18	277	- 8	0,440	- 0,125	30	127	- 3	0,950	- 0,12
7	142	- 7	0,630	- 0,38	19	286	- 7	0,650	- 0,234	31	295	- 6	0,250	- 0,11
8	132	-4	0,380	- 0,125	20	298	-4	0,750	- 0,215	32	125	- 5	0,340	- 0,25
9	122	- 9	0,530	- 0,234	21	324	- 9	0,150	- 0,11	33	234	- 2	0,240	- 0,11
10	113	- 3	0,480	- 0,215	22	332	- 3	0,860	- 0,230	34	342	- 8	0,560	- 0,240
11	214	- 6	0,750	- 0,34	23	345	- 6	0,460	- 0,138	35	182	- 7	0,740	- 0,24
12	225	- 5	0,950	- 0,24	24	355	- 5	0,780	- 0,230	36	286	- 7	0,650	- 0,234

4.4 Контрольні запитання

1) Що означає «машинне» представлення числа?
2) Що розуміється під розрядною сіткою ЦЕОМ?
3) Що розуміється під формою представлення числа? Які форми машинного представлення чисел вам відомі?
4) Якими символами кодуються додатні та від'ємні числа?
5) Як виглядає розрядна сітка для машинного представлення чисел у формі із плаваючою комою?
6) Яка мантиса називається нормалізованою?
7) Які правила запису чисел:
   - у прямому коді (ПК),
   - у оберненому коді (ОК),
   - у додатковому коді (ДК),
   - у модифікованих ПК, ОК, ДК?

5 ВИКОНАННЯ ОПЕРАЦІЙ АЛГЕБРАЇЧНОГО ДОДАВАННЯ ТА ЗСУВУ В ФОРМІ З ФІКСОВАНОЮ КОМОЮ

5.1 Мета заняття

Вивчити особливості виконання операцій алгебраїчного додавання та зсуву у цифрових ЕОМ. Набути навички додавання двійкових чисел у прямому, оберненому, додатковому кодах з урахуванням знаків чисел у формі з фіксованою комою.

5.2. Короткі основні відомості з теорії та вирішення типових завдань

5.2.1 Алгебраїчне додавання чисел в прямому коді

Можливо два випадки [1,2,7]:

1) доданки мають однакові знаки;
2) доданки мають різні знаки.

Алгоритм одержання суми в першому випадку:

1) додати абсолютні значення чисел;
2) сумі присвоїти знак одного з доданків.

У загальному випадку алгоритм одержання алгебраїчної суми буде наступним:

1) зрівняти знаки доданків, і якщо вони однакові, то виконати додавання по попередньому алгоритму, якщо ні, то виконати п.2.;
2) зрівняти доданки по абсолютній величині;
3) якщо є необхідність, то переставити доданки місцями, щоб відняти від більшого доданку менший;
4) виконати віднімання двох чисел;
5) результату присвоїти знак більшого доданка.

Приклади одержання суми в ПК.

Приклад 1.

$A > 0$; $B > 0$; $|A|+|B|< 1$; $C > 0$

$A_{(2)} = +0{,}01100101$ $\qquad$ $B_{(2)} = +0{,}10001100$

$|A| = 0{,}01100101$ $\qquad\qquad$ $[C]_{ПК} = 0{,}11110001$

$+$

$|B| = 0{,}10001100$ $\qquad\qquad$ $[C]_{ПК} = |A| + |B|$

$C = +\ 0{,}11110001 = [C]_{ПК}$

Приклад 2.

$A > 0;\ B < 0;\ C < 0$

$A_{(2)} = +0{,}01100101$ $\qquad\qquad$ $B_{(2)} = -\ 0{,}10001100$

Оскільки $|B| > |A|$, то $[C]_{ПК} = |B| - |A| + 1$

$|B| = 0{,}10001100$

$-$

$|A| = 0{,}01100101$

$0{,}00100111$

$+$

1

$1{,}00100111 = [C]_{ПК}$

$[C]_{ПК} = \underbrace{1,}_{\text{знак}}00100111$

Приклад 3.

$A < 0;\ B > 0;\ C > 0$

$A_{(2)} = -\ 0{,}01100101$ $\qquad\qquad$ $B_{(2)} = +\ 0{,}10001100$

Оскільки $|B| > |A|$, то $[C]_{ПК} = |B| - |A|$

$|B| = 0{,}10001100$

$-$

$|A| = 0{,}01100101$

$0{,}00100111 = [C]_{ПК}$

$[C]_{ПК} = \underbrace{0,}_{\text{знак}}00100111$

Приклад 4.

$A < 0;\ B < 0;\ C < 0$

$A_{(2)} = -0{,}01100101$ $\qquad\qquad$ $B_{(2)} = -\ 0{,}10001100$

Знаки однакові, $[C]_{ПК} = |A| + |B| + 1$

$|A| = 0{,}01100101$

$+$

$|B| = 0{,}10001100$

$0{,}11110001$

$+$

1

$1{,}11110001 = [C]_{ПК}$

$[C]_{ПК} = \underbrace{0,}_{\text{знак}}00100111$

$[C]_{ПК} = \underbrace{1,}_{\text{знак}}11110001$

Таким чином, у прямому коді знаковий розряд і цифрову частину числа не можна розглядати як єдине ціле, а виконання операції додавання утруднене тим, що крім суматора в машині необхідно мати ще й "пристрій що віднімає". У зв'язку із цими недоліками прямий код для виконання операції алгебраїчного додавання не застосовується, але він зручний при виконанні операцій множення і ділення.

5.2.2 Алгебраїчне додавання чисел в додаткових кодах

У додатковому коді операція віднімання заміняється операцією алгебраїчного додавання. При цьому знаковий розряд і цифрова частина числа розглядається як єдине ціле, у результаті чого з від'ємними числами машина оперує як з неправильними дробами. Правильний знак суми утворюється автоматично в процесі додавання значень знакових розрядів операндів і одиниці переносу із цифрової частини, якщо вона є [1,7,11].

Розглянемо 4 можливих варіанти додавання чисел у додатковому коді.

Приклад 1.

$A_{(2)} = +\,0{,}01100101$	$B_{(2)} = +0{,}10001100$
$A > 0$	$B > 0 \qquad C > 0$
$[A]_{ДК} = 0{,}01100101$	$[C]_{ПК} = [C]_{ДК} = 0{,}11110001$
+	
$[B]_{ДК} = 0{,}10001100$	$[C]_{ДК} = [A]_{ДК} + [B]_{ДК}$
$[C]_{ДК} = 0{,}11110001$	

Приклад 2.

$A_{(2)} = +\,0{,}01100101$	$B_{(2)} = -\,0{,}10001100$
$A > 0$	$B < 0 \quad \lvert B\rvert > \lvert A\rvert \qquad C < 0$
$[A]_{ДК} = 0{,}01100101$	$[C]_{ДК} = [A]_{ДК} + [B]_{ДК}$
+	
$[B]_{ДК} = 1{,}01110100$	$[C]_{ПК} = 1{,}00100111$
$[C]_{ДК} = 1{,}11011001$	(порівняй із прикладом №2 додавання в ПК)

Приклад 3.

$A_{(2)} = -\,0{,}01100101$	$B_{(2)} = +\,0{,}10001100$
$A < 0$	$B > 0 \quad \lvert B\rvert > \lvert A\rvert \quad C > 0$

$[A]_{ДК} = 1{,}10011011$ $\quad$ $[C]_{ДК} = 0{,}00100111 = [C]_{ПК}$

$+$

$[B]_{ДК} = 0{,}10001100$

$10{,}00100111$

не враховується

Приклад 4.

$A_{(2)} = -0{,}01100101$ $\quad$ $B_{(2)} = -0{,}10001100$

$A < 0$ $\quad$ $B < 0$ $\quad$ $C < 0$

$[A]_{ДК} = 1{,}10011011$ $\quad$ $[C]_{ДК} = 1{,}00001111$

$+$

$[B]_{ДК} = 1{,}01110100$ $\quad$ $[C]_{ПК} = 1{,}11110001$

$11{,}00001111$

не враховується

При додаванні в додатковому коді при неправильному виборі масштабних коефіцієнтів доданків у першому і четвертому випадках виконання операції додавання можливе переповнення розрядної сітки. Ознакою переповнення є відмінність знака отриманої суми від знаків доданків.

Приклади таких ситуацій:

1) $A_{(2)} = +0{,}111101$ $\quad$ $B_{(2)} = +0{,}001101$

$[A]_{ДК} = 0{,}111101$

$+$

$[B]_{ДК} = 0{,}001101$

$[C]_{ДК} = 1{,}001010$

знак «-» не відповідає дійсному

2) $A_{(2)} = -0{,}111101$ $\quad$ $B_{(2)} = -0{,}001101$

$[A]_{ДК} = 1{,}000011$

$+$

$[B]_{ДК} = 1{,}110011$

$[C]_{ДК} = \cancel{1}0{,}110110$

не враховується, знак «+» не відповідає дійсному

Для швидкого виявлення переповнення розрядної сітки використовується модифікований додатковий код, у якого під знак виділено 2 розряду, які означають:

00 – знак «+»;

11 – знак «-»;

10 – додатне переповнення;

01 – від’ємне переповнення.

І тоді раніше розглянуті приклади в МДК будуть виглядати так:

$$
\begin{array}{l}
[A]_{ДК}^{M} = 00{,}111101^{+} \\
[B]_{ДК}^{M} = 00{,}001101 \\
\hline
\underbrace{01}_{\text{додатне переповнення}}{,}001010
\end{array}
$$

$$
\begin{array}{l}
[A]_{ДК}^{M} = 11{,}000011 \\
+ \\
[B]_{ДК}^{M} = 11{,}110011 \\
\hline
\underset{\text{не враховується}}{\cancel{1}}\ \underbrace{10}_{\text{від'ємне переповнення}}{,}110110
\end{array}
$$

Таким чином, лівий знаковий розряд завжди зберігає правильний код знака результату.

5.2.3 Алгебраїчне додавання чисел в оберненому коді

Характерною рисою алгебраїчного додавання в оберненому коді є наявність циклічного переносу (якщо він виникає) зі знакового розряду в молодший розряд цифрової частини. Завдяки цьому здійснюється корекція суми на 2^{-n} [1,5,7,11].

Розглянемо чотири характерні приклади.

Приклад 1.

$A_{(2)} = +\ 0{,}01100101$ $\qquad$ $B_{(2)} = +\ 0{,}10001100$

$$
\begin{array}{l}
[A]_{ОК} = 0{,}01100101 \\
+ \\
[B]_{ОК} = 0{,}10001100 \\
\hline
[C]_{ОК} = 0{,}11110001 = [C]_{ПК}
\end{array}
$$

циклічного переносу немає.

Приклад 2.

$A_{(2)} = +\,0{,}01100101$ $\quad$ $B_{(2)} = -\,0{,}10001100$

$[A]_{ОК} = 0{,}01100101$

\+

$[B]_{ОК} = 1{,}01110011$

$[C]_{ОК} = 1{,}11011000$

$[C]_{ПК} = 1{,}00100111$

циклічного переносу немає.

Приклад 3.

$A_{(2)} = -\,0{,}01100101$ $\quad$ $B_{(2)} = +\,0{,}10001100$

$[A]_{ОК} = 1{,}10011010$

\+

$[B]_{ОК} = 0{,}10001100$

$10{,}00100110$

+1 циклічний перенос

$[C]_{ОК} = 0{,}00100111$ $\quad$ $[C]_{ПК} = 0{,}00100111$

Приклад 4.

$A_{(2)} = -\,0{,}01100101$ $\quad$ $B_{(2)} = -\,0{,}10001100$

$[A]_{ОК} = 1{,}10011010$

\+

$[B]_{ОК} = 1{,}01110011$

$11{,}00001101$

+1

$[C]_{ОК} = 1{,}00001110$ $\quad$ $[C]_{ПК} = 1{,}11110001$

Застосування модифікованого оберненого коду дозволяє швидко виявити переповнення розрядної сітки.

Приклади

1) $A_{(2)} = +\,0{,}111101$ $\quad$ $B_{(2)} = +\,0{,}001101$

$[A]_{OK}^{M} = 00{,}111101$

\+

$[B]_{OK}^{M} = 00{,}001101$

$[C]_{OK}^{M} = \underbrace{01}_{}{,}001010$

додатне переповнення

2) $A_{(2)} = -0,111101$ $B_{(2)} = -0,001101$

$$[A]^{M}_{OK} = 11,000010$$
$+$
$$[B]^{M}_{OK} = 11,110010$$

$$110,110100$$
$$\xrightarrow{} +1$$

$$[C]^{M}_{OK} = \underbrace{10,}_{\text{від'ємне переповнення}}110101$$

5.2.4. Операція зсуву в ЦЕОМ

До операції зсуву доводиться звертатися при виконанні додавання в машині із плаваючою комою, а також при виконанні операції множення і ділення в ЦЕОМ обох типів. У всіх цих операціях проводиться зсув мантис, тому будемо розглядати тільки числа з фіксованою комою.

Зсув прямого коду числа ***на k розрядів праворуч*** еквівалентний множенню числа на 2^{-k}. При цьому молодші розряди виходять за межі розрядної сітки машини і втрачаються, що приводить до збільшення погрішності представлення числа [1,4,8,11].

Розряди, що звільнилися, заповнюються нулями.

Наприклад:

$[A]_{ПК} = 1,011010$ $\quad \kappa = 1 \rightarrow$ (зсув праворуч) $2^{-1} \cdot [A]_{ПК} = 1,001101$

$\kappa = 2 \rightarrow$ $2^{-2} \cdot [A]_{ПК} = 1,000110$,

Таким чином, знаковий розряд залишається без зміни.

Зсув прямого коду числа ***на k розрядів ліворуч*** еквівалентний множенню числа на 2^{κ}. Ця операція коректна доти, поки старші значення цифри не почнуть виходити за межі розрядної сітки, тобто поки число по абсолютній величині не стане більше 1. При зсуві ліворуч вивільнювані розряди заповнюються нулями [1,11].

Наприклад, число $[A]_{ПК} = 1,011010$ коректно можна зсунути тільки на один розряд вліво:

$\kappa = 1 \leftarrow$ $\quad 2^{1} \cdot [A]_{ПК} = 1,110100$ (зсув ліворуч), Зсув додатного числа ліворуч або праворуч в ДК або в ОК нічим не відрізняється від зсуву додатного числа в ПК.

Загальним правилом зсуву дробів вправо в інверсному коді (ОК, ДК) є наявність передачі зі знакового розряду в старший цифровий розряд і відновлення знака, тобто вивільнювані розряди заповнюються «1».

Приклад:

$[A]_{ОК} = 1{,}011010$ $\qquad \kappa = 1 \rightarrow \qquad 2^{-1} \cdot [A]_{ОК} = 1{,}101101$

$\qquad \kappa = 2 \rightarrow \qquad 2^{-2} \cdot [A]_{ОК} = 1{,}110110$

При зсуві вліво вивільнювані праворуч розряди в ОК заповнюються «1», а в ДК – «0».

Зсув вліво у випадку інверсних кодів коректний доти, поки в старшому розряді числа не з'явиться «0».

5.3 Індивідуальні завдання

Дані числа **A, B, D** у десятковій системі числення (конкретні значення чисел для кожного студента задаються по варіантах у табл. 5.1).

Таблиця 5.1

№ вар	A	B	D	Код
1	- 2	0,650	- 0,34	ОК
2	- 8	0,750	- 0,24	ОК
3	- 7	0,150	- 0,125	ДК
4	-4	0,860	- 0,42	МДК
5	- 2	0,460	- 0,34	ОК
6	- 8	0,780	- 0,16	ДК
7	- 7	0,630	- 0,38	ДК
8	-4	0,380	- 0,125	МДК
9	- 9	0,530	- 0,234	ОК
10	- 3	0,480	- 0,215	ОК
11	- 6	0,750	- 0,34	ДК
12	- 5	0,950	- 0,24	МДК
13	- 9	0,250	- 0,125	ОК
14	- 3	0,42	-0,34	ДК
15	- 6	0,34	-0,240	ДК
16	- 5	0,560	- 0,16	МДК
17	- 2	0,740	- 0,38	ОК
18	- 8	0,440	- 0,125	ОК
19	- 7	0,650	- 0,234	ДК
20	-4	0,750	- 0,215	МДК
21	- 9	0,150	- 0,11	ОК
22	- 3	0,860	- 0,230	ДК
23	- 6	0,460	- 0,138	ДК
24	- 5	0,780	- 0,230	МДК
25	- 2	0,630	- 0,140	ОК
26	- 8	0,380	- 0,120	ОК
27	- 7	0,530	- 0,255	ДК
28	-4	0,480	- 0,122	ОК
29	- 9	0,750	- 0,32	ДК
30	- 3	0,950	- 0,12	ДК
31	- 6	0,250	- 0,11	ДК
32	- 5	0,340	- 0,25	МДК
33	- 2	0,240	- 0,11	ОК
34	- 8	0,560	- 0,240	ОК
35	- 7	0,740	- 0,24	ДК

Необхідно виконати і представити в письмовій формі хід наступних операцій.

1) Перевести числа в двійкову систему числення.

2) Виконати операцію $\mathbf{B}_{(2)}+\mathbf{D}_{(2)}$ з представленням чисел у форматі з фіксованою комою. Тип коду (ПК, ОК, ДК, МДК) задається по варіантах.

3) При виконанні завдань по п.2 перевірити правильність результату шляхом обчислення десяткових еквівалентів отриманих сум.

Для представлення правильних дробів використовувати 8 розрядів (1 байт) у полі числа.

5.4 Контрольні запитання

1) Викладіть алгоритм додавання двох чисел у прямих кодах для випадку, коли доданки мають однакові знаки.

2) Викладіть алгоритм додавання двох чисел у прямих кодах для випадку, коли доданки мають будь-які знаки.

3) Як виконується алгебраїчне додавання двох чисел у додатковому коді?

4) Що є ознакою переповнення при додаванні чисел у додатковому коді?

5) Як виявляється переповнення розрядної сітки при використанні модифікованого додаткового коду?

6) У чому полягає характерна риса алгебраїчного додавання в оберненому коді?

7) Викладіть правила зсуву числа на *k* розрядів при використанні

- прямого коду і зсуву вправо;
- прямого коду і зсуву вліво;
- інверсних кодів і зсуву вправо;
- інверсних кодів і зсуву вліво.

8) Доки зсув вліво є коректним?

6 ВИКОНАННЯ ОПЕРАЦІЙ АЛГЕБРАЇЧНОГО ДОДАВАННЯ У ФОРМІ З ПЛАВАЮЧОЮ КОМОЮ

6.1 Мета заняття

Вивчити особливості виконання операцій алгебраїчного додавання у формі з плаваючою комою. Придбати навички додавання двійкових чисел у прямому, оберненому, додатковому кодах з урахуванням знаків чисел у формі з плаваючою комою.

6.2 Короткі основні відомості з теорії та вирішення типових завдань

6.2.1 Алгоритм додавання чисел

1) Зрівнюються порядки доданків: менший порядок збільшується до більшого, а мантиса преутвореного числа зсовується вправо (число денормалізується) на відповідне число розрядів.
2) Проводиться перетворення мантис доданків в один з модифікованих кодів.
3) Мантиси доданків додаються за правилами додавання дробових чисел з фіксованою комою.
4) Якщо буде потреба, мантиса суми переводиться в прямий код, проводиться нормалізація суми і округлення її мантиси. Результату приписується загальний порядок.

Залежно від абсолютних величин мантис доданків сума може виявитися: нормалізованою, денормалізованою вправо, денормалізованою вліво.

Неспівпадіння цифр у знакових розрядах (в модифікованому коді) свідчить про денормалізацію вліво (переповнення), а співпадіння цифр знакового і старшого розряду мантиси - про порушення нормалізації вправо.

При денормалізації вліво мантиса результату зсувається на один розряд вправо, а порядок збільшується на «1». При денормалізації вправо мантиса зсувається вліво до появи в старшому розряді 1, при значенні знака 0, або 0, при значенні знака 1, а з порядку віднімається число одиниць, рівне числу зсувів мантиси.

Приклад:

Дано два числа: $A_{(10)} = -6$; $B_{(10)} = 0{,}250$

Потрібно:

1) Перевести числа у двійкову систему числення і представити їх у формі із плаваючою комою в прямому коді.

Для представлення порядку використовувати 3 двійкових розряди, а для представлення мантиси – 8 двійкових розрядів.

2) виконати алгебраїчне додавання чисел $A_{(2)} + B_{(2)} = C_{(2)}$, використовуючи модифікований додатковий код (МДК).

Рішення:

$$\begin{array}{r|l} -6 & 2 \\ 6 & -3 \;|\; 2 \\ \hline 0 & \;\;2 \;|\; 1 \\ & \;\;1 \end{array}$$

$A_{(2)} = -110_{(2)}$ $[A]_{ПК} = 1110$

Переводимо у формат із ПК, для чого представимо числа у вигляді правильного дробу і множника у вигляді порядку у відповідному ступені.

$$[A] = \underbrace{0}_{\text{зн. пор}}\underbrace{011}_{\text{пор}}\underbrace{1}_{\text{зн. числа}}\underbrace{11000000}_{\text{мантиса}}$$

0,	250 * 2
0,	500 * 2
1,	000 * 2
0,	000 * 2
0,	000 * 2
⋮	⋮

$B_{(2)} = +\,0{,}01000000$

$[B]_{ПК} = 0{,}01000000$

В формі з ПК

$$[B]_{ПК} = \underbrace{0}_{\text{зн. пор}}\underbrace{000}_{\text{пор}}\underbrace{0}_{\text{зн. числа}}\underbrace{01000000}_{\text{мантиса}}$$

Представимо числа [A] і [B] в МДК

$$[A]^{M}_{ДК} = 0001111{,}01000000$$

$$[B]^{M}_{ДК} = 0000000{,}01000000$$

Зрівняємо порядки чисел A і B, для чого знайдемо різницю порядків:

$$[m_a]^{M}_{ДК} + [m_b]^{M}_{ДК} = [\Delta m]^{M}_{ДК}$$

$$
\begin{array}{l}
\quad [m_a]^M_{ДК} = 00011 \\
+ \\
\quad [m_b]^M_{ДК} = 00000 \\
\hline
\quad [\Delta m]^M_{ДК} = 00011
\end{array}
\qquad [\Delta m]_{ПК} = 0011 \quad \Delta m = +3
$$

Знак «+» вказує, що порядок А більше порядку В.

Цифра 3 вказує число розрядів, на яке потрібно зсунути вправо мантису числа В, щоб зрівняти порядки.

Отже, приймаємо загальний порядок [m]заг = [ma] = [mb]

І тоді числа А та В, що мають загальний порядок $[m]^M_{заг.ДК} = 00011$, запишуться так:

$$[A]^M_{ДК} = 00011\overbrace{11{,}01000000}^{[a]^M_{ДК}}$$

$$[B']^M_{ДК} = 00011\underbrace{00{,}00001000}_{[b']^M_{ДК}}$$

Виконаємо додавання мантис у МДК і одержимо мантису суми:

$$
\begin{array}{l}
\quad [a]^M_{ДК} = 11{,}01000000 \\
+ \\
\quad [b']^M_{ДК} = 00{,}00001000 \\
\hline
\quad [c]^M_{ДК} = 11{,}01001000
\end{array}
$$

Мантиса суми нормалізована

Остаточний результат в МДК:

$$[C]^M_{ДК} = 0001111{,}01001000$$

Переведемо в прямий код:

$$[C]_{ПК} = 00111{,}10111000$$

Порядок суми $[m_c]_{ПК} = 0011$ $m_c=+3$

Мантиса суми:

$$[c]_{ПК} = 1{,}10111000$$

$$c_{(10)} = -\left(\frac{1}{2}+0+\frac{1}{8}+\frac{1}{16}+\frac{1}{32}\right) = -\frac{16+4+2+1}{32} = \frac{23}{32} = \frac{-23}{32} = -0{,}71875$$

$$C_{(10)} = 2^3\cdot(-0{,}71875) = -5.75$$

Задача для рішення в аудиторії.

Дані числа **A, B, D** в десятковій системі числення :

$$A_{(10)} = -6;\ B_{(10)} = +0{,}73;\ D_{(10)} = -0{,}26$$

Необхідно виконати наступне:

1) перевести числа в двійкову систему числення;

2) виконати операцію $\mathbf{B_{(2)}}+\mathbf{D_{(2)}}$ з представленням чисел у форматі з фіксованою комою. Тип коду: ПК, ОК, ДК, МДК;

3) виконати операцію $\mathbf{A_{(2)}}+\mathbf{B_{(2)}}$ з представленням чисел у форматі із плаваючою комою. Тип коду: МОК, МДК;

4) при виконанні завдань по п.2 і п.3 перевірити правильність результату шляхом обчислення десяткових еквівалентів отриманих сум.

Для представлення порядку числа використовувати 3 розряди в полі порядку.

Для представлення правильних дробів використовувати 8 розрядів (1 байт) у полі числа.

Рішення

$$A_{(10)} = -6;\ B_{(10)} = +0{,}73;\ D_{(10)} = -0{,}26$$

1) Переводимо A, B, D в двійкову систему

$$\begin{array}{r|l}
_6 & 2 \\ \cline{2-2}
6 & \begin{array}{r|l} _3 & 2 \\ \cline{2-2} 2 & 1 \\ \hline 1 & \end{array} \\ \hline
0 &
\end{array}$$

$$A_{(2)} = -110_{(2)}$$

0,	73 * 2
1,	46 * 2
0,	92 * 2
1,	84 * 2
1,	68 * 2
1,	36 * 2
0,	72 * 2
1,	44 * 2
0,	88
⋮	⋮

$B_{(2)} = +\,0{,}10111010$

$D_{(2)} = -\,0{,}01000010$

$[B]_{ПК} = 0{,}10111010$

$[D]_{ПК} = 1{,}01000010$

0,	26 $\times 2$
0,	52 $\times 2$
1,	04 $\times 2$
0,	08 $\times 2$
0,	16 $\times 2$
0,	32 $\times 2$
0,	64 $\times 2$
1,	28 $\times 2$
0,	56
⋮	⋮

2) Додаємо $B_{(2)} + D_{(2)}$ в прямому коді

$B > 0,\ D < 0,\ |B| > |D|$

$$
\begin{array}{l}
\ \ |B| = 0{,}10111010 \\
- \\
\ \ |D| = 0{,}01000010 \\
\hline
\ \ |C| = 0{,}01111000
\end{array}
$$

$[C]_{ПК} = |B| - |D| \qquad C > 0$

$[C]_{ПК} = 0{,}01111000$

$C(2) = +\,0{,}01111000$

Находимо десятковий еквівалент суми:

$$C(10) = 0\cdot 2^{-1} + 1\cdot 2^{-2} + 1\cdot 2^{-3} + 1\cdot 2^{-4} + 1\cdot 2^{-5} + 0\cdot 2^{-6} + 0\cdot 2^{-7} + 0\cdot 2^{-8}$$

$$= \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} = \frac{8+4+2+1}{32} = \frac{15}{32} = 0{,}46875_{(10)} \approx 0{,}47$$

Очікуваний результат: $C_{очік.} = 0{,}73 - 0{,}26 = 0{,}47$.

3) Додаємо в ОК

$[C]_{ОК} = [B]_{ОК} + [D]_{ОК}$

$$
\begin{array}{l}
\quad [B]_{ОК} = 0{,}10111010 \\
+ \\
\quad [D]_{ОК} = 1{,}10111101 \\
\hline
\qquad 10{,}01110111 \\
\qquad\quad \longrightarrow +1 \text{ циклічний перенос} \\
\hline
[C]_{ОК} = 0{,}01111000 \qquad = [C]_{ПК}
\end{array}
$$

4) Додаємо в ДК

$[C]_{ДК} = [B]_{ДК} + [D]_{ДК}$

$$
\begin{array}{l}
\quad [B]_{ДК} = 0{,}10111010 \\
+ \\
\quad [D]_{ДК} = 1{,}10111101 \\
\hline
[C]_{ДК} = 1\ 0{,}01111000 \qquad = [C]_{ПК}
\end{array}
$$

$C_{(10)} = + 0{,}47$

$[B]_{ДК} = [B]_{ПР}$

$[D]_{ДК} = [D]_{ОК} + 1*2^{-8}$

$$
\begin{array}{r}
[D]_{ДК} = 1{,}10111101 \\
+ \qquad\qquad\qquad\quad 1 \\
\hline
1{,}10111110
\end{array}
$$

5) Додаємо в МДК

$$[C]_{ДК}^{M} = [B]_{ДК}^{M} + [D]_{ДК}^{M}$$

$$
\begin{array}{l}
\quad [B]_{ДК}^{M} = 00{,}10111010 \\
+ \\
\quad [D]_{ДК}^{M} = 11{,}10111110 \\
\hline
[C]_{ДК}^{M} = 100{,}01111000 = [C]_{ПК}^{M}
\end{array}
$$

$C_{(2)} = + 0{,}01111000$ $\qquad$ $C_{(10)} = + 0{,}47$

Додаємо числа $A_{(2)} + B_{(2)}$ із представленням чисел в форматі з плаваючою комою

1) Тип коду – МДК

$A_{(2)} = -110_{(2)}$ $\qquad$ $B_{(2)} = + 0{,}10111010$

Представимо числа $A_{(2)}$ и $B_{(2)}$ в форматі з плаваючою комою, для чого представимо числа у вигляді правильного дробу і множника у вигляді порядку у відповідному ступені $A_{(2)} = -(10)^{+011} \cdot 0{,}11000000 = P^{m_a} \cdot a$, де $m_a = +011$ – порядок

$a = -0{,}11000000$ – нормалізована мантиса

$$[A]_{ПК} = \underbrace{\underbrace{0}_{\text{зн. пор}}\underbrace{011}_{\text{пор}}}_{[m_a]_{ПК}}\ \underbrace{\underbrace{1}_{\text{зн. числа}},\underbrace{11000000}_{\text{мантиса}}}_{[a]_{ПК}}$$

Представимо число А в МДК в форматі з плаваючою комою

$$[A]^{M}_{ДК} = \underbrace{00011}_{[m_a]^{M}_{ДК}}\underbrace{11{,}01000000}_{[a]^{M}_{ДК}}$$

Число $B_{(2)}$ в форматі з плаваючою комою

$$m_b = +000 \qquad b = +0{,}10111010$$

Прямий код числа В

$$[B]_{ПК} = \underbrace{0000}_{[m_b]_{ПР}}\underbrace{0{,}10111010}_{[b]_{ПР}}$$

МДК числа В

$$[B]^{M}_{ДК} = \underbrace{00000}_{[m_b]^{M}_{ДК}}\underbrace{00{,}10111010}_{[b]^{M}_{ДК}}$$

Зрівняємо порядки чисел А і В, для чого знайдемо різницю порядків:

$$[m_a]^{M}_{ДК} + [m_b]^{M}_{ДК} = [\Delta m]^{M}_{ДК}$$

$$[m_a]^{M}_{ДК} = 00011$$

$+$

$$[m_b]^{M}_{ДК} = 00000$$

$$[\Delta m]^{M}_{ДК} = 00011 \qquad [\Delta m]_{ПК} = 0011 \quad \Delta m_{(10)} = +3$$

Знак «+» вказує на те, що порядок А > порядку В.

Цифра 3 вказує число розрядів, на яке потрібно зсунути вправо мантису числа В, щоб зрівняти порядки. Отже, ухвалюємо загальний порядок

$[m]_{общ}=[m_a]=[m_c]$. І тоді числа А та В, що мають спільний порядок, $[m]^M_{общ.ДК} = 00011$, запишуться так:

$$[A]^M_{ДК} = 00011\overbrace{11{,}01000000}^{[a]^M_{ДК}}$$

$$[B]^M_{ДК} = 00011\underbrace{00{,}00010111}_{[b']^M_{ДК}}$$

Виконаємо додавання мантис у МДК і одержимо мантису суми:

$$[m_a]^M_{ДК} + [m_b]^M_{ДК} = [\Delta m]^M_{ДК}$$

$$\begin{array}{l} \quad [a]^M_{ДК} = 11{,}01000000 \\ + \\ \quad [b']^M_{ДК} = 00{,}00010111 \\ \hline \quad [c]^M_{ДК} = 11{,}01010111 \end{array}$$

$$\begin{array}{r} [c]^M_{ПК} = 11{,}10101000 \\ + \\ 1 \\ \hline 11,\ 10101001 \end{array}$$ - мантиса суми в ПК(нормалізована)

Додатковий код суми у форматі із плаваючою комою:

$[C]^M_{ДК} = 0001111{,}01010111$

Прямий код суми в форматі з плаваючою комою:

$[C]^M_{ПК} = 0001111{,}10101001$

Запис суми в двійковому коді в форматі з плаваючою комою (природна форма)

$C_{(2)} = -(10)^{+011} \cdot 0{,}10101001 = P^{m_C} \cdot c$

Знайдемо десятковий еквівалент суми

$C_{(10)} = -\,2^{+3}\cdot(1\cdot2^{-1} + 0\cdot2^{-2} + 1\cdot2^{-3} + 0\cdot2^{-4} + 1\cdot2^{-5} + 0\cdot2^{-6} + 0\cdot2^{-7} + 1\cdot2^{-8}) = -\,8\ \cdot$

$$\left(\frac{1}{2}+\frac{1}{8}+\frac{1}{32}+\frac{1}{256}\right) = -8\cdot\frac{128+32+8+1}{256} = -8\cdot\frac{169}{256} = -8\cdot0{,}661 = -5{,}28$$

Очікуваний результат: $C_{(10)} = -\,6 + 0{,}73 \underline{= -\,5{,}27}$

2) Додавання в МОК в форматі з плаваючою комою

Процедура додавання чисел A і B в МОК буде відрізнятися тільки процедурою додавання мантис

$$[A]^M_{OK} = 00011\overbrace{11{,}00111111}^{[a]^M_{OK}}$$

$$[B]^M_{OK} = \underbrace{00011}_{[m_{заг.}]^M_{ЗК}}\underbrace{00{,}00010111}_{[b']^M_{ЗК}}$$

$$\begin{array}{r} [a]^M_{ЗК} = 11{,}00111111 \\ + \\ [b']^M_{ЗК} = 00{,}00010111 \\ \hline [c]^M_{ЗК} = 11{,}01010110 \end{array}$$

$$[C]^M_{ЗК} = 0001111{,}01010110$$

$$[C]^M_{ПК} = 0001111{,}10101001$$

$C_{(2)} = -\,(10)^{+011}\cdot0{,}10101001$

$C_{(10)} = -\,2^{+3}\cdot(1\cdot2^{-1} + 0\cdot2^{-2} + 1\cdot2^{-3} + 0\cdot2^{-4} + 1\cdot2^{-5} + 0\cdot2^{-6} + 0\cdot2^{-7} + 1\cdot2^{-8}) =$

$$= -8\cdot\frac{169}{256} = -8\cdot0{,}661 = -5{,}28$$

Очікуваний результат: $C_{(10)} = -\,5{,}27$

6.3 Індивідуальні завдання

Дані числа **A, B, D** у десятковій системі числення (конкретні значення чисел для кожного студента задаються по варіантах у табл. 6.1).

Необхідно виконати і представити в письмовій формі хід наступних операцій.

1) Перевести числа в двійкову систему числення.

2) Виконати операцію $\mathbf{A}_{(2)}+\mathbf{B}_{(2)}$ з представленням чисел у форматі із плаваючою комою. Тип коду (МОК, МДК) задається по варіантах.

3) При виконанні завдань по п.2 перевірити правильність результату шляхом обчислення десяткових еквівалентів отриманих сум.

Для представлення порядку числа використовувати 3 розряди в полі порядку. Для представлення правильних дробів використовувати 8 розрядів (1 байт) у полі числа.

Таблиця 6.1

№ вар	A	B	D	Код		№ вар	A	B	D	Код
1	- 2	0,650	- 0,34	МДК		18	- 8	0,440	- 0,125	МДК
2	- 8	0,750	- 0,24	МОК		19	- 7	0,650	- 0,234	МОК
3	- 7	0,150	- 0,125	МОК		20	-4	0,750	- 0,215	МОК
4	-4	0,860	- 0,42	МДК		21	- 9	0,150	- 0,11	МДК
5	- 2	0,460	- 0,34	МДК		22	- 3	0,860	- 0,230	МДК
6	- 8	0,780	- 0,16	МОК		23	- 6	0,460	- 0,138	МОК
7	- 7	0,630	- 0,38	МОК		24	- 5	0,780	- 0,230	МОК
8	-4	0,380	- 0,125	МДК		25	- 2	0,630	- 0,140	МДК
9	- 9	0,530	- 0,234	МОК		26	- 8	0,380	- 0,120	МДК
10	- 3	0,480	- 0,215	МДК		27	- 7	0,530	- 0,255	МОК
11	- 6	0,750	- 0,34	МОК		28	-4	0,480	- 0,122	МДК
12	- 5	0,950	- 0,24	МДК		29	- 9	0,750	- 0,32	МОК
13	- 9	0,250	- 0,125	МДК		30	- 3	0,950	- 0,12	МДК
14	- 3	0,42	-0,34	МОК		31	- 6	0,250	- 0,11	МОК
15	- 6	0,34	-0,240	МОК		32	- 5	0,340	- 0,25	МОК
16	- 5	0,560	- 0,16	МОК		33	- 2	0,240	- 0,11	МДК
17	- 2	0,740	- 0,38	МДК		34	- 8	0,560	- 0,240	МДК
						35	- 7	0,740	- 0,24	МОК

6.4 Контрольні запитання

1) Викладіть алгоритм додавання двох чисел у прямих кодах для випадку, коли доданки мають однакові знаки.

2) Викладіть алгоритм додавання двох чисел у прямих кодах для випадку, коли доданки мають будь-які знаки.

3) Як виконується алгебраїчне додавання двох чисел у додатковому коді?

4) Що є ознакою переповнення при додаванні чисел у додатковому коді?

5) Як виявляється переповнення розрядної сітки при використанні модифікованого додаткового коду?

6) У чому полягає характерна риса алгебраїчного додавання у оберненому коді?

7) Викладіть алгоритм додавання чисел, представлених у формі із плаваючою комою.

8) У яких випадках мантиса суми вважається

- нормалізованою,
- ненормалізованою вправо,
- денормалізованою вліво.

9) Які правила нормалізації денормалізованих сум?

7 ВИКОНАННЯ ОПЕРАЦІЇ МНОЖЕННЯ В ЦЕОМ

7.1 Мета заняття

Вивчення основних алгоритмів множення чисел у цифрових ЕОМ. Придбання практичних навичок виконання операції множення.

7.2 Короткі основні відомості з теорії та вирішення типових завдань

Існують чотири основні методи машинного множення в прямому коді [1, 3, 5, 11]:

1) множення молодшими розрядами множника зі зсувом суми часткових добутків, що накопичується, вправо;
2) множення молодшими розрядами множника зі зсувом множеного вліво;
3) множення старшими розрядами множника зі зсувом суми часткових добутків, що накопичується, уліво;
4) множення старшими розрядами множника зі зсувом множеного вправо.

У всіх чотирьох методах спочатку визначається знак добутку шляхом додавання "по модулю 2" знакових розрядів співмножників.

У всіх чотирьох методах при множенні двох чисел кількість значущих цифр добутку може в межі досягатися подвійної кількості цифр співмножників. Однак у наслідку остаточний результат округляється з необхідною точністю.

7.2.1 1-й метод множення двох чисел

Нехай дано два числа:

$$A=\sum_{i=-1}^{-n} a_{-i}2^{i}=a_1 2^{-1}+a_2 2^{-2}+a_3 2^{-3}+\ldots+a_{n-1}2^{-(n-1)}+a_n 2^{-n}$$

Таким чином, код числа $A = 0,a_1a_2a_3\ldots a_{n-1}a_n$ - правильний дріб (множене).

$$B=\sum_{i=-1}^{-n} b_{-i}2^{i}=b_1 2^{-1}+b_2 2^{-2}+b_3 2^{-3}+\ldots+b_{n-1}2^{-(n-1)}+b_n 2^{-n}$$

Таким чином, код числа $B = 0,b_1b_2b_3\ldots b_{n-1}b_n$ - правильний дріб (множник).

Тоді результат множення С=А·В можливо представити наступним чином:

$$C=A\bullet B=A\sum_{i=-1}^{-n} b_{-i}2^{i}=Ab_1 2^{-1}+Ab_2 2^{-2}+Ab_3 2^{-3}+\ldots+Ab_{n-1}2^{-(n-1)}+Ab_n 2^{-n} \quad (7.1)$$

Вираз (7.1) стосовно до першого варіанту схеми множення можна представити у вигляді схеми Горнера для обчислення поліномів:

$$C=(((\ldots((0+Ab_n)2^{-1}+Ab_{n-1})2^{-1}+\ldots+Ab_{n-i})2^{-1}+\ldots+Ab_2)2^{-1}+Ab_1)2^{-1} \quad (7.2)$$

Яке може бути зведене до n-кратного виконання циклу:

$$C_{i+1}=(C_i+Ab_{n-i})2^{-1}, \quad (7.3)$$

при початкових значеннях i=0; $C_0=0$.

$$\begin{array}{ll} i=0 & C_1=(0+Ab_n)2^{-1}; \\ i=1 & C_2=(C_1+Ab_{n-1})2^{-1}; \\ i=2 & C_3=(C_2+Ab_{n-2})2^{-1} \\ \ldots & \ldots \\ i=n-1 & C_n=(C_{n-1}+Ab_1)2^{-1} \end{array} \quad (7.4)$$

З виразів (7.2 … 7.4) складається алгоритм одержання результату по першому методу:

1) вміст суматора обнуляється,
2) множене множиться на черговий розряд множника,
3) результат підсумується із умістом суматора,
4) вміст суматора зсувається на 1 розряд вправо,
5) пункти 2, 3, 4 повторюються (n-1) раз.

Структурна схема множення по першому варіанту має такий вигляд:

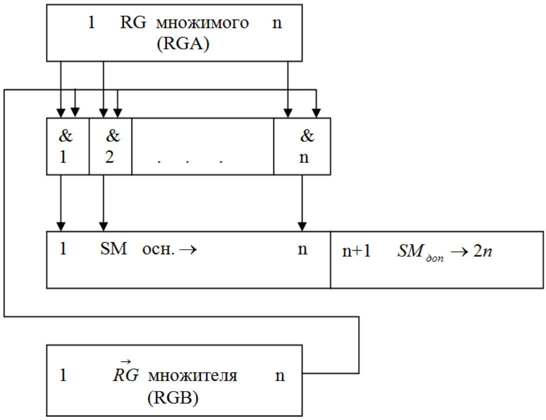


Рисунок 7.1

Приклад:

Дані 2 числа: $A_{(10)} = +0.4$ и $B_{(10)} = -0.8$.

Потрібно знайти добуток, $C_{(10)} = A_{(2)} * B_{(2)}$ використовуючи прямий код і 4 двійкових розряди для представлення правильних дробів. Множення виконати по першому алгоритму.

Рішення.

Переводимо $A_{(10)}$ и $B_{(10)}$ в двійкову систему числення

0,	4 * 2
0,	8 * 2
1,	6 * 2
1,	2 * 2
0,	4

$$A_{(2)} = +0{,}0110$$

$$[A]_{пр} = \underset{зн\, a_1 a_2 a_3 a_4}{0{,}0110}$$

0,	8 * 2
1,	6 * 2
1,	2 * 2
0,	4 * 2
0,	8

$$B_{(2)} = -0{,}1100$$

$$[B]_{пр} = \underset{зн\, b_1 b_2 b_3 b_4}{1{,}1100}$$

n=4, 2n=8, таким чином, потрібен 8-ми розрядний суматор + 1 знаковий розряд.

Знак добутку $S_gC=S_gA \oplus S_gB= 0 \oplus 1 = 1$- число від’ємне.

Наступні дії по алгоритму зручно представити в вигляді таблиці 7.1.

В табл.7.1 виконується позначення: $C'_{i+1} = C_i + Ab_{n-i}$

Таблиця 7.1.

Номер шага, і	розряди множника, b_{4-i}	Дія	Розряди суматора і RGA· b4-i								Примітка
			1	2	3	4	5	6	7	8	
0		Онулення SM	0	0	0	0	0	0	0	0	$C_0 = 0$
	$b_4 = 0$	Ab_4	0	0	0	0					
		C'_1	0	0	0	0	0	0	0	0	
		$C_1 = C'_1 2^{-1}$	0	0	0	0	0	0	0	0	1 частковий добуток.
1	$b_3 = 0$	Ab_3	0	0	0	0					
		C'_2	0	0	0	0	0	0	0	0	
		$C_2 = C'_2 2^{-1}$	0	0	0	0	0	0	0	0	2 частковий добуток.
2	$b_2 = 1$	Ab_2	0	1	1	0					
		C'_3	0	1	1	0	0	0	0	0	
		$C_3 = C'_3 2^{-1}$	0	0	1	1	0	0	0	0	3 частковий добуток.
3	$b_1 = 1$	Ab_1	0	1	1	0					
		C'_4	1	0	0	1	0	0	0	0	
		$C_4 = C'_4 2^{-1}$	0	1	0	0	1	0	0	0	остаточний результат

Остаточно $[C]_{ПР}$ = $[C_4]$ = *1,01001000*. Очікуваний результат $C_{(10)}=(+0,4)\cdot(-0,8)=-0,32$.

$$C_{(2)} = -0,01001000$$

$$C_{(10)} = -(0+\frac{1}{4}+0+0+\frac{1}{32}+0+\ldots+0) = -\frac{8+1}{32} = \frac{-9}{32} = 0,28125$$

$$\approx C_{(2)} = -0,0101 \approx C_{(10)} = -(0+\frac{1}{4}+0+\frac{1}{16}) = \frac{-5}{16} = 0,3125 - \textit{округлённый\_результат}.$$

7.2.2 2-й метод множення двох чисел

При множенні по другому методу вираз (7.1) представляється в наступному вигляді:

$$C = 2^{-n}(A \cdot b_n + 2A \cdot b_{n-1} + ... + 2^i \cdot A \cdot b_{n-i} + ... + 2^{n-1} \cdot A \cdot b_1) \qquad (7.5)$$

Обчислення виразу (7.5) зводиться до n-кратного виконання циклу:

$$C_{i+1} = C_i + A_i \cdot b_{n-i}, \qquad (7.6)$$

де $A_i = 2*A_{i-1}$ при початкових значеннях

$i = 0$, $C_0 = 0$, $A_0 = A$.

Для реалізації даного способу множення потрібно мати n-розрядний регістр множника, що зсувається, 2 n-розрядний регістр множеного, що зсувається, і 2 n-розрядний суматор.

Знайдемо добуток $C_{(2)} = A_{(2)} * B_{(2)}$ по другому алгоритму для тих же чисел, що і у попередньому прикладі.

$$A_{(2)} = +0{,}0110 \qquad B_{(2)} = -0{,}1100$$

$$[A]_{пр} = 0{,}0110 \qquad [B]_{пр} = 1{,}1100$$

Знак добутку $S_gC=S_gA \oplus S_gB = 0 \oplus 1 = 1$- число від'ємне. Результати наведені в табл. 7.2

Таблиця 7.2

Номер шага, i	розряди множника, b_{4-i}	Дії	Розряди суматора і RGA								Примітка
			1	2	3	4	5	6	7	8	
0		Онулення SM	0	0	0	0	0	0	0	0	$C_0 = 0$
		$A_0 = A$	0	0	0	0	0	1	1	0	
	$b_4 = 0$	$A_0 \cdot b_4$	0	0	0	0	0	0	0	0	
		$C_1 = C_0 + A_0 \cdot b_4$	0	0	0	0	0	0	0	0	1 частковий добуток
1		$A_1 = 2A_0$	0	0	0	0	1	1	0	0	Зсув ліворуч
	$b_3 = 0$	$A_1 \cdot b_3$	0	0	0	0	0	0	0	0	
		$C_2 = C_1 + A_1 \cdot b_3$	0	0	0	0	0	0	0	0	2 частковий добуток
2		$A_2 = 2A_1$	0	0	0	1	1	0	0	0	Зсув ліворуч
	$b_2 = 1$	$A_2 \cdot b_2$	0	0	0	1	1	0	0	0	

		$C_3 = C_2 + A_2 \cdot b_2$	0	0	0	1	1	0	0	0	3 частковий добуток
3		$A_3 = 2A_2$	0	0	1	1	0	0	0	0	Зсув ліворуч
	$b_1 = 1$	$A_3 \cdot b_1$	0	0	1	1	0	0	0	0	
		$C_4 = C_3 + A_3 \cdot b_1$	0	1	0	0	1	0	0	0	остаточний результат

$$[C]_{пр} = [C_4] = 1{,}01001000$$

$$C_{(2)} = -0{,}01001000$$

$$C_{(2)} = -0{,}0101 \qquad C_{(10)} \approx 0{,}3125$$

7.2.3 3-й метод множення двох чисел

При множенні по третьому методу вираз (7.1) представляється в наступному вигляді:

$$C = A \cdot B = 2^{-n} \cdot ((...(...((0 + A \cdot b_1) \cdot 2 + A \cdot b_2) \cdot 2 + ... + A \cdot b_i) \cdot 2 + ... + A \cdot b_{n-1}) \cdot 2 + A \cdot b_n) \qquad (7.7)$$

Обчислення виразу (7.7) зводиться до n-кратного повторення циклу:

$$C_{i+1} = (C_i + A \cdot b_{i+1}) \cdot 2 = C'_{i+1} \cdot 2 \qquad (7.8)$$

З початковими значеннями $i = 0$, $C_0 = 0$.

Для реалізації цього варіанту схеми множення необхідні: n-розрядний регістр множника, що зсувається вліво, n-розрядний регістр множеного та 2 n-розрядний суматор зі зсувом уліво.

Знайдемо добуток $C_{(2)} = A_{(2)} * B_{(2)}$ по третьому алгоритму для тих же чисел, що й у попередньому прикладі.

$$A_{(2)} = +0{,}0110 \qquad B_{(2)} = -0{,}1100$$

$$[A]_{пр} = 0{,}0110 \qquad [B]_{пр} = 1{,}1100$$

Знак добутку $S_gC = S_gA \oplus S_gB = 0 \oplus 1 = 1$ - число від'ємне. Результати наведені в табл. 7.3.

Таблиця 7.3

Номер шага, і	розряди множника, b_{i+1}	Дії	Розряди суматора і RG A								Примітка
			1	2	3	4	5	6	7	8	
0		Онулення SM	0	0	0	0	0	0	0	0	$C_0 = 0$
	$b_1 = 1$	$A \cdot b_1$					0	1	1	0	
		$C_1^{'}$	0	0	0	0	0	1	1	0	
		$C_1 = C_1^{'} \cdot 2$	0	0	0	0	1	1	0	0	Зсув ліворуч 1 частковий добуток.
1	$b_2 = 1$	$A_1 \cdot b_2$					0	1	1	0	
		$C_2^{'}$	0	0	0	1	0	0	1	0	
		$C_2^{'} \cdot 2$	0	0	1	0	0	1	0	0	Зсув ліворуч 2 частковий добуток.
2	$b_3 = 0$	$A \cdot b_3$					0	0	0	0	
		$C_3^{'}$	0	0	1	0	0	1	0	0	
		$C_3^{'} \cdot 2$	0	1	0	0	1	0	0	0	Зсув ліворуч 3 частковий добуток.
3	$b_4 = 0$	$A \cdot b_4$					0	0	0	0	
		$C_4^{'}$	0	1	0	0	1	0	0	0	
		C_4	0	1	0	0	1	0	0	0	Остаточний результат

$[C]_{np} = [C_4] = 1{,}01001000$

$C_{(2)} = -0{,}01001000$

$C_{(2)} = -0{,}0101$ $\qquad$ $C_{(10)} \approx 0{,}3125$

7.2.4 4-й метод множення двох чисел

При множенні по 4-му методу вираз (7.1) представляється в наступному вигляді:

$$C = A \cdot B = A \cdot 2^{-1} \cdot b_1 + A \cdot 2^{-2} \cdot b_2 + \ldots + A \cdot 2^{-i} b_i + \ldots + A \cdot 2^{-(n-1)} \cdot b_{n-1} + A \cdot 2^{-n} \cdot b_n$$

Таким чином, так, як воно було записане спочатку.

Обчислення виразу (7.1) може бути зведене до n-кратного виконання циклу:

$$A_{i+1} = A_i \cdot 2^{-1} \tag{7.9}$$
$$C_{i+1} = C_i + A_{i+1} \cdot b_{i+1} = C_i + C'_{i+1}$$

При початкових значеннях $i=0$; $A_0 = A$, $C_0 = 0$.

Тобто, у кожному циклі множене зсувається на 1 розряд вправо і залежно від значення старших розрядів множника або передається в SM, або ні.

Для реалізації методу необхідно мати: n-розрядний регістр множника зі зсувом праворуч і 2 n-розрядний суматор [1,7].

Знайдемо добуток $C_{(2)} = A_{(2)} * B_{(2)}$ по четвертому алгоритму для тих же чисел, що й у попередньому прикладі.

$$A_{(2)} = +0{,}0110 \qquad B_{(2)} = -0{,}1100$$
$$[A]_{пр} = 0{,}0110 \qquad [B]_{пр} = 1{,}1100$$

Знак добутку $S_gC=S_gA \oplus S_gB= 0 \oplus 1 = 1$ - число від'ємне. Результати наведені в табл. 7.4.

Таблиця 7.4.

Номер шага, i	розряди множника, b_{i+1}	Дія	Розряди суматора і RGA								Примітка
			1	2	3	4	5	6	7	8	
0		Онулення SM	0	0	0	0	0	0	0	0	$C_0 = 0$
		$A_0 = A$	0	1	1	0	0	0	0	0	
		$A_1 = A_0 \cdot 2^{-1}$	0	0	1	1	0	0	0	0	$\to A$
	$b_1 = 1$	$C'_1 = A_1 \cdot b_1$	0	0	1	1	0	0	0	0	
		$C_1 = C_0 + C'_1$	0	0	1	1	0	0	0	0	1 частковий добуток
1		$A_2 = A_1 \cdot 2^{-1}$	0	0	0	1	1	0	0	0	$\to A$
	$b_2 = 1$	$C'_2 = A_2 \cdot b_2$	0	0	0	1	1	0	0	0	
		$C_2 = C_1 + C'_2$	0	1	0	0	1	0	0	0	2 частковий добуток
2		$A_3 = A_2 \cdot 2^{-1}$	0	0	0	0	1	1	0	0	$\to A$
	$b_3 = 0$	$C'_3 = A_3 \cdot b_3$	0	0	0	0	0	0	0	0	
		$C_3 = C_2 + C'_3$	0	1	0	0	1	0	0	0	3 частковий добуток
3		$A_4 = A_3 \cdot 2^{-1}$	0	0	0	0	0	1	1	0	$\to A$
	$b_4 = 0$	$C'_4 = A_4 \cdot b_4$	0	0	0	0	0	0	0	0	
		$C_4 = C_3 + C'_4$	0	1	0	0	1	0	0	0	4-й частковий добуток - остаточний результат

$$[C]_{пр} = [C_4] = 1{,}01001000$$

$$C_{(2)} = -0{,}01001000$$

$$C_{(2)} = -0{,}0101 \qquad C_{(10)} \approx 0{,}3125$$

7.3 Індивідуальні завдання

Дані числа **А, В** у десятковій системі числення (конкретні значення чисел для кожного студента задаються по варіантах у табл. 7.5).

Необхідно виконати і представити в письмовій формі хід наступних операцій.

1) Перевести числа в двійкову систему числення.
2) Виконати операцію $\mathbf{A}_{(2)}$x$\mathbf{B}_{(2)}$ з представленням чисел у форматі із фіксованою перед старшим розрядом комою. Кількість розрядів n=6.

В таблиці використовуються такі позначення номеру алгоритму множення:

1 -множення молодшими розрядами множника зі зсувом суми часткових добутків, що накопичується, вправо;

2- множення молодшими розрядами множника зі зсувом множеного вліво;

3 -множення старшими розрядами множника зі зсувом суми часткових добутків, що накопичується, уліво;

4 -множення старшими розрядами множника зі зсувом множеного вправо.

При виконанні завдань перевірити правильність результату шляхом обчислення десяткових еквівалентів отриманих результатів.

Таблиця 7.5.

№ вар	B	A	№ алг. умн.
1	0,650	- 0,34	1
2	0,750	- 0,24	2
3	0,150	- 0,125	3
4	0,860	- 0,42	4
5	0,460	- 0,34	1
6	0,780	- 0,16	2
7	0,630	- 0,38	3
8	0,380	- 0,125	4
9	0,530	- 0,234	1
10	0,480	- 0,215	2
11	0,750	- 0,34	3
12	0,950	- 0,24	4
13	0,250	- 0,125	1
14	0,42	-0,34	2
15	0,34	-0,240	3

№ вар	B	A	№ алг. умн.
16	0,560	- 0,16	4
17	0,740	- 0,38	1
18	0,440	- 0,125	2
19	0,650	- 0,234	3
20	0,750	- 0,215	4
21	0,150	- 0,11	1
22	0,860	- 0,230	2
23	0,460	- 0,138	3
24	0,780	- 0,230	4
25	0,630	- 0,140	1
26	0,380	- 0,120	1
27	0,530	- 0,255	2
28	0,480	- 0,122	3
29	0,750	- 0,32	4
30	0,950	- 0,12	1

7.4 Контрольні запитання

1) Перелічіть чотири основних методи машинного множення в прямих кодах.
2) Як визначається знак добутку?
3) Якої кількості цифр у межі може досягатися результат множення двох n-розрядних чисел?
4) Поясніть 1-й алгоритм множення.
5) Поясніть 2-й алгоритм множення.
6) Поясніть 3-й алгоритм множення.
7) Поясніть 4-й алгоритм множення.

8 ВИКОНАННЯ ОПЕРАЦІЇ ДІЛЕННЯ В ЦЕОМ

8.1 Мета заняття

Вивчення основних алгоритмів ділення чисел у цифрових ЕОМ. Придбання практичних навичок виконання операції ділення.

8.2 Короткі основні відомості з теорії та вирішення типових завдань

Відомо 2 основних алгоритму виконання операції ділення [1,4,7,12, 13]:

1) ділення з відновленням залишків;
2) ділення без відновлення залишків.

В обох алгоритмах операції над знаковими розрядами й мантисою проводяться окремо. Знак результату визначається шляхом підсумовування по модулю 2 цифр, записаних у знакових розрядах. Щоб уникнути переповнення розрядної сітки слід дотримуватися умови $|A|<|B|$, де A – ділене, B – дільник.

У результаті ділення числа А на число В повинна бути визначена частка у вигляді:

$$Y = \frac{A}{B} = 0,y_1y_2y_3y_4...y_{i-1}y_i = y_1 \cdot 2^{-1} + y_2 \cdot 2^{-2} + y_3 \cdot 2^{-3} + ... + y_{i-1} \cdot 2^{-(i-1)} + y_i \cdot 2^{-i}$$

При діленні цифри частки послідовно визначаються, починаючи зі старшого розряду, тобто із y_1.

Кількість розрядів у частці визначається необхідною точністю результату.

8.2.1. Ділення чисел з відновленням залишків

Правило ділення з відновленням залишків формулюється таким чином [1,7,8,11]:

Дільник віднімається з діленого і визначається знак нульового (один за одним) залишку. Якщо залишок додатній, тобто $|A|>|B|$, то у псевдознаковому розряді частки проставляється 1, з появою якої формується ознака переповнення розрядної сітки і операція припиняється. Якщо залишок від'ємний, то у псевдознаковому розряді частки записується 0, а потім проводиться відновлення діленого шляхом додавання до залишку дільника. Далі виконується зсув відновленого діленого на один розряд ліворуч і повторне віднімання дільника. Знак одержуваного таким чином залишку визначає першу значущу цифру частки: якщо залишок додатній, то в першому розряді записується 1, якщо від'ємний, то записується 0. Далі, якщо залишок додатній, то він зсувається ліворуч на 1 розряд і від нього віднімається дільник для визначення наступної цифри частки.

Якщо залишок від'ємний, то до нього додається дільник для відновлення попереднього залишку, потім відновлений залишок зсувається на 1 розряд ліворуч і від нього віднімається дільник для визначення наступної цифри частки і т.д. до одержання необхідної кількості цифр частки з урахуванням одного додаткового розряду для округлення, тобто забезпечення необхідної точності ділення.

Оскільки в алгоритмі ділення присутня операція віднімання, то доцільно оперувати із числами, представленими в інверсних кодах, наприклад в модифікованому додатковому. І ще одне зауваження – після обробки знакових розрядів і визначення знаку результату із числами А та В оперують, як з додатніми числами, тобто, наприклад:

$$Y = \frac{A}{(-B)} = -\frac{A}{|B|}$$

Наприклад:

$$A_{(2)} = +0{,}0110 \qquad B_{(2)} = -0{,}1100$$

$$[A]_{пр} = \underset{S_gA}{0}{,}0110 \qquad [B]_{пр} = \underset{S_gB}{1}{,}1100$$

$S_gY=S_gA \oplus S_gB= 0 \oplus 1 = 1$, тобто знак результату буде від'ємний.

$$[A]_{дк}^{м} = 00{,}0110$$

$$\big[-|B|\big]_{дк}^{м} = 11{,}0100 = [B]_{дк}^{м} \qquad |B| = 0{,}1100 \qquad \big[|B|\big]_{дк}^{м} = 00{,}1100$$

Результати обчислень зведені в табл. 8.1. Символом * відзначені ситуації появи переносу зі знакового розряду, який, завдяки використанню додаткового коду, ігнорується.

Таблиця 8.1

№ циклу	№ такту	Найменування операцій	Дія	Знак розр.	Поле числа	Псевдознак, розряди частки
0	1 2 3	Віднімання дільника з діленого Встановлення знака нул. залишку і псевдо знака частки Відновлення діленого	$[A]^{м}_{дк} + [-\vert B\vert]^{м}_{дк}$ $R_0 + [\vert B\vert]^{м}_{дк}$ $[A]^{м}_{дк}$ *	00 11 11 00 00	0110 0100 1010 1100 0110	0, 1 0 0 0 0
1	1 2 3	Зсув діленого ліворуч Віднімання дільника Отримання залишку і першої значимої цифри частки	$[A]^{м}_{дк} \leftarrow$ $+ [-\vert B\vert]^{м}_{дк}$ * R_1	00 11 00	1100 0100 0000	
2	1 2 3	Зсув залишку ліворуч Віднімання дільника Отримання залишку і другої цифри частки	$R_1 \leftarrow$ $+ [-\vert B\vert]^{м}_{дк}$ R_2	00 11 11	0000 0100 0100	
3	1 2 3	Відновлення попереднього залишку Зсув залишку ліворуч Віднімання дільника Отримання залишку і встановлення наступної цифри частки	$R_2 + [\vert B\vert]^{м}_{дк}$ R_1 * $R_1 \leftarrow$ $+ [-\vert B\vert]^{м}_{дк}$ R_3	11 00 00 00 11 11	0100 1100 0000 0000 0100 0100	
4	1 2 3	Відновлення попереднього залишку Зсув залишку ліворуч Віднімання дільника. Отримання залишку і встановлення наступної цифри частки	R_3 $+ [\vert B\vert]^{м}_{дк}$ * R_1 $R_1 \leftarrow$ $+ [-\vert B\vert]^{м}_{дк}$ R_4	11 00 00 00 11 11	0100 1100 0000 0000 0100 0100	
5	1 2 3	Відновлення попереднього залишку Зсув залишку ліворуч Віднімання дільника. Отримання залишку і встановлення наступної цифри частки	R_4 $+ [\vert B\vert]^{м}_{дк}$ * R_1 $R_1 \leftarrow$ $+ [-\vert B\vert]^{м}_{дк}$	11 00 00 00 11 11	0100 1100 0000 0000 0100 0100	

$[Y]_{пр} = 1{,}10000$ $\quad Y_{(2)} = -0{,}10000$ $\quad Y_{(2)} = -0{,}1000$ □ $Y_{(10)} = -0{,}5$

Очікуваний результат: $Y = \dfrac{A}{B} = \dfrac{+0{,}4}{-0.8} = -0{,}5$.

Розглянемо ще один приклад з іншими числами і меншої розрядності.

A=0,101 B=-0,110 $\quad |A| > |B|$ $\quad [A]^{м}_{дк} = 00{,}101$ $\quad [|B|]^{м}_{дк} = 00{,}110$

$[-|B|]^{м}_{дк} = 11{,}010$ $\quad Y = \dfrac{A}{B}$ $\quad S_g A = 0$ $\quad S_g B = 1$ $\quad S_g A \oplus S_g B = 0 \oplus 1 = 1$

$A_{(10)} = 0{,}625$

$B_{(10)} = -0{,}75$ Очікуваний результат: $Y_{(10)} = -0{,}833$.

Результати обчислень зведені в табл. 8.2. Символом * відзначені ситуації появи переносу зі знакового розряду, який ігнорується.

Таблиця 8.2

№ циклу	№ такту	Найменування операцій	Дія	Знак розр.	Поле числа	Псевдознак, розряди частки
0	1 2 3	Віднімання дільника з діленого Встановлення знака нул. залишку і псевдо знака частки Відновлення діленого	$[A]^{м}_{дк} + [-\|B\|]^{м}_{дк}$ $R_0 + [\|B\|]^{м}_{дк}$ $[A]^{м}_{дк}$ *	00 11 11 00 00	101 010 111 110 101	0, 1 1 0 1
1	1 2 3	Зсув діленого ліворуч Віднімання дільника Отримання залишку і першої значимої цифри частки	$\leftarrow [A]^{м}_{дк}$ $+[-\|B\|]^{м}_{дк}$ * R_1	01 11 00	010 010 100	
2	1 2 3	Зсув залишку ліворуч Віднімання дільника Отримання залишку і другої цифри частки	$\leftarrow R_1$ $+[-\|B\|]^{м}_{дк}$ * R_2	01 11 00	000 010 010	
3	1 2 3	Зсув залишку ліворуч Віднімання дільника Отримання залишку і встановлення наступної цифри частки	$\leftarrow R_2$ $+[-\|B\|]^{м}_{дк}$ R_3	00 11 11	100 010 110	
4	1 2 3	Відновлення попереднього залишку Зсув залишку ліворуч Віднімання дільника. Отримання залишку і встановлення наступної цифри частки	R_3 $+[\|B\|]^{м}_{дк}$ * R_2 $\leftarrow R_2$ * $+[-\|B\|]^{м}_{дк}$ R_4	11 00 00 01 11 00	110 110 100 000 010 010	

Остаточний результат

$[Y]_{пр} = 1{,}1101$ $\quad Y_{(2)} = -0{,}1101$ $\quad Y_{(2)} = -0{,}111$ □ $Y_{(10)} = -0{,}875$

Порівняно низька точність обчислень у розглянутому прикладі обумовлена малою розрядністю представлення двійкових чисел.

З табл. 8.1 і 8.2 слідує, що кожний цикл ділення складається з трьох тактів. Скоротити число тактів до двох у кожному циклі можна, використовуючи алгоритм ділення двох чисел без відновлення залишку.

8.2.2 Ділення чисел без відновлення залишку

Правило ділення по цьому методу формулюється так [1,7,8,11]:

Щоб визначити чергову цифру частки, необхідно зсунути поточний залишок ліворуч на один розряд, а потім алгебраїчно додати до нього модуль дільника, якому приписується знак протилежний знаку поточного залишку. Знак отриманого таким чином наступного залишку і визначає наступну цифру частки: якщо залишок додатній, то в частці записується 1, якщо від'ємний – записується 0. Операція зсувів і алгебраїчних додавань повторюється доти, поки в частці не буде отримано необхідну кількість розрядів.

Приклад:

Дано:

$A_{(2)} = +0{,}0110$ $\quad B_{(2)} = -0{,}1100$ $\quad [B]_{пр} = 1{,}1100$

$[A]_{пр} = 0{,}0110$ $\quad |B| = 0{,}1100$ $\quad [-|B|]^{м}_{дк} = 11{,}0100$ $\quad [|B|]^{м}_{дк} = 00{,}1100$

$S_gY = S_gA \oplus S_gB = 0 \oplus 1 = 1$ (-) знак результату.

Результати обчислень показані в табл. 8.3.

Таблиця 8.3

№ циклу	№ так-ту	Найменування операцій	Дія	Знак розр.	Поле числа	Псевдознак, розряди частки
0	1 2	Вирахування дільника з діленого Формування нульового залишку і встановлення псевдознаку.	$[A]_{дк}^{м}+[-\lvert B\rvert]_{дк}^{м}$ R_0	00 11 11	0110 0100 1010	0,1 0 0 0 0
1	1 2	Зсув залишку ліворуч Додавання модуля дільника. і формування розряда частки	$R_0 \leftarrow$ $+[\lvert B\rvert]_{дк}^{м}$ * R_1	11 00 00	0100 1100 0000	
2	1 2	Зсув залишку ліворуч Віднімання модуля дільника і формування розряду частки	$R_1 \leftarrow$ $+[-\lvert B\rvert]_{дк}^{м}$ R_2	00 11 11	0000 0100 0100	
3	1 2	Зсув залишку ліворуч Додавання модуля дільника. і формування розряду частки	$R_2 \leftarrow$ $+[\lvert B\rvert]_{дк}^{м}$ R_3	10 00 11	1000 1100 0100	
4	1 2	Зсув залишку ліворуч Додавання модуля дільника. і формування розряду частки	$R_3 \leftarrow$ $+[\lvert B\rvert]_{дк}^{м}$ R_4	10 00 11	1000 1100 0100	
5	1 2	Зсув залишку ліворуч Додавання модуля дільника. і формування розряду частки	$R_4 \leftarrow$ $+[\lvert B\rvert]_{дк}^{м}$ R_5	10 00 11	1000 1100 0100	

І так далі

$[Y]_{пр}=1{,}10000$ $\quad Y_{(2)}=-0{,}10000$ $\quad$ □$Y_{(2)}=-0{,}1000$

□$Y_{(2)}=-0{,}1000$ $\quad$ □$Y_{(10)}=-0{,}5$

8.3 Індивідуальні завдання

Дані числа **A, B** у десятковій системі числення (конкретні значення чисел для кожного студента задаються по варіантах у табл. 8.1).

Необхідно виконати і представити в письмовій формі хід наступних операцій.

1) Перевести числа в двійкову систему числення.
2) Виконати операцію $\mathbf{A_{(2)}:B_{(2)}}$ з представленням чисел у форматі із фіксованою перед старшим розрядом комою. Алгоритм ділення – без відновлення залишків. Кількість розрядів n=6.

При виконанні завдань перевірити правильність результату шляхом обчислення десяткових еквівалентів отриманих результатів.

Таблиця 8.1.

№ вар	B	A	№ алг. ділення
1	0,650	- 0,34	без відновлення залишку
2	0,750	- 0,24	
3	0,150	- 0,125	
4	0,860	- 0,42	
5	0,460	- 0,34	
6	0,780	- 0,16	
7	0,630	- 0,38	
8	0,380	- 0,125	
9	0,530	- 0,234	
10	0,480	- 0,215	
11	0,750	- 0,34	
12	0,950	- 0,24	
13	0,250	- 0,125	
14	0,42	-0,34	
15	0,34	-0,240	

.№ вар	B	A	№ алг. ділення
16	0,560	- 0,16	без відновлення залишку
17	0,740	- 0,38	
18	0,440	- 0,125	
19	0,650	- 0,234	
20	0,750	- 0,215	
21	0,150	- 0,11	
22	0,860	- 0,230	
23	0,460	- 0,138	
24	0,780	- 0,230	
25	0,630	- 0,140	
26	0,380	- 0,120	
27	0,530	- 0,255	
28	0,480	- 0,122	
29	0,750	- 0,32	
30	0,950	- 0,12	

8.4 Контрольні запитання

1) Назвіть два основні алгоритми виконання операції ділення.
2) Як визначається знак частки.
3) Які умови дозволяють уникнути переповнення розрядної сітки.
4) Сформулюйте правило ділення з відновленням залишків.
5) Сформулюйте правило ділення без відновлення залишків.
6) Сформулюйте правило ділення чисел, представлених у формі з плаваючою комою.

9 ВИКОНАННЯ ОПЕРАЦІЇ ДОДАВАННЯ ЧИСЕЛ У ПРЯМИХ ДВІЙКОВО-ДЕСЯТКОВИХ КОДАХ

9.1 Мета заняття

Вивчити і придбати навички виконання арифметичних операцій з числами, які представлені у прямих двійково-десяткових кодах Д1 і Д2.

9.2 Короткі основні відомості з теорії та вирішення типових завдань

Двійково-десяткова арифметика (ДДА) основана на використанні двійково-десяткових кодів (ДДК, або Д-коди).

9.2.1 Д-коди

Д-Код (двійково-кодоване представлення) десяткового числа – це таке його представлення, у якому кожна десяткова цифра зображується тетрадою із двійкових символів [1, 3, 11]:

$$A_Д = a_{n-1}\, a_{n-2} \ldots a_i \ldots a_1\, a_0, \tag{9.1}$$

де $a_i = L_4^{(i)}\ L_3^{(i)}\ L_2^{(i)}\ L_1^{(i)}$ - десяткова цифра i-того розряду,

$L_j^{(i)}$ - двійкова цифра j-тої тетради $\in \{0,1\}$.

Існують різні Д-Коди, оскільки із 16 можливих комбінацій у кожній тетраді використовується тільки 10 [1, 3, 5, 7, 8, 10, 13].

Для однозначності переведення чисел у Д-Код і назад бажано, щоб усі розряди тетрад мали певні ваги. Тоді значення десяткової цифри a_i відповідає виразу:

$$a_i = L_4 \cdot G_4 + L_3 \cdot G_3 + L_2 \cdot G_2 + L_1 \cdot G_1, \tag{9.2}$$

де G_i – вага розряду тетради.

Наприклад, існують системи з вагами розрядів 8,4,2,1; 2,4,2,1; 5,4,2,1; 7,4,2,1 і таке інше. Однак не всі Д-Коди відповідають умові (9.2). У рамках цього практичного заняття розглянемо властивості та особливості виконання операцій алгебраїчного додавання чисел, представлених у кодах Д1 (код з вагами 8,4,2,1) – код природнього заміщення, Д2 (код з надлишком 3 ((8,4,2,1) + 3))

У таблиці 9.1 представлене кодування десяткових цифр у кодах Д1 і Д2.

Таблиця 9.1

Десяткові цифри	Еквіваленти в кодах	
	Д1 8 4 2 1	Д2 ((8421) +3)
0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 1 0 1
3	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 0 0
6	0 1 1 0	1 0 0 1
7	0 1 1 1	1 0 1 0
8	1 0 0 0	1 0 1 1
9	1 0 0 1	1 1 0 0

Інші комбінації в цих кодах заборонені (не використовуються)

З таблиці 9.1 неважко помітити, що код Д2 умові (9.2) не відповідає.

Керуючись табл. 9.1 легко переводити числа з десяткової системи числення у двійково-десяткові системи Д1 і Д2. Наприклад, числа

$A_{(10)} = +159$ $B_{(10)} = -0{,}258$

У кодах Д1 запишуться таким чином:

$A_{Д1} = +\ 0001\ 0101\ 1001$ $B_{Д1} = -0,\ 0010\ 0101\ 1000,$

а в кодах Д2: $A_{Д2} = +\ 0100\ 1000\ 1100$ $B_{Д2} = -0,\ 0101\ 1000\ 1011.$

Це були записи в природній формі. При запису в машинних кодах один або два розряди приділяється під знак числа. Наприклад, прямі коди чисел А і В у системах Д1 і Д2 запишуться як

$[A_{Д1}]_{пр.} = 0\ 0001\ 0101\ 1001$ $[B_{д1}]_{пр.} = 1,\ 0010\ 0101\ 1000$

$[A_{Д2}]_{пр.} = 0\ 0100\ 1000\ 1100$ $[B_{Д2}]_{пр.} = 1,\ 0101\ 1000\ 1011$

9.3 Додавання чисел в прямих кодах Д1

При додаванні чисел у прямих кодах Д1 можуть виникнути наступні випадки:

1) якщо $a_i + b_i + П_{i-1} < 10_{(10)}$ ($П_{i-1}$ – перенос із попередньої тетради), то при виконанні дій над розрядами тетради за правилами двійкової арифметики відразу утворюється правильний результат. Наприклад:

$A_{(10)} = +5$; $B_{(10)} = +2$ $П_0 = 1$. $C_{(10)} = 5+2+1 = 8$

$A_{Д1} = a_1 = +0101$ $B_{Д1} = b_1 = +0010$

$C_1 = +1000_{(д1)} = +8_{(10)}$

$$\begin{array}{r} a_1 = 0101 \\ +\ b_1 = 0010 \\ +П_0 = 0001 \\ \hline C_1 = 1000 \end{array}$$

2) Якщо $a_i + b_i + П_{i-1} \geq 10_{(10)}$, то виникає десятковий перенос. Тому сума в даній тетраді повинна дорівнювати:

$$C_i = a_i + b_i + П_{i-1} - П_i \times 10,$$

де $П_i = 1$ - це перенос з і-тої тетради в $(i+1)$-у, тобто потрібна корекція помилкового результату.

При цьому ознакою неправильного результату є:

- в одному випадку – виникнення потетрадного переносу $П_i{}'_{(10)} = 16$,

$$19 \geq (a_i + b_i + П_{i-1}) \geq 16,$$

- у другому випадку – поява забороненої комбінації, якщо

$$15 \geq (a_i + b_i + П_{i-1}) \geq 10.$$

У кожному із цих випадків необхідно скорегувати результат у даній тетраді введенням поправки *(+0110)*, що може привести до виникнення потетрадного переносу. Корекція обумовлена тим, що кожний перенос несе із собою з даної тетради 16 одиниць, а приносить у наступну тільки 10 одиниць.

Приклад 1

$a_i = 1000$ $b_i = 1001$ $П_{i-1} = 1$ очікувана сума $C_{(10)} = 8+9+1=18$

$a_i = 1000 \to 8$, $b_i = 1001 \to 9$

$$\begin{array}{r} a_i = 1000 \\ +\ \ b_i = 1001 \\ +\ П_{i-1} = 0001 \\ \hline C_i{}' = 1\ 0010 \end{array}$$

невідкорегована сума

$\downarrow$ $\downarrow$

$П_i{}'$ 2

1 2 - результат помилковий

$C_i = C_i{}' + 0110$

$$\begin{array}{r} C_i = 1\ \ 0010 \\ +\ \ \ \ 0110 \\ \hline 1\ \ 1000 \end{array}$$

корекція

$\downarrow$ $\downarrow$

$П_i{}'$ 8

1 8 - тепер результат вірний

Приклад 2:

$a_i = 1000$ $\rightarrow$ 8 $b_i = 0110$ $\rightarrow$ 6 $П_{i-1} = 1$ очікувана сума 8+6+1=15

$$
\begin{array}{l}
\quad a_i = 1000 \\
+ \\
\quad b_i = 0110 \\
+ \\
Пi\text{-}1 = 0001 \\
\hline
C_i' = 0\ 1111
\end{array}
$$

- переносу немає, а результат належить до забороненої в коді Д1 комбінації.

Виконуємо корегування суми доданням числа (+ 0110):

$$
\begin{array}{ll}
C_i = 1111 & \\
+ & \\
\quad 0110 & \text{корекція результату} \\
\hline
\quad 1\ 0101 & C_i = 1\ \ 0101 \quad \text{- правильний результат у кодах Д1} \\
\quad \downarrow \quad \downarrow & \\
\quad 1 \quad 5 & П_i = 1
\end{array}
$$

При цьому, якщо в декількох тетрадах, починаючи з (i+1)-ої розрядна сума дорівнює $(1001)_2 = 9_{(10)}$, то перенос приведе до формування забороненої комбінації в наступній (i+1) тетраді. У результаті цього буде потрібна корекція, яка приведе до забороненої комбінації в (i+2) тетраді і т.д. Отже, через послідовне поширення потетрадних переносів час додавання в кодах Д1 складе в найгіршому разі n-тактів, де n - кількість тетрад. Звичайно схеми будують таким чином, щоб перенос, що виникає при додаванні тетродної поправки, проходив крізь тетради, у яких попередня сума дорівнює $9_{(10)} = (1001)_2$ і скидав їх в «нуль». При цьому сума завжди формується за два такти.

Приклад додавання багаторозрядних чисел:

$A = 345_{(10)} = 0011\ 0100\ 0101_{(д1)}$ $П_0 = 0$ 345+591=936-очікуваний результат.

$B = 591_{(10)} = 0101\ 1001\ 0001_{(д1)}$

0011 0100 0101
+
0101 1001 0001

1000 1101 0110 помилковий результат, (2 тетрада містить заборонену комбінацію)
+
0000 0110 0000 корекція,

1001 ←0011 0110 – правильний результат

9 3 6

символом ← позначено потетрадний перенос

9.4 Додавання чисел в прямих кодах Д2

При додаванні в прямих кодах Д2 можливі наступні випадки, з урахуванням того, що $a_i' = a_i + 3$; $b_i' = b_i + 3$, де a_i' та b_i' – тетради для коду Д2.

1) Якщо $a_i' + b_i' + П_{i-1} \le 15$, то $a_i' + b_i' + П_{i-1} = C_i + 6 = C_i' + 3$, тобто з результату необхідно відняти $3_{(10)} = (0011)_2$. Оскільки суматори не роблять операцію віднімання, то поправка $(-0011)_2$ заміняється додаванням у дану тетраду числа $(13)_{10} = (1101)_2$ (додатковий код числа -3) і блокуванням виникаючого при цьому переносу з даної тетради в старшу.

2) Якщо ж $a_i' + b_i' + П_{i-1} > 15$, то $a_i' + b_i' + П_{i-1} = C_i' - 3$. Тут виникає десятковий перенос, який при переході в старшу тетраду міняє свою вагу з 16 на 10. Тому в цьому випадку потрібне виправлення результату на величину $(+0011)_2 = +3_{(10)}$.

Тобто, якщо при потетрадному додаванні перенос не виникає, то корекція дорівнює (1101), а якщо виникає, то корекція дорівнює +(0011).

Приклад.

$A=462_{(10)}$= 0111 1001 0101 $_{(Д2)}$; очікуваний результат 462

$B=374_{(10)}$= 0110 1010 0111 $_{(Д2)}$; +374

836

$A_{Д2}$= 0111 1001 0101
+$B_{Д2}$= 0110 1010 0111
1110←0011 1100 – помилковий результат,
+1101 0011 1101 – корекція,
$C_{Д2}$=*1011 0110*1001 – *- ігнорування переносу,
8 3 6 – правильний результат.

Таким чином, при додаванні в кодах Д2 не виникають проблеми наскрізного переносу і операція додавання виконується за два такти.

9.4 Індивідуальні завдання

Представити числа в Д-кодах у відповідності до варіанта завдання.

Таблица 9.1

№ вар.	B	A	код	код в сумм.	№ вар.	B	A	код	код в сумм.
1	0,658	- 0,341	Д1	ДК	16	0,564	- 0,166	Д2	ОК
2	0,759	- 0,247	Д1	ОК	17	0,748	- 0,381	Д1	ДК
3	0,156	- 0,128	Д2	ДК	18	0,448	- 0,126	Д1	ОК
4	0,860	- 0,423	Д2	ОК	19	0,655	- 0,233	Д2	ДК
5	0,467	- 0,343	Д1	ДК	20	0,752	- 0,219	Д2	ОК
6	0,781	- 0,166	Д1	ОК	21	0,659	- 0,118	Д1	ДК
7	0,635	- 0,384	Д2	ДК	22	0,863	- 0,239	Д1	ОК
8	0,387	- 0,126	Д2	ОК	23	0,462	- 0,135	Д2	ДК
9	0,535	- 0,239	Д1	ДК	24	0,788	- 0,237	Д2	ОК
10	0,481	- 0,215	Д1	ОК	25	0,636	- 0,146	Д1	ДК
11	0,754	- 0,348	Д2	ДК	26	0,381	- 0,122	Д1	ДК
12	0,953	- 0,244	Д2	ОК	27	0,537	- 0,255	Д1	ОК
13	0,259	- 0,123	Д1	ДК	28	0,485	- 0,126	Д2	ДК
14	0,425	-0,347	Д1	ОК	29	0,751	- 0,328	Д2	ОК
15	0,347	-0,242	Д2	ДК	30	0,953	- 0,129	Д1	ДК

9.5 Контрольні запитання

1) Яке представлення двійкових чисел називається двійково-кодованим?
2) Наведіть приклади Д-Кодів.
3) Зобразіть таблицю кодування десяткових чисел від 0 до 9 у кодах Д1 (з вагами 8, 4, 2, 1) і Д2 (код з надлишком 3)
4) Запишіть числа $A_{(10)} = +246$ і $B_{(10)} = -0,678$ у кодах Д1 і Д2 у природній формі запису і у прямих кодах.
5) У чому полягають особливості додавання чисел у прямих кодах Д1?
6) У чому полягають особливості додавання чисел у прямих кодах Д2?

10 ВИКОНАННЯ ОПЕРАЦІЇ ДОДАВАННЯ ЧИСЕЛ У ІНВЕРСНИХ ДВІЙКОВО-ДЕСЯТКОВИХ КОДАХ

10.1 Мета заняття

Вивчити і придбати навички виконання арифметичних операцій з числами, які представлені у інверсних двійково-десяткових кодах Д1 і Д2.

10.2 Короткі основні відомості з теорії та вирішення типових завдань

Д-Коди можуть бути представлені в розрядній сітці машини у формі з фіксованою комою, або і плаваючою комою. При цьому від'ємні числа можуть бути представлені в ПК, ОК, ДК. Тому якщо

$A_Д = 0, a_1 a_2 \dots a_i \dots a_{n-1} a_n$, где a_i – тетради, то [1, 3, 5, 7, 11]

$$
\begin{aligned}
[A_Д]_{ПР} &= 1, a_1 a_2 \dots a_i \dots a_{n-1} a_n \\
[A_Д]_{ОК} &= 1, \bar{\bar{a}}_1 \bar{\bar{a}}_2 \dots \bar{\bar{a}}_i \dots \bar{\bar{a}}_{n-1} \bar{\bar{a}}_n \qquad (10.1) \\
[A_Д]_{ДК} &= 1, \bar{\bar{a}}_1 \bar{\bar{a}}_2 \dots \bar{\bar{a}}_i \dots \bar{\bar{a}}_{n-1} \bar{a}_n,
\end{aligned}
$$

де

$\bar{\bar{a}}_i$ – доповнення до 9 у всіх тетрадах, тобто $\bar{\bar{a}}_i + a_i = 9$,

$\bar{a}_n$ – доповнення до 10 в молодшій тетраді, тобто $\bar{a}_n + a_n = 10$.

Найпростіше ОК і ДК числа формується в кодах Д2. Для одержання ОК досить проінвертувати усі розряди в тетрадах, а ДК числа утворюється додаванням 1 у молодшу тетраду.

Наприклад:

$A = -0{,}138_{(10)} = -0{,}\ 0100\ 0110\ 1011_{\ (Д2)}$ $\qquad [A_{Д2}]_{ПР} = 1,\ 0100\ 0110\ 1011$

$$
\begin{array}{r}
[A_{Д2}]_{ОК} = 1,\ 1011\ 1001\ 0100 \\
+ \qquad\qquad\qquad 1 \\
\hline
[A_{Д2}]_{ДК} = 1,\ 1011\ 1001\ 0101
\end{array}
$$

Трохи складніше переводяться числа в ОК і ДК у кодах Д1. Це пов'язане з тим, що просте інвертування набору тетрад у кодах Д1 означає одержання доповнення до 15. Отже, необхідно усунути різницю. Для цього в усі цифрові тетради числа в кодах Д1 додається $(+0110)_2 = +6_{(10)}$ після чого проводиться інвертування набору. Отримане зображення являє собою обернений код числа. Додаванням 1 у молодшу тетраду оберненого коду утворюється додатковий код числа.

Наприклад:

$A=-0,138_{(10)} = -0,0001\ 0011\ 1000_{(Д1)}$

Прямий код числа

$[A_{Д1}]_{ПР} = 1,0001\ 0011\ 1000$
$+\ 0110\ 0110\ 0110$ – корекція,

$1,0111\ 1001\ 1110$ – потім інвертуємо цифри тетрад

$[A_{Д1}]_{ОК} = 1,1000\ 0110\ 0001$
$+\ 1$

$[A_{Д1}]_{ДК} = 1,1000\ 0110\ 0010$

Розглянемо приклади алгебраїчного додавання чисел в ОК і ДК представлених у кодах Д1 і Д2.

Приклад 1:

Додати в кодах Д1 на суматорі оберненого коду числа:

$A_{(10)} = +0,264$, $B_{(10)} = -0,758$. Очікуваний результат $C_{(10)} = -0,494$.

$[A_{Д1}]_{ПР} = 0,0010\ 0110\ 0100$
$[A_{Д1}]_{ОК} = 0,0010\ 0110\ 0100$
$[B_{Д1}]_{ПР} = 1,0111\ 0101\ 1000$
$+\ 0110\ 0110\ 0110$ – корекція (поправка),

$1,1101\ 1011\ 1110$ – інвертуємо цифри тетрад,
$[B_{Д1}]_{ОК} = 1,0010\ 0100\ 0001$

Додаємо числа в ОК:

$[A_{Д1}]_{ОК} = 0,0010\ 0110\ 0100$
$+$
$[B_{Д1}]_{ОК} = 1,0010\ 0100\ 0001$

$[C_{Д1}]'_{ОК} = 1,0100\ 1010\ 0101$
$+\ 0000\ 0110\ 0000$ – потетрадна корекція.
$[C_{Д1}]_{ОК} = 1,0101\ 0000\ 0101$ – остаточний результат.

Тепер перевіримо правильність результату, перевівши його в прямий код, і знайдемо його десятковий еквівалент.

$[C_{Д1}]_{ОК}$ = 1, 0101 0000 0101 – корекція для переходу із оберненого коду в прямий.
+ 0110 0110 0110

$[C_{Д1}]'_{ОК}$ = 1, 1011 0110 1011

Після інвертування чисел розрядів отримуємо прямий код

$$[C_{Д1}]_{ПР} = 1,\ 0100\ 1001\ 0100,$$

десятковий еквівалент якого дорівнює $C_{(10)}$ = -0,494, тобто, дорівнює очікуваному.

Приклад 2:

Додати в кодах Д1 на суматорі оберненого коду числа

$A_{(10)}$ = -0,264, $B_{(10)}$ +0,758. Очікуваний результат $C_{(10)}$ =+0,494.

$[A_{Д1}]_{ПР}$ = 1, 0010 0110 0100
+ 0110 0110 0110 – корекція для одержання доповнення до 9

1, 1000 1100 1010 – результат інвертуємо

$[A_{Д1}]_{ОК}$ = 1, 0111 0011 0101 – обернений код
+
$[B_{Д1}]_{ОК}$ = 0, 0111 0101 1000

1, 1110 1000 1101 – потетрадна корекція (перенос, або сума >$10_{(10)}$)
+ 0110 0000 0110

1 0, 0100 1001 0011 – циклічний перенос,
+ 1

$[C_{Д1}]_{ОК}$ = 0, 0100 1001 0100 $[C_{Д1}]_{ПР}$ – остаточний результат, тому що сума > 0.
+0, 4 9 $4_{(10)}$

$C_{(10)}$ = +0,494

Приклад 3.

Додати в кодах Д1 на суматорі додаткового коду числа:

$A_{(10)}$ = -0,264, $B_{(10)}$ +0,758, $[A_{Д1}]_{ПР}$ = 1, 0010 0110 0100,

$[A_{Д1}]_{ОК}$ = 1, 0111 0011 0101, (див. приклад 2)

$[B_{Д1}]_{ОК}$ = 0, 0111 0101 1000= $[B_{Д1}]_{ДК}$

$[A_{Д1}]_{ДК}$ = $[A_{Д1}]_{ОК}$ + 1 молодша тетрада

$[A_{Д1}]_{ДК}$ = 1, 0111 0011 0101
+ 1

1, 0111 0011 0110
+
$[B_{Д1}]_{ДК}$ = 0, 0111 0101 1000

1, 1110 1000 1110
+ 0110 0000 0110 – корекція,

* 0, 0100 1001 0100 = $[C_{Д1}]_{ДК}$ = $[C_{Д1}]_{ПР}$ тому що C >0.
+ 0, 4 9 4

* - перенос зі знакового розряду ігнорується (особливість додавання в ДК).

Приклад 4:

Додати в кодах Д2 на суматорі оберненого коду числа:

$A_{(10)}$ = -0,264, $B_{(10)}$ +0,758.

$[A_{Д2}]_{ПР}$ = 1, 0101 1001 0111

$[A_{Д2}]_{ОК}$ = 1, 1010 0110 1000 $[B_{Д2}]_{ПР}$ = 0, 1010 1000 1011= $[B_{Д2}]_{ОК}$
+
$[B_{Д2}]_{ОК}$ =0,1010 1000 1011

1 0,0100 1111 0011
+ 1 – циклічний перенос.

0, 0100 1111 0100 потетрадна корекція, міжтетрадні переноси
+ ігноруються.
0011 1101 0011

$[C_{Д2}]_{ОК}$ = 0, 0111 1100 0111 = $[C_{Д2}]_{ПР}$ тому що C>0.
+0, 4 9 4

Приклад 5:

Додати в кодах Д2 на суматорі додаткового коду числа:

$A_{(10)}$ = +0,264, $B_{(10)}$= -0,758.

$[A_{Д2}]_{ПР}$ = 0, 0101 1001 0111

$[A_{Д2}]_{ДК}$ = 0, 0101 1001 0111

$[B_{Д2}]_{ПР}$ = 1, 1010 1000 1011

$[B_{Д2}]_{ДК}$ = 1, 0101 0111 0101

$[A_{Д2}]_{ДК}$ = 0, 0101 1001 0111
+
$[B_{Д2}]_{ДК}$ = 1, 0101 0111 0101

1, 1011 0000 1100 – помилкова сума,
1101 0011 1101 – корекція, міжтетрадні переноси ігноруються

$[C_{Д2}]_{ДК}$ = 1, 1000 0011 1001 – остаточний результат

Для контролю правильності розрахунків переводимо в прямий код і знаходимо десятковий еквівалент суми.

$[C_{Д2}]_{ПР}$ = 1, 0111 1100 0111. $C_{(10)}$ = - 0,494 - дорівнює очікуваній сумі.

- 0, 4 9 4

10.3 Індивідуальні завдання

Дані числа **A, B** у десятковій системі числення (конкретні значення чисел для кожного студента задаються по варіантах у табл. 10.1).

Необхідно виконати і представити в письмовій формі хід наступних операцій.

1) Представити числа в кодах Д1 та Д2.

2) Виконати операцію **A+B** в Д-кодах відповідно до завдання.

При виконанні завдань по п.2 перевірити правильність результату шляхом обчислення десяткових еквівалентів отриманих результатів.

Таблиця 10.1

№ вар	B	A	коді-ровка	код в сумм.	№ вар	B	A	коді-ровка	код в сумм.
1	0,658	- 0,341	Д1	ДК	16	0,564	- 0,166	Д2	ОК
2	0,759	- 0,247	Д1	ОК	17	0,748	- 0,381	Д1	ДК
3	0,156	- 0,128	Д2	ДК	18	0,448	- 0,126	Д1	ОК
4	0,860	- 0,423	Д2	ОК	19	0,655	- 0,233	Д2	ДК
5	0,467	- 0,343	Д1	ДК	20	0,752	- 0,219	Д2	ОК
6	0,781	- 0,166	Д1	ОК	21	0,659	- 0,118	Д1	ДК
7	0,635	- 0,384	Д2	ДК	22	0,863	- 0,239	Д1	ОК
8	0,387	- 0,126	Д2	ОК	23	0,462	- 0,135	Д2	ДК
9	0,535	- 0,239	Д1	ДК	24	0,788	- 0,237	Д2	ОК
10	0,481	- 0,215	Д1	ОК	25	0,636	- 0,146	Д1	ДК
11	0,754	- 0,348	Д2	ДК	26	0,381	- 0,122	Д1	ДК
12	0,953	- 0,244	Д2	ОК	27	0,537	- 0,255	Д1	ОК
13	0,259	- 0,123	Д1	ДК	28	0,485	- 0,126	Д2	ДК
14	0,425	-0,347	Д1	ОК	29	0,751	- 0,328	Д2	ОК
15	0,347	-0,242	Д2	ДК	30	0,953	- 0,129	Д1	ДК

10.4 Контрольні запитання

1) Яке представлення двійкових чисел називається двійково-кодованим?
2) Наведіть приклади Д-Кодів.
3) Зобразіть таблицю кодування десяткових чисел від 0 до 9 у кодах Д1 (з вагами 8, 4, 2, 1) і Д2 (код з надлишком 3)
4) Запишіть числа $A_{(10)} = +246$ і $B_{(10)} = -0,678$ у кодах Д1 і Д2 у природній формі запису і у прямих кодах.
5) Як отримати інверсний і додатковий код числа в кодах Д2?
6) Як отримати інверсний і додатковий код числа в кодах Д1?
7) Особливості алгебраїчного додавання чисел в ПК і ДК представлених у кодах Д1 і Д2.

11. БУЛЕВА АЛГЕБРА. ОСНОВНІ ПОНЯТТЯ БУЛЕВОЇ АЛГЕБРИ. МІНІМІЗАЦІЯ БУЛЕВИХ ФУНКЦІЙ.

11.1 Мета заняття

Вивчити основні поняття булевої алгебри та придбати навички виконання логічних операцій, навички роботи з булевими функціями, записаними в різних формах, опанувати основні методи мінімізації булевих функцій.

11.2 Короткі основні відомості з теорії та вирішення типових завдань

Технічні питання, пов'язані зі складанням логічних схем ЕОМ, можна вирішити за допомогою математичного апарату, об'єктом дослідження якого є функції, що приймають, так само як і їхні аргументи, тільки два значення - "0" і "1".

Таким апаратом є математична логіка (алгебра логіки, булева алгебра).

Логіка - це наука про закони і форми мислення.

Математична логіка займається вивченням можливостей застосування формальних методів для вирішення логічних завдань. Один з розділів математичної логіки є алгебра логіки.

Основне поняття алгебри логіки - висловлювання. Висловлення - це деякий вислів, про який можна стверджувати, що він істинний або хибний.

Будь-яке висловлювання можна позначити символом x і вважати, що x = 1, якщо висловлювання істинно, а x = 0 - якщо висловлювання помилкове. Істинному висловлюванню відповідає твердження - "Так", помилковому висловлюванню - твердження - "Ні".

Логічна (булева) змінна - така величина x, яка може приймати тільки два значення x = {0,1}.

Висловлення абсолютно істинне, якщо відповідна йому логічна величина приймає значення x = 1 при будь-яких умовах.

Висловлення абсолютно помилкове, якщо відповідна йому логічна величина приймає значення x = 0 при будь-яких умовах.

Функція f, що залежить від n змінних $x_1,x_2,...,x_n$, називається булевою, або перемикальною, якщо функція f і будь-який з її аргументів $x_i, i \in \overline{1,n}$ приймають значення тільки з множини {0,1}. Аргументи булевої функції також називаються мулевими [1, 3,11].

Способи завдання булевих функцій

Довільна булева функція задається одним з трьох способів: матричним (табличним), геометричним і аналітичним.

При матричному способі булева функція $f(x_1,...,x_n)$ задається таблицею істинності, в лівій частині якої представляються всі можливі двійкові набори довжини n, а в правій вказується значення функції на цих наборах.

Під двійковим набором розуміється сукупність значень аргументів $x_1,x_2,...,x_n$ булевої функції f. Двійковий набір має довжину n, якщо він представлений n цифрами з множини {0,1}.

Іноді двійкові набори в таблиці істинності булевої функції зручно представляти номерами наборів. Запишемо аргументи $x_1,x_2,...,x_n$ в порядку зростання їх індексів. Тоді будь-який двійковий набір можна розглядати як ціле двійкове число N, зване номером набору. Наприклад, двійкові набори 0101 і 1000 мають номера 5 і 8 відповідно. Очевидно, будь-яка булева функція може бути задана таблицею істинності, в якій двійкові набори замінені своїми номерами.

При аналітичному способі булева функція задається формулами, тобто аналітичними виразами, побудованими на основі операцій булевої алгебри. Аналітичний спосіб завдання булевих функцій займає особливе місце в проектуванні цифрових автоматів. Фактично, всі перетворення над булевими функціями, необхідні для побудови цифрових автоматів, ведуться на аналітичному рівні.

Розглянемо області визначення булевих функцій. Між двійковими наборами і двійковими числами існує взаємно однозначна відповідність. Отже, існує 2^n різних наборів двоїчних змінних.

Основні закони і тотожності булевої алгебри

Для булевої алгебри визначені одна одномісна (унарна) операція "заперечення" (інверсія) і дві двомісні (бінарні) операції кон'юнкція і диз'юнкція (позначаються «∧», «∨» відповідно). Наведемо деякі основні формули:

Формули диз'юнкції:

$0 \vee 0 = 0$	$0 \vee x = x$
$0 \vee 1 = 1$	$1 \vee x = 1$
$1 \vee 0 = 1$	$x \vee x \vee x \vee ... \vee x = x$
$1 \vee 1 = 1$	$x \vee \overline{x} = 1$
$x \vee x = x$	

Формули кон'юнкції:

$0 \wedge 0 = 0$	$0 \wedge x = 0$
$0 \wedge 1 = 0$	$1 \wedge x = x$
$1 \wedge 0 = 0$	$x \wedge x \wedge x \wedge \ldots \wedge x = x$
$1 \wedge 1 = 1$	$x \wedge \overline{x} = 0$
$x \wedge x = x$	

Формули інверсії:

$\overline{0} = 1$	$\overline{\overline{x}} = x$
$\overline{1} = 0$	

Формули суми по модулю 2 (mod 2, або $\oplus$):

$0 \oplus 0 = 0$	$x \oplus x = 0$
$0 \oplus 1 = 1$	$1 \oplus x = \overline{x}$
$1 \oplus 0 = 1$	$x \oplus \overline{x} = 1$
$1 \oplus 1 = 0$	$\underbrace{x \oplus x \oplus x \ldots \oplus x}_{n} = \begin{cases} x & \text{при n непарн.} \\ 0 & \text{при n парн.} \end{cases}$
$0 \oplus x = x$	

У цій алгебрі справедливі наступні аксіоми:

$x \vee y = y \vee x$; $x \wedge y = y \wedge x$	закон комутативності (перестановок)
$x \vee y \vee z = (x \vee y) \vee z$	закон асоціативності (сполучний) для диз'юнкції
$x \wedge y \wedge z = (x \wedge y) \wedge z$	закон асоціативності для кон'юнкції
$(x \vee y) \wedge z = x \wedge z \vee y \wedge z$	закон дистрибутивності (розподільний) для кон'юнкції відносно диз'юнкції
$x \vee y \wedge z = (x \vee y) \wedge (x \vee z)$	закон дистрибутивності для диз'юнкції відносно кон'юнкції

Під бінарною операцією на множині *A,* в загальному випадку розуміють відображення декартового добутку множин (А х *А)* в множину А. Іншими словами, результат застосування бінарної операції до будь-якої впорядкованої пари елементів з множини А є також елемент з множини А.

Під унарною операцією на множині А розуміють виділення (фіксацію) якого-небудь елемента множини А.

З комутативності і асоціативності диз'юнкції слідує, що диз'юнкція кількох змінних може виконуватися послідовно, причому порядок взяття диз'юнкції не впливає на результат.

Формули де Моргана:

$$\overline{x \vee y} = \overline{x} \wedge \overline{y} \quad \overline{x \wedge y} = \overline{x} \vee \overline{y}$$

Наслідки з формул де Моргана:

$\overline{\overline{x \wedge y}} = \overline{\bar{x} \vee \bar{y}}$ $\quad\quad\quad$ $\overline{\overline{x \vee y}} = \overline{\bar{x} \wedge \bar{y}}$

Формули для системи функцій:

операція поглинання (x поглинає y): $x \vee xy = x$; або $x(x \vee y) = x$.

операція склеювання: $xy \vee x\bar{y} = x$, або $(x \vee y) \wedge (x \vee \bar{y}) = x$

Аналітичне представлення булевих функцій

Найбільш широке впровадження набула досконала диз’юнктивна нормальна форма (*рус. совершенная дизъюнктивная нормальная форма* - СДНФ). Перш ніж перейти до її вивчення, наведемо визначення конституенти одиниці - поняття, яким будемо широко користуватися надалі.

Конституентою одиниці (кон’юнктивним термом) називається функція $f(x_1,x_2,...,x_n)$, яка приймає значення 1 тільки на одному єдиному наборі.

Конституента одиниці записується як логічний добуток *n* різних булевих змінних, деякі з них можуть бути з запереченнями. Можна легко виразити будь-яку булеву функцію як диз'юнкцію конституент одиниці, відповідних тим наборам, на яких функція приймає значення 1.

Диз'юнкція констітуент 1, рівних 1 на тих же наборах, що і задана функція, називається досконалою диз’юнктивною нормальною формою перемикальної функції (ДДНФ).

Набори, на яких функція f приймає значення 1, часто називаються одиничними, інші - нульовими наборами. Виписувати в ДДНФ має сенс тільки конституенти одиниці, що відповідають одиничним наборам.

Приклад. Випишемо ДДНФ для функції, заданої таблицею істинності (табл. 11.1.).

Таблиця 11.1.

№ набору	$x_1x_2x_3$	$f(x_1,x_2,x_3)$
0	0 0 0	0
1	0 0 1	1
2	0 1 0	1
3	0 1 1	0
4	1 0 0	0
5	1 0 1	1
6	1 1 0	0
7	1 1 1	1

$f_{СДНФ}(x_1,x_2,x_3) = \bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1x_2\bar{x}_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2x_3$

$f_{СКНФ}(x_1,x_2,x_3)= (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3)$

Інша відома форма носить назву досконалої кон'юнктивної нормальної форми (*рус. совершенная конъюнктивная нормальная форма* - СКНФ). Вона будується аналогічно ДДНФ.

Визначення. Конституентою нуля (диз'юнктивним термом) називається функція, що приймає значення 0 на єдиному наборі.

Конституента нуля записується у вигляді елементарної диз'юнкції всіх змінних. Кожному набору відповідає своя конституента 0. Наприклад, набору 0110 змінних x_1, x_2, x_3, x_4 відповідає конституента нуля $(x_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee x_4)$.

ДКНФ представляється як кон'юнкція конституент нуля, що відповідають нульовим наборам функції.

Мінімізація булевих функцій

Загальна задача мінімізації булевих функцій може бути сформульована таким чином: *знайти аналітичний вираз заданої булевої функції в формі, що містить мінімально можливе число букв*

Визначення. Елементарною кон'юнкцією називається кон'юнкція кінцевого числа різних між собою булевих змінних, кожна з яких може мати або не мати заперечення.

Визначення. Диз'юнктивною нормальною формою (ДНФ) називається диз'юнкція елементарних кон'юнкцій.

Визначення. Мінімальною диз'юнктивною нормальною формою булевої функції називається ДНФ, що містить найменше число букв (по відношенню до всіх інших ДНФ, що представляють задану булеву функцію).

Визначення. Булева функція $g(x_1,x_2,...,x_n)$ називається імплікантою булевої функції $f_1(x_1,x_2,...,x_n)$, якщо для будь-якого набору змінних, на якому $g = 1$, справедливо $f = 1$.

Приклад. Булева функція f задана таблицею 11.2.

Там же наведені всі її імпліканти. При запису функції та її імплікант в СДНФ маємо:

Таблиця 11.2.

x_1	x_2	x_3	f	g_1	g_2	g_3	g_4	g_5	g_6	g_7
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
1	1	0	1	0	1	1	0	0	1	1
1	1	1	1	1	0	1	0	1	0	1

f= $\overline{x}_1 x_2 x_3 \vee x_1 x_2 \overline{x}_3 \vee x_1 x_2 x_3$

g_1= $x_1 x_2 x_3$

g_2= $x_1 x_2 \overline{x}_3$

g_3= $x_1 x_2 x_3 \vee x_1 x_2 \overline{x}_3 = x_1 x_2$

g_4= $\overline{x}_1 x_2 x_3$

g_5= $\overline{x}_1 x_2 x_3 \vee x_1 x_2 x_3 = x_2 x_3$

g_6= $\overline{x}_1 x_2 x_3 \vee x_1 x_2 \overline{x}_3$

g_7= $\overline{x}_1 x_2 x_3 \vee x_1 x_2 \overline{x}_3 \vee x_1 x_2 x_3$

Визначення. Імпліканта g булевої функції f, що є елементарною кон'юнкцією, називається простою, якщо жодна частина імпліканти g не є імплікантою функції f.

З прикладу видно, що імпліканти g_3 і g_5 є простими імплікантами функції f. Інші імпліканти не є простими, так як їх частини є імплікантами функції f, наприклад g_3 є частиною g_1. Наведемо без доведення два твердження, корисні при отриманні мінімальної ДНФ.

1. Диз'юнкція будь-якого числа імплікант булевої функції f є також імплікантою цієї функції.

2. Будь-яка булева функція f еквівалентна диз'юнкції всіх своїх простих імплікант. Така форма представлення булевої функції називається скороченою ДНФ.

Перебір всіх можливих імплікант для булевої функції f з розглянутого прикладу дає можливість переконатися, що простих імплікант всього дві: g_3 і g_5. Отже, скорочена ДНФ функції f має вигляд:

$$f = g_3 \vee g_5 = x_1x_2 \vee x_2x_3$$

Як видно з табл. 11.2., імпліканти g_3 і g_5 в сукупності покривають своїми одиницями всі одиниці функції f. Отримання скорочених ДНФ є першим етапом відшукання мінімальних форм булевих функцій. В скорочену ДНФ входять всі прості імпліканти булевої функції. Іноді з скороченої ДНФ можна прибрати одну або кілька простих імплікант, не порушуючи еквівалентності вихідної функції. Такі прості імпліканти назвемо зайвими. Вилучення зайвих простих імплікант зі скорочених ДНФ - другий етап мінімізації.

Визначення. Скорочена ДНФ булевої функції називається **тупиковою**, якщо в ній відсутні зайві прості імпліканти.

Усунення зайвих простих імплікант зі скороченої ДНФ булевої функції не є однозначним процесом, тобто булева функція може мати кілька тупикових ДНФ.

Твердження. Тупикові ДНФ булевої функції f, що містять мінімальну кількість літер, є мінімальними. Мінімальних ДНФ теж може бути кілька.

Розглянемо кілька методів мінімізації.

Метод Квайна ґрунтується на застосуванні двох основних співвідношень.

1. Співвідношення склеювання

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

де *A* — будь який елементарний добуток.

2. Співвідношення поглинання

$$A\tilde{x} \vee A = A, \tilde{x} \in \{x, \bar{x}\}$$

Справедливість обох співвідношень легко перевіряється.

Теорема Квайна. Якщо в СДНФ булевої функції виконати всі можливі склеювання і поглинання, то в результаті буде отримана скорочена ДНФ.

Метод застосуємо до досконалої ДНФ. Зі співвідношення поглинання випливає, що довільний елементарний добуток поглинається будь-якою його частиною.

Мінімізація за методом Квайна виконується по наступному алгоритму:

1. Виконуються всі склеювання в ДДНФ.

2. Виконуються всі поглинання.

3. Результуюча функція перевіряється на можливість подальшого склеювання і поглинання.

4. Після отримання скороченої ДНФ будується імплікантна матриця, за якою знаходяться "зайві" імпліканти.

Приклад. Нехай є булева функція, задана таблицею істинності (табл 11.3). Її ДДНФ має вигляд:

$$f= \bar{x}_1\bar{x}_2x_3x_4 \vee \bar{x}_1\bar{x}_2x_3x_4 \vee \bar{x}_1x_2\bar{x}_3x_4 \vee \bar{x}_1x_2x_3x_4 \vee x_1x_2x_3\bar{x}_4 \vee x_1x_2x_3x_4$$

1 2 3 4 5 6

Для зручності викладу помітимо кожну констітуенту одиниці з ДДНФ функції f будь-яким десятковим номером (довільно). Виконуємо склеювання. Конституента 1 склеюється тільки з конституентою 2 (по змінній x_3) і з конституентою 3 (по змінній x_2) конституента 2 з конституентою 4 і т. д.

Таблиця 11.3.

x_1	x_2	x_3	x_4	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

В результаті виконання операції склеювання отримуємо:

1-2: $\bar{x}_1\bar{x}_2x_4$

1-3: $\bar{x}_1\bar{x}_3x_4$

2-4: $\bar{x}_1x_3x_4$

3-4: $\bar{x}_1x_2x_4$

4-6: $x_2x_3x_4$

5-6: $x_1x_2x_3$

Результатом склеювання є завжди елементарна кон'юнкція, що представляє собою загальну частину склеюваних конституант.

Далі проводимо склеювання одержуваних елементарних кон'юнкцій. Склеюються тільки ті кон'юнкції, які містять однакові змінні. Має місце два випадки склеювання:

$$\bar{x}_1\bar{x}_2x_4 \vee \bar{x}_1x_2x_4 = \bar{x}_1\bar{x}_2x_4 \vee \bar{x}_1x_2x_4 \vee \bar{x}_1x_4$$

$$\bar{x}_1\bar{x}_3x_4 \vee \bar{x}_1x_3x_4 = \bar{x}_1\bar{x}_3x_4 \vee \bar{x}_1x_3x_4 \vee \bar{x}_1x_4$$

з появою однієї і тієї ж елементарної кон'юнкції $\overline{x}_1 x_4$ подальші склеювання неможливі. Зробивши поглинання (з отриманої ДНФ викреслюємо всі елементарні кон'юнкції, які поглинаються), отримаємо скорочену ДНФ:

$x_2 x_3 x_4 \vee x_1 x_2 x_3 \vee \overline{x}_1 x_4$

Переходимо до наступного етапу. Для отримання мінімальної ДНФ необхідно видалити зі скороченою ДНФ всі зайві прості імпліканти. Це робиться за допомогою спеціальної імплікантної матриці Квайна. Рядки такої матриці відзначаються простими імплікантами булевої функції, тобто членами скороченої ДНФ, а стовпці - конституентами одиниці, тобто членами ДДНФ булевої функції.

Приклад (продовження). Імплікантна матриця має вид (табл. 11.4).

Таблиця 11.4

Прості імпліканти	Констітуєнти одиниці					
	$\overline{X}_1 \overline{X}_2 \overline{X}_3 X_4$	$\overline{X}_1 \overline{X}_2 X_3 X_4$	$\overline{X}_1 X_2 \overline{X}_3 X_4$	$\overline{X}_1 X_2 X_3 X_4$	$X_1 X_2 X_3 \overline{X}_4$	$X_1 X_2 X_3 X_4$
$\overline{X}_1 X_4$	X	X	X	X		
$X_2 X_3 X_4$				X		X
$X_1 X_2 X_3$					X	X

Як вже зазначалося, проста імпліканта поглинає деяку констітуенту одиниці, якщо є її власною частиною. Відповідна клітина імплікантної матриці на перетині рядка (з розглянутої простої імпліканти) і стовпця (з констітуентою одиниці) відзначається хрестиком (табл. 11.4). Мінімальні ДНФ будуються за імплікантною матрицею наступним чином:

1) знаходяться стовпці імплікантної матриці, що мають тільки один хрестик. Відповідні цим хрестиками прості імпліканти називаються ***базисними*** і складають так зване **ядро булевої функції**. Ядро обов'язково входить в мінімальну ДНФ.

2) розглядаються різні варіанти вибору сукупності простих імплікант, які накриють хрестиками інші стовпчики імплікантної матриці, і вибираються варіанти з мінімальним сумарним числом букв в такій сукупності імплікант.

Приклад (продовження). Ядром нашої функції є імпліканти $x_1 x_2 x_3$ і $\overline{x}_1 x_4$. Імпліканта $x_2 x_3 x_4$ - зайва, оскільки ядро накриває всі стовпці імплікантної матриці. Тому функція має єдину тупикову і мінімальну ДНФ:

$$f_{min} = x_1 x_2 x_3 \vee \overline{x}_1 x_4$$

Слід зазначити, що число N хрестиків в одному рядку завжди є ступенем 2. Більш того, можна легко переконатися в тому, що $N = 2^{n-k}$, де k - число букв, що містяться в простій імпліканті.

11.3 Індивідуальні завдання

1. Булеві функції задані таблично. Задать їх аналітично: f1- в СДНФ, f2 - в СКНФ. Перевести f1 в форму КНФ, використовуючи правило де-Моргана.

Таблица 11.5

			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
x_1	x_2	x_3	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f
0	0	0	0	1	0	0	1	0	1	1	1	1	1	0	1	1	0	1	0	1	0	0
0	0	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	0	1	0	1	1	1
0	1	0	0	1	1	0	0	1	1	0	1	1	0	1	0	0	1	0	0	0	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	1	1	0	1	1	1	0	0	0	1	0	1	0	1	1	1	1	1	0	1	1
1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1
1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

2. Задати функцію аналітично (СДНФ):

Таблица 11.6

	$x_4x_5x_6$							
$x_1x_2x_3$	000	001	010	011	100	101	110	111
000	1	0	0	0	0	0	0	1
001	0	0	1	0	0	0	0	0
010	0	0	0	0	0	0	0	0
011	1	0	0	0	0	0	1	0
100	0	0	1	0	0	1	0	0
101	0	0	0	0	0	0	0	0
110	1	0	0	0	1	0	0	1
111	0	0	0	0	0	0	0	0

3. Задати функцію таблично:

а) f(x)= $\overline{X}_1 X_2 X_3 \overline{X}_4 \vee \overline{X}_1 \overline{X}_2 X_3 X_4 \vee X_1 \overline{X}_2 \overline{X}_3 X_4 \vee \overline{X}_1 \overline{X}_2 \overline{X}_3 X_4$.

б) f(x)= $\overline{(X_1 \vee \overline{X}_2) \wedge \overline{X_2 \wedge X_3}}$

4. Примінити правило де-Моргана до наступних виразів:

$$f = xyz \vee x\overline{y}\overline{z} \vee \overline{x}\overline{y}z$$

$$f(x) = (x_1 \wedge x_2 \wedge \overline{x}_3) \wedge (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \wedge (\overline{x}_1 \wedge x_2 \wedge x_3).$$

5. Виконати операції склеювання

$$xyz \vee xy\overline{z} =$$

$$xyz \vee x\overline{y}z =$$

$$\overline{x}\overline{y}\overline{z} \vee x\overline{y}\overline{z} =$$

$$\overline{x}\overline{y}z \vee x\overline{y}z =$$

$$(x \vee y \vee z) \wedge (x \vee \overline{y} \vee z) =$$

$$(\overline{x} \vee y) \wedge (x \vee y) =$$

6. Виконати операції поглинання:

$$xyz \vee xy =$$

$$(\overline{x} \vee \overline{y} \vee \overline{z}) \wedge (\overline{x} \vee \overline{y}) =$$

$$x\overline{y}\overline{z} \vee x\overline{y} =$$

$$(x \vee \overline{y} \vee z) \wedge (\overline{y} \vee z) =$$

7. Перетворити функцію: $f(x) = x_1 \vee \overline{x}_2 \wedge x_2$

8. Перетворити функцію таким чином, щоб використовувались лише операції дизюнкції та інверсії:

$$f(x) = \overline{x}_1\overline{x}_2x_3 \vee x_1x_2 \vee x_1\overline{x}_2x_4 \vee \overline{x}_2x_4$$

9. Мінімізувати функцію, використавши метод Квайна.

$$f(x) = \overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4 \vee \overline{x}_1\overline{x}_2\overline{x}_3x_4 \vee x_1\overline{x}_2\overline{x}_3x_4 \vee \overline{x}_1x_2x_3x_4 \vee$$
$$\vee x_1x_2x_3x_4 \vee \overline{x}_1x_2\overline{x}_3\overline{x}_4$$

11.4 Контрольні запитання

1) Дайте визначення булевої змінної, булевої функції.
2) Які способи завдання булевих функцій Вам відомі?
3) Сформулюйте основні закони і тотожності булевої алгебри.
4) Дайте визначення конституенти одиниці (нуля), простої імпліканти, (імпліценти).
5) Дайте визначення ДКНФ і ДДНФ.
6) Дайте визначення скороченої, тупикової, мінімальної ДНФ, (КНФ).
7) Сформулюйте теорему Квайна.
8) Сформулюйте алгоритм мінімізації булевих функцій по методу Квайна.

12 МІНІМІЗАЦІЯ БУЛЕВИХ ФУНКЦІЙ ЗА МЕТОДОМ КВАЙНА-МАК-КЛАСКІ

12.1 Мета заняття

Закріпити вивчення основних понять і тотожностей булевої алгебри. Придбати навички мінімізації булевих функцій за методом Квайна-Мак-Класкі.

12.2 Короткі основні відомості з теорії та вирішення типових завдань

Метод Квайна-Мак-Класкі

Метод являє собою формалізований на етапі знаходження простих імплікант метод Квайна. Формалізація проводиться таким чином:

1. Всі конституенти одиниці з СДНФ булевої функції f записуються їх двійковими номерами.

2. Всі номери розбиваються на непересічні групи. Ознака належності до i-ї групи: i одиниць у кожному двійковому номері констітуенти одиниці.

3. Склеювання проводять тільки між номерами сусідніх груп. Склеювані номера відзначаються яким-небудь знаком (закресленням, зірочкою і т.д.).

4. Склеювання проводять всі можливі, як і в методі Квайна. Невідмічені після склеювання номери є простими імплікантами.

Знаходження мінімальних ДНФ далі проводиться по імплікантній матриці, як і в методі Квайна. Більш докладно розглянемо метод на прикладі вирішення наступного завдання.

Задача 11.1.

Задана булева функція чотирьох змінних $f(x_1,x_2,x_3,x_4)$ яка приймає одиничне значення на двійкових наборах з номерами (3,4,5,7,9,11,12,13). На інших наборах від 0 до 15 функція приймає нульове значення.

Потрібно:

1) Представити задану функцію в табличній формі.

2) Записати задану функцію в досконалій ДНФ і досконалій КНФ.

3) Мінімізувати задану функцію по методу Квайна-Мак-Класкі (знайти МДНФ).

4) Мінімізувати задану функцію за допомогою діаграм Вейча і знайти МДНФ і МКНФ.

5) Мінімізувати задану функцію за допомогою карт Карно і знайти МДНФ І МКНФ.

6)

Рішення

1) Представляємо задану функцію в табличній формі:

Таблиця 12.1.

Номер набора	x_1	x_2	x_3	x_4	f
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

2) Записуємо задану функцію в ДДНФ (запис по «1»):

$$f_{ДДНФ}(x_1,x_2,x_3,x_4)=\bar{x}_1\bar{x}_2x_3x_4\vee\bar{x}_1x_2\bar{x}_3\bar{x}_4\vee\bar{x}_1x_2\bar{x}_3x_4\vee\bar{x}_1x_2x_3x_4\vee x_1\bar{x}_2\bar{x}_3x_4\vee$$
$$\vee x_1\bar{x}_2x_3x_4\vee x_1x_2\bar{x}_3\bar{x}_4\vee x_1x_2\bar{x}_3x_4$$

Записуємо задану функцію в ДКНФ (запис по «0»):

$$f_{ДКНФ}(x_1,x_2,x_3,x_4)=(x_1\vee x_2\vee x_3\vee x_4)\wedge(x_1\vee x_2\vee x_3\vee\bar{x}_4)\wedge(x_1\vee x_2\vee\bar{x}_3\vee x_4)\wedge$$
$$\wedge(x_1\vee\bar{x}_2\vee\bar{x}_3\vee x_4)\wedge(\bar{x}_1\vee x_2\vee x_3\vee x_4)\wedge(\bar{x}_1\vee x_2\vee\bar{x}_3\vee x_4)\wedge(\bar{x}_1\vee\bar{x}_2\vee\bar{x}_3\vee x_4)\wedge$$
$$\wedge(\bar{x}_1\vee\bar{x}_2\vee\bar{x}_3\vee\bar{x}_4)$$

3) Задача мінімізації булевих функцій полягає в знаходженні аналітичного виразу заданої булевої функції у формі, що містить мінімально можливе число букв. Вираз в ДДНФ містить 32 букви, і в ДКНФ- 32 букви.

Мінімізація по методу Квайна-Мак-Класкі складається з наступних етапів:

1) Усі мін терми (імпліканти) з ДДНФ булевої функції f записуються їхніми двійковими номерами.

2) Усі номери розбиваються на непересічні групи. Ознака утворення і-ї групи – і одиниць у кожному двійковому номері мінтерма.

3) Виконується склеювання тільки між номерами сусідніх груп. Номери, що склеюються, відзначаються яким-небудь знаком, наприклад, закреслюванням.

4) Склеювання роблять усілякі, як і в методі Квайна. Невідмічені після склеювання номери є простими імплікантами. Одержання скороченої ДНФ.

5) Після одержання скороченої ДНФ будується імплікантна матриця, по якій знаходяться базисні (істотні) імпликанти і викреслюються «зайві» стовпці.

6) Знаходяться тупикові форми з яких вибираються мінімальні ДНФ .

Стосовно до заданої функції ці етапи будуть виглядати в такий спосіб:

1) $f(x_1,x_2,x_3,x_4)=0011 \vee 0100 \vee 0101 \vee 0111 \vee 1001 \vee 1011 \vee 1100 \vee$
$\vee$ 1101

2) Утворюємо групи двійкових номерів (див. табл. 12.2)

Таблиця 12.2

Номер групи	Двійкові номери мінтермів
0	-
1	0100
2	0011, 0101, 1001, 1100
3	0111, 1011, 1101
4	-

3) Склеюємо номери із сусідніх груп табл. 12.2

Номери, що склеюються, відзначаємо викреслюванням. На місці розрядів, по яких відбулося склеювання, ставимо *(або –X). Результати склеювання занесемо в табл. 12.3

Таблиця 12.3

Номер групи і	Двійкові номери імплікант
1(1-2)	010*,*100
2(2-3)	0*11,*011,01*1,*101,10*1,1*01,110*

Продовжимо склеювання сусідніх номерів групи в табл. 12.3. Склеюватися можуть тільки номери, що мають зірочки в однакових позиціях. Результати склеювання занесемо в табл. 12.4.

Таблиця 12.4

Номер групи i	Двійкові номери імплікант
1	*10*
2	0*11,*011,01*1,10*1,1*01

4) Подальше склеювання неможливе. З табл. 12.4. маємо 6 простих імплікант.

$g^1 = *10* = x_2\bar{x}_3$

$g^2 = 0*11 = \bar{x}_1 x_3 x_4$

$g^3 = *011 = \bar{x}_2 x_3 x_4$

$g^4 = 01*1 = \bar{x}_1 x_2 x_4$

$g^5 = 10*1 = x_1\bar{x}_2 x_4$

$g^6 = 1*01 = x_1\bar{x}_3 x_4$

Скорочена ДНФ функції $f(x_1, x_2, x_3, x_4)$

f= $g_1 \vee g_2 \vee g_3 \vee g_4 \vee g_5 \vee g_6$

5) Для знаходження базисних (істотних) імплікант будуємо таблицю Квайна (імплікантну матрицю) (див. табл. 12.5).

Таблиця 12.5

Двійкові номери простих імплікант	Двійкові номера мінтермів							
	0011	0100	0101	0111	1001	1011	1100	1101
10		▲					▲	▲
0*11	▲			▲				
*011	▲					▲		
01*1			▲	▲				
10*1					▲	▲		
1*01					▲			▲

Розставляємо мітки: якщо в деякий мінтерм СДНФ входить будь яка із простих імплікант, то на перетинанні відповідного стовпця і рядка ставиться мітка ▲.

Для знаходження базисних (істотних) імплікант шукаються стовпці в імплікантній матриці (табл. 12.5), що мають тільки одну мітку. Відповідні до цих міток прості імпліканти називаються базисами (істотними) і становлять ядро булевої функції. Ядро обов'язково входить у мінімальну ДНФ.

У нашому випадку базисна імпліканта одна:

$$g_1 = *10* = x_2 \bullet \overline{x_3}$$

Викреслюємо зайві стовпці. Зайвими є стовпці, що відповідають базисним імплікантам (g_1).

У підсумку одержуємо нову таблицю (див. табл. 12.6).

Таблиця 12.6

Двійкові номери простих імплікант	Двійкові номери мінтермів			
	0011	0111	1001	1011
$g_2 = 0*11$	▲	▲		
$g_3 = *011$	▲			▲
$g_4 = 01*1$		▲		
$g_5 = 10*1$			▲	▲
$g_6 = 1*01$			▲	

6) З урахуванням ядра функції і табл. 12.6.

Находимо можливі тупикові форми:

$$f = g_1 \vee g_2 \vee g_5 \ (1)$$

$$f = g_1 \vee g_2 \vee g_3 \vee g_6 \ (2)$$

$$f = g_1 \vee g_3 \vee g_4 \vee g_5 \ (3)$$

$$f = g_1 \vee g_3 \vee g_4 \vee g_6 \ (4)$$

Тупикова форма (1) і є мінімальна, тому що містить найменше число букв. Отже, МДНФ вихідної функції:

$$f(x_1, x_2, x_3, x_4) = (x_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \vee x_3 \wedge x_4) \vee (x_1 \vee \overline{x}_2 \vee x_4) \text{ (8 букв).}$$

12.3 Індивідуальні завдання

1. Булеві функції задані таблично. Задать їх аналітично та виконати мінімізацію з використаням алгоритму Квайна-Мак-Класкі.

			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
x_1	x_2	x_3	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f
0	0	0	0	1	0	0	1	0	1	1	1	1	1	0	1	1	0	1	0	1	0	0
0	0	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	0	1	0	1	1	1
0	1	0	0	1	1	0	0	1	1	0	1	1	0	1	0	0	1	0	0	0	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	1	1	0	1	1	1	0	0	0	1	0	1	0	1	1	1	1	1	0	1	1
1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1
1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

12.3 Контрольні запитання

9) Дайте визначення булевої змінної, булевої функції.
10) Які способи завдання булевих функцій Вам відомі?
11) Сформулюйте основні закони і тотожності булевої алгебри.
12) Дайте визначення конституенти одиниці (нуля), простої імпліканти, (імпліценти).
13) Дайте визначення ДКНФ і ДДНФ.
14) Дайте визначення скороченої, тупикової, мінімальної ДНФ, (КНФ).
15) Сформулюйте теорему Квайна.
16) Сформулюйте алгоритм мінімізації булевих функцій по методу Квайна.
17) Сформулюйте алгоритм мінімізації булевих функцій по методу Квайна-Мак-Класки.

13 МІНІМІЗАЦІЯ БУЛЕВИХ ФУНКЦІЙ ЗА МЕТОДОМ ДІАГРАМ ВЕЙЧА ТА КАРТ КАРНО

13.1 Мета заняття

Вивчити основні поняття і тотожності булевої алгебри. Придбати навички мінімізації булевих функцій за методом діаграм Вейча та карт Карно.

13.2 Короткі основні відомості з теорії та вирішення типових завдань

13.2.1 Мінімізація булевих функцій за допомогою карт Карно

Розглянемо візуальний метод мінімізації булевих функцій за допомогою карт (діаграм) Карно, який є одним з найбільш зручних методів при невеликому числі змінних (до 6). Даний метод був розроблений в 1953 р. американським вченим Морісом Карно.

Карта Карно є *координатним* способом представлення булевих функцій. При цьому способі завдання таблиця істинності функції представляється у вигляді координатної карти станів, яка містить 2^n кліток (по числу вхідних наборів булевої функції n змінних). Змінні функції розбиваються на дві групи так, що одна група визначає координати стовпця карти, а інша - координати рядка. При такому способі побудови кожна клітка визначається значеннями змінних, що відповідають певному двійковому набору. Усередині кожної клітки карти Карно ставиться значення функції на даному наборі. Змінні в рядках і стовпцях розташовуються так, щоб сусідні клітки карти Карно *різнилися тільки в одному розряді змінних,* тобто були сусідніми. Тому значення змінних у стовпцях і в рядках карти утворюють сусідній код Грея. Такий спосіб представлення дуже зручний для наочності при мінімізації булевих функцій. Карта Карно 4 змінних, з погляду визначення "сусідства" змінних, геометрично являє собою просторову фігуру тор або, простіше кажучи, "бублик".

x_1 \ x_2x_3	00	01	11	10
0	000	001	011	010
1	100	101	111	110

а)

x_1x_2 \ x_3x_4	00	01	11	10
00	0000	0001	0011	0010
01	0100	0101	0111	0110
11	1100	1101	1111	1110
10	1000	1001	1011	1010

б)

Рис. 13.1. Карти Карно: а) функції 3-х змінних; б) функції 4-х змінних.

Правила мінімізації з використанням карт Карно

1. У карті Карно групи одиниць (для одержання ДНФ) і групи нулів (для одержання КНФ) необхідно обвести чотирикутними контурами. Усередині контуру повинні перебувати тільки однойменні значення функції. Цей процес відповідає операції *склеювання* або знаходження імплікант даної функції.

2. Кількість кліток всередині контуру повинна бути цілим ступенем двійки (1, 2, 4, 8, 16...).

3. При проведенні контурів крайні рядки карти (верхні і нижні, ліві і праві), а також кутові клітки, уважаються сусідніми (для карт до 4-х змінних).

4. Кожний контур повинен включати максимально можливу кількість кліток. У цьому випадку він буде відповідати простій імпліканті.

5. Усі одиниці (нулі) у карті (навіть одиночні) повинні бути охоплені контурами. Будь-яка одиниця (нуль) може входити в контури довільну кількість разів.

6. Множини контурів, що покривають усі 1 (0) функції утворюють тупикову ДНФ (КНФ). *Метою мінімізації* є знаходження мінімальної з множини тупикових форм.

7. В елементарній кон'юнкції (диз'юнкції), яка відповідає одному контуру, залишаються тільки ті змінні, значення яких не змінюється усередині обведеного контуру. Змінні булевої функції входять в елементарну кон'юнкцію (для значень функції 1) без інверсії, якщо їх значення на відповідних координатах рівно 1 і з інверсією - якщо 0. Для значень булевої функції, рівних 0, записуються елементарні диз'юнкції, куди змінні входять без інверсії, якщо їх значення на відповідних координатах рівно 0 і з інверсією - якщо 1.

8. Кількість змінних, які залишаються в елементарній кон'юнкції (диз'юнкції), визначаються наступним чином:

$m=n-\log_2 M$,

де n- число змінних, M- число наборів, що склеюються (число одиниць в покритті).

13.2.2 Мінімізація булевих функцій за допомогою діаграм Вейча

Іноді зручно використовувати діаграми Вейча. В основі методу лежить завдання булевих функцій діаграмами деякого спеціального виду, що одержали назву діаграм Вейча. Приклад діаграма Вейча для функції 2-х змінних представлений на рас. 13. 2 а). Кожна клітина діаграми відповідає набору змінних булевої функції в її таблиці істинності. У клітці діаграми Вейча ставиться одиниця, якщо булева функція приймає одиничне значення на відповідному наборі. у тому випадку, коли мінімізація відбувається для диз’юнктивної форми, нульові значення булевої функції в діаграмі Вейча не ставляться. Для булевої функції трьох змінних діаграма Вейча має наступний вигляд (Рис. 13.2 б).).

	x_1	$\bar{x}_1$
x_2	11	01
$\bar{x}_2$	10	00

а)

	x_1		$\bar{x}_1$	
x_2	110	111	011	010
$\bar{x}_2$	100	101	001	000
	$\bar{x}_3$	x_3	x_3	$\bar{x}_3$

б)

	x_2		$\bar{x}_2$		
x_1	1100	1101	1001	1000	$\bar{x}_3$
x_1	1110	1111	1011	1010	x_3
$\bar{x}_1$	0110	0111	0011	0010	x_3
$\bar{x}_1$	0100	0101	0001	0000	$\bar{x}_3$
	$\bar{x}_4$	x_4	x_4	$\bar{x}_4$	

в)

Рис. 13.2. Діаграма Вейча: а) функції 2-х змінних; б) функції 3-х змінних; в) функції 4-х змінних.

Додавання до неї ще такої ж таблиці дає діаграму для функції 4-х змінних (рис. 13.2 в)). Таким же чином, тобто приписуванням ще однієї діаграми 4-х змінних, до тільки що розглянутої, можна одержати діаграму для функції 5-ти змінних і т. д.

Правила мінімізації булевих функцій із застосуванням діаграм Вейча такі самі, як і для карт Карно.

13.2.3 Особливості мінімізації булевих функцій з великим числом змінних

Якщо кількість змінних п’ять або більше, відобразити графічно функцію у вигляді однієї плоскої карти неможливо. У таких випадках будують комбіновану карту, що складається із сукупності більш простих базових карт, наприклад карт для функції 4-х змінних. Процедура мінімізації в цьому випадку полягає в тому, що спочатку знаходять мінімальні форми всередині базових карт, а потім, розширюючи поняття сусідніх клітин, знаходять мінімальні покриття для сукупності карт. Сусідніми клітинами є клітини, що збігаються при накладанні базових карт один на одного. Приклади карт Карно для булевих функцій 5-ти і 6-ти змінних представлені на рис. 13.3. і 13.4. відповідно:

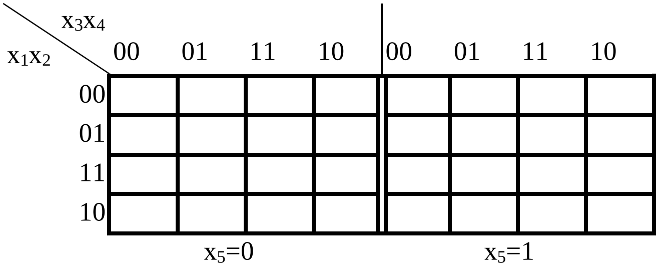


Рис. 13.3. Карта Карно для булевої функції 5-ти змінних.

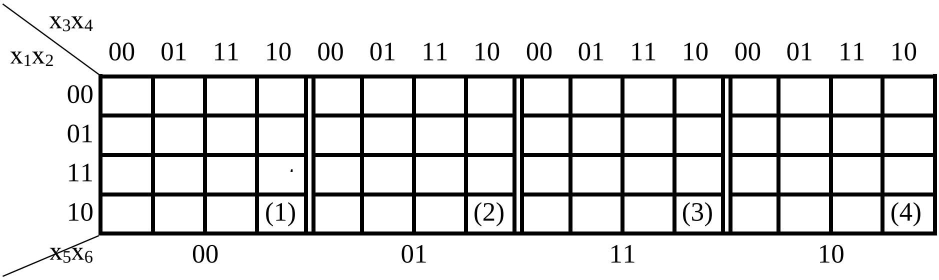


Рис. 13.4. Карта Карно для булевої функції 6-ти змінних.

За рис. 13.4. Можна зробити висновок, що сусідніми є для 1-ї базової карти - 2-а і 4-а; для 2-ї - 1-а і 3-я; для 3-ї 2-а і 4-а; для 4 - ї - 1-а і 3-я.

При збільшенні кількості змінних на одну, площа карти збільшується в два рази - до неї добавляється ще така ж карта. При цьому нова змінна дорівнює 1 на новій карті, і 0 на тій, яка була раніше.

13.2.4. Мінімізація частково визначених булевих функцій

У реальних задачах дуже часто буває так, що значення булевої функції на деяких наборах не визначено і може довизначити довільно. Вихідні сигнали на цих наборах можуть приймати будь-які значення - 0 або 1. Вхідні набори, які дають невизначене значення функції називаються забороненими. При синтезі схем, що реалізують неповністю визначені функції вихідним сигналам, що відповідають забороненим наборам, надають такі значення, при яких можна побудувати найбільш просту схему. У цьому випадку довизначення функції було б доцільно проводити таким чином, щоб її мінімальна нормальна форма мала найменше число букв з усіх можливих варіантів довизначення.

Алгоритм пошуку мінімальної ДНФ частково визначеної функції f можна представити таким чином.

1. Знайти будь-яким відомим способом скорочену ДНФ функції, довизначивши одиницями вихідні значення функції f на всіх невизначених наборах.

2. Вибрати мінімальну ДНФ по імпликантній матриці, де в стовпцях виписані лише ті конституенти одиниці функції f, які відповідають повністю визначеним одиничним наборам.

Аналогічний алгоритм (з довизначенням нульовими наборами) може бути запропонований для пошуку КНФ. При цьому довизначення таблиці істинності функції f може бути виконано по різному для КНФ і ДНФ.

Зауважимо, що для вирішення розглянутої задачі практично достатньо тих навиків, які були отримані при мінімізації повністю визначених булевих функцій безпосередньо по діаграмі Вейча. У випадках, коли мінімальних форм кілька, наводиться одна з них.

Розглянемо простий приклад. Функція задана діаграмою Вейча, представленої (Рис. 13.5)

	x_1			
x_2	0	–	–	1
	1	1	–	0
		x_3		

Рис. 13.5.

Довизначення функції на невизначених наборах одиницями (Рис. 13. 6).

	x_1			
x_2	0	**1**	**1**	1
	1	1	**1**	0
		x_3		

Рис. 13.6.

$$f_{min}(x_1, x_2, x_3) = x_3 \vee x_1\overline{x_2} \vee \overline{x_1}x_2 \qquad (13.1)$$

Довизначення функції на невизначених наборах нулями (Рис. 13. 7).

	x_1			
x_2	0	**0**	**0**	1
	1	1	**0**	0
		x_3		

Рис. 13.7.

$$f_{min}(x_1, x_2, x_3) = x_1\overline{x_2} \vee \overline{x_1}x_2\overline{x_3} \qquad (13.2)$$

Різні довизначення призводять до різних мінімальних ДНФ. Однак більш проста мінімальна ДНФ виходить, якщо зробити довизначення так, як це зроблено на діаграмі Вейча (Рис. 13. 8).

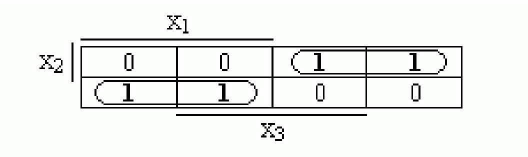


Рис. 13.8

$$f_{min}(x_1, x_2, x_3) = x_1\overline{x_2} \vee \overline{x_1}x_2 \qquad (13.3)$$

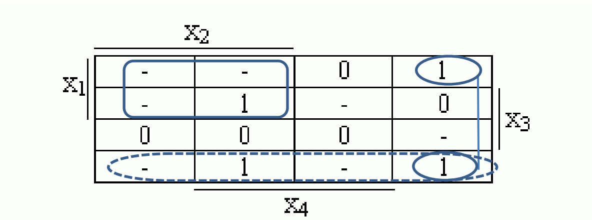
Алгоритм пошуку мінімальної ДНФ частково визначеної функції f можна представити таким чином:

- Знайти будь-яким відомим способом скорочену ДНФ функції, яку можна отримати довизначенням одиницями вихідної функції f на всіх невизначених наборах.
- Обрати мінімальну ДНФ по імпликантній матриці, де в стовпцях виписані лише ті конституенти одиниці функції f, які відповідають повністю визначеним одиничним наборам.

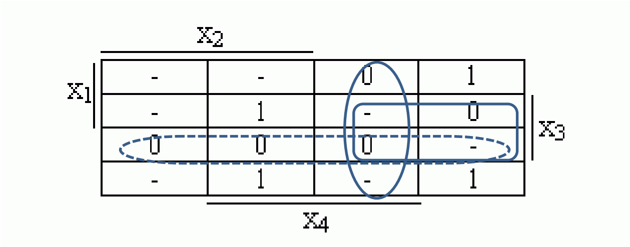
Аналогічний алгоритм (з довизначенням нульовими наборами) може бути запропонований для пошуку КНФ При цьому довизначення таблиці істинності функції f може бути зроблено по різному для КНФ і ДНФ.

Зауважимо, що для вирішення розглянутої задачі практично достатньо тих навиків, які були отримані при мінімізації повністю визначених булевих функцій безпосередньо по діаграмі Вейча. Наведемо приклад. У випадках, коли мінімальних форм кілька, наводиться одна з них.

Для функції, представленої на рис. 13.9.



а) - диз’юнктивна форма



б) - кон'юнктивна форма

Рис. 13.9

$$f_{ДНФ\,min}(x_1,x_2,x_3,x_4) = x_1x_2 \vee \overline{x}_1\overline{x}_3 \vee \overline{x}_3\overline{x}_4 \qquad (13.4)$$

$$f_{КНФ\,min}(x_1,x_2,x_3,x_4) = (x_1 \vee \overline{x}_3)(\overline{x}_3 \vee x_2)(x_2 \vee \overline{x}_4) \qquad (13.5)$$

Продовження рішення задачі 12.1

4) Мінімізуємо задану функцію за допомогою діаграми Вейча:
Покриття в діаграмі Вейча виконуємо по диз'юнктивній формі (по «1»).

Результат має вигляд:

	$\overline{x}_3$		x_3		
$\overline{x}_1$	0	0	1	0	$\overline{x}_2$
	1	1	1	0	x_2
x_1	1	1	0	0	
	0	1	1	0	$\overline{x}_2$
	$\overline{x}_4$	x_4		$\overline{x}_4$	

$$f_{ДНФ_{min}}(x_1,x_2,x_3,x_4) = x_2\overline{x}_3 \vee \overline{x}_1x_3x_4 \vee x_1\overline{x}_2x_4$$

Проведемо мінімізацію даною функції по кон'юнктивній формі (по «0»).

Результат має вигляд:

	$\overline{x}_3$		x_3		
$\overline{x}_1$	0	0	1	0	$\overline{x}_2$
	1	1	1	0	x_2
x_1	1	1	0	0	
	0	1	1	0	$\overline{x}_2$
	$\overline{x}_4$	x_4		$\overline{x}_4$	

$$f_{КНФ\,min}(x_1,x_2,x_3,x_4) =$$
$$(\overline{x}_3 \vee x_4)(x_2 \vee x_4)(x_1 \vee x_2 \vee x_3)(\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3)$$

5) Мінімізуємо задану функцію за допомогою карт Карно:

Мінімізація диз'юнктивної форми:

$x_1 x_2 \backslash x_3 x_4$	00	01	11	10
00	0	0	1	0
01	1	1	1	0
11	1	1	0	0
10	0	1	1	0

$$f_{ДНФ_{min}}(x_1, x_2, x_3, x_4) = x_2 \bar{x}_3 \vee \bar{x}_1 x_3 x_4 \vee x_1 \bar{x}_2 x_4$$

Мінімізація кон'юнктивної форми:

$x_1 x_2 \backslash x_3 x_4$	00	01	11	10
00	0	0	1	0
01	1	1	1	0
11	1	1	0	0
10	0	1	1	0

$$f_{КНФ_{min}}(x_1, x_2, x_3, x_4) =$$
$$(\bar{x}_3 \vee x_4)(x_2 \vee x_4)(x_1 \vee x_2 \vee x_3)(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$$

13.3 Індивідуальні завдання

Завдання 1.

Булева функція задана таблично (таблиця 13.3.1). Виконати мінімізацію диз'юнктивної форми функції за допомогою діаграм Вейча.

Таблиця 13.3.1.

№ набору					№ варіанта																			
	x_1	x_2	x_3	x_4	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
0	0	0	0	0		1			1			1			1			1			1			1
1	0	0	0	1	1	1			1	1	1		1		1	1	1	1			1	1	1	
2	0	0	1	0		1	1				1		1					1	1				1	
3	0	0	1	1		1		1		1			1		1	1		1		1		1		
4	0	1	0	0			1	1		1		1	1			1			1	1		1		1
5	0	1	0	1	1	1		1	1			1		1			1			1	1			1
6	0	1	1	0			1			1		1		1	1				1			1		1
7	0	1	1	1	1			1				1					1			1				1
8	1	0	0	0					1		1	1							1		1		1	1
9	1	0	0	1		1		1	1	1	1					1		1			1	1	1	
10	1	0	1	0			1				1		1			1			1		1		1	
11	1	0	1	1	1	1		1		1			1		1		1	1		1		1		
12	1	1	0	0					1		1			1							1		1	1
13	1	1	0	1	1		1							1	1		1		1					
14	1	1	1	0							1			1		1							1	
15	1	1	1	1	1		1	1					1	1		1	1			1				

Завдання 2.

Булева функція задана таблично (таблиця 13.3.2). Виконати мінімізацію кон'юнктивної форми функції за допомогою карт Карно.

Таблиця 13.3.2.

№ набору					№ варіанта																			
	x_1	x_2	x_3	x_4	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
0	0	0	0	0		0			0			0			0			0			0			0
1	0	0	0	1	0	0			0	0	0		0	0	0	0	0	0			0	0	0	
2	0	0	1	0		0	0				0		0					0	0				0	
3	0	0	1	1		0		0		0			0		0	0		0		0		0		
4	0	1	0	0			0	0	0	0		0	0			0			0	0		0		0
5	0	1	0	1	0	0		0	0			0		0			0			0	0			0
6	0	1	1	0		0			0			0			0			0			0			0
7	0	1	1	1	0	0			0	0	0		0		0	0	0	0			0	0	0	
8	1	0	0	0		0	0				0		0	0				0	0				0	
9	1	0	0	1	0	0		0		0			0		0	0		0		0		0		
10	1	0	1	0	0		0	0	0	0		0	0	0		0			0	0		0		0
11	1	0	1	1		0			0			0		0	0			0			0			0
12	1	1	0	0	0	0			0	0	0		0		0	0	0	0			0	0	0	
13	1	1	0	1		0	0				0		0					0	0				0	
14	1	1	1	0		0		0		0			0	0	0	0		0		0		0		
15	1	1	1	1	0		0	0	0	0		0	0	0		0			0	0		0		0

Завдання 3.

Булева функція задана таблично (таблиця 13.3.3.). Дана функція частково визначена:

* - на цих наборах функція не визначена;

1 – на цих наборах функція приймає значення 1;

На всіх інших наборах функція приймає значення 0.

Виконати мінімізацію за допомогою карт Карно.

Таблиця 13.3.3.

№ на-бора						№ варианта																			
	x_1	x_2	x_3	x_4		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
0	0	0	0	0	0	1		*		*	1		*			1	1	1	1	*	*			1	1
1	0	0	0	0	1		1		1					1	1		*			1		1	1		
2	0	0	0	1	0	*			*		1	1		*			1	1			*		*	*	1
3	0	0	0	1	1		1					1	*			1				1		*			
4	0	0	1	0	0	1	*		1	*	1			1	*	1			1			1		1	*
5	0	0	1	0	1			1	1	1	1	1	*	1		*	1	*			1	1	1	1	1
6	0	0	1	1	0		1				1				*		*		*	*				1	
7	0	0	1	1	1	*			*	1	1	1		*			1					*	1	1	1
8	0	1	0	0	0	*	1			*	1	*					1	1		1				1	1
9	0	1	0	0	1			*			*				1	1							*		*
10	0	1	0	1	0	1		*	1				1	1			1	1	1		*	1		*	1
11	0	1	0	1	1		*	1		1				*		*	*			*			1		
12	0	1	1	0	0						1	*	1		1			1	*				*	1	
13	0	1	1	0	1	1	*	1				1			*		1		1		1				1
14	0	1	1	1	0						*		1			*		*		*		*	1	*	
15	0	1	1	1	1	1		*				1				1			1						
16	1	0	0	0	0	1		1	1	1				1			1	1	1		1	1	1	*	1
17	1	0	0	0	1		1			1			1		1	1				1			1		*
18	1	0	0	1	0	*			1		*	*		1	*		1	1				1		*	1
19	1	0	0	1	1		1	*		*			*			*				1	*		*		
20	1	0	1	0	0	1	*		*			1		*		1		1	1					*	
21	1	0	1	0	1			1	1	1				1		1	1				1	1	1		1
22	1	0	1	1	0				1		1	1	1	1	*		*	*	*	1		1		1	
23	1	0	1	1	1		*				1					1							*	1	*
24	1	1	0	0	0				1	1	*						1	1		*	*	1	1		1
25	1	1	0	0	1	*	1			1			1	*	1	1				1			1		
26	1	1	0	1	0			1	1			*					1	1			1	1		1	1
27	1	1	0	1	1		1						*				*	*	*	1					
28	1	1	1	0	0					1	*			1	1			1				*	1	*	
29	1	1	1	0	1		1	1		*		1			1					1	1				1
30	1	1	1	1	0	1		*	*		1	1	1				1	1	1		*		*	1	
31	1	1	1	1	1	1		1	*		1	1			1	*			1	*				1	*

13.3 Контрольні запитання

1) Сформулюйте правила мінімізації булевих функцій за допомогою діаграм Вейча.
2) Сформулюйте правила мінімізації булевих функцій за допомогою карт Карно.
3) Які особливості мінімізація булевих функцій багатьох змінних Вам відомі?.
4) Накресліть карту Карно для булевої функції 6 змінних.

Які особливості мінімізація не повністю визначених булевих функцій Вам відомі?

14 ПРОЕКТУВАННЯ КОМБІНАЦІЙНИХ СХЕМ У БУЛЕВОМУ БАЗИСІ

14.1 Мета заняття

Вивчити методи і придбати практичні навички проектування комбінаційних схем в булевому базисі.

14.2 Короткі основні відомості з теорії та вирішення типових завдань

Комбінаційна схема являє собою цифровий автомат, реалізований на логічних елементах, що утворюють функціонально повну систему булевих функцій (ФПСБФ) [1,3,4,7,8,11].

ФПСБФ називається сукупність таких булевих функцій *(f_1, f_2,…, fk)*, що довільна булева функція *f* може бути записана у вигляді формули через функції цієї сукупності.

Сукупність булевих функцій *(f_1, f_2,…, fk), що* утворюють ФПСБФ прийнято називати базисом.

Прикладами ФПСБФ можуть служити системи, що використовують наступні базиси:

1) {І,АБО,НЕ} – булевий базис;
2) { І-НЕ} – монофункціональний базис;
3) { АБО-НЕ}- монофункціональний базис;
4) {І, $\oplus$,НЕ};
5) {І, $\oplus$,1}.

Найбільший інтерес представляють перші три базиси, тому що в електронному виді елементи, що їх реалізують, широко поширені.

У загальному випадку КС містить n входів і m виходів (див. рис. 14.1)

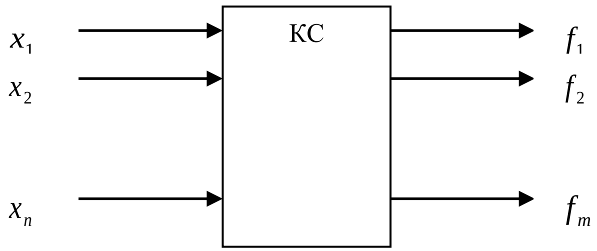


Рисунок 14.1

Для КС характерно відсутність зворотних зв'язків, тобто з'єднань виходів логічних елементів з їхніми входами, можливо через ланцюжок інших логічних елементів.

Це значить, що КС не містить елементів пам'яті і стан її виходів однозначно визначається станом входів у цей момент часу.

Результатом проектування КС повинна бути функціональна схема, на якій показані у вигляді умовних графічних зображень логічні елементи і взаємозв'язки між ними.

Процес проектування КС включає наступні етапи:

1) складання таблиці істинності, у якій перебираються всі можливі комбінації вхідних і вихідних змінних;

2) одержання ДДНФ або ДКНФ;

3) одержання МДНФ або МКНФ одним з методів;

4) складання функціональної схеми, якщо використовується булевий базис, або перетворення МДНФ або МКНФ у відповідний базис, з наступним складанням схеми.

Спроектовану КС, як технічний пристрій, прийнято характеризувати цілим рядом показників, серед яких слід звернути увагу на такі показники, як:

- складність КС (кількість логічних елементів, загальна кількість входів логічних елементів, кількість умовних корпусів інтегральних схем (ІС));

- швидкодія, яка характеризується, як сума середніх часів затримки розповсюдження ($t_{затр.р.серед}$) логічних елементів, що утворюють самий довгий ланцюжок на шляху від входів до виходів КС.

Завдання: Спроектувати КС, що відтворює булеву функцію 4-х змінних $f(x_1,x_2,x_3,x_4)$, яка приймає одиничне значення на двійкових наборах з номерами (3, 4, 5, 7, 9, 11, 12, 13). На інших наборах від 0 до 15 функція приймає нульове значення. Завдання вирішити з використанням МДНФ і МКНФ. Мінімізацію виконати за допомогою карт Карно.

Функціональні схеми зобразити в булевом базисі (І,АБО,НІ) і в монофункціональних базисах (І-НІ) і (АБО-НІ).(Усього 6 схем).

Оцінити складність спроектованих схем у кількості логічних елементів, необхідних для їхньої реалізації.

Оцінити швидкодію спроектованих схем в одиницях середнього часу затримки поширення одного логічного елемента ($t_{затр.р.серед}$).

Указати схеми, що мають меншу складність і кращу швидкодію.

Рішення

Етап1. Складання таблиці істинності:

№ набору	Входи КС				Вихід КС
№	x_1	x_2	x_3	x_4	f
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

Етап 2. Отримуємо ДДНФ:

$$f(x_1, x_{2,} x_3, x_4) = \overline{x_1}\,\overline{x_2}x_3x_4 \vee \overline{x_1}x_2\overline{x_3}\,\overline{x_4} \vee \overline{x_1}x_2\overline{x_3}x_4 \vee \overline{x_1}x_2x_3x_4 \vee x_1\overline{x_2}\,\overline{x_3}x_4 \vee x_1\overline{x_2}x_3x_4 \vee x_1x_2\overline{x_3}\,\overline{x_4} \vee$$
$$\vee\, x_1x_2\overline{x_3}x_4$$

Етап 3. Отримуємо МДНФ за допомогою карти Карно

$x_1x_2 \setminus x_3x_4$	00	01	11	10
00	0	0	1	0
01	1	1	1	0
11	1	1	0	0
10	0	1	1	0

g_2

g_1

g_3

Рис.14. 2

$g_1 = x_2\overline{x_3}$; $g_2 = \overline{x_1}x_3x_4$; $g_3 = x_1\overline{x_2}x_4$ - прості імпліканти;

$f_{\min}(x_1, x_2, x_3, x_4) = x_2\overline{x_3} \vee \overline{x_1}x_3x_4 \vee x_1\overline{x_2}x_4$ - МДНФ в булевому базисі. Цьому виразу відповідає наступна функціональна схема (див. рис. 14.2.)

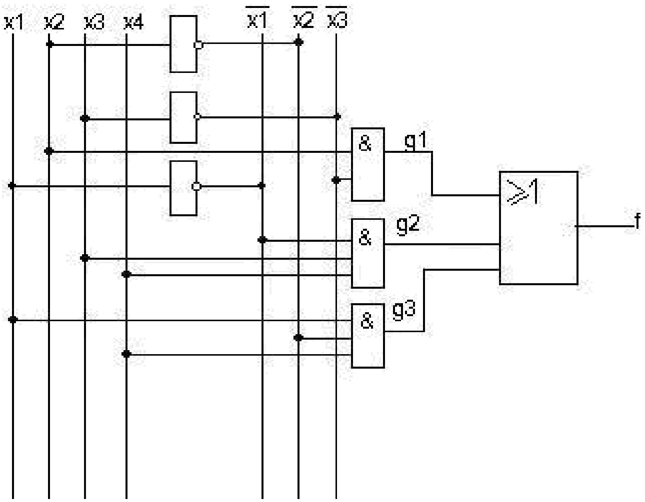


Рис.14.3

Складність схеми:

Потрібно:

3 логічних елемента НІ;

3 логічних елемента І (два елементи на три входи - 3І, один елемент на два входи - 2І);

1 логічний елемент АБО (на три входи – 3 АБО).

Усього потрібно 7 логічних елементів N_{KC1}=7 л. е.

Логічна глибина схеми L=3 (самий довгий шлях від входів до виходу). Якщо всі логічні елементи мають однакове $t_{затр.р.серед}$, то затримка спрацьовування розробленої схеми буде рівна: $t_{затр.кс1=}$ L*$t_{затр.р.серед}$ =3* $t_{затр.р.серед}$

Вирішимо цю ж задачу з використанням МКНФ. Для знаходження МКНФ скористаємося картою Карно:

$x_1x_2 \setminus x_3x_4$	00	01	11	10
00	0	0	1	0
01	1	1	1	0
11	1	1	0	0
10	0	1	1	0

g_3

g_2

g_1

g_2

g_4

g_2

g_2

Рис.14.4

Виписуємо імпліценти:

$$g_1 = \overline{x_3} \vee x_4$$

$$g_2 = x_2 \vee x_4$$

$$g_3 = x_1 \vee x_2 \vee x_3$$

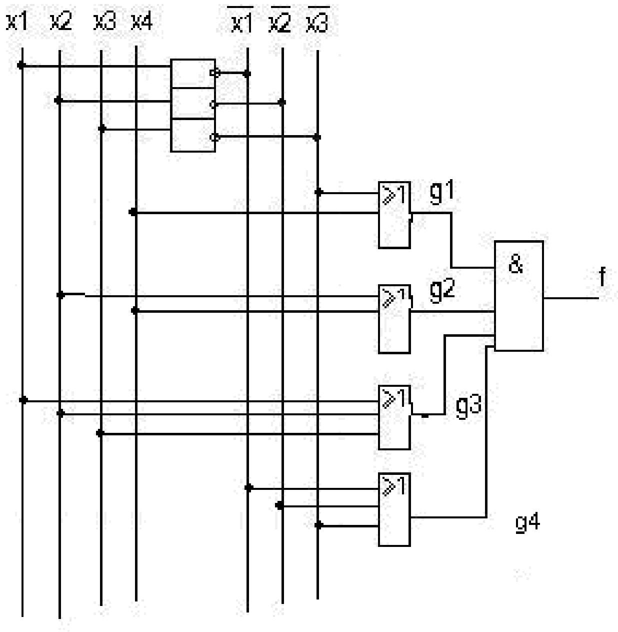
$$g_4 = \overline{x_1} \vee \overline{x_2} \vee \overline{x_3}$$

МКНФ:

$$f_{\min}(x_1, x_2, x_3, x_4) = g_1 g_2 g_3 g_4 = (\overline{x_3} \vee x_4)(x_2 \vee x_4)(x_1 \vee x_2 \vee x_3)(\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

- МКНФ в булевому базисі.

Функціональна схема:



Складність КС2:
N=8 логічних елементів
НІ – 3 шт.
АБО – 4 шт.(2АБО, - 2шт)
(3АБО – 2 шт.)
4І- 1 шт.
Логічна глибина:
L=3
$t_{затр.кс4}$= 3* $t_{затр.р.серед}$
Таким чином, КС2 складніше за КС1 .
Швидкодія КС2 така ж само, як у КС1 .

Рис.14.5

14.3 Індивідуальні завдання

Задана булева функція 4-х змінних f (x_1,x_2,x_3,x_4), яка приймає одиничне значення на двійкових наборах з номерами, зазначеними в таблиці варіантів. На інших наборах від 0 до 15 функція приймає нульове значення.

Необхідно спроектувати комбінаційні схеми (КС), що відтворюють задану булеву функцію. Завдання вирішити з використанням МДНФ і МКНФ. Мінімізацію виконати за допомогою карт Карно.

Функціональні схеми зобразити в булевом базисі (І,АБО,НІ).

Оцінити складність спроектованих схем у кількості логічних елементів, необхідних для їхньої реалізації.

Оцінити швидкодію спроектованих схем в одиницях середнього часу затримки розповсюдження одного логічного елемента ($t_{затр.р.серед.}$).

Указати схеми, що мають меншу складність і кращу швидкодію.

Таблиця 14.3.1.

№ на-бора					№ варианта																			
	x_1	x_2	x_3	x_4	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
0	0	0	0	0		1			1			1			1			1			1			1
1	0	0	0	1	1	1			1	1	1		1		1	1	1	1			1	1	1	
2	0	0	1	0		1	1				1		1					1	1				1	
3	0	0	1	1		1		1		1			1		1	1		1		1		1		
4	0	1	0	0			1	1		1		1	1			1			1	1		1		1
5	0	1	0	1	1	1		1	1			1		1			1			1	1			1
6	0	1	1	0			1			1		1		1	1				1			1		1
7	0	1	1	1	1			1				1					1			1				1
8	1	0	0	0					1		1	1							1		1		1	1
9	1	0	0	1		1		1	1	1	1					1		1			1	1	1	
10	1	0	1	0			1				1		1			1			1		1		1	
11	1	0	1	1	1	1		1		1			1		1		1	1		1		1		
12	1	1	0	0					1		1			1							1		1	1
13	1	1	0	1	1		1							1	1		1		1					
14	1	1	1	0							1			1		1							1	
15	1	1	1	1	1		1	1					1	1		1	1			1				

14.4 Контрольні запитання

1) Дайте визначення комбінаційної схеми.
2) Що таке глибина КС?
3) Дайте визначення функціонально-повного базису. Наведіть приклади.
4) Що таке логічний елемент? Дайте визначення наступних технічних параметрів логічних елементів: коефіцієнт об'єднання по входу; коефіцієнт об'єднання по виходу.

15 ПРОЕКТУВАННЯ КОМБІНАЦІЙНИХ СХЕМ В МОНОФУНКЦІОНАЛЬНИХ БАЗИСАХ І-НІ, АБО-НІ

15.1 Мета заняття

Вивчити методи і придбати практичні навички проектування комбінаційних схем в монофункціональних базисах І-НІ, АБО-НІ.

15.2 Вирішення типових завдань

Переведемо МДНФ з булевого базису в монофункціональний базис І-НІ, для чого скористаємося правилом Де Моргана а також законом подвійного заперечення [1,5,10, 12,13].

$$f(x_1,x_2,x_3,x_4)=\overline{\overline{x_2\overline{x_3}\vee\overline{x_1}x_3x_4\vee x_1\overline{x_2}x_4}}=\overline{\overline{(x_2\overline{x_3})}\,\overline{(\overline{x_1}x_3x_4)}\,\overline{(x_1\overline{x_2}x_4)}}$$

Функціональна схема, відповідна до цього виразу, буде мати вигляд:

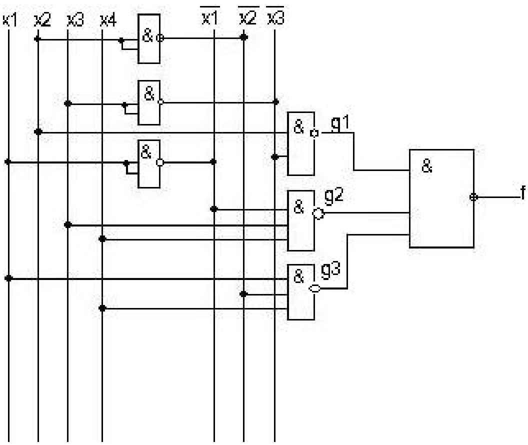


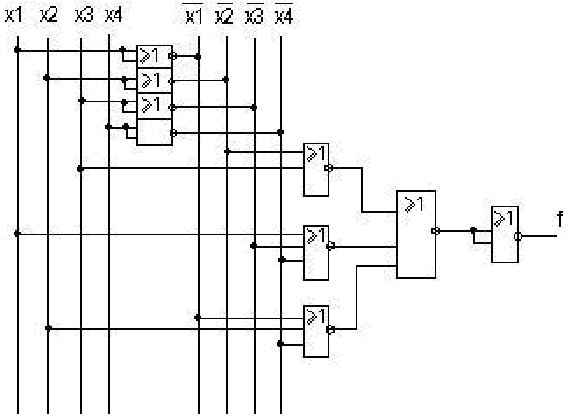
Рис.15.1

Складність КСЗ:
N=7 логічних елементів
2І-НІ – 4 шт.
3І-НІ – 3шт.
Логічна глибина
L=3
$t_{затр.кс3=}$ 3* $t_{затр.р.серед}$
Тобто ця схема рівноцінна КС1.

Переведемо МДНФ в монофункціональний базис (АБО-НІ):

$$f(x_1,x_2,x_3,x_4)=\overline{\overline{\overline{\overline{x_2\overline{x_3}}}\vee\overline{\overline{\overline{x_1}x_3x_4}}\vee\overline{\overline{x_1\overline{x_2}x_4}}}}=\overline{\overline{\overline{(\overline{x_2}\vee x_3)}\vee\overline{(x_1\vee\overline{x_3}\vee\overline{x_4})}\vee\overline{(\overline{x_1}\vee x_2\vee\overline{x_4})}}}$$

Схема, що відповідає цьому виразу, буде мати наступний вигляд:



Складність КС4:
N=9 логічних елементів
2АБО-НІ – 6 шт.
3АБО-НІ – 3шт.
Логічна глибина
L=4
$t_{затр.кс4=}$ 4* $t_{затр.р.серед}$
Тобто ця схема гірше попередніх трьох .

Рис.15.2

Перейдемо до базису І-НІ:

$$f(x_1,x_2,x_3,x_4)=\overline{\overline{\overline{\overline{(\bar{x}_3\vee x_4)}}\,\overline{\overline{(x_2\vee x_4)}}\,\overline{\overline{(x_1\vee x_2\vee x_3)}}\vee\overline{\overline{(\bar{x}_1\vee\bar{x}_2\vee\bar{x}_3)}}}}=\overline{\overline{\overline{(x_3\bar{x}_4)}\,\overline{(\bar{x}_2\bar{x}_4)}\,\overline{(\bar{x}_1\bar{x}_2\bar{x}_3)}\,\overline{(x_1x_2x_3)}}}$$

Будуємо схему в базисі І-НІ (див. рис. 15.3)

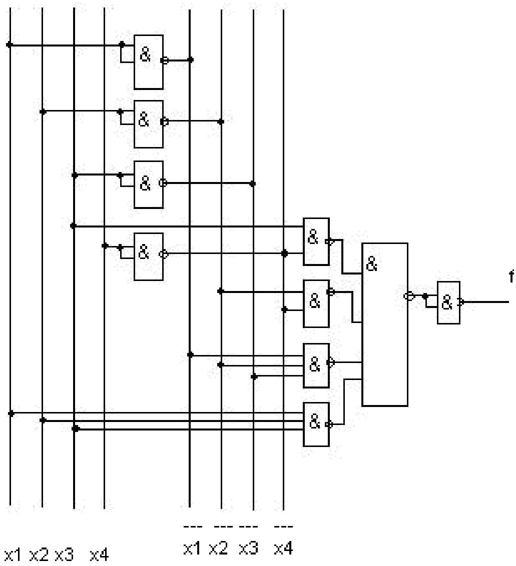


Рис.15.3

Складність КС5

N_{KC5}=10 логічних елементів І-НІ

2 І-НІ – 7шт

3 І-НІ – 2шт

4 І-НІ – 1шт

Логічна глибина L=4

$t_{затр.кс5=}$ 4* $t_{затр.р.серед}$

Таким чином, КС5 складніше за усі попередні схеми по швидкодії КС5 рівноцінна КС4.

Перейдемо до базису АБО-НІ:

$$f(x_1, x_2, x_3, x_4) = \overline{\overline{(\overline{x_3} \vee x_4)(x_2 \vee x_4)(x_1 \vee x_2 \vee x_3)(\overline{x_1} \vee \overline{x_2} \vee x_3)}} =$$

$$= \overline{\overline{(\overline{x_3} \vee x_4)} \vee \overline{(x_2 \vee x_4)} \vee \overline{(x_1 \vee x_2 \vee x_3)} \vee \overline{(\overline{x_1} \vee \overline{x_2} \vee \bar{x}_3)}}$$

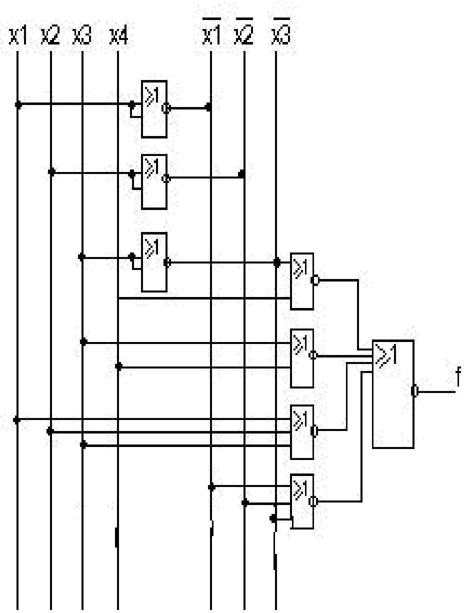


Рис.15.4

Складність:

$N_{KC6} = 8$*логічних елементів*

2 АБО-НІ – 5 шт.

3 АБО-НІ – 2 шт.

4 АБО-НІ – 1 шт.

Задержка:

L=3

$t_{затр.кс6=}$ 3* $t_{затр.р.серед}$

Висновок: найкращими схемами по комбінації параметрів складність плюс швидкодія є КС1 и КС3.

15.3 Індивідуальні завдання

Задана булева функція 4-х змінних f (x_1,x_2,x_3,x_4), яка приймає одиничне значення на двійкових наборах з номерами, зазначеними в таблиці варіантів. На інших наборах від 0 до 15 функція приймає нульове значення.

Таблиця 15. 1.

№ набора					№ варианта																			
	x_1	x_2	x_3	x_4	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
0	0	0	0	0		1			1			1			1			1			1			1
1	0	0	0	1	1	1			1	1	1		1		1	1	1	1			1	1	1	
2	0	0	1	0		1	1				1		1					1	1				1	
3	0	0	1	1		1		1		1			1		1	1		1		1		1		
4	0	1	0	0			1	1		1		1	1			1			1	1		1		1
5	0	1	0	1	1	1		1	1			1		1			1			1	1			1
6	0	1	1	0			1			1		1		1	1				1			1		1
7	0	1	1	1	1			1				1					1			1				1
8	1	0	0	0					1		1	1							1		1		1	1
9	1	0	0	1		1		1	1	1	1					1		1			1	1	1	
10	1	0	1	0			1				1		1			1			1		1		1	
11	1	0	1	1	1	1		1		1			1		1		1	1		1		1		
12	1	1	0	0					1		1			1							1		1	1
13	1	1	0	1	1		1							1	1		1		1					
14	1	1	1	0							1			1		1							1	
15	1	1	1	1	1		1	1					1	1		1	1			1				

Необхідно спроектувати комбінаційні схеми (КС), що відтворюють задану булеву функцію. Завдання вирішити з використанням МДНФ і МКНФ. Мінімізацію виконати за допомогою карт Карно.

Функціональні схеми зобразити в монофункціональних базисах (І-НІ) і (АБО-НІ).(Усього 4 схеми).

Оцінити складність спроектованих схем у кількості логічних елементів, необхідних для їхньої реалізації.

Оцінити швидкодію спроектованих схем в одиницях середнього часу затримки розповсюдження одного логічного елемента ($t_{затр.р.серед.}$).

Указати схеми, що мають меншу складність і кращу швидкодію.

15.4 Контрольні запитання

1) Дайте визначення комбінаційної схеми.
2) Що таке глибина КС?
3) Дайте визначення функціонально-повного базису. Наведіть приклади.
4) Що таке логічний елемент? Дайте визначення наступних технічних параметрів логічних елементів: коефіцієнт об'єднання по входу; коефіцієнт об'єднання по виходу.
5) Як перетворити функцію до монофункціонального базису?

16 АБСТРАКТНІ ЦИФРОВІ АВТОМАТИ. МЕТОДИ ЗАВДАННЯ І ЗВ'ЯЗОК МІЖ НИМИ

16.1 Мета заняття

Вивчити типи абстрактних автоматів, способи завдання абстрактних автоматів Мілі, Мура, С-автоматів (табличний, графічний, аналітичний) і взаємозв’язок між ними. [11, 12, 13]

16.2 Короткі основні відомості з теорії та вирішення типових завдань

16.2.1 Абстрактні автомати. Типи абстрактних автоматів.

Цифровий автомат можна трактувати як пристрій, який здійснює прийом, збереження та перетворення дискретної інформації за деяким алгоритмом.

Автоматом називається дискретний перетворювач інформації, здатний приймати різні стани, переходити під впливом вхідних сигналів з одного стану в інший і видавати вихідні сигнали.

Абстрактний цифровий автомат (ЦА) є математичною моделлю дискретного керуючого пристрою.

Абстрактний ЦА визначається множиною, що складається з шести елементів:

$S=\{X, A, Y, \delta, \lambda, a_o\}$, де:

$X=\{x_1, x_2, \dots x_n\}$ - множина вхідних сигналів (вхідний алфавіт);

$Y=\{y_1, y_2, \dots y_m\}$ - множина вихідних сигналів (вихідний алфавіт);

$A=\{a_0, a_1, a_2, \dots a_N\}$ - множина станів (алфавіт станів);

a_o- початковий стан ($a_o \in A$);

δ - функція переходів автомата, що задає відображення (XxA)$\rightarrow$A, тобто ставить у відповідність будь-якій парі елементів декартового добутку (XxA) елемент множини A;

λ - функція виходів автомата, що задає або відображення (XxA)$\rightarrow$Y, або відображення A$\rightarrow$Y.

Іншими словами, функція переходів δ показує, що автомат S, перебуваючи в деякому стані $a_j \in A$,, при появі вхідного сигналу $x_j \in X$ переходить у якийсь стан $a_p \in A$. Це можна записати:

$a_p = \delta\ (a_i, X_j)$.

Функція виходів показує, що автомат S, перебуваючи в деякому стані $a_j \in A$, при появі вхідного сигналу $x_j \in X$ видає вихідний сигнал $y_k \in Y$. Це можна записати:

$y_k = \lambda\ (a_i, X_j)$.

Поняття стан у визначення автомата було введено в зв'язку з тим, що часто виникає необхідність в описі поведінки систем, виходи яких залежать не тільки від стану входів в даний момент часу, але і від деякої передісторії, тобто від сигналів, які надходили на входи системи раніше. Стан як раз і відповідає деякій пам'яті про минуле, дозволяючи усунути час як явну змінну і виразити вихідні сигнали як функцію станів і входів в даний момент часу.

Абстрактний автомат функціонує в дискретному автоматному часу $t=0,1,2, \ldots$ і переходи зі стану в стан здійснюються миттєво. У кожен момент t дискретного часу автомат знаходиться в певному стані a(t) з множини A станів автомата, причому в початковий момент часу t=0 він завжди знаходиться в початковому стані a_o. У момент часу t будучи в стані a(t), автомат здатний сприйняти на вхідному каналі сигнал $x(t) \in X$ і видати на вихідному каналі сигнал $y(t) = \lambda\ (a(t), x(t))$, переходячи в стан $a(t+1) = \delta\ (a(t), x(t))$. Залежність вихідного сигналу від вхідного і від стану означає, що в його складі є пам'ять.

За способом формування функції виходів виділяють три типи абстрактних автоматів: автомат Мілі, автомат Мура, С-автомат. Автомат Мілі характеризується системою рівнянь:

$$y(t) = \lambda\ (a(t), x(t));$$
$$a(t+1) = \delta(a(t), x(t)). \qquad (1\text{-}1)$$

Автомат Мура - системою рівнянь:

$$y(t) = \lambda\ (a(t));$$
$$a(t+1) = \delta(a(t), x(t)). \qquad (1\text{-}2)$$

С-автомат - системою рівнянь:

$$y = y_1 \cup y_2$$
$$y_1(t) = \lambda_1 (a(t), x(t));$$
$$У_2(t) = \lambda_2(a(t));$$
$$a(t+1) = \delta(a(t), x(t)).$$

Якщо на вхід абстрактного автомата Мілі або Мура, встановленого в початковий стан a_o, подавати буква за буквою деяку послідовність букв вхідного алфавіту x(0), x(1), ...- вхідне слово, то на виході автомата будуть послідовно з'являтися літери вихідного алфавіту y(0), y(1),...- вихідне слово. Для випадку С-автомата на його виходах будуть з'являтися дві послідовності: $y_1(0)$, $y_1(1)$,... и $y_2(0)$, $y_2(1)$,.... В абстрактному С-автоматі вихідний сигнал $y_2(t)=\lambda_2(a(t))$ видається весь час, поки автомат знаходиться в стані a(t). Вихідний сигнал $y_1(t)=\lambda_1(a(t),x(t))$ видається під час дії вхідного сигналу x(t) при знаходженні С-автомата в стані a(t).

Таким чином, на рівні абстрактної теорії функціонування цифрового автомата мають на увазі перетворення вхідних слів у вихідні слова.

Виділяють повністю визначені і часткові автомати.

Повністю визначеним називається абстрактний цифровий автомат, у якого функція переходів або функція виходів, або обидві ці функції визначені для всіх пар переходів (x_i, a_j).

Частковим називається абстрактний цифровий автомат, у якого функція переходів або функція виходів, або обидві ці функції визначені не для всіх пар переходів (x_i, a_j).

Абстрактний цифровий автомат називається ініціальний, якщо на множині його станів A виділяється початковий стан a_o.

Щоб задати кінцевий автомат S, необхідно описати всі елементи множини: $S=\{X, A, Y, \delta, \lambda, a_o\}$. Існує кілька способів завдання роботи автомата, але найбільш часто використовується табличний (матричний), графічний, аналітичний.

При табличному способі автомат задається двома таблицями: таблицею переходів і таблицею виходів, або матрицею з'єднань. Таблиця переходів довільного повністю визначеного автомата будується наступним чином: рядки таблиці позначаються буквами вхідного алфавіту автомата, а стовпці таблиці - літерами алфавіту станів автомата; У клітинці таблиці переходів, що знаходиться на перетині рядка, зазначеного вхідним сигналом x_i, і стовпця зазначеного станом a_j, ставиться стан a_k, що є результатом переходу автомата зі стану a_j під впливом вхідного сигналу x_i, що визначається виразом $a_k=\delta(a_j, x_j)$.

Таблиця 16.1. Таблиця переходів автомата

Стани / Вхідні сигнали	a_1	a_2		a_k
x_1	$\delta(a_1,x_1)$	$\delta(a_2, x_1)$	...	$\delta(a_к, x_1)$
…				
x_j	$\delta(a_1,x_j)$	$\delta(a_2, x_j)$		$\delta(a_к, x_j)$

Приклад заповнення таблиці переходів деякого абстрактного повністю визначеного автомата з вхідним алфавітом X={x_1, x_2} і алфавітом станів A={a_1, a_2, a_3} представлений в табл. 16.2.

Таблиця 16.2

A / x	a_1	a_2	a_3
x_1	a_2	a_3	a_1
x_2	a_1	a_1	a_2

Якщо абстрактний автомат частковий, то в клітинці таблиці його переходів, що знаходиться, на перетині рядка, зазначеної вхідним сигналом і стовпця зазначеного відповідним станом (за умови, що перехід у цей стан під дією даного вхідного сигналу не визначено) ставиться прочерк, і будь яке вхідне слово , що приводить до зазначеного переходу є забороненим.

Заповнення інших клітин аналогічно випадку повністю визначеного автомата. Вид таблиці переходів не залежить від типу заданого автомата (автомат Мілі, Мура, С-автомат). Таблиці виходів автоматів Мілі, Мура, С-автомата мають відмінності.

Таблиця виходів повністю визначеного автомата Мілі будується наступним чином: ідентифікація стовпців і рядків, а також формат таблиці відповідають таблиці переходів повністю визначеного автомата. У клітинці таблиці виходів, що знаходиться на перетині рядка, зазначеної вхідним сигналом x_j, і стовпця, зазначеного станом $a_к$, ставиться вихідний сигнал y_m, який автомат видає, перебуваючи в стані a_k при наявності вхідного сигналу x_j, що визначається виразом: $y_m = \lambda(a_k, x_j)$.

Таблиця 16.3. Таблиця виходів абстрактного автомата Мілі.

Стани / Вхідні сигнали	a_1	a_2		a_k
x_1	$\lambda(a_1,x_1)$	$\lambda(a_2, x_1)$		$\lambda(a_k, x_1)$
…		….	...	
x_j	$\lambda(a_1,x_j)$	$\lambda(a_2, x_j)$		$\lambda(a_к, x_j)$

Приклад заповнення таблиці виходів деякого абстрактного повністю визначеного автомата Мілі з вхідним алфавітом X={x_1, x_2}, алфавітом станів A={a_1, a_2, a_3} та вихідним алфавітом Y={y_1, y_2, y_3} - представлений в табл. 61.4.

Таблиця 16.4

a / x	a_1	a_2	a_3
x_1	y_2	y_3	y_1
x_2	y_3	y_1	y_2

Таблиця виходів повністю визначеного автомата Мура будується простіше: кожному стану автомата ставиться у відповідність свій вихідний сигнал. Приклад таблиці виходів автомата Мура з алфавітом станів A={a_1, a_2, a_3} та вихідним алфавітом Y={y_1, y_2, y_3} - представлений в табл. 16.5.

Таблиця 16.5.

A	a_1	a_2	a_3
y	y_1	y_2	y_2

На практиці таблиці переходів і виходів часто поєднують в одну таблицю, звану зазначеною (поєднаною) таблицею переходів автомата. Приклади зазначених таблиць переходів представлені в табл. 16.6. -16.8. (Загальний вигляд зазначеної таблиці переходів - табл. 16.6., Зазначена таблиця переходів автомата Мілі - табл. 16.7., Зазначена таблиця переходів автомата Мура - табл. 16.8.).

Крім розглянутих вище таблиць переходів і виходів довільний абстрактний автомат може бути заданий матрицею з'єднань.

Матриця з'єднань є квадратною і містить стільки стовпців (рядків), скільки різних станів містить алфавіт станів даного автомата. Кожен стовпець (рядок) матриці з'єднань позначається буквою стану автомата. У клітинці, що знаходиться на перетині стовпця, поміченого буквою a_j і рядки з позначкою буквою a_s автомата, ставиться вхідний сигнал (або диз'юнкція вхідних сигналів), під впливом якого здійснюється даний перехід.

Для абстрактного автомата Мілі в клітці поруч із станом проставляється також вихідний сигнал, який автомат видає в результаті даного переходу (табл. 16.9.) Для автомата Мура вихідний сигнал проставляється в рядку поруч із станом (ці стани відповідають вихідним станам автомата).

Таблиця 16.6

Стани / Вхідні сигнали	a_1	a_2	…	a_k
x_1	δ (a_1,x_1)	δ (a_2,x_1)	…	δ (a_k,x_1)
	λ (a_1,x_1)	λ (a_2,x_1)	…	λ (a_k,x_1)
…	…	…	…	…
x_j	δ (a_1,x_j)	δ (A_2,X_J)	…	δ (a_k, x_j)
	λ (a_1,x_j)	λ (a_2, x_j)	…	λ (a_k, x_j)

Таблиця 16.7.

A / X	a_1	a_2	a_3
x_1	a_2/y_2	a_3/y_3	a_1/y_3
x_2	a_1/y_3	a_1/y_1	a_2/y_2

Таблиця 16.8.

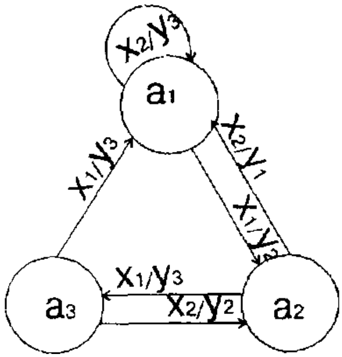
Y	y_1	y_2	y_3
A / X	a_1	a_2	a_3
x_1	a_2	a_3	a_1
x_2	a_1	a_1	a_2

Таблиця 16.9.

	a_1		a_n
a_1	$x_j(y_k)$		
a_n			$x_j(y_m)$

При графічному способі завдання абстрактні автомати представляються орієнтованими графами. Графом автомата називається орієнтований зв'язний граф, вершини якого відповідають станам автомата, а дуги між ними - переходам між станами. Дві вершини a_k і a_s з'єднуються дугою в тому випадку, якщо в автоматі є перехід зі стану a_k в стан a_s. Для автомата Мілі вхідний і вихідний сигнали проставляються на дузі, що відповідає даному переходу (рис. 6.1.), Для автомата Мура вхідний сигнал проставляється на дузі, а вихідний - поруч з вершиною, відповідної станом (рис 16.2.).

Тут розглядаються тільки детерміновані автомати, у яких виконана умова однозначності переходів: автомат, що знаходиться в деякому стані, під дією будь-якого вхідного сигналу не може перейти більш ніж в один стан. Стосовно до графічного способу завдання автомата це означає, що в графі автомата з будь-якої вершини не можуть виходити дві і більше дуги, позначені одним і тим же вхідним сигналом.



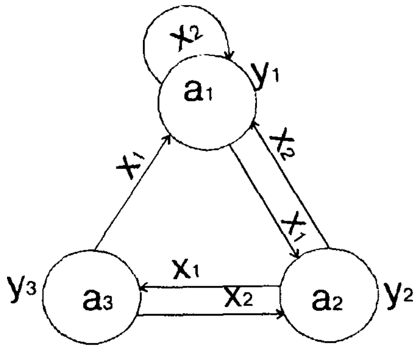


Рисунок 16.1-Граф автомата Мілі Рисунок 16.2-Граф автомата Мура

При аналітичному способі завдання абстрактні автомати задаються четвіркою об'єктів:

S = {X, A, Y, F}, де F задає для кожного стану a_j автомата відображення (XxA) → (AxY). Іншими словами, при аналітичному способі завдання для кожного стану автомата a_j вказується відображення F_{ai}, що представляє собою безліч всіх трійок a_p, x_m, y_k, і таких, що під впливом вхідного сигналу x_m автомат переходить зі стану a_j в стан a_p, видаючи при цьому вихідний сигнал y_k. Стосовно до загального визначенням абстрактного автомата, останнє рівнозначно опису функцій δ і λ у відповідності з виразом: $a_p = \delta(a_i, x_m)$, $y_к = \lambda\,(a_i, x_m)$.

Відображення F_{ai} записується таким чином:

$F_{ai}\{a_p(X_m/y_k),a_i(X_f/y_z)\ldots\}$.

Для абстрактного автомата Мілі (табл. 16.7.) аналітичне завдання має наступний вигляд:

$S=\{X,A,Y,F\}$, $X=\{x_1,x_2\}$, $A=\{a_1, a_2, a_3\}$, $Y=\{y_1, y_2, y_3\}$,

$F_{a1}=\{a_2(x_1/y_2), a_1(x_2/y_3)\}$,

$F_{a2}=\{a_3(x_1/y_3), a_1(x_2/y_1)\}$,

$F_{a3}=\{a_1(x_1/y_3), a_2(x_2/y_2)\}$.

Слід зазначити, що функція F_{ai} завжди записується для вихідного стану.

Задача 1.

Абстрактний автомат Мілі задано наступними таблицями переходів і виходів.

Таблиця 16.10 Таблиця переходів

A / X	a_1	a_2	a_3	a_4
x_1	a_1	a_3	a_3	a_3
x_2	a_2	a_4	a_1	a_1
x_3	a_1	a_2	a_4	a_2

Таблиця 16.11 Таблица виходів

A / X	a_1	a_2	a_3	a_4
x_1	y_1	y_3	y_2	y_2
x_2	y_1	y_3	y_1	y_1
x_3	y_2	y_3	y_3	y_3

Потрібно визначити:

1) Скільки елементів містить: вхідний алфавіт X, алфавіт станів A, вихідний алфавіт Y?

2) Визначте функції переходів: $\delta(a_1, x_1) = ?$; $\delta(a_2, x_1) = ?$; $\delta(a_3, x_2) = ?$; $\delta(a_4, x_3) = ?$ та прокоментуйте їх.

3) Визначте функції виходів $\lambda(a_4, x_2) = ?$; $\lambda(a_3, x_3) = ?$; $\lambda(a_4, x_1) = ?$ та прокоментуйте їх.

4) Складіть зазначену таблицю переходів заданого автомата Мілі.

5) Складіть матрицю з'єднань заданого автомата Мілі.

Рішення

1) Вхідний алфавіт $X = \{x_1, x_2, x_3\}$ – складає 3 букви;
Алфавіт станів A=$\{a_1, a_2, a_3, a_4\}$ - складає 4 букви;
Вихідний алфавіт Y= $\{y_1, y_2, y_3\}$ – складає 3 букви.

2) Функція переходів $\delta(a_1, x_1) = a_1$, свідчить про те, що автомат S, перебуваючи в стані a_1 при надходженні на його вхід вхідного сигналу x_1 переходить в стан a_1, тобто не змінює його стану (зберігає попередній стан).

$\delta(a_2, x_1) = a_3$- при надходженні вхідного сигналу x_1 здійснюється перехід $a_2 \rightarrow a_3$

$\delta(a_4, x_3) = a_2$- при надходженні вхідного сигналу x_3 здійснюється перехід $a_4 \rightarrow a_2$

3) Функція виходів $\lambda(a_1, x_2) = y_1$, свідчить про те, що при надходженні вхідного сигналу x_2 на вхід автомата, якщо він знаходився в стані a_1, то на виході автомата формується вихідний сигнал y_1;

$\lambda(a_3, x_3) = y_3$; $\lambda(a_4, x_1) = y_3$.

4) Зазначена (сумісна) таблиця переходів суміщає таблицю переходів і таблицю виходів і буде мати наступний вигляд:

Таблиця 16.12 – Зазначена таблиця переходів

X \ A	a_1	a_2	a_3	a_4
x_1	a_1 / y_1	a_3 / y_3	a_3 / y_2	a_3 / y_2
x_2	a_2 / y_1	a_4 / y_2	a_1 / y_1	a_1 / y_1
x_3	a_1 / y_2	a_2 / y_2	a_4 / y_2	a_2 / y_2

5) Матриця з'єднань представляє собою квадратну матрицю розміром А х А і вона також об'єднує таблиці переходів і виходів абстрактного автомата.

Для заданого автомата вона набуває наступний вигляд:

Таблиця 16.13 – Матриця з'єднань

a(t+1) \ a(t)	a_1	a_2	a_3	a_4
a_1	$x_1(y_1) \vee x_3(y_2)$	-	$x_2(y_1)$	$x_2(y_1)$
a_2	$x_2(y_1)$	$x_3(y_3)$	-	$x_3(y_3)$
a_3	-	$x_1(y_3)$	$x_1(y_2)$	$x_1(y_2)$
a_4	-	$x_2(y_3)$	$x_3(y_3)$	-

Задача 2.

Абстрактний автомат Мілі, заданий таблицею переходів і таблицею з'єднань із задачі 1, задати графічно.

Рішення.

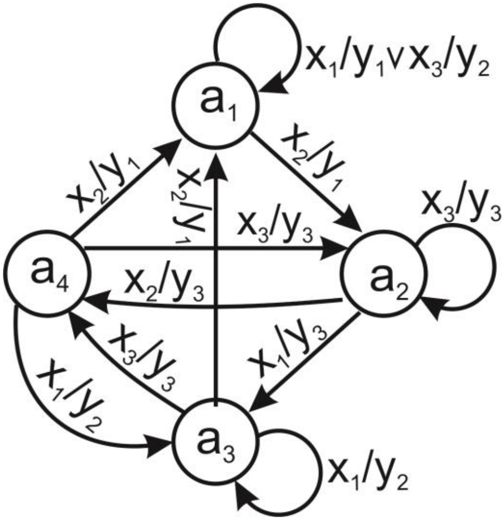


Рис 16.3- Граф автомата Мілі.

Задача 3. Цей же автомат задати аналітично.

Рішення.

Аналітично автомат задається четвіркою об'єктів:

$S = \{X, A, Y, F\}$

де F задає для кожного стану a_i автомата відображення декартового добутку

(XxA) → (AxY)

У нашому випадку аналітичне завдання абстрактного автомата Мілі буде виглядати так (див. табл. 16.12):

$X = \{x_1, x_2, x_3\}$; $A = \{a_1, a_2, a_3, a_4\}$; $Y = \{y_1, y_2, y_3\}$

$F_{a1} = \{a_1(x_1/y_1), a_2(x_2/y_1), a_1(x_3/y_2)\}$

$F_{a2} = \{a_3(x_1/y_3), a_4(x_2/y_3), a_2(x_3/y_3)\}$

$F_{a3} = \{a_3(x_1/y_2), a_1(x_2/y_1), a_4(x_3/y_3)\}$

$F_{a4} = \{a_3(x_1/y_2), a_1(x_2/y_1), a_2(x_3/y_3)\}$

Вирішимо аналогічні завдання для різних способів завдання абстрактного автомата Мура.

Задача 4.

Абстрактний автомат Мура заданий наступними таблицями переходів і виходів.

Таблиця 16.14 – Таблиця переходів

X \ A	a_1	a_2	a_3	a_4
x_1	a_2	a_1	a_2	a_4
x_2	a_1	a_4	a_1	a_3

Таблиця 16.15 – Таблиця виходів

A	a_1	a_2	a_3	a_4
Y	y_2	y_1	y_1	y_3

Потрібно:
1) визначити алфавіти X, A, Y:
2) визначити функції переходів $\delta(a_1, x_2) = ?$; $\delta(a_3, x_1) = ?$; $\delta(a_4, x_2) = ?$
3) визначити функції виходів $\lambda(a_1) = ?$; $\lambda(a_3) = ?$; $\lambda(a_4) = ?$
4) скласти зазначену таблицю переходів;
5) скласти матрицю з'єднань;
6) задати автомат графічно;
7) задати автомат аналітично.

Рішення.

1) $X = \{x_1, x_2\}$; $A = \{a_1, a_2, a_3, a_4\}$; $Y=\{y_1, y_2, y_3\}$
2) $\delta(a_1, x_1) = a_2$; $\delta(a_1, x_2) = a_1$; $\delta(a_2, x_1) = a_1$; $\delta(a_2, x_2) = a_4$; $\delta(a_3, x_1) = a_2$; $\delta(a_3, x_2) = a_1$; $\delta(a_4, x_1) = a_4$; $\delta(a_4, x_2) = a_3$
3) $\lambda(a_1) = y_2$; $\lambda(a_3) = y_1$; $\lambda(a_4) = y_3$; $\lambda(a_2) = y_1$

4) Складаємо зазначену таблицю переходів автомата Мура (див. табл. 16.16)

Таблиця 16.16

Y	y_2	y_1	y_1	y_3
X \ A	a_1	a_2	a_3	a_4
x_1	a_2	a_1	a_2	a_4
x_2	a_1	a_4	a_1	a_3

5) Складаємо матрицю з'єднань (див. табл. 16.17

Таблиця 16.17 – Матриця з'єднань

a(t) \ a(t+1)	a_1	a_2	a_3	a_4
a_1	x_2	x_1	x_2	-
a_2	x_1	-	x_1	-
a_3	-	-	-	x_2
a_4	-	x_2	-	x_1

6) Задаємо автомат Мура графічно (див. рис. 16.4)

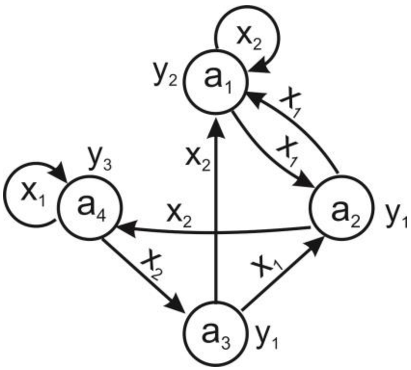


Рис. 16.4 Граф автомата Мура

7) Задаємо автомат аналітично:

$S = (X, A, Y, F)$

$X = \{x_1, x_2\}$; A = $\{a_1, a_2, a_3, a_4\}$; Y=$\{y_1, y_2, y_3\}$

$F_{a1} = \{a_2(x_1/y_1), a_1(x_2/y_2)\}$

$F_{a2} = \{a_1(x_1/y_2), a_4(x_2/y_3)\}$

$F_{a3} = \{a_2(x_1/y_1), a_1(x_2/y_2)\}$

$F_{a4} = \{a_4(x_1/y_3), a_3(x_2/y_1)\}$

16.3 Індивідуальні завдання

Завдання 1.

Абстрактний автомат Мура задано поєднаною таблицею:
- задати таблицю переходів даного автомата;
- задати таблицю виходів даного автомата;
- задати матрицю з'єднань даного автомата;
- побудувати граф даного автомата
- задати автомат аналітичним способом.

Варіант №1				
У	y_1	y_2	y_2	y_3
A / X	a_1	a_2	a_3	a_4
x_1	a_1	a_2	a_4	a_3
x_2	a_2	a_3	a_3	a_2
x_3	a_3	a_4	a_1	a_1

Варіант №2				
У	y_1	y_3	y_2	y_4
A / X	a_1	a_2	a_3	a_4
x_1	a_2	a_1	a_2	a_1
x_2	a_3	a_2	a_3	a_2
x_3	a_1	a_4	a_1	a_3

Варіант №3				
У	y_2	y_3	y_1	y_4
A / X	a_1	a_2	a_3	a_4
x_1	a_4	a_1	a_4	a_3
x_2	a_3	a_3	a_2	a_4
x_3	a_1	a_4	a_1	a_1

Варіант №4				
У	y_1	y_2	y_3	y_4
A / X	a_1	a_2	a_3	a_4
x_1	a_3	a_1	a_2	a_1
x_2	a_3	a_3	a_4	a_4
x_3	a_4	a_4	a_1	a_3

Варіант №5				
У	y_2	y_3	y_1	y_4
A / X	a_1	a_2	a_3	a_4
x_1	a_4	a_3	a_2	a_1
x_2	a_1	a_4	a_4	a_4
x_3	a_2	a_4	a_1	a_3

Варіант №6				
У	y_1	y_2	y_3	y_4
A / X	a_1	a_2	a_3	a_4
x_1	a_3	a_1	a_2	a_1
x_2	a_2	a_4	a_4	a_3
x_3	a_3	a_3	a_1	a_4

Варіант №7				
У	y_2	y_3	y_1	y_4
A / X	a_1	a_2	a_3	a_4
x_1	a_4	a_3	a_2	a_1
x_2	a_1	a_4	a_3	a_4
x_3	a_2	a_1	a_1	a_3

Варіант №8				
У	y_1	y_2	y_3	y_4
A / X	a_1	a_2	a_3	a_4
x_1	a_3	a_1	a_2	a_1
x_2	a_2	a_4	a_4	a_3
x_3	a_3	a_3	a_1	a_4

Варіант №9				
Y	y_2	y_3	y_1	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_1	a_3	a_2	a_2
x_2	a_3	a_4	a_3	a_4
x_3	a_2	a_4	a_1	a_1

Варіант №10				
Y	y_1	y_2	y_3	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_3	a_1	a_2	a_4
x_2	a_2	a_2	a_4	a_2
x_3	a_4	a_3	a_1	a_1

Завдання 2.

Абстрактний автомат Мілі задано поєднаною таблицею:

- задати таблицю переходів даного автомата;
- задати таблицю виходів даного автомата;
- задати матрицю з'єднань даного автомата;
- побудувати граф даного автомата
- задати автомат аналітичним способом.

Варіант №1					
X \ A	a_1	a_2	a_3	a_4	a_5
x_1	a_1/y_1	a_2/y_2	a_4/y_4	a_5/y_2	a_4/y_1
x_2	a_3/y_2	a_5/y_1	a_3/y_2	a_4/y_3	a_2/y_3
x_3	a_4/y_3	a_3/y_4	a_1/y_3	a_2/y_1	a_1/y_4

Варіант №2					
X \ A	a_1	a_2	a_3	a_4	a_5
x_1	a_1/y_2	a_2/y_3	a_4/y_2	a_3/y_4	a_4/y_4
x_2	a_2/y_3	a_3/y_4	a_5/y_3	a_2/y_2	a_3/y_1
x_3	a_5/y_4	a_4/y_1	a_1/y_1	A_5/y_4	a_1/y_2

Варіант №3					
X \ A	a_1	a_2	a_3	a_4	a_5
x_1	a_4/y_1	a_1/y_2	a_5/y_4	a_5/y_1	a_4/y_1
x_2	a_1/y_2	a_5/y_4	a_2/y_1	a_4/y_3	a_3/y_2
x_3	a_3/y_3	a_3/y_3	a_1/y_3	a_3/y_2	a_1/y_4

Варіант №4					
X \ A	a_1	a_2	a_3	a_4	a_5
x_1	a_1/y_2	a_1/y_2	a_4/y_2	a_5/y_3	a_3/y_4
x_2	a_4/y_1	a_2/y_3	a_5/y_4	a_3/y_2	a_2/y_1
x_3	a_5/y_3	a_3/y_1	a_1/y_3	a_2/y_4	a_1/y_2

Варіант №5					
X \ A	a_1	a_2	a_3	a_4	a_5
x_1	a_4/y_1	a_1/y_2	a_4/y_4	a_5/y_2	a_4/y_1
x_2	a_3/y_2	a_5/y_3	a_2/y_1	a_4/y_3	a_3/y_2
x_3	a_1/y_3	a_3/y_4	a_1/y_3	a_2/y_1	a_1/y_4

Варіант №6					
X \ A	a_1	a_2	a_3	a_4	a_5
x_1	a_1/y_2	a_2/y_3	a_4/y_2	a_3/y_3	a_3/y_4
x_2	a_4/y_1	a_1/y_4	a_3/y_4	a_5/y_2	a_2/y_1
x_3	a_5/y_3	a_3/y_1	a_1/y_1	a_2/y_4	a_1/y_2

Варіант №7					
$X \backslash A$	a_1	a_2	a_3	a_4	a_5
x_1	a_3/y_2	a_4/y_2	a_1/y_3	a_2/y_4	a_2/y_1
x_2	a_2/y_1	a_1/y_1	a_3/y_2	a_1/y_2	a_3/y_1
x_3	a_5/y_4	a_5/y_4	a_3/y_4	a_4/y_1	a_5/y_4

Варіант 8					
$X \backslash A$	a_1	a_2	a_3	a_4	a_5
x_1	a_1/y_2	a_2/y_3	a_4/y_2	a_3/y_3	a_3/y_4
x_2	a_5/y_3	a_4/y_4	a_1/y_3	a_1/y_1	a_2/y_1
x_3	a_1/y_2	a_1/y_1	a_3/y_2	a_2/y_1	a_5/y_2

Варіант №9					
$X \backslash A$	a_1	a_2	a_3	a_4	a_5
x_1	a_2/y_1	a_3/y_1	a_4/y_2	a_1/y_1	a_3/y_3
x_2	a_1/y_3	a_5/y_2	a_5/y_3	a_4/y_2	a_1/y_1
x_3	a_4/y_2	a_1/y_3	a_2/y_3	a_5/y_4	a_3/y_2

Варіант №10					
$X \backslash A$	a_1	a_2	a_3	a_4	a_5
x_1	a_2/y_3	a_4/y_3	a_2/y_2	a_4/y_1	a_1/y_2
x_2	a_1/y_1	a_2/y_1	a_3/y_4	a_2/y_3	a_3/y_2
x_3	a_5/y_3	a_3/y_1	a_1/y_1	a_2/y_4	a_4/y_2

16.4 Контрольні запитання

1) Дайте визначення поняття «цифровий автомат»;
2) Дайте визначення поняття «абстрактний цифровий автомат»;
3) Множиною з яких шести елементів визначається абстрактний цифровий автомат;
4) У зв'язку з чим було введено поняття «стан автомата»?;
5) Якою системою рівнянь задається автомат Мілі?;
6) Який системою рівнянь задається автомат Мура?;
7) Який системою рівнянь задається С-автомат?;
8) Як розрізняють повністю визначені і часткові автомати?;
9) Який автомат називається ініціальний?;
10) Які способи завдання абстрактних автоматів Вам відомі?;
11) Поясніть, як заповнюються таблиці переходів, виходів для автомата Мілі, Мура та С-автомата?;
12) Поясніть, як заповнюється матриця з'єднань для автомата Мілі, Мура та С-автомата?;
13) Як заповнюється зазначена таблиця переходів для автоматів Мілі і Мура.

17 ЕКВІВАЛЕНТНІ ПЕРЕТВОРЕННЯ АВТОМАТІВ

17.1 Мета заняття

Засвоїти методи і набути практичних навичок еквівалентного перетворення автоматів Мура в автомати Мілі і навпаки.

17.2 Короткі основні відомості з теорії та вирішення типових завдань

Два автомата S_A і S_B з однаковими вхідними і вихідними алфавітами називаються еквівалентними, якщо після встановлення їх у початковий стан їх реакції на будь-яке вхідне слово співпадають.

Можна показати, що для будь-якого автомата Мілі існує еквівалентний йому автомат Мура і, навпаки, для будь-якого автомата Мура існує еквівалентний йому автомат Мілі.

При описанні алгоритмів взаємної трансформації автоматів Мілі і Мура у відповідності з викладеним вище ми будемо нехтувати в автоматах Мура вихідним сигналом, пов'язаним з початковим станом (λ (a_1)).

Розглянемо спочатку перетворення автомата Мура в автомат Мілі.

Нехай дано автомат Мура: S_A={ X_A, A_A,$У_A$, δ_A, λ_A,a_{oA}}, де:

X_A={x_1, x_2, ... x_n}; Y={y_1, y_2, ... y_m}; A={ a_0,a_1,a_2,... a_N}; a_{0A} = a_0 – початковий стан ($a_{0A} \in A$); δ_A - функція переходів автомата, що задає відображення (X_Ax$A_A \rightarrow A_A$; λ_A - функція виходів автомата, що задає відображення $A_A \rightarrow У_A$.

Побудуємо автомат Мілі: S_B={ X_B, A_B, Y_B, δ_B, λ_B, a_{0B}}, у якого A_B=A_A; X_B=X_A; Y_B=$У_A$; δ_B= δ_A; a_{0B}=a_{0A}.. Функцію виходів λ_B визначимо наступним чином. Якщо в автоматі Мура $\delta_A(a_m, x_1)$= a_s і λ_A (a_s)=y_g, то в автоматі Мілі λ_B (a_m, x_1)=y_g

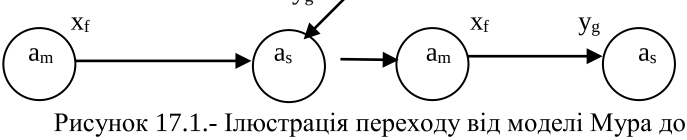


Рисунок 17.1.- Ілюстрація переходу від моделі Мура до моделі Мілі

Перехід від автомата Мура до автомата Мілі при графічному способі завдання ілюструється рис. 17.1. вихідний сигнал y_g записаний поруч з вершиною (a_s), переноситься на всі дуги, що входять в цю вершину.

При табличному способі завдання автомата таблиця переходів автомата Мілі збігається з таблицею переходів вихідного автомата Мура, а таблиця виходів виходить з таблиці переходів заміною символу a_s, що стоїть на перетині рядка x_f і стовпця a_m, символом вихідного сигналу y_g відзначає стовпець a_s в таблиці переходів автомата S_A.

З самого способу побудови автомата Мілі S_B очевидно, що він еквівалентний автомату Мура S_A. За індукції неважко показати, що будь-яке вхідне слово кінцевої довжини, подане на входи автоматів S_A і S_B, встановлених в стан a_m, викличе появу однакових вихідних слів і, отже, автомати SA і SB еквівалентні.

Перехід від автомата Мілі до автомата Мура.

Перш ніж розглянути трансформацію автомата Мілі в автомат Мура, накладемо на автомат Мілі наступне обмеження: у автомата не повинно бути перехідних (*переходящих – рус.)* станів. Під перехідним будемо розуміти стан, в який при заданні автомата у вигляді графа не входить жодна дуга, але яке має принаймні одну дугу, що виходить. Отже, нехай задано автомат Мілі:

$S_A=\{ X_A, A_A, Y_A, \delta_A, \lambda_A, a_{0A}\}$, де:

$X_A=\{x_1, x_2, ... x_n\}$; $Y=\{y_1, y_2, ... y_m\}$; $A=\{ a_0, a_1, a_2, ... a_N\}$; $a_{0A} = a_0$- початковий стан ($a_{0A} \in A$); δ_A - функція переходів автомата, що задає відображення ($X_A x A_A \rightarrow A_A$; λ_A - функція виходів автомата, що задає відображення $A_A \rightarrow У_A$.

Побудуємо автомат Мура: $S_B=\{X_B, A_B, Y_B, \delta_B, \lambda_B, a_{0B}\}$, у якого $X_B=X_A$; $Y_B=Y_A$.

Для визначення A_B кожному стану a_s A_A поставимо у відповідність множину A_s всіляких пар виду (a_s, y_g).

Функцію виходів δ_B визначимо наступним чином. Кожному стану автомата Мура S_B, який представляє собою пару вигляду (a_s, y_g), поставимо у відповідність вихідний сигнал y_g. Якщо в автоматі Мілі S_A був перехід $\delta_A(a_m, x_f) =a_s$ і при цьому видавався вихідний сигнал λ_A (a_m, x_f) = y_g, то в S_B буде перехід з множини станів A_m, породжуваних a_m, в стан (a_s, y_g) під дією вхідного сигналу x_f.

В якості початкового стану a_{0B} можна взяти будь-який з станів множини A_0, яке породжується початковим станом a_0 автомата S_A. При цьому вихідний сигнал в момент часу t = 0 не повинен враховуватися.

Розглянемо приклад. Нехай заданий автомат Мілі (табл. 17.1.)

Таблиця 17.1

X / A	x_1	x_2
a_0	a_2/y_1	a_0/y_1
a_1	a_0/y_1	a_2/y_2
a_3	a_0/y_2	a_1/y_1

Таблиця 17.2

X / A	x_1	x_2
a_0 b_0	a_2/y_1 b_{01}	a_0/y_1 b_{02}
a_1	a_0/y_1 b_{11}	a_2/y_2 b_{12}
a_2	a_0/y_2 b_{21}	a_1/y_1 b_{22}

Поставимо у відповідність кожній парі a_i/x_k стан b_{ik} (i-номер стану, k-номер вхідного сигналу), з урахуванням b_0.

Складемо таблицю переходів автомата Мура, керуючись наступними правилами:

1) Випишемо з таблиці 17.2 стани автомата Мілі і відповідні кожному з них множини станів автомата Мура (b_{ik}):

$a_0= \{b_0, b_{02}, b_{11}, b_{21}\}$; $a_1= \{b_{22}\}$; $a_2= \{b_{01}, b_{12}\}$;

2) Якщо стан автомата Мура b_{ik} входить до множини, яка відповідає стану a_p автомата Мілі, то в рядок таблиці переходів автомата Мура для стану b_{ik} слід записати рядок з таблиці переходів автомата Мілі, який відповідає стану a_p (з 17.1).

3) Функцію виходів автомата Мура визначимо наступним чином: $\lambda_B (b_{ik}) = \lambda_A (a_i, x_k)$. Для початкового стану b_0 значення вихідного сигналу можна вибрати довільно, але породжуваний початковим станом a_0 (з урахуванням поняття еквівалентності станів). Результуюча таблиця переходів і виходів автомата Мура еквівалентного автомату Мілі, заданому таблицею 17.1. представлена в таблиці 17.3.

4) Знайдемо в таблиці 17.3. еквівалентні стани і видалимо їх (замінимо на представника класу еквівалентності).

Якщо вихідний сигнал біля b_0 доозначити y_1, то виявиться, що в даній таблиці переходів знаходиться 3 еквівалентних стани (b_0, b_{11}, b_{02}). Замінивши клас еквівалентності одним представником (b_0), отримаємо остаточну таблицю переходів (табл. 17.4).

Таблиця 17.3

	x_1	x_2	Y
b_0	b_{01}	b_{02}	y_1
b_{01}	b_{21}	b_{22}	y_1
b_{02}	b_{01}	b_{02}	y_1
b_{11}	b_{01}	b_{02}	y_1
b_{12}	b_{21}	b_{22}	y_2
b_{21}	b_{01}	b_{02}	y_2
b_{22}	b_{11}	b_{12}	y_1

Таблиця 17.4.

	x_1	x_2	У
b_0	b_{01}	b_0	y_1
b_{01}	b_{21}	b_{22}	y_1
b_{12}	b_{21}	b_{22}	y_2
b_{21}	b_{01}	b_0	y_2
b_{22}	b_0	b_{12}	y_1

Викладені методи взаємної трансформації автоматів Мілі і Мура показують, що при переході від автомата Мура до автомата Мілі число станів автомата не змінюється, тоді як при зворотному переході число станів в автоматі Мура, як правило, зростає.

Задача 1.

Перетворити автомата Мура, заданий графом (рис. 17.2) в еквівалентний йому автомат Мілі.

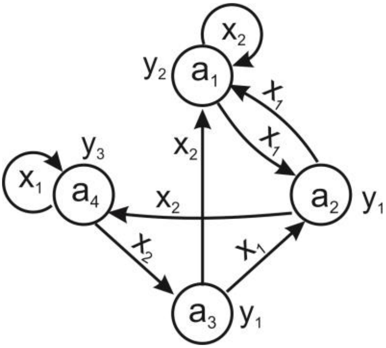


Рис. 17.2 Граф автомата Мура

Рішення

При цьому виді перетворення сигнал y_g, записаний поруч з вершиною a_S переноситься на всі дуги, що входять на цю вершину.

А це означає, що граф автомата Мілі, еквівалентний графу автомата Мура по рис. 17.2 буде виглядати наступним чином (див. рис. 17.3)

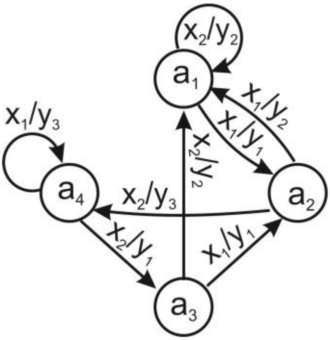


Рис. 17.3 Граф автомата Мілі

Щоб переконатися в еквівалентності перетворень знайдемо реакцію обох автоматів на довільне вхідне слово, наприклад слово X= $x_1\ x_2\ x_1\ x_2\ x_2\ x_1\ x_1$.

Реакція автомата Мура (див. рис. 17.2) на вхідне слово X:

Вхідне слово	x_1	x_2	x_1	x_2	x_2	x_1	x_1	
Стан	a_1	a_2	a_4	a_4	a_3	a_1	a_3	a_1
Вихідне слово	y_2	y_1	y_3	y_3	y_1	y_2	y_1	y_2

Реакція еквівалентного автомата Мілі (див. рис. 17.3) на вхідне слово X:

Вхідне слово	x_1	x_2	x_1	x_2	x_2	x_1	x_1	
Стан	a_1	a_2	a_4	a_4	a_3	a_1	a_3	a_1
Вихідне слово	y_1	y_3	y_3	y_1	y_2	y_1	y_2	

Без урахування реакції автомата Мура в початковому стані a_1/y_2, так як вона не залежить від x_1 в початковий момент часу, а визначається лише станом автомата в початковий момент часу (a_1), реакції автомата Мура та автомата Мілі є однаковими, а, значить, автомати еквівалентні.

Задача 2.

По графу автомата Мілі (см. рис. 17.3) скласти матрицю з’єднань.

Рішення

a(t) \ a(t+1)	a_1	a_2	a_3	a_4
a_1	x_2/y_2	x_1/y_2	x_2/y_2	-
a_2	x_1/y_1	-	x_1/y_1	-
a_3	-	-	-	x_3/y_1
a_4	-	x_2/y_3	-	x_1/y_3

Задача 3.

Використовуючи граф автомата Мілі (рис. 17.3) і матрицю з'єднань (зад. 2) скласти таблиці переходів і виходів, а також зазначену таблицю переходів.

Рішення

Таблиця переходів

A \ X	a_1	a_2	a_3	a_4
x_1	a_2	a_1	a_2	a_4
x_2	a_1	a_4	a_1	a_3

Таблиця виходів

A \ X	a_1	a_2	a_3	a_4
x_1	y_1	y_2	y_1	y_3
x_2	y_2	y_3	y_2	y_1

Зазначена таблиця переходів

A \ X	a_1	a_2	a_3	a_4
x_1	a_2 / y_1	a_1 / y_2	a_2 / y_1	a_4 / y_3
x_2	a_1 / y_2	a_4 / y_3	a_1 / y_2	a_3 / y_1

Задача 4.

Нехай автомат Мілі заданий зазначеною таблицею переходів (див. табл. 17.5)

Таблиця 17.5

X_A \ A_A	a_0	a_1	a_2
x_1	a_2 / y_1	a_0 / y_1	a_0 / y_2
x_2	a_0 / y_1	a_2 / y_2	a_1 / y_1

Потрібно побудувати еквівалентний йому автомат Мура.

Рішення

Поставимо у відповідність кожній парі a_i/x_k стан b_{ik} (i-номер стану, k-номер вхідного сигналу), з урахуванням b_0 (див. табл. 17.6).

Таблиця 17.6

X_A \ A_A	a_0 b_0	a_1	a_2
x_1	a_2 / y_1 b_{01}	a_0 / $y_{1\backslash}$ b_{11}	a_0 / y_2 b_{21}
x_2	a_0 / y_1 b_{02}	a_2 / y_2 b_{12}	a_1 / y_1 b_{22}

Складемо таблицю переходів автомата Мура, керуючись наступними правилами:

1) Випишемо з таблиці 17.6 стани автомата Мілі і відповідні кожному з них множини станів автомата Мура (b_{ik}).

$$a_0 = \{b_0, b_{02}, b_{11}, b_{21}\};\ a_1 = \{b_{22}\};\ a_2 = \{b_{01}, b_{12}\}.$$

2) Якщо стан автомата Мура b_{ik} входить в множину a_p автомата Мілі, то в стовпчик таблиці переходів автомата Мура для стану b_{ik} слід записати стовпчик з таблиці переходів автомата Мілі, який відповідає стану a_p (з табл. 17.6).

3) Функцію виходів автомата Мура визначимо наступним чином $\lambda_B=(b_{ik})=\lambda_A(a_i,x_k)$. Для початкового стану b_0 значення вихідного сигналу можна вибрати довільно, але породжуваний початковим станом a_0 (з урахуванням поняття еквівалентності стану).

Результуюча таблиця переходів і виходів автомата Мура еквівалентного автомату Мілі, заданому в табл. 17.6, представлена в табл. 17.7.

Таблиця 17.7

Y	y_1	y_1	y_1	y_1	y_2	y_2	y_1
A_B / X	b_0	b_{01}	b_{02}	b_{11}	b_{12}	b_{21}	b_{22}
x_1	b_{01}	b_{21}	b_{01}	b_{01}	b_{21}	b_{01}	b_{11}
x_2	b_{02}	b_{22}	b_{02}	b_{02}	b_{22}	b_{02}	b_{12}
	*		*	*			

4) Знайдемо в табл. 17.7 еквівалентні стани і видалимо їх (замінимо на представника класів еквівалентності).

Якщо вихідний сигнал біля b_0 довизначити y_1, то виявиться, що в даній таблиці переходів знаходиться три еквівалентних стани: $b_0 \equiv b_{02} \equiv b_{11}$. Замінивши клас еквівалентності одним представником (b_0), отримаємо остаточну зазначену таблицю переходів автомата Мура (табл. 17.8). При цьому стовпці, що відповідають еквівалентним станам (b_{02} і b_{11}) видаляємо з таблиці, а в інших стовпцях замінюємо їх представником класу еквівалентності, тобто b_0.

Таблиця 17.8

Y	y_1	y_1	y_2	y_2	y_1
A_B / X	b_0	b_{01}	b_{12}	b_{21}	b_{22}
x_1	b_{01}	b_{21}	b_{21}	b_{01}	b_0
x_2	b_0	b_{22}	b_{22}	b_0	b_{12}

17.3 Індивідуальні завдання

Завдання 1.

Абстрактний автомат Мура задано поєднаною таблицею:

Побудувати еквівалентний йому автомат Мілі, використовуючи табличний та графічний методи.

Варіант №1				
У	y_1	y_2	y_2	y_3
X \ A	a_1	a_2	a_3	a_4
x_1	a_1	a_4	a_2	a_3
x_2	a_2	a_3	a_3	a_2
x_3	a_3	a_4	a_1	a_1

Варіант №2				
У	y_1	y_3	y_2	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_2	a_1	a_2	a_1
x_2	a_3	a_2	a_4	a_2
x_3	a_4	a_4	a_1	a_3

Варіант №3				
У	y_2	y_3	y_1	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_4	a_1	a_4	a_2
x_2	a_3	a_2	a_2	a_4
x_3	a_1	a_4	a_1	a_3

Варіант №4				
У	y_1	y_2	y_3	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_3	a_1	a_2	a_1
x_2	a_2	a_3	a_3	a_4
x_3	a_4	a_4	a_1	a_3

Варіант №5				
У	Y_1	y_3	Y_2	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_4	a_3	a_1	a_1
x_2	a_1	a_4	a_4	a_2
x_3	a_2	a_4	a_3	a_3

Варіант №6				
У	y_1	y_2	y_3	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_3	a_1	a_2	a_1
x_2	a_2	a_4	a_4	a_3
x_3	a_1	a_2	a_1	a_4

Варіант №7				
У	y_2	y_3	y_1	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_4	a_3	a_2	a_1
x_2	a_1	a_4	a_3	a_4
x_3	a_2	a_1	a_1	a_3

Варіант №8				
У	y_1	y_2	y_3	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_3	a_1	a_2	a_1
x_2	a_2	a_4	a_4	a_3
x_3	a_3	a_3	a_1	a_4

Варіант №9				
У	y_2	y_3	y_1	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_1	a_3	a_2	a_2
x_2	a_3	a_4	a_3	a_4
x_3	a_2	a_4	a_1	a_1

Варіант №10				
У	y_1	y_2	y_3	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_3	a_1	a_2	a_4
x_2	a_2	a_2	a_4	a_2
x_3	a_4	a_3	a_1	a_1

Завдання 2.

Абстрактний автомат Мілі задано поєднаною таблицею:

Побудувати еквівалентний йому автомат Мура, використовуючи табличний метод.

Варіант №1					
A / X	a_1	a_2	a_3	a_4	a_5
x_1	a_1/y_1	a_2/y_2	a_4/y_4	a_5/y_2	a_4/y_1
x_2	a_3/y_2	a_5/y_1	a_3/y_2	a_4/y_3	a_2/y_3
x_3	a_4/y_3	a_3/y_4	a_1/y_3	a_2/y_1	a_1/y_4

Варіант №2					
A / X	a_1	a_2	a_3	a_4	a_5
x_1	a_1/y_2	a_2/y_3	a_4/y_2	a_3/y_4	a_4/y_4
x_2	a_2/y_3	a_3/y_4	a_5/y_4	a_2/y_2	a_3/y_1
x_3	a_5/y_4	a_4/y_1	a_1/y_1	A_5/y_4	a_1/y_2

Варіант №3					
A / X	a_1	a_2	a_3	a_4	a_5
x_1	a_4/y_1	a_1/y_2	a_5/y_4	a_5/y_1	a_4/y_1
x_2	a_1/y_2	a_5/y_4	a_2/y_2	a_4/y_3	a_2/y_2
x_3	a_3/y_3	a_3/y_3	a_1/y_3	a_3/y_2	a_1/y_4

Варіант №4					
A / X	a_1	a_2	a_3	a_4	a_5
x_1	a_1/y_1	a_1/y_2	a_4/y_2	a_5/y_3	a_3/y_4
x_2	a_4/y_1	a_4/y_5	a_5/y_4	a_3/y_2	a_2/y_3
x_3	a_5/y_3	a_3/y_1	a_1/y_3	a_2/y_4	a_1/y_2

Варіант №5					
A / X	a_1	a_2	a_3	a_4	a_5
x_1	a_4/y_1	a_1/y_2	a_4/y_4	a_5/y_2	a_4/y_1
x_2	a_3/y_1	a_5/y_3	a_4/y_2	a_4/y_3	a_3/y_2
x_3	a_1/y_3	a_3/y_4	a_5/y_1	a_2/y_1	a_1/y_4

Варіант №6					
A / X	a_1	a_2	a_3	a_4	a_5
x_1	a_1/y_2	a_5/y_3	a_4/y_2	a_3/y_3	a_3/y_4
x_2	a_4/y_1	a_1/y_4	a_2/y_4	a_5/y_2	a_2/y_1
x_3	a_5/y_3	a_3/y_1	a_1/y_1	a_2/y_4	a_1/y_2

Варіант №7					
A / X	a_1	a_2	a_3	a_4	a_5
x_1	a_3/y_2	a_4/y_2	a_1/y_3	a_2/y_4	a_2/y_1
x_2	a_2/y_1	a_1/y_1	a_3/y_2	a_1/y_2	a_3/y_1
x_3	a_5/y_4	a_5/y_4	a_3/y_4	a_4/y_1	a_5/y_4

Варіант 8					
A / X	a_1	a_2	a_3	a_4	a_5
x_1	a_1/y_2	a_2/y_3	a_4/y_2	a_3/y_3	a_3/y_4
x_2	a_5/y_3	a_4/y_4	a_1/y_3	a_1/y_1	a_2/y_1
x_3	a_1/y_2	a_1/y_1	a_3/y_2	a_2/y_1	a_5/y_2

Варіант №9					
X \ A	a_1	a_2	a_3	a_4	a_5
x_1	a_2/y_1	a_3/y_1	a_4/y_2	a_1/y_1	a_3/y_3
x_2	a_1/y_3	a_5/y_2	a_5/y_3	a_4/y_2	a_1/y_1
x_3	a_4/y_2	a_1/y_3	a_2/y_3	a_5/y_4	a_3/y_2

Варіант №10					
X \ A	a_1	a_2	a_3	a_4	a_5
x_1	a_2/y_3	a_4/y_3	a_2/y_2	a_4/y_1	a_1/y_2
x_2	a_1/y_1	a_2/y_1	a_3/y_4	a_2/y_3	a_3/y_2
x_3	a_5/y_3	a_3/y_1	a_1/y_1	a_2/y_4	a_4/y_2

17.4 Контрольні запитання

1) Дайте визначення поняття еквівалентності абстрактних автоматів.
2) Які алфавіти у еквівалентних автоматів збігаються, а які не збігаються?
3) Як визначаються функції виходів при перетворенні автомата Мура в автомат Мілі?
4) Як перетворити граф автомата Мура в еквівалентний граф автомата Мілі?
5) Як перетворити автомат Мура в еквівалентний автомат Мілі при табличному способі їх завдання?
6) Який стан автомата називається перехідним?
7) Чи допускається наявність перехідних станів в автоматі Мілі при його перетворенні в еквівалентний автомат Мура?
8) Як визначити число станів в еквівалентному автоматі Мура?
9) Як визначаються функції виходів при перетворенні автомата Мілі в автомат Мура?
10) Яким початковим станом можна задатися при перетворенні автомата Мура в автомат Мілі.

18 МІНІМІЗАЦІЯ ЧИСЛА ВНУТРІШНІХ СТАНІВ АВТОМАТА (АЛГОРИТМ АУФЕНКАМПА-ХОНА).

18.1 Мета заняття

Вивчити поняття еквівалентних станів абстрактного автомата. Засвоїти суть метода мінімізації числа внутрішніх станів абстрактного автомата Мілі і Мура з використанням алгоритму Ауфенкампа-Хона та набути практичних навичок шляхом вирішення типових задач.

18.2 Короткі основні відомості з теорії та вирішення типових завдань

Мінімізація числа внутрішніх станів автомата.

Алгоритм Ауфенкампа-Хона.

Основна ідея цього методу полягає в розбитті станів вихідного абстрактного автомата на класи еквівалентних станів, які попарно не перетинаються, і заміні кожного класу еквівалентності одним станом - представником даного класу.

Два стани автомата a_m і a_s називаються еквівалентними ($a_m \equiv a_s$), якщо $\lambda(a_m,X) = \lambda(a_s,X)$ для всіх можливих вхідних слів довжини X.

Якщо a_m і a_s не еквівалентні, вони різні. Більш слабкою еквівалентністю є k - еквівалентність. Стани a_m і a_s k-еквівалентні, якщо $\lambda (a_m, X_k) = \lambda (a_s, X_k)$ для всіх можливих вхідних слів довжини k.

При мінімізації числа внутрішніх станів автомата Мілі $S =\{X, Y, A, \lambda, \delta, a_0\}$ використовується алгоритм Ауфенкампа-Хона:

1. Знаходять послідовні розбиття $\pi_1, \pi_2, ..., \pi_k, \pi_{k+1}$, множини A на класи одно-, дво-, ..., k-, (k+1)-еквівалентних станів до тих пір, поки на якомусь (k +1) кроці не виявиться, що $\pi_k = \pi_{k+1}$. У цьому випадку k-еквівалентні стани є еквівалентними. Число кроків k, при якому $\pi_k = \pi_{k+1}$, не перевищує N-1, де N-число внутрішніх станів автомата.

2. У кожному класі еквівалентності π вибирають по одному елементу (представнику класу), які утворюють множини A' станів мінімального автомата S'.

3. Функцію переходів δ' і виходів λ' автомата S' визначають на множині A'xX. Для цього в таблиці переходів і виходів викреслюють стовпці станів, які не увійшли

до множини A', а в решті стовпців таблиці переходів всі стани замінюються на еквівалентні з множини A', (на представників).

4. В якості a'_0 вибирається один зі станів, еквівалентних стану a_0. Зокрема, зручно прийняти сам стан a_0.

При мінімізації автомата Мура вводиться поняття 0-еквівалентності станів і розбиття множини станів на 0-класи: 0-еквівалентними називаються будь-які, однаково зазначені вихідними сигналами, стани автомата Мура.

Задача 1.

Мінімізувати число внутрішніх станів автомата Мура, заданого таблицею переходів і виходів (Таблиця 18.1.), використовуючи алгоритм Ауфенкампа-Хона.

Таблиця 18.1

	y_2	y_3	y_4	y_1	y_4	y_2	y_1	y_4	y_2	y_3
A	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
x_1	5	2	9	5	3	1	2	6	5	10
x_2	9	1	6	7	2	3	4	10	1	9
x_3	2	3	8	6	4	9	7	10	10	3

Виконаємо розбиття π_0:

$\pi_0=\{B_1,B_2,B_3,B_4\}$;

$B_1=\{a_4, a_7\}$, $B_2=\{a_1, a_6, a_9\}$, $B_3=\{a_2, a_{10}\}$, $B_4=\{a_3, a_5, a_8\}$.

Побудуємо таблицю розбиття π_0:

	B_1		B_2			B_3		B_4		
A	a_4	a_7	a_1	a_6	a_9	a_2	a_{10}	a_3	a_5	a_8
x_1	B_4	B_3	B_4	B_2	B_4	B_3	B_3	B_2	B_4	B_2
x_2	B_1	B_1	B_2	B_4	B_2	B_2	B_2	B_2	B_3	B_3
x_3	B_2	B_1	B_3	B_2	B_3	B_4	B_4	B_4	B_1	B_3

Виконаємо розбиття π_1:

$\pi_1=\{C_1,C_2,C_3,C_4,C_5,C_6,C_7\}$;

$C_1=\{a_4\}$, $C_2=\{a_7\}$, $C_3=\{a_1, a_9\}$, $C_4=\{a_6\}$, $C_5=\{a_2, a_{10}\}$, $C_6=\{a_3\}$, $C_7=\{a_5\}$, $C_8=\{a_8\}$.

	C_1	C_2	C_3		C_4	C_5		C_6	C_7	C_8
A	a_4	a_7	a_1	a_9	a_6	a_2	a_{10}	a_3	a_5	a_8
x_1	C_7	C_5	C_7	C_7	C_3	C_5	C_5	C_3	C_6	C_4
x_2	C_2	C_1	C_3	C_3	C_6	C_3	C_3	C_4	C_5	C_5
x_3	C_4	C_2	C_5	C_5	C_3	C_6	C_6	C_8	C_1	C_5

Виконаємо розбиття π_2:

$\pi_2=\{D_1,D_2,D_3,D_4,D_5,D_6,D_7,D_8\}$;

$D_1=\{a_4\}$, $D_2=\{a_7\}$, $D_3=\{a_1,a_9\}$, $D_4=\{a_6\}$, $D_5=\{a_2,a_{10}\}$, $D_6=\{a_3\}$, $D_7=\{a_5\}$, $D_8=\{a_8\}$.

Розбиття π_2 повторює розбиття π_1 – процедуру розбиття завершено.

Можна зробити висновок: $a_1 \equiv a_9$; $a_2 \equiv a_{10}$;

Виберемо довільно з кожного класу еквівалентності D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8 по одному представнику - в даному випадку з мінімальним номером:

$A' = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8\}$.

Видаляючи з вихідної таблиці переходів "зайві" стани (a_9,a_{10}), визначаємо мінімальний автомат Мура:

	y_2	y_3	y_4	y_1	y_4	y_2	y_1	y_4
A	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
x_1	5	2	1	5	3	1	2	6
x_2	1	1	6	7	2	3	4	2
x_3	2	3	8	6	4	1	7	2

Задача 2.

Мінімізувати число внутрішніх станів автомата Мілі, заданого таблицею переходів і виходів (Таблиця 18.2.), використовуючи алгоритм Ауфенкампа-Хона.

Таблиця 18.2

A	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
x_1	a_1/y_2	a_0/y_1	a_2/y_4	a_3/y_2	a_5/y_1	a_0/y_1	a_1/y_2	a_7/y_4	a_5/y_1
x_2	a_3/ y_3	a_5/ y_2	a_3/ y_1	a_8/ y_3	a_3/ y_2	a_2/ y_2	a_3/ y_3	a_3/ y_1	a_3/ y_2
x_3	a_7/ y_4	a_6/ y_4	a_4/ y_2	a_4/ y_1	a_8/ y_3	a_8/ y_4	a_2/ y_4	a_8/ y_2	a_4/ y_3

Виконаємо розбиття π_1:

$\pi_1=\{B_1, B_2, B_3, B_4, B_5\}$;

$B_1=\{a_0, a_6\}$, $B_2=\{a_1, a_5\}$, $B_3=\{a_2, a_7\}$,. $B_4=\{a_3\}$, $B_5=\{a_4, a_8\}$.

Побудуємо таблицю розбиття π_1:

	B_1		B_2		B_3		B_4	B_5	
A	a_0	a_6	a_1	a_5	a_2	a_7	a_3	a_4	a_8
x_1	B_2	B_2	B_1	B_1	B_3	B_3	B_4	B_2	B_2
x_2	B_4	B_4	B_2	B_3	B_4	B_4	B_5	B_4	B_4
x_3	B_3	B_3	B_3	B_5	B_5	B_5	B_5	B_5	B_5

Виконаємо розбиття π_2:

$\pi_2=\{C_1, C_2, C_3, C_4, C_5, C_6\}$;

$C_1=\{a_0, a_6\}$, $C_2=\{a_1\}$, $C_3=\{a_5\}$, $C_4=\{a_2, a_7\}$, $C_5=\{a_3\}$, $C_6=\{a_2, a_8\}$

Побудуємо таблицю розбиття π_2:

	C_1		C_2	C_3	C_4		C_5	C_6	
A	a_0	a_6	a_1	a_5	a_2	a_7	a_3	a_4	a_8
x_1	C_2	C_2	C_1	C_1	C_4	C_4	C_5	C_3	C_3
x_2	C_5	C_5	C_3	C_4	C_5	C_5	C_6	C_5	C_5
x_3	C_4	C_4	C_1	C_6	C_6	C_6	C_8	C_6	C_6

Виконаємо розбиття π_3:

$\pi_3=\{D_1, D_2, D_3, D_4, D_5, D_6\}$;

$D_1=\{a_0, a_6\}$, $D_2=\{a_1\}$, $D_3=\{a_5\}$, $D_4=\{a_2, a_7\}$, $D_5=\{a_3\}$, $D_6=\{a_4, a_8\}$

Розбиття π_3 повторює розбиття π_2 – процедуру розбиття завершено.

Можна зробити висновок: $a_0 \equiv a_6$; $a_2 \equiv a_7$; $a_4 \equiv a_8$.

Виберемо довільно з кожного класу еквівалентності D_1, D_2, D_3, D_4, D_5, D_6 по одному представнику - в даному випадку з мінімальним номером:

$A' = \{a_0, a_1, a_2, a_3, a_4, a_5\}$.

Видаляючи з вихідної таблиці переходів "зайві" стани, визначаємо мінімальний автомат Мілі:

Таблиця 18.3

A	a_0	a_1	a_2	a_3	a_4	a_5
x_1	a_1/y_2	a_0/y_1	a_2/y_4	a_3/y_2	a_5/y_1	a_0/y_1
x_2	a_3/y_3	a_5/y_2	a_3/y_1	a_4/y_3	a_3/y_2	a_2/y_2
x_3	a_2/y_4	a_0/y_4	a_4/y_2	a_4/y_1	a_4/y_3	a_4/y_4

18.3 Індивідуальні завдання

Завдання 1.

Мінімізувати число внутрішніх станів автомата Мура, використовуючи алгоритм Ауфенкампа-Хона. Побудувати граф переходів мінімального автомата.

Варіант № 1								
Y	y_1	y_2	y_3	y_1	y_1	y_2	y_3	y_4
A \ X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_2	a_0	a_1	a_6	a_4	a_3	a_5	a_5
x_2	a_3	a_4	a_6	a_3	a_6	a_4	a_2	a_7
x_3	a_5	a_5	a_7	a_1	a_5	a_1	a_7	a_0

Варіант № 2								
Y	y_1	y_3	y_2	y_4	y_1	y_2	y_3	y_4
A \ X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_0	a_6	a_3	a_0	a_4	a_5	a_6	a_4
x_2	a_4	a_3	a_4	a_6	a_0	a_2	a_3	a_6
x_3	a_5	a_1	a_1	a_5	a_5	a_7	a_7	a_5

Варіант № 3								
У	y_1	y_2	y_3	y_1	y_1	y_2	y_3	y_4
A \ X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_3	a_4	a_1	a_3	a_4	a_4	a_5	a_5
x_2	a_2	a_0	a_6	a_6	a_6	a_3	a_2	a_6
x_3	a_5	a_5	a_7	a_1	a_5	a_1	a_7	a_3

Варіант № 4								
У	y_1	y_3	y_2	y_4	y_1	y_2	y_3	y_4
A \ X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_0	a_3	a_3	a_0	a_4	a_5	a_3	a_4
x_2	a_4	a_6	a_4	a_6	a_0	a_2	a_6	a_6
x_3	a_5	a_1	a_1	a_5	a_5	a_7	a_7	a_5

Варіант № 5								
У	y_1	y_2	y_3	y_4	y_5	y_2	y_3	y_4
A \ X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_3	a_4	a_1	a_7	a_4	a_6	a_1	a_3
x_2	a_6	a_6	a_6	a_3	a_0	a_2	a_6	a_7
x_3	a_1	a_5	a_7	a_1	a_5	a_7	a_7	a_1

Варіант № 6								
У	y_1	y_3	y_2	y_4	y_5	y_2	y_3	y_5
A \ X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_6	a_2	a_7	a_4	a_2	a_6	a_3
x_2	a_3	a_0	a_5	a_3	a_0	a_5	a_0	a_7
x_3	a_6	a_2	a_3	a_1	a_5	a_3	a_5	a_1

Варіант № 7								
У	y_1	y_2	y_3	y_4	y_5	y_2	y_3	y_4
A \ X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_3	a_4	a_1	a_6	a_4	a_6	a_1	a_3
x_2	a_7	a_7	a_7	a_3	a_0	a_2	a_6	a_6
x_3	a_1	a_5	a_6	a_1	a_5	a_7	a_6	a_1

Варіант № 8								
У	y_1	y_3	y_2	y_4	y_5	y_2	y_3	y_5
A \ X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_6	a_2	a_3	a_4	a_2	a_6	a_7
x_2	a_7	a_0	a_5	a_7	a_0	a_5	a_0	a_3
x_3	a_6	a_2	a_7	a_1	a_5	a_7	a_5	a_1

Варіант № 9								
У	y_1	y_2	y_3	y_4	y_5	y_2	y_3	y_4
A \ X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_3	a_4	a_1	a_7	a_4	a_6	a_1	a_3
x_2	a_6	a_6	a_6	a_3	a_0	a_2	a_6	a_7
x_3	a_1	a_5	a_7	a_1	a_5	a_7	a_7	a_1

Варіант № 10								
У	y_1	y_3	y_2	y_4	y_5	y_2	y_3	y_5
A \ X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_6	a_2	a_7	a_4	a_2	a_6	a_3
x_2	a_3	a_0	a_5	a_3	a_0	a_5	a_0	a_7
x_3	a_6	a_2	a_3	a_1	a_5	a_3	a_5	a_1

Завдання 2.

Мінімізувати число внутрішніх станів автомата Мілі, використовуючи алгоритм Ауфенкампа-Хона. Побудувати граф переходів мінімального автомата.

Варіант №1						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_1	a_2/y_2	a_4/y_4	a_5/y_2	a_5/y_1	a_6/y_2
x_2	a_5/y_2	a_5/y_1	a_3/y_2	a_4/y_3	a_1/y_2	a_1/y_1
x_3	a_4/y_3	a_3/y_4	a_1/y_3	a_2/y_1	a_4/y_3	a_3/y_4

Варіант №2						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_2	a_2/y_3	a_4/y_2	a_3/y_4	a_4/y_2	a_4/y_4
x_2	a_2/y_3	a_3/y_4	a_5/y_3	a_2/y_2	a_3/y_3	a_3/y_1
x_3	a_5/y_4	a_4/y_1	a_1/y_1	a_5/y_4	a_1/y_1	a_1/y_2

Варіант №3						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_4/y_1	a_1/y_2	a_5/y_4	a_5/y_1	a_4/y_1	a_6/y_2
x_2	a_1/y_2	a_5/y_4	a_2/y_1	a_4/y_3	a_3/y_2	a_1/y_1
x_3	a_3/y_3	a_3/y_3	a_1/y_3	a_3/y_2	a_1/y_3	a_3/y_4

Варіант №4						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_2	a_1/y_2	a_1/y_2	a_5/y_3	a_3/y_4	a_2/y_4
x_2	a_4/y_1	a_2/y_3	a_2/y_3	a_3/y_2	a_2/y_1	a_3/y_1
x_3	a_5/y_3	a_5/y_1	a_6/y_1	a_2/y_4	a_1/y_2	a_1/y_2

Варіант №5						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_2/y_1	a_1/y_2	a_4/y_4	a_5/y_2	a_6/y_1	a_1/y_2
x_2	a_3/y_2	a_5/y_3	a_2/y_1	a_4/y_3	a_3/y_2	a_1/y_3
x_3	a_1/y_3	a_3/y_4	a_1/y_3	a_2/y_1	a_1/y_3	a_3/y_4

Варіант №6						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_2	a_2/y_3	a_4/y_2	a_4/y_3	a_2/y_2	a_4/y_4
x_2	a_2/y_3	a_3/y_4	a_5/y_3	a_5/y_4	a_3/y_3	a_3/y_1
x_3	a_6/y_4	a_4/y_1	a_1/y_1	a_4/y_1	a_1/y_1	a_6/y_2

Варіант №7						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_4/y_1	a_1/y_2	a_5/y_4	a_5/y_1	a_4/y_1	a_6/y_2
x_2	a_1/y_2	a_5/y_4	a_2/y_1	a_4/y_3	a_3/y_2	a_1/y_1
x_3	a_3/y_3	a_3/y_3	a_1/y_3	a_3/y_2	a_1/y_3	a_3/y_4

Варіант №8						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_2	a_1/y_2	a_1/y_2	a_5/y_3	a_3/y_4	a_2/y_4
x_2	a_4/y_1	a_2/y_3	a_2/y_3	a_3/y_2	a_2/y_1	a_3/y_1
x_3	a_5/y_3	a_5/y_1	a_6/y_1	a_2/y_4	a_1/y_2	a_1/y_2

Варіант №9						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_2/y_1	a_1/y_2	a_4/y_4	a_5/y_2	a_6/y_1	a_1/y_2
x_2	a_3/y_2	a_5/y_3	a_2/y_1	a_4/y_3	a_3/y_2	a_1/y_3
x_3	a_1/y_3	a_3/y_4	a_1/y_3	a_2/y_1	a_1/y_3	a_3/y_4

Варіант №10						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_2	a_2/y_3	a_4/y_2	a_4/y_3	a_2/y_2	a_4/y_4
x_2	a_2/y_3	a_3/y_4	a_5/y_3	a_5/y_4	a_3/y_3	a_3/y_1
x_3	a_6/y_4	a_4/y_1	a_1/y_1	a_4/y_1	a_1/y_1	a_6/y_2

18.4 Контрольні запитання

1) В чому полягає завдання мінімізації абстрактного автомата?

2) Які стани a_m та a_s називаються еквівалентними?

3) Дайте означення k-еквівалентності?

4) Сформулюйте алгоритм Ауфенкампа-Хона для мінімізації внутрішніх станів автомата Мілі.

5) Сформулюйте алгоритм Ауфенкампа-Хона для мінімізації внутрішніх станів автомата Мура.

6) Що таке 0-еквівалентність станів автомата Мура?

19 КАНОНІЧНИЙ МЕТОД СТРУКТУРНОГО СИНТЕЗУ АВТОМАТІВ.

19.1 Мета заняття

Засвоїти суть канонічного методу синтезу структурних автоматів та набути практичних навичок складання структурних схем автоматів Мілі та Мура, виходячи із заданих алфавітів абстрактних автоматів Мілі та Мура, а також запису системи канонічних рівнянь в загальному вигляді.

Вивчити основні етапи канонічного методу структурного синтезу автоматів. Ознайомитися з основними принципами побудови структурної схеми автомата та запису канонічних рівнянь у загальному виді.

19.2 Короткі основні відомості з теорії та вирішення типових завдань

19.2.1. Структурні автомати. Основні поняття. Канонічний метод структурного синтезу автоматів. Теорема Глушкова про структурну повноту.

Наступним етапом абстрактного синтезу автоматів, що закінчується мінімізацією числа станів, слідує етап структурного синтезу, метою якого є побудова схеми, що реалізує автомат з логічних елементів заданого типу. У структурному автоматі враховується структура вхідних і вихідних сигналів автомата, а також його внутрішня побудова на рівні структурних схем. Основним завданням структурної теорії автоматів є знаходження загальних прийомів побудови структурних схем автоматів на основі композиції елементарних автоматів, що належать до заздалегідь заданого кінцевого числа типів.

У структурній теорії як вхідні так і вихідні канали вважаються складаються з елементарних вхідних (вихідних) каналів. По всіх елементарних вхідних (вихідних) каналах можуть передаватися тільки елементарні сигнали.

Набір можливих значень сигналів, що подаються на один зовнішній вхідний (вихідний) вузол, називається структурним вхідним (вихідним) алфавітом автомата. Алфавіт повинен бути обмеженим.

Вхідний і вихідний сигнали задаються обмеженими впорядкованими наборами елементарних сигналів, які називаються векторами, а складові їх елементарні сигнали - компоненти векторів. Число компонент вектора - це розмірність алфавіту.

Наприклад, $X=\{x_1, x_2, x_3, x_4, x_5\}$ - вхідний алфавіт абстрактного автомата.

Структурний вхідний алфавіт, розмірність якого дорівнює трьом:

$x_1 = 000$, $x_2 = 001$, $x_3 = 010$, $x_4 = 011$, $x_5 = 100$.

Векторне представлення вхідних і вихідних сигналів називається структурним вхідним вихідним сигналом, відповідно.

На етапі структурного синтезу попередньо вибираються елементарні автомати, з яких потім шляхом їх композиції будується структурна схема отриманого на етапі абстрактного синтезу автомата Мілі, Мура або С-автомата. Якщо рішення задачі структурного синтезу існує, говорять, що задана система автоматів структурно повна. Звичайно застосовується, так званий, канонічний метод структурного синтезу, при якому використовуються елементарні автомати деякого спеціального виду: автомати з пам'яттю, що мають більше одного стану, і автомати без пам'яті - з одним станом. Автомати першого класу носять назву елементів пам'яті, а автомати другого класу - комбінаційних або логічних елементів.

Теоретичним обґрунтуванням канонічного методу структурного синтезу автоматів є теорема про структурну повноту (теорема Глушкова):

Усяка система елементарних автоматів, яка містить автомат Мура з нетривіальною пам'яттю, який має повну систему переходів і повну систему виходів, і яку-небудь функціонально повну систему логічних елементів, є структурно повною.

Існує загальний конструктивний прийом (канонічний метод структурного синтезу), що дозволяє в розглянутому випадку звести задачу структурного синтезу довільних автоматів до задачі синтезу комбінаційних схем.

Результатом канонічного методу структурного синтезу є система логічних рівнянь, що виражає залежність вихідних сигналів автомата (функції виходів автомата) і сигналів, що подаються на входи елементів пам’яті, від сигналів, що приходять на вхід всього автомата в цілому, і сигналів, що знімаються з виходу елементів пам'яті (функції збудження елементів пам'яті автомата). Ці рівняння називаються **канонічними**.

Для правильної роботи схем, очевидно, не можна дозволяти, щоб сигнали на вході елементів пам'яті безпосередньо брали участь в утворенні вихідних сигналів, які по ланцюгах зворотного зв'язку подавалися б у той же самий момент часу на ці входи. У зв'язку з цим елементами пам'яті повинні бути не автомати Мілі, а автомати Мура.

Таким чином, структурно повна система елементарних автоматів повинна містити хоча б один автомат Мура. У той же час для синтезу будь-яких автоматів з мінімальним числом елементів пам'яті необхідно в якості таких елементів вибирати автомати Мура, які мають повну систему переходів і повну систему виходів - так звані повні автомати.

Повнота системи переходів означає, що для будь-якої пари станів (a_m, a_s) автомата знайдеться вхідний сигнал, який переводить перший елемент цієї пари a_m в другий - a_s тобто в такому автоматі в кожному стовпці таблиці переходів повинні зустрічатися всі стани автомата. Повнота системи виходів автомата Мура полягає в тому, що кожному стану автомата поставлений у відповідність свій особливий вихідний сигнал, відмінний від вихідних сигналів інших станів. Очевидно, що в такому автоматі число вихідних сигналів дорівнює числу станів автомата. Разом з попереднім затвердженням це призводить до того, що в автоматі Мура з повною системою виходів можна ототожнити стани автомата з його вихідними сигналами. У зв'язку з цим в автоматах пам'яті ми будемо використовувати одні й ті ж позначення і для станів, і для вихідних сигналів, тобто зазначена таблиця переходів в автоматах Мура з повною системою виходів перетворюється просто в таблицю переходів.

Наявність функціонально повної системи логічних елементів дозволяє реалізувати булеву функцію будь-якого ступеня складності.

Після вибору елементів пам'яті і кодування станів синтез структурного автомата зводиться до синтезу комбінаційної схеми, що реалізує канонічні рівняння.

19.2.2. Основні етапи канонічного методу структурного синтезу

У канонічному методі структурного синтезу можна виділити кілька основних етапів:

1. Кодування алфавітів автомата.
2. Вибір елементів пам'яті.
3. Вибір функціонально повної системи логічних елементів.
4. Запис і мінімізація канонічних рівнянь.
5. Побудова функціональної схеми автомата.

Вихідними даними для початку роботи даного методу є абстрактний цифровий автомат з пам'яттю, заданий таблицею переходів і виходів.

Задача 1.

Абстрактний автомат Мура задано наступними алфавітами:

$X=\{x_1, x_2, x_3\}$;

$Y=\{y_1, y_2, y_3\}$;

$A=\{a_1, a_2, a_3, a_4, a_5\}$.

Розробити структурну схему автомата та записати в загальному вигляді систему канонічних рівнянь для даного автомата.

Рішення.

1.Визначимо число компонент двійкових векторів, якими можливо закодувати елементи вхідного, вихідного алфавітів та алфавіту станів.

Число компонент двійкового вектора вхідного алфавіту визначається наступним чином:

$K_{вх} \geq int(\log_2 |X|)$;

де int(X) – округлення до найближчого цілого числа в бік збільшення; |X| - потужність вхідного алфавіту (кількість букв вхідного алфавіту абстрактного автомата).

В нашому випадку $|X|=3$, тобто $K_{вх} \geq int(\log_2 |3|)=2$;

Приймаємо $K_{вх} =2$;

Позначимо елементи двійкового вектора вхідного алфавіту структурного автомата, як $Z=\{z_1, z_2\}$.

Аналогічно визначаємо число компонент двійкового вектора вихідного алфавіту структурного автомата:

$K_{вих} \geq int(\log_2 |Y|)$; де $|Y|=3$;

$K_{вих} \geq int(\log_2 |3|)=2$; $K_{вих}=2$;

Позначимо елементи двійкового вектора вихідного алфавіту структурного автомата, як $W=\{w_1, w_2\}$.

Аналогічно визначаємо число компонент двійкового вектора алфавіту станів структурного автомата:

$K_{стан} \geq int(\log_2 |A|)$;   де $|A|=6$; $K_{стан} \geq int(\log_2 |6|)=3$;

$K_{стан}=3$;

Ця величина визначає кількість елементів пам'яті в структурному автоматі. Їх повинно бути 3.

Позначимо елементи двійкового вектора алфавіту станів структурного автомата, як А`={α_1, α_2, α_3}.

2. Виходячи із узагальненої структурної схеми автомата Мура, будуємо структурну схему автомата, заданого умовами даної задачі (Рис. 19.1.).

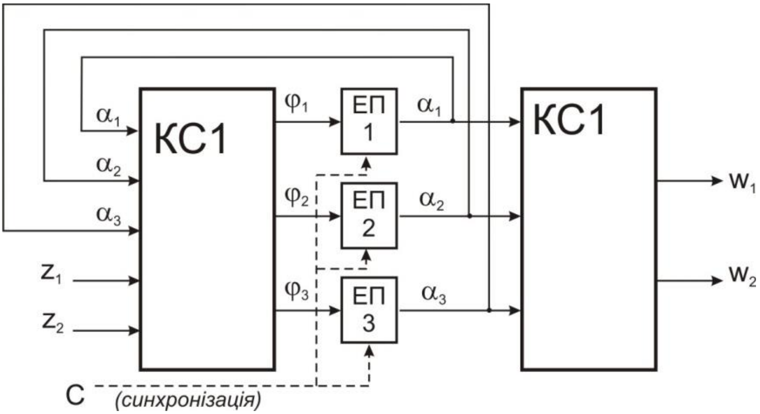


Рис. 19.1. Структурна схема автомата Мура.

Де, КС 1 – комбінаційна схема формування функцій збудження входів елементів пам'яті автомата,

КС 2 – комбінаційна схема, яка формує функції виходів елементів пам'яті автомата;

ЕП 1, ЕП 2, ЕП 3 – елементи пам'яті автомата.

3. Запишемо систему канонічних рівнянь, у відповідності до структурної схеми (Рис. 19.1.)

$\varphi_1 = \varphi_1(z_1, z_2, \alpha_1, \alpha_2, \alpha_3)$

$\varphi_2 = \varphi_2(z_1, z_2, \alpha_1, \alpha_2, \alpha_3)$

$\varphi_3 = \varphi_2(z_1, z_2, \alpha_1, \alpha_2, \alpha_3)$

$w_1 = w_1 (\alpha_1, \alpha_2, \alpha_3)$

$w_2 = w_2 (\alpha_1, \alpha_2, \alpha_3)$

Якщо відомі таблиці кодування всіх алфавітів структурного автомата, а також властивості елементів пам'яті, то синтез структурного автомата зводиться до синтезу комбінаційних схем КС1 і КС2, структура яких і визначається системою канонічних рівнянь.

Задача 2.

Абстрактний автомат Мілі задано наступними алфавітами:

$X=\{x_1, x_2, x_3, x_4, x_5\}$;

$Y=\{y_1, y_2, y_3, y_4\}$;

$A=\{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9\}$.

Розробити структурну схему автомата та записати в загальному вигляді систему канонічних рівнянь для даного автомата.

Рішення.

1.Визначимо число компонент двійкових векторів, якими можливо закодувати елементи вхідного, вихідного алфавітів та алфавіту станів.

Число компонент двійкового вектора вхідного алфавіту визначається наступним чином:

$К_{вх} \geq int(\log_2 |X|)$;

де int(X) – округлення до найближчого цілого числа в бік збільшення;
|X| - потужність вхідного алфавіту (кількість букв вхідного алфавіту абстрактного автомата).

В нашому випадку $|X|=5$, тобто $К_{вх} \geq int(\log_2 |5|)=3$;

Приймаємо $К_{вх} =3$;

Позначимо елементи двійкового вектора вхідного алфавіту структурного автомата, як $Z=\{z_1, z_2, z_3\}$.

Аналогічно визначаємо число компонент двійкового вектора вихідного алфавіту структурного автомата:

$К_{вих} \geq int(\log_2 |Y|)$; де $|Y|=4$;

$К_{вих} \geq int(\log_2 |4|)=2$; $К_{вих}=2$;

Позначимо елементи двійкового вектора вихідного алфавіту структурного автомата, як $W=\{w_1, w_2\}$.

Аналогічно визначаємо число компонент двійкового вектора алфавіту станів структурного автомата:

$К_{стан} \geq int(\log_2 |A|)$; де $|A|=9$; $К_{стан} \geq int(\log_2 |9|)=4$;

$К_{стан}=4$;

Ця величина визначає кількість елементів пам’яті в структурному автоматі. Їх повинно бути 4.

Позначимо елементи двійкового вектора алфавіту станів структурного автомата, як $A`=\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$.

2. Виходячи із узагальненої структурної схеми автомата Мілі, будуємо структурну схему автомата, заданого умовами даної задачі (Рис. 19.2.).

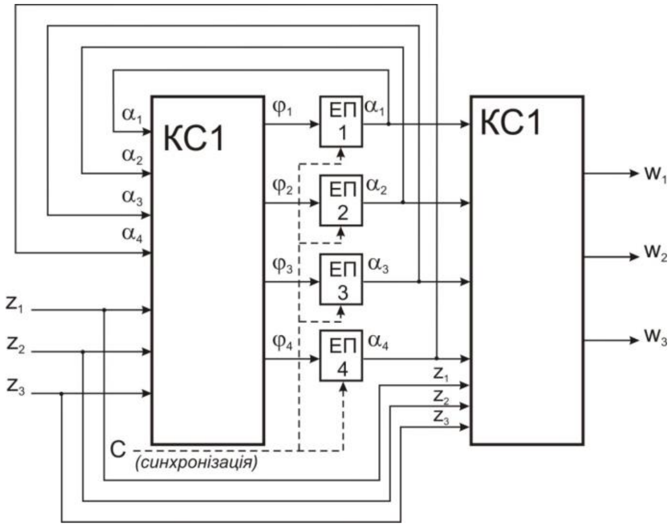


Рис. 19.2. Структурна схема автомата Мілі.

Де, КС 1 – комбінаційна схема формування функцій збудження входів елементів пам’яті автомата,

КС 2 – комбінаційна схема, яка формує функції виходів елементів пам’яті автомата Мілі. Зі схеми видно, що функції виходів автомата Мілі залежать не лише від станів автомата, але й від вхідних сигналів.

ЕП 1, ЕП 2, ЕП 3, ЕП 4 – елементи пам’яті автомата.

Слід відзначити, що в реальних схемах функції елементів пам’яті виконують прості елементи пам’яті, які отримали назву тригери. Елементи памяті, виконані у вигляді тригерів, можуть мати один, два та більше інформаційних входів і, як правило, два виходи (прямий та інверсний). Тому число виходів із КС1 може бути в 2 рази (та навіть більше) більшим.

Оскільки в схемах на рис 19.1 та 19.2 є зворотні зв’язки з виходів ЕП на їхні входи через КС1, то можлива втрата стійкості в роботі (гонки в автоматах). Для забезпечення стійкої роботи структурних автоматів ЕП виконують, як правило, у вигляді синхронних тригерів із внутрішньою затримкою інформації (2-х ступеневі тригери), і тому на схемах (рис. 19.1., 19.2.) вводиться сигнал синхронізації (С).

3. Запишемо систему канонічних рівнянь, у відповідності до структурної схеми (Рис. 19.2.)

$\varphi_1 = \varphi_1(z_1, z_2, z_3, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$

$\varphi_2 = \varphi_2(z_1, z_2, z_3, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$

$\varphi_3 = \varphi_3(z_1, z_2, z_3, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$
$\varphi_4 = \varphi_4(z_1, z_2, z_3, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$
$w_1 = w_1\ (z_1, z_2, z_3, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$
$w_2 = w_2\ (z_1, z_2, z_3, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$

Якщо відомі таблиці кодування всіх алфавітів структурного автомата, а також властивості елементів пам'яті, то синтез структурного автомата зводиться до синтезу комбінаційних схем КС1 і КС2, структура яких і визначається системою канонічних рівнянь.

19.3 Індивідуальні завдання

Завдання 1.

Розробити структурну схему автомата Мура та записати в загальному вигляді систему канонічних рівнянь для даного автомата.

Варіант № 1								
У	y_1	y_2	y_3	y_1	y_1	y_2	y_3	y_4
A X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_2	a_0	a_1	a_6	a_4	a_3	a_5	a_5
x_2	a_3	a_4	a_6	a_3	a_6	a_4	a_2	a_7
x_3	a_5	a_5	a_7	a_1	a_5	a_1	a_7	a_0

Варіант № 2								
У	y_1	y_3	y_2	y_4	y_1	y_2	y_3	y_4
A X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_0	a_6	a_3	a_0	a_4	a_5	a_6	a_4
x_2	a_4	a_3	a_4	a_6	a_0	a_2	a_3	a_6
x_3	a_5	a_1	a_1	a_5	a_5	a_7	a_7	a_5

Варіант № 3								
У	y_1	y_2	y_3	y_1	y_1	y_2	y_3	y_4
A X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_3	a_4	a_1	a_3	a_4	a_4	a_5	a_5
x_2	a_2	a_0	a_6	a_6	a_6	a_3	a_2	a_6
x_3	a_5	a_5	a_7	a_1	a_5	a_1	a_7	a_3

Варіант № 4								
У	y_1	y_3	y_2	y_4	y_1	y_2	y_3	y_4
A X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_0	a_3	a_3	a_0	a_4	a_5	a_3	a_4
x_2	a_4	a_6	a_4	a_6	a_0	a_2	a_6	a_6
x_3	a_5	a_1	a_1	a_5	a_5	a_7	a_7	a_5

Варіант № 5								
У	y_1	y_2	y_3	y_4	y_5	y_2	y_3	y_4
A X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_3	a_4	a_1	a_7	a_4	a_6	a_1	a_3
x_2	a_6	a_6	a_6	a_3	a_0	a_2	a_6	a_7
x_3	a_1	a_5	a_7	a_1	a_5	a_7	a_7	a_1

Варіант № 6								
У	y_1	y_3	y_2	y_4	y_5	y_2	y_3	y_5
A X	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_6	a_2	a_7	a_4	a_2	a_6	a_3
x_2	a_3	a_0	a_5	a_3	a_0	a_5	a_0	a_7
x_3	a_6	a_2	a_3	a_1	a_5	a_3	a_5	a_1

Варіант №7				
Y	y_2	y_3	y_1	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_4	a_3	a_2	a_1
x_2	a_1	a_4	a_3	a_4
x_3	a_2	a_1	a_1	a_3

Варіант №8				
Y	y_1	y_2	y_3	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_3	a_1	a_2	a_1
x_2	a_2	a_4	a_4	a_3
x_3	a_3	a_3	a_1	a_4

Варіант №9				
Y	y_2	y_3	y_1	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_1	a_3	a_2	a_2
x_2	a_3	a_4	a_3	a_4
x_3	a_2	a_4	a_1	a_1

Варіант №10				
Y	y_1	y_2	y_3	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_3	a_1	a_2	a_4
x_2	a_2	a_2	a_4	a_2
x_3	a_4	a_3	a_1	a_1

Завдання 2.

Розробити структурну схему автомата Мілі та записати в загальному вигляді систему канонічних рівнянь для даного автомата.

Варіант №1						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_1	a_2/y_2	a_4/y_4	a_5/y_2	a_5/y_1	a_6/y_2
x_2	a_5/y_2	a_5/y_1	a_3/y_2	a_4/y_3	a_1/y_2	a_1/y_1
x_3	a_4/y_3	a_3/y_4	a_1/y_3	a_2/y_1	a_4/y_3	a_3/y_4

Варіант №2						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_2	a_2/y_3	a_4/y_2	a_3/y_4	a_4/y_2	a_4/y_4
x_2	a_2/y_3	a_3/y_4	a_5/y_3	a_2/y_2	a_3/y_3	a_3/y_1
x_3	a_5/y_4	a_4/y_1	a_1/y_1	a_5/y_4	a_1/y_1	a_1/y_2

Варіант №3						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_4/y_1	a_1/y_2	a_5/y_4	a_5/y_1	a_4/y_1	a_6/y_2
x_2	a_1/y_2	a_5/y_4	a_2/y_1	a_4/y_3	a_3/y_2	a_1/y_1
x_3	a_3/y_3	a_3/y_3	a_1/y_3	a_3/y_2	a_1/y_3	a_3/y_4

Варіант №4						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_2	a_1/y_2	a_1/y_2	a_5/y_3	a_3/y_4	a_2/y_4
x_2	a_4/y_1	a_2/y_3	a_2/y_3	a_3/y_2	a_2/y_1	a_3/y_1
x_3	a_5/y_3	a_5/y_1	a_6/y_1	a_2/y_4	a_1/y_2	a_1/y_2

Варіант №5						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_2/y_1	a_1/y_2	a_4/y_4	a_5/y_2	a_6/y_1	a_1/y_2
x_2	a_3/y_2	a_5/y_3	a_2/y_1	a_4/y_3	a_3/y_2	a_1/y_3
x_3	a_1/y_3	a_3/y_4	a_1/y_3	a_2/y_1	a_1/y_3	a_3/y_4

Варіант №6						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_2	a_2/y_3	a_4/y_2	a_4/y_3	a_2/y_2	a_4/y_4
x_2	a_2/y_3	a_3/y_4	a_5/y_3	a_5/y_4	a_3/y_3	a_3/y_1
x_3	a_6/y_4	a_4/y_1	a_1/y_1	a_4/y_1	a_1/y_1	a_6/y_2

Варіант №7					
X \ A	a_1	a_2	a_3	a_4	a_5
x_1	a_3/y_2	a_4/y_2	a_1/y_3	a_2/y_4	a_2/y_1
x_2	a_2/y_1	a_1/y_1	a_3/y_2	a_1/y_2	a_3/y_1
x_3	a_5/y_4	a_5/y_4	a_3/y_4	a_4/y_1	a_5/y_4

Варіант 8					
X \ A	a_1	a_2	a_3	a_4	a_5
x_1	a_1/y_2	a_2/y_3	a_4/y_2	a_3/y_3	a_3/y_4
x_2	a_5/y_3	a_4/y_4	a_1/y_3	a_1/y_1	a_2/y_1
x_3	a_1/y_2	a_1/y_1	a_3/y_2	a_2/y_1	a_5/y_2

Варіант №9					
X \ A	a_1	a_2	a_3	a_4	a_5
x_1	a_2/y_1	a_3/y_1	a_4/y_2	a_1/y_1	a_3/y_3
x_2	a_1/y_3	a_5/y_2	a_5/y_3	a_4/y_2	a_1/y_1
x_3	a_4/y_2	a_1/y_3	a_2/y_3	a_5/y_4	a_3/y_2

Варіант №10					
X \ A	a_1	a_2	a_3	a_4	a_5
x_1	a_2/y_3	a_4/y_3	a_2/y_2	a_4/y_1	a_1/y_2
x_2	a_1/y_1	a_2/y_1	a_3/y_4	a_2/y_3	a_3/y_2
x_3	a_5/y_3	a_3/y_1	a_1/y_1	a_2/y_4	a_4/y_2

4.4 Контрольні запитання

1. Як виглядає в загальному вигляді структурна схема автомата з пам'яттю?
2. Чим структурний вхідний і вихідний алфавіти й алфавіт станів відрізняються від аналогічних алфавітів абстрактного автомата?
3. Сформулюйте теорему В.М.Глушкова про структурну повноту елементарних автоматів.
4. Дайте визначення елементарного автомата Мура з нетривіальною пам'яттю.
5. Яка система переходів є повною?
6. Яка система виходів є повною?
7. Наведіть приклади завдання автоматів Мура, що задовольняють умовам повноти системи переходів і виходів.
8. До чого зводиться канонічний метод структурного синтезу довільних скінченних автоматів?
9. Як в загальному вигляді записується система канонічних рівнянь?

10. Як визначити число компонент двійкових векторів, якими можна закодувати елементи вхідного алфавіту, вихідного алфавіту, алфавіту станів структурного автомата?

20 ВИБІР І ОПИС ЕЛЕМЕНТІВ ПАМ’ЯТІ

20.1 Мета заняття

Вивчити способи завдання закону функціонування тригерів у вигляді словесного опису, скороченої таблиці переходів, повної таблиці переходів, матриці переходів, характеристичного рівняння і взаємозв'язок між ними.

Набути практичних навичок різноманітних способів описання елементів пам’яті на прикладі синтезу асинхронного RS-тригера.

20.2. Короткі основні відомості з теорії та вирішення типових завдань

20.2.1 Вибір елементів пам’яті автомата

Елемент пам'яті (тригер) може бути заданий одним з декількох способів: скороченою таблицею переходів, повною таблицею переходів, характеристичним рівнянням, матрицею переходів. Розглянемо останній спосіб.

Для кожного з 4-х можливих переходів елементарного автомата (00, 01, 10, 11) завжди знайдеться значення вхідного сигналу, рівне 0 або 1, яке викликає даний перехід. Якщо елементарний автомат має 2 або більше входів, то на деякі переходи значення вхідних сигналів, що діють на одному або іншому вході виявляються несуттєвими.

Опишемо закон функціонування елементарного автомата, що має m входів, матрицею переходів:

$$\begin{array}{c} \\ 00 \\ 01 \\ 10 \\ 11 \end{array}
\left\| \begin{array}{cccccc}
x_1 & x_2 & \dots & x_k & \dots & x_m \\
b_{00}^1 & b_{00}^2 & \dots & b_{00}^k & \dots & b_{00}^m \\
b_{01}^1 & b_{01}^2 & \dots & b_{01}^k & \dots & b_{01}^m \\
b_{10}^1 & b_{10}^2 & \dots & b_{10}^k & \dots & b_{10}^m \\
b_{11}^1 & b_{11}^2 & \dots & b_{11}^k & \dots & b_{11}^m
\end{array} \right\|$$

Кількість рядків матриці завжди дорівнює 4 (за кількістю можливих переходів), кількість стовпців дорівнює числу вхідних сигналів. Елемент матриці b_{is}^k являє собою значення вхідного сигналу x_k, під дією якого елементарний автомат перейде зі стану i в стан s. При цьому b_{is}^k завжди дорівнює 0 або 1, або невизначений, якщо він не впливає на даний перехід.

Матриця переходів елементарного автомата складається за таблицею переходів.

Приклад 1.

Розглянемо приклад побудови матриці переходів тригера.

Нехай тригер, в загальному випадку, заданий скороченою таблицею переходів (табл. 20.1.). Побудувати повну таблицю переходів тригера і матрицю переходів.

Таблиця 20.1.

t		(t+1)
Х	У	Q
0	0	0
0	1	Q(t)
1	0	$\overline{Q(t)}$
1	1	1

Таблиця 20.2.

t			t+1
Х	У	Q	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Таблиця 20.3.

Q(t)	Q(t+1)	Х	У
0	0	0	0
		0	1
0	1	1	0
		1	1
1	0	0	0
		1	0
1	1	0	1
		1	1

Повна таблиця переходів тригера, побудована за скороченою, представлена в таблиці 20.2. Із повної таблиці переходів запишемо комбінації вхідних сигналів, які відповідають всім можливим переходам (табл. 20.3.)

Таким чином:

1. Для переходу "0-0" Х = 0, У може бути 0 або 1
2. Для переходу "0-1" Х = 1, У може бути 0 або 1
3. Для переходу "1-0" Х може бути 0 або 1, а У = 0.
4. Для переходу "1-1" Х може бути 0 або 1, а У = 1.

Тоді матриця переходів тригера запишеться у вигляді:

0 - 0	0	b1
0 - 1	1	b2
1 - 0	b3	0
1 - 1	b4	1

де b1, b2, b3, b4 – довільні сигнали (0 або 1)

Як правило, значення двох різних коефіцієнтів bi і bs з одного рядка матриці є залежними один від одного і знаходження цієї залежності з ростом числа інформаційних входів ускладнюється. Встановлення точної взаємозалежності між вхідними змінними тригера є обов'язковою умовою, що забезпечує можливість максимального спрощення схем з пам'яттю. В основі методики лежить таблиця, в якій представлені можливі поєднання вхідних змінних та відповідні їм комбінації (табл. 20.4.).

Таблиця 20.4. Таблиця довизначення вхідних сигналів тригерів.

			Номер варіанта														
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Вхідні сигнали		X1	0	0	1	1	00	01	01	01	01	11	001	001	011	011	0011
		X2	0	1	0	1	01	00	01	10	11	01	010	011	001	101	0101
Описання взаємозалежності	I	X1	0	0	1	1	-	b1	b1	b1	b1	-	b1	b1	b1	b1	b1
		X2	0	1	0	1	-	0	b1	$\overline{b1}$	1	-	$\overline{b1}\cdot b2$	$b1\vee b2$	$b1\vee b2$	$\overline{b1}\vee b2$	b2
	II	X1	0	0	1	1	0	-	b1	$\overline{b1}$	-	1	$\overline{b1}\cdot b2$	$b1\vee b2$	$b1\vee b2$	$\overline{b1}\vee b2$	b2
		X2	0	1	0	1	b1	-	b1	b1	-	b1	b1	b1	b1	b1	b1

З урахуванням довизначення вхідних змінних матриці переходів деяких стандартних тригерів мають вигляд (таб. 20.5.)

Таблиця 20.5. Матриці переходів стандартних тригерів.

Триггер-D			Триггер-T			Триггер-RS				Триггер-JK			
Q(t)	Q(t+1)	D	Q(t)	Q(t+1)	T	Q(t)	Q(t+1)	R	S	Q(t)	Q(t+1)	K	J
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	0	1	0	1	0	1	0	1
1	0	0	1	0	1	1	0	1	0	1	0	1	0
1	1	1	1	1	0	1	1	0	0	1	1	0	0

При побудові таблиць переходів синхронного тригера (додатково до входів X і Y вводиться вхід синхронізації C), слід мати на увазі, що при C = 0 внутрішній стан тригера не змінюється незалежно від станів входів X і Y, тобто Q (t +1) = Q (t), а при C = 1 синхронний тригер функціонує як відповідний асинхронний. З урахуванням цього отримуємо скорочену (табл. 20.6) і повну (табл. 20.7) таблиці переходів синхронного тригера. Із повної таблиці переходів запишемо комбінації вхідних сигналів, які відповідають всім можливим переходам (табл. 5.8.).

Таблиця 20.6.

t			(t+1)
C	X	Y	Q
0	0	0	Q(t)
0	0	1	Q(t)
0	1	0	Q(t)
0	1	1	Q(t)
1	0	0	0
1	0	1	Q(t)
1	1	0	$\overline{Q(t)}$
1	1	1	1

Таблиця 20.7.

t				t+1
C	X	У	Q	Q
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Таблиця 20.8.

Q(t)	Q(t+1)	X	У
0	0	0	0
		0	1
0	1	1	0
		1	1
1	0	0	0
		1	0
1	1	0	1
		1	1

Матриця переходів синхронного тригера повторює матрицю переходів аналогічного асинхронного тригера.

Приклад 2.

Словесний опис роботи асинхронного RS-тригера звучить наступним чином:

RS-тригер - двухвходовий, який при подачі активного сигналу на S-вхід (S = 1) і неактивного сигналу на R-вхід (R = 0) встановлюється в одиничний стан (Q = 1). При подачі активного сигналу на R-вхід (R = 1) і неактивного сигналу на S-вхід (S = 0) тригер встановлюється в нульовий стан (Q = 0).

При подачі двох не активних сигналів на S і R входи (S = 0; R = 0) тригер зберігає попередній стан (Q (t +1) = Q (t)).

Одночасна подача двох активних сигналів на S і R входи (S = 1; R = 1) заборонена. Якщо ж така ситуація виникає, то тригер вважається байдужим (невизначеним), (Q (t +1) = X).

Потрібно:

1) Скласти скорочену таблицю переходів асинхронного RS-тригера.

2) Скласти повну таблицю переходів асинхронного RS-тригера.

3) Скласти матрицю переходів асинхронного RS-тригера.

4) Записати характеристичне рівняння асинхронного RS-тригера Q (t +1) = f (R (t), S (t), Q (t)) ..

5) Побудувати граф переходів асинхронного RS-тригера.

Рішення:

1) Скорочена таблиця переходів асинхронного RS-тригера представлена в табл. 20.9.

Таблиця 20.9.

Входи		Вихід	**Режим**
t		t+1	
R(t)	S(t)	Q(t+1)	
0	0	Q(t)	Зберігання
0	1	1	Установка в 1
1	0	0	Установка в 0
1	1	x	Невизначений стан

2) На основі скороченою таблиці переходів будуємо повну таблицю переходів асинхронного RS-тригера (Табл. 20.10.)

Таблиця 20.10.

R	S	Q(t)	Q(t+1)	$\bar{Q}$ (t+1)
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	x	x
1	1	1	x	x

3) Щоб скласти матрицю переходів RS-тригера, спочатку представимо повну таблицю переходів в дещо іншому вигляді (Табл. 20.11).

Таблиця 20.11.

Q(t)	Q(t+1)	R(t)	S(t)
0	0	0	0
		1	0
0	1	0	1
1	0	1	0
1	1	0	0
		0	1

З таблиці 20.11. випливає, що для переходу 0 → 0 S(t) = 0, а R(t) може дорівнювати як 0, так і 1, тобто бути невизначеним. Позначимо невизначений стан R(t) символом b1.

Перехід від 0→1 відбувається при однозначній комбінації сигналів R(t)=0, а S(t) = 1. Як і перехід 1→0 можливий при однозначній комбінації сигналів R(t) = 1, а S(t) = 0.

При переході 1→1 R(t) = 0, а S(t) може бути як 0, так і 1. Позначимо невизначений стан S(t) символом b2. І тоді остаточно матриця переходів асинхронного RS-тригера, у якій показані всі можливі переходи Q(t) → Q(t +1) і відповідні цим переходам значення вхідних сигналів R(t) і S(t) прийме вигляд, представлений у таблиці 20.12 .

Таблиця 20.12

Q(t)	Q(t+1)	R(t)	S(t)
0	0	b_1	0
0	1	0	1
1	0	1	0
1	1	0	b_2

Матриця переходів RS-тригера буває необхідна при синтезі інших типів тригерів, а також при синтезі автоматів з пам'яттю.

4) Для запису характеристичного рівняння асинхронного RS-тригера, скористаємося повною таблицею переходів RS-тригера (табл. 20.10), розглядаючи її як табличне завдання функції Q(t +1) = f (R(t), S(t), Q(t)).

Виходячи з таблиці переходів, можна зробити висновок, що дана функція є не повністю визначеною. Тому для її мінімізації скористаємося методом карт Карно для функції 3-х змінних (Рис. 20.1.).

SQ R	00	01	11	10
0		1	1	1
1			*	*

Рис. 20.1.

В результаті довизначення невизначених наборів «1», отримаємо наступний результат мінімізації:

$$Q(t+1) = \overline{R}Q \vee S \qquad (20.1.)$$

Це і є характеристичне рівняння асинхронного RS-тригера.

5) Побудуємо граф переходів асинхронного RS-тригера (Рис. 20.2.).

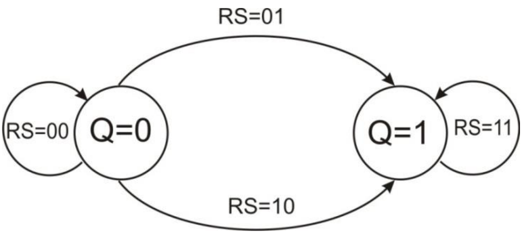


Рис. 20.2. Граф переходів асинхронного RS-тригера.

20.3 Індивідуальні завдання

Завдання 1

Асинхронний тригер з двома інформаційними входами X та У заданий скороченою таблицею переходів (табл. 20.13.) Побудувати повну таблицю переходів асинхронного тригера, повну таблицю переходів синхронного тригера, матрицю переходів асинхронного тригера, записати характеристичне рівняння даного тригера.

Таблиця 20.13.

Входи		Номер варіанта													
X	У	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	Q(t)	1	0	0	1	Q(t)	0	0	Q(t)	Q(t)	1	1	0	Q(t)
0	1	0	0	1	1	Q(t)	1	Q(t)	$\overline{Q}(t)$	1	1	0	1	Q(t)	1
1	0	Q(t)	Q(t)	0	Q(t)	0	0	1	0	0	1	1	Q(t)	0	0
1	1	1	Q(t)	Q(t)	1	0	1	Q(t)	1	0	Q(t)	Q(t)	0	1	0

Продовження таблиці 20.13.

Входи		Номер варіанта													
X	У	15	16	17	18	19	20	21	22	23	24	25	26	27	28
0	0	1	1	0	1	1	1	$\overline{Q}(t)$	Q(t)	Q(t)	1	0	0	Q(t)	0
0	1	Q(t)	Q(t)	1	1	0	Q(t)	1	0	Q(t)	$\overline{Q}(t)$	Q(t)	0	0	0
1	0	$\overline{Q}(t)$	0	Q(t)	0	Q(t)	1	Q(t)	1	1	Q(t)	1	Q(t)	$\overline{Q}(t)$	0
1	1	0	Q(t)	Q(t)	Q(t)	0	0	0	1	0	1	1	1	1	1

20.4 Контрольні запитання

1) Які елементи пам'яті прийнято називати тригерами?

2) Які тригери називаються бістабільними?

3) Як побудувати таблицю повну переходів тригера?

4) Як побудувати матрицю переходів тригера?

5) У чому полягає відмінність асинхронного тригера від синхронного?

6) Як можуть бути задані правила функціонування тригерів?

7) Скільки логічних станів на виході може мати бістабільний тригер і що вони означають?

8) Яка комбінація вхідних сигналів називається забороненою?

9)Скільки теоретично можна побудувати тригерів з X інформаційними входами?

21. ЗАБЕЗПЕЧЕННЯ СТІЙКОСТІ ФУНКЦІОНУВАННЯ ЦИФРОВИХ АВТОМАТІВ

21.1 Мета заняття

Провести аналіз причин появи гонок в автоматах, засвоєння апаратних та алгоритмічних способів боротьби з явищем гонок в автоматах, рішення задач протигоночного кодування станів автомата.

21.2 Короткі основні відомості з теорії та вирішення типових завдань

21.2.1.Гонки в автоматі

Завдання кодування станів є однією з основних задач канонічного методу структурного синтезу автоматів. Кодування полягає у встановленні взаємно-однозначної відповідності між множиною A = {a_1,. . . , a_M} станів автомата і множиною R-компонентних векторів {K_1,. . . , K_M}, K_m = (e_{m1}, ..., e_{mR}), де e_{mr} - стан r-го елементу пам'яті, r = 1,. . . , R. Якщо e_{mr} {0, 1}, тобто алфавіт станів елементів пам'яті двійковий, R $\geq$] $\log_2 M$ [. Далі заради простоти обмежимося використанням в якості елементів пам'яті RS-тригерів, які будемо позначати T_1,. . . , T_R.

Перехід автомата з одного стану в інший здійснюється за рахунок зміни станів елементів пам'яті. Так, якщо автомат переходить зі стану a_m з кодом 0101 в стан a_s з кодом 1001, то це означає, що тригер T_1 переходить зі стану 0 в стан 1, тригер T_2 - зі стану 1 у стан 0, а стану тригерів T_3 і T_4 НЕ змінюються.

При функціонуванні автомата можуть з'явитися так звані змагання. Явище змагань виникає внаслідок того, що елементи пам'яті мають різні, хоча і досить близькі, часи спрацьовування. Крім того, різні також затримки сигналів збудження, що надходять на вхідні канали елементарних автоматів по логічним ланцюгах неоднакової довжини. Якщо при переході автомата з одного стану в інший повинні змінити свої стани відразу декілька запам'ятовуючих елементів, то між ними починаються змагання. Той елемент, який виграє ці змагання, тобто змінює свій стан раніше, ніж інші елементи, може через ланцюг зворотного зв'язку змінити сигнали на входах деяких запам'ятовуючих елементів до того, як інші елементи, які приймають участь у змаганнях, змінять свої стани. Це може призвести до переходу автомата в стан, не передбачений його графом. Тому в процесі переходу зі стану a_m в стан a_s під дією вхідного сигналу z_f (рис. 21.1, а) автомат може опинитися в деякому проміжному стані a_k або $_a$l в залежності від того, який елемент пам'яті виграє змагання. Якщо потім при тому ж вхідному сигналі автомат з a_k і a_l перейде в стан a_s, то такі змагання є допустимими, або некритичними. Якщо ж в цьому автоматі є перехід, наприклад, з a_k в $a_j \neq a_s$ під дією того ж сигналу z_f (рис. 21.1, б), то автомат може перейти в a_j, а не в a_s і правильність його роботи тим самим буде порушена. Такі змагання називаються критичними змаганнями або гонками. При кодуванні станів гонки повинні бути усунені. Кодування з усуненням гонок називається протигоночним.

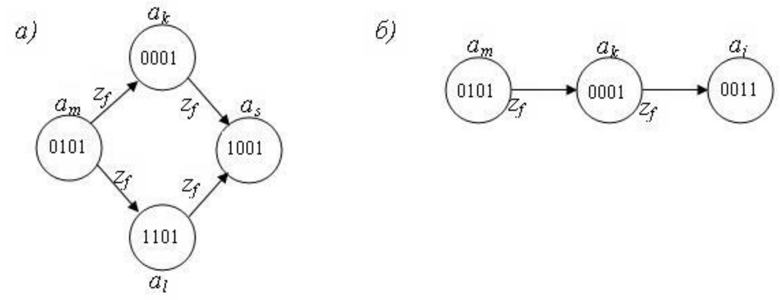


Рис. 21.1. Змагання між елементами пам'яті а) некритичні, б) критичні.

Один із способів ліквідації гонок полягає в тактуванні вхідних сигналів автомата імпульсами певної тривалості. Передбачається, що крім вхідних каналів $x_1, \ldots, x_L$ є ще один канал C від генератора синхроімпульсів (ГСІ), по якому надходить сигнал C = 1 в момент приходу імпульсу і C = 0 при його відсутності. У зв'язку з цим вхідним сигналом на переході (a_m, a_s) буде не z_f, а Cz_f. Тоді, якщо тривалість імпульсу t_C менше самого короткого шляху проходження тактованого сигналу зворотного зв'язку по комбінаційній схемі, то до моменту переходу в проміжний стан a_k (рис. 21.1, б) сигнал C дорівнює нулю і, отже, $Cz_f = 0$, що виключає гонки.

Інший спосіб ліквідації гонок полягає у введенні подвійної пам'яті (рис. 21.2). У цьому випадку кожен елемент пам'яті дублюється, причому перепис з нижнього елемента пам'яті в верхній відбувається в момент відсутності тактуючого імпульсу (C=0). Сигнали зворотного зв'язку для одержання функції збудження і функцій виходів автомата знімаються з верхнього ряду тригерів. Таким чином, змагання можуть виникнути тільки між нижніми тригерами, сигнали зворотного зв'язку не зможуть змінитися до тих пір, поки C не стане рівним нулю. Але тоді вхідний сигнал Cz_f, також дорівнює нулю, а тому гонок бути не може.

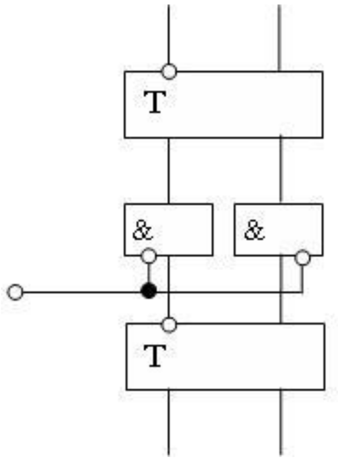


Рис. 21.2. Подвійна пам'ять

21.3 Контрольні запитання

1) Що є причиною гонок в автоматах?
2) В чому проявляється ефект гонок в автоматах?
3) За яких умов когут виникнути гонки?
4) В чому полягає суть синхронізації роботи автомата?
5) Які недоліки має метод синхронізації роботи автомата?
6) Як переключаються тригери основної та допоміжної ступенів автомата?
7) Які недоліки має використання двоступеневих тригерів?
8) Чому використання двоступеневої пам'яті виключає ефект гонок?
9) У чому полягає ідея протигоночного кодування?
10) Які недоліки має метод протигоночного кодування?

22. КОДУВАННЯ ВХІДНОГО І ВИХІДНОГО АЛФАВІТІВ СТРУКТУРНОГО АВТОМАТА.

22.1 Мета заняття

Вивчення основних алгоритмів кодування вхідного і вихідного алфавітів автомата двійковими кодами.

22.2 Короткі основні відомості з теорії та вирішення типових завдань

На структурному рівні кожна буква вхідного алфавіту $x \in X$, кожна буква вихідного алфавіту $y \in У$ і кожна буква алфавіту станів $a \in A$ кодується двійковими векторами (двійковими наборами), число компонент яких визначається таким чином:

$Квх=int(\log_2|X|)$;

$Квих=int(\log_2|У|)$;

$Кстан=int(\log_2|A|)$;

де int – найближче більше ціле число,
| X |, | У |, | A | - потужність алфавіту вхідного, вихідного і станів, відповідно. Потужністю алфавіту називається кількість різних символів входять в цей алфавіт.

Процес заміни буква алфавіту (X, У, A) абстрактного автомата двійковими векторами носить назву кодування і описується таблицями кодування. Таблиця кодування має наступний вигляд: у лівій частині перераховуються всі символи абстрактного алфавіту, а в правій - відповідні їм двійкові вектори.

Приклад.

Абстрактний автомат Мілі заданий таблицею переходів і виходів (табл. 22.1.). Виконати кодування алфавітів автомата.

Рішення.

Таблиця 22.1.

A \ X	x_1	x_2
a_1	a_2/y_1	a_3/y_3
a_2	a_1/y_2	a_2/y_1
a_3	a_2/y_2	a_1/y_1

Випишемо алфавіти автомата та визначимо довжини векторів кодів алфавітів:

$X = \{x_1, x_2\}$; Квх=int($\log_2$|2|)= 1,

$У = \{y_1, y_2, y_3\}$; Квих=int($\log_2$|3|)= 2,

$A = \{a_1, a_2, a_3\}$; Кстан=int($\log_2$|3|)= 2.

Запишемо таблиці кодування:

Таблиця 22.2.

X	
x_1	0
x_2	1

Таблиця 22.3.

У	
y_1	00
y_2	01
y_3	10

Таблиця 22.4.

A	
a_1	00
a_2	01
a_3	10

Результуюча таблиця - структурна таблиця переходів і виходів автомата (табл. 22.5.) Отриманням структурної таблиці переходів - виходів автомата закінчується етап кодування .

Таблиця 22.5.

A \ X	0	1
00	01/00	10 /10
01	00 /01	01/00
10	01 / 01	00 / 00

У загальному випадку, кожній кодованій букві може бути присвоєний довільний двійковий вектор, але обов'язково дві різні букви одного алфавіту повинні кодуватися різними векторами. Коди, які відповідають символам різних алфавітів можуть збігатися, в розглянутому прикладі - коди вихідних сигналів і станів.

При кодуванні вихідних сигналів використовується ваговий метод (метод вагових коефіцієнтів).

Алгоритм вагового методу кодування:

1. Кожному вихідному сигналу y_i ставиться у відповідність ціле число P_i, яке дорівнює числу появ сигналу y_i в таблиці виходів автомата.

2. Числа P_i сортуються за зменшенням.

3. Вихідний сигнал y_i з найбільшою вагою (P_i max) кодуються кодом, що містить всі 0 (00 ... 00).

4. Наступні L вихідних сигналів (де L-число розрядів у двійковому векторі вихідного сигналу) за списком зменшення ваги (див. п. 2) кодуються кодами, що містять одну 1 (00 ... 01, 00 ... 10, ..., 10 ... 00).

5. Для кодування наступних за списком зменшення вихідних сигналів використовуються всі коди, що містять дві одиниці, потім три одиниці і т.д., поки не будуть закодовані всі вихідні сигнали.

В результаті отримуємо таке кодування, при якому чим частіше зустрічається вихідний сигнал у таблиці виходів, тим менше одиниць міститься в його коді.

Приклад.

Автомат задано таблицею виходів (табл. 22.6). Виконати кодування вихідного алфавіту з використанням вагового методу. Кодування вхідного алфавіту виконати довільним методом.

Таблиця 22.6.

У	y_1	y_2	y_3	y_4	y_1	y_2	y_2
A / X	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_3	a_6	a_4	a_7	a_5	a_4
x_2	a_5	a_1	a_3	a_6	a_1	a_4	a_2
x_3	a_7	a_4	a_5	a_7	a_3	a_2	a_1

Рішення.

Кодування вихідного алфавіту.

Визначимо довжину вектора коду вихідного алфавіту.

Вихідні сигнали нашого автомата Мура Y = {y1, y2, y3, y4}, отже,

Квих = int (log (4)) = 2. Два розряди буде достатньо, щоб закодувати вихідні сигнали.

Кожному вихідному сигналу поставимо у відповідність ваговий коефіцієнт P. (таблиця 22.7.)

Таблиця 22.7 - вагові коефіцієнти вихідних сигналів.

У	P
y_1	2
y_2	3
y_3	1
y_4	1

Вагові коефіцієнти сортуються по зменшенню (таблиця 22.8.).

Таблиця 22.8

У	Р
y_2	3
y_1	2
y_3	1
y_4	1

Таблиця 22.9 – кодування вихідних сигналів.

У	w_1	w_2
y_2	0	0
y_1	0	1
y_3	1	0
y_4	1	1

Кодування вхідного алфавіту.

Визначимо довжину вектора коду вхідного алфавіту.

Вихідні сигнали нашого автомата Мура X = {x1, x2, x3}, отже,

Квх = int (log (3)) = 2. Два розряди буде достатньо, щоб закодувати вхідні сигнали. Вхідний алфавіт кодуємо довільним методом.

Таблиця 22.10 – кодування вхідних сигналів.

X	z_1	z_2
x_1	0	0
x_2	0	1
x_3	1	0

22.3 Індивідуальні завдання

Завдання 1.

Виконати кодування алфавітів автомата Мура:

а) вхідного алфавіту, використавши довільний метод;

б) вихідного алфавіту, використавши ваговий метод.

Варіант №1							
У	y_1	y_2	y_2	y_3	y_4	y_3	y_5
A / X	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_4	a_2	a_3	a_4	a_2	a_5
x_2	a_2	a_3	a_3	a_2	a_5	a_7	a_2
x_3	a_3	a_2	a_1	a_1	a_3	a_4	a_3
x_4	a_5	a_6	a_7	a_4	a_1	a_5	a_1
x_5	a_7	a_1	a_5	a_7	a_6	a_3	a_4

Варіант №2							
У	y_1	y_3	y_2	y_4	y_4	y_3	y_5
A / X	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_2	a_1	a_2	a_1	a_7	a_7	a_3
x_2	a_3	a_2	a_4	a_2	a_2	a_5	a_6
x_3	a_4	a_4	a_1	a_3	a_6	a_2	a_4
x_4	a_6	a_3	a_7	a_5	a_5	a_6	a_7
x_5	a_7	a_7	a_3	a_6	a_3	a_4	a_2

Варіант №3							
У	y_1	y_2	y_3	y_3	y_4	y_5	y_5
A / X	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_4	a_2	a_3	a_4	a_2	a_5
x_2	a_2	a_3	a_3	a_2	a_5	a_7	a_2
x_3	a_3	a_2	a_1	a_1	a_3	a_4	a_3
x_4	a_5	a_6	a_7	a_4	a_1	a_5	a_1
x_5	a_7	a_1	a_5	a_7	a_6	a_3	a_4

Варіант №4							
У	y_1	y_3	y_2	y_4	y_5	y_6	y_5
A / X	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_2	a_1	a_2	a_1	a_7	a_7	a_3
x_2	a_3	a_2	a_4	a_2	a_2	a_5	a_6
x_3	a_4	a_4	a_1	a_3	a_6	a_2	a_4
x_4	a_6	a_3	a_7	a_5	a_5	a_6	a_7
x_5	a_7	a_7	a_3	a_6	a_3	a_4	a_2

Варіант №5							
У	y_1	y_2	y_1	y_3	y_4	y_1	y_5
A / X	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_4	a_2	a_3	a_7	a_7	a_3
x_2	a_2	a_3	a_3	a_2	a_2	a_5	a_6
x_3	a_3	a_2	a_1	a_1	a_6	a_2	a_4
x_4	a_5	a_6	a_7	a_4	a_5	a_6	a_7
x_5	a_7	a_1	a_5	a_7	a_3	a_4	a_2

Варіант №6							
У	y_1	y_1	y_2	y_4	y_4	y_5	y_6
A / X	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_2	a_1	a_2	a_1	a_4	a_2	a_5
x_2	a_3	a_2	a_4	a_2	a_5	a_7	a_2
x_3	a_4	a_4	a_1	a_3	a_3	a_4	a_3
x_4	a_6	a_3	a_7	a_5	a_1	a_5	a_1
x_5	a_7	a_7	a_3	a_6	a_6	a_3	a_4

Варіант №7							
У	y_1	y_5	y_1	y_3	y_4	y_6	y_5
A / X	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_4	a_2	a_3	a_7	a_7	a_3
x_2	a_2	a_3	a_3	a_2	a_2	a_5	a_6
x_3	a_3	a_2	a_1	a_1	a_6	a_2	a_4
x_4	a_5	a_6	a_7	a_4	a_5	a_6	a_7
x_5	a_7	a_1	a_5	a_7	a_3	a_4	a_2

Варіант №8							
У	y_1	y_2	y_3	y_4	y_5	y_6	y_7
A / X	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_2	a_1	a_2	a_1	a_4	a_2	a_5
x_2	a_3	a_2	a_4	a_2	a_5	a_7	a_2
x_3	a_4	a_4	a_1	a_3	a_3	a_4	a_3
x_4	a_6	a_3	a_7	a_5	a_1	a_5	a_1
x_5	a_7	a_7	a_3	a_6	a_6	a_3	a_4

Варіант № 9								
У	y_1	y_2	y_3	y_4	y_5	y_2	y_3	y_4
X \ A	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_3	a_4	a_1	a_7	a_4	a_6	a_1	a_3
x_2	a_6	a_6	a_6	a_3	a_0	a_2	a_6	a_7
x_3	a_1	a_5	a_7	a_1	a_5	a_7	a_7	a_1

Варіант № 10								
У	y_1	y_3	y_2	y_4	y_5	y_2	y_3	y_5
X \ A	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_6	a_2	a_7	a_4	a_2	a_6	a_3
x_2	a_3	a_0	a_5	a_3	a_0	a_5	a_0	a_7
x_3	a_6	a_2	a_3	a_1	a_5	a_3	a_5	a_1

Варіант №11				
У	y_2	y_3	y_1	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_1	a_3	a_2	a_2
x_2	a_3	a_4	a_3	a_4
x_3	a_2	a_4	a_1	a_1

Варіант №12				
У	y_1	y_2	y_3	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_3	a_1	a_2	a_4
x_2	a_2	a_2	a_4	a_2
x_3	a_4	a_3	a_1	a_1

Завдання 2.

Виконати кодування алфавітів автомата Мілі:
а) вхідного алфавіту, використавши довільний метод;
б) вихідного алфавіту, використавши ваговий метод.

Варіант №1						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_1	a_2/y_2	a_4/y_4	a_5/y_2	a_5/y_1	a_6/y_2
x_2	a_5/y_2	a_5/y_1	a_3/y_2	a_4/y_3	a_1/y_2	a_1/y_1
x_3	a_4/y_3	a_3/y_4	a_1/y_3	a_2/y_1	a_4/y_3	a_3/y_4

Варіант №2						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_2	a_2/y_3	a_4/y_2	a_3/y_4	a_4/y_2	a_4/y_4
x_2	a_2/y_3	a_3/y_4	a_5/y_3	a_2/y_2	a_3/y_3	a_3/y_1
x_3	a_5/y_4	a_4/y_1	a_1/y_1	a_5/y_4	a_1/y_1	a_1/y_2

Варіант №3						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_4/y_1	a_1/y_2	a_5/y_4	a_5/y_1	a_4/y_1	a_6/y_2
x_2	a_1/y_2	a_5/y_4	a_2/y_1	a_4/y_3	a_3/y_2	a_1/y_1
x_3	a_3/y_3	a_3/y_3	a_1/y_3	a_3/y_2	a_1/y_3	a_3/y_4

Варіант №4						
X \ A	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_2	a_1/y_2	a_1/y_2	a_5/y_3	a_3/y_4	a_2/y_4
x_2	a_4/y_1	a_2/y_3	a_2/y_3	a_3/y_2	a_2/y_1	a_3/y_1
x_3	a_5/y_3	a_5/y_1	a_6/y_1	a_2/y_4	a_1/y_2	a_1/y_2

Варіант №5						
A X	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_2/y_1	a_1/y_2	a_4/y_4	a_5/y_2	a_6/y_1	a_1/y_2
x_2	a_3/y_2	a_5/y_3	a_2/y_1	a_4/y_3	a_3/y_2	a_1/y_3
x_3	a_1/y_3	a_3/y_4	a_1/y_3	a_2/y_1	a_1/y_3	a_3/y_4

Варіант №6						
A X	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_2	a_2/y_3	a_4/y_2	a_4/y_3	a_2/y_2	a_4/y_4
x_2	a_2/y_3	a_3/y_4	a_5/y_3	a_5/y_4	a_3/y_3	a_3/y_1
x_3	a_6/y_4	a_4/y_1	a_1/y_1	a_4/y_1	a_1/y_1	a_6/y_2

Варіант №7						
A X	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_4/y_1	a_1/y_2	a_5/y_4	a_5/y_1	a_4/y_1	a_6/y_2
x_2	a_1/y_2	a_5/y_4	a_2/y_1	a_4/y_3	a_3/y_2	a_1/y_1
x_3	a_3/y_3	a_3/y_3	a_1/y_3	a_3/y_2	a_1/y_3	a_3/y_4

Варіант №8						
A X	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_2	a_1/y_2	a_1/y_2	a_5/y_3	a_3/y_4	a_2/y_4
x_2	a_4/y_1	a_2/y_3	a_2/y_3	a_3/y_2	a_2/y_1	a_3/y_1
x_3	a_5/y_3	a_5/y_1	a_6/y_1	a_2/y_4	a_1/y_2	a_1/y_2

Варіант №9						
A X	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_2/y_1	a_1/y_2	a_4/y_4	a_5/y_2	a_6/y_1	a_1/y_2
x_2	a_3/y_2	a_5/y_3	a_2/y_1	a_4/y_3	a_3/y_2	a_1/y_3
x_3	a_1/y_3	a_3/y_4	a_1/y_3	a_2/y_1	a_1/y_3	a_3/y_4

Варіант №10						
A X	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_1/y_2	a_2/y_3	a_4/y_2	a_4/y_3	a_2/y_2	a_4/y_4
x_2	a_2/y_3	a_3/y_4	a_5/y_3	a_5/y_4	a_3/y_3	a_3/y_1
x_3	a_6/y_4	a_4/y_1	a_1/y_1	a_4/y_1	a_1/y_1	a_6/y_2

Варіант №11					
A X	a_1	a_2	a_3	a_4	a_5
x_1	a_2/y_1	a_3/y_1	a_4/y_2	a_1/y_1	a_3/y_3
x_2	a_1/y_3	a_5/y_2	a_5/y_3	a_4/y_2	a_1/y_1
x_3	a_4/y_2	a_1/y_3	a_2/y_3	a_5/y_4	a_3/y_2

Варіант №12					
A X	a_1	a_2	a_3	a_4	a_5
x_1	a_2/y_3	a_4/y_3	a_2/y_2	a_4/y_1	a_1/y_2
x_2	a_1/y_1	a_2/y_1	a_3/y_4	a_2/y_3	a_3/y_2
x_3	a_5/y_3	a_3/y_1	a_1/y_1	a_2/y_4	a_4/y_2

22.4 Контрольні запитання

1) Що таке потужність алфавіту?
2) Як визначити число компонент вектора вхідного алфавіту?
3) Як визначити число компонент вектора вихідного алфавіту?
4) Сформулюйте алгоритм вагового методу кодування вихідних сигналів.

23. ЕВРИСТИЧНИЙ МЕТОД КОДУВАННЯ ВНУТРІШНІХ СТАНІВ АВТОМАТА

23.1 Мета заняття

Вивчити і придбати навички кодування алфавіту внутрішніх станів автомата з використанням евристичного методу. Навчитись оцінювати ефективність кодування станів автомата.

23.2 Короткі основні відомості з теорії та вирішення типових завдань

Кодування внутрішніх станів двійковими символами робить істотний вплив на складність комбінаційної частини схеми автомата.

Д.А. Морозом запропонований евристичний алгоритм кодування внутрішніх станів автоматів, заснований на мінімізації сумарного числа змін станів елементів пам'яті автомата на всіх переходах автомата.

При такому критерії зменшується складність схем, що реалізують диз'юнкції на входах елементів пам'яті, тобто мінімізується комбінаційна схема автомата. Для оцінки кодування вводиться коефіцієнт ефективності кодування:
$K_{еф}$= W/P; де P – загальна кількість переходів автомата,

W - вагова функція : $W=\sum t_{ij}$,

де t_{ij} - відстань по Хеммінгу між кодами станів a_i та a_j, що дорівнює числу елементів пам'яті, що змінюють свій стан на даному переході.

Відзначимо, що при визначенні вагової функції підсумовування проводиться по всіх переходах автомата. Коефіцієнт ефективності дозволяє оцінити складність комбінаційної схеми автомата: чим менше його значення, тим менше складність комбінаційної схеми, і оптимальний варіант – $K_{еф}$=1.

Стани автомата $A=\{a_1,a_2\ldots,a_r\}$ кодуються векторами довжини $K_{стан}$=int(log(r)), де r потужність множини A (кількість станів). Відзначимо, що довжина вектора станів визначає і кількість елементів пам'яті (тригерів) даного автомата.

Алгоритм складається з наступних кроків:

1. Побудувати матрицю М, складену з усіх пар номерів (a_r, b_r) переходів автомата.

2. Переставимо рядки в матриці таким чином, щоб у кожному наступному рядку містився хоча б один елемент із попередніх рядків.

3. Закодуємо стан з першого рядка матриці М наступним чином: Ka_1=00...00, Kb_1=00...01.

4. Викреслимо з матриці М перший рядок із закодованими станами. Отримаємо матрицю М'

5. В силу п. 3 в першому рядку матриці М' закодовано один елемент. Виберемо з першого рядка матриці М' не закодованими елемент і позначимо його γ.

6. Побудуємо матрицю Мγ, вибравши з матриці М' рядки, що містять γ. Нехай $B\gamma=\{\gamma_1, \gamma_2, \dots, \gamma_f, \dots, \gamma_F\}$ - множина елементів з матриці Мγ, які вже закодовані. Їх коди позначимо через $K\gamma_1, K\gamma_2, \dots, K\gamma_f, \dots, K\gamma_F$ відповідно.

7. Для кожного $K\gamma_f$ $(f=1, 2, \dots, F)$ знайдемо $C^1_{\gamma f}$ - множину кодів, віддалених від $K\gamma_f$ на відстань Хеммінга, рівну 1 і ще не зайнятих для кодування станів автомата. Побудуємо множину $D^1_\gamma = \bigcup_{f=1}^{F} C^1_{\gamma f}$. Якщо $D^1_\gamma = 0$, то будуємо нову множину $D^2_\gamma = \bigcup_{f=1}^{F} C^2_{\gamma f}$, де $C^2_{\gamma f}$ множина кодів, у яких кодова відстань з кодом $K\gamma_f$ дорівнює 2. Якщо і $D^2_\gamma = 0$, будуємо D^3_γ і т.д., поки не знайдемо $D^n_\gamma \neq 0$.

Нехай $D^n_\gamma = \{K_{\gamma 1}, \dots, K_{\gamma g}, \dots, K_{\gamma G}\}$.

8. Для кожного $K_{\gamma G}$ знаходимо $w_{gf}=|K_{\gamma g}- K_{\gamma f}|^2$ - відстань Хеммінга між $K_{\gamma g}$ і всіма використовуваними кодами $K_{\gamma f}$ $(f=1, 2, \dots, F)$.

9. Знаходимо $w_g = \sum_{f=1}^{F} w_{gf}$, $(g=1, \dots, G)$.

10. Із D^n_γ вибираємо K_γ,, у якого $w_g = w_{g\,min}$.. Елемент γ кодуємо кодом K_γ.

11. З матриці М' викреслюємо рядки, в яких обидва елементи закодовані, в результаті чого отримуємо нову матрицю, яку також позначаємо М'. Якщо в матриці М' не залишилося ні одного рядка, переходимо до п. 12, інакше до п. 5.

12. Обчислюємо функцію $w=\sum t_{ms}$, где $t_{ms}|K_m-K_s|^2$.

13. Кінець.

В рамках даного практичного заняття розглянемо конкретний приклад.

Приклад 1.

Абстрактний автомат задано таблицею переходів (Таблиця 23.1). Виконати кодування алфавіту станів даного автомата, використовуючи евристичний метод кодування.

Таблиця 23.1.

X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_3	a_6	a_4	a_7	a_5	a_4
x_2	a_5	a_1	a_3	a_6	a_1	a_4	a_2
x_3	a_7	a_4	a_5	a_7	a_3	a_2	a_1

Рішення.

Стани автомата $A=\{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$ кодуються векторами довжини $K_{стан}=int(log(7))=3$.

Позначимо двійкові розряди вектора стану , як α_1, α_2, α_3. Довжина вектора станів визначає кількість елементів пам'яті автомата.

Для початку кодування побудуємо матрицю переходів автомата М та переставимо рядки в матриці таким чином, щоб у кожному наступному рядку містився хоча б один елемент із попередніх рядків. Елементи в першій строчці матриці М кодуємо: K1 = 000 і K5 = 001. Далі слідуємо евристичному методу кодування.

В матриці М викреслимо перший рядок (1-5) та ті рядки, які виявилися повністю закодованими (1-1, 5-1). Отримаємо матрицю М'.

$$M = \begin{vmatrix} 1\text{-}5 \\ 1\text{-}1 \\ 1\text{-}7 \\ 2\text{-}1 \\ 2\text{-}3 \\ 2\text{-}4 \\ 3\text{-}6 \\ 3\text{-}3 \\ 3\text{-}5 \\ 4\text{-}4 \\ 4\text{-}6 \\ 4\text{-}7 \\ 5\text{-}7 \\ 5\text{-}1 \\ 5\text{-}3 \\ 6\text{-}5 \\ 6\text{-}4 \\ 6\text{-}2 \\ 7\text{-}4 \\ 7\text{-}2 \\ 7\text{-}1 \end{vmatrix}
\qquad
M' = \begin{vmatrix} 1\text{-}7 \\ 2\text{-}1 \\ 2\text{-}3 \\ 2\text{-}4 \\ 3\text{-}6 \\ 3\text{-}3 \\ 3\text{-}5 \\ 4\text{-}4 \\ 4\text{-}6 \\ 4\text{-}7 \\ 5\text{-}7 \\ 5\text{-}3 \\ 6\text{-}5 \\ 6\text{-}4 \\ 6\text{-}2 \\ 7\text{-}4 \\ 7\text{-}2 \\ 7\text{-}1 \end{vmatrix}$$

В матриці М' в першому рядку не закодовано елемент 7. $\gamma=7$. Будуємо матрицю $M\gamma$ (M_7), вибираючи із матриці М' всі рядки, які містять γ (7).

$$M_7 = \begin{vmatrix} 1\text{-}7 \\ 4\text{-}7 \\ 5\text{-}7 \\ 7\text{-}4 \\ 7\text{-}2 \\ 7\text{-}1 \end{vmatrix}$$

B_7={1, 5}={000, 001} – множина елементів із матриці M_7, які вже закодовані.

Для кожного елемента цієї множини знайдемо множини:

$C^1_{\gamma f} = C^1_{71}$={010, 100};

$C^1_{\gamma f} = C^1_{75}$={011, 101};

D^1_7 = {010, 100, 011, 101};

Для кожного коду із множини D^1_7 знайдемо $w_{gf}= |\kappa_{\delta g}- \kappa_{\gamma f}|^2$ – міжкодову відстань по Хеммінгу

$$\begin{array}{ccccccc} & 1\text{-}7 & & 5\text{-}7 & & 7\text{-}1 & \\ W_{010}= & \begin{vmatrix} 000 \\ 010 \end{vmatrix} & + & \begin{vmatrix} 001 \\ 010 \end{vmatrix} & + & \begin{vmatrix} 010 \\ 000 \end{vmatrix} & = \quad 4 \end{array}$$

$$W_{100}= \begin{vmatrix} 000 \\ 100 \end{vmatrix} + \begin{vmatrix} 001 \\ 100 \end{vmatrix} + \begin{vmatrix} 100 \\ 000 \end{vmatrix} = \quad 4$$

$$W_{011}= \begin{vmatrix} 000 \\ 011 \end{vmatrix} + \begin{vmatrix} 001 \\ 011 \end{vmatrix} + \begin{vmatrix} 011 \\ 000 \end{vmatrix} = \quad 5$$

$$W_{101}= \left| \begin{matrix} 000 \\ 101 \end{matrix} \right| + \left| \begin{matrix} 001 \\ 101 \end{matrix} \right| + \left| \begin{matrix} 101 \\ 000 \end{matrix} \right| = 5$$

Вибираємо код, для якого w_g має мінімальне значення.

$К_7$=010.

Із матриці М' викреслюємо перший рядок (1-7) і ті рядки, які виявилися повністю закодованими (5-7, 7-1). В результаті отримаємо нову матрицю М', в першому рядку якої не закодовано елемент 2. γ=2. Будуємо матрицю Мγ ($М_2$), вибираючи із матриці М' всі рядки, які містять γ (2).

$$M'= \left| \begin{matrix} 2\text{-}1 \\ 2\text{-}3 \\ 2\text{-}4 \\ 3\text{-}6 \\ 3\text{-}3 \\ 3\text{-}5 \\ 4\text{-}4 \\ 4\text{-}6 \\ 4\text{-}7 \\ 5\text{-}3 \\ 6\text{-}5 \\ 6\text{-}4 \\ 6\text{-}2 \\ 7\text{-}4 \\ 7\text{-}2 \end{matrix} \right| \qquad M_2= \left| \begin{matrix} 2\text{-}1 \\ 2\text{-}3 \\ 2\text{-}4 \\ 6\text{-}2 \\ 7\text{-}2 \end{matrix} \right|$$

B_2= {1, 7} = {000, 010} – множина елементів із матриці $М_2$, які вже закодовані. Найдемо множини

$C^1_{\gamma f} = C^1_{21}$ ={100};

$C^1_{\gamma f} = C^1_{27}$ ={110, 011};

D^1_2 = {100, 110, 011};

Для кожного коду із множини D^1_2 найдемо

$w_{gf}= |к_{\delta g}- к_{\gamma f}|^2$ – міжкодову відстань по Хеммінгу.

$$\begin{array}{lcccc} & 2\text{-}1 & & 7\text{-}2 & \\ W_{100}= & \left| \begin{matrix} 100 \\ 000 \end{matrix} \right| & + & \left| \begin{matrix} 010 \\ 100 \end{matrix} \right| & = 3 \end{array}$$

$$W_{110}= \left| \begin{matrix} 110 \\ 000 \end{matrix} \right| + \left| \begin{matrix} 010 \\ 110 \end{matrix} \right| = 3$$

$$W_{011}= \left| \begin{matrix} 011 \\ 000 \end{matrix} \right| + \left| \begin{matrix} 010 \\ 011 \end{matrix} \right| = 3$$

Вибираємо $К_2$=100.

$К_7$=010.

Із матриці M' викреслюємо перший рядок (2-1) і ті рядки, які виявилися повністю закодованими (7-2). В результаті отримаємо нову матрицю M', в першому рядку якої не закодовано елемент 3. γ=3. Будуємо матрицю $M\gamma$ (M_3), вибираючи із матриці M' всі рядки, які містять γ (3).

$$M` = \begin{vmatrix} 2\text{-}3 \\ 2\text{-}4 \\ 3\text{-}6 \\ 3\text{-}3 \\ 3\text{-}5 \\ 4\text{-}4 \\ 4\text{-}6 \\ 4\text{-}7 \\ 5\text{-}3 \\ 6\text{-}5 \\ 6\text{-}4 \\ 6\text{-}2 \\ 7\text{-}4 \end{vmatrix} \qquad M_3 = \begin{vmatrix} 2\text{-}3 \\ 3\text{-}6 \\ 3\text{-}3 \\ 3\text{-}5 \\ 5\text{-}3 \end{vmatrix}$$

B_3 = {2, 5} = {100, 001} – множина елементів із матриці M_3, які вже закодовані. Найдемо множини

$C^1_{\gamma f} = C^1_{32}$ ={101, 110};

$C^1_{\gamma f} = C^1_{35}$ ={101, 011};

$D^1_3 = \{101, 110, 011\}$;

Для кожного коду із множини D^1_3 найдемо

$w_{gf} = |\kappa_{\delta g} - \kappa_{\gamma f}|^2$ – міжкодову відстань по Хеммінгу.

$$\begin{array}{lccccccccc} & 2\text{-}3 & & 3\text{-}3 & & 3\text{-}5 & & 5\text{-}3 & & \\ W_{101}= & \begin{vmatrix} 100 \\ 101 \end{vmatrix} & + & \begin{vmatrix} 101 \\ 101 \end{vmatrix} & + & \begin{vmatrix} 101 \\ 001 \end{vmatrix} & + & \begin{vmatrix} 001 \\ 101 \end{vmatrix} & = & 3 \end{array}$$

$$W_{110}= \begin{vmatrix} 100 \\ 110 \end{vmatrix} + \begin{vmatrix} 110 \\ 110 \end{vmatrix} + \begin{vmatrix} 110 \\ 001 \end{vmatrix} + \begin{vmatrix} 001 \\ 110 \end{vmatrix} = 7$$

$$W_{011}= \begin{vmatrix} 100 \\ 011 \end{vmatrix} + \begin{vmatrix} 011 \\ 011 \end{vmatrix} + \begin{vmatrix} 011 \\ 001 \end{vmatrix} + \begin{vmatrix} 001 \\ 011 \end{vmatrix} = 5$$

Вибираємо код з мінімальним значенням W.

K_3=101.

Із матриці M' викреслюємо перший рядок (2-3) і ті рядки, які виявилися повністю закодованими (3-3, 3-5, 5-3). В результаті отримаємо нову матрицю M', в першому рядку якої не закодовано елемент 4. γ=4. Будуємо матрицю $M\gamma$ (M_4), вибираючи із матриці M' всі рядки, які містять γ (4).

$$M` = \begin{vmatrix} 2\text{-}4 \\ 3\text{-}6 \\ 4\text{-}4 \\ 4\text{-}6 \\ 4\text{-}7 \\ 6\text{-}5 \\ 6\text{-}4 \\ 6\text{-}2 \\ 7\text{-}4 \end{vmatrix} \qquad M_4 = \begin{vmatrix} 2\text{-}4 \\ 4\text{-}4 \\ 4\text{-}6 \\ 4\text{-}7 \\ 6\text{-}4 \\ 7\text{-}4 \end{vmatrix}$$

B_4={2, 7}={100, 010} – множина елементів из матриці M_4, які вже закодовані. Найдемо множини

$C^1_{\gamma f} = C^1_{42}$={110};

$C^1_{\gamma f} = C^1_{47}$={011, 110};

D^1_4 = {011, 110};

Для кожного коду із множини D^1_4 найдемо

$w_{gf}=|\kappa_{\delta g}-\kappa_{\gamma f}|^2$ – міжкодову відстань по Хеммінгу.

$$\begin{array}{cccccccccc} & 2\text{-}4 & & 4\text{-}4 & & 4\text{-}7 & & 7\text{-}4 & & \\ W_{110}= & \begin{vmatrix} 100 \\ 110 \end{vmatrix} & + & \begin{vmatrix} 110 \\ 110 \end{vmatrix} & + & \begin{vmatrix} 110 \\ 010 \end{vmatrix} & + & \begin{vmatrix} 010 \\ 110 \end{vmatrix} & = & 3 \end{array}$$

$$W_{011} = \begin{vmatrix} 100 \\ 011 \end{vmatrix} + \begin{vmatrix} 011 \\ 011 \end{vmatrix} + \begin{vmatrix} 011 \\ 010 \end{vmatrix} + \begin{vmatrix} 010 \\ 011 \end{vmatrix} = 4$$

Вибираємо код з мінімальним значенням W.

K_4=110.

Із матриці M' викреслюємо перший рядок (2-4) і ті рядки, які виявилися повністю закодованими (4-4, 4-7, 7-4). В результаті отримаємо нову матрицю M', в першому рядку якої не закодовано елемент 6. γ=6. Будуємо матрицю Mγ (M_6), вибираючи із матриці M' всі рядки, які містять γ (6).

$$M` = \begin{vmatrix} 3\text{-}6 \\ 4\text{-}6 \\ 6\text{-}5 \\ 6\text{-}4 \\ 6\text{-}2 \end{vmatrix} \qquad M_6 = \begin{vmatrix} 3\text{-}6 \\ 4\text{-}6 \\ 6\text{-}5 \\ 6\text{-}4 \\ 6\text{-}2 \end{vmatrix}$$

B_6= {3, 4, 5, 2} = {101, 110, 001, 100} – множина елементів із матриці M_6, які вже закодовані. Найдемо множини

$C^1_{\gamma f} = C^1_{63}$={111};

$C^1_{\gamma f} = C^1_{64}$={111};

$C^1_{\gamma f} = C^1_{65}$={011};

$C^1_{\gamma f} = C^1_{62}$={Ø};

$D_6^1 = \{111, 011\}$;

Для кожного коду із множини D_6^1 найдемо

$w_{gf} = |к_{\delta g} - к_{\gamma f}|^2$ – міжкодову відстань по Хеммінгу.

3-6 4-6 6-5 6-4 6-2

$$W_{111} = \left|\begin{matrix}101\\111\end{matrix}\right| + \left|\begin{matrix}110\\111\end{matrix}\right| + \left|\begin{matrix}111\\001\end{matrix}\right| + \left|\begin{matrix}111\\110\end{matrix}\right| + \left|\begin{matrix}111\\100\end{matrix}\right| = 7$$

$$W_{011} = \left|\begin{matrix}101\\011\end{matrix}\right| + \left|\begin{matrix}110\\011\end{matrix}\right| + \left|\begin{matrix}011\\001\end{matrix}\right| + \left|\begin{matrix}011\\110\end{matrix}\right| + \left|\begin{matrix}011\\100\end{matrix}\right| = 10$$

Вибираємо код з мінімальним значенням W.

K_6=111.

Із матриці M' викреслюємо всі закодовані рядки. В матриці не залишилось незакодованих елементів.

Результат застосування евристичного методу кодування станів представлений у таблиці 24.2.

Таблиця 23.2.

A \ α	α_1	α_2	α_3
a_1	0	0	0
a_2	1	0	0
a_3	1	0	1
a_4	1	1	0
a_5	0	0	1
a_6	1	1	1
a_7	0	1	0

Обчислимо коефіцієнт ефективності кодування станів: K=W / P:

$$W = \left|\begin{matrix}000\\000\end{matrix}\right| + \left|\begin{matrix}000\\001\end{matrix}\right| + \left|\begin{matrix}000\\010\end{matrix}\right| + \left|\begin{matrix}100\\101\end{matrix}\right| + \left|\begin{matrix}100\\000\end{matrix}\right| + \left|\begin{matrix}100\\110\end{matrix}\right| + \left|\begin{matrix}101\\111\end{matrix}\right| +$$

$$+ \left|\begin{matrix}101\\101\end{matrix}\right| + \left|\begin{matrix}101\\001\end{matrix}\right| + \left|\begin{matrix}110\\110\end{matrix}\right| + \left|\begin{matrix}110\\111\end{matrix}\right| + \left|\begin{matrix}110\\010\end{matrix}\right| + \left|\begin{matrix}001\\010\end{matrix}\right| + \left|\begin{matrix}001\\000\end{matrix}\right| +$$

$$+ \left|\begin{matrix}001\\101\end{matrix}\right| + \left|\begin{matrix}111\\001\end{matrix}\right| + \left|\begin{matrix}111\\110\end{matrix}\right| + \left|\begin{matrix}111\\100\end{matrix}\right| + \left|\begin{matrix}010\\110\end{matrix}\right| + \left|\begin{matrix}010\\100\end{matrix}\right| + \left|\begin{matrix}010\\000\end{matrix}\right| = 22$$

W=22; P=21; K=22/21=1,048.

23.3 Індивідуальні завдання

Завдання 1.

Виконати кодування алфавіту станів автомата Мура, використавши евристичний алгоритм.

Варіант №1							
Y	y_1	y_2	y_2	y_3	y_4	y_3	y_5
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_4	a_2	a_3	a_4	a_2	a_5
x_2	a_2	a_3	a_3	a_2	a_5	a_7	a_2
x_3	a_3	a_2	a_1	a_1	a_3	a_4	a_3
x_4	a_5	a_6	a_7	a_4	a_1	a_5	a_1
x_5	a_7	a_1	a_5	a_7	a_6	a_3	a_4

Варіант №2							
Y	y_1	y_3	y_2	y_4	y_4	y_3	y_5
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_2	a_1	a_2	a_1	a_7	a_7	a_3
x_2	a_3	a_2	a_4	a_2	a_2	a_5	a_6
x_3	a_4	a_4	a_1	a_3	a_6	a_2	a_4
x_4	a_6	a_3	a_7	a_5	a_5	a_6	a_7
x_5	a_7	a_7	a_3	a_6	a_3	a_4	a_2

Варіант №3							
Y	y_1	y_2	y_3	y_3	y_4	y_5	y_5
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_4	a_2	a_3	a_4	a_2	a_5
x_2	a_2	a_3	a_3	a_2	a_5	a_7	a_2
x_3	a_3	a_2	a_1	a_1	a_3	a_4	a_3
x_4	a_5	a_6	a_7	a_4	a_1	a_5	a_1
x_5	a_7	a_1	a_5	a_7	a_6	a_3	a_4

Варіант №4							
Y	y_1	y_3	y_2	y_4	y_5	y_6	y_5
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_2	a_1	a_2	a_1	a_7	a_7	a_3
x_2	a_3	a_2	a_4	a_2	a_2	a_5	a_6
x_3	a_4	a_4	a_1	a_3	a_6	a_2	a_4
x_4	a_6	a_3	a_7	a_5	a_5	a_6	a_7
x_5	a_7	a_7	a_3	a_6	a_3	a_4	a_2

Варіант №5							
Y	y_1	y_2	y_1	y_3	y_4	y_1	y_5
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_4	a_2	a_3	a_7	a_7	a_3
x_2	a_2	a_3	a_3	a_2	a_2	a_5	a_6
x_3	a_3	a_2	a_1	a_1	a_6	a_2	a_4
x_4	a_5	a_6	a_7	a_4	a_5	a_6	a_7
x_5	a_7	a_1	a_5	a_7	a_3	a_4	a_2

Варіант №6							
Y	y_1	y_1	y_2	y_4	y_4	y_5	y_6
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_2	a_1	a_2	a_1	a_4	a_2	a_5
x_2	a_3	a_2	a_4	a_2	a_5	a_7	a_2
x_3	a_4	a_4	a_1	a_3	a_3	a_4	a_3
x_4	a_6	a_3	a_7	a_5	a_1	a_5	a_1
x_5	a_7	a_7	a_3	a_6	a_6	a_3	a_4

Варіант №7							
У	y_1	y_5	y_1	y_3	y_4	y_6	y_5
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_4	a_2	a_3	a_7	a_7	a_3
x_2	a_2	a_3	a_3	a_2	a_2	a_5	a_6
x_3	a_3	a_2	a_1	a_1	a_6	a_2	a_4
x_4	a_5	a_6	a_7	a_4	a_5	a_6	a_7
x_5	a_7	a_1	a_5	a_7	a_3	a_4	a_2

Варіант №8							
У	y_1	y_2	y_3	y_4	y_5	y_6	y_7
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_2	a_1	a_2	a_1	a_4	a_2	a_5
x_2	a_3	a_2	a_4	a_2	a_5	a_7	a_2
x_3	a_4	a_4	a_1	a_3	a_3	a_4	a_3
x_4	a_6	a_3	a_7	a_5	a_1	a_5	a_1
x_5	a_7	a_7	a_3	a_6	a_6	a_3	a_4

Варіант № 9								
У	y_1	y_2	y_3	y_4	y_5	y_2	y_3	y_4
X \ A	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_3	a_4	a_1	a_7	a_4	a_6	a_1	a_3
x_2	a_6	a_6	a_6	a_3	a_0	a_2	a_6	a_7
x_3	a_1	a_5	a_7	a_1	a_5	a_7	a_7	a_1

Варіант № 10								
У	y_1	y_3	y_2	y_4	y_5	y_2	y_3	y_5
X \ A	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_6	a_2	a_7	a_4	a_2	a_6	a_3
x_2	a_3	a_0	a_5	a_3	a_0	a_5	a_0	a_7
x_3	a_6	a_2	a_3	a_1	a_5	a_3	a_5	a_1

Варіант №11				
У	y_2	y_3	y_1	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_1	a_3	a_2	a_2
x_2	a_3	a_4	a_3	a_4
x_3	a_2	a_4	a_1	a_1

Варіант №12				
У	y_1	y_2	y_3	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_3	a_1	a_2	a_4
x_2	a_2	a_2	a_4	a_2
x_3	a_4	a_3	a_1	a_1

23.4 Контрольні запитання

1) На що має істотний вплив спосіб кодування внутрішніх станів автомата?
2) На чому заснований евристичний алгоритм кодування внутрішніх станів автомата?
3) Як визначити коефіцієнт ефективності кодування автомата?
4) Як визначається відстань Хеммінга між кодами станів?
5) Як визначається вагова функція?
6) Чому дорівнює оптимальне значення $К_{еф}$?
7) Як визначити потужність алфавіту станів?
8) Як визначити довжину двійкового вектора станів структурного автомата?
9) На що вказує довжина вектора станів структурного автомата?

24. ПОБУДОВА ТАБЛИЦЬ ПЕРЕХОДІВ І ВИХОДІВ СТРУКТУРНОГО АВТОМАТА, ТАБЛИЦЬ ФОРМУВАННЯ ФУНКЦІЙ ЗБУДЖЕННЯ ЕЛЕМЕНТІВ ПАМ’ЯТІ

24.1 Мета заняття

Засвоїти методику і набути практичних навичок побудови таблиць переходів і виходів структурного автомата, таблиць формування функцій збудження елементів пам'яті і функцій виходу, мінімізації функцій збудження елементів пам'яті і функцій виходу, складання системи канонічних рівнянь структурних автоматів Мілі і Мура.

24.2 Короткі основні відомості з теорії та вирішення типових завдань

Після того, як всі елементи алфавітів структурного автомата закодовані, складається зазначена таблиця переходів структурного автомата, в якій замість буквених символів алфавітів абстрактного автомата записуються їх двійкові коди. Розглянемо приклад складання таких таблиць для автомата Мура.

Приклад.

Нехай зазначена таблиця переходів мінімального абстрактного автомата Мура виглядає так: (див. табл. 24.1).

Таблиця 24.1

У	y_1	y_2	y_3	y_4	y_1	y_2	y_2
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_3	a_6	a_4	a_7	a_5	a_4
x_2	a_5	a_1	a_3	a_6	a_1	a_4	a_2
x_3	a_7	a_4	a_5	a_7	a_3	a_2	a_1

А таблиці кодування вхідного, вихідного алфавітів і алфавіту станів виглядають наступним чином (див. табл. 24.2, 24.3, 24.4).

Таблиця 24.2 – кодування вихідних сигналів.

У	w_1	w_2
y_1	0	1
y_2	0	0
y_3	1	0
y_4	1	1

Таблиця 24.3 – кодування вхідних сигналів.

X	z_1	z_2
x_1	0	0
x_2	0	1
x_3	1	0

Таблиця 24.4 – кодування алфавіту станів

A \ α	α_1	α_2	α_3
a_1	0	0	0
a_2	1	0	0
a_3	1	0	1
a_4	1	1	0
a_5	0	0	1
a_6	1	1	1
a_7	0	1	0

Тоді зазначена таблиця структурного автомата Мура буде виглядати так:

Таблиця 24.5 - зазначена таблиця структурного автомата Мура

У	Вихідні сигнали w_1 w_2						
	01	00	10	11	01	00	00
	Стани α_1 α_2 α_3						
X (z_1 z_2) \ A	000	100	101	110	001	111	010
00	000	101	111	110	010	001	110
01	001	000	101	111	000	110	100
10	010	110	001	010	101	100	000

Після кодування елементів алфавіту та вибору елементів пам'яті можна приступати до синтезу комбінаційних схем, що формують функції збудження елементів пам'яті (КС1) і функцій виходів структурного автомата (КС2).

Для цього спочатку необхідно побудувати таблиці формування булевих функцій збудження елементів пам'яті і функцій виходів структурного автомата. Вихідними даними для побудови зазначених таблиць є структурна таблиця переходів автомата і матриця переходів елемента пам'яті.

Зазначені таблиці можна поєднати в одній таблиці.

Розглянемо приклад побудови таких таблиць для автомата Мура, заданого зазначеною структурною таблицею переходів табл. 24.5.

Кількість функцій збудження елементів пам'яті автомата залежить від кількості розрядів вектора коду стану і від кількості інформаційних входів самого запам'ятовуючого елемента.

Припустимо, що в якості елементів пам'яті (їх всього 3 - T1, T2, T3) використовуються синхронні 2-х ступінчасті XY – тригери, які мають наступну матрицю переходів (див. табл. 24.6):

Таблиця 24.6

0 - 0	0	b1
0 - 1	1	b2
1 - 0	b3	0
1 - 1	b4	1

де b1, b2, b3, b4 – довільні сигнали (0 або 1)

Тоді суміщена таблиця формування функцій збудження елементів пам'яті і функцій виходів структурного автомата буде мати наступний формат (див. табл. 24.7).

Таблиця 24.7

A	Початковий стан елементів пам'яті (α_i^t)			Вхідний сигнал		Стан переходу елементів пам'яті (α_i^{t+1})			T1		T2		T3		Вихідний сигнал	
	α_1	α_2	α_3	Z_1	Z_2	α_1	α_2	α_3	X_1	Y_1	X_2	Y_2	X_3	Y_3	W_1	W_2
a_1	0	0	0	0	0	0	0	0	0	b1	0	b1	0	b1	0	1
				0	1	0	0	1	0	b1	0	b1	1	b2		
				1	0	0	1	0	0	b1	1	b2	0	b1		
a_2	1	0	0	0	0	1	0	1	b4	1	0	b1	1	b2	0	0
				0	1	0	0	0	b3	0	0	b1	0	b1		
				1	0	1	1	0	b4	1	1	b2	0	b1		
a_3	1	0	1	0	0	1	1	1	b4	1	1	b2	b4	1	1	0
				0	1	1	0	1	b4	1	0	b1	b4	1		
				1	0	0	0	1	b3	0	0	b1	b4	1		
a_4	1	1	0	0	0	1	1	0	b4	1	b4	1	0	b1	1	1
				0	1	1	1	1	b4	1	b4	1	1	b2		
				1	0	0	1	0	b3	0	b4	1	0	b1		
a_5	0	0	1	0	0	0	1	0	0	b1	1	b2	b3	0	0	1
				0	1	0	0	1	0	b1	0	b1	b4	1		
				1	0	1	0	1	1	b2	0	b1	b4	1		
a_6	1	1	1	0	0	0	0	1	b3	0	b3	0	b4	1	0	0
				0	1	1	1	0	b4	1	b4	1	b3	0		
				1	0	1	0	0	b4	1	b3	0	b3	0		
a_7	0	1	0	0	0	1	1	0	1	b2	b4	1	0	b1	0	0
				0	1	1	0	0	1	b2	b3	0	0	b1		
				1	0	0	0	0	0	b1	b3	0	0	b1		

Для синтезу КС 1, що має п'ять вхідних змінних ($\alpha_1, \alpha_2, \alpha_3, Z_1 Z_2$) і 6 виходів (X1, Y1, X2, Y2, X3, Y3), Необхідно отримати у мінімізованому вигляді наступну систему рівнянь:

$$\begin{cases} X_1 = f(\alpha_1, \alpha_2, \alpha_3, Z_1, Z_2)^t \\ Y_1 = f(\alpha_1, \alpha_2, \alpha_2, Z_1, Z_2)^t \\ X_2 = f(\alpha_1, \alpha_2, \alpha_3, Z_1, Z_2)^t \\ Y_2 = f(\alpha_1, \alpha_2, \alpha_2, Z_1, Z_2)^t \\ X_3 = f(\alpha_1, \alpha_2, \alpha_3, Z_1, Z_2)^t \\ Y_3 = f(\alpha_1, \alpha_2, \alpha_3, Z_1, Z_2)^t \end{cases} \qquad (24.1)$$

Булеві функції з системи (24.1) є не повністю визначеними, тому що вони визначені на 21 наборах із $2^5 = 32$ можливих. Тобто на решті 32 - 21 = 11 наборах ці функції можуть приймати байдужі значення. Для знаходження мінімальних форм не повністю визначених функцій скористаємося методом карт Карно.

Так як канонічні рівняння – це система булевих функцій, то і мінімізацію слід проводити відповідно до правил мінімізації системи булевих функцій. В даному випадку до визначення коефіцієнтів bi повинно бути однаковим для всієї системи функцій. Виберемо оптимальний варіант довизначення: b1=b2=b3=b4=0, порожні клітини можна довизначити до 0 або 1, виходячи з оптимальності покриттів.

z_1z_2 \ $\alpha_1\alpha_2$	00	01	11	10	00	01	11	10
00	0	1	b4	b4	0		b3	b4
01	0	1	b4	b3	0		b4	b4
11								
10	0	0	b3	b4	1		b4	b3
α_3	0				1			

Рис. 24.1. Мінімізація функції X1.

$$X_1 = \bar{z}_1\bar{\alpha}_1\alpha_2 \vee z_1\bar{\alpha}_1\alpha_3$$

z_1z_2 \ $\alpha_1\alpha_2$	00	01	11	10	00	01	11	10
00	b1	b2	1	1	b1		0	1
01	b1	b2	1	0	b1		1	1
11								
10	b1	b1	0	1	b2		1	0
α_3	0				1			

Рис. 24.2. Мінімізація функції Y1.

$$Y_1 = \bar{z}_1\alpha_1\alpha_2\bar{\alpha}_3 \vee \bar{z}_2\alpha_1\bar{\alpha}_2\bar{\alpha}_3 \vee z_1\alpha_2\alpha_3 \vee z_2\alpha_2\alpha_3 \vee \bar{z}_1\alpha_1\bar{\alpha}_2\alpha_3$$

z_1z_2 \ $\alpha_1\alpha_2$	00	01	11	10	00	01	11	10
00	0	1	b4	0	1		b3	1
01	0	0	b4	0	0		b4	0
11								
10	1	0	b4	1	0		b3	0
α_3	0				1			

Рис. 24.3. Мінімізація функції X2.

$$X_2 = \bar{z}_1\bar{z}_2\bar{\alpha}_1\alpha_2 \vee z_1\bar{z}_2\bar{\alpha}_2\bar{\alpha}_3 \vee \bar{z}_1\bar{z}_2\bar{\alpha}_2\alpha_3$$

z_1z_2 \ $\alpha_1\alpha_2$	00	01	11	10	00	01	11	10
00	b1	1	1	b1	b2		0	b2
01	b1	0	1	b1	b1		1	b1
11								
10	b2	0	1	b2	b1		0	b1
α_3	0				1			

Рис. 24.4. Мінімізація функції Y2.

$$Y_2 = \bar{z}_1\bar{z}_2\bar{\alpha}_1\alpha_2 \vee \alpha_1\alpha_2\bar{\alpha}_3 \vee z_2\alpha_2\alpha_3$$

z_1z_2 \ $\alpha_1\alpha_2$	00	01	11	10	00	01	11	10
00	0	0	0	1	b3		b4	b4
01	1	0	1	0	b4		b3	b4
11			.					
10	0	0	0	0	b4		b3	b4
α_3		0				1		

Рис. 24.5. Мінімізація функції X3.

$$X_3 = z_2\overline{\alpha}_1\overline{\alpha}_2\overline{\alpha}_3 \vee z_2\alpha_1\alpha_2\overline{\alpha}_3 \vee \overline{z}_1\overline{z}_2\alpha_1\overline{\alpha}_2\overline{\alpha}_3$$

z_1z_2 \ $\alpha_1\alpha_2$	00	01	11	10	00	01	11	10
00	b1	b1	b1	b2	0		1	1
01	b2	b1	b2	b1	1		0	1
11			.					
10	b1	b1	b1	b1	1		0	1
α_3		0				1		

Рис. 24.6. Мінімізація функції Y3.

$$Y_3 = \overline{z}_1\overline{z}_2\alpha_1\alpha_3 \vee z_2\overline{\alpha}_2\alpha_3 \vee z_1\overline{z}_2\overline{\alpha}_2\alpha_3$$

Довизначимо b1=b2=b3=b4=0, порожні клітини можна довизначити до 0 або 1, виходячи з оптимальності покриттів.

Для синтезу КС 2 необхідно отримати у мінімізованому вигляді іншу систему рівнянь:

$$\begin{cases} w_1 = f(\alpha_1,\alpha_2,\alpha_3)^t \\ w_2 = f(\alpha_1,\alpha_2,\alpha_3)^t \end{cases} \qquad (24.2)$$

α_1 \ $\alpha_2\alpha_3$	00	01	11	10
0	0	0	*	0
1	0	1	0	1

α_1 \ $\alpha_2\alpha_3$	00	01	11	10
0	1	1	*	0
1	0	0	0	1

Рис. 24.7. Мінімізація функції w_1 Рис. 24.8. Мінімізація функції w_2

$w_1 = \alpha_1\overline{\alpha}_2\alpha_3 \vee \alpha_1\alpha_2\overline{\alpha}_3$

$w_2 = \overline{\alpha}_1\overline{\alpha}_2 \vee \alpha_1\alpha_2\overline{\alpha}_3$

В результаті мінімізації отримали дві системи рівнянь: функції збудження елементів пам’яті автомата (25.3) та функції виходу (25.4)

$$\begin{cases} X_1 = \bar{z}_1\bar{\alpha}_1\alpha_2 \vee z_1\bar{\alpha}_1\alpha_3 \\ Y_1 = \bar{z}_1\alpha_1\alpha_2\bar{\alpha}_3 \vee \bar{z}_2\alpha_1\bar{\alpha}_2\bar{\alpha}_3 \vee z_1\alpha_2\alpha_3 \vee z_2\alpha_2\alpha_3 \vee \bar{z}_1\alpha_1\bar{\alpha}_2\alpha_3 \\ X_2 = \bar{z}_1\bar{z}_2\bar{\alpha}_1\alpha_2 \vee z_1\bar{z}_2\bar{\alpha}_2\bar{\alpha}_3 \vee \bar{z}_1\bar{z}_2\bar{\alpha}_2\alpha_3 \\ Y_2 = \bar{z}_1\bar{z}_2\bar{\alpha}_1\alpha_2 \vee \alpha_1\alpha_2\bar{\alpha}_3 \vee z_2\alpha_2\alpha_3 \\ X_3 = z_2\bar{\alpha}_1\bar{\alpha}_2\bar{\alpha}_3 \vee z_2\alpha_1\alpha_2\bar{\alpha}_3 \vee \bar{z}_1\bar{z}_2\alpha_1\bar{\alpha}_2\bar{\alpha}_3 \\ Y_3 = \bar{z}_1\bar{z}_2\alpha_1\alpha_3 \vee z_2\bar{\alpha}_2\alpha_3 \vee z_1\bar{z}_2\bar{\alpha}_2\alpha_3 \end{cases} \qquad (24.3)$$

$$\begin{cases} w_1 = \alpha_1\bar{\alpha}_2\alpha_3 \vee \alpha_1\alpha_2\bar{\alpha}_3 \\ w_2 = \bar{\alpha}_1\bar{\alpha}_2 \vee \alpha_1\alpha_2\bar{\alpha}_3 \end{cases} \qquad (24.4)$$

Для автомата Мілі функції виходу залежать (на відміну від автомата Мура) не лише від стану автомата, але й від вхідного сигналу. Тому система рівнянь для автомата Мілі в загальному вигляді записується так:

$$\begin{cases} X_1 = f(\alpha_1,\alpha_2,\alpha_3,Z_1,Z_2)^t \\ Y_1 = f(\alpha_1,\alpha_2,\alpha_2,Z_1,Z_2)^t \\ X_2 = f(\alpha_1,\alpha_2,\alpha_3,Z_1,Z_2)^t \\ Y_2 = f(\alpha_1,\alpha_2,\alpha_2,Z_1,Z_2)^t \\ X_3 = f(\alpha_1,\alpha_2,\alpha_3,Z_1,Z_2)^t \\ Y_3 = f(\alpha_1,\alpha_2,\alpha_3,Z_1,Z_2)^t \end{cases} \qquad (24.5)$$

$$\begin{cases} w_1 = f(\alpha_1,\alpha_2,\alpha_3,Z_1,Z_2)^t \\ w_2 = f(\alpha_1,\alpha_2,\alpha_3,Z_1,Z_2)^t \end{cases} \qquad (24.6)$$

24.3 Індивідуальні завдання

Завдання 1.

Використовуючи результати кодування вхідного і вихідного алфавітів автомата (тема № 22), результати кодування алфавіту станів (тема № 23) та матрицю переходів тригера (тема № 20) побудувати:

1) зазначену таблицю структурного автомата Мура;

2) суміщену таблицю формування функцій збудження елементів пам'яті і функцій виходів структурного автомата;

3) записати систему канонічних рівнянь в загальному вигляді;

4) виконати мінімізацію канонічних рівнянь.

Варіант №1							
У	y_1	y_2	y_2	y_3	y_4	y_3	y_5
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_4	a_2	a_3	a_4	a_2	a_5
x_2	a_2	a_3	a_3	a_2	a_5	a_7	a_2
x_3	a_3	a_2	a_1	a_1	a_3	a_4	a_3
x_4	a_5	a_6	a_7	a_4	a_1	a_5	a_1
x_5	a_7	a_1	a_5	a_7	a_6	a_3	a_4

Варіант №2							
У	y_1	y_3	y_2	y_4	y_4	y_3	y_5
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_2	a_1	a_2	a_1	a_7	a_7	a_3
x_2	a_3	a_2	a_4	a_2	a_2	a_5	a_6
x_3	a_4	a_4	a_1	a_3	a_6	a_2	a_4
x_4	a_6	a_3	a_7	a_5	a_5	a_6	a_7
x_5	a_7	a_7	a_3	a_6	a_3	a_4	a_2

Варіант №3							
У	y_1	y_2	y_3	y_3	y_4	y_5	y_5
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_4	a_2	a_3	a_4	a_2	a_5
x_2	a_2	a_3	a_3	a_2	a_5	a_7	a_2
x_3	a_3	a_2	a_1	a_1	a_3	a_4	a_3
x_4	a_5	a_6	a_7	a_4	a_1	a_5	a_1
x_5	a_7	a_1	a_5	a_7	a_6	a_3	a_4

Варіант №4							
У	y_1	y_3	y_2	y_4	y_5	y_6	y_5
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_2	a_1	a_2	a_1	a_7	a_7	a_3
x_2	a_3	a_2	a_4	a_2	a_2	a_5	a_6
x_3	a_4	a_4	a_1	a_3	a_6	a_2	a_4
x_4	a_6	a_3	a_7	a_5	a_5	a_6	a_7
x_5	a_7	a_7	a_3	a_6	a_3	a_4	a_2

Варіант №5							
У	y_1	y_2	y_1	y_3	y_4	y_1	y_5
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_4	a_2	a_3	a_7	a_7	a_3
x_2	a_2	a_3	a_3	a_2	a_2	a_5	a_6
x_3	a_3	a_2	a_1	a_1	a_6	a_2	a_4
x_4	a_5	a_6	a_7	a_4	a_5	a_6	a_7
x_5	a_7	a_1	a_5	a_7	a_3	a_4	a_2

Варіант №6							
У	y_1	y_1	y_2	y_4	y_4	y_5	y_6
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_2	a_1	a_2	a_1	a_4	a_2	a_5
x_2	a_3	a_2	a_4	a_2	a_5	a_7	a_2
x_3	a_4	a_4	a_1	a_3	a_3	a_4	a_3
x_4	a_6	a_3	a_7	a_5	a_1	a_5	a_1
x_5	a_7	a_7	a_3	a_6	a_6	a_3	a_4

Варіант №7							
У	y_1	y_5	y_1	y_3	y_4	y_6	y_5
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_4	a_2	a_3	a_7	a_7	a_3
x_2	a_2	a_3	a_3	a_2	a_2	a_5	a_6
x_3	a_3	a_2	a_1	a_1	a_6	a_2	a_4
x_4	a_5	a_6	a_7	a_4	a_5	a_6	a_7
x_5	a_7	a_1	a_5	a_7	a_3	a_4	a_2

Варіант №8							
У	y_1	y_2	y_3	y_4	y_5	y_6	y_7
X \ A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_2	a_1	a_2	a_1	a_4	a_2	a_5
x_2	a_3	a_2	a_4	a_2	a_5	a_7	a_2
x_3	a_4	a_4	a_1	a_3	a_3	a_4	a_3
x_4	a_6	a_3	a_7	a_5	a_1	a_5	a_1
x_5	a_7	a_7	a_3	a_6	a_6	a_3	a_4

Варіант № 9								
У	y_1	y_2	y_3	y_4	y_5	y_2	y_3	y_4
X \ A	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_3	a_4	a_1	a_7	a_4	a_6	a_1	a_3
x_2	a_6	a_6	a_6	a_3	a_0	a_2	a_6	a_7
x_3	a_1	a_5	a_7	a_1	a_5	a_7	a_7	a_1

Варіант № 10								
У	y_1	y_3	y_2	y_4	y_5	y_2	y_3	y_5
X \ A	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	a_1	a_6	a_2	a_7	a_4	a_2	a_6	a_3
x_2	a_3	a_0	a_5	a_3	a_0	a_5	a_0	a_7
x_3	a_6	a_2	a_3	a_1	a_5	a_3	a_5	a_1

Варіант №11				
У	y_2	y_3	y_1	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_1	a_3	a_2	a_2
x_2	a_3	a_4	a_3	a_4
x_3	a_2	a_4	a_1	a_1

Варіант №12				
У	y_1	y_2	y_3	y_4
X \ A	a_1	a_2	a_3	a_4
x_1	a_3	a_1	a_2	a_4
x_2	a_2	a_2	a_4	a_2
x_3	a_4	a_3	a_1	a_1

25.4 Контрольні запитання

1. Як складаються зазначені таблиці переходів структурного автомата Мілі?
2. Як складаються зазначені таблиці переходів структурного автомата Мура?
3. Як складається таблиця формування функцій збудження елементів пам'яті і функцій виходів структурного автомата?
4. Сформулюйте особливості мінімізації функцій збудження елементів пам'яті автомата.
5. Сформулюйте особливості мінімізації функцій виходів автомата.
6. Чим відрізняється система канонічних рівнянь автомата Мілі від системи канонічних рівнянь автомата Мура?

25 ПОБУДОВА ФУНКЦІОНАЛЬНОЇ СХЕМИ АВТОМАТА

25.1 Мета заняття

Засвоїти методику і набути практичних навичок запису канонічних рівнянь для функцій збудження елементів пам’яті і функцій виходів автомата. Знайти мінімальні форми перелічених функцій. Побудувати функціональну схему автомата в заданому функціональному базисі.

25.2 Короткі основні відомості з теорії та вирішення типових завдань

Останнім етапом канонічного методу структурного синтезу автоматів є побудова функціональної схеми автомата.

На підставі отриманих виразів для булевих функцій збудження елементів пам'яті автомата і булевих функцій виходів автомата будуються комбінаційна схема функцій збудження і комбінаційна схема формування вихідних сигналів автомата. Елементи пам'яті підключаються до побудованої комбінаційної схеми. Функціональні схеми автоматів Мура і Мілі відрізняється тільки комбінаційною схемою формування вихідних сигналів, яка будується на підставі рівнянь для булевих функцій виходів. Відзначимо, що реалізація комбінаційних схем може бути виконана в будь-якому функціонально-повному базисі.

В якості елементів пам’яті використовуються тригери. Кількість тригерів відповідає кількості розрядів коду алфавіту станів автомата. Для забезпечення стійкої роботи автомата, для ліквідації гонок в автоматі використовується подвійна пам'ять.

Елементи пам'яті підключаються до побудованих комбінаційних схем.

Приклад.

Використовуючи систему канонічних рівнянь (24.5) та (24.6) автомата Мура, синхронний ХУ-тригер із матрицею переходів (таблиця 24.6), функціонально повний базис логічних елементів І-АБО-НІ (кількість входів від 1 до 6, навантажувальна здатність до 10). Побудувати функціональну схему автомата Мура.

Рішення.

Для побудови функціональної схеми автомата Мура в базисі І-АБО-НІ є всі необхідні дані: обрані елементи пам'яті, закодовані вхідний, вихідний алфавіти й алфавіт станів, отримані булеві функції в потрібному базисі.

Для забезпечення стійкості функціонування цифрового автомата використовується двоступенева пам'ять. У цьому випадку кожен елемент пам'яті дублюється і перепис інформації з нижнього елемента пам'яті в верхній здійснюється по синхросигналу. В якості тригерів першого ступеню будемо використовувати дво-входовий ХУ-тригер. В якості тригерів другого ступеню будемо використовувати одно-входовий D-тригер.

Функціональна схема автомата Мура представлена на рис. 25.1.

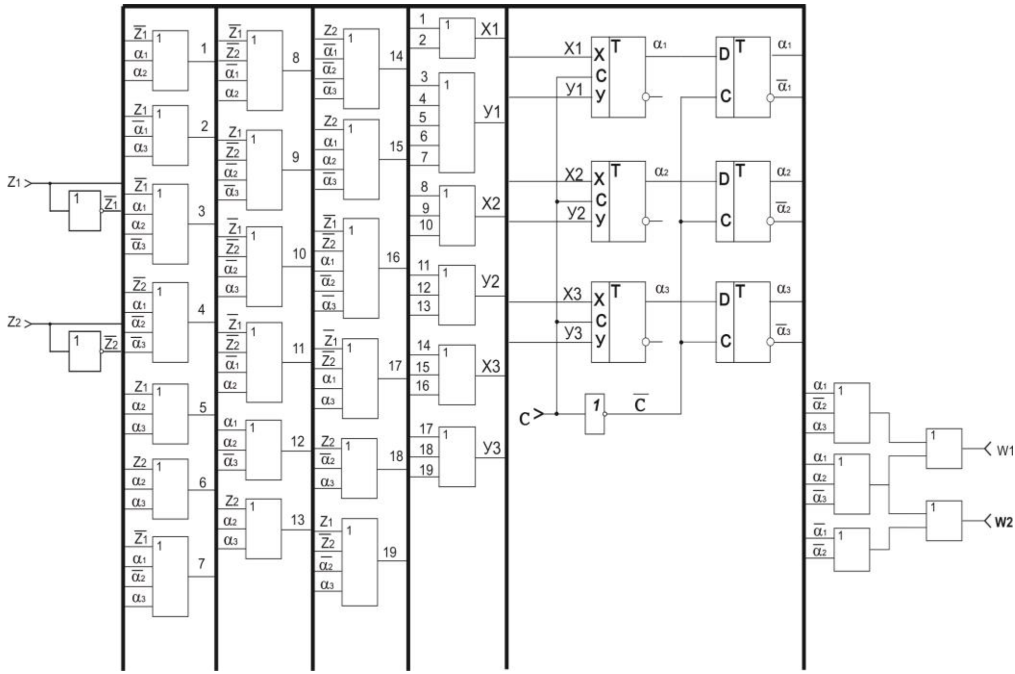


Рис. 25.1. Функціональна схема автомата Мура.

25.3 Індивідуальні завдання

Завдання 1.

Використовуючи системи канонічних рівнянь (індивідуальне завдання тема № 24), побудувати функціональну схему автомата.

В якості елементів пам'яті використати синхронний ХУ-тригер із матрицею переходів (індивідуальне завдання тема №20), функціонально повний базис логічних елементів:

а) І-АБО-НІ (кількість входів від 1 до 6, навантажувальна здатність до 10).

б) АБО-НІ (кількість входів від 1 до 6, навантажувальна здатність до 10).

в) І-НІ (кількість входів від 1 до 6, навантажувальна здатність до 10).

Побудувати функціональну схему автомата Мура.

25.4 Контрольні запитання

1. Які вихідні дані потрібні для побудови функціональної схеми автомата Мура?
2. Які вихідні дані потрібні для побудови функціональної схеми автомата Мілі?
3. Які методи використовуються для забезпечення стійкої роботи автомата?
4. Як визначити кількість елементів пам'яті автомата?
5. Чим відрізняються принципи побудови функціональної схеми автомата Мілі від принципів побудови функціональної схеми автомата Мура?

26 МІКРОПРОГРАМНІ АВТОМАТИ. ПОБУДОВА ГРАФА АВТОМАТІВ МІЛІ, МУРА по ГСА.

26.1 Мета заняття

Засвоїти методику і набути практичних навичок побудови змістовної ГСА мікропрограмного автомату (МПА) та її розмітки для автоматів Мілі та Мура.

26.2 Короткі основні відомості з теорії та вирішення типових завдань

Основні поняття. Принцип мікропрограмного управління

Мікропрограмні автомати представляють собою наступний крок на шляху ускладнення інтелекту цифрових схем. На основі мікропрограмних автоматів можна будувати пристрої, які працюють за досить складними алгоритмами, виконують різні функції, які визначаються вхідними сигналами, видають складні послідовності вихідних сигналів. При цьому алгоритм роботи мікропрограмного автомата може бути легко змінений заміною прошивки ПЗУ (постійний запам'ятовуючий пристрій).

На відміну від пристроїв на "жорсткій" логіці, що розглядалися раніше і принцип роботи яких однозначно визначається використовуваними елементами і способом їх з'єднання, мікропрограмні автомати за допомогою однієї і тієї ж схеми можуть виконувати самі різні функції. Тобто вони набагато гнучкіші, ніж схеми на "жорсткій" логіці. До того ж проектувати мікропрограмні автомати з точки зору схемотехніки досить просто.

При побудові пристроїв середньої і великої складності доцільно використовувати принцип мікропрограмного управління, що полягає в наступному:

1) будь-яка операція, реалізована пристроєм, розглядається як складна дія, яка поділяється на послідовність елементарних дій, званих **мікроопераціями**;

2) для управління порядком проходження мікрооперацій використовуються логічні умови x_i, що приймають залежно від результатів виконання мікрооперацій значення 1 або 0;

3) процес виконання операцій в пристрої описується у формі алгоритму, представленого в термінах мікрооперацій і логічних умов і називається **мікропрограмою**;

4) мікропрограма використовується як форма представлення функції пристрою, на основі якої визначаються його структура та порядок функціонування.

Все сказане можна розглядати, як змістовний опис принципу мікропрограмного управління.

Концепція управляючого та операційного автоматів

При використанні описаного принципу прийнято ділити дискретні пристрої на дві частини: операційний автомат (ОА) і управляючий автомат (УА) (Рис. 26.1).

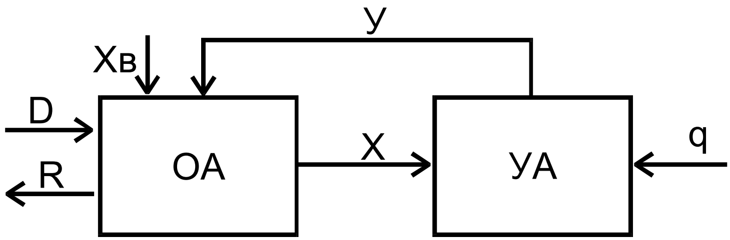


Рисунок 26.1- Узагальнена структурна схема мікропрограмного дискретного пристрою

ОА призначений для зберігання інформації, що надходить на вхід D, видачі результатів виконання операцій R, виконання заданого набору мікрооперацій, вироблення значень логічних умов $X=(x_0,x_1,..., x_m)$, які є для управляючого автомата тими, що сповіщають. УА генерує послідовність управляючих сигналів $Y=(y_1,y_2,...,y_n)$ відповідно до заданої мікропрограми і зі значеннями логічних умов X. Кожен управляючий сигнал ініціює виконання відповідної мікрооперації в ОА.

У загальному випадку ДП призначається для виконання ряду мікропрограм, і на УА подається зовнішній сигнал q, відповідно до якого починається виконання тієї чи іншої мікропрограми. Якщо ДП є частиною системи обробки інформації, то він може також обмінюватися спеціальними сигналами логічних умов X_B та управління Y_B з іншими блоками системи.

До складу ОА входять головним чином типові функціональні вузли: регістри, лічильники, суматори, дешифратори, шифратори, арифметико-логічні пристрої (АЛП), схеми порівняння, блоки пам'яті, схеми пересилання даних і т. п. Число елементів пам'яті (ЕП), що містяться в ОА, визначається розрядністю оброблюваних даних nД, яка може бути досить великою. Однак трудомісткість і складність проектування ОА, як правило, слабо залежать від nД в силу широкого використання стандартних вузлів. Таким чином, ОА є виконавчою частиною пристрою; його склад і структура можуть бути однаковими для реалізації багатьох алгоритмів одного класу.

Елементарний неподільний акт обробки інформації в операційному автоматі, що відбувається протягом одного моменту автоматного часу (одного такту роботи автомата), називається ***мікрооперацією***. Прикладами мікрооперацій можуть служити «Зсув інформації», «+1», «Інверсія змінної» і т. д.

Якщо в операційному автоматі одночасно реалізується декілька мікрооперацій, то така множина мікрооперацій називається ***мікрокомандою***. Не виключений випадок, коли множина мікрооперацій, що утворюють мікрокоманду, є порожньою. Реалізація такої мікрокоманди в операційному автоматі рівносильна відсутності виконання будь-яких елементарних операцій. У разі синхронних дискретних пристроїв порожня мікрокоманда інтерпретується як пропуск такту, коли ніякі сигнали від управляючого автомата на операційний автомат не надходять.

Мікрооперації побуджуються вихідними сигналами управляючого автомата, а їх послідовність у часі визначається функціями переходу управляючого автомата.

Сукупність мікрокоманд і функцій переходу утворює ***мікропрограму***. Таким чином, для опису мікропрограми необхідно задати множину мікрокоманд і функцій переходу, що визначають порядок їх виконання. Для опису мікропрограм зручно використовувати мову граф-схем алгоритмів (ГСА).

Управляючі автомати з жорсткою і програмуємою логікою

Обсяг обладнання УА залежить від складності реалізованого алгоритму і від структури цього автомата, яку можна виконати в трьох варіантах.

1. УА з жорсткою (схемною, довільною) логікою, при якій перемикальні функції, необхідні для формування заданої послідовності управляючих сигналів У, реалізуються за допомогою логічних елементів з довільними зв'язками (зазвичай із застосуванням схем з малою і середньою ступенями інтеграції). Тут використовується апаратний підхід до реалізації пристрою.

2. УА із збереженою в пам'яті (гнучкою, програмною) логікою, при якій сигнали У виробляються на основі сукупності управляючих слів, що зберігаються в пам'яті автомата. У цьому випадку написані мікропрограми використовуються в явній формі і зазвичай записуються в постійні запам'ятовуючі пристрої (ПЗУ), виконані на основі напівпровідникових БІС великої ємності, що дозволяє забезпечити регулярність структури УА і його компактність; тут використовується апаратно-програмний підхід до реалізації пристрою.

3. УА на основі програмованих логічних матриць (ПЛМ), в яких задані функції реалізуються за допомогою БІС ПЛМ, що дозволяє поєднувати чимало достоїнств перших двох.

Розглянемо порядок проектування мікропрограмного ДП, який складається з наступних основних етапів:

Запис алгоритму. За описом окремих алгоритмів, що реалізуються пристроєм, складається їх формалізований запис у вигляді граф-схем алгоритмів (ГСА). Для цього складається список необхідних мікрооперацій y_j, і відповідних їм управляючих сигналів y_j, а також логічних умов x_i;. далі при необхідності, проводиться мінімізація числа вершин ГСА і складається об'єднана ГСА, що є формою побудови ДП для виконання наступних етапів.

Побудова ОА. У загальному випадку ОА може бути побудований за канонічною схемою автомата і містить три основні частини: блок елементів пам'яті для зберігання операндів, а також проміжних і кінцевих результатів; комбінаційну схему, що реалізує набір мікрооперацій; комбінаційну схему, що виробляє значення логічних умов. Як вже зазначалося, при побудові ОА доцільно застосовувати типові вузли, а також прагнути використовувати окремі вузли для виконання декількох мікрооперацій.

Побудова УА. Спочатку вибирають варіант структури УА, враховуючи вимоги швидкодії, допустимий обсяг апаратури й інші обмеження. Далі здійснюється синтез УА відповідно до процедури, яка залежить від прийнятої структури автомата.

В результаті виконання цих етапів складають структурні схеми ОА і УА і переходять до технічного проектування, яке включає питання практичної реалізації схеми пристрою на обраній елементній базі, введення необхідних розв'язуючих, підсилюючих та формуючих каскадів, компоновку деталей на платах, складання монтажних схем і видачу технічної документації.

Граф-схеми алгоритмів (ГСА)

ГСА - це орієнтований зв'язний граф, що задає послідовність виконання операцій даного алгоритму і містить ряд операторних і умовних вершин, а також одну початкову та одну кінцеву вершини. Операторною називається вершина, якій співставляється одна або декілька мікрооперацій і відзначається відповідними управляючими сигналами У, а умовною - вершина, якій співставляється деяка логічна умова *X*.

Будь-яка вершина ГСА, крім вершини «Початок», має по одному входу. Вершина «Початок» входів не має. Вершина «Початок» и будь-яка операторна вершина мають по одному виходу. Вершина «Кінець» виходів не має. Кожна умовна вершина має два виходи, які позначаються символами «Так» і «Ні»: Замість цих символів можуть бути використані цифри «1» і «0» відповідно. Зображення вершин «Початок», «Кінець», операторної вершини і умовної вершини ГСА представлено на рисунку 26.2.

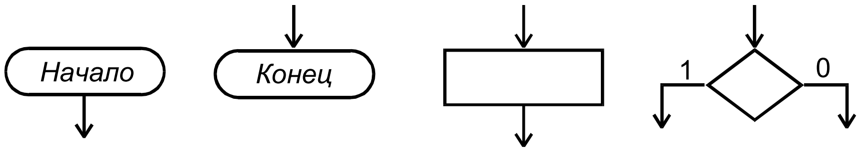


Рисунок 26.2-Граф-схеми алгоритмів

ГСА складають так, щоб забезпечити виконання необхідних операцій і перевірку логічних умов відповідно зі словесним описом алгоритму.

На підставі переліку мікрооперацій і функціональних вузлів, що реалізують їх, складається структурна схема ОА. На структурній схемй ОА стрілками показують шини, по яких передається інформація, та управляючі сигнали у, що керують роботою окремих вузлів або передачею інформації по шинах.

ГСА повинна задовольняти наступним умовам:

1. Входи і виходи вершин з'єднуються один з одним за допомогою дуг, спрямованих завжди від виходу до входу.

2. Кожен вихід з'єднаний тільки з одним входом.

3. Будь-який вхід з'єднується, принаймні, з одним виходом.

4. Будь-яка вершина ГСА лежить, принаймні, на одному шляху з вершини «Початок» в вершину «Кінець».

5. Один з виходів умовної вершини може з'єднуватися з її входом, що неприпустимо для операторної вершини. Такі умовні вершини іноді називаються зворотними.

6. У кожній умовній вершині записується логічна умова з множини логічних умов. Дозволяється в різних умовних вершинах записувати однакові логічні умови.

7. У кожній операторної вершині записується оператор, що представляє собою вихідний сигнал або сукупність вихідних сигналів управляючого автомата. Дозволяється в різних операторних вершинах записувати однакові оператори.

Змістовні граф-схеми алгоритмів

Зазвичай при проектуванні різних пристроїв попередньо складається так звана змістовна ГСА, в якій всередині умовних і операторних вершин записані логічні умови і мікрооперації в змістовних термінах.

Як приклад побудуємо змістовну ГСА пристрою, що обчислює функцію знака числа:

$$S = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases}$$

Відповідна змістовна ГСА представлена на рисунку. 26.3.

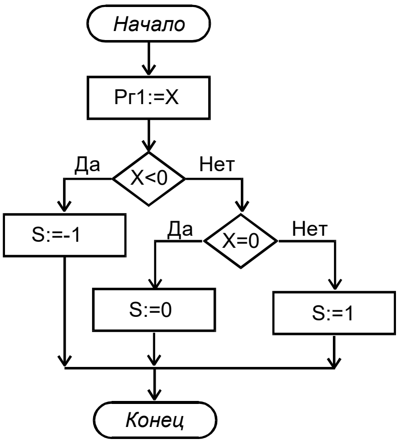


Рисунок 26.3 - Змістовна ГСА функції визначення знака числа.

Синтез управляючого автомата по граф-схемі алгоритму

Кінцевий управляючий автомат, який реалізує мікропрограму роботи дискретного пристрою, прийнято називати мікропрограмним автоматом. Як уже зазначалося, мікропрограма відображається за допомогою ГСА. Розглянемо послідовність етапів синтезу управляючого автомата по його ГСА.

1. Запис словесного алгоритму функціонування операційного автомата (виконуваних операцій) з урахуванням структури операційного автомата.
2. Побудова змістовної ГСА функціонування операційного автомата.
3. Побудова відміченої ГСА з урахуванням типу автомата.
4. Побудова графа переходів автомата або таблиці переходів.
5. Проведення структурного синтезу автомата за його графом переходів з використанням відомих методів, наприклад, за допомогою канонічного методу структурного синтезу.

Побудова відміченої ГСА проводиться по змістовній ГСА. Для автоматів Мілі та Мура процедура розмітки має відмінності.

Побудова відміченої ГСА автомата Мілі

Якщо необхідно побудувати мікропрограмний автомат Мілі, то змістовна ГСА управляючого автомата розмічається відповідно до наступних правил:

1) символом стану a_1 відзначити вхід вершини, наступної за вершиною «Початок», а також вхід вершини «Кінець»;

2) входи всіх вершин, що знаходяться за операторними, повинні бути відзначені символами а з послідовними індексами;

3) якщо вихід вершини відмічається, то тільки одним символом;

4) входи різних вершин, за винятком вершини «Кінець», відзначаються різними символами;

5) змістовні терміни мікрооперацій і логічних умов замінюються їх умовними позначеннями: у кожній операторній вершині послідовно проставляються символи вихідних сигналів, якщо в різних операторних вершинах записані однакові мікрооперації, то дозволяється їх відмічати однаковими символами вихідних сигналів; якщо в різних умовних вершинах записані однакові логічні умови, то дозволяється їх відмічати однаковими символами вхідних сигналів.

Відмічена ГСА для змістовної ГСА (мал. 26.4.) Після розмітки за наведеним алгоритмом представлена на малюнку. 26.5.

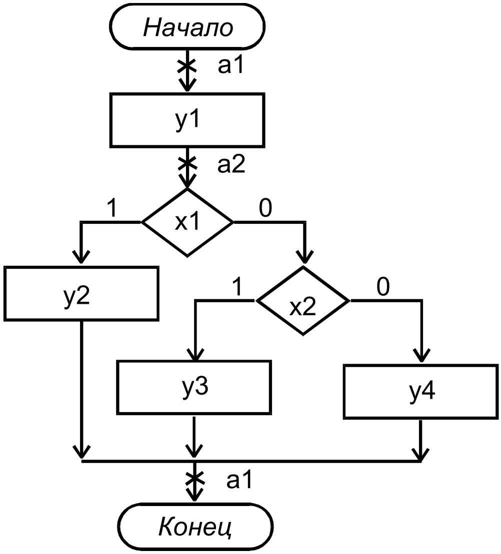


Рисунок 26.5- Відмічена ГСА автомата Мілі

Побудова відміченої ГСА автомата Мура

Якщо необхідно побудувати мікропрограмний автомат Мура, то змістовна ГСА управляючого автомата розмічається відповідно до наступних правил:

1) символом a_1 відмічаються вершини «Початок» і «Кінець»;

2) різні операторні вершини відмічаються різними символами;

3) всі операторні вершини повинні бути відмічені.

4) змістовні терміни мікрооперацій і логічних умов замінюються їх умовними позначеннями.

Змістовна ГСА (рис. 26.4.) після розмітки за наведеним алгоритмом представлена на рисунок 26.6.

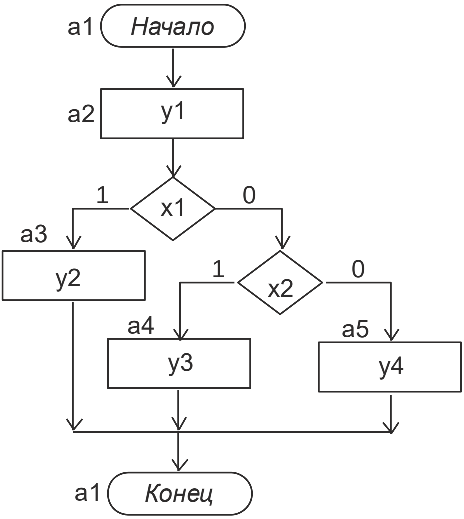


Рисунок 26.6- Відмічена ГСА автомата Мура

Після отримання відміченої ГСА будується граф переходів автомата. Він має стільки різних вершин, скільки різних букв a_i з індексами є на відміченій ГСА. Кожна вершина графа переходів автомата відзначається буквою а з відповідним індексом.

Між двома вершинами графа проводиться дуга, якщо на відміченій ГСА між вершинами з мітками a_i і a_k, є шлях. Над дугою ставиться вхідний сигнал, що дорівнює кон'юнкції логічних умов відповідного шляху у відміченій ГСА. При цьому виконанню логічної умови відповідає змінна без заперечення, а невиконанню логічної умови - змінна з запереченням на відповідній дузі графа переходів автомата.

Якщо у відміченій ГСА між згаданими вершинами з мітками a_i і a_k є кілька шляхів, то в графі переходів автомата на дузі, що зв'язує a_i і a_k через символ диз'юнкції перераховуються всі кон'юнкції, що відповідають наявним шляхам.

Якщо будується граф переходів автомата Мура, то символи мікрооперацій (вихідні сигнали управляючого автомата) записуються поряд з відповідними вершинами. Для автомата Мілі символи мікрооперацій записуються на відповідних дугах при кон'юнкції логічних умов, що описують шлях через операційну вершину з розглянутою мікрооперацією.

Якщо у відміченій ГСА є ***безумовний*** перехід між операторними вершинами, тобто є шлях, що не проходить ні через які умовні вершини, то на графі переходів автомата йому відповідає дуга, якій приписується вхідний сигнал «1», що показує, що даний перехід в автоматі здійснюється при надходженні чергового синхросигналу.

Надалі синтез проводиться за допомогою описаного раніше методу структурного синтезу. Підкреслимо, що вхідними сигналами синтезованого структурного автомата є кон'юнкції булевих змінних (або диз'юнкції кон'юнкцій), кожна з яких відображає шлях через відповідні умовні вершини відміченої ГСА, а вихідними сигналами - мікрооперації, що позначають або вершини, або дуги графа переходів автомата, залежно від його типу. Використовуючи канонічний метод структурного синтезу, можна побудувати функціональну схему цифрового автомата.

Приклад синтезу мікропрограмного автомата

Побудова змістовної ГСА виконання операції додавання і віднімання чисел, представлених в форматі з фіксованою комою

Число X з фіксованою комою представляється правильним двійковим дробом виду $X = \pm, x_1 x_2 \ldots x_n$, де x_i - 0 або 1, (i= 1, 2, . . ., n). Знак числа кодується двійковими символами 0 і 1. Знак «+» представляється символом 0 і знак «-» - символом 1.

При паралельному способі виконання операції додавання для представлення від'ємних чисел використовуються модифіковані обернені коди. Від'ємне число $X = -, x_1 x_2 \ldots x_n$ в модифікованому оберненому коді має вигляд $(X)_{обр} = \bar{1}\bar{1}, \bar{x}_1 \bar{x}_2 \ldots \bar{x}_n$, де старший розряд цілої частини використовується для представлення знака числа і наступний розряд цілої частини - для контролю переповнення розрядної сітки суматора. Для додавання двійкових чисел з фіксованою комою при використанні обернених модифікованих кодів застосовується наступний алгоритм:

1) якщо знак операнда додатний, то операнд вступає операцію додавання в прямому модифікованому коді; якщо знак операнда від’ємний, то операнд вступає в операцію у оберненому модифікованому коді;

2) проводиться додавання двійкових кодів операндів за всіма розрядами, включаючи знакові розряди; при цьому спадаюча одиниця зі старшого знакового розряду передається з якості одиниці переносу в молодший розряд суми;

3) якщо значення знакових розрядів суми дорівнює 00, то сума має додатне значення і представлена в прямому коді;

4) якщо значення знакових розрядів суми дорівнює 11, то сума має від'ємне значення і представлена у оберненому коді, для отримання прямого коду суми необхідно інвертувати значення в цифрових розрядах суми;

5) якщо значення в знакових розрядах суми дорівнює 01 або 10, то результат переповнює розрядну сітку машини.

Розглянемо схему операційної частини пристрою, орієнтованого тільки на виконання операції додавання (малюнок. 26.7.). У цьому пристрої для додавання використовується п + 2-розрядний накопичувальний суматор. Суматор охоплений ланцюгом циклічного переносу, по якому сигнал переносу із старшого знакового розряду суматора надходить на вхід молодшого розряду суматора. Передбачається, що перший доданок зберігається на суматорі в прямому коді. Другий доданок зберігається на регістрі Рг. Код операнда $X = x_0 x_1 x_2 \ldots x_n$ надходить по вхідний шині X. Результат операції видається в паралельному коді на вихідній шині У, $У = y_0 y_1 y_2 \ldots y_n$. Значення y_0 відповідає старшому знаковому розряду коду, що зберігається на суматорі.

Відповідно до прийнятого алгоритму додавання в операційній частині пристрою виконуються наступні мікрооперації:

M1: См:=(Рг)$_{обр}$; *M6: Рг[0]:= Рг[0];*

M2: См:=Рг; *M7: Рг:=X;*

M3: Додавання на суматорі *M8: Рг:=0.*

M4: См:=0; *M9: ПП:=1.*

M5: См:=(См)$_{обр}$;

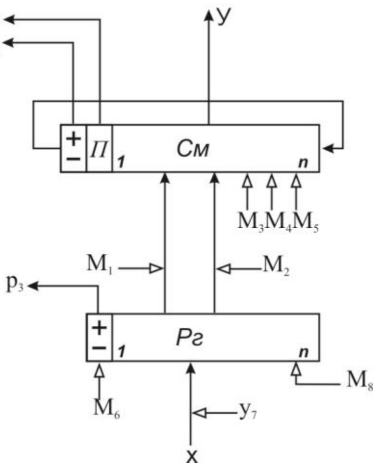


Рисунок. 26.7. Операційний пристрій для виконання операції додавання.

Передачі оберненого і прямого коду з регістра на суматор виконуються мікроопераціями М1 і М2. При передачі оберненого коду в знакові розряди суматора заноситься код 11. За сигналом М3 = 1 проводиться включення суматора на додавання. При цьому сигнал М3 = 1 включає ланцюги міжрозрядного переносу, які блокуються, якщо М3 = 0. Для додавання кодів, збережених на суматорі і регістрі, необхідно одночасно виконати наступні дві мікрооперації: М3 = 1, М2: См:=Рг. В результаті цього на суматорі виконується дія См:=См+Рг. Мікрооперація М5 використовується для інвертування цифрових розрядів коду, що зберігається на суматорі. При виконанні мікрооперації М5 значення знакових розрядів суматора не змінюється.

Мікрооперація М6 використовується для інвертування знаку коду, що зберігається на регістрі.

Для управління процесом додавання використовуються сигнали p_1, p_2, p_3, Сигнал p_1 визначає значення знакового розряду суматора sign См, сигнал p_2 - значення розряду переповнення См [П], та сигнал p_3 - значення знакового розряду регістра sign Рг.

Нехай на момент початку операції додавання на суматорі зберігається прямий код першого доданка і на регістрі - прямий код другого доданка. Мікропрограма операції додавання наведена на малюнку. 26.8. У мікропрограмі поруч з операторними вершинами вказані найменування сигналів, що приймають одиничне значення при виконанні відповідних операторів мікропрограми.

Якщо код має від'ємний знак sign См =1, то виконання мікропрограми починається з інвертування коду першого доданка, що зберігається на суматорі. Далі, в залежності від знаку другого доданка sign Рг проводиться додавання коду, що зберігається на суматорі, з прямим кодом (sign Рг = 0) або оберненим кодом (sign Рг =1) другого доданка, що зберігається на регістрі. Якщо знак результату sign См = 1, то проводиться перетворення оберненого коду від'ємного результату в прямий код. Переповненню розрядної сітки суматора відповідає нерівнозначність значень p_1 и p_2, що відповідають знаковим розрядами суматора. Якщо, $\overline{p_1} p_2 \vee p_1 \overline{p_2} = 1$ то в центральний пристрій управління видається сигнал М9, за яким признаку переповнення (ПП) присвоюється значення 1.

Час виконання мікропрограми, тобто час виконання операції додавання залежить від значень доданків. Якщо перший доданок і результат представляються додатними значеннями і не відбувається переповнення розрядної сітки суматора, то операція додавання виконується за один такт. У найгіршому випадку для виконання операції додавання потрібно чотири такти машинного часу.

Приклад побудови відміченої ГСА автомата Мілі

Проведемо розмітку змістовної ГСА відповідно до наступних правил:

1) символом стану a_1 відмітимо вхід вершини, наступної за вершиною «Початок», а також вхід вершини «Кінець»;

2) входи всіх вершин, наступних за операторними, відзначимо символами а з послідовними індексами;

3) якщо вихід вершини відмічається, то тільки одним символом;

4) входи різних вершин, за винятком вершини «Кінець», відмічаються різними символами;

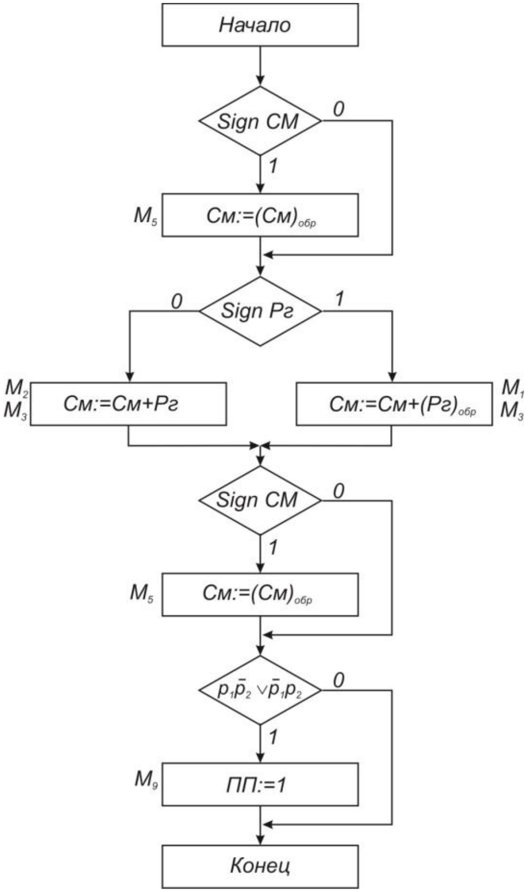


Рисунок. 26.8. Змістовна граф схема операції додавання чисел з фіксованою комою

5) змістовні терміни мікрооперацій і логічних умов замінюються їхніми умовними позначеннями: у кожній операторній вершині послідовно проставляються символи вихідних сигналів, якщо в різних операторних вершинах записані однакові мікрооперації, то дозволяється їх відзначати однаковими символами вихідних сигналів; якщо в різних умовних вершинах записані однакові логічні умови, то дозволяється їх відзначати однаковими символами вхідних сигналів.

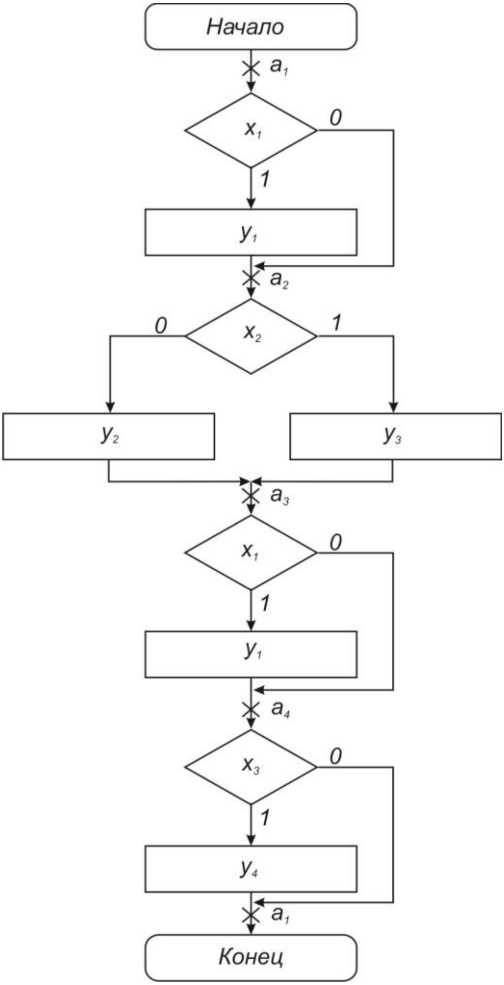


Рисунок 26.9.- Відмічена ГСА автомата Мілі

Відмічена ГСА автомата Мілі для змістовної ГСА (мал. 26.8.) після розмітки за наведеним алгоритмом представлена на малюнку. 26.9.

За відміченою ГСА можна побудувати таблицю переходів автомата. (Таблиця 26. 1.)

Таблиця 26.1

Вхідний сигнал / Стан	x_1	$\bar{x}_1$	x_2	$\bar{x}_2$	x_3	$\bar{x}_3$
a_1	a_2/ y_1	a_2/-				
a_2			a_3/ y_3	a_3/ y_2		
a_3	a_4/ y_1	a_4/-				
a_4					a_1 /y_4	a_1/-

На перетині рядка стану, із якого виходить автомат і стовпчика вхідного сигналу проставляється стан, в який переходить автомат під впливом даного вхідного сигналу і вихідний сигнал, що формується в результаті цього переходу.

Також можна побудувати граф автомата Мілі. Він матиме 4 різних вершини, тому що є 4 різних стани на відміченій ГСА.

Між двома вершинами графа проходить дуга, якщо на відміченій ГСА між вершинами з мітками a_i і a_k, проходить шлях. Над дугою ставиться вхідний сигнал, що дорівнює кон'юнкції логічних умов відповідного шляху у відміченій ГСА. При цьому виконанню логічної умови відповідає змінна без заперечення, а невиконанню логічної умови - змінна з запереченням на відповідній дузі графа переходів автомата.

Якщо у відміченій ГСА між згаданими вершинами з мітками a_i та a_k проходять кілька шляхів, то в графі переходів автомата на дузі, що зв'язує a_i та a_k через символ диз'юнкції перераховуються всі кон'юнкції, що відповідають наявним шляхам.

Граф автомата Мілі, побудований по таблиці переходів (таблиця 26.1) представлений на малюнку 26.10.

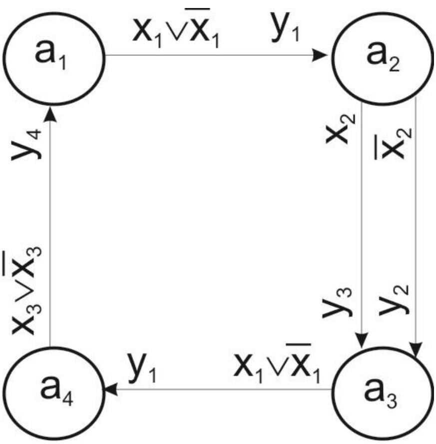


Рисунок 26.10.- Граф автомата Мілі.

Надалі синтез автомата може проводитися за допомогою описаного раніше канонічного методу структурного синтезу автомата.

Приклад побудови відміченої ГСА автомата Мура

Якщо необхідно побудувати мікропрограмний автомат Мура, то змістовна ГСА управляючого автомата розмічається відповідно до наступних правил:

1) символом a_1 відмічаються вершини «Початок» та «Кінець»;

2) різні операторні вершини відзначаються різними символами;

3) всі операторні вершини повинні бути відмічені.

4) змістовні терміни мікрооперацій і логічних умов замінюються їх умовними позначеннями.

Відмічена ГСА автомата Мура (по змістовній ГСА малюнок 26.8.) після розмітки за наведеним алгоритмом представлена на малюнку 26.11.

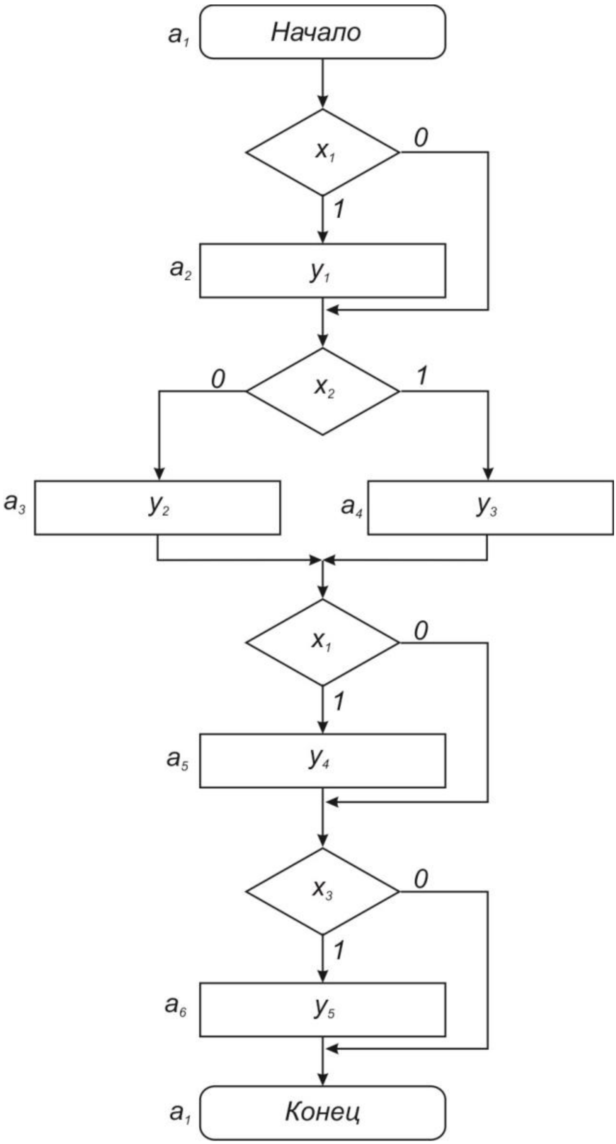


Рисунок 26.11.- Відмічена ГСА автомата Мура

За відміченою ГСА автомата Мура можна побудувати таблицю переходів даного автомата. (Таблиця 26.2.)

Таблиця 26.2

Вхідний сигнал / Стан	x_1	$\bar{x}_1\ \bar{x}_2$	$\bar{x}_1 x_2$	$\bar{x}_2$	x_2	$\bar{x}_1\ \bar{x}_3$	$\bar{x}_1 x_3$	x_3	$\bar{x}_3$	1	Вихідний сигнал
a_1	a_2	a_3	a_4								-
a_2				a_3	a_4						y_1
a_3	a_5					a_1	a_6				y_2
a_4	a_5					a_1	a_6				y_3
a_5								a_3	a_1		y_4
a_6										a_1	y_5

Після отримання відміченої ГСА будується граф переходів автомата. Він має стільки різних вершин, скільки різних букв a_i з індексами мається на відміченій ГСА. Кожна вершина графа переходів автомата відзначається буквою а з відповідним індексом.

Між двома вершинами графа проходить дуга, якщо на відміченій ГСА між вершинами з мітками a_i і a_k, проходить шлях. Над дугою ставиться вхідний сигнал, що дорівнює кон'юнкції логічних умов відповідного шляху у відміченій ГСА. При цьому виконанню логічної умови відповідає змінна без заперечення, а невиконанню логічної умови - змінна з запереченням на відповідній дузі графа переходів автомата.

Якщо у відміченій ГСА між згаданими вершинами з мітками a_i і a_k є кілька шляхів, то в графі переходів автомата на дузі, що зв'язує a_i і a_k через символ диз'юнкції перераховуються всі кон'юнкції, що відповідають наявним шляхам.

Якщо будується граф переходів автомата Мура, то символи мікрооперацій (вихідні сигнали управляючого автомата) записуються поряд з відповідними вершин.

Якщо у відміченій ГСА є безумовний перехід між операторними вершинами, тобто шлях, що не проходить ні через які умовні вершини, то на графі переходів автомата йому відповідає дуга, якій приписується вхідний сигнал «1», що показує, що даний перехід в автоматі здійснюється при надходженні чергового синхросигналу.

Надалі синтез проводиться за допомогою описаного раніше методу структурного синтезу. Підкреслимо, що вхідними сигналами синтезованого структурного автомата є кон'юнкції булевих змінних (або диз'юнкції кон'юнкцій), кожна з яких відображає шлях через відповідні умовні вершини відміченої ГСА, а вихідними сигналами - мікрооперації, що позначають або вершини, або дуги графа переходів автомата, залежно від його типу. Використовуючи канонічний метод структурного синтезу, можна побудувати функціональну схему автомата.

26.3 Індивідуальні завдання

Завдання 1.

Закон функціонування операційного автомата задано граф схемою алгоритму. (Варіант №1-№4)

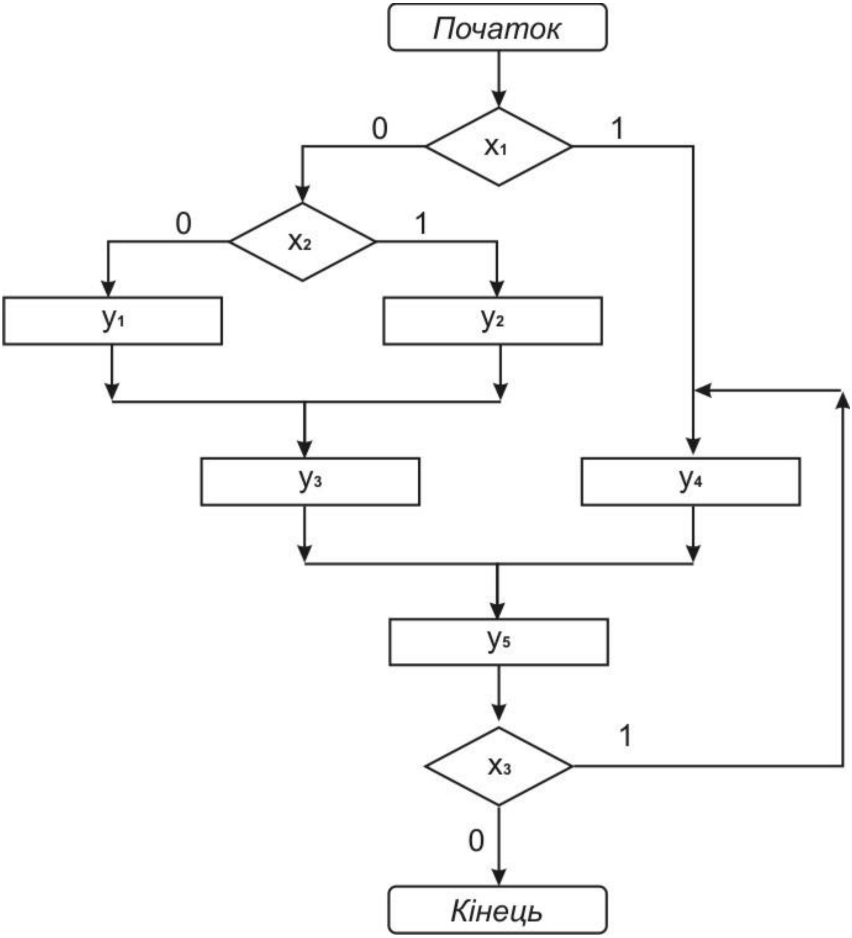
Побудувати таблицю переходів та граф переходів управляючого автомата Мілі.

Завдання 2.

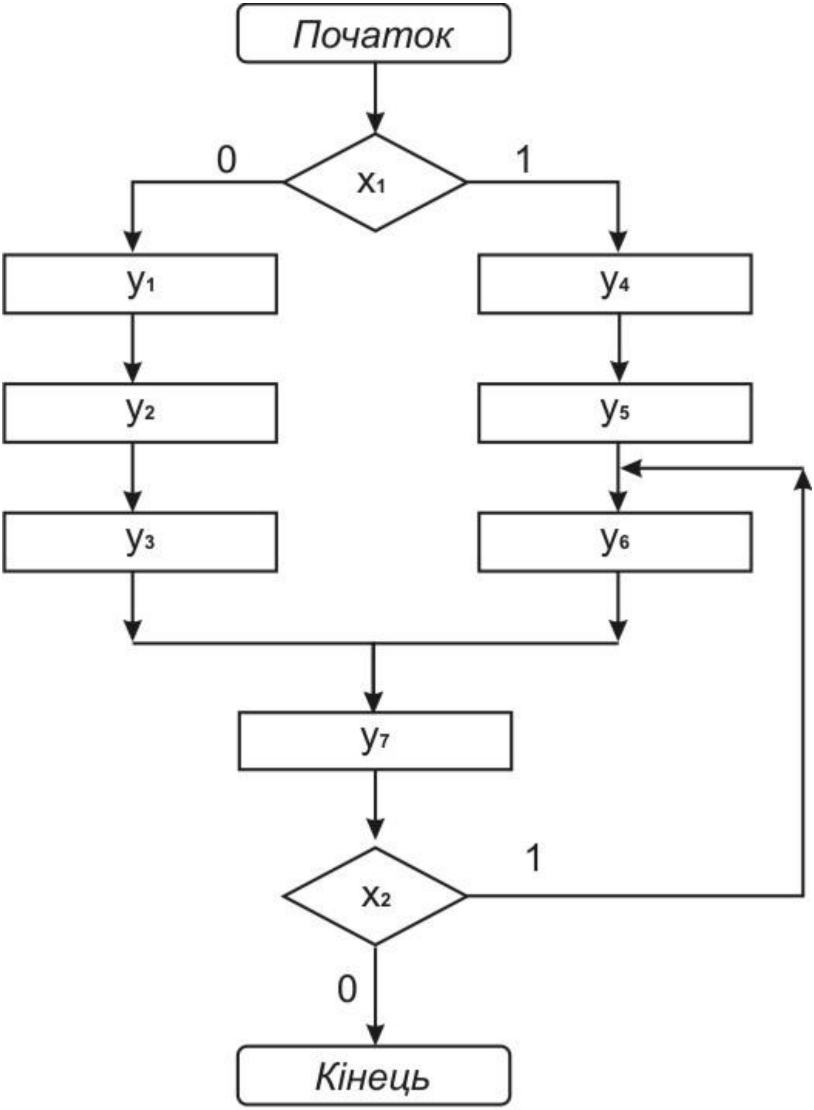
Закон функціонування операційного автомата задано граф схемою алгоритму. (Варіант №1-№4)

Побудувати таблицю переходів та граф переходів управляючого автомата Мура.

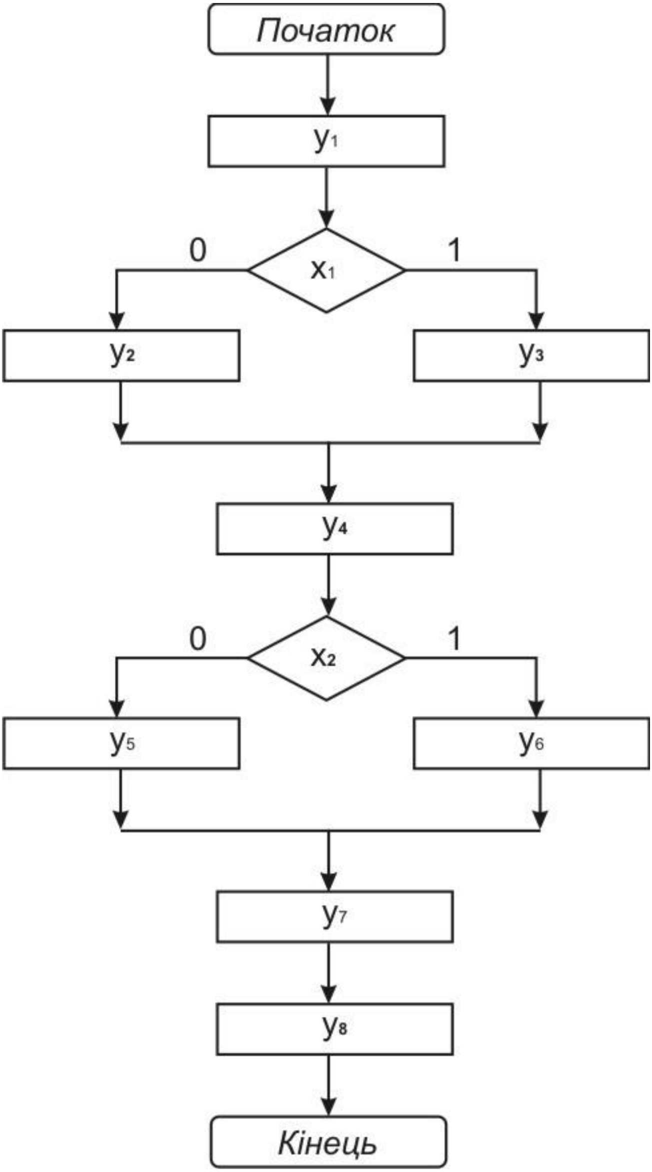
Варіант №1



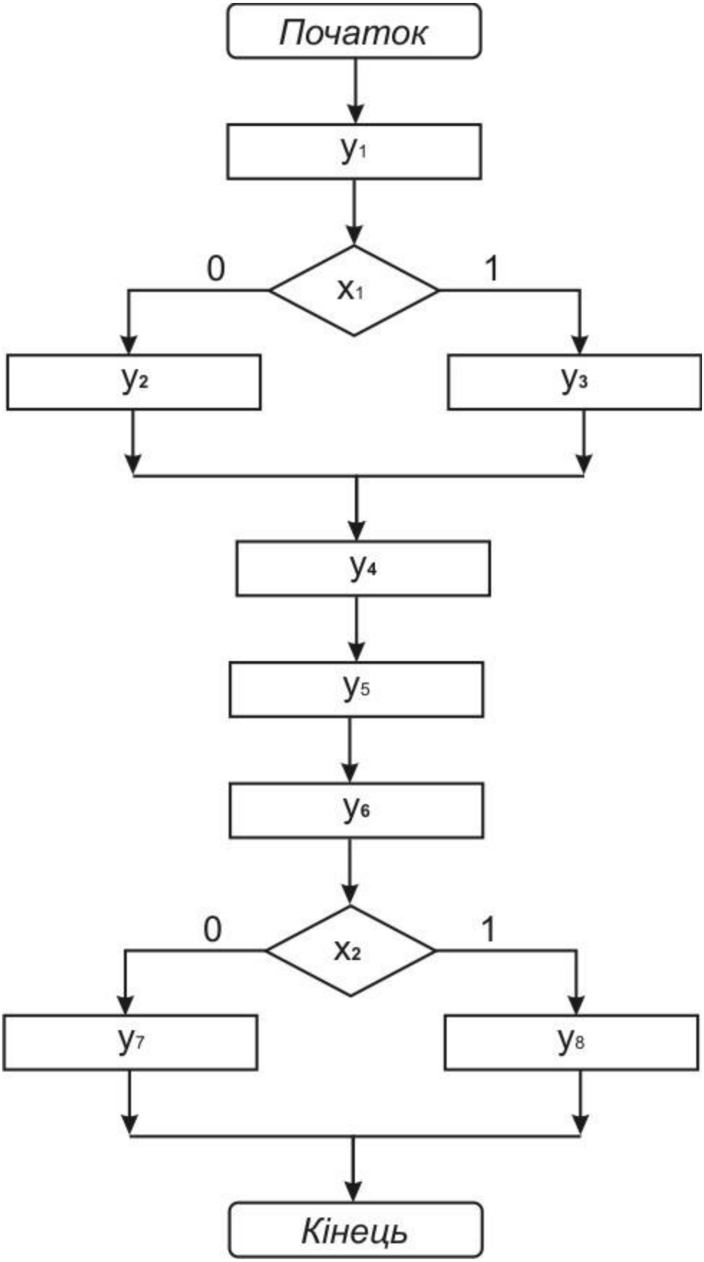
Варіант №2



Варіант №3



Варіант №4



27.4 Контрольні запитання

1. В чому полягає принцип мікропрограмного управління?
2. В чому полягає концепція операційного та управляючого автоматів?
3. Сформулюйте основні етапи проектування дискретних пристроїв на основі жорсткої логіки.
4. Що таке граф схема алгоритму? Правила побудови змістовної ГСА.
5. Сформулюйте основні етапи синтезу управляючого автомата по його змістовній ГСА.
6. Як побудувати відмічену ГСА для автомата Мура?
7. Як побудувати відмічену ГСА для автомата Мілі?

Список рекомендованої літератури

1.Самофалов К.Г., Романкевич А.М., Каневский И.С. Прикладная теория цифровых автоматов. – Киев: Вища школа, 1987.

2. Пєтух А.М., Войтко В.В. Прикладна теорія цифрових автоматів. Навчальний посібник. - Вінниця, ВДТУ, 2001. -77 с.

3. Кочубей О.О. Прикладна теорія цифрових автоматів. Логічні основи: навч. посіб./ О.О.Кочубей, О.В.Сопільник.-Д.: ДНУ; Вид-во ДНУ,2009.-264с.

4. Жабін В.І., Жуков І.А., Клименко І.А., Ткаченко В.В. Прикладна теорія цифрових автоматів: Навч. Посібник.- К.: Книжкове вид-во НАУ, 2007.-364с.

5. Матвієнко М. П. Комп'ютерна логіка. Навчальний посібник. — К.: Видавництво Ліра-К, 2012. — 288 с.

6. Разживін О.В. Комп’ютерна логіка. Навчальний посібник для студентів денної та заочної форм навчання спеціальності 123 "Комп’ютерна інженерія"/ О.В. Разживін, Єнікєєв О.Ф.: Краматорськ: ДДМА, - 2020.-116с.

7. Комп’ютерна логіка. Курсова робота. [Електронний ресурс] : навч. посібн. для здобувачів ступеня бакалавра за освітньою програмою «Комп’ютерні системи та мережі» спеціальності 123 «Комп’ютерна інженерія» / Укладачі: В. І. Жабін, О. А. Верба; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 1,67 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 52 с.

8. Теорія цифрових автоматів та формальних мов. Вступний курс : навч. посібник / Гавриленко С. Ю., Клименко А. М., Любченко Н.Ю. та ін. – Харків : НТУ "ХПІ", 2011. – 176 с

9. Лысиков Б.Г. Арифметические и логические основы цифровых автоматов. - Минск: Вышэйшая школа, 1980.

10. Поснов Н.Н. Арифметика вычислительных машин в упражнениях и задачах: системы счисления, коды. –Мн.: Изд-во ”Университетское”, 1984.

11. Конспект лекцій по дисципліні “Комп’ютерна логіка та кінцеві автомати“, напрям підготовки 6.050102 “ Комп’ютерна інженерія“, /Розр.: О.К. Лифар. – Сєвєродонецьк: Вид-во СНУ ім. В. Даля, 2017. – 239 с.

12. Глушков, В.М. Синтез цифровых автоматов [Текст] / В.М. Глушков. – М.: Государственное издательство физико-математической литературы, 1962.

13. Верьовкін Л.Л. Цифрові логічні автомати. Методичні рекомендації до практичних занять для здобувачів вищої освіти першого бакалаврського рівня за спеціальністю 153 «Мікро- та наносистемна техніка» освітньопрофесійної програми «Мікро- та наносистемна техніка». Запоріжжя : ЗНУ, 2021. 32 с.

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ
до практичних занять
з дисципліни

«Комп'ютерна логіка та цифрові автомати»

для здобувачів першого (бакалаврського) рівня освіти
за спеціальністю
F3 – Комп'ютерні науки
F7 – Комп'ютерна інженерія

(Електронне видання)

Укладач:
Лифар Олена Костянтинівна
Оригінал-макет О.К. Лифар

Підписано до друку__.__.202__.
Формат 60x84 1/16. Папір типогр. Гарнітура Times.
Друк офсетний. Умов. друк. арк. ___. Обл.-вид. арк.___.
Тираж ___ екз. Вид. № ___. Замов. № ____. Ціна договірна.
Видавництво Східноукраїнського національного університету
імені Володимира Даля

Свідоцтво про реєстрацію: серія ДК № 1620 від 18.12.03 р.
Адреса університета: вул. Іоанна Павла ІІ., 17
м. Київ, 01042, Україна
e-mail: vidavnictvosnu.ua@gmail.com