

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

## **МЕТОДИЧНІ ВКАЗІВКИ**

до самостійної роботи

з дисципліни

### **„СХЕМОТЕХНІКА АНАЛОГОВИХ І ЦИФРОВИХ ПРИСТРОЇВ ОБРОБКИ СИГНАЛІВ“**

#### **ЧАСТИНА ДРУГА**

для студентів спеціальностей

175 „Інформаційно-вимірювальні технології“,

освітня програма: „Інформаційні системи моніторингу і контролю“;

176 „Мікро- та наносистемна техніка“,

освітня програма: „Мікро- та наноелектронні прилади і пристрої“

першого (бакалаврського) рівня вищої освіти

денної й заочної форм навчання

Методичні вказівки до самостійної роботи з дисципліни „Схемотехніка аналогових і цифрових пристроїв обробки сигналів“. Частина друга. Для студентів спеціальностей: 175 „Інформаційно-вимірювальні технології“, освітня програма: „Інформаційні системи моніторингу і контролю“; 176 „Мікро- та наносистемна техніка“, освітня програма: „Мікро- та наноелектронні прилади і пристрої“ першого (бакалаврського) рівня вищої освіти денної й заочної форм навчання / Укл.: Ніна НАГОРНА. – Запоріжжя: НУ «Запорізька політехніка», 2025. – 41 с.

Укладач: Ніна НАГОРНА, ст. викладач,  
Рецензент: Валентин ПОГОСОВ, проф., д-р фіз.-мат. наук  
Відповідальний за випуск: Андрій КОРОТУН, канд. фіз.-мат. наук,  
професор

Затверджено  
на засіданні кафедри  
інформаційної безпеки та  
наноелектроніки

Протокол № 5  
від “22” січня 2025 р.

Рекомендовано до видання  
НМК ФІБЕК  
Протокол № 7  
від “24” лютого 2025 р.

## ЗМІСТ

### Частина друга

7 Самостійна робота №5 „Синтез схем на основі мультиплексорів”.....	37
7.1 Теоретичні відомості.....	37
7.2 Завдання.....	41
7.3 Порядок оформлення звіту.....	43
7.4 Контрольні запитання.....	44
8 Самостійна робота №6 „Синтез комбінаційних вузлів на основі шифраторів і дешифраторів”.....	45
8.1 Теоретичні відомості.....	45
8.2 Методика синтезу логічної структури перетворювача десятикового коду в заданий код.....	53
8.3 Завдання.....	56
8.4 Порядок оформлення звіту.....	59
8.5 Контрольні запитання.....	59
9 Самостійна робота №7 „Синтез регістрів”.....	60
9.1 Теоретичні відомості.....	60
9.2 Завдання.....	63
9.3 Порядок оформлення звіту.....	64
9.4 Контрольні запитання.....	64
10 Самостійна робота №8 „Синтез лічильників”.....	65
10.1 Теоретичні відомості.....	65
10.2 Завдання.....	69
10.3 Порядок оформлення звіту.....	69
10.4 Контрольні запитання.....	69
11 Перелік запитань, що виносяться на модульний контроль.....	71
11.1 Перший модульний контроль (7 семестр).....	71
11.2 Другий модульний контроль (7 семестр).....	72
Рекомендована література.....	74

## 7 САМОСТІЙНА РОБОТА №5 „СИНТЕЗ СХЕМ НА ОСНОВІ МУЛЬТИПЛЕКСОРІВ”

**Мета роботи** - дослідження структури і принципів дії мультиплексора у якості багатофункціонального вузла; ознайомлення з прийомами синтезу логічних схем на основі мультиплексорів.

### 7.1 Теоретичні відомості

**Мультиплексор** – це функціональний вузол, що здійснює операцію передачі сигналу з одного із інформаційних входів на один вихід.

Вибір того чи іншого інформаційного входу здійснюється в залежності від коду, що надходить на адресні входи мультиплексора. При наявності у мультиплексора  $m$  адресних входів можна реалізувати  $n = 2^m$  комбінацій адресних сигналів, кожна з яких забезпечує вибір визначеного інформаційного входу з  $n$  входів.

Окрім інформаційних входів у мультиплексора є *керуючі* входи. До них відносяться *адресні входи* і *вхід дозволу* (стробуючий вхід).

Вихід у мультиплексора один, але в деяких випадках передбачені два виходи для парафазного представлення вихідного сигналу.

Мультиплексори використовують в лініях зв'язку для комутації каналів зв'язку. Тому їх відносять до комутаторів, які здійснюють передачу інформації, що надходить по одному з вхідних каналів на один вихідний канал зв'язку. Об'єднанням декількох цифрових потоків у єдиний забезпечується стиск інформаційних сигналів.

В пристроях автоматики мультиплексор застосовують для послідовного чи адресного опитування заданого числа джерел інформаційних сигналів і передачі цих сигналів на один вихід.

За допомогою мультиплексора виконують також перетворення рівнобіжних (паралельних) цифрових кодів у послідовні.

В інтегральному виконанні випускаються мультиплексори з двома інформаційними входами (чотири елементи в одному корпусі), з чотирма інформаційними входами (два елементи в корпусі), з 8 і 16 інформаційними входами. Умовно-графічне позначення деяких з них показано на рис. 7.1.

В інтегрованих схемах мультиплексорів з виходом на три стани при відсутності сигналу дозволу на стробуючому вході вихід

переходить у стан «відключений» (високий імпеданс), тому такі виходи можна поєднувати за принципом монтажно́ї логіки.

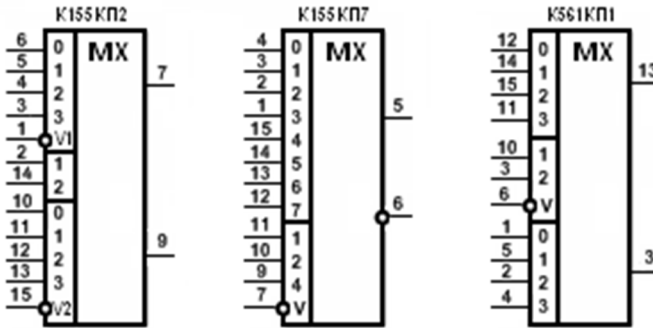


Рисунок 7.1 – Умовно-графічне позначення мультиплексорів

Швидкодія мультиплексорів залежить насамперед від типу комутуючих елементів та їхнього режиму роботи. Найбільшу швидкодію мають мультиплексори серій K1500, K500, 100, найменшу - мультиплексори малопотужних серій TTL, наприклад, K134, і серій ІС типу КМДН: K561, 564. При оцінці часових параметрів варто мати на увазі, що затримка поширення сигналу до виходу від інформаційного, керуючого і стробуючого входів різна. Наприклад, у восьмиканального мультиплексора K155KP7 значення цих параметрів дорівнюють 17, 27, 41 нс відповідно.

Вхід дозволу  $V$  мультиплексора використовується для стробування і нарощування числа входів мультиплексора. На рис. 7.2 показана схема мультиплексора 16:1, яка побудована на двох мультиплексорах K155KP7.

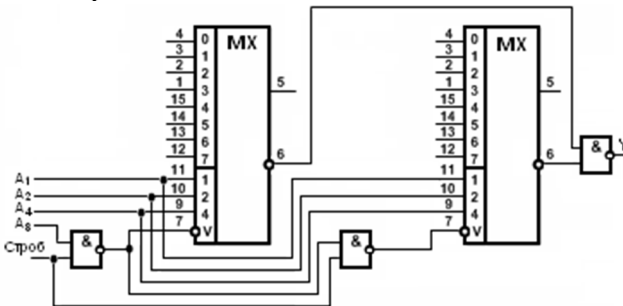


Рисунок 7.2 – Схема мультиплексора 16:1

Якщо кількість інформаційних входів у мультиплектора велика, то його схему можна одержати за допомогою каскадування більш простих мультиплексорів. На рис. 7.3 показана схема мультиплектора 16:1, що отримана за допомогою каскадування.

Мультиплексор може бути використаний у якості універсального логічного елемента, за допомогою якого можна реалізувати будь-яку логічну функцію, що має не більш, ніж  $m+1$  змінну, де  $m$  - кількість адресних входів мультиплектора.

Використання мультиплектора як універсального логічного елемента засноване на загальній властивості логічних функцій незалежно від кількості аргументів завжди приймати значення "1" чи "0".

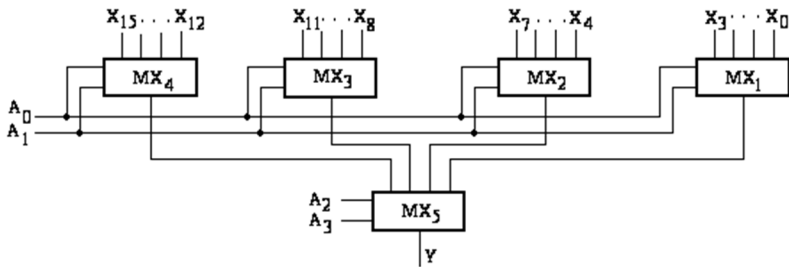


Рисунок 7.3 – Пірамідальне нарощування мультиплектора

Якщо на адресні входи мультиплектора послідовно подавати значення вхідних змінних відповідно наявним наборам таблиці істинності функції, а на відповідні інформаційні входи - значення функції, то в такий спосіб синтезується пристрій, що реалізує потрібну логічну функцію. Наприклад, для реалізації функції "2I" (табл. 7.1) із двома змінними, необхідно скористатися мультиплексором "4:1", у якого до адресних входів  $A_2$  і  $A_1$  потрібно підключити сигнальні лінії  $X_1$  і  $X_0$  відповідно, а на інформаційні входи подавати сигнали відповідно до табл. 2.1:  $DI_0 = DI_1 = DI_2 = "0"$ ,  $DI_3 = "1"$ .

Якщо кількість аргументів функції дорівнює  $m+1$ , то методика синтезу схеми буде іншою. Припустимо, що на основі мультиплектора "4:1" необхідно реалізувати функцію "3I", що має три змінні.

Таблиця істинності функції "3I" (табл. 7.2) розбивається на групи по два рядки в кожній групі. В обох рядках кожної з чотирьох груп змінні  $X_2$  і  $X_1$  мають однакові значення, а  $X_0$  змінюється.

Таблиця 7.1 - Таблиця істинності функції "2Г"

X1	X0	F
0	0	0
0	1	0
1	0	0
1	1	1

Тому зі значень  $X2$  і  $X1$  кожної групи можна утворити набори 00, 01, 10, 11, підключаючи сигнал  $X2$  до адресного входу  $A2$  мультиплексора, а сигнал  $X1$  - до адресного входу  $A1$ . На відповідні наборам інформаційні входи потрібно подавати "1", якщо в двох рядках групи функція приймає значення "1", якщо ж функція в групі має значення "0", - то подається "0". При наявності різних значень функції в групі в загальному випадку можуть бути два варіанти:  $F = X0$ , або  $F = \overline{X0}$ . Отже, для реалізації функції "3Г" потрібно подати сигнали:  $D0 = "0"$ ,  $D1 = "0"$ ,  $D2 = "0"$ ,  $D3 = "X0"$ ,  $A2 = X2$ ,  $A1 = X1$ .

Таблиця 7.2 - Таблиця істинності функції "3Г"

X2	X1	X0	F	Примітка
0	0	0	0	$F=D0=0$
0	0	1	0	
0	1	0	0	$F=D1=0$
0	1	1	0	
1	0	0	0	$F=D2=0$
1	0	1	0	
1	1	0	0	$F=D3=X0$
1	1	1	1	

Схемна реалізація логічної функції "3Г" на основі мультиплексора 4:1 показана на рис. 7.4.

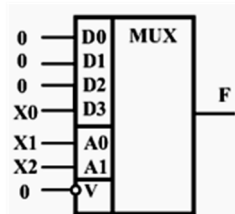


Рисунок 7.4 - Схемна реалізація логічної функції "3Г"

## 7.2 Завдання

**7.2.1** Перетворіть паралельний двійковий код у послідовний код за допомогою мультиплексора. Зобразіть часову діаграму роботи перетворювача. Вихідні дані приведені у табл. 7.3.

*Вказівка:*

- число інформаційних входів у мультиплексора повинно бути більше чи дорівнювати числу розрядів коду;
- на інформаційні входи слід подавати паралельний двійковий код, а сигнали на адресних входах необхідно змінювати циклічно в такій двійковій послідовності, щоб молодший розряд послідовного коду на виході мультиплексора сформувався раніше інших розрядів.

**7.2.2** Побудуйте мультиплексор 32:1 з використанням каскадного з'єднання чотирьох мультиплексорів 8:1 та одного мультиплексора 4:1.

*Вказівка:* зобразьте логічну структуру мультиплексора 32:1 у вигляді з'єднаних між собою мультиплексорів 8:1 та 4:1, що представлені у вигляді умовно-графічних позначень.

**7.2.3** Реалізуйте задану логічну функцію за допомогою мультиплексора. Логічна функція, що задається, приведена у табл. 7.3.

*Вказівка:*

- реалізація функції за допомогою мультиплексора виконується з використанням таблиці істинності функції;
- число адресних входів мультиплексора повинно бути на одиницю менше за число змінних заданої функції.

Таблиця 7.3 - Варіанти завдань

№ варіанта	Завдання 7.2.1	Завдання 7.2.3
	Паралельний код	Логічна функція $F$
1	1100	$\overline{(x_1 \bar{x}_2 + x_2 \bar{x}_3)}(x_1 + x_2)$
2	0110	$x_1 x_2 \oplus \bar{x}_3$
3	1010	$(x_1 + x_2 + \bar{x}_3)(\bar{x}_1 + x_3)(x_2 + \bar{x}_3)$
4	1100	$\overline{x_2 \bar{x}_2 + \bar{x}_1 \bar{x}_3 + \bar{x}_2 \bar{x}_3}$

## Продовження таблиці 7.3

№ вар.	Завдання 7.2.1	Завдання 8.2.3
	Паралельний код	Логічна функція $F$
5	0011	$\overline{\overline{x_1 \overline{x_3}} + \overline{x_2 \overline{x_3}} + x_1 x_2 x_3}$
6	0101	$\overline{\overline{\overline{x_1 \overline{x_2 \overline{x_3}} + x_1 x_2 x_3} + \overline{\overline{x_1} x_2}}$
7	1001	$\overline{\overline{x_1 \overline{x_2 \overline{x_3}} + \overline{x_1} \overline{x_2} x_3}$
8	1101	$\overline{\overline{x_1 x_2 x_3} \oplus \overline{x_1 \overline{x_2} x_3}}$
9	1011	$\overline{\overline{x_1 x_2 x_3} \oplus \overline{x_1 \overline{x_2} \overline{x_3}}}$
10	1110	$\overline{(x_1 \oplus x_2) \oplus \overline{x_3}}$
11	1000	$\overline{(x_1 \overline{x_2} + x_1 \overline{x_3}) \oplus \overline{\overline{x_2 \overline{x_3}}}}$
12	0100	$x_1 x_2 \oplus \overline{x_2} x_3$
13	0111	$\overline{x_1 \overline{x_2}} \oplus x_3$
14	0010	$\overline{\overline{x_1 \overline{x_2} + x_2 x_3}}$
15	0001	$\overline{x_1} \oplus x_2 \overline{x_3}$
16	1010	$\overline{\overline{\overline{x_1 \overline{x_2 \overline{x_3}} \oplus x_1 x_2 x_3} + \overline{\overline{x_1} x_2}}$
17	0110	$\overline{x_1 x_2 \oplus x_3 + \overline{x_1} x_2}$
18	1101	$\overline{\overline{x_1 \overline{x_3}} + \overline{x_2 \overline{x_3}} \oplus \overline{x_1 x_2 x_3}}$
19	1110	$\overline{\overline{x_1} + (x_2 \oplus \overline{x_3}) + \overline{x_2} x_3}$
20	1011	$\overline{(x_1 + \overline{x_2} \oplus x_3) + x_2 x_3}$
21	1100	$\overline{\overline{x_1 x_2} \oplus x_3 + \overline{x_1} x_2}$
22	0110	$\overline{\overline{x_1 x_2} \oplus \overline{x_2} x_3}$
23	1010	$\overline{\overline{x_1} x_2 \oplus x_3 \oplus \overline{x_1 \overline{x_2} x_3}}$
24	1100	$\overline{x_2 \overline{x_3} \oplus \overline{\overline{x_1 \overline{x_3}} + \overline{x_2 \overline{x_3}}}$

## Продовження таблиці 7.3

№ вар.	Завдання 7.2.1	Завдання 7.2.3
	Паралельний код	Логічна функція $F$
25	0011	$x_1x_2 \oplus \bar{x}_2x_3$
26	0101	$\overline{(x_1\bar{x}_2x_2\bar{x}_3)(x_1+x_2)}$
27	1001	$\overline{x_1\bar{x}_2} \oplus \overline{x_2x_3}$
28	1101	$\overline{(x_1\bar{x}_2 \oplus x_1\bar{x}_3) + \bar{x}_2\bar{x}_3}$
29	1011	$(x_1x_2 + \bar{x}_3) + \overline{\bar{x}_1x_3} + x_2\bar{x}_3$
30	1110	$\overline{(x_1+x_2) \oplus \bar{x}_3}$
31	1000	$\bar{x}_1\bar{x}_2\bar{x}_3 \oplus \overline{\bar{x}_1\bar{x}_2x_3}$
32	0100	$\overline{x_1+x_2x_3} \oplus \overline{x_1\bar{x}_2\bar{x}_3}$
33	0111	$\overline{x_1x_2} \oplus \bar{x}_2x_3$
34	0010	$\overline{x_2\bar{x}_2} + \overline{\bar{x}_1\bar{x}_3} + \overline{\bar{x}_2\bar{x}_3}$
35	0001	$\overline{\bar{x}_1\bar{x}_3} \oplus \overline{\bar{x}_2\bar{x}_3} \oplus \overline{x_1x_2x_3}$
36	1010	$\overline{x_1\bar{x}_2} + \overline{x_2x_3}$
37	0110	$\overline{(x_1+x_2 \oplus x_3)} + x_2x_3$
38	1101	$x_1x_2 \oplus \bar{x}_2x_3 + \overline{x_1x_3}$
39	1110	$\overline{x_1x_2} \oplus \overline{x_3} + \overline{\bar{x}_1x_2}$
40	1011	$\overline{x_1x_2x_3} + \overline{\bar{x}_1\bar{x}_2x_3}$

**7.3 Порядок оформлення звіту**

Звіт повинен містити:

- мету роботи;

- схему перетворювача паралельного двійкового коду у послідовний код на основі мультиплектора (пункт 7.2.1);
- часову діаграму роботи перетворювача;
- пірамідальну структурну схему мультиплектора 32:1 (пункт 7.2.2);
- схемну реалізацію заданої логічної функції на основі мультиплектора (пункт 7.2.3).

#### **7.4 Контрольні запитання**

- 1 Який функціональний вузол називається мультиплексором?
- 2 Яка залежність існує між кількістю інформаційних та адресних входів мультиплектора?
- 3 Які входи у мультиплектора є керуючими?
- 4 Яку функцію виконує у мультиплектора вхід дозволу?
- 5 Чому мультиплексор називається також комутатором?
- 6 Яким чином за допомогою мультиплектора паралельний код перетворюється у послідовний?
- 7 Для чого використовуються мультиплексори у пристроях автоматики?
- 8 Чи може мати мультиплексор два виходи?
- 9 Як працює схема мультиплектора 16:1, що показана на рис. 8.2?
- 10 Яку методику синтезу схеми на основі мультиплектора необхідно використати для реалізації довільної логічної функції, якщо число її аргументів співпадає з числом інформаційних входів мультиплектора?
- 11 Яку методику синтезу схеми на основі мультиплектора необхідно використати для реалізації довільної логічної функції, якщо число її аргументів на одиницю більше числа інформаційних входів мультиплектора?
- 12 Розробіть пірамідальну схему мультиплектора 64:1.
- 13 Яке умовне графічне позначення має мультиплексор 32:1?

## 8 САМОСТІЙНА РОБОТА №6

### „СИНТЕЗ КОМБІНАЦІЙНИХ ВУЗЛІВ НА ОСНОВІ ШИФРАТОРІВ І ДЕШИФРАТОРІВ”

#### Мета роботи:

- вивчення правила перекладу диз'юнктивної нормальної форми (ДНФ) у досконалу диз'юнктивну нормальну форму (ДДНФ);
- ознайомлення з системами кодування десяткових чисел;
- розгляд принципу функціонування і логічної структури шифратора;
- вивчення прийомів синтезу схем на функціонально-логічному рівні на основі дешифраторів та шифраторів;
- ознайомлення з методикою синтезу логічної структури перетворювача десяткового коду у заданий код.

#### 8.1 Теоретичні відомості

##### 8.1.1 Перетворення диз'юнктивної нормальної форми у досконалу диз'юнктивну нормальну форму

При синтезі цифрових вузлів, реалізованих на основі типових комбінаційних схем, у багатьох випадках необхідно виконувати перетворення заданих логічних рівнянь, представлених у нормальній формі, в досконалу нормальну форму.

Досконала нормальна форма (ДНФ) відрізняється від нормальної форми (НФ) тим, що містить терми максимального рангу і дає однозначне представлення функції.

Правило перетворення диз'юнктивної нормальної форми (ДНФ) у досконалу диз'юнктивну нормальну форму (ДДНФ) формулюється наступним чином.

Якщо ранг терму, що входить у ДНФ, менше рангу функції, то необхідно виконати логічне множення розглянутого терму на вираз  $(x_i + \bar{x}_i)$  стільки разів, скільки змінних не входить у розглянутий терм.

Нехай  $f_{\text{ДНФ}} = F_1$ ;  $x_i$  – змінна, що не входить у терм  $F_1$ . Тоді

$$f_{\text{ДДНФ}} = F_1 \cdot (x_i + \bar{x}_i), \quad (8.1)$$

Тобто, якщо максимальний ранг функції дорівнює  $r$ , а мінімальний ранг  $j$ -го терму дорівнює  $k$ , то перетворення (8.1) необхідно застосувати до  $j$ -го терму  $(r-k)$  разів.

Наприклад, при перетворенні функції  $F = \bar{x}_1x_3 + \bar{x}_2x_3 + x_1\bar{x}_2\bar{x}_3$  з ДНФ у ДДНФ необхідно виконати дії у послідовності:

$$f_1 = x_1x_3 = x_1x_3(x_2 + \bar{x}_2) = x_1x_2x_3 + x_1\bar{x}_2x_3;$$

$$f_2 = \bar{x}_2x_3 = \bar{x}_2x_3(x_1 + \bar{x}_1) = x_1\bar{x}_2x_3 + \bar{x}_1\bar{x}_2x_3;$$

$$F = f_1 + f_2 + x_1\bar{x}_2\bar{x}_3 = x_1x_2x_3 + x_1\bar{x}_2x_3 + \bar{x}_1\bar{x}_2x_3 + x_1\bar{x}_2\bar{x}_3.$$

### 8.1.2 Шифрування і дешифрування інформації

Шифрування і дешифрування (стискання даних і зворотне перетворення) є основними видами перетворення інформації. Дешифратори перетворюють двійковий код у код "1 з  $N$ ", а шифратори, навпаки, - код "1 з  $N$ " у двійковий код.

Розрізняють повні і неповні дешифратори. Число виходів повного дешифратора  $N_{\text{вих}}=2^m$ , неповного -  $N_{\text{вих}} < 2^m$ , де  $m$  – число адресних входів дешифратора.

Для неповних дешифраторів існують вхідні набори, при яких функція є невизначеною і які можна використовувати при мінімізації вихідних функцій.

### 8.1.3 Призначення і принцип роботи шифратора

**Шифратор** – це комбінаційний функціональний вузол, що перетворює унітарний код ("1 основою  $N$ ") у двійковий код, тобто при подачі сигналу на один з входів шифратора на його виходах формується паралельний двійковий код.

Повний шифратор має  $2^n$  входів і  $n$  виходів, де  $n$  – число розрядів двійкового коду.

Одне з основних призначень шифратора – введення даних з клавіатури. Наприклад, неповний шифратор "10:4", вбудований в клавіатуру, дозволяє при натисненні будь-якої десяткової цифри на клавіатурі сформувати її двійковий код.

У таблиці 8.1 наведено таблицю істинності такого шифратора.

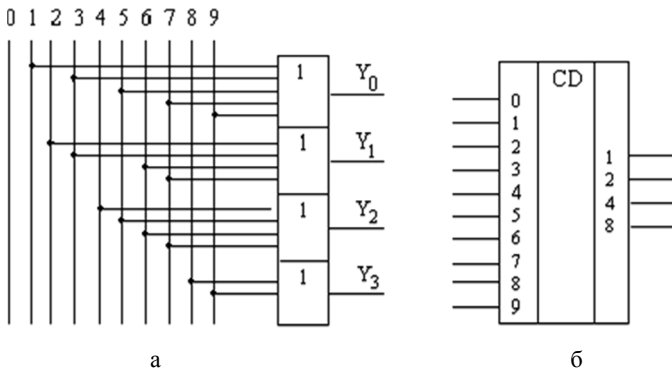
Таблиця 8.1 – Таблиця істинності шифратора "10:4"

Десяткове число	Входи										Виходи			
	$X_9$	$X_8$	$X_7$	$X_6$	$X_5$	$X_4$	$X_3$	$X_2$	$X_1$	$X_0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
2	0	0	0	0	0	0	0	1	0	0	0	0	1	0
3	0	0	0	0	0	0	1	0	0	0	0	0	1	1
4	0	0	0	0	0	1	0	0	0	0	0	1	0	0
5	0	0	0	0	1	0	0	0	0	0	0	1	0	1
6	0	0	0	1	0	0	0	0	0	0	0	1	1	0
7	0	0	1	0	0	0	0	0	0	0	0	1	1	1
8	0	1	0	0	0	0	0	0	0	0	1	0	0	0
9	1	0	0	0	0	0	0	0	0	0	1	0	0	1

Логічні рівняння, що описують роботу шифратора "10:4", будуються на основі таблиці істинності:

$$\begin{aligned}
 Y_0 &= X_1 + X_3 + X_5 + X_7 + X_9, \\
 Y_1 &= X_2 + X_3 + X_6 + X_7, \\
 Y_2 &= X_4 + X_5 + X_6 + X_7, \\
 Y_3 &= X_8 + X_9.
 \end{aligned}$$

Логічна схема шифратора "10:4" та її умовне позначення приведені на рис. 8.1.



а – логічна структура;  
б - умовне графічне позначення

Рисунок 8.1 – Логічна структура та умовне графічне позначення шифратора

### 8.1.4 Реалізація логічних функцій на основі дешифраторів

Нижче представлена ілюстрація синтезу логічних схем на основі повного дешифратора.

Нехай необхідно синтезувати логічну схему на основі дешифратора, яка реалізує логічну функцію виду

$$Y = x_1 \bar{x}_2 \vee x_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3 .$$

Рішення. Вихідна функція, що задана в ДНФ, перетворюється в ДДНФ:

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

Таблиця істинності отриманої функції (табл. 8.2) представлена нижче.

Таблиця 8.2 – ТІ функції  $Y$

$x_1$	$x_2$	$x_3$	$Y$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

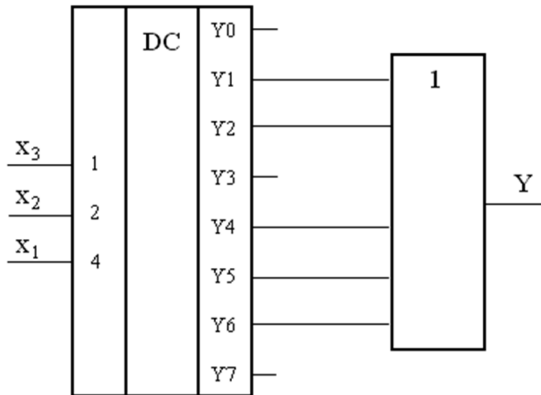
Таблиця істинності дешифратора, на основі якого буде побудована схема, приведена нижче (табл. 8.3). Кількість входів дешифратора відповідає кількості змінних заданої функції.

У таблиці 8.3 підкреслені набори змінних, на яких задана логічна функція приймає одиничні значення, а також підкреслені одиничні значення виходів дешифратора, що відповідають виділеним наборам.

На основі таблиць 8.2 і 8.3 можна синтезувати схему, що реалізує задану функцію (рис. 8.2).

Таблиця 8.3 – ТІ дешифратора ”3:8”

Входи			Виходи							
$x_1$	$x_2$	$x_3$	$y_0$	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$	$y_7$
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Рисунок 8.2 – Реалізація функції  $Y$  на основі дешифратора

### 8.1.5 Кодування чисел

**8.1.5.1 Кодування десяткових чисел. Вагові коди.** Поряд із загальною системою кодування алфавітно-цифрових символів у цифрових пристроях використовується також окрема система кодування тільки десяткових цифр.

Десяткові цифри 0, 1, ..., 9 кодуються двійковими цифрами за допомогою різних кодів, наприклад, за допомогою **двійково-десяткового коду (ДДК)**.

За допомогою чотирьох двійкових цифр можна утворити 16 різних комбінацій, а використовується 10, тому двійково-десятковий код володіє деякою надмірністю.

У двійково-десятковому кодї 8421 кожна цифра десяткового числа зображується відповідним чотирирозрядним двійковим числом. Це найбільш розповсюджений код.

Однак використання цього коду зв'язано з труднощами, що виникають при виявленні переносу в наступний десятковий розряд, а також зі складністю переходу до зворотних і додаткових кодів для десяткових чисел.

**Вагові коди.** На практиці використовуються такі коди, які забезпечують раціональну і просту обробку цифрової інформації. Найчастіше використовуються так звані вагові коди. *Ваговим двійково-десятковим кодом* називається такий код, у якого вага кожного розряду залишається незмінною для всіх 10 комбінацій.

Будується він з урахуванням таких умов:

- вага найменшого розряду дорівнює 1;
- вага другого за мінімальним значенням розряду може дорівнювати 1 або 2;
- вага, що відповідає двом останнім розрядам коду, підбирається так, щоб їх сума була більшою або дорівнювала 6.

Згідно з цими вимогами можна отримати 17 варіантів ДДК з додатною вагою: 8421, 7421, 6421, 5421, 4421, 7321, 6321, 5321, 4321, 3321, 6221, 5221, 4221, 6311, 5311, 4311, 5211.

Окрім коду 8421, решта 16 кодів не мають однозначності в зображенні десяткових чисел. Так, код 3321 дає змогу записати цифру 5 у двійковій формі як 1010 або 0110.

В інших ДДК, що не відповідають переліченим вище умовам, вага окремих розрядів може бути додатною або від'ємною.

Вагові коди дозволяють достатньо просто переводити десяткові цифри  $d$  в двійкові цифри  $b$  за формулою

$$d = v_3 b_3 + v_2 b_2 + v_1 b_1 + v_0 b_0, \quad (8.2)$$

де цифри  $v_3, \dots, v_0$  є ваговими коефіцієнтами відповідного коду, а символи  $b_3, \dots, b_0$  - двійкові цифри 0 і 1.

Наприклад, в кодi 7421  $v_3 = 7; v_2 = 4; v_1 = 2; v_0 = 1$ . Цифри окремих кодiв легко визначаються за допомогою співвiдношення (8.2):

$$(9)_{10} = 5 \cdot 1 + 4 \cdot 1 + 2 \cdot 0 + 1 \cdot 0 = (1100)_{5421},$$

$$(6)_{10} = 4 \cdot 1 + 3 \cdot 0 + 1 \cdot 1 + 1 \cdot 1 = (1011)_{4311},$$

$$(7)_{10} = 6 \cdot 1 + 4 \cdot 1 + (-2) \cdot 1 + (-1) \cdot 1 = (1111)_{64-2-1}.$$

**8.1.5.2 Складенi та рефлекснi коди.** Складенi коди базуються на складених системах, що мають двi i бiльше основ. При такому кодуваннi числа, заданi в системi з деякою основою  $q$ , зображаються за допомогою цифр iншої системи з основою  $p < q$ .

Серед складених кодiв найбільш розповсюдженi двiйково-десятковi коди, якi зазвичай використовуються як промiжнi при переводi десяткових кодiв в двiйковi i навпаки.

Використання найвiдомiшого двiйково-десяткового коду 8421 пов'язане з труднощами, що виникають при виявленнi переносу в наступний десятковий розряд, а також зi складнiстю переходу до зворотних i додаткових кодiв для десяткових чисел. Крім того, як впливає з таблицi 8.4, при переходi вiд зображення числа 3 до зображення числа 4 вiдбувається одночасна змiна цифр в трьох розрядах, а при переходi вiд зображення числа 7 до зображення числа 8 вiдбувається одночасна змiна цифр в чотирьох розрядах. Це може бути джерелом значних помилок при деяких способах кодування безперервних повiдомлень. Наприклад, при перетвореннi кута повороту вала в двiйковий код. Особливiстю таких перетворювачiв є те, що пiд час переходу зображення одного числа до зображення наступного числа (бiльшого або меншого) може мати місце неоднозначнiсть вiдлiку в тих розрядах, де вiдбувається змiна цифр. Наприклад, при переходi вiд зображення числа 7 до зображення числа 8 вiдбувається змiна цифр у всiх розрядах, тобто може мати місце неоднозначнiсть вiдлiку в будь-якому розрядi i навлiсть вiдразу у всiх розрядах. Внаслiдок цього, зокрема, може бути такий випадок, коли замість числа 1000 буде зафіксовано число 1111, тобто матиме місце велика похибка.

Відповідність десяткових чисел і кодів 8421 та Грея показана у табл. 8.4.

Таблиця 8.4 – Відповідність десяткових чисел і кодів 8421 та Грея

Десяткове число	0	1	2	3	4	5	6	7
Код 8421	0000	0001	0010	0011	0100	0101	0110	0111
Код Грея	0000	0001	0011	0010	0110	0111	0101	0100

Для усунення цього явища використовуються спеціальні двійкові коди, основою яких при переході від зображення одного числа до зображення наступного сусіднього числа змінюється значення цифри тільки одного розряду. Внаслідок цього похибка за рахунок неоднозначності прочитування не перевищує одиниці числа, яке відображається. До числа таких кодів відноситься **код Грея**, який називається також рефлексним (віддзеркаленим) кодом.

Код Грея є непозиційним кодом, в ньому вага одиниці не визначається номером розряду. У цьому коді можна виділити осі симетрії (осі “віддзеркалення”), відносно яких спостерігається ідентичність елементів в деяких розрядах. Так, наприклад, має місце симетрія відносно осі, проведеної між числами 7 і 8. У комбінаціях, симетричних відносно цієї осі, ідентичні три символи молодших розрядів. Відмічена особливість послужила основою для введення терміну “рефлексний код” (від англійського слова to reflect – віддзеркалювати).

Вага одиниці в коді Грея за абсолютною величиною у  $j$ -му розряді визначається виразом  $\sum_{i=0}^j 2^i$ , причому знак членів, що підсумовуються, додатний для всіх непарних цифр в числі, записаному в коді Грея (зчитувати зліва направо), і від’ємний для всіх парних одиниць. Наприклад, значення числа 1101101, записаного в коді Грея, буде

$$[1101101]_Г = \sum_{i=0}^6 2^i - \sum_{i=0}^5 2^i - \sum_{i=0}^3 2^i + \sum_{i=0}^2 2^i + \sum_{i=0}^0 2^i = 2^6 - 2^3 + 2^0 = 57.$$

## 8.2 Методика синтезу логічної структури перетворювача десяткового коду в заданий код

Синтез перетворювача десяткового коду в заданий код передбачає побудову структурної схеми пристрою, тобто визначення складу необхідних логічних елементів і з'єднань між ними, при яких забезпечується перетворення вхідних цифрових сигналів у вихідні відповідно до заданих умов роботи пристрою. В процесі синтезу необхідно мінімізувати апаратні затрати на реалізацію пристрою.

Методика синтезу логічної структури перетворювача складається з наступних етапів.

Етап 1. Запис умов функціонування перетворювача. Умови можуть бути задані словесно, за допомогою таблиць істинності або логічних рівнянь.

Етап 2. Запис і мінімізація логічних рівнянь. Якщо умови на етапі 1 задані словесно, то на їх основі заздалегідь складається таблиця істинності. Якщо логічні вирази вже визначені на етапі 1, то виконується їх мінімізація. В процесі мінімізації використовуються перетворення за допомогою аксіом і законів булевої алгебри, а також використовуються алгебраїчні і графічні методи.

Етап 3. Запис мінімізованих логічних рівнянь в заданому базисі.

Етап 4. Складання структурної схеми.

Нижче приведений синтез перетворювача десяткового коду в код Грея на основі запропонованої методики. Перетворювач синтезується на базі шифратора з високим активним рівнем.

Етап 1. Перетворювач десяткового коду в код Грея синтезується на основі шифратора. На виході перетворювача формується контрольний сигнал, підтверджуючий про виконання дій по перетворенню кодів.

Етап 2. Складається таблиця істинності перетворювача (табл.8.5), для чого заздалегідь необхідно визначити кількість його входів і виходів.

Кількість входів перетворювача відповідає заданому числу клавіш клавіатури, тобто дорівнює 10.

Для визначення числа виходів перетворювача заздалегідь визначається розрядність коду Грея, виходячи з подвійної нерівності  $2^{n-1} < 10 < 2^n$ , де  $n$  – розрядність коду Грея.

Для задоволення нерівності вибирається  $n = 4$ . Кількість виходів перетворювача складає  $n + 1 = 5$ , оскільки на одному з виходів відповідно умові повинен формуватися контрольний сигнал.

Таблиця 8.5 - Таблиця істинності перетворювача десяткового коду в код Грея

Входи										Виходи				
										Контроль- ний сигнал	Код Грея			
$x_9$	$x_8$	$x_7$	$x_6$	$x_5$	$x_4$	$x_3$	$x_2$	$x_1$	$x_0$	$y_4$	$y_3$	$y_2$	$y_1$	$y_0$
0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	1	0	0	0	1
0	0	0	0	0	0	0	1	0	0	1	0	0	1	1
0	0	0	0	0	0	1	0	0	0	1	0	0	1	0
0	0	0	0	0	1	0	0	0	0	1	0	1	1	0
0	0	0	0	1	0	0	0	0	0	1	0	1	1	1
0	0	0	1	0	0	0	0	0	0	1	0	1	0	1
0	0	1	0	0	0	0	0	0	0	1	0	1	0	0
0	1	0	0	0	0	0	0	0	0	1	1	1	0	0
1	0	0	0	0	0	0	0	0	0	1	1	1	0	1

Етап 3. Логічні вирази для кожного вихідного сигналу, представлені у ДНФ, записуються на основі таблиці істинності:

$$y_0 = x_1 + x_2 + x_5 + x_6 + x_9;$$

$$y_1 = x_2 + x_3 + x_4 + x_5;$$

$$y_2 = x_4 + x_5 + x_6 + x_7 + x_8 + x_9;$$

$$y_3 = x_8 + x_9;$$

$$y_4 = x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9.$$

При складанні логічних виразів використана наступна особливість вхідних змінних: при будь-якій комбінації вхідних сигналів тільки в одному розряді присутня логічна одиниця. Врахування вказаної особливості дозволило отримати рівняння, що не підлягають подальшій мінімізації.

Етап 4. Структурна схема перетворювача, складена на основі отриманих рівнянь, показана на рис. 8.3.

Часову діаграму перетворювача десяткового коду в код Грея показано на рис. 8.4.

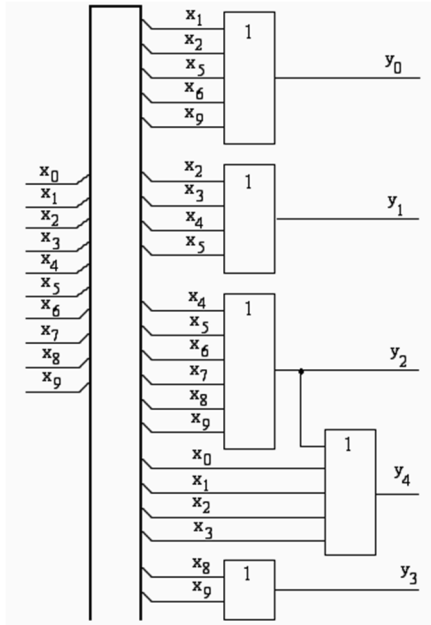


Рисунок 8.3 – Структурна схема перетворювача десяткового коду в код Грея

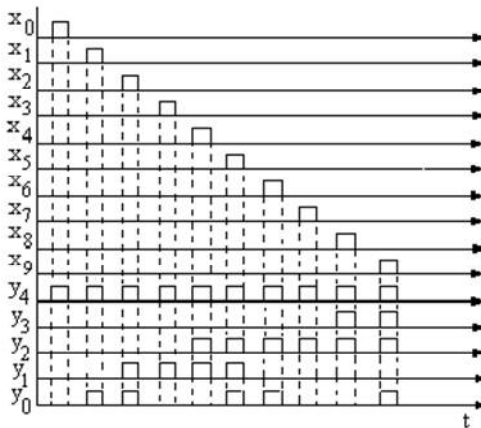


Рисунок 8.4 - Часова діаграма перетворювача десяткового коду в код Грея

### 8.3 Завдання

**8.3.1** Знайдіть досконалу диз'юнктивну нормальну форму (ДДФ) заданої функції відповідно варіанта (таблиця 8.6).

**8.3.2** На основі дешифратора реалізуйте у вигляді схеми логічну функцію  $f(x_1, x_2, x_3)$  із завдання 8.3.1.

**8.3.3** Складіть таблицю перетворення десяткових чисел  $0,1,2,\dots,9$  у заданий код.

Синтезуйте логічну структуру перетворювача десяткових чисел від 0 до 9 у заданий код на основі шифратора.

Побудуйте часову діаграму перетворювача десяткового коду в заданий код.

**Вказівки** до складання таблиці перетворень десяткових чисел  $0,1,2,\dots,9$  у деякі коди.

У чотирирозрядному самодоповнювальному коді з надлишком 3 всі тетради мають значення на 3 одиниці більше (тобто містять надлишок +0011), ніж тетради коду 8421.

У самодоповнювальному коді Айкена (система 2421) у числах від 0 до 4: тетради повторюють двійкові еквіваленти; у числах від 5 до 9: у порівнянні з двійковою системою кожна тетрада містить надлишок +0110. Це дає можливість будь-яку цифру однієї частини таблиці перетворити в її доповнення до 9 простим інвертуванням.

У чотирирозрядному коді 7421 у числах від 0 до 6 включно: тетради повторюють двійкові еквіваленти, число 7 представлено кодом 1000, далі – по наростанню. Особливість коду 7421: будь-яка кодова комбінація містить не більш двох одиниць. Ця особливість використовується для виявлення помилкових комбінацій.

У п'ятирозрядному коді 2 з 5 число 0 представляється кодом 11000, наступні числа 1, 2, 3, 4 формуються циклічним зсувом на один розряд праворуч коду попереднього числа. Число 5 представляється кодом 10100, далі коди чисел формуються по тому ж правилу. У коді 2 з 5 усі кодові комбінації містять тільки 2 одиниці. Ця властивість використовується для виявлення помилкових комбінацій. Збільшення або зменшення кількості одиниць говорить про наявність помилки.

Для побудови коду Грея варто почати з нульової комбінації, а потім змінювати на кожному кроці саму молодшу двійкову цифру, яку

тільки можливо, щоб комбінації не повторювалися. Такий код зручний при передачі інформації про процеси, що повільно змінюються.

У чотирирозрядному кодi 5121 у числах від 0 до 3 тетради повторюють двійкові еквіваленти. Число 4 представлено кодом 0111. У старшому розряді чисел від 5 до 9 включно присутня 1. Тріади з трьох молодших розрядів чисел від 5 до 9 включно повторюють такі ж тріади відповідних чисел від 0 до 4. Це дає можливість будь-який код верхньої частини таблиці перетворити у відповідний код з нижньої частини таблиці додаванням 1 у старший розряд.

У п'ятирозрядному самодоповнювальному кодi  $3a+2$  позначення  $a$  відповідає десятковому числу ( $a = 0, 1, \dots, 9$ ). Особливість коду  $3a+2$ : код є самодоповнювальним, тому будь-який код однієї частини таблиці можна перетворити в його доповнення до 9 простим інвертуванням.

Таблиця 8.6 - Варіанти завдань

№ вар.	Завдання 8.3.1	Завдання 8.3.3
	Функція $F$	Тип коду
1	$\bar{x}_1\bar{x}_2 + \bar{x}_1x_2 + x_1x_2x_3$	Чотирирозрядний код із надмірністю 3
2	$x_1\bar{x}_2 + \bar{x}_2x_3 + \bar{x}_1\bar{x}_2x_3$	Чотирирозрядний код Айкена (код 2421)
3	$x_1\bar{x}_2 + \bar{x}_2x_3 + x_1x_3$	Чотирирозрядний код 7421
4	$\bar{x}_1x_2 + \bar{x}_1x_3 + \bar{x}_1x_2x_3$	П'ятирозрядний код 2 з 5
5	$x_1\bar{x}_2 + \bar{x}_2x_3 + \bar{x}_1x_2\bar{x}_3$	Чотирирозрядний код Грея
6	$\bar{x}_1x_3 + \bar{x}_2\bar{x}_3 + \bar{x}_1x_3$	Чотирирозрядний код 732-1
7	$x_1\bar{x}_2 + x_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3$	Чотирирозрядний код 5121
8	$\bar{x}_1x_2 + \bar{x}_2\bar{x}_3 + \bar{x}_1x_2x_3$	П'ятирозрядний код $3a+2$
9	$x_1\bar{x}_2 + \bar{x}_1x_3 + \bar{x}_1x_2x_3$	Чотирирозрядний код 5311
10	$\bar{x}_1x_2 + \bar{x}_1\bar{x}_3 + \bar{x}_1\bar{x}_2\bar{x}_3$	Чотирирозрядний код 5421
11	$x_1x_2 + \bar{x}_1x_3 + x_1\bar{x}_2x_3$	Чотирирозрядний код 7421
12	$x_1x_2 + x_1x_3 + x_1x_2x_3$	Чотирирозрядний код 7321
13	$\bar{x}_1\bar{x}_2 + x_1x_3 + \bar{x}_1x_2\bar{x}_3$	Чотирирозрядний код 6421

## Продовження таблиці 8.6

№ вар.	Завдання 8.3.1	Завдання 8.3.3
	Функція $F$	Тип коду
14	$x_1 x_2 + \bar{x}_1 x_3 + x_1 \bar{x}_2 \bar{x}_3$	Чотирирозрядний код 6321
15	$x_1 \bar{x}_2 + \bar{x}_1 \bar{x}_3 + \bar{x}_1 \bar{x}_2 x_3$	Чотирирозрядний код 6311
16	$\bar{x}_1 \bar{x}_3 + x_2 \bar{x}_3 + x_1 \bar{x}_2 x_3$	Чотирирозрядний код 6221
17	$\bar{x}_1 x_3 + x_2 \bar{x}_3 + \bar{x}_1 x_2 \bar{x}_3$	Чотирирозрядний код 5221
18	$\bar{x}_1 x_3 + x_2 \bar{x}_3 + \bar{x}_1 x_2 \bar{x}_3$	Чотирирозрядний код 5211
19	$\bar{x}_1 \bar{x}_3 + x_2 x_3 + \bar{x}_1 x_2 \bar{x}_3$	Чотирирозрядний код 4421
20	$x_1 x_3 + \bar{x}_2 x_3 + x_1 \bar{x}_2 \bar{x}_3$	Чотирирозрядний код 4321
21	$x_1 \bar{x}_3 + x_2 x_3 + x_1 \bar{x}_2 x_3$	Чотирирозрядний код 4311
22	$\bar{x}_1 \bar{x}_3 + x_2 \bar{x}_3 + x_1 x_2 \bar{x}_3$	Чотирирозрядний код 4221
23	$x_1 x_3 + x_2 \bar{x}_3 + x_1 \bar{x}_2 x_3$	Чотирирозрядний код 3331
24	$x_1 \bar{x}_2 + x_1 \bar{x}_3 + x_1 x_2 \bar{x}_3$	П'ятирозрядний код 51111
25	$x_1 \bar{x}_2 + x_1 x_3 + \bar{x}_1 \bar{x}_2 x_3$	Чотирирозрядний код 54-2-1
26	$x_1 x_2 + \bar{x}_1 x_3 + \bar{x}_1 x_2 \bar{x}_3$	Чотирирозрядний код 63-2-1
27	$\bar{x}_1 x_2 + \bar{x}_1 \bar{x}_3 + \bar{x}_1 x_2 \bar{x}_3$	Чотирирозрядний код 63-1-1
28	$x_1 \bar{x}_2 + x_1 \bar{x}_3 + \bar{x}_1 x_2 \bar{x}_3$	Чотирирозрядний код 652-4
29	$\bar{x}_1 \bar{x}_2 + x_1 x_3 + \bar{x}_1 x_2 \bar{x}_3$	Чотирирозрядний код 632-4
30	$x_1 \bar{x}_2 + \bar{x}_1 x_3 + \bar{x}_1 \bar{x}_2 \bar{x}_3$	Чотирирозрядний код 751-4
31	$\bar{x}_1 x_2 + \bar{x}_1 x_3 + x_1 \bar{x}_2 x_3$	Чотирирозрядний код 842-3
32	$x_1 x_2 + x_1 x_3 + x_2 x_3$	Чотирирозрядний код 843-2
33	$\bar{x}_1 \bar{x}_2 + x_1 x_3 + x_2 x_3$	Чотирирозрядний код 652-4
34	$\bar{x}_1 x_2 + x_1 x_3 + \bar{x}_2 \bar{x}_3$	Чотирирозрядний код 442-1
35	$\bar{x}_1 x_2 + x_1 x_3 + x_2 \bar{x}_3$	Чотирирозрядний код 732-1
36	$\bar{x}_1 \bar{x}_2 + x_1 x_3 + x_2 x_3$	Чотирирозрядний код 632-1

## 8.4 Порядок оформлення звіту

Звіт повинен містити:

- мету роботи;
- запис заданої логічної функції у ДНФ, її таблицю істинності;
- таблицю істинності дешифратора "3:8";
- логічну схему, що реалізує задану функцію на основі дешифратора (відповідно завданню 8.3.2);
- таблицю істинності перетворювача десяткового коду у заданий код на основі шифратора, подібну табл. 8.5;
- систему логічних рівнянь, що описує роботу перетворювача, складену на основі таблиці істинності перетворювача;
- структурну схему перетворювача десяткового коду у заданий код;
- часову діаграму роботи синтезованого перетворювача.

## 8.5 Контрольні запитання

- 1 Яке правило переводу диз'юнктивної нормальної форми (ДНФ) логічної функції у досконалу диз'юнктивну нормальну форму (ДДНФ)?
- 2 Чим операція шифрації інформації відрізняється від операції дешифрації?
- 3 Яке функціональне призначення шифратора?
- 4 Чим повний шифратор відрізняється від неповного?
- 5 Який код називається ваговим двійково-десятковим кодом?
- 6 За яким правилом будується ваговий двійково-десятковий код?
- 7 Які властивості має код Грея?
- 8 Яке правило формування коду Грея?
- 9 Який код називається рефлексним?
- 10 Навіщо в таблиці істинності перетворювача десяткового коду в код Грея (табл. 8.5) використовується додатковий контрольний розряд?
- 11 З яких етапів складається методика синтезу логічної структури перетворювача десяткового коду в заданий код?
- 12 Чим відрізняється позиційний код від непозиційного?

## 9 САМОСТІЙНА РОБОТА № 7 „СИНТЕЗ РЕГІСТРІВ”

**Мета роботи** - вивчення принципів роботи регістрів і методів їх проектування.

### 9.1 Теоретичні відомості

#### 9.1.1 Структура регістра та виконувани ним операції

Узагальнена структурна схема регістра представлена на рис. 9.1.

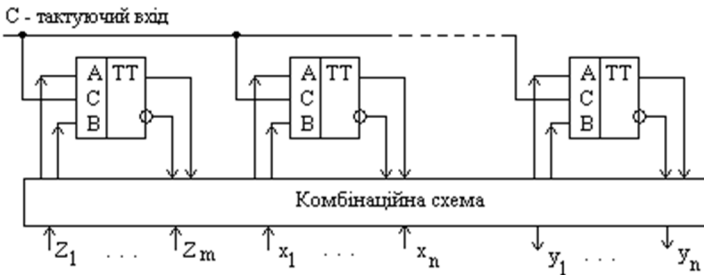


Рисунок 9.1 - Узагальнена структурна схема регістра

До основних операцій, що виконуються регістрами, відносяться:

- установка в початковий стан ( $Z_1$ ) – це установка розрядів регістра в 0 або 1, яка виконується сигналами, що подаються одночасно на асинхронні входи  $R$  і  $S$  всіх тригерів регістра;

- приймання (запис) двійкового числа ( $Z_2$ ) полягає в запису в  $i$ -й розряд регістра відповідної цифри  $x_i$ , тобто  $Q_i^{n+1} = x_i^n$ ;

- логічне множення двох чисел - порозрядна кон'юнкція ( $Z_3$ ), тобто  $Q_i^{n+1} = Q_i^n \cdot x_i^n$ ;

- логічне підсумовування двох чисел – порозрядна диз'юнкція ( $Z_4$ ), тобто  $Q_i^{n+1} = Q_i^n + x_i^n$ ;

- порозрядне підсумовування за модулем 2 ( $Z_5$ ), тобто  $Q_i^{n+1} = Q_i^n \oplus x_i^n$ ;

- зсув числа на  $j$ -розрядів ( $Z_6$ ) ліворуч, тобто  $Q_i^{n+1} = Q_{i-j}^n$ , або праворуч, тобто  $Q_i^{n+1} = Q_{i+j}^n$ ;
- інвертування всіх розрядів числа ( $Z_7$ ), при якому в кожному розряді регістра здійснюється перетворення:  $Q_i^{n+1} = \overline{Q_i^n}$ ;
- видача числа в прямому ( $Z_8$ ), інверсному ( $Z_9$ ), або парафазному кодах ( $Z_{10}$ ), для чого число, що зчитується з регістра, знімається з прямих, інверсних, або з обох виходів кожного тригера відповідно.

### 9.1.2 Методика проектування регістра

Методика проектування регістра, що виконує задану функцію, містить наступні пункти:

Пункт 1. Вибирається тип тригерів.

Пункт 2. Складається таблиця зміни станів і функцій збудження тригерів на основі табл. 9.1.

Таблиця 9.1 – Таблиця переходів і сигналів збудження тригерів

Переходи	Сигнали збудження					
	<i>J</i>	<i>K</i>	<i>S</i>	<i>R</i>	<i>D</i>	<i>T</i>
0→0	0	*	0	*	0	0
0→1	1	*	1	0	1	1
1→0	*	1	0	1	0	1
1→1	*	0	*	0	1	0

Пункт 3. Записуються логічні вирази функції збудження тригерів в мінімальній формі та в заданому логічному базисі.

Пункт 4. Будуються функціональна і принципова схеми регістра.

Приклад синтезу регістра за заданою функцією.

Синтезувати комбінаційну схему регістра на асинхронних *RS*-тригерах для логічного множення двох чисел, одне з яких зберігається в регістрі в розряді  $Q_i$ , а друге надходить на вхід  $x_i$ .

Рішення:

а) в якості елемента пам'яті відповідно завданню вибирається *RS*-тригер;

б) складається таблиця зміни станів і функцій збудження (табл. 9.2);

Таблиця 9.2 - Таблиця зміни станів і функцій збудження

Вхідні сигнали	Попередній стан	Наступний стан	Сигнали збудження	Примітка
$Z_3 \ x^n$	$Q^n$	$Q^{n+1}$	$R \ S$	Дозволу на множення немає (режим зберігання)
0 0	0	0	* 0	
0 0	1	1	0 *	
0 1	0	0	* 0	
0 1	1	1	0 *	Логічне множення
1 0	0	0	* 0	
1 0	1	0	1 0	
1 1	0	0	* 0	
1 1	1	1	0 *	

г) проводиться мінімізація функції  $R$  за допомогою карти Карно:

$$\begin{array}{c|c|c|c|c}
 & \text{00} & \text{01} & \text{11} & \text{10} \\
 \hline
 Z_3 \left| \begin{array}{l} 0 \\ 1 \end{array} \right. & * & 0 & 0 & * \\
 & * & 1 & 0 & *
 \end{array}$$

$$R = Z_3 \bar{x}^n$$

д) будується логічна схема одного розряду регістра (рис. 9.1).

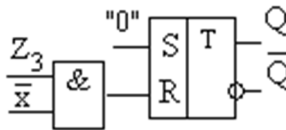


Рисунок 9.1 – Логічна структура одного розряду регістра

Вказівки:

- якщо в рівняннях для входів тригера в якості аргументу входить стан тригера  $Q_i^n$ , де  $i$  - номер розряду, то тригери в схемі повинні мати внутрішню затримку, тобто бути двоступеневими;
- для регістра, на якому можуть виконуватися декілька операцій з використанням однойменних входів тригерів, узагальнені функції збудження тригерів є диз'юнкцією однойменних функцій збудження, відповідних окремим операціям.

Наприклад, з використанням  $T$ -тригерів:

операція  $Z_5$ :  $T = Z_5 \bar{x}^n Q^n + Z_5 x^n \bar{Q}^n$ ; операція  $Z_2$ :  $T = Z_2 x^n$ .

Логічна структура одного розряду регістра при виконанні операцій  $Z_2$  та  $Z_5$  показана на рис. 9.2.

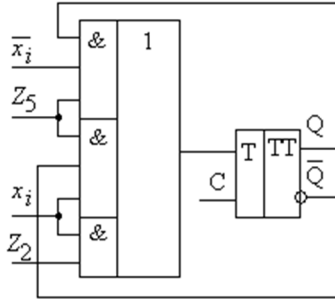


Рисунок 9.2 – Логічна структура одного розряду регістра при виконанні операцій  $Z_2$  та  $Z_5$

При використанні синхронних тригерів необхідно враховувати наступні особливості:

- сигнал дозволу операції  $Z_i$  і сигнали установлення функцій збудження повинні надходити раніше синхросигналу;

- якщо на регістрі виконується тільки одна операція, то на синхровхід можна подати сигнал дозволу операцій  $Z_i$  (тоді в таблицю переходів змінна  $Z_i$  не включається, що спрощує комбінаційну схему);

- якщо на регістрі виконується декілька операцій, то синтез комбінаційної схеми на синхронний тригер такий же, як і на асинхронний, тобто функції збудження синтезуються для кожної операції окремо, потім з'єднуються схемою АБО і подаються на входи тригера.

## 9.2 Завдання

Синтезувати регістр за заданими функціями. Варіанти завдання приведені у таблиці 9.3.

Таблиця 9.3 – Варіанти завдань

№ вар.	Номери функцій $z_i$	Тип тригера	№ вар.	Номери функцій $z_i$	Тип тригера
1	2,5	D	21	5,7	JK
2	3,5	JK	22	7,5	D
3	2,4	JK	23	4,5	RS
4	7,5	RS	24	5,7	JK
5	7,4	JK	25	3,5	D
6	3,7	RS	26	7,4	RS
7	3,5	D	27	7,2	JK
8	5,7	JK	28	3,5	RS
9	4,5	RS	29	2,5	D
10	3,7	D	30	7,2	JK
11	7,2	JK	31	3,4	RS
12	4,5	D	32	4,5	JK
13	7,3	JK	33	2,5	RS
14	3,5	RS	34	3,5	JK
15	7,4	JK	35	2,5	D
16	5,2	RS	36	7,4	JK
17	3,5	JK	37	2,4	D
18	3,5	JK	38	7,2	RS
19	2,5	RS	39	5,7	RS
20	2,7	D	40	7,3	D

### 9.3 Порядок оформлення звіту

Звіт повинен містити: мету роботи; узагальнену схему регістра; таблиці зміни станів і функцій збудження для заданих функцій відповідно варіанта; мінімізовані рівняння функцій збудження, що отримуються на основі таблиць; логічну структуру одного розряду регістра.

### 9.4 Контрольні запитання

- 1 Які пункти містить методика синтезу регістра з заданими функціями?
- 2 У яких випадках тригери повинні обов'язково мати внутрішню затримку?
- 3 Яке співвідношення повинно бути між моментами надходження сигналу дозволу, сигналами установлення операцій та синхросигналом у регістрі?

## 10 САМОСТІЙНА РОБОТА № 8 „СИНТЕЗ ЛІЧИЛЬНИКІВ”

**Мета роботи** - вивчення принципів функціонування кільцевих лічильників (лічильників Джонсона), освоєння методики синтезу кільцевих лічильників (лічильників Джонсона) на основі тригерів різних типів; освоєння методики синтезу лічильників з заданою послідовністю станів, яка періодично повторюється.

### 10.1 Теоретичні відомості

Кільцеві лічильники використовуються в системах керування для формування послідовностей імпульсів. Синтезуються вони на основі регістрів зсуву, для чого останній вихід регістра зсуву з'єднується зі входом першого тригера регістра (рис. 10.1).

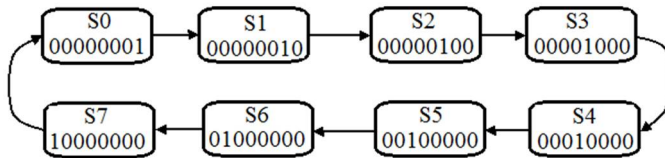


Рисунок 10.1 - Діаграма станів кільцевого лічильника

На діаграмі станів вхідні сигнали відсутні, оскільки єдиним вхідним сигналом для лічильника є синхронізуючий сигнал. Послідовно на кожному виході кільцевого лічильника –  $Q_7, Q_6, Q_5, Q_4, Q_3, Q_2, Q_1, Q_0$  – формується одиничний імпульс тривалістю в один період тактового сигналу. Ці імпульси можуть використовуватися для управління функціональними блоками цифрової системи.

Початковий стан лічильника встановлюється за допомогою асинхронних сигналів установки. Схема лічильника, що відповідає приведеній діаграмі станів, показана на рис. 10.2, а часова діаграма його роботи – на рис. 10.3.

Якщо сигнал з інверсного виходу останнього тригера подати на вхід першого тригера, то утворюється перехресний кільцевий лічильник, званий лічильником Джонсона (рис. 10.4).

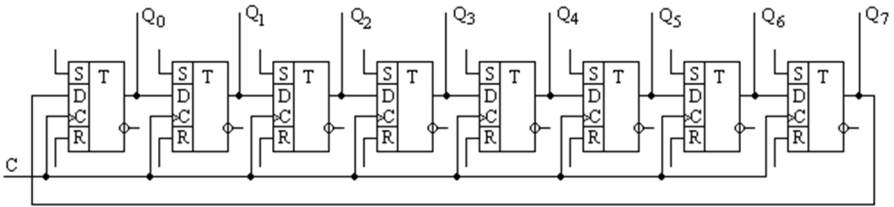


Рисунок 10.2 – Кільцевий лічильник зі входами скидання і установки

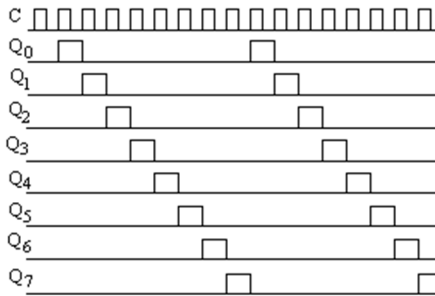


Рисунок 10.3 – Формування імпульсів керування за допомогою кільцевого лічильника

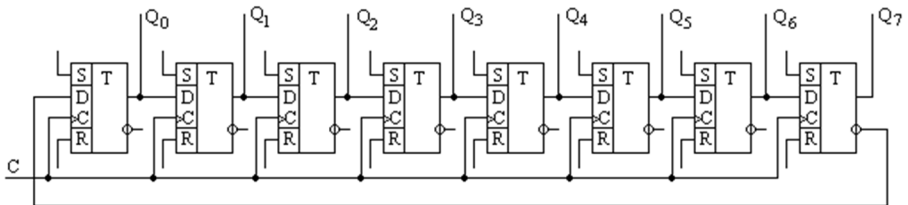


Рисунок 10.4 – Перехресний кільцевий лічильник

Лічильник Джонсона, що має нульовий початковий стан, формує повторювальну послідовність: 00000000, 00000001, 00000011, 00000111, 00001111, 00011111, 00111111, 01111111, 11111111, 11111110, 11111100, 11111000, 11110000, 11100000, 11000000, 10000000, 00000000, ... . Кожна кодова комбінація відрізняється від сусідніх кодових комбінацій тільки одним бітом, що надзвичайно зручно для багатьох областей застосувань, оскільки при зміні кодової комбінації на наступну затримки в тригерах не викликатимуть

короткочасної появи інших комбінацій, що могло б відбутися при одночасній зміні вмісту декількох розрядів.

При синтезі кільцевих лічильників необхідно враховувати початкові умови, оскільки при подачі живлення на логічну схему виходи всіх тригерів знаходяться, як правило, в довільних станах. Коли надходить на синхровхід лічильника перший синхроімпульс може виникнути ситуація, при якій сформована на виходах тригерів кодова комбінація не належить кодовій послідовності, яку необхідно отримати. У цьому випадку виникає можливість генерації нової послідовності станів, яка не включатиме жодної кодової комбінації з потрібної послідовності. Якщо ж на виходах лічильника з'явиться при включенні одна з комбінацій, що входять в потрібну послідовність, то лічильник почне працювати правильно.

Проблема початкових умов може бути розв'язана двома способами. Перший спосіб полягає в такому відображенні вихідних значень, коли будь-яка невизначена комбінація вихідних значень рано чи пізно переходить в одну із заданих вихідних комбінацій під час надходження чергового тактового імпульсу. У другому способі використовуються асинхронні входи скидання/установки для примусової установки потрібних початкових умов.

На основі першого методу спроектований лічильник, на виході якого повторюється послідовність 000, 001, 011, 110, 000, .... Діаграма станів такого лічильника приведена на рис. 10.5, де показано, що всі невизначені кодові комбінації переходять в стан 000 під час надходження наступного ж тактового імпульсу.

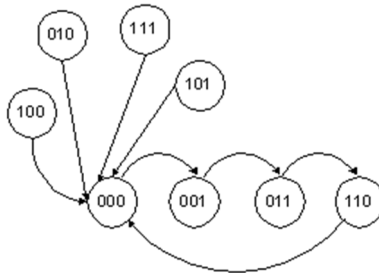


Рисунок 10.5 – Діаграма станів лічильника, в якій всі невизначені комбінації переходять у стан 000

Таблиця переходів і сигналів збудження  $D$ -тригерів, складена на основі діаграми станів, приведена у табл. 10.1. Якщо необхідно використовувати інші типи тригерів у схемі лічильника, то при формуванні таблиці переходів і сигналів збудження необхідно скористатися табл. 9.1 (самостійна робота № 7).

Таблиця 10.1 - Таблиця переходів і сигналів збудження  $D$ -тригерів

Попередній стан			Наступний стан			Сигнали збудження $D$ -тригерів		
$Q_2$	$Q_1$	$Q_0$	$Q_2$	$Q_1$	$Q_0$	$D_2$	$D_1$	$D_0$
1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	1
0	1	1	1	1	0	1	1	0
1	1	0	0	0	0	0	0	0

Карти Карно вхідних функцій тригерів зображені на рис. 10.6.

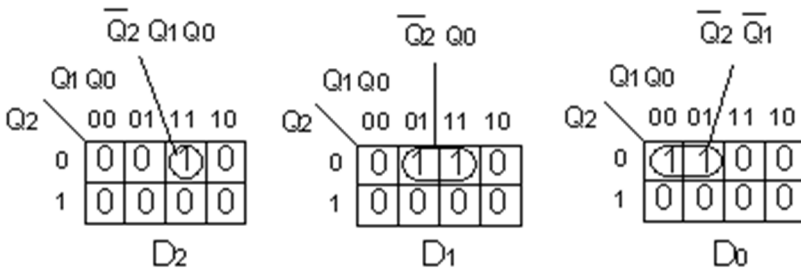
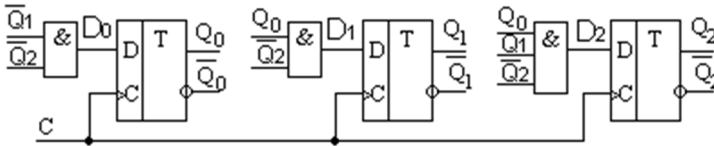


Рисунок 10.6 - Карти Карно вхідних функцій тригерів

Схема лічильника зображена на рис. 10.7. Для спрощення зображення схеми відповідні зв'язки між входами і виходами схеми не показані, на входах схеми наведені тільки відповідні надписи.

Рисунок 10.7 – Трирозрядний лічильник на  $D$ -тригерах

## 10.2 Завдання

**10.2.1** Складіть діаграму станів, схему і часову діаграму роботи 8-розрядного лічильника. Тип лічильника і його початковий стан задається відповідно до варіанта (табл. 10.2).

**10.2.2** Синтезуйте схему 3-розрядного лічильника із заданою послідовністю кодових комбінацій. При завданні початкових умов використовуйте спосіб 1. Типи тригерів задаються відповідно до варіанта.

## 10.3 Порядок оформлення звіту

Звіт повинен містити: мету роботи; діаграму станів, схему і часову діаграму роботи 8-розрядного лічильника відповідно завданню 10.2.1; діаграму станів трирозрядного лічильника (завдання 10.2.2), його таблицю переходів і сигналів збудження тригерів; карти Карно для мінімізації функцій збудження тригерів, що входять до складу трирозрядного лічильника; логічну структуру трирозрядного лічильника.

## 10.4 Контрольні запитання

- 1 На основі яких функціональних вузлів синтезуються кільцеві лічильники?
- 2 Навіщо в схемах кільцевих лічильників використовуються входи скидання і установки?
- 3 Чим схема кільцевого лічильника принципово відрізняється від схеми лічильника Джонсона?
- 4 Внаслідок чого виникає проблема початкових умов при синтезі лічильників?
- 5 Як розв'язується проблема початкових умов при синтезі лічильників?

Таблиця 10.2 – Вихідні дані до завдань 10.2.1 та 10.2.2

№ вар.	Завдання 10.2.1		Завдання 10.2.2	
	Тип лічильника, тип тригерів	Початковий стан	Тип тригерів	Послідовність кодових комбінацій (в вісімковій СЧ)
1	Кільцевий, JK	00001000	D	1,2; 0,2; 7,3; 4,5; 2,3; 3,6; 6,5; 5,2
2	Джонсона, D	11000000	JK	0,5; 6,7; 2,3; 1,3; 3,4; 4,5; 5,7; 7,3
3	Кільцевий, JK	00000010	D	0,4; 1,4; 5,4; 6,4; 4,7; 7,3; 3,2; 2,4
4	Джонсона, JK	11110000	JK	1,0; 2,7; 4,6; 3,6; 6,5; 5,0; 0,7; 7,6
5	Кільцевий, JK	00000001	D	4,1; 5,1; 6,0; 2,0; 0,3; 3,7; 7,1; 1,0
6	Джонсона, D	11100000	JK	5,0; 3,0; 2,0; 6,4; 4,1; 1,0; 0,7; 7,4
7	Кільцевий, JK	00000100	D	0,4; 6,1; 7,1; 2,1; 1,5; 5,3; 3,4; 4,1
8	Кільцевий, D	10000000	JK	1,7; 5,7; 2,6; 3,6; 6,4; 4,0; 0,7; 7,6
9	Джонсона, JK	00011111	D	5,2; 3,6; 0,6; 7,6; 6,4; 4,1; 1,2; 2,6
10	Кільцевий, D	00000010	JK	0,5; 3,5; 1,7; 2,7; 7,4; 4,5; 5,6; 6,7
11	Джонсона, D	00000001	JK	3,1; 6,1; 4,5; 7,5; 5,2; 2,0; 0,1; 1,5
12	Кільцевий, JK	01000000	D	0,4; 1,4; 2,7; 3,7; 7,6; 6,5; 5,4; 4,7
13	Кільцевий, JK	10000000	D	4,7; 1,6; 2,6; 5,6; 6,0; 0,3; 3,7; 7,6
14	Джонсона, D	00001111	JK	1,5; 0,3; 4,3; 3,7; 7,6; 6,2; 2,5; 5,3
15	Кільцевий, D	00010000	JK	0,2; 1,2; 3,6; 4,6; 6,2; 2,7; 7,5; 5,6
16	Джонсона, D	00000111	JK	1,3; 5,7; 2,6; 4,6; 6,0; 0,3; 3,7; 7,6
17	Кільцевий, JK	00100000	D	0,2; 5,3; 7,4; 4,6; 6,1; 1,2; 2,3; 3,4
18	Кільцевий, JK	10000000	D	7,2; 4,0; 3,6; 1,6; 6,2; 2,0; 0,5; 5,6
19	Джонсона, D	00000011	JK	6,7; 0,4; 2,4; 1,5; 5,3; 3,7; 7,4; 4,5
20	Кільцевий, D	01000000	JK	3,1; 0,7; 7,4; 4,5; 5,6; 6,1; 1,2; 2,7
21	Джонсона, D	11111000	JK	7,4; 6,4; 0,1; 2,5; 5,3; 3,4; 4,1; 1,5
22	Кільцевий, D	01000000	JK	5,4; 6,4; 7,3; 3,2; 2,1; 1,0; 0,4; 4,3
23	Кільцевий, D	00001000	JK	0,5; 1,7; 4,7; 6,7; 7,3; 3,2; 2,5; 5,7
24	Джонсона, D	11100000	JK	5,4; 0,4; 6,3; 3,1; 1,2; 2,7; 7,4; 4,3
25	Кільцевий, D	00000001	JK	3,0; 5,1; 7,4; 4,6; 6,0; 0,2; 2,1; 1,4
26	Джонсона, D	10000000	JK	3,1; 6,5; 4,5; 0,7; 7,1; 1,2; 2,5; 5,7
27	Кільцевий, JK	00100000	D	1,3; 0,4; 7,5; 5,6; 6,2; 2,3; 3,4; 4,5
28	Джонсона, JK	00000111	D	6,1; 2,5; 3,0; 4,0; 0,7; 7,1; 1,5; 5,0
29	Кільцевий, JK	00100000	D	0,6; 5,2; 3,2; 7,2; 2,4; 4,6; 6,1; 1,2
30	Джонсона, D	00001111	JK	0,5; 4,5; 6,2; 1,2; 2,3; 3,7; 7,5; 5,2
31	Кільцевий, D	00100000	JK	0,4; 1,4; 2,7; 3,7; 7,6; 6,5; 5,4; 4,7
32	Кільцевий, JK	00010000	D	4,7; 5,1; 6,0; 2,0; 0,3; 3,7; 7,1; 1,0
33	Джонсона, D	11111000	JK	1,2; 0,3; 4,3; 3,7; 7,6; 6,2; 2,5; 5,3
34	Кільцевий, JK	00000100	D	3,6; 0,7; 7,4; 4,5; 5,6; 6,1; 1,2; 2,7
35	Джонсона, JK	11110000	D	1,5; 2,6; 4,6; 3,6; 6,5; 5,0; 0,7; 7,6
36	Кільцевий, D	00010000	JK	7,2; 4,0; 3,6; 1,6; 6,2; 2,0; 0,5; 5,6
37	Джонсона, JK	11111000	JK	7,1; 6,5; 0,1; 2,5; 5,3; 3,4; 4,1; 1,5
38	Кільцевий, JK	00001000	D	7,5; 4,2; 3,6; 1,6; 6,2; 2,0; 0,5; 5,6
39	Кільцевий, JK	00001000	D	4,3; 1,0; 2,6; 5,6; 6,0; 0,3; 3,7; 7,6
40	Джонсона, D	00011111	JK	1,3; 0,2; 4,6; 3,7; 7,6; 6,2; 2,5; 5,3

## **11 ПЕРЕЛІК ЗАПИТАНЬ, ЯКІ ВІНОСЯТЬСЯ НА МОДУЛЬНИЙ КОНТРОЛЬ**

### **11.1 Перший модульний контроль (7 семестр)**

Перший модульний контроль містить питання з робочої програми дисципліни, що вивчаються при виконанні першого модуля, який містить змістовий модуль під назвою: „Базові компоненти та комбінаційні функціональні вузли”.

Запитання, які треба розглянути, щоб пройти перший модульний контроль, приведені нижче.

- 1 Задачі дисципліни.
- 2 Визначення цифрової системи.
- 3 Визначення цифрового пристрою.
- 4 Визначення цифрової техніки, цифрової схемотехніки, мікросхемотехніки.
- 5 Математичний апарат цифрової схемотехніки. Аксиоми і закони алгебри логіки.
- 6 Аналітичне представлення функцій алгебри логіки.
- 7 Мінімізація булевих функцій за допомогою карт Карно.
- 8 Метод Квайна.
- 9 Способи подання цифрової інформації. Види логік. Особливості реальних цифрових пристроїв. Паразитні зв'язки цифрових елементів по ланцюгам живлення.
- 10 Біполярний ключ, його параметри.
- 11 Ключові схеми на польових транзисторах.
- 12 Комплементарний ключ, його переваги і недоліки.
- 13 Перемикач струму. Емітерний повторювач.
- 14 Типи логічних ІС. Схемотехнічні рішення сучасних логічних схем.
- 15 ТТЛ. Робота базового логічного елемента 155 серії.
- 16 ТТЛШ. Мікросхеми ТТЛ з відкритим колектором.
- 17 Емітерно-зв'язана логіка. Робота базового логічного елемента серії К500.
- 18 КМОН-логіка. Буферний підсилювач. КМОН-інвертор з трьома вихідними станами.
- 19 КМОН-логіка. Ключ комутації з двома напрямками. Логічні елементи І-НЕ, АБО-НЕ.

- 20 Комбінаційні цифрові вузли. Мультиплектори. Реалізація логічних функцій на мультиплексорі. Способи нарощування мультиплексорів.
- 21 Комбінаційні цифрові вузли. Демультіплектори і дешифратори.
- 22 Комбінаційні цифрові вузли. Шифратори.
- 23 Арифметичні пристрої. Класифікація суматорів. Півсуматор.
- 24 Повний однорозрядний суматор. Реалізація однорозрядного суматора на двох півсуматорах.
- 25 Багаторозрядні суматори. Послідовний багаторозрядний суматор.
- 26 Паралельні багаторозрядні суматори. Паралельний суматор з послідовним переносом.
- 27 Багаторозрядний суматор з паралельним переносом. Суматор з груповим переносом.

## 11.2 Другий модульний контроль (7 семестр)

Другий модульний контроль містить питання з робочої програми дисципліни, що вивчаються при виконанні другого модуля, який містить змістовий модуль під назвою: „Послідовнісні функціональні вузли і компоненти”.

Запитання, які треба розглянути, щоб пройти другий модульний контроль, приведені нижче.

- 1 Поняття про послідовнісні логічні схеми. Способи опису цифрових автоматів.
- 2 Тригери, їхні властивості, види тригерів. Асинхронний  $RS$ -тригер на елементах АБО-НЕ.
- 3 Асинхронний  $\overline{R}\overline{S}$ -тригер на елементах І-НЕ.
- 4 Синхронний  $RS$ -тригер на елементах І-НЕ зі статичним керуванням.
- 5 Двоступеневі тригери. Двоступеневий  $RS$ -тригер з проміжним інвертором.
- 6  $JK$ -тригери.  $JK$ -тригер зі статичним керуванням.
- 7 Двоступеневі тригери. Двоступеневий  $JK$ -тригер з проміжним інвертором.
- 8  $D$ -тригери.  $D$ -тригер зі статичним керуванням.  $DV$ -тригер.

- 9 *D*- тригер з динамічним керуванням. Організація *D*- тригера на базі *JK*- тригера.
- 10 *T*- тригери. Організація *T*- тригерів на базі інших видів тригерів.
- 11 Регістри, їхня класифікація, регістри пам'яті. Чотирирозрядний регістр пам'яті на *D*- тригерах.
- 12 Регістри, їхня класифікація, регістри зсуву. Чотирирозрядний регістр зсуву з послідовним вводом і зсувом інформації.
- 13 Лічильники і дільники. Їхня класифікація. Асинхронний двійковий лічильник з послідовним переносом.
- 14 Лічильники з паралельним переносом. Лічильник з модулем лічби, рівним 8.
- 15 Синтез лічильників з заданим модулем лічби.
- 16 Програмовані логічні матриці. Матриці "Т" та "АБО".
- 17 ПЛМ. Дворівневі ПЛМ комбінаційного типу. Трирівневі ПЛМ комбінаційного типу.
- 18 Програмовані матриці вентилів (ПМВ). Програмовані матриці логіки (ПМЛ).
- 19 Пристрої пам'яті. Поняття про оперативні запам'ятовувальні пристрої (ОЗП).
- 20 Типова структурна схема ВІС ОЗП.
- 21 Статична пам'ять (*SRAM*). Види *SRAM*. Статичний запам'ятовувальний елемент на МОН-транзисторах.
- 22 Динамічна пам'ять (*DRAM*), принципи її роботи. Види *DRAM*.
- 23 Постійні запам'ятовувальні пристрої (ПЗП), їх особливості, області застосування.
- 24 Структура постійного запам'ятовувального пристрою з однорозрядною організацією пам'яті. Структура багаторозрядного ПЗП.
- 25 Класифікація постійних запам'ятовувальних пристроїв (ПЗП) за способом занесення інформації (масочні ПЗП, програмовані ПЗП, перепрограмовані ПЗП).
- 26 Цифроаналогові перетворювачі. ЦАП з підсумовуванням зважених струмів.
- 27 ЦАП на основі резистивної матриці R-2R.
- 28 Цифроаналогові перетворювачі. Організація виходу по напрузі для ЦАП зі струмовим виходом.

- 29 Аналого-цифрові перетворювачі. Послідовний АЦП з генератором східчастої напруги.
- 30 Аналого-цифрові перетворювачі. АЦП порозрядного врівноваження.
- 31 Паралельні аналогоцифрові перетворювачі.
- 32 Перетворювачі рівнів сигналів.
- 33 Детектори фронтів імпульсів.
- 34 Автоколебальний та очікуючий мультивібратори на дискретних та інтегральних компонентах.
- 35 Блокінг-генератори.
- 36 Формувачі та генератори лінійнозмінного струму.

## РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. **Коваленко, О.Є.** Комп'ютерна схемотехніка. Частина 2. Навч. посіб. [Текст] / О.Є. Коваленко, С.М. Волошин, Б.С. Гусев, Є.В. Нікітенко, В.В. Матієвський. – К.: НУБіП України, 2023.- 331с.
2. **Сенько, В.І.** Електроніка і мікропроцесорна техніка [Текст] / В.І. Сенько, В.П. Лисенко, О.М. Юрченко, В.Є. Лукін, А.А. Руденський. – К.: «Агроосвіта», 2015. – 676 с.
3. **Макаренко, В.В.** Цифрова та імпульсна схемотехніка. Моделювання та аналіз: Навч. посіб. [Текст] / В.В. Макаренко, В.М. Спивак. – Київ: НТУУ "КПІ", 2015. – 314 с.
4. **Петух, А.М.** Цифрова схемотехніка: Навч. посіб. [Текст] / А.М. Петух, Д.Т. Обідник, М.Д. Обідник. – Вінниця: ВНТУ, 2015. – 120 с.
5. **Рябенький, В.М.** Цифрова схемотехніка [Текст] / В.М. Рябенький, В.Я. Жуйков, В.Д. Гулий. – Львів: Новий Світ-2000, 2020. – 736 с.
6. **Колонтаєвський, Ю.П.** Промислова електроніка та мікросхемотехніка: теорія і практикум. Навч. посіб. [Текст] / Ю.П. Колонтаєвський, А.Г. Сосков. За ред А.Г. Соскова. - К.: Каравела, 2003. – 368 с.
7. **Бабич, М.П.** Комп'ютерна схемотехніка: Навч. посіб. [Текст] / М.П. Бабич, І.А. Жуков. – К.: "МК-Прес", 2004. – 412 с.