

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

**Методичні вказівки**

для самостійної роботи студентів заочної форми навчання,  
виконання контрольної роботи та курсового проекту з дисципліни  
**«КОМП'ЮТЕРНА СХЕМОТЕХНІКА ТА АРХІТЕКТУРА  
КОМП'ЮТЕРІВ»**

Напрямок підготовки: – комп'ютерні науки

Затверджено на засіданні  
Робочої групи методичної ради  
університету «Заочна та  
післядипломна освіта»

Одеса 2014

Методичні вказівки для виконання контрольної роботи та курсового проекту для студентів II- III курсів заочної форми навчання. Дисципліна « Комп'ютерна схемотехніка та архітектура комп'ютерів ». Напрямок підготовки – комп'ютерні науки. Препелиця Г.П. к.т.н., доц. , ст. викладач Пономаренко О.Л. – Одеса: ОДЕКУ, 2014р. – 66 с.

## ЗМІСТ

I Загальна частина .....	
1.1 Передмова .....	
1.2 Зміст розділу курсу.....	
1.3 Перелік навчальної літератури .....	
1.4 Перелік знань та вмінь .....	
1.5 Організація навчального процесу .....	
II Організація самостійної роботи студента .....	
2.1 Рекомендації по вивченню теоретичного матеріалу та виконанню контрольної роботи .....	
2.1.1 Загальні поради .....	
2.1.2 Рекомендації по вивченню 1-ї теми .....	
2.1.3 Рекомендації по вивченню 2-ї теми .....	
2.1.4 Рекомендації по вивченню 3-ї теми .....	
2.1.5 Рекомендації по вивченню 4-ї теми .....	
2.1.6 Рекомендації по вивченню 5-ї теми.....	
2.1.7 Рекомендації по вивченню 6-ї теми .....	
2.1.8 Рекомендації по вивченню 7-ї теми .....	
2.1.9 Рекомендації по вивченню 8-ї теми .....	
2.2 Перелік завдань на контрольну роботу .....	
2.2.1 Загальні поради .....	
2.2.2 Перелік завдань контрольної роботи .....	
2.3 Рекомендації по виконанню курсового проекту.....	
2.3.1 Основні вимоги й структура курсового проекту по дисципліні...	
2.3.2 Перелік завдань для курсового проекту.....	
III Організація контролю знань та вмінь .....	
3.1 Система контролю знань та вмінь студентів .....	
3.2 Форми контролю знань та вмінь студентів .....	
3.2.1 Поточний контроль .....	
3.2.2 Підсумковий контроль .....	
3.2.3 Перелік базових знань та вмінь .....	
3.3 Список скорочень	

# І ЗАГАЛЬНА ЧАСТИНА

## 1.1 Передмова

Дисципліна „Комп’ютерна схемотехніка та архітектура комп’ютерів ” належить до професійно-практичного циклу підготовки фахівців з напрямку „Комп’ютерні науки”, спеціальності – „Інформаційні управляючі системи та технології”.

Ця дисципліна є обов’язковою в освітньо-професійної підготовці студентів напрямку комп’ютерні науки.

Мета дисципліни – підготовка майбутніх фахівців в галузі „Комп’ютерні науки”.

Завдання дисципліни – дати студентам знання арифметичних основ цифрової техніки, основних принципів процесу цифрової обробки інформації, структури побудови мікропроцесорних систем обробки даних та керування, архітектури комп’ютерів.

Весь курс складається з восьми тем, які висвітлюють питання арифметичних основ цифрової техніки, логічних основ цифрової техніки, типових комбінаційних вузлів цифрових пристроїв та їх синтезу, архітектури комп’ютерів.

Дисципліна „Комп’ютерна схемотехніка та архітектура комп’ютерів ” надає студенту низку теоретичних знань і практичних навичок, які стануть йому в нагоді при вивченні спеціальних дисциплін. Вивчення курсу „Комп’ютерна схемотехніка та архітектура комп’ютерів ” ґрунтується на основних положеннях і знаннях, отриманих при вивченні дисциплін „Вища математика”, „Основи дискретної математики”, „Електротехніки та електроніки”.

Набуті знання та вміння будуть використані при вивченні навчальної дисципліни: „Технології комп’ютерного проектування”.

Методичні вказівки призначені допомогти студентам заочної форми навчання опанувати цю дисципліну, основні її положення з теорії комп’ютерної схемотехніки та архітектури комп’ютерів, а саме: арифметичні та логічні основи цифрової техніки, синтез та дослідження комбінаційних суматорів, схем порівняння, тригерних схем, регістрів зсуву та лічильників, архітектури комп’ютерів.

Методичні вказівки складаються з рекомендацій до виконання наступних видів робіт:

- самостійного вивчення основних теоретичних розділів дисципліни;
- виконання контрольної роботи та курсового проекту з практичної частини навчального курсу.

У першій частині розглядаються системи числення, способи перекладу чисел з однієї системи числення в іншу, кодування цифрової інформації,

синтез і дослідження комбінаційних суматорів, синтез та дослідження схем порівняння, структурний синтез ЦА, основні елементи і вузли цифрових пристроїв, питання архітектури комп'ютерів.

У другій частині розкривається порядок виконання контрольної роботи та курсового проекту.

В методичних вказівках розглядаються питання, які відповідають навчальній програмі дисципліни.

## 1.2 Зміст розділів дисципліни

*Тема 1 Вступ. Арифметичні основи цифрових пристроїв обробки даних*

Позиційні системи числення. Двійкова система числення. Вісімкова система числення. Шістнадцяткова система числення. Таблиця відповідності чисел у різних системах числення. Використання систем числення в цифрових пристроях. Арифметики систем числення. Переведення чисел з однієї системи числення в іншу. Окремі методи переведення чисел. Загальні методи переведення чисел. Переведення чисел із довільної  $СЧ$  в довільну з виконанням  $АО$  в новій  $h$   $СЧ$ . Переведення чисел з виконанням операцій в старій  $СЧ$ . Пряме переведення чисел з довільної  $СЧ$  в довільну. Переведення чисел з довільної  $СЧ$  в довільну з проміжним перетворенням в десяткову  $СЧ$ . Перетворення даних з врахуванням точності.

*Тема 2 Кодування цифрової інформації*

Прямий, зворотний і додатковий коди двійкових чисел. Прямий код двійкових чисел. Зворотний і додатковий коди. Зв'язок між негативним числом і його зворотним і додатковим кодами. Співвідношення числа і його додаткового коду. Зв'язок між зворотним і додатковим кодами. Нулі у кодах. Алгебраїчне додавання двійкових чисел, представлених зворотним кодом. Алгебраїчне додавання двійкових чисел, представлених додатковим кодом. Засоби виявлення переповнення розрядної сітки підсумовуючого пристрою процесора. Приклади алгебраїчного підсумовування двійкових чисел із використанням зворотних і додаткових кодів. Виявлення переповнення розрядної сітки. Коди чисел в інших системах числення. Кодування десяткових чисел. Кодування алфавітно-цифрової інформації.

*Тема 3 Логічні основи цифрових пристроїв перетворення даних*

Цифрові перемикальні ланцюги. Моделі цифрових пристроїв. Комбінаційні схеми. Перемикальні функції. Перемикальні функції однієї

змінної. Перемикальні функції двох змінних. Суперпозиція перемикальних функцій.

Функціонально-повні системи базових ПФ. Функціонально-повні системи перемикальних функцій двох змінних. Доказ функціональної повноти різних ФПС. Основні закони алгебри логіки. Доказ законів алгебри логіки. Доказ тотожності логічних виражень за допомогою таблиць істинності. Доказ тотожності булевих виражень за допомогою релейно-контактних схем Гаврилова. Аналітичний доказ тотожності логічних виражень. Нормальні форми представлення перемикальних функцій. Диз'юнктивні й кон'юнктивні нормальні форми представлення перемикальних функцій. Конституенти одиниці й нуля. Досконалі диз'юнктивні і кон'юнктивні нормальні форми перемикальних функцій. Одержання досконалих форм із таблиць істинності.

Перетворення довільної ДНФ (КНФ) у СДНФ (СКНФ). Числове представлення перемикальних функцій. Роз'яснення термінології форм перемикальних функцій.

#### *Тема 4 Аналіз і синтез комбінаційних схем*

Синтез однорозрядного суматора (ОКСМ). Способи організації підсумовування багаторозрядних слів. Організація переносів в паралельних суматорах. Суматори з послідовними переносами. Суматори з паралельними переносами. Суматори із комбінованими переносами. Суматори з пам'яттю.

Синтез та дослідження цифрових схем порівняння. Порівняння слів з константами. Порівняння слів. Порівняння слів за допомогою суматора.

#### *Тема 5 Типові комбінаційні вузли цифрових пристроїв обробки даних*

Цифрові автомати (ЦА). Абстрактні автомати (АА). Способи завдання ЦА. Кінцеві автомати Мілі і Мура. Абстрактний синтез ЦА. Загальна схема структурного автомату. Елементарні автомати (ЕА). Алгоритм структурного синтезу автоматів. Синтез структурних автоматів на JK-, D-, T-, DV-тригерах.

#### *Тема 6 Логічне проектування ЦА*

Основні положення . Структурний синтез цифрових автоматів.

#### *Тема 7 Основні елементи і вузли цифрових пристроїв з пам'яттю*

Тригерні схеми різноманітних типів на ЛЕ І-НЕ, АБО-НЕ. Двотактні тригерні схеми, принцип організації, структурна схема, синтез

принципової схеми, часові діаграми роботи. Перетворення (метаморфози) тригерних схем.

Регістри зберігання та зсуву. Принцип організації зсуву інформації. Зсувові регістри на двотактних тригерах. Синтез багатофункціональних зсувових регістрів на тригерах різних типів.

Лічильники. Синтез синхронних (двійково-десяткових) лічильників з довільним модулем і порядком лічення на тригерах різних типів. Синтез асинхронних лічильників на синхронних JK-, T-, D-тригерах. Синтез реверсивних синхронних лічильників.

## Тема 8 *Архітектура комп'ютерів*

Цифрові комп'ютери. Запам'ятовуючі пристрої. Flash-пам'ять. Процесори. Суперкомп'ютери. Паралельні обчислювальні системи. Універсальні мікропроцесори. Схеми підтримки МП на системних платах. Структури мікропроцесорних систем. RISC- та CISC-процесори.

### 1.3 Перелік навчальної літератури

#### *Основна*

1. Препелиця Г.П. Комп'ютерна схемотехніка: Конспект лекцій., Одеса: „Екологія” 2008 – 339 с.
2. Препелиця Г.П. Комп'ютерна схемотехніка: Практикум. Одеса: „Екологія” 2008 – 250 с.
3. Таненбаум Є. Архітектура комп'ютера. СПб.: Питер, 2005. – 704с.

#### *Допоміжна*

1. Самофалов Цифровая и вѳислительная техника: Учебник для студентов вузов . Под редакцией Є.В. Євреїнова – М. Радіо і зв'язь, 1991. – 464 с.

### 1.4 Перелік знань та вмінь

Після вивчення дисципліни «Комп'ютерна схемотехніка та архітектура комп'ютерів » студенти повинні:

#### **Знати**

- системи числення;
- знати способи перекладу чисел з однієї системи числення в іншу;
- коди чисел;
- алгебраїчне додавання двійкових чисел, представлених різноманітними кодами;
- моделі цифрових пристроїв;
- комбінаційні схеми;
- способи організації підсумовування багаторозрядних слів за допомогою суматорів;
- схеми порівняння.

- що таке абстрактний автомат;
- способи завдання абстрактних ЦА;
- автомати Мілі й Мура;
- задачі синтезу ЦА;
- структурно-повні системи автоматів;
- елементарні автомати;
- алгоритм структурного синтезу автоматів
- що таке тригери та їх властивості;
- що називається регістром їх класифікація;
- які функції виконують регістри;
- як здійснюється паралельний прийом слів в регістри;
- як здійснюється парафазний прийом слів в регістри;
- визначення та класифікація лічильників;
- асинхронні та синхронні лічильники;
- двійково-десяткові коди та їх властивості;
- як синтезуються синхронні, асинхронні та синхронні реверсивні лічильники;
- організація та структура цифрових комп'ютерів загального призначення;
- принципи роботи запам'ятовуючих пристроїв, їх класифікація;
- Flash-пам'ять, стекова пам'ять, RAM-пам'ять;
- конвеєрні обчислення;
- організація та структура процесорів, мікропроцесорів, ядра процесора;
- організація та архітектура суперкомп'ютерів, паралельні обчислювальні системи;
- організація універсальних та спеціалізованих мікропроцесорів;
- структури мікропроцесорних систем. RISC- та CISC-процесори.

### **Вміти**

- переводити числа з однієї системи числення в іншу різноманітними методами;
- представляти двійкові числа в прямому, зворотньому і додатковому кодах;
- здійснювати алгебраїчне підсумовування двійкових чисел із використанням зворотних і додаткових кодів;
- виявляти переповнення розрядної сітки підсумовуючого пристрою процесора;
- одержати досконалі форми із таблиць істинності;
- перетворити довільну ДНФ (КНФ) у СДНФ (СКНФ).;
- синтезувати однорозрядний комбінаційний суматор (ОКСМ);
- синтезувати схему порівняння слова з константою;
- синтезувати повну схему порівняння слів.
- синтезувати цифрові автомати



- синтезувати тригерні схеми різноманітних типів на базі асинхронного RS-тригера
- побудувати часову діаграму роботи тригера;
- скласти принципову схему тригера заданого типу;
- синтезувати регістри зсуву;
- на основі математичної моделі побудувати схему реверсивного регістру зсуву;
- синтезувати схеми синхронних лічильників з довільним модулем та порядком лічення;
- побудувати схему синхронного лічильника;
- побудувати часову діаграму роботи синхронного лічильника;
- на основі МДНФ функцій збудження синхронного лічильника та часової діаграми його роботи побудувати схему асинхронного лічильника;
- синтезувати реверсивний лічильник;
- побудувати часову діаграму його роботи;
- побудувати схему реверсивного лічильника;
- обирати тип пам'яті для комп'ютерів та мікропроцесорних систем;
- уміти використовувати регістри загального призначення мікропроцесорів для програмування на мові асемблер.

### 1.5 Організація навчального процесу

Вивчення дисципліни «Комп'ютерна схемотехніка та архітектура комп'ютерів» для студентів заочної форми навчання складається з трьох видів навчальних занять (установчі лекції на початку вивчення; лекційні, лабораторні заняття і по засвоєнню теоретичної частини курсу виконання контрольної роботи та курсового проекту (п. 2.2 та п.2.3).

Контроль самостійної роботи студента заочної форми навчання здійснюється шляхом перевірки контрольної роботи та курсового проекту, які надсилається студентом у встановлені деканатом строки, опитування на лекційних заняттях, захисту лабораторної роботи та курсового проекту і на заходах підсумкового контролю, що передбачені навчальним планом.

## II ОРГАНІЗАЦІЯ САМОСТІЙНОЇ РОБОТИ СТУДЕНТА

### 2.1 Рекомендації по вивченню теоретичного матеріалу та виконанню контрольної роботи

#### 2.1.1 Загальні поради

– зміст кожної теми курсу вивчається за допомогою наведеного у підрозділі 1.3 переліку навчальної та методичної літератури (як основні слід використовувати підручники у списку літератури під номерами [1], [2], [3], та рекомендації до цієї теми;

– якщо Ви вважаєте, що засвоїли зміст теми, що вивчається, то спробуйте відповісти на «Запитання для самоперевірки», наведені у кінці кожної теми. Якщо Ви не можете відповісти на якесь з цих питань, знайдіть відповідь у навчальній літературі [4];

– після того, як Ви переконалися, що зміст теми засвоєно, приступайте до виконання контрольної роботи, що відповідає цій темі;

– якщо у Вас виникли питання або труднощі при виконанні контрольної роботи, то потрібно звернутись до викладача, який читав установчу лекцію, письмово на адресу університету звичайною або електронною поштою: [kaf - infotech@ogmi.farlep.odessa.ua](mailto:kaf-infotech@ogmi.farlep.odessa.ua).

#### 2.1.2 Рекомендації по вивченню 1-ї теми „Вступ. Арифметичні основи цифрових пристроїв ”

Перша тема (с. 7-26 [1], с. 96-132 [2], с. 665- 674 [3]) знайомить з базовими поняттями, дає загальні відомості о системах числення, методах переведення чисел з однієї системи числення в іншу. Арифметики систем числення. Перетворення даних з врахуванням точності. Арифметичні операції над числами.

При вивченні першої теми необхідно звернути увагу на такі базові знання та вміння:

– позиційні системи числення. (с. 7-9 [1], с. 96-98 [2], с. 665-669 [3]);

– двійкова система числення; вісімкова система числення; шістнадцяткова система числення. Арифметики систем числення. (с. 9-13 [1], с. 98-102 [2], с. 670-674 [3]);

– переведення чисел з однієї системи числення в іншу. Частинні методи переведення чисел. (с.14-15 [1], с. 103-110 [2] с. 669-670 [3]);

– загальні методи переведення чисел. Переведення чисел із довільної СЧ в довільну з виконанням арифметичних операцій в новій  $h$  СЧ. (с. 15-16 [1], с. 105 [2]).

– переведення чисел з виконанням операцій в старій СЧ (с.17-26 [1], с. 106-111 [2]);

- переведення чисел з довільної СЧ в довільну з проміжним перетворенням в десяткову СЧ (с. 18-25 [1], с. 111-114 [2]);
- перетворення даних з врахуванням точності (с. 25-26 [1], с. 114-115 [2]).

#### Запитання для самоперевірки 1-ї теми

1. Дайте визначення та назвіть характеристики систем числення.
2. Які існують методи переведення чисел із довільної СЧ в довільну.
3. Коли можна застосовувати часні методи переведення чисел.
4. Правило переведення цілих чисел із довільної СЧ в довільну з виконанням арифметичних операцій в старій СЧ.
5. Правило переведення цілих чисел із довільної СЧ в довільну з виконанням арифметичних операцій в новій СЧ.
6. Правило переведення дробових чисел із довільної СЧ в довільну з виконанням арифметичних операцій в старій СЧ.
7. Правило переведення дробових чисел із довільної СЧ в довільну з виконанням арифметичних операцій в новій СЧ.
8. Коли буде виконуватися умова переведення числа без втрати точності.

Закріплення отриманих при вивченні першої теми знань та вмінь здійснюється за допомогою лабораторних робіт, для виконання яких потрібно застосувати придбані знання та вміння.

#### **2.1.3 Рекомендації по вивченню 2-ї теми „Кодування цифрової інформації”**

Друга тема (с. 37-56 [1], с. 20-21 [2], с. 167-172 [2]) формує у студентів уявлення про кодування цифрової інформації та виконання арифметичних дій у різноманітних кодах. При вивченні другої теми необхідно звернути увагу на такі базові знання та вміння:

- прямий код двійкових чисел. с. 37-38 [1];
- зворотний і додатковий коди. с. 38 [1];
- зв'язок між негативним числом і його зворотним і додатковим кодами. с. 38-39 [1];
- алгебраїчне додавання двійкових чисел, представлених зворотним кодом. с. 40-41 [1], с. 44-49 [1];
- алгебраїчне додавання двійкових чисел, представлених додатковим кодом. с. 41-42 [1], с. 44-49 [1];
- засоби виявлення переповнення розрядної сітки підсумовуючого пристрою процесора с. 43-44[1], с. 20-21 [2];
- коди чисел в інших системах числення. с. 47-49 [1], с. 167-172 [2];
- кодування десяткових чисел. с. 49-52 [1].

## Запитання для самоперевірки 2-ї теми

1. Як утворюється прямий код двійкових чисел.
2. Як утворюється зворотний і додатковий коди двійкових чисел.
3. Зв'язок між зворотним і додатковим кодами.
4. Зв'язок між негативним числом і його зворотним і додатковим кодами.
5. Алгебраїчне додавання двійкових чисел, представлених зворотним кодом.
6. Алгебраїчне додавання двійкових чисел, представлених додатковим кодом.
7. Як виявляють переповнення розрядної сітки.
8. Коди чисел в інших системах числення.
9. Кодування десяткових чисел.

Закріплення отриманих при вивченні другої теми знань та вмій здійснюється за допомогою лабораторної роботи, для виконання якої потрібно застосувати придбані знання та вміння.

### 2.1.4 Рекомендації по вивченню 3-ї теми „Логічні основи цифрових пристроїв перетворення даних”

Третя тема (с. 57-109 [1], с. 74-99 [2], с. 68-114 [3]) знайомить з формами представлення перемикальних функцій, основами алгебри логіки, з правилами мінімізації перемикальних функцій, діаграмами Вейча, неповністю визначеними перемикальними функціями та методами їх мінімізації (методом Квайна-Мак-Класки та за допомогою діаграм Вейча).

При вивченні третьої теми необхідно звернути увагу на такі базові знання та вміння:

- цифрові перемикальні ланцюги (с. 57-59 [1], с. 50-53 [3]);
- перемикальні функції (с. 59-63 [1], с. 41 [2]);
- класи перемикальних функцій. (с. 64-66 [1]);
- закони алгебри логіки (с. 69- 73[1], с. 41-42 [2]);
- нормальні форми представлення перемикальних функцій (с. 73-79 [1], с. 42-47 [2], с. 51-53 [3]);
- мінімізація перемикальних функцій (с. 80- 89[1],с. 47-49 [2]);
- діаграми Вейча (с.90-97[1], с. 49-50 [2], с. 50-57 [3]);
- способи завдання неповністю визначених перемикальних функцій, їх мінімізація (с. 97-102 [1]).

## Запитання для самоперевірки 3-ї теми

1. Як утворюється прямий код двійкових чисел.
2. Як утворюється зворотний і додатковий коди двійкових чисел.

3. Зв'язок між зворотним і додатковим кодами.
4. Зв'язок між негативним числом і його зворотним і додатковим кодами.
5. Алгебраїчне додавання двійкових чисел, представлених зворотним кодом.
6. Алгебраїчне додавання двійкових чисел, представлених додатковим кодом.
7. Як виявляють переповнення розрядної сітки.
8. Коди чисел в інших системах числення.
9. Кодування десяткових чисел.

Закріплення отриманих при вивченні третьої теми знань та вмій здійснюється за допомогою лабораторної роботи, та контрольної роботи для виконання яких потрібно застосувати придбані знання та вміння.

#### **2.1.4 Рекомендації по вивченню 4-ї теми «Аналіз і синтез комбінаційних схем»**

Четверта тема (с. 128-154[1], с. 7-40 [2]) знайомить з правилами синтезу однорозрядного суматора (ОКСМ), способами організації підсумовування багаторозрядних слів, організації переносів в паралельних суматорах. Знайомить з суматорами з послідовними переносами, з паралельними переносами та комбінованими переносами, а також із суматорами з пам'яттю. Також з правилами синтезу цифрових схем порівняння. Розглядається порівняння слів з константами, порівняння слів, порівняння слів за допомогою суматорів та порівняння слів зі знаками.

При вивченні четвертої теми необхідно звернути увагу на такі базові знання та вміння:

- логічні елементи та логічні оператори. ( с. 100-102 [1]);
- оцінка складності комбінаційних схем за Квайном. ( с. 102-107 [1], с. 13-14; [2]);
- описання ОКСМ в різноманітних операційних формах ( с. 107-27 [1], с. 13-14 [2]);
- позначення КСМ на функціональних і структурних схемах (с. 128-130 [1], с. 14 [2]);
- способи підсумовування багаторозрядних слів ( с.130-132 [1], с. 15-16 [2]);
- способи організації міжрозрядних переносів в паралельних КСМ ( с. 132-134 [1], с. 16 [2]) ;
- математичний опис паралельної організації міжрозрядних переносів в багато розрядних КСМ ( с. 132-134 [1], с. 16-19 [2]);
- суматори з комбінованими переносами. ( с. 134-135 [1], с. 16 [2]);
- суматори з пам'яттю. ( с. 135-136 [1], с. 19-20 [2]);

- засоби виявлення переповнення розрядної сітки суматора. (с. 136-138 [1], с. 20-21 [2]);

### **Запитання для самоперевірки 4-ї теми**

1. Дайте визначення комбінаційної схеми
2. Як графічно позначаються логічні елементи?
3. Як оцінити складність комбінаційних схем за Квайном?
4. Яку схему називають двійковим суматором?
5. Як позначаються на схемах комбінаційні суматори?
6. Як здійснюється підсумовування всіх розрядів числа в паралельному КСМ ?
7. За якою схемою будується послідовний суматор?
8. Скільки видів паралельних суматорів розрізняють за характером розповсюдження переносів?
9. Суматори з пам'яттю.
10. Як виявити переповнення суматорів?

Закріплення отриманих при вивченні четвертої теми знань та вмінь здійснюється за допомогою лабораторної роботи та контрольної, для виконання яких потрібно застосувати придбані знання та вміння.

### **2.1.5 Рекомендації по вивченню 5-ї теми « Типові комбінаційні вузли цифрових пристроїв обробки даних »**

П'ята тема (с. 128-187 [1], с. 22-55 [2]) знайомить з типовими комбінаційними вузлами цифрових пристроїв обробки даних. У цій темі розглядаються : комбінаційні схеми порівняння, дешифратори, комбінаційні перетворювачі довільних кодів.

При вивченні п'ятої теми необхідно звернути увагу на такі базові знання та вміння:

- схеми порівняння слів з константами на рівність, нерівність та більше-менше. ( с. 139-142; 151-153 [1], с. 29- 37; 38-40[2]);
- схеми порівняння слів на рівність, нерівність та більше-менше. (с. 142-150 [1], с. 25-29 [2]);
- порівняння слів за допомогою суматора. ( с. 150-151 [1], с. 37-38[2]);
- апаратні перетворювачі двійкових чисел у десяткові й навпаки ( с. 166-170 [1] );
- матричні, каскадні, та пірамідальні дешифратори ( с. 174-179 [1] );
- дешифрування логічних умов повної схеми порівняння слів ( с. 180-181 [1], с. 35 [2]);

## Запитання для самоперевірки 5-ї теми

1. Як спроектувати комбінаційну схему порівняння з константами?
2. Як спроектувати повну комбінаційну схему порівняння слів?
3. Як синтезувати перетворювач з десяткової системи числення у двійкову?
4. Як синтезувати перетворювач з двійкової системи числення у десяткову?
5. Яка схема називається повним дешифратором?
6. Види дешифраторів.
7. Як здійснюється дешифрування логічних умов повної схеми порівняння?

Закріплення отриманих при вивченні п'ятої теми знань та вмінь здійснюється за допомогою контрольної роботи, для виконання якої потрібно застосувати придбані знання та вміння.

### 2.1.6 Рекомендації по вивченню 6-ї теми « Логічне проектування цифрових автоматів ЦА.»

Шоста тема (с. 206-238 [1] ) знайомить з основними теоретичними положеннями й загальними принципами синтезу та проектування цифрових автоматів. У цій темі розглядаються способи завдання абстрактних цифрових автоматів, синтез та проектування їх.

При вивченні шостої теми необхідно звернути увагу на такі базові знання та вміння:

- автомати Мілі й Мура, способи завдання ЦА. (с.206-214 [1] ;
- елементарні автомати з одним входом (Т- та D-тригери). (с.214-216; 238-239 [1], с. 45-47 [2]);
- елементарні автомати із двома входами (DV-тригер, RS-тригер, JK - тригер). (с. 218-222; 240-241 [1], с. 47-49 [2]);
- двотактні тригери (с.245-247 [1]);
- синтез тригерних схем. ( с.223-227; 241-247 [1], с. 49-55 [2] );

## Запитання для самоперевірки 6-ї теми

1. Що таке моделі цифрових пристроїв?
2. Що таке комбінаційні схеми?
3. Які задачі синтезу ЦА?
4. Про що говорить теорема про структурну повноту?
5. Що таке тригер?
6. Який рівень вхідного сигналу називають активним?
7. У чому відмінність між синхронними та асинхронними тригерами?

8. Які умовні графічні позначення тригерів різноманітних типів та їх входів?
9. У чому особливість синтезу тригерних схем різноманітних типів порівняно із загальним методом синтезу структурних автоматів?
10. Які вихідні дані необхідні для синтезу тригерних схем різноманітних типів?
11. Як визначити за таблицею переходів активні рівні вхідних сигналів?
12. Як визначити скільки варіантів СТП може бути побудовано для тригера?
13. Поясніть структуру та роботу двотактного тригера.
14. Як проводиться синтез двотактних тригерів? Чим він відрізняється від синтезу одноктактних тригерів?
15. Наведіть приклади перетворень одного типу тригера у інший.

Закріплення отриманих при вивченні шостої теми знань та вмінь здійснюється за допомогою лабораторної роботи, для виконання якої потрібно застосувати придбані знання та вміння.

### **2.1.7 Рекомендації по вивченню 7-ї теми « Основні елементи і вузли цифрових пристроїв з пам'яттю .»**

Сьома тема (с. 238 - 275[1] с. 41-95 [2]) формує у студентів уявлення про основні вузли цифрової техніки – реєстри зберігання та зсуву, тригерні схеми та лічильники. У цій темі розглядаються синтез тригерних схем різноманітних типів, синтез та дослідження реєстрів зберігання і зсуву та синтез та дослідження схем лічильників з довільним модулем та порядком лічення .

При вивченні сьомої теми необхідно звернути увагу на такі базові знання та вміння:

- тригерні схеми довільних типів с. 238-248 [1]; с.41 – 55 [2]
- накопичувальні реєстри, паралельний та парафазний прийом слів в реєстри. с. 248-250 [1]; с.60 – 64 [2]
- способи видавання слів з реєстрів. с. 250-251 [1];
- зсувові реєстри. с. 252-255 [1]; с.64 – 67 [2]
- синтез реєстрів зсуву 255-259 [1], с. 68 - 73 [2];
- визначення та класифікація лічильників ( с. 260-261 [1], с. 78-79 [2]);
- асинхронні та синхронні лічильники ( с. 261 -264 [1], с. 79 - 88 [2]);
- двійково-десяткові коди та їх властивості ( с. 264-266 [1]);
- синтез простих синхронних двійково-десяткових лічильників з довільним порядком лічення ( с.267-269 [1]; с. 82-88 [2]);
- синтез асинхронних лічильників (с. 270- 275[1], с. 88- 93 [2]);



– синтез реверсивних синхронних лічильників з довільними модулем та порядком лічення (с. 275 - 278 [1], с. 94-95 [2]);

### Запитання для самоперевірки 7-ї теми

1. Які функції (мікрооперації) можуть виконувати регістри?
2. Функціональна класифікація регістрів.
3. Наведіть схеми регістрів зберігання з паралельним та з парафазним прийомом інформації. Поясніть їх роботу.
4. В чому відміна замкненого в кільце регістра зсуву від розімкненого?
5. Як здійснюється зсув за допомогою “навскісної передачі” слів?
6. Поясніть роботу нереверсивних регістрів зсуву на двотактних тригерах різноманітних типів.
7. Методологія та алгоритм синтезу реверсивних багатофункціональних регістрів зсуву.
8. Модуль та основні характеристики лічильника, що визначають його швидкодію.
9. Функціональна класифікація лічильних схем.
10. Пояснити роботу підсумовуючих двійкових АЛЧ на синхронних та асинхронних тригерах. Часова діаграма (ЧД) роботи ЛЧ.
11. Поясніть методіку проектування СЛЧ з довільними модулем та порядком рахування.
12. Організація перенесень між десятковими розрядами двійково-десятькового СЛЧ.

Закріплення отриманих при вивченні сьомої теми знань та вмій здійснюється за допомогою курсового проекту для виконання якого потрібно застосувати придбані знання та вміння.

### 2.1.8 Рекомендації по вивченню 8-ї теми « Архітектура комп'ютерів.»

Восьма тема (с. 238 - 275[3] ) знайомить студентів з основними компонентами комп'ютера: процесорами, пам'яттю , пристроями вводу-виводу даних. Дає стислий опис системної архітектури. Розглядаються різновиди паралельних комп'ютерів: суперкомп'ютери на основі декілька комп'ютерів з сумісною пам'яттю та мережі робочих станцій а також структури мікропроцесорних систем. RISC- та CISC-процесори.

При вивченні восьмої теми необхідно звернути увагу на такі базові знання та вміння:

- цифрові комп'ютери с. 29-54 [3]
- запам'ятовуючі пристрої. Flash-пам'ять.с. 69-105 [3]

- процесори. с. 56-64 [3];
- суперкомп'ютери. Паралельні обчислювальні системи с.65-73 [3], с. 557-643 [3];
- універсальні мікропроцесори. Схеми підтримки МП на системних платах ( с. 230-330 [3];
- структури мікропроцесорних систем ( с. 139 -183 [3]);
- RISC- та CISC-процесори ( с. 62-64 [3]);

### **Запитання для самоперевірки 8-ї теми**

1. Які існують типи комп'ютерів ?
2. В чому полягає закон Мура ?
3. У чому полягає відмінність трансляції та інтерпретації ?
4. Які складові частини процесора?
5. Що таке суперскалярна архітектура ?
6. Що називається мультипроцесором?
7. Які існують різновиди пам'яті комп'ютерів?
8. Які складові пристрою ввода- вивода ?
9. Що таке конвеєрні обчислення?
10. У чому переваги та недоліки RISC- та CISC-процесорів.

Закріплення отриманих при вивченні восьмої теми знань та вмій здійснюється за допомогою лабораторної роботи та курсового проекту для виконання яких потрібно застосувати придбані знання та вміння.

## **2.2 Перелік завдань на контрольну роботу**

### **2.2.1 Загальні поради по виконанню контрольної роботи**

1. За допомогою навчальної та методичної літератури, список якої наведено у попередній частині цих Методичних указівок, та рекомендацій, які сформовані у п.2.1, необхідно вивчити зміст теоретичної частини. Самоперевірка знань здійснюється за допомогою „Запитань для самоперевірки ” , які наводяться наприкінці рекомендацій по вивченню кожної теми.

2. Після засвоєння теоретичного матеріалу необхідно виконати контрольну роботу.

3. Надіслати виконану та оформлену за установленими деканатом заочного факультету вимогами контрольну роботу до університету на перевірку та рецензію до контрольної дати, яка також установлюється деканатом.

У п. 2.2.2 наведені 20 варіантів контрольних завдань. Вибір варіанту визначається роком в якому видається завдання, якщо рік парний , то варіант вибирається з перших десятиох, якщо непарний то з 11 по20

варіанти, а також останньою цифрою номера залікової книжки. Наприклад № залікової книжки закінчується 7-кою, а рік 2015 – то варіант 17, якщо рік 2016 – то варіант 7.

Контрольна робота виконується на папері формату А4.

### 2.2.2 Перелік завдань контрольної роботи

В контрольній роботі необхідно виконати синтез та дослідження цифрових схем порівняння.

Контрольна робота складається з теоретичної та практичної частини. В першій частині контрольної роботи (вона однакова для усіх варіантів і максимальна оцінка за цю частину 15 балів) необхідно письмово надати відповіді на наступні запитання:

1. Математичний опис повних і неповних *КСП* слів у загальному випадку.
2. Порівняння слів із константами на рівність-нерівність.
3. Як спроектувати *КСП* слів із константами на більше-менше?
4. Порівняння слів на рівність-нерівність.
5. Перетворіть вираз (3.11) у 7-у *ОФ* та розробіть однорозрядну *КСП* (*ОКСП*) на логічних елементах *I-АБО-НІ*.
6. Спроектуйте неповну *ОКСП* за 2-ою *ОФ* функції для виконання логічної умови  $X \leq Y$ .
7. Спроектуйте неповну *ОКСП* за 2-ою *ОФ* функції для виконання логічної умови  $X > Y$ .
8. Спростіть функцію переносу із молодшого розряду неповної *ОКСП* слів на менше, якщо умова  $X < Y$  кодується нулем.
9. Поясніть методику проектування повної *ОКСП* слів.
10. Порівняння слів за допомогою суматора.
11. Структура та робота повної *КСП* слів зі знаками.

В практичній частині роботи (яка оцінюється у 20 балів ) необхідно:

1. Синтезувати за заданим варіантом (табл. 3.1) принципову *КСП* із константою для заданої логічної умови, реалізувати схему на *ЛЕ I-НІ*.
2. Розробити принципову схему для повного порівняння дворозрядних слів згідно з кодами сигналів переносу (*СП*), заданих в табл. 3.1. (в 2-ій *ОФ*). Для одержання мінімальної системи функцій переносу  $p_i'$  та  $p_i''$  використати операцію винесення змінних  $z_i'$  та  $z_i''$  за дужки. Мінімізувати структуру молодшої *ОКСП*. Провести синтез неповного вихідного дешифратора умов.
3. Розробити за заданим варіантом функціональну схему неповного порівняння чотирирозрядних слів на базі *КСМ*.

Таблиця 2.1 – Таблиця варіантів

Номер варіанта	КСП слів с константою		Коди сигналів переносу повної КСП			Порівняння слів на КСМ	
	ЛУ	$K_{(10)}$	$x_i = y_i$	$x_i < y_i$	$x_i > y_i$	ЛУ	$p_0$
1.	$X=K$	12	00	01	10	$X<Y$	0
2.	$X\neq K$	12	00	10	01	$X<Y$	1
3.	$X<K$	10	01	00	10	$X\leq Y$	0
4.	$X\leq K$	10	01	10	00	$X\leq Y$	1
5.	$X>K$	11	10	00	01	$X>Y$	0
6.	$X\geq K$	11	10	01	00	$X>Y$	1
7.	$X=K$	13	00	01	11	$X\geq Y$	0
8.	$X\neq K$	13	00	11	01	$X\geq Y$	1
9.	$X<K$	11	01	00	11	$X<Y$	0
10.	$X\leq K$	11	01	11	00	$X<Y$	1
11.	$X>K$	12	11	00	01	$X\leq Y$	0
12.	$X\geq K$	12	11	01	00	$X\leq Y$	1
13.	$X=K$	14	00	10	11	$X>Y$	0
14.	$X\neq K$	14	00	11	10	$X>Y$	1
15.	$X<K$	12	10	00	11	$X\geq Y$	0
16.	$X\leq K$	12	10	11	00	$X\geq Y$	1
17.	$X>K$	13	11	00	10	$X<Y$	0
18.	$X=K$	11	01	10	11	$X\leq Y$	0
19.	$X\neq K$	11	01	11	10	$X\leq Y$	1
20.	$X\geq K$	13	11	10	00	$X<Y$	1

## Основні теоретичні положення

### Вступ

В цифрових пристроях (ЦП) використовується декілька різновидів операції порівняння слів: за модулем, з урахуванням знаків операндів, порівняння порядків слів, заданих у напівлוגарифмічній формі. Тут під словом слід розуміти групу двійкових розрядів, що спільно піддаються обробці при арифметичній чи логічній операції (у окремому випадку, багаторозрядне двійкове число). Найбільш повною є операція порівняння, за якою встановлюється факт виконання однієї із умов  $X = Y$ ,  $X < Y$ ,  $X > Y$ , де  $X$ ,  $Y$  – цілі чи дробові двійкові числа зі знаками.

Найбільш розповсюджені два способи реалізації операції порівняння. Перший полягає в тому, що із одного слова віднімають друге та за знаком і різницею судять про виконання наведених умов. Реалізується така операція на суматорі, до якого додатково підключається комбінаційна схема для фіксації нульового результату.

З метою скорочування часу реалізації операції порівняння та розвантаження суматора ЦП для виконання інших операцій використовується інший спосіб виконання порівняння слів – побудова спеціальної комбінаційної схеми порівняння (КСП).

Повною схемою порівняння  $n$ -розрядних слів у загальному випадку називають схему з  $(2n+2)$  входами та трьома виходами, що реалізує такі перемикальні функції:

$$\begin{aligned} f_{x=y} &= \begin{cases} 1(0) & \text{при } X=Y, \\ 0(1) & \text{при } X \neq Y; \end{cases} \\ f_{x < y} &= \begin{cases} 1(0) & \text{при } X < Y, \\ 0(1) & \text{при } X \geq Y; \end{cases} \\ f_{x > y} &= \begin{cases} 1(0) & \text{при } X > Y, \\ 0(1) & \text{при } X \leq Y. \end{cases} \end{aligned} \quad (2.1)$$

Цифри у дужках означають, що виконання відповідної логічної умови може кодуватися одиницею чи нулем згідно з вимогами завдання, що розв'язується.

Кожну функцію у повній схемі порівняння можна реалізувати незалежно. Однак, якщо схеми для  $f_{x=y}$  та  $f_{x < y}$  вже збудовані, то функцію  $f_{x > y}$  простіше реалізувати за формулою

$$f_{x > y} = \overline{f_{x=y}} \cdot \overline{f_{x < y}} = \overline{f_{x=y} \vee f_{x < y}}.$$

Аналогічно можна одержати значення будь-якої із функцій, виконаних схемою порівняння, при відомих двох інших:

$$\begin{aligned} f_{x=y} &= \overline{f_{x < y}} \cdot \overline{f_{x > y}} = \overline{f_{x < y} \vee f_{x > y}}, \\ f_{x < y} &= \overline{f_{x=y}} \cdot \overline{f_{x > y}} = \overline{f_{x=y} \vee f_{x > y}}. \end{aligned}$$

У окремому випадку схема порівняння може бути неповною, тобто реалізувати тільки одне чи два із наведених умов.

### Порівняння слів з константами

Порівняння на рівність. Нехай  $X = x_0 x_1 \dots x_{n-1}$  та  $K = k_0 k_1 \dots k_{n-1}$  відповідно  $n$ -розрядні слово та константа, що підлягають порівнянню. Необхідно спроектувати КСП, що реалізує деяку перемикальну функцію  $f_{x=k}$ , що набирає значення 1 при рівності слова та константи і значення 0 при їх нерівності:

$$f_{x=k} = \begin{cases} 1 & \text{при } X=K, \\ 0 & \text{при } X \neq K. \end{cases} \quad (2.2)$$

З умови завдання випливає, що КСП повинна описуватися конституентною одиницею, що відповідає двійковому набору, який визначається константою  $K$  (за визначенням, тільки на цьому наборі

конституента одиниці буде дорівнювати 1). Отже,  $f_{x=k} = \tilde{x}_0 \tilde{x}_1 \dots \tilde{x}_{n-1}$ , де  $\tilde{x}_i = 1$ , якщо  $k_i = 1$ , та  $\tilde{x}_i = 0$ , якщо  $k_i = 0$ .

Наприклад, при порівнянні на рівність константам  $K_0 = 0000_{(2)}$  та  $K_{10} = 1010_{(2)}$  відповідають такі функції:  $f_{x=0} = \bar{x}_0 \bar{x}_1 \bar{x}_2 \bar{x}_3$  та  $f_{x=10} = x_0 \bar{x}_1 x_2 \bar{x}_3$ , за якими і реалізуються необхідні схеми порівняння.

Порівняння на нерівність. Якщо умова рівності кодується нулем (а нерівності – одиницею), то КСП описується перемикальною функцією

$$f_{x \neq k} = \begin{cases} 0 & \text{при } X=K, \\ 1 & \text{при } X \neq K, \end{cases} \quad (2.3)$$

із якої випливає, що надана функція дорівнює нулю на одному єдиному наборі змінних  $x_i$  ( $i = 0, 1, \dots, n-1$ ) та одиниці – на усіх інших наборах.

Така функція уявляє собою конституенту нуля, що відповідає двійковому набору, який задається константою  $K$ , тобто  $f_{x \neq k} = \tilde{x}_0 \vee \tilde{x}_1 \vee \dots \vee \tilde{x}_{n-1}$ , де  $\tilde{x}_i = 1$ , коли  $k_i = 0$  та  $\tilde{x}_i = 0$ , коли  $k_i = 1$ .

Наприклад, тим самим константам  $K_0 = 0000_{(2)}$  та  $K_{10} = 1010_{(2)}$  при кодуванні умови рівності нулем відповідають перемикальні функції, за якими реалізовані КСП на рис. 2.1:

$$f_{x \neq 0} = x_0 \vee x_1 \vee x_2 \vee x_3 = \overline{\bar{x}_0 \bar{x}_1 \bar{x}_2 \bar{x}_3};$$

$$f_{x \neq 10} = \bar{x}_0 \vee x_1 \vee \bar{x}_2 \vee x_3 = \overline{x_0 \bar{x}_1 x_2 \bar{x}_3}.$$

Розглянуті КСП можна інтерпретувати як неповні дешифратори з одним виходом. Такі схеми широко застосовують у мікропроцесорних пристроях у вигляді дешифраторів адрес зовнішніх пристроїв (ЗП). Кожному ЗП привласнюється конкретний код адреси із адресного простору мікропроцесора (МП). Пристрій включається у роботу тільки в тому випадку, якщо на його спеціальному розв'язуючому вході вибірки кристалу (ВК), ввімкнутому до виходу дешифратора адреси, встановлюється активний нульовий рівень сигналу. Дешифратор адреси, входи якого з'єднані з адресною шиною МП, безупинно декодує адреси, що генеруються МП, та, при впізнанні коду "своєї" адреси, видає нульовий рівень вихідного сигналу, активізуючи тим самим ЗП.



Рисунок 2.1 – КСП з константами на нерівність

Інколи необхідно порівняти на рівність слово з деякими константами, тобто реалізувати наступну перемикальну функцію

$$f_{x=k} = \begin{cases} 1 & \text{при } X = K_0 \vee K_1 \vee \dots \vee K_m, \\ 0 & \text{при } X \neq K_0 \vee K_1 \vee \dots \vee K_m, \end{cases} \quad (2.4)$$

$$m = 0, 1, \dots$$

Очевидно, що досконала диз'юнктивна нормальна форма (ДДНФ) функції в даному випадку являє собою диз'юнкцію конститuent одиниці, відповідних наборам  $K_0, K_1, \dots, K_m$ . Ця функція мінімізується відомими засобами.

Наприклад, логічна умова  $X = K_8 \vee K_9 \vee K_{12} \vee K_{13}$ , виконання якої кодується одиницею, описується ДДНФ функції  $f_{x=k} = \vee (8, 9, 12, 13)$ . Мінімальна форма (МДНФ), одержана за допомогою діаграми Вейча чотирьох змінних,  $f_{x=k} = x_0 \bar{x}_2$ .

Порівняння з константами на більше-менше. Нехай необхідно спроектувати КСП, що виявляє вхідні слова, величина яких менша за наперед задане число (набору). Таку схему описують такою перемикальною функцією:

$$f_{x < k} = \begin{cases} 1 & \text{при } X < K_m, \\ 0 & \text{при } X \geq K_m, \end{cases}$$

$$m = 0, 1, \dots, 2^n - 1,$$

де  $m$  – десятковий номер набору;  $n$  – число розрядів у слові. Нагадаємо, що число різноманітних наборів від  $n$  змінних дорівнює  $2^n$ .

Логічна умова  $X < K_m$  означає, що функція  $f_{x < k}$  набирає одиничного значення на вхідних наборах, що дорівнюють константам  $K_0, K_1, \dots, K_{m-1}$ , та нульового значення на всіх інших наборах, що дорівнюють константам  $K_m, K_{m+1}, \dots, K_{2^n-1}$ . Отже, у досконалій формі функція являє собою диз'юнкцію конститuent 1, відповідних двійковим наборам констант  $K_0, K_1, \dots, K_{m-1}$ .

Наприклад, логічній умові  $X < K_{12}$  ( $K_{12} = 1100_{(2)}$ ) у наданому завданні відповідає ДДНФ функції  $f_{x < 12} = \vee (0, 1, \dots, 9, 10, 11)$ . Використовуючи діаграму Вейча чотирьох змінних (рис. 3.2,а), отримуємо МДНФ цієї функції  $f_{x < 12} = \bar{x}_0 \vee \bar{x}_1 = \overline{x_0 x_1}$ , що реалізується одним логічним елементом (ЛЕ) І-НІ (рис. 2.2,б).

КСП, що реалізують логічні умови  $X \leq K_m, X > K_m, X \geq K_m$ , проектуються аналогічно. Помітимо тільки, якщо нерівність нежорстка ( $X \leq K_m$  або  $X \geq K_m$ ), то ДДНФ функцій включають до себе і конститuentу одиниці, відповідну набору  $K_m$ . Наприклад, логічній умові  $X \leq K_{12}$  відповідає функція досконалої форми

$$f_{x < 12} = \vee (0, 1, \dots, 9, 10, 11, 12).$$

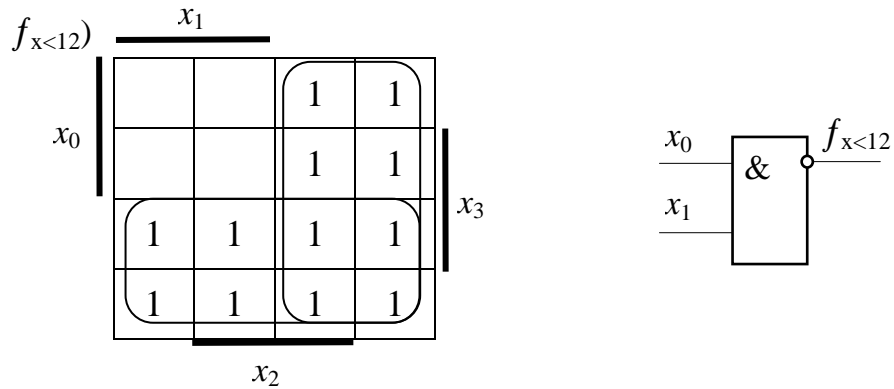


Рисунок 2.2 – Синтез КСП з константою на менше

### Порівняння слів на рівність і нерівність

Комбінаційну схему, що реалізує перемикальну функцію вигляду

$$f_{x=y} = \begin{cases} 1 & \text{при } X=Y, \\ 0 & \text{при } X \neq Y, \end{cases} \quad (2.6)$$

називають схемою порівняння багаторозрядних слів  $X$  та  $Y$  на рівність (схемою рівності слів), що проектується таким чином. Багаторозрядні слова  $X$  та  $Y$  рівні тільки в тому випадку, якщо одночасно будуть дорівнювати всі їх однойменні розряди  $x_i$  та  $y_i$  ( $i = 0, 1, \dots, n-1$ ), кожний з яких порівнюється незалежно від інших. Тому достатньо провести синтез однієї однорозрядної КСП (ОКСП), щоб розробити структуру схеми рівності слів довільної розрядності, що є однорідною. Робота ОКСП описується таблицею істинності (див. табл. 2.2), з якої отримуємо функцію рівності (еквівалентності) однойменних розрядів  $x_i$  і  $y_i$  у 1-ій та 2-ій ОФ:

$$r_i = x_i \equiv y_i = \overline{\overline{x_i y_i} \vee x_i \overline{y_i}} = \overline{\overline{x_i} \overline{y_i} \cdot x_i y_i}.$$

Функцію  $f_{x=y}$  рівності багаторозрядних слів  $X$  та  $Y$  одержимо з обліком того, що останнє можливо, якщо дотримується рівність всіх однойменних розрядів слів, тобто  $f_{x=y} = r_0 \cdot r_1 \cdot \dots \cdot r_{n-1}$ .

Перемикальна функція

$$f_{x \neq y} = \begin{cases} 0 & \text{при } X=Y \\ 1 & \text{при } X \neq Y \end{cases} \quad (2.7)$$

у якій умова рівності кодується нулем, описує КСП багаторозрядних слів на нерівність. Функція нерівності  $g_i$  однойменних розрядів  $x_i$  та  $y_i$ , одержана із табл. 3.2, визначається наступним виразом

$$g_i = \overline{r_i} = x_i \oplus y_i = \overline{\overline{x_i y_i} \vee x_i \overline{y_i}} = \overline{\overline{x_i} \overline{y_i} \cdot x_i y_i},$$



де  $\oplus$  – знак операції нерівнозначності або додавання за модулем два. Багаторозрядні слова  $X$  і  $Y$  не рівні, якщо хоч би у одному їх однойменному розряді спостерігається нерівність, тобто функція  $f_{x \neq y}$  нерівності багаторозрядних слів має вид  $f_{x \neq y} = g_0 \vee g_1 \vee \dots \vee g_{n-1}$ . Помітимо, що функції  $f_{x=y}$  та  $f_{x \neq y}$  інверсні, тобто  $f_{x=y} = \overline{f_{x \neq y}}$ , тому вони обидві легко реалізуються на практиці одночасно простим інвертуванням однієї з них, вже збудованої.

На рис. 2.3 показано КСП слів на рівність та нерівність дворозрядних слів  $X$  і  $Y$ , реалізована за виразом

$$f_{x=y} = \overline{g_0 \vee g_1} = \overline{\overline{g_0} \cdot \overline{g_1}} = \overline{\overline{r_0} \cdot \overline{r_1}}.$$

Складність комбінаційних схем за Квайном визначають сумарним числом входів усіх ЛЕ. Ціна по Квайну багаторозрядних КСП слів на рівність-нерівність, реалізованих за різноманітними операторними формами функцій, дорівнює  $7n$  (рис. 2.3).

Таблиця 2.2 – ТТ КСП

$x_i$	0	0	1	1
$y_i$	0	1	0	1
$r_i$	1	0	0	1
$g_i$	0	1	1	0

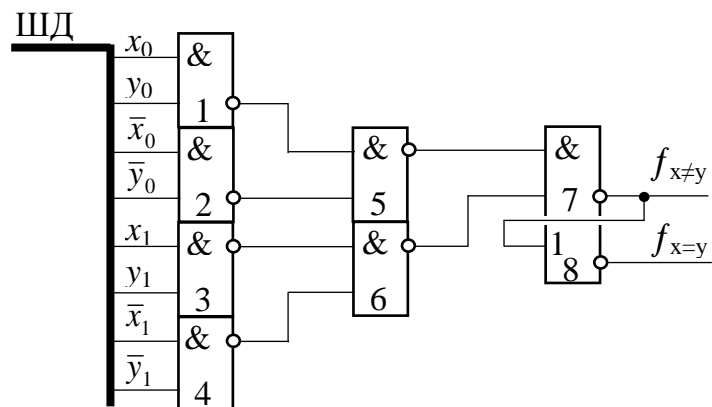


Рис. 2.3 – КСП слів на рівність та нерівність

### Порівняння слів на більше-менше

Порівнянням слів на більше-менше називається знаходження значень логічних умов  $X < Y$ ,  $X \leq Y$ ,  $X > Y$ ,  $X \geq Y$ . Схемна реалізація таких умов здійснюється складніше, ніж схем порівняння на рівність, які будуються із сукупності ОКСП, не зв'язаних між собою, і реалізують функцію логічної рівнозначності.

Розглянемо порівняння багаторозрядних слів на менше, що описується перемикальною функцією

$$f_{x < y} = \begin{cases} 1 & \text{при } X < Y, \\ 0 & \text{при } X \geq Y. \end{cases} \quad (2.8)$$

Так як вага старшого розряду слова у будь-якій позиційній системі числення на одиницю більше за суму ваг усіх його молодших розрядів, то в разі нерівності старших розрядів слів, що порівнюються, достатньо оцінити співвідношення цих розрядів, щоб з'ясувати співвідношення слів у цілому. Однак старші розряди слів можуть опинитися рівними і тоді для вияву співвідношення слів необхідно аналізувати сусідні молодші розряди. У загальному випадку необхідний аналіз усіх розрядів слів, що порівнюються. Сумма весов

Справді, функція  $f_{x<y} = 1$  в тому випадку, коли  $x_0 < y_0$ , або коли  $x_0 = y_0$ , але  $x_1 < y_1$ , або ж коли  $x_0 = y_0$ ,  $x_1 = y_1$ , але  $x_2 < y_2$  і так далі. Звідси, наприклад, випливає, що функція  $f_{x<y}$  для чотирирозрядних слів може бути записана у такому вигляді (виконання логічної умови  $X < Y$  кодується одиницею):

$$\begin{aligned}
 f_{x<y} &= \bar{x}_0 y_0 \vee (x_0 \equiv y_0) \bar{x}_1 y_1 \vee (x_0 \equiv y_0)(x_1 \equiv y_1) \bar{x}_2 y_2 \vee \\
 &\quad \vee (x_0 \equiv y_0)(x_1 \equiv y_1)(x_2 \equiv y_2) \bar{x}_3 y_3 = \quad (2.9) \\
 &= \bar{x}_0 y_0 \vee (x_0 \equiv y_0) \{ \bar{x}_1 y_1 \vee (x_1 \equiv y_1) [ \bar{x}_2 y_2 \vee (x_2 \equiv y_2) \bar{x}_3 y_3 ] \},
 \end{aligned}$$

де  $x_i \equiv y_i = x_i y_i \vee \bar{x}_i \bar{y}_i$  ( $i = 0, 1$ ) – функція рівнозначності (рівності)  $i$ -х розрядів слів.

Чотирирозрядна КСП, що реалізує функцію  $f_{x<y}$  в 1-ій ОФ, зображена на рис. 2.4. Таким чином, для вияву співвідношень більше-менше  $n$ -розрядних слів необхідно  $n-1$  разів виконати операцію рівнозначності. Використовуючи закон алгебри логіки

$$x \vee \bar{x}B = x \vee B \text{ (або інакше } xA \vee \bar{x}AB = xA \vee AB),$$

де  $A, B$  – довільні логічні вирази, функцію (3.9) можна спростити заміною операції рівнозначності  $(x_i y_i \vee \bar{x}_i \bar{y}_i)$  на імплікацію від  $x$  до  $y$  (в диз'юнктивній формі –  $\bar{x}_i \vee y_i$ ) та звести її до вигляду (докажіть самостійно):

$$f_{x<y} = \bar{x}_0 y_0 \vee (\bar{x}_0 \vee y_0)(\bar{x}_1 y_1 \vee (\bar{x}_1 \vee y_1)(\bar{x}_2 y_2 \vee (\bar{x}_2 \vee y_2) \bar{x}_3 y_3)). \quad (2.10)$$

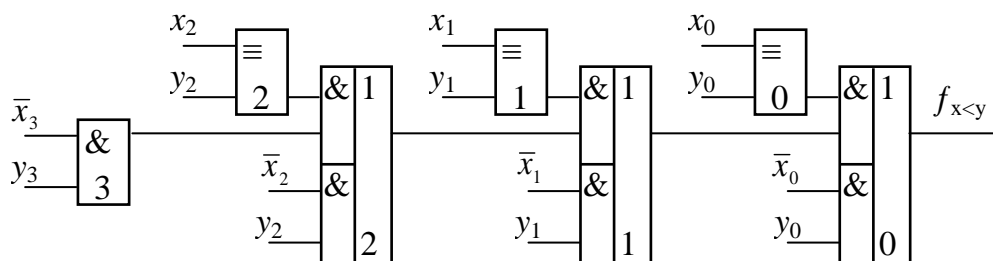


Рис. 2.4 – КСП слів на менше

Аналіз виразів (2.9) та (2.10) дозволяє зробити вивід про те, що багаторозрядна *КСП* на більше-менше може складатися з однотипних *ОКСП*, зв'язаних між собою ланцюгами передачі інформації про співвідношення молодших розрядів слів (рис. 2.5).

Слова у таких схемах аналізуються порозрядно послідовно, починаючи з молодших розрядів. На входи *i*-ої *ОКСП* має подаватися, крім значень *i*-х розрядів слів *X* та *Y*, також сигнал з виходу сусідньої молодшої (*i+1*)-ої *ОКСП*, який за аналогією з суматорами назвемо сигналом переносу  $z_i = p_{i+1}$ . Значення цього сигналу характеризує співвідношення молодших частин слів, що порівнюються і однозначно визначає результат порівняння *i*-х розрядів, а отже, і частин слів від молодшого розряду до наданого. Виходом схеми є сигнал переносу із старшої *ОКСП*.

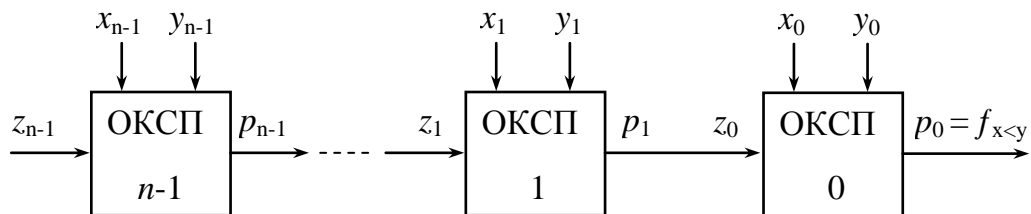


Рисунок 2.5 – Узагальнена структурна схема багаторозрядної *КСП*

Для проектування багаторозрядної *КСП* однорідної структури достатньо розробити всього лише одну *ОКСП*. Для цього спершу виконується етап кодування сигналів переносу, результати якого заносяться у таблицю кодів сигналів переносу – *ТКСП* (табл. 3.3). Сигнали переносу для функції (3.8) повинні відбивати тільки дві логічні умови  $x_i < y_i$  та  $x_i \geq y_i$ , що можуть бути закодовані значеннями однієї двійкової змінної – вхідної  $z_i$  та вихідної  $p_i$ .

Таблиця 2.3 – *ТКСП*

ЛУ \ СП	$z_i, p_i$
$x_i < y_i$	1
$x_i \geq y_i$	0

У *ТКСП* виконання логічної умови  $X < Y$  закодовано одиницею, проте можливий інверсний варіант кодування. При цьому сигнали  $z_i$  та  $p_i$  кодуються завжди однаково.

Збудуємо таблицю істинності згідно з принципом роботи *ОКСП* і кодуванням сигналів переносу (табл. 2.4).

Порядок заповнення таблиці істинності наступний: якщо на визначеному вхідному наборі розряди  $x_i$  та  $y_i$  дорівнюють, то вхідний сигнал переносу передається без змін на вихід *ОКСП*, тобто  $p_i = z_i$  (наприклад, набори 1, 6); при нерівності наданих розрядів значення функції  $p_i$  визначається тільки їх закодованим співвідношенням – без обліку сигналу переносу  $z_i$  (наприклад, набори 3, 5).

ДДНФ, витягнута з  $TI$ , має вигляд

$$p_i = \bar{x}_i \bar{y}_i z_i \vee \bar{x}_i y_i \bar{z}_i \vee \bar{x}_i y_i z_i \vee x_i y_i z_i = \vee(1, 2, 3, 7)$$

Таблиця 2.4 –  $TI$  ОКСП

$x_i$	0	0	0	0	1	1	1	1
$y_i$	0	0	1	1	0	0	1	1
$z_i$	0	1	0	1	0	1	0	1
$p_i$	0	1	1	1	0	0	0	1

Мінімізуючи функцію за допомогою діаграми Вейча трьох змінних, отримуємо:

$$p_i = \bar{x}_i y_i \vee \bar{x}_i z_i \vee y_i z_i = \bar{x}_i y_i \vee (\bar{x}_i \vee y_i) z_i. \quad (2.11)$$

Значення переносів  $z_i$  у кожну ОКСП, крім молодшої, визначають конкретним співвідношенням слів, що порівнюються. Перенос у молодший розряд  $z_{n-1}$  задається кодом рівності (див. табл. 2.3) і від величин слів, що порівнюються, не залежить. Відоме значення переносу у молодший розряд, визначене на етапі кодування, дозволяє спростити структуру молодшої ОКСП. Для наданого прикладу значення переносів при рівності кодується нулем (див. табл. 3.3), тобто  $z_{n-1} = 0$ . Підставивши це значення у (2.11), спростимо функцію переносу  $p_{n-1}$  у молодший розряд:

$$p_{n-1} = \bar{x}_{n-1} y_{n-1} \vee (\bar{x}_{n-1} \vee y_{n-1}) z_{n-1} = \bar{x}_{n-1} y_{n-1} \vee (\bar{x}_{n-1} \vee y_{n-1}) \cdot 0 = \bar{x}_{n-1} y_{n-1}.$$

Отже, трирозрядна КСП на менше, у якій виконання логічної умови кодується одиницею, описується наступною системою перемикальних функцій, що перетворені у другу ОФ:

$$\begin{aligned} p_2 &= \bar{x}_2 y_2 = \overline{\overline{\bar{x}_2 y_2}}, \\ p_1 &= \bar{x}_1 y_1 \vee (\bar{x}_1 \vee y_1) p_2 = \overline{\overline{\overline{\bar{x}_1 y_1 \vee (\bar{x}_1 \vee y_1) p_2}}} = \overline{\overline{\bar{x}_1 y_1 \cdot x_1 \bar{y}_1 \cdot p_2}}, \\ p_0 &= \bar{x}_0 y_0 \vee (\bar{x}_0 \vee y_0) p_1 = \overline{\overline{\overline{\bar{x}_0 y_0 \vee (\bar{x}_0 \vee y_0) p_1}}} = \overline{\overline{\bar{x}_0 y_0 \cdot x_0 \bar{y}_0 \cdot p_1}}. \end{aligned} \quad (2.12)$$

Принципова схема трирозрядної КСП, що реалізована за виразами (2.12), показана на рис. 2.6.

Багаторозрядні КСП, що виконують інші логічні умови ( $X \leq Y$ ,  $X > Y$ ,  $X \geq Y$ ), проектуються аналогічно розглянутій вище методиці.

### Синтез повної схеми порівняння слів

Найбільш складною є повна схема порівняння, що реалізує одночасно всю систему функцій (див. формулу (2.1)). Кожна ОКСП в ній має

приймати і видавати одну з трьох умов (рівно, менше, більше), кодування яких провадиться двома двійковими змінними – вхідними  $z_i', z_i''$  та вихідними  $p_i', p_i''$  (рис. 3.7).

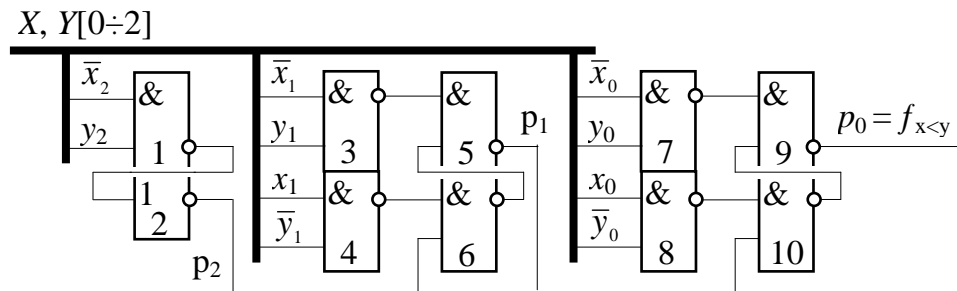


Рисунок 2.6 – Принципова КСП трирозрядних слів

Можливе порозрядне кодування логічних умов наведено у табл. 2.5 (див. перший варіант табл. 2.1). Видно, що набір 11 не використовується для кодування. Це має призвести до появи заборонених наборів, вільне до визначення яких допоможе спростити неповністю визначені функції  $p_i'$  і  $p_i''$  при їх мінімізації.

Згідно з кодами сигналів переносу та принципом роботи ОКСП збудуємо таблицю істинності (табл. 2.6), заповнення якої проводять за правилом, сформульованим вище для неповної схеми порівняння. Набори (3, 7, 11, 15), відповідні конституентам одиниці  $\tilde{x}_i, \tilde{y}_i, z_i', z_i''$ , є забороненими. Досконалі форми неповністю визначених функцій  $p_i'$  та  $p_i''$ , що витягнуті з ТІ, мають вигляд

$$p_i' = \vee (2, 8, 9, 10, 14) = \& (0, 1, 4, 5, 6, 12, 13),$$

$$p_i'' = \vee (1, 4, 5, 6, 13) = \& (0, 2, 8, 9, 10, 12, 14).$$

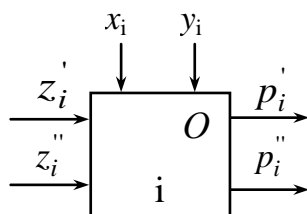


Рисунок 2.7 – Повна ОКСП слів

Таблиця 2.5 – ТКСП повної ОКСП

ЛУ \ СП	$z_i', p_i'$	$z_i'', p_i''$
$x_i = y_i$	0	0
$x_i < y_i$	0	1
$x_i > y_i$	1	0

Таблиця 2.6 – ТІ ОКСП

$x_i$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
$y_i$	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
$z_i'$	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
$z_i''$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$p_i'$	0	0	1	–	0	0	0	–	1	1	1	–	0	0	1	–
$p_i''$	0	1	0	–	1	1	1	–	0	0	0	–	0	1	0	–

Проведемо сумісну мінімізацію одержаних виразів за допомогою діаграм Вейча чотирьох змінних (рис. 2.8), довизначаючи значення функцій на заборонених наборах та виносячи змінні за дужки. Внаслідок одержимо мінімальні форми функцій

$$p_i' = x_i \bar{y}_i \vee x_i z_i' \vee \bar{y}_i z_i' = x_i \bar{y}_i \vee (x_i \vee \bar{y}_i) z_i',$$

$$p_i'' = \bar{x}_i y_i \vee \bar{x}_i z_i'' \vee y_i z_i'' = \bar{x}_i y_i \vee (\bar{x}_i \vee y_i) z_i''.$$

Враховуючи, що умова рівності закодована нульовими значеннями сигналів переносу, тобто  $z_{n-1}' = z_{n-1}'' = 0$  (табл. 2.5), спростимо структуру молодшої ОКСП:

$$p_{n-1}' = x_{n-1} \bar{y}_{n-1} \vee (x_{n-1} \vee \bar{y}_{n-1}) z_{n-1}' = x_{n-1} \bar{y}_{n-1} \vee (x_{n-1} \vee \bar{y}_{n-1}) \cdot 0 = x_{n-1} \bar{y}_{n-1},$$

$$p_{n-1}'' = \bar{x}_{n-1} y_{n-1} \vee (\bar{x}_{n-1} \vee y_{n-1}) z_{n-1}'' = \bar{x}_{n-1} y_{n-1} \vee (\bar{x}_{n-1} \vee y_{n-1}) \cdot 0 = \bar{x}_{n-1} y_{n-1}.$$

Закодовані значення функцій системи (2.1) у вигляді рівнів змінних  $p_0'$ ,  $p_0''$  з'являються на виходах старшої ОКСП. Для декодування результату порівняння слів розробимо схему неповного дешифратора, виходи якого будуть одночасно і виходами багаторозрядної КСП слів. Використовуючи коди логічних умов із ТКСП (табл. 2.5), одержуємо досконалі форми функцій вихідного дешифратора:

$$f_{x=y} = \vee(0) = \&(1, 2) = \bar{p}_0' \cdot \bar{p}_0'',$$

$$f_{x<y} = \vee(1) = \&(0, 2) = \bar{p}_0' \cdot p_0'',$$

$$f_{x>y} = \vee(2) = \&(0, 1) = p_0' \cdot \bar{p}_0'',$$

що зведемо до МДНФ за допомогою діаграми Вейча двох змінних (рис. 2.9):

$$f_{x=y} = \overline{p_0} p_0'', \quad f_{x<y} = p_0'', \quad f_{x>y} = p_0'.$$

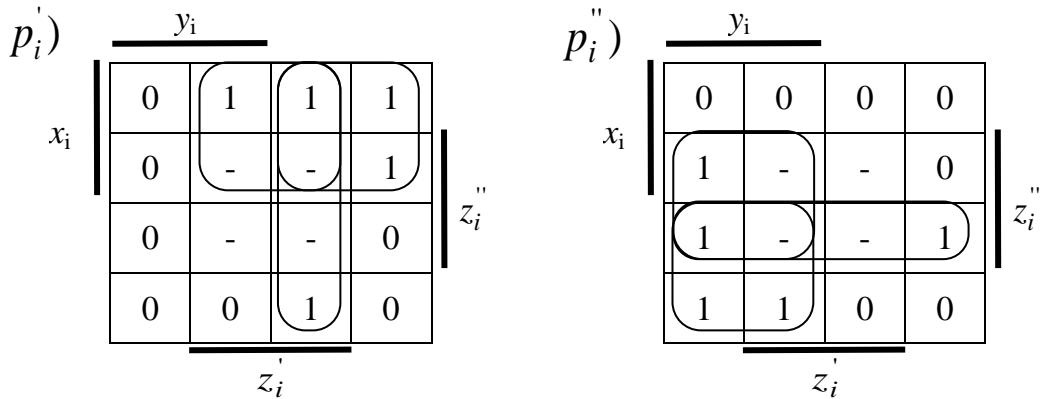


Рисунок 2.8 – Мінімізація функцій повної ОКСП

Таким чином, повна КСП трирозрядних слів з вихідним дешифратором описується такою системою перемикальних функцій, що перетворені в другу операторну форму (враховуючи, що  $p_{i+1} = z_i$ ):

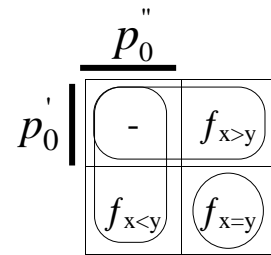


Рисунок 2.9 – Синтез вихідного

$$\left\{ \begin{array}{l} p_2' = x_2 \overline{y_2} = \overline{\overline{x_2 y_2}}, \\ p_2'' = \overline{x_2} y_2 = \overline{\overline{\overline{x_2} y_2}}, \\ p_1' = x_1 \overline{y_1} \vee (x_1 \vee \overline{y_1}) p_2' = \overline{\overline{\overline{x_1 \overline{y_1}} \cdot \overline{\overline{\overline{x_1} y_1}} \cdot p_2'}}, \\ p_1'' = \overline{x_1} y_1 \vee (\overline{x_1} \vee y_1) p_2'' = \overline{\overline{\overline{\overline{\overline{x_1} y_1}} \cdot \overline{\overline{\overline{\overline{x_1} y_1}} \cdot p_2''}}}, \\ p_0' = f_{x>y} = x_0 \overline{y_0} \vee (x_0 \vee \overline{y_0}) p_1' = \overline{\overline{\overline{\overline{x_0 \overline{y_0}} \cdot \overline{\overline{\overline{\overline{x_0} y_0}} \cdot p_1'}}}, \\ p_0'' = f_{x<y} = \overline{x_0} y_0 \vee (\overline{x_0} \vee y_0) p_1'' = \overline{\overline{\overline{\overline{\overline{\overline{\overline{x_0} y_0}} \cdot \overline{\overline{\overline{\overline{\overline{x_0} y_0}} \cdot p_1''}}}}}, \\ f_{x=y} = \overline{\overline{\overline{\overline{p_0' p_0''}}} = \overline{\overline{\overline{\overline{f_{x<y}} \cdot \overline{\overline{\overline{\overline{f_{x>y}}}}}}}}. \end{array} \right. \quad (2.13)$$

Помітимо, що логічні оператори  $\overline{\overline{x_i y_i}}, \overline{\overline{x_i \overline{y_i}}}$  ( $i = 0, 1$ ) у одержаній системі використовуються двічі, що відбито у структурі повної принципової КСП слів, що показана на рис. 2.10. ЛЕ, відмічені знаком  $\otimes$ ,

при реалізації та аналізі роботи схеми на лабораторному стенді мають бути обрані з індикаторами станів виходів (світлодіодами). Дослідження реалізованої схеми проводять за табл. 2.7, збудованої заздалегідь у відповідності з логікою роботи КСП.

Для того ж варіанта кодування вислови  $p_i'$  та  $p_i''$  повної ОКСП у 7-й ОФ, що реалізується на ЛЕ I-АБО-НІ, мають вигляд

$$p_i' = \overline{\bar{x}_i y_i} \vee (\bar{x}_i \vee y_i) \bar{z}_i' = \overline{\bar{x}_i y_i} \vee \overline{x_i \bar{y}_i} \cdot \bar{z}_i', \quad (2.14)$$

$$p_i'' = \overline{x_i \bar{y}_i} \vee (x_i \vee \bar{y}_i) \bar{z}_i'' = \overline{x_i \bar{y}_i} \vee \overline{\bar{x}_i y_i} \cdot \bar{z}_i''.$$

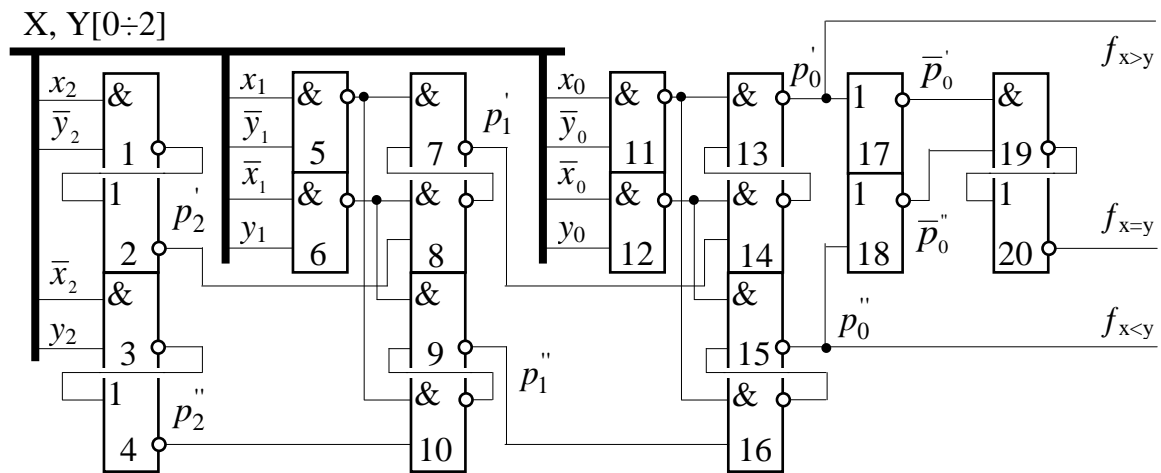


Рисунок 2.10 – Повна принципова КСП трирозрядних слів

Сигнали переносу в усіх розглянутих КСП виробляються та передаються послідовно від молодших розрядів до старших.

Швидкодіючі КСП будуються з паралельною та комбінованою організацією переносів, аналогічно схемам КСМ.

### Порівняння слів за допомогою суматора

Порівняємо вирази сигналів переносу  $p_i$  неповної ОКСП, синтез якої був проведений раніше, і однорозрядного суматора:

ОКСП 
$$p_i = \bar{x}_i y_i \vee (\bar{x}_i \vee y_i) \cdot z_i, \quad z_{n-1} = 0,$$

ОКСМ 
$$p_i = x_i y_i \vee (x_i \vee y_i) \cdot z_i.$$

Ці вирази відрізняються тільки кодами слів, що порівнюються (знаком інверсії над змінною  $x_i$ ). Видно, що для порівняння на менше слів  $X$  і  $Y$  на входи ОКСП необхідно подавати прямий код слова  $Y$  та інверсний код слова  $X$ , причому на вхід переносу молодшого розряду має подаватися



нульовий рівень, що у сукупності відповідає зворотному коду слова  $X$ . Тут під зворотним кодом будемо розуміти інверсію усіх розрядів числа без знаку. Отже, *ОКСП* виробляє сигнал переносу аналогічно суматору при арифметичному підсумовуванні прямого коду слова  $Y$  та зворотного коду слова  $X$ , що дозволяє використовувати схему суматора, тобто її частина, що виробляє переноси, для неповного порівняння на більше-менше. На етапі технічного проектування використання (навіть часткове) схеми *КСМ* замість *КСП* виправдано, оскільки суматори є стандартними елементами більшості серій інтегральних мікросхем.

Таблиця 2.7 – Дані для аналізу *КСП*

$X$	$Y$	$p_0' p_0''$	$f_{x<y}$	$f_{x=y}$	$f_{x>y}$
00	00	00	0	1	0
	01	01	1	0	0
	10	01	1	0	0
	11	01	1	0	0
01	00	10	0	0	1
	01	00	0	1	0
	10	01	1	0	0
	11	01	1	0	0
10	00	10	0	0	1
	01	10	0	0	1
	10	00	0	1	0
	11	01	1	0	0
11	00	10	0	0	1
	01	10	0	0	1
	10	10	0	0	1
	11	00	0	1	0

Аналіз роботи неповних *ОКСП*, що реалізують інші логічні умови типу нерівностей ( $X \leq Y$ ,  $X > Y$ ,  $X \geq Y$ ), показує, що всі вони описуються виразами двох типів (табл. 2.8), які відрізняються тільки кодами порівняльних слів. Реалізація жорсткої та нежорсткої нерівностей здійснюється *КСП* однакової структури та відрізняється лише значенням сигналу переносу  $z_{n-1}$  в молодший розряд багаторозрядної схеми. Таким чином, на схемах *КСМ* можна реалізувати будь-яке неповне порівняння слів, варіюючи коди цих слів та сигнали переносу у молодший розряд.

У табл. 2.9 наведені коди, в яких мають подаватися на *КСМ* слова, що порівнюються, для реалізації будь-яких логічних умов типу нерівностей з урахуванням їх кодування.

Таблиця 2.8 – Структури неповної ОКСП

Логічна умова	Логічна умова кодується одиницею ( $p_0 = 1$ )	Логічна умова кодується нулем ( $p_0 = 0$ )
$X < Y$	$p_i = \bar{x}_i y_i \vee (\bar{x}_i \vee y_i) z_i$	$p_i = x_i \bar{y}_i \vee (x_i \vee \bar{y}_i) z_i$
$X \leq Y$		
$X > Y$	$p_i = x_i \bar{y}_i \vee (x_i \vee \bar{y}_i) z_i$	$p_i = \bar{x}_i y_i \vee (\bar{x}_i \vee y_i) z_i$
$X \geq Y$		

Таблиця 2.9 – Порівняння слів на КСМ

Коди слів	Реалізуємі ЛУ	
	$p_0 = 1$	$p_0 = 0$
$X_{зв}, Y_{пр}$	$X < Y$	$X \geq Y$
$X_{дод}, Y_{пр}$	$X \leq Y$	$X > Y$
$X_{пр}, Y_{зв}$	$X > Y$	$X \leq Y$
$X_{пр}, Y_{дод}$	$X \geq Y$	$X < Y$

### Порівняння слів зі знаками

Розглянуті вище КСП виконують порівняння абсолютних величин слів. Порівняння слів зі знаками реалізуються складніше. Словесно функцію такої КСП можна описати наступним чином. Допустимо, треба порівняти слова зі знаками на менше. Слово  $X$  буде завжди менше слова  $Y$  в тому випадку, якщо перше з них від’ємне, а друге – позитивне. В разі збігу знаків задана логічна умова буде виконуватися, якщо при позитивних числах модуль першого менше модуля другого, а при від’ємних, навпаки, модуль першого більше модуля другого числа. Звідси випливає, що багаторозрядні КСП слів зі знаками на менше описуються такою перемикальною функцією  $f_{x < y}$ , у якій знаки від’ємних слів та задана логічна умова кодується одиницею:

$$f_{x < y} = x_0 \bar{y}_0 \vee \bar{x}_0 (x_0 \equiv y_0) \cdot f_{|x| < |y|} \vee x_0 (x_0 \equiv y_0) \cdot f_{|x| > |y|}. \quad (2.15)$$

Тут  $x_0, y_0$  – знакові розряди слів, що порівнюються;

$f_{|x| < |y|}, f_{|x| > |y|}$  – функції порівняння на менше і більше абсолютних величин слів  $X$  та  $Y$  відповідно;

$x_0 \equiv y_0 = x_0 y_0 \vee \bar{x}_0 \bar{y}_0$  – функція рівності знакових розрядів слів, що порівнюються.

Перший член виразу (2.15) дорівнює одиниці тільки при від’ємному  $X$ , коли  $x_0 = 1$ , і позитивному  $Y$ , коли  $y_0 = 0$ , а  $\bar{y}_0 = 1$ . Логічний добуток

$\bar{x}_0(x_0 \equiv y_0)$  відбиває факт позитивності та рівності знаків слів, що порівнюються.

Аналогічно можна одержати перемикальні функції порівняння слів зі знаками на більше та рівність:

$$\begin{aligned} f_{x>y} &= \bar{x}_0 y_0 \vee x_0 (x_0 \equiv y_0) \cdot f_{|x|<|y|} \vee \bar{x}_0 (x_0 \equiv y_0) \cdot f_{|x|>|y|} = \bar{f}_{x=y} \cdot \bar{f}_{x<y}, \quad (2.16) \\ f_{x=y} &= (x_0 \equiv y_0) \cdot f_{|x|=|y|} = \bar{f}_{x<y} \cdot \bar{f}_{x>y} = \overline{f_{x<y} \vee f_{x>y}}. \end{aligned}$$

Спростимо вираз (3.15), використовуючи закони алгебри логіки:

$$\begin{aligned} f_{x<y} &= x_0 \bar{y}_0 \vee \bar{x}_0 (x_0 \equiv y_0) f_{|x|<|y|} \vee x_0 (x_0 \equiv y_0) f_{|x|>|y|} = \\ &= x_0 \bar{y}_0 \vee \bar{x}_0 (x_0 y_0 \vee \bar{x}_0 \bar{y}_0) f_{|x|<|y|} \vee x_0 (x_0 y_0 \vee \bar{x}_0 \bar{y}_0) f_{|x|>|y|} = \\ &= x_0 \bar{y}_0 \vee \bar{x}_0 \bar{y}_0 \cdot f_{|x|<|y|} \vee x_0 y_0 \cdot f_{|x|>|y|} = \\ &= x_0 \bar{y}_0 \vee x_0 \bar{y}_0 \vee \bar{x}_0 \bar{y}_0 \cdot f_{|x|<|y|} \vee x_0 y_0 \cdot f_{|x|>|y|} = \\ &= (x_0 \bar{y}_0 \vee \bar{x}_0 \bar{y}_0 \cdot f_{|x|<|y|}) \vee (x_0 \bar{y}_0 \vee x_0 y_0 \cdot f_{|x|>|y|}) = \\ &= \bar{y}_0 (x_0 \vee \bar{x}_0 f_{|x|<|y|}) \vee x_0 (\bar{y}_0 \vee y_0 f_{|x|>|y|}) = \\ &= \bar{y}_0 (x_0 \vee f_{|x|<|y|}) \vee x_0 (\bar{y}_0 \vee f_{|x|>|y|}) = \\ &= x_0 \bar{y}_0 \vee \bar{y}_0 \cdot f_{|x|<|y|} \vee x_0 \cdot f_{|x|>|y|}. \end{aligned} \quad (2.17)$$

Із (2.17) випливає, що умова  $X < Y$  виконується, якщо перше слово від'ємне, а друге – додатне, чи, коли друге слово додатне, а по модулю воно більше першого, чи, коли перше слово від'ємне і за абсолютною величиною більше за друге.

Розглянемо структуру *КСП* слів зі знаками (рис. 2.11), яка реалізує вираз (2.17), що містить функції порівняння модулів слів на більше-менше. Порівняння абсолютних величин слів на менше провадиться схемою *КСМ*, на входи якого подаються цифрові розряди прямого коду  $Y$  і зворотного коду  $X$  (знак “ $\leftarrow$ ” на рис. 2.11). Сигнал переносу із старшого цифрового розряду *КСМ*  $p_1 = f_{|x|<|y|}$ , одиничне значення якого посвідчує про виконання умови  $|X| < |Y|$ , є виходом схеми порівняння модулів. Функцію  $f_{|x|>|y|}$  при готовій схемі для  $f_{|x|<|y|}$  технічно простіше реалізувати у вигляді

$$f_{|x|>|y|} = \bar{f}_{|x|=|y|} \cdot \bar{f}_{|x|<|y|} = \overline{f_{|x|=|y|} \vee f_{|x|<|y|}}.$$

Це пояснюється тим, що необхідну функцію рівності модулів можна частково реалізувати на тому ж *КСМ*, використовуючи результат додавання модулів. Справді, якщо слова дорівнюють за абсолютною величиною, то модуль суми зворотного коду  $X$  і прямого коду  $Y$  містить одиниці в усіх розрядах (наведіть приклад). Отже, операцію порівняння слів на рівність можна звести до більш простого порівняння на рівність результату на виході *КСМ* з константою  $K = (2^{n-1} - 1)_{(10)}$ , тобто  $f_{|x|=|y|} = f_{|s|=|k|} = s_1 \cdot s_2 \cdot \dots \cdot s_{n-1}$ , де

$s_i$  ( $i = 1, 2, \dots, n-1$ ) – цифрові розряди суми. Надана функція реалізується комбінаційною схемою порівняння з константою (СПК) – логічним елементом  $I$  із  $n-1$  входами.

Безпосередньо функцію  $f_{x < y}$  (формулу 2.17) реалізує схема порівняння слів зі знаками (СПЗ), для якої вхідними змінними є функції порівняння модулів та знаки слів  $X$  і  $Y$ .

Повна КСП слів із знаками повинна реалізувати і спрощені вирази (2.16). 3; 9-12]

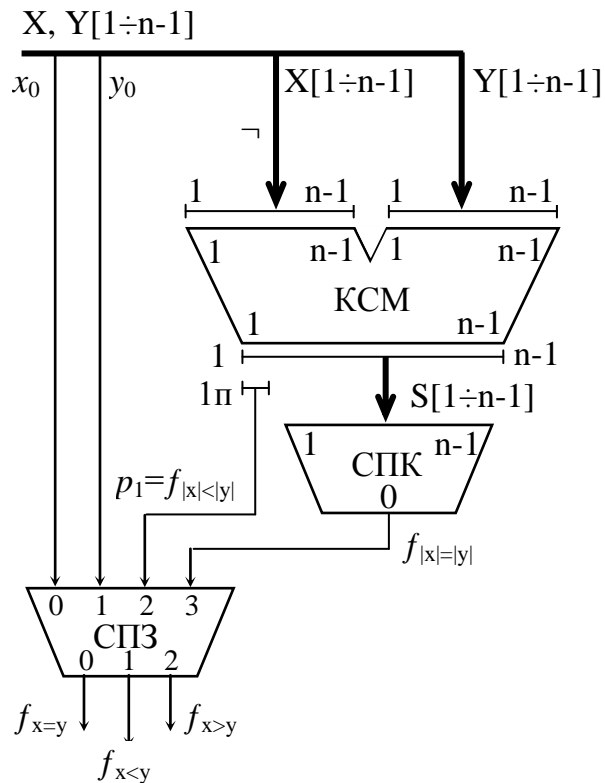


Рисунок 2.11 – КСП слів зі знаками

## 2.3 Рекомендації по виконанню курсового проекту

### 2.3.1 Основні вимоги й структура курсового проекту по дисципліні

Важливим етапом закріплення отриманих знань, показником уміння використати їх на практиці є робота над курсовим проектом.

Внаслідок виконання курсового проекту студенти повинні отримувати необхідні знання по використанню електронних цифрових пристроїв, уміти критично аналізувати і можливості окремих електронних блоків. Вони повинні вміти розробляти і вибирати раціональні схеми цифрових систем (структурну, функціональну і принципову) відповідно до початкових даних на проектування, проводити розрахунок основних блоків пристроїв, використовуючи сучасну елементну базу, оформляти пояснювальну записку і графічні матеріали відповідно до вимог діючих *ДЕСТ* і стандартів *ЕСКД*.

З метою придбання досвіду роботи із реальними електронними схемами в курсовому проекті передбачена *практична реалізація* розробленої цифрової схеми *асинхронного підсумовуючого лічильника* на лабораторному стенді й проведення її експериментального дослідження, усунення несправностей і налагодження.

Графічна частина, розташовувана в додатках, виконується відповідно до вимог *ЕСКД*. Формат аркушів – А4.

Надіслати виконаний та оформлений курсовий проект до університету на перевірку та рецензію до контрольної дати, яка також установлюється деканатом.

У п. 2.3.2 наведені 50 варіантів контрольних завдань. Вибір варіанту визначається таким чином: перша цифра 0, якщо рік закінчується цифрами 0, 5;

1, якщо рік закінчується цифрами 1, 6; 2, якщо рік закінчується цифрами 2, 7;

3, якщо рік закінчується цифрами 3, 8; 4, якщо рік закінчується цифрами 4, 9.

Друга цифра визначається останньою цифрою номера залікової книжки.

Розрахунково-пояснювальна записка повинна включати наступне.

Титульний аркуш

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Кафедра інформаційних технологій  
Заочний факультет

Курсовий проект  
з дисципліни Комп'ютерна схемотехніка та архітектура комп'ютерів  
На тему: «Синтез та дослідження схем лічильників з довільним модулем та  
порядком лічення»

Виконав студент гр \_\_\_\_\_  
\_\_\_\_\_ ПБ

Курсовий проект перевірен та  
допущен до захисту

«\_\_» \_\_\_\_\_ 201\_р.

Керівник \_\_\_\_\_

## Зміст

Вступ.....	.....
Список скорочень.....	.....
Завдання на проектування із вказівкою варіанта.....	.....
1 Основні теоретичні відомості про лічильникові схеми .....	.....
1.1 Визначення й класифікація схем лічильників .....	.....
1.2 Способи організації перенесень між розрядами лічильника. Синхронні й асинхронні лічильники.....	.....
1.3 Схеми асинхронних двійкових підсумовуючих і лічильників, що віднімають, на синхронних і асинхронних тригерах.....	.....
1.4 Двійково-десяткові коди (ДДК) і двійково-десяткові лічильники (ДДЛч) .....	.....
1.5 Організація перенесень між десятковими розрядами в ДДЛч.....	.....
2 Синтез підсумовуючого синхронного десяткового лічильника з довільним порядком лічення рахунку (що працює в коді .....) .....	.....
2.1 Побудова кодової таблиці переходів синхронного лічильника .....	.....
2.2 Побудова кодової таблиці функцій збудження тригерів заданого типу .....	.....
2.3 Одержання функцій збудження тригерів лічильника в досконалій формі.....	.....
2.4 Спільна мінімізація функцій збудження підсумовуючого лічильника.....	.....
2.5 Побудова схеми синхронного підсумовуючого лічильника.....	.....
2.6 Налагодження роботи лічильника на лабораторному стенді, усунення несправностей у роботі лічильника .....	.....
3 Синтез підсумовуючого асинхронного десяткового лічильника з довільним порядком лічення (що працює в коді .....) .....	.....
3.1 Суть метода проектування АЛч .....	.....
3.2 Побудова часової діаграми (ЧД) роботи лічильника.....	.....
3.3 Визначення по ЧД функцій синхронізації тригерів.....	.....
3.4 Спрощення функцій керування асинхронного лічильника по функціях збудження синхронного лічильника .....	.....
3.5 Побудова схеми асинхронного лічильника .....	.....
3.6 Налагодження роботи лічильника на лабораторному стенді, усунення несправностей у роботі лічильника .....	.....
4 Синтез реверсивного синхронного десяткового лічильника, що працює в коді .....	.....
4.1 Побудова кодової таблиці переходів реверсивного лічильника .....	.....

4.2 Побудова кодової таблиці функцій збудження тригерів для РСЛч .....	
4.3 Одержання функцій збудження тригерів лічильника в досконалій формі .....	
4.4 Спільна мінімізація функцій збудження реверсивного лічильника .....	
4.5 Побудова часової діаграми роботи РСЛч .....	
4.6 Побудова схеми реверсивного лічильника .....	
4.7 Налагодження роботи лічильника на лабораторному стенді, усунення несправностей у роботі лічильника .....	
Висновок (виводи).....	
Перелік посилань.....	
Додатки.....	
Додаток А – Схема принципова синхронного лічильника, що працює в коді .....	
Додаток Б – Схема принципова асинхронного лічильника, що працює в коді .....	
Додаток В – Схема принципова реверсивного лічильника, що працює в коді .....	



### 2.3.2 Перелік завдань на курсове проектування

В курсовому проекті необхідно виконати синтез схем лічильників з довільним модулем та порядком лічення.

1. Провести синтез підсумовуючого (віднімаючого) синхронного ЛЧ, що працює в заданому двійково-десятковому коді (табл. 3.1).

2. За результатами п.1 розробити схему простого СЛЧ на ЛЕ І-НЕ (І-АБО-НЕ).

Таблиця 3.1 – Варіанти завдань

Варіант	№ коду за табл. 3.2	Тип тригера	Варіант	№ коду за табл. 3.2	Тип тригера
1	1	T	26	10	D
2	1	D	27	11	JK
3	2	JK	28	11	T
4	2	T	29	11	D
5	2	D	30	12	JK
6	3	JK	31	12	T
7	3	T	32	12	D
8	3	D	33	13	JK
9	4	JK	34	13	T
10	4	D	35	13	D
11	5	JK	36	14	JK
12	5	T	37	14	T
13	5	D	38	14	D
14	6	JK	39	15	JK
15	6	D	40	15	T
16	7	JK	41	15	D
17	7	T	42	16	JK
18	7	D	43	16	T
19	8	T	44	17	JK
20	8	D	45	17	T
21	9	JK	46	17	D
22	9	T	47	18	JK
23	9	D	48	18	T
24	10	JK	49	18	D
25	10	T	00	19	JK

3 Побудувати часову діаграму (ЧД) роботи заданого двійково-десятькового ЛЧ без врахування часових затримок ЛЕ. ЧД будувати для лічильних (тактових) сигналів та прямих виходів тригерів.

4 Визначити по ЧД функцій синхронізації тригерів

5 Спростити функції керування асинхронного лічильника по функ

- ціях збудження синхронного лічильника .....
6. Побудувати схему асинхронного лічильника .....
7. Синтезувати реверсивний синхронний десятковий лічильник, що працює в коді
8. Побудувати часову діаграму роботи РСЛЧ .....
9. Побудувати схему реверсивного лічильника
- Графічна частина повинна містити:
- Принципову схему синхронного лічильника, що працює в коді ...
- Принципову схему асинхронного лічильника, що працює в коді
- Принципову схему реверсивного лічильника, що працює в коді

За курсовий проект студент може отримати 40 балів.

## Основні теоретичні положення

### Вступ

Лічильником (ЛЧ) називають цифровий автомат для зберігання довільного  $n$ -розрядного числа, що дозволяє збільшити (зменшити) це число на одиницю чи задану константу та часто має ланцюги установки нуля. Лічильники можуть виконувати також функції прийому і видачі чисел.

Максимальне число стійких внутрішніх станів лічильника називають його модулем  $N$ . Модуль – це максимальне число одиничних вхідних (лічильних сигналів), які може рахувати лічильник. Число тригерів (елементарних автоматів), необхідних для побудови лічильника, дорівнює числу його розрядів та визначається з формули  $n = \lceil \log_2 N \rceil$ . Дужки  $\lceil \cdot \rceil$  означають округлення в бік більшого цілого числа (округлення нагору). Вхідний сигнал обумовлює перехід лічильника з одного стійкого стану в інший. Номери станів відраховуються від деякого початкового (нульового) стану. Звичайно передбачається можливість переходу лічильника з довільного стану в початковий під дією спеціального керуючого сигналу установки до нуля ( $У0$ ). Крім того, лічильник може встановлюватися в початковий стан після завершення одного циклу роботи – підрахунку числа вхідних сигналів, яке дорівнює модулю лічильника.

За функціональними ознаками лічильники класифікуються наступним чином.

За модулем рахунку ЛЧ поділяють:

– на двійкові лічильники або лічильники за модулем  $N$ , де  $N = 2^n$  ( $n = 1, 2, 3, \dots$ ), тобто модуль двійкового лічильника дорівнює цілому ступеню числа 2;

– на недвійкові лічильники або лічильники за модулем  $M$  (лічильники з довільним модулем), де  $2^{n-1} < M < 2^n$ .

За напрямом рахування розрізняють:

– прості лічильники (тільки підсумовуючі або тільки віднімальні);

– реверсивні лічильники.

За видом порозрядного переносу (перенесення) розрізняють лічильники:

– з послідовним перенесенням (асинхронні лічильники);

– з паралельним або одночасним перенесенням (синхронні лічильники);

– з паралельно-послідовним (комбінованим) перенесенням.

За порядком зміни станів ЛЧ діляться на:

– лічильники з природничим порядком рахування;

– лічильники з довільним порядком рахування.

Прості лічильники можуть рахувати тільки в одному напрямку, тобто або тільки додавати вхідні сигнали до коду, що сформувався у лічильнику, або тільки віднімати вхідні сигнали з цього коду. Реверсивні лічильники залежно від керувальних сигналів можуть рахувати як в прямому, так і в зворотному напрямках.

Основні характеристики лічильників, які визначають їх швидкодію – розділювальна здатність та час встановлення (реєстрації) коду лічильника. Під розділювальною здатністю розуміють мінімально допустимий період  $T$  прямування вхідних сигналів, за яким лічильник працює без збоїв. Час встановлення коду являє собою інтервал часу між моментом надходження вхідного сигналу та моментом закінчення самого довгого перехідного процесу в схемі при переході до нового стійкого стану.

Важливим окремим випадком лічильників з довільним модулем є двійково-десяткові (чи просто десяткові) лічильники з модулем рахунку  $2^3 < M < 2^4$ . Для представлення однієї десяткової цифри двійково-десятковий лічильник повинен утримувати не менш як чотири тригери. Але оскільки чотирирозрядним лічильником можна представити 16 різноманітних станів, то синтез таких лічильників полягає у виключенні шести надмірних станів та забезпеченню обраного порядку перерахунку 10 станів, що залишились.

### **Способи організації перенесень. Асинхронні та синхронні лічильники.**

Проблема швидкодії лічильників вирішується використанням різноманітних методів перенесення міжрозрядної інформації. Простіше є послідовне перенесення, яке забезпечує найменшу швидкодію. Суть цього методу полягає в тому, що сигнали перенесення з кожного попереднього розряду послідовно розповсюджуються в наступні. Робота кожного наступного розряду можлива тільки після закінчення роботи попереднього. Лічильники з послідовним перенесенням називають *асинхронними*. Асинхронні лічильники можуть будуватися як на асинхронних (нетактованих), так і на синхронних (тактованих) тригерах з лічильним входом (Т-тригер). Схема двійкового підсумовуючого асинхронного лічильника (АЛЧ) на асинхронних Т-тригерах показана на рис. 3.1,а.

Лічильні сигнали  $k$  подаються на лічильний вхід першого тригера, на входи інших тригерів подаються сигнали з прямих виходів сусідніх молодших тригерів. Кожний Т-тригер є лічильником з модулем  $N = 2$ . Послідовним з'єднанням  $n$  таких тригерів за допомогою ланцюгів перенесення отримують модуль  $N = 2^n$ . Для побудови віднімаючого асинхронного лічильника організують ланцюги позики підімкненням входів тригерів до інверсних виходів сусідніх молодших тригерів (наведіть схему).

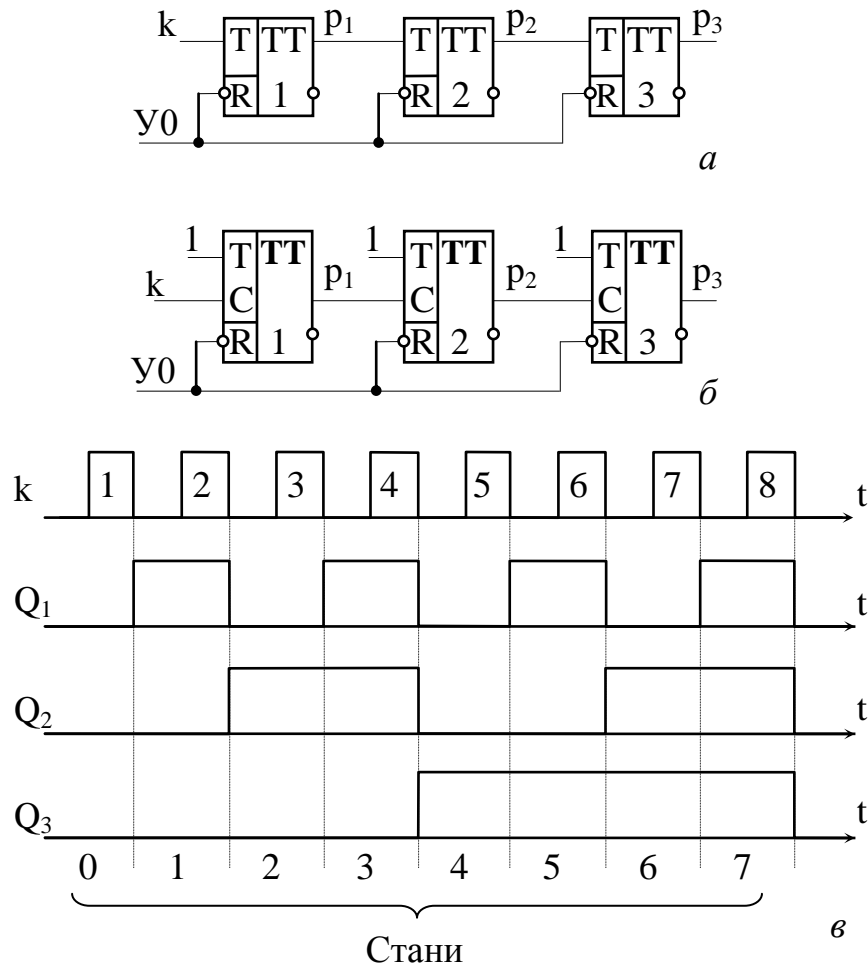


Рис. 3.1 – Двійкові АЛЧ на асинхронних (а) і синхронних (б) тригерах та їх часова діаграма роботи (в)

Схема асинхронного лічильника на синхронних Т-тригерах (рис. 3.1,б) отримується підімкненням входів синхронізації до прямих виходів сусідніх тригерів. На інформаційні входи Т в цьому випадку постійно подається рівень одиниці. Обидві розглянуті схеми мають ланцюг асинхронного скидання лічильника до нуля.

Часова діаграма роботи двійкових лічильників без врахування часових затримок зображена на рис. 3.1,в. Легко помітити, що лічильник за модулем  $N$  є дільником частоти вхідних лічильних сигналів в  $N$  разів, бо на виході  $i$ -го тригера з'являється лише  $2^i$ -й сигнал щодо числа сигналів  $k$ , які надійшли на вхід лічильника.

Максимальна швидкодія лічильника забезпечується паралельним перенесенням, яке реалізується за допомогою логічних елементів І, що вводяться в кожний розряд лічильника. Функції цих елементів полягають в тому, що за їх допомогою аналізується стан усіх молодших розрядів і залежно від комбінації їх станів виробляється сигнал перенесення.

Лічильники з паралельними перенесеннями називають *синхронними* (СЛЧ), бо всі тригери в них спрацьовують одночасно (синхронно) незалежно від їх розміщення в схемі та від початкового стану лічильника. На рис. 3.2,а зображена схема двійкового СЛЧ на синхронних Т-тригерах. Лічильний сигнал подається на входи синхронізації всіх тригерів лічильника разом. Вихідні сигнали перенесення  $p_i$  формуються в кожному розряді одночасно з допомогою схем І як незалежні функції стану всіх молодших розрядів лічильника:

$$p_i = \bigwedge_{j=1}^i Q_j = Q_1 \cdot Q_2 \cdot \dots \cdot Q_i, \text{ де } i = 1, 2, \dots, n. \quad (3.1)$$

Як буде показано далі, в лічильниках з довільним порядком лічення функції перенесення формуються із станів не тільки молодших, але і старших розрядів, в загальному випадку – із станів довільних розрядів.

Із виразу (3.1) та рис. 3.2,а видно, що число входів ЛЕ І зростає із збільшенням розрядності лічильника. Оскільки число входів в реальних ЛЕ скінчено та навантажувальна спроможність виходів тригерів обмежена, то розрядність лічильників з паралельним перенесенням звичайно невелика і на практиці не перевищує чотирьох.

Реальні JK-тригери на інтегральних елементах мають по три кон'юнктивно зв'язані входи  $J$  та  $K$  ( $J = J_1 \cdot J_2 \cdot J_3$  та  $K = K_1 \cdot K_2 \cdot K_3$ ), що дозволяє здійснити паралельне перенесення без додаткових ЛЕ І в групі з чотирьох тригерів. Тому паралельне перенесення в лічильниках інколи називають *груповим*. Схему чотирирозрядного СЛЧ на реальних JK-тригерах зображено на рис. 3.2,б.

Багаторозрядний лічильник підвищеної швидкодії розбивають на групи, кожна з яких містить не більше чотирьох тригерів. Групи поєднують між собою послідовно. При цьому останній тригер попередньої групи є джерелом синхронних сигналів для наступної. Такий метод організації перенесень називають паралельно-послідовним (або частково-груповим). З його допомогою забезпечується синхронний режим роботи всередині кожної групи і послідовна передача інформації від групи до групи.

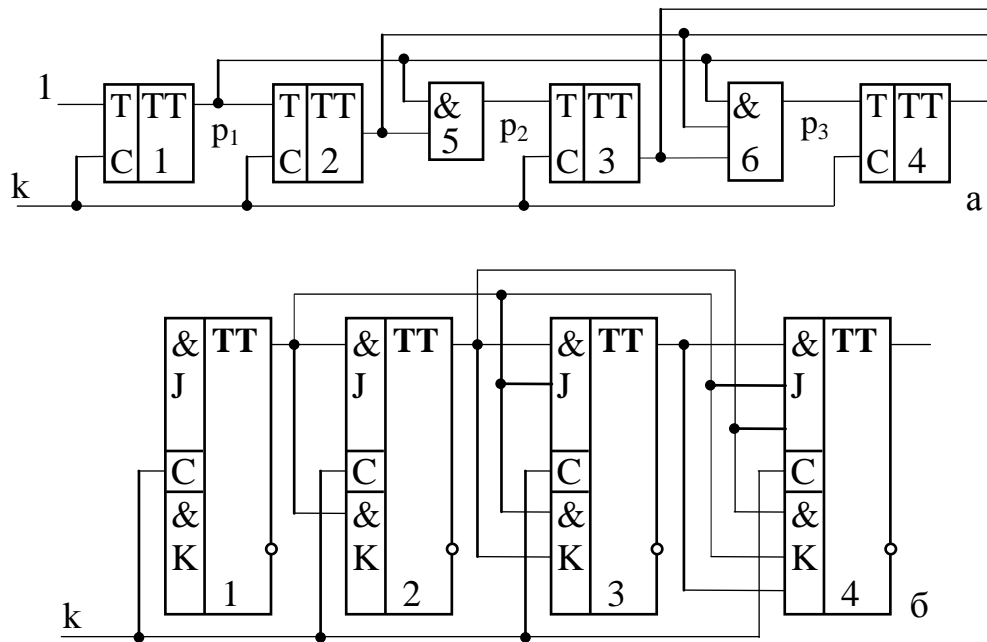


Рис. 3.2 – Двійковий СЛЧ на Т-тригерах (а)  
і реальних JK-тригерах (б)

Порівнюючи методи перенесень, визначимо, що перевагою асинхронних лічильників є простота їх структури. До недоліків АЛЧ відносяться:

- низька швидкодія через великий час установки коду;
- можливість появи проміжних нестійких станів при установці нового коду в лічильнику.

Справді, час установки  $t_{ycm}$  в асинхронних лічильниках зростає із збільшенням числа  $n$  тригерів:  $t_{ycm} = n \cdot t_m$ , де  $t_m$  – час затримки сигналу тригером. Крім того, при переході від одного коду до іншого асинхронний лічильник на короткий час може встановлюватися в проміжні нестійкі стани. Наприклад, перехід лічильника на рис. 3.1,а із стану  $3_{(10)}$  в стан  $4_{(10)}$  пов'язаний з послідовним проходженням крізь наступні стани:  $110_{(2)} \rightarrow 010_{(2)} \rightarrow 000_{(2)} \rightarrow 001_{(2)}$  (тут старший двійковий розряд згідно із схемою лічильника – правий). Якщо паралельний код з тригерів лічильника подається на дешифратор, то через зазначене неодночасне спрацювання тригерів на виході останнього можлива поява коротких невірних сигналів.

Цих недоліків позбавлені синхронні лічильники, час установлення яких не залежить від величини  $n$  та дорівнює  $t_{ycm} = t_i + t_m$ , де  $t_i$ ,  $t_m$  – відповідно є час затримки сигналу на ЛЕ I та тригері. Структурна організація синхронних лічильників є найскладнішою.

Лічильники з частково-груповим перенесенням за швидкодією та складністю займають проміжне місце між синхронними та асинхронними.

## Синтез простих синхронних лічильників з довільним порядком лічення

Лічильники з довільним порядком лічення відрізняються від лічильників з природним порядком лічення тим, що з приходом чергового вхідного сигналу  $k$  десятковий номер їх внутрішнього стану змінюється на значення, відмінне від одиниці. Причини, що спонукають використання таких лічильників, такі:

- можливість спрощення схеми дешифратора станів лічильника;
- можливість відрізнення всіх довільних станів лічильника взагалі без дешифратора (наприклад, в лічильниках з унітарним кодуванням – кільцевих регістрах, в яких циркулює всього одна одиниця);
- принципова можливість повного усунення критичних змагань в схемі лічильника при використанні сусіднього циклічного кодування станів.

В лічильниках з природним порядком лічення при переході від одного двійкового числа до сусіднього більшого чи меншого двійкового числа може виникати зміна цифр одночасно в декількох розрядах. Це інколи приводить до значних помилок при знятті закодованих кутових та лінійних переміщень. Ефективним засобом боротьби з помилками такого роду є використання спеціальних кодів, які називають відбитими (рефлексними). Відмінна особливість цих кодів полягає в тому, що сусідні кодові набори різняться цифрою тільки в одному розряді.

В лічильниках з сусіднім кодуванням будь-які два послідовні стани будуть відрізнятися тільки в одному розряді. Послідовні стани таких лічильників відтворюються на діаграмі Вейча переміщенням з будь-якої її клітинки в будь-яку сусідню (суміжну) з нею. Як приклад сусіднього кодування на діаграмі Вейча (рис. 3.3) показано зміну станів у чотирирозрядному лічильнику, працюючому в широко відомому в техніці коді Грея, що легко перетворюється в двійковий код. Для даного лічильника можна представити  $2^4$  варіантів сусіднього кодування.

Для зручності сприймання людиною-оператором цифрова вимірювальна інформація має бути представлена в десятковому коді. В цьому випадку для передачі, обробки та зберігання даних звичайно використовують двійково-десяткові коди, які легко перетворюються в десятковий код та просто реалізуються технічними засобами. Ці властивості обумовлюють широке застосування двійково-десяткових кодів в інформаційно-вимірювальних системах (ІВС) і

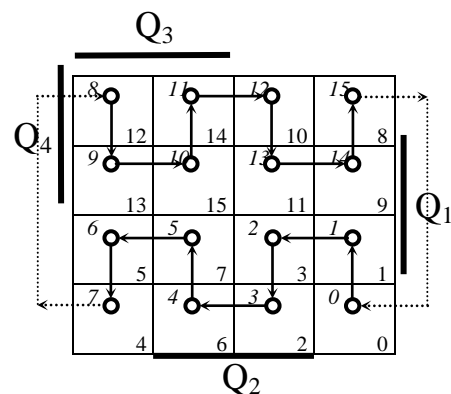


Рис. 3.3 – Приклад сусіднього кодування станів лічильника (код Грея)

цифрових пристроях вимірювання та обробки даних.

В двійково-десяткових кодах кожна десяткова цифра представляється групою цифр, що складається з чотирьох двійкових розрядів – двійковою тетрадою. Така група дозволяє сформувати 16 різноманітних наборів. В десятковій системі використовують тільки 10 цифр, тобто шість наборів надмірні. Оскільки надмірними можуть бути будь-які шість наборів, то це приводить до великого числа варіантів побудови двійково-десяткових кодів, частина з яких наведена у табл. 3.2.

Нехай кожна десяткова цифра  $N$  представляється у вигляді

$$N = \alpha_1 \cdot q_1 + \alpha_2 \cdot q_2 + \alpha_3 \cdot q_3 + \alpha_4 \cdot q_4,$$

де  $\alpha_i$  ( $i = 1, 2, 3, 4$ ) – двійкова цифра (0 або 1);  $q_i$  – вага  $i$ -го розряду.

Зрозуміло, що для кодування всіх десятичних цифр необхідно, щоб сума вагів була не менш як 9. Двійково-десяткові коди зображаються означенням ваги всіх чотирьох розрядів, наприклад, код 8421, код 7321, тощо (табл. 3.2).

Таблиця 3.2 – Деякі двійково-десяткові коди

Десяткове число	Двійковий код	Десяткові еквіваленти двійкових чисел у різних кодах																					
		несамодоповнювальні								самодоповнювальні								н/с					
		8	4	5	5	3	2	4	незважаєний код	5211				2421				4221				3 надміром	
		4	2	2	4	3	4	3		5	2	1	1	2	4	2	2	4	2	2	2	3	6
0	0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-		
1	0001	1	1	1	1	1	1	1	1	1	1	-	-	1	1	1	1	1	1	-	-		
2	0010	2	2	-	2	2	2	2	2	-	-	1	1	2	2	-	2	-	2	-	-		
3	0011	3	3	2	3	3	3	3	3	2	2	2	-	3	-	-	-	-	3	0	-		
4	0100	4	-	-	4	-	4	-	-	-	-	2	4	-	-	-	2	-	1	-	-		
5	0101	5	-	3	-	4	5	4	-	3	-	3	3	-	5	5	3	3	-	2	-		
6	0110	6	4	-	-	5	6	5	4	-	3	-	-	-	6	6	4	4	4	3	0		
7	0111	7	5	4	-	6	7	6	5	4	4	4	4	-	-	7	-	-	-	4	1		
8	1000	8	-	5	5	-	-	-	-	5	5	5	5	-	-	2	-	-	-	5	2		
9	1001	9	-	6	6	-	-	-	-	-	6	-	-	-	3	3	5	5	5	6	3		
10	1010	-	6	-	7	-	-	-	-	6	-	6	6	-	4	4	6	6	-	7	4		
11	1011	-	7	7	8	-	-	7	-	-	-	7	5	-	-	-	7	-	8	5	-		
12	1100	-	-	-	9	-	-	-	8	7	7	7	-	6	-	-	-	-	6	9	6		
13	1101	-	-	8	-	7	-	8	9	-	-	8	8	7	7	-	7	-	7	-	7		
14	1110	-	8	-	-	8	8	9	6	8	8	-	-	8	8	8	8	8	8	-	8		
15	1111	-	9	9	-	9	9	-	7	9	9	9	9	9	9	9	9	9	9	-	9		
Варіанти		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		



Десяткове число  $138_{(10)}$  в кодї 8421 зображується у вигляді 0001 0011 1000<sub>(2-10)</sub>, а в кодї 4321 – 0001 0100 1101<sub>(2-10)</sub>.

Особливу групу складають самодоповнювальні двійково-десяткові коди. Характерна особливість цих кодів – сума двійкового коду будь-якої десяткової цифри та її інверсного двійкового коду (що отримується заміною нулів на одиниці та навпаки) має дорівнювати двійковому коду цифри 9. Такі коди дозволяють легко виявити перенесення в старшу тетраду і отримати зворотний чи доповняльний коди при десятковому додаванні.

Всі перелічені двійково-десяткові коди називають зваженими. Кожному розряду в таких кодах поставлена у відповідність певна вага. Використання зважених двійково-десяткових кодів полегшує переведення чисел з одної системи числення в іншу. Однак розрізняють двійково-десяткові коди, що називають незваженими, в яких вага розрядів не визначена, наприклад, код “з надміром 3”.

Синтез будь-яких синхронних двійкових та недвійкових лічильників з природним та довільним порядком лічення проводиться однаково за допомогою загального методу синтезу цифрових автоматів.

Розглянемо особливості проектування синхронних лічильників з довільним модулем та порядком лічення на прикладі синтезу двійково-десяткових лічильників для одного десяткового розряду.

Початковими даними для синтезу є кодована таблиця переходів (КТП) проектованого лічильника і умовна таблиця переходів (УТП) обраного типу елементарного автомату (тригера). Суть синтезу полягає в визначенні функцій збудження кожного окремого тригера та побудові за одержаними функціями схеми синхронного лічильника.

Для прикладу спроектуємо підсумовувальний двійково-десятковий лічильник, що працює в незваженому кодї (див. табл. 3.2.). Як елементарний автомат оберемо універсальний JK-тригер, УТП якого наведена в табл. 3.3.

Складемо кодовану таблицю переходів КТП (табл. 3.4) лічильника в обраному кодї, в якій наведені всі можливі переходи лічильника з одного стану в інший.

Лічильний сигнал  $k$  в синхронних лічильниках подається на входи синхронізації всіх тригерів одночасно. Оскільки прості лічильники (підсумовуючі або віднімаючі) виконують тільки одну мікрооперацію, на яку орієнтована їх структура, то вони не містять керуючих шин ( $l = \log_2 l = 0$ ). Скидання лічильника до нуля будемо здійснювати за допомогою асинхронних установочних входів  $\bar{R}_a$  тригерів.

Таблиця 3.3 – УТП JK-тригера



Таблиця 3.4 – КТП СЛЧ

Десятковий еквівалент		0	1	2	3	4	5	6	7	8	9
x \ A	Q <sub>4</sub>	0	0	0	0	0	0	1	1	1	1
	Q <sub>3</sub>	0	0	0	0	1	1	1	1	1	1
	Q <sub>2</sub>	0	0	1	1	1	1	1	1	0	0
	Q <sub>1</sub>	0	1	0	1	0	1	0	1	0	1
1	Q <sub>4</sub>	0	0	0	0	0	1	1	1	1	0
	Q <sub>3</sub>	0	0	0	1	1	1	1	1	1	0
	Q <sub>2</sub>	0	1	1	1	1	1	0	0	0	0
	Q <sub>1</sub>	1	0	1	0	1	0	1	0	1	0

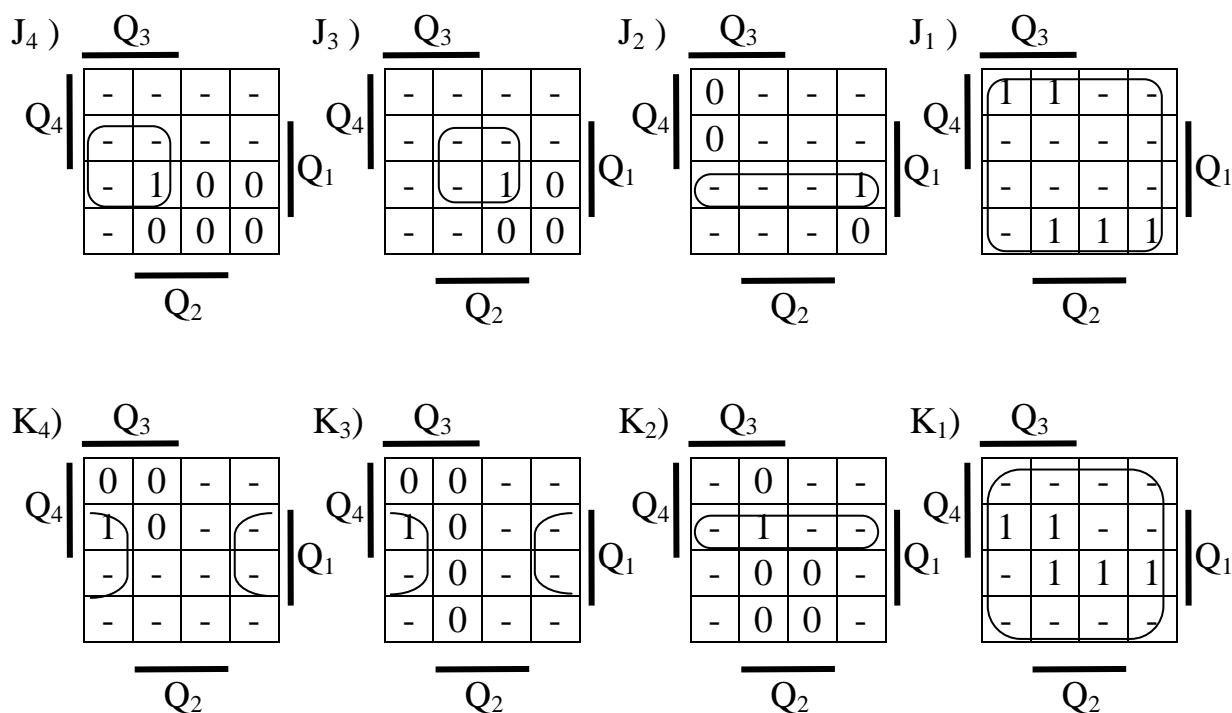
За КТП СЛЧ (табл. 3.4) та УТП (табл. 3.3) будемо кодовану таблицю функцій збудження КТФЗ (табл. 3.5), з якої вибираємо функції збудження тригерів в досконаліх формах:

$$\begin{aligned}
 J_4 &= \vee(7) = \&(0, 1, 2, 3, 6), & K_4 &= \vee(13) = \&(14, 15, 12); \\
 J_3 &= \vee(3) = \&(0, 1, 2), & K_3 &= \vee(13) = \&(6, 7, 14, 15, 12); \\
 J_2 &= \vee(1) = \&(0, 12, 13), & K_4 &= \vee(15) = \&(2, 3, 6, 7, 14); \\
 J_1 &= \vee(0, 2, 6, 14, 12) = \&(-), & K_1 &= \vee(1, 3, 7, 15, 13) = \&(-).
 \end{aligned}$$

Таблиця 3.5 – КТФЗ СЛЧ на JK-тригерах

x \ A	Q <sub>4</sub>	0	0	0	0	0	0	1	1	1	1
	Q <sub>3</sub>	0	0	0	0	1	1	1	1	1	1
	Q <sub>2</sub>	0	0	1	1	1	1	1	1	0	0
	Q <sub>1</sub>	0	1	0	1	0	1	0	1	0	1
1	J <sub>4</sub> K <sub>4</sub>	0-	0-	0-	0-	0-	1-	-0	-0	-0	-1
	J <sub>3</sub> K <sub>3</sub>	0-	0-	0-	1-	-0	-0	-0	-0	-0	-1
	J <sub>2</sub> K <sub>2</sub>	0-	1-	-0	-0	-0	-0	-0	-1	0-	0-
	J <sub>1</sub> K <sub>1</sub>	1-	-1	1-	-1	1-	-1	1-	-1	1-	-1

Проводимо сумісну мінімізацію функцій збудження за допомогою діаграм Вейча:



МДНФ функцій збудження тригерів проектованого синхронного лічильника мають такий вигляд:

$$\begin{aligned}
 J_4 &= Q_3 Q_1, & J_3 &= Q_2 Q_1, & J_2 &= \bar{Q}_4 Q_1, & J_1 &= 1, \\
 K_4 &= \bar{Q}_2 Q_1; & K_3 &= \bar{Q}_2 Q_1; & K_2 &= Q_4 Q_1; & K_1 &= 1.
 \end{aligned}$$

Схему спроектованого синхронного лічильника на реальних JK-тригерах та часову діаграму його роботи показано на рис. 3.4. Синтез лічильника на основі будь-якого іншого типу синхронного тригера (D, DV, T, RS) проводиться аналогічно, різниця полягає лише у використанні відповідної умовної таблиці переходів. Оскільки як реальні тригери інших типів не мають кон'юнктивно зв'язаних інформаційних входів, то реалізація функцій збудження таких тригерів проводиться за допомогою додаткових логічних елементів І-НЕ.

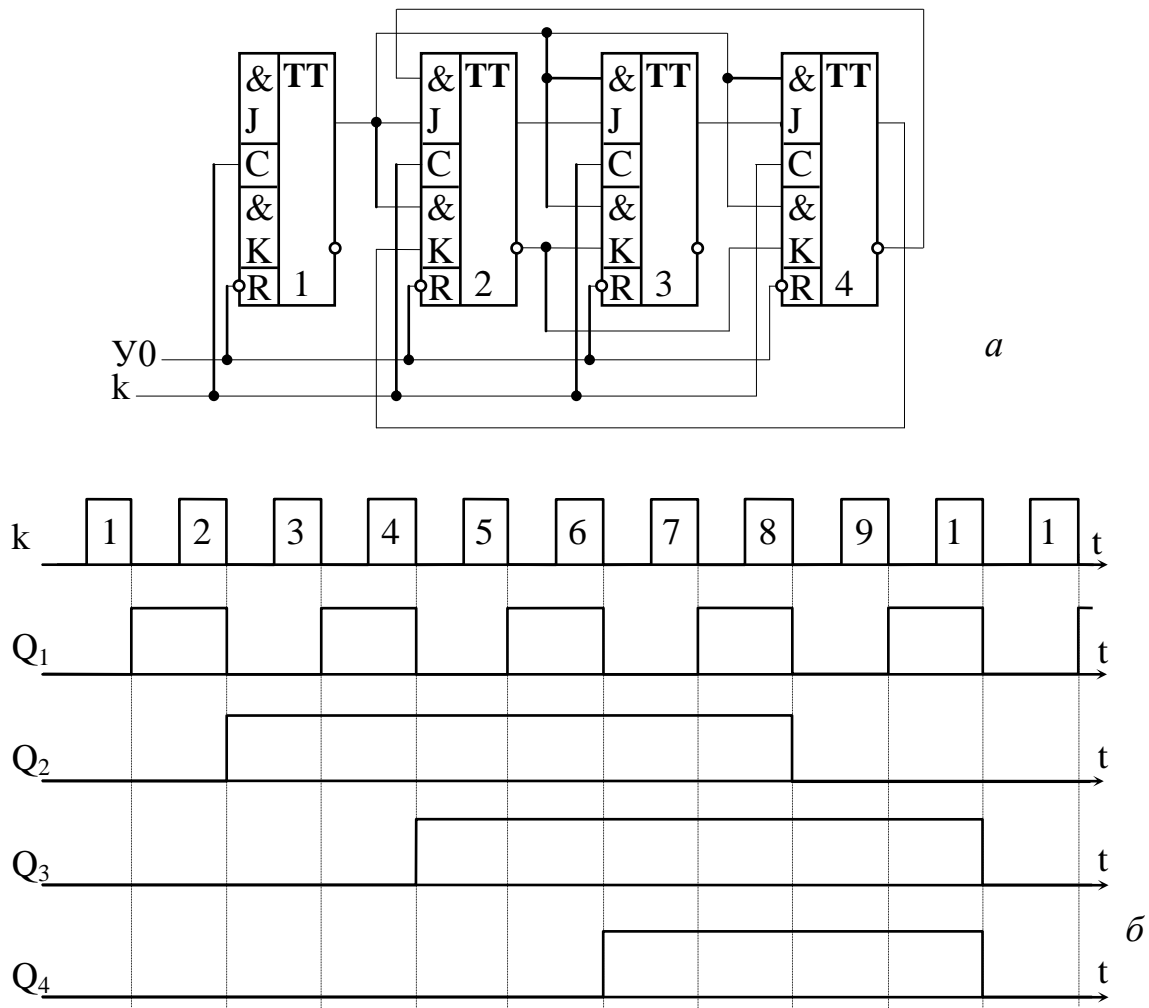


Рис. 3.4 – Схема СЛЧ, працюючого в незваженому коді (а) і часова діаграма його роботи (б)

Перенесення між десятковими розрядами двійково-десятькового лічильника утворюється при появі коду, відповідного цифрі "9". Якщо використовувати надмірні набори, то можна мінімізувати вираз для сигналу перенесення. Для цього необхідно занести в діаграму Вейча надмірні набори та набір, відповідний цифрі "9" в даному двійково-десятьковому коді, інші поля діаграми заповнюють нулями. Для розглянутого вище прикладу перенесення у старший розряд мінімізується діаграмою Вейча та дорівнює  $p_1 = Q_3 \bar{Q}_2 Q_1$  або  $p_1 = Q_4 \bar{Q}_2 Q_1$ . Перенесення  $p_1$  є синхросигналом для тригерів старшої групи.

## Синтез асинхронних та реверсивних синхронних лічильників

На основі синхронних тригерів можна будувати не тільки довільні синхронні, але і асинхронні лічильники. Асинхронні лічильники відрізняються при цьому від синхронних тим, що на синхровходи  $S$  деяких тригерів надходять не лічильні сигнали, а сигнали з виходів логічних елементів, зв'язаних з виходами сусідніх тригерів, або безпосередньо з виходів сусідніх тригерів.

Мета синтезу асинхронних лічильників – виявити можливість керування (синхронізації) деякими тригерами від сусідніх замість використання для цього лічильних сигналів. А оскільки число керуючих сигналів за одиницю часу з виходів сусідніх тригерів (через ділення частоти кожним тригером в 2 рази) менше за число лічильних сигналів, то асинхронна організація керування тригерами приводить до спрощення структури АЛЧ порівняно з СЛЧ. Найбільш просто реалізується керування даним синхронним тригером від сусіднього за так званим *двійковим переходом*, коли сигнал із виходу сусіднього тригера подається безпосередньо на вхід синхронізації даного тригера, що перетворився в асинхронний  $T$ -тригер.

Основою метода синтезу АЛЧ є заздалегідь отримані функції збудження відповідних СЛЧ. При цьому повинні враховуватися функціональні особливості обраного типу синхронного тригера. Внаслідок синтезу асинхронного лічильника знаходяться його функції керування, тобто функції збудження та функції синхронізації тригерів. Оскільки при синтезі враховуються функціональні особливості та структура елементарного автомату, то алгоритми синтезу асинхронних лічильників для різноманітних типів синхронних тригерів будуть різними. Розглянемо синтез асинхронних лічильників на прикладі проектування десяткових лічильників.

### Синтез асинхронних лічильників на синхронних JK-тригерах.

Заздалегідь необхідно отримати функції збудження  $J_i$  та  $K_i$  відповідного СЛЧ, що становлять кон'юнкції (або диз'юнкції кон'юнкцій), вихідних змінних тригерів  $Q_j$  ( $j = 1, 2, 3, 4$ ). З аналізу цих функцій видно, що одна і та сама змінна  $Q_j$  може входити в обидві функції збудження  $J_i$  та  $K_i$ . Згідно з структурою JK-тригера це означає, що відповідний  $i$ -й тригер АЛЧ може синхронізуватися змінною  $Q_j$ , тобто ця змінна є його функцією синхронізації  $C_i = \tilde{Q}_j$ . Символ " $\sim$ " зазначає, що змінна може бути з інверсією або без неї. При цьому частини кон'юнкцій, що залишились (диз'юнкції кон'юнкцій) є функціями збудження  $J_i$  та  $K_i$  АЛЧ.

Таким чином алгоритм синтезу АЛЧ на JK-тригерах такий:

– заздалегідь отримують розглянутим раніше методом функції збудження відповідного СЛЧ;

– загальна частина функцій збудження СЛЧ стає функцією  $C_i$  синтезуючого АЛЧ, а частини, що залишились – функціями збудження АЛЧ;

– якщо загальної частини в функціях збудження  $i$ -го тригера СЛЧ немає, то функції збудження відповідного тригера АЛЧ залишаються без зміни, а функція синхронізації  $C_i = k$ , де  $k$  – вхідний лічильний сигнал;

– за часовою діаграмою (ЧД) лічильника уточнюється можливість подальшого спрощення функцій керування АЛЧ.

На ЧД легко виявляються функції синхронізації  $C_i$  АЛЧ. Це здійснюється таким чином. Якщо протягом всього циклу рахування  $i$ -й тригер встановлюється на одиницю та скидається до нуля тільки в моменти часу, що відповідають спадам (заднім фронтам, скиданню до нуля)  $j$ -го тригера, то  $C_i = Q_j$ . В окремих випадках режими роботи деяких тригерів АЛЧ, синтезованого за наведеним алгоритмом, можуть бути спрощені за допомогою ЧД. Якщо за ЧД видно, що  $i$ -й тригер може керуватися безпосередньо від попереднього (за двійковим переходом), то одержані за алгоритмом функції керування  $i$ -го тригера замінюються більш простими:  $C_i = Q_{i-1}$ ,  $J_i = K_i = 1$  (режим асинхронного Т-тригера).

Розглянемо синтез десяткового алч, працюючого в коді 8421.

Функції збудження відповідного СЛЧ такі:

$$J_4 = Q_3 Q_2 Q_1, \quad J_3 = Q_2 Q_1, \quad J_2 = \bar{Q}_4 Q_1, \quad J_1 = 1,$$

$$K_4 = Q_1; \quad K_3 = Q_2 Q_1; \quad K_2 = Q_1; \quad K_1 = 1.$$

Виділяючи загальні частини виразів, одержимо функції синхронізації АЛЧ:

$$C_4 = Q_1, \quad C_3 = Q_2 Q_1 = Q_2 \text{ (двійковий перехід)}, \quad C_2 = Q_1, \quad C_1 = k.$$

Остаточно функції керування АЛЧ будуть:

$$\begin{aligned} C_4 = Q_1, J_4 = Q_3 Q_2, K_4 = 1; & \quad C_2 = Q_1, J_2 = \bar{Q}_4, K_2 = 1; \\ C_3 = Q_2, J_3 = K_3 = 1; & \quad C_1 = k, J_1 = K_1 = 1. \end{aligned}$$

На ЧД (рис. 3.5,а) стрілками показано, від яких ближчих змін можуть синхронізуватися відповідні тригери. Видно, що функції синхронізації, одержані алгоритмічно і за допомогою ЧД, повністю збігаються. Схема синтезованого АЛЧ зображена на рис. 3.5,б.

## Синтез АЛЧ на синхронних Т-тригерах

Функції збудження відповідного СЛЧ являють собою також або одну кон'юнкцію, або диз'юнкції деяких кон'юнкцій. Процедура синтезу АЛЧ така:

- одержання функцій збудження відповідного СЛЧ;
- визначення по ЧД функцій синхронізації  $S_i$  АЛЧ;
- визначення функцій збудження АЛЧ як залишкових частин однієї чи кількох кон'юнкцій функцій збудження СЛЧ.

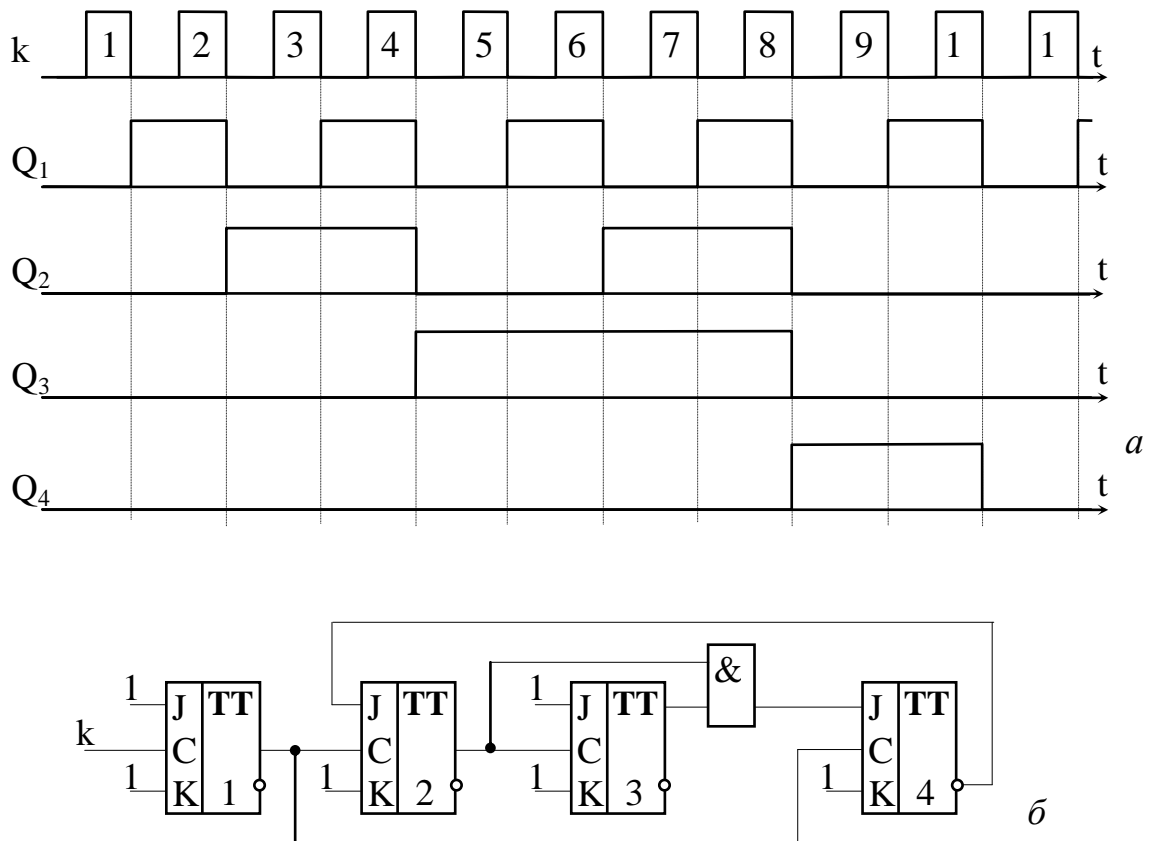


Рис. 3.5 – Часова діаграма (а) і схема АЛЧ (б)  
на JK-тригерах, працюючого в коді 8421

Розглянемо метод синтезу АЛЧ на Т-тригерах, працюючого в несамодоповнювальному коді 2421 (№ 6 в табл. 3.2). Функції збудження відповідного СЛЧ, заздалегідь одержані відомим методом, будуть

$$T_4 = Q_3 Q_2 Q_1; T_3 = Q_4 Q_1 \vee \bar{Q}_3 Q_2 Q_1; T_2 = Q_4 Q_1 \vee \bar{Q}_3 Q_1 \vee \bar{Q}_2 Q_1; T_1 = 1.$$

За ЧД (рис. 3.6,а) визначаємо функції синхронізації АЛЧ:  
 $S_4 = Q_1$ ,  $S_3 = Q_2$  (двійковий перехід),  $S_2 = Q_1$ ,  $S_1 = k$ .

очевидно, що зміна стану третього тригера на протилежний завжди збігається із спадом (заднім фронтом) сигналу  $q_2$ , що означає двійковий перехід третього тригера, тобто  $c_3 = q_2$ ,  $t_3 = 1$ . другий тригер, наприклад, не може керуватися за двійковим переходом від першого, бо за спадом змінної  $q_1$ , що збігається за часом зі спадом восьмого лічильного сигналу  $k$ , другий тригер повинен залишатися в стані одиниці. відмітимо, що чд будуються тільки за заданим кодом незалежно від типу тригера та лічильника (алч або слч).

Отже, функції керування даного АЛЧ:

$$\begin{array}{ll} C_4 = Q_1, T_4 = Q_3 Q_2; & C_2 = Q_1, T_2 = Q_4 \vee \bar{Q}_3 \vee \bar{Q}_2; \\ C_3 = Q_2, T_3 = 1; & C_1 = k, T_1 = 1. \end{array}$$

Схема АЛЧ показана на рис. 3.6,б.

### Синтез АЛЧ на D-тригерах.

Функціонування D-тригера відрізняється від роботи розглянутих вище типів тригерів. Щоб залишитись у колишньому положенні JK- та T-тригери вимагають подачі на їх інформаційні входи сигналів неактивного рівня. D-тригер, на відміну від них, для перебування в стані одиниці вимагає потактного підтвердження цього стану подачею на його інформаційний вхід діючого значення вхідного сигналу. Це приводить до того, що функції збудження D-тригера СЛЧ являють собою диз'юнкції досить великого числа кон'юнкцій. Оскільки структура АЛЧ простіша, то багато кон'юнкцій в функціях збудження СЛЧ виявляються зайвими (надмірними) для роботи АЛЧ.

Таким чином, синтез АЛЧ на D-тригерах повинен починатися з виявлення цих надмірних кон'юнкцій та формування функцій збудження, що містять тільки істотні (необхідні) кон'юнкції. Після цього синтез АЛЧ на D-тригерах аналогічний синтезу АЛЧ на синхронних T-тригерах.



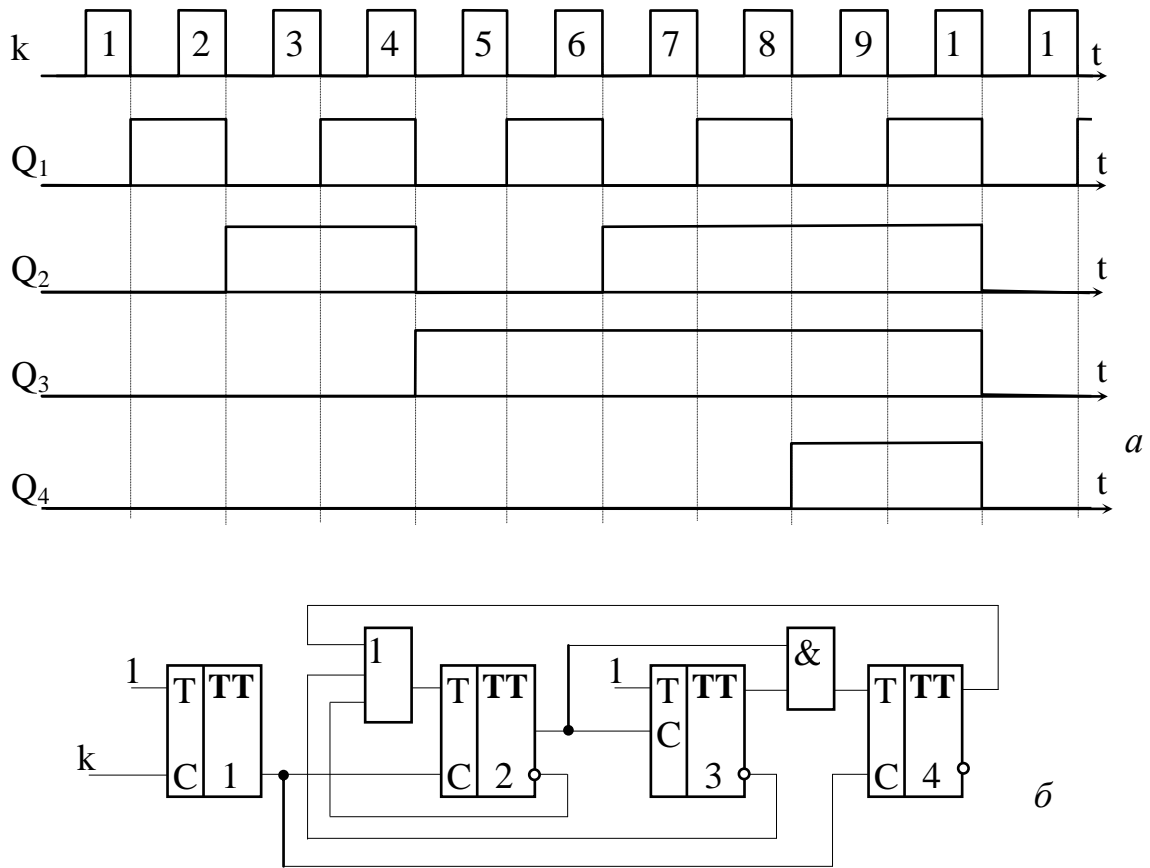


Рис. 3.6 – Часова діаграма (а) та схема АЛЧ (б) на синхронних Т-тригерах, працюючого в несамодоповнювальному коді 2421

Виявлення надмірних кон'юнкцій виконується за допомогою ЧД. Одиначні рівні вихідних сигналів  $Q_i$  протягом одного циклу роботи СЛЧ відзначаються номерами кон'юнкцій, які дозволяють установку до 1 або підтверджують одиничний стан тригерів під дією лічильних сигналів  $k$ . Потім з ЧД визначають функції синхронізації АЛЧ. Надмірні кон'юнкції знаходять за допомогою функцій синхронізації АЛЧ та проставлених номерів кон'юнкцій СЛЧ. Припустимо, з ЧД відомо, що  $i$ -й тригер може синхронізуватися ближчим вихідним сигналом молодшого ( $i-2$ )-го тригера. Тоді з функцій збудження  $i$ -го тригера СЛЧ виключаються надмірні кон'юнкції, положення номерів яких не збігаються на ЧД зі спадами вихідних сигналів ( $i-2$ )-го тригера. Кон'юнкції, положення номерів яких збігаються зі спадами вихідних сигналів ( $i-2$ )-го тригера, є істотними. Їх залишають для формування функцій збудження АЛЧ.

Розглянемо синтез АЛЧ на D-тригерах, працюючого в самодоповнювальному коді 4221 (№ 16, табл. 3.2). Функції збудження відповідного СЛЧ, одержані відомим способом, такі:

$$\begin{aligned}
D_4 &= \underbrace{Q_4 \bar{Q}_2}_1 \vee \underbrace{Q_4 \bar{Q}_1}_2 \vee \underbrace{Q_3 \bar{Q}_1}_3, \\
D_3 &= \underbrace{Q_4 \bar{Q}_1}_1 \vee \underbrace{Q_3 \bar{Q}_2}_2 \vee \underbrace{\bar{Q}_3 Q_2}_3, \\
D_2 &= \underbrace{Q_4 Q_3 \bar{Q}_1}_1 \vee \underbrace{\bar{Q}_2 Q_1}_2, \\
D_1 &= \bar{Q}_1.
\end{aligned}$$

Пронумеруємо кон'юнкції функцій  $D_4, D_3, D_2$ . З функції  $D_1$  видно, що перший тригер повинен працювати в режимі асинхронного Т-тригера, тобто перекидатися до протилежного стану кожним лічильним сигналом  $k$ . Побудуємо ЧД та відмітимо на ній вихідні сигнали тригерів номерами кон'юнкцій, збуджуючих D-тригери СЛЧ у визначені дискретні моменти часу під дією лічильних сигналів  $k$  (рис. 3.7,а).

Наприклад, третій тригер збереже в четвертому такті одиничний стан через значення одиниці другої кон'юнкції  $Q_3 \bar{Q}_2$  в попередньому третьому такті, інші дві кон'юнкції функції збудження  $D_3$  в цей час дорівнюють нулю. Тому на ЧД відзначаємо одиничний рівень вихідного сигналу  $Q_3$  в кінці третього такту цифрою 2.

Визначимо з ЧД функції синхронізації АЛЧ (стрілками відмічені ближчі сигнали, від яких можуть синхронізуватися відповідні тригери):  $C_4 = Q_3, C_3 = Q_2, C_2 = k, C_1 = k$ .

Викреслимо номери кон'юнкцій, положення яких не збігається зі спадами виявлених синхронізуючих сигналів відповідних тригерів. Тоді функції керування АЛЧ, складені із істотних кон'юнкцій, будуть

$$D_4 = Q_3 \bar{Q}_1, D_3 = \bar{Q}_3 Q_2, D_2 = Q_4 Q_3 \bar{Q}_1 \vee \bar{Q}_2 Q_1, D_1 = \bar{Q}_1.$$

Позбуваючись в цих виразах від змінних, що входять до відповідних функцій синхронізації, отримаємо кінцевий вигляд функцій управління АЛЧ:

$$\begin{aligned}
C_4 = Q_3, D_4 = \bar{Q}_1; & & C_3 = Q_2, D_3 = \bar{Q}_3; \\
C_2 = k, D_2 = Q_4 Q_3 \bar{Q}_1 \vee \bar{Q}_2 Q_1; & & C_1 = k, D_1 = \bar{Q}_1.
\end{aligned}$$

Із ЧД видно, що перший, третій та четвертий тригери мають працювати в режимі асинхронного Т-тригера з керуванням від попередніх тригерів. Звичайно такий режим D-тригера організується подачею на D-вхід сигналу з його інверсного виходу (наприклад,  $D_4 = \bar{Q}_4$ ).

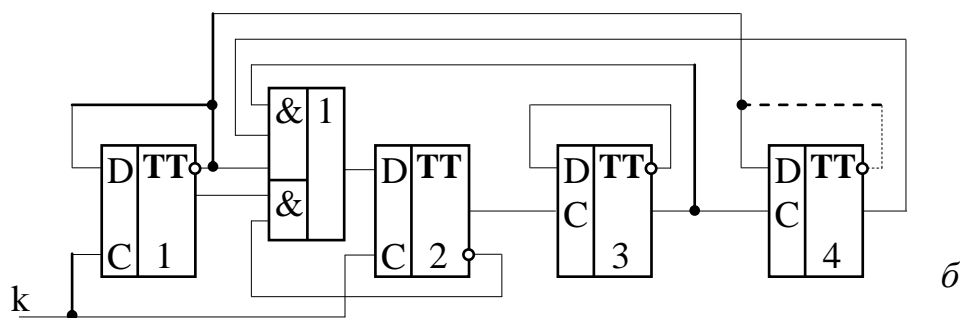
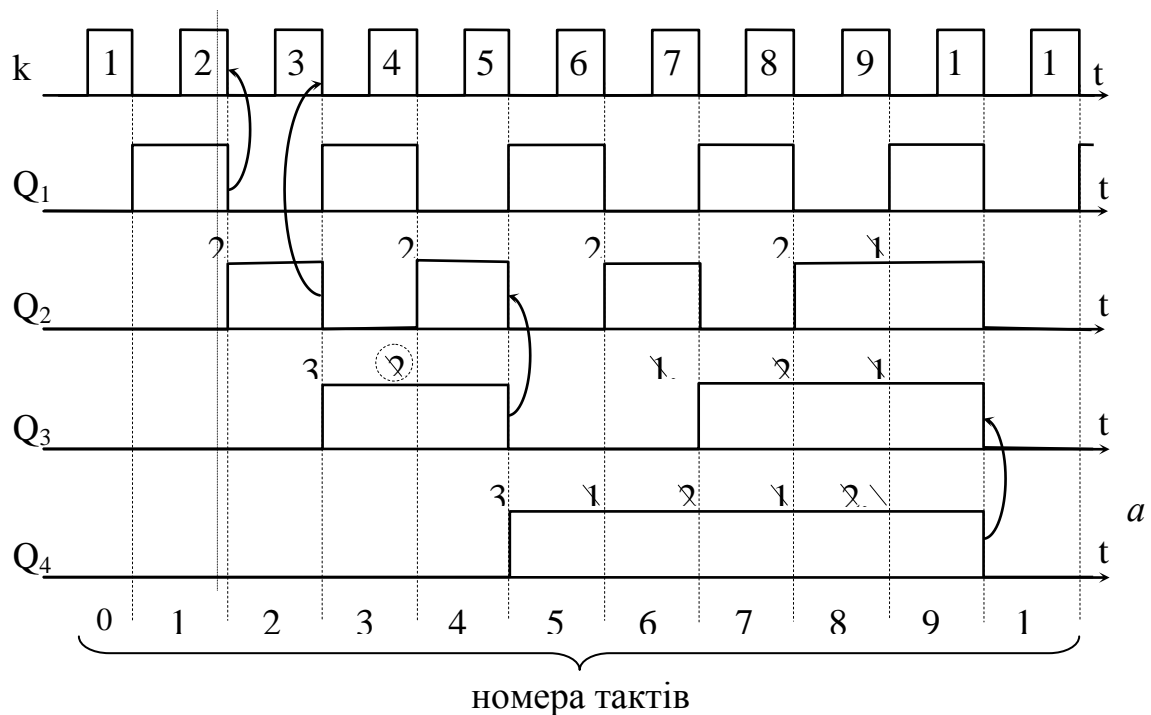


Рис. 3.7 – Часова діаграма (а) та схема АЛЧ (б) на D-тригерах, працюючого в самоповнювальному коді 4221 (№ 16, табл. 3.2)

В даному ж прикладі для четвертого тригера отримана рівноцінна функція збудження  $D_4 = \overline{Q_1}$ . Схему АЛЧ показано на рис. 3.7,б.

### Синтез реверсивних синхронних лічильників з довільними модулем та порядком лічення

Принципово нічим не відрізняється від синтезу простих синхронних лічильників. Різниця полягає тільки в кодованій таблиці переходів, яка у реверсивних лічильників містить два рядки переходів: одна – для мікрооперації додавання, друга – віднімання. Приклад КТП реверсивного лічильника для двійково-десятькового коду 5421 (див. № 4, табл. 3.2, ) наведено в табл. 3.6. Схема реверсивного лічильника містить дві лінії

управління –  $x$  та  $\bar{x}$ , які настроюють його на виконання відповідної мікрооперації.

Таблиця 3.6 – КТП РСЛЧ (код 5421)

Десятковий еквівалент		0	1	2	3	4	5	6	7	8	9
x	Q <sub>4</sub>	0	0	0	0	0	1	1	1	1	1
	Q <sub>3</sub>	0	0	0	0	1	0	0	0	0	1
	Q <sub>2</sub>	0	0	1	1	0	0	0	1	1	0
	Q <sub>1</sub>	0	1	0	1	0	0	1	0	1	0
0 (ддв)	Q <sub>4</sub>	0	0	0	0	1	1	1	1	1	0
	Q <sub>3</sub>	0	0	0	1	0	0	0	0	1	0
	Q <sub>2</sub>	0	1	1	0	0	0	1	1	0	0
	Q <sub>1</sub>	1	0	1	0	0	1	0	1	0	0
1 (вдм)	Q <sub>4</sub>	1	0	0	0	0	0	1	1	1	1
	Q <sub>3</sub>	1	0	0	0	0	1	0	0	0	0
	Q <sub>2</sub>	0	0	0	1	1	0	0	0	1	1
	Q <sub>1</sub>	0	0	1	0	1	0	0	1	0	1

Кодована таблиця функцій збудження реверсивного СЛЧ на Т-тригерах, побудована згідно з табл. 3.6 та за властивістю заданого елементарного автомата (або його УТП), наведена в табл. 3.7.

Неповністю визначені функції збудження тригерів, отримані з табл. 3.7 в досконалих формах, мають вигляд:

$$T_4 = \vee (4, 12, 16, 24) = \& (0, 1, 2, 3, 8, 9, 10, 11, 17, 18, 19, 20, 25, 26, 27, 28);$$

$$T_3 = \vee (3, 4, 11, 12, 16, 20, 24, 28) = \& (0, 1, 2, 8, 9, 10, 17, 18, 19, 25, 26, 27);$$

$$T_2 = \vee (1, 3, 9, 11, 18, 20, 26, 28) = \& (0, 2, 4, 8, 10, 12, 16, 17, 19, 24, 25, 27);$$

$$T_1 = \vee (0, 1, 2, 3, 8, 9, 10, 11, 17, 18, 19, 20, 25, 26, 27, 28) = \& (4, 12, 16, 24).$$

Таблиця 3.7 – КТФЗ РСЛЧ на Т-тригерах

x \ A	Q <sub>4</sub>	0	0	0	0	0	1	1	1	1	1
	Q <sub>3</sub>	0	0	0	0	1	0	0	0	0	1
	Q <sub>2</sub>	0	0	1	1	0	0	0	1	1	0
	Q <sub>1</sub>	0	1	0	1	0	0	1	0	1	0
0 (ддв)	T <sub>4</sub>	0	0	0	0	1	0	0	0	0	1
	T <sub>3</sub>	0	0	0	1	1	0	0	0	1	1
	T <sub>2</sub>	0	1	0	1	0	0	1	0	1	0
	T <sub>1</sub>	1	1	1	1	0	1	1	1	1	0
I (вдм)	T <sub>4</sub>	1	0	0	0	0	1	0	0	0	0
	T <sub>3</sub>	1	0	0	0	1	1	0	0	0	1
	T <sub>2</sub>	0	0	1	0	1	0	0	1	0	1
	T <sub>1</sub>	0	1	1	1	1	0	1	1	1	1

Сумісна мінімізація одержаних функцій виконується за допомогою діаграм Вейча п'ятьох змінних.

### III Організація контролю знань та вмінь студентів

#### 3.1 Система контролю знань та вмінь студентів

Поточний контроль здійснюється за наступними формами:

- перевірка контрольної роботи
- курсового проекту;
- перевірка знань студента під час лабораторних робіт.

Підсумковий контроль проводиться на основі накопиченої (інтегральної) суми балів, яку отримав студент за підсумками поточного контролю та підсумкового семестрового контролю (залік).

Накопичувальна підсумкова оцінка (ПО) засвоєння студентом навчальної дисципліни складається з:

- контрольної роботи та курсового проекту (ОМ – оцінка міжсесійна);
- захисту лабораторних робіт (ОЗЕ – оцінка сесійна);
- оцінювання заходу підсумкового контролю, (ОПК – залік).

Студент вважається допущеним до підсумкового семестрового контролю, якщо він виконав всі види робіт поточного контролю, передбачені робочою навчальною програмою дисципліни і набрав за накопичувальною системою суму балів не менше 50% від максимально можливої за дисципліну.

## 3.2 Форми контролю знань та вмінь студентів

### 3.2.1 Поточний контроль

Поточний контроль складається з:

– контрольної роботи за яку студент може отримати 35 балів. Контрольна робота вважається зарахованою, якщо студент отримав за неї не менше ніж 21 бал, тобто 60% від максимально можливої оцінки. Студенти, які виконали контрольну роботу та отримали за результатами перевірки не менше ніж 60% мають допуск до іспиту з дисципліни;

– захисту курсового проекту, за який він може отримати 40 балів (24 бала (60%) за оформлення курсової роботи згідно ДСТУ та відповідність змісту роботи її темі і 16 балів(40%) за захист курсового проекту). Курсовий проект вважається зарахованим, якщо студент отримав не менше ніж 24 бала (60%);

– лабораторні роботи, за які він може отримати 25 балів (лабораторні роботи вважаються зарахованими, якщо студент отримав за них 13 балів (50%)).

### 3.2.2 Підсумковий контроль

Підсумковий семестровий контроль (ОПК) здійснюється під час залікової контрольної роботи

Залікова контрольна робота складається з 20 тестових питань.

Накопичена підсумкова оцінка (ПО) розраховується таким чином:

$$ПО = 0,75 \times [0,5 \times (ОЗЕ + ОМ)] + 0,25 \times ОЗКР$$

ОЗЕ – кількісна оцінка (у відсотках від максимально можливої) за виконання лабораторних робіт;

ОМ – кількісна оцінка (у відсотках від максимально можливої) за контрольну роботу та курсовий проект.

ОЗКР – оцінка залікової контрольної роботи.

## 3.3 Перелік базових знань та вмінь

Узагальнюючи інформацію, що викладена у підпунктах 2.1.2-2.1.4, можна навести *повний перелік базових знань та вмінь* з дисципліни «Комп'ютерна схемотехніка та архітектура комп'ютерів»:

1) Перша тема:

- поняття про системи числення (СЧ);
- позиційні і непозиційні СЧ;
- загальні методи переведення чисел з однієї системи числення в іншу ( переведення чисел із довільної СЧ в довільну з

виконанням  $AO$  в новій  $h$  СЧ. Переведення чисел з виконанням операцій в старій СЧ);

- окремі методи переведення чисел з двійкової СЧ в вісімкову (шістьнадцяткову) та навпаки;
- переведення чисел з врахуванням точності;

## 2) Друга тема:

- кодування цифрової інформації;
- визначення прямого, зворотного і допоміжного кодів додатних та від'ємних чисел;
- алгебраїчні дії в зворотному і допоміжному кодах.
  - правила алгебраїчного складення;
- переповнення розрядної сітки суматора і методи його виявлення;

## 3) Третя тема:

- визначення, характеристика та способи представлення КС;
- перемикальні функції (ПФ) та їх змінні;
- способи завдання ПФ;
- суперпозиція перемикальних функцій;
- основні закони алгебри логіки;
- теорема Поста про функціональну повноту;
- диз'юнктивні й кон'юнктивні нормальні форми представлення перемикальних функцій;
- конституенти одиниці й нуля;
- досконалі диз'юнктивні й кон'юнктивні нормальні форми перемикальних функцій;
- одержання досконалих форм із таблиць істинності;
- перетворення довільної ДНФ (КНФ) у СДНФ (СКНФ);
- мінімізація перемикаючих функцій;
- діаграми Вейча;

## 4) Четверта тема:

- однорозрядний комбінаційний суматор (ОКС);
- спосіб організації утворення сум багаторозрядних слів;
- організація перенесення в паралельних суматорах;
- суматори з послідовним, паралельним і комбінованим перенесеннями;
- комбінаційна схема знаходження переповнення розрядної сітки суматора;
- комбінаційні схеми порівняння;
- порівняння слів з константами на рівність і більше-менше;
- порівняння слів на рівність;
- порівняння слів на більше-менше;

- повна схема порівняння слів;
- порівняння слів на суматорах;
- синтез КС порівняння слів;
- дешифратори;
- правила синтезу цифрових схем порівняння;
- порівняння слів з константами;
- порівняння слів.

5) П'ята тема:

- цифрові автомати, способи їх завдання ;
- кінцеві автомати Мілі і Мура;
- абстрактний синтез ЦА.;
- елементарні автомати;
- алгоритм структурного синтезу автоматів;
- тригерні схеми;
- синтез і дослідження тригерних схем різного типу;
- перетворення (метаморфози) тригерних схем.

6) Шоста тема:

- визначення та класифікація, регістрів;
- паралельний прийом інформації;
- прийом і видача парафазного коду.
- принцип організації зсуву інформації ;
- синтез і дослідження регістрів зберігання і зсуву на тригерах різних типів.

7) Сьома тема:

- визначення, модуль, функціональна класифікація лічильників;
- способи організації переносів ;
- асинхронні і синхронні лічильники та їх характеристика;
- синтез синхронних (двійково-десяткових) лічильників з довільним модулем і порядком лічення на тригерах різних типів;
- синтез асинхронних лічильників на синхронних JK-, T-, D-тригерах;
- синтез реверсивних синхронних лічильників.

8) Восьма тема:

- цифрові комп'ютери визначення;
- запам'ятовуючі пристрої ;
- Flash-пам'ять;
- розпаралелювання обчислювальних процесів;
- універсальні мікропроцесори;
- RISC- та CISC-процесори



### 3.4 Список скорочень

АЛЧ – асинхронний лічильник,  
ГІ – генератор імпульсів,  
ГПІ – генератор поодиноких імпульсів,  
ДВ – діаграма Вейча,  
ДДК – двійково-десятковий код,  
ДДНФ, СДНФ – досконала диз'юнктивна нормальна форма,  
ДКНФ, СКНФ – досконала кон'юнктивна нормальна форма,  
ДКС – додаткова комбінаційна схема,  
ЕА – елементарний автомат,  
ЕОМ – електронна обчислювальна машина,  
ІВС – інформаційно-вимірювальна система,  
КС – комбінаційна схема,  
КСМ – комбінаційний суматор,  
КСП – комбінаційна схема порівняння,  
КТП – кодована таблиця переходів,  
ЛЕ – логічний елемент,  
ЛКМ – локальна комп'ютерна мережа,  
ЛО – логічний оператор,  
ЛУ – логічна умова,  
ЛЧ – лічильник,  
МДНФ – мінімальна диз'юнктивна нормальна форма,  
МО – мікрооперація,  
МП – мікропроцесор,  
ОА – операційний автомат,  
ОКСМ – однорозрядний комбінаційний суматор,  
ОКСП – однорозрядна комбінаційна схема порівняння,  
ОФ – операторна форма,  
ПК – персональний комп'ютер,  
ППРС – переповнення розрядної сітки,  
ПСВ – повнота системи виходів,  
ПСП – повнота системи переходів,  
РГ (Rg) – регістр,  
РСЛЧ – реверсивний синхронний лічильник,  
СЛЧ – синхронний лічильник,  
СМ – суматор,  
СП – сигнал перенесення (переносу),  
СПЗ – схема порівняння слів зі знаками,  
СПК – схема порівняння з константою,  
СПП – схема організації паралельних перенесень,  
СТП – синхронна таблиця переходів,  
СЧ – система числення,  
СЧП – сума часткових добутоків,

ТІ – таблиця істинності,  
 ТКВхС – таблиця кодів вхідних сигналів,  
 УГП, УГО – умовне графічне позначення,  
 УЛС – універсальний лабораторний стенд,  
 ЦОМ – цифрова обчислювальна машина,  
 ЦП – цифровий пристрій,  
 ЧД – часова діаграма,  
 ШД – шина даних

Таблиця 1 Терміни перевірки контрольної роботи в міжсесійний період

Змістовний модуль	Блок	Срок контролю
1. Способи представлення інформації і схемотехніка цифрових елементів.	1.Форми зображення інформації. Логічні основи побудови елементів. 2.Схемотехніка комбінаційних вузлів, цифрових елементів, цифрових вузлів.	1-10 жовтня  1-10 листопада
2. Схемотехніка цифрових та аналогових вузлів.	3. Схемотехніка аналогових вузлів, обслуговуючих елементів, . Інтегральні системи елементів. 4. Схемотехніка комбінаційних вузлів.	1-10 січня  1-10 березня
3. Архітектура комп'ютера.	5. Цифрові комп'ютери. Запам'ятовуючі пристрої. Flash-пам'ять. Процесори. Паралельні обчислювальні системи. 6.Супер-комп'ютери. Універсальні мікропроцесори. Схеми підтримки МП на системних платах. Структури мікропроцесорних систем. RISC- та CISC-процесори.	1 -10 квітня

