

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

**УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І
ТЕХНОЛОГІЙ**
ННІ «Дніпровський інститут інфраструктури і транспорту»



Кафедра «Автоматика та телекомунікації»

В авторській редакції

ЦИФРОВІ СИСТЕМИ АВТОМАТИКИ ТА РОБОТОТЕХНІКИ

Навчально–методичні рекомендації до проведення практичних занять

Електронне видання

ДНІПРО
2026

УДК 004.94:656.25(076.5)

Ц 75

Автор:

Володимир Профатилов

Електронне видання

Схвалено Групою забезпечення якості освітньої програми
G7.2.02 «Автоматика та автоматизація на транспорті»
Протокол № 4 від 2.04.2026

Схвалено Групою забезпечення якості освітньої програми
J7.2.02 «Системи керування рухом поїздів»
Протокол № 4 від 3.04.2026

Ц 75 Цифрові системи автоматики та робототехніки: навчально–методичні рекомендації до проведення практичних занять / В. І. Профатилов; Укр. держ. ун–т науки і технологій. – Електрон. вид. – Дніпро: УДУНТ, 2026. – 51 с.

Навчально–методичні рекомендації призначені для виконання практичних занять студентами, які здобувають освітній ступінь «магістр» по спеціальностям G7 «Автоматизація, комп’ютерно–інтегровані технології та робототехніка» (ОПП Автоматика та автоматизація на транспорті) та J7 «Залізничний транспорт» (ОПП Системи керування рухом поїздів) денної форми навчання.

Навчально–методичні рекомендації містять вісім практичних занять, у яких розглянуто моделювання роботи логічних схем за допомогою САПР PROTEUS, схемотехніка пристроїв вводу аналогових сигналів та формування аналогових сигналів довільної форми за допомогою цифрових пристроїв, розробка, проектування та моделювання цифрових пристроїв на мові VHDL за допомогою САПР QUARTUS Prime. Навчально–методичні рекомендації містять основні теоретичні відомості, порядок виконання практичних занять та питання для самоконтролю.

Іл. 19. Табл. 7. Бібліогр. 9 назв.

© Профатилов В. І., укладання, 2026

©Укр. держ. ун–т науки і технологій, 2026

ЗМІСТ

Вступ	4
Практичне заняття 1 Моделювання роботи логічних схем за допомогою САПР Proteus	6
Практичне заняття 2 Ввід аналогового сигналу в комп'ютер для цифрової обробки сигналів....	10
Практичне заняття 3 Програмно–керовані джерела живлення та генератори сигналів довільної форми	18
Практичне заняття 4 Розробка комбінаційних логічних схем та тригерів на мові VHDL	25
Практичне заняття 5 Система автоматизованого проектування QUARTUS Prime.....	29
Практичне заняття 6 Розробка чотирьохрозрядного суматора на мові VHDL та моделювання його роботи в системі QUARTUS Prime.....	36
Практичне заняття 7 Розробка дешифратора на мові VHDL та моделювання його роботи в САПР QUARTUS Prime	40
Практичне заняття 8 Розробка структурного архітектурного тіла на мові VHDL	43
Список літератури та інформаційні ресурси	48
Додаток А. Правила оформлення практичних занять	49

ВСТУП

Метою вивчення дисципліни «Цифрові системи автоматики та робототехніки» є засвоєння принципів проектування та експлуатації цифрових систем автоматики та робототехніки на сучасній електронній елементній базі, засвоєння навиків роботи з апаратними та програмними засобами для проектування, діагностування та налаштування цифрових систем автоматики та робототехніки, а також знання напрямів та перспектив розвитку сучасних систем залізничної автоматики та робототехніки.

Студенти, що навчаються за освітньою програмою «Автоматика та автоматизація на транспорті», в результаті вивчення дисципліни отримують компетентності та матимуть здатність застосовувати спеціалізовані концептуальні знання, що включають сучасні наукові здобутки, а також критичне осмислення сучасних проблем у сфері автоматизації та комп'ютерно-інтегрованих технологій для розв'язування складних задач професійної діяльності, розробляти комп'ютерно-інтегровані системи управління складними технологічними та організаційно-технічними об'єктами, застосовуючи системний підхід із врахуванням нетехнічних складових оцінки об'єктів автоматизації, вміння проектувати та налагоджувати спеціальні вимірjuвальні та керуючі системи з урахуванням властивостей виробничо-технологічних комплексів, а також організувати проведення монтажних і налагоджуваних робіт систем автоматизації та телекомунікаційних систем.

Студенти, що навчаються за освітньою програмою «Системи керування рухом поїздів», в результаті вивчення дисципліни отримують компетентності та будуть знати і розуміти сучасні методи наукових досліджень, організації та планування експерименту, комп'ютеризованих методів дослідження та опрацювання результатів, вирішувати задачі зі створення, експлуатації, утримання, ремонту та утилізації об'єктів залізничного транспорту, у тому числі на межі із суміжними галузями, інженерними науками, фізикою, екологією та економікою, розробляти та пропонувати нові технічні рішення та застосовувати нові технології, вміння застосовувати у професійній діяльності універсальні та спеціалізовані системи управління життєвим циклом (PLM), автоматизованого проектування (CAD), виробництва (CAM) та інженерних досліджень (CAE), розраховувати характеристики об'єктів залізничного транспорту відповідно до спеціалізації, а також розробляти та оптимізувати параметри технологічних процесів, в тому числі з застосуванням автоматизованого комп'ютерного проектування виробництва вузлів, агрегатів та систем об'єктів залізничного транспорту.

Практичні заняття надають студентам можливість отримати практичні навички проектування сучасних цифрових системи автоматики на базі ПЛІС, діагностування і тестування цифрових системи автоматики та робототехніки за допомогою сучасних апаратних та програмних засобів. Практичні заняття передбачають роботу в спеціалізованій лабораторії та можливість користуватися сучасними спеціалізованими програмними й апаратними засобами експлуатації цифрових системи автоматики та робототехніки.

Вимоги до оформлення звіту з практичних занять:

- звіт можна оформляти у рукописному вигляді в зошиті або друкованому вигляді за допомогою комп'ютерної техніки на листах формату А4;
- якщо звіт виконується у друкованому вигляді, то необхідно дотримуватися вимог оформлення, що наведені у додатку А;
- титульний лист звіту повинен містити інформацію, яка однозначно ідентифікує його виконавця: назву дисципліни та вид занять, прізвище та ім'я студента, номер академічної групи, номер шифру або варіанту (якщо це є у вимогах практичних занять);
- звіт повинен включати номер практичних занять, тему та мету практичних занять;
- структура звіту повинна відповідати вимогам розділу практичних занять «Зміст звіту».

Вимоги з охорони праці та правила поведінки під час проведення практичних занять:

Перед початком проведення першого практичного заняття студент повинен пройти інструктаж з правил техніки безпеки та правил поведінки у приміщенні де проводяться заняття, а також розписатися в журналі проведення інструктажів. Студент який не пройшов такий інструктаж не допускається до виконання практичних занять.

Студент зобов'язаний виконувати вимоги правил техніки безпеки та правила поведінки у приміщенні де проводяться заняття. Якщо студент порушує вимоги правил техніки безпеки або правила роботи на комп'ютері, то викладач має право відсторонити студента від виконання практичних занять.

Якщо практичні заняття виконуються на комп'ютері, наданому університетом, то студент повинен виконувати наступні правила:

- забороняється змінювати налаштування операційної системи та програмного забезпечення, яке використовується для виконання практичних занять, окрім випадків, визначених викладачем;
- забороняється запускати стороннє програмне забезпечення, що не використовується для виконання практичних занять без дозволу викладача;
- у разі виникнення питань чи непередбаченої роботи обладнання необхідно повідомити викладача;
- забороняється підключати до комп'ютера власні накопичувачі на FLASH-пам'яті та копіювати з них на університетський комп'ютер файли без дозволу викладача.

Практичне заняття № 1

МОДЕЛЮВАННЯ РОБОТИ ЛОГІЧНИХ СХЕМ ЗА ДОПОМОГОЮ САПР PROTEUS

Мета заняття

Навчитись моделювати роботу логічних схем в САПР «Proteus Design Suite».

1. Короткі теоретичні відомості

Proteus Design Suite – це система автоматизованого проектування (САПР), яка представляє собою пакет програм, що об'єднує в собі дві основні програми: [1]

- ISIS – модуль для розробки та моделювання аналогових та цифрових електронних схем в режимі реального часу;
- ARES – модуль для розробки й автоматичного трасування друкованих плат.

Основні функції Proteus: [1]

- створення й редагування принципів схем;
- моделювання роботи розробленого пристрою;
- розробка й редагування друкованої плати.

2. Етапи розробки проекту цифрового пристрою

Перед створенням проекту необхідно створити папку для зберігання файлів проекту (наприклад, PZ1). Бажано, усі імена для папки та шляху до неї називати тільки з використанням англійських букв.

1 етап. Створення нового проекту

1. Запустити САПР Proteus і у вікні «Start» вибрати меню «New Project» або в меню «File» вибрати пункт «New Project».

2. У поле «Name» вести ім'я проекту: pz1.pdsprj, а в поле «Path» вибрати створену заздалегідь папку для зберігання проекту (папку можна вибрати нажавши кнопку «Browse»). Натиснути кнопку «Next».

3. Вибрати шаблон для розробки принципової схеми (Create a schematics from the selected template): за замовчуванням Default. Натиснути кнопку «Next».

4. Вибрати шаблон для розробки друкованої плати: Do not create a PCB layout (не створювати друковану плату). Натиснути кнопку «Next».

5. Вибрати тип проекту з мікропроцесором або мікроконтролером: No Firmware Project (без мікроконтролера). Натиснути кнопку «Next».

6. Перевірити остаточно всі налаштування проекту та натиснути кнопку «Finish». Якщо якісь параметри встановлені неправильно, можна повернутись назад натиснувши кнопку «Back».

Після цього на екрані з'явиться закладка «Schematic Capture», що призначена для малювання принципової схеми.

2 етап. Створення принципової схеми

1. Перейти на закладку Schematic Capture.

2. Створити базу даних компонентів необхідних для проектування принципової схеми логічного пристрою:

– перейти в режим «Component Mode» – друга зверху іконка у полі інструментів або у вікні «Devices» натиснути синю кнопку «P»;

– з'явиться вікно «Pick Devices», у якому у вікні «Category» будуть згруповані всі доступні компоненти. Після вибору категорії, у правому вікні «Results» з'являється список компонентів. Його можна уточнити по підкатегоріям «Sub-category» і по виробниках компонентів «Manufacturer»;

– після вибору компонента у вікні «Results», з'являється його графічне позначення у вікні «Preview». Для того щоб додати компонент в базу даних необхідно два рази клацнути на ньому лівою кнопкою миші. Після вибору усіх необхідних компонентів необхідно натиснути кнопку «OK» для закриття вікна «Pick Devices»;

– внести в базу даних наступні компоненти (Категорія → Підкатегорія → Ім'я компонента):

– інвертор: Modelling Primitives → INVERTER;

– логічний елемент 2І (2AND): Modelling Primitives → AND_2;

– логічний елемент 3І (3AND): Modelling Primitives → AND_3;

– логічний елемент 4І (4AND): Modelling Primitives → AND_4;

– логічний елемент 2АБО (2OR): Modelling Primitives → OR_2;

– логічний елемент АБО (OR): Modelling Primitives → BUFFER;

– індикатор логічного стану (Logic State Indicator): Debugging Tools → LOGICPROBE;

– джерело логічних сигналів (Logic State Source): Debugging Tools → LOGICSTATE.

Закрити вікно натиснувши кнопку «OK».

3. Намалювати принципову схему згідно заданого варіанту:

– розмістити необхідні логічні елементи на листі (вибрати компонент у вікні «Devices», а потім натиснути ліву клавішу миші на листі принципової схеми);

– з'єднати компоненти згідно заданої принципової схеми;

– підключити до кожного входу схеми джерело логічних сигналів;

– підключити до кожного виходу схеми індикатор логічного стану.

З'єднання компонентів:

– навести курсор на кінець виводу першого компонента, при цьому курсор прийме вид олівця, а на виводі з'явиться червоний квадрат;

– натиснути один раз ліву кнопку миші, і прокласти лінію з'єднання до виводу другого компонента. Якщо необхідно виконати поворот провідника необхідно щоразу натискати ліву кнопку миші;

– коли на виводі другого компонента з'явиться червоний квадрат, натиснути ліву кнопку миші кнопку ще раз, з'єднання компонентів буде завершено;

– для видалення з'єднання, необхідно виділити провідник (він стане червоним) і натиснути клавішу «Delete»;

– якщо провідників занадто багато, те краще для з'єднання елементів схеми використовувати шину. Для цього необхідно включити режим «Buses Mode» (шоста зверху іконка у полі інструментів). Початок шини наноситься шляхом натискання лівої клавіші миші, для повороту шини необхідно натиснути ще раз ліву клавішу миші, для закінчення малювання шини необхідно двічі натиснути ліву кнопку миші;

– для з'єднання провідників підключених до шини, необхідно призначити їм однакові імена. Для цього необхідно включити режим «Wire Label Mode» (четверта зверху іконка у полі інструментів)) або вибрати провідник, натиснути праву клавішу й вибрати контекстне меню «Place Wire Label». Відкриється вікно «Edit Wire Label», у поле «String» необхідно ввести ім'я провідника й натиснути кнопку «ОК».

3 етап. Моделювання роботи пристрою

1. Для запуску моделювання роботи проектного пристрою необхідно зайти в меню «Debug» або скористатися чотирма швидкими кнопками, які розташовані ліворуч у нижньому рядку програми.

У режимі моделювання симулятор дозволяє виконувати наступні операції:

– моделювання роботи пристрою в автоматичному режимі з підтримкою анімації: меню «Debug / Run Simulation», швидка кнопка «▶» або клавіша F12;

– покрокове моделювання роботи пристрою: меню «Debug / Start VSM Debugging», швидка кнопка «▶» або комбінація клавіш CTRL+F12. Використовується в основному для покрокового виконання програми мікроконтролера;

– для тимчасової зупинки роботи пристрою необхідно включити режим «PAUSE»: меню «Debug / Pause VSM Debugging», швидка кнопка «||» або клавіша «Pause»;

– для виходу з режиму моделювання роботи пристрою необхідно його вимкнути: меню «Debug / Stop VSM Debugging», швидка кнопка «■» або клавіші SHIFT + «Pause».

2. Провести моделювання роботи логічного пристрою, встановивши відповідності вхідних та вихідних сигналів:

– за допомогою джерел логічних сигналів (Logic State Source) необхідно подавати на входи логічної схеми потрібні комбінації сигналів (логічні сигнали перемикаються при натисканні на компоненті лівої кнопки миші);

– логічні сигнали на виходах пристрою контролюється за допомогою індикаторів логічного стану (Logic State Indicator);

– заповнити таблицю 1.1 для усіх можливих вхідних комбінацій.

3. Зміст звіту

1. Принципова схема цифрового пристрою на логічних елементах згідно заданого варіанту. Варіанти завдань до практичного заняття № 1 приведені у додатковому файлі який можна скачати в системі дистанційної освіти «Лідер».

2. Таблиця відповідності вхідних та вихідних сигналів цифрового пристрою.

Таблиця 1.1 – Логічна функція відповідності вхідних та вихідних сигналів цифрового пристрою

Входи				Виходи				
X3	X2	X1	X0	Y0	Y1	–	–	Yn
0	0	0	0					
0	0	0	1					
0	0	1	0					
0	0	1	1					
0	1	0	0					
0	1	0	1					
0	1	1	0					
0	1	1	1					
1	0	0	0					
1	0	0	1					
1	0	1	0					
1	0	1	1					
1	1	0	0					
1	1	0	1					
1	1	1	0					
1	1	1	1					

4. Контрольні запитання

1. Призначення САПР Proteus Design Suite.
2. Можливості САПР Proteus Design Suite.
3. Етапи розробки проекту цифрового пристрою в САПР Proteus Design Suite.
4. Умовне графічне позначення логічного елемента 2AND згідно стандарту ANSI 91–1984.
5. Принцип роботи логічного елемента 2AND (2I).
6. Умовне графічне позначення логічного елемента 2OR згідно стандарту ANSI 91–1984.
7. Принцип роботи логічного елемента 2OR (2АБО).
8. Умовне графічне позначення логічного елемента 2XOR згідно стандарту ANSI 91–1984.
9. Принцип роботи логічного елемента 2XOR (додавання по модулю два).
10. Умовне графічне позначення логічного елемента NOT згідно стандарту ANSI 91–1984.

Практичне заняття № 2

ВВЕДЕННЯ АНАЛОГОВОГО СИГНАЛУ В КОМП'ЮТЕР ДЛЯ ЦИФРОВОЇ ОБРОБКИ СИГНАЛІВ

Мета заняття

Вивчити схемотехнічні рішення введення аналогового сигналу в цифрові системи.

1. Короткі теоретичні відомості

Усі цифрові системи працюють із сигналами, що мають дискретне значення (лог.1 та лог.0), тому вони не здатні напряму працювати з аналоговими сигналами. Для обробки аналогових сигналів за допомогою цифрової системи, необхідно попередньо оцифрувати сигнал, тобто провести над ним операції дискретизації, квантування та кодування. Усі сучасні пристрої вводу аналогового сигналу в цифрові системи будуються на базі аналого–цифрових перетворювачів (АЦП), тобто спеціалізованих мікросхемах призначених для перетворення аналогового сигналу в пропорційний йому цифровий код. Сучасні досягнення мікроелектроніки дозволяють випускати функціонально завершені АЦП на одному кристалі, здатні працювати в автономному режимі без додаткових зовнішніх елементів. Окрім АЦП, до складу пристрою вводу аналогового сигналу в комп'ютер входять джерело живлення, а також дискретні елементи, що виконують допоміжні функції корекції сигналу, узгодження рівнів та захисту від перенапруги (рис. 2.1).

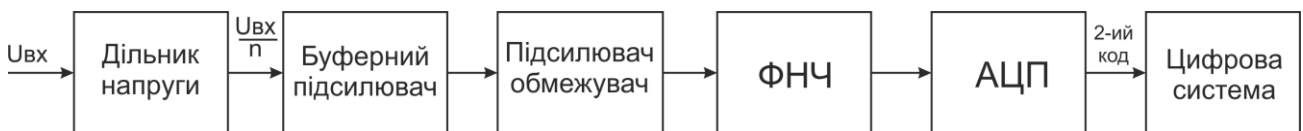


Рис. 2.1. Структурна схема вводу аналогового сигналу в цифрові системи

Дільник напруги (рис. 2.2) призначений для узгодження рівнів вхідного сигналу та сигналу, що подається на вхід наступного каскаду пристрою вводу аналогового сигналу, і представляє собою звичайний резистивний дільник напруги. Напруга на виході дільника визначається виразом (2.1)

$$U_{\text{вих}} = U_{\text{вх}} \frac{R_2}{R_1 + R_2} \quad (2.1)$$

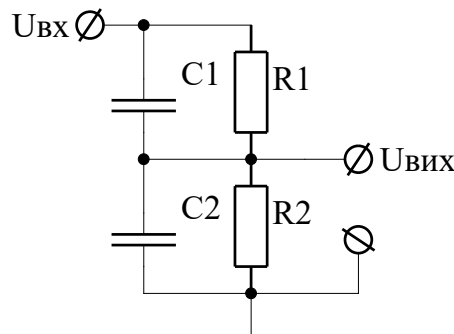
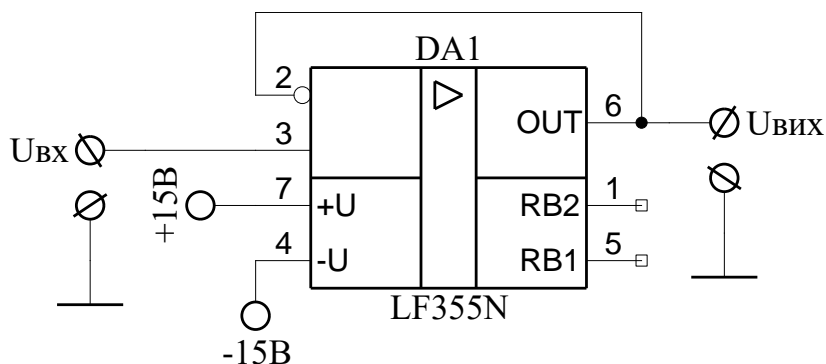


Рис. 2.2. Схема дільника напруги

Конденсатори $C1$ та $C2$ необхідні для компенсації ємнісної складової резисторів $R1$ та $R2$ при подачі на вхід пристрою височастотних сигналів (більше 10 МГц). Для вхідних сигналів постійного струму та низькочастотних сигналів конденсатори можна не встановлювати.

Буферний підсилювач (рис. 2.3) забезпечує разом з дільником напруги високий вхідний опір схеми (1 – 10 МОм), так як пристрій вводу аналогового сигналу в комп'ютер представляє собою вимірювальний пристрій (типу вольтметр), і найчастіше підключається до схеми працюючого електронного пристрою. Чим більше вхідний опір вимірювального пристрою, тем менше він впливає на роботу схеми, в якій проводяться вимірювання. Буферний підсилювач реалізується на операційному підсилювачі (ОП), що включений по схемі повторювача з коефіцієнтом підсилення $K \approx 1$ (підсилювач із 100% зворотним зв'язком). [2]



$$U_{вх} = U_{вих}$$

Рис. 2.3. Схема буферного підсилювача

Підсилювач – обмежувач (рис. 2.4) забезпечує підсилення вхідного сигналу до необхідного рівня та захисту АЦП від перевантаження по напрузі. Якщо напруга на вході підсилювача–обмежувача не перевищує напруги пробою стабілітронів $VD1$ і $VD2$, то ОП працює як звичайний підсилювач із коефіцієнтом підсилення, що дорівнює $K = -R3/R4$. Коли вхідна напруга перевищить напругу стабілізації $VD1$ або $VD2$, то відбувається їхній пробій і наступить обмеження напруги на виході підсилювача–обмежувача, тобто вона буде залишатися незмінною і незалежною від вхідної напруги. [2]

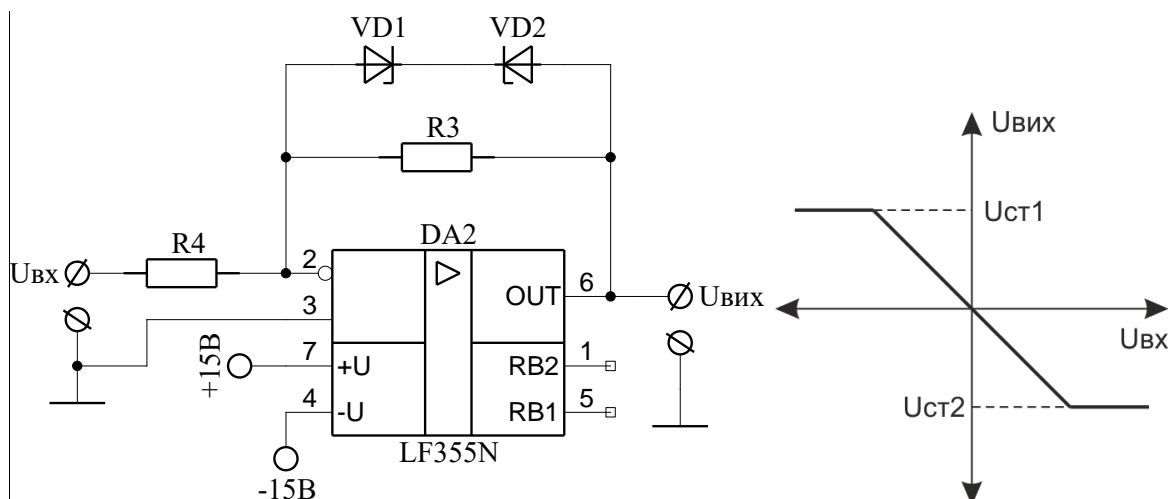


Рис. 2.4. Схема підсилювача–обмежувача та його передаточна характеристика

Фільтр нижніх частот (ФНЧ) призначений для усунення ефекту накладення спектрів, який виникає при оцифровці аналогового сигналу. Відповідно до теореми Найквіста (Котельникова) частота дискретизації сигналу F_s повинна бути більш ніж у два рази вище частоти даного сигналу. У результаті впливу на вхідний сигнал високочастотних завад і пульсацій від імпульсних блоків живлення, у ньому можуть бути присутні складові (гармоніки), частота яких вище $F_s/2$. Якщо ці гармоніки потраплять на вхід АЦП, то позбутися їх цифровими методами буде вже неможливо. На рис. 2.5,а показаний частотний спектр вхідного сигналу, що має п'ять гармонік. При подачі такого сигналу на вхід АЦП, і дискретизації сигналу із частотою F_s , АЦП однаково сприйме гармоніки із частотою нижче й вище $F_s/2$. На рис. 2.5,б показано спектр вихідного цифрового сигналу. Як видно на рисунку, гармоніка (1) буде оцифрована вірно, а гармоніки (2) – (5) будуть перенесені («згорнуті») у низькочастотний спектр і проявляться у вигляді низькочастотної складової, що призведе до викривлення частотного спектру дискретизованого сигналу. [3]

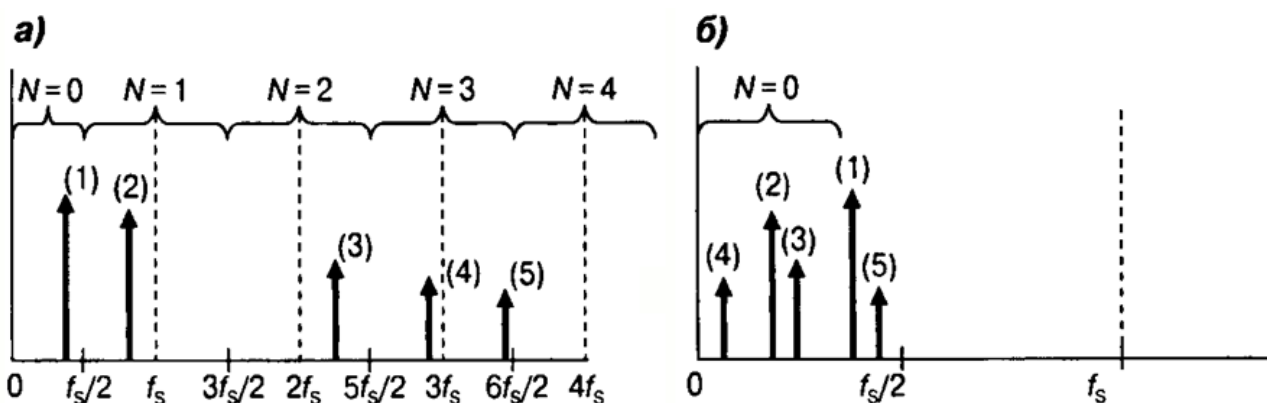


Рис. 2.5. Ефект накладення спектра при дискретизації аналогового сигналу: а) частотний спектр аналогового вхідного сигналу; б) частотний спектр сигналу в смузі від 0 до $F_s/2$, що буде відповідати дискретному вихідному коду

Гармоніки з частотою вище $F_s/2$ згортаються у відповідності з формулою:

$$F_a = | F_n - N * F_s |.$$

Наприклад, при частоті дискретизації $F_s = 10$ кГц і частотному спектрі вхідного сигналу $F1 = 4$ кГц, $F2 = 8$ кГц, $F3 = 23$ кГц, $F4 = 29$ кГц, $F5 = 35$ кГц, вихідний оцифрований сигнал буде мати наступний спектр:

$$F1a = | 4 - 0 * 10 | = 4 \text{ кГц};$$

$$F2a = | 8 - 1 * 10 | = 2 \text{ кГц};$$

$$F3a = | 23 - 2 * 10 | = 3 \text{ кГц};$$

$$F4a = | 29 - 3 * 10 | = 1 \text{ кГц};$$

$$F5a = | 35 - 4 * 10 | = 5 \text{ кГц}.$$

Якщо на вході АЦП поставити ФНЧ (антиелайзінговий фільтр) з частотою зрізу $F_c = F_s/2$ (рис. 2.6), то гармоніки (2) – (5) не попадуть на вхід АЦП і вхідний сигнал буде оцифрований без спотворень.

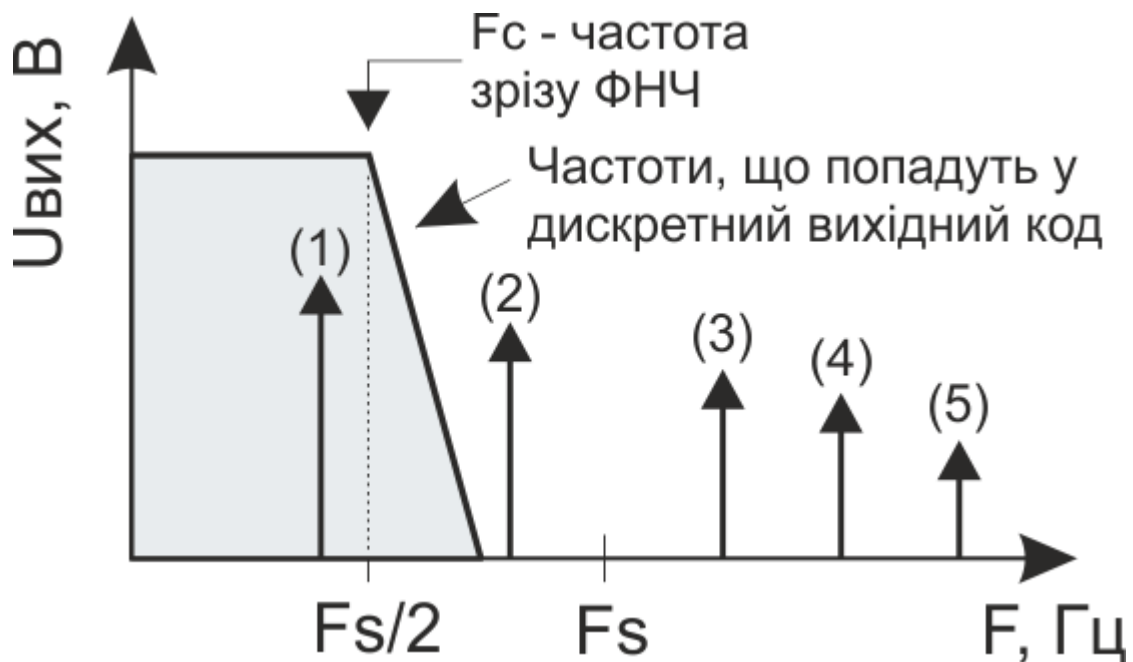


Рис. 2.6. АЧХ аналогового ФНЧ, що усуває ефект накладання спектрів

Для реалізації ФНЧ використовуються активні фільтри на ОП. В цей час найчастіше використовується модифікована схема Саллена – Кі (Sallen – Key), яка дозволяє реалізувати фільтр другого порядку будь-якого типу: фільтр Чебишева, фільтр Бесселя та фільтр Баттерворта (рис. 2.7). Крім того, схема Саллена – Кі не інвертує сигнал, а також може виконувати підсилення сигналу. Параметри фільтра визначаються значенням пасивних елементів R , R_1 , R_f і C . Якщо з'єднати дві такі схеми каскадно, то вийде фільтр четвертого порядку, а якщо з'єднати три такі схеми каскадно, то вийде фільтр шостого порядку і так далі. [4]

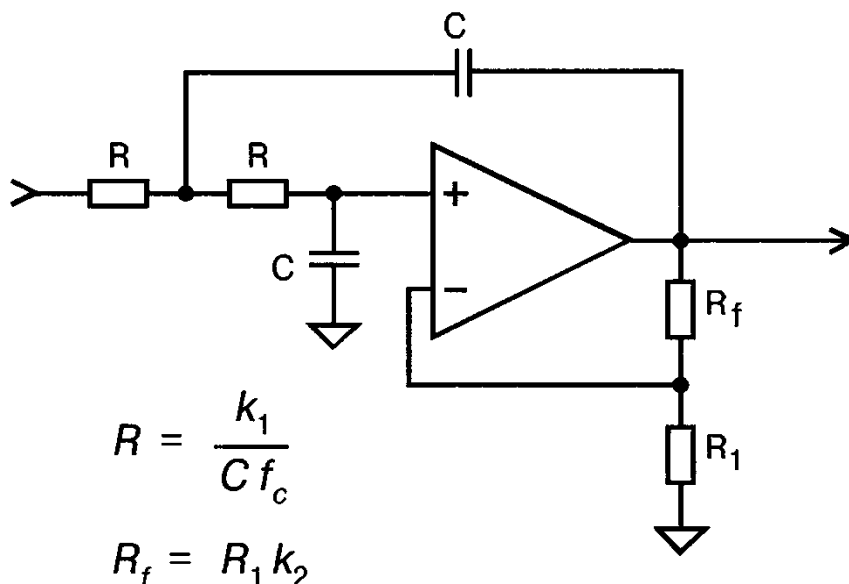


Рис. 2.7. Схема фільтра низьких частот Саллена–Кі

На рис. 2.8 приведені приклади імпульсних характеристик ФНЧ Чебишева, Бесселя і Баттерворта. Фільтр Чебишева має найкрутіший підйом та спад частотної характеристики, але має нерівномірну АЧХ у смузі пропускання. Фільтр Бес-

сея має кращу перехідну характеристику (перехідні процеси швидко закінчуються при невеликих викидах), але має досить полого частотну характеристику. Фільтр Баттерворта має максимально плоску АЧХ у смузі пропускання, але також має найгіршу крутість підйому та спаду частотної характеристики.

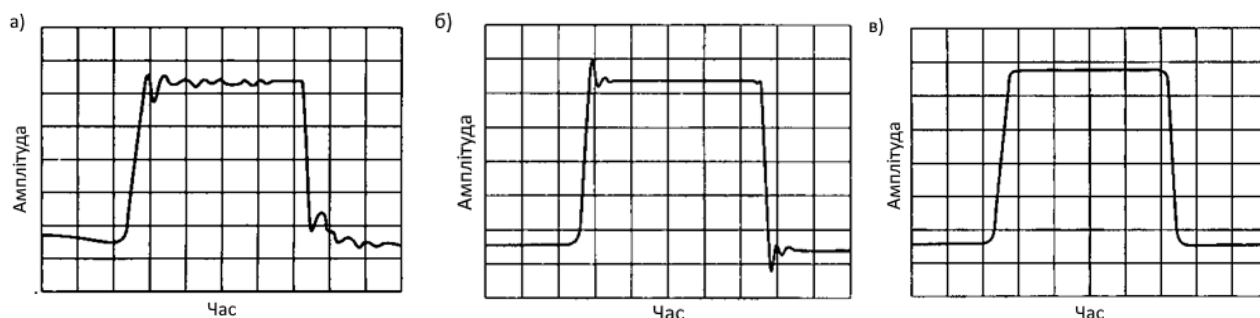


Рис. 2.8. Імпульсні характеристики ФНЧ: а) фільтр Чебишева, б) фільтр Бесселя і в) фільтр Баттерворта

Для розрахунку ФНЧ необхідно визначити необхідну частоту зрізу фільтра F_c , а також визначитись з типом і порядком фільтру, щоб можна було вибрати коефіцієнти фільтру k_1 і k_2 з таблиці 2.1. [3]

Потім необхідно задатись типовими значеннями резистора R_1 і конденсатора C (наприклад, $R_1 = 10$ кОм, $C = 0,01$ мкФ) і розрахувати по формулам, що приведені на рис. 2.7 значення резисторів R і R_f . Якщо після розрахунків значення резистора R виявиться менше 1 кОм, то необхідно зменшити значення конденсатора C (наприклад, в 10 разів) і повторити розрахунок значення резистора R .

Таблиця 2.1 – Коефіцієнти для розрахунку схеми ФНЧ Саллена–Кі

Кількість полюсів		Фільтр Бесселя		Фільтр Баттерворта		Фільтр Чебишева	
		k_1	k_2	k_1	k_2	k_1	k_2
2	Коло 1	0,1251	0,268	0,1592	0,586	0,1293	0,842
	Коло 2	0,0991	0,759	0,1592	1,235	0,1544	1,660
6	Коло 1	0,0990	0,040	0,1592	0,068	0,4019	0,537
	Коло 2	0,0941	0,364	0,1592	0,586	0,2072	1,448
	Коло 3	0,0834	1,023	0,1592	1,483	0,1574	1,846
8	Коло 1	0,0894	0,024	0,1592	0,038	0,5359	0,522
	Коло 2	0,0867	0,213	0,1592	0,337	0,2657	1,379
	Коло 3	0,0814	0,593	0,1592	0,889	0,1848	1,711
	Коло 4	0,0726	1,184	0,1592	1,610	0,1582	1,913

АЦП К1113ПВ1 представляє собою напівпровідникову інтегральну мікросхему (ІМС), яка призначена для використання в пристроях введення аналогового сигналу в цифрові системи. ІМС К1113ПВ1 представляє собою 10–розрядний АЦП, що працює з однополярними і двуполярними вхідними напругами з представлен-

ням результату в паралельному двійковому коді. АЦП К1113ПВ1 відноситься до класу АЦП миттєвого значення, і є АЦП послідовного наближення. [5]

Умовне графічне позначення (УГП) АЦП типу К1113ПВ1 та часова діаграма його роботи приведені на рис. 2.9. Переключення діапазону вхідного сигналу здійснюється шляхом подачі керуючого сигналу на вхід O/D. Процес перетворення аналогового сигналу в цифровий код запускається шляхом подачі на вхід START сигналу «ЗАПУСК» (лог.0). Для скидання поточного вихідного коду, перед запуском перетворення, необхідно подати на вхід START сигнал логічної одиниці тривалістю не менше 2 мкс. У процесі перетворення на виході АЦП READY підтримується сигнал логічної одиниці. По закінченню перетворення на шині даних АЦП з'явиться двійковий цифровий код пропорційний вхідному аналоговому сигналу, а на виході READY сигнал лог.0. При перетворенні негативних напруг результат виводиться на шину даних (ШД) АЦП у додатковому коді.

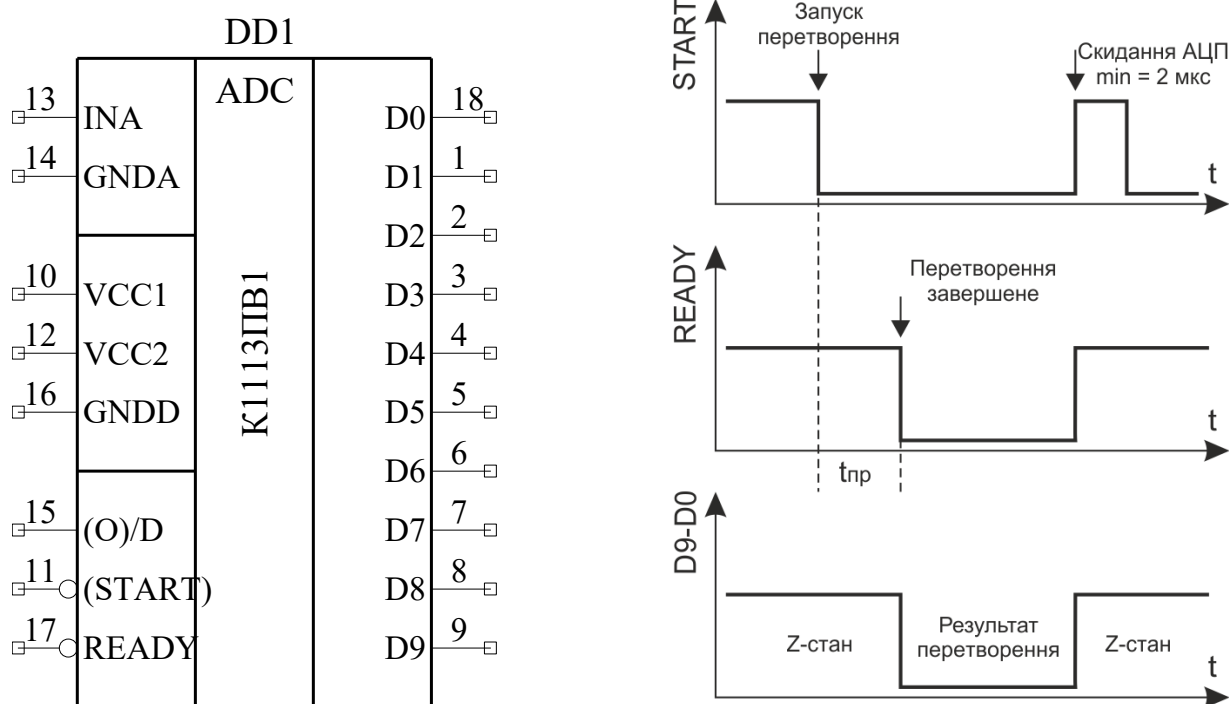


Рис. 2.9. УГП АЦП типа К1113ПВ1 і часова діаграма роботи

Призначення виводів АЦП К1113ПВ1:

D0 – D9 – цифрові виходи з Z-станом;

O/D – вибір типу вхідного сигналу. При подачі на цей вхід «цифрової землі» на вхід АЦП можна подавати однополярний сигнал у діапазоні 0 – +10,24 В, якщо вивід 15 нікуди не підключений (знаходиться в повітрі), то на вхід можна подавати біполярний сигнал у діапазоні –5,12 ÷ +5,12 В;

START – вхід запуску перетворення: лог.1 – скидання АЦП, лог.0 – запуск перетворення АЦП;

READY – вихід готовності даних: лог.1 – результат перетворення ще не готовий, лог.0 – перетворення завершено;

INA – вхід аналогового сигналу (надходить щодо аналогової землі GNDA);

VCC1 – вивід напруги живлення: +5 В ± 5% ;

VCC2 – вивід напруги живлення: –15 В ± 5%;

GNDA – аналогова земля;

GNDD – цифрова земля.

Параметри АЦП типа К1113ПВ1:

- розрядність – 10 біт;
- диференційна нелінійність – 0,1%;
- час перетворення – 25 мкс (частота дискретизації 40 кГц);
- напруга зсуву нуля – не більше 0,3 В;
- діапазон вхідних напруг: 0 ÷ 10,24 В або –5,12 ÷ +5,12 В;
- вхідний опір – 10 кОм;
- споживаний струм по колу VCC1 – не більше 10 мА;
- споживаний струм по колу VCC2 – не більше 20 мА;
- вихідні логічні сигнали – ТТЛ – стандарт.

Найбільш великими виробниками АЦП є компанії Analog Devices (USA), Micro Power (USA), Maxim, Philips, Texas Instruments та ін. Сучасні АЦП мають розрядність до 24 розрядів, але найпоширеніші 8, 10, 12 і 16 розрядні АЦП. Максимальна частота перетворення може перевищувати 100 МГц.

2. Дослідження точності перетворення пристрою вводу аналогового сигналу в комп'ютер на базі АЦП К1113ПВ1

1. Підключити для проведення дослідження пристрій введення аналогового сигналу до комп'ютера та блока живлення.
2. Увімкнути персональний комп'ютер і блок живлення пристрою вводу аналогового сигналу в комп'ютер.
3. Запустити програму: ADC_DAC.exe.
4. Регулюючи напругу на вході пристрою заповнити таблицю 2.2.
5. Визначити точність перетворення в заданому діапазоні напруг із кроком вимірювання 0,1 В (діапазон вимірювань задається викладачем).

Таблиця 2.2 – Результати вимірювань

$U_{\text{вх}}, \text{В}$	ШД АЦП шістнадцятковий код	ШД АЦП двійковий код	$U_{\text{вим}}, \text{В}$	$\delta, \%$
0				
0,1				
0,2				
1,9				
2,0				

3. Зміст звіту

1. Структурна схема пристрою вводу аналогового сигналу в цифрові системи з коротким описом призначення блоків.
2. Принципові схеми резистивного дільника напруги, буферного підсилювача, підсилювача–обмежувача і модифікованого ФНЧ Саллена – Кі, розрахованого по

заданому варіанту.

3. Розрахунки схеми ФНЧ Саллена – Кі згідно заданого варіанту (таблиця 2.3).
4. УГП АЦП типу K1113ПВ1, основні параметри та призначення виводів.
5. Результати вимірювань в таблиці 2.2 (з кроком $\Delta U = 0,1 \text{ В}$):
 - 1 група: 0 – 2 В;
 - 2 група: 2 – 4 В;
 - 3 група: 4 – 6 В;
 - 4 група: 6 – 8 В;
 - 5 група: 8 – 10 В.

Таблиця 2.3 – Варіанти завдань для розрахунків ФНЧ Саллена – Кі

Номер варіанта	Тип фільтра	Частота зрізу F_c , кГц
1	Фільтр Чебишева	10
2	Фільтр Бесселя	20
3	Фільтр Баттерворта	50
4	Фільтр Бесселя	10
5	Фільтр Чебишева	20
6	Фільтр Баттерворта	10
7	Фільтр Бесселя	50
8	Фільтр Баттерворта	20
9	Фільтр Чебишева	50
10	Фільтр Бесселя	40

4. Контрольні запитання

1. Призначення дільника напруги.
2. Призначення буферного підсилювача.
3. Призначення підсилювача – обмежувача.
4. Які елементи підсилювача – обмежувача задають значення напруги обмеження на виході схеми.
5. Призначення фільтра нижніх частот у пристрої введення аналогового сигналу в комп'ютер.
6. Призначення АЦП у пристрої введення аналогового сигналу в комп'ютер.
7. Який мінімальний вхідний опір повинен мати пристрій введення аналогового сигналу в комп'ютер.
8. Яку частоту зрізу повинен мати антиалайзінговий фільтр (ФНЧ), якщо частота дискретизації аналогового сигналу становить 10 кГц.
9. На скільки рівнів квантування може розбити аналоговий сигнал 10-розрядний АЦП.
10. Сигнал якої максимальної частоти можна подавати на вхід АЦП, якщо його максимальна частота дискретизації 40 кГц.

Практичне заняття № 3

ПРОГРАМНО–КЕРОВАНІ ДЖЕРЕЛА ЖИВЛЕННЯ ТА ГЕНЕРАТОРИ СИГНАЛІВ ДОВІЛЬНОЇ ФОРМИ

Мета заняття

Вивчити схемотехніку й принцип роботи програмно–керованого джерела живлення (ПКДЖ) та програмно–керованих генераторів сигналів довільної форми, а також провести дослідження точності перетворення ПКДЖ на базі ЦАП К572ПА1.

1. Програмно–кероване джерело живлення

Програмно–кероване джерело живлення формує на виході напругу постійного струму, значення якої задається вхідним цифровим кодом, що надходить від цифрового пристрою (мікроконтролера або комп'ютера). Програмно–керовані джерела живлення використовуються в сучасних джерелах живлення із цифровим керуванням, контрольно–вимірювальних пристроях, програматорах, тобто в цифрових пристроях в яких потрібно формувати різні аналогові сигнали (разові або періодичні). Усі програмно–керовані джерела живлення будуються на базі цифро–аналогових перетворювачів (ЦАП), які дозволяють перетворювати цифровий код в аналоговий сигнал.

Принципова схема програмно–керованого джерела живлення, що дозволяє формувати напругу постійного струму від 0 до 51,15 В с кроком 50 мВ, наведена на рис. 3.1.

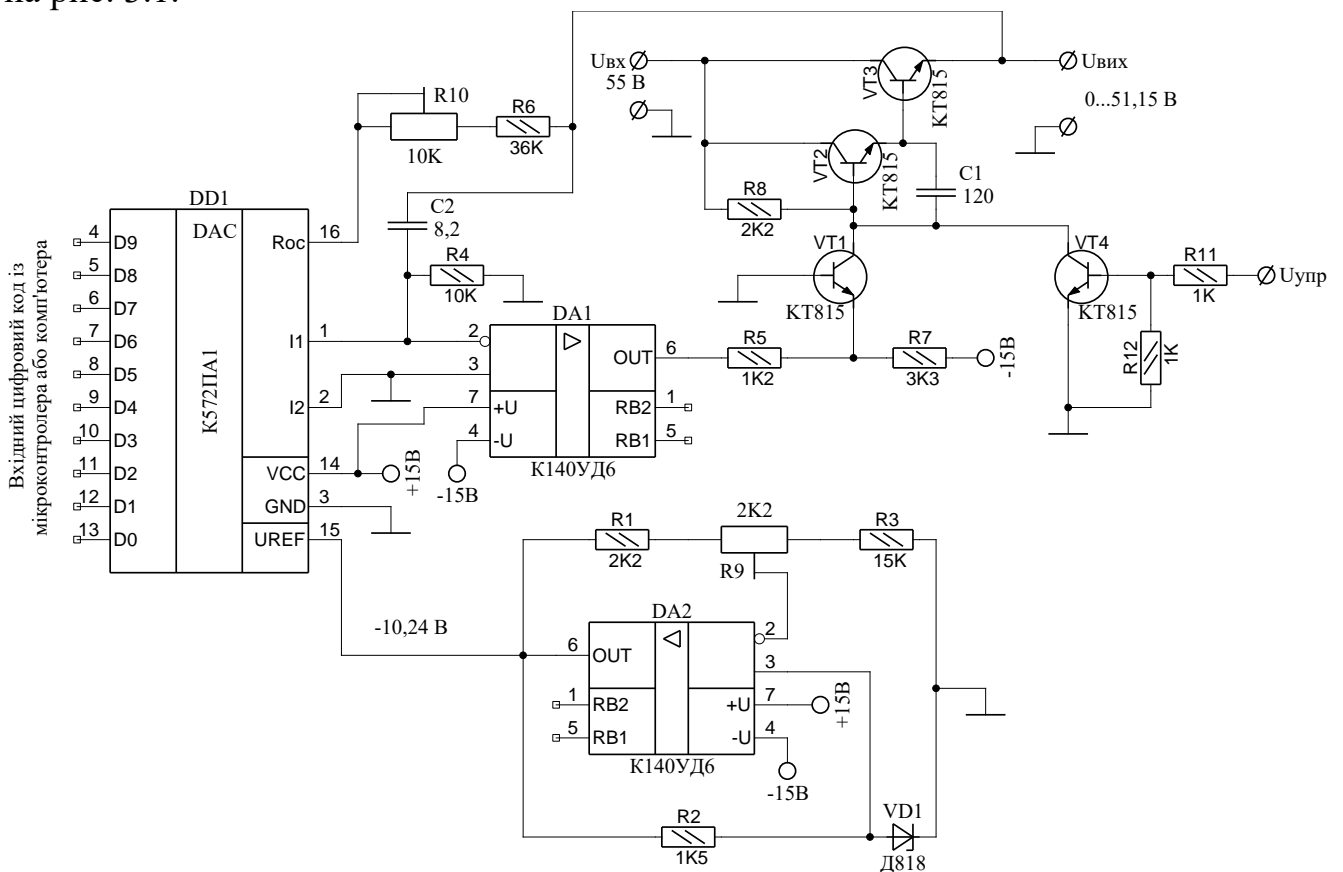


Рис. 3.1. Схема програмно–керованого джерела живлення

Основним вузлом програмно–керованого джерела живлення є ЦАП DD1 типу К572ПА1 [5], який призначений для перетворення 10–розрядного паралельного двійкового коду в аналоговий струм, пропорційний значенням двійкового коду на входах D0–D9 і опорної напруги на вході UREF(–10,24 В). До складу ЦАП входить матриця резисторів типу R–2R, підсилювачі–інвертори для керування струмовими ключами та струмові двопозиційні ключі, що виконані на КМОП транзисторах. ЦАП К572ПА1 має наступні переваги:

- висока точність перетворення (диференціальна нелінійність становить $\pm 0,1\%$);
- мала потужність споживання (струм споживання не більш 2 мА);
- висока швидкість перетворення (час встановлення вихідного струму не більш 5 мкс);
- низька вартість.

Напруга на виході програмно–керованого джерела живлення визначається виразом:

$$U_R = h(2^{(n-1)} a_{n-1} + 2^{(n-2)} a_{n-2} + \dots + 2^0 a_0), \quad (3.1)$$

де h – крок квантування, тобто збільшення вихідної напруги при зміні вхідного коду на одиницю, n – число розрядів ЦАП, a_i – значення i -ого біта коду на цифровому вході ЦАП.

Крок квантування вихідної напруги джерела живлення визначається наступним виразом:

$$h = U_{on} \frac{R_{oc}}{R} 2^n, \quad (3.2)$$

де U_{on} – опорна напруга, що надходить на вхід матриці резисторів R–2R ЦАП, R – значення опору матриці R–2R ЦАП, R_{oc} – значення опору кола зворотного зв'язку ЦАП, яке представляє собою суму значень внутрішнього резистора зворотного зв'язку самого ЦАП (10 кОм) і зовнішнього опору зворотного зв'язку (резистори R10 і R6).

Як впливає з виразів (3.1) і (3.2), максимальна вихідна напруга визначається двома складовими: значенням опорної напруги й значенням співвідношення R_{oc}/R . Відповідно до технічних характеристик ЦАП К572ПА1, опорна напруга не може бути більше ± 15 В, тому максимальний діапазон вихідної напруги встановлюється шляхом зміни значення зовнішнього опору кола зворотного зв'язку. Зокрема, при опорній напрузі $U_{on} = -10,24$ В та опору зворотного зв'язку $R_{oc} = 51,15$ кОм, максимальна напруга на виході буде мати значення 51,15 В. Так як резистор з таким опором не випускається, то в колі зворотного зв'язку використовується змінний резистор R10, за допомогою якого можна точно підстроїти значення максимальної вихідної напруги.

Із виразу (3.2) впливає, що для одержання мінімальної погрішності перетворення цифрового коду в аналоговий сигнал, необхідно мати стабільне значення опорної напруги. Для цього в схемі програмно–керованого джерела живлення використовується високостабільне джерело опорної напруги (ДОН) на операційному підсилювачі (ОП) DA2. Вихідна напруга даного ДОН зберігає стабільне значення 10,24 В при коливаннях напруги живлення в діапазоні від 11 до 18 В, при-

чому перекіс позитивної та від'ємної напруги живлення операційного підсилювача може також досягати до 7 В (рис. 3.2). Використання даного ДОН забезпечує досить високу точність роботи ЦАП ДК572ПА1. Стабілізацію напруги виконує стабілітрон $VD1$, а точне регулювання напруги на виході ДОН виконується за допомогою змінного резистора $R9$.

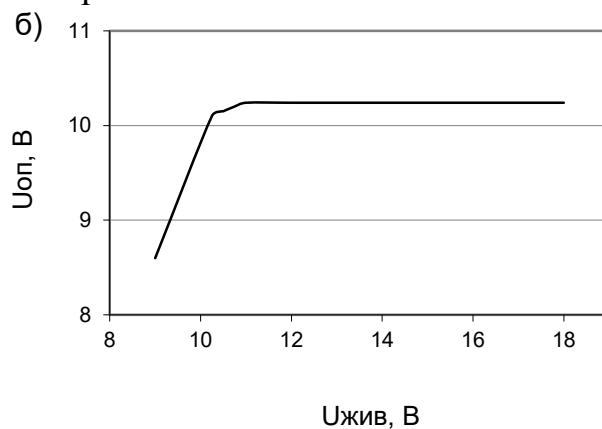


Рис. 3.2. Залежність вихідної напруги ДОН від напруги живлення

Принцип роботи. Мікросхема $DD1$ ЦАП К572ПА1 перетворює 10-ти розрядний паралельний двійковий код, який надходить на цифрові входи $D0-D9$ у струм на аналоговому виході $I1$. Вхідний струм ЦАП задається зовнішнім джерелом опорної напруги і послідовно ділиться у вузлах резистивної матриці $R-2R$ за двійковим законом. Двійковий закон розподілу струмів у колах резистивної матриці дотримується за умови рівності потенціалів виходів $I1$ і $I2$. Це забезпечується підключенням виходу $I1$ до інвертуючого входу ОП $DA1$, охопленого негативним зворотним зв'язком. Прямий вхід ОП $DA1$ з'єднується з виходом $I2$ ЦАП і з аналоговою землею. При цьому здійснюється перетворення струму з виходу ЦАП $I1$ у пропорційну йому напругу на виході ОП $DA1$.

Напруга керування з виходу ОП $DA1$ надходить через транзистор $VT1$ на базу транзисторів $VT2$ і $VT3$, які утворюють один складений транзистор великої потужності. Під дією цієї напруги складений транзистор змінює свій внутрішній опір (електричний опір переходу колектор – емітер), що дозволяє регулювати напругу на виході від 0 до 51,15 В. Наприклад, при значенні цифрового коду на вході ЦАП всі лог.0, транзистори $VT2$ і $VT3$ будуть повністю закриті і на виході схеми напруга буде дорівнювати нулю ($U_{\text{вих}} = 0$). А якщо на вході ЦАП всі сигнали будуть мати значення лог.1, то транзистори $VT2$ і $VT3$ будуть повністю відкриті і на виході схеми напруга буде мати максимальне значення ($U_{\text{вих}} = 51,15$ В). Інші цифрові коди між цими значеннями будуть формувати на виході схеми напругу постійного струму від 0 до 51,15 В з кроком 50 мВ.

Вхід $U_{\text{упр}}$ призначений для формування імпульсної напруги на виході джерела напруги. Коли на $U_{\text{упр}}$ сигнал низького рівня (лог.0), транзистор $VT4$ закритий, і на виході з'являється імпульс заданої амплітуди. Якщо на $U_{\text{упр}}$ сигнал високого рівня (лог.1), то транзистор $VT4$ відкритий, і напруга на виході ПКДЖ буде мати значення 0 В.

2. Програмно–керовані генератори сигналів довільної форми

Програмно–керовані генератори сигналів довільної форми дозволяють формувати аналогові сигнали заданої форми, амплітуди та частоти. Вони будуються на базі ЦАП, які дозволяють перетворювати цифровий код в аналоговий сигнал. Програмно–керовані генератори сигналів довільної форми будуються по одній з наступних структурних схем: 1) на базі двійкового лічильника (рис. 3.3); 2) на базі суматора з накопиченням (рис. 3.4).

2.1. Програмно–керований генератор сигналів довільної форми на базі двійкового лічильника

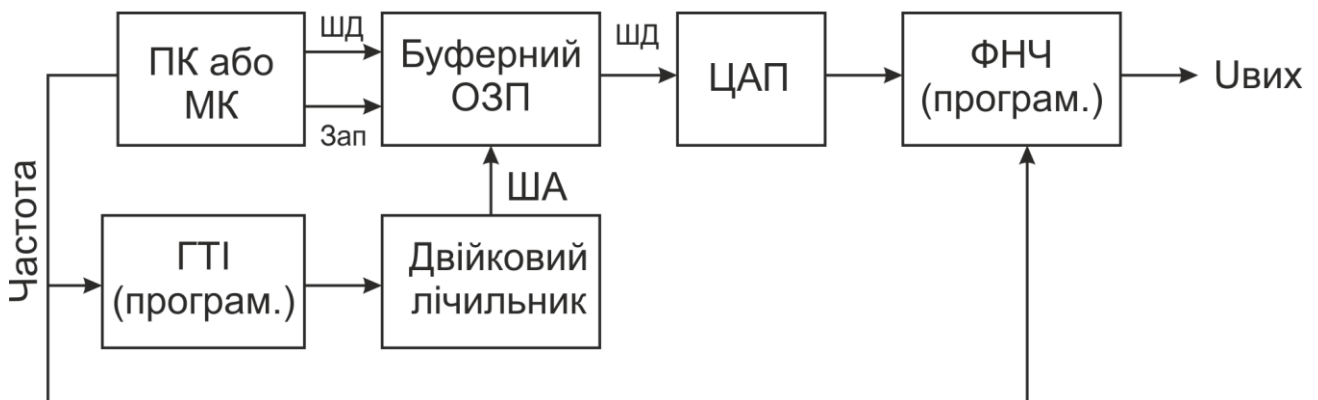


Рис. 3.3. Структурна схема генератора сигналів довільної форми на базі двійкового лічильника

Перед початком роботи комп'ютер (ПК) або мікроконтролер записує в буферний оперативний запам'ятовуючий пристрій (ОЗП) коди вибірок вихідного аналогового сигналу, що необхідно згенерувати, формуючи тим самим форму і амплітуду вихідного сигналу. У даній схемі генератор тактових імпульсів (ГТІ) та фільтр нижніх частот (ФНЧ) повинні бути програмно–керованими: у ГТІ записується частота вихідного сигналу, а також режим запуску генерації (разовий або періодичний).

ГТІ управляє двійковим лічильником, який перебирає підряд усі адреси буферного ОЗП, тому частота вихідного сигналу, буде визначатися частотою, з якої ці адреси перебираються. Коди вибірок з буферного ОЗП надходять на цифрові входи ЦАП із заданою частотою. ЦАП перетворює ці коди в рівні вихідного аналогового сигналу. Для фільтрації завад, що виникають через квантування вихідного сигналу, на виході генератора встановлений ФНЧ.

Перевагою даної схеми є те, що точність відтворення форми вихідного сигналу не залежить від його частоти, тому що в кожному разі задіюються всі чарунки ОЗП і кількість вибірок на період вихідного сигналу не змінюється.

Дана схема має один недолік, який полягає в тому, що частота завади квантування прямо пропорційна частоті вихідного сигналу, тому для фільтрації завад потрібно застосовувати спеціальні фільтри на ОП, з можливістю керування частотою зрізу. Це призводить до складності апаратної реалізації генераторів за даною схемою, а також вимагає застосування швидкодіючого ЦАП.

2.2. Програмно–керований генератор сигналів довільної форми на базі суматора з накопиченням

Схема генератора сигналів довільної форми на базі суматора з накопиченням відрізняється від попередньої способом завдання частоти вихідного сигналу. Для перебору адрес чарунок буферного ОЗП використовується суматор з накопиченням, що складається із двійкового суматора та регістру, охоплених зворотним зв'язком.

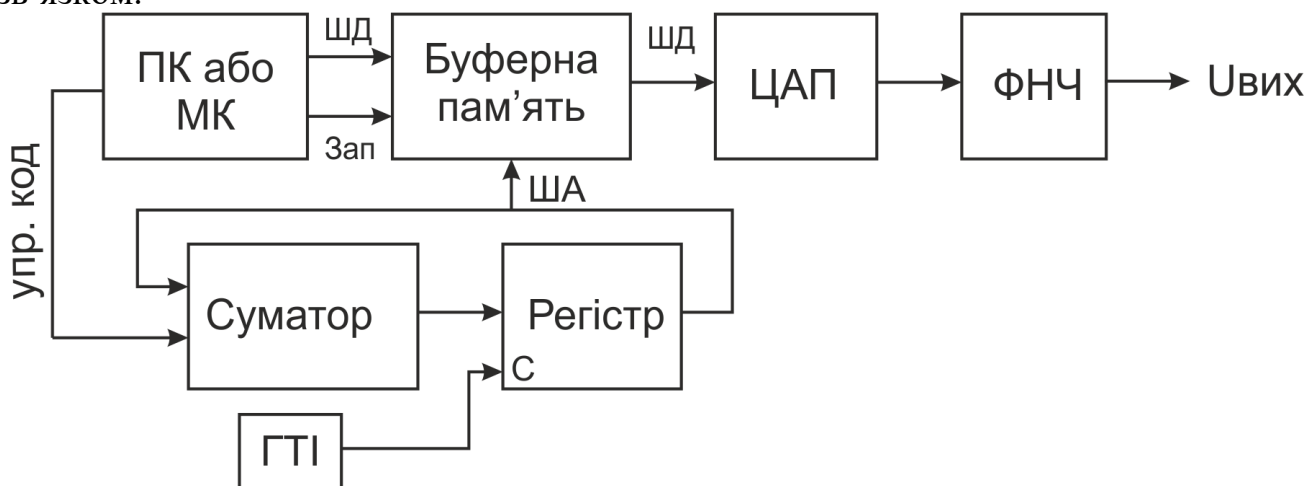


Рис. 3.4. Структурна схема генератора сигналів довільної форми на базі суматора з накопиченням

Такий спосіб завдання частоти вихідного сигналу простіше реалізується на апаратному рівні, тому що ГТІ та ФНЧ налаштовуються тільки на одну частоту, що спрощує їх схемотехніку. З кожним імпульсом ГТІ до вихідного коду регістру додається вхідний керуючий код N , і отримана сума знову записується в регістр. У результаті цього в кожному такті збільшення адреси чарунок ОЗП буде визначатися вхідним керуючим кодом суматора з накопленням суми. Змінюючи значення керуючого коду можна регулювати швидкість перебору адрес чарунок ОЗП а, отже, і частоту вихідного сигналу:

$$F_{\text{вих}} = \frac{F_{\text{ГТІ}} N}{M}, \quad (3.3)$$

де N – значення керуючого коду, M – кількість чарунок ОЗП.

Так як частота вихідного сигналу в даній схемі задається з постійним кроком у всьому частотному діапазоні, то відносна погрішність її установки максимальна в самому верху частотного діапазону, і мінімальна при найнижчих частотах. Перевагою даної схеми є те, що частота завади квантування постійна й дорівнює частоті сигналу ГТІ, тому її можна відфільтрувати найпростішим ФНЧ. Недолік схеми полягає в тому, що форма сигналу відтворюється з різною точністю на різних частотах, тому що кількість вибірок сигналу на період вихідного сигналу зворотно пропорційно його частоті (тобто чим вище частота, тем менше точність).

Наприклад, при $F_{\text{ГТІ}} = 2048$ кГц і $M = 2048$ (2 Кбайти):

$N = 1 \rightarrow F_{\text{вих}} = 1$ кГц (2048 вибірок);

$N = 2 \rightarrow F_{\text{вих}} = 2$ кГц (1024 вибірки);

$N = 10 \rightarrow F_{\text{вих}} = 10 \text{ кГц}$ (204 вибірки);
 $N = 2048 \rightarrow F_{\text{вих}} = 2048 \text{ кГц}$ (1 вибірка).

3. Дослідження точності перетворення ПКДЖ на базі ЦАП К572ПА1

1. Зібрати схему дослідження ПКДЖ (рис. 3.5).
2. Включити персональний комп'ютер і ПКДЖ.
3. Запустити програму: prov_relay.exe.
4. Перейти в тестовий режим і вибрати меню «Керування напругою на виході».
5. За допомогою миші натиснути на екрані монітора кнопки: «включити реле К2» та «вимкнути транзисторний ключ U1 ($U_1 = U_{\text{вих}}$)».
6. Регулюючи за допомогою комп'ютера напругу на виході ПКДЖ заповнити таблицю 3.1. Можливе покрокове регулювання коду на ШД ЦАП, або введення в текстовому полі коду в десятковій формі: покрокове регулювання напруги: «+1» – збільшення коду на ШД ЦАП на 1; «-1» – зменшення коду на ШД ЦАП на 1.
7. Визначити точність перетворення ПКДЖ у заданому діапазоні напруг із кроком вимірювання 0,5 В (діапазон вимірювання задається викладачем).



Рис. 3.5. Схема дослідження ПКДЖ

Таблиця 3.1 – Результати вимірювань

ШД ЦАП двійковий код	ШД ЦАП шістнадцятковий код	$U_0, \text{В}$	$U_{\text{вих}}, \text{В}$	$\delta, \%$

4. Зміст звіту

1. Принципова схема програмно–керованого джерела живлення й призначення елементів.
2. Структурна схема генератора сигналів довільної форми на базі двійкового лічильника й короткий опис його роботи.
3. Структурна схема генератора сигналів довільної форми на базі нагромаджуючого суматора й короткий опис його роботи.

4. Схема дослідження ПКДЖ і результати вимірювань в таблиці 3.1 (з кроком $\Delta U = 0,5 \text{ В}$):

- 1 група: 0 – 10 В;
- 2 група: 10 – 20 В;
- 3 група: 20 – 30 В;
- 4 група: 30 – 40 В;
- 5 група: 40 – 50 В.

5. Контрольні запитання

1. На базі якого пристрою будуються програмно–керовані джерела живлення.
2. Призначення програмно–керованого джерела живлення.
3. Принцип роботи ПКДЖ.
4. Чому в якості вихідної напруги ДОН використовують опорну напругу яка дорівнює $-10,24 \text{ В}$?
5. Які елементи схеми ПКДЖ задають максимальну напругу на виході?
6. Скільки значень аналогового сигналу може сформувати 10–розрядний ЦАП?
7. Що є базовим елементом ЦАП К572ПА1, що дозволяють перетворювати цифровий код в аналоговий струм?
8. Призначення програмно–керованого генератора сигналів довільної форми?
9. На базі якого пристрою будуються програмно–керовані генератори сигналів довільної форми?
10. По якій структурі можуть будуватися сучасні програмно–керовані генератори сигналів довільної форми?
11. Принцип роботи генератора сигналів довільної форми на базі двійкового лічильника.
12. Перевага й недолік генератора сигналів довільної форми на базі двійкового лічильника.
13. Принцип роботи генератора сигналів довільної форми на базі нагромаджуючого суматора.
14. Перевага й недолік генератора сигналів довільної форми на базі нагромаджуючого суматора.
15. Для чого на виході ЦАП включається ФНЧ?

Практичне заняття № 4

РОЗРОБКА КОМБІНАЦІЙНИХ ЛОГІЧНИХ СХЕМ ТА ТРИГЕРІВ НА МОВІ VHDL

Мета заняття

Навчитись розробляти комбінаційні логічні схеми та тригери на мові VHDL.

1. Короткі теоретичні відомості

Інтерфейс і архітектура пристрою на мові VHDL

Повний опис будь-якого цифрового пристрою мовою VHDL складається як мінімум із двох окремих частин: [6]

1. Опис інтерфейсу пристрою. Інтерфейс описується за допомогою оператора ENTITY, у якому визначаються входи й виходи пристрою, а також його параметри налаштування, які часто використовуються для налаштування стандартних пристроїв під конкретний проект.

Опис інтерфейсу мовою VHDL має вигляд:

```
Entity ім'я_об'єкту is  
[оголошення портів пристрою];  
[оголошення параметрів налаштування];  
End [ENTITY] ім'я_об'єкту;
```

Формат оголошення портів пристрою:

```
PORT (ім'я_порту : <режим> <тип>; [ім'я_порту : <режим> <тип>]);
```

Порти цифрового пристрою підтримують три режими роботи:

- IN – вхід;
- OUT – вихід;
- INOUT – двонаправлений порт.

Тип порту, характеризує значення сигналів, що можуть на нього поступати:

- integer – цілий;
- bit – бітовий;
- std_logic – сигнальний.

2. Опис архітектури пристрою. Архітектурне тіло пристрою описується за допомогою оператора ARCHITECTURE і представляє собою змістовний опис роботи цифрового пристрою, який обов'язково повинен бути підключений до відповідного інтерфейсу (оператора ENTITY). При цьому одному операторові ENTITY може відповідати декілька архітектурних тіл. Розрізняють два типи тіла архітектури (залежить від операторів, що використовуються):

- поведінкове архітектурне тіло – описує проект пристрою як сукупність дій, що їм виконуються;
- структурне архітектурне тіло – описує проект пристрою у вигляді сукупності готових компонентів та їх з'єднань між собою.

Формат опису архітектурного тіла мовою VHDL:

```
ARCHITECTURE ім'я_архітектури OF ім'я_Entity IS  
<розділ декларацій>  
BEGIN
```

<розділ операторів>

END [Architecture] ім'я_архітектури;

У розділі декларацій оголошуються константи, типи даних, сигнали й процедури, які будуть використовуватися при описі цифрового пристрою.

У розділі операторів описується архітектура цифрового пристрою за допомогою паралельних операторів мови VHDL.

Приклад розробки комбінаційних логічних схем мовою VHDL

На рис. 4.1 приведена принципова схема цифрового пристрою, а на рис. 4.2 принципова схема логічного блоку. Необхідно скласти опис проекту цифрового пристрою на мові опису апаратури VHDL, використовуючи поведінкове архітектурне тіло.

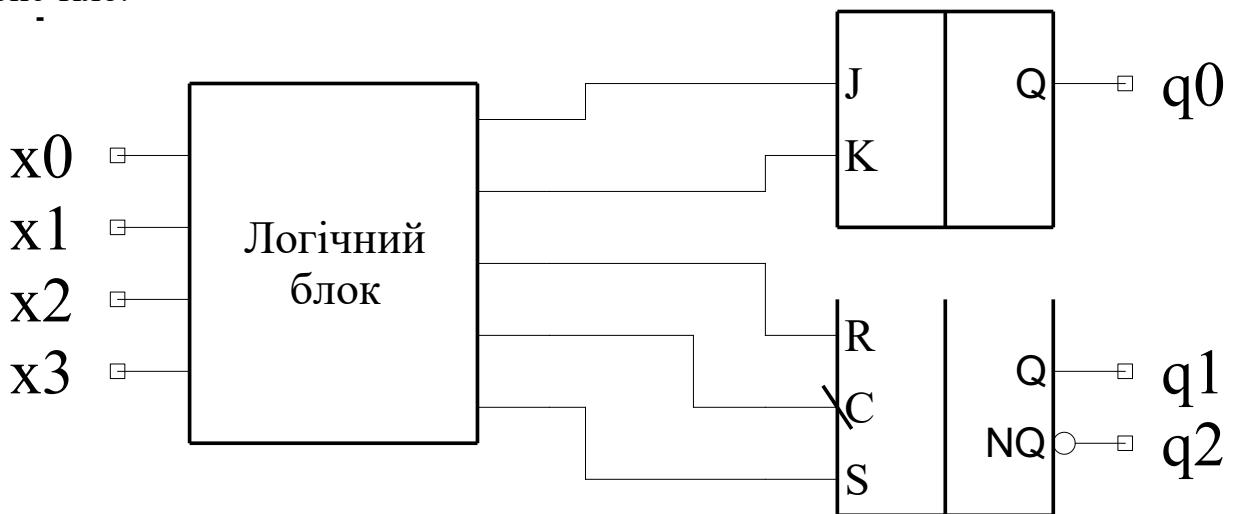


Рис. 4.1. Схема цифрового пристрою

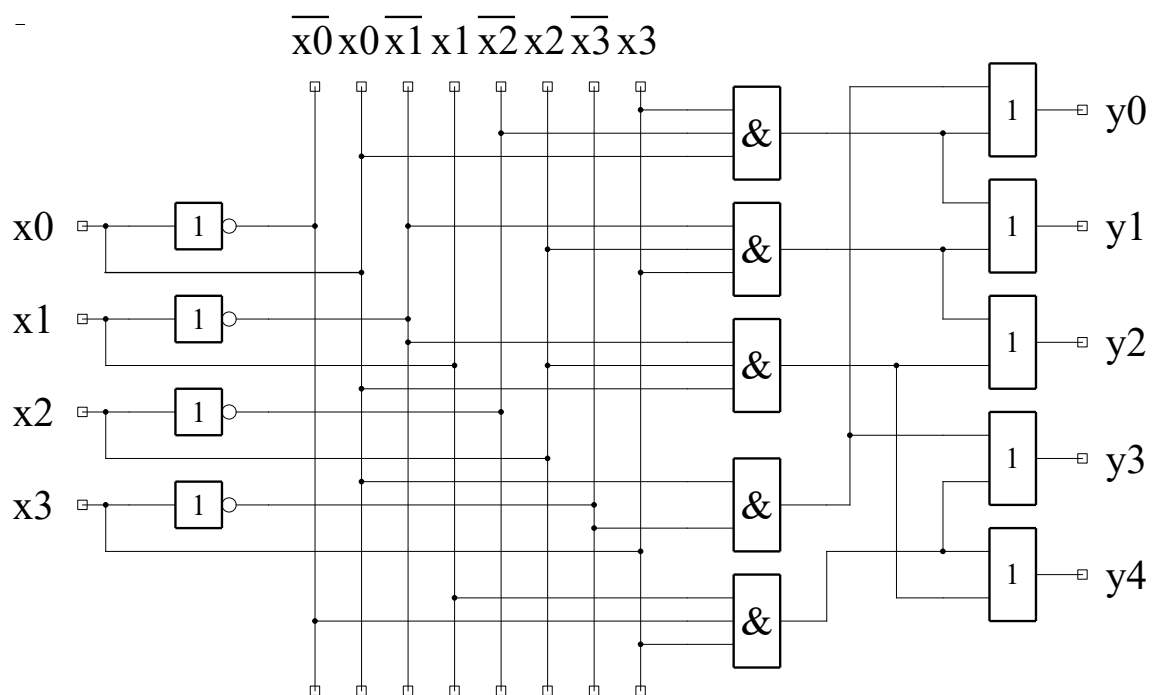


Рис. 4.2. Принципова схема логічного блоку

Опис цифрового пристрою на мові опису апаратури VHDL.

```
-- Опис проекту цифрового пристрою (варіант N)
LIBRARY IEEE;
USE ieee.std_logic_1164.all;
-- Опис інтерфейсу цифрового пристрою
ENTITY Digital_device IS
PORT (x0, x1, x2, x3: in std_logic; q0, q1, q2: out std_logic);
END Digital_device; — кінець опису інтерфейсу цифрового пристрою
-- Опис архітектури цифрового пристрою
ARCHITECTURE work_device OF digital_device IS
-- сигнали для з'єднання логічного блоку з тригерами
SIGNAL y0, y1, y2, y3, y4: std_logic;
-- резервування двох тригерів
SIGNAL qq0, qq1: std_logic;
BEGIN
-- Опис логічного блоку
Logic: Process (x0, x1, x2, x3)
BEGIN
y0 <= (x0 and not x3) or (x0 and not x2 and x3);
y1 <= (x0 and not x2 and x3) or (not x1 and x2 and x3);
y2 <= (not x1 and x2 and x3) or (x0 and not x1 and x2);
y3 <= (x0 and not x3) or (not x0 and x1 and x3);
y4 <= (not x0 and x1 and x3) or (x0 and not x1 and x2);
END Process Logic;
-- Опис асинхронного JK–тригера
JK: Process (y0, y1)
VARIABLE y: std_logic_vector (0 to 1);
BEGIN
y := y0 & y1;
CASE y IS
When "01" => qq0 <= '0';
When "10" => qq0 <= '1';
When "11" => qq0 <= NOT qq0;
When "00" => NULL;
End CASE;
END Process JK;
-- вивід значення JK–тригера на зовнішній вихід q0
q0 <= qq0;
-- Опис синхронного RS–тригера
RCS: Process (y3)
VARIABLE y: std_logic_vector (0 to 1);
BEGIN
y := y2 & y4;
IF (y3'EVENT AND y3 = '0') THEN
CASE y IS
```

```

When "10" => qq1 <= '0';
When "01" => qq1 <= '1';
When others => NULL;
END CASE;
END IF;
END Process RCS;
-- вивід значення RS–тригера на зовнішні виходи q1 та q2
q1 <= qq1; q2 <= NOT qq1;
END work_device; — кінець опису архітектури цифрового пристрою

```

2. Порядок виконання практичного заняття

По заданій принциповій схемі виконати описання цифрового пристрою на мові опису апаратури VHDL, використовуючи поведінкове архітектурне тіло. Схема логічного блоку розкрита нижче схеми цифрового пристрою у вигляді принципової схеми або табличної форми запису. Імена сигналів і виводів в проекті повинні співпадати з іменами, наведеними на схемі.

Варіанти завдань до практичного заняття № 4 приведені у додатковому файлі який можна скачати в системі дистанційної освіти «Лідер».

Варіант завдання видається викладачем.

3. Зміст звіту

1. Принципова схема заданого цифрового пристрою, а також принципова схема або таблична форма запису логічного блоку.
2. Опис розробленого цифрового пристрою на мові опису апаратури VHDL.

4. Контрольні запитання

1. З яких частин складається повний опис цифрового пристрою на мові VHDL.
2. Призначення оператора ENTITY.
3. Скільки операторів ENTITY може мати цифровий пристрій.
4. У яких режимах можуть працювати порти цифрового пристрою.
5. Призначення оператора Architecture.
6. Призначення паралельних операторів мови VHDL.
7. Які символи входять в алфавіт мови VHDL.
8. Який символ в мові VHDL є ознакою коментаря.
9. Яким символом у мові VHDL повинен закінчуватися оператор.
10. Які значення можуть приймати дані типу STD_LOGIC.
11. Які типи об'єктів можна використовувати в мові VHDL.
12. Який тип сигналу реалізує запис у дужках (C'event AND C='1').
13. Який тип сигналу реалізує запис у дужках (C'event AND C='0').
14. Які дії виконує операція конкатенації в мові VHDL.

Практичне заняття № 5

СИСТЕМА АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ QUARTUS PRIME

Мета заняття

Вивчити можливості системи автоматизованого проектування (САПР) QUARTUS Prime і засоби вводу проекту цифрових пристроїв. Навчитись проектувати цифрові системи на базі ПЛІС за допомогою САПР QUARTUS Prime.

1. Короткі теоретичні відомості

САПР Quartus Prime розроблена компанією Altera і представляє собою інтегроване середовище проектування цифрових пристроїв на базі ПЛІС компанії Altera. Система Quartus Prime має засоби зручного вводу проекту, швидкого тестування та безпосереднього програмування пристроїв. Система проектування Quartus Prime підтримує сімейства ПЛІС типу CPLD MAX II, MAX V і MAX 10, а також сімейства ПЛІС типу FPGA Cyclone, Arria і Stratix. Крім того, САПР Quartus Prime дозволяє використовувати у своїх проектах процесор NIOS II з програмним обчислювальним ядром. [7]

САПР Quartus Prime пропонує повний спектр можливостей для проектування цифрових систем та різноманітні засоби опису проектів з ієрархічною структурою.

Структура САПР QUARTUS Prime

САПР QUARTUS Prime включає в себе наступні редактори та додатки: [8]

– **Project Navigator** (Навігатор проекту) – відображає поточну ієрархічну структуру проекту (закладка Hierarchy), список файлів, що входять до проекту (закладка Files), а також структуру модулів проекту (закладка Design Units). За допомогою цього додатка можна додавати у проект нові файли або видаляти непотрібні; можна візуально визначити, чи є файл проекту схемним, текстовим або сигнальним; які допоміжні файли в проекті доступні користувачу для редагування; можна також безпосередньо відкрити або закрити один або декілька файлів та ввести призначення ресурсів для них.

– **Text Editor** (Текстовий редактор) – використовується для опису проекту на мові опису апаратури (HDL). САПР Quartus Prime підтримує проектування на мовах VHDL, Verilog, System Verilog та AHDL. Текстовий редактор дозволяє створювати та редагувати текстові файли проекту, крім того, в цьому редакторі можна створювати, переглядати та редагувати будь-які файли формату ASCII, що використовуються іншими додатками Quartus Prime. Створювати файли на мовах HDL можна в будь-якому текстовому редакторі, але текстовий редактор системи Quartus Prime має переваги у вигляді контекстної довідки, виділення кольором синтаксичних конструкцій і готових шаблонів для мов AHDL, VHDL і Verilog.

– **Block Editor** (Графічний редактор) – використовується для графічного способу опису проекту у вигляді структурної (Block Diagram) або електричної принципової схеми (Schematic). При розробці принципової схеми можна використовувати

вати розроблені компанією Altera примітиви, мегафункції та макрофункції, а також використовувати власні символи, що були розроблені раніше.

– **University Program VWF** (Waveform Editor – Сигнальний редактор) – призначений для моделювання розробленого цифрового пристрою за допомогою програми ModelSim. За допомогою даного редактора можна створювати тестові впливи для вхідних сигналів і після симуляції роботи цифрового пристрою побудувати часові діаграми вихідних сигналів.

– **Symbol Editor** (Редактор символів) – дозволяє створювати нові та редагувати існуючі символи модулів і блоків проекту. Редактор використовується для роботи із складними ієрархічними проектами, в яких кожному модулю може бути призначений графічний символ. Розроблені символи можна потім використовувати в графічному редакторі для проектування цифрової системи у вигляді структурної схеми.

– для розподілу ресурсів ПЛІС, яка вибрана для проекту цифрової системи, використовуються три редактори: **Chip Planner** – розміщення розроблених модулів цифрового пристрою по логічним блокам, **Pin Planner** – розподіл зовнішніх виводів ПЛІС між сигналами спроектованого пристрою, **Assignment Editor** – редагування параметрів ресурсів ПЛІС (зовнішніх виводів, логічних чарунок і блоків, часових параметрів ПЛІС та інше).

– **Compiler Tool** (Компілятор) – набір засобів для компіляції проекту, який включає у себе наступні інструменти:

– **Analysis & Synthesis** – перевірка синтаксису опису проекту, витяг списку з'єднань, логічний синтез проекту в схему на регістровому рівні (RTL опис), розподіл виводів ПЛІС;

– **Fitter** – трасування проекту включає в себе розведення та розміщення внутрішніх ресурсів ПЛІС з урахуванням обмежень на швидкість роботи та обсягу логічних ресурсів, що необхідні для реалізації проекту цифрового пристрою;

– **Assembler** – формування файлу програмування або завантаження для конфігурації ПЛІС – розширення «.prof» для ПЛІС типу CPLD або розширення «.sof» для ПЛІС типу FPGA;

– **Classic Timing Analysis** – створення часової моделі ПЛІС і аналіз проекту для оцінки часових затримок сигналів в розробленому цифровому пристрою;

– **EDA Netlist Writer** – отримання моделі з часовими параметрами ПЛІС для зовнішніх засобів моделювання (якщо вибраний такий пакет);

– **Compilation Report** – формування файлу з детальним звітом про всі етапи компіляції проекту. Виводить на екран повідомлення про помилки, попереджуючі та інформаційні повідомлення про стан проекту і дозволяє автоматично знайти джерело повідомлення в вихідному або допоміжному файлі (файлах) проекту.

– **ModelSim – Altera** (Симулятор) – набір інструментів для функціонального і часового моделювання (симуляції) роботи цифрового пристрою. Результати роботи симулятора зручно переглядати в сигнальному редакторі Waveform Editor.

– **Programmer** (Програматор) – призначений для безпосереднього завантаження конфігурації проекту в ПЛІС за допомогою програматора, наприклад USB–Blaster.

Під час інсталяції Quartus Prime в каталозі «quartus» створюється каталог «qdesigns», який містить приклади проектів для САПР Quartus Prime.

Етапи розробки проекту в САПР Quartus Prime

Проект цифрового пристрою на ПЛІС складається з великої кількості файлів, в тому числі допоміжних і вихідних файлів, які створюються автоматично самою САПР Quartus Prime, тому для кожного проекту слід створювати окремий каталог.

Алгоритм проектування цифрової системи в САПР Quartus Prime складається з наступних етапів: [8]

1. Створення проекту цифрового пристрою. При створенні проекту необхідно вибрати каталог для зберігання файлів проекту, задати ім'я проекту та ім'я файлу проекту верхнього рівня (top-level design), яке повинне збігатися з іменем оператора ENTITY розробленого пристрою, а також вибрати тип ПЛІС. Для зручності створення нового проекту САПР Quartus Prime має вбудований майстер «New Project Wizard».

2. Введення опису проекту в текстовому і (або) графічному вигляді з використанням різних редакторів САПР.

3. Синтез (компіляція) проекту. У випадку успішної компіляції можливе моделювання та часовий аналіз роботи розробленого цифрового пристрою. Якщо будуть виявлені помилки в проекті, то їх необхідно усунути й повторити компіляцію.

4. Моделювання (тестування) проекту. Для проведення моделювання потрібно спочатку створити тестові сигнали, що необхідно подавати на входи пристрою, користуючись сигнальним редактором. Після симуляції в сигнальному редакторі будуються часові діаграми сигналів на виходах цифрового пристрою. Для апаратного тестування ПЛІС можна використовувати логічний аналізатор «Signal tap II».

5. Конфігурація (програмування) ПЛІС – завантаження конфігурації в ПЛІС виконується за допомогою програматора USB-Blaster, а також будь-яких інших програматорів, які підтримують JTAG-інтерфейс.

2. Порядок виконання практичного заняття

1. Створити в каталозі «957M» папку «PZ5» для зберігання файлів проекту.

2. Запустити середовище проектування QUARTUS Prime 20.1 Lite Edition.

3. Створення нового проекту цифрового пристрою складається із шести кроків:

– зайти в меню: File / New Project Wizard і натиснути кнопку «Next» у вікні «Introduction».

1 крок – вибір каталогу для проекту, введення імені проекту та вибір Entity верхнього рівня (вікно Directory, Name, Top-Level Entity):

– вибрати робочу директорію для зберігання файлів проекту (в полі «What is the working directory for this project?»): натиснути кнопку «...» і в каталозі «957M» вибрати папку «PZ5»;

– ввести ім'я проекту (в полі «What is the name of this project?»): device_pz5;

– ввести ім'я файлу верхнього рівня проекту (в полі «What is the name of the top-level design entity for this project?»): device_pz5.

– натиснути кнопку «Next».

2 крок – вибрати тип проекту (вікно Project Type):

– вибрати пустий проект (Empty project) та натиснути кнопку «Next»;

3 крок – додати необхідні файли та бібліотеки до проекту (вікно Add Files):

– якщо файлів проекту ще не має, то цей крок необхідно пропустити та натиснути кнопку «Next»;

4 крок – вибір типу ПЛІС для свого проекту (вікно Family, Device & Board Setting):

– вибрати у вікні Family – MAX II, у вікні Devices – All, у вікні Available devices – EPM240T100C5;

– натиснути кнопку «Next»;

5 крок – вибір додаткових інструментів проектування (вікно EDA Tool Settings):

– у рядку «Simulation» в поле «Tool Name» було вибрано значення «ModelSim», а в полі «Format(s)» – значення «VHDL»;

– натиснути кнопку «Next»;

6 крок – підсумкова перевірка усіх параметрів проекту:

– на цьому кроці можна вернутися назад (кнопка «Back») і вибрати інші параметри проекту;

– натиснути кнопку «Finish».

4. Введення опису проекту:


– створити текстовий файл: зайти в меню: File / New, вибрати у розділі «Design Files» тип файлу VHDL File і натиснути кнопку «OK»;

– набрати текст опису пристрою на мові VHDL (приведений нижче);


– зберегти файл під іменем «device_pz5.vhd».

Для налаштування опцій текстового редактора (тип і розмір шрифту, кольорова схема, нумерація рядків та інше) необхідно викликати меню (Tools / Options) і у вікні «Category» вибрати розділ «Text Editor». Також на верхній панелі текстового редактора знаходяться швидкі кнопки, для виклику команд, що часто використовуються для роботи з текстовим описом проекту.

Для вставки шаблонів операторів або об'єктів мови VHDL необхідно викли-

кати меню (Edit / Insert Template) або натиснути швидку кнопку . У вікні «Language templates» відкрити розділ «VHDL». Наприклад, для вставки шаблонів операторів ENTITY і ARCHITECTURE необхідно відкрити розділ Constructs / Design Units, а шаблони операторів (IF, CASE та ін.) знаходяться в розділі Constructs / Sequential Statements. Після вибору необхідного шаблону необхідно натиснути кнопку «Insert».

5. Синтез (компіляція) проекту:

– запустити процес компіляції: натиснути швидку кнопку  або у вікні «Task» необхідно двічі клацнути лівою кнопкою миші на рядку «Compile Design» або викликати меню Processing / Start Compilation (CTRL+L).

Якщо при компіляції проекту будуть виявлені помилки, то їх необхідно усунути й повторити процес компіляції. Результати компіляції можна подивитися на закладці «Compilation Report» або у вікні повідомлень «Messages».

6. Розподіл виводів ПЛІС:

– викликати вікно управління виводами ПЛІС: натиснути швидку кнопку  або викликати меню Assignments / Pin Planner;

– підключити сигнали цифрового пристрою до заданих виводів ПЛІС:

– двічі натиснути ліву кнопку миші на виводі 2 (PIN_2), після чого повинно з'явитися вікно «Pin Properties». В цьому вікні вибрати в полі «Node name» сигнал «clock»;

– підключити до виводів ПЛІС: PIN_3 – сигнал «reset», PIN_4 – сигнал «count_output[0]», PIN_5 – сигнал «count_output[1]», PIN_6 – сигнал «count_output[2]», PIN_7 – сигнал «count_output[3]».

7. Вибрати режим роботи виводів ПЛІС, що не будуть використовуватись:

– викликати меню Assignments / Device;

– натиснути кнопку «Device and Pin Options»;

– вибрати у вікні «Категорія (Category)» рядок «Unused Pins»;

– вибрати у вікні «Reserve all unused pins:» режим виводів, що не використовуються «As input tri–stated» (вхід із Z–станом);

– натиснути кнопку «ОК»;

– запустить повторно компіляцію для того, щоб отримати фінальний проект цифрового пристрою.

Після вдалої компіляції можна приступати до тестування проекту.

8. Моделювання роботи цифрового пристрою:

– запустить редактор сигналів Waveform Editor: зайти в меню: File / New, вибрати в розділ «Verification/Debugging Files» тип файлу «University Program VWF» і натиснути кнопку «ОК»;

– на екрані з'явиться вікно «Simulation Waveform Editor».

9. Налаштувати редактор сигналів для тестування цифрового пристрою:

– натиснути праву кнопку миші в полі Name і вибрати меню «Insert Node or Bus», а потім у вікні, що з'явиться натиснути кнопку «Node Finder»;

– натиснути кнопку «List» – у вікні «Nodes Found:» з'явиться список усіх сигналів, що використовуються в розробленому пристрої. Вибрати сигнал «reset» і натиснути кнопку «>», сигнал з'явиться у вікні «Selected Nodes:». Таким же чином необхідно вибрати наступні сигнали: clock, count_output (Type – Output Group);

– натиснути кнопку «ОК»;

– в редакторі сигналів з'являться часові діаграми вибраних сигналів;

– зберегти налаштування у файлі «device_pz5.vwf» (меню File / Save As).


10. Налаштувати часові параметри симуляції роботи пристрою:


– встановити час симуляції пристрою 5 мкс: меню Edit / Set End Time, ввести в полі «End Time» – 5.0 us, натиснути кнопку «ОК»;

– встановити часовий крок симуляції 100 нс: меню Edit / Grid Size, ввести в полі «Period» – 100.0 ns, натиснути кнопку «ОК».

11. Задати початкове значення тестових сигналів на входах пристрою.

Значення сигналів зручно задавати за допомогою верхнього меню редактора сигналів. Для введення необхідних тестових сигналів необхідно виділити мишкою необхідний часовий інтервал (напроти відповідного сигналу) і ввести його значення в цьому інтервалі. Для одиночних сигналів можна ввести значення: лог.1, лог.0, Z–стан, інвертувати сигнал та інше:

– сигнал «reset»: виділити на часовій діаграмі сигналу «reset» інтервал від 0 до 1 us, і натиснути кнопку «Forcing High (1)» ( – лог.1). Інтервал від 1 до 5 мкс повинен мати значення лог.0;


– сигнал «clock»: виділити сигнал «clock» і натиснути кнопку «Overwrite Clock» ();

– встановити параметри сигналу «clock»: Period = 200.0 ns, Offset = 0, Duty cycle = 50%;

– натиснути кнопку «ОК»;

– зберегти налаштовані параметри у файлі «device_pz5.vwf».

12. Виконати симуляцію роботи пристрою:

– запустити симулятор через меню Simulation \ Run Functional Simulation або натиснути швидко кнопку ;

– якщо в процесі симуляції не буде помилок, то після закінчення моделювання з'явиться додаткове вікно редактора сигналів з результатами моделювання (з іменем device_pz5.sim.vwf). Це вікно призначене тільки для перегляду результатів моделювання роботи розробленого пристрою (його не можна редагувати й зберегти на диску).

Якщо в процесі моделювання симулятор видає помилку, то необхідно зробити наступні кроки:

– зайти в меню Simulation / Simulation Settings;

– відкриється вікно «Simulation Options»;

– натиснути кнопку «Restore Defaults»;

– у вікні «ModelSim Script» відредагувати рядок «vsim –novopt –c –t 1ps – ...».

В ньому треба видалити параметр –novopt;

– натиснути кнопку «Save».

Запустити повторно симулятор.

13. Замалювати часову діаграму роботи цифрового пристрою.

Текст опису пристрою на мові VHDL:

```
-- Проект 4–х розрядного лічильника на мові VHDL
```

```
-- Інкрементуючий лічильник з асинхронним скиданням
```

```
LIBRARY IEEE;
```

```

USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

ENTITY device_pz5 IS
PORT (clock: IN STD_LOGIC; reset: IN STD_LOGIC; count_output: OUT
STD_LOGIC_vector (3 downto 0));
END device_pz5;

ARCHITECTURE counter OF device_pz5 IS
Signal cnt: STD_LOGIC_vector (3 downto 0);
BEGIN
PROCESS (clock, reset)
BEGIN
IF (reset = '1') THEN cnt <= "0000";
ELSIF (clock'event AND clock = '1') THEN
cnt <= cnt + '1';
END IF;
END PROCESS;
count_output <= cnt;
END counter;

```

3. Зміст звіту

1. Структура системи проектування QUARTUS Prime.
2. Етапи розробки проекту цифрового пристрою в САПР QUARTUS Prime.
3. Текст опису цифрового пристрою мовою VHDL.
4. Часова діаграма роботи розробленого цифрового пристрою.

4. Контрольні запитання

1. Призначення САПР QUARTUS Prime?
2. Які типи ПЛІС підтримує САПР QUARTUS Prime.
3. Які мови опису апаратури підтримує САПР QUARTUS Prime.
4. Які способи вводу проекту цифрового пристрою підтримує САПР QUARTUS Prime.
5. Етапи розробки проекту цифрового пристрою в САПР QUARTUS Prime.
6. Призначення сигнального редактора (Waveform Editor) в САПР QUARTUS Prime?
7. Призначення редактора символів (Symbol Editor) в САПР QUARTUS Prime?
8. Які редактори в САПР QUARTUS Prime використовуються для розподілу ресурсів ПЛІС, що була вибрана для реалізації проекту цифрової системи.
9. Призначення набору інструментів компілятора (Compiler Tool) в САПР QUARTUS Prime?
10. Призначення програматора (Programmer) в САПР QUARTUS Prime?

Практичне заняття № 6

РОЗРОБКА ЧОТИРЬОХРОЗЯДНОГО СУМАТОРА НА МОВІ VHDL ТА МОДЕЛЮВАННЯ ЙОГО РОБОТИ В СИСТЕМІ QUARTUS PRIME

Мета заняття

Розробити на мові VHDL проєкт чотирьохрозрядного двійкового суматора з можливістю каскадного з'єднання, а також провести моделювання його роботи за допомогою САПР Quartus Prime.

1. Короткі теоретичні відомості

На рис. 6.1. представлено символічне зображення чотирьохрозрядного двійкового суматора з можливістю каскадного з'єднання. Двійкові числа поступають на входи *A* та *B* складаються і результат виводиться на вихід *S*. Для збільшення розрядності суматора їх можна з'єднувати каскадно. Для цього використовуються вхід *Carry* (Перенос) і вихід *Over* (Переповнення). Вхід *Carry* використовується для приймання сигналу переносу з молодшого каскаду, для цього він з'єднується з виходом *Over* попереднього каскаду. Для першого каскаду на цей вхід подається лог.0. Вихід *Over* використовується для передачі сигналу переповнення в старший каскад, для цього він з'єднується із входом *Carry* наступного каскаду.

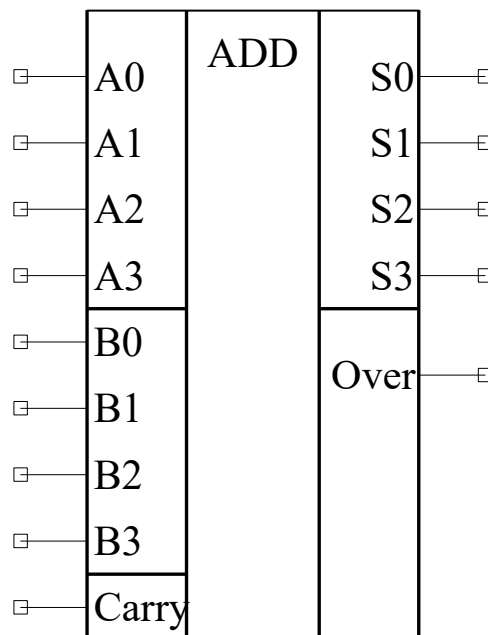


Рис. 6.1. Чотирьохрозрядний суматор з можливістю каскадного з'єднання

Опис 4-х розрядного двійкового суматора мовою VHDL:

-- підключення необхідних бібліотек

```
LIBRARY IEEE;
```

```
USE ieee.std_logic_1164.all;
```

```
USE ieee.std_logic_unsigned.ALL;
```

-- опис інтерфейсу суматора

```

ENTITY ADD4 IS
  PORT (A, B: IN std_logic_vector (3 downto 0); Carry: IN std_logic; S: OUT
std_logic_vector (3 downto 0); Over: OUT std_logic);
  END ADD4;

-- опис архітектури суматора
  ARCHITECTURE work_add OF add4 IS
  Signal A1, B1, S1: std_logic_vector (4 downto 0);
  BEGIN
  PROCESS (A, B, Carry)
  BEGIN
  A1<='0' & A; B1<='0' & B;
  S1<= A1 + B1 + Carry;
  S<= S1 (3 downto 0);
  Over <= S1 (4);
  END PROCESS;
  END work_add;

```

2. Порядок виконання практичного заняття

1. Створити в каталозі «957М» папку «PZ6» для зберігання файлів проекту 4–х розрядного двійкового суматора.

2. Запустити середовище проектування QUARTUS Prime 20.1 Lite Edition.


3. Створити проект двійкового суматора:

- зайти в меню: File / New Project Wizard;
- натиснути кнопку «Next»;
- вибрати каталог для зберігання файлів проекту: натиснути кнопку «…» і в каталозі «957М» вибрати папку «PZ6»;
- ввести ім'я проекту – ADD4;
- ввести ім'я файлу верхнього рівня проекту – ADD4;
- натиснути кнопку «Next»;
- вибрати пустий проект (Empty project);
- натиснути кнопку «Next»;
- додати необхідні файли або бібліотеки до проекту. Якщо файлів проекту ще не має, то цей крок необхідно пропустити та натиснути кнопку «Next»;
- вибрати тип ПЛІС для проекту: у вікні Family – MAX II, у вікні Devices – All, у вікні Available devices – EPM240T100C5;
- натиснути кнопку «Next»;
- вибрати додаткові системи проектування: у рядку «Simulation» в поле «Tool Name» було вибрано значення «ModelSim», а в полі «Format(s)» – значення «VHDL»;
- натиснути кнопку «Next»;
- натиснути кнопку «Finish».


4. Ввести опису проекту двійкового суматора:

- створити текстовий файл: зайти в меню: File / New, вибрати у розділі «Design Files» тип файлу VHDL File і натиснути кнопку «ОК»;
- набрати текст опису пристрою на мові VHDL;
- зберегти файл під іменем «ADD4.vhd».

5. Виконати компіляцію проекту:

- натиснути швидку кнопку  або у вікні «Task» необхідно двічі клацнути лівою кнопкою миші на рядку «Compile Design».

6. Розподіл виводів ПЛІС:

- викликати вікно управління виводами ПЛІС: натиснути швидку кнопку  або викликати меню Assignments / Pin Planner;
- підключити сигнали цифрового пристрою до заданих виводів ПЛІС:
- до виводів: 2–5 – сигнали «A», 6 – сигнал «Carry», 7, 8, 15, 16 – сигнали «B», 17–20 – сигнали «S», 21 – сигнал «Over».

7. Вибрати режим роботи виводів ПЛІС, що не будуть використовуватись:

- викликати меню Assignments / Device;
- натиснути кнопку «Device and Pin Options»;
- вибрати у вікні «Категорія (Category)» рядок «Unused Pins»;
- вибрати у вікні «Reserve all unused pins:» режим виводів, що не використовуються «As input tri–stated» (вивід із Z–станом);
- натиснути кнопку «ОК»;
- запустить повторно компіляцію для того, щоб отримати фінальний проект цифрового пристрою.

8. Провести тестування роботи двійкового суматора:

- запустить редактор сигналів Waveform Editor: зайти в меню: File / New, вибрати в розділі «Verification/Debugging Files» тип файлу «University Program VWF» і натиснути кнопку «ОК».

9. Налаштувати редактор сигналів для тестування двійкового суматора: [9]

- натиснути праву кнопку миші в полі Name і вибрати меню Insert / Insert Node or Bus, а потім у вікні, що з'явиться натиснути кнопку «Node Finder»;
- натиснути кнопку «List» – у вікні «Nodes Found:» з'явиться список усіх сигналів, що використовуються в розробленому пристрої.

Вибрати наступні сигнали: A, B, Carry, S, Over (вибрати сигнал і натиснути кнопку «>»), сигнал повинен з'явитися у вікні «Selected Nodes:»;


- натиснути кнопку «ОК»;
- в редакторі сигналів з'являться часові діаграми вибраних сигналів;
- зберегти налаштування у файлі «ADD4.vwf».

10. Налаштувати часові параметри симуляції роботи двійкового суматора:

- встановити час симуляції двійкового суматора 5 мкс: меню Edit / Set End Time, ввести в полі «End Time» – 5.0 us, натиснути кнопку «ОК»;

– встановити часовий крок симуляції 1 мкс: меню Edit / Grid Size, ввести в полі «Period» – 1.0 us, натиснути кнопку «ОК».


11. Задати початкове значення тестових сигналів на входах двійкового суматора.

Для введення необхідних тестових сигналів необхідно виділити мишкою необхідний часовий інтервал (напроти відповідного сигналу) і ввести його значення в цьому інтервалі. Для одиночних сигналів можна ввести значення: лог.1, лог.0, Z-стан, інвертувати сигнал та інше (допустимі значення вводяться з лівої вертикальної панелі інструментів). Для вводу значень групових сигналів використовується кнопка «Arbitrary Value» (

Сформувані тестові сигнали для моделювання роботи суматора. При цьому необхідно змоделювати наступні ситуації:

- при додаванні A і B відсутнє переповнення, при цьому перенос Carry = 0;
 - при додаванні A і B відсутнє переповнення, при цьому перенос Carry = 1;
 - при додаванні A і B настане переповнення, при цьому перенос Carry = 0;
 - при додаванні A і B настане переповнення, при цьому перенос Carry = 1.
- зберегти налаштовані параметри у файлі «ADD4.vwf».

12. Виконати симуляцію роботи пристрою:

– запустити симулятор через меню Simulation \ Run Functional Simulation або натиснути швидку кнопку ;

– повернутися у вікно редактора сигналів для перегляду результатів моделювання роботи розробленого двійкового суматора. Це вікно призначене тільки для перегляду результатів моделювання роботи розробленого пристрою (його не можна редагувати й зберегти на диску).

Замалювати часову діаграму роботи розробленого чотирьохрозрядного двійкового суматора.

3. Зміст звіту

1. Текст опису розробленого чотирьохрозрядного двійкового суматора на мові VHDL.

2. Часова діаграма тестування роботи розробленого чотирьохрозрядного двійкового суматора.

4. Контрольні запитання

1. Принцип роботи двійкового суматора.

2. Призначення виводів Carry (перенос) і Over (переповнення) в двійковому суматорі.

3. При яких умовах на виході Over (переповнення) в чотирьохрозрядному двійковому суматорі заявиться сигнал переповнення (лог.1)?

4. Які ситуації необхідно змоделювати, щоб перевірити всі можливі режими роботи двійкового суматора?

Практичне заняття № 7

РОЗРОБКА ДЕШИФРАТОРА НА МОВІ VHDL ТА МОДЕЛЮВАННЯ ЙОГО РОБОТИ В САПР QUARTUS PRIME

Мета заняття

Розробити на мові VHDL проект дешифратора, а також провести моделювання його роботи за допомогою САПР Quartus Prime.

1. Короткі теоретичні відомості

Одною з переваг розробки цифрових вузлів на мові опису апаратури VHDL, є можливість міняти або підлаштовувати під свій проект практично будь-які параметри. Наприклад, при розробці дешифратора на мові VHDL ми можемо встановити розрядність дешифратора, зробити будь-які виходи прямими або інверсними, додавати будь-яку кількість входів дозволу роботи дешифратора для їхнього каскадного з'єднання та інше.

Для прикладу на рис. 7.1. представлено умовне графічне позначення (УГП) дешифратора 3x8 із входом дозволу E (enable) і інверсними виходами. Вхід дозволу роботи дешифратора: E=1 – дозвіл роботи дешифратора, E=0 – заборона роботи дешифратора, тобто на всіх виходах сигнал буде мати значення лог.1.

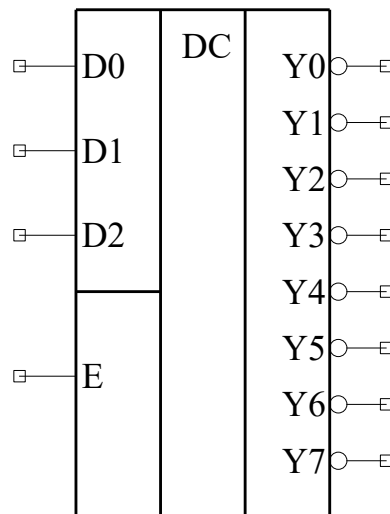


Рис. 7.1. Дешифратор 3x8 з входом дозволу та інверсними виходами

2. Порядок виконання практичного заняття

1. Використовуючи теоретичний матеріал лекцій розробити по заданому варіанту проект дешифратора на мові опису апаратури VHDL.
2. Створити в каталозі «957М» папку «PZ7» для зберігання файлів проекту дешифратора.
3. Запустити програму QUARTUS Prime 20.1 Lite Edition.
4. Створити проект заданого дешифратора.
5. Ввести опис проекту заданого дешифратора на мові VHDL.
6. Виконати компіляцію проекту дешифратора.

7. Розподілити виводи ПЛІС і вибрати режим роботи виводів ПЛІС, що не будуть використовуватись.

8. Запустити редактор сигналів Waveform Editor і налаштувати редактор сигналів для тестування дешифратора.

9. Налаштувати часові параметри симуляції роботи дешифратора:

– встановити час симуляції дешифратора 20 мкс: меню Edit / End Time, ввести в полі «Time» – 20.0 us, натиснути кнопку «ОК»;

– встановити часовий крок симуляції 1 мкс: меню Edit / Grid Size, ввести в полі «Period» – 1.0 us, натиснути кнопку «ОК».

10. Сформувані тестові сигнали для моделювання роботи дешифратора. При цьому необхідно змоделювати наступні ситуації:

– усі можливі комбінації на вході D при активному сигналі на вході дозволу;

– декілька вхідних комбінацій на вході D при пасивному сигналі на вході або входах дозволу.

11. Виконати моделювання роботи дешифратора.

12. Замалювати часову діаграму роботи розробленого дешифратора.

3. Варіанти завдань на розробку дешифратора

Таблиця 7.1 – Варіанти завдань на розробку дешифратора

№ варіанта	Тип дешифратора	Тип виходів	Кількість і тип входів дозволу
1	3x8	інверсні	немає входу
2	4x16	інверсні	немає входу
3	3x8	прямі	один інверсний
4	4x16	прямі	один інверсний
5	3x8	інверсні	один прямий
6	4x16	інверсні	один прямий
7	3x8	інверсні	один інверсний
8	4x16	інверсні	один інверсний
9	3x8	прямі	два входи (прямий і інверсний)
10	4x16	прямі	два входи (прямий і інверсний)
11	3x8	інверсні	два входи (прямий і інверсний)
12	4x16	інверсні	два входи (прямий і інверсний)
13	4x16	прямі	немає входу

4. Зміст звіту

1. УГП розробленого дешифратора.

2. Таблиця залежності вхідних та вихідних сигналів розробленого дешифратора.

3. Текст опису розробленого дешифратора на мові VHDL.

4. Часова діаграма тестування роботи розробленого дешифратора.

Таблиця 7.2 – Приклад таблиці залежності вхідних та вихідних сигналів для дешифратора 3x8 з одним входом дозволу (прямий)

Входи		Виходи
E	D	Y
0	xxx	00000000
1	000	00000001
1	001	00000010
1	010	00000100
1	011	00001000
1	100	00010000
1	101	00100000
1	110	01000000
1	111	10000000

Примітка. x – не має значення.

5. Контрольні запитання

1. Призначення та принцип роботи дешифратора.
2. Призначення керуючих входів E (enable) в дешифраторах.
3. Що означають цифри в типі дешифратора?
4. Якщо дешифратор має п'ять інформаційних входів, то скільки він повинен мати виходів?
5. Якщо дешифратор має 64 виходи, то скільки він повинен мати інформаційних входів?
6. Скільки вхідних тестових комбінацій необхідно сформулювати для повної перевірки дешифратора 8x256?

Практичне заняття № 8

РОЗРОБКА СТРУКТУРНОГО АРХІТЕКТУРНОГО ТІЛА НА МОВІ VHDL

Мета заняття

Навчитись розробляти складні цифрові пристрої на мові VHDL за допомогою структурного архітектурного тіла.

1. Короткі теоретичні відомості

Структурне архітектурне тіло дозволяє представити проект цифрового пристрою (ЦП) у вигляді набору компонентів та їх зв'язків. Структурне представлення проекту ЦП є найбільш ефективним засобом розробки складних систем, що мають ієрархічну структуру, тому що дозволяє спільно працювати над проектом групі фахівців, а також використовувати стандартні бібліотеки й модулі користувача, розроблені раніше.

Усі компоненти, що використовуються в проекті повинні мати свій оператор ENTITY і однозначно задане поведінкове архітектурне тіло. Опис компонентів приводиться в тому ж файлі, що і основний проект у вигляді структурного архітектурного тіла або заздалегідь протестовані поведінкові архітектурні тіла компонентів компілюють у бібліотеку, і потім просто підключають до свого проекту.

Структурне архітектурне тіло складається із трьох розділів: [6]

1. Розділ оголошення прототипів компонентів.

Формат оголошення прототипу компонента:

```
COMPONENT ім'я_Entity_компонента [IS]
```

```
[образ налаштувань];
```

```
[образ портів];
```

```
END COMPONENT [ім'я_Entity_компонента];
```

Образ налаштування й образ портів – це пряма копія розділів GENERIC і PORT з тексту оголошення оператора ENTITY відповідного компонента. Якщо в пристрої, що проектується буде використовуватися кілька однотипних компонентів, які використовують той самий оператор ENTITY, то прототип декларується один раз. Але кожному модулю проектованого пристрою, необхідно привласнити власне ім'я у вигляді мітки оператора входження компонента.

2. Розділ декларацій.

У цьому розділі оголошуються константи й сигнали, загальні для всього ЦП, що проектується. У даному розділі обов'язково треба оголосити всі сигнали (SIGNAL), за допомогою яких модулі пристрою з'єднуються між собою. Розрядність і тип сигналів повинен відповідати розрядності й типу виводів модулів, що з'єднуються. Якщо модулі, що з'єднуються, мають несумісні типи виводів, то між ними ставлять перетворювачі типів даних.

3. Розділ операторів.

При структурному описі цифрового пристрою, у цьому розділі використовують оператор входження компонентів, який має наступний формат:

```
мітка: [COMPONENT] ім'я_компонента
```

```
GENERIC MAP (список відповідностей параметрів налаштування);
```

PORT MAP (список відповідностей портів);

Список відповідностей має наступну структуру: (формальний параметр => фактичний параметр, формальний параметр => фактичний параметр).

У якості формальних параметрів виступають входи й виходи модулів пристрою, а також змінні й константи, а в якості фактичних параметрів зовнішні виводи пристрою й сигнали, які з'єднують виводи модулів, а також реальні значення змінних і констант. Якщо якийсь порт компонента не використовується, то у якості фактичного параметра використовується ключове слово OPEN.

Приклад розробки цифрового пристрою на мові VHDL за допомогою структурного архітектурного тіла

На рис. 8.1 приведена принципова схема цифрового пристрою. Необхідно скласти опис проекту цифрового пристрою на мові опису апаратури VHDL, використовуючи структурне архітектурне тіло.

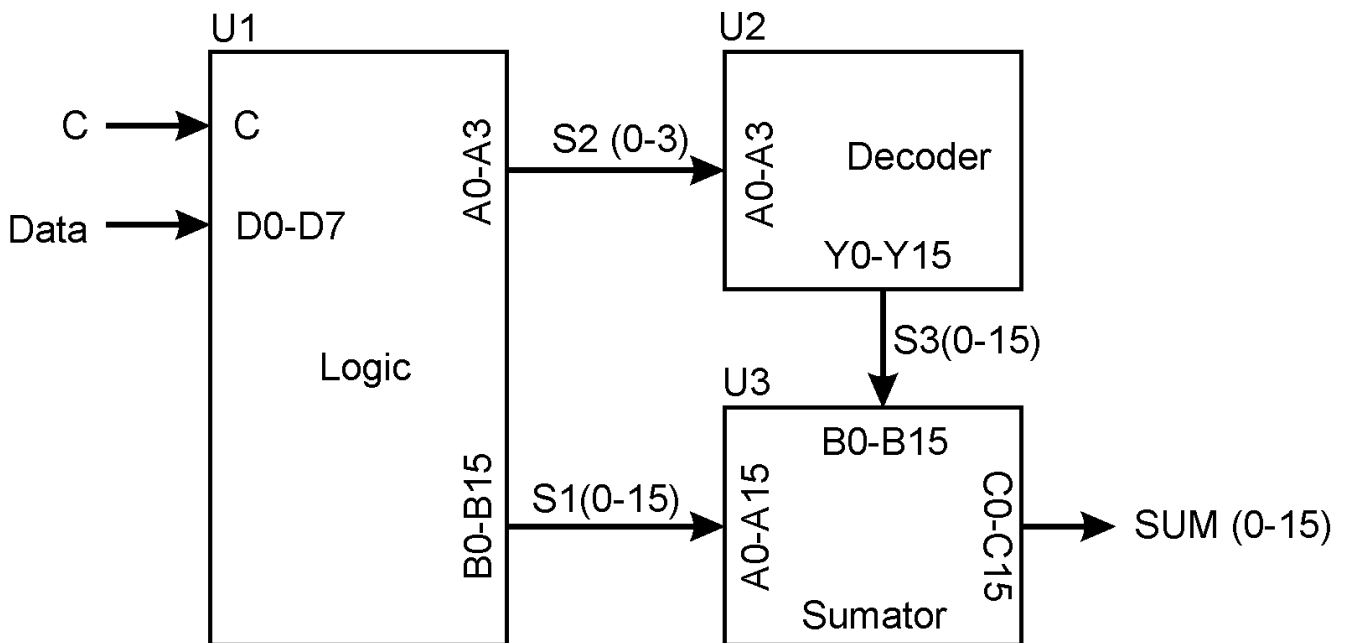


Рис.8.1. Структурна схема цифрового пристрою

Проект цифрового пристрою мовою опису апаратури VHDL:

```
LIBRARY IEEE;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.ALL;
```

-- опис логічного блока

```
ENTITY logic
```

```
PORT (D: IN std_logic_vector (7 downto 0); C: IN std_logic; A: OUT
std_logic_vector (3 downto 0); B: OUT std_logic_vector (15 downto 0));
```

```
END logic;
```

-- опис дешифратора 4x16

```
ENTITY decoder
```

```

    PORT (A: IN std_logic_vector (3 downto 0); Y: OUT std_logic_vector (15 downto
0));
    END ENTITY;
    ARCHITECTURE work_decoder OF decoder IS
    BEGIN
    PROCESS (A)
    VARIABLE I: integer RANGE 0 TO 7;
    BEGIN
    FOR I IN 0 TO 7 LOOP
    IF (CONV_INTEGER (A) = I) THEN Y(I)<='1';
    ELSE Y(I)<='0';
    END IF;
    END LOOP;
    END PROCESS;
    END work_decoder;

```

-- опис 16-розрядного суматора

```

    ENTITY adder
    PORT (A, B: IN std_logic_vector (15 downto 0); C: OUT std_logic_vector (15
downto 0); OVER: OUT std_logic);
    END adder;
    ARCHITECTURE work_add OF Adder IS
    Signal A1, B1, C1: std_logic_vector (15 downto 0);
    BEGIN
    PROCESS (A, B)
    BEGIN
    S1 <= A1 + B1;
    C <= C1;
    END PROCESS;
    END work_add;

```

-- опис структури цифрового пристрою

```

    ENTITY digital_device IS
    PORT (Data: IN std_logic_vector (7 downto 0); C: IN std_logic; Sum: OUT
std_logic_vector (15 downto 0));
    END digital_device;

```

```

    ARCHITECTURE work_device OF digital_device IS

```

-- оголошення прототипів компонентів

```

    COMPONENT logic
    PORT (D: IN std_logic_vector (7 downto 0); C: IN std_logic; A: OUT
std_logic_vector (3 downto 0); B: OUT std_logic_vector (15 downto 0));
    END COMPONENT;

```

```

    COMPONENT decoder

```

```

    PORT (A: IN std_logic_vector (3 downto 0); Y: OUT std_logic_vector (15 downto
0));
    END COMPONENT;

    COMPONENT adder
    PORT (A, B: IN std_logic_vector (15 downto 0); C: OUT std_logic_vector (15
downto 0); OVER: OUT std_logic);
    END COMPONENT;

-- розділ декларацій
    SIGNAL S1, S3: std_logic_vector (15 downto 0);
    SIGNAL S2: std_logic_vector (3 downto 0);

-- розділ операторів
    BEGIN
    U1: logic
    PORT MAP (C=>C, D=>Data, A=>S2, B=>S1);
    U2: decoder
    PORT MAP (A=>S2, Y=>S3);
    U3: adder
    PORT MAP (A=>S1, B=>S3, C=>SUM, OVER=>OPEN);
    END work_device;

```

2. Порядок виконання практичного заняття

По заданій структурній схемі розробити цифровий пристрій на мові опису апаратури VHDL:

- розробити окремо кожний модуль пристрою мовою VHDL, використовуючи поведінкове архітектурне тіло (крім логічного блоку);
- розробити цифровий пристрій, використовуючи структурне архітектурне тіло.

Імена входів і виходів модулів вибрати самостійно. Імена та розрядність (вказана у дужках після імені сигналу або зовнішнього виводу) модулів, їх зв'язків і зовнішніх виводів міняти не можна. При опису логічного блоку необхідно привести тільки його сутність (описати зовнішній інтерфейс логічного блоку).

Варіанти завдань до практичного заняття № 8 приведені у додатковому файлі який можна скачати в системі дистанційної освіти «Лідер».

Варіант завдання видається викладачем.

3. Зміст звіту

1. Структурна схема заданого цифрового пристрою.
2. Опис розробленого цифрового пристрою на мові опису апаратури VHDL.

4. Контрольні запитання

1. З яких розділів складається структурне архітектурне тіло?
2. Призначення оператора COMPONENT.
3. Скільки разів необхідно об'являти прототип компонента, якщо він буде використовуватися в проекті кілька раз?
4. У якому розділі архітектурного тіла необхідно описувати сигнали, які з'єднують модулі між собою?
5. При складанні списку відповідностей мапи портів (PORT MAP), які елементи відносяться до формальних параметрів?
6. При складанні списку відповідностей мапи портів (PORT MAP), які елементи відносяться до фактичних параметрів?
7. Що означає оператор OPEN, якщо він використовується в якості фактичного параметру в списку відповідностей мапи портів (PORT MAP)?

СПИСОК ЛІТЕРАТУРИ ТА ІНФОРМАЦІЙНІ РЕСУРСИ

1. <https://www.labcenter.com> – сайт компанії розробника системи автоматизованого проектування Proteus Design Suite.
2. Faulkenberry, Luces M. An introduction to operational amplifiers with linear IC applications [Text]: second edition / Luces M. Faulkenberry. – John Wiley & Sons: Texas State Technical Institute, 2015. – 572 p.
3. Smith, Steven W. Digital Signal Processing [Text]: A Practical Guide for Engineers and Scientists / Steven W. Smith. – Newnes, 2023. – 650 p. ISBN: 0-750674-44-X.
4. Marzalo, Carlos Fernández. Analog Filters Handbook using APL-2 [Text]: second edition / Carlos Fernández Marzalo. – Madrid, Spain: Springer, 2025. – 806 p. ISBN 978-3-031-80986-6.
5. Федорков, Б. Г. Мікросхеми ЦАП та АЦП [Текст]: функціонування, параметри, використання. Б. Г. Федорков, В. А. Телець. – К.: Енергоатоміздат, 2016. – 320 с.
6. IEEE Standard VHDL Language Reference Manual [Text]: IEEE Std 1076-2019. – New York, USA: IEEE, 2019. – 640 p. ISBN: 978-1-5044-6135-1.
7. <https://altera.com> – сайт компанії виробника ПЛІС Altera, містить документацію на ПЛІС, а також безплатну версію САПР Quartus Lite.
8. Intel Quartus Prime Standard Edition User Guides [Електронний ресурс] / Altera Corporation, UG-20173. – Altera Design Hub, 2019. – 1940 p. – Режим доступу: <https://www.altera.com/design/guidance/software/quartus-user-guides>.
9. Казимир, В.В. Проектування комп'ютерних систем на основі мікросхем програмованої логіки [Текст]: монографія / С.А. Іванець, Ю.О. Зубань, В.В. Казимир, В.В. Литвинов. – Суми: Сумський державний університет, 2017. – 313 с.

ПРАВИЛА ОФОРМЛЕННЯ ПРАКТИЧНИХ ЗАНЯТЬ

1. Формат листів:

- А4 (без рамок), орієнтування – книжне;
- поля: ліве і праве – по 2,25 см, верхнє – 1,8 см, нижнє – 3,0 см.

2. Шрифт тексту:

- тип – Times New Roman (TNR);
- розмір – 14;
- в набраному тексті не повинно бути шрифтів іншого типу.

3. Параметри абзацу для основного тексту:

- міжрядковий інтервал – 1,0 (одинарний);
- абзацний відступ на 0,77 см від початку рядка, однаковий по всьому основному тексту;

- вирівнювання абзаців – по ширині;

- абзаци основного тексту не відділяються один від одного.

4. Параметри абзацу для заголовків розділу:

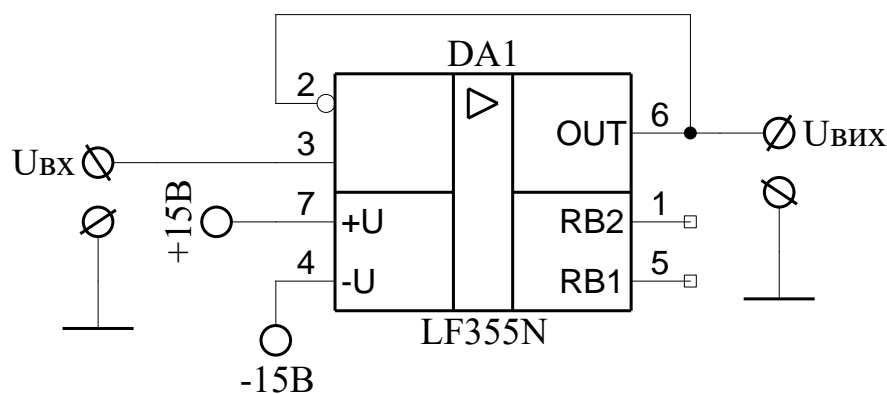
- шрифт TNR, розмір 14, напівжирний;

- вирівнювання абзаців – по центру;

- відступу першого рядка не має;

- від попереднього тексту підзаголовок відділяється інтервалом в 12 пунктів, а наступного – 6 пунктів.

5. Зразок оформлення рисунків:



$$U_{вх} = U_{вих}$$

Рис. 2. Схема буферного підсилювача

Параметри абзацу для рисунків та підписом під рисунком:

- вирівнювання абзаців – по центру;

- відступу першого рядка не має;

- від попереднього тексту рисунок та підпис відділяється інтервалом в 6 пунктів, а наступного – 12 пунктів.

- властивість рисунка «Обтекание текстом» – «В тексте»;

- нумерація рисунків наскрізна: рис. 1, рис. 2 и т.д.

6. Зразок оформлення таблиці:

Таблиця 1– Таблиця істинності дешифратора 74ALS138

Входи						Виходи							
E1	E2	E3	D2	D1	D0	0	1	2	3	4	5	6	7
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1

Параметри абзацу для номеру та для назви таблиці:

- вирівнювання абзацу – по центру;
- відступу першої строки не має;
- від попереднього тексту номер таблиці відділяється інтервалом в 6 пунктів;
- нумерація таблиць наскрізна: табл. 1, табл. 2 і т.д.

В таблиці для економії місця можна використовувати шрифт TNR, розмір шрифту 12. Після таблиці необхідно вставити один пустий рядок, щоб вона не зливалась з текстом основних абзаців.

Навчально–методичне видання

Профатилов Володимир Іванович

ЦИФРОВІ СИСТЕМИ АВТОМАТИКИ ТА РОБОТОТЕХНІКИ

Навчально–методичні рекомендації до проведення практичних занять

Електронне видання

Експертний висновок склав канд. техн. наук, доц. Костянтин Гончаров

Зареєстровано НМВ УДУНТ (№ 1.869 від 14.04.2026)

В авторській редакції
Комп'ютерна верстка Профатилов В.І.

Формат 60x84^{1/16}. Ум. друк. арк. 3,0. Обл.–вид. арк. 2,95
Зам. № 35

Видавець: Український державний університет науки і технологій
вул. Лазаряна, 2, м. Дніпро, 49010.

Свідоцтво суб'єкта видавничої справи ДК № 7709 від 14.12.2022

Адреса видавця та дільниці оперативної поліграфії:
вул. Лазаряна, 2, м. Дніпро, 49010