

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма: «Розробка програмного забезпечення»

Група: 4РП-07

# Дипломний проект

здобувача освіти денної форми навчання

РП.07.20.000.ДП

**ТАРАХТІЯ  
ВЛАДИСЛАВА  
ОЛЕКСАНДРОВИЧА**

м. Одеса  
2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»  
Освітньо-професійна програма: «Розробка програмного забезпечення»

Група: 4РП-07

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

до дипломного проекту на тему:

**Розробка програмного застосунку  
для розрахунку схем напівпровідникових випрямлячів**

Проектний матеріал складається з пояснювальної записки на 80 сторінках та графічного (презентаційного) матеріалу на 13 аркушах (слайдах)

Дипломник \_\_\_\_\_  (Тарахтій В.О.)

Керівник \_\_\_\_\_  (Скорняков В.С.)

**Консультанти:**

з економічного розділу \_\_\_\_\_  (Іванченков В.С.)

з розділу охорони праці та техніки безпеки \_\_\_\_\_  (Чорновол Н.І.)

з нормоконтролю \_\_\_\_\_  (Петрашова В.І.)

старший консультант \_\_\_\_\_  (Кривченко Ю.В.)

**До захисту допущений**

Голова циклової комісії \_\_\_\_\_  (Кривченко Ю.В.)

Завідувач відділення \_\_\_\_\_  (Скорнякова О.В.)

Захист « 17 » 06 2024 р. Протокол ЕК № 1

Оцінка ЕК 4(добре) / 80б.

Секретар ЕК \_\_\_\_\_ 

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Відділення комп'ютерних систем Комісія КТ та ПІ  
Спеціальність 121 «Інженерія програмного забезпечення»  
Освітньо-професійна програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР Беркань І.В.

« 15 » 01 2024 р.

**ЗАВДАННЯ**

**на дипломний проект**

Здобувачеві освіти Тарахтію Владиславу Олександровичу  
(прізвище, ім'я, по батькові)

1. Тема проекту Розробка програмного застосунку для розрахунку схем напівпровідникових випрямлячів

затверджена наказом по коледжу від «02» листопада 2023 р. № 244-А2-ОД

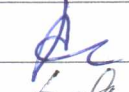
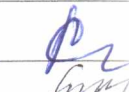

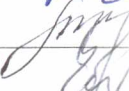
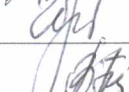

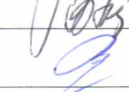



2. Термін здачі закінченого проекту 10.06.2024 р.

3. Вихідні дані до проекту Схеми випрямлячів з ємкісним та індуктивним навантаженням; Перелік ключових параметрів випрямлячів напруги різних типів; Програма має розраховувати числові значення параметрів елементів випрямлячів; Передбачити відповідні попередження про некоректні початкові дані та точність обчислення; Програма повинна мати візуальний інтерфейс з можливістю необхідних налаштувань за допомогою списків вибору та перемикачів

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)  
Аналіз методів розрахунку параметрів напівпровідникових випрямлячів. Аналітичний огляд програмного забезпечення для математичного моделювання електронних схем. Короткий аналіз і вибір програмних засобів для розробки. Реалізація алгоритмів розрахунку напівпровідникових випрямлячів з ємкісною та індуктивною реакцією навантаження. Реалізація візуального інтерфейсу та тестування розробленого застосунку

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)  
Структурна схема випрямного пристрою; Схеми випрямлячів, що працюють на ємкість та індуктивність; Блок-схеми алгоритмів розрахунку випрямлячів з ємкісною та індуктивною реакцією навантаження; Блок-схема алгоритму розрахунку для випрямляча, що живиться напругою прямокутної форми; Середовище розробки Embarcadero RAD Studio C++Builder; Візуальний інтерфейс застосунку для розрахунку схем напівпровідникових випрямлячів з ємкісним та індуктивним навантаженням; Порівняння результатів ручного та програмного розрахунків

6. Консультанти по проекту, із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання при
Основний розділ	Скорняков В.С.		
Економічний розділ	Іванченков В.С.		
Розділ охорони праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Кривченко Ю.В.		

7. Дата видачі завдання \_\_\_\_\_

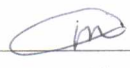
Керівник

Скорняков В.С.

  
(підпис)

Завдання прийняв до виконання

Тарахтій В.О.

  
(підпис)

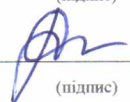
КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту	Термін виконання етапів дипломного проекту (роботи)	Відм. вик.
1	Вступ. Постановка мети та задач проектування	28.04.2024р.	Вик.
2	Дослідження роботи блоків живлення та їх параметрів	1.05.2024р.	Вик.
3	Загальний опис реалізації моделі	3.05.2024р.	Вик.
4	Розрахунок випрямних пристроїв з ємкісним навантаж.	7.05.2024р.	Вик.
5	Розрахунок випрямних пристроїв з індуктивним навант.	10.05.2024р.	Вик.
6	Вибір і опис інструментарію для реалізації моделі	13.05.2024р.	Вик.
7	Огляд програмних засобів розрахунку електронних схем	16.05.2024р.	Вик.
8	Розробка візуального інтерфейсу для застосунку	18.05.2024р.	Вик.
9	Розробка алгоритмів розрахунку випрямлячів	20.05.2024р.	Вик.
10	Реалізація алгоритмів розрахунку мовою C++	23.05.2024р.	Вик.
11	Випробування програмного застосунку	27.05.2024р.	Вик.
12	Аналіз результатів, підготовка слайдів презентації	1.06.2024р.	Вик.
13	Економічні розрахунки та питання з охорони праці	3.06.2024р.	Вик.
14	Підготовка графічної частини проекту	6.06.2024р.	Вик.
15	Підготовка проекту до захисту та тестування ПЗ	9.06.2024р.	Вик.

Дипломник

  
(підпис)

Керівник

  
(підпис)



# ЗМІСТ

Вступ.....	7
1 Основний розділ .....	8
1.1 Аналіз методів розрахунку параметрів напівпровідникових випрямлячів.....	8
1.1.1 Аналітичний огляд схем напівпровідникових випрямлячів.....	8
1.1.2 Розрахунок напівпровідникових випрямлячів з ємкісною реакцією навантаження.....	14
1.1.3 Розрахунок напівпровідникових випрямлячів з індуктивною реакцією навантаження.....	17
1.1.4 Розрахунок напівпровідникових випрямлячів, що живляться напругою прямокутної форми.....	19
1.2 Аналітичний огляд програмного забезпечення для математичного моделювання електронних схем.....	22
1.3 Вибір операційної системи для створюваного програмного забезпечення.....	24
1.4 Короткий аналіз і вибір програмних засобів для розробки.....	24
1.5 Аналіз можливостей інтегрованого середовища розробки Embarcadero RAD Studio C++Builder .....	33
1.6 Реалізація алгоритму розрахунку напівпровідникових випрямлячів з ємкісною реакцією навантаження .....	35
1.7 Реалізація алгоритму розрахунку напівпровідникових випрямлячів з індуктивною реакцією навантаження.....	43
1.8 Реалізація алгоритму розрахунку напівпровідникових випрямлячів, що живляться напругою прямокутної форми.....	46
1.9 Реалізація візуального інтерфейсу та тестування розробленого застосунку.....	51
1.9.1 Тестування у режимі двохнапівперіодної однофазної мостової схеми з ємкісним навантаженням.....	52
1.9.2 Тестування у режимі однонапівперіодної трифазної схеми з	

індуктивним навантаженням.....	53
2 Економічний розділ .....	56
2.1 Резюме.....	56
2.2 Визначення трудомісткості розробки програмного забезпечення.....	56
2.3 Розрахунок ціни програмного продукту.....	59
3 Розділ охорони праці та техніки безпеки.....	61
3.1 Вступ.....	61
3.2 Аналіз небезпечних та шкідливих чинників, що впливають на працівника.....	61
3.3 Розробка заходів з охорони праці.....	62
3.3.1 Виробничі приміщення.....	62
3.3.2 Мікроклімат робочої зони працівників, вентиляція.....	62
3.3.3 Освітлення робочого місця, шум, вібрація.....	63
3.3.4 Організація робочого місця користувача ПК.....	63
3.3.5 Електробезпека.....	64
3.4 Пожежна безпека.....	65
Висновки.....	66
Перелік використаних інформаційних джерел .....	67
ДОДАТОК А. Код функцій Rectifier_GetInputValues, btnInductiveClick, btnCapacityClick, btnRFClick мовою С++ .....	68
ДОДАТОК Б. Слайди мультимедійної презентації .....	74

## ВСТУП

Напівпровідникові випрямлячі в складі блоків живлення (джерел вторинного електроживлення) є невід'ємною частиною сучасних електронних приладів, зокрема комп'ютерної техніки, та у значній мірі визначають їх техніко-економічні показники.

Враховуючи, що проблема швидкої розроблювання засобів сучасної електроніки посилюється, як правило, малим часовим інтервалом, що відводиться інженерів-розробників на проектування схеми, треба застосовувати сучасні комп'ютеризовані методики обчислення, аналізу та оптимізації. Метою застосування цих методів є підвищення продуктивності розроблювання та якості розроблюваної апаратури.

Необхідність автоматизації обчислення блоків живлення, тобто створення алгоритмічного і програмного забезпечення, обумовлена досить складним та тривалим процесом ручного обчислення. При створенні відповідного програмного забезпечення із'явиться можливість істотно скоротити даний процес. Окрім цього, виключаються можливі помилки, котрі можуть виникнути при ручному обчислення. Електронні схеми півпровідникових випрямників характеризуються наявністю компонентів із нелінійними характеристиками. Аналітичний обчислення таких ланцюгів представляє значну трудність навіть при використанні програмних засобів. Проте, існують методи наближеного обчислення схем півпровідникових випрямників, що дозволяють із достатньою задля інженерної практики точністю визначити основні параметри проектованої схеми.

Мета дипломного проекту полягає в реалізації методики обчислення схем півпровідникових випрямників, що входять до складу блоків живлення сучасних електронних приладів, зокрема комп'ютерної техніки. На основі цієї методики треба розробити алгоритми обчислення та виконати їх програмну реалізацію мовою C++. Розроблюваний програмний застосунок дозволить розраховувати параметри елементів, що входять до складу блоків живлення поза початковими даними.

					РП 07. 20 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

# 1 ОСНОВНИЙ РОЗДІЛ

## 1.1 Аналіз методів обчислення значень півпровідникових випрямників

### 1.1.1 Аналітичний огляд схем півпровідникових випрямників

Напівпровідниковий випрямляч (рис. 1.1) зазвичай містить силовий трансформатор, електричні вентиля та згладжуючий фільтр, вихід якого із'єднується із навантаженням (споживачем енергії постійного струма).



Рисунок 1.1. Структурна схема діодного приладу

Трансформатор служить задля отримання необхідної величини напруження, що забезпечує необхідне значення випрямленої напруження при заданій напрузі мережі та задля електричної ізоляції між вхідним та вихідним ланцюгами випрямника.

Основним елементом діодного приладу є електричний вентиль – нелінійний прилад, супротив якого задля прямого напрямку струма у сотні-тисячі разів менше, ніж задля зворотного. Саме ця властивість та використовується в випрямлячах задля перетворення змінного струма у постійний. У якості вентилів зазвичай використовують напівпровідникові діоди.

Поза поміччю фільтра зменшуються змінні складові випрямленого струма, обумовлені наявністю нелінійного елемента (вентилів). Іншими словами, фільтр призначений задля згладжування імпульсів напруження на навантаженні.

Як виняток, в складі діодного приладу спроможне існувати відсутнім згладжуючий фільтр чи трансформатор. Випрямлячі без згладжуючого фільтра застосовуються порівняно рідко. Проте їх застосовують задля живлення комутаційної апаратури (реле, контактори) та у тих випадках, коли пульсація напруження на навантаженні не може істотного значення. Багатофазні схеми

випрямників (наприклад, схема Ларіонова) часто дозволяють обійтися без згладжуючого фільтра.

Відсутність трансформатору на вході діодного приладу передбачає можливість живлення випрямника безпосередньо з мережі без зміни живлячої (мережевої) напруги та без перетворення числа фаз випрямленого струму. Таких випадків спроможне опинитися багато, проте вони не є характерними задля більшості приладів електроживлення.

Задля випрямників принципове значення може характер навантажування, включеного на виході, тобто схема згладжуючого фільтра. В загальному випадку індуктивний характер навантажування приводить до збільшення, а ємкісний – до зменшення часу протікання струму через вентиль. У результаті міняються розрахункові співвідношення струмів та напруги, виникають фазові зсуви між струмами та напругами, значно ускладнюються перехідні процеси.

Випрямлячі, що працюють на ємність, володіють нижчим ККД та великим внутрішнім опором у порівнянні із випрямлячами, що працюють на індуктивність, проте дозволяють отримати добре згладжування випрямленої напруги при малих розмірах фільтра. Випрямлячі із ємкісним навантаженням використовуються у широкому діапазоні випрямленої напруги (з одиниць вольт до десятків кіловольт). Щоб уникнути вживання конденсаторів дуже великої ємності, випрямлені струми не повинні перевищувати декількох сотень міліампер.

Випрямлячі із індуктивним характером навантажування застосовуються при великих струмах навантажування (понад 1 А) у широкому діапазоні випрямленої напруги. Окрім цього, ці випрямлячі доцільно застосовувати у тих випадках, коли потрібно обмежити величину імпульсу струму через вентиль. Якщо змінна напруга джерела стабілізована по середньому значенню, як це буває у джерелах напруги прямокутної форми із інтервалом між імпульсами, то задля збереження стабільності випрямленої напруги треба застосовувати випрямлячі, що працюють тільки на фільтр із індуктивністю на вході. Недоліками такого типу випрямників є перенапруги, що виникають на

					РП 07. 20 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

вихідній ємкості фільтра при включенні випрямника та при змінах опору навантажування. Це спроможне привести до пошкоджень у ланцюгах навантажування та особливо небезпечно задля півпровідникових схем.

Проектування випрямника зводиться до вибору схеми та типу вентилів, обчислення режиму вентилів, ефективних значень струмів та напруження обмоток трансформатору та визначенню значень згладжуючого фільтра.

Як правило, всі елементи діодного приладу повинні розраховуватися комплексно, оскільки кожен подальший елемент істотним чином впливає на режим роботи попереднього та подальшого блоку (елементу) схеми. Наприклад, згладжуючий фільтр різко міняє режим роботи та розрахункові співвідношення струмів та напруження у ланцюзі вентилів та трансформаторі.

Проте, в багатьох випадках важливо розраховувати кожен блок приладу окремо, обумовлюючи та враховуючи, якою мірою позначається взаємний вплив режимів роботи решти блоків схеми, використовуючи дані обчислення одного блоку у якості початкових даних обчислення задля подальшого та попереднього блоку. Наприклад, у результаті обчислення випрямника виходять початкові дані задля обчислення трансформатору та згладжуючого фільтра; при обчислення згладжуючого фільтра можливо отримати необхідні дані задля обчислення випрямника та стабілізатора напруження.

Початковими даними задля обчислення є необхідні значення вихідного струма та напруження при заданих допустимих значеннях коефіцієнта пульсації. При цьому разом з цим враховуються умови живлення випрямника, задані додаткові експлуатаційні вимоги та властивості розраховуваного діодного приладу. У процесі обчислення окремих блоків доводиться зіставляти отримувані дані із тими, котрі відомі чи отримані при обчислення блоків, передуючих у схемі розраховуваним блокам. Методика обчислення того чи іншого діодного приладу залежить з режимів роботи цього приладу та його типу.

Випрямлячі та у загальному випадку нестабілізовані випрямні пристрої можливо класифікувати по наступних ознаках:

- типові використовуваних вентилів;

					РП 07. 20 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

- характеру управління випрямленою напругою;
- характеру навантаження;
- схемі випрямлення.

Окрім приведених класифікаційних ознак можуть існувати та додаткові. Прийнято розрізняти потужні та малопотужні випрямлячі, низькочастотні (промислової частоти 50 Гц) та підвищеної частоти (400 та 1000 Гц), імпульсні та ін.

На рисунку 1.2 та 1.3 приведені основні схеми випрямників, вживані задля живлення радіоелектронної апаратури.

Однонапівперіодну схему (рисунок 1.2 а) зазвичай застосовують у малопотужних випрямлячах та у тих випадках, коли не вимагається високого ступеня згладжування випрямленої напруги. Ця схема застосовується тільки при роботі на ємність.

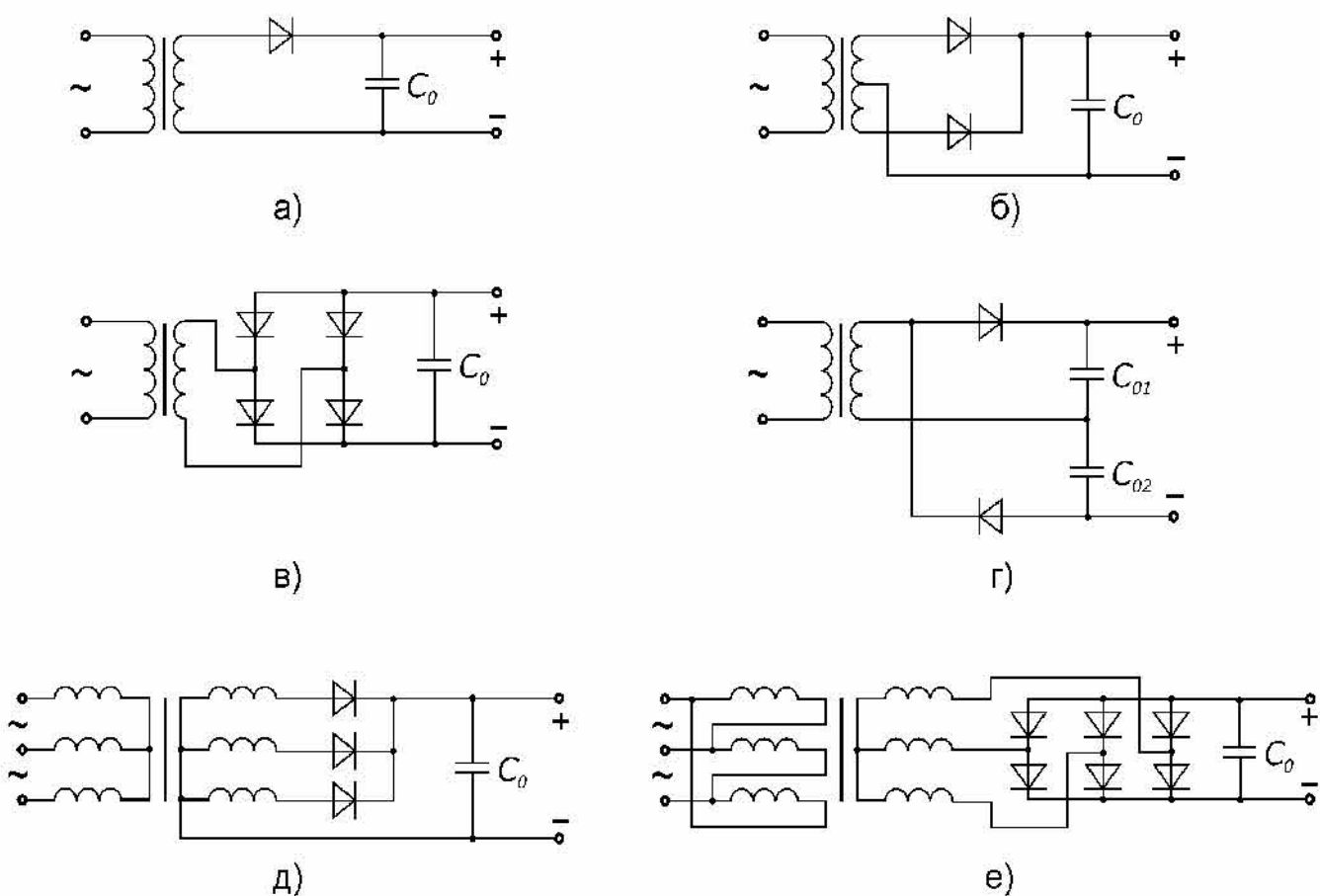


Рисунок 1.2. Схеми випрямників, що працюють на ємність

Двохнапівперіодна схема (рисунки 1.2 б та 1.3 а) застосовується у пристроях невеликої потужності при випрямленій напрузі до 500-600 У. Періодичність імпульсів випрямленої напруги у 2 рази більш частоти мережі живлення, що надає зменшити згладжуючий фільтр. У порівнянні із однофазною мостовою схемою ця схема надає у два рази зменшити число вентилів (при низькій напрузі).

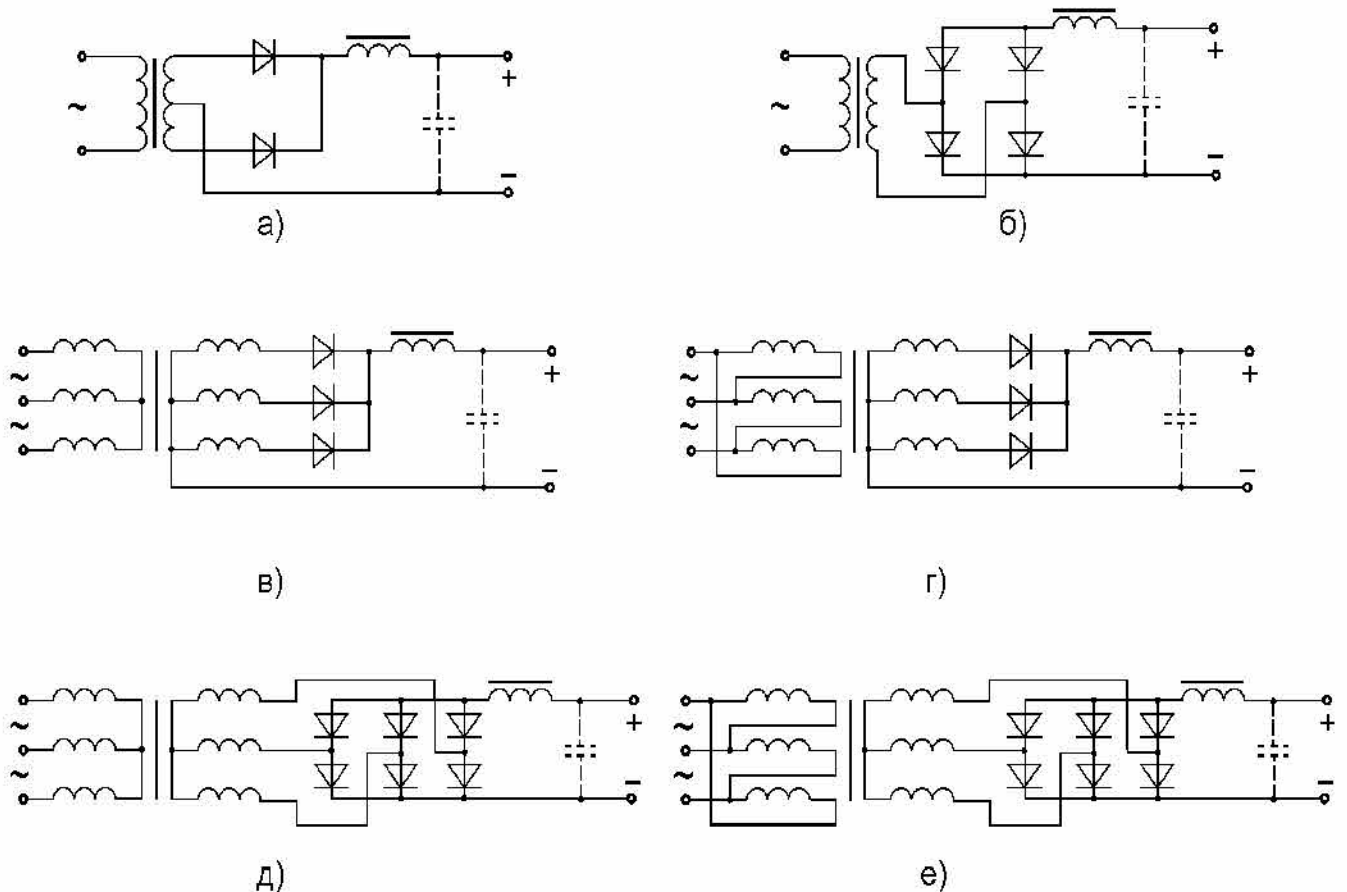


Рисунок 1.3. Схеми випрямників, що працюють на індуктивність

В однофазній мостовій схемі, яку інколи називають двохнапівперіодною мостовою (рисунки 1.2 у та 1.3 б), обернена напруга на вентилі у два рази менше, ніж у одно- та двохнапівперіодній схемах, а періодичність пульсації така ж, як у двохнапівперіодній схемі. Мостова схема характеризується добрим використанням потужності трансформатора.

В схемі подвоєння напруги (рисунок 1.2 з), яку часто називають схемою Латура, обернена напруга на вентилі у два рази менше, ніж у одно- та двохнапівперіодній схемах. Схема застосовується при випрямленій напрузі

вище 700-800 У. Періодичність пульсації випрямленої напруги у два рази більш частоти мережі.

Трифазна схема (рисунки 1.2 *д* та 1.3 *у, з*) використовується задля випрямлення не дуже високої напруги у тих випадках, коли по котрих-небудь причинах не можливо чи незручно застосовувати більш трьох вентилів. Через мале падіння напруги на вентилях трифазна схема доцільна при дуже низькій випрямленій напрузі. Періодичність пульсації у три рази більш частоти живлячої мережі. Схема характеризується поганим використанням потужності трансформатора та порівняно великою зворотною напругою на вентилях.

В шостифазній схемі Ларіонова (рисунки 1.2 *е* та 1.3 *д, е*) обернена напруга на вентилі майже у 2 рази менше, ніж у трифазній схемі, а періодичність імпульсів у шість разів більш частоти живлячої мережі. Амплітуда першої гармоніки пульсації поза відсутності перекосу фаз не перевищує 6% з випрямленої напруги. Схема володіє високими енергетичними параметрами у порівнянні із іншими схемами випрямників. В схемі Ларіонова не рекомендується сполучати вторинну обмотку трансформатора у трикутник через можливе перевантаження обмоток зрівняльними струмами.

Часто виникає необхідність перетворення низької постійної напруги у змінну із подальшим її підвищенням чи пониженням та випрямленням. Задля перетворення постійної напруги у змінну вигідно застосовувати перетворювачі на транзисторах, що працюють у ключовому режимі та відповідно володіють високим ККД. Такі перетворювачі виробляють змінну напругу в вигляді імпульсів прямокутної форми, амплітуда котрих прямо пропорційна постійній напрузі джерела живлення.

Задля стабілізації змінної напруги, що виробляється транзисторними перетворювачами, застосовують схеми із регулюванням тривалості прямокутного імпульсу, що надає зберегти високий ККД перетворювача. Стабілізацію змінної напруги можливо здійснити, підтримуючи незмінними середнє (випрямлене) чи ефективне (діюче) значення напруги.

					РП 07. 20 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

### 1.1.2 Обчислення півпровідникових випрямників із ємнісною відповіддю навантажування

Задля обчислення випрямників із ємнісною відповіддю навантажування використовують метод, заснований на наближених графоаналітичних обчисленнях. Задля обчислення випрямника треба мати наступні дані:

- спрямлена напруга  $E_0$ ;
- випрямлений current  $I_0$ ;
- рівень імпульсів  $a_{\text{п0}}$ ;
- напруга живлячої мережі  $U_c$ ;
- частоту напруги  $f$ .

Методика обчислення задля різнотипних типових схем однакова. Відзнаки при обчислення схем полягають тільки у різнотипних коефіцієнтах  $K_i$  ( $i = \overline{1, 10}$ ) в формулах. Задля кожної схеми ці значення містять певне значення та представлені у таблиці 1.1.

Таблиця 1.1. Значення задля обчислення схем півпровідникових випрямників із ємнісною відповіддю навантажування

Тип випрямника	Значення									
	$K_1$	$K_2$	$K_3$	$K_4$	$K_5$	$K_6$	$K_7$	$K_8$	$K_9$	$K_{10}$
1п1ф	1	2,3	1	1	1	1	2	1	2,828	1,84
1п2ф	0,5	4,7	2	1	0,5	1	1,8	1	2,828	1,63
2п1ф	0,5	3,5	2	1	0,707	0,707	1,5	2	1,414	1,63
2п1фу	1	0,9	1	0,5	1,414	0,707	1,5	1	2,828	2,95
1п3ф	0,333	6,9	3	1	0,333	1	2,15	1	2,828	0,72
2п3ф ★	0,333	4,5	6	0,578	0,333	0,707	1,25	2	2,44	0,92
2п3ф Δ	0,333	13,5	6	1	0,193	1,23	1,25	2	1,414	0,53

Після вибору схеми переходять до оцінки вентилів. Середній спрямлений current вентиля

$$I_{\text{ов}} = K_1 I_0. \quad (1.1)$$

Задля зворотної напруги та імпульсного струма вентиля є наближені формули, проте після обчислення трансформатору можливо отримати їх точні значення. Обчислення трансформатору починається із визначення індукції  $B_T$ , котра знаходиться по таблиці на основі приблизних розрахунків потужності

трансформатору. Проте задля програмної реалізації використовуватимемо апроксимуючу формулу

$$B_T = 1,2 - 0,4 \sin(0,003 E_0 I_0). \quad (1.2)$$

Після цього обчислюємо супротив фази вторичної котушки

$$r_T = K_2 \cdot \frac{E_0}{I_0 f B_T} \cdot 4 \sqrt{\frac{f B_T}{E_0 I_0}}. \quad (1.3)$$

Активний супротив фази випрямника рівний

$$r = K_8 R_i + r_T, \quad (1.4)$$

де  $R_i \approx \frac{0,2}{I_{0B}}$  – внутрішній супротив вентилів.

Визначаємо допоміжний параметр

$$A_0 = \operatorname{tg} \Theta - \Theta = \frac{\pi r I_0}{K_3 E_0}. \quad (1.5)$$

Вирішуючи трансцендентне рівняння  $\operatorname{tg} \Theta - \Theta = A_0$ , знаходимо кут відсічення струма  $\Theta$ . При ручному обчислення він визначається по графіку. При програмній реалізації задля знаходження  $\Theta$  використовуватимемо чисельний метод ділення відрізка навпіл.

ЕДС вторичної котушки трансформатору рівна

$$U_{2x} = \frac{K_4 E_0}{\sqrt{2} \cos \Theta}. \quad (1.6)$$

Визначаємо допоміжний рівень

$$D_0 = \frac{\sqrt{\pi [\Theta (1 + 0,5 \cos 2\Theta) - 0,75 \sin 2\Theta]}}{\sin \Theta - \Theta \cos \Theta}, \quad (1.7)$$

який при ручному обчислення можливо визначити по графіку. Знаючи рівень  $D_0$ , знаходимо ефективний current вторичної котушки

$$I_2 = K_5 D_0 I_0. \quad (1.8)$$

Ефективний current первинної котушки

$$I_1 = \frac{K_{10} I_0 U_{2x}}{U_c}. \quad (1.9)$$

Габаритна потужність трансформатору приблизно

$$P_{\text{габ}} \approx K_7 P_0 = K_7 E_0 I_0. \quad (1.10)$$

Визначаємо параметри діодів. Обернена напруження вентиля рівна

$$U_{обр} = K_9 U_{2x}. \quad (1.11)$$

Ефективний current вентиля

$$I_B = K_6 I_2. \quad (1.12)$$

Знаходимо допоміжний рівень  $F_0$

$$F_0 = \frac{\pi(1 - \cos \Theta)}{\sin \Theta - \Theta \cos \Theta}, \quad (1.13)$$

та розраховуємо імпульсний current вентиля

$$I_m = \frac{F_0 I_0}{K_3}. \quad (1.14)$$

Мощність, що виділяється на одному вентилі при протіканні струма у прямому напрямку

$$P_B \approx \frac{I_B^2 \cdot 0,2}{I_{0B}}. \quad (1.15)$$

Необхідне значення ємкості конденсатора фільтра обчислюється поза формулою

$$C_o = \frac{H_\phi}{r \cdot a_{n0} \cdot f}, \quad (1.16)$$

де  $H_\phi$  – допоміжний рівень, залежний з кута  $\Theta$  та з числа фаз випрямлення (рівень  $K_3$ ). Він знаходиться по формулі

$$H_\phi = 25330 \cdot (2\Theta - \sin 2\Theta) \cdot \cos \Theta, \quad (1.17)$$

при  $K_3 = 1$ , чи

$$H_\phi = 101000 \cdot \frac{[\sin(K_3 \Theta) \cdot \cos \Theta - K_3 \cdot \cos(K_3 \Theta) \cdot \sin \Theta]}{K_3 \cdot (K_3^2 - 1) \cdot \cos \Theta}, \quad (1.18)$$

при  $K_3 > 1$ . При ручному обчислення рівень  $H_\phi$  визначається із графіка.

Задля обчислення внутрішнього опору випрямника треба визначити проміжну точку характеристики навантажування:

$$E = \frac{U_{2x} \sqrt{2} \cos \Theta}{K_4}, \quad (1.19)$$

$$I = 0,45 \cdot \frac{K_3 U_{2x} \cdot \left( \sin \frac{\Theta}{2} - \frac{\Theta}{2} \cdot \cos \frac{\Theta}{2} \right)}{K_4 \cdot r}. \quad (1.20)$$



Тоді спрямлена напруга при холостому ході рівна

$$E_{0X} = E_0 + \Delta E_r + \Delta E_x + \Delta E_{в\ cх} + \Delta E_{др}, \quad (1.23)$$

де падіння напруги на активному опорі трансформатора  $\Delta E_r = K_2 I_0 r_T$ , на реактивному –  $\Delta E_x = K_3 I_0 f L_S$ , на вентилях у схемі –  $\Delta E_{в\ cх} = K_4 \cdot \Delta E_v \approx K_4 \cdot 0,6$ ; падіння напруги на дроселі  $\Delta E_{др} \approx 0,005 \cdot E_0$ .

ЕДС вторичної котушки трансформатора рівна

$$U_{2x} = K_6 E_{0X}. \quad (1.24)$$

Ефективний current вторичної котушки

$$I_2 = K_7 I_0. \quad (1.25)$$

Ефективний current первинної котушки

$$I_1 = K_{14} I_0 n, \quad (1.26)$$

де  $n = U_{2x}/U_1$  – рівень трансформації.

Габаритна потужність трансформатора рівна

$$P_{габ} = K_9 E_{0X} I_0. \quad (1.27)$$

Визначаємо параметри діодів. Обернена напруга вентиля рівна

$$U_{обр} = K_5 U_{2x}. \quad (1.28)$$

Середній спрямлений current вентиля

$$I_{0в} = K_1 I_0. \quad (1.29)$$

Імпульсний спрямлений current  $I_M = I_0$ . Розсіювана діодом потужність

$$P_v = \Delta E_v I_{0в} \approx 0,6 I_{0в}. \quad (1.30)$$

Знаходимо мінімальну індуктивність дроселя

$$L_{др\ min} = \frac{2E_0}{(K_3^2 - 1) K_3 \pi f I_0}. \quad (1.31)$$

Тоді ємність фільтра

$$C = \frac{q_{п} \cdot 10^6}{K_3^2 \cdot 4\pi^2 f^2 L_{др\ min}}, \quad (1.32)$$

де  $q_{п} = a_{п0}/a_{п1} = K_{10}/a_{п1}$ ,  $a_{п0}$  – рівень імпульсів на вході фільтра (постійний задля даної схеми). Робоча напруга конденсаторів може існувати

розрахована на холостий хід випрямника, тобто  $U_{раб} \geq E_{0x} C > 0 = K_{11} U_{2x}$ , де  $E_{0x} C > 0$  – спрямлена напруження холостого ходу випрямника при  $C > 0$ .

Внутрішній супротив випрямника рівний

$$R_e = \frac{E_{0x} - E_0}{I_0}. \quad (1.33)$$

Критична точка характеристики навантажування визначається по формулах

$$I_{0кр} = \frac{E_0}{(K_3^2 - 1) K_3 \pi f L_{др мин}}, \quad (1.34)$$

$$U_{0кр} = E_0 + (I_0 - I_{0кр}) \cdot R_e. \quad (1.35)$$

В підрозділі 1.7 розглянуто алгоритм обчислення та його програмна реалізація, створені на основі даної методики.

#### **1.1.4 Обчислення півпровідникових випрямників, що живляться напруженням прямокутної форми**

Початковими даними задля обчислення є:

- спрямлена напруження  $E_0$ ;
- випрямлений current  $I_0$ ;
- рівень пульсації  $a_n$ ;
- напруження живлячої мережі  $U_1$ ;
- періодичність напруження  $f$ ;
- шпаруватість  $\beta$ .

Нижче приводиться узагальнена методика обчислення у разі індуктивної та ємкісної реакції навантажування. Поза невеликим винятком, відмінність при обчислення задля цих типів реакції навантажування полягає у різнотипних коефіцієнтах  $K_i$  ( $i = \overline{1,10}$ ) в формулах. Деякі із цих коефіцієнтів залежать з значення шпаруватості  $\beta$ , відповідно бажано їх обчислити до початку основних розрахунків. Значення та формули задля обчислення коефіцієнтів представлені у таблиці 1.3. Решта відзнак між розрахунками задля ємкісної та індуктивної реакцій навантажування буде вказана в міру викладу методики. Спочатку визначаємо супротив трансформатору:

$$r_T = K_1 \cdot \sqrt{\frac{1,2E_0}{f I_0^3}} \quad (1.36)$$

Після цього знаходимо ЕДС вторичної котушки

$$U_{2x} = K_2 E_0 + K_3 r_T I_0 + K_4 \quad (1.37)$$

та габаритну потужність трансформатору

$$P_{\text{габ}} = 1,1 \cdot U_{2x} I_0 K_5 \quad (1.38)$$

Таблиця 1.3. Значення задля обчислення схем півпровідникових випрямників, що живляться напруженням прямокутної форми

	Схема					
	Із індуктивною відповіддю навантажування		Із ємнісною відповіддю навантажування			
	1n2φ із ср.точ.	2n1φ мостова	Одно-напів-періодна	Із серед.точ.	Мостова	Подвоєння
K <sub>1</sub>	5,5	4,08	1,9	3,9	2,9	0,75
K <sub>2</sub>	1,01	1,01	1	1	1	0,5
K <sub>3</sub>	1	1	2β	β	2β	2
K <sub>4</sub>	1	2	√2β	√2β	2√2β	√ $\frac{1}{\beta}$
K <sub>5</sub>	$\frac{1+\sqrt{\beta+1}}{2}$	1	$\frac{1+\sqrt{\beta+1}}{2}$	1,18	1	2
K <sub>6</sub>	$\frac{1}{2}$	1	2	$\frac{1}{2}$	1	14
K <sub>7</sub>	$\frac{\beta}{\sqrt{2}}$	β	$\sqrt{\frac{2\beta-1}{\beta}}$	0,846	1	1
K <sub>8</sub>	β	$\frac{\beta}{2}$	1	1	1	1
K <sub>9</sub>	1	1	2	1	1	2
K <sub>10</sub>	$\frac{1}{2}$	$\frac{1}{2}$	√2β	$\sqrt{\frac{\beta}{2}}$	$\sqrt{\frac{\beta}{2}}$	0,7 · √2β

Ефективний current вторичної котушки

$$I_2 = I_0 \sqrt{K_6 \beta} \quad (1.39)$$

Ефективний current первинної котушки

$$I_1 = 1,1 \cdot K_7 I_0 n, \quad (1.40)$$

де  $n = U_{2x}/U_1$  – рівень трансформації.

Визначаємо параметри діодів. Обернена напруга діоду дорівнює

$$U_{обр} = 2 K_8 U_{2x}. \quad (1.41)$$

Середній спрямлений current вентиля при обчислення діодного приладу із індуктивною відповіддю навантажування обчислюється поза формулою

$$I_{0B} = \frac{I_2}{2}, \quad (1.42)$$

а у разі ємкісної реакції навантажування:

$$I_{0B} = \frac{I_0 K_9}{2}. \quad (1.43)$$

Імпульсний спрямлений current вентиля

$$I_M = I_0 \beta K_9. \quad (1.44)$$

Розсіювана діодом потужність

$$P_B = K_{10} I_0. \quad (1.45)$$

Обчислюємо ємність конденсатору фільтра. При індуктивній реакції навантажування вона рівна

$$C = \frac{10^7 \cdot I_0}{4\pi^3 a_{\pi} E_0 f}. \quad (1.46)$$

У разі ємкісної реакції навантажування, ємність конденсатору обчислюється поза формулою

$$C = \frac{K_9 I_0 \cdot K_9 \beta \cdot \sin\left(\frac{\pi}{K_9 \beta}\right) \cdot 10^5}{2 a_{\pi} E_0 f}. \quad (1.47)$$

При обчислення випрямника із індуктивною відповіддю навантажування треба ще розрахувати індуктивність дроселя та внутрішній супротив випрямника. Індуктивність дроселя

$$L_{др} = 0,1 \cdot \frac{E_0 \beta \sin \frac{\pi}{\beta}}{I_0 f}. \quad (1.48)$$

Внутрішній супротив випрямника

$$R_e = \frac{U_{2x} - E_0}{I_0} \cdot \frac{2}{K_4}. \quad (1.49)$$

## 1.2 Аналітичний огляд програмного забезпечення задля математичного моделювання електронних схем

У даний час рівень математичного моделювання електронних схем виключно високий. Упевнено моделюються складні фізичні процеси у найрізноманітніших схемах, число типів котрих дійсно невичерпно. Проте тільки останніми роками системи машинного моделювання стали доступними масам інженерів, науковців та студентів.

Зокрема, інтегровані системи моделювання схемотехніки Micro-CAP вигідно відрізняються з інших таких систем задля ПК (наприклад, PSPICE) своїми сервісами. Найважчий етап проектування (завдання схеми та її топологічний та математичний опис) у них реалізований простим та наочним графічним діалогом. Він нагадує збірку схем поза поміччю конструктора, що містить кубики із компонентами електронних схем, із котрих радіоаматор збирає потрібну схему. Не вимагається знання жодних вхідних мов задля завдання схем та управління системою у ході аналізу схем та їх моделювання. Дані аналізу виходять як у числовій (табличній формі), так та в вигляді графіків, що нагадують осцилограми, отримувані при дослідженні схеми поза поміччю електронного осцилографа, характеріографа чи вимірювача частотних характеристик.

Після опису схеми система автоматично формує складні системи нелінійних та диференціальних рівнянь, що описують роботу схем, вирішує їх та виводить дані обчислення у зручному задля оператора вигляді. Як вона це робить – секрет розробника. Тільки по окремих нюансах досвідчений фахівець спроможне розпізнати деякі секрети системи. Наприклад, що вона реалізована на мові QuickBasic (на це вказують пускові модулі цієї мови фірми Microsoft Inc.), що при аналізі у тимчасовій області використовується змінний крок у часі (йде адаптація до швидкості модельованих процесів) та так далі. Отже, головною відмітною особливістю системи автоматизованого аналізу електронних приладів (схем) Micro-CAP є завдання електронної схеми, що підлягає обчислення, у графічному вигляді із контролем її зображення на екрані дисплея.

					РП 07. 20 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

Система моделювання PSpice призначена задля аналізу аналогових електронних схем різного функціонального призначення: підсилювачів, активних фільтрів, компараторів, генераторів та відповідно подібне. До складу досліджуваних схем можуть входити компоненти наступних видів:

- пасивні компоненти R,L,C-компоненти;
- незалежні та залежні з часу джерела напруження та струма;
- напівпровідникові діоди;
- транзистори;
- лінійні багатообмотувальні трансформатори;
- залежні джерела струма, керовані струмом чи напруженням;
- залежні джерела напруження, керовані струмом чи напруженням;
- функціональні вузли.

Система PSpice надає проводити наступні види аналізу:

- визначення статичного режиму нелінійних схем та обчислення передавальної процедури по постійному струму;
- визначення чутливості зміни статичного режиму та напруження на переходах транзисторів до варіації значень компонентів схеми;
- визначення частотних характеристик схеми;
- визначення внутрішніх шумів схеми у частотній області із обчисленням рівня теплового, дробу та фліккер-шуму на виході схеми і шуму, приведенного до входу;
- визначення вкладу кожного джерела шуму у загальний рівень шуму схеми;
- визначення часових характеристик лінійних та нелінійних схем;
- Фур'є-визначення у перехідному режимі;
- температурний визначення.

### **1.3 Вибір операційної системи задля створюваного програмного забезпечення**

Операційна система забезпечує управління всіма апаратними компонентами та підтримку роботи програм оператора, надаючи йому можливість загального управління машиною.

При виборі середовища створення програми треба враховувати декілька чинників, а саме:

- складність та трудомісткість процесу проектування програмного забезпечення задля конкретного середовища;
- наявність інструментальних засобів розроблювання програмного забезпечення;
- можливість внесення коректив у програму у процесі експлуатації;
- наявність засобів проектування призначеного задля оператора інтерфейсу;
- швидкість створення програми;
- надійність роботи програми та захищеність з програмних збоїв.

Можливо виділити ряд переваг задля розроблювання програмного забезпечення під управлінням ОС Windows:

- Графічний, призначений задля оператора інтерфейс;
- Незалежність з зовнішніх приладів;
- Доступність всього об'єму оперативної пам'яті;
- Підтримка масштабованих шрифтів.

Основна перевага при створенні програмного застосунку задля обчислення схем півпровідникових випрямників саме під ОС Windows 10 – зручний графічний інтерфейс оператора і широко розповсюджені засоби розроблювання.

### **1.4 Короткий визначення та вибір програмних засобів розроблювання**

У даному підрозділі будуть розглянуті засоби, застосування котрих спроможне існувати доцільним в розробці програмного застосунку задля обчислення схем півпровідникових випрямників, і із котрих будуть обрані найкращі інструменти. У першу чергу розглянемо доступний набір

високорівневих мов програмування, на котрих буде написане ядро даного проекту. Задля поліпшення процесу розроблювання мови програмування повинні відповідати наступним вимогам:

1. Зрозумілий і чіткий синтаксис;
2. Статична типізація;
3. Поглиблений інструментарій задля кращого доступу до пам'яті;
4. Наявність вичерпної документації;
5. Підтримка поза потреби інтеграції сторонніх засобів.

У даному випадку мова, котра найбільше підходить задля написання даного програмного продукту буде відібрана на основі відповідності вищезазначеним критеріям і судженню автора про доцільність її застосування, відповідно будуть проаналізовані п'ять наступних мов програмування, обрані поза індексом ТЮВЕ, це: Java, C, C++, C# і Python.

Мова Java – це мова програмування, початково розроблювана компанією Sun Microsystems. Основною особливістю Java-програм є перетворення написаного користувачем тексту в спеціальний байт-код, завдяки чому такі програми можуть існувати запуснені в будь якому середовищі, що підтримує машину Java. Станом на 2023 рік Java є однією із найпопулярніших мов програмування. Це в свою чергу є гарантією того, що задля рішення існує вичерпна документація. Перевагою створення програми на віртуальній машині є повна незалежність виконуваного коду з програмного та апаратного середовищ, що надає запускати програми на Java будь-де, де існує працююча віртуальна машина. Разом з цим одною із особливостей Java є стійка система безпеки, котра заснована на відповідно, що створення програми на Java контролює сама віртуальна машина. Відповідно дії, що перевищують встановлені права програми, можуть викликати негайне переривання. Програми Java переводяться у байт-код, який виконується відповідною Java-машиною, котра у свою чергу обробляє байт-код та надсилає інструкції до апаратної частини як інтерпретатор. Одним із основних недоліків даної віртуальної машини є суттєве зниження

					<b>РП 07. 20 000. 00 ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

продуктивності. Однак ряд деяких поліпшень дещо збільшив швидкість створення програм Java. Серед них можливо виділити наступні, такі як:

1. Застосування технології перекладу байт-коду в машинний код безпосередньо під час створення програми (JIT-технологія) із можливістю збереження версій класу у машинному кодї;

2. Застосування специфічного задля платформи коду в стандартних бібліотеках;

3. Апаратне забезпечення, що значно прискорює оброблення байт-коду (наприклад, технологію Jazelle, підтримувану деякими процесорами архітектури ARM).

Згідно із інформацією, наданою сайтом wikipedia, задля тестового набору задач, час створення програми Java у середньому становить в 1.5-2 рази більш, ніж задля C/C++. Варто зазначити, що іноді Java працює швидше, а іноді в сім разів повільніше. Разом з цим одним із недоліків можливо зазначити той факт, що споживання фізичної пам'яті Java у 10–30 разів перевищувало аналогічне задля програми, написаної на C/C++. Разом з цим спроможне заслуговувати на увагу дослідження, що представила компанія Google, яке разом з цим вказує на те, що може місце значно нижча продуктивність та більш високе застосування пам'яті задля програм Java порівняно із аналогічними програмами C++.

У загалом, ідеї, втілені у концепції на різномісних реалізаціях середовища віртуальних машин Java, надихнули багатьох ентузіастів розширити список мов, котрі можливо застосовувати задля створення програм, що працюють поза помічною віртуальних машин. Ці ідеї разом з цим відображені у специфікації спільної мовної інфраструктури CLI, котра є основою платформи із загальною назвою Microsoft .NET. Мова C – мова програмування загального призначення. Початковою ціллю розроблювання була реалізація в операційній системі UNIX, проте згодом дана мова програмування була перенесена на багато інших платформ. Конструкції цієї мови тісно пов'язані із типовими машинними інструкціями, відповідно вона знайшла застосування у проектах, задля котрих доцільно застосовувати мову асемблера, тобто як у операційних системах, так та

у різнотипних прикладних програмах задля багатьох приладів – з суперкомп'ютерів до вбудованих систем. Мова програмування С суттєво вплинула на розвиток індустрії програмного забезпечення, та його синтаксис став основою задля таких мов програмування, як С++, С# і Java. Мова С була розроблена як мова системного програмування, задля якої можливо створити однопрохідний компілятор. Стандартна бібліотека мови разом з цим не дуже велика. Відповідно ця мова доступна на багатьох платформах. Крім того, незважаючи на свій низький рівень, мова орієнтована на багатоплатформність. Метою мови було полегшити написання великих програм із мінімізацією помилок в порівнянні із асемблером, дотримуючись принципів процедурного програмування, проте уникаючи всього, що спроможне призвести до додаткових накладних витрат, характерних задля мов високого рівня. Можливо виділити наступні основні особливості С:

1. Простий синтаксис, із якого в стандартну бібліотеку було перенесено багато основних функцій, таких як математичні процедури чи процедури роботи із файлами;

2. Орієнтація на процедурне програмування;

3. Слаба статична типізація;

4. Застосування препроцесора задля абстрагування подібних операцій;

5. Доступ до пам'яті поза поміччю вказівників;

6. Невелика кількість ключових слів;

7. Наявність покажчиків на процедури та статичних змінних;

8. Області імен.

Однак в С відсутні:

1. Вкладені процедури;

2. Засоби автоматичного управління пам'яттю;

3. Вбудовані засоби об'єктно-орієнтованого програмування;

4. Засоби функціонального програмування.

Деякі із відсутніх можливостей можуть існувати змодельовані поза поміччю вбудованих інструментів (наприклад, імітацію значень можливо

					<i>РП 07. 20 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

виконувати поза помічно функцій `setjmp` та `longjmp`), деякі додаються поза помічно сторонніх бібліотек (наприклад, задля підтримки багатозадачності і задля мережевих функцій можуть застосовувати `pthread`, сокети тощо, є бібліотеки задля підтримки автоматичного збору сміття), частина реалізована у деяких компіляторах як розширення мови (наприклад, вкладені процедури у компілятор `GCC`). Разом з цим існує дещо громіздкий, проте цілком працездатний метод, який надає реалізувати механізми ООП, засновані на фактичному поліморфізмі вказівників в мові `C` та підтримці вказівників на процедури в цій мові. Механізми ООП реалізуються у бібліотеці `Glib` та активно використовуються у рамках фреймворку `GTK+`. `Glib` забезпечує базовий клас `GObject`, здатність спадкування з одного класу та реалізацію декількох інтерфейсів. У загалом, мова досить розповсюджена, що виступає гарантом того, що мова підтримується на різноманітних платформах і задля неї існує сформована і вичерпна документація, яку зручно застосовувати задля розроблювання програмного забезпечення, а отже програмісти містять можливість досить точно зрозуміти, як саме виконуються їхні програми. Завдяки своїй близькості до мов низького рівня програми на мові `C` працюють більш ефективно, аніж написані на багатьох інших мовах високого рівня.

Мова програмування `C++` – це складена, статично типізована мова програмування загального призначення. Підтримує такі парадигми програмування, як процедурне програмування, об'єктно-орієнтоване програмування, загальне програмування, а разом з цим може багату стандартну бібліотеку, котра включає загальні контейнери і алгоритми, регулярні вирази, підтримку багатопоточності і інші процедури. `C++` поєднує властивості як мов високого рівня, так та мов низького рівня. В порівнянні зі своїм попередником, мовою `C`, найбільшу увагу приділено підтримці об'єктно-орієнтованого та узагальненого програмування. Мова `C++` широко використовується задля розроблювання програмного забезпечення і є однією із найпопулярніших мов програмування. У загалом існує багато реалізацій даної мови програмування, як вільних, так та комерційних, та задля різноманітних платформ. Разом з цим варто

					РП 07. 20 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

вказати, що C++ мала величезний вплив на інші мови програмування, насамперед на Java та C#. Синтаксис C++ успадковується з C, адже одним із основних принципів розроблення була підтримка сумісності із C. Мова C++ містить засоби розроблення програмного забезпечення із контрольованою ефективністю задля широкого кола задач, з утиліт низького рівня і драйверів до дуже складних програмних пакетів. Серед основних переваг цієї мови можливо виділити наступні:

1. Висока сумісність із мовою C. Тобто код C можливо компілювати із мінімальними модифікаціями компілятором C++. Зовнішній мовний інтерфейс прозорий, відповідно бібліотеки C можливо викликати із C++ без додаткових витрат;

2. Як наслідок попереднього пункту – обчислювальна продуктивність. Ця мова покликана дати програмісту максимальний контроль над усіма аспектами структури і порядку створення програми. Жодна із функцій мови, котрі призводять до додаткових накладних витрат, не є обов'язковими задля застосування. Разом з цим існує можливість працювати із пам'яттю на низькому рівні;

3. Підтримка різнотипних стилів програмування: традиційне імперативне програмування (структуроване, об'єктно-орієнтоване), узагальнене програмування, функціональне програмування;

4. Автоматичний виклик деструкторів об'єктів у належному порядку (зворотний виклик задля конструкторів) спрощує та підвищує надійність керування пам'яттю і іншими ресурсами (відкриті файли, мережні підключення, із'єднання із базами даних тощо);

5. Перевантаження оператора надає коротко та просторо писати вирази задля користувацьких типів у природному алгебраїчному вигляді;

6. Можливо контролювати константність об'єктів (модифікатори const, mutable, volatile). Застосування постійних об'єктів підвищує надійність та служить підказками задля оптимізації;

7. Шаблони C++ [6] дозволяють створювати загальні контейнери і алгоритми задля різнотипних типів даних. Разом з цим шаблони дозволяють виконувати розрахунки на етапі компіляції;

8. Можливість вбудовування предметно-орієнтованих мов програмування у основний код. Цей підхід використовує, наприклад, бібліотеку Boost.Spirit, котра надає вказувати граматику EBNF парсерів безпосередньо у C++ кодї;

9. Доступність. Задля C++ існує величезна кількість підручників, перекладених на різні мови. ТА хоча мова може високий поріг вступу, проте серед усіх мов такого роду він може найширші можливості. Із попередньо перерахованих пунктів впливає те, що дана мова містить в собі весь необхідний інструментарій задля створення майже будь-якої програми, а її розповсюдженість є гарантом наявності вичерпної документації.

Мова C# – об'єктно-орієнтована мова програмування, розроблена групою інженерів Microsoft як мова розроблювання додатків задля Microsoft .NET Framework. Мова C# відноситься до сімейства мов із C-подібним синтаксисом та його синтаксис найбільш близький до мов C++ чи Java. Мова C# багато перейняла з своїх попередників – C++, Pascal, Modula, Smalltalk та, зокрема, Java. Відповідно C#, виходячи із практики їх застосування, виключає деякі моделі, що виявилися проблематичними при розробці програмного забезпечення. Наприклад, C#, на відміну з C++ та деяких інших мов, не підтримує наслідування з декількох класів. Мова C# була розроблена як мова на рівні додатків задля CLR та відповідно залежить насамперед з можливостей самого CLR. Це стосується, у першу чергу, системи типу C#, котра відображає BCL. Наявність чи відсутність певних виразних ознак мови диктується тим, чи спроможне конкретна мовна особливість існувати переведена в відповідні конструкції CLR. ТА хоча таку взаємодію та слід очікувати у майбутньому, вона частково була порушена із виходом C# 3.0, котра є розширенням мови, що не покладається на розширення .NET. Разом з цим C# підтримує інструменти задля безпосередньою роботи із пам'яттю, проте такі шматки коду містять позначатися ідентифікатором unsafe, і вимагають спеціального дозволу. Згідно із

інформацією, наданою сайтом [mattwarren.org](http://mattwarren.org), код, написаний на C# в порівнянні із C/C++ є приблизно на 30% повільніший поза рахунок більшої різниці синтаксису з низькорівневого машинного коду, який виконується комп'ютером. Та хоча існуюча документація стандартних бібліотек спроможне відрізнитися з версії до версії, у загалом вона цілком прийнятна і добре зрозуміла задля сприйняття, що разом з цим гарантується великою кількістю розробників, що спеціалізуються на даній мові.

Мова Python – мова програмування загального призначення, орієнтована на підвищення продуктивності розробника та читаності коду. Синтаксис, підтримуваний ядром Python, може досить малий об'єм. Однак стандартна бібліотека включає у себе велику кількість корисних додаткових функцій. Це разом з цим створює необхідність досліджувати сторонні бібліотеки на наявність потрібного у даному випадку модулю. Python підтримує структуроване, об'єктно-орієнтоване, функціональне, імперативне і аспектно-орієнтоване програмування. Основними архітектурними особливостями є динамічна типізація, автоматичне управління пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопотокових обчислень, високорівневі структури даних. Python є мовою програмування, що активно розвивається, нові версії із додаванням/зміною властивостей мови виходять приблизно раз на два із половиною роки. Мова не підлягала офіційній стандартизації, роль стандарту фактично виконувала CPython, розроблена під контролем автора мови. Варто зазначити, що портування ядра Python виконано майже задля всіх платформ, однак в той же час, на відміну з багатьох інших портативних систем, Python підтримує технології, характерні задля цієї платформи. Прикладом спроможне слугувати Microsoft COM / DCOM. Більш того, існує версія Python розроблена задля Java – Jython, котра надає інтерпретатору працювати на усіх системах, котрі підтримують Java, тоді як Java-класи можливо застосовувати безпосередньо із Python та навіть написані на Python. Разом з цим декілька розширень містять поза мету забезпечити сумісність Microsoft .NET, головною із котрих є IronPython та Python.Net. Із зазначених вище тез впливає те, що Python

					РП 07. 20 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

хоча та залежний з встановлення свого ядра, так само як та Java, однак спроможне виконуватися на великій кількості сучасних платформ. Разом з цим Python є популярною мовою програмування, насамперед задля рішень різнотипних математичних алгоритмів. Із цього випливає величезна кількість вичерпної технічної документації, котра майже повністю покриває опис доступних модулів.

У даному підрозділі буде проведено порівняння мов програмування поза такими параметрами як: швидкодія, зручність синтаксису, гнучкість засобів задля роботи із компіляторами і доступність навчальних засобів і документації. Оцінка буде проводитись поза шкалою з 0 до 10 і є відображенням власної думки автора. Дані порівняння можливо переглянути в табл. 1.4.

Таблиця 1.4. Порівняльна характеристика властивостей мов програмування

	Java	C	C++	C#	Python
Швидкодія	7	10	10	8	8
Синтаксис	8	8	8	10	9
Засоби	9	5	10	8	9
Документація	10	9	10	10	10
Середня оцінка	8	8	9.5	9	9

У даному випадку вирішено віддати перевагу мові C++, відповідно що дана мова поєднує в собі властивості високорівневих мов програмування із можливістю роботи із пам'яттю, а разом з цим може велику кількість різноманітних матеріалів на дану тему. Отже, беручи до уваги власний досвід в програмуванні у інтегрованому середовищі розроблення RAD Studio C++Builder і вищезазначені аргументи можливо вважати доцільним обрати мовою реалізації програмного застосунку задля обчислення схем півпровідникових випрямників саме мову програмування C++ і ІСР Embarcadero RAD Studio C++Builder.



оптимізує компілятор. Проте сучасних розробників хвилює не тільки ефективність виконуваного коду. Їм потрібний засіб, який був би, із одного боку, досить потужним та гнучким, щоб виконати будь-яке завдання, що стоїть перед ними, та, із іншого боку, досить простим та зручним у роботі. Розробники хочуть мати систему, побудовану на принципах об'єктно-орієнтованого програмування, та у той же час, що надає застосовувати, в разі потреби, асемблер. Їм потрібна система, котра генерувала б не тільки виконувані файли в форматі EXE, проте та могла створювати файли DLL та драйвери приладів.

До складу C++Builder входить велика бібліотека компонентів, поза поміччю якої можливо уникнути ручного написання програм, що широко поширене у інших засобах розроблювання. Можливо створювати додатки в візуальному режимі, просто поміщаючи потрібні компоненти на форму та, у той же час, зберігаючи доступ до всіх функцій програмного інтерфейсу Windows, системних повідомлень та процесів. Працюючи у C++Builder програміст спроможне поза поміччю натиснення однієї клавіші створити виконуваний файл в форматі EXE, проте, при необхідності, можливо компілювати та файли DLL, драйвери приладів, а разом з цим консольні додатки.

Використовуючи C++Builder можливо створювати додатки задля MS Windows із мінімальними витратами часу оскільки у її основі лежить концепція швидкого створення додатків (RAD).

Інтегроване середовище розроблювання додатків надає створювати, компілювати, тестувати та редагувати проект чи групу проектів у єдиному середовищі програмування.

Візуальна технологія розроблювання програм надає швидко створювати додатки шляхом розміщення в формі стандартних компонентів. При цьому відповідний код програми автоматично генерується C++Builder. Така технологія визволяє розробника з рутинної роботи із створення призначеного задля оператора інтерфейсу та надає приділити більш уваги внутрішній організації даних та обробці даних.

Технологія Two Ways Tools робить ефективнішою роботу із компонентами. При зміні програмного коду в вікні редактора C++Builder відповідним чином змінює та самі компоненти. Із іншого боку, при зміні властивостей компонентів у інспекторві редактора об'єктів (Object Inspector) вони негайно відбиваються в вікні редактора коду.

Насамперед C++Builder призначений задля професіоналів-розробників корпоративних інформаційних систем. Проте C++Builder призначений не тільки задля програмістів-професіоналів. Будь-який програміст на C++ здатний практично відразу професійно освоїти C++Builder.

На відміну з більшості компіляторів, що транслюють у р-код, у C++Builder програмний текст компілюється безпосередньо у машинний код, внаслідок чого C++Builder-додатки виконуються у 10-20 разів швидше (особливо додатки, що використовують математичні процедури). Готовий додаток спроможне існувати виготовлений чи в вигляді виконуваного модуля, чи в вигляді динамічної бібліотеки, яку можливо застосовувати у додатках, написаних на інших мовах програмування. Завдяки такій архітектурі додатки, виготовлені поза поміччю C++Builder, працюють надійно та стійко. Embarcadero RAD Studio C++Builder підтримує застосування вже існуючих об'єктів, включаючи DLL, написані на C та C++, OLE сервера, VBX, об'єкти, створені поза поміччю C++Builder. Із готових компонент працюючі додатки збираються дуже швидко.

Бібліотека об'єктів VCL включає стандартні об'єкти побудови призначеного задля оператора інтерфейсу, об'єкти управління даними, графічні об'єкти, об'єкти мультимедіа, діалоги та об'єкти управління файлами, управління DDE та OLE.

## **1.6 Реалізація алгоритму обчислення півпровідникових випрямників із ємнісною відповіддю навантажування**

Алгоритм обчислення півпровідникових випрямників із ємнісною відповіддю навантажування складено по методиці обчислення, розглянутій в підрозділі 1.1.

					<b>РП 07. 20 000. 00 ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

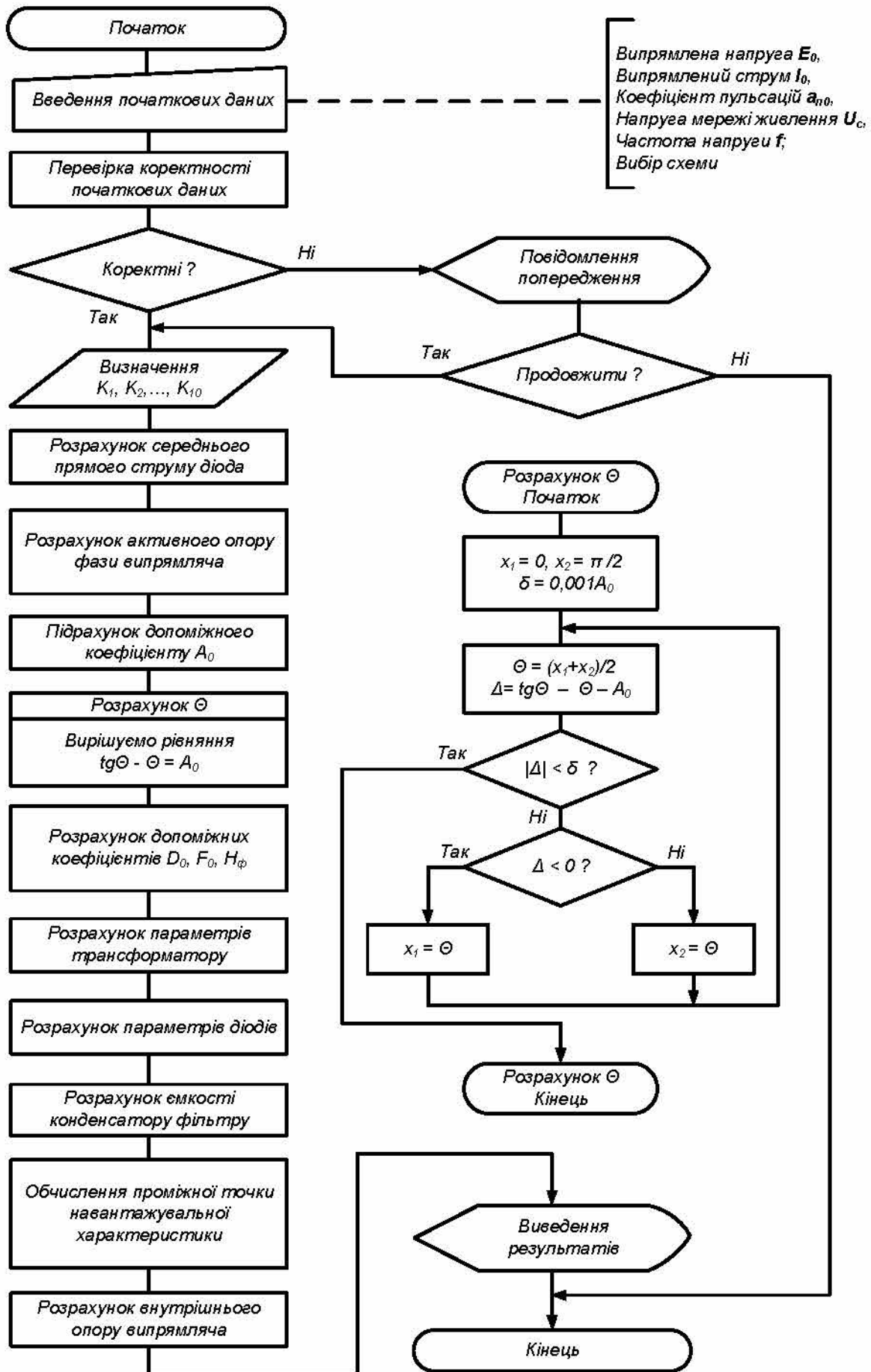


Рисунок 1.5. Алгоритм обчислення випрямника ємнісного навантажування

Слід ще раз звернути увагу, що при ручному обчислення деякі параметри та допоміжні значення визначаються по графіках та таблицях. При програмній реалізації алгоритму використовуються відповідні апроксимуючі формули, по котрих отримують приблизне значення. При дотриманні обмежень на вхідні дані отримуваний результат може достатній ступінь точності. Інакше, при порушенні певних умов, у алгоритмі передбачено виведення повідомлення про зниження точності обчислення.

Після введення початкових даних здійснюється їх діагностика на наявність їх некоректності. На вхідні дані накладаються наступні обмеження:

- недопустимі негативні числа;
- вихідна потужність ( $E_0 I_0$ ) не повинна перевищувати 500 Вт;
- вихідна напруга  $E_0$  не менше 3 В;
- періодичність напруги  $f$  не більш 5 кГц;
- задля двохнапівперіодних трифазних схем рівень імпульсів  $a_{\text{п0}}$ .

Поза наявності у початкових даних негативних чисел видається помилка та подальший обчислення неможливий. Якщо порушена якась інша умова, то буде тільки видане попередження про зниження точності результату та питання про необхідність продовження обчислення.

На рис. 1.5 представлено розроблений алгоритм обчислення задля діодного приладу із ємнісною відповіддю навантажування. Нижче представлено фрагмент програми, що реалізовує даний алгоритм мовою C++, і відповідні коментарі, що пояснюють роботу програми:

```
// Функція, що викликається при обчислення випрямника
// із ємнісною відповіддю навантажування
void __fastcall TApp::btnCapacityClick(TObject *Sender)
{
    // -----
    // Обчислення випрямника із ємнісною відповіддю навантажування
    // -----
    /* id – вибрана схема:
       0 - однонапівперіодна
       1 - однонапівперіодна двофазна С.Т.
       2 - двохнапівперіодна однофазна мостова
```



```

double B, C, D; // Допоміжні значення
double ТБ = M_PI/2, ТН = 0, ТТ; // ТТ – кут відсічення струма
// цикл (поки не буде досягнута потрібна точність 0,1%)
while( true )
{
    ТТ = (ТБ + ТН)/2;
    В = tan(ТТ) - ТТ - V;
    С = .001*V;
    if( C >= fabs(B) || !B )
        break;
    else
        if( B < 0 ) ТН = ТТ;
        else ТБ = ТТ;
}
//----- Обчислення допоміжних коефіцієнтів -----
D = sqrt( M_PI*( ТТ*(1+.5*cos(2*ТТ)) - .75*sin(2*ТТ)) /
(sin(ТТ) - ТТ*cos(ТТ)));
В = M_PI*(1-cos(ТТ))/(sin(ТТ)-ТТ*cos(ТТ));
if( K[3]> 1 ) // K[3] - число фаз випрямлення
{
    С = 101E3*( sin(K[3]*ТТ)*cos(ТТ) -
    K[3]*cos(K[3]*ТТ)*sin(ТТ) );
    С = C/( K[3]*(pow(K[3],2)-1)*cos(ТТ) );
}
else
{
    С = 25330*(2*ТТ - sin(2*ТТ))*cos(ТТ);
}
//----- Обчислення вихідних значень -----
double Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10, Y11, Y12;
// напруження Х.Х. трансформатору
Y1 = E*K[4]/1.41/cos(ТТ);
// обернена напруження вентиля
Y2 = Y1*K[9];
// ефективні струми первинної та вторичної обмоток
Y3 = K[10]*A*Y1/U;
Y4 = K[5]*A*D;
// ефективний та імпульсний струми вентиля
Y5 = K[6]*Y4;
Y6 = В*A/K[3];
// ємність конденсатору
Y7 = C/(X*P*F);
// розсіювана діодом потужність
Y8 = pow(Y5,2) *.2/Y;
// габаритна потужність трансформатору

```

					<i>РП 07. 20 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

```

Y9 = K[7]*E*A;
// напруження та current задля побудови навантажувальної хар-ки
Y10 = 1.41*Y1*cos(TT/2)/K[4];
Y11 = .45*K[3]*Y1*(sin(TT/2)-TT/2*cos(TT/2))/K[4]/X;
// внутрішній супротив випрямника
Y12 = (Y10 - E) / (A - Y11);
//----- Готуємо дані задля виводу -----
// виводимо (якщо треба) інформацію про початкові дані
FillRectifierInputText();
AnsiString t;
char buf[128];
t = "\r\n ВИМОГИ ДО ЗНАЧЕНЬ ТРАНСФОРМАТОРУ:\r\n";
sprintf(buf " %-45s%6.1f\r\n", "Габаритна потужність, Вт",
Y9); t += buf;
sprintf(buf " %-45s%6.1f\r\n", "ЕДС вторичної котушки, У", y1);
t += buf; sprintf(buf " %-45s%6.2f\r\n", "Ефективний current вторичної
котушки, А, Y4); t += buf;
sprintf(buf " %-45s%6.3f\r\n", "Ефективний current первинної котушки, А,
Y3); t += buf;
t += "\r\n ВИМОГИ ДО ЗНАЧЕНЬ ДІОДІВ:\r\n";
sprintf(buf " %-45s%6.1f\r\n", "Обернена напруження, У", Y2);
t += buf;
sprintf(buf " %-45s%6.2f\r\n", "Середній спрямлений current, А, Y); t +=
buf;
sprintf(buf " %-45s%6.2f\r\n", "Імпульсний спрямлений current, А, Y6); t +=
buf;
sprintf(buf " %-45s%6.2f\r\n", "Виділяема потужність, Вт", Y8); t += buf;
t += "\r\n ПРОМІЖНА ТОЧКА НАВАНТАЖУВАЛЬНОЇ
ХАРАКТЕРИСТИКИ:\r\n";
sprintf(buf " %-45s%6.3f\r\n", "Current, А, Y11); t += buf;
sprintf(buf " %-45s%6.1f\r\n\r\n", "Напруження, У", Y10);
t += buf;
sprintf(buf " %-45s%6.0f\r\n", "Ємність конденсатору, мкФ", Y7);
t += buf;
sprintf(buf " %-45s%6.2f\r\n", "Внутрішній супротив випрямника, Ом",
Y12); t += buf;
// strRectifierResult – рядок, що містить рез-ти обчислення
strRectifierResult = t;
// Виведення даних на екран
// Якщо відмічена галочка "+ вхідні дані"
// тоді вони теж будуть виведені
checkInputRectifierClick(0);
}
// Функція, що перевіряє вхідні дані
// У разі помилки повертає false

```

```

bool __fastcall TApp::Rectifier_GetInputValues(
    double& U, int& F, double& E, double& A, double& P )
{
    // Записуємо початкові дані у ці змінні
    U = editU->Text.ToDouble(); // Напруження мережі живлення, У
    F = editF->Text.ToInt(); // Періодичність напруження живлення, Гц
    A = editA->Text.ToDouble(); // Випрямлений current, А
    E = editE->Text.ToDouble(); // Спрямлена напруження, У
    P = editP->Text.ToDouble(); // Коэф. імпульсів
    bool result = U > 0 && F > 0 && A > 0 && E > 0 && P > 0;
    if( !result )
    {
        // Негативні числа недопустимі
        ErrMess( "Вхідні дані повинні існувати\nпозитивними числами!" );
    }
    else
    {
        // Перевірка решти обмежень
        if( E < 3 )
        {
            if( IDNO == Application->MessageBox(
                "Вихідна напруження < 3 У.\nТочність обчислення
знижується.\n\nПродовжити?", "УВАГА", MB_YESNO) )
                // Якщо натиснули "Відміна", то не продовжуємо обчислення
                result = false;
        }
        if( result && E*A > 500 )
        {
            if( IDNO == Application->MessageBox(
                "Вихідна потужність перевищує 500 Вт.\nТочність обчислення
знижується.\n\nПродовжити?",
                "УВАГА", MB_YESNO) )
                result = false;
        }
        if( result && F > 5000 )
        {
            if( IDNO == Application->MessageBox(
                "Періодичність > 5 кГц. Точність обчислення
знижується.\n\nПродовжити?",
                "УВАГА", MB_YESNO) )
                result = false;
        }
    }
    return result;
}

```



## 1.7 Реалізація алгоритму обчислення півпровідникових випрямників із індуктивною відповіддю навантажування

Методику, по якій складено алгоритм обчислення півпровідникових випрямників із індуктивною відповіддю навантажування, розглянуто в підрозділі 1.1.

На почальні дані задля обчислення випрямників із індуктивною відповіддю навантажування накладаються ті ж обмеження, що та задля випрямників, котрі працюють на ємність.

На рис. 1.6 представлено розроблений алгоритм обчислення задля діодного приладу із індуктивною відповіддю навантажування. Нижче представлено фрагмент програми, що реалізовує даний алгоритм мовою C++, і відповідні коментарі, що пояснюють роботу програми:

```
// Функція, що викликається при обчислення випрямника
// із індуктивною відповіддю навантажування
void __fastcall TApp::btnInductiveClick(TObject *Sender)
{
    // Значення K1-K14 задля кожної із шести схем;
    // перші значення порожні (=0), щоб потім було
    // зручніше посилатися (тобто не K[0], а K[1])
    double T[6][15]= {
        { 0, .5, 1, 2, 1, 2.828, 1.11, .707, .707, 1.34, .67, 1.414, 7, .0055, 1 },
        { 0, .5, 1, 2, 2, 1.414, 1.11, 1, .707, 1.11, .67, 1.414, 5.2, .0064, 1 },
        { 0, .333, 1, 3, 1, 2.449, .855, .58, .58, 1.35, .25, 1.414, 6.6, .0033, .47 },
        { 0, .333, 2, 6, 2, 2.449, .43, .82, .58, 1.045, .057, 2.449, 2.5, .001, .82 },
        { 0, .333, .667, 6, 2, 1.449, .74, .41, .58, 1.045, .057, 1.414, 7.6, .0033, .47
    },
        { 0, .167, 1, 6, 1, 2.828, .74, .41, .41, 1.43, .057, 1.414, 6, .0033, .47 }
    };
    /* id – вибрана схема:
    0 - однонапівперіодна двофазна С.Т.
    1 - двухнапівперіодна однофазна мостова
    2 - однонапівперіодна трифазна
    3 - двухнапівперіодна трифазна зірка
    4 - двухнапівперіодна трифазна трикутник
    5 - однонапівперіодна шестифазна зірка
    */
    int id = comboScheme->ItemIndex;
    // Масив коефіцієнтів K1, ..., K14 задля вибраної схеми
```

```

double *K = T[id];
double U      // Напруження мережі живлення, У
      E      // Спрямлена напруження, У
      A,    // Випрямлений current, А
      P;    // Коеф. імпульсів
int F;      // Періодичність напруження живлення, Гц
// Набуваемо значень вхідних даних (там же йде перевірка)
if( !Rectifier_GetInputValues(U, F, E, A, P) )
    return;    // якщо помилка, то виходимо

// Задля схем 2п3ф рівень імпульсів д.б. < 0.07
if( id == 3 || id == 4) && P > 0.007 )
    if( IDNO == Application->MessageBox(
        "Рівень імпульсів д.б. < 0.07\n\nПродовжити?"
        "УВАГА", MB_YESNO)
    )
        return;
    // індукція трансформатору
double B = 1.2 - .4*sin(.003*E*A);
// супротив трансформатору
double RT = K[12]* E/F/B/A * sqrt(sqrt(F*B/E/A));
// індуктивність розсіяння трансформатору
double XS = K[13]* E/F/A/B/sqrt(sqrt(F*B/E/A));
    //----- Обчислення вихідних значень -----
double Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10, Y11, Y12, Y13, Y14,
Y15, Y16, Y17;
    // спрямлена напруження при Х.Х. та C = 0
Y1 = E + K[2]*A*RT + K[3]*F*XS*A + .6*K[4]+ .05*E;
// напруження Х.Х. трансформатору
Y2 = Y1*K[6];
// ефективні струми первинної та вторичної обмоток
Y3 = K[14]*Y2*A/U;
Y4 = K[7]*A;
// габаритна потужність трансформатору
Y5 = K[9]*Y1*A;
// середній, ефективний та імпульсний струми вентиля
Y6 = K[1]*A;
Y7 = K[8]*A;
Y8 = A;
// обернена напруження вентиля
Y9 = K[5]*Y2;
// розсіювана діодом потужність
Y10 = .6*Y6;
// мінімальна індуктивність дроселя
Y11 = .637*E/(pow(K[3],2) - 1) /K[3]/F/A;

```

						<b>РП 07. 20 000. 00 ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата			44

```

// рівень згладжування фільтрів
double Q = K[10]/P;
// ємність фільтра
Y12 = 25330*Q/pow(K[3],2) /pow(F,2) /Y11;
// робоча напруження конденсатору
Y13 = K[11]*Y2;
// критичний current
Y14 = E/(pow(K[3],2)-1) /K[3]/M_PI/F/Y11;
// внутрішній супротив випрямника
Y15 = (Y1-E)/A;
// критична напруження
Y16 = E + (A-Y14)*Y15;
// спрямлена напруження X.X. випрямника при C > 0
Y17 = K[11]*Y2;
//----- Виводимо дані -----
FillRectifierInputText();
AnsiString t;
char buf[128];
t = "\r\n ВИМОГИ ДО ЗНАЧЕНЬ ТРАНСФОРМАТОРУ:\r\n";
sprintf( buf " %-45s%6.2f\r\n"
"Габаритна потужність, Вт", Y5 ); t += buf;
sprintf( buf " %-45s%6.2f\r\n"
"Напруження холостого ходу, У", Y2 ); t += buf;
sprintf( buf " %-45s%6.3f\r\n"
"Ефективний current вторичної котушки, А, Y4 ); t += buf;
sprintf( buf " %-45s%6.3f\r\n"
"Ефективний current первинної котушки, А, Y3 ); t += buf;
t += "\r\n ВИМОГИ ДО ЗНАЧЕНЬ ДІОДІВ:\r\n";
sprintf( buf " %-45s%6.2f\r\n"
"Обернена напруження, У", Y9 ); t += buf;
sprintf( buf " %-45s%6.2f\r\n"
"Середній спрямлений current, А, Y6 ); t += buf;
sprintf( buf " %-45s%6.2f\r\n"
"Імпульсний спрямлений current, А, Y8 ); t += buf;
sprintf( buf " %-45s%6.3f\r\n\r\n"
"Потужність, що виділяється, Вт", Y10 ); t += buf;
sprintf( buf " %-45s%6.2f\r\n"
"Мінімальна індуктивність дроселя, мГн", Y11*1000 );
t += buf;
sprintf( buf " %-45s%6.2f\r\n"
"Спрямлена напруження при X.X. та C = 0, У", Y1 );
t += buf;
sprintf( buf " %-45s%6.2f\r\n"
"Спрямлена напруження при X.X. та C > 0, У", Y17 );
t += buf;

```

					РП 07. 20 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

```

printf( buf " %-45s%6.0f\r\n"
"Ємність конденсатору, мкФ", Y12 ); t += buf;
printf( buf " %-45s%6.2f\r\n"
"Робоча напруга конденсатору >=, У", Y17 ); t += buf;
printf( buf " %-45s%6.2f\r\n"
"Внутрішній супротив випрямника, Ом", Y15 ); t += buf;
t += "\r\n КРИТИЧНА ТОЧКА НАВАНТАЖУВАННЯ
ХАРАКТЕРИСТИКИ:\r\n";
printf( buf " %-45s%6.2f\r\n", "Current, А, Y14 ); t += buf;
printf( buf " %-45s%6.2f\r\n", "Напруга, У", Y16 );
t += buf;
strRectifierResult = t;
checkInputRectifierClick(0);
}

```

## 1.8 Реалізація алгоритму обчислення півпровідникових випрямників, що живляться напругою прямокутної форми

Методика, по якій складено алгоритм обчислення півпровідникових випрямників, що живляться напругою прямокутної форми, розглянуто в підрозділі 1.1.

На початкові дані задля обчислення випрямника, що живиться напругою прямокутної форми, накладаються ті ж обмеження, що та задля випрямників, котрі працюють на індуктивність і ємність.

На рис. 1.7 представлено алгоритм обчислення діодного приладу задля випрямника, що живиться напругою прямокутної форми. Нижче представлено фрагмент програми, що реалізовує даний алгоритм мовою С++, і відповідні коментарі, що пояснюють роботу програми:

```

// Функція, що викликається при обчислення випрямника, який
// живиться змінною напругою прямокутної форми
void __fastcall TApp::btnRFClick(TObject *Sender)
{
    double U1,      // Вхідна напруга, У
           U0,      // Спрявлена напруга, У
           Y0,     // Випрямлений current, А
           A;      // Коеф. імпульсів
    int F;         // Періодичність напруги живлення
    // Шпаруватість
    double B = editSkvazhnost->Text.ToDouble();
    // Набуваємо значень вхідних даних (там же йде перевірка)

```

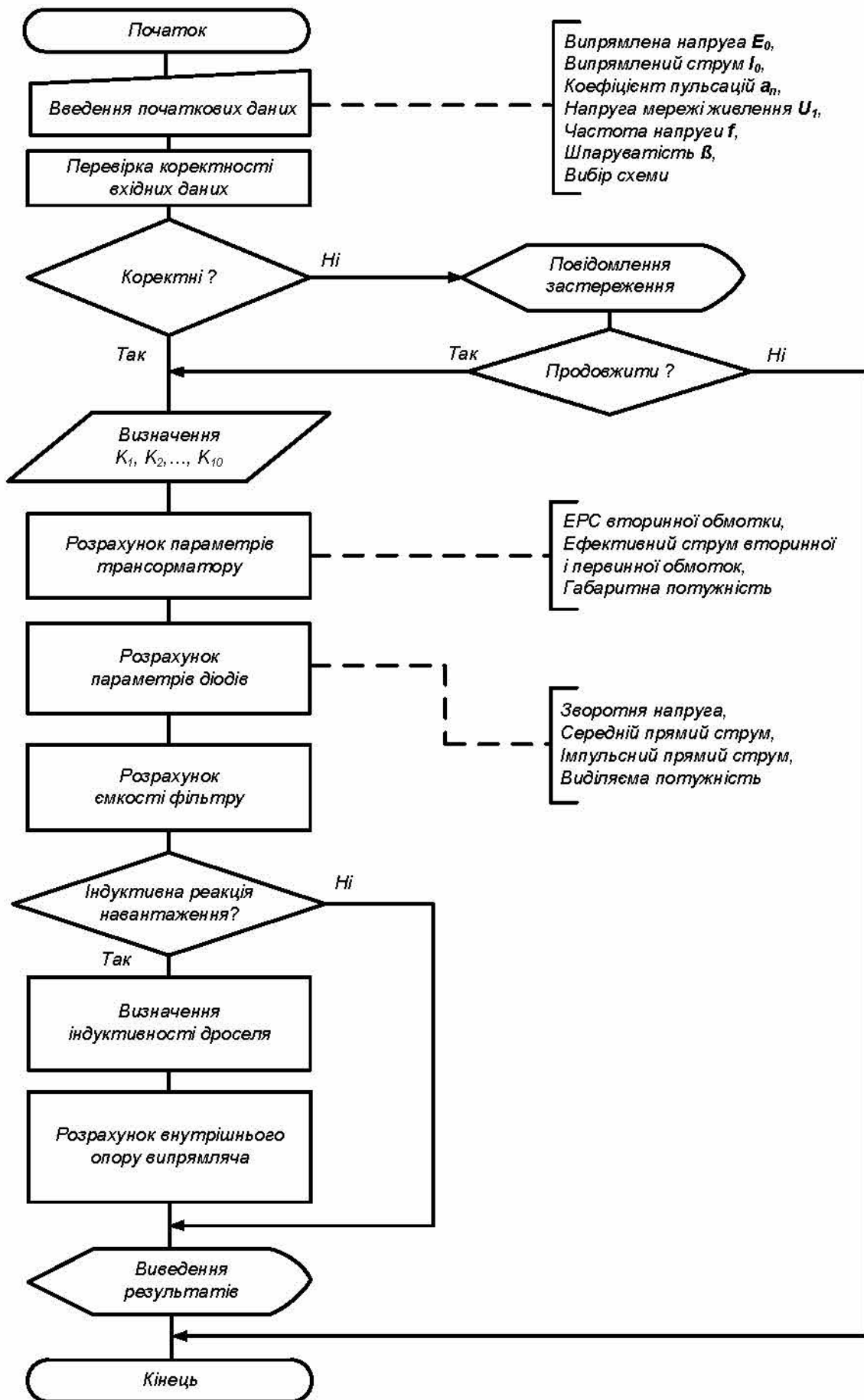


Рисунок 1.7. Алгоритм рахунку задля випрямника, що живиться напруженням прямокутної форми

```

        if( !Rectifier_GetInputValues(U1, F, U0, Y0, A) )
            return; // якщо помилка, то виходимо
//----- Обчислення вихідних значень -----
double U2, // напруження Х.Х. трансформатору
        U10, // обернена напруження вентиля
        Y1, Y2 // струми первинної та вторичної обмоток
        YI, // імпульсний current вентиля
        C, // ємність конденсатору
        PI, // розсіювана діодом потужність
        P, // габаритна потужність трансформатору
        R, // внутрішній супротив випрямника
        Y, // середній current вентиля
        D; // індуктивність дроселя
double RT;
double G = sqrt(1.2*U0/(pow(Y0,3)*F));
/* id – схема
Задля індуктивної реакції навантажування:
0 - із середньою крапкою
1 - мостова
Задля ємкісної реакції навантажування:
0 - однонапівперіодна
1 - однонапівперіодна двофазна С.Т.
2 - двохнапівперіодна однофазна мостова
3 - двохнапівперіодна однофазна подвоєння */
int id = comboSchemeRF->ItemIndex;
if( !comboReaction->ItemIndex )
{
    // індуктивна реакція навантажування
    int k12 = id + 1; // 1 чи 2
    int k21 = 2 - id; // 2 чи 1
    RT = G * (lid ? 5.5: 4.08);
    U2 = 1.01*U0 + RT*Y0 + k12;
    R = k21*(U2-U0)/Y0;
    P = lid ? (1 + sqrt(B+1))*U2*Y0/2: 1.1*U2*Y0;
    Y2 = Y0*sqrt(B/k21);
    U10= k21*U2*B;
    Y1 = 1.1*U2*Y2*sqrt(B) /U1;
    Y = Y2/2;
    YI = Y0*B;
    PI = Y0/2;
    D = .1*B*U0*sin(M_PI/B)/(F*Y0);
    C = 25E5*Y0/(pow(M_PI,3)*A*U0*F);
}
else
{

```

```

// ємкісна реакція навантажування
if( id == 0 )
{
    RT = 1.9*G;
    U2 = U0 + 2*Y0*B*(RT + sqrt(2/B)/(2*Y0));
    P = 1.1*U2*Y0*(.7+sqrt((2*B-1)/B) /2);
    Y1 = 1.1*U2*Y0*sqrt((2*B-1)/B) /U1;
    Y2 = Y0*sqrt(2*B);
}
else if( id == 1 )
{
    RT = 3.9*G;
    U2 = U0 + Y0*B*(RT+sqrt(2/B) /Y0);
    P = 1.3*Y0*U2;
    Y1 = 1.1*Y0*U2/U1;
    Y2 = sqrt(B/2)*Y0;
}
else if( id == 2 )
{
    RT = 2.9*G;
    U2 = U0 + 2*Y0*B*(RT+2*sqrt(2/B) /Y0);
    P = 1.1*Y0*U2;
    Y1 = 1.1*Y0*U2/U1;
    Y2 = sqrt(B) /Y0;
}
else // if( id == 3 )
{
    RT = .75*G;
    U2 = U0/2 + 2*Y0*(RT+Y0*sqrt(1/B) /2);
    P = 2.2*Y0*U2;
    Y1 = 2.2*Y0*U2/U1;
    Y2 = 2*sqrt(B)*Y0;
    PI = .7*Y0*sqrt(2*B);
}
U10 = 2*U2;
if( id == 0 || id == 3 )
{
    Y = Y0;
    YI = 2*B*Y0;
    PI = Y0*sqrt(2*B);
    C = 2E6*B*Y0*sin(M_PI/2/B)/(10*U0*A*F);
}
else // if( id == 1 || id == 2 )
{
    Y = Y0/2;

```

```

        YI = B*Y0;
        PI = Y0*sqrt(B/2);
        C = 1E5*B*Y0*sin(M_PI/B)/(2*U0*A*F);
    }
}
//----- Готуємо дані задля виводу -----
// виводимо (якщо треба) інформацію про почальні дані
FillRectifierInputText();
AnsiString t;
char buf[128];
t = "\r\n ВИМОГИ ДО ЗНАЧЕНЬ ТРАНСФОРМАТОРУ:\r\n";
sprintf( buf " %-45s%6.2f\r\n"
"Габаритна потужність, Вт", P ); t += buf;
sprintf( buf " %-45s%6.2f\r\n"
"ЕДС вторичної котушки, У", U2 ); t += buf;
sprintf( buf " %-45s%6.2f\r\n"
"Ефективний current вторичної котушки, А, Y2 ); t += buf;
sprintf( buf " %-45s%6.2f\r\n"
"Ефективний current первинної котушки, А, Y1 ); t += buf;
t += "\r\n ВИМОГИ ДО ЗНАЧЕНЬ ДІОДІВ:\r\n";
sprintf( buf " %-45s%6.2f\r\n"
"Обернена напруга, У", U10 ); t += buf;
sprintf( buf " %-45s%6.2f\r\n"
"Середній спрямлений current, А, Y ); t += buf;
sprintf( buf " %-45s%6.2f\r\n"
"Імпульсний спрямлений current, А, YI ); t += buf;
sprintf( buf " %-45s%6.2f\r\n\r\n"
"Мощність, що виділяється, Вт", PI ); t += buf;
sprintf( buf " %-45s%6.0f\r\n"
"Ємність конденсатору, мкФ", C ); t += buf;
// якщо індуктивна реакція навантажування
if( !comboReaction->ItemIndex )
{
    sprintf( buf " %-45s%6.4f\r\n\r\n"
"Індуктивність дроселя, Гн", D ); t += buf;
    sprintf( buf " %-45s%6.2f\r\n"
"Внутрішній супротив випрямника, Ом", R); t += buf;
}
// strRectifierResult – рядок, що містить дані обчислення
strRectifierResult = t;
// Виведення даних на екран
// Якщо відмічена галочка "+ вхідні дані"
// тоді вони теж будуть виведені
checkInputRectifierClick(0);
}

```









В таблиці 1.5 представлені дані порівняння ручного та автоматизованого розрахунків. Слід зазначити, що при ручному обчисленні потрібно деякі допоміжні значення визначати по графіках та разом з цим треба будувати характеристику навантажування. Ручний обчислення займає близько години, тоді як при обчисленні поза поміччю програми потрібно менше хвилини задля того, щоб ввести початкові дані та вибрати схему, отримати і роздрукувати результат. Як видно із таблиці 1.5, максимальне відхилення склало біля 10% задля внутрішнього опору випрямника. Задля приблизних розрахунків це допустиме значення. По решті значень погрішність була не більш 5%.

Очевидно, що особливо при обчисленні декількох схем та виборі оптимальної, розроблений програмний застосунок буде значно економити робочий час розробника електронної техніки та виключати появу помилок, можливих при ручному обчисленні.

					<i>РП 07. 20 000. 00 ДП ПЗ</i>	Арк.
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		55

## 2 ЕКОНОМІЧНИЙ РОЗДІЛ

### 2.1 Резюме

У даному дипломному проекті розроблено програмний застосунок задля обчислення схем півпровідникових випрямників. Створено прикладне програмне забезпечення задля ПК із встановленою ОС сімейства MS Windows, яке надає поза початковими даними отримувати необхідні параметри елементів типових схем джерел вторинного електроживлення. Програма створена на мові програмування C++ в інтегрованому середовищі візуальної розроблювання Embarcadero RAD Studio. Ефективність кожного програмного продукту визначається його якістю і ефективністю процесу розроблювання. Якість ПП визначається наступними складовими: із точки зору оператора; із позиції застосування ресурсів; створення вимог до програмного забезпечення.

Оцінка якості програмного продукту із точки зору оператора визначається необхідним на стадії функціонування розміром оперативної пам'яті ЕОТ, витратами машинного часу, пропускнуою спроможністю каналів передачі даних. Оцінка якості програмного продукту включає визначення трудомісткості та вартості його створення.

### 2.2 Визначення трудомісткості розроблювання програмного забезпечення

Тривалість розроблювання програмного продукту залежить з його обсягу, трудомісткості розроблювання, кваліфікації виконавців, а разом з цим планових термінів, визначених умовами ринку. Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, в тисячах умовних машинних команд програми аналога

Каталог аналогів

В таблиці 2.1 представлені аналоги програмного забезпечення, процедури котрих, в більшому чи меншому ступені, виконує розроблений програмний продукт. Задля нашого варіанта виділено сірим кольором.

					РП 07. 20 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

Найменування ПЗ	Обсяг процедури ПП = $V_0$ ум. машинних команд
1. ПП СУБД	1300 – 8600
2. ПП введення інформації	1800 – 8800
3. ПП оптимізаційних розрахунків	13000 – 10200

Вибравши аналог ПП, що містить  $V_0$  у умовних машинних командах, трудомісткості визначати на основі табл. 2.2.

Таблиця 2.2

Обсяг ПП, тис. умов. машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262

На підставі отриманого значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розроблювання ПП, тобто у умовах комп'ютера,  $K_k=0,7 \div 0,8$ ):  $T_{ар} = 244 \times 0,8 = 195,20$  (люд/годин).

Трудомісткість програмного продукту визначається по кожному етапу розроблювання окремо на підставі трудомісткості аналога із урахуванням складності

розроблювання, ступеня новизни та ступеня застосування у розробці стандартних модулів на підставі формул:

$$T_{тз} = T^a p \times L_1 \times K_H \quad (2.1)$$

$$T_{тп} = T^a p \times L_2 \times K_H \quad (2.2)$$

$$T_{рп} = T^a p \times L_3 \times K_H \times K_T \quad (2.3)$$

Задля обчислення необхідні наступні значення:

$L_i$  – питома вага  $i$ -го етапу розроблювання (див. табл. 2.2.);

$K_H$  – поправочний рівень, що враховує ступінь новизни (див. табл. 2.3.);  $K_T$  – поправочний рівень, що враховує ступінь застосування у розробці типових програм (див. табл. 2.4.).

Таблиця 2.3 Значення питомих коефіцієнтів трудомісткості стадії у загальній трудомісткості розроблювання ПП

Код стадії	Ступінь новизни		
	А	Б	У
ТЗ (L <sub>1</sub> )	0,15	0,12	0,12
ТП (L <sub>2</sub> )	0,16	0,15	0,11
РП (L <sub>3</sub> )	0,55	0,58	0,61

Задля нашого варіанта виділено сірим кольором.

Таблиця 2.4. Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K <sub>н</sub>
А	Принципово нові ПЗ	1,75 – 1,2
Б	ПЗ – розвиток визначеного параметричного ряду	1,0 – 0,8
У	ПЗ маючий аналог	0,7

Задля нашого варіанта виділено сірим кольором.

Таблиця 2.4 Значення коефіцієнта ступеня застосування у розробці типових програм

Ступінь охоплення реалізованих функцій розроблювального ПО типовими програмами, %	Значення K <sub>г</sub>
60 та вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Задля нашого варіанта виділено сірим кольором. Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{ТЗ}=I*L_1*K_n=195,2 *0,12*0,7= 16,40 \text{ (люд/годин)}$$

Трудомісткість розроблювання технічного проекту

$$T_{ТП}=I*L_2*K_n=195,2 *0,11*0,7 = 15,04 \text{ (люд/годин)}$$

Трудомісткість розроблювання робочого проекту

$$T_{РП}=I*L_3*K_n*K_g=195,2 *0,61*0,7*0,7= 58,35 \text{ (люд/годин)}$$

Задля подальших розрахунків визначили кількість папера, витраченого на кожен етап: технічне завдання N<sub>ТЗ</sub>=2 (стр), розробка ТП N<sub>ТП</sub>=15(стр), розробка робочого проекту N<sub>РП</sub>=25 (стр), пояснювальна записка відповідно N<sub>ПЗ</sub>=50 (стр)

Обчислення зведений в таблицю 2.6

Таблиця 2.6 Обчислення трудомісткості ПП

Найменування етапів	Обчислення, годин.		
	1	2	3
1.ТЗ	$T_{РТЗ}=16,40$	$T_{КК}=0,7 \cdot N_{ТЗ}=0,7 \cdot 2=1,4$	$T_{НК}=0,15 \cdot N_{ТЗ}=0,15 \cdot 2=0,30$
2.Розробка ТП	$T_{РТП}=15,04$	$T_{КК}=0,7 \cdot N_{ТП}=0,7 \cdot 15=10,50$	$T_{НК}=0,15 \cdot N_{ТП}=0,15 \cdot 15=2,25$
3.Розробка РП	$T_{РРП}=58,35$	$T_{КК}=0,7 \cdot N_{РП}=0,7 \cdot 25=17,5$	$T_{НК}=0,15 \cdot N_{РП}=0,15 \cdot 25=3,75$
4.Розробка ПЗ	$T_{РПЗ}=1,5 \cdot N_{ПЗ}=1,5 \cdot 50=75$	$T_{КК}=0,7 \cdot N_{ПЗ}=0,7 \cdot 50=35$	$T_{НК}=0,15 \cdot N_{ПЗ}=0,15 \cdot 50=7,5$
Усього, у т.ч.:	230,2		
- на розробку	$\Sigma T_p=152$		
- контроль керівника		$\Sigma T_{КК}=64,4$	
- нормоконтроль			$\Sigma T_{НК}=13,8$

### 2.3 Обчислення ціни програмного продукту

В цьому розділі задля визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, вартість машино – години та витрати на розробку ПЗ. Обчислення основної заробітної плати виконавців приведені в таблиці 2.7. Відповідно до статті 8 «Закону про Державний бюджет України на 2024» встановлено мінімальну заробітну плату.

Таблиця 2.7 Обчислення основної заробітної плати виконавців

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн.	Обчислення, грн.
1.Розробка ПП	152	39.26	5958,15
2.Контроль керівника	65	38,50	2502,50
3.Нормоконт-роль	14	38,50	539,00
Усього	-	-	$\Sigma Z_0=8999,15$

Зробимо обчислення матеріальних витрат на розробку ПП. Обчислення зведемо у таблицю 2.8.

					<b>РП 07. 20 000. 00 ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

Таблиця 2.8. Обчислення матеріальних витрат на розробку ПЗ

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	60	2.60	160,0
Разом	-	-	-	$V_{мга} = 160,0$
Транспортно–заготівельні витрати (10%)				$V_{трз} = 0,1 \times V_{м1} = 0,1 \times 160,0 = 16,0$
Усього				$V_{м} = V_{мі} + V_{трз} = 176,0$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості у цілому ПП поза формою, приведеною у таблиці 2.9.

Таблиця 2.9 Обчислення статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула обчислення
1. Матеріали	176,0	$V_{м}$ (див. табл. 2.7)
2. Основна заробітна плата	8999,15	$Z_o$ (див. табл. 2.6)
3. Додаткова заробітна плата	1349,87	$Z_d = 0,15 \times Z_o = 8999,15 \times 0,15$
4. Відрахування до єдиного фонду соціального внеску	2276,78	$V_{е.с.в.} = 0,22 \times (Z_o + Z_d) = 0,22 \times (8999,15 + 1349,87)$
5. Накладні витрати	2699,75	$V_{нак.} = 0,3 \times Z_o = 0,3 \times 8999,15$
6. Повна собівартість	15501,55	$C_{пов} = V_{м} + Z_o + Z_d + V_{е.с.в.} + V_{нак.} = 176,0 + 8999,15 + 1349,87 + 2276,78 + 2699,75$

Розмір прибутку, що включається у ціну, визначаємо по наступній формулі:

$$\Pi = (C_{п} * P) / 100 = (15501,55 * 10) / 100 = 1550,15 \text{ грн} \quad (2.4)$$

Де  $p$  – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$Ц_o = C_{п} + \Pi = 15501,55 + 1550,15 = 17051,70 \text{ грн} \quad (2.5)$$

Податок на додану вартість визначаємо по наступній формулі:

$$\text{ПДВ} = 0,2 * Ц_o = 17051,70 * 0,2 = 3410,34 \text{ грн}; \quad (2.6)$$

Виходячи із отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$Ц_p = Ц_o + \text{ПДВ} = 17051,70 + 3410,34 = 20462,04 \text{ грн} \quad (2.7)$$

					<b>РП 07. 20 000. 00 ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

## **3 РОЗДІЛ ОХОРОНИ ПРАЦІ І ТЕХНІКИ БЕЗПЕКИ**

### **3.1 Вступ**

Охорона праці, як соціальний чинник, відіграє на підприємстві важливу, роль оскільки, якими б важливими не були трудові здобутки, вони не можуть компенсувати людині втраченого здоров'я, а тим більш життя. Те та інше дається тільки один раз. Треба пам'ятати, що внаслідок нещасних випадків і аварій гинуть на виробництві не просто робітники і службовці, на підготовку котрих держава вкладає значні кошти, а перш поза все люди – годувальники сімей, батьки і матері дітей. Незадовільний стан охорони праці відображається на економіці держави. Служба охорони праці створюється на підприємствах незалежно з форми власності і видів діяльності задля створення правових, організаційно-технічних, санітарно-гігієнічних, соціально-економічних та лікувально-профілактичних заходів, спрямованих на запобігання нещасних випадків, професійних захворювань та аваріям у процесі праці. Основна мета всіх цих заходів – створити на підприємстві безпечні і здорові умови праці.

В розділі охорона праці дипломного проекту наведені характеристики приміщень, де експлуатуються ВДТ. До розгляду взято робоче місце програміста (оператора ЕОМ).

### **3.2 Визначення небезпечних і шкідливих чинників, що впливають на працівника**

Оператори ПК та програмісти зіштовхуються із впливом таких фізично небезпечних та шкідливих виробничих факторів, як підвищений рівень шуму, підвищена температура зовнішнього середовища, недостатня освітленість робочої зони, електричний current і інші. Відповідно на робочому місці програміста повинні існувати створені умови задля високопродуктивної праці.

Перетворення та обробка інформації проводиться поза поміччю ПК. Робота спроможне кваліфікуватися як робота оператором ЕОМ.

					<b>РП 07. 20 000. 00 ДП ПЗ</b>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		61

### 3.3 Розробка заходів із охорони праці

#### 3.3.1 Виробничі приміщення

При плануванні виробничого приміщення врахована санітарна характеристика виробничих процесів, дотримуються норми корисної площі задля працюючих, а разом з цим нормативи площ задля розташування устаткування, що забезпечують безпечну роботу і зручне обслуговування устаткування.

Об'ємно-планувальні рішення будівель і приміщень задля роботи із ВДТ містять відповідати вимогам ДСанПіН 3.3.2.007-98. Розміщення робочих місць із ВДТ ЕОМ та ПЕОМ в підвальних приміщеннях, на цокольних поверхах заборонено. Площа на одне робоче місце становить не менше ніж 6,0 м<sup>2</sup>, а об'єм – не менше ніж 20,0м<sup>3</sup>.

Виробничі приміщення повинні обладнуватися шафами задля зберігання документів, полицями, стелажми, тумбами тощо, із урахуванням вимог до площі приміщення.

В приміщеннях із ВДТ слід щоденно робити вологе прибирання. Приміщення повинні існувати оснащені аптечками першої медичної допомоги.

#### 3.3.2 Мікроклімат робочої зони працівників, вентиляція

В виробничих приміщеннях на робочих місцях із ВДТ містять забезпечуватись оптимальні значення значень мікроклімату: температури, відносної вологості й рухливості повітря ( ДСанПіН 3.3.2.007-98).

Таблиця 3.1. Норми мікроклімату задля приміщень із ВДТ ЕОМ і ПЕМ

Пора року	Категорія робіт	Температура повітря, С, не більш	Відносна вологість повітря %	Швидкість руху повітря, м/с
Холодна	Легка-1а	22-24	40-60	0,1
	Легка-1б	21-23	40-60	0,1
Тепла	Легка-1а	23-25	40-60	0,1
	Легка-1б	22-24	40-60	0,1

Рівні позитивних та негативних іонів в повітрі приміщень із ВДТ містять відповідати санітарно-гігієнічним нормам № 2152-80.

Таблиця 3.2.

Рівні	Число іонів у 1 см <sup>3</sup> повітря	Число іонів у 1 см <sup>3</sup> повітря
	n+	n-
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Максимально допустимі	50000	50000

### 3.3.3 Освітлення робочого місця, шум, вібрація

Штучне освітлення у приміщеннях із робочими місцями, обладнаними ВДТ може здійснюватись системою загального рівномірного освітлення. В виробничих і адміністративних приміщеннях, в разі переважної роботи із документами, допускається застосування системи комбінованого освітлення – крім системи загального освітлення додатково встановлюються світильники місцевого освітлення.

Значення освітленості на поверхні робочого столу у зоні розміщення документів може становити 300-500лк.

Як джерела світла задля штучного освітлення містять застосовуватись переважно люмінесцентні лампи типу ЛД. Допускається застосування ламп розжарювання в світильниках місцевого освітлення.

### 3.3.4 Організація робочого місця оператора ПК

Робочі місця слід так розташовувати відносно світових прорізів, щоб природне світло падало збоку, переважно зліва. При розміщенні робочих столів із ВДТ слід дотримуватися таких відстаней: між бічними поверхнями ВДТ -1,2м; з тильної поверхні одного ВДТ до екрану іншого – 2,5м.

Екран ВДТ може розташовуватися на оптимальній відстані з очей оператора, що становить 600...700 мм, проте не ближче ніж поза 600 мм із урахуванням розміру літерно-цифрових знаків та символів.

Клавіатуру розташовують на поверхні столу на відстані 100...300 мм з краю, зверненого до працюючого. В конструкції клавіатури може передбачатися

опорний пристрій, який дає змогу змінювати кут нахилу поверхні клавіатури в межах 5...15°. При оснащенні робочого місця лазерним принтером параметри лазерного випромінювання повинні відповідати вимогам СанПіН № 5804-91.

ЕОМ ВДТ та ПК , інше устаткування , електропроводи і кабелі поза виконанням та ступенем захисту містять відповідати класу зони поза НПАОП 40.1-1.01-97, мати апаратуру захисту з струма короткого замикання і інших аварійних режимів. В приміщеннях, де одночасно експлуатується понад п'ять ЕОМ встановлюється аварійний резервний вимикач, який спроможне повністю вимкнути електричне живлення приміщення, крім освітлення. Не допускається підключати ЕОМ із ВДТ та ПК до звичайної двопровідної електромережі, у відповідно числі – із використанням перехідних приладів

### **3.3.5 Електробезпека**

Це система організаційних та технічних заходів і засобів, що забезпечують захист людей з шкідливої та небезпечної дії електричного струма, електричної дуги, електричного поля та статичної електрики.

Основні технічні засоби та заходи забезпечення електробезпеки при нормальному режимі роботи електроустановок включають:

- ізоляцію струмовідних частин;
- недоступність струмовідних частин;
- блоківки безпеки;
- засоби орієнтації у електроустановках;
- створення електроустановок, ізольованих з землі;
- захисне розділення електричних мереж;
- компенсацію ємнісних струмів замикання на землю;
- вирівнювання потенціалів.

Із метою підвищення рівня безпеки, залежно з призначення, умов експлуатації та конструкції, у електроустановках застосовується одночасно більшість із перерахованих технічних засобів та заходів. Особа відповідальна поза електрогосподарство призначається із числа працівників, котрі містять не нижче IV групи із електробезпеки і відповідний стаж роботи задля

					<i>РП 07. 20 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

обслуговування електроустановок несе персональну відповідальність поза допущення працівника застосовувати у роботі електричну енергію

### 3.4 Пожежна безпека

Пожежна небезпека – це можливість виникнення і розвитку пожежі у будь-якій речовині, процесі, стані.. Коли людина перебуває у зоні впливу пожежі, то вона спроможне потрапити під дію наступних небезпечних і шкідливих факторів: токсичні продукти згорання, вогонь, підвищена температура середовища, дим, недостатність кисню, руйнування будівельних конструкцій, вибухи, паніка.

Усі працівники повинні вміти користуватись наявними вогнегасниками, іншими первинними засобами пожежогасіння, знати місце їх знаходження.

До первинних засобів пожежогасіння відносяться:

- вогнегасники;
- пожежний інвентар (покривала із негорючого теплоізоляційного полотна, грубововняної тканини чи повсті;
- ящики із піском;
- бочки із водою, пожежні відра, совкові лопати) і пожежний інструмент (гаки, ломи, сокири тощо).

Пожежні щити (стенди) встановлюються на території об'єкта із обчислення один щит (стенд) на площу 5000 м<sup>2</sup>. Ящики задля піску повинні мати місткість 0,5, 1,0 чи 3,0 м<sup>3</sup> і існувати укомплектованими совковою лопатою. Вибір типу і визначення необхідної кількості вогнегасників здійснюється відповідно до Типових норм належності вогнегасників, затверджених наказом Міністерства України із питань надзвичайних ситуацій і в справах захисту населення з наслідків Чорнобильської катастрофи з 02.04.2004 № 151 і зареєстрованих в Міністерстві юстиції України 29.04.2004 поза № 554/9153. Евакуаційні шляхи та виходи повинні втримуватися відповідно до НАПБ А.01.001 2004, існувати вільними, нічим не зашарашуватися та в разі виникнення пожежі забезпечувати безпеку під час евакуації всіх людей, котрі перебувають в приміщеннях будівель і споруд.

					<b>РП 07. 20 000. 00 ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

## ВИСНОВКИ

Протягом дипломного проектування виконано визначення методик обчислення класичних схем випрямних приладів, котрі входять в склад блоків живлення сучасної комп'ютерної техніки, а разом з цим розроблено алгоритми та здійснено їх програмну реалізацію. На основі побудованих блок-схем алгоритмів створено прикладний програмний застосунок під ОС сімейства MS Windows, який надає поза початковими даними розраховувати необхідні параметри півпровідникових випрямників. Програмний застосунок створено в інтегрованому середовищі візуальної розроблювання Embarcadero RAD Studio мовою програмування C++.

В роботі докладно розглянуто найбільш популярні в розробників електронної апаратури види випрямних приладів та представлені методики їх обчислення. Визначено, що задля випрямних приладів основними розраховуваними параметрами є параметри трансформатору, діодів, фільтра, внутрішній супротив випрямника. Створені методики обчислення дозволили здійснити їх подальшу програмну реалізацію. У окремих випадках використовувалися відповідні наближені формули.

Наведені в роботі фрагменти лістингу програми мовою C++ супроводжені докладними коментарями, що дозволить в майбутньому модифікувати та удосконалити застосунок. Виконано тестові випробування, котрі показали, що розрахунки, проведені поза поміччю розробленого програмного забезпечення містять достатню задля інженерної практики ступінь точності. Максимальне відхилення склало 10% при обчислення внутрішнього опору випрямника. Задля приблизних розрахунків це допустиме значення. По решті значень погрішність не перевищила 5%. Із урахуванням того, що аналогічний обчислення класичним методом (уручну) спроможне віднімати дуже багато часу, доцільність створення такого програмного застосунку є очевидною, особливо, якщо треба виконати пошук найкращого варіанту схеми побудови блоку живлення електронної апаратури.

					РП 07. 20 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

## ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Дейбук В.Г. Віртуальний електронний практикум: Навчальний посібник – Чернівці: Чернівецький нац. ун-т, 2021. – 188 с.
2. Трофименко О.Г. С++. Алгоритмізація та програмування: підручник / О.Г. Трофименко, Ю.В. Прокоп, Н.І. Логінова, О.В. Задерейко. 2-ге вид. перероб. і доповн. – Одеса : Фенікс, 2019. – 477 с.
3. Азаров О. Д., Гарнага В. А., Клятченко Я. М., Тарасенко В. П. Комп'ютерна схемотехніка: підручник. – Вінниця: ВНТУ, 2018. – 230 с.
4. Архангельский А.Я. Программирование в С++Builder, 7-е изд. – М.: ООО Бином-Пресс, 2010 г. – 1230с., ил.
5. Нікулін В.С. Перетворювальні пристрої, ведені мережею: Конспект лекцій. –Харків: УкрДАЗТ, 2008. – Ч.4. – 85 с.
6. Мосіюк О.О., Федорчук А.Л. Операційні системи та системне програмування: навчально-методичний посібник. Житомир: Вид-во ЖДУ ім. Івана Франка, 2022. – 76 с.
7. Stroustrup B. A Tour of C++ (Second Edition). – Addison-Wesley, 2018. – 240 p. – ISBN 978-0-13-499783-4.
8. Матвієнко М.П., Розен В. П. Комп'ютерна схемотехніка: навчальний посібник. – К.: Видавництво Ліра-К, 2020. – 192 с.
9. Гусев Б.С. Комп'ютерна схемотехніка [навчальний посібник] // – К.: НУБіП України, 2022. – 264с.
10. Каганюк О.К. Комп'ютерна схемотехніка: Навчальний посібник. – Луцьк: РРВ Луцького НТУ, 2016. – 236 с.
11. Кожарский Г.В., Орехов В.И. Методы автоматизированного проектирования источников вторичного электропитания. – М.: Радио и связь, 1985. – 184 с.
12. Источники вторичного электропитания/Под ред. Ю.И.Конева. – 2-ое изд., перераб. и доп. – М.: Радио и связь, 1990. – 280 с.
13. Бібліотека MSDN [Електронний ресурс]. – Режим доступу: URL <http://msdn.microsoft.com/ru-ru/library/default.aspx>.

					<b>РП 07. 20 000. 00 ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

## ДОДАТОК А. Код функцій Rectifier\_GetInputValues, btnInductiveClick, btnCapacityClick, btnRFClick мовою C++

```
bool __fastcall TApp::Rectifier_GetInputValues( double& U, int& F, double& E, double& A,
double& P )
{
    U = editU->Text.ToDouble();
    F = editF->Text.ToInt();
    A = editA->Text.ToDouble();
    E = editE->Text.ToDouble();
    P = editP->Text.ToDouble();
    bool result = U > 0 && F > 0 && A > 0 && E > 0 && P > 0;
    if( !result )
        ErrMess( "Вхідні дані мають бути\позитивними числами!" );
    else
    {
        if( E < 3 )
        {
            if( IDNO == Application->MessageBox( "Вихідна напруга < 3 В.\п
Точність розрахунку знижується.\п\пПродовжити?", "УВАГА", MB_YESNO)
                )
                result = false;
        }
        if( result && E*A > 500 )
        {
            if( IDNO == Application->MessageBox( "Вихідна потужність перевищує 500 Вт.\п
Точність розрахунку знижується.\п\пПродовжити?", "УВАГА", MB_YESNO)
                )
                result = false;
        }
        if( result && F > 5000 )
        {
            if( IDNO == Application->MessageBox( "Частота > 5 кГц.\п
Точність розрахунку знижується.\п\пПродовжити?", "УВАГА", MB_YESNO)
                )
                result = false;
        }
    }
    return result;
}
//-----

void __fastcall TApp::btnInductiveClick(TObject *Sender)
{
    double T[6][15] = {
        { 0, .5, 1, 2, 1, 2.828, 1.11, .707, .707, 1.34, .67, 1.414, 7, .0055, 1 },
        { 0, .5, 1, 2, 2, 1.414, 1.11, 1, .707, 1.11, .67, 1.414, 5.2, .0064, 1 },
        { 0, .333, 1, 3, 1, 2.449, .855, .58, .58, 1.35, .25, 1.414, 6.6, .0033, .47 },
        { 0, .333, 2, 6, 2, 2.449, .43, .82, .58, 1.045, .057, 2.449, 2.5, .001, .82 },
```

```

        { 0, .333, .667, 6, 2, 1.449, .74, .41, .58, 1.045, .057, 1.414, 7.6, .0033, .47 },
        { 0, .167, 1, 6, 1, 2.828, .74, .41, .41, 1.43, .057, 1.414, 6, .0033, .47 }
    };
    int id = comboScheme->ItemIndex;
    double *K = T[id];
    double U, E, A, P;
    int F;
    if( !Rectifier_GetInputValues(U, F, E, A, P) )
        return;
    if( (id == 3 || id == 4) && P < 0.007 )
        if( IDNO == Application->MessageBox( "Коефіцієнт пульсацій м.б. > 0.07\n\nПродовжити?",
            "УВАГА", MB_YESNO) )
            return;
    double B = 1.2 - .4*sin(.003*E*A);
    double RT = K[12] * E/F/B/A * sqrt(sqrt(F*B/E/A));
    double XS = K[13] * E/F/A/B/sqrt(sqrt(F*B/E/A));
    double Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10, Y11, Y12, Y13, Y14, Y15, Y16, Y17;
    Y1 = E + K[2]*A*RT + K[3]*F*XS*A + .6*K[4] + .05*E;
    Y2 = Y1*K[6];
    Y3 = K[14]*Y2*A/U;
    Y4 = K[7]*A;
    Y5 = K[9]*Y1*A;
    Y6 = K[1]*A;
    Y7 = K[8]*A;
    Y8 = A;
    Y9 = K[5]*Y2;
    Y10 = .6*Y6;
    Y11 = .637*E/(pow(K[3],2) - 1)/K[3]/F/A;
    double Q = K[10]/P;
    Y12 = 25330*Q/pow(K[3],2)/pow(F,2)/Y11;
    Y13 = K[11]*Y2;
    Y14 = E/(pow(K[3],2)-1)/K[3]/M_PI/F/Y11;
    Y15 = (Y1-E)/A;
    Y16 = E + (A-Y14)*Y15;
    Y17 = K[11]*Y2;
    FillRectifierInputText();
    AnsiString t;
    char buff[128];
    t = "\r\n ВИМОГИ ДО ПАРАМЕТРІВ ТРАНСФОРМАТОРУ:\r\n";
    sprintf( buff, " %-45s%6.2fr\n", "Габаритна потужність, Вт", Y5 ); t += buff;
    sprintf( buff, " %-45s%6.2fr\n", "Напруга холостого ходу, В", Y2 ); t += buff;
    sprintf( buff, " %-45s%6.3fr\n", "Ефективний струм вторинної обмотки, А", Y4 ); t += buff;
    sprintf( buff, " %-45s%6.3fr\n", "Ефективний струм первинної обмотки, А", Y3 ); t += buff;
    t += "\r\n ВИМОГИ ДО ПАРАМЕТРІВ ДЮДІВ:\r\n";
    sprintf( buff, " %-45s%6.2fr\n", "Зворотна напруга, В", Y9 ); t += buff;
    sprintf( buff, " %-45s%6.2fr\n", "Середній прямий струм, А", Y6 ); t += buff;
    sprintf( buff, " %-45s%6.2fr\n", "Імпульсний прямий струм, А", Y8 ); t += buff;
    sprintf( buff, " %-45s%6.3fr\n\r\n", "Потужність, що виділяється, Вт", Y10 ); t += buff;
    sprintf( buff, " %-45s%6.2fr\n", "Мінімальна індуктивність дроселя, мГн", Y11*1000 );
    t += buff;

```

```

sprintf( buf, " %-45s%6.2f\r\n", "Випрямлена напруга при XX. i C = 0, B", Y1 ); t += buf;
sprintf( buf, " %-45s%6.2f\r\n", "Випрямлена напруга при XX. i C > 0, B", Y17 ); t += buf;
sprintf( buf, " %-45s%6.0f\r\n", "Смність конденсатору, мкФ", Y12 ); t += buf;
sprintf( buf, " %-45s%6.2f\r\n", "Робоча напруга конденсатору >=, B", Y17 ); t += buf;
sprintf( buf, " %-45s%6.2f\r\n", "Внутрішній опір випрямляча, Ом", Y15 ); t += buf;
t += "\r\n КРИТИЧНА ТОЧКА НАВАНТАЖУВАЛЬНОЇ ХАРАКТЕРИСТИКИ:\r\n";
sprintf( buf, " %-45s%6.2f\r\n", "Струм, А", Y14 ); t += buf;
sprintf( buf, " %-45s%6.2f\r\n", "Напруга, В", Y16 ); t += buf;
strRectifierResult = t;
checkInputRectifierClick(0);
}
//-----

void __fastcall TApp::btnCapacityClick(TObject *Sender)
{
    int id = comboSchemeC->ItemIndex;
    double T[7][11] = {
        { 0, 1, 2.3, 1, 1, 1, 1, 2, 1, 2.828, 1.84 },
        { 0, .5, 4.7, 2, 1, .5, 1, 1.8, 1, 2.828, 1.63 },
        { 0, .5, 3.5, 2, 1, .707, .707, 1.5, 2, 1.414, 1.63 },
        { 0, 1, .9, 1, .5, 1.414, .707, 1.5, 1, 2.828, 2.95 },
        { 0, .333, 6.9, 3, 1, .333, 1, 2.15, 1, 2.828, .72 },
        { 0, .333, 4.5, 6, .578, .333, .707, 1.25, 2, 2.44, .92 },
        { 0, .333, 13.5, 6, 1, .193, 1.23, 1.25, 2, 1.414, .53 }
    };
    double *K = T[id];
    double U, E, A, P;
    int F;
    if( !Rectifier_GetInputValues(U, F, E, A, P) )
        return;
    if( (id == 5 || id == 6) && P > 0.007 )
        if( IDNO == Application->MessageBox( "Коефіцієнт пульсацій м.б. < 0.07\n\nПродовжити?",
            "УВАГА", MB_YESNO) )
            return;
    double X = 1.2 - .4*sin(.003*E*A);
    X = K[2] * E/F/X/A * sqrt(sqrt(F*X/E/A));
    double Y = K[1]*A;
    X = K[8]*.2/Y + X;
    double V = M_PI*X*A/(K[3]*E);
    double B, C, D;
    double TB = M_PI/2, TH = 0, TT;
    while( true )
    {
        TT = (TB + TH)/2;
        B = tan(TT) - TT - V;
        C = .001*V;
        if( C - fabs(B) >= 0 || !B )
            break;
        else
            if( B < 0 ) TH = TT;
    }
}

```

```

        else TB = TT;
    }
    D = sqrt( M_PI*( TT*(1+.5*cos(2*TT))- .75*sin(2*TT) ) ) / (sin(TT) - TT*cos(TT));
    B = M_PI*(1-cos(TT))/(sin(TT)-TT*cos(TT));
    if( K[3] > 1)
    {
        C = 101E3*( sin(K[3]*TT)*cos(TT) - K[3]*cos(K[3]*TT)*sin(TT) );
        C = C/( K[3]*(pow(K[3],2)-1)*cos(TT) );
    }
    else
    {
        C = 25330*(2*TT - sin(2*TT))*cos(TT);
    }
    double Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10, Y11, Y12;
    Y1 = E*K[4]/1.41/cos(TT);
    Y2 = Y1*K[9];
    Y3 = K[10]*A*Y1/U;
    Y4 = K[5]*A*D;
    Y5 = K[6]*Y4;
    Y6 = B*A/K[3];
    Y7 = C/(X*P*F);
    Y8 = pow(Y5,2)*.2/Y;
    Y9 = K[7]*E*A;
    Y10 = 1.41*Y1*cos(TT/2)/K[4];
    Y11 = .45*K[3]*Y1*(sin(TT/2)-TT/2*cos(TT/2))/K[4]/X;
    Y12 = (Y10 - E) / (A - Y11);
    FillRectifierInputText();
    AnsiString t;
    char buff[128];
    t = "\r\n ВИМОГИ ДО ПАРАМЕТРІВ ТРАНСФОРМАТОРУ:\r\n";
    sprintf( buff, " %-45s%6.1fr\n", "Габаритна потужність, Вт", Y9); t += buff;
    sprintf( buff, " %-45s%6.1fr\n", "ЕРС вторинної обмотки, В", Y1); t += buff;
    sprintf( buff, " %-45s%6.2fr\n", "Ефективний струм вторинної обмотки, А", Y4); t += buff;
    sprintf( buff, " %-45s%6.3fr\n", "Ефективний струм первинної обмотки, А", Y3); t += buff;
    t += "\r\n ВИМОГИ ДО ПАРАМЕТРІВ ДІОДІВ:\r\n";
    sprintf( buff, " %-45s%6.1fr\n", "Зворотна напруга, В", Y2); t += buff;
    sprintf( buff, " %-45s%6.2fr\n", "Середній прямий струм, А", Y); t += buff;
    sprintf( buff, " %-45s%6.2fr\n", "Імпульсний прямий струм, А", Y6); t += buff;
    sprintf( buff, " %-45s%6.2fr\n", "Потужність, що виділяється, Вт", Y8); t += buff;
    t += "\r\n ПРОМІЖНА ТОЧКА НАВАНТАЖУВАЛЬНОЇ ХАРАКТЕРИСТИКИ:\r\n";
    sprintf( buff, " %-45s%6.3fr\n", "Струм, А", Y11); t += buff;
    sprintf( buff, " %-45s%6.1fr\n\r\n", "Напруга, В", Y10); t += buff;
    sprintf( buff, " %-45s%6.0fr\n", "Емність конденсатора, мкФ", Y7); t += buff;
    sprintf( buff, " %-45s%6.2fr\n", "Внутрішній опір випрямляча, Ом", Y12); t += buff;
    strRectifierResult = t;
    checkInputRectifierClick(0);
}

```

//-----

```

void __fastcall TApp::btnRFClick(TObject *Sender)
{
    double U1, U0, Y0, A;
        int F;
    double B = editSkvazhmost->Text.ToDouble();
    if( !Rectifier_GetInputValues(U1, F, U0, Y0, A) )
        return;
    double U2, U10, Y1, Y2, YI, C, PI, P, R, Y, D;
    double RT;
    double G = sqrt(1.2*U0/(pow(Y0,3)*F));
    int id = comboSchemeRF->ItemIndex;
    if( !comboReaction->ItemIndex )
    {
        int k12 = id + 1;
        int k21 = 2 - id;
        RT = G * (!id ? 5.5: 4.08);
        U2 = 1.01*U0 + RT*Y0 + k12;
        R = k21*(U2-U0)/Y0;
        P = !id ? (1 + sqrt(B+1))*U2*Y0/2: 1.1*U2*Y0;
        Y2 = Y0*sqrt(B/k21);
        U10= k21*U2*B;
        Y1 = 1.1*U2*Y2*sqrt(B)/U1;
        Y = Y2/2;
        YI = Y0*B;
        PI = Y0/2;
        D = .1*B*U0*sin(M_PI/B)/(F*Y0);
        C = 25E5*Y0/(pow(M_PI,3)*A*U0*F);
    }
    else
    {
        if( id == 0 )
        {
            RT = 1.9*G;
            U2 = U0 + 2*Y0*B*(RT + sqrt(2/B)/(2*Y0));
            P = 1.1*U2*Y0*(.7+sqrt((2*B-1)/B)/2);
            Y1 = 1.1*U2*Y0*sqrt((2*B-1)/B)/U1;
            Y2 = Y0*sqrt(2*B);
        }
        else if( id == 1 )
        {
            RT = 3.9*G;
            U2 = U0 + Y0*B*(RT+sqrt(2/B)/Y0);
            P = 1.3*Y0*U2;
            Y1 = 1.1*Y0*U2/U1;
            Y2 = sqrt(B/2)*Y0;
        }
        else if( id == 2 )
        {
            RT = 2.9*G;
            U2 = U0 + 2*Y0*B*(RT+2*sqrt(2/B)/Y0);

```

```

    P = 1.1*Y0*U2;
    Y1 = 1.1*Y0*U2/U1;
    Y2 = sqrt(B)/Y0;
}
else // if( id == 3 )
{
    RT = .75*G;
    U2 = U0/2 + 2*Y0*(RT+Y0*sqrt(1/B)/2);
    P = 2.2*Y0*U2;
    Y1 = 2.2*Y0*U2/U1;
    Y2 = 2*sqrt(B)*Y0;
    PI = .7*Y0*sqrt(2*B);
}
U10 = 2*U2;
if( id == 0 || id == 3 )
{
    Y = Y0;
    YI = 2*B*Y0;
    PI = Y0*sqrt(2*B);
    C = 2E6*B*Y0*sin(M_PI/2/B)/(10*U0*A*F);
}
else // if( id == 1 || id == 2 )
{
    Y = Y0/2;
    YI = B*Y0;
    PI = Y0*sqrt(B/2);
    C = 1E5*B*Y0*sin(M_PI/B)/(2*U0*A*F);
}
}
FillRectifierInputText();
AnsiString t;
char buf[128];
t = "\r\n ВИМОГИ ДО ПАРАМЕТРІВ ТРАНСФОРМАТОРУ:\r\n";
sprintf( buf, " %-45s%6.2f\r\n", "Габаритна потужність, Вт", P ); t += buf;
sprintf( buf, " %-45s%6.2f\r\n", "ЕРС вторинної обмотки, В", U2 ); t += buf;
sprintf( buf, " %-45s%6.2f\r\n", "Ефективний струм вторинної обмотки, А", Y2 ); t += buf;
sprintf( buf, " %-45s%6.2f\r\n", "Ефективний струм первинної обмотки, А", Y1 ); t += buf;
t += "\r\n ВИМОГИ ДО ПАРАМЕТРІВ ДІОДІВ:\r\n";
sprintf( buf, " %-45s%6.2f\r\n", "Зворотна напруга, В", U10 ); t += buf;
sprintf( buf, " %-45s%6.2f\r\n", "Середній прямий струм, А", Y ); t += buf;
sprintf( buf, " %-45s%6.2f\r\n", "Імпульсний прямий струм, А", YI ); t += buf;
sprintf( buf, " %-45s%6.2f\r\n\r\n", "Потужність, що виділяється, Вт", PI ); t += buf;
sprintf( buf, " %-45s%6.0f\r\n", "Ємність конденсатора, мкФ", C ); t += buf;
if( !comboReaction->ItemIndex )
{
    sprintf( buf, " %-45s%6.4f\r\n\r\n", "Індуктивність дроселя, Гн", D ); t += buf;
    sprintf( buf, " %-45s%6.2f\r\n", "Внутрішній опір випрямляча, Ом", R ); t += buf;
}
strRectifierResult = t;
checkInputRectifierClick(0); }

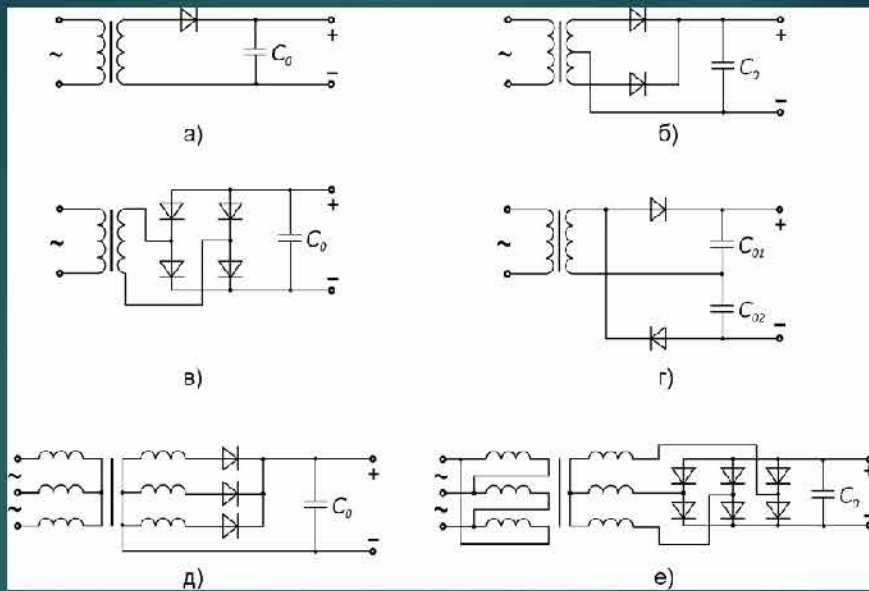
```

# Розробка програмного застосунку для розрахунку схем напівпровідникових випрямлячів

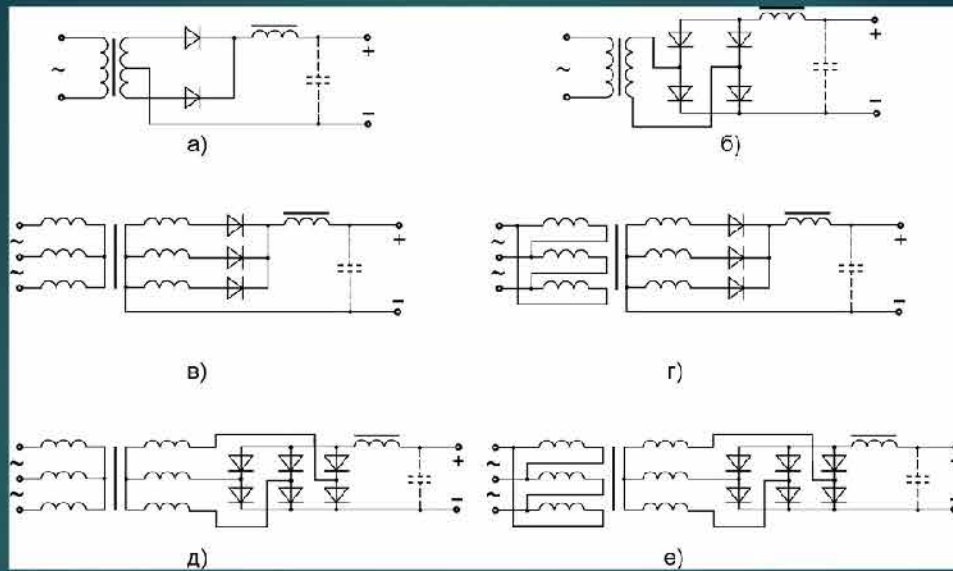
Тарахтій Владислав, гр. 4РП-07  
ВСП «ОФК ОНУ»



*Структурна схема випрямного пристрою*



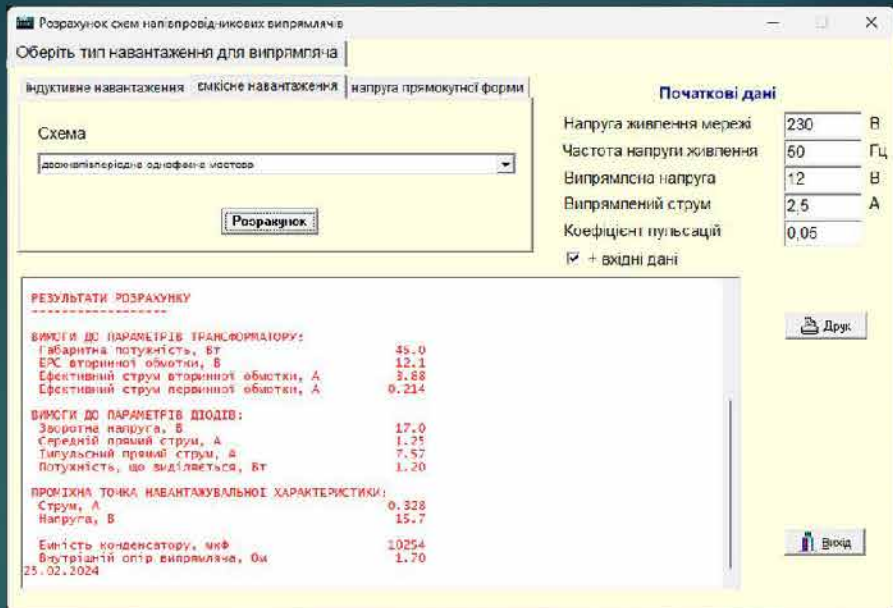
Схеми випрямлячів для однофазної та трифазної мережі, що працюють на ємність



Схеми випрямлячів для однофазної та трифазної мережі, що працюють на індуктивність



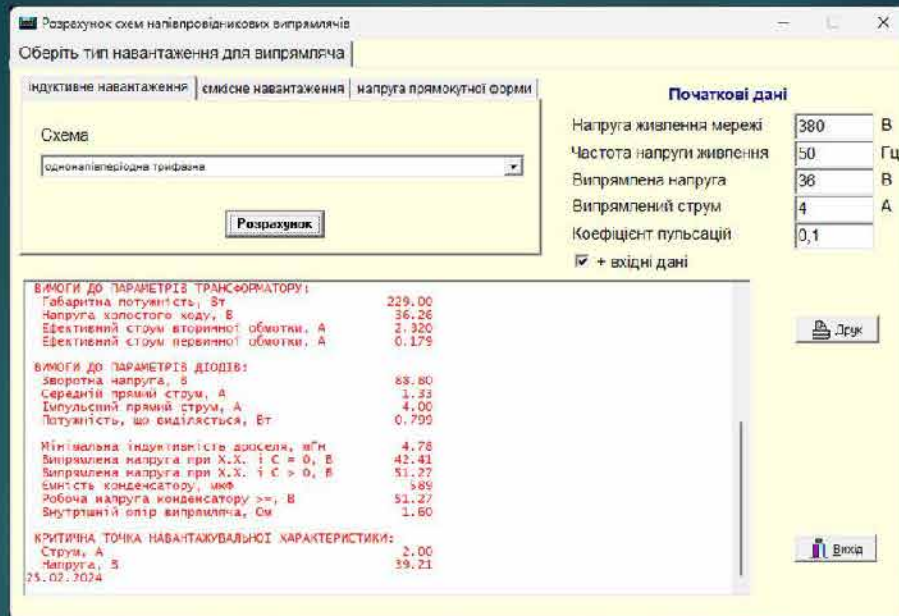




Візуальний інтерфейс застосунку для розрахунку схем напівпровідникових випрямлячів з ємкісним навантаженням

Параметр	Ручний розрахунок	Програмний	Відхилення %
Вимоги до параметрів трансформатора:			
Габаритна потужність, Вт	45.1	45.0	-0.22
ЕРС вторинної обмотки, В	11.8	12.1	2.54
Ефективний струм вторинної обмотки, А	3.93	3.88	-1.27
Ефективний струм первинної обмотки, А	0.22	0.214	-2.73
Вимоги до параметрів діодів:			
Зворотна напруга, В	16.9	17.0	0.6
Середній прямий струм, А	1.23	1.25	1.63
Імпульсний прямий струм, А	7.6	7.57	0.39
Потужність, що виділяється, Вт	1.21	1.20	-3.23
Проміжна точка характеристики навантаження:			
Струм, А	0.32	0.328	2.5
Напруга, В	15.9	15.7	-1.26
Ємність конденсатора, мкФ	10457	10254	-1.94
Внутрішній опір випрямляча, Ом	1.68	1.70	1.19

Порівняння результатів ручного та програмного розрахунків для двофазної періодично-однофазної мостової схеми з ємкісним навантаженням



Візуальний інтерфейс застосунку для розрахунку схем напівпровідникових випрямлячів з індуктивним навантаженням

Параметр	Ручний розрахунок	Програмний	Відхилення, %
Вимоги до параметрів трансформатора:			
Габаритна потужність, Вт	231.2	229.00	-0.95
Напруга холостого ходу, В	35.69	36.26	1.6
Ефективний струм вторинної обмотки, А	2.35	2.32	-1.28
Ефективний струм первинної обмотки, А	0.181	0.179	-1.1
Вимоги до параметрів діодів:			
Зворотна напруга, В	88.92	88.80	-0.14
Середній прями́й струм, А	1.34	1.33	0.75
Імпульсний прями́й струм, А	4.1	4.0	-2.44
Потужність, що виділяється, Вт	0.8	0.799	0.125
Мінімальна індуктивність дроселя, мГн	4.75	4.78	0.63
Випрямлена напруга при X.X. і C = 0, В	42.5	42.41	-0.21
Випрямлена напруга при X.X. і C > 0, В	51.32	51.27	-0.1
Ємність конденсатора, мкФ	585	589	0.68
Робоча напруга конденсатора >, В	51.3	51.27	0.05
Внутрішній опір випрямляча, Ом	1.45	1.60	10.34
Критична точка характеристики навантаження:			
Струм, А	1.9	2.00	5.26
Напруга, В	39.6	39.21	-0.98

Порівняння результатів ручного та програмного розрахунків для однонапівперіодної трифазної схеми з індуктивним навантаженням

## Висновки

Тестові випробування показали, що розрахунки, проведені за допомогою розробленого програмного застосунку мають достатню для інженерної практики ступінь точності. Максимальне відхилення склало 10% при розрахунку внутрішнього опору випрямляча. Для приблизних розрахунків це допустиме значення. По решті параметрів погрішність не перевищила 5%. З урахуванням того, що аналогічний розрахунок класичним методом (уручну) може віднімати дуже багато часу, доцільність створення такого програмного застосунку є очевидною, особливо, якщо необхідно виконати пошук найкращого варіанту схеми побудови блоку живлення електронної апаратури.

**ВІДГУК**

керівника на дипломний проект здобувача (здобувачки) освіти  
відділення комп'ютерних систем

*Тарахтію Владиславу Олександровичу*

(прізвище, ім'я та по батькові)

Спеціальність: *121 «Інженерія програмного забезпечення»*

Освітня програма: *«Розробка програмного забезпечення»*

Тема дипломного проекту: *Розробка програмного застосунку для розрахунку схем напівпровідникових випрямлячів*

**ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ**

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) *Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка до дипломного проекту містить 80 сторінок. У пояснювальній записці описано етапи розробки програмного забезпечення для розрахунку схем напівпровідникових випрямлячів блоків живлення у ICP Embarcadero RAD Studio C++Builder. Графічна частина складається з окремих слайдів, оформлених у вигляді презентації, передбачених технічним завданням. Якість виконання пояснювальної записки та слайдів добра.*

б) самостійність роботи над проектом: *Протягом виконання дипломного проекту здобувач освіти Тарахтій Владислав поступово та послідовно виконував всі етапи, проявив ініціативу в створенні загальної концепції та реалізації роботи. Всі роботи здобувач освіти виконував самостійно, з оглядом на рекомендації керівника.*

в) теоретична підготовка випускника (випускниці): *Здобувач освіти Тарахтій Владислав під час роботи над дипломним проектом вивчив достатньо багато літературних та інтернет-джерел за даною тематикою.*

*Вважаю, що теоретична підготовка дипломника достатня і він готовий до захисту проекту.*

г) вміння розв'язувати виробничі та конструкторські питання Під час виконання дипломного проекту здобувач освіти Тарахтій Владислав показав вміння організовано працювати над поставленим завданням, застосовувати знання у галузі програмування, електроніки, математики, розробляти, встановлювати та налаштовувати спеціалізоване програмне забезпечення, оформлювати слайди та складати презентації, користуючись сучасними комп'ютерними програмними засобами, такими як Embarcadero RAD Studio, Microsoft PowerPoint, Microsoft Visio та ін.

Оцінка розрахункової частини Відмінно  
Оцінка графічної частини Відмінно  
Загальна оцінка Відмінно

Прізвище, ім'я, по батькові керівника дипломного проекту Скорняков В'ячеслав Сергійович

Місце роботи і посада керівника дипломного проекту ВСП «Одеський технічний фаховий коледж ОНТУ», викладач спецдисциплін циклової комісії комп'ютерної техніки та програмної інженерії

Підпис 

« 5 » червня 2023 р.

**РЕЦЕНЗІЯ**

на дипломний проект здобувача (здобувачки) освіти  
відділення комп'ютерних систем

*Тарахтій Владислав Олександровича*

(прізвище, ім'я та по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Розробка програмного забезпечення»

Керівник дипломного проекту (роботи) Скорняков В'ячеслав Сергійович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка програмного застосунку для розрахунку схем напівпровідникових випрямлячів

Обсяг розрахунково-пояснювальної записки 80 сторінок

Обсяг графічної (презентаційної) частини 13 аркушів (слайдів)

**ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)**

а) заключення про ступінь відповідності виконаного дипломного проекту завданню

*Представлений на рецензію дипломний проект відповідає затвердженій темі та виконаний відповідно технічному завданню. Дипломний проект присвячений проблемі розрахунку схем напівпровідникових випрямлячів та складається з пояснювальної записки, додатку з програмним кодом та мультимедійної презентації, що містить приклади роботи програми.*

б) характеристика виконання кожного розділу дипломного проекту

*Пояснювальна записка складається з основного розділу (аналізу предметної області, проектування застосунку, реалізації застосунку, тестування застосунку), економічного розділу, розділу охорони праці та додатків. Перелічені розділи поетапно охоплюють розробку, виконані докладно та обґрунтовано. Розділ охорони праці містить загальну інформацію та вимоги до техніки безпеки оператора КТ. Економічний розділ проекту містить розрахунок витрат на НДР та реалізацію проекту.*

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту

*Графічна частина складається з 13 слайдів мультимедійної презентації, виконаної у програмному продукті MS PowerPoint, які містять ілюстративні схеми, скріншоти роботи програмного застосунку, передбачені технічним завданням. Пояснювальна записка виконана акуратно та у відповідності до норм. Якість виконання графічної частини проекту та пояснювальної записки д, розробку виконано у повному обсязі.*

г) перелік позитивних якостей дипломного проекту Реалізований алгоритм розрахунку схем напівпровідникових випрямлячів дозволяє достатньо точно виконувати розрахунки багатьох важливих параметрів за спрощеними методиками, передбачає перевірку некоректних даних.

У роботі наведені докладні блок-схеми алгоритмів розрахунку випрямлячів, виконано порівняння ручного та автоматичного розрахунку.

д) основні недоліки дипломного проекту \_\_\_\_\_

1. Існують більш точні методи розрахунку параметрів випрямлячів.

2. У застосунку бажано було б також передбачити англomовний інтерфейс.

Оцінка розрахункової частини \_\_\_\_\_ Відмінно

Оцінка графічної частини \_\_\_\_\_ Відмінно

Загальна оцінка \_\_\_\_\_ Відмінно

Прізвище, ім'я, по батькові рецензента \_\_\_\_\_ к.т.н. Кіреєв Ігор Анатолійович

Місце роботи і посада рецензента \_\_\_\_\_ Державний університет інтелектуальних технологій і зв'язку, доцент каф. інформаційної безпеки та передачі даних

Підпис: \_\_\_\_\_  
  
« 14 » 06 2024 р.

Ім'я користувача:  
Катерина Григоріївна Краснокутська

ID перевірки:  
1016245784

Дата перевірки:  
12.05.2024 21:51:23 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
12.05.2024 22:11:37 EEST

ID користувача:  
100011688

Назва документа: 4РП-07\_Владислав\_Тарахтій

Кількість сторінок: 60 Кількість слів: 13544 Кількість символів: 94154 Розмір файлу: 977.11 KB ID файлу: 1016030626

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

**28.3%**  
**Схожість**

Найбільша схожість: 9.94% з Інтернет-джерелом ([https://stud.wiki/physics/3c0b65635b3bc78b4d53a89521206d37\\_0.html](https://stud.wiki/physics/3c0b65635b3bc78b4d53a89521206d37_0.html))

28.3% Джерела з Інтернету 885

Сторінка 62

Не знайдено джерел з Бібліотеки

**0% Цитат**

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

**0%**  
**Вилучень**

Немає вилучених джерел

**Модифікації**

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 103

Підозріле форматування 18 сторінок

**ДОЗВІЛ  
НА РОЗМІЩЕННЯ  
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ  
(ДИПЛОМНОГО ПРОЕКТУ)  
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

***Тарахтій Владислав Олександрович,***

здобувач освіти гр. 4РП-07, та

***Скорняков В'ячеслав Сергійович,***

керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

***«Розробка програмного застосунку для розрахунку схем напівпровідникових випрямлячів» (автор роботи – Тарахтій В.О., керівник роботи – Скорняков В.С.)***

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2024 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець

  
\_\_\_\_\_

/ Тарахтій В.О. /

Керівник

  
\_\_\_\_\_

/ Скорняков В.С. /

«10» червня 2024 р.