



Апаратно-орієнтоване програмування

Робоча навчальна програма (силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>F Інформаційні технології</i>
Спеціальність	<i>F2 (121) Інженерія програмного забезпечення; F7 (123) Комп'ютерна інженерія, F6 (126) Інформаційні системи та технології.</i>
Освітня програма	<i>Комп'ютерні системи та мережі Інженерія програмного забезпечення комп'ютерних систем Інтегровані інформаційні системи Інформаційні управляючі системи та технології Інформаційне забезпечення робототехнічних систем Інженерія програмного забезпечення інформаційних систем</i>
Статус дисципліни	<i>Вибіркова</i>
Форма навчання	<i>очна</i>
Рік підготовки, семестр	<i>4 рік (8 семестр)</i>
Обсяг дисципліни	<i>4 кредитів (ECTS). Виділення часу -120 годин, включаючи лекцій 30 годин, лабораторних робіт 14 годин та СРС 76 годин</i>
Семестровий контроль/ контрольні заходи	<i>залік/ лабораторні роботи</i>
Розклад занять	<i>Згідно розкладу на осінній семестр поточного навчального року за адресою http://shedule.kpi.ua</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор і викладач лабораторних занять: д.т.н., проф., Анатолій Михайлович Сергієнко, email a.serhienko@kpi.ua</i>
Розміщення курсу	<i>На платформі дистанційного навчання «Сікорський»: https://classroom.google.com/c/ODUwMzcyNzgxMzc0?cjc=eqr6ib6p Лекційний матеріал: https://bbb.comsys.kpi.ua/rooms/cgx-vn4-bgv-1rg/join</i>

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Викладання основ автоматизованого проектування КС в університеті має такі аспекти:

Академічний аспект. Системи автоматизованого проектування КС - це наука, яка дає прикладні знання. Мета - ознайомити студентів з основними поняттями та методами автоматизованого проектування сучасних складних технічних засобів.

Професійний аспект. Сьогодні множина архітектур мікропроцесорів, реалізованих на кристалі, стрімко розширюється. Програмовані логічні інтегральні схеми (ПЛІС) стають невід'ємним атрибутом технології вбудованих обчислювальних засобів. Зростає потреба в інженерах-проектувальників, які володіють технологіями проектування систем на кристалі (СНК). Від них

вимагається вміння описувати мовою VHDL або Verilog пристроїв із заданою функціональністю, включаючи мікропроцесор, переналагоджувати готові блоки під вимоги проекту, з'єднувати модулі в структуру СНК і програмувати вбудоване матзабезпечення. Мета цього курсу – сформувати навички використання мови VHDL для опису апаратних проектів середньої складності, які реалізуються в ПЛІС. Основу лабораторних робіт становить проектування вузлів мікропроцесорів з використанням мови VHDL і відповідних засобів САПР, включаючи симулятор VHDL і компілятор-синтезатор логічних схем. Студент, який виконує курс лабораторних робіт, вивчає не тільки основи схемотехніки мікропроцесорів, але і сучасну технологію їх проектування для реалізації в ПЛІС.

Інтелектуальні та освітні аспекти. Вивчення систем автоматизованого проектування КС сприяє розвитку когнітивних навичок, що є суттєвим інтелектуальним фактором у процесі створення як апаратного, так і програмного забезпечення.

Вивчення дисципліни підсилює наступні загальні та фахові компетенції:

Для спеціальності F2 (121) Інженерія програмного забезпечення

ЗК01. Здатність до абстрактного мислення, аналізу та синтезу

ЗК02. Здатність застосовувати знання у практичних ситуаціях

ЗК05. Здатність вчитися і оволодівати сучасними знаннями

ЗК07. Здатність працювати в команді

ФК01. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення

ФК03. Здатність розробляти архітектури, модулі та компоненти програмних систем

ФК04. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами

ФК05. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу

ФК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя

ФК14. Здатність до алгоритмічного та логічного мислення

ФК16. Здатність розробляти мобільні системи, вбудовані системи та системи реального часу

ФК17. Здатність розробляти та використовувати програмне забезпечення для високопродуктивних комп'ютерних систем

ФК19. Здатність розробляти та використовувати системи штучного інтелекту

Для спеціальності F7 (123) «Комп'ютерна інженерія»

ЗК 1 Здатність до абстрактного мислення, аналізу та синтезу

ЗК02. Здатність вчитися і оволодівати сучасними знаннями

ЗК3 Здатність застосовувати знання у практичних ситуаціях

ЗК08. Здатність працювати в команді

ФК2 Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення

ФК05. Здатність використовувати засоби і системи автоматизації проектування до розроблення компонентів комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем тощо.

ФК7 Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності

ФК12 Здатність ідентифікувати, класифікувати та описувати роботу програмно-технічних засобів, комп'ютерних та кіберфізичних систем, мереж та їхніх компонентів шляхом використання аналітичних методів і методів моделювання.

ФК16. Здатність проектувати, розробляти, впроваджувати та обслуговувати програмно-апаратне забезпечення для високопродуктивних комп'ютерів та комп'ютерних систем (паралельних, вбудованих, розподілених) та їх складових на сучасній елементній базі (зокрема, ПЛІС) з використанням систем автоматизованого проектування.

Для спеціальності F6 (126) Інформаційні системи та технології

К31. Здатність до абстрактного мислення, аналізу та синтезу

К32. Здатність застосовувати знання у практичних ситуаціях

К37. Здатність розробляти та управляти проектами

КС6. Здатність використовувати засоби і системи автоматизації проектування до розроблення компонентів комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем тощо.

КС13. Здатність проводити обчислювальні експерименти, порівнювати результати експериментальних даних і отриманих рішень

КС15. Здатність аналізувати відомі реалізації компонентів нижнього рівня інформаційних систем з урахуванням вимог технологічних та бізнес-процесів, виконувати синтез контролерів, регуляторів, давачів, виконавчих пристроїв, будувати адекватні моделі нижнього рівня ієрархії інформаційних систем та здійснювати їх інтеграцію

У відповідності до вищезазначеного, підсилені загальні та фахові компетенції дадуть наступні результати навчання:

1. Здатність проектувати, розробляти та реалізовувати системи на базі ПЛІС – розробляти повноцінні апаратно-програмні рішення на базі ПЛІС з використанням засобів САПР і мови VHDL.

2. Вміти користуватися сучасними САПР для розробки електроніки – володіти навичками роботи в середовищі ActiveHDL, ModelSim, Vivado, Quartus або аналогічних, створювати бібліотеки модулів, здійснювати перевірку працездатності та параметрів швидкодії і апаратних витрат.

3. Підсилена здатність працювати з технічною документацією та дотримуватись стандартів проектування – читати, створювати та застосовувати технічну документацію, дотримуватись норм ЕМС, вимог безпеки та виробничих стандартів у розробці електронних систем.

4. Мати навички проведення експериментів, збирання даних та моделювання в комп'ютерних системах.

5. Знати новітні технології в галузі комп'ютерної інженерії.

6. Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.

7. Вміти виконувати експериментальні дослідження за професійною тематикою.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Передумови, тобто дисципліни, вивчення яких має передувати вивченню цієї дисципліни:

- Структури даних і алгоритми.
- Комп'ютерна логіка.
- Комп'ютерна електроніка.
- Архітектура комп'ютерів.
- Комп'ютерна схемотехніка
- Паралельні та розподілені обчислення

Постреквізити - дисципліни, вивченню яких має передувати вивчення даної дисципліни:

- Преддипломна практика.
- дипломне проектування.

3. Зміст навчальної дисципліни

Тема 1: Хід проектування схем з VHDL

Особливості технології проектування схем за допомогою VHDL. Проектування систем на кристалі (СНК). Процес розробки обчислювального пристрою (ОП). Складність проектів СНК і можливості технології проектування. Технологія віртуальних модулів.

Моделі обчислювачів для VHDL. Основи VHDL. Відображення алгоритмів за допомогою VHDL. Синтез логічної схеми. Основні особливості VHDL. Послідовні оператори в VHDL. Паралельні оператори в VHDL. Процедури і функції. Пакети. Атрибути.

Тема 2: Технологія розробки систем на кристалі.

Процес розробки СНК на ПЛІС. Складність проектів СНК і можливості технології проектування. Технологія IP Cores. Сумісне апаратно-програмне проектування SOC. Системний синтез SOC. ПЛІС - як елементна база сучасної електроніки.

Будова і архітектура ПЛІС. Бібліотека IEEE для проектування пристроїв. Відображення алгоритмів на VHDL. Загальні вимоги до проектування систем на кристалі. Конструкції VHDL для синтезу. Стандартні пакети VHDL для синтезу. Проектування комбінаційних схем. Проектування арифметичних пристроїв. Проектування схем з пам'яттю. Проектування керуваних схем. Керування ходом синтезу. Програмування пам'яті.

Тема 3: Проектування конвеєрних обчислювачів.

Особливості конвеєрної обробки даних. Відображення алгоритму в апаратуру. Періодичні алгоритми. Граф синхронних потоків даних (ГСПД). Сигнальний граф. Класифікація графів потоків даних.

Проектування RISC-процесорів. Проектування спеціалізованих обчислювачів. Метод синтезу конвеєрного обчислювача. Оптимізація структури. Конвеєрне виконання алгоритму. Швидкодія ГСПД. Перетворення графа алгоритму в ГСПД. Оптимізація графів алгоритму. Перспективи інтегральної технології. Перспективи ПЛІС.

4. Навчальні матеріали

Базова література

1. Автоматизація проектування цифрових пристроїв [Електронний ресурс] : навч. посіб. уклад.: С. М. Шолохов, В. Д. Голь, О. А. Хахлюк. — Київ : КПІ ім. Ігоря Сікорського, 2025. — 290 с. <https://ela.kpi.ua/handle/123456789/76290>

2. Рябенський В. М. VERILOG. Практика проектування цифрових пристроїв на ПЛІС : Навч. посіб. / В.М. Рябенський, О.О. Ушкаренко ; Нац. ун-т кораблебудування ім. адм. Макарова. — Миколаїв : Іліон, 2007. — 324 с.

3. Автоматизоване проектування апаратних компонентів нейромережових систем керування на FPGA [Електронний ресурс] : монографія / А. Ю. Дорошенко, П. І. Кравець, В. М. Шимкович, О. А. Яценко ; КПІ ім. Ігоря Сікорського. — Електрон. текст. дані (1 файл: 6,71 Мбайт). — Київ : КПІ ім. Ігоря Сікорського, 2026. — 166 с. —

<https://ela.kpi.ua/handle/123456789/82267>

Додаткова література

1. Ashenden P.J., Lewis J. The designer's guide to VHDL. 3-d Ed. Morgan Kaufman. 909 p.
https://www.academia.edu/9455362/The_Designers_Guide_to_VHDL
2. Khan S.A. Digital Design of Signal Processing Systems. A practical Approach. Wiley, 2011. 608 p.
<https://www.academia.edu/34580646/>
3. Harris S., Harris D. Digital Design and Computer Architecture, RISC-V Edition. Morgan Kaufmann. 2021. 592 p.

Обладнання, що необхідне для проведення занять.

Лекційні заняття проводяться в аудиторії, яку обладнано проектором, практичні заняття – в комп'ютерному класі з застосуванням відлагоджувальних плат з ПЛІС. Для виконання лабораторних робіт використовується студентські версії САПР ActiveHDL, Vivado, Quartus.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Головна мета курсу – прищепити студентам основи технології програмування ПЛІС мовами опису мікроелектронної апаратури таких як VHDL і Verilog.

В курсі даються знання про основні властивості архітектур, які програмуються мовою VHDL: , програмної і апаратної моделей та симулятора. Студенти також вивчають як правильно будувати і описувати моделі апаратних пристроїв на основі методу відображення графа синхронних потоків даних (ГСПД) у структуру апаратних засобів. Студенти закріплюють навички програмування у системі ActiveHDL на лабораторних роботах. Викладання предмета з нахилом на міркування проектування апаратури як опису алгоритмів, які вона виконує, дає змогу більш глибоко засвоїти матеріал, пов'язати цей матеріал з практичними потребами при розв'язанні задач проектування нових застосунків на ПЛІС.

Навчальний зміст дисципліни складається з лекцій та лабораторних робіт.

Лекції.

Тема 1: Хід проектування схем з VHDL

Лекція 1. Вступ.

Нова золота ера комп'ютерних архітектур. Підручники, презентації лекційного матеріалу, лабораторні роботи. Рейтингова система оцінювання. Особливості технології проектування схем за допомогою VHDL. Порівняння схемного і модельного проектування. САПР з VHDL. Проектування систем на кристалі (СНК). Процес розробки обчислювального пристрою (ОП). Продуктивність праці розробника ОП. Складність проектів СНК і можливості технології проектування. Технологія віртуальних модулів.

Лекція 2. Моделі обчислювачів для VHDL.

Основи VHDL. Типи даних, що підтримують опис апаратури. Оператори, атрибути, що підтримують поведінку системи в часі. Обчислювальна модель для реалізації VHDL. Архітектура симулятора VHDL. Апаратна модель для реалізації VHDL. Відображення алгоритмів за допомогою VHDL. Синтез логічної схеми.

Лекція 3. Основні особливості VHDL. Об'єкт проекту і архітектура. Об'єкти, типи і вирази мови VHDL. Entity - опис інтерфейса. Architecture - опис поведінки моделі. Поведінкова модель ОП. Типи даних в VHDL. Тип Standard Logic з пакету 1164. Сигнал, змінна, константа.

Лекція 4. Послідовні оператори в VHDL.

Процес і послідовні оператори. Список чутливості. Операції мови VHDL. Вирази мови VHDL. Оператор присвоювання сигналу. Оператори чекання події wait. Оператор присвоювання змінній. Умовні оператори. Оператор вибору. Оператор циклу. Виклик процедури і повернення з неї. Оператори Assert і Report.

Лекція 5. Паралельні оператори в VHDL.

Обчислювальна модель для реалізації VHDL. Паралельні оператори в VHDL. Оператор процесу. Список чутливості. Присвоювання сигналу. Паралельний виклик процедури. Вставка екземпляра компонента. Оператор GENERATE. Оператор BLOCK. Паралельний оператор ASSERT і стенд для іспитів.

Лекція 6. Процедури і функції. Пакети.

Підпрограми. Виклик функції. Виклик процедури. Пакет. Перезавантаження підпрограм. Перетворення типів. Обчислювальна модель для реалізації VHDL. Структура програми. Entity і Architecture. Об'ява конфігурації.

Лекція 7. Атрибути.

Атрибути для скалярного типу. Атрибути для регулярного типу. Атрибути сигналів. Атрибути користувача. Псевдоніми. Мітки в програмі.

Тема 2: Технологія розробки систем на кристалі.

Лекція 8. Технологія розробки систем на кристалі.

Процес розробки систем на кристалі (CHK) на ПЛІС. Складність проектів SOC і можливості технології проектування. Технологія IP Cores. Сумісне апаратно-програмне проектування SOC. Системний синтез SOC. ПЛІС - як елементна база сучасної електроніки. Переваги використання ПЛІС. Приклади ефективного застосування ПЛІС.

Лекція 9. Будова і архітектура ПЛІС.

Апаратні ресурси ПЛІС Xilinx Virtex. Логічна таблиця. Тригер. Конфігуровний логічний блок. Реалізація багатовходових мультиплексорів. Розподілена пам'ять. Регістр зсуву з регульовною затримкою. Блочна пам'ять. Блок DSP48. Буфери вводу-виводу. Мережа розподілення синхросигналів.

Лекція 10. Бібліотека IEEE для проектування пристроїв.

Процес і послідовні оператори. Обчислювальна модель для реалізації VHDL. Апаратна модель для реалізації VHDL. Відображення алгоритмів на VHDL. Синтез логічної схеми. Загальні вимоги до проектування систем на кристалі. Принцип двохтактої синхронізації. Проектування з асинхронними тригерами. Принцип однотої синхронізації. Перекручування однотої синхронізації. Конструкції VHDL для синтезу. Пакет std_logic_1164. Кілька джерел сигналу і функція вирішення. Функції пакета std_logic_1164. Пакети std_logic_arith, std_logic_signed і std_logic_unsigned. Пакети numeric_bit і numeric_std.

Лекція 11. Проектування комбінаційних схем.

Комбінаційна схема за булевським рівнянням. Селективне присвоювання сигналу. Умовне присвоювання сигналу. Комбінаційна схема за умовним присвоюванням. Комбінаційна схема за викликом підпрограми. Комбінаційна схема за послідовним оператором. Приклад розробки комбінаційної схеми. Використання логіки з трьома станами. Довільний стан в Std_Logis. Непередбачувані асинхронні тригери. Проектування арифметичних пристроїв.

Лекція 12. Проектування схем з пам'яттю.

Непередбачувані асинхронні тригери. Принцип однотактної синхронізації. Перекручування однотактної синхронізації. Синтез схем з пам'яттю. Регістри в поведінковому VHDL. Функції фронту/спаду. Оператор WAIT. Змінні в процесі. Процес з регістром. Вставлення компоненту типу регістр. Приклад проектування схеми з пам'яттю.

Лекція 13. Проектування керування і керування проектуванням.

Концепція операційного і керуючого автоматів. Автомат Мура. Автомат Мілі. Приклад проектування автомата. Способи кодування станів автомата. Приклади проектування автомата. Приклад проектування лічильника на регістрі зсуву. Проектування складних автоматів. Мікропрограмний автомат. Приклад проектування блоку обчислення спецфункції. Керування проектуванням. Атрибути керування синтезом. Атрибути обмеження розміщення. Атрибути ініціалізації. Обмеження часу.

Лекція 14. Програмування пам'яті.

Розподілена пам'ять. Регістр зсуву з регульовною затримкою. Блочна пам'ять. FIFO. Пам'ять з коректуючими кодами. Пам'ять з підвищеною ємністю. Програмування розподіленої пам'яті. Програмування BlockRAM. Ініціалізація RAM. Програмування постійної пам'яті.

Тема 3: Проектування конвеєрних обчислювачів.

Лекція 15. Періодичні алгоритми і конвеєрні обчислювачі.

Прискорення обчислень за допомогою обробки даних на ПЛІС. Послідовна і паралельна обробка даних. Система на базі ПЛІС. Особливості конвеєрної обробки даних. Відображення алгоритму в апаратуру. Періодичні алгоритми. Алгоритм, що заданий циклом. Граф періодичного алгоритму. Відображення графа періодичного алгоритму. Граф синхронних потоків даних (ГСПД, SDF). Неоднорідний і однорідний ГСПД. Редукований граф алгоритму. Сигнальний граф. Класифікація графів потоків даних.

Лекція 16. Проектування RISC-процесорів

Основи RISC-процесорів. Конвеєризація. Регістрова пам'ять. Виконання команд під час обробки команди переходу. Спрощена адресація. Простий формат команд. Мікропроцесорне ядро з лабораторних робіт. Приклад синтезу RISC-процесора. Приклад синтезу стекового RISC-процесора.

Лекція 17. Проектування спеціалізованих обчислювачів

Методика синтезу конвеєрного обчислювача. Оптимізація структури. Приклад синтезу блока для обчислення спецфункції. Конвеєрне виконання алгоритму. Швидкодія ГСПД. Перетворення графа алгоритму в ациклічний ГСПД. Оптимізація графів алгоритму. Ресинхронізація. Розгортання ГСПД. Згортання ГСПД.

Лекція 18. Перспективи впровадження ПЛІС. Модульна контрольна робота.

Мови програмування для опису апаратури. Перспективи інтегральної технології. Перспективи ПЛІС.

Лабораторні роботи

Мета лабораторних робіт – експериментальне дослідження та випробування алгоритмів та структур, що вивчаються, при програмуванні завдань лабораторних робіт.

Тематика лабораторних робіт:

Лабораторна робота 1.
Арифметико-логічний пристрій.
Лабораторна робота 2.
Канонічний автомат.
Лабораторна робота 3.
Блок обчислення спецфункції.
Лабораторна робота 4.
Цифровий фільтр.

6. Самостійна робота студента

Види самостійної роботи (66 годин):

- підготовка до лекцій шляхом вивчення попереднього лекційного матеріалу, а також літературних джерел, на яких ґрунтується матеріал попередніх лекцій (0,5 годин x 18 лекцій = 9 годин);
- підготовка до лабораторної роботи через ознайомлення із завданням та методичними вказівками до лабораторних робіт, включаючи вивчення теоретичного матеріалу, необхідного для відповіді на контрольні запитання до лабораторних робіт, складання і програмування алгоритму, компіляцію та відлагодження програми, розгортання програмного середовища та завантаження вихідних кодів програмного забезпечення для виконання лабораторних робіт, робота з системою контролю версій Git, а також оформлення звіту (рекомендовано 5 годин x 4 роботи = 20 годин);
- самостійне опрацювання теоретичного матеріалу (20 годин);
- підготовка до 2 контрольних робіт (4 години x 2 роботи = 8 годин);
- підготовка до МКР (8 годин);
- завантаження та збирання програмного забезпечення для виконання лабораторних робіт (5 годин).

Теми на самостійне опрацювання

Тема 1: Тема 1: Хід проектування схем з VHDL.

Основи роботи з IDE. Графічне введення моделей. Пакети Math_Real, Math_Complex (5 годин).

Тема 2: Технологія розробки систем на кристалі.

Блок множення-ділення частоти з цифровим чи фазовим зворотнім зв'язком у ПЛІС. Блоки послідовного вводу-виводу. Вбудовані мікропроцесори. Шини Avalon і AXI. (10 годин).

Тема 3: Проектування конвеєрних обчислювачів.

Soft-моделі мікропроцесорів Cortex, Microblaze, RISC-V. (5 годин).

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Система вимог до студентів:

- студент зобов'язаний відвідувати лекції та лабораторні заняття, активно працювати над засвоєнням викладеного на них матеріалу;
- на лекції лектор використовує власний презентаційний матеріал;
- лабораторні роботи захищаються у два етапи: перший етап - студенти виконують завдання, складають електронний звіт та надсилають викладачу; другий етап - захист

лабораторних робіт у лабораторії. Контроль знань на лабораторних роботах здійснюється шляхом перевірки звіту про лабораторні роботи, а також питань з матеріалів робіт.

- модульна контрольна робота пишеться як тест на лекції з використанням усіх наявних матеріалів, але без використання комп'ютерних засобів.

Дедлайни. Для виконання лабораторних робіт встановлюються дедлайни. Дедлайни уточнюються і оприлюднюються викладачем на початку семестру.

Заохочувальні бали (бонуси) виставляються за: активну участь на лекціях та лабораторних заняттях, ведення конспекту, підготовку повідомлення з презентацією по одній із тем СРС дисципліни тощо. Кількість заохочуваних балів не більше 5.

Лабораторні роботи. Виконанню кожної лабораторної роботи передуює реалізація індивідуального завдання і оформлення його у вигляді протоколу. Протокол виконання лабораторного завдання має бути вислано на пошту викладача не пізніше ніж за добу до лабораторного завдання. До протоколу додаються посилання на Git з кодом програм (за необхідності) і посилання на всі необхідні супровідні матеріали для захисту, зокрема на відеозапис виконання лабораторного завдання (за необхідності, в дистанційному режимі або за використання віддаленого устаткування, тощо). Студент, який відвідав заняття без підготовленого протоколу до захисту лабораторної роботи не допускається. Першим етапом студент захищає результати виконання індивідуального завдання до лабораторної роботи, на другому етапі – захищає теорію шляхом усного опитування або тестування. В дистанційному режимі більшість лабораторних робіт супроводжуються тестами для оцінки вивченого теоретичного та лабораторного матеріалу замість усного опитування. Бали отримані за виконання лабораторної роботи - за тест та за протокол складають оцінку за лабораторну роботу. Тестування проводиться на лабораторному занятті після перевірки результатів виконання індивідуальної роботи. Студент, який не виконав індивідуальне завдання до лабораторної роботи до тесту не допускається.

Виконання лабораторних робіт та МКР є обов'язковими для допуску до семестрового контролю (заліку). Загальною умовою допуску до семестрового контролю (заліку) є зарахування всіх лабораторних робіт, МКР та загальний рейтинг не менше 60 балів.

Оцінка, яку студент може отримати за виконання кожної лабораторної роботи наведені у розділі 8 силабуса.

Здобувачі, які виконали всі умови допуску до заліку та мають рейтингову оцінку менше 60 балів, а також здобувачі, які бажають підвищити свою рейтингову оцінку, на останньому за розкладом занятті складають залік у вигляді виконання залікового завдання та співбесіди.

З метою посилення зацікавленості здобувачів у якісному виконанні індивідуальних семестрових завдань, передбачених індивідуальним навчальним планом здобувача, рейтингова (семестрова) оцінка, у разі виконання залікової контрольної роботи, визначається, як оцінка за залікову контрольну роботу та сума балів за індивідуальні семестрові завдання (що відповідають балам за лабораторні роботи протягом семестру). Всі інші бали, які набрані протягом семестру (окрім балів за лабораторні роботи) анулюються.

Модульна контрольна робота є обов'язковою для допуску до семестрового контролю. Модульна контрольна робота (МКР) пишеться на лекційному занятті без застосування допоміжних засобів (мобільні телефони, планшети та ін.); результат пересилається до відповідної директорії Google -диску через Google форму.

У разі виконання модульної контрольної роботи рейтингова (семестрова) оцінка визначається як сума балів за модульну контрольну роботу, балів за індивідуальні семестрові завдання (лабораторні роботи).

До індивідуальних семестрових завдань вноситься індивідуальна робота студента що стосується виконання лабораторних робіт. Кількість балів за індивідуальну роботу за семестр дорівнює від 38 до 60 балів. Максимальна оцінка за МКР дорівнює 40 балів.

Правила відвідування занять. Відвідування занять є вільним, бали за присутність на лекціях та лабораторних заняттях не додаються, відповідно штрафні бали не передбачаються. Втім, вагома частина рейтингу формується через активну участь у заходах на лабораторних заняттях.

Дистанційний режим навчання. У разі запровадження обмежень на відвідування університету, пов'язаних з введенням карантину або режиму воєнного стану в державі, освітній процес здійснюється у дистанційному режимі відповідно до Положення про дистанційне навчання в КПІ ім. Ігоря Сікорського (<https://osvita.kpi.ua/index.php/node/188>), Регламенту організації освітнього процесу в дистанційному режимі (<https://profkom.kpi.ua/reglament-organizatsiyi-osvitnogo-protsesu-v-distsantsiynomu-rezhimi>) та Регламенту проведення семестрового контролю в дистанційному режимі (<https://osvita.kpi.ua/node/148>). У режимі дистанційного навчання заняття відбуваються у вигляді онлайн-конференції на платформах BigBlueButton, Google Meet, Zoom. Посилання на конференцію видається на початку семестру і розміщується в АС «Електронний кампус». З метою забезпечення якісної підготовки здобувачів, дистанційний курс дисципліни розміщено на Платформа дистанційного навчання «Сікорський» (<https://www.sikorsky-distance.org>). Результати оцінювання висвітлюють у АС «Електронний кампус» на особистій сторінці здобувача (<https://ecampus.kpi.ua>).

Правила поведінки на заняттях. На заняттях слід дотримуватись норм етичної поведінки визначених у Кодексі честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>). На території університету здобувачі мають поводити себе відповідно до Правил внутрішнього розпорядку (<https://kpi.ua/admin-rule>).

Визнання результатів навчання, набутих у неформальній/інформальній освіті. Положення про визнання результатів навчання, набутих у неформальній / інформальній освіті (<https://osvita.kpi.ua/index.php/node/179>) регламентує визнання результатів навчання, набутих у неформальній/інформальній освіті.

Неформальна освіта. У разі проходження здобувачем онлайн курсу чи іншого елементу неформальної освіти здобувачеві/здобувачці можуть бути зараховані окремі змістовні модулі або теми дисципліни. В такому разі здобувач звільняється від виконання відповідних лабораторних робіт, отримуючи за них максимальний бал відповідно до рейтингової системи оцінювання. У разі навчання на неформальній освіті за самостійного її обрання проходиться процедура валідації, що передбачає подання здобувачем заяви на ім'я декана, декларації підтверджувальних документів. Рішення про визнання чи не визнання приймається комісією у складі завідувача кафедри, викладача, гаранта освітньо-професійної програми.

Політика використання штучного інтелекту. Використання штучного інтелекту регламентується «Політикою використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського» (<https://osvita.kpi.ua/node/1225>). Усі завдання, як під час виконання навчальних завдань з дисципліни, так і індивідуальні завдання, мають бути результатом власної оригінальної роботи здобувача. Використання ШІ для автоматичної генерації відповідей без подальшого їх аналізу та доопрацювання заборонено.

Порушення академічної доброчесності: використання плагіату. У разі виявлення фактів використання плагіату під час виконання навчальних завдань, студент втрачає право на подальше виконання поточних завдань у межах навчальної дисципліни до з'ясування обставин порушення. Студент зобов'язаний аргументовано пояснити причини використання плагіату, а також засвідчити готовність дотримуватись принципів академічної доброчесності в подальшому навчанні; повторно виконати завдання або, за потреби, виконати додаткове завдання відповідно до бачення викладача з урахуванням вимог щодо самостійності.

Зазначені дії регламентуються Кодексом честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>)

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

На першому занятті студенти знайомляться з політикою оцінювання, яка ґрунтується на Положенні про систему оцінювання результатів навчання https://document.kpi.ua/files/2020_1-273.pdf. Рейтинг студента на курсі складається з балів, які він/вона отримує за захищені лабораторні роботи (R1), контрольні роботи (R2) і насамкінець, відповіді на іспиті (R3).

$$R_s = R_1 + R_2 + R_3 = 100 \text{ балів}$$

В результаті максимальний середній бал дорівнює:

$$4 \text{ лабораторних роботи} \times 12 \text{ балів} = 48 \text{ балів}$$

$$2 \text{ контрольні роботи} \times 6 \text{ балів} = 12 \text{ балів}$$

$$\text{Залікова контрольна робота} = 40 \text{ балів.}$$

Згідно з університетськими правилами щодо моніторингу успішності студента (https://kpi.ua/document_control), існує два тижні оцінювання (атестація), зазвичай протягом 7/8 та 14/15 тижнів семестру, коли оцінюється прогрес у навчанні студентів та відповідність до критеріїв політики оцінювання курсу.

Умовою першої атестації є отримання не менше 10 балів (на момент атестації). Умова другої атестації - набрати не менше 25 балів (на момент атестації).

Критеріями оцінки є:

- Виконання лабораторних робіт:
 - бездоганна робота оцінюється у 11 балів;
 - є певні недоліки в оформленій роботі - 10-9 балів;
 - є деякі недоліки у виконанні робочої програми - 8-6 бали;
 - Робота не виконана або не захищена - 0 балів.

За роботу, подану вчасно, діє заохочення - 1 бал (сумарно не більше - 6 балів).

Дозволяється зарахувати 2 лабораторні роботи з оцінкою 10 за рахунок проходження курсу вивчення програмування мовою VHDL у системі Coursera та їй подібній.

- Контрольна робота оцінюється 6 балами. Контрольна робота складається з 2 практичних задач зі списку, наведеного в додатку до робочої програми.

Відповідь на практичне завдання оцінюється 3 балами за такими критеріями:

- "відмінно" - правильний текст програми на VHDL з коментарями - 3 бали;
- "добре" - текст програми, загалом правильний, але є синтаксичні помилки - 2 бали;
- "задовільно" - є деякі принципові помилки в тексті програми - 1 бал;
- "незадовільно" - незадовільна відповідь - 0 балів.

Студенти можуть отримати до 6 заохочувальних балів за виконання творчих робіт з кредитного модуля (складання тез, участь у конкурсах, у дослідженнях тощо).

Студенти, чий остаточний бал успішності перевищує 30, повинні скласти іспит. Студенти, чий бал нижче 30, не допускаються до складання іспиту.

Модульна контрольна робота оцінюється 40 балами. Контрольна робота складається з 16 тестових питань, а також практичного завдання (скласти програму) зі списку, наведеного в додатку до робочої програми. За кожну правильну відповідь на тестове питання нараховується 2 бали. Відповідь на практичне завдання оцінюється 8 балами за такими критеріями:

- "відмінно" - правильний текст програми з коментарями - 8 - 7 балів;
- "добре" - текст програми, загалом правильний, але не використані прийоми оптимізації - 6-5 балів;

- "задовільно" - є деякі принципові помилки в тексті програми - 4 - 3 бали;
- "незадовільно" - незадовільна відповідь - 0 балів.

Студенти, які нарешті наберуть необхідну кількість балів (≥ 60), можуть:

- отримати остаточну залікову оцінку відповідно до рейтингового балу;
- виконати залікову контрольну роботу, щоб підвищити оцінку.

Студенти можуть отримати до 6 заохочувальних балів за виконання творчих робіт з кредитного модуля (складання тез, участь у конкурсах, у дослідженнях тощо).

Студенти, чий остаточний бал за успішність R_s нижче 60 балів, але більше за 30, повинні виконати залікову контрольну роботу. Студенти, чий остаточний бал за успішність R_s вище або дорівнює 60 балів, можуть виконати залікову контрольну роботу, щоб підвищити оцінку. В обох випадках застосовується вимога - попередній рейтинг студента складає: $0.7R_1$. Студенти, чий бал нижче 30, не допускаються до заліку.

Максимальна оцінка R_3 за залікову контрольну роботу складає 60 балів. Контрольна робота складається з 25 тестових питань, а також практичного завдання (скласти програму). За кожну правильну відповідь на тестове питання нараховується 2 бали. Відповідь на практичне завдання оцінюється 10 балами за такими критеріями:

- "відмінно" - правильний текст програми з коментарями - 10 - 9 балів;
- "добре" - текст програми, загалом, правильний, але не використані прийоми оптимізації - 8-6 балів;

- "задовільно" - є деякі принципові помилки в тексті програми - 5 - 4 балів;
- "незадовільно" - незадовільна відповідь - 0 балів.

Рейтингова оцінка разом з оцінкою за залікову контрольну роботу дорівнює

$$R_s = 0.7(R_1 + R_2) + R_3.$$

Таблиця. Остаточна оцінка R_s приймається за системою оцінювання університету

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація про курс

Завдання в контрольній роботі виглядають як наступні.

Побудувати часову діаграму сигналу, що генерується наступним процесом

```
process begin
```

```
    A<='0';
```

```
    wait for 10 ns;
```

```
    A<='1'; B<=A;
```

```
    wait for 20 ns;
```

```
end process;
```

2. Описати на VHDL мультиплексор MUX5 з однорозрядними входами даних A,B,C,D,E входом селекції S і виходом X за допомогою булевських рівнянь без використання бібліотеки IEEE. Скільки операторів NOT слід використати?

3. Описати на VHDL синхронний T – тригер з входами рахування T, синхросерії C, асинхронного встановлення в нуль R і виходом Q без використання бібліотеки IEEE. Скільки операторів if потрібно використати?

4. Розробити спецпроцесор, який виконує задану функцію за послідовним алгоритмом

$$Y1 = 0.98 * X1 - 0.195 * X2; \quad Y2 = 0.195 * X1 + 0.98 * X2;$$

Початкові дані X1, X2 – 16-розрядні в доповнюючому коді, результати Y1, Y2 – 16-розрядні, приходять і виходять по окремих шинах.

Робочу програму навчальної дисципліни (силабус):

Складено д.т.н., проф., А. М. Сергієнко

Ухвалено кафедрою обчислювальної техніки (протокол №18 від 24.06.2026)

Погоджено Методичною комісією ФІОТ (протокол №12 від 26.06.2026)