

Робоча програма дисципліни «Технології проектування комп'ютерних систем»

СПЕЦІАЛЬНІСТЬ 123 · КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

БАКАЛАВР

150

Годин

Загальний обсяг дисципліни

5

Кредити ЄКТС

Відповідно до стандарту

5

Модулі

Змістовних блоки курсу



Загальна інформація про дисципліну

Ідентифікація

Назва: Технології проектування комп'ютерних систем

Спеціальність: 123 Комп'ютерна інженерія

Ступінь: Бакалавр

Обсяг: 150 годин / 5 кредитів ЄКТС

Модулів: 5 змістовних блоки

Лекції: 45 год. | **Лабораторні:** 30 год. | **СРС:** 75 год.

Призначення дисципліни

Дисципліна «Технології проектування комп'ютерних систем» формує у студентів системне розуміння методів, засобів і технологій проектування сучасних комп'ютерних систем — від аналізу вимог до верифікації та тестування готового рішення. Курс охоплює CASE-засоби, методології структурного та об'єктно-орієнтованого проектування, апаратно-програмні аспекти, а також технології автоматизованого проектування (САПР).

Дисципліна є обов'язковою для підготовки бакалаврів зі спеціальності «Комп'ютерна інженерія» та забезпечує формування фахових компетентностей, передбачених освітньо-професійною програмою.

Загальні компетентності

Системне мислення

Здатність до аналізу, синтезу та критичного оцінювання технічних рішень: виявлення причинно-наслідкових зв'язків між вимогами та архітектурою системи, порівняння альтернативних підходів до проектування.

Комунікація

Здатність до усної та письмової технічної комунікації: чітко пояснювати архітектурні рішення, оформлювати технічне завдання, звіти та проектну документацію відповідно до ДСТУ.

Самоорганізація

Уміння самостійно навчатися, планувати роботу та організовувати виконання проектних завдань у визначені строки, зокрема під час виконання лабораторних робіт і курсових проектів.

Командна робота

Відповідальність, ініціативність та готовність працювати в команді під час групового проектування систем, рецензування архітектурних рішень і спільного захисту технічних проектів.

Студент повинен вміти: аналізувати технічну та наукову інформацію; формувати вимоги до системи; планувати власну проектну діяльність і дотримуватися строків; грамотно спілкуватися з викладачами та колегами; брати участь у командному проектуванні та розподіляти ролі в команді.

Фахові компетентності

Проектування комп'ютерних систем

Здатність проектувати комп'ютерні системи та мережі з урахуванням цілей, обмежень, технічних, економічних та правових аспектів. Знання архітектурних стилів, шаблонів проектування та принципів побудови надійних систем.

CASE-засоби та САПР

Здатність застосовувати системи автоматизованого проектування (САПР) та CASE-засоби для моделювання, аналізу й верифікації комп'ютерних систем. Практичне володіння сучасними інструментами проектування.

Моделювання та верифікація

Здатність будувати та досліджувати моделі комп'ютерних систем; застосовувати методи функціонального, структурного та об'єктно-орієнтованого моделювання; виконувати верифікацію та тестування проектних рішень.

Нормативно-технічна база

Орієнтування в нормативно-технічній документації у сфері проектування комп'ютерних систем; використання стандартів ДСТУ, ISO/IEC та IEEE для оформлення технічної документації та обґрунтування проектних рішень.

Результати навчання

01

Аналіз вимог до системи

Аналізувати вимоги замовника, формувати технічне завдання, визначати функціональні та нефункціональні вимоги до комп'ютерної системи, оцінювати їх повноту й несуперечливість.

03

Застосування CASE/САПР

Застосовувати CASE-засоби та системи автоматизованого проектування для синтезу та аналізу комп'ютерних систем; виконувати моделювання на інструментальних платформах Quartus II, ModelSim, Enterprise Architect.

02

Проектування архітектури

Розробляти архітектуру комп'ютерних систем із застосуванням структурних та об'єктно-орієнтованих методів; будувати діаграми UML, DFD, IDEF0; обирати архітектурні патерни та обґрунтовувати рішення.

04

Верифікація та документування

Виконувати верифікацію проектних рішень; складати проектну документацію відповідно до стандартів; оцінювати якість, надійність і ефективність розробленої системи.

Основи проектування комп'ютерних систем

1

Тема 1.1

Вступ до дисципліни. Поняття комп'ютерної системи, її структура та класифікація

2

Тема 1.2

Методології та технології проектування: структурний, об'єктно-орієнтований та модельно-орієнтований підходи

3

Тема 1.3

Стандарти проектування та оформлення технічної документації (ДСТУ, ISO/IEC, IEEE)

4

Тема 1.4

Аналіз вимог. Технічне завдання: структура, методи розробки, верифікація вимог

Вступ. Поняття та класифікація комп'ютерних систем

Лекційний матеріал (3 год.)

Лекція 1: Предмет і завдання дисципліни. Поняття «комп'ютерна система» (КС): визначення, основні характеристики, відмінність від комп'ютерної мережі та окремого пристрою. Принципи Неймана та сучасні архітектурні парадигми.

Лекція 2: Класифікація комп'ютерних систем: за призначенням (вбудовані, серверні, мейнфреймні, суперкомп'ютери); за архітектурою (RISC/CISC, SMP, NUMA, кластерні); за масштабом. Тенденції розвитку: IoT, Edge Computing, AI-accelerated systems.

Ключові питання лекції

- Визначення комп'ютерної системи та її компонентного складу
- Класифікація КС за архітектурою та призначенням
- Сучасні тенденції розвитку комп'ютерних систем
- Місце дисципліни в підготовці інженера зі спеціальності 123

Лабораторна робота (2 год.)

Завдання 1. Огляд та порівняльний аналіз архітектур сучасних КС: вивчення специфікацій реальних систем (сервер, вбудована система, НРС-кластер), заповнення порівняльної таблиці за критеріями (архітектура, продуктивність, призначення, масштабованість).

Завдання 2. Ознайомлення з інструментальним середовищем курсу: встановлення та базова конфігурація Quartus II / ModelSim, огляд інтерфейсу Enterprise Architect.

Самостійна робота (8 год.)

- Опрацювання: Артюхов В. Г. та ін. «Технології комп'ютерного проектування» — КПІ ім. Ігоря Сікорського, 2022 — розділ 1
- Підготовка реферату: «Сучасні тенденції розвитку комп'ютерних систем: IoT, Edge та AI-accelerated архітектури»
- Конспектування стандарту IEEE 1471 «Recommended Practice for Architectural Description»



Очікуваний результат: студент вміє класифікувати комп'ютерні системи, описувати їх архітектурні характеристики та орієнтуватися в сучасних тенденціях розвитку галузі.

Методології проектування комп'ютерних систем

1

Структурний підхід

DFD, IDEF0, IDEF1X. Декомпозиція функцій. Низхідне проектування.

2

Об'єктно-орієнтований

UML-нотація. Діаграми класів, послідовностей, компонентів. Патерни GoF.

3

Модельно-орієнтований

MDA/MDD. Platform Independent Model (PIM). Platform Specific Model (PSM). Трансформації.

4

Agile-підходи

Scrum, Kanban у контексті проектування систем. Ітераційне уточнення вимог та архітектури.

Лекційний матеріал (3 год.)

Структурна методологія проектування: діаграми потоків даних (DFD), функціональне моделювання IDEF0, інформаційне моделювання IDEF1X. Об'єктно-орієнтована методологія: UML 2.x, нотація, основні типи діаграм. Модельно-орієнтований підхід (MDA): рівні моделей, трансформації. Порівняльний аналіз методологій. Критерії вибору підходу залежно від типу проекту.

Лабораторна робота (2 год.) та СРС (8 год.)

Лабораторна: Побудова функціональної моделі IDEF0 для типової комп'ютерної системи; розробка діаграми варіантів використання (Use Case) та діаграми класів в Enterprise Architect; порівняльний аналіз структурного та ОО-підходів.

СРС: Опрацювання: Ізмайлова О. В. «Проектування інформаційних систем» — Київ : КНУБА, 2023; підготовка аналітичної записки «Методологія проектування: структурний vs. об'єктно-орієнтований підхід — переваги та обмеження».

Стандарти проектування та аналіз вимог

Тема 1.3 — Стандарти та документування (Лекція 4–5, 3 год.)

Міжнародні стандарти проектування: ISO/IEC 25010 (якість системи), ISO/IEC 12207 (ЖЦ ПЗ), IEEE 829 (тестова документація), ДСТУ 3973, ГОСТ 34.601. Структура та зміст конструкторської документації. Вимоги до оформлення технічного завдання (ТЗ), пояснювальної записки, специфікації. Використання шаблонів і засобів автоматичного генерування документації.

Лабораторна (2 год.): Розробка структури технічного завдання на проектування КС відповідно до ДСТУ; оформлення специфікації вимог у середовищі Enterprise Architect.

СРС (8 год.): Опрацювання: Артюхов В. Г. та ін. «Технології комп'ютерного проектування» — КПІ ім. Ігоря Сікорського, 2022 — розділ «Документування»; підготовка ТЗ на власний проект.

Тема 1.4 — Аналіз вимог (Лекція 6, 3 год.)

Поняття вимог до КС. Класифікація вимог: функціональні, нефункціональні (продуктивність, безпека, надійність, зручність). Методи збору вимог: інтерв'ювання, анкетування, прецеденти. Специфікація вимог (SRS — Software/System Requirements Specification). Трасування вимог. Прототипування як метод уточнення вимог. Управління змінами вимог. Верифікація та валідація вимог.

Лабораторна (2 год.): Формування SRS для заданої КС: виявлення та специфікація вимог, побудова матриці трасування вимог, рецензування SRS однокурсника.

СРС (8 год.): Опрацювання: Ізмайлова О. В. «Проектування інформаційних систем» — Київ : КНУБА, 2023 — розділ «Аналіз вимог»; підготовка специфікації вимог до власного проекту; аналіз ДСТУ 3973-2000.

Архітектурне проектування комп'ютерних систем

Тема 2.1

Архітектура комп'ютерної системи: рівні абстракції, архітектурні стилі та патерни. Принципи SOLID.

Тема 2.2

Проектування апаратної підсистеми: мікропроцесорна база, пам'ять, периферія, інтерфейси.

Тема 2.3

Проектування програмно-апаратного стека. Взаємодія ОС, драйверів та прикладного ПЗ.

Тема 2.4

Розподілені та паралельні КС. Архітектура клієнт-сервер, мікросервісна, хмарна.

Архітектурні стилі та патерни проектування

Лекційний матеріал (3 год.)

Поняття архітектури КС. Рівні архітектурного опису: концептуальний, логічний, фізичний. Архітектурні стилі: монолітний, клієнт-серверний, шаровий (N-tier), подієво-орієнтований, мікросервісний, SOA. Архітектурні патерни: MVC, MVP, MVVM, Repository, CQRS, Event Sourcing. Принципи SOLID та їх застосування в проектуванні КС. Поняття технічного боргу.

Ключові питання

- Рівні архітектурного опису комп'ютерної системи
- Порівняльна характеристика архітектурних стилів
- Застосування патернів GoF у проектуванні КС
- Принципи SOLID: зміст та практика застосування
- Критерії вибору архітектурного рішення

Лабораторна робота (2 год.)

Завдання 1. Аналіз та порівняння архітектурних стилів: складання порівняльної таблиці за критеріями (масштабованість, надійність, складність, типові застосування).

Завдання 2. Проектування архітектури тестової КС з використанням шаблону MVC у середовищі Enterprise Architect: побудова діаграм компонентів та розгортання.

Завдання 3. Тестування за ключовими термінами теми (10 питань).

Самостійна робота (8 год.)

- Опрацювання: Авраменко В. С., Авраменко А. С. «Проектування інформаційних систем» — Харків, 2023 — розділ «Архітектурне проектування»
- Підготовка реферату: «Мікросервісна архітектура: переваги, недоліки та сфери застосування»
- Аналіз відкритих архітектурних рішень (AWS, Azure, Google Cloud) та їх зіставлення з класичними стилями



Очікуваний результат: студент вміє описувати та порівнювати архітектурні стилі, обирати патерн проектування відповідно до вимог задачі та обґрунтовувати своє рішення.

Проектування апаратної підсистеми



Лекційний матеріал (3 год.)

Апаратна архітектура КС: принципи вибору мікропроцесорної платформи. Ієрархія пам'яті та методи оптимізації доступу. Периферійні інтерфейси: стандарти, характеристики, сфери застосування. Проектування тактового дерева та підсистеми живлення. Вступ до FPGA-проектування: структура, переваги, типові застосування. Засоби HDL (VHDL, Verilog).

Лабораторна (2 год.) та СРС (8 год.)

Лабораторна: Виконання лабораторної роботи в Quartus II: синтез простого цифрового пристрою мовою VHDL, верифікація у ModelSim, аналіз часових діаграм.

СРС: Опрацювання: Артюхов В. Г. та ін. «Технології комп'ютерного проектування. Комп'ютерний практикум» — КПІ ім. Ігоря Сікорського, 2022 — розділи з VHDL та Quartus II; підготовка аналітичної записки «Порівняльний аналіз платформ FPGA для побудови вбудованих КС».

Програмно-апаратний стек та розподілені системи

Тема 2.3 — Програмно-апаратний стек (Лекції, 3 год.)

Взаємодія ОС і апаратного забезпечення: HAL (Hardware Abstraction Layer), BSP (Board Support Package). Розробка та налагодження драйверів пристроїв. RTOS (Real-Time Operating System): класифікація, критерії вибору (FreeRTOS, Zephyr, VxWorks). Стек ПЗ вбудованої КС: bootloader, ОС, middleware, застосунок. Інструменти відлагодження: JTAG, SWD, логічний аналізатор.

Лабораторна (2 год.): Налаштування середовища розробки для вбудованої КС: конфігурація BSP, компіляція та завантаження ПЗ на тестову платформу (STM32 або FPGA-based); аналіз роботи HAL.

СРС (8 год.): Опрацювання: Доценко С. І. «Організація та системи керування базами даних» — Харків : УкрДУЗТ, 2023 (розділ про системне ПЗ); підготовка реферату «RTOS у вбудованих системах: порівняльний аналіз FreeRTOS та Zephyr».

Тема 2.4 — Розподілені та паралельні КС (Лекції, 3 год.)

Архітектура розподілених систем: клієнт-сервер, P2P, хмарні обчислення. Паралельні обчислення: SMP, NUMA, кластери. Парадигми міжпроцесної взаємодії: MPI, OpenMP, gRPC, REST API, Message Queues. Проектування масштабованих систем: горизонтальне та вертикальне масштабування, балансування навантаження. Виклики розподілених систем: теорема CAP, узгодженість, відмовостійкість.

Лабораторна (2 год.): Проектування та тестування простої розподіленої системи: реалізація клієнт-серверного застосунку з REST API; аналіз продуктивності при різних навантаженнях.

СРС (8 год.): Підготовка аналітичної записки «Хмарні архітектури: IaaS, PaaS, SaaS — порівняння та критерії вибору для побудови корпоративних КС».

CASE-засоби та автоматизоване проектування

Тема 3.1

Класифікація та огляд CASE-засобів. Засоби верхнього, середнього та нижнього рівня. Enterprise Architect, Rational Rose, DOORS.

Тема 3.2

Системи автоматизованого проектування (САПР): EDA-засоби, Quartus II, Vivado, ModelSim. Проектування цифрових пристроїв.

Тема 3.3

Моделювання КС: імітаційне, аналітичне, формальне. Засоби: MATLAB/Simulink, SystemC, UPPAAL.

Тема 3.4

Генерація коду та синтез: перетворення моделей у виконуваний код/схему. MDA-трансформації, RTL-синтез.

CASE-засоби: класифікація та застосування

Лекційний матеріал (3 год.)

Поняття та призначення CASE (Computer-Aided Software Engineering). Класифікація CASE-засобів: Upper CASE (аналіз та моделювання), Middle CASE (проектування), Lower CASE (генерація коду, тестування), Integrated CASE. Огляд провідних інструментів: Enterprise Architect (Sparx Systems), IBM Rational Rose, Microsoft Visio, DOORS, StarUML. Критерії вибору CASE-засобу. Інтеграція CASE-засобів у процес розробки. Управління вимогами з DOORS. Репозиторій проекту та командна робота.

Ключові питання

- Класифікація CASE-засобів та їх функціональні можливості
- Порівняння Enterprise Architect, IBM Rational та StarUML
- Управління вимогами та трасування в DOORS
- Колективна робота з CASE-засобами в команді
- Критерії вибору інструментарію для конкретного проекту

Лабораторна робота (2 год.)

Завдання 1. Комплексна робота в Enterprise Architect: створення проекту, побудова діаграм Use Case, Class, Sequence та Deployment для модельної КС.

Завдання 2. Генерація звіту та документації з Enterprise Architect; аналіз отриманих артефактів на відповідність стандартам.

Завдання 3. Тестування за ключовими термінами теми (10 питань).

Самостійна робота (8 год.)

- Опрацювання: Ізмайлова О. В. «Проектування інформаційних систем» — Київ : КНУБА, 2023 — розділ «CASE-засоби»
- Підготовка порівняльного аналізу «Enterprise Architect vs. StarUML: функціональність, вартість, інтеграція»
- Виконання самостійного практикуму: побудова моделі власної КС у StarUML (безкоштовна альтернатива)



Очікуваний результат: студент вміє класифікувати CASE-засоби, застосовувати Enterprise Architect для повного циклу UML-моделювання та генерувати проектну документацію.

САПР та EDA–засоби проектування цифрових систем

Лекційний матеріал (3 год.)

Поняття EDA (Electronic Design Automation). Засоби Intel Quartus Prime: структура проекту, синтез, Place & Route, аналіз часових обмежень (Timing Analyzer). Xilinx Vivado: порівняння з Quartus, Flow Manager. ModelSim: функціональна та часова симуляція, написання тестбенчів. HDL-мови: VHDL та Verilog — синтаксис, семантика, стилі написання синтезованого RTL-коду. Принцип «схема — HDL» та зворотний зв'язок у проектуванні.

Лабораторна (2 год.) та СРС (8 год.)

Лабораторна: Повний цикл EDA-проективання: розробка RTL-моделі скінченного автомата мовою VHDL → синтез у Quartus Prime → функціональна симуляція в ModelSim → аналіз часових діаграм → завантаження у FPGA (DE0 або аналог).

СРС: Опрацювання: Артюхов В. Г. та ін. «Технології комп'ютерного проектування. Комп'ютерний практикум» — КПІ ім. Ігоря Сікорського, 2022 — цикл лабораторних робіт з VHDL; підготовка аналітичної записки «Порівняльний аналіз Quartus Prime та Vivado: можливості, обмеження, екосистема».

- ❑ **Очікуваний результат:** студент вміє розробляти RTL-моделі мовою VHDL, виконувати синтез та симуляцію у Quartus Prime та ModelSim.

Моделювання та генерація коду

Тема 3.3 — Моделювання КС (Лекції, 3 год.)

Класифікація методів моделювання: імітаційне (дискретно-подієве), аналітичне (теорія масового обслуговування), формальне (автоматна верифікація). Системи MATLAB/Simulink для моделювання реального часу та вбудованих систем. SystemC як мова системного рівня. Засоби формальної верифікації: UPPAAL (мережі автоматів), SPIN (моделювання паралелізму). Параметрична оптимізація системи на основі симуляції.

Лабораторна (2 год.): Імітаційне моделювання підсистеми КС: розробка Simulink-моделі вбудованої системи керування; аналіз часових характеристик та оптимізація параметрів.

СРС (8 год.): Опрацювання: навчальний посібник з MATLAB/Simulink (КПІ, 2022); підготовка реферату «Формальна верифікація КС: методи та сучасні інструменти».

Тема 3.4 — Генерація коду та синтез (Лекції, 3 год.)

Концепція Model-Driven Architecture (MDA): PIM → PSM → Code. Автоматична генерація коду з UML-моделей (Enterprise Architect, Acceleo). RTL-синтез: перетворення HDL-опису у схему вентиляного рівня. High-Level Synthesis (HLS): синтез із C/C++ у RTL (Intel HLS Compiler, Xilinx Vitis HLS). Переваги та обмеження підходу. Оцінка якості синтезованого рішення: площа, швидкодія, споживання.

Лабораторна (2 год.): Виконання HLS-синтезу: написання C-функції з прагмами оптимізації → HLS-синтез у Vitis HLS → порівняння результатів з ручним RTL-кодом за метриками площа/швидкодія.

СРС (8 год.): Підготовка аналітичної записки «High-Level Synthesis у сучасному проектуванні КС: переваги для прискорення циклу розробки»; виконання індивідуального завдання з MDA-трансформацій.

Верифікація, тестування та забезпечення якості КС

Тема 4.1

Поняття верифікації та валідації. Рівні тестування КС: модульне, інтеграційне, системне, приймальне.

Тема 4.2

Методи функціональної верифікації: тестбенчи, Coverage-Driven Verification (CDV), Universal Verification Methodology (UVM).

Тема 4.3

Надійність та відмовостійкість КС. Методи підвищення надійності: резервування, FMEA, FTA. Стандарт IEC 61508.

Тема 4.4

Метрики якості КС. ISO/IEC 25010. Оцінювання продуктивності, безпеки та зручності використання системи.

Верифікація та тестування комп'ютерних систем

Лекційний матеріал (3 год.)

Поняття верифікації та валідації (V&V). Стандарт IEEE 829: план тестування, специфікація тест-кейсів, звіт. Рівні тестування: модульне (Unit), інтеграційне, системне, приймальне (UAT). Методи «чорного» та «білого» ящика. Стратегії тестування вбудованих систем: Hardware-in-the-Loop (HIL), Software-in-the-Loop (SIL). Перспективне тестування. Засоби автоматизації: Jenkins, GitLab CI, pytest, GTest.

Ключові питання

- Відмінність верифікації від валідації
- Рівні тестування та їх цілі
- HIL та SIL тестування вбудованих систем
- Автоматизація тестування у CI/CD-конвеєрі

Лабораторна робота (2 год.)

Завдання 1. Розробка плану тестування КС відповідно до IEEE 829: визначення рівнів, методів, критеріїв входу/виходу, переліку тест-кейсів.

Завдання 2. Реалізація модульних тестів для компонента КС на Python/C++ з використанням unittest/GTest; аналіз покриття коду.

Завдання 3. Демонстрація CI/CD-конвеєра: автоматичний запуск тестів при push у репозиторій (GitLab CI).

Самостійна робота (8 год.)

- Опрацювання: Шульга В. В. та ін. «Інформаційна безпека комп'ютерних мереж та систем» — Харків : ХНУПС, 2023 — розділ «Тестування та верифікація»
- Підготовка реферату «Hardware-in-the-Loop: принципи, застосування в автомобільній та авіаційній промисловості»
- Розробка тест-плану для власного проекту КС



Очікуваний результат: студент вміє розробляти плани тестування, реалізовувати модульні тести та організовувати автоматизоване тестування у CI/CD-конвеєрі.

Функціональна верифікація: CDV та UVM

Лекційний матеріал (3 год.)

Функціональна верифікація цифрових схем: симуляційна та формальна верифікація. Написання тестбенчів мовою SystemVerilog. Coverage-Driven Verification (CDV): функціональне покриття, покриття станів автомата, перехресне покриття. Universal Verification Methodology (UVM): архітектура, компоненти (agent, driver, monitor, scoreboard, coverage collector). Рандомізоване тестування з обмеженнями (constrained-random). Порівняння CDV та формальної верифікації (equivalence checking, model checking).

Лабораторна (2 год.) та CPC (8 год.)

Лабораторна: Розробка UVM-тестбенча для простого цифрового пристрою (FIFO або FSM) у ModelSim/Questa: реалізація driver, monitor, scoreboard; аналіз метрик функціонального покриття.

CPC: Опрацювання: Артюхов В. Г. та ін. «Технології комп'ютерного проектування. Комп'ютерний практикум» — КПІ ім. Ігоря Сікорського, 2022 — розділ «Верифікація»; підготовка аналітичної записки «UVM у промисловій верифікації: методологія та типові пастки».

- ❑ **Очікуваний результат:** студент вміє розробляти CDV-тестбенчи та базові UVM-компоненти, аналізувати метрики функціонального покриття для цифрових КС.

Надійність та метрики якості КС

Тема 4.3 — Надійність та відмовостійкість (Лекції, 3 год.)

Поняття надійності КС: відмова, помилка, несправність (fault-error-failure). Кількісні показники надійності: MTTF, MTBF, MTTR, доступність (availability). Методи аналізу надійності: FMEA (Failure Mode and Effects Analysis), FTA (Fault Tree Analysis). Методи підвищення надійності: апаратне резервування (TMR, cold/hot standby), програмне резервування, watchdog. Стандарт функціональної безпеки IEC 61508, рівні SIL.

Лабораторна (2 год.): Проведення FMEA-аналізу для модельної КС: складання таблиці видів відмов, оцінка критичності, розробка заходів з підвищення надійності.

СРС (8 год.): Опрацювання: Шульга В. В. та ін. «Інформаційна безпека комп'ютерних мереж та систем» — Харків : ХНУПС, 2023 — розділ «Надійність систем»; підготовка реферату «Функціональна безпека IEC 61508: вимоги та шляхи досягнення рівнів SIL».

Тема 4.4 — Метрики якості КС (Лекції, 3 год.)

Стандарт ISO/IEC 25010 (SQuaRE): модель якості продукту — функціональна придатність, продуктивність, сумісність, зручність використання, надійність, захищеність, супроводжуваність, переносимість. Метрики продуктивності: пропускна здатність, час відгуку, використання ресурсів. Оцінювання якості архітектурних рішень: технічний борг, складність, зв'язність/зчепленість модулів. Інструменти аналізу якості: SonarQube, CAST AIP.

Лабораторна (2 год.): Оцінювання якості власного проекту КС за ISO/IEC 25010: розробка чеклисту, проведення огляду архітектури, виявлення та документування технічного боргу.

СРС (8 год.): Підготовка аналітичної записки «Метрики якості ПЗ у проектуванні КС: практика застосування SonarQube»; формування підсумкового звіту про якість власного проекту.

Сучасні технології та практика проектування КС

Тема 5.1

Безпека комп'ютерних систем у проектуванні: Secure by Design, загрози, моделювання загроз (STRIDE, DREAD). Захист апаратного та програмного рівнів.

Тема 5.2

IoT та вбудовані КС: архітектурні особливості, протоколи (MQTT, CoAP, Zigbee), платформи (Arduino, ESP32, STM32).

Тема 5.3

Штучний інтелект у проектуванні КС: ML-акселератори, нейроморфні системи, AutoML. AI-assisted design (Copilot, Synopsys DSO.ai).

Тема 5.4

Комплексний курсовий проект. Захист проекту КС: презентація, рецензування, підсумкова оцінка компетентностей.

Безпека комп'ютерних систем у процесі проектування

Лекційний матеріал (3 год.)

Парадигма «Security by Design»: принципи, рівні захисту. Класифікація загроз: програмні (CWE Top 25), апаратні (side-channel attacks, ін'єкції несправностей). Моделювання загроз: методологія STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege), метод оцінки DREAD. Криптографічні засоби на рівні апаратного забезпечення: TPM, Secure Enclave, HSM. Захист вбудованих систем: Secure Boot, Code Signing. Нормативна база: ДСТУ EN 62443.

Ключові питання

- Принципи Secure by Design та їх практичне застосування
- Методологія STRIDE у моделюванні загроз
- Апаратні механізми безпеки: TPM та Secure Enclave
- Side-channel атаки та методи протидії
- Стандарт ДСТУ EN 62443 для промислових КС

Лабораторна робота (2 год.)

Завдання 1. Побудова діаграми загроз (Threat Model) для тестової КС методом STRIDE: ідентифікація активів, точок входу, загроз та заходів протидії у Microsoft Threat Modeling Tool.

Завдання 2. Аналіз архітектури спроектованої КС на наявність вразливостей; формування переліку рекомендацій із безпеки.

Завдання 3. Тестування за ключовими термінами теми (10 питань).

Самостійна робота (8 год.)

- Опрацювання: Шульга В. В. та ін. «Інформаційна безпека комп'ютерних мереж та систем» — Харків : ХНУПС, 2023 — розділ «Моделювання загроз»
- Підготовка реферату: «Апаратна безпека в сучасних КС: TPM 2.0, Secure Enclave та їх роль у захисті даних»
- Розробка моделі загроз для власного проекту КС



Очікуваний результат: студент вміє проводити моделювання загроз методом STRIDE, визначати механізми захисту на апаратному та програмному рівнях.

IoT та вбудовані комп'ютерні системи



Апаратна платформа IoT

Мікроконтролери (STM32, ESP32, nRF52) та одноплатні ПК (Raspberry Pi). Вибір за критеріями продуктивності та споживання.



Протоколи комунікації

Провідні: I2C, SPI, UART, CAN, Modbus. Бездротові: Wi-Fi, BLE, Zigbee, LoRaWAN, NB-IoT. MQTT та CoAP для M2M.



Хмарна інтеграція

Платформи IoT: AWS IoT Core, Azure IoT Hub, Google Cloud IoT. Edge Computing: обробка даних на краю мережі.

Лекційний матеріал (3 год.)

Архітектура IoT-систем: рівні (edge, fog, cloud). Вибір апаратної платформи. Протоколи зв'язку та їх порівняння. Операційні системи для IoT: FreeRTOS, Zephyr, Mbed OS. Управління живленням у вбудованих КС. Безпека IoT: OWASP IoT Top 10. Хмарні IoT-платформи та їх архітектура.

Лабораторна (2 год.) та СРС (8 год.)

Лабораторна: Проектування та реалізація прототипу IoT-вузла: програмування ESP32 (FreeRTOS), підключення сенсора, передача даних по MQTT до брокера (Mosquitto), візуалізація у Node-RED.

СРС: Опрацювання: Авраменко В. С., Авраменко А. С. «Проектування інформаційних систем» — Харків, 2023 — розділ «Розподілені та IoT-системи»; підготовка аналітичної записки «LoRaWAN для промислового IoT: архітектура, переваги та обмеження».

AI у проектуванні КС та захист курсового проекту

Тема 5.3 — ШІ у проектуванні КС (Лекції, 3 год.)

Штучний інтелект у циклі проектування КС: AI-assisted EDA (Synopsys DSO.ai, Cadence Cerebrus), автоматична оптимізація Place & Route. ML-акселератори: архітектура NPU, Google TPU, NVIDIA Tensor Core. Нейроморфні обчислення: принципи, приклади (Intel Loihi). AutoML: автоматичний підбір архітектури нейронних мереж. Large Language Models у генерації та рецензуванні коду (GitHub Copilot, ChatGPT). Перспективи AI-асистованого проектування.

Лабораторна (2 год.): Дослідження ML-акселератора: розгортання TensorFlow Lite моделі на вбудованій платформі (ESP32 або Raspberry Pi); вимірювання продуктивності інференції; порівняння з CPU-виконанням.

СРС (9 год.): Опрацювання наукових статей з IEEE Xplore (2023–2024) щодо AI-assisted EDA; підготовка аналітичної записки «Штучний інтелект в EDA: поточний стан та перспективи для проектування КС».

Тема 5.4 — Курсовий проект (Захист)

Комплексний курсовий проект є підсумком вивчення дисципліни та охоплює повний цикл проектування КС: від аналізу вимог до верифікації. Студент самостійно обирає тему відповідно до переліку, затвердженого кафедрою.

Склад проекту:

- Технічне завдання (відповідно до ДСТУ)
- Специфікація вимог (SRS)
- Архітектурний опис (UML-діаграми, опис рішень)
- Реалізація або симуляційна модель ключового компонента
- План і результати верифікації (тест-звіт)
- Аналіз якості та надійності (FMEA або метрики ISO 25010)
- Пояснювальна записка та презентація

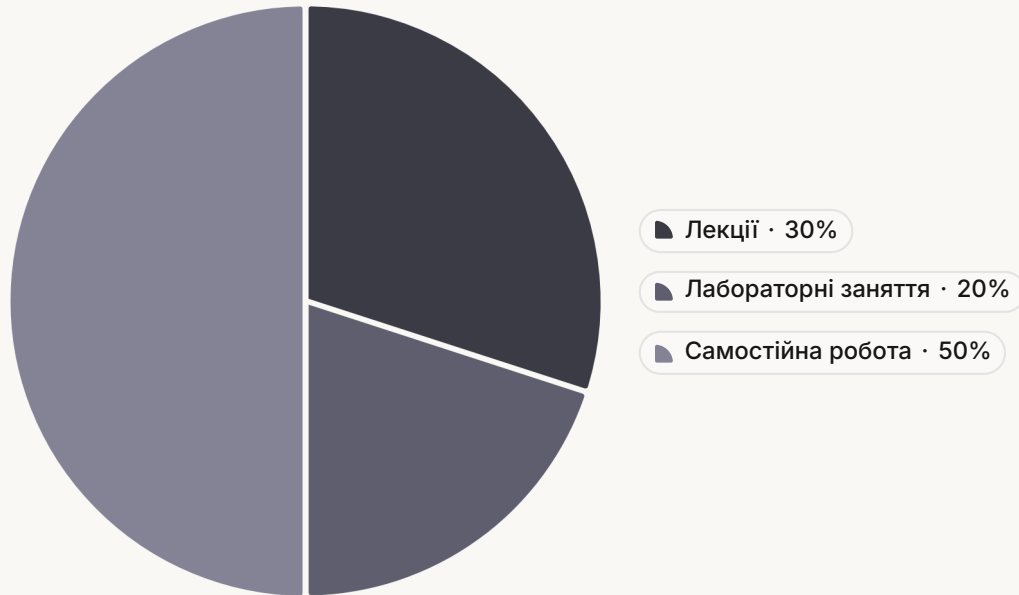
Захист (на підсумковому занятті): 10-хвилинна презентація + питання від комісії + взаємне рецензування проектів однокурсників.

Розподіл навчального часу — 150 годин

Модуль	Тема	Лекції (год.)	Лабораторні (год.)	СРС (год.)
Модуль 1	Тема 1.1 — Вступ. Класифікація КС	3	2	8
	Тема 1.2 — Методології проектування	3	2	8
	Тема 1.3 — Стандарти та документування	3	2	8
	Тема 1.4 — Аналіз вимог. ТЗ	3	2	8
Модуль 2	Тема 2.1 — Архітектурні стилі та патерни	3	2	8
	Тема 2.2 — Апаратна підсистема	3	2	8
	Тема 2.3 — Програмно-апаратний стек	3	2	8
	Тема 2.4 — Розподілені системи	3	2	8
Модуль 3	Тема 3.1 — CASE-засоби	3	2	8
	Тема 3.2 — САПР та EDA	3	2	8
	Тема 3.3 — Моделювання КС	3	2	8
	Тема 3.4 — Генерація коду та синтез	3	2	7
Модуль 4	Тема 4.1 — Верифікація та тестування	3	2	8
	Тема 4.2 — CDV та UVM	3	2	8
	Тема 4.3 — Надійність та відмовостійкість	3	2	8
	Тема 4.4 — Метрики якості КС	3	2	8
Модуль 5	Тема 5.1 — Безпека КС	3	2	8
	Тема 5.2 — IoT та вбудовані системи	3	2	8
	Тема 5.3 — ШІ у проектуванні КС	3	2	9
	Тема 5.4 — Курсовий проект (Захист)	—	2	—
Разом	20 тем / 5 модулів	45	30	75

 **Загальний обсяг: 150 годин = 5 кредитів ЄКТС.** Розподіл є орієнтовним і може коригуватися відповідно до специфіки навчального процесу.

Структура навчального часу



Коментар до розподілу

Загальний обсяг — **150 годин (5 кредитів ЄКТС)** — розподілено між трьома формами роботи для забезпечення балансу між теорією та практикою.

Лекції — 45 год. (30%)

Системне подання теоретичного матеріалу з використанням наочних схем, UML-діаграм, демонстрацій роботи у CASE/EDA-засобах та прикладів з реальних промислових проектів.

Лабораторні — 30 год. (20%)

Практична робота з CASE-засобами (Enterprise Architect), EDA-інструментами (Quartus II, ModelSim), вбудованими платформами (ESP32, STM32) та засобами CI/CD.

Самостійна робота — 75 год. (50%)

Поглиблене вивчення, підготовка завдань, виконання курсового проекту та робота з технічною документацією. Самостійна робота становить **50% обсягу дисципліни**.

Самостійна робота студентів — Детальний опис



Опрацювання літератури

Поглиблене вивчення теоретичного матеріалу за навчальними посібниками з проектування КС (Артюхов, Ізмайлова, Авраменко, Шульга та ін.), стандартами ISO/IEC, IEEE, ДСТУ, науковими статтями у фахових виданнях. Студент конспектує ключові положення, виділяє головні концепції та порівнює підходи різних авторів.



Робота з інформаційними ресурсами

Опрацювання офіційних ресурсів: IEEE Xplore (ieeexplore.ieee.org), документація Intel Quartus (intel.com/quartus), Sparx EA (sparxsystems.com), НБУВ (nbuv.gov.ua). Аналіз наукових статей та пошук актуальної інформації для навчальних завдань.



Індивідуальні завдання

Виконання аналітичних завдань (порівняльний аналіз архітектур, характеристика CASE-засобів, аналіз стандартів); творчих завдань — реферати, аналітичні записки, міні-дослідження з актуальних питань проектування КС та новітніх технологій.



Курсовий проект

Виконання комплексного курсового проекту: розробка ТЗ, SRS, архітектурного опису, симуляційної або апаратної реалізації компонента, плану верифікації та пояснювальної записки. Підготовка до захисту: оформлення презентації та рецензія на проект однокурсника.

Форми контролю знань

1

Поточний контроль

Завдання: тестові та аналітичні вправи за темами змістовного модуля.

Оцінюються регулярність виконання лабораторних робіт, якість звітів та рівень засвоєння матеріалу. Поточний контроль формує накопичувальну оцінку протягом семестру.

Проводиться після кожної теми у формі тестування (10–15 питань) або захисту лабораторної роботи.

2

Модульний контроль

Модульна контрольна робота — комплексне завдання або захист розробленої моделі/документа, що охоплює теми відповідного змістовного модуля (теоретичні питання, аналіз архітектурного рішення, практичне завдання). Оцінюються повнота відповіді, обґрунтованість рішень.

Проводиться після завершення кожного з 5-ти змістовних модулів.

3

Підсумковий контроль

Захист курсового проекту та іспит відповідно до навчального плану. Форма проведення — захист проекту (презентація + питання) та письмовий іспит (тестування + відкриті питання). Підсумкова оцінка враховує результати поточного, модульного та підсумкового контролю: **100-бальна система** з переведенням у національну шкалу та ЄКТС (A, B, C, D, E, FX, F).

Методи викладання

Викладання дисципліни «Технології проектування комп'ютерних систем» поєднує класичні академічні та інтерактивні методи, що забезпечує формування не лише міцної теоретичної бази, а й практичних навичок інженерного проектування, необхідних для майбутньої професійної діяльності у сфері комп'ютерної інженерії.



Лекції

Системне подання теоретичного матеріалу з використанням UML-діаграм, схем архітектур, прикладів із реальних проектів та демонстрацій у CASE/EDA-засобах.



Лабораторні роботи

Практична робота з CASE-засобами, EDA-інструментами та вбудованими платформами; розробка та верифікація компонентів у реальних або симульованих середовищах.



Кейс-стаді

Аналіз реальних проектів: розбір архітектурних рішень відомих систем, дослідження post-mortem звітів провалених проектів та успішних кейсів впровадження.



Технічні дискусії

Обговорення дискусійних питань проектування: порівняння підходів, вибір архітектурних рішень, рецензування проектів однокурсників (peer review).



Проектно-орієнтоване навчання

Виконання комплексного курсового проекту протягом семестру; поетапне ускладнення завдань — від аналізу вимог до верифікації кінцевого рішення.



Моделювання та симуляція

Побудова функціональних, структурних та поведінкових моделей КС; симуляція роботи систем у Simulink, ModelSim, SystemC для верифікації проектних рішень.



Усі методи спрямовані на формування компетентностей, передбачених освітньо-професійною програмою спеціальності 123 «Комп'ютерна інженерія».

Рекомендована література — Підручники та посібники (2022–2024 рр.)

1. Артюхов В. Г. та ін.

Технології комп'ютерного проектування.
Комп'ютерний практикум : навч. посіб. для студентів спеціальності 122 «Комп'ютерні науки» / В. Г. Артюхов, О. А. Бритов, В. Ш. Гіоргізова-Гай та ін. — Київ : КПІ ім. Ігоря Сікорського, 2022. — 218 с.

2. Ізмайлова О. В.

Проектування інформаційних систем : навч. посіб. — Київ : КНУБА, 2023. — 88 с.

3. Шульга В. В., Васишин В. І., Шаповалов О. В. та ін.

Інформаційна безпека комп'ютерних мереж та систем.
Частина 1 : підручник / за заг. ред. С. В. Смелякова. — Харків : ХНУПС, 2023. — 240 с.

4. Пархоменко Д. О., Хижняк І. А., Захарченко І. В. та ін.

Комп'ютерна електроніка : підручник для студентів вищих навчальних закладів. — Харків : ХНУПС, 2022. — 216 с.

5. Авраменко В. С., Авраменко А. С.

Проектування інформаційних систем : навчальний посібник. — Харків : Вид-во ХНЕУ ім. С. Кузнеця, 2023. — 215 с.

6. Доценко С. І.

Організація та системи керування базами даних : навч. посіб. — Харків : УкрДУЗТ, 2023. — 117 с.

7. Новотарський М. А.

Комп'ютерна інженерія : навч. посіб. / КПІ ім. Ігоря Сікорського. — Київ : КПІ ім. Ігоря Сікорського, 2022. — [Електронний ресурс]. — 17.93 Мбайт.

8. Петров О. В., Бороzeneць І. О., Мельничук М. Г. та ін.

Комп'ютерні мережі. Частина 1 : підручник / за заг. ред. І. В. Рубана. — Харків : ХНУПС, 2022. — 256 с.

Додаткова література та наукові статті

9. Жуков І. А., Корочкін О. В.

Паралельні та розподілені обчислення : навч. посіб. — 2-ге вид. — Київ : Корнійчук, 2022. — 284 с. *Рекомендується для тем 2.4, 5.2 (розподілені та паралельні системи).*

10. Нестеренко О. В.

Метод пріоритезації вимог в програмній інженерії // Проблеми програмування. — 2024. — № 2–3. — С. 132–139.
Рекомендується для тем 1.4, 5.4 (аналіз та управління вимогами).

11. Тарнавський Ю. А., Кузьменко І. М.

Організація комп'ютерних мереж : навч. посіб. — Київ : КПІ ім. Ігоря Сікорського, 2022. — 259 с. *Рекомендується для тем 2.3–2.4 (мережеві аспекти КС).*

12. Лупенко С. А., Пасічник В. В., Тиш Є. В.

Комп'ютерна логіка : навч. посіб. для ВНЗ. — Вид. «Магнолія», 2023. — 354 с. *Рекомендується для теми 2.2 (цифрова схемотехніка та FPGA).*

Нормативно-правова база

→ ДСТУ 3973-2000

Система розроблення та поставлення продукції на виробництво. Правила виконання науково-дослідних робіт. Загальні положення. zakon.rada.gov.ua

→ ДСТУ ISO/IEC 25010:2023

Системна та програмна інженерія — Вимоги та оцінювання якості систем і програмного забезпечення (SQuaRE) — Моделі якості системи та програмного забезпечення. uas.org.ua

→ Закон «Про вищу освіту» від 01.07.2014 № 1556-VII

Визначає засади освітньої діяльності у сфері вищої освіти в Україні, стандарти підготовки бакалаврів. zakon.rada.gov.ua

→ Стандарт вищої освіти України: спеціальність 123 «Комп'ютерна інженерія»

Затверджений МОН України. Визначає компетентності та результати навчання для бакалаврського рівня. mon.gov.ua

→ IEEE Std 1471-2000 / ISO/IEC 42010

Recommended Practice for Architectural Description of Software-Intensive Systems. Міжнародний стандарт архітектурного опису КС. ieeexplore.ieee.org

Офіційні ресурси та фахові видання

IEEE Xplore Digital Library

ieeexplore.ieee.org — провідна база наукових статей і стандартів IEEE/ACM з комп'ютерної інженерії, проектування систем та суміжних напрямів.

Intel FPGA Academic Program

intel.com/quartus — офіційна документація, навчальні матеріали та академічні ліцензії на Quartus Prime та ModelSim для університетів.

Sparx Systems Enterprise Architect

sparxsystems.com — документація, навчальні відео та академічні ліцензії на провідний CASE-засіб для UML-моделювання та генерації документації.

МОН України — Освітні стандарти

mon.gov.ua — стандарти вищої освіти, освітні програми, методичні матеріали для підготовки бакалаврів зі спеціальності 123.

Національна бібліотека України ім. В. І. Вернадського

nbuv.gov.ua — електронні ресурси, наукові статті, монографії та навчальні посібники з комп'ютерної інженерії та суміжних дисциплін.

Репозитарій КПІ ім. Ігоря Сікорського

ela.kpi.ua — відкритий доступ до навчальних посібників, конспектів лекцій та методичних матеріалів кафедр КПІ.

Журнал «Проблеми програмування» НАН України

pp.isoftware.kiev.ua — фахове наукове видання з програмної та комп'ютерної інженерії, індексоване у провідних наукометричних базах.

ACM Digital Library

dl.acm.org — електронна бібліотека ACM з фундаментальних та прикладних питань комп'ютерних наук і систем.

Зв'язок дисципліни зі спеціальністю 123 «Комп'ютерна інженерія»

Чому «Технології проектування КС» — ключова дисципліна?

Дисципліна «Технології проектування комп'ютерних систем» є фундаментальною складовою підготовки бакалаврів зі спеціальності 123 «Комп'ютерна інженерія». Вона інтегрує знання з архітектури комп'ютерів, програмування, схемотехніки та системного аналізу в єдиний інженерний цикл проектування.

Сучасний фахівець у сфері комп'ютерних систем має розуміти повний цикл — від формування вимог до верифікації готового рішення — і вільно застосовувати CASE/EDA-засоби, які використовуються в провідних IT- та інженерних компаніях України та світу.

Практичне значення для майбутньої кар'єри

- Проектування апаратних та програмно-апаратних систем для ринку праці (embedded, FPGA, SoC)
- Застосування CASE-засобів та САПР — обов'язкова вимога роботодавців у сфері KI
- Навички верифікації та тестування систем — критичні для IoT, automotive та aerospace
- Розуміння стандартів якості та надійності (ISO 25010, IEC 61508) для сертифікації продуктів
- Вміння аналізувати вимоги та проектувати архітектуру — базова компетентність системного інженера
- Навички роботи в команді над технічним проектом з дотриманням термінів і документування

Ключові компетентності курсу — Підсумок



Курс охоплює повний інженерний цикл проектування комп'ютерних систем — від аналізу вимог до верифікації та забезпечення якості — і формує у студентів цілісну інженерну компетентність, затребувану на сучасному ринку праці.

Бажаємо успіхів у вивченні дисципліни!

150

Годин

Загальний обсяг дисципліни

5

Кредити ЄКТС

Відповідно до стандарту

20

Тем

У 5 змістовних модулях

8

Джерел

Рекомендованої літератури 2022–2024 рр.

«Інженер — це той, хто може робити за один долар те, що будь-хто може зробити за два.» — Артур М. Веллінгтон

Спеціальність 123 · Комп'ютерна інженерія · Бакалавр · 5 кредитів ЄКТС