

*Затверджено  
на засіданні кафедри  
електронних обчислювальних машин  
Протокол № 1 від 29.08.2025 р.*

**Перелік завдань для формування білетів та тестів у ВНС  
з навчальної дисципліни  
«Автоматизоване проектування комп'ютерних систем»**

1. Компоненти сучасних комп'ютерних систем та підходи при їх проектуванні.
2. Рівні проектування комп'ютерних систем.
3. Основні поняття процесу проектування.
4. Визначення CAD.
5. Визначення САМ.
6. Визначення САЕ.
7. Класифікація САПР.
8. Структура САПР.
9. Представлення графічної інформації для САПР.
10. Підходи і методи проектування за допомогою САПР.
11. Завдання синтезу і аналізу при проектуванні.
12. Системний рівень проектування(ESL)
13. Моделювання рівня транзакцій(TLM).
14. Високорівневий синтез(HLS).
15. Сумісне проектування апаратури і програм (hardware/software co-design).
16. Засоби автоматизації системного проектування з використанням SystemVerilog.
17. Засоби автоматизації системного проектування з використанням SystemC
18. Застосування MATLAB для системного проектування.
19. Системна шина AMBA для автоматизації системного проектування.
20. Системне проектування на основі IP-ядер RISC-V.
21. Системне проектування на основі IP-ядер ARM.
22. Потік проектування VLSI рівня топології кристалу.
23. Стили дизайну VLSI рівня топології кристалу.
24. Формування списку з'єднань(NetList) на основі RTL-моделі.
25. Планування частин кристалу(Floorplanning).
26. Розміщення компонентів(Placement).
27. Трасування з'єднань(Routing).
28. Формування масок(Layout Generation).
29. Масштабування елементів топології VLSI та характеристик режиму функціонування.
30. Netlist-формат даних процесу проектування VLSI.
31. EDIF-формат даних процесу проектування VLSI.
32. DEF-формат даних процесу проектування VLSI.
33. LIB-формат даних процесу проектування VLSI.
34. DB-формат даних процесу проектування VLSI.
35. LEF-формат даних процесу проектування VLSI.
36. TF-формат даних процесу проектування VLSI.

37. TLU-формат даних процесу проектування VLSI.
38. IO-формат даних процесу проектування VLSI.
39. SDC-формат даних процесу проектування VLSI.
40. SDF-формат даних процесу проектування VLSI.
41. VCD-формат даних процесу проектування VLSI.
42. SPEF- та SPF-формати даних процесу проектування VLSI.
43. GDS- та GDSII-формати даних процесу проектування VLSI.
44. OASIS-формат даних процесу проектування VLSI.
45. Використання мови TLC для формування сценаріїв автоматизації процесу проектування.
46. Використання мови Perl для формування сценаріїв автоматизації процесу проектування.
47. Використання мови Python для формування сценаріїв автоматизації процесу проектування.
48. Використання SPICE/PSPICE для моделювання при проектуванні.
49. Застосування засобу Genus Synthesis Solution(Cadence) для синтезу на основі RTL-опису проекту.
50. Застосування засобу Design Compiler(Synopsys) для синтезу на основі RTL-опису проекту.
51. Застосування засобу Virtuoso Schematic Editor(Cadence) для внесення змін у проект.
52. Застосування засобу Innovus Implementation System(Cadence) для розміщення компонентів кристалу та трасування.
53. Застосування засобу IC Compiler II(Synopsys) для розміщення компонентів кристалу та трасування.
54. Застосування засобу Virtuoso Layout Suite(Cadence) для роботи з макетом кристалу.
55. Задача функціональної верифікації VLSI.
56. Застосування засобу Incisive Enterprise Verifier(Cadence).
57. Застосування засобу Xcelium Logic Simulator(Cadence).
58. Застосування засобу SpyGlass(Synopsys).
59. Завдання верифікації проекту VLSI (*англ. Signoff*).
60. Застосування засобу Virtuoso ADE Product Suite(Cadence) для верифікації аналогової частини проекту.
61. Застосування засобу Spectre Simulation Platform(Cadence) для верифікації аналогової частини проекту.
62. Концепція DFT(*англ. Design For Testability*).
63. Концепція ATPG(*англ. Automatic Test Pattern Generation*).
64. Засоби TetraMax/TestMax(Cadence).
65. Універсальна методика верифікації(*англ. Universal Verification Methodology, UVM*).
66. Задача формальної верифікації VLSI.
67. Засіб Conformal Equivalence Checker(Cadence).
68. Форми статичного часового аналізу VLSI.
69. Засіб Tempus Timing Signoff Solution(Cadence).
70. Засіб Primetime-SI(Synopsys) для відстеження цілісності сигналів(*англ. Signal Integrity*).

71. Аналіз спаду напруги(*англ. IR-Drop*) на ланках проекту VLSI.
72. Засіб Voltus IC Power Integrity Solution(Cadence).
73. Засіб PrimePower(Synopsys).
74. Фізична верифікація VLSI.
75. Перевірка правил проектування(*англ. Design Rule Checking*) та засіб Calibre nmDRC(Siemens, Mentor Graphics).
76. Відповідність макету схемі(*англ. Layout Versus Schematic*) та засіб Calibre nmLVS(Siemens, Mentor Graphics).
77. Екстракція негативних побічних ефектів та їх усунення.
78. Формування схеми(*англ. Circuit Extraction*) з топологічної моделі VLSI.
79. Екстракція негативних побічних ефектів(*англ. Parasitic Extraction*) та усунення негативних побічних ефектів проекту VLSI.
80. Засіб Quantus Extraction Solution(Cadence).
81. Проектування для процесу виробництва(*англ. Design for Manufacturing*).
82. Етапи процесу виробництва VLSI.
83. Врахування впливу ефектів процесу травлення(*англ. Effects of Etching Process*) при проектуванні VLSI.
84. Врахування процесу хіміко-механічної планаризації(*англ. Chemical Mechanical Planarization*) при проектуванні VLSI.
85. Врахування особливостей процесу літографії(*англ. Lithography*) при проектуванні VLSI.
86. Врахування варіації ширини металу(*англ. Metal Width Variation*) та варіації товщини металу(*англ. Metal Thickness Variation*) при проектуванні VLSI.
87. Засіб Calibre YieldEnhancer(Siemens, Mentor Graphics).
88. Засіб Calibre YieldAnalyzer(Siemens, Mentor Graphics).
89. Засіб Calibre CMPAnalyzer (Siemens, Mentor Graphics).
90. Засіб Calibre LFD(Siemens, Mentor Graphics).
91. Етапи проектування друкованої плати.
92. Формування бібліотеки елементної бази для проектування друкованої плати.
93. Розміщення компонентів на друкованій платі.
94. Автоматизоване та ручне трасування друкованої плати.
95. Застосування САПР Altium Designer для проектування друкованої плати.
96. Підходи до автоматизації проектування прикладного програмного забезпечення комп'ютерних систем.
97. Застосування патерну MVC при автоматизації проектування прикладного програмного забезпечення комп'ютерних систем.
98. Фреймворк Spring та його відмінність від Java EE.
99. Впровадження залежностей та IoC контейнер при застосуванні фреймворку Spring.
100. Фреймворк Spring. Spring Boot.
101. Фреймворк Spring. Spring MVC.
102. Фреймворк Spring. Spring Data.
103. Фреймворк Spring. AOP та RESTful web-сервіси.