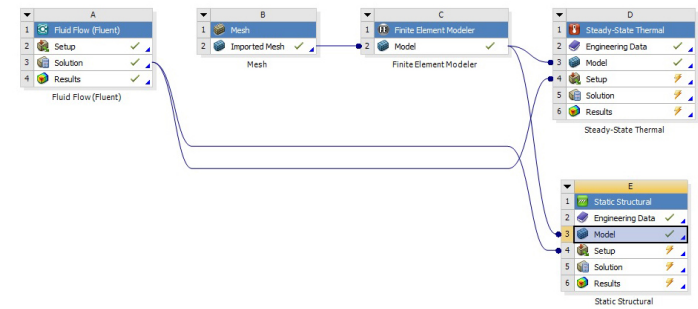


**Т. О. Савчук, О. В. Ольшанська**

## Технологія комп'ютерного проектування



Міністерство освіти і науки України  
Вінницький національний технічний університет

**Т. О. Савчук, О. В. Ольшанська**

# **Технологія комп'ютерного проектування**

Лабораторний практикум

Вінниця  
ВНТУ  
2018

УДК 004.41(075)

C13

Рекомендовано до друку Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 14 від 23.06.2016 р.)

Рецензенти:

**О. Н. Романюк**, доктор технічних наук, професор ВНТУ

**А. Я. Кулик**, доктор технічних наук, професор ВНМУ

**С. М. Захарченко**, кандидат технічних наук, доцент ВНТУ

**Савчук, Т. О.**

C13    Технологія комп'ютерного проектування : лабораторний практикум / Т. О. Савчук, О. В. Ольшанська. – Вінниця : ВНТУ, 2018. – 122 с.

У навчальному посібнику наведено теоретичні відомості за тематикою лабораторних досліджень та порядок їх проведення. Запропоновано варіанти використання різних систем автоматизованого проектування, проведення досліджень за тематикою кожної з лабораторних робіт, варіанти індивідуальних завдань, а також перелік питань для перевірки набутих при виконанні лабораторних робіт теоретичних знань та практичних навичок.

УДК 004.41(075)

© ВНТУ, 2018

## ЗМІСТ

Вступ.....	4
Лабораторна робота № 1. Дослідження складних об'єктів і систем на структурному рівні за допомогою мереж Петрі .....	5
Лабораторна робота № 2. Загальна характеристика систем автоматизованого проектування програмного забезпечення argoUML для UML 2.0 .....	15
Лабораторна робота № 3. Створення нової моделі проекту та побудова діаграм за допомогою обраної САПР .....	26
Лабораторна робота № 4. Створення діаграм компонентів в обраному середовищі САПР .....	35
Лабораторна робота № 5. Створення діаграм розгортання в обраному середовищі САПР .....	43
Лабораторна робота № 6. Загальна характеристика систем автоматизованого проектування. Побудова структурних схем пристроїв в обраній САПР .....	52
Лабораторна робота № 7. Побудова принципів схем пристроїв та моделювання процесів їх функціонування в обраному середовищі САПР кількох вибірок.....	65
Лабораторна робота № 8. Вивчення методів пакування схеми на друковану плату за допомогою системи проектування друкарських плат.....	88
Лабораторна робота № 9. Трасування друкованих плат за допомогою програми трасування обраної САПР .....	113

## ВСТУП

Останнім часом в інженерній практиці обчислювальна техніка широко використовується для аналізу інформації, організації та планування експериментальних досліджень, а також автоматизації проектування складних об'єктів й систем та оброблення результатів. Поява нових моделей процесорів і комплектувальних, версій операційних систем та програмного забезпечення відбувається на тлі постійного ускладнення не тільки окремих фізичних і програмних компонентів, але й концепцій, що лежать в їх основі. Розроблення і вдосконалення інформаційних систем призводить до необхідності підтримання єдиного стилю для різних версій програм при їх постійному доопрацюванні та модифікації.

Ефективність виконання задач проектування складних об'єктів та систем визначається розвитком і швидким поширенням інформаційних технологій, що відкривають нові можливості в підвищенні ефективності виробничих процесів в сфері їх проектування та виробництва. Це приводить не тільки до спрощення процесу проектування, але й до значного зростання обсягів і підвищення складності означеного процесу, що пов'язано з особливостями процесу життєвого циклу найскладнішого об'єкта.

Для вирішення цього завдання останнім часом з'явилися і стали активно розроблятися комп'ютерні системи, що забезпечують комплексну підтримку всіх етапів життєвого циклу складного об'єкта на основі концепції PLM (Product Life Cycle Management). В час еволюційного розвитку технологій комп'ютерного проектування, для якого характерним є функціональне розширення можливостей продуктів та підвищення рівня структуризації інформації, швидких обертів набуває технологія CALS, орієнтована на підтримку продукту на всіх етапах його життєвого циклу.

Не зважаючи на появу нових напрямків в галузі науки і техніки, використання технологій комп'ютерного проектування, розрахунків і управління такими технологіями як автоматизоване проектування – CAD (computer-aided-design), автоматизоване виробництво – CAM (computer-aided-manufacturing), автоматизоване розроблення та конструювання – CAE (computer-aided-engineering) є невід'ємною складовою частиною науково-технічного прогресу. Знайомство з ними, без сумніву, має підвищити рівень фахівців у галузі інформаційних технологій, збагатити інтелектуальну скарбничку сучасного інженера-програміста.

## ЛАБОРАТОРНА РОБОТА № 1

### ДОСЛІДЖЕННЯ СКЛАДНИХ ОБ'ЄКТІВ І СИСТЕМ НА СТРУКТУРНОМУ РІВНІ ЗА ДОПОМОГОЮ МЕРЕЖ ПЕТРІ

Мета: отримати практичні навички побудови моделей складних об'єктів і систем та перевірити їх роботоздатність за допомогою програми Petri NetWork (програма для синтезу та аналізу мереж Петрі).

#### Порядок виконання роботи

1. Ознайомитися з теоретичними відомостями щодо структури та принципів побудови мереж Петрі.

2. Визначитись з основними підходами щодо моделювання, функціонування складних об'єктів і систем за допомогою “дерев досяжності”.

3. Задати мережу Петрі складного об'єкта чи системи відповідно до варіанта завдання.

4. Побудувати “дерево досяжності” отриманої в пункті 3 мережі Петрі та дослідити функціонування заданого складного об'єкта за основними властивостями мережі Петрі.

5. За допомогою програми Petri NetWork задати мережу Петрі, побудованої в пункті 3.

6. За допомогою програми Petri NetWork задати “дерево досяжності”, отримане в пункті 4:

– запустити програму Petri NetWork (рис. 1.1);



Рисунок 1.1 – Вікно запуску програми Petri NetWork

– за допомогою панелі інструментів (рис. 1.2) створити мережу;

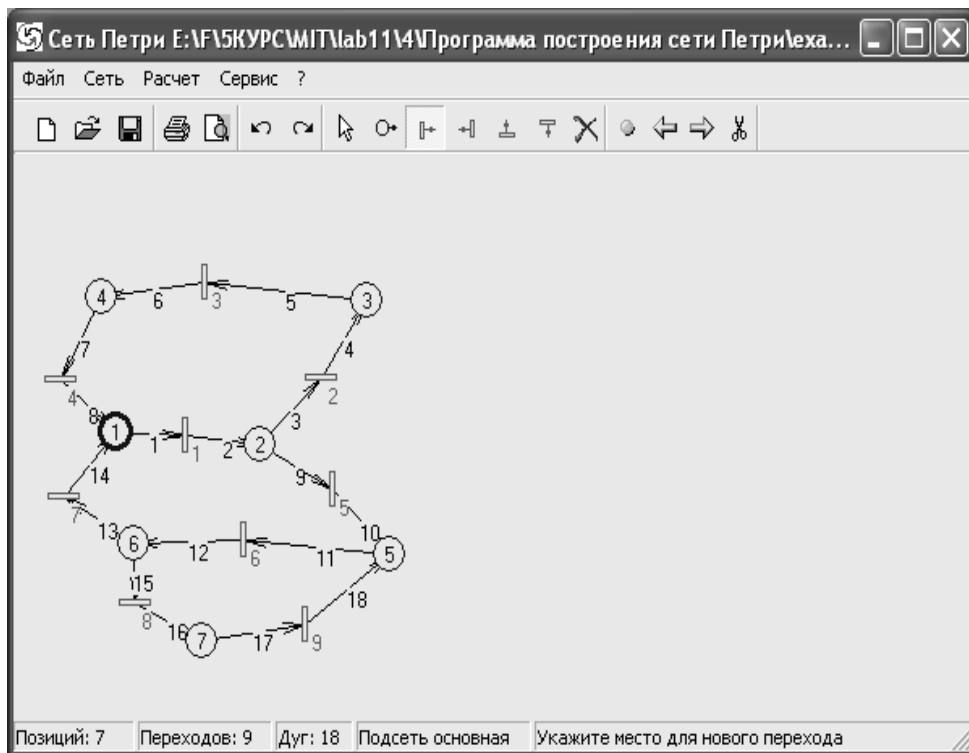


Рисунок 1.2 – Мережа, створена за допомогою панелі інструментів

- якщо дії щодо синтезу мережі незрозумілі, то скористайтеся *Довідкою* в головному меню. В результаті з’явиться вікно “Справка по Petri NetWork” з основними розділами (рис. 1.3);

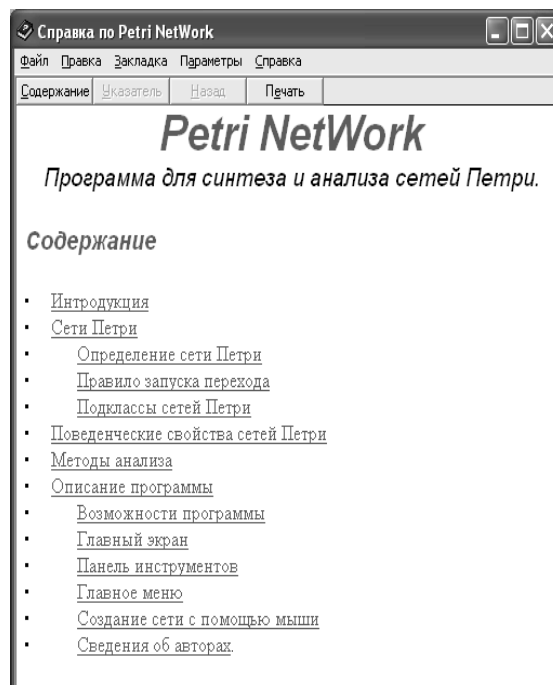


Рисунок 1.3 – Вікно “Справка по Petri NetWork” з основними розділами

- після того, як мережу побудовано, необхідно задати маркування мережі (рис. 1.4);

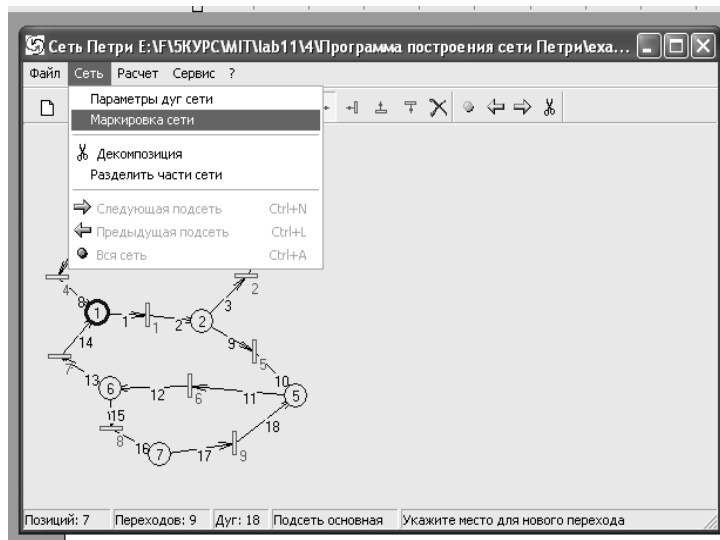


Рисунок 1.4 – Маркована мережа

- після цього, скориставшись пунктом *Расчет* головного меню, дослідити створену мережу (рис. 1.5).

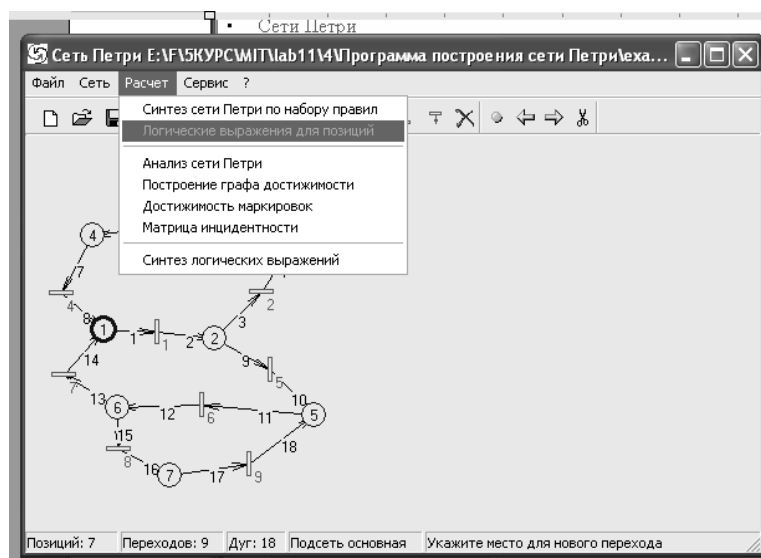


Рисунок 1.5 – Вікно аналізу мережі Петрі

Розрахунок – процедура, що реалізує основні можливості програми:

- *синтез сети Петри по набору правил* – при натисканні буде виведене вікно редагування, у яке можна завантажити набір правил, записаних на диск у вигляді текстового файлу, або створити ці правила заново;
- *логические выражения для позиций* – після генерації мережі за набором правил дозволяє переглянути логічні вирази, що відповідають кожній позиції;
- *анализ сети Петри* – аналізує поточну мережу і виводить її властивості: обмеженість, досяжність, безпека, живучість, збереженість; множину пасивних переходів при заданому початковому



маркуванні, клас мережі (Автомат, Маркований граф, Мережа з Вільним Вибором);

- *построение графа достижимости* – якщо мережа створена і задане початкове маркування, то ця опція дозволяє вивести граф досяжності мережі з усіма можливими маркуваннями. Якщо маркувань мережі (станів) багато, то буде виведене повідомлення про це. У цьому випадку мережу спрощують операцією декомпозиції;
- *достижимость маркировок* – розрахунок кількості переходів, що запускаються, необхідних для досягнення іншого маркування від початкової маркованої позиції;
- *матрица инцидентности* – розрахунок матриці інцидентності;
- *синтез логических выражений* – працює у випадку, коли задане початкове маркування. У результаті роботи видаються логічні вирази для проміжних величин  $R_i$ , а також для керуючих впливів, зазначених у частині <тоді/then> правил керування, і для проміжних величин з урахуванням виразів для позицій мережі (останні вирази знаходяться тільки у випадку генерації мережі за набором правил).

7. Зробити висновки та оформити звіт.

## Теоретичні відомості

### *Формалізація процедур аналізу*

На системному рівні функціонування систем розглядається з інформаційної точки зору при повному абстрагуванні від фізичної суті процесів, які відбуваються в системі. В системах автоматизованого проектування за допомогою математичних методів моделюються процеси введення, оброблення та виведення даних. При цьому застосовуються як аналітичні, так і імітаційні моделі. Можливості отримання аналітичних моделей для складних систем обмежені, тому аналіз найчастіше виконується методом імітаційного моделювання (як найбільш універсальним).

Потрібно зазначити, що системний аналіз неможливо виконати на прикладі окремої задачі. Необхідно отримати описи, узагальнені на весь клас задач. Узагальнення відбувається завдяки врахуванню статистичних закономірностей надходження та оброблення даних. Тому на системному рівні найчастіше для аналізу використовують моделі систем масового обслуговування (СМО) та апарат мереж Петрі.

### *Мережі Петрі*

Мережею Петрі є четвірка:  $P^* = \{T, P, I(t), U(t)\}$ ,

де  $T$  – множина умов, що характеризує можливості зміни системи;

$P$  – множина позицій, що відображає можливі події в результаті впливу умов;

$I(t)$  – вхідна множина, функція відображення переходів-умов у можливі позиції-події;

$U(t)$  – вихідна функція, що показує чи є відображення переходів-умов в множину функцій позицій-подій.

Мережа Петрі, що містить в собі чотири можливих події за чотирьох наявних умов:

$$P = \{P_1, P_2, P_3, P_5\};$$

$$T = \{t_1, t_2, t_3, t_4\};$$

$$I(t_1) = \{P_1\};$$

$$I(t_2) = \{P_1\};$$

$$I(t_3) = \{P_2, P_3\};$$

$$I(t_4) = \{P_3\};$$

$$U(t_1) = \{P_1, P_2\};$$

$$U(t_2) = \{P_2, P_3\};$$

$$U(t_3) = \{P_3\};$$

$$U(t_4) = \{P_5\}.$$

Входи і виходи переходів є комплектами позицій. Комплект є узагальненням множини, яка охоплює елементи, що можуть багаторазово повторюватись. Використання комплектів, а не множин для входів і виходів переходу дозволяє позиції бути кратним входом або кратним виходом переходу.

З метою пояснення основних понять мережі Петрі скористаємось її графічним поданням.

Приклад графічного подання мережі Петрі наведений на рисунку 1.6.

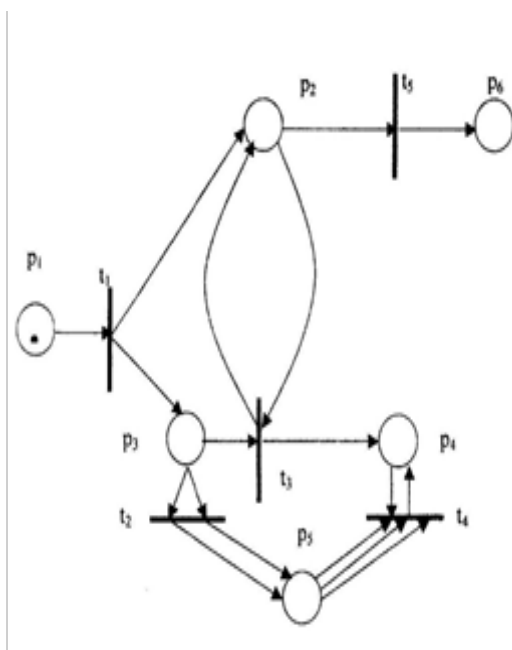


Рисунок 1.6 – Графічне подання мережі Петрі

Опис мережі Петрі, зображеної на рисунку 1.6:

$$\begin{aligned}C &= (P, T, I, O); \\P &= \{p_1, p_2, p_3, p_4, p_5, p_6\}; \\T &= \{t_1, t_2, t_3, t_4, t_5\}; \\I(t_1) &= \{p_1\}; \\I(t_2) &= \{p_3\}; \\I(t_3) &= \{p_2, p_3\}; \\I(t_4) &= \{p_4, p_5, p_5\}; \\I(t_5) &= \{p_2\}; \\O(t_1) &= \{p_2, p_3\}; \\O(t_2) &= \{p_3, p_5, p_5\}; \\O(t_3) &= \{p_2, p_4\}; \\O(t_4) &= \{p_4\}; \\O(t_5) &= \{p_6\}.\end{aligned}$$

Як видно з рисунка 1.6, теоретично-графовим поданням мережі Петрі є дводольний орієнтований мультиграф.

Структура мережі Петрі – це сукупність позицій та переходів. Відповідно до цього граф мережі Петрі має два типи вузлів. Коло є позицією, а планка – переходом. Орієнтовані дуги (стрілки) з'єднують позиції та переходи, при цьому деякі дуги спрямовані від позицій до переходів, а інші – від переходів до позицій.

Мережа Петрі є мультиграфом, тому що допускається існування кратних дуг від однієї вершини до іншої. Оскільки дуги є спрямованими, то це орієнтований мультиграф. Крім того, вершини графа можна поділити на дві множини ( $P$  і  $T$ ), таким чином, що кожна дуга буде спрямована від елемента однієї множини до елемента іншої множини. Такий граф є дводольним орієнтованим мультиграфом.

За допомогою мереж Петрі моделюються процеси, що подаються у вигляді послідовності подій. Прийнято вважати, що події відбуваються миттєво і в різні моменти часу. Кожній з можливих подій відповідає перехід. Подія відбувається, якщо виконані деякі умови. Кожній умові в мережі Петрі відповідає певна позиція. Виконання умов відображується за допомогою маркерів (фішок), які розташовують у вигляді крапок всередині відповідних позицій. Кількість станів мережі Петрі визначається кількістю можливих маркувань. Маркуванням називається розподіл фішок по позиціях. Так, в мережі Петрі, що наведена на рисунку 1.6, маркування визначається так:  $\{1,0,0, 0,0,0\}$ .

Моделювання процесів за допомогою мережі Петрі відбувається шляхом пересування маркерів (фішок) між позиціями. Пересування відбувається через запуск переходів. Перехід запускається видаленням фішок з його вхідних позицій і утворенням нових фішок у вихідних позиціях даного переходу.

Перехід може запускатися, якщо він є *дозволеним*. Перехід називається дозволеним, якщо в кожній його вхідній позиції кількість фішок принаймні дорівнює кількості дуг з позицій в перехід. Кратні фішки необхідні для кратних вхідних дуг.

Просте подання системи мережею Петрі базується на двох основних поняттях: подіях та умовах. Події – це дії, які мають місце в системі. Виникненням подій управляє стан системи. Стан системи може бути описаний множиною умов. Умова – це предикат або логічний опис стану системи. Умова може приймати значення “істина” або значення “хибно”.

Природно, що події в системі можуть відбуватися. Для того, щоб подія відбулась, необхідно виконання відповідних умов. Ці умови називають передумовами події. Виникнення події може викликати порушення передумов та привести до виникнення інших умов, *післяумов*.

Таким чином, ми підійшли до необхідності розгляду методів аналізу мереж Петрі. Але перед цим потрібно вяснити, які властивості мереж Петрі є найбільш важливими.

*Обмеженість* означає, що кількість фішок в будь-якій позиції не може перевищувати деякого числа  $K$  ( $K$  – обмеженість).

*Безпека* визначається умовою  $K=1$ , тобто кількість фішок в безпечній мережі Петрі не перевищує одиниці. Безпека передбачає можливість досягнення певних маркувань.

*Досяжність*. Аналіз досяжності дозволяє визначити множину маркувань, в які можливі переходи в процесі функціонування мережі. Потрібно зазначити, що до аналізу досяжності певних маркувань зводиться багато задач, зокрема проектування пристроїв для виконання певних функцій.

*Збереженість* – це властивість мережі зберігати (не збільшувати і не зменшувати) кількість фішок в усіх позиціях разом.

*Живучість* (активність) означає, що з будь-якого стану, який може бути досяжним з початкового, можливий перехід в будь-який інший стан. Аналіз живучості дозволяє виявити наявність тупиків, зациклювань, блокувань тощо, наприклад, в процесі передавання повідомлень в обчислювальній мережі або при виконанні паралельних програм в багатопроесорній системі.

Щоб зрозуміти чи правильно мережа Петрі моделює роботу тригера, потрібно провести аналіз мережі. Один з методів полягає в побудові “дерева досяжності”. Воно показує всі маркування, досяжні з початкового маркування. Відзначимо, що на графі початкове маркування відповідає нульовим станам виходів тригера.

### ***Можливості програми для синтезу та побудови мереж Петрі Petri NetWork:***

- побудова мережі Петрі на екрані за допомогою миші;
- синтез мережі по заданому набору правил керування типу <якщо:то:>;

- аналіз властивостей побудованої мережі Петрі (живучість, збереженість, безпека, обмеженість, досяжність, визначення пасивних переходів при заданому початковому маркуванні, розрахунок матриці інцидентності й ін.);
- побудова графа досяжності для даної мережі при заданому початковому маркуванні;
- визначення досяжності одного маркування мережі від іншого і кількості переходів мережі, що запускаються при цьому;
- синтез логічних виразів для допоміжних змінних стану, а також для керуючих впливів (у випадку синтезу мережі за правилами).

### ***Побудова мережі за допомогою миші***

Якщо мережа Петрі задана в графічному вигляді, то вона може бути введена декількома натисненнями миші у відповідних точках поля, де повинні розташовуватися вузли мережі (позиції і переходи). За замовчуванням першим елементом мережі ставиться будь-яка позиція. При наступних натисненнях на поле поперемінно з'являються переходи і позиції, уже пов'язані між собою дугами, тобто генерується ланцюжок <позиція – перехід – позиція>.

Для того, щоб нарисувати іншу дугу чи частину мережі, потрібно обрати відповідну опцію на панелі інструментів. Якщо обрати опцію <Позиція>, то при наступному натисненні на полі з'явиться нова позиція. Якщо обрати опцію <Перехід> з відповідною стрілкою (вправо, вліво, доверху, донизу), то з'явиться відповідним чином спрямований перехід. Також за допомогою миші є можливість переміщувати розставлені позиції і переходи по полю. Для цього потрібно натиснути ліву кнопку миші на обраному об'єкті і, утримуючи її натиснутою, перемістити його на нове місце. При цьому, якщо до об'єкта були проведені дуги, вони також змінять своє положення.

### ***Головне меню***

1. *Файл* – містить стандартний набір опцій:
  - *створити* – створення нової мережі. Якщо до цього на полі вже була створена мережа, то буде виведений запит на її збереження;
  - *відкрити* – дозволяє завантажити раніше збережену мережу;
  - *зберегти* – збереження поточної мережі;
  - *зберегти як :* – збереження з запитом нового імені мережі;
  - *вихід* – вихід із програми.
2. *Розрахунок* – процедури, що реалізують основні можливості програми:
  - *синтез мережі Петрі по набору правил* – при натисканні буде виведене вікно редагування, у яке можна завантажити набір правил, записаних на диск у вигляді текстового файлу, або створити ці правила заново;

- *логічні вирази для позицій* – після генерації мережі по набору правил дозволяє переглянути логічні вирази, що відповідають кожній позиції;
- *аналіз мережі Петрі* – аналізує поточну мережу і виводить її властивості: живучість, збереженість, безпека, обмеженість, досяжність, визначення пасивних переходів при заданому початковому маркуванні, розрахунок матриці інцидентності й клас мережі (Автомат, Маркований граф, Мережа з Вільним Вибором), а також кількість незв'язних одна з одною частин;
- *побудова “дерева досяжності”* – якщо мережа створена і задане початкове маркування, то ця опція дозволяє вивести дерево досяжності мережі з усіма можливими маркуваннями. Якщо маркувань мережі (станів) занадто багато, то буде виведене повідомлення про це. У цьому випадку можна спробувати скористатися декомпозицією мережі;
- *досяжність маркувань* – розрахунок кількості переходів, що запускаються, необхідних для досягнення другого маркування від першої маркованої позиції;
- *матриця інцидентності* – розрахунок матриці інцидентності;
- *синтез логічних виразів* – працює у випадку, якщо задане початкове маркування. У результаті роботи видаються логічні вирази для проміжних величин  $R_i$ , а також для керуючих впливів, зазначених у частині <тоді/then> правил керування, і для проміжних величин з урахуванням виразів для позицій мережі (останні вирази знаходяться тільки у випадку генерації мережі за набором правил).

### 3. Сервіс – сервісні функції:

- *матричні обчислення* – матричний калькулятор;
- *настроювання* – настроювання програми, що дозволяють змінювати інтерфейс програми.

### 4. Підмережі – робота з підмережами, згенерованими при декомпозиції:

- *декомпозиція* – процедура розбиття на складові;
- *розділити частини мережі* – якщо мережа фрагментована на непов'язані між собою мережі, то для спрощення розрахунків і наочності можна замість декомпозиції застосувати цю процедуру;
- *наступна підмережа/попередня підмережа* – переміщення по згенерованих підмережах;
- *уся мережа* – повернення до основної мережі.

### ***Панель інструментів***

На панелі інструментів програми міститься набір опцій, згрупованих за функціональними ознаками.

Призначення груп опцій наведені нижче один по одному, зліва направо:

1. Робота з диском:

- *створити нову мережу* – очищає поле для рисування чи синтезу нової мережі;
- *відкрити* – завантаження мережі Петрі, заздалегідь збереженої на диску. При натисканні опції на екран виводиться діалог відкриття файлу. За замовчуванням розширення файлів даних, у яких зберігаються мережі Петрі, –.dat, але за бажанням воно може бути іншим;
- *зберегти* – збереження поточної мережі у файлі. Якщо мережа раніше не зберігалася в окремому файлі, то буде запитане ім'я нового файлу.

2. Робота з принтером (у демо-версії не працюють):

- *друк* – вивід поточної мережі на принтер;
- *попередній перегляд* – дозволяє переглянути поточну мережу перед друком.

3. UnDo і ReDo – скасування останньої дії і повторення дії.

4. Редагування мережі:

- *вибір об'єкта* – дозволяє вибрати будь-який вузол мережі (позицію чи перехід). Також використовується для переміщення позицій чи переходів по полю з метою кращого розташування мережі;
- *позиція* – при обраній опції наступне натиснення миші на поле породжує нову позицію;
- *переходи (вправо, вліво, догори, донизу)* – аналогічно опції “позиція”, але натиснення миші породжує новий перехід, спрямований відповідно;
- *видалення* – видалення будь-якого вузла (вузлів) мережі. При видаленні вузла разом з ним видаляються пов'язані з ним дуги.

5. Робота з підмережами, отриманими при декомпозиції:

- *уся мережа* – повернення до вихідної мережі, від якої створені всі підмережі;
- *попередня підмережа/наступна підмережа* – перегляд підмереж;
- *декомпозиція* – проведення процедури декомпозиції над поточною мережею. У результаті декомпозиції створюються підмережі.

### Контрольні питання

1. Дайте означення мережі Петрі.
2. Як відбувається виконання мережі Петрі?
3. Дайте характеристику основним властивостям мереж Петрі та методам їх аналізу.
4. Як за допомогою мереж Петрі моделюють процеси в складних системах?

5. З чим пов'язане виникнення великої кількості різновидів мереж Петрі?
6. В чому полягає алгоритм побудови “дерева досяжності”?
7. Для вирішення яких задач застосовують “дерево досяжності”?

### **Рекомендована література**

1. Норенков И. П. Основы теории и проектирования САПР : учебник для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1990. – 335 с.
2. Норенков И. П. Системы автоматизированного проектирования электронной и вычислительной аппаратуры : учебное пособие для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1983. – 272 с.
3. Питерсон Дж. Теория сетей Петри и моделирование систем / Питерсон Дж. ; пер. с англ. – М. : Мир, 1984. – 264 с.
4. Рафикузаман М. Микропроцессоры и машинное проектирование микропроцессорных систем : в 2-х кн. / Рафикузаман М. ; пер. с англ. – М. : Мир, 1988.
5. Роїк О. М. Основи автоматизованого проектування складних об'єктів і систем : навчальний посібник / Роїк О. М., Савчук Т. О., Ткаченко О. М. – Вінниця : ВДТУ, 2001. – 111 с.

## **ЛАБОРАТОРНА РОБОТА № 2**

### **ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ArgoUML ДЛЯ UML 2.0**

Мета: отримати початкові навички роботи в обраній САПР, ознайомитись з призначенням програми, навчитися конфігурувати і налаштувати проекти досліджуваної САПР.

### **Порядок виконання роботи**

1. Ознайомитись з системою автоматизованого проектування, що використовується для проектування програмного забезпечення.
2. Використовуючи інструментарій САПР, виконати дії щодо створення бібліотечних елементів за індивідуальним завданням (табл. 2.1).
3. Розробити алгоритм реалізації функціональних задач проектованого засобу з використанням блочно-ієрархічного підходу та подати його схему.
4. Синтезувати структуру програмного засобу з використанням об'єктно-орієнтованого підходу з визначенням його складових в їх взаємозв'язках.



5. Занести до бібліотеки як розроблені складові та програмний засіб в цілому.

6. Зробити висновки та оформити звіт.

Таблиця 2.1 – Варіанти завдань для створення бібліотечних елементів

Варіант	Завдання
1	2
1	Написати програму, у вікні якої міститься одна опція. В результаті натиснення на опцію змінюється заголовок вікна (текст підпису).
2	Написати програму, у вікні якої міститься одна опція. Як результат натискання мишею на вікні – опція з'являється та зникає.
3	Написати програму, у вікні якої міститься одна опція і поле введення. В результаті натиснення на опцію текст поля введення стирається.
4	Написати програму, у вікні якої міститься одна опція. В результаті натиснення на опцію змінюється текст підпису опції.
5	Написати програму, у вікні якої міститься одна опція і два поля введення. В результаті натиснення на опцію текст першого поля копіюється в друге поле.
6	Написати програму, у вікні якої міститься одна опція і поле введення. В результаті натиснення на опцію змінюється заголовок вікна (текст підпису) на текст, який введено в поле введення.
7	Написати програму, у вікні якої міститься одна опція. В результаті натиснення на опцію вона ховається.
8	Написати програму, у вікні якої міститься один прапорець (CheckBox) і поле введення. В результаті натиснення на прапорець зникає/з'являється поле введення.
9	Написати програму, у вікні якої міститься одна опція. В результаті натиснення на опцію її висота збільшується на 1.
10	Написати програму, у вікні якої міститься одна опція і поле введення. В результаті натиснення на опцію поле введення переміщується на одиницю вгору.

### Теоретичні відомості

ArgoUML – засіб UML моделювання (Unified Modeling Language (UML) – уніфікована мова візуального моделювання бізнес-процесів). ArgoUML є відкритим програмним забезпеченням і розповсюджується під ліцензією BSD. ArgoUML повністю написаний на Java і для роботи йому підходить будь-яка операційна система з встановленою Java 2 JRE або JDK версії 1.4 або вище. Також ArgoUML підтримує специфікацію UML 2.0, яка описує уніфіковану мову моделювання, що використовується, щоб визначити карти або моделі проекту.

ArgoUML є областю середовища орієнтованої розробки, яка забезпечує когнітивну підтримку об'єктно-орієнтованого проектування. ArgoUML забезпечує ті ж функції автоматизації, що і комерційні CASE-інструменти, але він зосереджений на особливостях, які підтримують пізнавальні потреби конструкторів. Ці пізнавальні потреби описують три когнітивних теорії:

- відображення в дії;
- поліпшений дизайн;
- розуміння і вирішення проблем за допомогою технічної підтримки.

У цій САПР також є можливість синхронізувати код і модель елементів, проектувати та генерувати базу даних елементів. Від ваших моделей, ви можете швидко експортувати високоякісну документацію.

Функціональність ArgoUML містить:

- 1) підтримку специфікацій UML 1.3, 1.4, 1.0 XMI, 1.1, 1.2;
- 2) 9 видів діаграм UML (діаграми класів, станів, кооперації, послідовності, діяльності, прецедентів, об'єктів, компонентів, розгортання);
- 3) підтримку OCL для класів;
- 4) генерацію вихідного коду Java, C ++, C # і PHP;
- 5) зворотній інжиніринг з вихідного коду та байткоду Java;
- 6) автоматичну верифікацію моделі UML (дизайн критики).

### ***Опис інтерфейсу програми***

При запуску програми головне вікно програми Ideas Modeler має вигляд, що зображений на рисунку 2.1. Воно складається з таких основних зон:

1. Головне меню (зона 1);
2. Стандартна панель (зона 2);
3. Панель створення діаграм (зона 3);
4. Список усіх створених діаграм в межах даного проекту (зона 4);
5. Панель з вкладками параметрів елементів (зона 5);
6. Панель визначення пріоритетів діаграм (зона 6).

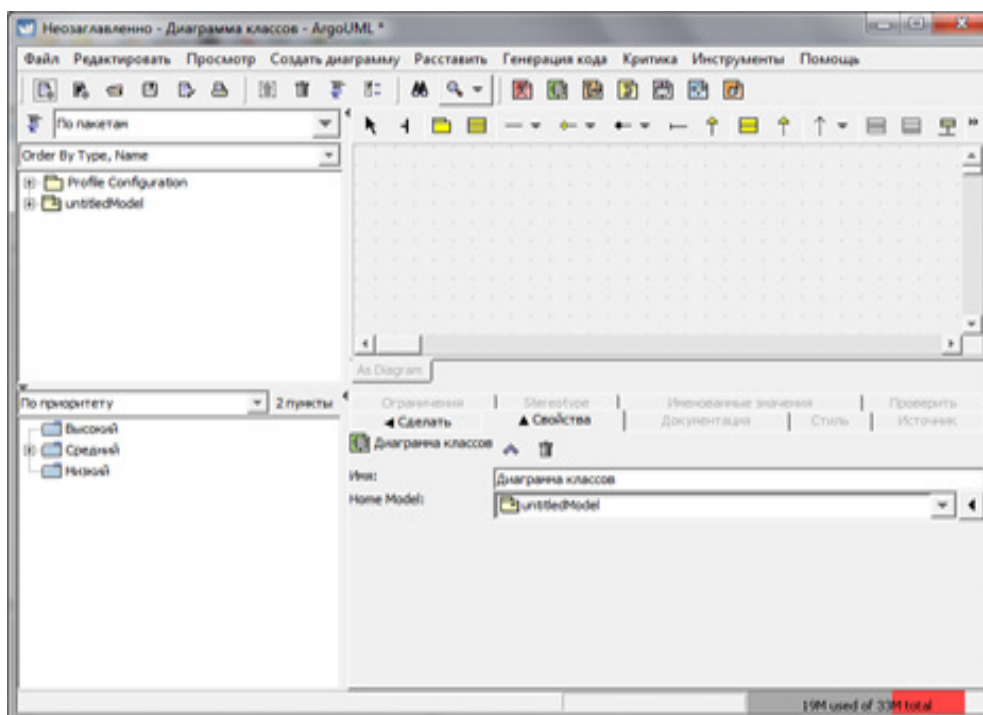


Рисунок 2.1 – Стартове головне вікно програми

Обрання в меню потрібної діаграми змінить стартове головне вікно програми, як зображено на рисунку 2.2. При цьому відбудеться активування певних вкладок панелі параметрів залежно від типу діаграми.

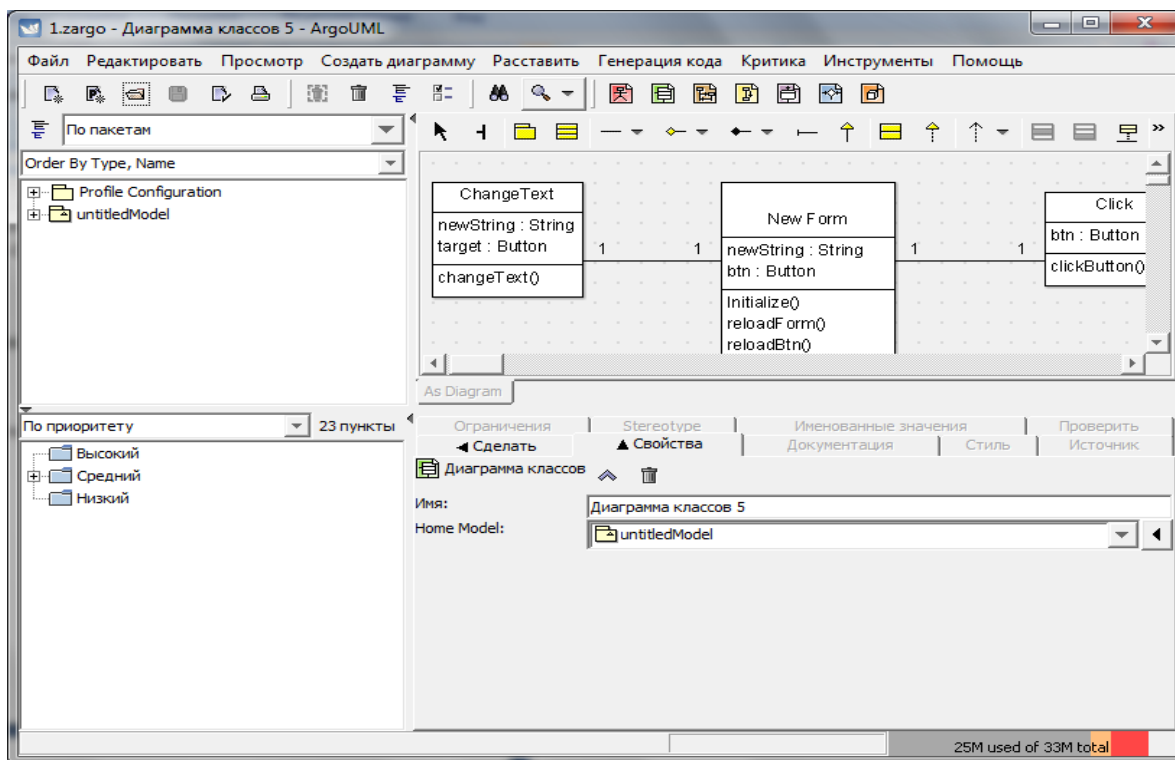


Рисунок 2.2 – Головне вікно програми при створенні діаграми

Головне меню містить опції, що зображені на рисунках 2.3 – 2.10.

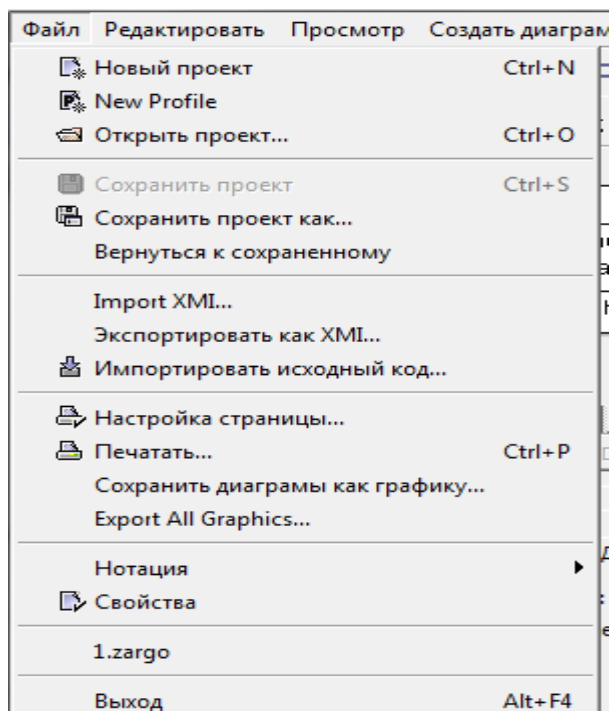


Рисунок 2.3 – Опція “Файл” головного меню

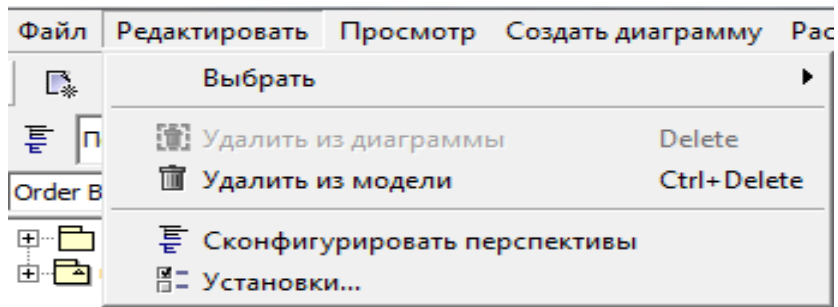


Рисунок 2.4 – Опция “Редактировать” головного меню

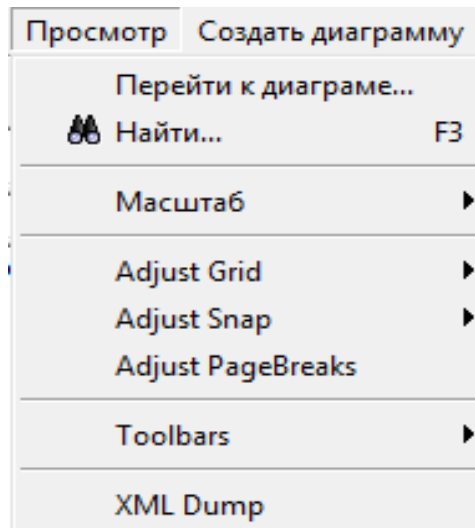


Рисунок 2.5 – Опция “Просмотр” головного меню

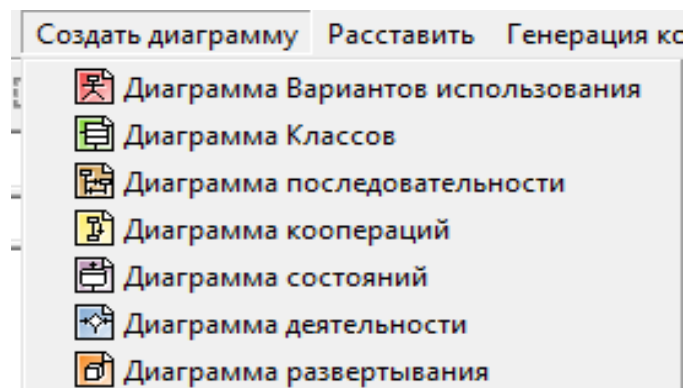


Рисунок 2.6 – Опция “Создать диаграмму” головного меню

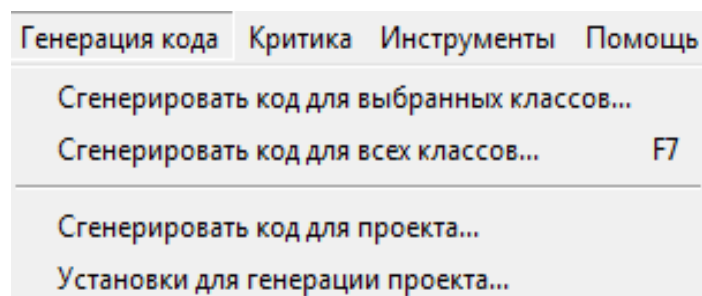


Рисунок 2.7 – Опция “Генерация кода” головного меню

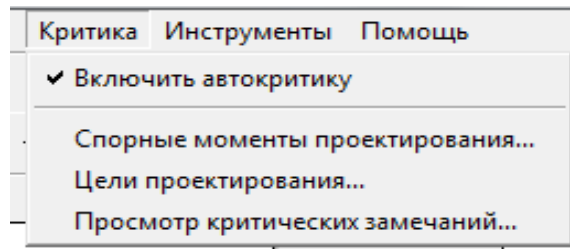


Рисунок 2.8 – Опція “Критика” головного меню

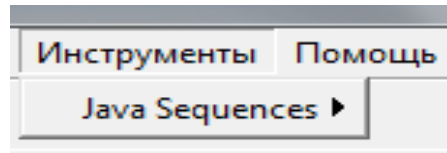


Рисунок 2.9 – Опція “Инструменты” головного меню

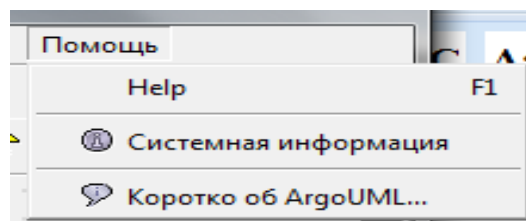


Рисунок 2.10 – Опція “Помощь” головного меню

Стандартна панель має вигляд, як показано на рисунку 2.11 та містить такі опції:

- 1) тип курсору;
- 2) переривання;
- 3) новий пакет;
- 4) новий клас;
- 5) нова асоціація;
- 6) нова композиція;
- 7) новий полюс асоціації;
- 8) нове узагальнення;
- 9) новий інтерфейс;
- 10) нова реалізація;
- 11) нова залежність;
- 12) новий атрибут;
- 13) нова операція;
- 14) новий асоціативний клас;
- 15) новий тип даних;
- 16) новий коментар;
- 17) новий коментар-посилання;
- 18) прямокутник.



Рисунок 2.11 – Вигляд стандартної панелі

Нехай за завданням необхідно написати програму, у вікні якої міститься одна опція. В результаті вибору цієї опції має змінитися її назва.

Бібліотечні елементи в САПР додаються автоматично під час створення діаграм для того, щоб не обтяжувати проект. Потрібно лише вибрати тип діаграми та створити її, а система автоматично створить бібліотечні елементи. Внесені в проект бібліотечні елементи можна переглянути у лівій верхній панелі. Панель буде містити всі розміщені на моделі елементи, а також усі типи створених діаграм (рис. 2.12).

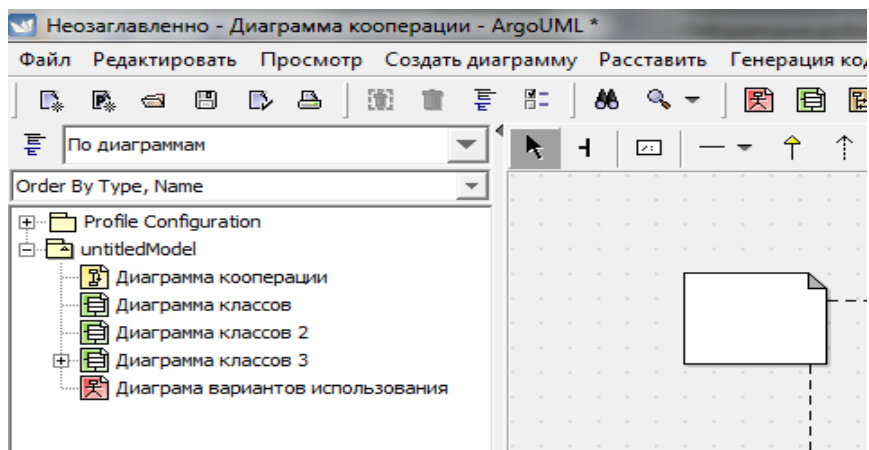


Рисунок 2.12 – Панель відображення бібліотечних елементів

Структуру програмного засобу найбільш повно показує діаграма класів, для створення якої з панелі інструментів потрібно вибрати необхідні компоненти, перенести їх в робочу зону та з'єднати потрібними типами зв'язків з панелі інструментів. Для додавання атрибутів або методів класу необхідно натиснути правою кнопкою миші по доданому класу та у висхідному меню обрати пункт “Добавить” – “Новая операция” або “Новый атрибут” (рис. 2.13). Приклад структури програмного засобу з використанням об'єктно-орієнтованого підходу з визначенням його складових в їх взаємозв'язках наведено на рисунку 2.14.

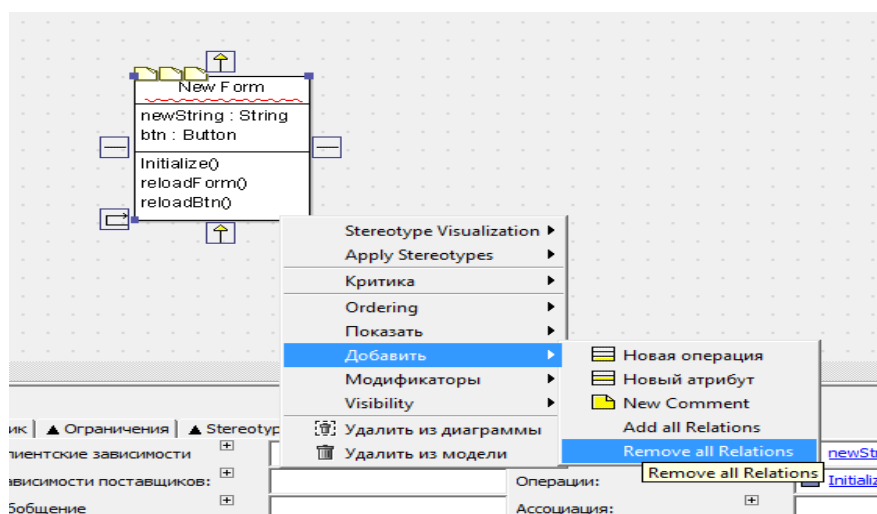


Рисунок 2.13 – Додавання атрибутів та операцій класу

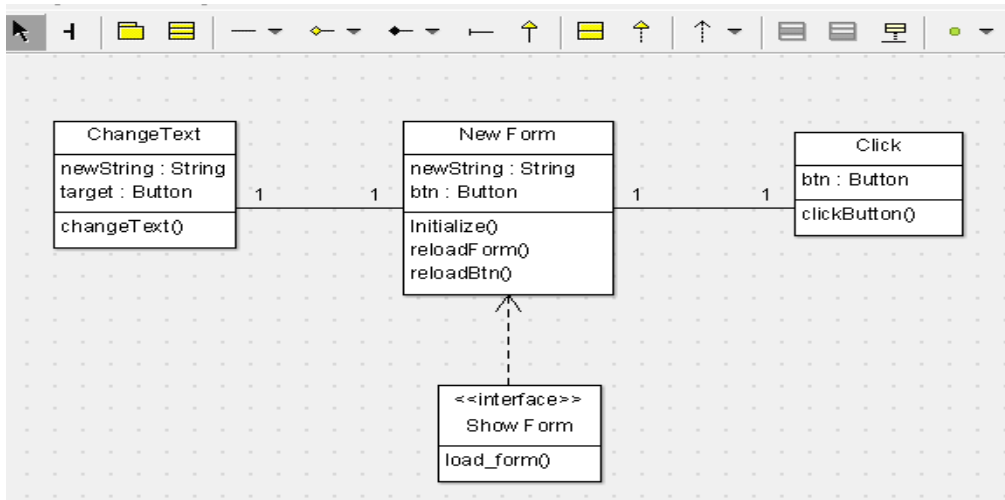


Рисунок 2.14 – Структура програмного засобу з використанням об’єктно-орієнтованого підходу з визначенням його складових в їх взаємозв’язках

Генерація програмного коду виконується шляхом вибору опції “Генерація коду” головного меню ArgoUML. При цьому необхідно обрати тип генерації коду – генерувати код для усіх компонентів моделі або для обраного елемента.

Після цього з’явиться вікно вибору мови програмування, для якої буде згенеровано код моделі (рис. 2.15).

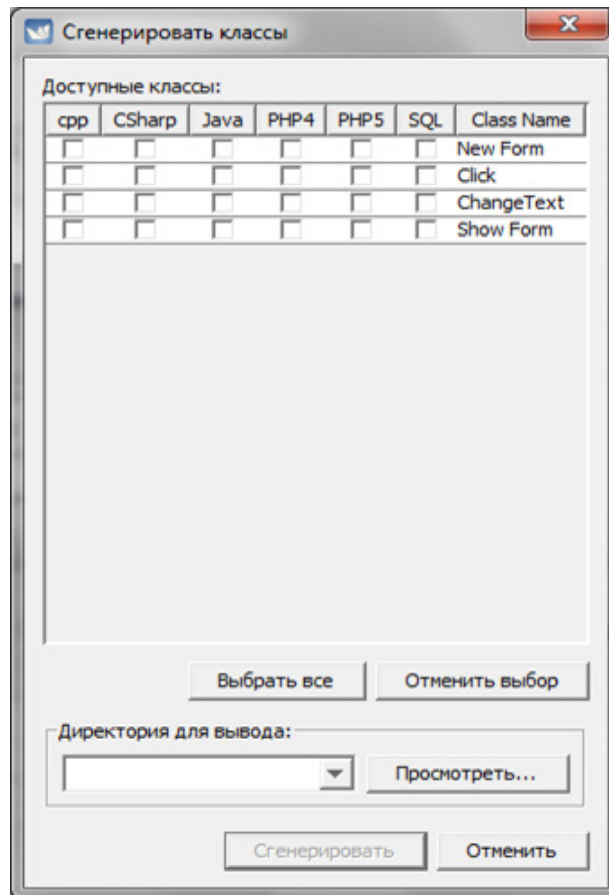


Рисунок 2.15 – Вікно “Сгенерировать классы”

Для правильної генерації коду необхідно вказати директорію, до якої експортуватиметься модель. В іншому випадку код буде згенеровано, але тимчасово збережено у САПР.

Також є можливість обрати будь-яку з шести мов програмування для генерації класів. Потрібно зауважити, що список доступних мов для генерації коду може бути розширений за рахунок завантаження та встановлення розширень САПР для відповідної мови, або, за необхідності, створення розширення власноруч. Для цього є значна кількість довідкової інформації про API-системи, структуру модулів та інші особливості розробки компонентів САПР ArgoUML.

Результат генерації коду можна побачити на одній з встановлених мов програмування у реальному часі у вкладці “Источник” при обраному класі моделі (рис. 2.16).

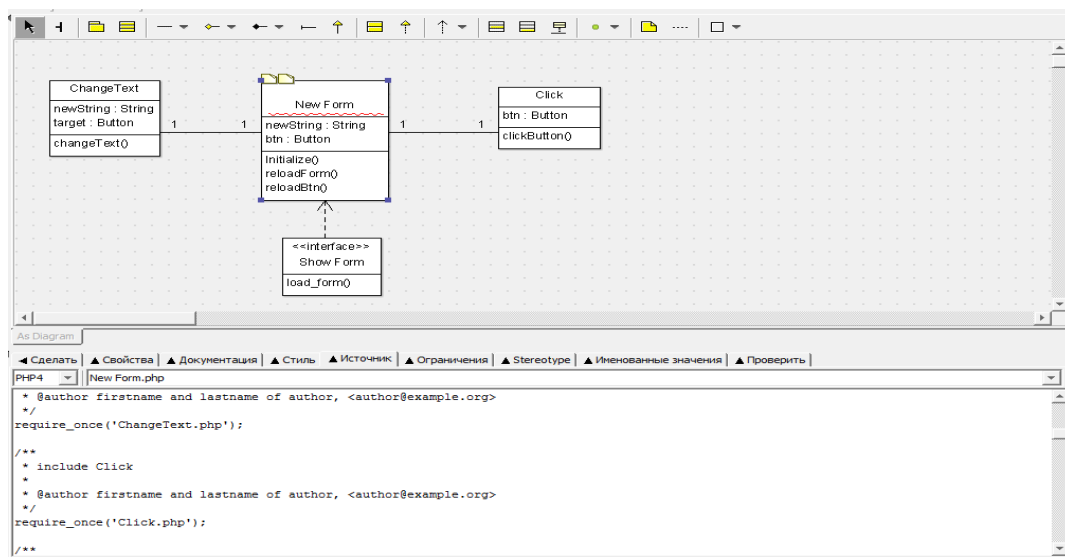


Рисунок 2.16 – Панель згенерованого коду класу “New Form”

Обрану САПР доцільно використовувати для проектування програмного забезпечення, адже вона дозволяє створювати всі необхідні при цьому моделі. Простота використання даної САПР забезпечується інтуїтивно зрозумілим інтерфейсом. ArgoUML є САПР з відкритими вихідними кодами, що забезпечує швидкий розвиток додаткових компонентів, таких як розширення кількості мов програмування для генерації коду, а також швидку технічну підтримку.

До недоліків обраної САПР потрібно віднести те, що вона генерує код для обмеженої кількості мов програмування, встановлених у системі за замовчуванням.

### Контрольні питання

1. ArgoUML є відкритим програмним продуктом?
  - а) так, код програми повністю відкритий;



- б) код частково відкритий для написання розширень;
  - в) ArgoUML є платним продуктом із закритим кодом;
  - г) ArgoUML є безкоштовним продуктом із закритим кодом.
2. На якій мові програмування написано ArgoUML?
- а) C++;
  - б) C;
  - в) C#;
  - г) Java.
3. Яку специфікацію підтримує Argo UML?
- а) UML 0.9;
  - б) UML 1.0;
  - в) UML 1.1;
  - г) UML 2.0.
4. Скільки видів UML-діаграм є доступними у цій САПР?
- а) 3;
  - б) 5;
  - в) 7;
  - г) 9.
5. На якій мові програмування неможлива генерація вихідного коду?
- а) PHP;
  - б) C++;
  - в) Ruby;
  - г) C#.
6. Якого пункту немає у головному меню програми?
- а) файл;
  - б) редагувати;
  - в) вигляд;
  - г) критика.
7. Де можна побачити усі діаграми, що належать до певної моделі?
- а) стандартна панель;
  - б) панель створення діаграм;
  - в) панель складу моделі;
  - г) панель з вкладками параметрів елементів.
8. Яка UML-діаграма найбільш повно відображає структуру програмного засобу?
- а) діаграма прецедентів;
  - б) діаграма пакетів;
  - в) діаграма розгортання;
  - г) діаграма класів.
9. Яким чином можна створити діаграму класів у САПР ArgoUML?
- а) ”Файл” – ”Создать диаграмму классов”;
  - б) відповідна кнопка на стандартній панелі;
  - в) натиснути ”Ctrl” + ”N”;
  - г) ”Создать диаграмму” – ”Диаграмма классов”.

10. Як можна додати метод до певного класу при створенні діаграми класів?

а) натиснути правою кнопкою миші та “Добавить” – “Новая переменная”;

б) меню “Редактировать” та “Добавить” – “Новая операция”;

в) правою кнопкою мишу та “Добавить” – “Новая операция”;

г) меню “Файл” та “Добавить” – “Новый метод”.

11. Згенерувати код побудованої діаграми класів, можна за допомогою:

а) меню “Файл” – “Генерировать код”;

б) правої кнопки миші “Генерация кода”;

в) панелі властивостей моделі, кнопка “Генерация кода”;

г) пункту головного меню “Генерировать код”.

12. Якого типу генерації коду немає у САПР ArgoUML?

а) генерація коду у реальному часі для певного елемента;

б) генерація коду у реальному часі для всіх елементів;

в) генерація коду та його експорт у необхідне місце;

г) генерація коду та його експорт у необхідне місце, лише певного елемента моделі.

13. Яка кількість мов програмування доступна для генерації коду у САПР ArgoUML?

а) 1;

б) 4;

в) 8;

г) 6.

14. У якій вкладці панелі властивостей можна отримувати згенерований код у реальному часі?

а) “Источник”;

б) “Документация”;

в) “Стиль”;

г) “Исключение”.

15. Який елемент відсутній на стандартній панелі?

а) новий блок;

б) нове узагальнення;

в) новий інтерфейс;

г) нова реалізація.

### Рекомендована література

1. Норенков И. П. Основы теории и проектирования САПР : учебник для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1990. – 335 с.

2. Норенков И. П. Системы автоматизированного проектирования электронной и вычислительной аппаратуры : учебное пособие для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1983. – 272 с.

3. Питерсон Дж. Теория сетей Петри и моделирование систем / Питерсон Дж. ; пер. с англ. – М. : Мир, 1984. – 264 с.

4. Рафикузаман М. Микропроцессоры и машинное проектирование микропроцессорных систем : в 2-х кн. / Рафикузаман М. ; пер. с англ. – М. : Мир, 1988.

5. Системы автоматизированного проектирования : учебное пособие для вузов : в 9 кн. / [под ред. И. П. Норенкова]. – М. : Высшая школа, 1986. – 159 с.

6. Системы автоматизированного проектирования в радиоэлектронике : [справочник] / Е. В. Авдеев, А. Т. Еремин, И. П. Норенков, М. И. Песков ; под ред. И. П. Норенкова. – М. : Радио и связь, 1986. – 368 с.

7. Соловьев В. В. Проектирование функциональных узлов цифровых систем на программируемых логических устройствах : учебное пособие для вузов / Соловьев В. В. – М. : ПКООО Бестпринт, 1996. – 252 с.

8. Роїк О. М. Основи автоматизованого проектування складних об'єктів і систем : навчальний посібник / Роїк О. М., Савчук Т. О., Ткаченко О. М. – Вінниця : ВДТУ, 2001. – 111 с.

### **ЛАБОРАТОРНА РОБОТА № 3**

#### **СТВОРЕННЯ МОДЕЛІ ПРОЕКТУ ТА ПОБУДОВА ДІАГРАМ ЗА ДОПОМОГОЮ ОБРАНОЇ САПР**

Мета: вивчення методів побудови діаграм та генерації готового програмного коду в обраній САПР.

#### **Порядок виконання роботи**

1. Ознайомитися з основними функціями обраної САПР, що можуть бути використаними для проектування програм.

2. Оволодіти методикою і способами створення діаграм класів в обраній САПР.

3. Формалізувати задачу, враховуючи об'єктно-орієнтований підхід щодо проектування прикладних додатків, побудувати діаграму класів додатка за індивідуальним завданням (табл. 3.1).

4. До створених класів додати необхідні методи та поля, що потрібні для вирішення поставленого завдання.

5. Використовуючи засоби обраної САПР, отримати згенерований програмний код.

6. Доповнити отриманий програмний код з урахуванням індивідуального завдання.

7. Зробити висновки.

8. Оформити звіт.

Таблиця 3.1 – Варіанти завдання для створення компонентів

Варіант	Завдання
1	2
1	Область MyPanel, яка масштабується в обох напрямках. Один з розмірів зменшується до 0 або виходить за межі (змінює напрям масштабування на протилежний).
2	Прапорець MyCheckBox, який сам встановлюється та скидається з певною частотою.
3	Текстовий елемент MyLabel, який переміщується з певною швидкістю в горизонтальному напрямі та змінює напрям руху на протилежний, коли доходить до краю.
4	Текстове поле MyEdit, яке самостійно вводить у себе заданий текст з певною швидкістю.
5	Текстовий елемент MyLabel, який змінює свій колір з певною частотою (використати не менше 5 кольорів).
6	Клавіша MyButton, яка переміщується з певною швидкістю у вертикальному напрямі та змінює напрям руху на протилежний, коли доходить до краю.

### Теоретичні відомості

В САПР ArgoUML для проектування програм реалізовані такі основні функції:

- створення проектів (для моделювання масштабних систем);
- створення UML-діаграм різних типів;
- генерація програмного коду.

#### **Створення проектів**

Для створення нового проекту потрібно натиснути в головному меню програми “Файл” → “Новый проект” (Ctrl+N). Після цього необхідно обрати та додати до моделі необхідний тип діаграми у меню “Создание диаграммы”.

#### **Створення UML-діаграм різних типів**

Діаграми створюються за допомогою панелі “Создание диаграммы”. Елементи моделі, які вже знаходяться в моделі, але не на діаграмі, можуть бути додані до діаграми, обравши елемент моделі в провіднику з використанням опції “Добавить” з низхідного меню над цим елементом моделі та натисненням в позиції потрібного місця на діаграмі.

Інші необхідні UML-елементи можна знайти на панелі інструментів, яка містить загальні графічні об’єкти (прямокутники, круги, лінії, багатокутники, криві, текст) та додаткову інформацію про діаграми та їх елементи.

Щоб створити зв'язок між елементами, необхідно обрати потрібний тип зв'язку в панелі інструментів та провести його між заданими елементами.

### **Генерація програмного коду**

Після створення діаграми класів програми, що розробляється, є можливість згенерувати код для подальшого перенесення його до основного засобу розробки.

Для генерації коду необхідно:

1. Виділити необхідні класи для генерації коду, як показано на рисунку 3.1.

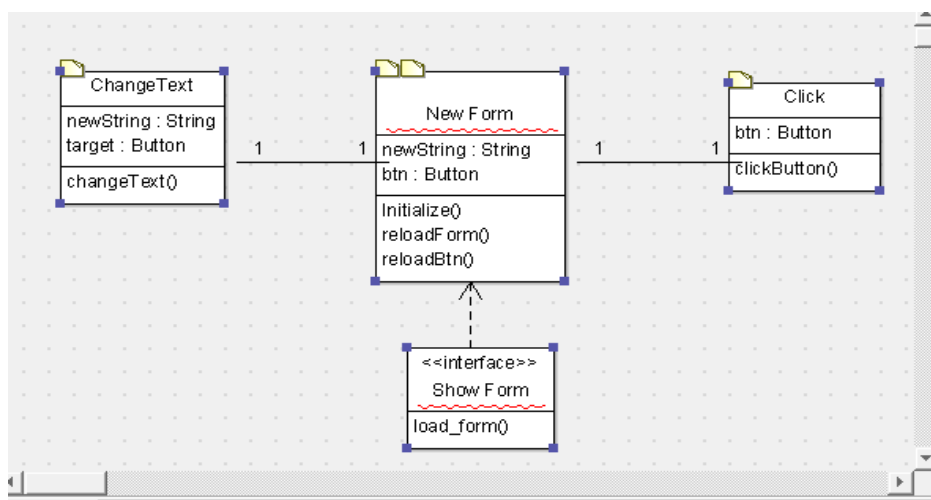


Рисунок 3.1 – Виділення класів для генерації програмного коду

2. Обрати опцію “Сгенерировать код для выбранных классов” з меню “Генерация кода” (рис. 3.2).

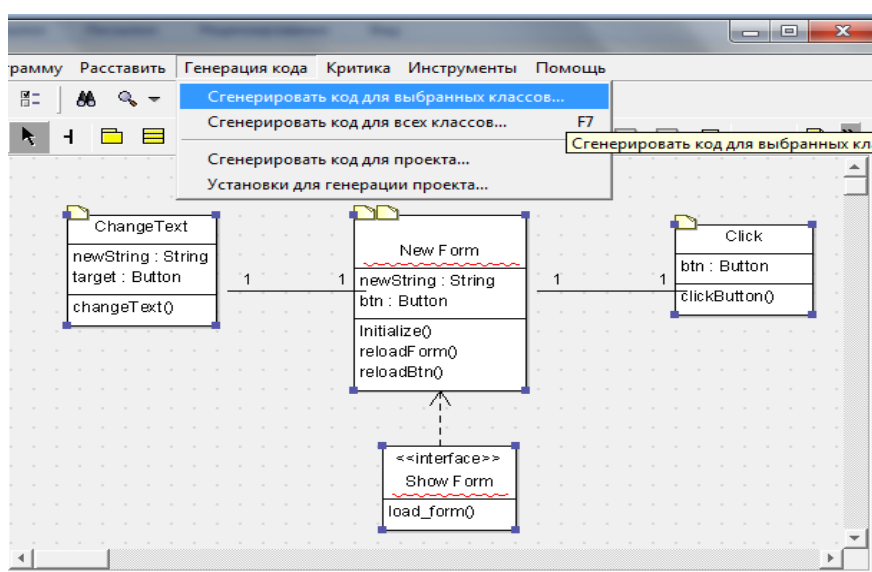


Рисунок 3.2 – Обрання опції “Сгенерировать код для выбранных классов”

3. Обрати необхідні мови програмування для результувального коду програми. На рисунку 3.3 наведено приклад обрання мови C# (CSharp).

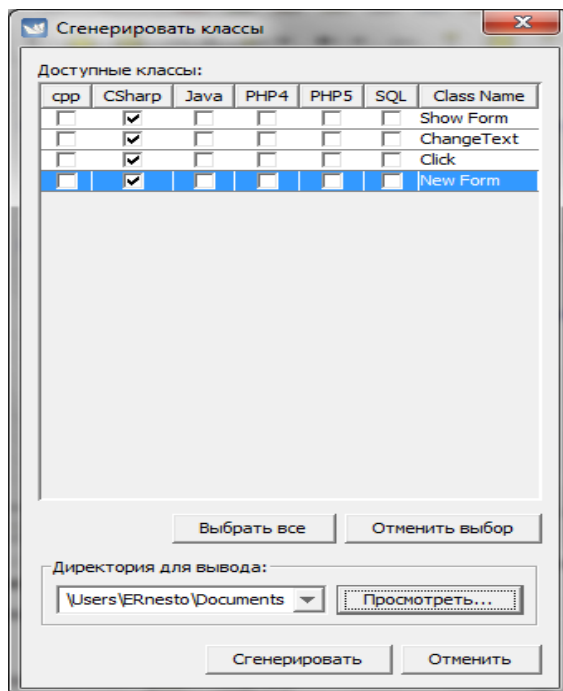


Рисунок 3.3 – Обрання мови програмування C# (CSharp)

В результаті виконання описаних вище дій буде отримано набір файлів з певним розширенням (залежно від обраної мови програмного коду). В подальшій розробці можна просто приєднати згенеровані файли до основного проекту.

Серед інших особливої уваги заслуговують діаграми класів.

### ***Створення діаграми класів у ArgoUML***

Існують такі способи створення діаграми класів:

- на стартовій сторінці в меню “Создать диаграмму” → “Диаграмма классов”;
- на головній панелі інструментів обрати піктограму, що наведена на рисунку 3.4.





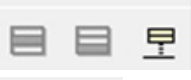


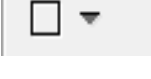


Рисунок 3.4 – Піктограма “Диаграмма классов”

Після створення діаграми класів головна панель інструментів змінить свій вигляд і буде містити лише ті елементи, що можуть бути використані при створенні діаграми класів (рис. 3.5).



Рисунок 3.5 – Вигляд головної панелі інструментів при створенні діаграми класів

-  – “Новый пакет”. Пакети використовуються у структурній моделі та ілюструють їх ієрархію. Класи можуть бути вкладеними або використовуватися для визначення мультизалежності пакетів;
-  – “Новый класс”. Класи містять дії та властивості і пов’язані з іншими класами через відношення асоціації або наслідування. Клас має такі властивості як ім’я, стереотип відношення до інших класів;
-  – “Разные типы связей”. Існують між пакетами і показують, що класи використовують в межах пакета, від якого вони залежать;
-  – “Новый интерфейс” та “Новая реализация”. Абстрактні класи, що не містять реалізацій. Клас, який наслідує інтерфейс, наслідує його методи і поля;
-  – додання атрибутів та методів до виділеного класу;
-  – “Новый тип данных”;
-  – створення коментарів для будь-яких частин діаграми;
-  – геометричні фігури.

Розглянемо приклад побудови діаграми класів при проектуванні прикладного додатка. Нехай діаграма класів матиме вигляд, наведений на рисунку 3.6.

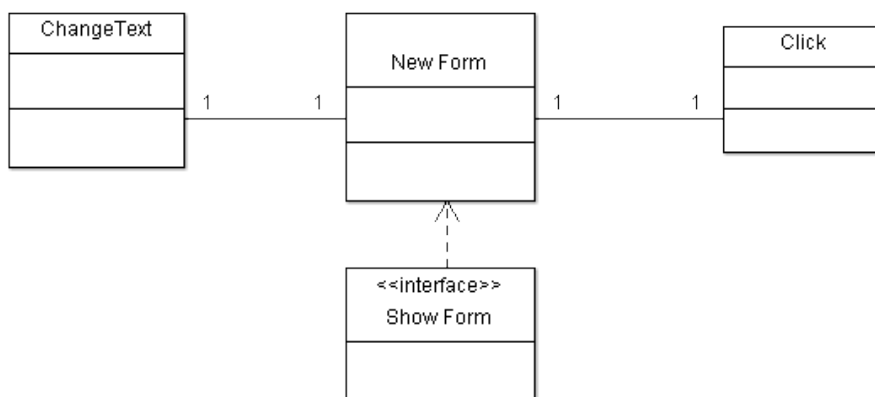


Рисунок 3.6 – Діаграма класів додатка

До створених класів додамо необхідні методи та поля, що потрібні для вирішення індивідуального завдання (рис. 3.7).

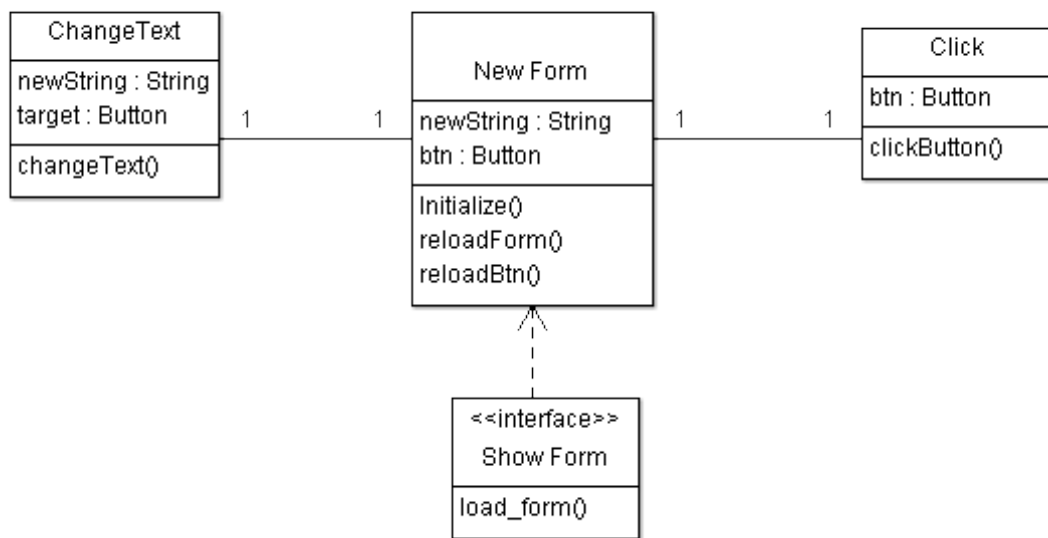


Рисунок 3.7 – Діаграма класів додатка з необхідними методами та полями

Тоді згенерований програмний код матиме вигляд:

```

// FILE: C:/Users/ERnesto/Documents//NewForm.cs
public class New Form: Show Form
{
    public String newString;
    public Button btn;
    public ArrayList myClick;
    public ArrayList myChangeText;
    public void Initialize()
    {
    }

    public void reloadForm()
    {
    }

    public void reloadBtn()
    {
    }

    public void load_form()
    {
    }
}
  
```



Отриманий програмний код доповнено відповідно до індивідуального завдання:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace lab_1
{
    public class New Form: Show Form
    {
        public String newString;
        public Button btn;
        public ArrayList myClick;
        public ArrayList myChangeText;

        public void Initialize()
        {
            InitializeComponent();
        }

        int n = 25;
        private void button1_Click(object sender, EventArgs e)

        public void reloadForm()
        {
        }

        public void reloadBtn()
        {
            n = n + 1;
            button1.Size = new System.Drawing.Size(150, n);
        }

        public void load_form()
        {
        }
    }
}
```

## Контрольні питання

1. Яка функція для проектування програм не реалізована у САПР ArgoUML?

- а) створення проектів (для моделювання масштабних систем);
- б) створення UML-діаграм різних типів;
- в) генерація програмного коду;
- г) генерація документації проекту.

2. В якому випадку не вдасться створити новий проект?

- а) Ctrl+N;
- б) “Файл” → “Новый проект”;
- в) “Файл” → “Новый документ”;
- г) Кнопка “Новый документ”.





3. Яким чином можна створити діаграму у САПР ArgoUML?

- а) “Файл” – “Создать диаграмму”;
- б) відповідна кнопка на стандартній панелі;
- в) натиснути “Ctrl” + “N”;
- г) “Создать диаграмму” – “Диаграмма”.

4. На якій панелі знаходяться клавіші додавання елементів та зв'язків у діаграмі?

- а) головне меню;
- б) стандартна панель;
- в) панель створення діаграм;
- г) діаграми проекту.

5. Яка піктограма відповідає за створення діаграми класів?







- а) ;
- б) ;
- в) ;
- г) .


6. Як змінюється вигляд стандартної панелі інструментів залежно від створення певного типу діаграм?

- а) повністю замінюється вміст панелі;
- б) змінюються лише деякі елементи;
- в) додаються певні елементи на панель;
- г) вміст панелі завжди незмінний.

7. Піктограми  означають...

- а) типи залежностей;
- б) види інтерфейсів;
- в) з'єднання елементів;
- г) геометричні фігури.

8. Яким чином можна вказати оголошення нового типу даних?
- меню “Редактировать”;
  - меню “Файл” – “Добавить”;
  - кнопка  стандартної панелі;
  - кнопка .
9. Пунктирна стрілка є позначенням...
- відношення класів;
  - відношення блоків;
  - відношення інтерфейсів;
  - відношення пакетів.
10. Як можна додати метод до певного класу при створенні діаграми класів?
- правою кнопкою миші та “Добавить” – “Новая переменная”;
  - меню “Редактировать” та “Добавить” – “Новая операция”;
  - правою кнопкою миші та “Добавить” – “Новая операция”;
  - меню “Файл” та “Добавить” – “Новый метод”.
11. Згенерувати код побудованої діаграми класів можна за допомогою...
- меню “Файл” – “Генерировать код”;
  - права кнопка миші “Генерация кода”;
  - панель властивостей моделі, кнопка “Генерация кода”;
  - пункт головного меню “Генерировать код”.
12. Якою піктограмою позначений елемент “пакет” на стандартній панелі?
-  ;
  -  ;
  -  ;
  - .
13. Для чого використовуються пакети при створенні діаграми класів?
- для ілюстрації ієрархії;
  - для вираження мультизалежності;
  - для групування елементів;
  - для позначення окремого типу класів.
14. Якої, властивої лише класу, властивості не має елемент клас?
- стереотип;
  - залежність;
  - пріоритетність;
  - видимість.

15. За додання якого типу елементів відповідає піктограма  ?
- а) геометричні фігури;
  - б) коментар до певної частини діаграми;
  - в) інтерфейс;
  - г) відношення.

### Рекомендована література

1. Норенков И. П. Основы теории и проектирования САПР : учебник для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1990. – 335 с.
2. Норенков И. П. Системы автоматизированного проектирования электронной и вычислительной аппаратуры : учебное пособие для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1983. – 272 с.
3. Питерсон Дж. Теория сетей Петри и моделирование систем / Питерсон Дж. ; пер. с англ. – М. : Мир, 1984. – 264 с.
4. Рафикузаман М. Микропроцессоры и машинное проектирование микропроцессорных систем : в 2-х кн. / Рафикузаман М. ; пер. с англ. – М. : Мир, 1988.
5. Системы автоматизированного проектирования : учебное пособие для вузов : в 9 кн. / [под ред. И. П. Норенкова]. – М. : Высшая школа, 1986. – 159 с.
6. Системы автоматизированного проектирования в радиоэлектронике : [справочник] / Е. В. Авдеев, А. Т. Еремин, И. П. Норенков, М. И. Песков ; под ред. И. П. Норенкова. – М. : Радио и связь, 1986. – 368 с.
7. Соловьев В. В. Проектирование функциональных узлов цифровых систем на программируемых логических устройствах : учебное пособие для вузов / Соловьев В. В. – М. : ПКООО Бестпринт, 1996. – 252 с.
8. Роїк О. М. Основи автоматизованого проектування складних об'єктів і систем : навчальний посібник / Роїк О. М., Савчук Т. О., Ткаченко О. М. – Вінниця : ВДТУ, 2001. – 111 с.

## ЛАБОРАТОРНА РОБОТА № 4

### СТВОРЕННЯ ДІАГРАМ КОМПОНЕНТІВ В ОБРАНІЙ САПР СЕРЕДОВИЩІ САПР

Мета: вивчення методів побудови діаграм компонентів та типів відношень всередині цих діаграм в обраній САПР.

## Порядок виконання роботи

1. Ознайомитися з основними функціями обраної САПР, що можуть бути використані для проектування програм.
2. Оволодіти методикою і способами створення діаграм компонентів в обраній САПР.
3. Вивчивши основні типи відношень всередині діаграм компонентів, побудувати довільну діаграму вашого додатка за індивідуальним завданням.
4. Створення діаграми компонентів повинно відбуватись покроково з відповідними поясненнями.
5. В створеній діаграмі обов'язково показати відношення між елементами, що потрібні для кращого розуміння вашого завдання.
6. Оформити звіт.

Таблиця 4.1 – Варіанти створення компонентів

Варіант	Завдання
1	Область MyPanel, яка масштабується в обох напрямках. Коли один з розмірів зменшується до 0 або виходить за межі форми, – змінює напрямок масштабування на протилежний.
2	Прапорець MyCheckBox, який сам встановлюється та скидається з певною частотою.
3	Текстовий елемент MyLabel, який переміщується по формі з певною швидкістю в горизонтальному напрямку, а коли доходить до краю форми змінює напрямок руху на протилежний.
4	Текстове поле MyEdit, яке самостійно вводить у себе заданий текст з певною швидкістю.
5	Текстовий елемент MyLabel, який змінює свій колір з певною частотою (використати не менше 5 кольорів).
6	Кнопку MyButton, яка переміщується по формі з певною швидкістю у вертикальному напрямку, а коли доходить до краю форми змінює напрямок руху на протилежний.

## Теоретичні відомості

*Діаграма компонентів* (Component diagram) призначена для розподілу класів і об'єктів за компонентами при фізичному проектуванні системи. Часто цей тип діаграм називають діаграмами модулів.

Повний проект програмної системи є сукупністю моделей логічного та фізичного рівнів, які повинні бути узгоджені між собою. У мові UML для фізичного подання моделей систем використовуються діаграми реалізації (implementation diagrams), які містять діаграму компонентів і діаграму розгортання. Діаграма компонентів описує особливості фізичного подання

системи. Вона дозволяє визначити архітектуру розроблюваної системи, встановивши залежності між програмними компонентами, в ролі яких може виступати вихідний і виконуваний код. Основними графічними елементами діаграми компонентів є компоненти, інтерфейси і залежності між ними. Діаграма компонентів розробляється для таких цілей:

- візуалізації загальної структури вихідного коду програмної системи;
- специфікації виконуваного варіанта програмної системи;
- забезпечення багаторазового використання окремих фрагментів програмного коду;
- подання концептуальної і фізичної схем баз даних.

У розробці діаграм компонентів беруть участь як системні аналітики та архітектори, так і програмісти. Діаграма компонентів забезпечує узгоджений перехід від логічного подання до конкретної реалізації проекту у вигляді програмного коду. Одні компоненти можуть існувати тільки на етапі компіляції програмного коду, інші – на етапі його виконання. Діаграма компонентів відображає загальні залежності між компонентами, розглядаючи останні як класифікатори.

### ***Компоненти***

Для подання елементів у мові UML застосовується спеціальний термін – компонент (component). Компонент реалізує деякий набір інтерфейсів і слугує для загального позначення елементів фізичного подання моделі. Для графічного подання компонента використовується спеціальний символ – прямокутник зі вставленими зліва двома дрібнішими прямокутниками. Усередині великого прямокутника записується ім'я компонента і, за необхідності, деяка додаткова інформація. Зображення цього символу може незначно варіюватися залежно від характеру; асоціюється з компонентом інформації.

*Ім'я компонента* підпорядковується загальним правилам іменування елементів моделі в мові UML і може складатися з будь-якої кількості літер, цифр і деяких знаків пунктуації.

Окремий компонент може бути поданий на рівні типу або на рівні екземпляра. Графічне зображення в обох випадках однакове, але правила запису імені компонента відрізняються. Якщо компонент подається на рівні типу, то його ім'я записується з великої літери як ім'я типу. Якщо ж компонент подається на рівні екземпляра, то його ім'я записується як “ім'я компонента”：“ім'я типу”. При цьому весь рядок імені підкреслюється.

Як прості імена прийнято використовувати імена виконуваних файлів (із зазначенням розширення exe після крапки-роздільника), динамічних бібліотек (розширення dll), Web-сторінок (розширення html), текстових файлів (розширення txt чи doc) або файлів довідки (hip), файлів баз даних (DB) або файлів з вихідними текстами програм (розширення h, cpp для мови C++, розширення java для мови Java), скрипти (pi, asp) та інші.

Оскільки конкретна реалізація логічного подання моделі системи залежить від використовуваного програмного інструментарію, то й імена компонентів визначаються особливостями синтаксису відповідної мови програмування.

В окремих випадках до простого імені компонента може бути додана інформація про ім'я доступного пакета і про конкретну версію реалізації цього компонента. У такому випадку номер версії записується як позначене значення в фігурних дужках. В інших випадках символ компонента може бути розділений на секції, щоб явно вказати імена реалізованих у ньому інтерфейсів.

У мові UML виділяють три види компонентів:

- розгортання, які забезпечують безпосереднє виконання системою своїх функцій. Такими компонентами можуть бути спільні бібліотеки з розширенням `dll`, `Web`-сторінки на мові розмічування гіпертексту з розширенням `html` і файли довідки з розширенням `hlp`;
- робочі продукти. Як правило, це файли з вихідними текстами програм, наприклад, з розширеннями `h` або `CPP` для мови `C++`;
- виконання, що являють собою виконавчі модулі – файли з розширенням `exe`.

Ці елементи іноді називають артефактами, підкреслюючи при цьому їх закінчений інформаційний зміст, залежний від конкретної технології реалізації відповідних компонентів.

Іншим способом специфікації різних видів компонентів є явна вказівка його стереотипу перед ім'ям. У мові UML для компонентів визначені такі стереотипи:

- `library` – визначає перший різновид компонента, який подається у вигляді динамічної або статичної бібліотеки;
- `table` – визначає різновид компонента, який подається у вигляді таблиці бази даних;
- `file` – визначає різновид компонента, який подається у вигляді файлів з вихідними текстами програм;
- `document` – визначає різновид компонента, який подається у вигляді документа;
- `executable` – визначає різновид компонента, що виконується.

### ***Інтерфейси***

Важливими елементами діаграми компонентів є інтерфейси. У загальному випадку, інтерфейс графічно зображується колом, яке з'єднується з компонентом відрізком лінії без стрілок. Ім'я інтерфейсу має починатися з великої літери "I" і записуватися поруч з колом. Семантично лінія означає реалізацію інтерфейсу, а наявність інтерфейсів у компонента означає, що даний компонент реалізує відповідний набір інтерфейсів.

Іншим способом подання інтерфейсу на діаграмі компонентів є його зображення у вигляді прямокутника класу зі стереотипом "інтерфейс" і

можливими секціями атрибутів та операцій. Як правило, цей варіант позначення використовується для подання внутрішньої структури інтерфейсу, яка може бути важлива для реалізації.

При розробці програмних систем інтерфейси забезпечують не тільки сумісність різних версій, але і можливість вносити зміни в одні частини програми, не змінюючи інші. Таким чином, призначення інтерфейсів істотно ширше ніж специфікація взаємодії з користувачами системи.

### ***Відношення***

Відношення не є асоціацією та використовується для подання факту наявності такого зв'язку, коли зміна одного елемента моделі впливає або призводить до зміни іншого її елемента. Відношення на діаграмі компонентів зображується пунктирною лінією зі стрілкою, спрямованою від залежного елемента до незалежного.

Відношення можуть відображати зв'язок модулів програми на етапі компіляції та генерації об'єктного коду. В іншому випадку, відношення може відображати наявність у незалежному компоненті описів класів, які використовуються у залежному компоненті для створення відповідних об'єктів. У діаграмі компонентів відношення можуть поєднувати компоненти й імпортовані цими компонентами інтерфейси, а також різні види компонентів між собою.

У першому випадку рисують стрілку від компонента до імпортованого інтерфейсу. Наявність стрілки означає, що компонент не реалізує відповідний інтерфейс, а використовує його в процесі свого виконання. На цій же діаграмі може бути присутній і інший компонент, який реалізує цей інтерфейс.

Наявність відношення між різними видами компонентів означає, що внесення змін до вихідних текстів програм або динамічних бібліотек призводить до змін самого компонента. При цьому характер змін може бути відзначений додатково.

На діаграмі компонентів можуть бути також подані відношення між компонентами та реалізованими в них класами. Ця інформація має значення для забезпечення узгодження логічного та фізичного подання моделі системи. Якщо потрібно підкреслити, що деякий компонент реалізує окремі класи, то для позначення компонента використовується розширений символ прямокутника. При цьому прямокутник компонента ділиться на дві секції горизонтальною лінією. Верхня секція слугує для запису імені компонента, а нижня секція – для вказання додаткової інформації.

Всередині символу компонента можуть зображатися інші елементи графічної нотації, такі як класи (компонент рівня “тип”) чи об'єкти (компонент рівня “екземпляр”). У цьому випадку компонент зображується таким чином, щоб вмістились ці додаткові символи.

Об'єкти, які знаходяться в окремому компоненті-примірнику, зображуються вкладеними в символ цього компонента. Подібна вкладеність



означає, що виконання компонента вилучає виконання відповідних об'єктів.

Діаграми компонентів містять:

- відношення використання, при яких зміна в специфікації одного тягне за обою зміну іншого. Графічно подається пунктирною стрілкою між залежними елементами;

- узагальнення – будь-який екземпляр підтипу є також примірником супертипу. Наприклад: тварини – супертип ссавців, які, в свою чергу, є супертипом приматів і так далі. Цей взаємозв'язок найлегше описується фразою “А – це Б” (примати – це ссавці, ссавці – це тварини);

- асоціації, які показують, що об'єкти однієї сутності пов'язані з об'єктами іншої сутності;

- реалізації відношення між двома елементами моделі, в якому один елемент реалізує поведінку, задану іншим.

У САПР Argo UML створення діаграм компонентів та діаграм об'єктів об'єднано у один тип діаграм – діаграми розгортання (Deployment diagram). Для створення діаграми компонентів необхідно обрати пункт “Создать диаграмму” – “Диаграмма развертывания” (рис. 4.1).

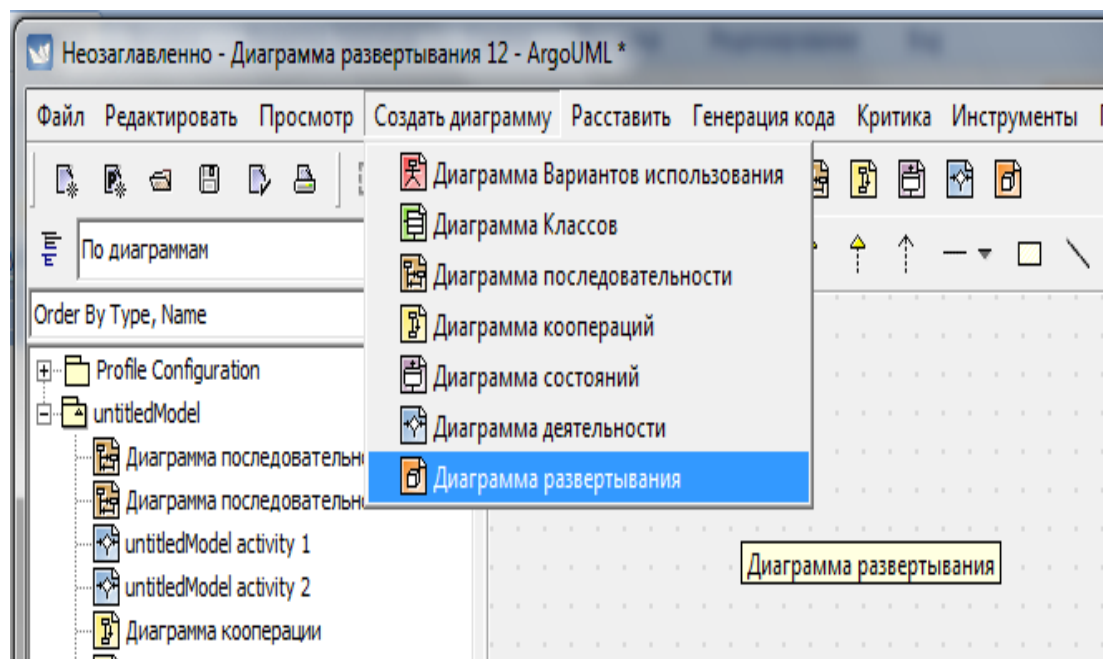


Рисунок 4.1 – Створення діаграми розгортання

Моделювання початкового коду системи здійснюється таким чином:

- ідентифікація наборів початкових файлів, що являють інтерес, і моделювання їх у вигляді компонентів із стереотипом file;

- використання пакетів, щоб показати групи початкових файлів для потужних систем;

– подання за допомогою помічених значень такої інформації, як номер версії файлу, його автора і дата останньої зміни (для відстеження таких значень використовують інструментальні засоби);

– моделювання залежності компіляції між початковими файлами за допомогою відношень залежності. Як приклад, на рисунку 4.2 показано п'ять початкових файлів, з яких `signal.h` – заголовок, а також наведено три його версії, починаючи з найостаннішої. Кожна версія помічена значенням, що містить її номер (рис. 4.2).

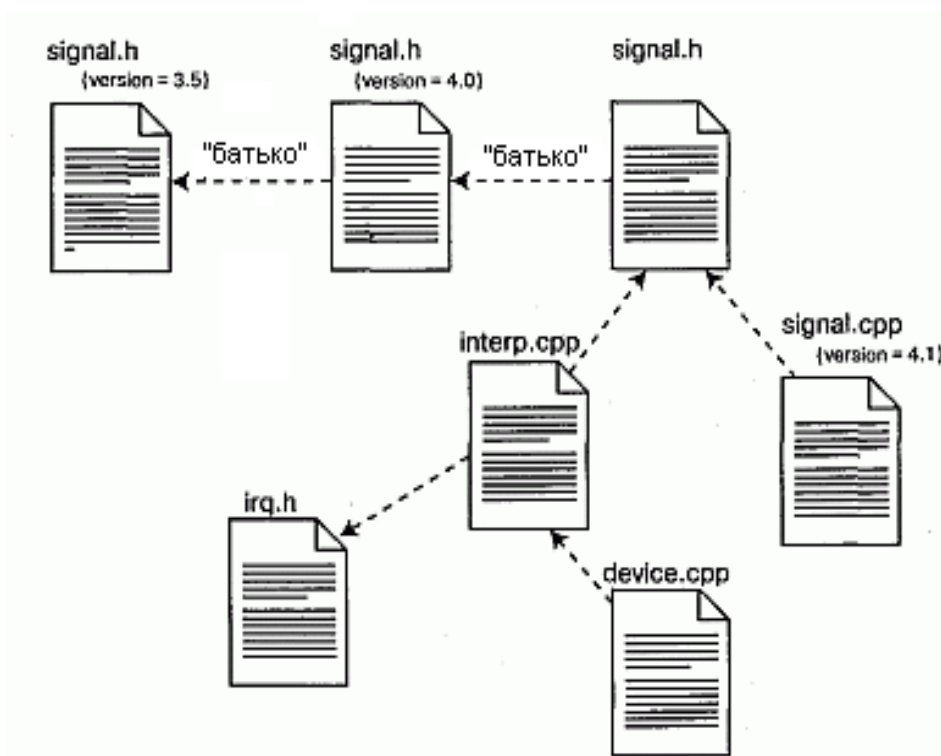


Рисунок 4.2 – Моделювання вихідного коду

Файл-заголовок (`signal.h`) використовується двома іншими файлами (`interp.cpp` і `signal.cpp`). Один з них (`interp.cpp`) залежить при компіляції від іншого файлу-заголовка (`irq.h`). У свою чергу файл `device.cpp` залежить від `interp.cpp`. За наявності такої діаграми компонентів легко прослідкувати, що відбудеться при змінах. Так, зміна початкового файлу `signal.h` приведе до перекомпіляції трьох інших файлів: `signal.cpp`, `interp.cpp` і, транзитивно, `device.cpp`. З тієї ж діаграми видно, що файлу `irq.h` така зміна не торкнеться.

Розглянемо початкові файли:

- `Program.cs`;
- `Form1.cs`;
- `Form1.Designer.cs`;
- `Form1.resx`.

Змоделюємо їх у вигляді компонентів із стереотипом `file` (рис. 4.3).

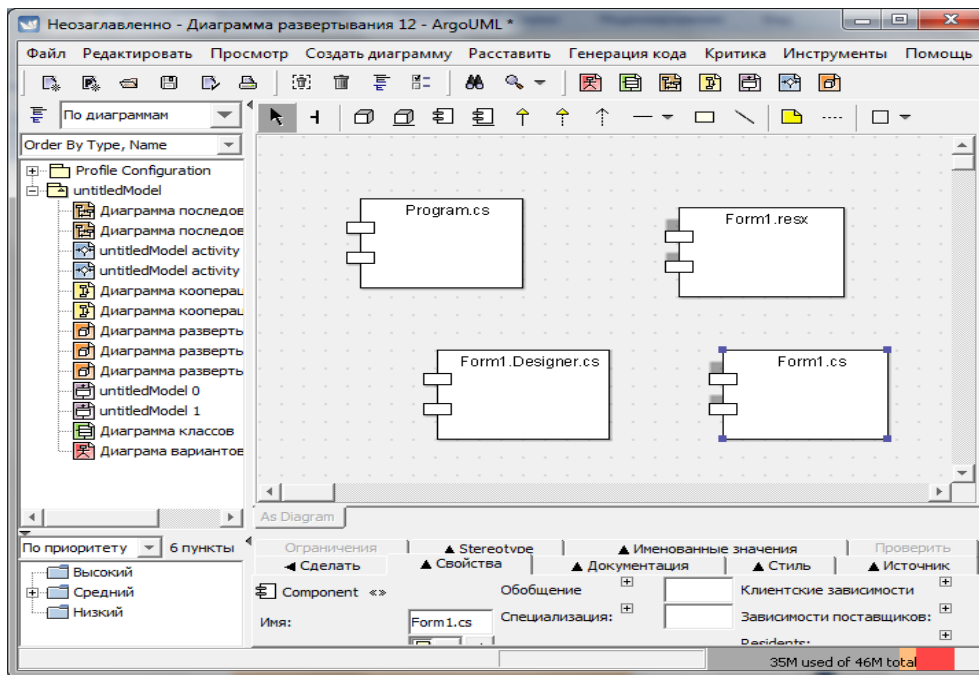


Рисунок 4.3 – Компоненти із стереотипом file

Змоделюємо залежність компіляції між початковими файлами за допомогою відношень залежності (рис. 4.4).

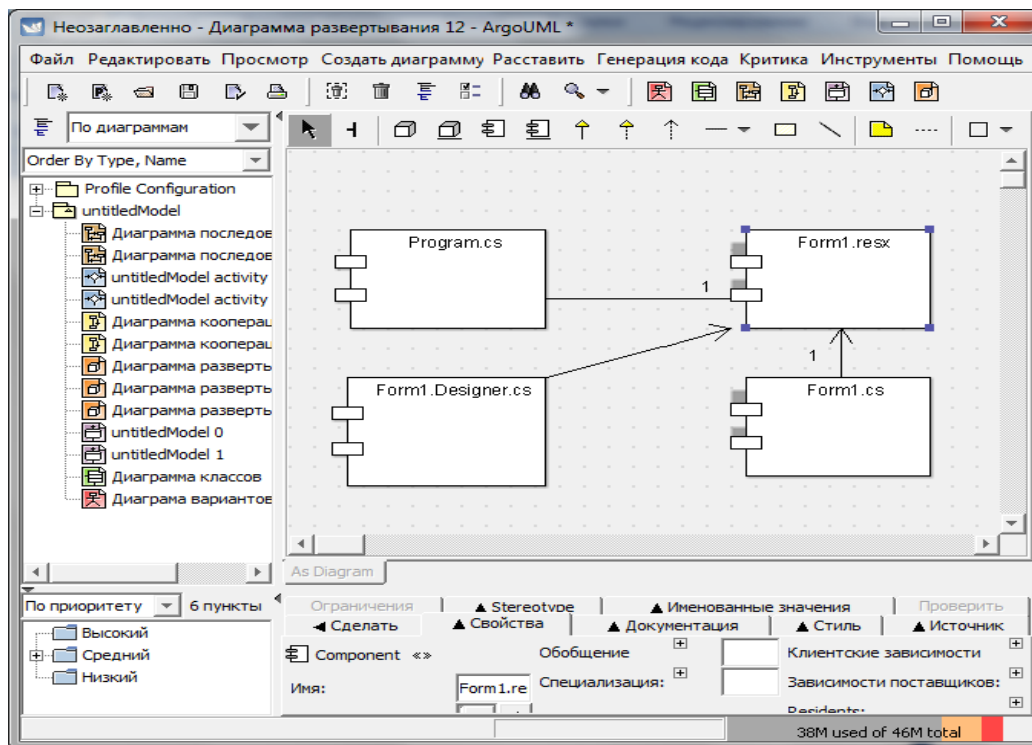


Рисунок 4.4 – Створення відношень залежності

### Контрольні питання

1. Визначте основні функції обраної САПР, що можуть бути використаними для проектування програм.

2. Методики та способи створення діаграм компонентів в обраній САПР.

3. Основні етапи побудови довільної діаграми вашого додатка за індивідуальним завданням.

4. Діаграма компонентів. Цілі для розробки діаграми компонентів.

5. Поняття компоненти. Три види компонентів у мові UML.

6. Інтерфейс як важливий елемент діаграми компонентів.

7. Характеристика діаграми компонентів.

### **Рекомендована література**

1. Норенков И. П. Основы теории и проектирования САПР : учебник для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1990. – 335 с.

2. Норенков И. П. Системы автоматизированного проектирования электронной и вычислительной аппаратуры : учебное пособие для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1983. – 272 с.

3. Питерсон Дж. Теория сетей Петри и моделирование систем / Питерсон Дж. ; пер. с англ. – М. : Мир, 1984. – 264 с.

4. Рафикузаман М. Микропроцессоры и машинное проектирование микропроцессорных систем : в 2-х кн. / Рафикузаман М. ; пер. с англ. – М. : Мир, 1988.

5. Системы автоматизированного проектирования : учебное пособие для вузов : в 9 кн. / [под ред. И. П. Норенкова]. – М. : Высшая школа, 1986. – 159 с.

6. Системы автоматизированного проектирования в радиоэлектронике : [справочник] / Е. В. Авдеев, А. Т. Еремин, И. П. Норенков, М. И. Песков ; под ред. И. П. Норенкова. – М. : Радио и связь, 1986. – 368 с.

7. Соловьев В. В. Проектирование функциональных узлов цифровых систем на программируемых логических устройствах : учебное пособие для вузов / Соловьев В. В. – М. : ПКООО Бестпринт, 1996. – 252 с.

8. Роїк О. М. Основи автоматизованого проектування складних об'єктів і систем : навчальний посібник / Роїк О. М., Савчук Т. О., Ткаченко О. М. – Вінниця : ВДТУ, 2001. – 111 с.

### **ЛАБОРАТОРНА РОБОТА № 5**

#### **СТВОРЕННЯ ДІАГРАМ РОЗГОРТУВАННЯ В ОБРАНІЙ САПР**

Мета: вивчення методів побудови діаграм розгортання та типів відношень в діаграмах з використанням обраної САПР.

## Порядок виконання роботи

1. Ознайомитися з основними функціями обраної САПР, що можуть бути використаними для проектування програм.
2. Оволодіти методикою і способами створення діаграм розгортання в обраній САПР.
3. Вивчивши основні типи відношень всередині діаграм розгортання, побудувати діаграму вашого додатка за індивідуальним завданням.
4. Зробити висновки та оформити звіт.

## Теоретичні відомості

В САПР ArgoUML для проектування програм реалізовані такі основні функції:

- створення проектів (для моделювання масштабних систем);
- створення UML-діаграм різних типів;
- генерація програмного коду;
- генерація документації проекту.

### *Створення проектів*

Для створення нового проекту потрібно обрати в головному меню програми “Файл” → “Новый проект” (Ctrl+N). Після цього необхідно обрати та додати до моделі необхідний тип діаграми у меню “Создания диаграммы”.

### *Створення UML-діаграм різних типів*

Діаграми створюються за допомогою панелі “Создания диаграммы” панелі інструментів.

Елементи моделі, які вже знаходяться в моделі, але не на діаграмі, можуть бути додані до діаграми обранням елемента моделі в провіднику з використанням “Добавить в диаграмму”. Інші необхідні UML-елементи можна знайти на панелі інструментів, яка містить загальні графічні об’єкти (прямокутники, круги, лінії, багатокутники, криві, текст) для надання додаткової інформації про діаграми, а також специфічні для кожного типу діаграм елементи.

Щоб створити зв’язок між елементами, необхідно обрати потрібний тип зв’язку в панелі інструментів та провести цей зв’язок між заданими елементами.

Після створення діаграми класів програми, що розробляється, є можливість згенерувати код для перенесення до розроблюваного засобу, за допомогою якого відбуватиметься повне тестування та подальша розробка.

Якщо розробляється програма, що виконується локально на комп’ютері користувача і не використовує периферійних пристроїв і ресурсів, то в розробці додаткових діаграм немає необхідності. При розробці корпорати-

вних додатків наявність таких діаграм може бути необхідною для розв'язання задач раціонального розміщення компонентів з метою ефективного використання розподілених обчислювальних та комунікаційних ресурсів мережі, забезпечення безпеки та інше.

Для подання загальної конфігурації і топології розподіленої програмної системи в UML використовуються діаграми розгортання.

Діаграма розгортання призначена для візуалізації елементів і компонентів програми, які існують лише на етапі її виконання (runtime). При цьому подаються тільки компоненти-екземпляри програми, які є виконуваними файлами або динамічними бібліотеками. Ті компоненти, які не використовуються на етапі виконання, на діаграмі розгортання не показуються. Так, компоненти з вихідними текстами програм можуть бути присутніми тільки на діаграмі компонентів. На діаграмі розгортання вони не вказуються.

Діаграма розгортання містить графічні зображення процесорів, пристроїв, процесів і зв'язків між ними. На відміну від діаграм логічного подання, діаграма розгортання є єдиною для системи в цілому, оскільки повинна цілком відображати особливості її реалізації. Розробка діаграми розгортання, як правило, є останнім етапом специфікації моделі програмної системи.

При розробці діаграми розгортання необхідно:

- визначити розподіл компонентів системи за її фізичними вузлами;
- показати фізичні зв'язки між усіма вузлами реалізації системи на етапі її виконання;
- реконфігурувати топологію системи для досягнення необхідної продуктивності.

Вузол (node) діаграми являє собою деякий фізично існуючий елемент системи, що має певний обчислювальний ресурс. Як обчислювальний ресурс сайту може розглядатися наявність деякої пам'яті або процесор. В останній версії мови UML поняття вузла розширено і може містити не тільки обчислювальні пристрої, але й інші механічні або електронні пристрої, такі як датчики, принтери, модеми, цифрові камери, сканери і маніпулятори.

Графічно на діаграмі розгортання вузол зображується у вигляді тривимірного куба. Вузол має власне ім'я, яке вказується всередині його графічного символу. Самі вузли можуть подаватися як типи, так і як примірники.

У першому випадку ім'я вузла записується без підкреслення і починається з великої літери. У другому – ім'я вузла-примірника записується у вигляді <ім'я вузла ':' ім'я типу вузла>. Ім'я типу вузла вказує на деякий різновид вузлів, присутніх в моделі системи.

Наприклад, апаратна частина системи може складатися з декількох комп'ютерів, кожен з яких відповідає окремому вузлу-екземпляру в моде-

лі. Проте, всі ці вузли-екземпляри належать до одного типу вузлів, а саме вузла з ім'ям типу "Комп'ютер".

Так само, як і на діаграмі компонентів, зображення вузлів можуть розширюватися, щоб вмістити деяку додаткову інформацію про специфікації вузла. Якщо додаткова інформація належить до назви вузла, то вона записується під цим ім'ям.

Якщо необхідно явно вказати компоненти, які розміщуються на окремому сайті, то це можна зробити двома способами:

- розділити графічний символ вузла на дві секції горизонтальною лінією. У верхній секції записати ім'я вузла, а в нижній – розміщені на цьому сайті компоненти;

- показати на діаграмі розгортання вузли з вкладеними зображеннями компонентів. Однак потрібно враховувати, що як такі вкладені компоненти можуть виступати тільки виконувані компоненти.

Як доповнення до імені вузла можуть використовуватися різні стереотипи, які явно специфікують призначення цього вузла. Хоча в мові UML стереотипи для вузлів не визначені, в літературі зустрічаються такі їх варіанти: "процесор", "датчик", "модем", "мережа" та інші, які самостійно можуть бути визначені розробником. На діаграмах розгортання для різних фізичних пристроїв також допускаються спеціальні графічні позначення, що пояснюють і розкривають призначення або виконувані пристроєм функції.

Крім зображень вузлів на діаграмі розгортання вказуються відношення між ними. Як відношення виступають фізичні з'єднання між вузлами і залежності між вузлами та компонентами, зображення яких теж можуть бути присутніми на діаграмах розгортання.

З'єднання є різновидом асоціації та зображуються відрізками ліній без стрілок. Наявність такої лінії вказує на необхідність організації фізичного каналу для обміну інформацією між відповідними вузлами. Характер з'єднання може бути додатково специфіковане приміткою, помічене значенням або обмеженням.

Крім з'єднань, на діаграмі розгортання можуть бути присутніми відношення залежності між вузлом і розгорнутими на ньому компонентами. Подібний спосіб є альтернативою вкладеному зображенню компонентів всередині символу вузла, особливо при великій кількості розгорнутих на вузлі компонентів.

Діаграми розгортання можуть мати складну структуру, що містять вкладені компоненти, інтерфейси й інші апаратні пристрої.

У САПР Argo UML створення діаграм компонентів та діаграм об'єктів об'єднано у один тип діаграм – діаграми розгортання (Deployment diagram). Отже, для створення діаграми компонентів необхідно обрати "Создать диаграмму" → "Диаграмма развертывания" (рис. 5.1).

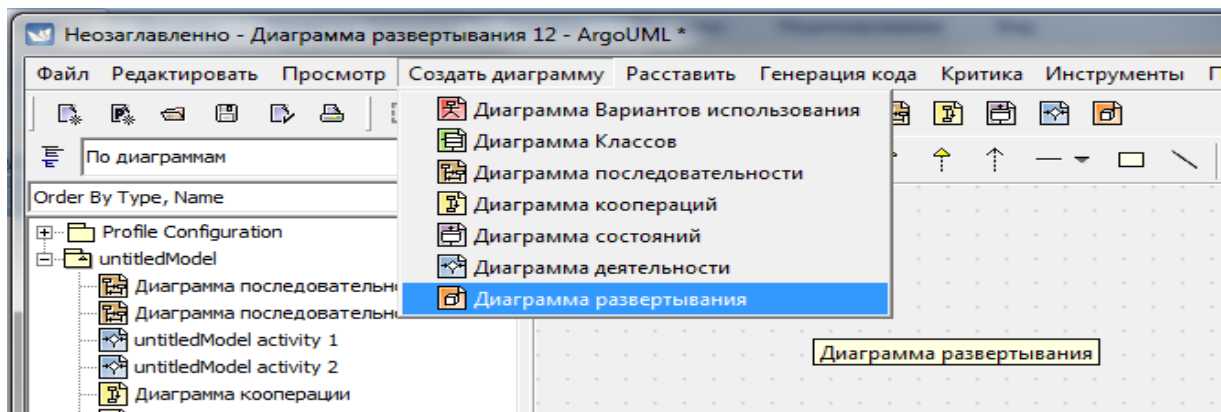


Рисунок 5.1 – Створення діаграми розгортання

У стандартній панелі інструментів з'являються бібліотечні елементи для побудови діаграми розгортання (рис. 5.2).



Рисунок 5.2 – Набір бібліотечних елементів

Для додавання елемента в діаграму потрібно обрати його в панелі інструментів і розмістити в робочій області, як показано на рисунку 5.3.

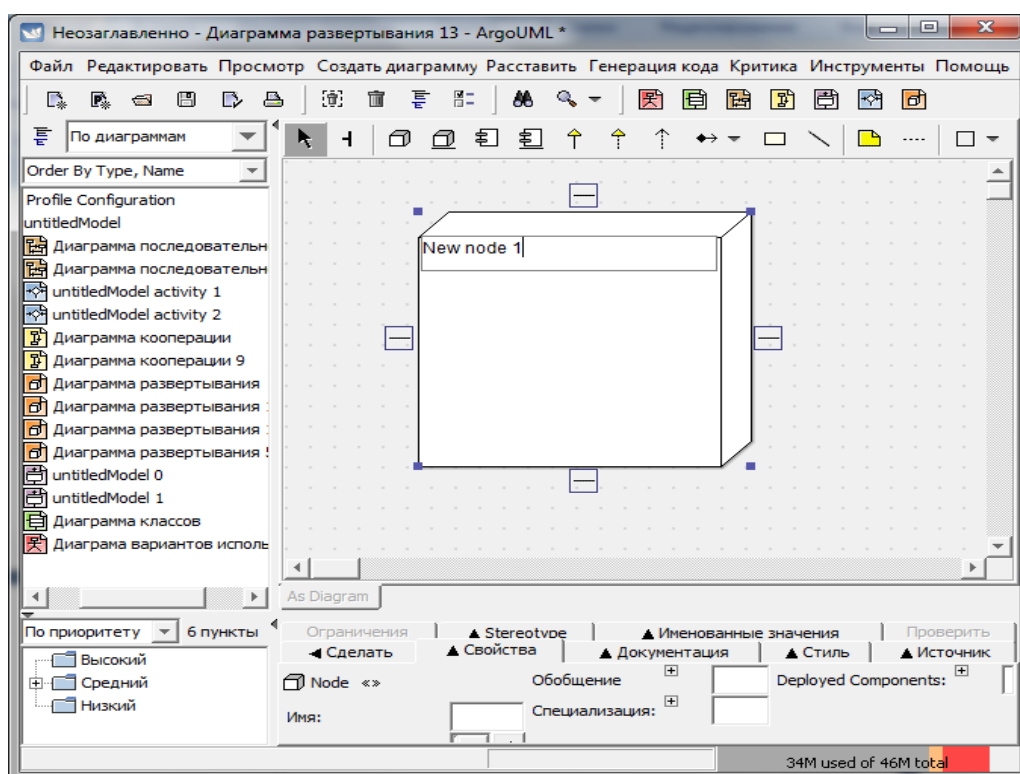


Рисунок 5.3 – Розміщення бібліотечного елемента в робочій області

При створенні вузла активується поле для введення назви, яке фіксується після натиснення клавіші Enter.



Редагування стає доступним при обранні необхідної вкладки у нижній панелі інструментів.

Для створення взаємозв'язку між компонентами, потрібно обрати його в меню інструментів і провести між обраними компонентами, як показано на рисунку 5.4.

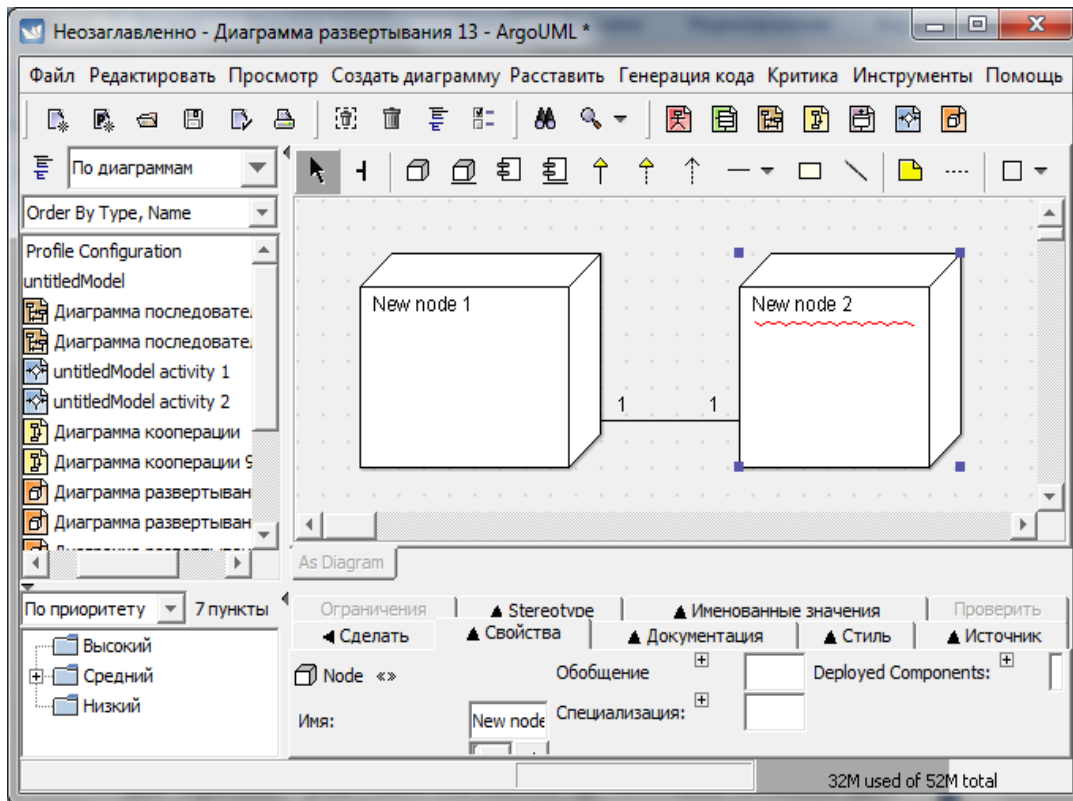


Рисунок 5.4 – Відображення взаємозв'язку між компонентами

Діаграми розгорткування зазвичай містять відношення асоціації. Відношення асоціації відповідає наявності деякого відношення між класами. Це відношення позначається неперервною лінією з додатковими спеціальними символами, які характеризують окремі властивості конкретної асоціації. Як додаткові спеціальні символи можуть використовуватися ім'я асоціації, а також імена і кратність класів-ролей асоціації. Ім'я асоціації є обов'язковим елементом її позначення. Якщо воно задано, то записується з великої літери поруч з лінією відповідної асоціації.

Найбільш простий випадок цього відношення – бінарна асоціація. Вона пов'язує два класи і, як виняток, може пов'язувати клас із самим собою (рис. 5.5). Для бінарної асоціації на діаграмі може бути вказаний порядок проходження класів з використанням трикутника у вигляді стрілки поруч з ім'ям цієї асоціації. Напрямок цієї стрілки вказує на порядок класів, один з яких є першим (з боку трикутника), а інший – другим (з боку вершини трикутника). Відсутність означеної стрілки поруч з ім'ям асоціації означає, що порядок проходження класів у розглянутому відношенні невизначений.

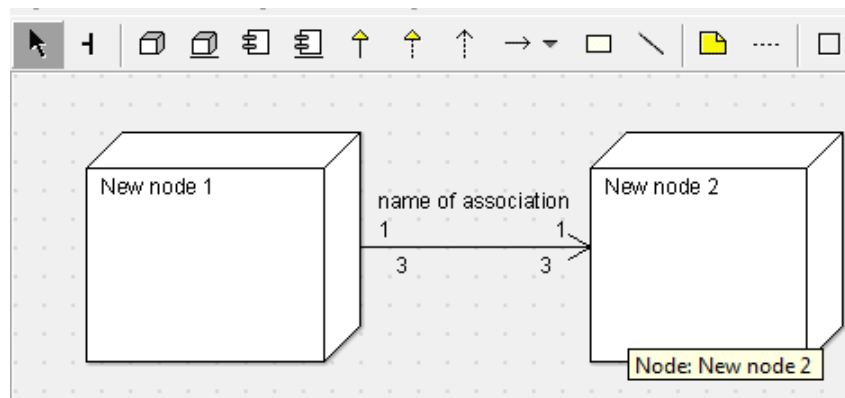


Рисунок 5.5 – Створення відношення асоціації

Асоціації більш високого порядку в загальному випадку називаються N-арними асоціаціями. Така асоціація показує відношення трьох і більше класів, при цьому один клас може брати участь в асоціації більше ніж один раз. Клас асоціації має певну роль у відповідному відношенні, що може бути явно вказано на діаграмі. Кожен екземпляр N-арної асоціації являє собою N-арний кортеж значень об'єктів з відповідних класів. N-арна асоціація графічно позначається ромбом, від якого позначають зв'язки до символів класів цієї асоціації. У цьому випадку ромб з'єднується з символами відповідних класів неперервними лініями. Зазвичай лінії проводяться від вершин ромба або від середини його сторін. Ім'я N-арної асоціації записується поруч з ромбом відповідної асоціації.

Порядок класів в N-арній асоціації, на відміну від порядку множин у відношенні, на діаграмі не фіксується. Клас може бути приєднаним до ромба перервною лінією. Це означає, що цей клас забезпечує підтримку властивостей відповідної N-арної асоціації, а сама N-арна асоціація має атрибути, операції і/або асоціації. Іншими словами, така асоціація, в свою чергу, є класом з відповідним позначенням у вигляді прямокутника, як самостійний елемент мови UML – асоціацією-класом (Association Class). Окремий клас асоціації має власну роль у відношенні. Ця роль може бути зображена графічно на діаграмі класів. З цією метою в мові UML вводиться поняття “кінець асоціації” (Association End), яке графічно відповідає точці з'єднання лінії асоціації з окремим класом. Кінець асоціації є частиною асоціації, але не класу. Кожна асоціація має два або більше кінців асоціації. Найбільш важливі властивості асоціації вказуються на діаграмі поряд з цими елементами асоціації і мають перемішатися разом з ними.

Одним з таких додаткових позначень є ім'я ролі окремого класу, що входить в асоціацію. Ім'я ролі є рядком тексту поруч з кінцем асоціації для відповідного класу. Воно вказує на специфічну роль, яку відіграє клас, що є кінцем розглянутої асоціації. Ім'я ролі не є обов'язковим елементом позначень і може бути відсутнім на діаграмі.

Важливим елементом позначень є кратність окремих класів, які є кінцями асоціації. Кратність окремого класу позначається у вигляді інтервалу

цілих чисел аналогічно кратності атрибутів і операцій класів. Інтервал записується поруч з кінцем асоціації і для N-арної асоціації означає потенційну кількість окремих примірників або значень кортежів цієї асоціації, які можуть мати місце, коли інші (N-1) примірники або значення класів фіксовані.

Спеціальною формою відношення асоціації є відношення агрегації, яке теж має спеціальну форму – відношення композиції.

Відношення агрегації (рис. 5.6) має місце між декількома класами в тому випадку, якщо один з класів є деякою сутністю, що містить, як складові частини, інші сутності. Це відношення має фундаментальне значення для опису структури складних систем, оскільки застосовується для подання системних взаємозв'язків типу “частина – ціле”. З точки зору моделі, окремі складові системи можуть виступати як у вигляді елементів, так і у вигляді підсистем, які, у свою чергу, теж можуть утворювати складові компоненти або підсистеми. Це відношення за своєю суттю описує декомпозицію або розбиття складної системи на простіші складові частини, які також можуть бути декомпонованими, якщо в цьому виникне необхідність.

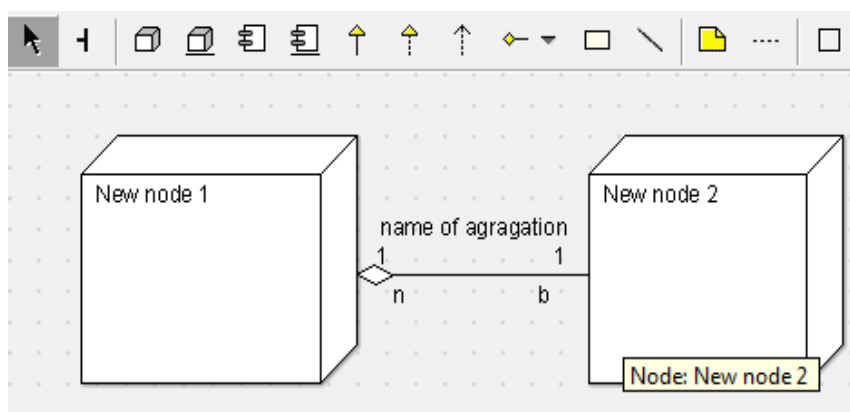


Рисунок 5.6 – Створення відношення агрегації

Графічно відношення агрегації зображується неперервною лінією, один з кінців якої є незафарбованим всередині ромбом. Цей ромб вказує на той з класів, який являє собою “ціле”. Решта класів є його “частинами”. Відношення композиції є окремим випадком відношення агрегації і слугує для виділення спеціальної форми відношення “частина – ціле”. Специфіка взаємозв'язку між ними полягає в тому, що частини не можуть існувати окремо від цілого, тобто із знищенням цілого знищуються і всі його складові частини.

Розглянемо приклад створення діаграми розгортання, яка буде демонструвати систему доступу до баз даних на основі Internet-технології.

Система складається з двох комп'ютерів. Один з них виконує функції web-сервера з можливістю виконання ASP чи CGI і готує дані для відображення на комп'ютері клієнта. Для підготовки даних web-сервер виконує запити до SQL – сервера, що зберігає дані й обробляє їх. Усе це

робиться за запитом з комп'ютера клієнта. Саме на нього посилаються результати виконання запитів у вигляді інтернет-сторінки (HTML) (рис. 5.7).

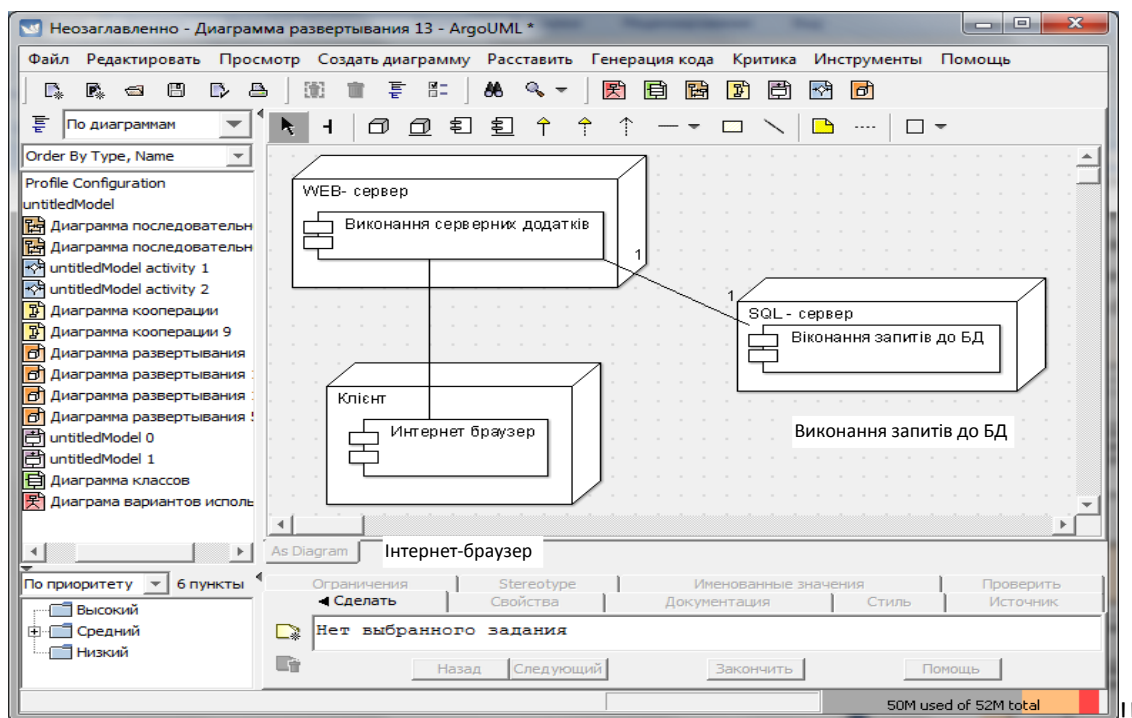


Рисунок 5.7 – Створення діаграми розгортання системи доступу до бази даних через Internet

### Контрольні питання

1. Основні функції обраної САПР, що можуть бути використані для проектування програм.
2. Методика і способи створення діаграм розгортання в обраній САПР.
3. Основні типи відношень всередині діаграм розгортання.
4. Додавання елемента в діаграму розгортання.
5. Створення взаємозв'язку між компонентами в діаграмах розгортання.
6. Відношення асоціації в діаграмах розгортання.
7. Відношення агрегації в діаграмах розгортання.

### Рекомендована література

1. Норенков И. П. Основы теории и проектирования САПР : учебник для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1990. – 335 с.
2. Норенков И. П. Системы автоматизированного проектирования электронной и вычислительной аппаратуры : учебное пособие для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1983. – 272 с.

3. Питерсон Дж. Теория сетей Петри и моделирование систем / Питерсон Дж. ; пер. с англ. – М. : Мир, 1984. – 264 с.

4. Рафикузаман М. Микропроцессоры и машинное проектирование микропроцессорных систем : в 2-х кн. / Рафикузаман М. ; пер. с англ. – М. : Мир, 1988.

5. Системы автоматизированного проектирования : учебное пособие для вузов : в 9 кн. / [под ред. И. П. Норенкова]. – М. : Высшая школа, 1986. – 159 с.

6. Системы автоматизированного проектирования в радиоэлектронике : [справочник] / Е. В. Авдеев, А. Т. Еремин, И. П. Норенков, М. И. Песков ; под ред. И. П. Норенкова. – М. : Радио и связь, 1986. – 368 с.

7. Соловьев В. В. Проектирование функциональных узлов цифровых систем на программируемых логических устройствах : учебное пособие для вузов / Соловьев В. В. – М. : ПКООО Бестпринт, 1996. – 252 с.

8. Роїк О. М. Основи автоматизованого проектування складних об'єктів і систем : навчальний посібник / Роїк О. М., Савчук Т. О., Ткаченко О. М. – Вінниця : ВДТУ, 2001. – 111 с.

## ЛАБОРАТОРНА РОБОТА № 6

### ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ. ПОБУДОВА СТРУКТУРНИХ СХЕМ ПРИСТРОЇВ В ОБРАНІЙ САПР

Мета: загальна характеристика систем автоматизованого проектування. Побудова структурних схем пристроїв в обраній САПР.

#### Порядок виконання роботи

1. Ознайомитись з системою автоматизованого проектування.
2. Використовуючи інструментарій САПР, виконати дії щодо створення бібліотечних елементів відповідно до індивідуальних завдань.
3. Сформуванню синтезовану структуру як об'єкт, встановивши параметри компонентів, та створити власну бібліотеку компонентів.
4. Оформити звіт.
5. Зробити висновок щодо доцільності обраної САПР для проектування програмного забезпечення.

**Якщо ви будете використовувати САПР PULSONIX 4.5, то порядок виконання роботи передбачає такі дії:**

1. Ознайомитись з системою автоматизованого проектування PULSONIX, що використовується для подання структурних схем.

2. За допомогою програми Library Generator Toolkit, що входить до складу PULSONIX, необхідно створити нову бібліотеку та додати до неї існуючі бібліотеки Fairchild.ssl, Lt.ssl. Переглянути вміст першої бібліотеки, замінити символ елемента FAIRCHILD-39 на LT-1. Зберегти новий файл на диску.

3. У програмі PULSONIX Schematic Design, яка використовується для побудови схем, побудувати блок-схему пошуку мінімального числа у масиві.

4. Сформувати синтезовану структуру як об'єкт та занести її до бібліотеки.

5. Оформити звіт.

6. Зробити висновок щодо доцільності обраної САПР для побудови структурних схем.

## Теоретичні відомості

### *САПР PULSONIX v. 4.5*

Призначення системи PULSONIX – проектування багатошарових друкованих плат як цифрових, так і аналогових, а також мішаних пристроїв.

PULSONIX виконує повний цикл проектування друкованих плат, що містить графічне введення схем, моделювання роботи схеми, розміщення схеми на друкованій платі, ручне розміщення компонентів, ручне, інтерактивне чи автоматичне розведення друкованих плат, контроль помилок у схемі та друкованій платі, а також випуск документації. Ця САПР забезпечує можливість перегляду розробленої друкованої плати у 3-D форматі.

Результати роботи системи можуть бути безпосередньо використані на виробництві, оскільки програма PCB Design, що входить до складу САПР, генерує файли для фотоплотерів Gerber.

### *Структура пакета прикладних програм PULSONIX*

**SchematicDesign** – програма, що призначена для проектування принципних схем. Особливості цієї програми такі: можливість розробки як простої принципної схеми, так і частини схеми складного проекту; можливість введення ієрархічних зв'язків для розподілення складного проекту на блоки (блочно-ієрархчний підхід); наявність великого набору бібліотек з необхідними наборами елементів для будь-якої схеми. Також до складу програми входить потужний графічний редактор, що дозволяє створювати графічні примітиви.

**Simulation** – програма для моделювання роботи розробленої схеми пристрою. Ця програма пов'язана з редактором принципних схем **SchematicDesign**, тому моделювання роботи розробленої схеми можна проглядати під час розробки самої схеми пристрою.

**PCBLayout** – потужна програма для створення друкованих плат. Ця програма підтримує найновіші технології щодо розробки друкованих плат.

Кількість шарів плати обмежена 8. Графічний редактор друкованих плат **PCBLayout** може бути викликаний автономно чи з редактора схем **SchematicDesign**. В останньому випадку автоматично генерується список з'єднань схеми та на поле друкованої плати переносяться зображення корпусів компонентів з вказанням електричних сполучень між ними. Ця операція називається упакуванням схеми на друковану плату. Після цього креслиться контур друкованої плати, на ньому розміщуються компоненти і починається трасування провідників. Розроблену друковану плату можна відразу ж побачити у 3-D варіанті, тобто у такому вигляді, в якому вона буде після виготовлення.

**AutoroutingandAutoplacement** – програма для трасування друкованих плат (автотрасувальники). Трасування провідників є найскладнішою задачею всього комплексу проектування друкованих плат, трудомісткість цього етапу найвища, тому застосування машини для виконання перебору великої кількості варіантів розміщення провідників є необхідним.

Для врахування досвіду людини в редакторі **PCBLayout** поряд з режимами автотрасування існують режими ручного трасування провідників.

**Library Generator Toolkit** – менеджер бібліотек. За допомогою вбудованого графічного редактора можна виконувати створення нових графічних компонентів, що відповідатимуть позначенням нових елементів, а також модифікувати існуючі. Так, оскільки система PULSONIX є закордонною розробкою, вона “не знає” про стандарти ГОСТ. Але за допомогою програми **Library Generator Toolkit** досить легко адаптувати існуючі елементи під стандарти ГОСТу.

Також за допомогою **Library Generator Toolkit** можна виконувати розбиття та злиття бібліотек, об'єднувати окремі файли з багатьох бібліотек в одну (це необхідно, наприклад, щоб усі елементи вже готового проекту скомпонувати в окрему бібліотеку).

Текстова інформація, що міститься у бібліотеках, координується за допомогою тієї ж програми. Вся інформація про компонент та його атрибути вноситься до окремих таблиць, зручних для перегляду і редагування. Таким чином, підвищується достовірність та узгодженість введення цієї інформації. Присутні потужні можливості щодо пошуку, перевірки та створення звітів за компонентами.

Зазвичай, існує сумісність системи PULSONIX з бібліотеками інших потужних редакторів схем, такими як Accel, Cadstar, PADS, P-CAD, Protel, Ultimate.

**Importnetlists** – програма для експорту файлу з'єднань схеми із середовищами проектування: Viewdraw, OrCAD, MultiSIM, EDIF 2.0.0.

**Linkstoexternalautorouters** – потужний провідник для зв'язку з трасувальниками таких середовищ проектування електронних пристроїв: Cadence Specetra, Zuken Route Editor.

**Creationofmanufacturingdata** – потужний редактор для створення звітів щодо створених проектів як цифрових так і аналогових електронних пристроїв.

**ChipPackagingToolkit** – програма для створення і редагування опорної поверхні друкованої плати, забезпечує можливість створення отворів, штампів, контактів між блоками схеми.

Також в системі реалізований багатовіконний інтерфейс (MDI), що дає можливість одночасно працювати як зі схемами, так і з їхніми компонентами або друкованими платами.

Багатовіконність дає можливість більшою мірою застосовувати ієрархічні підходи під час проектування пристрою. Наприклад, в одному вікні можна працювати зі схемою, компоненти якої розробляються в іншому вікні, при цьому відповідні внесення у схеми більш високого рівня ієрархії будуть вноситись автоматично. При роботі з друкованою платою її також можна розглядати на різних рівнях ієрархії, наприклад, можна міняти розташування елементів на ній (більш високий рівень) чи розташування виводів окремого елемента чи навіть графіка окремих контактних площадок (більш низький рівень ієрархії).

**Якщо ви будете використовувати САПР Schemagee, то порядок виконання роботи передбачає такі дії:**

1. Ознайомитись з системою проектування принципів схем Schemagee.

2. Використовуючи інструментарій САПР Schemagee, виконати дії щодо створення бібліотечних компонентів відповідно до індивідуальних завдань, які видані викладачем.

3. Використовуючи інструментарій САПР Schemagee, виконати дії щодо створення бібліотечних елементів до кожного створеного компонента відповідно до індивідуальних завдань.

4. Провести імпортування компонентів за допомогою САПР PCAD.

5. Сформувані синтезовану структуру як об'єкт, встановивши параметри компонентів та створити власну бібліотеку компонентів.

6. Зробити висновок щодо доцільності обраної САПР для побудови принципів схем та оформити звіт по роботі.

## **Теоретичні відомості**

### ***САПР Schemagee***

Система автоматизованого проектування принципів схем Schemagee була створена групою розробників GeeTeeSoft. Основна маса доробок, які були внесені в останню версію системи Schemagee 2.5, була здійснена відповідно до побажань користувачів і направлена на досягнення більш тісної інтеграції цієї САПР в існуючі потоки проектування. Загальний вигляд головного вікна САПР Schemagee поданий на рисунку 6.1.



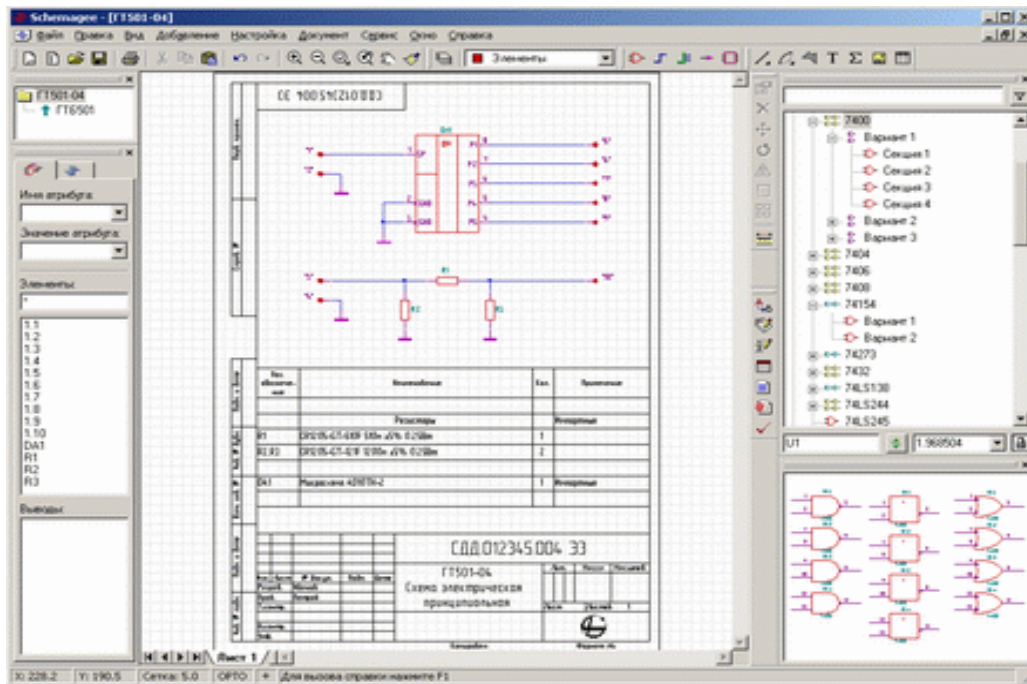


Рисунок 6.1 – Загальний вигляд головного вікна САПР Schemagee

Остання версія САПР Schemagee містить і підтримує такі основні можливості:

- реалізований новий більш зручний спосіб додання елементів на схему, що проектується;
- створено додатковий механізм налаштування та перейменування різних об'єктів схеми;
- можливість розмноження масивів графічних об'єктів;
- можливість розміщення на полі схеми рисунків у форматах BMP, JPEG, GIF;
- додано нову можливість заповнення основного надпису схеми;
- можливості імпорту та експорту у різних широко відомих форматах, ця можливість більш детально описана нижче;
- можливість редагування у одному діалоговому вікні всіх атрибутів, компонентів схеми або бібліотеки;
- реалізована функція автоматичного збереження файлу схеми і бібліотеки, що є досить корисним у проектуванні великих і складних схем.

Програма Schemagee є першим російським редактором, що дозволяє оформляти принципові схеми у строгій відповідності з вимогами ГОСТ під керуванням операційної системи Windows 98/ME/2000/XP.

Однак, на відміну від аналогічних рішень, побудованих на базі системи AutoCAD або інших САПР, редактор Schemagee є саме EDA продуктом, тобто він призначений для розробки електронних пристроїв і має на виході список з'єднань, який можна використовувати для проектування друкованих плат або моделювання.

Редактор Schemagee – це автономний додаток, що надає користувачу можливість розробляти власні бібліотеки компонентів і будувати на їх ос-

нові принципові схеми будь-якої складності та рівня ієрархії. Компоненти можуть мати декілька (в том числі і неоднорідних) секцій, а також довільну кількість атрибутів. Користувач має можливість підключати або імпортувати раніше розроблені бібліотеки системи P-CAD 4.5–8.7 і 200x, а також вилучати інформацію про компоненти з готових проектів. Гнучкість налагодження імпорту даних дозволяє правильно завантажувати бібліотеки, підготовлені у нестандартних масштабах, наприклад у “російському дюймі”.

Створені в редакторі схеми можуть містити блоки, які можуть знаходитися як у зовнішньому файлі, так і на кресленні основної схеми. Потужна система верифікації дозволяє виявити більшість основних помилок, що виникають при розробці схеми. Подальша робота над проектом здійснюється за допомогою програми TDD, яка створена тією ж групою розробників, дані в цю програму передаються через список матеріалів, що використовуються (BOM – Bill of Material), або напряму через COM-інтерфейс.

Інтеграція з програмою TDD дає можливість створювати схеми з розташованими на них же переліками елементів, що допускається ГОСТ, але це неможливо в жодній іншій подібній системі проектування.

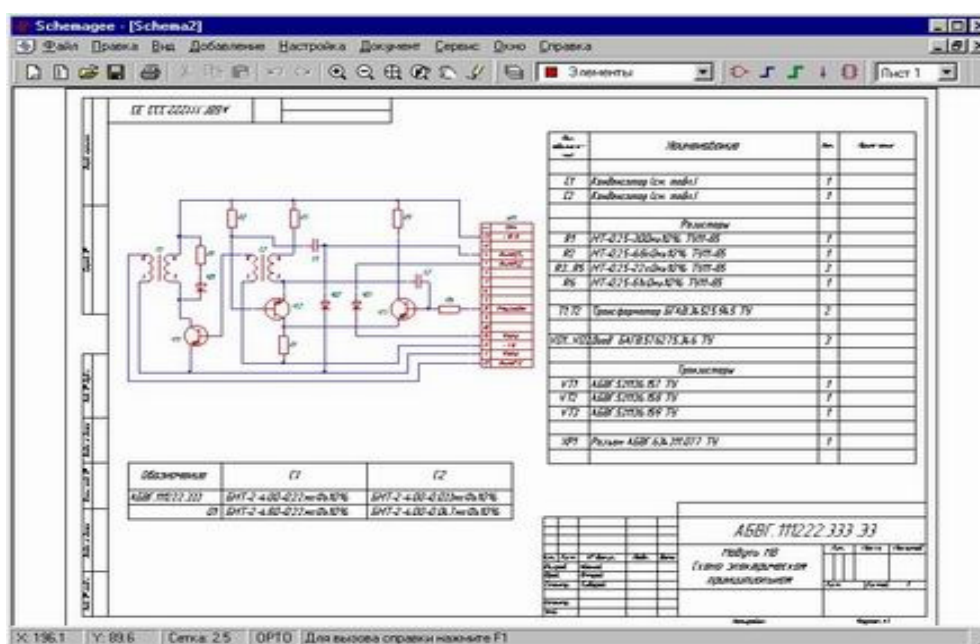


Рисунок 6.2 – Розташування на схемі переліку елементів

Редактор Schemagee також має можливості експорту схеми у форматі DXF, а також генерації списків з'єднань у форматах Tango, P-CAD ASCII, P-CAD 4.5 ALT, SPICE і Protel. Нижче показано результат імпорту списків з'єднань однієї і тієї ж схеми, розробленої в редакторі P-CAD 2002 Schematic и Schemagee (рис. 6.3). Підтримка популярного формату Tango робить Schemagee сумісною з більшістю сучасних редакторів друкованих плат, наприклад, Protel DXP чи OrCAD.

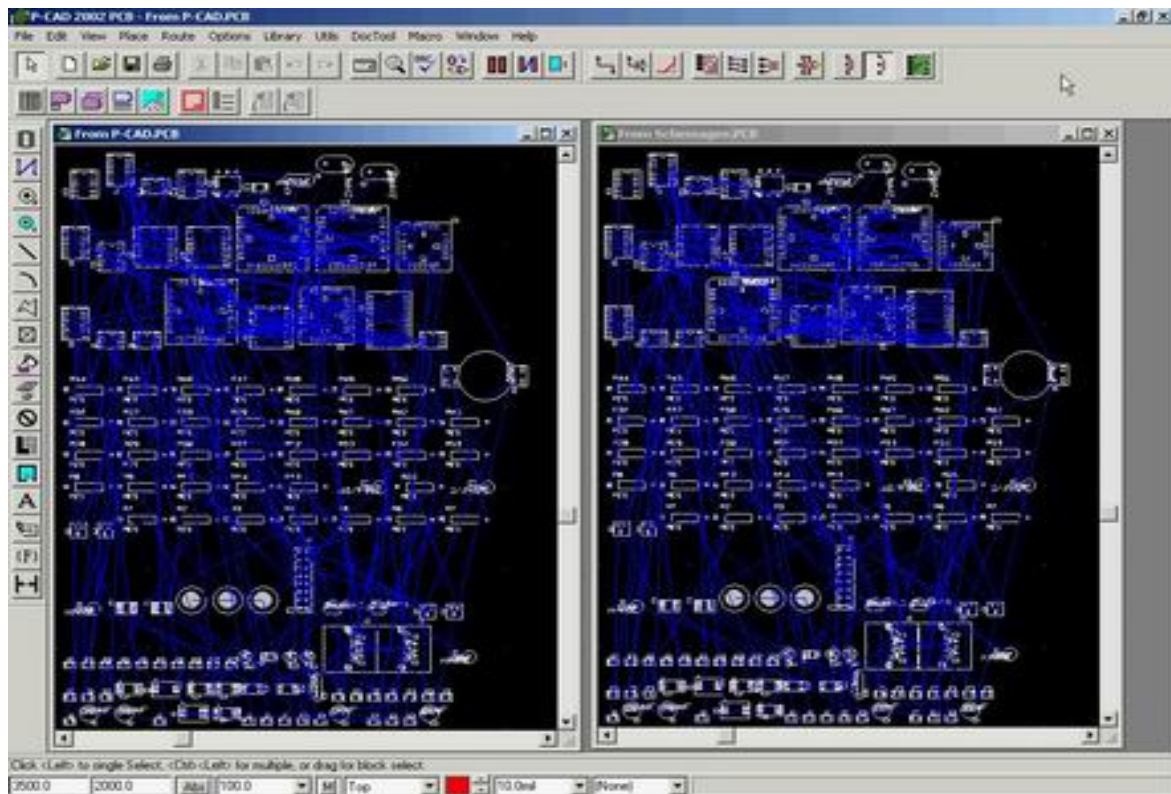


Рисунок 6.3 – Порівняння схем, створених за допомогою різних САПР

**Якщо ви будете використовувати САПР Target 3001!, то порядок виконання роботи передбачає такі дії:**

1. Ознайомитись з системою проектування принципів схем Target 3000!.
2. Використовуючи інструментарій САПР Target 3001!, виконати дії щодо створення бібліотечних компонентів відповідно до індивідуального завдання, що видане викладачем.
3. Використовуючи інструментарій САПР Target 3001!, виконати дії щодо створення бібліотечних елементів до кожного створеного компонента відповідно до індивідуального завдання.
4. Сформуванню синтезовану структуру як об'єкт, встановивши параметри компонентів, та створити власну бібліотеку компонентів.
5. Зробити висновок щодо доцільності обраної САПР для побудови принципів схем та оформити звіт по роботі.

### Теоретичні відомості

#### **САПР TARGET 3001!**

Одним з базових понять комп'ютерної графіки є поняття графічного примітива – об'єкта, який є неподільним, тобто не складається з інших об'єктів. До двовимірних графічних примітивів належить точка, пряма, коло, багатокутник тощо, до тривимірних – куб, сфера, піраміда, призма, циліндр тощо. Редактор **Target 3001!** належить до категорії двовимірних

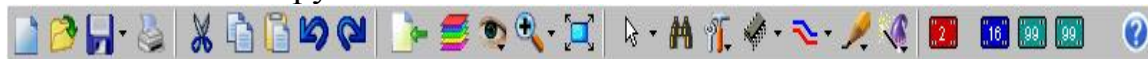
графічних редакторів та має всі основні графічні примітиви та деякі додаткові.



Toolbar з схематичної точки зору:



Toolbar з точки зору РСВ:















**toolbar** – показує найважливіші команди TARGET як піктограми.






### Важливо:

toolbar може бути активовано лише за допомогою миші, і не всі команди дублюються гарячими клавішами.






### Основні піктограми

-  – створення нового проекту (SheetP.T3001 для РСВ);
-  – відкриття існуючого проекту;
-  – зберігання проекту;
-  – друк поточного вигляду проекту;
-  – вирізання і копіювання в буфер;
-  – копіювання в буфер;
-  – вставка з буфера;
-  – відміна/повтор останньої дії;
-  – перетворення у вигляд РСВ;
-  – перетворення в схематичний вигляд;
-  – вибір схематичної сторінки;
-  – шари РСВ.









 Вигляд:

-  – коефіцієнт масштабування;
-  – сітка;
-  – вибір кольору;
-  – додаткове схематичне вікно;
-  – додаткове вікно РСВ.

 Масштаб:







-  – зменшення;
-  – збільшення;
-  – оновлення поточного екрана;
-  – центрування;
-  – коефіцієнт масштабування.

 Показчик:




-  – виділення тільки обраного сигнального сегмента;
-  – виділення тільки комбінованого сигнального сегмента;
-  – виділення повного сигналу;
-  – точно зігнутий;
-  – зігнуте дугою кола;
-  – зігнуте в криву;
-  – якщо цей засіб активовано, вибрана компонента завжди виділяє цілий сегмент/пакет;
-  – якщо цей засіб активовано, кожен обраний елемент highlights є цілим символом/пакетом.


 Пошук компоненти.


 Редактор:








-  – відстані міри;
-  – вибір і виділення єдиного елемента;
-  – видалення вибраних елементів;
-  – переміщення виділених елементів;
-  – відображення горизонтально;
-  – відображення вертикально;
- **E** – редагування (зміна) виділених елементів.

 Включити пакетний символ в РСВ:

-  – експорт вибраних елементів як пакет в пакетну бібліотеку;
-  – початок перегляду бібліотечного вікна;
-  – видалення пакета з бібліотеки.






 Формування сигнальних зв'язків (schematic):

-  – безпосередній сигнальний зв'язок;

-  – формування сигнального зв'язку спочатку в горизонтальному напрямку;
-  – формування сигнального зв'язку спочатку у вертикальному напрямку;
-  – формування сигнального зв'язку спочатку в ортогональному напрямку;
-  – формування сигнального зв'язку спочатку в діагональному напрямку;
-  – чотири індивідуально запрограмованих опції для фіксації шаблону за замовчуванням при формуванні сигнальних зв'язків;
-  – встановлення з'єднання;
-  – встановлення шини.









#### Формування зв'язків на друкованій платі:

-  – формування доріжки безпосередньо;
-  – формування доріжки спочатку в горизонтальному напрямку;
-  – формування доріжки спочатку у вертикальному напрямку;
-  – формування доріжки спочатку в ортогональному напрямку;
-  – формування доріжки спочатку в діагональному напрямку;

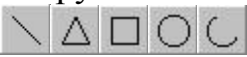











– чотири індивідуально запрограмованих опції фіксації шаблону за замовчуванням при формуванні доріжок;

-  – редагування сформованої доріжки;
-  – встановлення з'єднання;
-  – встановлення мосту;
-  – панель паяльника;
-  – формування спіралі, редагування спіральних доріжок;
-  – формування навісним проводом (ratsnest) (тільки для РСВ без schematic).














#### Інструментальні засоби:

-  – порожні елементи;
-  – непорожні елементи;
-  – вибір в схематичному режимі встановлення штира/з'єднання/шини;
-  – режим РСВ: ratsnest;

-  – введення тексту;
-  – логічні операції: AND, OR, NOT, EXOR буфер;
-  – створення компонентів згідно з DIN/IEEE;
-  – зовнішній опис області для створення компонентів згідно з DIN/IEEE;
-  – внутрішній опис області для створення компонентів згідно з DIN/IEEE;
-  – креслення елементів.




#### Автоматичні функції:

-  – відстань міри;
-  – штиркове з'єднання (свопінг) (тільки в схематичному режимі);
-  – з'єднання (свопінг) паяльником (тільки в режимі РСВ);
-  – автотрасування (тільки в режимі РСВ);
-  – автомаршрутизатор (тільки в режимі РСВ);
-  – діалог для перерахування ratsnest (тільки в режимі РСВ);
-  – підтримка groundplane (тільки в режимі РСВ);
-  – контрольний проект (схематичний і РСВ);
-  – змішаний режим моделювання schematic;
-  – ЕМС-аналіз РСВ;
-  – реорганізація проекту (оптимізація внутрішньої бази даних).

### ***Побудова прямих ліній***

Для побудови прямих ліній використовується меню Elements-Draw Line. Вибираємо цю команду, курсор миші перетворюється на хрест. Підводимо мишу до початкової точки лінії та натискаємо ліву кнопку миші. Не відпускаючи кнопку, ведемо курсор миші до кінцевої точки лінії, де відпускаємо ліву кнопку. Для того, щоб продовжити побудову лінії, знову натискаємо ліву кнопку, при цьому початковою точкою нового сегмента лінії буде кінцева точка попереднього сегмента. Таким чином можна отримати навіть замкнений багатокутник. Для того, щоб зафіксувати останню точку лінії, достатньо натиснути кнопку Esc клавіатури.

### ***Редагування параметрів прямих ліній***

Для того, щоб змінити параметри лінії, необхідно на панелі інструментів вибрати кнопку  та натиснути мишею на обраний відрізок. Натисніть правою кнопкою мишки всередині штрихової рамки та виберіть

“Properties” з низхідного меню. У вікні властивостей, що відкриється, ви зможете встановити початкову та кінцеву координату відрізка, а також стиль, колір та товщину.

### ***Побудова дуг та кіл***

Для того, щоб побудувати дугу або коло, виберіть пункт меню Elements – Draw Arcs/Circles.

Для побудови дуги натисніть лівою кнопкою мишки в необхідній вам точці та переміщуйте курсор мишки доти, доки не отримаєте хорду дуги, яка вам потрібна.

Для побудови кола натисніть лівою кнопкою миші в певній точці рисунка. Відпустіть кнопку, наведіть курсор мишки у центр майбутнього кола та ще раз натисніть та відпустіть ліву кнопку миші.


Якщо побудована дуга чи коло відрізняються від потрібної, то їх параметри завжди можна змінити за допомогою такої самої процедури, що описана для зміни параметрів прямих ліній. Для дуги можна встановити товщину лінії, центральну точку, радіус, початковий та кінцевий кути. Потрібно зауважити, що редактор **Target 3001!** не дозволяє дузі/колу мати штрихові контури.

### ***Побудова багатокутників***

Побудова багатокутників схожа на побудову ліній. Для цього використовується команда меню Elements – Draw Filled Polygon.

Для побудови багатокутників необхідно вибрати першу точку багатокутника та натиснути лівою кнопкою миші, потім другу, третю і так далі. По завершенню необхідно натиснути один раз правою клавішею миші. При цьому побудований багатокутник зафарбовується сірим кольором.


Необхідно зауважити, що в системі Target 3001! можна будувати лише опуклі багатокутники, тобто такі, сторони яких не перетинають самі себе. У випадку порушення цього правила буде видано попередження та багатокутник побудований не буде.

Якщо за допомогою кнопки  вибрати багатокутник, то він стане підсвічений жовтим, а всі вершини та середини його сторін будуть відмічені білими квадратами. За допомогою перетаскування мишею цих квадратиків можна редагувати вже створений багатокутник.

### ***Нанесення написів***

Для нанесення написів потрібно вибрати команду меню Elements – Insert Text. Далі в необхідному місці креслення натиснути лівою кнопкою миші. На екрані з’явиться вікно Text, де потрібно ввести текст, вибрати для нього вирівнювання та стиль.



Якщо за допомогою кнопки  вибрати напис та викликати вікно Properties, то з'явиться те саме вікно, що й при введенні тексту, в якому можна буде виконувати такі ж маніпуляції.

### ***Видалення об'єктів***

Щоб видалити один об'єкт, виділіть його мишею та натисніть клавішу Delete. Для видалення групи об'єктів за допомогою миші замкніть цю групу у рамку й також натисніть Delete.

### ***Копіювання та вставка об'єктів***

Редактор Target 3001! дозволяє працювати з буфером обміну Windows так само, як і в будь-якій Windows-програмі. Для копіювання об'єкта в буфер виділіть його та натисніть Ctrl-C або викличте контекстне меню правою кнопкою миші і виберіть Copy. Щоб вставити об'єкт з буфера, скористайтесь комбінацією клавіш Ctrl-V чи командою Paste з контекстного меню. При вставці, якщо натиснути правою кнопкою миші, об'єкт обертається на 90 градусів.

## **Контрольні питання**

1. Дайте характеристику обраної системи автоматизованого проектування.
2. Які програми входять до складу обраної системи автоматизованого проектування?
3. Що таке бібліотечний елемент? Як використовуються бібліотечні елементи на рівні структурного проектування?
4. Який інструментарій обраної вами САПР використовується для створення бібліотечних елементів?
5. Як формується об'єкт?
6. Що таке власна бібліотека компонентів та для чого вона використовується?

## **Рекомендована література**

1. Норенков И. П. Основы теории и проектирования САПР : учебник для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1990. – 335 с.
2. Норенков И. П. Системы автоматизированного проектирования электронной и вычислительной аппаратуры : учебное пособие для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1983. – 272 с.
3. Питерсон Дж. Теория сетей Петри и моделирование систем / Питерсон Дж. ; пер. с англ. – М. : Мир, 1984. – 264 с.

4. Рафикузаман М. Микропроцессоры и машинное проектирование микропроцессорных систем : в 2-х кн. / Рафикузаман М. ; пер. с англ. – М. : Мир, 1988.

5. Системы автоматизированного проектирования : учебное пособие для вузов : в 9 кн. / [под ред. И. П. Норенкова]. – М. : Высшая школа, 1986. – 159 с.

6. Системы автоматизированного проектирования в радиоэлектронике : [справочник] / Е. В. Авдеев, А. Т. Еремин, И. П. Норенков, М. И. Песков ; под ред. И. П. Норенкова. – М. : Радио и связь, 1986. – 368 с.

7. Соловьев В. В. Проектирование функциональных узлов цифровых систем на программируемых логических устройствах : учебное пособие для вузов / Соловьев В. В. – М. : ПКООО Бестпринт, 1996. – 252 с.

8. Роїк О. М. Основи автоматизованого проектування складних об'єктів і систем : навчальний посібник / Роїк О. М., Савчук Т. О., Ткаченко О. М. – Вінниця : ВДТУ, 2001. – 111 с.

## **ЛАБОРАТОРНА РОБОТА № 7**

### **ПОБУДОВА ПРИНЦИПОВИХ СХЕМ ПРИСТРОЇВ ТА МОДЕЛЮВАННЯ ПРОЦЕСІВ ЇХ ФУНКЦІОНУВАННЯ В ОБРАНІЙ САПР**

Мета: вивчення методів побудови принципів схем та генерація переліку з'єднань в обраній САПР.

#### **Порядок виконання роботи**

1. Ознайомитися з програмами обраної САПР, що можуть бути використаними для побудови принципів схем пристроїв.

2. Оволодіти методикою і способами створення принципів схем в обраній САПР.

3. Враховуючи блочно-ієрархічний метод проектування складних об'єктів та систем, синтезувати структурну схему пристрою за індивідуальним завданням.

4. Синтезувати функціональну схему пристрою за індивідуальним завданням з урахуванням його структури.

5. Обрати відповідно до функціональної схеми спроектованого пристрою та з урахуванням можливостей бібліотек обраної САПР компоненти його принципової схеми.

6. Побудувати схему пристрою на основі стандартних обраних компонентів в рамках програми САПР.

7. Використовуючи засоби обраної САПР, забезпечити генерацію переліку з'єднань створеної принципової схеми пристрою.

8. Оформити звіт.

**Якщо ви будете використовувати САПР PULSONIX 4.5, то порядок виконання роботи передбачає такі дії:**

1. Ознайомитися з програмами ППП PULSONIX, що можуть бути використаними для побудови принципових схем пристроїв.

2. Оволодіти методикою і способами створення принципових схем в ППП PULSONIX.

3. Враховуючи блочно-ієрархічний метод проектування складних об'єктів та систем, синтезувати задану структурну схему.

4. Синтезувати функціональну схему відповідно до індивідуального завдання.

5. Обрати відповідно до функціональної схеми та з урахуванням можливостей бібліотек ППП PULSONIX компоненти його принципової схеми.

6. Побудувати схему на основі стандартних обраних компонентів в рамках програми Schematic Design. Для цього необхідно виконати такі дії:

6.1. Ввійти в систему PULSONIX в програму Schematic Design.

6.2. Створення принципової схеми:

6.2.1. В режим розміщення символів компонентів на схемі переходять за командою Insert>Component.

6.2.2. Обирається бібліотека у рядку Look in ...

6.2.3. Вибирається компонент, при цьому у вікні Preview можна побачити вигляд елемента на схемі, далі за допомогою курсора в формі хрестика з розривом в центрі наноситься необхідна кількість компонентів цього типу.

6.2.4. Якщо необхідно, ввести шини, то вони встановлюються за допомогою команди Insert>Bus.

6.2.5. Встановити ланцюги між компонентами за допомогою команди Insert>Connection.

7. Використовуючи засоби ППП PULSONIX, забезпечити генерацію переліку з'єднань створеної принципової схеми трирозрядного віднімача. Для цього необхідно виконати такі дії:

7.1. Створити перелік з'єднань за командою Simulation>Create SPICE Net List File>Whole Design, у відкритому вікні вибрати файл для зберігання даних.

8. Оформити звіт.

## **Теоретичні відомості**

### **САПР PULSONIX 4.5**

ППП Pulsonix здатен виконати кожен з основних етапів проектування апаратних складних об'єктів та систем:

- етап системного (структурного) синтезу;
- етап функціонально-логічного проектування;
- етап конструкторського проектування.

Schematic Design та PCB Design – графічні редактори принципів схем і друкованих плат (ДП), що мають сучасні системи меню в стилі інших програм для Windows. Порядок створення принципової схеми такий:

1. *Завантаження бібліотек.* Перед нанесенням на схему символів компонентів за командою Library забезпечується доступ до обраних бібліотек. Обравши клавішу Add, додають імена бібліотек в список відкритих бібліотек (Open Libraries).

2. *Розташування компонентів на схемі.* В режим розміщення символів компонентів на схемі переходять за командою Insert>Component або натисканням за піктограмою “-||-”. Після цього натисканням курсором в будь-якій точці схеми відкриває меню вибору компонента. На панелі Insert Component необхідно обрати ім'я потрібного компонента із списку або ввести у верхньому рядку. Натисканням клавіші Preview дозволяє продивитись графічне зображення символу компонента. Після вибору в меню Insert>Component потрібного компонента та введення всіх необхідних параметрів натискають клавішу “OK”. Розміщення символу компонента на схемі виконується після натискання курсором в будь-якій точці робочого вікна.

3. *Розташування електричних ланцюгів.* Після вибору команди Insert>Connection курсор приймає форму хрестика. Натисканням миші позначається початкова точка ланцюга. Повторне натискання лівої кнопки миші фіксує точку злому. Присвоїти ланцюгу ім'я можливо двома способами. По-перше, індивідуально обрати кожний ланцюг і натисканням правої кнопки миші відкрити в меню редагування пункт Properties. На екрані з'явиться інформація про обраний ланцюг. У вікні Net можна змінити “Ім'я ланцюга”, замінивши в рядку Net Name, наприклад, NET00001 на A1. Це ім'я буде показуватись на схемі. Тип та розмір шрифту вказують після натискання на клавішу Text Styles.

4. *Розташування шин.* Вибір команди Insert>Bus активізує режим введення шин. Натисканням курсору відмічають початок шини та точки злому шини. Побудова шини закінчується натисканням правої кнопки миші або клавіші Esc. Після цього за командою Insert>Connection розташовують ланцюги. Імена ланцюгів, які створюють шину, задаються під'єднаними до ланцюгів за командою Insert>Page Link спеціальними портами. Після вибору цієї команди, натисканням миші на будь-якій точці схеми відкривається меню вибору порту. В рядку Net Name вказують ім'я ланцюга.

5. *Створення переліку з'єднань.* Перелік з'єднань містить перелік компонентів і ланцюгів з вказанням номерів виводів компонентів, до яких вони під'єднані. Він використовується для так званої процедури “запаковування схеми на друковану плату” – розташування на полі ДП корпусів компонентів з вказанням їх електричних зв'язків відповідно до принципової схеми. За ко-

мандою Simulation>Create SPICE Net List File>Whole Design відкривають вікно для збереження файлу з'єднань на диску.

**Якщо ви будете використовувати САПР Schemagee, то порядок виконання роботи передбачає такі дії:**

1. Створити нову схему та розмістити на ній елементи з різних бібліотек, в тому числі і власної бібліотеки елементів.
2. Використовуючи лінії з'єднання та шини, провести з'єднання елементів схеми.
3. Створити та приєднати порти до деяких виходів схеми.
4. Перевірити коректність схеми.
5. Зробити висновок щодо доцільності обраної САПР для побудови принципів схем та оформити звіт по роботі.

## Теоретичні відомості

### *САПР Schemagee*

Побудува принципової схеми на базі створеної у лабораторній роботі № 1 власної бібліотеки компонентів. Для цього натиснути на пункт меню “Создать схему” (рис. 7.1).



Рисунок 7.1 – Побудува принципової схеми

Після цього з'явиться нове вікно з рамкою, яка відповідає ГОСТ, в цьому вікні буде відображатися графічне зображення нашої принципової схеми. Оберемо пункт меню “Элемент” (рис. 7.2).

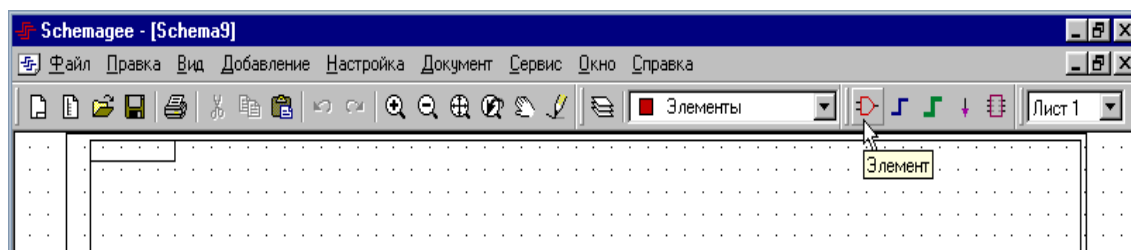


Рисунок 7.2 – Приклад вікна з рамкою, яка відповідає ГОСТ

З'явиться вікно додання нового елемента, оберемо зовнішнє джерело додання елементів та натиснемо на кнопку “Изменить список” (рис. 7.3).

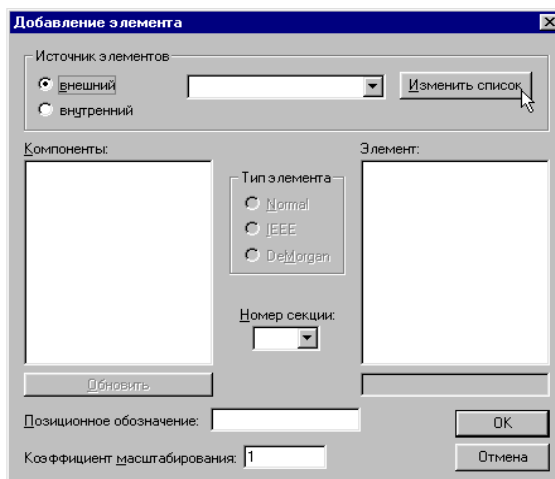


Рисунок 7.3 – Пример вікна додавання нового елемента

З'явиться вікно вибору джерела елементів, додамо у список нашу бібліотеку, вказавши відповідний файл. Також джерелами елементів можуть слугувати файли бібліотек і схем системи P-CAD (ACCEL EDA). Оберемо зі списку всі необхідні джерела елементів (рис. 7.4).

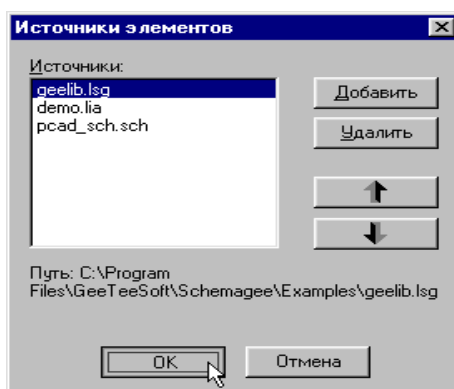


Рисунок 7.4 – Пример вікна вибору джерела елементів

Тепер ми можемо обрати створений нами раніше елемент із власної бібліотеки та помістити його на схему (рис. 7.5).

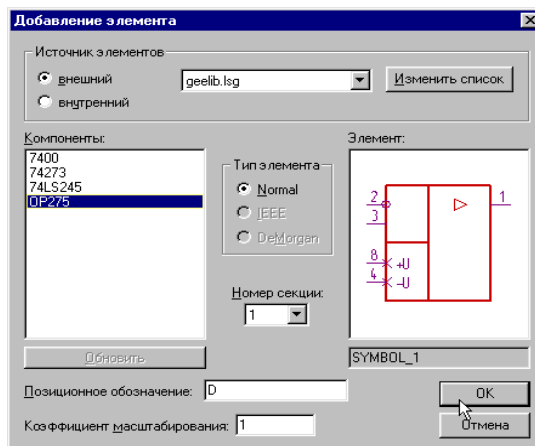


Рисунок 7.5 – Пример обрання елемента із власної бібліотеки

Додамо до схеми декілька різних елементів із різних бібліотек (рис. 7.6).

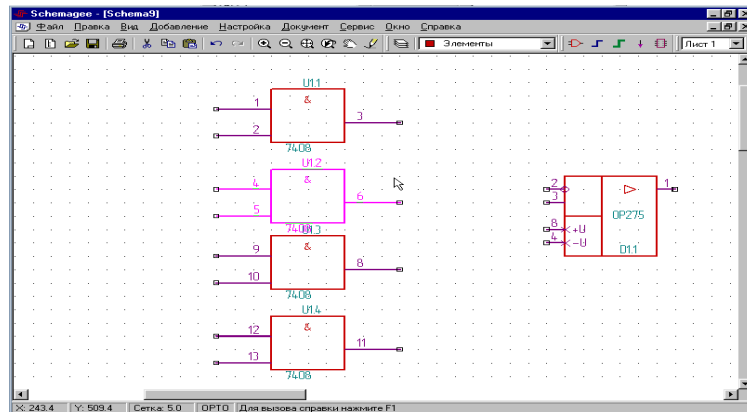


Рисунок 7.6 – Приклад додання до схеми елементів із різних бібліотек

Властивості кожного елемента можна змінювати за допомогою контекстного меню (рис. 7.7).

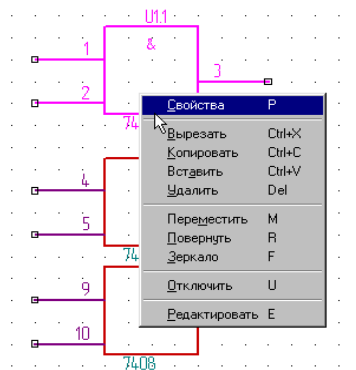


Рисунок 7.7 – Приклад контекстного меню

Вікно властивостей елемента має чотири вкладки і виглядає таким чином (рис. 7.8).

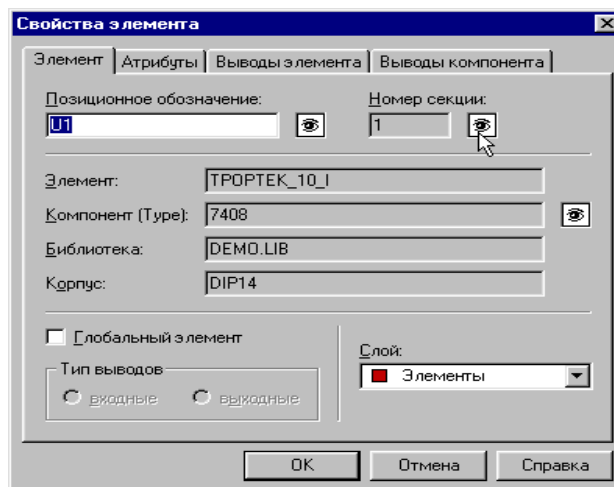


Рисунок 7.8 – Приклад вікна властивостей

За допомогою цього інструмента можна змінити будь-які властивості обраного елемента та встановити нові властивості, якщо це потрібно. Можна виділяти групу елементів та змінювати їх спільні властивості таким самим чином.

Також можна змінювати графічні зображення елементів, розміщених на схемі, для цього використовується пункт “Редактировать” контекстного меню елемента (рис. 7.9).

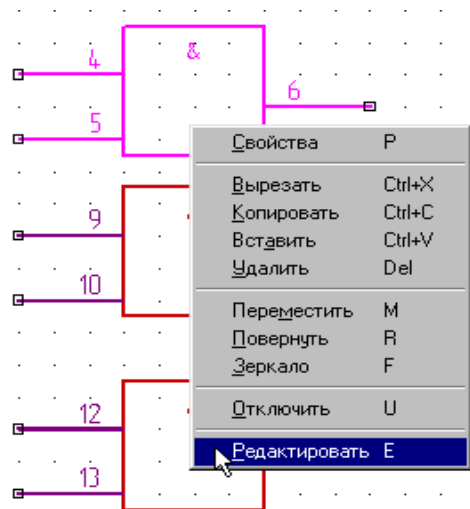


Рисунок 7.9 – Пункт “Редактировать” контекстного меню элемента

Після вибору цього пункту контекстного меню графічне зображення елемента відкриється для редагування у окремому вікні (рис. 7.10).

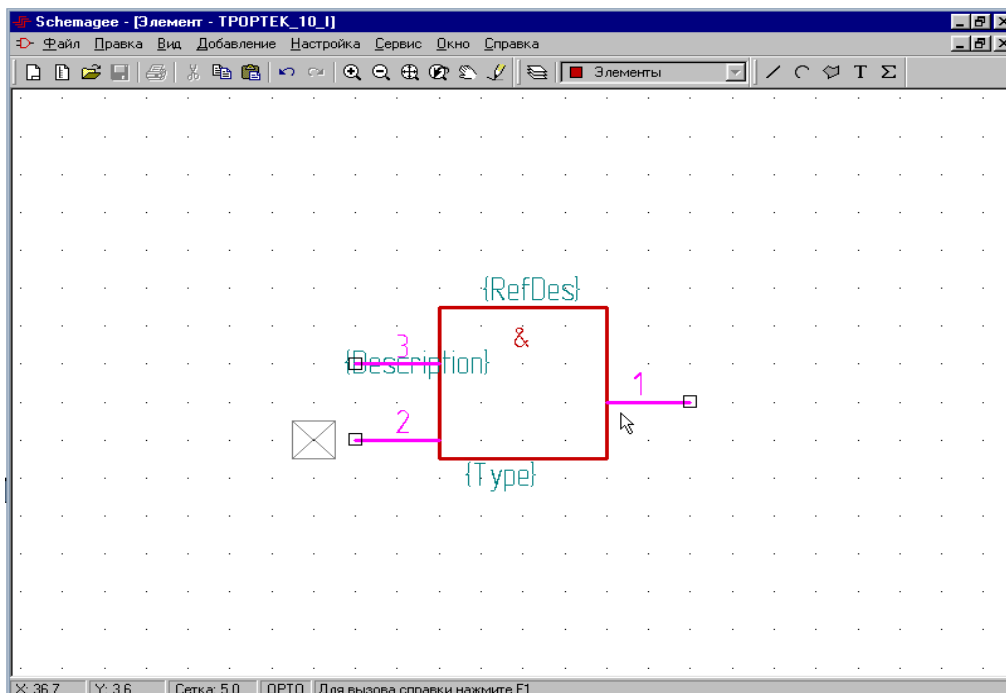


Рисунок 7.10 – Пример графического изображения элемента для редактирования у отдельного окна



Додамо ще декілька елементів до схеми із зовнішньої елементної бази бібліотеки P-CAD (рис. 7.11).

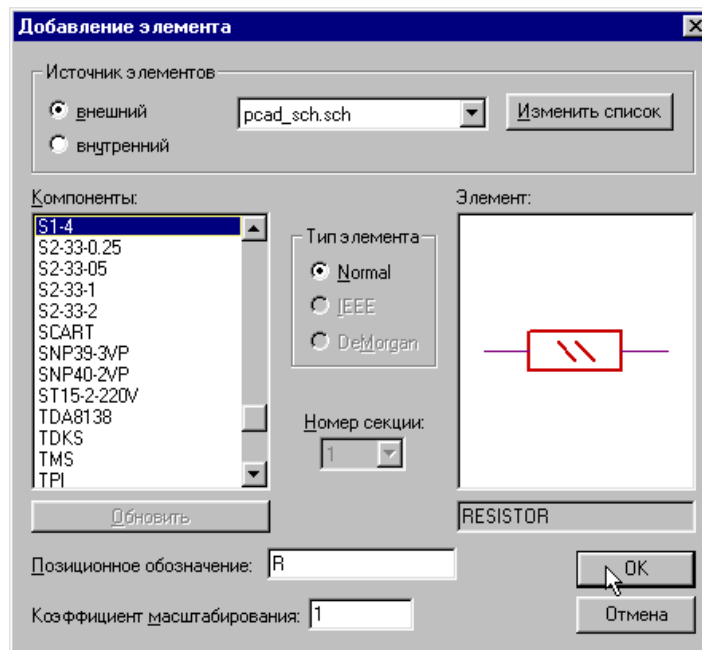


Рисунок 7.11 – Приклад додання елементів до схеми із зовнішньої елементної бази

У ліній зв'язку також є контекстне меню і їх властивості також можна редагувати (рис. 7.12).

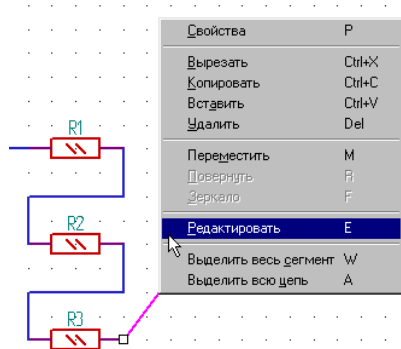


Рисунок 7.12 – Приклад контекстного меню для редагування

Додавати нові лінії зв'язку можна за допомогою відповідного пункту меню "Линии связи" (рис. 7.13).

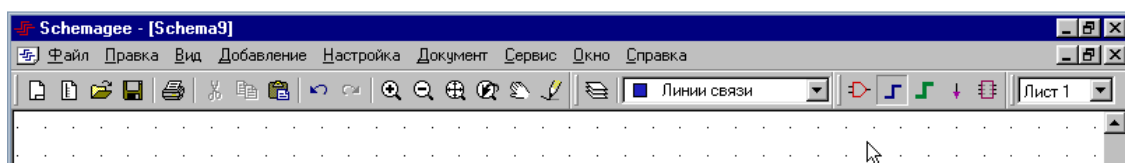


Рисунок 7.13 – Приклад додання лінії зв'язку за допомогою відповідного пункту меню "Линии связи"

На схемі можна створити декілька шарів, для цього застосуємо пункт меню “Слои” (рис. 7.14).

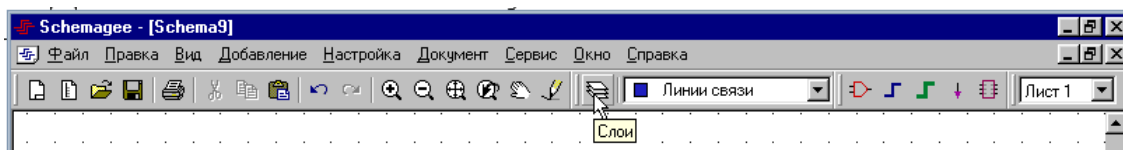


Рисунок 7.14 – Приклад застосування пункту меню “Слои”

Після цього з’явиться вікно зі списком шарів, в якому можна додати до схеми новий шар (рис. 7.15).

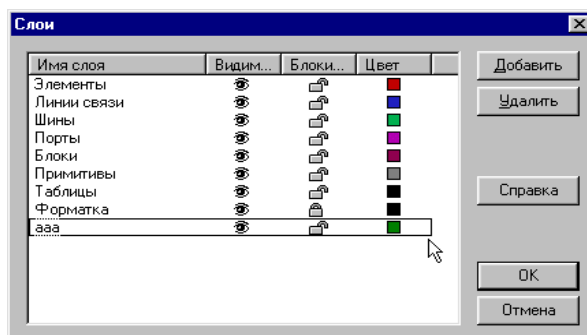


Рисунок 7.15 – Приклад вікна зі списком шарів

Вибір активного шару здійснюється за допомогою відповідного низхідного списку (рис. 7.16).

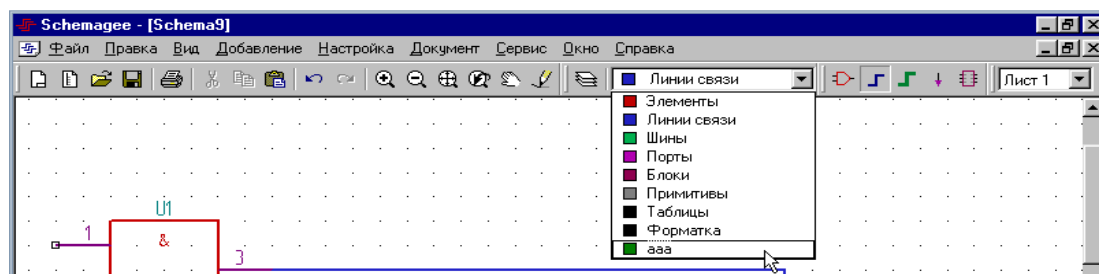


Рисунок 7.16 – Приклад вибору активного шару

Після вибору активного шару всі нові елементи, які наносяться на схему, будуть належати цьому шару.

Сегменти кіл, що розташовані на одному листі схеми, об’єднуються в одне коло за допомогою шини або портів. Створити шину можна за допомогою відповідного пункту меню “Шина” (рис. 7.17).

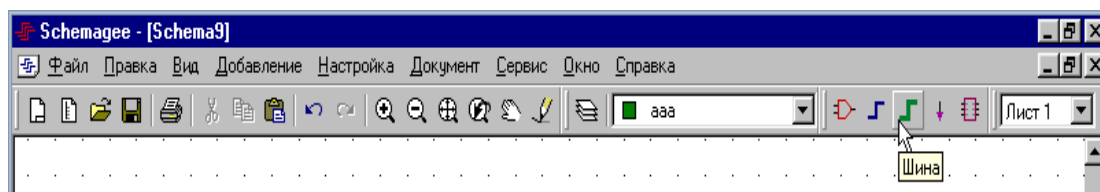


Рисунок 7.17 – Приклад створення шини

Результат створення шини та з'єднань з нею показаний на рисунку 7.18.

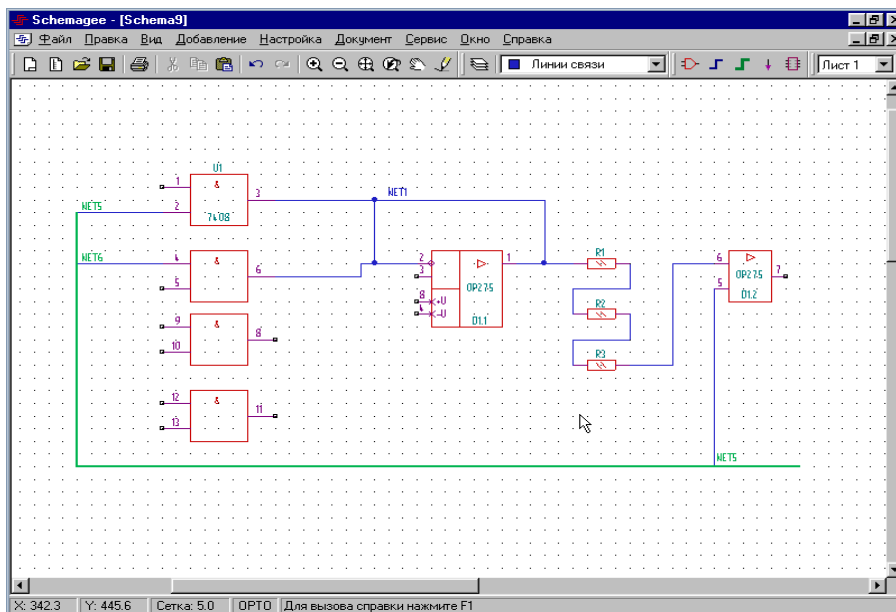


Рисунок 7.18 – Результат створення шини та з'єднань з нею

Порти створюються за допомогою відповідного пункту меню “Порт” (рис. 7.19).

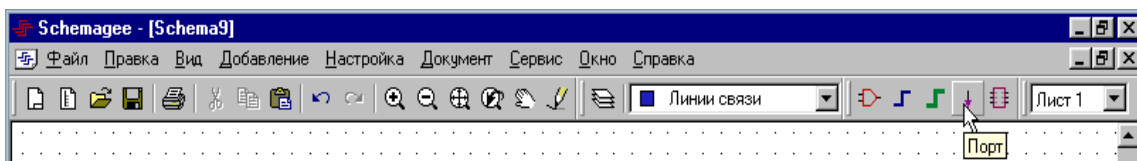


Рисунок 7.19 – Приклад створення портів

При натисканні на цей пункт меню з'являється нове вікно з налаштуваннями параметрів портів (рис. 7.20).

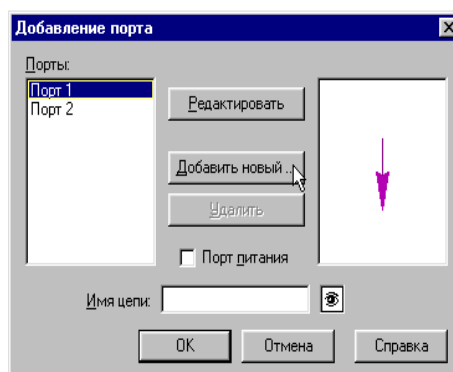


Рисунок 7.20 – Приклад нового вікна з налаштуваннями параметрів портів

При створенні нового порту його графічне зображення відкривається у новому вікні, де його можна редагувати (рис. 7.21).

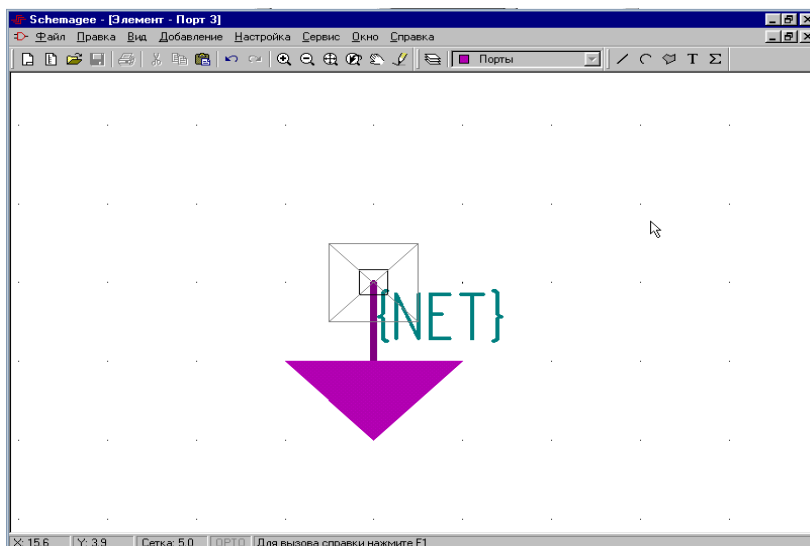


Рисунок 7.21 – Приклад нового вікна для редагування

Після редагування та збереження порту він з'являється у списку портів. З цього списку можна вибрати порт і помістити його на схему (рис. 7.22).

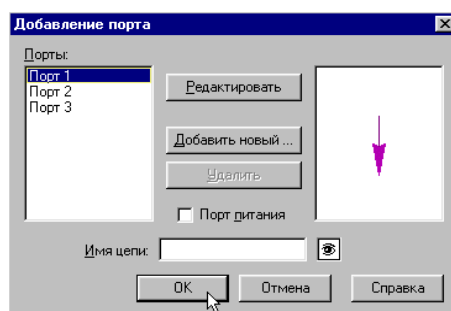


Рисунок 7.22 – Приклад списку портів

Кінцева принципова схема матиме вигляд (рис. 7.23).

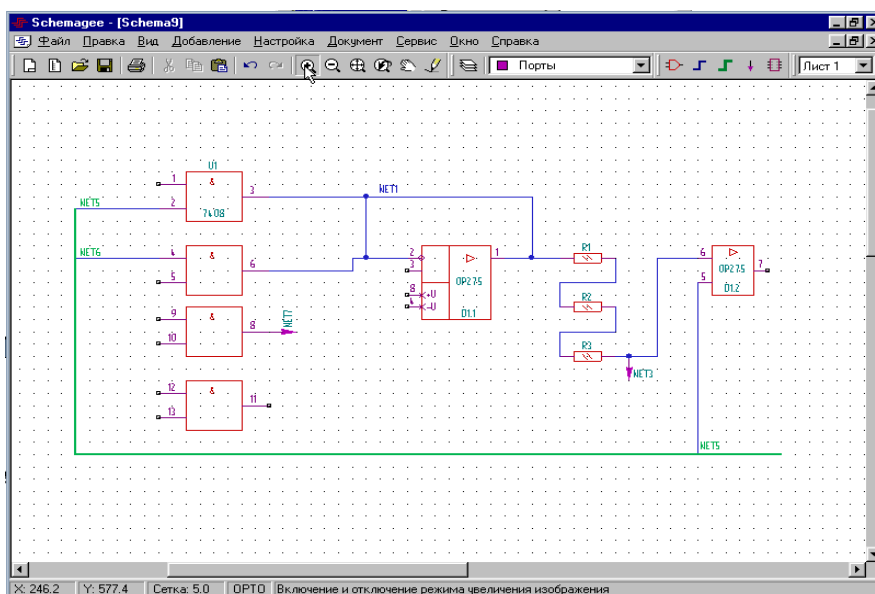


Рисунок 7.23 – Кінцева принципова схема

Шини, лінії зв'язку і порти, так само як і інші елементи схеми, мають контекстне меню, за допомогою якого можна змінювати їхні параметри. Також за допомогою портів можна з'єднувати схеми з різних листів проекту.

Перевірити правильність спроектованої схеми можна, використавши пункт головного меню програми “Сервіс→Перевірити схему” (рис. 7.24).

В результаті чого в нижній частині вікна програми з'явиться область з результатами перевірки, якщо ця область пуста, то помилок немає, в іншому випадку – в цій області відобразяться помилки, які наявні у схемі.

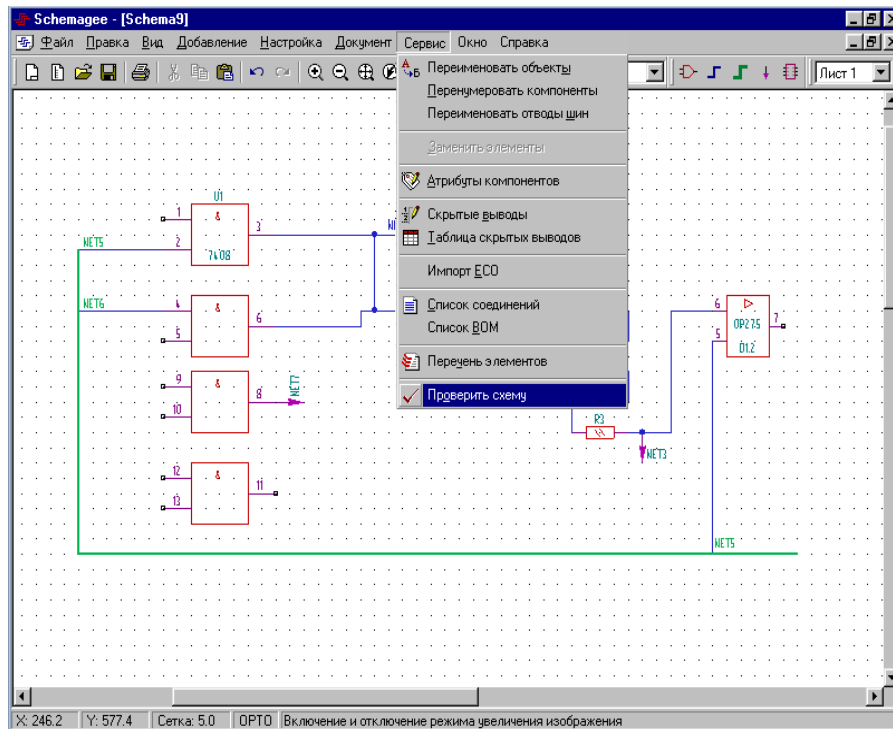


Рисунок 7.24 – Приклад перевірки правильності спроектованої схеми

**Якщо ви будете використовувати САПР TARGET3001!, то порядок виконання роботи передбачає такі дії:**

1. Створити нову схему.
2. Використовуючи лінії з'єднання та шини, провести з'єднання елементів схеми.
3. Створити та приєднати порти до деяких виходів схеми.
4. Перевірити коректність схеми.
5. Зробити висновок щодо доцільності обраної САПР для побудови принципних схем та оформити звіт по роботі.

### Теоретичні відомості

#### **САПР TARGET3001!**

Для моделювання електричної схеми необхідно:

1. Створити або відкрити проект зі схемою.

При створенні нового проекту з'являється головне вікно програми TARGET3001!, яке має такий вигляд (рис. 7.25):

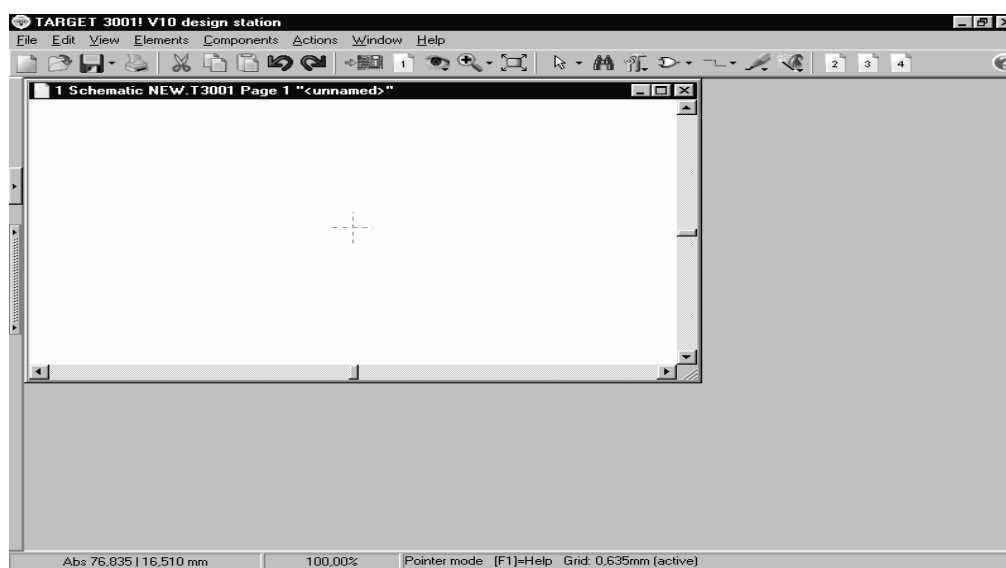


Рисунок 7.25 – Головне вікно програми TARGET3001!

**Опис команд головного меню (рис. 7.26):**

*File:*

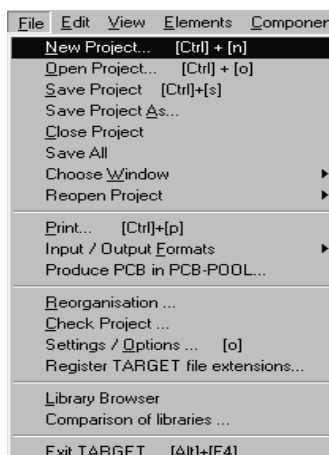


Рисунок 7.26 – Опис команд головного меню

*NewProject ...* – створення нового проекту;  
*OpenProject ...* – відкриття існуючого проекту;  
*SaveProject* – збереження відкритого проекту;  
*SaveProjectAs ...* – збереження відкритого проекту під новим ім'ям;  
*CloseProject* – закриття відкритого проекту;  
*SaveAll* – збереження усіх відкритих файлів, якщо вони були змінені;  
*ChooseWindow* – перехід між вікнами програми;  
*ReopenProject* – відкриття проектів, що нещодавно редагувалися;  
*Print ...* – виведення на друк схеми або генерування графічного файлу формату .TIF;

*ProducePCBinPCB-POOL ...* – генерування друкованої плати за допомогою PCB-POOL (засіб, що дозволяє за допомогою Інтернет замовляти виготовлення друкованих плат);

*CheckProject ...* – перевірка правильності складових проекту (правильність розміщення компонентів, перевірка на правильність з'єднань, правильність використання компонентів);

*Settings/Options ...* – дозволяє змінити широкий спектр опцій програми;

*LibraryBrowser* – пошук та редагування елементів і їх PCB-корпусів;

*Comparisonoflibraries ...* – порівняння бібліотек елементів.

Інструментарій вкладок головного меню подано на рисунках 7.27 – 7.33.

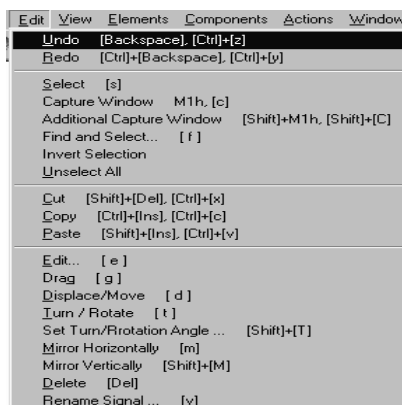


Рисунок 7.27 – Інструментарій вкладки *Edit*

*Undo/Redo* – відміна або повтор виконаної операції;

*FindandSelect ...* – пошук компонентів на схемі;

*InvertSelection* – інвертування вибраних елементів;

*UnselectAll* – зняття помітки з усіх елементів;

*Edit ...* – редагування параметрів виділеного елемента або групи;

*Displace/Move* – інструмент переміщення елементів по схемі;

*Turn/Move* – поворот виділених елементів на визначений кут;

*SetTurn/RotationAngle ...* – визначення кута повороту;

*RenameSignal ...* – перейменування виділеного провідника.

*View:*

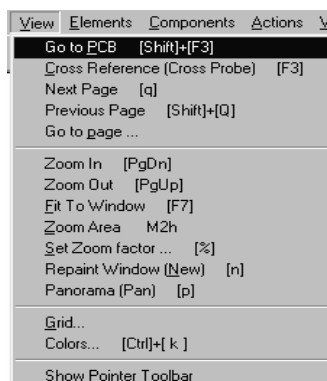


Рисунок 7.28 – Інструментарій вкладки *View*

*GotoPCB* – перехід до вікна монтажного розміщення компонентів;  
*CrossReference* – перехід до вікна монтажного розміщення компонентів з переходом до виділеного елемента та навпаки;  
*NextPage, PreviousPage, GotoPage* – переміщення між сторінками схеми;  
*ZoomIn, ZoomOut, ...* – зміна масштабу при редагуванні та перегляді схеми;  
*RepaintWindow* – оновлення екрана та повторне рисування схеми;  
*Panorama* – зручний засіб переміщення по схемі;  
*Grid ...* – налаштування сітки;  
*Colors ...* – налаштування кольорової гами;  
*ShowPointerToolbar* – відображення панелі додаткових можливостей інструмента вказівник;  
*Elements* – вибір інструмента для рисування схеми, редагування елементів та для додаткових рисунків;  
*ImportSymbols ...* – вставка компонента з існуючої бібліотеки;  
*ImportRestsofComponents ...* – вставка залишків компонентів об'єднаних у однакові корпуси;  
*Componentandsymbolwizard ...* – редактор функціонування компонентів;  
*SearchandReplaceComponentShape* – пошук та заміщення форми компонентів у схемі із зовнішньої бібліотеки;  
*Exportallcomponents into a library ...* – експортування усіх елементів до бібліотеки (створення нової бібліотеки з розміщених на схемі компонентів);  
*DeleteLibraryEntry ...* – видалення елементів з бібліотеки;  
*Convert Target V3, V7, V8, V9-library ...* – перетворення бібліотек інших версій програми до сумісного формату;  
*ImportanEaglelibraryscript ...* – імпортування скриптів функціонування компонентів програми *Eagle*.

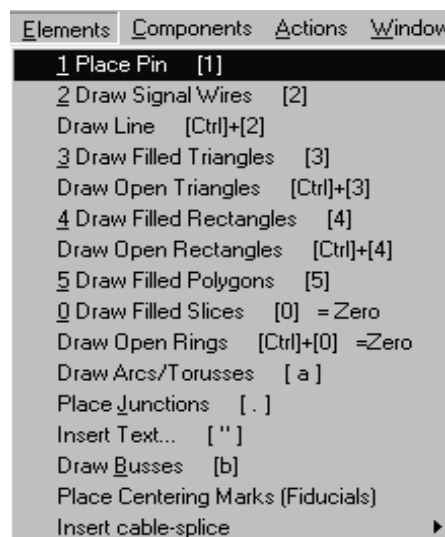


Рисунок 7.29 – Інструментарій вкладки *Elements*



*Components* – робота з компонентами:



Рисунок 7.30 – Інструментарій вкладки *Components*

*Actions*:

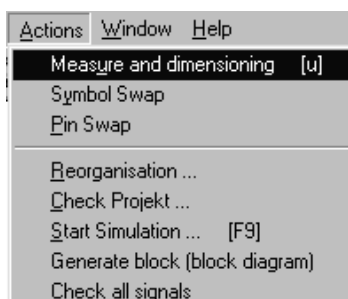


Рисунок 7.31 – Інструментарій вкладки *Actions*

*StartSimulation ...* – запуск програми моделювання схеми;

*Generateblock* – генерація схеми-блока;

*Checkallsignals* – перевірка сигналів (тимчасово не працює).

*Window* – функції для роботи з вікнами:

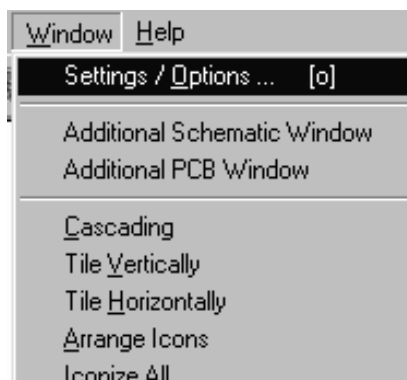


Рисунок 7.32 – Інструментарій вкладки *Window*

*AdditionalSchematicWindow* – відкрити додаткове вікно зі схемою;

*AdditionalPCBWindow* – відкрити додаткове вікно;

*Help* – допомога при роботі з програмою.



Рисунок 7.33 – Інструментарій вкладки *Help*

Тоді проект схеми матиме вигляд, поданий на рисунку 7.34.

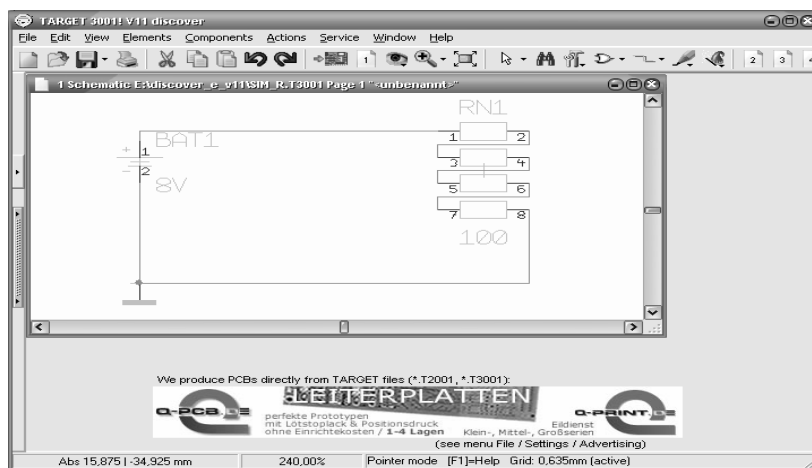


Рисунок 7.34 – Створений проект схеми

2. Перевірити проект на помилки та за їх наявності – виправити (рис. 7.35, 7.36).

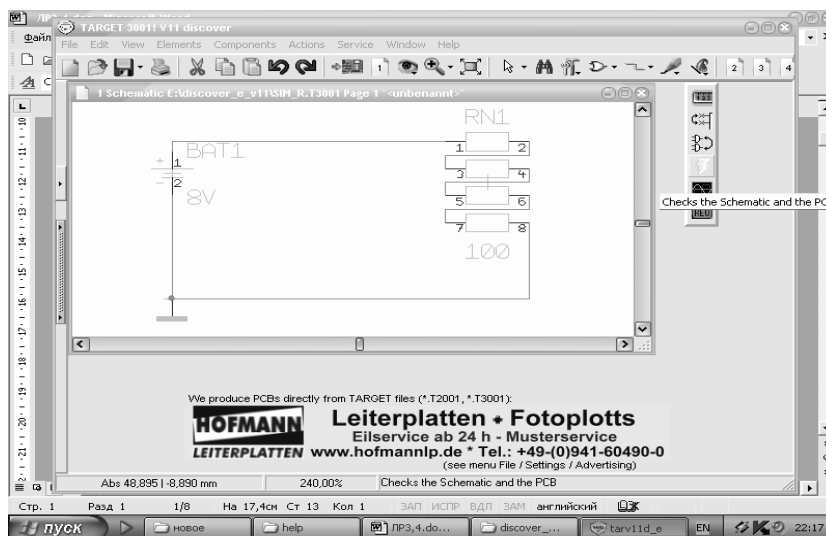


Рисунок 7.35 – Перевірка проекту на помилки

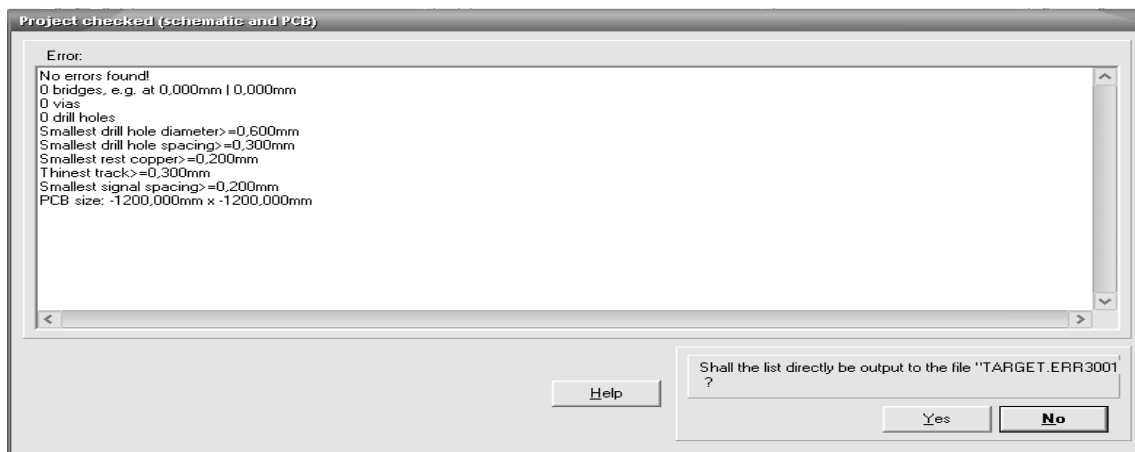




Рисунок 7.36 – Виправлення помилок

3. Запустити моделювання за допомогою команди *Actions/StartSimulation...* або натиснути в меню  кнопку .
4. Обрати тип аналізу *transient* на вкладці *Simulation*.
5. На вкладці *View* натиснути кнопку *Voltageagainstzero-potential* та вказати сигнали, які необхідно продивитися під час моделювання.
6. На вкладці *TRAN* обрати параметри діапазону та кроку дослідження сигналів.
7. Натиснути кнопку *Tothesimulator*.
8. Після запуску програми *Oscilloscope* з'явиться вікно, в якому будуть відображатися результати моделювання, натискаємо клавішу *Start* та за відсутності помилок у вікні з'являться результати моделювання. Якщо моделювання виконується повторно, то результати попереднього моделювання зберігаються. Для їх видалення необхідно просто натиснути клавішу *Start*.

**Якщо ви будете використовувати САПР PCAD, то порядок виконання роботи передбачає такі дії:**

1. Ознайомитися з програмами ППП PULSONIX, що можуть бути використаними для побудови принципівих схем пристроїв.
2. Оволодіти методикою і способами створення принципівих схем в PCAD.
3. Враховуючи блочно-ієрархічний метод проектування складних об'єктів та систем, синтезувати задану структурну схему.
4. Синтезувати функціональну схему відповідно до індивідуального завдання.
5. Обрати відповідно до функціональної схеми та з урахуванням можливостей бібліотек PCAD компоненти його принципової схеми.
6. Побудувати принципову схему на основі стандартних обраних компонентів в рамках програми.
7. Використовуючи засоби PCAD, забезпечити генерацію переліку з'єднань створеної принципової схеми.

8. Оформити звіт та зробити висновки по виконаній роботі.

### Теоретичні відомості

#### **PCAD**

Складання функціональних моделей цифрових мікросхем у прикладній програмі PCAD 4.5 складається з таких етапів:

1. Створення символічного зображення принципової схеми за допомогою графічного редактора PC-CAPS, файл з розширенням \*.SCH.

2. За допомогою програми PC-NODES вилучається список електричних зв'язків і створюється файл \*.NLT або, якщо схема велика, \*.XNL за допомогою PC-LINK.

3. Викликається програма PPESIM для створення опису схеми, що моделюється, в форматі \*.NET програми PC-LOGS.

4. Перетворення текстового файлу з функціональним описом мікросхеми у двійковий файл, який має розширення \*.MDL, за допомогою програми PC-MODEL, та перетворення у PRESIM для подання у необхідному форматі у програму PC-LOGS.

5. Безпосереднє одержання результатів моделювання в програмі PC-LOGS.

6. Перегляд результатів у вигляді часових діаграм та таблиць істинності – POSTSIM.

Приведемо список команд програми PC-LOGS, які вводяться у відповідь на підказку:

<i>Команда</i>		<i>Формат</i>	<i>Призначення</i>
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>
HELP	HE	HELP HELP <команда> ? ? <команда>	Виклик списку команд і їх опис
LOAD	LO	LOAD <ім'я файлу> >.NET LOAD n LOAD 0 {PR} {MO}	Завантаження файлу опису ланцюга або файлу стану моделювання (n=1 або 2 – завантаження файлів AUTOSA-V1.NET, AUTOSAV2.NET, створених програмою PC-LOGS; PR, MO – збереження поточного списку вузлів, створених командами PRO-BE, MONITOR відповідно, за наявності команди SAVE 0; розширення файлу .NET вказується

1	2	3	4
ATTR	AN	ATTRN PCL=(r, f, "ls", "hs") <список вузлів> ATTRN PCL <список вузлів>	Зміна PCL-атрибута одного або декількох компонентів, виходячи з'єднані з вказаними вузлами; якщо при цьому частину параметрів необхідно залишити незмінними, на відповідній позиції вказується знак *
CYCLE	CY	CYCLE <число кроків>	Визначення тривалості циклу в кроках
GEN	GE	GEN [r f {ім'я керуючого вузла }] < ім'я вузла ><тимчасова діаграма>	Задання тимчасової діаграми вхідного сигналу в циклах (r, f – затримки переднього і заднього фронту сигналу в кроках: сигнал має рівень "0", якщо керуючий вузол знаходиться в стані "0")
CLOCK	CL	CLOCK {ім'я керуючого вузла}< ім'я вузла ><тимчасова діаграма>	Задання тимчасової діаграми вхідного сигналу в кроках
ISPLAY	DI	DISPLAY n/p DISPLAY ON DISPLAY OFF	Виведення графіків тимчасових діаграм на екран дисплея (виводиться стан вузлів через n кроків p=1, 2, 4, 8 – параметр масштабування, який дозволяє зжати тимчасову діаграму в часі; за замовчуванням p=0)
ERASE	ER	ERASE	Стирання тимчасової діаграми на екрані
CONFIG	CF	CONFIG <номер порту> або S/NS H/NH V/NV	Вказівка номера порту, до якого підключений прилад захисту, або перевизначення конфігурації тимчасової діаграми при заповненні екрана (S – переміщення; V – запам'ятовування, H – зупинка, суфікс N означає заперечення, виділене шрифтом, стан встановлюється за замовчуванням)
TYPE	TY	TYPE <число> TYPE OC TYPE ON TYPE OFF	Виведення на екран таблиці списку стану вузлів (<число> – послідовність виведення в кроках; OC – видача інформації про виведення, що змінили свій стан

1	2	3	4
FORMAT	FM	FORMAT B або NB S або NS	Визначення формату виведення у вигляді таблиць (використовується разом з командою TYPE; B – вставлення пропусків між колонками, S – вставлення значень вихідних опорів разом з логічними рівнями)
SYMBOL	SY	SYMBOL <стан 1>= <символ 1>=<символ 2>...	Визначення вихідних символів при використанні команди TYPE (за замовчуванням стан SO позначається символом SO тощо)
CMODE	CM	CMODE	Переведення екрана з графічного режиму в текстовий і назад після виконання моделювання при виконанні команди DISPLAY ON
SPOOL	SL	SPOOL ON SPOOL OFF SPOOL ALL	Запам'ятовування вихідної інформації в файлі з розширенням .SPL (в режимі ON запам'ятовуються стани вузлів, вказаних в команді MONITOR, в режимі ALL – всіх вузлів)
PROBE	PR	PROBE <вузол 1>... PROBE – ALL PROBE – <вузол 1>... PROBE – #<номер>	Вибір вузлів, стан яких зображується на екрані за командою DISPLAY (знак “мінус” означає видалення з списку, #<номер> – номер вузла в списку вузлів)
MONITOR	MO	MONITOR <група вузлів 1>... MONITOR – <вузол 1>... MONITOR – #<номер> MONITOR – ALL MONITOR – <вказівник> (MSB – вузол)	Вибір вузлів, стан яких зображується на екрані і запам'ятовується на диску при використанні команд TYPE або SPOOL, не більше 250 (<група вузлів> має вигляд: <вузол 1><вузол 2>... або HEX (<вузол 1>...<вузол 4>) або OCT (<вузол 1>...<вузол 3>); параметр <вказівник> – HEX або OCT, <MSB-вузол> – ім'я першого вузла в списку HEX або OCT
SHOW	SH	SHOW <ім'я 1>...	Визначення стану одного або декількох вказаних вузлів
SPIKE	SK	SPIKE ON SPIKE OFF SPIKE <вузол 1>... SPIKE – <вузол 1>... SPIKE ALL	Аналіз піка напруги у вузлах, вказаних в списку, але не більше 16 (за замовчуванням контролюються всі вузли схеми, їх може бути і більше 16; вузли, помічені знаком “мінус”, видаляються з списку вузлів, що аналізуються)
STABLE	SB	STABLE ON STABLE OFF STABLE <ім'я 1> STABLE ALL STABLE – <вузол 1>...	Аналіз стабільності вузлів (в режимі STABLE OFF неможливо виконувати команду BREAK для контролю нестабільності)

1	2	3	4
LIST	LI	LIST SPIKE {<номер циклу: номер кроку >} LIST STABLE {< номер циклу: номер кроку >}	Виведення списку виявлених піків напруг і нестабільних вузлів (в фігурних дужках вказується початковий момент часу, за замовчуванням він збігається з початком моделювання)
INIT	IN	INIT (<вузол 1>...) <стан>...	Задання початкових значень логічним станом вказаних вузлів
NOINIT	NI	NOINIT ON NOINIT OFF	Розв'язання (параметр ON) або заборона (OFF) задання початкових станів вузлів
FORCE	FC	FORCE (<вхід><вихід>) <стан>	Штучне установлення стану вхідного вузла компонента (<вхід> і <вихід> – імена вхідного і вихідного вузла типових компонентів)
MEMLOAD	ME	MEMLOAD <ім'я><вказівник><дані>	Установлення початкового стану даних ПЗП і ОЗП (параметр <ім'я> – ім'я компонента, <вказівник
			> приймає значення HEX, OCT і BIN; <дані> мають формат: /<адреса><список даних> ? <визначник кінця даних>; адреса задається в шістнадцятковому вигляді, за замовчуванням призначення початкового адреса дорівнює 0)
STEP	SE	STEP <число кроків>	Установлення довжини моделювання в кроках
SIM	SI	SIM <число циклів> SIM TO <номер циклу>	Установлення довжини моделювання в циклах
BREAK	BR	BREAK ON SPIKE BREAK ON STABLE <число> BREAK – SPIKE BREAK – STABLE	Зупин моделювання при наявності піка напруги чи нестабільності вузла, відміна зупину (знак "-")
PAUSE	PA	PAUSE <довжина>	Переривання моделювання на заданий час (в секундах); моделювання продовжується після натискання будь-якої клавіші
AUTOSAVE	AU	AUTOSAVE <число> AUTOSAVE ON AUTOSAVE OFF	Автоматичне запам'ятовування результатів моделювання в файлах AUTOSAV1.NET і AUTOSAV2.NET
SAVE	SA	SAVE <ім'я файлу> SAVE 0	Запам'ятовування результатів моделювання в файлі поточного сеансу

1	2	3	4
STATUS	SS	STATUS	Виведення значень параметрів моделювання
EXIT	EX	EXIT	Завершення роботи з програмою моделювання PC-LOGS

### Контрольні питання

1. Які програми обраної САПР можуть бути використаними для побудови принципів схем пристроїв?
2. Назвіть способи створення принципів схем в обраній САПР.
3. В чому сутність блочно-ієрархічного підходу до проектування складних об'єктів та систем?
4. Основні етапи синтезу функціональних схем пристроїв в обраній САПР.
5. Які стандартні компоненти в рамках програми обраної САПР можна використати для побудови схем пристроїв?
6. Як згенерувати перелік з'єднань принципової схеми пристрою?

### Рекомендована література

1. Норенков И. П. Основы теории и проектирования САПР : учебник для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1990. – 335 с.
2. Норенков И. П. Системы автоматизированного проектирования электронной и вычислительной аппаратуры : учебное пособие для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1983. – 272 с.
3. Питерсон Дж. Теория сетей Петри и моделирование систем / Питерсон Дж. ; пер. с англ. – М. : Мир, 1984. – 264 с.
4. Рафикузаман М. Микропроцессоры и машинное проектирование микропроцессорных систем : в 2-х кн. / Рафикузаман М. ; пер. с англ. – М. : Мир, 1988.
5. Системы автоматизированного проектирования : учебное пособие для вузов : в 9 кн. / [ под ред. И. П. Норенкова]. – М. : Высшая школа, 1986. – 159 с.
6. Системы автоматизированного проектирования в радиоэлектронике : [справочник] / Е. В. Авдеев, А. Т. Еремин, И. П. Норенков, М. И. Песков ; под ред. И. П. Норенкова. – М. : Радио и связь, 1986. – 368 с.
7. Соловьев В. В. Проектирование функциональных узлов цифровых систем на программируемых логических устройствах : учебное пособие для вузов / Соловьев В. В. – М. : ПКООО Бестпринт, 1996. – 252 с.
8. Роїк О. М. Основи автоматизованого проектування складних об'єктів і систем : навчальний посібник / Роїк О. М., Савчук Т. О., Ткаченко О. М. – Вінниця : ВДТУ, 2001. – 111 с.



## ЛАБОРАТОРНА РОБОТА № 8

### ВИВЧЕННЯ МЕТОДІВ ПАКУВАННЯ СХЕМИ ЗА ДОПОМОГОЮ СИСТЕМИ ПРОЕКТУВАННЯ ДРУКАРСЬКИХ ПЛАТ

Мета: вивчити основні методи та принципи пакування схеми на друковану плату та закріпити ці знання, виконавши індивідуальне завдання за розміщенням схеми на друкованій платі.

#### Порядок виконання роботи

1. Ознайомитися із загальними відомостями про методи пакування схеми на друковану плату за допомогою обраної системи проектування.
2. Набрати схему у редакторі принципів схем обраної САПР згідно з варіантом завдання лабораторної роботи.
3. Занести логічні елементи у відповідні їм мікросхеми.
4. Визначити коефіцієнт використання елементів для кожної мікросхеми та для всієї схеми пристрою.
5. Розробити файл списку ланцюгів схеми, що набрана.
6. Провести пакування компонентів на друковану плату.
7. Створити друковану плату та параметри розводки згідно з вашим варіантом.
8. Оформити звіт та зробити висновки по виконаній роботі.

**Якщо ви будете використовувати САПР PULSONIX 4.5, то порядок виконання лабораторної роботи передбачає такі дії:**

1. Ознайомитися із загальними відомостями про методи пакування схеми на друковану плату за допомогою САПР PULSONIX.
2. Набрати схему у редакторі Schematic згідно з варіантом завдання лабораторної роботи.
3. Занести логічні елементи у відповідні їм мікросхеми.
4. Визначити коефіцієнт використання елементів для кожної мікросхеми та для всієї схеми пристрою.
5. Сформувати файл списку ланцюгів схеми, що набрана.
6. Провести пакування компонентів на друковану плату.
7. Створити друковану плату та параметри розводки згідно з вашим варіантом.
8. Оформити звіт та зробити висновки по виконаній роботі.

#### Теоретичні відомості

##### *САПР PULSONIX 4.5*

##### *Налаштування конфігурації та вибір необхідних параметрів*

Для розміщення компонентів та ручного трасування друкованих плат використовується редактор РСВ. На рисунку 8.1 показано вікно редактора разом із завантаженою друкованою платою.

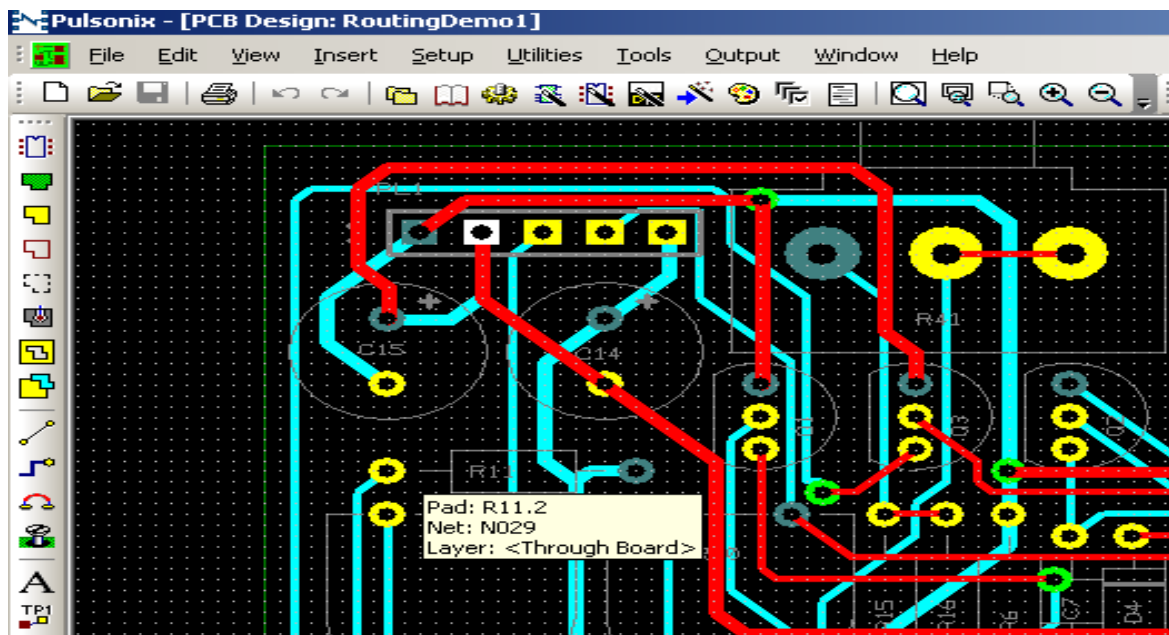


Рисунок 8.1 – Головне вікно програми РСВ

Налаштування конфігурації графічного редактора РСВ відбувається майже так як в редакторі Schematic, але існують свої особливості.

Для налаштування параметрів графічного редактора РСВ, використовується команда Tools – >Options. Діалогове вікно, яке використовується для конфігурації графічного редактора має вигляд, як показано на рисунку 8.2.

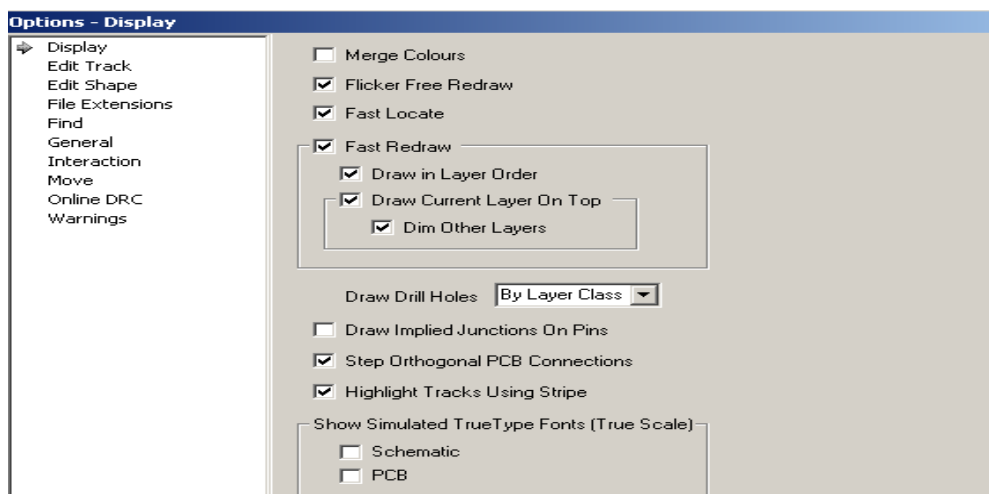


Рисунок 8.2 – Діалогове вікно для конфігурації графічного редактора

Для запуску діалогу вибору конфігурації параметрів ручної та автоматичної розводки, необхідно обрати Tools – > Auto Route – >Route Whole

Design і обрати необхідні налаштування для проекту у вікні AutoRouter (рис. 8.3).

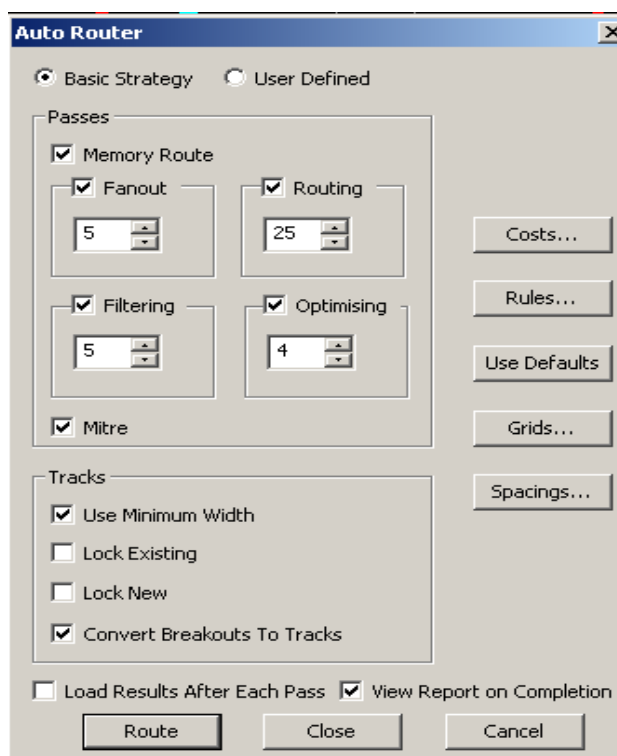


Рисунок 8.3 – Діалогове вікно Auto Rout

В цьому діалоговому вікні користувач може обрати різні види проходів (Passes):

**Memory** – це прохід, що посилає регулярні масиви шпильок, які мають колінеарні зв'язки, де шпильки розділяють загальні координати X або Y. Це ефективно для масивів пам'яті.

**Fanout** – розгалуження. Це ряд проходів, які ви можете конкретизувати.

**Routing** – задаються параметри інтерактивної розводки.

### ***Розробка топології друкованих плат***

Після налаштування конфігурації та вибору необхідних параметрів приступають безпосередньо до розробки друкованої плати. Першою операцією є рисування контуру друкованої плати. Зазвичай для цього використовують команди *Insert/Line* та *Insert/Arc*. Однак, можна скористатися вже готовим контуром плати, який встановлений за замовчуванням. В програмі РСВ є всі можливості, щоб удосконалити існуючий за замовчуванням контур на той, який потрібно (дуже потужний графічний редактор).

Наступним кроком є пакування схеми пристрою із програми Schematic на друковану плату (програма РСВ). Для цього використовується команда *Tools/PCB*. При цьому друкована плата здебільшого має такий вигляд (рис. 8.4):

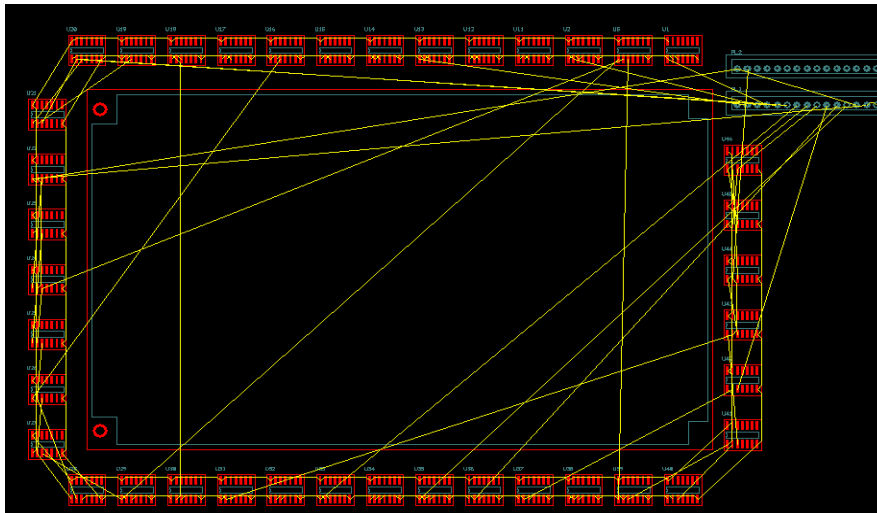


Рисунок 8.4 – Друкована плата

Зазвичай у такому вигляді плату залишати не можна, тому потрібно обробити її спеціальними утилітами, які покращать з'єднання між елементами, щоб збільшити раціональність розміщення елементів плати.

Для цього потрібно обрати із меню *Tools* – *>AutoPlaceAll*, яка використовується для організації оптимального інтерактивного розміщення компонентів на платі, оптимізацію ланцюгів “на льоту”, перестановку компонентів та організацію роботи над проектом в цілому та його окремими частинами (рис. 8.5).

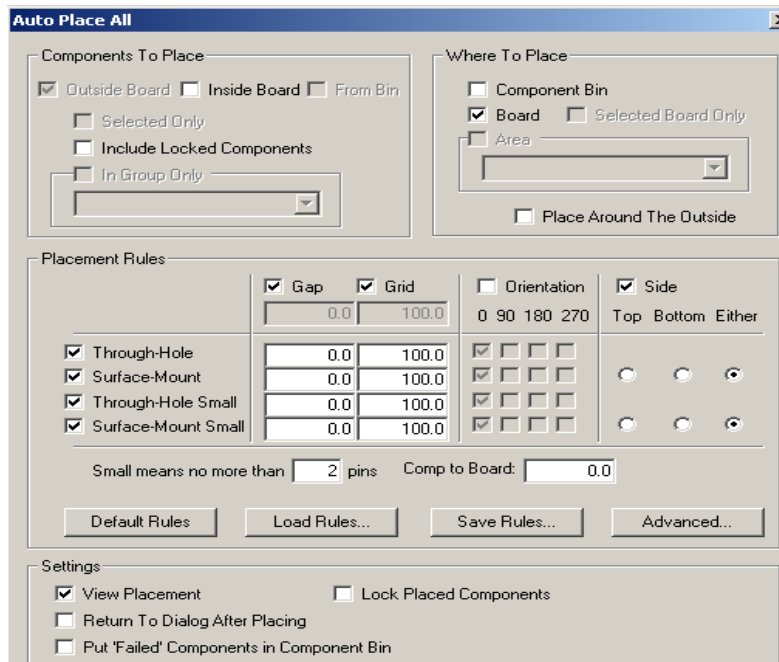


Рисунок 8.5 – Діалогове вікно Auto Place All

В діалоговому вікні *AutoPlaceAll* потрібно обрати, де саме розміщувати елементи схеми: за межами контуру (*Outsideboard*) або в межах контуру (*Insideboard*).

Після виконання цієї операції друкована плата матиме вигляд (рис. 8.6):

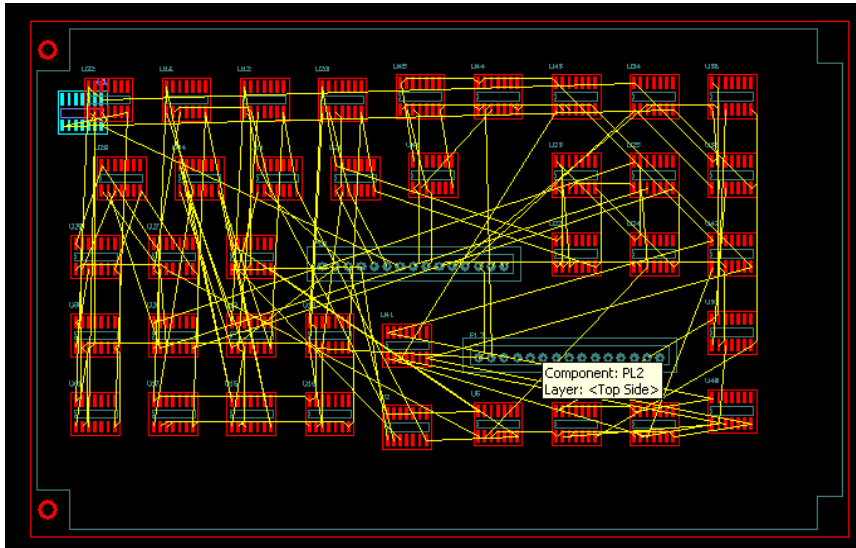


Рисунок 8.6 – Друкована плата після виконання операції AutoPlaceAll

Розроблену друковану плату можна удосконалити, обравши в меню *Tools* – *>OptimizeNets* – *>AllNets* або *SelectedNets*. Ця команда оптимізує з'єднання елементів схеми (всіх або обраних).

Також можна провести автозгладжування елементів плати за допомогою команди: *Tools* – *>AutoSmooth*.

Після виконання зазначених вище операцій плата матиме вигляд (рис. 8.7):

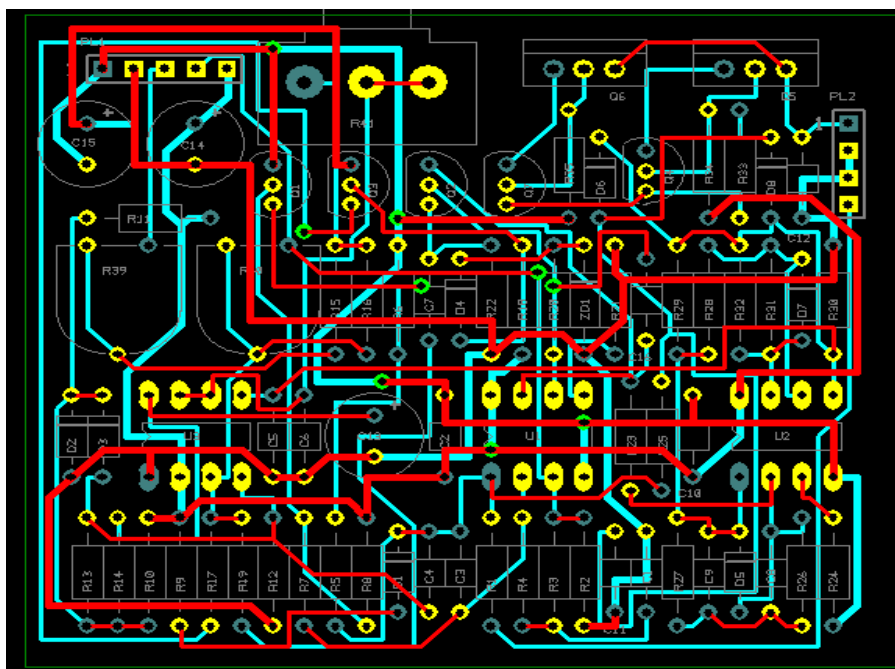


Рисунок 8.7 – Кінцевий вигляд друкованої плати

В єдиному ПППІ Pulsionix з усіх на сьогодні існуючих САПР є можливість перегляду друкованої плати в 3D-варіанті (рис. 8.8).

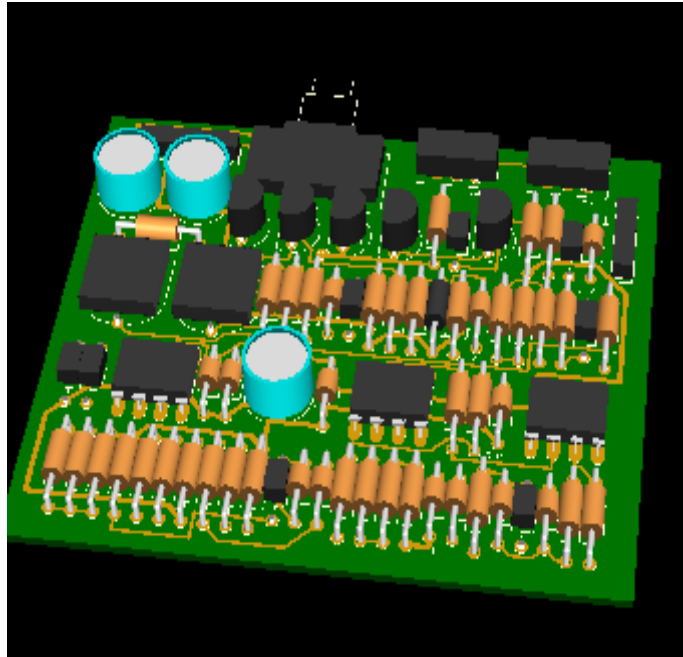


Рисунок 8.8 – Кінцевий вигляд друкованої плати у 3D-варіанті

**Якщо ви будете використовувати САПР P-CAD, то порядок виконання роботи передбачає такі дії:**

1. Ознайомитися із загальними відомостями про методи пакування схеми на друковану плату за допомогою САПР P-CAD.
2. Набрати схему у редакторі Schematic згідно з варіантом завдання лабораторної роботи.
3. Занести логічні елементи у відповідні їм мікросхеми.
4. Визначити коефіцієнт використання елементів для кожної мікросхеми та для всієї схеми пристрою.
5. Зробити файл списку ланцюгів схеми, що набрана.
6. Провести пакування компонентів на друковану плату.
7. Створити друковану плату та параметри розводки згідно з вашим варіантом.
8. Оформити звіт та зробити висновки по виконаній роботі.

### **Теоретичні відомості**

#### ***САПР P-CAD***

#### ***Налаштування конфігурації та вибір необхідних параметрів***

Для розміщення компонентів та ручного трасування друкованих плат використовується редактор PCB. На рисунку 8.9 показано вікно редактора разом із завантаженою друкованою платою.

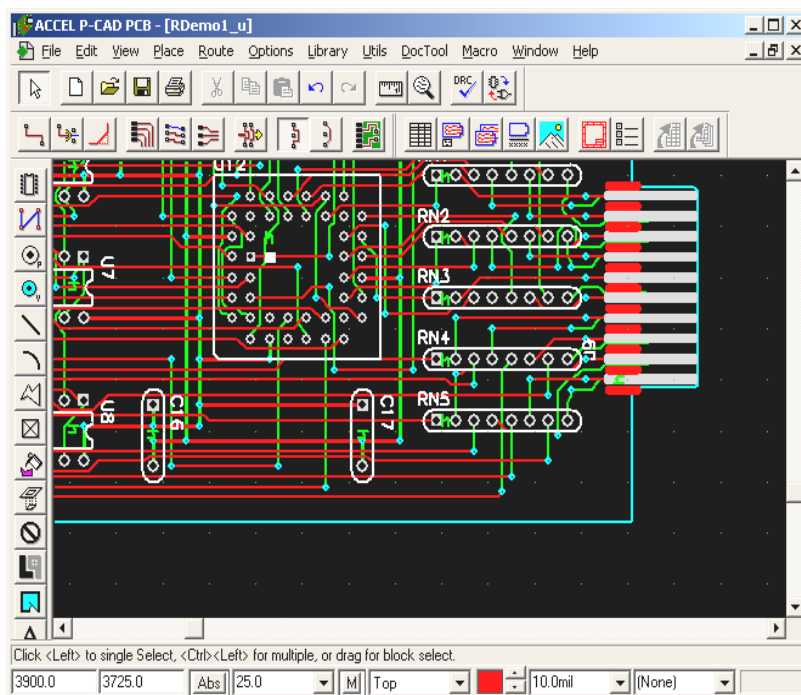


Рисунок 8.9 – Головне вікно програми РСВ

Налаштування конфігурації графічного редактора РСВ виконується так як в редакторі Schematic, але існують свої особливості.

Для налаштування параметрів графічного редактора РСВ, використовується команда Options/Configure. Вікно, яке з'являється при виконанні цієї команди має чотири режими: General, Online DRC, Route, PDM. Діалогове вікно, яке використовується для конфігурації графічного редактора має вигляд, як показано на рисунку 8.10.

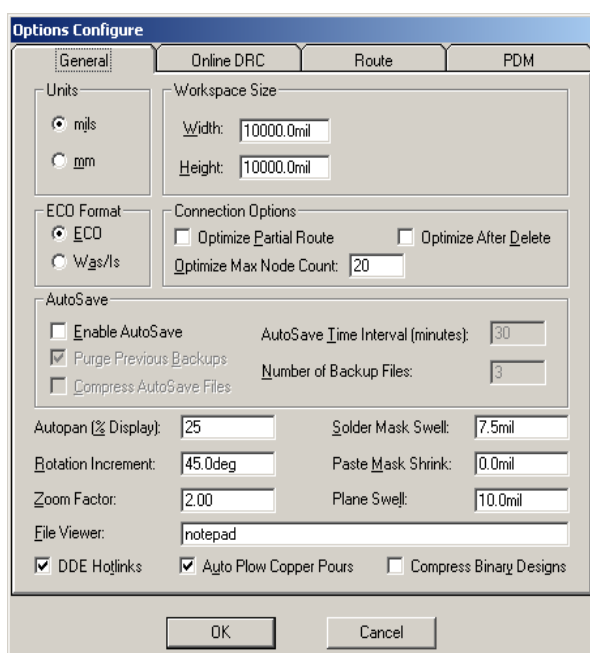


Рисунок 8.10 – Діалогове вікно для конфігурації графічного редактора (режим “General”)

Розглянемо детальніше кожен з параметрів, що може встановлюватися в цьому вікні:

- група параметрів “*Units*” дозволяє встановити систему одиниць вимірювання, які використовуються в проекті. Причому, в будь-який момент можна перейти від однієї системи до іншої (1mil = 0.001 дюйма);

- розміри робочої поверхні редактора задаються за допомогою параметрів *Width* та *Height* у групі “*WorkspaceSize*”;

- у групі параметрів “*AutoSave*” встановлюються параметри автоматичного зберігання проекту;

- у групі “*ECOFormat*” встановлюється формат файлу зміни схеми (EngineeringChangeOrders). При встановленому прапорці “*Was/Is*” утиліта *ECO* робить запис тільки зміни атрибута позиційного позначення компонента RefDes;

- група параметрів “*ConnectionOptions*” містить два прапорці. Перший параметр “*OptimizepartialRoute*” використовується для вказівки чи повинен оптимізуватися зв’язок для досягнення мінімальної “манхетенівської довжини” після ручного трасування. Якщо увімкнена ця опція, то при ручній розводці провідник закінчується з’єднанням з найближчим ланцюгом. Опція “*OptimizeAfterDelete*” використовується для того щоб вказати, чи з’явилися нові траси після видалення траси і вводу її оптимальним способом. Як правило ці прапорці не відмічають, оскільки оптимальність з точки зору “манхетенівської довжини” ще не означає найкращого розміщення траси на реальній платі;

- у графі “*SolderMaskSwell*” задається величина, на яку радіус вирізу під контактну площину в шарі захисної паяльної маски більше діаметра самої контактної площини;

- у графі “*PasteMaskShrink*” задається величина, на яку розміри самого виводу компонента менші, ніж розмір контактної площини;

- у графі “*PlaneSwell*” визначають зазор між суцільним шаром металізації та контактною площиною чи перехідним отвором, що не належить цьому шару;

- якщо встановити прапорець “*AutoPlowCopperPours*”, то дозволяється автоматичне зменшення кількості перехідних отворів при оптимізації;

- у графі “*RotationIncrement*” встановлюється приріст кута при обертанні за натискання клавіші R.

Інші параметри діалогового вікна у режимі “General” аналогічні до команд конфігурації графічного редактора програми Schematic.

Для запуску діалогу конфігурації параметрів ручної та автоматичної розводки, необхідно натиснути на ярлику “*Route*” у вікні *Options/Configure*:

- прапорець “*T-RouteDefault*” вмикає режим Т-образної розводки як основної. В цьому випадку траса ланцюга є основним провідником з відводами, які підходять до нього в будь-яких місцях;



– група параметрів “*Highlight While Routing*” задає режим підсвічування поточного ланцюга в процесі його трасування. Встановлення “*PadsOnly*” дозволяє підсвічувати тільки контактні площини, що належать цьому ланцюгу. Вибір “*Pads, Traces and Connections*” дає можливість виділити кольором як контактні площини так і провідники та лінії з’єднань на платі;

– у групі “*MiterMode*” вибирається режим згладжування зломів провідника. Якщо встановити прапорець “*45DegreeLine*”, отримаємо згладжування злomu відрізком провідника під кутом 45°. Прапорець “*90DegreeArc*” вмикає режим спряження кутів дугою окружності;

– у групі “*ManualRoute*” встановлюються параметри ручного трасування. Прапорець “*RightMousetoComplete/SlashKeytoSuspend*” вибирається у випадку, якщо для автоматичного завершення траси за найкоротшим шляхом використовується права кнопка миші, а для зупинки трасування ланцюга в будь-якому місці – клавіша “/”. У випадку, якщо вибраний прапорець “*Slash Key to Complete/Right Mouse to Suspend*”, для зупинки трасування ланцюга використовується права кнопка миші, а для її автоматичного завершення – клавіша “/”. Приклад діалогового вікна для конфігурації графічного редактора поданий на рисунку 8.11;

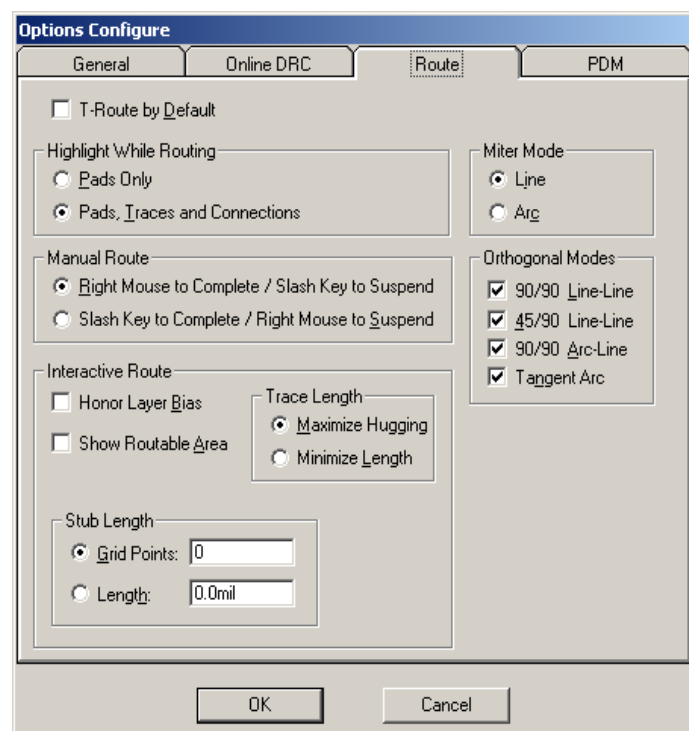


Рисунок 8.11 – Діалогове вікно для конфігурації графічного редактора (режим “Route”)

– у групі “*InteractiveRoute*” задаються параметри інтерактивної розводки. У полі “*StubLength*” вводиться мінімальна довжина в дискетах сітки сегмента лінії, яка використовується, для організації з’єднання з контакт-

ною площиною. В групі “*TraceLength*” задається контроль за довжиною провідника при прокладанні траси. Режим “*MaximizeHugging*” забезпечує максимально тісне огинання вже прокладених трас, режим “*MinimizeLength*” використовується при прокладанні траси мінімальної дожини з мінімумом перехідних отворів. Встановлення прапорця “*HonorLayerBias*” означає, що використовуються напрямки трасування шарів, які визначені у вікні команди “*OptionsLayer*”. Прапорець “*Show Routable Area*” визначає видимість області трасування.

Особливості виведення попереджень при роботі з векторними шрифтами встановлюються у вікні команди Options/Configure на вкладці “*PDMTabOptions*”;

– група параметрів “*PDMWarnings*” дозволяє вмикати чи вимикати появу попереджень при зберіганні чи друкуванні файлів, які містять шрифти TrueType. Цю опцію потрібно встановлювати, якщо файли будуть використовуватися у програмному забезпеченні, яке використовують тільки TrueType чи тільки векторні шрифти. Відмічаючи відповідний прапорець, можна ввімкнути чи вимкнути попередження при друку шрифтів TrueType як векторних (*Printing TrueType as Stroke*) чи при зберіганні шрифтів TrueType як векторних (*Saving TrueType as Stroke*).

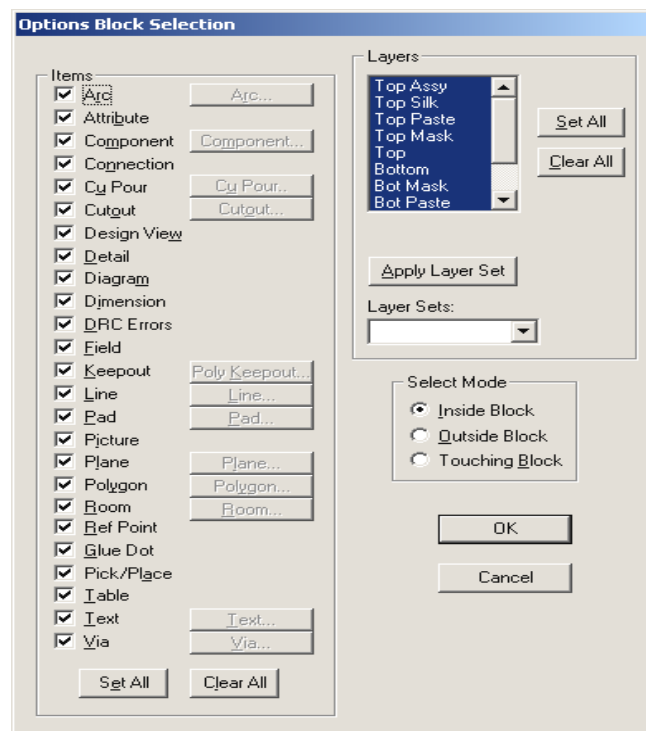


Рисунок 8.12 – Вікно конфігурації режимів вибору

P-CAD2000 має можливість конфігурації функцій вибору при роботі в графічному редакторі. Конфігурація режимів вибору виконується за допомогою команди Options/Block Selection. Вікно цієї команди наведено на рисунку 8.12.

Налаштування фільтрів вибору виконується відміткою відповідних прапорців в стовпці *Items*. Кнопки *SetAll* та *ClearAll* дозволяють встановити та скинути виділення усіх можливих об'єктів. За замовчуванням встановлюються усі фільтри.

Нижче приводиться таблиця 8.1 з описом кожного з прапорців в стовпці *Items*. Ті об'єкти, які не відмічені, не будуть виділятися при виборі блока.

Таблиця 8.1 – Опис прапорців

<b>Об'єкт</b>	<b>Описання</b>
<i>1</i>	<i>2</i>
Arc	Дуги
Attribute	Атрибути
Component	Компоненти
Connection	Зв'язки
Copper Pour	Області металізації
Cut Out	Вирізи
Detail	Деталі
Diagram	Діаграми шарів
Dimension	Розмірні лінії
<i>1</i>	<i>2</i>
DRC Errors	Кольорове виділення похибок контролю технологічних параметрів проекту (Design Rule Check)
Field	Поля
Keepout	Бар'єри
Line	Лінії
Pad	Контактні площини
Picture	Зображення (наприклад, товарний знак)
Plane	Шари металізації
Polygon	Полігони
Room	Кімнати
Ref Point	Точки прив'язки
Glue Dot	Точки приклеювання компонентів
Pic/Place	Точки розміщення
Table	Таблиці
Text	Текстова інформація
Via	Перехідні отвори

Для таких об'єктів як дуги (*Arc*), компоненти (*Components*), області металізації (*CopperPour*), лінії (*Line*), контактні площини (*Pad*), шари металізації (*Plane*), кімнати (*Room*), текстова інформація (*Text*), перехідні отвори (*Via*) існує можливість встановлення властивостей, за якими виби-

раються об'єкти при натисканні відповідної клавіші. В цьому випадку запускається діалог властивостей об'єкта.

Можна також задати фільтр вибору для тих чи інших шарів та груп шарів проекту, використовуючи поля *Layers* та *ApplyLayerSet*, відповідно.

Опції вибору режиму задаються параметрами *SelectMode*. Існує три режими вибору у блоці (табл. 8.2).

Таблиця 8.2 – Режими вибору у блоці

Режим	Описання
Inside Block	Вибір усіх об'єктів, що знаходяться в межах блока. Встановлюється за замовчуванням.
Outside Block	Вибір для усіх об'єктів, які знаходяться за межами блока.
Touching Block	Вибір усіх об'єктів, які знаходяться в межах блока та тих, що торкаються до його границі.

Для комфортної роботи існує можливість налаштування параметрів дисплея за допомогою команди *Options/Display*. Вікно цієї команди наведено на рисунку 8.13.

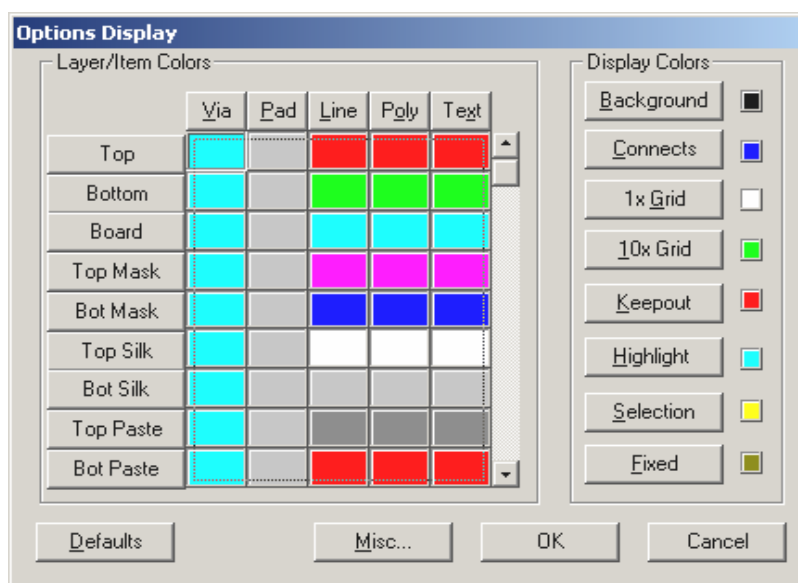


Рисунок 8.13 – Вікно команди Options/Display

В групі “*Layer/ItemColors*” встановлюються кольори тих чи інших об'єктів, що належать різним шарам друкованої плати. За допомогою кнопок *Layer/ItemColors* можна задати усім однорідним об'єктам однаковий колір для усіх шарів чи, навпаки, зафарбувати об'єкти одного шару в один колір.

Крім цього можлива зміна кольору службової інформації (фону, сітки, бар'єрів) та об'єктів, над якими було зроблено ті чи інші дії (виділення).

При натисканні клавіші “Misc” з’являється вікно додаткових параметрів дисплея (рис. 8.14).

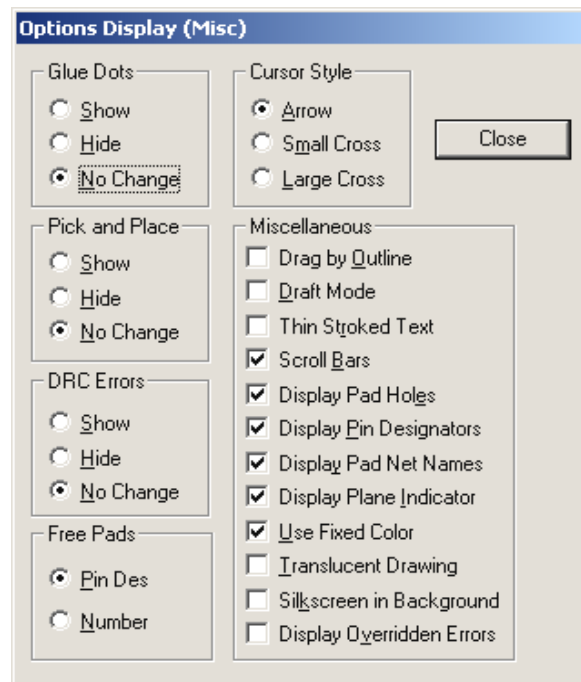


Рисунок 8.14 – Вікно налаштування додаткових параметрів відображення

У групах “GlueDots” та “PickandPlace” задають параметри виведення на дисплей прив’язок компонентів при автоматичному монтажі.

Можливі три варіанти вибору:

Show	Показувати
Hide	Сховати
No Change	Не міняти (порівняно з бібліотечним посадочним місцем)

Аналогічним чином в полі *DRCErrors* задаються параметри виведення похибок технологічних норм.

Вигляд курсору можна встановити за допомогою групи опцій *CursorStyle*. Можливі три варіанти:

Arrow	Стандартний курсор Windows у вигляді стрілки
Small cross	Курсор у вигляді маленького хрестика
Large cross	Курсор у вигляді хрестика розміром на все робоче вікно

Швидке перемикання між різними типами курсору виконується за допомогою клавіші X.

У групі параметрів *Miscellaneous* задаються різні особливості відображення на екрані тих чи інших елементів проекту. Можливі параметри подано в таблиці 8.3.

Таблиця 8.3 – Означення параметрів

Drag by Outline	Відображує тільки габарити об'єктів при їх переміщенні, тим самим прискорюючи рисування екрана
DraftMode	Відображує тільки контури товстих ліній (товщиною 1 піксель), багатокутників та тексту без заповнення
Thin Stroked Text	Відображення тексту тонкими лініями
Scroll Bars	Відображення лінійок прокручування дисплея
Display Pad Holes	Відображення отворів в контактних площинах
Display Pin Designators	Відображення номерів виводів компонента
Display Pad Net Names	Відображення номерів ланцюгів, що з'єднані з компонентом
Display Plane Indicator	Відображення перехідного отвору чи контактної площини, що підключена до довгого шару металізації у вигляді хрестика, який пофарбовано у колір цього шару
Use Fixed Color	Відображує зафіксовані компоненти кольором
Translucent Drawing	Напівпрозоре відображення об'єктів
Silkscreen in Background	Відображення графіків у шарі шовкографії на задньому плані

Одною з важливих операцій при конфігурації графічного редактора друкованих плат є установлення структури шарів друкованої плати. Для установлення їх структури є команда *Options/Layers*. Вікно команди *Options/Layers* показано на рисунку 8.15.

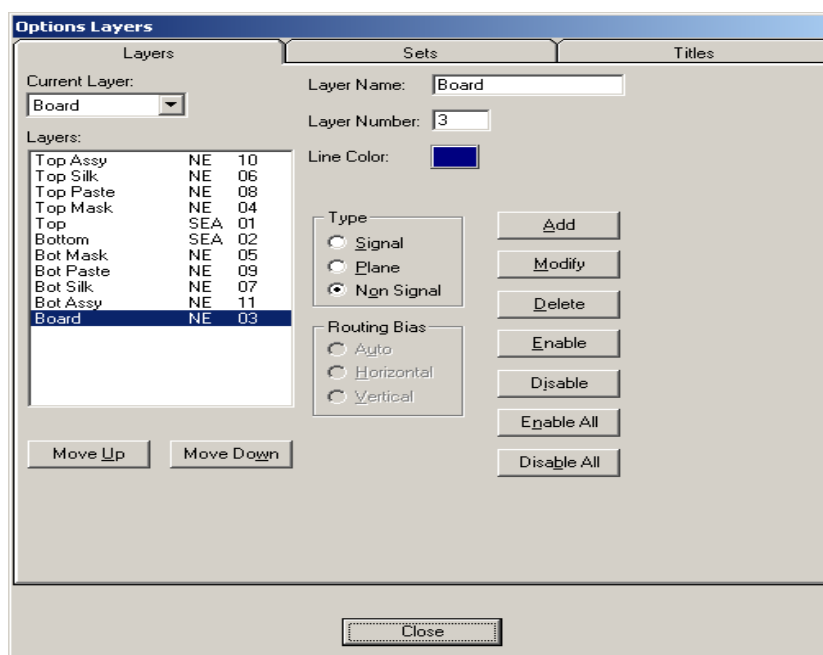


Рисунок 8.15 – Вікно команди Options/Layers

Потрібно зазначити, що за замовчуванням встановлюється структура шарів для друкованої плати з двома сигнальними шарами. Для проектування друкованих плат з декількома сигнальними шарами, а також із неперервними шарами металізації необхідно виконати завдання нових шарів. В таблиці 8.4 подана структура шарів друкованої плати, задана за замовчуванням, та їх відповідність шарам системи P-CAD. Список шарів проекту виводиться у вікні *Layers*.

Таблиця 8.4 – Структура шарів друкованої плати, задана за замовчуванням, та їх відповідність шарам системи P-CAD

Top	Провідники з верхньої сторони друкованої плати (сторони встановлення компонентів, аналогічні до шарів COMP, PADCOM, PINTOP у системі P-CAD)
Top Assy	Атрибути з верхньої сторони друкованої плати (шари DVSTOP, DEVICE у системі P-CAD)
Top Silk	Шовкографія з верхньої сторони друкованої плати (шари SLKTOP, SLKSCR, REFDES, REFDTF у системі P-CAD)
Top Paste	Графіка пайки з верхньої сторони друкованої плати (шар PSTGPT у системі P-CAD)
Top Mask	Графіка маски пайки з верхньої сторони друкованої плати (шар MSKGTP у системі P-CAD)
Bottom	Провідники з нижньої сторони друкованої плати (сторони монтажу, аналогічні шару SOLDER, PADSLD, PINBOT у системі P-CAD)
Bot Mask	Графіка маски пайки з нижньої сторони друкованої плати (шари SLDMSK, MSKGBT у системі P-CAD)
Bot Paste	Графіка пайки з нижньої сторони друкованої плати (шар PSTGBT у системі P-CAD)
Bot Silk	Шовкографія з нижньої сторони друкованої плати (шари SLKBOT, REFDBT у системі P-CAD)
Bot Assy	Атрибути з нижньої сторони друкованої плати (шар DVCBOT в P-CAD)
Board	Границі друкованої плати (шар аналогічний шару BRDOUT в P-CAD)

У полі *CurrentLayer* відображається поточний шар. Тип шару відображається у списку *Layers* наявністю відповідної букви в середній колонці таблиці.

Signal	S	Шар розводки сигнальних провідників
Plane	P	Загальний шар металізації для підключення ланцюгів живлення
Nonsignal	N	Шар, що не містить інформацію про електричні ланцюги (тільки графічні шари)

Кожен шар може бути вимкнений натисканням кнопки *Enable* чи увімкнений за допомогою кнопки *Disable*. Відповідно, у середній колонці в списку *Layers* відображається буква *E* або *D*.

У полі *RoutingBias* вказують основну орієнтацію друкованих провідників у вибраному сигнальному шарі. В середній колонці у списку *Layers* відображається буква, яка вказує пріоритетну орієнтацію провідників.

Auto	A	Автоматична орієнтація провідників
Horizontal	H	Горизонтальна орієнтація провідників
Vertical	V	Вертикальна орієнтація провідників

Основні технологічні параметри проекту зручно визначати за допомогою команди *Options/DesignRules*. Вікно цієї команди має шість закладок, для визначення таких параметрів проекту як глобальні зазори проекту (*DesignTab*), параметри зазорів в сигнальних шарах та шарах металізації (*LayerTab*), параметри кімнат (*RoomsTab*), правила трасування класів ланцюгів (*NetClassTab*), окремих ланцюгів (*NetTab*), а також правила трасування між ланцюгами різних класів (*ClassToClassTab*).

Зкладка для задання глобальних зазорів проекту (*DesignTab*) команди *Options/DesignRules* зображена на рисунку 8.16.

У полі *Name* вказується ім'я глобального параметра (атрибута), у полі *Value* – його величина.

Додавання нового параметра до списку виконується за допомогою кнопки *Add*, а редагування величини існуючого – при натисканні кнопки *Properties*, видалення параметра зі списку виконується за допомогою кнопки *Delete*.

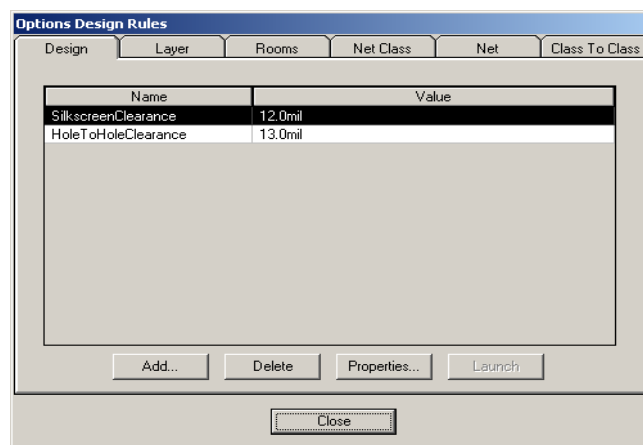


Рисунок 8.16 – Зкладка для задання глобальних зазорів проекту (*DesignTab*) команди *Options/DesignRules*

На рисунку 8.17 зображена закладка для задання зазорів в сигнальних шарах та шарах металізації (*LayerTab*) команди *Options/DesignRules*.

У формі закладки відображується значення зазорів для кожного шару (*Clearance*) між контактними площинами (*PadtoPad*), контактною площи-



ною та доріжкою провідника (*PadtoVia*), між доріжкою та перехідним отвором (*LinetoVia*), між перехідними отворами (*ViatoVia*). Задання зазорів виконується у відповідних полях для кожного шару. При натисканні кнопки *SetDefaults* усі зазори встановлюють рівними 12 mil (0,3mm). Для внесення змін до конфігурації необхідно натиснути кнопку *Update*.

Інші закладки аналогічні закладкам графічного редактора ACCELSchematic. Їх було описано в попередніх лабораторних роботах.

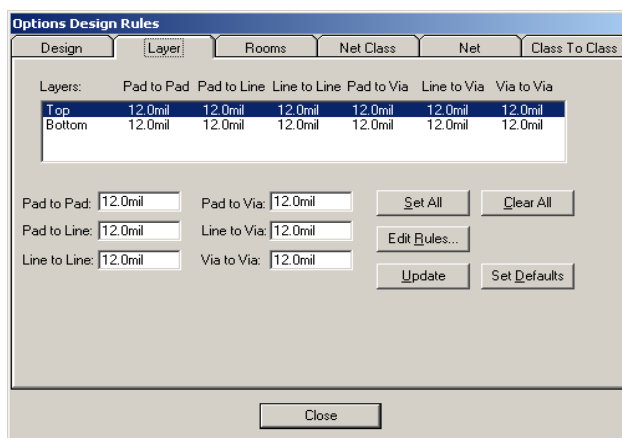


Рисунок 8.17 – Закладка для задання зазорів в сигнальних шарах та шарах металізації (*LayerTab*) команди *Options/DesignRules*

### ***Розробка топології друкованих плат***

Після налаштування конфігурації та вибору необхідних параметрів приступають безпосередньо до розробки друкованої плати. Першою операцією є створення контуру друкованої плати в шарі *Board*. Зазвичай для цього використовують команди *Place/Line* та *Place/Arc*. В деяких випадках, коли контур плати має складну форму, краще виконати рисування границь плати в програмі САПР типу *AutoCAD*, а потім за допомогою команди імпорту *File/DXFIn* загрузити отриманий контур в графічний редактор *PCB*.

Наступним кроком є пакування списку з'єднань на друковану плату. Для цього використовується команда *Utils/LoadNetlist*, вікно якої зображено на рисунку 8.18.

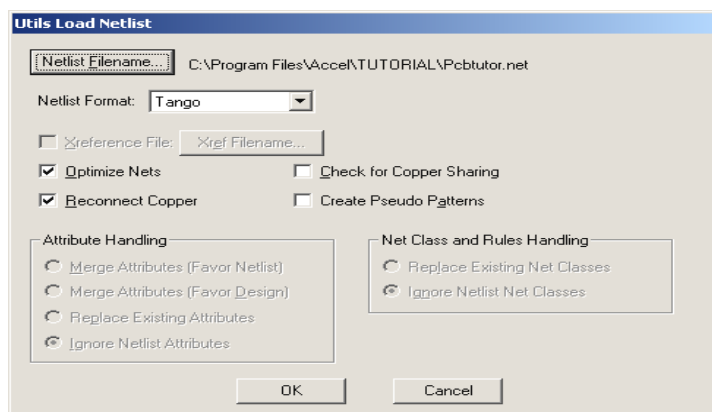


Рисунок 8.18 – Вікно команди *Utils/Load Netlist*

Формат файлу, що повинен бути зчитаний, вибирається в графі NetlistFormat. Система дає можливість вибрати серед трьох форматів:

- *ACCELASCII* – текстовий формат, який містить інформацію про атрибути компонентів та ланцюгів (розширення файлу .NET);
- *P-CADALT* – текстовий формат P-CAD (розширення файлу .ALT);
- *Tango* – формат TangoPro (розширення файлу .NET).

Якщо завантажується список ланцюгів у форматі P-CADALT, то за допомогою кнопки *XrefFilename* вибирається файл перехресних посилань (*xrf* – файл). Якщо компонент не має створеного посадочного місця (*Pattern*), відмічається прапорець *CreatePseudoPattern*, в цьому випадку створюється так зване уявне посадочне місце (*PseudoPattern*) і видається відповідне попередження.

Прапорцем *OptimizeNets* вмикається режим мінімізації довжин з'єднань на платі шляхом перестановки логічно еквівалентних вентилів та виводів. Розумніше при пакуванні схеми не займатися оптимізацією, залишивши цей прапорець невідміченим.

Відмічений прапорець *ReconnectCopper* дозволяє підключення до вже існуючих місць металізації ланцюгів схеми, що пакується. Як правило, передчасну розводку ланцюгів до пакування усієї схеми проводять не часто, тому цей прапорець треба залишити невідміченим.

Опція *CheckforCopperSharing* дозволяє ввімкнути перевірку плати з передчасно розташованими компонентами і передчасною розводкою частини ланцюгів на наявність помилок.

Якщо список ланцюгів виконаний у форматі ACCELASCII, то можна налаштувати параметри передавання атрибутів *AttributeHandling* і правила класів ланцюгів *NetClassandRulesHandling*.

Можливі параметри налаштування передавання атрибутів подані в таблиці 8.5.

Таблиця 8.5 – Параметри налаштування передавання атрибутів

Merge Attribute (Favor Netlist)	Злиття атрибутів списку ланцюгів з атрибутами проекту, пріоритет за атрибутами списку ланцюгів
Merge Attribute (Favor Design)	Злиття атрибутів списку ланцюгів з атрибутами проекту, пріоритет за атрибутами проекту
Replace Existing Attribute	Заміна поточних атрибутів проекту на атрибути списку ланцюгів
Ignore Netlist Attribute	Ігнорування атрибутів списку ланцюгів

Можливе налаштування наступних правил та класів ланцюгів:

Replace Existing Net Classes	Замінити існуючі класи ланцюгів проекту класами ланцюгів зі списку ланцюгів
Ignore Netlist Net Classes	Ігнорувати визначення класів ланцюгів зі списку ланцюгів

Як правило, якщо передчасне розміщення компонентів та трасування ланцюгів не було проведено, то ці опції несуттєві.

Безпосереднє завантаження виконується при натисканні кнопки ОК. При цьому в робочій області розміщуються компоненти і відображуються зв'язки між ними (рис. 8.19).

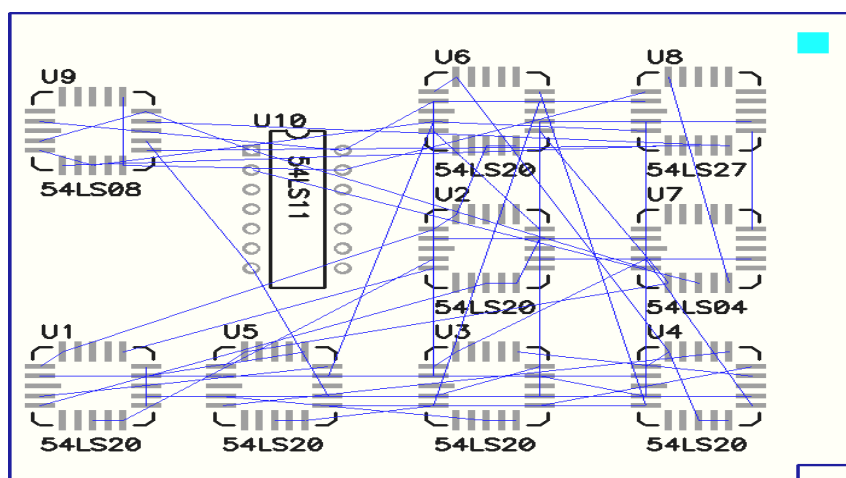


Рисунок 8.19 – Схема, що запакована на друковану плату

У випадку, коли розмір робочого простору при завантаженні плати недостатній для розміщення компонентів, з'явиться повідомлення про помилку. В цьому випадку необхідно змінити розмір робочого простору (*Workspacesize*) за допомогою команди *Options/Configure*.

Після пакування схеми настає відповідальний етап розміщення компонентів на платі. Автоматизація цього процесу тільки шкодить справі. Версія ACCELEDA 15.0 не має у переліку програм програми для автоматизованого розміщення компонентів на платі. Існує програма, що входить до складу програми SPECSTRA, для автоматичного розміщення компонентів на друкованій платі.

### **Утиліти інтерактивного розміщення та управління проектом**

В пакет ACCEL EDA 15.0 входить програма ACCEL InterPlace/ACCEL Parametric Constraint Solver (PCS), яка використовується для організації оптимального інтерактивного розміщення компонентів на платі, оптимізації ланцюгів “на льоту”, перестановки компонентів та організації роботи над проектом в цілому та його окремими частинами.

При розміщенні компонентів з використанням ACCEL InterPlace/Parametric Constraint Solver (PCS) використовуються задані при

налаштуванні схеми правила розміщення об'єктів (компонент, ланцюгів). При цьому відображається доступна для розміщення компонента область плати.

При об'єднанні компонентів в кімнати використовується широкий клас атрибутів компонентів, що дозволяє сформувати групу відповідності до різних вимог до розміщення компонентів.

ACCEL InterPlace/Parametric Constraint Solver (PCS) забезпечує просту та зручну динамічну індикацію електричних ланцюгів при розміщенні компонентів (dynamic ratsnest), при цьому видна вся необхідна інформація про ланцюг.

Дуже хорошою можливістю цієї програми є те, що вона має можливість виконання інтерактивного розміщення та трасування з урахуванням різних параметричних обмежень, можуть бути задані як у вигляді атрибутів так, і у вигляді виразів.

Використання цього механізму дозволяє виконувати трасування ланцюгів та розміщення компонентів з урахуванням рівня наведень, імпедансів провідників, а також механічних та теплових властивостей плати.

Утиліта ACCEL InterPlace має стандартний DBX інтерфейс і викликається з графічних редакторів ACCEL Schematic чи ACCEL PCB.

Але пізніше як додаток вийшла програма ACCEL Parametric Constraint Solver, яка розроблена для ітеративного розміщення компонентів на друкованій платі.

Далі приводять деяку корисну інформацію для ручного розміщення компонентів в системі ACCELEDA.

При розміщенні компонентів, розвертання їх на 90° за годинниковою стрілкою виконується натисканням клавіші *R*, а перенесення на протилежний бік друкованої плати – натисканням клавіші *F*.

“Павутиння” ліній, якщо зображають зв'язки, переміщуються разом з компонентом, що переміщується. Можна ввімкнути підсвічування ліній зв'язків, дозволити або заборонити їх відображення тощо. Для цих цілей слугує вікно *Edit/Nets*, що приведено на рисунку 8.20.

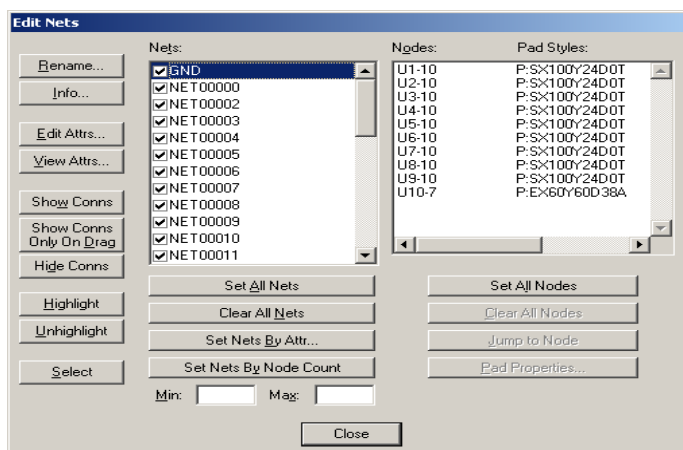


Рисунок 8.20 – Вікно команди Edit/Nets

У полі *Nets* відображується список усіх ланцюгів проекту. У полі *Nodes* виводиться список усіх вузлів, що належать вибраному ланцюгу. За допомогою кнопки *Rename* можна змінити назву ланцюга.

Кнопка *SetAllNets* дозволяє вибрати усі ланцюги у списку. За допомогою кнопки *SetNetsByAttr* можна виділити усі ланцюги, що мають задане значення атрибута (наприклад, ширина провідника *WIDTH*). Виділення за атрибутом виконується у діалоговому вікні (рис. 8.21).

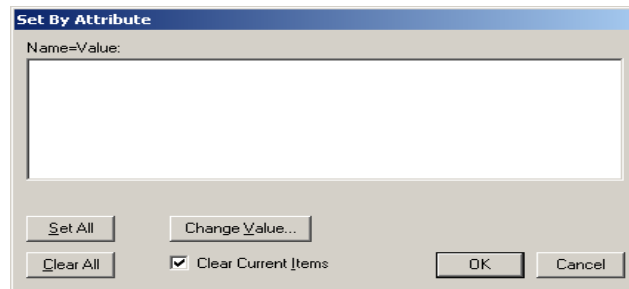


Рисунок 8.21 – Діалогове віно для вибору ланцюгів за атрибутами

В цьому вікні задаються назва (*Name*) та значення атрибута (*Value*).

В полі *SelectNetsByNodeCount* можна вибрати ланцюги, які мають задану кількість вузлів. Кількість вузлів задається мінімальним (*Min*) і максимальним (*Max*) значеннями.

Кнопка *EditAttrs* дозволяє перейти до діалогу редагування атрибутів ланцюгів. Натисканням кнопки *ViewAttrs* відкривається стандартний редактор Windows notepad для проглядання атрибутів ланцюгів.

Вигляд запакованої схеми приведений на рисунку 8.22.

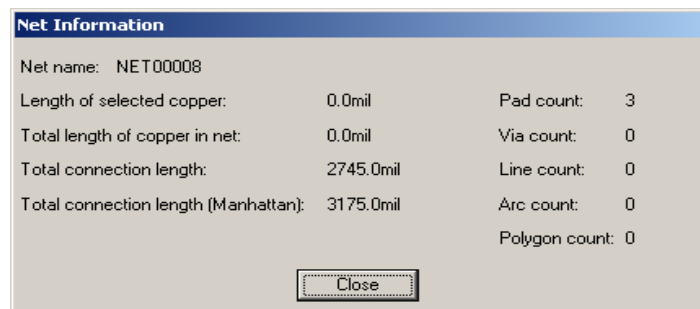


Рисунок 8.22 – Інформація про ланцюг

Корисною можливістю ACCEL EDA є механізм автоматичного вирівнювання компонентів. Для цього вибираються декілька компонентів (використовуючи усю міць удосконаленої підсистеми вибору), які будуть вирівняні відносно один одного. Зручно виконувати вибір компонентів клацанням лівої клавiшi мишi при натисканнi клавiшi Ctrl. Після того, як компоненти вибрані, потрібно натиснути праву клавiшу мишi і у контекстному меню, яке з'явиться, вибрати точку відліку (*Selection Point*) і встановити її у вибраному місці робочого простору. Після цього знову викликається контекстне меню і вибирається опція вирівнювання (*Align*). Можна

також вибрати в основному меню команду Edit/Align Components, але це менш зручно.

В цьому випадку з'являється діалогове вікно, яке зображено на рисунку 8.23.

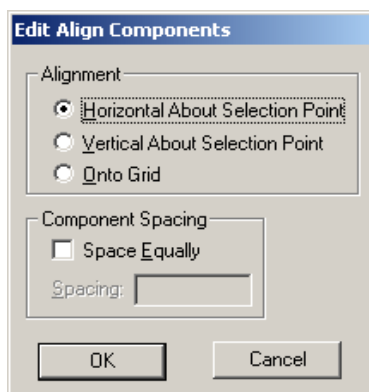


Рисунок 8.23 – Вікно команди вирівнювання компонентів Edit/Align Components

В цьому вікні можна вибрати три варіанти вирівнювання компонентів:

- |                                  |  |
|----------------------------------|--|
| Horizontal About Selection Point | Вирівнювання по горизонталі відносно точки відліку |
| Vertical About Selection Point   | Вирівнювання по вертикалі відносно точки відліку   |
| Onto Grid                        | Вирівнювання у вузлові точки сітки                 |

Припустимо, ми нарисували схему деякого пристрою за допомогою графічного редактора ACCELSchematic. Ця схема подана на рисунку 8.24.

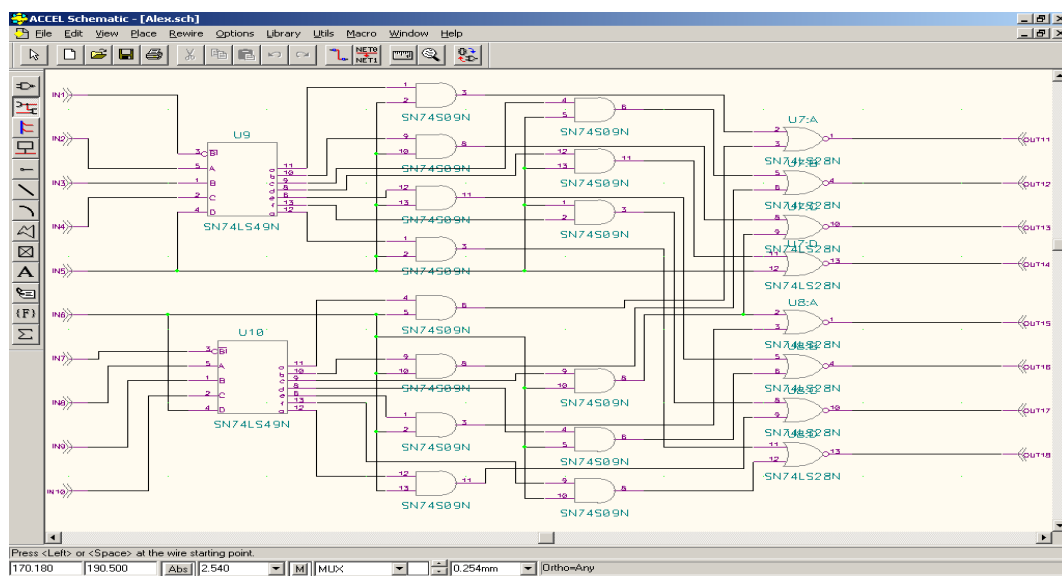


Рисунок 8.24 – Вікно редактора ACCELSchematic з набраною схемою

Записавши у файл список ланцюгів за допомогою команди Utils/Generate Netlist, схема готова для завантаження у графічний редактор ACCEL PCB. Для цього зайдемо у редактор PCB та за допомогою команди Utils/Load Netlist прочитаємо список ланцюгів з файлу, який записали перед цим.

На екрані графічного редактора з'являться корпуси мікросхем та зв'язки між ними ("павутинка") (рис. 8.25).

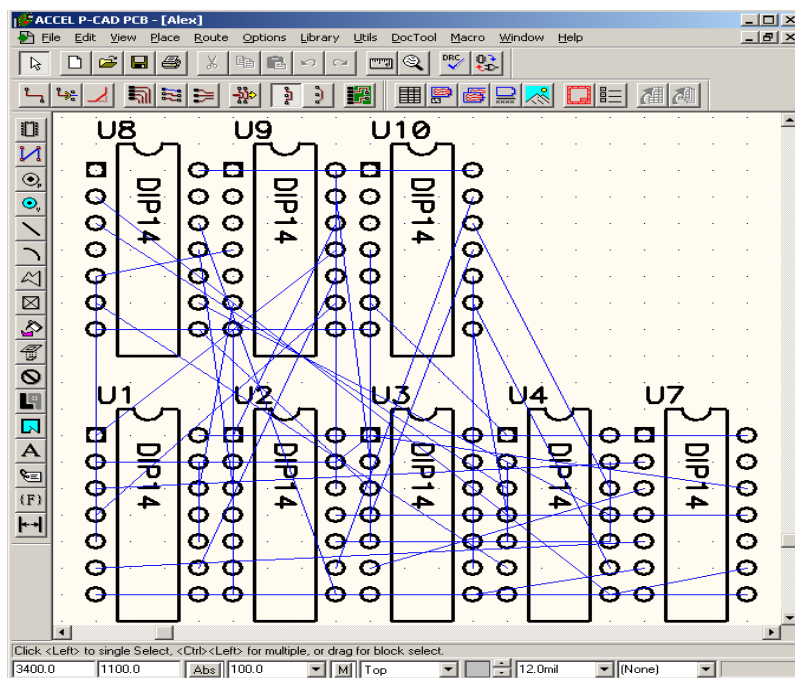


Рисунок 8.25 – Вікно графічного редактора ACCEL PCB з прочитаним файлом списку ланцюгів схеми

Надалі, перед тим як скористатися програмами автоматичної розводки чи виконати її вручну, потрібно зробити декілька важливих речей по підготовці схеми для розводки на друкованій платі:

1. Для безпосередньої розводки потрібно мати контури друкованої плати, на якій планується набирати цю схему. Для цього потрібно нарисувати за допомогою редактора ACCEL PCB контури друкованої плати на спеціально для цього призначеному шарі.

2. Якщо потрібно, наносять різні отвори, шари металізації, місця під з'єднувальні елементи, як наприклад, стандартний 9-піновий порт, який використовується у кожному комп'ютері. Роблять додаткові з'єднання, якщо вони потрібні.

3. Розміщують компоненти, які використовуються на схемі. Під час розміщення користуються автоматичними засобами розміщення, але, як правило, на більш менш складних платах цей спосіб себе зовсім не виправдовує. Тому майже завжди у реальному житті користуються способом ручної розстановки компонентів, виходячи із власного досвіду та існуючих обмежень.

В результаті буде отримана майже готова схема, яку потрібно тільки розвести. Приклад схеми, підготовленої для автоматичної розводки, наведений на рисунку 8.26. Хоча під час безпосередньо розводки виникає теж дуже багато моментів, які необхідно враховувати.

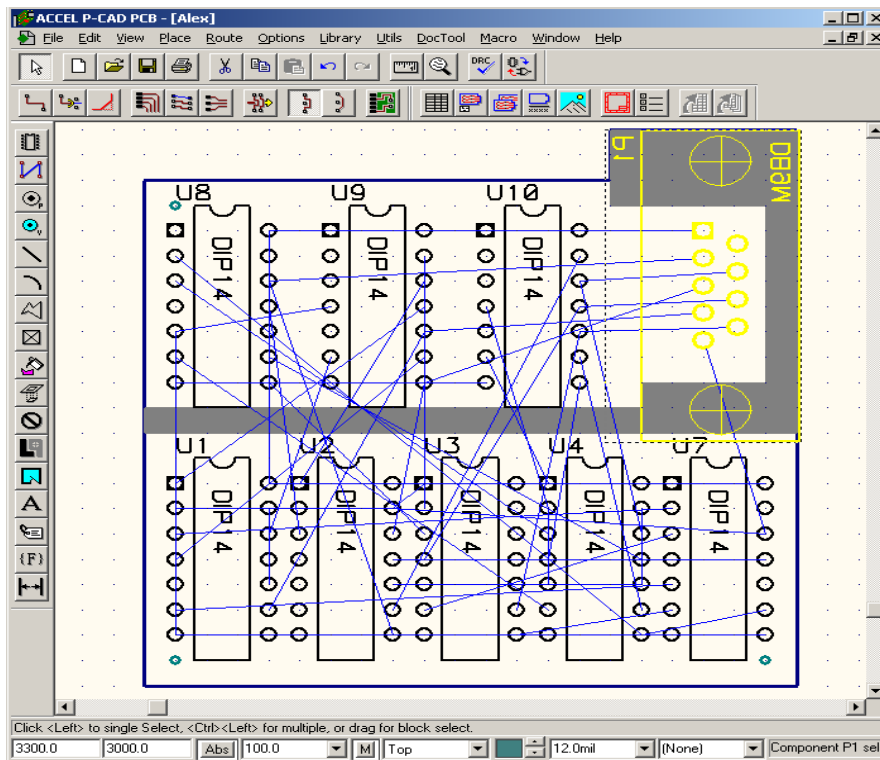


Рисунок 8.26 – Схема, підготовлена для автоматичної розводки

Після розводки друкована плата може мати, наприклад, такий вигляд (рис. 8.27):

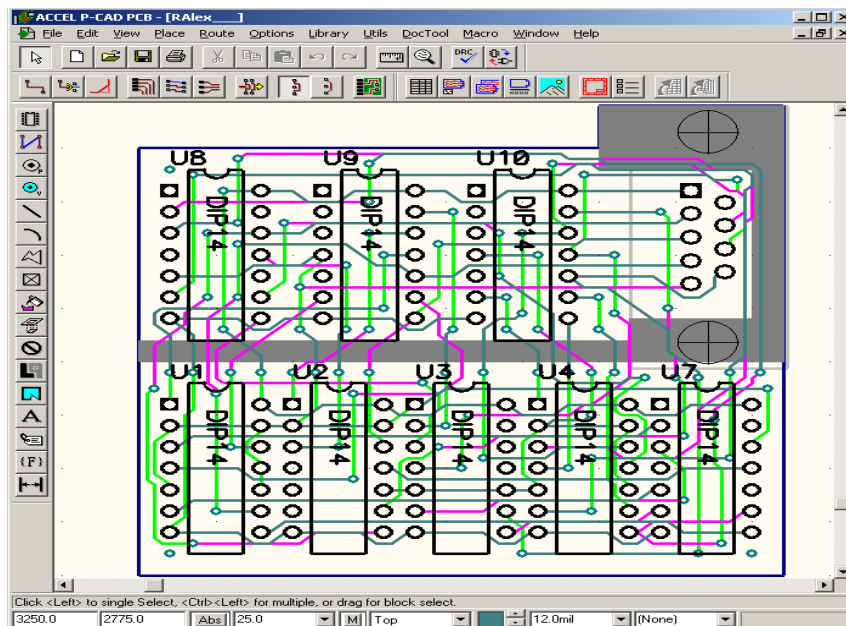


Рисунок 8.27 – Остаточний вигляд схеми



## Контрольні питання

1. Назвіть відомі вам методи пакування схем на друковану плату.
2. Як запакувати схему за допомогою обраної вами системи проектування?
3. Які кроки потрібно виконати, щоб набрати схему у редакторі принципів схем обраної САПР?
4. Як формуються мікросхеми з відповідних логічних елементів?
5. Як визначається коефіцієнт використання мікросхеми схеми пристрою?
6. Що таке файл списку ланцюгів схеми і як він формується?
7. Наведіть етапи пакування компонентів на друковану плату.

## Рекомендована література

1. Норенков И. П. Основы теории и проектирования САПР : учебник для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1990. – 335 с.
2. Норенков И. П. Системы автоматизированного проектирования электронной и вычислительной аппаратуры : учебное пособие для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1983. – 272 с.
3. Питерсон Дж. Теория сетей Петри и моделирование систем / Питерсон Дж. ; пер. с англ. – М. : Мир, 1984. – 264 с.
4. Рафикузаман М. Микропроцессоры и машинное проектирование микропроцессорных систем : в 2-х кн. / Рафикузаман М. ; пер. с англ. – М. : Мир, 1988.
5. Системы автоматизированного проектирования : учебное пособие для вузов : в 9 кн. / [под ред. И. П. Норенкова]. – М. : Высшая школа, 1986. – 159 с.
6. Системы автоматизированного проектирования в радиоэлектронике : [справочник] / Е. В. Авдеев, А. Т. Еремин, И. П. Норенков, М. И. Песков ; под ред. И. П. Норенкова. – М. : Радио и связь, 1986. – 368 с.
7. Соловьев В. В. Проектирование функциональных узлов цифровых систем на программируемых логических устройствах : учебное пособие для вузов / Соловьев В. В. – М. : ПКООО Бестпринт, 1996. – 252 с.
8. Роїк О. М. Основи автоматизованого проектування складних об'єктів і систем : навчальний посібник / Роїк О. М., Савчук Т. О., Ткаченко О. М. – Вінниця : ВДТУ, 2001. – 111 с.

## ЛАБОРАТОРНА РОБОТА № 9

### ТРАСУВАННЯ ДРУКОВАНИХ ПЛАТ ЗА ДОПОМОГОЮ ПРОГРАМИ ТРАСУВАННЯ ОБРАНОЇ САПР

Мета: вивчити методи та принципи трасування компонентів за допомогою програми трасування обраної САПР та закріпити ці знання, виконавши індивідуальне завдання по трасуванню компонентів на платі.

#### Порядок виконання роботи

1. Ознайомитися із загальними відомостями про методи автотрасування за допомогою обраної системи проектування.
2. Набрати схему в редакторі принципів схем згідно з варіантом завдання лабораторної роботи.
3. Провести розводку спроектованої схеми відповідно до завдання, виданого викладачем.
4. Отримати статистичну інформацію по закінченню процесу розводки спроектованої схеми.
5. Оформити звіт та зробити висновки по виконаній роботі.

**Якщо ви будете використовувати САПР PULSONIX, то порядок виконання лабораторної роботи передбачає такі дії:**

1. Ознайомитися із загальними відомостями про методи автотрасування в САПР PULSONIX за допомогою програми AutoRoute.
2. Набрати схему у редакторі Schematic згідно з варіантом завдання лабораторної роботи № 1.
3. Провести розводку спроектованої схеми в САПР PULSONIX відповідно до завдання, виданого викладачем.
4. Отримати статистичну інформацію по закінченню процесу розводки спроектованої схеми.
5. Оформити звіт та зробити висновки по виконаній роботі.

#### Теоретичні відомості

##### САПР PULSONIX

##### *Налаштування конфігурації та вибір необхідних параметрів*

Для розміщення компонентів та ручного трасування друкованих плат використовується редактор РСВ. На рисунку 9.1 показано вікно редактора разом із завантаженою друкованою платою.

Налаштування конфігурації графічного редактора РСВ виконується майже так само як в редакторі Schematic, але існують свої особливості.

Для налаштування параметрів графічного редактора РСВ використовується команда Tools – >Options. Діалогове вікно, яке використовується

для конфігурації графічного редактора має вигляд, як показано на рисунку 9.2.

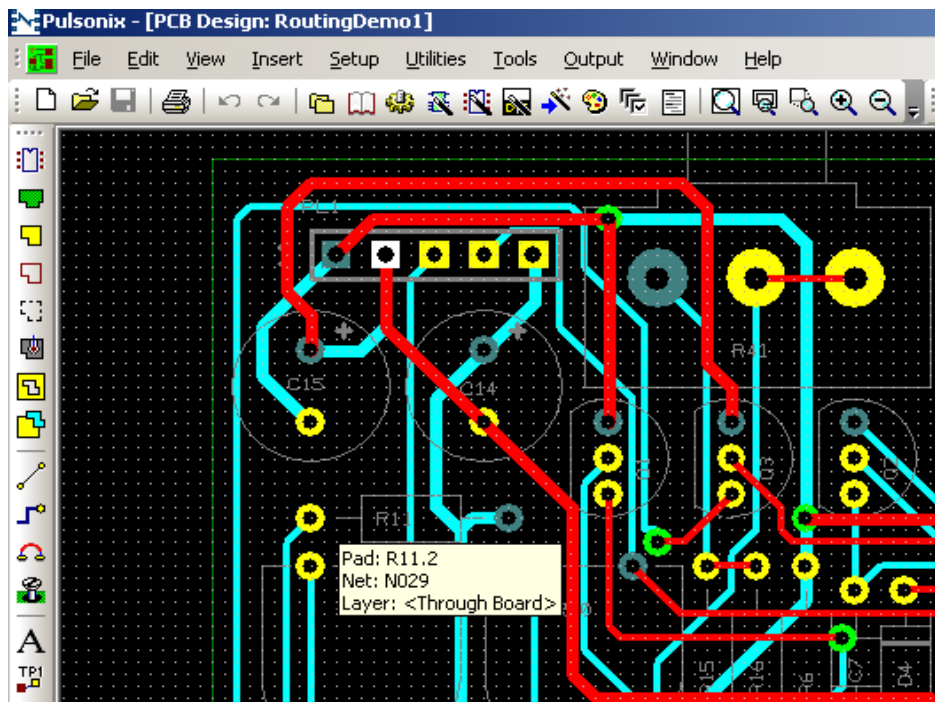


Рисунок 9.1 – Головне вікно програми РСВ

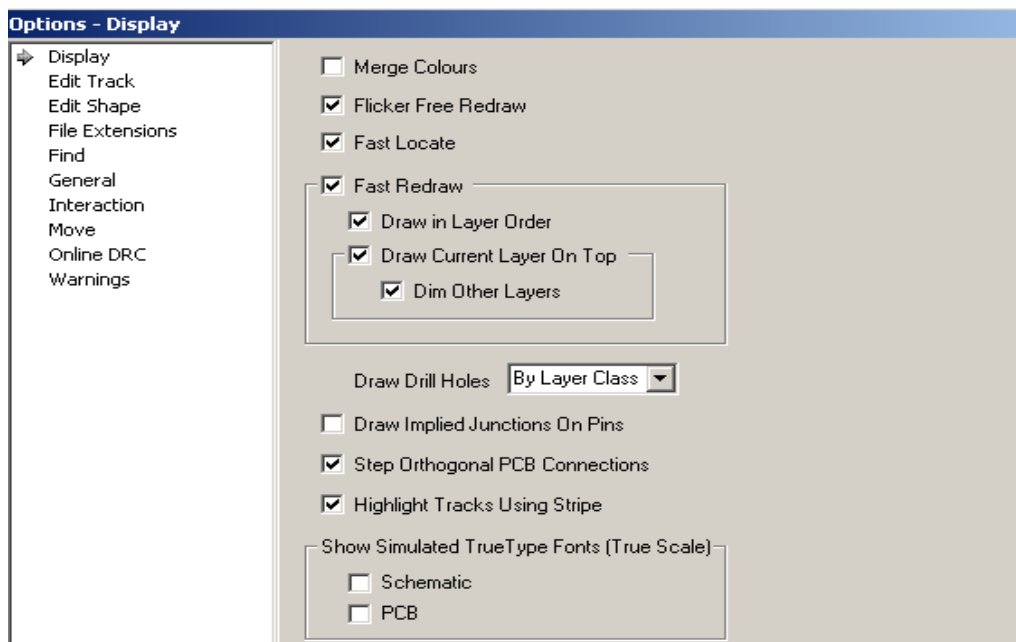


Рисунок 9.2 – Діалогове вікно для конфігурації графічного редактора

Для запуску діалогу конфігурації параметрів ручної та автоматичної розводки необхідно виконати *Tools* → *AutoRoute* → *RouteWholeDesign* і обрати необхідні налаштування для проекту у вікні Auto Router (рис. 9.3).

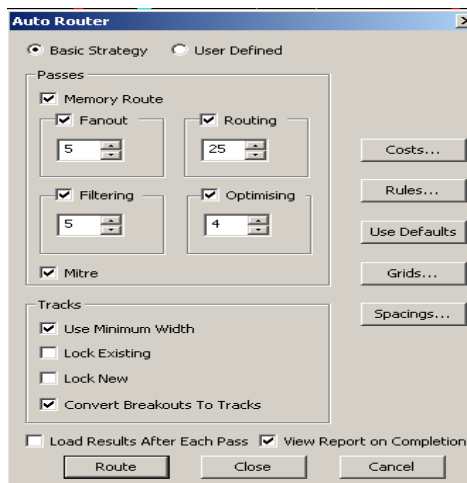


Рисунок 9.3 – Вікно Auto Router

В цьому діалоговому вікні користувач може обрати різні види проходів (Passes):

*Memory* – це прохід, що посилає регулярні масиви шпильок, які мають колінеарні зв'язки, де шпильки розділяють загальні координати X або Y. Це ефективно для масивів пам'яті.

*Fanout* – розгалуження. Це ряд проходів, які ви можете конкретизувати.

*Routing* – задаються параметри інтерактивної розводки.

Використовуючи засоби ППП PULSONIX, можна навести генерацію переліку з'єднань створеної принципової схеми. Для цього в програмі Schematic Design обирають Simulation>Create SPICE Net List File> Whole Design.

Обравши в меню команду *Tools* -> *TranslatetoPCB*, можна отримати схему, готову для розводки (рис. 9.4).

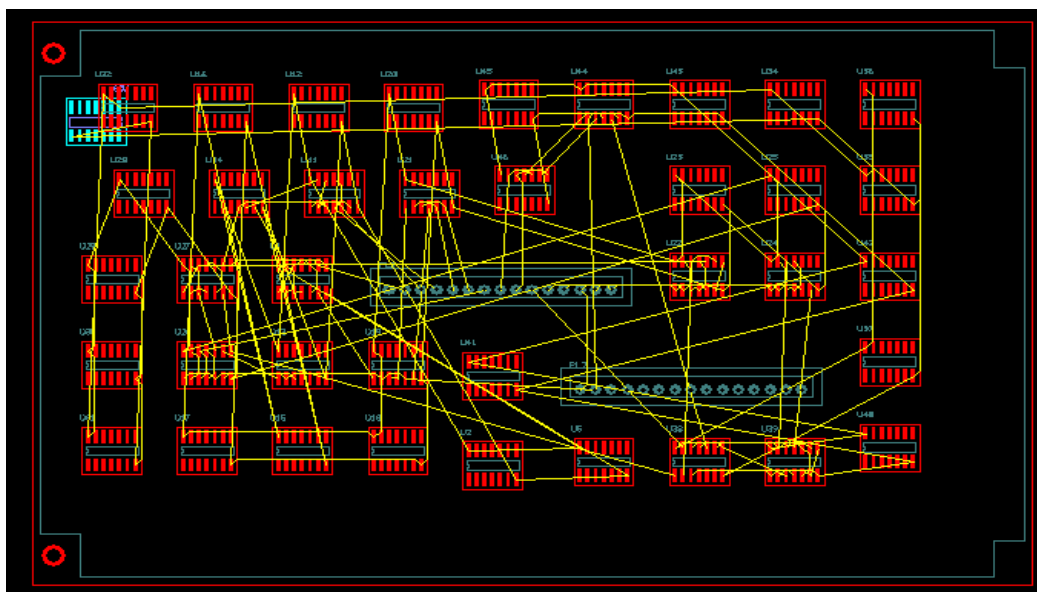


Рисунок 9.4 – Схема, готова для розводки

Автоматична розводка плати виконується за допомогою меню *Tools* → *AutoRoute* → *RouteWholeDesign*. На рисунку 9.5 зображена схема, яка була отримана за допомогою використання *Auto Route*.

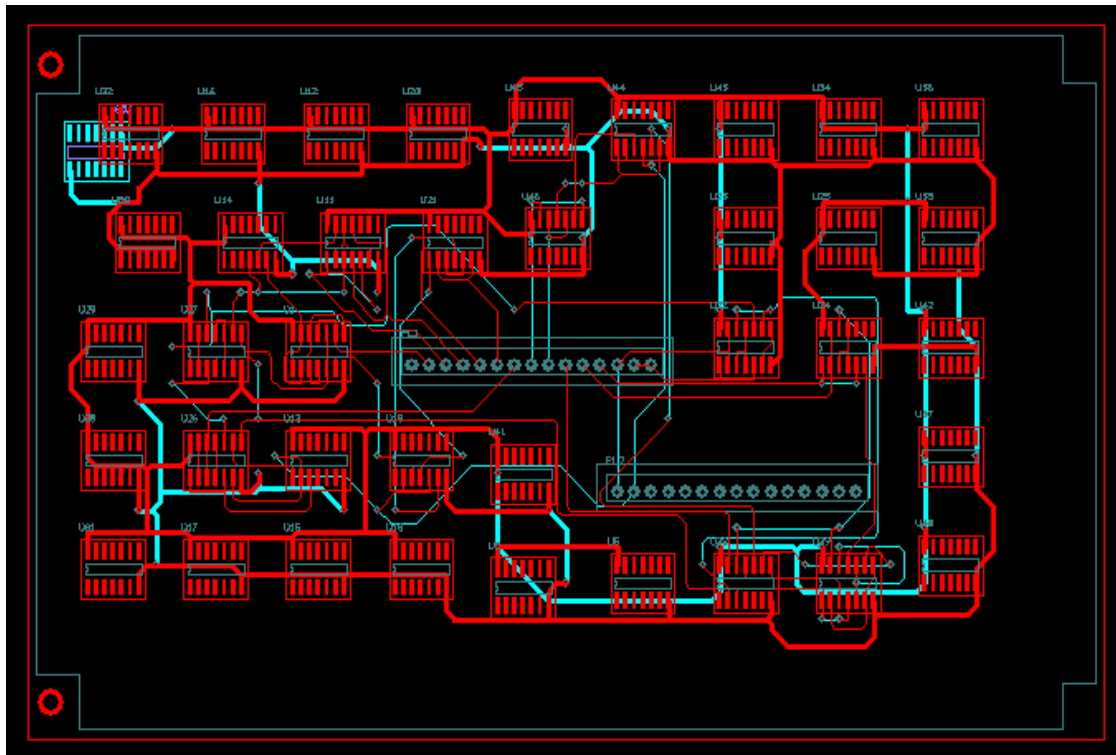


Рисунок 9.5 – Схема, отримана за допомогою використання *Auto Route*

**Якщо ви будете використовувати програму ручного трасування *DG Work*, то порядок виконання лабораторної роботи передбачає такі дії:**

1. Ознайомитися із загальними відомостями про методи ручного трасування компонентів за допомогою програми *DG Work*.
2. Набрати схему згідно з варіантом завдання лабораторної роботи № 2.
3. Провести розводку спроектованої схеми відповідно до завдання, виданого викладачем, в програмі *DG Work* із використанням евристичного алгоритму трасування з'єднань.
4. Отримати статистичну інформацію по закінченню процесу розводки спроектованої схеми.
5. Оформити звіт та зробити висновки по виконаній роботі.

## Теоретичні відомості

### *Програма *DG Work**

**Евристичні алгоритми** трасування з'єднань є більш швидкодієвими. На відміну від хвильового алгоритму, в них не аналізуються всі можливі траси для вибору оптимальної, а одразу намагаються прокласти трасу по найкоротшому шляху.



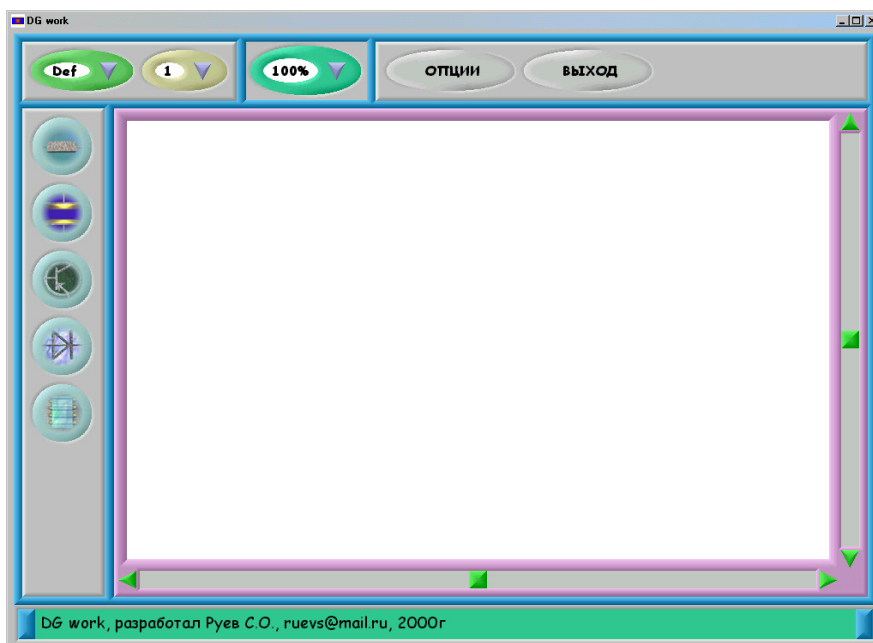


Рисунок 9.7 – Головне вікно середовища DG Work

Пояснимо значення кожного елемента управління, оскільки для правильної розводки друкованої плати необхідно обрати коректні параметри відповідної програми.

### ***Кнопка панелі інструментів***

При натисканні на цю кнопку можна обрати один з двох пунктів меню, а саме Def або Add і таким чином обрати один з двох наборів інструментів (рис. 9.8).



Рисунок 9.8 – Вибір панелі інструментів

### ***Елементи головної панелі інструментів***

На головній панелі інструментів знаходяться такі елементи як опір, ємність, транзистор, діод, мікросхема. Вигляд цих елементів показаний на рисунку 9.9 (послідовність відповідно до вищеперерахованих елементів).



Рисунок 9.9 – Елементи головної діагоналі

### *Елементи допоміжної панелі інструментів*

На допоміжній панелі інструментів знаходяться такі елементи як зажим, вузол з'єднання, ключ та стабілізатор. Вигляд цих елементів показаний на рисунку 9.10 (послідовність відповідно до вищеперерахованих елементів).

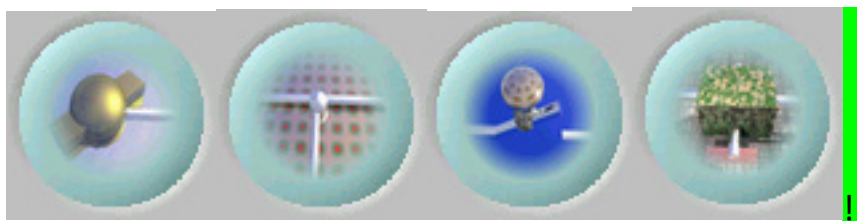


Рисунок 9.10 – Елементи допоміжної діагоналі

**Кнопка вибору шару** має вигляд, як показано на рисунку 9.11. При натисненні на неї обирається активний шар для розведення електричних доріжок. Коли обраний один з шарів, то всі доріжки, які розташовані в іншому шарі є неактивними і мають відповідний вигляд (лінії наведені сірим кольором).

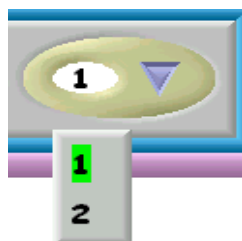


Рисунок 9.11 – Кнопка вибору шару

**Кнопка збільшення масштабу** має вигляд як показано на рисунку 9.12. При натисненні на кнопку всі елементи в робочому вікні збільшують свої розміри відповідно до обраного масштабу.

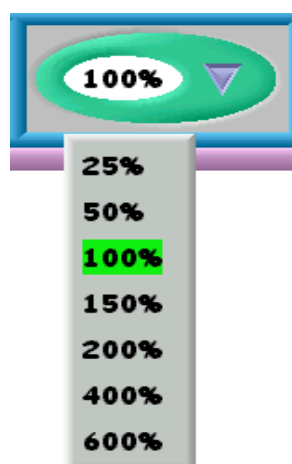


Рисунок 9.12 – Кнопка збільшення масштабу



**Кнопка головного меню.** При натисненні на цю кнопку на екрані з'являється цілий ряд кнопок, серед яких є такі команди як збереження проекту на диск, відкриття раніше створеного проекту, створення нового проекту, вихід тощо (рис. 9.13).



Рисунок 9.13 – Головне меню програми

Приклад розводки доріжок показаний на рисунку 9.14.

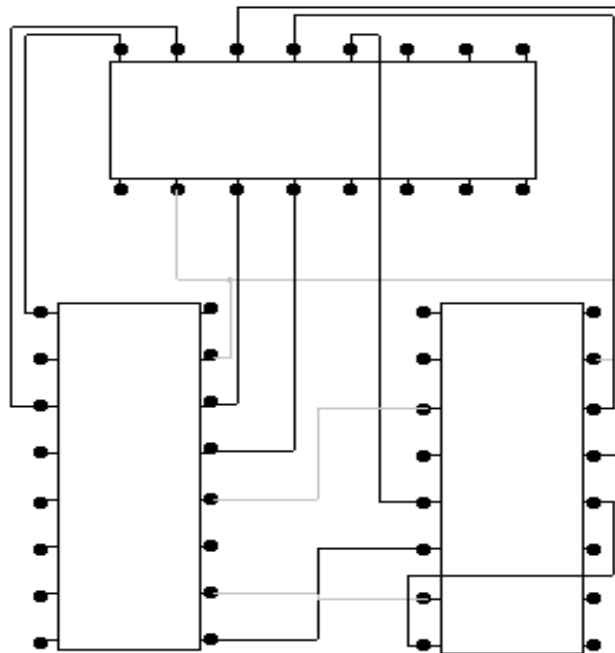


Рисунок 9.14 – Приклад розводки доріжок

### Контрольні питання

1. Методи трасування друкованих плат.
2. Програми обраної САПР для трасування друкованих плат.

3. Інструментарій програми обраної САПР для трасування друкованих плат.
4. Вимоги до трас друкованих плат.
5. Порівняльний аналіз класичних методів трасування друкованих плат та методу трасування друкованих плат обраної САПР.

### **Рекомендована література**

1. Норенков И. П. Основы теории и проектирования САПР : учебник для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1990. – 335 с.
2. Норенков И. П. Системы автоматизированного проектирования электронной и вычислительной аппаратуры : учебное пособие для вузов / И. П. Норенков, В. Б. Маничев. – М. : Высшая школа, 1983. – 272 с.
3. Питерсон Дж. Теория сетей Петри и моделирование систем / Питерсон Дж. ; пер. с англ. – М. : Мир, 1984. – 264 с.
4. Рафикузаман М. Микропроцессоры и машинное проектирование микропроцессорных систем : в 2-х кн. / Рафикузаман М. ; пер. с англ. – М. : Мир, 1988.
5. Системы автоматизированного проектирования : учебное пособие для вузов : в 9 кн. / [под ред. И. П. Норенкова]. – М. : Высшая школа, 1986. – 159 с.
6. Системы автоматизированного проектирования в радиоэлектронике : [справочник] / Е. В. Авдеев, А. Т. Еремин, И. П. Норенков, М. И. Песков ; под ред. И. П. Норенкова. – М. : Радио и связь, 1986. – 368 с.
7. Соловьев В. В. Проектирование функциональных узлов цифровых систем на программируемых логических устройствах : учебное пособие для вузов / Соловьев В. В. – М. : ПКООО Бестпринт, 1996. – 252 с.
8. Роїк О. М. Основи автоматизованого проектування складних об'єктів і систем : навчальний посібник / Роїк О. М., Савчук Т. О., Ткаченко О. М. – Вінниця : ВДТУ, 2001. – 111 с.

*Навчальне видання*

**Савчук Тамара Олександрівна  
Ольшанська Ольга Вікторівна**

**ТЕХНОЛОГІЯ КОМП'ЮТЕРНОГО ПРОЕКТУВАННЯ**  
Лабораторний практикум

Редактор Є. Плетньова

Оригінал-макет підготовлено Т. Савчук

Підписано до друку 15.01.2018.  
Формат 29,7×42¼. Папір офсетний.  
Гарнітура Times New Roman.  
Друк різнографічний. Ум. друк. арк. 7,02  
Наклад 50 (1-й запуск 1-20) пр. Зам. № 2018-019.

Видавець та виготовлювач  
інформаційний редакційно-видавничий центр.  
ВНТУ, ГНК, к. 114.  
Хмельницьке шосе, 95,  
м. Вінниця, 21021.  
Тел. (0432) 65-18-06.  
**press.vntu.edu.ua;**  
*E-mail: kivc.vntu@gmail.com.*

Свідоцтво суб'єкта видавничої справи  
серія ДК № 3516 від 01.07.2009 р.