

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

**Розробка друкованих плат
електронних пристроїв
у системі автоматизованого проектування**
P-CAD 2001

Методичні вказівки
до виконання лабораторних і курсових робіт
з дисципліни “ЕОМ та мікропроцесорні системи”

для студентів спеціальностей 6.0804.00
“Інформаційні управляючі системи та технології”,
“Інформаційні технології проектування”

КИЇВ КНУБА 2003

ББК 65.9(2)421
М27

Укладачі: О.А.Щербина, канд. техн. наук, доцент
М.М.Орлова, канд. техн. наук, доцент

Рецензент Л.І.Цилюрик, канд. техн. наук, доцент

Затверджено на засіданні кафедри інформаційних технологій, протокол № 2
від 23 вересня 2002 р.

Видається в авторській редакції.

М27 Розробка друкованих плат електронних пристроїв у системі автоматизованого проектування P-CAD 2001. Методичні вказівки до виконання лабораторних і курсових робіт з дисципліни “ЕОМ та мікропроцесорні системи” для студентів спеціальностей 6.0804.00 “Інформаційні управляючі системи та технології”, “Інформаційні технології проектування” /Укл. О.А.Щербина, М.М.Орлова. - К: КНУБА, 2002. – 52 с.

Викладені основи методики розробки друкованих плат у системі автоматизованого проектування P-CAD 2001, наведені завдання і методичні рекомендації до виконання лабораторних і курсових робіт.

Призначені для студентів спеціальностей 6.0804.00 “Інформаційні управляючі системи та технології”, “Інформаційні технології проектування”.

1. Загальні відомості про технологію виробництва друкованих плат

Конструктивною основою сучасних електронних пристроїв служить друкована плата. Ця назва вказує на те, що провідники, які з'єднують виводи встановлених на ній електронних компонентів, виготовляються методом друку. Заготовкою для плати служить покритий фольгою діелектрик, на який певним чином наноситься малюнок із зображенням провідників. Наступна технологічна операція травлення розчиняє незахищену малюнком поверхню фольги, утворюючи на поверхні діелектрика потрібну мережу з'єднань.

1.1. Типи плат

За типом монтажу розрізняють одnobічні, двобічні та багат шарові плати.

Одnobічні плати є найбільш простими за технологією виготовлення. Вони мають друковані провідники лише по один бік плати. Електронні компоненти вставляються в плату зі зворотного, непокритого фольгою боку.

Зрозуміло, що друковані провідники не можуть перетинатися, не з'єднуючись між собою. Тому схема з'єднань, що ними реалізується, має бути "пласкою". Оскільки не кожену схему можна представити в плоскому вигляді, то деяку кількість з'єднань доводиться виконувати за допомогою перемичок, тобто не друкованих, а звичайних провідників, що вставляються в плату так само, як і компоненти. Це, а також низька ефективність використання поверхні плати є суттєвим недоліком одnobічних плат.

Двобічні плати мають друковані провідники з обох боків. У порівнянні з одnobічними, їх виготовлення потребує лише однієї додаткової технологічної операції - металізації отворів. Металізовані перехідні отвори забезпечують з'єднання друкованих провідників протилежних боків плати. В результаті проблема уникнення перетину друкованих провідників вирішується дуже легко, бо з кожного боку плати траси прокладки провідників паралельні між собою. Переважний напрямок трас з одного боку плати можна умовно назвати північ-південь, з іншого схід-захід. Наприклад, для прокладки траси у південно-західному напрямку, її можна провести спочатку на південь, потім через перехідний отвір перейти на зворотний бік плати і далі - на захід. Це правило не є абсолютним, бо там, де інші провідники цьому не заважають, доцільніше проводити зв'язок по найкоротшій лінії, уникаючи зайвих перехідних отворів.

Отже, двобічні плати забезпечують можливість з'єднання без перемичок будь-яких точок на платі. Обмеження залишається тільки на кількість провідників, які можуть бути прокладені через певну ділянку плати. Визначається воно шириною цієї ділянки, шириною друкованих провідників та зазорів між ними.

Багат шарові плати за своєю будовою нагадують сандвіч, складений із окремо виготовлених тонких двобічних плат. Отже крім двох зовнішніх, вони мають ще певну кількість внутрішніх шарів друкованих провідників. Звичайно, технологія виготовлення таких плат є складнішою (найбільш складнощі викликає створення перехідних отворів між шарами плат, що виготовлялися окремо), проте їх перевага полягає, поперше, у більш щільному монтажі, а отже у зменшенні габаритів плати, по-друге, у можливості прокладки поряд з друкованими провідниками також і друкованих екранів, що захищають друковані провідники від різноманітних перешкод, наприклад, електромагнітних полів, що випромінюються сусідніми провідниками.

1.2. Формування малюнка з'єднань

На рис. 1 показаний фрагмент малюнка з'єднань друкованої плати. Основними його елементами є контактні площадки (1, 2, 3) та друковані провідники (4). Контактні площадки розміщуються навколо монтажних отворів, у які вставляються виводи ком-

понентів (1), перехідних отворів (2), що служать для з'єднання друкованих провідників у різних шарах плати, а також у місцях припаювання до плати виводів планарних компонентів (3).

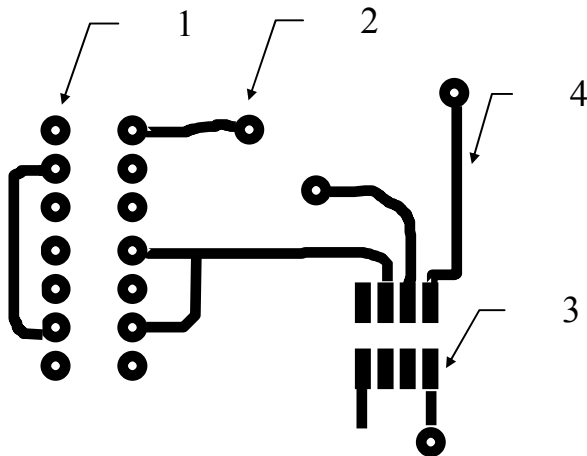


Рис. 1

Для побудови зображень провідників і контактних площадок найчастіше використовують фотокоординатограф. Це пристрій виводу векторного типу, який відрізняється від звичайного плотера тим, що зображення формується ним на світлочувливому матеріалі за допомогою променя світла, який проходить через спеціальний отвір, що називається апертурою. Наприклад, контактні площадки 1 і 2 можна сформувати апертурою, що має форму кільця, а контактну площадку 3 – прямокутною апертурою.

Елемент зображення відноситься до типу *flash* (спалах, проблеск),

якщо під час його формування промінь не рухається по поверхні малюнка.

До типу *line* (лінія) належать елементи, зображення яких формуються при переміщенні променя. Наприклад, зображення провідників 4 можна прокреслити круглою апертурою, діаметр якої дорівнює товщині провідника, а при відсутності прямокутної апертури для площадки 3, її зображення можна побудувати переміщенням квадратної апертури, що має таку ж, або меншу ширину. У останньому випадку це доведеться виконувати більше, ніж за один прохід.

Форми і розміри апертур дуже різноманітні. Існують навіть апертури з зображенням букв і цифр. Нажаль, загальних стандартів відповідності між кодом апертури і її формою не існує. Координатографи різних марок мають різні набори і різне кодування апертур. Те саме можна сказати й про кодування діаметрів свердел у верстаті з числовим програмним управлінням, що використовується для свердління отворів у платі. Тому файли, отримані в результаті автоматизованого проектування, потребують певного узгодження з характеристиками конкретного технологічного обладнання, настройки драйверів фотокоординатографа, свердлильного верстата тощо. Зрозуміло, що ці та інші специфічні питання технології виготовлення друкованих плат не розглядаються в рамках даного навчального курсу.

На завершення відзначимо, що спосіб формування зображення за допомогою фотокоординатографа не забезпечує і не потребує точної фотографічної відповідності між малюнком плати на екрані комп'ютера і на реальній платі, виготовленій за цим малюнком. Бо замість справжніх зображень елементів малюнка на екрані комп'ютера відображаються лише їх умовні позначення, тобто позначення центру та номеру апертури, якою виконується даний елемент зображення.

Останнім часом для нанесення малюнка з'єднань на поверхню плати застосовують також спеціальні фрезерні верстати, які, зрізуючи ділянки металізованого покриття плати, перетворюють суцільний шар металізації на окремі металізовані провідники. Звісно, така технологія набагато простіша, бо не потребує застосування хімічної обробки та світлочувливих матеріалів. Однак за якістю плат вона поки що поступається традиційній. Тому метод фрезерування застосовують здебільшого там, де треба швидко виготовити невелику кількість не дуже складних плат.

1.3. Встановлення компонентів

Компоненти зі штирковими виводами вставляються в отвори на платі по один її бік, так, що з іншого боку кінці виводів виступають над поверхнею плати на 1-2 мм. В умовах масового виробництва цю операцію виконує робот-маніпулятор.

Потім кінці виводів припаюються до контактних площадок. Найвищу якість забезпечують технологічні установки для паяння хвилею. Вони мають спеціальний резервуар, в якому циркулює розплавлене олово. Нижній бік плат, що рухаються на конвеєрній лінії над поверхнею резервуару, омивається хвилею розплавленого металу. В результаті краплинки олова затримуються на кінцях виводів і контактних площадках навколо них, що й забезпечує надійний контакт між ними.

Планарні компоненти мініатюрніші за штиркові, бо мають меншу відстань між виводами, до того ж вони можуть встановлюватися з обох боків плати. Однак процес виготовлення плат з планарними компонентами гірше піддається автоматизації.

Найбільш сучасними і досконалими є компоненти, що випускаються у корпусах без виводів. Роль зовнішніх контактів виконують у них металізовані ділянки поверхні корпусу, до яких прикріплені краплинки олова. Такі компоненти набагато мініатюрніші і дешевші за звичайні. Випускаються вони прикріпленими до довгої паперової стрічки, намотаної на бобіну. Це полегшує взяття компонента роботом-маніпулятором.

Спочатку ці компоненти фіксуються на платі за допомогою клею. Потім плату поміщають у спеціальну піч. Там краплинки олова розплавляються і, стікаючи вниз, змочують поверхню контакту на корпусі компонента і контактну площадку плати під ним. В результаті, після охолодження між компонентом і друкованою платою встановлюється надійне механічне і електричне з'єднання.

В корпусах без виводів випускаються поки що компоненти, які мають невелику кількість зовнішніх електричних зв'язків.

2. Система автоматизованого проектування P-CAD

Система P-CAD¹ за статистикою є одним з найпоширеніших у світі засобів розробки друкованих плат. Більш того, з моменту своєї появи в 1989 році ця система стала в нашій країні фактично єдиним промисловим стандартом у даній галузі. Найбільш поширеною була версія для DOS – P-CAD 4.5, яка ще й досі застосовується багатьма підприємствами. Таке тривале застосування цієї версії програми пояснюється тим, що у пізніших версіях стали використовуватися інші формати вихідних файлів, а це ускладнювало застосування існуючого технологічного обладнання і робило перехід на нові версії програми економічно невиправданим.

Однак з плином часу і технічним переоснащенням вітчизняної промисловості нові версії програми набувають все ширшого практичного застосування. Тому в даній роботі розглядається остання на сьогодні версія програми – P-CAD 2001.

2.1. Структура і склад системи P-CAD 2001

Система P-CAD 2001 створена для роботи у середовищі Windows, тому в порівнянні з версіями для DOS вона має не тільки більш зручний і звичний для користувача інтерфейс, а й значно меншу кількість програм, які доводиться використовувати під час роботи з системою.

До складу P-CAD 4.5 входив не один десяток програм, кожна з яких виконувала свою специфічну функцію, формуючи на виході файли, які потім використовувались іншими програмами в якості вхідних. Наприклад, щоб надрукувати малюнок, створе-

¹ В літературі досить часто цю назву пишуть також і без дефісу: PCAD. Крім того, певний час правами власності на даний програмний продукт володіла фірма ACCEL Technologies, тому нові версії програм, які виходили в цей період, відомі під назвою ACCEL EDA. Зараз права власності на систему перейшли до австралійської фірми Protel International, яка відновила історичну назву програми – P-CAD.

ний в одному з графічних редакторів, треба було спочатку сформувати в цьому редакторі відповідний файл, а вже потім виводити його на друк за допомогою спеціальної програми PCPRINT.

P-CAD 2001 також має чимало утиліт, наприклад тих, що виконують перетворення форматів даних, імпортованих з інших програм, проте всю роботу зі створення принципів схем і розробки друкованих плат можна виконати, застосовуючи всього дві чи три програми, а саме:

- *Schematic* – графічний редактор для створення і редагування принципів схем;
- *PCB* – графічний редактор для створення і редагування друкованих плат;
- *Library Executive* – програму для роботи з бібліотеками компонентів, що дозволяє з використанням програм *Symbol Editor* і *Pattern Editor* створювати і редагувати їх позначення на принциповій схемі та зображення їх корпусів на платі.

Графічні редактори *Schematic*, *PCB*, та їх спрощені варіанти *Symbol Editor* і *Pattern Editor* мають чимало спільних рис, які ми й розглянемо спочатку.

2.2. Графічні редактори P-CAD 2001

Графічні редактори системи P-CAD 2001 мають традиційну для застосувань Windows організацію та інтерфейс: у верхній частині вікна програми розташоване основне меню, під ним – кнопки найбільш вживаних операцій, знизу – статусний рядок (рис. 2), куди виводяться системні повідомлення та інформація про стан редактора.

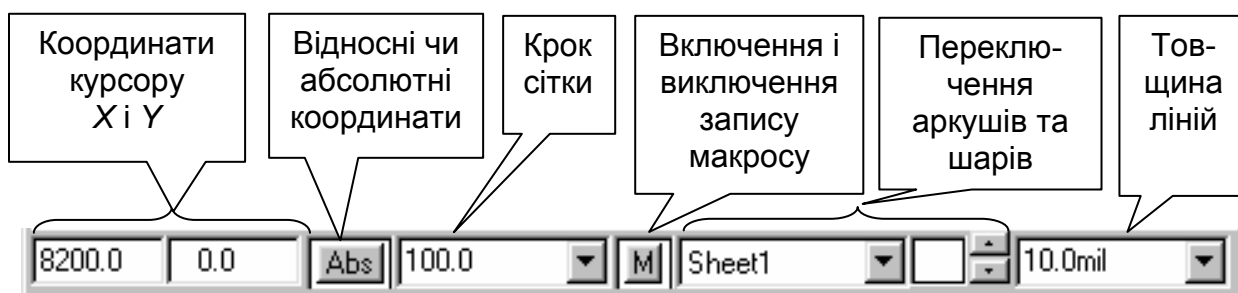


Рис. 2

В перших двох полях статусного рядка відображається інформація про поточні координати курсору X і Y . Далі розташована кнопка **Abs | Rel**, в залежності від стану якої в попередніх полях відображаються абсолютні чи відносні координати. В першому випадку вони відраховуються від лівого нижнього кута аркуша, в другому – від довільної точки, абсолютні координати якої можна задати у вікні, що викликається командою *Options > Girds...*².

Справа від вказаної кнопки відображається крок координатної сітки. Він визначає дискретність переміщення курсору, а отже й щільність розташування елементів на створюваному малюнку. Значення кроку можна вводити в це поле з клавіатури або вибирати зі списку раніше введених значень. Вводити числові значення можна також і в поля координат X і Y . Тоді курсор автоматично переміститься у відповідну точку аркуша.

Координати можуть відображатися в міліметрах (*mm*), дюймах (*inch*) або мілах (*mil*). Один міл дорівнює 0,001 дюйма. Зображення у P-CAD 2001, в залежності від обраної системи одиниць, виконуються з точністю 0,01 мм або 0,1 міла. Змінити одиницю виміру можна на будь-якому етапі роботи з проектом, відкривши вікно конфігурації командою *Option > Configure* (рис. 3).

² Тут і далі символом ">" відділяємо одну опцію меню від іншої, що вибирається після неї.

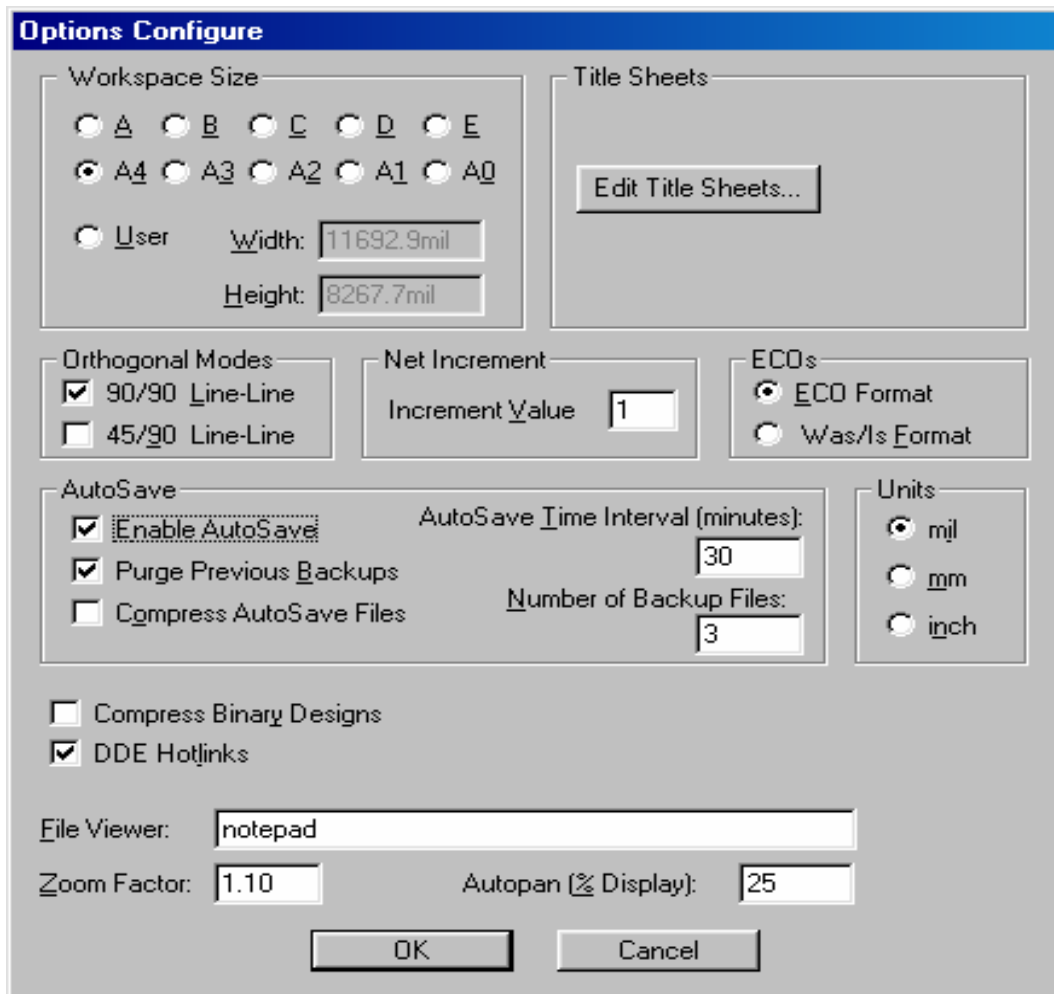


Рис. 3

Праворуч від поля кроку координатної сітки (рис. 2) знаходиться кнопка **M**, натискання якої розпочинає або закінчує запис команд у макрос. Це робиться для того, щоб потім можна було швидко повторити всі записані в макрос команди. Для цього достатньо натиснути клавішу **E**.

Далі в статусному рядку йдуть два поля, які дозволяють вибрати аркуш схеми в редакторі *Schematic* чи шар зображення в редакторі *PCB*. У другому полі відображається колір, яким показані елементи, розташовані в даному шарі.

Останнє поле статусного рядка задає товщину ліній. Справа від нього відображається скорочена інформація про виділений на екрані об'єкт. Виділення об'єктів і робота з ними відбувається за загальними правилами, прийнятими у Windows. Щоб отримати детальніші відомості про виділений об'єкт, а при потребі й відредагувати деякі з них, треба клацнути по екрану правою клавішею мишки і вибрати в контекстному меню опцію *Property*.

Для перегляду зображень у графічних редакторах користуються клавішами переміщення курсору, лінійками прокрутки та командами меню *View*. Зокрема команда

- *View > All* відображає у вікні редактора весь аркуш;
- *View > Extent* відображає всю заповнену частину аркуша в максимальному для поточних розмірів вікна масштабі;
- *View > Last* показує попереднє зображення вікна;
- *View > Center* намагається розмістити зображення так, щоб точка, на яку вказує курсор, стала центром вікна. Ця команда виконується також при натисканні клавіші **C**. Нею зручно користуватись для "панорування" зображення.

3. Робота з графічним редактором *Schematic*

3.1. Встановлення конфігурації

Перед початком створення принципової схеми виконайте команду *Option > Configure* (рис. 3) та виберіть формат аркуша в європейській (A4, A3, ...A0) чи американській (A, B, ... E) системі. Майте на увазі, що аркуш у зазначених вище стандартних форматах завжди орієнтується так, що його довша сторона розташована горизонтально. Можна також задати аркуш довільних розмірів і орієнтації, вибравши радіокнопку *user* та заповнивши поля *Width* (ширина) і *Height* (висота).

Аркуш, формат якого перевищує можливості вашого принтера, друкуватиметься на декількох сторінках, які потім можна склеїти. Альтернативою цьому є побудова схеми на декількох окремих аркушах (сторінках), що зберігаються в одному файлі. Створення, вилучення та перейменування аркушів виконується за допомогою кнопок *Add*, *Delete*, *Modify*, розташованих на закладці *Sheets* у вікні, що відкривається командою *Option > Sheets*. Аркуш, що відображається у вікні редактора, вибирається кнопками статусного рядка (рис. 2). Провідники переходять з аркуша на аркуш за допомогою ліній групового зв'язку (див. розділ 3.4).

У вікні *Option > Configure* задають інші важливі параметри, зокрема:

- *Orthogonal Modes* визначає кути, під якими можна проводити лінії на схемі: 90/90, дозволяє проводити лінії тільки під кутами, кратними 90°, а 45/90 – також під кутами, кратними 45°. Обмеження, що задаються цим параметром, можна включати і виключати клавішею *O*.
- *Net Increment* задає величину, на яку змінюватиметься позиційне позначення компонента, при розміщенні на схемі його наступного екземпляра. Наприклад, якщо *Increment Value = 1*, то після розміщення резистора *R1* його наступним екземплярам автоматично надаватимуться позиційні позначення *R2*, *R3* і т.д.
- *Auto Save* дозволяє задати параметри автоматичного зберігання файлів на диску для запобігання втрати інформації при раптовому виключенні живлення чи зависанні комп'ютера.
- *Units* задає одиниці виміру: міли (*mil*), міліметри (*mm*) або дюйми (*inch*). Один міл дорівнює 0,001 дюйма, один дюйм дорівнює 2,54 мм.
- *DDE Hotlinks* встановлює "гарячий зв'язок" між редакторами *Schematic* і *PCB* так, що виділення кольором (*Highlight*)³ певних компонентів чи провідників в одному з редакторів відразу робить з ними те саме в іншому.
- *Zoom factor* – це коефіцієнт збільшення чи зменшення масштабу зображення при натисканні клавіш "+" і "-" на цифровій (сірій) клавіатурі або відповідних кнопок на панелі інструментів. Щоб забезпечити можливість плавного регулювання масштабу зображення рекомендуємо встановити *Zoom factor = 1,1*.
- *Autopan (% Display)* показує на скільки відсотків свого розміру прокручуватиметься зображення аркуша, якщо курсор переміщувати клавіатурою за межі видимої частини вікна.


Внесення змін до конфігурації завершується натисканням кнопок *OK* або *Modify* і *Close*. Натискання лише кнопки *Close* закриває вікно без фіксації внесених змін.

Щоб працювати з принциповою схемою було зручно, встановіть в статусному рядку (рис. 2) крок сітки, що дорівнює відстані між выводами її компонентів.

³ Для виділення кольором компонентів чи провідників, треба спочатку просто їх виділити, а потім скористатися командою *Highlight* чи *Highlight Attached Net*. Остання виділяє кольором всі провідники зв'язані з даним компонентом. Для зняття виділення кольором використовуються відповідно команди *Unhighlight* та *Unhighlight Attached Net*, а також команда *Unhighlight all*, яка знімає всі виділення кольором. Вказані команди знаходяться в контекстному меню виділеного об'єкта та в меню *Edit*.

3.2. Розміщення символів

Умовні зображення компонентів (мікросхем, резисторів, роз'ємів тощо) називаються в P-CAD символами. Символи компонентів зберігаються в спеціальних бібліотеках. Отже, щоб розмістити на схемі потрібні символи, треба підключити бібліотеки, в яких вони описані. Для цього виконайте команду *Library > Setup* і за допомогою кнопок *Add...* (додати) і *Delete* (вилучити) сформуєте потрібний вам список бібліотек. Для виконання лабораторних і курсових робіт ми рекомендуємо використовувати бібліотеки, записані в папку *Lib_Course*, де ім'я бібліотеки здебільшого відповідає позиційним позначенням зібраних в ній компонентів (*R.lib* – резистори, *C.lib* – конденсатори, *VD.lib* – діоди) або серії інтегральних мікросхем (*_555.lib* – серія 555) тощо.

Щоб переключитися в режим розміщення символів, натисніть кнопку  або виберіть в меню *Place > Part* і клацніть мишкою по екрану. В результаті відкриється вікно діалогу (рис. 4), де в полі *Library* ви можете вибрати бібліотеку, а в полі *Component Name* – компонент цієї бібліотеки. Якщо натиснути кнопку *Browse*, то вибраний компонент відобразиться в правій частині вікна діалогу. Перш ніж розмістити компонент на схемі, можна заповнити поля *RefDes* і *Value*, що визначають позиційне позначення компонента і його номінал. Поле *RefDes* можна використати для того, щоб замінити американські позначення вітчизняними. Наприклад у нас мікросхеми прийнято позначати літерою *D*, а в США – *U*. В полі *Value* вказують номінали резисторів, конденсаторів, кварцових резонаторів, марки діодів, транзисторів тощо.

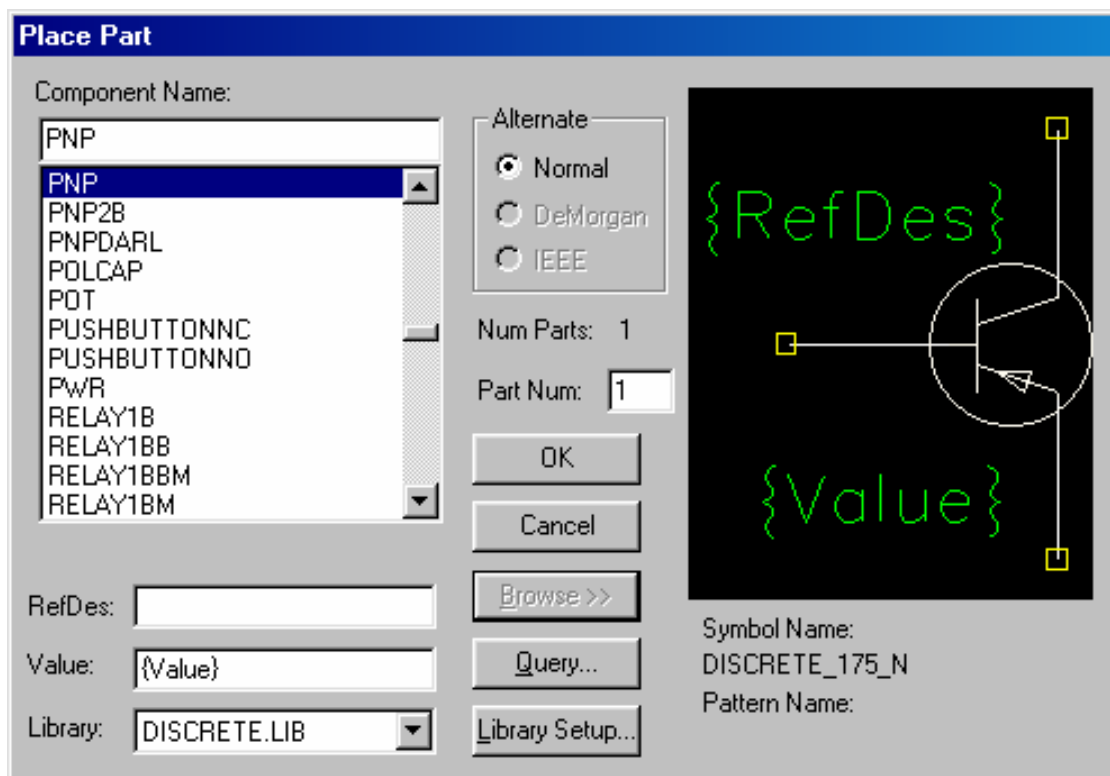


Рис. 4

Після натискання кнопки *OK* вікно діалогу закривається, а курсор набуває форми літери *X*. Екземпляри вибраного компонента вставлятимуться скрізь, де ви клацнете цим курсором⁴. При цьому їх позиційні позначення нумеруватимуться автоматично.

⁴ Компонент не розміститься, якщо клацнути мишкою по краю аркуша, де для компонента бракує місця.

Наприклад, якщо ви дали транзисторові позиційне позначення VT1, то клацнувши вдруге, ви розмістите транзистор з позиційним позначенням VT2 і т.д.

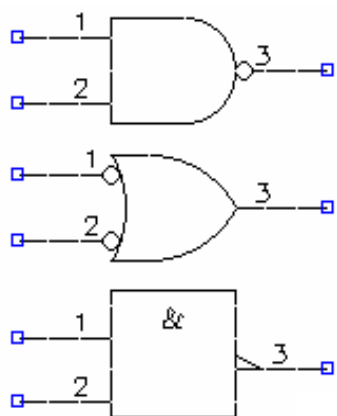


Рис. 5

Якщо компонент складається з декількох секцій (наприклад у мікросхемі 555ЛА3 розміщені чотири дво-входові елементи І-НЕ), то спочатку змінюватимуться позначення секцій: U1:A, U1:B, U1:C і U1:D, і тільки після розміщення останньої секції даного компонента розміщуватимуться секції наступного: U2:A, U2:B і т.д.



Зазначимо, що вказаним вище способом позначаються секції компонентів у стандарті США. За вітчизняними стандартами вони позначалися б D1.1, D1.2, D1.3 та D1.4, потім: D2.1, D2.2 і т.д. Як вже відзначалося, позиційне позначення D можна задати за допомогою поля *RefDes*. Що ж до цифрової нумерації секцій, то її можна забезпечити відповідними настройками програми (див. розділ 5.2.3, рис. 19).

Для деяких компонентів в бібліотеці передбачено можливість альтернативних варіантів їх графічного відображення, які можна обирати в полі *Alternate*. Наприклад, згаданий вище елемент І-НЕ в варіантах *Normal*, *DeMorgan* та *IEEE* відображається так, як показано на рис. 5: перший – це звичайне зображення І-НЕ, другий (згідно з правилом де Моргана) – у вигляді елемента НЕ-АБО, третій – у відповідності до стандартів IEEE.

Розміщення екземплярів вибраного компонента завершується натисканням правої кнопки мишки або клавіші *Esc*. Після цього описаним вище способом можна вибирати і розміщувати на схемі наступні компоненти.

Зображення компонента можна повернути або дзеркально відобразити. Для цього треба виділити компонент і натиснути відповідно клавішу *R* чи *F*.

3.3. Розміщення ліній електричного зв'язку (провідників)

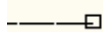
Для розміщення ліній електричного зв'язку (провідників) треба натиснути кнопку  або вибрати в меню: *Place > Wire*. Зазначимо, що ніякі інші лінії, наприклад ті, що розміщуються за допомогою кнопки , не утворюють електричних зв'язків між точками, які вони сполучають.


Щоб розмістити провідник, треба клацнути лівою кнопкою мишки у точці, де він починається, в точках, де він перегинається або з'єднується з іншими провідниками, а також у точці, якою закінчується цей провідник.

Можна також натиснути ліву кнопку мишки в початковій точці та утримувати її протягом всього процесу прокладки провідника, відпускаючи і знову натискаючи кнопку мишки в точках, де провідник перегинається або з'єднується з іншими провідниками. Цей спосіб дозволяє зразу бачити трасу провідника, що прокладається.

Можна прокладати провідник, поєднуючи обидва способи. Але, в будь-якому випадку, після завершення прокладки даного провідника і перед початком прокладання наступного, треба натиснути праву кнопку мишки або клавішу *Esc*. Тобто, як ви вже, певно, помітили, натискання правої кнопки мишки чи клавіші *Esc* використовується в P-CAD як ознака завершення "багатоходової" команди.

В P-CAD кінці ні з чим не з'єднаних провідників відображаються квадратами



, а місця з'єднання провідників – точками . Завдяки цим позначенням, можна легко пересвідчитись у наявності електричних контактів там, де вони потрібні, та їх відсутності там, де вони не потрібні. Якщо провідник прокладено не так, як ви хотіли, то можна добудувати чи перебудувати його або, скориставшись кнопкою

чи командою *Edit > Undo*, відмінити останню команду й побудувати провідник заново.

За деякими винятками, всі провідники в цифровій техніці прийнято проводити тільки горизонтально або вертикально. Тому в полі *Orthogonal Modes* вікна конфігурації треба відмітити *90/90 Line-Line*, як це зроблено на рис. 3. Однак не забувайте, що дане обмеження на кути прокладання ліній можна включати й виключати клавішею *O*. Якщо це обмеження виключене, то в правій частині статусного рядка, відображається *Ortho=Any*, якщо воно включене, то *Ortho=90*.

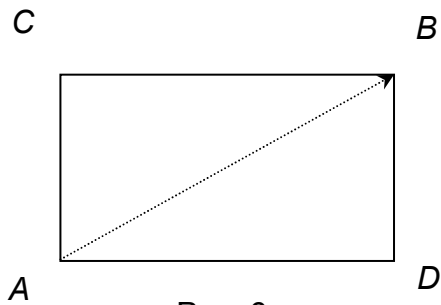


Рис. 6

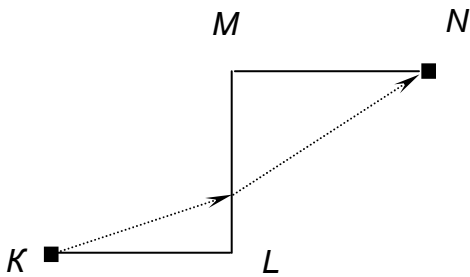




Рис. 7

В останньому випадку, якщо, натиснувши ліву клавішу мишки, провести курсором по діагоналі з точки *A* в точку *B* (рис. 6), то на екрані замість гіпотенузи *AB* відобразатимуться два катети: *AC* і *CB* або *AD* і *DB*. Натисканням клавіші *F* ви можете обирати, через яку точку *C* чи *D* пройде провідник. Відпустивши мишку в точці *B*, ви зафіксуєте точку перегину: *C* або *D* (але не *B*!) і продовжите прокладання провідника.

Наприклад, найпростіший спосіб з'єднати точки *K* і *N* відрізками *KL*, *LM* і *MN* (рис. 7) – це натиснути ліву кнопку мишки в точці *K*, відпустити її у довільній точці, що належатиме відрізку *LM*, знову натиснути її і відпустити в кінцевій точці *N*.

Якщо в подальшому ви побажаєте, наприклад, змінити місце проходження відрізка *LM*, то для цього достатньо натиснути піктограму , виділити відрізок *LM*, клацнувши по ньому мишкою, та перетягнути його ліворуч чи праворуч. Отже при редагуванні траєкторії прокладання провідника ключову роль відіграють його вузлові точки. Ви можете переміщувати їх разом з відрізками, якими вони сполучаються, чи розривати зв'язки між виділеними вузловими точками за допомогою клавіші *Delete*. В


складніших випадках може знадобитися встановлення додаткових вузлових точок на існуючих відрізках. Для цього використовується кнопка .

Щоб прокладати провідники було зручно, встановіть в статусному рядку крок координатної сітки рівним відстані між выводами компонентів. Тоді ви не зможете промахнутися мимо виводу компонента. Також, натискаючи клавішу *X*, ви можете встановлювати зручну для вас форму курсору: стрілку, мале перехрестя чи перехрестя на весь екран.

За замовчуванням провідникам надаються імена *NET00001*, *NET00002* і т.д. При бажанні, можна змінити ім'я провідника чи відобразити його на схемі. Для цього треба виділити провідник, вибрати в контекстному меню опцію *Property*, та заповнити відповідні поля в вікні діалогу.

Якщо дати провідникам однакові імена, то вони вважатимуться з'єднаними. Але в даній версії P-CAD це можна зробити лише для провідників, імена яких є глобальними, зокрема тих, що підключені до ліній групового зв'язку (див. наступний розділ).

3.4. Розміщення ліній групового зв'язку (шин)

Лінії групового зв'язку (шини) розміщуються на схемі за допомогою кнопки  чи команди *Place > Bus*. Це робиться аналогічно описаному вище процесу прокладання провідників. Потім прокладаються й самі провідники. Кінець провідника, що заходить в шину чи виходить з неї, автоматично згинається (рис. 9) способом, вибраним в полі *Bus Connection Mode* вікна діалогу, що відкривається командою *Options > Display*. В цьому вікні можна задати й інші параметри зображення, зокрема кольори різних елементів схеми.

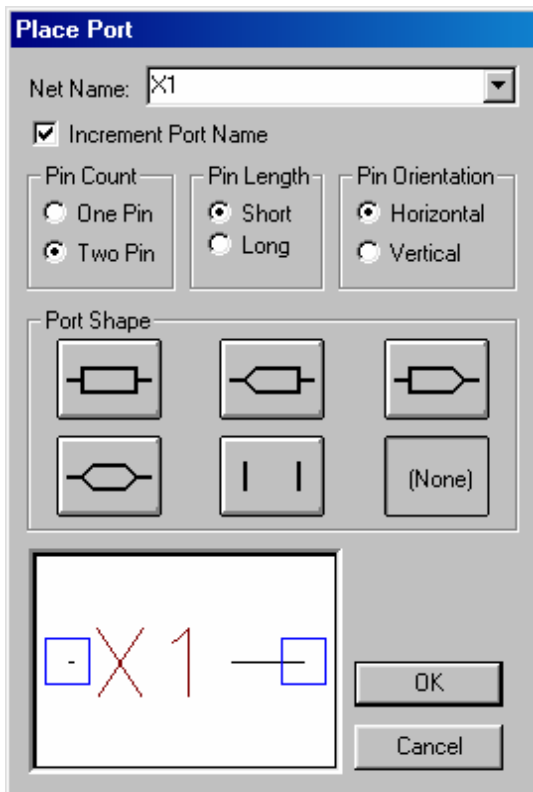



Рис. 8

Як відомо, відповідність між провідниками, що входять і виходять в лінію групового зв'язку, встановлюється за іменами провідників, які в даному випадку називаються портами. Щоб задати ці імена використовують кнопку  або команду *Place > Port*. Якщо після цього клацнути мишкою по екрану, то відкриється вікно діалогу *Place Port* (рис. 8), де можна задати наступні параметри.

- Ім'я провідника. Його можна або ввести в поле *Net Name*, або вибрати зі списку раніше введених імен.
- *Increment Net Name*. Відмітка в цьому полі забезпечує автоматичне збільшення номеру в імені провідника. Наприклад після введення імені *X1* наступним провідникам автоматично надаватимуться імена *X2*, *X3* і т.д.
- Спосіб графічного відображення імені. Ми радимо обирати *Two Pin*, *Horizontal*, *None*. Тоді ім'я провідника відобразиться без рамки в розриві горизонтальної лінії (рис. 9).

Після натискання кнопки *OK* це вікно закривається і курсор набуває форми літери *X*. Клацаючи цим курсором по провідниках, ми надаємо їм імена.

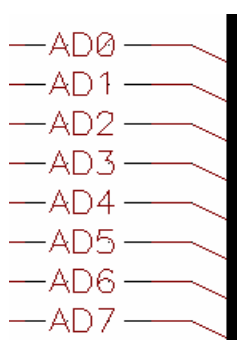


Рис. 9

Щоб пришвидшити побудову великої групи підключених до шини провідників, що мають однотипні імена, можна скористатися звичними правилами копіювання об'єктів, прийнятими у Windows. Наприклад, група провідників, показаних на рис. 9, будувалася так. Спочатку був прокладений верхній провідник, якому було надано ім'я *AD0*. Потім були виділені ім'я та всі фрагменти цього провідника, – від контактної площадки мікросхеми до шини. Провідники *AD1*, *AD2* і т.д. створювалися за допомогою перетягування мишкою виділеного фрагменту схеми при постійно натиснутій клавіші *Ctrl*. Цю роботу радимо виконувати при збільшеному масштабі зображення.

Тип і розмір шрифту, яким відображаються імена портів, можна вибрати, якщо виділити будь-який зразок такого імені, та вибрати в контекстному меню: *Property > Port > Text Styles > Port Style > Properties*. Внесені зміни стосуватимуться всіх імен портів даної схеми, в тому числі тих, що були побудовані раніше.

Так само можна змінювати стилі тексту, що використовується для позначень інших елементів схеми.

3.5. Друк схемних файлів

Перш за все слід відзначити, що друк в P-CAD не зводиться до простого перенесення зображення з екрана монітора на аркуш паперу. Під час друку воно може зазнавати значних трансформацій. Це потребує відповідних настройок програми, які ми й розглянемо нижче.

Спочатку, по команді *File > Print Setup*, як і в інших програмах Windows, треба виконати звичайні настройки принтера. Якщо в системі встановлено декілька драйверів принтерів, то виберіть той, що використовуватиметься в P-CAD, а також задайте формат і орієнтацію аркуша, виберіть лоток подачі паперу тощо.

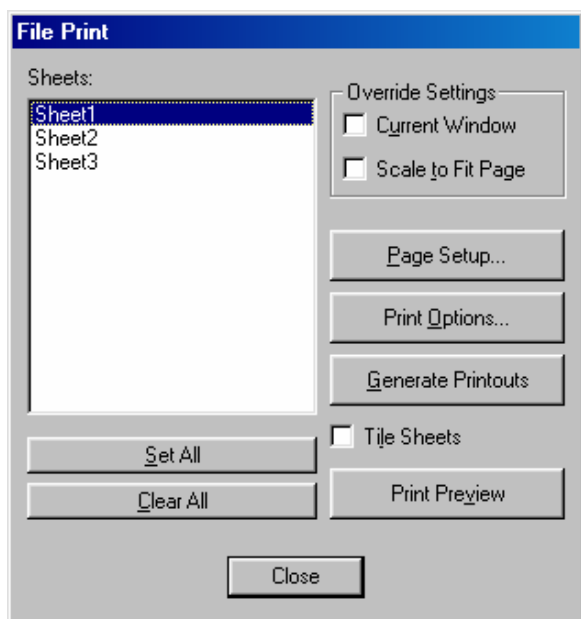


Рис. 10

При виконанні команди *File > Print* або натисканні відповідної кнопки панелі інструментів на екран виводиться вікно діалогу, показане на рис. 10.

В графі *Sheets* виділяють один чи декілька аркушів схеми, які треба надрукувати. Кнопки *Set All* і *Clear All* виділяють чи знімають виділення з усіх аркушів схеми.

Позначки в полях цього вікна означають:

- *Current Window* – друкувати не весь аркуш, а тільки ту його частину, яку видно в вікні редактора.
- *Scale to Fit Page* – привести масштаб зображення у відповідність до розмірів сторінки, що виводиться на принтер.
- *Title Sheets* – друкувати рамку зі штампом (форматку).

Кнопкою *Page Setup* задаються параметри, пов'язані з масштабом і орієнтацією зображення,

а кнопкою *Print Option* – визначають елементи зображення, що друкуються, та їх кольори при виводі на принтер. Вони відрізняються від кольорів на екрані.

Кнопка *Print Preview* дозволяє переглянути на екрані підготовлене для друку зображення, а кнопка *Generate Printouts* видає його безпосередньо на принтер.

3.6. Створення файла зв'язків

Схеми, побудовані в редакторі *Schematic*, зберігаються в файлах з розширенням *sch*. Вони відкриваються і записуються на диск звичайними командами *File > Open*, *File > Save*, *File Save As...*

Однак передача інформації з графічного редактора *Schematic*, де була створена принципова схема пристрою, в графічний редактор *PCB*, де розроблятиметься його друкована плата, здійснюється не через схемні файли, а через файли зв'язків, що зазвичай мають розширення *net* чи *alt* і містять перелік розташованих на схемі компонентів та опис зв'язків між їх выводами.

Для створення файла зв'язків в графічному редакторі *Schematic* треба вибрати в меню: *Utils > Generate Netlist*. В результаті відкриється вікно діалогу, в якому вибирається ім'я файла зв'язків та його формат.

Ім'я файла та папку для його розміщення, запропоновані за замовчуванням, можна змінити за допомогою кнопки *Netlist Filename*.

Вибір формату файла зв'язків залежить від вибору роутера – програми, що прокладатиме траси друкованих провідників. Для плат невеликого рівня складності, до яких належать студентські роботи, ми радимо користуватися форматом *Tango*.

4. Робота з графічним редактором *PCB*

4.1. Встановлення конфігурації

Основні параметри конфігурації редактора *PCB* задаються командою *Option > Configure*. Хоча вікно конфігурації в цьому редакторі й відрізняється від показаного на рис. 3, основні параметри, які тут вводяться, приблизно ті самі, що й описані в розділі 3.1. Це одиниця виміру довжини, розмір аркуша (його треба вибирати дещо більшим за розмір плати) тощо. У вікні *Option Configure* та інших вікнах конфігурації встановлюються також багато інших параметрів, які ми тут не розглядатимемо, бо детальне вивчення особливостей технології виробництва друкованих плат виходить за рамки нашого навчального курсу.

При роботі з редактором *PCB* важливе значення мають настройки екрана, що встановлюються командою *Option > Display*. Вона відкриває однойменне вікно діалогу, що містить таблицю, де стовпцям відповідають різні елементи зображення (*Items*), а рядкам – його шари (*Layers, слоу*). Наприклад, червоним кольором за замовчуванням, відображаються такі елементи зображення як лінії, полігони й тексти, розташовані в шарі *Top*, тобто в верхній частині плати (з боку встановлення компонентів), а ті самі елементи з протилежного боку плати (шар *Bottom*) – зеленим.



Перелік назв і призначень елементів зображення та шарів зображення, що за замовчуванням встановлюється при створенні нової плати, приведений в додатку 1. Цей набір шарів описує двобічну плату (див. розділ 1.1). Інші шари, в тому числі додаткові шари провідників для створення багатошарових плат, можуть додаватися (*Add*) чи вилучатися (*Delete*) користувачем у вікні діалогу *Options Layers*, де в графі *Layers* інформація про призначення, стани і номери шарів відображається так.





Літерою *S (Signal)* позначають сигнальні шари, в яких прокладаються друковані провідники, літерою *P – (Plane)* суцільні шари металізації для екранування та підведення живлення, а літерою – *N (Non signal)* інші не сигнальні, допоміжні шари.


Стан шару (виключений чи включений) позначається літерами *D* і *E* відповідно. Виключити чи включити відображення будь-якого виділеного шару, крім поточного (*Current*), можна за допомогою кнопок *Disable* і *Enable*, а кнопки *Disable All* і *Enable All* виключають і включають відображення всіх шарів плати. Переважний напрямок прокладки провідників у сигнальних шарах позначається літерами: *H* – горизонтальний, *V* – вертикальний, *A* – той, що обирається автоматично (за замовчуванням).

На завершення зазначимо, що вказані вище настройки редактора зберігаються у створюваних ним файлах друкованих плат і завантажуються разом з ними. Тому, для економії часу, потрібні настройки можна записати в порожній файл, з відкриттям якого встановлюватимуться ці настройки.

4.2. Способи розробки друкованих плат

В принципі, редактор *PCB* дозволяє розробляти друковані плати й без принципових схем. Для цього можна, скориставшись кнопкою  чи командою *Place > Component*, розмістити на платі необхідні компоненти, а кнопкою  або командою *Place > Connection* встановити зв'язки між їх виводами. Після цього можна запускати програму роутер, яка належним чином прокладе траси друкованих провідників.

Можна взагалі від початку до кінця розробити друковану плату вручну. Для цього потрібно вручну розмістити на платі контактні площадки компонентів  (*Place > Pad*), і знову ж таки вручну з'єднати їх друкованими провідниками в формі ліній  (*Place > Line*), дуг  (*Place > Arc*) та полігонів  (*Place > Polygon*). Звісно, провід-

ники доведеться прокладати в різних шарах, з'єднуючи їх де потрібно перехідними отворами  (*Place > Via*).

Однак зрозуміло, що така технологія створення друкованих плат придатна лише для найпростіших схем, до яких мікропроцесорні пристрої аж ніяк не належать. Тому далі ми розглядатимемо тільки автоматизовану розробку друкованих плат, виходячи з принципових схем, побудованих в редакторі *Schematic*. Однак, наведені вище засоби, при потребі, можуть використовуватися для ручного внесення певних коректив у друковані плати, розроблені засобами автоматизованого проектування.

Як вже відзначалося, передача інформації з графічного редактора *Schematic*, де була створена принципова схема пристрою, в графічний редактор *PCB*, де розроблятиметься його друкована плата, здійснюється за допомогою файла зв'язків. Але, перед завантаженням цього файла, в редакторі *PCB* треба обов'язково підключити бібліотеки всіх компонентів, розташованих на принциповій схемі, інакше при завантаженні файла зв'язків виникнуть помилки.

Підключення бібліотек виконується так само, як це робилося в редакторі *Schematic*. Треба виконати команду *Library > Setup* і за допомогою кнопок *Add* (дати) і *Delete* (вилучити) сформувані потрібний список бібліотек. На відміну від попередніх версій, P-CAD 2001 має спільні бібліотеки для редакторів *Schematic* і *PCB*.

Завантаження файла зв'язків здійснюється командою *Utils > Generate Netlist*. В результаті відкривається вікно діалогу, де можна обрати ім'я і тип файла. Застосування імені файла за замовченням тут, як і в редакторі *Schematic*, дозволяє економити час, бо передачу даних з редактора *Schematic* в редактор *PCB* доводиться виконувати досить часто. При цьому треба поставити позначку в полі *Optimize Nets*, що забезпечить передачу інформації потрібної для оптимізації мережі зв'язків.

Після успішного завантаження файла зв'язків, на екрані відображається зовнішній вигляд компонентів схеми, а зв'язки між їх выводами відображаються відрізками прямих.


4.3. Розміщення компонентів на платі

Перед розміщенням компонентів на платі, в статусному рядку або в меню *Options > Grids* встановлюють крок сітки, що дорівнює відстані між выводами компонентів. Зверніть увагу, що у вітчизняних компонентів ця відстань здебільшого прив'язана до міліметра, а в імпортованих – до дюйма. Зокрема для компонентів зі штирковими выводами вона становить відповідно 2,5 мм та 0,1 дюйма = 100 міл = 2,54 мм.

При розробці друкованих плат дуже важливим є оптимальне розміщення компонентів на платі. Від нього залежить довжина друкованих провідників, кількість перехідних отворів, інші характеристики, що визначають показники якості й надійності плати, а інколи й саму можливість її реалізації в заданих габаритах.

P-CAD 2001 має в своєму складі програму SPECCTRA, яка вирішує задачу оптимального розміщення компонентів на платі й подальшої прокладки провідників (трасування). Однак ця програма не проста в застосуванні, а її можливості дещо обмежені. Бажаючи, однак, можуть познайомитися в Інтернеті з правилами її використання, переглянувши сторінку [11]. Ми ж в подальшому обмежимося застосуванням ручного розміщення компонентів.

Після завантаження файла зв'язків на екрані відображаються всі компоненти схеми. З'єднання выводів компонентів показуються відрізками прямих. Якщо розміри плати задані, то побудуйте контури плати в шарі *Board* поруч з компонентами чи навколо них. Якщо розміри плати можуть бути довільними, то її контури можна побудувати пізніше, по завершенні розміщення компонентів, а деякі роутери, наприклад *QuickRoute*, дозволяють зробити це й після трасування.

Щоб вручну змінити розташування компонента на платі, треба в режимі  (*Select*) виділити цей компонент, клацнувши на ньому мишкою, і відбуксирувати його мишкою в потрібне місце. При переміщенні компонента разом з ним виділяються і його зв'язки з іншими компонентами. Розташування компонентів треба обирати так, щоб довжина зв'язків і кількість точок, в яких вони перетинаються, були мінімальними. Наприклад з двох варіантів розміщення мікросхеми, показаних на рис. 11, кращим є варіант *b*, бо його реалізація потребуватиме меншої кількості перехідних отворів.

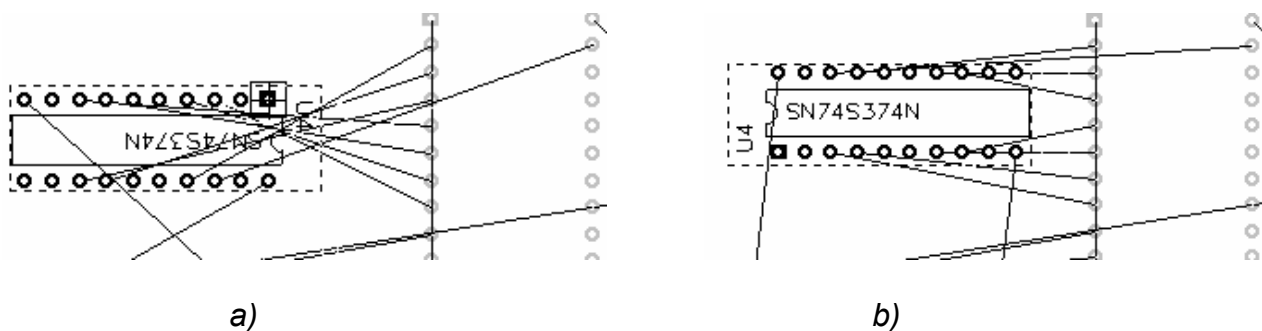


Рис. 11

Варіант *b*) був отриманий з варіанту *a*) шляхом повороту мікросхеми *U4* на 180° . Поворот виділеного компонента навколо своєї осі здійснюється за допомогою клавіші *R*. Клавіша *F* виконує дзеркальне відображення компонента, що фактично означає його перенесення на протилежний бік плати. Як відомо, розміщення компонентів по обидва боки плати застосовується лише для компонентів з планарними виводами.

Щоб, під час розміщення компонентів, було зручно відслідковувати проходження певних ліній електричного зв'язку, в команді *Edit > Nets* передбачена можливість включати і виключати відображення кожної з них. Для цього можна встановити чи зняти позначку біля назви лінії зв'язку в графі *Nets* або виділити потрібну групу ліній та скористатися кнопками *Hide Conns* (приховати зв'язки) і *Show Conns* (показати зв'язки). Там же є кнопки *Highlight*, *Unhighlight*, що застосовуються для виділення обраних ліній кольором, а також інші корисні функції. Наприклад, виділивши в графі *Nets* лінію зв'язку, в графі *Nodes* можна побачити перелік виводів компонентів, до яких вона підключена, а виділивши вивід в графі *Nodes* і натиснувши кнопку *Jump to Node*, можна встановити курсор на цей вивід.

Не слід забувати, що місце розміщення на платі певних компонентів диктується не тільки вимогами зручності прокладання провідників, а й функціональним призначенням цих компонентів. Наприклад, для уникнення перешкод у лініях живлення, на платі завжди встановлюють фільтруючі конденсатори. Так для мікросхем, виконаних за технологією TTL, бажано встановити по одному конденсатору ємністю 1-2 нФ між входами живлення мікросхеми у безпосередній близькості до них. Ці конденсатори нейтралізують високочастотну складову перешкод, що поширюються по лініях живлення в момент, коли переключення в інший стан однієї чи декількох мікросхем спричиняє різку зміну струму їх живлення. Для поглинання низькочастотної складової цих перешкод, встановлюють електролітичні конденсатори з розрахунку 1 мкФ на кожні 5-10 мікросхем, причому не менше половини цієї ємності треба зосередити у безпосередній близькості біля контактів роз'єму, через які живлення подається на плату.

Завершивши розміщення компонентів на платі, бажано провести оптимізацію мережі з'єднань шляхом перестановки логічно еквівалентних секцій і виводів⁵, що виконується командою *Utils > Optimize Nets*. В меню команди оберіть опцію *Auto* (автоматична оптимізація).

⁵ Поняття логічної еквівалентності секцій та виводів обговорюється в розділі 5.2.4.

Після цього слід зберегти створений файл на диску. Результуючий файл має стандартне розширення *pcb*, але може мати як двійковий (*Binary File *.pcb*), так і текстовий (*ASCII File *.pcb*) формат. Двійковий формат компактніший, проте він сприймається не всіма зовнішніми програмами. Для сумісності з ними застосовують текстовий формат.

4.4. Автоматичне трасування плати

Трасуванням називають процес прокладки трас друкованих провідників, що виконується спеціальною програмою – роутером.

Щоб виконати трасування, треба відкрити файл *pcb*, в якому виконано розміщення компонентів на платі. Якщо передбачається використання роутерів *PRO Route* чи *SPECCTRA*, то в шарі *Board* обов'язково мають бути проведені контури плати.

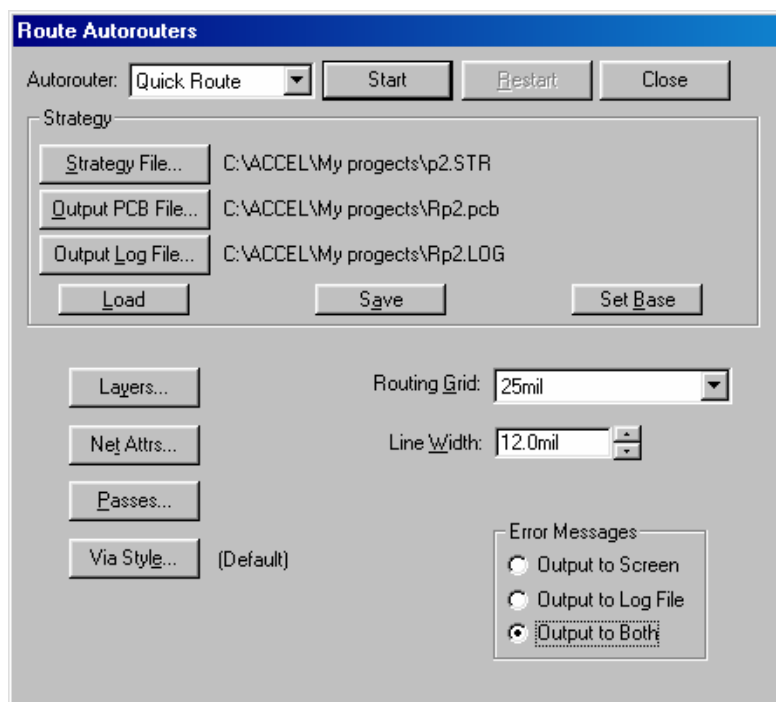


Рис. 12

Якщо плата має певні ділянки, які не повинні пересікатися друкованими провідниками, то контури цих ділянок, що називаються бар'єрами трасування, мають бути нанесені на плату з використанням команди *Place > Keepout* або піктограми . Нанесенню бар'єру передують команди *Options > Current Keepout*, яка визначає його параметри: що являє собою бар'єр: лінію чи полігон, а також чи поширюється даний бар'єр тільки на поточний шар провідників чи на всі шари, як, наприклад, у випадку отвору в платі.

Вікно роутера (рис. 12) відкривається командою *Route Autorouter*. Перш за все в полі

Autorouter треба вибрати тип роутера. Ми радимо обрати *Quick Route* – просту програму, що дозволяє трасувати плати з числом шарів металізації не більше чотирьох. Її можливостей цілком достатньо для виконання студентських робіт.

Нижче у вікні *Route Autorouters* вказуються імена за замовчуванням для файла стратегії трасування, а також вхідного і вихідного файлів.

Файл стратегії описує правила, за якими відбувається трасування. Недосвідченим користувачам ми не радимо вносити змін до стратегії, що використовується за замовчуванням, а якщо це випадково сталося, то відновити стандартні настройки можна, натиснувши кнопку *Set Base*. Ті ж, хто бажають глибше ознайомитись з цими питаннями, можуть звернутися до літератури [1-3].

Вхідним файлом роутера *Quick Route* може бути *pcb*-файл, у якому певна кількість провідників вже прокладена. Тоді роутер, не вносячи змін до вже прокладених провідників, виконує трасування решти. Така особливість цього роутера дозволяє при потребі виконувати трасування поетапно. Цьому сприяє й те, що вихідний файл має той самий формат і те ж саме розширення, що і вхідний – *pcb*. За замовчуванням ім'я вихідного файла формується з імені вхідного шляхом додавання попереду літери *R*. Звісно, роутер не може заново створити вихідний файл, якщо його попередня версія, що має те саме ім'я і розташована в тій самій папці, відкрита редактором *PCB* чи ін-

шою програмою. В такому випадку роутер видає повідомлення з проханням закрити цей файл в інших вікнах.

В нижній частині вікна *Route Autorouters* розташовані кнопки, що застосовуються для внесення змін до параметрів шарів (*Layers*), провідників (*Net Attrs*), проходів трасування (*Passes*) та перехідних отворів (*Via Style*), а також поля, що дозволяють вибирати крок сітки трасування (*Routing Grid*), ширину провідника (*Line Width*) та місце виведення інформації про помилки (*Error Messages*).

Якщо живлення на мікросхеми подається за допомогою друкованих провідників, а не суцільних шарів металізації, то ці провідники виконують товщими за інші, бо через них проходить більший струм. До того ж зменшення опору ліній живлення сприяє зниженню рівня перешкод, що поширюються по них внаслідок зміни струму, що споживається мікросхемами.

Щоб збільшити товщину цих провідників треба перед трасуванням надати їм відповідних атрибутів. Для цього в вікні *Route Autorouters* натисніть кнопку *Net Attrs* або виконайте команду *Edit > Nets*. В графі *Nets* виділіть імена ліній живлення: *GND*, *+5V* тощо. Потім у вікні *Edit > Nets* натисніть кнопку *Edit Attrs*, у вікні *Attributes* – кнопку *Add*, а у графі *Name* вікна *Place Attributes* виберіть *Width* (ширина) та введіть в поле *Value* (значення) бажану ширину цих провідників, – не менше 20 міл.

Автоматичне трасування плати розпочинається натисканням кнопки *Start* у вікні *Route Autorouters*. Хід і результати трасування можна спостерігати у вікні редактора *PCB*, де автоматично закривається вхідний і відкривається вихідний файл.

4.5. Друк файлів PCB

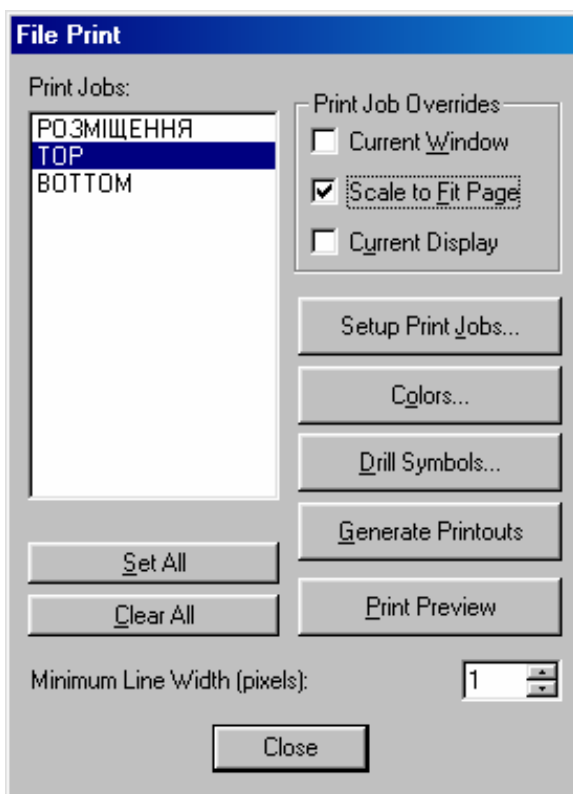


Рис. 13

Як вже відзначалось, друк в P-CAD не зводиться до простого перенесення зображення з екрана монітора на аркуш паперу. Під час друку воно може зазнавати значних трансформацій. Особливо це стосується саме редактора *PCB*, що працює зі складними багатшаровими зображеннями.

При виконанні команди *File > Print* або натисканні відповідної кнопки панелі інструментів на екран виводиться вікно діалогу, показане на рис. 13.

Якщо даний файл раніше не друкувався, то графа *Print Jobs* (завдання для друку) буде пустою. В цьому випадку можна або, поставивши позначку в полі *Current Display*, просто надрукувати всі елементи зображення, що відображається у поточному вікні редактора, або сформувати одне чи декілька завдань *Print Jobs*, в яких детально задаються всі параметри друку.

Для створення завдання треба натиснути кнопку *Setup Print Jobs* і перейти в вікно діалогу, показане на рис. 14. Кнопки *Add*, *Modify*, *Delete* дозволяють відповідно додати

завдання, ім'я якого вказується в полі *Print Job Name*, змінити або вилучити його.

Кожне завдання характеризується списком шарів, зображення яких воно друкуватиме. Щоб виділити чи зняти виділення шару у графі *Layers*, треба клацнути по ньому мишкою при натиснутій клавіші *Ctrl*. Можна також робити це одразу для певних груп шарів, користуючись розташованими нижче кнопками.

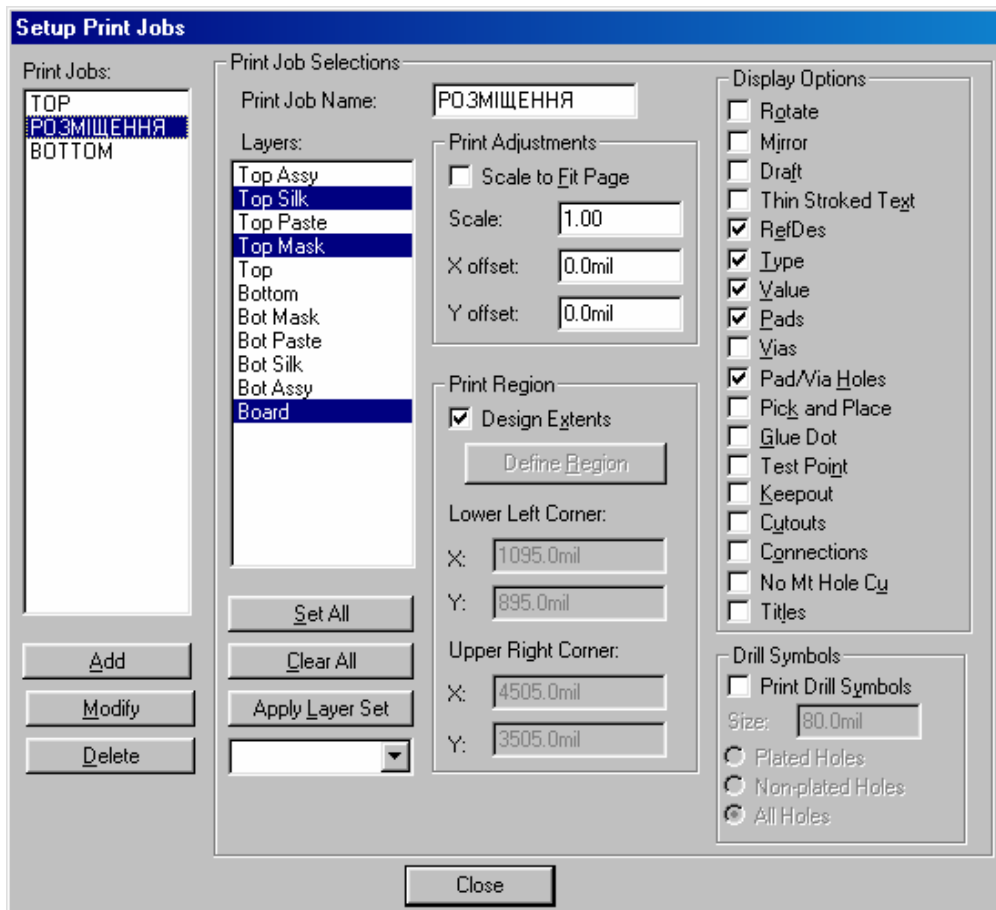


Рис. 14

Праворуч від графи *Layers* розташовані групи полів *Print Adjustment*, що дозволяють встановити масштаб (*Scale*) та зміщення зображення від краю аркуша (*X offset*, *Y offset*), а також група полів *Print Region*, що дозволяють задати координати точок, які обмежують фрагмент зображення, що друкуватиметься. Позначка в полі *Scale to Fit Page* приводить масштаб зображення у відповідність до розмірів аркуша, на якому воно буде надруковане, а позначка в полі *Design Extent* автоматично виділяє для друку тільки заповнену частину аркуша, відсікаючи його порожні ділянки.

Призначення опцій *Display Option* подані в додатку 2. Як правило, при виконанні курсових робіт, студенти розробляють нескладні двобічні плати. В цьому випадку файл *pcb*, одержаний в результаті автоматичного трасування, друкується в вигляді:

- 1) схеми розміщення компонентів на платі;
- 2) вигляду плати з боку встановлення компонентів (шар *Top*);
- 3) вигляду плати з її протилежного боку (шар *Bottom*).

У відповідності до цього, в графі *Print Jobs* треба створити три завдання. Перелік шарів, які в них друкуватимуться, та їх настройки *Display Option* наведені в колонках 1 – 3 таблиць додатків 1 і 2. Завдання для друку файлу зберігаються в його складі.

Після внесення до завдання будь-яких змін, не забувайте натискати кнопку *Modify*. Потім кнопкою *Close* закрийте вікно *Setup Print Jobs* та поверніться до вікна *File Print* (рис. 13). В графі *Print Jobs* цього вікна можна вибрати одне чи декілька завдань для друку. Кнопки *Set All* і *Clear All* встановлюють чи знімають виділення з усіх завдань.

Позначки в полях вікна *File Print* означають:

- *Current Window* – друкувати не весь аркуш, а тільки ту його частину, яку видно в вікні редактора.
- *Scale to Fit Page* – привести масштаб зображення у відповідність до розмірів сторінки, що виводиться на принтер.

Вікно також має кнопку *Colors*, що служить для вибору кольорів, якими друкуватимуться різні елементи зображення в різних шарах, кнопку *Drill Symbols* для вибору графічних символів, якими позначаються отвори різного діаметру, а також поле *Minimum Line Width (pixels)*, що задає мінімальну товщину ліній в пікселях.

Кнопка *Print Preview* дозволяє переглянути на екрані підготовлене для друку зображення, а кнопка *Generate Printouts* видає його безпосередньо на принтер.

5. Робота з бібліотеками компонентів в P-CAD

Перш за все, слід відзначити, що, хоча P-CAD 2001 постачається з великою кількістю бібліотек, що містять описи тисяч компонентів багатьох виробників, завжди існує потреба в їх поповненні новими компонентами. Крім того, деякі особливості використання компонентів в конкретних схемах інколи потребують внесення змін до опису існуючих компонентів у бібліотеках (див. розділ 5.2.4).

5.1. Структура бібліотек компонентів в P-CAD та програми для роботи з ними

Бібліотека P-CAD являє собою файл, що має розширення *lib* і може містити в собі об'єкти трьох типів:

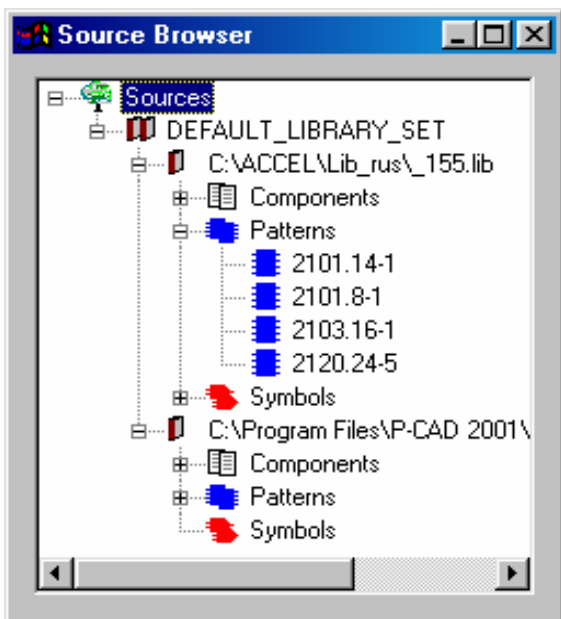


Рис. 15

1. *Symbols* – це умовні графічні позначення компонентів на принциповій схемі;
2. *Patterns (корпуси)* – це графічні зображення корпусів компонентів на платі;
3. *Components* – це текстові описи компонентів, де серед іншого є й посилання на зображення їх символів та корпусів.

Для створення і редагування зображень символів і корпусів застосовуються графічні редактори *Symbol Editor* і *Pattern Editor*, які є спрощеними і пристосованими саме для цієї роботи варіантами розглянутих вище редакторів *Schematic* і *PCB*. Ці редактори можна запускати або безпосередньо з Windows, або з програми *Library Executive*, що застосовується для редагування текстових даних компонента та виконує загальні функції управління бібліотеками.

Як відомо, чимало компонентів, що позначаються на принципових схемах однаковим символом, можуть розміщуватися в різних корпусах. Ще більше існує компонентів, що упаковані в однакові корпуси, хоча вони виконують абсолютно різні функції й позначаються на принципових схемах абсолютно різними символами.

Звісно, для економії пам'яті, в бібліотеках не доцільно повторювати однакові зображення символів і корпусів.





Переглядати зміст бібліотек, зображення символів, корпусів та описи компонентів можна за допомогою вікна *Source Browser* (рис. 15), що зазвичай відкривається при запуску програми *Library Executive*, або при виконанні в ній команди *View > Source Browser*. Набори бібліотек (*Library Set*), що відображаються в цьому вікні, та перелік бібліотек, які входять до кожного набору, встановлюються командою *Library > Setup*.

5.2. Приклад створення нового компонента

Створити повний опис нового компонента можна декількома способами. У розглянутому нижче прикладі, ми спочатку створимо і запишемо в бібліотеку зображення символу – двовходового елемента I-HE та корпусу DIP14, що має 14 штиркових виводів. Потім програмою *Library Executive* створимо і запишемо в бібліотеку й сам компонент – мікросхему K155ЛА3, де в такому корпусі містяться чотири секції двовходових елементів I-HE.

5.2.1. Створення символу компонента в редакторі *Symbol Editor*

Запустимо програму *Symbol Editor* безпосередньо із Windows та виберемо в її статусному рядку крок сітки, що дорівнює відстані між выводами символу новостворюваного компонента. Її треба обирати такою ж, якою вона є в інших символах, бо дуже незручно працювати зі схемою, де різні компоненти мають різні відстані між выводами. Зазвичай відстань між выводами символів на цифрових схемах дорівнює 100 міл.

Малювати зображення символів компонентів можна користуючись звичайними засобами панелі інструментів:  - вивід,  - лінія,  - дуга,  - точка прив'язки⁶ тощо. Проте простіше й швидше це робиться за допомогою команди *Symbol > Symbol Wizard*, що відкриває вікно діалогу, показане на рис. 16. Це вікно можна відкрити також із програми *Library Executive* командою *Symbol > New*.

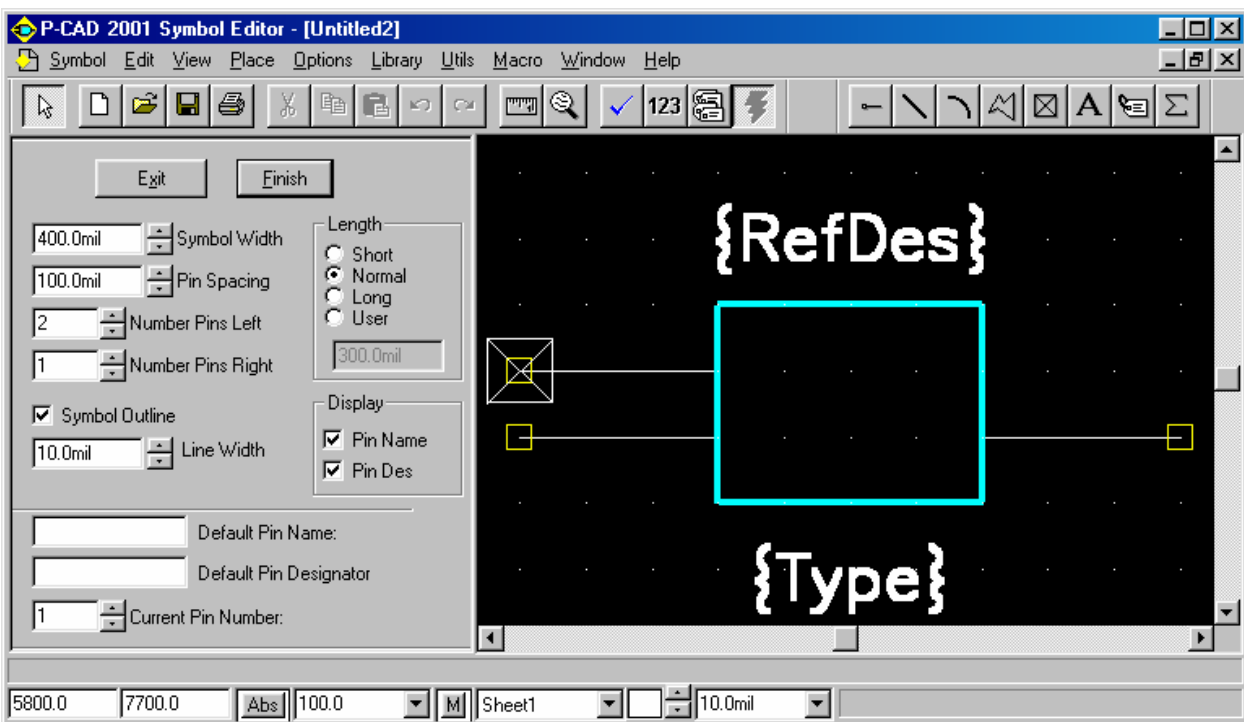


Рис. 16

В цьому вікні задаються такі параметри:

- *Symbol Width* – ширина символу, її також треба обирати кратною кроку сітки;
- *Pin Spacing* – відстань між выводами;
- *Number Pin Left* – кількість выводів зліва;
- *Number Pin Right* – кількість выводів справа;
- *Symbol Outline* – зобразити лінію контуру символу;
- *Line Width* – ширина ліній контуру символу;

⁶ Точка прив'язки позначається на рисунку квадратом з діагоналями. Коли ви розміщуєте символи на схемі чи компоненти на платі, то точка прив'язки розміщується саме там, де ви клацнули мишкою.

- *Length* – довжина виводів (*Short* - короткі, *Normal* - звичайні, *Long* - довгі, *User* – визначені користувачем у розташованому нижче полі);
- *Display* – відображати: *Pin Name* – імена (мітки) виводів, *Pin Des* – позиційні позначення виводів, що зазвичай співпадають з їх номерами;
- *Default Pin Name*, *Default Pin Designator* – поля для вводу імен та позиційних позначень виводів;
- *Current Pin Number* – поточний номер виводу. Виводи нумеруються, починаючи з лівого верхнього, проти стрілки годинника.

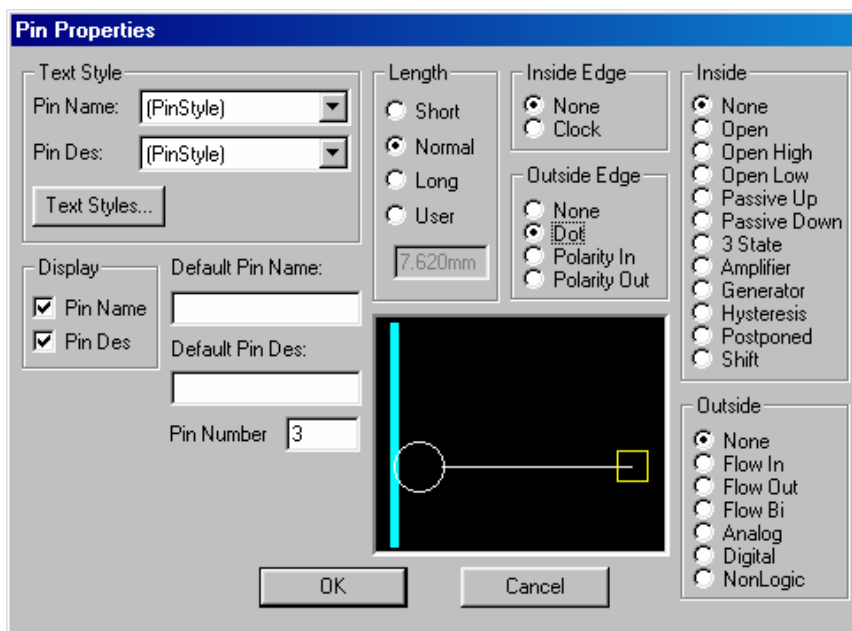


Рис. 17

Можливий варіант заповнення полів для прикладу, що розглядається, показаний на рис. 16. Заповнивши всі поля та натиснувши кнопку *Finish*, ми відкриємо створене зображення символу в звичайному вікні, де виконується решта роботи з його редагування. В нашому прикладі залишається, скориставшись кнопкою **A** - (введення тексту), розмістити на символі знак & та зробити вихід символу інверсним. Для виконання останньої операції треба

виділити цей вивід (можна також виділяти групу виводів), клацнути правою кнопкою мишки і вибрати в контекстному меню опцію *Property*. В результаті відкриється вікно властивостей виводу, яке треба заповнити так, як показано на рис. 17.

Частина полів цього вікна повторює поля попереднього (рис. 16), тому нижче пояснимо призначення лише нових.

Поля *Text Style* задають стилі шрифту для імен і номерів (позначень) виводів.

Inside Edge – тип виводу (позначається всередині):

- *None* – статичний,
- *Clock* – динамічний.

Outside Edge – тип виводу (позначається зовні):

- *None* – прямий,
- *Dot* – інверсний,
- *Polarity In* – вхід з позначкою полярності (логічній одиниці відповідає менш позитивний потенціал),
- *Polarity Out* - вихід з позначкою полярності;

Inside – тип вихідного каскаду:

- *None* – звичайний,
- *Open* – відкритий вивід (загальне позначення),
- *Open High* – відкритий вивід: колектор *PNP*- або емітер *NPN*-транзистора, стік *P* каналу чи витік *N* каналу уніполярного транзистора,
- *Open Low* – відкритий вивід: колектор *NPN*- або емітер *PNP*-транзистора, стік *N* каналу чи витік *P* каналу уніполярного транзистора,
- *Passive Up* – пасивний високий,
- *Passive Down* – пасивний низький,

- *3 State* – тристабільний (з трьома станами),
- *Amplifier* – підсилений,
- *Generator* – генератор,
- *Hysteresis* – з гістерезисною характеристикою,
- *Postponed* – вихід, зміна значення якого не відбувається доти, доки сигнал, що спричинив цю зміну, не повернеться в початковий стан,
- *Shift* – зсув.

Outside – направленість виводу:

- *None* – звичайний (вхід або вихід, в залежності від розташування),
- *Flow In* – вхід,
- *Flow Out* – вихід,
- *Flow Bi* – двонаправлений,
- *Analog* – аналоговий,
- *Digital* – цифровий,
- *NonLogic* – вивід, що не несе логічної інформації.

На завершення треба натиснути кнопку *OK* та записати створений символ в бібліотеку. Для цього треба виділити весь створений символ і виконати команду *Symbol > Save As*. В діалоговому вікні вказують файл бібліотеки, та ім'я символу. Поля *Match Default Pin Designators to Pin Numbers* та *Create Component* в даному випадку треба залишити пустими.

5.2.2. Створення корпусу компонента в редакторі *Pattern Editor*

Перш за все зауважимо, що створювати описи корпусів доводиться доволі рідко, бо, розробляючи нові мікросхеми, конструктори намагаються використовувати стандартні корпуси й створюють нові тільки тоді, коли це справді доцільно. До того ж, до

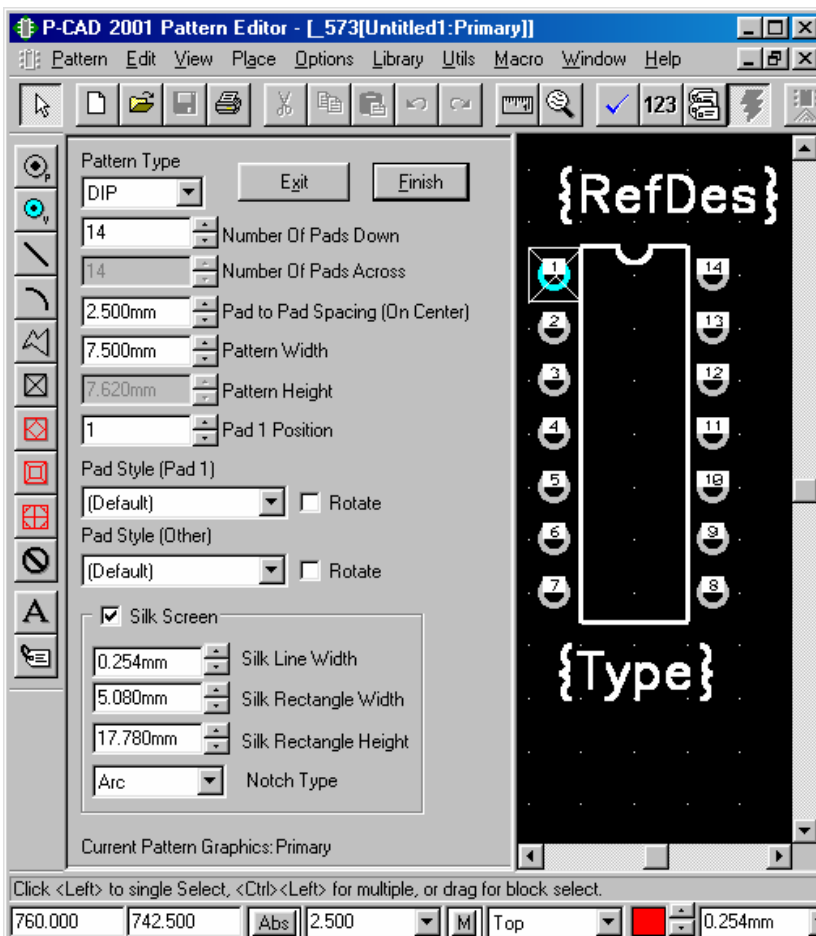


Рис. 18

складу P-CAD 2001 входять багато бібліотек, серед яких і бібліотеки з типовими корпусами мікросхем і роз'ємів:

- *PCBMAIN.LIB* – мікросхеми зі штирковими виводами,
- *PCBSMT.LIB* – мікросхеми з планарними виводами,
- *PCBCONN.LIB* – роз'єми.

В додатку 3 приводиться опис змісту цих бібліотек. Зазвичай вони знаходяться в папці *c:\Program Files\P-CAD 2001\Lib\Old Lib*.

Проілюструємо процес створення опису корпусу на прикладі мікросхеми K155ЛА3, корпус якої відноситься до типу DIP і має 14 виводів. Вони розташовані з двох боків корпусу по 7 виводів в ряду. Відстань

між центрами виводів в ряду – 2,5 мм, відстань між рядами – 7,5 мм.

Запустимо з Windows програму *Pattern Editor*. Виконаємо команду *Options > Configure* та виберемо одиницю виміру – міліметри, а в статусний рядок введемо крок координатної сітки, що дорівнює 2,5 мм.

Зауважимо, що в більшості країн світу крок координатної сітки при виготовленні електронних компонентів і друкованих плат є величиною похідною від дюйма. На сьогодні це стало світовим стандартом, тоді як в країнах колишнього радянського блоку традиційно застосовувалась метрична система. Однак і в СРСР для експорту випускались мікросхеми в дюймовому стандарті. Їх позначення містить перед номером серії літеру *Э* – *експорт*. Виводи в них розташовані з кроком 1,27 і 2,54 мм (0,05 і 0,1 дюйма) замість прийнятих у метричній системі 1,25 і 2,5 мм. Якщо компонент має невелику кількість виводів, то розбіжність у 0,02 - 0,04 мм на вивід не є суттєвою і такий компонент може використовуватись в платах, виготовлених за обома стандартами, чого не скажеш про компоненти з більшою кількістю виводів.

В редакторі *Pattern Editor* корпус компонента можна намалювати, користуючись тільки кнопками панелі інструментів цієї програми, але значно швидше це можна зробити за допомогою команди *Pattern > Pattern Wizard*. Вікно діалогу, що відкривається при цьому, та спосіб його заповнення для даного прикладу показано на рис. 18. Це вікно можна відкрити також із програми *Library Executive* командою *Symbol > New*. Нижче дано пояснення призначень його полів.

- *Pattern Type* – тип корпусу,
- *Number of Pads Down* – число рядків в масиві виводів,
- *Number of Pads Across* – число стовпців в масиві виводів,
- *Pad to Pad Spacing (On Center)* – відстань між центрами виводів,
- *Pattern Width* – ширина корпусу,
- *Pattern Height* – висота корпусу,
- *Pad 1 Position* – положення першого виводу (йому відповідає контактна площадка особливої форми),
- *Pad Stile (Pad 1)* – стиль контактної площадки першого виводу (за замовчуванням – квадрат),
- *Pad Stile (Other)* – стиль решти контактних площадок (за замовчуванням – круг),
- *Rotate* – поворот зображення контактної площадки,
- *Silk Screen* – параметри зображення контурів корпусу:
 - *Silk Line Width* – ширина лінії,
 - *Silk Rectangle Width* – ширина прямокутника габаритів корпусу,
 - *Silk Rectangle Height* – висота прямокутника габаритів корпусу,
 - *Notch Type* – тип графічного відображення ключа – місця, звідки починається нумерація виводів.

Заповнивши всі поля, натисніть кнопку *Finish* та командою *Pattern > Save As* запишіть зображення корпусу в файл потрібної бібліотеки.

5.2.3. Створення опису компонента в програмі *Library Executive*

Запустіть програму *Library Executive*, виконайте в ній команду *Component > New* та виберіть файл бібліотеки, куди буде записано новостворюваний компонент. В результаті ви перейдете до вікна діалогу, показаного на рис. 19.

Перш за все потрібно натиснути кнопку *Select Pattern* та вибрати тип корпусу, в якому знаходиться даний компонент. Потім у поле *Number of Gate* ввести кількість секцій у ньому. Після цього можна буде заповнити наступні поля:

- *Refdes Prefix* – початкові літери позиційного позначення компонента;
- *Component Type* – тип компонента, тут завжди обираємо *Normal* – звичайний;

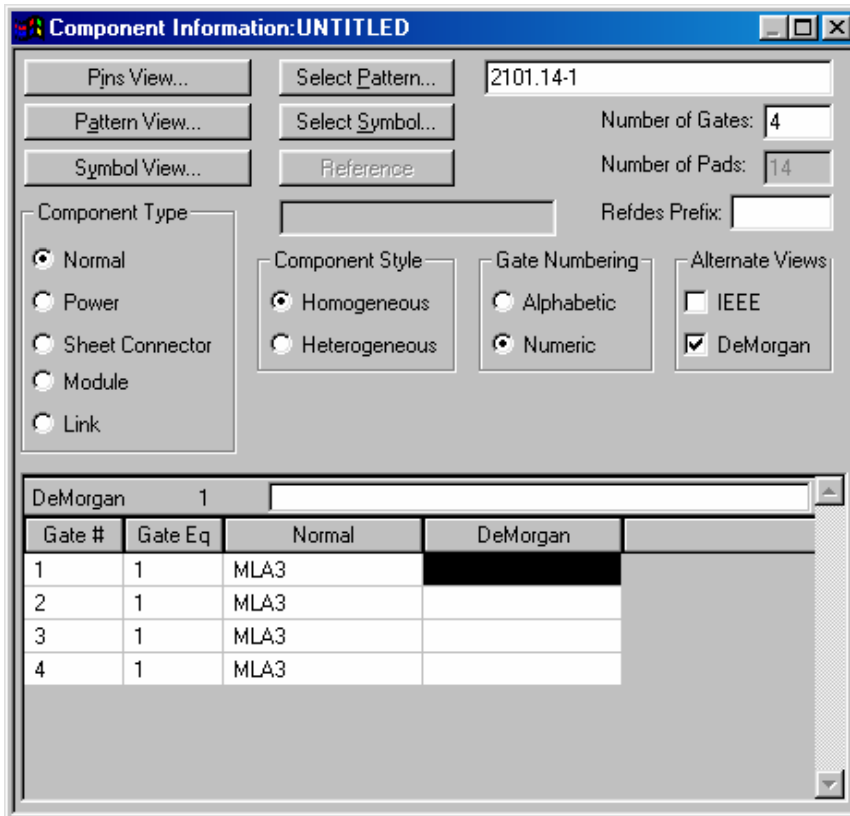


Рис. 19

- *Component Style* – з яких секцій складається компонент: однакових (*Homogeneous*) чи різних (*Heterogeneous*);
- *Gate Numbering* - як позначати секції: літерами А, В, С... (*Alphabetic*) чи цифрами 1, 2, 3... (*Numeric*);
- *Alternate Views* – чи потрібно включати до складу компонента альтернативні позначення його символів згідно зі стандартом *IEEE* та правилом де Моргана (*DeMorgan*). Заповнивши ці поля, встановить курсор в одну із колонок таблиці: *Normal*, *IEEE* чи

DeMorgan і, натиснувши кнопку *Select Symbol*, виберіть відповідне їй графічне зображення символу. Потім треба натиснути кнопку *Pins View* і заповнити таблицю виводів, показану на рис. 20.

В колонці *Pad #* вказується номер, а в колонці *Pin Des* – позиційне позначення ви-

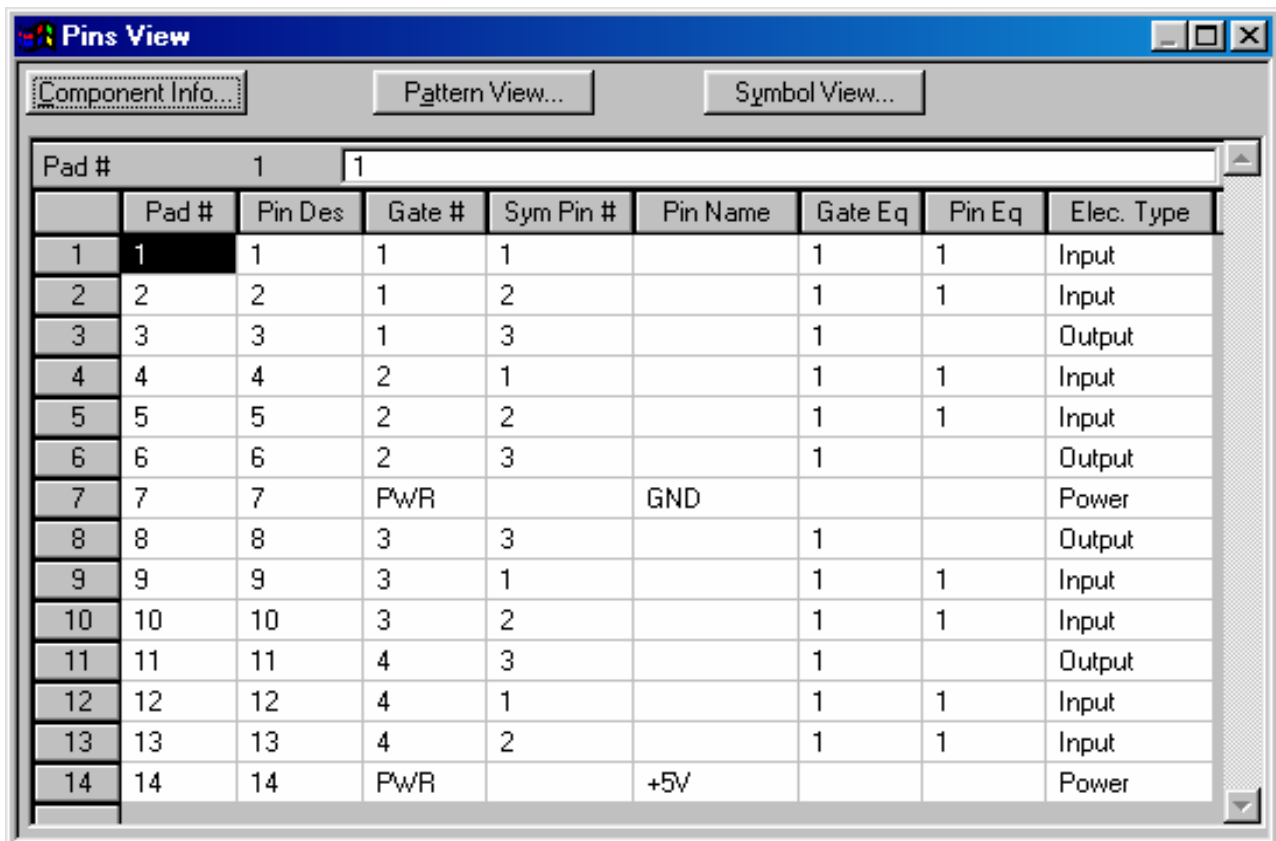


Рис. 20

воду, яке для переважної більшості компонентів співпадає з його номером (в деяких компонентах позначення виводів може складатися, наприклад, з літери, що позначає ряд, і числа, що позначає стовпець).

Колонка *Gate #* показує номер секції, з яким зв'язаний даний вивід.

Колонка *Sym Pin #* задає відповідність між номерами виводів у межах секції і номерами виводів у межах корпусу.

В колонці *Pin Name* вказуються імена⁷ виводів. В даному прикладі імена мають тільки виводи для підключення живлення: *GND* (земля) і *+5V* (5 вольт). Імена виводів живлення треба надавати такі самі, як і в інших компонентах схеми, бо в P-CAD існує правило, що лінії з однаковими іменами вважаються з'єднаними.

В колонках *Gate Eq* і *Pin Eq* вказуються дані про логічну еквівалентність секцій і виводів. Поняття логічної еквівалентності розглядається в наступному розділі.

Остання колонка *Elec.Type* містить дані про електричний тип виводу. Пояснення більшості типів дані в переліку *Inside* на рис. 17. Крім них тут використовуються такі типи як: *Unknown* – невідомий, *Input* – вхід, *Output* – вихід, *Power* – вивід живлення.

Для вводу типу можна або вибрати його зі списку, або ввести першу літеру його назви: *I* – *Input*, *O* – *Output* тощо. Якщо даною літерою починається декілька назв, то її треба натискати доти, доки не встановиться потрібна вам назва типу. Наприклад, для вводу типу *Power* клавішу *P* доведеться натиснути декілька разів.

Створення компонента завершуються командою *Component > Save As* з меню програми *Library Executive*, якою цей компонент записується у бібліотеку. Але перед цим доцільно перевірити компонент на наявність помилок, виконавши команду *Component > Validate*.

5.2.4. Логічна еквівалентність секцій і виводів

Дві чи більше секцій компонента називаються логічно еквівалентними, якщо вони можуть замінити одна одну. Врахування логічної еквівалентності дозволяє оптимізувати прокладку друкованих провідників. P-CAD може замінити секції, показані на принциповій схемі, іншими секціями, логічно еквівалентними даним, якщо така заміна дозволяє спростити трасування (зменшити довжину друкованих провідників чи кількість перехідних отворів тощо). Те саме стосується логічної еквівалентності виводів. Наприклад входи елемента I-HE, розглянутого в попередньому прикладі, є логічно еквівалентними. Отже заміна одного входу іншим також може сприяти спрощенню трасування плати. Нагадаємо, що оптимізація з'єднань на платі за рахунок перестановки логічно еквівалентних секцій і виводів, виконується в редакторі *PCB* командою *Utils > Optimize Nets* після розміщення компонентів, перед трасуванням плати.

В таблиці виводів інформація про логічну еквівалентність секцій і виводів вводиться так. Нулями або пустими клітинками в стовпці *Gate Eq* позначають унікальні секції, які не можна замінити ніякими іншими, а в стовпці *Pin Eq* те саме роблять з виводами. Відмінне від нуля значення в цих стовпцях означає, що дану секцію чи даний вивід можна замінити іншою секцією чи іншим виводом з таким самим значенням логічної еквівалентності. В розглянутому вище прикладі, всі чотири секції логічних елементів I-HE позначені одиницею в полі *Gate Eq* (див. рис. 19 і 20), бо вони можуть замінити одна одну. Навіть, коли компонент складається лише з однієї секції, її виводи треба позначати відмінною від нуля цифрою в полі *Gate Eq*. В межах секції виводи, що можуть замінити один одного, також позначені одиницями в полі *Pin Eq* (рис. 20).

В практиці розробки плат трапляються, наприклад, такі ситуації, коли деякі сигнали необхідно виводити на контакти роз'єму з заздалегідь визначеними номерами, тоді як номери контактів роз'єму для інших сигналів можуть обиратися довільно. Щоб

⁷ У вітчизняному стандарті їх прийнято називати мітками.

скористатися цією свободою вибору для спрощення трасування плати, можна задати для фіксованих контактів роз'єму логічну еквівалентність 0, для решти контактів – 1.

Останній приклад засвідчує, що вміння працювати з бібліотеками P-CAD потрібне не тільки для їх поповнення новими компонентами.

5.3. Інші операції з бібліотеками

Вище було розглянуто створення нового компонента "з нуля" і його запис в існуючу бібліотеку.

Внесення змін до існуючих компонентів, корпусів і символів виконується в програмі *Library Executive* аналогічним чином, тільки замість команд *Component > New*, *Pattern > New* і *Symbol > New* виконуються команди *Component > Open*, *Pattern > Open* і *Symbol > Open*, що відкривають відповідні об'єкти для редагування. Команди *Pattern > Open* і *Symbol > Open* можна також виконати з меню редакторів *Pattern Editor* і *Symbol Editor*.

Результат редагування корпусу чи символу у відповідних графічних редакторах можна записати як в поточну, так і в іншу бібліотеку, виконавши команду *Pattern > Save As* чи *Symbol > Save As*. Однак командою *Component > Save As* результат редагування компонента можна зберегти тільки в поточній бібліотеці.

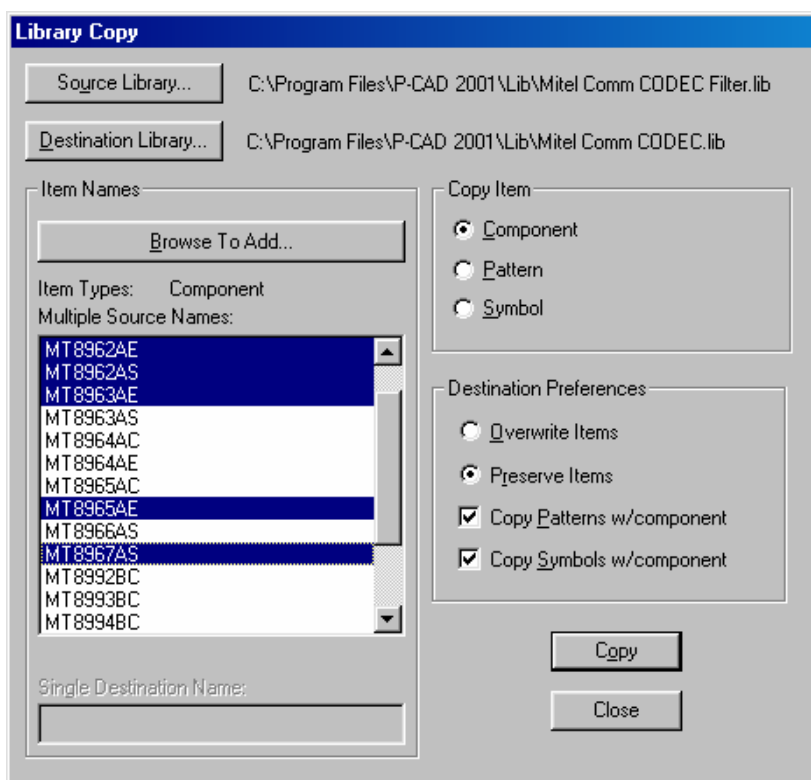


Рис. 21

воли, на які посилаються ці компоненти. Вибираючи список об'єктів для копіювання, можна переглянути зображення їх корпусів та символів за допомогою кнопки *Browse To Add*. Саме копіювання відбувається при натисканні кнопки *Copy*.

При цьому треба мати на увазі, що зміни в складі бібліотек інколи не зразу відображаються у вікні *Source Browse*. В такому випадку треба заново завантажити бібліотеку в вікно *Source Browse* або заново запустити саму програму *Library Executive*.

Звісно, зручність роботи в P-CAD залежить від того наскільки продуманою й зручною є структурна організація його бібліотек. Радимо розпочинати роботу над проектом саме з підбору бібліотек і при потребі створення нових.

Щоб переписати компонент, корпус чи символ з однієї бібліотеки до іншої, потрібно виконати команду *Library > Copy*, в результаті відкриється вікно (рис. 21), де кнопками *Source Library* і *Destination Library* задають з якої бібліотеки і в яку відбудеться копіювання.

В полі *Copy Item* вказують тип об'єктів, що копіюватимуться: *Component*, *Pattern* чи *Symbol*, а в графі *Multiple Source Names* вибирають їх перелік, виділяючи потрібні об'єкти мишкою при натиснутій клавіші *Ctrl*.

Якщо відмітити поля *Copy Patterns w/component* і *Copy Symbol w/component*, то разом з компонентами копіюватимуться й корпуси та сим-

Пуста бібліотека створюється командою *Library > New*. Також нову бібліотеку можна створити шляхом копіювання існуючої з наступним внесенням до неї потрібних змін.

6. Зміст лабораторних робіт

Підготовка до виконання робіт передбачає вивчення відповідних розділів лекційного курсу, системи P-CAD, а також складання протоколу, що містить потрібні схеми, таблиці, план виконання роботи тощо.

Загальною вимогою при цьому є те, що вся підготовча робота, яку можливо виконати напередодні заняття, обов'язково має бути виконаною саме напередодні, а не під час його проведення. Виконанню роботи передуює співбесіда з викладачем, під час якої перевіряється наявність у студентів необхідних знань і належним чином оформленого протоколу. При їх відсутності студенти до роботи не допускаються.

6.1. Завдання на лабораторну роботу № 1. Робота з бібліотеками P-CAD

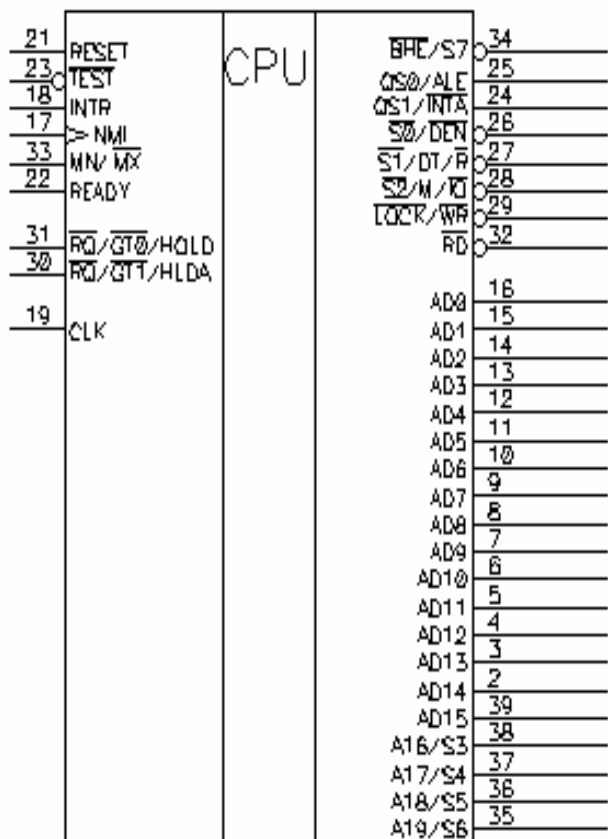
- 1) Вивчити розділи 2.2 та 5.
- 2) Створити нову бібліотеку, а також створити і записати до неї символи, корпуси і компоненти двох мікросхем. Першу мікросхему виберіть з таблиці 1, другу – з рис. 22, де під символом вказано: номер варіанту, назву мікросхеми й в дужках – тип корпусу. Основне поле символів зробіть рівним 400 міл, додаткові поля, в залежності від кількості символів в іменах виводів, – довільними, але кратними 100 міл.

Таблиця 1

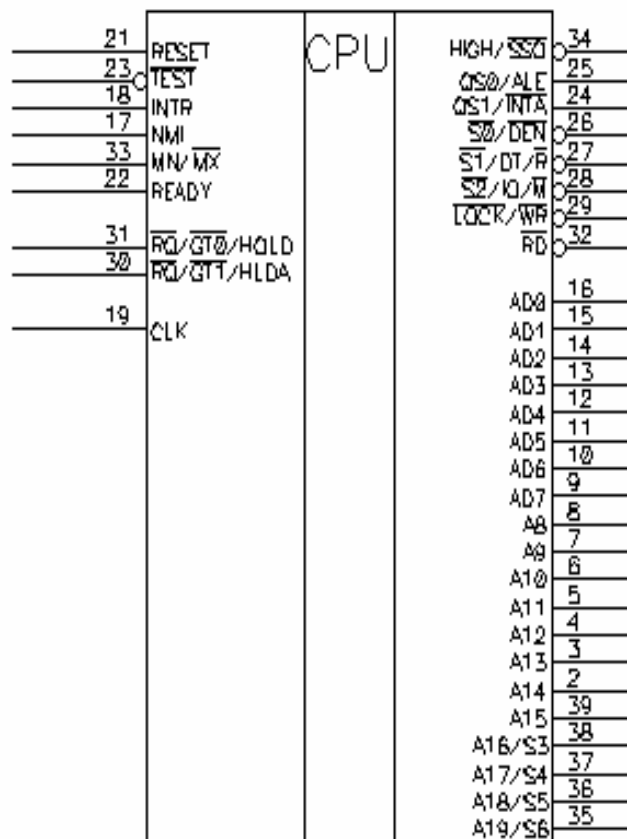
Варі- ант	Мікросхе- ма	Функція	Кількість секцій	Номери входів/виходів	Тип корпусу
1	555ЛА4	3І-НЕ	3	1,2,13/12; 3,4,5/6; 9,10,11/8	DIP14
2	555ЛА7	4І-НЕ з відкритим колектором	2	1,2,4,5/6; 9,10,12,13/8	DIP14
3	555ЛЕ1	2АБО-НЕ	4	1,2/3; 4,5/6; 9,10/8; 12,13/11	DIP14
4	555ЛП5	Нерівнозначність	4	1,2/3; 4,5/6; 9,10/8; 12,13/11	DIP14
5	555ТМ2	D-тригер з динамічним синхровходом С та асинхронними інверсними входами R і S	2	C,D,S,R/Q,~Q ⁸ : 3,2,4,1/5,6; 11,12,10,13/9,8	DIP14
6	555ТВ6	JK-тригер з інверсним динамічним синхровходом С та асинхронним інверсним входом R	2	C,J,K,R/Q,~Q: 12,1,4,13/3,2; 9,8,11,10/5,6	DIP14
7	555ТВ9	JK-тригер з інверсним динамічним синхровходом С та асинхронними інверсними входами R і S	2	C,J,K,S,R/Q,~Q: 1,3,2,4,15/5,6; 13,11,12,10,14/9,7	DIP16

⁸ Тут тильдою (~) позначається знак інверсії, як це прийнято в таблицях виводів символів P-CAD.

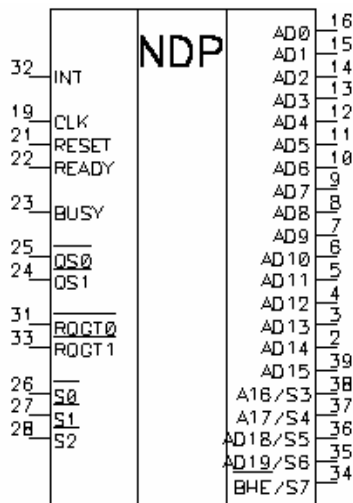
Вариант	Мікросхема	Функція	Кількість секцій	Номери входів/виходів	Тип корпусу
8	155ТВ1	JK-тригер з інверсним динамічним синхровходом С, асинхронними інверсними входами R і S та вхідною логікою 3І: $J=J1*J2*J3$, $K=K1*K2*K3$	1	Q, ~Q/С, J1, J2, J3, K1, K2, K3, S, R: 6, 8/12, 3, 4, 5, 9, 10, 11, 13, 2	DIP14



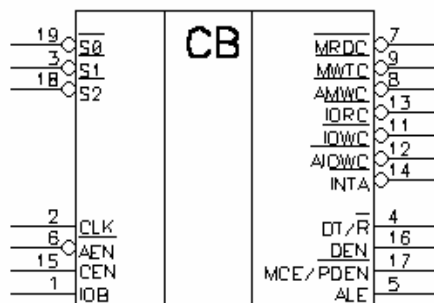
1) K1810BM86 (DIP40)



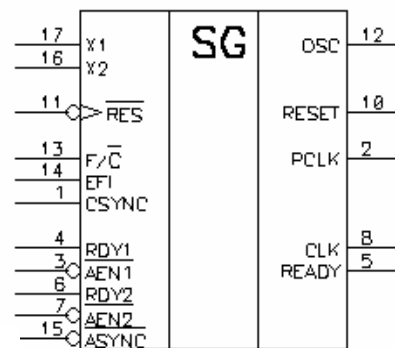
2) K1810BM88 (DIP40)



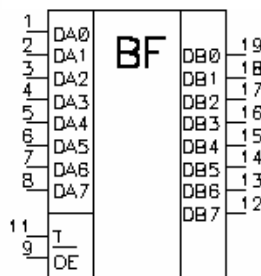
3) K1810BM88 (DIP40)



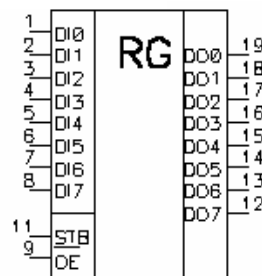
4) K1810BF88 (DIP20)



5) K1810GF84 (DIP18)



6) K1810BA86 (DIP20)



7) K1810IP82 (DIP20)

Рис. 22

Габарити посадочних місць корпусів, що описуються в цій лабораторній роботі, та підключення до них ліній живлення приведені в табл. 2

Таблиця 2

Тип корпусу:	DIP14	DIP16	DIP18	DIP20	DIP40
Кількість виводів	14 (2*7)	16 (2*8)	18 (2*9)	20 (2*10)	40 (2*20)
Відстань між выводами в ряду (міл)	100	100	100	100	100
Відстань між рядами виводів (міл)	300	300	300	300	600
Виводи, до яких підключається лінія живлення +5 В (ім'я лінії – +5V)	14	16	18	20	40
Виводи, до яких підключається лінія живлення "земля" (ім'я лінії – GND)	7	8	9	10	1, 20

6.2. Завдання на лабораторну роботу № 2.

Розробка принципової схеми і друкованої плати процесорного модуля

- 1) Вивчіть розділи 3 і 4.
- 2) Обміняйтесь зі студентами своєї групи файлами, створеними в попередній роботі, та створіть бібліотеку, що містить символи, корпуси і компоненти мікросхем, наведених у табл. 3.
- 3) Побудуйте в редакторі *Schematic* принципову схему процесорного модуля, який включає центральний процесор, синхрогенератор та схеми, що забезпечують формування сигналів шин адреси, даних та управління. Ці шини виведіть на роз'єм і дайте їх провідникам стандартні імена: *A0, A1, A2...* ; *D0, D1, D2...* ; *~MEMRD, ~MEMWR, ~IORD, ~IOWR*. Також на роз'єм виведіть сигнал встановлення початкового стану мікропроцесора *RESET* та лінії живлення: *+5V, GND*. Перелік обов'язкових для використання мікросхем наведених в табл. 3, а приклад схеми їх з'єднання – в додатку 4. Крім них, незалежно від варіанту, в схемі треба встановити необхідні дискретні елементи, рекомендований перелік яких подано в табл. 4.
- 4) Розробіть двобічну друковану плату процесорного модуля.

Таблиця 3

Тип мікросхеми	Варіант	1	2	3	4	5	6	7	8
	Кількість								
Центральний процесор K1810BM88	1		1		1		1		
Центральний процесор K1810BM86		1			1		1		1
Арифметичний співпроцесор K1810BM87						1	1	1	1
Синхрогенератор K1810ГФ84	1	1	1	1	1	1	1	1	1
Регістри-защипки шини адреси K1810IP82	3	3	2	2	3	3	2	2	2
Шинний формувач K1810BA86	1	2	1	2	1	2	1	2	2
Контролер системної шини K1810BF88	1	1	1	1	1	1	1	1	1
Кількість розрядів шини адреси	20	20	16	15	20	20	16	15	

Таблиця 4

Тип та призначення дискретного елемента	Ім'я бібліотеки	Ім'я компонента
Фільтруючі конденсатори, що встановлюються паралельно входам живлення кожної мікросхеми	C.lib	C_100W120
Фільтруючий конденсатор, що встановлюється паралельно входам живлення роз'єму плати	C.lib	C_100W120RP
Резистор, що підключається до лінії +5V для формування сигналу рівня "логічна одиниця"	R.lib	R_100_0250_01_V
Резистор, що підключається до входу RES синхрогенератора	R.lib	R_100_0125_01_V
Конденсатор, що підключається до входу RES синхрогенератора	C.lib	C_100W120RP
Діод, що підключається до входу RES синхрогенератора	VD.lib	VD_100_142_V
Кварцовий резонатор синхрогенератора	Z.lib	Z
Конденсатор біля кварцового резонатора	C.lib	C_100W120
Роз'єм на 20 виводів	X.lib	EDG20F
Роз'єм на 36 виводів	X.lib	EDG36F
Роз'єм на 62 виводи	X.lib	EDG62F

6.3. Завдання на лабораторну роботу № 3.

Розробка принципової схеми і друкованої плати модуля пам'яті

- 1) Розробити і побудувати принципову схему модуля пам'яті згідно з варіантом, наведеним в таблиці 5. Сигнали шини адреси, даних і управління, а також лінії живлення подаються на схему через роз'єм. Всім цим провідникам обов'язково дайте ті самі імена, що і в попередніх лабораторних роботах. Конденсатори фільтрів у лініях живлення рекомендується вибрати з таблиці 4, мікросхеми пам'яті – з бібліотек *_537.lib* і *_573.lib*, а необхідні дешифратори та логічні елементи – з бібліотеки *_555.lib*.

Таблиця 5

Варіант	Розрядність шини даних, біт	Оперативна пам'ять (RAM)		Постійна пам'ять (ROM)	
		Діапазон адрес	Тип мікросхем	Діапазон адрес	Тип мікросхем
1	8	00000-007FF	K537PY13	FE000-FFFFFF	K573PФ2
2	16	00000-01FFF	K537PY8	FF000-FFFFFF	K573PФ1
3	8	0000-07FF	K537PY13	E000-FFFF	K573PФ5
4	16	0000-1FFF	K537PY9	E000-FFFF	K573PФ5
5	8	00000-007FF	K537PY13	E0000-FFFFFF	K573PФ8
6	16	00000-01FFF	K537PY8	FF000-FFFFFF	K573PФ1
7	8	0000-07FF	K537PY13	8000-FFFF	K573PФ4
8	16	0000-1FFF	K537PY9	8000-FFFF	K573PФ4

- 2) Розробити друковану плату модуля з чотирма шарами металізації, причому зовнішні шари *Top* і *Bottom* використати для прокладки друкованих провідників, а лінії живлення +5V і *GND* підвести до компонентів через внутрішні шари суцільної металізації.

6.4. Завдання на лабораторну роботу № 4.

Розробка принципової схеми і друкованої плати модуля портів зовнішніх пристроїв

- 1) Розробити принципову схему модуля портів зовнішніх пристроїв згідно з варіантом, наведеним в таблиці 6. Потрібні модулю сигнали шини адреси, даних і управління та лінії живлення подаються на схему через роз'єм. Всім цим провідникам обов'язково дайте ті самі імена, що і в попередніх лабораторних роботах. Крім того виведіть на роз'єм виводи портів. Зазначені в таблиці 6 мікросхеми можна взяти в бібліотеці *mp.lib*, необхідні дешифратори та логічні елементи – в бібліотеці *_555.lib*, а конденсатори фільтрів у лініях живлення – в бібліотеках, вказаних у таблиці 4.
- 2) Розробити друковану плату модуля з чотирма шарами металізації, причому, два зовнішні шари – *Top* і *Bottom* використати для прокладки друкованих провідників, а лінії живлення +5V і *GND* підвести до компонентів через внутрішні шари суцільної металізації.

Таблиця 6

Тип та призначення мікросхеми	Варіант	1	2	3	4	5	6	7	8
		Кількість							
Паралельний програмований інтерфейс 580BB55		1		1		1		1	1
Послідовний програмований інтерфейс 580BB51			1		1		1	1	2
Регістр 589IP12 як порт вводу		1	1	1					
Регістр 589IP12 як порт виводу					1	1	1		

6.5. Завдання на лабораторну роботу № 5.

Розробка принципової схеми і друкованої плати контролера

- 1) Побудувати принципову схему контролера, що складається з розроблених в попередніх лабораторних роботах процесорного модуля, модулів пам'яті та портів зовнішніх пристроїв, розташувавши всі модулі на одній спільній друкованій платі. Для цього скопіювати схеми попередніх лабораторних робіт у єдиний схемний файл (можна такий, що містить декілька аркушів), та внести до нього відповідні виправлення. Зокрема шини адреси, даних і управління повинні з'єднувати ці модулі не через роз'єми, а безпосередньо друкованими провідниками. На роз'єм вивести тільки лінії живлення, сигнал *RESET* і виводи портів.
- 2) Розробити друковану плату контролера, що має чотири шари металізації, причому всі чотири шари використовуються для прокладки друкованих провідників.

7. Зміст курсової роботи

Типовим завданням для курсової роботи є розробка принципової схеми і друкованої плати мікропроцесорного контролера, призначеного для виконання функцій управління певним об'єктом, та написання на асемблері програм для реалізації цих функцій у даному контролері. Об'єктом управління може виступати, наприклад, автоматизована технологічна лінія, нескладний верстат з числовим програмним управлінням, периферійний пристрій комп'ютера чи будь-який інший об'єкт, що може управлятися мікропроцесором.

Курсова робота виконується в декілька етапів. Для кожного етапу встановлюється об'єм роботи і граничний термін його виконання і здачі. Подібно до авторалі, кожний етап відіграє роль проміжного фінішу, запізнення на який карається штрафними балами, що впливають на поточну атестацію студента й на підсумкову оцінку курсової роботи, яка враховує не тільки якість виконання й захисту "кінцевого продукту", а й оцінки, одержані при здачі окремих етапів роботи.

Здаючи виконаний етап, студент узгоджує з викладачем зміст роботи, що виконуватиметься ним на наступному етапі (за аналогію з авторалі – маршрут до наступного проміжного фінішу). Отже студент, що відвідує консультації нерегулярно, й переходить до наступного етапу без узгодження з викладачем, ризикує "збитися з дороги" й бути вимушеним повертатися на декілька етапів назад та переробляти зроблене.

Результат виконання етапу подається на перевірку тільки в письмовій формі або в формі файлів. Усний звіт про виконані роботи не приймається.

Така організація знімає зі студентів і викладачів пікове навантаження перед початком сесії й заохочує до регулярної роботи протягом всього семестру, що сприяє підвищенню якості роботи та об'єктивності її оцінки.

Типовий набір етапів виконання курсової роботи, в якій розробляється мікропроцесорний контролер, включає:

1. Вибір теми курсової роботи.
2. Загальний опис складу і функціонування об'єкту управління.
3. Опис джерел і споживачів інформації об'єкта управління та форматів портів вводу-виводу контролера.
4. Розробка алгоритмів управління об'єктом.
5. Розробка програм, що реалізують дані алгоритми в контролері.
6. Розробка принципової схеми контролера.
7. Розробка друкованої плати контролера.
8. Оформлення і захист курсової роботи.

Нижче ми коротко розглянемо основний зміст цих етапів та відповідних їм розділів пояснювальної записки, а також типові питання, що виникають при їх виконанні.

7.1. Вибір теми

Студенти заохочуються самостійно обирати тему курсової роботи. Зокрема, якщо в ній розробляється мікропроцесорний контролер, що управляє певним об'єктом, то бажано, щоб кожний студент сам вибрав об'єкт, який йому більш цікавий чи краще відомий. Єдине обмеження полягає в тому, щоб вибрана тема за обсягом роботи відповідала вимогам до курсових робіт і щоб вона не була надто подібною до тем, що вже видані іншим студентам.

7.2. Загальний опис складу і функціонування об'єкту управління

На цьому етапі виконання курсової роботи і в цьому розділі її пояснювальної записки приводяться рисунки і дається опис об'єкта управління: з яких частин він складається і як вони функціонують.

Опис робиться в довільній формі, подібній до тієї, яку ми зустрічаємо, наприклад, в інструкціях до різного роду приладів. Скажімо в інструкції до магнітофону ми знайдемо опис клавішів управління, елементів індикації та способів виконання різних операцій: запису, перемотування стрічки тощо.

Проте, якщо ми розробляємо мікропроцесорний контролер, встановлений у магнітофоні, то крім цього нам треба в загальних рисах описати також склад і функціонування його кінематичної схеми. Наприклад, при натисканні клавіші "Запис" спрацьовує електромагніт, що притискає магнітну стрічку до магнітних голівок і валу, який її рухає з постійною швидкістю, а також включаються магнітні голівки стирання і запису та електродвигун, що намотує стрічку на праву котушку в касеті. Такий опис треба зробити для всіх режимів роботи, включно з "нештатними ситуаціями" такими як закінчення чи обрив стрічки.

Звісно, добре, коли опис, який ви приводите в цьому розділі, точно відповідає реальному об'єкту управління. Однак ми застерігаємо вас від того, щоб розбирати побутові прилади чи надовго занурюватися в читання багатотомної документації, де описані реальні об'єкти. Зрештою, мета даної курсової роботи – навчитися розробляти мікропроцесорні контролери, а не магнітофони чи інші пристрої. Тому не біда, якщо віртуальний об'єкт, який ви опишете в цьому розділі, дещо відрізнятиметься від реального. Це питання треба узгодити з викладачем під час консультацій.

7.3. Опис джерел і споживачів інформації об'єкта та форматів портів вводу-виводу контролера

Робота будь-якого контролера полягає в постійному зчитуванні інформації про поточний стан об'єкта та формуванні управляючих впливів на цей об'єкт згідно з закладеною в пам'ять контролера програмою.

Інформація про стан об'єкта надходить з встановлених на ньому датчиків у порти вводу контролера, а сформовані контролером управляючі впливи видаються через порти виводу контролера на виконавчі механізми об'єкта.

В техніці використовуються дуже багато типів датчиків, принцип дії яких заснований на найрізноманітніших фізичних явищах. Наприклад для вводу інформації про положення деталі на конвеєрі можна застосувати фотоелемент або датчик, що використовує явище зміни індуктивності чи ємності при наближенні до нього деталі.

При виконанні курсової роботи, крім спеціально оговорених викладачем випадків, ми не будемо займатися ні вибором конкретних датчиків, ні їх підключенням до портів контролера. Ми просто вважатимемо, наприклад, що одиниця в даному біті даного порту вводу свідчить про наявність деталі поблизу датчика, нуль – про її відсутність і т.п.

Те саме стосується виконавчих механізмів. За деякими винятками, ми не займатимемось ні їх вибором, ні підключенням до портів контролера. Ми просто вважатимемо, що, коли контролер запише одиницю в даний біт даного порту виводу, то це викличе, наприклад, включення двигуна, індикатора, нагрівача чи відкриття клапану тощо, а запис сюди нуля викличе його виключення чи закриття.

Вказані вище порти вводу і виводу називають відповідно цифровими входами і виходами контролера. Крім них деякі контролери мають також аналогові входи й виходи. Наприклад датчики температури, ваги, тиску і т.п. можуть видавати результати вимірювань у вигляді аналогового сигналу – напруги. Так само деякі виконавчі меха-

нізми управляються аналоговими сигналами. Наприклад зусилля, що розвивається гальмівною системою, яка приводиться в дію електромагнітом, буде тим більшим, чим більша напруга на нього подається.

Звісно, сам мікропроцесор працює лише з цифровою інформацією. Тому інформація, що надходить з аналогових входів і видається на аналогові виходи, має перетворюватись з аналогового вигляду в цифровий і навпаки. Якщо контролер потребує аналогових входів і виходів, то студент повинний за узгодженням з викладачем або

Порти вводу:

Порт T1

7	6	5	4	3	2	1	0
T7	T6	T5	T4	T3	T2	T1	T0

Порт T2

7	6	5	4	3	2	1	0
DT				RY		T9	T8

T9-T0 – входи, зв'язані з виходами АЦП,

RY = 1 – аналого-цифрове перетворення закінчене, 0 – ні.

DT = 1 – деталь знаходиться навпроти інструменту, 0 – ні

Порти виводу:

Порт K1

7	6	5	4	3	2	1	0
D1	D0		ST		N2	N1	N0

N2-N1 – сигнали управління аналоговим комутатором, які задають номер входу, що підключається до АЦП.

ST = 1 – розпочати аналого-цифрове перетворення, 0 – ні.

D1-D0 = 10 – обертати двигун вліво,

= 01 – обертати двигун вправо,

= 00 – зупинити двигун.

Рис. 23

включити аналого-цифрові й цифро-аналогові перетворювачі до складу своєї схеми, або обмежитись тільки портами вводу і виводу, до яких ці перетворювачі будуть підключені вже за межами схеми, що розробляється в даній курсовій роботі.

Отже, на даному етапі виконання курсової роботи і в даному розділі її пояснювальної записки треба вирішити, які датчики та виконавчі механізми повинен мати об'єкт управління для того, щоб контролер міг управляти його роботою так, як це описано в попередньому розділі.

Крім того, тут треба вибрати відповідність між датчиками і бітами портів вводу, виконавчими механізмами і бітами портів виводу, тобто вибрати і описати формати портів: дати імена портам, вибрати імена та розташування бітів приблизно так, як це зроблено в фрагменті, показаному на рис. 23.

7.3.1. Деякі приклади

Нехай проектується верстат з числовим програмним управлінням, призначений для свердління отворів у друкованій платі.

Уявімо, що свердло вже обертається і знаходиться в точці з потрібними координатами. Як управляти свердлінням?

Відповідь, на перший погляд, дуже проста: подати на виконавчий механізм команду опустити свердло, а потім команду його підняти. Але в цій відповіді не зовсім зрозумілий термін "потім". Якщо в програмі ці дві команди йтимуть безпосередньо одна за одною, то, враховуючи швидкість роботи мікропроцесора, між ними не пройде й тисячної долі секунди. За цей час свердло не встигне навіть зрушити з місця.

Перше, що спадає на думку, це встановити між двома командами часову затримку, наприклад, дати мікропроцесорові виконати програму, що містить декілька порожніх циклів в циклі, щоб час її виконання був достатнім для свердління отвору. В багатьох випадках подібне рішення цілком прийнятне, але тут воно має суттєві недолі-

ки, бо на час свердління можуть впливати різні сторонні фактори й ми ризикуємо або не просвердлити отвір до кінця, або спричинити зайві простої верстату.

Кращим рішенням в даному випадку є застосування двох датчиків, які б засвідчували факт знаходження свердла у нижньому положенні (коли отвір вже просвердлений), і у верхньому положенні (коли свердло вже підняте з просвердленого отвору). Тоді команди підйому свердла та його переміщення в наступну точку свердління подаватимуться контролером зразу після того, як спрацює відповідний датчик. Це забезпечить швидку й надійну роботу верстата. Він не зламає свердла, бо рух до наступної точки свердління відбуватиметься тільки тоді, коли свердло вже вийняте з просвердленого отвору. Якщо ж свердло й зламається з інших причин, то верстат відразу зупиниться, бо не спрацює датчик його нижнього положення.

З цього простого прикладу можна зробити важливе узагальнення, що найбільш якісним процес управління буде тоді, коли робота виконавчих механізмів об'єкта управління контролюється за допомогою встановлених на ньому датчиків.

Розглянемо тепер, які датчики і виконавчі механізми потрібні для того, щоб перемістити свердло у точку з потрібними координатами X, Y. Зауважимо, що переміщення об'єктів – це досить типова задача, яка вирішується і в багатьох інших пристроях.

Для вводу координат застосовують датчики лінійних чи кутових переміщень, так звані кодувальники (кодирувальники, encoders). Вони містять поверхню з нанесеними на неї білими й чорними (або прозорими й непрозорими) смугами, яка рухається повз фотоелементи. Кожній координаті відповідає своя комбінація кольорів цих смуг, що зчитується фотоелементами і видається у вигляді двійкового коду. Правда тут використовують не звичайний двійковий код, а код Грея.

Недоліком звичайного двійкового коду є велика похибка зчитування у випадку, коли смуги зупиняться там, де змінюються відразу декілька бітів коду. Наприклад, якщо це станеться між координатами 0111 і 1000 (рис. 24), то з фотоелементів може бути зчитаний який завгодно код, бо кожний з них видаватиме чи то нуль, чи то одиницю.

X	Двійковий код				Код Грея			
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Рис. 24

Код Грея побудований так, що будь-які сусідні кодові комбінації відрізняються лише одним бітом. Отже, якщо смуги зупиняться на межі двох координат, то з фотоелементів завжди буде прочитана тільки одна з них.

Значення бітів числа в коді Грея можна одержати з його двійкового коду за допомогою простого правила, яке називають "цифровим диференціюванням": якщо в двійковому коді цифра даного біту співпадає з сусідньою лівою цифрою, то цей біт в коді Грея має значення 0, а якщо відрізняється, то 1.

Однак на практиці описані вище датчики, що безпосередньо видають код поточної координати, використовуються доволі рідко через складність конструкції і відносно високу ціну. Частіше застосовують кодувальники інкрементного типу.

Якщо напрямок руху не змінюється, то

такий датчик може мати тільки один фотоелемент і одну смужку, де чергуються чорні й білі мітки. Під час руху кількість міток перетворюються фотоелементом на кількість імпульсів, що підраховуються лічильником, з якого й зчитується координата.

0	0	0
1	0	1
2	1	1
3	1	0
0	0	0
1	0	1
2	1	1
3	1	0
0	0	0

Рис. 25

Там, де можлива зміна напрямку руху, застосовують датчик з двома фотоелементами, що видають інформацію в коді Грея (рис. 25). Порівнюючи поточне й попереднє значення, зчитане з такого датчика, легко визначити в який бік змінилася координата й додати одиницю в лічильник чи відняти її. Сигнали такого датчика називаються квадратурними. Як видно з рис. 25, мітки обох фотоелементів мають однакову ширину й зміщені одна відносно одної на чверть періоду. Це дозволяє замість двох смужок з мітками використовувати тільки одну, а на чверть періоду змістити один фотоелемент відносно іншого, як це зроблено, наприклад, в звичайній комп'ютерній мишці.

Кодувальники інкрементного типу набагато простіші, але їх використання потребує ініціалізації, тобто попереднього встановлення в точку з нульовими координатами, де лічильник обнуляють. До речі, сам лічильник може бути реалізований, як апаратно – на окремій мікросхемі, так і програмно – мікропроцесором. Отже, разом з кодувальниками інкрементного типу треба застосовувати датчики початкового положення – так звані кінцеві вимикачі. Наприклад, після включення матричного чи струменевого принтера їх робота розпочинається з встановлення голівки в крайнє ліве положення, де спрацьовує саме такий датчик.

Однак, що стосується принтера та інших механізмів, в тому числі й свердлильного верстату, який ми тут розглядаємо, то в них може й не виникнути потреби в застосуванні розглянутих вище датчиків для зчитування координат, якщо лінійні переміщення реалізуються в них двигунами спеціальних конструкцій, такими, як крокові двигуни. Наприклад, у принтері, подача на такий двигун одного імпульсу викликає переміщення друкуючої голівки на один піксел.

При застосуванні звичайного двигуна, треба слідкувати за датчиком координат, можливо зменшити швидкість руху при наближенні до потрібної координати й остаточно зупинити двигун при її досягненні.

Розглянемо ще одну задачу, з якою часто доводиться мати справу при виконанні даної курсової роботи, а саме організацію зчитування даних з клавіатури.

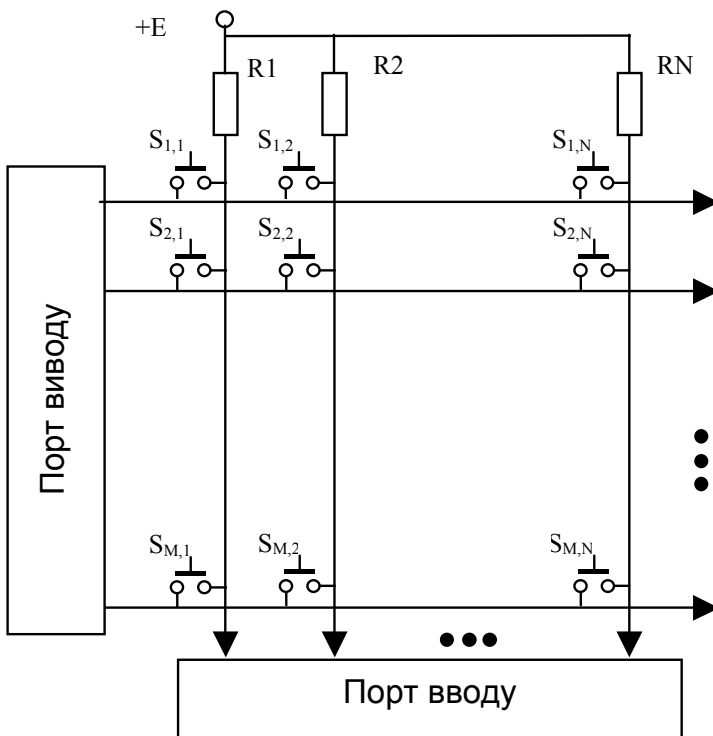


Рис.26

Якщо клавішів небагато, то можна просто підключити кожен з них до певного біту порту вводу, значення якого відповідатиме станіві клавіші.

Для великої кількості клавішів більш раціональним є їх об'єднання у двомірну матрицю (рис. 26), де кожна клавіша замикає контакт між горизонтальною і вертикальною шинами. Опитування стану клавішів такої клавіатури відбувається наступним чином. Спочатку перевіряють, чи є натиснуті клавіші в ряду підключених до першої горизонтальної шини. Для цього в дану горизонтальну шину подають сигнал рівня 0, а в решту горизонтальних шин – сигнал рівня 1. Після цього зчитують дані з вертикальних шин. Нулі будуть там, де клавіші натиснуті,

одиниці – де ні. Потім так само опитують стан клавішів, підключених до наступної горизонтальної шини, і т.д. Отже, для зчитування інформації з $M \cdot N$ клавішів, достатньо підключити M бітів портів виводу до горизонтальних шин, і N бітів портів вводу – до вертикальних шин. Цю кількість бітів можна значно зменшити, дещо вдосконаливши схему клавіатури. Наприклад, якщо сигнали в горизонтальні шини подавати з дешифратора, вхід якого підключений до порту виводу, то в ньому замість M бітів достатньо мати лише $\log_2 M$. Якщо не передбачається одночасного натискання декількох клавішів, підключених до однієї горизонтальної шини, то, підключивши входи шифратора до вертикальних шин, а виходи – до порту виводу, можна й кількість бітів цього порту зменшити з N до $\log_2 N$. В такому випадку мікропроцесор видаватиме і зчитуватиме не просторові, а двійкові коди горизонтальної та вертикальної шини.

Важливою проблемою, яку треба вирішувати при роботі з елементами, що містять механічні контакти: клавішами, кнопками, різними вимикачами тощо є явище *дрижання контактів* (*дребезг контактов, chatter of contacts*). Справа в тому, що якими б якісними не були механічні контакти, але внаслідок їх пружності й неідеальності мікрорельєфу поверхонь, що контактують, замикання чи розмикання контакту ніколи не відбувається одномоментно. Їм завжди властиве дрижання, коли електричний контакт на якийсь момент встановлюється, потім розривається, знову встановлюється, і так декілька разів, поки контакт не буде встановлено чи розірвано остаточно.

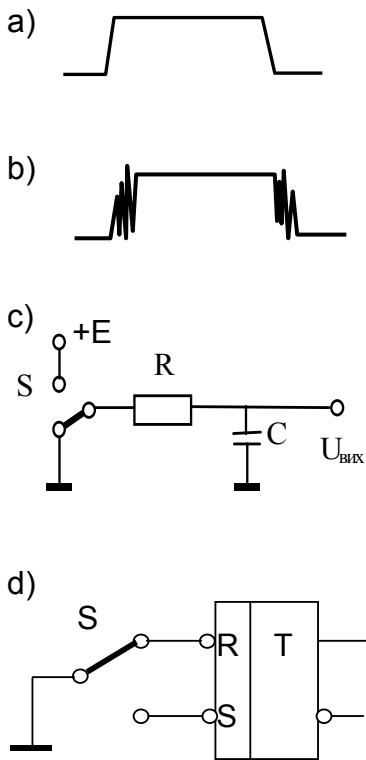


Рис. 27

якщо натиснути, а потім відпустити клавішу, то замість одного імпульсу (рис. 27, а) завжди формуються дві пачки імпульсів (рис. 27, б). Мікропроцесор працює дуже швидко. Тому, якщо не вжити належних заходів, то це може помилково сприйматися ним як багаторазове натискання клавіші.

Дрижання контактів можна усувати схемним шляхом, використовуючи, наприклад, інтегруючий RC-ланцюжок (рис. 27, с) чи асинхронний RS-тригер (рис. 27, д).

Більш дешевими є програмні вирішення, що базуються на штучному створенні часових затримок. Мікропроцесор протягом декількох сотих секунди (часу, достатнього для завершення дрижання контактів) може виконувати будь-яку програму, наприклад, декілька порожніх циклів в циклі.

Існують спеціальні мікросхеми, призначені саме для зчитування даних з клавіатури. Вони самі виробляють сигнали, що подаються в горизонтальні шини й опитують стан вертикальних шин. При виявленні натиснутої клавіші, така мікросхема виробляє сигнал переривання для мікропроцесора і видає йому готовий код натиснутої клавіші. Деякі з таких мікросхем, наприклад КР580ВВ79 [5], поєднують це з функцією управління цифровим індикатором.

В клавіатурах сучасних персональних комп'ютерів застосовують прості 8-розрядні мікропроцесори, наприклад Intel 8048, що програмно усувають дрижання контактів і передають в системний блок відповідну інформацію (так званий скан-код) не тільки коли ви натискаєте клавішу, а й коли ви відпускаєте її [10].

7.4. Розробка алгоритмів управління об'єктом

Якщо на двох попередніх етапах ви описали процеси управління об'єктом і дали імена бітам, зв'язаним з датчиками і виконавчими механізмами, то даний етап зводиться до формального опису алгоритмів, які реалізують ці процеси. Для складних

об'єктів, за узгодженням з викладачем, алгоритми, а потім і програми, можна розробляти не для всіх функцій управління, що реалізуються контролером. Але, в такому разі, в принциповій схемі контролера треба обов'язково передбачити додатковий об'єм пам'яті, куди ці програми можна буде згодом записати.

Взагалі контролер може реалізувати багато різних алгоритмів. Наприклад при включенні контролера запускається фоновий алгоритм. Перш за все він встановлює початковий стан портів виводу, змінних пам'яті, показника стека *SP* тощо. Потім він може виконувати функції управління об'єктом. Цікавою особливістю цього алгоритму є те, що він має початок, але не має кінця. Дійсно управління об'єктом може тривати як завгодно довго і припинятися хіба що шляхом виключення живлення контролера.

Крім фонового алгоритму, контролер може реалізувати алгоритми обробки різного роду переривань. Це найкращий спосіб організації швидкого реагування на події, які можуть наставати в довільний момент часу незалежно одна від одної. Настання кожної такої події викликає відповідне апаратне переривання, в алгоритм обробки якого закладена реакція контролера на цю подію. Наприклад, якщо розробляється контролер, що виконує команди, які в будь-який момент можуть завантажуватися в його пам'ять іншим пристроєм, то це доцільно реалізувати так: фоновий алгоритм зчитує з пам'яті команди і виконує їх, а алгоритм обробки переривання, викликаного надходженням в контролер нового байта команди, просто дозаписує прийнятий байт команди у відповідну область пам'яті.

Більш складними є випадки, коли один мікропроцесор вимушений одночасно управляти декількома процесами, що протікають паралельно в часі. Один з варіантів вирішення такої задачі полягає в написанні окремих алгоритмів для кожного процесу, а також алгоритму обробки апаратного переривання, що викликається через короткі проміжки часу й переключає мікропроцесор з виконання одного алгоритму на інший. Переключення може відбуватись шляхом збереження в пам'яті всіх регістрів мікропроцесора (включно з лічильником адреси команди) й завантаження в них раніше збережених регістрів іншого алгоритму. В результаті виконання цього алгоритму відновлюється з того місця, де воно було зупинене минулим перериванням.

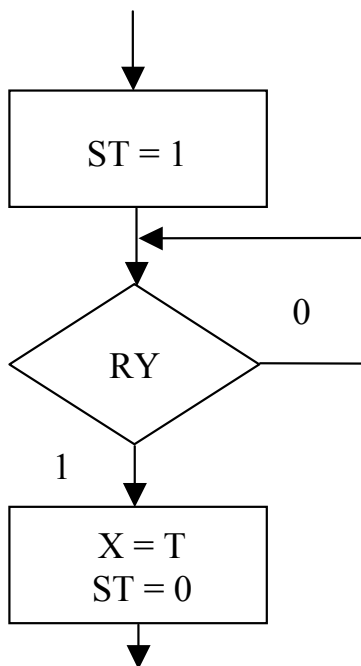


Рис. 28

Універсальнішим, проте складнішим підходом до вирішення цієї задачі є реалізація паралельних процесів так, як це робиться в мові VHDL. Там кожний процес має свій список чутливості, причому зміна значення хоча б одного з елементів цього списку викликає виконання алгоритму, що описує даний процес. В результаті можуть змінюватися значення, що входять до списків чутливості інших процесів і т.д. Такий підхід забезпечує переключення мікропроцесора на виконання алгоритму іншого процесу тільки тоді, коли в цьому справді є потреба, тоді як розглянутий вище метод почергового виділення кожному процесові фіксованого кванту часу роботи мікропроцесора не так ефективно використовує його ресурс.

На рис. 28 наводиться приклад фрагменту алгоритму, що запускає роботу аналого-цифрового перетворювача (АЦП) і, дочекавшись закінчення перетворення, зчитує його результат у змінну пам'яті *X* та повертає в початковий стан сигнал запуску АЦП. Позначення бітів, використаних в цьому алгоритмі, приведені на рис. 23.

7.5. Розробка програм управління об'єктом

Якщо ви маєте алгоритми управління об'єктом, то розробка програм фактично зводиться до опису цих алгоритмів на асемблері вибраного мікропроцесора. Потім з лістингу компіляції програм одержують адреси і код, який за допомогою програматора записують в постійну пам'ять (ROM) контролера.

Мікропроцесори, спроектовані спеціально для використання в подібних контролерах, мають в своїй системі команд спеціальні бітові операції, якими зручно користуватися для зміни значень окремих бітів в портах, регістрах чи пам'яті, а також виконання умовних переходів в залежності від значень певних бітів.

Однак значна частина студентів віддають перевагу використанню більш звичного для них асемблера сучасних персональних комп'ютерів. Тому вони вибирають для свого контролера Intel 8088 чи Intel 8086 – найпростіші мікропроцесори, для якого можна програмувати на цьому асемблері. Саме для них ми наводимо нижче деякі рекомендації та приклад написання програми.

Перш за все слід зазначити, що перша команда, яку виконує ці та інші мікропроцесори даного ряду після їх включення або подачі сигналу *RESET*, має адресу, що складається з усіх одиниць і завершується чотирма нулями. В персональних комп'ютерах ця адреса належить постійній пам'яті, де записана програма початкового завантаження комп'ютера. В контролері, що розробляється в даній курсовій роботі, за цією адресою треба записати команду безумовного переходу на початок основної програми приблизно так, як це зроблено на рис. 29.

Зазвичай псевдокоманди *ORG* задають адресу розташованих за нею команд відносно початку сегмента, але в даному випадку це будуть абсолютні адреси пам'яті вашого контролера. Саме тому програму й дані треба розмістити в одному спільному сегменті, де перша *ORG* задає адресу, з якої розпочинається адресний простір постійної пам'яті (ROM), а друга *ORG* – адресу першої команди, яку мікропроцесор виконає після включення. Адресу *0FFF0H* тут можна писати завжди, навіть тоді, коли контролер використовує не всю шину адреси мікропроцесора. В цьому випадку старші біти адреси нікуди не підключаються, отже ігноруються.

Розглянемо детальніше показаний на рис. 29 приклад програмування алгоритму, зображеного на рис. 28, призначення й розташування бітів якого подане на рис. 23.

Псевдокоманди *EQU* задають відповідність між символічними *T1*, *T2*, *K1* та справжніми *0*, *1*, *2* адресами портів. При написанні програм завжди зручніше користуватись символічними адресами. Справжні адреси стають достеменно відомими після розробки принципової схеми. Тоді їх і вказують в операторах *EQU*

Те, що адреси портів *T1* і *T2* розташовані одна за одною, дозволяє однією командою *IN AX, T1* завантажувати порт *T1* в регістр *AL*, а порт *T2* – в регістр *AH*.

Псевдокоманди *DW* і *DB* виділяють в оперативній пам'яті місце для слова, де зберігатиметься значення температури (біти *T9-T0*) і байту, де зберігатиметься копія порту виводу *K1*

Необхідність збереження копій портів виводу пояснюється тим, що в цьому мікропроцесорі немає команд, що змінюють окремі біти порту. Це доводиться робити на акумуляторі, а потім переписувати результат з акумулятора в порт. Щоб встановити потрібні біти в нуль використовують команду порозрядної кон'юнкції (AND) між поточним значенням порту й константою, що містить в цих бітах нулі. Так само для встановлення потрібних бітів в одиницю використовують команду порозрядної диз'юнкції (OR) між поточним значенням порту й константою, що містить в цих бітах одиниці. Але звідки взяти поточне значення порту? Прочитати його з самого порту виводу мікропроцесор не може, як не може він і записати дані в порт вводу. Тому й доводиться щоразу при виводі даних в порт зберігати їх копію в пам'яті чи регістрі мікропроцесора, звідки, при потребі, їх можна буде прочитати.

. Запис одиниці, а пізніше нуля в біт *ST* відбувається саме за описаною вище схемою: на акумулятор завантажується копія порту *K1*, де її четвертий біт встановлюється в 1 чи в 0, а результат записується в порт *K1* та в його копію в пам'яті.

Якщо при виконанні цих команд виникне переривання, яке змінить значення інших бітів порту *K1*, наприклад тих, що управляють роботою двигуна, то ці команди можуть відновити попередні значення вказаних бітів, що приведе до помилки. Щоб цього не сталося, перед початком даного фрагменту програми виконується команда *CLI*, яка забороняє апаратні переривання, а після його завершення – команда *STI*, яка знову дозволяє їх.

Для перевірки значення біту *RY*, порт, в якому він знаходиться, записується в акумулятор, де командою *TEST AL,00001000B* обнуляються всі біти, крім *RY*. Якщо *RY=0*, то й весь байт результату дорівнюватиме нулю, отже команда умовного переходу *JZ M1* забезпечить повернення до мітки *M1*.

```

T1      EQU 0
T2      EQU 1
K1      EQU 2
        . . .

X       DW ?
COPY_K1 DB ?           ; Копія порту K1 в RAM.
        . . .

START:  ORG ????       ; ???? - початкова адреса ROM.
        . . .         ; Ініціалізація портів, регістрів і
                        ; змінних пам'яті.

        CLI
        MOV AL,COPY_K1
        OR AL,00010000B ; Запис 1 в 4-й біт (ST=1).
        OUT K1,AL
        MOV COPY_K1,AL
        STI

M1:     IN AL,T2
        TEST AL,00001000B ; Перевіряємо 3-й біт T2 (RY=0 ?).
        JZ M1

        IN AX,T1        ; AL := T1,   AH := T2.
        AND AH,00000011B ; обнуляємо старші біти з T2.
        MOV X,AX

        CLI
        MOV AL,COPY_K1
        AND AL,11101111B ; Запис 0 в 4-й біт (ST=0).
        OUT K1,AL
        MOV COPY_K1,AL
        STI
        . . .
        ORG 0FFFF0H
        JMP START      ;Перша команда, що виконується
                        ;після включення мікропроцесора.

```

Рис. 29

Якщо ж $RY=1$, то виконуватимуться наступні команди, які записують зміст портів $T1$, $T2$ в регістри AL і AH , обнуляють шість старших бітів AH , які не мають відношення до температури, та записують результат її виміру в пам'ять за адресою X . Далі описаним вище способом обнуляється біт ST .

На завершення відзначимо, що коли порт $K1$ знаходиться в мікросхемі паралельного програмованого інтерфейсу $K580BB55$ (Intel 8255), то для встановлення біту ST в 0 і 1 можна скористатися спеціальною командою цієї мікросхеми. Для виконання цієї команди треба записати байт $1000n_3n_2n_1d$ в порт, двійкова адреса якого закінчується на 11 . В результаті біт з двійковим номером $n_3n_2n_1$ в порті, адреса якого закінчується на 10 , набуде значення d . В нашому випадку біт ST – це 4-й біт 2-го порту. Отже, щоб встановити його в одиницю (або в нуль), замість шести команд, приведенних в програмі на рис. 29, можна використати тільки дві:

```
MOV AL,10001001B      ; (або MOV AL,10001000B)
OUT 3,AL
```

При цьому нема потреби ані зберігати в пам'яті копію порту $K1$, ані забороняти, а потім знову дозволяти апаратні переривання. Ті ж, хто "відважаться" опанувати асемблер ще одного мікропроцесора, наприклад Intel 8051 (MCS-51, KM1816BE51), зможуть зробити те саме однією командою:

```
SETB P2.4             ; (або CLR P2.4).
```

7.6. Розробка принципової схеми контролера

Всі відправні дані для розробки принципової схеми контролера одержують в результаті виконання попередніх етапів роботи. Так необхідна кількість портів вводу і виводу з'ясовується ще під час опису джерел і споживачів інформації об'єкта. Вибір асемблера для написання програм значною мірою обумовлює вибір мікропроцесора, а з лістингу компіляції цієї програми стає відомим необхідний об'єм постійної пам'яті (ROM) для збереження самої програми й, можливо, потрібних для неї констант, а також об'єм оперативної пам'яті (RAM), де зберігатимуться поточні змінні програми, її стек тощо.

Ті, хто має невелику кількість змінних, інколи намагаються взагалі відмовитись від використання RAM і зберігати всі змінні в регістрах самого мікропроцесора. В принципі, це можливо, але не слід забувати, що в RAM розташований також і стек. Тому відмова від використання RAM означає й відмову від використання в програмі всіх команд, що працюють зі стеком, зокрема підпрограм і переривань.

Всі зовнішні зв'язки принципової схеми контролера: виводи його портів, ліній живлення, сигналу *RESET* тощо треба вивести на роз'єм.

Виводи портів на принциповій схемі обов'язково повинні мати те саме розташування й ті самі імена, які були їм дані під час опису портів (рис. 23) і використовувались в алгоритмах (рис. 28).

Перелік компонентів, що використовуються в принциповій схемі, узгоджується з викладачем.

7.7. Розробка друкованої плати контролера

Як правило, в курсових роботах розробляється двобічна друкована плата контролера. Остаточо це питання узгоджується з викладачем. Способи розробки друкованих плат в системі P-CAD описані вище.

7.8. Оформлення і захист курсової роботи

Пояснювальна записка курсової роботи виконується в текстовому процесорі Word і друкується на папері формату A4. Зразок виконання титульного аркуша приведений в додатку 5. На другій сторінці записки приводиться її зміст, який автоматично ство-

рюється програмою Word при виконанні команди *Вставка > Оглавление и указатели > Оглавление*. Для цього треба застосувати стиль *Заголовок 1* для заголовків всіх розділів пояснювальною записки, приблизний перелік яких приведено нижче.

1. Загальний опис складу і функціонування об'єкту управління.
2. Опис джерел і споживачів інформації об'єкта управління та форматів портів вводу-виводу контролера.
3. Розробка алгоритмів управління об'єктом.
4. Розробка програм управління об'єктом.
5. Розробка принципової схеми контролера.
6. Розробка друкованої плати контролера.

В кінці пояснювальної записки приводяться:

1. відкомпільований лістинг програми,
2. принципова схема контролера,
3. схема розміщення компонентів на платі,
4. зовнішній вигляд плати (кожний шар друкованих провідників окремо).

Крім надрукованої пояснювальної записки до захисту подається її файл та всі інші робочі файли виконаної роботи.

Під час захисту студент повинен продемонструвати знання з усіх питань, що стосуються змісту виконаної курсової роботи, зокрема розуміння роботи схеми, програми, прийомів роботи в системі P-CAD тощо.

Оцінка за курсову роботу враховує якість її виконання, рівень знань, продемонстрованих під час захисту, а також оцінки, одержані студентом під час здачі окремих етапів роботи.

8. Література

1. Саврушев Э.Ц. P-CAD для Windows. Система проектирования печатных плат. Практик. пособие. М.: ЭКОМ, 2002. – 320 с.
2. Разевиг В. Д. Проектирование печатных плат в P-CAD 2001. – М.: Солон-Р, 2001. - 560 с.
3. Уваров А. P-CAD 2000, ACCEL EDA. Конструирование печатных плат. Учебный курс. – М.: ДМК Пресс, 2001. – 320 с.
4. Микропроцессорный комплект K1810: Структура, программирование, применение: Справочная книга / Под ред. Ю.М.Казаринова. - М.: Высш.шк., 1990. - 296 с.
5. Хвощ С.Т. и др. Микропроцессоры и микроЭВМ в системах автоматического управления: Справочник - Л.: Машиностроение. Ленингр. отд., 1987. - 640 с.
6. Лебедев О.Н. Микросхемы памяти и их применение. - М.: Радио и связь, 1990 - 160 с.
7. Аналоговые и цифровые интегральные микросхемы. Справочное пособие /Под ред. С.В.Якубовского. - 2-е изд. перераб. и доп. - М.: Радио и связь, 1985. - 432 с.
8. Применение интегральных микросхем в электронной вычислительной технике.: Справочник / Р.В.Данилов и др.; Под ред. Б.Н.Файзулаева и Б.В.Тарабрина. - М.: Радио и связь, 1986. - 384 с.
9. Корнейчук В.И., Тарасенко В.П., Мишинский Ю.Н. Вычислительные устройства на микросхемах: Справочник - К.: Техніка, 1986. - 264 с.
10. Standart IBM PC. Справочник. Устройство, установка, техническое обслуживание и ремонт персональных компьютеров / Составитель Г. Карпов. – Кишинев, ВИРТ, 1991. – 192 с.

9. Интернет

11. Застосування програми SPECCTRA:
<http://www.electrade.ru/index.sema?a=pages&id=104>
12. Офіційний сайт розробника програми P-CAD 2001
<http://www.p-cad.com>
13. Сайти користувачів P-CAD (форуми, бібліотеки, утиліти, документація, FAQ тощо)
<http://www.rizne.by.ru/accel.htm>
<http://www.electrade.ru/index.sema?a=pages&id=118>
http://www.rodnik.ru/htmls/f_1_3_2_index.htm
http://www.triton.ru/site/sapr_prog_obespech.shtml
<http://avdey.nm.ru/doc/doc09.htm>
<http://www.p-cad.ru/>

Додаток 1

Елементи та шари зображень

Таблиця 7. Елементи зображення

<i>Ім'я елемента зображення</i>	<i>Призначення елемента зображення</i>
Via	Перехідні отвори
Pad	Виводи компонентів
Line	Провідники та лінії
Poly	Полігони
Text	Текст

Таблиця 8. Шари зображення

<i>Ім'я шару</i>	<i>Призначення шару</i>	1	2	3
Top	Верхня сторона плати (бік встановлення компонентів)		+	
Bottom	Нижня сторона плати			+
Board	Контур плати	+	+	+
Top Mask	Маска пайки на верхній стороні плати	+		
Bot Mask	Маска пайки на нижній стороні плати			
Top Silk	Шовкографія (контури компонентів тощо) на верхній стороні плати	+		
Bot Silk	Шовкографія на нижній стороні плати			
Top Paste	Вставка пайки на верхній стороні плати			
Bot Paste	Вставка пайки на нижній стороні плати			
Top Assy	Допоміжні дані (атрибути) на верхній стороні плати			
Bot Assy	Допоміжні дані (атрибути) на нижній стороні плати			

В колонках 1 – 3 символом "+" відмічені шари, які треба включити для друку:

- 1 – схеми розміщення компонентів,
- 2 – вигляду плати з боку встановлення компонентів,
- 3 – вигляду плати з її протилежного боку.

Додаток 2

Опції друку файлів РСВ

Таблиця 9. Опції друку файлів РСВ

<i>Опція</i>	<i>Призначення</i>	1	2	3
Rotate	Повернути зображення на 90° за стрілкою годинника	?	?	?
Mirror	Друкувати зображення в дзеркальному відображенні			+
Draft	Друкувати тільки контури товстих ліній і полігонів			
Thin Stroked Text	Друкувати текстові написи на платі	+	+	+
RefDes	Друкувати позиційні позначення компонентів	+		
Type	Друкувати назви компонентів	+		
Value	Друкувати номінали компонентів	+		
Pads	Друкувати контактні площадки	+	+	+
Vias	Друкувати перехідні отвори		+	+
Pad/Via Holes	Друкувати отвори всередині контактних та перехід-	+	+	+

Опція	Призначення	1	2	3
	них площадок			
Pick and Place	Друкувати точки прив'язки для автоматичного монтажу компонентів			
Glue Dot	Друкувати точки приклеювання			
Test Point	Друкувати контрольні точки			
Keepout	Друкувати бар'єри трасування			
Cutout	Друкувати вирізані ділянки			
Connections	Друкувати з'єднання			
Titles	Друкувати заголовки			

В колонках 1 – 3 символом "+" відмічені опції, які треба включити для друку:

- 1 – схеми розміщення компонентів,
- 2 – вигляду плати з боку встановлення компонентів,
- 3 – вигляду плати з її протилежного боку.

Додаток 3

Зміст спеціальних бібліотек P-CAD

Таблиця 10. Зміст бібліотеки *PCBMAIN.LIB* (корпуси зі штирковими виводами)

Імена компонентів	Призначення компонентів
DIPxx	Корпус с двобічним розташування виводів. Суфікси N або W вказують на вузький чи широкий корпус
DO-xx	Діоди
JMPxx	Клемні перемикачі. Суфікс T означає T-подібну форму
LEDxxx	Світлодіоди. Число xxx - відстань між центрами виводів
PBxxx	Кнопки. Число xxx - відстань між центрами виводів
PGAxx/yy	Багаторядове розташування виводів (Pin Grid Arrays). xx - число рядів, yy - число стовпців або рядків в ряду
POTxx	Потенціометри. Суфікси S або T - різні форми корпусів
REEDRLYxx	Язичкове реле
RESxx	Резистори. Число xx - відстань між центрами виводів
SIPxx	Корпус с однобічним розташування виводів
SW-DIPxx	Перемикачі з двохрядним розташування виводів
TO-xx	Транзистори. В дужках вказаний порядок розташування виводів
XTAL-OSC	Кварцові резонатори
ZIPxx	Корпус з двобічним розташуванням виводів, зміщених вправо

Таблиця 11. Зміст бібліотеки *PCBSMT.LIB* (корпуси зі штирковими виводами)

Імена компонентів	Призначення компонентів
Soxx	Корпус з двобічним розташуванням виводів, мала ширина корпусу
SOLxx	Корпус с двобічним розташуванням виводів, велика ширина корпусу
SOJxx	Корпус с двобічним розташуванням виводів, середня ширина корпусу
MO-xx/xx	Корпус с двобічним розташуванням виводів, xx - тип корпусу, по стандарту JEDEC, уу - число виводів
QFPxx	Корпус с чотирибічним розташуванням виводів
PLCCxx	Пластмасовий корпус. Суфікс А - тип корпусу за стандартом JEDEC, J - корпус квадратної форми, R - корпус прямокутної форми, L - корпус великого розміру, M – мініатюрний корпус
RCxx	Резистори
CCxx	Конденсатори
TCxx	Танталові конденсатори
WWIND/A	Котушки індуктивності
MELF1/4W	Трубчаті резистори
MLLxx,SODxx	Трубчаті резистори, конденсатори, котушки індуктивності або діоди
SOTxx	Мініатюрні транзистори або діоди

Таблиця 12. Зміст бібліотеки *PCBCONN.LIB* (роз'єми, xx – кількість виводів)

Імена компонентів	Призначення компонентів
DINxx	Роз'єми типу DIN. Суфікси: M – штиркові виводи, F – розетки.
DBxx	Мініатюрні роз'єми. Суфікси: M – штиркові виводи, F – розетки, R – зміщення виводів вправо/
CONxxPIN	Роз'єми з однорядним розташуванням виводів
IDCxx	Вертикальні роз'єми типу IDC. Суфікс R - зміщення виводів вправо
EDGExx/уу	Роз'єми з планарними виводами. уу – відстань між центрами виводів
ECONxx/уу	Роз'єми з планарними виводами фірми Texas Instruments уу – відстань між центрами виводів

Додаток 4

Схема процесорного модуля на базі мікропроцесорів Intel 8086/8087

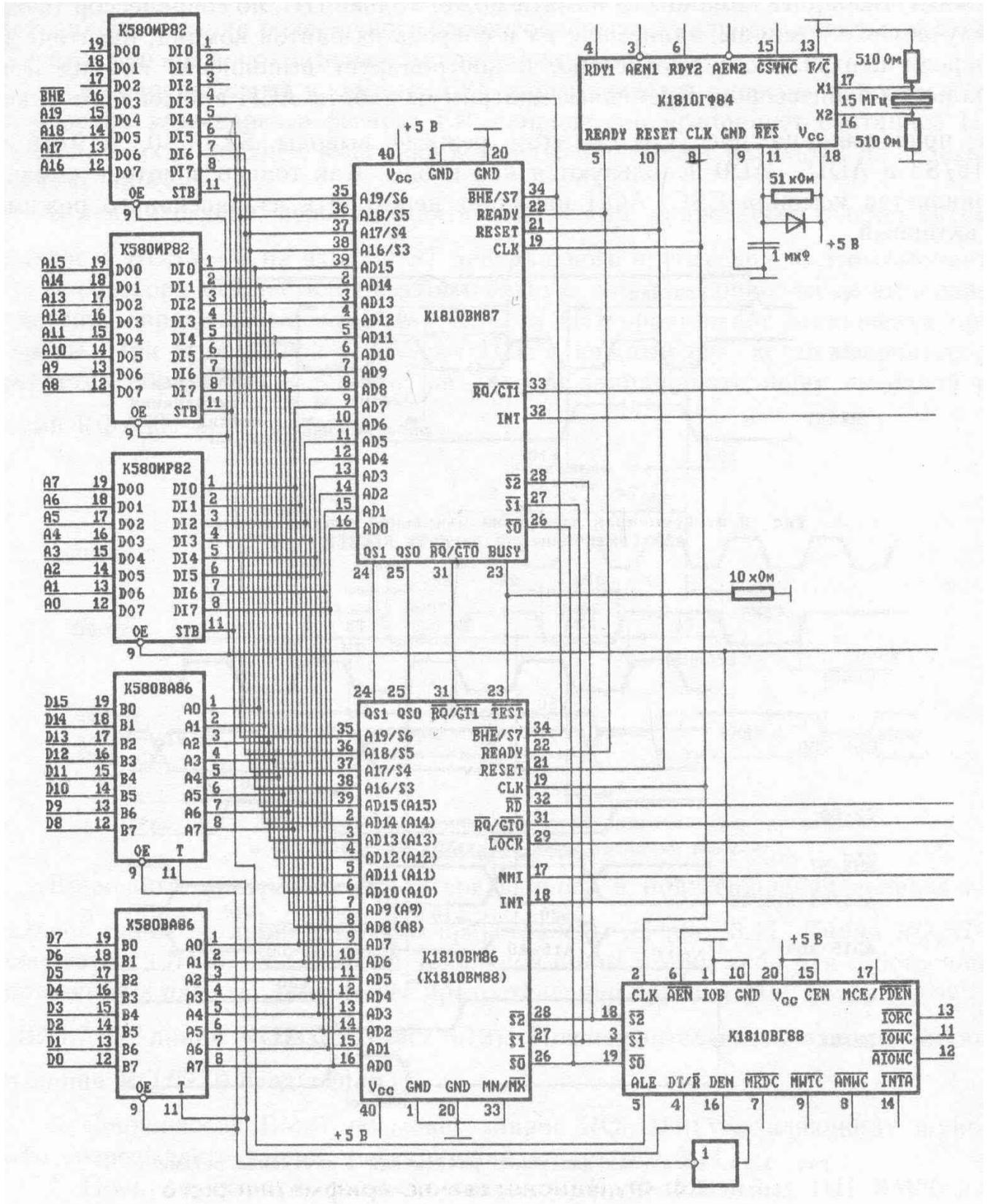


Рис. 30

Додаток 5
Зразок титульного аркуша

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І АРХІТЕКТУРИ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Курсова робота
з дисципліни “ЕОМ та мікропроцесорні системи”
на тему
“Контролер автоматичної лінії виробництва цегли”

Виконав студент гр. ІТЕП-31

_____ Цегельник М.І.

Прийняв з оцінкою _____

_____ доцент Щербина О.А.

" ___ " _____ 2001 р.

КИЇВ КНУБА 2001

Зміст

1. Загальні відомості про технологію виробництва друкованих плат	3
1.1. Типи плат	3
1.2. Формування малюнка з'єднань	3
1.3. Встановлення компонентів	5
2. Система автоматизованого проектування P-CAD	5
2.1. Структура і склад системи P-CAD 2001	5
2.2. Графічні редактори P-CAD 2001	6
3. Робота з графічним редактором <i>Schematic</i>	8
3.1. Встановлення конфігурації	8
3.2. Розміщення символів	9
3.3. Розміщення ліній електричного зв'язку (провідників)	10
3.4. Розміщення ліній групового зв'язку (шин)	12
3.5. Друк схемних файлів	13
3.6. Створення файлу зв'язків	13
4. Робота з графічним редактором <i>PCB</i>	14
4.1. Встановлення конфігурації	14
4.2. Способи розробки друкованих плат	14
4.3. Розміщення компонентів на платі	15
4.4. Автоматичне трасування плати	17
4.5. Друк файлів PCB	18
5. Робота з бібліотеками компонентів в P-CAD	20
5.1. Структура бібліотек компонентів в P-CAD та програми для роботи з ними	20
5.2. Приклад створення нового компонента	21
5.2.1. Створення символу компонента в редакторі <i>Symbol Editor</i>	21
5.2.2. Створення корпусу компонента в редакторі <i>Pattern Editor</i>	23
5.2.3. Створення опису компонента в програмі <i>Library Executive</i>	24
5.2.4. Логічна еквівалентність секцій і виводів	26
5.3. Інші операції з бібліотеками	27
6. Зміст лабораторних робіт	28
6.1. Завдання на лабораторну роботу № 1. Робота з бібліотеками P-CAD	28
6.2. Завдання на лабораторну роботу № 2. Розробка принципової схеми і друкованої плати процесорного модуля	31
6.3. Завдання на лабораторну роботу № 3. Розробка принципової схеми і друкованої плати модуля пам'яті	32
6.4. Завдання на лабораторну роботу № 4. Розробка принципової схеми і друкованої плати модуля портів зовнішніх пристроїв	33
6.5. Завдання на лабораторну роботу № 5. Розробка принципової схеми і друкованої плати контролера	33
7. Зміст курсової роботи	34
7.1. Вибір теми	34
7.2. Загальний опис складу і функціонування об'єкту управління	35
7.3. Опис джерел і споживачів інформації об'єкта та форматів портів вводу-виводу контролера	35
7.3.1. Деякі приклади	36
7.4. Розробка алгоритмів управління об'єктом	39
7.5. Розробка програм управління об'єктом	41
7.6. Розробка принципової схеми контролера	43
7.7. Розробка друкованої плати контролера	43
7.8. Оформлення і захист курсової роботи	43
8. Література	45
9. Інтернет	45
Додаток 1 Елементи та шари зображень	46
Додаток 2 Опції друку файлів PCB	46
Додаток 3 Зміст спеціальних бібліотек P-CAD	47
Додаток 4 Схема процесорного модуля на базі мікропроцесорів Intel 8086/8087	49
Додаток 5 Зразок титульного аркуша	50

Навчально-методичне видання

Розробка друкованих плат електронних пристроїв у системі автоматизованого проектування P-CAD 2001.

Методичні вказівки до виконання лабораторних і курсових робіт
з дисципліни “ЕОМ та мікропроцесорні системи”
для студентів спеціальностей 6.0804.00
“Інформаційні управляючі системи та технології”,
“Інформаційні технології проектування”

Укладачі: **Щербина** Олександр Андрійович
Орлова Марія Миколаївна

Підписано до друку

Формат 60x84 ^{1/16}.

Папір офсетний. Гарнітура Аріал. Друк на різнографі.

Ум.- друк. арк.

Облік.-вид. арк. 0,75.

Ум.-фарбовідб.

Тираж 100 прим. Вид. №

Зам. №

Редакційно-видавничий відділ КНУБА.
03037, Київ-37, Повітрофлотський проспект, 31

Віддруковано в Центрі інформаційних технологій
Київського національного університету будівництва і архітектури