

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

# ПРОЄКТУВАННЯ НАПІВПРОВІДНИКОВИХ ПРИЛАДІВ ТА ІНТЕГРАЛЬНИХ МІКРОСХЕМ КОМП'ЮТЕРНИЙ ПРАКТИКУМ

**Навчальний посібник**

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського  
як навчальний посібник для здобувачів ступеня магістра  
за освітньою програмою «Мікро- та наноелектроніка»  
спеціальності 153 Мікро- та наносистемна техніка

Укладачі: Ю. В. Діденко, А. Т. Орлов, Д. Д. Татарчук

Електронне мережне навчальне видання

Київ  
КПІ ім. Ігоря Сікорського  
2022

Рецензент *Мельник І. В.*, доктор технічних наук, професор, професор кафедри електронних пристроїв та систем, КПІ ім. Ігоря Сікорського

Відповідальний редактор *Свєчніков Г.С.*, канд. фіз.-мат. наук, старш. наук. співроб.

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського  
(протокол № 5 від 26.05.2022 р.)  
за поданням Вченої ради факультету електроніки  
(протокол № 4/22 від 25.04.2022 р.)*

У посібнику наведено інструкції до виконання комп'ютерного практикуму з дисципліни "Проектування напівпровідникових приладів та інтегральних мікросхем" з використанням системи автоматизованого проектування Electric VLSI Design System. Послідовно розглянуто етапи проектування інтегральних мікросхем від одиночного МОН транзистора до функціонального вузла. Значна кількість ілюстрацій допоможе читачеві легко засвоїти базові операції роботи з САПР Electric VLSI Design System.

Посібник призначений для здобувачів ступеня магістра, які навчаються за освітньою програмою «Мікро- та наноелектроніка» спеціальності 153 Мікро- та наносистемна техніка.

Реєстр. № НП 21/22-503. Обсяг 13,7 авт. арк.

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
проспект Перемоги, 37, м. Київ, 03056  
<https://kpi.ua>

Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

## ЗМІСТ

Вступ.....	4
Інструкція зі встановлення та налаштування <i>Electric VLSI Design System</i> ....	8
Практичне заняття 1. Моделювання резистивного подільника напруги ....	12
Практичне заняття 2. Моделювання роботи <i>NMOS</i> і <i>PMOS</i> транзисторів..	42
Практичне заняття 3. Моделювання роботи КМОН інвертора.....	75
Практичне заняття 4. Проєктування КМОН схеми І-НІ .....	99
Практичне заняття 5. Проєктування кільцевого генератора .....	121
Практичне заняття 6. Розміщення макетів схем на площинці для фабричного виробництва.....	143
Список використаних джерел .....	164
Рекомендована література.....	164

## Вступ

*Electric VLSI Design System* – це система автоматизованого проєктування (САПР), використовувана для розробки електричних схем і проєктування топології друкованих плат й інтегральних схем. Крім іншого, це зручний інструмент для використання мов описування апаратури, таких як *VHDL* і *Verilog*.

*Electric VLSI Design System* був відкритим програмним забезпеченням (англ. *open source software*) протягом багатьох років, і зараз доступний через некомерційну організацію «*Free Software Foundation*» («Фонд вільного програмного забезпечення»).

*Electric VLSI* має безліч стилів редагування, що включають топологію, схематику, ілюстрації, архітектурне проєктування. *Electric VLSI* може взаємодіяти з різними специфікаціями й форматами файлів, таких як *VHDL*, *CIF*, *GDS*.

Найцінніша вбудована в *Electric VLSI* можливість – це система прив'язок, яка дає можливість здійснювати проєктування зверху вниз із дотриманням цілісності всіх з'єднань [1].

Завдяки наведеним перевагам *Electric VLSI* широко використовується університетами та компаніями-розробниками інтегральних схем у всьому світі (рис. В.1).

*Electric* поєднує в собі безліч різних тестів й аналізуючих інструментів:

- *Design Rule Checking (DRC)* – програма контролю правил проєктування;
- *Electrical Rule Checking (ERC)* – це програма, яка контролює дотримання правил проєктування електричних схем;
- *Simulation* – два вбудовані симулятори та інтерфейси для більш ніж дюжини промислових інструментів (*Spice*, *Verilog* тощо);
- *Routing* – містить у собі п'ять різних трасовиків для широкого кола завдань;

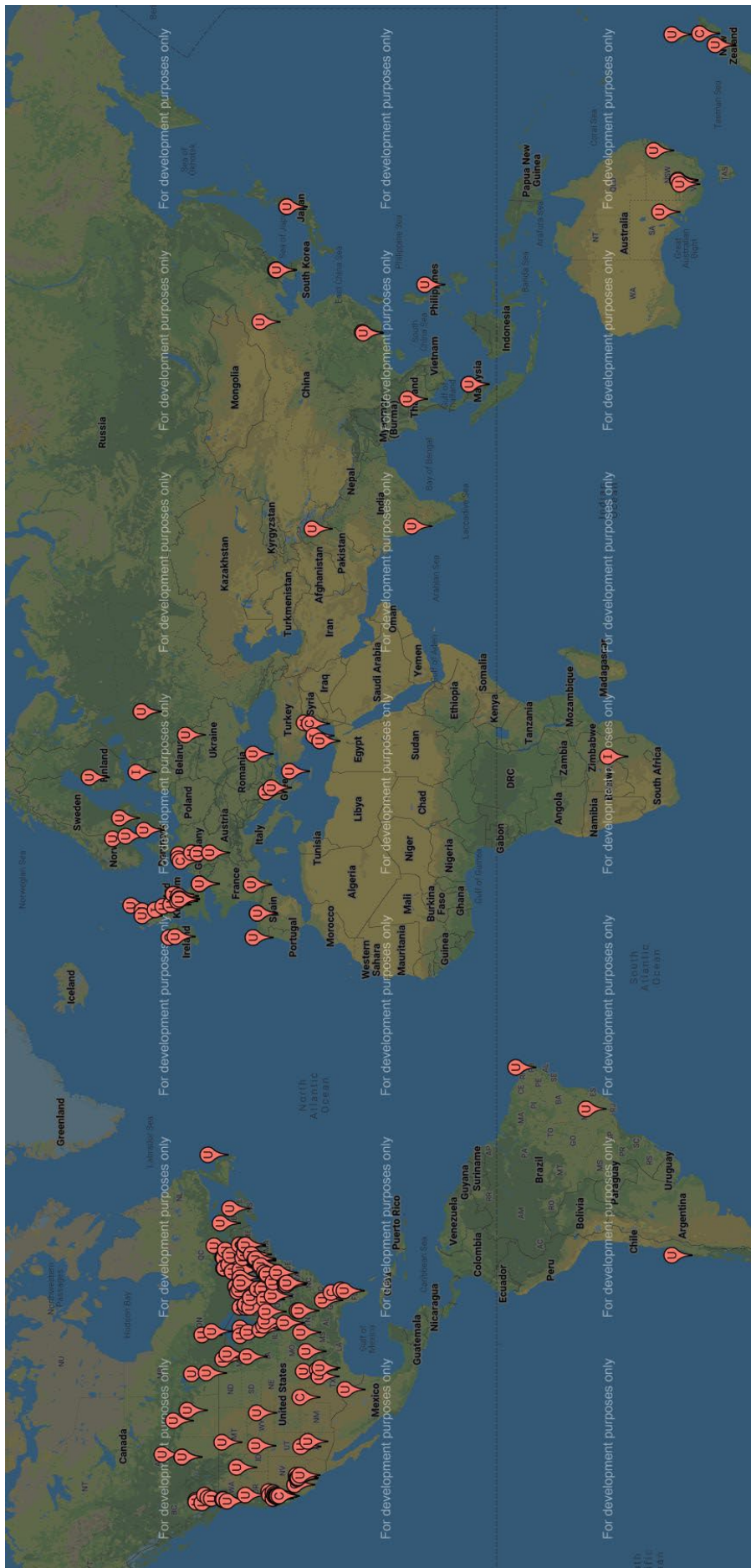


Рис. В.1. Використання *Electric VLSI Design System* у світі [2]

- *Logical Effort* – інструмент для аналізу схеми і зміни її компонентів з урахуванням методу логічного зусилля;
- *LVS (layout vs. schematic)* – інструмент для порівняння деяких обраних еквівалентів схем (у деяких версіях *Electric* позначається як *NCC*);
- Читання/Запис – здатність зчитувати й записувати опис схеми у безліч форматів, включаючи *CIF*, *GDS*, *EDIF*, *DXF* і *VHDL*.

*Electric* підтримує безліч різних технологій, наприклад: КМОН; N-МОН; біполярну.

У більшості САПР використовується два способи проектування інтегральних мікросхем: забезпечення зв'язаності й геометричний. *Electric VLSI Design System* відрізняється від інших тим, що він використовує зв'язаність для всього проекту, включаючи топологію інтегральної схеми. Це означає, що користувач розміщує компоненти (МОН транзистори, контакти тощо) і рисує провідники (метал1-2, полікристалічний кремній тощо) для їхнього з'єднання. На екрані відображається реальна геометрична форма (рис. В.2).

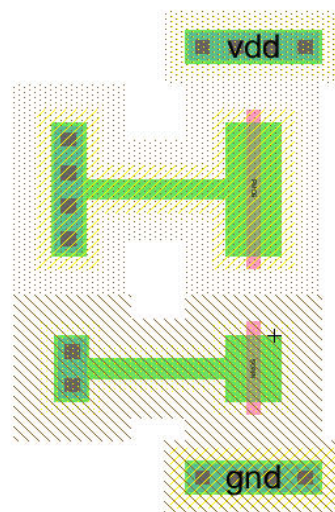


Рис. В.2. Приклад топології, реалізованої засобами *Electric VLSI Design System*

Проектування топології ІС із забезпеченням зв'язаності виключає геометричні помилки. Складні компоненти більше не складаються з

незв'язаних геометричних частин, які можуть переміщатися незалежно одна від одної.

У системах прорисовування (*paint systems*), користувач може випадково відсунути область заслона від транзистора, таким чином, знищуючи транзистор. В *Electric CAD* транзистор – це єдиний об'єкт, який не може випадково зруйнуватися.

У системах конструкторського проектування електронних пристроїв поширення набули програми моделювання, об'єднані загальною назвою – *SPICE*-програми (наприклад, програми *PSpice*, *TSpice*, *XSpice* тощо).

Текстова форма опису електричних схем для *SPICE*-програм фактично стала стандартом для систем автоматизованого проектування. Крім *SPICE*-програми, система проектування *LTspice* включає схемний редактор і засоби відображення результатів моделювання (графічний постпроцесор), які дають змогу проводити подальший аналіз. До складу системи входить також вбудована база даних для більшості імпульсних пристроїв компанії *Linear Technology* і багатьох пасивних компонентів. База даних пристроїв, схемний редактор, програма моделювання й система графічного відображення результатів інтегровані в одну систему.

Система *LTspice* була розроблена для моделювання імпульсних систем живлення й регулювання та відрізняється від інших *SPICE*-програм спеціальною логікою, яка керує перемиканнями, забезпечуючи прийнятний час моделювання.

Система моделювання дає змогу одержати практично довільну форму імпульсів. У *LTspice* вбудовані схемні елементи відповідні фізично існуючим компонентам. Конденсатори й котушки індуктивності можуть бути змодельовані з послідовними опорами та іншими паразитними елементами, без використання підсхем або внутрішніх вузлів. Це дає змогу врахувати кінцевий внутрішній опір елементів на всіх частотах.

Програма вільно поширюється через вебсайт компанії *Linear Technology*.

## Інструкція зі встановлення та налаштування *Electric VLSI Design System* та *LTspice*

### **Встановлення *Electric VLSI Design System* для *Windows*:**

1. Встановіть останню версію ПЗ *java* на ПК (завантажити програму можна з сайту <https://java.com>);
2. Створіть директорію *C:\Electric\*;
3. Завантажте файл *electric-X.XX.jar* (де *X.XX* – версія програми; на момент написання даного посібника останньою була версія 9.07) за посиланням <ftp://ftp.gnu.org/pub/gnu/electric/electric-X.XX.jar> і збережіть в каталог *C:\Electric\*. Простежте, щоб розширення файлу було *\*.jar*, тому що під час завантаження *Windows* може змінити розширення;
4. Двічі клацніть лівою клавiшею мишки по завантаженому файлу. Запуститься *Electric VLSI Design System*.

### **Налаштування ярлика *Electric VLSI Design System*:**

1. Клацніть правою кнопкою мишки на робочому столі й оберіть Створити → Ярлик;
2. У полі вибору об'єкта введіть «*C:\Electric\electric-X.XX.jar -sdi*». Параметр «*-sdi*» запускає програму в стилі роздільних вікон. Це зручно якщо Ви використовуєте монітор з малою діагоналлю. Також, це налаштування знаходиться в меню *File* → *Preferences* → *Display* → *Display Control* → *Display style*. Щоб зміни набули чинності потрібно запустити знову *Electric*;
3. Далі, укажіть назву для ярлика (наприклад, *Electric VLSI Design System*) і натисніть на кнопку «Готово». Значок має з'явитися на робочому столі;
4. Потім завантажте іконку *Electric VLSI System* за посиланням <http://cmosedu.com/cm0s1/electric/electric.ico> і збережіть її в *C:\Electric\*;
5. Клацніть правою кнопкою миші по іконці *Electric* на робочому столі й виберіть Властивості → Ярлик → Змінити значок. Виберіть значок у директорії *C:\Electric\*;



6. Подвійне клацання лівою клавішею мишки по значковій *Electric VLSI Design System* на робочому столі запустить програму.

### **Збільшення обсягу RAM в Java Virtual Machine (JVM):**

Щоб збільшити обсяг доступної пам'яті *Electric VLSI* (за замовчуванням вона становить 64 МБ), запустіть програму *Electric VLSI* і перейдіть у меню *File* → *Preferences* → *General*, збільшіть максимальний розмір пам'яті, як показано на рисунку В.3.

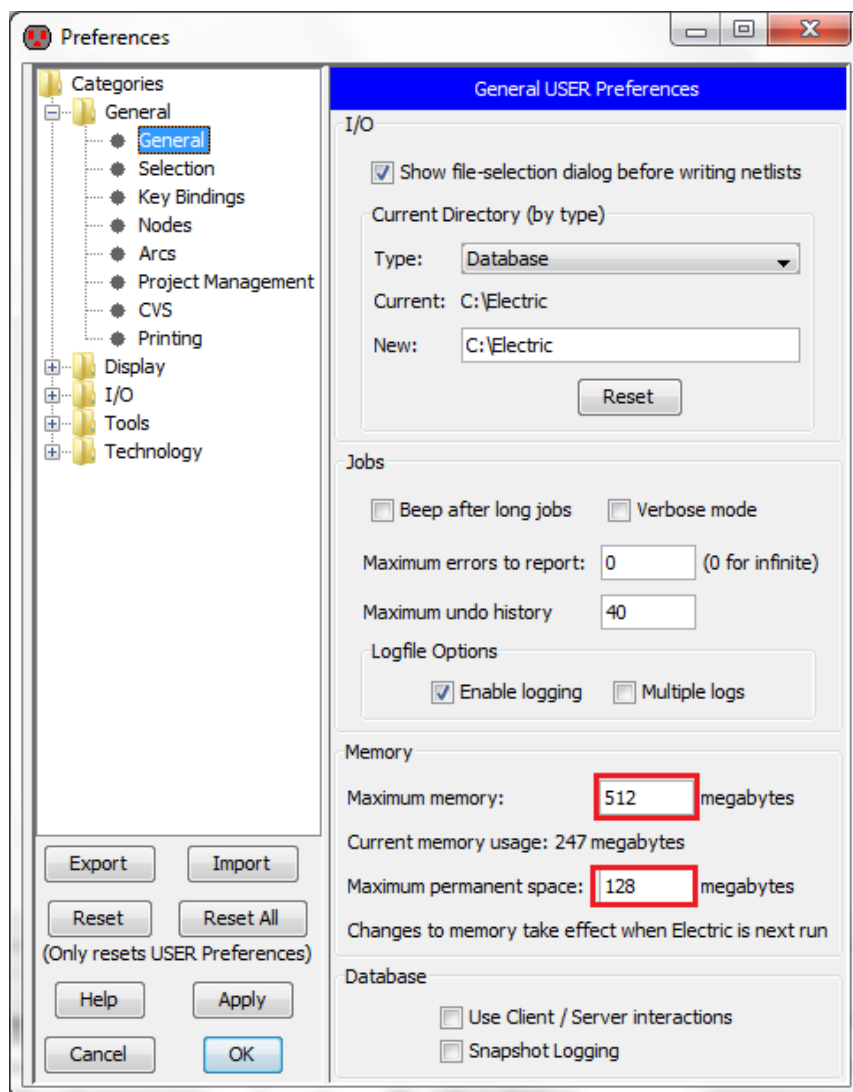


Рис. В.3. Налаштування для збільшення обсягу доступної пам'яті

Якщо після виконання зазначених дій програма перестала запускатися (така ситуація МОЖЕ виникнути на ДЕЯКИХ комп'ютерах), значить

параметр «*Maximum permanent space*» потрібно знову встановити рівним нулю. Для цього в реєстрі перейдіть у папку *HKEY\_CURRENT\_USER\Software\Javasoft\Prefs\com\sun\electric\tool\user* і встановіть значення параметра */Perm/Size* рівним 0 (або видаліть його).

### Налаштування *LTspice*

1. Встановіть *LTspice* на комп'ютер (програму можна завантажити за посиланням <https://ltspice.analog.com/software/LTspiceIV.exe>); укажіть шлях встановлення: «*C:\LTspiceIV*».
2. Запустіть *Electric VLSI* і перейдіть у меню *File* → *Preferences* → *Tools* → *Spice*;
3. Виставте параметри так, як зазначено на рисунку В.4.

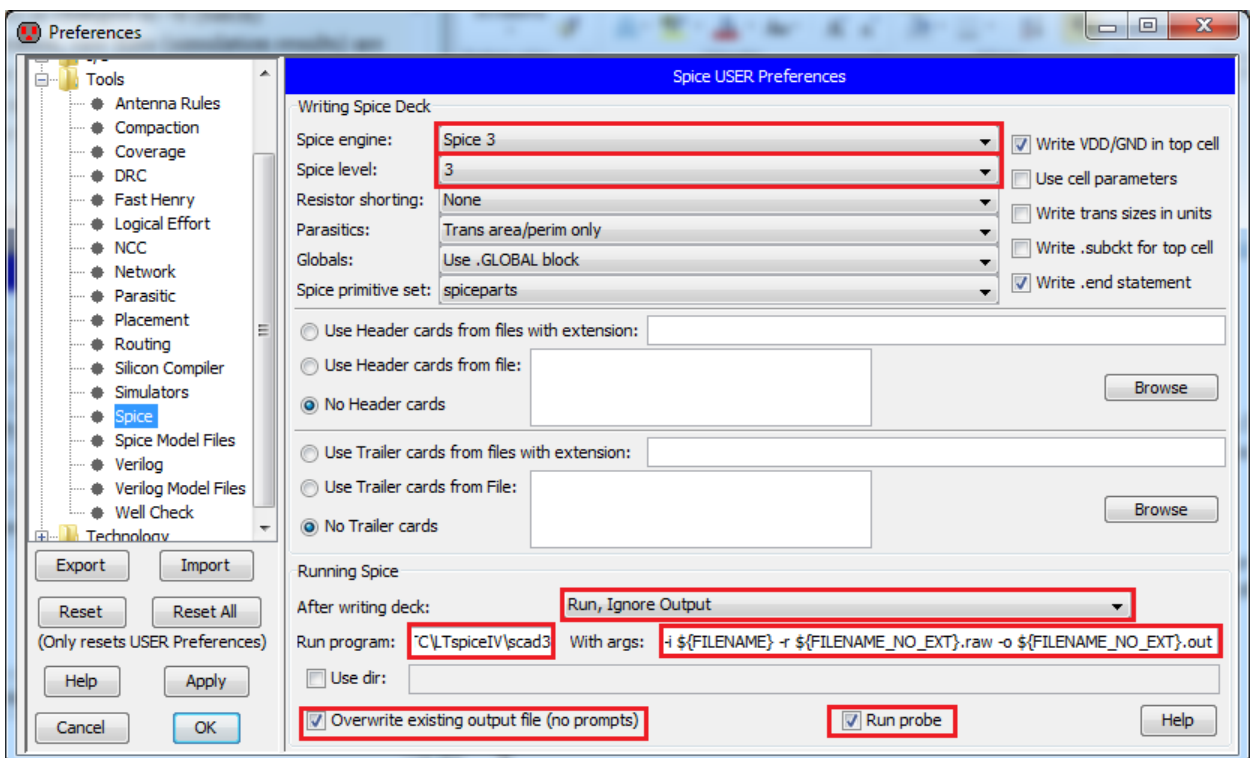


Рис. В.4. Налаштування *LTspice*

4. У рядку «*Run program*» (шлях запуску програми) укажіть місце знаходження файлу *scad3.exe*: «*C:\LTspiceIV\scad3.exe*».
5. У рядок «*With args*» (з аргументами) введіть:

```
-i ${FILENAME} -r ${FILENAME_NO_EXT}.raw  
-o ${FILENAME_NO_EXT}.out
```

Зверніть увагу, що поле «*Run program*» не чутливе до регістру, а поле «*With args*» заповнюється з урахуванням регістру.

### ***Видалення Electric VLSI Design System***

Для коректного видалення програми потрібно в реєстрі видалити папку `HKEY_CURRENT_USER\Software\Javasoft\Prefs\com\sun\electric\`. Після чого можна видалити папку `C:\Electric\`.

### ***Видалення LTspice***

Для видалення програми *LTspice* натисніть Пуск→Панель Керування→Програми й компоненти. Клацніть правою кнопкою мишки по пункту *LTspice* списку встановлених програм. Оберіть «Змінити/Видалити». Далі додержуйтесь підказок програми «Майстра видалення програм».

## Практичне заняття №1

### Моделювання резистивного подільника напруги

За замовчуванням каталог зі збереженими роботами – *C:\Electric\*.

Переконайтеся, що Ви збільшили обсяг оперативної пам'яті в *JVM*, як описано в інструкції зі встановлення та налаштування *Electric VLSI*.

Запустіть *Electric*. Ви побачите стартове вікно (рис. 1.1).

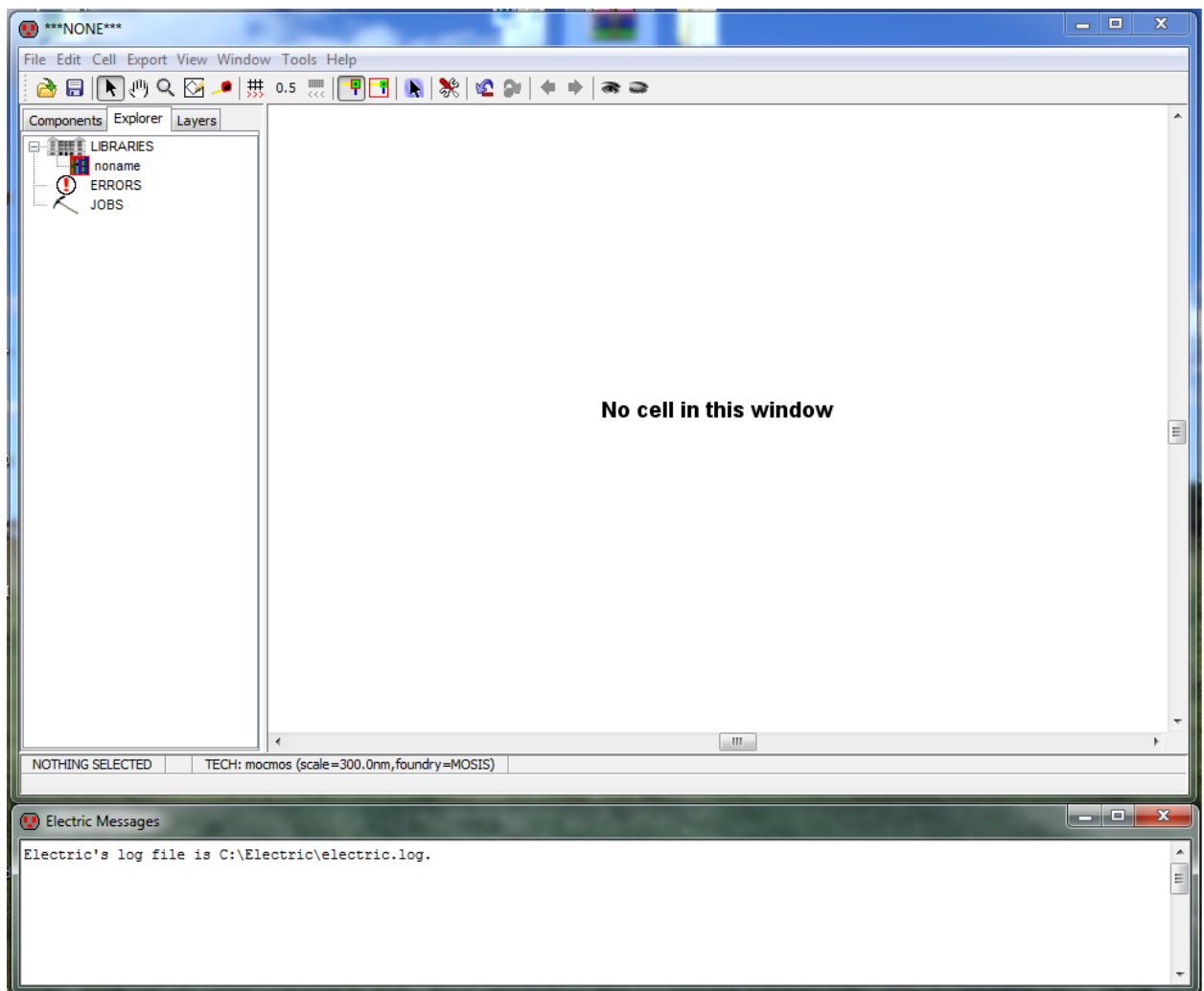



Рис. 1.1. Стартове вікно *Electric VLSI*

Тепер необхідно виставити в налаштуваннях *Electric VLSI* правила згідно зі стандартами *ON Semiconductor* для технологічного процесу *C5* і виробництва через *MOSIS* (цей процес використовується в *MOSIS* масштабованих *CMOS (SCMOS) IC*). Для цього перейдіть у меню

*File* → *Preferences* (або просто натисніть на значок ключ/викрутка ), оберіть категорію *Technology* → *Technology*. У результаті Ви побачите відповідне вікно налаштувань (рис. 1.2).

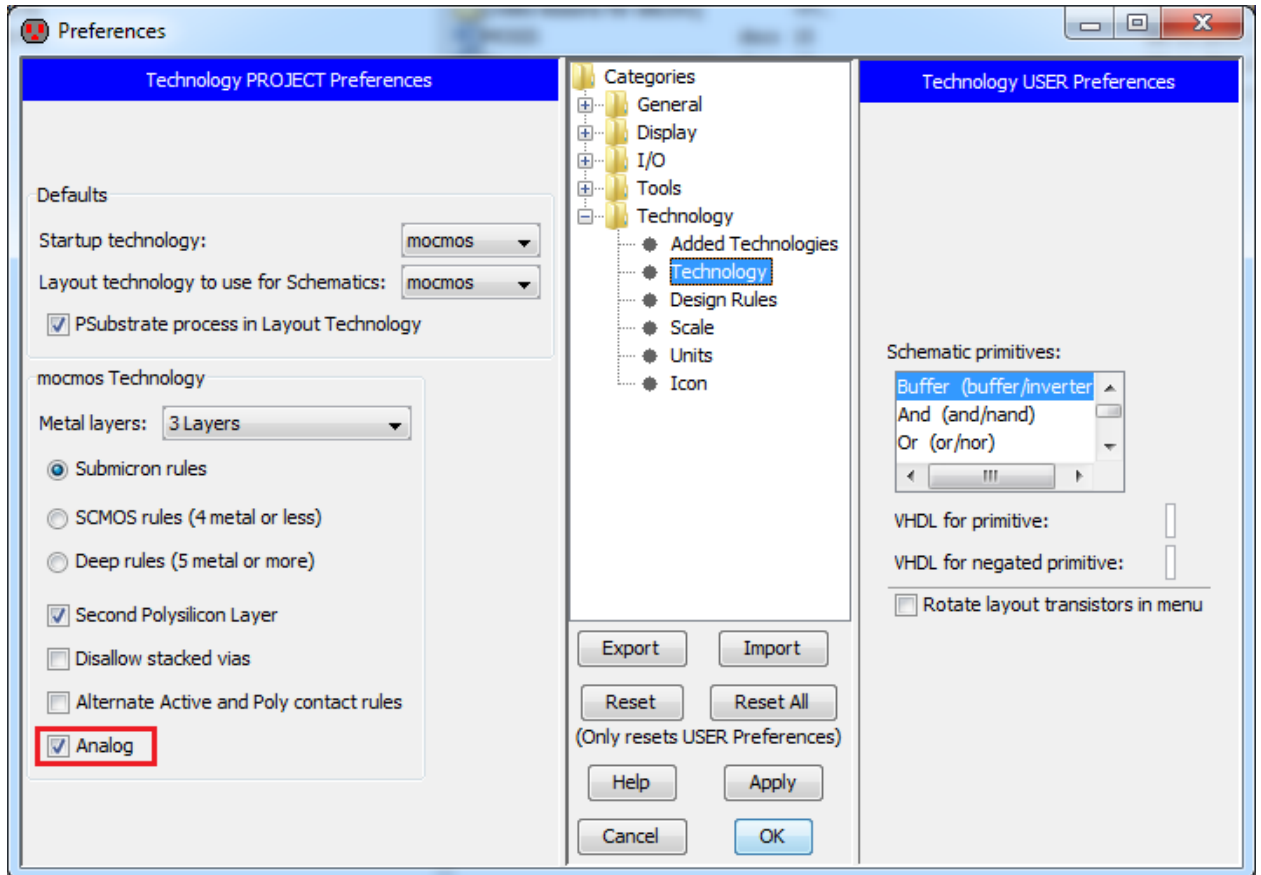


Рис. 1.2. Налаштування правил для технологічного процесу *C5*

Проконтролюйте, щоб значення, зазначені в налаштуваннях технології, відповідали значенням, зазначеним на рисунку.

Зверніть увагу, у групі налаштувань “*mocmos Technology*”, обраний пункт “*Analog*”.

Далі, укажіть норму проєктування для *C5* технології. Вона становить 300 нм, згідно із правилами субмікронного дизайну *MOSIS* для масштабованих *CMOS*.

*Electric* представляє всі відстані в безрозмірних одиницях. Для переведення цих одиниць у реальні відстані, у кожній технології є своя норма проєктування (*scale*), представлена в нанометрах. Вона показана в рядку

стану після назви технології. Наприклад, у технології *MOSIS CMOS (mosmos)* проєктна норма дорівнює 200 нм, і транзистор розміром 2 x 3 умовних одиниць фактично має розмір 400 x 600 нм. Поняття проєктної норми існує лише для технологій інтегральних мікросхем. Воно не застосовне до технологій *Artwork*, *Schematics* та інших технологій, які не пов'язані із кремнієвим рівнем проєктування.

Для встановлення проєктної норми перейдіть у меню *File* → *Preferences* → *Technology* → *Scale*, і встановіть *mosmos scale* 300 нм (рис. 1.3).

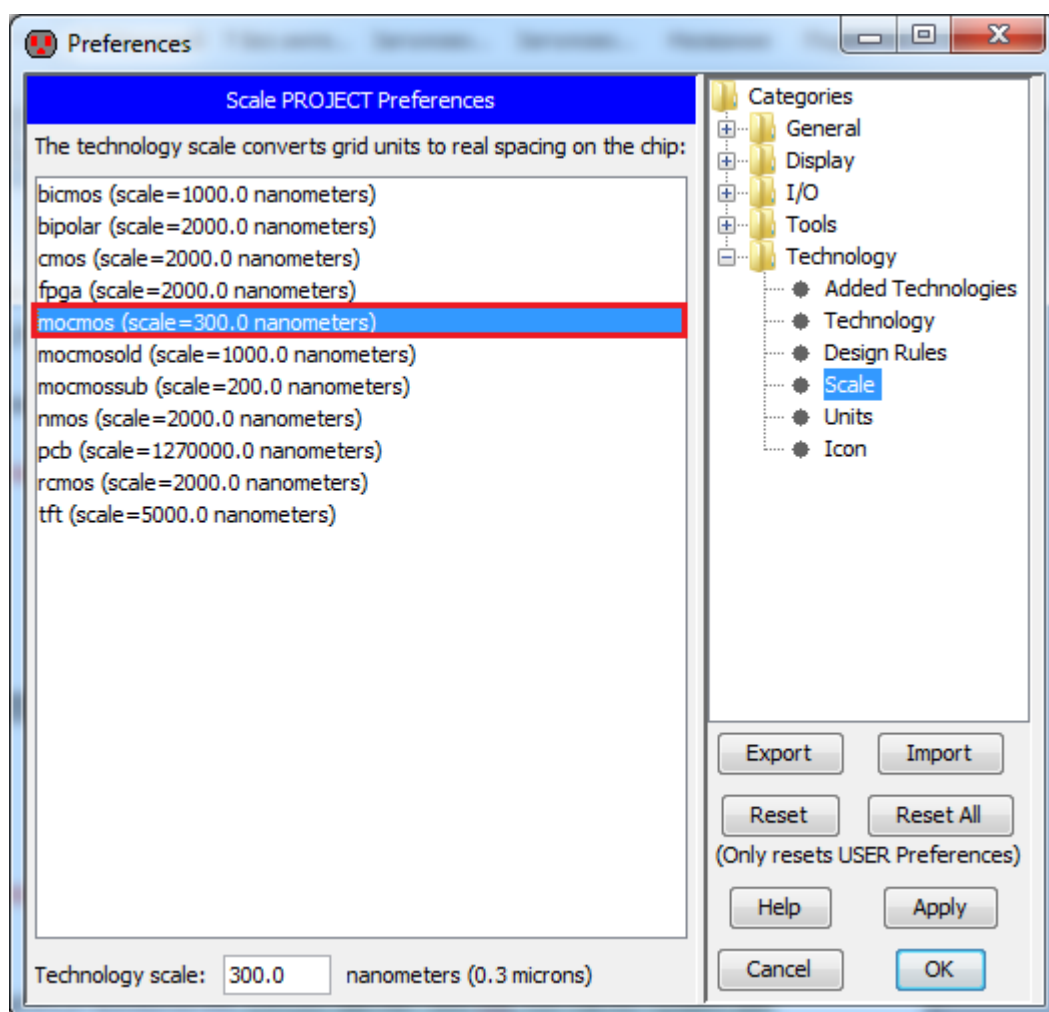


Рис. 1.3. Встановлення проєктної норми

Натисніть “OK” і закрийте меню налаштувань.

У вікні, що з'явилося, “*Saving Project Preferences Changes*” натисніть *Mark All Libs*, щоб внесені зміни вплинули на всі бібліотеки. Якщо *Electric*

*VLSI* використовується вперше, то вікно “*Saving Project Preferences Changes*” з'являтися не буде.

Налаштування, які Ви вказали, можна зберегти у файл і використовувати надалі. Це зручно якщо операторів ПК декілька. За необхідності Ви можете завантажити налаштування з нього: *File* → *Import* → *User Preferences*, укажіть на файл і натисніть «*Open*».

У результаті Ви маєте налаштовану програму для проектування чипів за технологічним процесом *C5* через *MOSIS* (Технологічний код *SCN3ME\_SUBM*). Далі збережіть Вашу першу бібліотеку: меню *File* → *Save Library As* → *tutorial\_1.jelib*.

Почнемо проектувати схему резистивного подільника.

Переходимо в меню *Cell* → *New Cell*. Назвемо її *R\_divider*, режим перегляду – *schematic* (рис. 1.4).

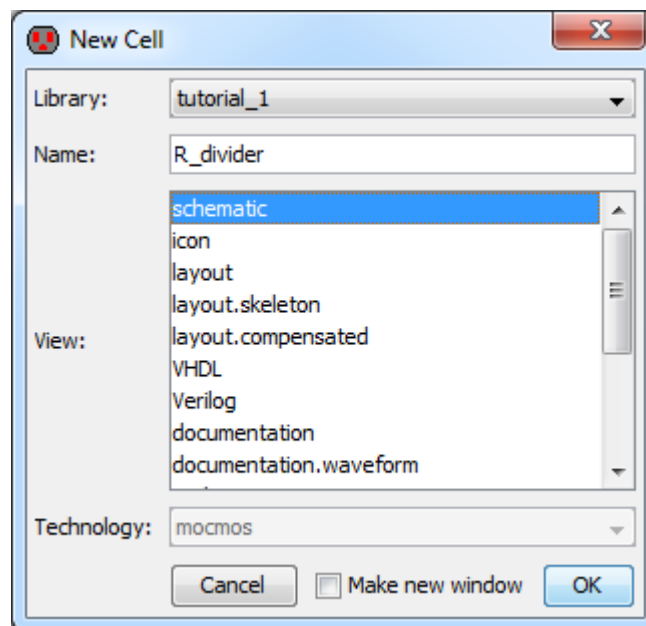


Рис. 1.4. Створення схемної комірки

Натискаємо “*OK*”.

Розглянемо ліву панель робочого вікна *Electric VLSI*. У ній є три закладки (рис. 1.5):

- *Components* – вкладка вибору компонентів;

- *Explorer* – вкладка, у якій відображається ієрархія бібліотек і комірок;
- *Layers* – відображення шарів макета (включити/виключити відображення певних шарів).

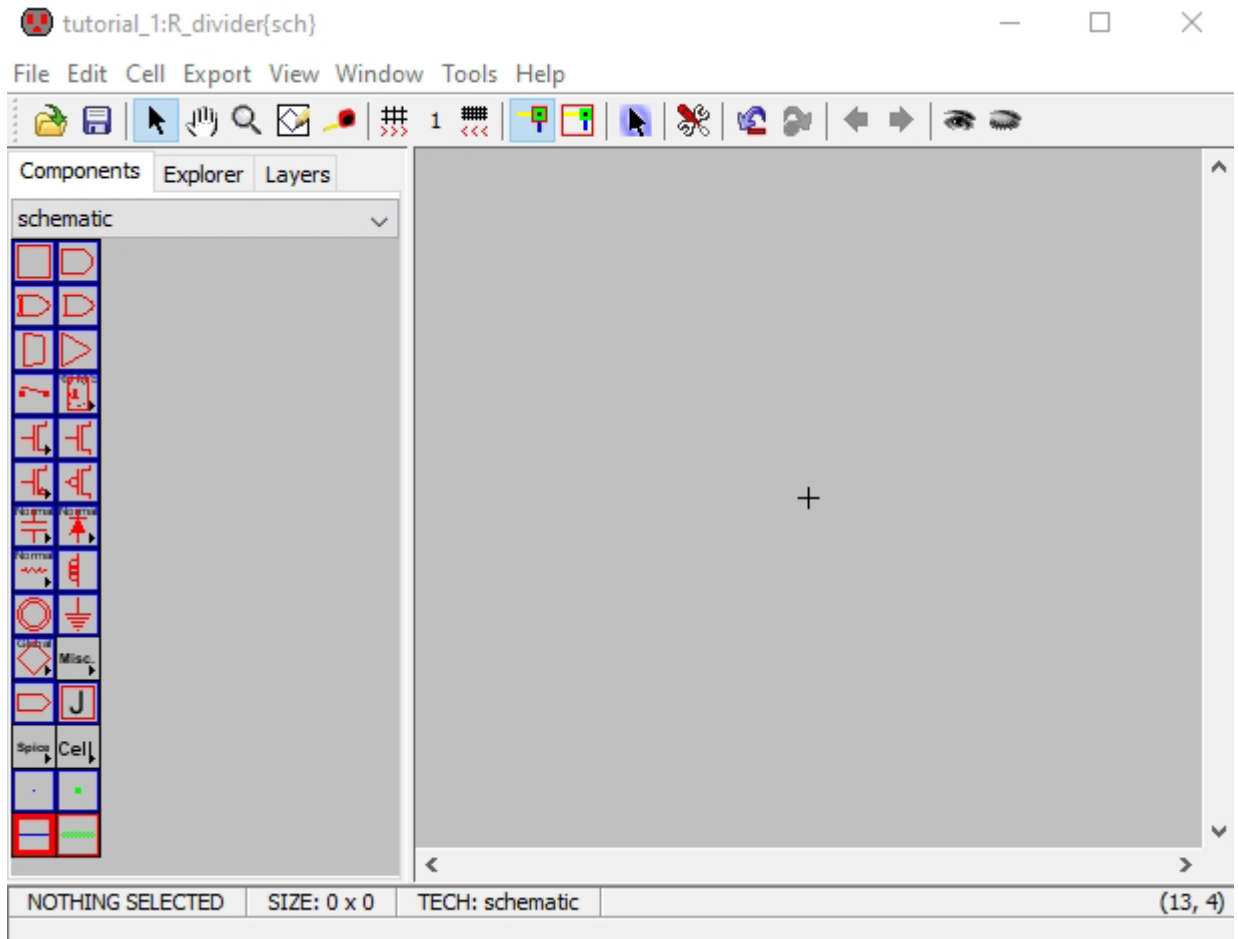


Рис. 1.5. Робоче вікно *Electric VLSI*

У закладці “*Components*” знайдіть комірку, яка містить резистор і слово “*Normal*”. Натисніть на стрілку в цій комірці й виберіть *n-well* – Ви вибрали схематичне позначення *N-Well* резистора.

В *Electric VLSI* є безліч компонентів, які можуть бути використані в схемі або макеті. Приклади вузлів включають транзистори, резистори, конденсатори тощо.

Отже, виберемо *n-well* резистор у меню *Components*, клацнувши лівою кнопкою мишки. Тепер, клацніть лівою кнопкою мишки по області



проектування, для вставки компонента (рис. 1.6). Використовуючи в меню *Window* команди *zoom out/in*, масштабуйте зображення так, як Вам буде зручно.

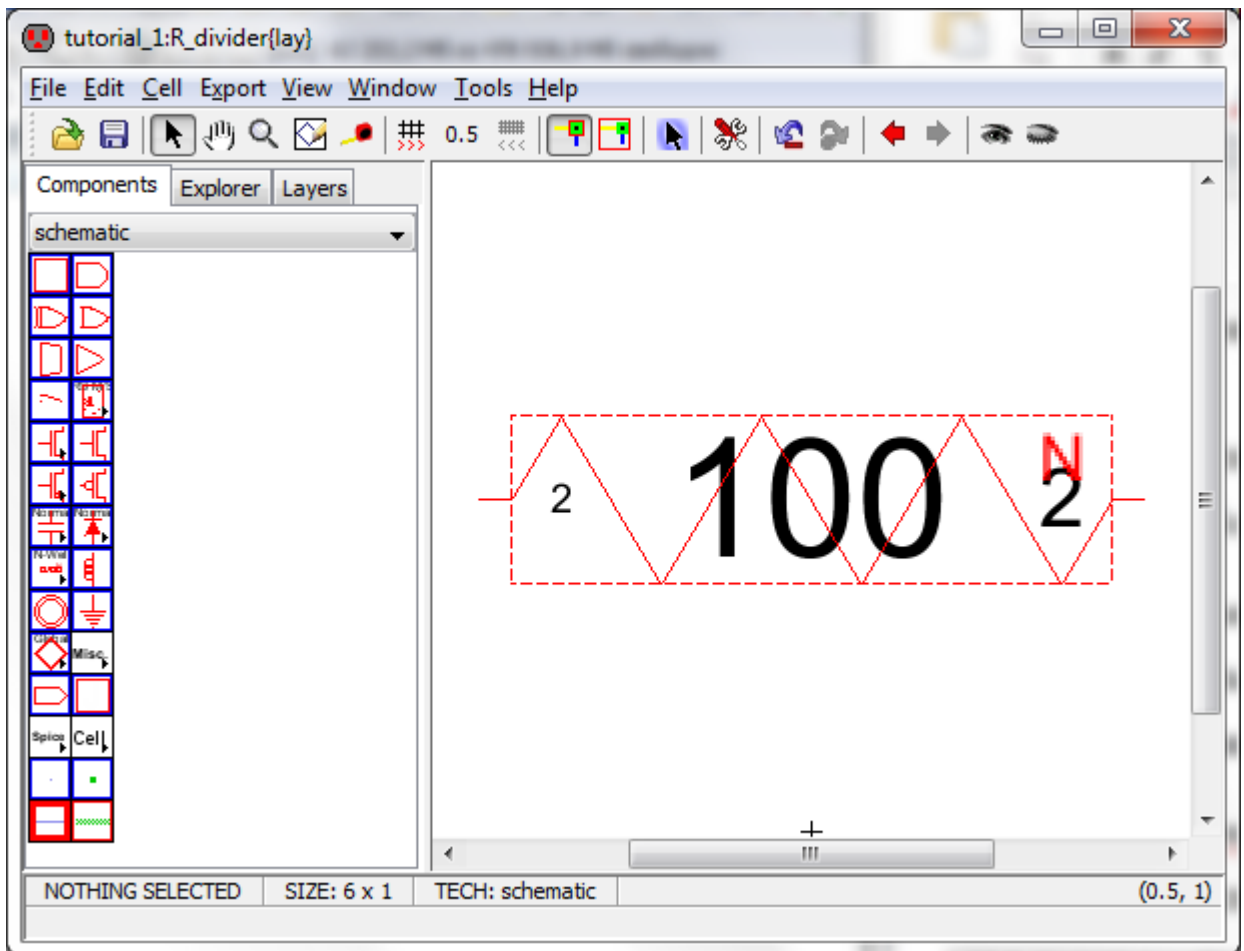


Рис. 1.6. *N-Well* резистор в області проектування

Якщо навести на компонент курсор, то довкола нього з'явиться рамка. Це говорить про те, що компонент може бути виділений, і відредагований. Нажавши на *Shift* під час клацання лівою кнопкою миші можна вибирати\скасувати вибір декількох об'єктів.

Виділіть резистор, клацнувши на ньому лівою кнопкою мишки. Тепер виберіть у меню *Edit*  $\rightarrow$  *Properties*  $\rightarrow$  *Object Properties* (або просто *Ctrl+I*) для редагування властивостей компонента. Поверхневий опір *n*-кармана в *C5* технології становить приблизно 800 Ом. Мінімальна ширина *n*-кармана становить  $12\lambda$  ( $\lambda$  – норма проектування; для *C5* технології  $\lambda = 300$  нм).

Зробимо резистор опором 10 кОм, шириною 15 одиниць і довжиною 187,5 одиниць. Введіть значення (рис. 1.7) та натисніть «OK».

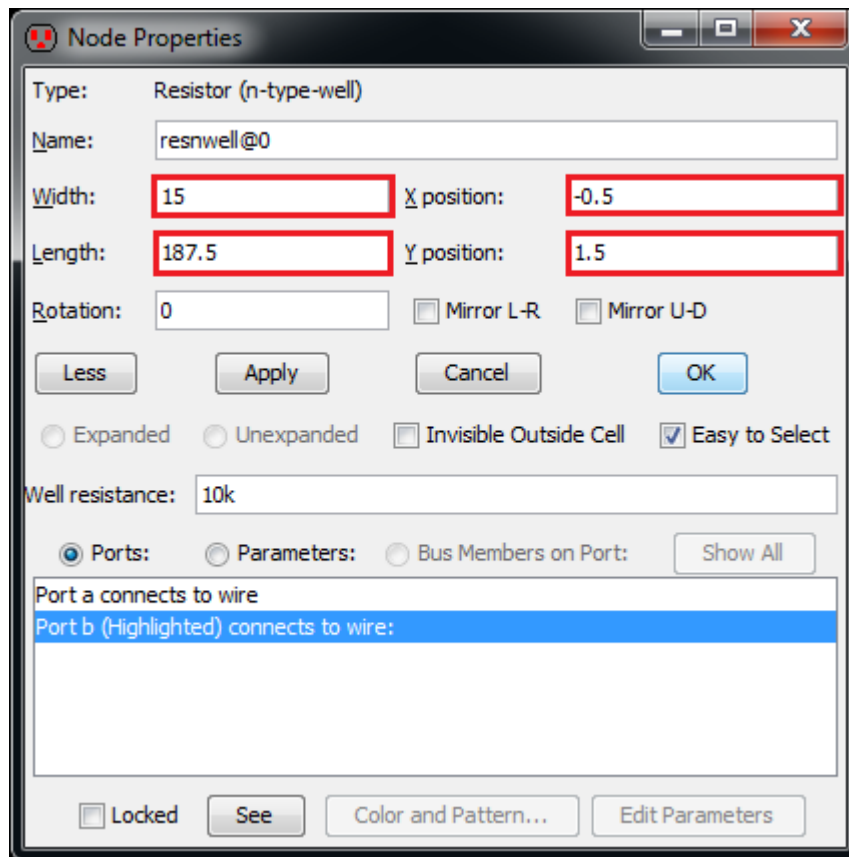


Рис. 1.7. Введення параметрів резистора

Ми одержали схематичне зображення 10k n-well резистора (рис. 1.8).

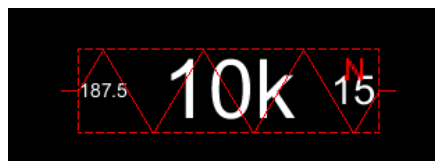


Рис. 1.8. Схематичне позначення n-well резистора

Перевіримо дотримання правил проєктування – *Design Rule Checking (DRC)*. Перейдіть у меню *Tools* → *DRC* → *Check Hierarchically* (або просто натисніть клавішу *F5*), щоб перевірити схему на наявність помилок. Якщо Ви все зробили так, як було описано вище, то у вікні *Electric Messages* буде

виведене повідомлення про відсутність помилок (рис. 1.9), що свідчить про те, що все зроблене правильно.

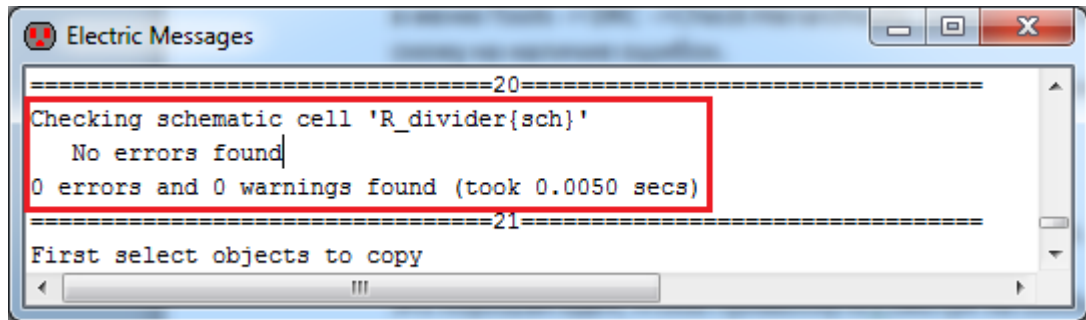


Рис. 1.9. Вікно повідомлень

За повідомленнями у вікні *Electric Messages* потрібно постійно стежити. Це заощадить Ваш час.

Тепер створимо топологію цього резистора. Створіть новий шар: меню *Cell* → *New Cell*. Укажіть параметри відповідно до рис. 1.10 й натисніть «OK».

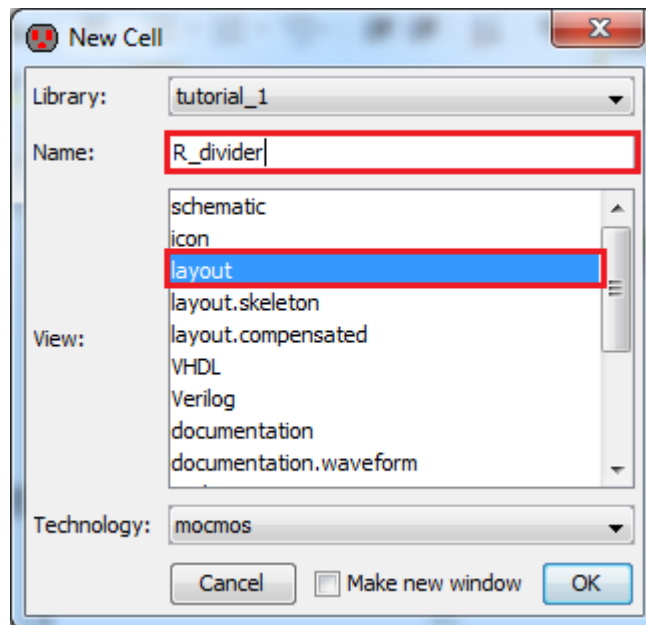


Рис. 1.10. Створення комірки топології

Ми створили групу комірок *R\_divider* (рис. 1.11). Натиснувши на знак "+" у закладці *Explorer* можна переглянути комірки групи *R\_divider*.

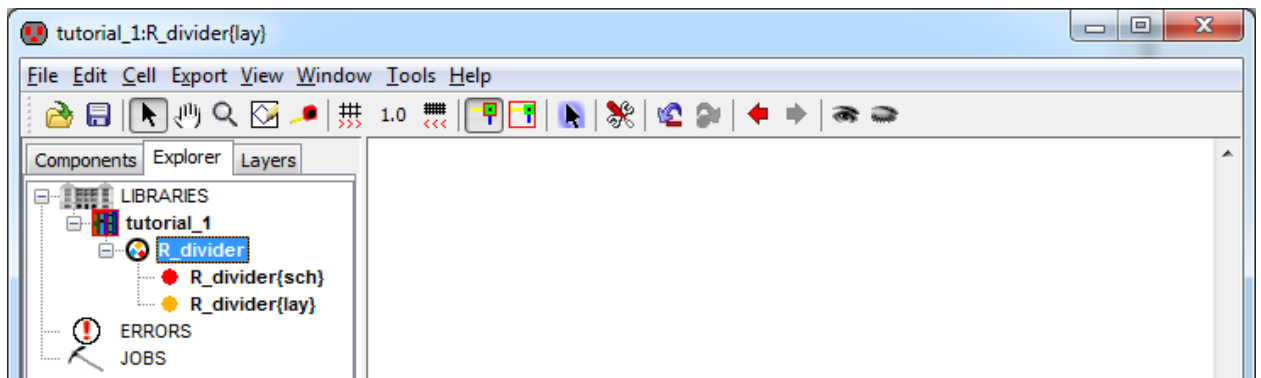


Рис. 1.11. Група комірок  $R\_divider$

У групі комірок діє система індикаторів, вона полегшує візуальне сприйняття комірок. Червоний круг біля назви означає схематичний макет комірки, жовтий – топологічний макет. Також, зверніть увагу на нижню частину екрана. Там виведена інформація про технологію й про норму проєктування. Праворуч вказуються координати розміщення курсору.

Виділіть в закладці *Explorer* шар  $R\_divider\{lay\}$  і перейдіть у закладку *Components*. У закладці *Components* (у лівому нижньому куті панелі *Components*) виберіть *N-Well Resistor* (рис. 1.12).

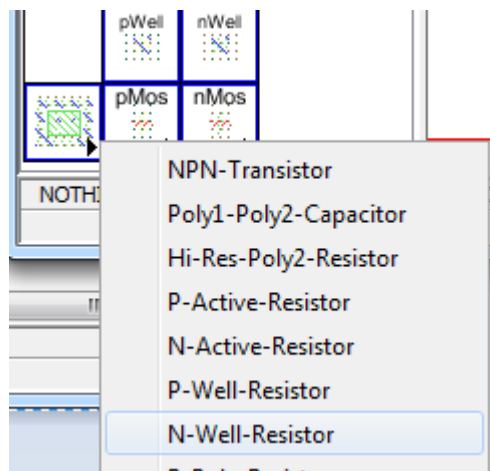


Рис. 1.12. Вибір зображення топології *N-Well* резистора

Якщо ж цей пункт недоступний, то Ви неправильно налаштували *Electric VLSI*. Перевірте, чи встановлена галочка в

*File* → *Preferences* → *Technology* → *Technology: Technology PROJECT Preferences: mosmos Technology* напроти слова *Analog* (див. рис. 1.2).

Установимо параметри створеного резистора. Для цього перейдіть у меню *Edit* → *Properties* → *Object Properties* або натисніть *Ctrl+I*. Встановіть значення *Length=187.5*; *Width=15*; *Well resistance =10k*. Натисніть «OK».

Відмасштабуйте вікно *Window* → *Fill Window* (*Ctrl+9*). У результаті, Ви отримаєте макет топології *N-Well* резистора (рис. 1.13).

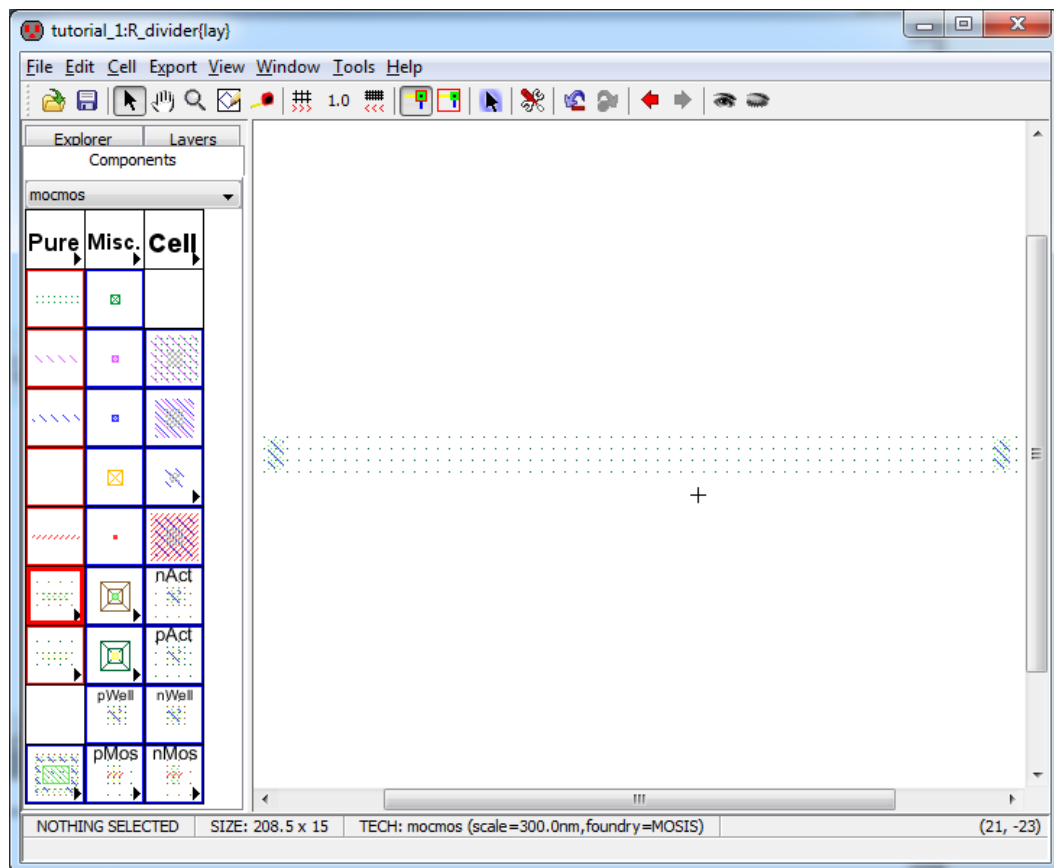


Рис. 1.13. Макет топології *N-Well* резистора

Перевіримо наш макет на наявність помилок у *DRC*: меню *Tools* → *DRC* → *Check Hierarchecally* або клавіша *F5*. В *Electric Messages* отримаємо повідомлення про відсутність помилок (рис. 1.14).

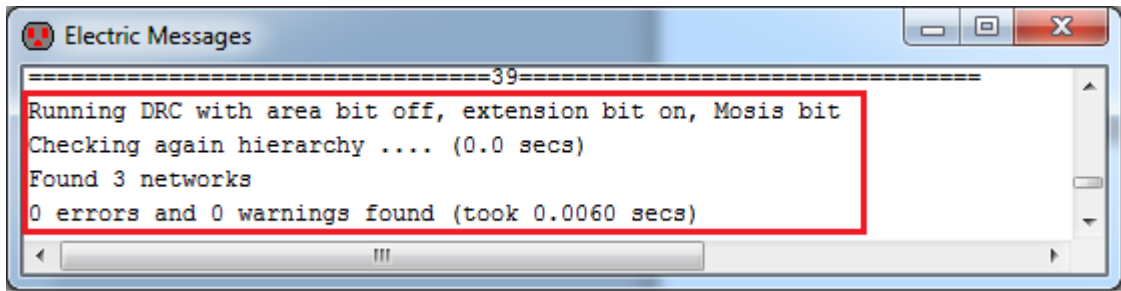


Рис. 1.14. Вікно повідомлень

Подивимося, що буде, якщо помилка все-таки буде допущена. Змінимо у властивостях резистора який-небудь параметр на неприпустиме значення. Натисніть *Ctrl+I* і змініть значення *Width=5*. Перевірте через *DRC (F5)*.

У результаті отримаємо повідомлення, що знайдено 3 помилки (рис. 1.15).

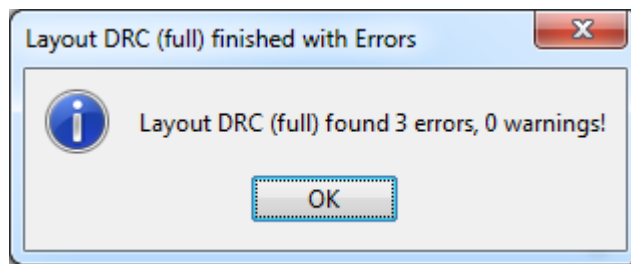


Рис. 1.15. Спливаюче вікно із повідомленням про виявлення помилок

В Electric Messages з'являться відповідні записи (рис. 1.16).

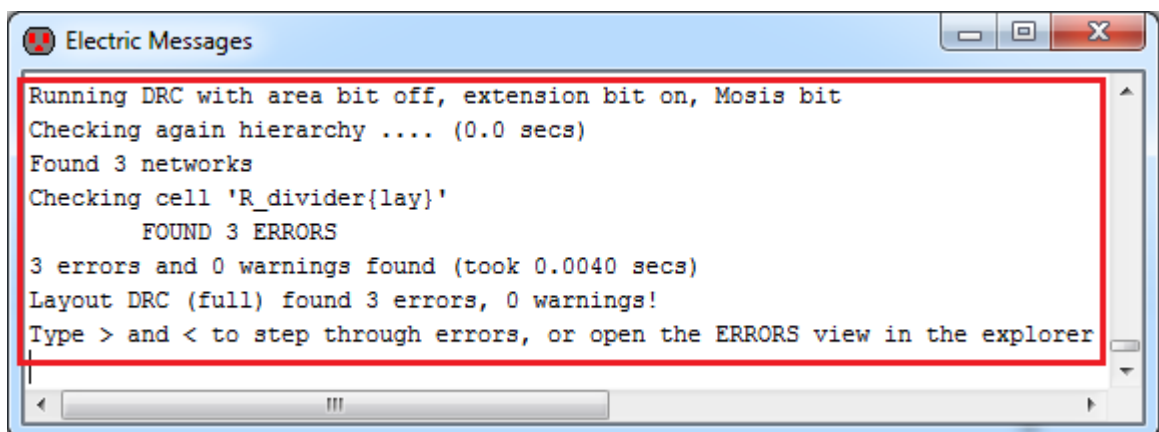


Рис. 1.16. Вікно повідомлень

Для того щоб побачити, що саме спричиняє помилку, потрібно перейти в закладку *Explorer* → *ERRORS* → *Layout DRC (full)[Current]*. Оскільки помилки

стосуються *R\_divider*, натискаємо «+» біля нього, і бачимо список помилок. Подвійне клацання по пункту списку виділить на макеті місце виникнення помилки (рис. 1.17).

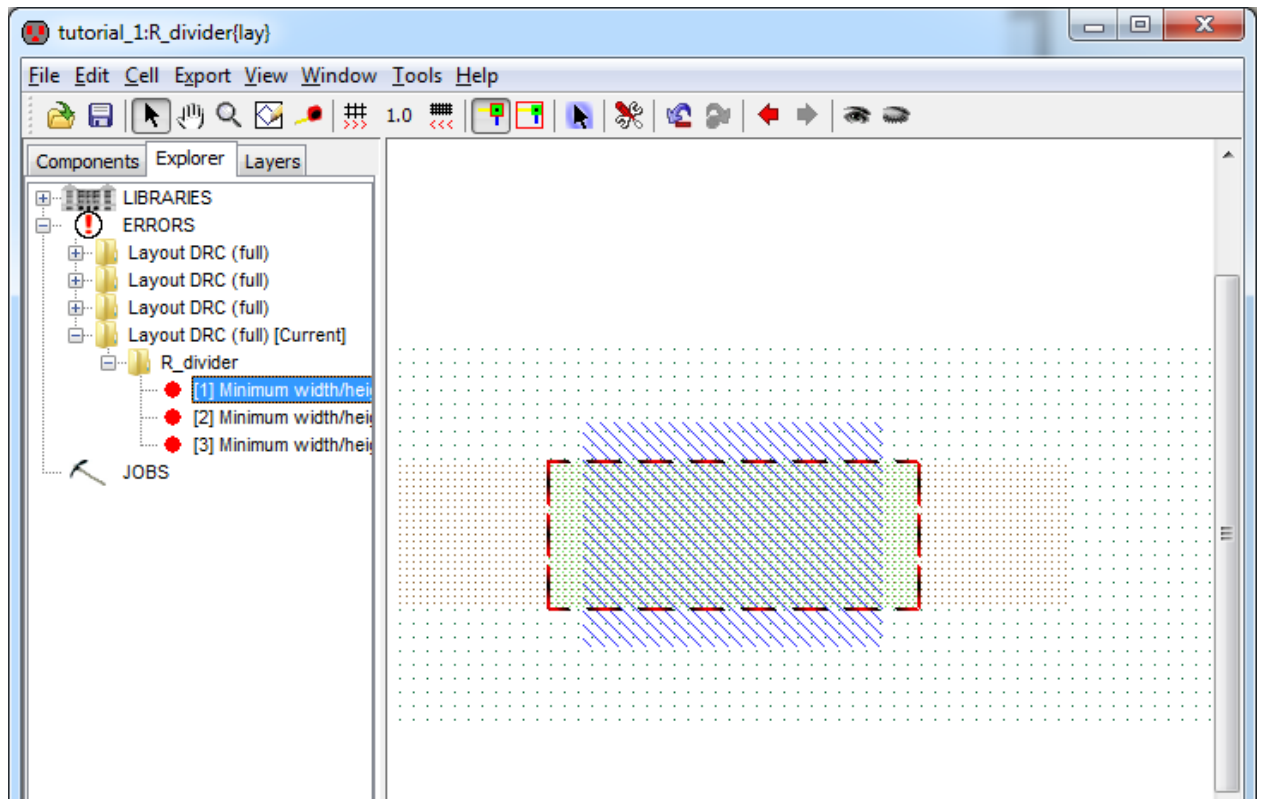


Рис. 1.17. Виділення помилки на макеті топології

Зверніть увагу на текст повідомлення помилки: «*Minimum width/height error((Y axis)): cell 'R\_divider{lay}' node N-Well-Resistor['resnwell@0'], layer 'N-Well' LESS THAN 12 WIDE (IS 5) [rule '1.1 Mosis, SUBM']*» – «Причина помилки: Ім'я комірки: вид вузла[‘Ім'я вузла’] та інформація про проєкт і технологію».

Виправимо помилку, і у властивостях резистора вкажемо коректне значення *Width=15*: виділимо резистор, натиснемо *Ctrl+I* і вкажемо *Width=15*. Також можна скористатися комбінацією клавіш *Ctrl+Z* для повернення на одну команду назад.

Знову, перевіримо через правила *DRC*, для того щоб бути впевненим у тому, що нічого зайвого ми, випадково, не встановили. Натискаємо *F5* і для коректного перегляду робочої області *Window* → *Fill Window*.

Тепер застосуємо інструмент *LVS* (*layout versus schematic*). Цей інструмент перевіряє узгодженість схематичного й топологічного макетів між собою. В *Electric VLSI* він використовується в інструменті *NCC* (*Network Consistency Checking*). Вибираємо *Tools* → *NCC* → *Schematic and Layout Views of Cell in Current Window*. Ніяких помилок не має бути (рис. 1.18).

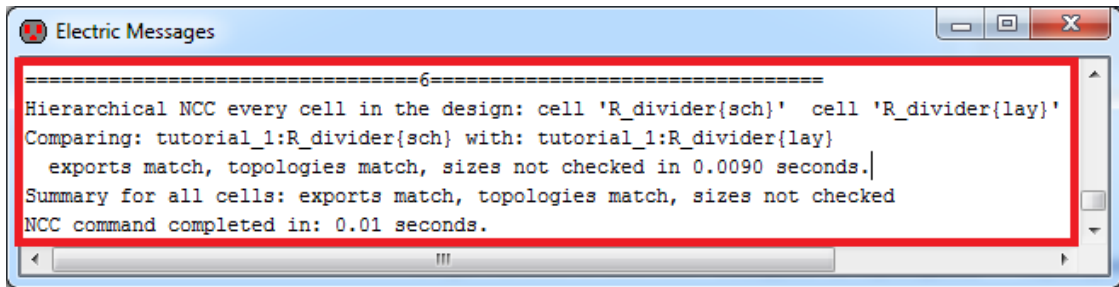


Рис. 1.18. Вікно повідомлень

Можна встановити «гарячі клавіші» для *NCC*, тому що його використання буде достатньо частим. Для цього вибираємо в меню *File* → *Preferences* → *General* → *Key Bindings*, далі в списку, що розкривається, *Tools* → *Tools|Ncc* → *Tools|Ncc|schematic and Layout Views of Cell in Current Window*. Тепер в області *Shortcuts*: потрібно натиснути *Add*, і у вікні, що з'явилося, указати клавішу або комбінацію клавіш, після натискання яких ми бажаємо щоб включався *NCC*. Нехай це буде клавіша «*L*».

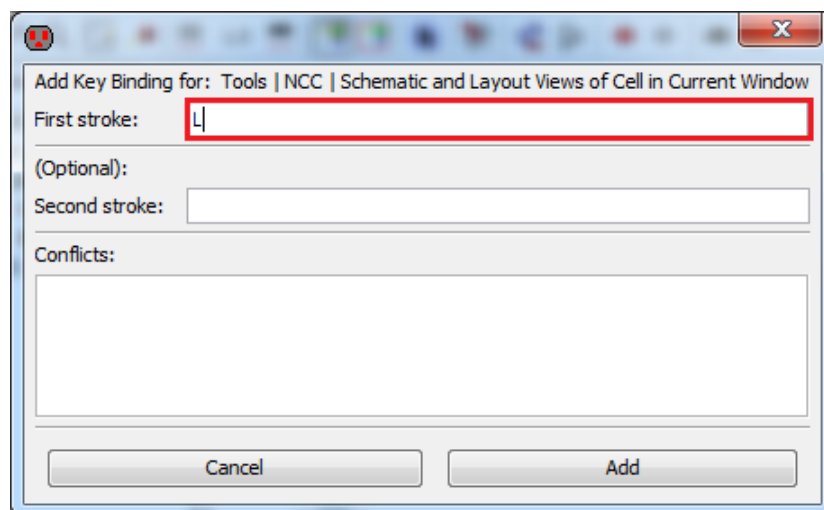


Рис. 1.19. Встановлення «гарячої клавіші» для *NCC*



Якщо на різні інструменти, або команди вказати однакову клавішу або комбінацію клавіш, то це призведе до некоректної роботи програми. Звісно *Electric* виведе повідомлення з попередженням про конфлікт.

Перед подальшим проектуванням подільника напруги потрібно поговорити про контакт *n*-кармана й *p*-підкладки. У даній роботі ми проектуємо макет за технологією *C5* з *n*-карманами й заземленою *p*-підкладкою (це є загальним для всіх *NMOS* пристроїв). В *Electric VLSI* реалізований інструмент *ERC* (*electrical rule checks*), який перевіряє заземлення *p*-карманів (у нашому випадку це *p*-підкладка) і контакт *n*-карманів до живлення. Крім того, якщо макет містить тільки цифрові схеми, то *n*-кармани бажано приєднувати до живлення (*VDD*).

Скористаємося інструментом *ERC*: перейдемо в меню *File* → *Preferences* → *Tools* → *Well Check* (рис. 1.20) і перевіримо, щоб у налаштуваннях для *n*-карманів було відмічено «*Must connect to Power*».

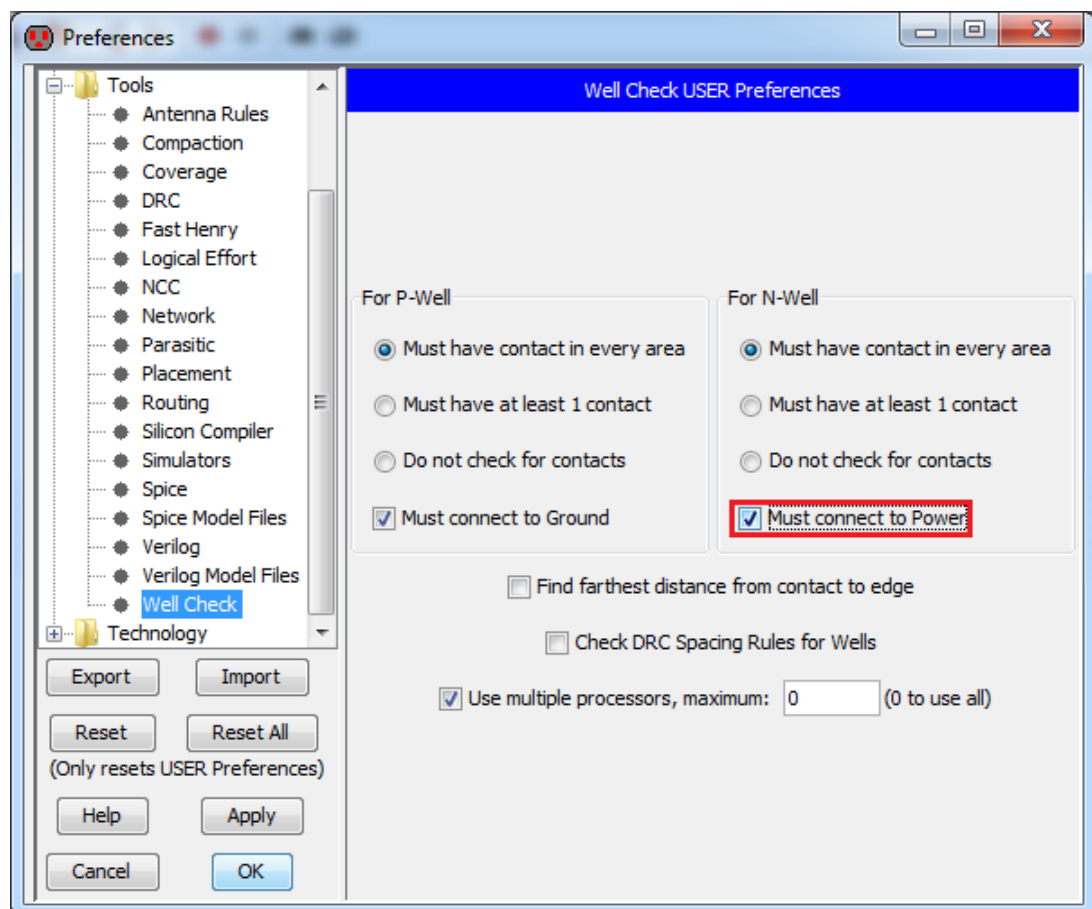


Рис. 1.20. Налаштування підключення *n*-карманів до джерела живлення

Зазвичай, коли звертаються до інструмента *ERC*, то прагнуть з'ясувати наявність контактів у кожній області й переконатися в тому, що *p*-підкладка (*p*-карман) прив'язана до землі. Однак у даній ситуації ми прагнемо перевірити, що *n*-карман прив'язаний до живлення.

Запустимо *Tools* → *ERC* → *Check Wells*. З'явиться вікно із повідомленням про помилки (рис. 1.21).

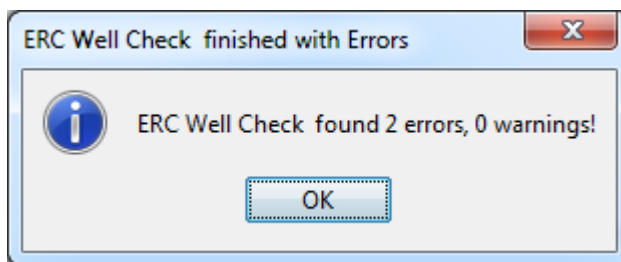


Рис. 1.21. Спливаюче вікно із повідомленням про виявлення помилок

Помилки обумовлені тим, що за попередньо установленими налаштуваннями пристрій є аналоговим (*File* → *Preferences* → *Technology* → *Technology* → *Technology Project Preferences: mosmos Technology* налаштування *Analog*). Тому *n*-карман не має контактувати з живленням. Приберемо галочку *Must connect to Power* в *File* → *Preferences* → *Tools* → *Well Check: For N-Well*. Якщо з'явиться вікно з питанням, чи застосувати налаштування до всіх відкритих бібліотек, то потрібно буде погодитися. Запустимо перевірку *ERC*. Тепер помилок не має бути.

Створимо другий резистор. Перейдемо у вкладку *Explorer* і в групі комірок *R\_divider* виберемо макет схеми (Вкладка *Explorer* → розкритий список *Libraries* → *tutorial\_1* → *R\_divider* → *R\_divider{sch}*). Оскільки нам потрібний такий же резистор, то просто виділимо його, скопіюємо, нажавши *Ctrl+C*. Для того, щоб його вставити, клацніть мишкою в будь-якій точці області рисунку, натисніть *Ctrl+V* і вкажіть місце розташування нового елемента. У результаті одержимо два однакові резистори (рис. 1.22).

Для того щоб розгорнути резистор для комутації переходимо в меню *Edit* → *Rotate* → *90 Degrees Counterclockwise* (або *Ctrl+J*) (рис. 1.23).

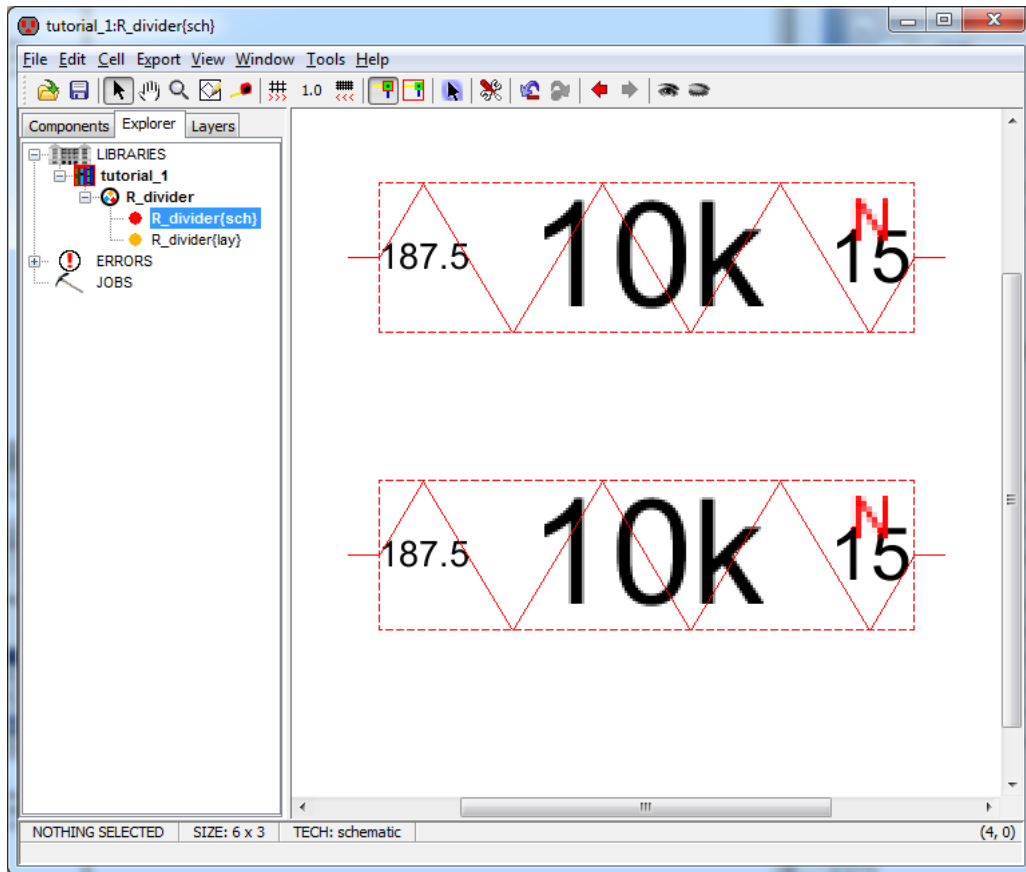


Рис. 1.22. Копіювання елемента

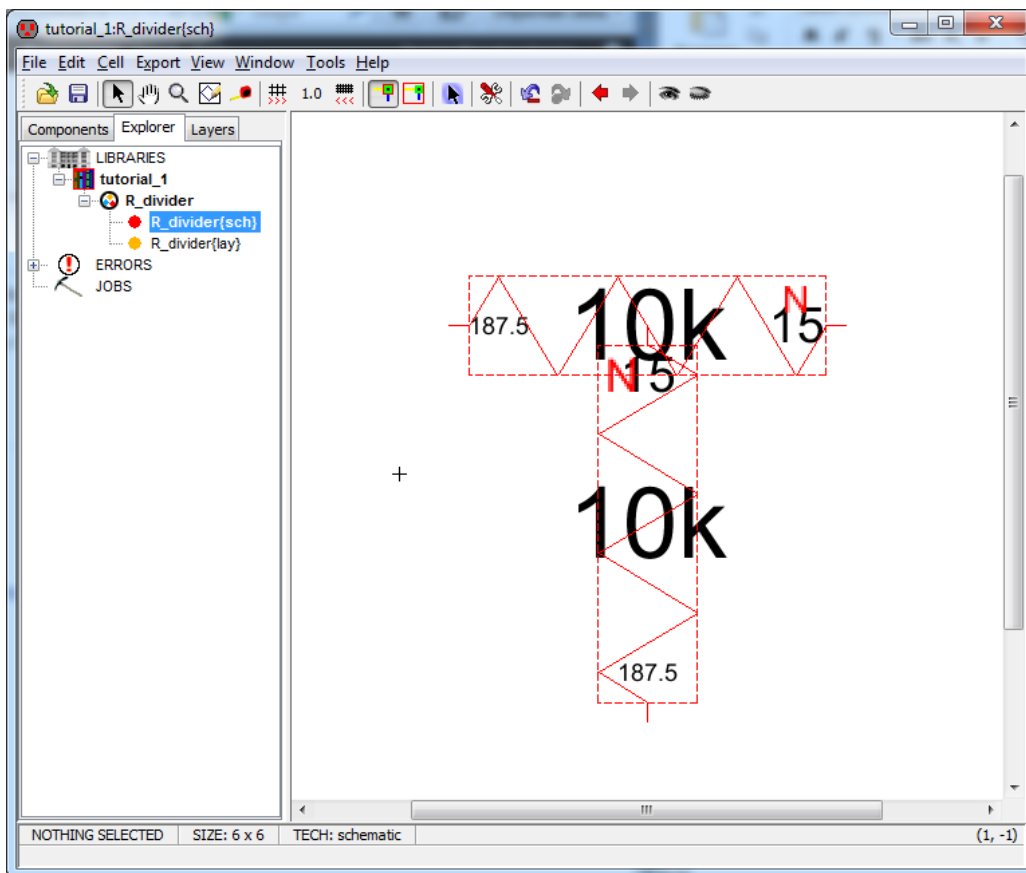


Рис. 1.23. Поворот резистора на 90°

Перемістимо резистори так, щоб горизонтально орієнтований вузол був вище вертикально орієнтованого вузла. Виділимо один з резисторів. Наведемо курсор на порт резистора й із затиснутою правою кнопкою мишки потягнемо убік. З'явиться з'єднання (точка з'єднання), яке позначається квадратом з жирною крапкою усередині (рис. 1.24), далі ведемо з'єднання до порту другого резистора вниз (рис. 1.25).

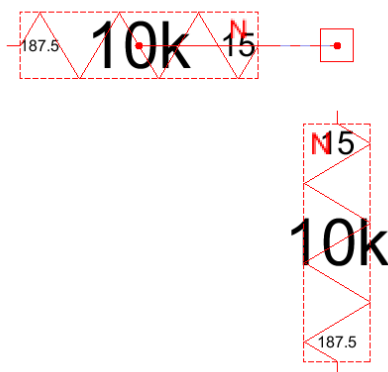


Рис. 1.24. Встановлення точки з'єднання

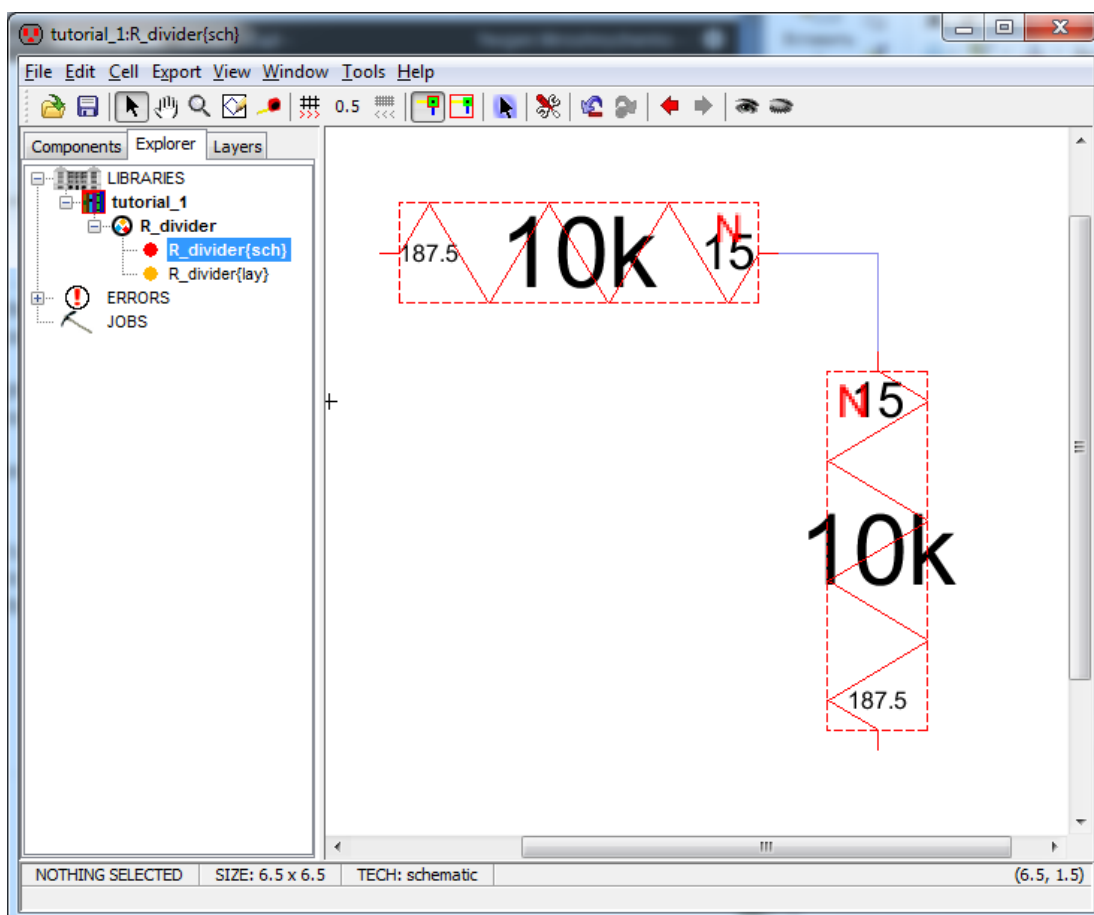


Рис. 1.25. Електричне з'єднання двох резисторів

Перевіримо схему на наявність помилок (F5). Якщо виявляється помилка, то найчастіше через зайве з'єднання. Для автоматичного видалення зайвих з'єднань можна скористатися меню *Edit* → *Cleanup Cell* → *Cleanup Pins*.

Додамо ще пару з'єднань, як показано на рис. 1.26. Потрібно клацнути лівою кнопкою миші для вибору порту резистора й правою – для протягання з'єднання. Сині лінії, що з'явилися при цьому, – це з'єднувальні проводи, які ми будемо називати дугами.

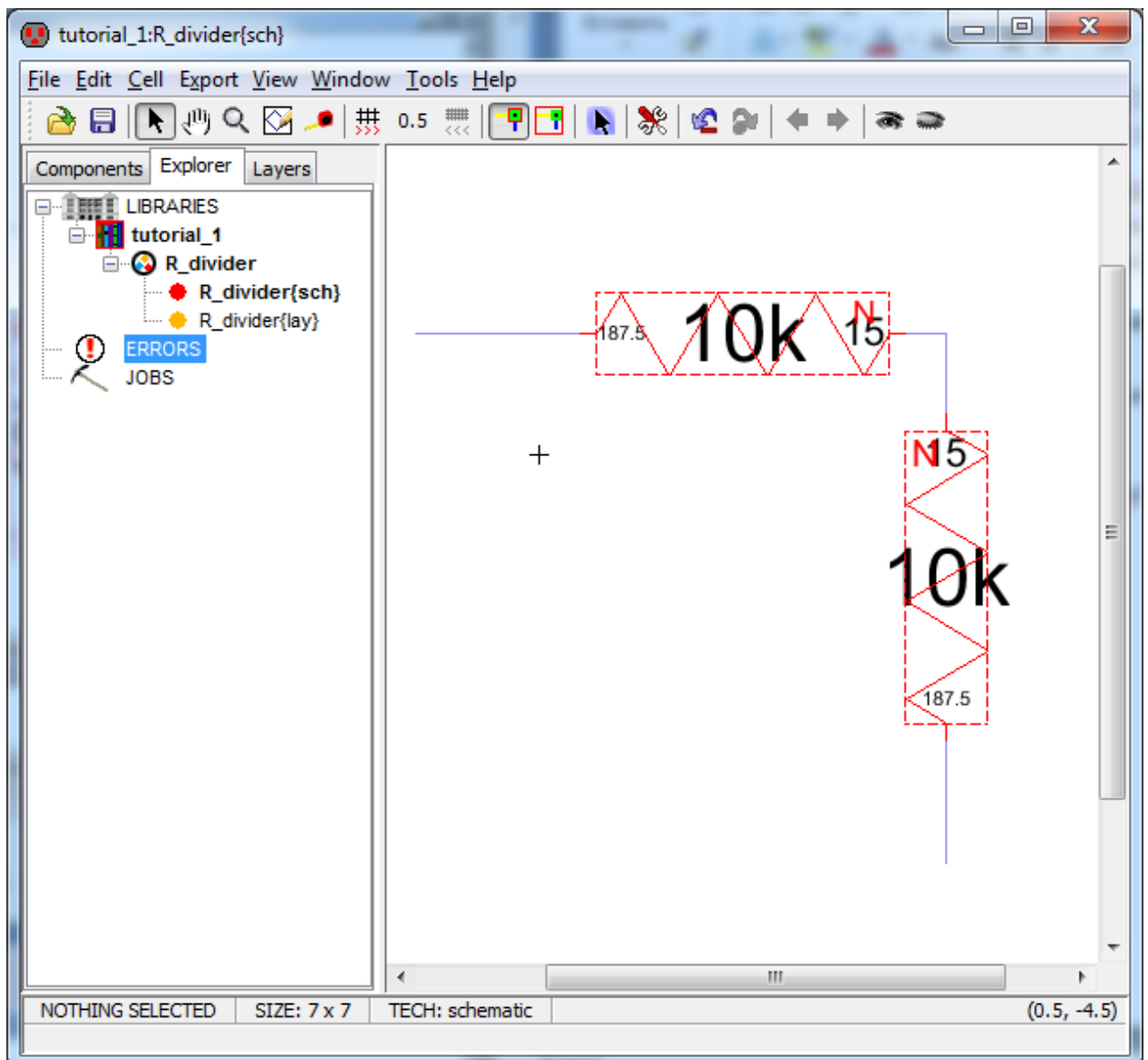


Рис. 1.26. Створення дуг

Назвемо дуги, для цього клацніть двічі мишкою по дузі або виділіть дугу й натисніть *Ctrl+I*. У вікні, що відкрилося, у полі *name* напишіть назву

дуги відповідно до рис. 1.27. Зверніть увагу, що в назві використовуються малі літери.

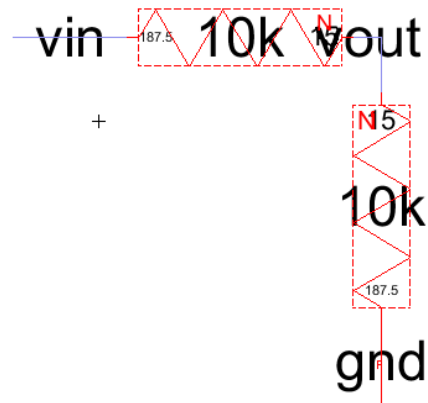


Рис. 1.27. Називання дуг

Для більш естетичного вигляду посунемо назви. У результаті отримаємо схему подільника напруги (рис. 1.28).

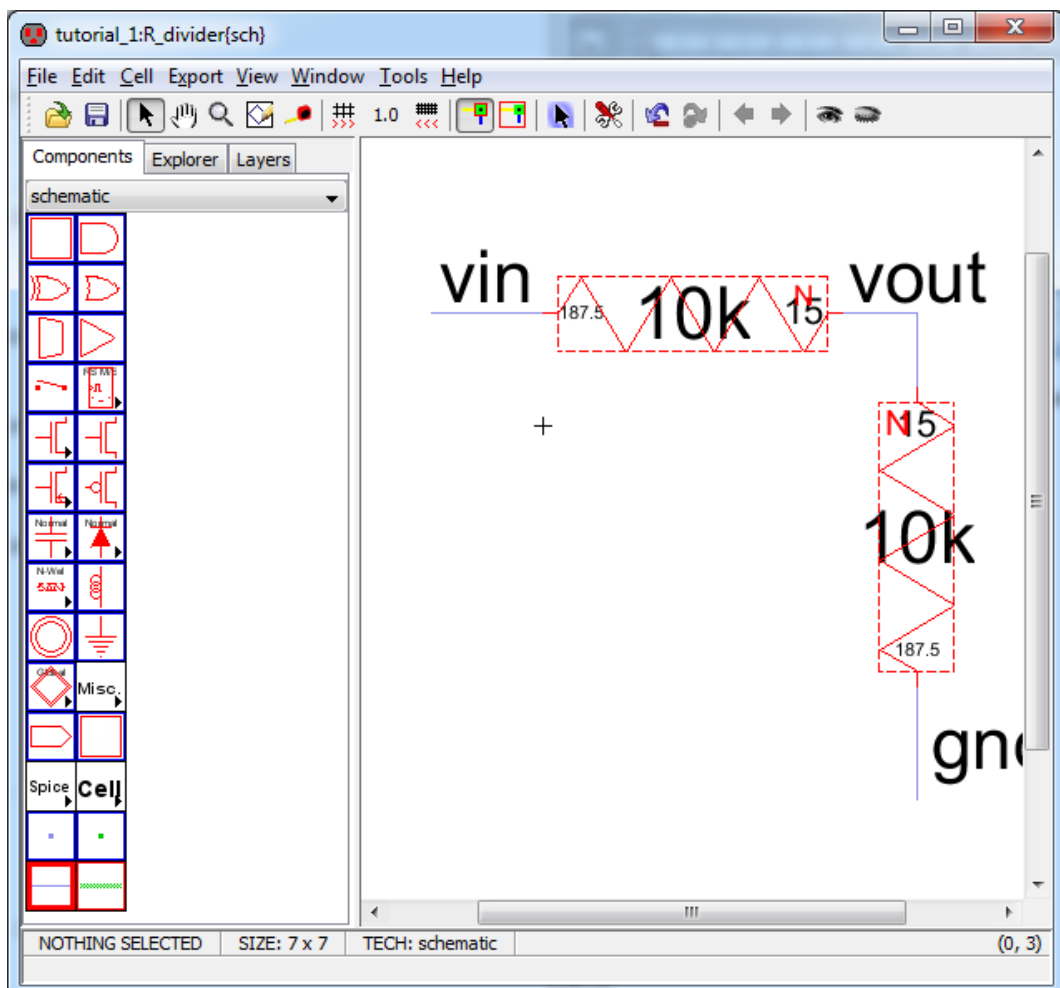


Рис. 1.28. Схема подільника напруги

Додамо *Spice* код. Перейдемо у вкладку *Components* → *Misc* → *Spice code*. Мишкою в робочій області вкажемо місце для напису. Відкриємо властивості *Spice* коду: виділимо його й натиснемо *Ctrl+I*. У вікні, що з'явилося, укажемо в полі «*Spice code*» текст (попередньо включивши *Multi-line Text*) (рис. 1.29):

```
vin vin 0 DC 1  
.tran 0 1
```

Цей текст визначає джерело вхідної напруги й початкові дані для аналізу перехідних процесів у програмі *LTspice*.

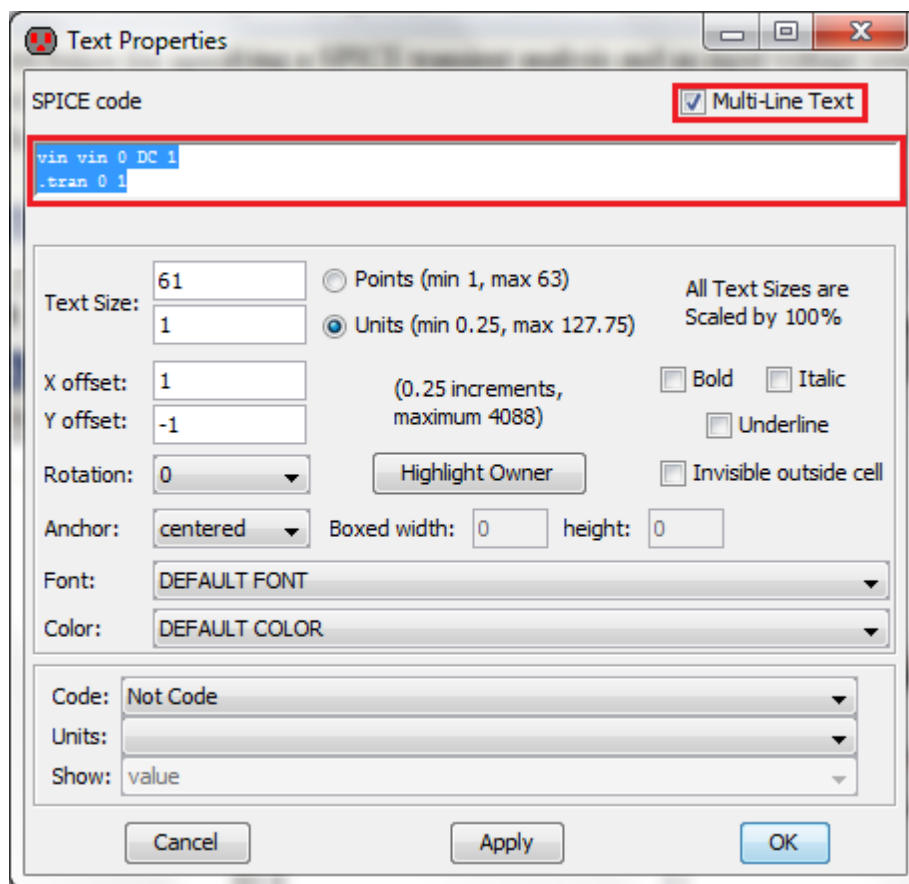


Рис. 1.29. Додавання *Spice* коду

Джерело вхідної сталої напруги описується командою:

```
vname n1 n2 dc val
```

«*vname*» – це назва джерела, а буква «*v*» перед цією назвою вказує, що це джерело напруги; «*n1*» і «*n2*» – вузли підключення (*n1* – це «+вузол», *n2* – це

«–вузол»); «*dc*» указує на те, що джерело напруги стало; «*val*» – номінал, видаваний джерелом при аналізі на сталому струмі.

Таким чином, рядок *Spice* коду «*vin vin 0 DC 1*» означає, що ми підключили джерело сталої напруги (*v*) з іменем «*in*» між вузлами «*vin*» і «нульовим». Номінал джерела дорівнює 1 В.

Команда *.tran* керує аналізом перехідних процесів:

*.tran vstep vstop*

«*vstep*» – часовий інтервал прирісту; «*vstop*» – кінцевий час аналізу.

Отже, рядок *Spice* коду «*.tran 0 1*» означає, що ми задаємо безперервний аналіз (*vstep* = 0) протягом 1 сек.

Таким чином, після додавання *Spice* коду робоче вікно програми буде виглядати як на рис. 1.30.

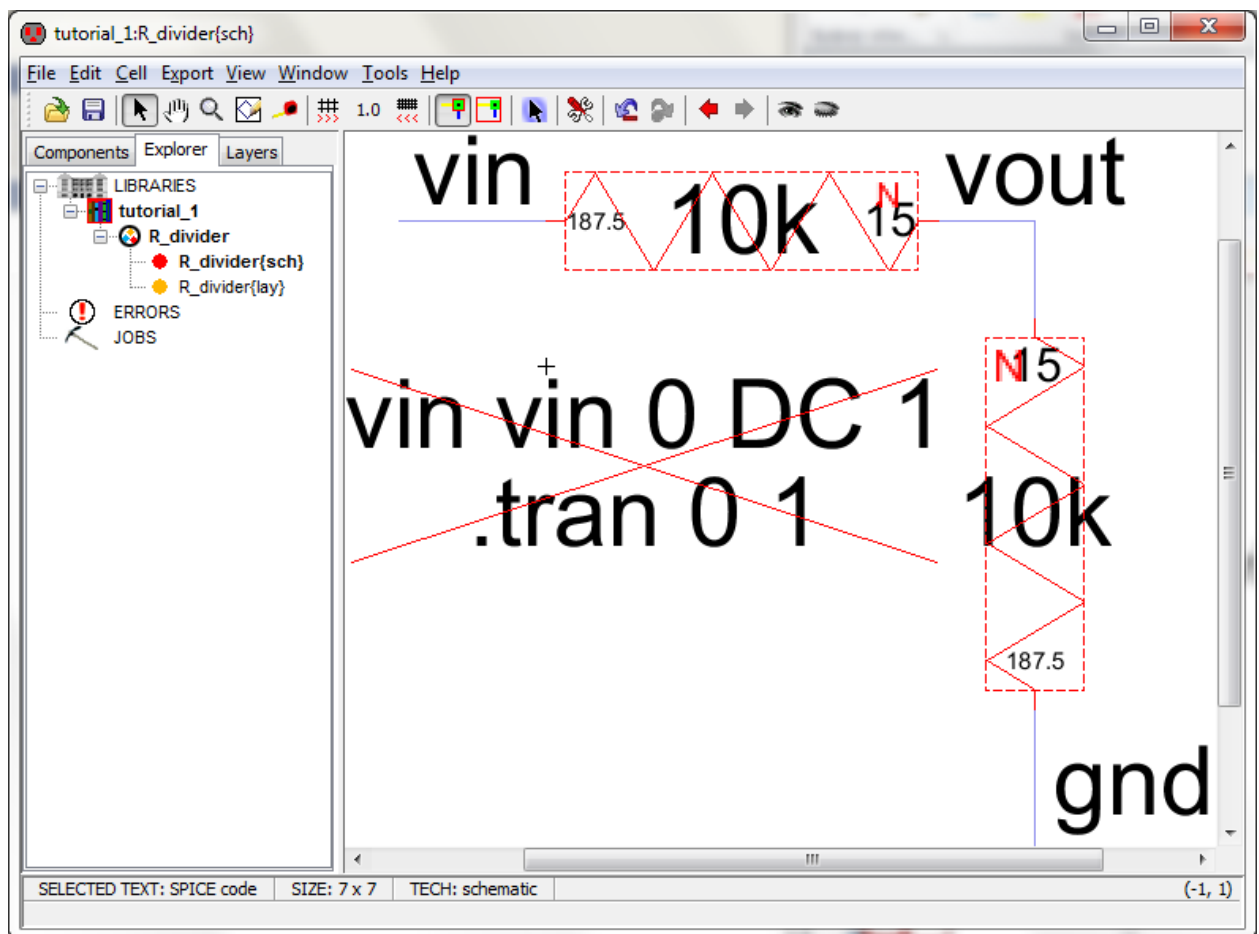


Рис. 1.30. Схема подільника напруги із *Spice* кодом



Перевіримо схему на помилки ( $F5$ ) і збережемо бібліотеку.

Для продовження роботи необхідно щоб на ПК був встановлений *LTspice. Electric VLSI* має бути настроєний на спільну роботу з *LTspice*.

Проведемо моделювання роботи схеми, перейдемо в меню *Tools* → *Simulation (Space)* → *Write Spice Deck*.

Відкриється вікно *LTspice* (рис. 1.31).

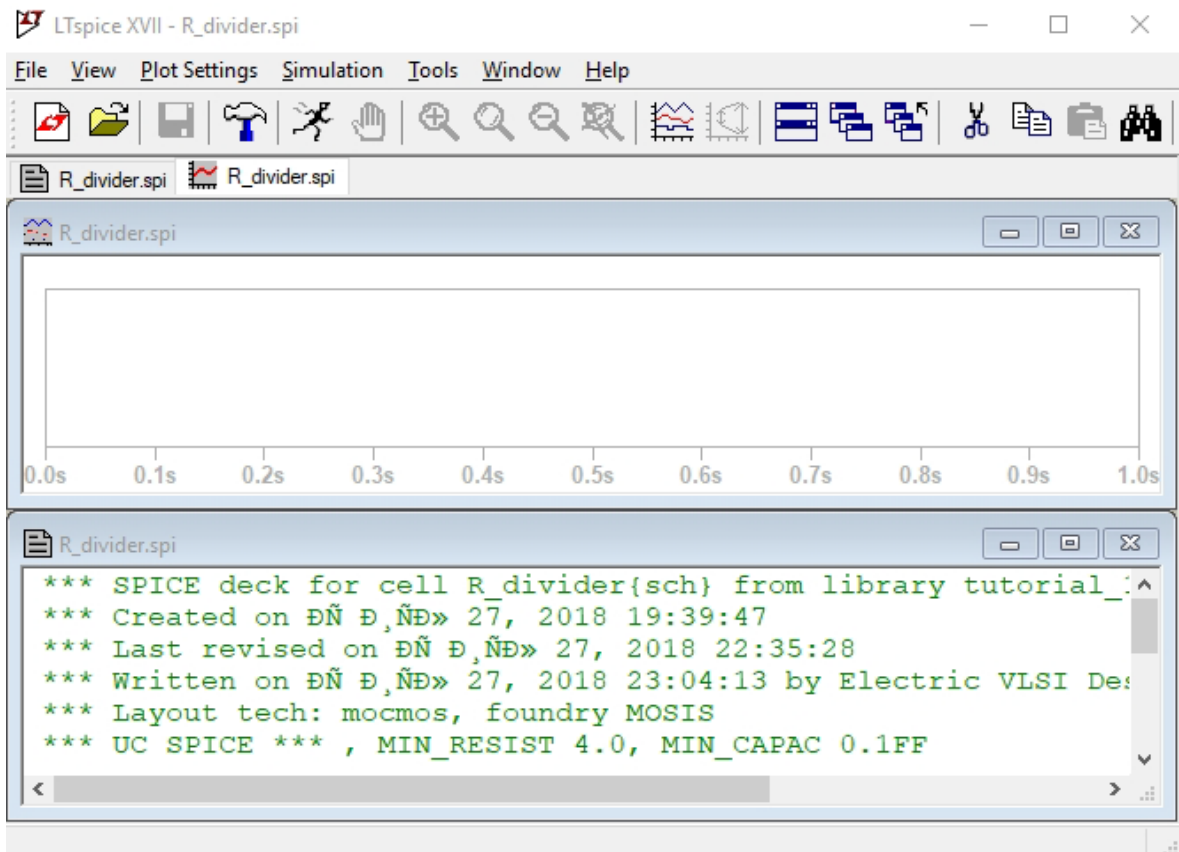


Рис. 1.31. Вікно *LTspice*

Натисніть на іконку, зазначену на рис. 1.32, для відображення доступних для побудови графіків сигналів.

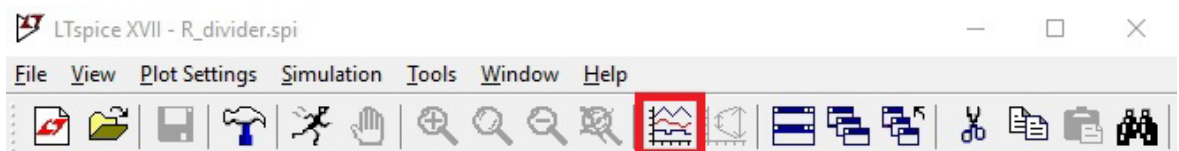


Рис. 1.32. Панель інструментів *LTspice*

Зі списку доступних сигналів оберіть  $V(vin)$  і  $V(vout)$  (рис. 1.33). Для того, щоб вибрати кілька пунктів (сигналів), затисніть клавішу *Ctrl*.

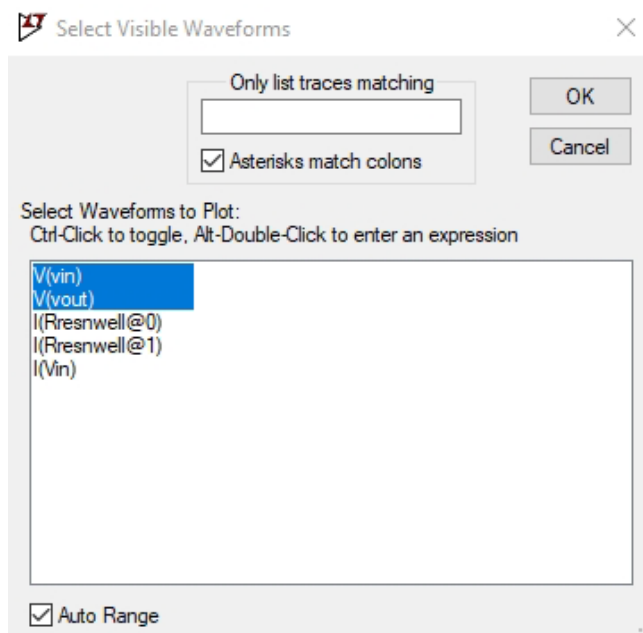


Рис. 1.33. Вікно вибору доступних сигналів

Натисніть «OK». У результаті отримаємо графіки зазначені по нашому *Spice* коду (рис. 1.34).

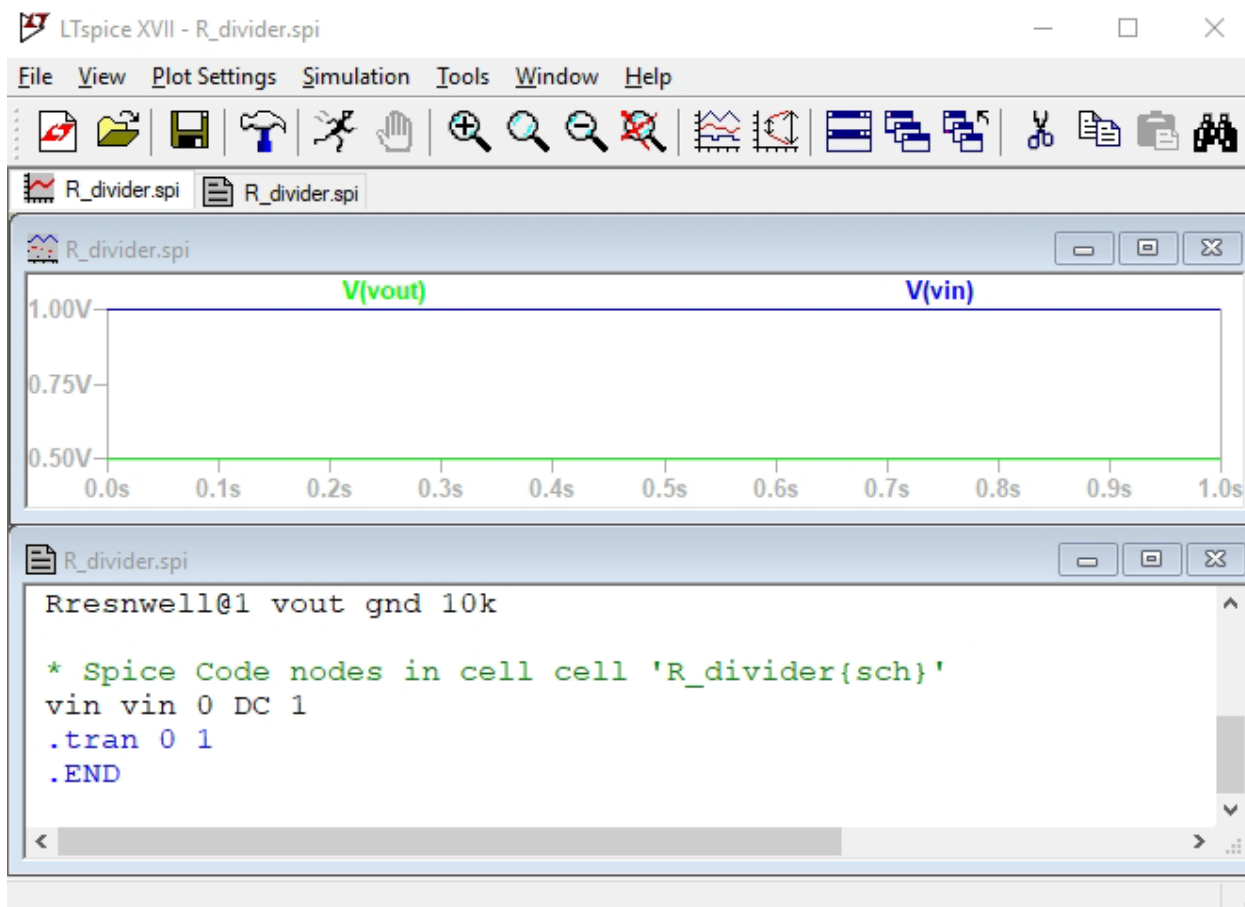


Рис. 1.34. Графіки роботи подільника напруги в *LTspice*

Симулятор *LTspice* дуже корисний і зручний інструмент, але крос-зондування між зондами *Electric VLSI* та *LTspice* не підтримується.

Для побудови в *Electric VLSI* пророблених результатів з *LTspice*, необхідно переконатися, що в *LTspice* при закритті автоматично не видаляються \*.raw файли. Для цього в *LTspice* перейдемо в меню *Tools* → *Control Panel* → *Operation* і перевіримо, щоб була знята галочка напроти параметра «*Automatically delete .raw files [\*]*» (рис. 1.35).

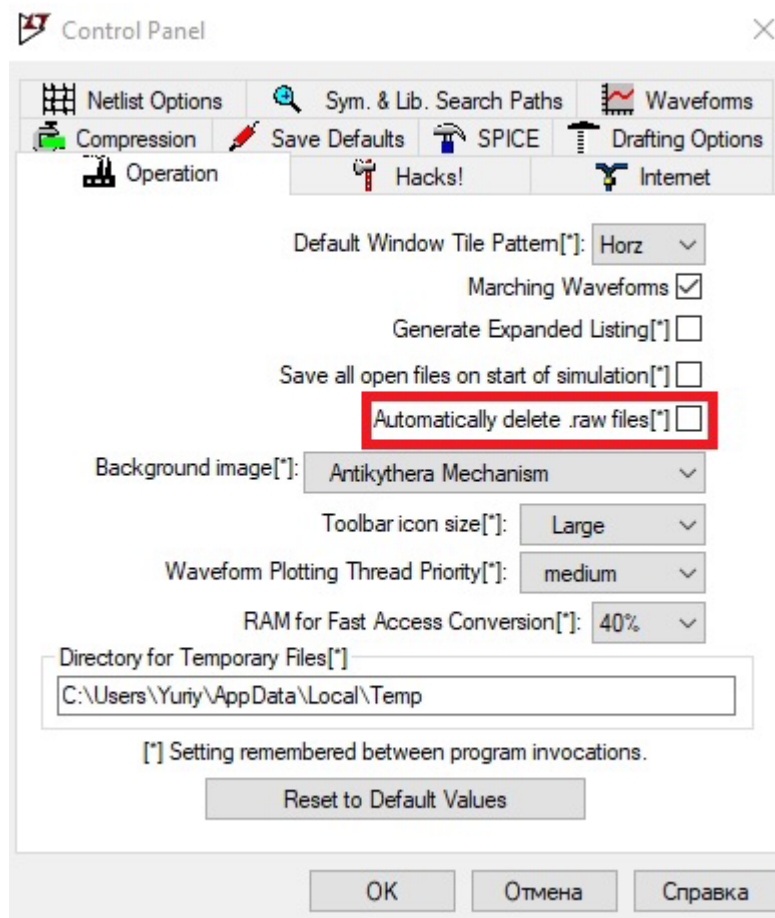


Рис. 1.35. Вікно налаштувань *LTspice*

Після цього закриваємо *LTspice*. Далі автоматично запускається зонд *Electric VLSI*. У закладці *Explorer* у списку, що розкривається, оберіть подвійним клацанням лівої кнопки миші сигнали для відображення (*vin* та *vout*). На екрані з'являться графіки роботи подільника напруги (рис. 1.36).

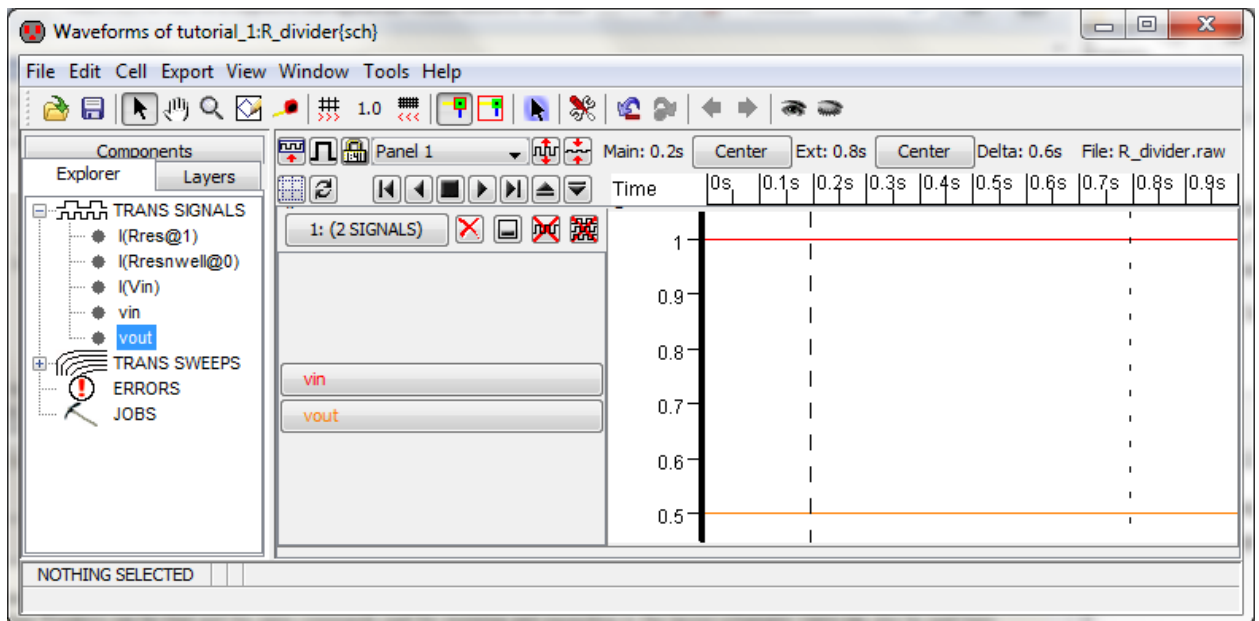


Рис. 1.36. Графіки роботи подільника напруги в *Electric VLSI*

Вікно *Waveforms of tutorial\_1:R\_divider{sch}* згортаємо або закриваємо.

Дорисуємо топологію резистивного подільника. Відкриваємо топологію нарисованого нами резистора, для цього в закладці *Explorer* у списку переходимо в *LIBRARIES* → *tutorial\_1* → *R\_divider* → *R\_divider{lay}*. Виділяємо комірку з резистором (натиснути потрібно на контактну область) і копіюємо її, меню *Edit* → *Copy* (*Ctrl+C*). Тепер зніміть виділення з резистора (клацніть десь по робочій області лівою кнопкою мишки) і вставте скопійований резистор: меню *Edit* → *Paste* (*Ctrl+V*). У результаті з'явиться рамка, яка вказує на межі нового резистора, укажіть мишкою місце розташування нового резистора. Для точного його розміщення використовуйте стрілки клавіатури або вкажіть точне значення його координат у властивостях, викликавши їх через меню *Edit* → *Properties* → *Object Properties*.

Пункт на панелі інструментів, обведений на рис. 1.37, вказує на розмір сітки вирівнювання. Він повинен мати значення 0,5.

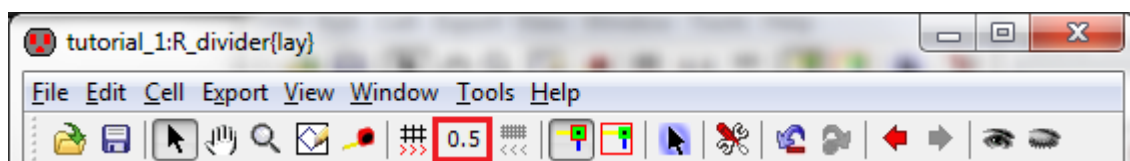


Рис. 1.37. Панель інструментів *Electric VLSI*

Запустимо *DRC* перевірку (*F5*). Можливо, ми отримаємо повідомлення про помилку (рис. 1.38, 1.39).

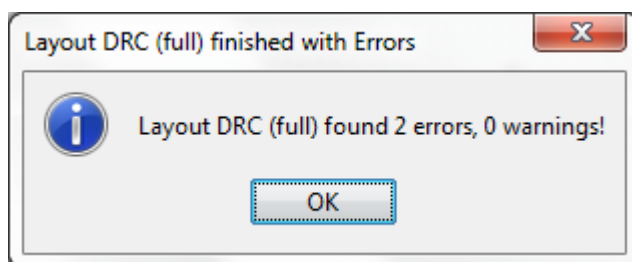


Рис. 1.38. Спливаюче вікно із повідомленням про виявлення помилок

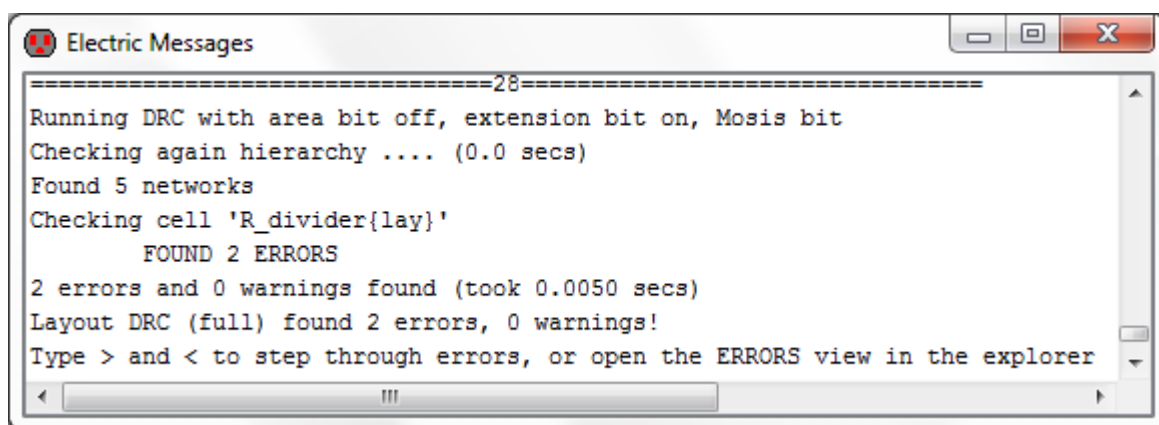


Рис. 1.39. Вікно повідомлень

Ці помилки обумовлені малою відстанню між *n*-комірками. Для того, щоб візуально побачити причину помилки, перейдемо в закладку *Explorer* → *ERRORS* → *Layout DRC (full) [Current]* або комбінація клавіш *Shift+>/<*. Далі йде перерахування комірок, де спостерігаються помилки (у нашому випадку це комірка *R\_divider*), і в цьому списку зазначені помилки. Двічі клацаємо по пункту списку помилок комірки *R\_divider* і у робочому вікні бачимо, що навколо резисторів з'явився мерехтливий штриховий контур, який вказує на місце помилки, а у вікні *Electric Messages* відобразився більш детальний опис помилки:

```
Notch (layer 'N-Well'): node N-Well-Resistor['resnwell@1'] LESS (BY 2.5)
THAN 6 TO node N-Well-Resistor['resnwell@0'] [rule '1.3 Mosis']
```

Збільшуйте відстань між резисторами доти, поки повідомлення про помилку перестануть з'являтися.

Оберемо правий порт верхнього резистора, клацнувши по ньому лівою кнопкою мишки. Тепер затисніть праву кнопку мишки й протягніть курсор до правого порту нижнього резистора, поки він (правий порт нижнього резистора) не виділиться, тепер відпустіть кнопку мишки. Ми тільки що створили контакт між резисторами. Зробіть також металеві виводи від лівих портів резистивного подільника так, як показано на рис. 1.40.

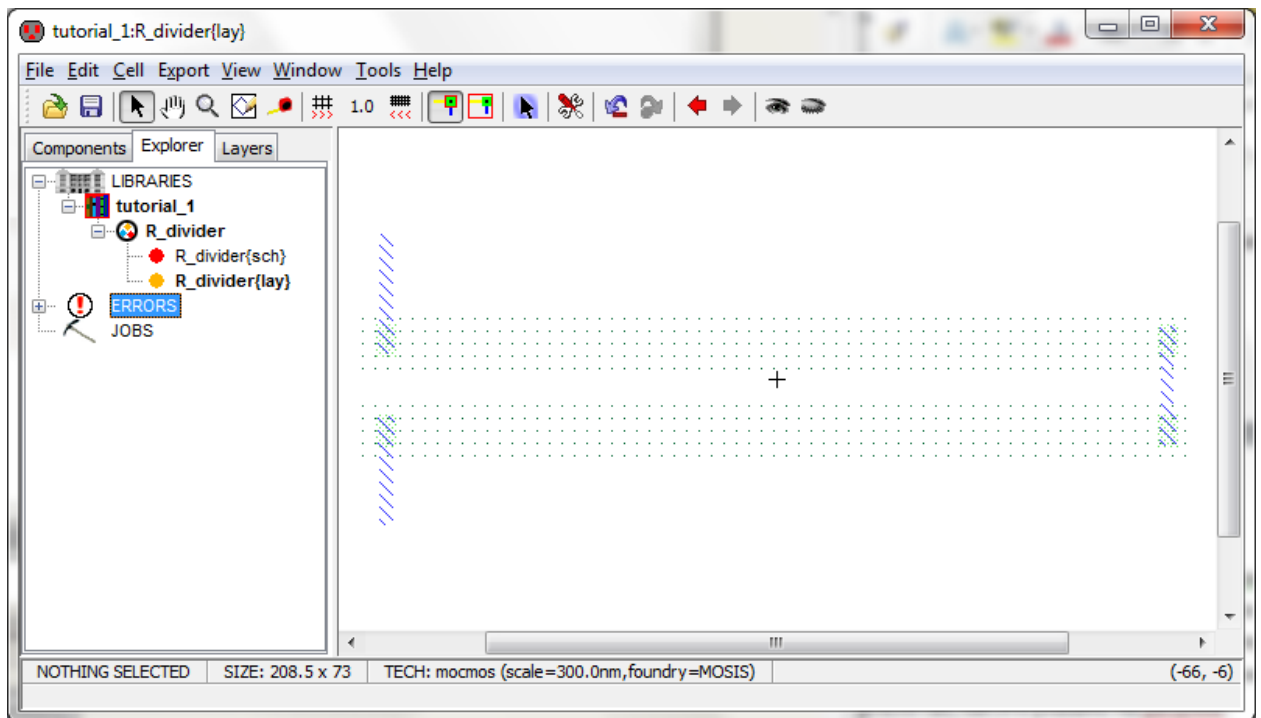


Рис. 1.40. Топологія подільника напруги із металевими виводами

Намагайтеся якнайчастіше перевіряти проєкт на наявність помилок. Перевіримо макет на наявність схематичних помилок (*F5*).

Змініть ширину металевих дуг на значення 4. Для цього потрібно виділити металеву дугу (для виділення декількох об'єктів використовуйте клавішу *Shift*), викликати вікно меню *Edit* → *Properties* → *Object Properties* або *Ctrl+I*. У рядку *Width*: указати потрібне значення. Не забуваємо перевіряти макет на помилки (*F5, L*).

Промаркуємо топологію так, як маркували схему. Для того, щоб позначити дуги потрібно зайти в їхні властивості *Ctrl+I* (або двічі клацнути по дузі) і змінити назву кожної дуги. Дуга, яка з'єднує два резистори, має називатися *vout*. Ми перейменували кожну дугу, але оскільки розмір

маркувань дуже малий, ми його не бачимо (рис. 1.41). Якщо збільшимо зображення, то побачимо наші маркування.

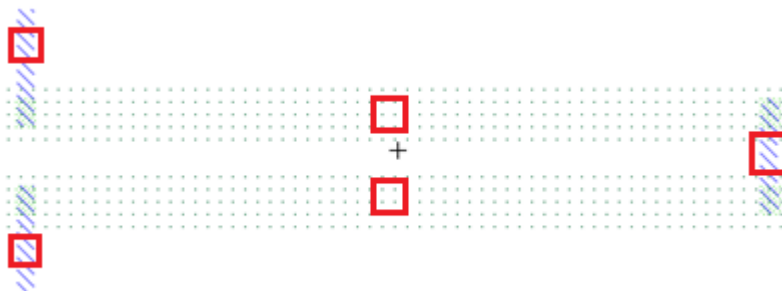


Рис. 1.41. Топологія подільника напруги із маркованими дугами

Виділимо напис. Ви можете збільшувати картинку доти, поки з написом буде зручно працювати, і потім його виділити, або ж виділити його, клацнувши приблизно по маркуванню із затиснутою клавішею *Ctrl*. Причому якщо треба змінити назву групи об'єктів, то затискаємо *Ctrl+Shift* і клацаючи лівою кнопкою мишки вибираємо маркування. Викликаємо властивості маркувань *Ctrl+I* і у поле *Unit Size*: вказуємо розмір шрифту, у нашому випадку – 10 (рис. 1.42).

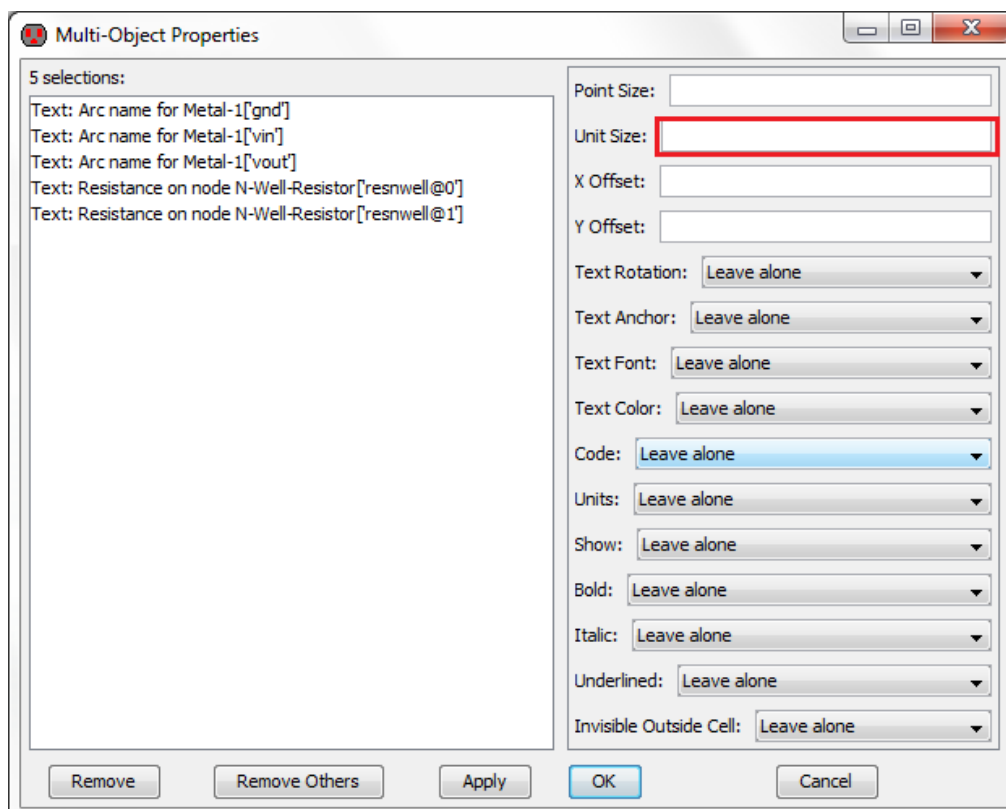


Рис. 1.42. Встановлення розміру шрифту маркувань

У результаті отримаємо збільшені (видимі) маркування дуг (рис. 1.43).

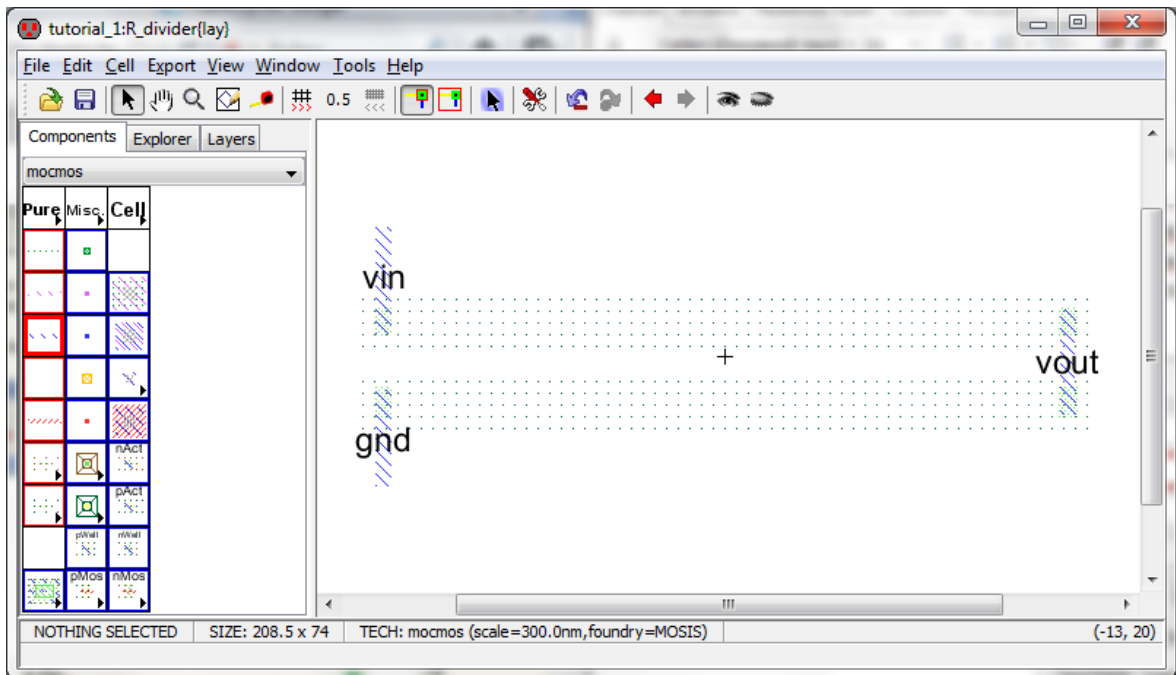


Рис. 1.43. Топологія подільника напруги із маркованими дугами

Перевіримо макет на наявність помилок *DRC (F5)* та *NCC (L)*.

Вставимо в макет *spice* код. Скопіюємо його із комірки, де зображена схема подільника, і вставимо в топологію (*Ctrl+C / Ctrl+V*). Може знадобитися змінити розмір шрифту відображення *spice* коду. Також змініть розміри маркування резисторів. У результаті топологія має виглядати, як на рис. 1.44.

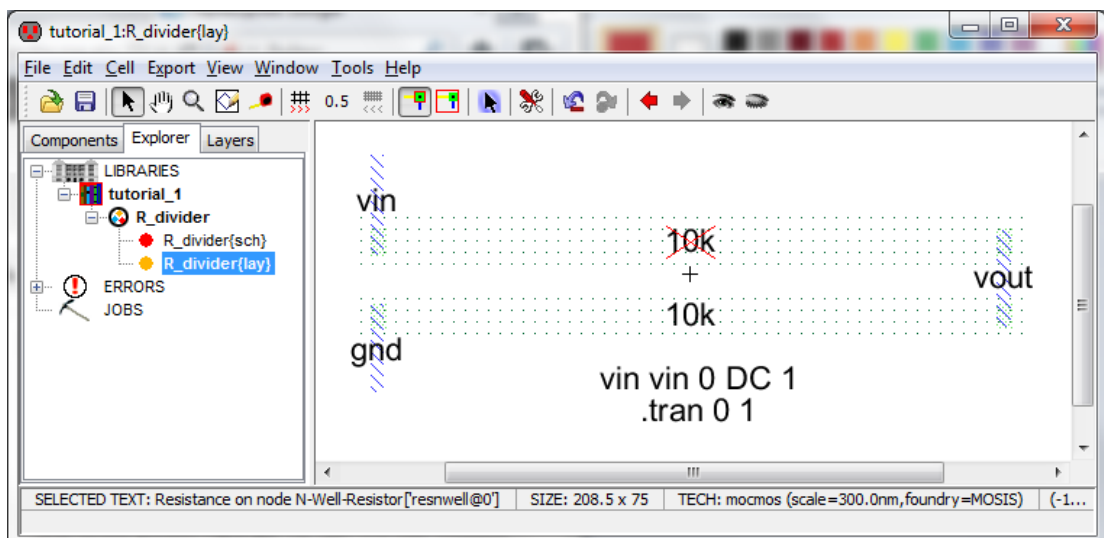


Рис. 1.44. Топологія подільника напруги із *spice* кодом



Перевіримо макет на наявність помилок *DRC (F5)* та *NCC (L)*.

Змодельуйте цю комірку за допомогою *LTspice*. Результат має бути таким же, як і при моделюванні комірки зі схемою.

## Практичне заняття №2

### Моделювання роботи *NMOS* і *PMOS* транзисторів

Моделювати роботу *NMOS* і *PMOS* транзисторів будемо з використанням налаштувань для техпроцесу *C5*, виставлених у ході практичного заняття №1. Спочатку відкрийте бібліотеку *tutorial\_1.jelib* і збережіть її під іменем *tutorial\_2.jelib*.

Простежте, щоб у папці *C:\Electric\* перебував файл *C5\_models.txt*. Цей файл містить *MOSFET* моделі процесу *C5*.

Створіть нову комірку *NMOS\_IV* (рис. 2.1).

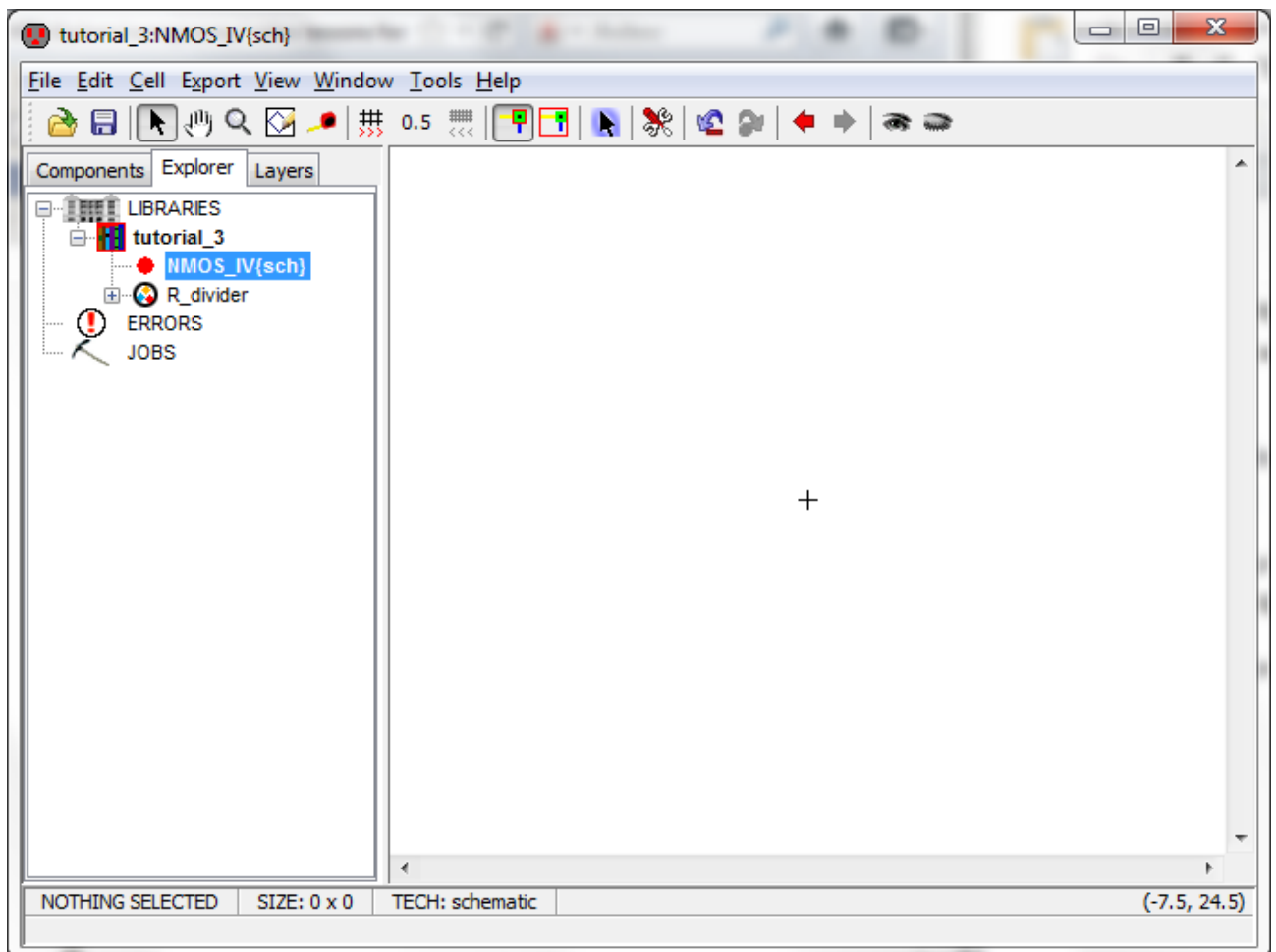


Рис. 2.1. Створення нової комірки

Зверніть увагу на те, що група комірок *R\_divider* з першого практичного заняття збереглась.

Виберіть у закладці *Components* *NMOS* транзистор, і перетягніть його в робочу область (рис. 2.2).

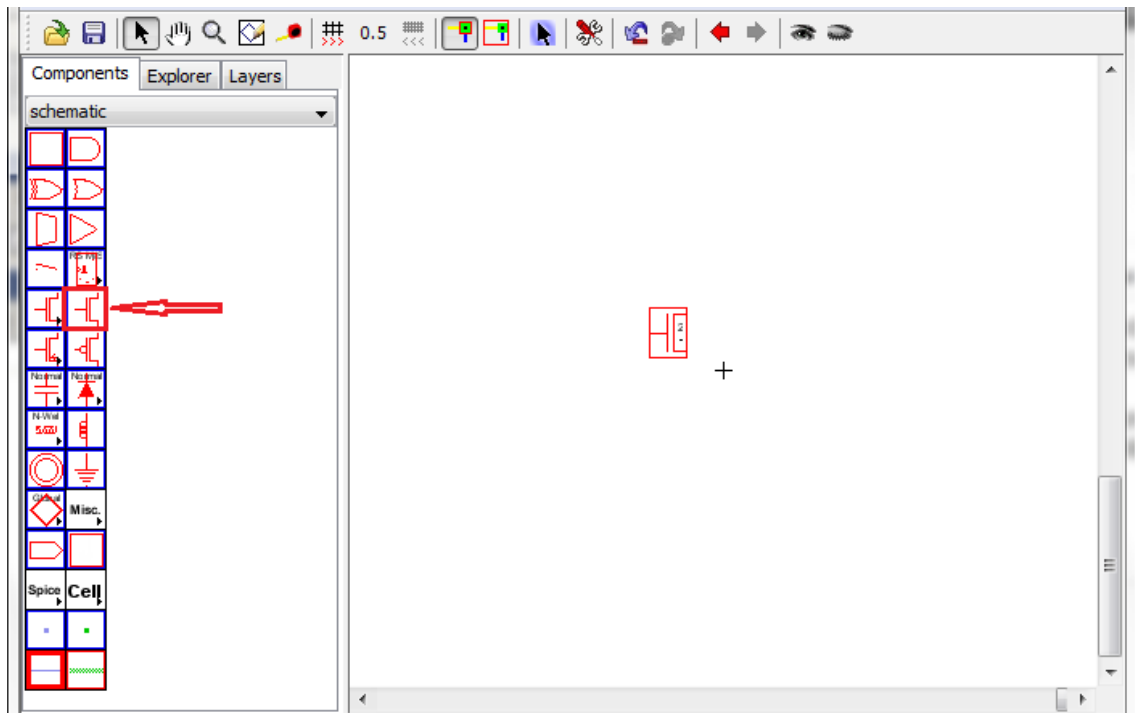


Рис. 2.2. Вибір *NMOS* транзистора

Переходимо в меню *Window* → *Fill Window* або *Ctrl+9* для масштабування зображення по робочій області (рис. 2.3).

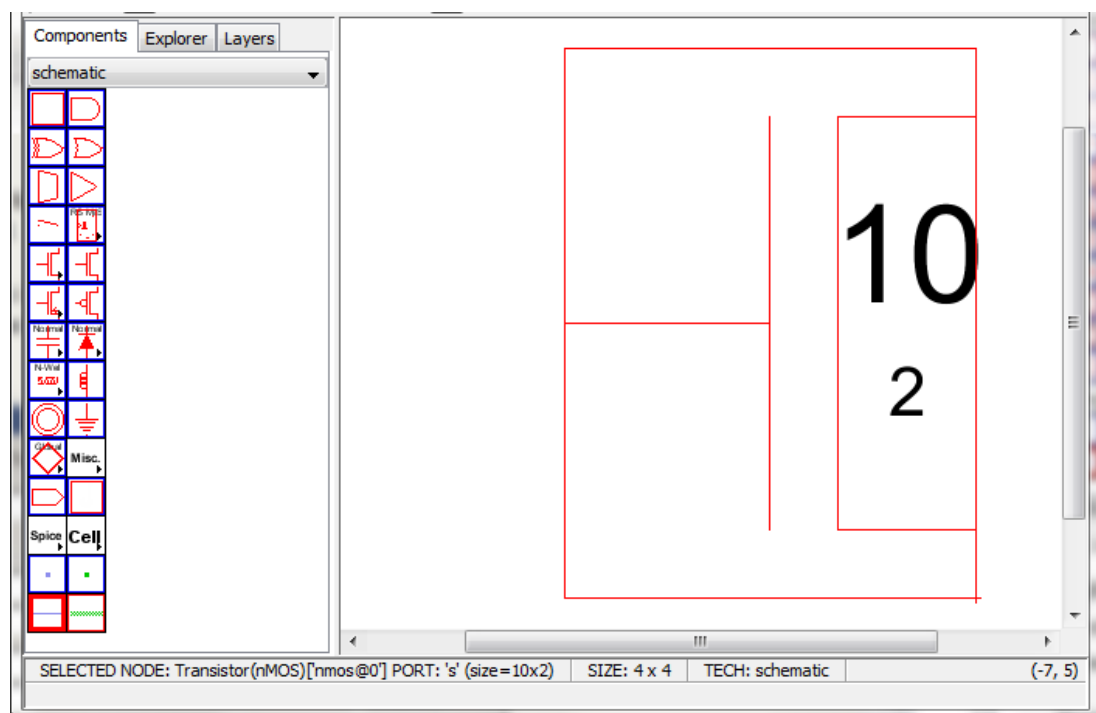


Рис. 2.3. Відмасштабоване зображення транзистора по робочій області

Виділяємо транзистор й редагуємо властивості вузла, меню *Edit* → *Properties* → *Object Properties* (*Ctrl+I*). Встановіть значення ширини й довжини пристрою: *Width: 10, Length: 2* (рис. 2.4). Натисніть «OK».

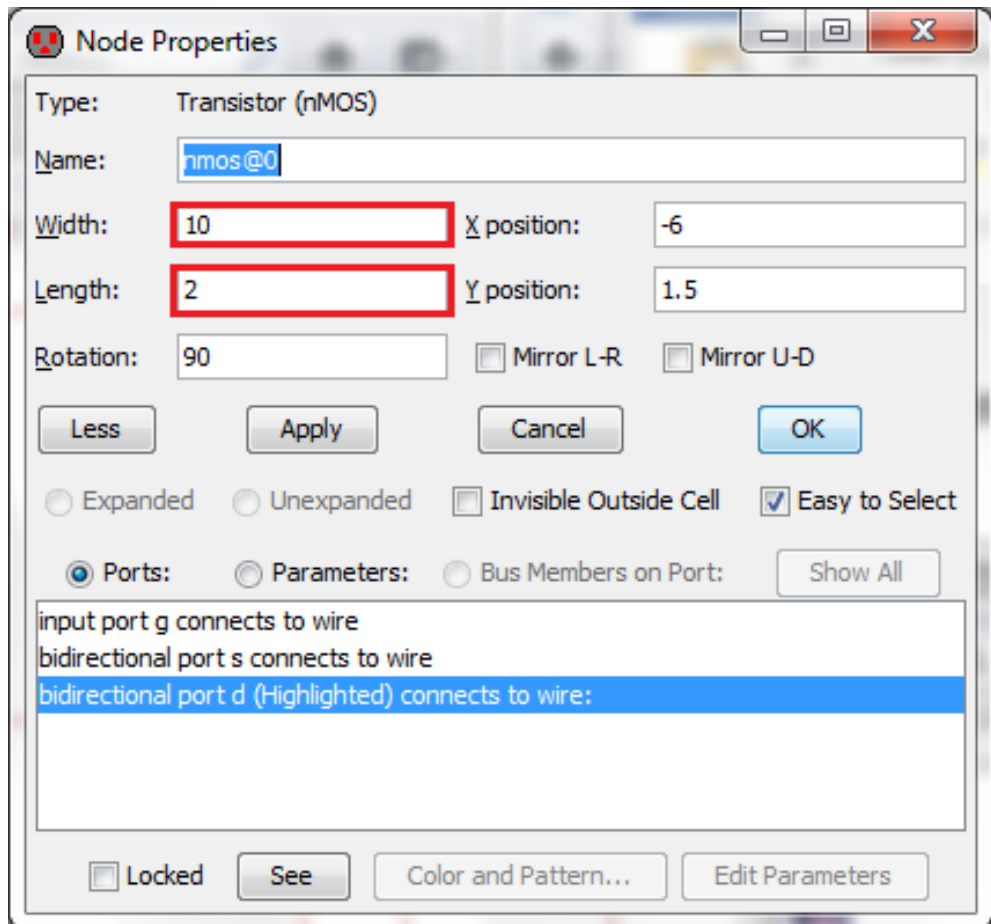


Рис. 2.4. Встановлення параметрів *NMOS* транзистора

Якщо є необхідність повернути/змінити розмір напису, який позначає ширину або довжину пристрою, то для цього потрібно натиснути із затиснутою клавішею *Ctrl* лівою кнопкою мишки по тексту й у його властивостях (*Ctrl+I*) указати потрібні значення. Для повороту тексту також можна використовувати комбінацію клавіш *Ctrl+J* або *Edit* → *Rotate* → значення кута повороту. Розмістіть написи так, як показано на рис. 2.5. Також простежте, щоб сітка вирівнювання мала значення 0,5.

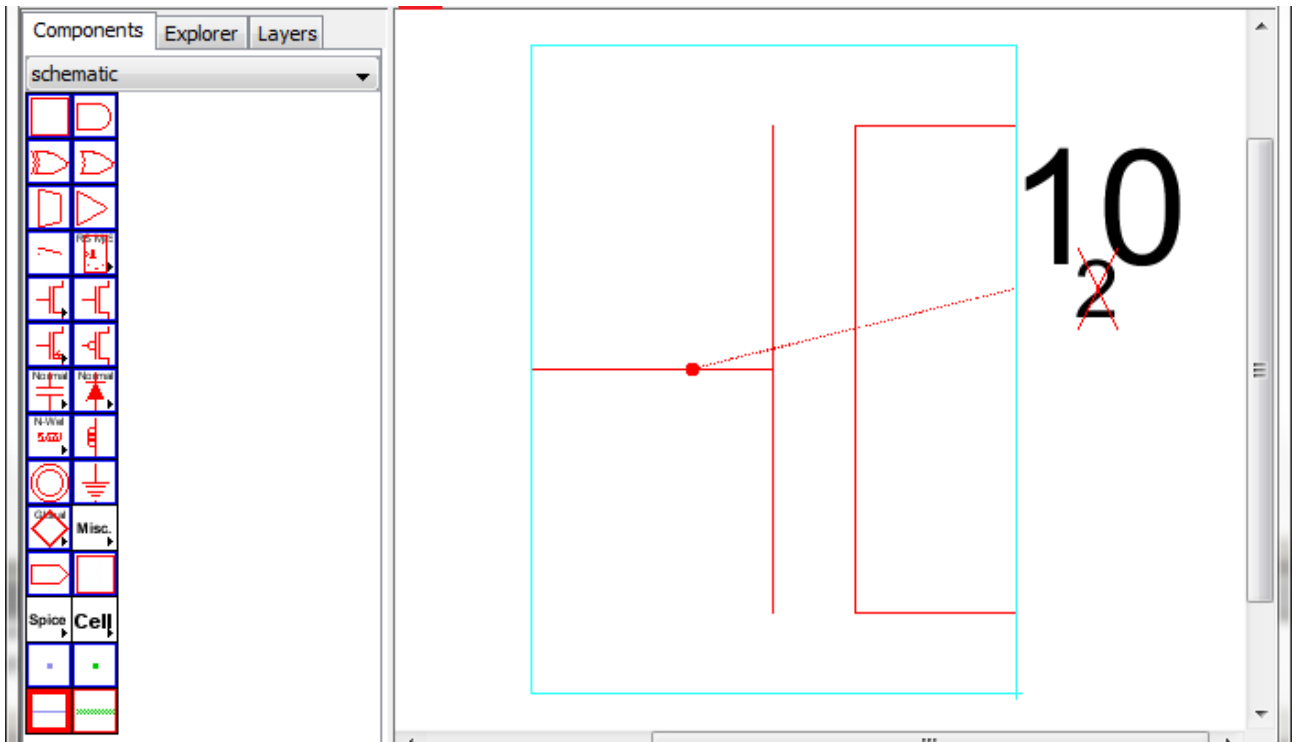
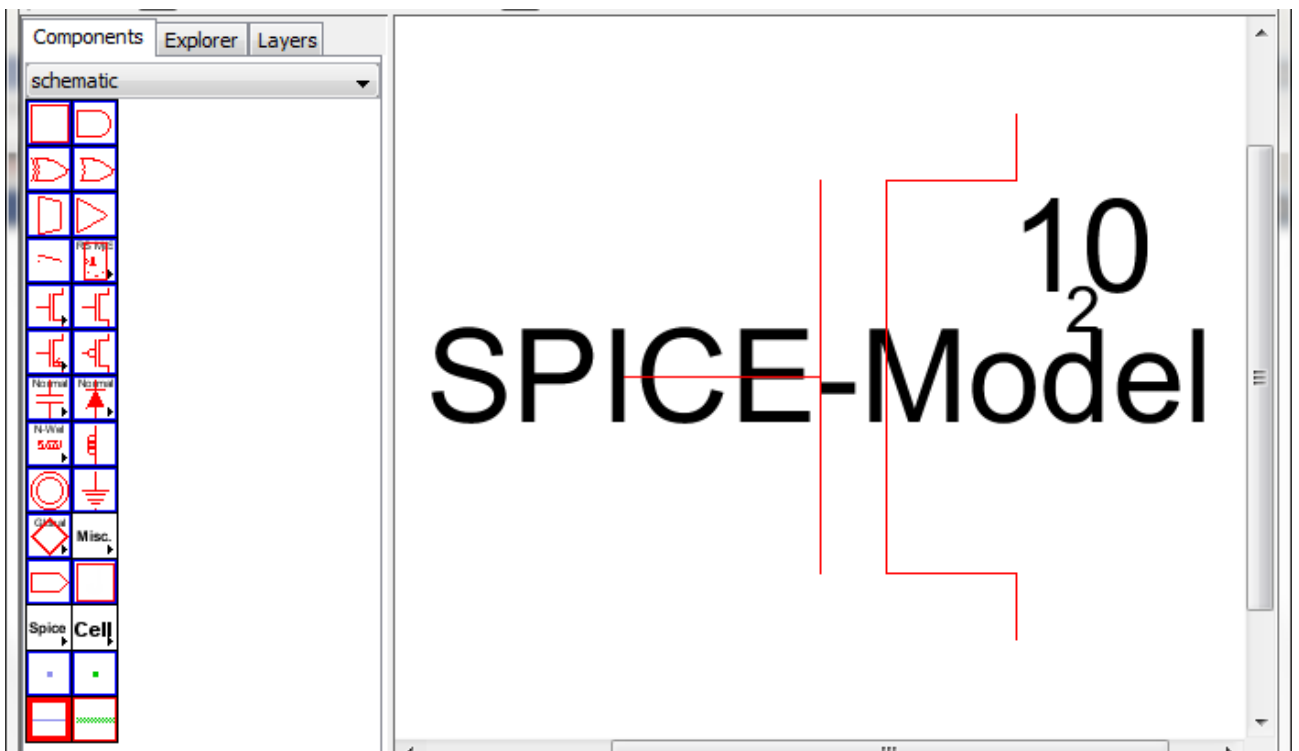


Рис. 2.5. *NMOS* транзистор із встановленими параметрами

Вставте *Spice* модель для вузла. Виділіть *NMOS* вузол (транзистор) і перейдіть у меню *Tools* → *Simulation (Spice)* → *Set Spice Model* (рис. 2.6).



Риса. 2.6. Вставлення *Spice* моделі для вузла

Змініть текст *Spice* моделі, зайшовши в її властивості (для цього виділіть *Spice*-Текст і натисніть *Ctrl+I*).

Відкрийте файл *C5\_models.txt*. У ньому Ви побачите, що назва *NMOS* моделі називається *NMOS*, для *PMOS* моделі аналогічно.

Змінюємо текст *Spice* моделі на *NMOS*, як показано на рис. 2.7.

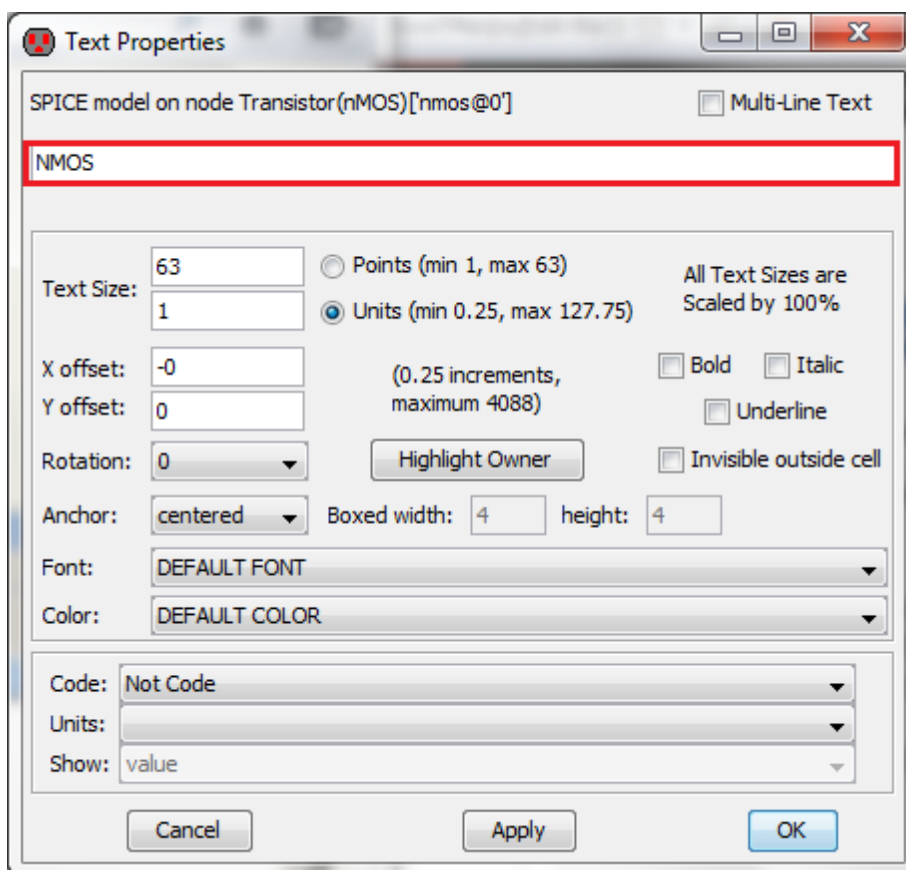


Рис. 2.7. Встановлення *Spice* моделі транзистора

Для зручності перемістіть *Spice* текст як показано на рис. 2.8.

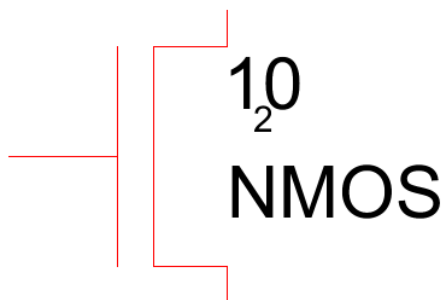


Рис. 2.8. Розміщення *Spice* тексту біля вузла

Тепер, надалі, за необхідності, можна скопіювати *NMOS* вузол, не повторюючи пройдених дії щоразу, коли знадобиться *NMOS* транзистор.

Повторимо описані дії для створення *PMOS* транзистора.

Ім'я нової комірки – *PMOS\_IV*. Результат має бути таким, як показано на рис. 2.9.

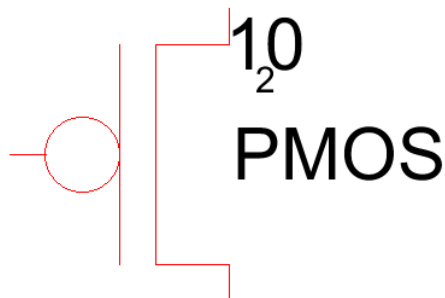


Рис. 2.9. Створений *PMOS* транзистор

Виділяємо *PMOS\_IV* вузол, потім переходимо в меню *Edit* → *Change* або клавіша «С». Змінимо вузол на 4-портовий пристрій. У списку, що розкривається, переходимо *Primitives* → *4-port-transistor*, вибираємо пункт *pmos-4*, натискаємо *Change*, потім *Done* або подвійне клацання по пункту *pmos-4*.

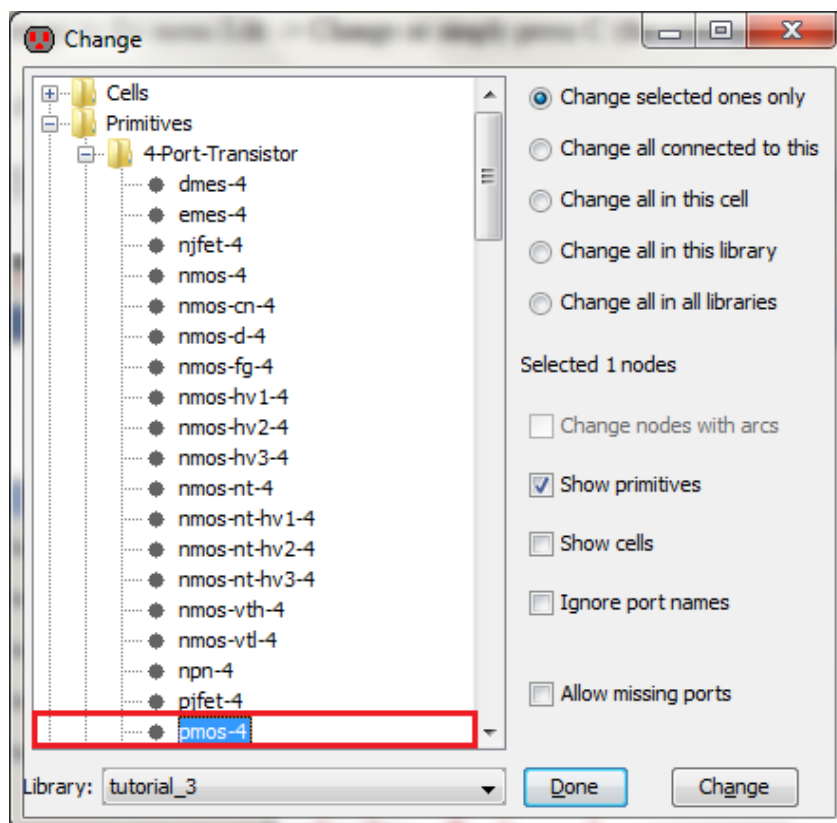


Рис. 2.10. Встановлення *PMOS* транзистора, як 4-портового пристрою

Скористаємося функцією *Mirror*, «перевернемо» вузол. Виділіть вузол *PMOS\_IV* і виберіть меню *Edit*→*Mirror*→*Up*↔*Down*. У результаті контакт вузла, який ми додали, має виявитися зверху, як показано на рис. 2.11.

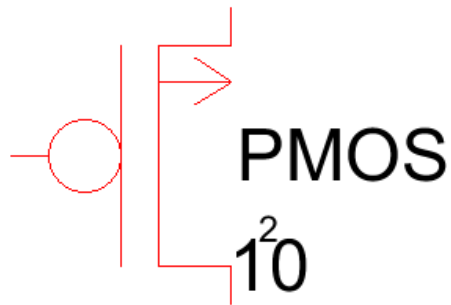


Рис. 2.11. Перевернутий 4-портовий *PMOS* транзистор

Додамо четвертий порт в *NMOS* транзистор (рис. 2.12).

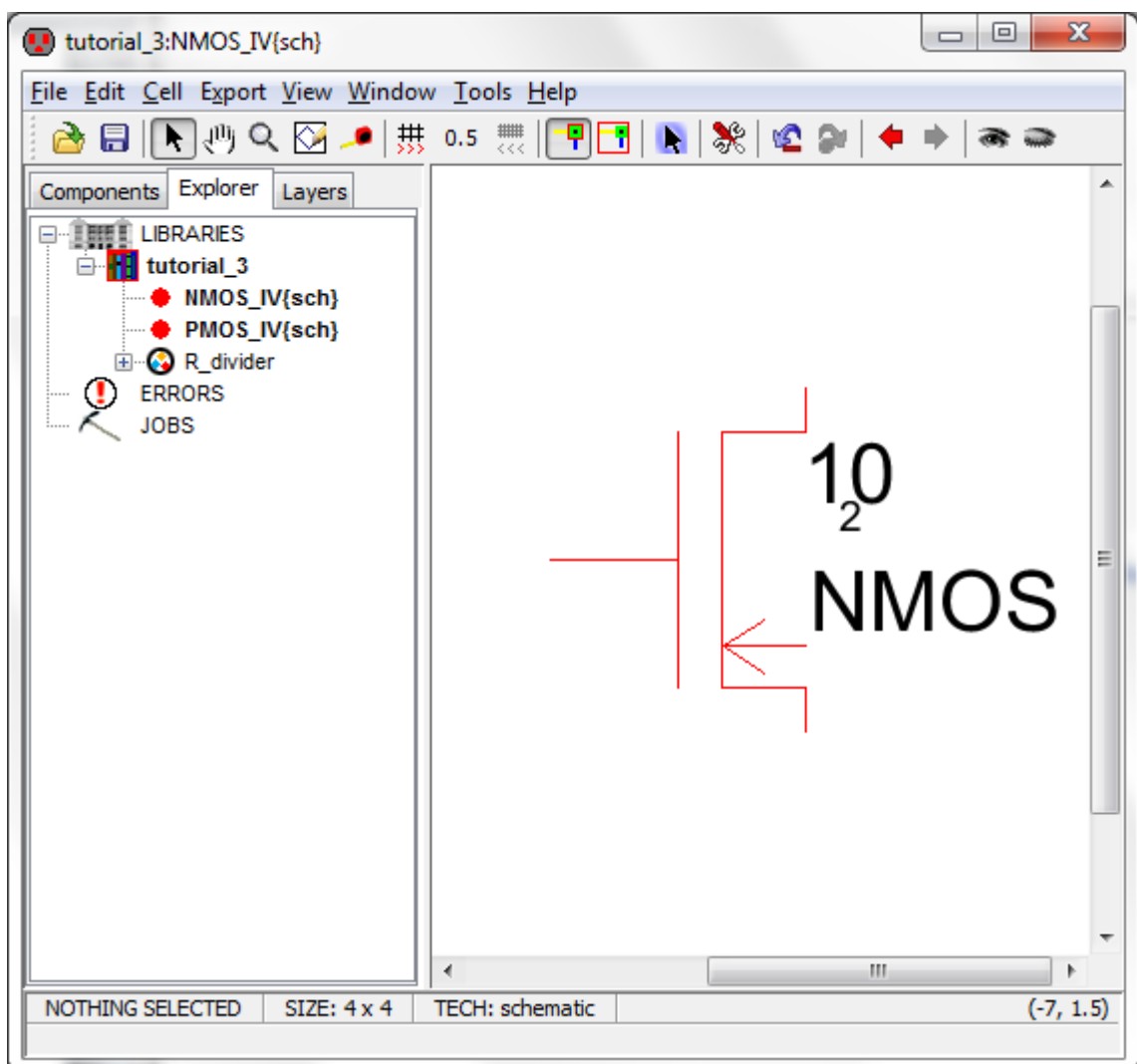


Рис. 2.12. Встановлення *NMOS* транзистора, як 4-портової пристрою



Створимо топології комірок. Створимо нову комірку (*Cell* → *New Cell* або *Ctrl+N*). Назвемо її *NMOS\_IV*, режим перегляду – *layout* (рис. 2.13).

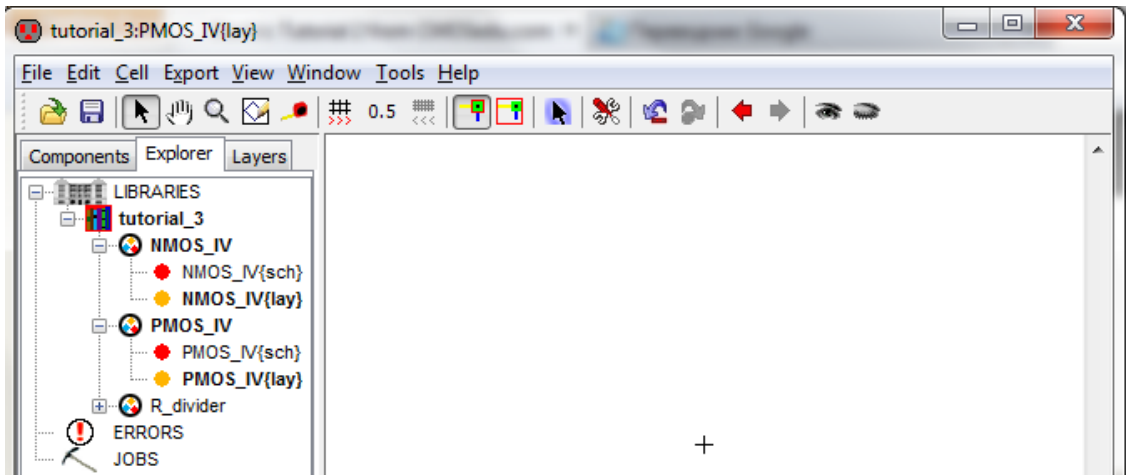


Рис. 2.13. Створення тологічних комірок

Далі виділяємо комірку *NMOS\_IV{lay}*. Переходимо в меню *Components* і перетягуємо звідти *nMOS* транзистор (у правому нижньому куті меню *Components*) у робочу область (рис. 2.14).

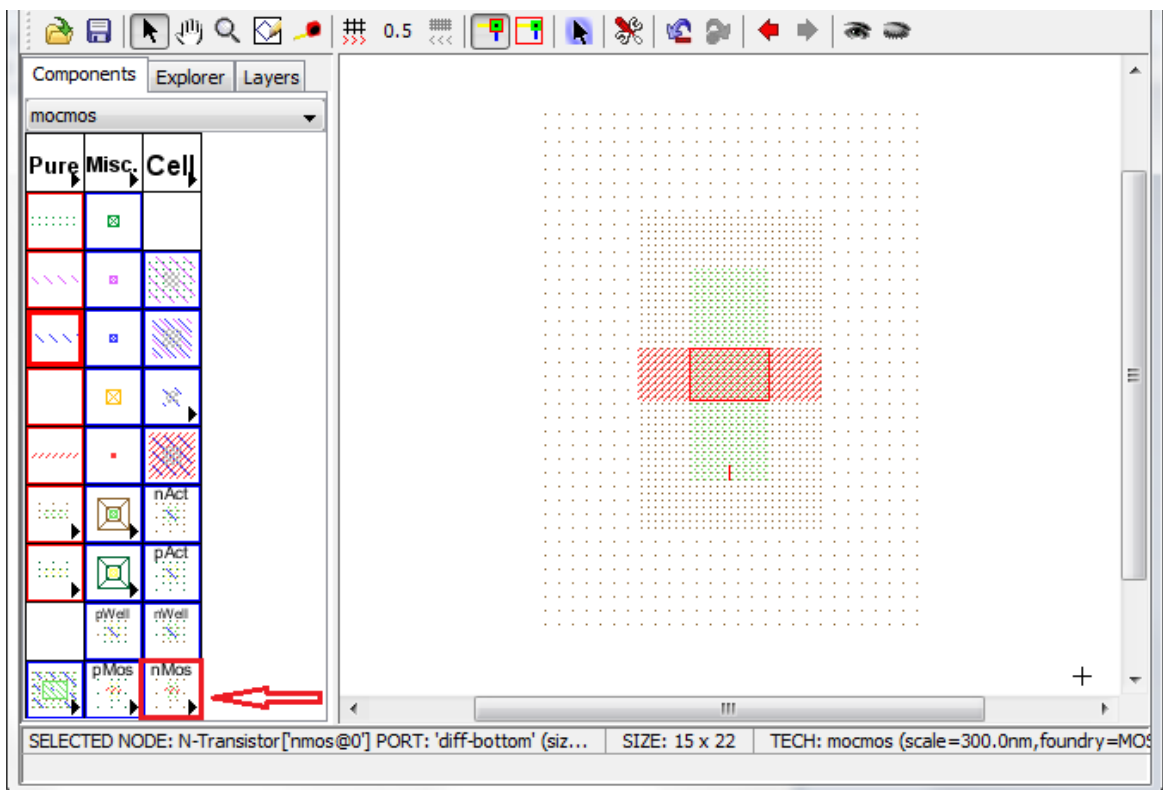


Рис. 2.14. Топологічне зображення заслона *NMOS* транзистора

Тепер створимо області, які будуть формувати стік і витік пристрою. У вкладці *Components* вибираємо вузол *nAct* і ставимо їх так, як показано на рис. 2.15.

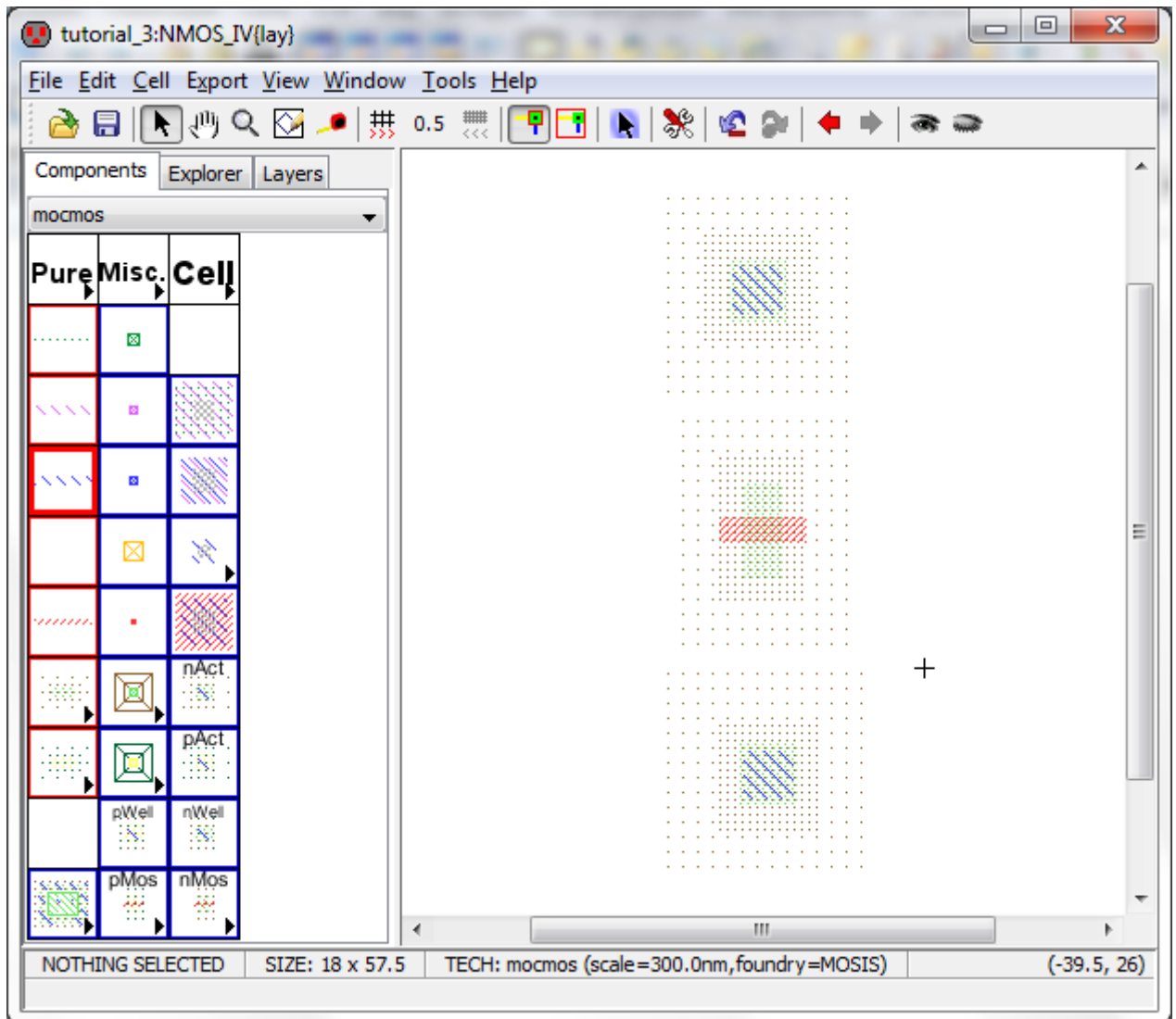


Рис. 2.15. Розміщення стоку й витіку *NMOS* транзистора

Далі виводимо вузол метал1–полікремній1 для підключення до заслона *MOSFET* транзистора. Також нам потрібний вузол для підключення підкладки до землі. Для цього внесемо в область проектування *p*-карман (*pWell*).

У результаті в області проектування має бути 5 вузлів (рис. 2.16).

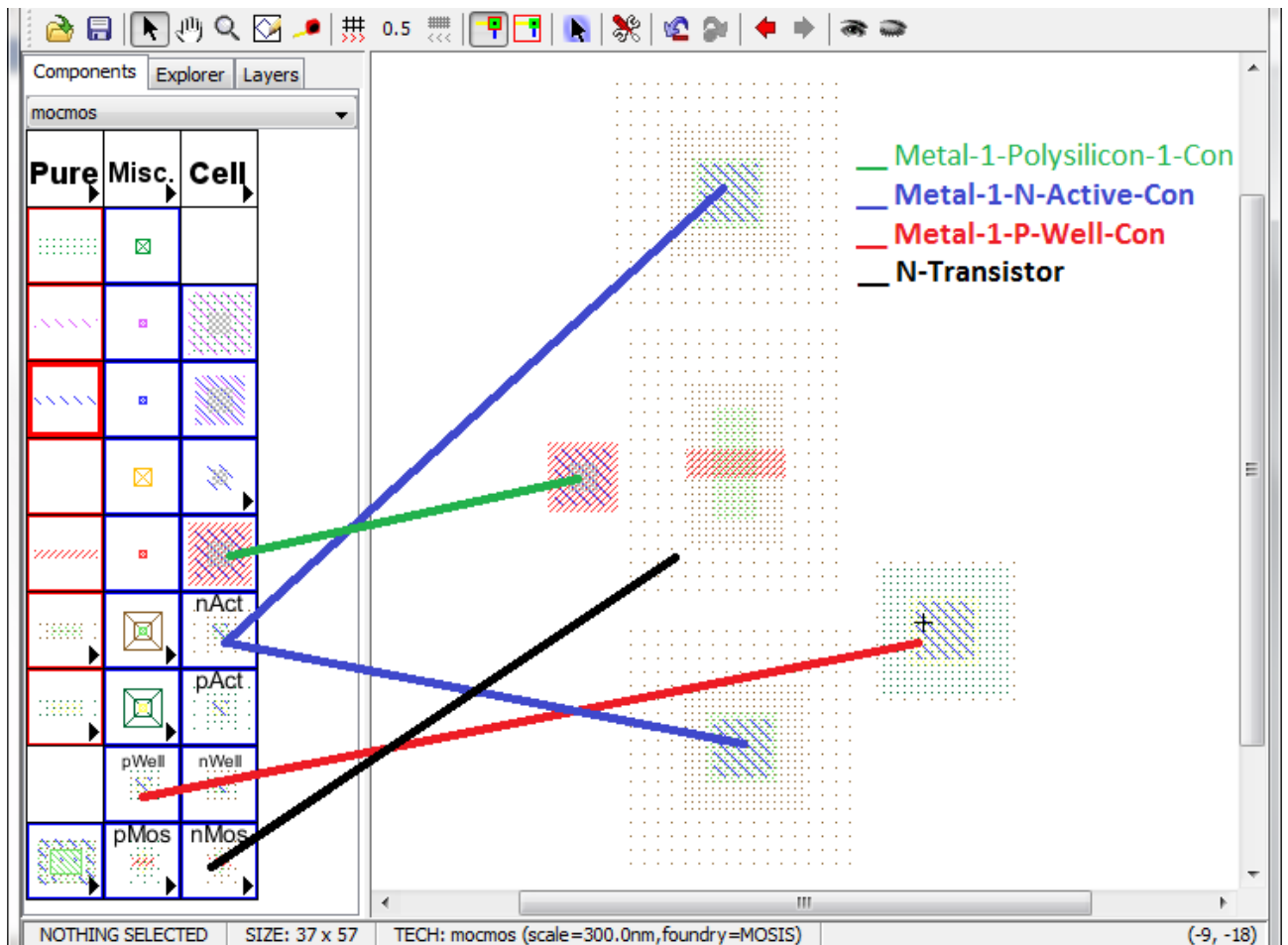


Рис. 2.16. Вузли для топології *NMOS* транзистора

Змініть ширину вузла *N-Transistor* на значення 10. Для того, щоб змінити розмір вузлів зайдіть у властивості об'єкта, меню *Edit* → *Properties* → *Object Properties* (або *Ctrl+I*), у полі «*width:*» укажіть значення 10.

Змініть ширину вузлів *Metal-1-N-Active-Con* і *Metal-1-P-Well-Con* на значення 10. (Для однотипних елементів використовуйте групове виділення *Shift+left click*).

Створіть *Spice* модель для вузла *NMOS* транзистора. Для цього виділіть вузол *N-Transistor* і перейдіть у меню *Tools* → *Simulation (Spice)* → *Set Spice Model*. У рядку вказування *Spice* коду вкажемо ім'я *Spice* моделі «*NMOS*» (згідно з файлом *C5\_models.txt*, де зазначені параметри моделей). Для того, щоб виділити *Spice* код, затисніть клавішу *Ctrl*, натискаючи ліву кнопку миші. Результат зазначений на рис. 2.17.

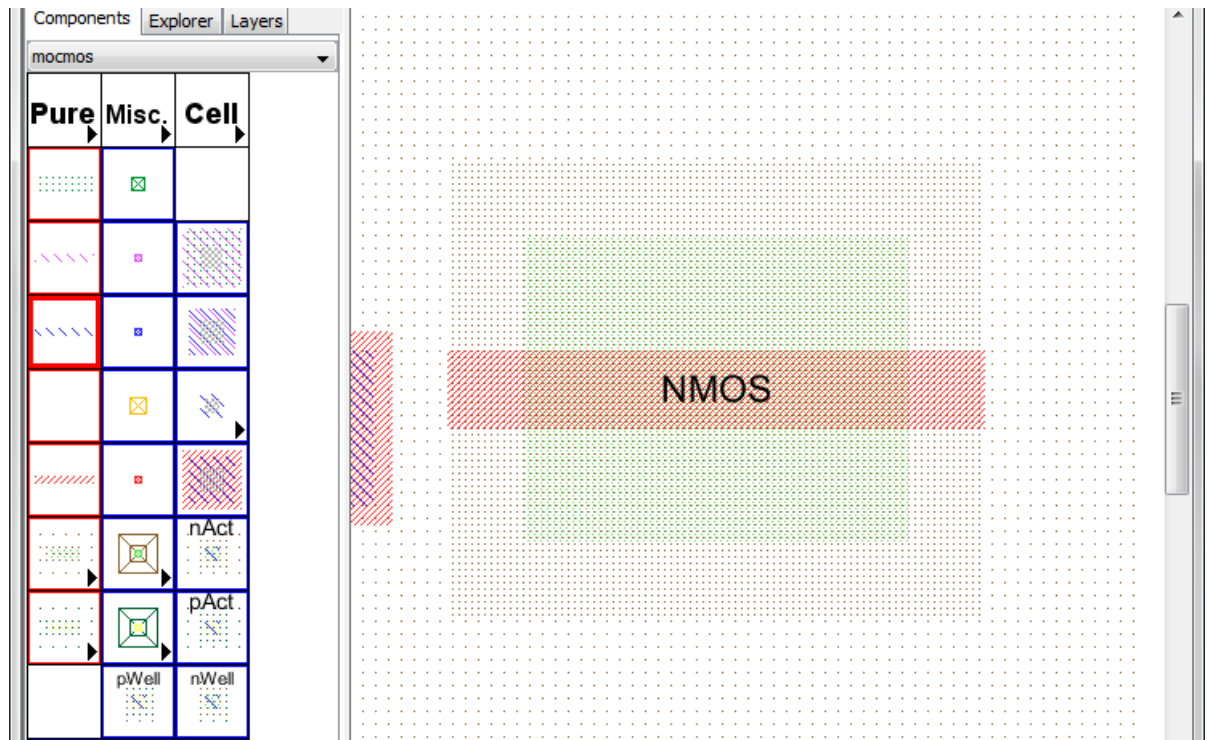


Рис. 2.17. Вставлення *Spice* моделі для вузла *NMOS* транзистора

Відмасштабуйте робочу область так, щоб було видно всю топологію (*Ctrl+9*).

Виділіть верхній порт вузла *nMos* як показано на рис. 2.18.

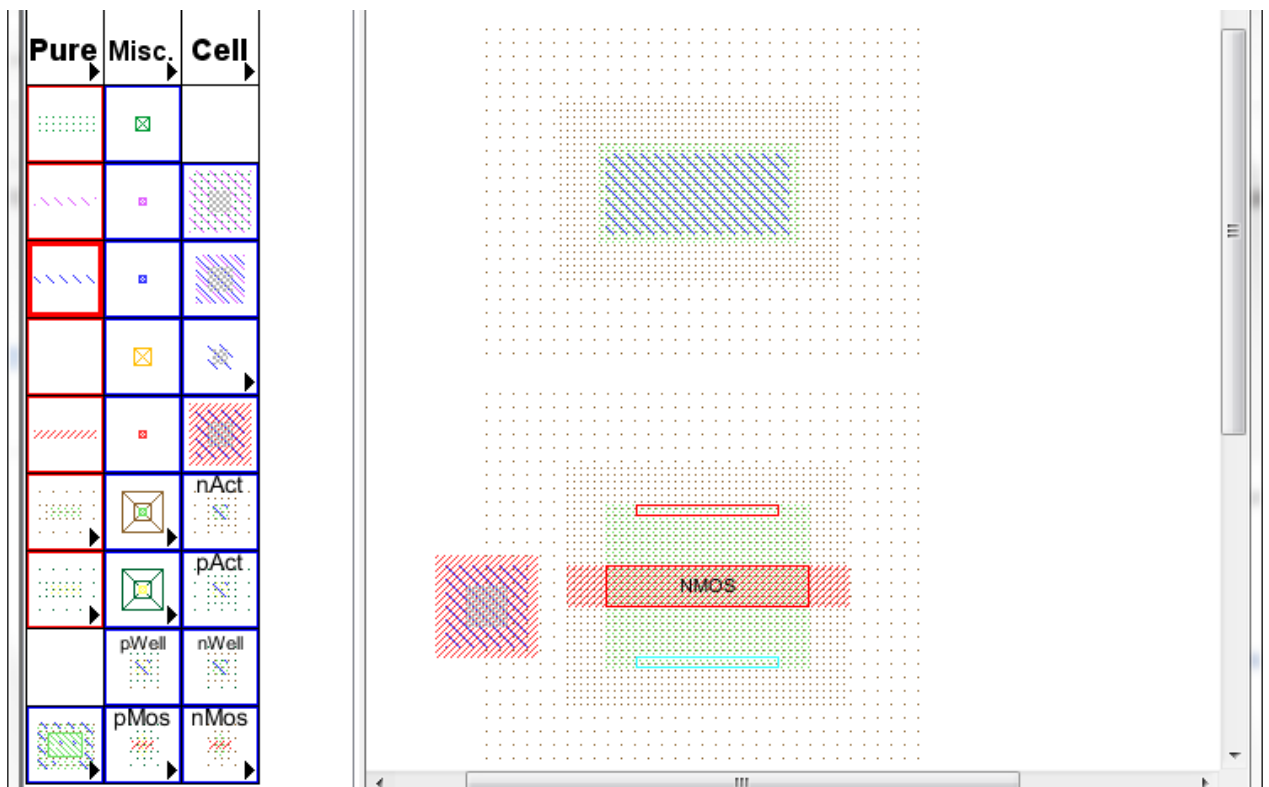


Рис. 2.18. Виділення верхнього порта вузла *nMos*

Клацніть правою кнопкою мишки по вузлу *Metal-1-N-Active-Con* зверху. Дуга *N-Active* була додана між вузлами (рис. 2.19).

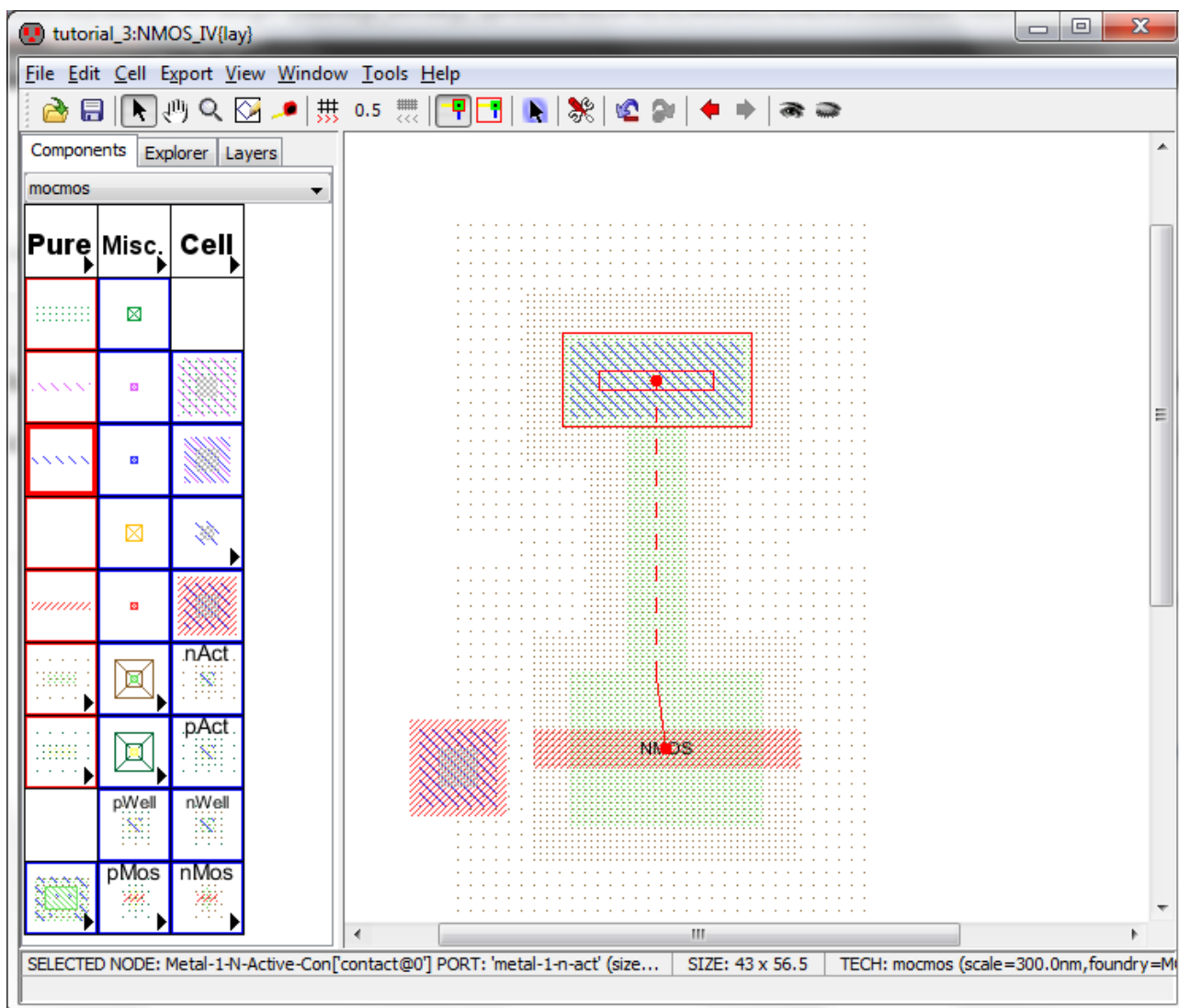


Рис. 2.19. Додавання дуги між вузлами

Перевірте топологію на наявність помилок, скориставшись інструментом *DRC* (*F5*). Можливі помилки некоректного розміщення вузлів (вузол *Metal-1-Polysilicon-1-Con* «залазить» на *N-Transistor*, перемістіть його лівіше). Виправте помилки.

Наблизимо максимально вузол *Metal-1-N-Active-Con* до *N-Transistor*, якщо відстань між вузлами буде занадто малою, то з'явиться повідомлення над маніпульовуваним вузлом і штрихпунктирна мітка в області помилки, як показано на рис. 2.20.

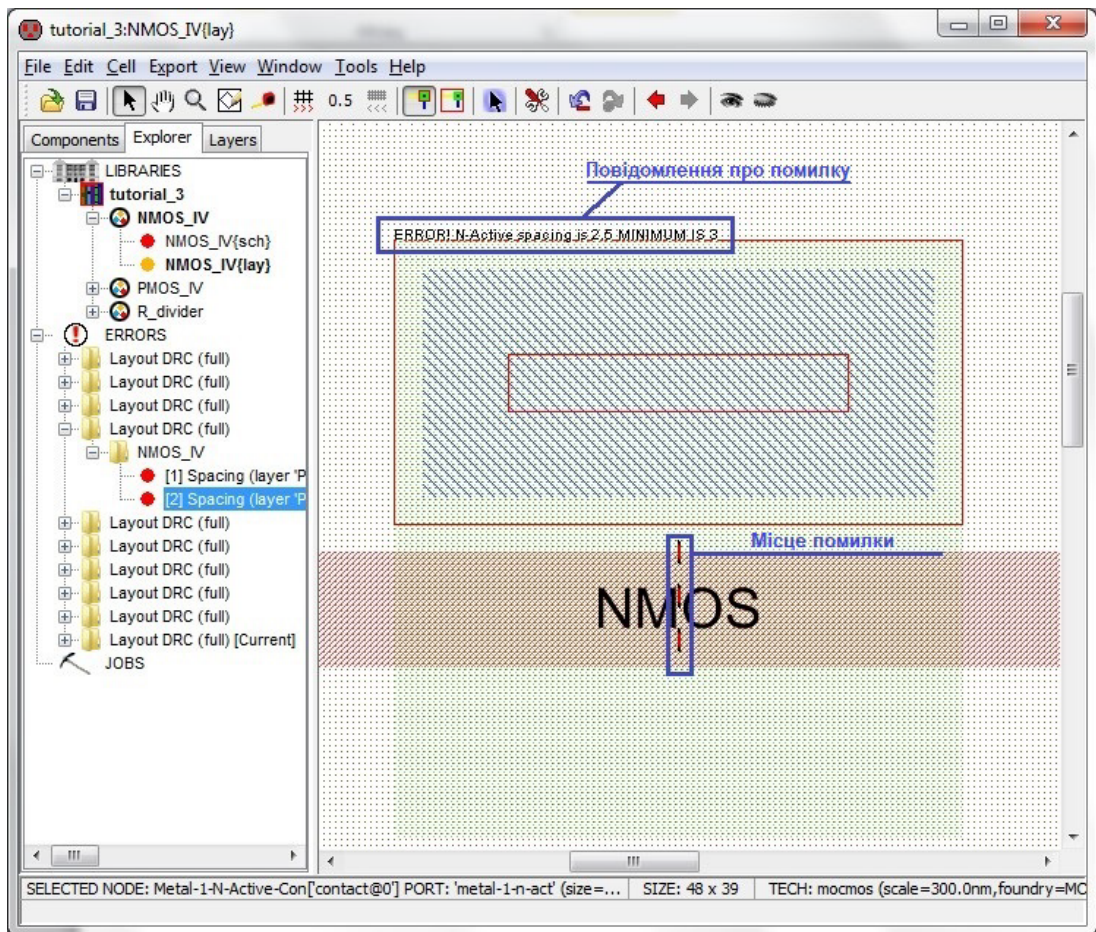


Рис. 2.20. Повідомлення про помилку в топології

Звісно, *DRC* не пропустить надалі помилку.

При правильному виконанні топологія має виглядати приблизно як на рис. 2.21.

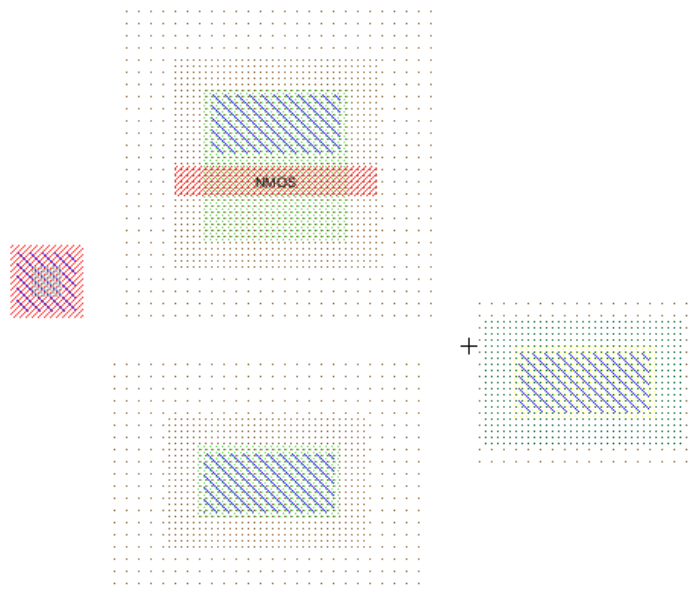


Рис. 2.21. Зближені стік і заслін *NMOS* транзистора

З'єднаємо нижній вузол *Metal-1-N-Active-Con* і *N-Transistor*. Для початку змістить витік (нижній вузол *Metal-1-N-Active-Con*) вправо, поки не відключиться прив'язка до транзистора. Тепер з'єднаємо витік із транзистором.

У результаті *n*-активна дуга буде мати значення ширини таке ж, як і значення ширини вузла, тобто 10.

Якщо виділити витік й рухати його в сторони, то дуга буде «тягтися» за вузлом. Для того, щоб дугу жорстко зв'язати з вузлом використовуйте налаштування «*Rigid*» у властивостях дуги.

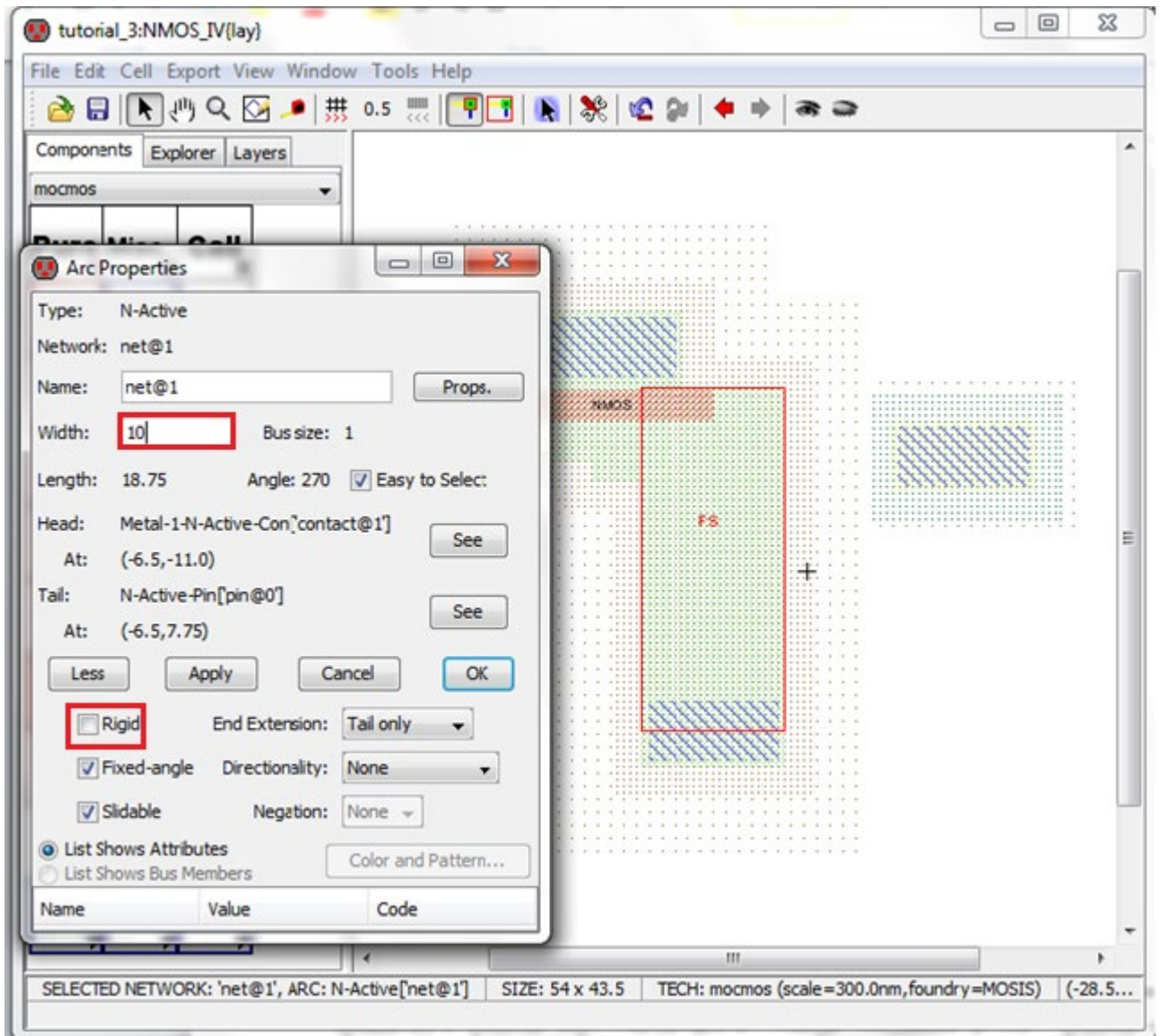


Рис. 2.22. Налаштування властивостей дуги

Розмістіть нижній *N-Active* вузол (витік) дзеркально верхньому вузлу *N-Active* (стік) відносно *N-Transistor* вузла, як показано на рис. 2.23. Перевірте макет на наявність *DRC* помилок.

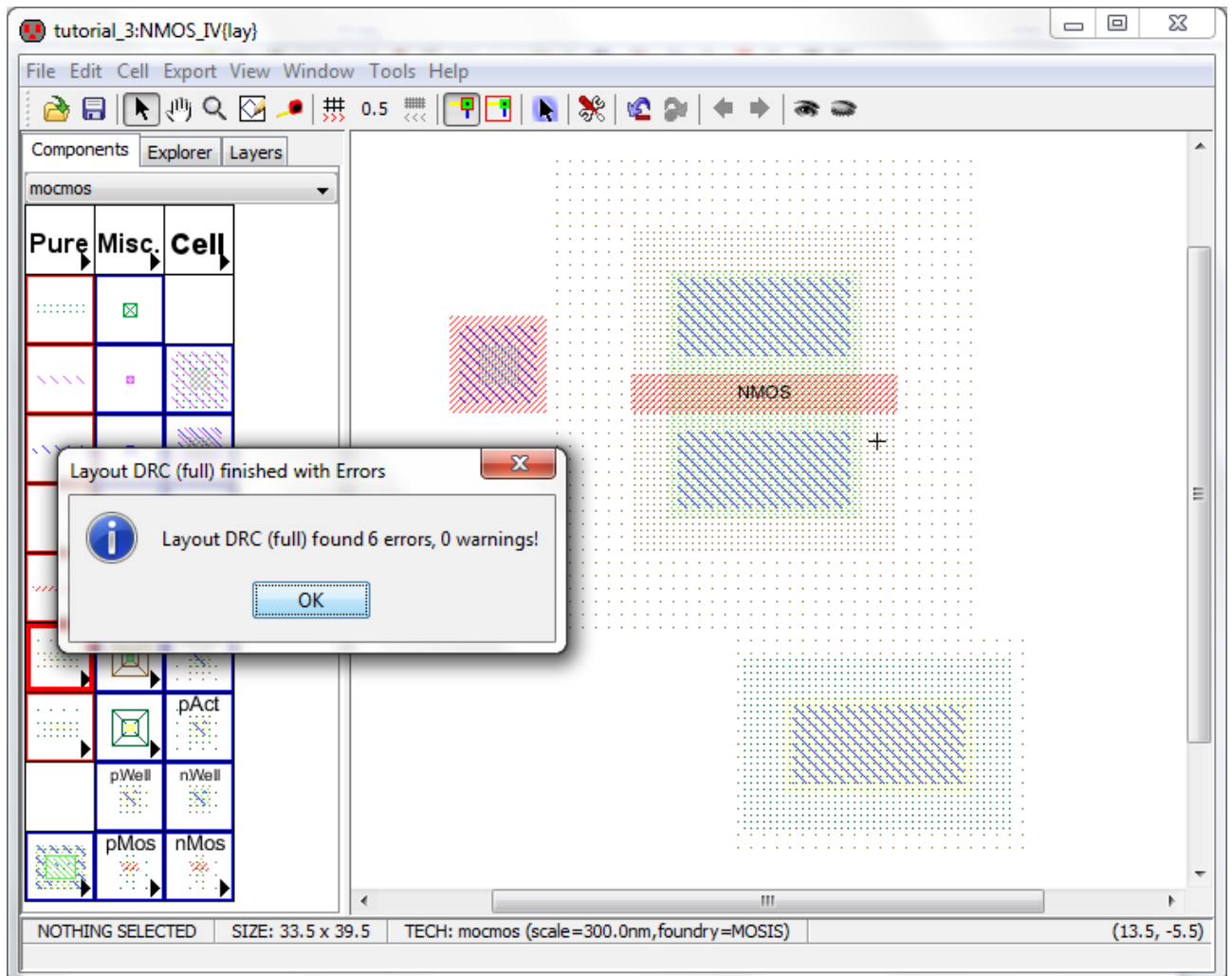


Рис. 2.23. Топологія *NMOS* транзистора із помилками

Щоб побачити, які місця спричиняють помилки, затискайте клавіші *Shift+>/<*. При цьому, у вікні *Electric Messages* будуть з'являтися повідомлення про дану помилку. Дана помилка викликана тим, що вузол *N-Transistor* торкається *N-Active* дуги. Виділіть дугу (для виділення використовуйте *Ctrl+left-click*), і у її властивостях у налаштуванні *End Extension* оберіть пункт *Neither end*. Після цього помилок *DRC* не має бути.

Приєднаємо заслін *MOSFET* транзистора до контактної площинки *Metal-1-Polysilicon-1-Con*. Оберемо лівий порт заслона транзистора й



клацнемо правою кнопкою мишки по контактній площинці. Помилки *DRC* мають бути відсутніми. Результат показаний на рис. 2.24.

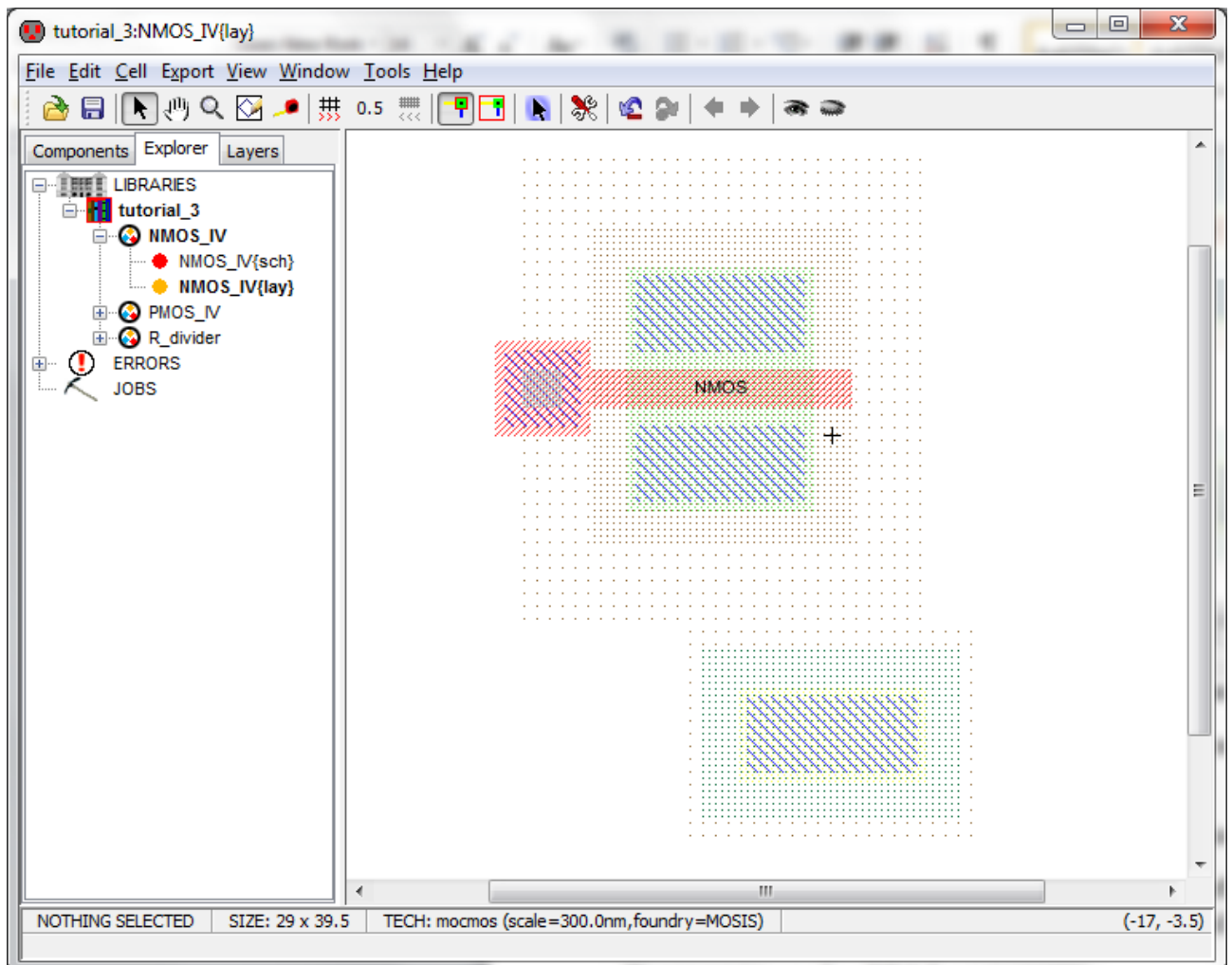


Рис. 2.24. Встановлення контакту заслона

Перевіримо топологію на помилки *ERC* (*Tools* → *ERC* → *Check Wells*). У результаті одержимо повідомлення про дві помилки. Перевірте налаштування *ERC* (*File* → *Preferences* → *Tools* → *Well Check*). Налаштування мають бути такими ж, як і в першому практичному завданні (рис. 2.25).

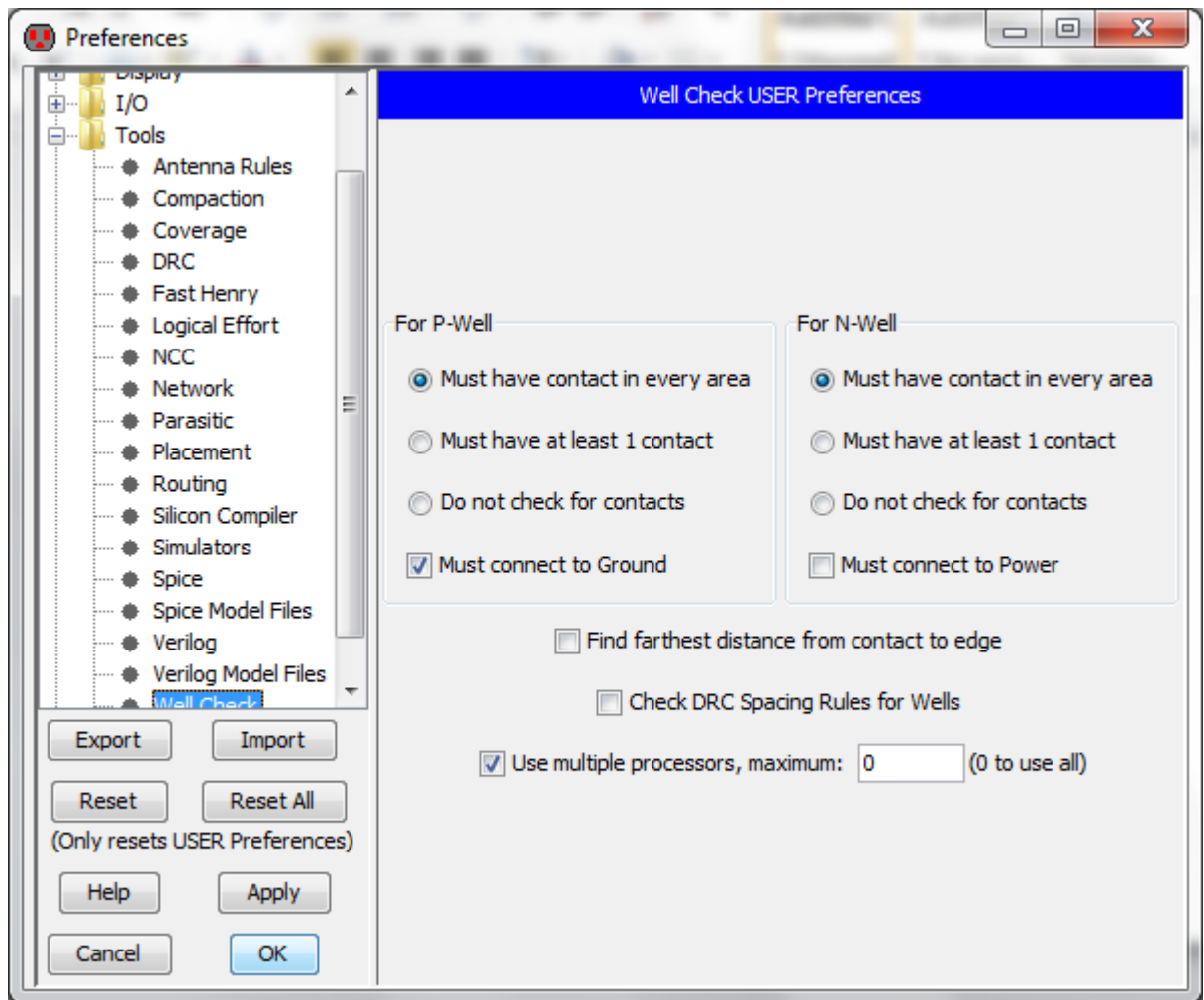


Рис. 2.25. Налаштування *ERC*

Ці дві помилки обумовлені тим, що *p*-область не підключена до «землі», і не має контакт із *NMOS* транзистором.

Повідомлення з *Electric Massager* при перегляді помилок (*Shift+>*) такі:  
*ERC Well Check error 1 of 2: P-Well contact 'substr@2' not connected to ground*

*ERC Well Check error 2 of 2: No P-Well contact in this area*

Переміщайте *p*-область доти, доки вона не перекриє *p*-область *NMOS* пристрою, як показано на рис. 2.26.

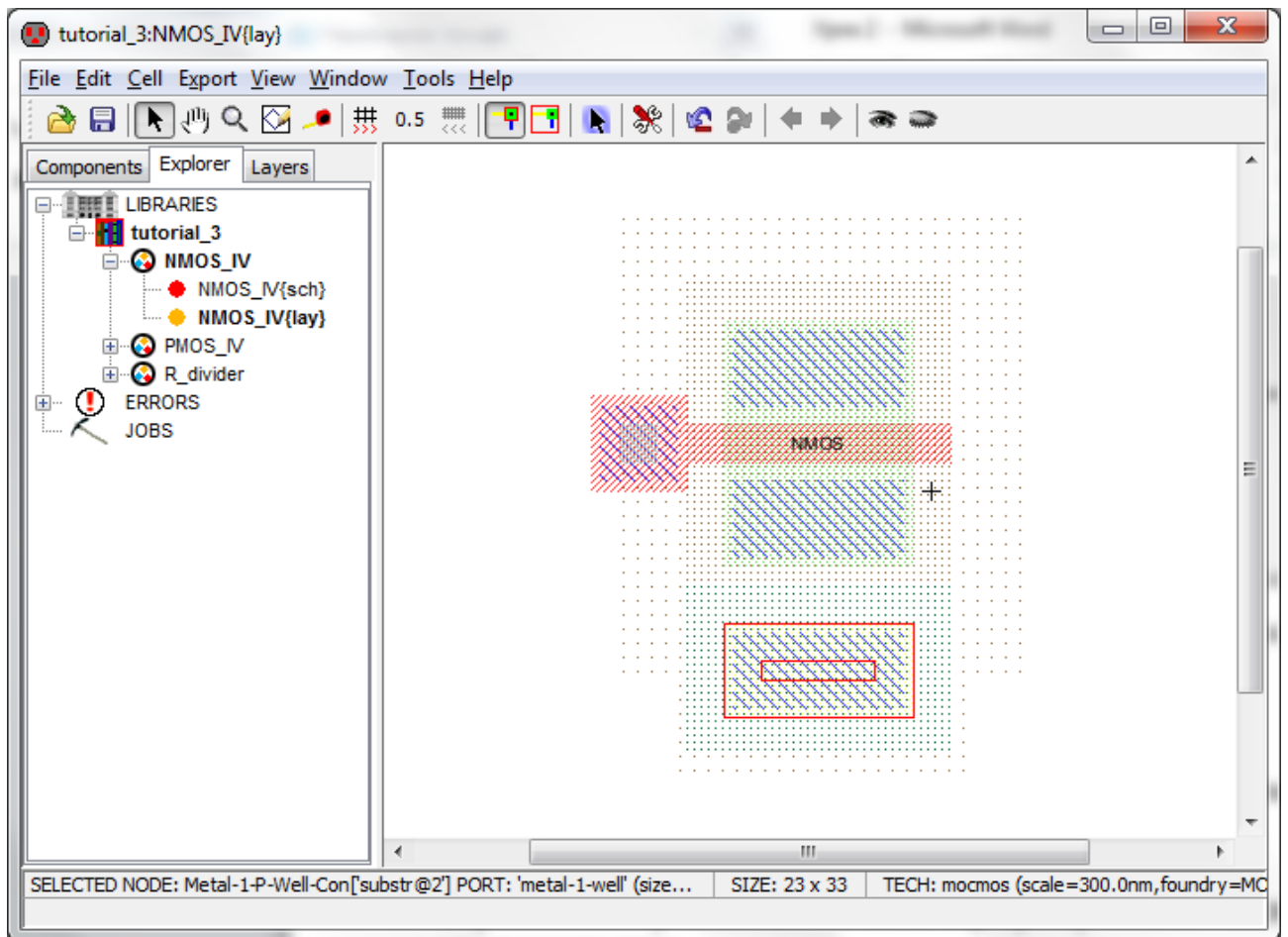


Рис. 2.26. Зближення  $p$ -областей

Перевіримо на помилки *DRC* і *ERC*. Результат має бути, як на рис. 2.27.

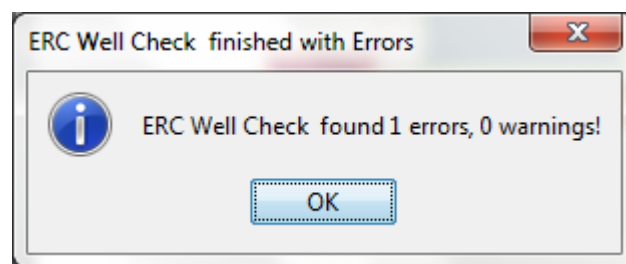


Рис. 2.27. Повідомлення про помилку

Залишилася одна помилка *ERC*.

Щоб *Electric* визначив, до якого вузла підключена «земля», потрібно поставити мітку «*gnd*» на контакт вузла. Виділіть вузол *Metal-1-P-Well-Con* і протягніть від нього металеву дугу, як показано на рис. 2.28. Дуга може прорисовуватися із шириною рівною 10, змініть її на 5 у властивостях дуги.

Експортованими можуть бути тільки вузли, але не дуги. Тому виділення кінця дуги як контакту вузла виглядає так, як показано на рис. 2.28. Для більш зручного виділення використовуйте *Ctrl+left-click*. Виділіть контакт вузла на дузі й виберіть у меню *Export* → *Create Export* або *Ctrl+E*. У рядку *Export name* укажіть *gnd*, як показано на рис. 2.28.

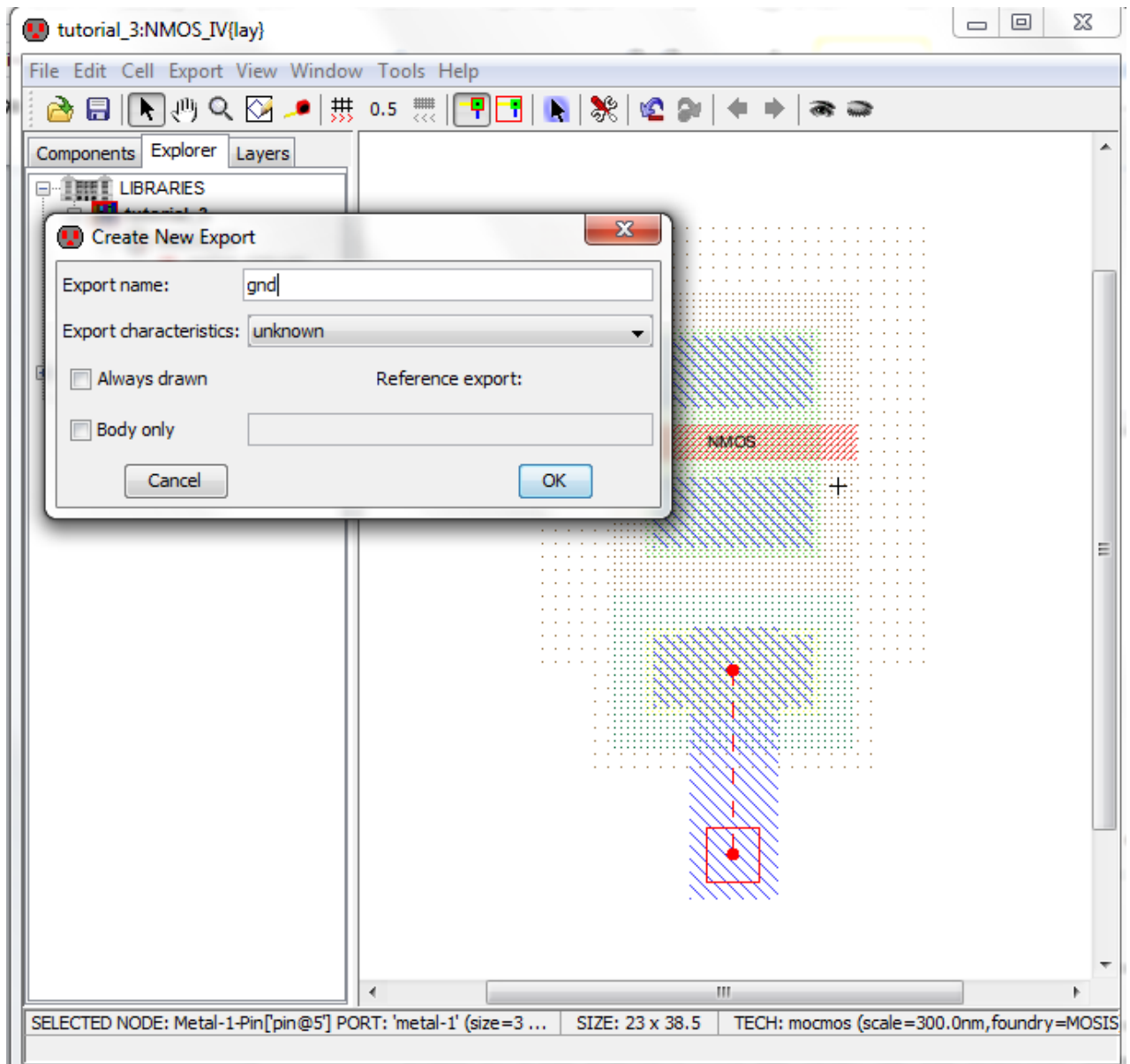


Рис. 2.28. Експорт вузлів

Перевіряємо на помилки *ERC* і *DRC* – помилок немає.

Додайте тепер металеві дуги до інших трьох терміналів *MOSFET* і експоруйте їх так, як показано на рис. 2.29.

Простежте, що би правила *DRC* і *ERC* були дотримані.

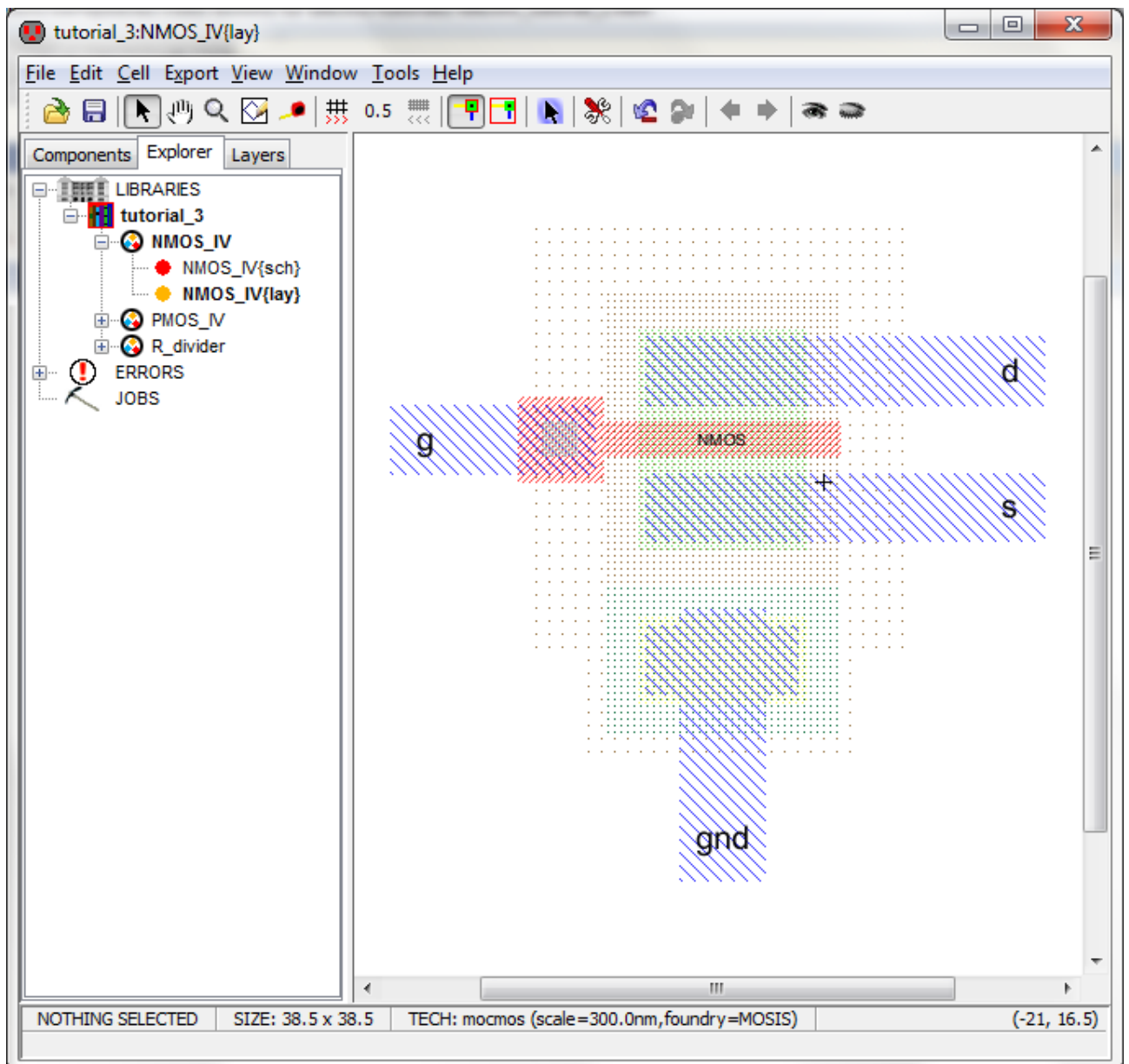


Рис. 2.29. Топологія *NMOS* транзистора із експортованими вузлами

Збережіть бібліотеку.

Тепер, дотримуючись того ж алгоритму, нарисуйте топологію для *PMOS* транзистора.

Перейдіть у комірку *PMOS\_IV{lay}*. Перейдіть у вкладку *Components* і перетягніть вузли у робочу область, як показано на рис. 2.30.

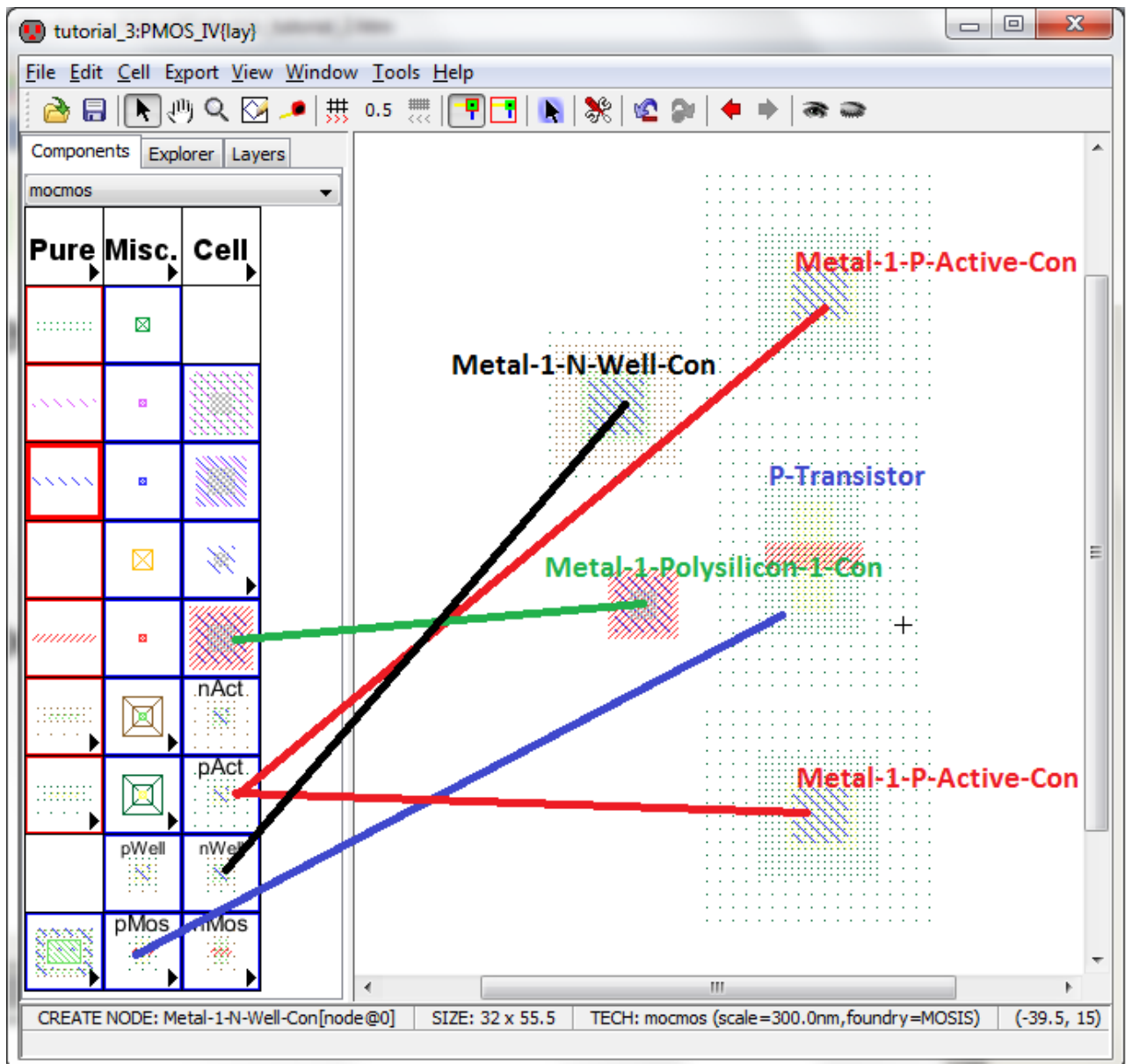


Рис. 2.30. Вузли для топології *PMOS* транзистора

Змініть ширину всіх вузлів на 10, крім вузла *Metal-1-Polysilicon-1-Con*. Далі вставте *Spice* модель транзистора. Для цього виділіть транзистор, перейдіть у меню *Tools* → *Simulation (Spice)* → *Set Spice Model*, укажіть *PMOS* (рис. 2.31).

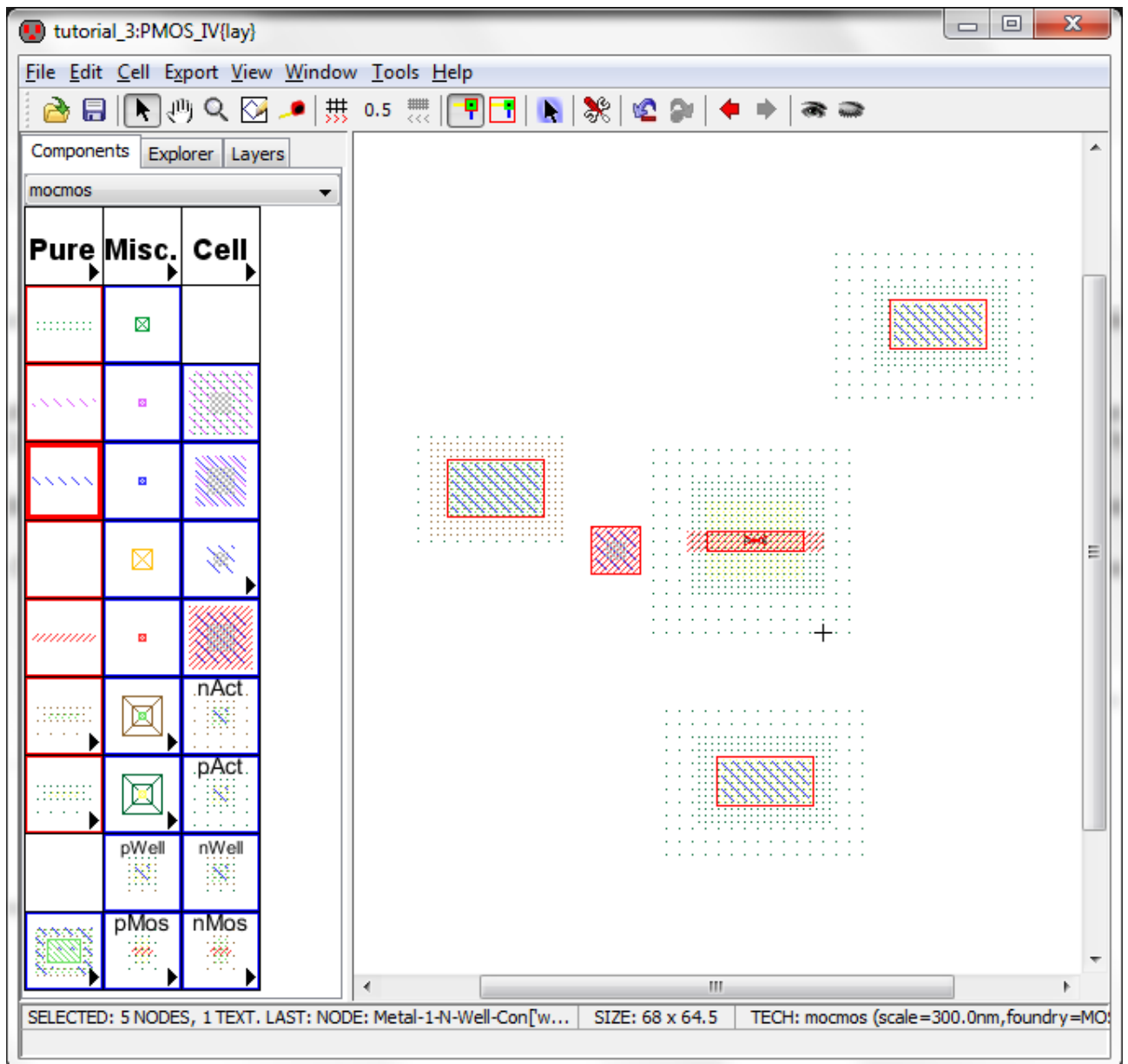


Рис. 2.31. Вставлення *Spice* моделі для вузла *PMOS* транзистора

Перевірка макета на помилки *DRC* укаже на наявність однієї помилки. *N*-область не достатньо довга (*y-size*). Змініть *y*-розмір *N-Well* вузла на значення 6.

Також, Ви можете спостерігати й інші помилки, спричинені близьким розміщенням вузлів один відносно одного, але ці помилки пропадуть одразу після того, як Ви з'єднаєте вузли. З'єднайте вузли разом, додайте контактні доріжки й експортуйте їх. Результат має бути таким, як на рис. 2.32.

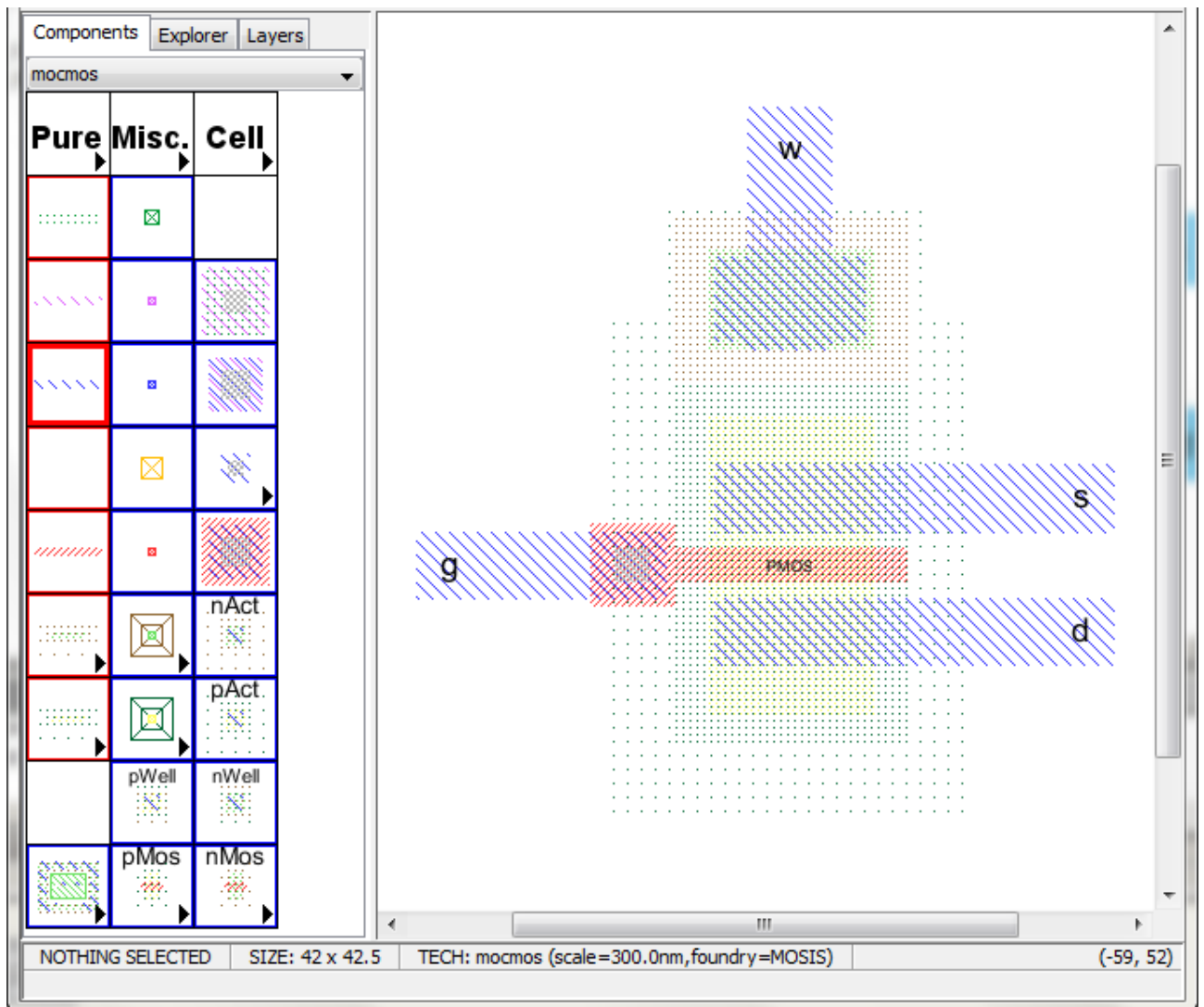


Рис. 2.32. Топологія *PMOS* транзистора із експортованими вузлами

Перевірте на наявність помилок *DRC* і *ERC*. Помилки не має бути.

Задамо *Spice* код для моделювання роботи транзистора. У вкладці *Components* обираємо комірку *Misc* → *Spice Code*, вставляємо її в робочу область. У властивостях *Spice* коду вказуємо параметри як показано нижче (рис. 2.33).

```

vs s 0 dc 0
vw w 0 dc 0
vg g 0 dc 0
vd d 0 dc 0
.dc vd 0 -5 -1m vg 0 -5 -1
.include C:\Electric\C5_models.txt

```



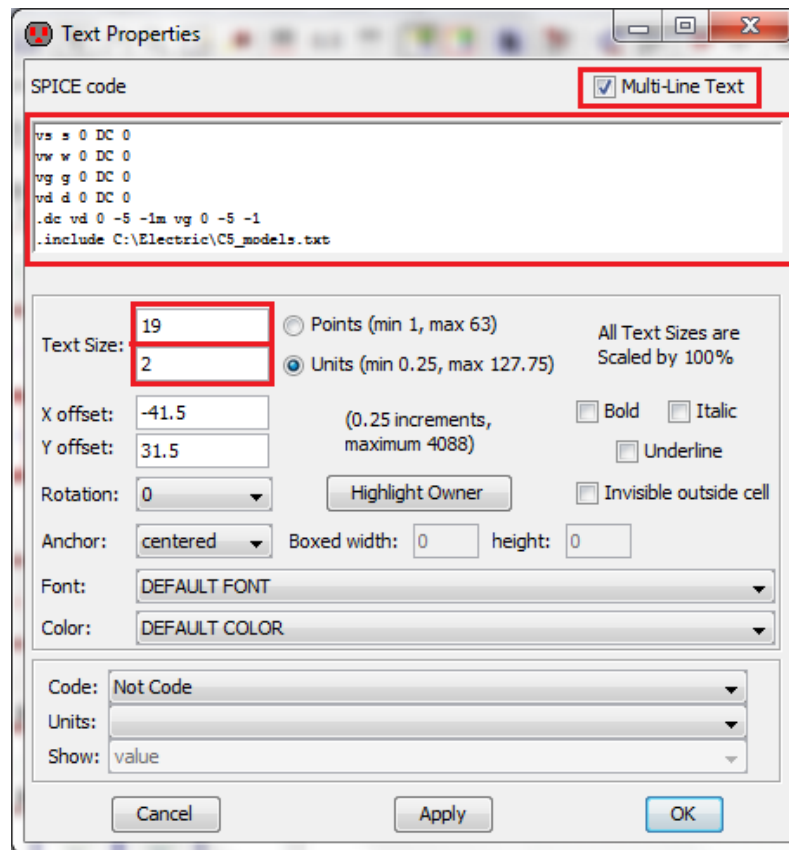


Рис. 2.33. Вставлення *Spice* коду для *PMOS* транзистора

Перші чотири рядки *Spice* коду задають значення сталої напруги на виводах транзистора. «*vs s 0 dc 0*» – стала напруга на виводі «*s*» (*source* – витік), між вузлами «*s*» і «нульовим» дорівнює 0 В. Аналогічно для виводів «*d*» (*drain* – стік), «*g*» (*gate* – заслін) і «*w*» (*well* – карман).

Команда «*.dc*» забезпечує розрахунки передатних характеристик за сталим струмом. Синтаксис команди:

*.dc sname1 vstart1 vstop1 vincr1 [sname2 vstart2 vstop2 vincr2]*

При задаванні команди *.dc* визначаються межі зміни напруг і струмів у незалежних джерелах напруги й струму з іменами *sname1* і *sname2*. Початкові значення – *vstart1* і *vstart2*; кінцеві значення – *vstop1* і *vstop2*; крок збільшень – *vincr1* і *vincr2*. Опис другого джерела з іменем *sname2* може бути відсутнім. У випадку введення двох джерел з іменами *sname1* і *sname2*, сигнал, обумовлений першим з них, буде змінюватися у своїх межах для кожного значення другого джерела.

Таким чином, рядок *Spice* коду «*.dc vd 0 -5 -1m vg 0 -5 -1*» означає, що для розрахунків передатних характеристик (*.dc*) підключається джерело сталої напруги (*v*) до стоку (*d*); напруга змінюється від 0 В до –5 В з кроком –1 мВ; також підключається джерело сталої напруги (*v*) до заслона (*g*); напруга змінюється від 0 В до –5 В з кроком –1 В.

Команда *.include* визначає ім'я файлу, вміст якого Ви бажаєте включити у вихідний опис без яких-небудь змін.

У результаті, робоче вікно програми матиме вигляд, як на рис. 2.34.

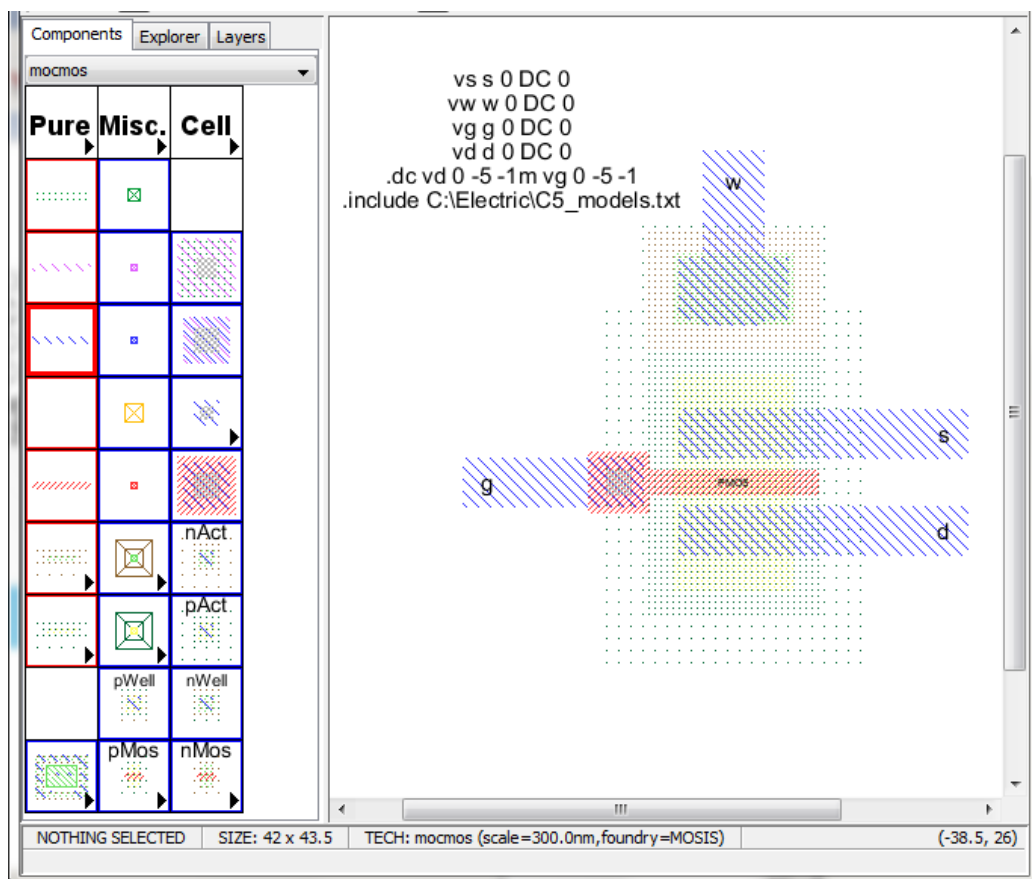


Рис. 2.34. Топологія *PMOS* транзистора із *Spice* кодом

Простежте, щоб в *C:\Electric* перебував файл *C5\_models.txt*. Без нього подальша робота неможлива.

Промодельюємо роботу *PMOS* транзистора за допомогою *LTspice*: *Tools* → *Simulation (Spice)* → *Write Spice Deck*. Обираємо залежність для побудови, наприклад *Is(Mpms@1)* (рис. 2.35).

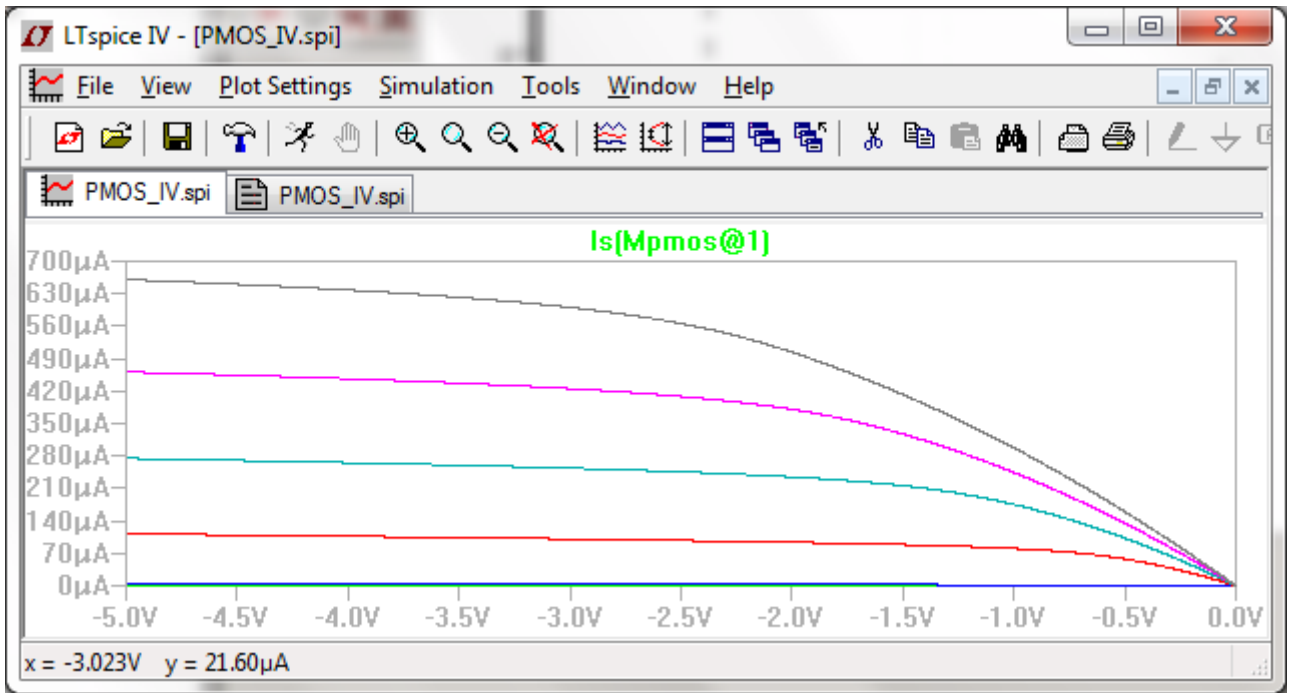


Рис. 2.35. ВАХ *PMOS* транзистора, побудована в *LTspice*

Закриваємо *LTspice*, і в сканері *Electric*, що відкрився, у вікні зонда вкладки *Explorer* обираємо потрібну нам залежність  $I_s(Mpmos@1)$  (рис. 2.36).

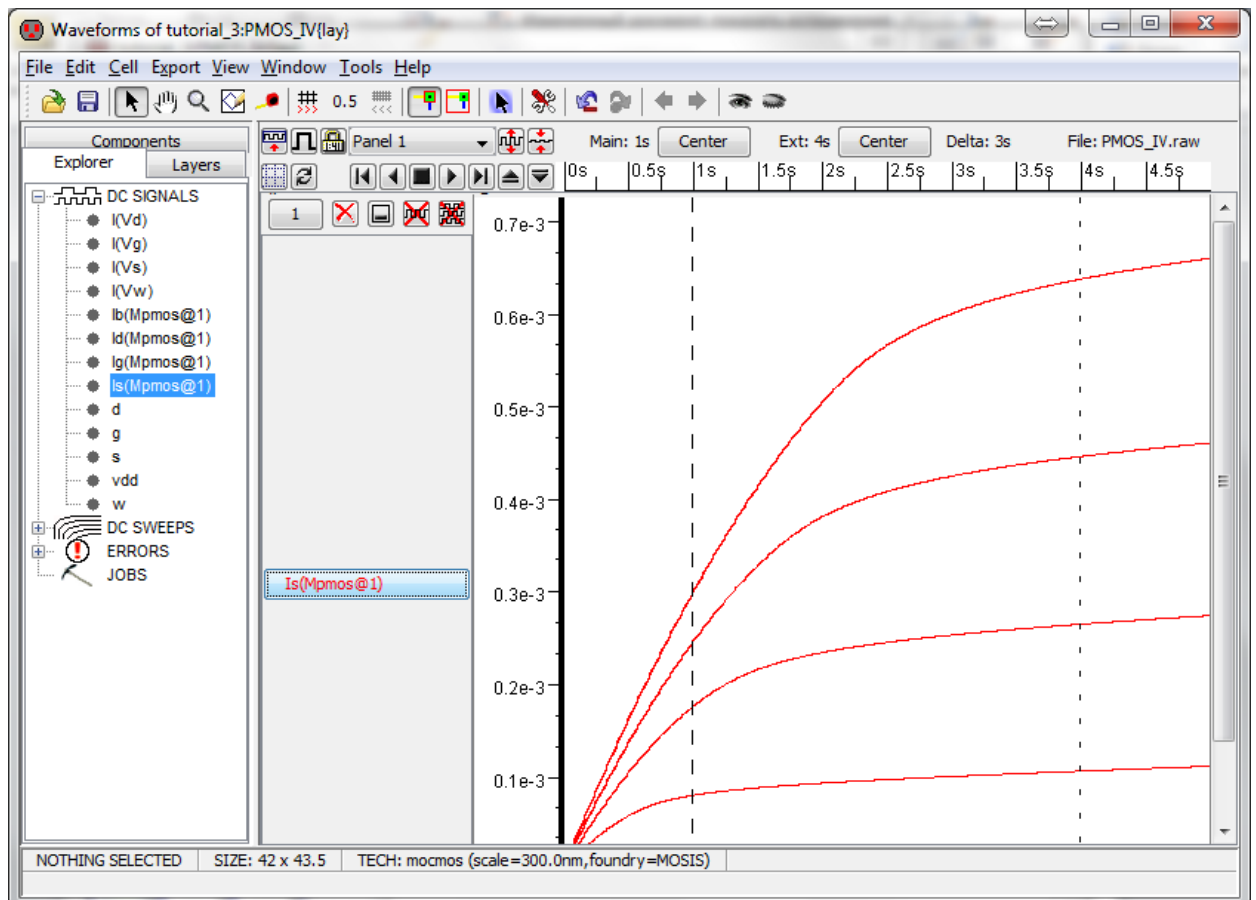


Рис. 2.36. ВАХ *PMOS* транзистора в *Electric*

Додайте *Spice* код для *NMOS* транзистора. І промоделуйте його роботу аналогічним способом (рис. 2.37, 2.38, 2.39).

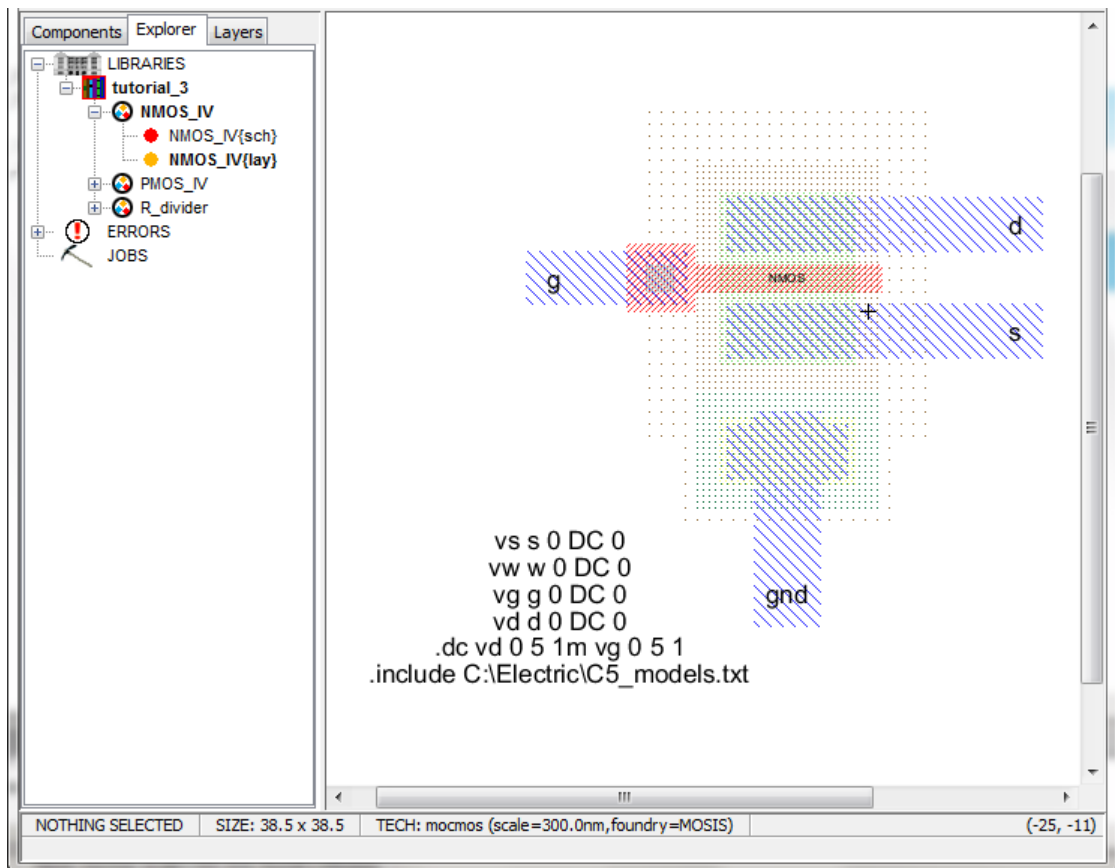


Рис. 2.37. Топологія *NMOS* транзистора із *Spice* кодом

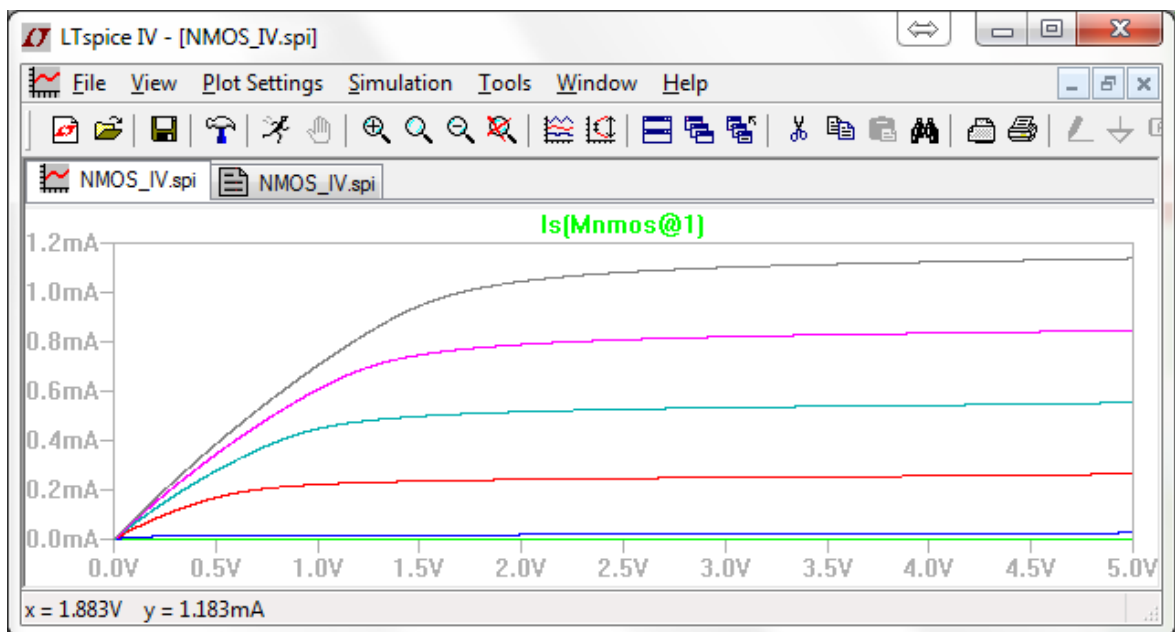


Рис. 2.38. Моделювання характеристики за допомогою *LTspice*

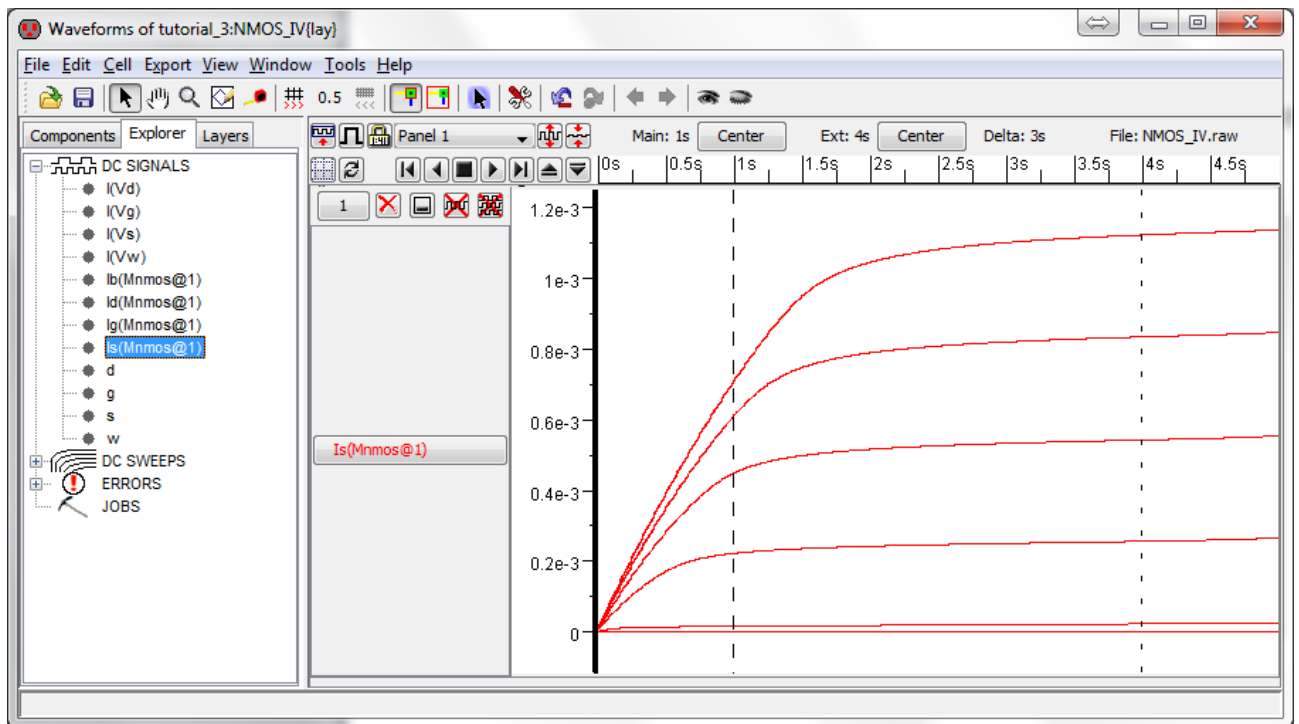


Рис. 2.39. Моделювання характеристики за допомогою сканера *Electric*

Між схемами й топологіями транзисторів є невідповідності. Усунемо їх (помилки *NCC*). Скопіюйте *Spice* код з *NMOS\_IV{lay}* і вставте його в *NMOS\_IV{sch}*. Змініть розмір тексту *Spice* коду зі значення 2 на 0,5 (рис. 2.40).

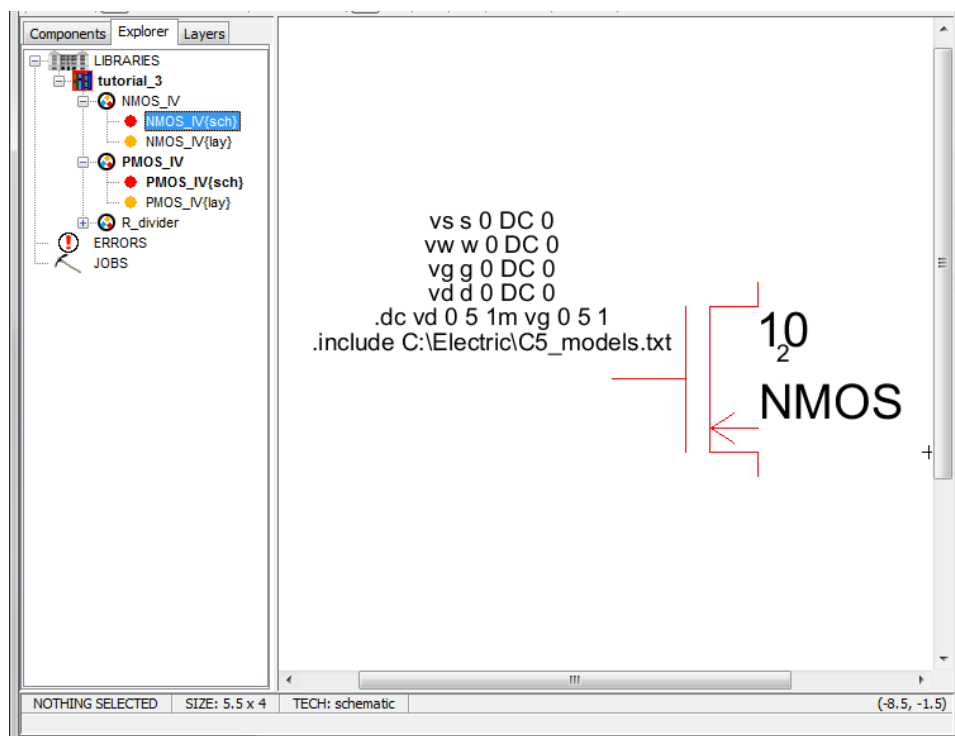


Рис. 2.40. Схематичне позначення *NMOS* транзистора із *Spice* кодом

Повторіть ті ж дії у групі комірок *PMOS* (рис. 2.41).

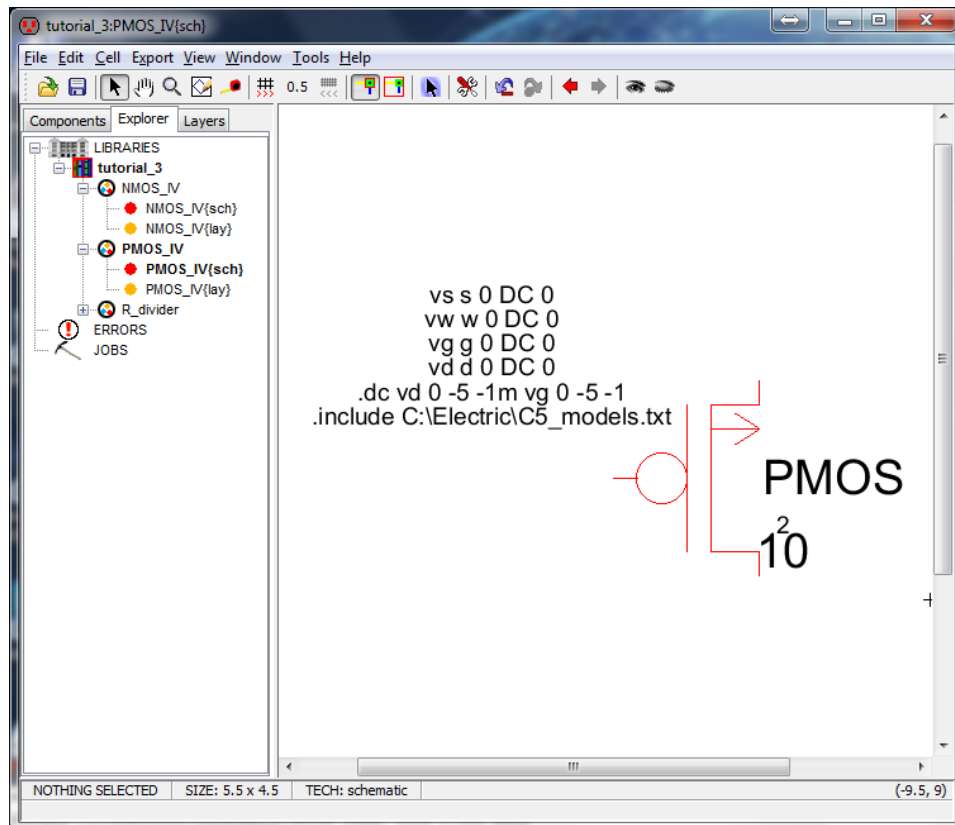


Рис. 2.41. Схематичне позначення *PMOS* транзистора із *Spice* кодом

У комірку *NMOS\_IV{sch}* із вкладки *Components* додайте вузол заземлення.

Для зручності відобразимо сітку *Window* → *Toggle Grid* або *Ctrl+G*.

Оскільки символ, що позначає заземлення, може бути більшим, ніж *MOSFET* транзистор, відмасштабуйте його, вибравши меню *Edit* → *Size* → *Interactively* або натиснуть комбінацію клавіш *Ctrl+B*. З'єднайте «землю» із транзистором і додайте металеві виводи до витоків, стоку й заслону транзистора. Далі потрібно експортувати контакти відповідно до топології (правила відповідності *NCC*). Для того, щоб побачити всі контакти, виділіть всі вузли схеми (виділіть мишкою або скористайтесь комбінацією клавіш *Ctrl+A*), результат наведено на рис. 2.42.

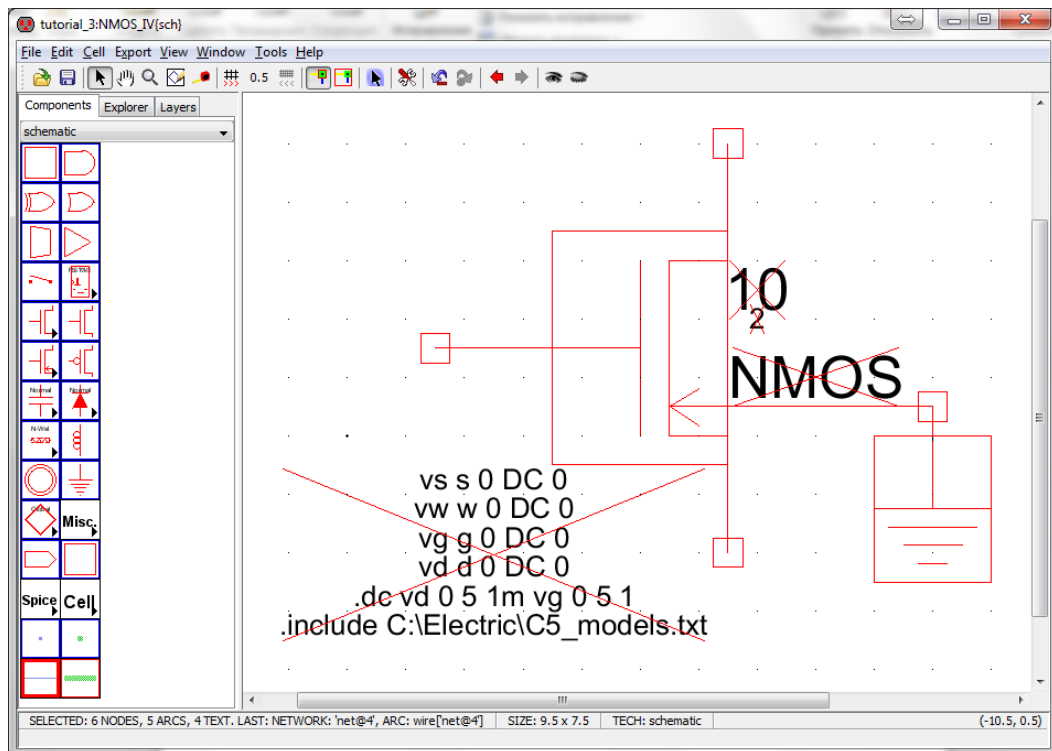


Рис. 2.42. Виділення всіх елементів схеми

Експортуйте контакти відповідно до топології, для цього зайдіть в меню *Export* → *Create Export* (або *Ctrl+E*) (рис. 2.43). Для більш зручного виділення використовуйте *Ctrl+left-click*.

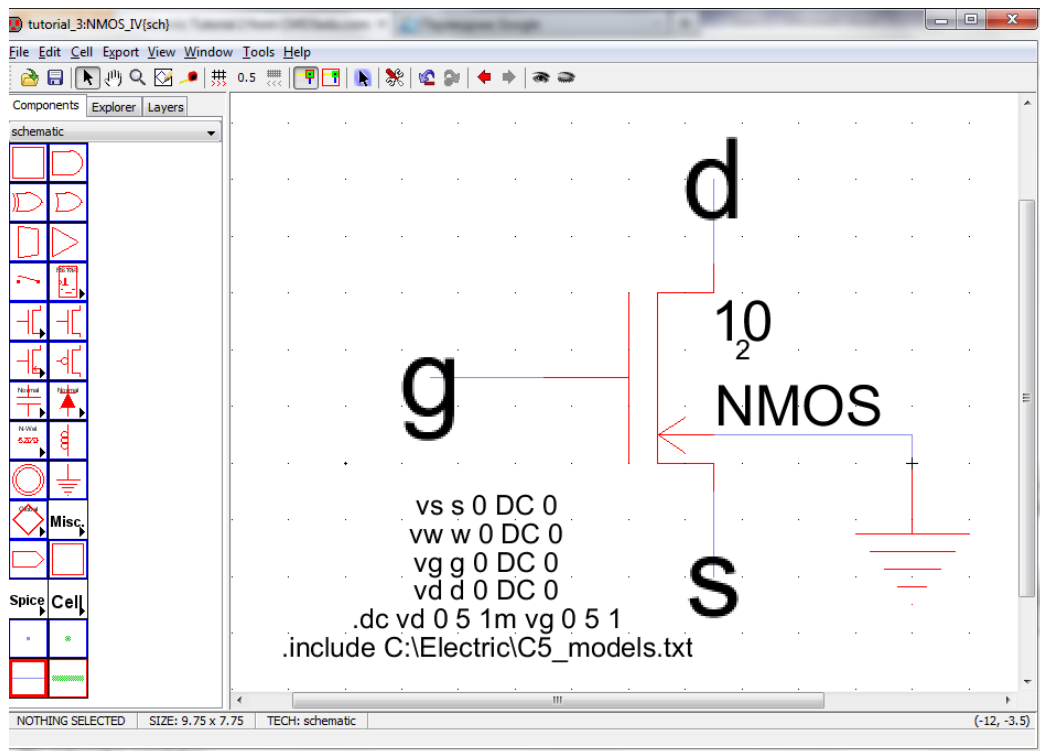


Рис. 2.43. Схема NMOS транзистора з експортованими вузлами

Перевірте схему на відповідність *NCC*. Помилки не має бути.

Те ж саме потрібно зробити з *PMOS* транзистором. Результат має бути таким, як показано на рис. 2.44.

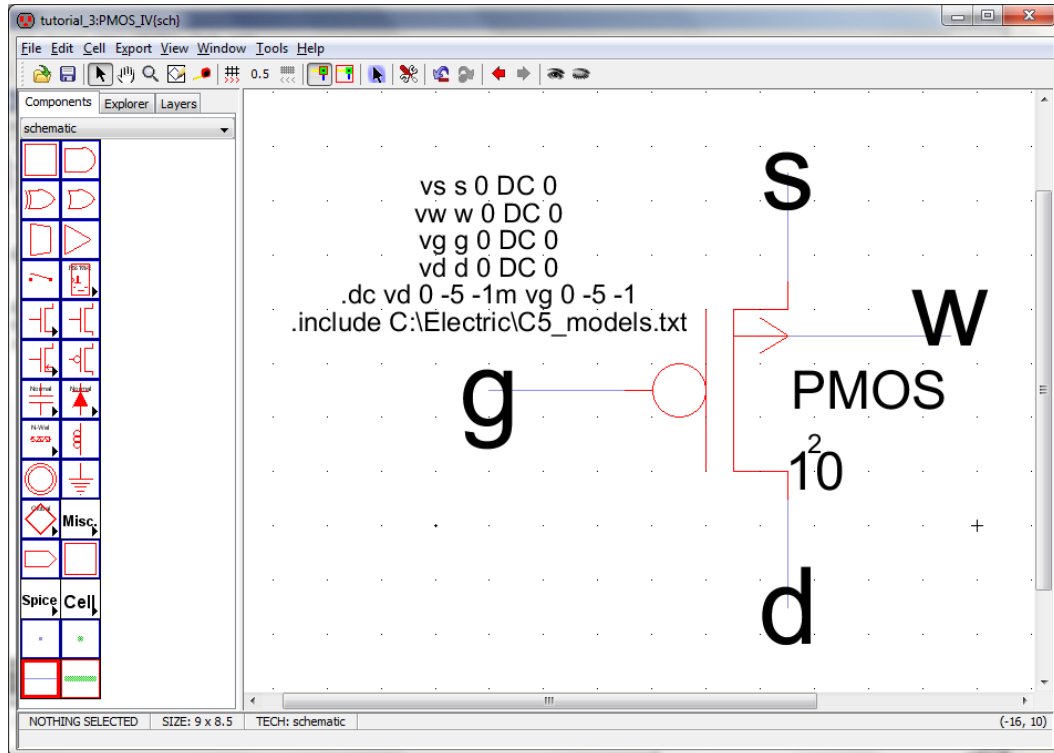


Рис. 2.44. Схема *PMOS* транзистора з експортованими вузлами

Перевірте схему на наявність помилок *DRC* і *NCC*.

### Варіанти завдань

№ варіанта	1	2	3	4	5	6
<i>NMOS</i>						
<i>W</i>	11	17	13	35	23	30
<i>L</i>	2	3	4	5	6	5,5
<i>PMOS</i>						
<i>W</i>	5	10	10	60	12	80
<i>L</i>	2	12	10	5	7	8



## Контрольні питання

1. Які ефекти лежать в основі роботи МДН транзисторів?
2. Від чого залежить значення електричного опору каналу?
3. Чим обумовлене утворення вбудованого заряду? Які наслідки утворення вбудованого заряду?
4. Які функції може виконувати МДН транзистор в мікросхемі?
5. Назвіть недоліки використання пасивних напівпровідникових інтегральних елементів.
6. Які електричні параметри МДН транзистора впливають на його швидкодію? Як вони впливають?
7. Як можна знизити порогову напругу МДН транзистора?
8. Що таке рівні Тамма?
9. Як рівні Тамма впливають на порогову напругу МДН транзистора?
10. Як збільшити крутизну стоко-заслонної характеристики МДН транзистора? Для чого це потрібно?
11. Які паразитні ємності та опори виникають в МДН транзисторі?
12. Чому струм стоку повільно змінюється при швидкій зміні вхідної напруги?
13. Як зменшити параметри паразитних елементів у МДН транзисторі?
14. Які функції виконує товстий шар оксиду за межами області каналу?
15. Які переваги мікросхем на *n*-канальних МДН транзисторах перед мікросхемами на *p*-канальних МДН транзисторах?
16. Яким чином багат шаровий підзаслонний діелектрик дозволяє керувати пороговою напругою МДН транзистора?
17. Назвіть переваги структури МДН транзистора з багат шаровим діелектриком нітрид кремнію – оксид кремнію.
18. Чому як підзаслонний діелектрик не використовують тільки нітрид кремнію?

19. Назвіть переваги структури МДН транзистора з багат шаровим діелектриком оксид алюмінію – оксид кремнію.
20. Чому в МДН транзисторах з алюмінієвим заслоном необхідно створювати області перекриття заслона з областями витоку і стоку?
21. Які негативні наслідки створення в МДН транзисторах з алюмінієвим заслоном області перекриття заслона з областями витоку і стоку?
22. Чим обумовлені труднощі зменшення розмірів областей перекриття заслона з областями витоку і стоку в МДН транзисторах з алюмінієвим заслоном?
23. Які переваги були отримані за використання полікремнію як матеріалу заслона?
24. Яким чином використання полікремнію як матеріалу заслона дозволило перейти до застосування джерел живлення з робочою напругою 5 В?

### Практичне заняття №3

#### Моделювання роботи КМОН інвертора

Моделювати КМОН інвертор будемо з використанням налаштувань під технологічний процес C5. Збережіть бібліотеку *tutorial\_2.jelib* як *tutorial\_3.jelib* у папку *C:\Electric\*. Також, простежте, щоб у папці *C:\Electric\* перебував файл із *spice* моделями *C5\_models.txt*.

Запустіть *Electric* і відкрийте *tutorial\_3.jelib*. Відкрийте комірку *NMOS\_IV{sch}* і скопіюйте вузол *NMOS* у буфер (рис. 3.1).

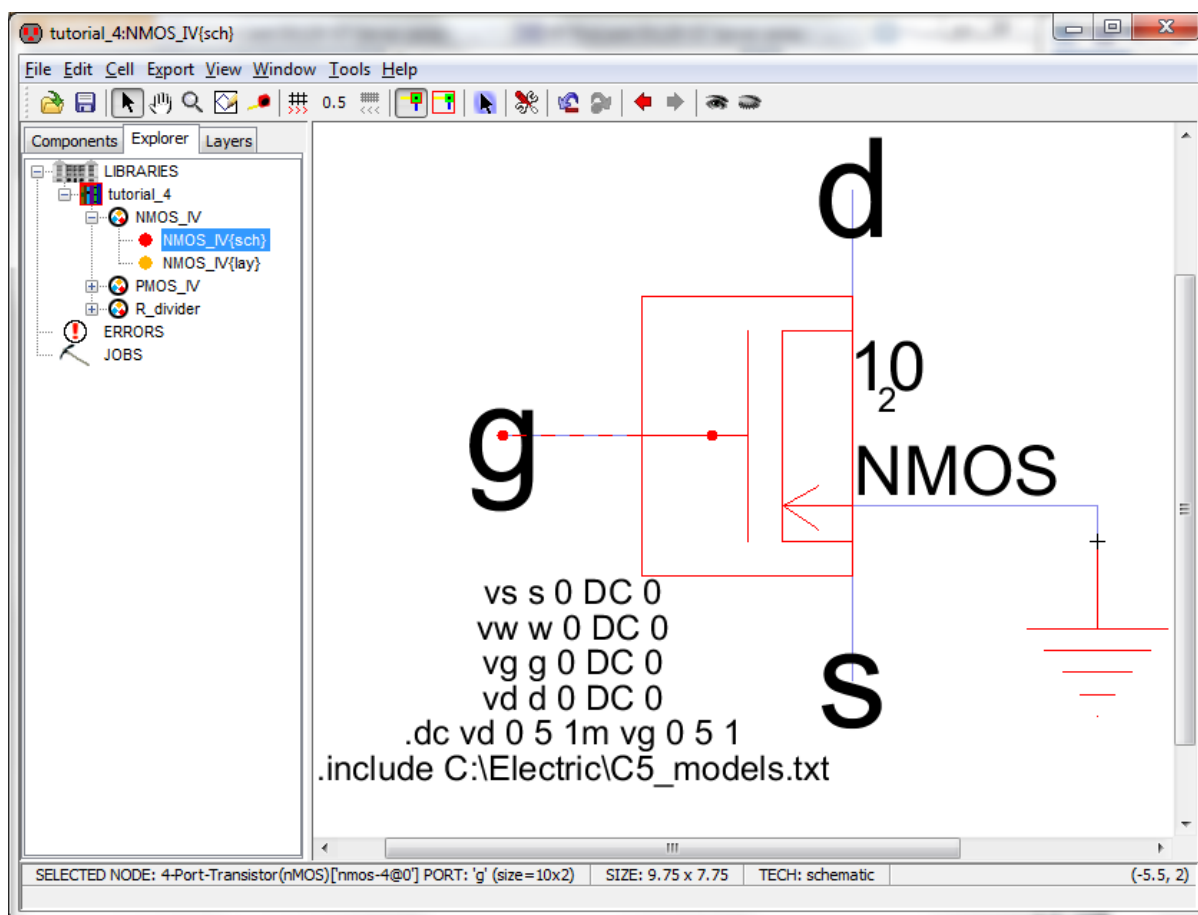


Рис. 3.1. Вузол *NMOS* у комірці *NMOS\_IV{sch}*

Створіть нову комірку (*Ctrl+N*), виберіть схемний перегляд (*schematic*) і назвіть її *inv\_20\_10* (КМОН/CMOS інвертор із шириною каналу *NMOS* транзистора 20 і *PMOS* транзистора 10) (рис. 3.2). Вставте скопійований раніше вузол (рис. 3.3).

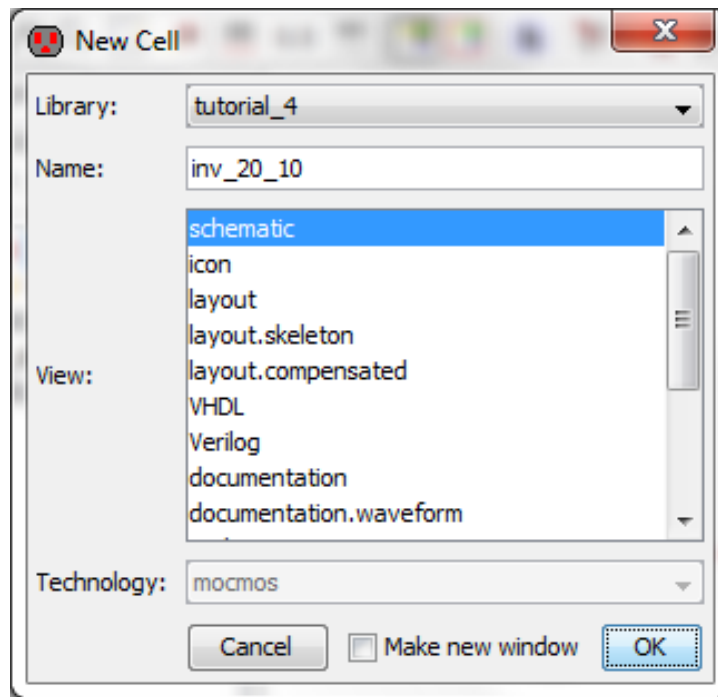


Рис. 3.2. Створення комірки *inv\_20\_10*

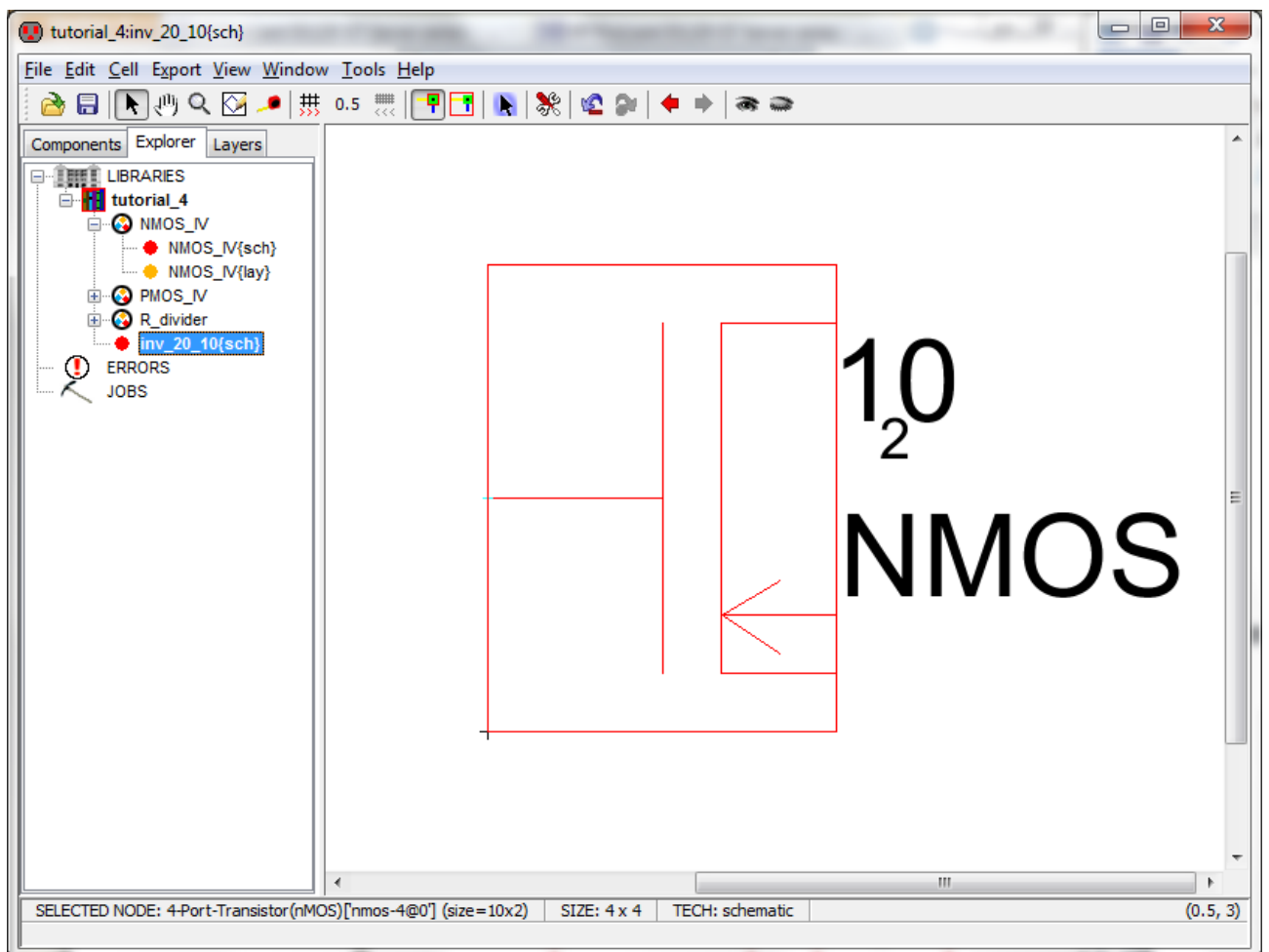


Рис. 3.3. Вузол *NMOS* у комірці *inv\_20\_10*

Необхідно позбутися виводу, який ми використовували для підключення  $p$ -підкладки до «землі», тому що підкладка завжди буде на «землі».

Для цього виділіть вузол  $NMOS$ , зайдіть у меню *Edit* → *Change* (клавіша «C») і змініть вузол, як показано на рис. 3.4. Зверніть увагу, що спочатку треба натиснути *Change*, а потім *Done*.

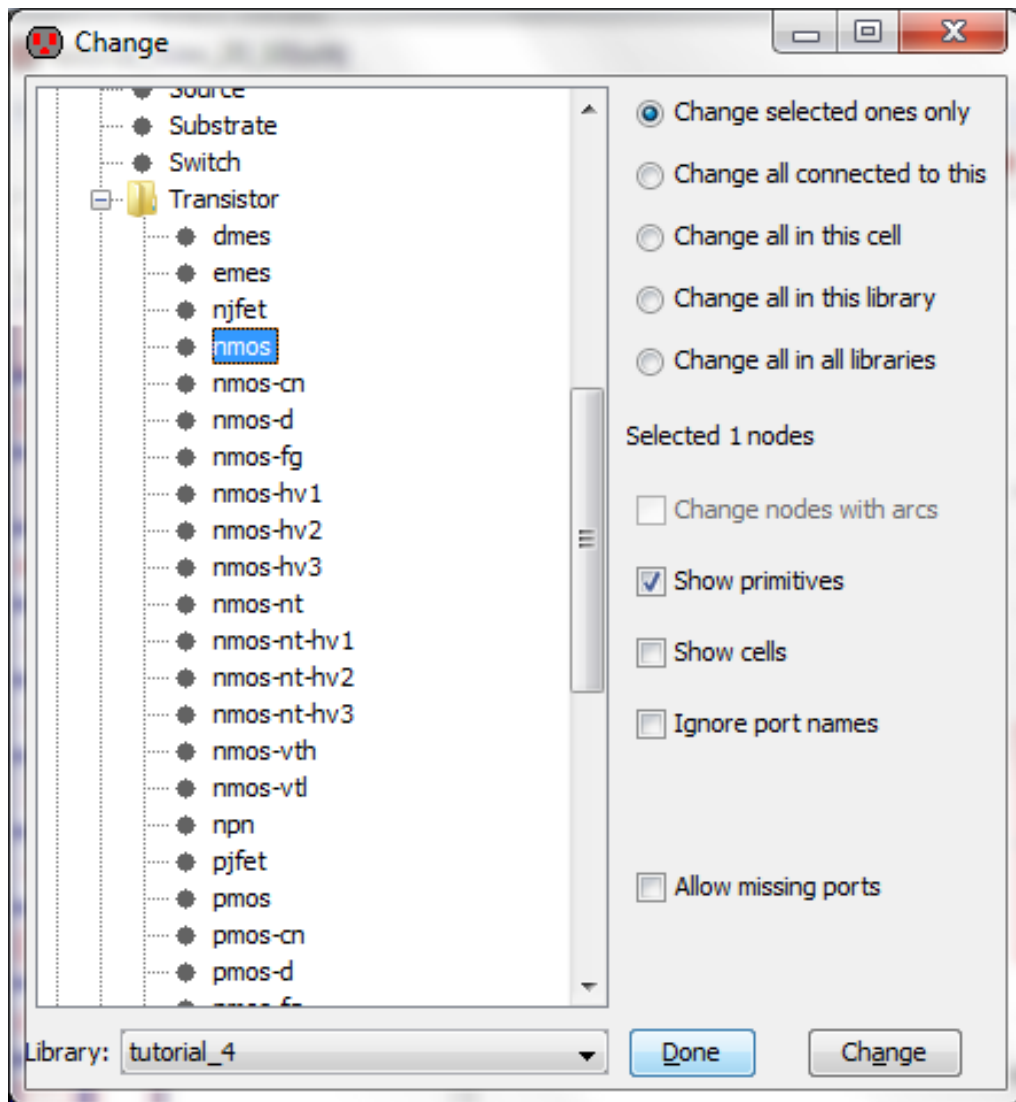


Рис. 3.4. Зміна представлення вузла  $NMOS$

Зробіть те ж саме з вузлом  $PMOS$  транзистора комірки  $PMOS_{IV}$ . Результат наведено на рис. 3.5.

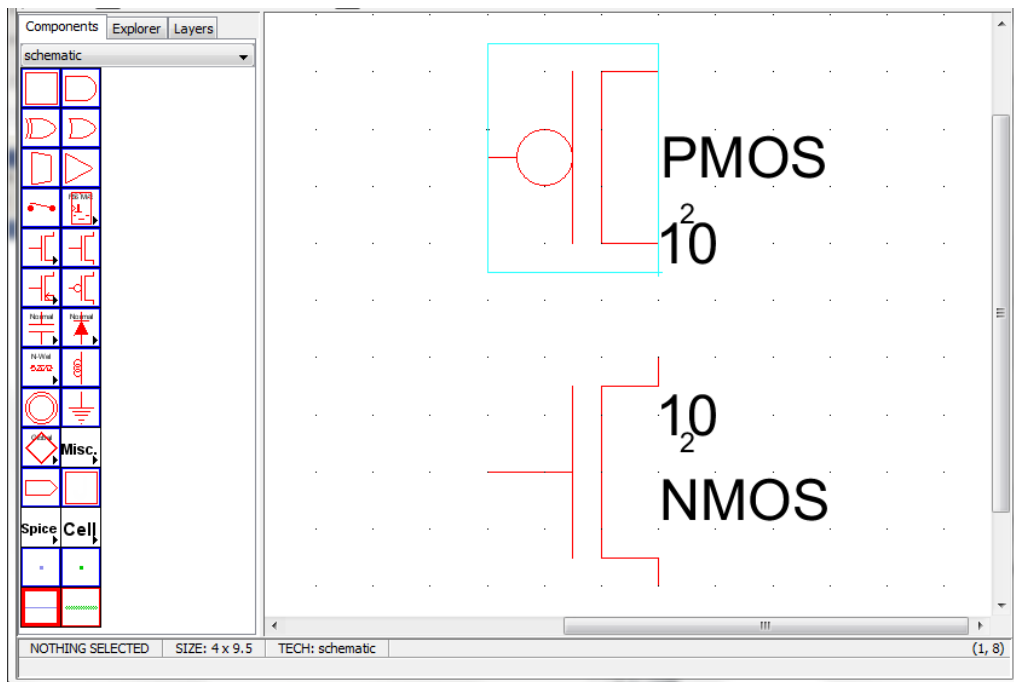


Рис. 3.5. Транзистори *NMOS* і *PMOS*, представлені як триполюсники

Змініть у властивостях (*Ctrl+I*) ширину *PMOS* транзистора з 10 на 20 (рис. 3.6).

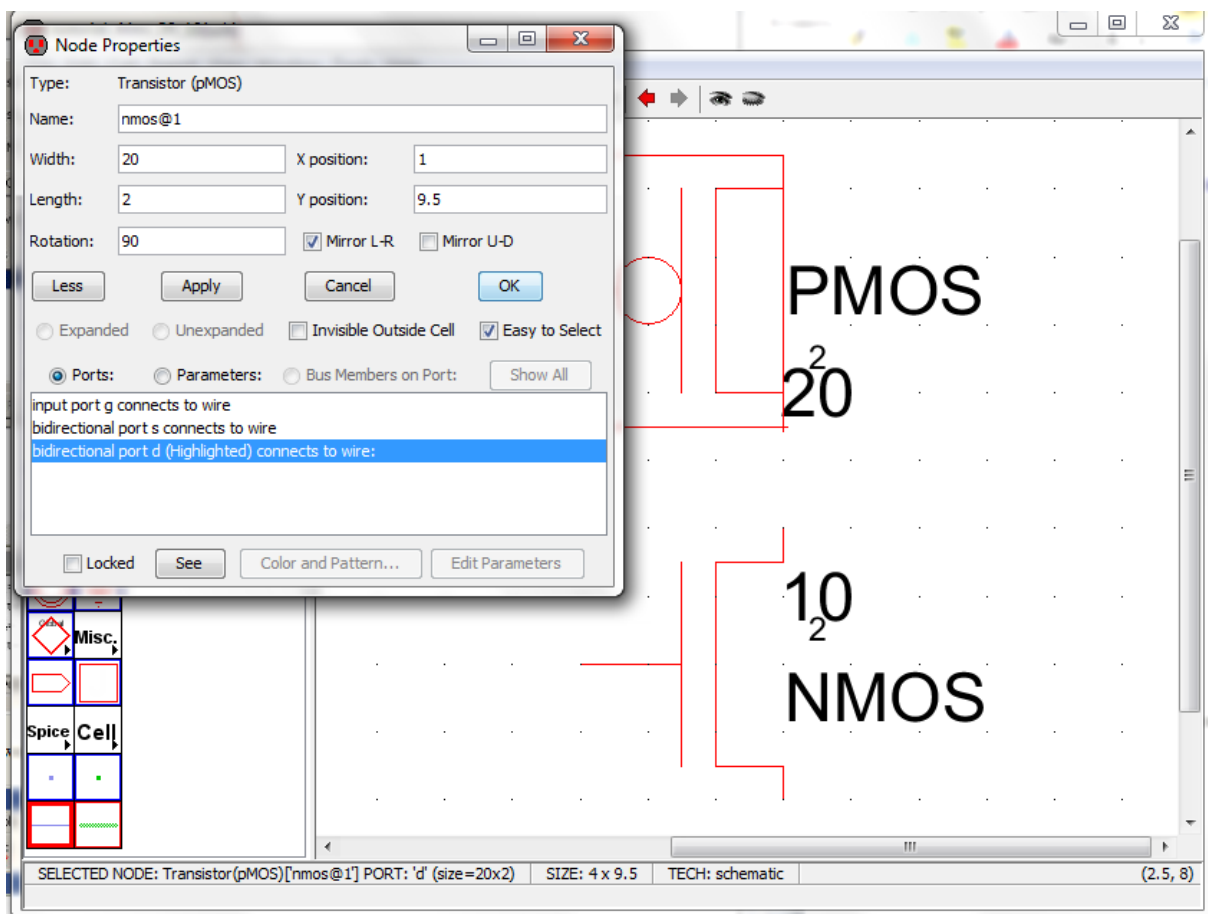


Рис. 3.6. Зміна параметрів транзистора *PMOS*

Увімкніть відображення сітки (*Ctrl+G*). Вставте символи живлення (*Power*) і заземлення (*Ground*), а також вставте вузли *Off-Page*, які будуть позначати інтерфейси входу й виходу інвертора. З'єднаєте вузли між собою, як зазначено на рис. 3.7.

Експортуйте (*Ctrl+E*) вузли:

- Вхід  $\rightarrow$  *in* (лівий порт лівого вузла *Off-Page*);
- Вихід  $\rightarrow$  *out* (правий порт правого вузла *Off-Page*).

Результат наведено на рис. 3.7.

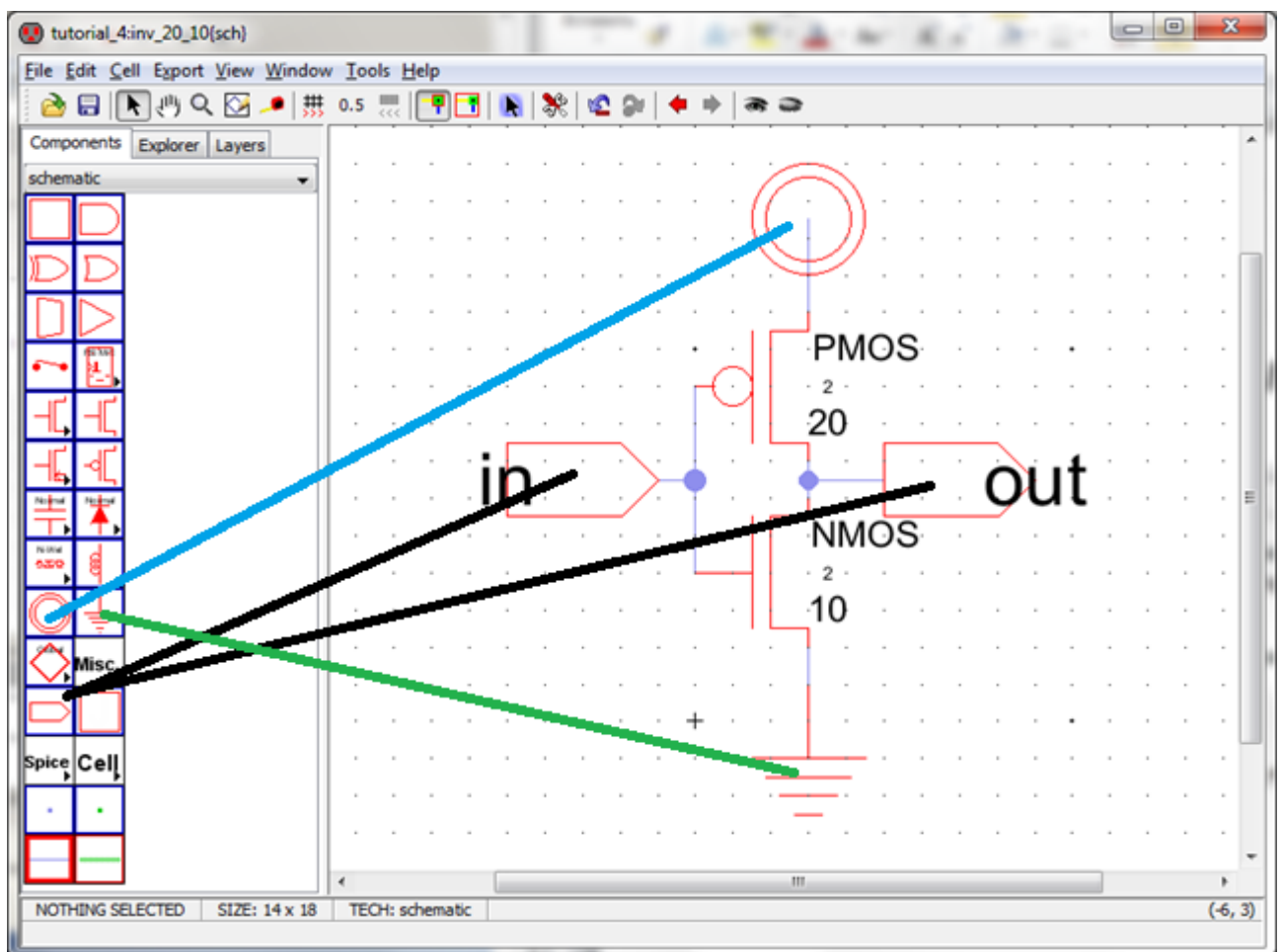


Рис. 3.7. КМОП інвертор з імпортованими вузлами

Не забувайте перевіряти схему на наявність помилок.

Створимо піктограму для даної схеми, меню *View*  $\rightarrow$  *Make Icon View*. Піктограма відобразиться у правому верхньому куті (рис. 3.8). Зверніть увагу, що у вкладці *Explorer* з'явилася комірка із синьою міткою

*inv\_20\_10{ic}* (рис. 3.9). Це свідчить про те, що в групі комірок є піктограма, схемне позначення якої є еквівалентом однойменної комірки зі схемою.

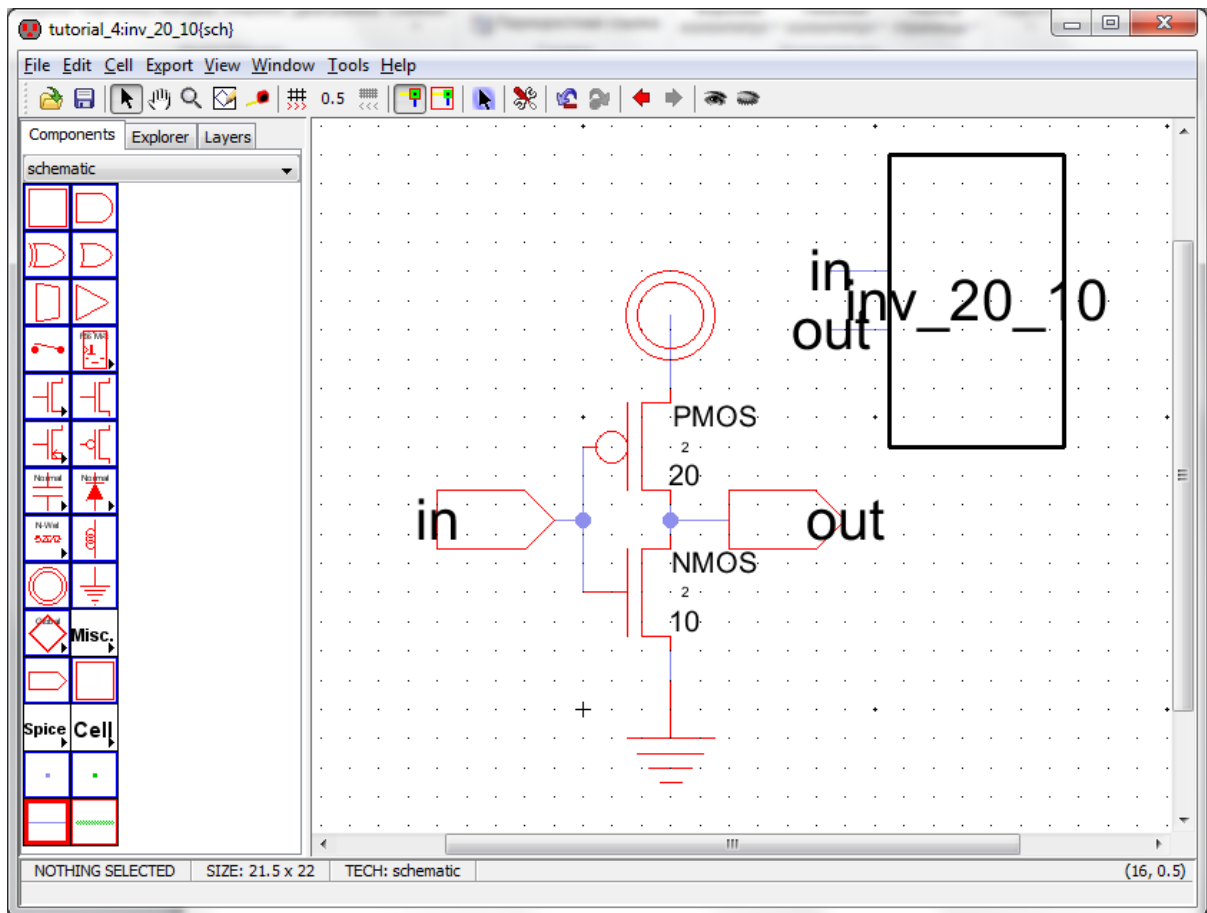


Рис. 3.8. Створення піктограми

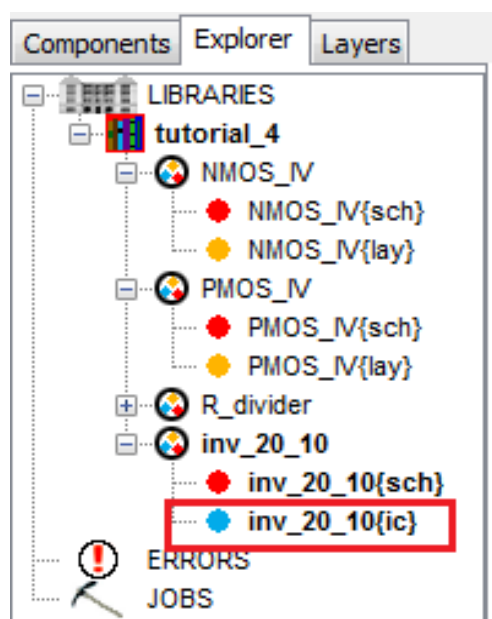


Рис. 3.9. Комірка із синьою міткою – комірка піктограми



Таким чином, ми маємо деяку ієрархію Піктограма→Схема. Для того щоб перемикатися на нижні рівні ієрархії можна використовувати меню *Cell*→*Down Hierarchy*→*Down Hierarchy* (або *Ctrl+D*), на верхні – *Cell*→*Up Hierarchy*→*Up Hierarchy* (або *Ctrl+U*).

Приведемо піктограму до схемного позначення інвертора. Для цього потрібно перейти на верхній рівень ієрархії схеми, а саме в комірку *inv\_20\_10{ic}*. Видаліть прямокутник і текст у ньому, залишивши тільки інтерфейси входу й виходу (рис. 3.10).

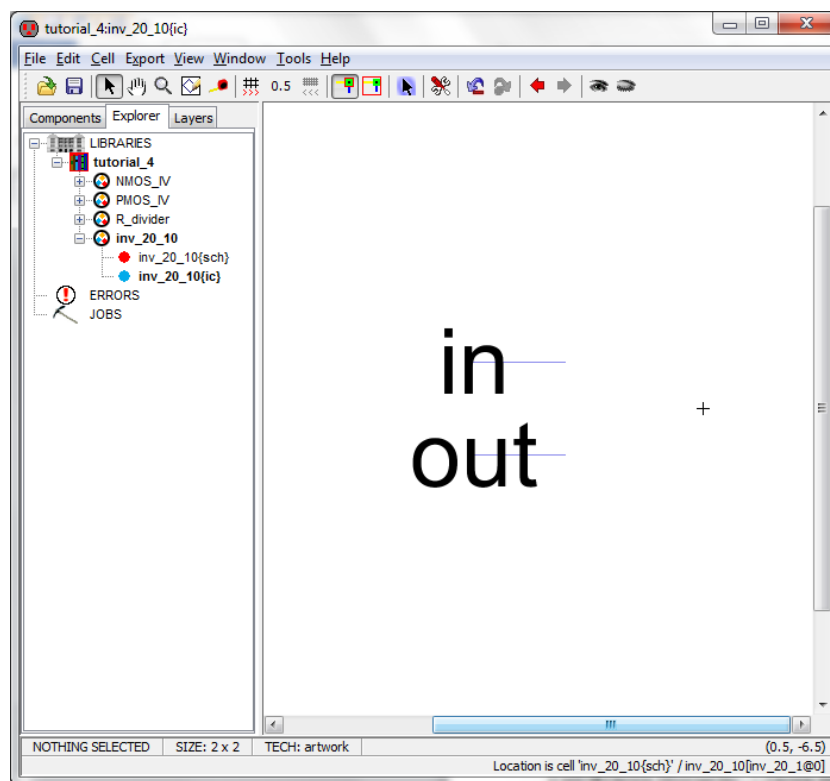


Рис. 3.10. До пояснення редагування піктограми

Виділіть вихід піктограми «out», поверніть його (*Ctrl+J*) і розташуйте, як показано на рис. 3.11.

Перейдіть у вкладку *Components* і перетягніть у робочу область трикутник і коло. Для повороту використовуйте комбінацію клавіш *Ctrl+J*, для зміни геометричних розмірів – *Ctrl+B*. Розміри також можна правити й через властивості об'єкта (*Ctrl+I*), змініть радіус кола на значення 1 (рис. 3.12). Розташуйте трикутник і коло так, як показано на рис. 3.13.

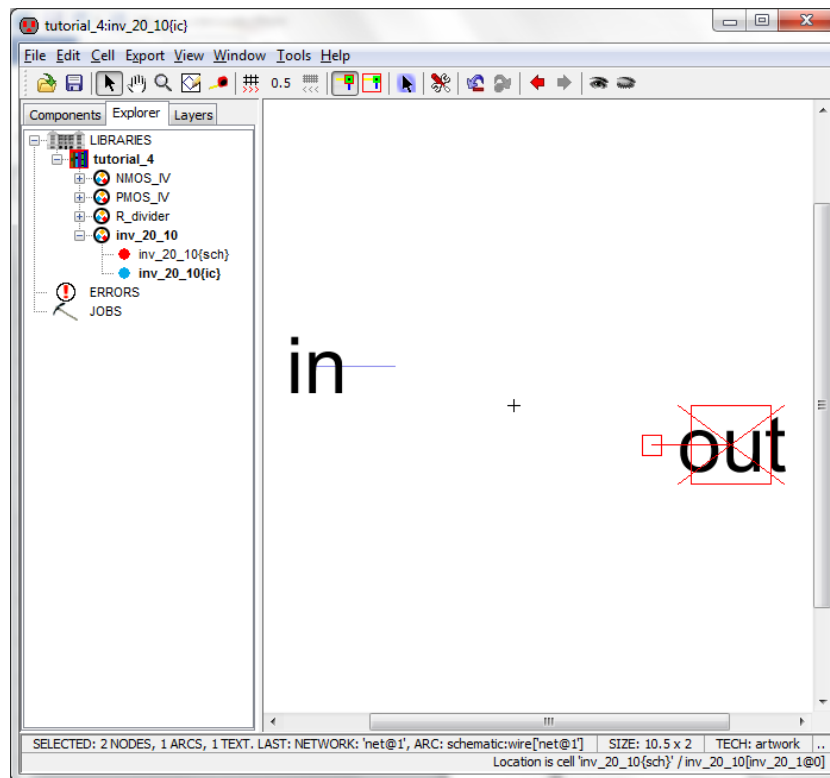


Рис. 3.11. До пояснення редагування піктограми

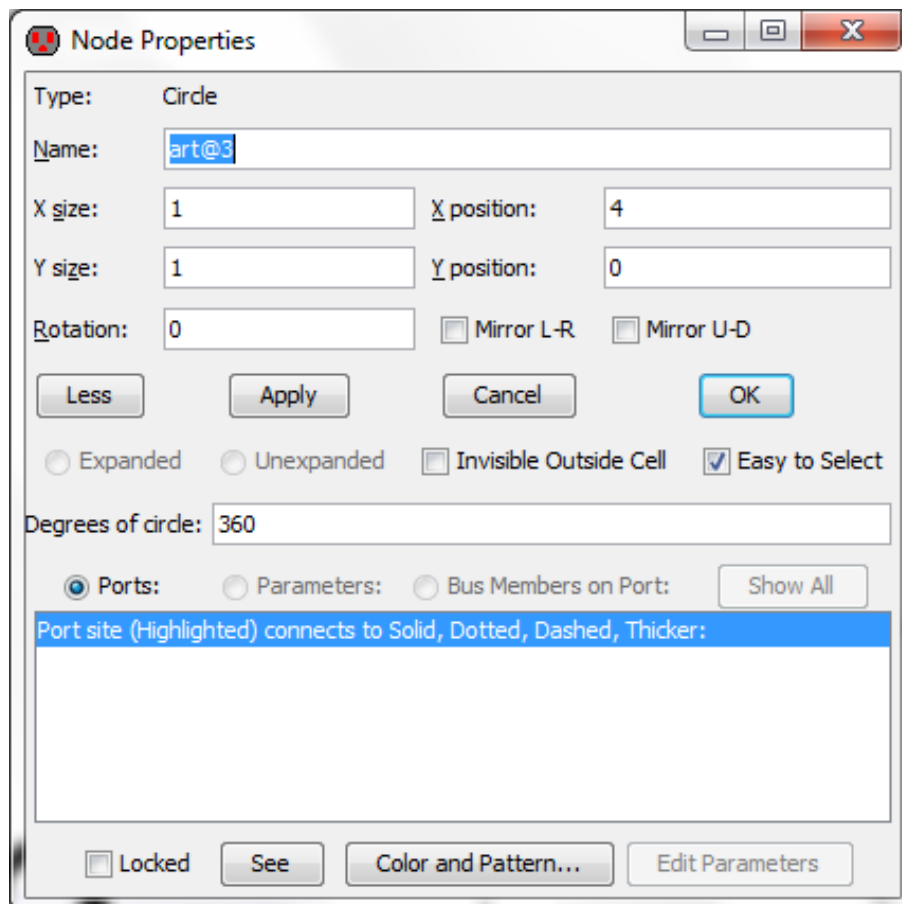


Рис. 3.12. Редагування властивостей об'єкта

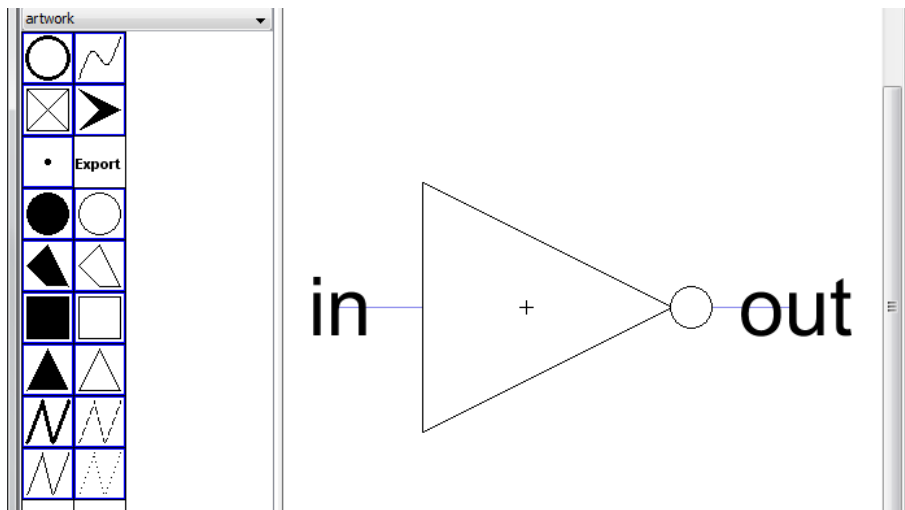


Рис. 3.13. Піктограма КМОН інвертора

Піктограма інвертора закінчена. Поверніться у комірку  $inv\_20\_10\{sch\}$ .

Промодельюємо роботу інвертора. Створіть нову схемну комірку, назвіть її  $inverter\_sim$ . Вставте в  $inverter\_sim\{sch\}$  піктограму інвертора (рис. 3.14). Це можна зробити двома способами:

1. Перетягнути із вкладки *Explorer* комірку  $inv\_20\_10\{ic\}$  в робочу область;
2. У вкладці *Components*  $\rightarrow$  *Cell*  $\rightarrow$   $inv\_20\_10$ .

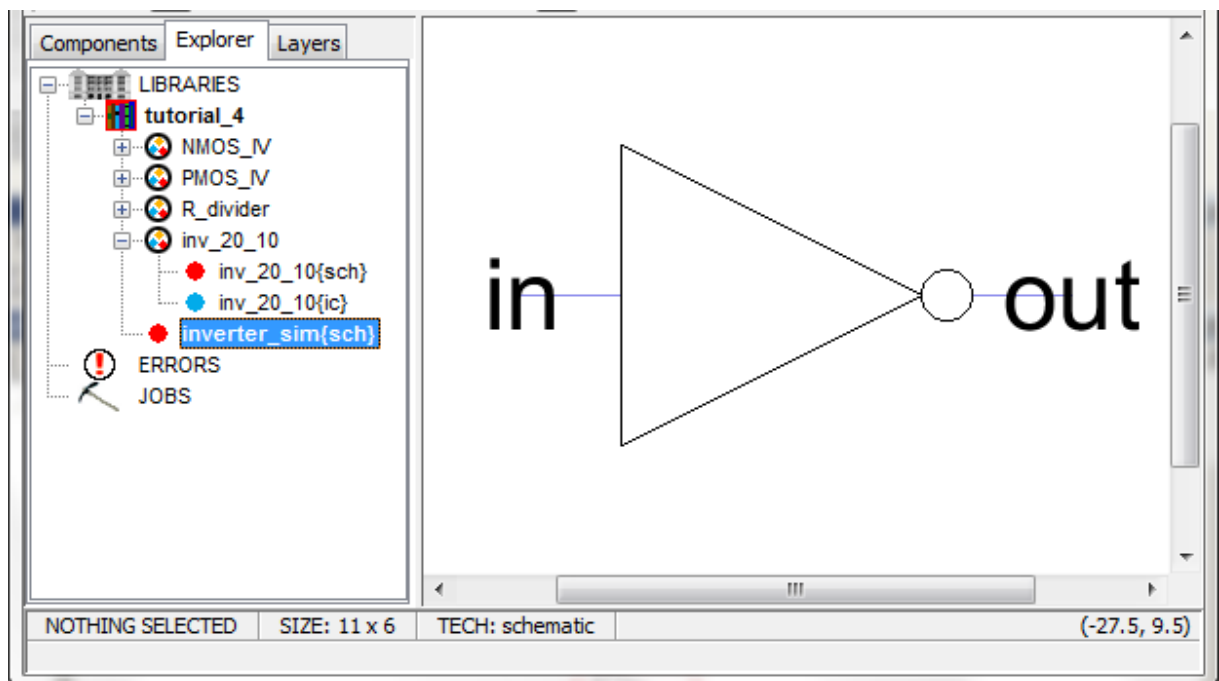


Рис. 3.14. Піктограма інвертора в комірці  $inverter\_sim\{sch\}$

Додайте дуги на вхід і вихід інвертора. Також, додайте *Spice* код *Components* → *Misc* → *Spice Code* (рис. 3.15).

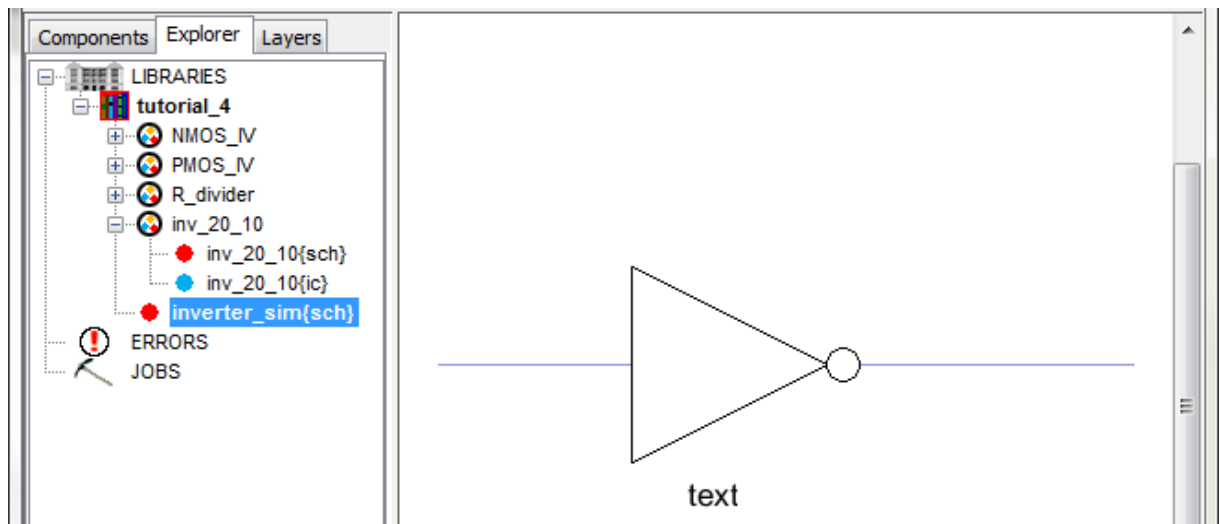


Рис. 3.15. Дуги та область *Spice* кода для моделювання інвертора

Промаркуйте дуги (назвіть їх у властивостях дуги *Ctrl+D*), і відредагуйте *Spice* код (у властивостях *spice* коду). Результат наведено на рис. 3.16.

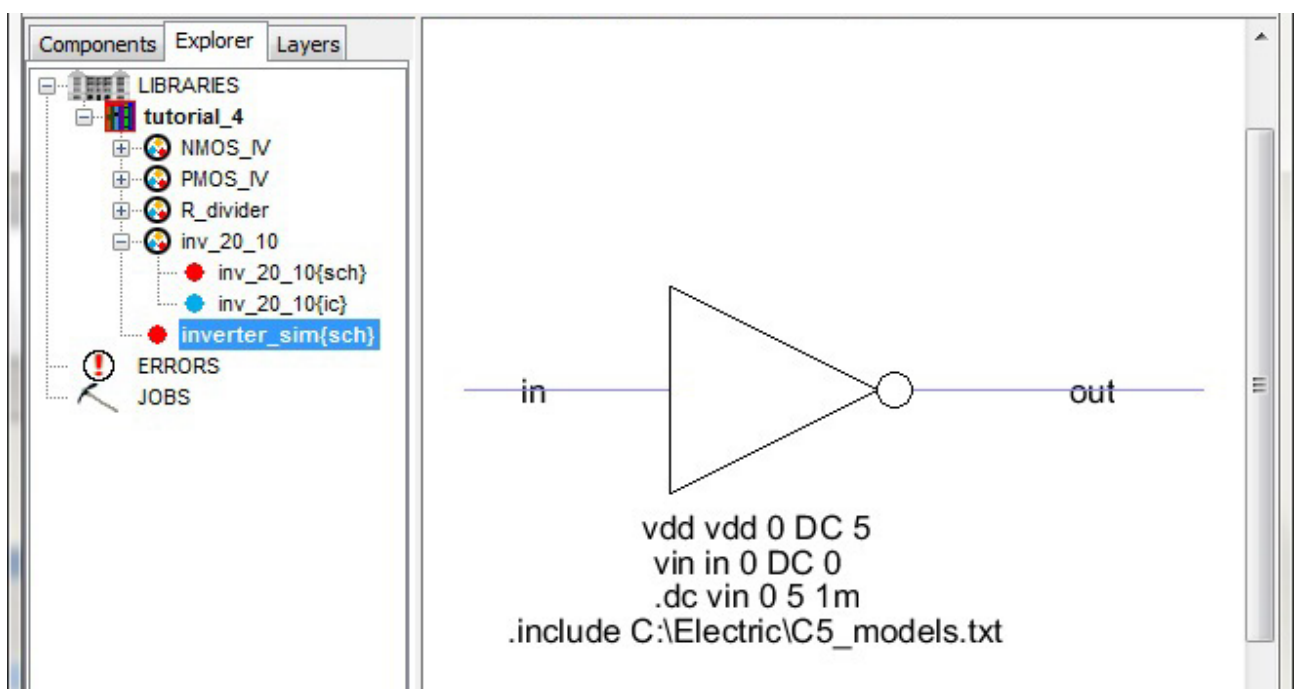


Рис. 3.16. Промарковані дуги та *Spice* код для моделювання інвертора

Промодельюємо роботу інвертора *Tools* → *Simulation (Spice)* → *Write Spice Deck*. У результаті одержимо залежність  $U_{out}(U_{in})$  (рис. 3.17).

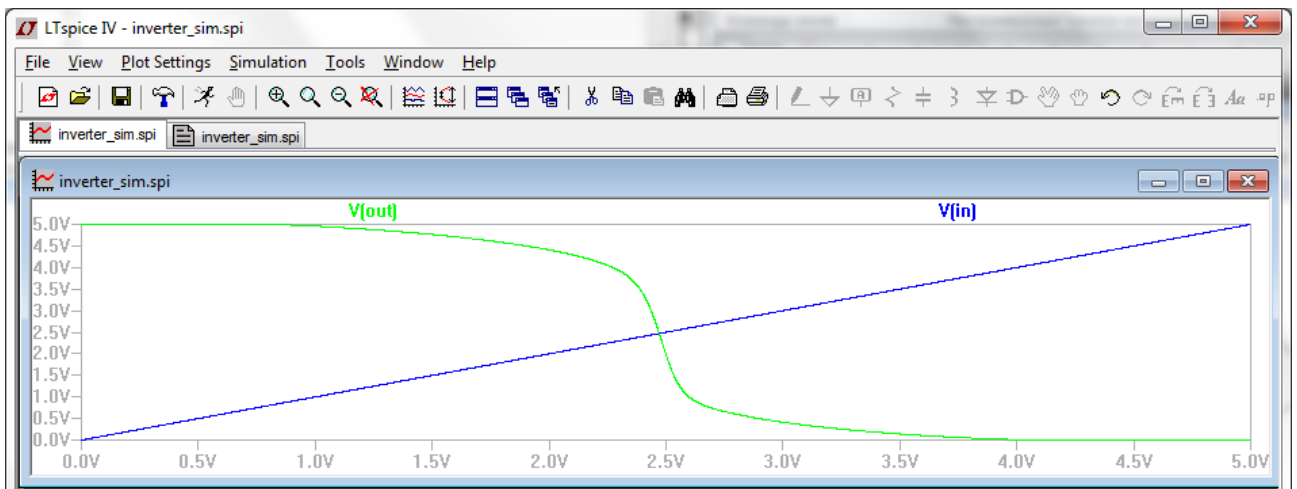


Рис. 3.17. Передаткова характеристика КМОН інвертора в *LTspice*

Закрийте *LTspice* і продовжимо роботу в сканері *Electric* (він запускається відразу ж після закриття *LTspice*). У сканері *Electric* відобразить залежність  $U_{out}(U_{in})$  (рис. 3.18).

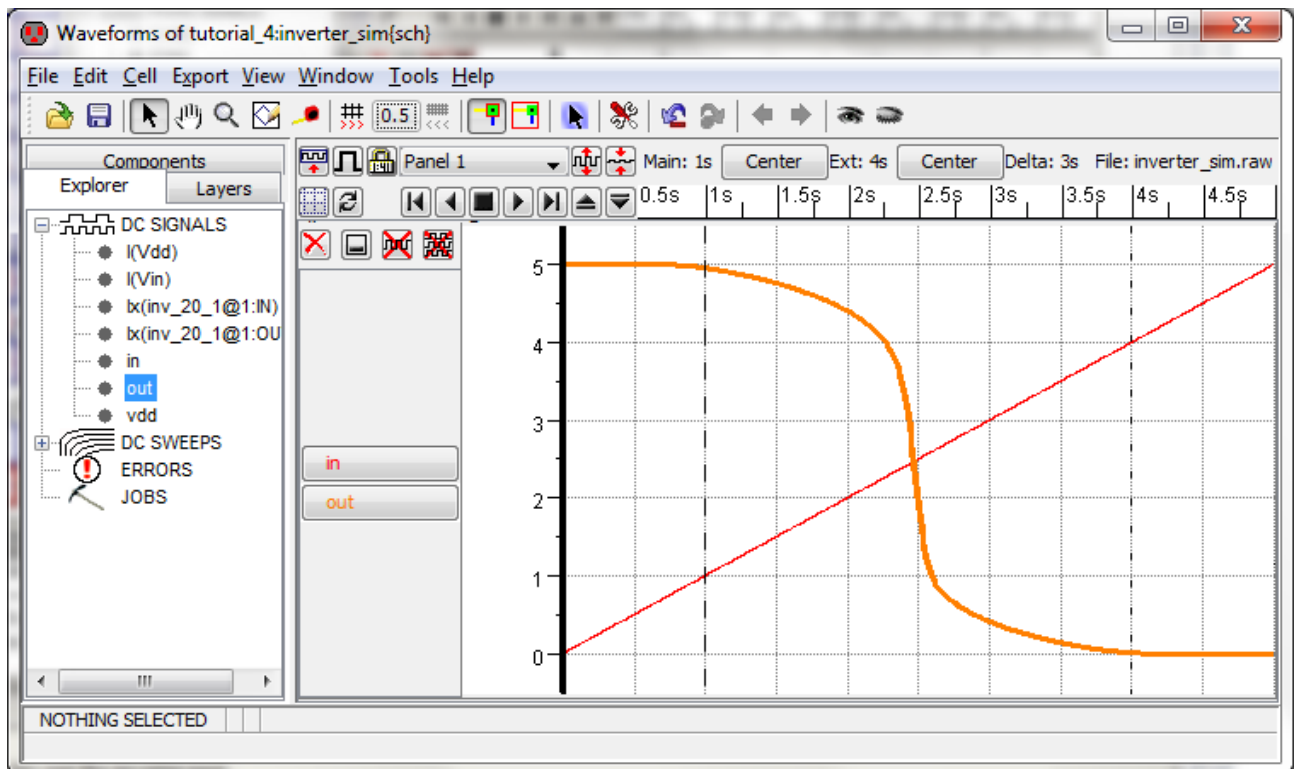


Рис. 3.18. Передаткова характеристика КМОН інвертора в сканері *Electric*

Після того, як ми переконалися в коректності роботи інвертора, приступимо до проектування топології.

Створіть нову комірку (*Ctrl+N*), назвіть її *inv\_20\_10*, режим проектування – *layout*. Додайте необхідні вузли в нову комірку (рис. 3.19).

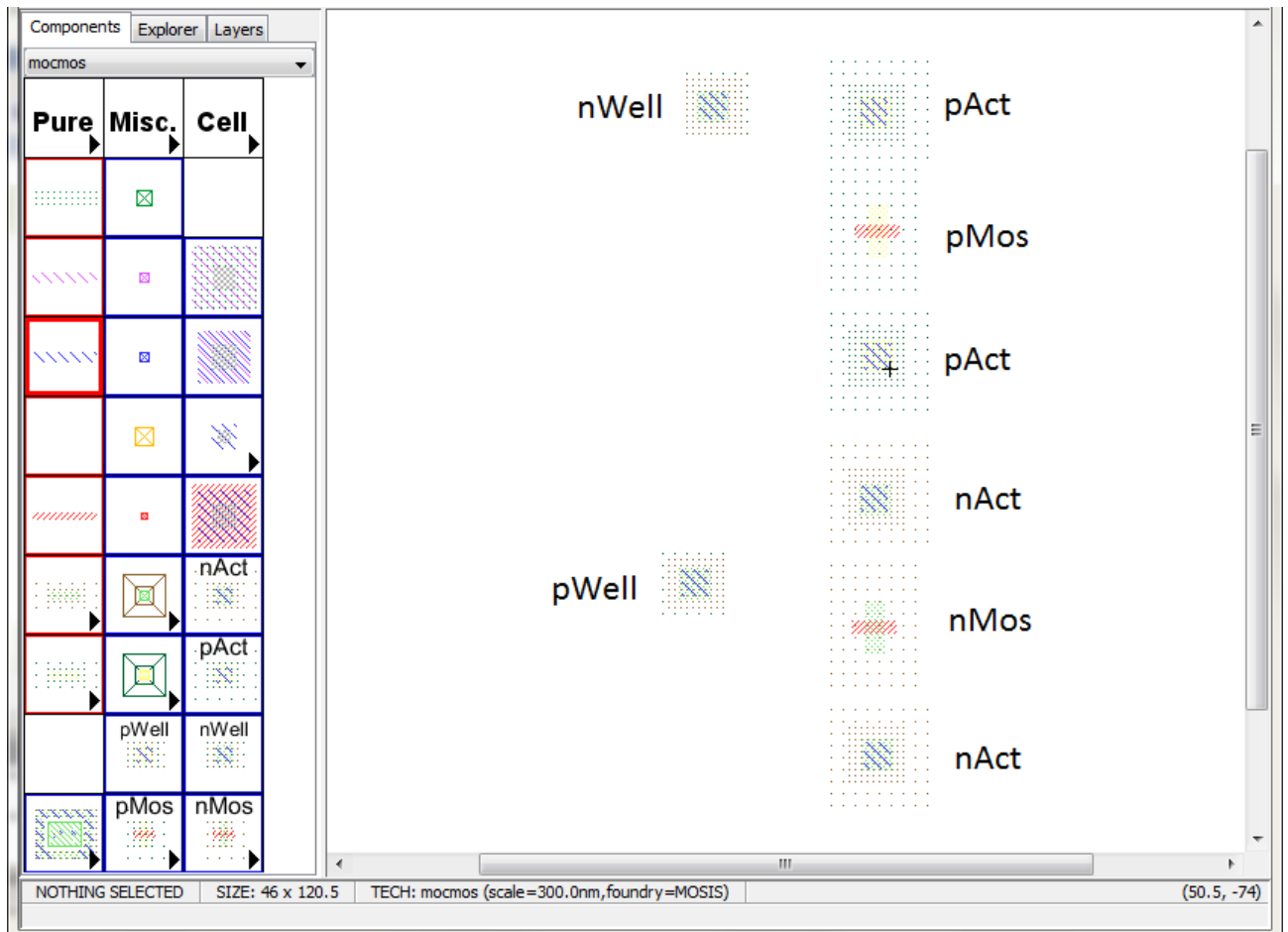


Рис. 3.19. Вузли для проектування топології КМОН інвертора

Для транзисторів встановіть назву *Spice* моделей для *NMOS* і *PMOS* транзистора (як у практичному завданні №2).

Змініть ширину й довжину вузлів. Для *PMOS* транзистора *Width* = 20, *Length* = 2, для *NMOS* – *Width* = 10, *Length* = 2 (це стосується й ширини вузлів, які формують стік і витік транзисторів). Вузли *nWell* і *pWell* повинні мати ширину 20. Довжина вузла *nWell* дорівнює 6, а *pWell* – 5 (рис. 3.20). Додайте дуги між активними областями транзисторів (рис. 3.21).

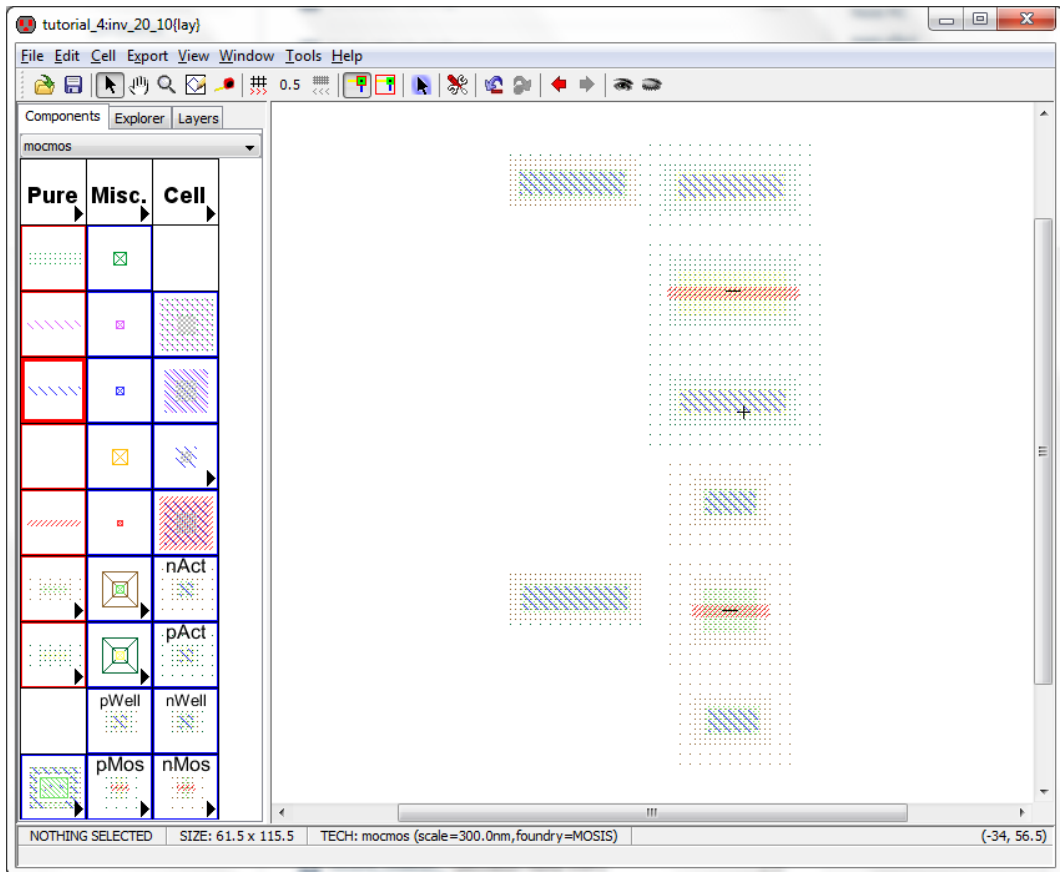


Рис. 3.20. Вузли для проєктування топології КМОН інвертора

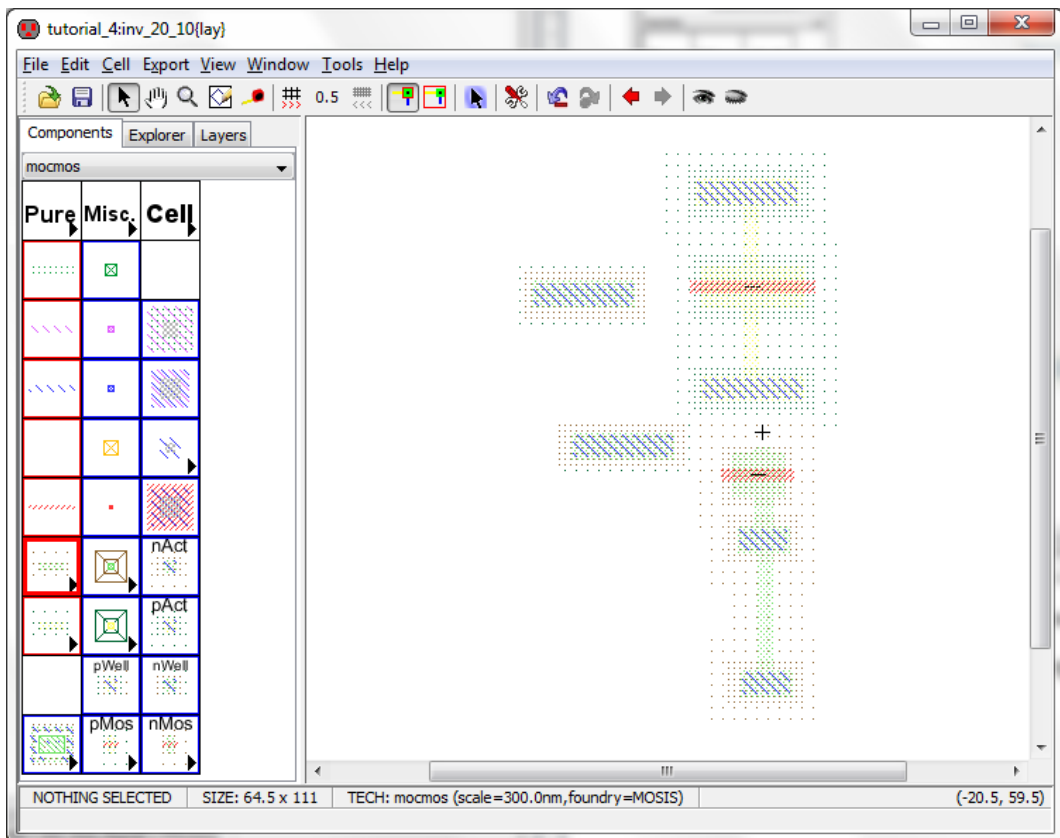


Рис. 3.21. Додавання дуг між активними областями транзисторів

Перемістіть активні області транзисторів до заслону (рис. 3.22).

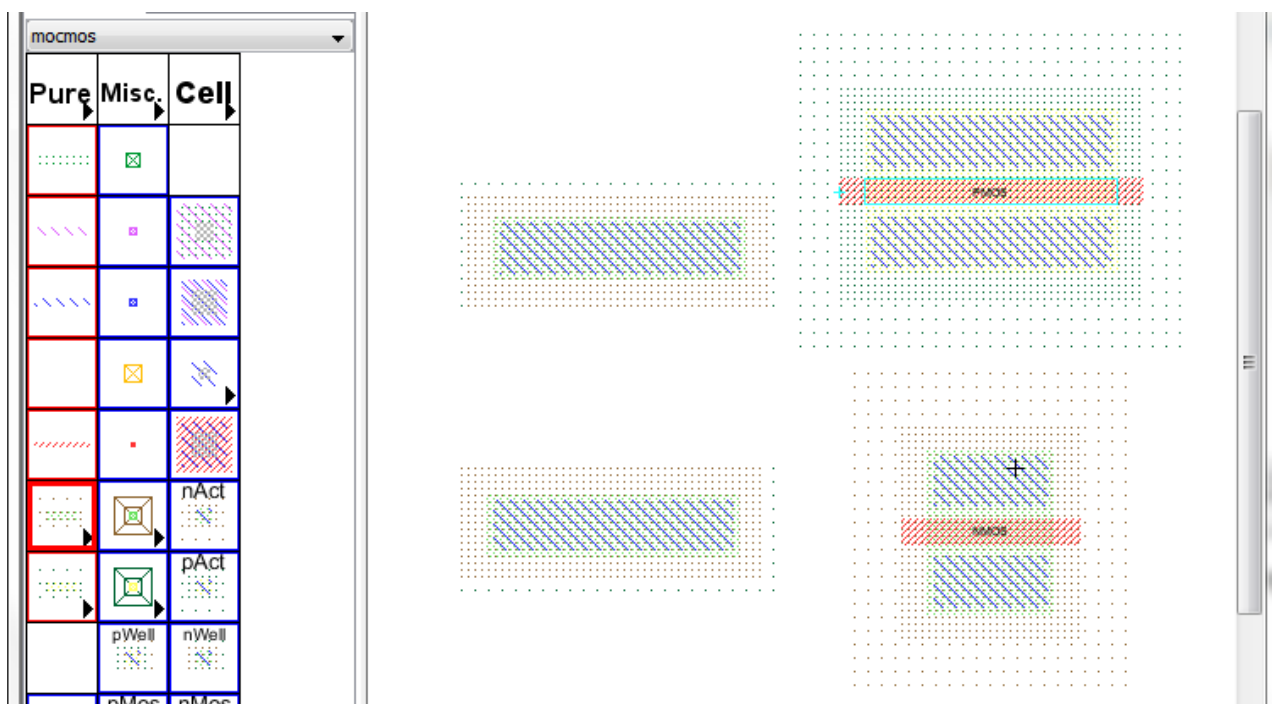


Рис. 3.22. Зближення активних областей транзисторів і заслону

Поверніть (*Ctrl+J*) транзистори й сумістіть вузли як показано на рис. 3.23. Перевірте на помилки *DRC*.

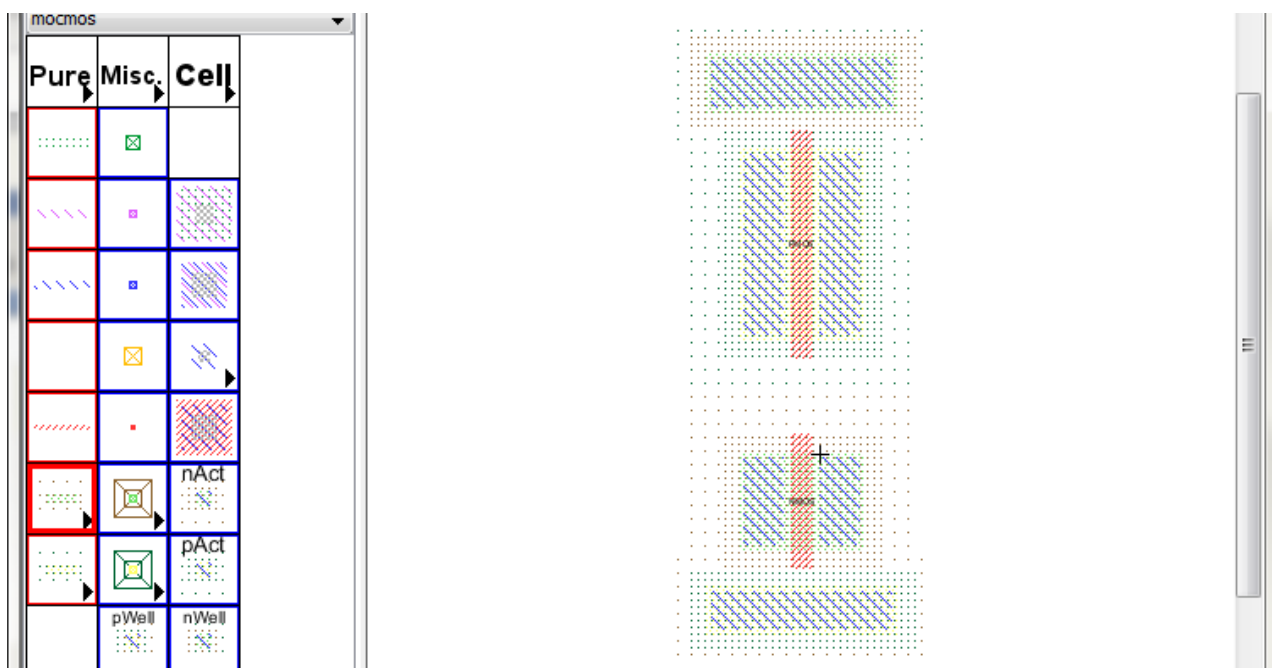


Рис. 3.23. Розміщення вузлів для побудови топології КМОП інвертора



З'єднайте між собою заслони транзисторів. Витік *PMOS* транзистора підключіть до *nWell* вузла, стік – до стоку *NMOS* транзистора. Витік *NMOS* транзистора підключіть до *pWell* вузла (рис. 3.24).

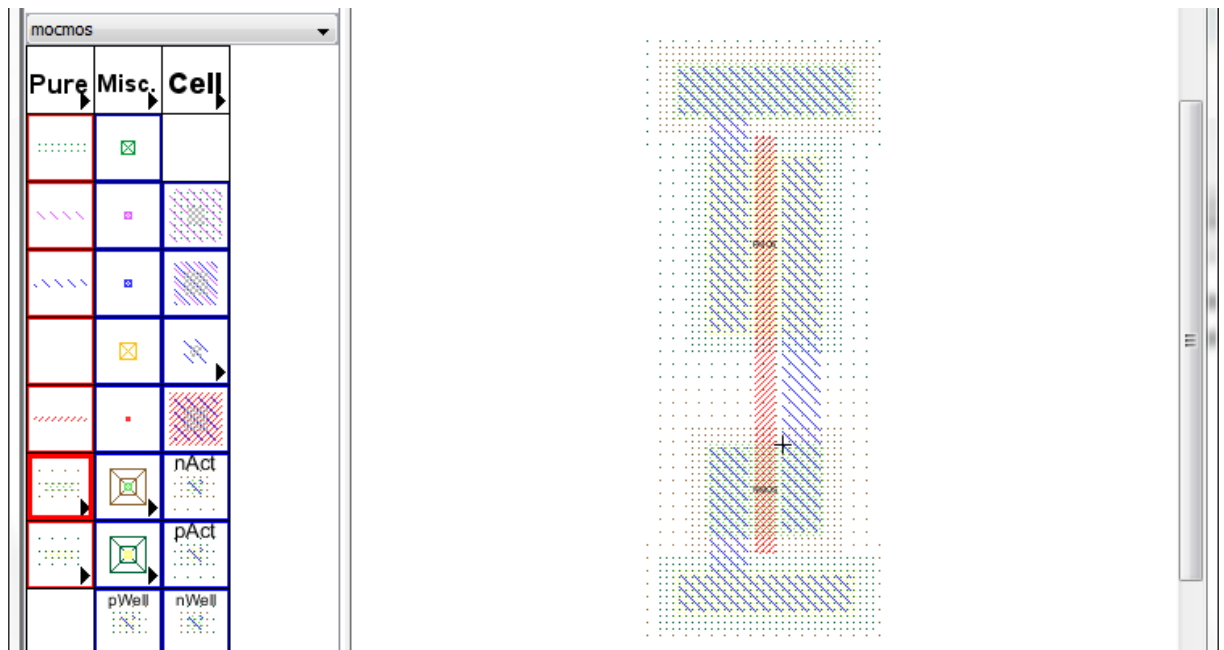


Рис. 3.24. До побудови топології КМОН інвертора

Тепер додайте ліворуч контакт полікремній-метал (*Metal-1-Polysilicon-I-Con*), а праворуч – контакт метал (*Metal-1-Pin*) (рис. 3.25).

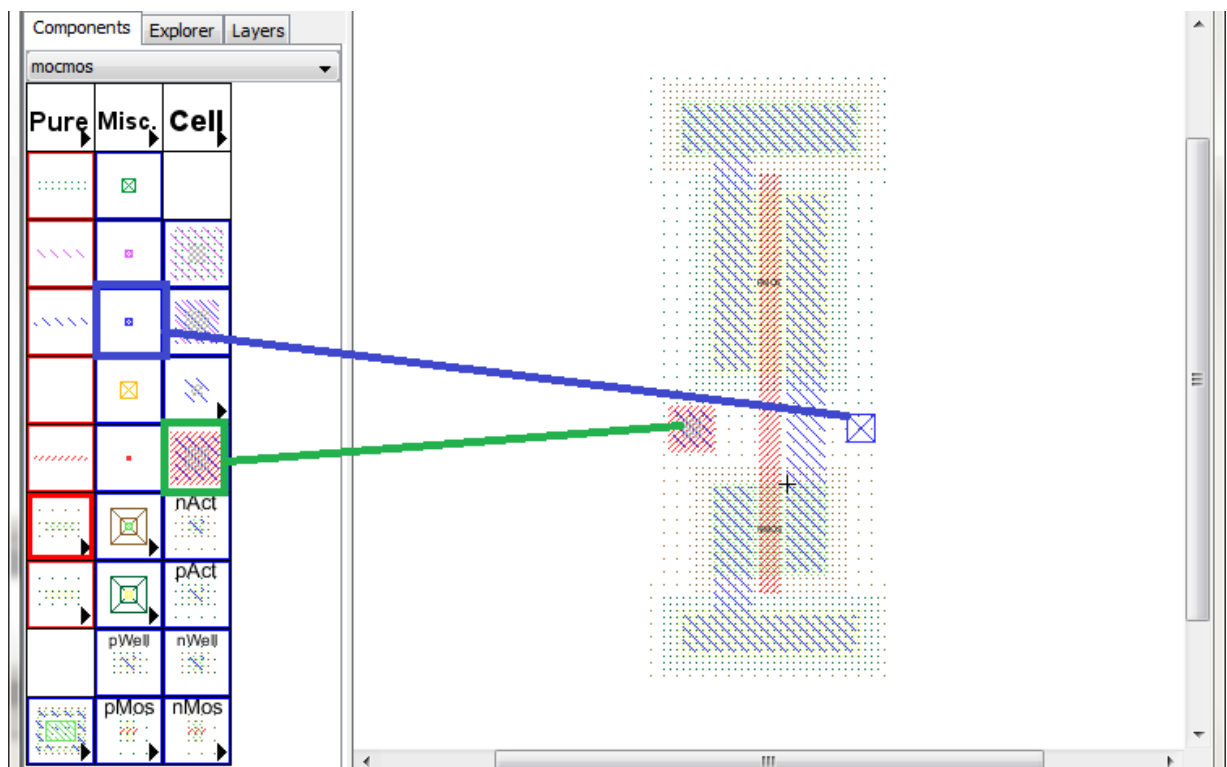


Рис. 3.25. До побудови топології КМОН інвертора

Приєднайте вставлені контакти до полікремнієвої дуги між заслонами й до металевої дуги між стоком і витокком (рис. 3.26). Ширина полікремнієвої дуги має дорівнювати 2.

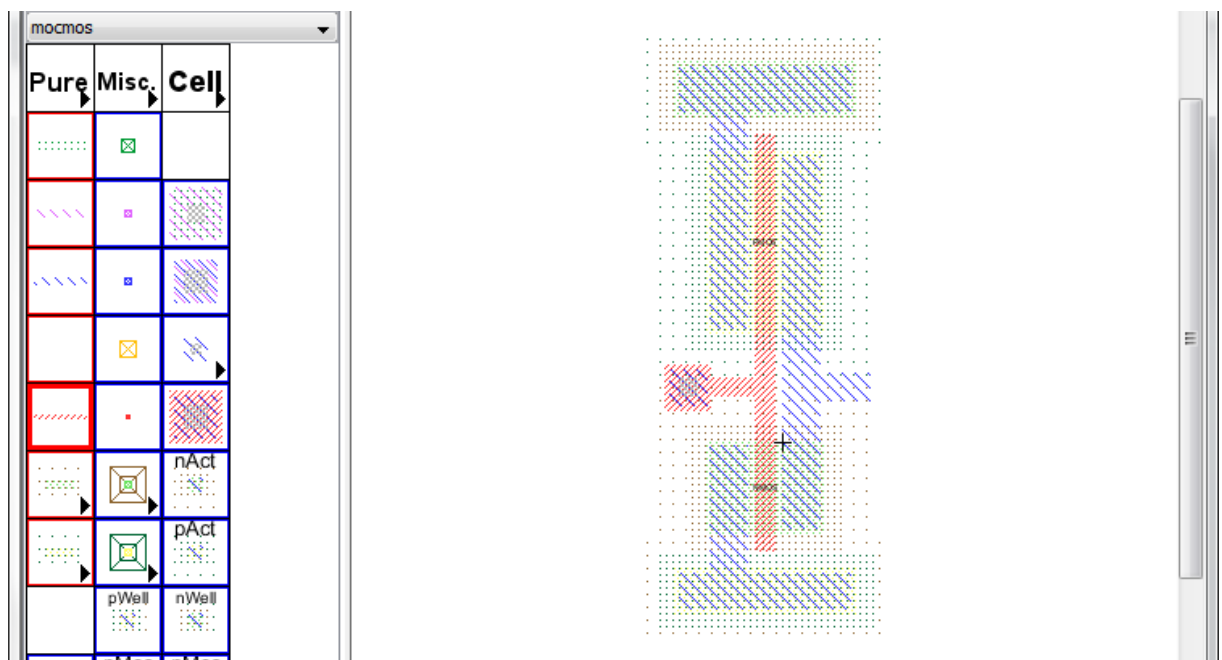


Рис. 3.26. До побудови топології КМОН інвертора

Експортуйте (*Ctrl+E*) *pWell* на «землю», *nWell* на живлення, *Metal-1-Polysilicon-1-Con* на вхід, *Metal-1-Pin* на вихід. Розмір тексту – 5 (рис. 3.27, 3.28).

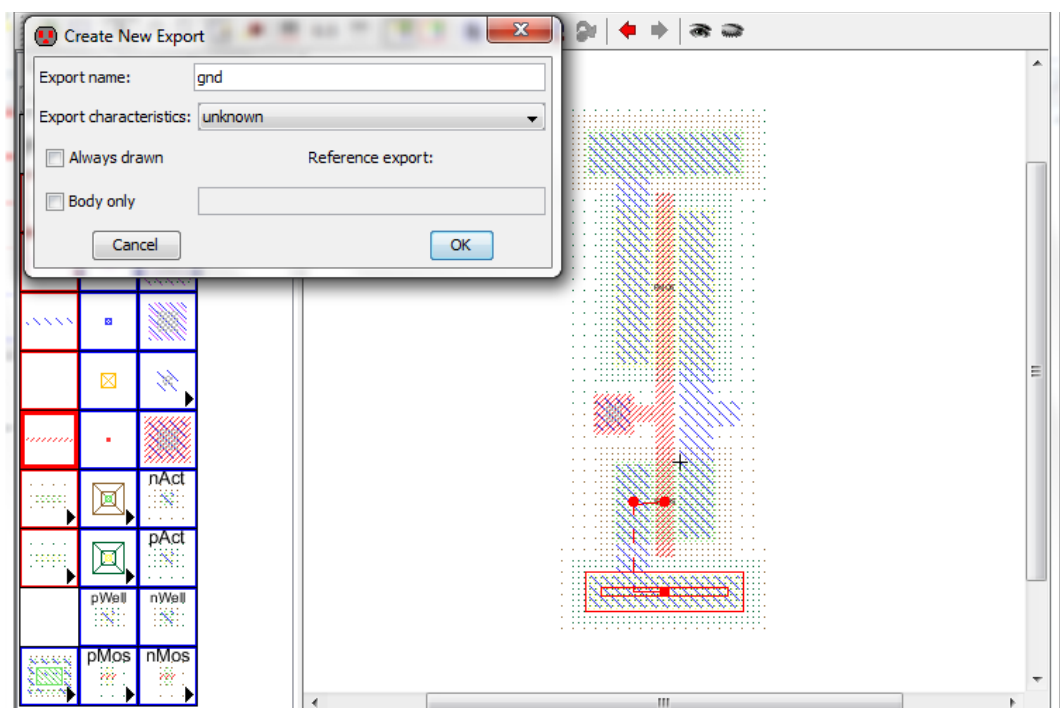


Рис. 3.27. Експорт контактів топології КМОН інвертора

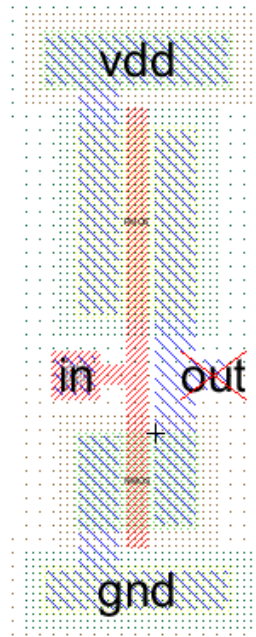
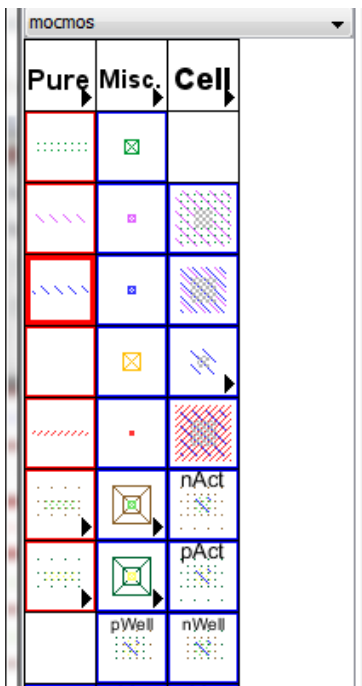


Рис. 3.28. Топологія КМОН інвертора

Промодельюємо роботу пристрою. Створіть нову комірку з іменем *inverter\_sim* у режимі проєктування *layout*. Вставте в неї комірку *inv\_20\_10{lay}* з інвертором. В області проєктування з'явиться клітка, яка позначає інвертор (рис. 3.29).

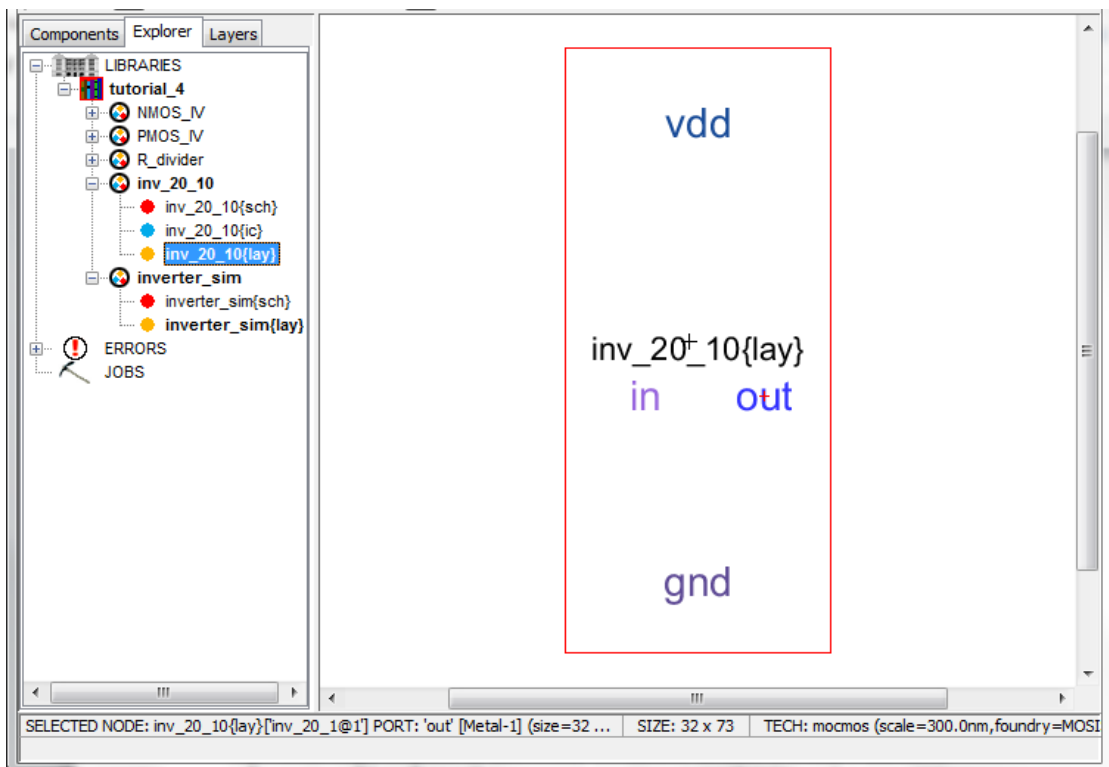


Рис. 3.29. Клітка-інвертор

Є два стилі перегляду комірки: символічний і з відображенням топології. Щоб переключатися між ними, використовуйте кнопки відкритого й закритого ока в панелі інструментів (рис. 3.30).



Рис. 3.30. Перемикач стилів перегляду комірки

Відредагуйте властивості клітки з інвертором (*Ctrl+I*). За необхідності, відключіть функцію *Easy to Select*, це зручно, якщо макет великий (рис. 3.31).

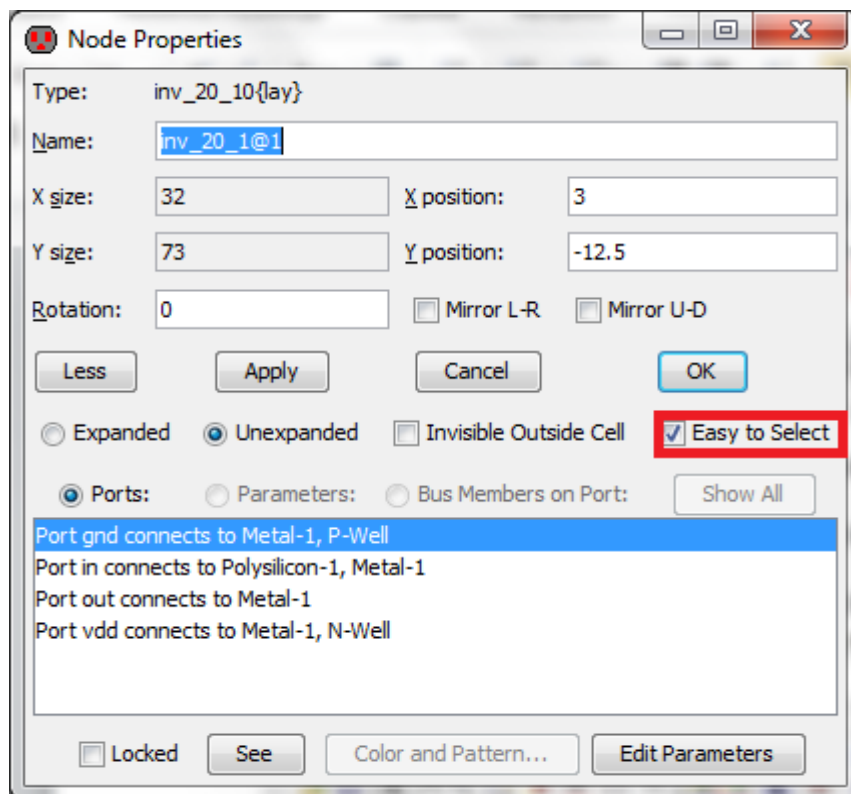


Рис. 3.31. Вікно властивостей вузла

При використанні *Pure* шарів із вкладки *Components* використовуйте спеціальне виділення (рис. 3.32).



Рис. 3.32. Спеціальне виділення при використанні *Pure* шарів

Виведіть металеві дуги від інвертора, як показано на рис. 3.33.

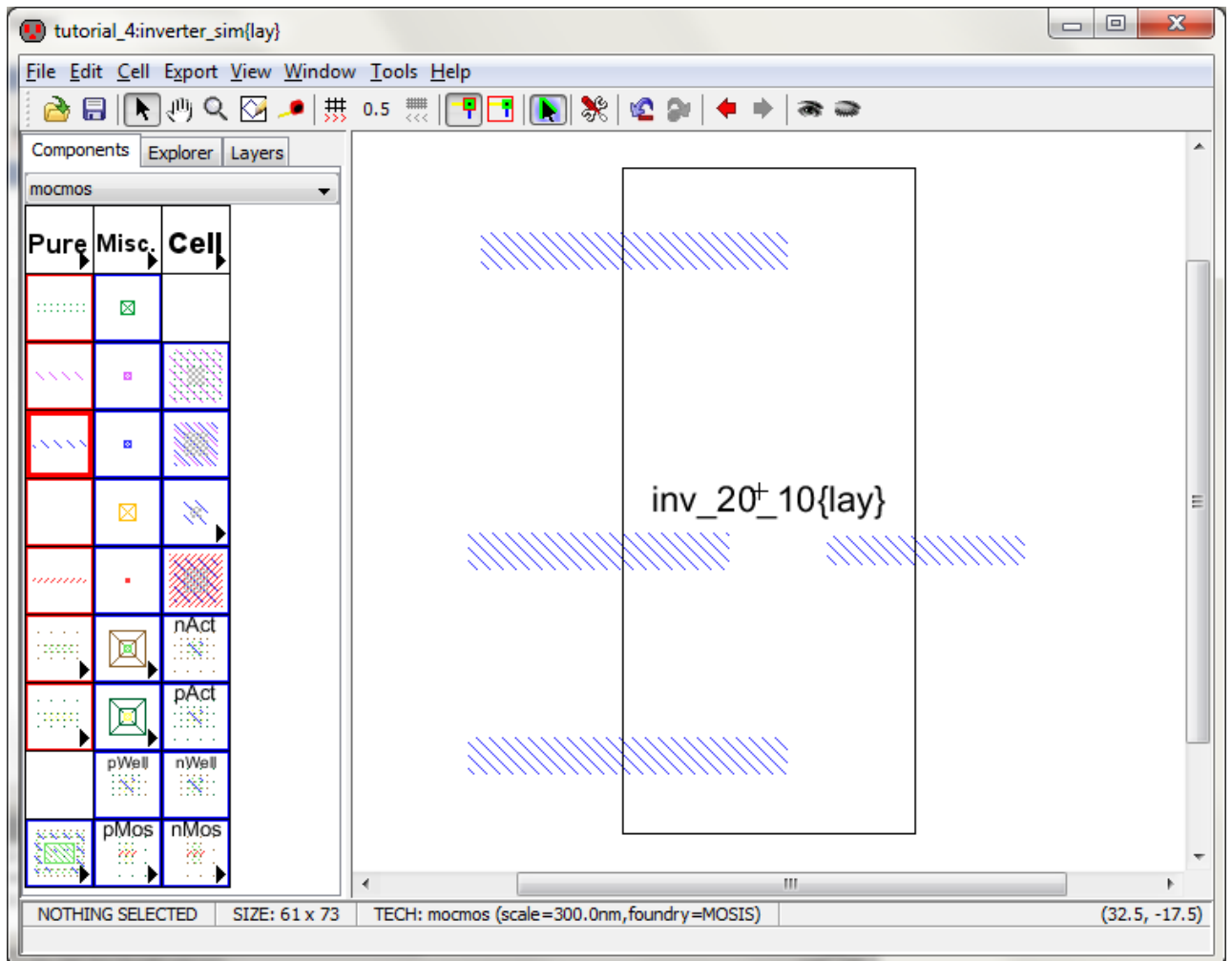


Рис. 3.33. Металеві дуги від інвертора

Скопіюйте з комірки *inverter\_sim{sch}* у комірку *inverter\_sim{lay}* Spice код.

```
vdd vdd 0 DC 5
vin in 0 DC 0
.dc vin 0 5 1m
.include C:\Electric\C5_models.txt
```

Промаркуйте дуги у відповідності зі схемою у комірці *inverter\_sim{lay}* (рис. 3.34). Розмір тексту при цьому вкажіть 5.

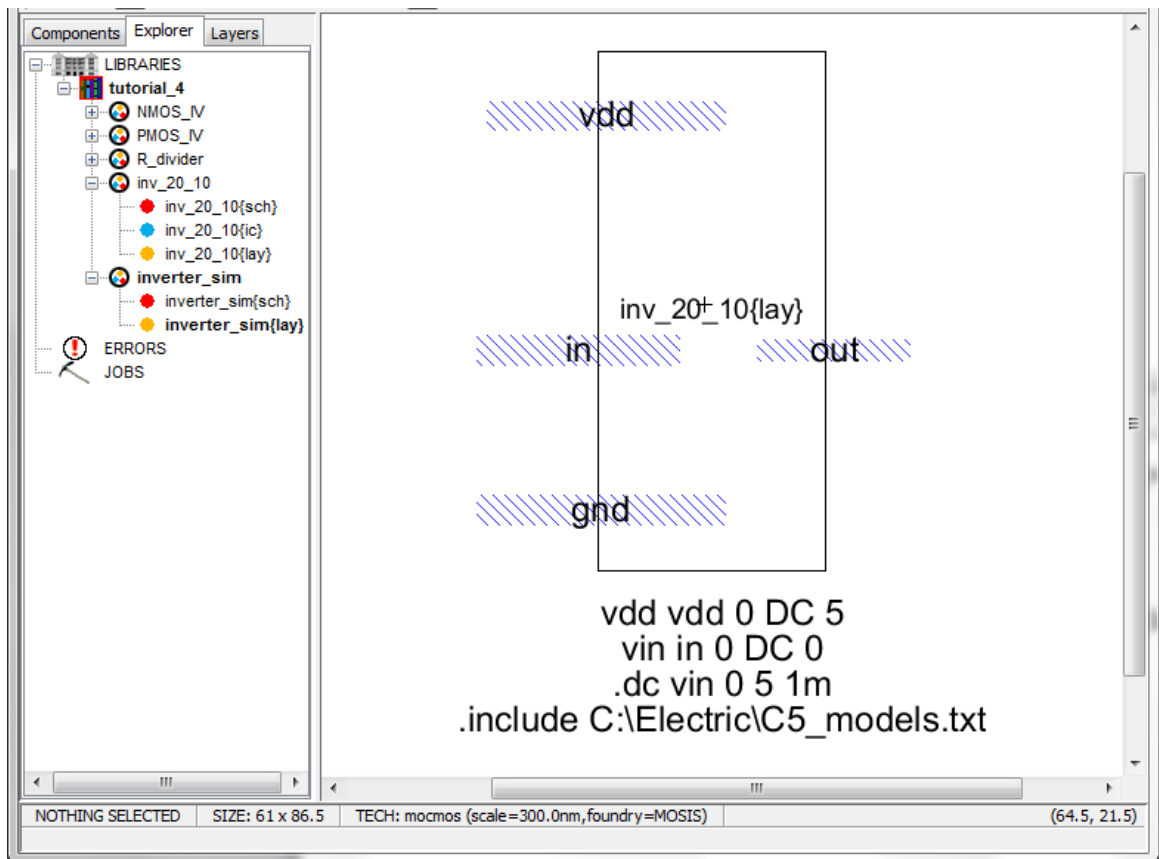


Рис. 3.34. Інвертор із промаркованими дугами

Перевірте макет на помилки *DRC* (F5), *NCC* (L), *ERC*.

*DRC* помилок не має бути.

*NCC* помилки виникли внаслідок невідповідності між комірками (у комірці *inverter\_sim{lay}* не позначені «земля» й живлення) (рис. 3.35).

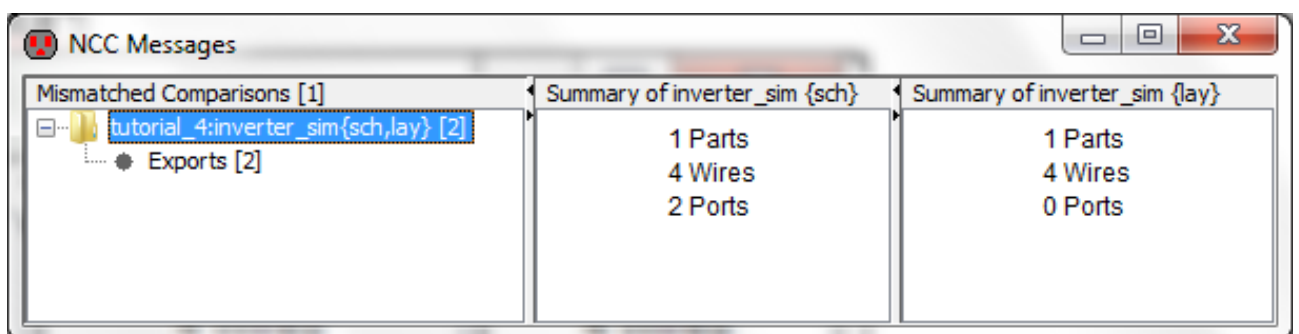


Рис. 3.35. Повідомлення про *NCC* помилки

Перейшовши в списку стовпця *Mismatched Comparisons* [1] у пункт *Export* [2], побачите причину помилок більш детально. Зеленим кольором

виділена помилка, яка мається на увазі (рис. 3.36). Якщо клацнути по напису, то *Electric* укаже місце помилки в робочій області.

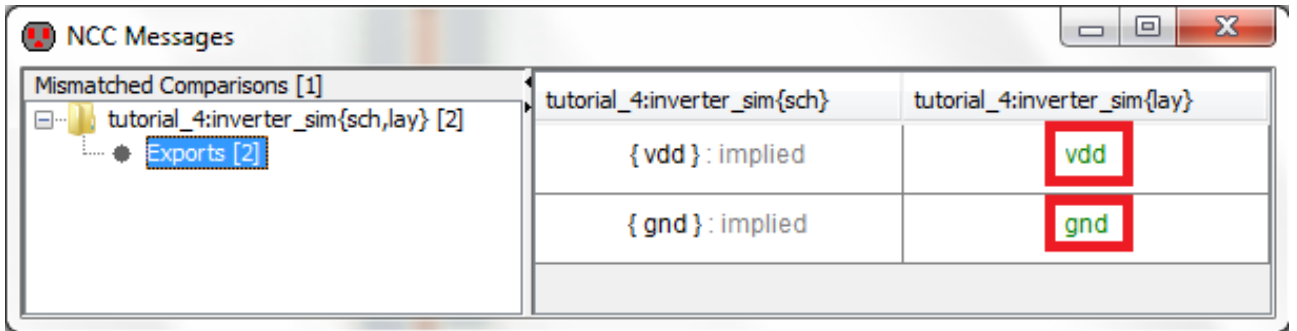


Рис. 3.36. Повідомлення про причини NCC помилок

Помилка *ERC* викликана тією ж причиною – контакт від *pWell* комірки не приєднаний до «землі».

Для того, щоб виправити помилки, потрібно експортувати контакти на «землю» (*gnd*) і на живлення (*vdd*). Щоб не виникала тавтологія, видаліть маркування дуг *vdd* і *gnd*, і експортуйте контакти цих дуг. Результат таких дій відображений на рис. 3.37. Перевірте на помилки; тепер їх не має бути.

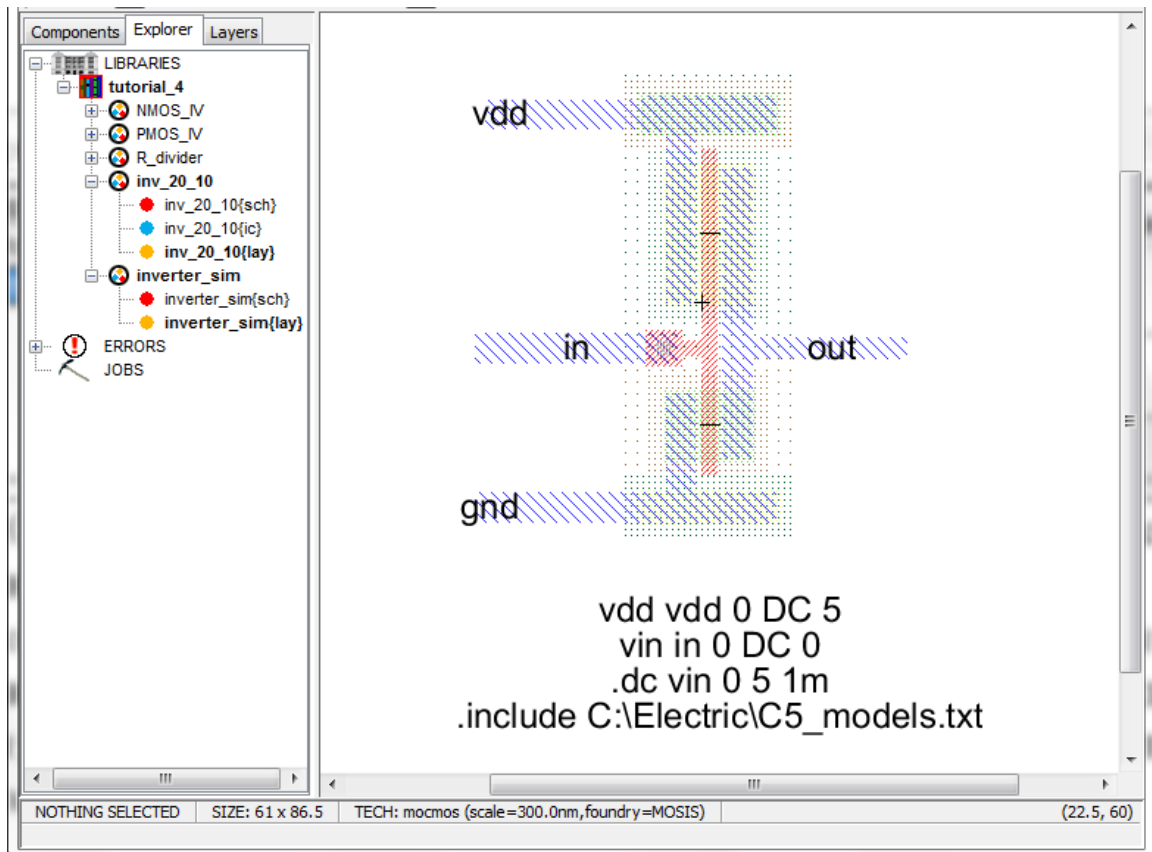
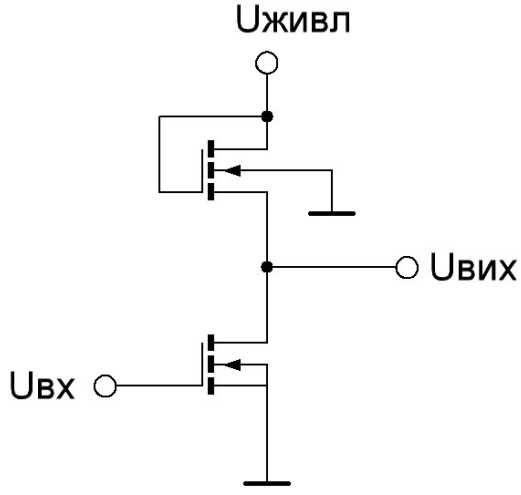
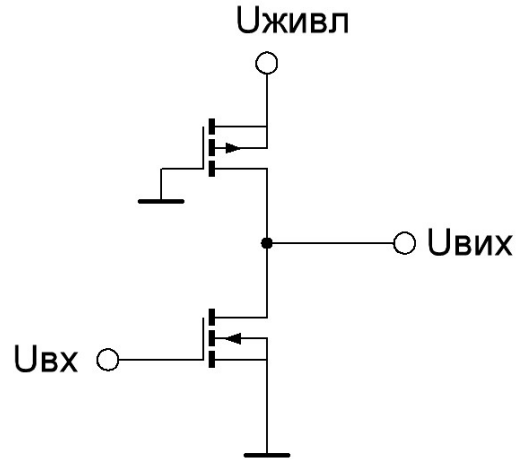


Рис. 3.37. Топологія КМОП інвертора

### Варіанти завдань

№ варіанта	Назва роботи	Схема
1, 3, 5	Моделювання роботи <i>n</i> -МОН інвертора	
2, 4, 6	Моделювання роботи псевдо- <i>n</i> -МОН інвертора	



## Контрольні питання

1. Які транзистори керуються вхідною напругою в КМОН-інверторі?
2. Опишіть роботу КМОН-інвертора (стан транзисторів за різних рівнів напруги).
3. Чому КМОН-інвертори характеризуються мінімальним споживанням енергії?
4. Як можна зменшити потужність, яка розсіюється інвертором?
5. Чим обумовлена підвищена завадостійкість КМОН-схем?
6. Чому КМОН-інвертори характеризуються низькою швидкодією? Як можна збільшити їх швидкодію?
7. З чим пов'язана зміна передавальної характеристики КМОН-елемента з багатьма входами при зміні кількості входів, на яких сигнал змінюється одночасно?
8. Опишіть причину заціпання КМОН-мікросхем. Як можна уникнути появи цього ефекту?
9. Назвіть умови виконання співвідношення  $U_0 < U_{\text{пор}}$  в  $n$ -МОН інверторі (співвідношення геометричних розмірів каналів транзисторів).
10. Чому в  $n$ -МОН інверторі максимальна напруга логічної одиниці завжди менше напруги живлення?
11. Опишіть роботу  $n$ -МОН інвертора (стан транзисторів за різних рівнів напруги).
12. Чим визначається швидкодія інвертора?
13. Чим визначається потужність, споживана  $n$ -МОН інвертором?
14. Чому в КМОН-інверторах  $p$ -канальний транзистор роблять ширше, ніж  $n$ -канальний?
15. Що називають псевдо- $n$ -МОН-елементом?
16. Назвіть основну перевагу псевдо- $n$ -МОН-елемента порівняно з КМОН.
17. Чому для псевдо- $n$ -МОН-елемента  $U_1 = U_{\text{живл}}$ ?
18. Чому для псевдо- $n$ -МОН-елемента  $U_0 \neq 0$ ?

19. Які схеми називають стандартизованими, а які – нестандартизованими (по залежності рівнів сигналу від розмірів транзисторів)?
20. Як можна максимально зменшити  $U_0$  в псевдо- $n$ -МОН-елементі?
21. Назвіть основний недолік псевдо- $n$ -МОН-елемента.

## Практичне заняття №4

### Проектування КМОН схеми І-НІ

Проектувати КМОН схему І-НІ будемо на основі технологічного процесу *C5*, і налаштувань, установлених у попередніх практичних роботах.

Збережіть бібліотеку *tutorial\_3.jelib* як *tutorial\_4.jelib*, також простежте, щоб файл *C5\_models.txt* знаходився у директорії *C:\Electric\*.

Запустіть *Electric* і відкрийте *tutorial\_4.jelib*.

Створіть дублікат групи комірок *inv\_20\_10* і назвіть його *NAND\_2* (де цифра «2» означає кількість виходів): вкладка Explorer → LIBRARIES → tutorial\_4 → inv\_20\_10 → контекстне меню (правою кнопкою мишки) → Duplicate Cell in Group (рис. 4.1, 4.2).

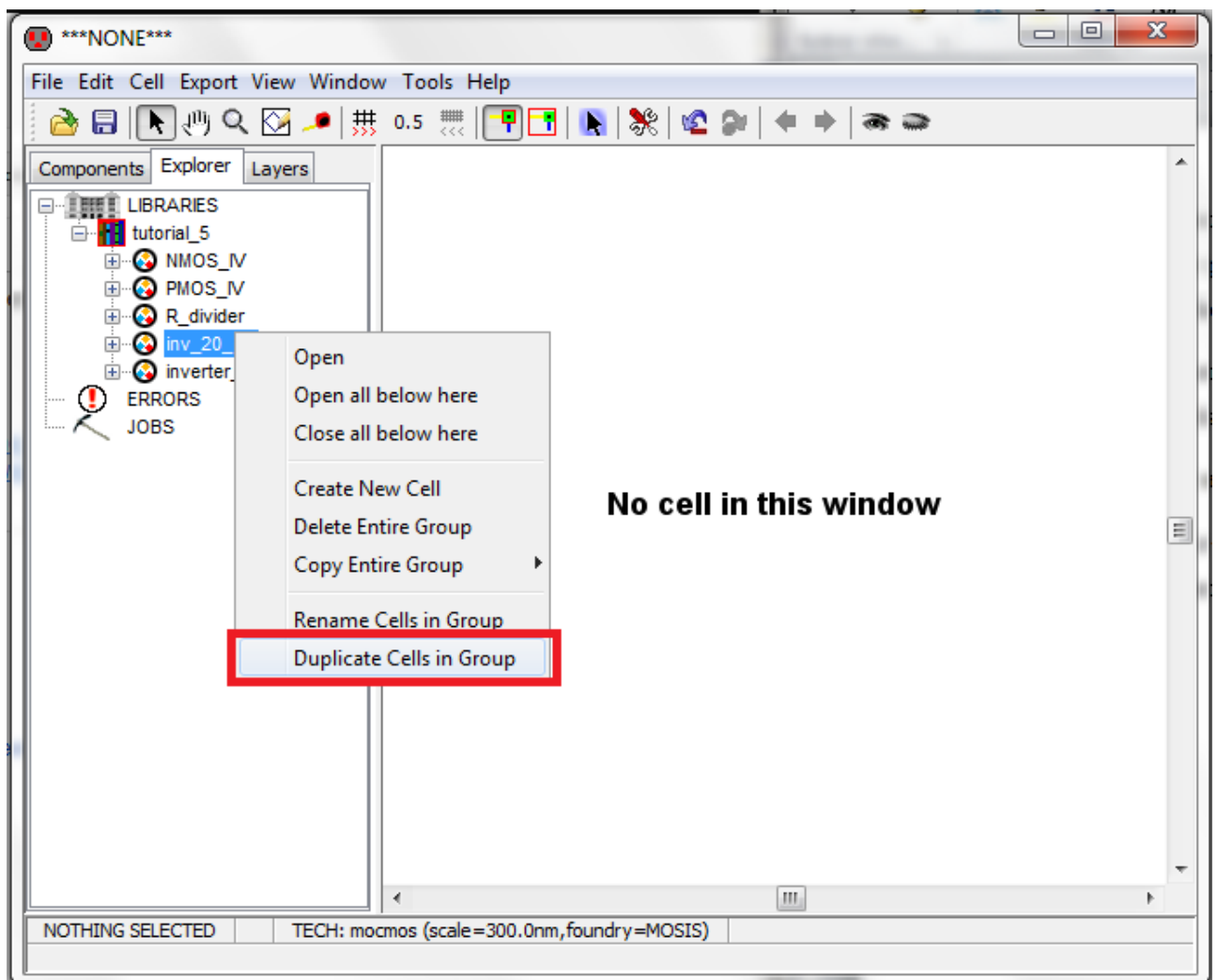


Рис. 4.1. Створення дублікату групи комірок

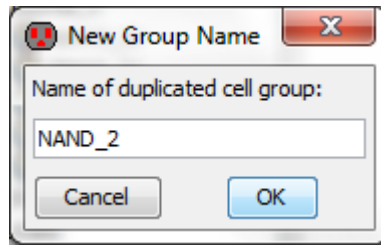


Рис. 4.2. Призначення імені групі комірок

Відкрийте комірку  $NAND\_2\{sch\}$  і видаліть дуги вхід/вихід і піктограму. Змініть ширину  $PMOS$  транзистора з 20 на 10. Приведіть комірку до вигляду, зображеного на рис. 4.3 (або створіть нову комірку й нарисуйте схему заново).

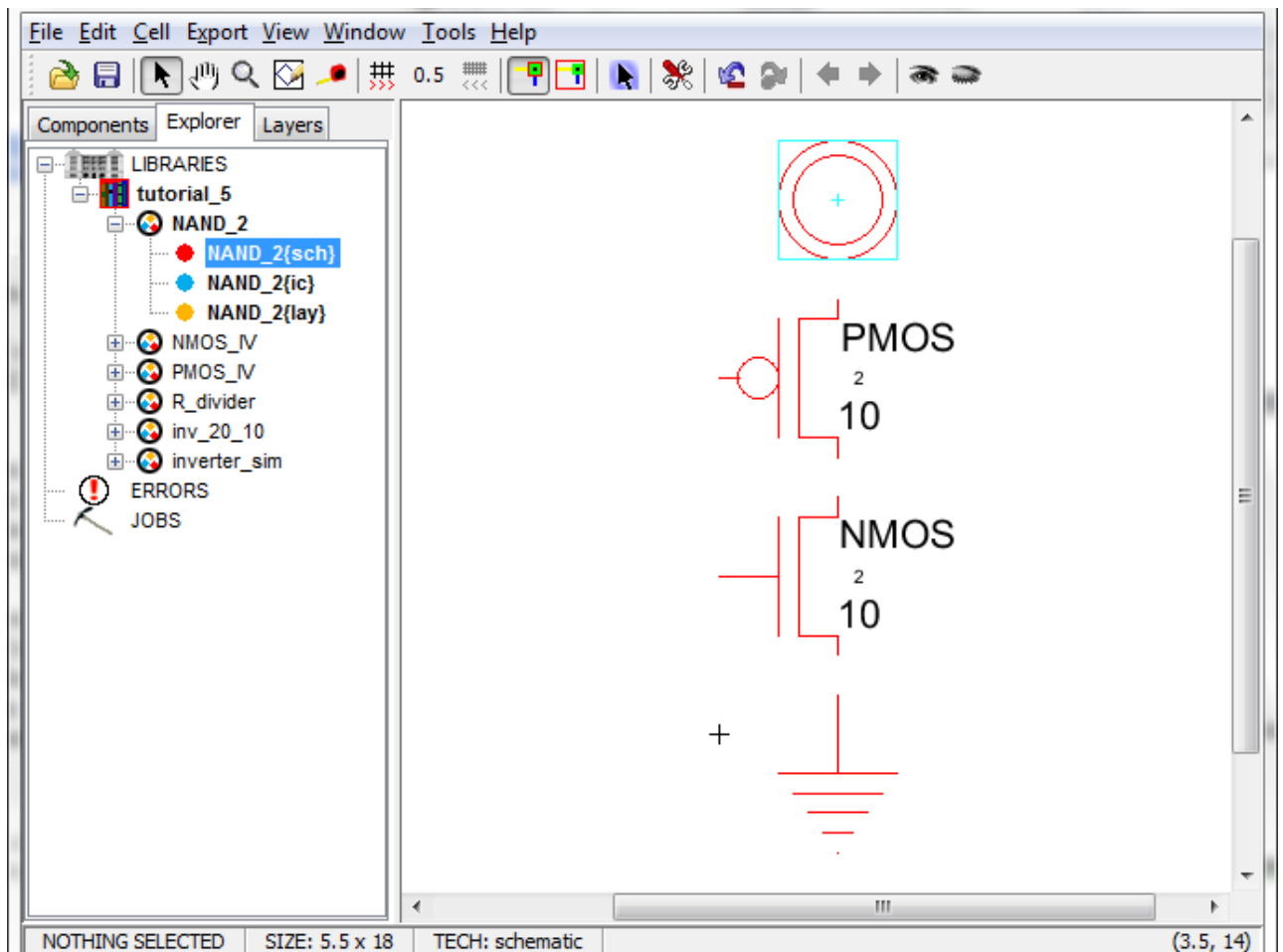


Рис. 4.3. До побудови КМОН схеми І-НІ

Тепер, копіюючи вузли-транзистори нарисуйте схему І-НІ та експоруйте контакти  $A$ ,  $B$ ,  $AnandB$ . Результат наведено на рис. 4.4.

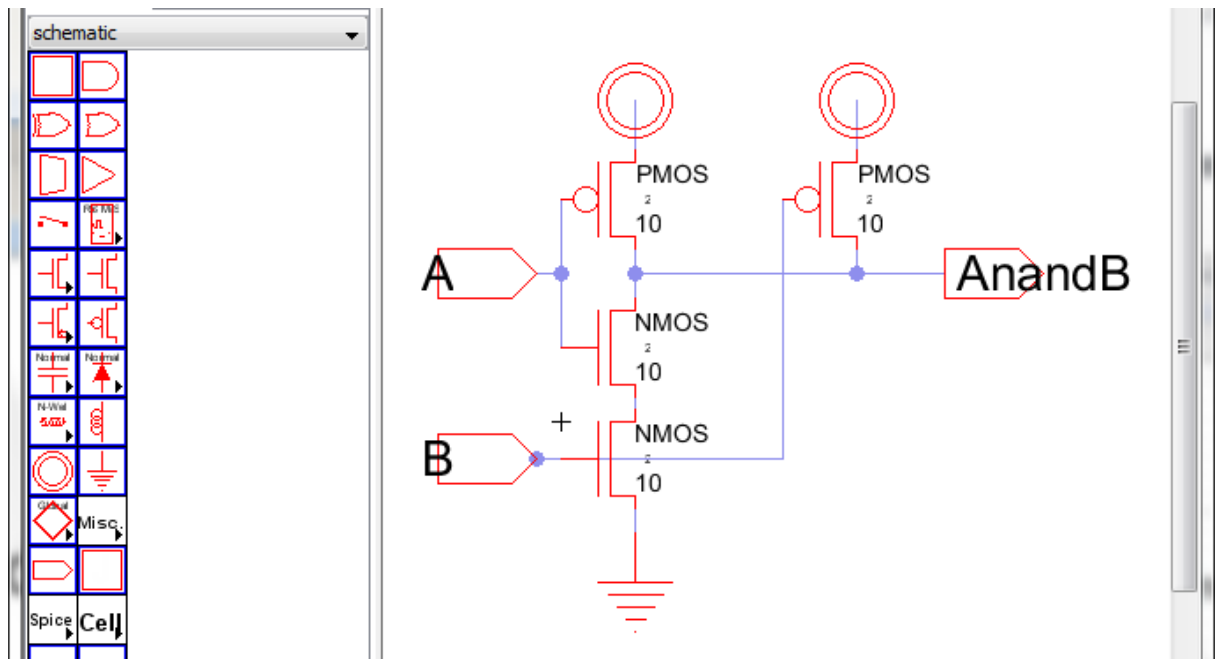


Рис. 4.4. КМОП схема І-НІ

В *Explorer* видаліть комірку *NAND\_2{ic}* (комірку піктограми схеми І-НІ). Потім створіть нову комірку піктограми для схеми *View* → *Make Icon View* (рис. 4.5).

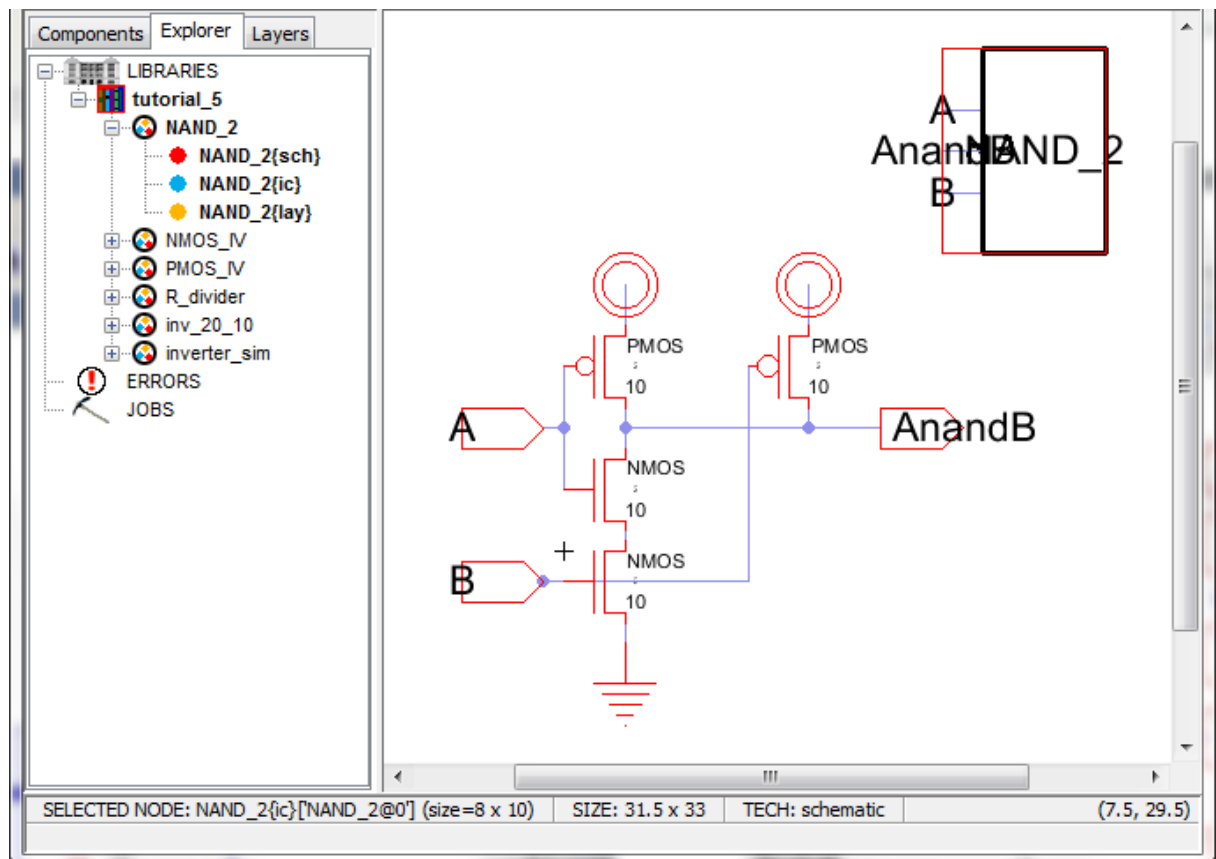


Рис. 4.5. Створення піктограми елемента І-НІ

Відкрийте комірку  $NAND\_2\{ic\}$  для редагування вигляду піктограми.  
Приведіть піктограму у відповідність із рис. 4.6.

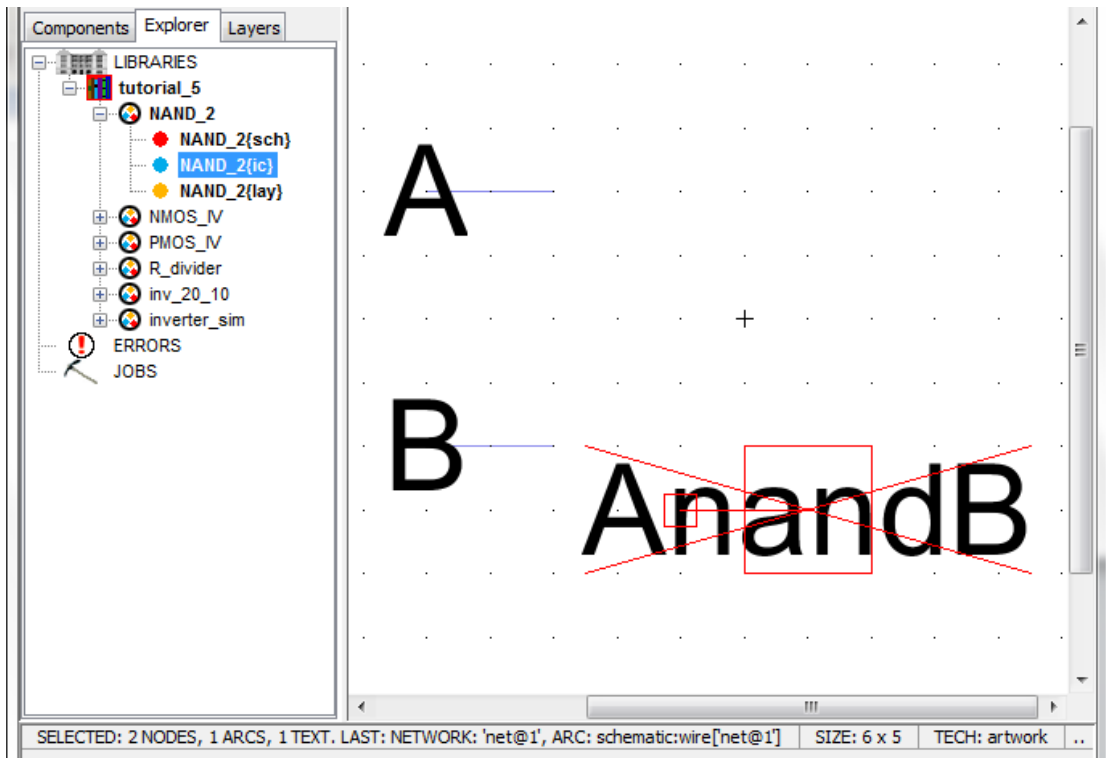


Рис. 4.6. До створення піктограми елемента І-НІ

Тепер треба намалювати дугу в 180 градусів. Для цього вставте із вкладки коло й у його властивостях укажіть градус кола 180 (рис. 4.7).

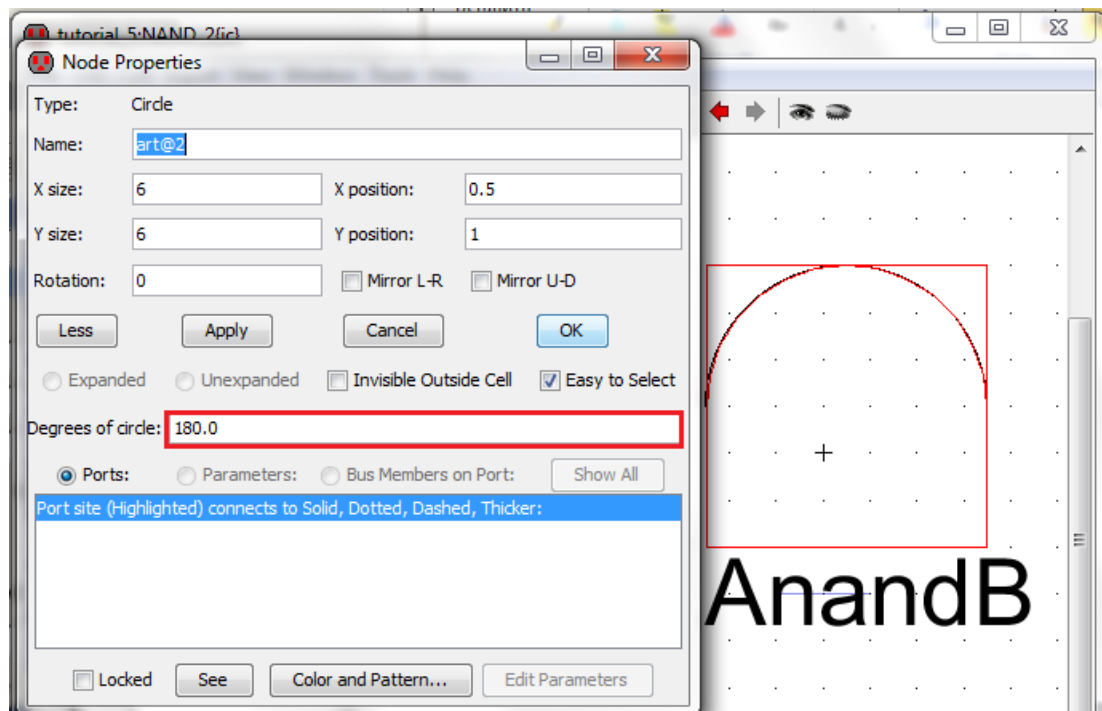


Рис. 4.7. Властивості вузла – кола

Поверніть отриману дугу на 90 градусів за годинниковою стрілкою й додайте ще одне коло радіусом 1. Перемістіть об'єкти так, як показано на рис. 4.8.

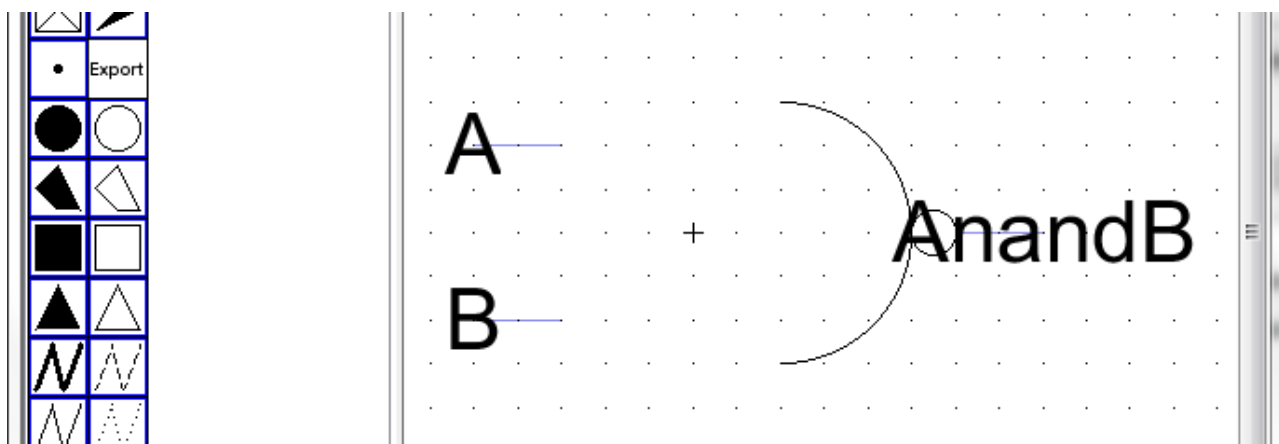


Рис. 4.8. До створення піктограми елемента І-НІ

Вставте вузол відкритий полігон (*Opened-Polygon*) (рис. 4.9).

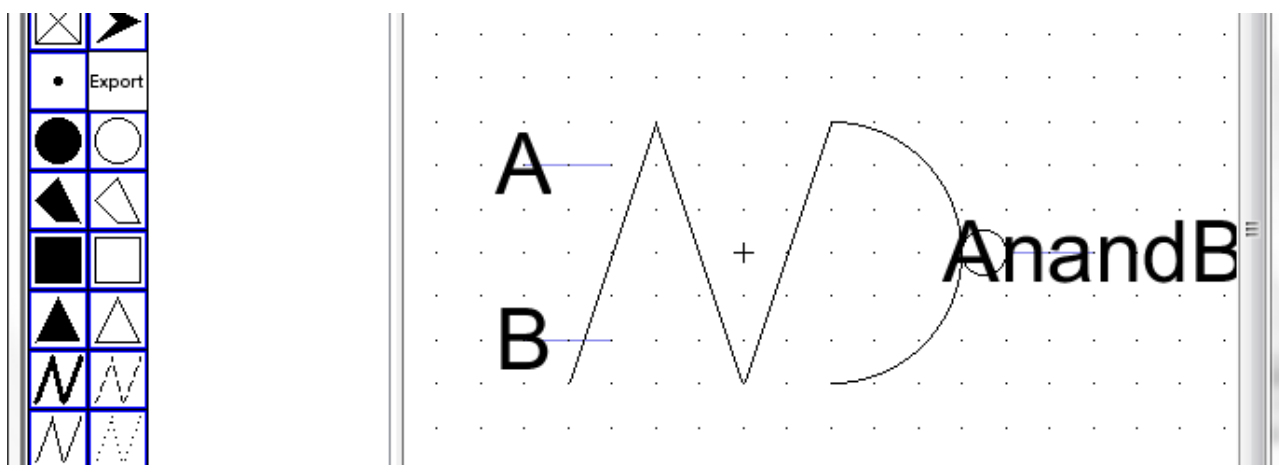


Рис. 4.9. Додавання вузла відкритий полігон

Для того, щоб відредагувати зовнішній вигляд полігона потрібно його виділити й перейти в меню *Edit* → *Modes* → *Edit* → *Toggle Outline Edit* (або просто натиснути клавішу «Y»). У цьому режимі курсор відображається у вигляді олівця.

Для додавання вершин до полігона, клацніть правою кнопкою мишки так само, як додавали дуги в попередніх практичних роботах.

Для виходу з режиму редагування використовуйте меню *Edit* → *Modes* → *Edit* → *Toggle Outline Edit* або натисніть клавішу «Y».

Приведіть піктограму до вигляду, відповідному до піктограми на рис. 4.10.

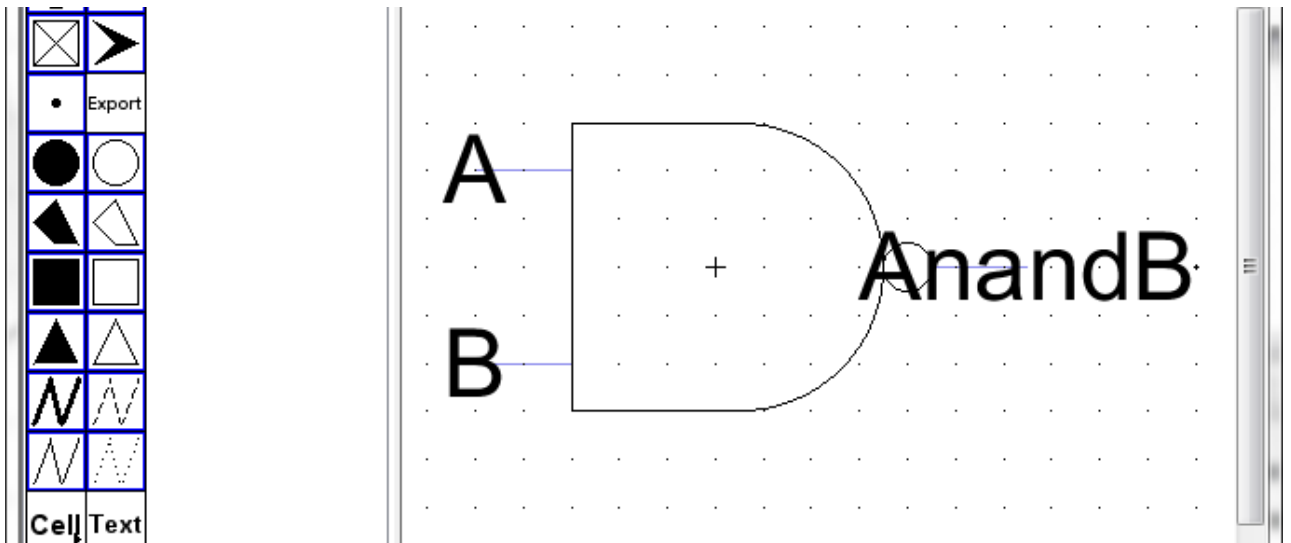


Рис. 4.10. Піктограма елемента І-НІ

Відкрийте комірку *NAND\_2{sch}*. У ній зображена схема І-НІ з піктограмою (рис. 4.11).

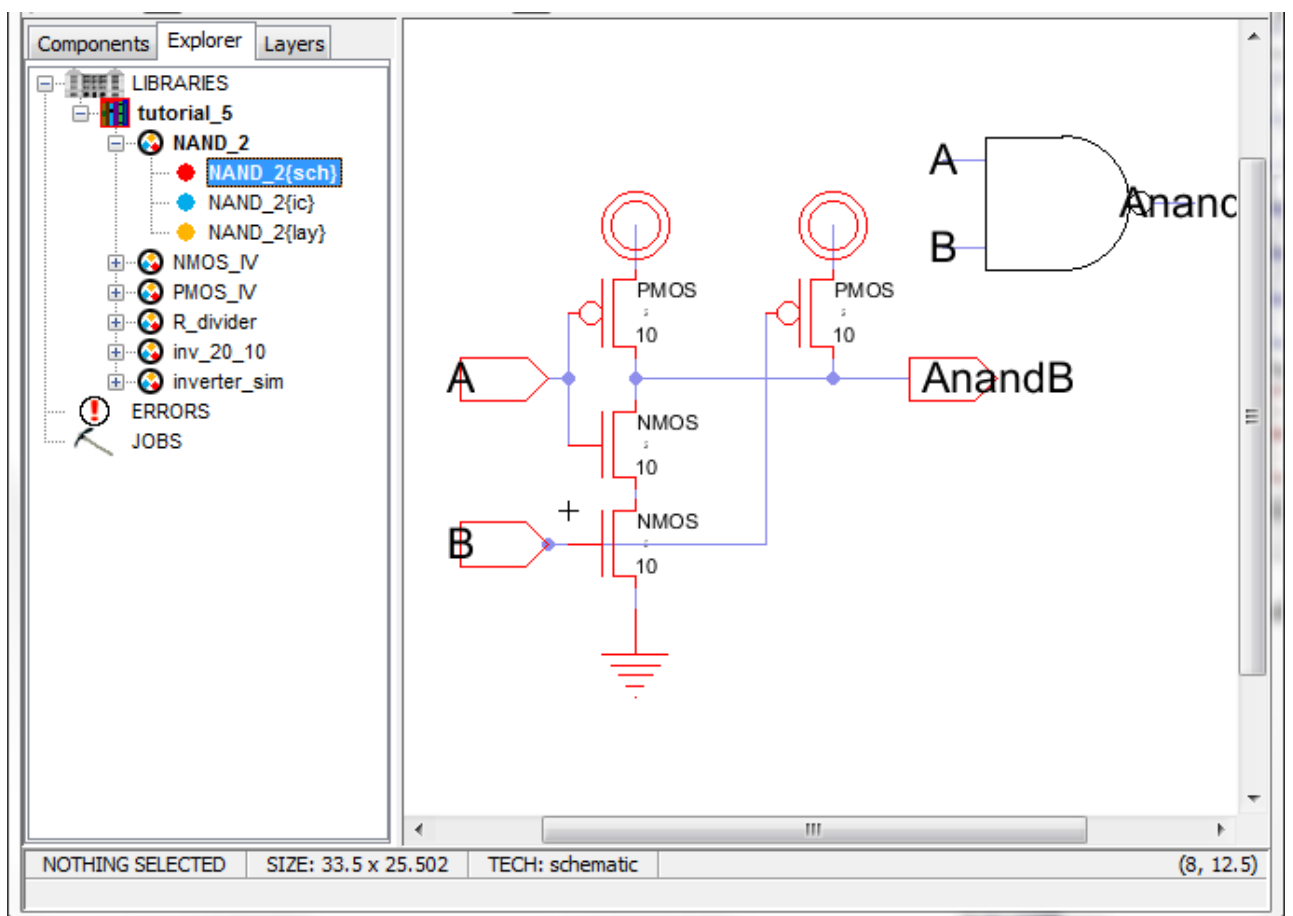


Рис. 4.11. Схема І-НІ з піктограмою



Промодельуйте роботу схеми І-НІ. Для цього створіть комірку  $NAND\_sim\{sch\}$ . Вставте в неї піктограму І-НІ схеми ( $NAND\_2\{ic\}$ ). Це можна зробити, перетягнувши комірку в робочу область або ж скористатися вкладкою  $Components \rightarrow Cell \rightarrow NAND\_2\{ic\}$ .

Підключіть дуги до виходів схеми. Результат наведено на рис. 4.12.

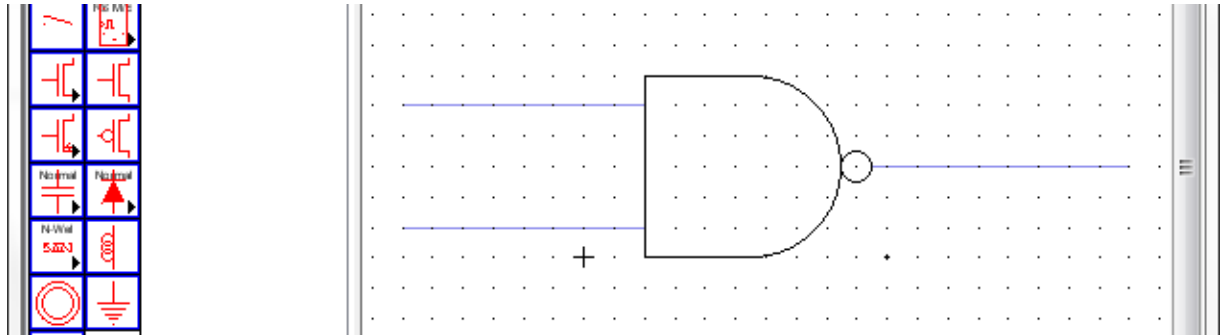


Рис. 4.12. Підключення дуг до виходів схеми І-НІ

Перевірте схему на помилки  $DRC$ . Ви отримаєте 3 помилки. Ці помилки пов'язані з тим, що дуги не визначені. Промаркуйте дуги, вхід  $A - in$ , вихід  $AnandB - out$ . До входу  $B$  підключіть живлення (рис. 4.13).

Також вставте  $Spice$  код з параметрами сигналів, живлення й опису моделей транзисторів для моделювання схеми ( $Components \rightarrow Misc \rightarrow Spice Code$ ):

```
vdd vdd 0 dc 5
vin in 0 dc 0 pulse 0 5 10n 1n
cload out 0 250f
.tran 0 40n
.include C:\Electric\C5_models.txt
```

Рядок  $Spice$  коду « $vin in 0 dc 0 pulse 0 5 10n 1n$ » означає, що ми підключили джерело сталої напруги ( $v$ ) з іменем « $in$ » між вузлами « $in$ » і «нульовим». Початковий номінал джерела дорівнює 0 В. У цьому рядку  $Spice$  коду наявний опис форми сигналу, який видає джерело при проведенні аналізу перехідних процесів:

*pulse unit puls delay up dow width peri*

Слово « $pulse$ » означає імпульс; « $unit$ » – початковий рівень імпульсу (В); « $puls$ » – рівень вершини імпульсу (В); « $delay$ » – затримка імпульсу (сек);

«*up*» – час наростання переднього фронту (сек); «*dow*» – час спаду заднього фронту (сек); «*width*» – ширина вершини імпульсу (сек); «*peri*» – період імпульсу (сек).

Таким чином, через 10 нс після початку аналізу з'явиться імпульс із амплітудою 5 В і тривалістю наростання переднього фронту – 1 нс. Параметри «*dow*», «*width*» і «*peri*» не задані, а отже, вони мають значення за замовчуванням:  $dow = vstep$ ,  $width = vstop$  і  $peri = vstop$ . Нагадаємо, що «*vstep*» і «*vstop*» задаються як параметри команди «*.tran*».

Команда «*sxxxx*» описує конденсатор:

*sname n1 n2 value*

буква «*s*» означає конденсатор; відразу після неї пишеться ім'я конденсатора (у нашому випадку конденсатор називається «*load*»); «*n1*», «*n2*» – вузли підключення; «*value*» – номінал, Ф.

Таким чином, рядок *Spice* коду «*load out 0 250f*» означає, що ми підключили конденсатор (*c*) з іменем «*load*» між вузлами «*out*» і «нульовим». Ємність конденсатора – 250 фФ (рис. 4.13).

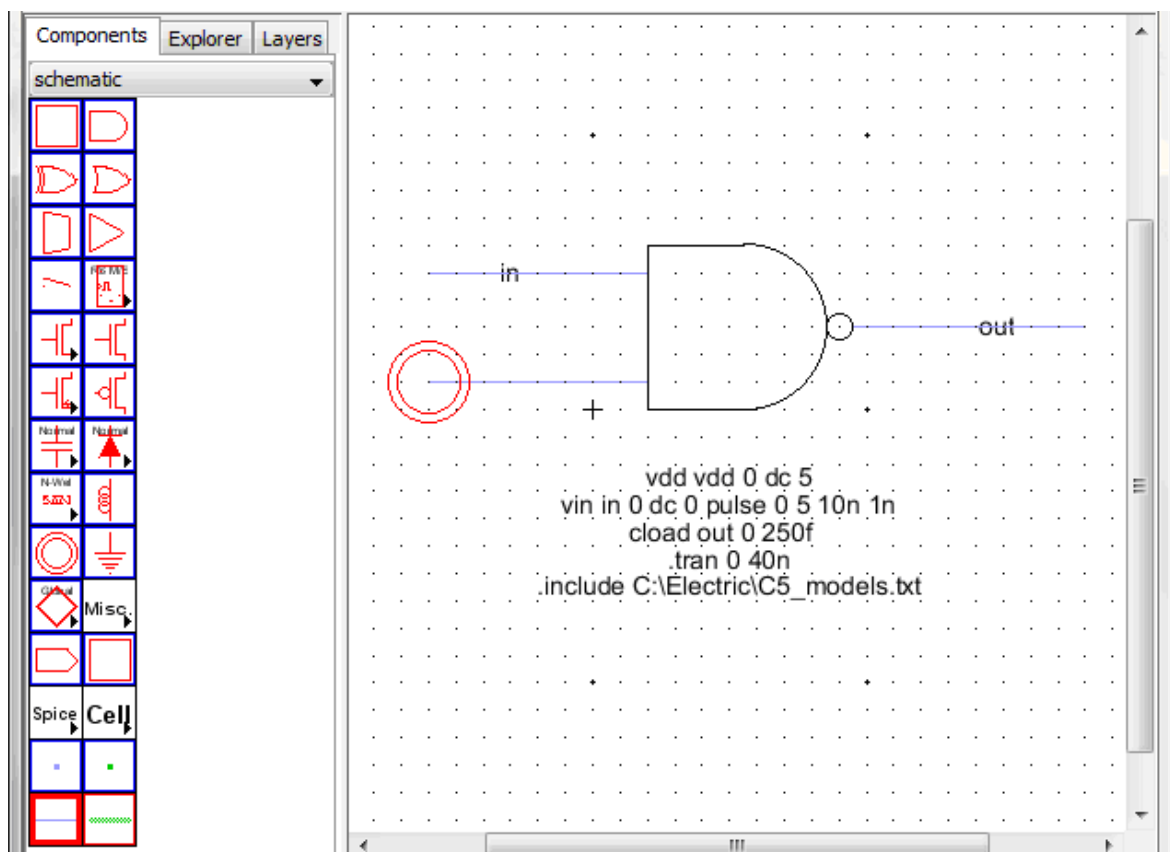


Рис. 4.13. Елемент І-НІ з промаркованими дугами та *Spice* кодом

Перевірте схему на помилки *DRC*. Отримаєте 2 помилки. Після перегляду місць, де допущені помилки (*Shift+>/<*), бачимо, що живлення не підключене до дуги. Для виправлення подібних помилок виділіть місце помилки й скористайтеся меню *Tools* → *Routing* → *Auto Stitch Highlighted Now* або натисніть клавішу *F2*. Тепер вузол приєднаний до дуги. Перевірте ще раз схему на помилки *DRC*, виявите ще одну помилку – зайвий контакт. Видаліть зайві контакти: *Edit* → *Cleanup Cell* → *Cleanup Pins*.

Після цього помилок не має бути.

Промодельуйте роботу схеми в *LTspice* (рис. 4.14) і перейдіть у зонд *Electric* для обробки отриманих *LTspice* результатів (рис. 4.15).

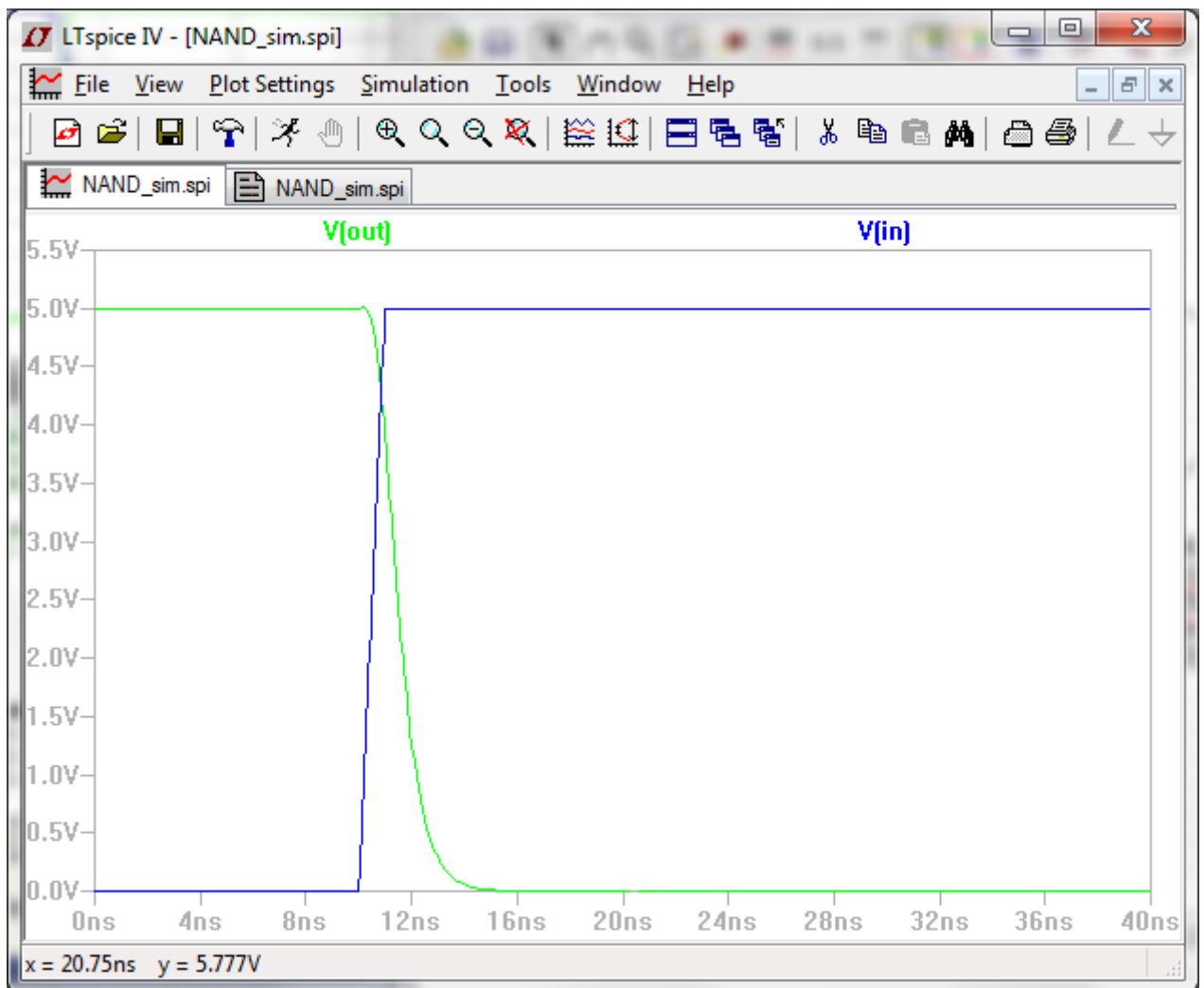


Рис. 4.14. Часова характеристика елемента І-НІ в *LTspice*

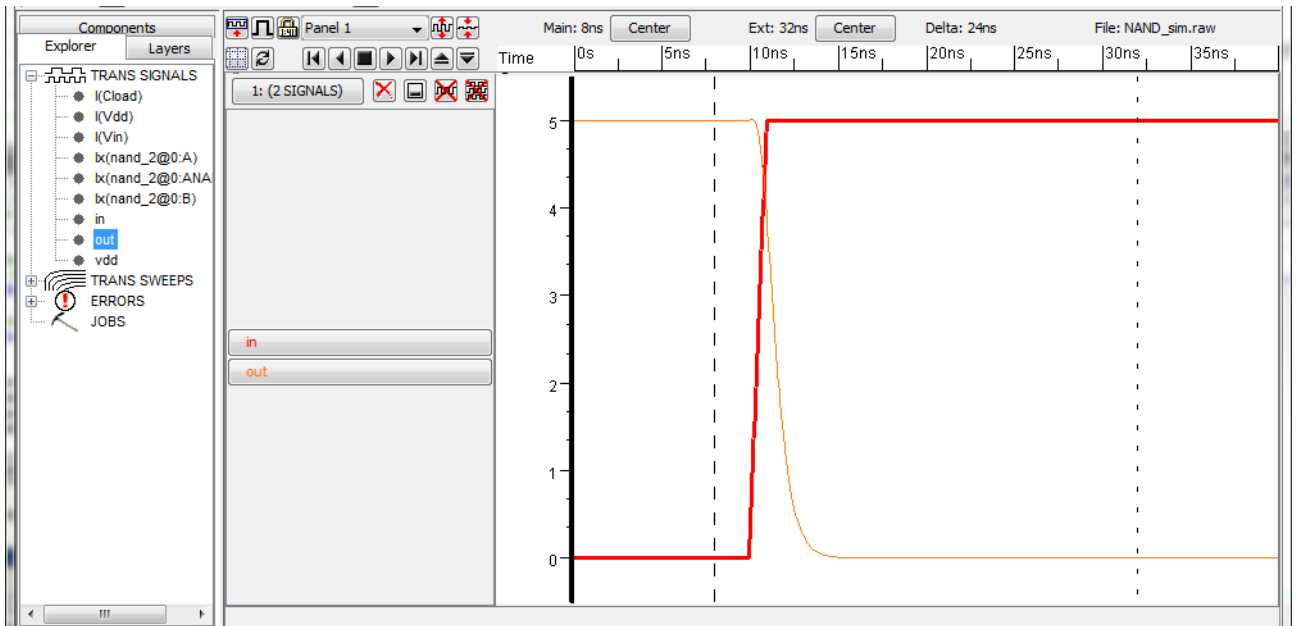


Рис. 4.15. Часова характеристика елемента І-НІ в *Electric*

Після того, як ми переконалися в справності схеми, перейдемо до проектування її топології.

Відкрийте комірку  $NAND\_2\{lay\}$ , де знаходиться спроектована в минулій практичній роботі топологія інвертора. Видаліть дуги, контакти й екпорти, поки не отримаєте топологію як на рис. 4.16.

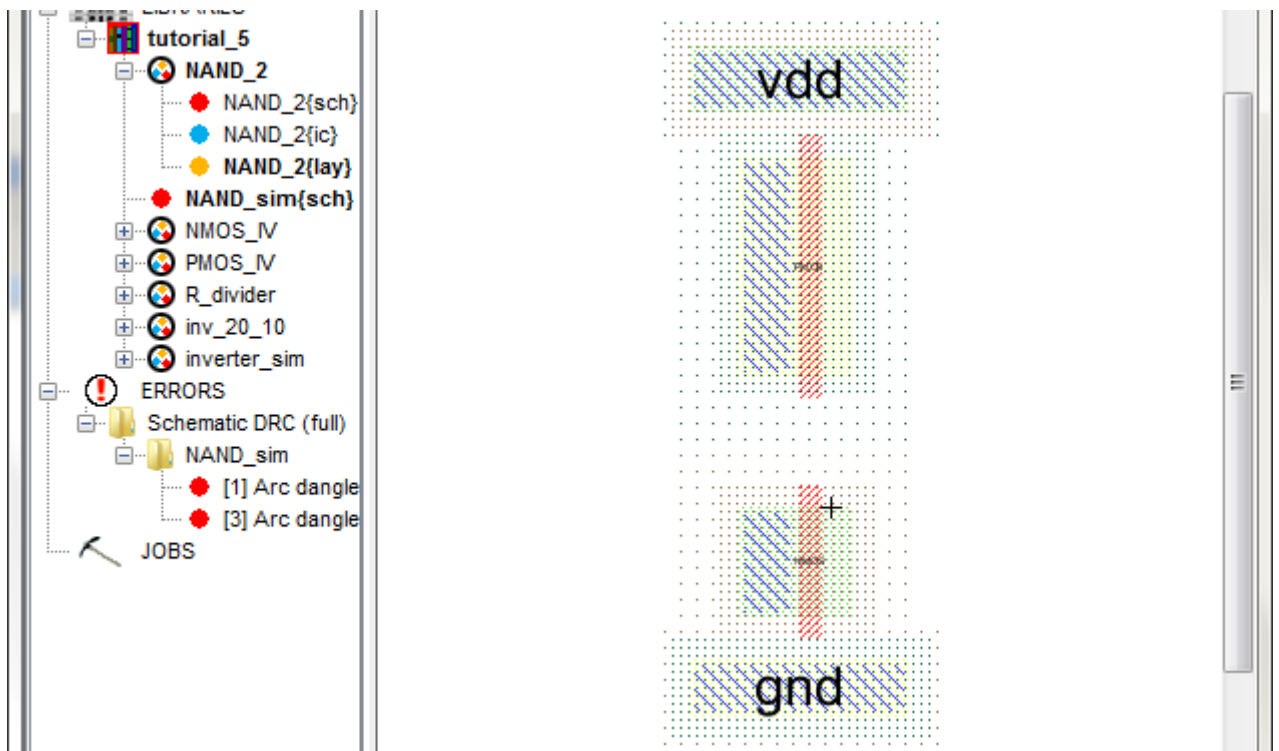


Рис. 4.16. До побудови топології схеми І-НІ

Перемістіть  $pAct$  і  $nAct$  вузли від заслона транзистора й змініть ширину вузлів  $pAct$  і  $P$ -Transistor на значення 10 (рис. 4.17).

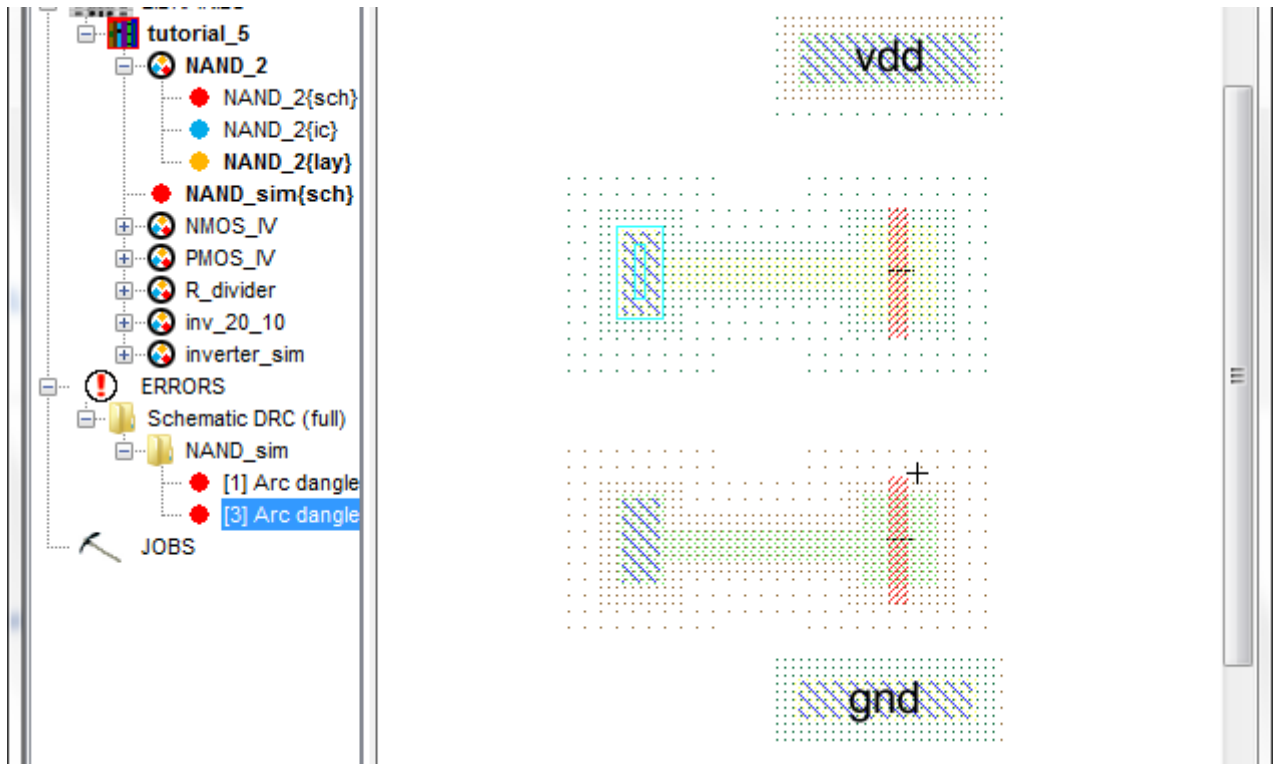


Рис. 4.17. До побудови топології схеми І-НІ

Скопіюйте  $pAct$  області 2 рази,  $P$ -Transistor один раз,  $nAct$  один раз,  $N$ -Transistor один раз. Розташуйте їх так, як зазначено на рис. 4.18.

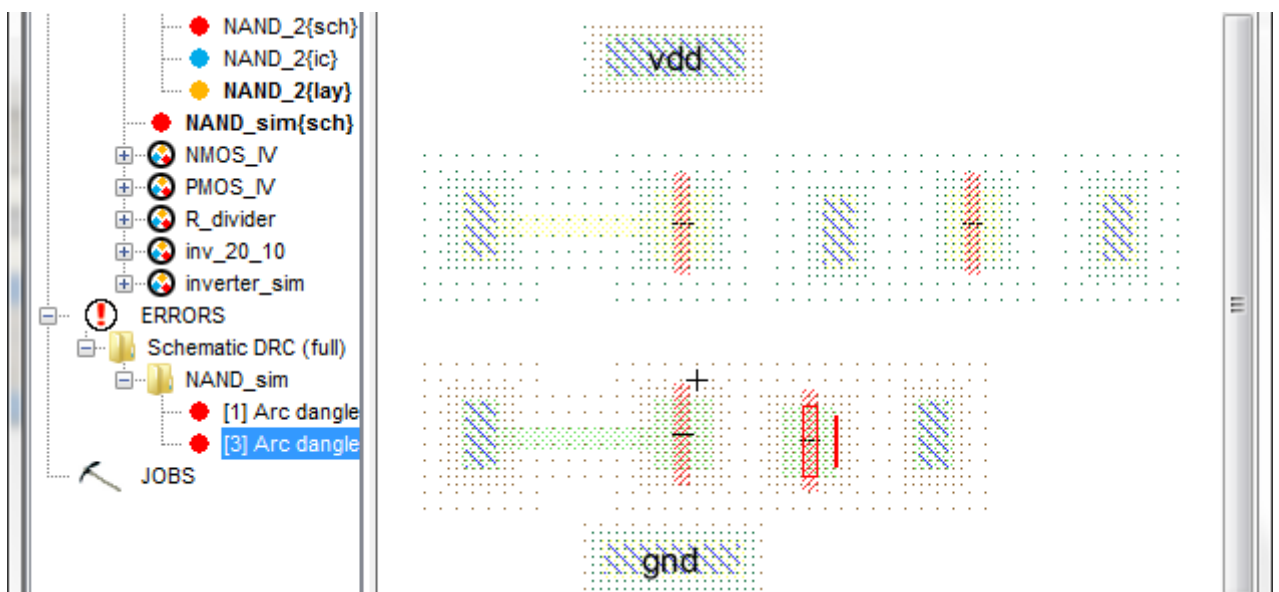


Рис. 4.18. До побудови топології схеми І-НІ

З'єднаєте їх між собою (рис. 4.19).

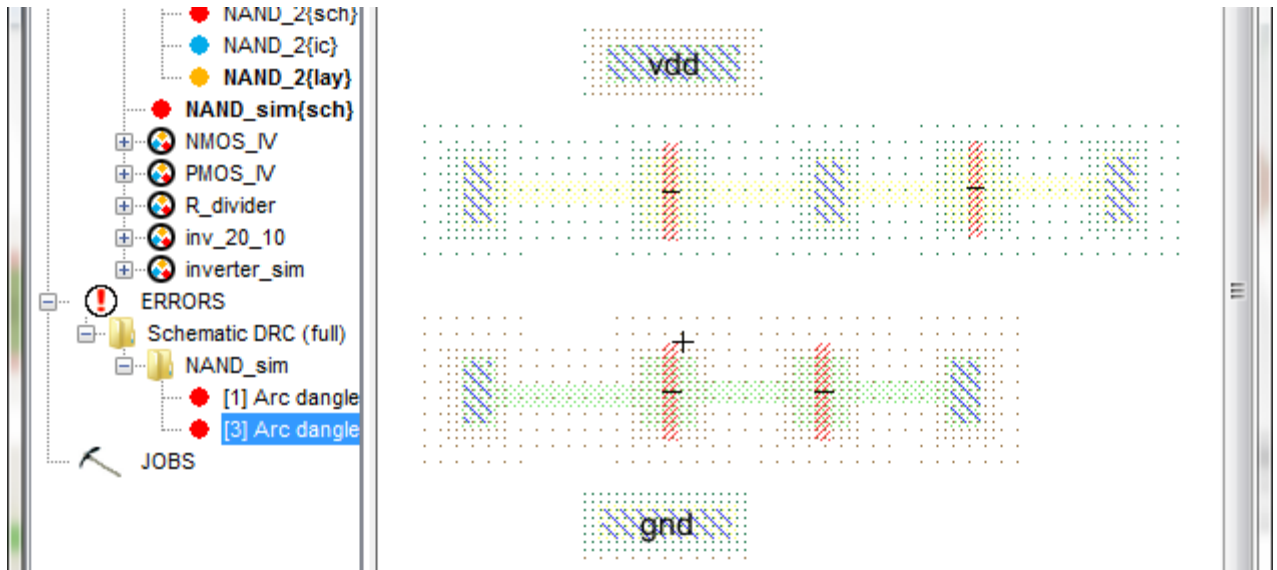


Рис. 4.19. До побудови топології схеми І-НІ

Зсуньте вузли разом, як показано на рис. 4.20. Переконайтеся, що помилки *DRC* відсутні.

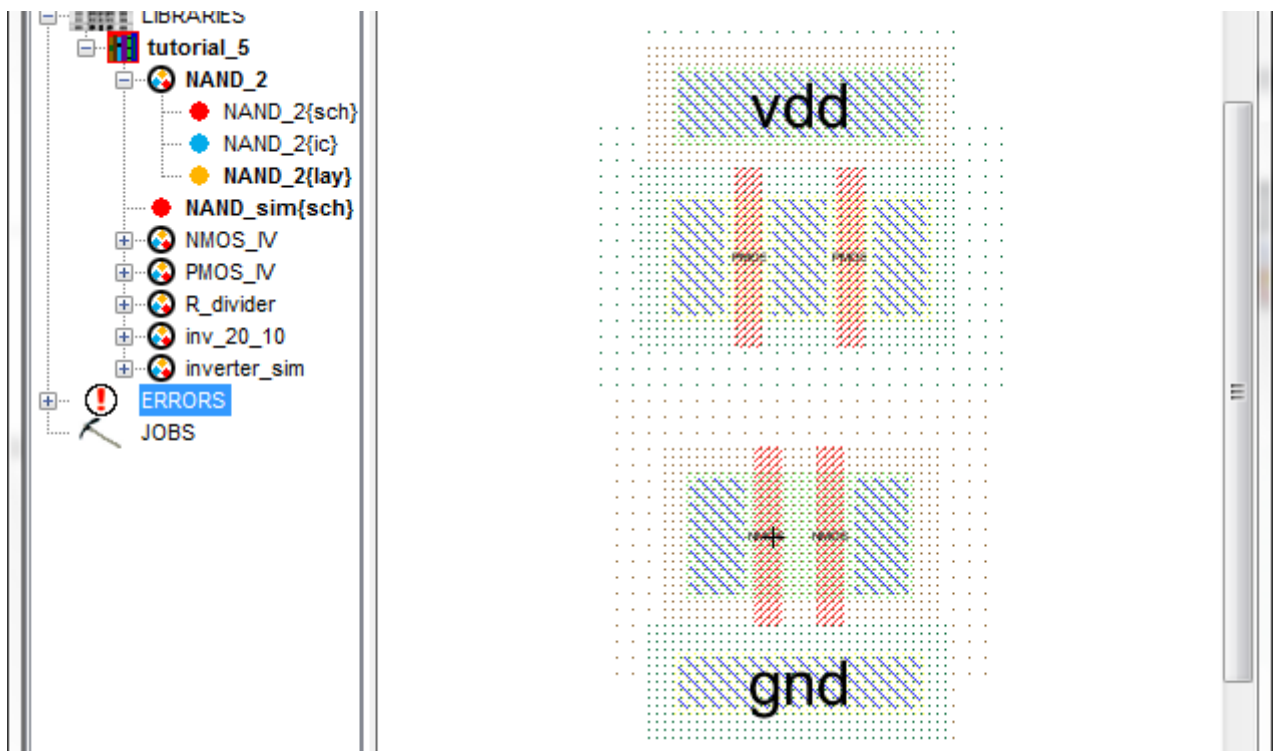


Рис. 4.20. До побудови топології схеми І-НІ

Збільшіть розміри  $pWell$  і  $nWell$  до значення 30 (рис. 4.21).

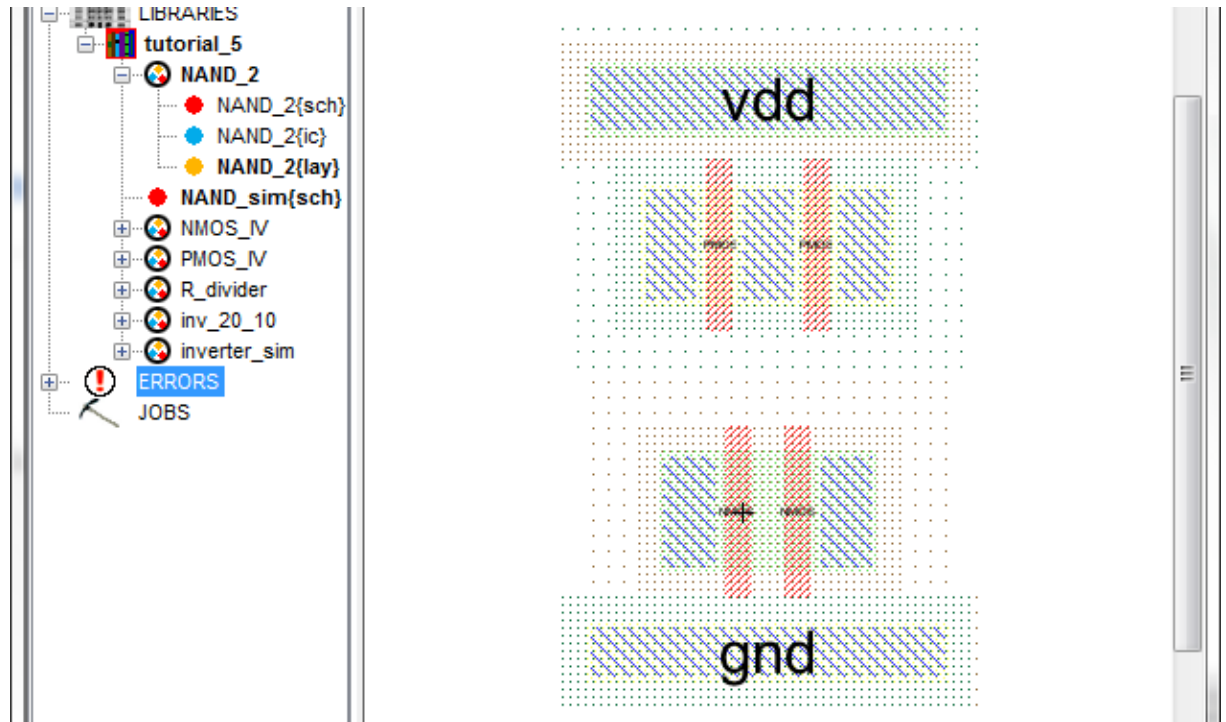


Рис. 4.21. До побудови топології схеми І-НІ

З'єднайте вузли дугами, як показано на рис. 4.22. Переконайтеся, що використовуєте металеві дуги першого шару металізації ( $metal-1$ ). Також з'єднайте заслони транзисторів, як показано на рис. 4.22.

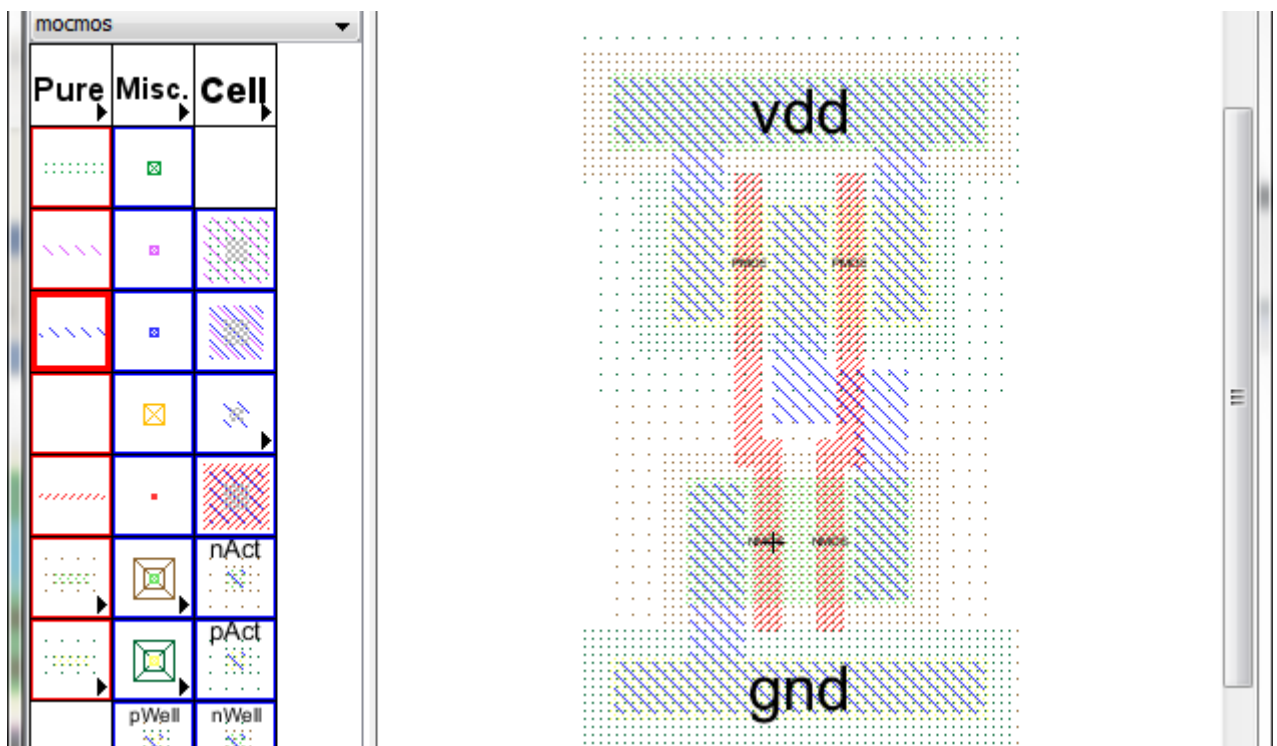


Рис. 4.22. До побудови топології схеми І-НІ

Додайте контактні вузли для заслонів *metall-poly1* і з'єднайте їх із заслоном (рис. 4.23).

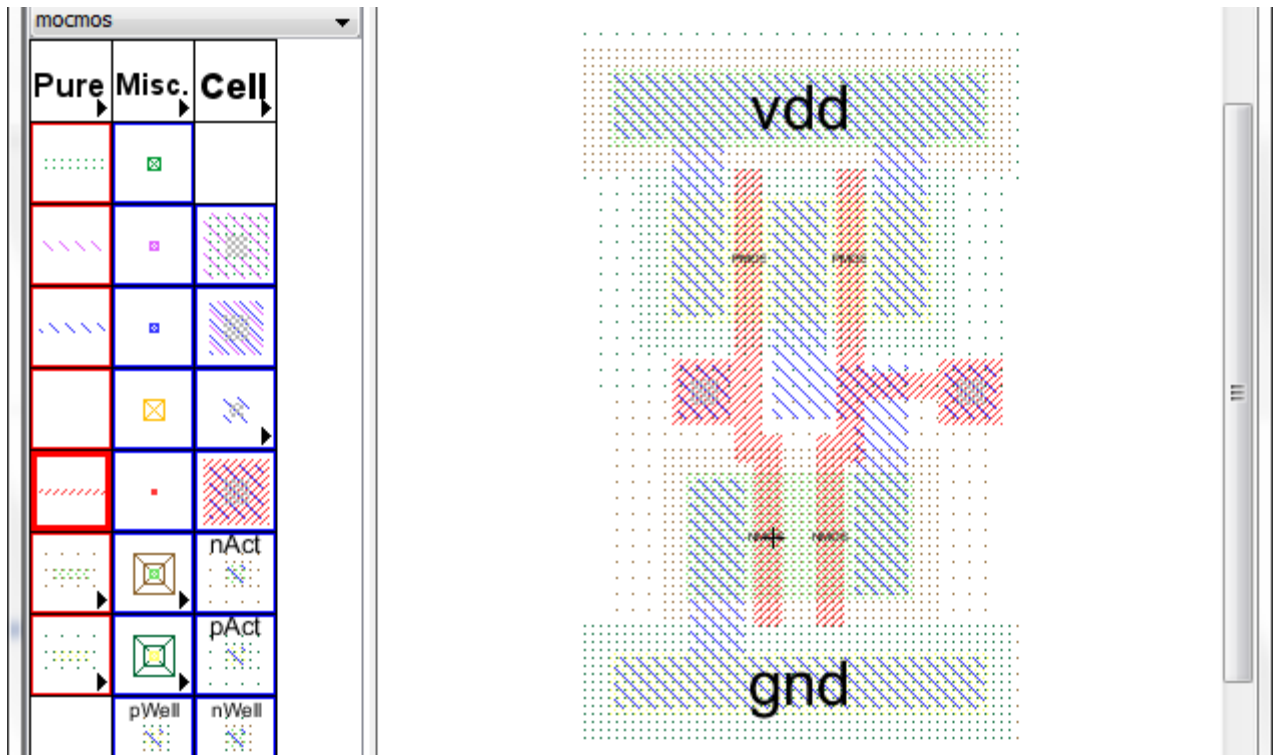


Рис. 4.23. Топологія схеми І-НІ

Перевірте на наявність помилок, якщо вони є – виправте їх.

Щоб побачити топологію в 3D використовуйте меню *Window* → *3D Window* → *3D View* (рис. 4.24). Режим перегляду в 3D доступний, тільки якщо встановлена програма *Java3d*.

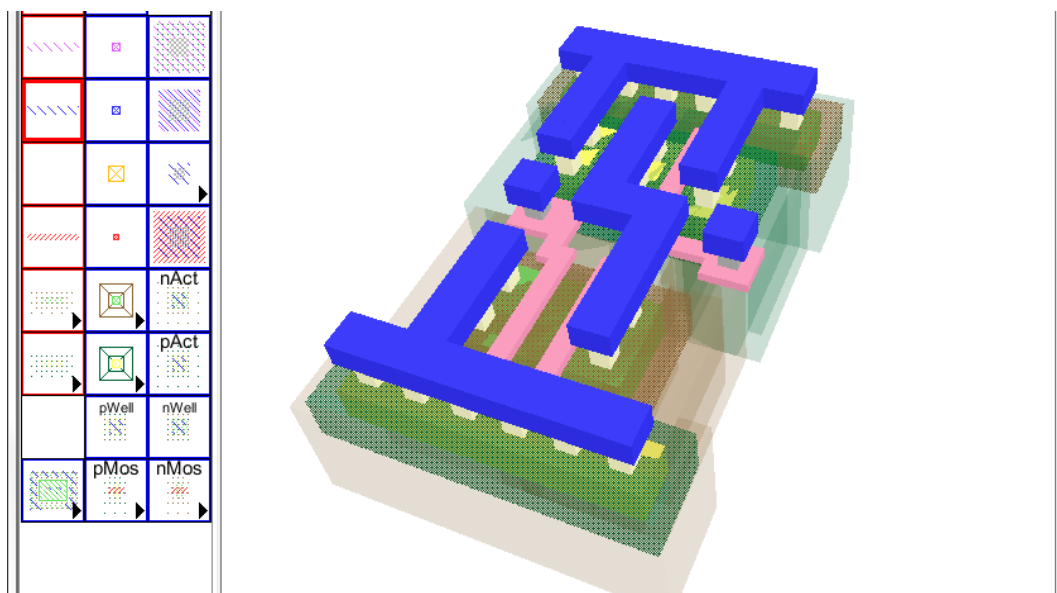


Рис. 4.24. Топологія схеми І-НІ в 3D



Додайте вузли метал1–метал2 контакти (*metal1–metal2*) і з'єднаєте їх металевими дугами (рис. 4.25). Розмістіть контакти метал1–метал2 над контактами метал1–полікремній1. Для кращого розуміння «що відбулося», переглянете топологію в 3D.

Перевірте на наявність помилок.

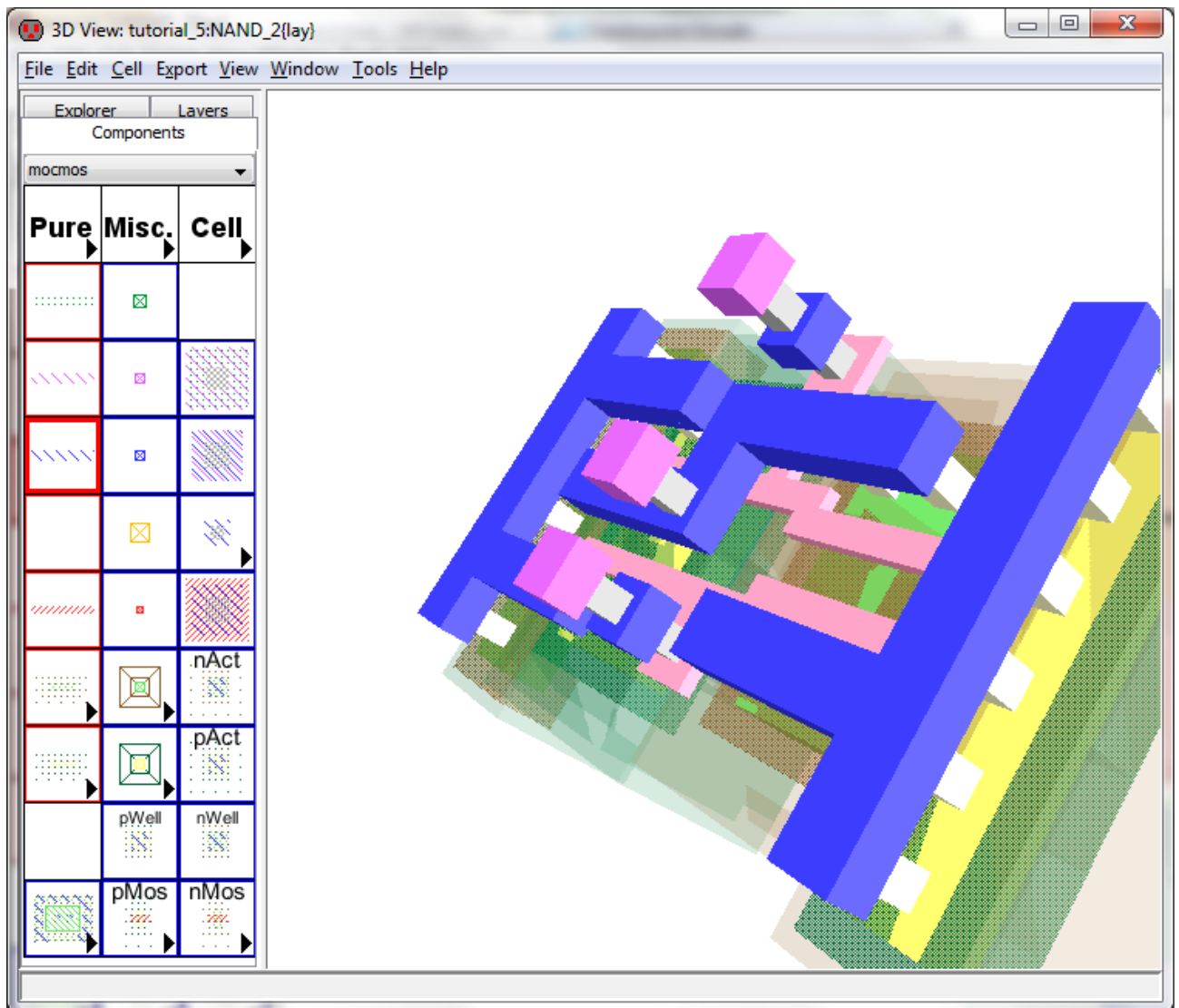


Рис. 4.25. Топологія схеми І-НІ в 3D

Експортуйте контакти *metal1-metal2*, як показано на рис. 4.26. Розмір тексту експорту – 3.

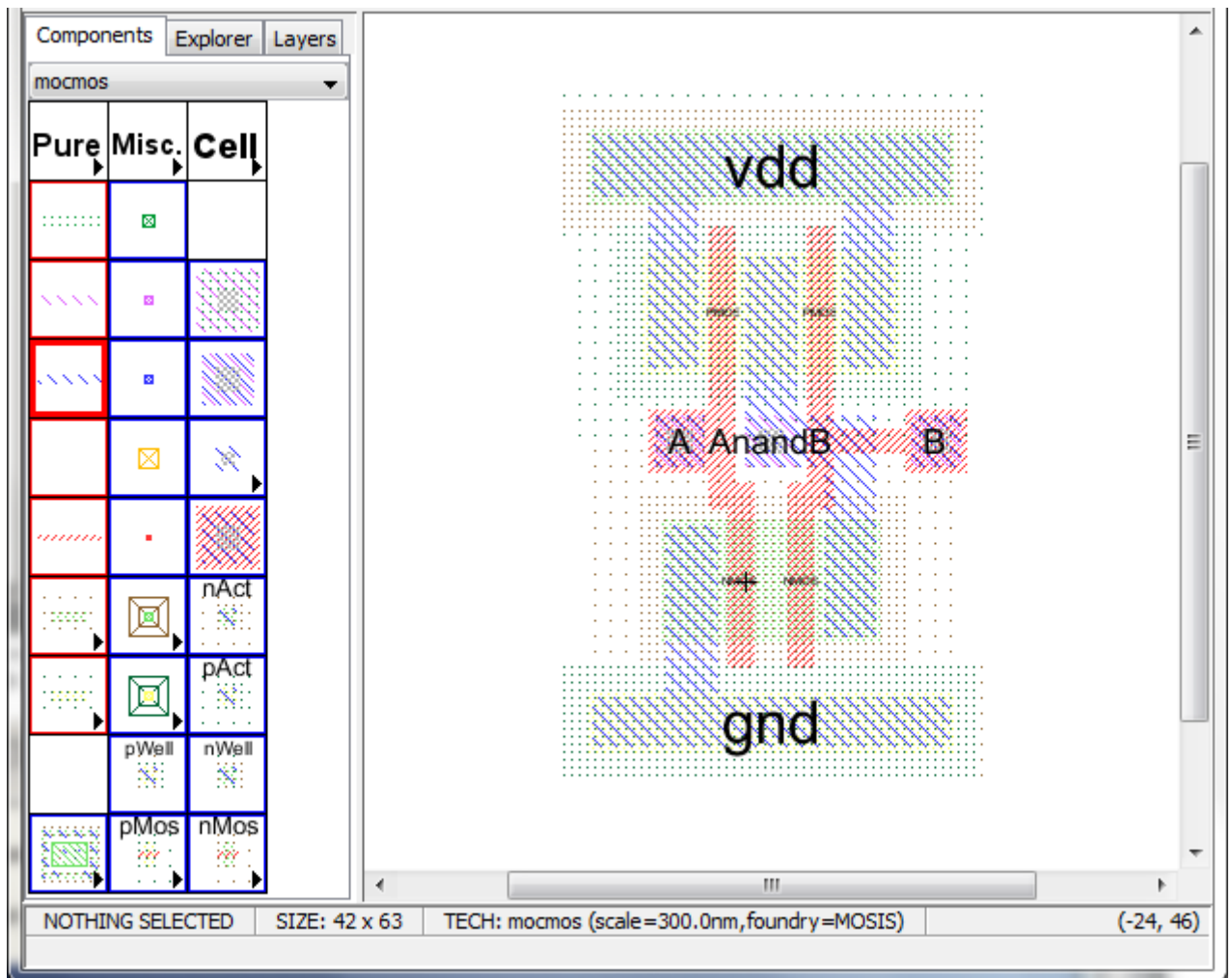


Рис. 4.26. Топологія схеми І-НІ з експортованими контактами

Перевірте макет на помилки *DRC* (*F5*), *ERC*, *NCC* (*L*). Крім помилок невідповідності (*NCC*) (рис. 4.27) макетів ніяких помилок не має бути.

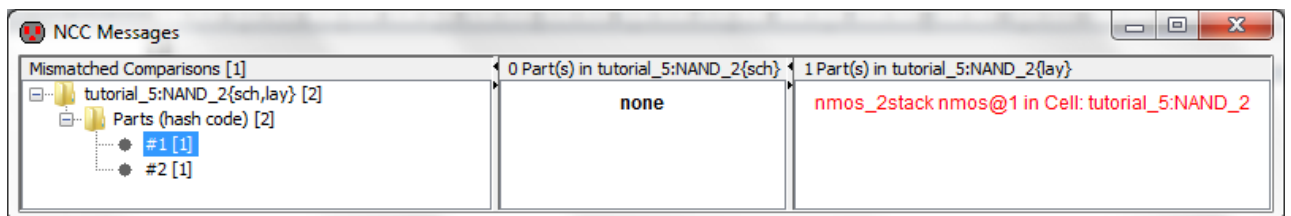


Рис. 4.27. Повідомлення про *NCC* помилки

Ці помилки викликані тим, що ми переплутали входи *A* і *B* при експорті контактів у комірці топології. Виправте помилку (рис. 4.28).

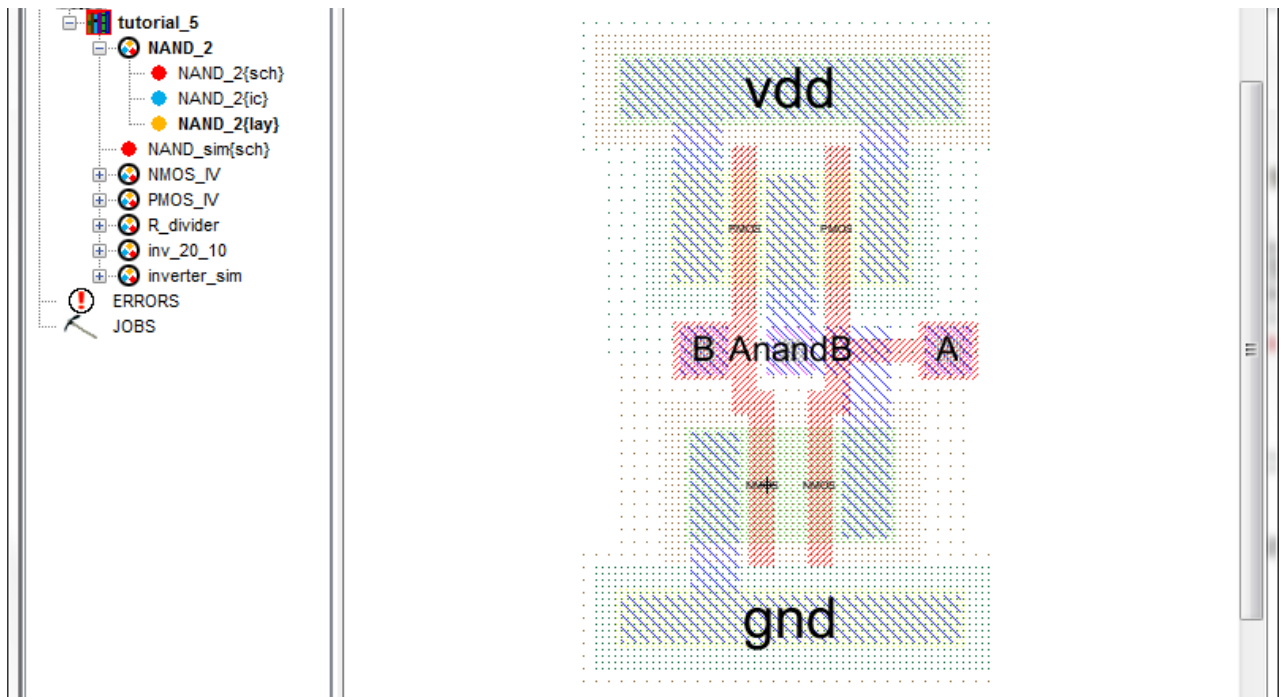


Рис. 4.28. Топологія схеми І-НІ з правильно експортованими контактами

Перевірте макет на помилки. Тепер їх не має бути.

Створіть макет комірки  $NAND\_sim\{lay\}$ . Вставте в неї комірку  $NAND\_2\{lay\}$ . Вставте виводи як показано на рис. 4.29.

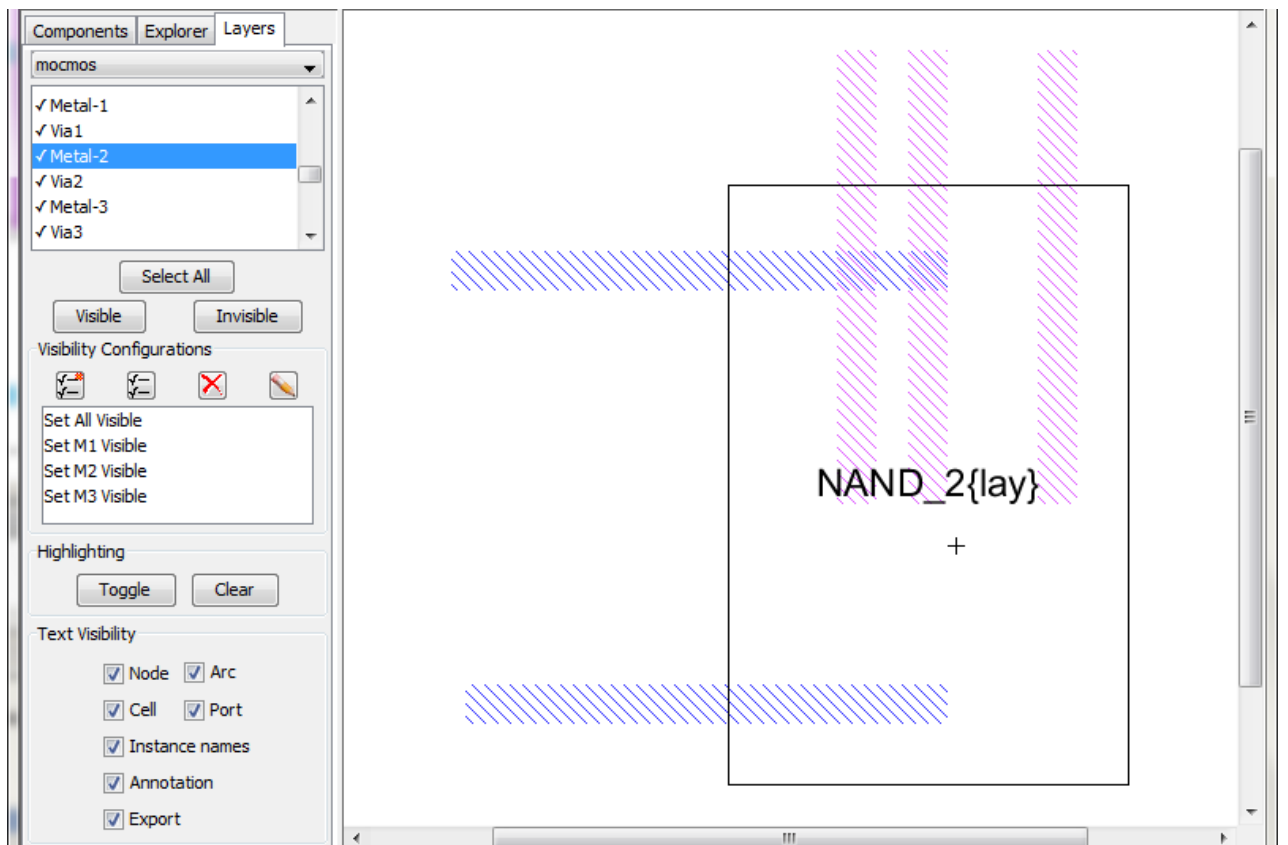


Рис. 4.29. Виводи у комірці  $NAND\_sim\{lay\}$

Промаркуйте металеві дуги *metal2* і експоруйте *vdd* і *gnd*. Маркування дуг зробіть відповідно до комірки *NAND\_sim{sch}*.

Топологія не проходить перевірку *NCC* (рис. 4.30).

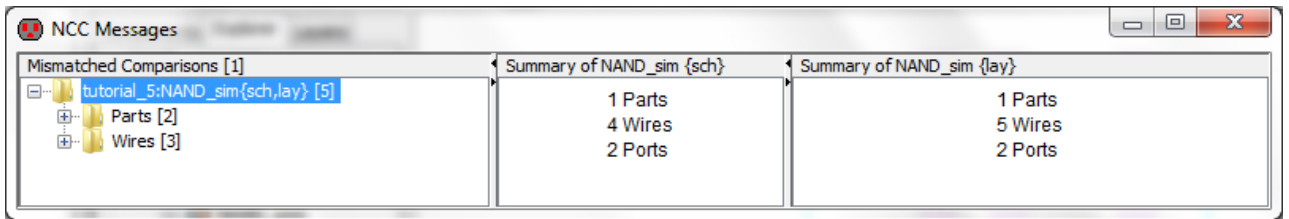


Рис. 4.30. Повідомлення про *NCC* помилки

Ця помилка викликана тим, що в *NAND\_sim{sch}* на вхід *B* ми подали живлення, а в топології вхід *B* не підключений до живлення. Приєднайте вхід *B* до *vdd*. Скопіюйте у комірку *Spice* код для подальшого моделювання (рис. 4.31).

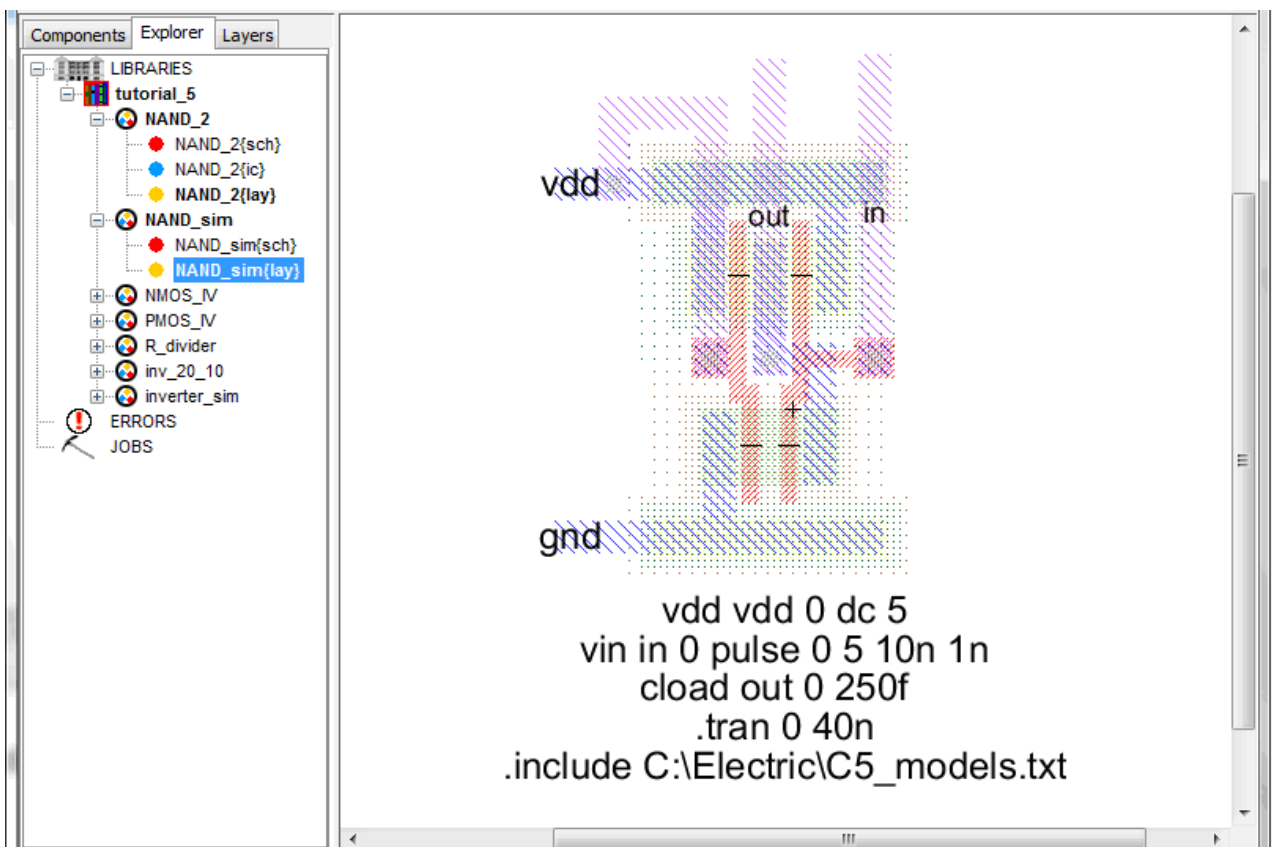


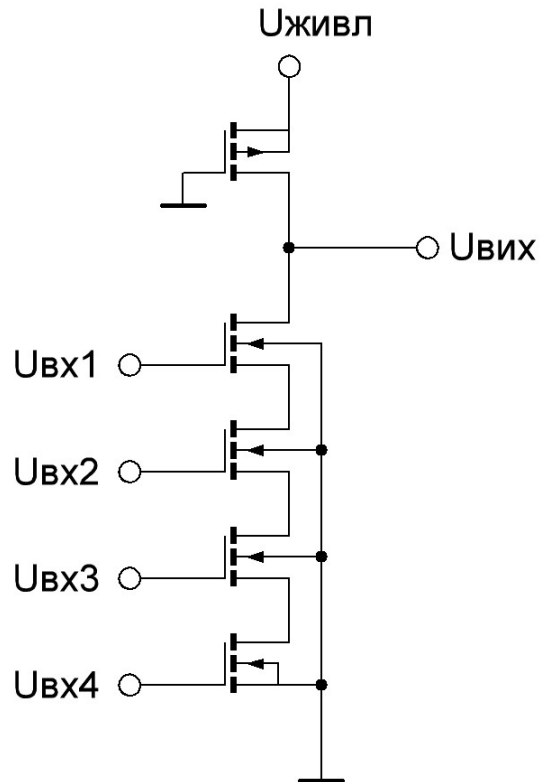
Рис. 4.31. Топологія схеми I-НІ зі *Spice* кодом

Якщо помилки *DRC*, *NCC* і *ERC* відсутні, то практична робота зроблена вірно. Збережіть бібліотеку.

## Варіанти завдань

### Варіант 1

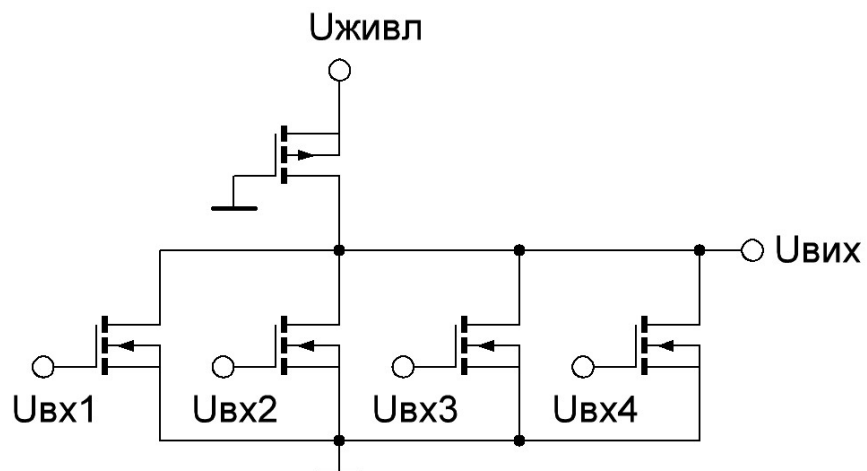
Проектування псевдо-*n*-МОП схеми 4 І-НІ



Установити ємність конденсатора 25 фФ.

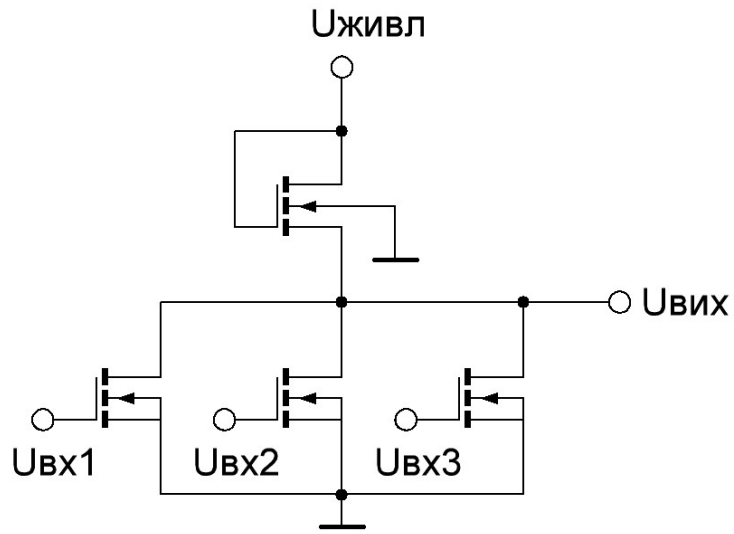
### Варіант 2

Проектування псевдо-*n*-МОП схеми 4 АБО-НІ



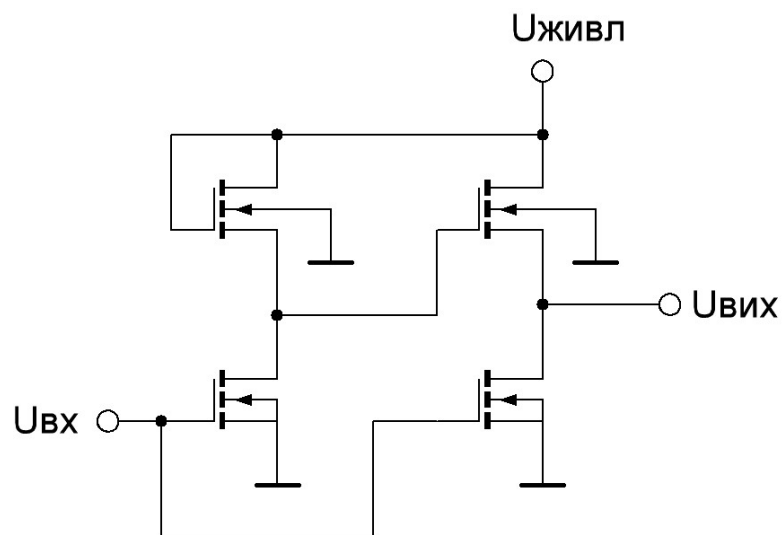
### Варіант 3

Проектування  $n$ -МОН схеми 3 АБО-НІ



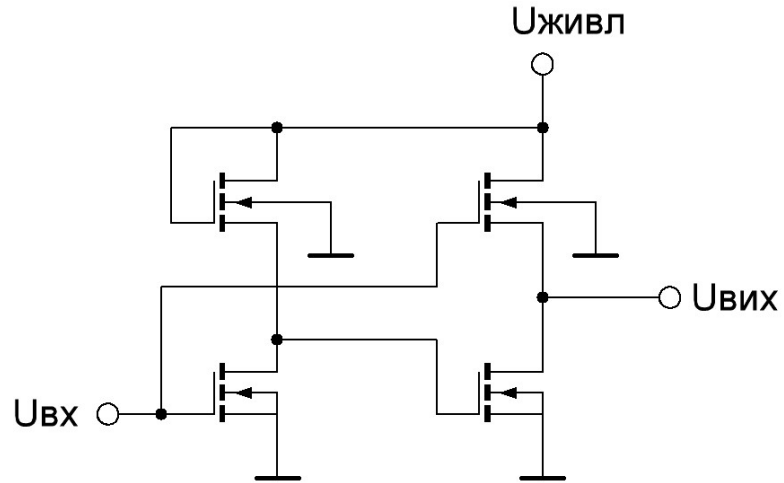
### Варіант 4

Проектування  $n$ -МОН схеми буферного підсилювача з інвертуванням сигналу



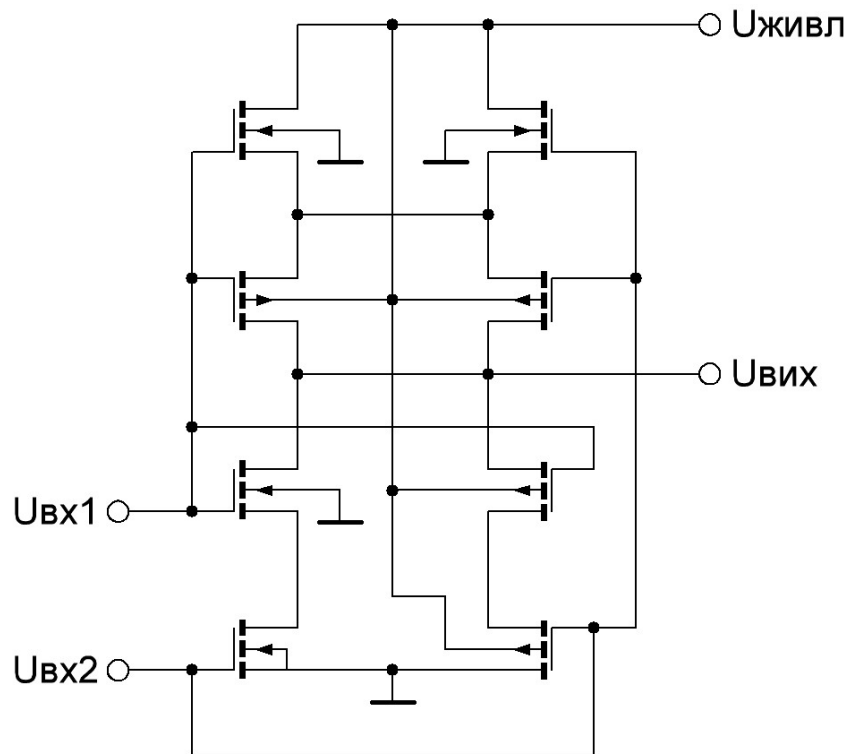
### Варіант 5

Проектування  $n$ -МОН схеми буферного підсилювача без інвертування сигналу



### Варіант 6

Проектування КМОН схеми І-АБО-НІ



## Контрольні питання

1. Дайте визначення поняття «комбінаційна логіка».
2. Перерахуйте основні метрики розробки комбінаційних пристроїв.
3. Яка з метрик є переважною?
4. Назвіть основні переваги КМОН-структур.
5. Які схеми називають статичними?
6. Яку функцію виконує навантажувальний ланцюг статичного КМОН-елемента?
7. Яку функцію виконує розвантажувальний ланцюг статичного КМОН-елемента?
8. Для чого навантажувальний і розвантажувальний ланцюги створюють за взаємовиключним принципом?
9. Транзистори якого типу використовують для створення розвантажувального ланцюга? Чому?
10. Транзистори якого типу використовують для створення навантажувального ланцюга? Чому?
11. Якій функції відповідають з'єднані послідовно  $n$ -МОН пристрої?
12. Якій функції відповідають з'єднані паралельно  $n$ -МОН пристрої?
13. Якій функції відповідають з'єднані послідовно  $p$ -МОН пристрої?
14. Якій функції відповідають з'єднані паралельно  $p$ -МОН пристрої?
15. Як реалізувати неінвертуючі булеві функції, використовуючи КМОН-логіку?
16. Скільки потрібно транзисторів для реалізації логічного елемента з  $N$  входами?
17. Чому дорівнює повний розмах сигналу в КМОН-елементах?
18. Чому КМОН схеми не мають статичного розсіювання енергії?
19. Від чого залежить розсіювання потужності КМОН-елементів?
20. З чим пов'язана поява імпульсних перешкод в КМОН-схемі?
21. Опишіть суть диференційної каскадної логіки.
22. Чому в диференціальних схемах відсутнє статичне розсіювання потужності?



## Практичне заняття №5

### Проектування кільцевого генератора

Збережіть бібліотеку *tutorial\_4.jelib* під іменем *tutorial\_5.jelib* у папку *C:\Electric*. Простежте, щоб файл *C5\_models.txt* зі *Spice* моделями перебував у папці *C:\Electric*. У бібліотеці *tutorial\_5.jelib* створіть комірку *inverter\_4p{sch}* (4*p* вказує на чотири інвертори підключених паралельно). У цій комірці інвертор із комірки *inv\_20\_10* буде повторюватися 4 рази. Розмістіть інвертори в комірці *inverter\_4p{sch}* як показано на рис. 5.1. Спробуйте різні способи вставлення вузлів.

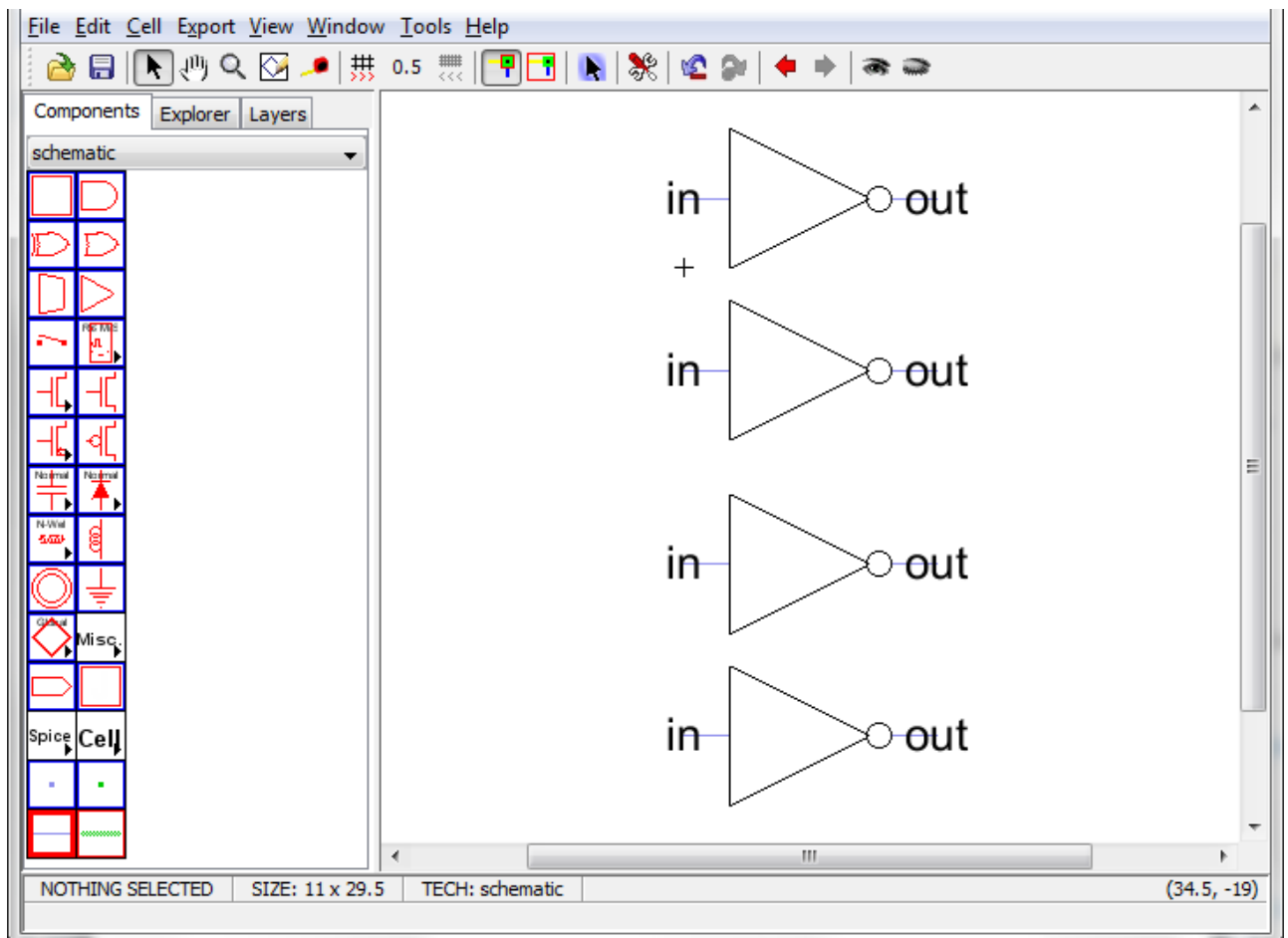


Рис. 5.1. Чотири інвертори у комірці *inverter\_4p{sch}*

Створіть нову версію комірки *inverter\_4p{sch}*, правий клац по комірці *inverter\_4p{sch}*, у контекстному меню *Create New Version of Cell* (рис. 5.2).

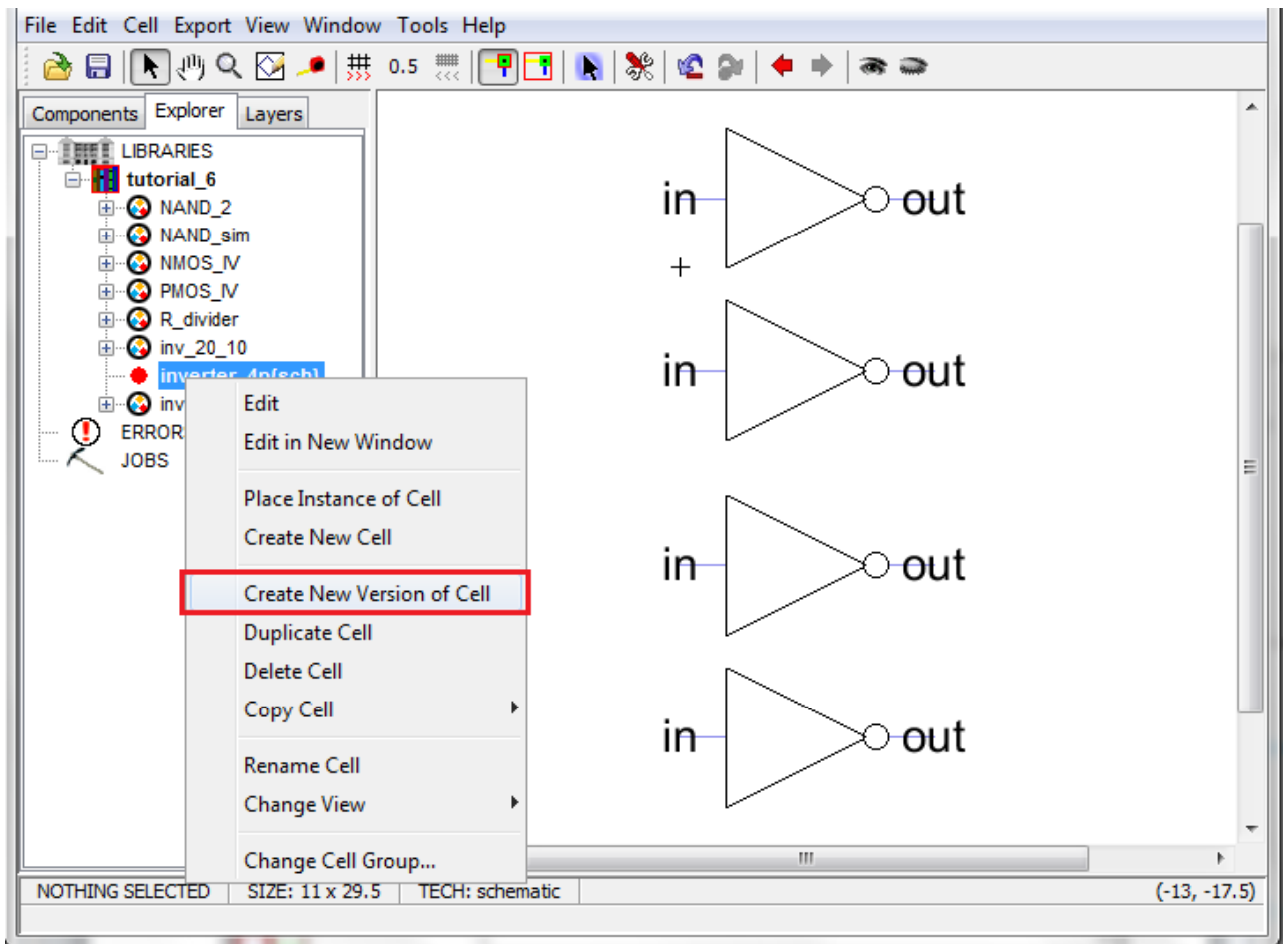


Рис. 5.2. Створення нової версії комірки  $inverter\_4p\{sch\}$

Зверніть увагу, що при цьому колишня версія комірки  $inverter\_4p;1\{sch\}$  залишилася.

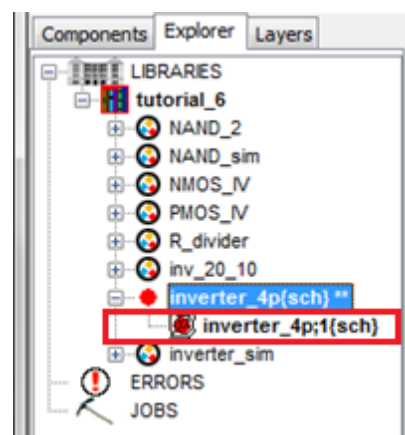


Рис. 5.3. Нова і стара комірки у бібліотеці

Відкрийте комірку  $inverter\_4p;1\{sch\}$  і видаліть 3 верхніх інвертора (рис. 5.4).

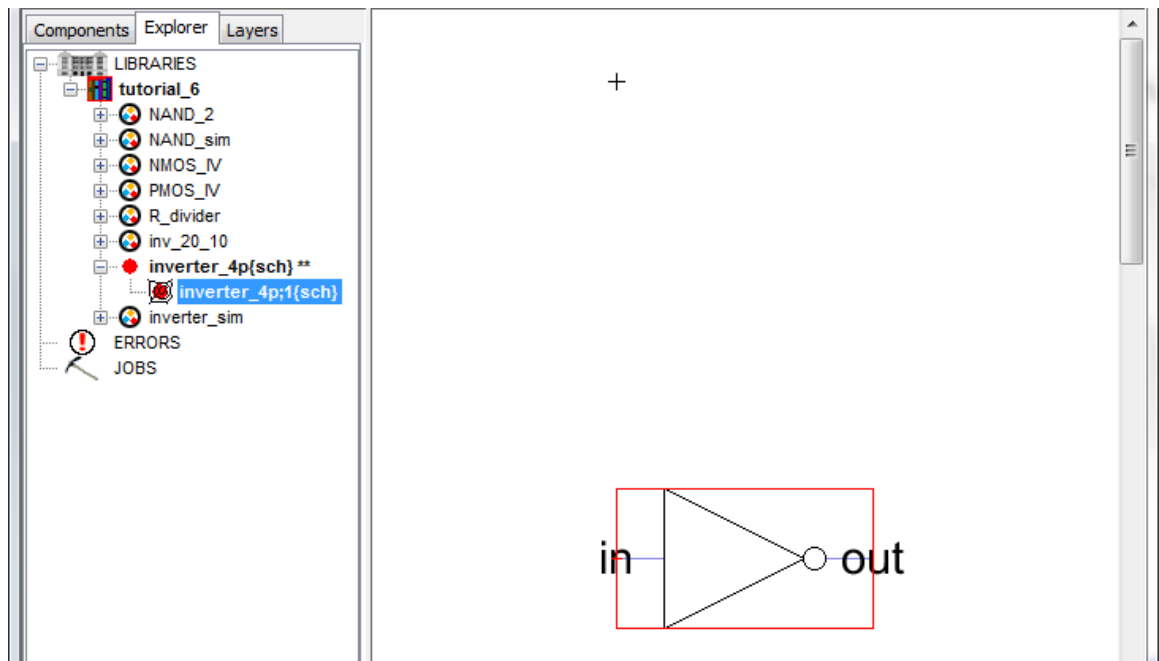


Рис. 5.4. Видалення трьох інверторів

Виділіть вузол *inv\_20\_10* (інвертор, який залишився) і перейдіть у меню *Edit* → *Array* (або клавіша *F6*). У вікні, що з'явилося, поставте значення повтор по осі  $X=1$  (*X repeat factor*), по  $Y=4$  (*Y repeat factor*). Перекриття (*edge overlaps*) по обом осям – 0 (рис. 5.5). Результат наведено на рис. 5.6.

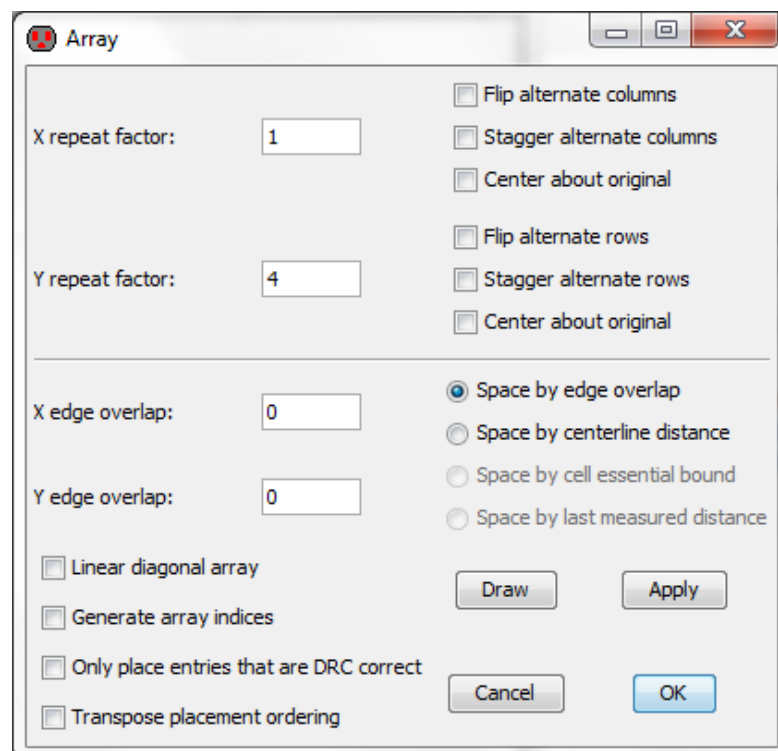


Рис. 5.5. Вставлення параметрів багатократного копіювання

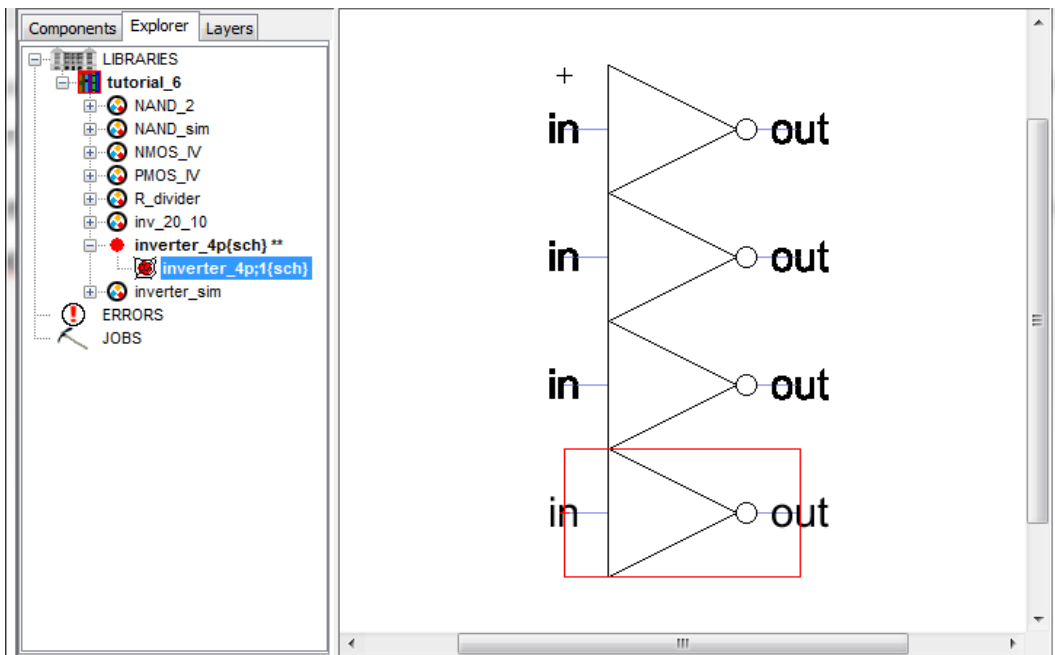


Рис. 5.6. Створення масиву інверторів

Оскільки значення перекриття дорівнює 0, то інвертори примикають один до одного.

Скауйте команду «Array» (*Ctrl+Z*) і викличте *Array* знову. Поставте значення перекриття по *Y* “-5”. Ви побачите, що відстань між піктограмами збільшилася (рис. 5.7).

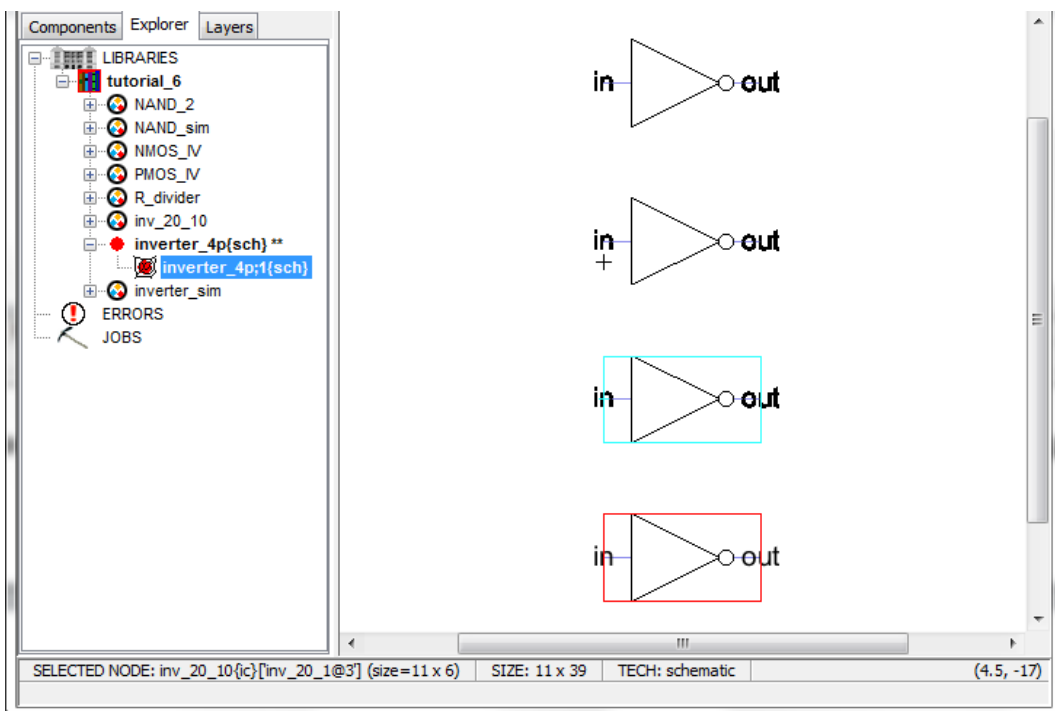


Рис. 5.7. Масив інверторів

Цей інструмент дуже корисний при проектуванні як схем, так і топологій.

Повернемося у комірку *inverter\_4p{sch}*, залишіть тільки одну піктограму інвертора, інші видаліть. Відцентруйте піктограму інвертора, задавши координати його розташування  $x = 0, y = 0$  (рис. 5.8).

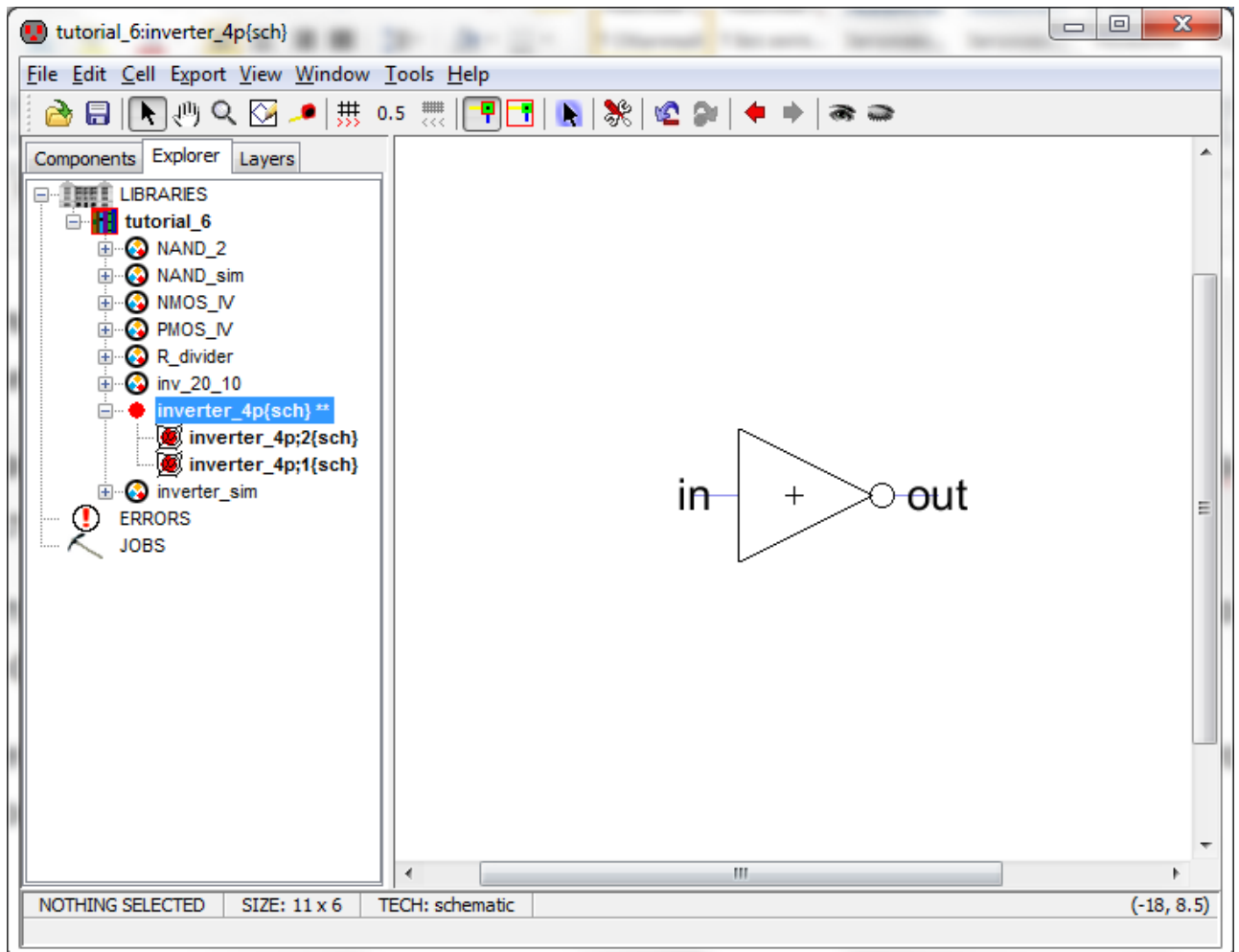


Рис. 5.8. Відцентрована піктограма інвертора

Перейменуйте вузол, указавши в імені (у властивостях вузла) індикацію інверторів від 0 до 3 (рис. 5.9).

Тепер необхідно приєднати дуги до інвертора. У нашому випадку звичайну дугу приєднати не можна, тому що всі чотири входи інвертора будуть замкнені (рис. 5.10).

Якщо ми прагнемо мати доступ до кожного входу інвертора, то для цього нам потрібно скористатися шиною дуг (рис. 5.11).

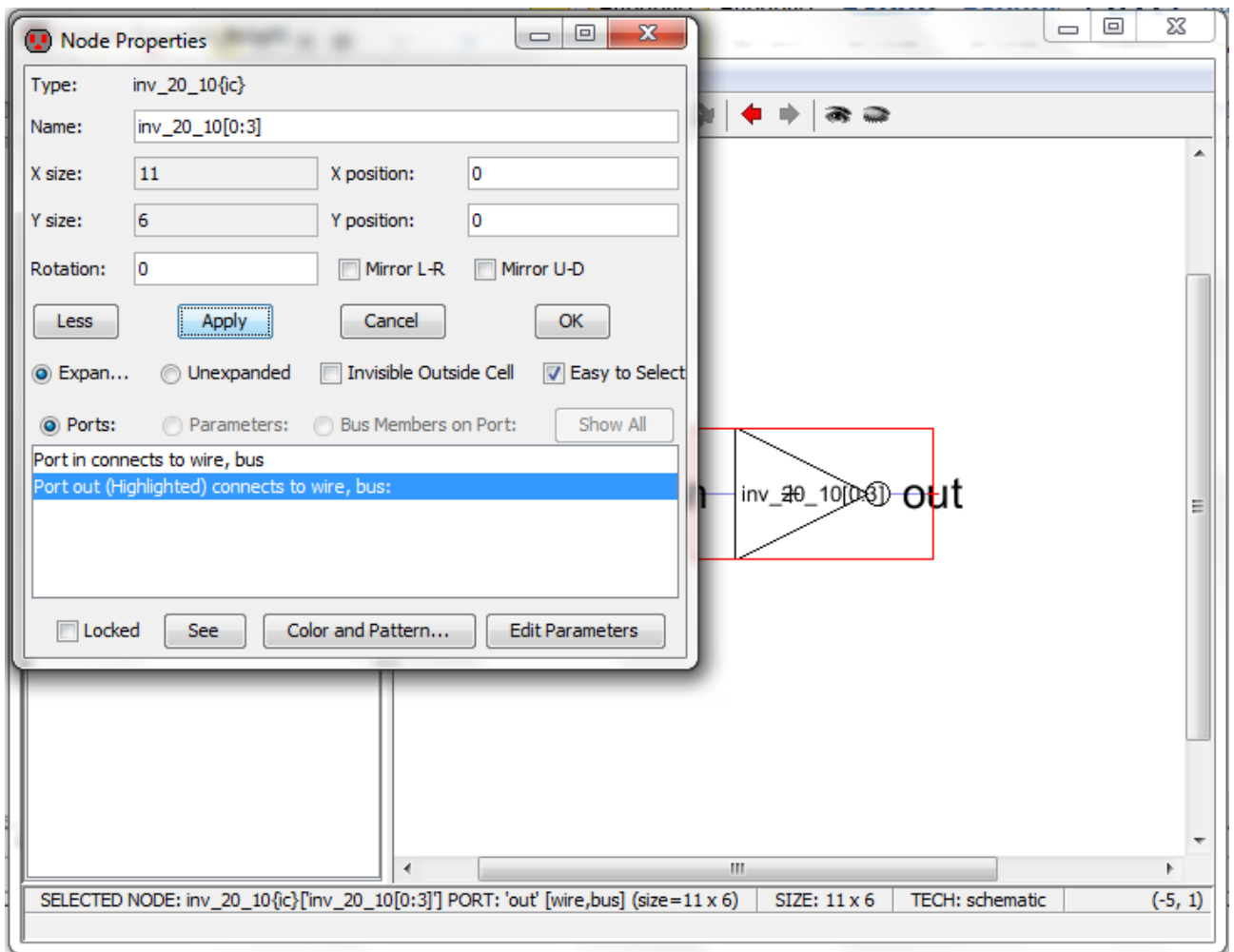


Рис. 5.9. Створення масиву інверторів

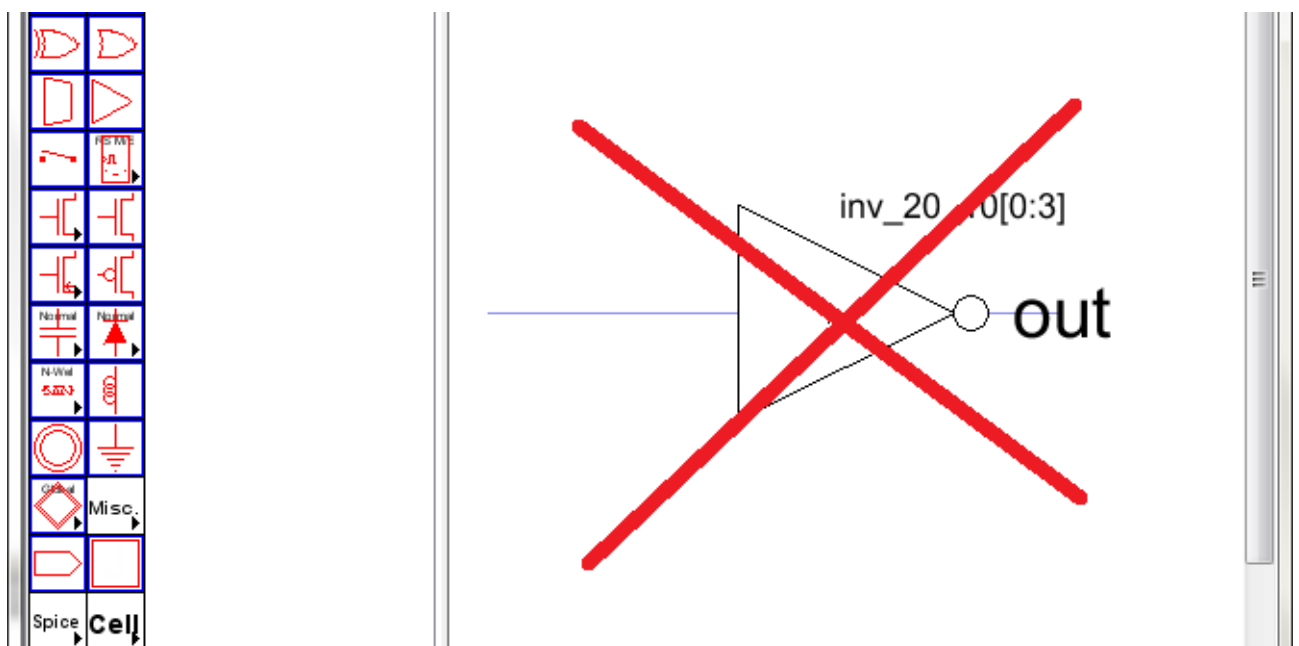


Рис. 5.10. Не можна під'єднувати звичайну дугу до масиву інверторів

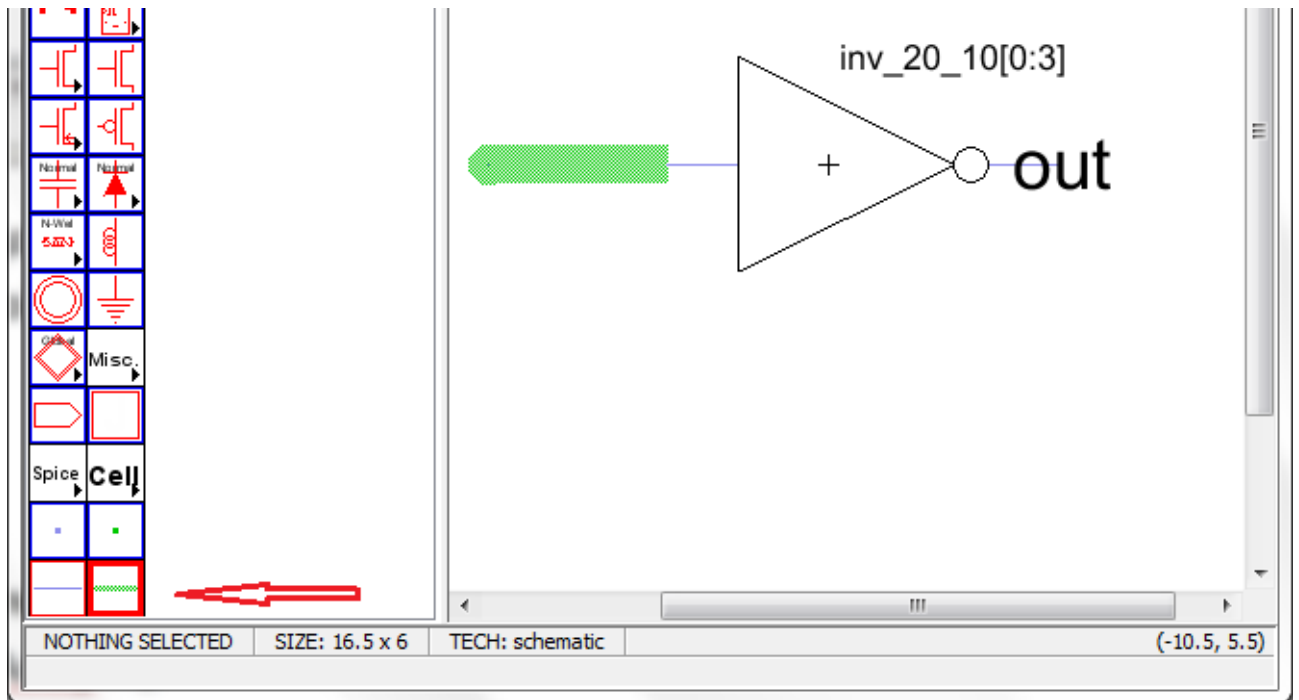


Рис. 5.11. Підключення шини дуг до масиву інверторів

Другий спосіб підключення шини: створіть контакт шини перед виходом інвертора (рис. 5.12).

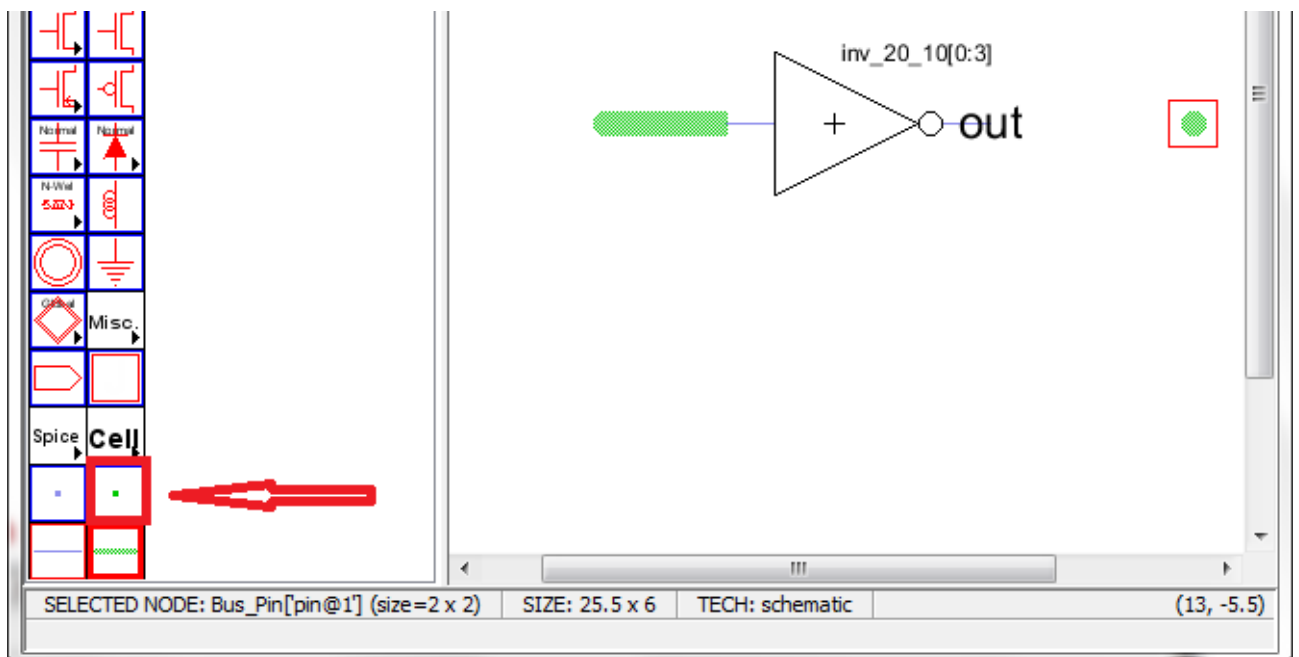


Рис. 5.12. Створення контакту для шини дуг

Виділіть контакт і клацніть правою кнопкою мишки по контакту виходу інвертора. Буде прорисована шина дуг (рис. 5.13).

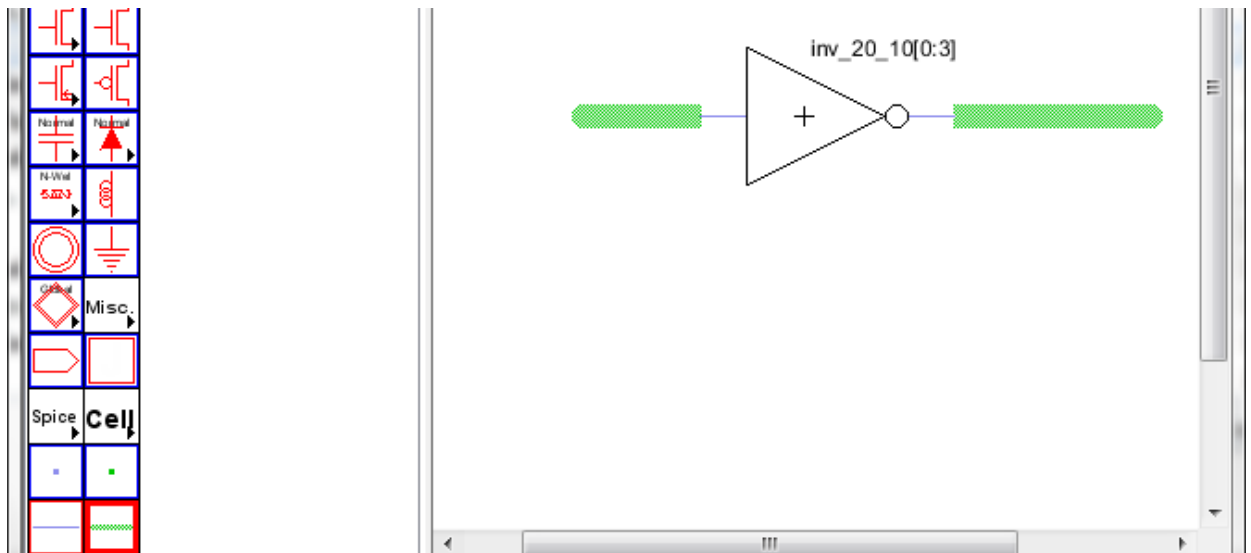


Рис. 5.13. Підключення шини дуг до масиву інверторів

Шину входу інвертора назвемо  $in[0:3]$ . Така назва вказує кількість входів у шині. Тобто, у шині присутні входи:  $in[0]$ ,  $in[1]$ ,  $in[2]$ ,  $in[3]$ .

Вихідну шину назвемо  $out[12:15]$  (рис. 5.14).

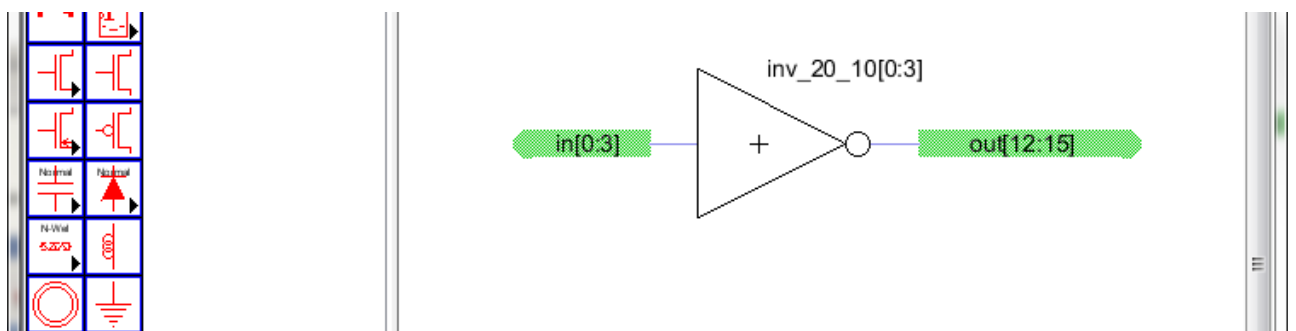


Рис. 5.14. Позначення вхідної та вихідної шин

Перевірте схему на наявність помилок *DRC*.

Інвертор  $inv\_20\_10[0]$  має вхід  $in[0]$  і, відповідно, вихід  $out[12]$ .

Інвертор  $inv\_20\_10[1]$  має вхід  $in[1]$  і вихід  $out[13]$ .

Інвертор  $inv\_20\_10[2]$  має вхід  $in[2]$  і вихід  $out[14]$ .

Інвертор  $inv\_20\_10[3]$  має вхід  $in[3]$  і вихід  $out[15]$ .

Для того щоб підключитися до шини можна використовувати звичайну дугу.

Підключимо до шини входу дугу й назвемо її  $in[0]$  (рис. 5.15).



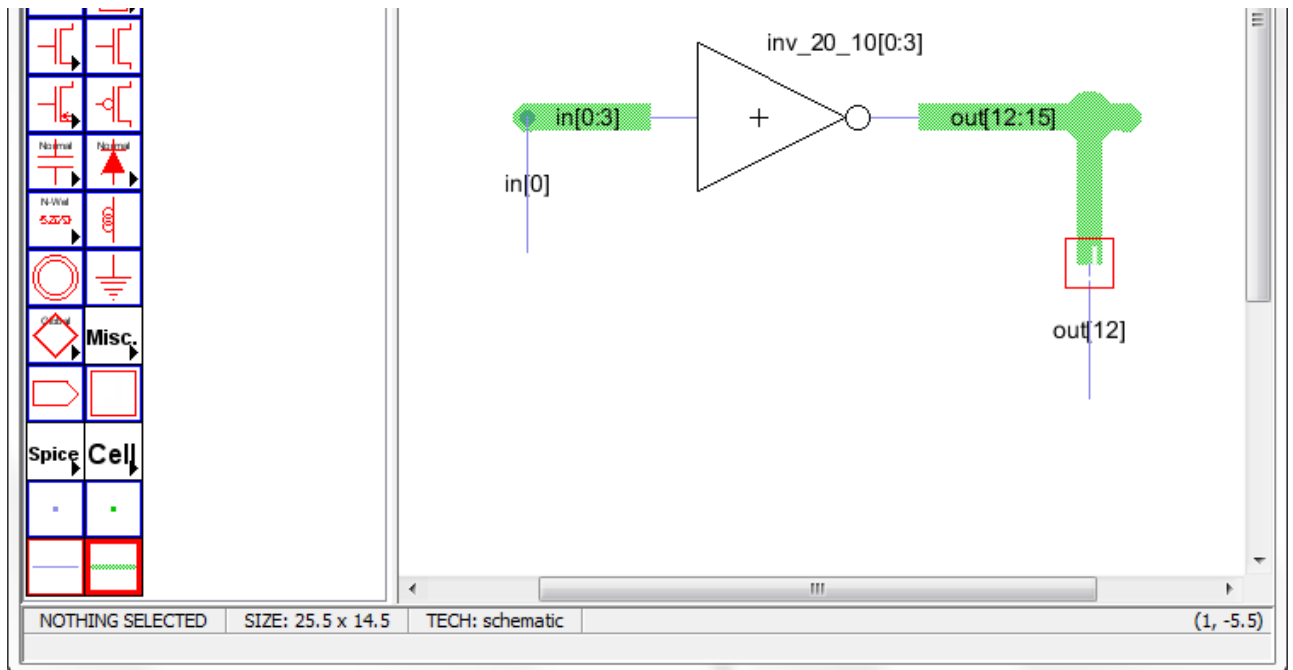


Рис. 5.15. Підключення звичайних дуг до шини

Для приєднання дуги до вихідної шини скористаємося вузлом *Wire\_Con*. Вставте цей вузол в область проектування, виділіть його й клацніть правою кнопкою мишки по шині *out[12:15]*. Результат зображений на рис. 5.16.

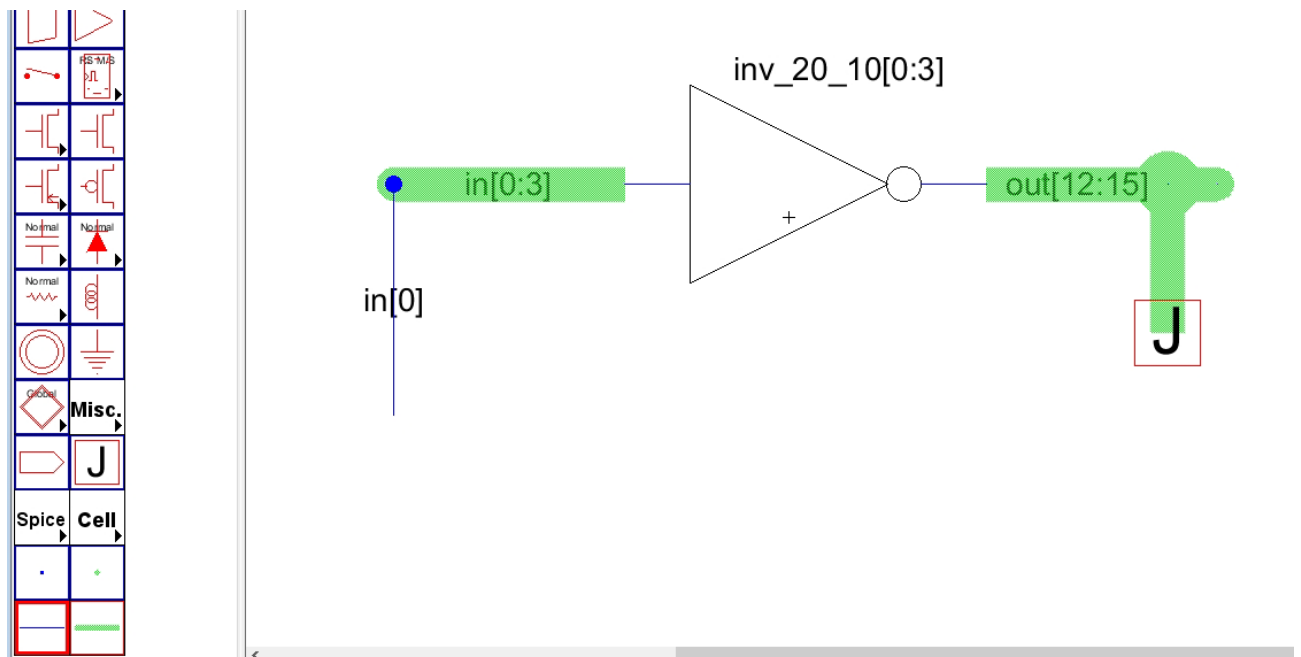


Рис. 5.16. Вставлення вузла *Wire\_Con*

Приєднайте до *Wire\_Con* дугу й назвіть її *out[12]* (рис. 5.17).

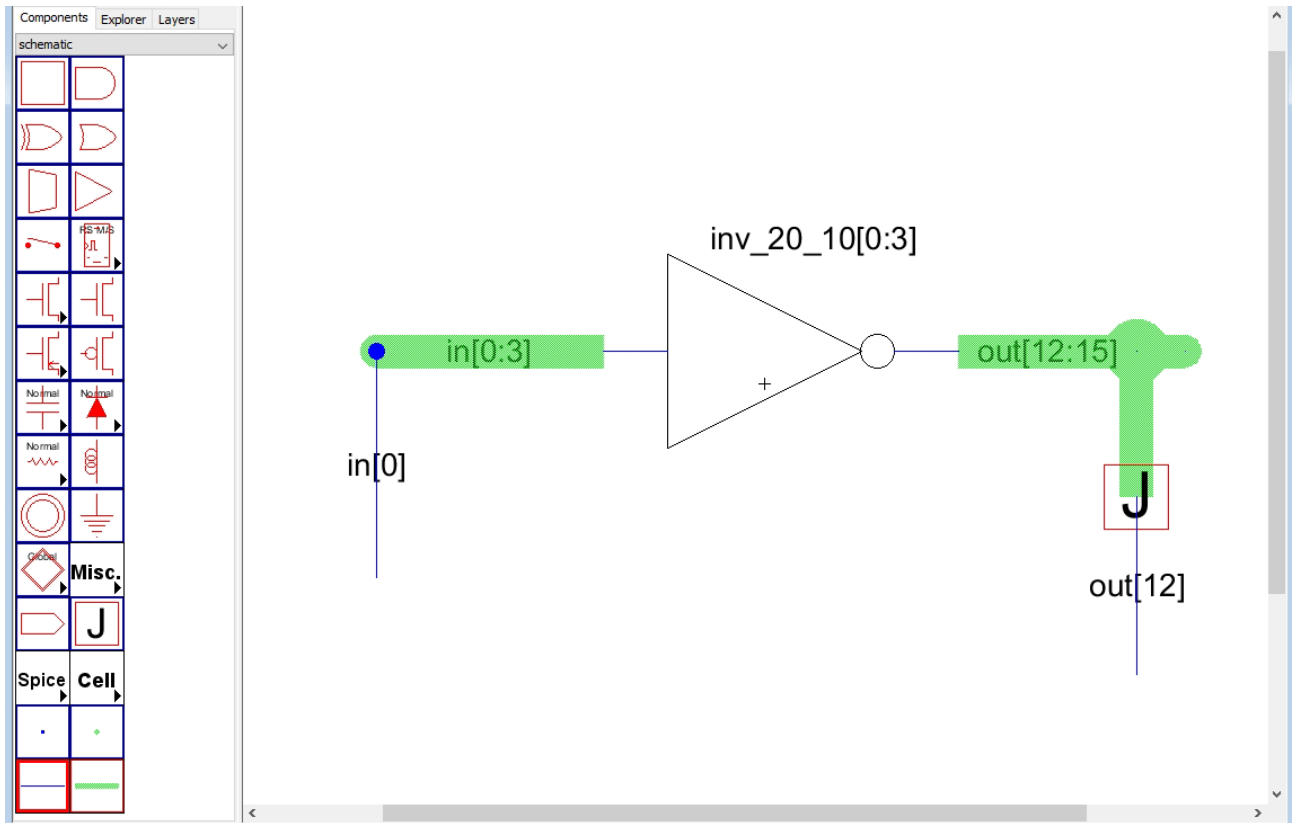


Рис. 5.17. Підключення звичайних дуги до вузла *Wire\_Con*

Тепер почнемо проектування схеми кільцевого генератора.

Створіть комірку *ring\_oscillator{sch}* і помістіть в неї інвертор (рис. 5.18).

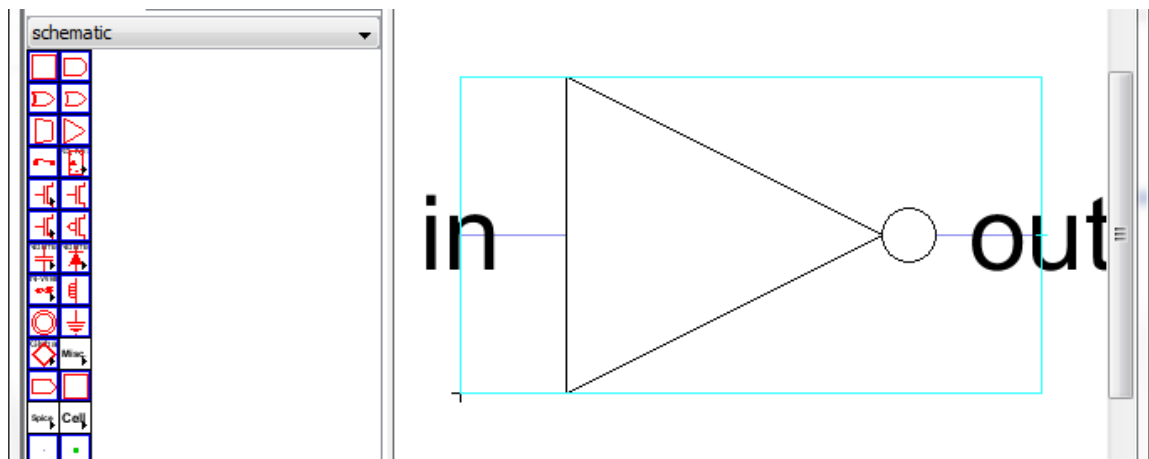


Рис. 5.18. Інвертор в комірці *ring\_oscillator{sch}*

Виділіть інвертор і відкрийте вікно з параметрами створення масиву з виділеного елемента (*F6*). Укажіть кількість повторюваних вузлів по осі  $X = 11$ . Перекриття по осі  $X = -5$  (рис. 5.19).

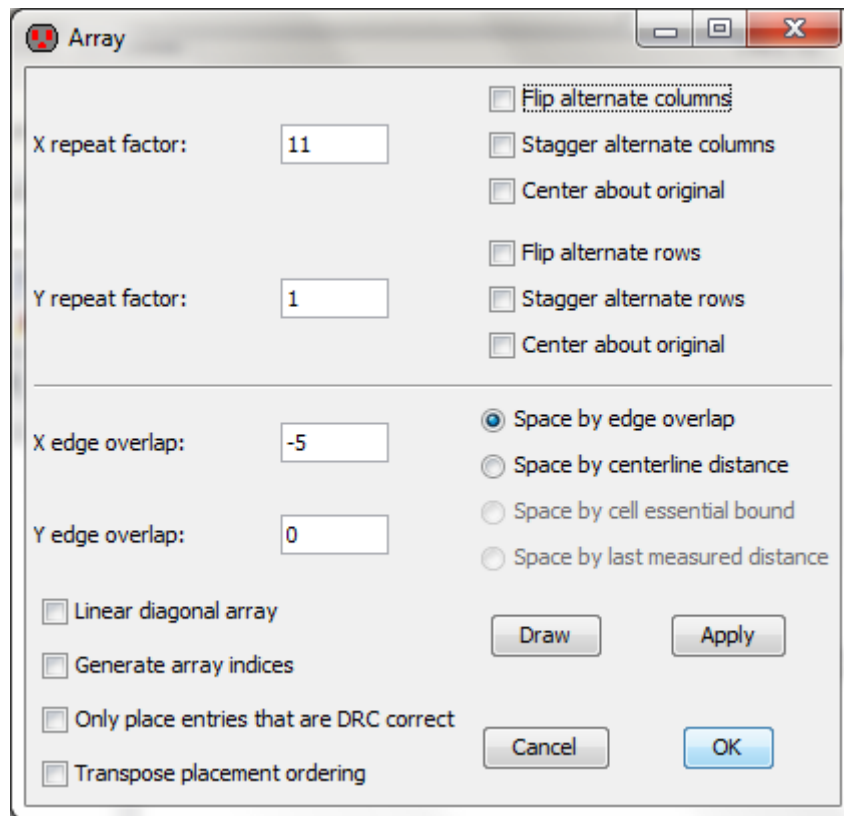


Рис. 5.19. Вставлення параметрів багатократного копіювання

У результаті одержимо масив з 11-ти інверторів, заповнений по горизонталі (рис. 5.20).

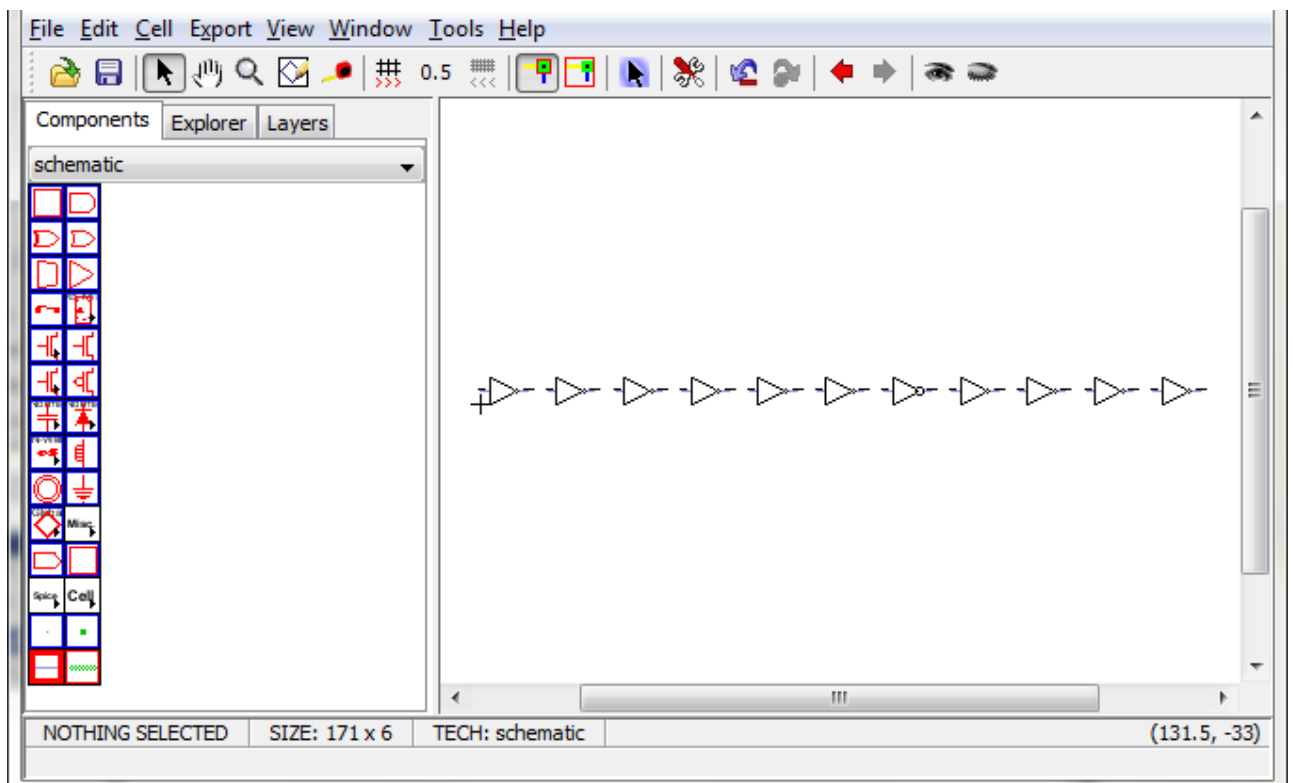


Рис. 5.20. Горизонтальний масив з 11-ти інверторів

Тепер потрібно з'єднати інвертори між собою. Це можна зробити за допомогою мишки або, для економії часу, включити функцію *Enable Mimic-Stitching*. Для цього виділіть один з інверторів і перейдіть до меню *Tools* → *Routing* → *Enable Mimic-Stitching*. Далі як зазвичай з'єднайте два інтерфейси між собою. У результаті, усі вузли з'єдналися між собою (рис. 5.21). Після цього параметр *Enable Mimic-Stitching* потрібно відключити: меню *Tools* → *Routing* → *Enable Mimic-Stitching*.

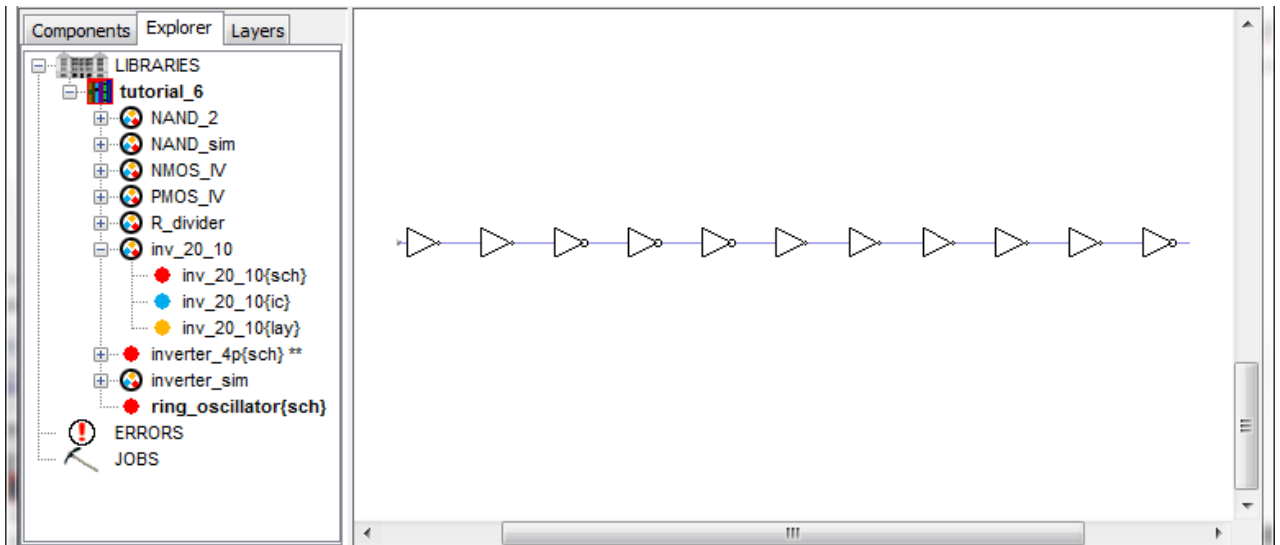


Рис. 5.21. Масив з'єднаних між собою інверторів

Тепер з'єднайте дугою вхід і вихід крайніх інверторів і назвіть її *osc\_out*. Вставте *Spice* код. Розмір *Spice* коду й назви дуги вкажіть рівним 5 (рис. 5.22).

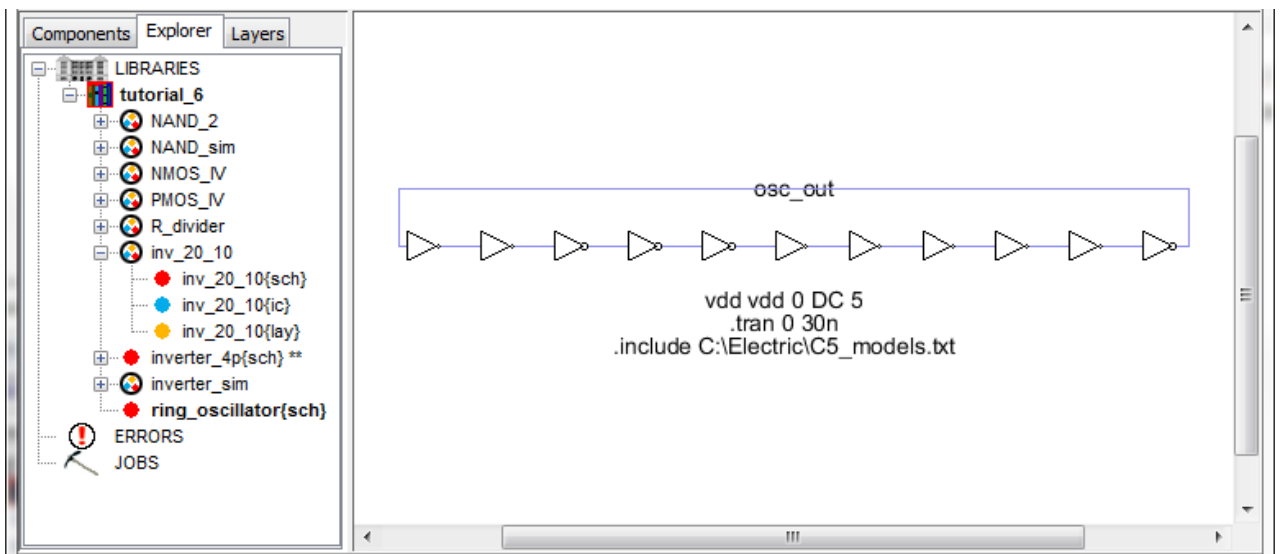


Рис. 5.22. Схема кільцевого генератора із *Spice* кодом

Промодельюємо роботу генератора в *LTspice*. Для відображення оберіть сигнал  $V(osc\_out)$  (рис. 5.23).

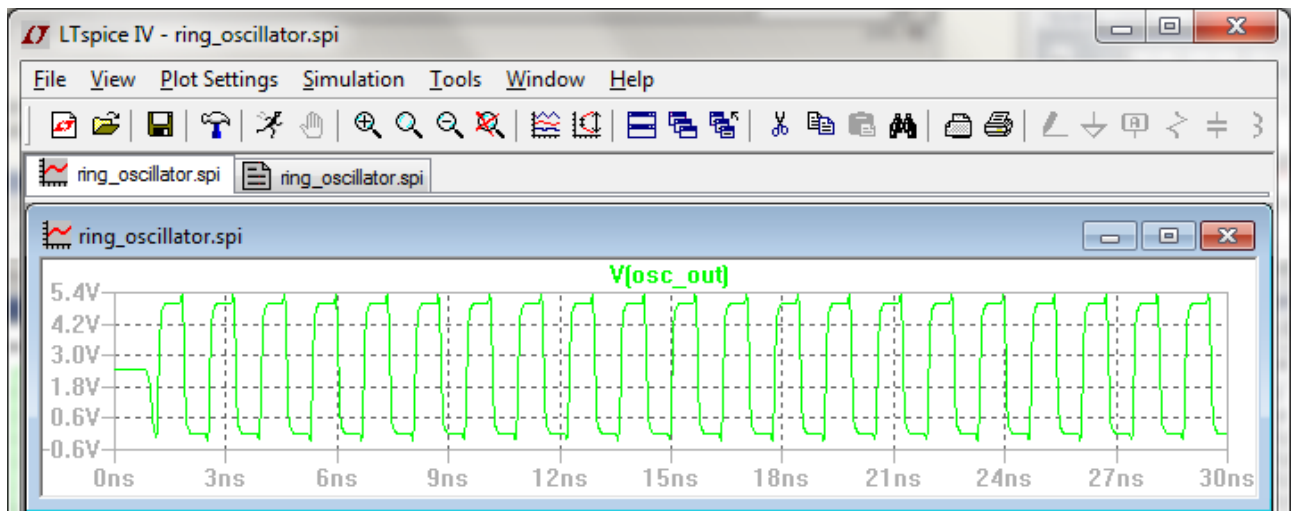


Рис. 5.23. Часова діаграма роботи кільцевого генератора в *LTspice*

Якщо характеристика не така, як на рис. 5.23, то спробуйте переписати *Spice* код, виправивши рядок `.tran 0 30n` на `.tran 0 3n UIC` (*UIC* – *use initial conditions*). Ви зменшили часовий інтервал і вказали *LTspice*, що на всіх вузлах на початку роботи напруга рівна 0.

Перейдемо в зонд *Electric* (рис. 5.24).

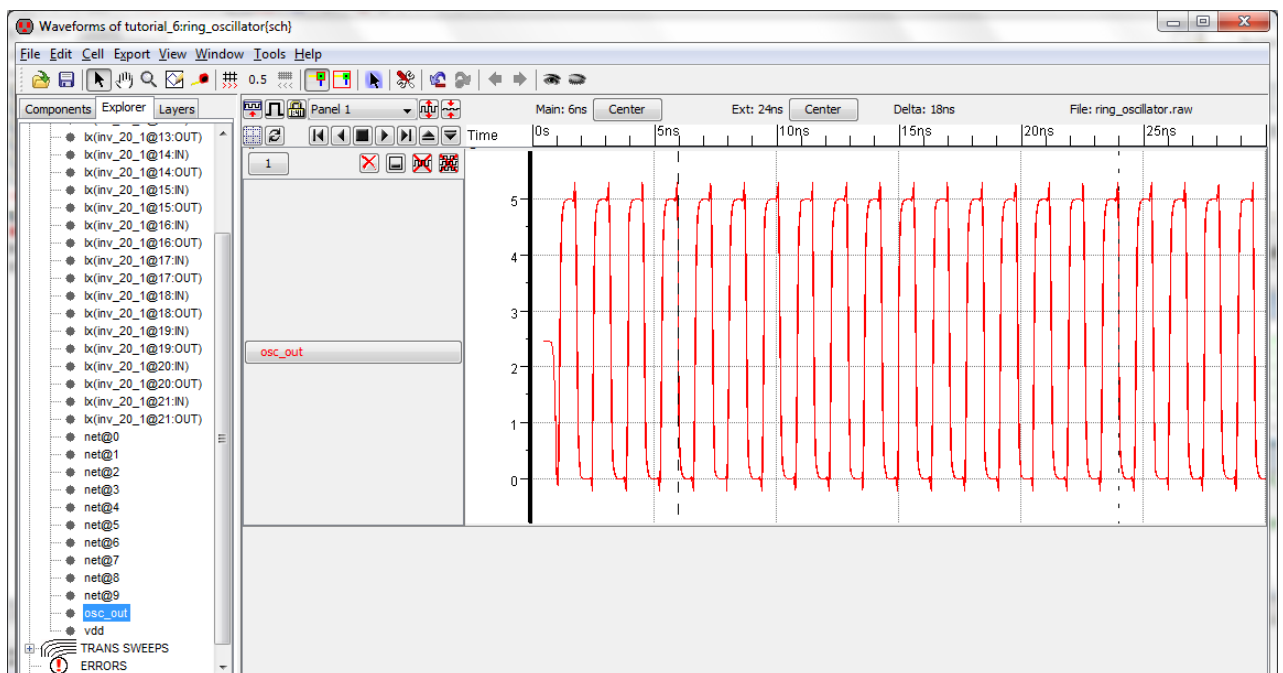


Рис. 5.24. Часова діаграма роботи кільцевого генератора в *Electric*

Після одержання характеристик з *raw* файлів закрийте зонд *Electric* і створіть нову версію комірки *ring\_oscillator{sch}*. Змініть в ній розмір *Spice* коду до значення 2 і видаліть всі інвертори, крім одного (рис. 5.25).

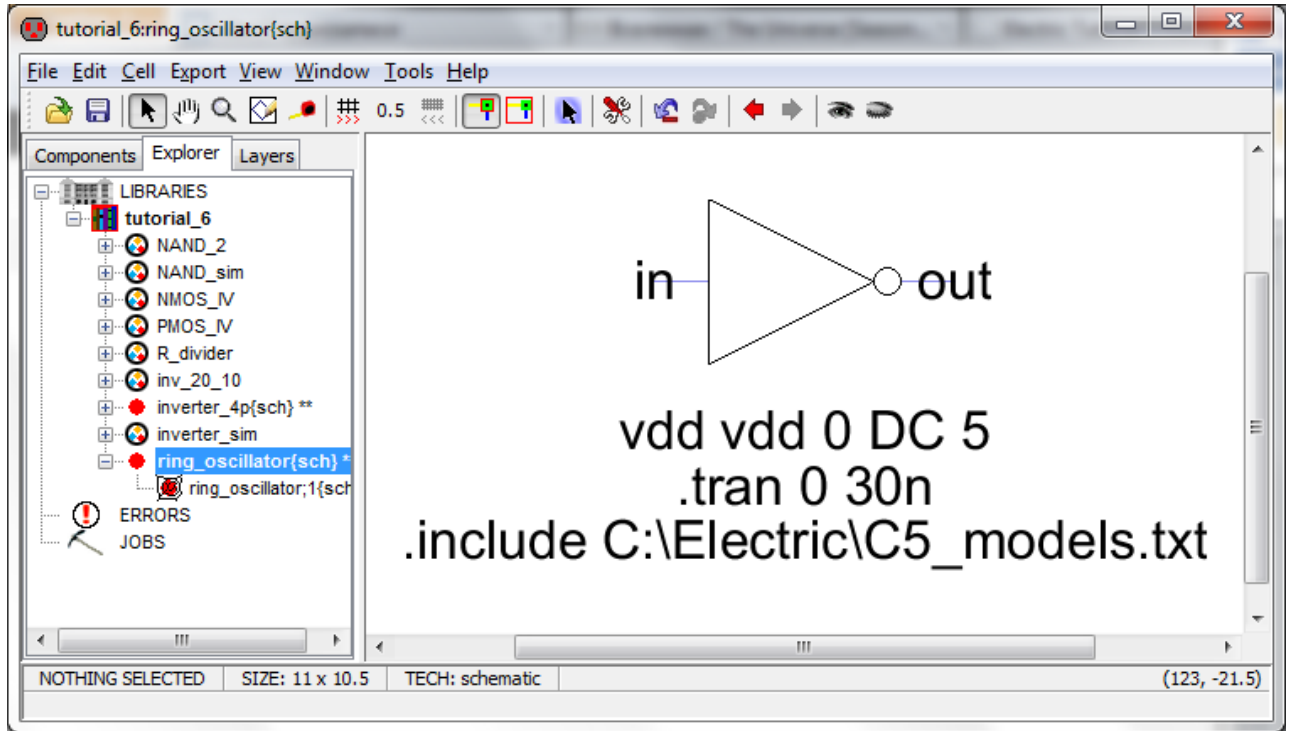


Рис. 5.25. Інвертор зі *Spice* кодом

Назвіть інвертор *inverter[0:10]* і перемістіть ім'я вище самого інвертора. Додайте шини на вхід і на вихід, укажіть їхні імена, як показано на рис. 5.26.

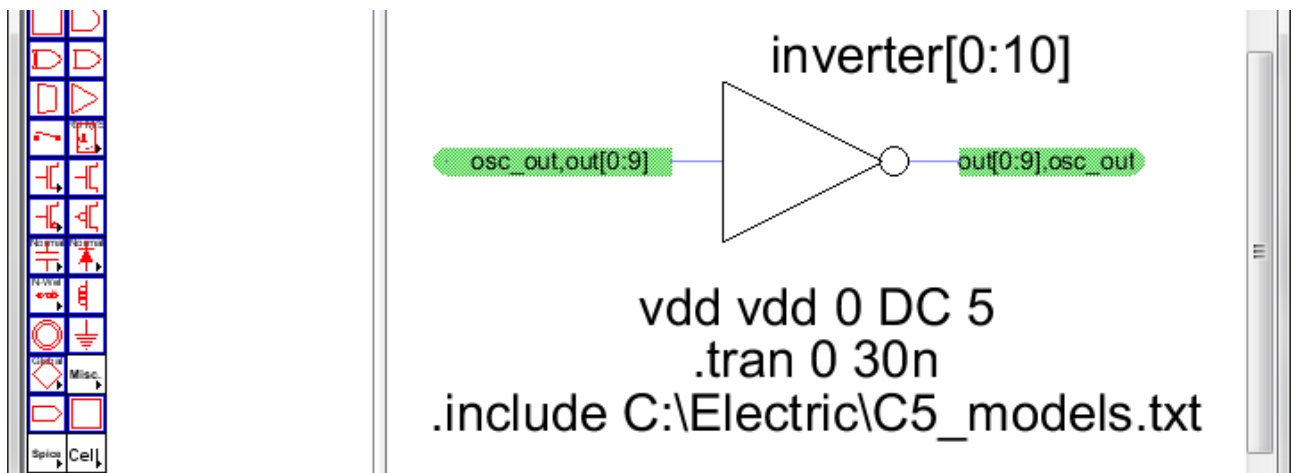


Рис. 5.26. Підключення шин дуг до інвертора

Інвертор *inverter[0]* має вхід *osc\_out* і вихід *out[0]*.

Інвертор *inverter[1]* має вхід *out[0]* і вихід *out[1]*.

Інвертор *inverter[2]* має вхід *out[1]* і вихід *out[2]*.

.....

Інвертор *inverter[10]* має вхід *out[9]* і вихід *osc\_out*.

Промоделюйте цю схему. Результат має бути ідентичним.

Наступним кроком буде моделювання топології.

Створіть нову комірку *ring\_oscillator{lay}*. У ній розмістіть комірку *inv\_20\_10{lay}*. Створіть горизонтальний масив з 11 комірок *inv\_20\_10{lay}* і перекриттям по осі *X* рівним 0 (рис. 5.27).

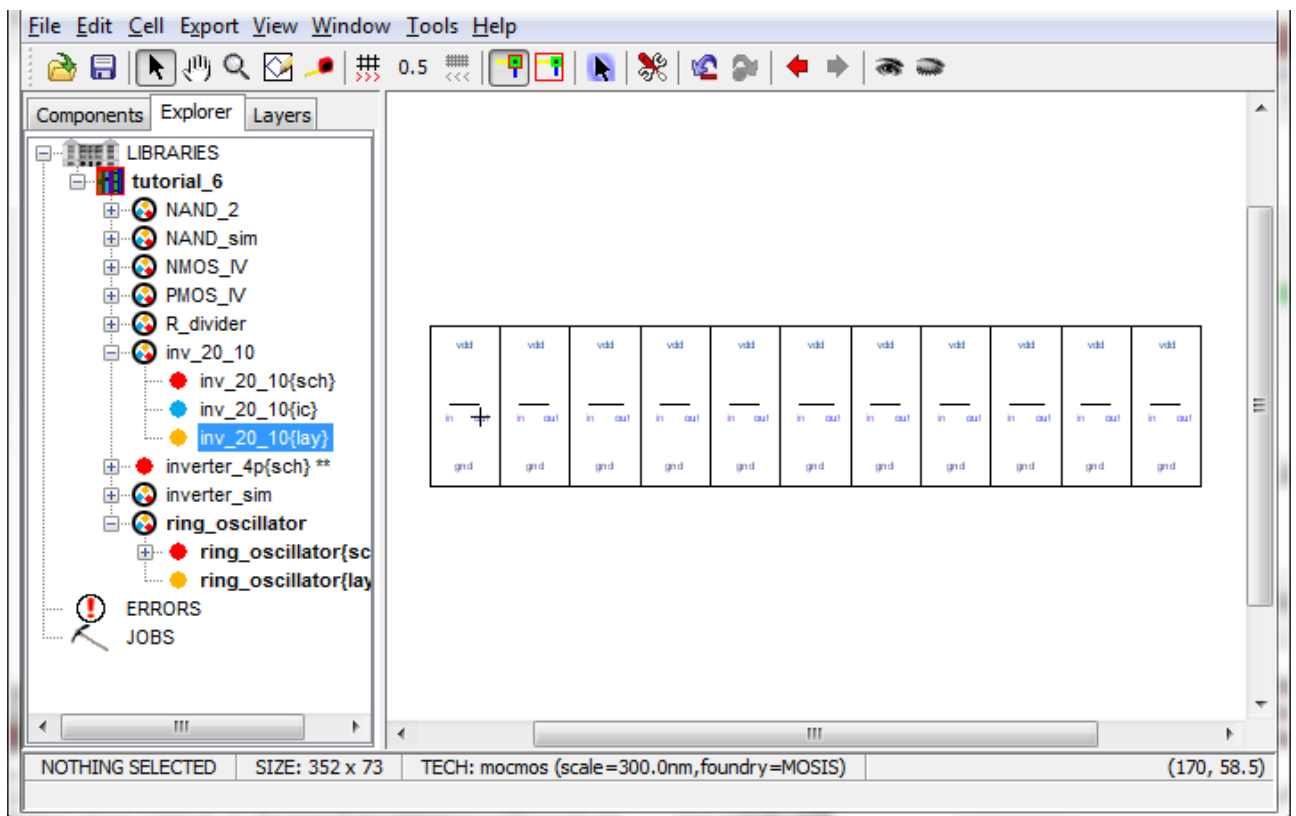


Рис. 5.27. Масив з 11 комірок *inv\_20\_10{lay}*

Щоб побачити, що собою представляє трансльований нами макет, скористайтеся зміною режиму відображення (закрите/відкрите око) (рис. 5.28).

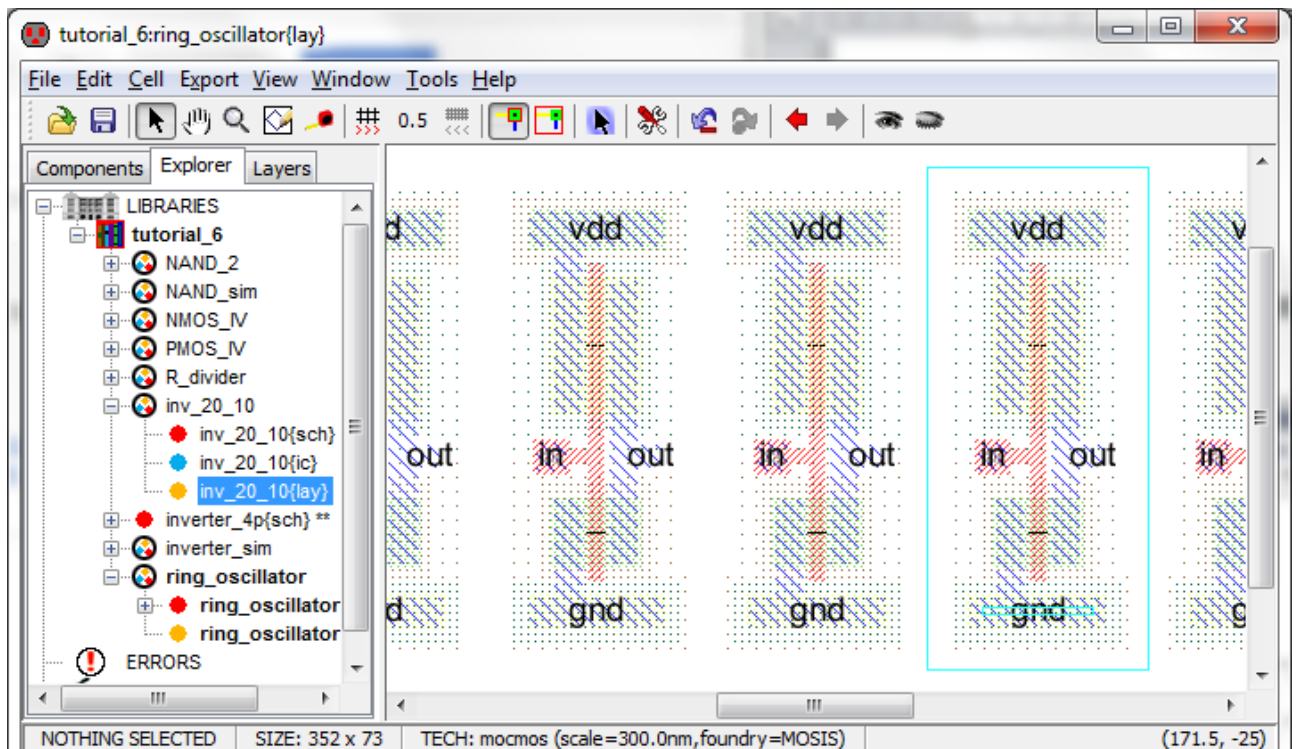


Рис. 5.28. Масив з 11 комірок у режимі відображення «відкрите око»

Перевірте схему на наявність помилок *DRC*. Якщо помилки є, зробіть перекриття по осі *X* рівним мінус 6.

Далі з'єднайте між собою суміжні входи *vdd/gnd* та *in/out* (рис. 5.29).

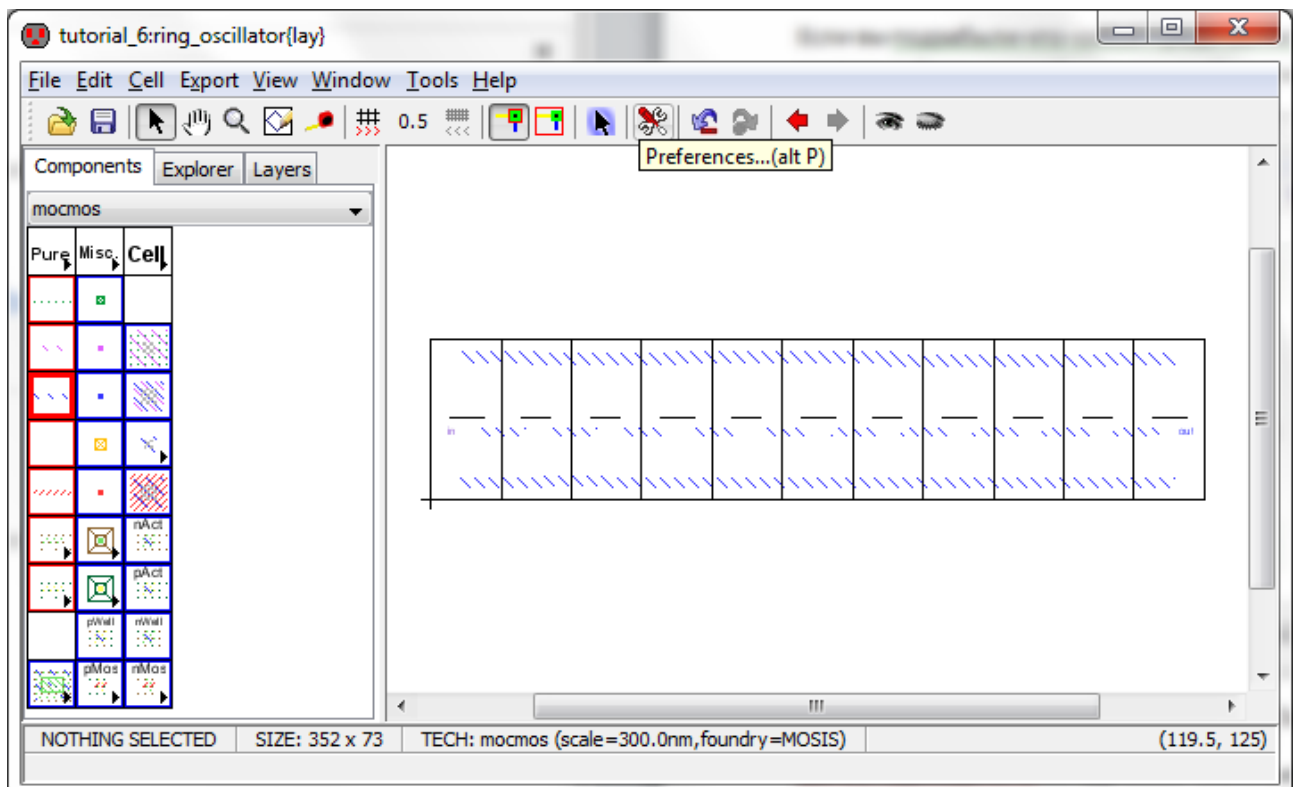


Рис. 5.29. Масив з 11 комірок зі з'єднаними суміжними входами



Додайте контактні області *metall-metal2* на краях генератора. З'єднайте дугою *metall* вузол *metall-metal2* зліва із входом, зправа з виходом (рис. 5.30, 5.31).

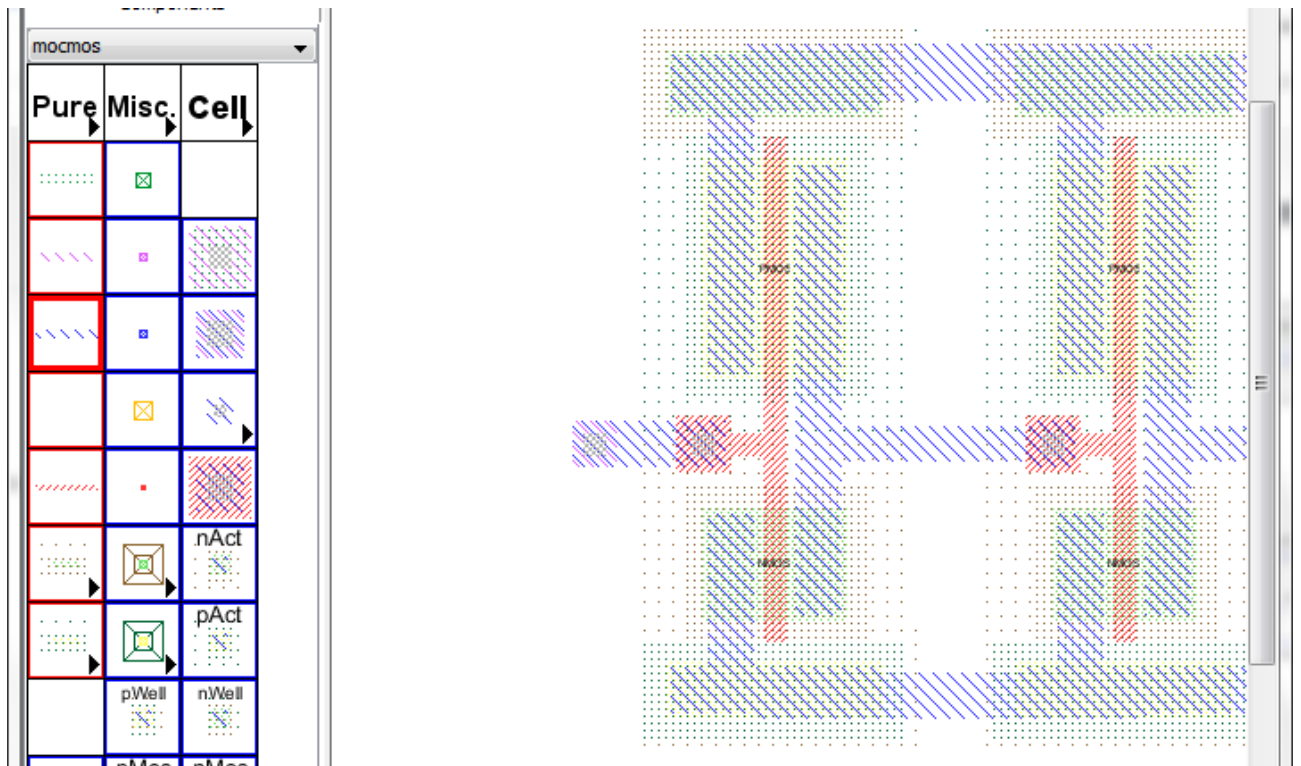


Рис. 5.30. Частина топології кільцевого генератора (вхід схеми)

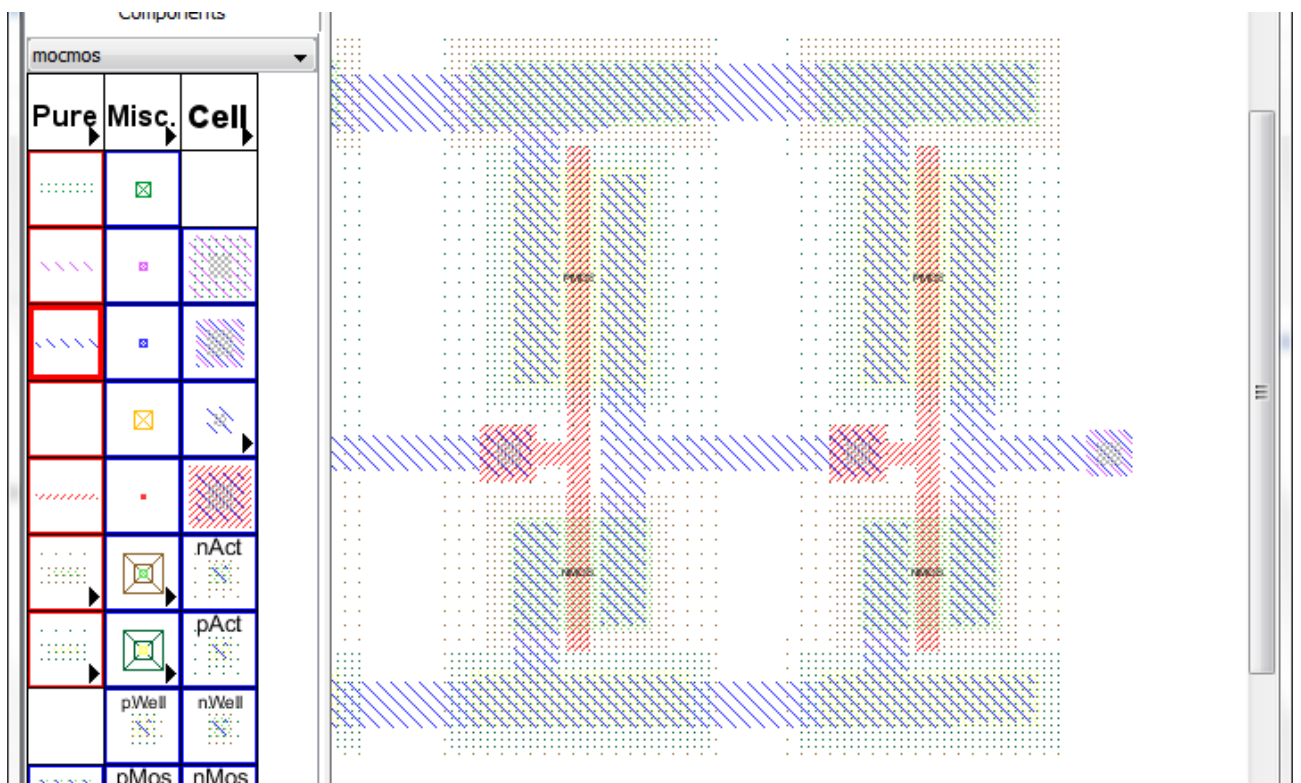


Рис. 5.31. Частина топології кільцевого генератора (вихід схеми)

З'єднайте вузли *metal1-metal2* між собою дугою *metal2* (рис. 5.32).

Будьте уважні! Перед тем як з'єднати вузли *metal1-metal2* між собою, оберіть дугу *metal2*, інакше вони будуть з'єднані дугою *metal1*.

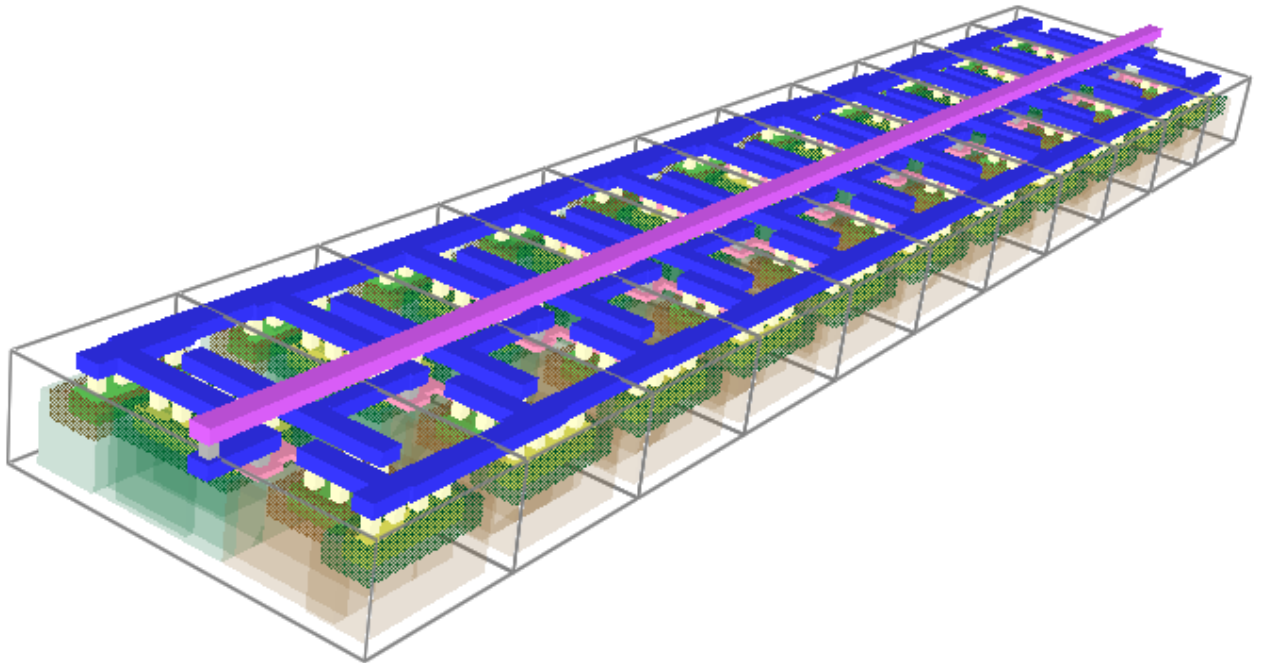


Рис. 5.32. 3D зображення кільцевого генератора після з'єднання входу та виходу

Промаркуйте дугу *metal2*, назвавши її *osc\_out*, розмір тексту – 10.

Перевірте на помилки *DRC*.

Виділіть верхній порт крайнього правого інвертора та експортуйте його на живлення (*vdd*). Виділіть нижній порт крайнього правого інвертора й експортуйте його на «землю» (*gnd*) (рис. 5.33).

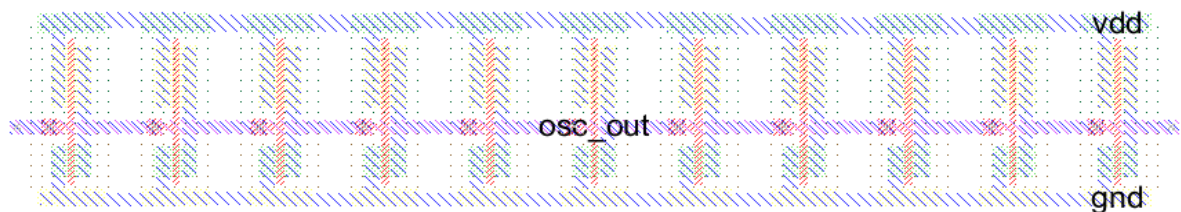


Рис. 5.33. Топологія кільцевого генератора із експортованими портами

Скопіюйте із комірки *ring\_oscillator{sch}* *Spice* код і вставте його у комірку *ring\_oscillator{lay}*.

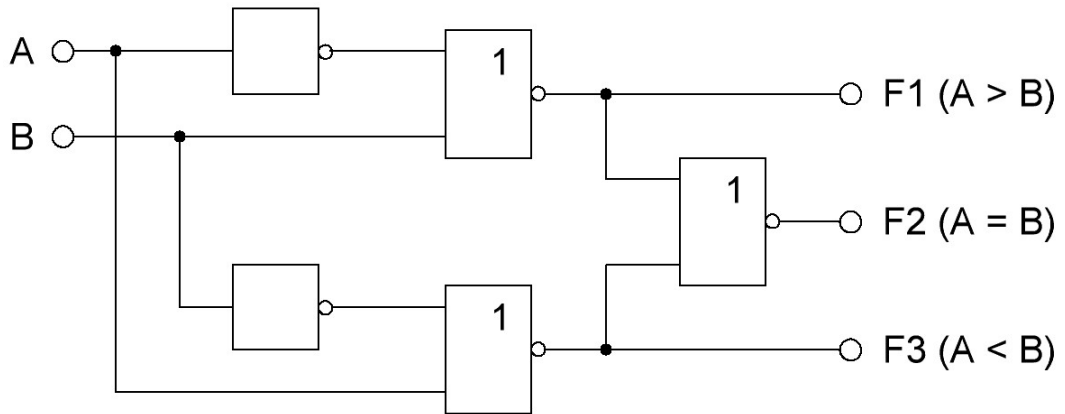
Перевірте на наявність помилок (*DRC*, *NCC*, *ERC*). Їх не має бути.

## Варіанти завдань

Усі схеми проектувати за технологією КМОП.

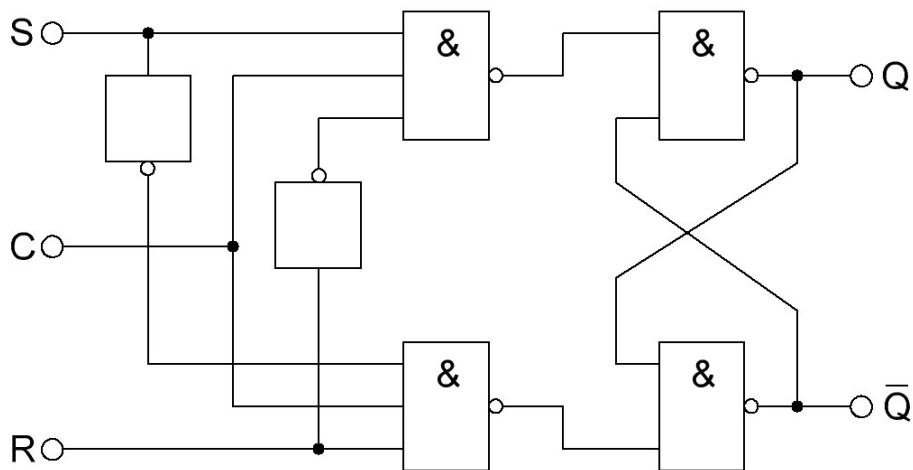
### Варіант 1

Проектування однорозрядного цифрового компаратора



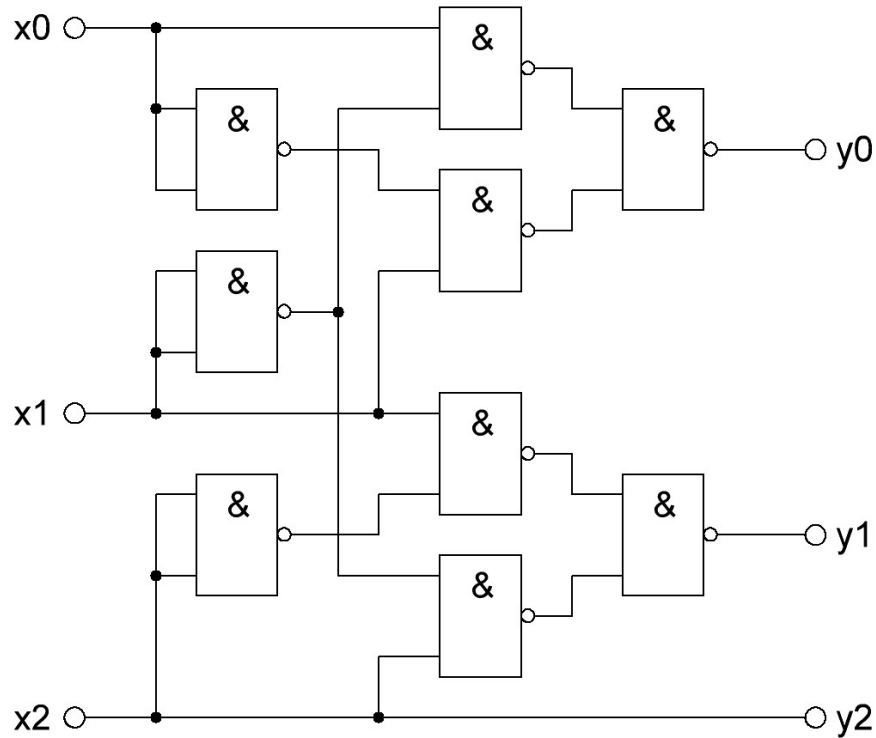
### Варіант 2

Проектування E-тригера



### Варіант 3

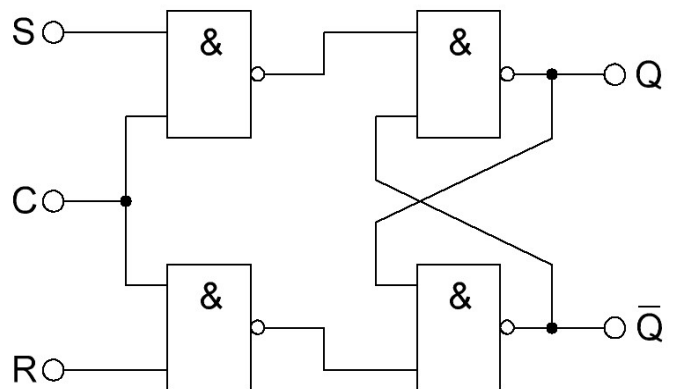
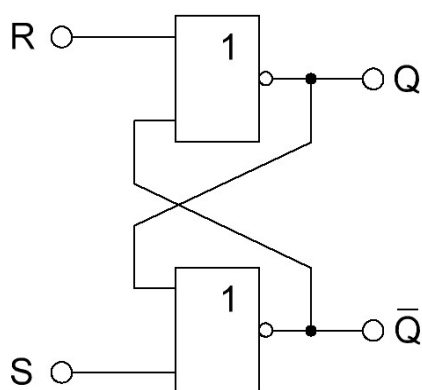
Проектування перетворювача двійкового коду 421 у 3-розрядний код Грея



Зверніть увагу на наявність прямого з'єднання між *Off-page* вузлами «x2» і «y2». *Ltspice* не може моделювати «порожню» схему (тобто схему без перетворення сигналу), тому після моделювання в списку доступних сигналів «y2» буде відсутній. Поставте перед *Off-page* вузлом «y2» резистор номіналом 100 Ом. Після цього сигнал «y2» з'явиться у списку доступних сигналів.

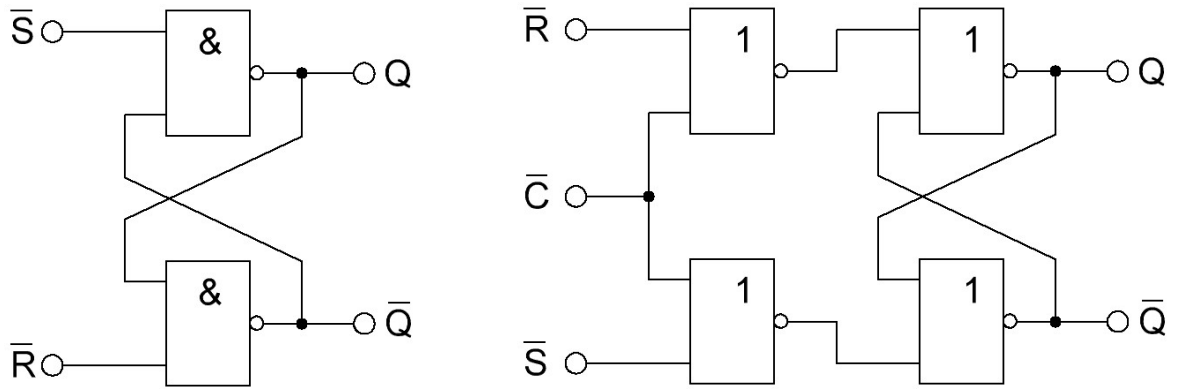
### Варіант 4

Проектування асинхронного *RS*-тригера на логічних елементах АБО-НІ та синхронного *RS*-тригера на логічних елементах І-НІ



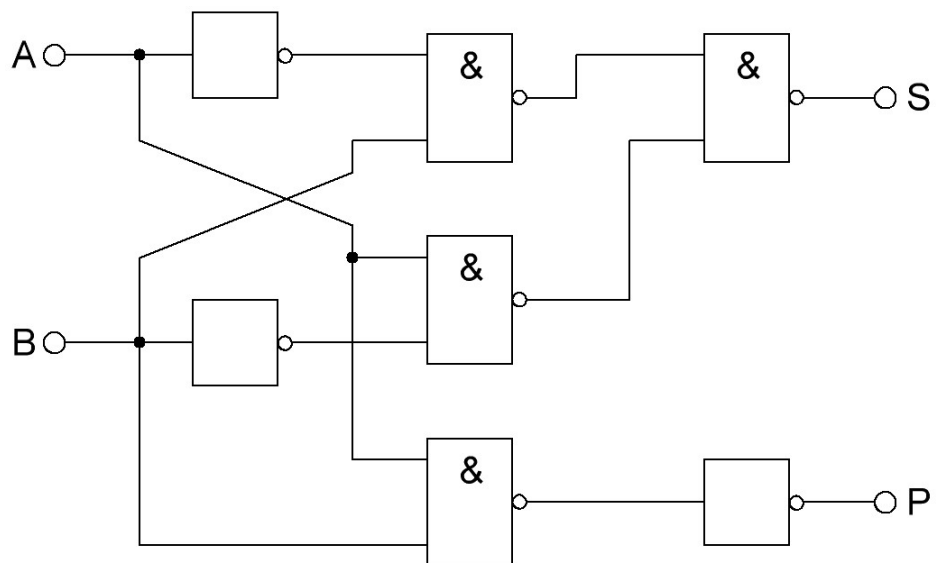
### Варіант 5

Проектування асинхронного  $RS$ -тригера на логічних елементах І-НІ та синхронного  $RS$ -тригера на логічних елементах АБО-НІ



### Варіант 6

Проектування напівсуматора



## Контрольні питання

1. Які логічні схеми називають комбінаційними?
2. Які логічні схеми називають послідовними?
3. Які послідовні схеми називають синхронними?
4. Чим визначається наступний стан послідовної синхронної схеми?
5. У чому різниця між регістрами, які активізуються додатним і від'ємним фронтами?
6. Якими часовими параметрами характеризуються регістри?
7. Що називають часом встановлення синхронного регістра?
8. Що називають часом утримання синхронного регістра?
9. Що називають затримкою поширення синхронного регістра?
10. Опишіть принцип роботи статичної пам'яті.
11. Опишіть принцип роботи динамічної пам'яті.
12. Що таке заціпка?
13. Що називають прозорим режимом заціпки?
14. Що називають режимом утримання заціпки?
15. Яку заціпку називають додатньою?
16. Яку заціпку називають від'ємною?
17. Опишіть принцип роботи регістрів.
18. Що таке розфазування тактових імпульсів?
19. Як можна уникнути збоїв, викликаних розфазуванням тактових імпульсів?

## Практичне заняття №6

### Розміщення макетів схем на площинці для фабричного виробництва

Розмістимо створені у практичних роботах №№ 3, 4 і 5 схеми на площинці для фабричного виробництва.

Збережіть бібліотеку *tutorial\_5.jelib* як *tutorial\_6.jelib* у папку *C:\Electric*. Створіть комірку *pad{lay}*. У вкладці *Components* оберіть набір *artwork* і в область проєктування вставте елемент *box* (рис. 6.1).

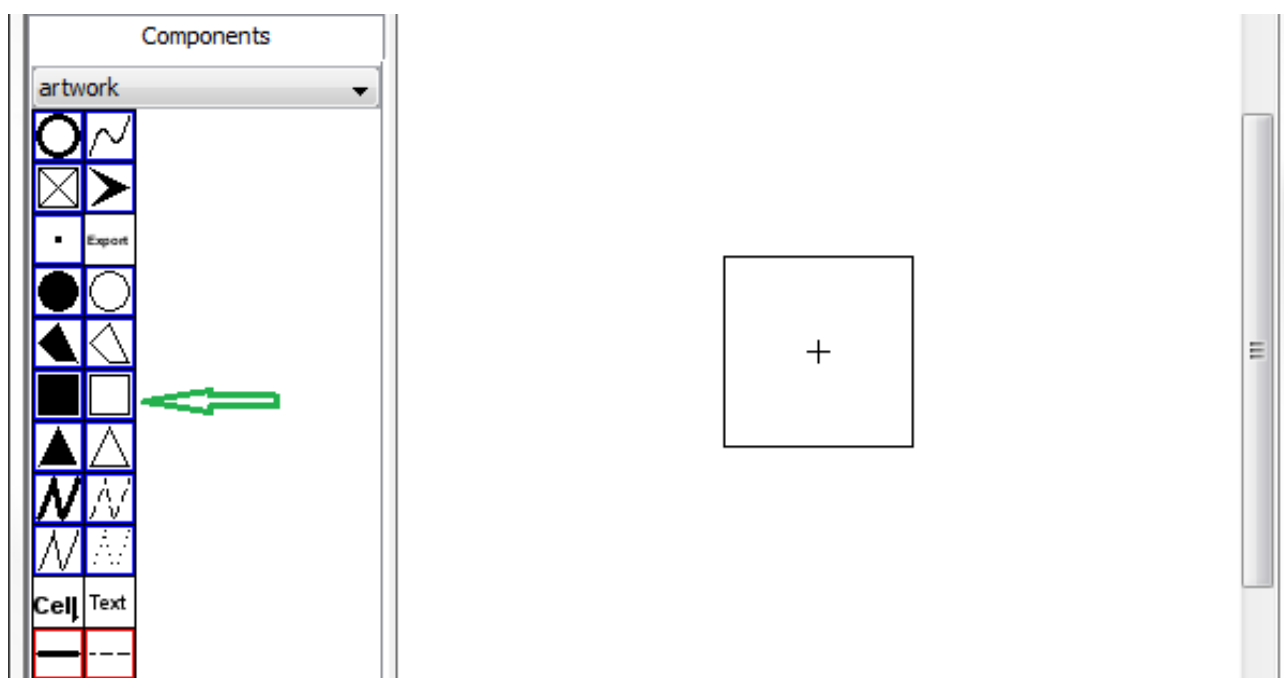


Рис. 6.1. Елемент *box* в області проєктування

Як і в попередніх практичних роботах, будемо використовувати налаштування під техпроцес *C5* виробництва через *MOSIS*. Також у цьому процесі використовуються правила субмікронного дизайну *MOSIS* масштабованих *CMOS* (*SCMOS*) мікросхем.

Чип виготовлений на основі техпроцеса *C5* за допомогою програми *MOSIS Educational Program (MEP)* і має розміри 1,5 мм на 1,5 мм.

Необхідно визначити розміри площинки у комірці *pad{lay}*.

Проектна норма (масштабний коефіцієнт – лямбда) використовувана в установках для техпроцеса C5 становить 300 нм. Розмір макета 5000 на 5000 відповідає 1500 мкм на 1500 мкм. Якщо на стороні чипа є 12 кліток (10 контактних площинок плюс 2 кути), то кожна клітка повинна мати сторону  $5000 / 12 = 416,66$ .

Зменшимо сторону клітки до 400, чип вийде трохи менше максимального розміру. Змініть розміри фігури *box*:  $x = 400$ ,  $y = 400$  (рис. 6.2).

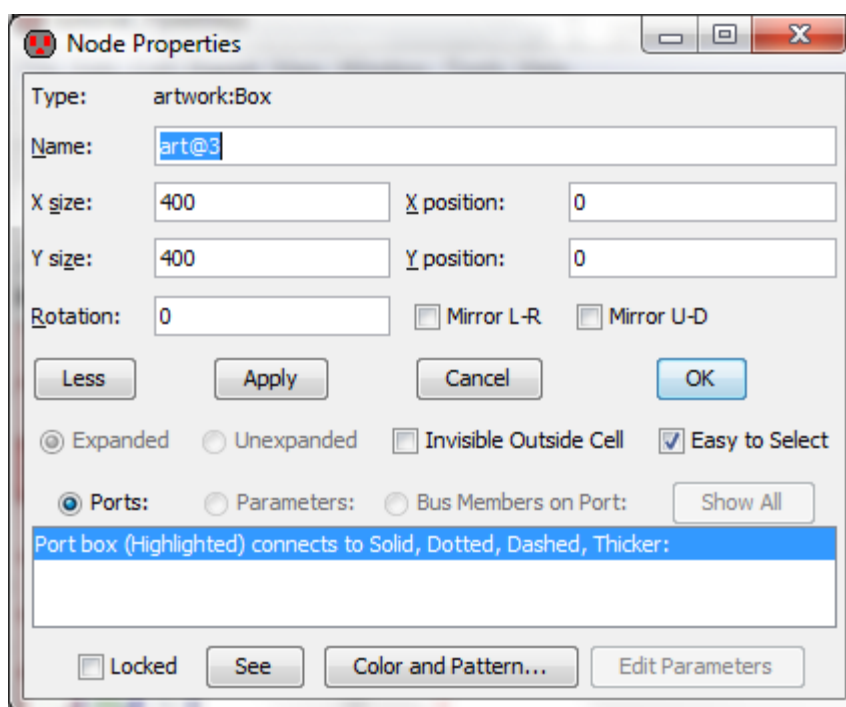


Рис. 6.2. Властивості вузла *box*

Перед тим як додати щось на макет, створіть контактну площинку.

Створіть комірку *padframe{lay}*. Вставте в неї комірку *pad{lay}* (рис. 6.3).



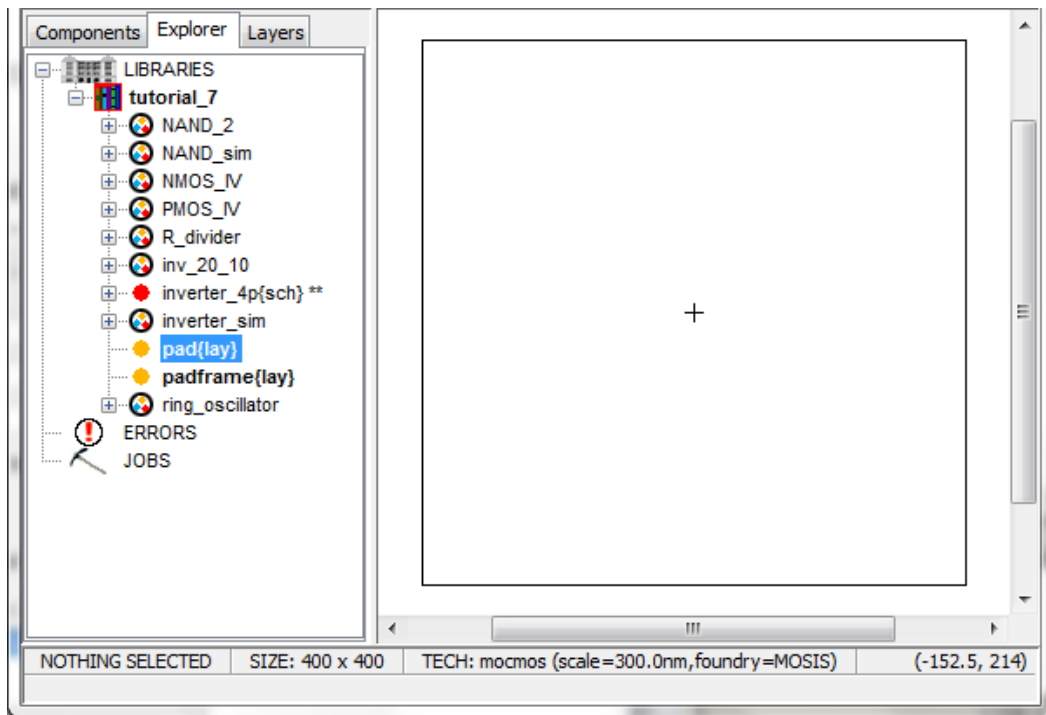


Рис. 6.3. Комірка  $pad\{lay\}$  в комірці  $padframe\{lay\}$

Скористайтеся командою *Array* для створення з комірки  $pad\{lay\}$  у комірці  $padframe\{lay\}$  масиву 12x12 (*Edit* → *Array* або *F6*) (рис. 6.4, 6.5).

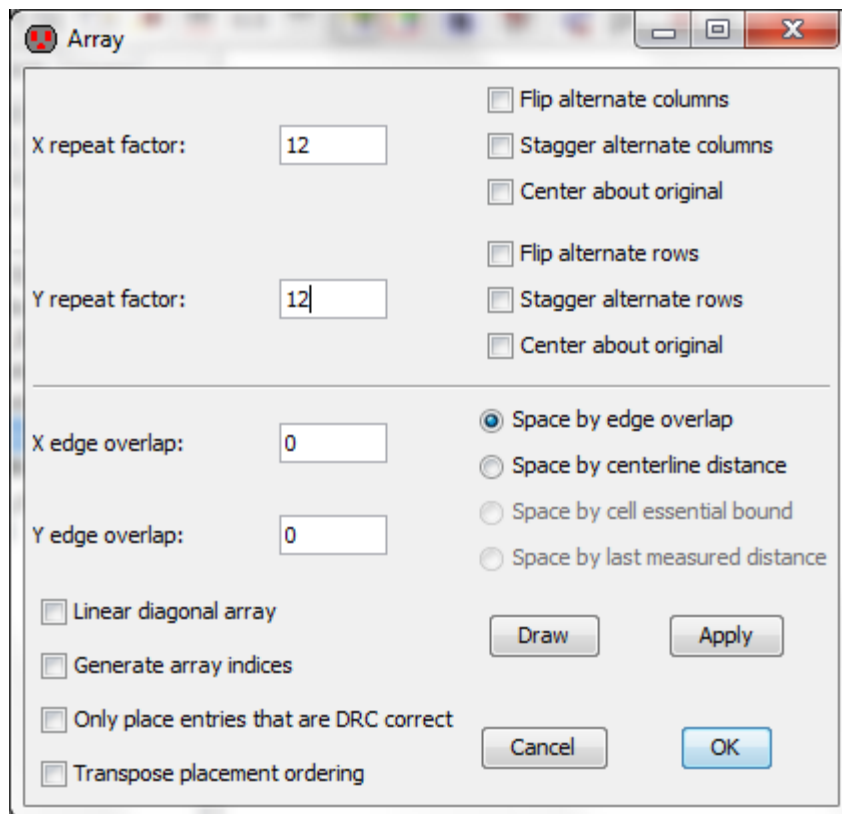


Рис. 6.4. Створення масиву комірок  $pad\{lay\}$

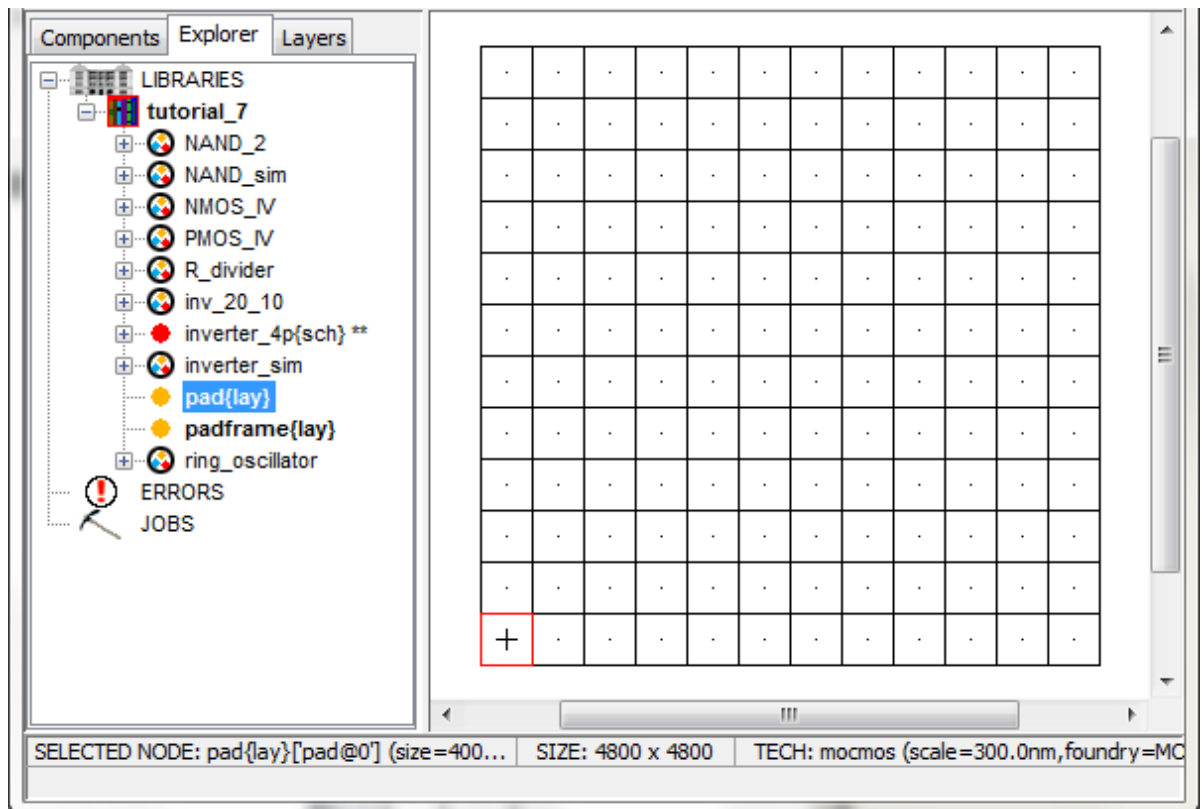


Рис. 6.5. Массив комірок  $pad\{lay\}$

Видалить кутові й внутрішні комірки масиву (рис. 6.6).

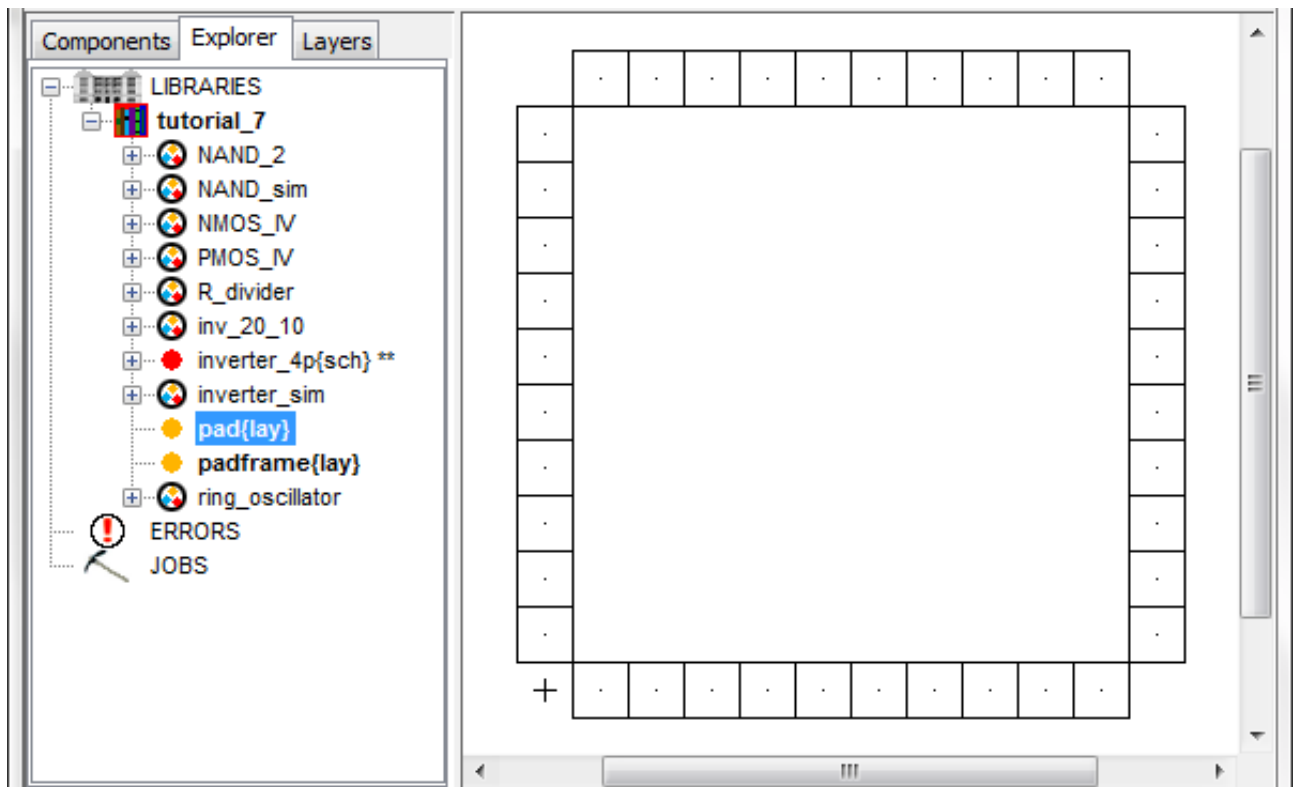



Рис. 6.6. Массив комірок  $pad\{lay\}$  після видалення кутових й внутрішніх комірок

Визначте розміри отриманої площинки. Використовуйте інструмент у меню *Window* → *Measurements* → *Toggle Measurement Mode* (або натисніть кнопку  на панелі інструментів).

Курсор відобразиться у вигляді маркера. Виміряйте ширину площинки, клацнувши мишкою по протилежних сторонах крайніх прямокутників. На екрані відобразиться вимірювана відстань (рис. 6.7). Також, розмір відображається у нижній частині вікна.

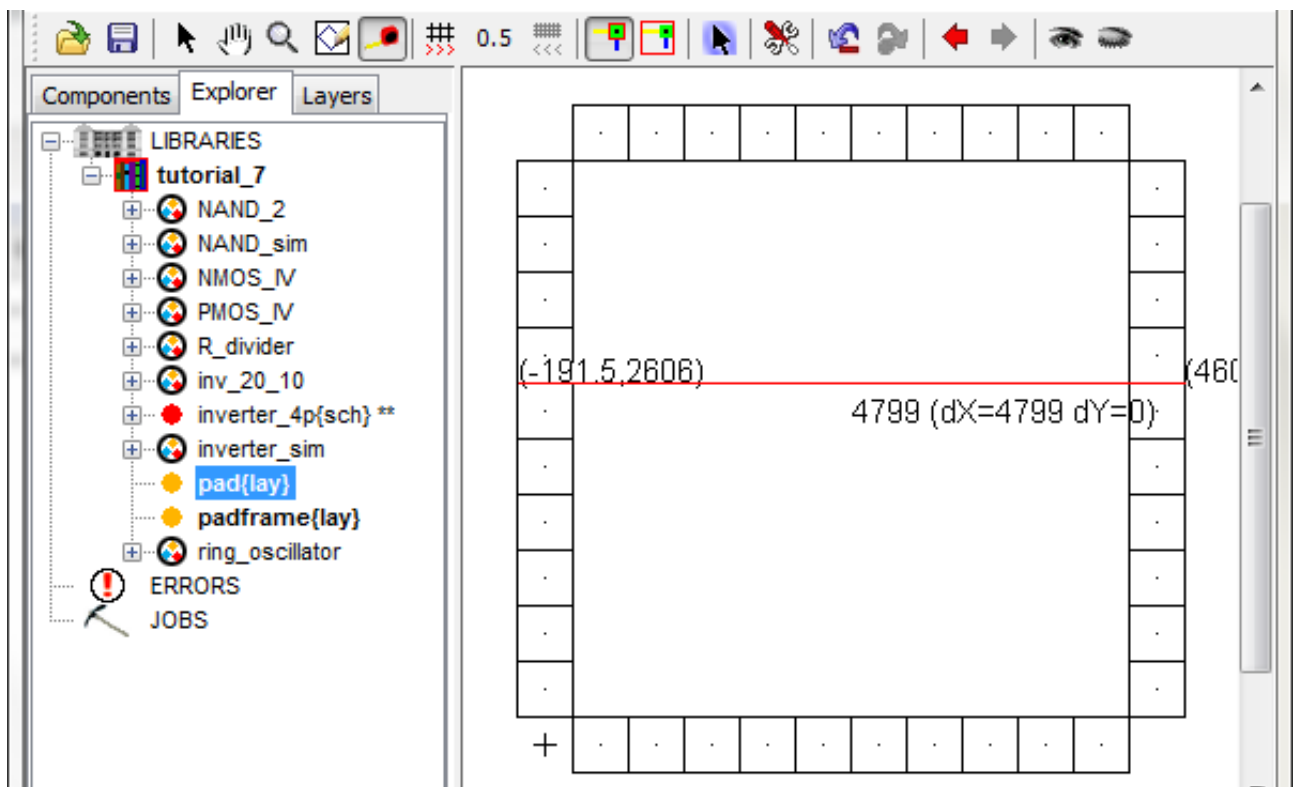


Рис. 6.7. Вимірювання ширини площинки

Для того, щоб очистити екран від результатів вимірювань, натисніть клавішу *Esc*. Щоб відключити режим вимірювань, натисніть відповідну кнопку вимикання маркера вимірювань (*Window* → *Measurements* → *Clear Measurements* або клавіша «*M*»).

Поверніться до комірки *pad{lay}* (виділіть будь-яку клітку й натисніть *Ctrl+D*) і покрийте її металом 75 x 75 мкм. Оскільки проектна норма (лямбда) становить 300 нм, то 75 x 75 мкм буде відповідати 250 x 250 лямбда.

Додайте контактний вузол *metal2-metal3* на площинку як показано на рис. 6.8, і у властивостях укажіть розміри 250 x 250 і положення  $x = 0, y = 0$ .

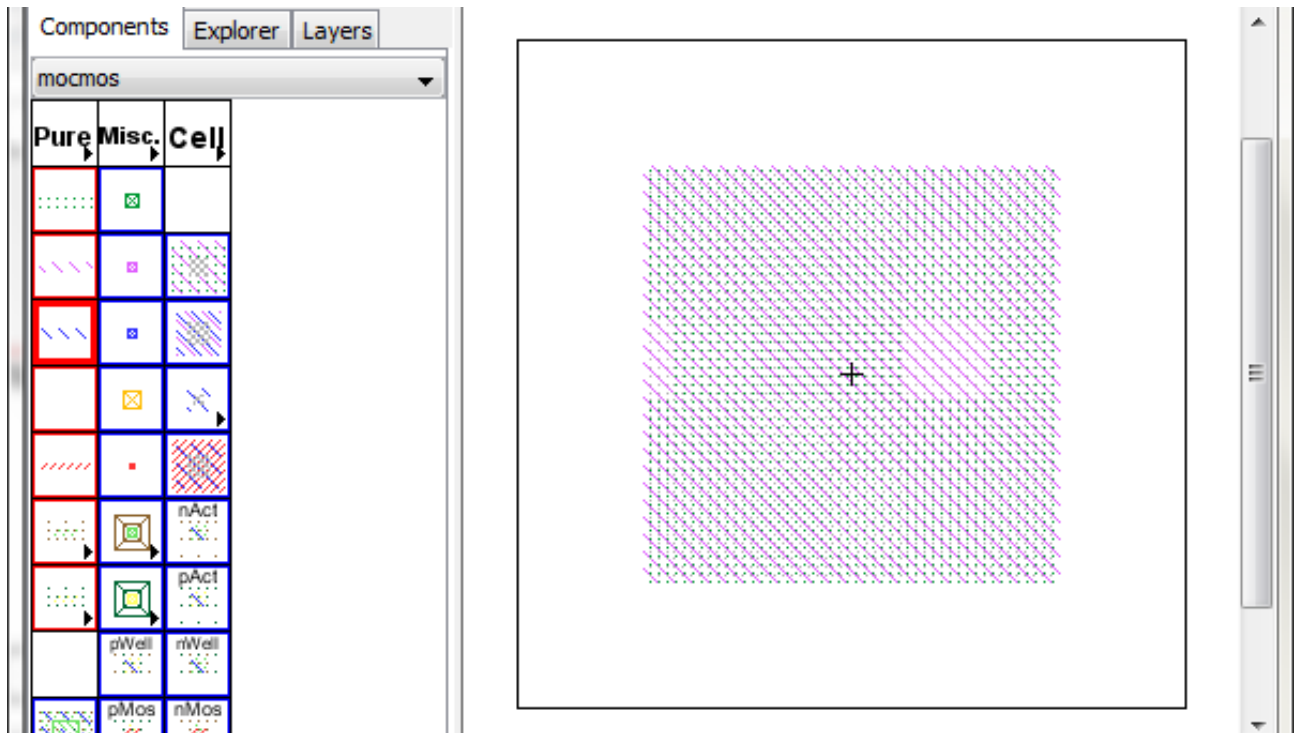


Рис. 6.8. Металізація контактної площинки

В комірці *padframe{lay}* перемикніть режим відображення (відкрите око в панелі інструментів), контактні площинки змінять вигляд (рис. 6.9.).

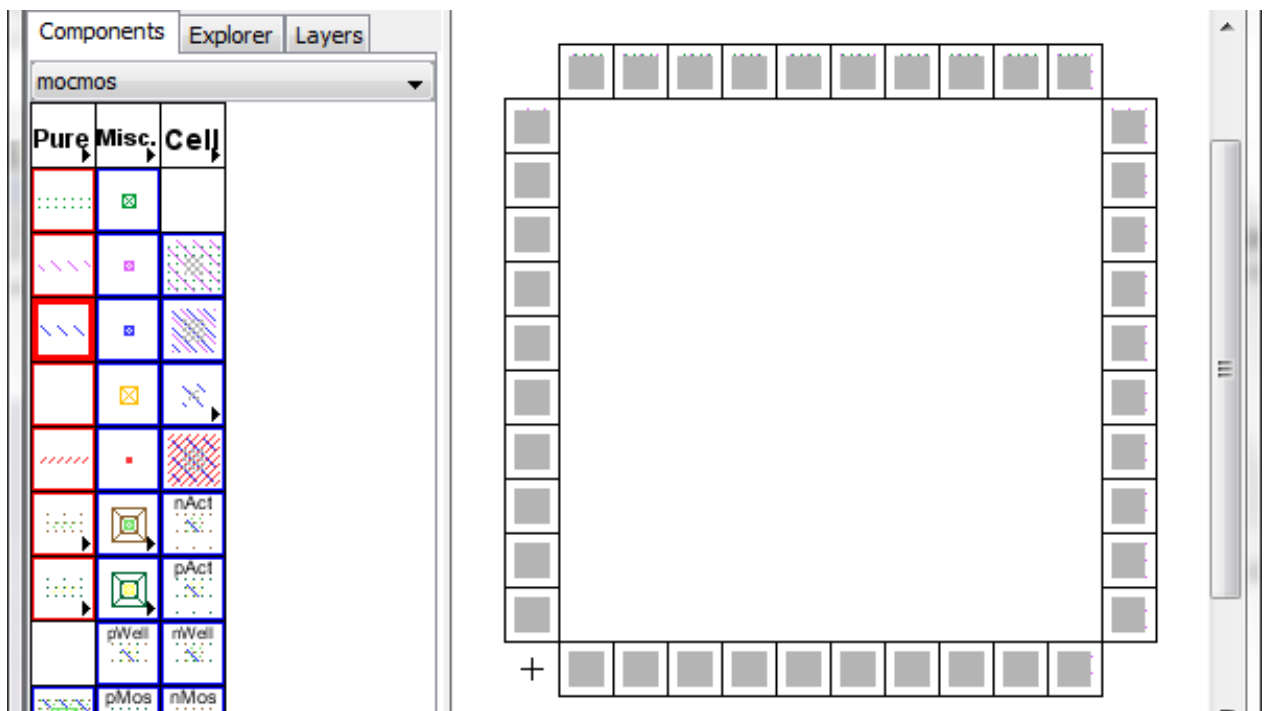


Рис. 6.9. Чип з металізованими контактними площинками

Поверніться до комірки  $pad\{lay\}$ .

Необхідно експортувати вузол  $metal2-metal3$  і додати шар пасивації так, щоб верхній шар видалити й одержати доступ до вузла  $metal3$ .

Відповідно до норм мінімальна сторона квадрата площини шару пасивації дорівнює 60 мкм, що з урахуванням коефіцієнта лямбда становить 200 одиниць.

Додайте чистий шар пасивації на контактну площинку:  $Components \rightarrow Pure \rightarrow Passivation-Node$  (рис. 6.10).

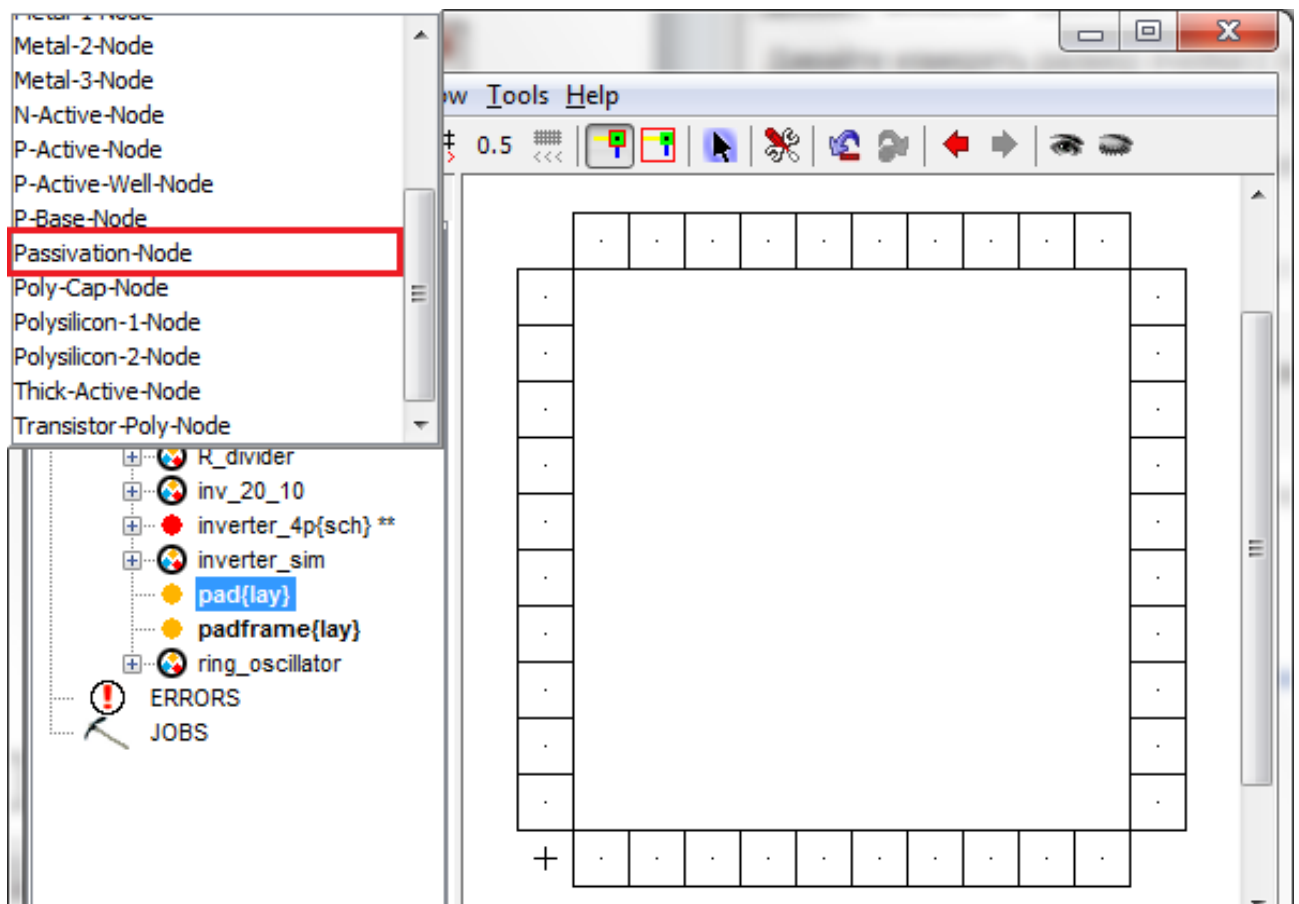


Рис. 6.10. Додавання чистого шару пасивації

Змініть розмір цього вузла на 200 x 200 і координати  $x = 0$ ,  $y = 0$  (рис. 6.11, 6.12).

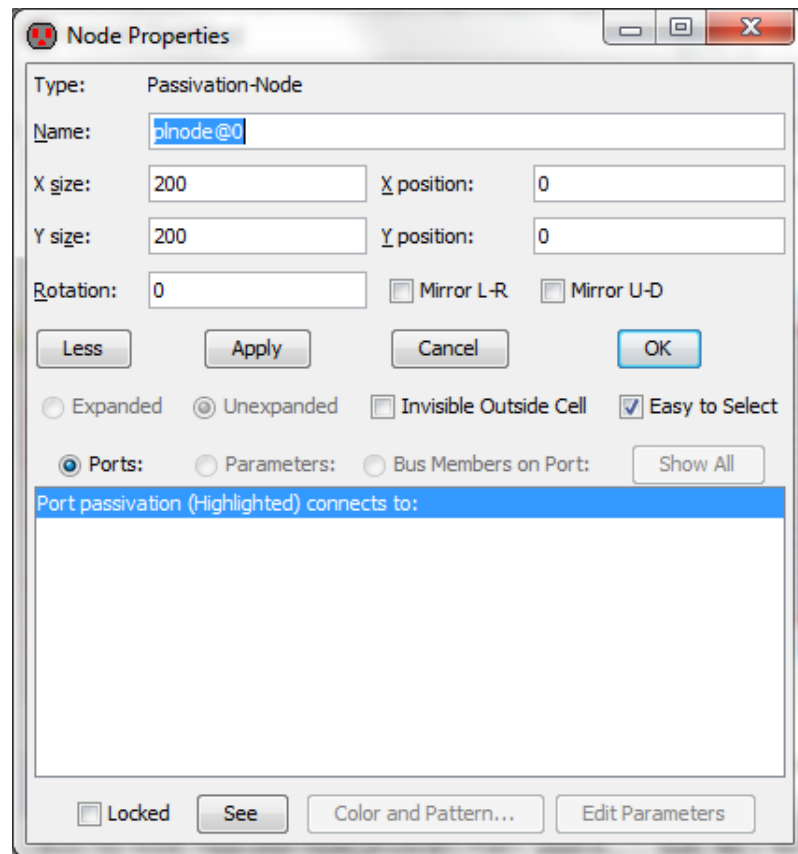


Рис. 6.11. Встановлення параметрів шару пасивування

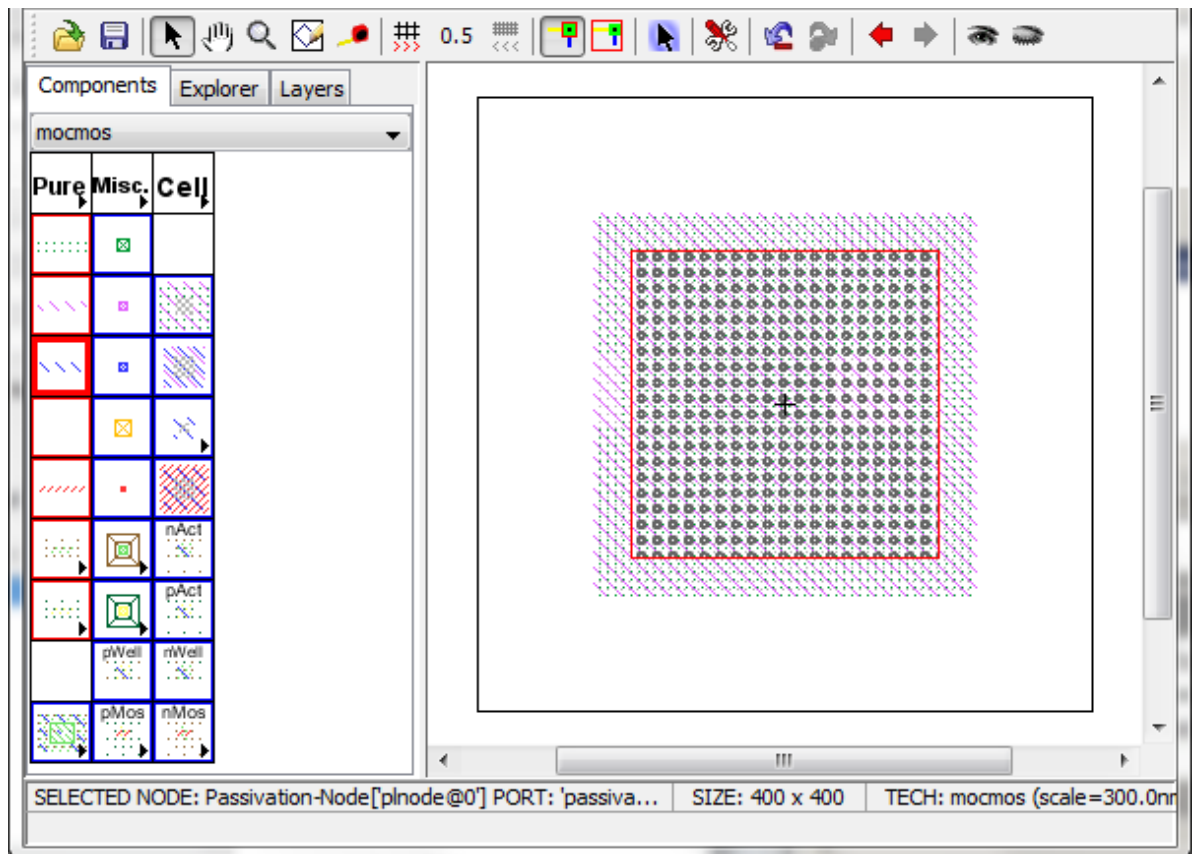


Рис. 6.12. Металізована контактна площинка з шаром пасивації

Експортуйте вузол *metal2-metal3* як *inout*, розмір тексту – 25 (рис. 6.13).

Перевірте на помилки *DRC*.

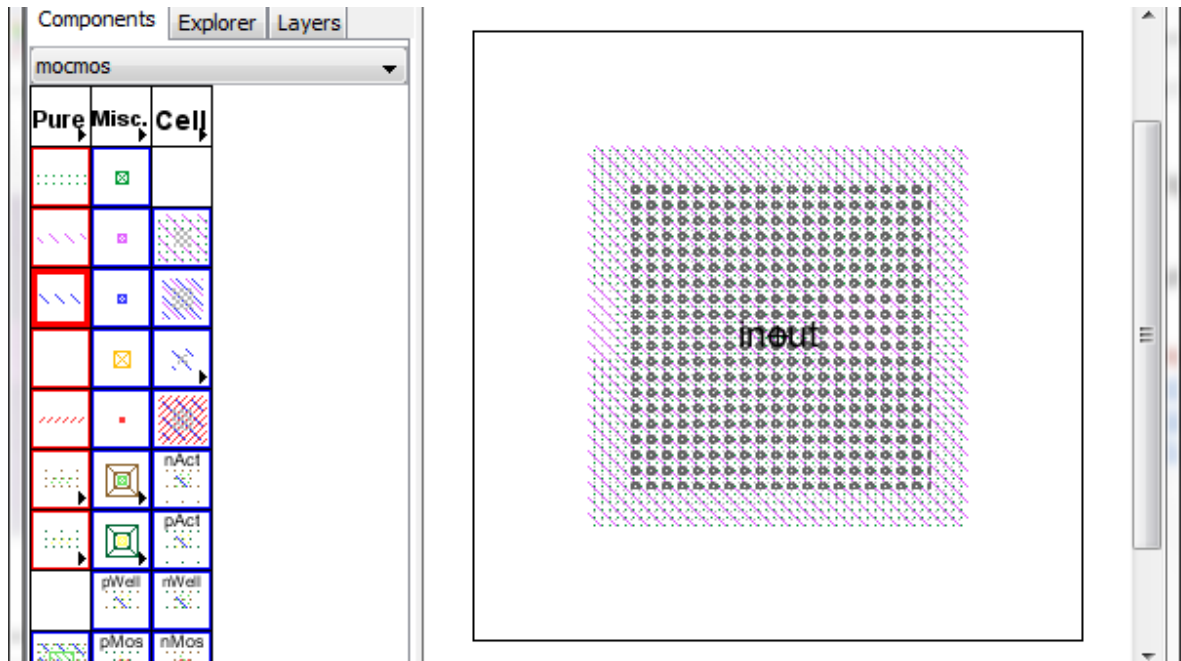


Рис. 6.13. Експортований вузол

Приступимо до побудови схеми цього макета. Створіть комірку *pad{sch}*. Додайте в *pad{sch}* вузол *off-page* (вхід) і експортуйте його як *inout*.  
Перевірте макет на помилки *DRC* і *NCC*.

Створіть піктограму комірки *pad*: *View* → *Make Icon View* (рис. 6.14).

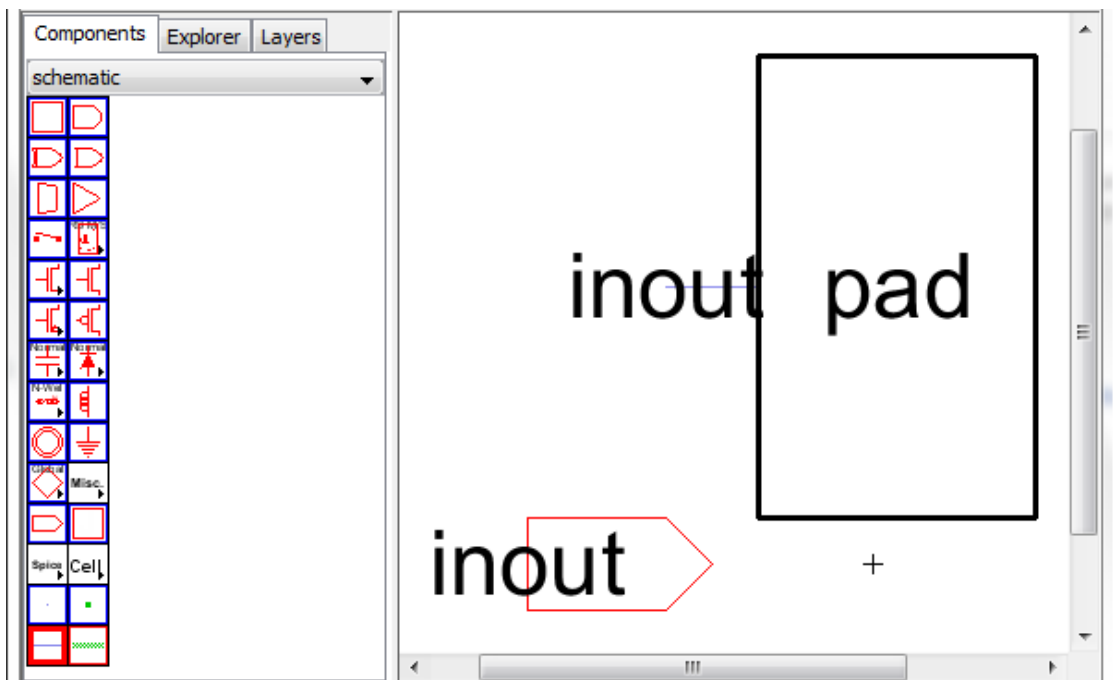


Рис. 6.14. Піктограма комірки *pad*

Перейдіть у комірку  $pad\{ic\}$  і відредагуйте піктограму ( $Edit \rightarrow Modes \rightarrow Edit \rightarrow Toggle Outline Edit$  або натисніть клавішу «Y», другий варіант – натисніть  $Ctrl+B$ ). У результаті, маєте одержати квадрат (рис. 6.15).

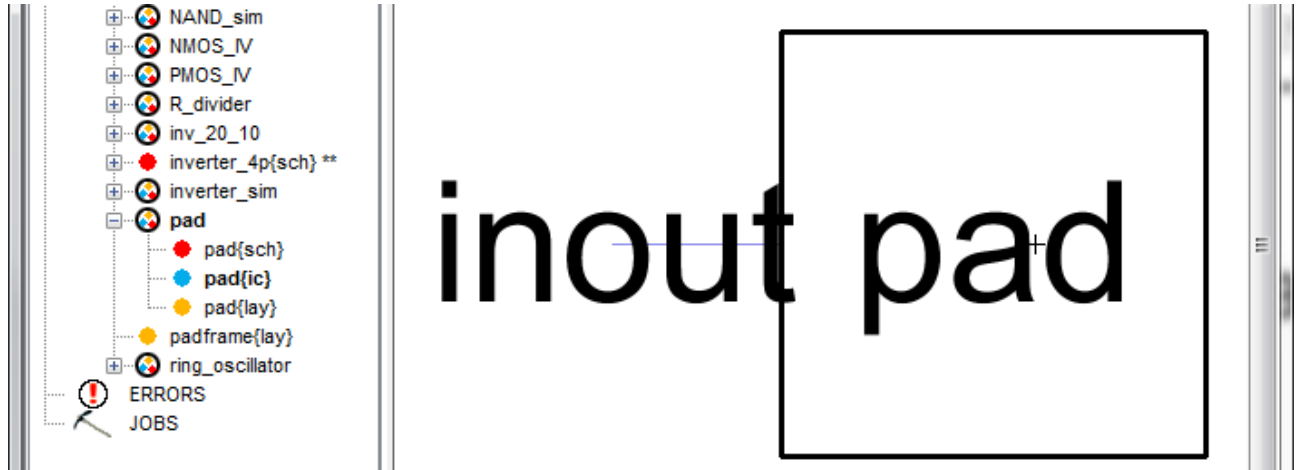


Рис. 6.15. До створення піктограми комірки  $pad$

Схематичне зображення площинки в комірці  $pad\{sch\}$  має виглядати, як на рис. 6.16.

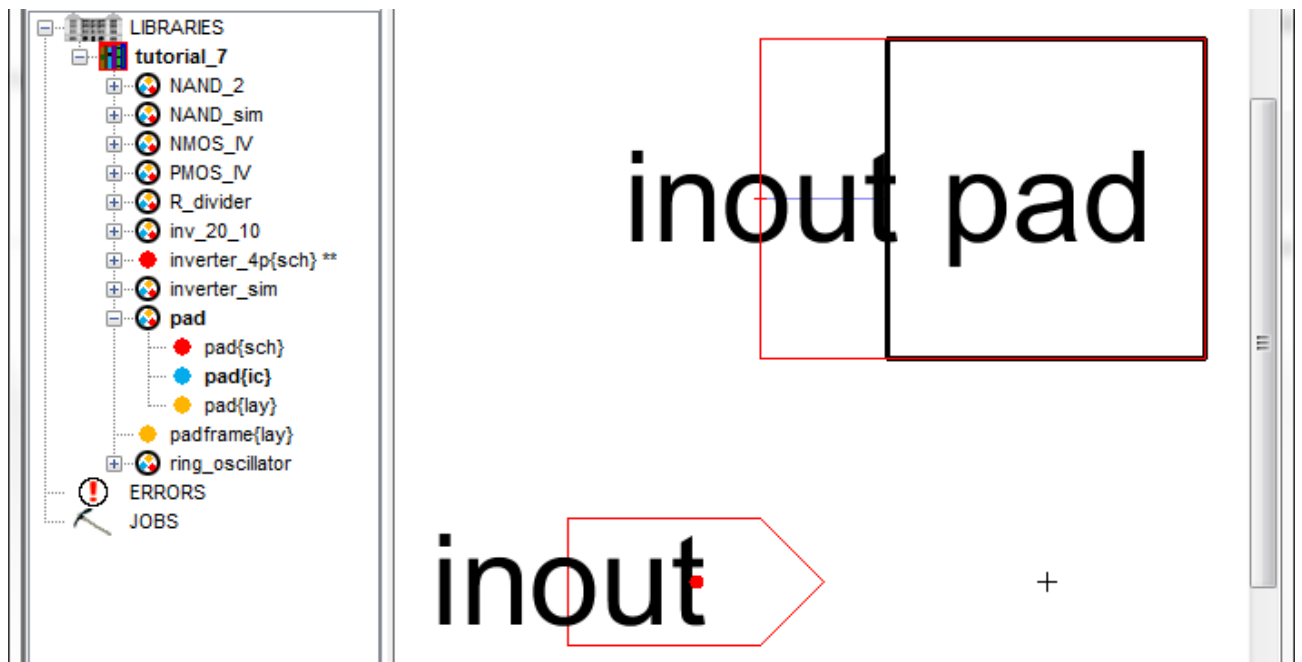


Рис. 6.16. Схематичне зображення площинки в комірці  $pad\{sch\}$



Створіть комірку  $padframe\{sch\}$ . У неї помістіть комірку  $pad\{ic\}$ , і змініть ім'я вставленої комірки на  $pad[1:40]$ . Назву комірки перемістіть вгору (рис. 6.17).

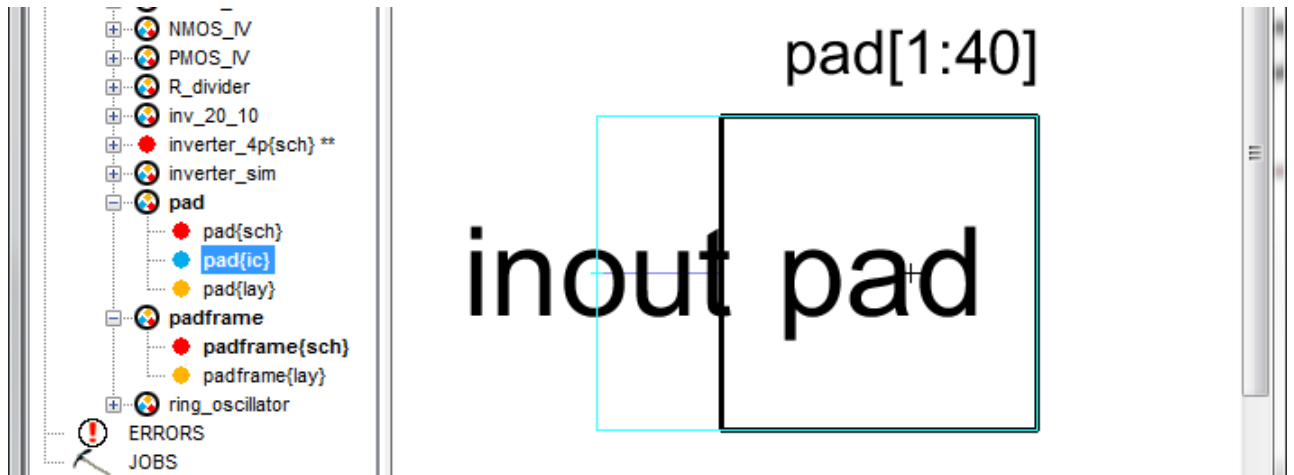


Рис. 6.17. До побудови схеми чипа

Додамо виходи до піктограми  $pad$ . Вставте в область проєктування вузол *off-page* і експортуйте його як  $pin[1:40]$  (рис. 6.18).

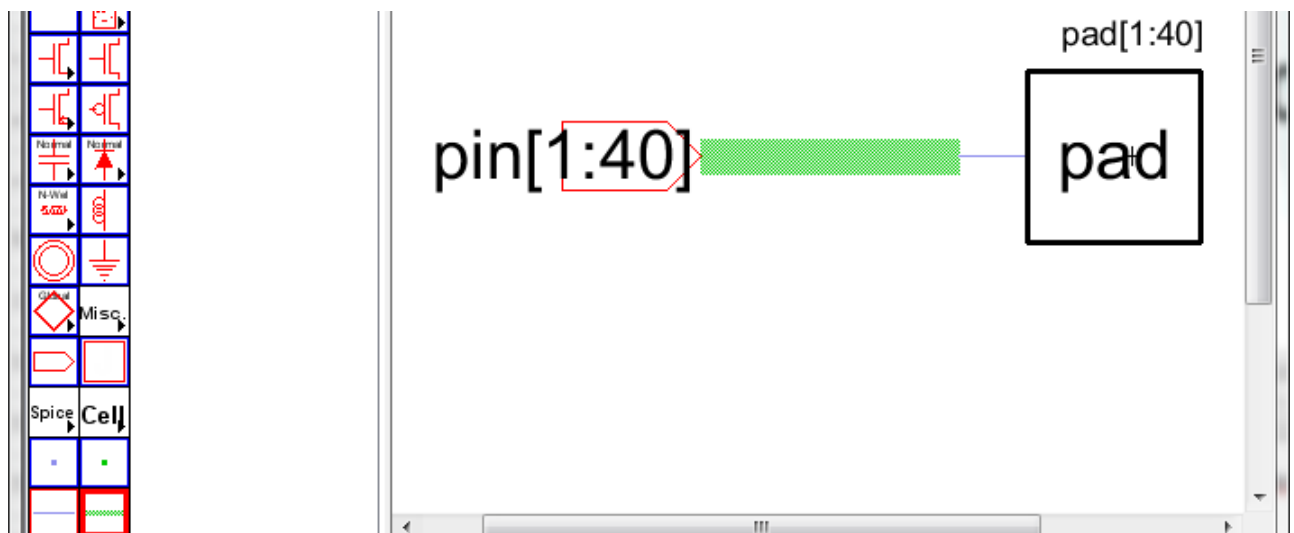


Рис. 6.18. До побудови схеми чипа

Поверніться в комірку  $padframe\{lay\}$  для того, щоб експортувати контактні площинки відповідно до комірки  $padframe\{sch\}$ .

*MOSIS* в освітній програмі використовує 40 контактний *DIP* корпус. Розглянемо діаграму розведення виводів *DIP* корпусу (рис. 6.19). Самі макети будуть перебувати в середині фігури.

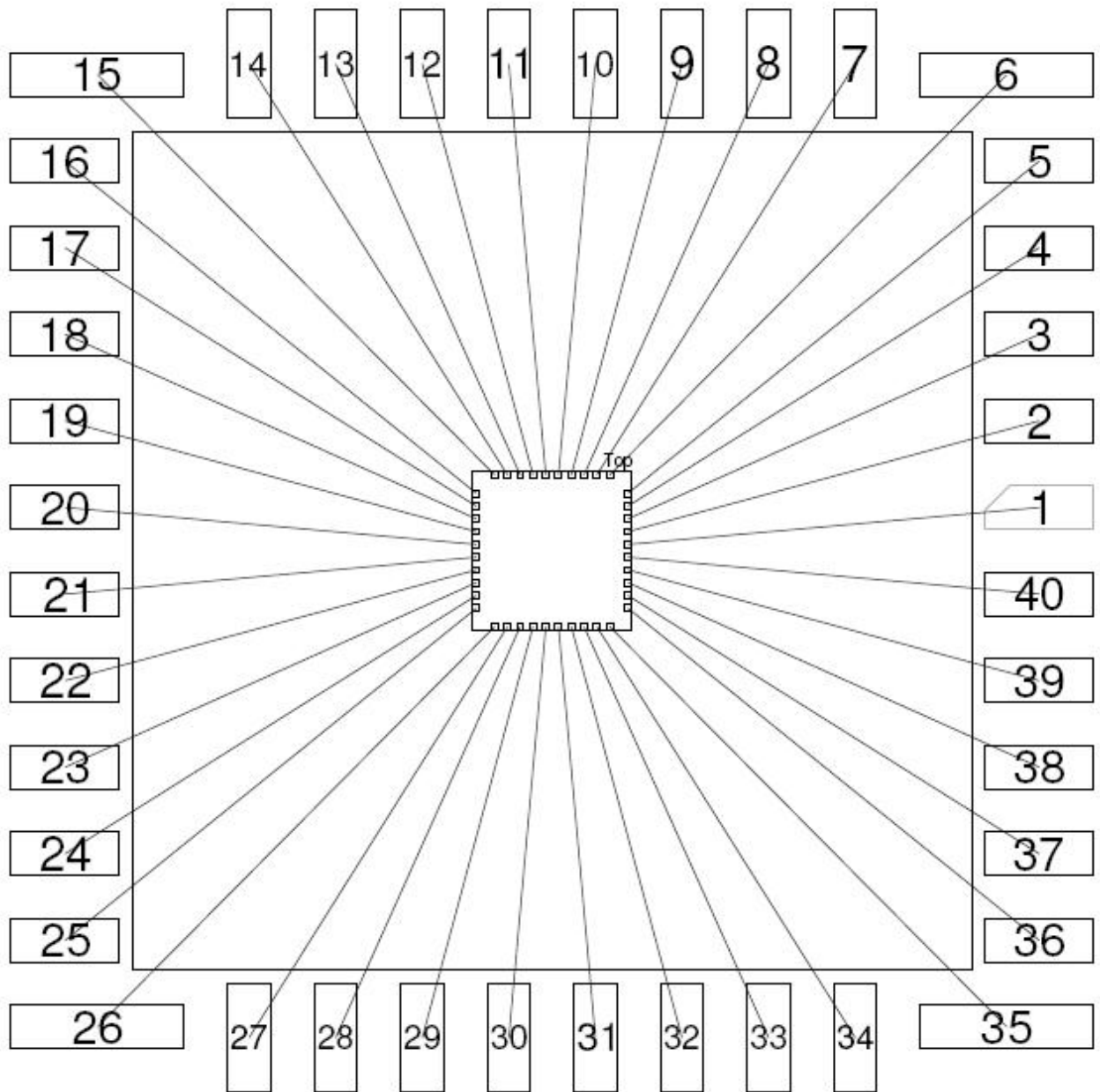


Рис. 6.19. Діаграма розведення виводів *DIP* корпусу

У відповідності зі схемою (див. рис. 6.19) *pin*[1] перебуває з правої сторони, п'ята комірнка зверху вниз. Експортуйте всі контактні площинки від *pin*[1] до *pin*[40] проти годинникової стрілки. Потім виділіть всі екпорти й змініть розмір тексту експортів на значення 125.

Результат наведено на рис. 6.20.

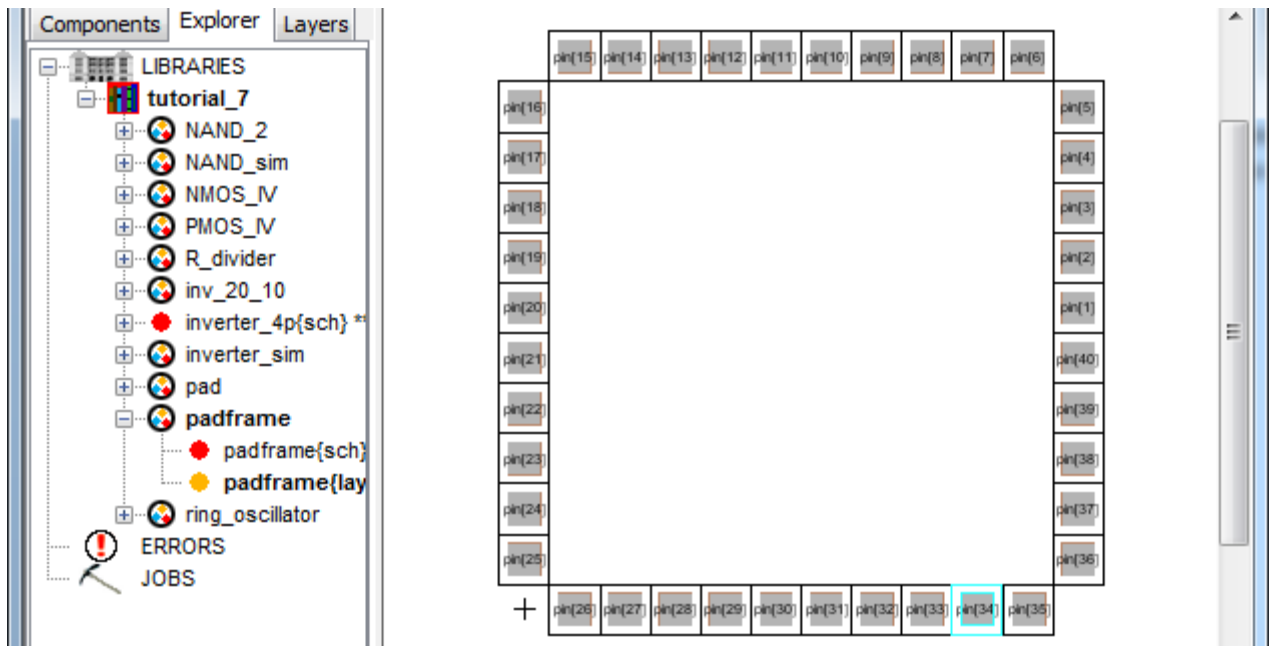


Рис. 6.20. Експортовані контакти на схемі чипа

Створіть піктограму комірки *padframe*. Для цього поверніться до комірки *padframe{sch}* і перейдіть в меню *View* → *Make Icon View* (рис. 6.21).

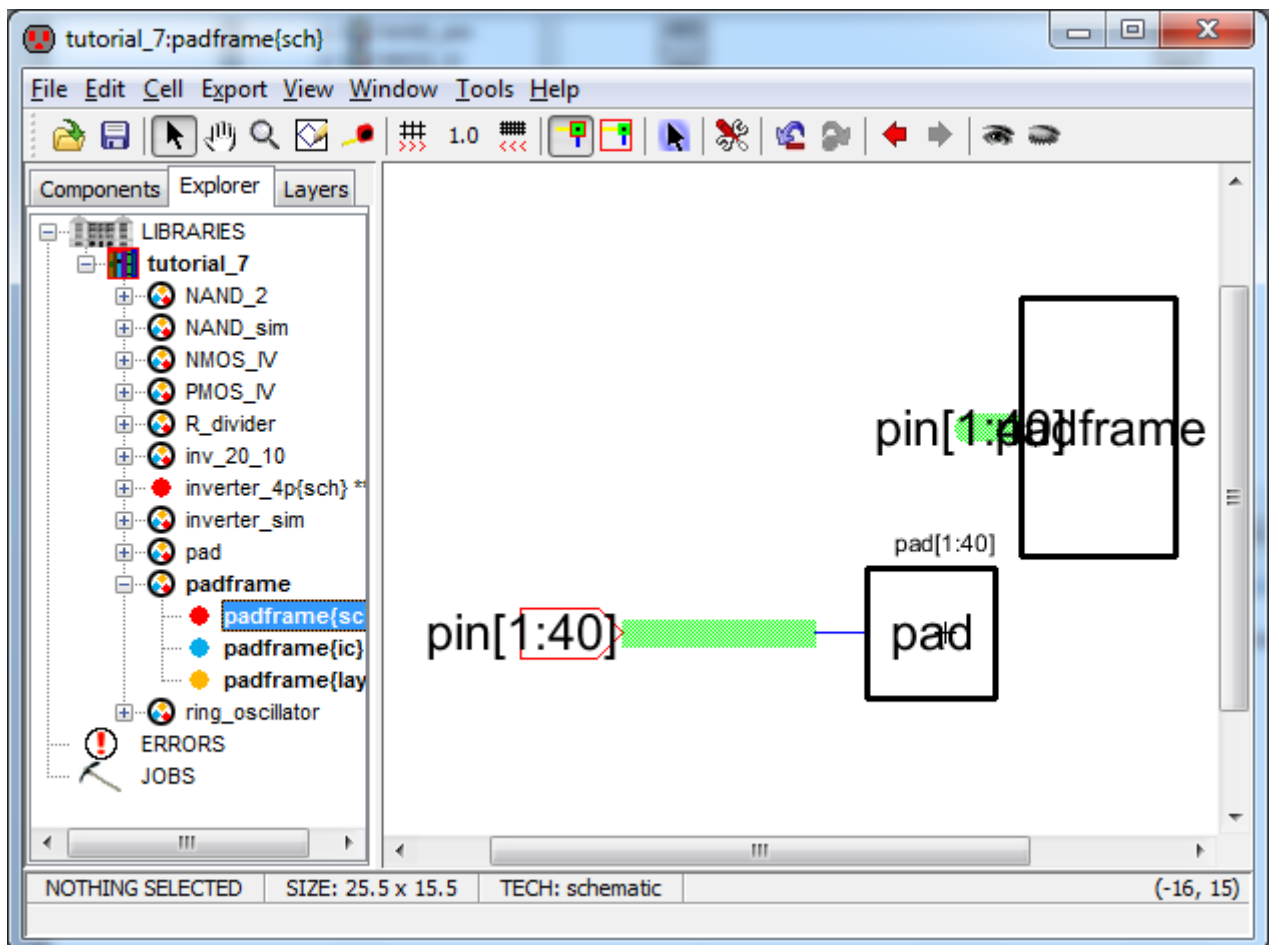


Рис. 6.21. Створення піктограми комірки *padframe*

Перейдіть в комірку *padframe{ic}* і відредагуйте піктограму. Для цього перейдіть в меню *Edit* → *Modes* → *Edit* → *Toggle Outline Edit* (або натисніть клавішу «Y»). Перемістіть текст експорту й змініть розмір прямокутника, як на рис. 6.22.

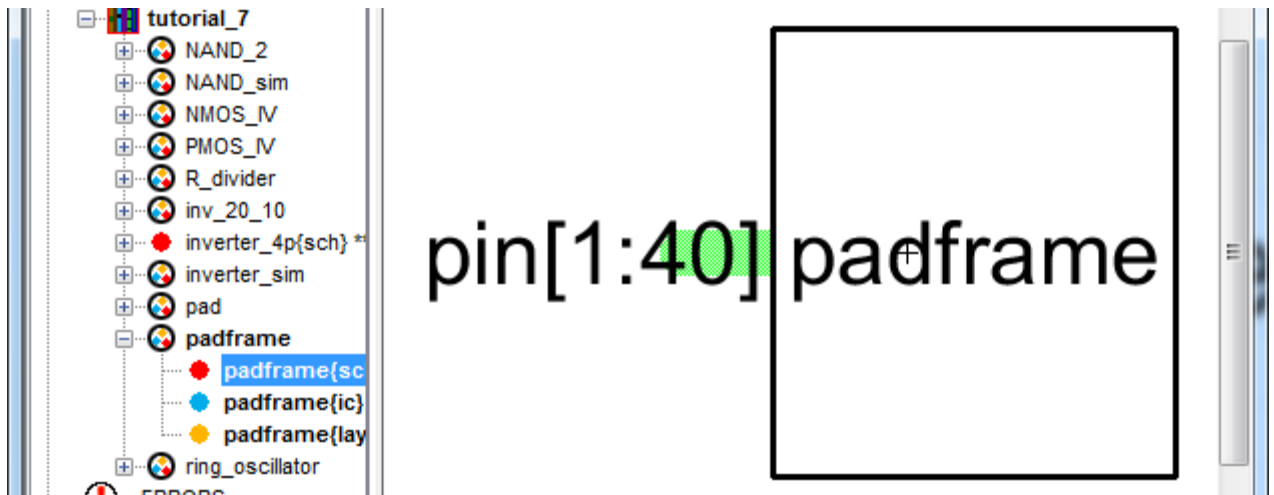


Рис. 6.22. Створення піктограми комірки *padframe*

Збережіть бібліотеку.

Створіть комірку, у якій будете проєктувати сам чип, назвіть її *A\_final\_chip{sch}*. Таку назву дайте для того, щоб комірка розміщала в самому верху вкладки *Explorer*.

На цьому чипі будемо розміщувати *NAND\_2*, *inv\_20\_10* і *ring\_oscillator*.

Оскільки дві комірки зі схемою кільцевого генератора мають вставки *spice* коду, то створіть нову версію комірки та видаліть звідти *spice* код.

Експортуйте вихід *osc\_out* як показано на рис. 6.23.

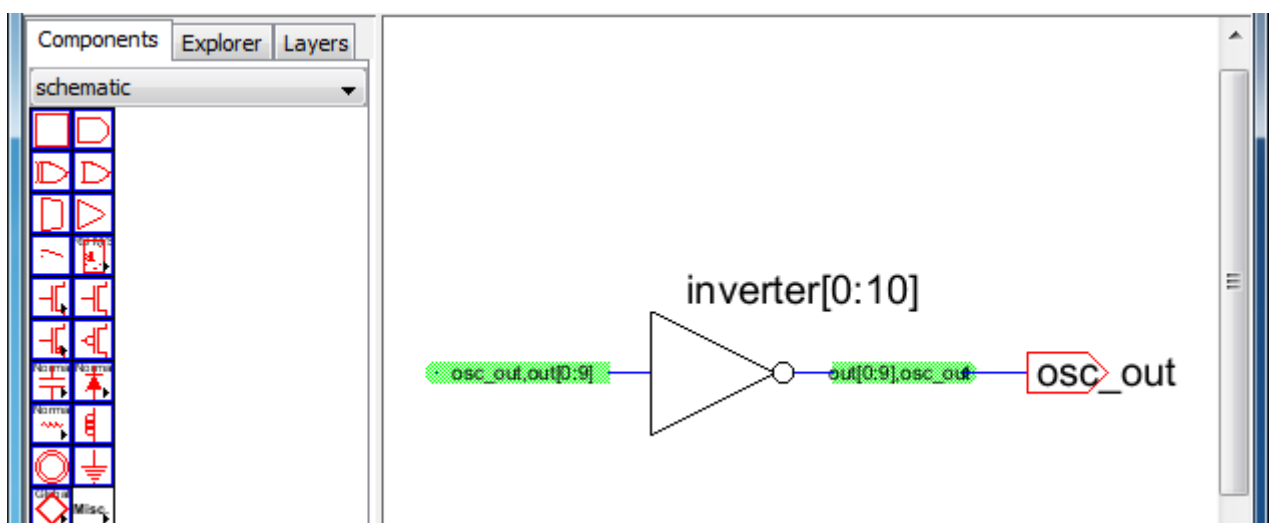


Рис. 6.23. Експорт виходу *osc\_out* кільцевого генератора

Також потрібно відкоригувати й комірку *ring\_oscillator{lay}* (за бажанням можете створювати шари дублікати). Видаліть у комірці *ring\_oscillator{lay}* назву дуги *metal1-metal2* і *spice* код, експортуйте крайній правий контакт дуги, який з'єднає вхід і вихід кільцевого генератора, як *osc\_out*. Розмір тексту – 10 (рис. 6.24).

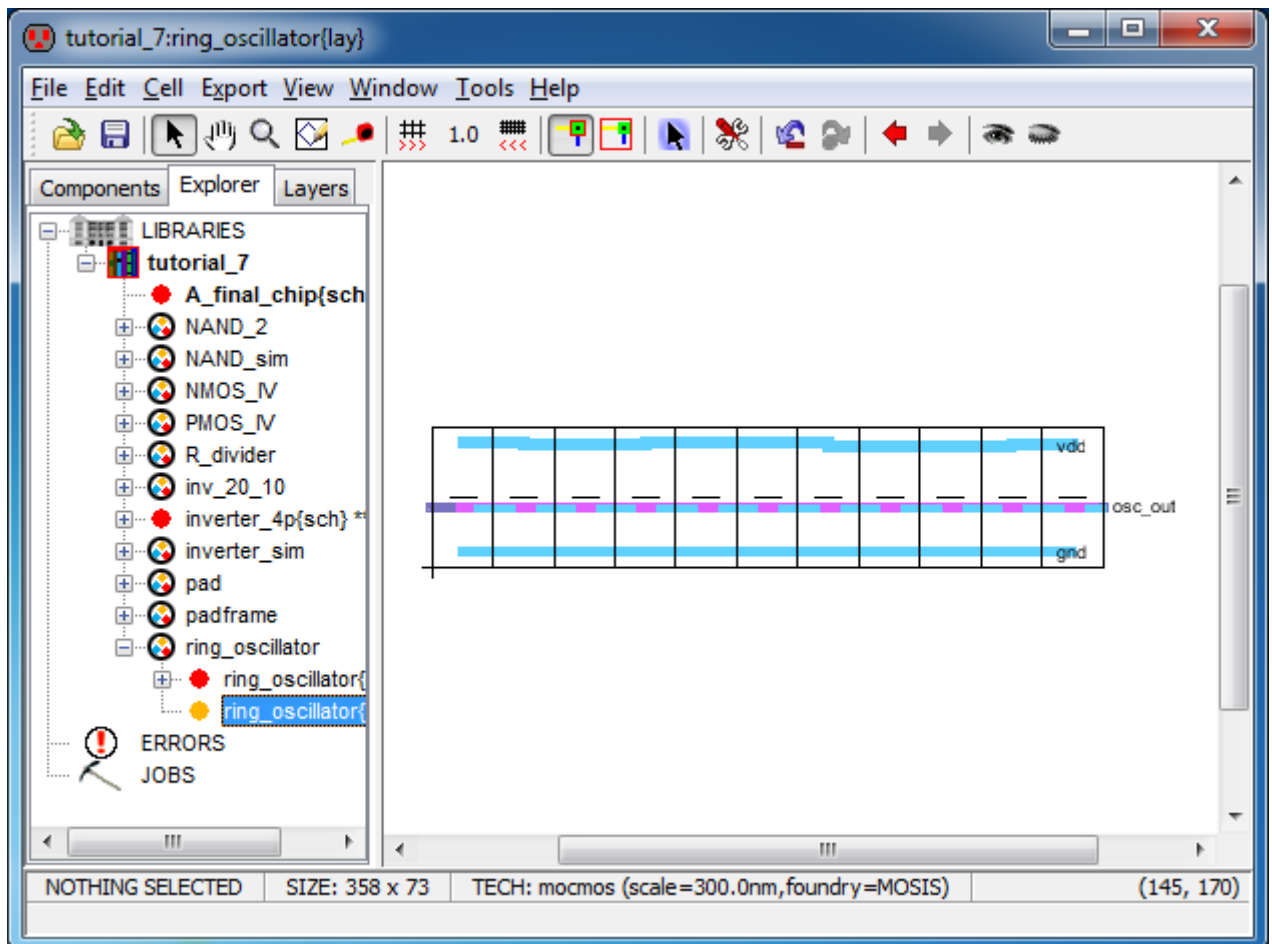


Рис. 6.24. Комірка *ring\_oscillator{lay}* після коригування

Перевірте на помилки *DRC*, *ERC*, *NCC*.

Відкрийте схемний вигляд комірки *A\_final\_chip* (*A\_final\_chip{sch}*). У цю комірку додайте піктограми *NAND\_2*, *inv\_20\_10*, *padframe* і схемне зображення кільцевого генератора (рис. 6.25).

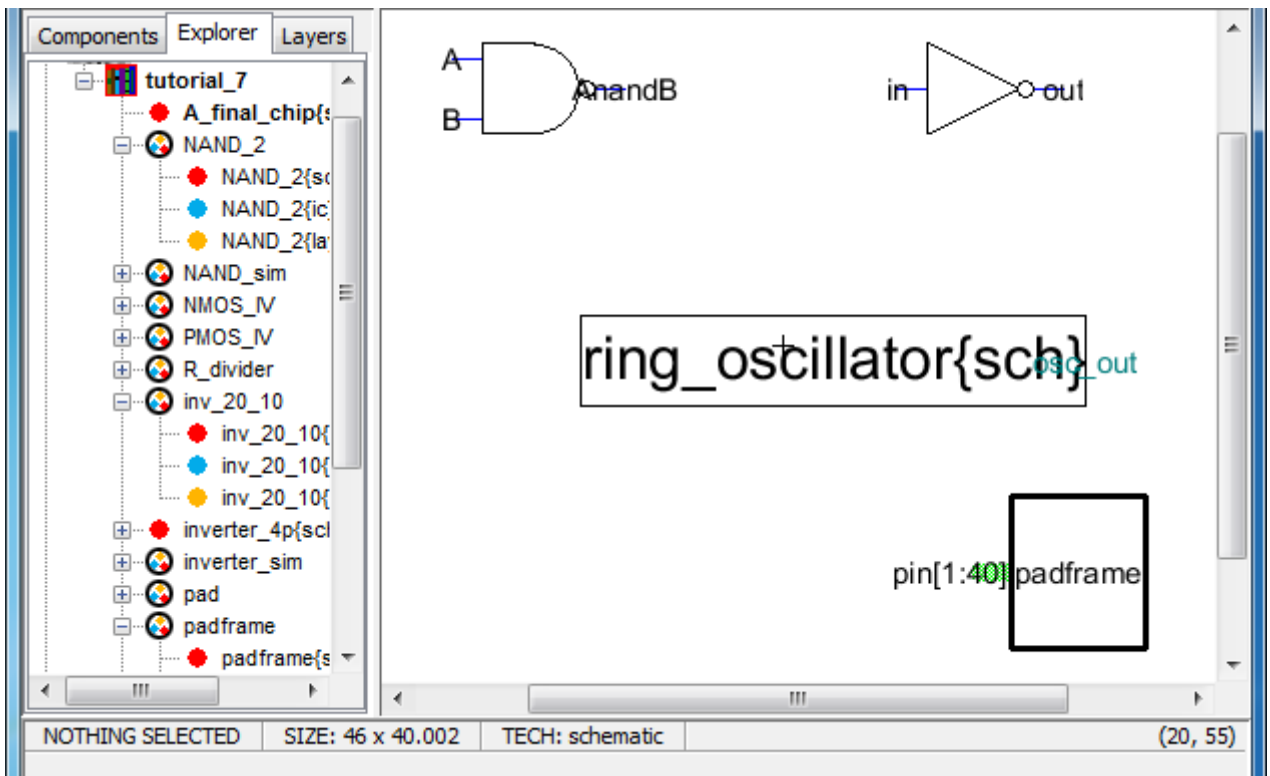


Рис. 6.25. Додавання піктограм елементів схеми у комірку  $A\_final\_chip\{sch\}$

Спочатку під'єднайте шину до піктограми  $padframe$ . Ім'я шини дуг –  $pin[1:40]$  (рис. 6.26).

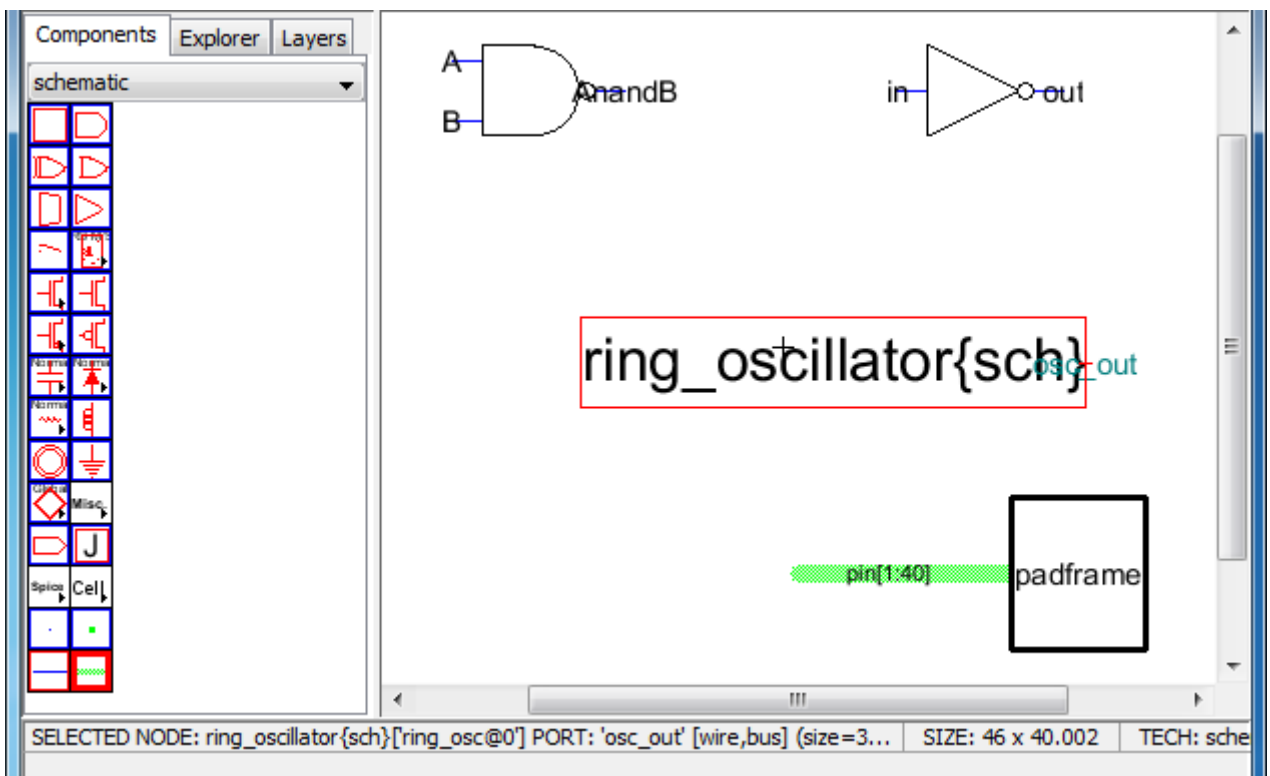


Рис. 6.26. Під'єднання шини дуг до піктограми  $padframe$

Додайте одиночні дуги до кільцевого генератора, інвертора та схеми I-НІ (рис. 6.27).

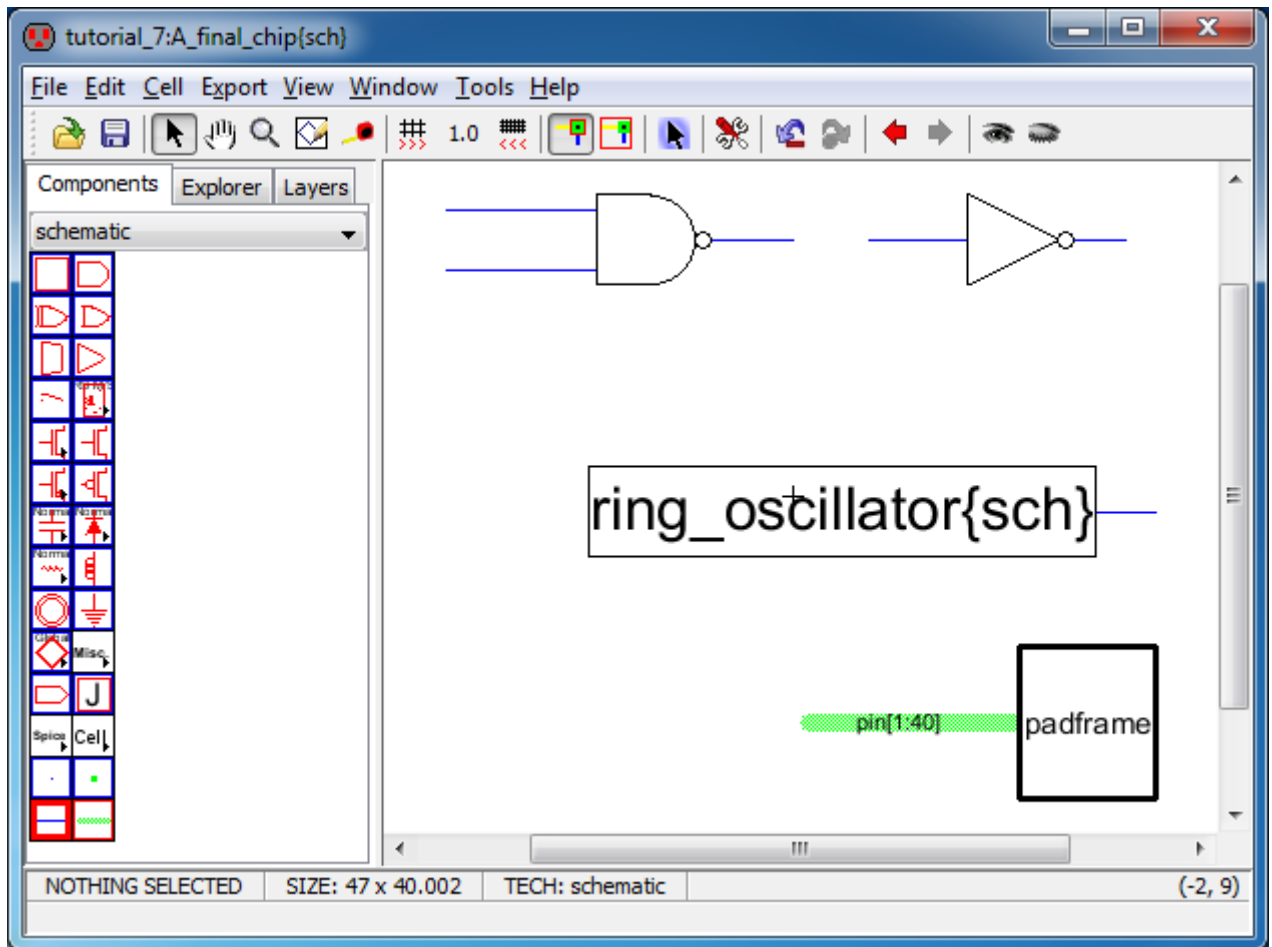


Рис. 6.27. Під'єднання одиночних дуг до кільцевого генератора, інвертора та схеми I-НІ

Також у комірку *A\_final\_chip{sch}* вставте піктограми живлення (*vdd*) і «землі» (*gnd*).

Під'єднайте контактні дуги:

- «Землю» (*gnd*) під'єднайте до *pin[20]*;
- Живлення (*vdd*) під'єднайте до *pin[40]*;
- В елементі *NAND\_2* вхід *A* під'єднайте до *pin[1]*, *B* – до *pin[2]*, *AnandB* – до *pin[3]*;
- Вхід інвертора під'єднайте до *pin[4]*, вихід – до *pin[5]*;
- Кільцевий генератор під'єднайте до *pin[6]*.

Результат таких дій наведено на рис. 6.28.

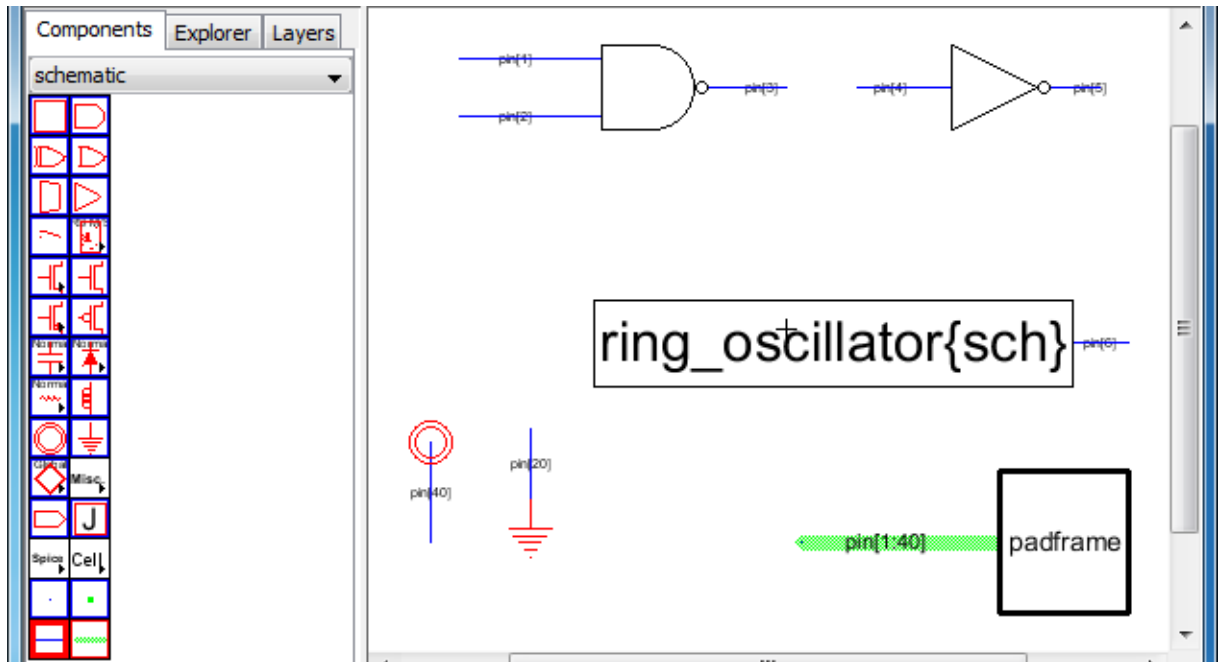


Рис. 6.28. Під'єднання контактних дуг у комірці  $A\_final\_chip\{sch\}$

Необхідно розмістити  $NAND\_2$ ,  $inv\_20\_10$  і  $ring\_oscillator$  на макеті  $A\_final\_chip$ . Тобто, потрібно створити комірку  $A\_final\_chip\{lay\}$ , і розмістити там потрібні макети.

Розмістіть у комірці  $A\_final\_chip\{lay\}$  комірку  $padframe\{lay\}$ . Перенесіть у чип (тобто в зображення з  $padframe\{lay\}$  перенесене в  $A\_final\_chip\{lay\}$ )  $NAND\_2$ ,  $inv\_20\_10$  і  $ring\_oscillator$  (рис. 6.29).

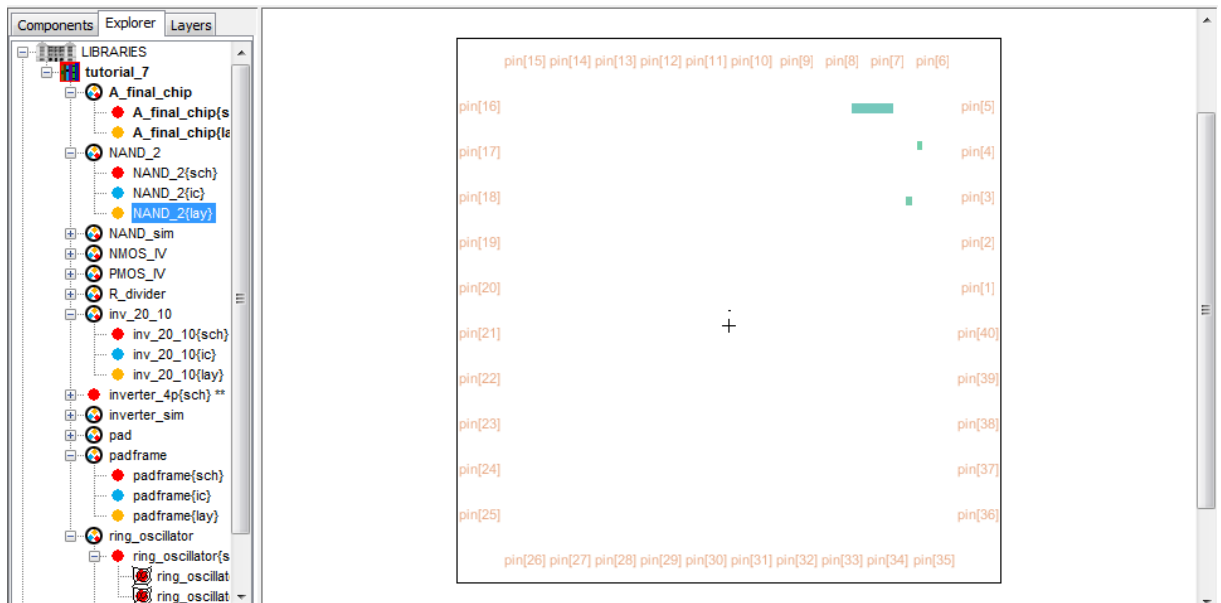


Рис. 6.29. Розміщення макетів  $NAND\_2$ ,  $inv\_20\_10$  і  $ring\_oscillator$  у комірці  $A\_final\_chip\{lay\}$



Для зручності змініть режим вибору *padframe* з легкого способу вибору на важкий: зніміть галочку біля *Easy to select* у властивостях *padframe* (рис. 6.30).

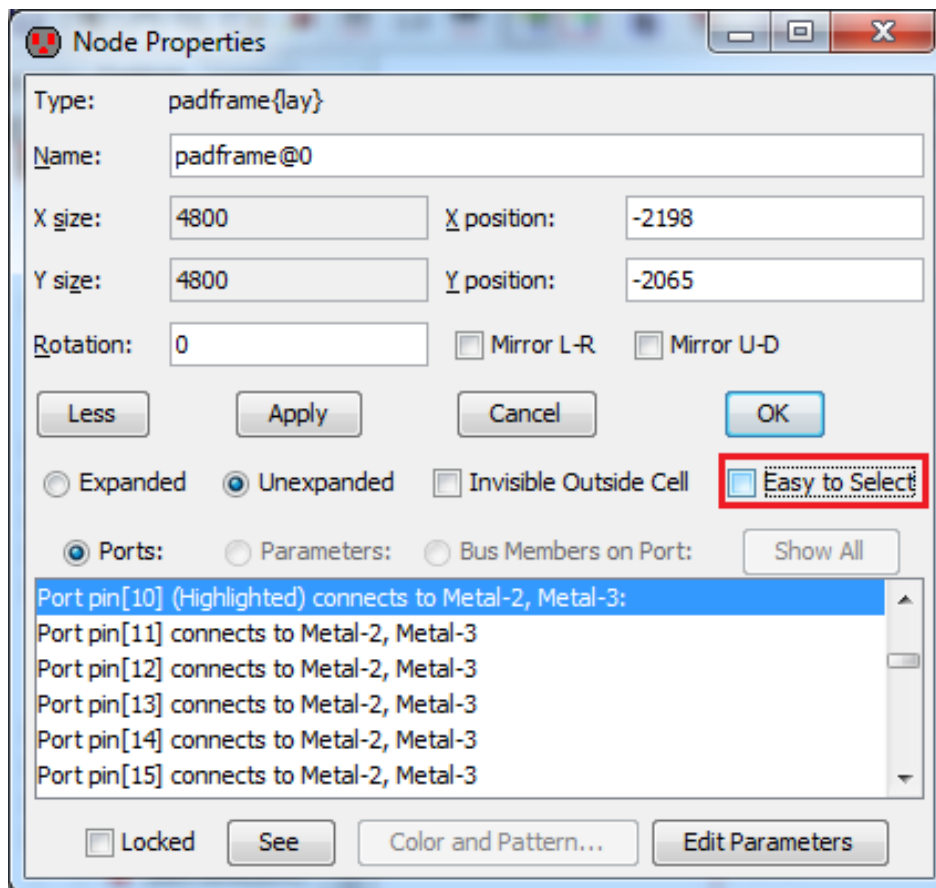



Рис. 6.30. Зміна режиму вибору у властивостях *padframe*

Якщо цього не зробити, то буде важко вибрати розміщені з контактними площинками елементи.

Щоб мати можливість вибирати елементи з відключеною можливістю легкого вибору використовуйте інструмент *Toggle Special Select* .

Під'єднайте *pin[20]* до заземлення на *NAND\_2*. Для цього увімкніть *Toggle Special Select* і виберіть *pin[20]*, після вибору вимкніть *Toggle Special Select*. Збільшіть зображення *NAND\_2* і клацнувши правою кнопкою мишки по контакту заземлення *NAND\_2* з'єднайте *pin[20]* із заземленням *NAND\_2*. Зверніть увагу, автоматично прорисовалася дуга *metal2-metal1* (рис. 6.31).

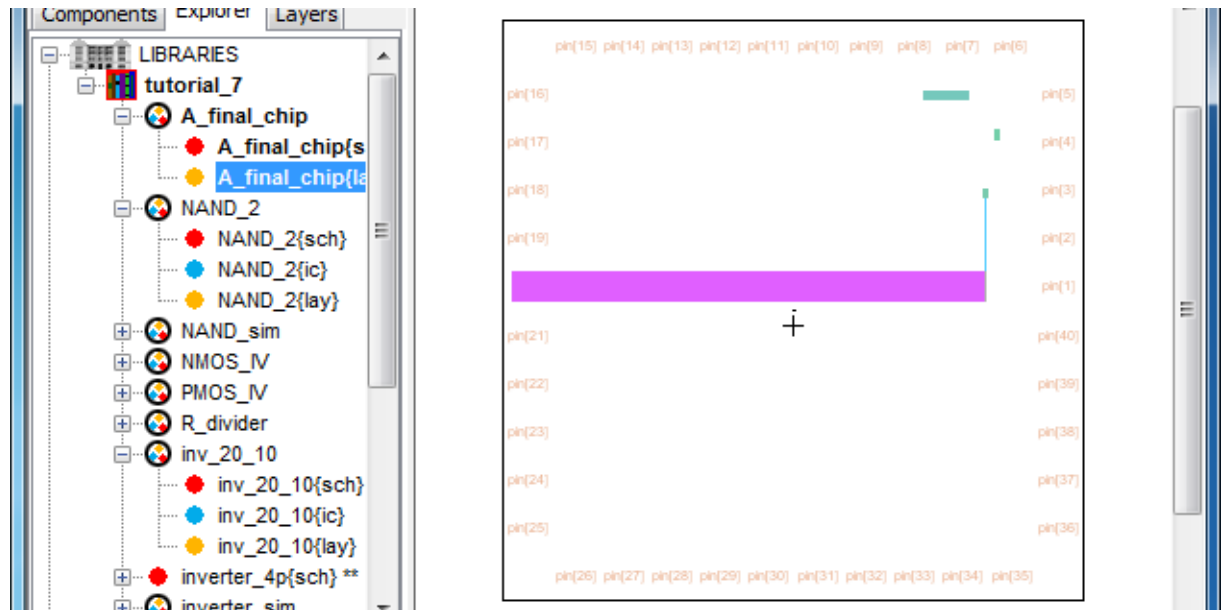


Рис. 6.31. Під'єднання  $pin[20]$  до заземлення на  $NAND_2$

Аналогічно під'єднайте «землю» до інвертора (рис. 6.32).

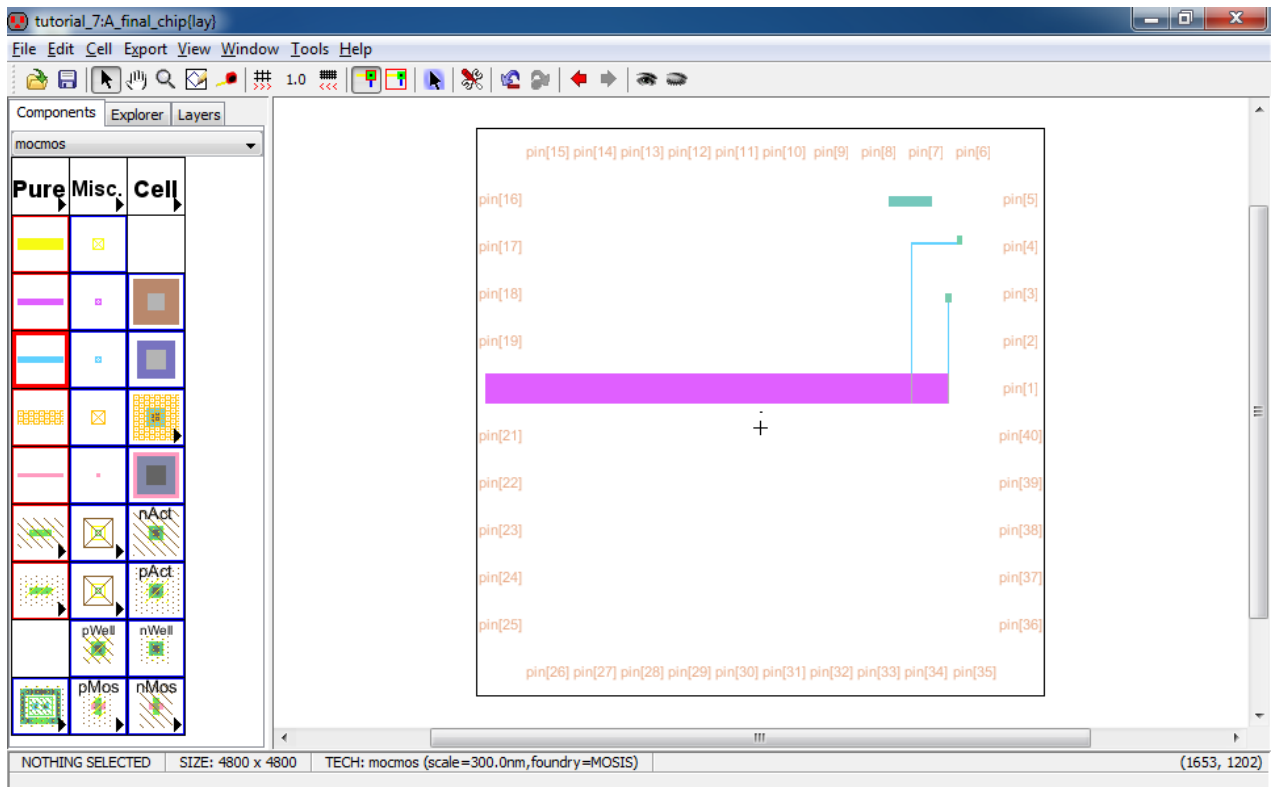


Рис. 6.32. Під'єднання  $pin[20]$  («землі») до інвертора

Зверніть увагу! Оскільки практична робота є тренувальним заняттям, і з урахуванням того, що на чипі багато вільного місця, ми не задаємо значення ширини провідників.

За аналогією з попереднім під'єднанням контактів, під'єдняйте інші контакти схем у відповідності із розміткою в комірці  $A\_final\_chip\{sch\}$ . Також екпоруйте  $vdd$  і  $gnd$ , розмір тексту – 125 (рис. 6.33).

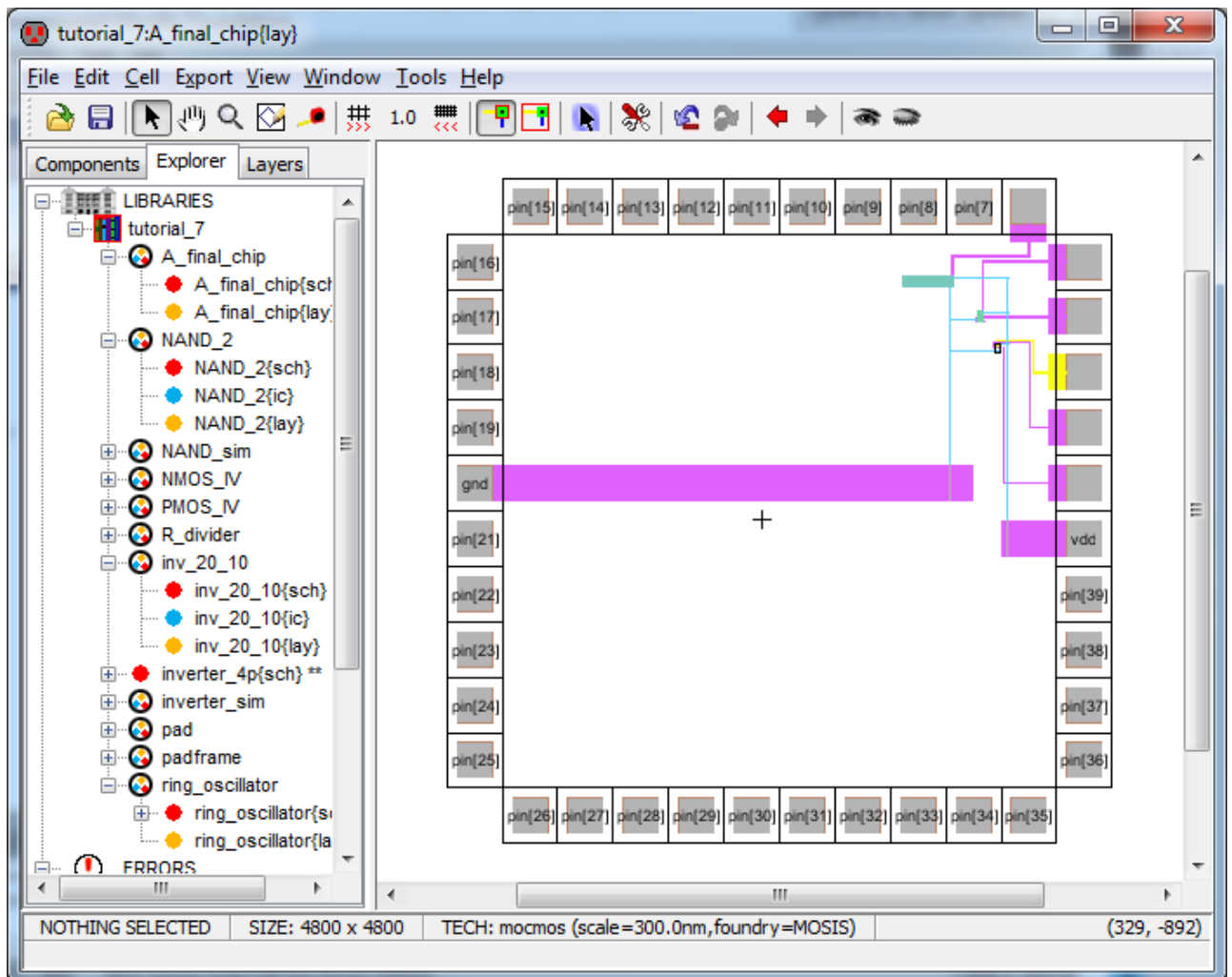


Рис. 6.33. Готовий чип

У процесі розведення можуть виникнути помилки. Як правило, це помилки відстаней між вузлами. Вважається, що, якщо Ви дійшли до трасування, то проробили попередні практичні роботи, і в змозі самостійно провести роботу над помилками.

Перевірте макет. Помилки не має бути.

Залишилося екпортувати проєкт у файл формату *GDSII*: меню *File* → *Export* → *GDS II (Stream)*.

Експортований файл має всю необхідну інформацію про даний чип. Відправивши цей файл на виробництво, Ви отримаєте готову схему.

## Список використаних джерел

1. Rubin S. M. Using the Electric VLSI Design System, version 9.07 / Steven M. Rubin. – R.L. Ranch Press, 2016. – 400 p.
2. <https://www.staticfreesoft.com>

## Рекомендована література

1. Rubin S. M. Using the Electric VLSI Design System, version 9.07 / Steven M. Rubin. – R.L. Ranch Press, 2016. – 400 p.
2. Рабаи Ж. М. и др. Цифровые интегральные схемы. Методология проектирования. 2-е изд. / Жан М. Рабаи, Ананта Чандракасан, Боривож Николич. – М. : ООО «И.Д. Вильямс», 2007. – 912 с.
3. Коледов Л. А. Технология и конструкции микросхем, микропроцессоров и микросборок : Учебное пособие. 3-е изд., стер. / Л. А. Коледов. – СПб. : Лань, 2009. – 400 с.
4. Микроэлектроника : Учеб. пособие для втузов. В 9 кн. / Под ред. Л. А. Коледова. Кн. 2. Полупроводниковые интегральные микросхемы / Г. Г. Казеннов, В. Я. Кремлев. – М. : Высш. шк., 1987. – 144 с.