

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний авіаційний університет

В.Ю. ЛАРІН, В. П. ХАРЧЕНКО

**АВТОМАТИЗАЦІЯ  
СХЕМОТЕХНІЧНОГО  
ПРОЕКТУВАННЯ**

*Рекомендовано  
Вченою радою Національного авіаційного університету  
як підручник для студентів  
вищих навчальних закладів*

Київ 2017

## З М І С Т

<b>ВСТУП</b> .....	3
<b>Розділ 1. АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ ПРИСТРОЇВ З АНАЛОГОВИМИ КОМПОНЕНТАМИ</b> .....	5
1.1. Відомості про теорію автоматизації схемотехнічного проекткування .....	5
1.1.1. Концепція крізного проектування .....	6
1.1.2. Структура та особливості роботи з програмними пакетами крізного проектування .....	7
1.2. Узагальнена методологія автоматизації схемотехнічного проекткування .....	18
1.2.1. Класифікація моделей та бібліотек моделей компонентів, застосовуваних у програмах EDA.....	18
1.2.2. Відомості про еталонну мову схемотехнічного проекткування Spice.....	20
1.3. Особливості автоматизованого проектування схем із пасивними базовими дискретними компонентами .....	25
1.4. Особливості автоматизованого проектування схем із активними дискретними компонентами.....	37
1.5. Особливості автоматизованого проектування схем із активними функціональними блоками.....	82
1.6. Автоматизація аналізу імітаційних структур схем радіоелектронних пристроїв в програмах EDA.....	90
1.7. Створення схем із дискретними аналоговими компонентами у програмах автоматизації схемотехнічного проектування.....	102
1.8. Створення схем із активними функціональними компонентами у програмах автоматизації схемотехнічного проектування.....	105
<b>Розділ 2. АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ ПРИСТРОЇВ З ЦИФРОВИМИ КОМПОНЕНТАМИ ТА КОМПОНЕНТАМИ ЗМІШАНОЇ ФУНКЦІОНАЛЬНОСТІ</b> .....	108
2.1. Теоретичне підґрунтя сучасної цифрової схемотехніки .....	108
2.2. Схемотехнічна реалізація логічних функцій. Внутрішня структура цифрових мікросхем.....	1120
2.3. Характеристики цифрових мікросхем, особливості застосування мікросхем, створених за різними інтегральними технологіями .....	127
2.4. Особливості реалізації цифрової схемотехніки у програмах автоматизації схемотехнічного проектування.....	132
2.5. Представлення цифрових компонентів у програмах автоматизації схемотехнічного проектування.....	154
2.6. Програмовані логічні інтегральні схеми.....	167
2.6.1. Відомості про програмовані логічні інтегральні схеми... ..	167

2.6.2. Синтез програмованих логічних інтегральних схем за допомогою програм автоматизації схемотехнічного проектування.....	171
2.7. Особливості автоматизованого проектування схем із багато розрядними цифро-аналоговими та аналого-цифровими перетворювачами.....	176
2.8. Автоматизоване проектування схем із запам'ятовуючими та мікропроцесорним пристроями.....	180
СПИСОК ЛІТЕРАТУРИ.....	203

## РОЗДІЛ 1

### АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ ПРИСТРОЇВ З АНАЛОГОВИМИ КОМПОНЕНТАМИ

#### 1.1. Відомості про теорію автоматизації схемотехнічного проектування

Теорія автоматизації схемотехнічного проектування є системною дисципліною. Вона з'явилася на стику низки базових та суміжних дисциплін : класичної теорії електричних ланцюгів, теорії графів, теорії програмування, обчислювальної математики, фізики твердого тіла, мікроелектроніки, параметричної оптимізації. Якщо розглядати сукупність методів, використовуваних в кожній базовій дисципліні, в якості множин, то теорію АСхП можна уявити як перетинання цих множин у простір таких факторів, як час розрахунку, стійкість обчислення кількості початкової інформації, ти незалежних змінних та ін.

Найбільш складною складовою частиною теорії АСхП є схемотехнічне моделювання, яке базується на використанні математичного апарату обчислювальних методів. Тому варто навести вимоги, які висуває теорія АСхП до обчислювальних методів. Розташуємо ці вимоги у порядку актуальності. Перша вимога – це алгоритмічна надійність метода, тобто гарантоване отримання вірного прикінцевого результату при будь-яких чисельних значеннях параметрів і для будь-яких видів функціональних залежностей в задачах даного класу. Друга вимога до обчислювальних методів, полягає у можливості їх алгоритмізації. Третя вимога до методів на сьогоднішній час не є дуже актуальною завдяки сучасному розвитку обчислювальної техніки. Вона полягає у тому, щоб метод повинен достатньо швидко збігатися з метою усунення збільшення часу обчислення. Ці жорсткі вимоги одразу скорочують безліч розроблених обчислювальних методів до досить малої кількості методів, які можна реалізувати в алгоритмічному забезпеченні програмних модулів схемотехнічного моделювання. [ ]

Через це у даному підручнику не буде приділено багато уваги розгляданню суті та особливостей застосування того чи іншого обчислювального методу, застосованого в програмному забезпеченні засобу АСхП. Крім того, ці запитання достатньо детально розглянуто як у вітчизняній так і у закордонній навчальній літературі, яка присвячена саме моделюванню електронних систем. У підручнику буде зазначено метод, використовуваний для автоматизованого аналізу, його коротка характеристика, зазначені обмеження на його використання із наданням необхідних порад щодо настроювання програмного інтерфейсу, вказання вхідних початкових значень при виконанні автоматизованого аналізу імітаційної моделі схеми за допомогою цього методу.

### 1.1.1. Концепція крізного проектування

Крізне проектування це ідеологія сучасного автоматизованого проектування, яка передбачає швидке корегування файлів віртуального проекту пристрою завдяки міжмодульним зв'язкам в структурі одного програмного пакету АСхП. Це дозволяє синхронізувати роботу над проектом, коли вона виконується конструкторським колективом (design team), де кожен відповідає за свою ділянку роботи, і при наявності корекції, наприклад, віртуальної принципової схеми, автоматично вносити ці зміни у проект монтажною схемою друкованої плати. Безумовно, ці всі операції можна застосовувати, коли роботу над проектом виконує один проектувальник. Таким чином крізне проектування є досить гнучкою технологією. Згідно [ ] в технології крізного проектування виділяють такі етапи:

- системне або функціональне проектування, у процесі якого виникає функціональна схема пристрою, і прораховуються загальносистемні характеристики.
- схемотехнічне проектування – кожному функціональному вузлу ставлять у відповідність схемне рішення з урахуванням наявної елементної бази. При проектуванні цифрових пристроїв цей етап коректніше називати логічним проектуванням.

- конструкторське проектування – проробляють основні конструктивні параметри пристрою з обліком необхідної елементної бази, призначення, заданими массогабаритними параметрами, діапазоном частот, споживаною потужністю, температурним режимом і т.п.
- технологічне проектування – вибирається підходяща послідовність технологічних операцій, для кожної операції конкретна технологія, матеріали, режим технологічного устаткування й т.п.

Кожен етапі закінчується створенням необхідної документації. Процес проектування, найчастіше є ітераційною процедурою, при якій доводиться неодноразово вертатися до вихідного варіанта схемного рішення [ ]. Повномасштабне автоматизоване проектування, що включає всі етапи, імовірна справа майбутнього, хоча відомі його реалізації при розробці великих інтегральних схем. Розробка програмного забезпечення, навіть для окремого етапу проектування, вимагає величезних попередніх витрат. У розробці програмного забезпечення звичайно беруть участь, як програмісти математики, так і інженери відповідних спеціальностей. У зв'язку з тим, що ЕОМ є широко застосовуваним у інженерній практиці, вивчення питань автоматизації проектування входить у сучасні навчальні плани радіотехнічних спеціальностей. Зокрема у Російській Федерації, при підготовці студентів за фахом "Радіотехніка", освітнім стандартом передбачене вивчення питань автоматизованого схемотехнічного проектування [ ].

#### 1.1.2. Структура та особливості роботи з програмними пакетами крізного проектування

Реалізація операцій крізного проектування виконується завдяки наявності програмних процедур прямого та зворотного корегування в алгоритмічному апараті пакетів EDA. Більшість сучасних програмних пакетів АСхП, як комерційних так і з категорії безкоштовних, є системами які підтримують більшість

процедур крізного проектування – OrCAD, ACCEL EDA, Multisim, MicroCap, LTspice. Зазвичай програми АСхП, які підтримують крізне проектування містять редактор схем та компонентів, симулятор (модуль схемотехнічного моделювання), редактор монтажних схем друкованих плат із функцією автоматичного трасування. Крім того в такий програмний пакет можуть входити додаткові програмні модулі – редактор моделей компонентів, модуль параметричної оптимізації, редактор сигналів та інші.

Однією з перших систем крізного проектування АСхП був Design Lab 8.0[ ] уточнена структура якого показана на рис. 1.1. Структура, наведена на цьому рисунку демонструє основні модулі цього програмного пакету АСхП. На схемі не показані модулі перетворення інформації про проект у формати інших програм EDA P-CAD, PADS, Tango. На рис.1.1 показані такі програмні модулі: Schematics – редактор схем та компонентів; PSpice – модуль схемотехнічного моделювання; PCBoards – редактор монтажних схем друкованих плат (без функції авто трасування); Specstra – зовнішній авто трасувальник; Pspice Optimizer – модуль параметричної оптимізації; StmEd – редактор сигналів; Parts– редактор математичних моделей компонентів; PLSyn – модуль синтезу програмованих логічних структур; Probe – модуль графічного відображення результатів моделювання; Polaris – модуль розрахунку цілісності сигналів (для ВЧ схем).

Рис. 1.1 – Структура програмних модулів пакету Design Lab 8.0

Також показані основні бази даних бібліотек – математичних моделей компонентів, графіки елементів, пакування, контактних площадок. Недоліками цього, дійсно потужного програмного середовища схемотехнічного проектування, вважають

- відсутність власного авто трасувальника, для цієї мети використовували інтерфейс із програмою автотрасування Specstra (фірми Cyan&Cooper Technology);
- не дуже зручний спосіб відображення результатів аналізу перехідних процесів (вихідний файл цих результатів оформлюють у форматі відео файлу формату *.dat*);

- наявність великої кількості вікон, які відкриваються під час розрахунку, із складними взаємозв'язками між ними.

Але останнє твердження є скоріше недоліком с точки зору складності процесу навчання, зокрема студентів. Звісно, що для серйозних проектувальних робіт із підтримкою кризного проектування ця особливість не є проблемою, до того ж цей процес можна оптимізувати шляхом відповідного настроювання.

Після об'єднання корпорації–розробника пакету Design Lab 8.0 із фірмою OrCAD наступна версія цього середовища, яка вийшла під назвою OrCAD 9.2 мала власний вбудований модуль автоматичного трасування.

Згідно структурі Design Lab 8.0 (рис.1.1), роботу над проектом починають із створення імітаційної моделі принципової схеми пристрою (або її конвертації з інших систем EDA) у графічному редакторі схем та компонентів Schematics. Редактор схем дозволяє розміщувати на полі проекту елементи схем, які витягають із інтегрованих бібліотек компонентів, джерела сигналів та поєднувати їх віртуальними лініями зв'язку. Після створення файлу проекту схеми пристрою, і на його основі автоматичного створення файлів списків з'єднань можна було виконувати імітаційне моделювання схеми шляхом настроювання потрібних директив моделювання за допомогою модуля PSpice, виконувати параметричну оптимізацію за критерієм забезпечення заданого значення цільової функції при виконанні лінійних та нелінійних обмежень за допомогою модуля Pspice Optimizer, створювати монтажну схему друкованої плати за допомогою модуля PCBboards, розраховувати цілісність сигналів друкованої плати з урахуванням впливу паразитних параметрів конструкцій друкованих плат за допомогою модуля Polaris. Незважаючи на наявність розширеної бібліотеки стандартних сигналів, в структуру Design Lab уведено модуль редактора сигналів StmEd, який дозволяє створити модель унікального сигналу з метою оцінювання реакції моделі спроектованої схеми на дію цього сигналу. За допомогою модуля Parts інженер-проектувальник може створити нову або скорегувати коефіцієнти існуючої математичної моделі компонентів, які використовуються в процедурах моделювання схеми. Модуль Probe дозволяє відображати у вигляді графіків результати деяких



директив моделювання – аналізу перехідних процесів, аналізу частотних характеристик схеми, багатоваріантного аналізу, тобто цей модуль функціонує як віртуальний осцилограф. Особливістю Design Lab є наявність в його структурі PLSyn – вбудованого модуля синтезу програмованих логічних структур на основі виконаної у редакторі Schematics принципової схеми. Зазвичай, інші пакети АСхП не забезпечують таку можливість. Сучасний стрімкий розвиток цієї галузі мікроелектроніки викликав появу окремих спеціалізованих програмних середовищ синтезу програмованих логічних інтегральних схем (ПЛИС). Також окремо слід виділити особливості проектування НВЧ пристроїв. Серед програмних середовищ автоматизації проектування НВЧ-схем можна виділити Microwave Office, хоча ця програма за своєю структурою не є середовищем кризного проектування.

Подібну структуру, тобто орієнтовно схожий склад програмних модулів мають всі програмні середовища АСхП. Відміна полягає в побудові екранного інтерфейсу, розширенні файлів, компонуванні модулів та деяких специфічних програмних особливостях. Таким чином можна стверджувати, що при набутті навичок роботи із одним з пакетів АСхП студент буде здатен самостійно, у відносно стислі строки вивчити особливості побудови іншого середовища АСхП, або при виникненні певних окремих складних питань вдаватися до допомоги спеціалізованих курсів із підвищення кваліфікації з цього фаху. Особливо підвищення кваліфікації актуально, враховуючі постійну зміну версій та нових програм АСхП. Але є певний базис знань з дисципліни “Автоматизація схемотехнічного проектування”, який потрібно вивчити та зрозуміти. Маючи цей базис та навички роботи, майбутній спеціаліст здатен вдосконалювати свою майстерність у творчому процесі проектування, шляхом постійної та методичної роботи.

## 1.2 Узагальнена методологія проектування за допомогою програм АСхП

Слід навести базову методологію проектування, якої варто дотримуватися при проектуванні пристроїв за допомогою сучасних

програмних середовищ автоматизації схемо технічного проектування.

Відповідно до технології проектування розрізняють такі основні стадії проектування [ ]:

I – формулювання завдання. Визначення напрямку пошуку рішення;

II – вибір і оптимізація шляхи рішення завдання;

III – інженерний синтез (моделювання й оптимізація).

Ескізне проектування. Оцінка проекту, ухвалення рішення.

IV – технічне проектування. Виготовлення робочої документації.

V – перевірка й обґрунтування рішення шляхом випробування

експериментальних зразків.

Розглянемо детальний граф процесу проектування. Згідно рис. 1.2. процес проектування містить 24 проектні процедури, які виконують на відповідних стадіях. Тоді процес проектування містить такі процедури: 1 - визначення потреби в об'єкті, дослідження й прогнозування умов функціонування проектованого об'єкта; 2 - загальне й часткове формулювання завдання, пошук її рішення; 3 - аналіз альтернативних рішень завдання, оптимізація обраного рішення; 4 - опис функціонування об'єкта, розробка його структури; 5 - обґрунтування й вибір елементної бази й комплектуючих виробів, оцінка матеріальних, енергетичних і трудових витрат на створення об'єкта; 6 - розробка економічних показників, визначення ефективності; 7 - ескізне проектування; 8 - відпрацювання ескізного проекту на моделях або експериментальних зразках; 9 - захист ескізного проекту, ухвалення рішення щодо проекту; 10 - детальне проектування окремих блоків, вузлів і елементів; 11 - загальне компонування проекту, відпрацювання структурних, конструкторських і технологічних рішень; 12 - виготовлення й перевірка комплексу технічної документації; 13 - патентний аналіз окремих розробок; 14 - оцінка проекту, узгодження технічних завдань із виконавцями; 15 - складання калькуляційної вартості реалізації проекту; 16 - захист технічного проекту з комплектом технічної документації, ухвалення рішення щодо реалізації проекту; 17 - створення моделі об'єкта, побудова дослідного зразка; 18 - розробка програми випробувань дослідного зразка; 19 - виконання випробувань

дослідного зразка; 20 - документальне оформлення результатів випробувань і досліджень; 21 - аналіз результатів випробувань, уточнення характеристик об'єкта, повторне конструювання слабких ланок і корегування технічної документації; 22 - розробка робочої документації, технічних описів і інструкцій для експлуатації; 23 - оформлення патентних або заявочних матеріалів; 24 - остаточне затвердження проекту.

Звісно досягти рівня повної інтеграції процедур проектування в межах одного програмного середовища АСхП дуже складно та навіть не потрібно. Адже деякі процедури не підлягають автоматизації.

#### Рисунок 1.2 – Граф процесу проектування

При аналізі сутності процесу проектування стає помітним рекурсивність деяких процедур проектування. 21 проектна процедура передбачає аналіз результатів випробувань дослідного зразка й на його основі виявлення слабких ланок проекту. При виявленні цих слабких ланок іноді доводилося повертатися до початкових позицій, тобто розробок нових варіантів схем. При виконанні автоматизованого проектування у конструкторів є можливість шляхом створення адекватної імітаційної моделі схеми пристрою та всебічного автоматизованого аналізу її функціонування при різних збурюючих дій зменшити ймовірність помилки проекту та виготовлення дослідного зразка з хибними функціональними властивостями. Автоматизація аналізу схеми дає можливість швидко виявляти недоліки її структури або алгоритму функціонування. Таким чином використання технології автоматизації схмотехнічного проектування дозволяє зменшити час розробки й відповідно заощадити кошти на перепроєктування.

Протягом останніх десяти – п'ятнадцяти років сформувалися певні узагальнені рекомендації щодо виконання проекту за допомогою програм АСхП. Загальні правила автоматизованого проектування досить прості. Необхідно усвідомити, що проектування електронних пристроїв з використанням програмних модулів схмотехнічного проектування містить у собі кілька етапів. С певними доповненнями та корективами, по зрівнянню з етапи, наведеними в [ ], варто виділити такі етапи автоматизації схмотехнічного проектування:

- визначення частини завдання на виконання проекту, яке підлягає автоматизації;
- виконання автоматизованого розрахунку вузлів схеми;
- аналіз розробленої схеми, розкладання її на функціональні вузли й вибір припущень, що спрощують;
- побудова імітаційної моделі принципової схеми (або її частин) проектного пристрою з урахуванням спрощуючих припущень;
- проведення схемотехнічного моделювання побудованої моделі й аналіз отриманих результатів;
- максимально можливе наближення моделі до схеми спроектованого пристрою, одержання остаточних результатів і їхній аналіз;
- побудова монтажно-ї схеми друкованої плати пристрою, автоматичне трасування, виконання перевірки впливу паразитних параметрів плати на цілісність сигналів;
- підготовка комплексу конструкторської документації (за усіма типами проектів – ескізним, технічним, робочим).

Розглянемо ці етапи докладніше.

**Визначення частини завдання на виконання проекту, яке підлягає автоматизації.** Цей етап може починатися із четвертої проектно-ї процедури, згідно якої виконують, зокрема, розробку структури, тобто структурної та функціональної схем розробки. Деякі програми АСхП дозволяють створювати структурні та принципові схеми й навіть виконувати їх структурно-функціональне моделювання. Таким чином, можливо починати роботу над проектом одразу після виконання суто творчих перших трьох процедур.

**Виконання автоматизованого розрахунку вузлів схеми.** Після визначення структури і, якщо дозволяють можливості пакету її моделювання, розроблюють ескіз принципової схеми и виконують інженерні розрахунки елементів схеми. Існують декілька простих у використанні програм для виконання автоматизованого розрахунку елементів схем які відносять до безкоштовно розповсюдженого програмного забезпечення. Наприклад у Design Lab передбачено інтерфейс із програмою

розрахунку активних та пасивних фільтрів Filter Designer. Для роботи із цією програмою потрібно обрати тип та потрібну структуру фільтру, вказати необхідні вхідні та бажані вихідні параметри – полосу частот, імпеданси. Після цього програма виконує автоматичний розрахунок фільтру та відображає його прикінцеву структуру та значення номіналів елементів фільтру. Аналогічні програми розрахунків існують для схем різних типів підсилювачів та ін. Якщо ж такі програми відсутні, то можна виконати інженерний розрахунок елементів у приблизному варіанті.

**Аналіз розробленої схеми, розкладання її на функціональні вузли й вибір припущень, що спрощують.**

Більшість електронних пристроїв занадто складні для безпосереднього аналізу. Якщо в якості моделі використовувати повну принципову схему, час моделювання стає невиправдано великим або таке моделювання не вдається виконати зовсім. Однак аналіз будь-якої схеми показує, що вона складається з основних і допоміжних функціональних вузлів. Допоміжні вузли забезпечують задані режими роботи основних вузлів, і моделювання їхньої роботи недоцільно (принаймні на першому етапі). До них відносять ланцюги живлення, джерела струму й напруги зсуву, генератори що задають й т.п. Як правило, всі ці вузли цілком доцільно замінити стандартними моделями пакету АСхП [ ].

Найбільш характерні приклади спрощених моделей: джерело живлення схеми замінюють на модель ідеального джерела постійної (змінної) напруги; джерело напруги зсуву, в якості якого у схемі застосовують наприклад, стабілітрон замінюють також на модель ідеального джерела постійної напруги; генератор що задає, або генератор несучої замінюють на відповідну модель ідеального джерела синусоїдальної (пилкоподібної, прямокутної) напруги; мережевий трансформатор замінюють на модель ідеального джерела синусоїдальної напруги; варикап замінюють на ємність; транзисторний ключ замінюють на модель керованого напругою (або) струмом джерела напруги/струму.

**Побудова імітаційної моделі принципової схеми проєктованого пристрою з урахуванням спрощуючих припущень.** Виконують реалізацію схеми у модулі редактора схем середовища АСхП з урахуванням вже виконаних на попередньому етапі спрощень. Виконують розташування моделей компонентів, їх поєднання згідно ескізу принципової схеми, зазначення потрібних

номіналів та інших атрибутів. Побудована у редакторі схем модель буде базовою для декількох супутніх модулів пакету АСхП. Цей етап складається із суто механічних дій.

**Проведення схемотехнічного моделювання побудованої моделі й аналіз отриманих результатів.** Схемотехнічне моделювання електронного пристрою передбачає, що схему пристрою попередньо розроблено, проведений розрахунок його компонентів інженерними засобами та створено файл імітаційної моделі схеми. Тому в завдання моделювання можуть входити:

підтвердження правильності проведених інженерних розрахунків і перевірка працездатності пристрою;

дослідження чутливості до розкиду параметрів компонентів;

дослідження нестационарних і аварійних режимів роботи;

дослідження температурної нестабільності пристрою;

визначення шумових властивостей схеми;

підбір коригувальних ланцюгів.

Оскільки моделювання у середовищі АСхП є автоматичним, то перед користувачем цього пакету стоїть завдання зазначення потрібних директив моделювання, а головне коректного завдання параметрів директив моделювання того, чи іншого процесу, який відбувається у пристрої. При невірних або надлишкових значеннях початкових параметрів внутрішній компілятор середовища виведе повідомлення про наявність помилки моделювання, а у гіршому випадку відбудеться “підвисання” усього ПК.

Крім того потрібно застерегти розробників від традиційних помилок. Розповсюдженою помилкою є побудова відразу повної моделі пристрою. Якщо модельований пристрій досить складний, то для побудови працездатної моделі доцільно застосовувати метод поблочного налаштування, який використовують для налагодження реальних електронних пристроїв. Суть його полягає в тому, що спочатку досягають працездатності окремих вузлів і лише потім поєднують їх разом. Наприклад, при аналізі підсилювача потужності доцільно спочатку промодельовати вхідний каскад на ОП (не забувши відповідним чином замкнути зворотний зв'язок), потім приєднати вихідні каскади, підібрати напругу зсуву цих каскадів і лише потім сформуванати загальний зворотний зв'язок і додати ланцюги термостабілізації, корекції й захисту по струму.

Зневага цим правилом іноді сильно утрудняє одержання працездатної моделі.

Крім того, не варто забувати, що пошук моделей конкретних компонентів (наприклад, точної моделі якого-небудь транзистора, використовуваного в реальній схемі) у переважній більшості випадків є недоцільним. Завдання в стандартній моделі того ж транзистора основних довідкових параметрів практично гарантовано дає цілком прийнятний результат (якщо, звичайно, метою моделювання не є дослідження поведінки конкретного транзистора в даній схемі) [ ].

Взагалі ж для первинного аналізу доцільно застосовувати базові моделі компонентів. Але в той же час необхідно знати їхні особливості. Наприклад, відсутність насичення в найпростішій моделі операційного підсилювача приведе до непрацездатності ряду схем, у яких використовується саме цей режим роботи компонента.

Після отримання позитивних результатів моделювання окремих блоків розроблюваного пристрою можна починати поєднання блоків та виконання моделювання уже зібраного пристрою. Потім можна замінювати елементи схеми, які представлені ідеальними джерелами сигналів на моделі реальних компонентів. На цьому ж етапі проводиться точний підбір ланцюгів зсуву й корекції, а також статистичний аналіз і визначення чутливості до параметрів компонентів.

**Максимально можливе наближення моделі до схеми спроектованого пристрою, одержання остаточних результатів і їхній аналіз.** На цьому етапі проводять остаточний розрахунок по скорегованій моделі, одержують всі необхідні характеристики й на основі їхнього аналізу роблять остаточні висновки про працездатність моделі. На основі остаточного аналізу результатів моделювання можливо виконати корегування схеми пристрою.

**Побудова монтажної схеми друкованої плати пристрою, автоматичне трасування, виконання перевірки впливу паразитних параметрів плати на цілісність сигналів.** Для створення файлу друкованої плати потрібен максимально правдоподібна, прикінцево скорегована модель принципової схеми пристрою. В редакторі друкованих плат інженер виконує такі дії: визначає геометричні розміри друкованої плати пристрою,

розміщує моделі корпусів компонентів схеми по поверхні друкованої плати, виконую трасування схеми. При проектуванні високочастотних схем можливо виконання перевірки проекту печатної плати на наявність паразитних ефектів та урахування їх впливу на роботу схеми. На основі результатів такого моделювання може виникнути необхідність перепроєктування проекту плати з метою усунення негативного впливу паразитних ефектів на сигнальні ланцюги пристрою. Програмою курсу “Автоматизація схемотехнічного проектування” не передбачено вивчення особливостей автоматизованого проектування друкованих плат. Тому в даному підручнику цей етап не буде описаний. Проектування друкованих плат в модулі PCBboards пакету Design Lab та його взаємодія із автоматичним трасувальником Spectsra добре описано в [ ] і може бути предметом самостійного вивчення.

**Підготовка комплекту конструкторської документації.** Зазвичай сучасні програмні середовища крізного проектування є продуктом іноземного виробництва і відповідно, вони не оформлюють конструкторську документацію згідно діючою в Україні та СНД системою ЄСКД. Але, наприклад, в Design Lab є можливість створення бібліотеки графічних символів елементів користувача, що надає можливість створення умовно-графічних зображень елементів схем за ЄСКД і, відповідно, підготовки пакету конструкторської документації. Текстову частину конструкторської документації готують за допомогою окремих текстових редакторів.

### 1.2.1 Класифікація моделей та бібліотек моделей компонентів, застосовуваних у програмах EDA

Для опису компонента в програмних середовищах EDA застосовують декілька різновидів моделей. Умовно ці моделі можна розділити на дві категорії : математичні та графічні. Призначення математичних моделей – імітація фізичних процесів, які відбуваються у реальних електронних приладах. Для складних компонентів (напівпровідникових структур) ці моделі побудовані як математичний опис еквівалентної схеми. Наприклад, для опису звичайного біполярного транзистора використовують математичні моделі двох типів – Еберса-Мола та Гумеля-Пуна. Внутрішня



програмна структура математичних моделей компонентів є недоступною для редагування в межах середовища АСхП, хоча, наприклад в Design Lab передбачено інтерфейс із додатковим програмним модулем Device Equations, який в звичайний набір цього програмного пакету не входить. Завдяки Device Equations, шляхом редагування програмних заготовок для збірки програми моделювання Pspice інженер-програміст або проектувальник має можливість вносити зміни у топологію еквівалентних схем напівпровідникових схем заміщення, змінювати структуру рівнянь моделей, додавати нові параметри, змінювати імена параметрів та ін. Після необхідних змін програмного коду, виконують компіляцію оновленої версії Pspice для виконання потрібних рівнів моделювання. Звісно подібну операцію може робити тільки досвідчений інженер-проектувальник, який маючи глибокі теоретичні знання в галузі електроніки та схемотехнічного моделювання та практичні знання мови програмування С++ може вносити необхідні зміни. Також цей інструмент потрібен вченим, які спеціалізуються у створенні нових електронних компонентів на підприємствах–розробниках електронної апаратури.

Звичайному користувачеві середовища АСхП для виконання проектувальних робіт цілком достатньо існуючих моделей. За потреби він може скористуватися стандартним апаратом редагування – шляхом редагування атрибутів (числових параметрів) моделей компонентів для виконання одноразових досліджень. Також можна у подальшому використовувати ці зміни, для цього потрібно зберегти їх у новому файлі моделі або бібліотеки моделей. В середовищі Design Lab можна використати модуль редагування моделей Part, який дозволяє одночасно із зміною числових коефіцієнтів моделей спостерігати вплив цих змін на поведінку тої ці іншої характеристики редагуемого пристрою.

Для характеристики графічних властивостей компонента використовують декілька графічних моделей. По-перше це графіка його умовно-графічного зображення, за допомогою якої він стає візуально доступним при витяганні його на поле проекту із бібліотеки компонентів. В більшості сучасних професіональних середовищ АСхП модуль редактора схем функціонує у двох режимах – режимі власне редактора схем та режимі редактора графіки компонентів. За допомогою цього другого режиму проектувальнику надано можливість створення графіки компонентів згідно вимог національних стандартів (наприклад за ЄСКД для України та СНД). Звісно для відображення конкретного

компоненту, наприклад біполярного транзистору достатньо двох зображень (npn- та pnp- транзистора відповідно), але для вказання **конкретного** транзистора із цією графікою асоціюють його і'мя, і'мя його математичної моделі та деякі інші атрибути.

Крім того, при створенні графіки компоненту в полі відповідного атрибуту необхідно вказати і'мя графіки пакування цього компоненту – тобто графічного зображення корпусу компоненту у якому розміщують цей компонент. Без зазначення пакування (package) при спробі створення проекту монтажної схеми друкованої плати програмою буде виведено повідомлення про помилку, тому що модуль редактора друкованих плат не буде знаходити графічний символ якого корпусу потрібно виводити на поле монтажної схеми. Тому друга графічна бібліотека, яку застосовують разом із бібліотекою умовно-графічних зображень – це бібліотека корпусів компонентів (footprint library).

Інша графічна бібліотека – це бібліотека контактних площадок, яку подекуди називають стекком контактних площадок. Контактна площадка – це геометричне місто контакту виводу компонента із контактним шаром друкованої плати. Ця бібліотека містить опис контактної площадки (код форми, зовнішній розмір, внутрішній розмір), графіка якої буде відображена на проекті монтажної схеми при виведенні на поле проекту корпусу того, чи іншого компоненту.

Для деяких компоненти схеми можуть бути відсутні корпуси. Це стосується компонентів які використовують тільки для схемотехнічного моделювання проекту. Зазвичай це всі компоненти із бібліотек сигналів, які не виводять на друковану плату.

1.2.2. Відомості про еталонну мову схемотехнічного проектування Spice

Оскільки командний файл у формат Spice окрім опису моделей та директив моделювання описує ще й топологію схеми, це дає підстави вважати мову Spice саме мовою **проектування**. Крім того, на основі результатів моделювання відбувається корекція топології як схеми так і проекту друкованої плати.

SPICE (Simulation Program with Integrated Circuit Emphasis) – імітатор електронних схем загального призначення із відкритим вихідним кодом. Spice використовують як при розробці інтегральних схем так і друкованих плат для перевірки цілісності сигналів схем а також для аналізу її поведінки. Програму SPICE

було розроблено в Electronics Research Laboratory в Каліфорнійському університеті в Берклі Лоуренсом Найгелем та його науковим керівником Дональдом Педерсоном. Починаючи із ранішніх версій SPICE було програмним забезпеченням із відкритими вихідними кодами, що сприяло його широкому розповсюдженню та застосуванню. Вихідні коди SPICE розповсюджувалися університетом Берклі за номінальною вартістю (за ціною магнітної стрічки). Тому SPICE швидко стала індустріальним стандартом імітації електронних схем [Pescovitz, David. 1972: The release of SPICE, still the industry standard tool for integrated circuit design, Lab Notes: Research from the Berkeley College of Engineering]. Програма SPICE слугувала основою для розробки багатьох інших програм імітації схем, як в академічному середовищі та і у промисловості. Комерційна версія PSPICE, яка застосована у таких середовищах, як Desig Lab, Orcad належить зараз фірмі Cadence Design Systems, версія HSPICE належить фірмі Synopsys. Академічна версія XSPICE із підтримкою аналого-цифрових пристроїв, яка застосована у Multisim належить Georgia Tech. Деякі відомі виробники електронних пристроїв використовують власні пропріетарні версії SPICE. Наприклад, ADICE у компанії Analog Devices, LTspice у Linear Technology, Mica у Freescale Semiconductor, TISPACE у Texas Instruments. У 2011 році було відмічено, що SPICE та його похідні стали невід'ємною частиною майже будь-якої інтегральної схеми. SPICE набув популярності, оскільки, крім відкритості кодів підтримував аналіз та містив моделі, необхідні для розробки інтегральних схем того часу і при цьому був достатньо швидкодіючим, що було позитивним фактором при практичному використанні. При використанні словосполучення «SPICE- подібні засоби схемотехнічного моделювання» використовують, коли потрібно підкреслити що під час моделювання не використовують спрощуючих припущень, які знижують вірогідність результату, як у системах “прискореного” моделювання. Слід зазначити, що у даному курсі передбачено використання програми SPICE для навчання проектуванню електронних пристроїв на друкованих платах. Окремо буде розглянуто проектування ПЛІС, яке має відміни та при якому SPICE не використовуватиметься.

Перевагою формату SPICE є те, що принципову схему електронного пристрою або його частини можна описати за допомогою коротких та цілком зрозумілих позначень. Формат потребує спершу опису елементів та їх зв'язків, а потім зазначення

директив моделювання. Елементи описують за такою методикою: спочатку вказують позиційне позначення елемента, при цьому для того, щоб програма SPICE “розпізнала” саме яку модель застосовувати при виконанні моделювання для цього елемента це позиційне позначення повинне починатися з певної префіксної літери. Так, позиційне позначення резистора повинно починатися із літери R, конденсатора – літери C, індуктивності – літери L, біполярного транзистора – літери Q, польового транзистора із керованим р-n перетином – літери J, мікросхеми – літери U. Після префіксної літери вказують без пробілу або цифру, яка вказує порядковий номер елемента або можна вказувати іншу літеру. Після позиційного позначення елемента через пробіл вказують номери (або імена) електричних ланцюгів до яких ці елементи під’єднані. Літери замість цифр елементам або електричним ланцюгам надають якщо необхідно виділити цей елемент або ланцюг з метою зручності трактовки результатів моделювання. Після зазначення вузлів, до яких підключений елемент вказують номінал елемента для пасивних компонентів, або і’мя моделі для активних компонентів. Також в частині опису схеми зазначають джерела сигналів, джерела живлення та інші елементи.

Після секції опису схеми вказують моделі компонентів які потрібно використовувати при моделюванні, а також ідентифікатори директив моделювання і виведення результатів моделювання. Директива моделювання – це різновид псевдокоманди для ядра програми SPICE. Перед ідентифікатором кожної моделі компонента та кожної директиви моделювання обов’язково проставляють крапку. Для компілятора програми SPICE крапка є ознакою того, після неї йде псевдокоманда на проведення певного виду схемотехнічного моделювання. Після ідентифікатора директиви через пробіл зазвичай вказують числові параметри – вхідні змінні або початкові значення для вбудованого математичного апарату ядра програми. Наприклад директива на виконання спектрального аналізу може мати такий вигляд – .FOUR 36MEG V(9). Всі елементи завдання у форматі SPICE записують латинськими літерами. Кириличні символи можна використовувати лише у коментарях. Рядок коментарю потрібно починати із символу “зірочка” – \*. В кінці файлу завдання у форматі SPICE вказують директиву .END. Крапку в кінці рядка опису елемента схеми або рядка опису директиви не проставляють. Компілятор програми ідентифікує кожен нову строку як новий опис.

Розглянемо схему транзисторного мультивібратора та завдання на виконання його імітаційного моделювання у форматі PSpICE для середовища Design Lab 8.0. Мультивібратор це пристрій, який генерує прямокутні коливання.

Рисунок 1.3 – Схема транзисторного мультивібратора

На рис.1.3 в кружках зазначено номери вузлів – тобто еквіпотенціальних електричних ланцюгів схеми. Для даної схеми потрібно провести автоматизований аналіз перехідних процесів. Тоді файл опису схеми та директив моделювання виглядатиме таким чином

```

Multivibrator
R1 1 2 1K
R2 5 0 750
R3 1 3 100K
R4 1 4 100K
R5 1 6 1K
R6 7 0 750
C1 2 3 0.01uF
C2 4 6 0.01uF
Q1 2 4 5 2N2218
Q2 6 3 7 2N2218
VP q1 0 DC 10V
.LIB bipolar.lib
.IC V(3)=3.46V
.TRAN 100us 1.2ms
.PROBE
.END

```

Перша строка файла опису за умовчанням вважається його ім'ям. В пакетах крізного проектування усі файли одного проекту мають одне ім'я та різні розширення. Звернемо увагу на опис напівпровідникових елементів схеми – транзисторів. При опису біполярних транзисторів після вказання позиційного позначення вказують на першому місці номер вузла до якого підключений колектор, на другому – номер вузла до якого підключена база, на третьому – номер вузла до якого підключений емітер. Цю черговість потрібно враховувати. Після номерів вузлів вказують

ім'я моделі транзистора. В кінці опису елементів вказують джерело живлення, яке необхідно для роботи схеми – це джерело постійної напруги 10В з ім'ям VP, яке підключено до вузлів 1 (позитивний затискач) та 0 (негативний затискач).

Далі вказують директиви. Директива LIB вказує компілятору SPICE в якому файлі бібліотек шукати вказану в описі модель транзистора. Якщо ця бібліотека знаходиться не в стандартній папці бібліотека то в цієї директиві можна прописати її повний шлях (адресу) звичайним способом.

Хоча в описі не вказують, але програма SPICE автоматично виконує розрахунок схеми за постійним струмом, результат якого записують у вихідний файл результатів моделювання. Оскільки мультівібратор є симетричним пристроєм, то за постійним струмом він знаходиться у стані рівноваги. Тобто потенціал вузла 2 та потенціал вузла 6 будуть рівними. Якщо виконувати директиву аналізу перехідних процесів одразу після розрахунку за постійним струмом, то генератор, який знаходиться у стані рівноваги не збудеться і відповідно на екрані програми відображення графіки буде пряма лінія. Тому є два способи зрушити його рівновагу – перший передбачає зміну на 1-2% значення резистора або ємності у колі якогось з транзисторів або завдання початкового значення потенціалу одного з вузлів. Ми обрали другий шлях. Завдамо початкові умови (initial conditions) для вузла 3. Після виконання розрахунку за постійним струмом проглянемо вихідний файл результатів моделювання та відзначимо значення потенціалу вузла 3 що дорівнює 3,4889 В. Тоді в директиві завдання початкових умов для вузла 3 вкажемо трохи інше значення – .IC V(3)=3.46V. Після цього вкажемо директиву аналізу перехідних процесів TRAN із указанням початкового та фінального часу моделювання. Директива .PROBE вказує компілятору на підключення графічного редактора результатів моделювання після закінчення обчислень модуля PSPICE.

Таким чином можна виконати моделювання нескладної принципової схеми, яка накреслена на папері навіть лише якщо ми маємо тільки модуль SPICE та простий моношрифтовий текстовий редактор. Але звісно, якщо ведеться робота над складним проектом, який містить десятки, сотні елементів, якщо потрібно проаналізувати декілька складових характеристик створюваного пристрою, створити монтажну схему друкованої плати, тоді просто необхідно мати повноцінне середовище схемотехнічного проектування.

### 1.3. Особливості автоматизованого проектування схем із пасивними базовими дискретними компонентами

Дискретними компонентами в програмах АСхП називають компоненти які мають два контакти, тобто будь-які двох полюсники. Базовими пасивними компонентами ми будемо вважати елементи класичної теорії електричних кіл – резистор, конденсатор та індуктивність.

Одразу слід зазначити, що у програмах АСхП використовують різні моделі цих компонентів. З навчальних дисциплін, які присвячені основам електротехніки та електроніки ми знаємо, що ці елементи можуть бути як лінійними так нелінійними і відповідно для відтворення їх поведінки слід застосовувати різні моделі. Застосування нелінійних моделей наближує імітаційну схем розроблюваного пристрою до відтворення реальних процесів, але ускладнює обчислювальний процес внаслідок втрат часу та ресурсів машинної пам'яті.

В програмах АСхП застосовують математичні вирази, які за допомогою спеціальних коефіцієнтів дозволяють апроксимувати характеристики елементів у моделі. Інженеру потрібно лише зазначити необхідні значення цих коефіцієнтів, які подекуди називають **атрибутами** або **параметрами моделі**, в діалогових вікнах екранного інтерфейсу. Тому слід розглянути вирази, які відтворюють характеристики елементів із тлумаченням використовуваних у них коефіцієнтів.

Опір елемента “Резистор” у програмах АСхП обчислюють з урахуванням уведених значень параметрів моделі за допомогою двох виразів:

– якщо вказано експоненційний температурний коефіцієнт опору (TCE) то вираз для обчислення опору моделі такий :

$$\text{значення опору} = \langle \text{value} \rangle \cdot R \cdot 1.01^{TCE \cdot (T - T_{nom})}$$

де  $\langle \text{value} \rangle$  – позитивне значення опору (може мати також негативне значення, але не нульове), яке зазначив інженер-розробник у діалоговому вікні атрибутів конкретного резистора у схемі. Наведемо приклад діалогових вікон з різних програмних середовищ АСхП (рис.1.4, рис1.5.) в яких вказують значення опору резистору;

$T$  – поточне значення температури для конкретної схеми, яке вказують коли у середовищі АСхП присутній програмний

модуль розширеного температурного моделювання (у тому числі за списком), значення цього параметру повинно відрізнятися від  $T_{nom}$ ;

$T_{nom}$  – номінальне значення температури, це глобальний параметр, який за умовчанням дорівнює  $27^{\circ}\text{C}$ , його зазвичай встановлюють в розділах глобального налаштування середовища АСхП.

– якщо не вказано експоненційний температурний коефіцієнт опору, то вираз для обчислення опору моделі буде виглядати інакше:

$$\text{значення опору} = \langle \text{value} \rangle \cdot R \cdot (1 + TC1 \cdot (T - T_{nom}) + TC2 \cdot (T - T_{nom})^2)$$

Значення подальших параметрів моделі резистору, які використовують у двох вищевказаних виразах наведено у таблиці 1.1.

Рисунок 1.4 – Фрагмент діалогового вікна із указанням значення резистору у програмі Multisim 7.0 і вищих версій

Рисунок 1.5 – Фрагмент діалогового вікна із указанням значення резистору у програмі Electronic Workbench 5.12

Таблиця 1.1 – Параметри (коефіцієнти) моделі резистора

Параметр	Позначення параметра	Розмірні	
R	Масштабний множник опору		
TC1	Лінійний температурний коефіцієнт опору	$^{\circ}\text{C}^{-1}$	
TC2	Квадратичний температурний коефіцієнт опору	$^{\circ}\text{C}^{-2}$	
TCE	Експоненційний температурний коефіцієнт опору	$\%/^{\circ}\text{C}$	
T_MEASURED	Температура вимірювання	$^{\circ}\text{C}$	



T_ABS	Абсолютна температура	<sup>0</sup> С	
T_REL_GLOBAL	Відносна температура	<sup>0</sup> С	
T_REL_LOCAL	Різниця між температурою пристрою та моделі прототипу	<sup>0</sup> С	

Останні чотири параметри відносять до температури, вони не входять у структуру моделі, але їх використовують у певних середовищах АСхП для зазначення температурних змін у різних видах автоматичного аналізу схеми. Окрему увагу тлумаченню цих параметрів буде приділено при розгляданні директив моделювання.

Шумовий струм резистору при виконанні аналізу рівня шумів обраховують при умові полоси пропускання яка дорівнює 1Гц. Резистор генерує тепловий шум за формулою Найквіста для спектральної щільності енергії (на одиницю полоси пропускання):

$$i^2 = 4kT/resistance$$

Ємність елемента “Конденсатор” у програмах АСхП обчислюють з урахуванням уведених значень параметрів моделі за допомогою виразу:

$$значення\ ємності = \langle value \rangle \cdot C \cdot (1 + VC1 \cdot V + VC2 \cdot V^2) \cdot (1 + TC1 \cdot (T - Tnom) + TC2 \cdot (T - Tnom)^2)$$

де  $\langle value \rangle$  – позитивне значення ємності яке зазначив інженер-розробник у діалоговому вікні атрибутів конкретного резистора у схемі;

$V$  – напруга на конденсаторі при розрахунку перехідних процесів (Transient), при розрахунку частотних характеристик (AC Sweep) ємність вважають постійною величиною, яка визначається у робочій точці по постійному струму; значення параметрів  $T$  та  $Tnom$  розтлумачено вище.

Значення подальших параметрів моделі конденсатора, які використовують у наведеному виразі зазначено у таблиці 1.2

Таблиця 1.2 – Параметри (коефіцієнти) моделі ємності

Параметр	Позначення параметра	Розмірніс	
C	Масштабний множник ємності		
VC1	Лінійний коефіцієнт напруги	V <sup>-1</sup>	

VC2	Квадратичний коефіцієнт напруги	$V^{-2}$	
TC1	Лінійний температурний коефіцієнт ємності	$^{\circ}C^{-1}$	
TC2	Квадратичний температурний коефіцієнт ємності	$^{\circ}C^{-2}$	
T_MEASURED	Температура вимірювання	$^{\circ}C$	
T_ABS	Абсолютна температура	$^{\circ}C$	
T_REL_GLOBAL	Відносна температура	$^{\circ}C$	
T_REL-LOCAL	Різниця між температурою пристрою та моделі прототипу	$^{\circ}C$	

Останні чотири температурних параметри, як і для резистора, не входять у структуру моделі, їх також можна використовувати для зазначення температурних змін у різних видах автоматичного аналізу схеми.

У деяких середовищах АСхП для конденсатора потрібно зазначити початкові умови (initial condition, або скорочено – IC), тобто для ємності це буде значення початкової напруги, яке використовує емулятор SPICE при розрахунку по постійному струму.

Наступний елемент дозволяє імітувати лінійну індуктивність. Значення індуктивності елемента “індуктивність” у програмах АСхП обчислюють з урахуванням уведених значень параметрів моделі за допомогою виразу:

$$\text{значення індуктивності} = \langle \text{value} \rangle \cdot L \cdot (1 + IL1 \cdot I + IL2 \cdot I^2) \cdot (1 + TC1 \cdot (T - T_{nom}) + TC2 \cdot (T - T_{nom})^2)$$

де  $\langle \text{value} \rangle$  – позитивне значення індуктивності яке зазначив інженер-розробник у діалоговому вікні атрибутів конкретної індуктивності у схемі;

$I$  – струм через котушку індуктивності при розрахунку перехідних процесів (Transient), при розрахунку частотних характеристик (AC Sweep) індуктивність вважають постійною величиною, яка визначається у робочій точці по постійному струму;

параметри  $T$  та  $T_{nom}$  такі, як і у випадку із резистором та конденсатором;

Значення подальших параметрів моделі котушки індуктивності, які використовують у наведеному виразі зазначено у таблиці 1.3

Таблиця 1.3 – Параметри (коефіцієнти) моделі індуктивності

Параметр	Позначення параметра	Розмірніс	
L	Масштабний множник індуктивності		
IC1	Лінійний коефіцієнт струму	A <sup>-1</sup>	
IC2	Квадратичний коефіцієнт струму	A <sup>-2</sup>	
TC1	Лінійний температурний коефіцієнт ємності	<sup>0</sup> C <sup>-1</sup>	
TC2	Квадратичний температурний коефіцієнт ємності	<sup>0</sup> C <sup>-2</sup>	
T_MEASURED	Температура вимірювання	<sup>0</sup> C	
T_ABS	Абсолютна температура	<sup>0</sup> C	
T_REL_GLOBAL	Відносна температура	<sup>0</sup> C	
T_REL_LOCAL	Різниця між температурою пристрою та моделі прототипу	<sup>0</sup> C	

Так як і для конденсатора, для індуктивності потрібно зазначити початкові умови, але для котушки це буде не напруга, а струм через неї при розрахунку перехідних процесів.

Якщо потрібно імітувати роботу нелінійної індуктивності, тобто реального елемента – котушки індуктивності, яка існує у вигляді мідного проводу, який намотують на певне осердя, то потрібно використовувати інші моделі. За допомогою котушок індуктивності можливо побудувати пристрої, які функціонують на основі електромагнітного зв'язку між котушками індуктивності, які розташовані на одному осерді. Для цього в програмах АСхП використовують елемент “взаємна індуктивність”, який дозволяє враховувати електромагнітний зв'язок між декількома котушками.

Взаємну індуктивність за різними графічними форматами описують у середовищах EDA. Але параметри опису у форматі SPICE є стандартними. Якщо не вказано ім'я моделі осердя то вираз прийме вигляд:

$K$ <ім'я, або номер>  $L$ <ім'я, або номер першої індуктивності> ...  $L$ < ім'я або номер  $n$ -ї індуктивності > <значення коефіцієнту зв'язку

Вказують таку кількість індуктивностей, яку необхідно зв'язати за допомогою однієї взаємної індуктивності. Значення взаємної індуктивності  $M_{ij}$  двох індуктивностей  $L_i$  та  $L_j$  можна обрахувати після зазначення коефіцієнту зв'язку (англ.- *coupling value* ) за допомогою виразу:

$$\langle \text{coupling value} \rangle = \frac{M_{ij}}{\sqrt{L_i L_j}}.$$

Напругу на котушці  $L_i$  з урахуванням взаємної індуктивності при виконанні аналізу перехідних процесів визначають за виразом

$$U_{L_i} = L_i \frac{dl_i}{dt} + M_{ij} \frac{dl_j}{dt} + M_{ik} \frac{dl_k}{dt} + \dots$$

Наведемо приклад опису взаємної індуктивності такої схеми [ ]

Рисунок 1.6 – Схема трансформатора із трьома обмотками

Формат опису цієї схеми такий:

I1	1	0	AC	1MA
L1	1	0	10UH	
L2	2	3	10UH	
L3	3	4	10UH	
K12	L1	L2	L3	0.8

Перший вузол в описах індуктивностей L1, L2, L3 означає початок обмотки (тобто для індуктивності L1 початок обмотки підключено до вузла 1, для індуктивності L2 початок обмотки підключено до вузла 2, для індуктивності L3 початок обмотки підключено до вузла 3). До того слід зазначити, що наприклад струм скрізь котушку L2 буде мати протилежний напрям ніж скрізь

катушку L1. Також потрібно зазначити, що при відсутності моделі осердя атрибут *value* усіх індуктивностей, які пов'язані взаємною індуктивностей повинен бути зазначений у генрі. Для покращення збіжності методів розрахунку між вузлами до яких підключено початки обмоток всіх катушок підключають резистор величина опору якого дорівнює 1/GMIN (де GMIN – значення глобального параметра мінімальної провідності гілки електричного ланцюга). Ще один спосіб покращення збіжності – підключення резистору паралельно досліджуваній катушці, опір якого є дуже малим, наприклад таким що дорівнює 0,1 Ом.

При проектуванні деяких пристроїв необхідно застосовувати катушки індуктивності, електромагнітний зв'язок між якими відбувається через магнітне осердя. Для опису магнітного осердя в SPICE використовують майже такий самі формат опису як і у випадку взаємної індуктивності, але також додають ім'я моделі осердя, яке потім також вказують із зазначенням необхідних параметрів моделі.

Формат опису електромагнітного зв'язку декількох катушок індуктивності за допомогою магнітного осердя такий:

*K*<імя, або номер> *L*<імя, або номер першої індуктивності> ...  
*L*< імя або номер n-ї індуктивності > <значення коефіцієнту зв'язку > <імя моделі осердя>

Як бачимо, в опису додається ще один компонент – ім'я моделі осердя. Але потрібно пам'ятати, що при зазначенні моделі осердя в опису, в атрибуті *value* індуктивностей які пов'язують за допомогою цього магнітного осердя, потрібно вказувати не значення індуктивностей у генрі, а **кількість витків** проводу кожної катушки. Через це, значення вказане в атрибуті *value* катушки може бути тільки позитивним і тільки цілим. Наведемо приклад схеми трансформатора із магнітним осердям та його опис:

Рисунок 1.7 – Схема трансформатора із магнітним осердям

```
L1 15 10 50
L2 10 0 150
K2 L1 L2 0.99 Ti125V
.MODEL Ti125V CORE (LEVEL=2 MS=334E3
ALPHA=2.5E-2 +A=4.05E3 K=166 C=0.05 AREA=0.064
PATH=2.25)
```

В опису індуктивності L1 цифра 50 означає 50 витків проводу котушки, в опису індуктивності L2 цифра 150 означає 150 витків проводу котушки. Вказано коефіцієнт зв'язку котушок, що дорівнює 0,99, а також модель магнітного осердя Ti125V. В секції атрибутів моделі обов'язково вказують ім'я цієї моделі після вказання директиви MODEL, параметр опису CORE вказує компілятору SPICE що це модель саме осердя. В дужках наведені конкретні значення параметрів моделі осердя. Що означають ці параметри показано у таблиці 1.4.

Таблиця 1.4 – Параметри (коефіцієнти) моделі осердя

LEVEL	індекс моделі		
A	Параметр форми безгістерезисної кривої	A/л	
AREA	Площа поперечного перерізу магнітопроводу	см <sup>2</sup>	
C	Стала пружного зсуву границь доменів		
GAP	Ширина повітряного зазору	см	
K	Стала рухомості доменів	A/л	
MS	Намагнічуваність насичення	A/л	
PACK	Коефіцієнт заповнення осердя		
PATH	Середня довжина магнітної силової лінії	см	

Слід зазначити, що в програмах АСхП при імітації роботи магнітного осердя здебільшого використовують модель осердя, яка була запропонована Джилсом та Аттертоном [1]. Наводити математичні вирази цієї та інших моделей осердя, а також прийняті припущення при їх побудові у даному підручнику не будемо. Модель Джилса та Аттертона має обмеження при використанні у імітації високочастотних пристроїв, оскільки вона не враховує втрати при перемагнічуванні. Відома також модель Чена [2], яка застосована у середовищі OrCAD 9.2 та яка більш придатна для моделювання поведінки осердя при намагнічуванні змінним полем. Але до сьогодні імітаційне моделювання пристроїв із магнітним осердям часто створює проблеми інженерам-проектувальникам. Тому серед вчених та розробників продовжуються пошуки адекватних моделей магнітного осердя. Наприклад ідуть роботи по імплементації до структур модельного апарату програм АСхП моделі Ізінга [3].

Розробники середовищ АСхП створили користувачам їх програмних засобів додаткову опцію моделювання магнітного

осердя. Вони заклали у програмний апарат можливість внесення в атрибути магнітного осердя координати кривої намагнічування, числові дані якої можуть бути отримана, наприклад, експериментальним шляхом або взята із літературних джерел, або з довідникових даних. При використанні цієї опції користувачеві за допомогою екранного інтерфейсу потрібно ввести декілька (від 10 до 15) координат, тобто значень магнітної індукції як функції від значення напруженості магнітного поля. На рис. 1.8 показана екранна форма для зазначення координат магнітного осердя компоненту нелінійний трансформатор програми Electronic Workbench, Multisim.

У середовищі Design Lab інженеру-розробнику окрім зазначених вище опцій надано можливість продивитись графік кривої намагнічування обраного осердя за допомогою графічних засобів програмного модуля Parts.

Рисунок 1.8 – Вікно для редагування координат кривої намагнічування програми EWB

Рисунок 1.9 – Вікно для редагування координат кривої намагнічування програми Multisim

1.4 Особливості автоматизованого проектування схем із активними дискретними компонентами.

До активних дискретних компонентів будемо відносити елементи діодної групи та транзистори.

У програму SPICE вбудована модель напівпровідникового діоду, яку можна представити у вигляді такої еквівалентної схеми рис. 1.10.

Рисунок 1.10 – Еквівалентна схема діоду

На еквівалентній схемі представлено нелінійне залежне джерело струму  $I(V)$  яке підключено паралельно до ємності p-n перетину  $C$  та послідовно підключеного питомого опору  $RS$ . Це звісно узагальнена схема діоду, оскільки ми знаємо, що характеристики напівпровідникового перетину залежать від декількох факторів – полярності підвимої напруги, частоти струму, температури.

Параметри SPICE-моделі діоду наведені у таблиці 1.5 [1]. Потрібно чітко розуміти, що ці параметри імітують поведінку окремих характеристик діоду, тому певна група параметрів моделі входить до рівняння імітації одної характеристики, та не буде міститися у рівнянні для імітації іншої характеристики. Розглянемо, за допомогою яких рівнянь представляють характеристики діоду.

Таблиця 1.5 – Параметри моделі діоду

Параметр	Позначення параметра	Значення умовч	
<b>IS</b>	Струм насичення при температурі 27°C	$10^{-14}$	
<b>RS</b>	Питомий опір	0	
<b>N</b>	Коефіцієнт інжекції	1	
<b>ISR</b>	Параметр струму рекомбінації	0	
<b>NR</b>	Коефіцієнт емісії для струму рекомбінації	2	

продовження таблиці 1.5

<b>IKF</b>	Максимальний струм при високому рівні інжекції	$\infty$	
<b>TT</b>	Час перенесення заряду	0	
<b>CJO</b>	Бар'єрна ємність при нульову зсуві	0	
<b>VJ</b>	Контактна різниця потенціалів	1	
<b>M</b>	Коефіцієнт лавинного множення	0,5	
<b>EG</b>	Ширина забороненої зони	1,11	
<b>FC</b>	Коефіцієнт не лінійності бар'єрної ємності зміщеного у прямому напрямі перетину	0,5	
<b>BV</b>	Зворотна напруга пробою (позитивна величина)	$\infty$	
<b>IBV</b>	початковий струм пробою, що відповідає напрузі BV (позитивна величина)	$10^{-10}$	
<b>NBV</b>	Коефіцієнт не ідеальності на ділянці пробою	1	
<b>IBVL</b>	Початковий струм пробою низького рівня	0	



<b>NBVL</b>	Коефіцієнт не ідеальності на ділянці пробою низького рівня	1	
<b>XTI</b>	Температурний (експоненційний) коефіцієнт струму насичення	3	
<b>TIKF</b>	Лінійний температурний коефіцієнт IKF	0	
<b>TBV1</b>	Лінійний температурний коефіцієнт BV	0	
<b>TBV2</b>	Квадратичний температурний коефіцієнт BV	0	
<b>TRS1</b>	Лінійний температурний коефіцієнт RS	0	
<b>TRS2</b>	Квадратичний температурний коефіцієнт RS	0	
<b>KF</b>	Коефіцієнт фліккер - шуму	0	
<b>AF</b>	Показник ступеню у формулі фліккер-шуму	1	
<b>T_MEASU RED</b>	Температура вимірювань		
<b>T_ABS</b>	Абсолютна температура		
<b>T_RELGL OBAL</b>	Відносна температура		
<b>T_REL_LO CAL</b>	Різниця між температурою діоду та моделі прототипу		

Вольт-амперні характеристики. Струм діоду представлено у вигляді різниці струмів

$$I_d = I_{fwd} - I_{rev}. \quad (1.1)$$

де  $I_{fwd}$  – прямий струм діоду;  $I_{rev}$  – зворотний струм.

Залежність прямого струму

$$I_{fwd} = I_n \cdot K_{inj} + I_{rec} \cdot K_{gen}, \quad (1.2)$$

апроксимує ВАХ діоду при позитивному значенні напруги  $V$  на р-п перетині. Складові рівняння 1.2 обраховують за допомогою виразів: нормальної складової струму

$$I_n = IS \cdot \left( e^{\frac{V}{NR \cdot V_t}} - 1 \right), \quad (1.3)$$

та струму рекомбінації

$$I_{rec} = ISR \cdot \left( e^{\frac{V}{NR \cdot V_t}} - 1 \right) \quad (1.4)$$

де  $V_t = \frac{kT}{q}$  – температурний потенціал перетину, в виразі якого:

$k$  – стала Больцмана, що дорівнює  $1,38 \cdot 10^{-23}$  Дж/°С,  $T$  – поточна температура р-п перетину,  $q$  – заряд електрону, що дорівнює  $1,16 \cdot 10^{-19}$  Кл.

Коефіцієнт інжекції  $K_{inj}$  у виразі (1.2) обчислюють за допомогою

$$K_{inj} = \begin{cases} \sqrt{\frac{IKF}{(IKF + I_n)}} \text{ для } IKF > 0 \\ 1 \text{ для } IKF \leq 0 \end{cases} \quad (1.5)$$

Коефіцієнт генерації  $K_{gen}$  у виразі (1.2) обчислюють за допомогою

$$K_{gen} = \left( \left( 1 - \frac{V}{VJ} \right)^2 + 0,005 \right)^{M/2} \quad (1.6)$$

Значення змінних  $I$ ,  $IS$ ,  $NR$ ,  $IKF$ ,  $VJ$ ,  $M$  з виразів (1.3) – (1.6) наведені у таблиці 1.5 і є параметрами моделі.

Струм діоду при негативному значенні напруги на діоді характеризує явище пробою. Цей струм також складається з двох компонент :

$$I_{rev} = I_{rev.high} - I_{rev.low} \quad (1.7)$$

які обчислюють за допомогою виразів

$$I_{rev.high} = IBV \cdot e^{-\frac{(V+BV)}{(NBV \cdot V_t)}} \quad (1.8)$$

$$I_{rev.low} = IBVL \cdot e^{-\frac{(V+BV)}{(NBVL \cdot V_t)}} \quad (1.9)$$

Значення змінних  $IBV$ ,  $IBVL$ ,  $BV$ ,  $NBV$ ,  $NBVL$ , з виразів (1.7) – (1.9) наведені у таблиці 1.5 і також є параметрами моделі діоду. Ємність перетину обраховують за допомогою виразу

$$C = C_t + Area \cdot C_j \quad (1.10)$$

де дифузійну ємність  $C_t$  знаходять як  $C_t = TT \cdot G$ , а бар'єрну  $C_j$  як:

$$C_j = \begin{cases} CJO \cdot \left(1 - \frac{V}{VJ}\right)^{-M} & \text{и и} \\ CJO \cdot (1 - FC)^{-(1+M)} \cdot \left(1 - FC(1+M) + M \cdot \frac{V}{VJ}\right) & \text{при } V > F \end{cases}$$

де  $G = Area \cdot \frac{dI_{fwd}}{dV}$  – диференційна провідність перетину для поточних значень струму та напруги, множник  $Area$  – який також присутній у формулі (1.10), коефіцієнт кратності, який дозволяє імітувати паралельне включення декількох однакових компонентів, якщо паралельних компонентів у схемі немає, значення  $Area$  дорівнює одиниці.

При виконанні аналізу шумових складових схем із діодами використовують модель із двома джерелами шумового струму. Для цього в моделі наведеній на рис. 1.10 замінюють джерело струму на провідність  $G$  а також підключають паралельно до неї джерело шумового струму  $I$  яке імітуватиме дробовий та фліккер шум діоду та яке має спектральну щільність струму, яку обраховують за виразом

$$S_d = 2qI + KF \cdot \frac{I^{AF}}{f}$$

де  $q$  – заряд електрона,  $I$  – поточне значення струму,  $f$  – поточная частота, параметри  $KF$  та  $AF$  – є параметрами моделі діоду (див. табл.1.5).

Рисунок 1.11 – Модель діоду для імітації шумових складових

Паралельно об'ємному опору  $RS$  також підключено джерело шуму, яке імітує тепловий струм, спектральна щільність якого дорівнює

$$S_d = \frac{4kT}{RS / Area}$$

При зміні значення температурних показників схеми, яке виконують з метою виконання температурного аналізу в моделі діоду передбачено зміну таких температурно-залежних параметрів :

$$IS(T) = IS \cdot e^{\left(\frac{T}{T_{ном}} - 1\right) \cdot \frac{EG}{(N \cdot Vt)}} \cdot \left(\frac{T}{T_{ном}}\right)^{\frac{XTI}{N}};$$

$$ISR(T) = ISR \cdot e^{\left(\frac{T}{T_{ном}} - 1\right) \cdot \frac{EG}{(NR \cdot Vt)}} \cdot \left(\frac{T}{T_{ном}}\right)^{\frac{XTI}{NR}};$$

$$IKF(T) = IKF \cdot (1 + TIKF \cdot (T - T_{ном}));$$

$$BV(T) = BV \cdot (1 + TVB1 \cdot (T - T_{ном}) + TVB2 \cdot (T - T_{ном})^2);$$

$$RS(T) = RS \cdot (1 + TRS1 \cdot (T - T_{ном}) + TRS2 \cdot (T - T_{ном})^2);$$

$$VJ(T) = \frac{VJ \cdot T}{T_{ном}} - 3Vt \cdot \ln\left(\frac{T}{T_{ном}}\right) - \frac{Eg(T_{ном}) \cdot T}{T_{ном}} + Eg(T);$$

$$Eg(T_{ном}) = \begin{cases} 1,11 \text{ eV для Si} \\ 0,67 \text{ eV для Ge} \\ 0,69 \text{ eV для бар'єру Шоттки} \end{cases};$$

$$Eg(T) = Eg_0 - \frac{a \cdot T^2}{b + T} = 1,16 - \frac{0,000702 \cdot T^2}{1108 + T};$$

$$CJO(T) = CJO \cdot \left(1 + M \cdot \left(0,0004 \cdot (T - T_{ном}) + \left(1 - \frac{VJ(T)}{VJ}\right)\right)\right);$$

$$KF(T) = KF \cdot \frac{VJ(T)}{VJ};$$

$$AF(T) = AF \cdot \frac{VJ(T)}{VJ}.$$

Для діодів Шоттки вказують інше значення температурного коефіцієнту струму насичення  $X_{TI}$  яке дорівнюватиме 2. Для діодів із р-n перетином  $X_{TI}=3$ .

Всі елементи діодної групи із р-n перетином (окрім тунельних діодів), включаючи діоди Шоттки, в яких присутній перетин типу метал-напівпровідник описують за допомогою вищевказаних рівнянь. При імітації стабілітрон до моделі діоду додається два параметри:

Параметр	Позначення параметра	Значення умовч	
<b>IZT</b>	номінальний струм стабілізації	$10^{-4}$	
<b>VZT</b>	напруга стабілізації при номінальному струмі стабілізації	1	

#### БПТ транзистор

В програмах автоматизації схемотехнічного проектування використовують схему заміщення біполярного транзистора (БПТ) у вигляді моделі Гумеля—Пуна яка дозволяє враховувати ефекти, що виникають при великих значеннях зсувів на перетинах транзистору. Ця модель автоматично спрощується до моделі Еберса – Молла, якщо усунути деякі параметри. Параметри повної математичної моделі БПТ наведені у табл. 1. На рис. 1.1 наведено схему моделі БПТ Гумеля – Пуна, а на рис. 1.1 схему моделі Еберса – Молла.

Таблиця 1. – Параметри SPICE-моделі БПТ

Параметр	Позначення параметра	Значення умовч	
AF	Показчик ступеня, що визначає залежність спектральної щільності флікер-шуму від струму через р-n перетин	1	
BF	Максимальний коефіцієнт підсилення за струмом у нормальному режимі у схемі з загальним емітером (без урахування струмів витікання)	100	
BR	Максимальний коефіцієнт підсилення за струмом у	1	

	інверсному режимі у схемі з загальним емітером		
CJC	Ємність колекторного перетину при нульовому значенні напруги на ньому	0	
CJE	Ємність емітерного перетину при нульовому значенні напруги на ньому	0	
CJS (CCS)*	Ємність перетину колектор - підкладка при нульовому значенні напруги на ньому	0	
EG	ширина забороненої зони	1.11	
FC	Коефіцієнт нелінійності бар'єрних ємностей зміщених у прямому напрямі перетинів	0,5	
GAMM A	Коефіцієнт легування епітаксильної області	$10^{-11}$	
IKF (IK)*	Струм початку спаду залежності BF від струму колектору у нормальному режимі	$\infty$	
IKR*	Струм початку спаду залежності BR від струму емітеру у інверсному режимі	$\infty$	
IRB*	Струм бази, при якому опір бази зменшується наполовину повного перепаду між RB і RBM	$\infty$	
IS	Струм насичення при температурі 27°C	$10^{-16}$	
ISC (C4)*	Струм насичення витікання перетину база-колектор	0	
ISE (C2)*	Струм насичення витікання перетину база-емітер	0	
ISS	Зворотний струм p-n перетину підкладки	0	
ITF	Струм, який характеризує залежність TF від струму колектора при великих струмах	0	
KF	Коефіцієнт, що визначає спектральну щільність флікер-шуму	0	
MJC (MC)*	Коефіцієнт плавності колекторного перетину	0.33	
MJE (ME)*	Коефіцієнт плавності емітерного перетину	0.33	
MJS (MS)*	Коефіцієнт плавності перетину колектор-підкладка	0	
NC*	Коефіцієнт неідеальності колекторного перетину	2.0	
NE*	Коефіцієнт неідеальності емітерного перетину	1.5	
NF	Коефіцієнт не ідеальності у нормальному режимі	1.0	
NK	Коефіцієнт, що визначає множник $Q_b$	0.5	
NR	Коефіцієнт не ідеальності в інверсному режимі	1.0	
NS	Коефіцієнт не ідеальності перетину підкладки	1.0	
PTF	Додатковий фазовий зсув на граничній частоті	0	

	транзистора $f_{cp} = 1/(2\pi \cdot TF)$		
<b>QCO</b>	Множник, що визначає заряд в епітаксильній області	0	
<b>RB</b>	Питомий опір бази (макс.) при нульовому значенні напруги на перетині база-емітер	0	
<b>RBM*</b>	Мінімальний опір бази при великих струмах	0	
<b>RC</b>	Питомий опір колектора	0	
<b>RCO</b>	Опір епітаксильної області	0	
<b>RE</b>	Питомий опір емітера	0	
<b>TF</b>	Час переносу заряду через базу в нормальному режимі	0	
<b>TR</b>	Час переносу заряду через базу в інверсному режимі	0	
<b>TRB1</b>	Лінійний температурний коефіцієнт RB	0	
<b>TRB2</b>	Квадратичний температурний коефіцієнт RB	0	
<b>TRC1</b>	Лінійний температурний коефіцієнт RC	0	
<b>TRC2</b>	Квадратичний температурний коефіцієнт RC	0	
<b>TRE1</b>	Лінійний температурний коефіцієнт RE	0	
<b>TRE2</b>	Квадратичний температурний коефіцієнт RE	0	
<b>TRM1</b>	Лінійний температурний коефіцієнт RBM	0	
<b>TRM2</b>	Квадратичний температурний коефіцієнт RBM	0	
<b>T_ABS</b>	Абсолютна температура		
<b>T_MEASURED</b>	Виміряна температура		
<b>T_REL_GLOBAL</b>	Відносна до струму температура		
<b>T_REL_LOCAL</b>	Різниця між температурою транзистора і моделі прототипу (АКО)		
<b>VAF (VA)*</b>	напруга Ерлі у нормальному режимі	$\infty$	
<b>VAR (VB)*</b>	напруга Ерлі в інверсному режимі	$\infty$	
<b>VJC (PC)*</b>	Контактна різниця потенціалів перетину база-колектор	0.75	
<b>VJE (PE)*</b>	Контактна різниця потенціалів перетину база-емітер	0.75	
<b>VJS (PS)*</b>	Контактна різниця потенціалів перетину колектор-підкладка	0.75	
<b>VO</b>	Напруга, що визначає перегиб залежності струму епітаксильної області	10	
<b>VTF</b>	Напруга, що характеризує залежність TF від зсуву база-колектор	$\infty$	



<b>XCJC</b>	Коефіцієнт розщеплення ємності база-колектор CJC	1	
<b>XTB</b>	температурний коефіцієнт BF і BR	0	
<b>XTF</b>	Коефіцієнт, що визначає залежність TF від зсуву база-колектор	0	
<b>XTI (PT)*</b>	Температурний коефіцієнт IS	3	

Знак \* справа від параметра означає, що даний параметр застосовують тільки для моделі Гумеля-Пуна.

**РИСУНОК 1. – СХЕМА ЗАМІЩЕННЯ БІПОЛЯРНОГО ТРАНЗИСТОРА У ВИГЛЯДІ МОДЕЛІ ГУМЕЛЯ -ПУНА**

**РИСУНОК 1. – СХЕМА ЗАМІЩЕННЯ БІПОЛЯРНОГО ТРАНЗИСТОРА У ВИГЛЯДІ МОДЕЛІ ЕБЕРСА–МОЛЛА**

Наведемо співвідношення для статичного режиму роботи біполярного транзистора. В цих та подальших співвідношеннях крім вказаних у таблиці параметрів будемо використовувати скорочені позначення таких параметрів моделі :  $I_b$  – струм бази,  $I_c$  – струм колектора,  $I_{be1}$ – струм колектора у нормальному режимі,  $I_{bc1}$ – струм колектора в інверсному режимі,  $I_{be2}$ ,  $I_{bc2}$ – складові струму переходу база-емітер, що викликані неідеальністю переходу,  $I_s$  – струм підкладки,  $V_{be}$ ,  $V_{bc}$  – напруга на переході внутрішня база-емітер і внутрішня база-колектор,  $V_{bs}$  – напруга внутрішня база-підкладка,  $V_{bn}$  – напруга внутрішня база-підкладка для режиму квазінасичення,  $V_{bw}$  – напруга внутрішня база-колектор для режиму квазінасичення,  $V_{bx}$  – напруга база-внутрішній колектор,  $V_{ce}$  – напруга внутрішній колектор-внутрішній емітер,  $V_{js}$  – напруга внутрішній колектор-підкладка для NPN транзистора, напруга підкладка внутрішній колектор для PNP транзистора, напруга внутрішня база-підкладка для LPNP транзистора. Також позначено температурний потенціал  $V_t = k \cdot T / q$ .

$$I_b = Area \left( \frac{I_{be1}}{BF} + I_{be2} + \frac{I_{bc1}}{BR} + I_{bc2} \right);$$

$$I_c = Area \left( \frac{I_{be1}}{Q_b} - \frac{I_{bc1}}{Q_b} - \frac{I_{bc1}}{BR} - I_{bc2} \right);$$

$$I_{be1} = IS \left[ \exp \left( \frac{V_{be}}{(NF \cdot Vt)} \right) - 1 \right];$$

$$I_{be2} = ISE \left[ \exp \left( \frac{V_{be}}{(NE \cdot Vt)} \right) - 1 \right];$$

$$I_{bc1} = IS \left[ \exp \left( \frac{V_{bc}}{(NR \cdot Vt)} \right) - 1 \right];$$

$$I_{bc2} = ISC \left[ \exp \left( \frac{V_{bc}}{(NC \cdot Vt)} \right) - 1 \right];$$

$$Q_b = \frac{Q_1 \left[ 1 + (1 + 4Q_2)^{NK} \right]}{2};$$

$$Q_1 = \frac{1}{\left( 1 - \frac{V_{bc}}{VAF} - \frac{V_{be}}{VAR} \right)}; \quad Q_2 = \frac{I_{be1}}{IKF} + \frac{I_{bc1}}{IKR};$$

$$I_s = Area \cdot ISS \left[ \exp \left( \frac{V_{js}}{(NS \cdot Vt)} \right) - 1 \right].$$

Коефіцієнт  $Area$  дозволяє врахувати паралельне поєднання однакових транзисторів. За замовчанням, тобто при наявності одного аналізованого транзистора він дорівнює одиниці.

Питомий опір бази  $Rb$  характеризують за допомогою двох складових. Перша складова  $RB$  визначає опір виводів бази та опір зовнішньої області бази, якій не залежить від струму бази  $Ib$ . Друга складова  $RBM$  характеризує опір активної області бази, який знаходиться безпосередньо під емітером, цей опір залежить від струму  $Ib$ . Питомий опір бази  $Rb$  визначають за допомогою таких співвідношень в залежності від значення параметру  $IRB$ .

$$R_b = \begin{cases} RBM + \frac{(RB - RBM)}{(Area \cdot Q_b)}, & \text{якщо } IRB = \infty \\ RBM + 3(RB - RBM) \frac{tgX - X}{Arrea}, & \text{якщо } IRB > 0 \end{cases}$$

де 
$$X = \frac{\sqrt{1 + \frac{14,59025 \cdot I_b}{(Area \cdot IRB) - 1}}}{2.4317 \sqrt{\frac{I_b}{(Area \cdot IRB)}}};$$

Динамічні властивості переходів враховані введенням в модель транзистора ємностей колектора, емітера і підкладки, які мають дифузійні та бар'єрні складові. Ємність переходу база – емітер дорівнює сумі дифузійної та бар'єрної складових :

$$C_{be} = C_{tbe} + Area \cdot C_{jbe}$$

де  $C_{tbe} = t_f \cdot G_{be}$ ; а  $G_{be} = \frac{dI_{be}}{dV_{be}}$  – диференційна провідність переходу база-емітер в робочій точці за постійним струмом.  
;

$$I_{be} = I_{be1} + I_{be2};$$

$$C_{jbe} = \begin{cases} CJE \cdot \left(1 - \frac{V_{be}}{VJE}\right)^{-MJE} & \text{at } V_{be} \leq FC \cdot VJE; \\ CJE \cdot (1 - FC)^{-(1+MJE)} \cdot \left[1 - FC(1 + MJE) + MJE \cdot \frac{V_{be}}{VJE}\right] & \text{а} \end{cases}$$

Ємність переходу база–колектор розкладають на дві складові :

- ємність між внутрішньою базою та колектором :

$$C_{bc} = C_{tbc} + Area \cdot XCJC \cdot C_{jbe};$$

$$\text{де } C_{tbc} = TR \cdot G_{bc}; \quad G_{bc} = \frac{dI_{bc}}{dV_{bc}};$$

$$I_{bc} = I_{bc1} + I_{bc2}$$

$$C_{jbc} = \begin{cases} CJC \cdot \left(1 - \frac{V_{bc}}{VJC}\right)^{-MJC} & \text{at } V_{bc} \leq FC \cdot VJC; \\ CJC \cdot (1 - FC)^{-(1+MJC)} \cdot \left[1 - FC(1 + MJC) + MJC \cdot \frac{V_{bc}}{VJC}\right] & ; \end{cases}$$

- ємність між зовнішнім виводом бази та колектором

$$C_{bx} = Area \cdot (1 - XCJC) \cdot C_{jbx};$$

$$C_{jbx} = \begin{cases} CJC \cdot \left(1 - \frac{V_{bx}}{VJC}\right)^{-MJC} & \text{at } V_{bx} \leq FC \cdot VJC; \\ CJC \cdot (1 - FC)^{-(1+MJC)} \cdot \left[1 - FC(1 + MJC) + MJC \cdot \frac{V_{bx}}{VJC}\right] & \text{a} \end{cases}$$

Ємність колектор-підкладка дорівнює

$$C_{js} = \begin{cases} Area \cdot CJS \cdot \left(1 - \frac{V_{js}}{VJS}\right)^{-MJS} & \text{at } V_{js} \leq 0; \\ Area \cdot CJS \cdot \left(1 + \frac{MJS \cdot V_{js}}{VJS}\right)^{-MJS} & \text{at } V_{js} > 0 \end{cases}$$

Режим квазінасищення характеризує прямий зсув переходу внутрішня база-колектор коли переход зовнішня база-колектор залишається зсунутим у зворотному напрямку. В моделі Гумеля – Пуна цей ефект імітують за допомогою додаткового керованого джерела струму  $I_{epi}$  і двох нелінійних ємностей, заряди яких у схемі на рис. 1.1 позначено  $Q_0$  та  $Q_w$ . Ці зміни вносять в модель, якщо задано параметр RCO.

$$I_{epi} = Area \cdot \frac{VO \left\{ Vt \left[ K(V_{bc}) - K(V_{bn}) - \ln \left( \frac{1 + K(V_{bc})}{1 + K(V_{bn})} \right) \right] \right\}}{RCO \cdot (|V_{bc} - V_{bn}| + VO)}$$

$$Q_0 = Area \cdot QCO \left[ K(V_{bc}) - 1 - \frac{GAMMA}{2} \right];$$

$$Q_w = Area \cdot QCO \left[ K(V_{bn}) - 1 - \frac{GAMMA}{2} \right];$$

$$K(V) = \sqrt{1 + GAMMA \cdot \exp\left(\frac{V}{Vt}\right)}$$

Температурні залежності параметрів елементів еквівалентної схеми БПТ встановлюють за допомогою таких співвідношень:

$$IS(T) = IS \cdot e^{\left(\frac{EG}{V_t(T)}\right) \left(\frac{T}{T_{НОМ}} - 1\right)} \cdot \left(\frac{T}{T_{НОМ}}\right)^{XTI};$$

$$ISE(T) = \frac{ISE}{\left(\frac{T}{T_{НОМ}}\right)^{XTB}} \cdot e^{\left(\frac{T}{T_{НОМ}} - 1\right) \cdot \frac{EG}{(NE \cdot Vt)}} \cdot \left(\frac{T}{T_{НОМ}}\right)^{\frac{XTI}{NE}}$$

;

$$ISC(T) = \frac{ISC}{\left(\frac{T}{T_{НОМ}}\right)^{XTB}} \cdot e^{\left(\frac{T}{T_{НОМ}} - 1\right) \cdot \frac{EG}{(NC \cdot Vt)}} \cdot \left(\frac{T}{T_{НОМ}}\right)^{\frac{XTI}{NC}}$$

;

$$ISS(T) = \frac{ISS}{\left(\frac{T}{T_{НОМ}}\right)^{XTB}} \cdot e^{\left(\frac{T}{T_{НОМ}} - 1\right) \cdot \frac{EG}{(NS \cdot Vt)}} \cdot \left(\frac{T}{T_{НОМ}}\right)^{\frac{XTI}{NS}} ;$$

$$BF(T) = BF \cdot \left(\frac{T}{T_{НОМ}}\right)^{XTB} ;$$

$$BR(T) = BR \cdot \left(\frac{T}{T_{НОМ}}\right)^{XTB} ;$$

$$RE(T) = RE \cdot \left(1 + TRE1 \cdot (T - T_{НОМ}) + TRE2 \cdot (T - T_{НОМ})^2\right)$$

;

$$RB(T) = RB \cdot \left(1 + TRB1 \cdot (T - T_{НОМ}) + TRB2 \cdot (T - T_{НОМ})^2\right)$$

;

$$RBM(T) = RBM \cdot \left(1 + TRM1 \cdot (T - T_{НОМ}) + TRM2 \cdot (T - T_{НОМ})^2\right)$$

;

$$RC(T) = RC \cdot \left(1 + TRC1 \cdot (T - T_{НОМ}) + TRC2 \cdot (T - T_{НОМ})^2\right)$$

;

$$VJE(T) = VJE \cdot \frac{T}{T_{НОМ}} - 3Vt \cdot \ln\left(\frac{T}{T_{НОМ}}\right) - EG(T_{НОМ}) \cdot \frac{T}{T_{НОМ}} +$$

;

$$VJC(T) = VJC \cdot \frac{T}{T_{НОМ}} - 3Vt \cdot \ln\left(\frac{T}{T_{НОМ}}\right) - EG(T_{НОМ}) \cdot \frac{T}{T_{НОМ}} +$$

;

$$VJS(T) = VJS \cdot \frac{T}{T_{НОМ}} - 3Vt \cdot \ln\left(\frac{T}{T_{НОМ}}\right) - EG(T_{НОМ}) \cdot \frac{T}{T_{НОМ}} +$$

;

$$EG(T) = EG_0 - \frac{a \cdot T^2}{b + T} = 1,16 - \frac{0,000702 \cdot T^2}{1108 + T} ;$$

$$\begin{aligned}
CJE(T) &= CJE \cdot \left( 1 + MJE \cdot \left( 0.0004 \cdot (T - T_{НОМ}) + \left( 1 - \frac{VJE(T)}{VJE} \right) \right) \right); \\
CJC(T) &= CJC \cdot \left( 1 + MJC \cdot \left( 0.0004 \cdot (T - T_{НОМ}) + \left( 1 - \frac{VJC(T)}{VJC} \right) \right) \right); \\
CJS(T) &= CJS \cdot \left( 1 + MJS \cdot \left( 0.0004 \cdot (T - T_{НОМ}) + \left( 1 - \frac{VJS(T)}{VJS} \right) \right) \right); \\
KF(T) &= KF \cdot \frac{VJC(T)}{VJC}; \\
AF(T) &= AF \cdot \frac{VJC(T)}{VJC}.
\end{aligned}$$

При виконанні автоматизованого аналізу із урахуванням шумових властивостей транзистору у схему додатково вмикають джерела шуму у вигляді флуктуаційних джерел шуму. На рис. 1.1 показано лінійну модель БПТ із включеними джерелами шуму.

Рисунок 1.1 – Лінійна схема заміщення БПТ із джерелами шуму

Теплові струми  $I_{uRB}$ ,  $I_{uRE}$  і  $I_{uRC}$  які створюють резистори  $RC$ ,  $RB$  та  $RE$  мають спектральні щільності

$$S_{RB} = \frac{4kT}{RB}; \quad S_{RC} = \frac{4kT}{RC/Area}; \quad S_{RE} = \frac{4kT}{RE/Area}.$$

Джерела струму  $I_{ub}$ ,  $I_{uc}$  які характеризують дробовий та фліккер-шуми в ланцюгах бази та колектора, мають відповідні спектральні щільності



$$S_b = 2qI_b + KF \cdot \frac{I_b^{AF}}{f}; \quad S_c = 2qI_c.$$

#### Моделювання транзисторів НВЧ

Радіонавігаційні пристрої, пристрої авіаційного зв'язку містять ланцюги для опрацювання надвисокочастотних сигналів із частотою від сотень мегагерц до десятків гигагерц. Зазвичай стандартні конфігурації пакетів автоматизованого проектування не містять програмних модулів для імітаційного моделювання НВЧ пристроїв, тому для проектування таких ланцюгів слід використовувати спеціалізовані програми (наприклад Microwave). Але існує декілька способів імітації поведінки надвисокочастотних пристроїв у стандартних програмах АСХП. Наведемо один з таких способів.

В моделях транзисторів на високих частотах у першу чергу беруть до уваги індуктивності виводів і паразитні ємності між выводами і корпусом транзистора. Через це найбільш природнім є рішення, яке передбачає створення високочастотного транзистора на основі існуючої, стандартної моделі, яка оснащена пасивними LC-ланцюгами. Слід додати, що існує багато схем технічних конфігурацій таких ланцюгів, які враховують конструктивні особливості транзистора. В якості приклада розглянемо як побудовано модель НВЧ транзистора 2Т934А.

Рисунок 1.1 – Нелінійна макромодель БПТ транзистора

VТ1 = 2Т934А  
 С1 = 1,84 пФ;  
 С2 = 0,96 пФ;  
 С3 = 1,53 пФ;  
 Le = 1,3 нГн;  
 Lb = 3,1 нГн;  
 Lc = 2,5 нГн.

На рис.1.1 наведено нелінійну макромодель біполярного транзистора 2Т934А яку створено на основі ідеальної SPICE-моделі, параметрам якої надано потрібних числових значень. Нижче наведено фрагмент текстового завдання на моделювання у форматі PSPICE із описом цієї макромоделі :

```
.subckt 2Т934А 1 2 3
```

```

QVT1 4      5      6      Qvhf
Le    6      3      1.3n
Lb    2      5      3.1n
Lc    ..... 1      4
C1    4      3      1.84p
C2    4      2      .96p
C3    1      4      1,53p
.model          Qvhf          NPN
(IS=2nA BF=50 BR=4.6 NF=1.4 NR=1.5 CJE=32p
+
CJC=20p VJE=0.8 VJC=0.8 MJE=0.4 MJC=0.5 TF=0.18n TR=4.5
n
+ VAF=80 RB=1.4 RC=1.3)
.ends

```

Така модель дозволяє розраховувати як режим по постійному струму, так і динамічні характеристики пристрою.

Також можливо використовувати малосигнальні лінійні моделі транзисторів в яких використовують кероване джерело енергії. Ідентифікацію параметрів мало сигнальних схем заміщення транзисторів виконують на основі розрахунку їх Y-параметрів. Ще один спосіб засновано на застосуванні методики розрахунку S-параметрів малосигнальних схем заміщення. Такі методи можливо також застосовувати для транзисторів на основі польового ефекту.

#### МОП транзистор [ ]

Модель характеризується следующими параметрами:

Rd — сопротивление стокового слоя;

RS — сопротивление истокового слоя;

BETA — удельная крутизна передаточной характеристики транзистора.

BETA TC — температурный коэффициент BETA;

VTO — пороговое напряжение;

VTO TC — температурный коэффициент порогового напряжения;

GAMMA — коэффициент эффекта Бодэ;

PHI — потенциал инверсии зон.  
 LAMBDA — коэффициент модуляции канала;  
 Cgd — емкость затвор-сток;  
 Cgs — емкость затвор-исток;  
 Cbs- емкость подложка-исток;  
 Cbd- емкость подложка-сток;  
 Cgb- емкость затвор-подложка;  
 Rb — сопротивление слоя подложки;  
 Rds — сопротивление сток-исток закрытого транзистора.

Вычисление BETA осуществляется по величине сопротивления канала транзистора на начальном участке выходных характеристик. Несложно показать, что коэффициент BETA для крутого участка ВАХ вычисляется по формуле:

$$BETA = \frac{1}{2 * RON * (V_{gs} - V_{TO} - RON * I_d)}$$

где RON — выходное статическое сопротивление транзистора на начальном (крутом) участке характеристик (Ом);

- Vgs — напряжение затвор-исток;
- Id — ток стока.

Параметр Lambda определяет линейное нарастание BETA с ростом Vds за счет модуляции длины канала.

Эквивалентная схема рис. 1 является базовой для моделей шести уровней, различающихся учетом тех или иных эффектов. Модель уровня 1 (LEVEL1) наиболее проста и используется, если не требуется высокая точность расчетов. Модель уровня 2, наоборот, довольно сложная и характеризуется большим числом трудно определяемых параметров. Широко известны физические модели BSIM3 и BSIM4 (аббревиатура BSIM есть сокращение фразы Berkeley short-channel IGFET model, где IGFET - insulated-gate field-effect transistor, т.е. BSIM - модель МОП-транзистора с изолированным затвором и коротким каналом, разработанная в Беркли).

Пользовий транзистор

ПТ із керуванням р-п перетином (Junction FET) в програмах АСхП описують за допомогою моделі Шихмана-Ходжеса у відповідності до еквівалентних схем, які представлені на Рисунку 1.16 та рис. 1.17.

Рисунок 1.16 – Нелінійна еквівалентна схема польового транзистора із керованим  $p-n$  перетином та каналом  $n$ -типу

Рисунок 1.17 – Лінійна еквівалентна схема польового транзистора

В схемі на рис.1.17 включено додаткові джерела флуктуаційних струмів. Теплові шуми, які створюють резистори  $R_S$  та  $R_D$ , мають спектральні щільності

$$S_{RS} = 4kT / (R_S \cdot AREA), S_{RD} = 4kT / (R_D \cdot AREA).$$

Параметри моделі польового транзистора наведені у табл.1.17.

**ТАБЛИЦЯ 1.7**  
ПАРАМЕТРИ SPICE-МОДЕЛІ ПОЛЬОВОГО ТРАНЗИСТОРА

Параметр	Позначення параметра	Розмірність	Значення за умовчанням
AF	Показчик ступеня, що визначає залежність спектральної щільності фліккер-шуму від струму через $p-n$ перетин		1
<b>ALPHA</b>	Коефіцієнт іонізації	В	0
<b>BETA</b>	Коефіцієнт пропорційності	A/B	$10^{-4}$
<b>BETA TCE</b>	Температурний коефіцієнт BETA	%/ $^{\circ}$ C	0
CGD	Ємність переходу затвор-сток при нульовому зсуві	Ф	0
CGS	Ємність переходу затвор-джерело при нульовому зсуві	Ф	0
FC	Коефіцієнт нелінійності бар'єрних ємностей зміщених у прямому напрямі переходів		0,5
IS	Струм насичення $p-n$ переходу при температурі $27^{\circ}$ C	A	$10^{-16}$
<b>ISR</b>	Параметр струму рекомбінації $p-n$ переходу затвор-канал	A	0
KF	Коефіцієнт, що визначає спектральну	A	0

	щільність флікер-шуму		
<b>LAMBDA A</b>	Параметр модуляції довжини каналу	1/B	0
<b>M</b>	Коефіцієнт лавинного множення збідненого <i>p-n</i> переходу затвор - канал		0,5
<b>N</b>	Коефіцієнт неідеальності <i>p-n</i> переходу затвор- канал		1
<b>NR</b>	Коефіцієнт емісії для струму ISR		1.0
<b>PB</b>	Контактна різниця потенціалів <i>p-n</i> переходу затвору	B	1
<b>RD</b>	Питомий опір області стоку	Ом	0
<b>RS</b>	Питомий опір області джерела	Ом	0
<b>T_ABS</b>	Абсолютна температура	°C	
<b>T_MEASURED</b>	Виміряна температура	°C	
<b>T_REL_GLOBAL</b>	Відносна до струму температура	°C	
<b>T_REL_LOCAL</b>	Різниця між температурою транзистора і моделі прототипу (АКО)	°C	
<b>VK</b>	Напруга іонізації для переходу затвор-канал	B	0
<b>VT0</b>	напруга відсічки	B	-2
<b>VTO TC</b>	Температурний коефіцієнт VT0	B/°C	0
<b>XTI</b>	Температурний коефіцієнт струму IS		3

Статичні характеристики польового транзистора описують такими залежностями:

Сумарний струм затвора складається з двох складових – струму  $I_{gs}$  та струму  $I_{gd}$  та дорівнює

$$I_g = Area(I_{gs} + I_{gd}),$$

де  $I_{gs} = I_n + I_r K_{gs}$  — струм витікання затвор-джерело містить

також дві складові:

$$I_n = IS[\exp(V_{gs}/(N \cdot V_t) - 1)] - \text{нормальний струм};$$

$$I_r = ISR[\exp(V_{gs}/(NR \cdot V_t) - 1)] - \text{струм рекомбінації};$$

$V_{gs}$  — напруга затвор-джерело,  $V_t$  – температурний потенціал;

$$K_{gs} = [(1 - V_{gs} / PB)^2 + 0,005]^{M/2} \text{ — фактор генерації;}$$

$$I_{gd} = I_{nd} + I_{rd} \cdot K_{gd} + I_I \text{ — струм витікання затвор–сток;}$$

$$I_{nd} = IS[\exp(V_{gd} / (N \cdot V_T)) - 1] \text{ — нормальний струм;}$$

$$V_{gd} \text{ — напруга затвор–сток,}$$

$$I_{rd} = ISR[\exp(V_{gd} / (NR \cdot V_T)) - 1] \text{ — струм рекомбінації;}$$

$$K_{gd} = [(1 - V_{gd} / PB)^2 + 0,005]^{M/2} \text{ — фактор генерації;}$$

$$I_I \text{ — струм іонізації, що дорівнює}$$

$$I_I = \begin{cases} I_{drain} \cdot ALPHA \cdot V_{dif} \cdot \exp(-VK / V_{dif}) & \text{при } 0 < V_{gs} < V_{ds} \text{ (режим насичення);} \\ 0 & \text{в інших діапазонах} \end{cases}$$

$$V_{dif} = V_{ds} - (V_{gs} - VTO); \quad V_{ds} \text{ — напруга сток–джерело;}$$

Польовий транзистор збідненого типу характеризує негативне значення  $VTO < 0$  (для каналів  $p$ - та  $n$ -типів), а транзистор збагаченого типу – позитивним ( $VTO \geq 0$ ).

Струми стока та джерела відповідно дорівнюють :

$$I_d = Area(I_{drain} - I_{gd}), \quad I_s = Area(-I_{drain} - I_{gs}).$$

В нормальному режимі ( $V_{ds} \geq 0$ ) струм  $I_{drain}$  розраховують згідно виразів:

$$I_{drain} = \begin{cases} 0 & \text{при } V_{gs} - VTO \leq 0; \\ BETA(1 + LAMBDA \cdot V_{ds})(V_{gs} - VTO)^2 & \text{при } 0 < V_{gs} - VTO \leq V_{ds} \text{ (режим нас)} \\ BETA(1 + LAMBDA \cdot V_{ds})V_{ds} [2(V_{gs} - VTO) - V_{ds}] & \text{при } V_{ds} < V_{gs} - VTO \text{ (лин} \end{cases}$$

Область  $V_{gs} - VTO < 0$  відповідає режиму відсічки. В інверсному режимі ( $V_{ds} < 0$ )

$$I_{drain} = \begin{cases} 0 & \text{при } V_{gd} - VTO \leq 0; \\ -BETA(1 - LAMBDA \cdot V_{ds}) \cdot (V_{gd} - VTO)^2 & \text{при } 0 < V_{gd} - VTO \leq -V_{ds}; \\ BETA(1 - LAMBDA \cdot V_{ds}) \cdot V_{ds} \cdot [2(V_{gd} - VTO) + V_{ds}] & \text{при } -V_{ds} < V_{gd} - VTO; \end{cases}$$

Ємності затвор–джерело та затвор–сток описують виразами:

$$C_{gs} = \begin{cases} CGS(1 - V_{gs} / PB)^{-M} & \text{при } V_{gs} \leq FC \cdot PB; \\ CGS(1 - FC)^{-(1+M)} [1 - FC(1+M) + M \cdot V_{gs} / PB] & \text{при } V_{gs} > FC \cdot PB; \end{cases}$$

$$C_{gd} = \begin{cases} CGS(1 - V_{gd} / PB)^{-M} & \text{при } V_{gd} \leq FC \cdot PB; \\ CGS(1 - FC)^{-(1+M)} [1 - FC(1+M) + M \cdot V_{gd} / PB] & \text{при } V_{gd} > FC \cdot PB. \end{cases}$$

Температурні ефекти характеризують такими залежностями:

$$VTO(T) = VTO + VTO TC \cdot (T - T_{nom});$$

$$BETA(T) = BETA \cdot 1,01^{BETATCE(T - T_{nom})};$$

$$IS(T) = IS \cdot \exp[EG(T_{nom}) / (N \cdot V_t) \cdot (T / T_{nom} - 1)] (T / T_{nom})^{XTI / N};$$

$$ISR(T) = ISR \cdot \exp[EG(T_{nom}) / (NR \cdot V_t) \cdot (T / T_{nom} - 1)] (T / T_{nom})^{XTI / NP};$$

$$PB(T) = PB \cdot T / T_{nom} - 3V_t I_{\Gamma} (T / T_{nom}) - EG(T_{nom}) \cdot T / T_{nom} + EG(T);$$

$$CGS(T) = CGS \{1 + M [0,0004(T - T_{nom}) + 1 - PB(T) / PB]\};$$

$$CGD(T) = CGD \{1 + M [0,0004(T - T_{nom}) + 1 - PB(T) / PB]\};$$

$$KF(T) = KF \cdot PB(T) / PB, AF \cdot PB(T) / PB.$$

Скалярний коефіцієнт *AREA* дозволяє врахувати паралельне поєднання транзисторів одного типу, для чого в наведеній вище моделі на цей коефіцієнт помножуються такі параметри моделі: *IS*, *BETA*, *RD*, *RS*, *CGS*, *CGD*. Якщо не вказано інше, значення цього коефіцієнту дорівнює одиниці.

Арсенід-галійові польові транзистори (GaAsFET) мають еквівалентну схему, яка наведена на рис. 1.18

Рисунок 1.18 – Нелінійна схема заміщення GaAsFET

Існує п'ять варіантів цієї базової моделі запропоновані різними дослідниками (Куртиса, Рейтеона, дві моделі TriQiu та варіант Паркера-Скеллера). Варіант моделі Куртиса надає задовільні результати при моделюванні статичних режимів, інші застосовують також і при моделюванні динамічних характеристик.

Статичний режим описують такими співвідношеннями.

1) струм затвора дорівнює:

$$I_g = Area(I_{gs} + I_{gd})$$

Для моделей (крім моделі Паркера-Скеллерна)

$$I_{gs} = IS[\exp(V_{gs}/(N \cdot V_t)) - 1]; \quad I_{gd} = IS[\exp(V_{gd}/(N \cdot V_t)) - 1].$$

Для моделі Паркера-Скеллерна

$$I_{gs} = I_{gsf} + I_{gsr}$$

де

$$I_{gsf} = IS[\exp(V_{gs}/(N \cdot V_t)) - 1] + V_{gs} \cdot GMIN ;$$

$$I_{gsr} = IBD[1 - \exp(-V_{gs}/VBD)].$$

$$I_{gd} = I_{gdf} + I_{gdr}$$

де

$$I_{gdf} = IS[\exp(V_{gd}/(N \cdot V_t)) - 1] + V_{gd} \cdot GMIN ;$$

$$I_{gdr} = IBD[1 - \exp(-V_{gd}/VBD)].$$

Струм стока та джерела

$$I_d = Area(I_{drain} - I_{gd}), \quad I_s = Area(-I_{drain} - I_{gs}).$$

Струм  $I_{drain}$  в моделі Куртиса (LEVEL=1) в нормальному режимі

( $V_{ds} \geq 0$ ) описують співвідношеннями:

$$I_{drain} = \begin{cases} 0 & \text{при } V_{gs} - VTO < 0; \\ BETA(1 + LAMBDA \cdot V_{ds})(V_{gs} - VTO)^2 th(ALPHA \cdot V_{ds}) & \text{при } V_{gs} - VTO \geq 0; \\ \text{(режим насичення та лінійний режим)} \end{cases}$$

В моделі Рэйтеона (LEVEL=2) в нормальному режимі:

$$I_{drain} = \begin{cases} 0 & \text{при } V_{gs} - VTO < 0; \\ \frac{BETA(1 + LAMBDA \cdot V_{ds})(V_{gs} - VTO)^2 K_t}{[1 + B \cdot (V_{gs} - VTO)]} & \text{при } V_{gs} - VTO \geq 0, \end{cases}$$

де поліноміальна апроксимація гіперболічного тангенса має вигляд



$$K_t = \begin{cases} 1 - \left( \frac{1 - V_{ds} \cdot ALPHA}{3} \right)^3 & \text{при } \frac{0 < V_{ds} < 3}{ALPHA} \text{ (лінійний режим)} \\ 1 & \text{при } V_{ds} \geq \frac{3}{ALPHA} \text{ (режим насичення)} \end{cases}$$

Для моделі TriQuit (LEVEL=3) в нормальному режимі

$$I_{drain} = \begin{cases} 0 & \text{при } V_{gs} - V_{to} < 0; \\ \frac{I_{dso}}{(1 + DELTA \cdot V_{ds} \cdot I_{dso})} & \text{при } V_{gs} - V_{to} \geq 0 \\ \text{(режим насичення та лінійний режим)} \end{cases}$$

де

$$I_{dso} = BETA(V_{gs} - V_{to})^Q K_t;$$

$$V_{to} = V_{TO} - GAMMA \cdot V_{ds}$$

В інверсному режимі ( $V_{ds} < 0$ ) струми стока та джерела в наведених вище виразах змінюють місця.

Динамічний режим. Ємність переходу джерело-сток дорівнює  $C_{ds} = CDS$ . В моделі Куртиса ємності  $C_{gs}$ ,  $C_{gd}$  визначають за допомогою виразів:

ємність затвор-джерело дорівнює

$$C_{gs} = \begin{cases} CGS \cdot \left(1 - \frac{V_{gs}}{VBI}\right)^{-M} & \text{при } V_{gs} < FC \cdot VBI; \\ CGS \cdot (1 - FC)^{-(1+M)} \left[ \frac{1 - FC \cdot (1+M) + M \cdot V_{gs}}{VBI} \right] & \text{при } V_{gs} > FC \cdot VBI; \end{cases}$$

ємність затвор-сток дорівнює

$$C_{gd} = \begin{cases} CGD \cdot \left(1 - \frac{V_{gd}}{VBI}\right)^{-M} & \text{при } V_{gd} < FC \cdot VBI; \\ CGD \cdot (1 - FC)^{-(1+M)} \left[ \frac{1 - FC \cdot (1+M) + M \cdot V_{gd}}{VBI} \right] & \text{при } V_{gd} > FC \cdot VBI; \end{cases}$$

В моделі Рейтеона та TOM1 TriQuit ці дві ємності визначають виразами:

$$C_{gs} = CGS \cdot K_2 \cdot K_1 / \sqrt{1 - V_n / VBI} + CGD \cdot K_3;$$

$$C_{gd} = CGS \cdot K_3 \cdot K_1 / \sqrt{1 - V_n / VBI} + CGD \cdot K_2,$$

де

$$K_1 = 0,5 \left[ 1 + \frac{V_e - VTO}{\sqrt{(V_e - VTO)^2 + VDELTA^2}} \right];$$

$$K_2 = 0,5 \left[ 1 + \frac{V_{gs} - V_{gd}}{\sqrt{(V_{gs} - V_{gd})^2 + 1/ALPHA^2}} \right];$$

$$K_3 = 0,5 \left[ 1 - \frac{V_{gs} - V_{gd}}{\sqrt{(V_{gs} - V_{gd})^2 + 1/ALPHA^2}} \right];$$

$$V_e = 0,5 \left[ V_{gs} + V_{gd} + \sqrt{(V_{gs} - V_{gd})^2 + 1/ALPHA^2} \right];$$

$$V_n \begin{cases} 0,5 \left[ V_e + VTO + \sqrt{(V_e - VTO)^2 + VDELTA^2} \right] \\ \text{при } V_e + VTO + \sqrt{(V_e - VTO)^2 + VDELTA^2} < VMAX \\ VMAX \text{ в інших діапазонах} \end{cases}$$

В лінійній схемі заміщення транзистора GaAsFET, яка наведена на рис.1.19, додатково вмикають джерела флуктуаційних струмів. Теплові шуми  $I_{RS}$ ,  $I_{RD}$ ,  $I_{RG}$ , створювані резисторами  $RS$ ,  $RD$ ,  $RG$  мають спектральні щільності

$$S_{RS} = 4kT / (RS / AREA), S_{RD} = 4kT / (RD / AREA), S_{RG} = 4kT / RG.$$

Джерело струму  $I_{nd}$  яке характеризує дробовий та фліккершуми, має спектральну щільність  $S_d = 8k \cdot T \cdot G_m / 3 + KF \cdot I_{drain}^{AF} / f$

де  $G_m = \frac{dI_{drain}}{dV_{gs}}$  - диференційна провідність в робочій точці за постійним струмом.

Рисунок 1.19 – Лінійна еквівалентна схема GaAsFET-транзистора

Температурні ефекти впливають на такі залежності

$$IS(T) = IS \cdot \exp\left[\frac{EG}{(N \cdot V_t)} \cdot \left(\frac{T}{T_{nom}} - 1\right)\right] \left(\frac{T}{T_{nom}}\right)^{XTI/N};$$

$$VBI(T) = VBI \cdot T/T_{nom} - 3V_t \ln(T/T_{nom}) - EG(T_{nom}) \cdot T/T_{nom} + EG(T);$$

$$CGS(T) = CGS \{1 + M[0,0004(T - T_{nom}) + 1 - VBI(T)/VBI]\};$$

$$CGD(T) = CGD \{1 + M[0,0004(T - T_{nom}) + 1 - VBI(T)/VBI]\};$$

$$VTO(T) = VTO + VTOTC \cdot (T - T_{nom});$$

$$BETA(T) = BETA \cdot 1,01^{BETATC(T - T_{nom})};$$

$$RG(T) = RG(1 + TRG1(T - T_{nom}));$$

$$RD(T) = RD(1 + TRD1(T - T_{nom}));$$

$$RS(T) = RS(1 + TRS1(T - T_{nom}));$$

$$KF(T) = KF \cdot VBI(T)/VBI, AF \cdot VBI(T)/VBI.$$

При паралельному включенні декількох транзисторів потрібно враховувати скалярний коефіцієнт *Area*, який змінює такі параметри

$$IS = IS \cdot Area; BETA = BETA \cdot Area; RD = RD / Area; RS = RS / Area;$$

$$CGS = CGS \cdot Area; CGD = CGD \cdot Area, CDS = CDS \cdot Area.$$

### МДП транзистор

Польові транзистори з ізолюваним затвором, до яких відносять метал-діелектрик-напівпровідник (МДН) транзистори та найбільш розповсюджені метал-оксид-напівпровідник (МОН) транзистори (MOSFET) описують за допомогою шести моделей. Таку увагу до цієї групи транзисторів можна пояснити тим, що на основі польової

технології транзисторів з ізольованим затвором побудована вся сучасна цифрова та більшість аналогової електронної техніки.

В перших трьох моделях МДН-транзисторів використовують геометричні параметри структури транзистора. Для їх розуміння наведемо структуру МДН транзистора із каналом *n*-типу (рис.1.20).

Рисунок 1.20 – Структура МДН-транзистора із каналом *n*-типу

В структурі на рис.1.20 вказані такі параметри моделі:

<i>L</i>	– довжина каналу провідності
<i>LD</i>	– довжина області бокової дифузії
<i>L<sub>eff</sub></i>	– ефективна довжина каналу провідності
<i>RS</i>	– питомий опір джерела
<i>RD</i>	– питомий опір стока
<i>TOX</i>	– товщина шару оксиду
<i>XJ</i>	– глибина металічного переходу областей стока та джерела
<i>W</i>	– ширина каналу

Оскільки в структурі на рис.1.20 вказано, що в якості діелектрика використано оксид кремнію, то далі будемо розуміти МОН-транзистор.

Частина параметрів моделі задають на вхідній мові програмі SPICE – це параметри *L*, *W*, *AD*, *AS*, *PD*, *PS*, *NRD*, *NRS*, *NRG*, *NRB*, *M*. При цьому параметрам *L*, *W*, *AD*, *AS* встановлюють значення за умовчанням за директивою OPTIONS. Питомі опори *RD*, *RS*, *RG*, *RG* задають безпосередньо вказуючи числове значення, або обраховують шляхом множення питомого опору *RSH* на *NRD*, *NRG*, *NRS*, *NRB*.

Усі моделі використовують одну еквівалентну схему, яку наведено на рис. 1.21.

Рисунок 1.21 – Нелінійна еквівалентна схема МДН-транзистора

При розрахунках параметрів моделей МОН-транзисторів використовують такі фізичні константи:

$EPS_0 = 8,86 \cdot 10^{-12}$  Ф/м – абсолютна діелектрична проникність;

$EPS_{ox} = 3,9 \cdot EPS_0$  – діелектрична проникність оксиду

$EPS_{sil} = 11,7 \cdot EPS_0$  – діелектрична проникність кремнію;  
 $q = 1,60 \cdot 10^{-19}$  К – заряд електрона;  
 $k = 1,38 \cdot 10^{-23}$  Дж/°С – стала Больцмана;  
 $V_i = kT/q$  – температурний потенціал р–n переходу.

Нижче наведено співвідношення, які є загальними для усіх варіантів моделей МОН-транзистора:

крутизна управління по затвору  $KP = UO \cdot C_{ox}$  ;

крутизна прохідної характеристики МОН-транзистора  $Beta = 0,5 \cdot KP \cdot W / L$  ;

максимальна ємність оксиду  $C_{ox} = EPS_{ox} / TOX$  ;

коефіцієнт впливу потенціалу підкладки на граничну напругу  $GAMMA = \sqrt{2EPS_{sil} \cdot EPS_0 \cdot N_a / C_{ox}}$  ; поверхневий

потенціал  $PHI = 2V_i \cdot \ln(N_a / N_i)$  , де  $N_a = 10^4 NSS$  – рівень легування підкладки,  $N_i$  – власна концентрація носіїв, що визначається рівнянням  $N_i^2 = 1,5 \cdot 10^{32} T^3 \exp(-E_g / V_i)$  (при  $T = 300$

К:  $N_i = 1,45 \cdot 10^{16} \text{ м}^{-3}$ );

$F_p = \pm PHI / 2$  – нормований рівень Фермі (електростатичний потенціал, при кімнатній температурі й концентрації домішків в порядку  $10^{16} F_p \sim 0,3$  В);

$VTO = V_i \pm GAMMA \sqrt{PHI}$  – гранична напруга при нульовому зміщенні  $V_{bs} = 0$  (знак «+» відповідає підкладці n-типу, а знак «-» підкладці p-типу);

$V_{bl} = W_s - 10^4 \cdot NSS \cdot q / C_{ox}$  ;

$W_s = W_g - (F_p + 3,25 + 0,5 \cdot E_g)$  ;

$W_g = \begin{cases} 3,2 - \text{для затвору з металу} (TPG = 0); \\ 3,25 + E_g - \text{для кремнієвих затворів з тим же металом,} \\ \text{як у підкладці} (TPG = -1); \\ 3,25 - \text{для кремнієвого затвору протилежного типу} (TPG = +1), \end{cases}$

де  $V_{bl}, W_s$  — ширина заборонених зон підкладки і неметалевого оксиду;

$E_g = 1,16 - 0,000702 \cdot T^2 / (T + 1108)$  — нормована ширина заборонених зон;

$X_d = \sqrt{2 \cdot EPS_{sil} / (q \cdot 10^6 \cdot NSUB)}$  — нормована глибина об'єднаного шару (індукованого шару);

$CJ = \sqrt{0,5 \cdot EPS_{sil} \cdot q \cdot 10^6 \cdot NSUB / PB}$  — питома ємність  $p-n$  переходу;

$L_{eff} = L \cdot 2 \cdot LD$  — ефективна довжина каналу;

$W_{eff} = W \cdot 2 \cdot WD$  — ефективна ширина каналу;

$CGBO = C_{ox} \cdot L_{eff}$  — питома ємність перекриття затвор-підкладка;

$CGDO = C_{ox} \cdot L_{dd}$  — питома ємність перекриття затвор-сток;

$CGSO = C_{ox} \cdot L_{ds}$  — питома ємність перекриття затвор-джерело, де  $L_{dd}, L_{ds}$  — довжини бокових поверхонь дифузії в областях стока та джерела, очевидно, що для більшості транзисторів  $CGDO = CSGO$ .

$CJSW = CJ \cdot LD$  — питома ємність бічної поверхні переходу;

$CBS = CJ \cdot AS$  — ємність переходу джерело-підкладка при нульовому зсуві;

$CBD = CJ \cdot AD$  — ємність переходу сток-підкладка при нульовому зсуві.

Розробник може задати значення  $CBS$  та  $CBD$  або  $AS$  та  $AD$ . Якщо ці параметри не задано, вони дорівнюють нулю.

Параметри еквівалентних схем  $KP, GAMMA, PHI$  та ін. розраховують по наведеним вище виразам через фізичні параметри  $NSUB, TOX$  та ін., якщо в опису МОН-транзистора за допомогою директиви MODEL відсутній будь-який параметр.

Статичний режим МОН-транзистора. Моделі усіх типів характеризує нульовий струм затвора  $=0$  і струмом підкладки

$I_b = I_{bs} + I_{bd}$ , де  $I_{bs}, I_{bd}$  — струми переходів підкладка-джерело та підкладка-сток.

$$I_{bs} = I_{ss} \left[ \exp \left( \frac{V_{bs}}{N \cdot V_t} - 1 \right) \right]$$

$$I_{bd} = I_{ds} \left[ \exp \left( \frac{V_{bd}}{N \cdot V_t} - 1 \right) \right]$$

При завданні нульового значення одного з параметрів  $JS$ ,  $AS$ ,  $AD$  приймають  $I_{ss}=I_{ds}=IS$ , в протилежному випадку  $I_{ss} = AS \cdot JS + PS \cdot JSSW$ ,  $I_{ds} = AD \cdot JS + PD \cdot JSSW$ . Струм стока  $I_d = I_{bd} - I_{drain}$ , струм джерела  $I_s = I_{bs} + I_{drain}$ .

Струм  $I_{drain}$  визначають співвідношеннями

1. Для моделі першого варіанту в нормальному режимі () використовують рівняння Шихмена-Ходжеса:

$$I_{drain} = \begin{cases} 0 & \text{при } V_{gd} - V_{to} \leq 0; \\ \text{Beta} \cdot (1 + \text{LAMBDA} \cdot V_{ds}) \cdot (V_{gs} - V_{to})^2 & \text{при } 0 < V_{gd} - V_{to} \leq -V_{ds}; \text{ (режим насич.)} \\ \text{Beta} \cdot (1 + \text{LAMBDA} \cdot V_{ds}) \cdot V_{ds} \cdot [2(V_{gs} - V_{to}) - V_{ds}] & \text{при } V_{ds} < V_{gs} - V_{to} \text{ (лін. режим),} \end{cases}$$

де  $BETA = 0,5 \cdot KP \cdot W/L$  – крутизна прохідної характеристики МОН-транзистора;  $V_{to} = VTO + GAMMA \cdot \left[ \sqrt{PHI - V_{BS}} - \sqrt{PHI} \right]$  – гранична напруга.

Граничний зсув позитивний (негативний) в режимі збагачення і негативно (позитивно) в режимі збіднення для транзистора с  $n$ -каналом ( $p$ -каналом).

Для області зворотного зсуву ( $V_{ds} < 0$ )

$$I_{drain} = \begin{cases} 0 & \text{при } V_{gd} - V_{to} \leq 0; \\ -\text{Beta} \cdot (1 - \text{LAMBDA} \cdot V_{ds}) \cdot (V_{gd} - V_{to})^2 & \text{при } 0 < V_{gd} - V_{to} \leq -V_{ds}; \\ \text{Beta} \cdot (1 - \text{LAMBDA} \cdot V_{ds}) \cdot V_{ds} \cdot [2(V_{gd} - V_{to}) + V_{ds}] & \text{при } 0 < -V_{ds} \leq V_{gd} - V_{to}, \end{cases}$$

де  $V_{to} = VTO + GAMMA \cdot \left[ \sqrt{PHI - V_{bd}} - \sqrt{PHI} \right]$ .

Параметр  $LAMBDA$  характеризує нахил вихідної характеристики транзистора в області насичення. У відповідності з рис. 1.22 він визначається згідно рівняння:

$$\frac{I_{d2}}{I_{d1}} = \frac{1 + LAMBDA \cdot V_{d2}}{1 + LAMBDA \cdot V_{d1}},$$

де  $V_{d1} = V_{t0} + 1$ , В,  $V_{d2} = V_{t0} + 3$ , В, при  $V_{gs} = V_{t0} + 1$ , В.

Рисунок 1.22 – Визначення параметра модуляції довжини каналу

Крутизна МОН-транзистора  $KP$  визначається переважно рухомістю носіїв, товщиною і матеріалом діелектрика. Оскільки в моделі першого рівня параметри технологічного процесу ігноруються, завдання параметрів  $TOX$  та  $UO$  не змінює значення  $KP$ . Попередньо встановлене значення  $KP=20$  мкА/В<sup>2</sup>.

Параметри  $VTO$ ,  $GAMMA$ ,  $KP$  визначають з експериментальних даних (рис. 1.23) за виразами

Рисунок 1.23– Дані для визначення параметрів  $VTO$ ,  $GAMMA$ ,  $KP$

$$VTO = V_0, \quad KP^* = W/L \cdot KP, \quad \sqrt{KP^*} = \frac{\sqrt{2I_d}}{V_{GS} - V_0},$$

$$GAMMA = \frac{V_2 - V_0}{\sqrt{2 + PHI} - \sqrt{PHI}}.$$

Аналітична залежність  $GAMMA$  від параметрів матеріалу наведена вище.

2. Для моделі другого рівня граничну напругу розраховують за більш точним виразом, запропонованим Мейером:

$$V_{to} = VTO + \gamma_d \sqrt{PHI - V_{bs}} (ETA - 1) \cdot (PHI - V_{bs}) + V_t \cdot XN$$

Коефіцієнт  $\gamma_d$  враховує співвідношення геометричних розмірів каналу і металічної глибини переходу:

$$\gamma_d = \begin{cases} GAMMA \text{ при } XJ = 0; \\ GAMMA \cdot (1 - A_s - A_d) \text{ при } XJ > 0; \end{cases}$$

$$A_s = 0,5 \left( \sqrt{1 + \frac{2W_s}{XJ}} - 1 \right) XJ / L_{eff};$$

$$A_d = 0,5 \left( \sqrt{1 + \frac{2W_d}{XJ}} - 1 \right) XJ / L_{eff};$$

$$W_s = X_d \sqrt{PHI - V_{bs}};$$

$$W_d = X_d \sqrt{PHI - V_{ds} - V_{bs}}.$$



Вплив ширини каналу враховують за допомогою коефіцієнта

$$ETA = 1 + 0,25\pi DELTA \cdot \left( \frac{EPS_{sil}}{EPS_{ox}} \right) TOX / W.$$

Температурна залежність граничної напруги враховують за допомогою коефіцієнта

$$XN = 1 + q \cdot 10^4 NFS \cdot C_{OX} + \frac{0,5\gamma_d(PHI - V_{bs})}{\pi^{3/2}} + 0,25 \cdot \left[ \frac{(PHI + V_{ds} - V_{bs})}{\sqrt{1 + 2W_d / XJ}} \right] + \frac{(PHI - V_{bs})}{\sqrt{1 + 2W_s / XJ}} \cdot \frac{X_d \cdot \pi^{3/2}}{L_{eff}}.$$

3. Для моделі третього рівня використовують полумпіричні рівняння статичних характеристик МОН-транзистора, де гранична напруга дорівнює

$$V_{to} = VTO - \sigma \cdot V_{ds} + GAMMA \cdot F_s \cdot \sqrt{PHI - V_{bs}} + F_n \cdot (PHI - V_{bs}) + V_t \cdot XN$$

де  $\sigma = ETA \cdot (8,15 \cdot 10^{-22}) / C_{OX} / (L_{eff})^3$  – коефіцієнт статичного зворотного зв'язку;

$F_s$  – коефіцієнт, що враховує вплив геометрії прибору на скорочення каналу;

$F_n = 0,5 \cdot DELTA \cdot \pi \cdot EPS_{sil} / C_{ox} / W$  – коефіцієнт, що враховує вплив ширину каналу на граничну напругу;

$$XN = 1 + 0,5 \cdot (GAMMA \cdot F_s \cdot \sqrt{PHI - V_{bs}} + F_n \cdot (PHI - V_{bs})) / (PHI - V_{bs}) + q \cdot 10^4 NFS /$$

Ефективна рухомість носіїв з урахуванням напруженості поля в каналі:

$$\mu_{eff} = UO \cdot \mu_{fact}, \text{ де } \mu_{fact} = 1 / (1 + THETA \cdot (V_{os} - V_{to})).$$

Напруга насиченості визначається за формулою

$$VD_{sat} = V_a + V_b - \sqrt{V_a^2 + V_b^2},$$

де параметр  $V_a = (V_{gs} - V_{to}) / (1 + F_b)$  враховує збільшення напруженості поля в каналі; параметр

$V_b = L_{eff} \cdot VMAX / \mu_{eff}$  враховує обмеження швидкості носіїв у каналі;

$$F_b = F_n + 0,5 \cdot GAMMA \cdot F_s / \sqrt{PHI - V_{bs}}.$$

Ефективна довжина каналу визначається відношенням:

$$L_{eff} = \begin{cases} L \text{ при } V_{ds} < VD_{sat}; \\ L - \Delta L \text{ при } V_{ds} > VD_{sat}, \end{cases}$$

де

$$\Delta L = \begin{cases} L_1 \text{ при } L_1 < 0,5L; \\ XL(1 - 0,5XL/L_1), \text{ якщо } L_1 > 0,5L; \end{cases}$$

$$L_1 = \begin{cases} \sqrt{A}, \text{ якщо параметр } VMAX \text{ незаданий}; \\ \sqrt{A + B^2}, \text{ якщо параметр } VMAX \text{ заданий}; \end{cases}$$

$$A = KAPPA \cdot (V_{ds} - VD_{sat}).$$

Струм  $I_{drain}$  визначається з рівнянь:

$$I_{drain} = \begin{cases} 0 \text{ при } V_{gs} \leq V_{t0}; \\ Beta[V_{gs} - V_{t0} - 0,5(1 + F_b) \cdot V_{ds}] \cdot \frac{V_{ds}}{\left(1 + \frac{V_{ds} \mu_{eff} \cdot VMAX}{L_{eff}}\right)} \text{ при } V_{gs} \geq V_{t0}. \end{cases}$$

4. Для моделі четвертого рівня використовують коротко каналну модель польового транзистора з ізольованим затвором BSIM, розроблену в університеті Берклі. ця модель враховує такі ефекти : залежність рухомості носіїв від вертикального поля; насичення швидкості носіїв; залежність граничної напруги від напруги стоку; розподілення заряду збідненої області між стоком і джерелом; неоднорідне легування для транзисторів, виготовлених за технологією іонної імплантації; модуляцію довжини каналу; підпорогову провідність; залежність усіх параметрів від геометрії транзистора.

В моделі BSIM струм стока представляють сумой струмів потужної та слабкої інверсії (під підкладковий режим):

$$I_{drain} = I_{drainS} + I_{drainW}.$$

Це дозволяє забезпечити неперервність перших похідних і за рахунок цього значно покращити сходження розрахунку статистичного режиму по зрівнянню з модулями 2 та 3-го рівнів.

У режимі потужної інверсії гранична напруга визначається рівнянням

$$V_{t0} = V_{FB} + PHI + K1\sqrt{PHI \cdot V_{bs}} - K2(PHI - V_{bs}) - \eta V_{ds}.$$

Параметр  $K1$  схожий з параметром  $GAMMA$  у моделі першого рівня. Параметр  $K1$ ,  $K2$  сумісно модулюють неоднорідність легування. Параметр  $ETA$  частково модулює зменшення довжини каналу і граничної напруги через статичну індукцію.

Струм стоку в режимі сильної інверсії визначається наступним співвідношенням.

У режимі відсічки ( $V_{gs} \leq V_{t0}$ )  $I_{drainS} = 0$ .

У лінійній області ( $V_{gs} > V_{t0}$  і  $0 < V_{ds} < VD_{sat}$ )

$$I_{drainS} = \frac{MUS}{1 + U_0(V_{gs} - V_{t0})} \cdot \frac{C_{ox} \cdot W / L}{(1 + U_1 \cdot V_{ds} / L)} \cdot [(V_{gs} - V_{t0}) \cdot V_{ds} - \frac{a}{2} \cdot (V_{ds})^2],$$

де

$$a = 1 + \frac{g \cdot K1}{2\sqrt{PHI - V_{bs}}};$$

$$g = 1 \cdot \frac{1}{1,744 + 0,8364(PHI - V_{bs})}.$$

Незвичний вид формули для параметру  $a$  пов'язаний з тим, що застосовано дробно-поліноміальну апроксимацію добре відомою трансцендентної залежності

$$a = 2 / 3[(V_{ds} + PHI - V_{bs})^{3/2} - (PHI - V_{bs})^{3/2}].$$

У режимі насичення ( $V_{gs} > V_{t0}$  і  $V_{ds} \geq VD_{sat}$ )

$$I_{drainS} = \frac{\mu_0}{1 + U_0 \cdot (V_{gs} - V_{t0})} \cdot \frac{C_{ox} \cdot W / L (V_{gs} - V_{t0})^2}{2a \cdot K},$$

де

$$K = (1 + V_c + \sqrt{1 + 2 \cdot V_c}) / 2, \quad V_c = \frac{U_1}{L} \cdot \frac{V_{gs} - V_{t0}}{a} - \text{швидкість носіїв};$$

$VD_{sat} = (V_{gs} - V_{t0}) / (a\sqrt{K})$  – напруга насичення.

Складова струму стоку за рахунок слабкої інверсії дорівнює

$$I_{drainW} = \frac{I_{exp} \cdot I_{lim}}{I_{exp} + I_{lim}},$$

де

$$I_{exp} = \mu_0 C_{ox} \frac{W}{L} \left(\frac{kT}{q}\right)^2 \exp\left[1,8 + \frac{(V_{gs} - V_{t0})q}{kTn}\right] \cdot [1 - \exp(-\frac{V_{ds} \cdot q}{kT})],$$

$$I_{lim} = \frac{\mu_0 \cdot C_{ox}}{2} \frac{W}{L} \left(\frac{3kT}{q}\right)^2.$$

Показник ступеня 1,8 обраний експериментально для найкращої сумісності зі складовою струму сильної інверсії. Параметри підготовки режиму  $NO$ ,  $NB$  та  $ND$  визначають коефіцієнт нахилу під граничної характеристики:

$$\eta = NO + NB \cdot V_{bs} + ND \cdot V_{ds}.$$

Параметри  $VFB$ ,  $PHI$ ,  $K1$  і  $K2$  вважаються незалежними від режиму по постійному струму. Інші параметри розраховуються за формулами:

$$U_0 = X2MZ + X2U0 \cdot V_{bs};$$

$$U_1 = U1 + X2U1 \cdot V_{bs} + X3U1(V_{ds} - V_{dd});$$

$$h = ETA + X2E \cdot V_{bs} + X3E(V_{ds} - V_{dd}),$$

де  $V_{dd}$  – напруга живлення, за якої вимірювали параметри транзистору.

Залежність параметру  $\mu_0$  від напруги  $V_{ds}$  і  $V_{bs}$  апроксимується квадратичним поліномом, коефіцієнти якого визначаються на основі розрахунку  $\mu_0$  при  $V_{ds} = 0$  і  $V_{ds} = V_{dd}$  за формулами.

$$\mu_0 \Big|_{V_{ds} = 0} = MUZ + X2MZ \cdot V_{bs};$$

$$\mu_0 \Big|_{V_{ds} = V_{dd}} = MUS + X3MS \cdot V_{bs}.$$

Динамічні властивості МОН транзистора. Враховані в модель ємності збідненого шару підкладки–джерело  $C_{bs}$  і ємність збідненого шару підкладки–сток  $C_{bd}$ . Ці ємності складаються з ємності даної поверхні та бічної поверхні переходу. Якщо параметри  $CBS$  і  $CBD$  не задано, то їх розраховують за формулами:

$$C_{bs} = AS \cdot CJ \cdot C_{bej} + PS \cdot CJSW \cdot C_{bs} + TT \cdot G_{bs};$$

$$C_{bd} = AS \cdot CJ \cdot C_{bdj} + PD \cdot CJSW \cdot C_{bda} + TT \cdot G_{ds}.$$

У протилежному випадку (тобто при заданих абсолютних значення  $CBS$  і  $CBD$ ):

$$C_{bs} = CBS \cdot C_{bej} + PS \cdot CJSW \cdot C_{bas} + TT \cdot G_{bs};$$

$$C_{bd} = CBD \cdot C_{bdj} + PD \cdot CJSW \cdot C_{bds} + TT \cdot G_{ds},$$

де  $C_{bs} = dI_{bs} / dV_{bs}$ ,  $G_{ds} = dI_{bd} / dV_{bd}$  — диференціальні провідності в околицях робочої точки.

Скалярні множники, що відображають залежність бар'єрних ємностей від напруги зміщення, розраховуються наступним чином:

при  $V_{bs} \leq FC \cdot PB$

$$C_{bsj} = (1 - V_{bs} / PB)^{-MJ};$$

$$C_{bss} = (1 - V_{bs} / PBSW)^{-MJSW};$$

при  $V_{bs} \geq FC \cdot PB$

$$C_{bsj} = (1 - FC)^{-(1+MJ)} \cdot [1 - FC \cdot (1 + MJ) + MJ \cdot V_{bs} / PB];$$

$$C_{bss} = (1 - FC)^{-(1+MJSW)} \cdot [1 - FC \cdot (1 + MJSW) + MJSW \cdot V_{bs} / PBSW];$$

при  $V_{bd} \leq FC \cdot PB$

$$C_{bdj} = (1 - V_{bd} / PB)^{-(MJ)};$$

$$C_{bds} = (1 - V_{bd} / PBSW)^{-MJSW};$$

при  $V_{bd} > FC \cdot PB$

$$C_{bdj} = (1 - FC)^{-(1+MJ)} \cdot [1 - FC \cdot (1 + MJ) + MJ \cdot V_{bd} / PB];$$

$$C_{bds} = (1 - FC)^{-(1+MJSW)} \cdot [1 - FC \cdot (1 + MJSW) + MJSW \cdot V_{bd} / PBSW].$$

Ємності затвор-джерело, затвор-сток і затвор-підкладка розраховуються за формулами:  $C_{gs} = CGSO \cdot W$ ,  $C_{gd} = CGDO \cdot W$ ,  $C_{bd} = CGBO \cdot L$ .

Температурні залежності параметрів елементів еквівалентної схеми МДН-транзистора враховуються з допомогою наступних виразів:

$$\begin{aligned} IS(T) &= IS \cdot \exp\{[EG(T_{nom}) \cdot T / T_{nom} - EG(T) / V_t]\}; \\ JS(T) &= JS \cdot \exp\{[EG(T_{nom}) \cdot T / T_{nom} - EG(T) / V_t]\}; \\ JSSW(T) &= JSSW \cdot \exp\{[EG(T_{nom}) \cdot T / T_{nom} - EG(T) / V_t]\}; \\ PB(T) &= PB \cdot T / T_{nom} - 3V_t \ln(T / T_{nom}) - EG(T_{nom}) \cdot T / T_{nom} + EG(T); \\ PBSW(T) &= PBSW \cdot T / T_{nom} - 3V_t \ln(T / T_{nom}) - EG \cdot T / T_{nom} + EG(T); \\ PHI(T) &= PHI \cdot T / T_{nom} - 3V_t \ln(T / T_{nom}) - EG \cdot (T_{nom}) \cdot T / T_{nom} + EG(T); \\ EG(T) &= 1,16 - 0,000702 \cdot T^2 / (T + 1108); \\ CBD(T) &= CBD \{1 + MJ[0,0004(T - T_{nom}) + 1 - PB(T) / PB]\}; \\ CBS(T) &= CBS \{1 + MJ[0,0004(T - T_{nom}) + 1 - PB(T) / PB]\}; \\ CJ(T) &= CJ \{1 + MJ[0,0004(T - T_{nom}) + 1 - PB(T) / PB]\}; \\ CJSW(T) &= CJSW \{1 + MJSW[0,0004(T - T_{nom}) + 1 - PB(T) / PB]\}; \\ KP(T) &= KP \cdot (T / T_{nom})^{-3/2}; UO(T) = UO \cdot (T / T_{nom})^{-3/2}; \\ KF(T) &= KF \cdot PB(T) / PB; AF(T) = AF \cdot PB(T) / PB; \\ MIUS(T) &= MUS \cdot (T / T_{nom})^{-3/2}; MUZ(T) = MUZ \cdot (T / T_{nom})^{-3/2}; \\ X3MS(T) &= X3MS \cdot (T / T_{nom})^{-3/2}, \end{aligned}$$

де  $T_{nom}$  – номінальна температура, при якій виміряні параметри транзистора;  $T$  – дана температура.

Лінійна еквівалентна схема з урахуванням джерел шуму наведено на рис. 1.23. Джерело струму  $I_{шо}$ , що характеризує дробовий та фліккер-шум, має спектральну щільність:

$$S_d = 8kT \cdot G_m / 3 + KF \cdot I_{drain}^{AF} / (f \cdot K_{chan}),$$

де  $G_m = dI_{drain} / dV_{gs}$  – диференціальна провідність в робочій точці за постійним струмом;  $K_{chan} = (L_{eff})^2 \cdot EPS_{sil} / TOX$ .

Рисунок 1.23 – Лінійна еквівалентна схема МДН-транзистора

В якості прикладу приведемо опис моделі МОП – транзистора IRF140, використовуючи модель третього типу:

```
.model IRF140 NMOS (LEVEL=3 GAMMA=0 DELTA=0 ETA=0  
THETA=0 KAPPA=0  
+VMAX=0 XJ=0 TOX=100n OU=600 PHI=6 RS=46.07 m KP=20.83 u  
W=1.1 L=2 u  
+VTO=3.691 RD=8/065m RDS=400K CBD=2.329n PB=8 MJ=5 FC=5  
+CGSO=2.208n CGDO=773.2p RG=1.927 IS=12.11f)
```

Звернемо увагу, що при ввімкненні цього МДН-транзистора в описі схеми на вихідній мові програми PSpice неможна вказувати значення параметрів  $L$ ,  $W$ ,  $AD$ ,  $AS$ ,  $PD$ ,  $PS$ ,  $NRD$ ,  $NRS$ , так як частина з них вже приведена за директивою .MODEL, а для інших використані значення за змовчанням.

В практичних цілях, при проектуванні пристроїв із МДН-транзисторами зазвичай використовують модель першого рівня, якщо не потрібна висока точність. До її переваг відносять найменший час обчислення завдяки простоті математичних рівнянь. Недоліками моделі першого рівня вважають такі: модель не враховує залежність рухомості носіїв від електричного поля; модель не враховує під граничний режим; модель не враховує залежність граничної напруги від параметрів  $L$ ,  $W$ ,  $V_{ds}$ ; усі ємності розраховуються спрощено; модель не враховує неоднорідність легування.

Модель другого рівня основана на досить точному теоретичному підґрунті, але низку її параметрів важко оцінити за експериментальними даними. Модель є відносно складною, потребує збільшених обчислювальних затрат при моделюванні і має проблему збіжності при обрахунку методом Ньютона-Рафсона при розрахунку режиму за постійним струмом.

Напівемпірична модель третього рівня потребує значно менших обчислювальних затрат і її рекомендують використовувати для практичних розрахунків потужних вертикальних транзисторів із коротким каналом. Недоліком моделі є складність оцінювання параметрів за експериментальними даними.

Модель четвертого рівня (BSIM1) основана на фізиці роботи транзисторів із малими розмірами і розрахована на отримання параметрів моделей виготовлених зразків за допомогою спеціальної програми, яка керує роботою тестера напівпровідникових приладів.

Моделі п'ятого та шостого рівнів (BSIM3) точно описують субмікронні МОН-транзистори. Використання моделі шостого рівня є найбільш рекомендованим.

Модель другого рівня основана на досить точному теоретичному підґрунті, але низку її параметрів важко оцінити за експериментальними даними. Модель є відносно складною, потребує збільшених обчислювальних затрат при моделюванні і має проблему збіжності при обрахунку методом Ньютона-Рафсона при розрахунку режиму за постійним струмом.

Напівемпірична модель третього рівня потребує значно менших обчислювальних затрат і її рекомендують використовувати для практичних розрахунків потужних вертикальних транзисторів із коротким каналом. Недоліком моделі є складність оцінювання параметрів за експериментальними даними.

Модель четвертого рівня (BSIM1) основана на фізиці роботи транзисторів із малими розмірами і розрахована на отримання параметрів моделей виготовлених зразків за допомогою спеціальної програми, яка керує роботою тестера напівпровідникових приладів.

Моделі п'ятого та шостого рівнів (BSIM3) точно описують субмікронні МОН-транзистори. Використання моделі шостого рівня є найбільш рекомендованим.



## 1.5 Особливості автоматизованого проектування схем із активними функціональними блоками

Сучасний рівень розвитку електронної промисловості дозволив створити значну кількість аналогових функціональних модулів – мікросхем, які часто виконують дуже специфічні функції у складі тих чи інших електронних пристроїв. Оскільки ці специфічні аналогові мікросхеми застосовують у обмеженому, подекуди одиничному класі апаратури, імітаційні моделі таких мікросхем відсутні у програмах АСхП. Розробники програм АСхП вносять у електронні бібліотеки своїх середовищ лише дані про корпус у який упаковано специфічну мікросхему. Ця інформація містить відомості про зовнішні (геометричні, зовнішні топологічні) параметри таких специфічних мікросхем – тип корпусу, відповідно його геометричні розміри, кількість, розташування та призначення виводів. Наявність такої інформації дозволяє за допомогою середовища АСхП створювати файли креслень принципів схем та файлів монтажних схем друкованих плат. Але виконати імітаційне моделювання без наявності математичної моделі такої мікросхеми у середовищі АСхП буде неможливо.

З метою вирішення цієї проблеми, інженер-проектувальник який використовує у своїй роботі середовище автоматизованого проектування може обрати декілька шляхів. Якщо відома внутрішня структура мікросхеми, можна створити її імітаційну Spice-модель для подальшого використання при імітаційному моделюванні схем. Але виробники мікросхем захищають дані про внутрішню топологію своїх виробів, тому ця інформація досить рідко потрапляє за межі підприємства. Крім того внутрішні структури аналогових функціональних блоків досить складні навіть для реалізації у сучасних середовищах АСхП. Тому цей шлях можуть собі дозволити лише спеціалізовані підприємства які мають достатню інструментальну базу та штат професіоналів.

Інший шлях пов'язано із створенням спрощеної імітаційної функціональної моделі, або декількох моделей функціонального блоку. Така модель імітує вхідні та вихідні електричні параметри (наприклад, внутрішні вхідні та вихідні імпеданси), а функціональне призначення імітують за допомогою вбудованого апарата математичних функцій та змінних, за допомогою яких конструюють основні та допоміжні рівняння перетворення, притаманні блоку, що імітується.

У середовищі Spice вбудовано модель найбільш часто застосовуваного в електроніці аналогового блоку – операційного підсилювача, за допомогою якого можливо спроектувати велику кількість функціональних одиниць електронних пристроїв – власне підсилювачів, фільтрів, генераторів, дискримінаторів, модулів, що реалізують математичні операції та інших. Потрібно також зазначити, що найбільш вдалі моделі операційних підсилювачів вбудовують у внутрішні структури більш складних пристроїв – наприклад мікро– та наноелектронних вимірювальних перетворювачів (сенсорів). Тому проектування схем на операційних підсилювачах є дуже актуальним і сьогодні.

Розглянемо внутрішню принципову схему одного із найпростіших операційних підсилювачів (рис. 1.25).

Рисунок 1.25 – Внутрішня схема операційного підсилювача [13]

У схемі на рис.1.25 показані основні складові частини операційного підсилювача. Вхідний диференційний підсилювальний каскад зібрано на біполярних транзисторах VT1 і VT2. В ланцюгу колекторів диференційного каскаду включено рефлектор струму (струмове дзеркало) на транзисторах VT3 і VT4, яке виконує функцію динамічного навантаження. Проміжні підсилювальні каскади : каскад по схемі із загальним колектором на транзисторі VT5, каскад по схемі з загальним емітером на транзисторі VT6. Вихідний підсилювач потужності класу В зібрано на транзисторах VT7 і VT8. Джерела стабільного струму (ДСС1 і ДСС2) у ланцюгу емітерів диференційного каскаду та ланцюгу бази транзистору VT7 виконують функції електронного еквіваленту резистора великого номіналу.

Це схема операційного підсилювача, який було побудовано на початку 70-х років минуло сторіччя. Сучасні мікросхеми операційних підсилювачів містять набагато більше елементів.

Якщо скласти точну імітаційну модель структури, показаної на рис. 1.25 знадобиться не менше десяти біполярних транзисторів та чотирьох діодів, оскільки навіть спрощені схеми ДСС окрім резисторів містять як мінімум один транзистор та один діод. Моделі та їх математичні залежності біполярних транзисторів, наведені у п.1.3 задалегідь наочно демонструють складність

моделювання такої схеми. Тому у програмах АСхП запропоновано використання декількох еквівалентних схем операційних підсилювачів. Ці еквівалентні схеми розділені за рівнем складності. Самі прості моделі дозволяють моделювати зміну лише основних параметрів операційного підсилювача : власний коефіцієнт підсилення, вхідний та вихідний опір, частоту одиничного підсилення, швидкість зростання вихідної напруги. Потрібно зазначити, що у програмах АСхП операційні підсилювачі лише графічно представлено у вигляді моноблоку. Командний файл у форматі Spice описує ОП як підсхему (subcircuit), яка має “зовнішні” виводи. Підсхема власне і є еквівалентною схемою пристрою, що імітують.

На рис.1.26 зображено спрощену модель операційного підсилювача [14] яка дозволяє імітувати : власний коефіцієнт підсилення ОП, частоту першого полюса АЧХ, параметри вихідної ланки ОП.

Рисунок 1.26– Спрощена модель операційного підсилювача

Полюсом АЧХ називають значення частоти на якій відбувається зміна (зменшення) попереднього значення коефіцієнта підсилення ОП[15].

Звісно, що структура представлена на рис.1.26 не дозволяє відтворити поведінку багатьох параметрів реальних операційних підсилювачів, тому розвиток обчислювальної техніки дозволяв поступово ускладнювати еквівалентні схеми ОП, застосовувані для імітації його роботи у пакетах АСхП. Розглянемо модель Бойла для операційного підсилювача [16] яка наведена на рис. 1.27. Ця модель була базовою при формуванні еталонних версій програми Spice. На рис. 1.27 показано варіант схеми із вхідним диференціальним підсилювачем на біполярних транзисторах. Деякі пакети АСхП дозволяють конструктору обрати тип вхідного каскаду – на біполярних транзисторах чи польових. За допомогою цієї моделі можливо визначити чотири режими роботи ОП : параметри великого сигналу, коефіцієнт підсилення без наявності зворотного зв'язку (для двох типів вхідних каскадів), фазочастотна

характеристика без зворотного зв'язку, максимальне значення вихідних опорів.

Позитивним фактором створення імітаційних моделей схем пристроїв із операційними підсилювачами є те, що конструктору – користувачу програм АСхП не потрібно шукати або самостійно розраховувати параметри тої чи іншої моделі, застосованої у програмі. Для полегшення процесу автоматизованого проектування пристроїв користувачу потрібно лише ввести довідникові параметри підсилювача у відповідні поля вікна настроювання параметрів ОП, якщо його модель відсутня у базі компонентів програмного пакету АСхП. У таблицях 1.8–1.12 наведені довідникові дані ОП, які водить конструктор – користувач пакету АСхП та перелік параметрів макромоделі (моделі Бойла), які розраховує пакет. На рис.1.28, рис.1.29 наведено вікна настроювання параметрів операційного підсилювача програми Multisim та EWB. Додамо, що ці два пакети АСхП не дозволяють обрати тип транзисторів вхідного каскаду.

Рекомендовано при проектуванні схем із операційними підсилювачами дотримуватися загальної методи проектування за допомогою програмних середовищ АСхП – тобто виконувати по-ланкове проектування, обираючи спочатку спрощені моделі операційних підсилювачів та при наявності позитивних результатів моделювання виконувати поступове ускладнення імітаційної моделі схеми. Середовища АСхП дозволяють обирати різні моделі ОП, які відрізняють навіть за зовнішнім виглядом – сама проста модель ОП містить лише три виводи: два входи та один вихід; наступна по складності модель містить ще два додаткові виводи для підключення джерел живлення. Ще більш складніша модель містить крім вже вказаних ще й виводи для підключення зовнішньої корекції зсуву нуля або частотних характеристик.

Рисунок 1.27 – Стандартна макромодель операційного підсилювача (модель Бойла)

Таблиця 1.8

## Параметри режиму великого сигналу

Позначення параметру	Параметр	Параметри елементів моделі	
		Ім'я	Значення по умовчанню
+Uип	напруга живлення позитивного джерела	VC VE	2 В 2В
-Uип	напруга живлення негативного джерела		
+Uвых.макс	максимальне значення вихідної напруги позитивної полярності		
-Uвых.макс	максимальне значення вихідної напруги негативної полярності		
+V <sub>Увых</sub> (В/мкс)	максимальна швидкість зростання вихідної напруги позитивної полярності		
-V <sub>Увых</sub> (В/мкс)	максимальна швидкість зростання вихідної напруги негативної полярності		
P <sub>пот</sub>	споживана потужність у статичному режимі		

Таблиця 1.9

## Параметри режиму при відсутності зворотного зв'язку (вхідний каскад на біполярних транзисторах)

Позначення параметру	Параметр	Параметри елементів моделі	
		Ім'я	Значення по умовчанню
C <sub>кор</sub>	ємність корекції (при значенні 30пФ)	BF1 BF2	75 75

$I_{cm}$ , (мкА)	вхідний струм зсуву (при 100нА)	C2	30пФ
$K_{yu}$	коефіцієнт підсилення ОП (без ланцюга зворотного зв'язку) (при значенні 200000)	CEE GA	0 $189 \cdot 10^{-6}$
$f_l$	частота одиничного підсилення (1МГц)	GSM IS1 IS2	$1,9 \cdot 10^{-9}$ $8 \cdot 10^{-16}$ $8 \cdot 10^{-16}$
КОСС	коефіцієнт ослаблення синфазного сигналу (100000)	IEE RC	$15 \cdot 10^{-6}$ 5305
$\Delta I_{cm}$ , (мкА)	різниця вхідних струмів зсуву нуля	RE	1832
$U_{cm}$ , (мкА)	напруга зсуву нуля	REE RP	13810 18160

Таблиця 1.10

**Параметри режиму при відсутності зворотного зв'язку (вхідний каскад на польових транзисторах)**

Позначення параметру	Параметр	Параметри елементів моделі	
		Ім'я	Значення по умовчанню
$C_{кор}$	ємність корекції (при значенні 10пФ)	ВЕТ А	$789 \cdot 10^{-6}$
$K_{yu}$	коефіцієнт підсилення ОП (без ланцюга зворотного зв'язку) (при значенні 200000)	C2 CSS	10 pF 0
$f_l$	частота одиничного підсилення (1МГц)	GA	$63 \cdot 10^{-6}$
КОСС	коефіцієнт ослаблення синфазного сигналу (100000)	GSM IS	$63 \cdot 10^{-6}$ $15 \cdot 10^{-12}$
$\Delta I_{cm}$ , (мкА)	різниця вхідних струмів зсуву нуля	SS	$5 \cdot 10^{-6}$
$U_{cm}$ , (мкА)	напруга зсуву нуля	D	15.9
$I_{cm}$ , (мкА)	вхідний струм зсуву (при 30пА)	RSS	$40 \cdot 10^6$

		RP	18000
--	--	----	-------

Таблиця 1.11

**Параметри режиму фазочастотної характеристики (без ланцюга зворотного зв'язку)**

Позначення параметру	Параметр	Параметри елементів моделі	
		Ім'я	Значення по умовчанню
$\varphi$	Запас по фазі на частоті одиничного підсилення ( $60^0$ )	C1	8,6 пФ

Таблиця 1.12

**Параметри режиму максимального значення вихідних опорів**

Позначення параметру	Параметр		Параметри елементів моделі	
			Ім'я	Значення по умовчанню
$R_{\text{вих}} = R_{\text{вих.dc}} + R_{\text{вих.ac}}$	$R_{\text{dc}} = 0,7 \cdot R_{\text{вих}}$	вихідний опір на низьких частотах	R01 R02	50 50
	$R_{\text{ac}} = 0,3 \cdot R_{\text{вих}}$	вихідний опір на високих частотах	GB	424.4
$I_{\text{кз}}$	максимальний струм короткого замикання			

Рисунок 1.28 – Екранний інтерфейс налаштування довідникових параметрів ОП для розрахунку параметрів його моделі у програмі EWB



Рисунок 1.29 – Екранний інтерфейс налаштування довідникових параметрів ОП для розрахунку параметрів його моделі у програмі Multisim

Таким чином можна сказати, що середовища АСхП мають досить потужний апарат для автоматизованого проектування схем пристроїв різного призначення, що містять операційні підсилювачі.

#### 1.6 Автоматизація аналізу імітаційних структур схем радіоелектронних пристроїв в програмах EDA.

Математичний апарат програми SPICE дозволяє реалізувати декілька різновидів аналізу імітаційних моделей електронних схем. В залежності від версії програми SPICE, яку використовує конкретна програма EDA, склад інструментарію моделювання може відрізнятися, але в цілому існує стандартний набір. Перелічимо ці види аналізу :

- аналіз ланцюга за постійним струмом у “робочій точці” – Bias Point (або в деяких програмах відомий як Operational Point);
- багатоваріантний аналіз ланцюга за постійним струмом (при варіації джерел постійної напруги, струму, температури або інших параметрів) – DC Sweep;
- багатоваріантний аналіз лінеаризованого ланцюга в частотній області при дії декількох сигналів – AC Sweep;
- спектральний аналіз – Fourier Analysis;
- аналіз перехідних процесів при дії сигналів різної форми – Transient Analysis;
- аналіз мало сигнальних передаточних функцій в режимі за постійним струмом –Transfer Function Analysis;
- аналіз спектральної щільності внутрішнього шуму – Noise analysis;
- аналіз чутливості характеристик ланцюга до варіацій параметрів компонентів в режимі за постійним струмом – Sensitivity Analysis;
- статистичний аналіз за методом Монте-Карло – Monte-Carlo Analysis;
- аналіз найгіршого випадку – Worst Case Analysis.

Представлення результатів аналізу може бути у вигляді графіків та числовим (або таблиць), або тільки числовим (табличним). Оскільки для реалізації вказаних видів аналізу імітаційних структур електронних схем використовують певні математичні методи, то інженеру, якому потрібно імітувати роботу своєї розробки необхідно пам'ятати, що найбільш точна реалізація таких методів залежить від декількох факторів. По-перше, це апаратні можливості комп'ютера на якому виконують автоматизоване проектування розробки – це стосується об'єму оперативної пам'яті, “віртуальної” пам'яті жорсткого диску, тактової частоти та потужності центрального процесору та деякі інші характеристики. По-друге, це необхідність врахування недоліків реалізованих математичних методів, шляхом встановлення оптимальних для кожного математичного методу початкових, прикінцевих параметрів, обмежень та спрощень. На жаль, обсяг даного навчального видання не дозволяє найбільш повно розглянути особливості кожного з методів аналізу, тому ми надамо узагальнене тлумачення основ математичної реалізації вказаних методів аналізу, а також розглянемо формат опису директив моделювання у програмі SPICE.

**Аналіз ланцюга за постійним струмом у “робочій точці” – Bias Point.** Параметри “робочої точки” імітаційної моделі схеми розроблюваного пристрою за постійним струмом розраховують попередньо перед виконанням будь-якого виду аналізу, незалежно від того, встановив розробник цей вид аналізу у відповідному вікні аналізу або у тестовому завданні на моделювання чи ні. Але від включення, або виключення **Bias Point** у завданні на моделювання, залежить яку саме інформацію запише програма SPICE у вихідний файл результатів моделювання. Якщо **Bias Point** виключено, то у вихідний файл буде записано інформацію про значення напруг у вузлах аналогової частини схеми та стани цифрової частини (рівні логічного нуля або одиниці). Якщо **Bias Point** включено, то вихідний файл записують таку інформацію :

- перелік напруг у вузлах аналогової частини схеми;
- перелік станів цифрової частини схеми;
- струми усіх джерел напруги та їхню сумарну потужність;

- перелік мало сигнальних параметрів для всіх пристроїв.

Для того, щоб обчислити параметри робочої точки (це також справедливо для DC Sweep та Transient Analysis) вбудований математичний апарат SPICE вирішує систему нелінійних алгебраїчних рівнянь (СНАР), які описують поведінку схеми. Для цього SPICE використовує ітеративний алгоритм Ньютона-Рафсона, для чого потрібно початкове наближення до вирішення, яке у наступних ітераціях покращується до тих пір доки послідовні значення напруг та струмів зійдеться до цього результату. У деяких випадках SPICE не може знайти вирішення рівнянь нелінійного ланцюга, що є проявом так званої “проблеми збіжності”. Є деякі рекомендації для усунення цієї проблеми. За відсутності збіжності рекомендується збільшити максимальну кількість ітерацій, тобто параметр IТL1, який є доступним у глобальних налаштуваннях (за умовчанням IТL1=40). Для підвищення швидкості збіжності рекомендується встановлювати початкові значення нульових потенціалів, якомога більш близьких до очікуваних за постійним струмом (за відсутності цієї директиви всі вузлові потенціали на початковій ітерації вважаються нульовими). Якщо рішення методом Ньютона-Рафсона не збігається, програма автоматично переходить до методу варіації напруги джерела живлення, при цьому ціною збільшення витрат машинного часу забезпечує збіжність рішення у більшості випадків. Іноді спочатку застосовується метод варіації мінімальної провідності параметру GMIN та після цього, у випадку його невдачі – метод варіації напруги.

Директива моделювання має дуже простий формат – OP (або BP в певних версіях SPICE), яку вказують без будь яких атрибутів.

#### **Багатоваріантний аналіз ланцюга за постійним струмом DC Sweep.**

Цей вид аналізу виконують за постійним струмом при варіації:

- незалежних джерел постійного струму та/або наруги;
- температури;
- параметрів моделей компонентів схеми;
- глобальних параметрів, які визначають при зазначенні директиви глобальних параметрів.

Вказані компоненти можуть бути використані в якості варіюємої змінної. Необхідно додати, що з параметрів моделей компонентів можна використовувати всі параметри окрім таких: температурні коефіцієнти моделей (такі як TC1, TC2, TCE) резисторів та параметри довжина (L) та ширини (W) каналу МДП транзистора. В цілому, с точки зору застосованого алгоритму моделювання, виконання цього виду аналізу є подібним до **Bias Point**, при цьому враховується варіація параметрів. Характер варіації змінних може бути лінійним, логарифмічним (масштаб зміни октавами та декадами) та задаватися переліком значень. Розглянемо шаблон завдання багатоваріантного аналізу ланцюга за постійним струмом DC Sweep у форматі SPICE. Лінійний характер варіації змінної описують за допомогою шаблону:

```
.DC [LIN] <sweep variable name>  
+ <start value> <end value> <increment value>  
+ [nested sweep specification]
```

В цьому шаблоні атрибут <sweep variable name> – ім'я змінної, що змінюється у вказаних діапазонах, від початкового – <start value> до кінцевого – <end value> з кроком зміни – <increment value>. Атрибут шаблону [nested sweep specification] – дозволяє вказати ім'я другої змінної (вкладений компонент), а також її початкові, кінцеві значення із кроком. При зазначенні такої другої (вкладеної) змінної, перша змінна змінюється у вказаних межах для кожного значення цієї другої змінної. За такою формою можливо, наприклад, побудувати сімейство вольт-амперних характеристик певного транзистора. У квадратних дужках вказують, як ми вже знаємо, необов'язкові атрибути, тобто форму із вкладеним компонентом застосовують у потрібних випадках. Також зрозуміло, що атрибут [LIN] можливо також не зазначати. Наведемо приклади завдання на проведення багатоваріантного аналізу ланцюга за постійним струмом DC в лінійному масштабі:

```
.DC VIN -.25 .25 .05  
.DC LIN I2 5mA -2mA 0.1mA  
.DC RES RMOD(R) 0.9 1.1 .01V першому прикладі задано  
діапазон зміни джерела напруги від мінус 250 мВ до + 250 мВ із  
кроком 50 мВ. У другому прикладі задано діапазон зміни джерела
```

струму від 5мА до мінус 2 мА із кроком 0,1 мА. В третьому прикладі задано зміну опору резистору від 0.9 до 1,1 Ом із кроком 10 мОм.

Логарифмічний характер варіації змінної описують за допомогою шаблону:

```
.DC <logarithmic sweep type> <sweep variable name>  
+ <start value> <end value> <points value>  
+ [nested sweep specification]
```

В цьому шаблоні атрибут <logarithmic sweep type> – визначає конкретний тип логарифмічної зміни, який може бути або декадним ( специфікація DEC), або октавним ( специфікація OCT). Атрибут <sweep variable name> – ім'я змінної, що змінюється у вказаних діапазонах. Атрибути <start value> та <end value> визначають, як і у попередньому випадку початкові та кінцеві значення змінної. Атрибут <points value> є цілочисленним та визначає кількість точок відповідно на октаву, або на декаду. Атрибут шаблону [nested sweep specification] – подібно до попередньо описаного. Наведемо приклад завдання на проведення багатоваріантного аналізу ланцюга за постійним струмом DC в логарифмічному масштабі:

```
.DC DEC NPN QK312(BF) 20 150 15
```

В цьому прикладі варіюється коефіцієнт передачі струму біполярного транзистора (параметр BF його моделі у SPICE).

Характер варіації змінної можна також вказувати заданим переліком значень. Тоді шаблон багатоваріантного аналізу ланцюга за постійним струмом DC матиме такий вигляд:

```
.DC <sweep variable name> LIST <value>  
+[nested sweep specification]
```

Після атрибуту LIST вказують через пробіл потрібні значення переліку. Приклад, який наведено нижче демонструє завдання аналізу ланцюга за постійним струмом при зміні температури (атрибут TEMP) яку задано переліком значень при варіації глобального параметра (атрибут PARAM) джерела напруги :

```
.DC TEMP LIST 0 20 27 50 80 100 PARAM VPower 7.5 15 .5
```

Звісно, якщо інженер має не лише ядро програми SPICE, а повноцінний програмний пакет АСхП, то йому необхідно лише вказати потрібні значення параметрів у полях відповідного екранного інтерфейсу. Такий інтерфейс для багатоваріантного аналізу ланцюга за постійним струмом DC програми DesignLab наприклад має простий та інтуїтивно зрозумілий інтерфейс.

Рисунок 1.30 – Екранний інтерфейс багатоваріантного аналізу ланцюга за постійним струмом DC програми DesignLab

Рисунок 1.31 – Екранний інтерфейс багатоваріантного аналізу ланцюга за постійним струмом DC програми Multisim

Екранний інтерфейс програми Multisim наприклад 12 версії має більш складну структуру, оскільки версія XSPICE яку використовує ця програма дозволяє проводити більш детальні дослідження. Як можна бачити з рис.1.31 вікно DC Sweep аналізу містить чотири вкладники, в яких інженер може вказувати потрібні параметри аналізу, визначати вихідні змінні, виконувати налаштування глобальних змінних програми та проглядати статистичну інформацію за обраною схемою.

Результати аналізу наводять у графічних модулях відображення результатів програм АСхП, а також у вихідний текстовий файл формату ASCII у вигляді текстових таблиць з числовими показниками результатів моделювання.

**Багатоваріантний аналіз лінеаризованого ланцюга в частотній області.** Аналіз дозволяє отримати частотні характеристики ланцюга або схеми, що моделюється. Змінні струми зумовлюють змінну різницю потенціалів (змінну напругу на елементах електричного кола). Змінна складова напруги або струму додається до постійної складової. В результаті відповідна характеристика може бути уніполярною або біполярною, але вона завжди асиметрична. Змінний струм є двонаправленим. Він спочатку протікає в одному напрямку, а потім - у протилежному. Змінний струм обумовлений змінною ЕРС.

При розрахунку частотних характеристик (малосигнальний аналіз) у якості мат.моделей використовується система лінійних алгебраїчних рівнянь з величинами елементів матриці  $A$ , які отримано при моделюванні за постійним струмом. Особливості

моделювання схеми у частотній області складаються у виділенні безреактивної (див. п. 2.3.) та реактивної частин схеми, які використовуються для формування дійсної та мнімої схемних матриць. Основна відмінність рішення система лінійних алгебраїчних рівнянь у частотній області полягає у виконанні дій над комплексними числами.

Розрахунок амплітудно-частотних та фазово-частотних характеристик здійснюється двома методами: методом прямого розрахунку передавальної функції у заданих точках частотного діапазону та розрахунком передавальної функції за допомогою відношення двох поліномів, коефіцієнти яких отримано з використанням дискретного перетворення Фур'є у комплексній області.

Перший метод полягає у наступному. Передавальна функція схеми  $K_{i,l}$ , яка описує передачу від  $i$ -го входу до  $l$ -го виходу, визначається відношенням вихідної характеристики  $y_l$  (струму чи напруги) до обумовленого нею вхідного впливу (джерело струму чи напруги):

$$K_{i,l} = y_l / x_i.$$

При дії синусоїдальних джерел струму та представлення компонентів схеми комплексними числами, напруги та струми можуть бути записані у такому вигляді:

$$U_k = |U_k| \cdot e^{j\varphi}; \quad I_k = |I_k| \cdot e^{j\varphi}$$

При умові, що на вході схеми підключено джерело синусоїдальної одиничної амплітуди та нульової фази  $x_i(p) = 1 \cdot e^{j0}$  чи  $x_i(t) = \sin(\omega \cdot t)$ , одержимо значення дійсної та мнімої складової передавальної функції  $K_{i,l}$  на частоті  $\omega$ , як дійсної (Re) та мнімої (Im) частини реакції схеми  $y_l$  на вплив  $i$ -го одиничного джерела:

$$K_{i,l}(j\omega) = \text{Re}(y_l) + j \text{Im}(y_l).$$

Звідси визначаються значення амплітудно-частотних (АЧХ), фазочастотних (ФЧХ) та логарифмічних амплітудно-частотних (ЛАЧХ) характеристик схеми:

$$K(\omega) = |K(j\omega)| = \sqrt{(\text{Re}(y_l))^2 + (\text{Im}(y_l))^2},$$

$$\varphi(\omega) = \operatorname{arctg} \left( \frac{\operatorname{Im}(y_l)}{\operatorname{Re}(y_l)} \right),$$

$$K_{dB}(\omega) = 20 \lg(K(\omega)),$$

Значення  $y_i$  визначаються за допомогою рішення системи лінійних алгебраїчних рівнянь у комплексній площині:

$$(A_r + j \cdot A_i) \cdot X = B,$$

де  $A_r$  і  $A_i$  – дійсна та мніма частини схемних матриць відповідно;  $B$  – вектор правих частин, значення елементів якого залежить від типу джерела вхідного сигналу та вузлів його підключення.

Другий метод полягає у наступному. Передавальна функція схеми може бути представлена у вигляді відношення поліному чисельника до поліному знаменника із дійсними коефіцієнтами щодо ступенів комплексної змінної  $Z$  (комплексна частота):

$$K_{i,l}(z) = \frac{\sum_{k=0}^n A_k \cdot Z^k}{\sum_{k=0}^m B_k \cdot Z^k} \quad (1.6.1)$$

Якщо значення комплексної змінної розташувати еквідистантно на побудованому у комплексній площині колі одиничного радіусу, тобто

$$Z = \cos\left(\frac{\pi \cdot b}{N}\right) + j \sin\left(\frac{\pi \cdot b}{N}\right); \quad b=0,1,2,\dots,2N-1,$$

то поліноми чисельника та знаменника (1.6.1) відповідають стандартній формі дискретного перетворення Фур'є у комплексній області:

$$Q_1(Z_b) = \sum_{b=0}^n A \left[ \cos\left(k \frac{\pi \cdot b}{N}\right) + j \sin\left(k \frac{\pi \cdot b}{N}\right) \right];$$

$$Q_2(Z_b) = \sum_{b=0}^m B \left[ \cos\left(k \frac{\pi \cdot b}{N}\right) + j \sin\left(k \frac{\pi \cdot b}{N}\right) \right];$$

Якщо  $n$  і  $m$  менше ніж  $2N$ , то коефіцієнти чисельника та знаменника можуть бути обчислено із співвідношень:



$$A_k = \frac{1}{2N} \sum_{b=0}^{2N-1} Q_1(Z_b) \cdot Z_b^k; \quad B_k = \frac{1}{2N} \sum_{b=0}^{2N-1} Q_2(Z_b) \cdot Z_b^{-k};$$

Значення знаменника  $Q_2(Z_b)$  у заданих точках на одиничному колі може бути обчислено як добуток діагональних членів матриці  $L$ , при  $LU$ -перетворенні. Потім обчислюється значення  $Q_{i,l}(Z_b)$  – рішення системи  $LU_X=B$ . При відомому  $Q_{i,l}(Z_b)$  та  $Q_2(Z_b)$ , знаходиться  $Q_1(Z_b)=K_{i,l}(Z_b) \cdot Q_2(Z_b)$ .

У режимі аналізу частотних характеристик визначають початкові умови за допомогою статичного розрахунку, після цього всі нелінійні компоненти лінеаризуються, та здійснюється розрахунок комплексних амплітуд вузлових потенціалів та струмів особливих ланцюгів. Амплітуда вхідного синусоїдального сигналу дорівнює одиниці. Мат.моделі отримано у вигляді системи алгебраїчних рівнянь.

Тепер розглянемо шаблон завдання багатоваріантного аналізу ланцюга в частотній області AC Sweep у форматі SPICE.

```
.AC <sweep type> <points value>
+ <start frequency value> <end frequency value>
```

Атрибут шаблону <sweep type> має подібне значення до аналогічного атрибуту в опису аналізу за постійним струмом. Атрибут <points value> при вказанні лінійного характеру зміни визначає загальну кількість точок по частоті. При логарифмічному характерові зміни він означає кількість точок на декаду або октаву, в залежності який саме тип логарифмічної зміни обрано. Атрибут <start frequency value> визначає значення початкової частоти, атрибут <end frequency value> визначає значення кінцевої частоти. Таким чином аналіз виконується у діапазоні цих двох частот – від початкової до кінцевої.

Наведемо приклади завдання директиви виконання багатоваріантного аналізу ланцюга у частотній області у форматі SPICE.

```
.AC LIN 101 100Hz 200Hz
.AC OCT 10 1kHz 16kHz
.AC DEC 20 1MEG 100MEG
```

Наведені приклади не потребують роз'яснень.

Екранний інтерфейс для багатоваріантного аналізу ланцюга у частотній області АС програми DesignLab має вигляд як це показано на рис. 1.32. Як бачимо в нижній частині робочого поля інтерфейсу встановлено елементи для настроювання аналізу шуму. Про цей вид аналізу більш детально буде розглянуто далі. Екранний інтерфейс АС SWEEP аналізу програми Multisim показано на рис. 1.33.

Рисунок 1.32 – Екранний інтерфейс багатоваріантного аналізу ланцюга у частотній області АС програми DesignLab

Рисунок 1.33 – Екранний інтерфейс багатоваріантного аналізу ланцюга у частотній області АС програми Multisim

**Аналіз спектральної щільності внутрішнього шуму Noise Analysis.** Цей вид аналізу вважають додатковим, оскільки він потребує попереднього проведення аналізу АС Sweep, оскільки в частотному аналізі розробник задає діапазон частот. Джерелами шуму є резистори, перемикачі та напівпровідникові пристрої в яких є шумові моделі (див., наприклад шумову модель діоду у п.1.2). На кожній частоті розраховується спектральна щільність вихідної напруги  $S_{u \text{ вих.}}(f)$ , яка обумовлена наявністю статично незалежних джерел внутрішнього шуму. Розглянемо специфікацію завдання аналізу шуму у форматі SPICE.

**.NOISE V(<node> [,<node>]) <name> [interval value]**

Атрибут специфікації V(<node> [,<node>]) визначає вузли контрольованої вихідної напруги. Атрибут <name> визначає ім'я незалежного джерела напруги або струму, який підключається автоматичного до вхідних вузлів ланцюга, що аналізується. Таке джерело не є джерелом реального сигналу, воно потрібно лише для позначення тих вузлів, які будуть вважатися вхідними для ланцюга, що аналізують, і по відношенню до яких розраховують вихідний шум. Якщо до входу підключають джерело напруги, то на вході розраховують еквівалентну спектральну щільність напруги  $S_{u \text{ екв.}}(f)$ , В<sup>2</sup>/Гц, якщо ж до входу підключають джерело струму, то розраховують еквівалентну спектральну щільність струму  $S_{i \text{ екв.}}(f)$ ,

$A^2/Гц$ . Рівень шуму перераховують с виходу на вхід поділенням спектральної щільності вихідної напруги  $S_{u_{вих}}$  на квадрат модуля відповідної передаточної функції. Слід взяти до уваги, що внутрішній опір реального генератора сигналу ( $R_u$  або  $R_r$ ) потрібно обов'язково увести в блок опису схеми, що аналізують, як окремий опір цієї схеми.

Атрибут [interval value] відповідає значенню атрибута <points value> у опису попереднього аналізу АС. У даному випадку він визначає кількість точок, які будуть записані у вихідний файл із результатами моделювання. Наведемо приклади завдання на виконання аналізу спектральної щільності внутрішнього шуму.

```
.NOISE V(5) VIN
.NOISE V(101) VSRC 20
.NOISE V(4,5) ISRC
.NOISE V([OUT1],[OUT2]) V1
```

Другий приклад як раз є прикладом зазначення кількості характеристик, які будуть записані у вихідний файл.

При запису результатів аналізу шуму у вихідний файл використовують зарезервовані ідентифікатори змінних. Для вхідного шуму використовують змінні: INOISE – змінна у відносних одиницях; DB (INOISE) – у децибелах. Для вихідного шуму використовують змінні: ONOISE – змінна у відносних одиницях; DB (ONOISE) – у децибелах. Для обох типів шумів визначають СКВ, тобто  $\sqrt{S_{ex}(f)}$  – для вхідного шуму, та  $\sqrt{S_{u_{вих}}(f)}$  – для вихідного.

За результатами розрахунку спектральної щільності внутрішнього шуму можна обрахувати диференційний коефіцієнт шуму лінійного чотирьохполюсника. Відомо, що диференційний коефіцієнт шуму

$$K_u(f) = \frac{S_{u_{вих.екв}}(f)}{S_{uR}}$$

де  $S_{u_{вих.екв}}(f)$  – спектральна щільність напруги, обумовленого шумом опору генератора  $R_r$  та внутрішнім шумом чотирьохполюсника (рис.1.34), перерахована до його входу,  $S_{u_{вих.екв}}(f) = \text{INOISE}^2$ ;  $S_{uR} = 4kT_0R_r$  – спектральна щільність напруги

шуму опору генератора;  $k=1,38 \cdot 10^{-23}$  Дж/°С – стала Больцмана;  $T_0=300$  К – абсолютна температура.

Рисунок 1.34 – Чотирьохполосник при підключенні до входу джерела напруги або струму

Оскільки використані константи у виразі для обчислення шуму, то можна записати вираз для розрахунку диференційного коефіцієнту шуму

$$K(f) = \frac{\text{INOISE}^2}{1,656 \cdot 10^{-20} \cdot R_r}$$

де опір  $R_r$  – вказують в омах.

Наведемо приклад завдання на розрахунок диференційного коефіцієнту шуму чотирьохполосника з рис. 1.34а.

```
.TEMP      80
VG 1      0      AC      1
RG 1      2      75
RLOAD     3      4      500
CLOAD     3      4      30pF
.AC LIN   21     0      1000HZ
.NOISE    V(3,4) VG
.PRINT    INOISE      ONOISE
```

Схожим образом обраховують диференційний коефіцієнт шуму при підключенні на вхід чотирьохполосника джерела струму (рис 1.34б):

$$K(f) = \frac{\text{INOISE}^2}{1,656 \cdot 10^{-20} / R_r}$$

Розрахунок СКВ вихідної напруги шуму виконують за допомогою виразу

$$\sigma_{u_{вих}} = \sqrt{\int S_{u_{вих}}(f) df}$$

В професійних програмах АСхП у складі набору вбудованих функцій обов'язково є функція інтегрування у складі функцій опрацювання результатів – S(x). Тоді обчислення СКВ у завданні на

моделювання у форматі SPICE виглядатиме так:

$$\text{SQRT}(S(V(\text{ONoise})*V(\text{ONoise})))$$

СКВ шуму  $\sigma_{u, \text{вих}}$  дорівнює значенню цієї функції на верхній границі діапазону частот.

При розрахунку коефіцієнта шуму за допомогою атрибута T\_ABS необхідно вказати такі значення температури для елементів, які приймають участь у розрахунках: опору генератора  $R_T$  необхідно приписати номінальну температуру  $T_0$ , чотирьохполоснику – його фізичну температуру, а опору навантаження – температуру абсолютного нуля, оскільки його шуми зазвичай беруть до уваги при розрахунку коефіцієнта шуму наступного каскаду.

Для вирішення певних задач необхідна наявність незалежних джерел шуму. Їх можна представити в якості залежних джерел, керованих струмом резистора, що шумить

Рисунок 1.35 – Незалежне джерело напруги та струму широкополосного шуму

На рис.1.35а наведено модель незалежного джерела напруги шуму. Опір резистора, що шумить, цього джерела пов'язаний із спектральною щільністю напруги шуму  $S_u$  співвідношенням  $RN = S_u / (4kT)$ . Наприклад при  $S_u = 10^{-18} \text{ В}^2/\text{Гц}$  опис моделі на рис.1.35а має вигляд

RN 1	0	60.4	
VN 1	0	DC	0
HN 2	3	VN	1

На рис. 1.35б наведена модель незалежного джерела шумового струму. Для неї опір резистора, що шумить, пов'язаний із спектральною щільністю шумового струму  $S_i$  співвідношенням  $RN = 4kT / S_i$ . Наприклад при  $S_i = 10^{-24} \text{ А}^2/\text{Гц}$  опис моделі на рис.1.35б має вигляд

RN 1	0	16.56K		
VN 1	0	DC	0	
FN	2	3	VN	1

В програмі Multisim можна налаштувати частотні параметри аналізу спектральної щільності шуму та аналізу ланцюга у частотній області окремо. Як можна бачити з рис. 1.36 екранний інтерфейс цього аналізу містить вкладнику Frequency Parameters яка дозволяє вводити початкову та кінцеву частоти виключно для шумового аналізу.

Рисунок 1.36 – Екранний інтерфейс аналізу спектральної щільності внутрішнього шуму програми Multisim

## РОЗДІЛ 2

Автоматизоване проектування пристроїв з цифровими компонентами та компонентами змішаної функціональності

### 2.1. Теоретичне підґрунтя сучасної цифрової схемотехніки

**АЛГЕБРА ЛОГІКИ Є МАТЕМАТИЧНОЮ ОСНОВОЮ ЦИФРОВОЇ СХЕМОТЕХНІКИ. АЛГЕБРА ЛОГІКИ ДОЗВОЛЯЄ ЗАПИСУВАТИ СКЛАДНІ ЛОГІЧНІ ВИСЛОВЛЮВАННЯ В БІЛЬШІ ПРОСТІШІЙ ФОРМІ, ОСКІЛЬКИ ПРАВИЛА ДЛЯ ОБРОБКИ ТА ЗНАХОДЖЕННЯ МІНІМАЛЬНОЇ ФОРМИ ЛОГІЧНИХ ВИСЛОВЛЮВАНЬ. ІСНУЄ РЯД ТЕОРЕМ ТА ТОТОЖНОСТЕЙ АЛГЕБРИ ЛОГІКИ. ВКАЗАНІ НИЖЧЕ ОДИНАДЦЯТЬ ПРАВИЛ Є ОЧЕВИДНИМИ. ЛОГІЧНІ ВИСЛОВЛЮВАННЯ ПРЕДСТАВЛЕНІ У ВИГЛЯДІ ЛОГІЧНИХ ЗМІННИХ ЯКИ МОЖУТЬ ПРИЙМАТИ ТІЛЬКИ ДВА СТАНИ – ЛОГІЧНОГО НУЛЯ ТА ЛОГІЧНОЇ ОДИНИЦІ. ЛОГІЧНІ ЗМІННІ БУДЕМО ЗАПИСУВАТИ ВЕЛИКИМИ ЛІТЕРАМИ ЛАТИНСЬКОГО АЛФАВІТУ. ЛОГІЧНІ ОПЕРАЦІЇ ТА, АБО БУДЕМО ЗАПИСУВАТИ СИМВОЛАМИ “·” ТА “+” ВІДПОВІДНО. ОПЕРАЦІЮ НІ – СИМВОЛОМ ІНВЕРСІЇ НАД ЛОГІЧНОЮ ЗМІННОЮ. ТОДІ ПЕРШІ ОДИНАДЦЯТЬ ПРАВИЛ АЛГЕБРИ ЛОГІКИ МАТИМУТЬ ТАКИЙ ВИГЛЯД:**

1.  $A \cdot A = A$ ;
2.  $A + A = A$ ;
3.  $A \cdot 1 = A$ ;
4.  $A \cdot 0 = 0$ ;
5.  $A + 1 = 1$ ;
6.  $A + 0 = A$ ;
7.  $\overline{\overline{A}} = A$ ;
8.  $A \cdot \overline{A} = 0$ ;

9.  $A + \bar{A} = 1;$
10.  $A + B = B + A;$
11.  $A \cdot B = B \cdot A.$

Наступні два правила алгебри логіки носять назву “закони в дужках”:

12.  $(A + B) + C = A + (B + C) = A + B + C;$
13.  $(A \cdot B) \cdot C = A \cdot (B \cdot C) = A \cdot B \cdot C.$

Наступні два правила алгебри логіки носять назву “закони поглинання”:

14.  $A + A \cdot B = A;$
15.  $A \cdot (A + B) = A.$

Наступні два правила називають “закони Моргана” або “закони розподілу”:

16.  $A + B \cdot C = (A + B) \cdot (A + C);$
17.  $A \cdot (B + C) = A \cdot B + A \cdot C.$

Вісімнадцате правило називають “закон антikon'юнкції” або “штрихи Шеффера”:

18.  $\overline{A \cdot B} = \bar{A} + \bar{B}.$

Дев'ятнадцате правило називають “закон антідиз'юнкції” або “стрілки Пірса”:

19.  $\overline{A + B} = \bar{A} \cdot \bar{B}.$

Якщо перші одинадцять правил є очевидними, то наступні потребують доказів. Ці докази можна виконувати також і експериментальним шляхом.

Базові логічні компоненти мають таке зображення у програмах автоматизації схеми технічного проектування:

- інвертор (НІ) –
- кон'юнктор (ТА) –
- диз'юнктор (АБО) –



Також дуже розповсюдженим є логічний елемент “виключне АБО” який зображують за допомогою такого символу

Розглянемо експериментальний шлях доведення законів алгебри логіки на прикладі доказу одного з законів Моргана (правила 17).

Для проведення доказу експериментальним шляхом потрібно визначити по окремо таблиці істинності лівої та правої частини досліджуваної тотожності. Для цього в програмі складемо моделі логічних схем цих частин які можуть виглядати наприклад наступним чином. Ліву частину правила 17 можна представити наступною схемою (рис.2.1).

Рисунок 2.1– Логічна схема виразу  $A \cdot (B + C)$

Потім намалюємо таблицю станів для трьох вхідних логічних змінних А,В,С та одного виходу Y (див. табл.1).

Таблиця 2.1

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

У трьох лівих стовпцях перелічимо усі можливі комбінації вхідного дискретного сигналу. Далі увімкнемо схему для моделювання та шляхом почергового перемикавання перемикачів А,В,С забезпечимо записані у таблиці комбінації вхідних сигналів, та фіксуємо після кожного нового стану вхідних сигналів стан вихідного індикатора, заповнюємо останній стовпчик таблиці істинності який показує стан виходу схеми Y. Після проведення експерименту отримаємо такі результати (табл.1).

Тепер складемо модель логічної схеми правої частини правила 17. Вона матиме наступний вигляд (рис.2.2).

Рисунок 2.2 – Логічна схема виразу  $A \cdot B + A \cdot C$

*Таблиця 2.2*

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Потім накреслимо таблицю станів для трьох вхідних логічних змінних A, B, C та одного виходу Y. У трьох лівих стовпцях перелічимо усі можливі комбінації вхідного дискретного сигналу. Далі увімкнемо схему для моделювання та шляхом почергового перемикавання перемикачів A, B, C забезпечимо записані у таблиці комбінації вхідних сигналів, та фіксуємо після кожного нового стану вхідних сигналів стан вихідного індикатора, заповнюємо останній стовпчик таблиці істинності який показує стан виходу схеми Y. Після проведення експерименту отримаємо такі результати (табл.2.2).

Як бачимо, результати в обох таблицях є ідентичним, тобто 17 правил є доведеним.

Базовими логічними блоками цифрової схемотехніки є три елемента, які реалізують три базові операції алгебри логіки, тобто : НІ (NOT), ТА (AND), АБО (OR). В мікросхематичному виконанні можуть зустрічатися більш складніші варіанти логічних операцій, та вони все рівно є похідними від трьох базових операцій.

Наступне питання яке необхідно розглянути – це мінімізація логічних висловлювань. Це питання є дуже актуальним, тому що мінімізація дозволяє скоротити початковий логічний вираз, що призводить до зменшення кількості логічних компонентів у цифровій мікросхемі та кількості зв'язків між цими компонентами. Мінімізацію можна виконати за допомогою різних інструментів. По-перше мінімізацію можна виконати суто розрахунковим шляхом, застосовуючи закони алгебри логіки провести мінімізацію початкового виразу.

По-друге можна використати спеціально розроблені табличні способи. Але перед розглядом цих способів потрібно розглянути

представлення логічних висловлювань у диз'юнктивній нормальній формі (ДНФ) та кон'юнктивній нормальній формі.

Існують дві стандартні форми представлення логічних висловлювань. Перша форма – це диз'юнктивна нормальна форма (ДНФ). Це така форма, при якій логічне висловлювання записано у вигляді суми добутоків простих висловлювань (або їх заперечень). Прикладом ДНФ служить вираз.

$$A \cdot B + C \cdot \overline{D} + \overline{A} \cdot \overline{F}.$$

Друга форма – це кон'юнктивна нормальна форма (КНФ). Це форма представлення логічних виразів, при якій цей вираз записують у вигляді добутоків суми простих висловлювань (або їх заперечень). Прикладом КНФ служить вираз:

$$(A + B) \cdot (\overline{C} + \overline{D})$$

**РОЗРОБЛЕНО КІЛЬКА СПОСОБІВ МІНІМІЗАЦІЇ СКЛАДНИХ ЛОГІЧНИХ ВИСЛОВЛЮВАНЬ. ОКРЕМІ СПОСОБИ ДЛЯ ДОСЯГНЕННЯ МЕТИ МІНІМІЗАЦІЇ ПЕРЕДБАЧАЮТЬ ПОБУДОВУ СПЕЦІАЛЬНИХ ТАБЛИЦЬ АБО КАРТ, ТАКИХ, ЯК, НАПРИКЛАД КАРТИ КАРНО, КАРТИ ВЕЙЧА, ТАБЛИЦІ ВЕНА. ОДНИМ З ДОСТАТНЬО ПОШИРЕНИХ СПОСОБІВ МІНІМІЗАЦІЇ Є СПОСІБ, ЯКИЙ ПЕРЕДБАЧАЄ ПОБУДОВУ КАРТИ КАРНО. РОЗГЛЯНЕМО СУТНІСТЬ ЦЬОГО СПОСОБУ. КАРТИ КАРНО Є СИСТЕМОЮ ГРАФІЧНОГО ПРЕДСТАВЛЕННЯ ТА СПРОЩЕННЯ ЛОГІЧНИХ ВИСЛОВЛЮВАНЬ. КАРТА КАРНО УТВОРЕНА ІЗ ПАРНОЇ КІЛЬКОСТІ КВАДРАТІВ. КІЛЬКІСТЬ КВАДРАТІВ У КАРТІ ВІДПОВІДАЄ КІЛЬКОСТІ МОЖЛИВИХ КОМБІНАЦІЙ ЛОГІЧНИХ ЗМІННИХ У ВИСЛОВЛЮВАННІ. КІЛЬКІСТЬ КВАДРАТІВ  $n$  МОЖНА ПІДРАХУВАТИ ЗА ФОРМУЛОЮ  $n = 2^x$ ,  $x$  – КІЛЬКІСТЬ ЗМІННИХ У ВИСЛОВЛЮВАННІ. ЯКЩО МАЄМО ДВІ ЗМІННІ ( $A$  ТА  $B$ ), ТО В КАРТІ БУДЕ ЧОТИРИ КВАДРАТИ. ПРИ ТАКОМУ ЗОБРАЖЕННІ КВАДРАТ 1 НА КАРТІ КАРНО**

ВІДПОВІДАЄ ЛОГІЧНОМУ ДОБУТКУ  $\bar{A} \cdot \bar{B}$ , КВАДРАТ 2 – ЛОГІЧНОМУ ДОБУТКУ  $\bar{A} \cdot B$ , КВАДРАТ 3 – ЛОГІЧНОМУ ДОБУТКУ  $A \cdot B$ , КВАДРАТ 4 – ЛОГІЧНОМУ ДОБУТКУ  $A \cdot \bar{B}$ .

ПОТРІБНО ПАМ'ЯТАТИ, ЩО ПЕРЕХОДІ ВІД ОДНОГО КВАДРАТА ДО СУСІДНЬОГО ЗМІНЮВАТИСЯ ПОВИННА ЛИШЕ ОДНА ЗМІННА, А НЕ БІЛЬШЕ. У ВИПАДКУ ДВОХ ЗМІННИХ ЗМІНЮВАТИСЯ ПРИ ПЕРЕХОДІ ДО СУСІДНЬОГО КВАДРАТУ ПОВИННА АБО ЗМІННА А, АБО ЗМІННА В, АЛЕ НЕ ОБИДВІ РАЗОМ.

У КВАДРАТИ ЗАПИСУЮТЬ РЕЗУЛЬТАТИ ЛОГІЧНОЇ ОПЕРАЦІЇ З ЗМІННИМИ В КОЖНОМУ РЯДКУ. ЗАЗВИЧАЙ ЗАПИСУЮТЬ ТІЛЬКИ ТОЙ РЕЗУЛЬТАТ, ЯКИЙ ДАЄ ЛОГІЧНУ ОДИНИЦЮ. ЗАПОВНЕНА КАРТА БУДЕ ГОТОВОЮ ДЛЯ ПОБУДОВИ. НАВЕДЕМО МЕТОДИКУ СКЛАДАННЯ КАРТИ КАРНО. МЕТОДИКА МІСТИТЬ ШІСТЬ КРОКІВ:

1 КРОК. ЛОГІЧНЕ ВИСЛОВЛЮВАННЯ, ЩО ПІДЛЯГАЄ МІНІМІЗАЦІЇ ТРАНСФОРМУЮТЬ У ДИЗЬЮНКТИВНУ НОРМАЛЬНУ ФОРМУ (ДНФ). ДНФ – ЦЕ СУМА ДОБУТКІВ ПРОСТИХ ВИСЛОВЛЮВАНЬ АБО ЇХ ЗАПЕРЕЧЕНЬ. НАПРИКЛАД, Є ТАКЕ ЛОГІЧНЕ ВИСЛОВЛЮВАННЯ, ЯКЕ НЕ ПРЕДСТАВЛЕНЕ У ДНФ:  $Y = \overline{\bar{A} \cdot B \cdot C}$ . ВИКОНАЄМО ТРАНСФОРМУВАННЯ ЦЬОГО ВИСЛОВЛЮВАННЯ В ДНФ. ТОДІ :

$$Y = \overline{\bar{A} \cdot B \cdot C} = (A + \bar{B}) \cdot C = A \cdot C + \bar{B} \cdot C.$$

ОСТАННЄ ВИРАЖЕННЯ Є ДИЗЬЮНКТИВНОЮ НОРМАЛЬНОЮ ФОРМОЮ ЛОГІЧНОГО ВИСЛОВЛЮВАННЯ.

2 КРОК. СКЛАДАЄМО КАРТУ КАРНО. ЛОГІЧНІ ЗМІННІ МАЮТЬ БУТИ РОЗТАШОВАНІ В САМОМУ ЛІВОМУ СТОВПЧИКУ ТА У САМОМУ ВЕРХНЬОМУ РЯДКУ КАРТИ ТАКИМ ЧИНОМ, ЩОБ ПЕРЕХІД ВІД КОЖНОГО КВАДРАТА КАРТИ ДО СУСІДНЬОГО, ЗМІНЮВАВ БИ СТАН ТІЛЬКИ

**ОДНОГО ВХОДУ (БІТА). У ВІДПОВІДНІ КВАДРАТИ  
ЗАНОСЯТЬ ОДИНИЦІ.**

**3 КРОК. СУСІДНІ ОДИНИЦІ ПОЄДНУЮТЬ ОВАЛЬНИМИ  
КОНТУРАМИ, ЯКІ ОХОПЛЮЮТЬ 2, 4 АБО 8 КВАДРАТІВ.  
ФОРМУВАННЯ КОНТУРІВ МОЖЛИВЕ ПО ГОРИЗОНТАЛІ  
АБО ВЕРТИКАЛІ, АЛЕ НЕ ПО ДІАГОНАЛІ. ПОБУДОВУ  
КОНТУРІВ ПРОДОВЖУЮТЬ ДО ТИХ ПІР, ДОКИ ВСІ  
ОДИНИЦІ НЕ ОПИНЯТЬСЯ ВСЕРЕДИНІ КОНТУРУ.**

**4 крок.** Виконують спрощення, виключаючи члени вираження, які доповнюють одне одного всередині контуру.

**5 крок.** Члени, що залишилися після спрощення (по одному в кожному вираженні) поєднують логічною функцією АБО.

**6 крок.** Записують отримане спрощене логічне вираження в ДНФ (у вигляді суми добутоків).

Складемо карту Карно для виразу  $Y = \bar{A} \cdot B + A \cdot \bar{B} + A \cdot B$ . Логічні одиниці розміщують у всіх квадратах, яким відповідають добутки у початковому бульовому висловлюванні (рис.2.3).

Рисунок 2.3 – Складання карти Карно

Наразі заповнену таким чином карту Карно можна будувати. Цей процес продемонстровано на рис. 2.4.

Рисунок 2.4 – Побудова карти Карно

У відповідності до методики одиниці у сусідніх квадратах по горизонталі або по вертикалі об'єднують в один контур групами по дві, чотири або вісім одиниць. побудову контурів виконують до тих пір, доки всі одиниці, по можливості не попадуть в середину контуру. Кожен контур уявляє з себе новий член спрощеного бульового виразу. Як бачимо, на рис.2.4 ми отримали тільки два контури. Це означає, що нове, спрощене висловлювання буде складатися лише з двох членів, які будуть пов'язані логічним АБО. Розглянемо горизонтальний контур. Змінна А поєднана в ньому із комбінацією В і  $\bar{B}$  (рис.2.5).

Рисунок 2.5 – Запис мінімізованого висловлювання

У відповідності із правилом 8,  $B$  і  $\bar{B}$  доповнюють одне одного (тобто їх добуток дорівнює нулю) і їх можна скоротити. Так, у горизонтальному контурі залишилася лише змінна  $A$ . Аналогічно цьому у вертикальному контурі скорочується комбінація  $A$  і залишається лише  $B$ . В результаті залишаються лише  $A$  й  $B$ , котрі поєднують функцією АБО.

Аналогічно розглянемо більш складний вираз – який складено із чотирьох змінних :

Складемо карту Карно й занесемо туди одиниці (рис.2.6).

Рисунок 2.6 – Побудова карти Карно для чотирьох змінних

Створимо контура та виконаємо спрощення. (рис.2.7).

Рисунок 2.7 – Побудова контурів

Запишемо мінімізований вираз (рис. 2.8)

Рисунок 2.8– Отримання мінімізованого виразу

Звісно, що обидва способи передбачають виконання розробником певних механічних дій. Тому створювачі сучасних програм АСхП надали інженерам можливість для автоматизації операцій, пов'язаних із мінімізацією логічних висловлюван.

У сучасних програмах та пакетах класу EDA є вбудовані віртуальні інструменти для роботи із логічними схемами. В пакеті Multisim, у наборі Instrument є елемент, який носить назву логічного

конвертора (Logic Converter). Він має наступний вигляд (рис. 2.9). На рис. 2.10 наведено робоче вікно для реалізації логічних операцій.

Рисунок 2.9 – Елемент Logic Converter

Рисунок 2.10 – Екранний інтерфейс елемента Logic Converter

Цей логічний конвертор дозволяє реалізувати такі операції: логічний аналіз входового пристрою із одним виходом; синтез логічного пристрою за заданою таблицею істинності; синтез логічного пристрою за заданим логічним висловлюванням. У верхній частині конвертора розташовані вісім вхідних віртуальних клем (А-Н). Та у правому верхньому куті розташована клемка, до якої під'єднують вихід логічного пристрою – Out. Якщо потрібно отримати таблицю істинності логічної схеми, то таку схему попередньо складають на робочому полі, потім входи цієї схеми під'єднують до клем (А-Н), вихід схеми під'єднують до клемки Out та натискають у секції перетворювань (Conversions) вікна конвертора віртуальну клавішу. Після цього таблиця істинності з'являється на екрані конвертора.

Якщо потрібно тепер отримати логічне висловлювання, яке реалізує досліджувану логічну схему потрібно лише натиснути клавішу і це логічне висловлювання з'явиться у нижньому рядку екрана конвертора.

Якщо тепер потрібно мінімізувати логічне висловлювання то необхідно натиснути наступну клавішу, та попередній логічний вираз у нижньому рядку екрана буде замінено на його мінімізовану форму. Не є обов'язковим складання логічної схеми для того, щоб отримати мінімізовану форму її логічного висловлювання. Можна просто ввести логічне висловлювання, яке ми хочемо мінімізувати у нижньому рядку екрана конвертора та потім натиснути віртуальну клавішу мінімізації. Під час запису логічного виразу у нижньому рядку екрана треба пам'ятати, що змінні записувати потрібно тільки латинськими літерами від А до Н, операцію інверсії позначають апострофом ('), кон'юнкцію не позначають (наприклад логічний добуток змінних В та С слід записати так: ВС), диз'юнкцію

позначають знаком +. Якщо необхідно, можна застосовувати круглі дужки для позначення пріоритету операцій.

Якщо потрібно отримати логічну схему на основі даного логічного висловлювання, то таке висловлювання потрібно спочатку записати у нижньому рядку екрана конвертора за вищенаведеними правилами. Далі можливо отримати два варіанти логічної схеми. Якщо ми хочемо отримати логічну схему складену з базових логічних блоків, то потрібно натиснути віртуальну клавішу. Якщо ми натиснемо клавішу, то ми отримаємо логічну схему складену на базі тільки логічних елементів ГА-НІ.

**ПРИ НАТИСКАННІ КЛАВІШІ МИ ОТРИМАЄМО НА ЕКРАНІ КОНВЕРТОРА ТАБЛИЦЮ ІСТИННОСТІ ЛОГІЧНОГО ВИСЛОВЛЮВАННЯ, ЯКЕ ЗАПИСАНЕ У НИЖНЬОМУ РЯДКУ.**

## **2.2. Схемотехнічна реалізація логічних функцій. Внутрішня структура цифрових мікросхем**

**ВНУТРІШНЯ СТРУКТУРА ЦИФРОВИХ МІКРОСХЕМ ЗАЛЕЖИТЬ ВІД ТИПУ ТРАНЗИСТОРНОЇ ТЕХНОЛОГІЇ, ЯКА ЗАСТОСОВУЄТЬСЯ ПРИ ПОБУДОВІ МІКРОСХЕМ. Є ДВІ БАЗОВІ ТЕХНОЛОГІЇ ДЛЯ ПОБУДОВИ ЦИФРОВИХ МІКРОСХЕМ – ЦЕ БІПОЛЯРНА ТА ПОЛЬОВА ТРАНЗИСТОРНІ ТЕХНОЛОГІЇ. В МЕЖАХ ТЕХНОЛОГІЇ ТАКОЖ Є ДОДАТКОВІ ВІДМІННОСТІ ЯКІ ПОЄДНУЮТЬ РІЗНІ СЕРІЇ У МЕЖАХ ОДНІЄЇ ТЕХНОЛОГІЇ.**

**ЦИФРОВІ МІКРОСХЕМИ НА ОСНОВІ ПОЛЬОВОЇ ТЕХНОЛОГІЇ, ЯКІ**



**ВИКОНУЮТЬ ЛОГІЧНІ ФУНКЦІЇ МІСТЯТЬ  
У СВОЇ СТРУКТУРІ ПАРУ  
КОМПЛІМЕНТАРНИХ МЕТАЛ-ОКСИД-  
НАПІВПРОВІДНИКОВИХ (КМОН)  
ТРАНЗИСТОРІВ ІЗ ВБУДОВАНИМ  
КАНАЛОМ. ЧЕРЕЗ ЦЕ ДАНИЙ ТИП  
ЦИФРОВОЇ ЛОГІКИ НАЗВАНО ЛОГІКОЮ  
КМОН. СПРОЩЕНА СТРУКТУРА  
ІНВЕРТОРА НА ОСНОВІ КМОН-ЛОГІКИ  
НАВЕДЕНА НА РИС. 2.11.**

Рисунок 2.11 – Інвертор КМОН

**У СХЕМІ ЗАСТОСОВАНО ПОСЛІДОВНЕ  
З'ЄДНАННЯ P- І N- КАНАЛЬНОГО МОН  
ТРАНЗИСТОРІВ VT1 ТА VT2. ПОСЛІДОВНЕ  
ВКЛЮЧЕННЯ МОН ТРАНЗИСТОРІВ  
ДОЗВОЛЯЄ ПІДВИЩИТИ ШВИДКОДІЮ  
СХЕМИ. ЗАТВОРИ ОБОХ ТРАНЗИСТОРІВ  
З'ЄДНАНО РАЗОМ І ПІДКЛЮЧЕНО ДО  
КЕРУЮЧОГО ВИВОДУ (ДВИЖКА)  
ПОТЕНЦІОМЕТРА R. СТОКИ ОБОХ  
ТРАНЗИСТОРІВ ТАКОЖ З'ЄДНАНІ ОДНЕ З  
ОДНИМ. З ТОЧКИ ПОЄДНАННЯ СТОКІВ  
БУДЕ ЗНІМАТИСЯ ВИХІДНИЙ ЛОГІЧНИЙ  
СИГНАЛ. ЯКЩО ДВИЖОК  
ПОТЕНЦІОМЕТРА ПЕРЕМІСТИТИ  
“ДОНИЗУ” (БЛИЖЧЕ ДО НУЛЯ), ТО НА**

ОБИДВА ЗАТВОРИ ПОТРАПИТЬ  
НУЛЬОВИЙ РІВЕНЬ НАПРУГИ, ЧЕРЕЗ ЩО  
ПОВНІСТЮ ВІДКРИТИМ БУДЕ ТІЛЬКИ  
КАНАЛ ПРОВІДНОСТІ Р-КАНАЛЬНОГО  
ТРАНЗИСТОРА VT1. ЧЕРЕЗ ВІДКРИТИЙ  
КАНАЛ ТРАНЗИСТОРА VT1 В ЙОГО СТОК  
ПОТРАПИТЬ ПОТЕНЦІАЛ ДЖЕРЕЛА  
ЖИВЛЕННЯ. N- КАНАЛ ТРАНЗИСТОРА VT2  
БУДЕ В ЦЕЙ ЧАС ПОВНІСТЮ ЗАКРИТИМ.  
ТОМУ ВЕЛИЧИНА ВИХІДНОЇ НАПРУГИ  
 $U_{вих}$  ПРИ НУЛЬОВОМУ ЗНАЧЕННІ  
НАПРУГИ НА ВХОДІ БУДЕ ПРАКТИЧНО  
ДОРІВНЮВАТИ ЗНАЧЕННЮ НАПРУГИ  
ДЖЕРЕЛА ЖИВЛЕННЯ. ТОМУ МОЖНА  
ЗАПИСАТИ  $U_{вих}^1 = U_{н.л.}$ .

ТЕПЕР, ЯКЩО ДВИЖОК  
ПОТЕНЦІОМЕТРА ПЕРЕМІСТИТИ У  
МАКСИМАЛЬНО “ВЕРХНЄ” ПОЛОЖЕННЯ  
(БЛИЖЧЕ ДО ПРОВОДУ ЖИВЛЕННЯ), ТО  
НА ОБИДВА ЗАТВОРИ ПОТРАПИТЬ  
ВИСОКИЙ ПОЗИТИВНИЙ РІВЕНЬ, ЧЕРЕЗ  
ЩО ПОВНІСТЮ ВІДКРИТИМ БУДЕ  
ТІЛЬКИ N- КАНАЛ ТРАНЗИСТОРА VT2 ТА  
ЗАКРИТИМ Р-КАНАЛ ТРАНЗИСТОРА VT1.  
ЗАРАЗ У ВУЗЛІ ДЕ З’ЄДНАНІ СТОКИ ОБОХ  
ТРАНЗИСТОРІВ БУДЕ НУЛЬОВИЙ

## **ПОТЕНЦІАЛ, ЯКИЙ ПОТРАПЛЯЄ ЧЕРЕЗ ВІДКРИТИЙ N- КАНАЛ ТРАНЗИСТОРА VT2. ТОДІ МОЖНА ЗАПИСАТИ $U_{вих}^0 = 0$ .**

Побудова цифрових мікросхем на основі транзисторно-транзисторної логіки має декілька схемних рішень, але вони не дуже відрізняються від базового варіанту.

Схема логічного елемента базової ТТЛ логіки представлена на рис. 2.12. Логічну функцію у такій схемі виконує багатомітерний транзистор VT1 [ ].

Якщо хоча б один із перемикачів X1 або X2 буде замкнуто на нульовий провід, то відповідний емітерний перехід транзистора VT1 буде відкритим, і через нього буде протікати струм по колу: позитивний затискач джерела живлення – через резистор R1 – перехід база-емітер VT1 – провід заземлення – негативний затискач джерела живлення. В колі колектора VT1, а відповідно і колі бази VT2 струм буде відсутнім, тобто VT2 буде перебувати у режимі відсічки. Струм через транзистор VT2, а відповідно і струм бази VT4 буде практично дорівнювати нулю. Транзистор VT4, через це також буде перебувати у режимі відсічки і на виході схеми Y буде високий рівень напруги логічної одиниці. При цьому напруга на колекторі VT2 і на базі VT3 буде максимальною і VT3 буде перебувати у повністю відкритому стані (режим насичення).

Рисунок 2.12 – Логічний елемент ТТЛ-логіки

Якщо обидва перемикачі X1 та X2 перемкнуті на лінію логічної одиниці “1”, то обидва емітерні переходи VT1 закриються. Струм тоді буде протікати по колу: позитивний затискач джерела живлення – через резистор R1 – перехід база-колектор VT1 на базу VT2. Транзистор VT2 після цього увійде в режим насичення. Струм через нього, а відповідно і струм бази VT4 буде максимальним, і транзистор VT4 також перейде у режим насичення. Тому на виході з’явиться низький рівень напруги логічного нуля. При цьому напруга на колекторі VT2 і на базі VT3 буде близька до нуля і транзистор VT3 перейде у закритий стан (режим відсічки).

Наступна схемна реалізація ТТЛ має назву ТТЛ із відкритим колектором, спрощена схема якого наведена на рис. 2.13. Колектор вихідного транзистора VT3 не під'єднано до жодної точки схеми. Тому, для забезпечення працездатності між виводом колектора і позитивним виводом джерела живлення необхідно підключити зовнішній опір.

Рисунок 2.13 – Схема комірки ТТЛ із відкритим колектором

Цей різновид ТТЛ є достатньо простим для реалізації та відповідно дешевим. Мікросхеми ТТЛ із відкритим колектором використовують для підключення, наприклад, елементів індикації – світлодіоди, сьомисегментні індикатори.

Мікросхеми ТТЛ із Z-станом. Z-станом цифрових мікросхем (або високоімпедансним, або третім, якщо першим вважати стан логічного нуля, другим – стан логічної одиниці) називають стан заборони прийому або передавання інформації. При переході мікросхеми у Z-стан вихідний опір логічного елементу наближується до нескінченності, а вихідний струм наближується до нуля. На рис.2.14, справа наведено комірку ТТЛ, яка може переходити до третього стану, а зліва, таблицю істинності комірки.

Рисунок 2.14 – Схема комірки ТТЛ із Z-станом

Якщо, на вхід заборони E0 інвертора DD1 встановити рівень логічного нуля, то на виході інвертора DD1 буде встановлено високий рівень логічної одиниці, діод VD1 буде запертим, що еквівалентно розриву ланцюга і схема буде функціонувати, подібно до базового елементу ТТЛ, який виконує функцію ТА-НІ. Якщо змінити на вході заборони E0 інвертора DD1 сигнал логічного нуля на сигнал логічної одиниці, то на виході інвертора DD1 буде встановлено низький рівень логічного нуля і діод VD1 буде відкритим. Точка <A> схеми буде знаходитися під напругою, яка буде майже дорівнювати нулю вихідного сигналу інвертора DD1. Це призведе до того, що транзистор VT3 буде запертим. Струм через

транзистор VT2, а відповідно і струм бази VT4 буде близько до нульового значення і транзистор VT4 також буде запертим. Таким чином, обидва транзистори і VT3 і VT4 – будуть одночасно запертими незважаючи на рівень сигналу (хоч нуль, хоч одиниця) на інформаційних входах X1 і X2. Це і є Z – станом мікросхеми.

Потрібно зазначити, що мікросхеми із третім станом, при знаходженні у цьому стані споживають набагато менш електроенергії, ніж інші, тому сучасні реалізації мікросхем не тільки ТТЛ а і КМОН мають здатність переходити до третього стану.

Одним з недоліків ТТЛ є відносно невелика швидкодія. Цей недолік можна пояснити тим, що при перемиканні транзистора з режиму насичення в режим відсічки база транзистора стає насиченою неосновними носіями заряду. І колекторний струм транзистора буде протікати до тих пір, поки неосновні носії заряду не перемістяться за бази до колектора. Для підвищення швидкодії було запропоновано між базою і колектором транзистора ТТЛ комірки включити швидкодіючий перетин Шоттки. Таку модифіковану ТТЛ-структуру називають **ТТЛШ**. В основі транзистора із діодом (перетином) Шоттки знаходиться схема ненасиченого РТЛ-ключа. При наявності діода Шоттки між базою та колектором неосновні носії будуть переходити з бази в колектор не скрізь колекторний р-п перетин, а скрізь перетин Шоттки. Ефект Шоттки знижує порогову напругу на відміну від звичайних 0,7 В для кремнієвих діодів до 0,2–0,3 В. Це дозволяє знизити життя неосновних носіїв у напівпровіднику. На схемах транзистор с перетином Шоттки позначають, як показано на рис. 2.15.

Рисунок 2.15 – Зображення транзистора з перетином Шоттки

На основі ТТЛШ електронна промисловість почала випускати цифрові мікросхеми серії 74S та 74LS (літера “S” означає прізвище фізика Schottky)– аналогі вітчизняних 531 та 555 серій мікросхем. Мікросхеми 74LS (Low Schottky) окрім високої швидкодії мають додаткову позитивну властивість – низьке споживання електроенергії, тобто вони є дуже економічними. Як бачимо з рис.2.16 на якому наведено схему комірки мікросхеми 74LS замість вхідного багатомірного транзистора застосовано матрицю діодів

Шоттки, і ця схема нагадує схему діодно-транзисторної логічної комірки.

Рисунок 2.16 – Комірки мікросхеми: а) серії 74S; б) серії 74LS

**Емітерно–зв’язна логіка** (ЕЗЛ). Мікросхеми серії ЕЗЛ є найбільш швидкодіючими логічними мікросхемами. Але їх схемотехнічна структура вкрай відрізняється від попередніх КМОН та ТТЛ мікросхем. В мікросхемах ЕЗЛ використовують біполярні транзистори, та на цьому схожість із ТТЛ закінчується. По-перше, транзистори в мікросхемах ЕЗЛ функціонують у лінійному (активному) режимі, та не переходять у ключовий режим (відсічки та насичення). Базова комірка елементу ЕЗЛ наведена на рис. 2.17. цей елемент реалізує логічну функцію АБО (АБО-НІ по виходу Y1).

Рисунок 2.17 – Комірка елементу ЕЗЛ

Основою ЕЗЛ є вхідний диференційний емітерний каскад зібраний на транзисторах VT1 та VT2. Якщо

$$U_{ax1} = U_{ax2}; I_1 = I_2; U_{ax1} > U_{ax2}; I_1 = I_0 + \Delta I; I_2 = I_0 - \Delta I.$$

З цих виразів зрозуміло, що рівні логічного нуля і логічної одиниці є невеликими, та слабо відрізняються. Ця особливість потенційно знижує завадостійкість комірки ЕЗЛ. Тому було запропоновано таке схемне рішення. Для підвищення завадостійкості колектори VT1 та VT2 під’єднані до нульового проводу, в ланцюг емітерів цих транзисторів включено джерело стабільного струму на транзисторі VT3, який є джерелом опорної напруги (ДОН)  $U = -5,2В$ . При потраплянні на обидва інформативних входи X1 та X2 логічного нуля, опорна напруга стає більш позитивною, ніж потенціали баз VT1 та VT2, відповідно транзистор VT3 відкрито у більшій ступені, ніж VT1 та VT2. Це означає, що струм скрізь резистор R3 буде більшим, чим через R1 і напруга на колекторі VT3 буде більш негативною, чим на колекторах VT1 та VT2.

ДОН зібрано на транзисторі VT4 по схемі емітерного повторювача. Подільник напруги, що складається з резисторів R5 і R6 та діодів VD1 и VD2 забезпечує постій напругу на базі транзистора, а відповідно струм через транзистор VT4 буде постійним і падіння напруги на резисторі R4 також буде постійним. Це постійне падіння напруги на резисторі R4 подано на базу

транзистора VT3 як опорне. діоді VD1 и VD2 крім усього також забезпечують температурну стабілізацію ДОН.

Для підвищення коефіцієнта розгалуження на виході елементу ЕЗЛ до колекторів диференційного каскаду VT1-VT2 підключено емітерні повторювачі на транзисторах VT5 і VT6.

### **2.3 Характеристики цифрових мікросхем, особливості застосування мікросхем, створених за різними інтегральними технологіями**

Не зважаючи на спосіб проектування цифрового радіоелектронного пристрою, розробник повинен знати та розуміти параметри, які характеризують роботу цифрової мікросхеми. Оскільки розробники програм автоматизації схеми технічного проектування намагаються максимально наблизити процес імітації роботи електронного пристрою до реального, то варто чекати на те, що при роботі з такими програмами ми будемо стикатися з характеристиками моделей, які максимально наближені до реальних пристроїв.

**ПАРАМЕТРИ ЦИФРОВИХ ІНТЕГРАЛЬНИХ МІКРОСХЕМ (ІМС) ПОДІЛЕНІ НА ДВІ ГРУПИ: СТАТИЧНИХ ПАРАМЕТРІВ ТА ДИНАМІЧНИХ ПАРАМЕТРІВ. СЛІД ЗАЗНАЧИТИ, ЩО ПАРАМЕТРИ ЯКІ БУДУТЬ РОЗГЛЯНУТО НИЖЧЕ ПРИТАМАННІ ВСІМ ЦИФРОВИМ МІКРОСХЕМАМ, НЕЗВАЖАЮЧИ НА КОНКРЕТНУ ФУНКЦІЮ, ЯКУ ВИКОНУЄ КОНКРЕТНА МІКРОСХЕМА. РІЗНИЦЮ МІЖ ПАРАМЕТРАМИ МОЖНА ЛИШЕ ПОБАЧИТИ МІЖ ПАРАМЕТРАМИ МІКРОСХЕМ, ПОБУДОВАНИХ ЗА РІЗНИМИ**

**ТЕХНОЛОГІЯМИ, ТОБТО ТТЛ, ТТЛШ АБО КМОН-МІКРОСХЕМИ БУДУТЬ МАТИ ТИ Ж САМІ ПАРАМЕТРИ, АЛЕ ЦІ ПАРАМЕТРИ МАТИМУТЬ РІЗНІ ЧИСЛОВІ ЗНАЧЕННЯ.**

**СТАТИЧНІ ПАРАМЕТРИ ХАРАКТЕРИЗУЮТЬ РОБОТИ МІКРОСХЕМИ ПРИ СТАТИЧНИХ (НЕЗМІННИХ) ЛОГІЧНИХ РІВНЯХ НА ВХОДАХ ТА ВИХОДАХ МІКРОСХЕМИ. ДО СТАТИЧНИХ ПАРАМЕТРІВ НАЛЕЖАТЬ ТАКІ:**

1. Напруга джерела живлення –  $U_{жив}$ , В.
2. Вхідна напруга логічного нуля -  $U_{вх}^0$ , В.
3. Вхідна напруга логічної одиниці -  $U_{вх}^1$ , В.
4. Вихідна напруга логічного нуля -  $U_{вих}^0$ , В.
5. Вихідна напруга логічної одиниці -  $U_{вих}^1$ , В.
6. Вхідний струм логічного нуля -  $I_{вх}^0$ , мА.
7. Вхідний струм логічної одиниці -  $I_{вх}^1$ , мА.
8. Вихідний струм логічного нуля -  $I_{вих}^0$ , мА.
9. Вихідний струм логічної одиниці -  $I_{вих}^1$ , мА.
10. Коефіцієнт розгалуження за виходом –  $K_p$  визначає кількість входів мікросхем-навантаження (цієї ж серії), які можна підключити до даної мікросхеми без втрати її працездатності, тобто цей параметр визначає навантажувальну здатність ІМС.
11. Коефіцієнт об'єднання за входом –  $K_{об}$  визначає кількість входів мікросхеми, за якими реалізується виконувана функція.



12. Порогова напруга –  $U_{пор}$ . В, це найменше та найбільше значення відповідних рівнів напруги, при яких починається перехід логічного елемента до протилежного стану. Відповідно розрізняють  $U_{пор}^1$  та  $U_{пор}^0$ .
13. Напруга статичної перешкоди –  $U_{ст.п}$  В, це максимально допустимий рівень статичної напруги на вході при якому мікросхема не втрачає своєї працездатності, тобто цей параметр характеризує завадостійкість ІМС.
14. Середня споживана потужність –  $P_{ср.}$ , мВт, визначає величину електричної потужності яку споживає ІМС.

Динамічні параметри характеризують роботу ІМС в момент переключення напруги логічного нуля в напругу логічної одиниці та навпаки. Динамічні параметри визначають швидкодію мікросхеми за рахунок часу передавання сигналу зі входу на вихід. Швидкодію визначають час перемикання з різних рівнів логічного сигналу, а також час затримки сигналу. До динамічних параметрів належать такі:

1. Час перемикання із рівня логічного нуля в рівень логічної одиниці –  $t_{0,1}$ , нс, це час, за який напруга сигналу на виході або вході зростає від 0,1 до 0,9 значення рівня логічної одиниці.
2. Час перемикання із рівня логічної одиниці в рівень логічного нуля –  $t_{1,0}$ , нс, це час, за який напруга сигналу на виході або вході зменшується від 0,9 до 0,1 значення рівня логічної одиниці.
3. Час затримки поширення сигналу при перемиканні із нуля в одиницю –  $t_{0,1.затр.}$ , нс. Це інтервал часу між вхідним та вихідним імпульсами при переході вихідної напруги від рівня логічного нуля до рівня логічної одиниці, який виміряний, згідно ГОСТ 19489-74, на рівні 0,5 логічної одиниці.
4. Час затримки поширення сигналу під час перемикання із одиниці в нуль –  $t_{1,0.затр.}$ , нс. Це інтервал часу між вхідним та вихідним імпульсами при переході вихідної напруги

- від рівня логічної одиниці до рівня логічного нуля, який виміряний на рівні 0,5.
5. Середній час затримки поширення сигналу –  $t_{\text{сер.затр.}}$ , нс, час, який характеризує швидкодію сигналу. Це інтервал часу, який дорівнює половині суми часу затримки поширення сигналу при включенні та виключенні логічного елемента. Тобто  $t_{\text{сер.затр.}} = (t_{0,1.\text{затр.}} + t_{1,0.\text{затр.}}) / 2$ .
  6. Ємність навантаження –  $C_n$ , пФ. Це сумарна ємність зовнішніх ланцюгів, які підключено до виходу мікросхеми;
  7. Опір навантаження –  $R_n$ , кОм. Це сумарний активний опір зовнішніх ланцюгів, які підключено до виходу мікросхеми;

На рис. 2.18 продемонстровано графік вхідного та вихідного сигналу інвертора де показано графічне роз'яснення динамічних параметрів.

До параметрів цифрових мікросхем також відносять **порогові напруги** логічного нуля і логічної одиниці. Пороговою напругою вважають найменше для  $U_{\text{вх.нрз}}^1$  та найбільше для  $U_{\text{вх.нрз}}^0$  значення відповідних рівнів, при якій починається перемикання логічного елемента в інший стан. Кількісно його характеризують точкою на амплітудній характеристиці логічного елемента, в якій модуль диференційного коефіцієнту підсилення мікросхеми дорівнює одиниці.

Ще одним параметром, що характеризує цифрову мікросхему є **завадостійкість**. Статичну завадостійкість оцінюють як мінімальне різницю між значеннями вихідного і вхідного сигналів даного рівня:

Під статичною завадостійкістю будемо вважати мінімальне значення напруги завади на виході логічного елемента, яке може

викликати спрацьовування підключеного до нього елемента цієї ж серії мікросхем.

При малій тривалості завади, якщо її тривалість менше або приблизно дорівнює значенню затримки розповсюдження, напруга завади може бути значно більшою, оскільки для вимірювання стану перемикачів, що входять до складу логічного елемента, потрібна не тільки амплітуда сигналу, але певний електричний заряд. Він забезпечує перезарядження конденсаторів і розсмоктування накопиченого надлишкового заряду в базах перемикачів на біполярних транзисторах.

Динамічну завадостійкість зазвичай визначають за допомогою графіку, який пов'язує припустиме значення напруги завади та її тривалість.

#### Рисунок 2.18 – Визначення динамічних параметрів цифрових мікросхем

**2.4. Особливості реалізації цифрової схемотехніки у програмах автоматизації схемотехнічного проектування**

**СПОЧАТКУ ПРОГРАМА PSPICE БУЛА ПРИЗНАЧЕНА ДЛЯ МОДЕЛЮВАННЯ ЧИСТО АНАЛОГОВИХ ПРИСТРОЇВ. У ВЕРСІЇ PSPICE 3.05 З'ЯВИЛАСЯ МОЖЛИВІСТЬ ЛОГІЧНОГО МОДЕЛЮВАННЯ ЦИФРОВИХ ПРИСТРОЇВ І ІНТЕРФЕЙСУ МІЖ АНАЛОГОВИМИ Й ЦИФРОВИМИ КОМПОНЕНТАМИ. ОДНАК ПРИ ЦЬОМУ АНАЛОГОВІ Й ЦИФРОВІ БЛОКИ ПОВИННІ БУТИ РОЗТАШОВУВАТИСЯ ПОСЛІДОВНО. СПОЧАТКУ, НАПРИКЛАД, МОДЕЛЮЄТЬСЯ АНАЛОГОВИЙ БЛОК, РЕЗУЛЬТАТИ РОЗРАХУНКІВ**

**ОЦИФРОВУЮТЬСЯ Й ЗАПИСУЮТЬСЯ У ФАЙЛ, ЩО ПЕРЕДАЄТЬСЯ В ОКРЕМУ ПРОГРАМУ ЛОГІЧНОГО МОДЕЛЮВАННЯ ЦИФРОВОГО ПРИСТРОЮ, І НАВПАКИ. ПОЧИНАЮЧИ З ВЕРСІЇ 4 ПРОГРАМА PSpICE ЗАБЕЗПЕЧУЄ ПРИНЦИПОВО НОВУ МОЖЛИВІСТЬ МОДЕЛЮВАННЯ ДОВІЛЬНИХ ЗМІШАНИХ АНАЛОГО–ЦИФРОВИХ ЛАНЦЮГІВ ЗІ ЗВОТНИМИ ЗВ'ЯЗКАМИ, ЯКІ, ЗОКРЕМА МОЖУТЬ СКЛАДАТИСЯ ТІЛЬКИ ІЗ ЦИФРОВИХ ПРИСТРОЇВ І НЕ МІСТИТИ АНАЛОГОВИХ БЛОКІВ. ЗВИЧАЙНО ЗМІШАНІ ЛАНЦЮГИ МОДЕЛЮЮТЬСЯ В РЕЖИМІ .TRAN (РОЗРАХУНОК ПЕРЕХІДНИХ ПРОЦЕСІВ), ОДНАК ІНШІ РЕЖИМИ ТАКОЖ ДОСТУПНІ. У РЕЖИМІ DC ЗАТРИМКИ СИГНАЛІВ У ЦИФРОВИХ БЛОКАХ ІГНОРУЮТЬСЯ Й РОЗРАХОВУЮТЬСЯ ЛОГІЧНІ РІВНІ ВИХОДІВ ЦИФРОВИХ ПРИСТРОЇВ У СТАЦІОНАРНОМУ РЕЖИМІ. У РЕЖИМАХ .AC, .NOISE, .TF І .SENS ЦИФРОВІ КОМПОНЕНТИ НЕ БЕРУТЬ УЧАСТЬ В АНАЛІЗІ МАЛОСИГНАЛЬНИХ ЧАСТОТНИХ ХАРАКТЕРИСТИК ЛАНЦЮГА, ЛИШЕ ДЛЯ АНАЛОГОВИХ ЧАСТИН АНАЛОГО–ЦИФРОВИХ І ЦИФРО–АНАЛОГОВИХ ІНТЕРФЕЙСІВ СКЛАДАЮТЬСЯ ЛІНЕАРИЗОВАНІ СХЕМИ ЗАМІЩЕННЯ ЇХ ВХІДНИХ І ВИХІДНИХ КОМПЛЕКСНИХ ОПОРІВ. РОЗГЛЯНЕМО СПЕЦИФІКУ МОДЕЛЮВАННЯ ЦИФРО–АНАЛОГОВИХ ПРИСТРОЇВ.**

Рисунок 2.19– Модель цифрової інтегральної схеми

Реальні цифрові інтегральні схеми (IC) у програмі PSpice представлені у вигляді примітивів Uxxx, що відображують їхнє функціонування на логічному рівні, і двох аналого–цифрових і цифро–аналогових інтерфейсів А/Ц и Ц/А, що відображають їх вхідні й вихідні каскади (рис. 2.19). У завданні на моделювання вказуються тільки примітиви цифрових пристроїв Uxxx. Якщо при цьому цифрові IC з'єднуються безпосередньо один з одним, то блоки інтерфейсів до уваги не беруть. Якщо ж до входу або виходу IC підключений аналоговий компонент, то необхідно включити відповідний інтерфейс. У принципі вони можуть бути вручну

включені в опис завдання на моделювання (у файлі .sig) як окремі компоненти, однак програма PSpice робить це автоматично.

Отже, змішані аналого–цифрові ланцюги складаються з компонентів трьох типів:

- 1) аналогові компоненти;
- 2) пристрої сполучення аналогових і цифрових компонентів (пристрої інтерфейсу);
- 3) цифрові компоненти.

Відповідно розрізняють три типи вузлів:

- 1) аналогові вузли, до яких підключені тільки аналогові пристрої;
- 2) цифрові вузли, до яких підключені тільки цифрові пристрої;
- 3) вузли інтерфейсу, до яких підключена комбінація аналогових і цифрових пристроїв

Програма PSpice автоматично розщеплює кожний вузол інтерфейсу на два вузли – суто аналоговий і суто цифровий – виключає між ними макро– модель аналого–цифрового або цифро–аналогового інтерфейсу. Крім того, до моделей інтерфейсів автоматично підключається джерело живлення цифрових схем.

Логічні рівні цифрових вузлів приймають одне із шести значень [2]:

- 1– високий рівень;
- 0– низький рівень;
- R – позитивний фронт (Raise, перехід зі стану “0” в “1”);
- F– негативний фронт (Fall, перехід зі стану “1” в “0”);
- X – невизначений стан (може приймати значення “0”, “1” проміжний або нестабільний стан);
- Z – стан високого вихідного опору (логічний рівень може бути високим, низьким, проміжним або нестабільним).

**ПРИ ОБЧИСЛЕННІ ЛОГІЧНИХ РІВНІВ ВУЗЛІВ, ДО ЯКИХ ПІДКЛЮЧЕНО КІЛЬКА ЦИФРОВИХ КОМПОНЕНТІВ, ПРИЙМАЮТЬСЯ В УВАГУ ВИХІДНІ ОПОРИ ДЖЕРЕЛ СИГНАЛІВ. НАВЕДЕМО ДАЛІ ОПИС ПРИСТРОЇВ ІНТЕРФЕЙСУ Й ЦИФРОВИХ КОМПОНЕНТІВ.**

Пристрої інтерфейсу

Пристрої інтерфейсу включаються між аналоговими й цифровими компонентами й виконують дві функції. По-перше, з

їхньою допомогою при моделюванні електричних процесів в аналоговій частині ланцюга задаються схеми заміщення вхідних і вихідних каскадів цифрових компонентів, з'єднаних безпосередньо з аналоговими компонентами. По-друге, вони забезпечують перетворення електричної напруги в логічний рівень і навпаки щоб забезпечити обмін даними між підпрограмами моделювання електричних процесів в аналоговій частині ланцюга й логічного моделювання цифрової частини, вони підрозділяються на пристрої передачі даних від аналогових на вхід цифрових компонентів, які називають інтерфейс А/Ц (Digital Output - цифровий вихід), і на пристрої передачі даних від цифрових на вхід аналогових компонентів, або інтерфейс Ц/А (Digital Input - цифровий вхід).

Якщо аналогові й цифрові компоненти взаємодіють у процесі моделювання, пристрої інтерфейсу включаються в схему заміщення ланцюга автоматично, коли вони з'єднуються один з одним. Користувач при цьому не включає пристрій сполучення на схему або у файл опису ланцюга з розширенням \*.cir. Для забезпечення такого режиму попередньо в бібліотеки цифрових компонентів включаються асоційовані з кожним компонентом моделі пристроїв інтерфейсу, оформлені у вигляді макромоделей. При розщепленні вузла інтерфейсу для автоматичного включення пристрою інтерфейсу програма PSpice створює новий цифровий вузол. Перелік цих вузлів, створених програмою, міститься у вихідному файлі з розширенням \*.out у розділі Generated Ato and Dto. Відзначимо, що вузол інтерфейсу характеризується електричною напругою, а додатковий цифровий вузол - логічним станом. Користувач має право безпосередньо включити в опис ланцюга пристрою інтерфейсу А/Ц и Ц/А, при цьому програма нові пристрої включати вже не буде.

Розглянемо як приклад змішаний ланцюг на рис. 2.20а. Він складається із двох цифрових компонентів - генератора цифрового сигналу U1 і логічного вентиля 2 I-HE, представленого у вигляді макромоделі X1. На один вхід вентиля X1 підключається генератор аналогового сигналу VSIN, на іншій - генератор цифрового сигналу U1, а до виходу – аналоговий RC-ланцюг. Таким чином, тут є два вузли інтерфейсу 1 і 2. До вузла 1 підключений аналоговий компонент VSIN і вхід цифрового компонента X1, тому між цими компонентами програма включить у схему заміщення інтерфейс

A/Ц и створить додатковий цифровий вузол 1\$Ato. Аналогічно між виходом цифрового компонента X1 і аналоговим RC-ланцюгом буде включений інтерфейс Ц/А и створений додатковий цифровий вузол 3\$Dto, як показано на рис.2.20, б. Імена додаткових цифрових вузлів складаються за наступним правилом: на початку їхнього імені повторюється ім'я вузла інтерфейсу, до нього додається символ \$ і потім суфікс Ato або Dto залежно від типу інтерфейсу (ім'я вузла на виході А/Ц здобуває суфікс Ato, на вході Ц/А - Dto). Якщо до одного вузла інтерфейсу підключається не один, а кілька однотипних компонентів, то утворюються додаткові цифрові вузли, наприкінці імен яких додаються цифри 2, 3 ... Крім того, у схемі заміщення на рис. 2.20 б до інтерфейсу Ц/А автоматично підключається джерело живлення через глобальний вузол \$G\_POS.

Відзначимо, що схема заміщення на рис. 2.20 б містить аналогові вузли 1, 3, \$G\_POS і цифрові вузли 1\$AtoD, 2, 3\$DtoA. Потенціали аналогових вузлів і логічні стани цифрових вузлів можна вивести на друк, наприклад, за директивою

```
.PRINT TRAN V(1) D(2) V(3)
```

а)

б)

**Рисунок 2.20– Приклад змішаного аналого-цифрового ланцюга (а) і його схеми заміщення (б)**

Отже, кожному реальному цифровому компоненту в програмі PSpice ставляться у відповідність:

1) два пристрої інтерфейсу для сполучення з аналоговими пристроями, які можуть підключатися до його входу й виходу, названі для стислості інтерфейсами Ц/А и А/Ц; вони здійснюють обмін даними між підпрограмами моделювання аналогових і цифрових пристроїв програми PSpice;

2) моделі вхід/вихід, що відображають його вхідні й вихідні комплексні опори;

3) моделі динаміки, що враховують запізнювання сигналів. Моделі цифрових компонентів і асоційовані з ними супутні моделі містяться в спеціальних бібліотеках.

Модель вхід/вихід

Моделі вхід/вихід, асоційовані з кожним цифровим компонентом, мають тип UIO і задаються форматом

.MODEL <ім'я моделі вхід/вихід> UIO) [<параметри моделі>]

Параметри моделі вхід/вихід наведені в табл.2.5

Таблиця 2.5

Параметри моделі вхід/вихід

Ім'я параметру	Параметр	розмірність	значення по умовчання
INLD	Вхідна ємність	Ф	0
OUTLD	Вихідна ємність	Ф	0
DRVH	Вихідний опір високого рівня	Ом	50
DRVL	Вихідний опір низького рівня	Ом	50
DRVZ	Вихідний опір витоку ланцюга, що моделюється як ланцюг зберігання заряду	Ом	$250 \cdot 10^3$
INR	Вхідний опір витоку ланцюга, що моделюється як ланцюг зберігання заряду	Ом	$30 \cdot 10^{-3}$
TSTOREMN	Мінімальний час збереження заряду ланцюга, що моделюється як ланцюг зберігання заряду	с	$1 \cdot 10^{-3}$
AtoD1	Ім'я макромоделі інтерфейсу А/Ц першого рівня		AtoDDefault
DtoA1	Ім'я макромоделі інтерфейсу Ц/А першого рівня		DtoADefault
AtoD2	Ім'я макромоделі		AtoDDefault



	інтерфейсу А/Ц другого рівня		
DtoA2	Ім'я макромоделі інтерфейсу Ц/А другого рівня		DtoADefault
AtoD3	Ім'я макромоделі інтерфейсу А/Ц третього рівня		AtoDDefault
DtoA3	Ім'я макромоделі інтерфейсу Ц/А третього рівня		DtoADefault
AtoD4	Ім'я макромоделі інтерфейсу А/Ц четвертого рівня		AtoDDefault
DtoA4	Ім'я макромоделі інтерфейсу Ц/А четвертого рівня		DtoADefault
TSWLH1	Час перемикання 0-1 для DtoA1	с	0
TSWLH2	Час перемикання 0-1 для DtoA2	с	0
TSWLH3	Час перемикання 0-1 для Dto3	с	0
TSWLH4	Час перемикання 0-1 для Dto4	с	0
TSWHL1	Час перемикання 1-0 для Dto1	с	0
TSWHL2	Час перемикання 1-0 для Dto2	с	0
TSWHL3	Час перемикання 1-0 для Dto3	с	0
TSWHL4	Час перемикання 1-0 для Dto4	с	0
TPWRT	Граничне значення тривалості імпульсу	с	Дорівнює мінімальній затримці
DIGPOWER	Ім'я макромоделі		DIGIFPWR

	джерела живлення		
--	------------------	--	--

Вхідна й вихідна ємності INLD, OUTLD приймаються до уваги при розрахунку часу затримки. Вихідні опори цифрових пристроїв задаються параметрами DRVH, DRVL моделі вхід/вихід UIO (рис.2.21 а). Вихідний опір компонента, що перебуває в стані "1", позначається як DRVH, у стані "0" - DRVL. У програмі PSpice вихідні опори компонентів приймають значення в діапазоні від DIGDRVF (Forcing strength) до DIGDRVZ (Z strength), що у логарифмічному масштабі розбивається на 64 рівня (максимальному опору DIGDRVZ привласнюють код 0, а мініимальному DIGDRVF – код 63). За замовчуванням DIGDRVF=2 Ом, DIGDRVZ=20 кОм; їхні значення перепризначаються за директивою .OPTIONS. У конфліктних ситуаціях, коли до одного вузла підключаються вентиля з різними вихідними опорами, логічний рівень вузла встановлюється вентилям з мінімальним вихідним опором, код якого більше кодів інших опорів у задане число раз. Це відношення кодів опорів задається параметром DIGOVRDRV директиви .OPTIONS, що за замовчуванням дорівнює 3. Коли є кілька вентилів із близькими вихідними опорами й різними логічними рівнями, вузлу привласнюється невизначений стан X. Часи перемикання вихідних каскадів цифрових ІС задаються параметрами TSWLHn, TSWHLn (труднощі їхнього визначення за довідковим даними полягають у тім, що звичайно приводяться значення загального часу перемикання всієї ІС).

Макромоделі інтерфейсів складаються користувачами й включаються в бібліотечний файл. Ці моделі відбивають характер вхідних/вихідних опорів цифрових компонентів різним ступенем подробности. Моделі, наявні в стандартній бібліотеці інтерфейсів програми PSpice, наведені в табл.2.6

Таблиця 2.6  
Моделі інтерфейсу

IO_LEVEL	Визначення
0	Поточне значення параметра DIGIOLVL. директиви .OPTIONS (по умовчання дорівнює 1)
1	Основна { найпростіша} модель, що має логічні стани 0,1,

	X,R і F(Ato1/Dto2)
2	Основна (найпростіша) модель без проміжного стану X (Ato2/Dto2)
3	Складна модель із проміжним станом X (Ato3/Dto3)
4	Складна модель без проміжних станів X, R і F (Ato4/Dto4)

**СКЛАДНІ МОДЕЛІ ТОЧНІШЕ ІМІТУЮТЬ НЕЛІНІЙНОСТІ ВХІДНИХ ОПОРІВ ЦИФРОВИХ ІС, ОДНАК ВИМАГАЮТЬ БІЛЬШИХ ОБЧИСЛЮВАЛЬНИХ ВИТРАТ. ІМЕНА МАКРОМОДЕЛЕЙ ІНТЕРФЕЙСІВ ВКАЗУЮТЬСЯ ЗА ДОПОМОГОЮ ПАРАМЕТРІВ АТО1, ДТО1, ..., АТО4, ДТО4. ВИБІР РІВНЯ МОДЕЛІ ІНТЕРФЕЙСУ ДЛЯ КОЖНОГО КОНКРЕТНОГО ЦИФРОВОГО ПРИСТРОЮ ВИРОБЛЯЄТЬСЯ ЗА ДОПОМОГОЮ ПАРАМЕТРА IO\_LEVEL**

Схеми заміщення найпростіших інтерфейсів першого рівня показані на рис. 2.21 У текстовому виді вони записуються в такий спосіб. Макромодель стандартного TTL-Інтерфейсу А/Ц першого рівня має вигляд:

```
subckt AtoD_STD A D DPWR DGND params: CAPACITANCE=0
O0 A DGND DO74 DGTLEN=D IO_STD
C1 A DGND {CAPACITANCE+0.1 pF}
.ends
```

```
.model D074 doutput (
+ s0name="X" s0vlo=0.8 s0vhi=2.0 s1name="0" s1vlo=-1.5 s1vhi=0.8
+ s2name="R" s2vlo=0.8 s2vhi=1.4 s3name="R" s3vlo=1.3
s3vhi=2.0
+s4name="X" s4vlo=0.8 s4vhi=2.0 s5name=1 s5vlo=2.0
s5vhi=7.0
+s6name="F" s6vlo=1.3. s6vhi=2,0 s7name="P" s7vlo=0.8
s7vhi=1.4)
```

```
model IO_STD uio (drvh=96.4 drvl=104
+ AtoD1="AtoD.STP"; AtoD2="AtoD_STD_NX"
AtoD3="AtoD_STD_E"
```

```

+ AtoD4="AtoD_STD_NX_E"
+ DtoA1="DtoA_STD" DtoA2="DtoA_STD_NX"
+ DtoA3="DtoA_STD_E"
+ DtoA4="DtoA_STD_NX_E"
+ tswl1=1.373ns tswfh1=3.382ns tswl2=1.346ns
+ tswlh2=3.424ns
+ tswl3=1.511ns tswlh3=3.517ns tswl4=1.487ns tswlh4=3.564ns
+ DIGROWER = "DIGIFPWR")

```

**МАКРОМОДЕЛЬ СТАНДАРТНОГО ТТЛ-ІНТЕРФЕЙСУ  
Ц/А ПЕРШОГО РІВНЯ МАЄ ВИГЛЯД:**

```

.subckt DtoASTD D A DPWR DGND params: DRVL=0 DRVH=0
CAPACITANCE=0
N1 A DGND DPWR DIN74 DGTLEN=D IO_STD
C1 A DGND {CAPACITANCE+0.1 pF}
.ends
.model DIN74 dinput(
+ s0name="O" s0tsw=3.5ns s0rlo=7.13 s0rhi=389 ; 7ohm, 0.09v
+ s1name="1" s1tsw=5.5ns s1rlo=467 s1rhi=200; 140ohm, 3.5v
+ s2name="X" s2tsw=3.5ns s2rlo=42.9 s2rhi=116; 31.3ohm, 1.35v
+ s3name="R" s3tsw=3.5ns s3rlo=42.9 s3rhi=116; 31.3ohm, 1.35v
+ s4nafne="F" s4tsw=3.5ns s4rlo=42.9 s4rhi=116; 31.3ohm, 1.35v
+ SSNAME="Z" S5TSW=3.5NS S5RLO=200K S5RHI=200K)

```

**БІЛЬШ СКЛАДНА МОДЕЛЬ ІНТЕРФЕЙСА А/Ц  
ПЕРШОГО РІВНЯ МАЄ СХЕМУ ЗАМІЩЕННЯ, НАВЕДЕНУ  
НА РИС. 2.22 ЇЇ ТЕКСТОВИЙ ОПИС МАЄ ВИГЛЯД:**

```

subckt AtoD__STD_E A D DPWR DGND params;
CAPACITANCE=0
O0 A DGND DO74 DGTNET=D Io_STD
C1 A DGND {CAPACITANCE+0.1pF}
DO DGND a D74CLMP
D1 1 2 D74
D2 2 DGND D74
R1 DPWR 3 4k
Q1 1 3 A 0 Q74; підкладинку підєднати до DGND

```

```
.ends
.model D74 D(IS=1e-16 RS=25 CJO=2pf)
.model D74CLMP D(IS=1e-15 RS=2 CJO=2pf)
.model Q74 NPN(ISE=1e-16 ISC=4e-16 BF=49 BR=.03 CJE=1pf
CJC=.5pf
+ CJS=3pf VJE=0.9v VJC=0.8v VJS=0.7v MJE=0.5 MJC=0.33
MJS=0.33
+ F=0.2NS TR=10NS RB=50 RC=20)
```

### ПРИСТРОЇ ЖИВЛЕННЯ

У програмі PSpice прийнято, що макромодель джерела живлення має ім'я DIGIFPWR і на неї автоматично робиться посилання при наявності в схемі макромоделей аналого-цифрових інтерфейсів. Конкретний зміст макромоделі джерела живлення визначається користувачем, а її текст міститься в бібліотеку цифрових пристроїв. Наведемо як приклад макромодель джерела напруги, що приймає за замовчуванням значення 5 В:

```
.subckt DIGIFPWR AGND optional: DPWR=$G_DPWR
+DGND=$G_DGND params: VOLTAGE=5.0v REFERENCE=0v
VDPWR DPWR DGND {VOLTAGE}
R1 DPWR AGND 1MEG
VDGND DGND AGND {REFERENCE}
R2 DGND AGND 1MEG
.ends
```

Тут вузлам цифрового живлення й цифрової "землі" привласнені глобальні імена \$G\_DPWR, \$G\_DGND (при бажанні їх можна змінити тут і одночасно в моделях всіх цифрових пристроїв). Вузол AGND - загальний вузол джерела живлення, що при включенні його в загальну схему з'єднується з вузлом аналогової "землі" 0. Для створення макромоделі джерела іншої напруги необхідно в завданні на моделювання (у файлі .cir) включити пропозицію виклику макромоделі джерела живлення, задавши за допомогою параметра VOLTAGE потрібне значення напруги живлення, і вказати після номера вузла "землі" 0 імена (номера) вузлів його виводів, наприклад:

XMPPOWER 0 MY\_PWR MY\_GND DIGIFPWR

+ params: VOLTAGE=9.0v

**ТУТ ВИКЛИКАЄТЬСЯ ДЖЕРЕЛО НАПРУГИ 9 В, ВИВОДИ ЯКОГО МАЮТЬ ІМЕНА MY\_PWR І MY\_GND(ЦІ ІМЕНА ВКАЗУЮТЬСЯ В МОДЕЛЯХ ПРИМІТИВІВ ЦИФРОВИХ ПРИСТРОЇВ, ЩО ПІДКЛЮЧАЮТЬСЯ ДО ЦЬОГО ДЖЕРЕЛА, ДИВ. НИЖЧЕ). УСКЛАДНИВШИ МАКРОМОДЕЛЬ, МОЖНА СТВОРИТИ ДЖЕРЕЛО ДЕКІЛЬКОХ НАПРУГ.**

### ГЕНЕРАТОРИ ЦИФРОВИХ СИГНАЛІВ

Генератори цифрових сигналів можна задати двома способами.

1. Визначення форми цифрового сигналу в завданні на моделювання за форматом (пристрій STIM)

Uxxx STIM (<кількість сигналів>,<формат>)

+ <+вузол джерела живлення> <-вузол джерела живлення>

+ <список вузлів>\* <ім'я моделі вхід/вихід>

+ [IO\_LEVEL=<номер макромоделі інтерфейсу вхід/вихід>]

+ [STIMULUS=<ім'я впливу> [TIMESTEP=<крок за часом>]

+ <команди опису форми сигналу>\*

Змінна <кількість сигналів> визначає кількість виходів генератора та дорівнює кількості різних генеруємих цифрових сигналів. Змінна <формат> – це специфікація формату змінної <дані>, у якій представлені логічні рівні сигналів генератора. Ця змінна являє собою послідовність цифр, загальне число яких дорівнює значенню змінної <кількість сигналів>. Кожна цифра приймає значення 1, 3 або 4, що означає двійкову, восьмеричну й шостнадцятиричну систему числення.

Підключення джерела живлення задається номерами вузлів <+вузол джерела живлення>, <-вузол джерела живлення>. Номера підключення виходів генератора до схеми задаються <списком вузлів>. Ім'я моделі вхід/вихід задається параметром <ім'я моделі вхід/вихід>, як для будь-якого цифрового пристрою.

STIMULUS - необов'язковий параметр, що позначає ім'я сигналу;

IO\_LEVEL - необов'язковий параметр для вибору однієї із чотирьох макромоделей інтерфейсу вхід/вихід (за замовчуванням 0);

TIMESTEP - не обов'язковий параметр для завдання періоду квантування (або кроку). При завданні моментів часу номером кроку

(мають суфікс "C") він множиться на величину кроку. За замовчуванням установлюється TIMESTEP=0. Цей параметр не приймається до уваги, якщо задані абсолютні значення моментів часу (мають суфікс "S").

Параметр <опис форми сигналу> являє собою довільну комбінацію однієї або декількох наступних рядків:

```
<t>, <логічний рівень>  
LABEL=<імя мітки>  
<t> GOTO <імя мітки> <n> TIMES  
<t> GOTO <імя мітки> UNTIL GT <дані>  
<t> GOTO <імя мітки> UNTIL GE <дані>  
<t> GOTO <імя мітки> UNTIL LT <дані>  
<t> GOTO <імя мітки> UNTIL LE <дані>  
<t> INCR BY <дані>  
REPEAT FOEREVER  
REPEAT <n> TIMES  
ENDREPEAT  
FILE=<імя файлу>
```

Кожна цифра змінної <дані> являє собою логічний рівень відповідного вихідного сигналу, що представлений у системі числення  $2m$ , де  $m$ —відповідна цифра змінної <формат>. Кількість вузлів в <списку вузлів> повинна дорівнювати значенню змінної <кількість сигналів>. Змінна <t> визначає моменти часу, у які задаються логічні рівні сигналу. Якщо перед значенням змінної <t> є символ "+", то ця змінна задає збільшення щодо попереднього моменту часу; у протилежному випадку вона визначає абсолютне значення відносно початку відліку часу  $t = 0$ . Суфікс "S" указує розмірність часу в секундах (допускається суфікс "n" - наносекунди й т.п.). Суфікс "C" означає вимір часу в кількості циклів, розмір яких визначається параметром TIMESTEP (змінна <крок за часом>). Змінна <дані> складається із символів "0", "1", "X", "R", "F або "Z", що інтерпретуються у заданому форматі. Змінна <n> задає кількість повторюваних циклів GOTO; значення  $n = -1$  задає нескінченне повторення циклу. Змінну <ім'я мітки> використовують при організації циклу за допомогою оператора переходу GOTO, що

передає керування на рядок, що впливає за оператором LABEL=<ім'я мітки>.

REPEAT FOREVER - початок нескінченно повторюваного циклу (еквівалентно конструкції REPEAT -1 TIMES).

REPEAT <n> TIMES – повторення n разів.

ENDREPEAT – кінець циклу REPEAT.

FILE – указание ім'я файлу, у якому знаходиться опис одного або декількох вхідних сигналів.

Наведемо приклад:

```
USIGNAL STIM(2,11) $G_DGND 1 2 IO_STM TIMESTEP =1ns
+0C 00; В момент часу t =0 стан обох вузлів дорівнював "0"
+LABEL=M
+1C 01 ; При t = 1 нс стан вузла 1 дорівнює "0", вузла 2 – "1"
+2C 11 ; При t = 2 нс стан обох вузлів дорівнює "1"
+3C GOTO M 3 TIMES; В момент часу =3 нс перехід на мітку M і
* виконується перша інструкція без додаткової затримки. Цикл
* повторюється 3 рази
```

2. Зчитування часової діаграми цифрового сигналу з файлу (пристрій FSTIM)

Пристрій FSTIM задають за форматом:

```
Uxxx FSTIM(<кількість виходів>) <+вузол джерела живлення>
+ <-вузол джерела живлення> <список вузлів>*
+<ім'я моделі вхід/вихід>
+ FILE=<ім'я файлу сигналів>
+ [IO_LEVEL=<номер макромоделі інтерфейсу вхід/вихід>]
+ [SIGNAMES=<ім'я впливу>
```

**ПАРАМЕТР <КІЛЬКІСТЬ ВИХОДІВ> ВИЗНАЧАЄ КІЛЬКІСТЬ ВУЗЛІВ, ДО ЯКИХ ПІДКЛЮЧАЮТЬСЯ СИГНАЛИ. ІНШІ ПАРАМЕТРИ МАЮТЬ ТОЙ ЖЕ ЗМІСТ, ЩО В П. 1. ЗВЕРНЕМО УВАГУ, ЩО ЗАВДАННЯ ЗА ДОПОМОГОЮ ПАРАМЕТРА FILE ІМ'Я ФАЙЛУ, У ЯКОМУ ОПИСАНІ СИГНАЛИ, ОБОВ'ЯЗКОВО. ФАЙЛ ЦИФРОВОГО СИГНАЛУ МОЖЕ БУТИ ОТРИМАНИЙ ШЛЯХОМ РЕДАГУВАННЯ ФАЙЛУ РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯ АБО СКЛАДЕНИЙ ВРУЧНУ.**

Файл цифрового сигналу має дві секції:



- 1) заголовок (header), що містить список імен сигналів;
- 2) список значень сигналу (transitions), що містить на одній або більше рядках моменти часу змін сигналів і колонки їхніх значень.

Наведемо спочатку приклад файлу цифрових сигналів:

\* Заголовок: містить імена сигналів

CLOCK, RESET, IN1, IN2 ; імена 4-х сигналів

\*Початок опису сигналів; відділяється від заголовка порожнім рядком

```
0 0000; двійкові коди сигналів
10ns 1100
20ns 0101
30ns 1110
40ns 0111
```

Заголовок має наступний формат:

[TIMESCALE=<значення>]

<ім'я сигналу 1>.,.<ім'я сигналу n>...

OCT(< 3-й біт сигналу>... 1-й біт сигналу>)...

HEX(< 4-й біт сигналу>...< 1-й біт сигналу>)...

**ІМЕНА СИГНАЛІВ МОЖУТЬ РОЗДІЛЯТИСЯ КОМАМИ АБО ПРОБІЛАМИ. ВОНИ РОЗТАШОВУЮТЬСЯ НА ОДНІЙ АБО ДЕКІЛЬКОХ РЯДКАХ, АЛЕ РЯДКИ ПРОДОВЖЕННЯ НЕ МАЮТЬ НА ПОЧАТКУ ЗНАКА "+". МАКСИМАЛЬНА КІЛЬКІСТЬ СИГНАЛІВ 255, НА ОДНОМУ РЯДКУ МОЖУТЬ РОЗМІЩАТИСЯ НЕ БІЛЬШЕ 300 СИМВОЛІВ. ІМЕНА СИГНАЛІВ ПЕРЕРАХОВУЮТЬСЯ В ТІМ ЖЕ ПОРЯДКУ, У ЯКОМУ ЇХНІ ЗНАЧЕННЯ ПРИВОДЯТЬСЯ В ЧАСОВИХ ДІАГРАМАХ. ЯКЩО ПЕРЕД ІМ'ЯМ СИГНАЛУ НЕ ВКАЗУЄТЬСЯ НАЗВА СИСТЕМИ ЧИСЛЕННЯ, ВОНА ВВАЖАЄТЬСЯ ДВІЙКОВОЮ. ІМЕНА СИГНАЛІВ, ЗАПИСАНИХ У ВОСЬМЕРИЧНОМУ КОДІ, ВИПЕРЕДЖАЮТЬСЯ КОДОМ OCT І ГРУПУЮТЬСЯ ПО 3 СИГНАЛИ (ІМЕНА ГРУП СИГНАЛІВ ЗАНОСЯТЬ У КРУГЛІ ДУЖКИ). ІМЕНА ШОСТНАДЦЯТИРІЧНИХ СИГНАЛІВ МАЮТЬ КОД HEX І ГРУПУЮТЬСЯ ПО 4 СИГНАЛИ.**

Наведемо ще один приклад файлу сигналів:

```
Clock Reset In1 In2
```

## HEX (Addr7 Addr6 Addr5 Addr4) HEX (Addr3 Addr2 Addr1 Addr0)

ReadWrite

0n 0000 00 0; для наочності між групами можна включати пробіли  
 10n 1100 4E 0  
 20n 0101 4E 1  
 30n 1110 4E 1  
 40n 0111 FF 0

Тут спочатку наведені імена чотирьох двійкових сигналів, потім дві групи сигналів, що задаються в шостнадцятирічному вигляді, і потім ще один бінарний сигнал. У наступному списку значень сигналів дані 7 колонок, що відповідають наведеним у заголовку сигналам.

Список значень має формат:

<час> <значення сигналу>\*

Список значень відділяється від заголовка порожнім рядком. Моменти часу й список значень сигналів повинні бути розділені принаймні одним пробілом. Для моментів часу змін сигналів вказуються їхні абсолютні значення (у секундах), наприклад 25ns 1.2E-9, 5.8, або збільшення щодо попереднього моменту часу. Ознака збільшення - знак + перед значенням моменту часу, наприклад +5ns. Кожне значення сигналу відповідає одиночному бінарному сигналу або групі сигналів, імена яких поміщені в групи з покажчиками системи числення ОСТ або HEX. Загальна кількість значущих цифр у значеннях сигналів повинне бути дорівнює загальній кількості бінарних сигналів і груп сигналів, позначених символами ОСТ і HEX. Розряди даних сигналу можуть приймати значення, наведені в табл. 2.7

Таблиця 2.7

### Значення сигналів

Сигнали	Двійкові	Восьмеричні	Шостнадцятирічні
Логічні/цифрові	0,1	0..7	0..F
Невизначені	X	X	X
Високого імпедансу	Z	Z	Z
Передній фронт	R	R	-

Задній фронт	F	F	-
--------------	---	---	---

Відмітимо, що задній фронт не може в шостнадцятирічній системі позначатися як F, тому що цей символ зайнятий під чисельне значення. Наведемо приклади генераторів цифрових сигналів типу FSTIM.

1. Нехай є файл dig1.stm, у якому описана часова діаграма сигналу IN1. Тоді в завдання на моделювання можна включити наступний опис генератора:

```
U1 FSTIM(1) $G_DPWR $G_DGND IN1 IO_STM FILE=DIG1 .STM
```

2. Нехай є файл flipflop.stm наступного змісту:

```
J K PRESET CLEAR CLOCK
```

```
0 0 0 010
```

```
10ns 0 0 111
```

**У ЗАВДАННЯ НА МОДЕЛЮВАННЯ ВКЛЮЧИМО ОПИС  
ГЕНЕРАТОРА СИГНАЛУ:**

```
U2 FSTIM(4) $G_DPWR $G_DGND CLK PRE J K IO_STM
```

```
+ FILE=ffipflop.stm SIGNAMES = CLOCK PRESET
```

У цьому прикладі першим двом вузлам CLK і PRE відповідають сигнали CLOCK і PRESET з файлу сигналів, ця відповідність встановлюється за допомогою опції SIGNAMES. Останнім вузлом J і K відповідають однойменні сигнали з файлу сигналів, тому їхні імена немає необхідності включати в опцію SIGNAMES. Сигнал CLEAR у даному прикладі не використовується (але на нього можна послатися в іншому генераторі FSTIM).

**2.5 ПРЕДСТАВЛЕННЯ ЦИФРОВИХ КОМПОНЕНТІВ У  
ПРОГРАМАХ АВТОМАТИЗАЦІЇ СХЕМОТЕХНІЧНОГО  
ПРОЕКТУВАННЯ**

Цифрові компоненти (примітиви) задаються по формату

- Uxxx<тип> [(<список параметрів>\*)]
- + <вузол джерела живлення>
- + <-вузол джерела живлення> <список вузлів>\*
- + <ім'я моделі динаміки>
- + <ім'я моделі вхід/вихід>
- + [MNTYMXDLY=<вибір значення затримки>]
- + [IO\_LEVEL=<рівень моделі інтерфейсу>]

Параметр <тип> указує тип логічного пристрою (їхній перелік наведений нижче, наприклад AND, NOR); у круглих дужках вказуються значення одиної або більше параметрів через кому (наприклад, для схеми И вказується кількість входів). Після списку вузлів підключення логічного пристрою їдуть імена двох моделей. Перша модель описує динамічні властивості пристрою, друга – характеристики вхідних і вихідних опорів.

Моделі динаміки мають ключові слова, наведені в табл. 2.8

Таблиця 2.8

### Ключові слова моделі динаміки

Ім'я моделі динаміки	Тип компонента
UADC	Аналого-цифровий перетворювач
UBTG	Двонаправлений перемикаючий вентиль
UDAC	Цифро-аналоговий перетворювач
UDLY	Цифрова лінія затримки
UEFF	Тригер із динамічним керуванням
UGATE	Стандартний вентиль
UGFF	Тригер з потенційним керуванням
UIO	Моделі входу/виходу цифрового пристрою
UPLD	Програмовані логічні матриці
UROM	Постійний запам'ятовуючий пристрій
URAM	Оперативний запам'ятовуючий пристрій
UTGATE	Вентиль із трьома станами

Моделі вхід/вихід мають ключове слово UIO.

Параметр MNTYMXDLY дозволяє конкретному індивідуальному пристрою призначити мінімальне, типове або максимальне значення часу затримки, зазначене в специфікації моделі його динаміки:

0 – значення затримки, задане параметром DIGMNTYMX директиви .OPTIONS

(за замовчуванням параметр дорівнює 2);

1 – мінімальне значення;

2 – типове значення;

3 – максимальне значення;

4 – розрахунок найгіршого випадку (мінімум/максимум).

Параметр IO\_LEVEL указує тип цифро-аналогового й аналого-цифрового інтерфейсу даного цифрового пристрою:

0 – у відповідності зі значенням параметра DIGIOLVL директиви .OPTIONS (за замовчуванням він дорівнює 1);

1 – інтерфейс Ato1/Dto1;

2 – інтерфейс Ato2/Dto2;

3 – інтерфейс Ato3/Dto3;

4 – інтерфейс Ato4/Dto4.

Запізнювання сигналу в примітивах цифрових пристроїв визначається у двох моделях: динаміки й вхід/вихід.

Модель динаміки визначає затримки поширення й такі тимчасові обмеження, як час установки (setup) і втримання (hold). Модель вхід/вихід задає вхідні й вихідні опори, ємності й час перемикання. Коли вихід примітива з'єднується з іншим примітивом, загальний час затримки поширення першого примітива дорівнює сумі часу встановлення напруги на його навантаженні й часі поширення сигналу, зазначеного в моделі динаміки. Час установлення напруги на навантаженні (loading delay) розраховується за формулою

$$\tau_{нав.} = 0,69R_{вих}C_{вих}$$

де  $R_{вих}$  – вихідний опір пристрою, дорівнює DRVH або DRVL залежно від логічного рівня на виході;  $C_{вих}$  – сума вхідних і вихідних ємностей цифрових пристроїв INLD, OUTLD, підключених до даного виводу.

Коли цифровий примітив підключений до аналогового пристрою, затримка поширення зменшується на величину, рівну часу перемикання, заданого в моделі вхід/вихід. Мінімальна тривалість сигналу на вході цифрового примітива, необхідна для зміни його логічного стану, повинна перевищувати час затримки, наведене в моделі динаміки (це обмеження не ставиться до цифрових ліній затримки). Більше короткі вхідні імпульси не викличуть на виході ніякого ефекту.

Наведемо прикладопис змішаного ланцюга на рис. 2.20, а:

```
Analog/Digital Interface Example
.OPT ACCT LIST LIBRARY EXPAND RELTOL =.001
.LIB DIG.LIB
VSIN 1 0 SIN(0 5V 1MEG)
U1 STIM(1, 1) $G_DPWR $G_DGND 2 IO_STD
+TIMESTEP =10ns
+(0C, 1) LABEL=BEGIN
+ (1C, 0) (2C, 1) (3C, X) (4C, 0) (5C, Z)
+6C GOTO BEGIN -1 TIMES
X1 1 2 3 133LA3
RL 3 0 25k
CL 3 0 5Pf
.tran 5ns 500ns
.PRINT TRAN V(1) D(2) V(3)
.PROBE
.END
```

Тут є звернення до бібліотечного файлу моделей цифрових компонентів dig.lib. Звернемо увагу, що цифрові ІС, навіть найпростіші, для яких є примітиви, представлені в бібліотеці у вигляді макромоделей, імена яких збігаються з позначенням по ЄСКД (правда, у латинській транскрипції). Це дозволяє користувачеві не замислюватися про правила опису кожної конкретної ІС надаючи це права розроблювачам бібліотек моделей.

Інформація про макромоделі пристроїв сполучення, що включаються автоматично в схему, їхніх іменах і іменах нових цифрових вузлів міститься у вихідному файлі з розширенням out:

```
***Generated AtoD and DtoA Interfaces***
*Analog/Digital interface for node 1
*Moving X1.U1:IN1 from analog node 1 to new digital node 1$AtoD
X$1_AtoD1 1 1$AtoD1 AtoD
*Analog/Digital interface for node 3
*Moving X1.U1:OUT1 from analog node 3 to new digital node 3$AtoD
X$3_DtoA1 3$DtoA 3 DtoA
*Analog/Digital interface power supply subckt
```

## X\$DIGIFPWR 0 DIGIFPWR

Як приклад розрахунку перехідних процесів наведемо фрагмент видачі даних у табличній формі за директивою .PRINT:

TIME	V(1)		D(2)	V(3)
0.000E+00	0.000E+00	1	3.551E+00	
5.000E-09	1.570E-01	1	3.551E+00	
1.000E-08	3.139E-01	0	3.551E+00	
1.500E-08	4.704E-01	0	3.551E+00	
2.000E-08	6.264E-01	1	3.551E+00	
2.500E-08	7.820E-01	X	3.551E+00	
3.000E-08	9.369E-01	X	3.551E+00	

Перейдемо тепер до опису цифрових компонентів різних типів, згрупувавши їх по наступних категоріях:

- багаторозрядні АЦП і ЦАП;
- вентиля;
- тригери;
- програмувальні логічні матриці;
- пристрої пам'яті;
- джерела постійних логічних сигналів;
- лінію затримки;
- функціональний опис цифрових пристроїв.

### Вентилі

Вентилі у програмах АСхП розділяють на базові (прості) та складні. До простих відносять ті, що наведені у п.2.1. Прості вентиля мають один вихід та декілька входів. Складні (або комбінаційні) вентиля поєднують у своїй структурі декілька простих вентилів. Приклади базових вентилів NOT, OR, AND наведені на рис. 2.1. Приклади складних вентилів наведено нижче.

### Рисунок 2.27 – Складні вентиля

Крім того, вентиля підрозділяються на два типи: стандартні вентиля й вентиля із трьома станами. Вентилі із трьома станами управляються сигналами дозволу. Коли цей сигнал має рівень "0",

вихідний сигнал вентиля має невизначений рівень "X" при високому вихідному опорі Z. Всі вентиля описуються по формату, який є загальним для всіх цифрових компонентів. Стандартні вентиля перелічені в табл. 2.11

Таблиця 2.11

### Параметри моделей стандартних вентилів

Тип	Параметр	Порядок слідування виводів	Функціонал. призначення
BUF	немає	Вх., вих.	Буфер
INV	немає	Вх., вих.	Інвертор
AND	N	Вх.1, вх2, ... вих.	Логічне ТА
NAND	N	Вх.1, вх2, ... вих.	Логічне ТА-НІ
OR	N	Вх.1, вх2, ... вих.	Логічне АБО
NOR	N	Вх.1, вх2, ... вих.	Логічне АБО-НІ
XOR	немає	Вх.1, вх2, вих.	Виключне АБО
NXOR	немає	Вх.1, вх2, вих.	Виключне АБО-НІ
BUFA	L	Вх.1, вх2, ... вих.1, вих.2...	Збірка буферів
INVA	L	Вх.1, вх2, ... вих.1, вих.2...	Збірка інверторів
ANDA	N, L	Вх.1, вх2, ... вих.1, вих.2...	Збірка логіки ТА
NANDA	N,L	Вх.1, вх2, ... вих.1, вих.2...	Збірка логіки ТА-НІ
ORA	N,L	Вх.1, вх2, ... вих.1, вих.2...	Збірка логіки АБО
NORA	N, L	Вх.1, вх2, ... вих.1, вих.2...	Збірка логіки АБО-НІ
XORA	L	Вх.1, вх2, ... вих.1, вих.2...	Збірка логіки виключне АБО
NXORA	L	Вх.1, вх2, ... вих.1, вих.2...	Збірка логіки виключне АБО-НІ
AO	N, L	Вх.1, вх2, ... вих.	Збірка логіки ТА-АБО
OA	N, L	Вх.1, вх2, ... вих.	Збірка логіки АБО-ТА
OAI	N, L	Вх.1, вх2, ... вих.	Збірка логіки АБО-ТА-НІ
AOI	N, L	Вх.1, вх2, ... вих.	Збірка логіки ТА-АБО-НІ

N – кількість входів, L – кількість вентилів

Модель динаміки вентилів має формат  
 .MODEL <ім'я моделі> UGATE [(параметри)]



Параметри моделей вентилів типу UGATE наведені в табл. 2.12 (значення за замовчуванням - 0, одиниця виміру - с).

Таблиця 2.12

### Параметри моделей стандартних вентилів

Ідентифікатор	Параметр
TPLHMN	Затримка при переході від низького рівня до високого, мінімальне значення
TPLHTY	Те ж, типове значення
TPLHMX	Те ж, максимальне значення
TRPLMN	Затримка при переході від високого рівня до низького, мінімальне значення
TRPLTY	Те ж, типове значення
TRPLMX	Те ж, максимальне значення

Вентилі із трьома станами наведені нижче (табл. 2.12)

Таблиця 2.12 Вентилі із трьома станами

Тип	Параметр	Порядок слідування виводів	Функціонал. призначення
BUF3	немає	Вх., дозв., вих.	Буфер
INV3	немає	Вх., дозв., вих.	Інвертор
AND3	N	Вх.1, вх2, ... дозв., вих.	Логічне ТА
NAND3	N	Вх.1, вх2, ... дозв., вих.	Логічне ТА-НІ
OR3	N	Вх.1, вх2, ... дозв., вих.	Логічне АБО
NOR3	N	Вх.1, вх2, ... дозв., вих.	Логічне АБО-НІ
XOR3	немає	Вх.1, вх2, дозв., вих.	Виключне АБО
NXOR3	немає	Вх.1, вх2, дозв., вих.	Виключне АБО-НІ
BUF3A	L	Вх.1, вх2, ... дозв., вих.1, вих.2...	Збірка буферів
INV3A	L	Вх.1, вх2, ... дозв., вих.1, вих.2...	Збірка інверторів
AND3A	N, L	Вх.1, вх2, ... дозв., вих.1, вих.2...	Збірка логіки ТА
NAND3A	N, L	Вх.1, вх2, ... дозв., вих.1, вих.2...	Збірка логіки ТА-НІ
OR3A	N, L	Вх.1, вх2, ... дозв., вих.1, вих.2...	Збірка логіки АБО
NOR3A	N, L	Вх.1, вх2, ... дозв., вих.1, вих.2...	Збірка логіки АБО-НІ

		вих.2...	
XOR3A	L	Вх.1, вх2, ... дозв., вих.1, вих.2...	Збірка логіки виключне АБО
NXOR3A	L	Вх.1, вх2, ... дозв., вих.1, вих.2...	Збірка логіки виключне АБО-НІ
N – кількість входів, L – кількість вентилів			

Модель динаміки цих вентилів має формат

.MODEL <ім'я моделі> UTGATE <параметри>]

Параметри моделей вентилів типу UTGATE наведені в табл.2.13(значення за замовчуванням - 0, одиниця виміру - с).

Таблиця 2.13. Параметри моделей вентилів із трьома станами

Ідентифікатор	Параметр
TPLHMN	Затримка при переході від низького рівня до високого, мінімальне значення
TPLHTY	Те ж, типове значення
TPLHMX	Те ж, максимальне значення
TRHLMN	Затримка при переході від високого рівня до низького, мінімальне значення
TRHLY	Те ж, типове значення
TRHLMX	Те ж, максимальне значення
TPLZMN	Затримка при переході від низького рівня до рівня Z, мінімальне значення
TPLZTY	Те ж, типове значення
TPLZMX	Те ж, максимальне значення
TRHZMN	Затримка при переході від високого рівня до рівня Z, мінімальне значення
TRHZTY	Те ж, типове значення
TRHZMX	Те ж, максимальне значення
TPZLMN	Затримка при переході від рівня Z до низького рівня, мінімальне значення
TPZLY	Те ж, типове значення
TPZLMX	Те ж, максимальне значення
TPZHMN	Затримка при переході від рівня Z до високого рівня до низького, мінімальне значення
TPZHLY	Те ж, типове значення
TPZHMX	Те ж, максимальне значення

**Двонаправлений вентиль** являє собою пасивний пристрій, що з'єднує або роз'єднує два вузли. Комутація цих вузлів визначається логічним станом вхідного керуючого вузла. Пристрої типу NBTG з'єднують два вузли при подачі на вхід логічної "1" і роз'єднують при подачі "0". Пристрої типу PBTG з'єднують два вузли при подачі на вхід логічного "0" і роз'єднують при подачі "1". Вони задаються по формату

Uxxx NBTG (або PBTG )

+ <+вузол джерела живлення> <-вузол джерела живлення>

+ <вузол затвора> <вузол каналу 1><вузол каналу 2>

+ <ім'я моделі динаміки> <ім'я моделі вхід/вихід>

+ [MNTYMXDLY=<вибір значення затримки>]

+ [IO\_LEVEL=<рівень моделі інтерфейсу >

Ці пристрої не мають параметрів, модель їх динаміки має вигляд:

.MODEL <ім'я моделі> UBTG

Параметри DRVH та DRVL моделі вхід/вихід використовують для керування комірками запам'ятовування заряду, сигнали з яких передають крізь двонаправлені вентилі. Якщо двонаправлений вентиль поєднаний з ланцюгом, до якого підключено хоча б один вхід пристрою, в моделі вхід/вихід якого є параметр INLD із значенням більш нуля, або вихід пристрою, в моделі якого параметр OUTLD більш нуля, то цей ланцюг буде промодельований як комірка запам'ятовування заряду.

Двонаправлені вентилі мають стандартний формат запису.

Наведемо приклад:

U4 NBTG \$G\_DRWR \$\_DGND GATE SD1 SD2 BTG1 10\_BTG

.MODEL BTG1 VBTG

Рисунок 2.28 – Тригери: а - JK-Тригер з негативним фронтом спрацьовування й низьким рівнем сигналів установки й скидання; б - D-Тригер з позитивним фронтом спрацьовування й низьким рівнем сигналів установки й скидання; в - синхронний двотактний RS-Тригер; г- синхронний одноктактний D-Тригер

## Тригери

Тригери мають динамічне й потенційне керування. Кожний компонент може містити один або кілька тригерів у корпусі, у яких загальними є сигнали установки, скидання й тактової синхронізації (рис. 2.28).

Перелік тригерів наведений у табл. 2.14

Таблиця 2.14. Перелік тригерів

Тип	Параметр	Порядок слідування виводів	Функціонал. призначення
Тригери з динамічним керуванням			
JKFF	кількість тригерів	S, R, C, J1, J2, ..., K1, K2, ..., Q1, Q2, ..., Q1, Q2	JK-тригер з негативним фронтом спрацьовування і низьким рівнем сигналу встановлення і скидання
DFF	кількість тригерів	S, R, C, D1, D2, Q1, Q2, ..., Q1, Q2	D-тригер з позитивним фронтом спрацьовування і низьким рівнем сигналу встановлення і скидання
Тригери з потенціальним керуванням			
SRFF	кількість тригерів	S, R, G, S1, S2, ..., R1, R2, ..., Q1, Q2, ..., Q1, Q2	Двохтактний синхронний RS-тригер
DLTCH	кількість тригерів	S, R, G, D1, D2, Q1, Q2, ..., Q1, Q2	Однотактний синхронний D - тригер

Моделі динаміки тригерів з динамічним керуванням мають формат

.MODEL <ім'я моделі> UEFF [(параметри)]

Параметри моделі тригерів з динамічним управлінням типу UEFF наведені в табл. 2.15 (значення за замовчуванням-0, одиниця виміру-з). Коса риса / означає "або"; наприклад, запис S/R означає сигнал S або R.

Таблиця 2.15

**Параметри моделей тригерів з динамічним керуванням**

Ідентифікатор	Параметр
TPPCQLHMN	Затримка переходу "0"- "1" із входу S/R до виходів Q/Q', мінімальне значення
TPPCQLHTY	Те ж, типове значення
TPPCQLHMX	Те ж, максимальне значення
TPPCQHLMN	Затримка переходу "1"- "0" із входу S/R до виходів Q/Q', мінімальне значення
TPPCQHLY	Те ж, типове значення
TPPCQHLMX	Те ж, максимальне значення
TWPCLMN	Максимальна тривалість сигналу "0" на вході S/R, мінімальне значення
TWPCLTY	Те ж, типове значення
TWPCLMX	Те ж, максимальне значення
TPCLKQLHMN	Затримка переходу "0"- "1" від фронту імпульсу C/C' до виходу Q/Q', мінімальне значення
TPCLKQLHTY	Те ж, типове значення
TPCLKQLHMX	Те ж, максимальне значення
TPCLKQHLMN	Затримка переходу "1"- "0" від фронту імпульсу C/C' до виходу Q/Q', мінімальне значення
TPCLKQHLY	Те ж, типове значення
TPCLKQHLMX	Те ж, максимальне значення
TWCLKLMN	Мінімальна тривалість сигналу "0" на вході C/C', мінімальне значення
TWCLKLY	Те ж, типове значення
TWCLKLMX	Те ж, максимальне значення
TWCLKHMN	Мінімальна тривалість сигналу "1" на вході C/C', мінімальне значення
TWCLKHTY	Те ж, типове значення
TWCLKHMX	Те ж, максимальне значення

TSUDCLKMN	Час підготовки до роботи із входів J/K/D перед дією фронту синхроімпульсу С, мінімальне значення
TSUDCLKTY	Те ж, типове значення
TSUDCLKMX	Те ж, максимальне значення
TSUPCCLKHMN	Тривалість сигналу "1" на входах S/R при дії фронту синхроімпульсу С/С', мінімальне значення
TSUPCCLKHTY	Те ж, типове значення
TSUPCCLKHMX	Те ж, максимальне значення
THDCLKMN	Тривалість сигналу на вході J/K/D після дії фронту синхроімпульсу С/С', мінімальне значення
THDCLKTY	Те ж, типове значення
THDCLKMX	Те ж, максимальне значення

Модель динаміки тригерів з потенційним керуванням має формат .MODEL <ім'я моделі> UGFF [(параметри)]

Параметри моделей тригерів з потенційним управлінням типу UGFF наведені в табл. 2.16 (значення за замовчуванням - 0, одиниця виміру - с).

Таблиця 2.16

### Параметри тригерів з потенційним керуванням

Ідентифікатор	Параметр
TRPCQLHMN	Затримка переходу "0" - "1" із входу S/R до виходів Q/Q', мінімальне значення
TRPCQLHTY	Те ж, типове значення
TRPCQLHMX	Те ж, максимальне значення
TRPCQHLMN	Затримка переходу "1" - "0" із входу S/R до виходів Q/Q', мінімальне значення
TRPCQHLTY	Те ж, типове значення
TRPCQHLMX	Те ж, максимальне значення
TWPCLMN	Мінімальна тривалість сигналу "0" на вході S/R, мінімальне значення
TWPCLTY	Те ж, типове значення
TWPCLMX	Те ж, максимальне значення
TPGQLHMN	Затримка переходу "0" - "1" від фронту імпульсу синхронізації G до виходу Q/Q', мінімальне значення
TPGQLHTY	Те ж, типове значення
TPGQLHMX	Те ж, максимальне значення

TPGQHLMN	Затримка переходу "1"- "0" від фронту імпульсу синхронізації G до виходу Q/Q', мінімальне значення
TPGQHLY	Те ж, типове значення
TPGQHLMX	Те ж, максимальне значення
TPDQLHMN	Затримка переходу "0"- "1" від входу S/R/D до виходу Q/Q'
TPDQLHY	Те ж, типове значення
TPDQLHMX	Те ж, максимальне значення
TPDQHLMN	Затримка переходу "1" - "0" від фронту імпульсу синхронізації G до виходу Q/Q', мінімальне значення
TPDQHLY	Те ж, типове значення
TPDQHLMX	Те ж, максимальне значення
TWGHMN	Мінімальна тривалість сигналу "1" на вході G, мінімальне значення
TWGHLY	Те ж, типове значення
TWGHMX	Те ж, максимальне значення
TSUDGMN	Час підготовки до роботи із входів S/R/D перед дією фронту синхроімпульсу G, мінімальне значення
TSUDGTY	Те ж, типове значення
TSUDGMX	Те ж, максимальне значення
TSUPCGHMN	Тривалість сигналу "1" на входах S/R при дії фронту синхроімпульсу G, мінімальне значення
TSUPCGHTY	Те ж, типове значення
TSUPCGHMX	Те ж, максимальне значення
THDGMN	Тривалість сигналу на вході S/R/D після дії фронту синхроімпульсу G, мінімальне значення
THDGTY	Те ж, типове значення
THDGMX	Те ж, максимальне значення

За замовчуванням у початковий момент часу вихідні стани тригерів прийняті невизначеними (стани X). Вони залишаються такими до подачі сигналів установки або скидання або переходу тригера в певний стан. Певний початковий стан тригерів встановлюється за допомогою параметра DIGINITSTATE директиви .OPTIONS.

У моделях тригерів є параметри, що характеризують мінімальні тривалості сигналів установки й скидання й мінімальну тривалість

імпульсів. Якщо ці параметри більше нуля, то в процесі моделювання вимірювані значення тривалості імпульсів дорівнюють заданим даним й при наявності занадто коротких імпульсів на екран видаються попереджувачі повідомлення (Warning messages), які також передаються в програму Probe і заносяться у вихідний файл із розширенням \*.out.

## **2.6 Програмовані логічні інтегральні схеми**

### **2.6.1 Відомості про програмовані логічні інтегральні схеми**

**ПЛІС (ПРОГРАМУВАЛЬНІ ЛОГІЧНІ ІНТЕГРАЛЬНІ СХЕМИ) АБО, FPGA (FIELD PROGRAMMABLE GATE ARRAYS), ЯВЛЯЮТЬ СОБОЮ ЦИФРОВІ ІНТЕГРАЛЬНІ МІКРОСХЕМИ ЯКІ СКЛАДЕНІ ІЗ ПРОГРАМУВАЛЬНИХ ЛОГІЧНИХ БЛОКІВ І ПРОГРАМУВАЛЬНИХ З'ЄДНАНЬ МІЖ ЦИМИ БЛОКАМИ. МОЖЛИВІСТЬ КОНФІГУРУВАТИ ЦІ ПРИСТРОЇ ДОЗВОЛЯЄ ІНЖЕНЕРАМ-РОЗРОБЛЮВАЧАМ ВИРІШУВАТИ БЕЗЛІЧ РІЗНИХ ЗАВДАНЬ [23].**

**ПЕРШІ ПЛІС З'ЯВИЛИСЯ В СЕРЕДИНІ 80-Х РОКІВ. У ТОЙ ЧАС ВОНИ ВИКОРИСТОВУВАЛИСЯ ПЕРЕВАЖНО ДЛЯ СТВОРЕННЯ СПОЛУЧНОЇ ЛОГІКИ, ДЛЯ РЕАЛІЗАЦІЇ КІНЦЕВИХ АВТОМАТІВ СЕРЕДНЬОЇ СКЛАДНОСТІ Й ДЛЯ РІШЕННЯ ДЕЯКИХ ЗАВДАНЬ ОБРОБКИ ДАНИХ. У МІРУ УСКЛАДНЕННЯ Й ЗБІЛЬШЕННЯ РОЗМІРІВ ПЛІС ПОЧИНАЮТЬ КОРИСТУВАТИСЯ БІЛЬШИМ ПОПИТОМ. НА ПОЧАТКУ 90-Х МИНУЛОГО СТОЛІТТЯ НАЙБІЛЬШИЙ ОБСЯГ ПРОДАЖІВ ВІДЗНАЧАВСЯ В ОБЛАСТІ МЕРЕЖ І ТЕЛЕКОМУНІКАЦІЙ, У ЯКИХ ПЕРЕДБАЧАЛИСЯ ОБРОБКА Й ПЕРЕДАЧА БІЛЬШИХ ПОТОКІВ ІНФОРМАЦІЇ. ДО КІНЦЯ 90-Х ПОПИТ НА ПЛІС РІЗКО ЗРІС У СПОЖИВЧІЙ, АВТОМОБІЛЬНІЙ І ВИРОБНИЧІЙ СФЕРАХ.**



**ПЛІС, ЯК ПРАВИЛО, ВИКОРИСТОВУВАЛИСЯ ДЛЯ СТВОРЕННЯ ПРОТОТИПІВ ЗАМОВЛЕНИХ МІКРОСХЕМ АБО ДЛЯ СТВОРЕННЯ ІСПИТОВИХ СТЕНДІВ, НА ЯКИХ ПЕРЕВІРЯЄТЬСЯ ФІЗИЧНА РЕАЛІЗОВАНІСТЬ НОВИХ АЛГОРИТМІВ. ОДНАК ЗАВДЯКИ НИЗЬКИМ ВИТРАТАМ НА ВИРОБНИЦТВО Й МАЛІ СТРОКИ ВИХОДУ НА РИНОК ЦІ МІКРОСХЕМИ ВСЕ ЧАСТІШЕ ВИКОРИСТОВУЮТЬ ЯК КІНЦЕВИЙ ПРОДУКТ. У ДЕЯКИХ ВЕЛИКИХ ПОСТАЧАЛЬНИКІВ ПЛІС Є ПРИСТРОЇ, ЯКІ СТАНОВЛЯТЬ ПРЯМУ КОНКУРЕНЦІЮ ЗАМОВЛЕНИМ МІКРОСХЕМАМ. ЗАЛЕЖНО ВІД СПОСОБУ ВИГОТОВЛЕННЯ ПЛІС МОЖУТЬ ПРОГРАМУВАТИСЯ АБО ОДИН РАЗ, АБО БАГАТОРАЗОВО. ПРИСТРОЇ, ЯКІ МОЖУТЬ ПРОГРАМУВАТИСЯ ТІЛЬКИ ОДИН РАЗ, НАЗИВАЮТЬСЯ ОДНОКРАТНО ПРОГРАМУВАЛЬНИМИ. СЛОВОСПОЛУЧЕННЯ «FIELD PROGRAMMABLE», ЩО МІСТИТЬСЯ В РОЗШИФРОВЦІ АБРЕВІАТУРИ FPGA, ОЗНАЧАЄ, ЩО ПРОГРАМУВАННЯ FPGA-ПРИСТРОЇВ ВИКОНУЄТЬСЯ НА МІСЦІ, «У ПОЛЬОВИХ УМОВАХ» (НА ВІДМІНУ ВІД ПРИСТРОЇВ, ВНУТРІШНЯ ФУНКЦІОНАЛЬНІСТЬ ЯКИХ ЖОРСТКО ПРОПИСАНА ВИРОБНИКОМ). ВОНО МОЖЕ ТАКОЖ ОЗНАЧАТИ, ЩО FPGA-ПРИСТРОЇ (ПЛІС) КОНФІГУРУЮТЬСЯ В ЛАБОРАТОРНИХ УМОВАХ, АБО СВІДЧИТИ ПРО ТЕ, ЩО МОВА ЙДЕ ПРО МОЖЛИВІСТЬ МОДИФІКАЦІЇ ФУНКЦІЙ ПРИСТРОЮ, ВБУДОВАНОГО В ЕЛЕКТРОННУ СИСТЕМУ, ЩО УЖЕ ЯКОСЬ ВИКОРИСТОВУЄТЬСЯ. ЯКЩО ПРИСТРІЙ МОЖЕ БУТИ ЗАПРОГРАМОВАНО, ЗАЛИШАЮЧИСЬ У СКЛАДІ СИСТЕМИ БІЛЬШЕ ВИСОКОГО РІВНЯ, ЙОГО НАЗИВАЮТЬСЯ ВНУТРІСИСТЕМНО ПРОГРАМУВАЛЬНИМ. ІСНУЄ БЕЗЛІЧ РІЗНИХ ТИПІВ ЦИФРОВИХ МІКРОСХЕМ, У ТОМУ ЧИСЛІ Й ТАКІ ЯК «РОЗСИПНА ЛОГІКА (НЕВЕЛИКІ КОМПОНЕНТИ, ЩО МІСТЯТЬ КІЛЬКА ПРОСТИХ ФІКСОВАНИХ ЛОГІЧНИХ ФУНКЦІЙ), ПРИСТРОЇ ПАМ'ЯТІ Й МІКРОПРОЦЕСОРИ. У ЦЬОМУ ВИПАДКУ ІНТЕРЕС ПРЕДСТАВЛЯЮТЬ ПРОГРАМУВАЛЬНІ ЛОГІЧНІ ПРИСТРОЇ (ПЛП), СПЕЦІАЛІЗОВАНІ ЗАМОВЛЕНІ ІНТЕГРАЛЬНІ МІКРОСХЕМИ (ASIC — APPLICATION**

**SPECIFIC INTEGRATED CIRCUIT, СПЕЦІАЛІЗОВАНА ІНТЕГРАЛЬНА СХЕМА Ї ASSP— APPLICATION SPECIFIC STANDARD PARTS, СПЕЦІАЛІЗОВАНА СТАНДАРТНА СХЕМА <sup>1)</sup> І, ПРИРОДНО, ПЛІС. ПРИЧОМУ ТЕРМІН ПЛП ПОЄДНУЄ ДВА ТИПИ ПРИСТРОЇВ: ПРОСТІ ПРОГРАМУВАЛЬНІ ЛОГІЧНІ ПРИСТРОЇ (ПРОСТІ ПЛП) І СКЛАДНІ ПРОГРАМУВАЛЬНІ ЛОГІЧНІ ПРИСТРОЇ (СКЛАДНІ ПЛП)<sup>2)</sup>.**

**ВНУТРІШНЯ АРХІТЕКТУРА ПЛП ВИЗНАЧЕНА ВИРОБНИКОМ, ТАКИМ ЧИНОМ, ЩО ВОНИ МОЖУТЬ БУТИ СКОНФІГУРОВАНІ (ПЕРЕПРОГРАМОВАНІ) «НА МІСЦІ» ДЛЯ ВИКОНАННЯ ВСІЛЯКИХ ФУНКЦІЙ. НА ВІДМІНУ ВІД ПЛІС ЦІ ПРИСТРОЇ МІСТЯТЬ МЕНШУ КІЛЬКІСТЬ ВЕНТИЛІВ І ВИКОРИСТОВУЮТЬСЯ ДЛЯ РІШЕННЯ НЕВЕЛИКИХ І ДОСИТЬ ПРОСТИХ ЗАВДАНЬ.**

**РАЗОМ З ТИМ, ІСНУЮТЬ ЗАМОВЛЕНІ ІНТЕГРАЛЬНІ СХЕМИ ASIC І ASSP, ЯКІ МІСТЯТЬ СОТНІ МІЛЬЙОНІВ ЛОГІЧНИХ ВЕНТИЛІВ І МОЖУТЬ ВИКОНУВАТИ НЕЙМОВІРНО БІЛЬШІ Ї СКЛАДНІ ФУНКЦІЙ. В ОСНОВІ ASIC І ASSP ЛЕЖАТЬ ТІ САМІ КОНСТРУКТОРСЬКІ РІШЕННЯ, І В НИХ ТА САМА ТЕХНОЛОГІЯ ВИРОБНИЦТВА. ОБИДВА ТИПИ ПРИСТРОЇ РОЗРОБЛЯЮТЬСЯ ДЛЯ ВИКОРИСТАННЯ В СКЛАДІ СПЕЦІАЛЬНИХ ДОДАТКІВ, АЛЕ ПРИ ЦЬОМУ ASIC РОЗРОБЛЯЮТЬСЯ Ї ВИРОБЛЯЮТЬСЯ ЗА ЗАМОВЛЕННЯМ СПЕЦІАЛІЗОВАНИХ КОМПАНІЙ, А ASSP ПРИЗНАЧАЮТЬСЯ МАСОВОМУ КОРИСТУВАЧЕВІ. НЕЗВАЖАЮЧИ НА ТЕ, ЩО ПРОПОНОВАНІ КОРИСТУВАЧЕВІ ЗАМОВЛЕНІ МІКРОСХЕМИ ВІДРІЗНЯЮТЬСЯ ВИСОКИМ СТУПЕНЕМ ІНТЕГРАЦІЇ, РІВНЕМ СКЛАДНОСТІ РОЗВ'ЯЗУВАНИХ ЗАВДАНЬ І ПРОДУКТИВНІСТЮ, ЇХНЯ РОЗРОБКА Ї ВИРОБНИЦТВО ДОВОЛІ ТРИВАЛИЙ І ДОРОГИЙ ПРОЦЕС. ДО ТОГО Ж, ОСТАТОЧНИЙ ВАРІАНТ СХЕМИ «ЗАМОРОЖУЄТЬСЯ В КРЕМНІЇ», І ДЛЯ ЇЇ МОДИФІКАЦІЇ ПОТРІБНЕ СТВОРЕННЯ НОВОЇ ВЕРСІЇ [23].**

**ТАКИМ ЧИНОМ, ПЛІС ЗАЙМАЮТЬ ПРОМІЖНЕ ПОЛОЖЕННЯ МІЖ ПЛП Ї ЗАМОВЛЕНИМИ**

**ІНТЕГРАЛЬНИМИ СХЕМАМИ. З ОДНОГО БОКУ, ЇХНЯ ФУНКЦІОНАЛЬНІСТЬ МОЖЕ БУТИ ЗАДАНА БЕЗПОСЕРЕДНЬО НА МІСЦІ ВІДПОВІДНО ДО ВИМОГ ЗАМОВНИКА-КОРИСТУВАЧА. З ІНШОГО БОКУ, ВОНИ МОЖУТЬ МІСТИТИ МІЛЬЙОНИ ЛОГІЧНИХ ВЕНТИЛІВ І, ОТЖЕ, РЕАЛІЗОВУВАТИ НАДЗВИЧАЙНО БІЛЬШІ Й СКЛАДНІ ФУНКЦІЇ, ЯКІ СПОКОНВІЧНО МОГЛИ БУТИ РЕАЛІЗОВАНІ ТІЛЬКИ ЗА ДОПОМОГОЮ ЗАМОВЛЕНИХ ІНТЕГРАЛЬНИХ СХЕМ.**

**ЩО СТОСУЄТЬСЯ ВАРТОСТІ ПЛІС, ТО ВОНА НАБАГАТО НИЖЧЕ ВАРТОСТІ ЗАМОВЛЕНИХ ІНТЕГРАЛЬНИХ СХЕМ (ХОЧА ОСТАТОЧНА ВЕРСІЯ ЗАМОВЛЕНОЇ МІКРОСХЕМИ ПРИ МАСОВОМУ ВИРОБНИЦТВІ ВИЯВЛЯЄТЬСЯ БІЛЬШЕ ДЕШЕВОЮ). ДО ТОГО Ж, У ВИПАДКУ ВИКОРИСТАННЯ ПЛІС ВНЕСЕННЯ ЗМІН У ПРИСТРІЙ НЕ ВИКЛИКАЄ ОСОБЛИВИХ УТРУДНЕНЬ І ІСТОТНО СКОРОЧУЮТЬСЯ СТРОКИ ВИХОДУ ТАКИХ ПРИСТРОЇВ НА РИНОК. ВСЕ ЦЕ РОБИТЬ ПЛІС ПРИВАБЛИВИМИ НЕ ТІЛЬКИ ДЛЯ ВЕЛИКИХ РОЗРОБЛЮВАЧІВ, АЛЕ Й ДЛЯ НЕВЕЛИКИХ НОВАТОРСЬКИХ КОНСТРУКТОРСЬКИХ БЮРО, ЯКІ ЗАВДЯКИ ЗДАТНОСТІ ПЛІС СТВОРИТИ «РАЙ У КУРЕНІ» ЗАЛИШАЮТЬСЯ ЖИТТЄЗДАТНИМИ. ІНАКШЕ КАЖУЧИ, «АПАРАТНІ» АБО «ПРОГРАМНІ» ІДЕЇ ОКРЕМИХ ІНЖЕНЕРІВ АБО НЕВЕЛИКИХ ГРУП ІНЖЕНЕРІВ МОЖУТЬ БУТИ РЕАЛІЗОВАНІ У ВИГЛЯДІ ІСПИТОВИХ СТЕНДІВ НА ОСНОВІ ПЛІС БЕЗ БІЛЬШИХ ОДНОРАЗОВИХ ВИТРАТ НА ПРОЕКТУВАННЯ АБО ЗАКУПІВЛЮ ДОРОГОГО ОСНАЩЕННЯ, НЕОБХІДНОЇ ДЛЯ РОЗРОБКИ ЗАМОВЛЕНИХ МІКРОСХЕМ. САМЕ ЦИМ ПОЯСНЮЄТЬСЯ ТОЙ ФАКТ, ЩО В 2003 РОЦІ, БУЛО РОЗПОЧАТО МАЙЖЕ 450000 РОЗРОБОК, ЩО ПЕРЕДБАЧАЮТЬ ВИКОРИСТАННЯ ПЛІС, УСЬОГО 5000 РОЗРОБОК З ВИКОРИСТАННЯМ ЗАМОВЛЕНИХ МІКРОСХЕМ ASSP І ТІЛЬКИ ВІД 1500 ДО 4000 РОЗРОБОК З ВИКОРИСТАННЯМ ЗАМОВЛЕНИХ МІКРОСХЕМ ASIC, ПРИЧОМУ ЦІ ЦИФРИ СТІМКО ПАДАЮТЬ РІК У РІК.**

ДО ПОЧАТКУ ДРУГОГО ТИСЯЧОРІЧЧЯ З'ЯВИЛИСЯ ВИСОКОПРОДУКТИВНІ ПЛІС, ЯКІ МІСТЯТЬ МІЛЬЙОНИ ВЕНТИЛІВ. ДЕЯКІ З НИХ МІСТЯТЬ ВБУДОВАНІ МІКРОПРОЦЕСОРНІ ЯДРА, ВИСОКОШВИДКІСНІ ІНТЕРФЕЙСИ ВВОДУ/ВИВОДУ Й ІНШІ ПРИСТРОЇ. СУЧАСНІ ПЛІС ЗНАХОДЯТЬ ЗАСТОСУВАННЯ ПРАКТИЧНО В БУДЬ-ЯКІЙ СФЕРІ, ВКЛЮЧАЮЧИ ПРИСТРОЇ ЗВ'ЯЗКУ Й ПРОГРАМУВАЛЬНІ РАДІОСТАНЦІЇ. ПЛІС ЗАСТОСОВУЮТЬ У РАДІОЛОКАЦІЇ, ОБРОБЦІ ЗОБРАЖЕНЬ І В ІНШИХ ДОДАТКАХ ЦИФРОВОЇ ОБРОБКИ СИГНАЛІВ (ЦОС). ПЛІС ВИКОРИСТОВУЮТЬ УСЮДИ, У ТОМУ ЧИСЛІ Й В ОДНОКРИСТАЛЬНИХ СИСТЕМАХ УТРИМУЮЧІ ПРОГРАМНІ Й АПАРАТНІ МОДУЛІ. У ЦЕЙ ЧАС ПЛІС ЗАПОВНЮЮТЬ ЧОТИРИ ВЕЛИКИХ СЕГМЕНТИ РИНКУ: ЗАМОВЛЕНІ ІНТЕГРАЛЬНІ СХЕМИ, ЦИФРОВА ОБРОБКА СИГНАЛІВ, СИСТЕМИ НА ОСНОВІ МІКРОКОНТРОЛЕРІВ, ЩО ВБУДОВУЮТЬСЯ, І МІКРОСХЕМИ, ЩО ЗАБЕЗПЕЧУЮТЬ ФІЗИЧНИЙ РІВЕНЬ ПЕРЕДАЧІ ДАНИХ. КРІМ ТОГО, З ПОЯВОЮ ПЛІС ВИНИК НОВИЙ СЕКТОР РИНКУ — СИСТЕМИ З ПЕРЕБУДОВУВАНОЮ АРХІТЕКТУРОЮ, ЩО, АБО RECONFIGURABLE COMPUTING (RC) [23]. ОТЖЕ МОЖНА ВИДІЛИТИ:

- ЗАМОВЛЕНІ ІНТЕГРАЛЬНІ СХЕМИ. ЯК МИ ВЖЕ ВІДЗНАЧАЛИ, СУЧАСНІ ПЛІС ВИКОРИСТОВУЮТЬСЯ ДЛЯ СТВОРЕННЯ ПРИСТРОЇВ ТАКОГО РІВНЯ, ЩО ДО ЦЬОГО МОГЛИ ЗАБЕЗПЕЧИТИ ТІЛЬКИ ЗАМОВЛЕНІ МІКРОСХЕМИ.
- ЦИФРОВА ОБРОБКА СИГНАЛІВ. ВИСОКОШВИДКІСНА ЦИФРОВА ОБРОБКА СИГНАЛІВ ТРАДИЦІЙНО ВИРОБЛЯЛАСЯ ЗА ДОПОМОГОЮ СПЕЦІАЛЬНО РОЗРОБЛЕНИХ МІКРОПРОЦЕСОРІВ, НАЗИВАНИХ ЦИФРОВІ СИГНАЛЬНІ ПРОЦЕСОРИ (ЦСП) АБО DIGITAL SIGNAL PROCESSORS (DSP). ОДНАК СУЧАСНІ ПЛІС МІСТЯТЬ ВБУДОВАНІ ПОМНОЖУВАЧІ, СХЕМИ АРИФМЕТИЧНОГО ПЕРЕНОСУ Й ВЕЛИКИЙ ОБСЯГ ОПЕРАТИВНОЇ ПАМ'ЯТІ УСЕРЕДИНІ КРИСТАЛА.

**ВСЕ ЦЕ В СПОЛУЧЕННІ З ВИСОКИМ СТУПЕНЕМ ПАРАЛЕЛІЗМУ ПЛІС ЗАБЕЗПЕЧУЄ ПЕРЕВАГУ ПЛІС НАД НАЙШВИДШИМИ СИГНАЛЬНИМИ ПРОЦЕСОРАМИ В 500 І БІЛЬШЕ РАЗ.**

**– МІКРОКОНТРОЛЕРИ, ЩО ВБУДОВУЮТЬСЯ. НЕСКЛАДНІ ЗАВДАННЯ КЕРУВАННЯ ЗВИЧАЙНО ВИКОНУЮТЬСЯ ВБУДОВАНИМИ ПРОЦЕСОРАМИ СПЕЦІАЛЬНОГО ПРИЗНАЧЕННЯ, ЯКІ НАЗИВАЮТЬСЯ МІКРОКОНТРОЛЕРАМИ. ЦІ НЕДОРОГІ ПРИСТРОЇ МІСТЯТЬ УБУДОВАНУ ПРОГРАМУ, ПАМ'ЯТЬ КОМАНД, ТАЙМЕРИ, ІНТЕРФЕЙСИ УВЕДЕННЯ/ВИВОДУ, РОЗТАШОВАНІ ПОРУЧ ІЗ ЯДРОМ НА ОДНОМУ КРИСТАЛІ. ЦІНИ НА ПЛІС ПАДАЮТЬ, ДО ТОГО Ж, НАВІТЬ НАЙПРОСТІШІ З НИХ МОЖНА ВИКОРИСТОВУВАТИ ДЛЯ РЕАЛІЗАЦІЇ ПРОГРАМНОГО МІКРОПРОЦЕСОРНОГО ЯДРА З НЕОБХІДНИМИ ФУНКЦІЯМИ УВЕДЕННЯ/ВИВОДУ. У РЕЗУЛЬТАТІ ПЛІС СТАЮТЬ УСЕ БІЛЬШЕ ПРИВАБЛИВИМИ ПРИСТРОЯМИ ДЛЯ РЕАЛІЗАЦІЇ ФУНКЦІЙ МІКРОКОНТРОЛЕРІВ.**

**– ФІЗИЧНИЙ РІВЕНЬ ПЕРЕДАЧІ ДАНИХ. ПЛІС ВЖЕ ДАВНО ВИКОРИСТОВУЮТЬ ЯК СПОЛУЧНУ ЛОГІКУ, ЩО ВИКОНУЄ ФУНКЦІЮ ІНТЕРФЕЙСУ МІЖ МІКРОСХЕМАМИ, ЯКІ РЕАЛІЗУЮТЬ ФІЗИЧНИЙ РІВЕНЬ ПЕРЕДАЧІ ДАНИХ, І ВИЩИМИ РІВНЯМИ МЕРЕЖЕВИХ ПРОТОКОЛІВ. ТОЙ ФАКТ, ЩО СУЧАСНІ ПЛІС МОЖУТЬ МІСТИТИ БЕЗЛІЧ ВИСОКОШВИДКІСНИХ ПЕРЕДАВАЧІВ, ОЗНАЧАЄ, ЩО МЕРЕЖЕВІ Й КОМУНІКАЦІЙНІ ФУНКЦІЇ МОЖУТЬ БУТИ РЕАЛІЗОВАНІ В ОДНОМУ ПРИСТРОЇ.**

**– СИСТЕМИ З ПЕРЕБУДОВУВАНОЮ АРХІТЕКТУРОЮ. МОЖНА ВИКОРИСТОВУВАТИ «АПАРАТНЕ ПРИСКОРЕННЯ» ПРОГРАМНИХ АЛГОРИТМІВ, ҐРУНТУЮЧИСЬ НА ТАКИХ ВЛАСТИВОСТЯХ ПРОГРАМУВАЛЬНИХ ЛОГІЧНИХ ІНТЕГРАЛЬНИХ СХЕМ (ПЛІС), ЯК ПАРАЛЕЛІЗМ І**

**ПЕРЕНАСТРОЮВАНІСТЬ. У ЦЕЙ ЧАС РІЗНІ  
КОМПАНІЇ ЗАЙНЯТІ СТВОРЕННЯМ ВЕЛИЧЕЗНИХ  
ОБЧИСЛЮВАЛЬНИХ МАШИН, ЩО  
ПЕРЕНАСТРОЮЮТЬСЯ, НА ОСНОВІ ПЛІС. ТАКІ  
СИСТЕМИ МОЖУТЬ ВИКОРИСТОВУВАТИСЯ ДЛЯ  
ВИКОНАННЯ ШИРОКОГО СПЕКТРА ЗАВДАНЬ - ВІД  
МОДЕЛЮВАННЯ АПАРАТУРИ ДО  
КРИПТОГРАФІЧНОГО АНАЛІЗ.**

**2.6.2. Синтез програмованих логічних структур за допомогою програм автоматизації схеми технічного проектування**

Програмувальні логічні структури (ПЛС) мають ряд входів, які формують стовпці–матриці, і ряд виходів, що утворюють рядки. Кожний вихід (рядок) управляється одним логічним елементом. Сукупність керуючих сигналів становлять програму для ПЛС, яка визначає, які входи з'єднуються з логічними елементами. До складу примітивів ПЛС входять тільки однотипні вентиля (ТА, АБО, ТА–НІ й т.п.), тому реальні ІС ПЛС складаються з декількох примітивів у вигляді макромоделей [24].

Програма ПЛС вводиться в завдання на моделювання подвійно:

1) попередньо записавши її у файл у форматі JEDEC і вказавши його ім'я в описі примітива конкретної ПЛС;

2) включивши дані програми безпосередньо в опис ПЛС (використовуючи конструкцію DATA=...), що менш зручно.

Однак при використанні бібліотек, у яких ПЛС оформлені у вигляді макромоделей, користувачеві не потрібно розбиратися в деталях їхніх моделей–достить вказати ім'я моделі ПЛС, список вузлів включення й за допомогою опції TEXT вказати ім'я JEDEC–Файлу, що містить опис програми ПЛС, як показано на наступному прикладі:

**X1 IN1 IN2 IN3 IN4 IN5 IN6 IN7 IN8 IN9 IN10 IN11  
IN12 IN13 +IN14 OUT1 OUT2 OUT3 OUT4  
PAL14H4**

+ TEXT: JEDEC\_FILE = "myprog.jed"

Тут у схему включена ПЛС типу PAL14H4, програмувальна з файлу myprog. jed.

## Рисунок 2.29 – Програмувальна логічна структура

Примітиви ПЛС вводяться в завдання на моделювання за допомогою виразу (рис. 2.29):

```
Uxxx <тип ПЛС> (<кількість входів>, <кількість виходів>)  
+ <+вузол джерела живлення> <-вузол джерела живлення>  
+ <вхідний вузол>* <вихідний вузол>*  
+ <ім'я моделі динаміки> <ім'я моделі вхід/вихід>  
+ [FILE=<ім'я файлу>]  
+ [DATA=<прапор системи числення> $ <дані програми>$]  
+ [MNTYMXDLY= <вибір значення затримки>]  
+ [I_LEVEL=<рівень моделі інтерфейсу>]
```

Існують ПЛС наступних типів:

PLAND – матриця логіки І;

PLOR – матриця логіки АБО;

PLXOR – матриця логіки що виключає АБО;

PLNAND – матриця логіки Й–НЕ;

PLNOR – матриця логіки АБО–НЕ;

PLNXOR – матриця логіки що виключає АБО–НЕ;

PLANDC – матриця логіки І, що містить для кожного входу стовпці прямого й додаткового коду;

PLORC – матриця логіки АБО, що містить для кожного входу стовпці прямого й додаткового коду;

PLXORC – матриця логіки що виключає АБО, що містить для кожного входу стовпці прямого й додаткового коду;

PLNANDC – Матриця логіки І–НІ, що містить для кожного входу стовпці прямого й додаткового коду;

PLNORC – матриця логіки АБО–НЕ, що містить для кожного входу стовпці прямого й додаткового коду;

PLNXORC – матриця логіки що виключає АБО, що містить для кожного входу стовпці прямого й додаткового коду.

Після ключового слова FILE вказується ім'я файлу у форматі JEDEC, у якому записана програма ПЛС. Воно може бути зазначене як текстова константа (і тоді вказується в лапках""") або як текстове вираження (полягає між вертикальними рисками ||). Якщо наведено

опцію FILE, то будь-які дані, наведені після опції DATA, ігноруються. Спосіб адресації, прийнятий у файлі JEDEC для розташування даних, визначається параметрами моделі динаміки.

Прапор системи числення приймає значення:

В – двійкова система числення;

О – Восьмерична система числення (біт старшого розряду розташований по молодшій адресі);

Х – шостнадцятирічна шестнадцятирична система числення (біт старшого розряду розташований по молодшій адресі).

Дані програми містяться між знаками долара \$ і можуть розташовуватися як разом, так і розділятися одним або декількома пробілами. Потік даних починається з нульової адреси, по якому програмується з'єднання першого входу з вентилем, що управляє першим виходом. Наявність "0" означає, що вхід не приєднаний до вентиля, а "1" – наявність такого з'єднання (спочатку всі входи не приєднані ні до яких виходів). Дані по наступній адресі управляють з'єднанням доповнення першого входу до вентиля, що керує першим виходом (тільки для ПЛС із прямими й додатковими входами), або другого входу, підключеного до вентиля, що керує першим виходом. Кожна додаткова "1" або "0" програмує з'єднання наступного входу або його доповнення з вентилем, що управляє першим виходом, доти, поки не переберуть всі входи (і їхнього доповнення). Наступні дані програмують з'єднання входів із другим виходом і т.д.

Модель динаміки ПЛС має формат

.MODEL <ім'я моделі динаміки>

+UPLD [(параметри моделі динаміки)]

Параметри цієї моделі наведені в табл. 2.17.

Таблиця 2.17

### Параметри моделі динаміки ПЛС

Ідентифікатор	Параметр	Значення за змовчанням	Одиниця вимірювання
TPLHMN	Час затримки на вихід при перемиканні 0-1, мінімальне значення	0	с
TPLHTY	Те ж, типове значення	0	с
TPLHMX	Те ж, максимальне значення	0	с



TRPLMN	Час затримки на вихід при перемиканні 1-0, мінімальне значення	0	c
TRPLTY	Те ж, типове значення	0	c
TRPLMX	Те ж, максимальне значення	0	c
OFFSET	Адреса даних, керуючих підключенням першого входу до першого виходу (у файлі JEDEC)	0	
COMPOFFSET	Адреса даних, керуючих підключенням доповнення першого входу до першого виходу (у файлі JEDEC)	1	
INSCALE	Кількість адрес для програмування зміни стану кожного входу (у файлі JEDEC)	1	
OUTSCALE	Кількість адрес у файлі JEDEC для програмування зміни стану кожного виходу (вентиля)	2	

Наведемо приклад дешифратора 3x8 (рис. 2.30). Вхідні вузли позначимо IN1 (старший розряд), IN2, IN3 (молодший розряд). Якщо всі входи перебувають у стані "0", вихід OUT1= "1". Якщо IN1 і IN2 у стані "1", а IN3 – у стані "1", OUT2="1" і т.д. Дані програми для зручності читання записані у вигляді масиву. У коментарях зверху від програми зазначені імена вхідних вузлів, що перебувають у стані "1" – true (T) і "0" – false (F, додатковий код); у коментарях наприкінці рядків зазначені імена вихідних вузлів, керованих вентиляем.

Рисунок 2.30– Дешифратор 3x8, реалізований на ПЛС

## **2.7 Особливості автоматизованого проектування схем із багато розрядними цифро-аналоговими та аналого-цифровими перетворювачами**

Багаторозрядні АЦП задаються за форматом  
 Uxxx ADC(<m><список вузлів> <модель динаміки>  
 + <модель вхід/вихід>  
 + [MNTYMXDLU=<вибір значення затримки>]  
 + [IO\_LEVEL=<рівень моделі інтерфейсу>]

Імена вузлів перелічують в <списку вузлів> у наступному порядку (рис.):

<+вузол джерела живлення>, <-вузол джерела живлення>, <аналоговий вхід>, <опорна напруга>, <"земля">, <сигнал дозволу>, <сигнал перетворення>, <сигнал переповнення>, <m-й розряд>, ...< 1-й розряд>

**СЕНС ІНШИХ ПАРАМЕТРІВ ТАКИЙ ЖЕ, ЩО Й ДЛЯ ІНТЕРФЕЙСУ АЦ ТИПУ OXXX.**

Модель динаміки має формат

.MODEL <ім'я моделі> UADC [(параметри)]

Рисунок 2.23 –Аналого-цифровий перетворювач

**ПАРАМЕТРИ ЦІЄЇ МОДЕЛІ НАВЕДЕНІ В ТАБЛ. 2.9 (ЗНАЧЕННЯ ЗА ЗАМОВЧУВАННЯМ-0, ОДИНИЦЯ ВИМІРУ-С).**

Таблиця 2.9

**Параметри моделі динаміки ЦАП**

Ідентифікатор	Параметр
TPCSMN	Час затримки запуску - запізнювання між передніми фронтами імпульсів дозволу й запуску, мінімальне значення
TPCSTY	Те ж, типове значення
TPCSMX	Те ж, максимальне значення
TPSDMN	Час циклу кодування - інтервал часу між переднім фронтом імпульсу запуску й переходом вихідного сигналу в новий стан, мінімальне значення
TPSDTY	Те ж, типове значення
TPSDMX	Те ж, максимальне значення
TPDSMN	Запізнювання заднього фронту сигналу запуску відносно моменту переходу вихідного сигналу в новий стан, мінімальне значення

TPDSMTY	Те ж, типове значення
TPDSMX	Те ж, максимальне значення

**ЧАСОВА ДІАГРАМА АЦП ПОКАЗАНА НА РИС. 2.24  
ВИХІДНИЙ СИГНАЛ АЦП ДОРІВНЮЄ НАЙБЛИЖЧОМУ  
ЦІЛОМУ ВИРАЖЕННЯ**

$$\frac{V(< \text{аналоговий вихід} >, < \text{"земля"} >)}{V(< \text{опорна напруга} >, < \text{"земля"} >)} \cdot 2^m$$

**ДЕ М - КІЛЬКІСТЬ РОЗРЯДІВ. ЯКЩО ЦЕ ВИРАЖЕННЯ  
БІЛЬШЕ 2<sup>М</sup>-1, ВСІ РОЗРЯДИ ДАНИХ І РОЗРЯД  
ПЕРЕПОВНЕННЯ ПРИЙМУТЬ ЗНАЧЕННЯ 1.**

Якщо воно менше нуля, розряди даних приймуть нульове значення, а розряд переповнення - 1. Таким чином, опорна напруга встановлює діапазон вхідної напруги АЦП.

Рисунок 2.24 –Перехідні процеси в АЦП

Сигнал дозволу (convert pulse) може мати кожну, у тому числі й нульовій тривалості. Якщо цикл кодування  $t_{psd}=0$ , то  $m$  розрядів даних і розряд переповнення, не приймаючи невизначеного стану, відразу приймають нове значення. Між вузлами <опорна напруга> і <"земля"> включається резистор з опором, рівним  $1/GMIN$ .

Вибірki вхідних напруг виробляються по передньому фронті імпульсу дозволу, причому швидкість зміни вхідних напруг не впливає на результат перетворення: Приведемо приклад опису 4-розрядного АЦП:

```
U3 ADC(4) $G_DPWR G_DGND1 10 0 conv stat over out3
out2 + out1 out0
+ DINAM IO_ADC
.MODEL DINAM UADC(
+tpcsmn=5ns,          tpcsty=8ns,  tpcsmx=10ns
+tpsdmn=16ns,        tpsdty=20ns, tpsdmx=22ns
+tpdsmn=4ns,         tpdsty=5ns,  tpsdmx=6ns)
.MODEL IO_ADC UIO(drvh=50 drvl=50)
```

Багаторозрядні ЦАП задається по формату  
 Uxxx DAC(<m>) <список вузлів> <модель динаміки>  
 +<модель вхід/вихід>  
 + [MNTYMXDLY=<вибір значень затримки>]  
 + [IO\_LEVEL=<рівень моделі інтерфейсу>]  
 Імена вузлів перелічують в <списку вузлів> у наступному  
 порядку (рис. 2.25):  
 <+ВУЗОЛ ДЖЕРЕЛА ЖИВЛЕННЯ>, <-ВУЗОЛ ДЖЕРЕЛА  
 ЖИВЛЕННЯ>, <АНАЛОГОВИЙ ВИХІД>, <ОПОРНА  
 НАПРУГА> <ИЗЕМЛЯ">, <М-Й РОЗРЯД ВХІДНОГО  
 СИГНАЛУ> <1-Й РОЗРЯД ВХІДНОГО СИГНАЛУ>  
 Модель динаміки ЦАП має вигляд:  
 .MODEL <ім'я моделі> UDAC

Рисунок 2.25 – Цифроаналоговий перетворювач

Рисунок 2.26 – Перехідні процеси в ЦАП

Параметри цієї моделі наведені в табл. 2.10 (значення за замовчуванням - 0, одиниця виміру - с).

Таблиця 2.10  
 Параметри моделі динаміки АПЦ

Ідентифікатор	Параметр
TSWMN	Час установлення (від моменту зміни вхідного коду до моменту досягнення вихідною напругою рівня 0,9 сталого значення), мінімальне значення
TSWTY	Те ж, типове значення
TSWMX	Те ж, максимальне значення

Між вузлами <аналоговий вихід> і <"земля"> включається джерело напруги з нульовим внутрішнім опором, ЕДС якого дорівнює:

$$V(< опорна напруга >, <"земля">) \times \frac{\text{бінарний вхідний сигнал}}{2^m}$$

**ОПОРНА НАПРУГА ВИЗНАЧАЄ ДІАПАЗОН ВИХІДНОЇ  
АНАЛОГОВОЇ НАПРУГИ. МІЖ ВУЗЛОМ ДЖЕРЕЛА  
ОПОРНОЇ НАПРУГИ Й "ЗЕМЛЕЮ" ВКЛЮЧАЄТЬСЯ ОПІР,  
РІВНЕ 1/GMIN.**

Якщо який-небудь розряд вхідного цифрового сигналу не визначений, вихідна напруга дорівнює половині різниці двох напруг. Одне з них являє собою вихідна напруга ЦАП, якщо всі невизначені стани "X" замінити на "1", друге - якщо ці стани замінити на логічний "0". При зміні станів всіх розрядів вихідна напруга лінійно змінюється протягом інтервалу перетворення, як показано на рис. 2.26.

**2.8 Автоматизоване проектування схем із  
запам'ятовуваними та мікропроцесорними  
пристроями**

**ЗАПАМ'ЯТОВУВАЛЬНІ ПРИСТРОЇ (ЗУ)  
ПІДРОЗДІЛЯЮТЬСЯ НА ПОСТІЙНІ ЗУ (ROM, READ ONLY  
MEMORIES) І ОПЕРАТИВНІ ЗУ (RAM, RANDOM ACCESS  
READ-WRITE MEMORIES).**

**ПОСТІЙНІ ЗАПАМ'ЯТОВУВАЛЬНІ ПРИСТРОЇ (ПЗУ).  
ІСНУЄ ДВА СПОСОБИ ЗАПИСУ ДАНИХ У ПЗУ ДЛЯ  
НАСТУПНОГО МОДЕЛЮВАННЯ: 1) ЗВИЧАЙНО ДАНІ  
ПОПЕРЕДНЬО ЗАПИСУЮТЬСЯ У ФАЙЛ У ФОРМАТІ INTEL  
HEX І ПЕРЕД ПОЧАТКОМ МОДЕЛЮВАННЯ ВОНИ З НЬОГО  
ЗЧИТУЮТЬСЯ; 2) ДАНІ ЗАПИСУЮТЬСЯ БЕЗПОСЕРЕДНЬО  
В ОПИСІ ПЗУ ЗА ДОПОМОГОЮ КОНСТРУКЦІЇ DATA=...**

**МОДЕЛЬ ПЗУ (РИС. 2.31) ЗАДАЄТЬСЯ ПО ФОРМАТУ  
UXXX ROM (<КІЛЬКІСТЬ АДРЕСНИХ ВХОДІВ>, КІЛЬКІСТЬ  
ВИХОДІВ>)  
+ <+ВУЗОЛ ДЖЕРЕЛА ЖИВЛЕННЯ> <-ВУЗОЛ ДЖЕРЕЛА  
ЖИВЛЕННЯ>  
+ <ВХІД ДОЗВОЛУ ЧИТАННЯ> <СТАРШИЙ РОЗРЯД  
АДРЕСИ>...  
+ <МОЛОДШИЙ РОЗРЯД АДРЕСИ>  
+ <СТАРШИЙ РОЗРЯД ВИХОДУ> ... <МОЛОДШИЙ РОЗРЯД  
ВИХОДУ>**

+ <ІМ'Я МОДЕЛІ ДИНАМІКИ> <ІМ'Я МОДЕЛІ ВХІД/ВИХІД>  
+ [FILE=<ІМ'Я ФАЙЛУ>]  
+ [DATA=<ПРАПОР СИСТЕМИ ЧИСЛЕННЯ>\$<ДАНИ  
ПРОГРАМИ>\$]  
+ [MNTYMXDLY=<ВИБІР ЗНАЧЕННЯ ЗАТРИМКИ>]  
+ [IO\_LEVEL=<РІВЕНЬ МОДЕЛІ ІНТЕРФЕЙСУ>

РИСУНОК 2.31 – ПОСТІЙНИЙ ЗАПАМ'ЯТОВУВАЛЬНИЙ  
ПРИСТРІЙ

ПІСЛЯ КЛЮЧОВОГО СЛОВА FILE ВКАЗУЄТЬСЯ ІМ'Я  
ФАЙЛУ У ФОРМАТІ INTEL HEX, У ЯКОМУ ЗАПИСАНІ  
ДАНИ ПЗУ. ВОНО МОЖЕ БУТИ ЗАЗНАЧЕНЕ ЯК ТЕКСТОВА  
КОНСТАНТА (І ТОДІ ПОЛЯГАЄ В ЛАПКИ "'") АБО ЯК  
ТЕКСТОВЕ ВИРАЗ (ПОЛЯГАЄ МІЖ ВЕРТИКАЛЬНИМИ  
РИСКАМИ ||). ЯКЩО НАВЕДЕНО ОПЦІЮ FILE, ТО БУДЬ-  
ЯКІ ДАНИ, НАВЕДЕНІ ПІСЛЯ ОПЦІЇ DATA, ІГНОРУЮТЬСЯ.  
ПРАПОР СИСТЕМИ ЧИСЛЕННЯ ПРИЙМАЄ ЗНАЧЕННЯ:

**B** – ДВІЙКОВА СИСТЕМА ЧИСЛЕННЯ;

**O** – ВОСЬМЕРИЧНА СИСТЕМА ЧИСЛЕННЯ (БІТ  
СТАРШОГО РОЗРЯДУ РОЗТАШОВАНИЙ ПО МОЛОДШІЙ  
АДРЕСІ);

**X** – ШОСТНАДЦЯТИРІЧНА СИСТЕМА ЧИСЛЕННЯ (БІТ  
СТАРШОГО РОЗРЯДУ .РОЗТАШОВАНИЙ ПО МОЛОДШІЙ  
АДРЕСІ).

ДАНИ ПРОГРАМИ МІСТЯТЬСЯ МІЖ ЗНАКАМИ ДОЛАРА  
\$ І МОЖУТЬ РОЗТАШОВУВАТИСЯ ЯК РАЗОМ, ТАК І  
РОЗДІЛЯТИСЯ ОДНИМ АБО ДЕКІЛЬКОМА ПРОБІЛАМИ.  
ПОТІК ДАНИХ ПОЧИНАЄТЬСЯ З НУЛЬОВОЇ АДРЕСИ, ПО  
ЯКОМУ РОЗМІЩАЄТЬСЯ ПЕРШИЙ РОЗРЯД ДАНИХ.

НАСТУПНИЙ БІТ СТАВИТЬСЯ ДО ДРУГОГО РОЗРЯДУ  
ДАНИХ І ТАК ДОТИ ПОКИ НЕ БУДУТЬ ВИЗНАЧЕНІ СТАНИ  
ВСІХ РОЗРЯДІВ ПО ЦЬЙ АДРЕСІ. ПІСЛЯ ЦЬОГО  
ПЕРЕРАХОВУЮТЬСЯ ДАНИ ПО НАСТУПНІЙ АДРЕСІ Й Т.Д.  
МОДЕЛЬ ДИНАМІКИ ПЗУ МАЄ ФОРМАТ:

**.MODEL <ІМ'Я МОДЕЛІ ДИНАМІКИ> UROM [(ПАРАМЕТРИ  
МОДЕЛІ ДИНАМІКИ)]**  
**ПАРАМЕТРИ ЦІЄЇ МОДЕЛІ НАВЕДЕНІ В ТАБЛ. 2.18**  
**(ЗНАЧЕННЯ ЗА ЗАМОВЧУВАННЯМ – 0, ОДИНИЦЯ ВИМІРУ**  
**– С).**

*ТАБЛИЦЯ 2.18*

ПАРАМЕТРИ МОДЕЛІ ПЗУ

Ідентифікатор	Параметр
TPADHMN	Час вибірки адреси при перемиканні вхідних даних 0-1, мінімальне значення
TPADHTY	Те ж, типове значення
TPADHMX	Те ж, максимальне значення
TPADLMN	Час вибірки адреси при перемиканні вхідних даних 1-0, мінімальне значення
TPADLTY	Те ж, типове значення
TPADLMX	Те ж, максимальне значення
TPEDHMN	Час вибірки дозволу при перемиканні виходів Z-1, мінімальне значення
TPEDHTY	Те ж, типове значення
TPEDHMX	Те ж, максимальне значення
TPEDLMN	Час вибірки дозволу при перемиканні виходів Z -0, мінімальне значення
TPEDLTY	Те ж, типове значення
TPEDLMX	Те ж, максимальне значення
TPEDHZMN	Час вибірки дозволу при перемиканні виходів 1 -Z, мінімальне значення
TPEDHZTY	Те ж, типове значення
TPEDHZMX	Те ж, максимальне значення
TPEDLZMN	Час вибірки дозволу при перемиканні виходів 0-Z, мінімальне значення
TPEDLZTY	Те ж, типове значення
TPEDLZMX	Те ж, максимальне значення

**ПЕРЕХІДНІ ПРОЦЕСИ ПРИ ЗЧИТУВАННІ ІЗ ПЗУ**  
**ПОКАЗАНІ НА РИС. 2.32 НА ВХІД ДОЗВОЛУ ЧИТАННЯ**  
**НЕОБХІДНО ПОДАТИ "1", А СТАНУ ВУЗЛІВ ВИХІДНИХ**  
**ДАНИХ ЗМІНЮЮТЬСЯ ВІД СТАНУ ВИСОКОГО**  
**ІМПЕДАНСУ Z ДО ВІДПОВІДНОГО СТАНУ ЧЕРЕЗ ДЕЯКИЙ**

**ЧАС ТРЕ. ПРОТЯГОМ ЧАСУ, ПОКИ СИГНАЛ ДОЗВОЛУ  
ЧИТАННЯ ПЕРЕБУВАЄ В СТАНІ "1", СИГНАЛИ АДРЕСИ  
МОЖУТЬ ЗМІНЮВАТИСЯ, І ЯКЩО ЦЕ ТАК, ТО НОВІ ДАНІ  
ДОСТУПНІ НА ВИХОДАХ ЧЕРЕЗ ЯКИЙСЬ ЧАС ЗАТРИМКИ**

**РИСУНОК 2.32 – ПЕРЕХІДНІ ПРОЦЕСИ В ПЗУ**

**ПРИВЕДЕМО ПРИКЛАД ПЗУ 8X8:  
UMULTIPLY ROM (8,8); МОДЕЛЬ ПЗУ 256X8 РОЗРЯДІВ  
+ \$G\_DPWR \$G\_DGND ; ВУЗЛИ ДЖЕРЕЛА ЖИВЛЕННЯ Й  
"ЗЕМЛІ"  
+ ENABLE ; ВХІД ДОЗВОЛУ ЧИТАННЯ  
+ AIN3 AIN2 AIN1 AIN0 ; ПЕРШІ 4 БІТИ АДРЕСИ  
+ BIN3 BIN2 BIN1 BIN0 ; ДРУГІ 4 БІТИ АДРЕСИ  
+ OUT7 OUT6 OUT5 OUT4 OUT3 OUT2 OUT1 OUT0 ; ВИХОДИ  
+ ROM\_MDL ; ІМ'Я МОДЕЛІ ДИНАМІКИ  
+ IO\_STD ; ІМ'Я МОДЕЛІ ВХІД/ВИХІД  
+ DATA=X\$**

**ДАНІ У ДВІЙКОВОМУ КОДІ:**

* 0 1 2 3 4 5 6 7 8 9 A B C D E F	
+ 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ;	A=0
+ 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F ;	A=1
+ 00 02 04 06 08 0A 0C 0E 10 12 14 16 18 1A 1C 1E ;	A=2
+ 0003 06090C 0F 12 15 18 1B 1E 2124 27 2A 2D ;	A=3
+ 00 04 08 0C 10 1418 1C 20 24 28 2C 30 34 38 3C ;	A=4
+ 00 05 0A 0F 1419 1E 23 2820 32 37 3C 41 46 48 ;	A=5
+ 00 06 0C 12 18 1E 24 2A 30 36 3C 42 48 4E54SA ;	A=6
+ 00 07 0E 15 1C 23 2A 31 38 3F 46 4D 54 58 62 69 ;	A=7
+ 00 08 10 18 20 28 30 38 40 48 50 58 60 68 70 78 ;	A=8
+ 00 09 12 1B 24 2D 36 3F 48 51 5A 63 8C 75 7E 87 ;	A=9
+ 00 0A 14 1E 28 32 3C 48 50 5A 64 7E 78 82 8C 96 ;	A=A
+ 00 0B 16 21 2C 37 42 4D 58 63 6E 79 84aF 9A AS ;	A=B
+ 00 0C 18 24 30 3C 48 54 60 6C 78 84 90 9C A8 B4	A=C
+ 00 0D 1A 27 34 41 4E 5B 68 75 82 8F 9C A9 B6 C3 ;	A=D
+ 00 0E 1C 2A 38 46 54 6270 7E 8C 9A A8 B6 C4 D2 ;	A=E



+ 00 0F 1E 2D 3C 48 5A 69 78 87 96 A5 B4 C3 D1 E1\$; A=F  
.MODEL ROM\_MDL UROM (...)

**ОПЕРАТИВНІ ЗАПАМ'ЯТОВУВАЛЬНІ ПРИСТРОЇ (ОЗУ).  
ЗВИЧАЙНО В ПОЧАТКОВИЙ МОМЕНТ ЧАСУ В ОЗУ  
ВСТАНОВЛЮЮТЬСЯ ДОВІЛЬНІ ДАНІ ПО ВСІХ АДРЕСАХ.  
ІСНУЄ ДВА СПОСОБИ ЗАПИСУ ПОЧАТКОВИХ ДАНИХ В  
ОЗУ ПРИ МОДЕЛЮВАННІ:**

- 1) ДАНІ ПОПЕРЕДНЬО ЗАПИСУЮТЬСЯ У ФАЙЛ У  
ФОРМАТІ INTEL HEX І ПЕРЕД ПОЧАТКОМ  
МОДЕЛЮВАННЯ З НЬОГО ЗЧИТУЮТЬСЯ;**
  - 2) ДАНІ ЗАПИСУЮТЬСЯ БЕЗПОСЕРЕДНЬО В ОПИСІ ОЗУ  
ЗА ДОПОМОГОЮ КОНСТРУКЦІЇ DATA=...**
- МОДЕЛЬ ОЗУ (РИС. 2.33) ЗАДАЄТЬСЯ ПО ФОРМАТІ  
UXXX RAM (<КІЛЬКІСТЬ АДРЕСНИХ ВХОДІВ>, <КІЛЬКІСТЬ  
ВИХОДІВ>)**
- + <+ВУЗОЛ ДЖЕРЕЛА ЖИВЛЕННЯ> <-ВУЗОЛ ДЖЕРЕЛА  
ЖИВЛЕННЯ>**
  - + <ВХІД ДОЗВОЛУ ЧИТАННЯ> <ВХІД ДОЗВОЛУ ЗАПИСУ>**
  - + <СТАРШИЙ РОЗРЯД АДРЕСИ>... <МОЛОДШИЙ РОЗРЯД  
АДРЕСИ>**
    - + <СТАРШИЙ РОЗРЯД ВХОДУ ДАНИХ>...**
    - + <МОЛОДШИЙ РОЗРЯД ВХОДУ ДАНИХ>**
  - + <СТАРШИЙ РОЗРЯД ВИХОДУ> . . . МОЛОДШИЙ РОЗРЯД  
ВИХОДУ>**
  - + <ІМ'Я МОДЕЛІ ДИНАМІКИ> <ІМ'Я МОДЕЛІ ВХІД/ВИХІД>  
+ [FILE=<ІМ'Я ФАЙЛУ>]**
  - + [ DATA=<ПРАПОР СИСТЕМИ СЧИСЛЕННЯ >\$<ДАНОЇ  
ПРОГРАМИ>\$ ]**
  - + [ MNTYMXDLY=<ВИБІР ЗНАЧЕННЯ ЗАТРИМКИ>]**
  - + [ IO\_LEVEL=<РІВЕНЬ МОДЕЛІ ІНТЕРФЕЙСУ>**

**РИСУНОК 2.33 –ОПЕРАТИВНИЙ ЗАПАМ'ЯТОВУВАЛЬНИЙ  
ПРИСТРІЙ**

**ПІСЛЯ КЛЮЧОВОГО СЛОВА FILE ВКАЗУЄТЬСЯ ІМ'Я ФАЙЛУ У ФОРМАТІ INTEL HEX, У ЯКОМУ ЗАПИСАНІ ДАНІ ЗУ. ВОНО МОЖЕ БУТИ ЗАЗНАЧЕНЕ ЯК ТЕКСТОВА КОНСТАНТА (І ТОДІ ПОЛЯГАЄ В ЛАПКИ"." ) АБО ЯК ТЕКСТОВЕ ВИРАЗ (ПОЛЯГАЄ МІЖ ВЕРТИКАЛЬНИМИ РИСКАМИ || ). ЯКЩО НАВЕДЕНО ОПЦІЮ FILE, ТО БУДЬ-ЯКІ ДАНІ, НАВЕДЕНІ ПІСЛЯ ОПЦІЇ DATA, ІГНОРУЮТЬСЯ. ПРАПОР СИСТЕМИ ЧИСЛЕННЯ ПРИЙМАЄ ЗНАЧЕННЯ:**

**В – ДВІЙКОВА СИСТЕМА ЧИСЛЕННЯ;**

**О – ВОСЬМЕРИЧНА СИСТЕМА ЧИСЛЕННЯ (БІТ СТАРШОГО РОЗРЯДУ РОЗТАШОВАНИЙ ПО МОЛОДШІЙ АДРЕСІ);**

**Х – ШОСТНАДЦЯТИРІЧНА СИСТЕМА ЧИСЛЕННЯ (БІТ СТАРШОГО РОЗРЯДУ РОЗТАШОВАНИЙ ПО МОЛОДШІЙ АДРЕСІ).**

**ДАНІ ПРОГРАМИ МІСТЯТЬСЯ МІЖ ЗНАКАМИ ДОЛАРА \$ І МОЖУТЬ РОЗТАШОВУВАТИСЯ ЯК РАЗОМ, ТАК І РОЗДІЛЯТИСЯ ОДНИМ АБО ДЕКІЛЬКОМА ПРОБІЛАМИ. ПОТІК ДАНИХ ПОЧИНАЄТЬСЯ З НУЛЬОВОЇ АДРЕСИ, ПО ЯКОМУ РОЗМІЩАЄТЬСЯ ПЕРШИЙ РОЗРЯД ДАНИХ. НАСТУПНИЙ БІТ СТАВИТЬСЯ ДО ДРУГОГО РОЗРЯДУ ДАНИХ, І ТАК ДОТИ, ПОКИ НЕ БУДУТЬ ВИЗНАЧЕНІ СТАНИ ВСІХ РОЗРЯДІВ ПО ЦІЙ АДРЕСІ. ПІСЛЯ ЦЬОГО ПЕРЕРАХОВУЮТЬСЯ ДАНІ ПО НАСТУПНІЙ АДРЕСІ Й Т.Д., ЯК І В ПЗУ. МОДЕЛЬ ОЗУ СКЛАДАЄТЬСЯ ІЗ ДВОХ СЕКЦІЙ ЗАПISУ Й ЗЧИТУВАННЯ ДАНИХ, ЯКІ МАЮТЬ РІЗНІ ВИВОДИ ДЛЯ ПОДАЧІ СИГНАЛІВ ДОЗВОЛУ, РІЗНІ ВИВОДИ ДЛЯ ЗАПISУ Й ЗЧИТУВАННЯ ДАНИХ І ЗАГАЛЬНІ ВИВОДИ АДРЕСУ (РИС. 2.33).**

**РИСУНОК 2.34 – ПЕРЕХІДНІ ПРОЦЕСИ В ОЗУ ПРИ ЗАПISУ**

**РИСУНОК 2.35 – ПЕРЕХІДНІ ПРОЦЕСИ В ОЗУ ПРИ ЧИТАННІ**

ПРИ ЗАПИСІ ДАНИХ В ОЗУ НЕОБХІДНО СПОЧАТКУ ПОДАТИ СИГНАЛИ НА АДРЕСНІ ВХОДИ Й ВХОДИ ДАНИХ І НЕ ЗМІНЮВАТИ ЇХ ПРОТЯГОМ ПЕВНОГО ЧАСУ – ЧАСУ ВСТАНОВЛЕННЯ  $T_{SUAEW}$ ,  $T_{SUDEW}$  ВІДПОВІДНО, ПІСЛЯ ЧОГО ВСТАНОВИТИ "1" НА ВХОДІ ДОЗВОЛУ ЗАПISУ. ЦЕЙ СИГНАЛ ПОВИНЕН УТРИМУВАТИСЯ ПРОТЯГОМ ДЕЯКОГО МІНІМАЛЬНОГО ІНТЕРВАЛУ ЧАСУ  $T_{wew}$  І ПОТІМ МОЖЕ БУТИ СКИНУТИЙ В "0". ПРИ ЦЬОМУ СИГНАЛИ АДРЕСИ Й ДАНИХ НЕ ПОВИННІ ЗМІНЮВАТИСЯ ПРОТЯГОМ ЧАСУ, КОЛИ СИГНАЛ ДОЗВОЛУ ЗАПISУ ПЕРЕБУВАЄ В СТАНІ "1" І ВТРИМУЄТЬСЯ ЩЕ ЯКИЙСЬ ЧАС  $T_{HAew}$  ТА  $T_{HDew}$ , ПЕРШ НІЖ ЗМІНИТИСЯ.

ДЛЯ ЧИТАННЯ З ОЗУ НА ВХІД ДОЗВОЛУ ЧИТАННЯ НЕОБХІДНО ПОДАТИ "1", А СТАНУ ВУЗЛІВ ВИХІДНИХ ДАНИХ ЗМІНЮЮТЬСЯ ВІД СТАНУ ВИСОКОГО ІМПЕДАНСУ  $Z$  ДО ВІДПОВІДНОГО СТАНУ ЧЕРЕЗ ДЕЯКИЙ ЧАС  $T_{RED}$ . ПРОТЯГОМ ЧАСУ, ПОКИ СИГНАЛ ДОЗВОЛУ ЧИТАННЯ ПЕРЕБУВАЄ В СТАНІ "1, СИГНАЛИ АДРЕСИ МОЖУТЬ ЗМІНЮВАТИСЯ, І ЯКЩО ЦЕ ТАК, ТО НОВІ ДАНІ ДОСТУПНІ НА ВИХОДАХ ЧЕРЕЗ ЯКИЙСЬ ЧАС ЗАТРИМКИ  $T_{RAD}$ .

ПЕРЕХІДНІ ПРОЦЕСИ ПРИ ЗАПИСІ ПОКАЗАНИ НА РИС. 2.34 Й НА РИС. 2.35 ПРИ ЗЧИТУВАННІ З ОЗУ У МОДЕЛІ ОЗУ НІЩО НЕ ПЕРЕШКОДЖАЄ ОДНОЧАСНО ВСТАНОВИТИ "1" НА ВХОДАХ ДОЗВОЛУ ЧИТАННЯ Й ЗАПISИ, ХОЧА В БІЛЬШОСТІ РЕАЛЬНИХ ОЗУ ЦЕ НЕ ДОПУСКАЄТЬСЯ. НОВІ ЛІЧЕНІ ДАНІ ПОСИПАЮТЬСЯ НА ВИХОДИ ДАНИХ ПІСЛЯ ПЕРЕХОДУ СИГНАЛУ ДОЗВОЛУ ЗАПISУ З "1" В "0",  
МОДЕЛЬ ДИНАМІКИ ОЗУ МАЄ ФОРМАТ  
.MODEL <ІМ'Я МОДЕЛІ ДИНАМІКИ>  
+ URAM [(ПАРАМЕТРИ МОДЕЛІ ДИНАМІКИ)]  
ПАРАМЕТРИ ЦІЄЇ МОДЕЛІ НАВЕДЕНІ В ТАБЛ. 2.19  
(ЗНАЧЕННЯ ЗА ЗАМОВЧУВАННЯМ – 0, ОДИНИЦЯ ВИМІРУ – С).

ТАБЛИЦЯ 2.19

ПАРАМЕТРИ МОДЕЛІ ОЗУ

Ідентифікатор	Параметр
TPADLMN	Час вибірки адреси при перемиканні виходів 0-1, мінімальне значення
TPADHTY	Те ж, типове значення
TPADHMX	Те ж, максимальне значення
TPERDHMN	Час вибірки адреси при перемиканні виходів 1-0, мінімальне значення
TPADLTY	Те ж, типове значення
TPADLMX	Те ж, максимальне значення
TPERDLMN	Час вибірки дозволу при перемиканні виходів Z- 1, мінімальне значення
TPERDHTY	Те ж, типове значення
TPERDHMX	Те ж, максимальне значення
TPERDHzMN	Час вибірки дозволу при перемиканні виходів Z- 0, мінімальне значення
TPERDLTY	Те ж, типове значення
TPERDLMX	Те ж, максимальне значення
TPERDLZMN	Час вибірки дозволу при перемиканні виходів 1 -Z, мінімальне значення
TPERDHzTY	Те ж, типове значення
TPERDHzMX	Те ж, максимальне значення
TSUDEWMN	Час вибірки дозволу при перемиканні виходів 0 -Z, мінімальне значення
TPERDLZTY	Те ж, типове значення
TPERDLZMX	Те ж, максимальне значення
TSUAEWMN	Час установлення даних щодо переднього фронту сигналу дозволу, мінімальне значення
TSUDEWTY	Те ж, типове значення
TSUDEWMX	Те ж, максимальне значення
TWEWHMN	Час установлення адреси щодо переднього фронту сигналу дозволу, мінімальне значення
TSUAEWTY	Те ж, типове значення
TSUAEWMX	Те ж, максимальне значення
TWEWLMN	Тривалість сигналу дозволу при записі 1, мінімальне значення
TWEWHTY	Те ж, типове значення
TWEWHMX	Те ж, максимальне значення
THDEWMN	Тривалість сигналу дозволу при записі 0, мінімальне значення
TWEWLTU	Те ж, типове значення

TWEWLMX	Те ж, максимальне значення
THAEWMN	Час утримання вхідних даних щодо заднього фронту сигналу дозволу запису, мінімальне значення
THDEWTY	Те ж, типове значення
THDEWMX	Те ж, максимальне значення
THAEWMN	Час утримання адреси стосовно заднього фронту сигналу дозволу запису, мінімальне значення
THAEWTY	Те ж, типове значення
THAEWMX	Те ж, максимальне значення

**ПРИ ПРОЕКТУВАННЯ ЦИФРОВИХ ПРИСТРОЇВ ТАКОЖ  
МОЖУТЬ ЗНАДОБИТИСЯ ДОДАТКОВІ, ДОПОМІЖНІ  
ПРИСТРОЇ.**

Джерела постійних логічних сигналів. Ці компоненти мають виходи, але не мають входів. Логічний рівень вихідного сигналу дорівнює "1" для джерел типу PULLUP і "0" для джерел типу PULLDN.

Внутрішні опори джерел задаються при описі моделі вхід/ вихід. Модель динаміки ці джерела не мають. Приведемо приклад завдання джерел логічних сигналів зі своїми моделями вхід/вихід:

```
U4 PULLUP(2) $G_DPWR $G_DGND PIN0 P1N1 R2K ; Два
; джерела "1"
U5 PULLDN(4) $G_DPWR $G_DGND BUS0, BUS1, BUS2, BUS3
+ R50; Чотири джерела "0"
.MODEL R2K UIO (drvh=2K)
.MODEL R50 UIO (drvl=50)
```

Цифрові лінії затримки. Вони здійснюють затримку вхідного сигналу будь-якої тривалості (помітимо, що вентиля не пропускають імпульси, тривалість яких менше часу затримки). Лінії затримки мають тип DLYNE, у списку вузлів після перерахування вузлів підключення джерела живлення вказуються <вузол входу> і <вузол виходу>.

Модель динаміки лінії затримки має форму  
.MODEL <ім'я моделі> UDLY [(параметри)]

Параметри лінії затримки наведені в табл. 2.20 (значення за замовчуванням - 0, одиниця виміру - с):

## Параметри цифрових ліній затримки

Ідентифікатор	Параметр
DLYMN	Мінімальна затримка
DLYTY	Типова затримка
DLYMX	Максимальна затримка

При складанні моделей складних цифрових компонентів зручно застосовувати наступні примітиви:

Logic Expression - складання логічних виразів (примітив LOGICEXP);

Pin-to-Pin Delay - завдання затримок поширення сигналів (примітив PINDLY);

Constraint Checker-Правила перевірки тимчасових співвідношень, таких, як мінімальний час установки/скидання, мінімальна тривалість імпульсів і т.п. (примітив CONSTRAINT). При порушень цих обмежень у процесі моделювання видаються попереджувачі співвідношення.

### Логічні вирази

Примітив LOGICEXP задається по формату

```
Uxxx LOGICEXP (<кількість входів>,<кількість виходів>)
+ <+вузол джерела живлення> <-вузол джерела живлення>
+ <вхідний вузол 1>... <вхідний вузол n> <вихідний вузол 1>...
<вихідний вузол n>
+ <ім'я моделі динаміки> <ім'я моделі вхід/вихід>
+ [_LEVEL=<рівень моделі інтерфейсу>}
+ [MNTYMXDLY=<вибір значення затримки>]
+ LOGIC: <логічне призначення>
```

Логічне призначення записується подвійно:

<номер вихідного вузла>={<логічне вираження>}

<проміжна змінна>={<логічне вираження>}

Кожний згаданий у специфікації вихідний вузол повинен мати один логічний вираз. Призначені один раз проміжні змінні можуть

використовуватися в наступних логічних виразах. Проміжним змінним привласнюються імена по тим ж правилам, що іменам вузлів.

Логічні вирази полягають у фігурні дужки { }. Вони записуються на одному або більше рядках, рядки продовження мають символ "+" у першій позиції. Наведемо перелік логічних операторів у порядку їхнього старшинства:

- логічне заперечення;
- & - логічне І;
- ^ - логічне що виключає АБО;
- I - логічне АБО.

В якості операндів можуть бути:

- вхідні вузли;
- попередньо певні тимчасові змінні;
- попередньо певні вихідні вузли;
- логічні константи 0,1, X, R, F.

При записі виразів можна застосовувати круглі дужки для групування операндів. Всі ці логічні вирази можуть використовуватися також і в програмі Probe.

Модель динаміки примітива LOGICEXP має такий же формат, як стандартні вентилі:

`.MODEL <ім'я моделі> UGATE [(параметри)]`

Логічні вирази обчислюються в процесі моделювання в порядку їхнього проходження. Затримка появи сигналів на вихідних вузлах визначається в моделі динаміки. Внутрішні зворотні зв'язки в логічних вираженнях не допускаються, однак зовнішні зворотні зв'язки як і раніше можливі. Наведемо приклад опису логіки функціонування арифметико-логічного пристрою 74181. Часові затримки будуть задані нижче за допомогою примітивів PINDLY і CONSTRAINT:

Завдання затримок поширення

Примітив PINDLY дозволяє призначити затримки поширення сигналів складним пристроям. Один єдиний примітив PINDLY

дозволяє моделювати часові співвідношення й вихідні характеристики цілої інтегральної схеми, що має у своєму составі, зокрема, тристабільні вентиля. Крім того, в опис примітива PINDLY допускається включення специфікації контролю тривалості імпульсів SETUP, HOLD, WIDTH, FREQ і GENERAL, які використовуються в примітиві CONSTRAINT.

Примітив PINDLY задається по формату:

```
Uxxx PINDLY (<кількість шляхів>, <кількість входів дозволу>
+ кількість додаткових вузлів) <+вузол джерела живлення>
+ <-вузол джерела живлення> <вхідний вузол 1> ...
+ <вхідний вузол n>
+ [<вузол дозволу 1 ... <вузол дозволу n>]
+ [<внутрішній вузол 1> ...
+ внутрішній вузол n] <вихідний вузол 1> ...
+ <вихідний вузол n>
+ <ім'я моделі вхід/вихід>
+ [ IO_LEVEL=<рівень моделі інтерфейсу>]
+ [ MNTYMXDLY=<вибір значення затримки>]
+ [ BOOLEAN: <бульовське призначення>*]
+ PINDLY: <призначення затримок>*
+ [TRISTATE: ENABLE LO I HI <вузол дозволу>
+<призначення затримок>*]
+ [ SETUP_HOLD:
+<специфікація часів встановлення/утримання>]
+ [ WIDTH: <специфікація ширини імпульсу>]
+ [ FREQ:<специфікація частоти повторення>]
+ [ GENERAL: <загальна специфікація>]
```

Тут прийняті наступні позначення:

<кількість шляхів> - кількість шляхів поширення сигналів від вхідних до вихідних вузлів; при цьому кількість входів шляхів повинне бути дорівнює кількості виходів. З кожним шляхом вхід-вихід асоціюється затримка, що обчислюється за певними правилами;

<кількість входів дозволу> - специфікація тристабільних вузлів, наявних у примітиві (може бути рівним нулю). Тристабільні вузли використовуються в секціях TRISTATE;



<кількість внутрішніх вузлів> - внутрішні вузли використовуються у вираженнях, що визначає затримки, але вони не входять у шляху вхід-вихід (може бути рівним нулю).

Секція BOOLEAN. За допомогою <булевського призначення> визначаються проміжні змінні, які можуть бути використані в <визначенні затримки>. Секція BOOLEAN може бути включена на будь-якому рядку в описі примітива PINDLY. Вираження <булевське призначення> має вигляд:

<булевська змінна> = { <булевское вираз> }

Ім'я <булевська змінна> складається по тим ж правилам, що ім'я вузла.

Вираження <булевській вираз> приймає значення TRUE (логічна "1") або FALSE (логічний "0"). Подібно всім іншим виразам воно повинне бути укладене у фігурні дужки {}.

У якості операндів булевських виражень можуть брати участь:

- попередньо певні <булевські змінні>;
- функцію додаткових вузлів (див. функції змін (див. нижче);
- <булевские константи> TRUE і FALSE.

Додатково оператори "==" і "Is" можуть бути застосовані до логічних значень <вхідні вузли> і <логічні константи>. Це дозволяє проаналізувати логічні стани вузлів, наприклад: вираження "CLEAR==1" приймає значення TRUE, якщо вузол CLEAR має значення логічної "1", і FALSE - у протилежному випадку.

Керуючі функції використовуються для визначення змін станів <внутрішніх вузлів> або <вихідних вузлів>. Всі керуючі функції приймають логічні значення й тому можуть входити до складу <булевських виразів>. Наведемо список цих функцій і їхніх аргументів:

CHANGED (<вузол>, <інтервал часу>)

CHANGED\_LH (<вузол>, <інтервал часу>\*)

CHANGED\_HL (<вузол>, <інтервал часу>)

Функція CHANGED приймає значення TRUE, якщо зазначений <вузол> змінював свій стан з "0" в "1" на зазначеному <інтервалі

часу>, що передує сучасний момент часу, у протилежному випадку - FALSE.

Аналогічно функція CHANGED\_LH приймає значення TRUE, якщо зазначений <вузол> змінював свій стан на зазначеному <інтервалі часу>, що передує сучасний момент часу, у протилежному випадку. Відзначимо, що CHANGED\_LH контролює тільки саму останню зміну.

Нарешті, функція CHANGED\_HL приймає аналогічні значення, контролюючи переходи з "1" в "0". Якщо <інтервал часу> задати рівним нулю, то розглянуті функції приймуть значення TRUE, якщо в цей момент часу стан вузла змінюється. Це дає можливість розбивати модель цифрових компонентів на дві частини: перша моделює логіку функціонування з нульовими затримками, а друга враховує реальні затримки.

Функції змін призначені для контролю за зміною станів <вихідних вузлів>, для яких обчислюються <вираження для затримок. Подібно додатковим функціям вони приймають значення логічного "0" або "1". Однак на відміну від них вони не мають аргументів і просто фіксують зміну станів вихідних вузлів у сучасний момент часу. Вони мають вигляд:

TRN\_pn

де p-значення попереднього стану, а n- нового стану.

Логічні значення станів позначаються символами: L (низький рівень), H (високий рівень), 2 (великий вихідний опір) і \$ (будь-який рівень). Так, наприклад, функція TRN\_H\$ контролює перехід зі стану логічна "1" у будь-який інший стан.

Функції TRN\_p і TRN\_Zn приймають значення TRUE, тільки якщо вони використовуються в секції TRISTATE. Хоча вихідні вузли пристроїв з відкритим колектором переходять у стан високого імпедансу Z (замість И), у довідниках звичайно приводяться значення затримок TPLH і TPHL. Тому в математичних моделях пристроїв з відкритим колектором необхідно використовувати функції TRN\_LH і TRNHL, а в тристабільних пристроях - TRN\_LZ, TRN\_HZ, TRN\_ZL і TRN\_ZH.

Секція PINDLY. Ключове слово PINDLY відзначає початок секції, що містить одне або трохи <призначень затримок>, які мають формат <вихідний вузол>\*={ <вирази для затримок>}

Кожний <вихідний вузол>, перерахований у специфікації примітива, повинен мати одне вираження для визначення затримок. Причому кілька вихідних вузлів можуть мати загальне <вирази для затримок>, тоді в лівій частині призначення затримок міститься список їхніх імен, поділених пробілами або комами.

<Вираження для затримок> полягає у фігурні дужки й може розташовуватися на декількох рядках. Це вираження має три значення затримок:

мінімальне, типове й максимальне. У найпростішому випадку <вираження для затримок> являє собою <значення затримок> виду:

DELAY(<min>, <typ>, <max>),

де <min>, <typ>, <max> - константи із плаваючою комою або вирази (у тому числі й параметри), у секундах.

Для специфікації невідомої величини використовується -1. Наприклад, DELAY(20ns, -1,35ns) задає мінімальну затримку 20 нс, установлювану програмою за замовчуванням типову затримку й максимальну затримку 35 нс.

У більше складних <виразах для затримок> використовуються функції CASE. Аргументами функції CASE є пари <булевскій вираз>, <вираз для затримок>, що замикаються фінальним <вираженням для затримок>, що визначають значення затримки за замовчуванням. При обчисленні функції CASE обчислюються <булевські вирази> у порядку їхнього проходження доти, поки не буде отриманий результат TRUE. Після цього затримкам привласнюються значення з відповідного вираження. Якщо жодне з виражень не має значення TRUE, затримкам привласнюються значення з останнього вираження.

Функціонування примітива PINDLY. Стану вихідних вузлів примітива PINDLY змінюються при зміні станів будь-якого вхідного вузла або вузла дозволу. Кожному вхідному вузлу відповідає вихідний вузол. Спочатку визначаються стани внутрішніх змінних у секції BOOLEAN, потім обчислюються <вираження для затримок> у секціях PINDLY або TRISTATE. Після цього стан, що змінився, вхідного вузла привласнюється відповідний йому вихідному вузлу із затримкою.

Контроль часових співвідношень. Примітив CONSTRAINT виконує перевірку дотримання тимчасових співвідношень у процесі моделювання. Контролюється мінімальний час установки/скидання,

мінімальна тривалість імпульсів, частота перемикачів і передбачений загальний механізм перевірки умов, які формулює користувач. Примітив CONSTRAINT тільки повідомляє про порушення тимчасових повідомлень і не змінює логічні стани вузлів і затримки. Він задається по формату

```
Uxxx CONSTRAINT (<кількість входів> )
+ <+вузол джерела живлення> <-вузол джерела живлення>
+ <вхідний вузол 1> ... <вхідний вузол n> <ім'я моделі
вхід/вихід>
+ [ IO_LEVEL<рівень моделі інтерфейсу>]
+ [ BOOLEAN: <булевське призначення> * ]...
+ [ SETUP_HOLD: <специфікація часів
установки/утримання>]...
+ [ WIDTH:<специфікація ширини імпульсу> ]...
+ [ FREQ:<специфікація частоти повторення>]...
+ [ GENERAL: <загальна специфікація> ] ...
```

Секція BOOLEAN. Містить одне або декілька <булевських призначень> виду:

<булевская змінна> = { <булевський вираз> }

Секції BOOLEAN можуть включатися в будь-якому порядку при описі примітива CONSTRAINT. Синтаксис <булевських виразів> такий же, що й у примітиві PINDLY, за винятком того, що не можуть бути використані функції змін.

Секція SETUP\_HOLD виконує перевірку часів установки/скидання, вона має формат

```
+ SETUP_HOLD:
+ CLOCK <визначення типу переходу>=<вхідний вузол>
+ DATA <кількість входів даних>=<вхідний вузол>...
+ < вхідний вузол k>
+ [ SETUPTIME=<значення інтервалу часу>]
+ [ HOLDTIME=<значення інтервалу часу >]
+ [ RELEASETIME=<значення інтервалу часу>]
+ [ WHEN { <булевское вираження> } ]
+ [ MESSAGE="<текст додаткового повідомлення>"]
+ [ ERRORLIMIT=<значення> ]
```

+ [ AFFECTS\_ALL | AFFECTS\_NONE I  
+ AFFECTS (#OUTPUTS)=<список вихідних вузлів>]

CLOCK задає вузол, щодо якого виконується вимір тривалості часів установки/утримання/відпускання. Параметр визначення типу переходу> приймає значення LH або HL; він вказує, від якого фронту імпульсу (0-1 або 1-0) відлічується час установки/утримання.

DATA визначає перелік вузлів, для яких виконується вимір часів установки/утримання.

SETUPTIME визначає мінімальний час, протягом якого всі вузли, перераховані в розділі DATA, повинні перебувати в незмінному стані до моменту початку відліку часів установки/утримання. <значення інтервалу часу> повинне бути ненегативною константою або вираженням (у секундах).

Деякі пристрої мають різні вимоги до часу установки залежно від стану вхідних даних ("0" або "1") у момент початку відліку. У такому випадку замість SETUPTIME застосовується одна з форм:

SETUPTIME\_L0=<значення інтервалу часу>]  
SETUPTIME\_H1=<значення інтервалу часу>]

HOLDTIME визначає мінімальний час, протягом якого всі вузли, перераховані в розділі DATA, повинні перебувати в незмінному стані після моменту початку відліку часів установки/утримання; <значення інтервалу часу> повинне бути ненегативною константою або вираженням (у секундах). Деякі пристрої мають різні вимоги до часу установки залежно від стану вхідних даних ("0" або "1") у момент початку відліку. У такому випадку замість HOLDTIME застосовується одна з форм:

HOLDTIME\_LO=<значення інтервалу часу>]  
HOLDTIME\_H1=<значення інтервалу часу>]

RELEASETIME містить специфікацію часу відновлення - мінімального інтервалу часу стабільного стану, що передує перемиканню. Якщо час відновлення залежить від напрямку

перемикання вхідних даних, то замість RELEASETIME застосовується одна з форм:

```
RELEASETIME _LH=<значення інтервалу часу>]
RELEASETIME _HL=<значення інтервалу часу>]
```

Розходження між перевіркою часу відновлення й установлення полягає в тому, що при контролі часу відновлення не дозволяється одночасна зміна CLOCK/DATA. Тому, якщо навіть час утримання HOLDTIME явно не задано, воно вважається більше нуля. Ця властивість дозволяє задати значення часу відновлення безпосередньо в тексті опису моделі. Із цієї причини час відновлення звичайно задається окремо, незалежно від специфікацій SETUPTIME або HOLDTIME. Перевірка часів установлення/утримання/відновлення починається після того, як наступила задана зміна стану вузла CLOCK (LH або HL). У цей момент часу обчислюється логічне вираження WHEN. Якщо воно приймає значення TRUE, то виконуються всі перевірки, що мають ненульові значення інтервалів часу. WIDTH містить специфікацію мінімальної тривалості імпульсу наступного виду:

```
+WIDTH
+ NODE=<вхідний вузол>
+ { MIN_HI=< значення інтервалу часу > ]
+ [ MIN_LO=< значення інтервалу часу > ]
+ [ WHEN {<булевській вираз>} ]
+ [MESSAGE="<текст додаткового повідомлення>"]
+ [ ERRORLIMIT=<значення> ]
+ [AFFECTS_ALL | AFFECTS_NONE |
+ AFFECTS (#OUTPUTS)=<список вихідних вузлів> ]
```

Змінна NODE указує вхідний вузол, для якого проводиться контроль тривалості імпульсу. MIN\_HI задає мінімальний час, протягом якого вхідний вузол повинен перебувати в стані "1", а MIN\_LO - у стані "0". Якщо значення MIN\_HI не зазначене, то воно вважається рівним нулю, що означає відсутність обмежень на мінімальну тривалість імпульсів.

FREQ містить специфікацію припустимої частоти перемикань:

```

+ FREQ
+ NODE=<вхідний вузол>
+ [ MINFREQ=< значення частоти > ]
+ [ MAXFREQ=< значення частоти > ]
+ [ WHEN { <булевський вираз> } ]
+ [ MESSAGE="<текст додаткового повідомлення>" ]
+ [ ERRORLIMIT=<значення> ]
+ [ AFFECTS_ALL I AFFECTS_NONE I
+ AFFECTS (#OUTPUTS)=<список вихідних вузлів>]

```

Змінна NODE указує вхідний вузол, для якого проводиться контроль частоти перемикачів. MINFREQ задає мінімальне частоту перемикачів, а MAXFREQ - максимальну.

При моделюванні виводяться попереджуючі повідомлення, коли період перемикачів більше/менше значення 1/<значення частоти>.

GENERAL містить специфікацію перевірок, які формулює користувачем:

```

+ GENERAL
+ WHEN {<булевський вираз> }
+ MESSAGE="<текст додаткового повідомлення>"
+ [ ERRORLIMIT=<значення> ]
+ [ AFFECTS_ALL I AFFECTS_NONE I
+ AFFECTS (#OUTPUTS)=<список вихідних вузлів>]

```

Змінна WHEN задає правило контролю у вигляді булевського виразу. Текст повідомлення про помилку задається за допомогою змінної MESSAGE.

При проектуванні пристроїв із мікропроцесорами та мікроконтролерами необхідно пам'ятати про програмну керованість цих пристроїв. Для того, що спроектувати пристрій з мікроконтролером, програмний засіб автоматизації схемотехнічного проектування потрібен мати додаткові одиниці, а саме:

- бібліотеку мікроконтролерів;
- редактор програм мікро контролера;
- дебагер для відладки програми;

- відповідний набір команд меню для роботи з мікроконтроллером;

Програми АСхП, що згадувалися у даному підручнику, а саме Multisim, Design Lab, Electronic WorkBench не мають можливості забезпечити проектування пристроїв з мікропроцесорами та мікроконтроллерами. Такі можливості є, наприклад у відомій програмі Proteus 7.2 та його попередні модифікації.

Розгляд особливостей автоматизованого проектування схем з мікроконтроллерами потребує особливої уваги та займає великий обсяг матеріалу. Курс дисципліни “Автоматизація схемо технічного проектування”, на жаль, не може вмістити ці відомості і їх слід розглядати у відповідних курсах та модулях.



## Перелік літературних джерел

1. Ильин В.Н. Автоматизация схемотехнического проектирования: учеб. пособие для вузов [Текст] / В.Н. Ильин, В.Т. Фролкин, А.И. Бутко и др. – под ред. В.Н. Ильина. – М. : Радио и связь, 1987, – 368 с.
2. Кологривов, В.А. Основы автоматизированного проектирования радиоэлектронных устройств [Текст] / В.А. Кологривов. – Томск : ТУСУР, 2002. – 221 с.
3. Валуєв, М.І., Системотехніка та основи проектування аеронавігаційних систем. [Текст] / М.І. Валуєв, В.П. Харченко, А.Н. Яппаров. – К.: НАУ. - 2003р.-120с.
4. Амелина, М.А. Программа схемотехнического моделирования Micro-Cap 8 [Текст] / М.А. Амелина, С.А. Амелин. – М. : Горячая линия-Телеком, 2007. – 464 с.
5. Ларин В.Ю. Проектирование печатных плат в пакете PCBworks [Текст] / В. Ю. Ларин. – Донецк: Норд-Пресс, 2005. – 128 с.
6. Pescovitz, David. The release of SPICE, still the industry standard tool for integrated circuit design, Lab Notes / D. Pescovitz – Berkeley College of Engineering. 1972. – 13p.
7. Разевиг В.Д. Система схемотехнического моделирования и проектирования печатных плат Design Center. [Текст] / В.Д. Разевиг. – М.: «СК Пресс» – 1996г. – 272с.
8. Jiles D.C. Theory of ferromagnetic hysteresis / D.C. Jiles, D.L. Atherton // Magnetism and magnetic materials. – 1983. – Vol. 61. – P. 48–68.
9. Chan J.H. Nonlinear transformer model for circuit simulation / J.H. Chan, A. Vladimiresku, X-Ch. Gao // IEEE transactions on computer-aided design. – 1991. – Vol. 10 – P. 476–482.
10. Ларін В.Ю. Ефективність розподілених обчислень при застосуванні моделі Ізінга / В.Ю. Ларін, Є.В. Шкурніков / Восточно – Европейский журнал передовых технологий. – 2011. – № 5/4 (53). – С. 19–23.
11. MicroSim PSpice A/D & Basics+. User Guide. – Irvine, California : MicroSim Corporation. – 567 p.
12. Модели МОП-транзисторов [Електронний ресурс] / БиГОР (База и генератор образовательных ресурсов). Режим доступа

- .
- [http://bigor.bmstu.ru/?cnt/?doc=010\\_EDA/eda104.mod/?cou=Default/020\\_ECAD.cou](http://bigor.bmstu.ru/?cnt/?doc=010_EDA/eda104.mod/?cou=Default/020_ECAD.cou).
13. Красько, А.С. Схемотехника аналоговых электронных устройств: учебное пособие [Текст] / А.С. Красько. – Томск : ТУСУР, 2005. – 178 с.
  14. Basic Opamp model [Электронный ресурс] / Eircuitcentre – Режим доступа : url \www/  
<http://www.ecircuitcenter.com/Circuits/opmodel1/opmodel1.html>
  15. Фолкенберри Л. Применение операционных усилителей и линейных интегральных схем. [Текст] / Л. Фолкенберри – М.: Мир, 1985. – 572 с.
  16. Boyle, Graeme R. Macromodeling of integrated Circuit Operational Amplifiers. / Graeme R. Boyle// IEEE Journal of solid state circuits. – Vol.sc-9. – №6. – 1974. – PP. 45-58.
  17. Ларін В.Ю. Лабораторний практикум із дисципліни “Схемотехніка та автоматизоване проектування” для студентів напряму підготовки 6.070102 “Аеронавігація” [Текст] / / В.Ю. Ларін. – К.: НАУ, 2012. – 129 с.
  18. Шило В.Л. Популярные цифровые микросхемы: Справочник. [Текст] / В.Л. Шило – М.: Металлургия. – 1988г. – 352с.
  19. Москатов Е.А. Электронная техника / Е.А. Москатов. – Таганрог : Радио,2 004. – 121 с.
  20. Clive Maxfield. The Design Warrior’s Guide to FPGA. – Burlington : Elsevier, 2005. – 389 p.

Навчальне видання

ЛАРІН Віталій Юрійович  
ХАРЧЕНКО Володимир Петрович

## **АВТОМАТИЗАЦІЯ СХЕМОТЕХНІЧНОГО ПРОЕКТУВАННЯ**

Підручник

Редактор *Н. П. Мельник*  
Технічний редактор *А. І. Лавринович*  
Коректор *Л. М. Романова*  
Художник обкладинки *О. О. Зайцева*  
Комп'ютерна верстка *Л. Т. Колодіної*

Підп. до друку 20.04.13. Формат 60x84/16. Папір офс.  
Офс. друк. Ум. друк. арк. 14,65. Обл.-вид. арк. 15,75.  
Тираж 100 пр. Замовлення № 86-1.

Видавець і виготовлювач  
Національний авіаційний університет  
03680. Київ – 58, проспект Космонавта Комарова, 1  
Свідоцтво про внесення до Державного реєстру ДК №  
977 від 05.07.2002