

В.Р. Медвідь, В.П. Пісьціо

ПРОЕКТУВАННЯ ТА АНАЛІЗ ЕЛЕКТРИЧНИХ СХЕМ В ПРОГРАМНОМУ СЕРЕДОВИЩІ Proteus VSM

Тернопільський національний технічний
університет імені Івана Пулюя

Кафедра автоматизації
технологічних процесів
і виробництв

Проектування та аналіз електричних схем в програмному середовищі Proteus VSM

Методичні вказівки до самостійної роботи
студентів
з курсу

"Проектування мікропроцесорних систем
керування технологічними процесами"

Тернопіль 2018

Проектування та аналіз електричних схем в програмному середовищі Proteus VSM. Методичні вказівки до самостійної роботи студентів з курсу "Проектування мікропроцесорних систем керування технологічними процесами". Медвідь В.Р., Пісьціо В.П., Тернопіль: ТНТУ, 2018 - 26 с.

Для студентів напряму підготовки: 151 "Автоматизація та комп'терно-інтегровані технології"

Розглянуто і затверджено на засіданні кафедри автоматизації технологічних процесів і виробництв (протокол № 1 від 29.08.2018 року)

Укладачі:

доцент, к.т.н. Медвідь В.Р.,

асистент Пісьціо В.П.

Методичні вказівки по проектуванню та аналізу електричних схем в програмному середовищі Proteus VSM

Вступ

Proteus - це комерційний пакет програм класу САПР, який об'єднує в собі дві основні програми:

- ISIS - засіб розробки і налагодження в режимі реального часу електронних схем і
- ARES - засіб розробки друкованих плат. В якості автоматичного вбудованого трасувальника в ARES, починаючи з версії 7.4, використовується програма ELECTRA Autorouter. Відмінність від аналогічних за призначенням пакетів програм, наприклад, Electronics Workbench Multisim, MicroCap, Tina і т.п. - в розвиненій системі симуляції (інтерактивної налагодження в режимі реального часу і покрокової) для різних сімейств мікроконтролерів: 8051, PIC (Microchip), AVR (Atmel), і ін.

Proteus має великі бібліотеки компонентів, в тому числі і периферійних пристроїв: світлодіодні і РК індикатори, температурні датчики, годинник реального часу - RTC, інтерактивних елементів введення-виведення: кнопок, перемикачів, віртуальних портів і віртуальних вимірювальних приладів, інтерактивних графіків, які не завжди присутні в інших подібних програмах.

1. Початок роботи з Proteus VSM

Proteus VSM може використовуватися двома досить відмінними способами — або для Інтерактивного Моделювання або для Моделювання на Основі Діаграм.

Як правило, інтерактивне моделювання використовують, щоб побачити, чи працює розробка взагалі, а графічне - щоб дослідити, чому вона не працює або проводити деталізовані вимірювання.

1.1. Налаштування середовища проектування

Нижче наведено основне вікно програми ISIS з поясненнями по призначенню основних елементів інтерфейсу: *ліве меню, верхнє меню команд, верхнє основне меню, кнопки симуляції, селектор об'єктів*.

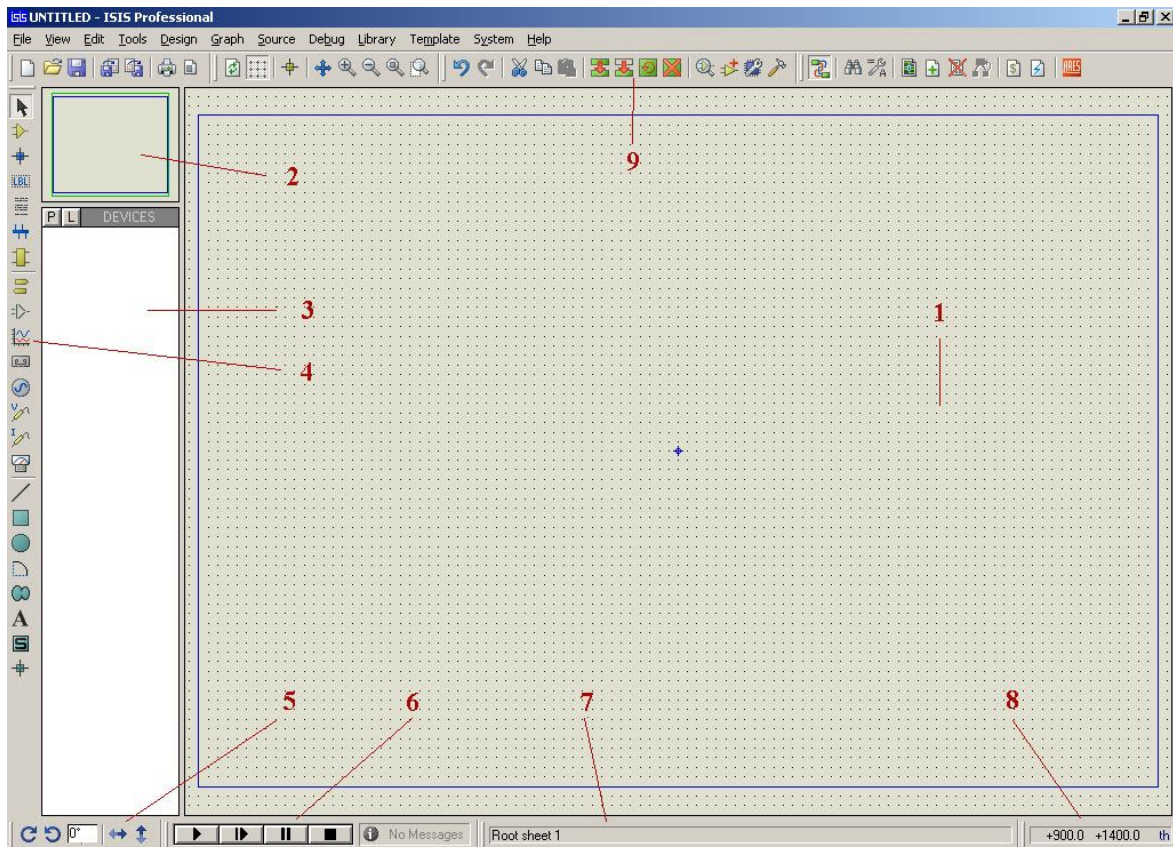


Рис. 1 Вигляд основного вікна програми ISIS

Основні елементи головного робочого вікна:

1. Вікно редагування **Editing Window** — в ньому відбуваються основні операції створення, редагування і відлагодження моделі.
2. Вікно навігації і попереднього перегляду **Overview Window** — допомагає побачити в вікні редагування потрібну частину схеми, якщо вона повністю в ньому не поміщається. Достатньо курсором миші вибрати потрібну точку зменшеного зображення схеми у вікні попереднього перегляду.
3. Вікно вибору об'єктів **Object Selector** — в процесі роботи тут розміщується список компонентів, що використовуються у створюваній моделі, символів та інших об'єктів.
4. Панель інструментів і вибору режимів:

Title	Toolbar
Основні режими	
Пристрої	
2D Графіка	

Рис. 2 Панель інструментів і вибору режимів

4.1. Основні режими:

Редагування	Компоненти	Точка з'єднання провідників	Маркування провідників	Додати опис	Шина	Підсхема

- безпосереднє редагування;
- компоненти проекту;
- встановлення точки з'єднання провідників вручну;
- маркування провідників;
- додавання опису;
- шини;
- підсхеми.

4.2. Пристрої:

Термінали	Виводи пристроїв	Графи	Магнітофон	Генератори	Пробник напруги	Пробник струму	Вимірювальні пристрої

- термінали (живлення, “земля”, міжблокові з'єднання, виводи);
- виводи пристроїв;
- графи;
- “магнітофон” (використовується для запису у файл і відтворення даних);
- генератори;
- пробник напруги;
- пробник струму;
- віртуальні вимірювальні пристрої.

5. Панель повороту і дзеркального відображення.
6. Анімаційна панель:

- “**Play**” — початок симуляції або продовження призупиненої симуляції,
 - “**Step**” — виконати мінімальний часовий крок,
 - “**Pause**” — призупинення симуляції,
 - “**Stop**” — зупинка симуляції.
7. Рядок стану — в ньому відображаються підказки, повідомлення про помилки, час виконання симуляції та ін.
8. Вікно координат курсора.
9. Панель інструментів команд:

Title	Toolbar
Файл/друк команди	
Команди відображення	
Команди редагування	
Проектні інструменти	

Рис. 3 Панель інструментів команд

9.1. Команди управління файлами та друком:

Новий проект	Відкрити файл	Зберегти файл	Друк	Вказати область друку	Імпортувати секцію	Експортувати секцію

9.2. Команди управління відображенням:

Перемалювати	Сітка	Точка відліку	Панорама	Збільшити	Зменшити	Переглянути повністю	Переглянути область

9.3. Команди редагування:

Відмінити	Виконати повторно	Вирізати	Копіювати	Вставити	Копіювання блоку	Переміщення блоку

Видалення блоку	Поворот блоку	Додати компонент	Створити компонент	Інструмент для корпусу	Розбити компонент

9.4. Інструменти управління проектом:

авто трасування провідників	Знайти мітку	Інструмент призначення властивостей	Оглядач проекту	Новий лист
Видалити аркуш	Повернутись до основного листа	Звіт по матеріалам	Перевірка електричних з'єднань	Список зв'язків для ARES

2. Створення нового проекту. Інтерактивне моделювання

2.1 Побудова схеми

Розміщення елементів

Створимо модель пристрою, схема якого показана на рис. 4. Це своєрідний мікропроцесорний “мультивібратор”. Частоту спалахів світлодіоду HL1 можна змінювати резистором R2.

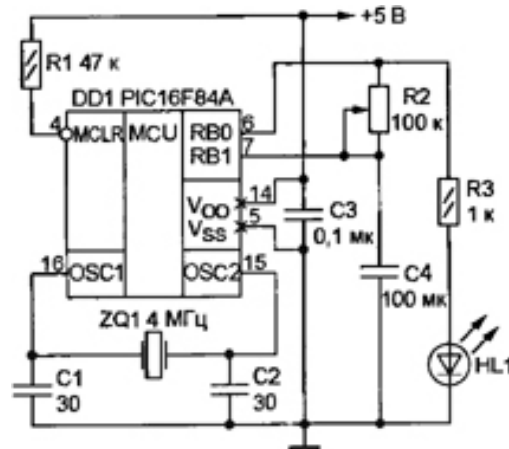


Рис. 4 Схема пристрою для моделювання

При запуску ISIS створює новий проект автоматично. В іншому випадку, для створення нового проекту необхідно вибрати в меню **File** пункт **New Design**, або натиснути відповідну піктограму.

Виберіть на панелі інструментів піктограму **Компоненти (Component Mode)**. Натисніть на піктограмі з літерою “P”, що знаходиться над **вікном вибору об'єктів (Object Selector)**, щоб відкрити вікно **Браузера Бібліотек (Library Browser)**.

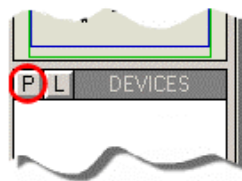


Рис. 5 Виклик Браузера Бібліотек

У вікні **Браузера Бібліотек** в полі **Keywords** введіть **назву мікроконтролера PIC16F84A**. Щоб перенести знайдений компонент у вікно вибору об'єктів, натисніть два рази лівою кнопкою миші на рядку з його найменуванням, що з'явиться у вікні **Results**.

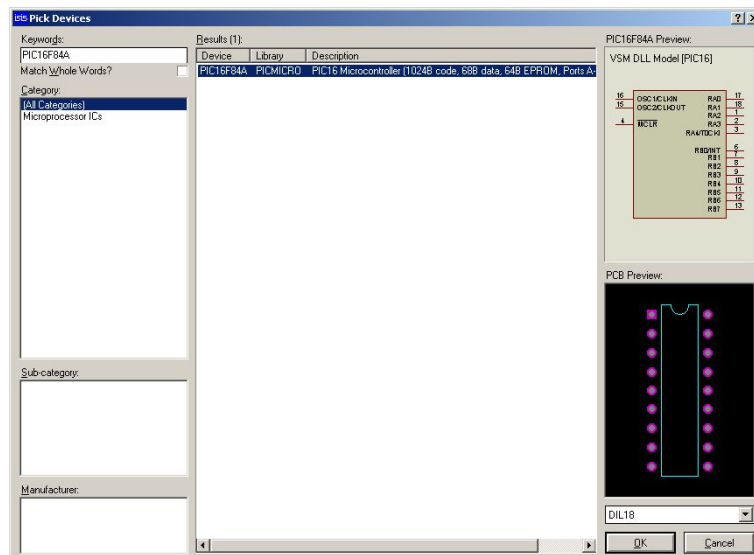


Рис. 6 Вікно Браузера Бібліотек

Знайдіть в бібліотеці резистор (**RES**), змінний резистор (**POT-LIN**), конденсатор (**CAP**) і світлодіод червоного свічення (**LED-RED**). Компоненти вибираються в одному екземплярі, розмножити їх можна згодом, просто вибираючи їх у вікні вибору об'єктів.



Рис. 7 Вибір компонентів у вікні вибору об'єктів

Щоб розмістити елемент на схемі, виберіть його у вікні вибору об'єктів (елемент буде показаний в вікні попереднього перегляду в правому верхньому куті екрану), перемістіть курсор у потрібне місце в вікні редагування та натисніть ліву кнопку миші.

З'явиться контур об'єкту, який буде переміщуватися разом із курсором. При натисненні лівої кнопки ще раз, компонент буде встановлено.

Перед тим, як помістити елемент на схему, його можна попередньо повернути на певний кут чи відобразити дзеркально, використовуючи панель повороту і дзеркального відображення.

Встановіть на схему мікроконтролер. Тепер виберіть компонент **RES**, натисніть на піктограмі повороту проти годинникової стрілки. У вікні попереднього перегляду побачите резистор, розвернутий на 90°. Встановіть його відповідно до схеми. Тепер, аналогічно, встановіть на схему конденсатор, змінний резистор та світлодіод.

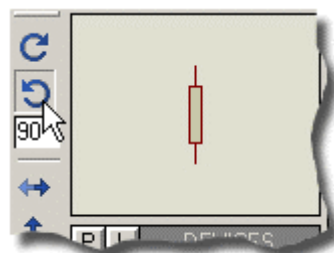


Рис 8 Поворот компонента

Щоб змінити розміщення елемента, виберіть його (права кнопка миші), натисніть ліву кнопку миші і перетягніть елемент на необхідну позицію. Натиснення правої кнопки миші над попередньо вибраним об'єктом видалить його. Повернути ж компонент після його розташування на схему можна, виділивши його правою кнопкою миші і вибравши відповідний пункт в контекстному меню.

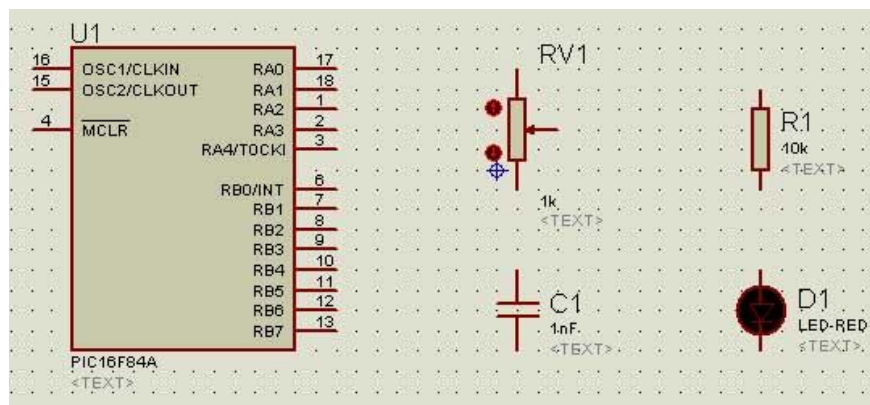


Рис 9 Попереднє розташування компонентів на схемі

Маркування і зміна властивостей моделей

При додаванні компонентів на схему, одразу ж відбувається автоматична анотація цих компонентів. Якщо уважно роздивитися резистор, то можна помітити, що ISIS маркує його величиною по замовчуванню (R) і певним унікальним значенням (порядковим номером). Дану властивість можна знайти в меню **Tools**, пункт **Real Time Annotation**, по замовчуванню вона ввімкнена.

Щоб змінити маркування і інші властивості, відкриємо вікно редагування компоненту.

Виберіть режим безпосереднього редагування (☞) на панелі інструментів, виділіть резистор R1 правою кнопкою миші та натисніть ліву, або, після натиснення правої кнопки, виберіть пункт **Edit Properties** (або комбінація клавіш CTRL+“E”) в контекстному меню.



Рис. 10 Виклик вікна редагування компонента
Змініть його позиційне позначення
Змініть його позиційне позначення

(**Component Reference**) з промаркованого R1 на R3 (для більшої відповідності початковій схемі) і опір (**Resistance**) на 1 кОм (1k).

У випадках, коли важливим є лише логічний сигнал, аналогові елементи намагаються замінити цифровими, що виконують аналогічну функцію. Цим комп'ютер звільняється від рутинного розрахунку перехідних процесів в аналогових елементах. Моделі багатьох компонентів мають “цифровий” і “аналоговий” варіанти. Змінюйте типи моделей, проте лише там, де це дозволяється схемою.

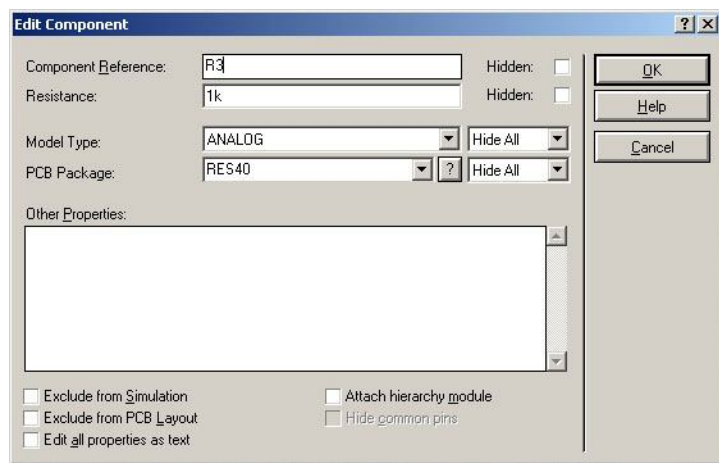


Рис. 11 Вигляд вікна редагування компонента

Виберіть конденсатор, змініть його позначення на C4, а ємність (**Capacitance**) - на 100 мкФ (100 uF). Змініть також позначення світлодіоду на HL1 та його номінальний струм (**Full drive current**) до 2 мА (mA), позначення змінного резистора на R2 і номінал до 100 кОм.

Засоби масштабування і прив'язки. Трасування з'єднань

Як правило, при розводці схеми необхідно змінювати масштаб певної території.

Натиснення клавіші <F6> або піктограми **Збільшити (Zoom In)** збільшить масштаб навколо поточного положення курсору миші, або, в якості альтернативи, затисніть

клавішу **SHIFT**, і, затиснувши ліву кнопку миші, виділіть область, яку необхідно збільшити.

Щоб зменшити масштаб, натисніть клавішу **<F7>** або піктограму **Зменшити (Zoom Out)**, або, якщо ви хочете побачити робочий аркуш повністю, натисніть клавішу **<F8>** або використовуйте коліщатко миші, щоб зменшувати або збільшувати необхідну область схеми. Відповідні команди можуть бути також доступні в меню **View**.

Найпростіший спосіб з'єднати компоненти схеми — використовувати функцію автотрасування (**Wire Auto Router**), меню **Tools**. Переконайтесь, що вона ввімкнена, підведіть курсор миші на кінець виводу 6 мікроконтролера (**RB0/INT**), вигляд курсору зміниться, натисніть ліву кнопку миші, щоб почати з'єднання, і перемістіть курсор до верхнього виводу змінного резистора R2. Коли курсор знову змінить свій вигляд, натисніть ліву кнопку миші, щоб завершити з'єднання.

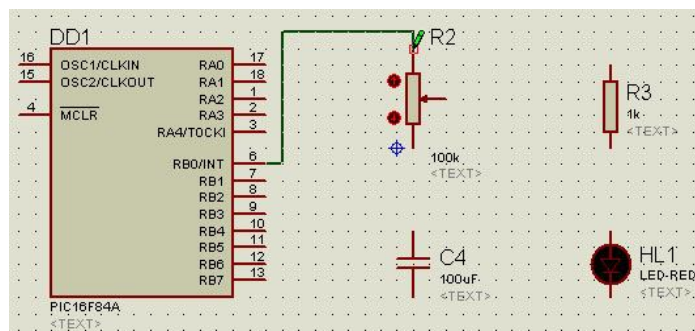


Рис. 12 З'єднання компонентів

Якщо ж вам не сподобається трасування провідників, зроблене автоматично, то можете відредагувати його вручну. Якщо ж хочете прокласти провідник без автотрасування, то можете просто натиснути лівою кнопкою миші на перший вивід деталі, і далі натискати в тих точках, де ви хочете зробити кут, і, нарешті в кінцевій точці.

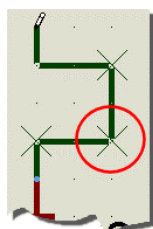


Рис. 13 Прокладення з'єднань вручну

З'єднайте всі компоненти відповідно до схеми.

Тепер потрібно “заземлити” виводи конденсатора і світлодіода. Елементи такого типу (термінали) вибираються у відповідному режимі **Terminals Mode**. Виберіть елемент **GROUND** (земля) і розмістіть його на схемі під конденсатором і світлодіодом.

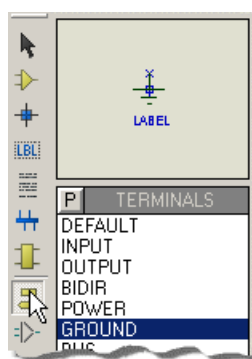


Рис. 14 Вибір терміналів

Також виберіть режим “Віртуальні вимірювальні пристрої” (*Virtual Instruments Mode*) та додайте до зібраної моделі вольтметр (DC VOLTMETER). Підключіть вольтметр паралельно конденсатору C4.

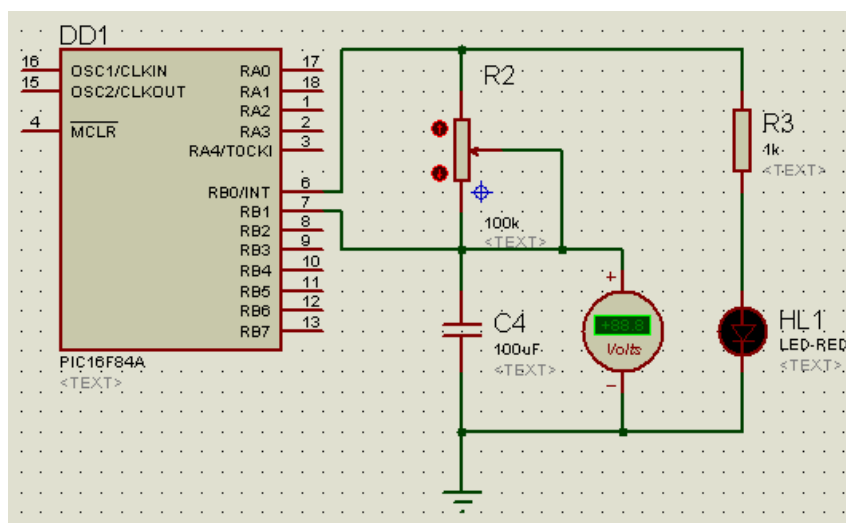


Рис. 15 Вигляд побудованої схеми

Підключати до моделі мікроконтролера живлення, встановлювати резистор R1, конденсатори C1-C3 і кварцевий резонатор не потрібно. В Proteus ці елементи емулюються програмно і вважаються вже підключеними. Деякі властивості, такі, наприклад, як частота, задаються безпосередньо в параметрах процесора.

Відкрийте вікно редагування мікроконтролера і натисніть на кнопку “*Hidden Pins*” (приховані виводи) — відкриється додаткове меню, в якому показано, як по замовчуванню називаються вузли (провідники) схеми, до яких підключені живлення мікроконтролера — VDD і його загальний провід — VSS. Зміна цих параметрів може знадобитися при живленні мікроконтролера або інших компонентів різними напругами, або від різних джерел.

Збережіть створений проект. Виберіть пункт меню *File > Save Design*. В наступному діалоговому вікні вкажіть, де ви хочете зберегти проект та назву проекту. Назвіть проект *Mult*. Тип файлу *.DSN* буде призначений автоматично. Файли з таким розширенням є файлами проектів Proteus VSM.

2.2 Написання програми

Proteus можна використовувати як автономний налагоджувач, або як віртуальний схемний емулятор (*Virtual ICE*). Ми будемо розглядати Proteus в якості автономного налагоджувача.

Система управління початковим кодом

Система управління початковим кодом виконує дві основні функції:

- Реєстрація файлів початкового коду в ISIS, з можливістю їх виклику для редагування без ручного переключення до інших прикладних програм.
- Встановлення правил для компіляції початкового коду в об’єктний код.

Створення початкового файлу

Початковий текст програми на мові асемблера MPASM, по якій повинен працювати “мультивібратор”, приведений нижче. Наберіть його в будь-якому текстовому редакторі і збережіть в папці проекту (*Mult*) під назвою *Mult.asm*.

Всі мітки повинні починатися з початку рядка, а в останньому рядку повинна бути директива **end**.

	list p=16f84a #include <p16F84A.inc>		
	org	0x00	;Встановити початок програми у вектор ;скидання 0x00
	clrf	PORTB	;Очистити регістр PORTB
	bsf	STATUS, RP0	;Перейти в банк 1 (встановити в 1 RP0 ;біт регістру STATUS)
	movlw	0xFE	;Записати константу 0xFE в акумулятор
	movwf	TRISB	;Встановити RB0 як вихід, RB1 як вхід
	bcf	STATUS, RP0	;Перейти в банк 0 (встановити в 0 RP0 ;біт регістра STATUS)
LOOP	bsf	PORTB, 0	;Встановити на виході RB0 одиницю
LOOPH	btfs	PORTB, 1	;Перевірити перший біт регістру
	goto	LOOPH	;PORTB. Якщо він дорівнює 0, то виконується ;наступна команда і відбувається “зациклення” ;(перевірка наявності одиниці на вході RB1)
	bcf	PORTB, 0	;Встановити на виході RB0 нуль
LOOPL	btfs	PORTB, 1	;Перевірити перший біт регістру
	goto	LOOPL	;PORTB. Якщо він дорівнює 1, то виконується ;наступна команда і відбувається “зациклення” ;(перевірка наявності нуля на вході RB1)
	goto	LOOP	;Перехід до основного циклу
	end		

Приєднання початкового файлу до проекту

Щоб додати початковий файл до проекту:

1. Виберіть команду **Add/Remove Source Files** в меню **Source**.
2. Натисніть на кнопку **New** та в наступному діалоговому вікні виберіть або введіть назву файлу початкового коду (якщо ввести назву не існуючого файлу, ISIS запропонує його створити).
3. В полі **Code Generation Tool** виберіть засіб трансляції програми. В нашому випадку, це асемблер MPASM. Версія цього асемблера, що поставляється з Proteus, не є найновішою. В разі необхідності, її можна замінити більш свіжою.

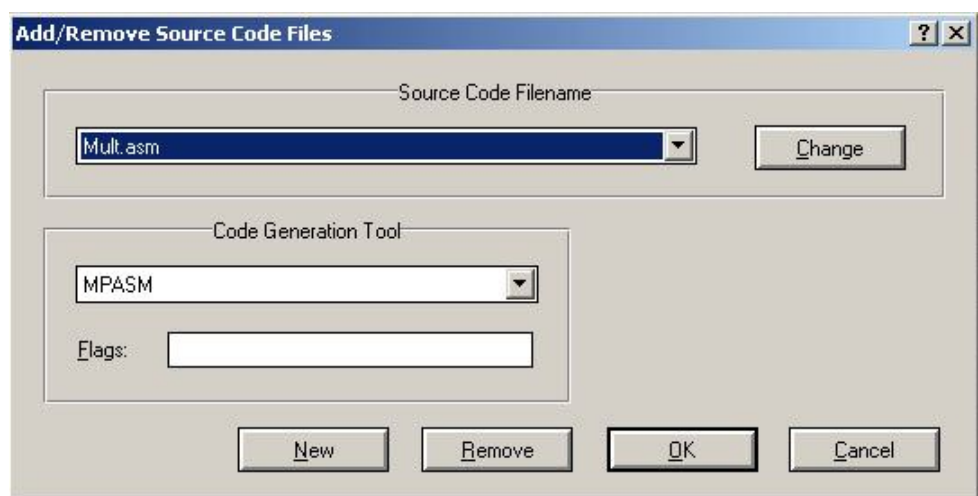


Рис. 16 Приєднання початкового файлу до проекту

4. Натисніть **OK**, щоб додати початковий файл до проекту.

Відредагувати вже додані файли можна в меню **Source**, вибравши потрібний файл.

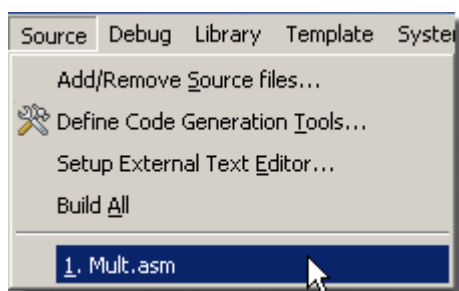


Рис. 17 Виклик початкового файлу для редагування

Proteus VSM поставляється з простим текстовим редактором — **SRCEDIT**, який використовується для редагування початкових файлів. **SRCEDIT** — це розширена версія редактора **NOTEPAD**, з можливостями відкриття складних початкових файлів, і відповіді на DDE запити. Змінити редактор на будь-який інший можна в меню **Source** > **Setup External Text Editor**.

Компіляція початкового файлу

Для того, щоб скомпілювати початковий файл, виберіть пункт меню **Source** > **Build All**. Після вибору вказаного пункту, меню початковий текст програми зберігається і запускається програма MPASM. Після завершення компіляції, на екрані з'явиться вікно з повідомленнями компілятора про знайдені помилки або їх відсутність.

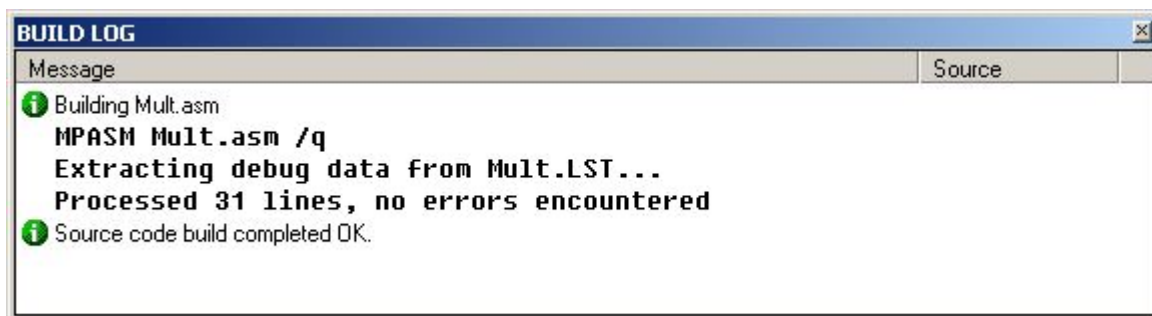


Рис. 18 Вигляд вікна звіту компіляції

Якщо в тексті були допущені помилки, можете виправити їх, відредагувавши початковий файл. Після редагування, збережіть зміни і повторіть компіляцію (**Build All**).

Встановлення додаткових засобів трансляції

В поставку Proteus вже входять деякі компілятори (MPASM, MPASMWIN, AVRASM та ін.). Проте, існує можливість додавати нові:

- Виберіть пункт **Define Code Generation Tools** в меню **Source**.

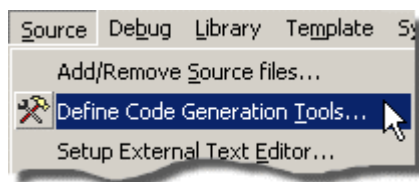


Рис. 19 Виклик вікна встановлення додаткових засобів трансляції

- Натисніть **New** і вкажіть розміщення виконавчого файлу транслятора. Ви також можете реєструвати командні файли як засоби трансляції.
- Введіть розширення для файлів початкового і об'єктного кодів. Вони визначають типи файлів, які ISIS буде шукати, вирішуючи, використати засіб для

специфічного початкового файлу, чи ні. Якщо ввімкнете опцію **Always Build**, то цей засіб буде завжди викликатися і тоді розширення об'єктного коду не потрібне.

- Визначіть орієнтаційний командний рядок (**Command Line**) для транслятора. Використовуйте %1, щоб представити файл початкового коду і %2 для файлу об'єктного коду; %\$, щоб вказати шлях до директорії Proteus і %~ для директорії розміщення DSN файлу.

Останній пункт використовується для встановлення параметрів командного рядка, які необхідні для більш злагодженого (тобто без пауз на введення додаткових параметрів користувачем) запуску засобів трансляції, і для визначення шляхів до директорій спеціальних заголовочних файлів і т. і.

В останньому блоці налаштувань (**Debug Data Extraction**) вибирається спеціальна програма — **Debug Data Extractor (DDX)**, яка зчитує інформацію з файлу лістинга (*.LST), що створюється асемблером або компілятором і генерує файл, який використовується згодом для покрокового налагодження (*.SDI, якщо засобом трансляції є асемблер).

Якщо компілятор, який ви використовуєте, генерує COF, ELF/DWARF та ін. файли, тоді використовувати DDX немає необхідності.

2.3 Налагодження програми

Використання Proteus як автономного налагоджувача вимагає, щоб символічний формат налагодження, вироблений вашим компілятором, підтримувався одним із спеціальних завантажувачів, доступних в Proteus. Завантажувач відслідковує адреси кожного рядка початкового тексту в програмі мови високого рівня, і, де це можливо, розміщення програмних змінних.

Формати налагодження, що підтримуються Proteus — **COD** (підтримується багатьма компіляторами, використовується при налагодженні PIC), **UBROF** — використовується всіма IAR - компіляторами, **ELF/DWARF** (підтримується багатьма компіляторами), **COFF** (підтримується багатьма компіляторами) та **OMF** (генерується компіляторами Keil). **SDI** формат застосовується лише для програм, написаних на асемблері та проасембльованих безпосередньо в ISIS.

Якщо використовувався асемблер або компілятор, який підтримує Proteus, то він створить вікно початкового коду для кожного початкового файлу, задіяного в проекті. Ці вікна з'являться в меню **Debug**, коли симуляція буде призупинена (поставлена на паузу).

Симуляція схеми

Після успішної трансляції в папці **Mult** (папка проекту) з'явиться файл **Mult.HEX**, який необхідно “завантажити” в модель мікроконтролера. Для цього відкриємо вікно його властивостей (виділимо правою кнопкою миші та виберемо **Edit Properties** в контекстному меню). В полі **Program File** за допомогою піктограми справа від поля вкажіть шлях до файлу **Mult.HEX**. Встановіть тактову частоту процесора в полі **Processor Clock Frequency** — 4 МГц (4MHz). Слово конфігурації (**Program configuration word**), якщо воно, як в нашому випадку, не було задане в початковому тексті програми, можна ввести в шістнадцятковій формі або прийняти встановлене по замовчуванню. За допомогою меню **Advanced Properties** встановлюються додаткові параметри мікроконтролера.

Для початку симуляції натисніть на кнопку “**Play**” на анімаційній панелі в правому нижньому куті екрану. Програма ISIS почне симуляцію спроектованого пристрою.

Біля виводів компонентів з'являться помітки у вигляді квадратів, кольори яких відповідають поточним логічним рівням: червоний — високий, синій — низький, сірий — невизначений (проміжний). Покази вольтметра зростають — йде зарядка конденсатора C4. Світлодіод HL1 ввімкнений.

Як тільки напруга на конденсаторі досягне значення, що сприймається мікроконтролером як логічна “1”, програма змінить високий рівень на виводі 6 мікроконтролера на низький, світлодіод погасне і почнеться розрядка конденсатора C4.

Як тільки напруга на ньому знизиться до рівня, який сприймається як низький (вхід мікроконтролера має характеристику тригера Шмітта), на виводі 6 знову буде встановлений високий рівень, що приведе до ввімкнення світлодіоду і зарядки конденсатора.

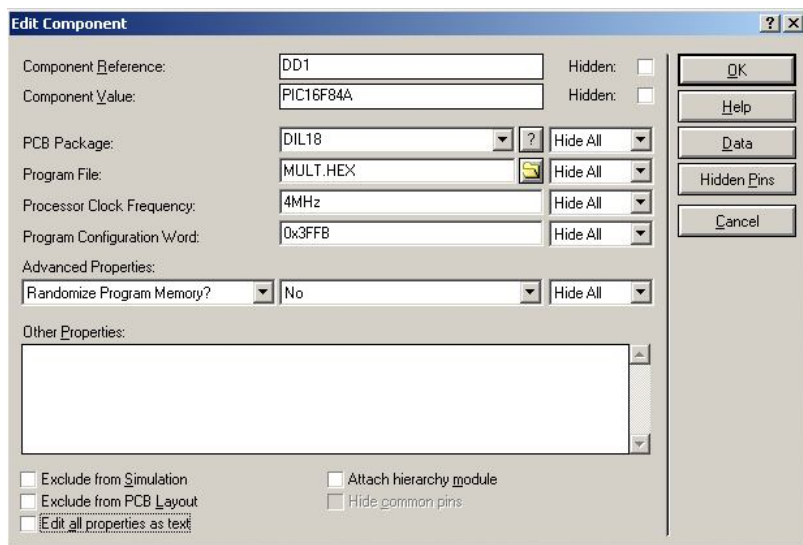


Рис. 20 Вікно редагування мікроконтролера

Далі описаний процес повторюється періодично.

Переконайтеся, що частоту спалахів світлодіоду можна регулювати змінним резистором R2. Положення движка цього резистора змінюють, натискаючи на кнопки зі стрілками біля його умовного позначення.

Вікно початкового коду

Вікно початкового коду є первинним інструментом для налагодження проектів в Proteus. У вікно налагодження початкового коду програма завантажується через налагоджувальний файл, згенерований компілятором, або через SDI-файл, згенерований системою управління початковим кодом.

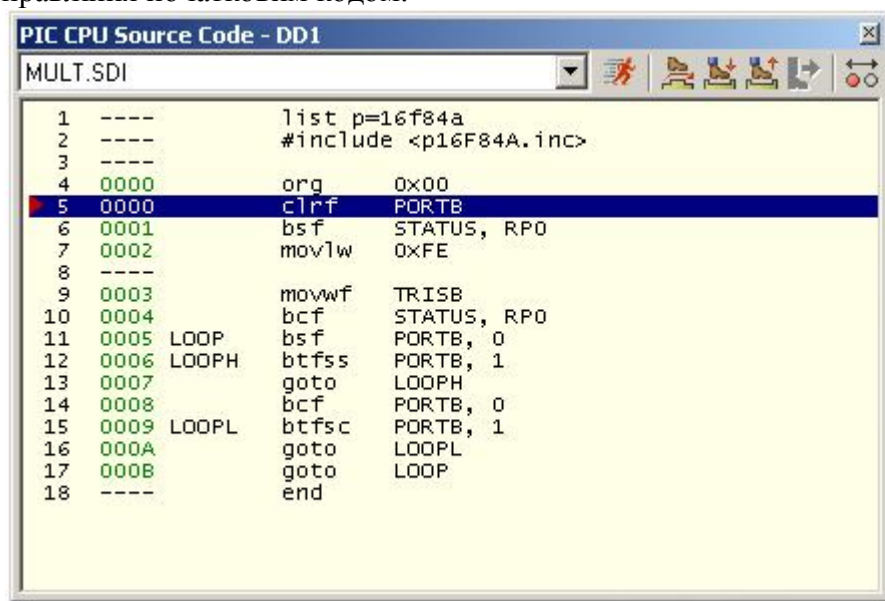


Рис. 21 Вікно налагодження

Зупиніть симуляцію, натиснувши на кнопку “**Stop**”, а потім знову запустіть, на цей раз кнопкою “**Step**”. В результаті, модель буде ініціалізована, але зупинена в початковому стані. На екрані з’явиться вікно початкового коду (вікно налагодження) з виведеним в ньому текстом початкової програми. Якщо таке вікно не з’явилося або ви його випадково закрили, воно може бути активоване в меню **Debug** (пункт **CPU Source Code**) разом з іншими інформаційними вікнами.

Рядок програми, який відповідає поточному значенню програмного лічильника (в даному випадку — нульовому), відмічений червоним трикутником.

Випадаючий список у верхній частині вікна дозволяє вибрати необхідний файл налагодження (для програм, проасембльованих в ISIS) або необхідний початковий файл, якщо їх є декілька.

Основні команди налагодження (доступні також в меню **Debug**):

- **Execute** — запуск або продовження виконання програми (еквівалент кнопки “Start”).
- **Step Over** — крок без входу в підпрограму.
- **Step Into** — крок з входом в підпрограму.
- **Step Out** — виконувати симуляцію до виходу з підпрограми (зупинка відбувається на наступній команді після команди повернення).
- **Step To** — виконувати програму до виділеного рядка.

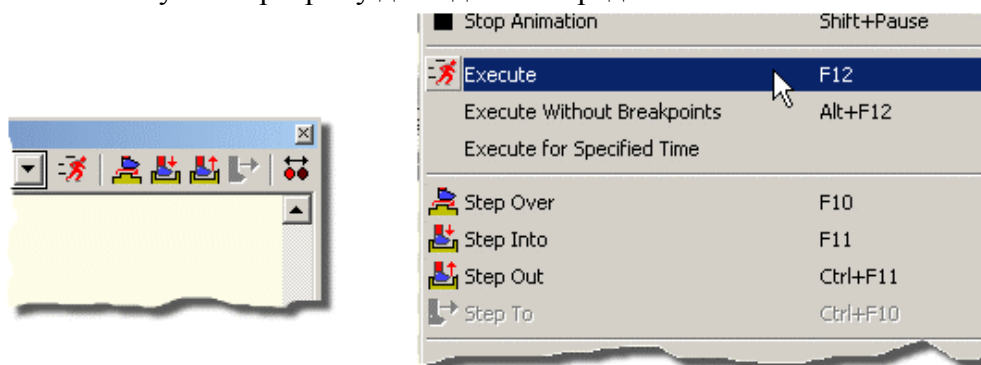


Рис. 22 Основні команди налагодження

Натиснувши праву кнопку миші в полі вікна налагодження, можна налаштувати деякі його параметри (ввімкнути або вимкнути відображення номерів рядків, адрес команд, опкодів; вибрати шрифт відображення інформації; колір відображення еле-

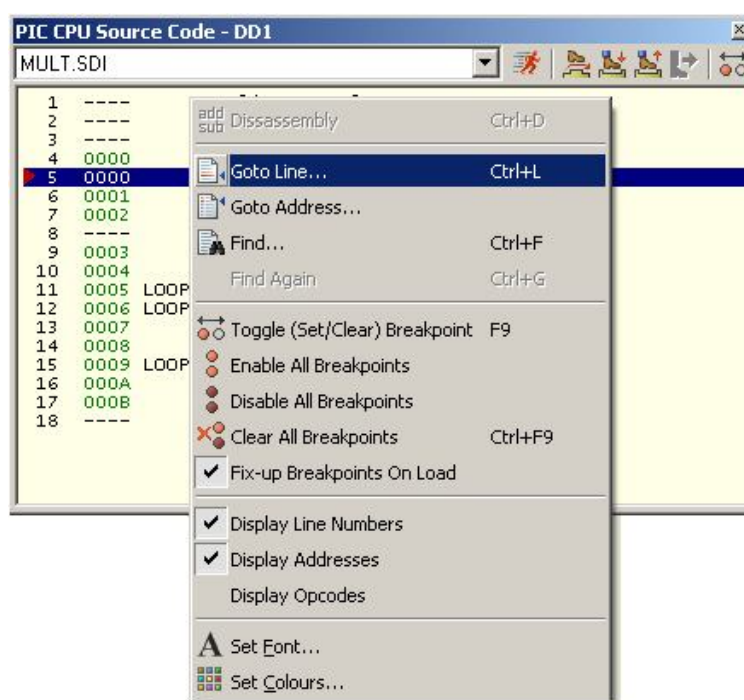


Рис. 23 Контекстне меню вікна налагодження

ментів), знайти потрібний рядок програми по заданій адресі чи порядковому номеру, управляти контрольними точками.

Точки зупинки

Точки зупинки — один з основних інструментів налагодження програми. Це спеціальна мітка інструкції в програмі, дійшовши до якої виконання програми зупиняється. Встановлювати точки зупинки можна лише на інструкціях, що мають об'єктний код.

Виділіть рядок з адресою 0005. Натиснувши на кнопку **Toggle Breakpoint** в правому верхньому кутку вікна налагодження, поставте тут точку зупинки. Дійшовши до неї в процесі симуляції, програма буде автоматично зупинена. Рядок з активною точкою зупинки відмічається червоним овалом. Якщо натиснути на кнопку **Toggle Breakpoint** ще раз, овал стане порожнім — точка зупинки стане пасивною, при третьому натисненні — зникне (точка зупинки знищена).

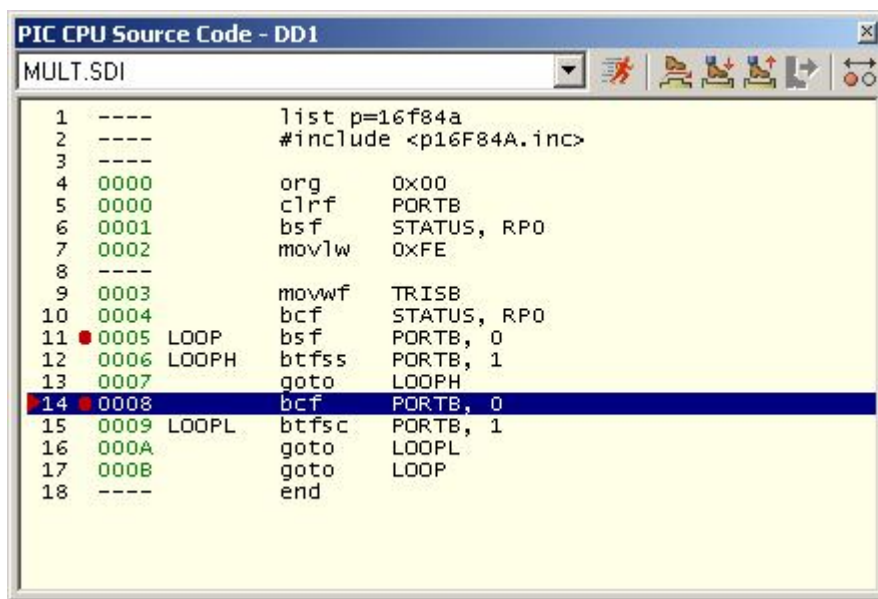


Рис. 24 Встановлення точок зупинки

Встановіть точку зупинки і за адресою 0008. Якщо проаналізувати програму, можна побачити, що задані точки зупинки відповідають моментам зміни логічного рівня на лінії 0 порту В мікроконтролера.

Запустіть модель кнопкою **“Start”**. Пропрацювавши деякий час, вона зупиниться в точці з адресою 0008. Вольтметр показує 2,5 В — точне значення напруги верхньої межі переключення входу мікроконтролера, встановлене в його моделі. Після наступного запуску зупинка відбудеться за адресою 0005, а вольтметр покаже 2,3 В — значення напруги нижньої межі.

Відкриття додаткових вікон

Крім вікна початкового коду, відмітивши в меню **Debug** відповідні пункти, можна вивести на екран вікна спецрегістрів процесора, пам'яті програми, даних, EEPROM і вмісту апаратного стеку. Це дає можливість відслідковувати зміни, що відбуваються в мікроконтролері, при виконанні програми в покроковому режимі. Використовувати додаткові вікна під час неперервної симуляції неможливо. Як тільки симуляція буде ввімкнена, усі активовані додаткові вікна зникнуть, а при призупиненні симуляції — з'являться знову.

Усі додаткові вікна мають контекстне меню, завдяки якому можна управляти відображенням, встановлювати формат даних, які відображаються в вікні, та ін.

Для відслідковування значень регістрів під час неперервної симуляції використовується спеціальне вікно — вікно спостереження (**Watch Window**).

Інші елементи, які містять певні цифрові дані або мають пам'ять, також мають свої інформаційні вікна. Додаткові вікна всіх компонентів, що входять до схеми, можуть бути активовані в меню **Debug** (див. рис.25).

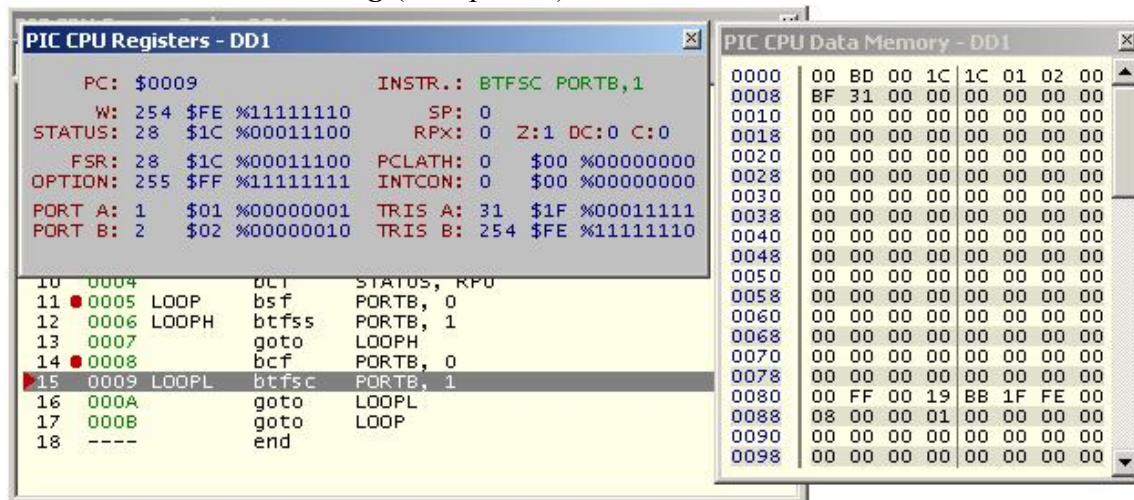


Рис. 25 Додаткові вікна. Вікно спеціальних регістрів. Вікно пам'яті даних

Вікно спостереження

Вікна змінних, пам'яті, регістрів відображаються лише тоді, коли симуляція моделі призупинена. Вікно спостереження дозволяє відстежувати всі значення в режимі реального часу.

Призупиніть симуляцію. Виберіть пункт меню **Debug > Watch Window**. Відкриється порожнє вікно спостереження. Його необхідно заповнити, вказавши імена або адреси регістрів, значення яких будуть відслідковуватися.

Натисніть на полі вікна правою кнопкою миші і, в меню, що з'явилося, виберіть пункт **Add Items (By Name)**.... На екран буде виведений список регістрів мікроконтролера. Знайдіть в списку "PORTB" і натисніть два рази лівою кнопкою миші на його імені. У вікні спостереження з'явиться відповідний рядок. Можна задати спостереження за будь-якою кількістю регістрів, не покидаючи список, а потім закрити його, натиснувши на кнопку **Done**.

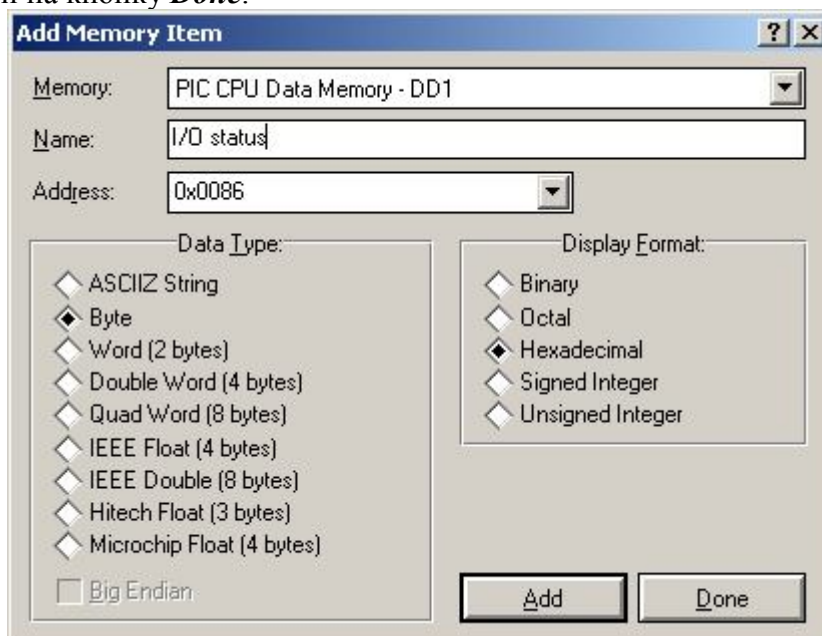


Рис. 26 Додавання елементів у вікно спостереження

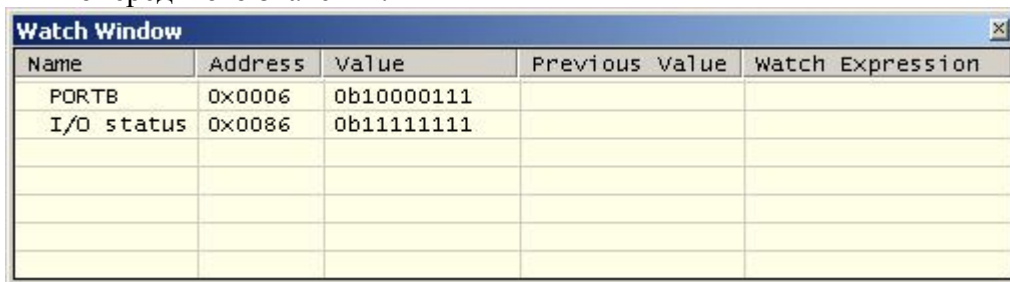
Додайте також регістр TRISB за адресою 0x0086. Для цього натисніть в полі вікна спостереження правою кнопкою миші та виберіть пункт **Add Items (By Address)**... в контекстному меню. В полі **Name** вкажіть назву для цього регістра (назву у вікні

спостереження, назвіть, наприклад “*I/O status*”), в полі *Address* — адресу регістра. Тут також можна встановити тип об’єкту спостереження (байт, слово, подвійне слово та ін.) і формат представлення його значення (двійковий, десятковий, шістнадцятковий та ін.)

Додайте регістр у вікно спостереження, натиснувши кнопку **Add**.

Змінити тип об’єкту і формат представлення для вже доданого регістра можна в його контекстному меню (опції **Data Type** та **Display Format**). Також, контекстне меню дозволяє проводити деякі маніпулятивні дії (перейменування, копіювання, видалення змінних), вмикати або вимикати відображення адрес регістрів, попереднього значення, умов зупинки, налаштовувати шрифт, колір елементів та ін.

Встановіть відображення обох регістрів в двійковій формі та ввімкніть відображення попереднього значення.



Name	Address	Value	Previous Value	Watch Expression
PORTB	0x0006	0b10000111		
I/O status	0x0086	0b11111111		

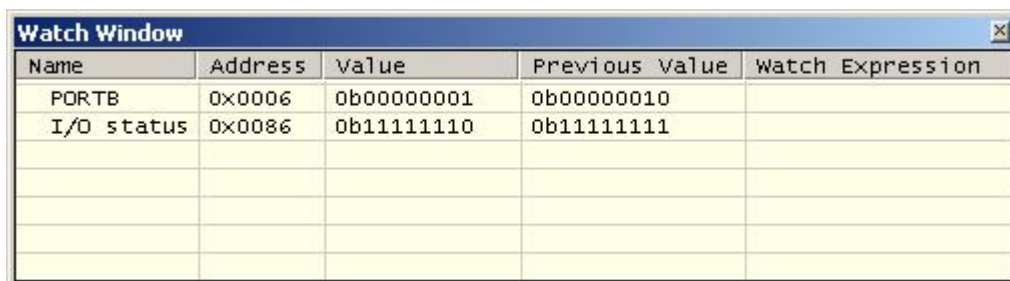
Рис. 27 Вікно спостереження

Початок симуляції

Почніть симуляцію заново (натисніть “**Stop**”, а потім “**Start**”). Вікно спостереження з’явиться з початком симуляції. Значення регістра *I/O status* (TRISB) одразу ж зміниться (вивід RB0 встановиться для роботи на вихід, всі інші — на вхід).

Далі значення цього регістра змінюватись не буде.

Значення двох молодших розрядів порту (PORTB) будуть відображати зміну логічних рівнів на відповідних виводах мікроконтролера.




Name	Address	Value	Previous Value	Watch Expression
PORTB	0x0006	0b00000001	0b00000010	
I/O status	0x0086	0b11111110	0b11111111	

Рис. 28 Вікно спостереження.

Зміна значення регістрів в процесі симуляції

Вікно спостереження дозволяє зупинити симуляцію, коли значення певного його елементу задовольнятиме визначену заздалегідь умову. Найпростіший приклад — зміна значення елементу. Проте, в разі необхідності, можна встановлювати значно складніші умови зупинки.



Name	Address	Value	Previous Value	Watch Expression
PORTB	0x0006	0b		
I/O status	0x0086	0b		

Add Items (By Name)... Alt+N
 Add Items (By Address)... Alt+A
 Watchpoint Condition...

Рис. 29 Встановлення зупинки за умовою

Визначення умов зупинки:

1. Натисніть **CTRL+F12** (команда “*Step*” анімаційної панелі), або призупиніть симуляцію, якщо вона вже виконується.
2. Виберіть команду **Watchpoint Condition** з контекстного меню вікна спостереження.
3. Встановіть глобальну умову зупинки (**Global Break Condition**).
 - **Turn off (disable) watch points** — вимкнути зупинку за умовою.
 - **Suspend the simulation if ANY expression is true** — зупинити симуляцію, якщо виконується будь-яка умова.
 - **Stop the simulation only when ALL expressions are true** — зупинити симуляцію, тільки коли виконуються всі умови.

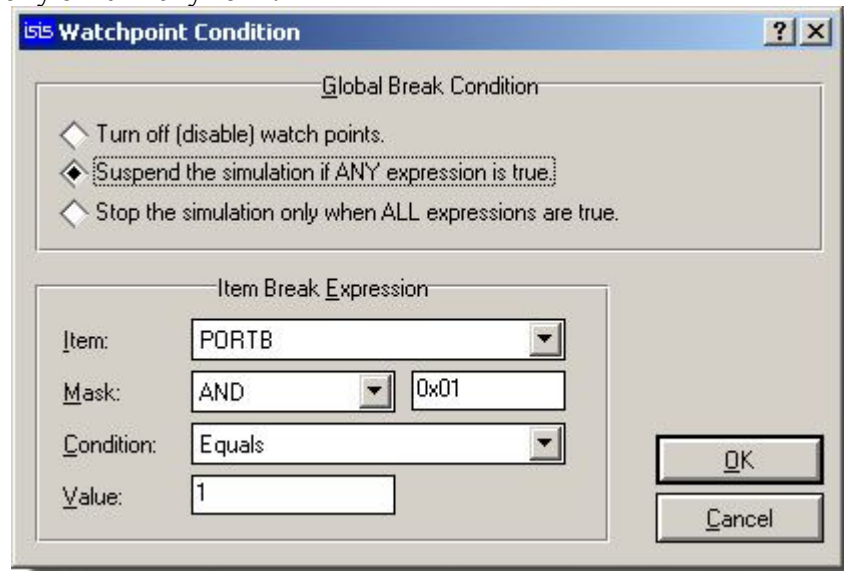


Рис. 30 Вибір умов зупинки

Виберемо другий пункт.

4. Задайте безпосередні умови зупинки — вираз зупинки елемента (*Item Break Expression*)

- **Item** — відслідковуваний елемент, для якого встановлюється умова. Виберіть PORTB.
- **Mask** — маска AND, OR, XOR і значення маски (дозволяє активізувати умову для певних бітів регістра, наприклад). Виберіть AND, значення маски встановіть 0x01.
- **Condition** — вибір умови (зміна значення, дорівнює, не дорівнює, більше ніж, менше ніж і т.д.). Виберіть **Equals** (дорівнює).
- **Value** — значення, з яким порівнюється елемент. Введіть 1.

Натисніть **OK**, щоб завершити встановлення умови зупинки. В результаті, програма буде зупинятись при виконанні умови (PORTB and 1)=1, тобто, при високому рівні на виводі 6 мікроконтролера.

2.4 Моделювання на основі діаграм

Схема, яку ми збираємося моделювати, - аудіо підсилювач на основі 741 операційного підсилювача (ОП), який показано нижче.

Вона демонструє ОП 741 в незвичному включенні: він працює від однополярного джерела напруги 5 В. Резистори зворотного зв'язку R3 і R4 встановлюють коефіцієнт підсилення близько 10. Елементи R1, R2 і C1 встановлюють зсув на неінвертуючому вході і його розділення по постійному струму від входу схеми.

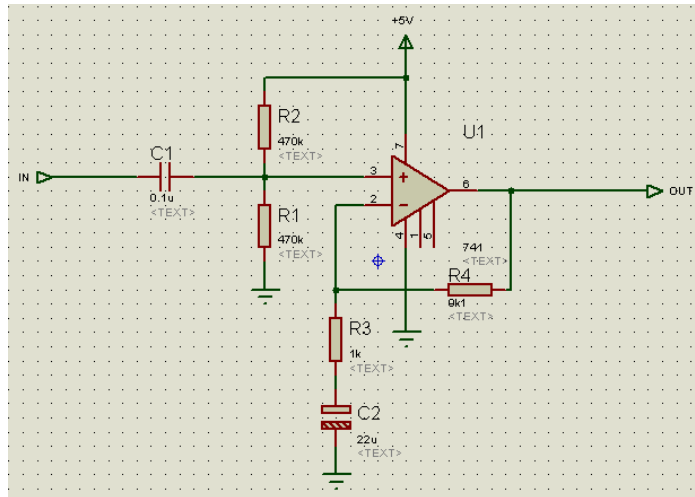


Рис. 31 Схема ввімкнення операційного підсилювача

Зробимо аналіз перехідних процесів схеми. Ця форма аналізу якнайповніша, така, що дає багато інформації про схему. Закінчивши опис моделювання з аналізом перехідних процесів, останні форми аналізу її доповнюватимуть.

Генератори

Щоб перевірити схему, ми повинні забезпечити її відповідним вхідним сигналом. Використовуємо генератор прямокутних імпульсів для нашого випробувального сигналу. Об'єкт «генератор» використовується для отримання необхідного сигналу.

Клацніть на іконі Генератор, перемикач об'єктів покаже список доступних типів генераторів. Для нашого моделювання виберемо генератор імпульсів (**Pulse**). Виберіть тип Pulse, перемістіть курсор миші на вікно редактора, праворуч від клемі IN, і клацніть лівою кнопкою на провіднику, щоб розмістити генератор.

Об'єкти генераторів, як і інші об'єкти ISIS, підтримують операції попереднього перегляду і повороту перед розміщенням, редагування, переміщення, повороту і видалення об'єкту після розміщення.

Генератори можуть бути розміщені на схемі звичайним способом, з подальшим з'єднанням їх провідниками, або, опустивши його на вже наявний провідник. Якщо ви пересунете генератор в інше місце, то ISIS вирішить, що ви хочете від'єднати його, і не перетягне за ним провідник, як він робить це для елементів.

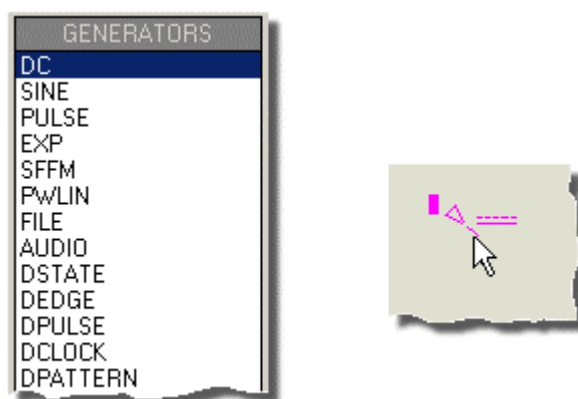


Рис. 32 Вибір генератора

Зверніть увагу, як генератор автоматично встановив зв'язок з клемою IN. Коли генератор під'єднується до об'єкту провідником (або розташовується прямо на наявний провідник), йому встановлюється ім'я вузла схеми, до якого він приєднаний. Якщо у вузла немає імені, тоді, за умовчанням, використовується ім'я виведення найближчого елемента.

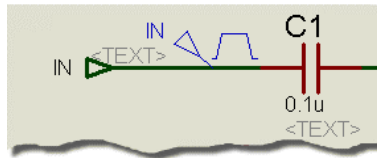


Рис. 33 Встановлення генератора

Налаштуємо генератор, щоб встановити форму імпульсів, яка потрібна. Щоб редагувати генератор, виділіть його правою кнопкою, а потім лівою клацніть на ньому, що зробить доступною діалогову форму редагування генератора (**Edit Generator**). Виберіть поле “Максимальна напруга” (**High Voltage**) і встановіть величину 10 мВ. Також встановіть тривалість імпульсу (**pulse width**) 50 мкс (50u). Натисніть кнопку **OK**, щоб прийняти зміни. Генератори і Пробники дають повний опис властивостей, підтримуваних всіма типами генераторів. Для цієї схеми потрібний лише один генератор, але їх кількість на схемі може бути не обмежена.

Пробники

Визначивши вхід нашої схеми, використовуючи генератор, тепер потрібно розмістити пробники в точках, які потрібно контролювати.

Щоб встановити пробник, клацніть лівою клавішею на іконці “Пробник напруги” (**Voltage Probe**) (переконайтеся, що випадково не вибрали пробник струму).

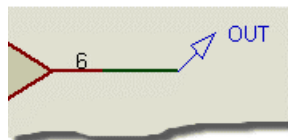


Рис. 34 Встановлення пробника

Пробники можуть встановлюватися на провідники або спочатку розміщені, а потім сполучені, провідниками таким же способом, як і генератори. Перемістіть мишку на вікно редактора, зліва від виводу 3 U1, і клацніть лівою кнопкою, щоб встановити провідник пробника, що сполучає вивід 3 з R1 і R2. Переконайтеся, що встановили пробник на провідник, оскільки не можна його встановлювати на сам вивід.

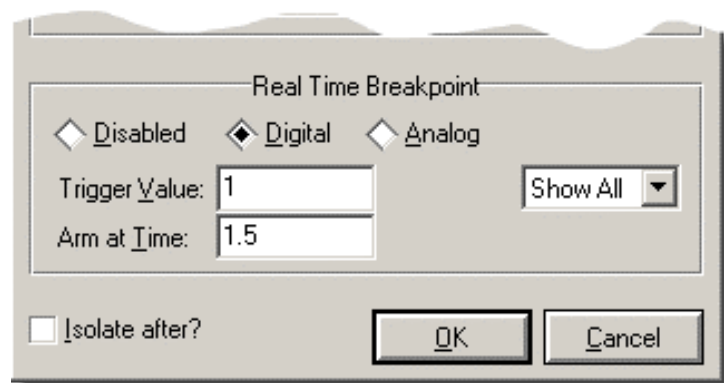


Рис. 35 Встановлення зупинки моделювання від пробника

Зверніть увагу, що він отримав ім'я найближчого елементу, до якого приєднаний, з ім'ям виводу в дужках. Тепер встановіть другий пробник клацанням лівою кнопкою миші, але вже зліва від клемою OUT, на провіднику між вузлом і клемою.

Об'єкти пробників, подібно до генераторів і більшості інших об'єктів ISIS, підтримують процедури попереднього перегляду і орієнтування перед розміщенням, і редагування, переміщення, перевертання і видалення після розміщення.

Пробники можуть редагуватися, як правило, для зміни їх мітки-показчика.

Діаграми

Діаграми відіграють важливу роль в моделюванні: вони не лише діють як засоби демонстрації результатів, але і визначають, яке моделювання виконується.

Розмістивши одну або більше діаграм і вказавши типи даних, які ви передбачаєте побачити на діаграмі (логічні рівні, напруга, імпеданс і так далі), ISIS визнає, якого типу або типів моделювання виконати, і які частини схеми мають бути включені в моделювання.

Тепер, щоб остаточно підготувати схему для моделювання, необхідно розмістити діаграми для відображення результатів.

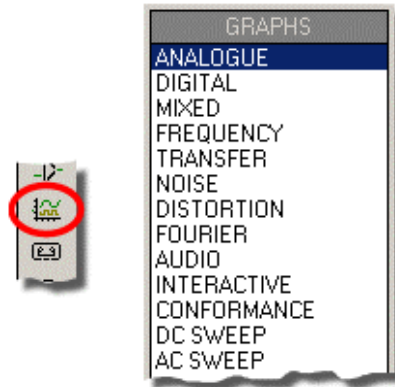


Рис. 36 Встановлення діаграм

Для аналізу перехідних процесів потрібний аналоговий тип діаграми (*Analogue*). Назва “аналоговий” підходить більше, ніж “перехідна характеристика”, для того, щоб відрізнити його від цифрового типу діаграм (*Digital*), який використовується для відображення результатів цифрового аналізу, що є, насправді, спеціалізованою формою аналізу перехідних процесів. Обидва можуть відображатися одночасно, використовуючи змішану діаграму (*Mixed*).

Щоб розмістити діаграму, виберіть іконку Діаграма (*Graph*): перемикач об'єктів покаже список доступних типів діаграм. Виберіть тип Аналоговий (*Analogue*), перемістіть мишку на вікно редактора, натисніть і утримуйте натиснутою ліву кнопку миші, розтягніть прямокутник до відповідного розміру, а потім відпустіть кнопку миші, щоб розмістити діаграму.

Діаграми працюють подібно до інших об'єктів ISIS, хоча вони мають деякі особливості.

Можете виділити діаграму звичайним способом, правою кнопкою миші, а потім (використовуючи ліву кнопку миші) перетягнете одну з міток-маніпуляторів або всю діаграму, щоб змінити її розмір і перемістити діаграму.

Тепер нам потрібно додати генератор і пробники на діаграму. Кожен генератор має асоційований з ним пробник, тому не потрібно розташовувати пробник прямо на генератор, щоб побачити форму коливань на вході.

Є три способи додавання пробників і генераторів на діаграму:

Перший спосіб – по черзі виділяти кожний пробник/генератор, перетягувати їх на діаграму і відпускати там – якраз так само, як би ми переміщали об'єкт.

ISIS виявить, що саме ви намагаєтесь помістити на діаграму, поверне пробник/генератор в первинне положення і додасть запис в діаграму з таким же покажчиком, як і в пробнику/генераторі.

Записи можуть бути асоційовані з правою і лівою осями на аналоговій діаграмі, і пробники/генератори додадуться до осі, ближче до якої вони були розташовані.

Не потрібно вибирати, куди опустити пробник/генератор, новий запис завжди додасться нижче вже створених.

Другий і третій способи додавання пробників/генераторів на діаграму використовують команду “Додати запис” (*Add Trace*) в меню Діаграма (*Graph*); ця команда завжди додає пробники в поточну діаграму (коли є більше ніж одна діаграма; поточна діаграма – діаграма, яка вибрана в даний момент в меню *Graph*).

Якщо активувати команду *Add Trace* без виділених пробників і генераторів, тоді з'явиться діалогова форма “Додати короткий запис”, і пробник може бути вибраний із списку всіх пробників проекту (включаючи пробники на інших схемах).

Якщо пробники/генератори вже виділені, то, активувавши команду *Add Trace*, ви отримаєте пропозицію швидко додати виділені пробники в поточну діаграму; вибір варіанту *NO* активує діалогову форму “Додати короткий запис”, описаний раніше.

Вибір варіанту *YES* додасть всі виділені пробники/генератори на поточну діаграму в алфавітному порядку.

Скористаємося *Quick Add* для додавання наших пробників і генератора на діаграму. Виділіть будь-який пробник або генератор, або, якщо їх більше одного, розтягніть прямокутник виділення довкола всієї схеми – команда *Quick Add* ігнорує всі виділені об'єкти, окрім пробників і генераторів.

Виберіть опцію *Add Trace* з меню *Graph* і відповідайте на нагадування *Yes*. Записи з'являться на діаграмі (ISIS думає, що це поточна діаграма, оскільки він єдиний і використовувався останнім).

Записи складаються з імені (на лівій осі) і порожньої області даних (основне тіло діаграми). Якщо записи не з'явилися на діаграмі, то це, вірогідно, через те, що вона дуже мала, щоб ISIS накреслив їх. Змініть розмір діаграми, виділивши її і потягнувши за кут, щоб зробити більшою.

Тепер ви можете, використовуючи ліву кнопку миші, перетягнути запис вгору або вниз, або редагувати запис (ліvim клацанням без переміщення миші), правою кнопкою видалити запис (тобто видалити її з діаграми). Щоб зняти виділення зі всіх записів, клацніть правою кнопкою миші де-небудь на діаграмі, але не на мітці запису (це виділить або видалить запис).

Залишився **останній етап установки**, який потрібно виконати перед початком моделювання, і це **установка часу виконання моделювання**. ISIS моделює схему стільки часу, скільки вказано на кінці осі «х» діаграми, а для нової діаграми за умовчанням - це одна секунда. Для нашого випадку потрібно, щоб вхідні прямокутні імпульси були досить високої звукової частоти, скажімо, біля 10 кГц. У результаті потрібний період- 100 мкс.

Виділіть діаграму і клацанням лівої кнопки миші викличе діалогову форму редагування діаграми перехідних процесів. Ця форма містить поля, які дозволять створити заголовок діаграми, точно встановити час початку і закінчення моделювання (це відповідає лівій і правій крайнім точкам на осі «х»), промаркувати ліву і праву осі (це не відображується на цифрових діаграмах), так само вказати основні властивості циклу моделювання.

Все, що потрібно – це змінити час зупинки з 1.00 на 100 мкс (можна записати 100 u – ISIS конвертує це в 100E-6) і вибрати **OK**.

Проект тепер готовий для моделювання. На цьому етапі потрібно завантажити нашу версію проекту ("Samples\tutorials\asimtut2.DSN").

Як альтернативу, можна продовжити працювати із схемою, яку ввели самі, і завантажити Asimtut2.DSN лише тоді, коли виникнуть проблеми.

Моделювання

Щоб змоделювати схему, все, що необхідно, це запустити команду «Моделювати» (*Simulate*) в меню *Graph* (або використовувати “гарячу” клавішу клавіатури: **пропуск**).

Команда *Simulate* забезпечить моделювання схеми і оновлення поточної діаграми (яка відмічена в меню *Graph*) відповідно до результатів моделювання.

Рядок стану покаже, як швидко проходить процес моделювання.

Коли моделювання буде завершено, діаграма перемалюється заново з новими даними.

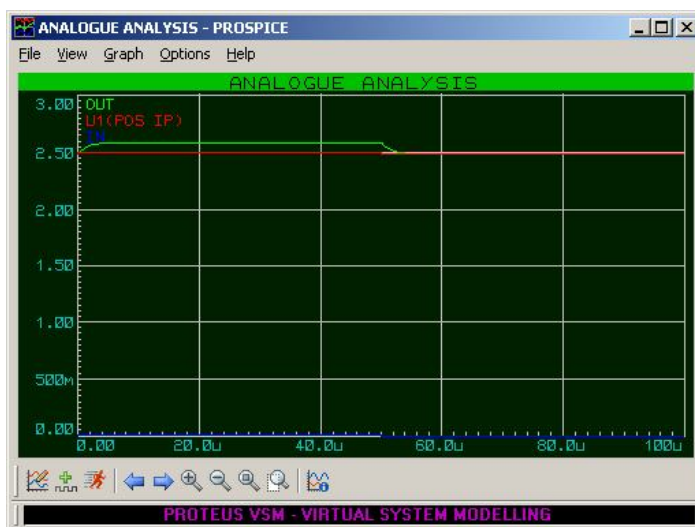


Рис 37 Аналіз перехідних процесів

Для поточної версії ISIS і ядра симулятора початковий час діаграми ігнорується – моделювання завжди починається з нульового моменту і продовжується, доки не досягне часу зупинки, або доки симулятор не досягне стану спокою. Ви можете перервати моделювання в середині натисненням клавіші **ESC**.

Протокол моделювання зберігається для останнього виконаного моделювання.

Можна побачити протокол, використовуючи команду **View Log** в меню **Graph** (або гарячими клавішами Ctrl+"v").

Протокол моделювання надає повну інформацію, яка важкодоступна в записах діаграми. Якщо запустити команду **Simulate** удруге, ви помітите, що моделювання не відбувається.

Якщо з якоїсь причини потрібно моделювати діаграму заново, то потрібно поставити галочку “Завжди моделювати” (**Always Simulate**) на діаграмі в діалоговій формі редагування аналізу перехідних характеристик.

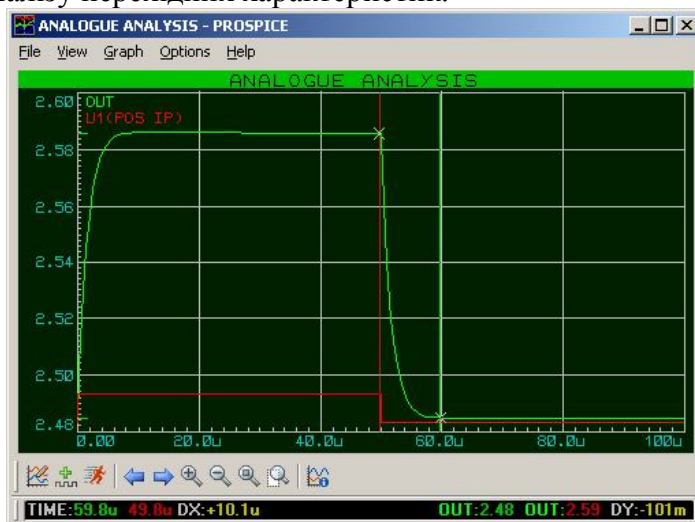


Рис. 38 Вимірювання часу спаду

Якщо є сумніви, що моделювання було насправді, можна перевірити галочку “Протокол списку з'єднань” (**Log Netlist**) в тій же діалоговій формі. Це включить список з'єднань симулятора в протокол моделювання. Перше моделювання завершено.

Використання пробників струму

Закрити вікно діаграми можна звичайним способом або, щоб швидше, натиснути ESC на клавіатурі.

Використовуємо пробник струму, щоб перевірити струм в колі зворотного зв'язку, вимірявши струм через резистор R4.

Пробники струму використовуються подібно до пробників напруги, але з однією важливою відмінністю. Пробник струму потребує напряду, поставленого йому у відповідність. Пробники струму розривають провідник і вставляються в розрив, отже, потрібно знати, в якому напрямі їх розвернути. Це робиться шляхом їх розміщення.

При орієнтації за замовчанням (нахил вправо) в горизонтальному провіднику пробник вимірює силу струму зліва направо. Щоб виміряти силу струму у вертикальному провіднику, необхідно повернути пробник на 90° або 270° . Розміщення пробника на вершині кута – помилка, про яку буде повідомлено при запуску моделювання. Стрілка умовного позначення вказує на напрям струму.

Виберіть пробник струму, клацнувши на іконці "Пробник струму" (**Current Probe**). Клацніть лівою кнопкою на іконці обертання за годинниковою стрілкою, щоб стрілка вказувала вниз. Потім помістіть пробник на вертикальному провіднику між правою стороною R4 і виводом 6 U1. Додайте пробник на праву сторону діаграми виділенням і перетягуванням пробника на правий край згорнутої діаграми.

Натисніть клавішу «Пропуск», щоб заново змодельовати діаграму, і графік з'явиться.

Навіть по згорнутій діаграмі можемо побачити, що струм в колі зворотного зв'язку близький за формою до сигналу на виході, як і передбачалося для операційного підсилювача. Струм змінюється від 10 мкА до 0 на високій і низькій частинах графіка відповідно. Діаграму можна розгорнути до максимуму і розглянути графік детальніше.

Частотний аналіз

Так само, як і аналіз перехідної характеристики, є декілька інших типів аналізу, корисних для моделювання аналогового кола.

Всі вони використовуються майже таким же способом, з діаграмами, пробниками і генераторами, але відрізняються різними варіантами.

Наступний тип аналізу- це частотний аналіз. У частотному аналізі вісь «х» стає частотою (у логарифмічному масштабі), а амплітуда і фаза в досліджуваних точках можуть відображувати на осях "у".

Щоб виконати частотний аналіз, необхідна частотна діаграма. Клацніть лівою кнопкою миші на іконці «Діаграма» (**Graph**), щоб відобразити список типів діаграм в перемикачі об'єктів, і клацніть на типі «Частотний» (**Frequency**). Потім розмістіть діаграму на схемі, як і раніше, перетягуючи прямокутник лівою кнопкою миші. Не потрібно видаляти наявну діаграму перехідної характеристики, але коли це потрібно, щоб збільшити вільний простір, клацніть двічі правою кнопкою на діаграмі, щоб видалити її.

Тепер додамо пробники. Додамо обидва пробники напруги, **OUT** і **U1(POS IP)**. На частотній діаграмі дві осі «у» (ліва і права) мають спеціальне призначення. Ліва вісь «у» використовується, щоб відображати амплітуду досліджуваного сигналу, а права – фазу.

Щоб їх обох бачити окремо, потрібно додати пробники на обидва краї діаграми.

Для цього виділіть пробник OUT і перетягніть його на лівий край діаграми, а потім - на правий. Кожен графік, як завжди, має окремий колір, але вони обидва мають однакові імена.

Тепер виділіть і перетягніть пробник U1(POS IP) лише на лівий край діаграми.

Величини амплітуди і фази мають бути вказані по відношенню до деякої еталонної величини. У ISIS це робиться налаштуванням опорного генератора. Опорний генератор завжди має на виході 0 дБ (1 вольт) на 0° . В якості опорного генератора можна вказати

який-небудь з наявних генераторів. Всі інші генератори кола в частотному аналізі ігноруються.

Щоб встановити генератор IN як опорний в нашому колі, просто виділіть і перетягніть його на діаграму, начебто для того, щоб додати його як пробник. Оскільки це генератор, ISIS передбачить, що ви додасте його як опорний, і виведе в рядку стану повідомлення, підтверджуюче це. Потрібно зробити це без помилок, інакше моделювання працюватиме некоректно.

Редагувати властивості діаграми не потрібно, оскільки частотний діапазон, вибраний за замовчуванням, підходить для наших цілей.

Проте, якщо ви зробите це (наведенням на діаграму і натисненням Ctrl+e), ви побачите, що діалогова форма редагування частотної діаграми мало відрізняється від такої ж форми для перехідної характеристики. Тут не потрібно позначати осі, оскільки їх призначення фіксоване, і тут є “галочка”, яка перемикає відображення графіка амплітуди в децибелах або в абсолютних одиницях. Цю опцію краще встановити в дБ.

Натисніть «Пропуск» (або мишкою над частотною діаграмою), щоб почати моделювання. Коли воно закінчиться, клацніть лівою кнопкою миші на рядку заголовка діаграми, щоб розгорнути її.

Розглянемо спочатку графік амплітуди **OUT**. Можемо побачити, що посилення в смузі пропускання ледве вище 20 дБ (як і очікувалося), а використовуваний діапазон частот приблизно від 50 Гц до 20 кГц. Маркери працюють так само, як і в попередньому прикладі.

Графік фази **OUT** показує вірогідні фазові спотворення на краях частотної характеристики, спад до -90° справа на графіці, на частоті одиничного підсилення.

Ефект фільтру верхніх частот вхідного кола може бути виразно видний, якщо досліджувати графік амплітуди U1(POS IP). Зверніть увагу, що вісь «х» - в логарифмічному масштабі, і щоб прочитати значення з осі, краще використовувати маркери.

Контрольні запитання

1. Назвіть основні елементи головного робочого вікна середовища Proteus VSM.
2. Назвіть основні пристрої середовища Proteus VSM.
3. Як здійснюється вибір потрібного компонента середовища?
4. Як виконується зсув вибраного компонента?
5. Як виконується поворот вибраного компонента?
6. Як виконується з'єднання вибраних компонентів?
7. Як створити початковий файл проекту?
8. Як здійснюється компіляція початкового файлу проекту?
9. Що таке «точки зупинки» і яке їх призначення?
10. Які ви знаєте додаткові вікна середовища?.
11. Які компоненти середовища Proteus VSM відносяться до інтерактивних?
12. Як здійснюється моделювання на основі діаграм?
13. Для чого використовуються діаграми в середовищі Proteus VSM?
14. Які ви знаєте способи створення підсхеми або ієрархічного блоку?
15. Що потрібно виконати для запуску симуляції в проекті?
16. Назвіть основні віртуальні прилади середовища Proteus VSM.

Література

1. Максимов А. Моделирование устройств на микроконтроллерах с помощью программы ISIS из пакета PROTEUS VSM / А. Максимов // Радио. — 2005. — № 4, 5, 6. — С. 30—33, 31—34, 30—32.
2. Цирульник С. М. Застосування програми ISIS пакету Proteus VSM при вивченні курсу «Мікропроцесорна техніка» // С. М. Цирульник, В. К. Задорожний // Матеріали XIII міжнародної конференції з автоматизації управління (Автоматика 2006). — Вінниця: Універсум-Вінниця. — 2007. — С. 526—530. — ISBN 978-966-641-210-5.