

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

Минайленко Р.М., Коноплицька-Слободенюк О.К., Гермак В.С.

Комп'ютерна схемотехніка

*Навчальний посібник
(видання друге, доповнене)*

Кропивницький
2022

УДК 004.75
ББК 32.973.2
М 61

Рекомендовано Вченою радою Центральноукраїнського національного технічного університету, протокол № 8 від 26.04.2022 р

Рецензенти:

доктор технічних наук, професор Одарченко Р.С.

доктор технічних наук, професор Євсєєв С.П.

Минайленко Р.М., Коноплицька-Слободенюк О.К., Гермак В.С.

М 61 Комп'ютерна схемотехніка: навч. посіб. — Кропивницький: Видавець Лисенко В. Ф., 2022. — 153 с.

В посібнику викладено основні поняття та концепції комп'ютерної схемотехніки.

Крім теоретичного матеріалу в посібнику наведено практичні завдання, розв'язання яких відбувається з використанням електронних емуляторів, що дозволяє в процесі складання електронних схем краще засвоїти теоретичний матеріал.

Навчальний посібник призначений для студентів, які навчаються за спеціальністю «Комп'ютерна інженерія», а також може бути корисним для спеціалістів в галузі інформатики, обчислювальної техніки та програмування.

ББК 32.973.2

© Минайленко Р.М., Коноплицька-Слободенюк О.К., Гермак В.С., 2022

© Видавець Лисенко В. Ф., 2022

ЗМІСТ

Вступ	5
Лекція 1	
Основи схемотехніки цифрових пристроїв. Логічні елементи.....	6
1.1 Алгебра логіки.....	6
1.2. Реалізація простих логічних функцій.....	8
Питання для самоконтролю.....	10
Лекція 2	
Тригери.....	11
2.1. Загальні відомості про тригери та їхнє призначення.....	11
2.2. Тригери на логічних елементах.....	11
2.2.1. RS-тригер.....	13
2.2.2. Тригер D-типу (D-тригер).....	14
2.2.3. Тригер T-типу (T-тригер).....	15
2.2.4. JK – тригер.....	15
Питання для самоконтролю.....	17
Лекція 3	
Цифрові мікроелектронні пристрої.....	18
3.1 Поняття про цифрові мікроелектронні пристрої.....	18
3.2 Реалізація складних логічних функцій.....	19
3.3 Дешифратори.....	19
3.4 Мультиплексори.....	21
3.5 Лічильники імпульсів.....	23
Питання для самоконтролю.....	27
Лекція 4	
Цифроаналогові (ЦАП) та аналого-цифрові перетворювачі (АЦП)	28
4.1 Будова і робота ЦАП.....	28
4.2 Будова і робота АЦП.....	32
Питання для самоконтролю.....	34
Лекція 5	
Схемотехніка програмованих цифрових пристроїв. Мікропроцесорні пристрої.....	35
5.1 Загальні положення. Історія процесорів.....	35
5.2 Особливості роботи і використання мікропроцесорів.	
Двійкова система числення. Структура мікропроцесорної системи.....	38
5.3 Структура мікропроцесора.....	41
Питання для самоконтролю.....	42

Лекція 6	
Запам'ятовуючі пристрої.....	43
6.1 Загальні відомості.....	43
6.2 Запам'ятовуючі пристрої для зберігання оперативної інформації.....	44
Питання для самоконтролю.....	46
Лекція 7	
Завдання і виконання програм мікропроцесорного пристрою.....	47
7.1 Алгоритм. Способи представлення алгоритму.....	47
7.2 Програми МПП.....	47
7.3 Шістнадцяткова система числення.....	48
7.4 Асемблер.....	49
Питання для самоконтролю.....	52
Лекція 8	
Програмовані логічні контролери. Мікроконтролери.....	53
8.1 Програмовані логічні контролери.....	53
8.2 Мікроконтролери.....	55
8.3 Структура мікроконтролера.....	58
8.4 Зіставлення основних етапів створення пристроїв керування на основі цифрових автоматів з жорсткою логікою і програмованих.....	60
Питання для самоконтролю.....	60
Система моделювання Electronics Workbench.....	61
1 Структура вікна і система меню.....	61
1.1 Меню File.....	62
1.2 Меню Edit.....	64
2 Створення схем.....	75
2.1 Технологія підготовки схем.....	75
3 Елементна база.....	77
3.1 Джерела струму.....	77
3.2 Індикаторні прилади.....	85
3.3 Комутаційні пристрої.....	91
3.4 Конденсатори.....	95
3.5 Резистори.....	98
3.6 Напівпровідникові діоди.....	100
3.7 Цифрові мікросхеми.....	106
Література.....	110

Вступ

Широке використання мікропроцесорної техніки в усіх сферах людської діяльності, ефективність мікропроцесорних систем пов'язані як із розвитком багатьох суміжних технічних розробок, так і з рівнем підготовки у цій галузі спеціалістів найрізноманітнішого профілю. Відповідність функціональних можливостей мікропроцесорних систем і технологічного призначення пов'язаних з ними об'єктів зумовлюють необхідність відповідної підготовки спеціалістів у далекій від їхніх професійних інтересів галузі.

Задачею викладання є вивчення будови і принципів роботи пристроїв і систем програмної логіки з метою застосування у різноманітних електричних пристроях.

Кінцевим результатом вивчення є набуття вміння оцінювати техніко-економічну ефективність застосування мікропроцесорних пристроїв, формувати технічні вимоги до різного роду пристроїв з мікропроцесорним керуванням, складати алгоритми їхнього функціонування, а також вміння побудови взаємозв'язків між окремими вузлами та пристроями, що об'єднуються мікропроцесорною системою.

Лекція 1

Основи схемотехніки цифрових пристроїв. Логічні елементи

1. Алгебра логіки.
2. Реалізація простих логічних функцій. Логічні елементи.

Основні поняття

Цифрові (логічні) ІМС; математична логіка (алгебра Буля); подія; двійкова (логічна) змінна; двійкова (логічна, перемикальна функція); операція НІ (логічне заперечення, операція інверсії); операція АБО (логічне додавання, диз'юнкція), операція І (логічне множення, кон'юнкція); закони алгебри логіки; тотожності алгебри логіки; правила де Моргана; логічні схеми, логічні елементи; логічні (цифрові) автомати; таблиця істинності; динамічний режим роботи логічних елементів.

1.1 Алгебра логіки

Основу сучасних пристроїв обробки інформації складають цифрові (логічні) інтегральні мікросхеми (ІМС). Аналіз роботи цифрових пристроїв базується на використанні апарату математичної логіки – алгебри Джорджа Буля (1815 – 1864 рр.). В її основі лежить поняття події, що оцінюється з точки зору її настання: вона може статися або не статися. Тоді кожному подію можна вважати істинною, що може моделюватися одиницею "1" (високим рівнем напруги при електричному моделюванні), або хибною – моделюється нулем "0" (низьким рівнем напруги). Обробка інформації, поданої у вигляді подій, проводиться у двійковій системі числення. Вона має тільки дві цифри: 0 і 1.

Величина, яка може набувати тільки ці два значення, називається двійковою (логічною) змінною. Складна подія, що залежить від декількох двійкових змінних, називається двійковою (логічною, перемикальною) функцією

$$y = f(x_0, x_1, \dots, x_n), \quad (0.1)$$

де $x_i = \{1, 0\}$.

Алгебра логіки дозволяє виконувати математичний запис логічних подій і зв'язків між ними, а це дає можливість аналітично описувати будову і роботу цифрових пристроїв (нагадаємо: цифрові пристрої обробляють інформацію, представлену у вигляді сигналів, що змінюються за законом дискретної функції).

В алгебрі логіки є три основних логічних дії (операції, функції):

1) операція НІ – логічне заперечення:

функція має зворотне значення до змінної, від якої вона залежить:

$$y = \bar{x}, \quad (0.2)$$

(читається: у дорівнює не x);

2) операція АБО – логічне додавання (диз'юнкція):

функція істинна, якщо істинна хоча б одна з незалежних змінних, що до

неї входять:

$$y = x_1 + x_2, \quad (0.3)$$

3) операція І – логічне множення (кон'юнкція):

функція істинна, якщо істинні усі незалежні змінні, що до неї входять:

$$y = x_1 \cdot x_2. \quad (0.4)$$

Зверніть увагу: в алгебрі логіки немає операцій віднімання і ділення. Порядок дій в алгебрі логіки такий: перш за все виконується операція НІ, потім І й насамкінець АБО. Для зміни порядку дій, як і в звичайній алгебрі, застосовують дужки.

Для алгебри логіки справедливі закони:

1) переставний (комутативний):

$$x_1 + x_2 = x_2 + x_1 \text{ } \partial \text{ } x_1 \cdot x_2 = x_2 \cdot x_1; \quad (0.5)$$

2) сполучний (асоціативний)

$$x_1 + x_2 + x_3 = x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3 \quad (0.6)$$

$$\text{та } x_1 \cdot x_2 \cdot x_3 = x_1 (x_2 \cdot x_3) = (x_1 \cdot x_2) x_3;$$

3) розподільний (дистрибутивний)

$$x_1 (x_2 + x_3) = x_1 \cdot x_2 + x_1 \cdot x_3. \quad (0.7)$$

Зазначимо, що такі самі закони діють і в звичайній алгебрі.

Тотожності алгебри логіки:

$$x + x = x \text{ } \partial \text{ } x \cdot x = x, \quad (0.8)$$

$$x + \bar{x} = 1 \text{ } \partial \text{ } x \cdot \bar{x} = 0, \quad (0.9)$$

$$x + 0 = x \text{ } \partial \text{ } x \cdot 0 = 0, \quad (0.10)$$

$$\overline{\overline{x}} = x, \quad (0.11)$$

$$x_1 + x_1 \cdot x_2 + x_1 \cdot x_3 = x_1, \quad (0.12)$$

$$x_1 + \overline{x_1 \cdot x_2} = x_1 + x_2, \quad (0.13)$$

$$x_1 \cdot x_2 + \overline{x_1 \cdot x_2} = x_1, \quad (0.14)$$

$$x_1 (x_1 + x_2) = x_1, \quad (0.15)$$

$$x_1 (\overline{x_1} + x_2) = x_1 \cdot x_2 \quad (0.16)$$

$$(x_1 + x_2)(x_1 + x_3) = x_1 + x_2 \cdot x_3. \quad (0.17)$$

До основних законів алгебри логіки також належать закони інверсії для логічних додавання та множення – правила де Моргана:

$$\overline{x_1 + x_2} = \overline{x_1} \cdot \overline{x_2} \text{ } \partial \text{ } \overline{x_1 \cdot x_2} = \overline{x_1} + \overline{x_2}. \quad (0.18)$$

Ці закони відіграють важливу роль при синтезі схем цифрових пристроїв, часто призводячи до суттєвого спрощення логічних функцій, а отже, і схем пристроїв, що їх реалізують.

Взагалі, знання законів алгебри логіки дозволяє отримувати оптимальну за заданими критеріями схему пристрою, що забезпечує використання мінімального числа уніфікованих елементів, високу швидкодію та надійність.

У кінці зазначимо, що у практичних цілях алгебру Буля першим у 1938 році застосував один із родоначальників математичної теорії інформації і кібернетики Клод Шеннон (США) при дослідженні електричних кіл з

контактними перемикачами.

1.2 Реалізація простих логічних функцій

Логічні елементи

Практична реалізація аналітичного опису подій алгебри логіки виконується у вигляді логічних схем, що будуються з логічних елементів як логічні (цифрові) автомати. При цьому проектувальника, як правило, не цікавить внутрішня будова логічних елементів. Вони розглядаються як функціональні вузли обробки цифрової інформації.

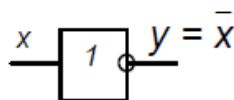


Рисунок 1.1 – Елемент НЕ

Отже, однією з найпростіших логічних функцій є функція заперечення НЕ, яку ще називають операцією інверсії.

Графічне позначення елемента, що реалізує таку функцію, наведене на рис. 1.1.

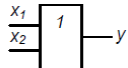
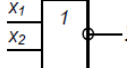
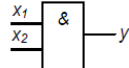
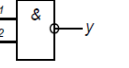
Функцію, що її виконує логічний пристрій, для полегшення сприйняття часто представляють у вигляді таблиці, яку називають таблицею істинності. Кількість стовпців цієї таблиці дорівнює числу змінних, що входять до функції, і є ще один стовпець, у якому вказують значення функції для кожної з можливих комбінацій вхідних змінних, а їхньому числу відповідає кількість рядків таблиці. У загальному випадку кількість рядків дорівнює 2^n , де n – число змінних.

Об'єднана таблиця істинності деяких основних логічних функцій, що залежать від двох змінних (усього таких функцій шістнадцять), наведена в табл. 1.1. На рисунку також представлено інформацію про математичний запис функцій, їхні назви, графічне позначення.

З таблиці видно, що, наприклад, функція АБО істинна (дорівнює 1), якщо істинною є хоча б одна із змінних, що до неї входять, а функція І – тільки у випадку істинності обох змінних.

Таблиця 1. 1 – Деякі логічні функції двох змінних

Вхідні змінні		Функція у			
x_1	x_2	АБО	АБО-НЕ	І	І-НЕ
0	0	0	1	0	1
0	1	1	0	0	1
1	0	1	0	0	1
1	1	1	0	1	0

Математичний запис (формула)	$y = x_1 + x_2 = x_1 \vee x_2$	$y = \overline{x_1 + x_2}$	$y = x_1 \cdot x_2 = x_1 \wedge x_2$	$y = \overline{x_1 \cdot x_2}$
Назва функції	Логічне додавання (диз'юнкція) – функція АБО	Заперечення логічного додавання (стрілка Пірса) – функція АБО-НІ	Логічне множення (кон'юнкція) – функція І	Заперечення логічного множення (штрих Шеффера) – функція І-НІ
Графічне позначення елемента, що реалізує функцію				

Якщо вхідні сигнали змінювати з часом, як це показано на рис. 1.2, можна отримати часові діаграми, що відповідають динамічному режиму роботи логічних елементів.

У загальному випадку кількість вхідних змінних (кількість входів) логічних елементів, необхідних для реалізації складних логічних функцій, може бути будь-якою. Реально у елементів, що випускаються у вигляді ІМС, вона, як правило, становить 2 (чотири елементи в одному корпусі ІМС, що мають спільні кола живлення), 3 (три елементи), 4 (два елементи), 8 (один елемент). Частіше це елементи І-НІ, АБО-НІ.

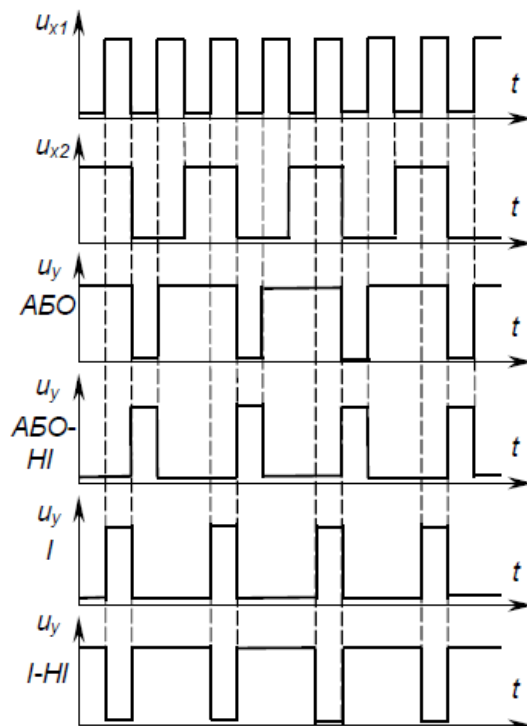


Рисунок 1.2 – Часові діаграми роботи деяких двовходових логічних елементів

Питання для самоконтролю

1. Поясніть, на чому базується аналіз роботи цифрових пристроїв?
2. Зазначте, як оцінюють подію в алгебрі логіки?
3. Поясніть, як можна моделювати події алгебри логіки?
4. Поясніть, що таке логічна (двійкова) змінна, логічна функція?
5. Які найпростіші логічні функції ви знаєте?
6. Наведіть можливі способи реалізації простих логічних функцій на елементах електроніки.
7. Складіть таблицю істинності для функції $y = \bar{a} \cdot b + \bar{b} \cdot c$.
8. Виконайте схемну реалізацію функції $y = \bar{a} \cdot b + \bar{b} \cdot c$ у базисі I, АБО, НІ.

Лекція 2

Тригери

1. Загальні відомості про тригери та їхнє призначення.
2. Тригери на логічних елементах.
3. RS-тригер.
4. D-тригер.
5. T-тригер.
6. JK-тригер.

Основні поняття

Тригери; тригери в інтегральному виконанні; вихід прямий; вихід інверсний; вхід інформаційний; вхід тактовий (синхронізуючий); вхід прямий; вхід інверсний; вхід потенціальний; вхід імпульсний; тригери асинхронні, тригери синхронні; таблиця переходів; стан тригера (нульовий, одиничний, невизначений); двовходовий асинхронний RS-тригер з прямими й інверсними входами; D-тригер; T-тригер; JK – тригер.

2.1 Загальні відомості про тригери та їхнє призначення

Основою послідовнісних логічних пристроїв (пристроїв із пам'яттю) є тригери. Тригер забезпечує запам'ятовування елементарного обсягу дискретної інформації – 1 біт.

Тригери (від англійського *trigger* – заціпка) – це спускові імпульсні пристрої з позитивним зворотним зв'язком, що мають два сталих стани рівноваги і можуть переходити із одного стану в інший під дією сигналу, який перевищує за значенням деякий рівень – поріг спрацювання пристрою.

Використовуються тригери для таких цілей:

- 1) перетворення імпульсу довільної форми у прямокутну, тобто застосовуються як формувачі імпульсів прямокутної форми (тригери Шмітта);
- 2) створення електронних реле;
- 3) створення пристроїв підрахунку імпульсів і ділення частоти надходження імпульсів;
- 4) зберігання інформації у двійковому коді.

2.2 Тригери на логічних елементах

Тригери в інтегральному виконанні будуються з простих логічних елементів типу АБО-НІ, І-НІ. Як правило, мікросхема вміщує 1–4 тригери із спільними колами живлення, а інколи і спільними колами синхронізації або керування.

У загальному випадку тригери складаються з логічного пристрою керування та власне тригера як елемента пам'яті. Є велика кількість

різноманітних схем тригерів з різними функціональними можливостями .
Узагальнена структурна схема тригера зображена на рис. 2.1.

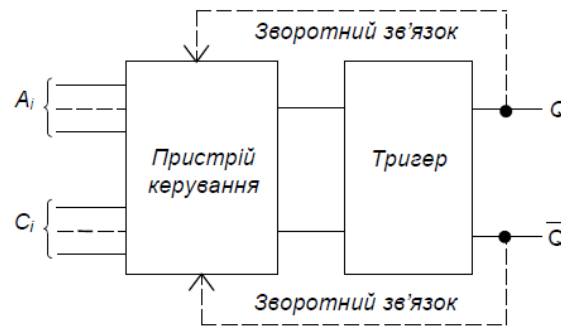


Рисунок 2.1 – Структурна схема тригера

Пристрій керування призначений для перетворення сигналів, що надходять до входів A_i , у вигляді, придатний для керування власне тригером, що виконує функцію елемента пам'яті.

Тригер має два виходи: Q – прямий (одиничний), \bar{Q} – інверсний (нульовий).

Входи A_i називаються інформаційними, а входи C_i – тактовими, або синхронізувальними.

За способом занесення інформації тригери поділяються на асинхронні, що змінюють свій стан одразу після надходження сигналу на певний інформаційний вхід, і синхронні, які спрацьовують не лише за наявності сигналів на інформаційних входах, а лише після надходження синхронізувального (тактового) сигналу на певний вхід синхронізації.

Описують роботу тригерів (і послідовнісних пристроїв взагалі) також за допомогою логічних функцій або частіше задля наочності за допомогою таблиць переходів. У таблицях зазначають усі можливі комбінації сигналів на інформаційних входах у даний момент часу t_i і стан, в який перейде тригер під дією цих сигналів у наступний момент часу t_{i+1} . Причому наступний момент часу у асинхронного тригера настає одразу після зміни комбінації сигналів на інформаційних входах, а у синхронного – після надходження тактового сигналу (як правило, це імпульс) на відповідний вхід синхронізації.

Стани тригера в таблицях переходів, як правило, позначають так:

0 – тригер має сигнал на виході $Q = 0$ (нульовий стан) незалежно від сигналів на входах;

1 – тригер має сигнал на виході $Q = 1$ (одиничний стан) незалежно від сигналів на входах;

Q_i – стан тригера не змінюється при зміні сигналів на входах;

\bar{Q}_i – стан тригера змінюється на протилежний при зміні сигналів на входах;

X – невизначений стан тригера, коли він після зміни сигналів на входах рівноможливо може опинитися в нульовому ($Q = 0$) або в одиничному ($Q = 1$)

стані.

Стверджують, що навіть за найпростішої конфігурації тригерного пристрою, яка має один інформаційний вхід і два виходи, можна отримати 25 функціональних різновидів тригерів. При двох входах їх буде вже 625. Практично ж застосовують 6 - 8 типів.

Найбільш поширені з них *RS* - тригери, *D* - тригери, *T* - тригери, *JK*-тригери. Часто тригери будують як комбіновані: *RSD* - тригер, *RST* - тригер і т.п.

2.2.1 *RS*-тригер

Умовні позначення двовходових асинхронних *RS* - тригерів з прямими (такими, що реагують на наявність 1) і інверсними (такими, що реагують на наявність 0) входами наведено на рис. 2.2.

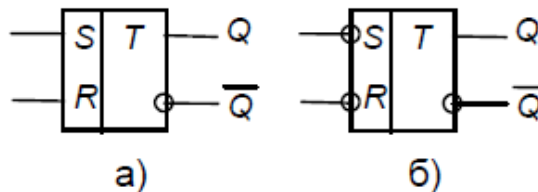


Рисунок 2.2 – Умовні позначення *RS*-тригера з прямими (а) й інверсними (б) входами

Свою назву *RS*-тригер одержав від перших літер англійських слів *set* – встановлювати (*S*) та *reset* – відновлювати (*R*).

S – інформаційний вхід призначений для установаження тригера в одиничний стан ($Q=1$), а *R*– вхід призначений для повернення тригера у нульовий стан ($Q=0$).

Роботу тригерів описують відповідні таблиці переходів, наведені в табл. 2.1.

Таблиця 2.1–Таблиці переходів *RS*-тригерів

а) з прямими входами

t_i		t_{i+1}
<i>S</i>	<i>R</i>	<i>Q</i>
1	0	1
0	1	0
0	0	Q_i
1	1	<i>X</i>

б) з інверсними входами

t_i		t_{i+1}
<i>S</i>	<i>R</i>	<i>Q</i>
1	0	0
0	1	1
1	1	Q_i
0	0	<i>X</i>

Схеми таких *RS*-тригерів, побудованих на елементах І-НІ та АБО-НІ,

зображені на рис. 2.3.

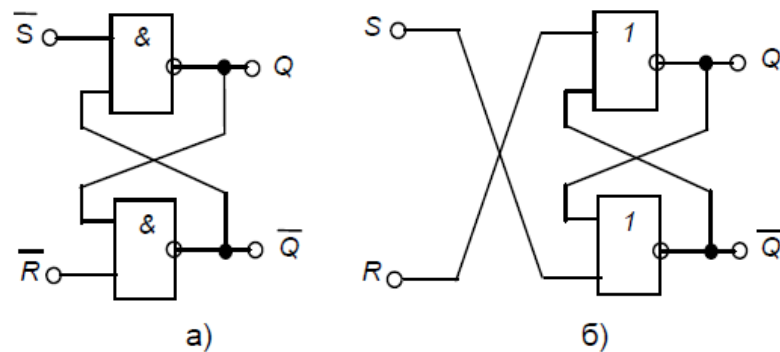


Рисунок 2.3 – RS -тригер з інверсними входами на елементах І-НІ (а) та з прямими входами на елементах АБО-НІ (б)

Схема і умовне позначення синхронного RS -тригера з прямими входами, побудованого на елементах І-НІ, наведені на рис. 2.4.

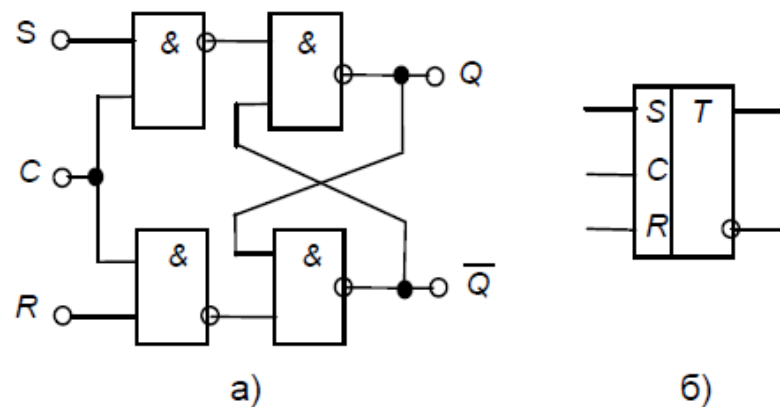


Рисунок 2.4 – Синхронний RS -тригер

Необхідно зазначити, що тактові входи бувають потенціально прямі, як у даному випадку (тригер змінює свій стан при надходженні сигналу 1 на вхід C), та інверсні (тригер змінює стан при надходженні сигналу 0), або імпульсні, також прямі й інверсні (коли тригер змінює свій стан при зміні сигналу на тактовому вході з 0 на 1 або з 1 на 0 відповідно).

2.2.2 Тригер D -типу (D -тригер)

D -тригер (від англійського *delay* – затримка) має два входи: D – інформаційний та C – тактовий (синхронізуючий): D -тригер синхронний. А це означає, що інформація, яка надходить на вхід D , запам'ятовується лише при надходженні синхронізуючого імпульсу на вхід C , тобто із затримкою на час надходження останнього. Тому D -тригер ще називають тригером затримки.

Умове позначення D -тригера з прямим імпульсним входом синхронізації та таблиця переходів наведені на рис. 2.5, а часові діаграми його

роботи – на рис. 2.6.

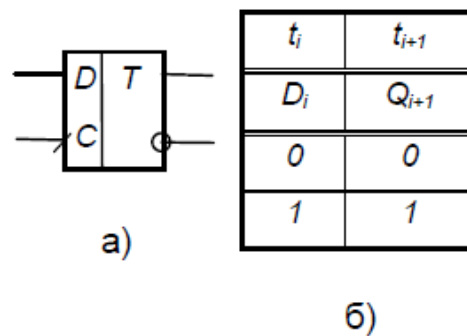


Рисунок 2.5 – Умовне позначення (а) і таблиця переходів (б) D -тригера

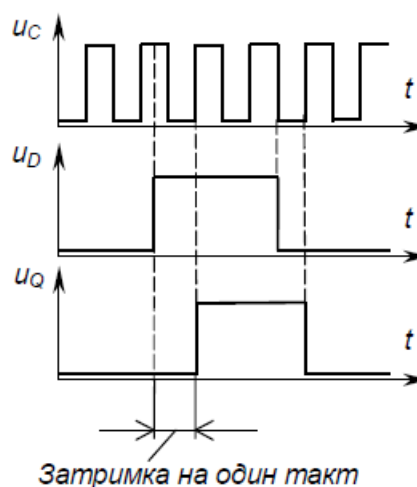


Рисунок 2.6 – Часові діаграми роботи D -тригера

2.2.3 Тригер Т-типу (Т-тригер)

Т-тригер (від англійського *toggle* – перекидатись) ще називають тригером-поділювачем на два, або лічильним тригером. Тригер має тільки один тактовий вхід, а його стан змінюється на протилежний з надходженням на вхід кожного імпульсу. Цей тригер використовують для лічення та ділення частоти імпульсів.

Умовне позначення та часові діаграми роботи Т-тригера з інверсним імпульсним входом наведені на рис. 2.7.

2.2.4 JK - тригер

Синхронний JK -тригер має два інформаційних входи J і K та тактовий C . Умовне позначення та таблиця переходів JK -тригера з прямим імпульсним тактовим входом наведені на рис. 2.8.

JK -тригер є універсальним, оскільки він може виконувати роль RS -тригера, якщо використовувати вхід J як S , а K – як R (при цьому таблиця переходів RS - тригера відповідає першим трьом рядкам таблиці переходів JK -

тригера). Якщо задати одиниці на обох інформаційних входах, JK -тригер стає T -тригером.

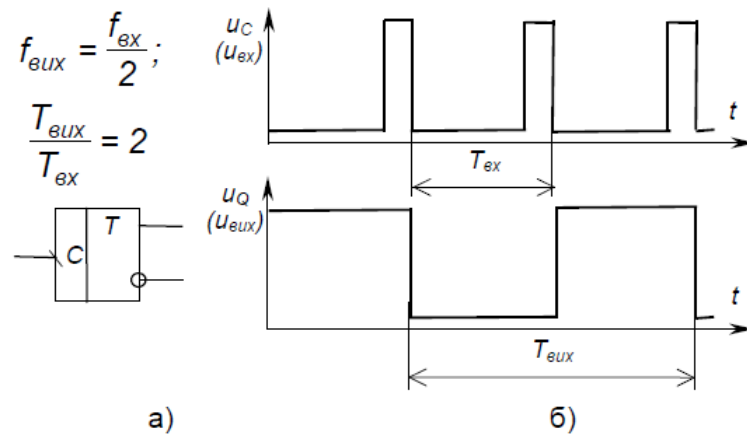


Рисунок 2.7 – Умовне позначення (а) та часові діаграми роботи (б) T -тригера

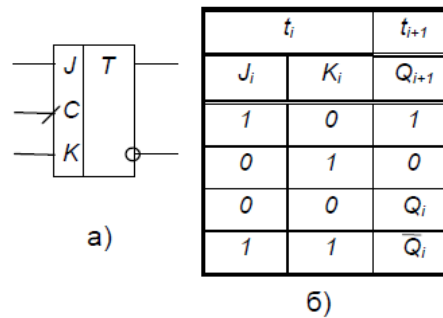


Рисунок 2.8 – Умовне позначення (а) та таблиця переходів (б) JK -тригера

Схеми використання JK -тригера як T -тригера та D -тригера зображені на рис. 2.9.

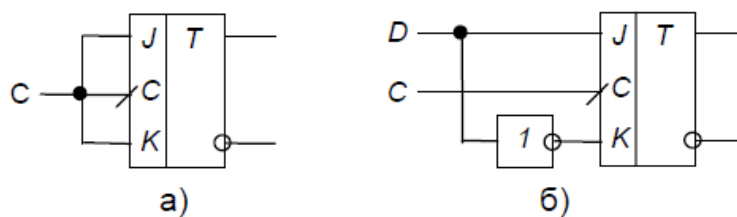


Рисунок 2.9 - Використання JK -тригера (а) та D -тригера (б) як T -тригера

Питання для самоконтролю

- 1. Поясніть, що таке тригери і для чого вони можуть бути застосовані?*
- 2. Наведіть узагальнену структурну схему тригера, побудованого на логічних елементах.*
- 3. Поясніть різницю між асинхронним і синхронним тригерами.*
- 4. Поясніть, як описують роботу тригерів? Наведіть приклади.*
- 5. Поясніть значення термінів "прямий вхід", "інверсний вхід", "потенціальний вхід", "імпульсний вхід".*
- 6. Наведіть умовне позначення, таблиці переходів і схеми RS-тригерів з прямими й інверсними входами, побудованих на логічних елементах.*
- 7. Наведіть схему і поясніть роботу синхронного RS-тригера, побудованого на логічних елементах І-НІ.*
- 8. Наведіть умовні позначення, таблиці переходів та поясніть роботу D-тригера, T-тригера, JK-тригера.*
- 9. Поясніть, чому JK- тригер вважають універсальним? Наведіть та поясніть приклади його використання для побудови тригерів інших типів.*
- 10. Як за допомогою D-тригера побудувати T-тригер?*

Лекція 3

Цифрові мікроелектронні пристрої

1. Поняття про цифрові мікроелектронні пристрої.
2. Реалізація складних логічних функцій.
3. Дешифратори.
4. Мультиплексори.
5. Лічильники імпульсів.

Основні поняття

Цифрові мікроелектронні пристрої; комбінаційні пристрої; послідовнісні пристрої; функціонально повна система логічних елементів (базис); дешифратор; двійковий дешифратор; двійково-десятковий дешифратор; двійково-семисегментний дешифратор; мультиплексор; лічильник імпульсів (простий, віднімаючий, реверсивний, послідовний, паралельний, двійково-десятковий); послідовний двійковий код; двійково-десятковий код; регістр.

3.1 Поняття про цифрові мікроелектронні пристрої

Цифрові мікроелектронні пристрої являють собою дискретні цифрові автомати, виконані на ІМС і призначені для обробки інформації, що представлена у вигляді цифрового коду. Вони використовуються для створення цифрових інформаційних, вимірювальних систем та систем керування.

Усі цифрові пристрої поділяються на два великих класи: комбінаційні і послідовнісні.

Комбінаційні пристрої реалізують функції, які залежать тільки від комбінації змінних, що до них входять, у даний момент часу і не залежать від стану пристрою у попередній момент часу. Найпростішими прикладами таких пристроїв є логічні елементи.

Послідовнісні (від слова "послідовність") пристрої реалізують функції, що залежать не лише від комбінації входних змінних у даний момент часу, а ще й від стану пристрою у попередній момент часу: вони мають пам'ять. Найпростішими прикладами таких пристроїв є тригери.

Основними (найбільш застосовуваними) мікроелектронними цифровими пристроями є:

- 1) дешифратори;
- 2) мультиплексори;
- 3) лічильники імпульсів;
- 4) регістри;
- 5) цифроаналогові та аналого-цифрові перетворювачі.

Будуються ці пристрої на логічних елементах і тригерах.

3.2 Реалізація складних логічних функцій

Складні логічні функції реалізують на ІМС простих логічних елементів.

Мінімальний набір логічних елементів, що реалізують деякі прості логічні функції і за наявності необмеженої кількості яких можна реалізувати наскільки завгодно складну логічну функцію, називають функціонально повною системою логічних елементів, або базисом.

Найбільш відомими функціонально повними системами є:

- 1) елементи, що реалізують функції алгебри Буля – І, АБО, НІ;
- 2) елемент, що реалізує функцію штрих Шеффера – І-НІ;
- 3) елемент, що реалізує функцію стрілка Пірса – АБО-НІ.

Якщо уважно подивимося на таблицю істинності логічних елементів (див. табл. 1.1), то побачимо, що, наприклад, елемент І для одиниць є елементом АБО для нулів. Тобто якщо для прямих значень

$$y = x_1 \cdot x_2, \text{ то для інверсних } \overline{y} = \overline{x_1} + \overline{x_2}. \quad (3.1)$$

Для елемента І-НІ можна записати

$$y = \overline{x_1 \cdot x_2} = \overline{x_1} + \overline{x_2}, \quad (3.2)$$

а для елемента АБО-НІ –

$$y = \overline{\overline{x_1} + \overline{x_2}} = \overline{x_1} \cdot \overline{x_2}. \quad (3.3)$$

Вирази (3.2) і (3.3) називають правилами де Моргана. Якщо задати, наприклад, x_2 таким, що дорівнює 1 або 0, то матимемо

$$y = \overline{x_1 \cdot x_2} = \overline{x_1 \cdot 1} = \overline{x_1}, \quad (3.4)$$

$$\text{або} \quad y = \overline{\overline{x_1} + \overline{x_2}} = \overline{\overline{x_1} + 0} = \overline{x_1}. \quad (3.5)$$

Тобто ми можемо використовувати багатовходові логічні елементи з інверсією на виході як інвертори, задаючи на всіх, крім одного, входах сигнал 1 (*const 1*) або 0 (*const 0*).

Якщо на всі входи цих елементів подати одну й ту саму змінну, також матимемо інвертори:

$$y = \overline{\overline{x_1} \cdot \overline{x_1}} = \overline{x_1}, \quad y = \overline{\overline{x_1} + \overline{x_1}} = \overline{x_1}. \quad (3.6)$$

Коли кількості входів конкретного логічного елемента не вистачає, можна вчинити так: якщо, наприклад, потрібен тривходовий елемент, а маємо тільки двовходові, то тривходовий можна реалізувати на основі таких співвідношень.

Оскільки очевидно, що

$$\overline{\overline{x}} = x, \quad (3.7)$$

$$\text{то } y = \overline{x_1 \cdot x_2 \cdot x_3} = \overline{\overline{\overline{x_1 \cdot x_2 \cdot x_3}}} = \overline{\overline{x_1 \cdot x_2} \cdot \overline{1 \cdot x_3}}, \quad (3.8)$$

$$\text{або } y = \overline{x_1 \cdot x_2 \cdot x_3} = \overline{\overline{\overline{x_1 \cdot x_2 \cdot x_3}}} = \overline{\overline{x_1 + x_2 + 1 + x_3}}. \quad (3.9)$$

Отже, реалізувати тривходовий елемент можна на трьох двовходових,

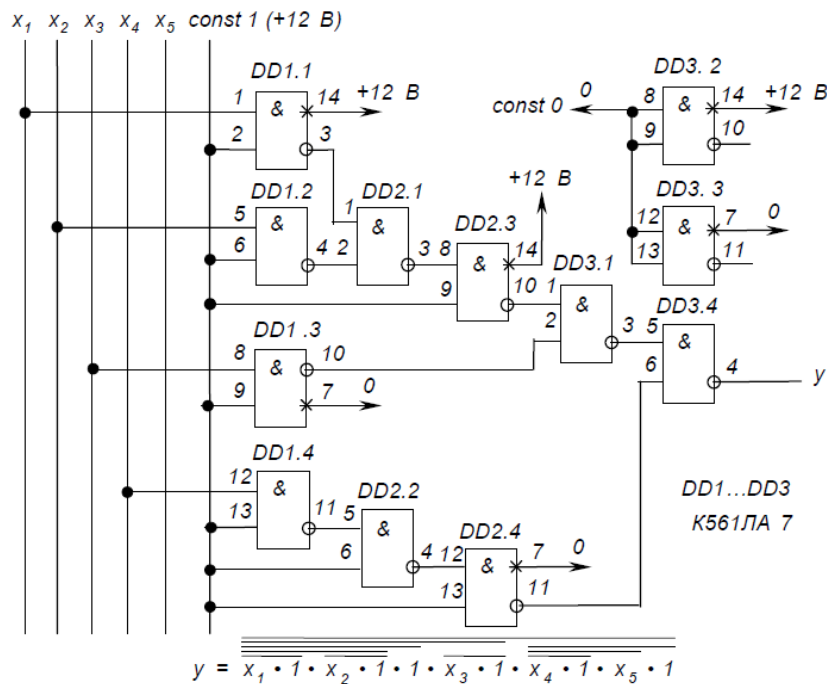


Рисунок 3.2 – Схемна реалізація логічної функції на однотипних елементах

Зауважимо, що на таких схемах відносно ланцюгів живлення ІМС на вільному полі в правій частині схеми в технічних вимогах дають вказівки щодо під'єднання відповідних контактів або називають відповідні контакти ІМС як «нелогічні» виводи елементів і позначають місця їх під'єднання (див. рис. 3.2 – контакти 7 і 14). На входи елементів, що не задіяні, подають сигнали *const 1* або *const 0*, а їхні виходи залишають вільними (див. рис. 3.2 – елементи DD3.2 і DD3.3).

3.3 Дешифратори

Дешифратори (декодери) є комбінаційними пристроями, призначеними для перетворення кодованих двійкових входних сигналів у сигнали керування виконавчими пристроями, пристроями відображення інформації і т.п.

У загальному випадку дешифратор має декілька входів (за кількістю розрядів двійкових чисел, які необхідно декодувати) і декілька виходів.

Кожній комбінації входних сигналів відповідає певна комбінація вихідних (зрозуміло, що дешифратори, як комбінаційні пристрої, будуються на логічних елементах і їх випускають у вигляді ІМС).

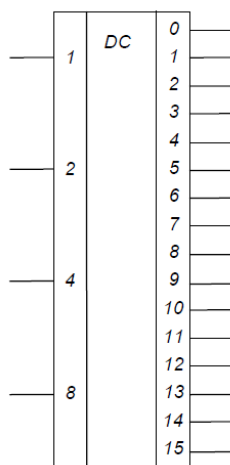


Рисунок 3.3 – Двійковий дешифратор

Наприклад, двійковий дешифратор, умовне позначення якого наведене на рис. 3.3, має чотири входи ($n=4$) з ваговими коефіцієнтами 1, 2, 4, 8, що відповідає чотирьом розрядам послідовного двійкового коду ($2^0, 2^1, 2^2, 2^3$), і шістнадцять виходів: від 0 до 15 ($N=2^n=2^4=16$). Кожній комбінації нулів і одиниць на входах відповідає одиниця на відповідному виході.

Наприклад,

$$y_5 = x_1 \cdot \overline{x_2} \cdot x_4 \cdot \overline{x_8},$$

$$(5=1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3). \quad (3.12)$$

У двійково-десятькового дешифратора з інверсними виходами, умовне позначення якого наведене на рис. 3.4, кожній з перших десяти двійкових комбінацій (двійково-десятьковий код) відповідає нуль на відповідному виході. Такі дешифратори у вигляді ІМС застосовують для керування десятковими неоновими індикаторами, у яких десяткові знаки являють собою фігурні катоди неонові лампи.

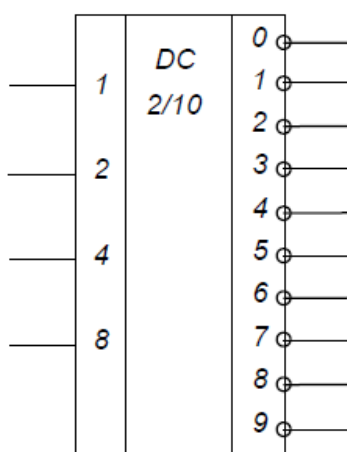


Рисунок 3.4 – Двійково-десятьковий дешифратор

Двійково-семисегментний дешифратор, зображений на рис. 3.5 а, перетворює двійкову комбінацію входних сигналів у комбінацію вихідних, необхідну для вмикання відповідної комбінації сегментів семисегментного

десятькового індикатора – рис. 3.5 б.

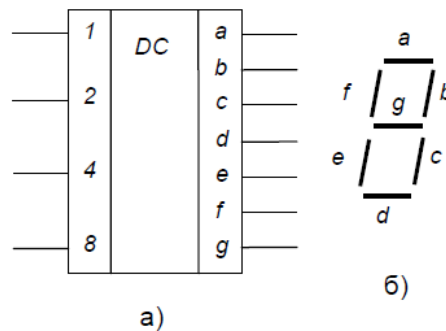


Рисунок 3.5 – Двійково-семисегментний дешифратор (а) і схема розміщення сегментів індикатора (б)

3.4 Мультиплексори

Мультиплексори (комутатори) – це комбінаційні пристрої, які підключають до виходу вхід (передають на вихід інформацію із входу), номер якого задає комбінація нулів і одиниць на адресних входах.

Схема чотиривходового мультиплексора і його таблиця істинності наведені на рис. 3.6.

Логічна функція, що її реалізує цей мультиплексор, така:

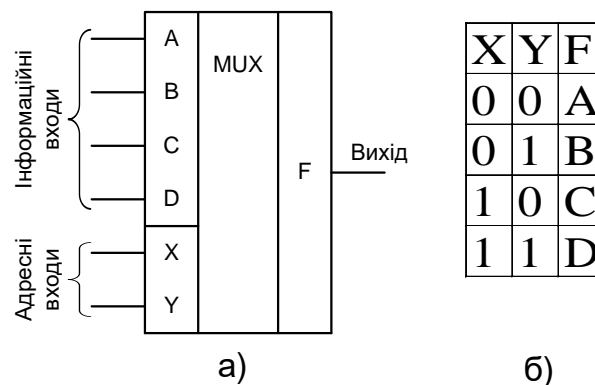


Рисунок 3.6 – Чотиривходовий мультиплексор (а) і його таблиця істинності (б)

$$F = \overline{A}x\overline{y} + B\overline{x}y + Cx\overline{y} + Dxy.$$

На основі мультиплексорів може бути реалізована будь-яка логічна функція, тому їх часто застосовують у складі програмованих ВІС.

Зауважимо, що К-МОН мультиплексори забезпечують комутацію не тільки цифрових, а також і аналогових сигналів.

3.5 Лічильники імпульсів

Однією з найбільш поширених операцій у пристроях інформаційно-обчислювальної і цифрової вимірювальної техніки є фіксування кількості

імпульсів – підрахунок їхньої кількості. Реалізують таку операцію лічильники імпульсів. Лічильники також забезпечують представлення інформації про кількість імпульсів у вигляді двійкового коду (завдяки принципові побудови).

Лічильники бувають прості (підсумовуючі, у яких код збільшується на одиницю після надходження на вхід кожного імпульсу; віднімаючі, у яких код відповідно зменшується після надходження на вхід кожного імпульсу) і реверсивні (суміщають властивості підсумовуючих і віднімаючих – можуть працювати в тому або іншому режимі за зовнішньою командою).

Як правило, лічильники будують на основі тригерів.

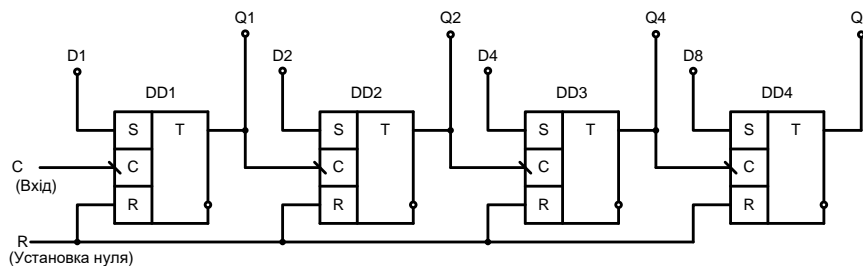


Рисунок 3.7 – Чотирирозрядний послідовний двійковий лічильник

Схема чотирирозрядного підсумовуючого послідовного двійкового лічильника, виконаного на комбінованих *RST* – тригерах з імпульсними інверсними входами, наведена на рис. 3.7, його умовне позначення – на рис. 3.8, часові діаграми роботи – на рис. 3.9, таблиця переходів – у табл. 3.1.

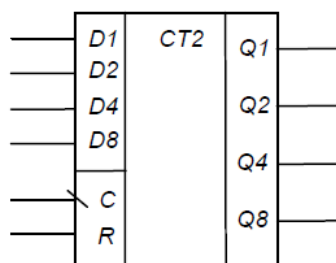


Рисунок 3.8 – Умовне позначення чотирирозрядного послідовного двійкового лічильника

Таблиця 3.1 – Таблиця переходів чотирирозрядного послідовного двійкового лічильника

Стан	Q_8	Q_4	Q_2	Q_1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1

Стан	Q_8	Q_4	Q_2	Q_1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Лічильник називається послідовним, тому що вихід тригера кожного попереднього розряду з'єднано з лічильним входом тригера наступного, в результаті чого передача інформації – перемикання тригерів розрядів лічильника – відбувається послідовно одного за одним. Це визначає низьку швидкість лічильника.

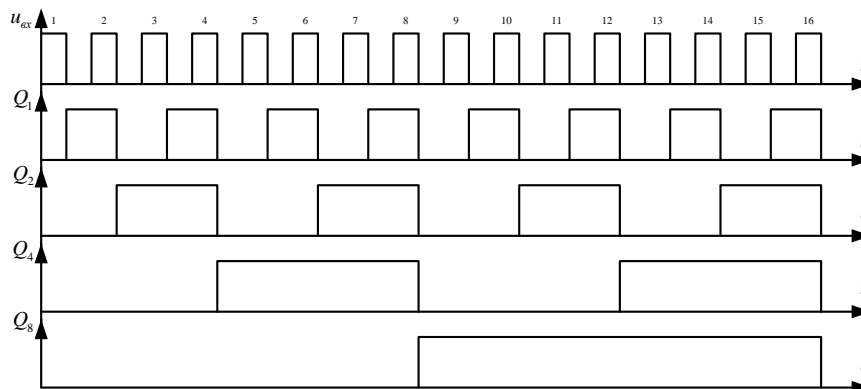


Рисунок 3.9 – Часові діаграми роботи чотирирозрядного підсумовуючого послідовного двійкового лічильника з імпульсним інверсним лічильним входом

У паралельних лічильників інформація з розряду в розряд передається за допомогою спеціальної комбінаційної схеми, а входи синхронізації тригерів з'єднано разом і перемикання всіх тригерів відбувається одночасно.

Як видно з таблиці переходів і часових діаграм, при безперервній роботі лічильника на його виходах Q_1 , Q_2 , Q_4 , Q_8 формується послідовний двійковий код.

Вхід R призначений для установки лічильника в нульовий стан (у даному разі - подачею сигналу логічної 1).

Входи D_1 , D_2 , D_4 , D_8 призначені для паралельного занесення чисел у лічильник – для завдання початкового стану, що відрізняється від нульового.

Послідовну роботу лічильника можна порушити, обмеживши кількість його станів. Це можна зробити, вводячи зворотний зв'язок, як, наприклад, показано на рис. 3.10. Тепер при досягненні десятого стану лічильник одразу переходить в нульовий стан – ми отримали двійково-десятковий лічильник,

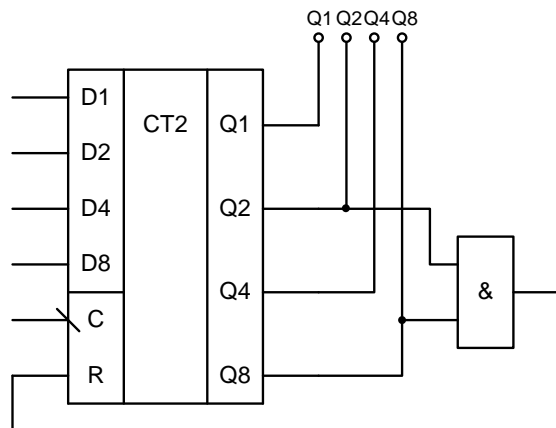


Рисунок 3.10 – Отримання двійково-десятькового лічильника з двійкового

Таблиця 3.2 – Таблиця переходів чотирирозрядного послідовного двійково-десятькового лічильника

Стан	$Q8$	$Q4$	$Q2$	$Q1$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0

Застосування таких лічильників разом із двійково-десятковими або двійково-семисегментними дешифраторами дозволяє на основі стандартних ІМС будувати схеми керування багаторозрядними десятковими індикаторами (розряди десяткові, а у межах десяткового розряду рахунок двійковий).

Питання для самоконтролю

1. Назвіть класи цифрових пристроїв.
2. Поясніть, що являють собою цифрові мікроелектронні пристрої?
3. Назвіть найбільш застосовувані цифрові мікроелектронні пристрої.
4. Що таке функціонально повна система логічних елементів? Наведіть приклади.
5. Поясніть, як, виходячи з правил де Моргана, можна побудувати елемент АБО-НІ на основі елемента І-НІ ?
6. Поясніть, як багатовходовий логічний елемент з інверсією на виході можна використати як інвертор?
7. Поясніть, як можна отримати логічний елемент з кількістю входів, що перевищує кількість входів елемента, обраного за базовий при побудові логічної схеми?
8. Як реалізують складні логічні функції?
9. Що таке дешифратори? Для чого вони призначені?
10. Поясніть різницю між двійковим і двійково-десятковим дешифраторами.
11. Що таке мультиплексор? Наведіть його таблицю істинності і логічну функцію, яку він виконує.
12. Поясніть, для чого призначені, які бувають і на чому можуть бути побудовані лічильники імпульсів?
13. Наведіть схему, умовне позначення і поясніть роботу чотирирозрядного підсумовуючого послідовного двійкового лічильника імпульсів, виконаного на комбінованих RST-тригерах.
14. Поясніть, як з підсумовуючого послідовного двійкового лічильника можна отримати двійково-десятковий? Для чого призначені двійково-десяткові лічильники?

Лекція 4

Цифроаналогові (ЦАП) та аналого-цифрові перетворювачі (АЦП)

1. Будова і робота ЦАП.
2. АЦП.

Основні поняття

Цифроаналогові перетворювачі (ЦАП); аналого-цифрові перетворювачі (АЦП); опорна напруга; матриця R-2R; метод перетворення; підсилювач з програмованим коефіцієнтом підсилення; перетворювач струму у напругу; ЦАП на основі методу перерозподілу заряду; АЦП послідовного наближення; функціонально закінчений АЦП; АЦП паралельного кодування.

Для узгодження цифрових пристроїв вимірювання і керування, що працюють з інформацією, представленою у двійковому коді, з виконавчими пристроями і датчиками, які мають аналогові відповідно вхідні та вихідні сигнали, застосовують цифроаналогові і аналого-цифрові (ЦАП і АЦП) перетворювачі.

4.1 Будова і робота ЦАП

ІМС ЦАП, як правило, являють собою резисторні матриці. Матриця може бути виконана як дільник напруги із співвідношенням опорів, кратним 2^n : R-2R-4R-8R-16R і т.д. (застосовують рідко, оскільки технологічно важко забезпечити точні значення великих опорів резисторів, що відповідають старшим розрядам). Частіше матриця ЦАП являє собою дільник струмів із співвідношенням опорів R-2R. Також до ІМС, як правило, входять транзисторні ключі (наприклад, на польових К-МОН-транзисторах), що забезпечують вмикання потрібної комбінації резисторів за сигналами двійкового коду.

Так, ІМС K572ПА1 (виконана за К-МОН-технологією) призначена для перетворення 10-розрядного прямого паралельного двійкового коду на цифрових входах у струм на аналоговому виході. Її спрощену структуру наведено на рис. 4.1. Видно, що, окрім власне резисторної матриці R-2R, вона має ключі і вузли керування ними ВК. Вхідним аналоговим сигналом ЦАП є опорна напруга $U_{оп}$, що визначає значення напруги, яка відповідає молодшому двійковому розряду. Вхідний струм I_0 , що проходить під дією $U_{оп}$, ділиться за двійковим законом у вузлах резисторної матриці R-2R. З рис. 4.2 можна бачити: приведені до місць перетинів А-А, В-В та С-С опори частини резисторів, що відтинаються у напрямку стрілок, дорівнюють $2R$ і тому значення струмів, які проходять через резистори $2R$, зменшуються удвічі зліва направо.

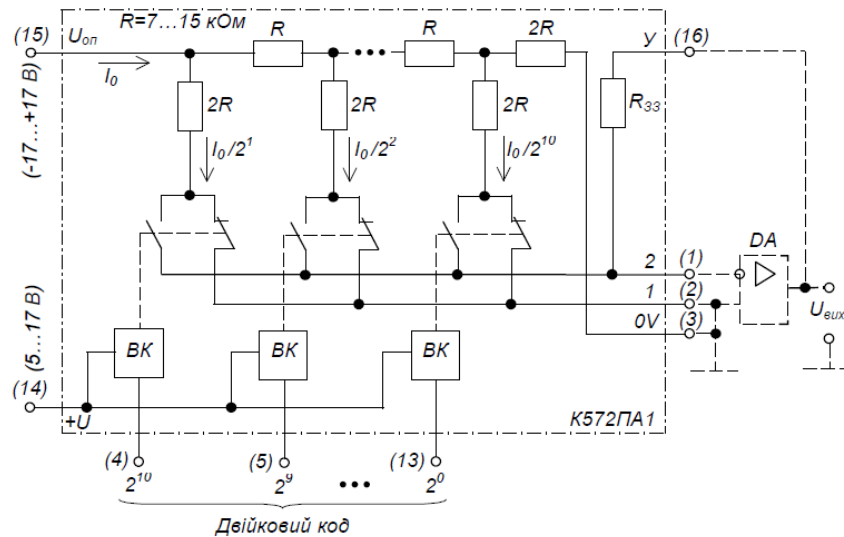


Рисунок 4.1 – Структура ІМС К572ПА1

Метод перетворення полягає в підсумовуванні у відповідності до заданих значень двійкового коду усіх розрядних струмів, зважених за двійковим законом і пропорційних значенню опорної напруги.

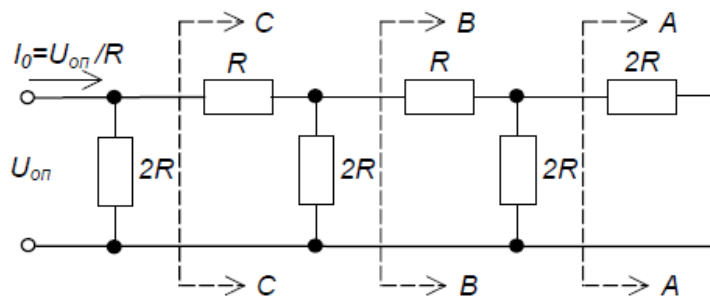


Рисунок 4.2 – Резисторна матриця R-2R

Струми віток резисторної матриці через ключі на К-МОН-транзисторах надходять на два аналогові виходи: на той чи інший – залежно від значень двійкових розрядів коду на відповідних цифрових входах. Значення струмів становлять:

$$\text{для першого виходу} - I_1 = \frac{U_{\text{оп}}}{R_{\text{екв}}} \sum_{i=1}^{10} 2^{-i} a_i; \quad (4.1)$$

$$\text{для другого виходу} - I_2 = \frac{U_{\text{оп}}}{R_{\text{екв}}} \sum_{i=1}^{10} 2^{-i} \bar{a}_i, \quad (4.2)$$

де $U_{\text{оп}}$ – опорна напруга;

$R_{\text{екв}}$ – еквівалентний опір резисторної матриці, що має номінальне значення 10 кОм (фактично – від 7 до 15 кОм);

a_i – пряме значення (для $a_i=1$) двійкового коду на i -му вході;

\bar{a}_i – інверсне значення (для $a_i=0$) двійкового коду на i -му вході.

Двійковий закон розподілення струмів у вітках матриці виконується за умови однаковості потенціалів обох її виходів. Це забезпечується підмиканням першого виходу до інвертуючого входу допоміжного ОП,

охопленого ВЗЗ, а другого – до неінвертуючого входу ОП і нульової точки схеми (як відомо, потенціали інвертуючого і неінвертуючого входів ОП, охопленого ВЗЗ, за умови роботи у лінійному режимі, майже однакові).

Отже, матриця R-2R підключається до ОП, утворюючи з ним інвертуючий підсилювач з програмованим коефіцієнтом підсилення. Його можна розглядати також як перетворювач струму, що надходить з виходу резисторної матриці, у напругу на виході ОП. У результаті кожному значенню двійкового коду на цифрових входах ЦАП відповідає деяке значення напруги на виході ОП, оскільки, забезпечуючи різницю потенціалів між своїми входами такою, що близька до нуля, ОП відводить через резистор зворотного зв'язку R_{ЗЗ} сумарний струм, що надходить до виходу (1) ІМС. Цей струм складається зі струмів резисторів 2R, підключених ключами до виходу (1) відповідно до значення двійкового коду і визначається розрядами $a_i = 1$.

Слід зазначити, що при практичній реалізації подібних пристроїв для зменшення взаємних впливів, що можуть призводити до збоїв у роботі, так звані “цифрову землю” і “аналогову землю” (обидві відповідають нульовому потенціалу схеми) слід виконувати у вигляді окремих провідників.

Аналітичний вираз, що зв'язує значення напруги на виході ОП зі значенням двійкового коду на цифрових входах ЦАП, такий:

$$U_{вих} = U_{ОП} R_{ЗЗ} (2^{b-1} a_1 + 2^{b-2} a_2 + \dots + 2^{b-i} a_i + 2^0 a_b) / 2^b R, \quad (4.3)$$

де R_{ЗЗ} – опір резистора зворотного зв'язку;

b – кількість розрядів перетворювача (у даному разі – 10), причому старшим є розряд з індексом b .

Значення вихідної напруги в кінцевій точці діапазону становить

$$U_{вих_макс}^* = U_{ОП} R_{ЗЗ} (1 - 2^{-b}) / R, \quad (4.4)$$

а найменше при усіх $a_i = 0$ – дорівнює нулеві.

Розрахунковий приріст вихідної напруги при зміні вхідного коду на одиницю молодшого розряду (крок квантування) становить

$$h = U_{ОП} R_{ЗЗ} / 2^b R. \quad (4.5)$$

Реально найбільше значення вихідної напруги при значеннях усіх розрядів двійкового коду $a_i = 1$ становить

$$U_{вих_макс} = \frac{U_{ОП}}{R_{ЗЗ}} - h, \quad (4.6)$$

оскільки десятирозрядний ЦАП працює з числами, що у десятковій системі числення відповідають від 0 до 1023.

Для забезпечення стабільності роботи задають $R_{ЗЗ} = R$, а сам резистор R_{ЗЗ} розміщують на кристалі ІМС разом із матрицею R-2R.

Електричну принципову схему цифроаналогового перетворювача, виконаного на основі ІМС К572ПА1, наведено на рис. 4.3. Якщо задати $U_{оп} = 10,24$ В, то отримаємо $U_{вих_макс}^* = 10,24$ В, а $h = 10$ мВ при номінальному значенні вхідного струму матриці 1 мА (фактично – від 0,5 до 2 мА).

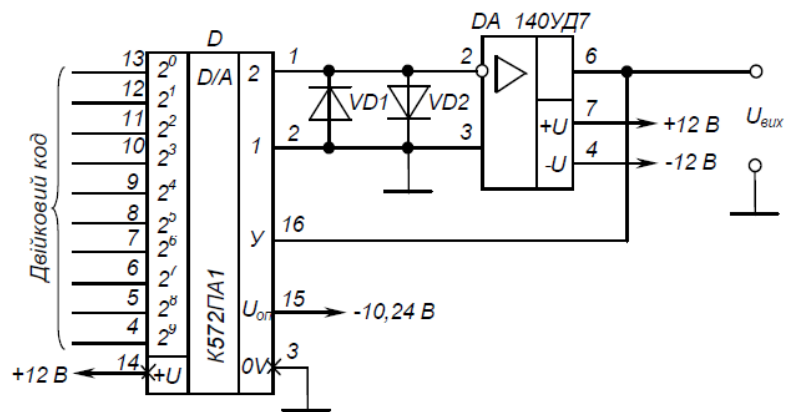


Рисунок 4.3 – Цифроаналоговий перетворювач на основі ІМС К572ПА1

Точнісні показники перетворювача залежать від точності виконання співвідношень $R_{33}/R = 1$ та $R/2R = 0,5$ для усіх ланок матриці.

Якщо формулу (4.3) переписати, вважаючи, що $R_{33}/2^b R = k$,

$$2^{b-1}a_1 + 2^{b-2}a_2 + \dots + 2^{b-i}a_i + 2^0a_b = X,$$

$$\text{отримаємо } U_{\text{вих}} = kU_{\text{оп}}X, \quad (4.7)$$

звідки видно, що даний ЦАП може бути використано також як помножувач аналогової величини $U_{\text{оп}}$ (що може набувати значення від мінус 17 до +17 В) на двійкове число X .

Більш точними є ЦАП на основі методу перерозподілу заряду. Вони будуються на основі конденсаторної матриці із співвідношенням ємностей, кратним 2^n .

Такий ЦАП має три етапи роботи.

На першому етапі ОП, що також входить до його складу, працює як повторювач, а усі конденсатори матриці підімкнено до вхідного контакту пристрою і накопичують заряд, пропорційний вхідній напрузі.

На другому етапі схема керування перемикає конденсатори від входу на нульовий провідник. Тепер на вхід ОП, що працює вже як нуль-орган, подається напруга, що дорівнює вхідній $U_{\text{вх}}$, але з протилежним знаком. На третьому етапі розпочинається перерозподіл заряду, коли до джерела опорної напруги $U_{\text{оп}}$ підмикається спочатку конденсатор старшого розряду матриці. На вході нуль-органу отримуємо напругу, що дорівнює $(0,5U_{\text{оп}} - U_{\text{вх}})$. Схема керування залежно від того, змінився стан на виході нуль-органу чи ні, залишає конденсатор у попередньому стані або знову перемикає його на нульовий провідник. Надалі те саме робиться з іншими конденсаторами, поки напруга на вході нуль-органу не знизиться до нуля і зміниться стан на його виході. Тоді відношення сумарної ємності, підімкненої до нульового провідника, до ємності, що була підімкнена до $U_{\text{оп}}$, буде еквівалентна відношенню $U_{\text{вх}}/U_{\text{оп}}$.

Підвищена точність ЦАП забезпечується тим, що пристрій час від часу переводиться у режим самоналаштування. У цьому режимі до кожного з конденсаторів матриці підмикаються допоміжні конденсатори малої ємності до того часу, поки сумарна ємність не стане більшою за суму ємностей конденсаторів молодших розрядів на ємність конденсатора наймолодшого

розряду. При цьому точність налаштування становить 0,25 ємності цього розряду.

4.2 АЦП

Світ, що нас оточує, є аналоговим. До аналогових відносять процеси, що змінюються за законом безперервної функції. Так, звуки та зображення надходять до наших органів відчуття у вигляді коливань – звукових або електромагнітних. Ці коливання сприймаються органами відчуттів (слух, зір) і у вигляді імпульсів передаються до мозку. Але інформація, що передається аналоговим способом, легко викривляється у тракті передачі та потребує величезних обсягів пам'яті при використанні в техніці. Спосіб «оцифровування» інформації спрощує процеси передачі і обробки інформації. Оцифровування – це процес перетворення аналогової інформації в цифрову. До цифрових відносять процеси, що змінюються за законом дискретної функції, як правило, це двійковий цифровий код. Техніка, яка працює з такою інформацією, називається цифровою. Цифрова інформація легко контролюється, дає стабільну і регульовану якість обробки і представлення процесів. Вона потребує менших ємностей для зберігання. Отже, для світу техніки цифрова інформація підходить набагато більше, ніж аналогова.

Процес оцифровування аналогової інформації проходить два основні етапи. На першому – аналогова інформація розбивається на невеликі рівні частини.

На другому етапі кожна частина аналізується і зашифровується спеціальними алгоритмами у коди з послідовності одиниць і нулів.

АЦП може бути побудовано на основі ЦАП, лічильника імпульсів і компаратора. Спрощена структурна схема такого АЦП наведена на рис. 4.4.

Цикл перетворення аналогового сигналу, представленого як напруга $U_{вх}$, у двійковий код складається з таких операцій: вхід пристрою – один з входів компаратора К. Сигнал з виходу компаратора дозволяє роботу генератора імпульсів ГІ. Сигнал Пуск встановлює нульовий стан і дозволяє роботу лічильника імпульсів ЛІ, що починає заповнюватись імпульсами ГІ.

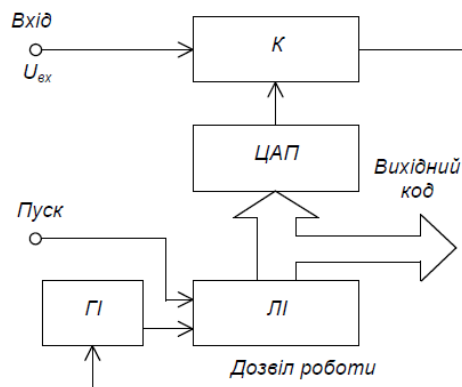


Рисунок 4.4 – Структурна схема АЦП

Код з виходу ЛП подається на цифрові входи ЦАП (входи керування ключами). У результаті з виходу ЦАП ступінчасто зростаюча напруга надходить на другий вхід компаратора. Після досягнення цієї напругою значення $U_{вх}$ компаратор забороняє роботу генератора, а на виході ЛП маємо прямий паралельний двійковий код, що відповідає значенню $U_{вх}$.

Реально для збільшення швидкості перетворення застосовують більш складний алгоритм. До входів ЦАП замість лічильника підмикається регістр пам'яті. Після пуску схема керування встановлює регістр у стан, якому відповідає одиниця у старшому розряді і нулі в усіх інших. При цьому на виході ЦАП формується напруга, що дорівнює половині діапазону перетворення. Якщо вона менша за $U_{вх}$, то в старший розряд регістра записується нуль, а у другий за старшинством – одиниця. Це відповідає напрузі на виході ЦАП, що дорівнює половині попередньої. У разі перевищення $U_{вх}$ цієї напруги одиниця встановлюється у третьому за старшинством розряді, і вихідна напруга ЦАП збільшується у 1,5 рази. Описана процедура повторюється до того часу, поки на виході ЦАП не сформується напруга, що відрізняється від $U_{вх}$ не більш ніж на ту, що відповідає одиниці молодшого розряду ЦАП.

Перетворювачі, що працюють за таким алгоритмом, називають АЦП послідовного наближення.

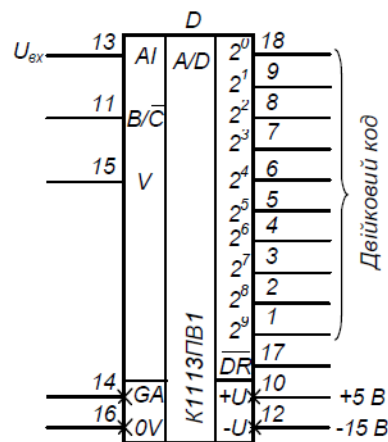


Рисунок 4.5 – ІМС АЦП К1113ПВ1

На рис. 4.5 наведено умовне позначення ІМС функціонально закінченого (такого, що не вимагає використання допоміжних елементів) АЦП послідовного наближення.

Він призначений для роботи з мікропроцесорними пристроями і є сумісним з мікропроцесорами, що працюють з ТТЛ-рівнями сигналів.

Його вихідні вузли мають три стани: два логічних (0 та 1) і третій – стан високого імпедансу – стан відімкнення, що забезпечує просте спраження з шиною даних мікропроцесора.

Режим роботи АЦП у мікропроцесорній системі визначається сигналами керування від мікропроцесора.

При надходженні на вхід гасіння і перетворення $U_{оп} B/\bar{C}$ логічного нуля

АЦП починає цикл перетворення вхідної напруги $U_{\text{вх}}$ у двійковий код.

Після завершення перетворення на виході готовності даних \overline{DR} з'являється сигнал логічної 1, що є запитом для мікропроцесора на прийом коду. У вихідному стані і стані перетворення на цьому виході утримується сигнал 1. Після прийому коду мікропроцесор подає на вхід B/C сигнал логічної 1, що встановлює АЦП у вихідний стан, після чого він готовий до нового циклу перетворення.

Цей АЦП може обробляти вхідну інформацію у вигляді однополярної напруги до 10,24 В або двополярної 5,12 В. У однополярному режимі на вхід керування зсувом нуля V треба подати сигнал логічного 0, а у двополярному – залишити його вільним. Точності перетворення $1/2$ одиниці молодшого розряду цифрового коду досягають вмиканням між виводом “аналогової землі” GA і корпусом OV резистора опором від 5 до 50 Ом (використовують резистор змінного опору).

На кінець слід зазначити, що елементарним пристроєм перетворення аналогової величини у дискретну є компаратор, який фіксує факт перевищення однієї напруги іншою і може мати на виході сигнали, що відповідають логічним 0 або 1. При цьому найбільш швидкодіючим є побудований на основі лінійки компараторів АЦП паралельного кодування. Напруга $U_{\text{вх}}$ тут подається на перші входи усіх компараторів одразу. На другі їхні входи подається напруга з дільника, що складається з резисторів одного номіналу. Таким чином, значення напруги, з яким відбувається порівняння $U_{\text{вх}}$ двома сусідніми компараторами, відрізняється на напругу, що відповідає молодшому розряду. Схема шифрування формує вихідний код, який відповідає старшому компаратору, з тих, що спрацювали. Цей код записується у вихідний регістр пам'яті. Але апаратні затрати цього методу надзвичайно великі. Так, для восьмирозрядного АЦП потрібно 255 компараторів і близько $3 \cdot 10^4$ активних компонентів (транзисторів).

Питання для самоконтролю

1. Для чого призначені ЦАП і АЦП?
2. Поясніть, як побудовано ЦАП?
3. Чому ЦАП будують на основі матриці $R-2R$?
4. Поясніть метод перетворення цифрової величини у аналогову ЦАП на основі матриці $R-2R$.
5. Яку роль у складі ЦАП на основі матриці $R-2R$ відіграє операційний підсилювач?
6. Наведіть структурну схему і поясніть принцип роботи АЦП послідовного наближення, побудованого на основі ЦАП.
7. Для чого в АЦП послідовного наближення використовується ЦАП?
8. Поясніть поняття „функціонально закінчений АЦП”.
9. Поясніть будову і принцип дії АЦП паралельного кодування.

Лекція 5

Тема „Схемотехніка програмованих цифрових пристроїв. Мікропроцесорні пристрої”

1. Загальні положення. Історія процесорів.
2. Особливості роботи і використання мікропроцесорів.
Двійкова система числення. Структура мікропроцесорної системи.
3. Структура мікропроцесора.

Основні поняття

Алгоритм; жорстка логіка; програмована логіка; мікропроцесор (МП); структура процесора; арифметико-логічний пристрій (АЛП); пристрій керування (ПК); програма; пристрій вводу і виводу (ПВВ); запам'ятовуючий пристрій (ЗП); постійний запам'ятовуючий пристрій (ПЗП); оперативний запам'ятовуючий пристрій (ОЗП); десяткова система числення; двійкова система числення; мікропроцесорний пристрій (МПП); інтерфейс; засоби сполучення (ЗС); магістральна схема; шина адреси (ША), шина даних (ШД); шина керування (ШК); логічний стан; третій (нелогічний) стан; структура МП; генератор тактових імпульсів (ГТІ); регістри; акумулятор; регістр стану; лічильник команд; регістр адреси пам'яті; регістр команд; робочі регістри; буферні регістри інтерфейсу; вихідні підсилювачі; пристрій керування.

5.1 Загальні положення. Історія процесорів

З розвитком технології і схемотехніки ІМС виникла можливість створення на одному кристалі цілого електронного пристрою, наприклад, радіоприймача, вимірювального приладу чи пристрою керування. Але зрозуміло, що при цьому ІМС (ВІС) стають вузькоспеціалізованими. І якщо у випадку застосування пристроїв масового призначення, що продукуються десятками тисяч (радіоприймач, вимірювальний прилад) це можна прийняти, то з пристроями керування виникають великі проблеми.

Перш за все алгоритм (послідовність виконання дій, що веде до успіху) керування часто є індивідуальним (наприклад, для кожного з автоматичних маніпуляторів, що працюють на складальному конвеєрі з виробництва автомобілів). Крім того, у процесі експлуатації технологічного устаткування часто виникає необхідність у заміні всього алгоритму керування або його частин (додавання або вилучення операцій, зміна послідовності їх виконання). Застосування спеціалізованих ВІС у такому разі з економічної точки зору недоцільне.

Але й побудова пристроїв керування на основі універсальних ІМС середнього ступеня інтеграції, хоча це й менше коштує, теж не є вдалим виходом, оскільки зміни алгоритму при цьому ведуть до зміни електричної схеми пристрою керування (повністю або у деяких її частинах).

До речі, пристрої керування з незмінним алгоритмом називають пристроями з жорсткою логікою.

Усунути протиріччя між ступенем складності ВІС і можливістю її універсального використання вдалося за рахунок застосування програмування виконуваних функцій (у найпростішому випадку це, наприклад, програмування коефіцієнта перерахунку лічильника імпульсів).

Зараз основним типом програмованих ВІС, що мають найбільшу інформаційну потужність, є мікропроцесор.

Мікропроцесор (МП) – це програмно керований пристрій обробки цифрової інформації, виконаний у вигляді однієї (рідше декількох) ВІС. За функціями МП аналогічний центральному процесору цифрової ЕОМ. Його можна вважати одним з найбільших досягнень сучасної мікроелектроніки.

Тут слід зазначити, що спроби автоматизувати обчислювальні роботи робилися дуже здавна, і займалися цим найвидатніші мислителі і дослідники. Але рівень техніки і технології не дозволяв створити відповідні пристрої. І лише із середніх віків починаються зрушення у цій справі.

Так, у 1642 році фізик Б. Паскаль побудував механічну машину, що виконувала операції додавання та віднімання (надалі було застосовано до п'ятдесяти таких машин). Подібні пристрої розробляли німецький математик Г.В. Лейбніц і російський математик П. Л. Чебишев (орієнтовані на розв'язання рівнянь вищої математики), а пізніше російський інженер В. Т. Орднер. “Колесо Орднера” стало основою арифмометрів. Далі були електромеханічні пристрої – на електромагнітних реле – та, нарешті, електронні – спочатку на лампах, а потім на транзисторах та ІМС.

Найбільш близьким прообразом сучасних цифрових ЕОМ є механічна “аналітична машина” англійського математика Ч. Беббіджа, запропонована ним у 1833 році і пізніше реалізована його сином (додавання двох чисел тривало 2 секунди, а множення – хвилини). У цій машині вперше була запропонована і реалізована ідея процесора, що по чергово виконує у заданій послідовності наперед задані математичні операції над числами. Структура процесора Ч. Беббіджа, що наведена на рис. 5.1, не зазнала суттєвих змін і до нашого часу.

Арифметико-логічний пристрій АЛП тут виконує арифметичні або логічні дії над числами (кодами) у порядку, що задається пристроєм керування ПК згідно із програмою, яка вводиться до нього (наприклад, з перфоративної стрічки).

Практичне застосування процесора можливе за наявності пристрою вводу і виводу чисел ПВВ і запам'ятовувального пристрою ЗП, необхідних для вводу вихідних даних, виводу результатів та збереження проміжних результатів.

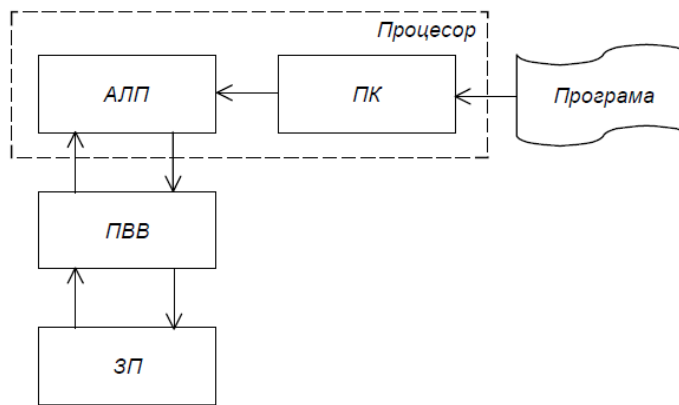


Рисунок 5.1 – Структура процесора

Повна механічна обчислювальна машина, що мала процесор, ПВВ і ЗП, побудована 1936 року у Німеччині під керівництвом К. Цузе. Тоді ж з'являється й перша алгоритмічна мова для написання програм для цієї машини. З цього часу розпочинається стрімкий розвиток процесорів як основної ланки обчислювальної машини.

Так, 1938 року також під керівництвом К. Цузе створено процесор на електромагнітних реле, що збільшило швидкість виконання операцій у десятки разів.

1946 року у США побудовано першу електронну обчислювальну машину ENIAC, що містила 18000 електронних ламп (додавання і віднімання тривало 200 мікросекунд, а множення 2800 мікросекунд).

Одна з перших радянських ЕОМ, призначена для серійного виробництва, "Стрела" була створена у 1953 році. Вона містила 6000 ламп, споживала потужність 150 кВА і розміщувалася у приміщенні площею 300 квадратних метрів: уніфіковані її комірки (логічні елементи, тригери та ін.) містилися у 6 вертикальних стійках висотою 2,5 метри та у пульті оператора. Вона виконувала 2000 операцій за секунду. Її постійно обслуговували 5-7 техніків.

Уже у 1955 році з'являються транзисторні процесори, а з початку 60-х років – виконані на ІМС середнього ступеня інтеграції (процесор виконувався десь на п'ятдесяти ІМС).

Для появи мікропроцесора (процесора у мікровиконанні – у вигляді ІМС) необхідна була не тільки технологічна база, що вже з'явилася на кінець 60-х років, а й усвідомлення, розуміння того, що програмно керована логіка більш ефективна і простіша у виконанні при реалізації складних функцій, ніж схеми з жорсткою логікою.

Так, у 1969 році співробітнику фірми Intel (США) М. Хоффу надійшла пропозиція від японської фірми BASICOM розробити комплект ІМС для виконання усіх математичних дій і розрахунку усіх стандартних функцій універсальним калькулятором. Передбачалося створення дванадцяти ІМС на 2000 транзисторів кожна, що мали б виконувати відповідні групи із задуманих функцій. Однак М. Хофф запропонував використати лише одну універсальну ІМС із програмним керуванням що виконувала б усі задані функції, а за

необхідності й інші.

Єдиним недоліком було збільшення часу обчислень, що незворотне при програмній реалізації (порівняно з жорсткою логікою). Але для калькулятора це було несуттєво: збільшення тривалості обчислень з десятків мікросекунд до десятків мілісекунд користувачем не помічалось. Отже, 1971 року фірма INTEL створила мікропроцесор INTEL 4004, що виконував операції з чотирирозрядними двійковими числами і вміщував 2250 транзисторів.

З'явившись у мікрокалькуляторах, мікропроцесори (програмна логіка) стали застосовуватись у найрізноманітніших обчислювальних і керуючих пристроях, витісняючи схеми з жорсткою логікою.

5.2 Особливості роботи і використання мікропроцесорів. Двійкова система числення. Структура мікропроцесорної системи

Особливістю сучасних процесорів є те, що вони працюють з числами, представленими не у десятковій, а у двійковій системі числення: число представляється не як сума ступенів числа 10, помножених на відповідні числа від 0 до 9:

$$N_{10} = a_1 \cdot 10^0 + a_2 \cdot 10^1 + a_3 \cdot 10^2 + \dots + a_{n+1} \cdot 10^n, \quad (5.1)$$

де $a_i = \{0;1;2;3;\dots;9\}$,

а як сума ступенів числа 2, помножених на 0 або 1:

$$N_2 = a_1 \cdot 2^0 + a_2 \cdot 2^1 + a_3 \cdot 2^2 + \dots + a_{n+1} \cdot 2^n, \quad (5.2)$$

де $a_i = \{0;1\}$.

У результаті основою такого процесора є двостабільний (такий, що має два стани), а не десятистабільний елемент.

Прикладом двостабільного елемента, призначеного для запам'ятовування інформації, є тригер, а десятистабільного – може бути двійково-десятковий лічильник.

Двійково-десятковий лічильник, що має 10 станів, побудовано на чотирьох тригерах. У той самий час на чотирьох тригерах може бути побудовано пристрій (наприклад, також лічильник або регістр), що має 16 станів. Звідси видно, чому за основу взято двійкову систему числення. Її застосування при побудові цифрових електронних засобів з обробки інформації дозволяє уникнути зниження їхніх потенційних можливостей більш ніж на третину. Отже, двійкова система при технічній реалізації є більш економною з точки зору апаратних затрат.

У 1946 році ВМС США замовили двом університетам – Гарвардському і Принстонському – розробку цифрової обчислювальної машини для систем керування артилерією лінкорів. Гарвардська структура мала два пристрої пам'яті: програм і даних. Це забезпечувало високу швидкодію, але вимагало використання двох трактів передачі інформації. Оскільки такі машини тоді будували на електронно-вакуумних лампах, то введення до їхнього складу будь-якого вузла вело до значного збільшення габаритів і зменшення надійності всієї системи. Отже, перемогла принстонська структура – структура фон Неймана, при якій дані і програма зберігаються в одному й тому

самому ЗП і для обміну даними між АЛП й ЗП та між ПК й ЗП використовуються однакові процедури. Після звернення МП до ЗП перший код вважається кодом команди, а наступні – адресами, даними чи командами у контексті виконання першої команди. Структуру фон Неймана застосовують при побудові і сучасних універсальних процесорів. Але зараз, особливо в мікроконтролерах, про які піде мова дещо пізніше, застосовують і гарвардську структуру, що значно підвищує їхню швидкодію. А надійність систем визначається дуже високою надійністю ВІС – надійність мікросхеми вища за надійність навіть однієї лампи.

На основі МП виконуються мікропроцесорні пристрої (МПП), перш за все мікроЕОМ – пристрій, що містить МП, запам'ятовувальні пристрої, органи керування і засоби зв'язку із зовнішніми пристроями - інтерфейс.

Інтерфейс (англ. *interface* – засіб спряження, сполучення) є сукупністю уніфікованих технічних і програмних засобів, необхідних для підключення зовнішніх пристроїв. Він забезпечує перетворення сигналів МП у сигнали, що сприймаються зовнішніми пристроями і навпаки, підсилення сигналів являє собою апаратні засоби і набір програм передачі даних (уніфікований протокол обміну інформацією).

Якщо мікроЕОМ призначена для керування деяким об'єктом, то вона доповнюється засобами сполучення (узгодження) з об'єктом (датчики, аналогоцифрові і цифро-аналогові перетворювачі, виконавчі пристрої і т.п.). Сукупність мікроЕОМ і засобів сполучення називають мікропроцесорною системою. Структура такої системи наведена на рис. 5.2.

МП є мозком МПП, в якому відбувається обробка інформації (команд і даних), що представляються у двійковому коді.

МП може виконувати дуже складні завдання з обчислень і керування, уміючи виконувати лише елементарні логічні й арифметичні операції, операції пересилання даних, порівняння двох чисел і деякі інші, за рахунок багаторазового їх повторення відповідно до заданої користувачем програми (визначеної послідовності команд).

Для забезпечення виконання операцій МП містить пристрої вибірки інформації з пам'яті і її дешифрування, арифметико-логічний пристрій, що становить сукупність схем, що реалізують арифметичні і логічні операції над даними, пристрій керування, який забезпечує виконання операцій МП, різні регістри, що слугують для тимчасового зберігання (надоперативний ЗП) і перетворення даних і команд, тактовий генератор, що задає темп роботи МП.

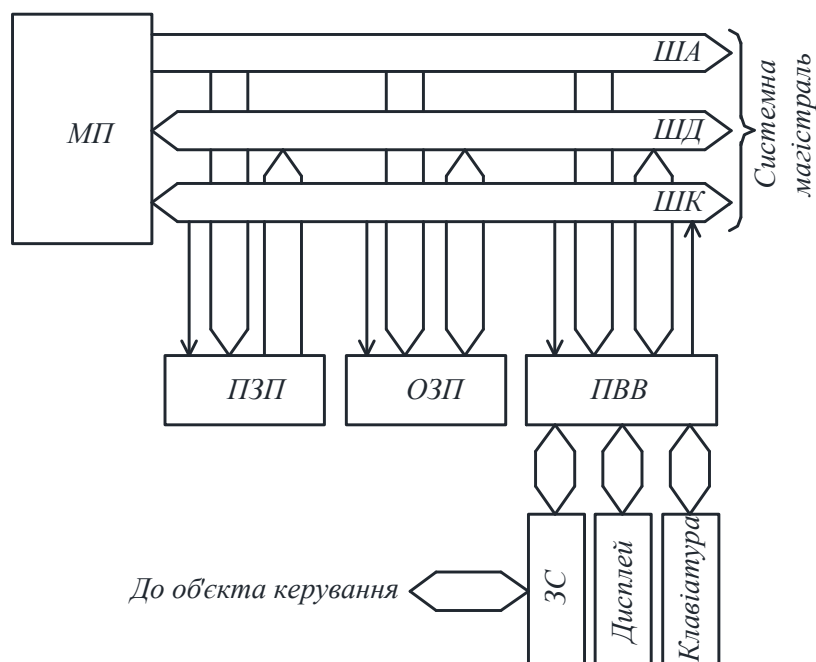


Рисунок 5.2 – Мікропроцесорна система

Для зберігання інформації є два види ЗП: постійний запам'ятовувальний пристрій ПЗП та оперативний запам'ятовувальний пристрій ОЗП.

ПЗП призначений для зберігання інформації, що заноситься в нього при виготовленні МПП і зберігається при відключенні живлення. Завдяки цьому МПП після вмикання знає, як підготувати себе до роботи і що потрібно робити при одержанні тих чи інших зовнішніх команд і дій. ПЗП може працювати тільки у режимі видачі інформації. Вміст ПЗП можна змінити заміною його ІМС на інші з новим набором програм або перепрограмуванням ІМС, якщо вони це допускають.

ОЗП призначений для тимчасового зберігання даних і програм користувача, проміжних результатів роботи і працює як у режимі запису, так і у режимі видачі інформації. Якщо інформацію, що міститься в ОЗП, необхідно зберегти довгостроково, то її треба записати у зовнішній пристрій пам'яті (наприклад, на магнітофон або жорсткий диск).

ПВВ забезпечує зв'язок МП із зовнішніми пристроями – дисплеєм, клавіатурою (засоби з'єднання з оператором) та засобами з'єднання з керованим об'єктом ЗС (датчики, виконавчі пристрої, АЦП, ЦАП і т.п.).

Зв'язок між розглянутими модулями МПП здійснюється на основі магістральної схеми. Відповідно до неї модулі підключені до системної магістралі, що складається із шини адреси ША, шини даних ШД і шини керування ШК. Кожна із шин являє собою набір з деякого числа провідників – ліній.

По ША МП повідомляє, з яким із модулів чи з якою коміркою пам'яті (зазначається їхній код – адреса) він буде працювати у даний момент часу.

Розрядність ША (число провідників) визначає число модулів, з яким може взаємодіяти МП. Як правило становить 16 (число модулів до 216 –

(АЛП), регістри даних (РД) і пристрій керування (ПК). Ці та інші пристрої, що беруть участь у пересиланні даних, з'єднано шинами внутрішньої магістралі: ША, ШД і ШК.

Синхронізує і задає темп роботи (швидкодію) МП генератор тактових імпульсів (ГТІ).

АЛП являє собою комбінаційний пристрій, у якому відбувається виконання арифметичних і логічних операцій над даними.

Регістри забезпечують виконання функцій МП за рахунок зберігання кодів даних, адрес, команд та ін.

Акумулятор є основним регістром МП при діях з даними. Більшість арифметичних і логічних операцій з двома словами даних здійснюється з використанням АЛП і акумулятора.

У регістрі стану (ознака прапорців) зберігаються ознаки результату побіжної операції АЛУ (нульовий, додатний, від'ємний та ін.). Ці ознаки використовуються при виконанні наступних команд, наприклад, для розгалуження програми.

Лічильник команд зберігає номер команди, що виконується і тієї, що буде виконуватися наступною. Перед початком виконання програми до нього заноситься адреса її першої команди.

Регістр адреси пам'яті містить адресу пам'яті, що буде використана МП у наступному такті роботи. Його виходом є ША.

Регістр команд зберігає код поточної виконуваної команди.

Робочі регістри є внутрішньою надоперативною пам'яттю МП і застосовуються для проміжного зберігання кодів даних і адрес, забезпечуючи підвищення швидкодії.

Підключення внутрішньої магістралі МП до системної магістралі МПП відбувається через буферні регістри інтерфейсу і вихідні підсилювачі.

Пристрій керування узгоджує роботу всіх вузлів МП.

Питання для самоконтролю

- 1. Поясніть, що таке мікропроцесор і для чого він призначений.*
- 2. Наведіть структуру процесора і поясніть призначення його вузлів.*
- 3. Наведіть структуру мікропроцесорної системи і поясніть призначення її складових частин.*
- 4. Зазначте різницю між постійним та оперативним запам'ятовувальними пристроями.*
- 5. Поясніть, що таке системна магістраль і як здійснюється підключення до неї модулів мікропроцесорної системи.*
- 6. Поясніть, що являє собою інтерфейс.*

Лекція 6

Тема „Запам'ятовуючі пристрої”

1. Загальні відомості.
2. Запам'ятовуючі пристрої для зберігання оперативної інформації.

Основні поняття

Інформаційна ємність; біт; слово (байт); організація ЗП; швидкодія ЗП; ЗП RAM типу; статичні ЗП – SRAM; динамічні ЗП – DRAM; енергозалежна пам'ять; енергонезалежна пам'ять; ЗП для зберігання постійної інформації ROM типу; масочні ЗП типу ROM(M); ЗП типу PROM; ЗП типу EPROM; репрограмовані ІМС пам'яті; транзистори ЛІЗМОН типу; ЗП типу EEPROM; флеш-пам'ять.

6.1 Загальні відомості

Запам'ятовуючі пристрої (ЗП) забезпечують зберігання програми роботи, вихідних даних і результатів обробки, обмін інформацією між окремими частинами мікропроцесорної системи. Будуються вони на напівпровідникових ІМС. Мікросхеми пам'яті становлять близько 40 % у загальному обсязі випуску ІМС. Також ЗП входять до складу ВІС: мікропроцесорів, мікроконтролерів та ін.

Максимально можливий об'єм інформації ЗП визначається його інформаційною ємністю в бітах (біт – мінімальний обсяг цифрової інформації, може набувати значення 0 або 1) або словах (байтах, що складаються з декількох бітів, наприклад, з чотирьох або восьми). Біт зберігається у запам'ятовувальному елементі (ЗЕ), а слово – у запам'ятовувальній комірці, що є групою ЗЕ, до яких можливе лише одночасне звернення.

Інформаційна ємність вимірюється, наприклад, в кілобітах ($1 \text{ Кбіт} = 2^{10} = 1024 \text{ біти}$), кілобайтах ($1 \text{ Кбайт} = 2^{10} = 1024 \text{ байти}$), або в мегабітах ($1 \text{ Мбіт} = 2^{20} = 1048576 \text{ бітів}$), мегабайтах ($1 \text{ Мбайт} = 2^{20} = 1048576 \text{ байтів}$) і т.п.

Організація ЗП визначає, скільки слів і якої розрядності він зберігає. Так, наприклад, при ємності 512 бітів може зберігатися 64 восьмирозрядних слова, або 128 чотирирозрядних.

Швидкодія ЗП оцінюється часом зчитування і запису.

Цифрові обчислювальні пристрої мають шинну структуру, при якій звертання до комірки ЗП відбувається за номером, що називається адресою.

6.2 ЗП для зберігання оперативної інформації (ОЗП)

Такі ЗП **RAM** типу (Random Acces Memory – ОЗП) поділяються на статичні і динамічні.

У статичних ЗП – **SRAM** (Static RAM) – за ЗЕ використовуються тригери

з ланцюгами установки і встановлення нуля. При К-МОН-технології тригер складається з шести транзисторів. Тому такі ЗП досить дорогі і займають багато місця на кристалі, але мають велику швидкодію.

У динамічних ЗП – **DRAM** (Dynamic RAM) – дані зберігаються у вигляді зарядів ємностей МОН-структур. Основою ЗЕ тут є просто конденсатор невеликої ємності. Такий ЗЕ значно простіший за тригерний, що дозволяє розмістити на кристалі значно більше ЗЕ. Оскільки конденсатор з часом втрачає свій заряд, то зберігання даних вимагає їхньої періодичної регенерації (через кожні декілька мікросекунд) за допомогою спеціальних підсилювачів-регенераторів. Зараз ємність динамічних ЗП становить до 128 Мбіт, у стані розроблення ЗП на 256 Мбіт (з числом транзисторів на кристалі до 300 мільйонів). Динамічні ЗП у 4-5 разів дешевші за статичні й у стільки ж разів мають більшу інформаційну ємність.

Обидва ці типи пам'яті є енергозалежними – при вимкненні джерела живлення ІМС інформація безповоротно втрачається.

6.3 Запам'ятовуючі пристрої для зберігання постійної інформації (ПЗП)

Є декілька типів ЗП для зберігання постійної інформації – **ROM** типу (Read On Memory – ПЗП). Як в них використовують перемички, діоди, біполярні і МОН-транзистори.

У масочні ЗП типу **ROM(M)** інформація записується при виготовленні ІМС за допомогою спеціального шаблону – маски. Матриця діодного ЗП такого типу являє собою координатну сітку з горизонтальних ліній вибірки слів і вертикальних ліній зчитування. Код слова визначається наявністю діода (що відповідає одиниці) або його відсутності (відповідає нулю) у вузлах координатної сітки.

Такі ЗП компактні і дешеві. Їх застосовують для зберігання інформації масового призначення – кодів літер і цифр, таблиць типових функцій, стандартного програмного забезпечення і т.п. Користувач змінити інформацію, що зберігається, не може.

Мікросхеми ЗП типу **PROM** (Programmable ROM – програмовані ПЗП) програмують одноразово видаленням або створенням перемичок у вузлах координатної сітки. У вихідній заготовці наявні (або відсутні) всі перемички. При програмуванні користувачем за допомогою спеціального програмуючого пристрою залишаються (або видаляються) тільки необхідні. Перемички можуть являти собою або плавкі елементи, ввімкнені послідовно з діодом, або два зустрічно увімкнених діоди, один з яких пробивається при програмуванні.

ЗП типу **EPROM** дозволяють не тільки записувати в них інформацію, а й стирати її та замінювати на нову – вони є репрограмовуваними. Як ЗЕ в них використано транзистори ЛІЗМОН типу (МОН- транзистори з лавинною інжекцією заряду). Такі транзистори мають так званий плаваючий затвор – обмежену з усіх боків діелектриком провідну зону. Введений у неї як у пастку в результаті лавинного пробоя під дією імпульсу напруги у 20-25 В заряд

зберігається дуже тривалий час. Цей заряд забезпечує закритий стан транзистора. Стирання інформації відбувається під дією ультрафіолетового опромінення, для чого корпус ІМС має спеціальне прозоре віконце. Виникнення фото- і теплових струмів дозволяє заряду покинути плаваючий затвор. Стирання триває декілька хвилин, одразу стирається вся інформація. Опромінення веде до змін властивостей матеріалів транзисторів, тому число циклів перепрограмування становить 10-100.

Новітні ЗП типу **EEPROM** також є репрограмованими. Їхньою основою є МНОН-транзистори, що на відміну від звичайних МОН-транзисторів мають двошаровий підзатворний діелектрик – окрім тонкого шару SiO_2 є ще більш товстий шар нітриду кремнію Si_3N_4 (звідси літера Н в аббревіатурі). Під дією електричного поля достатньо високої напруженості носії заряду проходять через тонкий шар і скупчуються на межі поділу шарів. Після зняття поля заряд залишається у примежевому шарі нітриду кремнію і вже не розсмоктується, що і забезпечує зберігання інформації впродовж десятків років. Для стирання інформації необхідно видалити носії заряду з примежевого шару, для чого подається напруга, що створює електричне поле протилежної направленості. При цьому інформацію можна стерати не зі всього кристалу, а вибірково. Тривалість процесу досить коротка. Кількість циклів перепрограмування становить 104-106 разів.

Найвищим досягненням у розробленні напівпровідникових програмованих ЗП є створення флеш-пам'яті. Її ЗЕ подібні до елементів пам'яті типу EEPROM, але в схемах флеш-пам'яті не передбачене стирання окремих слів. Інформація може стиратися або вся одразу, або досить великими блоками за єдиним сигналом, миттєво (flash – спалах). За своєю дією флеш-пам'ять вже подібна до пам'яті ОЗУ, але є енергонезалежною. Пристрої флеш-пам'яті зі збільшенням кількості запам'ятовуваної інформації (з розвитком технології) поступово витісняють такі накопичувачі, як жорсткі магнітні диски, оптичні диски та інші носії. Вони у сотні разів скорочують споживану потужність, мають велику надійність, невеликі розміри і вагу та у декілька разів збільшують швидкодію пристроїв пам'яті. Однією з найважливіших властивостей є також відсутність в ЗП цього типу рухомих механічних частин, в тому числі таких, що обертаються, і повна технологічна сумісність із ІМС. Флеш-пам'ять може бути частиною багатьох типів ІМС-мікроконтролерів (що забезпечує універсальність їхнього застосування за рахунок можливості багатократного перепрограмування) та ін. Також зазначимо, що це є одним із прикладів поступової відмови людства від колеса у багатьох застосуваннях.

Питання для самоконтролю

- 1. Поясніть, у чому вимірюється інформаційна ємність ЗП?*
- 2. Що визначає організація ЗП?*
- 3. За якими показниками оцінюють швидкодію ЗП?*
- 4. Які різновиди ЗП RAM типу ви знаєте?*
- 5. Які різновиди ЗП ROM типу ви знаєте?*
- 6. Що використовують як запам'ятовуючі елементи в ЗП різного типу?*
- 7. Як працює флеш-пам'ять?*
- 8. Поясніть, які типи пам'яті є енергозалежними і чому?*
- 9. Які типи пам'яті є енергонезалежними і чому?*

Лекція 7

Тема „Завдання і виконання програм мікропроцесорного пристрою”

Питання лекції

1. Алгоритм. Способи представлення алгоритму.
2. Програми МПП.
3. Шістнадцяткова система числення.
4. Асемблер.

Основні поняття

Алгоритм; представлення алгоритму; ініціалізація виконання програми; машинна програма; шістнадцяткова система числення; асемблер; програма-ретранслятор; мнемоніка; мікрокоманда; мікрооперація; внутрішній ПЗП; система команд; вихідна програма; об'єктна програма; потужність МП.

7.1 Алгоритм. Способи представлення алгоритму

Алгоритм (від латинської трансляції імені математика аль-Хорезмі) – це спосіб (програма) розв'язання обчислювальних та інших задач, що точно вказує, як і в якій послідовності отримати результат, однозначно визначений вихідними даними.

Отже, послідовність виконання дій пристрою керування і є алгоритмом. Задати алгоритм можна різними способами.

Перш за все - це описове представлення алгоритму на “людській” мові. Така форма представлення прийнятна навіть для неспеціаліста у пристроях керування, але громіздка, –допускає тлумачення. А отже, непридатна для створення більш-менш складного пристрою керування, як з жорсткою логікою, так і програмованого.

Більш впорядкованим є представлення алгоритму у вигляді таблиці.

Досить популярним є графічне представлення алгоритму – у вигляді граф-схеми. Воно забезпечує наочність, оперативність огляду.

На основі граф-схеми складають схему електричну принципову пристрою керування з жорсткою логікою або програму роботи програмованого.

Це, врешті, теж є способами представлення алгоритму.

7.2 Програми МПП

Для забезпечення виконання МПП необхідних функцій у його ПЗП (іноді у ОЗП) записується набір програм, (стандартних, що реалізують, наприклад, арифметичні і логічні операції, обчислення тригонометричних функцій і т.п., а також і спеціальних, що реалізують, наприклад, алгоритми керування конкретним об'єктом).

При складанні програм програміст повинен дати МП детальний опис його дій. МП може працювати з великою швидкістю і точністю впродовж тривалого часу, але лише за чітким завданням, що не допускає тлумачень. МПП може реагувати на зміну умов, якщо в пам'яті у нього є програма, що повідомляє МП, як йому поводитись у нових обставинах. МП діє логічно, але не має творчих здібностей (принаймні у даний час). Удавана розумність різних МПП є результатом великого числа програм, закладених у них. Хоча слід зазначити, що в наш час інтенсивно ведуться роботи із створення так званого штучного інтелекту (набору програм, що імітують інтелектуальну діяльність людини). І на цьому шляху, як відомо, здобуто чимало досягнень. Вважається, що створення “розумних” машин буде у першому наближенні завершено тоді, коли вони навчатися створювати собі подібних.

Для ініціалізації виконання програми в МП заноситься її початкова адреса (адреса комірки ПЗП чи ОЗП) і подається команда ПУСК.

При роботі МП забезпечує таку послідовність дій:

- 1) витяг даних (коду команди, числа) з пам'яті (ПЗП чи ОЗП);
- 2) розшифрування команди;
- 3) виконання команди і т.д. спочатку.

При цьому, наприклад, щоб переслати число з ПЗП до ОЗП, МП, будучи тактованим (синхронним) пристроєм, у найближчому такті роботи за сигналами тактового генератора задає згідно з програмою на ША код адреси комірки пам'яті ПЗП. Після завершення перехідних процесів, пов'язаних із зарядом паразитних ємностей провідників (ліній), ША (від чого, до речі, значною мірою залежить швидкодія пристрою), у другому такті роботи МП по ШК дає команду на видачу коду з ПЗП на ШД (читання), а у наступному такті – команду по ШК на прийом коду у свій внутрішній регістр пам'яті, що називається акумулятором.

Надалі, у подальших тактах роботи, відповідно задається на ША номер комірки ОЗП та подається по ШК команда запису, і число, нарешті, потрапляє на визначене програмою місце.

З опису цієї послідовності дій зрозуміло, чому програмні пристрої обчислення і керування значною мірою поступаються у швидкодії порівняно із пристроями з жорсткою логікою, де код з одного вузла до іншого, як правило передається за один такт по окремо виділених лініях зв'язку.

На робочому рівні МП реагує на список команд, представлений, як уже зазначалося, у вигляді набору двійкових кодів, що називається машинною програмою.

Таке представлення програм є винятково важким для оперативного сприйняття людиною.

7.3 Шістнадцяткова система числення

Деяке спрощення при написанні і читанні команд дає використання шістнадцяткової системи числення. Вона містить 16 символів, кожному з яких відповідає чотирирозрядне число у двійковому коді: число подається як сума

ступенів числа 16, помножених відповідно на 0, 1, 2, ... , 9, A, B, C, D, E, F:

$$N_{16} = a_1 \cdot 16^0 + a_2 \cdot 16^1 + a_3 \cdot 16^2 + \dots + a_{n+1} \cdot 16^n, \quad (7.1)$$

де $a_i = \{0; 1; 2; 3; \dots; 9; A; B; C; D; E; F\}$.

Табл. 7.1 показує відповідність запису чисел у десятковому, двійковому і шістнадцятковому кодах.

Перетворити двійковий код у шістнадцятковий дуже легко: двійкове число поділяється на групи по чотири розряди, починаючи з молодшого – на тетради. Кожна група замінюється відповідним шістнадцятковим символом. При зворотному перетворенні кожен символ замінюється двійковою тетрадою.

Наприклад, 0101 0111 1101 1111 00012 = 57DF116.

Таблиця 7.1 – Запис чисел у різних кодах (системах числення)

Десятковий код	Двійковий код	Шістнадцятковий код
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Таким чином, шістнадцятковий код, з точки зору застосування для запису програм для МПП, можна вважати більш компактним записом двійкового.

7.4 Асемблер

Програма, записана у шістнадцятковому коді, усе ж таки залишається складною для розуміння.

Виходом є використання мови простого програмування – асемблера. Відповідна програма-ретранслятор переводить слова і фрази програми, представлені у вигляді мнемонік (англійських скорочень назв) асемблера, у машинний код, який потім буде занесено у ІМС ПЗП, що підключаються для

цього до спеціального адаптера машини-програматора.

Приклади мнемонік асемблера та їхня відповідність у шістнадцятковому коді наведені у табл. 7.2.

Відзначимо також, що кожній команді асемблера відповідає мікрокоманда МП – набір дій МП (мікрооперацій), запрограмованих при виробництві МП у його внутрішньому ПЗП. Ініціює виконання мікропрограми пристрій керування МП після отримання коду команди.

При розробленні програм для МПП, що виконує функції керування, складають узагальнений алгоритм виконання програми – послідовність дій з опитування входних даних, їхньої обробки та формування команд керування на відповідних виходах.

Далі алгоритм розбивають на окремі фрагменти і деталізують дії у кожному з них до рівня, що відповідає системі команд вибраного типу МП (як правило – від декількох десятків і до півтори сотень).

Програма спочатку складається у вигляді списку мнемонік, що називається вихідною програмою. Кожному коду присвоюється номер комірки пам'яті. Потім вручну або за допомогою ЕОМ програма переводиться на машинну мову. Тепер вона називається об'єктною програмою і заноситься в ПЗП чи ОЗП МПП.

Таблиця 7.2 – Приклади команд асемблера

Команда	Мнемоніка	Шістнадцятковий код
1. Читання даних з об'єкта з адресою F у внутрішній регістр пам'яті A	INF F	DB FA
2. Порозрядне логічне множення чисел, що знаходяться у внутрішніх регістрах пам'яті A і B	ANA B	A0
3. Додавання числа, що зберігається у внутрішньому регістрі C до числа, що зберігається у внутрішньому регістрі A	ADD C	81
4. Виведення числа, що зберігається у внутрішньому регістрі A до об'єкта, адреса якого знаходиться у внутрішніх регістрах пам'яті B і C	STAX B	02

Використання алгоритмічної мови високого рівня (BASIC, FORTRAN або ін.) недоцільне, оскільки веде до втрати програмістом контролю за використанням ресурсів пам'яті. Це може призвести після її трансляції у машинні коди до значного перевищення оптимально можливого обсягу ПЗП

та втрати швидкодії. А це, у свою чергу, веде до збільшення апаратних витрат (числа ІМС ПЗП) та загрожує втратою можливості керуючого МПП працювати у реальному масштабі часу, коли реакція МПП на зміну ситуації у об'єкті керування повинна відбуватися зі швидкістю, не меншою за ту, з якою відбуваються процеси в об'єкті.

При використанні МПП для керування можливість зміни програми ПЗП заміною ІМС чи їхнім перепрограмуванням (залежно від типу) забезпечує використання даного МПП у різних застосуваннях для вирішення різних задач без будь-яких схемних змін в апаратурі – функція МПП задається його програмуванням.

Поліпшення техніко-економічних характеристик МП дозволило надзвичайно ефективно використовувати їх у різних пристроях цифрової обробки інформації і керування (як у центральних процесорах, так і у периферійних пристроях ЕОМ, у мікрокалькуляторах, у системах керування і регулювання поточними лініями, верстатами-автоматами, роботами-маніпуляторами, стабілізованими джерелами живлення, в електронних пристроях – телевізорах, радіостанціях, вимірювальних приладах, цифрових фотоапаратах, іграшках і т.п.).

Це підняло на новий рівень ефективність роботи технічних пристроїв і систем.

Завдяки використанню МПП ЕОМ у наш час перетворилися у предмет побутової техніки. Взагалі МПП немає різниці, чим керувати: ракетою, світлофором чи пральною машиною. Але звичайно для конкретних застосувань вибирають з широкої номенклатури МП необхідної інформаційної потужності і будують МПП у відповідному конструктивному та кліматичному виконанні.

Потужність МП визначається його здатністю обробляти дані і оцінюється трьома параметрами: довжиною слова даних (кількість розрядів ШД), довжиною слова пам'яті (кількість розрядів ША – визначає кількість об'єктів, до яких може звертатися МП), швидкістю виконання команд.

Питання для самоконтролю

- 1. Поясніть, як ініціюється виконання програми мікропроцесором.*
- 2. Поясніть, у якому вигляді може бути записана програма роботи мікропроцесорного пристрою.*
- 3. Чим, на вашу думку, можна пояснити те, що швидкодія програмованих пристроїв керування принципово менша, ніж у пристроїв із жорсткою логікою?*
- 4. Поясніть, як використовується шістнадцяткова система числення при запису програм мікропроцесорного пристрою. Як перевести число, записане у шістнадцятковому (двійковому) коді у двійковий (шістнадцятковий) код?*
- 5. У чому полягає робота пристрою керування у реальному масштабі часу?*
- 6. Поясніть, у чому полягає ефективність застосування мікропроцесорних пристроїв у порівнянні з іншими пристроями обробки цифрової інформації.*
- 7. Чим визначається інформаційна потужність мікропроцесора?*

Лекція 8

Тема „Програмовані логічні контролери. Мікроконтролери”

1. Програмовані логічні контролери.
2. Мікроконтролери.
3. Структура мікроконтролера.
4. Зіставлення основних етапів створення пристроїв керування на основі цифрових автоматів з жорсткою логікою і програмованих.

Основні поняття

Програмований логічний контролер; багатоплатний ПЛК; одноплатний ПЛК; мікропроцесор; контролер; мікроконтролер; мікрокомп'ютер; центральний процесорний елемент (ЦПЕ); шина адрес/даних; паралельні і один послідовний порти вводу/виводу; запити на переривання; таймер-лічильник.

8.1 Програмовані логічні контролери

Зрозуміло, що продуктивність устаткування та точність виконання технологічних операцій значною мірою залежать від ступеня автоматизації. Також зрозуміло, що у наш час найкраще забезпечують це електронні пристрої керування, особливо цифрові.

Тут слід зазначити, що цифрові пристрої керування реалізуються як автомати з жорсткою логікою або програмовані.

Апаратні методи побудови цифрових пристроїв керування з жорсткою логікою близькі до методів побудови значно поширених раніше аналогових систем керування: для виконання кожної функції використовується окремий елемент або група елементів, і введення нової функції вимагає введення нових елементів. Системи керування, реалізовані такими методами, мають високу швидкодію та надійність, традиційні принципи їхньої побудови не викликають труднощів при розробленні та експлуатації (з точки зору застосовуваного при регламентних та ремонтних роботах обладнання, приладів, з огляду на рівень кваліфікації персоналу). У той самий час вони складаються з великої кількості електронних компонентів (в основному інтегральних мікросхем – ІМС – середнього ступеня інтеграції), громіздкі, мають малу гнучкість за необхідності змін алгоритму керування (або частіше взагалі не дозволяють цього робити) та дорого коштують. Вони вимагають індивідуального розроблення, причому кількість типів функціональних модулів, що створюють функціонально закінчену систему, яка використовується для побудови різних систем керування, виявляється значною. Отже, використання пристроїв керування з жорсткою логікою вимагає у кожному окремому випадку розроблення спеціального унікального пристрою.

Для створення МПП керування технічними пристроями зручно використовувати програмовані мікроконтролери, що випускаються

промисловістю.

Програмовані пристрої будуються на основі процесорів, як правило, у мікрвиконанні – мікропроцесорів.

Необхідно зазначити, що побудова пристрою керування класу, що нами розглядається, на основі мікропроцесорного набору (власне процесор, оперативний та постійний запам'ятовувальні пристрої, інтерфейс та ін.) за величиною апаратних затрат мало відрізняється від його реалізації на ІМС середнього ступеня інтеграції. Вартість розроблення та реалізації такого пристрою виявляється навіть більшою за вартість пристрою з жорсткою логікою на ІМС середнього ступеня інтеграції. Крім того, мікропроцесорні пристрої висувають зовсім інші вимоги до кваліфікації як розробників апаратури, так і до тих, що забезпечують експлуатацію. Але гнучкість програмованих пристроїв керування за необхідності змін алгоритму, що має величезне значення як на етапах розроблення, так і при експлуатації одиночно або малосерійно продукowanego технологічного устаткування, робить їхнє застосування дуже перспективним.

Тут могли б бути у пригоді програмовані логічні контролери.

Зазначимо, що раніше контролерами називали багатопозиційні перемикачі для комутації електричних кіл (наприклад, як ті, що застосовуються для керування трамваєм).

Програмований логічний контролер – це програмований логічний пристрій керування з мовою програмування, доступною для неспеціалістів з інформатики (спочатку, наприклад, мали мову ступінчастих діаграм і призначалися для заміни релейних шаф керування послідовними логічними процесами в умовах промисловості у реальному масштабі часу).

Малогабаритні і гнучкі у програмуванні ПЛК спочатку використовувалися для заміни логічних пристроїв керування, виконаних на електромагнітних реле, що займали цілі шафи. Надалі, з появою мікроелектронних ЦАП і АЦП, їх почали застосовувати і для розв'язання задач, пов'язаних з обчисленням та порівнянням аналогових сигналів.

ПЛК монтуються на одній або декількох платах (у вигляді одного або декількох модулів) і доповнюються пристроями програмування та індикації з клавіатурою та цифровим дисплеєм.

Більшу універсальність мають багатоплатні ПЛК, що являють собою набір модулів: процесора, ПЗП, ОЗП, ЦАП, АЦП, вводу та виводу даних, підсилення потужності, джерела живлення та ін. З цих модулів створюється необхідна у конкретному випадку конфігурація ПЛК, для чого необхідний їх набір встановлюється у так звану корзину – конструкцію з основою, на якій розміщено рознімачі для під'єднання модулів до системної магістралі і джерела живлення. Під'єднання до джерел сигналів та до виконуючих пристроїв об'єкта керування здійснюється окремими кабелями.

Таким чином, багатоплатний ПЛК є універсальним конструктором для створення програмованих пристроїв керування.

Такі ПЛК широко застосовувались до того часу, поки з розвитком мікросхемотехніки стало можливим усі вузли (до 95-98 % слабкострумових

елементів) контролера розміщувати у одній ВІС – отримали одноплатний ПЛК. Силові елементи і джерело живлення при цьому розміщують на окремих платах.

Одноплатні ПЛК мають значно менші габаритні розміри, ніж багатоплатні, простіші у виготовленні і дешевші. Вони більш надійні, оскільки не мають корзини з великою кількістю дорогих і малонадійних рознімачів. Але вони не є універсальними. Тому фірми-виробники, як правило, випускають набори таких ПЛК з різними можливостями для забезпечення керування типовими для деякої галузі техніки об'єктами.

Виходячи з недоліків чи переваг одно- і багатоплатних ПЛК у конкретних застосуваннях використовують як одні, так і інші.

Побудовані на мікропроцесорах контролери мають спрощену систему команд, випускаються серійно і досить дешеві порівняно із вартістю розроблення та виготовлення пристрою керування з жорсткою логікою. Але як пристрої керування технологічного устаткування програмовані логічні контролери значного поширення не знайшли, оскільки були спочатку громіздкими, все ж таки досить дорогими і для багатьох можливих застосувань (наприклад, для керування електронними ключами джерел живлення технологічного устаткування) повільно діючими. Також важливим чинником у обмеженні їхнього поширення була обмежена доступність для пересічного споживача як власне контролерів, так і інформації про користування ними.

8.2 Мікроконтролери

Починаючи з середини 80-х років минулого сторіччя на світовому ринку електронної продукції з'являються ІМС мікроконтролерів. Так, перші восьмирозрядні мікроконтролери були створені фірмою Intel (США) у 1982 році. Зараз такі мікроконтролери у світі продукують більш ніж тридцять фірм (деяку незручність створює несумісність мікроконтролерів різних фірм-виробників).

До речі, подібні ІМС з'явилися наприкінці 80-х років минулого сторіччя і в СРСР. Вони називалися однокристальними електронними обчислювальними машинами (серія КР1816). Знову ж таки, вони були дорогими, малодоступними, бракувало інформації про їхнє застосування.

Мікроконтролери поставляються в основному фірмами Atmel та Microchip Technology Inc. (США), виготовлені у Південно-Східній Азії. Ці мікроконтролери мають особливу популярність також і у розробників електронної техніки малої та середньої складності всього світу. Їхня номенклатура надзвичайно велика. Так, лише фірма Atmel наприкінці 90-х років минулого сторіччя випустила 140 моделей упродовж одного року (різної інформаційної потужності, у корпусах для різного типу монтажу та ін.).

Наведемо сучасне розуміння термінів у цій галузі.

Мікропроцесор є центральним обчислювальним ядром комп'ютерної системи.

Контролер – закінчений електронний пристрій, як правило, виконаний

на друкованій платі і призначений для прийому та обробки сигналів від датчиків, а також для керування зовнішніми пристроями на основі результатів обробки прийнятих сигналів.

Мікроконтролер – програмно керована ІМС, що застосовується для побудови контролерів. Мікроконтролер, окрім ядра, має оперативний та постійний запам'ятовувальні пристрої, таймери, лічильники, канали вводу та виводу інформації, інші пристрої – це самостійна комп'ютерна система, що вміщує процесор, допоміжні схеми та пристрої вводу-виводу даних, розміщені у спільному корпусі.

Мікрокомп'ютер має ще й відео та звуковий процесори.

Нагадаємо, традиційні великі комп'ютери, що будувалися ще з сорокових років минулого століття спочатку на лампах, а потім на ІМС середнього ступеня інтеграції, мали так звану принстонську архітектуру (запропоновану у 1949 році фон Нейманом, що працював у Принстонському університеті), при якій як для програм, так і для даних використовувався єдиний простір пам'яті. Це, внаслідок зменшення апаратних затрат, підвищувало надійність машин, але за рахунок зниження швидкодії.

Мікроконтролери, у яких за рахунок інтегральної технології виконання відпала проблема впливу кількості складових мікроелементів на надійність виробу, мають гарвардську архітектуру (запропоновану тоді ж таки Гарвардським університетом, але відхилену із зрозумілих міркувань), за якої для даних і команд використано окремі області пам'яті та шини. Це забезпечує високу швидкість за рахунок можливості доступу за один цикл як до пам'яті, так і до даних при одночасному виконанні побіжної команди і вибірки наступної.

Мікроконтролери мають низьке споживання (струм живлення становить кілька міліампер за рахунок побудови на К-МОН-структурах), мінімальні габарити (випускаються залежно від інформаційної потужності у 8-, 18-, 20-, 28-, 40-, 44- та 64-вивідних корпусах, найчастіше застосовують контролери у перших чотирьох типах корпусів), потребують мінімум (буквально декілька) зовнішніх дискретних компонентів.

Пам'ять даних у них становить від 36 до 368 байтів, а пам'ять програм від 0,5 до 128 кілобайтів.

Кількість ліній вводу-виводу, що можуть за вибором передавати інформацію у одному з напрямків, становить від 6 до 53.

Робоча частота 1-24 МГц.

Напруга живлення може знаходитись у межах 1,8-6 В (номінальне значення 5 В).

Вони можуть мати у своєму складі аналогові компаратори (наприклад, два), аналого-цифрові перетворювачі (наприклад, у деяких мікроконтролерів AVR фірми Atmel від 4 до 11), таймери (1-2), широтно-імпульсні модулятори (1-8), різні інтерфейси зв'язку із зовнішніми пристроями.

У мікроконтролерів PIC (Peripheral Interface Controller – периферійний контролер інтерфейсу) фірми Microchip є можливість захисту коду програми від несанкціонованих змін або копіювання (один або два біти захисту).

За технологією виконання пам'яті програм є три типи мікроконтролерів:

1) масочно-програмовані – мають найменшу вартість, але застосовуються лише при масовому виробництві, оскільки програмуються на заводі-виробнику ІМС у процесі виготовлення кристалу;

2) з ультрафіолетовим стиранням – дозволяють перепрограмування, але процес цей досить тривалий і після ультрафіолетового опромінення можлива нечітка робота контролера, особливо при зниженій напрузі живлення;

3) із флеш-пам'яттю – дозволяють багаторазове електричне стирання (дозволяють здійснювати розроблення і відпрацьовування програм на готовому пристрої), але у декілька разів дорожчі від контролерів першого типу.

Та все ж вартість масово застосовуваних мікроконтролерів останнього типу сягає лише чотирьох доларів США. Саме такі мікроконтролери є найприйнятнішими для застосування при розробленні пристроїв керування технологічного устаткування, виходячи з таких їхніх особливостей:

1) практично необмежена кількість циклів перепрограмування, що в умовах одиничного та малосерійного виробництва забезпечує виконання налагоджувальних і доводочних робіт, а також оперативну зміну алгоритму керування при експлуатації устаткування;

2) досить високі робочі частоти;

3) наявність простих, дешевих і доступних для самостійної реалізації моделей програматорів;

4) мінімальні габарити, вартість, висока надійність;

5) доступність і вкрай прийнятна вартість;

6) наявність моделей із вбудованими аналоговими компараторами та аналого-цифровими перетворювачами (з розвитком інтегральної технології з'явилася можливість розміщення на одному кристалі як цифрових, так і аналогових елементів);

7) можливість захисту програм від несанкціонованих змін та копіювання;

8) можливість отримання інформації від сучасних інтегральних датчиків та передачі інформації комп'ютеру у відповідних протоколах інтерфейсу;

9) велика кількість доступної довідкової та іншої технічної інформації, програмного забезпечення (для складання програм, їхнього налагодження, занесення у пам'ять мікроконтролера), інформації з типових застосувань та досвіду використання як у друкованому вигляді, так і в INTERNET-джерелах.

На жаль, сучасні мікроконтролери втратили початкову особливість програмованих логічних контролерів – доступність мови програмування для неспеціалістів з інформатики. Хоча процесори мікроконтролерів на відміну від CISC-процесорів (Complex Instruction Set Computer) EOM із складною системою команд відносять до RISC (Reduced Instruct Set Computers)-процесорів, у яких набір виконуваних команд скорочено до мінімуму (залежно від типу мікроконтролера його асемблер має від 33 до 133 команд), їхнє програмування на асемблері особливо складне і доступне власне спеціалістам з мікроконтролерів. Дещо простіше програмування на алгоритмічній мові C++

або на спеціальній версії Basic, але також вимагає впевненого знання комп'ютера та веде до збільшення об'єму програм до тридцяти відсотків.

Це висуває вимоги високої кваліфікації персоналу (вміння користування комп'ютером, програмування), використання при обслуговуванні й ремонті пристроїв керування на основі мікроконтролерів досить дорогого комп'ютерного устаткування.

Але сучасний рівень виробництва все більше висуває саме такі вимоги як до персоналу, так і до власне устаткування.

Окрім того, зрештою, кінцеве зниження строків виробництва устаткування, зниження його вартості, збільшення надійності роботи, зручності експлуатації варті цього.

8.3 Структура мікроконтролера

Отже, перші мікроконтролери з'явилися у 1976 році, коли в одній ВІС стало можливим сумістити процесор, ОЗП, ПЗП і елементи інтерфейсу. На відміну від універсальних мікроЕОМ в мікроконтролерах невелика за розміром пам'ять (декілька сотень байт ОЗП та декілька десятків кілобайт ПЗП) і простий інтерфейс для зв'язку із зовнішніми пристроями. Це впливає із специфіки використання мікроконтролерів. Їх не застосовують в універсальних обчислювальних системах. Мікроконтролери призначені для створення високоефективних і дешевих систем керування і регулювання. Вони реалізують відносно нескладні алгоритми, тому потребують об'ємів пам'яті, на декілька порядків менших, ніж ЕОМ універсального призначення. Їх застосовують у системах керування маніпуляторами, вимірювальних приладів, технологічного устаткування, верстатів, автомобілів, побутової техніки та ін.

Типову структуру мікроконтролера наведено на рис 8.1.

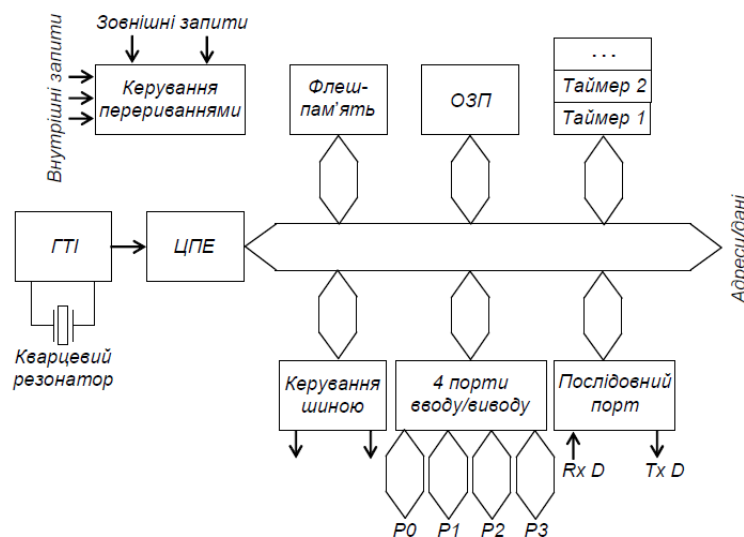


Рисунок 8.1 – Типова структура мікроконтролера

Ємності ОЗП у декілька сотень байт цілком досить для тимчасового зберігання даних та проміжних результатів при виконанні програм керування. Програми зберігаються у флеш-пам'яті ємністю у десятки кілобайт, чого, як правило, також повністю вистачає. За необхідності передбачено можливість нарощування об'ємів пам'яті за рахунок підключення зовнішніх ВІС оперативної та постійної пам'яті.

Зрозуміло, що роботою всіх вузлів керує центральний процесорний елемент ЦПЕ. Синхронізують його роботу імпульси ГТІ із частотою 12-24 МГц, що задається зовнішнім кварцевим резонатором. Це забезпечує високу стабільність частоти, що необхідно при відпрацюванні проміжків часу.

Адреси і дані передаються по шині адрес/даних з розрядністю слів, які обробляються мікроконтролером.

Введення і виведення інформації здійснюється через чотири паралельних і один послідовний порти вводу/виводу. Через порти P0-P3 обмін із зовнішніми пристроями відбувається 8-розрядними шинами. Зараз також випускають 16- і 32- розрядні мікроконтролери. Кожен із портів забезпечує взаємодію з одним пристроєм. Функції ліній шин у різних режимах можуть мати різне значення (введення або виведення).

Через вхід $Rx D$ послідовного порту відбувається прийняття даних у послідовному коді, через вхід $Tx D$ – передача.

Мікроконтролер може обслуговувати декілька як внутрішніх, так і зовнішніх запитів на переривання програми для переходу до виконання більш пріоритетних підпрограм.

Таймери-лічильники призначені для відпрацювання інтервалів часу або підрахунку кількості імпульсів.

Систему команд мікроконтролера орієнтовано на виконання задач керування, тому поряд зі звичайними командами, характерними для всіх мікропроцесорів, є й специфічні. Оскільки всі вузли мікропроцесорної системи тут знаходяться всередині однієї ВІС, то непотрібні звернення до зовнішніх пристроїв. Тому більшість команд короткі за часом виконання, наприклад, близько 1 мкс при тактовій частоті 12 МГц.

Таблиця 8.1 – Зіставлення основних етапів створення пристроїв керування

Пристрій керування з жорсткою логікою	Програмований пристрій керування
1. Розроблення алгоритму керування	
2. Розроблення електричної принципової схеми пристрою керування на основі обраного типу ІМС	2. Розроблення програми роботи пристрою керування на основі обраного типу МПП (наприклад, ПЛК)
3. Створення макету пристрою керування	3. Занесення програми в ОЗП МПП

4. Налагодження макету пристрою керування і відповідне коригування його електричної принципової схеми	4. Налагодження і відповідне коригування програми
5. Розроблення конструкторської і технологічної документації для виробництва пристрою керування	5. Придбання типового МПП (наприклад, ПЛК)
6. Виробництво пристрою керування	6. Занесення програми в пам'ять мікроконтролера або в ІМС постійної пам'яті і встановлення їх в ПЗП МПП
7. Доставлення пристрою керування замовнику	

8.4. Зіставлення основних етапів створення пристроїв керування на основі цифрових автоматів з жорсткою логікою і програмованих

У табл. 8.1 наведено зіставлення основних етапів створення пристроїв керування з жорсткою логікою і програмованих. З неї видно, що при застосуванні типових, таких, що можна придбати готовими, програмованих пристроїв керування, у циклі створення готового пристрою відсутні процедури, пов'язані з розробленням конструкторської і технологічної документації та власне виробництвом. Усі процедури фактично пов'язано виключно з розробленням та налагодженням програми.

На закінчення зазначимо, що специфіка роботи з МПП відносить їх швидше до галузі інформатики, оскільки у більш-менш серйозних застосуваннях вартість розроблення програмного забезпечення становить до 80 % вартості всього пристрою керування. Виходячи з цього ми обмежилися лише наданням загальних понять, особливо у тому, що стосується питань програмування МПП.

Питання для самоконтролю

- 1. Що таке програмований логічний контролер? Які види програмованих логічних контролерів ви знаєте?*
- 2. Поясніть поняття „мікропроцесор”, „контролер”, „мікроконтролер”, „мікрокомп'ютер”.*
- 3. Які способи виконання пам'яті мікроконтролерів ви знаєте?*
- 4. Зазначте особливості мікроконтролерів, що визначають їх широке застосування?*

Система моделювання Electronics Workbench

Завдання:

1. Навчитися працювати з бібліотекою елементів емулятора.
2. Скласти найпростіші електронні схеми. Провести їх діагностику з допомогою осцилографа, мультиметра та інших засобів вимірювання.

Теоретичні відомості

Electronics Workbench (EWB) — розробка фірми Interactive Image Technologies (www.interactiv.com). Особливістю програми EWB є наявність контрольно-вимірювальних приладів; по зовнішньому вигляду, органах керування і характеристиках максимально наближених до їхніх промислових аналогів, що сприяє набуттю практичних навичок роботи з найбільш розповсюдженими приладами: мультиметром, осцилографом, вимірювальним генератором і ін. Програма легко освоюється і досить зручна в роботі. Після складання схеми і її спрощення шляхом оформлення підсхем моделювання починається натисканням звичайного вимикача.

Історія створення програми Electronics Workbench (EWB) починається з 1989 р. Ранні версії програми склалися з двох незалежних частин. За допомогою однієї половини програми можна було моделювати аналогові пристрої за допомогою іншої — цифрові. Такий роздвоєний стан створював певні незручності, особливо при моделюванні змішаних аналого-цифрових пристроїв. У 1996 р. у версії 4.1 ці частини були об'єднані, а через півроку була випущена п'ята версія програми з розширеними можливостями при розмірному аналізі в обсязі програми Micro-Cap V і переробленою бібліотекою компонентів. Додаткові засоби аналізу ланцюгів у EWB 5.0 виконані в типовому для всієї програми ключі — мінімум зусиль з боку користувача. Подальшим розвитком EWB є програма EWB Layout, призначена для розробки друкованих плат, а також EWB версії 6.02.

Програма EWB має наступність знизу вгору, тобто всі схеми, створені у версіях 3.0 і 4.1, можуть бути промодельовані у версії 5.0. Крім того EWB дозволяє також моделювати пристрої, для яких завдання на моделювання підготовлено в текстовому форматі SPICE, чим забезпечується сумісність із програмами Micro-Cap і PSpice.

Програма EWB 4.1 розрахована для роботи в середовищі Windows 3.xx або 95/98 і займає близько 5 Мбайт дискової пам'яті, EWB 5.0 — у середовищі Windows 95/98 і NT 3.51, необхідний обсяг дискової пам'яті — близько 16 Мбайт. Для розміщення тимчасових файлів потрібно додатково 10...20 Мбайт вільного простору.

1 Структура вікна і система меню

Робочі вікна програм версій 4.1 і 5.0 показані на рис. 1.1 і 1.2 відповідно.

Вікно програми EWB 4.1 (рис. 1.1) містить поле меню, лінійку контрольно-вимірювальних приладів і лінійку бібліотек компонентів, одна з яких у розгорнутому виді показана в лівій частині вікна. У робочому полі програми розташовується моделююча схема з підключеними до неї іконками контрольно-вимірювальних приладів і короткий опис схеми (description), на жаль, тільки англійською мовою. При необхідності кожний із приладів може бути розгорнутий для установки режимів його роботи і спостереження результатів. Лінійки прокручування використовуються тільки для переміщення схеми.

Вікно програми EWB 5.0 (рис. 1.2) відрізняється додатковим меню Analysis, наявністю лінійки інструментів і більш компактним представленням бібліотек у розгорнутому виді. Крім того, лінійка контрольно-вимірювальних приладів розташована в одному полі з бібліотеками компонентів. Варто згадати також про наявність підсвічуваних підказок призначення всіх кнопок.

Подальший виклад матеріалу будемо вести паралельно по програмах версій 4.1 і 5.0, причому спочатку буде розглядатися версія 4.1, як більш проста і доступна, а потім у плані відмінних ознак і додаткових можливостей — версія 5.0.

Розглянемо команди меню програми EWB 4.1 у порядку їхнього розташування на рис. 1.1.

1.1 Меню File

Меню File призначено для завантаження і запису файлів, одержання твердої копії обраних для друкування складових частин схеми, а також для імпорту/експорту файлів у форматах інших систем моделювання і програм розробки друкованих плат.

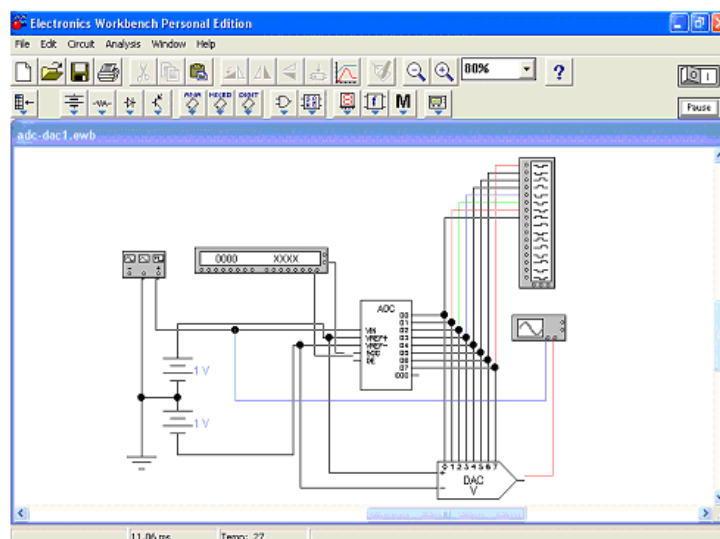


Рисунок 1.1 - Вікно програми EWB 4.1

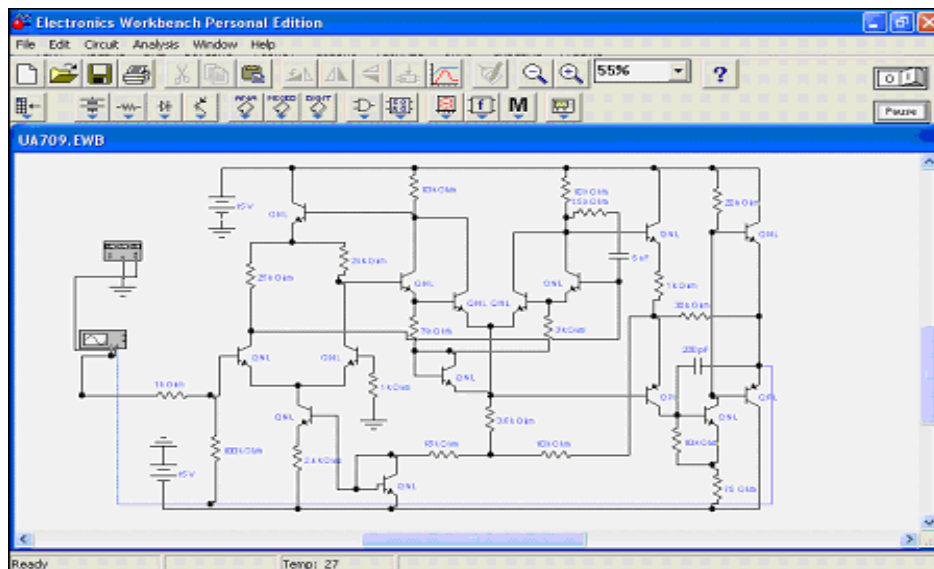


Рисунок 1.2 - Вікно програми EWB 5.0

1. Перші чотири команди цього меню: New (Ctrl+N), Open... (Ctrl+O), Save (Ctrl+S), Save As... — типові для Windows команди роботи з файлами і по цьому пояснень не вимагають. Для цих команд у п'ятій версії існують кнопки (іконки) зі стандартним зображенням. Схемні файли програми EWB мають наступні розширення: .cd3 — аналогові і цифрові схеми для EWB 3.0, .ca4 — аналого-цифрові схеми для EWB 4.1, .ewb — аналого-цифрові схеми для EWB 5.0.

2. Revert to Saved... — стирання всіх змін, внесених у поточному сеансі редагування, і відновлення схеми в первісному вигляді.

3. Print... (CTRL+P) — вибір даних для виведення на принтер: Schematic — схеми (опція включена за замовчуванням); Description — опису до схеми;

Part list — переліку виведених на принтер документів; Label list — списку позначень елементів схеми; Model list — списку наявних у схемі компонентів;

Subcircuits — підсхем (частин схеми, що є закінченими функціональними вузлами й позначаємих прямокутниками з назвою усередині);

Analysis options — переліку режимів моделювання; Instruments — списку приладів (див. розділ 3).

У цьому ж підменю можна вибрати опції друкування (кнопка Setup) і відправити матеріал на принтер (кнопка Print). У програмі EWB 5.0 передбачена також можливість зміни масштабу виведених на принтер даних у межах від 20 до 500%.

4. Print Setup... — налаштування принтера.

5. Exit (ALT + F4) — вихід із програми.

6. Install... — установка додаткових програм із гнучких дисків

7. Import from SPICE — імпорт текстових файлів опису схеми і завдання на моделювання у форматі SPICE (з розширенням .cir) і автоматична побудова

схеми по її текстовому опису.

8. Export to SPICE — складання текстового опису схеми і завдання на моделі у форматі SPICE.

9. Export to PCB — складання списків з'єднань схеми у форматі OrCAD і інших програм розробки друкованих плат.

Одноименне меню програми EWB п'ятої версії відрізняється від розглянутого тим, що в підменю Import/Export передбачені можливості обміну даними з програмою розробки друкованих плат EWB Layout.

1.2 Меню Edit

Меню Edit дозволяє виконувати команди редагування схем і копіювання екрана.

1. Cut (CTRL+X) — стирання (вирізання) виділеної частини схеми зі збереженням її в буфері обміну (Clipboard). Виділення одного компонента здійснюється кліком миші на зображенні (значку) компонента. Для виділення частини схеми або декількох компонентів необхідно поставити курсор миші в лівий кут уявного прямокутника, що охоплює виділювану частину, натиснути ліву кнопку миші і, не відпускаючи її, простягнути курсор по діагоналі цього прямокутника (косинця), контури якого з'являються вже на початку руху миші, і потім відпустити кнопку. Виділені компоненти зафарбовуються в червоний колір.

2. Copy (CTRL+C) — копіювання виділеної частини схеми в буфер обміну.

3. Paste (CTRL+V) — вставка вмісту буфера обміну на робоче поле програми. Оскільки в EWB немає можливості поміщати імпортоване зображення схеми або її фрагмента в точно зазначене місце, то безпосередньо після вставки, коли зображення ще є відзначеним (виділене червоним) і може виявитися накладеним на створювану схему, його можна перемістити в потрібне місце клавішами курсору або схопивши мишею за один з відмічених компонентів. У такий же спосіб переміщуються і попередньо виділені фрагменти вже наявної на робочому полі схеми.

4. Delete (Del) — стирання виділеної частини схеми.

5. Select All (CTRL+A) — виділення всієї схеми.

6. Copybits (CTRL+I) — команда перетворює курсор миші в хрестик, яким за правилом прямокутника можна виділити потрібну частину екрана, після відпускання лівої кнопки миші виділена частина копіюється в буфер обміну, після чого його вміст може бути імпортований в будь-який додаток Windows. Копіювання всього екрана виробляється натисканням клавіші Print Screen; копіювання активної в даний момент частини екрана, наприклад, діалогового вікна — комбінацією Alt+Print Screen. Перераховані команди дуже зручні при підготовці звітів по моделюванню, наприклад, при оформленні лабораторних робіт.

7. Show Clipboard — показати вміст буфера обміну.

Однотименне меню EWB 5.0 аналогічно розглянутому, за винятком назви команди копіювання екрана в п. 6. Вона називається Copy As Bitmap і, нажаль, не має клавіатурного дублювання (комбінації CTRL+I для EWB 4.1), що в деяких випадках ускладнює можливість копіювання елементів схеми.

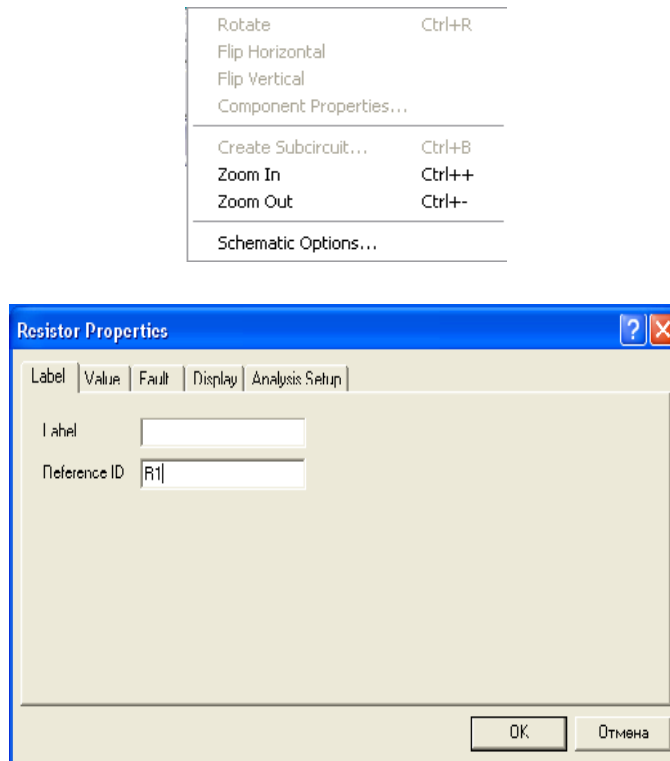


Рисунок 1.3 - Вікно вводу позиційного значення резистора

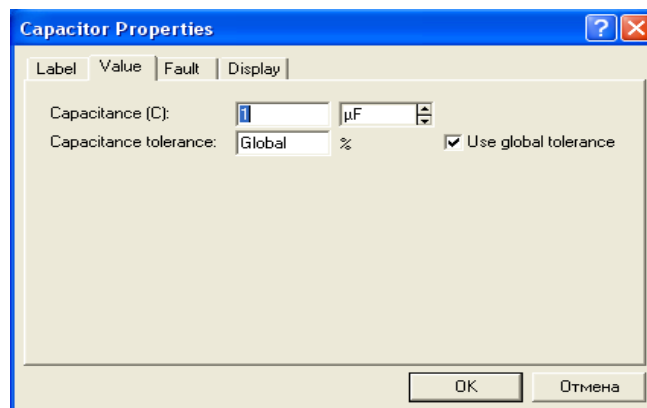


Рисунок 1.4 - Вікно вводу номінального значення компонента

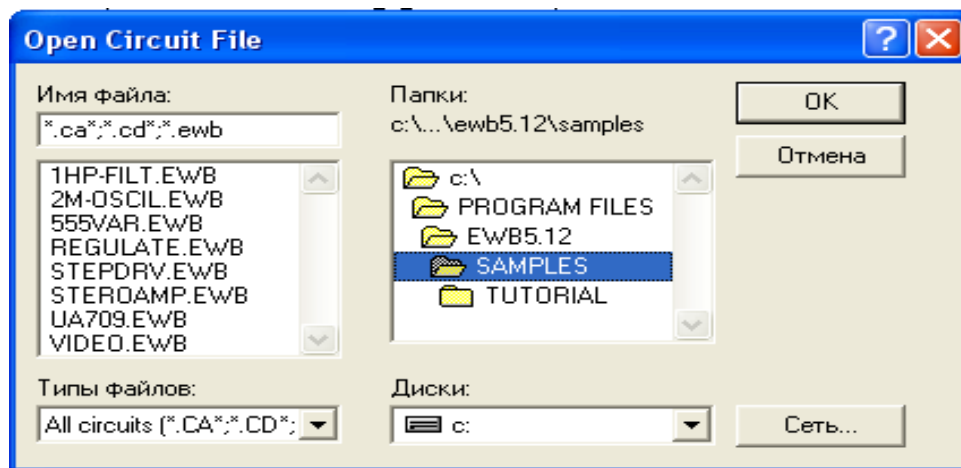


Рисунок 1.5 - Меню вибору моделі операційного підсилювача з можливістю редагування його параметрів

6. Model... (CTRL+M) — вибір моделі компонента (напівпровідникового приладу, операційного підсилювача, трансформатора й ін.); команда виконується також подвійним клацанням мишкою по компоненті. У меню команди (рис. 1.5) вибираються:

Library — перелік бібліотек, у яких знаходяться компоненти обраного типу;

Model -перелік моделей компонентів обраної бібліотеки;

Vd-після натискання цієї кнопки викликається діалогове вікно з параметрами вибраної моделі, показане для операційного підсилювача на мал. 1.6. При необхідності редагування доцільно по команді New Library створити окрему бібліотеку (щоб не псувати параметри бібліотечного компонента), куди переноситься компонент, що редагується, за допомогою команд:

Copy — копіювання відзначеного в колонку Model компонента в буфер обміну;

Paste — вставка скопійованої в буфер обміну моделі компонента в обрану колонку в Library бібліотеку (у тому числі і знову створену) з наступним редагуванням її параметрів без зміни характеристик компонента основної бібліотеки;

Rename — перейменування відзначеної моделі компонента.

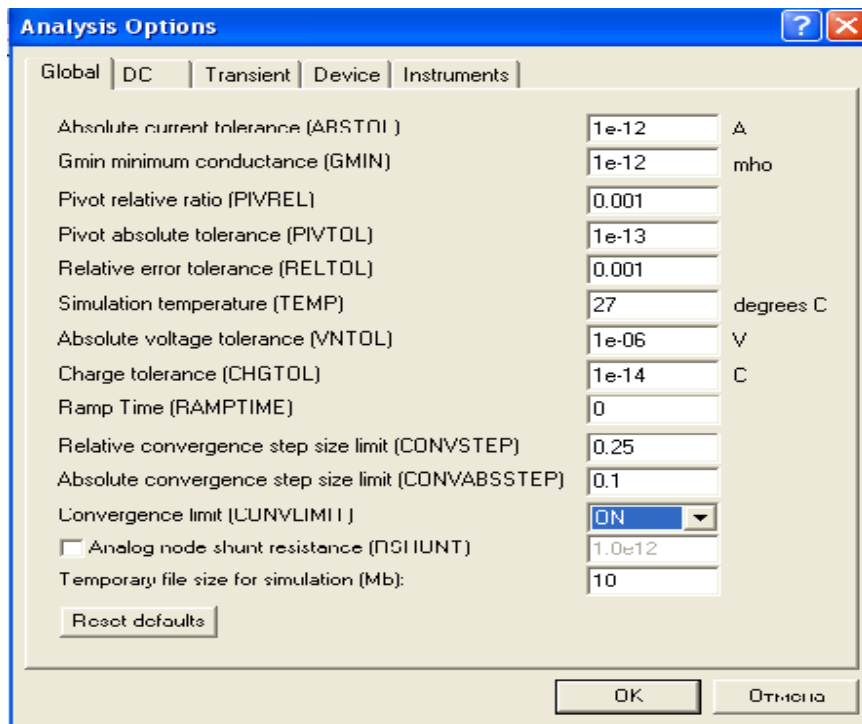


Рисунок.1.6 - Вікно редагування параметрів моделі операційного підсилювача

Робота з меню, як і у всіх інших подібних випадках, закінчується натисканням кнопок Accept або Cancel — зі збереженням або без збереження внесених змін. При створенні бібліотеки моделей вітчизняних компонентів доцільно діяти в наступній послідовності:

- створити бібліотеку, наприклад, під ім'ям `usr_lib`;
- скопіювати в цю бібліотеку модель компонента, найбільш близького по параметрах до вітчизняного компонента;
- перейменувати скопійовану модель, привласнивши їй, наприклад, ім'я K140UD5 (латинська транскрипція ДО140УД5);
- при необхідності відредагувати значення параметрів перейменованої моделі, використовуючи дані каталогів вітчизняних мікросхем або літературних джерел.

7. Zoom (CTRL+Z) — розкриття (розгортання) виділеної підсхеми або контрольно-вимірювального приладу, команда виконується також подвійним клацанням миші по іконці компонента або приладу.

8. Rotate (CTRL+R) — обертання виділеного компонента; більшість компонентів повертаються по годинниковій стрілці (у EWB 5.0 — проти) на 90° при кожному виконанні команди, для вимірювальних приладів (амперметр, вольтметр і ін.) міняються місцями клеми підключення; команда використовується при підготуванні схем. У готовій схемі користуватися командою недоцільно, оскільки це найчастіше приводить до плутанини, — у таких випадках компоненти потрібно спочатку відключити, а потім обертати.

9. Fault (CTRL+F) — імітація несправності виділеного компонента шляхом введення:

Leakage — опору витоку;

Short — короткого замикання;

Open — обриву;

None — несправність відсутня (включена за замовчуванням).

10. Subcircuit... (CTRL+B) — перетворення попередньо виділеної частини схеми в підсхему. Виділяема частина схеми повинна бути розташована таким чином, щоб у виділену область не потрапили не стосовні до неї провідники і компоненти. У результаті виконання команди викликається діалогове вікно, у рядку Name якого вводиться ім'я підсхеми, після чого можливі наступні варіанти:

Copy from Circuit — підсхема копіюється з зазначеною назвою в бібліотеку Custom без внесення змін у вихідну схему;

Move from Circuit — виділена частина вирізується з загальної схеми й у виді підсхеми з привласненням їй ім'ям копіюється в бібліотеку Custom;

Replace in Circuit — виділена частина замінюється у вихідній схемі підсхемою із привласненням їй ім'ям з одночасним копіюванням у бібліотеку Custom.

Для перегляду або редагування підсхеми потрібно двічі клацнути мишею по її значку. Редагування підсхеми здійснюється за загальними правилами редагування схем. При створенні додаткового виходу необхідно з відповідної точки підсхеми курсором миші простягнути провідник до краю її вікна до появи незафарбованої прямокутної контактної площадки, після чого відпустити ліву кнопку миші. Для видалення виходу необхідно курсором миші схопитися за його прямокутну площадку в краю вікна підсхеми і винести її за межі вікна.

Як приклад на рис. 1.7. приведена підсхема sensor підсилювача, що неінвертує, на ОП, використовуваного у фільтрі (див. рис. 1.22). Тут доречно відзначити, що введення в підсхему "землі" недоцільно, оскільки при великій кількості таких підсхем сповільнюється процес моделювання. Тому в схемі на рис. 1.7 для резистора R передбачений окремий вивід, що у схемі фільтра на рис. 1.22 заземлюється.

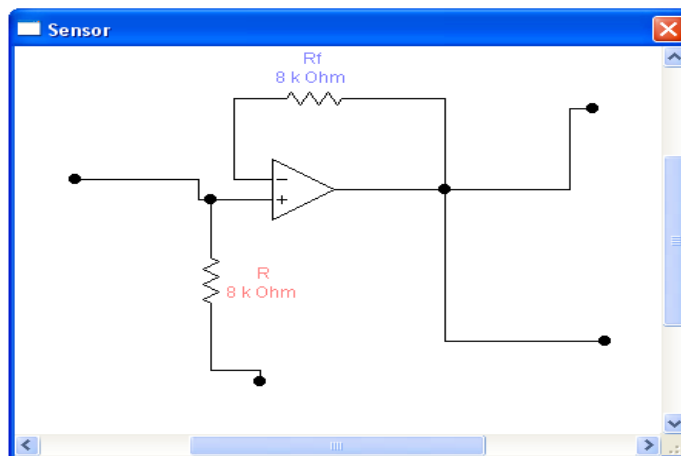


Рисунок.1.7 - Підсхема сенсор

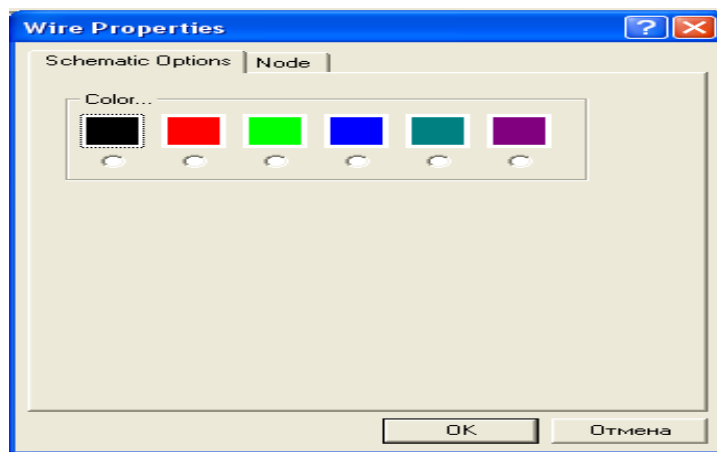


Рисунок.1.8 - Вибір кольору провідника

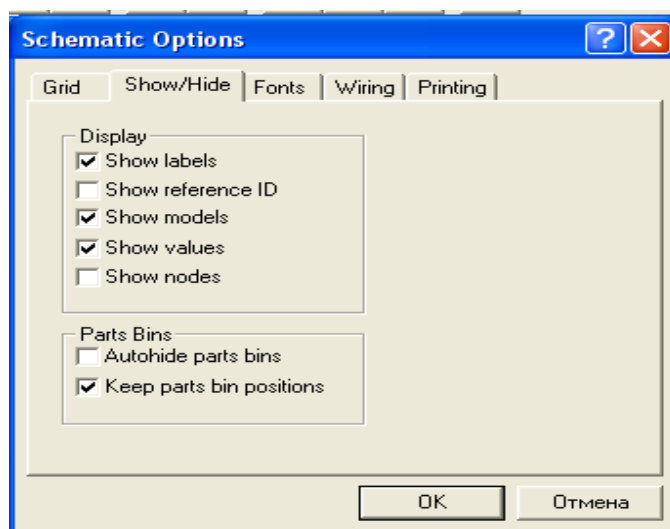


Рисунок.1.9 - Вибір елементів оформлення схеми

11. Wire Color. — зміна кольору попередньо виділеного провідника (виділений провідник товщає). Більш простий спосіб виконання команди — подвійне клацання мишею на провіднику, після чого в меню (рис. 1.8) вибирається один із шести пропонованих кольорів. Необхідність розцвічення особливо важлива для провідників, що з'єднують контрольні точки (вузли) схеми з осцилографом або логічним аналізатором, — у цьому випадку колір провідника визначає колір осцилограми.

12. Preferences... (CTRL+E) — вибір елементів оформлення схеми відповідно до меню (див. рис. 1.9), у якому опції мають наступні призначення:

Show grid — показувати сітку для зручності малювання схеми (за замовчуванням ця опція виключена, інші включені); опція активна тільки при включеній опції Use grid — використовувати сітку;

Show labels — показувати позиційні позначення компонентів, наприклад, C1, C2 для конденсаторів;

Show models — показувати найменування моделей компонентів, наприклад, типів транзисторів;

Show values — показувати номінали компонентів, наприклад, опори резисторів.

Слід зазначити, що в програмі EWB 3.0 у розглянутій команді можна було використовувати пароль. Наприклад, виведення на екран номінальних значень компонентів або їхніх типів можливе тільки при знанні пароля. А це дозволяло, у свою чергу, навмисно вводити сховані в такий спосіб несправності компонентів і відпрацьовувати учнями навички їхнього пошуку.

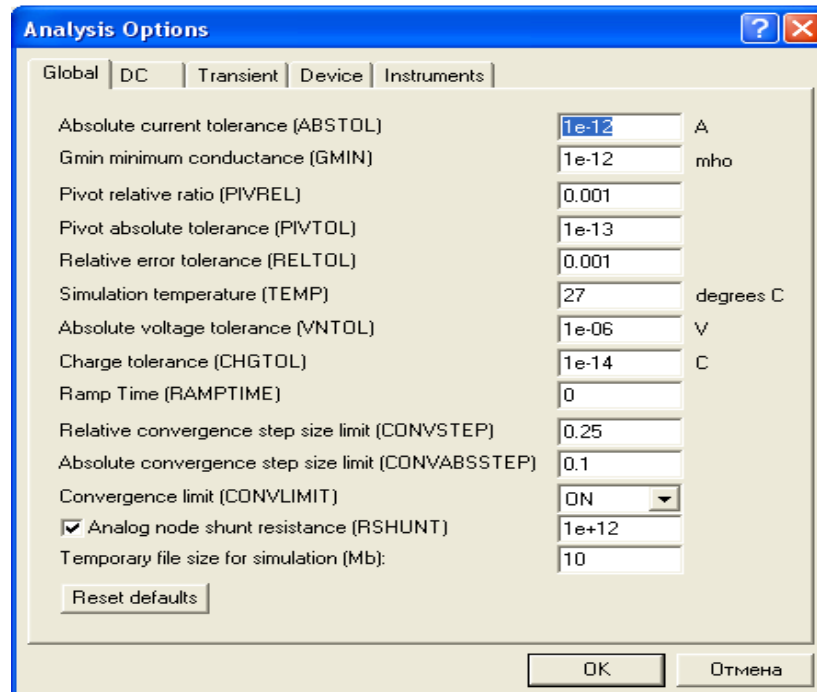


Рисунок 1.10 - Вікно вибору режимів моделювання

13. Analysis Options... (RL+Y) — вибір режимів моделювання в діалоговому вікні (див. рис. 1.10) з установкою наступних опцій:

Transient — розрахунок перехідних процесів після включення джерела живлення (результати представляються на екрані осцилографа в графічному виді);

Steady-state — розрахунок стаціонарного режиму схеми (режиму по постійному струму);

Assume linear operation — при розрахунку прийняти лінеаризовану модель активних компонентів (Active Component Simulation);

Pause after each screen — пауза після заповнення екрана осцилографа (Oscilloscope Display);

Store results for all nodes — збереження (запам'ятовування) результатів моделювання для всіх контрольних точок (вузлів) схеми;

Tolerance — задання припустимої похибки моделювання (за замовчуванням 1%); чим менша похибка моделювання, тим більші витрати часу на моделювання;



Time domain points per cycle — вибір кількості відліків відображуваного на екрані осцилографа сигналу (за замовчуванням — 100 точок на період, може бути збільшене у 100 разів). Зі збільшенням кількості відліків форма сигналу розраховується більш точно при одночасному уповільненні процесу




моделювання; у деяких випадках занижене (установлене за замовчуванням) значення параметра може привести до істотних перекручувань результатів моделювання;

Code Analysis points per cycle — вибір кількості розрахункових точок для відображення результатів моделювання на екрані вимірювача амплітудно-частотних і фазово-частотних характеристик; збільшення кількості точок дозволяє зменшити похибку від дискретності установки візирної лінійки по осі X (осі частот).

Temporary file size for simulation [Mb] — розмір тимчасового файлу для збереження результатів моделювання (за замовчуванням 10 Мбайт), при необхідності може бути змінений.

Меню Circuit у п'ятій версії EWB помітно відрізняється від розглянутого. Відмінності полягають у наступному:

1. Виключено команди по пп. 1, 2 і 3, вони перенесені в меню Analysis і можуть виконуватися також натисканням кнопок  .

2. Введено додаткові команди керування розташуванням графічного зображення компонентів: Flip Horizontal — дзеркальне відображення компонента по горизонталі і Flip Vertical — те ж, але по вертикалі. Команди Rotate, Flip Horizontal і Flip Vertical можуть бути виконані також натисканням кнопок   .

Введені доповнення дуже корисні, оскільки істотно розширюють можливості оформлення схем. Наприклад, у версіях 3.0 і 4.1 не вдавалося відповідним чином розташувати на схемі транзистори, конденсатори змінної ємності й інші компоненти у відповідності зі стандартами. Як приклад на рис. 1.11 приведені результати застосування зазначених команд для перетворення графічного зображення конденсатора змінної ємності: на рис. 1.11, а — зображення конденсатора у вихідному стані, на рис. 1.11, б — після застосування команди Rotate (поворот на 90 проти годинникової стрілки), на рис. 1.11, в — після застосування команди Flip Horizontal (дзеркальне відображення по горизонталі) і на рис. 1.11, г — після застосування команди Flip Vertical (дзеркальне відображення по вертикалі).

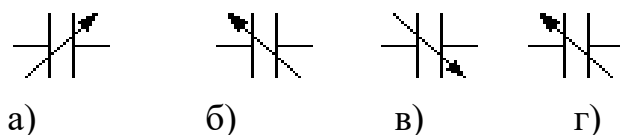



Рисунок 1.11 - Результати застосування команд для перетворення графічного зображення конденсатора змінної ємності

3. Команди по пп. 4, 5 і 6 об'єднані в команду Component Properties (властивості компонента). Команда виконується також після подвійного клацання по компоненту або натискання кнопки , При виконанні команди відкривається діалогове вікно (рис. 1.12) з декількох закладок. Зміст закладки Label відповідає команді по п. 4 за винятком рядка Reference ID: у ній

вказується позиційне позначення компонента, використовуване надалі при виконанні команд меню Analysis. При виборі закладки Value задаються номінальний опір компонента (резистора), значення лінійного (TC1), квадратичного (TC2), температурних коефіцієнтів опору. З врахуванням цих параметрів дійсний опір резистора R_d визначається виразом:

$$R_d = R(1 + TC1(T - T_n) + TC2(T - T_n)^2)$$

де R — номінальний опір резистора; $T_n = 27^\circ\text{C}$ — номінальна температура; T — поточне значення температури резистора.

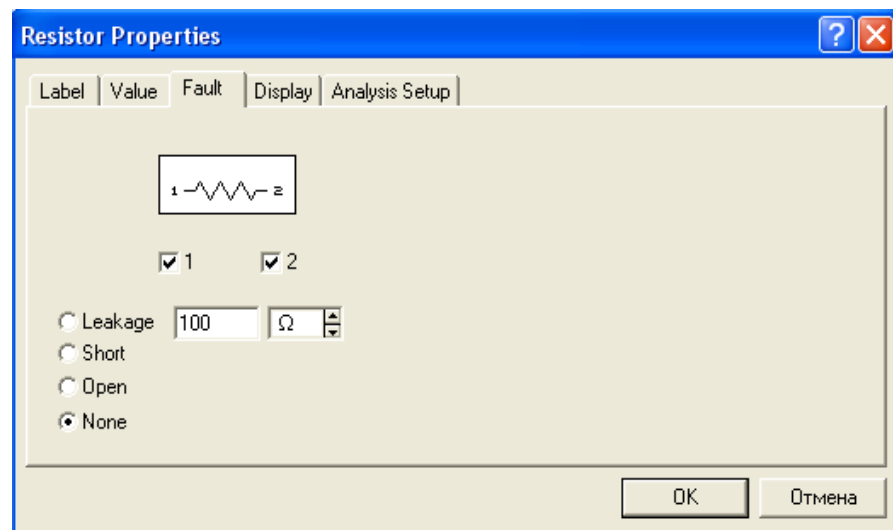


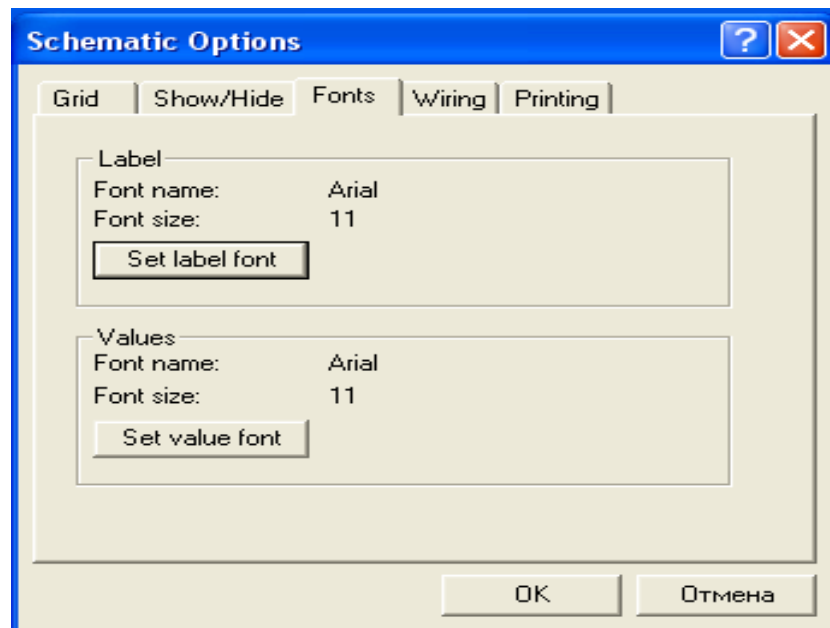
Рисунок.1.12. - Діалогове вікно команди Component (зкладка Fault)

При виборі закладки Fault (рис. 1.12, а) приводяться умови моделювання по п. 9 і набір висновків компонента з опцією на кожен висновок, що дозволяє вибірково імітувати ту або іншу несправність. Наприклад, якщо потрібно імітувати порушення контакту виходу 1 резистора, то в цьому випадку включаються опції 1 і Open (відкрито — обрив). Уведення таких дефектів у схему дозволяє відпрацьовувати учнями навички пошуку і локалізації несправностей.

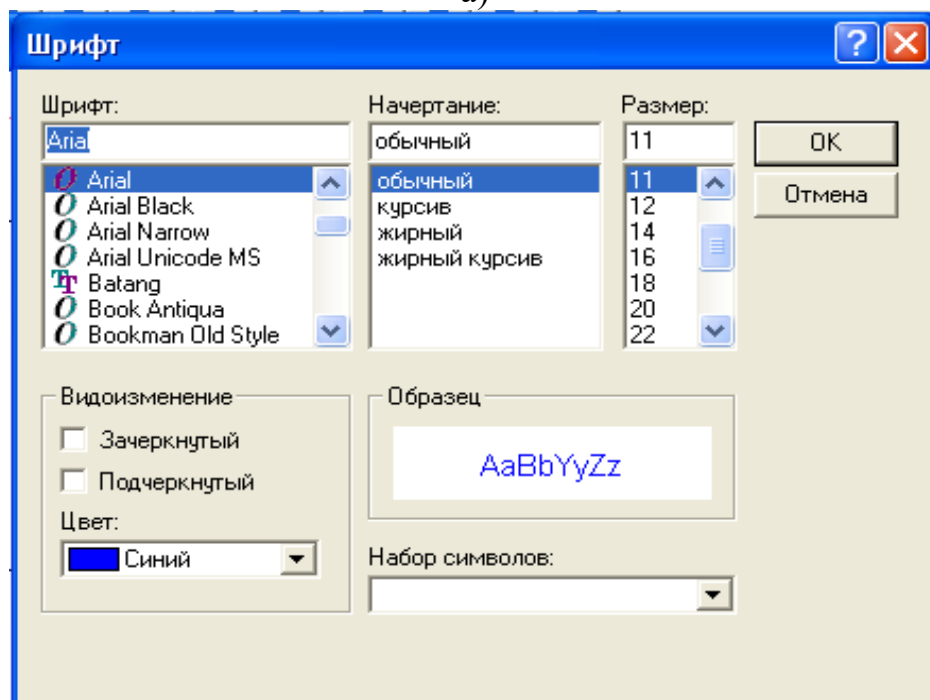
При виборі закладки Display задається характер виведення на екран позначень компонента; при виборі опції Use Schematic Options global setting використовуються установки, прийняті для всієї схеми, у протилежному випадку використовується індивідуальне налаштування виведення на екран позиційного позначення і номінального значення для кожного компонента.

Діалогове вікно при виборі закладки Analysis Setup дозволяє установити температуру для кожного компонента індивідуально або використовувати її номінальне значення, прийняте для всієї схеми (Use global temperature).

Для активних компонентів меню команди Component Properties містить меню Models, за допомогою якого вибирається тип бібліотечного компонента, редагуються його параметри, створюється нова бібліотека і виконуються інші команди по п. 6 (див. рис.1.6).



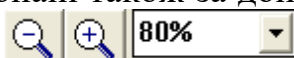
а)



б)

Рисунок 1.13 - Вікно команди Schematic Options, закладки Fonts (а) з установкою типу шрифта і його атрибутів (б)

Введено додаткові команди: масштабування схеми: збільшення Zoom In і зменшення Zoom Out із указанням масштабу в діапазоні 50 — 200%. Такі команди можуть бути виконані також за допомогою мнемонічних засобів з її стандартним позначенням:



5. Замість команди Preferences введена команда Schematic Options, діалогове вікно якої (рис. 1.13) складається з ряду закладок. Закладки Grid у Show/Hide, у порівнянні з EWB 4.1, містять наступні додаткові опції:

Show Nodes — показувати нумерацію нод — усіх точок з'єднання компонентів;

Autohide part bins — за замовчуванням не показувати склад бібліотеки компонентів використовуваної в даній схемі;

Keep parts bin positions — зберігати положення використовуваної бібліотеки компонентів на екрані при оформленні схеми. Звичайно вибір нової бібліотеки компонентів приводить до вимикання попередньої. Для збереження на екрані зразу декількох бібліотек їх необхідно рознести по екрані, при цьому їх положення при виборі нової бібліотеки залишаться незмінним.

При виборі закладки Fonts (рис. 1.13, а) можна установити тип (Font name і розмір (Font size) шрифту роздільно для позначення компонента (кнопка Set label font) і номінального значення його параметра (кнопка Set value font). У якості прикладу на рис. 1.13, б показано вікно установки типу і розміру шрифту позначення компонента. Помітимо, що для міток (і тільки) можна вибрати шрифти кирилиці Ma Cyr і Mt Cyr (для EWB 5.0c).

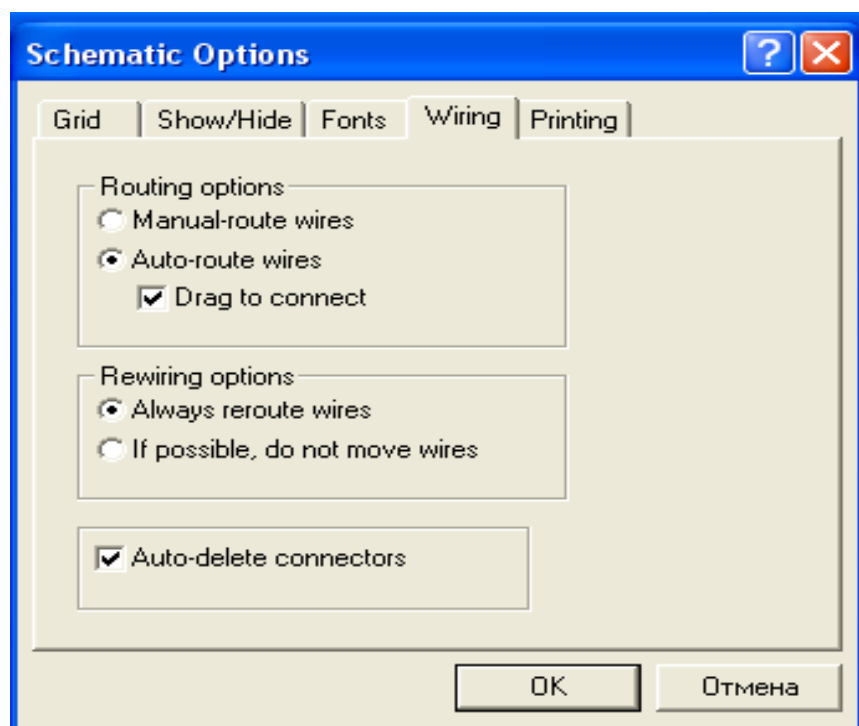


Рисунок 1.14 - Вікно команди Schematic Options, закладка Wiring

Вікно команди Schematic Options при виборі закладки Wiring (рис. 1.14) містить опції, зв'язані з прокладанням провідників на схемі й організацією їхніх взаємних з'єднань (Routing options), видаленням провідників (Rewiring options) і з'єднань (Auto-delete connectors — автоматичне видалення невикористовуваних з'єднань, наприклад, що дублюють один одного). При виборі закладки Printing можна установити масштаб виведеної на принтер інформації. Варто помітити, що в мережній версії EWB 5.0z передбачена (за аналогією з EWB 3.0) установка пароля і режиму "тільки для читання" (меню Circuit/Restrictions..., закладка General), що обмежує доступ до інформації з введеними в схему несправностями, за схемою підсхеми, по

використовуваним у схемі моделям компонентів і їхніх номінальних значень (наприклад, опорам резисторів) (зкладка Componets) і можливостям вибіркового використання розглянутих нижче методів аналізу з меню Analysis (однойменна закладка у вікні меню Cicut/Restrictions...).

2 Створення схем

У даному розділі розглядається процес підготовки схем, склад бібліотек компонентів EWB 4.1 і 5.0 і їхні короткі характеристики.

2.1 Технологія підготовки схем

Перш ніж створювати креслення принципової схеми засобами програми EWB, необхідно на листі паперу підготувати її ескіз зі зразковим розташуванням компонентів і з урахуванням можливості оформлення окремих фрагментів у виді підсхем. Доцільно також ознайомитися з бібліотекою готових схем програми для вибору аналога (прототипу) або використання наявних рішень у якості підсхем.

У загальному випадку процес створення схеми починається з розміщення на робочому полі EWB компонентів з бібліотек програми відповідно до підготовленого ескізу. Одинадцять розділів бібліотеки програми EWB 4.1 по черзі можуть бути викликані за допомогою меню Window або за допомогою іконок, розташованих під лінійкою контрольно-вимірювальних приладів (див. рис. 1.1). Каталог обраної бібліотеки розташовується у вертикальному вікні праворуч або ліворуч від робочого поля (встановлюється в будь-яке місце перетаскуванням стандартним способом — за шапку заголовка). Для відкриття каталогу потрібної бібліотеки необхідно підвести курсор миші до відповідної іконки і натиснути один раз її ліву кнопку, після чого сірий фон іконки міняється на жовтий. Необхідний для створення схеми значок (символ) компонента переноситься з каталогу на робоче поле програми рухом миші при натиснутій лівій кнопці, після чого кнопка відпускається (для фіксування символу) і здійснюється подвійне клацання по значку компонента. У діалоговому вікні, що розкривається, встановлюються необхідні параметри (опір резистора, тип транзистора і т.д.) і вибір підтверджується натисканням кнопки Ассерпт або клавіші Enter. На цьому етапі необхідно передбачити місце для розміщення контрольних точок і іконок контрольно-вимірювальних приладів.

Якщо в схемі використовуються компоненти однакового номіналу (наприклад, резистори з однаковим опором), то номінал такого компонента рекомендується задати безпосередньо в каталозі бібліотеки і потім переносити компоненти в потрібній кількості на робоче поле. Для зміни номіналу компонента необхідно два рази клацнути мишею по символу його графічного зображення й у полі, що розкривається, цього вікна і внести зміни.

При розміщенні компонентів схеми на робочому полі програми EWB 5.0 можна скористатися динамічним меню, описаним наприкінці розділу.

Після розміщення компонентів здійснюється з'єднання їхніх виводів провідниками. При цьому необхідно враховувати, що до виводу компонента можна підключити тільки один провідник. Для виконання підключення курсор миші підводиться до виводу компонента і після появи прямокутної площадки синього кольору натискається ліва кнопка і провідник, що з'являється при цьому, протягується до виводу іншого компонента до появи на ньому такої ж прямокутної площадки, після чого кнопка миші відпускається, і з'єднання готове. При необхідності підключення до цих виводів інших провідників у бібліотеці Passive вибирається крапка (символ з'єднання) і переноситься на раніше встановлений провідник. Щоб крапка почорніла (спочатку вона має червоний колір), необхідно клацнути мишею по вільному місцю робочого поля. Якщо ця крапка дійсно має електричне з'єднання з провідником, то вона цілком зафарбовується чорним кольором. Якщо на ній видний слід від провідника, що перетинає її, то електричного з'єднання немає і крапку необхідно установити заново. Після вдалої установки до крапки з'єднання можна підключити ще два провідники. Якщо з'єднання потрібно розірвати, курсор підводиться до одного з виводів компонентів або крапки з'єднання і з появою площадки натискається ліва кнопка, провідник переноситься на вільне місце робочого поля, після чого кнопка відпускається. Якщо необхідно підключити вивід до наявного на схемі провідника, то провідник від виводу компонента курсором підводиться до зазначеного провідника і після появи крапки з'єднання кнопка миші відпускається. Слід зазначити, що прокладання сполучних провідників здійснюється автоматично, причому перешкоди — компоненти й інші провідники — огинаються по ортогональних напрямках (по горизонталі або вертикалі).

Крапка з'єднання може бути використана не тільки для підключення провідників, але і для введення написів (наприклад, вказання величини струму в провіднику, його функціонального призначення і т.п.). Для цього необхідно двічі клацнути по крапці й у вікні, що розкрилося, ввести необхідний запис (не більше 14 символів), причому запис можна зміщати вправо шляхом уведення ліворуч потрібної кількості пробілів. Ця властивість може бути використана й у тому випадку, коли позиційне позначення компонента (наприклад C1, R10) накладається на провідник що йде поруч або інші елементи схеми.

Якщо необхідно перемістити окремий сегмент провідника, до нього підводиться курсор, натискається ліва кнопка і після появи у вертикальній або горизонтальній площині подвійного курсору здійснюються потрібні переміщення.

Підключення до схеми контрольно-вимірювальних приладів здійснюється аналогічно. Причому для таких приладів, як осцилограф або логічний аналізатор, з'єднання доцільно проводити кольоровими провідниками, оскільки їхній колір визначає колір відповідної осцилограми. Кольорові провідники доцільні не тільки для позначення провідників однакового функціонального призначення, але і для провідників, що знаходяться в різних частинах схеми (наприклад, провідники, шини даних до

і після буферного елемента). Приклади такого оформлення можна знайти в каталогах готових схем (див. файл adc-dacl.ca4).

При позначенні компонентів необхідно дотримуватись рекомендацій і правил, передбачених ЄСКД (єдиною системою конструкторської документації). Що стосується пасивних компонентів, то при виборі їхніх позначень особливих труднощів не виникає. Труднощі виникають при виборі активних елементів —: мікросхем, транзисторів і т.п., особливо при необхідності використання компонентів вітчизняного виробництва, коли потрібно установити точну відповідність функціональних позначень виводів і параметрів закордонних і вітчизняних компонентів. Для полегшення цієї задачі можна скористатися таблицями відповідності закордонних і вітчизняних компонентів.

При імпортуванні в створювану схему іншої схеми або її фрагментів доцільно діяти в наступній послідовності:

командою File>Save As записати у файл створювану схему, указавши його ім'я в діалоговому вікні (розширення імені файлу вказувати не обов'язково, програма зробить це автоматично);

командою File>Open завантажити на робоче поле імпортовану схему стандартним для Windows методом (деякі особливості описані наприкінці розділу);

командою Edit>Select All виділити схему, якщо імпортується вся схема, або виділити її потрібну частину;

3 Елементна база

У даному розділі приводяться короткі відомості про моделі компонентів радіоелектронної апаратури (РЕА), що є в програмі EWB. Додаткові відомості по таких компонентах будуть приводитися в інших розділах у міру їхнього використання в конкретних схемах.

3.1 Джерела струму

У загальному випадку джерела струму можуть бути представлені у вигляді генератора напруги або генератора струму. Джерела струму поділяються на джерела постійного струму, змінного струму і керовані (функціональні) джерела. Крім того, вони підрозділяються на вимірювальні джерела і джерела для електроживлення.

Прикладом вимірювального джерела є розглянутий у розділі 3 функціональний генератор. З джерел постійного струму в якості вимірювального широко використовується так званий нормальний елемент (електрохімічне джерело), що володіє високою стабільністю вихідної напруги і використовується у високоточних зразкових установках для перевірки вольтметрів, амперметрів і інших вимірювальних приладів.

Джерела для електроживлення є самими масовими пристроями. Їх прийнято поділяти на первинні і вторинні. До первинних джерел відносяться:

електрогенератори, що перетворюють механічну енергію в електричну, термоелектрогенератори, сонячні й атомні батареї, електрохімічні джерела. У вторинних джерелах струму відбувається перетворення струму первинного джерела.

Джерела постійного струму в програмі EWB представлені на рис. 3.1.

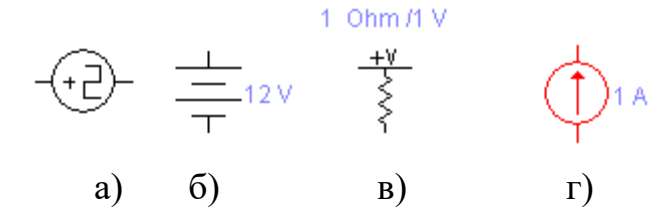


Рисунок 3.1 - Джерела постійного струму

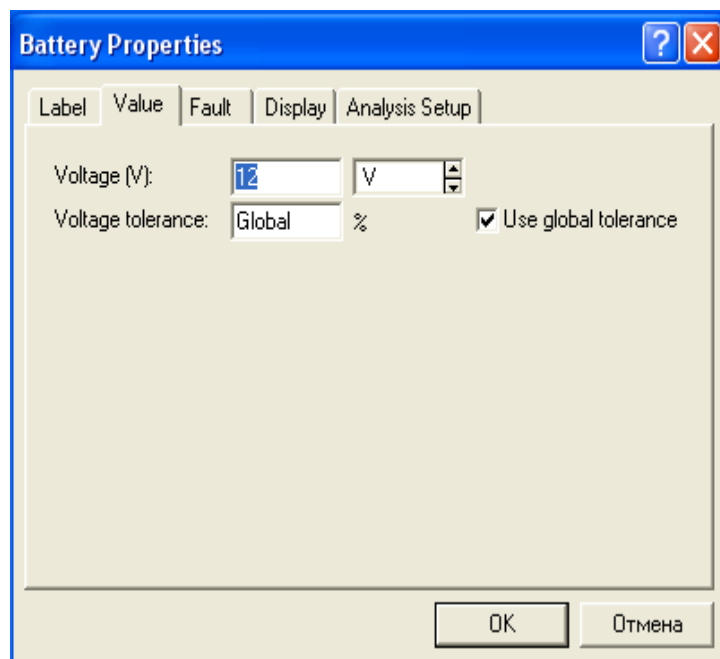


Рисунок 3.2 - Вікно задання ЕРС джерела живлення

Ідеальне (із внутрішнім опором $R_i = 0$) джерело постійної напруги +5 В (рис. 3.1, а) призначене, в основному, для логічних схем. На рис. 3.1. показане ідеальне джерело постійної напруги. ЕРС задається в діалоговому вікні на рис.3.2.

Значення параметрів джерела напруги, які характеризуються ЕРС (Pull-Up Voltage) і внутрішнім опором (Resistance) (рис. 3.1, в), встановлюються за допомогою діалогового вікна (див. рис. 3.3).

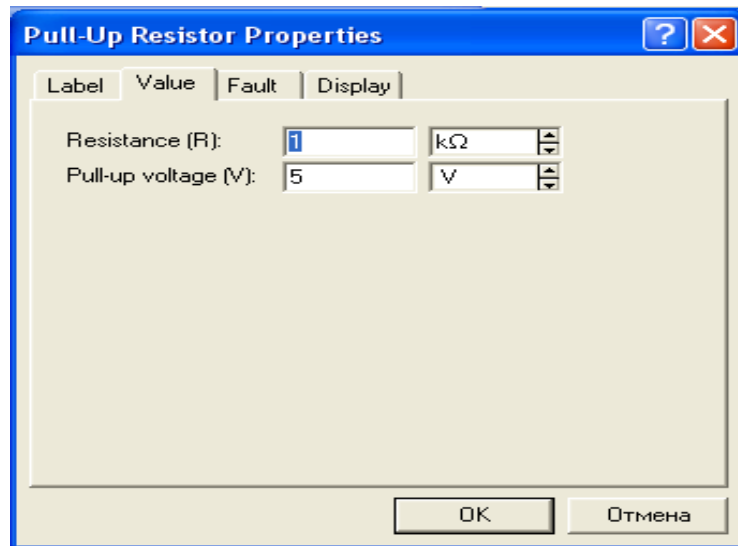


Рисунок 3.3 - Вікно встановлення параметрів джерела живлення

Джерела змінного струму в програмі EWB підрозділяються на джерела не модульованих (рис. 3.4) і модульованих (рис. 3.8) сигналів. Для ідеального генератора змінної напруги (рис. 3.4, а) напруга (Voltage), частота (Frequency) і початкова фаза (Phase) синусоїдального сигналу задаються у вікні на рис. 3.5.

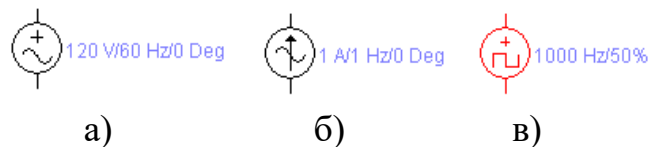


Рисунок 3.4 - Джерела змінного струму

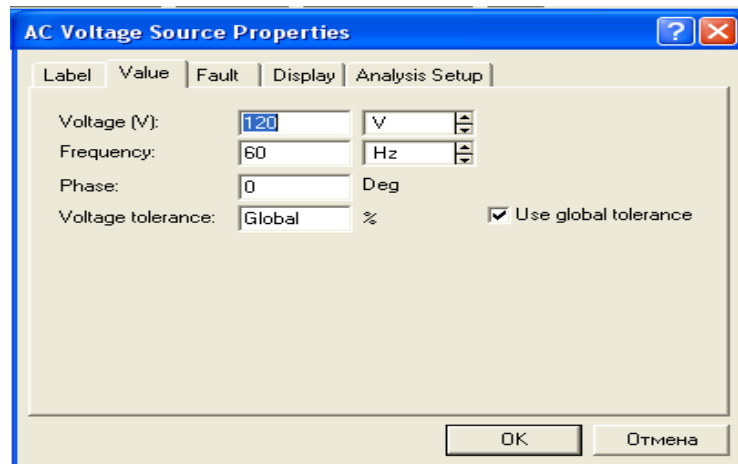


Рисунок 3.5 - Вікно встановлення параметрів джерела синусоїдальної напруги

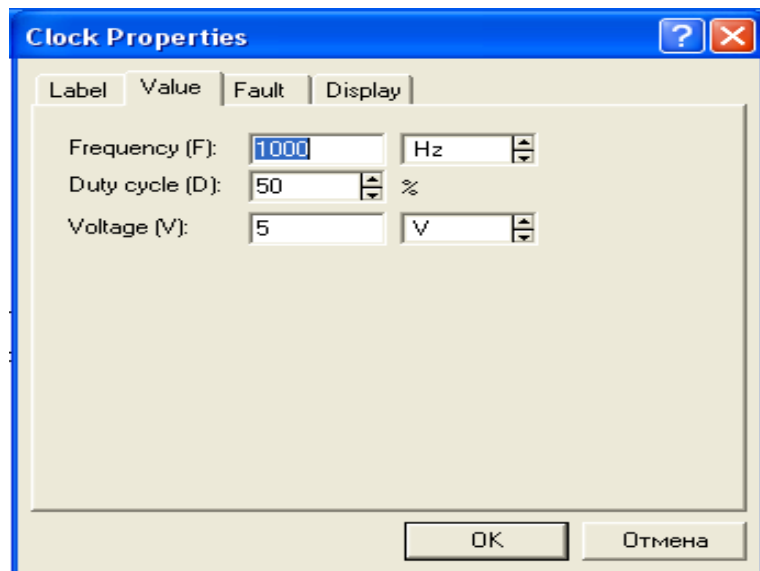


Рисунок 3.6 - Вікно встановлення параметрів джерела імпульсної напруги прямокутної форми

Встановлення струму, частоти і початкової фази ідеального генератора змінного струму (рис. 3.4, б) здійснюється аналогічно джерелу синусоїдальної напруги.

Ідеальний генератор імпульсної напруги (рис. 3.4, в) є джерелом полярних імпульсів із заданою амплітудою, частотою проходження і коефіцієнтом заповнення (Duty Cycle), (вікно на рис. 3.6).

При зазначеному на рис. 3.6 значенні коефіцієнта заповнення 50% (тривалість імпульсу дорівнює половині періоду) періодична імпульсна послідовність називається меандром. Такий сигнал може бути представлений у виді суми гармонійних складових (простих синусоїд) шляхом розкладання в ряд Фур'є:

$$U(x) = U_m/2 + (2U_m/7\pi)[\cos(2\pi) - 0,333 \cos (6 \pi F) + 0, \cos (10 \pi F) - ...]. \quad (3.1)$$

Перший доданок виразу (3.1) — постійна складова, рівна половині амплітуди U_m , перший доданок у квадратних дужках — перша гармоніка, другий — друга гармоніка і т.д. У графічному виді таке розкладання звичайно представляється у вигляді так званого лінійчатого спектра, коли по осі X відкладається частота (номер гармоніки), а по осі Y у виді вертикальної лінії — амплітуда гармоніки. Для одержання такого спектра засобами програми EWB 5.0 (див. роз. 1) необхідно скласти ланцюг із джерела (рис. 3.4, в), резистора, заземлення і застосувати команду Analysis>Fourier. Отриманий при цьому спектральний розподіл гармонік для розглянутої імпульсної послідовності при $U_m = 2$ В показано на рис. 3.7. Для того щоб у чорно-білому зображенні була видна постійна складова, у меню Graph Properties>Left Axis був обраний білий колір для осі X. З рис. 3.7 видно, що постійна складова дійсно дорівнює $U_m/2 = 1$ В, амплітуда першої гармоніки $2U_m/7i = 1,27$ В.

Помітимо, що для імпульсної послідовності при шпаруватості, не рівної 2, вираз (3.1) трохи ускладнюється.

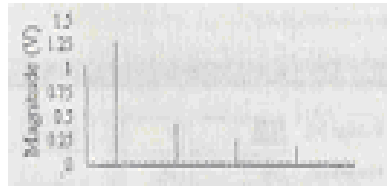


Рисунок 3.7 - Лінійний спектр послідовності прямокутних імпульсів типу меандр



Рисунок.3.8 - Джерела модульованої напруги

Джерела модульованої напруги в програмі EWB представлені компонентами, показаними на рис. 3.8.

Джерело на рис. 3.8, а — ідеальний генератор амплітудно-модульованих коливань (АМ), параметри якого задаються в діалоговому вікні (рис. 3.9), у якому позначено: Carrier Amplitude — амплітуда несучої, Carrier Frequency — частота несучої, Modulation Index — коефіцієнт модуляції, Modulation Frequency — частота коливання, що модулюється.

Осцилограма АМ-сигналу при $M = 0,5$ і значеннях інших параметрів, зазначених у вікні на рис. 3.9, показана на рис. 3.10. Коефіцієнт модуляції визначається як відношення амплітуди що обгинає (на осцилограмі — 0,5 В) до її середнього значення, тобто до амплітуди несучої (1 В). Коефіцієнт модуляції завжди менше або дорівнює одиниці.

Аналітичне вираження для АМ-сигналу записується в наступному вигляді:

$$U(t) = U_c[1 + M\sin(2\pi F_m t)]\sin(2\pi F_c t)$$

Цей вираз після тригонометричних перетворень може бути представлений в більш наочному вигляді:

$$U(t) = U_c[\cos(2\pi F_c t) + 0,5M\cos 2\pi(F_c + F_m)t + 0,5M\cos 2\pi(F_c - F_m)t]. \quad (3.2)$$

Перший доданок виразу (3.2) називається несучим коливанням, другий доданок — коливанням з верхньою бічною, третій — коливанням з нижньою бічною частотою.

Параметри джерела частотно-модульованих коливань (ЧМ) на рис. 3.8, б задаються в діалоговому вікні (рис. 3.11), аналогічному по набору параметрів вікну на рис. 3.9.

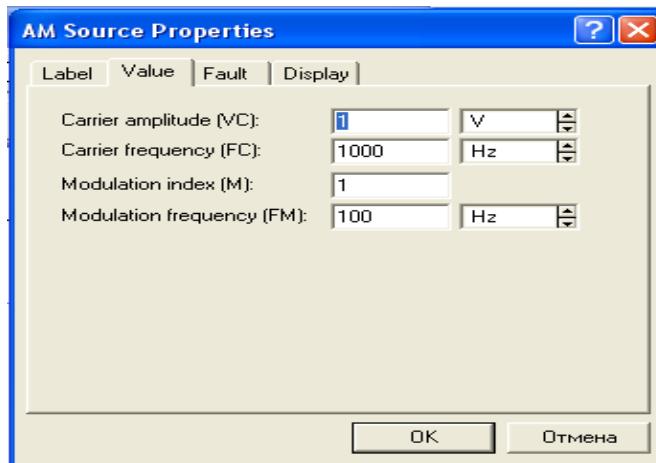


Рисунок 3.9 - Вікно установки параметрів джерела АМ-коливань

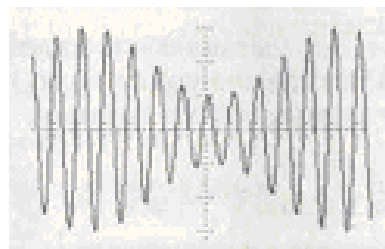


Рисунок 3.10 - Осцилограма АМ-коливань

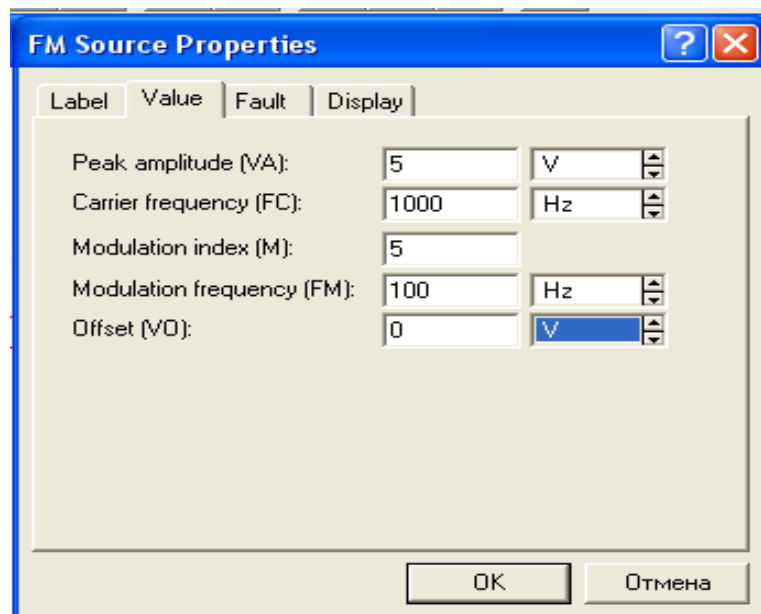


Рисунок 3.11 - Вікно установки параметрів джерела ЧМ-коливань

Аналітичний вираз для ЧМ-сигнала має такий вигляд:

$U(t) = U_c \sin[2\pi f_c t + M \sin(2\pi f_m t)]$. Цей вираз для зручності інтерпретації перетворюється до виду:

$$U(t) = U_c J_0(z) \cos(2\pi f_c t) + U_c \sum_{n=1}^{\infty} J_n(z) \cos 2\pi (f_c + n f_m) t + U_c \sum_{n=1}^{\infty} J_n(z) \cos 2\pi (f_c - n f_m) t$$

$J_0(z), J_n(z)$ — функції Бесселя нульового і n -го порядку від аргументу $z = M$.

У приведеному виразі, як і у випадку АМ-сигналу, перший доданок називається несучим коливанням, другий доданок — гармонійною складовою верхньої бічної смуги частот, третє — складовою нижньої бічної смуги частот. Кількість верхніх і нижніх бічних частот теоретично нескінченна. Практично ж при великих значеннях M складові, починаючи приблизно з $n = M + 1$, можна не враховувати.

Осцилограма ЧМ-коливання, отримана при індексі модуляції $M = 5$, приведена на рис. 3.12.



Рисунок 3.12 - Осцилограма ЧМ-коливань

Керовані джерела програми EWB показані на рис. 3.13. Джерело на рис. 3.13, а являє собою джерело напруги, кероване струмом. У діалоговому вікні цього джерела задається єдиний параметр — коефіцієнт передачі, дорівнює відношенню вихідної напруги до струму керування; параметр має розмірність опору. Для джерела струму, керованого напругою (рис. 3.13, б), цей параметр має розмірність провідності, оскільки коефіцієнт передачі дорівнює відношенню вихідного струму до напруги керування.

Джерела на рис. 3.13 являють собою джерела напруги і струму, керовані відповідно напругою і струмом. Коефіцієнт передачі цих пристроїв — величина безрозмірна.

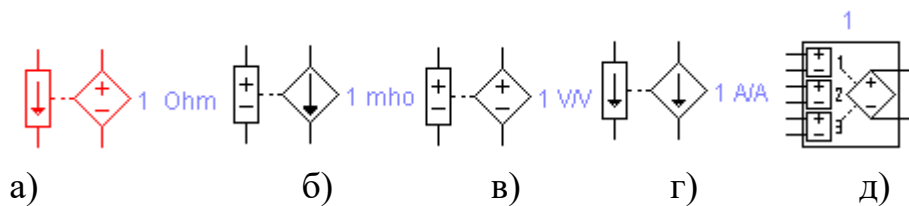
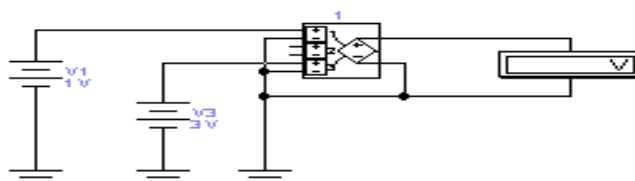


Рисунок 3.13 - Керовані джерела напруги і струму



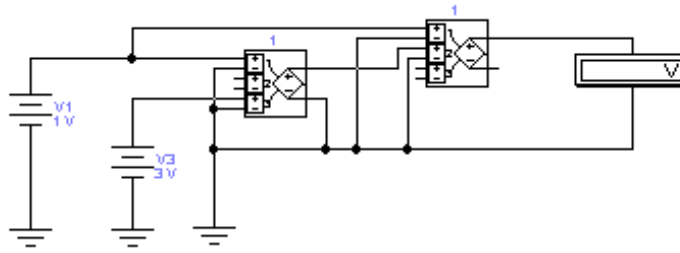


Рисунок 3.14 - Схеми включення поліноміального джерела

Вихідний сигнал керованого джерела на рис. 3.13, д визначається поліноміальною функцією. Для більш детального знайомства з таким джерелом розглянемо деякі схеми їхнього включення (рис. 3.14). Перша (верхня) схема виконує підсумовування напруг $V1$ і $V3$, середня схема — множення однойменних напруг, а нижня підносить до кубічного степеня напругу $V1$.

Вихідна напруга розглянутого джерела описується поліномом наступного виду:

$$Y = A + B \cdot V1 + C \cdot V2 + D \cdot V3 + E(V1)^2 + F \cdot V1 \cdot V2 + G \cdot V1 \cdot V3 + H(V2) + I \cdot V1 \cdot V2 + J(V3)^2 + K \cdot V1 \cdot V2 \cdot V3 \quad (3.3)$$

Коефіцієнти полінома задаються за допомогою вікна (рис.3.15)

Label	Value	Fault	Display
Constant (A):	0		
Coefficient of V1 (B):	0		
Coefficient of V2 (C):	0		
Coefficient of V3 (D):	0		
Coefficient of V1 (E):	1		
Coefficient of V1*V2 (F):	0		
Coefficient of V1*V3 (G):	0		
Coefficient of V2 (H):	0		
Coefficient of V2*V3 (I):	0		
Coefficient of V3 (J):	0		
Coefficient of V1*V2*V3 (K):	0		

Рисунок 3.15 - Значення параметрів поліноміального джерела для піднесення $V1$ до другого степеня

Контрольні питання і завдання

1- Є джерело напруги 10 В з внутрішнім опором 10 МОм і навантаженням, що змінюється в межах від 10 до 100 Ом. Чи можна таке джерело назвати генератором струму?

2- Проведіть розрахунок виразу (3.1) при $U_m = 2$ В і порівняйте отримані результати з даними рис. 3.7.

3. Формула (3.3) отримана шляхом зіставлення даних на рис. 3.14 і значень коефіцієнтів полінома в діалоговому вікні джерела для кожної схеми. Проведіть такий порівняльний аналіз самостійно. Яким іншим, більш простим способом можна одержати формулу (3.3)?

4. Керуючись формулою (3.3), дослідіть можливі варіанти зведення постійної напруги 3 В в кубічний ступінь, у четвертий і п'ятий ступінь на базі одного поліноміального джерела. Складіть схеми пристроїв і перевірте їхню працездатність.

5. Складіть схему для дослідження спектрального складу АМ- і ЧМ-сигналів.

Результати моделювання зіставте з розрахунковими по формулах (3.2) і (3.2a).

6. Як відомо, потужність у ланцюзі постійного струму визначається за формулою: $W = VI = V^2/R$. Для створення моделі найпростішого ватметра, що реалізує цю формулу, використовуйте поліноміальне джерело, а в якості вимірювального приладу — амперметр W, включений послідовно з навантаженням $R = 100 \text{ Ом}$.

7. Ознайомтеся з утворювачами математичних функцій з використанням керованих джерел (схемний файл math.ca4).

3.2 Індикаторні прилади

Індикаторні прилади програми EWB 3.1 показані на рис. 3.16.

Вольтметри й амперметри забезпечують відлік вимірюваної величини з точністю до третього знака (у EWB 5.0 — до четвертого). Параметри приладів задаються в діалоговому вікні (для вольтметра — на рис. 3.17). У поле першого параметра задається вхідний опір вольтметра, у поле другого — режим виміру постійного (DC) або змінного (AC) струму. Діалогове вікно для амперметра — аналогічне розглянутому з тією лише різницею, що перший параметр — це внутрішній опір амперметра, що значно менше вхідного опору вольтметра. Негативна клемма для підключення цих приладів позначена широкою чорною смугою і може бути розміщена на будь-якій грані іконки при обертанні зображення компонента (обертання виконується натисканням комбінації клавіш Ctrl + R).

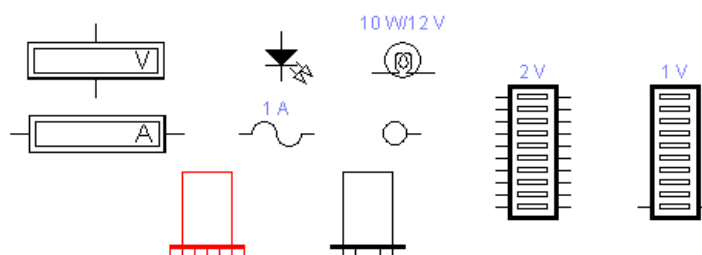


Рисунок 3.16 - Індикаторні прилади

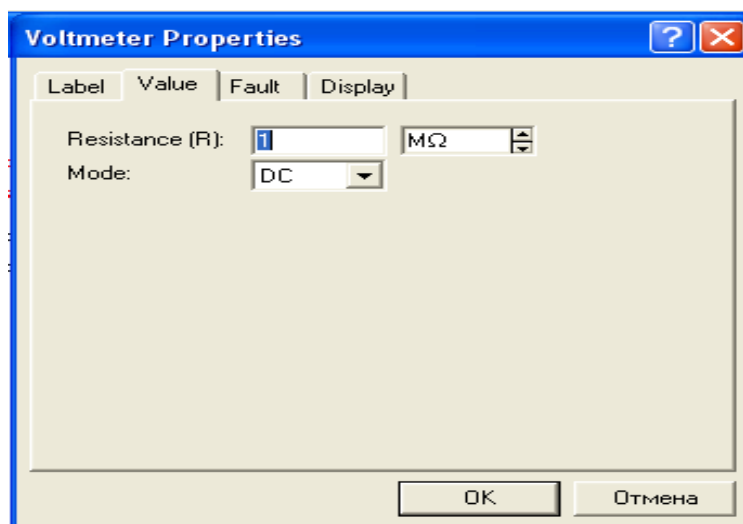


Рисунок 3.17 - Вікно установки параметрів вольтметра

7-сегментний цифровий індикатор — модель широко використовуваних у цифровій техніці алфавітно-цифрових індикаторів. Параметри індикатора задаються в діалоговому вікні (рис. 3.18), у якому перший параметр — максимальне значення вхідної напруги, другий — його мінімальне значення, третій і четвертий параметри — час затримки переключення при переході від низького (мінімального) рівня вхідної напруги до верхнього (максимального) і навпаки, п'ятий параметр — гранична вхідна напруга, при якій починається світіння. Вітчизняними аналогами таких приладів є індикатори типу АЛС, ЗЛС і ін.

Чотирьохвхідний індикатор відрізняється від розглянутого наявного вбудованого дешифратора, що дозволяє підключати його безпосередньо до виходів двійково-десяткових лічильників з кодом 8-4-2-1. Правий вивід такого індикатора - молодший або нульовий розряд, при його активізації на індикаторі висвічується 1. Лівий вивід індикатора — старший або третій розряд, при його активізації відображається цифра 8. Якщо сигнал логічної одиниці подати на усі входи індикатора, то буде відображатися буква F, що позначає в шістнадцятковій системі числення десяткове число 15 (сума чисел 8, 4, 2 і 1). При всіх можливих комбінаціях вхідних сигналів на індикаторі можна відображати числа 0...9 і букви A, B, C, D, E и F. Вітчизняними аналогами таких приладів є індикатори типу 490ИП1, 490ИП2 з тією відмінністю, що вони можуть відображати тільки цифри.

Світлодіод — параметри цього індикатора будуть описані при розгляді напівпровідникових діодів.

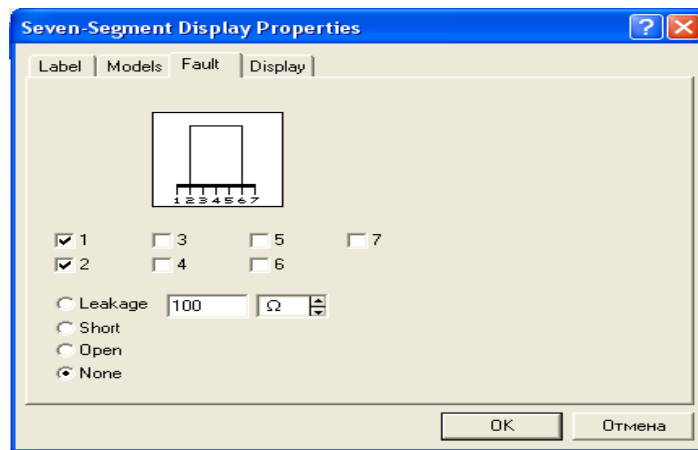


Рисунок 3.18 - Вікно встановлення параметрів 7-сегментного індикатора

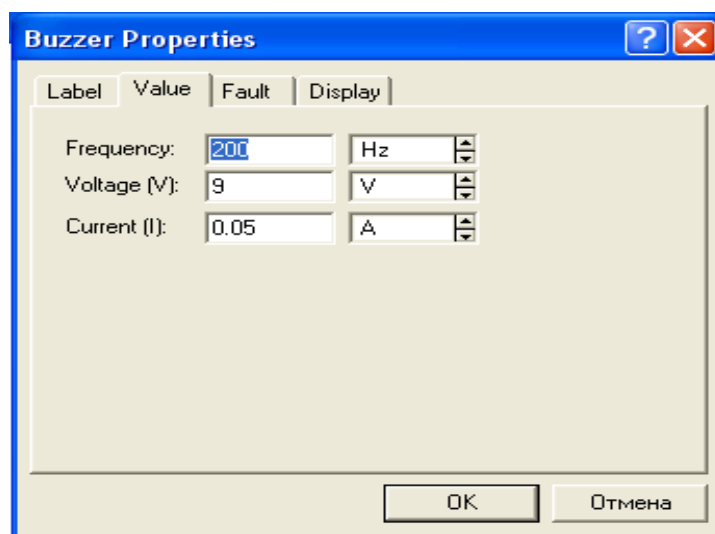


Рисунок 3.19 - Вікно встановлення параметрів звукової сигналізації

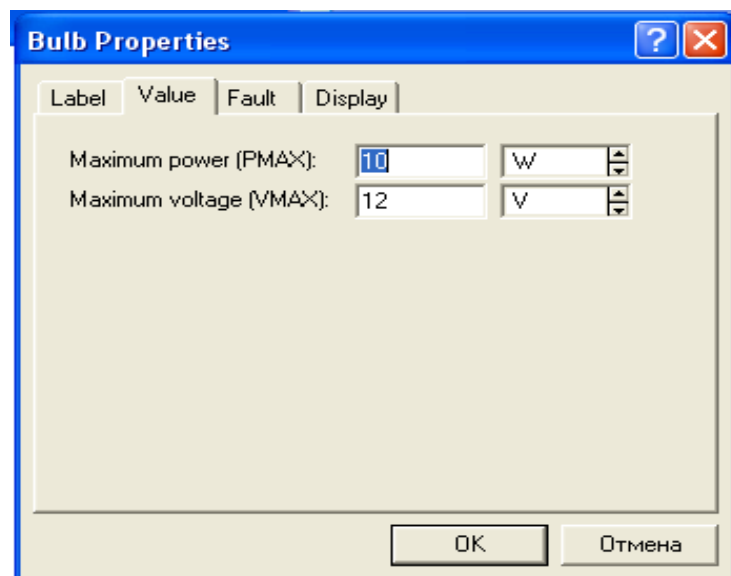


Рисунок 3.20 - Вікно установки параметрів лампи розжарювання

Запобіжник — модель плавкого запобіжника, спрацьовування якого при заданому струмі супроводжується провалом на його значку зигзагоподібної перемички між вихідними затискачами.

Логічний пробник — характеризується напругою спрацьовування 2,4 В, що відповідає мінімальному значенню сигналу логічної одиниці цифрової ТТЛ-серії (з живленням +5 В). Спрацьовування супроводжується червоним або синім світінням.

Звукова сигналізація (зумер) — параметри задаються в діалоговому вікні (рис.3.19), у якому перший параметр — частота сигналу, подаваного на гучномовець комп'ютера, два інших — напруга і струм спрацьовування.

Лампа розжарення характеризується потужністю і номінальною напругою (рис.3.20). Напруга, при якій лампочка загорається, приблизно дорівнює половині номінальної. При напрузі, що перевищує номінальну на невелику величину, лампочка перегоряє і ланцюг обривається, тобто цей компонент може бути використаний також як запобіжник, що спрацьовує при заданих значеннях напруги і струму, рівного відношенню потужності до напруги.

Десятисегментний індикатор містить лінійку з десяти незалежних індикаторів, параметри яких встановлюються в діалоговому вікні (рис. 3.21), де перший параметр — напруга спрацьовування, другий і третій — номінальний і мінімальний струм. Вітчизняними аналогами цього індикатора є шкальні індикатори типу ЗЛС317, ЗЛС343А, ЗЛС362 і ін. .

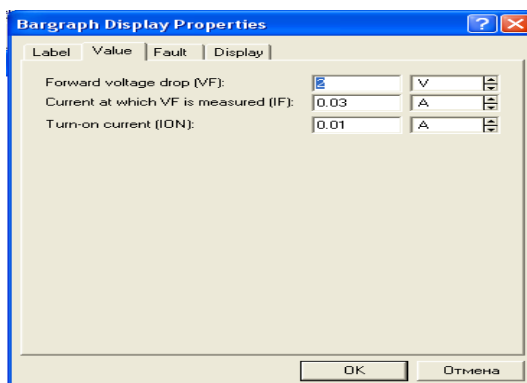


Рисунок 3.21 - Вікно встановлення параметрів 10-сегментного індикатора

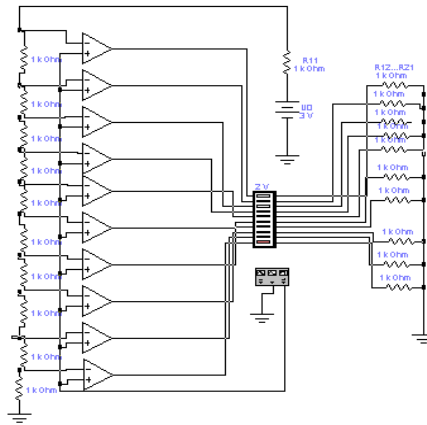


Рисунок 3.22 - Схема включення десятисегментного індикатора

Приклад використання 10-сегментного індикатора приведений на рис. 3.22. Ця схема в трохи зміненому виді запозичена з каталогу готових схем програми EWB. Вона містить 10 компараторів напруги на ОП. На один вхід кожного ОП подається опорна напруга, формована джерелом напруги U_0 і ділником на резисторах $R1...R11$. Другі входи всіх ОП об'єднані і підключені до виходу функціонального генератора, використовуваного в режимі синусоїдальних або трикутних коливань з постійною складовою. Амплітуда вихідного сигналу вибирається рівною $U_0/2$, частота — близько 1 Гц. Виходи ОП підключені до сегментів індикатора, додаткові резистори $R12...R21$ служать для установки струму через кожен сегмент.

Після включення джерела живлення (початок моделювання) вихідна напруга функціонального генератора порівнюється з опорною. Це приводить до послідовного спрацьовування компараторів, що фіксується індикатором і візуально сприймається в такий же спосіб, як і індикація рівня в аудіосистемах (безупинно і синхронно з рівнем голосності світіння, що переміщається по довжині індикатора,).

Принцип роботи розглянутої схеми використаний в іншому десятисегментному індикаторі (рис. 3.16), у якого всього два виводи. Параметри індикатора задаються в діалоговому вікні (рис. 3.23), у якому перший параметр визначає напругу спрацьовування першого (нижнього) сегмента, другий — напругу спрацьовування верхнього (десятого) сегмента. Для визначення напруги спрацьовування інших сегментів можна скористатися простою схемою, що складається з послідовно включених індикатора і джерела постійної напруги. Змінюючи напругу джерела, неважко визначити напругу спрацьовування кожного сегмента візуальним контролем за його світінням.

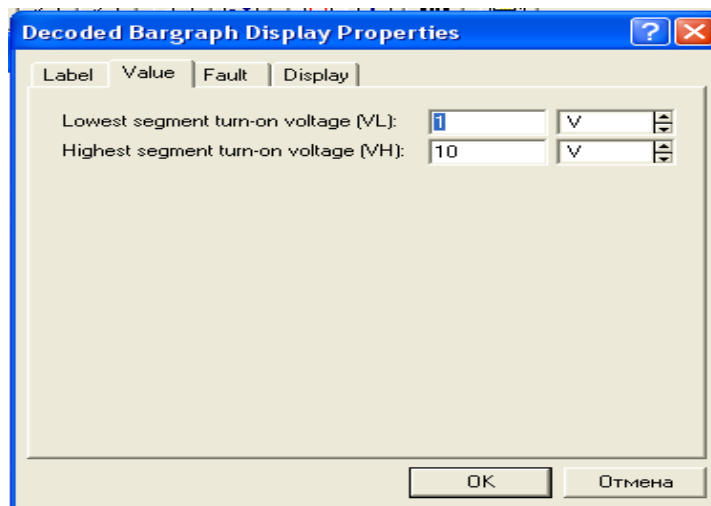


Рисунок.3.23 - Встановлення параметрів 10-сегментного індикатора

Контрольні питання і завдання

1. Визначити напругу спрацьовування кожного сегмента 10-сегментного індикатора при вихідних даних, зазначених на рис. 3.23. Встановіть характер залежності напруги спрацьовування від номера сегмента.
2. Використовуючи схему на рис. 3.24, визначити струм I , при якому лампочка L перегорить при різних значеннях припустимої потужності.

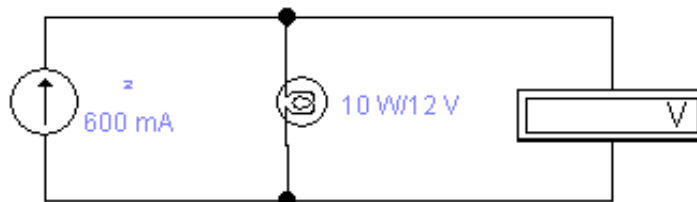


Рисунок 3.23 - Застосування лампи розжарювання в якості запобіжника

- 3- Складіть схему з джерела постійної напруги і логічного пробника. Змінюючи напругу джерела, встановіть напругу спрацьовування пробника з точністю до десятих часток вольт.
- 4- Складіть схему, що складається зі звукового сигналізатора (зумера), джерела постійної напруги, резистора, амперметра і вольтметра. Переконайтеся в правильності спрацьовування звукової сигналізації при встановлених у діалоговому вікні значеннях струму і напруги спрацьовування.
5. Яку кількість амперметрів і вольтметрів можна використовувати в модельованій схемі?
6. Проведіть випробування семисегментного індикатора (схемний файл 7segdemo.ca4).
7. Практична конструкція запобіжників являє собою найчастіше трубчастий скляний або керамічний корпус з металевими ковпачками-клемами на торцях, до яких припаюється протягнутий в порожнині корпуса тонкий дріт, матеріал і діаметр якого визначає струм плавлення. При діаметрі $d = 0,02 \dots 0,2$ мм струм плавлення визначається формулою $I = (d - 0,005)/k$, [A],

у якій коефіцієнт $k = 0,034$ — для міді; $0,05$ — для латуні; $0,07$ — для константану; $0,127$ — для заліза. Для більш товстих провідників струм плавлення $I = m(d^3)^{1/2}$, [A], де $m = 80$ — для міді; $24,6$ — для заліза; $12,8$ — для олова. Оскільки розплавлення супроводжується своєрідним вибухом і розбризкуванням розплавленого металу, то зі збільшенням струму і відповідно діаметра проводу діаметр захисного трубчастого корпуса збільшується для запобігання його руйнування. Розрахуйте струм плавлення мідних дрітів діаметрами 1,3; 2,2; 3,4; 5,7 і 10 мм.

3.3 Комутаційні пристрої

Під комутаційними пристроями (КП) розуміються пристрої, що стрибкоподібно змінюють значення своїх параметрів при визначеному (граничному) значенні керуючого сигналу. У пристроях, призначених для комутації електричних ланцюгів, це реалізується практично миттєвою зміною електричного опору або провідності їхніх виконавчих систем (безпосередньо комутуючих елементів). Комутаційні пристрої програми EWB 3.1 представлені на рис. 3.25.

Пристрій на рис. 3.25, а — перемикач типу однополюсного тумблера, керованого натисканням призначеної клавіші клавіатури (за замовчуванням клавіші Space — пробіл). Ім'я клавіші встановлюється в діалоговому вікні (рис. 3.26).

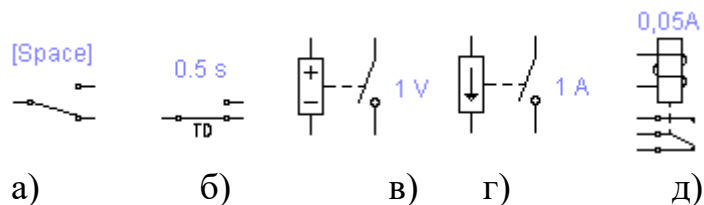


Рисунок 3.25 - Комутаційні пристрої

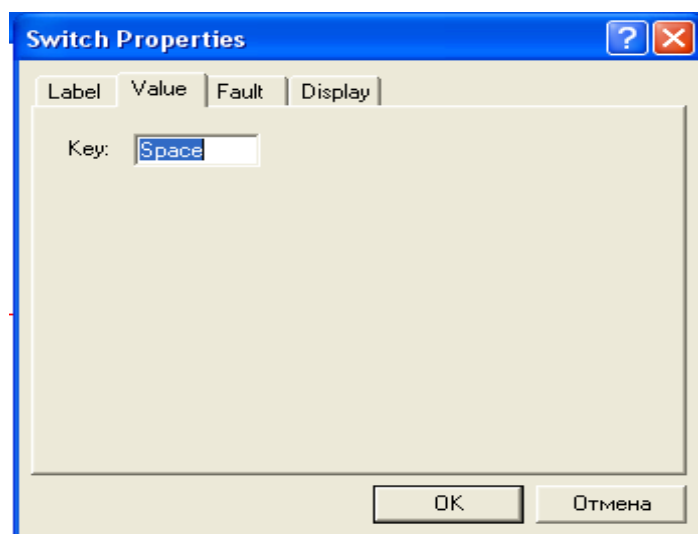


Рисунок 3.26 - Вікно встановлення клавіші управління ключем

КП на рис. 3.25, б — реле часу (перемикач із програмованим часом перемикання). Його параметри задаються в діалоговому вікні (рис. 3.27), де параметр Ton — час включення розімкнутого у вихідному стані контакту після початку моделювання; параметр Toff — час вимикання (переведення контактів у початковий стан), цей час також відраховується від моменту початку моделювання.

Як приклад використання програмувального КП розглянемо схему рис. 3.28. Вона містить джерело живлення $U = 5$ В, два перемикачі S1, S2 і алфавітно-цифровий індикатор. Параметри перемикачів обрані наступні для перший Ton = 3 с, Toff = 10 с; для другий Ton = 6 с, Toff = 15 с. Після включення живлення (початок моделювання) сигнал логічної одиниці (+5 В) буде поданий на виводи 0 і 2 індикатора. Оскільки індикатор працює в коді 8-4-2-1, то при цьому висвічується цифра 5 (сума чисел 4 і 1). Через 3 із ключем S1 переводиться у верхнє положення і сигнал +5 В подається на вхід 3 — займе цифра 9 (сума чисел 8 і 1). Оскільки початок відліку для всіх проміжків часу однаково, те через 3 зі спрацює перемикач S2, у результаті чого сигнал +5 В буде поданий на вхід 1 — займе буква А (шістнадцятковий еквівалент десяткової цифри $10 = 8 + 2$). Потім через 4с спрацює перемикач S1, у результаті чого напруга +5 В буде подана на вхід 2 і займе цифра 6 (сума $4 + 2$). І нарешті через 5 зі спрацює перемикач S2, і схема повернеться у вихідний стан.

КП на рис. 3.25 — однополюсні вимикачі, керовані напругою або струмом. Параметри ланцюга керування задаються в діалоговому вікні на рис. 3.29 (для компонента на рис. 3.25, в), де перший параметр — напруга включення, другий — напруга вимикання (для компонента на рис. 3.25 — струм включення і вимикання відповідно).

Як приклад розглянемо схему на рис. 3.30. У ній є дві керованих напруги ключа S1, S2.

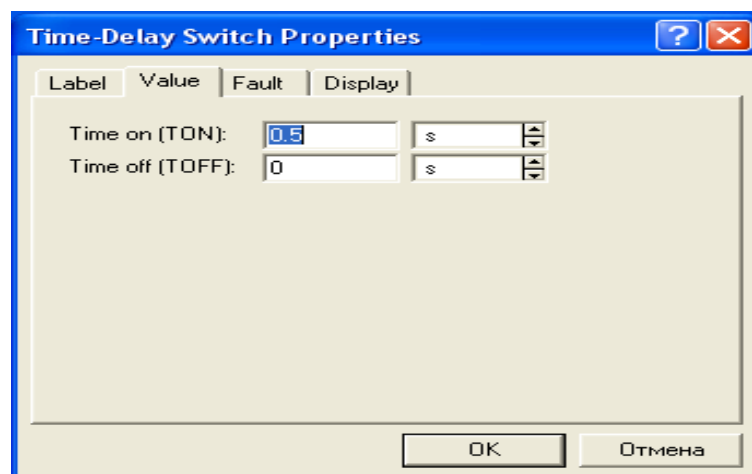


Рисунок 3.27 - Вікно встановлення параметрів реле часу

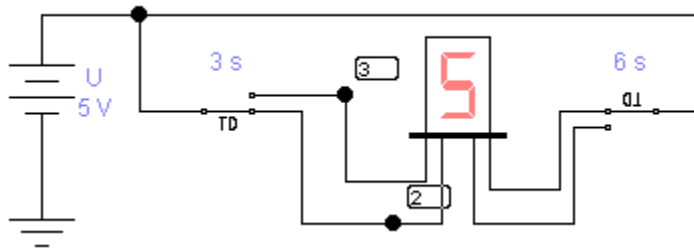


Рисунок 3.28 - Схема з програмуванням КП

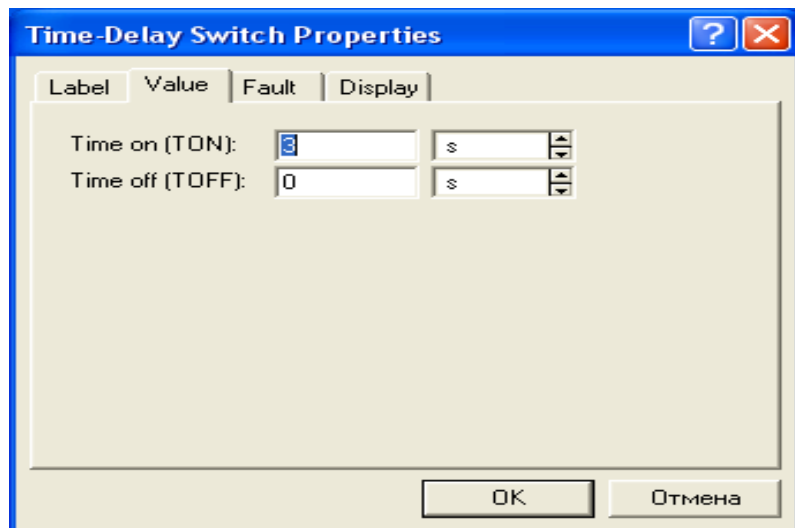
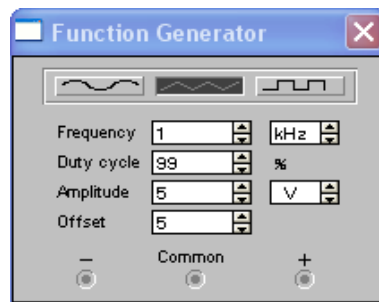
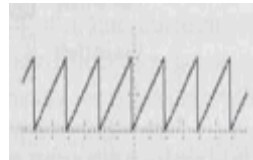


Рисунок 3.29 - Вікно встановлення параметрів КП, який керується напругою

Керування ключами здійснюється від функціонального генератора, вихідна напруга якого контролюється осцилографом. У силовій частині схеми використане джерело напруги U , логічний пробник P і лампочка L . Параметри ланцюга керування ключів обрані в такий спосіб: для першого ключа $U_{on} = 1$ В, $U_{off} = 2$ В; для другого ключа $U_{on} = 5$ В, $U_{off} = 7$ В. Режим роботи функціонального генератора показаний на рис. 3.31, а, осцилограма його вихідної напруги — на рис. 3.31, б. Як видно з осцилограми, генератор дозволяє одержати пилкоподібні однополярні імпульси. З рис. 3.31 видно, що швидкість наростання пилкоподібної напруги складає 10 В/с. Якщо розглянути роботу схеми за один період, то включення логічного пробника відбудеться через $0,1$ з послуги початку формування пилкоподібного імпульсу, оскільки для ключа $S1$ напруга спрацьовування обрана рівною 1 В ("пройдений шлях" у 1 В потрібно розділити на швидкість 10 В/с). Потім при напрузі 2 В, тобто через $0,1$ с, ключ $S1$ розмикається і логічний пробник вимикається. Коли пилкоподібна напруга досягає 5 В ($0,5$ з початку формування імпульсу), спрацьовує ключ $S1$, запалюється лампочка і залишається в такому стані $0,2$ с, поки пилкоподібний імпульс не досягне значення 7 В, при якому ключ $S2$ розмикається. Через $0,3$ с процес повторюється, оскільки пилкоподібний імпульс досягає свого максимального значення 10 В.



а)



б)

Рисунок 3.31 - Режим роботи функціонального генератора (а) і осцилограма його вихідної напруги б)

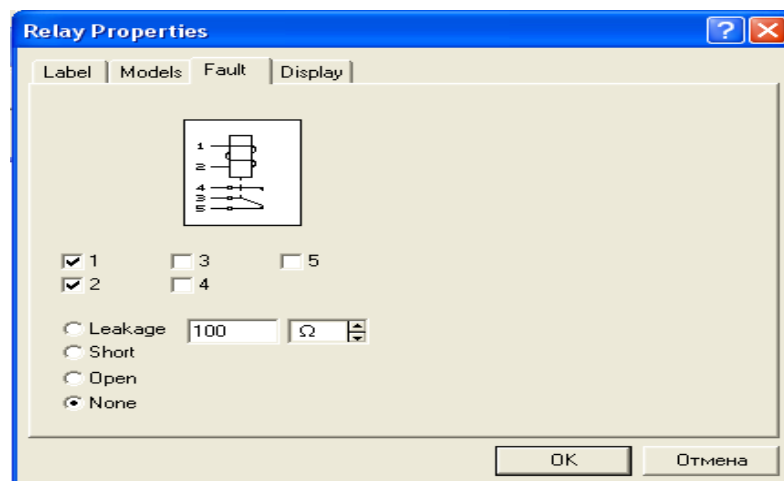


Рисунок 3.32 - Вікно встановлення параметрів електромагнітного реле

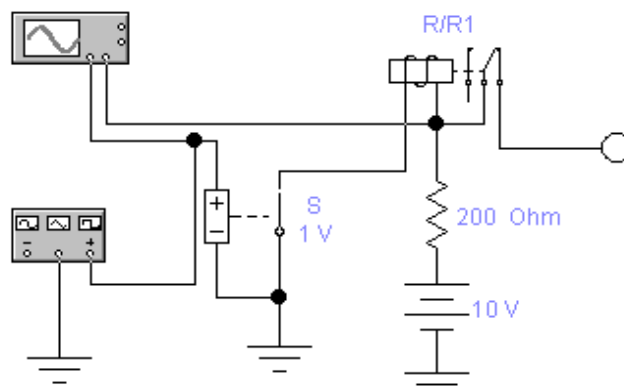


Рисунок 3.33 - Схема включення електромагнітного реле

Пристрій на рис 3.25 — електромагнітне реле з перекидними контактами. Параметри його керуючого ланцюга задаються в діалоговому вікні на рис. 3.3: перший параметр — індуктивність котушки реле, другий і третій — струм спрацювання й утримання.

Як приклад на рис. 3.33 приведена схема включення реле з керуванням від КП (напруга включення 1В, вимикання 8В). Для індикації стану контактів реле використовується логічний пробник Р. Другий канал осцилографа підключений у ланцюг живлення обмотки після струмозадаючого резистора R_d . Осцилограми сигналів (другий канал зміщений вниз) показані на рис. 3.34 при індуктивності обмотки 0,001 і 0,1 Гн. З порівняння осцилограм видно, що при великій індуктивності в ланцюзі керування спостерігаються загасаючі коливання.

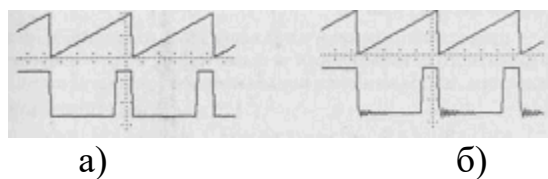


Рисунок 3.33 - Осцилограми напруги на обмотці електромагнітного реле при індуктивності обмотки 0,001Гн (а) і 0,1(б).

3.4 Конденсатори

Конденсатори відносяться до одного з найбільш розповсюджених компонентів РЕА. У програмі EWB 3.1 конденсатори представлені трьома типами (рис. 3.35, а).

Перший тип охоплює практично всі конденсатори, другий — електролітичні, третій — підстроювальні; значення ємності кожного конденсатора може бути встановлене в межах від 10^{'''} пф до 10' Ф. Ємність підстроювального конденсатора може змінюватися натисканням призначеної користувачем клавіші клавіатури (за замовчуванням — клавіші 3), починаючи від максимального значення до мінімального з заданим кроком (від 1 до 100%) (рис. 3.36).

При розрахунку перехідних процесів у програмі використовується схема заміщення конденсатора (рис. 3.35, б), параметри якої визначаються виразами:

$$R_{c_n} = h / 2C; I_{c_n} = 2C * U_n / h + I_n$$

при чисельному інтегруванні по методу трапецій;

$$R_{c_n} = h / C; I_{c_n} = C * U_n / h$$

при використанні методу Гіра.

Тут h — збільшення часу на кожному кроці інтегрування; I_n — значення струму еквівалентного джерела на певному кроці; R_{c_n} , U_n і I_{c_n} — опір шунтуючого резистора, напруга на конденсаторі і струм на певному кроці.

Як приклад розглянемо використовувану на практиці схему ємнісного дільника (рис. 3.37), вихідна напруга якого, вимірювана мультиметром, визначається формулою:

$$U_0 = U_i \cdot C_1 / (C_1 + C_2) \quad (3.4)$$

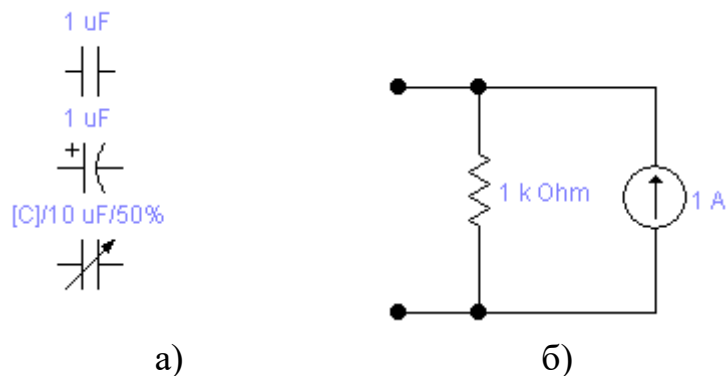


Рисунок 3.35 - Графічне позначення конденсаторів(а) і схема їх заміщення в режимі розрахунку перехідних процесів(б)

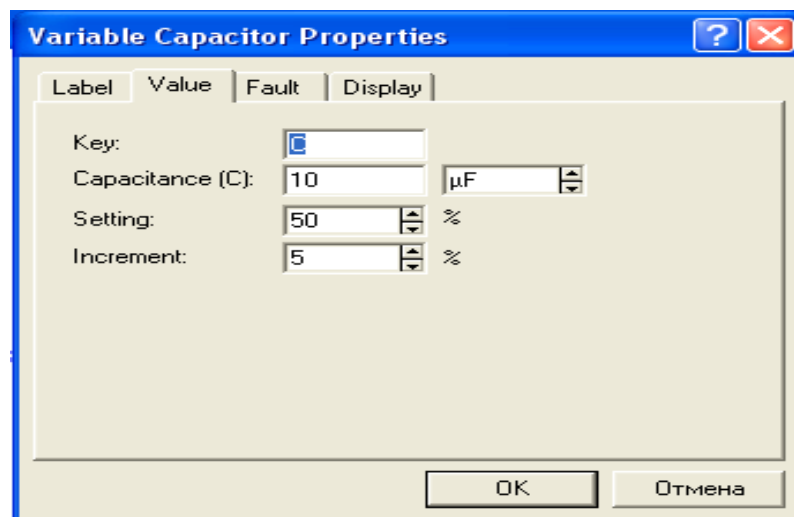


Рисунок 3.36 - Вікно встановлення параметрів підстроювального конденсатора

Оскільки виміри можна проводити при різній формі напруги функціонального генератора, то при зіставленні результатів розрахунку по формулі (3.4) і результатів моделювання необхідно враховувати, що мультиметр вимірює ефективне значення напруги, що для синусоїдального сигналу складає 0,707 від амплітудного, 0,578 — для трикутного і 1 — для меандру (прямокутний сигнал зі шпаруватістю 2). Розглянемо можливість використання в якості підстроювального конденсатора варікапа — спеціально сконструйованого діода, бар'єрна ємність р-п-перехода якого залежить від зворотної напруги відповідно до формули:

$$C_u = C_i / (1 + U_r / U_c)^m \quad (3.5)$$

де C_u — ємність переходу при зворотній напрузі U_c , C_i — ємність при нульовій напрузі, U_t — температурний потенціал (при кімнатній температурі він складає 26 мВ), $m = 0,5$ — для різких (сплавних) і $0,333$ — для плавних (дифузійних) переходів.

Основний параметр варікапа — ємність C_n при номінальній напрузі зсуву. Крім того, указуються максимальна C_{\max} і мінімальна C_{\min} ємності при мінімальній і максимальній напругах зсуву відповідно. Інколи, в числі характеристик варікапа приводиться коефіцієнт перекриття ємності — відношення максимальної ємності до мінімальної.

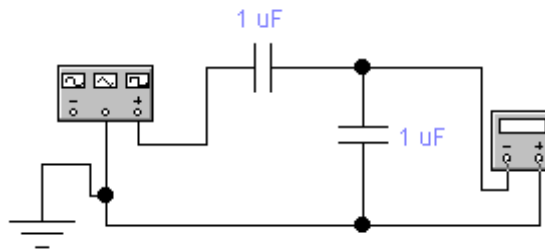


Рисунок 3.37 – Схема включення ємнісного дільника

Якість конденсатора характеризується добротністю, що визначається як відношення реактивного опору до повного опору втрат діода на заданій частоті. Підвищення добротності досягається шляхом зменшення витоків.

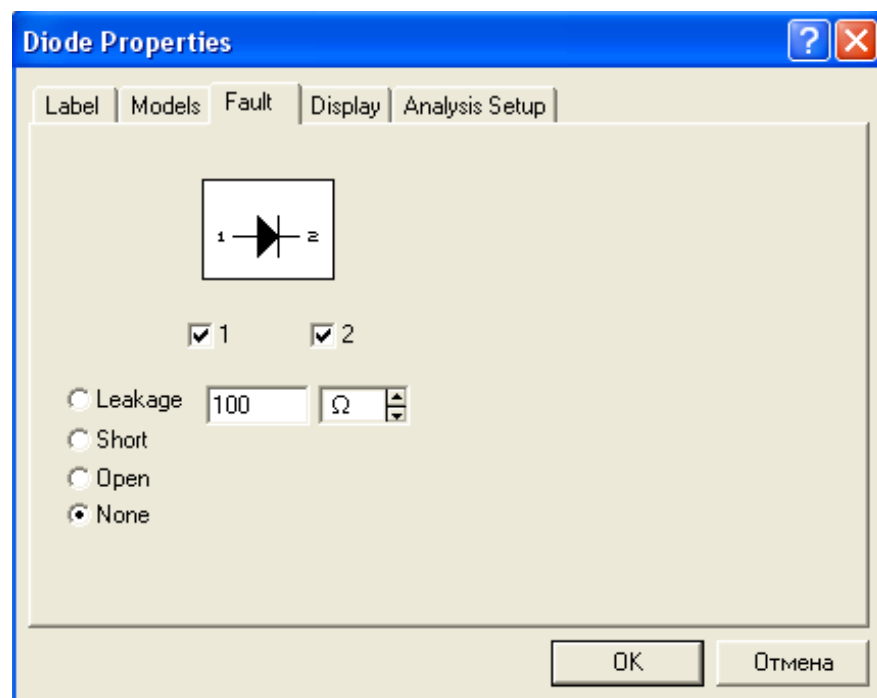


Рисунок 3.38 - Діалогове вікно встановлення параметрів діодів

У програмі EWB немає спеціальної моделі варікапа, замість неї можна використовувати модель діода. У перелік параметрів діода входять наступні (див. рис. 3.38, у квадратних дужках приведені позначення параметрів, прийняті в EWB5.0):

Saturation current I_s [IS], A — зворотний струм діода (за замовчуванням 10^{14} A);

Ohmic Resistance r_s [RS], Ом — об'ємний опір (від десятків до десятих часток Ом);

Zero-bias junction capacitance C_j [CJO], Ф — бар'єрна ємність p-n-переходу при нульовій напрузі (від одиниць до десятків пф);

Junction potential v_j [VJ], V — контактна різниця потенціалів (0,75 V);

Tranzit time τ [TT], c — час переносу заряду;

Junction grading coefficient m [M] — конструктивний параметр p-n-переходу: див. формулу (3.5), у більшості випадків $m = 0,333$;

Revers Bias Breakdown Voltage V_{br} [BV], V — максимальну зворотну напругу (задається зі знаком мінус, для стабілітронів параметр не нормується).

Для стабілітронів у перелік параметрів включаються:

Zener test current I_{zt} [IZT], A — номінальний струм стабілізації (від одиниць до десятків mA);

Zener test Voltage at I_{zt} U_{zt} [VZT], V — напруга стабілізації при номінальному струмі стабілізації.

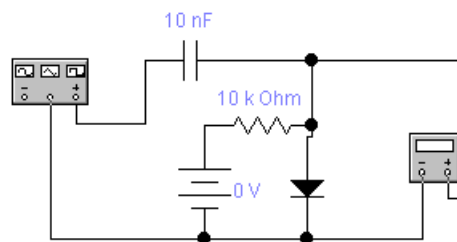


Рисунок 3.39 - Ємнісний дільник з діодом

Схема ємнісного дільника з використанням діода (рис. 3.39) містить; ланцюг зсуву (ланцюг керування бар'єрною ємністю), що складається з джерела напруги U_c і резистора R , генератор (амплітуда 1 В, частота 1 МГц), мультиметр, еталонний конденсатор C і досліджуваний діод VD типу kl (перейменована модель *Ideal* для можливості редагування параметрів) з бар'єрною ємністю $C_i = 100$ пФ при нульовій напрузі на переході. Конденсатори C і C_i утворюють ємнісний дільник, вихідна напруга якого визначається виразом (3.4).

$$C_i = C_0(U_i/U_0 - 1).$$

За допомогою цього виразу можна визначити ємність.

3.5 Резистори

Резистори є самими масовими виробами електронної техніки. У програмі EWB 3.1 резистори представлені трьома типами — постійним, підстроювальним і набором з восьми резисторів (рис. 3.40).

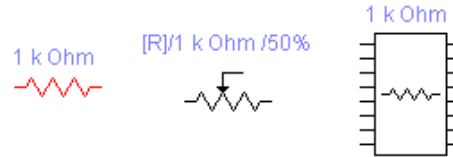


Рисунок 3.40 - Графічні позначення резисторів

Зміна опору підстроювального резистора здійснюється по тому ж принципу, що і для підстроювального конденсатора (див. рис. 3.41). У наборі резисторів опір встановлюється однаковим для усіх восьми резисторів 3.6.

Індуктивні елементи

До індуктивних елементів відносяться котушка постійної індуктивності, котушка індуктивності налаштовувана і трансформатор (див. рис. 3.44, а).

При розрахунку перехідних процесів у програмі використовується схема заміщення котушки індуктивності (рис. 3.44, б), параметри якої визначаються виразами:

$$R_n = 2L/h; I_n = hU_n/2L + I_n$$

при чисельному інтегруванні по методу трапецій;

$$R_n = L/h; I_n = hU_n/L$$

при використанні методу Гіра.

У приведених формулах h — крок збільшення часу; I_n - струм еквівалентного джерела на певному кроці; R_n , U_n і I_n — опір шунтуючого резистора, напруга на індуктивності і струм на певному кроці.

Математична модель трансформатора (рис. 3.44, в) містить керовані джерела струму і напруги, за допомогою яких встановлюється коефіцієнт трансформації, а також елементи, параметри яких задаються в діалоговому вікні (див. рис. 3.45). Відповідно до інструкції користувача виводи 2 і 5 при використанні трансформатора повинні бути заземлені, що в деяких випадках істотно знижує можливості його застосування.

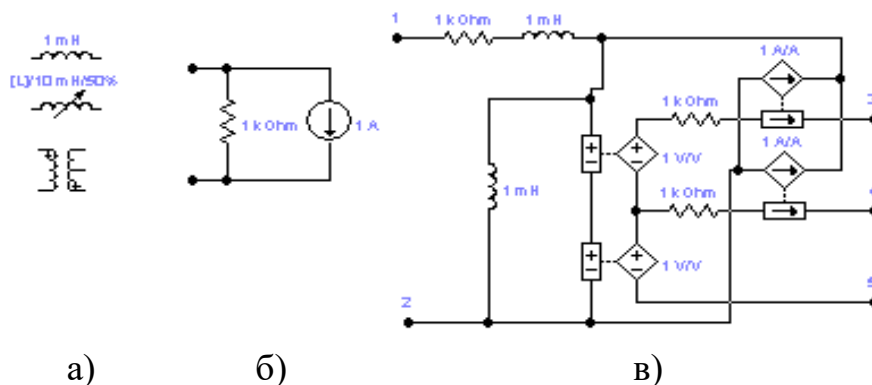


Рисунок 3.43 - Індуктивні компоненти EWB(а), схеми заміщення індуктивності(б) і трансформатора(в)

Параметри котушок з постійною індуктивністю та налаштовуваною задаються за допомогою діалогових вікон, аналогічних вікнам для конденсаторів і резисторів. У діалоговому вікні установки параметрів лінійних трансформаторів (їх ще називають повітряними) задаються (див. рис. 3.45): коефіцієнт трансформації n , індуктивність розсіювання L_e , індуктивність первинної обмотки L_m , опір первинної R_p і вторинної R_s обмоток. При $n > 1$ трансформатор є понижуючим, при $n < 1$ - підвищуючим.

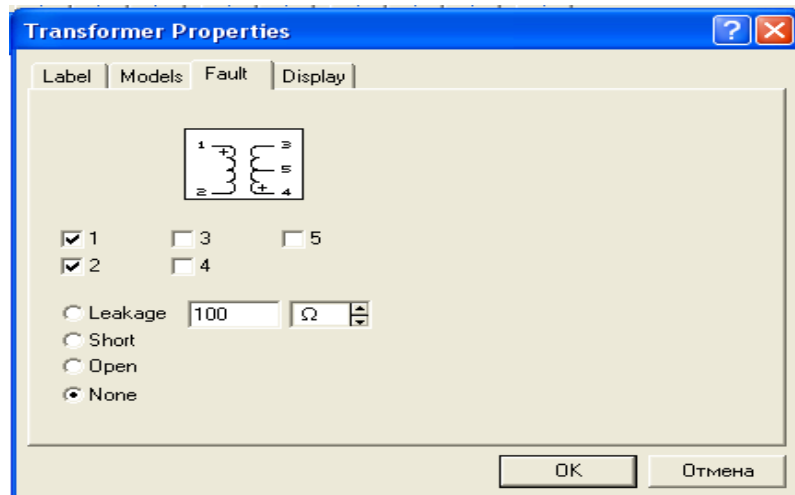


Рисунок 3.45 - Вікно установки параметрів трансформаторів

3.6 Напівпровідникові діоди

Комбінація двох напівпровідникових шарів з різним типом провідності (p — діркової і n — електронної) має випрямляючі властивості: вона набагато краще пропускає струм в одному напрямку, чим в іншому. Полярність напруги, що відповідає великим струмам, називається прямою, а меншим — зворотною. Звичайно користуються термінами пряма і зворотна напруга, прямий і зворотний струм. Поверхня, по якій контактують p - і n -шари, називається металургійною границею, а прилягаюча до неї область об'ємних зарядів — електронно-дірковим переходом.

Електронно-діркові переходи класифікують по різкості металургійної границі і співвідношенню питомих опорів шарів.

Східчастими переходами (коефіцієнт плавності переходу $m = 0,5$, у EWB 5.0 має позначення M) називають переходи з ідеальною границею, по одну сторону якої знаходяться дірки, а по іншу — електрони. Такі переходи найбільш прості для аналізу, тому всі реальні переходи намагаються, якщо це можливо, розглядати як східчасті.

Плавними переходами ($m = 0,333$) називають такі, у яких в області металургійної границі концентрація одного типу домішки поступово зменшується, а іншого типу — росте. Сама металургійна границя в цьому випадку відповідає рівності концентрацій домішок. Усі реальні p - n -переходи

— плавні, ступінь їхнього наближення до східчастого залежить від градієнта ефективної концентрації в районі металургійної границі.

По співвідношенню концентрацій домішок у р- і n-шарах переходи поділяються на симетричні, несиметричні і однобічні. Симетричні переходи нетипові для напівпровідникової техніки. Основне поширення мають несиметричні переходи, у яких концентрації не однакові. У випадку різкої асиметрії, коли концентрації домішок (а виходить, і основних носіїв) розрізняються на один-два порядки і більш, переходи називають однобічними.

Вольт-амперна характеристика р-n-переходу описується виразом:

$$I=I_0(\exp(U/U_t-1)) \quad (3.7)$$

де I — струм через перехід при напрузі U , I_0 — зворотний струм, U_t — температурний потенціал, рівний при кімнатній температурі 26 мВ.

Якщо до переходу підключити зворотну напругу, то при визначеному її значенні перехід пробивається. Розрізняють три види пробоя: тунельний, лавинний і тепловий. Перші два зв'язані зі збільшенням напруженості електричного поля в переході, а третій — зі збільшенням потужності, що розсіюється, і, відповідно, температури.

В основі тунельного пробоя лежить тунельний ефект, тобто "просочування" електронів крізь тонкий потенційний бар'єр переходу. В основі лавинного пробоя лежить "розмноження" носіїв у сильному електричному полі, що діє в області переходу. Електрон і дірка, прискорені полем на довжині вільного пробігу, можуть розірвати один з ковалентних зв'язків напівпровідника. У результаті народжується нова пара електрон-дірка і процес повторюється вже за участю нових носіїв. При досить великій напруженості поля, коли вихідна пара носіїв у середньому породжує більш однієї нової пари, іонізація здобуває лавинний характер, подібно самотійному розрядові в газі. При цьому струм буде обмежуватися тільки зовнішнім опором. Явище пробоя знаходить практичне застосування в стабілітронах — приладах, призначених для стабілізації напруги.

В основі теплового пробоя лежить саморозігрів переходу при протіканні зворотного струму. З ростом температури зворотні струми різко зростають, відповідно збільшується потужність, що розсіюється в переході; це викликає додатковий ріст температури і т.д. Як правило, тепловий пробій не має самотійного значення: він може початися лише тоді, коли зворотний струм уже придбав досить велику величину в результаті лавинного або тунельного пробоя.

Раніше (у розд. 3.4) ми вже говорили про бар'єрну ємність. Її прийнято розділяти на дві складові: бар'єрну ємність, що відображає перерозподіл зарядів у переході, і дифузійну ємність, що відображає перерозподіл носіїв у базі. Такий поділ у загальному умовний, але він зручний на практиці, оскільки співвідношення обох ємностей різне при зміні полярності прикладеної напруги. При прямій напрузі головну роль грають надлишкові заряди в базі і, відповідно, дифузійна ємність. При зворотній напрузі надлишкові заряди в

базі малі і головну роль грає бар'єрна ємність. Обидві ємності нелінійні: дифузійна ємність залежить від прямого струму, а бар'єрна — від зворотної напруги.

Набір параметрів, що задаються, для діодів у EWB 5.0 помітно більше в порівнянні з EWB 3.1. Діалогове вікно для задання параметрів діодів у EWB 5.0 складається з двох однакових на зовнішній вигляд закладок (перша з них показана на рис. 3.49), за допомогою яких можна додатково (у порівнянні з вікном на рис. 3.38) задати наступні параметри:

N- коефіцієнт інжекції;

EG - ширина забороненої зони, еВ; (для германія — 0,72 еВ, для кремнію — 1,1 еВ); FC - коефіцієнт нелінійності бар'єрної ємності прямо зміщеного переходу; BV - напруга пробою (позитивна величина, у EWB 3.1 вона прийнята негативною), В; для стабілітронів замість цього параметра використовується параметр VZT — напруга стабілізації;

IBV — початковий струм пробою при напрузі BV (позитивна величина), А; Для стабілітронів замість цього параметра використовується параметр IZT — початковий струм стабілізації;

XTI — температурний коефіцієнт струму насичення; KF — коефіцієнт фліккер-шуму;

AF — показник ступеня у формулі для фліккер-шуму; TNOM — температура діода, °С.

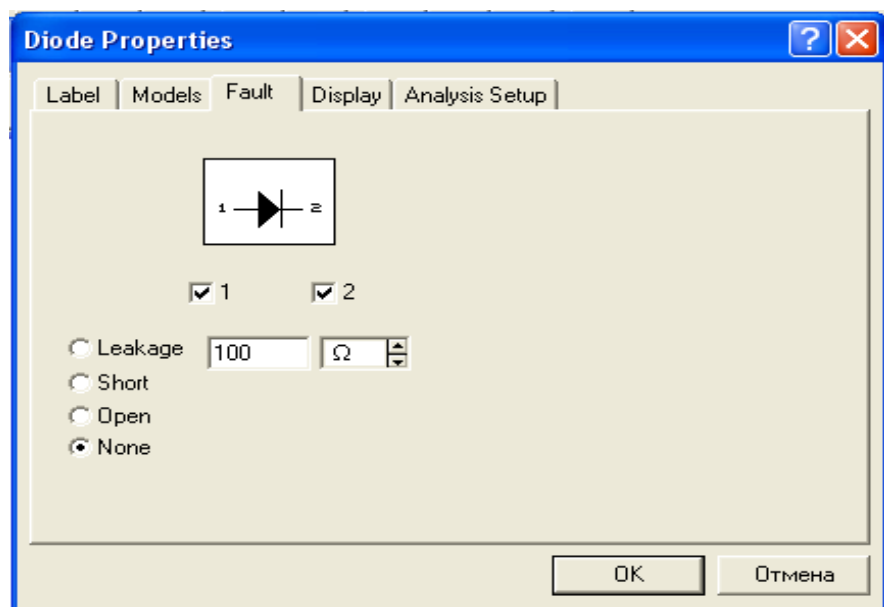


Рисунок 3.49 - Діалогове вікно параметрів діодів

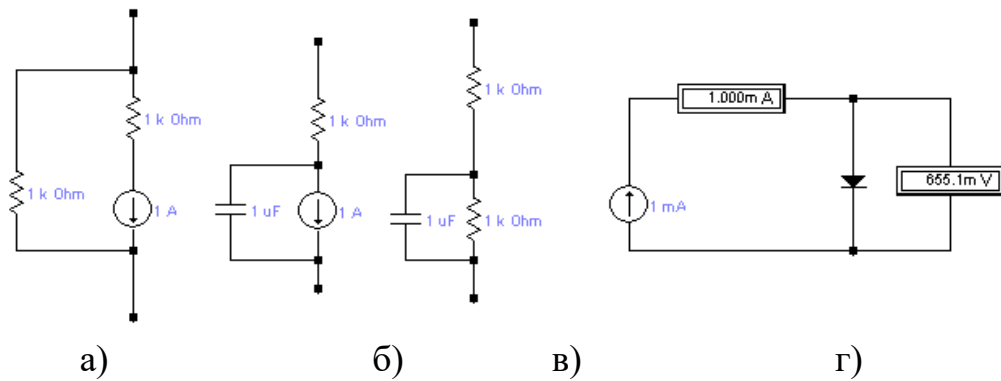


Рисунок 3.50 - Еквівалентні схеми діода при розрахунку на постійному струмі(а), в діапазоні частот (б, в) і схема для дослідження прямої гілки ВАХ(г)

Еквівалентні схеми діода показані на рис. 3.50, а, б, на яких позначено: А — анод, ДО — катод, І — джерело струму, R_s — об'ємний опір, C — ємність переходу, G_{min} — провідність, обумовлена витоками (у EWB 5.0 задається в діалоговому вікні, див. рис. 1.17). Вольт-амперна характеристика <ВАХ> діода визначається наступними виразами:

для прямої гілки

$$I = I_0(\exp(U/(N \cdot U_t)) - 1) + U \cdot G_{min} \text{ для } U \geq -5N \cdot U_t;$$

для зворотної гілки

$$I = I_0(\exp(U/(N \cdot U_t)) - 1) + U \cdot G_{min} \text{ для } 0 \geq U \geq -5N \cdot U_t;$$

$$I = -I_0 + U \cdot G_{min} \text{ для } -BV < U < -5N \cdot U_t;$$

$$I = -IBV \text{ для } U = -BV;$$

$$I = I_0 \{ \exp(-(U + BV)/(N \cdot U_t)) - 1 \} + BV/U_t \text{ для } U < -BV \}.$$

Тут $I_0 = I_s$ — зворотний струм діода при температурі T_{nom} ; N — коефіцієнт інжекції; BV , IBV — напруга і струм пробою; U_t — температурний потенціал переходу; U — напруга на діоді.

При розрахунку перехідних процесів використовується еквівалентна схема діода (див. рис. 3.50, б), для якої ємність переходу визначається за допомогою виразів:

$$C = t(di/dU) + C_{jo}(1 - U/U_t)^{-m} \text{ для } U < FC \cdot V_j;$$

$$C = t(di/dU) + C_{jo}(F^3 - mU/U_t)/F^2 \text{ для } U \geq FC \cdot V_j;$$

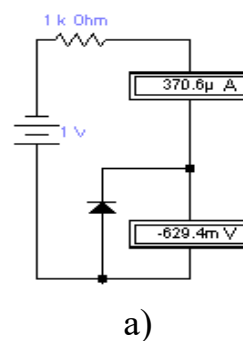
У приведених формулах t — час переносу заряду; C_{jo} — бар'єрна ємність при нульовому зсуві на переході; V_j — контактна різниця потенціалів; $m = 0,33 \dots 0,5$ — параметр переходу.

При малих рівнях сигналів використовується лінеалізована еквівалентна схема (рис. 3.50, в), на якій провідність $G = dl/d = I_0 \exp(U/(NU_t))/(NU_t)$. При цьому ємність переходу визначається формулами:

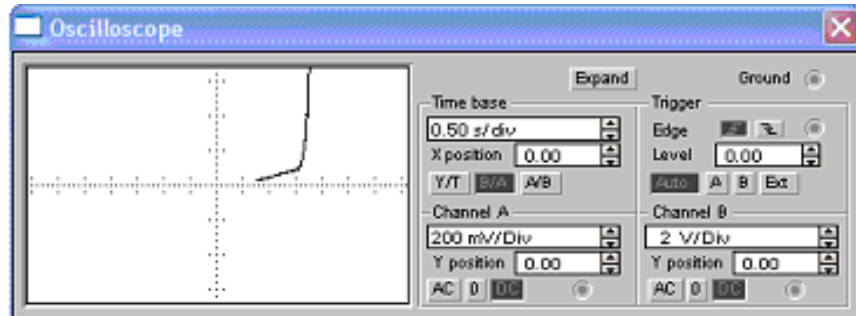
$$C = tG + C_{JO}(1 - U/U_t)^{-m} \text{ для } U < FC * V_J;$$

$$C = tG + C_{JO}(F^3 - mU/U_t)/F^2 \text{ для } U \geq FC * V_J;$$

Дослідження прямої гілки ВАХ діодів може бути проведене за допомогою схеми на рис. 3.50, г. Вона складається з джерела струму I , амперметра A (можна обійтися і без нього), досліджуваного діода VD і вольтметра V для вимірювання напруги на діоді.



а)



б)

Рисунок 3.51 - Схема для дослідження зворотного зв'язку діода а) і ВАХ діода (б)

Процес дослідження ВАХ може бути автоматизований за допомогою характеристик осцилографа (рис. 3.51), у якому формування зображення ВАХ виконується в режимі розгортання В/А осцилографа, при цьому використовується сигнал з функціонального генератора і з навантаження діода.

Крім одиночних діодів, у бібліотеці EWB є також діодний місток, і для якого можна додатково задати коефіцієнт емісії N (Emission I Coefficient). Світлодіод — спеціально сконструйований діод, у якому передбачена можливість виведення світлового випромінювання з області переходу крізь прозоре вікно в корпусі.

При проходженні через діод струму в прилягаючих до переходу областях напівпровідника відбувається інтенсивна рекомбінація носіїв зарядів — електронів і дірок. Частина вивільнюваної енергії виділяється у виді квантів світла. У залежності від ширини забороненої зони напівпровідника випромінювання може мати довжину хвилі або в області видимого світла, або невидимого інфрачервоного випромінювання. Випромінювання переходів на основі арсеніду галію має довжину хвилі близько 0,8 мкм. Переходи з карбіду кремнію або фосфіду галію випромінюють видиме світло в діапазоні від червоного до блакитного кольору. Найважливішими параметрами світлодіода є яскравість, вимірювана в нітах при визначеному значенні прямого струму, і колір світіння (або спектральний склад випромінювання). Для світлодіода додатково вказується мінімальний струм у прямому напрямку Turn-on current (I_{on}), при перевищенні якого світлодіод запалюється. Для виміру ВАХ світлодіодів можна використовувати приведені вище схеми. Перемикаючі діоди з p-n-p-n- або n-p-n-p-структурами — це тиристори. Тиристори, що мають виводи від крайніх електродів, називають диністорами, а прилади з третім виходом (від одного із середніх електродів) — триністорами. Крім того, до класу тиристорів відносяться симістори — симетричні диністори (діаки), симетричні триністори (тріаки) і досить рідкий тип диністора — діод Шоклі, у якому структура n-p-n організована за рахунок наявності пасток, формованих шляхом легування. На рис. 3.52 приведені позначення перемикаючих діодів, моделі яких є в програмі EWB 3.1: (лівий правий) діод Шоклі, симетричний диністор (діак, двонаправлений диністор), триністор (тріодний тиристор) і симетричний триністор (симістор).

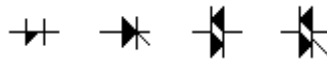


Рисунок 3.52 - Діоди, які самі перемикаються

Для перемикальних діодів задаються значення наступних параметрів (для EWB 5.0 їхні позначення вказуються в квадратних дужках):

Saturation current I_s [IS], A — зворотний струм диністора;

Peak Off-state Current I_{drm} [IDRM], A — те ж, але для триністора;

Switching Voltage V_s [VS], V — напруга, при якій диністор переключається у відкритий стан;

Forward Breakover Voltage V_{drm} [VDRM], V — те ж, але для триністора при нульовій напрузі на керуючому електроді;

Peak On-State Voltage V_{tm} [VTM], V — спадання напруги у відкритому стані;

Forward Current at which V_{tm} is measured I_{tm} [ITM], A — струм у відкритому стані;

Turn-off time T_g [TG], c — час переключення в закритий стан;

Holding current I_h [IH], A — мінімальний струм у відкритому стані (якщо він менший встановленого, то прилад переходить у закритий стан);

Critical rate of f-state Voltage rise dv/dt [DV/DT], В/мкс — припустима швидкість зміни напруги на аноді триністора, при якому він продовжує залишатися в закритому стані (при більшій швидкості триністор відкривається);

Zero-bias junction capacitance C_j [C_{JO}], Ф — бар'єрна ємність диністора при нульовій напрузі на переході;

Gate Trigger Voltage V_{gt} [VGT], В — напруга на керуючому електроді відкритого триністора;

Gate Trigger current I_{gt} [IGT], А — струм керуючого електрода;

Voltage at which I_{gt} is measured V_d [VD], В — напруга відмикання, на керуючому електроді.

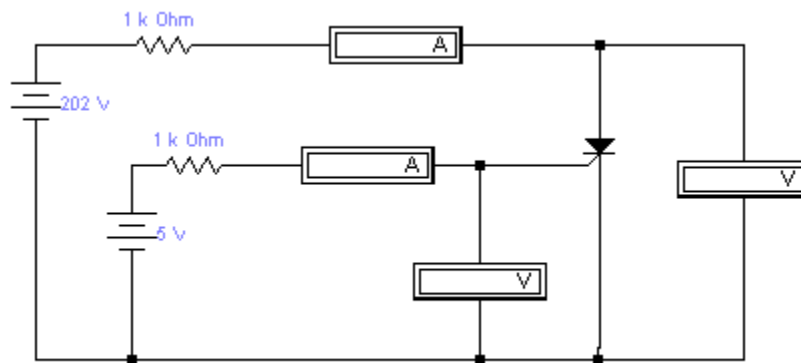


Рисунок 3.52 - Схема для дослідження триністорів

3.7 Цифрові мікросхеми

Напівпровідникова електроніка бере свій початок у 1948 р., коли групою розробників фірми Bell був створений перший транзистор. Через 11 років інженерами фірми Texas Instruments була розроблена перша мікросхема, що складалася усього із шести транзисторів, а в 1971 р. нині всесвітньо відома фірма Intel розробила перший 4-розрядний мікропроцесор 4004, що містив більш 2000 транзисторів. Надалі мікромініатюризація електронних компонентів досягла таких темпів, що це послужило приводом для досить образного порівняння в журналі Scientific American (1982 р.): "Якби авіапромисловість в останні 25 років розвивалася настільки ж стрімко, як і промисловість засобів обчислювальної техніки, то зараз літак "Боїнг-767" коштував би 500 доларів і робив обліт земної кулі за 20 хвилин, затрачаючи при цьому 5 галонів палива". Разючі результати, досягнуті в мікроелектроніці, стали можливі завдяки не тільки новітнім напівпровідниковим технологіям, але і величезному багажеві схемотехнічних рішень, накопиченому протягом десятиліть багатомільйонною армією розробників. Незважаючи на вражаючу уяву кількість транзисторів, зібраних на малюсіньких напівпровідникових кристалах, варто все-таки пам'ятати, що вони являють собою набори з найпростіших елементів, до розгляду яких ми і перейдемо.

У залежності від технології виготовлення інтегральні мікросхеми (ІМС) підрозділяються на серії (сімейства), що розрізняються фізичними параметрами базових елементів і їхнім функціональним призначенням. Найбільше поширення одержали ІМС, виготовлені по ТТЛ- і КМДН-технологіям. (ТТЛ — транзисторно-транзисторна логіка з використанням біполярних транзисторів, КМДН — з використанням комплементарних МДН-транзисторів).

Першою була випущена ТТЛ-серія SN74/SN54 (74 — комерційна, 54 — для військових застосувань). Вітчизняним аналогом серії SN74 стала популярна у свій час серія 155. У 1967 р. додатково розроблені сімейства SN74H/54H (High speed — швидкодіючі, вітчизняні аналоги — серії 131 і 130) і SN74L/54L (Low power — малопотужна, аналоги — серії 158 і 136).

У 1969 р. розроблена серія SN74S/54S (серії 531 і 530), у 1971 р. — серія SN74LS/54LS (серії 555 і 533), у 1979 р. — серія SN74F/54F фірми Fairchild (FAST — Fairchild's Advanced Schottky TTL, серія 1531), у 1980 р. — серія SN74ALS/54ALS (серія 1533), у 1982 р. — серія SN74AS/54AS (у позначеннях серій S — Schottky, LS — Low power Schottky, ALS — Advanced Low power Schottky, AS — Advanced Schottky, Advanced — удосконалена). Використання діодів з бар'єром Шоткі дозволило значно підвищити швидкодію ІМС за рахунок запобігання глибокого насичення транзисторів у ключовому режимі. Приємною для розроблювача особливістю всіх перерахованих серій є повне співпадання номерів виводів і позначення типу для ІМС однакового функціонального призначення. Наприклад, якщо SN7472 — JK-тригер, то позначення 72 буде присутнє для нього у всіх серіях. Цей же принцип використовується й у вітчизняних ІМС, хоча тип тут позначається буквами. Помітимо, що в EWB 5.0 для всіх цифрових ІС уведена нумерація виводів, що істотно полегшує задачу визначення їхнього функціонального призначення при зіставленні з вітчизняними аналогами.

Як транзистор VT1 використовується багатоемітерний транзистор, відсутній у бібліотеці EWB. Принцип дії вхідного каскаду легко зрозуміти, якщо перейти база-емітер представити у вигляді діодів, як показано на рис. 3.70, б. Тоді очевидно, що якщо вхідні діоди (входи А, В) підключені до шини з високою напругою (3...5 В), то струм резистора R1 потече через колекторний діод у базу транзистора VT2. Якщо ж хоча б один із вхідних діодів підключений до заземлюючої шини або до шини з низькою напругою, то в такий же спосіб, виявиться підключеним і резистор R1. На базі транзистора VT1 при цьому буде низька напруга (перевищуюча вхідну на величину напруги база-емітер) і базовий струм транзистора VT2 стане рівним нулеві.

Таким чином, при високих напругах на обох входах на колекторі транзистора VT1 також буде висока напруга; якщо ж хоча б на один із входів подана близька до нуля напруга, то на колекторі VT1 установиться низька напруга, а це означає, що вхідний транзистор виконує логічну функцію І.

Фазорозподільний каскад виконаний на транзисторі VT2 і резисторах R2, R3, у яких приблизно рівні опори (близько 0,25...0,33 від R1). При цьому насичення транзистора VT2 досягається вже при досить малому коефіцієнті

підсилення струму. Коли на всі логічні входи схеми подана висока напруга, через перехід бази-колектора транзистора VT1 у базу VT2 подається керуючий струм, у результаті чого VT2 відкривається. При цьому напруга в точці Е може зрости тільки до напруги база-емітер транзистора VT4, а напруга в точці С (на колекторі VT2) знизиться до значення, рівного сумі напруг відкритого діода VD і транзистора VT3. Якщо хоча б на один з логічних входів подається низька напруга (сигнал логічного нуля), то транзистор VT1 відкривається, відключаючи керуючий базовий струм транзистора VT2, у результаті чого VT2 закривається і через резистори R2, R3 протікає тільки струм витоку, тому напруги в точках Е і С близькі до нуля і U_{cc} відповідно. Логічна функція АБО може бути реалізована при рівнобіжному з'єднанні двох або більш подібних фазорозподільних каскадів (у точках С і Е).

Основним транзистором вихідного каскаду є транзистор VT3. Коли на входи А, В (рис. 3.70, б) подана висока напруга, транзистори VT2 і VT3 відкриті. У цьому випадку напруга в точці С буде дорівнювати, як зазначено вище, напрузі двох відкритих р-п-переходів. Якщо тимчасово виключити з розгляду транзистор VT4 і розглядати тільки ланцюг, що містить діод VD і транзистор VT3, то напруга в точці S буде нижче напруги в точці С на величину, рівну напрузі на двох р-п-переходах. При цьому напруга на базі транзистора VT4 буде достатньою (саме за рахунок діода VD) для підтримки його у відкритому стані, тобто на виході S буде діяти напруга, рівна напрузі насичення транзистора VT4 (сигнал логічного нуля).

Якщо хоча б на один із входів вентиля А або В подана низька напруга, то транзистори VT2 і VT4 закриті. Через резистор R2 тече тільки струм витоку транзистора VT2, тому напруга в точці С близька до напруги живлення U_{cc} , а потенціал у точці S нижче потенціалу С на величину спадання напруги на двох відкритих переходах. Спаданням напруги на резисторі R2 від базового струму транзистора VT4 можна знехтувати. Таким чином, при наявності хоча б на одному з входів вентиля низької напруги вихідна напруга вентиля нижче напруги живлення на спадання напруги на двох р-п-переходах. У різних серіях ТТЛ використовуються різні схеми вихідних каскадів, однак завжди між шиною U_{cc} і виходом S є два послідовно включених р-п-переходи. Резистор R4 служить для захисту транзистора VT3 при закорочуванні виходу S на "землю".

Базовий елемент серії 54/74 (155) не дуже відрізняється від розглянутого (рис. 3.53). Основна відмінність полягає в тому, що діод VD включений у емітерний, а не в базовий ланцюг транзистора.

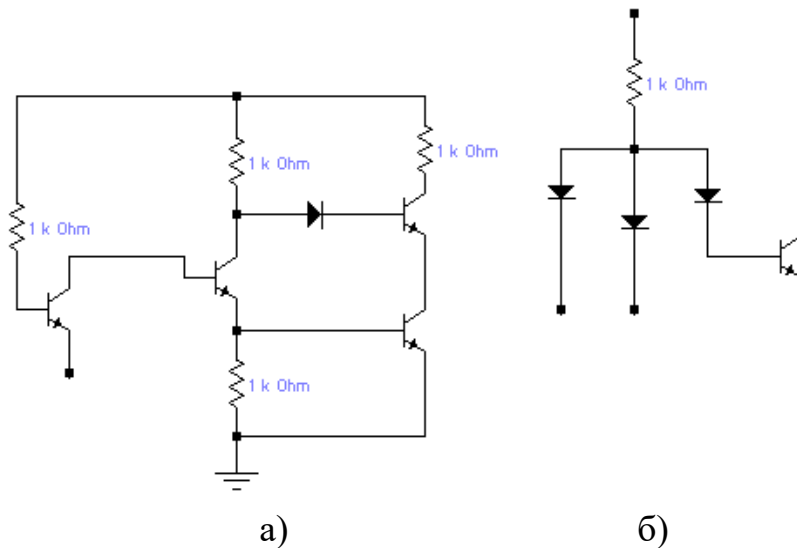


Рисунок 3.53 - Базова схема елемента ТТЛ-серії (а) і еквівалентна схема багатоемітерного транзистора(б)

Контрольні питання та завдання

1. Коли і де були створені перша мікросхема і перший мікропроцесор?
2. Що із себе представляють ТТЛ- і КМДН-серії цифрових ІМС?
3. Що дозволило радикально підвищити швидкодію ТТЛ-серії?
4. Яку основну перевагу мають цифрові КМДН-мікросхеми в порівнянні з ТТЛ і на яких частотах вона виявляється?
5. Які функції виконує багатоемітерний транзистор у ІМС ТТЛ-серії?

Література

1. Бабич М.П., Жуков І.А. Комп'ютерна схемотехніка, МК-Пресс 2004. – 412с.
2. Матвієнко М. П., Розен В.П. Комп'ютерна схемотехніка. Навчальний посібник. — К.: Видавництво Ліра-К, 2016. — 192с.
3. Бучинський М.Я., Горик О.В., Чернявський А.М., Яхін С.В. ОСНОВИ ТВОРЕННЯ МАШИН / [За редакцією О.В. Горика, доктора технічних наук, професора, заслуженого працівника народної освіти України]. – Харків : Вид-во «НТМТ», 2017. — 448 с.
4. Рябенський В.М. Жушков В.Я. Ямненко Ю.С. Заграничний А.В. Схемотехніка: Пристрої цифрової електроніки Електронний підручник для вищих навчальних закладів. – Київ, 2016 – 400 с.
5. Матвієнко М.П. Комп'ютерна схемотехніка. Навчальний посібник. – К.: Видавництво Ліра-К, 2013. – 192 с.
6. Колонтаєвський Ю. П., Сосков А. Г., Промислова електроніка: Теорія і практикум: Підручник./ За ред. А. Г. Соскова. – К.: Каравела, 2013. – 496 с.
7. Матвієнко М.П. Основи електротехніки та електроніки. Підручник. – К.: Видавництво Ліра-К, 2016. – 228 с.
8. Комп'ютерна схемотехніка та архітектура комп'ютерів. Лабораторний практикум/ Укладачі В.М. Єфимець, Є.В. Красовська та ін. – К: НАУ, 2013. - 64с.
9. Комп'ютерна схемотехніка: підручник/ [Азаров О. Д., Гарнага В. А., Клятченко Я. М., Тарасенко В. П.]. – Вінниця: ВНТУ, 2018. – 230 с
10. Комп'ютерна схемотехніка: методичні рекомендації для студентів денної та заочної форми навчання за спеціальністю 123 “Комп'ютерна інженерія”/ уклад. Сидоренко В.В., Минайленко Р.М., Михайлов С.В — Кропивницький: ЦНТУ, 2019. — 72 с.
11. http://citforum.ck.ua/database/advanced_intro
12. ХНАДУ "Файловий архів": <http://files.khadi.kharkov.ua/mekhatroniktransportnikh-zasobiv.html> Конспекти лекцій, методичні вказівки.
13. Світ електронних схем. [Електронний ресурс] – Режим доступу: <http://ua.nauchebe.net>
14. <http://asm.shadrinsk.net>
15. Дистанційна освіта ЦНТУ. – URL: <http://moodle.kntu.kr.ua/my/>