ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АЛМ — арифметично-логічний пристрій

АМ — амплітудна модуляція

АІМ — амплітудно-імпульсна модуляція

БіКМОН — поєднання технологій КМОН з біполярними транзисторними ключами на виході

БМК — базовий матричний кристал

БТ — багатоемітерний транзистор

ВІС — велика інтегральна схема

ЕЗЛ — емітерно-зв'язана логіка

ЕП — елемент пам'яті

ЗП — запам'ятовуючий пристрій

ДТЛ — діодно-транзисторна логіка

ДНФ — диз'юнктивна нормальна форма запису логічних функцій

ДДНФ — досконала диз'юнктивна нормальна форма запису логічних функцій

ДКНФ — досконала кон'юнктивна нормальна форма запису логічних функцій

ІМС — інтегральна мікросхема

ІС — інтегральна схема

ІКМ — імпульсно-кодова модуляція

ІІЛ — інтегральна інжекційна логіка

КНФ — кон'юнктивна нормальна форма запису логічних функцій

КМОН (CMOS) — комплементарні МОН-структури

ЛЕ — логічний елемент

ЛІЗМОН — МОН структури з лавинною інжекцією заряду

МОН — структура на базі з'єднання метал-окисел-напівпровідник

МС — мікросхема

НВІС — надвелика інтегральна схема

ПЗЗ — прилад з зарядовим зв'язком

ПЗП — постійний запам'ятовуючий пристрій

ПКЧ — перетворювач код-частота

ПЛІС — програмована логічна інтегральна схема

ПЛМ — програмована логічна матриця

ПМД — послідовність максимальної довжини

ПМЛ — програмована матрична логіка

ПТП — початкова таблиця переходів скінченного автомата
ОЗП — оперативний запам’ятовуючий пристрій
РПЗП — репрограмований постійний запам’ятовуючий пристрій
РТЛ — резистивно-транзисторна логіка
ТТЛ (TTL) — транзисторно-транзисторна логіка
ТТЛШ — транзисторно-транзисторна логіка з діодами Шоткі
ФАПЧ — фазова автопідстройка частоти
ФД — фазовий детектор
ФМ — фазова модуляція
ФІМ — фазо-імпульсна модуляція
ФНЧ — фільтр нижніх частот
ЦА — цифровий автомат
ЦІС — цифрова інтегральна схема
ЧМ — частотна модуляція
ЧІМ — частотно-імпульсна модуляція
ШІМ — широтно-імпульсна модуляція
CPLD — Complex Programmable Logic Device
EPROM (РПЗП) — Electrically Programmable ROM
PROM (РПЗП) — Programmable Read Only Memory
FPGA — Field Programmable Gate Array
Flex — Flexible Logic Element Matrix
GA (БМК) — Gate Array
ISP — In-System Programming
JFET — Junction Field Effect Tranzistor
MESFET — Metal Semicondactor Field Effect Tranzistor

ВСТУП

Сучасний рівень електронної техніки в значній мірі визначається розвитком технологій цифрової схемотехніки. Зменшення розміру дискретного транзистора і збільшення площі використованих кремнієвих пластин дають можливість забезпечувати схемотехнічну реалізацію алгоритмів досить високої складності. Це дало поштовх у розвитку комп'ютерної техніки, телекомунікацій, мікропроцесорної техніки для управління складними технологічними процесами, побутової електроніки, пристроїв енергетичної електроніки. Поряд з розвитком технологій, інтенсивно вдосконалюються програмні засоби, які значно полегшують задачі проектування і моделювання складних електронних пристроїв на базі мікроконтролерів, ОЕОМ, програмованих логічних матриць.

Сучасні програмні засоби і їх використання на основі мікроконтролерів та програмованих матриць для формування цифрових потоків, цифрової фільтрації, сучасних технологій кодування в значній мірі становляться відірваними від елементної бази, на якій реалізуються складні цифрові системи. До неї звертаються в тих випадках, коли відпрацьовані на моделях алгоритми дають похибки в реальних системах. І тоді розробники вимушені згадувати і про критичні змагання, що мають місце в цифрових схемах, і про порогові рівні і про перешкодостійкість.

Оволодіння цими досягненнями можливе лише на основі знань фундаментальних основ побудови пристроїв і алгоритмів цифрової електроніки, базових алгоритмів функціонування цифрових систем і умінь грамотно і коректно їх використовувати у взаємозв'язку.

Пропонована книга призначена для вивчення основ цифрової схемотехніки, її базових алгоритмів з різноманітних аспектів їх використання для розв'язання практичних задач.

Оскільки програма Multisim дозволяє використовувати два стандарти зображення електронних пристроїв (американський ANSI та міжнародний IEC), а в спеціалізованій літературі можна зустріти оба, в підручнику використані схеми, побудовані за обома стандартами.

ТЕМА 1
ЛОГІЧНІ ЗМІННІ, ОПЕРАЦІЇ ТА ФУНКЦІЇ

1.1. Основні теоретичні положення

Основні визначення. У практиці інженерної діяльності часто мають місце ситуації, при яких має значення не рівень сигналів, що поступають з відповідних датчиків, а лише наявність чи відсутність таких сигналів. Наприклад, у системах охоронної сигналізації необхідно знати, замкнені чи не замкнені двері або вікна в приміщенні, що охороняється і т. п.

Схеми, що дають можливість розв'язувати поставлені задачі, можуть описуватись виразами типу: «лампочка на пульті охоронної сигналізації горить, якщо всі вікна замкнені (точніше, замкнено перше **і** друге **і** третє **і**… вікно)». Або «лампочка не горить, якщо хоча б одне вікно відкрите (тобто може бути відкритим перше **або** друге **або** третє **або** перше **і** друге **або**...)». Такі вирази називаються *логічними*.

При проектуванні подібних систем задаються відповідним рівнем напруги живлення, і наявність чи відсутність її дає можливість одержувати відповіді на поставлені питання. Оскільки рівень напруги може бути різним і задаватись прийнятою елементною базою, то з метою формалізації опису подібних схем приймаються деякі умови. Як приклад, високий рівень напруги приймається за «**1**», низький — відповідно, за «**0**». У такому разі приведені вище вирази можуть бути формалізовані: якщо контакти, що фіксують положення вікон, позначити як аргументи x_1, x_2, …, x_n, які можуть приймати лише значення «**1**» або «**0**», то напругу на лампочці можемо розглядати як функцію y, яка теж приймає одне з двох аналогічних значень.

Математичний апарат, що оперує з аргументами та функціями, які набувають тільки двох значень — «**0**» та «**1**» — називається *двійковою* (*булевою*) *алгеброю* або *алгеброю логіки*. Логічні змінні (аргументи), як і змінні звичайної алгебри, позначаються літерами латинського алфавіту з різними індексами — наприклад, x_0, x_1, x_2, x_3, … Індекс при змінній може одночасно означати розряд двійкового числа. Якщо змінна x_i набуває значення $x_i = 1$, то таке її значення називають *істинним*. Протилежне $x_i = 0$ називають *хибним* і умовно позначають $\overline{x_i}$, що означає заперечення істинного значення аргументу (в зарубіжній практиці операція заперечення позначається апострофом x'). Два елементи булевої алгебри — подія істинна і подія хибна — називають її *константами*.

Булева функція позначається літерою y і є двійковою функцією двійкових аргументів. Умовне її позначення

$$y = f(x_1, x_2, ..., x_n).$$

Булева функція, яка залежить від n аргументів, називається n-вимірною і є повністю визначеною, якщо вказані значення її для всіх двійкових наборів значень її аргументів. Кількість таких наборів дорівнює 2^n. *Тобто, областю визначеності функції **n** змінних є сукупність дискретних точок **n**-вимірного простору, причому кожна з точок є комбінацією значень цих змінних* (*кодовою комбінацією*). Оскільки можливі 2^n різних комбінацій логічних змінних, то область визначення функції складається зі скінченої величини — 2^n точок. Це, в свою чергу, означає, що кожна функція може бути задана таблицею значень, які вона приймає в точках її області визначеності.

Функція повністю визначена, якщо задані її значення в усіх точках області визначеності. Значення функції вибираються з множини «**0**» і «**1**». Якщо ж значення функції не задано в одній або кількох точках, то вона є *неповністю визначеною*. Кодові комбінації, при яких функція невизначена, називаються *факультативними*. У практиці цифрової схемотехніки існує велика кількість неповністю визначених функцій. Довизначення їх, якщо це необхідно, забезпечується встановленням їх значень — «**0**» або «**1**» — довільним шляхом.

Усі можливі логічні функції n змінних можна створити за допомогою трьох основних операцій:

а) логічне заперечення (інверсія, операція **НІ**); позначається рискою над відповідною функцією або аргументом;

б) логічне додавання (диз’юнкція, операція **АБО**), яке позначається символами (V), (+);

в) логічне множення (кон’юнкція, операція **І**), яке позначається символами ($\wedge$), ($\cdot$), (&). Для позначення еквівалентності логічних виразів використовується знак (=).

Запереченням (*інверсією*)називається такий зв’язок між аргументом x та функцією y,при якому y істинна тоді і тільки тоді, коли значення x хибне, і навпаки.

Логічним множенням (*кон'юнкцією*) декількох змінних називається така функція, яка істинна тоді і тільки тоді, коли одночасно істинні всі логічні змінні.

Логічним додаванням (*диз'юнкцією*)декількох змінних називається така функція, яка хибна тоді і тільки тоді, коли одночасно хибні всі додавані змінні.

Слід пам'ятати, що операція кон'юнкції є старшою операцією і виконується раніше диз'юнкції.

Прикладом найпростіших функцій є наступні:

$$y_1 = \overline{x_1}; \qquad y_2 = x_1 \cdot x_2; \qquad y_3 = x_1 + \overline{x_2}.$$

Приклад 1.1. Записати вираз функції трьох змінних, яка приймає істинні значення при умові, що будь-яка пара змінних одночасно має істинні значення.

Розв'язання. Введемо умовні позначення змінних x_0, x_1, x_2.

У загальному плані функція матиме вигляд:

$$y = f(x_2, x_1, x_0).$$

Оскільки істинні значення функції визначаються будь-якою парою логічних змінних, тобто або x_0 і x_1, або x_0 і x_2, або x_1 і x_2, то аналітична форма запису функції прийме вигляд:

$$y_1 = x_0 \cdot x_1 + x_0 \cdot x_2 + x_1 \cdot x_2.$$

Функція може мати і дещо іншу форму запису, якщо враховувати, що при виконанні будь-якої з умов, що закладені в функцію, обмежень на значення третьої змінної не накладається.

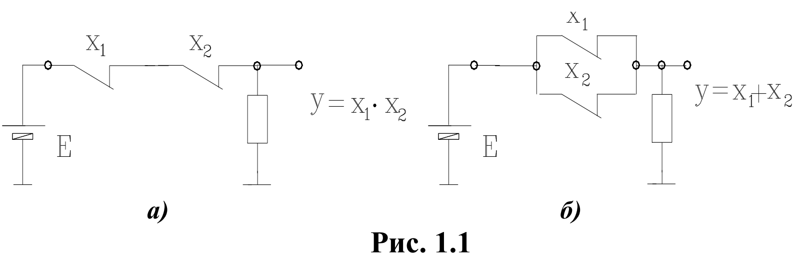

Рис. 1.1

Технічна реалізація булевих функцій, а, відповідно, і їх фізична інтерпретація добре ілюструється за допомогою контактних схем, в яких логічна змінна x_i відповідає замкненому контакту. Схеми, що ілюструють реалізацію операцій кон'юнкції та диз'юнкції, наведені відповідно на рис. 1.1, ***а***, ***б***.

<u>*Закони і тотожності алгебри логіки.*</u> В алгебрі логіки використовується ряд аксіом (тотожностей) та законів. Основними з них є наступні: *переміщувальний* (*властивість комутативності*); *сполучний* (*властивість асоціативності*); *розподільний* (*властивість дистрибутивності*); *інверсії* (*теорема де Моргана*). Головні аксіоми та закони булевої алгебри наведені у табл. 1.1.

Використовуючи наведені у табл. 1.1 закони та тотожності, які використовуються при перетворенні логічних функцій, можна створювати нові, наприклад:

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

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

(у подальшому крапки, що відображають операцію логічного множення у формулах, для спрощення запису приводити не будемо).

Закони інверсії, які відображають властивість взаємного перетворення операцій логічного множення і додавання в алгебрі логіки, називають *принципом подвійності*.

<u>*Способи задання логічних функцій.*</u> Існують такі способи задання або запису логічних функцій — *аналітичний*, *табличний*, *за допомогою карт Карно*, *графічний* та *кубічний*.

Таблиця 1.1

Назва аксіоми чи закону	Вирази
Аксіоми (тотожності)	$0 \cdot x = 0$ $1 + x = 1$ $0 + x = x$ $x \cdot x = x$ $x + x = x$ $x \cdot \overline{x} = 0$ $x + \overline{x} = 1$ $\overline{\overline{x}} = x$
Закони комутативності	$x_1 + x_2 = x_2 + x_1$ $x_1 \cdot x_2 = x_2 \cdot x_1$
Закони асоціативності	$x_1 + x_2 + x_3 = x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3 =$ $= (x_1 + x_3) + x_2$

Закінчення табл. 1.1

Назва аксіоми чи закону	Вирази
	$x_1 \cdot x_2 \cdot x_3 = x_1 \cdot (x_2 \cdot x_3) =$ $= x_2 \cdot (x_1 \cdot x_3) = x_3 \cdot (x_1 \cdot x_2)$
Закони дистрибутивності	$x_1 \cdot (x_2 + x_3) = x_1 \cdot x_2 + x_1 \cdot x_3$ $x_1 + x_2 \cdot x_3 = (x_1 + x_2) \cdot (x_1 + x_3)$
Закони інверсії (теорема де Моргана, принцип подвійності)	$\overline{x_1 \cdot x_2} = \overline{x_1} + \overline{x_2}$ $\overline{x_1 + x_2} = \overline{x_1} \cdot \overline{x_2}$
Закони поглинання	$x_1 + x_1 \cdot x_2 = x_1$ $x_1 \cdot (x_1 + x_2) = x_1$

Аналітично логічна функція може бути записана різними комбінаціями кон'юнкцій та диз'юнкцій логічних змінних. Зазвичай логічні функції записуються або у вигляді суми добутків логічних змінних (диз'юнкція кон'юнкцій) або у вигляді логічного добутку їх сум (кон'юнкція диз'юнкцій). Наведення функції у вигляді диз'юнкції кон'юнкцій називають *диз'юнктивною нормальною формою* (**ДНФ**):

$$y = x_1 \overline{x_2} + \overline{x_1} x_3 + x_1 x_2 \overline{x_3},$$

а запис у вигляді кон'юнкції диз'юнкцій — відповідно, *кон'юнктивною нормальною формою* (**КНФ**):

$$y = (x_1 + x_2)(x_2 + \overline{x_3})(\overline{x_1} + x_2 + x_3).$$

Інверсія у відповідності з теоремою де Моргана будь-якої функції, приведеній в одній формі, призводить до заміни запису на іншу форму.

Наприклад, інверсія функції

$$y = x_1 + x_2 \overline{x_3} + x_1 \overline{x_2} x_3$$

представляється у вигляді:

$$\overline{y} = \overline{x_1}(\overline{x_2} + x_3)(\overline{x_1} + x_2 + \overline{x_3}).$$

Будь-яка логічна функція, задана в аналітичній формі, може бути перетворена на **ДНФ** або **КНФ** за допомогою тотожностей та законів алгебри логіки. При цьому для однієї і тієї ж функції може існувати декілька рівнозначних диз'юнктивних та кон'юнктивних нормальних форм.

У той же час, існує лише один вид **ДНФ** та **КНФ**, в яких функція може бути записана єдиним чином. Такі форми називаються *досконалими диз'юнктивними* (*кон'юнктивними*) *нормальними формами* (**ДДНФ**, **ДКНФ**). Вони характеризуються тим, що в **ДДНФ** кожна кон'юнкція, а в **ДКНФ** кожна диз'юнкція містять усі логічні змінні даної функції, з інверсіями або без них.

Прикладами **ДДНФ** та **ДКНФ** запису є функції чотирьох змінних

$$y_1 = x_3 \overline{x_2} x_1 \overline{x_0} + \overline{x_3} x_2 x_1 \overline{x_0} + x_3 \overline{x_2} x_1 x_0 \;;$$

$$1010 \qquad 0110 \qquad 1011$$

$$y_2 = (x_1 + \overline{x_2} + \overline{x_3} + x_4)(\overline{x_1} + x_2 + \overline{x_3} + x_4)\,.$$

Оскільки кожна кон'юнкція функції, що наведена у **ДДНФ**, визначає її істинне значення, відповідаюче «**1**», то такі кон'юнкції називаються *конституєнтами одиниці* (*мінтермами*). Аналогічно, диз'юнкції функції, що наведені у **ДКНФ**, називаються *конституєнтами нуля* (*макстермами*).

Якщо замінити логічні змінні та їх заперечення одиницями та нулями, то кожна кон'юнкція буде представляти собою двійкове число.

Це дозволяє, наприклад, вище наведену логічну функцію y_1 записати у вигляді:

$$y_1 = \bigvee_0^{15} 6, 10, 11.$$

Така форма називається *досконалою скороченою диз'юнктивною формою* або *канонічною сумою*.

Аналогічно, функцію можна зобразити і у вигляді добутку макстермів. Така форма запису називається *канонічним добутком*. Наприклад:

$$y = \bigwedge_0^{7} 2, 4 = (x_2 + \overline{x_1} + x_0)(\overline{x_2} + x_1 + x_0)\,.$$

Легко бачити можливість конвертації в представленні функції у вигляді макстермів та мінтермів, оскільки кожна з них доповнює функцію до повного перебору логічних змінних. Як приклади, можемо записати:

$$y = \bigvee_0^{7} 2, 6, 7 = \bigwedge_0^{7} 0, 1, 3, 4, 5\;;$$

$$y = \bigvee_0^{7} 0, 1, 3, 5 = \bigwedge_0^{7} 2, 4, 6, 7\;;$$

$$y = \bigvee_0^{15} 0, 4, 5, 9, 11, 13, 15 = \bigwedge_0^{15} 1, 2, 3, 6, 7, 8, 10, 12, 14\,.$$

Таблиця 1.2

x_1	x_0	y_1	y_2	y_3	y_4	y_5
0	0	0	0	0	0	1
0	1	1	1	1	0	0
1	0	1	1	1	0	0
1	1	x	1	0	1	1

Індекси біля умовних позначень операцій диз’юнкції та кон’юнкції вказують на діапазон можливих мінтермів та макстермів логічних функцій. Нижній індекс іноді не вказується.

Досконала диз’юнктивна нормальна форма запису дозволяє легко перейти до інших форм запису — *табличної* та *карт Карно*.

У табл. 1.2 наведені функції $y_1 \ldots y_5$ двох змінних x_0 та x_1. Табличний спосіб полягає у тому, що функція задається у вигляді таблиці відповідності (таблиці істинності станів). У таблицю вписують усі можливі комбінації аргументів у порядку зростання їх індексів і при кожній комбінації встановлюється значення функції. Кількість всіх можливих сполук аргументів, а, отже, і кількість значень функції дорівнює 2^n, де n — кількість логічних змінних. З табличної форми запису легко перейти до аналітичної, використовуючи досконалу диз’юнктивну форму запису логічних функцій. Для цього функція записується як диз’юнкція конституєнт одиниці. Наприклад, функцію y_3 з табл.1.2 можемо записати у вигляді:

$$y_3 = \overline{x_1}\, x_0 + x_1\, \overline{x_0}\,.$$

Ця функція може бути записана і з використанням нульових її значень:

$$\overline{y_3} = \overline{x_1}\, \overline{x_0} + x_1\, x_0.$$

Використовуючи властивість подвійної інверсії, легко встановити тотожність обох форм запису.

Логічна функція іноді може бути неповністю визначеною. При переході до аналітичної форми запису вона повинна бути до визначена, тобто, довільним шляхом слід встановити значення змінних, оскільки від них значення функції не залежить.

Найбільш широке використання знаходять функції **І**, **АБО**, **І-НІ**, **АБО-НІ**, **ВИКЛ. АБО**. Вони універсальні тому, що за їх допомогою можна записати будь-яку іншу функцію.

Розроблений математичний апарат аналізу та синтезу булевих функцій найбільш відповідає цим функціям. Набір логічних функцій **І**, **АБО**, **НІ** дозволяє реалізувати будь-яку іншу функцію і називається *функціонально повним*.

1.2. Використання *Multisim* для проведення досліджень

Для вивчення основ побудови логічних функцій, різних форм їх представлення та мінімізації в програмному продукті ***Multisim*** використовується логічний перетворювач, ***Logic converter*** (рис. 1.2), за допомогою якого можна розв'язувати вказані задачі.

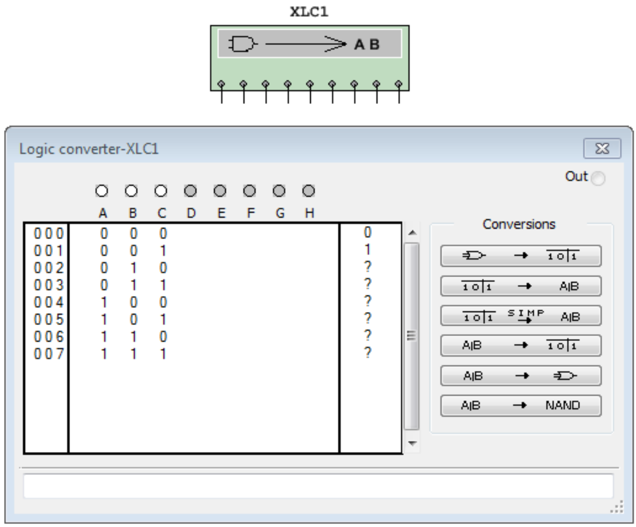

Рис. 1.2

Логічні змінні в конверторі представлені у вигляді заглавних букв латинського алфавіту ***A***, ***B***, ***C***, ***D***, ***E***, ***F***, ***G***, ***H***. Звідси витікає, що таблиця перебору станів логічних змінних може містити 256 строчок, а розмір строчки буде досягати 1 байт. Кожна строчка в лівому стовбці нумерується відповідним десятковим кодом — номером мінтерму. В правому стовбці за допомогою лівої кнопки «миші» встановлюється істинне, або хибне значення функції, яка задається. Таким шляхом може бути сформульована таблиця істинності логічної функції. Зчитуючи істинні значення функції з таблиці можна отримати **ДДНФ** її запису, в тому числі у вигляді канонічної суми.

З правої сторони вікна розміщення таблиці знаходяться кнопки, за допомогою яких конвертор виконує ті, чи інші перетворення: нумеруючи їх зверху до низу запишемо основні операції конвертора:

1. Забезпечення перетворення логічної схеми в таблицю станів логічної функції.

2. Забезпечення перетворення таблиці станів у логічну функцію, представлену у вигляді **ДДНФ**.

3. Забезпечення перетворення таблиці станів у мінімізовану логічну функцію.

4. Перетворення логічної функції у її таблицю істинності.

5. Перетворення логічної функції у логічну схему в базисі елементів **І-АБО-НІ**.

6. Перетворення логічної функції у логічну схему в базисі елементів **І-НІ**.

Для дослідження часових характеристик логічних функцій можна використовувати багатопроменевий осцилограф, за допомогою якого можна спостерігати за 4-ма процесами (три логічні змінні і одна логічна функція), але в ***Multisim*** для такої мети може використовуватись логічний аналізатор (***Logic Analyzer***), зовнішній вигляд якого в робочому стані і умовне позначення наведені на рис. 1.3.

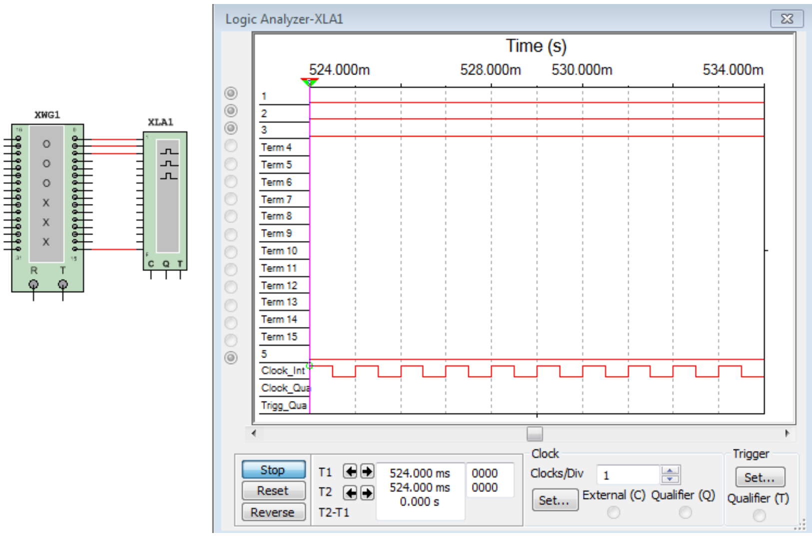

Рис. 1.3

Налагодження логічного аналізатора забезпечується установками в двох вікнах . Вікно установки тактової частоти (***Clock Setup***) виконується у вікні, яке активізується кнопкою ***Set…***, що розміщена на середині панелі керування. Установка фронту/зрізу синхронізації вхідних сигналів виконується у вікні, що активізується за допомогою кнопки ***Set…***, що розміщена справа на панелі керування аналізатора. Слід пам'ятати, що керування кольором екрану аналізатора можливо лише у тому випадку, коли до виводів, що зображені на його умовному позначенні, приєднані інформаційні провідники цифрової схеми. Зліва від екрану аналізатора на вертикальній панелі активізуються відповідні входи.

Для генерації бінарного коду, що відповідає таблиці перебору логічних змінних використовується генератор бінарних слів (***Word generator***) (рис. 1.4.)

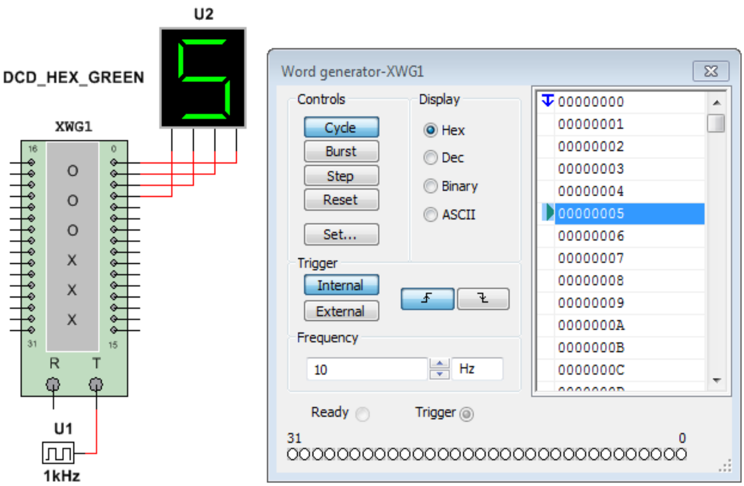

Рис. 1.4

На умовному позначенні його (рис. 1.4, зліва) передбачені 32 виводи, на виходах яких появляються логічні сигнали «1» та «0» (молодші розряди починаються на правій панелі зверху). Тобто, генератор може генерувати бінарні слова (комбінації логічних змінних) розміром в 32 розряди. Як видно з панелі керування генератор може створювати гексадецимальний, бінарний, десятковий коди, а також код ASCII — для кодування букв латинського алфавіту. В поточній темі необхідно використовувати гексадецимальний код. Цей код, рівно, як і інші, може генеруватись у циклічному режимі, (***Cycle***) тобто, повторюватись довільну кількість разів, в пакетному режимі (***Burst***), в покроковому режимі (***Step***).

Кнопкою ***Set…*** активізується вікно налагодження (рис. 1.5).

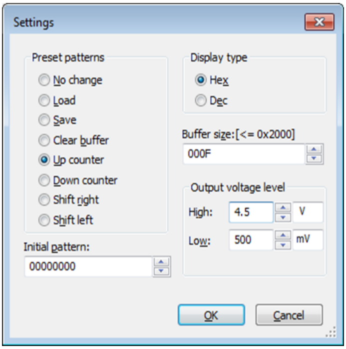

Рис. 1.5

В ньому установлюються початкове і кінцеве значення коду, що генерується, а також порядок генерації бінарних комбінацій логічних змінних (бінарних слів). Логічні елементи можна вибирати з меню ***Place TTL***, меню ***Place CMOS***, меню ***Place Misc Digital***. Рекомендується, поки-що, вибирати типові логічні елементи з меню ***Place Misc Digital***, опція ***TTL***.

На рис. 1.6 наведена група логічних елементів, що призначені для реалізації функцій двох змінних та інвертор ***NOT***. Для проведення досліджень можна вибрати елементи, що мають більше входів.

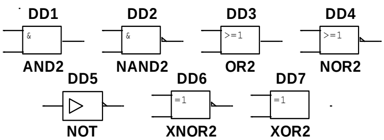

Рис. 1.6

В якості прикладу на рис. 1.7. наведена схема для демонстрації часових характеристик типових двоходових логічних елементів за допомогою приладів, що описані вище.

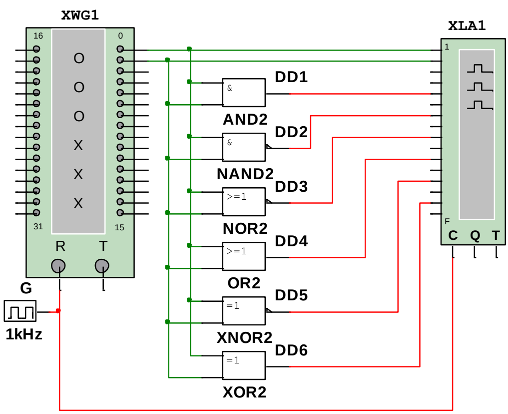

Рис.1.7