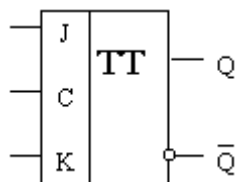


ОСНОВИ СУЧАСНОЇ ЕЛЕКТРОНІКИ

ЧАСТИНА 2

ДИСКРЕТНА ЕЛЕКТРОНІКА

Лабораторний практикум



Навчально-методичний посібник
для студентів фізико-математичного факультету

Рецензенти:

Сидорук В.А. – кандидат фізико-математичних наук, старший науковий співробітник відділу чисельних методів та комп'ютерного моделювання Інституту кібернетики імені В.М. Глушкова НАН України.

Іванюк В. А. – кандидат технічних наук, доцент, доцент кафедри інформатики Кам'янець-Подільського національного університету імені Івана Огієнка.

Укладачі: Поведа Р.А., Оптасюк С.В.

Поведа Р.А. *Основи сучасної електроніки. Лабораторний практикум. Частина 2. Дискретна електроніка: навчально-методичний посібник для студентів* / Р.А. Поведа, С.В. Оптасюк. – Кам'янець-Подільський: Аксіома, 2021. – 70 с.

Навчально-методичний посібник містить лабораторні роботи з основ сучасної цифрової електроніки, які запропоновано виконувати в сучасному віртуальному середовищі моделювання процесів в електронних схемах Multisim Electronics Work Bench фірми National Instruments. В процесі виконання завдань лабораторних робіт студенти мають можливість детально дослідити роботу основних компонентів цифрових електронних схем (логічних елементів, тригерів, лічильників, регістрів зсуву, цифро-аналогових та аналогово-цифрових перетворювачів, дешифраторів, мультиплексорів) та експериментально встановити їх характеристики та логіку роботи. Кожна робота містить відповідні до теми розгорнуті теоретичні відомості, вказівки та рекомендації до виконання завдань. Особливістю посібника є розміщення QR-посилань, завдяки яким студенти мають можливість з допомогою смартфона до кожної роботи завантажити готовий до виконання файл Electronics Work Bench відповідної лабораторної роботи.

Посібник рекомендований для студентів освітнього рівня підготовки «магістр» та «бакалавр» фізико-математичного факультету Кам'янець-Подільського національного університету імені Івана Огієнка та усіх, кого цікавлять питання сучасної електроніки.

Рекомендовано до друку науково-методичною радою фізико-математичного факультету Кам'янець-Подільського національного університету імені Івана Огієнка, протокол № 3 від 23 березня 2021 року.

© Поведа Р.А., Оптасюк С.В. 2021

Передмова

Досягнення сучасної цивілізації ХХІ століття (комп'ютери, інтернет, навігація, мобільний зв'язок, освоєння космосу) були б неможливі без дискретної (цифрової електроніки), основи функціонування якої заклали Джордж Буль та Чарлз Беббідж ще у ХІХ столітті.

Навчально-методичний посібник **«Основи сучасної електроніки. Частина 2. Дискретна електроніка.»** дозволяє ознайомитись з основними «цеглинками» - функціонально достатніми елементами сучасної цифрової електроніки. В процесі виконання завдань лабораторних робіт студенти мають можливість детально дослідити роботу основних компонентів цифрових електронних схем: основних логічних елементів, усіх основних типів тригерів, лічильників, регістрів зсуву, цифро-аналогових та аналогово-цифрових перетворювачів, дешифраторів, мультиплексорів.

Аналіз стану програмного забезпечення з схемотехнічного моделювання довів, що на етапі початкового освоєння методів автоматизованого проектування та на етапах проведення пошуково-дослідницьких робіт доцільно використовувати програмне середовище Electronics Workbench (далі ЕWB), яке легко засвоюється та досить зручне в роботі. Програма Electronics Workbench розроблена фірмою National Instruments Electronics Workbench Group і використовується для схемотехнічного моделювання аналогових та цифрових радіоелектронних пристроїв різного призначення та дає змогу простими засобами будувати на екрані монітора аналогову або цифрову електронну схему, підключати електро- та радіовимірювальні прилади та отримувати результати або в вигляді цифрових даних на моніторах вимірювальних приладів або на логічному аналізаторі.

Студенти спеціальностей фізико-математичного факультету повинні добре знати фізичні засади комп'ютера, розуміти логічний взаємозв'язок всіх його функціональних блоків, взаємодію апаратного та програмного складників, щоб осмислено ставити та вирішувати складні практичні завдання.

Лабораторні роботи виконуються для закріплення теоретичних знань, отриманих студентами під час лекційних занять та самопідготовки. Метою виконання лабораторних робіт є вироблення у студентів навичок та умінь самостійної роботи при вивченні фізичних процесів, які проходять в електронних приладах. У кожній лабораторній роботі навчального-методичного посібника наведені теоретичні положення за темою проведення роботи, практичні схеми та докладний опис моделювання та аналізу схеми для кожного прикладу лабораторної роботи.

Лабораторна робота №1

ЛОГІЧНІ ЕЛЕМЕНТИ І СХЕМИ

МЕТА РОБОТИ:

Ознайомлення з основними характеристиками логічних елементів і основами синтезу логічних схем.

ТЕОРЕТИЧНІ ВІДОМОСТІ ТА РОЗРАХУНКОВІ ФОРМУЛИ

1. ВИЗНАЧЕННЯ КОМБІНАЦІЙНИХ І ПОСЛІДОВНИХ ПРИСТРОЇВ

Пристрої, що реалізують функції алгебри логіки, називають *логічними* чи *цифровими* і класифікують за різними ознаками. Так, за характером інформації на входах і виходах логічні пристрої підрозділяють на пристрою послідовної, паралельні і змішаного дії, а по схемному рішенню і характеру зв'язку між вхідними і вихідними змінними з урахуванням їх зміни по тактах роботи – на комбінаційні і послідовні.

У *комбінаційних* пристроях значення сигналів (0 або 1) на виходах в кожен момент часу повністю визначаються значеннями вхідних цифрових сигналів, що діють в даний момент. У *послідовних* же пристроях значення вихідних сигналів в n -такті визначаються не лише значеннями вхідних сигналів в цьому такті, але і залежать від внутрішніх станів пристроїв, які сталися в результаті дії вхідних сигналів в попередні такти.

Ця робота присвячена вивченню простих комбінаційних логічних пристроїв, що реалізують логічні функції додавання, множення і заперечення.

2. ОСНОВНІ ЕЛЕМЕНТИ АЛГЕБРИ ЛОГІКИ

Аналіз комбінаційних пристроїв зручно проводити за допомогою алгебри логіки, що оперує тільки з двома поняттям: істинним (логічна 1) і неправдивим (логічний 0). У результаті, функції, що відображають інформацію, приймають в кожен момент часу тільки значення 0 або 1. Такі функції називають *логічними* а сигнали - *двійковими* (бінарними).

Схемні елементи, за допомогою яких здійснюється перетворення сигналів, що поступають на їх входи і виконується реалізація логічних операцій, називають *логічними* пристроями.

У загальному випадку логічний пристрій може мати n входів і m виходів. Розглядаючи вхідні сигнали x_1, x_2, \dots, x_n у якості аргументів, можна відповідні вихідні сигнали представляти у вигляді функції $y_i = f(x_0, x_1, x_2, \dots, x_n)$ за допомогою операцій алгебра логіки.

Функції алгебра логіки (ФАЛ), що іноді називаються *комутуючими* функціями, зазвичай представляють у формі алгебри, наприклад $y_i = (x_0 \wedge x_1) \vee (x_1 \wedge x_2)$, або у вигляді таблиць істинності.

Таблиця істинності містить всілякі комбінації (набори) бінарних значень вхідних змінних з тими, що відповідають їм бінарними значеннями вихідних перемінних; кожному набору вхідних сигналів відповідає певне значення

вихідного сигналу – значення логічної функції y_i . Максимальне число можливих різних наборів (рядків) залежить від числа вхідних змінних n і рівне 2^n .

У булевій алгебрі виділяють три основні функції: кон'юнкція, диз'юнкція, заперечення. Інші функції є похідними від приведених вище.

Основні логічні операції складаються з наступних елементарних перетворень двійкових сигналів:

- логічне складання чи диз'юнкція яка позначається символом " \vee " (" $+$ ") і називається також операцією **АБО**. При цьому число аргументів (доданків x) може бути будь-яким. Ця операція для функції двох змінних x_1 і x_2 описується у вигляді логічної формули

$$y = x_1 \vee x_2 = x_1 + x_2.$$

Це означає, що y істинно (рівне 1), якщо істинний хоч би один з доданків x_1 чи x_2 . І тільки у разі, коли усі доданки x рівні 0, результат логічного складання y також рівний 0.

Умовне позначення, таблиця істинності і інші показники цієї логічної функції приведені в другому стовпці таблиці. 1.1;

- логічне множення чи кон'юнкція що позначається символом " \wedge " (" \cdot ") і називається також операцією **І**. При цьому число аргументів (співмножників x) може бути будь-яким. Ця операція для функції двох змінних x_1 і x_2 описується у вигляді логічної формули

$$y = x_1 \wedge x_2 = x_1 \cdot x_2 = x_1 x_2.$$

Це означає, що y істинно (рівне 1), якщо істинні співмножники x_1 і x_2 . У випадку, якщо хоч би один із співмножників дорівнює 0, результат логічного множення y рівний 0.

Умовне позначення, таблиця істинності і інші показники логічної функції **І** приведені в третьому стовпці таблиці. 1.1;

- логічне заперечення чи інверсія, позначається рискою над змінною і називається операцією **НЕ**. Ця операція записується у виді

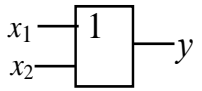
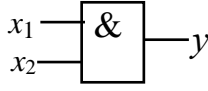
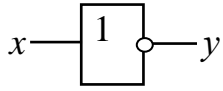
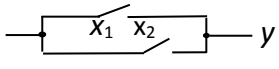
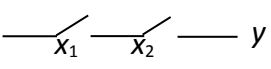
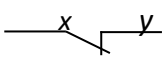
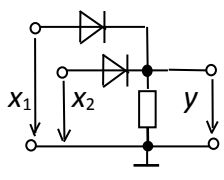
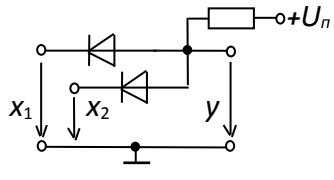
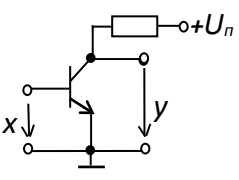
$$y = \bar{x}.$$

Це означає, що y істинно (рівне 1), якщо x неправдиво (рівне 0) і навпаки. Очевидно, що операція y виконується над однією змінною x і її значення завжди протилежно цій змінній (четвертий стовпець таблиці. 1.1).

Основні логічні операції АБО, І та НЕ дозволяють аналітично описати, а логічні елементи АБО, І, НЕ – реалізувати логічними пристроями будь-якої складності: операції $y = x_1 + x_2$, $y = x_1 x_2$ і $y = \bar{x}$ володіють функціональною повнотою і складає функціонально повний набір.

Як приклад розглянемо функцію нерівнозначності у двох змінних x_1 і x_2 що приймає значення 1 при $x_1 \neq x_2$ і значення 0 при $x_1 = x_2 = 0$ або при $x_1 = x_2 = 1$, тобто $y = \bar{x}_1 x_2 + x_1 \bar{x}_2$.

Таблиця 1.1

Форми відображення основних логічних функцій																																							
Найменування функції	Диз'юнкція	Кон'юнкція	Інверсія																																				
Символічна	$\vee / +$	\wedge / \cdot	\bar{x}																																				
Буквена	АБО	І	НЕ																																				
Умовна графічна																																							
Аналітична	$y = x_1 \vee x_2 = x_1 + x_2$	$y = x_1 \wedge x_2 = x_1 x_2$	$y = \bar{x}$																																				
Таблична (істинності)	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>x_1</td><td>x_2</td><td>y</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	x_1	x_2	y	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>x_1</td><td>x_2</td><td>y</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	x_1	x_2	y	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>x</td><td>y</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	x	y	0	1	1	0
x_1	x_2	y																																					
0	0	0																																					
0	1	1																																					
1	0	1																																					
1	1	1																																					
x_1	x_2	y																																					
0	0	0																																					
0	1	0																																					
1	0	0																																					
1	1	1																																					
x	y																																						
0	1																																						
1	0																																						
Контактна																																							
Схемотехнічна																																							

Операцію нерівнозначності частіше називають *підсумовуванням по модулю 2* і позначають $y = x_1 \oplus x_2$.

Приклади контактної і простої схемної реалізацій диз'юнктора, кон'юнктора і інвертора приведені в передостанній і останній рядках таблиці. 1.1.

3. БАЗОВІ ЛОГІЧНІ ЕЛЕМЕНТИ

Особливе значення в цифровій електроніці мають універсальні логічні елементи, здатні утворити функціонально повний набір, з допомогою яких можна реалізувати синтез пристроїв будь-якої складності. При інтегральній технології зручність виготовлення одного базового елемента має вирішальне значення. Тому базові логічні пристрої складають основу більшості цифрових ИМС.

До універсальних логічних операцій відносять два базових елемента:

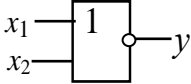
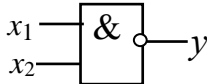
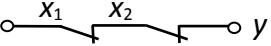
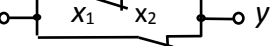
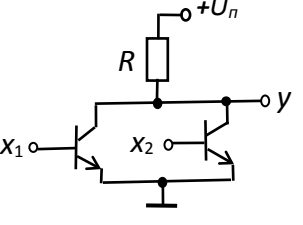
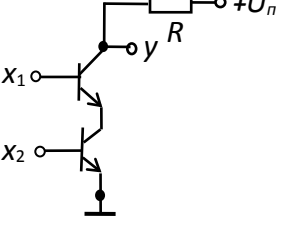
1- *функція Пірсу*, яка позначається символічно вертикальною стрілкою \downarrow (стрілка Пірсу) що відповідає операції **АБО-НЕ**. Для простої функції двох змінних x_1 і x_2 функція $y = 1$ тоді і тільки тоді, коли $x_1 = x_2 = 0$:

$$y = x_1 \downarrow x_2 = x_1 + x_2;$$

2- функція Шеффера, яка позначається символічно вертикальною рисою $|$ (штрих Шеффера) що відповідає операції **I-НЕ**. Для простої функції двох змінних x_1 і x_2 функція $y = 0$ тоді і тільки тоді, коли $x_1 = x_2 = 1$:

$$y = x_1 | x_2 = \overline{x_1 x_2}.$$

Т а б л и ц я 1.2

Форми відображення базових логічних функцій																																
Найменування функції	Функція Пірсу	Функція Шеффера																														
Символічна	↓																															
Буквена	АБО-НЕ	I-НЕ																														
Умовна графічна																																
Аналітична	$y = x_1 \downarrow x_2$	$y = x_1 x_2$																														
Таблична (істинності)	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td>x_1</td><td>x_2</td><td>y</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	x_1	x_2	y	0	0	1	0	1	0	1	0	0	1	1	0	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td>x_1</td><td>x_2</td><td>y</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	x_1	x_2	y	0	0	1	0	1	1	1	0	1	1	1	0
x_1	x_2	y																														
0	0	1																														
0	1	0																														
1	0	0																														
1	1	0																														
x_1	x_2	y																														
0	0	1																														
0	1	1																														
1	0	1																														
1	1	0																														
Контактна																																
Схемотехнічна																																

При одних і тих же значеннях аргументів обидві функції відображають операцію інверсії. Найважливіші показники функцій Шеффера і Пірсу представлені в таблиці. 1.2.

У останньому рядку таблиці. 1.2 наведені приклади побудови двохходової схеми АБО-НЕ, в якій до резистора навантаження R підключені колектори двох паралельно включених біполярних транзисторів $p-n-p$ -типа, емітери яких заземлені, і схеми I-НЕ, в якій послідовно включені два біполярні транзистори $p-n-p$ -типа (емітер нижнього транзистора підключений до землі) де R - резистор навантаження.

4. ПРЕДСТАВЛЕННЯ ЛОГІЧНИХ ФУНКЦІЙ МАТЕМАТИЧНИМИ ВИРАЗАМИ

Найбільш поширеним способом представлення логічних функцій є таблична форма. Таблиці істинності дозволяють повно і однозначно встановити всі існуючі логічні зв'язки.

При табличному представленні логічні функції записують в одній з канонічних форм: досконалій диз'юнктивній нормальній формі (ДДНФ) або досконалій кон'юнктивній нормальній формі (ДКНФ).

Математичне вираження логічної функції в ДДНФ отримують з таблиці істинності таким чином: для кожного набору аргументів, на якому функція рівна 1, записують елементарні добутки змінних, причому змінні, значення яких дорівнюють нулю, записують з інверсією. Отримані добутки, що називаються *конституентами одиниці* або *мінтермами* додають.

Запишемо логічну функцію у трьох змінних a, b і c наданою у вигляді таблиці.

Таблиця 1.3

№	a	b	c	y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0

1.3, в ДДНФ:

$$y(a, b, c) = \bar{a}bc + a\bar{b}c + abc\bar{c} + abc.$$

Досконалою кон'юнктивною нормальною формою (ДКНФ) називають логічний добуток елементарних сум, в кожній з яких аргумент або його заперечення входять один раз.

При цьому для кожного набору аргументів таблиці істинності, на якому функція y рівна 0, складають елементарну суму, якщо значення змінних дорівнює 1, записують з запереченням. Отримані суми, що називають *конституентами нуля* або *макстермами*,

об'єднують операцією логічного множення.

Для функції (таблиця 1.3) ДКНФ

$$y(a, b, c) = (a + b + c)(a + b + \bar{c})(\bar{a} + b + c)(\bar{a} + \bar{b} + c).$$

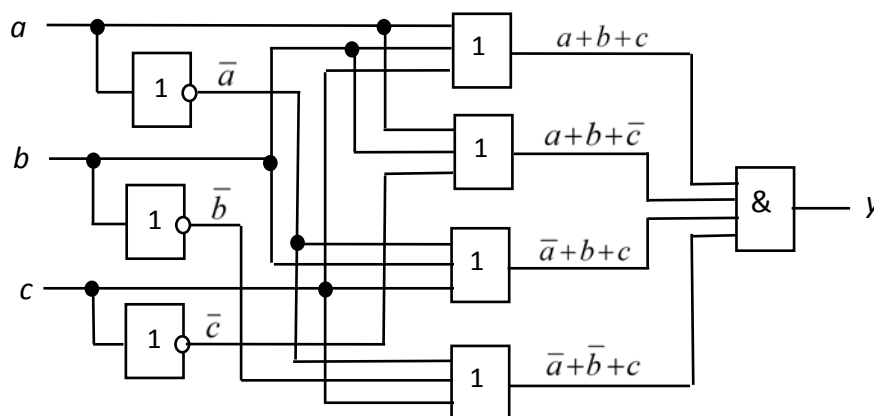
5. ПЕРЕХІД ВІД ЛОГІЧНОЇ ФУНКЦІЇ ДО ЛОГІЧНОЇ СХЕМИ

Для побудови логічної схеми необхідно логічні елементи розташовувати в порядку, вказаному у булевому виразі починаючи від входу.

Побудуємо структуру логічного пристрою, що реалізовує логічну функцію трьох змінних

$$y = (a + b + c)(a + b + \bar{c})(\bar{a} + b + c)(\bar{a} + \bar{b} + c).$$

Ліворуч розташовуємо входи a, b і c з відгалуженнями на три інвертори, потім чотири елементи АБО та елемент І на виході (мал. 1.1).



Мал. 1.1

Отже, будь-яку логічну функцію можна реалізувати безпосередньо по виразах, представленим у вигляді ДДНФ або ДКНФ. Проте, отримана таким чином схема, як правило, не оптимальна з точки зору її практичної реалізації: вона громіздка, містить багато логічних елементів і виникають труднощі в забезпеченні її високої надійності.

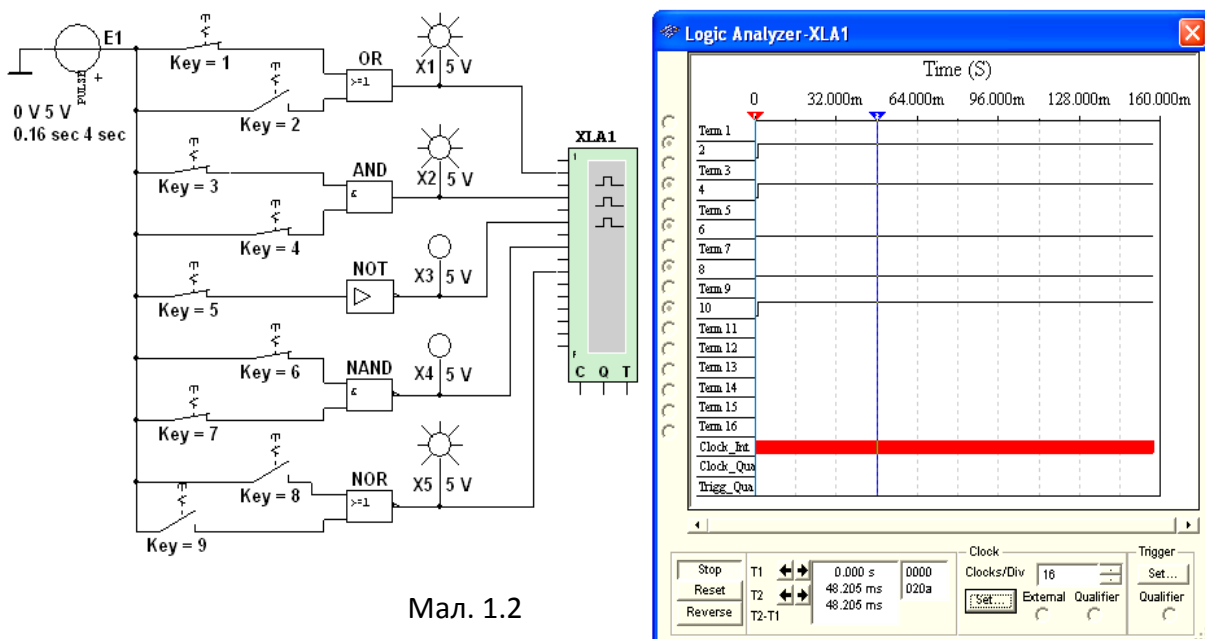
Алгебра логіки дозволяє спростити формули, що описують складні висловлювання. Це допомагає в кінцевому підсумку визначити оптимальну структуру того або іншого логічного пристрою. Під оптимальною структурою прийнято розуміти таку побудову логічного пристрою, при якому число елементів, що входять до його складу, мінімальне або стандартизоване до базових технологічних вимог.

ЗАВДАННЯ І МЕТОДИЧНІ ВКАЗІВКИ ДО ЇХ ВИКОНАННЯ

Завдання 1. Запустити комплекс Multisim Electronics Workbench. На робочому полі віртуальної лабораторії зібрати схему для дослідження основних і базових логічних елементів (див. мал. 1.2) або відкрити відповідний файл і встановити у діалогових вікнах компонентів їх параметри або режими роботи. Скопіювати схему (мал. 1.2) на сторінку звіту в електронному файлі звіту.

Схема (мал. 1.2) зібрана на двійкових основних [OR (АБО), AND (І), NOT (НЕ)] і універсальних (базових) [NAND (І-НЕ) і XOR (АБО-НЕ)] логічних елементах, розташованих у бібліотеці **Misc Digital/TIL** з рівнем логічної «1» у 5В. В схему включені ключі 1, 2, ..., 9 пробники X1, X2, ..., X5 з пороговою напругою 5В, генератор прямокутних сигналів E1 з амплітудою $E=5В$, тривалістю імпульсу $t_i=0,16 з$ і періодом $T=4 з$, і логічний аналізатор XLA1.

Для зручності виміру сигналів виходи логічних елементів підключені до входів 2, 4, 6, 8 і 10 аналізатора XLA1. При моделюванні відбувається повільна розгортка часових діаграм у вікні аналізатора. По досягненню інтервалу часу



Мал. 1.2

рівному 70-80% ширина вікна, слід за допомогою кнопки **Run/Stop** вимикати процес моделювання.

Оперуючи ключами **1, 2, ..., 9**, сформувані всі можливі комбінації аргументів x_1 і x_2 (00, 01, 10 і 11) на вході диз'юнктора (**OR**), кон'юнктора (**AND**), штриха Шеффера (**NAND**) і стрілки Пірсу (**NOR**) і записати значення вихідних логічних функцій y_k (0 або 1) у таблицю 1.4.

Зауважемо, що якщо ключ замкнутий, то на цей вхід елементу буде подана логічна одиниця (потенціал 5 В), а при розімкненому ключі - логічний нуль. Оскільки інвертор (**NOT**) має один вхід, то для формування двох значень вхідного сигналу (логічної «1» або «0») вистачає одного ключа **5**.

Значення функцій досліджуваних елементів можна контролювати за допомогою пробників **X1, X2, ..., X5**: якщо вихідний сигнал елементу рівний логічній одиниці, то включений на виході цього елементу пробник «світиться». Так, при положенні ключів схеми як на мал. 1.2 функції елементів **OR, AND** і **NOR** дорівнюють логічній одиниці.

Таблиця 1.4

Диз'юнктор [ЧИ (OR)]			Кон'юнктор [І (AND)]			Інвертор [НЕ (NOT)]		Штрих Шеффера [І-НЕ (NAND)]			Стрілка Пірсу [ИЛИ-НЕ (NOR)]		
x_1	x_2	y	x_1	x_2	y	x	y	x_1	x_2	y	x_1	x_2	y
0	0		0	0		0		0	0		0	0	
0	1		0	1				0	1		0	1	
1	0		1	0		1		1	0		1	0	
1	1		1	1				1	1		1	1	

Завдання 2. "Перетягнути" з бібліотеки **Misc Digital\TTL** на робоче поле середовища Multisim Electronics Workbench необхідні логічні елементи і зібрати схему для реалізації заданою в таблиці 1.5 логічній функції y з трьома аргументами a, b і c . Скопіювати зібрану логічну схему на сторінку звіту.

Таблиця 1.5

Варіант	Логічна функція
1, 6, 11, 16, 21, 26	$y = (\bar{a}b + \bar{c})(\bar{a} + \bar{b} + c)(a + b + c).$
2, 7, 12, 17, 22, 27	$y = (a + b + \bar{c})(\bar{a} + \bar{b}c)(a + \bar{b} + \bar{c}).$
3, 8, 13, 18, 23, 28	$y = (b + a\bar{c})(\bar{a} + bc)(a + \bar{b} + c).$
4, 9, 14, 19, 24, 29	$y = (\bar{a}\bar{b} + \bar{c})(a + \bar{b} + c)(ab + \bar{c}).$
5, 10, 15, 20, 25, 30	$y = (a + \bar{b}c)(\bar{a} + b + \bar{c})(ab + c).$

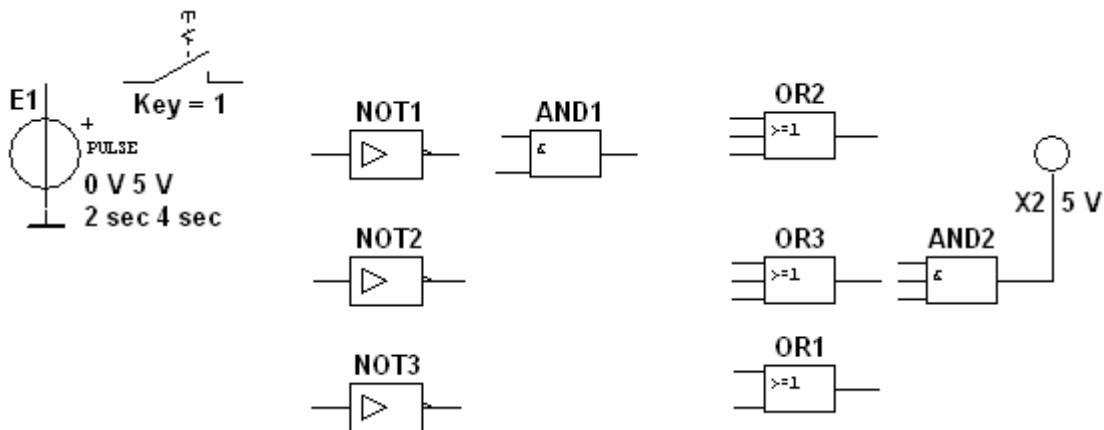
Для прикладу зберемо схему для реалізації логічної функції

$$y = (ab + \bar{c})(\bar{a} + \bar{b} + c)(a + b + c).$$

Аналіз функції показує, що для побудови логічної схеми нам знадобляться три інвертори, три диз'юнктора, причому один диз'юнктор з двома, а два – з трьома входами і два кон'юнктора, причому один з двома, а інший з трьома входами.

"Перетягнемо" на робоче поле середовища Multisim Electronics Workbench необхідні моделі логічних елементів з бібліотеки **Misc Digital\TTL** розташувавши їх, починаючи з входу, а саме:

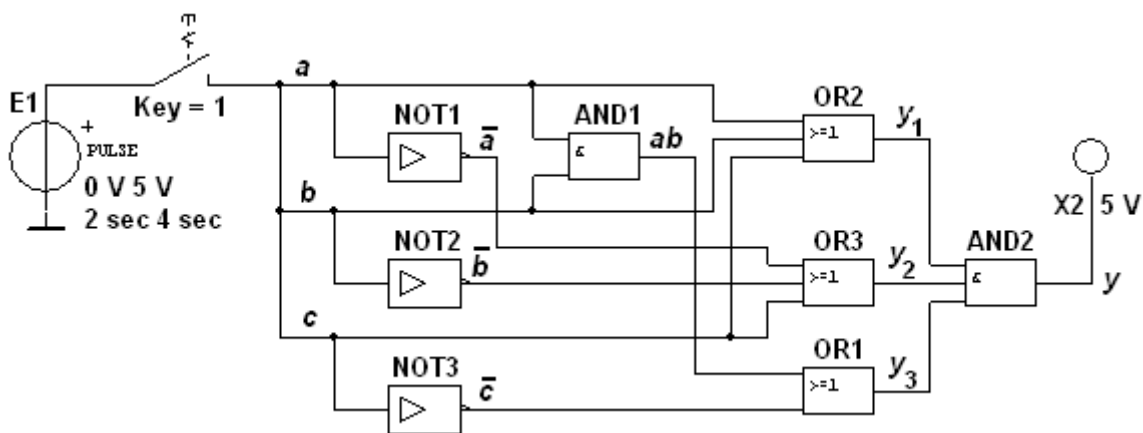
- три інвертора **NOT1**, **NOT2** і **NOT3**) для отримання інверсій \bar{a} , \bar{b} і \bar{c} аргументів a , b і c ;
- кон'юнктор **AND1** з двома входами для реалізації функції ab ;
- три диз'юнктора: **OR2** для реалізації функції $y_1 = a + b + c$, **OR3** для реалізації функції $y_2 = \bar{a} + \bar{b} + c$ і **OR1**, що реалізовує функцію $y_3 = ab + \bar{c}$, розмістивши їх друг під другом (мал. 1.3).



Мал. 1.3

Для виконання функції логічного множення $y = y_1 y_2 y_3$ додамо в схему кон'юнктор **AND2** з трьома входами, до виходу якого підключимо логічного пробника **X2** для сигналізації появи логічної одиниці на виході схеми. "Перетягнемо" з відповідних бібліотек на робоче поле джерело прямокутних сигналів **E1** і ключ **1**.

З'єднавши "провідниками" входи і виходи елементів у відповідності з логічними виразами складових заданої функції і записавши у звіті очікувані результати виконання операцій на виходах елементів (мал. 1.4), приступимо до моделювання.



Мал. 1.4

З цією метою спочатку клацнемо мишею на кнопці **Run/Stop**, потім натиснемо клавішу, що керує ключем, з цифрою **1**. Якщо з'єднання елементів

виконані правильно, то пробник **X2** «засвітиться». При розмиканні ключа **1** пробник «гасне».

Примітки. Основним вимірювальним приладом для перевірки цифрових електронних схем є логічний пробник. Після подвійного кліка мишею на його зображенні у вікні, що відкрилося, треба задати рівень напруги, наприклад, 5 В (мал 1.4) при якому він світиться.

*Таблиці істинності для бібліотечних логічних елементів можна викликати натисненням клавіші **F1** після виділення на схемі відповідного елементу.*

ЗМІСТ ЗВІТУ

1. Найменування і мета роботи.
2. Перелік приладів, використаних в експериментах з їх характеристиками.
3. Зображення електричної схеми для дослідження логічних елементів і зібраної схеми для реалізації заданої логічної функції.
4. Таблиці істинності, що відображають роботу досліджуваних логічних елементів.
5. Висновки по роботі.

Лабораторна робота №2

ТРИГЕРИ

МЕТА РОБОТИ:

Ознайомлення з основними характеристиками і дослідження інтегральних тригерів RS , D , T і JK .

ТЕОРЕТИЧНІ ВІДОМОСТІ ТА РОЗРАХУНКОВІ ФОРМУЛИ

Тригер - це електронний пристрій з двома стійкими станами рівноваги, призначений для запису і зберігання інформації. Під дією вхідних сигналів тригер може перемикатися з одного стійкого стани в інший. При цьому напруга на його виході стрибкоподібно змінюється з низького рівня на високий або навпаки.

За способом запису інформації тригери ділять на *асинхронні*, які перемикаються у момент подання вхідного сигналу, і *синхронні* (тактовані), які перемикаються тільки при поданні синхронізуючих імпульсів, а момент перемикання пов'язаний з певним рівнем синхросигналу (*статичні*) чи з моментом перепаду напруги на тактованому вході (*динамічні* тригери).

Як правило, тригер має два виходи: прямий Q і інверсний \bar{Q} . Число входів залежить від структури і функцій, що виконуються тригером. Наприклад, асинхронні RS -тригери мають два входи: вхід S для установки в стан логічної *одиниці* прямого виходу Q і вхід R що встановлює стан виходу Q у логічний *нуль*. Синхронні тригери для занесення в них інформації, окрім інформаційних входів S (J) і R (K) мають синхронізуючий C чи рахунковий T вхід, а тригери затримки – інформаційний вхід D .

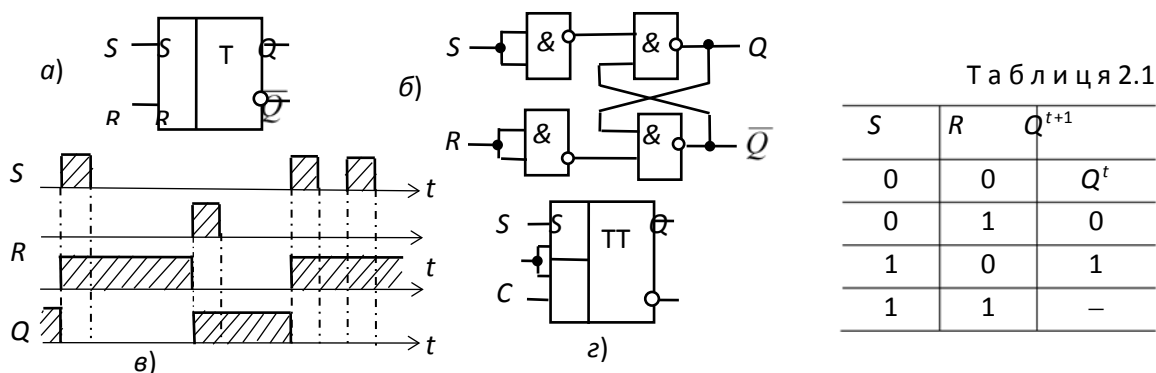
Найбільше поширення в цифрових пристроях отримали тригери RS , D , T і JK .

1. АСИНХРОННИЙ І СИНХРОННИЙ RS -ТРИГЕРИ

Простим тригером є *асинхронний* RS -тригер, умовне графічне зображення якого представлено на мал. 2.1, а принцип його роботи пояснюється таблицею істинності (таблиця 2.1). Тригер має два інформаційних входу: R і S і два виходи: Q і \bar{Q} . Незалежним є один вихід Q , оскільки інверсний сигнал \bar{Q} можна отримати з допомогою зовнішнього інвертора.

Розглянемо таблицю 2.1. Позначимо Q^t сигнал на виході тригера до вступу сигналу 1 на його вхід S . При поданні сигналів $S = 1$ і $R = 0$ тригер переходить в стан $Q^{t+1} = 1$. При отриманні сигналів $R = 1$ і $S = 0$ на виході встановлюється $Q^{t+1} = 0$. За відсутності нових команд стан тригера не змінюється: тригер зберігає інформацію про останню з команд, що поступила. Природно, що комбінація

сигналів $S = 1$ і $R = 1$ відноситься до заборонених, оскільки при її поданні на входи тригера на його виході Q^{t+1} встановлюється або 1, або 0.



Мал. 2.1

На підставі таблиці. 2.1 запишемо аналітичний вираз функціонування RS -тригера:

$$Q^{t+1} = S + Q^t \bar{R}.$$

На мал. 2.1, *в* зображена часова діаграма, що ілюструє його роботу. В момент, коли подається сигнал $S = 1$, тригер переходить в стан $Q = 1$. За відсутності вхідних сигналів стан тригера не змінюється, а у момент подання сигналу $R = 1$ тригер перемикається в стан $Q = 0$, в якому перебуває до нового сигналу на S -вхід.

RS -тригер може бути побудований на різних логічних елементах. На мал. 31.1, *б* показана схема реалізації RS -тригера на базових елементах І-НЕ, в якій використаний позитивний зворотний зв'язок з виходів тригера на входи логічних елементів.

Асинхронний RS -тригер можна перетворити в *синхронний* якщо додати третій синхронізуючий вхід C (мал. 2.1, *з*) сполучений, наприклад, з нижніми, задалегідь розділеними, входами двох лівих елементів І-НЕ (див. мал. 2.1, *б*).

Вхід C забезпечує функціонування RS -тригера згідно із законом

$$Q^{t+1} = Q^t (\bar{C} + \bar{R}) + CS.$$

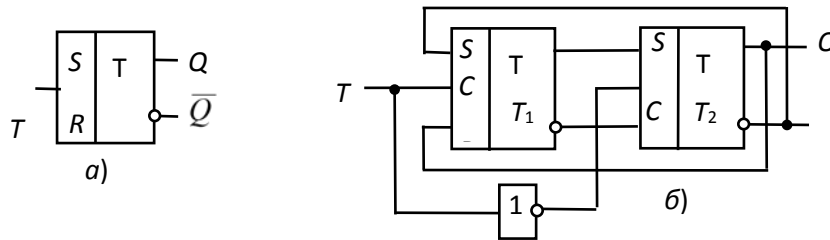
Перемикання синхронного RS -тригера в стан $Q = 1$ відбувається при $S = 1$ (або в стан $Q = 0$ при $R = 1$) у момент приходу синхроімпульса C . При $C = 0$ інформація з S - і R -входів на тригер не передається.

2. T-ТРИГЕР

Тригер з рахунковим запуском повинен перемикатися кожним імпульсом, що подається на єдиний рахунковий вхід T (мал. 2.2, *а*). Функціонування T -тригера визначається рівнянням

$$Q^{t+1} = Q^t \bar{T} + \bar{Q}^t T.$$

Він може бути реалізований, наприклад, на базі двох синхронних *RS*-тригерів (мал. 2.2, б). З появою фронту тактового імпульсу тригер T_1 перемикається в стан, протилежний до стану тригера T_2 . Але це не викликає сигналів на виходах Q і \bar{Q} оскільки за рахунок інвертора на тактовий вхід C тригера T_2 в даний момент поданий логічний 0. Тільки на зрізі імпульсу на вході T_1 перемкнеться тригер T_2 і станеться зміна сигналів на виходах Q і \bar{Q} а також на S - і R -входах першого ступеня.



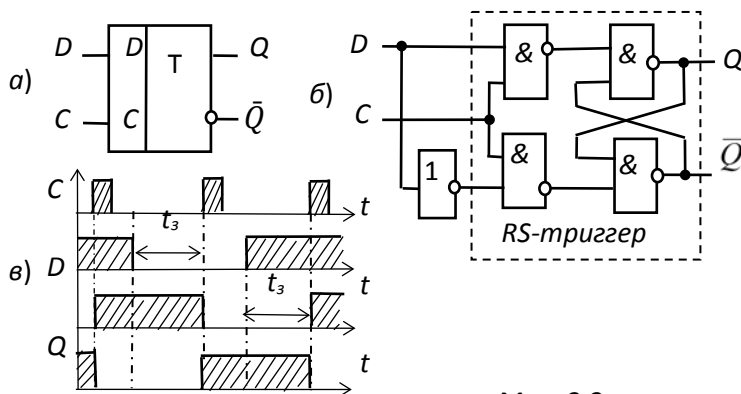
Мал 2.2

3. D-ТРИГЕР

Тригер затримки (*D-тригер*) може бути тільки синхронним, оскільки має один інформаційний D -вхід, інформація з якого переписується на вихід тригера тільки за тактовим сигналом, що подається на C -вхід. Умовне зображення D -тригера наведено на мал. 2.3, а. Реалізувати його можна на різних логічних елементах, у тому числі, на основі синхронного RS -тригера, доповненого інвертором (мал. 2.3, б). З аналізу таблиці. 2.2 функції перемикача D -тригера

$$Q^{t+1} = \bar{C}^t Q^t + C^t D^t$$

витікає, що за відсутності синхроімпульса ($C = 0$) стан тригера незмінний. За умови ж $C = 1$ тригер передає на вихід сигнал, що поступив на його вхід D у попередньому такті, вихідний сигнал Q^{t+1} змінюється з *затримкою* на один період імпульсів синхронізації.



Мал. 2.3

Таблиця 2.2

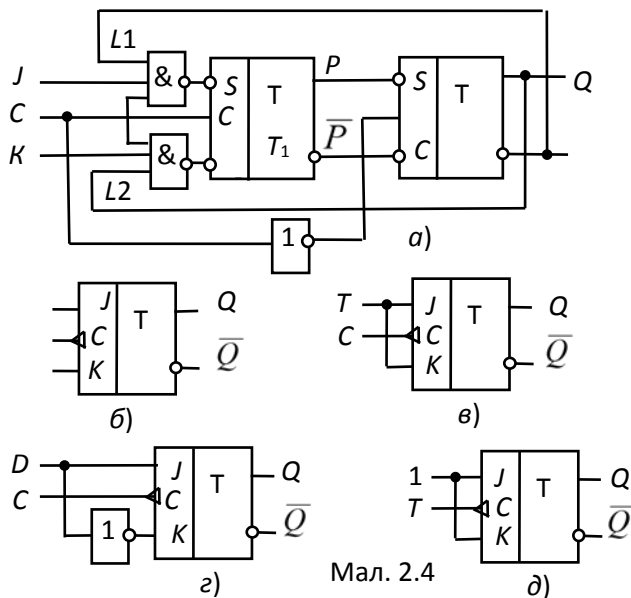
C^t	D^t	Q^t	Q^{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

З аналізу часової діаграми D -тригера (мал. 2.3, у) також витікає, що вихідний сигнал Q тригера повторює стан D -входа зі вступом чергового тактового імпульсу на вхід C із затримкою t_3 відносно логічного стану на D -вході.

4. JK-ТРИГЕР

JK -тригери виконують тактованими. JK -тригер має інформаційні входи J і K які по своїй дії на пристрій аналогічні входам S і R синхронного RS -тригера: при $J = 1$ і $K = 0$ тригер по тактовому імпульсу C встановлюється в стан $Q = 1$; при $J = 0$ і $K = 1$ – перемикається в стан $Q = 0$, а при $J = 0$ і $K = 0$ – зберігає раніше прийняту інформацію.

На відміну від синхронного RS -тригера одночасна присутність логічних



Таблиця 2.3

J^t	K^t	Q^t	Q^{t+1}
0	0	0	0
1	0	0	1
0	1	0	0
1	1	0	1
0	0	1	1
1	0	1	1
0	1	1	0
1	1	1	0

одиниць на інформаційних входах не є для JK -тригера забороненою комбінацією; при $J = 1$ і $K = 1$ тригер працює в рахунковому режимі - перемикається кожним тактовим імпульсом на вході C .

На мал. 2.4, а зображена одна з функціональних схем JK -тригера. Вона відрізняється від схеми T -тригера (див. мал. 2.2, б) двома трьохвходовими елементами І-НЕ $L1$ і $L2$ першого ступеня JK -тригера. Вхід C – динамічний (мал. 2.4, б): перемикання JK -тригера відбувається у момент перепаду синхроімпульса з рівня $C = 1$ на рівень $C = 0$.

При $J = 0$ і $K = 0$ на виході елементів $L1$ і $L2$ встановлюються логічні одиниці, які для тригерів з інверсними входами є пасивними сигналами: тригер T_1 і, отже, JK -тригер в цілому зберігають колишній стан (див. мал. 2.4, а). Логічна 1 на одному з входів елемента І-НЕ не визначає 1 на його виході і комбінація $J = 1, K$

= 1 ніяк не впливає на вхідну логіку першого ступеня, тому схеми T - і JK -тригерів (див. мал. 2.2, б і мал. 2.4, а) принципово не відрізняються: обое працюють у рахунковому режимі.

Тільки при комбінації сигналів $J = 1, K = 1$ і $\bar{Q} = 1$ на вході елементу $L1$ тригер T_1 перемкнеться в стан $P = 1$. Аналогічно логічний 0 буде на виході елементу $L2$, коли $K = 1, C = 1$ і $Q = 1$.

Таким чином, комбінація $J = 1, K = 0$ обумовлює по тактовому імпульсу $C = 1$ перемикач JK -тригера в цілому в стан $Q = 1$, а комбінація $J = 0, K = 1$ – у стан $Q = 0$.

З аналізу таблиці 2.3 функції перемикача JK -тригера

$$Q^{t+1} = \bar{K}^t Q^t + J^t \bar{Q}^t$$

витає, що стан тригера визначається не лише рівнями сигналів на інформаційних входах J і K , але і станом Q^t , у якому раніше знаходився JK -тригер. Так, при комбінації $J = 0, K = 0$ тригер зберігає попередній стан ($Q^{t+1} = Q^t$); комбінація $J = 1, K = 1$ призводить до того, що тактовим імпульсом тригер перемикається в стан, протилежний до попереднього: $Q^{t+1} = \bar{Q}^t$. Комбінації $J = 1, K = 0$ і $J = 0, K = 1$ дають дозвіл тригеру перемкнутися відповідно в стан $Q = 1$ і $Q = 0$.

На основі JK -тригера (мал. 2.4, б) можуть бути виконані синхронний (мал. 2.4, в) і асинхронний (мал. 2.4, з) T -тригери, D -тригер (мал. 2.4, д) і синхронний RS -тригер (мал. 2.4, е).

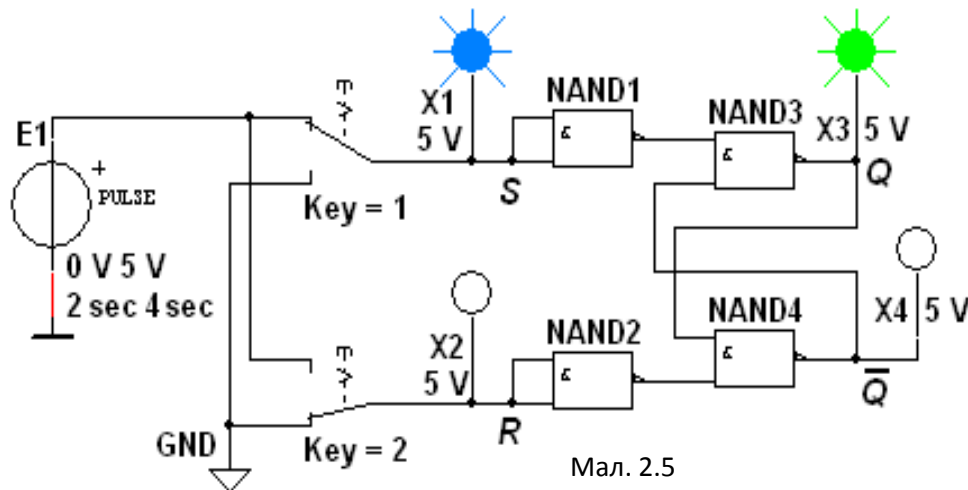
При проектуванні складних логічних схем потрібні тригери різних типів, які можна було б виконати на основі одного універсального тригера і використати його в різних режимах роботи і модифікаціях. У інтегральній схемотехніці найбільше поширення отримали D - і JK -тригери.

ЗАВДАННЯ І МЕТОДИЧНІ ВКАЗІВКИ ДО ЇХ ВИКОНАННЯ

Завдання 1. Відкрити файл або зібрати на робочому полі середовища Multisim WorkBench схему для дослідження *асинхронного RS-тригера* (мал. 2.5) і встановити у діалогових вікнах компонентів їх параметри або режими роботи. Скопіювати схему (мал. 2.5) на сторінку звіту.

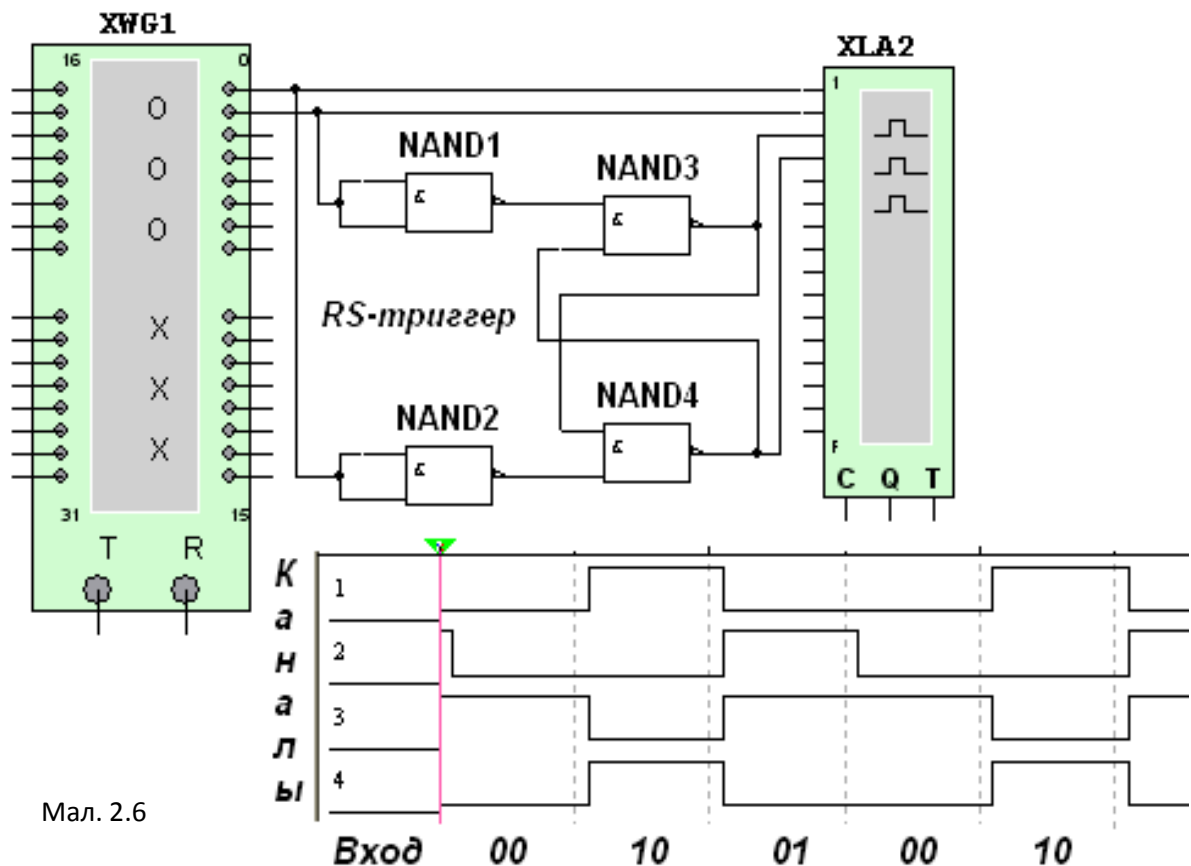
Схема (мал. 2.5) зібрана на чотирьох логічних елементах І-НЕ (**NAND**). На входи S і R елементів **NAND1** і **NAND2** через ключі **1** і **2** подаються логічні сигнали 1 чи 0 від джерела прямокутних імпульсів **E1** з амплітудою 5 В. До виходів Q і \bar{Q} елементів **NAND3** і **NAND4**; до виходів тригера, як і до його входів S і R , підключені пробники **X1, X2, X3** і **X4** з пороговою напругою 5 В.

Скориставшись порядком засвічування різноколірних пробників і задаючи коди (00, 01, 10) ключів **1** і **2** (вхідних сигналів), **скласти** таблицю істинності *RS*-тригера. Наприклад, сформувавши з допомогою ключів сигнали $S = 1$ і $R = 0$ і подавши їх на вхід тригера, отримаєте на його виході сигнали $Q = 1$ і $\bar{Q} = 0$ (див. мал. 2.5). Переконайтеся, що при забороненому коді «1,1» вхідних сигналів, на виході *RS*-тригера можуть засвітитися обидва пробника, або обидва не світяться.



Завдання 2. Підключити до входів тригера логічний генератор (генератор слова) **XWG1** (мал. 2.6), запрограмувавши його перші три осередки кодами 00, 10 і 01 і з'єднавши входи і виходи тригера з входами логічного аналізатора **XLA2**.

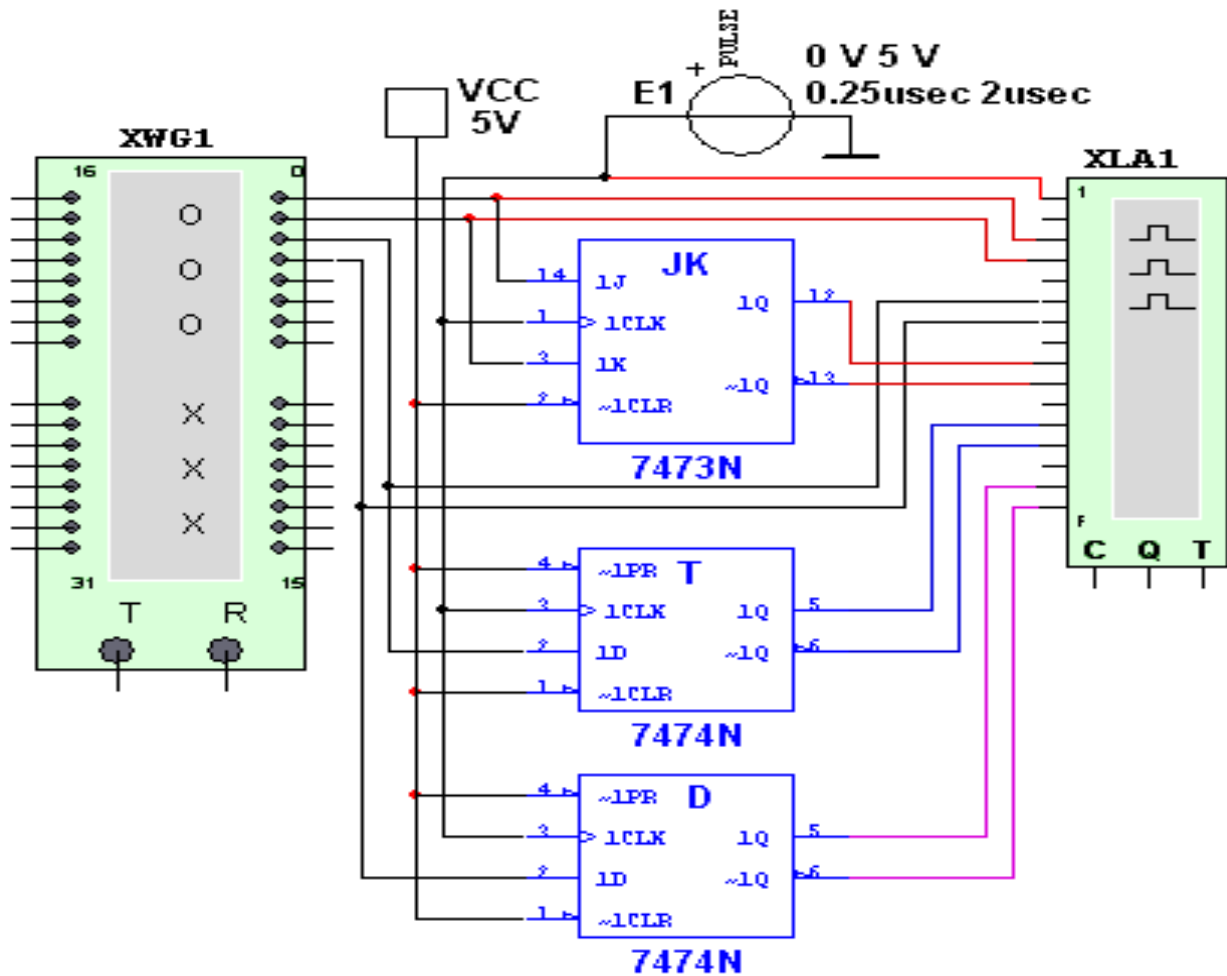
У діалоговому вікні генератора слова **XWG1** встановити частоту $f_2 = 10$ кГц і два цикли моделювання сигналів (в режимі **Burst**), а у вікні аналізатора **XLA2**



– частоту $f_a = 0,1$ МГц таймера, рівень високої напруги $U_m = 5$ В, число імпульсів $\text{Clocks/div} = 8$ таймера на одну поділку.

Отримати на екрані аналізатора **XLA2** часову діаграму станів *RS*-тригера (див. мал. 2.6, внизу). **Скопіювати** схему дослідження і часову діаграму стану *RS*-тригера на сторінку звіту.

Завдання 3. **Відкрити** файл або **зібрати** на робочому полі середовища EWB схему для дослідження *тригерів* **JK**, **T** і **D** (мал. 2.7) і **встановити** у діалогових вікнах компонентів їх параметри або режими роботи. **Скопіювати** схему (мал.



Мал. 2.7

2.7) на сторінку звіту.

У схему (мал. 2.7) включені: генератор **XWG1** (частота $f_2 = 500$ кГц); логічний аналізатор **XLA1**; тригери у інтегральному виконанні: універсальний **JK**, рахунковий **T** і тригер затримки **D**.

На $\overline{1CLR}$ - і $\overline{1PR}$ -входи тригерів подається постійна напруга 5 В (що імітує сигнал 1) джерела **VCC**, а на $1C$ -входи тригерів і на вхід 20 аналізатора **XLA1** поступають тактові імпульси з амплітудою 5 В і частотою 500 кГц, сформовані генератором **E1**.

З виходів 1 і 2 генератори **XWG1** сигнали подаються на входи, що керують **1J** і **1K** *JK*-тригера, з виходу 3 – на вхід **1D** *T*-тригера, а з виходу 4 – на вхід **1D** *D*-тригера.

Для формування вихідних сигналів генератор **XWG1** треба запрограмувати - ввести в елементи пам'яті кодові комбінації з одиниць і нулів згідно з варіантом (таблиця. 2.4).

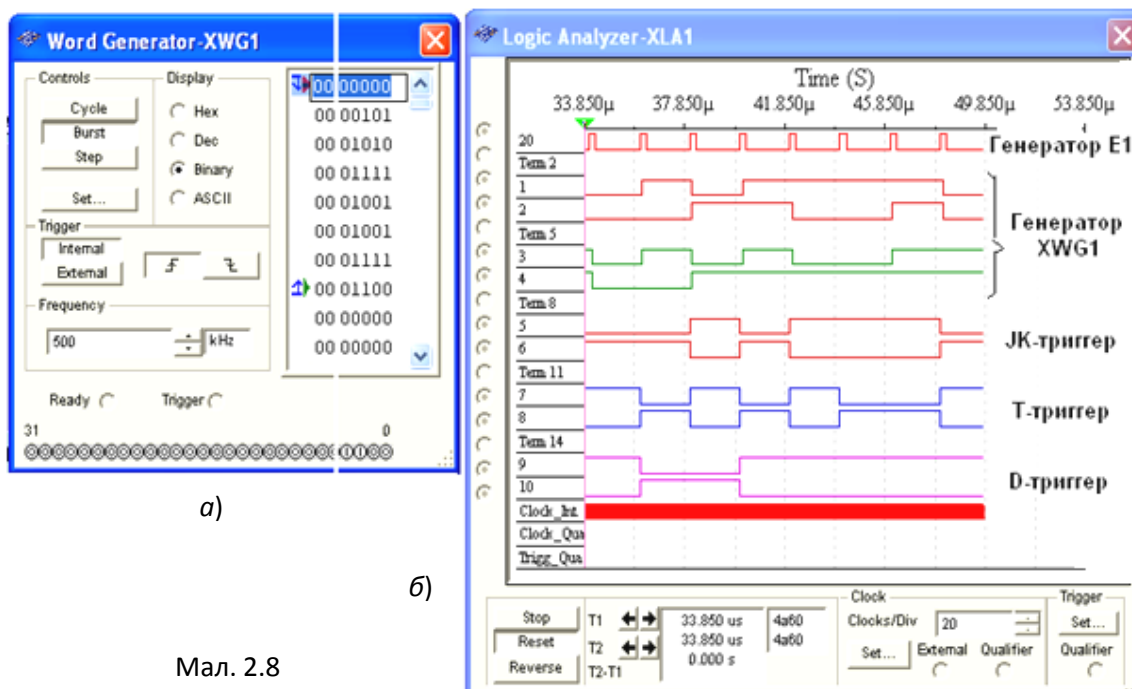
У якості прикладу введемо в перші вісім елементів пам'яті генератора чотирирозрядні кодові комбінації (мал. 2.8, *a*):

0000, 0101, 1010, 1111, 1001, 1001, 1111, 1100.

При моделюванні генератор послідовно і циклічно виводить у кожному комірці пам'яті (від початкової до кінцевої) на виходи 1, 2, 3 і 4, формуючи на них наступні коди сигналів: 01011110, 00110010, 01010011 і 00111111 (див. сигнали на каналах 1, 2, 3 і 4 логічний аналізатора **XLA1** (мал. 2.8, *б*)). Перед моделюванням **виділіть** у вікні генератора **XWG1** осередок з адресою 0 початку рахунку і виведення сигналів.

Т а б л и ц я 2.4

Варіант	Вміст елементів пам'яті генератора слова XWG1
1, 6, 11, 16, 21, 26	0000, 1010, 1111, 1001, 1001, 1101, 1100, 0000
2, 7, 12, 17, 22, 27	0000, 1100, 1010, 1011, 1001, 1111, 1110, 0000
3, 8, 13, 18, 23, 28	0000, 1010, 1011, 1001, 1001, 1111, 1101, 0000
4, 9, 14, 19, 24, 29	0000, 1111, 1101, 1001, 1011, 1011, 1100, 0000
5, 10, 15, 20, 25, 30	0000, 1011, 1101, 1001, 1100, 1111, 1010, 0000



a)

б)

Мал. 2.8

Провести моделювання роботи тригерів в режимах **Step** чи **Burst** генератора **XWG1**, **скопювати** у звіт *часові* діаграми, **скласти** і **заповнити** *таблиці істинності* роботи тригерів **JK**, **T** і **D** при заданому у таблиці 2.4 варіанті вхідних кодових комбінацій. Зокрема, **описати** стани *JK*-тригера з приходом тактового сигналу $C = 1$, коли сигнали $J = 1$ і $K = 1$, а $Q = 0$ або $Q = 1$.

Примітка. *Таблиці істинності для розглянутих бібліотечних тригерів можна викликати натисненням клавіші допомоги **F1** після виділення тригера на схемі.*

ЗМІСТ ЗВІТУ

1. Найменування і мета роботи.
2. Перелік приладів, використаних в експериментах з їх характеристиками.
3. Зображення електричних схем для дослідження тригерів *RS*, *JK*, *T* і *D* за допомогою логічних пробників і логічного аналізатора **XLA1**.
4. Копії часових діаграм і таблиці істинності, що відображають роботу тригерів.
5. Висновки по роботі.

Лабораторна робота №3

ЛІЧИЛЬНИКИ

МЕТА РОБОТИ:

Ознайомлення з будовою і функціонуванням лічильників, дослідження *синхронного суматора, реверсивного і десяткового лічильників*.

ТЕОРЕТИЧНІ ВІДОМОСТІ ТА РОЗРАХУНКОВІ ФОРМУЛИ

1. КЛАСИФІКАЦІЯ ЛІЧИЛЬНИКІВ

Лічильник призначений для рахунку імпульсів, що поступають на його вхід, в інтервалі між якими він повинен зберігати інформацію про їх кількість. Тому лічильник складається з запам'ятовуючих комірок – тригерів *D*- або *JK*-типа. Між собою комірки лічильника сполучають так, щоб кожному числу імпульсів відповідали стани 1 або 0 певних комірок. При цьому сукупність одиниць і нулів на виходах *n* комірок, що називаються *розрядами* лічильника, є *n*-розрядне двійкове число, яке однозначно визначає кількість імпульсів, що пройшли через входи.

Кожен розряд лічильника може знаходитися в двох станах. Число стійких станів, яке може приймати цей лічильник, називають *коефіцієнтом перерахунку Крах*.

Якщо з кожним вхідним імпульсом записане в лічильнику число збільшується, то такий лічильник являється *суматором*, якщо ж воно зменшується, то – *різницеvim лічильником*. Лічильник, який може працювати як на додавання, так і на віднімання, називають *реверсивним*.

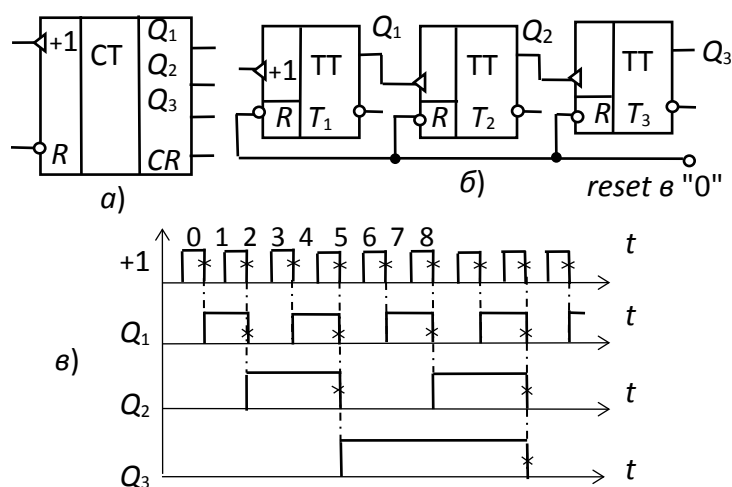
Лічильники, у яких під впливом вхідного імпульсу перемикання відповідних розрядів відбувається послідовно один за одним, називають *асинхронними*, а коли перемикання відбувається одночасно – *синхронними*. Максимальне число *N* яке може бути записане в лічильнику, рівне $(2^n - 1)$, де *n* - число розрядів лічильника.

По способу кодування послідовних станів розрізняють *двійкові лічильники* з коефіцієнтами перерахунку (обнулення) $Kрах=2^n$ у яких порядок зміни станів тригерів відповідає послідовності двійкових чисел, і *недвійкові*, у яких $Kрах < 2^n$ (наприклад, десяткові з коефіцієнтом $Kрах=10$ або дільники частоти з коефіцієнтом ділення $Kрах \neq 2^n$).

Лічильники входять до складу різноманітних цифрових пристроїв: електронного годинника, дільників частоти, розподільників імпульсів, обчислювальних пристроїв. Інтегральні лічильники, що випускаються промисловістю, є схемами середньої інтеграції (наприклад, мікросхеми серій К155, К176, К561 та ін.); серед них багаторозрядні бінарні лічильники на додавання і реверсивні лічильники з входами *R* і *S* для усіх розрядів, з постійними і довільними коефіцієнтами перерахунку.

2. ЛІЧИЛЬНИК З БЕЗПОСЕРЕДНІМИ ЗВ'ЯЗКАМИ

Умовне зображення трьохрозрядного суматора показано на мал. 3.1, а на якому символом R позначений вхід загального скидання, символами Q_1 , Q_2 і Q_3 - виходи лічильника, CR - вихід перенесення одиниці. Вхід додавального входу позначається $+1$, віднімального -1 . Це рахункові входи. У асинхронних лічильників ці входи помічені спеціальними символами: $>$ чи $<$ що вказують полярність перепаду вхідного сигналу: 1/0 або 0/1, при якій відбувається перемикання тригерів лічильника.



Т а б л и ц я 3.1

Номер вхідного імпульса	Q_3	Q_2	Q_1	CR
0	0	0	0	
1	0	0	1	
2	0	1	0	
3	0	1	1	
4	1	0	0	
5	1	0	1	1
6	1	1	0	

Мал. 3.1

7 1 1 1

Для перемикання тригерів в лічильниках використовують наступні зв'язки: безпосередню, тракт послідовного перенесення, тракт паралельного перенесення. Схема лічильника з безпосередніми зв'язками показана на мал. 3.1, б. Перший тригер лічильника T_1 утворює молодший розряд. Він рахує вхідні імпульси по модулю 2, а стан його виходу сприймається наступним T_2 тригером як вхідні сигнали і знову перераховуються на 2 і т. д.

Повне уявлення про стани лічильника (мал. 3.1, б) залежно від числа поданих на вхід імпульсів, дає таблиця 3.1 перемикача і часові діаграми (мал. 3.1, в) де зображені послідовність вхідних імпульсів (на вході $+1$), а також стану тригерів - першого (Q_1), другого (Q_2) і третього (Q_3). Фронти імпульсів на діаграмах показані ідеальними: потенціал, що відповідає логічному 0, вважається рівним нулю, перемикальні перепади для наочності помічені хрестиками.

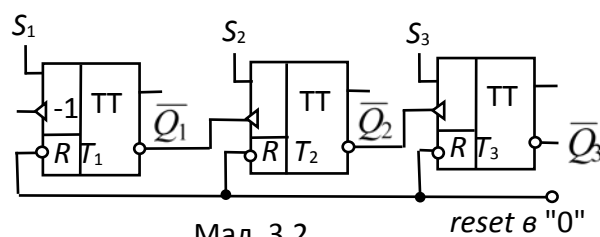
Розглянемо дію на лічильник шостого (позначеного на діаграмі цифрою 5) імпульсу. По його спаду тригер T_1 встановлюється в 0, перепад 1/0 на його виході Q_1 перемикає в 1 тригер T_2 , а тригер T_3 залишається в колишньому (одичному) стані, оскільки перепад 0/1 на виході Q_2 не є для нього перемикальним.

З діаграми видно, що частота імпульсів на виході кожного тригера удвічі менша частоти імпульсів на його вході. У момент, передуючий перемиканню чергового розряду, усі попередні розряди лічильника знаходяться в стані 1.

Восьмий імпульс для трьохрозрядного лічильника (таблиця 3.1) є імпульсом переповнення: усі тригери встановлюються в 0 (лічильник "обнуляється").

Якщо в лічильнику використовуються тригери, що перемикаються перепадом 0/1, то вхід подальшого тригера треба з'єднати з інверсним виходом попереднього, на якому формується цей перепад, коли по основному виходу тригер перемикається з 1 в 0.

Схема віднімаючого лічильника приведена на мал. 3.2, в якій по входах S у розряди лічильника заносять двійкове число, з якого треба відняти число, що представляється кількістю вхідних імпульсів. Нехай, наприклад, в лічильник (мал. 3.2) занесено число $5_{10} = 101_2$. Першим вхідним імпульсом тригер T_1 перемикається з 1 в 0 (по основному виходу); при цьому на інверсному виході \overline{Q}_1 виникає перепад 0/1, яким тригер T_2 перемикнути не може; у лічильнику залишиться число $100_2 = 4_{10}$.



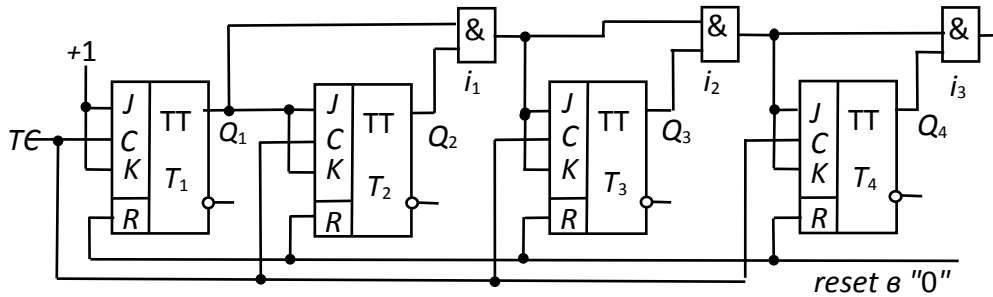
Мал. 3.2

Другий вхідний імпульс встановлює тригер T_1 у стана 1, на виході \overline{Q}_1 з'являється перепад 1/0, який перемикає T_2 у стана 1, а зформований при цьому на \overline{Q}_2 перепад 1/0 перемикає T_3 у стана 0. У лічильнику залишається число $011_2 = 3_{10}$. Аналогічно можна розглянути дію подальших вхідних імпульсів.

У лічильнику з безпосереднім зв'язком перемикання тригерів, що зумовлене перепадом вхідного сигналу, відбувається один за одним, *послідовно* і затримка поширення n -розрядного лічильника, у разі зміни усіх 1 на 0, – у n раз більше затримки одного T -триггера. Якщо розрядів багато, то велика затримка може виявитися серйозним недоліком такого лічильника. Через неможливість виконати зміну стану усього лічильника в єдиний момент часу, лічильники з безпосереднім зв'язком бувають тільки *асинхронними*.

3. СИНХРОННИЙ СУМАТОР

У *синхронному* лічильнику розряди переходять в новий стан одночасно (синхронно). Для того, щоб на входи усіх розрядів кожен рахунковий імпульс постував одночасно, а перемикання розрядів відбувалося в потрібній послідовності, у схему додають логічні ланки, які забезпечують перемикання одних розрядів, а інші утримують від перемикання.



Мал. 3.3

У схемі (мал. 3.3) чотирирозрядного синхронного лічильника на JK -триггерах на тактові входи C усіх тригерів рахункові імпульси поступають одночасно з входу T . Інформаційні входи J і K кожного тригера об'єднані. Тригер T_1 перемикається кожним рахунковим імпульсом, оскільки на його входи J і K постійно подається 1. Інші тригери перемикаються рахунковими імпульсами при наступних умовах:

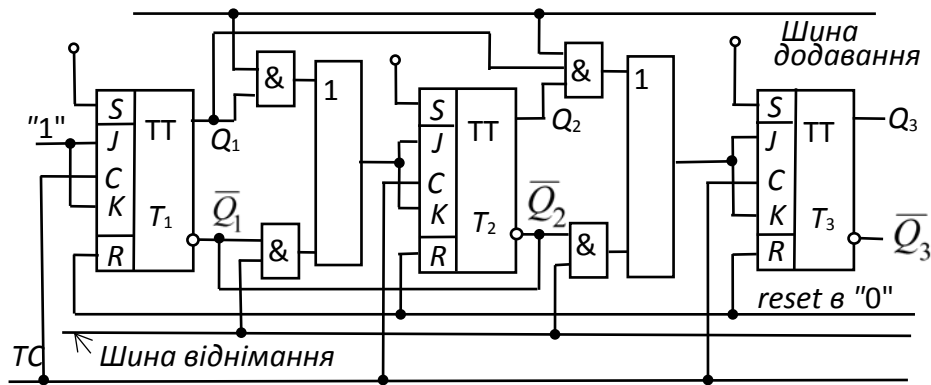
T_2 – при $Q_1 = 1$; T_3 – при $Q_1 = 1$ і $Q_2 = 1$; T_4 – при $Q_1 = 1$, $Q_2 = 1$ і $Q_3 = 1$.

Щоб забезпечити вказані умови перемикання тригерів, в схему (мал. 3.3) додані кон'юнктори i_1 , i_2 та i_3 . На інформаційний вхід кожного з тригерів T_2 , T_3 і T_4 подається кон'юнкція сигналів з основних виходів попередніх тригерів. Одиниця, що дозволяє перемикання, поступить на вхід відповідного тригера, якщо усі попередні тригери знаходяться в стані 1, і по рахунковому сигналу він перемикається.

4. РЕВЕРСИВНИЙ СИНХРОННИЙ ЛІЧИЛЬНИК

Реверсивний лічильник, фрагмент якого зображений на мал. 3.4, працює як на додавання, так і на віднімання. Для переходу від додавання до віднімання і назад змінюють підключення входу наступного тригера до виходів попереднього.

На об'єднані входи J і K кожного тригера подається через диз'юнктори кон'юнкція сигналів з виходів попередніх тригерів: основні виходи попередніх тригерів приєднуються через кон'юнктори верхнього ряду (при додаванні), а інверсні виведення – через кон'юнктори нижнього ряду (при відніманні). При складанні подають 1 на шину додавання, якою вводяться в дію кон'юнктори верхнього ряду; при цьому на шині віднімання є присутній 0, внаслідок чого кон'юнктори нижнього ряду вимкнені. Віднімання здійснюється при поданні 1 на шину віднімання і 0 на шину додавання. Рахункові імпульси поступають на вхід T .



Мал. 3.4

Як відзначалося раніше, кожен тригер перемикається по тактовому входу C при $J=K=1$, що має місце, коли на виходах усіх попередніх тригерів (на основних - при додаванні, на інверсних - при відніманні) будуть одиниці.

5. ДЕСЯТКОВИЙ ЛІЧИЛЬНИК

Найбільше поширення серед недвійкових лічильників, у яких коефіцієнт перерахунку $K_{рах} < 2^n$ мають *десяткові* лічильники, у яких $K_{рах} = 10$. При проектуванні недвійкового лічильника спочатку визначають кількість його розрядів n так, щоб 2^n було найбільшим до $K_{рах}$ числом. Потім тим або іншим способом (наприклад, примусовою установкою деяких розрядів лічильника в 1) виключають надмірні стани лічильника, число яких рівне $2^n - K_{рах}$.

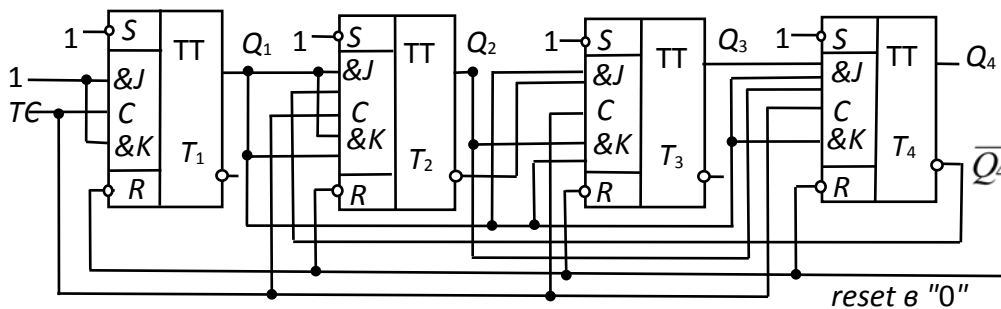
Так, для отримання $K_{рах}=10$ однорозрядний лічильник повинен містити чотири тригери, а надмірні стани $2^n - K_{рах} = 16 - 10 = 6$ виключають тим або іншим способом. При проектуванні десяткового лічильника частіше використовують двійково-десяткове кодування чисел. У цій системі, наприклад, число 375 записується як 0011 0111 0101, де збережені позиції десяткових розрядів: $0011_2=3_{10}$, $0111_2=7_{10}$, $0101_2=5_{10}$.

У зв'язки з цим десятковий лічильник повинен складатися з послідовно сполучених декад, інформація про кожного з дев'яти імпульсів накопичується в декаді, а десятим імпульсом вона обнуляється, і одиниця переноситься в наступну декаду. Кожна декада працює в натуральному двійковому коді починаючи з старшого відповідно рівними 8, 4, 2, 1, тобто, декада працює в коді 8-4-2-1. Якщо до виходів декад підключити індикатори, то вони показуватимуть записані числа в декадах в звичному десятковому коді.

На мал. 3.5 зображена функціональна схема десяткового лічильника з паралельним перенесенням на JK -триггерах зі вбудованими логічними елементами, що реалізовує функції перемикачів:

$$J_1 = 1; J_2 = Q_1 \bar{Q}_4; J_3 = Q_1 \bar{Q}_2; J_4 = Q_1 Q_2 Q_3;$$

$$K_1 = 1; K_2 = Q_1; K_3 = Q_1 Q_2; K_4 = Q_1.$$



Мал. 3.5

Розглянемо роботу схеми. Нехай по тактовому входу T на тригер T_1 поступило сім імпульсів і покази лічильника 0111. При цьому на входах K тригерів T_1 , T_2 і T_3 будуть логічні одиниці. Восьмий імпульс викличе перемикання усіх тригерів лічильника, тобто у нього буде записаний код 1000. Дев'ятий імпульс викличе перемикання тільки першого тригера, оскільки інші тригери заблоковані по входу J рівнями логічного нуля з тригерів T_1 , T_2 і T_3 відповідно. Показання лічильника буде 1001. Десятий вхідний імпульс викличе перемикання тригерів T_1 і T_4 оскільки тригери T_2 і T_3 заблоковані по входу J рівнями 0 з виходів відповідних тригерів. Лічильник зафіксує двійковий код 0000, тобто встановиться в початковий стан.

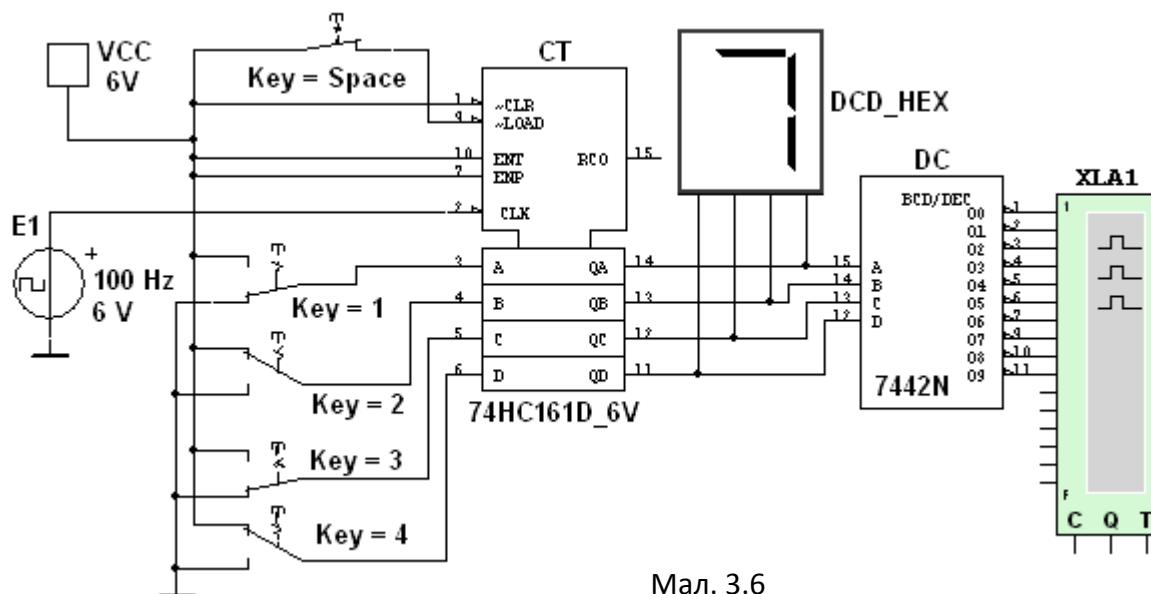
Зменшення числа стійких станів в лічильнику прямого рахунку досягнуто за рахунок введення зворотних зв'язків, за допомогою яких сигнал з старшого розряду поступає в молодші, забезпечуючи при цьому зміну послідовності двійкових чисел при підрахунку вхідних імпульсів. Цим способом можна будувати лічильники із заданим коефіцієнтом перерахунку.

ЗАВДАННЯ І МЕТОДИЧНІ ВКАЗІВКИ ДО ЇХ ВИКОНАННЯ

Завдання 1. Запустити комплекс Multisim Electronics Workbench. На робочому полі віртуальної лабораторії зібрати схему для дослідження синхронного двійкового лічильника (мал. 3.6) або відкрити відповідний файл і встановити у діалогових вікнах компонентів їх параметри або режими роботи. Скопіювати схему (мал. 3.6) на сторінку звіту.

У бібліотеці програмного середовища Multisim Electronics Workbench є 4-, 8- і 12-розрядні лічильники різних типів. Серед них: лічильники асинхронні (SN7493, SN74393) синхронні (74NC161, SN74163) реверсивні (SN74191) двійково-десяткові (SN7493, SN74160) та ін.

У схему (мал. 3.6) включений синхронний двійковий 4-розрядний лічильник 74HC161 до входу \overline{CLK} якого підключений джерело тактових імпульсів E1 а до виходів QA, QB, QC і QD – шістнадцятиричний 7-сегментний індикатор DCD_HEX і дешифратор DC 4x10. Вихід дешифратора сполучений з входами логічного аналізатора XLA1.



Мал. 3.6

До входів **A**, **B**, **C** і **D** лічильника **СТ** підключений джерело постійного напруги **VCC**, перемикачі **1-4** для формування вхідних двійкових кодів і ключ **Space** для зміни режиму роботи лічильника. У синхронному лічильнику задані з допомогою ключів рівні сигналів подаються на входи усіх тригерів, як і тактові імпульси, які подаються на рахункові входи **CLK** усіх розрядів лічильника.

При замкненому ключі **Space** число поданих від генератора **E1** на вхід лічильника імпульсів висвічується на індикаторі **DCD_HEX** у десятковому коді, від 0 до 15, після чого лічильник обнуляється і знову починається рахунок. При цьому на одному з виходів дешифратора **DC** формується сигнал низького рівня (логічний 0), номер якого відповідає коду вхідного числа: від 0000 до 1001 (9_{10}).

При розімкненому ключі **Space** сформоване за допомогою перемикачів на вході лічильника 4-разрядне двійкове число висвічується на індикаторі в десятковому коді, а на екрані аналізатора на одному з виходів, що відповідає вхідному коду лічильника, формується логічний 0.

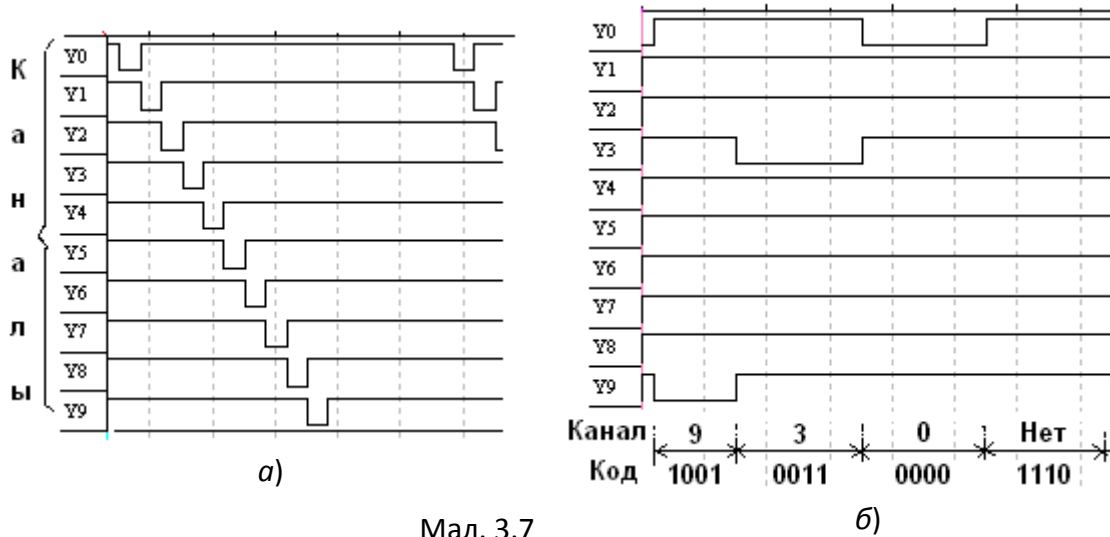
Завдання 2. Замкнути ключ **Space**, запустити програму моделювання суматора, і спостерігати за показами індикатора. Переконайтесь, що на екрані аналізатора **XLA1** логічні нулі перестають формуватися після приходу 11-го тактового імпульсу і з'являються знову тільки з приходом 17-го імпульсу (мал. 3.7, а).

Розімкнути ключ **Space**. Встановити у діалоговому вікні аналізатора **XLA1** напруга $V=5$ В частоту таймера $f_a=2$ кГц, число імпульсів, що припадають на одне ділення, **Clocks/div=60**. (При такому режимі промені повільно переміщуються на екрані аналізатора). За допомогою активних клавиш 1, 2, 3 і 4 клавіатури сформувати довільні (чи за вказівкою викладача) двійкові вхідні числа (коди), наприклад 1001, 0011, 0000, 1110 і подавати їх на входи **D**, **C**, **B** і **A** лічильника.

Переконайтеся, що при поданні числа 1110_2 (14_{10}) ні на одному виході дешифратора 4×10 не сформувався низький рівень сигналу (мал. 3.7, б).

Скопіювати на сторінку звіту результати моделювання синхронного сумуючого лічильника (див. мал. 3.7).

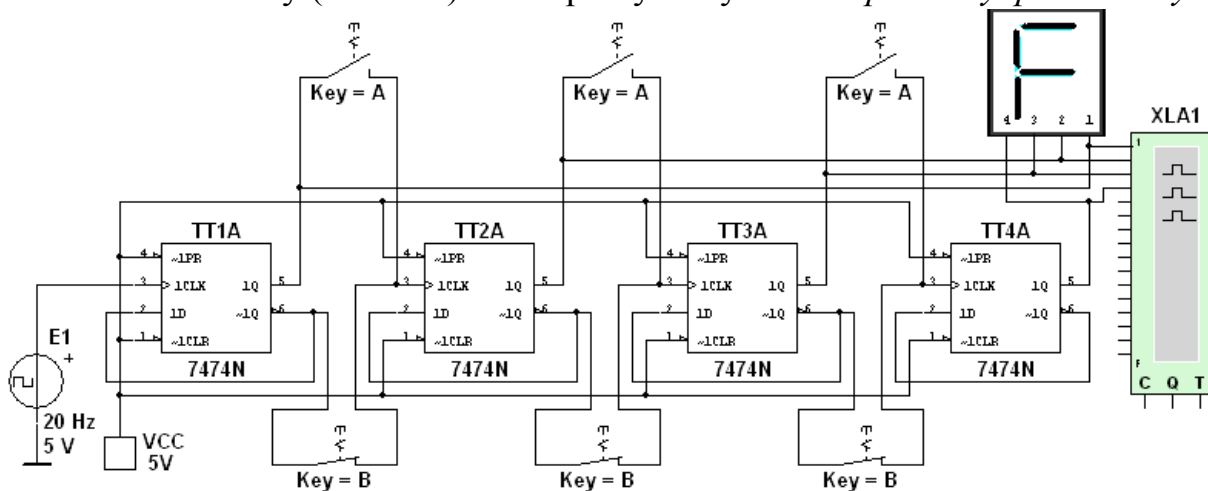
Завдання 3. Запустити комплекс Multisim Electronics Workbench. На робочому полі віртуальної лабораторії зібрати схему для дослідження реверсивного двійкового лічильника (мал. 3.8) або відкрити відповідний файл і



Мал. 3.7

б)

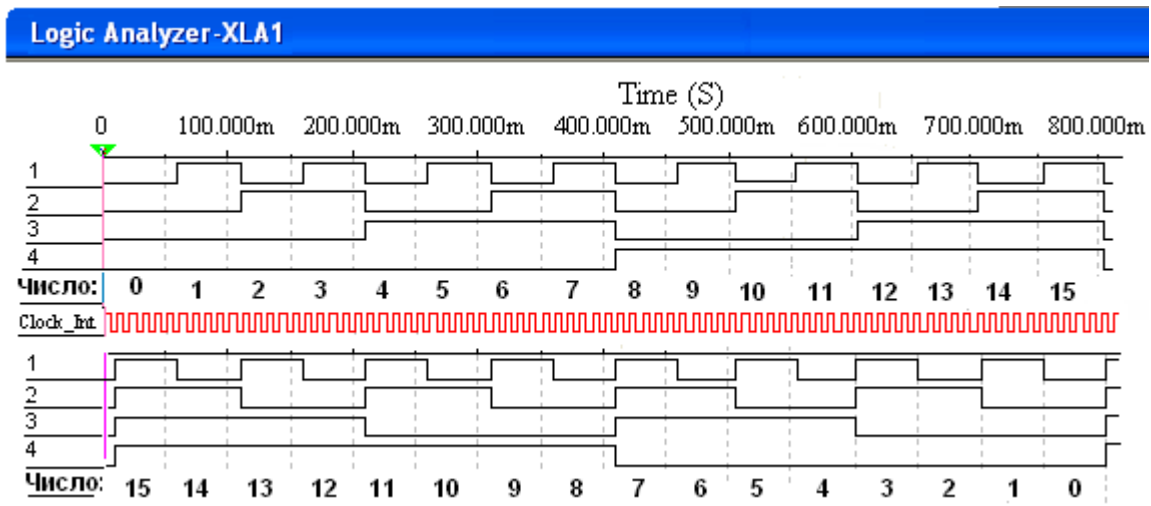
встановити у діалогових вікнах компонентів їх параметри або режими роботи. Скопіювати схему (мал. 3.8) на сторінку звіту в електронному файлі звіту.



Мал. 3.8

У схемі реверсивного двійкового лічильника (мал. 3.6) з допомогою груп ключів **A** і **B** здійснюється комутація попереднього тригера з входами $\overline{1PR}$ чи $\overline{1CLR}$ тригера наступного розряду, причому при замкнутих ключах **A** і розімкнених **B** (режим додавання) з кожним тактовим імпульсом збільшується результат рахунку, а при замкнутих ключах **B** і розімкнених **A** (режим віднімання) – результат рахунку зменшується.

Встановити у діалоговому вікні аналізатора **XLA1** напруга $V=5$ В частоту таймера $f_a=2$ кГц, число імпульсів, що припадають на одну подіку, **Clocks/div=60**. Розімкнути ключі **B** і замкнути ключі **A**. Запустити симуляцію роботи лічильника. При індикації числа 15 на 7-сегментному індикаторі клацнути мишею на кнопці **Stop** (зупинки моделювання) і скопіювати вікно аналізатора з результатами моделювання на сторінку звіту в *електронному файлі звіту*.



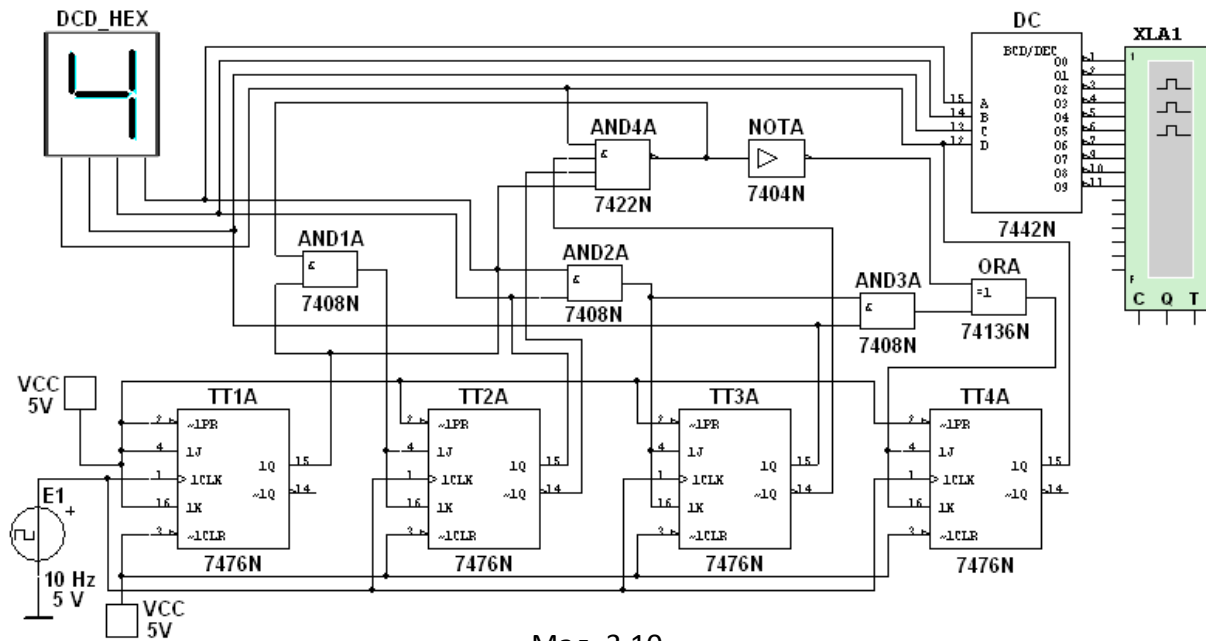
Мал. 3.9

Розімкнути ключі **A** і замкнути ключі **B**. Клацнути мишею на кнопке **Stop** (продовжити моделювання), зупинити моделювання при висвіченні числа 0 на індикаторі і скопіювати вікно аналізатора з результатами моделювання в звіт (див. нижню частину мал. 3.9).

Завдання 4. Запустити комплекс Multisim Electronics Workbench. На робочому полі віртуальної лабораторії зібрати схему для дослідження десятичного лічильника (мал. 3.10) або відкрити відповідний файл і встановити у діалогових вікнах компонентів їх параметри або режими роботи. Скопіювати схему (мал. 3.10) на сторінку звіту в *електронному файлі звіту*.

Результати моделювання 4-розрядного двійкового лічильника (мал. 3.8 і мал. 3.9) показали, що з його допомогою можна лічити до 15. Отже, для створення лічильника натуральних десятичних чисел в двійковому коді для однієї декади треба в схему двійкового 4-розрядного лічильника ввести зворотні зв'язки логічними елементами, за допомогою яких сигнал з якогось старшого розряду поступає в молодші і забезпечуючи вироблення лічильником двійкового еквіваленту рахункової декади.

У функціональній схемі десятичного лічильника (3.10), зібраній на *JK*-тригерах, на кожний одинадцятий тактовий імпульс результат рахунку скидається в нуль і далі результат рахунку збільшується. Повернення лічильника на одинадцятому тактовому імпульсу в початковий стан забезпечується додатковою комбінаційною схемою із вбудованими логічними елементами **I** (**AND**), **АБО** (**OR**), **НІ** (**NOT**).



Мал. 3.10

Лічильник працює так само, як синхронний двійковий лічильник до вступу сьомого імпульсу. Запустити програму моделювання десяткового лічильника і скопіювати вікно аналізатора з результатами моделювання на сторінку звіту в *електронному файлі звіту*.

ЗМІСТ ЗВІТУ

1. Найменування і мета роботи.
2. Перелік приладів, використаних в експериментах з їх краткими характеристиками.
3. Зображення електричних схем для випробування реверсивного і десяткового лічильників.
4. Копії часових діаграм, що відображають роботу досліджуваних лічильників.
5. Висновки по роботі.

Лабораторна робота №4

РЕГІСТРИ

МЕТА РОБОТИ :

Ознайомлення з будовою та функціонуванням регістрів і регістрової пам'яті; дослідження інтегрального універсального регістра зсуву.

ТЕОРЕТИЧНІ ВІДОМОСТІ І РОЗРАХУНКОВІ ФОРМУЛИ

1. КЛАСИФІКАЦІЯ РЕГІСТРІВ

Регістр – це пристрій для послідовного запису, зберігання і (або) зсуву інформації, яка поступає і зберігається в регістрі у вигляді n -розрядних двійкових чисел. У загальному випадку регістр може видавати інформацію в послідовній або паралельній формі, *перетворювати* прямий код числа в зворотний (коли одиниці замінюються нулями, а нулі – одиницями), і навпаки, а також виконувати логічне *додавання* і логічне *множення* двійкових чисел.

Залежно від способу введення і виведення розрядів числа розрізняють регістри паралельні, послідовні і паралельно-послідовні. У *паралельному* регістрі введення і виведення усіх розрядів кодового числа здійснюється одночасно, в *послідовному* – розряди числа вводяться і виводяться послідовно, а в *паралельно-послідовному* регістрі введення числа виконується в паралельній формі, а виведення – у послідовній, і навпаки. Перетворення паралельного коду в послідовний (і навпаки) - дуже актуальне завдання, оскільки передача цифрової інформації в мережах передачі даних здійснюється в послідовному коді, а обробка її в мікропроцесорах обчислювальних пристроїв - в паралельному.

Регістр, в якому можна здійснити зсув числа, називають *зсувним*, причому зсув може бути або в один бік (у бік молодшого розряду – *прямий зсув* (або *правий зсув*) чи у бік старшого розряду - *зворотний* (або *лівий*) зсув, або в обидві сторони (*реверсивний зсувний регістр*). У цьому сенсі послідовний і паралельно-послідовний регістри відносять до *зсувних*.

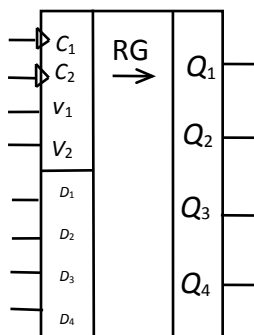


Рис. 4.1

Промисловість випускає багато типів регістрів у вигляді мікросхем. У якості прикладу на мал. 4.1 приведено зображення чотирирозрядного регістра (мікросхема серії К155). При $V_2 = 0$ розрядів числа вводять послідовно в регістр через вхід V_1 ; синхроімпульси, що поступають на вхід C_1 забезпечують зсув управо розрядів числа; мікросхема працює як *послідовний регістр зсуву*. У мікросхемі (див. мал. 4.1) передбачено також паралельне введення усіх розрядів числа по синхроімпульсу на вході C_2 з входів D_1, \dots, D_4 при $V_2 = 1$. У цьому випадку регістр працює як *паралельний регістр зсуву*.

Якщо виходи останнього тригера регістра зсуву з'єднати з входами першого, те вийде *кільцевий регістр зсуву*, що також називається *кільцевим лічильником*. Його коефіцієнт перерахунку дорівнює числу розрядів n : одиниця, записана в один з розрядів, періодично з'являється на виході лічильника після того, як пройдуть n синхроімпульсів.

2. ПАРАЛЕЛЬНИЙ РЕГІСТР НА RS-ТРИГЕРАХ

Будь-який реєстр складається з пов'язаних між собою тригерів з динамічним або статичним управлінням і логічних елементів, причому кількість тригерів дорівнює кількості розрядів в числі, яке можна записати у реєстр. Синтез реєстра зводиться до вибору типу триггерів і логічних елементів ІНЕ, АБО для реалізації заданих операцій.

Розглянемо роботу паралельного реєстра на RS-тригерах (мал. 4.2). Введення (запис) числа здійснюється в два такти. Щоб уникнути помилкового запису числа $x_1x_2\dots x_n$ у першому такті усі тригери реєстра обнуляються. Для цього на шину "0" подається логічний 0. В другому такті за сигналом I на шині "П" ("Прийом") через кон'юнктори одночасно записується у відповідні розряди реєстра двійкове число $x_1x_2\dots x_n$. Виведення (зчитування) числа $y_1y_2\dots y_n$ у прямому коді відбувається за сигналом I на шині $U_{пр}$ а в зворотному – за сигналом I на шині $U_{зв}$.

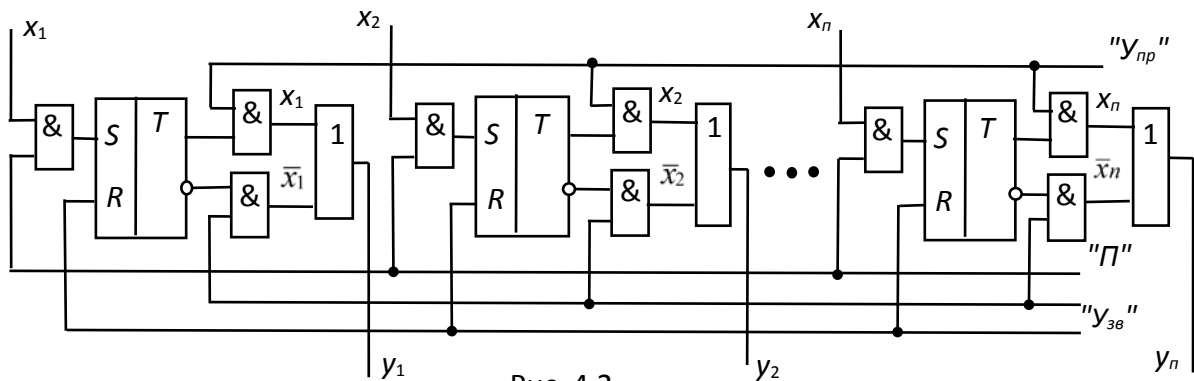


Рис. 4.2

Об'єднавши в одній мікросхемі декілька реєстрів і додавши на вході дешифратор DCW, а на виході мультиплексор MS отримують *реєстрову (надоперативну) пам'ять* (мал. 4.3). Входи D_i чотирьох або восьми реєстрів, як правило, 4-х розрядних, підключають до загальної вхідної шини даних DIN . Вхід завантаження необхідного реєстра вибирається дешифратором запису DCW на основі того, що поступає на його вхід адреси запису WA , тобто кода номера завантажуваного реєстра. *Запис* присутніх на шині даних, DIN відбувається у момент подання сигналу дозволу запису WE .

Виходи регістрів мультиплексором MS підключаються до вихідної шини *DOUT*. Номер регістра, з якого відбувається *читання* визначається за допомогою коду адреси читання *RA*. Дозвіл видачі даних в шину *DOUT* відбувається по сигналу *RE*. Оскільки дешифрування адреси запису і адреси читання виконується двома незалежними вузлами, що мають автономні адресні входи *WA* і *RA*, у

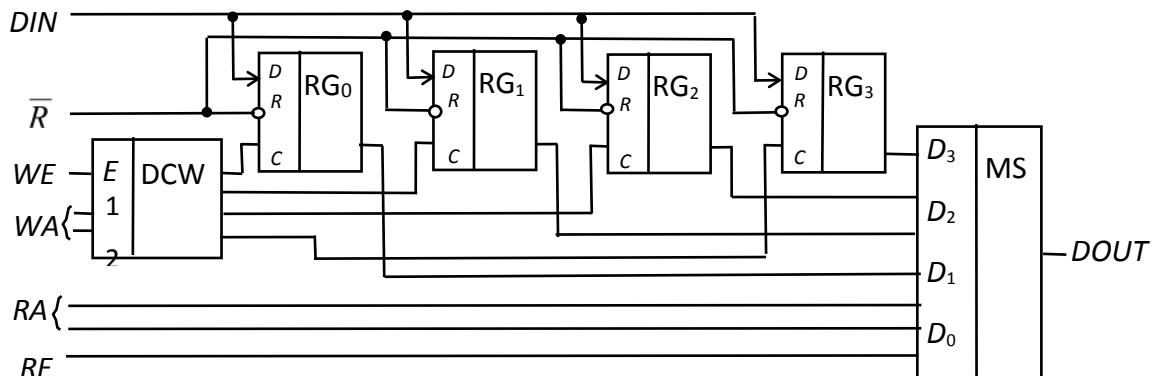


Рис. 4.3

регістрову пам'ять можна одночасно записувати бінарне число в один з регістрів і прочитувати число з іншого. Описана структура використана в кристалах мікросхем К155РП1, ІР11 і ІР12 серій К561 і К564.

Мікросхеми регістрової пам'яті легко нарощуються по розрядності і дозволяють нарощування числа регістрів. Вони розроблені для побудови блоків *регістрів загального призначення* (РЗП) призначених для тимчасового зберігання початкових даних і проміжних результатів розрахунку в мікропроцесорах.

НАВЧАЛЬНІ ЗАВДАННЯ І МЕТОДИЧНІ ВКАЗІВКИ ДО ЇХ ВИКОНАННЯ

Завдання 1. Відкрити файл або зібрати на робочому полі середовища Multisim WorkBench схему для дослідження *універсального регістра зсуву* (мал. 4.4) і **встановити** у діалогових вікнах компонентів їх параметри або режими роботи. **Скопіювати** схему (мал. 4.4) на сторінку звіту.

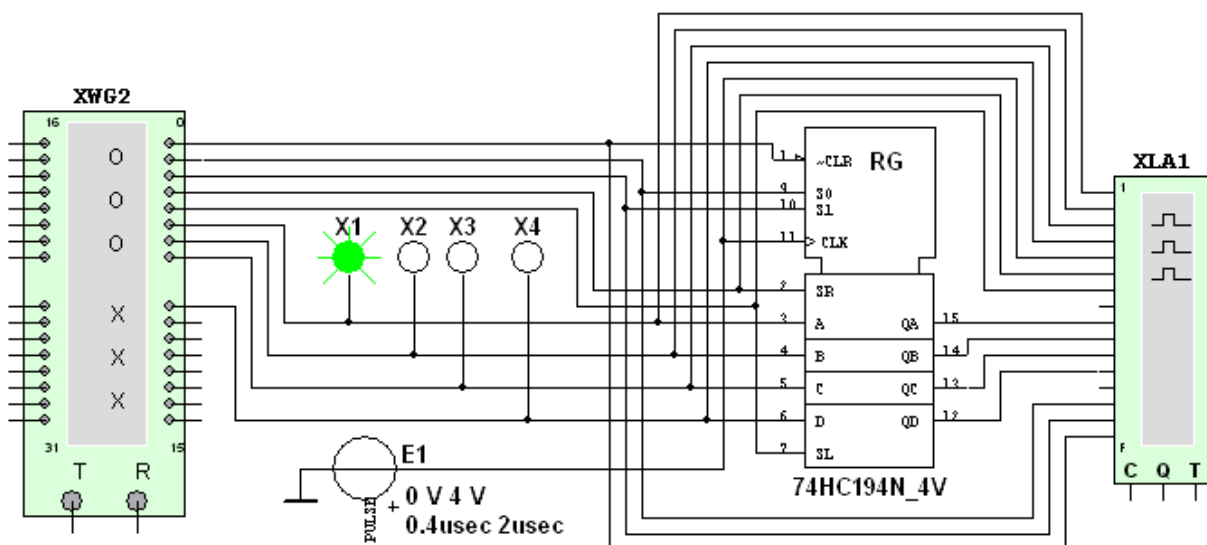


Рис. 4.4

Універсальний 4-х розрядний регістр зсув **74HC194N_4V** (аналогі-мікросхеми К230ІР2, КМ155ІР1, К176ІР3) здатний зсувати інформацію і вправо, і вліво, можлива як паралельний, так і послідовний запис даних. Регістр має паралельні входи (**A, Y, C, D**) паралельні виходи (**QA, QB, QC, QD**) послідовні входи (**SR, SL**) ланка прямого очищення регістра по входу \overline{CLR} і входи, що курують (**S0 і S1**) - входи встановлення режиму:

- **S0 = 1, S1 = 1** - запис даних в регістр по входах **A, B, C, D**;
- **S0 = 1, S1 = 0** - зсув даних вліво в напрямі від **QA** до **QD**;
- **S0 = 0, S1 = 1** - зсув даних вправо в напрямі від **QD** до **QA**;
- **S0 = 0, S1 = 0** - входи регістра недоступні (блокування).

Завдання 2. Скласти план дослідження паралельного регістра зсув, заповнивши елементи пам'яті генератора слова **XWG1** на основі правил функціонування регістра **74HC194_4V** наданих у таблиці 4.1.

Т а б л и ц я 4.1

Входи										Виходи			
Reset	Start	Режим		Послід. вхід		Паралельний вхід							
\overline{CLR}	\overline{CLX}	S0	S1	SR	SL	A	B	C	D	QA	QB	QC	QD
0	x	x	x	x	x	x	x	x	x	0	0	0	0
1	0	x	x	x	x	x	x	x	x	QA ₀	QB ₀	QC ₀	QD ₀
1	↑	1	1	x	x	A	B	C	D	A	B	C	D
1	↑	1	0	1	x	x	x	x	x	1	QA _n	QB _n	QC _n
1	↑	1	0	0	x	x	x	x	x	0	QA _n	QB _n	QC _n
1	↑	0	1	x	1	x	x	x	x	QB _n	QC _n	QD _n	1
1	↑	0	1	x	0	x	x	x	x	QB _n	QC _n	QD _n	0
1	x	0	0	x	x	x	x	x	x	QA ₀	QB ₀	QC ₀	QD ₀

Примітка. 0 - низький рівень; 1 - високий рівень; x - будь-який стан; ↑ - додатній перепад (з низького рівня на високий); QA₀, QB₀, QC₀, QD₀ - стаціонарні рівні A, B, C, D до установки вказаних станів на входах; QA_n, QB_n, QC_n, QD_n - відповідно рівні A, B, C, D перед початком проходження фронту самого останнього тактового імпульсу.

Запустити програму моделювання паралельного регістра, **скопіювати** у звіт програму (див. мал. 4,5) і часові діаграми сигналів на входах і виходах регістра (див. мал. 4.6).

Введення (запис) і виведення (зчитування) інформації виконується паралельним кодом. Введення забезпечується тактовим імпульсом, з приходом чергового тактового імпульсу записана інформація оновлюється. Зчитування інформації відбувається в прямому коді в інтервалі між синхроімпульсами, коли тригери знаходяться в режимі зберігання.

Керуючись схемою з'єднання генератора **XWG1** з регістром (див. мал. 4.4), при записі чисел в комірки пам'яті генератора в молодший розряд 9-розрядних чисел треба заносити значення сигналу \overline{CLR} : логічний 0 для очищення регістра або логічна 1 – дозвіл запису числа, зсув цих та інших у наступні два розряди – значення (1 або 0) сигналів **S0 і S1**, що визначають режим роботи регістра; у два

наступних – вводити значення сигналів **SR** і **SL**, що визначають напрям зсуву записаної інформації в напрямі від **QA** до **QB**, **QC**, а потім до **QD** після кожного позитивного перепаду імпульсу на тактовому вході $\overline{\text{CLK}}$ чи навпаки від **QD** до **QA**. У старші розряди треба занести довільні (чи за вказівкою викладача) значення 4-х розрядних чисел **DCBA** які передаються на відповідні виходи.

Як приклад на мал. 4.5 наведена запис 9-ти розрядних кодових комбінацій у 15 елементів пам'яті генератора **XWG1**, а на мал. 4.6 - реалізація програми моделювання паралельного регістра у вигляді часових діаграм сигналів (що виводяться у вікні аналізатора **XLA1**) на його входах і виходах при кроковому (**Step**) режимі роботи генератора **XWG1**.

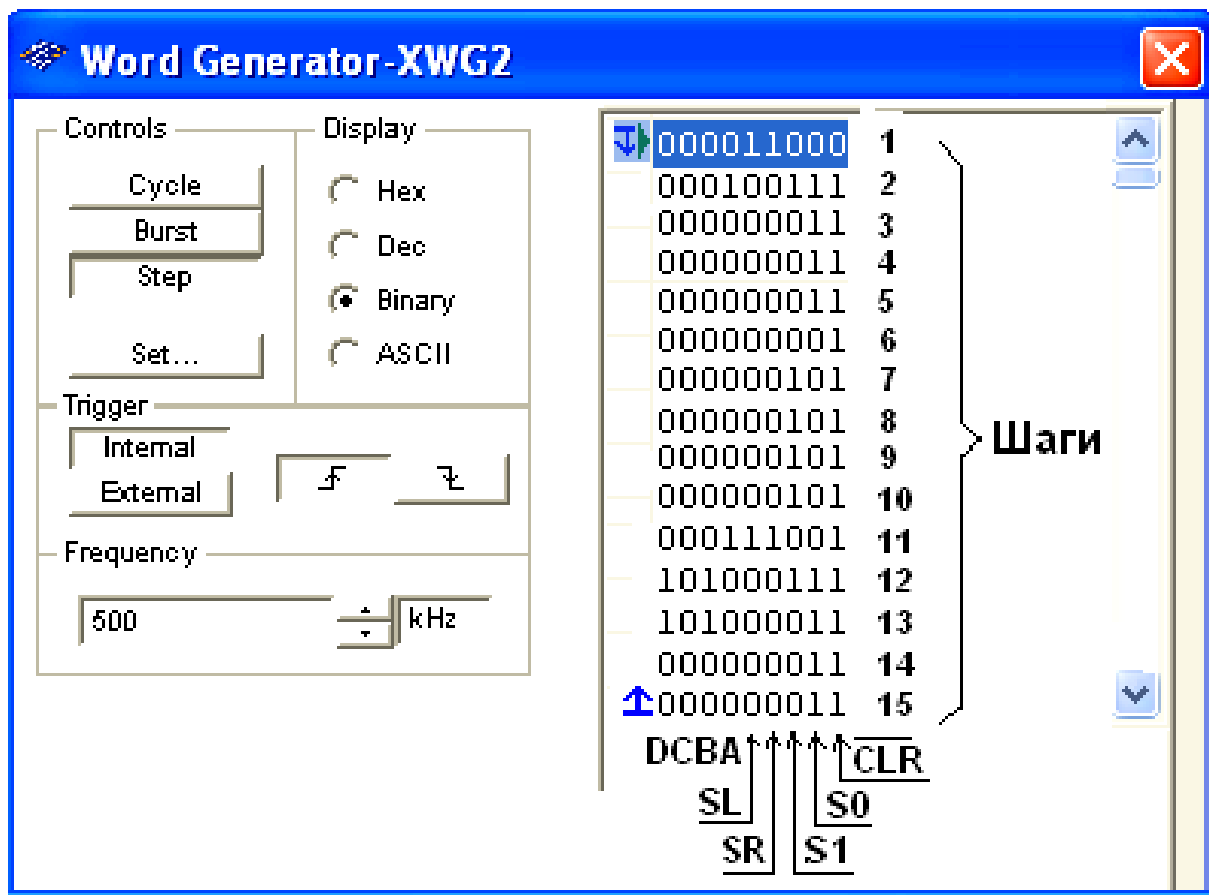


Рис. 4.5

На першому кроці (першому такті роботи генератора і регістра) при поданні сигналу $\text{CLR} = 0$ (дивитись перший рядок таблиці. 4.1 і мал. 4.6), на усіх входах і виходах регістра встановилися нульові значення. На другому кроці при $\text{CLR} = 1$, $\text{SR} = 0$, $\text{SL} = 0$ (дозвіл запису числа в регістр), $\text{S0} = 1$ і $\text{S1} = 1$ (заборона зсув даних в час синхронною паралельному запису числа в регістр, дивитись третій рядок таблиці. 4.1) відбувається завантаження 4-розрядного двійкового числа $\text{DCBA} = 0001$ в регістр.

При встановленні напрямку зсув даних вліво ($\text{S0} = 1$ і $\text{S1} = 0$, такт або крок 3) сигнал 0001 виводиться на виходи: $\text{QD} = 0$, $\text{QC} = 0$, $\text{QB} = 0$ і $\text{QA} = 1$. З приходом чергового тактового імпульсу (кроки 4, 5 і 6) відбувається перезапис (зсув) утримуваного тригера кожного розряду в сусідній розряд (від розряду **A** до

розряду **D**) без зміни дотримання одиниць і нулів. Після закінчення шостого тактового імпульсу на виході встановлюється число 1000 (дивитись мал. 4.6). Якщо виконати ще один крок $S0 = 1$ і $S1 = 0$ то занесена в регістр інформація буде повністю з нього виведена. Якщо при роботі регістра в режимі зсув вліво (дивитись кроки 3-6 на мал. 4.6) у елементи пам'яті генератора внести $SL = 1$, то сигнал 1 формуватиметься на виході **QA** і зрушуватися ліворуч від **QA** до **QD** при кожному тактовому імпульсі. В результаті, після шостого імпульсу на виході, встановиться сигнал 1111.

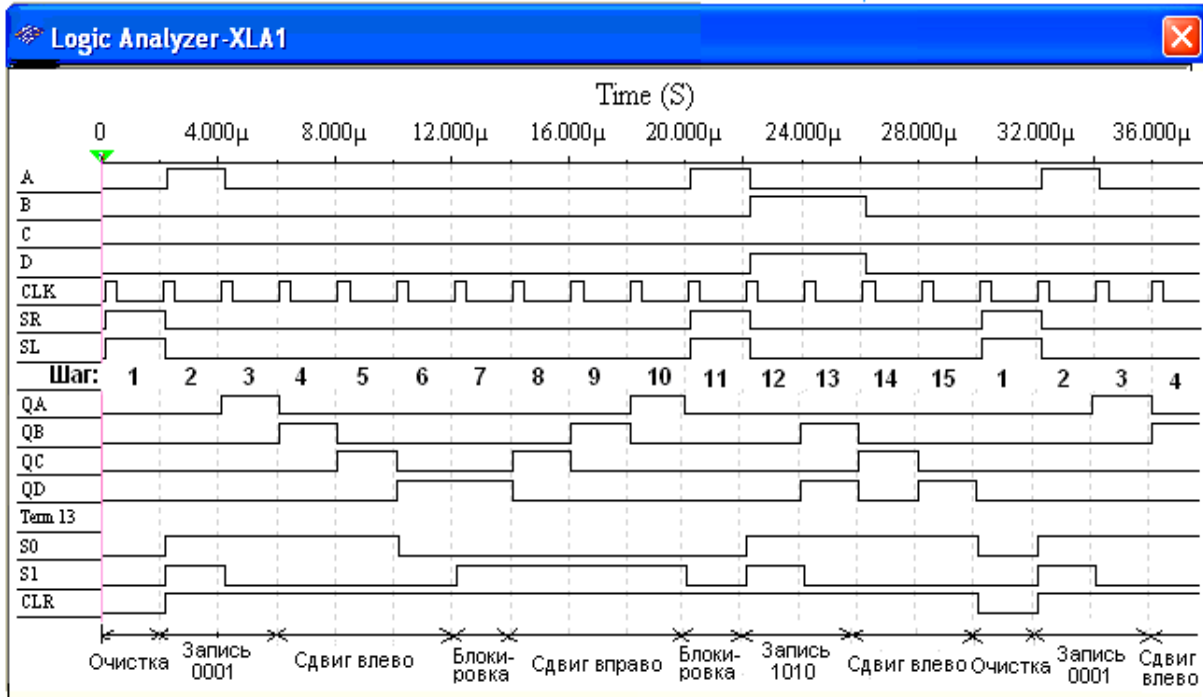


Рис. 4.6

Режим блокування реалізується при поданні на обидва керуючих входи сигналів низького рівня $S0 = S1 = 0$ при $CLR = 1$ (див. крок 7 і крок 11 на мал. 4.6). У режимі блокування дані в регістрі не зсуваються ні управо, ні вліво, а залишаються на своїх позиціях. При встановленні сигналів $S0 = 0$ і $S1 = 1$ з настанням 8, 9 і 10-го тактових імпульсів відбувається зсув сигналу 1000 управо і його повне виведення з регістра. Якщо при роботі регістра в режимі зсув вправо (див. кроки 8-10, на мал. 4,6) в елементи пам'яті генератора внести $SR = 1$, то сигнал 1 формуватиметься на виході **QD** і буде зсуватися вправо від **QD** до **QA** при кожному тактовому імпульсі. Що призведе до встановлення на виході сигнал 1111 після десятого імпульсу.

При установці $S0 = S1 = 0$ з приходом 11-го імпульсу відбувається блокування виходів, на наступному кроці виконується паралельний запис числа **DCBA** = 1010 у регістр, далі зсув даних вліво і т. д.

Завдання 3. Відкрити файл або зібрати на робочому полі середовища Multisim WorkBench схему для дослідження послідовного регістра зсув (мал. 4.7) і встановити у діалогових вікнах компонентів їх параметри або режими роботи. Скопіювати схему (мал. 4.7) у звіт.

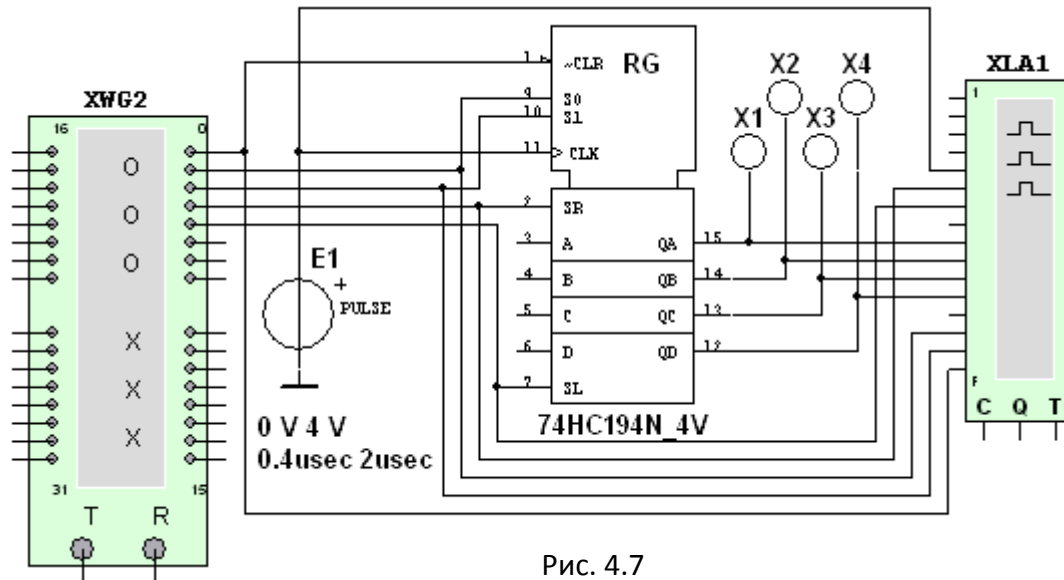


Рис. 4.7

Щоб мікросхема **74HC194N_4V** працювала в як *послідовний регістр зсуву вліво* треба подати на вхід, що управляє **S0** високий рівень напруги, а на вхід **S1** - низький рівень - встановити **S0 = 1** і **S1 = 0**, і подавати в послідовній формі на вхід **SR** дані, наприклад, 1, 0, 1 і 0, які записуються в розряд **A** і передаються на вихід **QA** (мал. 4.8). Регістр послідовно зсуває вліво ці сигнали від **QA** до **QD** на виході **QD** вони втрачаються (див. кроки 3- 9 на мал. 4.8).

При установці **S1 = 0** і **S1 = 1** і поданню на вхід **SL** даних в послідовній формі

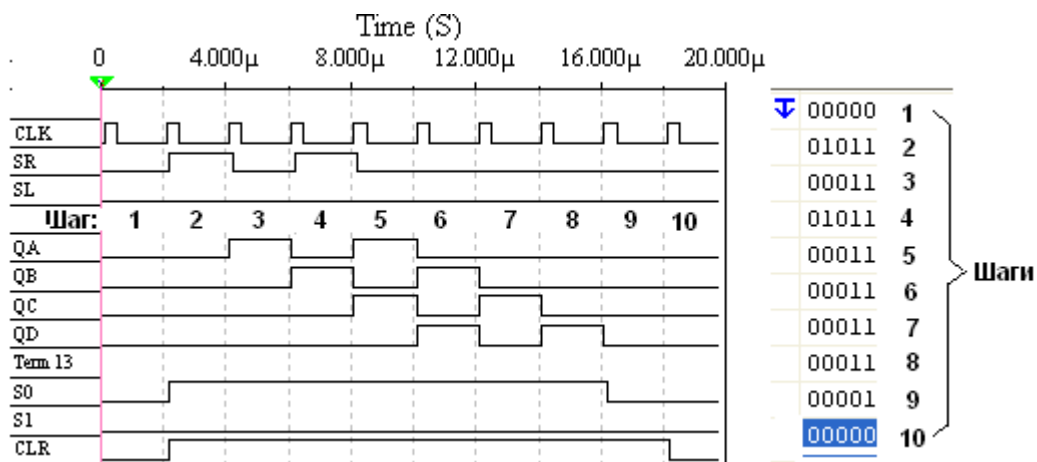


Рис. 4.8

наприклад, 1, 0, 0 і 1, які записуються в розряд **D** (і передаються на вихід **QD**) мікросхема працює в режимі *послідовного регістра зсув вправо* (без кільцевого переміщення даних): сигнали 1, 0, 0 і 1 зсуваються по напрямку до розряду **A**, на виході **QA** вони втрачаються (кроки 3-9, мал. 4.9).

Завдання 4. Скласти план дослідження послідовного регістра **74HC194_4V**, заповнивши елементи пам'яті генератора **XWG1** довільними (або за завданням викладача) 4-розрядними кодovими комбінаціями, що вводяться послідовно спершу в регістр **A** а потім в регістр **D**.

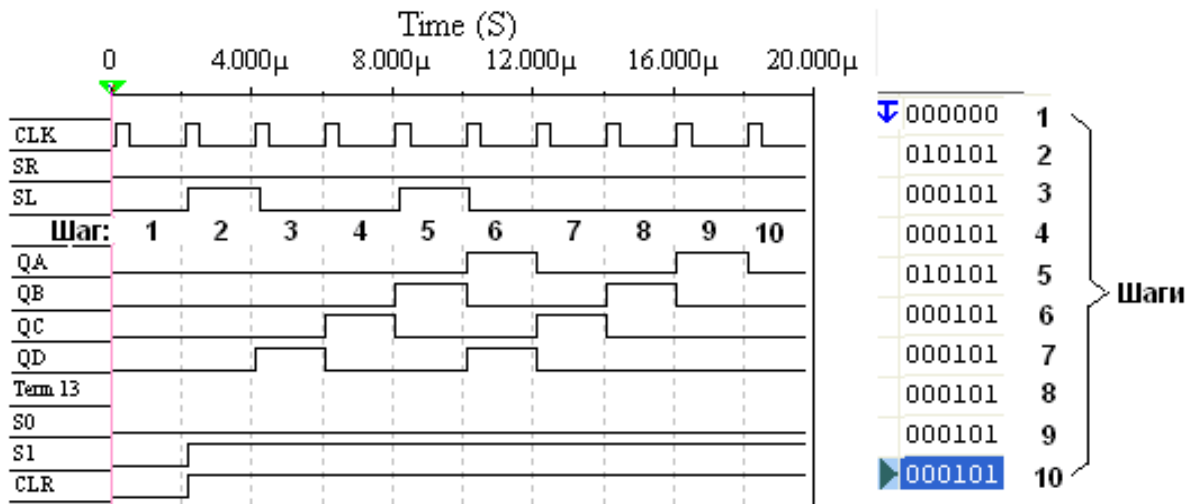


Рис. 4.9

Запустити програму моделювання послідовного регістра, **скопіювати** у звіт часові діаграми сигналів на входах і виходах регістра при зсуву даних вліво (дивитись мал. 4.8) і вправо (дивитись мал. 4.9).

ЗМІСТ ЗВІТУ

1. Найменування і мета роботи.
2. Перелік приладів, використаних в експериментах з їх короткими характеристиками.
3. Зображення електричних схем для випробування паралельного і послідовного регістрів.
4. Копії часових діаграм, що відображають роботу досліджуваних регістрів.
5. Висновки по роботі.

Лабораторна робота №5

АНАЛОГО-ЦИФРОВИЙ ПЕРЕТВОРЮВАЧ

МЕТА РОБОТИ:

Ознайомлення з принципом роботи і дослідження інтегрального 8-ми розрядного аналого-цифрового перетворювача.

ТЕОРЕТИЧНІ ВІДОМОСТІ ТА РОЗРАХУНКОВІ ФОРМУЛИ

1. СТРУКТУРНА СХЕМА АЦП ПОСЛІДОВНОЇ ДІЇ

Аналого-цифровий перетворювач (АЦП) - пристрій, призначений для перетворення аналогових величин в їх цифровий еквівалент в різних системах числення. Вхідним сигналом АЦП впродовж деякого проміжку часу Δt є постійна напруга, рівна відліку $u_{ex}(k\Delta t)$ вхідній аналоговій функції u_{ex} . За цей час на виході АЦП формується цифровий (зазвичай двійковий) код

$$A_i (a_{n-1} a_{n-2} \dots a_1 a_0),$$

що відповідає дискретному відліку напруги $u_{ex}(k\Delta t)$. Кількісний зв'язок для будь-якого моменту часу визначається співвідношенням

$$A_i = u_{ex}(k\Delta t) / \Delta u \pm \delta_i,$$

де Δu – крок квантування вхідної аналогової напруги u_{ex} ; δ_i - погрішність перетворення напруги $u_{ex}(k\Delta t)$ на цьому кроці.

Фізичний процес аналого-цифрового перетворення складається з дискретизації за часом аналогового сигналу, квантування по рівню і кодуванням. Процес *дискретизація* аналогового сигналу тривалістю t_{ex} виконується відповідно до теореми Котельнікова, де необхідний крок дискретизації $\Delta t \leq 1/(2f_m)$ де f_m - максимальна частота спектру вхідного сигналу, і число кроків $M = t_{ex}/\Delta t$.

Процес *квантування за рівнем* функції, що дискретизує $u_{ex}(k\Delta t)$ полягає у відображенні її значень на деяку безліч кінцевих значень $u_d(k)$ рівне числу рівнів квантування $N = u_{ex,max}/\Delta u$. Процес квантування за рівнем (округлення кожного значення $u_{ex}(k\Delta t)$ до найближчого рівня $u_d(k)$) призводить до виникнення помилки (шуму) квантування, максимальне значення якої $\pm 1/2\Delta u$ і визначається розрядністю використовуваного вихідного коду. При збільшенні розрядності вихідного коду помилка квантування може бути зменшена до скільки завгодно малої величини, але не може бути зведена до нуля вибором параметрів пристрою, оскільки вона властива цьому алгоритму.

Процес *кодування* полягає в заміні знайдених квантованих $N + 1$ значень вхідного сигналу $u_d(k)$ деякими цифровими кодами.

На мал. 5.1, а надана характеристика ідеального АЦП в нормованих одиницях вхідної напруги $u_{ex,n} = u_{ex}/u_{ex,max}$. Окрім помилки квантування, при оцінці точності АЦП враховують додаткові погрішності: *інструментальну* і *апертурну*, що

виникає через невідповідність значення вхідного сигналу $u_0(k)$ перетвореному цифровому коду A_i . Невідповідність виникає, якщо вхідний сигнал впродовж інтервалу дискретизації Δt змінюється більш ніж на значення кроку квантування Δu .

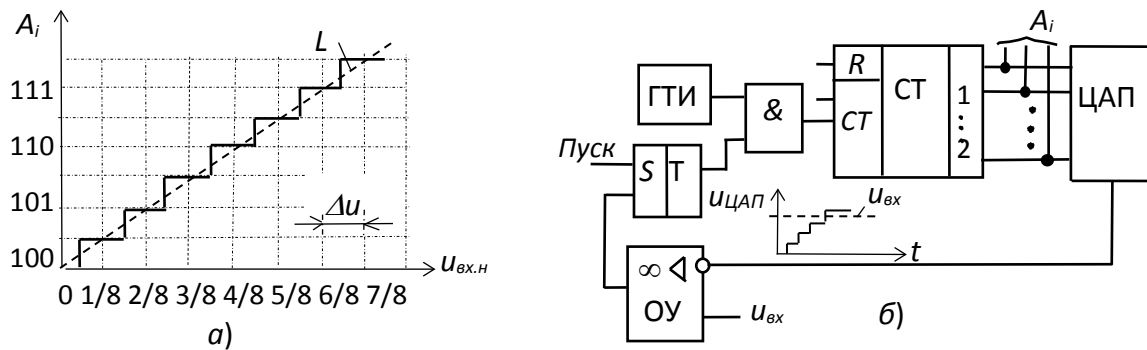


Рис. 5.1

2. ОСНОВНІ ПАРАМЕТРИ АЦП

До основних параметрів АЦП відносять:

- число розрядів вихідного коду $n = 8..16$, що відображає початкову аналогову величину, яке може формуватися на виході АЦП. При використанні двійкового коду $n = \log_2(N + 1)$, де $N + 1$ – максимальне число кодових комбінацій (рівнів квантування) на виході АЦП;

- діапазон зміни вхідної напруги $u_{вх.max}$. Відмітимо, що АЦП може обробляти вхідну інформацію у вигляді однополярної аналогової напруги з межами 0 . $u_{вх.max}$ і двуполярного $\pm u_{вх.max} / 2$;

- абсолютна роздільна здатність $СМР = \Delta u$ (значення молодшого розряду) - середнє значення мінімальної зміни вхідного сигналу $u_{вх}$, що обумовлює збільшення або зменшення вихідного коду на одиницю. Значення СМР визначається розрядністю вихідного коду і діапазоном вхідної напруги;

- абсолютна похибка δ_i перетворення в кінцевій точці шкали є відхилення реального максимального значення вхідного сигналу $u_{вх.max}$ від максимального значення ідеальної характеристики L АЦП (див. мал. 5.1, а). Зазвичай δ_i вимірюється в СМР;

- максимальна частота перетворення (десятки і сотні кілогерц);

- час перетворення вхідного сигналу: $t_{np.max} \leq (1/2)\Delta t$.

Реалізація АЦП на відміну від ЦАП може змінюватися в значній мірі залежно від обраного методу перетворення і способу його реалізації. Найбільше поширення отримали три основних методи: послідовного обрахунку, порозрядного кодування і прочитування.

Метод послідовного обрахунку заснований на урівноваженні вхідної величини сумою однакових за величиною еталонів (сумою кроків квантування). Момент урівноваження визначається з допомогою одного компаратора, а кількість еталонів, що урівноважують вхідну величину, підраховується за допомогою лічильника.

Метод порозрядного кодування (урівноваження) передбачає наявність декількох еталонів (часто реалізованих у вигляді урівноважуючого регістра зсуву), зазвичай пропорційних за величиною мірам числа 2, і порівняння цих еталонів з аналоговою величиною. Порівняння розпочинається з еталону старшого розряду. У

залежності від результату цього порівняння формується значення старшого розряду вихідного кода. Якщо еталон більше вхідної величини, то в старшому розряді ставиться 0 і далі робиться урівноваження вхідної величини наступним за значенням еталоном. Якщо еталон рівний або менше вхідної величини, то в старшому розряді вихідного кода ставиться 1 і надалі виконується урівноваження різниці між вхідною величиною і першим еталоном.

Найбільшою швидкодією володіють перетворювачі, побудовані за методом прочитування. *Метод прочитування* має на увазі наявність $2^n - 1$ еталонів при n -розрядному двійковому коді. Вхідна аналогова величина одночасно порівнюється з усіма еталонами. В результаті перетворення виходить паралельний код у вигляді логічних сигналів на виходах $2^n - 1$ компараторів.

3. ВАРІАНТ РЕАЛІЗАЦІЇ АЦП ПОСЛІДОВНОГО ТИПУ

Як приклад розглянемо структурну схему АЦП послідовного типу з ЦАП в колі зворотного зв'язку (мал. 5.1, б) і варіант її реалізації (мал. 5.2). По сигналу "Пуск" на вхід обнуленого лічильника СТ починають подаватися імпульси генератора тактової частоти ГТИ (див. мал. 5.1, б). У міру вступу цих імпульсів росте вхідний код ЦАП і ступінчасто підвищується напруга $u_{\text{цп}}$ на його виході, причому рівень ступеня відповідає кроку квантування Δu вхідної напруги $u_{\text{вх}}$ АЦП.

Процес перетворення закінчується, коли напруга $u_{\text{цп}}$ стане трохи більше вхідної напруги $u_{\text{вх}}$ АЦП, поданого на вхід ОУ, на якому зібраний компаратор. При цьому робота лічильника припиняється, а на його виході встановлюється код A_i , що є цифровим еквівалентом напруги $u_{\text{вх}}$.

Згідно розглянутій структурній схемі АЦП на мал. 5.2 наведено варіант реалізації моделі 4-х розрядного АЦП послідовного типу з ЦАП, що складається з операційного підсилювача **ОРАМР1** і резистивної матриці **R1-R4** з зваженими опорами. Перемикачі **Key1-Key4** у схемі (при розімкненому ключі **Space**) для перевірки роботи лічильника СТ, а осцилограф **XSC1** – для зняття осцилограм напруги з виходу ЦАП і входу компаратора.

При запуску моделювання АЦП сформовані генератором **E1** імпульси подаються на вхід лічильника СТ, число яких послідовно висвічується на 7-ми сегментному індикаторі. Вихідні порозрядні сигнали з лічильника поступають також на входи логічного аналізатора **XLF1** і входи резистивної матриці **R1-R4**, а сумарна напруга з матриці – на вхід ОУ. Ступінчаста напруга $u_{\text{цп}}$ з виходу **ОРАМР1** (мал. 5.3) подається на вхід компаратора, зібраного на операційному підсилювачі **ОРАМР2**. На цей же вхід подано постійна напруга $u_{\text{вх}}$ з генератора **E7** через дільник **R6-R7**. В момент, коли вказана напруга порівнюється, компаратор спрацьовує, на елемент І (**AND**) подається логічний 0 і припиняється

робота лічильника, а на індикаторі висвічується цифровий код (число кроків

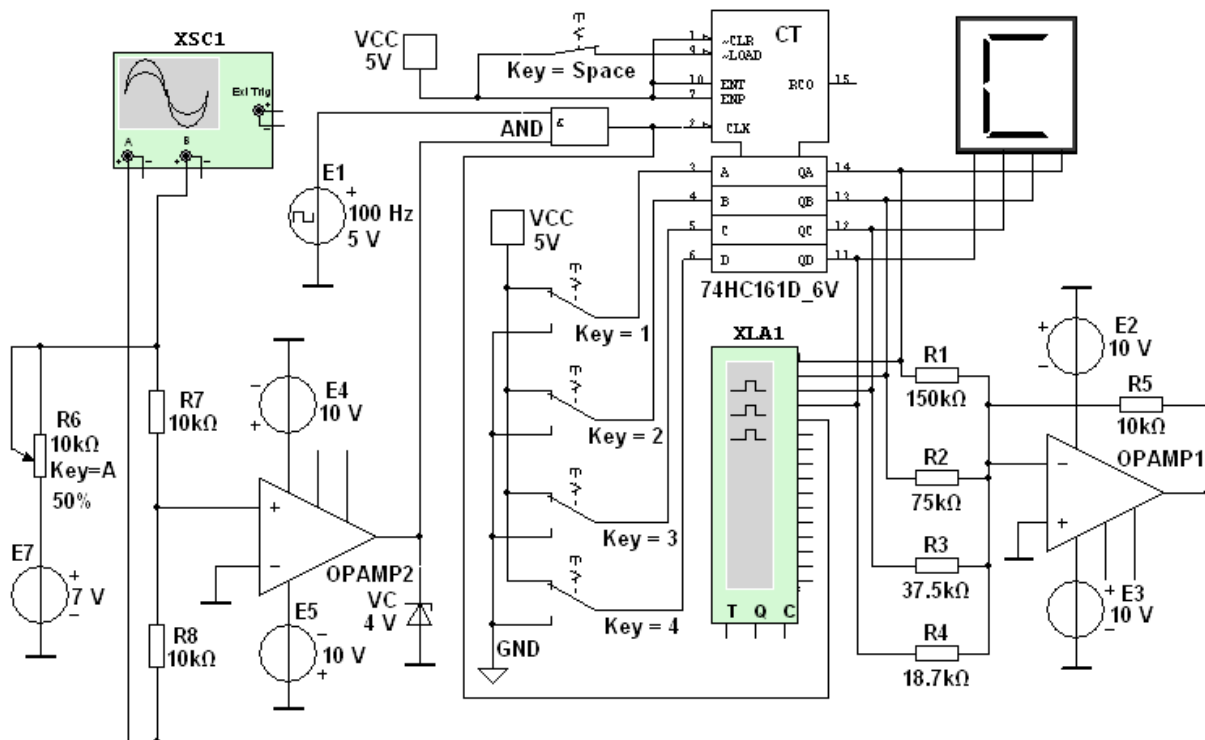


Рис. 5.2

квантування), що відповідає рівню $u_{\delta}(k)$.

Аналіз часових діаграм сигналів з виходу лічильника і осцилограм напруги з входів компаратора (дивись мал. 5.3) показує, що лічильник перервав з приходом дванадцятого тактового імпульсу, тому на 7-ми сегментному індикаторі висвітилося число C_{16} (12_2) (див. мал. 5.2).

Похибка перетворення залежить від кроку квантування (висоти ступеня напруги $u_{\text{цан}}$) погрішності у формуванні ступінчастої напруги $u_{\text{цан}}$ і помилки компаратора у визначенні рівності $u_{\text{вх}}$ і $u_{\text{цан}}$. Час перетворення не має постійного значення і залежить від рівня напруги $u_{\text{вх}}$. При заданому числі розрядів АЦП час перетворення визначається числом періодів рахункових імпульсів.

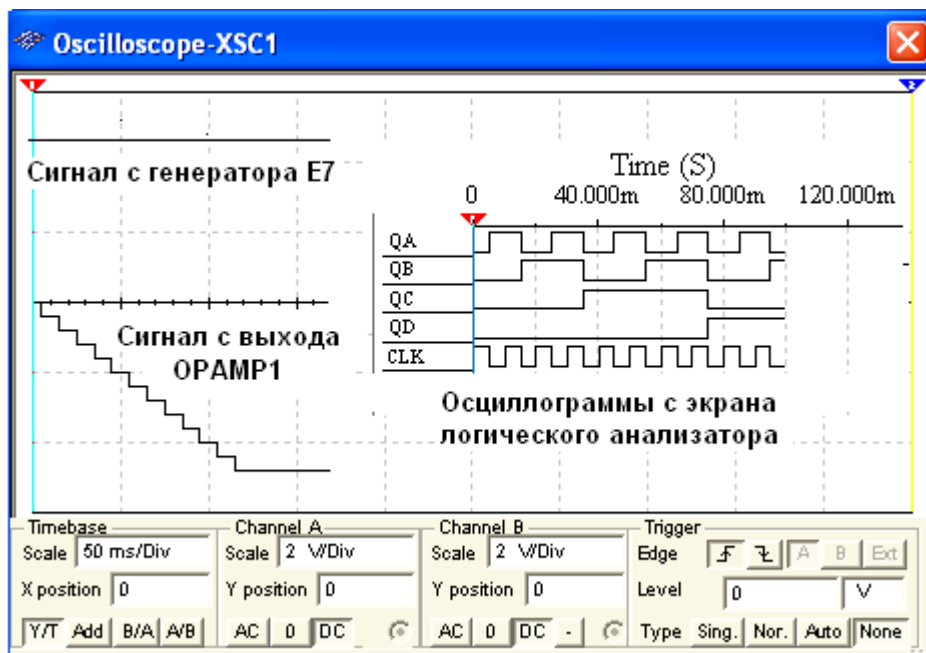


Рис. 5.3

Плати АЦП/ЦАП, наприклад застосовують у цифрових вимірювальних приладах, в системах обробки і відображення інформації, в автоматичних системах контролю і управління, в підсистемах введення-виведення інформації ЕОМ, тощо.

Основні напрями розвитку АЦП - підвищення швидкодії основних вузлів, підвищення їх точності, збільшення розрядності перетворювачів до 24, використання мікропроцесорів в перетворювачах.

ЗАВДАННЯ І МЕТОДИЧНІ ВКАЗІВКИ ДО ЇХ ВИКОНАННЯ

Завдання 1. Відкрити файл або зібрати на робочому полі середовища Multisim WorkBench схему для дослідження *аналого-цифрового перетворювача* з ЦАП (мал. 5.4) і **встановити** у діалогових вікнах компонентів їх параметри або режими роботи. **Скопіювати** схему (мал. 5.4) на сторінку звіту.

У схему (мал. 5.4) включені бібліотечний 8-ми розрядний АЦП (ADC); джерела опорної напруги **E1** і **E2** (підключені до входів **Vref+** і **Vref-** АЦП); генератор **E4** для синхронізації роботи (підключений до входу **SOC**) і дозволи (вхід **OE**) на видачу двійкової інформації на виходи **D0-D7** АЦП, з якими сполучені входи логічного аналізатора **XLA1** і пробники **X0-X7**; функціональний генератор **XFG1** як джерело вхідного сигналу u_{ex} (підключений до входу **Vin**);

ЦАП (DAC) і осцилограф XSC1. Вихід EOC служить для передачі двійкової

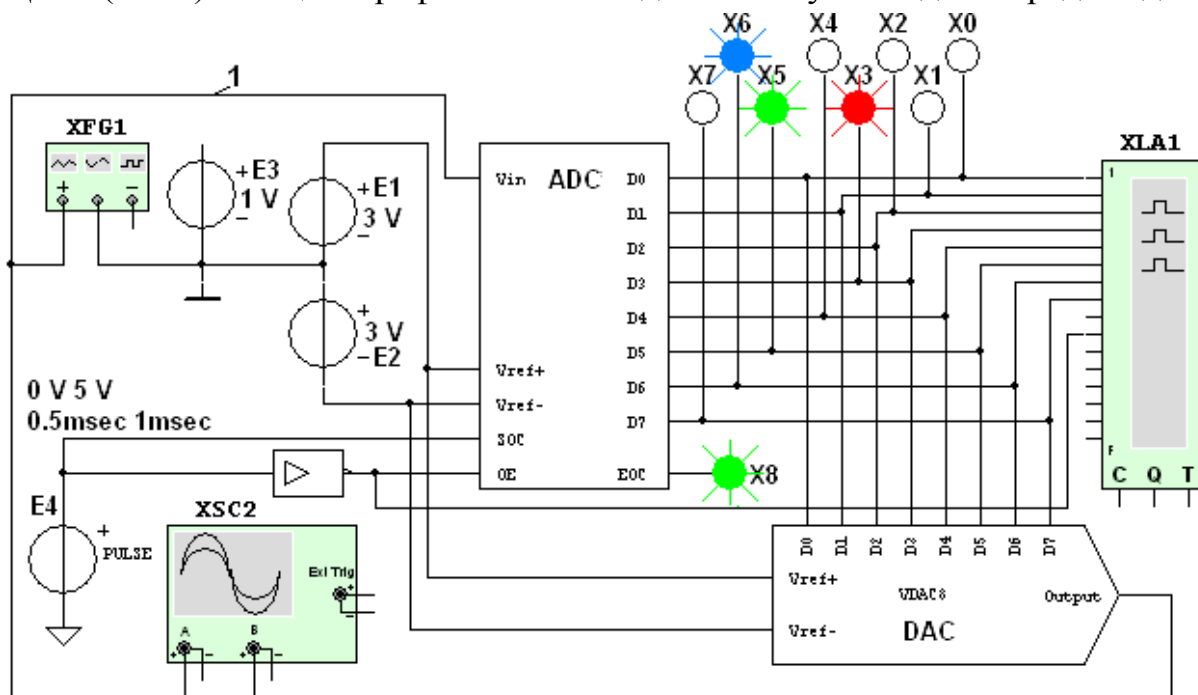


Рис. 5.4

інформації АЦП, наприклад, на ЕОМ.

Завдання 2. Дослідити точність перетворення АЦП вхідної напруги $u_{вх}$ у цифровий код з допомогою пробників X0-X7 логічного аналізатора XLA1, а також ЦАП і осцилографа XSC1.

З цією метою:

– тимчасово **видалити** дріт 1 (дивись мал. 5.4) і підключити вхід **Vin** АЦП до позитивного полюса джерела постійної напруги **E3**;

– **скласти** таблицю, аналогічну таблиці 5.1, в перший стовпець якої записати рівні напруги

$$u_{вх} = 0,1; 0,2; 0,5; 1,0; 1,5; 2,0; 2,4; -0,5; -1,0; -2,0 \text{ В}$$

по черзі, що задаються у діалоговому вікні генератора **E3**;

– **встановити** у діалогових вікнах генераторів **E1** і **E2** $E_1 = 2,5 \text{ В}$ і $E_2 = -2,5 \text{ В}$;

– **запустити** програму моделювання АЦП і **занести** у поля складеної таблиці значення напруги $u_{вих(ЦАП)}$ з виходу ЦАП, вимірювані на екрані осцилографа за допомогою візирної лінії; двійковий еквівалент $D_{(2)}$ напруги, що визначається по світінню пробників **X7-X0**; 16-ний код $D_{(16)}$ прочитується з дисплея аналізатора **XLA1**;

– отримувані з виходу АЦП десяткові інверсні сигнали $D_{(10)інв}$ **перерахувати** на неінверсні $D_{(10)}$ за виразом

$$D_{(10)} = D_{(10)інв} - 128$$

і **занести** у відповідні стовпці таблиці;

– розрахункові десяткові еквіваленти $D_{(10)роз}$ двійкового коду $D_{(2)}$ на виході АЦП при заданому значенні вхідної напруги $u_{вх}$ **визначити** по формулі

$$D_{(10)роз} = 256u_{вх} / (E_1 + |-E_2|),$$

і **занести** у другий справа стовпець таблиці;

– **розрахувати** похибку виміру напруги за виразом

$$\Delta U\% = 100 \cdot (u_{вих(ЦАП)} - u_{ex}) / u_{ex}$$

і **занести** у правий стовпець таблиці.

У якості прикладу в таблиці 5.1 приведені дані вимірів при моделювання АЦП при $E_1 = 3$ В і $E_2 = -3$ В які близькі до розрахункових значень. Так, при $E_1 = |E_2| = 3$ В і $u_{ex} = E_3 = 1$ В розрахунковий десятковий еквівалент $D_{(10)роз} = 256 \cdot 1/6 \approx 42,67$ при вимірюваному $D_{(2)} = 10101010$ і $D_{(10)} = 42$. При цьому похибка виміру склала 3,56%.

Т а б л и ц я 5.1

u_{ex} , В	$u_{вих(ЦАП)}$, В	$D_{(2)}$	$D_{(16)}$	$D_{(10).інв}$	$D_{(10)}$	$D_{(10)роз}$	$\Delta U\%$
0,1	0,09375	10000100	84	132	4	4,27	6,25
0,5	0,5156	10010101	95	149	21	21,33	3,12
1,0	0,9644	10101010	AA	170	42	42,67	3,56
2,0	2,017	11010101	D5	213	85	85,34	0,85
2,5	2,484	11101010	EA	234	106	106,67	0,64
2,9	2,906	11111011	FB	251	123	123,74	0,21
-1,0	-0,9844	01010101	55	85	-43	-42,67	3,56

Завдання 3. Досліджувати процес перетворення вхідної напруги трикутної форми в цифрові коди, а потім з допомогою ЦАП – у ступінчасту напругу, що апроксимує напруга u_{ex} .

Для цього:

– **видалити** дріт, що сполучає вихід генератора **E3** з входом **Vin** АЦП, і **відновити** дріт 1, сполучаючий вихід "+" функціонального генератора **XFG1** з входом **Vin** АЦП (див. мал. 5.4);

– **встановити** параметри генератора **XFG1** (мал. 5.5, а): напруга трикутної форми $N = 99$ і амплітудою 1 В (діапазон від -1 В до 0,98 В) і його частоту $f_2 = 50$ Гц;

– **запустити** програму моделювання АЦП;

– **отримати** і **скопійювати** на сторінку звіту осцилограму вхідної напруги u_{ex} осцилограму ступінчастої напруги $u_{вих(ЦАП)}$ з виходу ЦАП (см мал. 5.5, б) і часові діаграми сигналів з виходів **D0-D7** АЦП, що поступають на входи логічного

аналізатора **XLA1** і що є двійковими еквівалентами дискретних відліків $u_{ex}(k\Delta t)$ вхідної напруги (мал. 5.6);

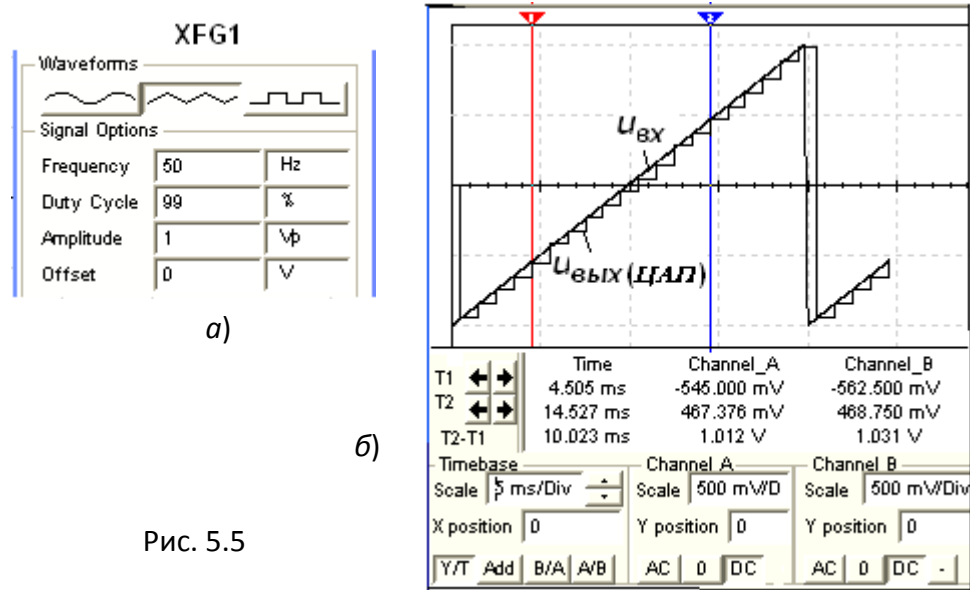


Рис. 5.5

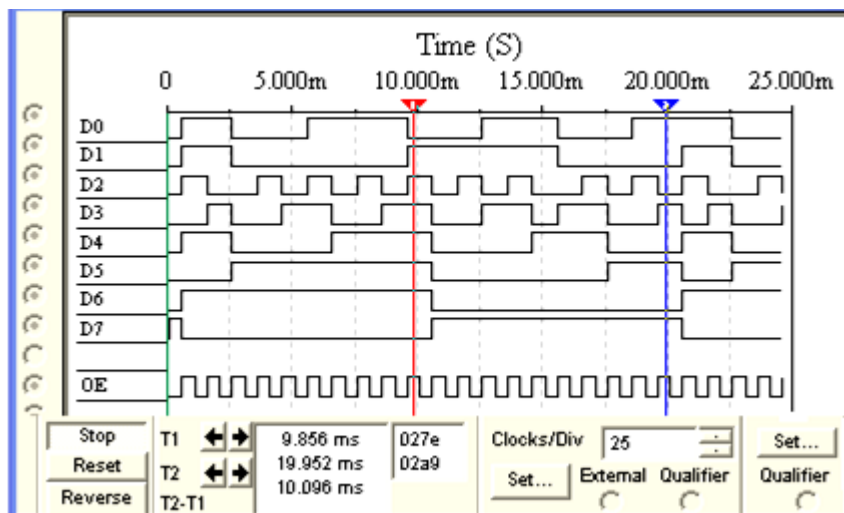


Рис. 5.6

– скориставшись візирними лініями, **провести аналіз** формування напруги $u_{вих(ЦАП)}$, що апроксимує вхідну напругу u_{ex} , зокрема, **виміряти** напругу і висоту його східців в різні моменти перетворення (з інтервалом в 1 мс в моменти позитивного перепаду тактового імпульсу синхронізації) і **порівняти** їх з відліками $u_{ex}(k\Delta t)$ напруги u_{ex} .

Так, при частоті синхронізації $f_3 = 1$ кГц і частоті пилоподібної напруги $f_2 = 50$ Гц утворилося на виході ЦАП двадцять східців напруги $u_{вих(ЦАП)}$ середня висота яких рівна $U_{cm} \approx 93,7$ мВ при розрахунковому значенні $\Delta u = u_{ex,max}/(N + 1) = 1,98/21 = 94$ мВ. Перший ступінь заввишки 66 мВ сформувалася після закінчення 0,5 мс з моменту включення моделювання при рівні вхідної напруги $u_{ex} = -93,4$ мВ, друга – при $u_{ex} = -0,849$ В амплітудою 93,75 мкВ і і т. д.

Завдання 4 (виконується факультативно або за вказівкою викладача). **Дослідити** процес перетворення АЦП вхідної синусоїдальної напруги у цифрові коди, а потім з допомогою ЦАП - в ступінчасту напругу.

З цією метою:

– **клацнути мишею** на кнопки "Синусоїдальна напруга" генератора XFG1 (див. мал. 5.5, а) і **встановити** частоту напруги $f_2 = 25$ Гц, а потім, при зупинці моделювання, $f_2 = 5$ Гц зі зміною часу розгортки променів осцилографа з 10 мс/под. на 50 мс/под. **Змістити** вгору на 0,6 ділень осцилограму вхідної напруги u_{ex} (мал. 5.7);

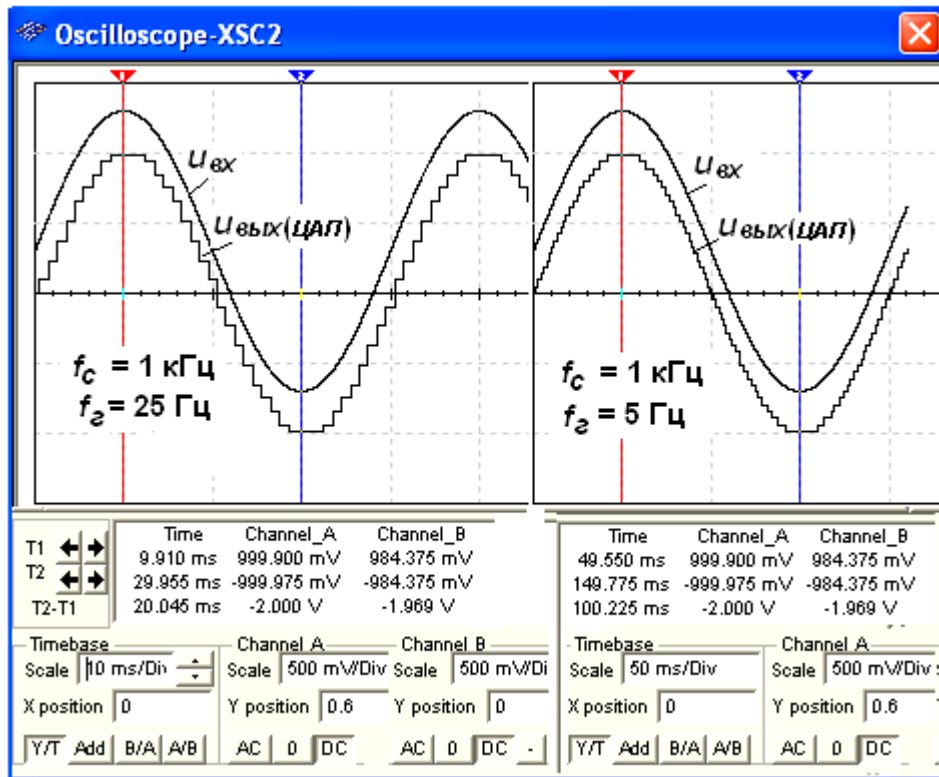


Рис. 5.7

– **виміряти** напруга $u_{вих(ЦАП)}$ і висоту його східців в різні моменти перетворення і **порівняти** їх з відліками напруга $u_{ex}(k\Delta t)$ вхідної напруга u_{ex} для моментів позитивного перепаду тактового імпульсу синхронізації.

Двійкові еквіваленти відліків напруги $u_{ex}(k\Delta t)$ з виходів АЦП перетворюються з допомогою ЦАП в аналоговий ступінчастий сигнал $u_{вих(ЦАП)}$ (дивись мал. 5.7). При цьому зі зменшенням частоти сигналу збільшується число східців і перетворена крива добре апроксимує вхідний сигнал. Висота східців змінна, від 46 мВ до 141 мВ, оскільки інтервал дискретизації Δt при заданій частоті синхронізації постійний. Особливо помітна верхня і нижня сходинка з відхиленням від амплітуди вхідної напруги приблизно на 15,5 мВ, оскільки на інтервалах дискретизації біля максимуму амплітуд швидкість зміни напруги мінімальна.

ЗМІСТ ЗВІТУ

1. Найменування і мета роботи.
2. Перелік приладів, використаних в експериментах, з їх короткими характеристиками.
3. Зображення електричної схеми для дослідження аналого-цифрового перетворювача.
4. Копії осцилограм і часових діаграм сигналів з різних вузлів схеми, що відображають роботу досліджуваного АЦП.
5. Таблиця з результатами вимірів і розрахунків вхідних відліків вхідної напруги і вихідних кодів АЦП.
6. Висновки по роботі.

Лабораторна робота №6

ЦИФРО-АНАЛОГОВИЙ ПЕРЕТВОРЮВАЧ

МЕТА РОБОТИ:

Ознайомлення з принципом роботи і дослідження інтегрального цифро-аналогового перетворювача.

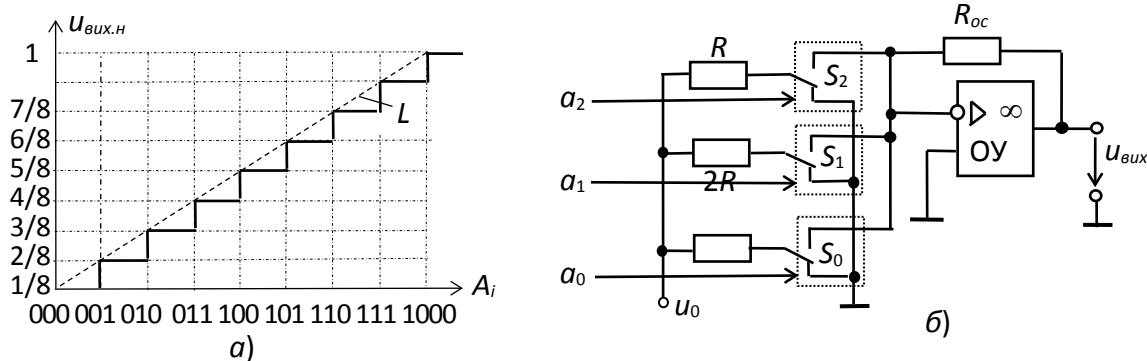
ТЕОРЕТИЧНІ ВІДОМОСТІ І РОЗРАХУНКОВІ ФОРМУЛИ

1. СТРУКТУРА РЕЗИСТИВНИХ МАТРИЦЬ ЦАП

При побудові пристроїв, що зв'язують цифровий пристрій з об'єктами в реальному світі, необхідно забезпечити транслявання цифрового коду в певні значення величин реального аналогового світу. Пристрій, що здійснює автоматичне перетворення вхідних значень, представлених числовими кодами, в еквівалентні їм значення фізичної величини (напруги, струму та ін.), називають *цифро-аналоговим перетворювачем* (ЦАП).

Отже, цифро-аналоговий перетворювач призначений для прямого перетворення вихідного двійкового кода, наприклад, $A_i(a_2a_1a_0)$ у аналоговий еквівалент. Вихідна аналогова величина напруга $u_{вих}$, іноді нормована $u_{вих.н} = u_{вих}/u_{вих.мах}$ відповідає кодовій комбінації A_i , що поступила на вхід, і відтворюється для дискретних моментів часу (мал. 6.1, а).

Існує два поширених способи цифро-аналогового перетворення з використанням:



Мал. 6.1

- резистивної матриці з ваговими двійково-зваженими опорами;
- резистивної матриці з двома номіналами опорів, яку зазвичай називають матрицею $R-2R$.

ЦАП з ваговими *двійково-зваженими опорами* (мал. 6.1, б) складається:

- з n перемикачів S_i (по одному на кожен розряд) керованих двійковим кодом A_i ;
- з матриці двійково-зважених резисторів з опорами $2^{n-1}R$;
- з джерела опорної напруги u_0 і вихідного операційного підсилювача ОП, з якого сумуються струми, що протікають через резистори з

двійково-зрівноваженими опорами, для отримання аналогової вихідної напруги $u_{вих}$.

Кожен i -й розряд керує перемикачем S_i який підключається до джерела опірної напруги u_0 коли $a_i = 1$, або до загальної шини, коли $a_i = 0$. Опори резисторів $2^{n-1}R$ (n - номер розряду вхідного коду), сполучених з ключами, такі, що забезпечують пропорційність в них струму двійковій вазі відповідного розряду вхідного коду. Отже, струм на вході ОУ і вихідна напруга ЦАП:

$$i = \frac{a_{n-1}u_0}{R} + \frac{a_{n-2}u_0}{2R} + \dots + \frac{a_1u_0}{2^{n-1}R} + \frac{a_0u_0}{2^n R}; \quad u_{вих} = -R_{oc}i = -u_0 \frac{R_{oc}}{2^n R} \sum_{i=0}^{n-1} a_i 2^i.$$

Напруга на виході ЦАП пропорційно "вазі" присутнього на входах кода, а максимальне значення має місце, коли усі розряди набудуть значення 1 тобто

$$u_{max} = \left| u_0 \frac{(2^n - 1)R_{oc}}{2^n R} \right|,$$

і воно завжди менше за опірну напругу на крок квантування $u_0 R_{oc} / (2^n R)$.

Номинали опорів резисторів в молодшому і старшому розрядах відрізняються в 2^{n-1} раз і мають бути витримані з високою точністю. Наприклад, для 12-ти розрядного ЦАП використання в старшому розряді резистора з опором 10 кОм зажадає включення в молодший розряд резистора з опором близько 20 МОм. Широкий набір номіналів резисторів і вимоги їх високої точності, особливо при

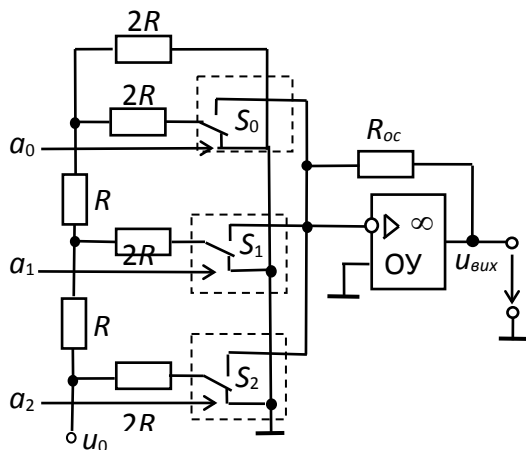


Рис. 6.2

значному числі розрядів n вхідного коду, створюють труднощі при реалізації ЦАП за допомогою інтегральної технології.

У іншій схемі ЦАП з матрицею R - $2R$ використовують резистори з двома номіналами опорів, причому резистори з опором R включені в кожен розряд (дивись мал. 6.2 при $n = 3$). Проте в цій схемі збільшується внесок паразитних ємностей.

Принцип функціонування схеми полягає у властивості резисторного дільника R - $2R$ зберігати постійний опір навантаження для джерела опірної напруги при замиканні

ключів. Внаслідок цього на резистор R , починаючи із старшого $n - 1$ розряду, опірна напруга послідовно ділиться навпіл, як і струм, що входить в кожен вузол матриці. При цьому напруга на виході перетворювача з матрицею R - $2R$:

$$u_{вих} = -u_0 \frac{R_{oc}}{R} (a_{n-1}2^{-1} + a_{n-2}2^{-2} + \dots + a_12^{-(n-1)} + a_02^{-n}) = -u_0 \frac{R_{oc}}{2^n R} \sum_{i=0}^{n-1} a_i 2^i.$$

Таким чином, вихідна напруга ЦАП пропорційно сумі напруги, зумовленими перемикачами, підключеними до джерела опірної напруги u_0 .

Недоліком ЦАП з матрицею R - $2R$ є сильний вплив на точність перетворення нестабільності опорів перемикачів у замкнутому стані, що знижує часову і температурну стабільність характеристик ЦАП. Цей недолік значною мірою вдається усунути в схемах код-напруга, виконаних на базі напівпровідникової

технології з використанням тонкоплівкових резисторів на кристалі і перемикачів на КМДП-транзисторах, в яких нелінійність від $\pm 0,8\%$ до $\pm 0,003\%$ від опірної напруги u_o , час встановлення струму від 5 мкс до десятих долей мікросекунд і менше, часто вихідний діапазон напруги ± 5 В. Опірна напруга в схемах ЦАП може обертися різної полярності або бути двуполярною.

2. ОСНОВНІ ПАРАМЕТРИ ЦАП

Основними параметрами ЦАП є число розрядів $n = 8-24$ і *абсолютна роздільна здатність* – середнє значення мінімальної зміни сигналу на виході ЦАП, зумовлене збільшенням або зменшенням його коду на одиницю.

Теоретично цифро-аналоговий перетворювач, який перетворює n -розрядні двійкові коди, повинен забезпечити 2^n різних значень вихідного сигналу з роздільною здатністю $1/(2^n-1)$. При числі розрядів $n = 8$ кількість незалежних квантів (східців) вихідної напруги ЦАП рівне $2^8 - 1 = 255$, при $n = 12$, $2^{12} - 1 = 4095$...

Абсолютне значення мінімального кванта напруги визначається як граничним, що приймається числом $2^n - 1$, так і максимальною вихідною напругою ЦАП, що називається також напругою шкали або опірною напругою u_o . Значення абсолютної роздільної здатності ЦАП, часто ЗМР (значення молодшого розряду), що означає, при $n = 8$ і опірній напрузі $u_o = 5$ В

$$\text{ЗМР} = u_o / (2^8 - 1) = 5 / 255 \approx 0,0196 \text{ В} = 19,6 \text{ мВ}.$$

Відмінність реального значення роздільної здатності від теоретичного обумовлено похибками і шумами вузлів, що входять в ЦАП. Точність ЦАП визначається значенням абсолютної погрішності δ_a і нелінійністю перетворювача δ_n . *Абсолютна погрішність* δ_a характеризується відхиленням максимального значення вихідної напруги u_{max} від розрахункового, такого, що відповідає кінцевій точці характеристики ідеального цифро-аналогово перетворювача, і вимірюється в одиницях ЗМР.

Нелінійність перетворювача δ_n характеризує відхилення дійсною характеристики від лінійної L (дивись мал. 6.1, а) проведеною через центри сходинок або через нуль і точку максимального значення вихідного сигналу.

З динамічних параметрів найбільш важливою є максимальна частота перетворення f_{max} (десятки і сотні кілогерц) - найбільша частота дискретизації, при якій параметри ЦАП відповідають заданим значенням.

Робота ЦАП часто супроводжується специфічними перехідними імпульсами у вихідному сигналі, що виникають через різницю часу відкривання і закривання аналогових перемикачів в ЦАП. Особливо значні викиди помітні коли вхідний код 01111 змінюється кодом 10000, а перемикач старшого розряду ЦАП відкривається пізніше, ніж закриваються перемикачі молодших розрядів. Внаслідок визначеної ідеалізації при моделюванні бібліотечних ЦАП середовища Multisim WorkBench не завжди вдається зафіксувати дані явища.

Бібліотечні інтегральні схеми ЦАП середовища Multisim WorkBench вимагають для своєї роботи підключення тільки постійної еталонної напруги, заземлення і вхідних сигналів.

НАВЧАЛЬНІ ЗАВДАННЯ І МЕТОДИЧНІ ВКАЗІВКИ ДО ЇХ ВИКОНАННЯ

Завдання 1. Відкрити файл або зібрати на робочому полі середовища Multisim WorkBench схему для дослідження інтегрального цифро-аналогового перетворювача (мал. 6.3, а) і встановити у діалогових вікнах компонентів їх параметри або режими роботи. Скопіювати схему (мал. 6.3, а) на сторінку звіту.

У схемі (мал. 6.3, а) використаний бібліотечний (Mixed) 8-ми розрядний цифро-аналоговий перетворювач DAC на входи якого подаються сформовані за допомогою перемикачів 0- 7 двійкові коди від 00000000 до 11111111₂ (FF₁₆ чи 255₁₀). Вихідну напругу ЦАП можна виміряти за допомогою вольтметра V1 або осцилографа XSC1 скориставшись візирними лініями на його екрані.

Завдання 2. Отримати на екрані осцилографа XSC1 ступінчасту вихідну

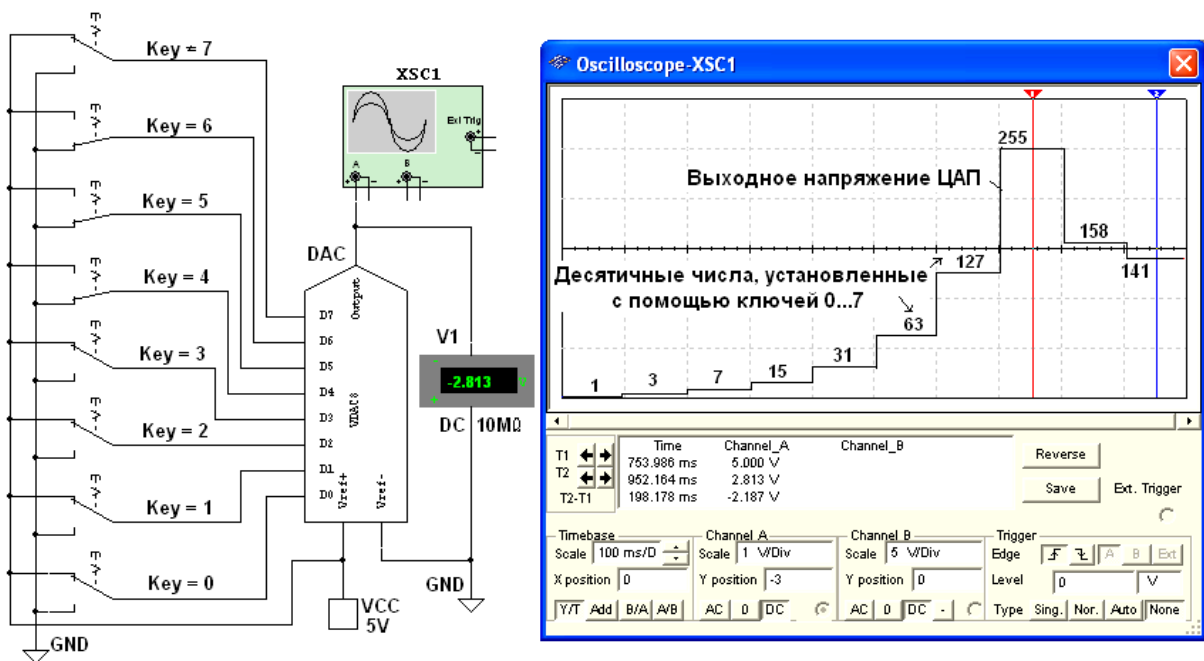


Рис. 6.3

напругу ЦАП (мал. 6.3, б). Для цього треба спочатку замкнути перемикач 0 і подати напруга 5 В на вхід D0 ЦАП, і запустити програму моделювання. На виході ЦАП формується напруга, рівна ЗМР. Потім в паузі моделювання замикає по черзі перемикачі 1, 2-7, подаючи вхідні десяткові комбінації 3, 7, 15, 31, 63, 127, 255 на входи D0- D7 ЦАП (мал. 6.3, б).

Повторити експеримент, подаючи на входи ЦАП сформовані за допомогою перемикачів шістнадцятиричні коди від 0 до FF (255₁₀) через крок 10₁₆ (16₁₀) і заносючи в таблицю 6.1 покази вольтметра V1 (значення вихідної напруги $u_{вих}$ ЦАП) при напрузі джерела VCC $u_o = 5$ В.

Знайти часткові і усереднене значення східців, часткові і усереднене значення ЗМР.

Побудувати графік $u_{вих}(N)$, вибравши відповідні масштаби для напруги і вхідних десяткових чисел N.

Таблиця 6.1

№ п/п	Вхідний десятковий код N	Вихідна напруга $u_{вих}В$	Напруга східців $u_{вих2} - u_{вих1}В$	Значення молодшого розряду $ZMP = (u_{вих2} - u_{вих1})/16, В$
1	0	0	0	–
2	15			
3	31			
4	47			
5	63			
6	79			
7	95			
8	111			
9	127			
10	143			
11	159			
12	175			
13	191			
14	207			
15	223			
16	239			
17	255			

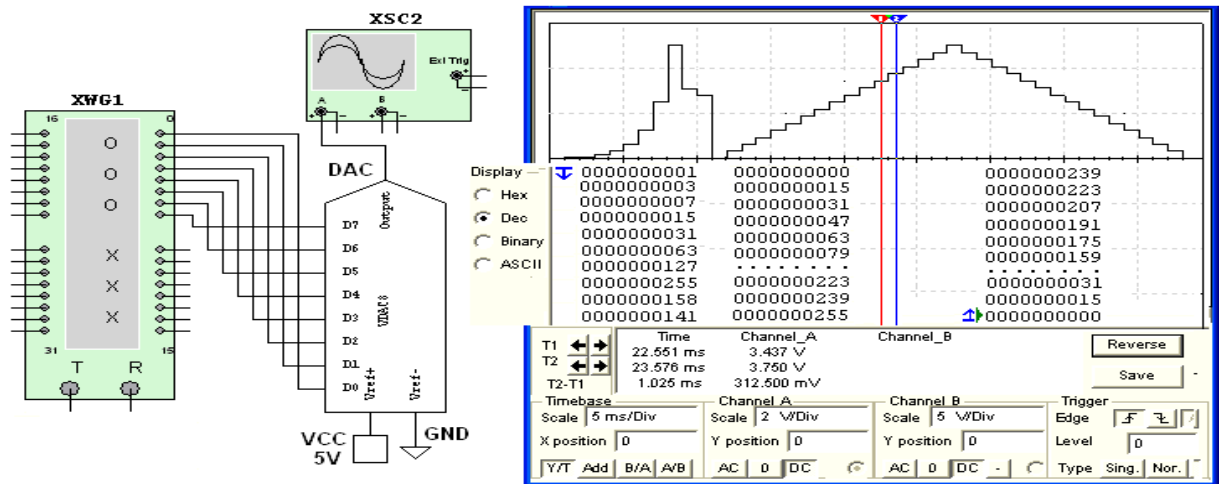
Завдання 3. Відкрити файл або зібрати на робочому полі середовища Multisim WorkBench схему для дослідження *цифро-аналогового перетворювача* (мал. 6.4, а) і встановити у діалогових вікнах компонентів їх параметри або режими роботи. Скопіювати схему (мал. 6.4, а) на сторінку звіту.

Провести моделювання ЦАП, запрограмувавши генератор XWG1 (частота генерації сигналів $f_2 = 1$ кГц) на зростання та зменшення шістнадцятиричних чисел від 0 до FF (255_{10}) при кроці 10_{16} (16_{10}).

Скласти таблицю і занести у неї вихідну напругу ЦАП і величину східців, які виводяться в нижньому вікні осцилографа XSC2.

Вимір напруги проводити за допомогою візирних ліній осцилографа, встановлюючи їх на двох сусідніх східцях (мал. 6.4, б) при різних кодових комбінаціях на виході генератора XWG1 і напрузі $u_o = 5$ В джерела VCC.

Так, при вхідних десяткових числах 175 і 191 і напрузі $u_o = 5$ В вихідна напруга ЦАП відповідно рівні 3,437 В і 3,750 В, а напруга східці – 312,5 мВ. При



Мал. 6.4

цьому $ZMP = 312,5/16 = 19,53$ В. Знайти і порівняти усереднене значення ЗМР з розрахунковим значенням.

Встановити напругу $u_o = 10$ В джерела VCC і **повторити** моделювання ЦАП при опірній напрузі 10 В.

Побудувати графіки $u_{вих}(N)$ при $u_o = 5$ В і $u_o = 10$ В на одному малюнку, обравши відповідні масштаби для напруг і вхідних десяткових чисел N , що відкладаються по відповідним осям координат.

ЗМІСТ ЗВІТУ

1. Найменування і мета роботи.
2. Перелік приладів, використаних в експериментах з їх короткими характеристиками.
3. Зображення електричних схем для дослідження цифро-аналогового перетворювача.
4. Копії графіків вихідної напруги досліджуваного ЦАП, що відображають його роботу.
5. Графіки $u_{вих}(N)$ при різних значеннях опірної напруги.
6. Висновки по роботі.

Лабораторна робота № 7

ДОСЛІДЖЕННЯ МУЛЬТИПЛЕКСОРА ТА ДЕШИФРАТОРА

МЕТА РОБОТИ:

Ознайомлення з основними характеристиками і дослідження інтегральних перетворювачів кодів (дешифратора, (шифратора), демультимплексора, (мультимплексора)).

ТЕОРЕТИЧНІ ВІДОМОСТІ ТА РОЗРАХУНКОВІ ФОРМУЛИ

Кодом називають систему символів для представлення інформації у формі, зручній для обробки, зберігання та передачі. У цифровій техніці для запису кодових символів, або просто коду, використовують дві цифри: 0 і 1. Перетворювачі кодів служать для переведення однієї форми бінарного числа (кової комбінації) в іншу, наприклад, перетворення двійково-десятькового коду в семисегментний код індикатора. Вхідні і вихідні коди перетворювачів пов'язані між собою. Цей зв'язок задають логічними функціями або у вигляді таблиці. Розглянемо найбільш поширені в цифровій техніці види перетворювачів кодів.

1. ДЕШИФРАТОР

Дешифратор (DC) або декодер - комбінаційна схема з n входами і $m = 2^n$ виходами ($m > n$), яка перетворює двійковий вхідний n -код (кодове слово) в унітарний. На одному з m виходів дешифратора з'являється логічна 1, а саме на той, номер якого відповідає поданим на вхід двійкового коду.

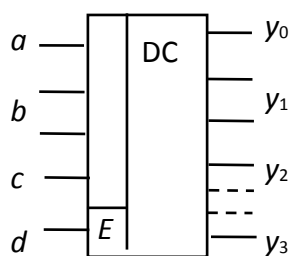


Рис. 7.1

На всіх інших виходах дешифратора вихідні сигнали дорівнюють нулю. Дешифратор використовують, коли потрібно звертатися до різних цифрових пристроїв за адресою, представленому двійковим кодом.

Умовне зображення дешифратора 4x16 (читається "чотири в шістнадцять") на схемах дано на рис. 7.1. Дешифратор містить число виходів, рівне числу комбінацій вхідних змінних: від $y_0 = \bar{a}\bar{b}\bar{c}\bar{d}$ до $y_{15} = abcd$ при $n = 4$ и $m = 2^n = 16$.

Застосовуються також неповні дешифратори з меншим числом виходів (10 або 12 при чотирьох змінних на вході, тоді ряд комбінацій на вході не використовується).

Кожен вихід повного дешифратора реалізує кон'юнкцію вхідних змінних (код адреси) або їх інверсій: при наборі $\bar{a}bcd(0111)$ $y_7 = 1$, при $abcd(1111)$ $y_{15} = 1$ і так далі.

Дешифратори часто мають *дозвільний* (керуючий) вхід E . При $E = 1$ дешифратор функціонує як зазвичай, при $E = 0$ на всіх виходах встановлюється 0 незалежно від вхідного коду адреси. Дешифратор широко використовують у багатьох пристроях, в тому числі в якості перетворювачів двійкового коду в десятиковий.

2. ШИФРАТОР

Шифратор (CD) або *кодер* виконує функцію, зворотну функції дешифратора. Умовне зображення шифратора 16×4 (16 в 4) на схемах показано на рис. 7.2, а. Класичний шифратор має n входів і m виходів ($m < n$), і при подачі сигналу 1 на один з входів (і не більше) на виході кодера з'являється двійковий код номера збудженого виходу. Число входів і виходів такого шифратора

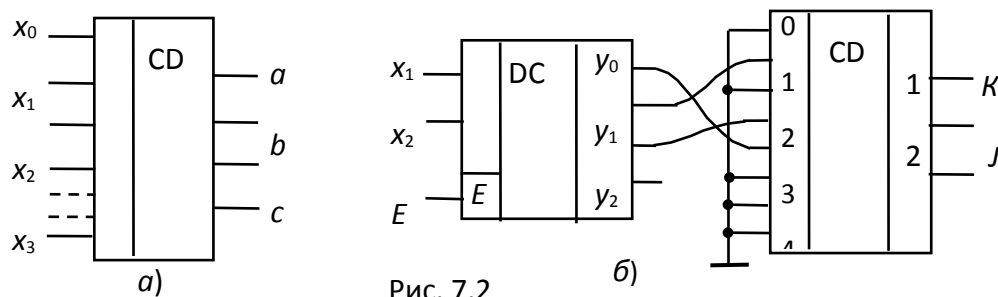


Рис. 7.2

пов'язано співвідношенням $n = 2^m$.

Області використання шифраторів – відображення у вигляді двійкового коду номера натиснутої кнопки або положення багатопозиційного перемикача, а також номера пристрою, який подав сигнал на обслуговування в мікропроцесорних системах. Шифратори входять до складу мікросхем контролерів переривань, наприклад КР580ВН59.

Для вирішення багатьох конкретних завдань необхідно синтезувати перетворювачі різних кодів. Як приклад на рис. 7.2, б показано схема кодового перетворювача, що складається з пари декодер DC - кодер CD, що реалізує логіку роботи ($y = a + b\bar{c} + \bar{b}c$) деякого світлофора з кольорами K, J і Z , керованого двохрозрядним двійковим кодом X . При цьому спочатку дешифрується кожна комбінація вхідного коду, в результаті чого на відповідному виході декодера з'являється логічна 1. Потім цей логічний сигнал, значення якого визначено номером виходу декодера, подається на кодер і на його виходах встановлюється перетворений код.

Число входів дешифратора DC дорівнює двом (x_1 та x_2), число виходів – трьом (числу виходів перетворювача) y_0 , y_1 та y_2 . З'єднання дешифратора і шифратора виконані відповідно до заданої логічної функцією y . Частина виходів декодера і входів кодера не використовується.

Ефективно стикаються один з одним декодер і кодер, побудовані на елементах І-НЕ: перший має інверсні виходи, а другий – інверсні входи. Якщо деяким вхідним комбінаціям відповідає одна і та ж вихідна, то відповідні виходи декодера об'єднують на елементі АБО і вихід останнього подають на потрібний вхід кодера.

Проектування кодової перетворювальної схеми на парі декодер-кодер виявляється в середньому більш вигідним і по числу корпусів, і по швидкодії, ніж при проектуванні з готових базових логічних мікросхем І-НЕ і АБО-НЕ. Однак споживана потужність в цьому випадку може виявитися більше, ніж у схеми з окремих елементів. Витрати часу інженера на логічне проектування по схемі декодер-кодер незрівнянно менше, ніж витрати на проектування перетворювача з логічних мікросхем низького ступеня інтеграції

3. МУЛЬТИПЛЕКСОР

Мультиплексор (MS) – це функціональна одиниця, що здійснює підключення (комутацію) одного з декількох входів до виходу y . На вихід такого пристрою передається логічний рівень того інформаційного розряду, номер якого в двійковому коді заданий на адресних входах x_1 і x_2 . Умовне зображення мультиплексора на чотири входи і можливий варіант його структурної схеми

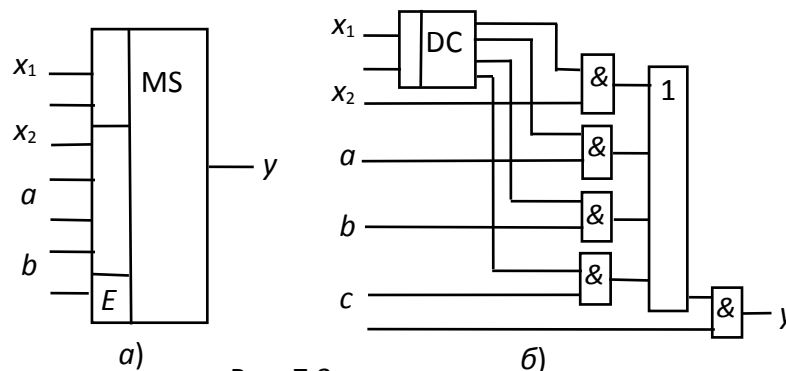


Рис. 7.3

показані на рис. 7.3, а і б.

При $x_1 = 0$ і $x_2 = 0$, $y = a$; при $x_1 = 0$ і $x_2 = 1$, $y = b$; при $x_1 = 1$ і $x_2 = 0$, $y = c$ і при $x_1 = 1$ і $x_2 = 1$, $y = d$.

Функціонування мультиплексора описується виразом

$$y = a\bar{x}_1\bar{x}_2 + b\bar{x}_1x_2 + cx_1\bar{x}_2 + dx_1x_2.$$

Вхід E - дозволяє: при $E = 1$ мультиплексор працює як зазвичай, при $E = 0$ вихід вузла знаходиться в неактивному стані, мультиплексор замкнений. Серійні вузли випускаються з числом адресних входів $n = 2, 3$ і 4 при можливому числі 2^n комутуваних входів. При необхідності комутувати більшу кількість входів використовують кілька мультиплексорів. Мультиплексори знаходять широке застосування в пристроях відображення інформації в різних пристроях управління.

Так як мультиплексор може пропустити на вихід сигнал з будь-якого інформаційного входу, адреса якого встановлена на відповідних адресних входах, то на основі мультиплексорів реалізують логічні функції, подавши на інформаційні входи логічні 1 або 0 відповідно до таблиці комутації, а на адресні входи – аргументи функції.

4. ДЕМУЛЬТИПЛЕКСОР

Демультимплексор (DMS) виконує функцію, зворотну функції мультиплексора – здійснює комутацію одного вхідного сигналу на 2^n виходів, де n – число адресних входів x_i . Він здійснює перетворення інформації з послідовної форми (послідовно-паралельної) в паралельну. Демультимплексор має один інформаційний вхід D і кілька виходів, причому вхід підключається до виходу y_i , має задану адресу.

Як приклад на рис. 7.4, а дано умовне графічне позначення демультимплексора, що має чотири виходи, закон функціонування якого заданий (таб. 7.1). Користуючись таб. 7.1, запишемо комутаційні функції для виходу пристрою:

$$y_0 = D\bar{x}_1\bar{x}_2; y_1 = D\bar{x}_1x_2; y_2 = Dx_1\bar{x}_2; y_3 = Dx_1x_2.$$

Функціональна схема демультимплексора, що реалізує ці вирази, наведена на рис. 7.4, б.

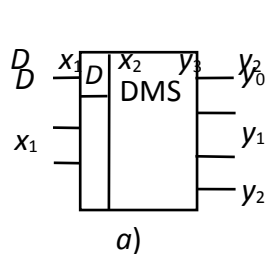
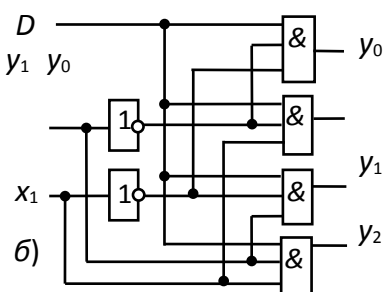


Рис. 7.4



Т а б л и ц я 7.1

1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Якщо загальне число виходів проєктованого пристрою перевищує наявні в інтегральних мікросхемах, то використовують паралельне підключення декількох схем.

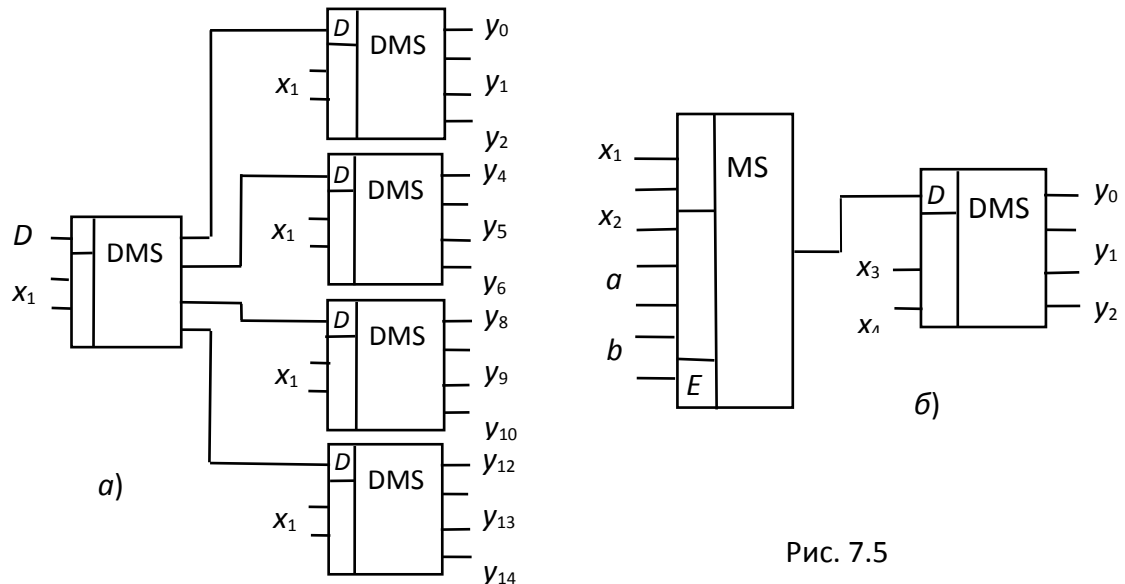


Рис. 7.5

На рис. 7.5, а показано демультимплексорне дерево, побудоване на мультимплексорах з чотирма виходами. Об'єднуючи мультимплексор з демультимплексором, отримують комбінаційний пристрій, в якому по заданих адресах один з входів підключається до одного з його виходів (рис. 7.5, б).

НАВЧАЛЬНІ ЗАВДАННЯ І МЕТОДИЧНІ ВКАЗІВКИ ДО ЇХ ВИКОНАННЯ

Завдання 1. Запустити комплекс Multisim Electronics Workbench. На робочому полі віртуальної лабораторії зібрати схему для дослідження *дешифратора DC* (рис. 7.6) або відкрити відповідний файл і встановити у діалогових вікнах компонентів їх параметри або режими роботи. Скопіювати схему (мал. 7.6) на сторінку звіту.

Схема (рис. 7.6) містить:

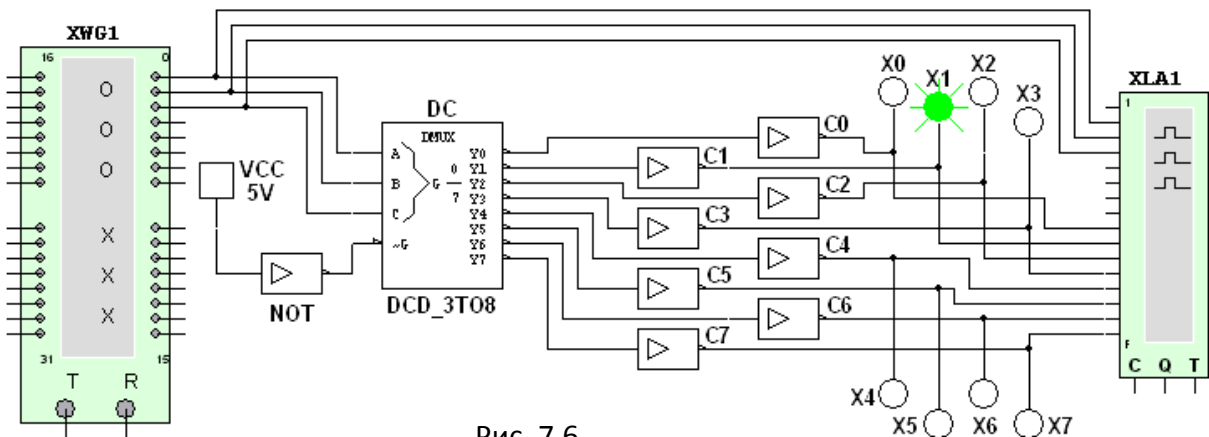


Рис. 7.6

- інтегральний дешифратор DC (decoder) 3x8, який має 3 інформаційні входи A, B і C (для кода 4-2-1), 8 виходів (Y0 - Y7) і перетворює позиційний 3-розрядний двійковий код в унітарний "1 з 8": у вихідній 8-розрядній кодовій комбінації тільки одна позиція зайнята одиницею, а всі інші - нульові (див. рис. 7.7, праворуч). Залежно від вхідного двійкового коду, наприклад 001, на виході DC з'являється сигнал 1 тільки на одній (другий, див., рис. 7.6) з 8-ми вихідних ліній, до яких підключені пробники X0 - X7.

Даний тип шифратора відноситься до Шифратора з різним рівнем вхідних і вихідних сигналів: активні вхідні рівні відповідають рівню логічної 1, а активні вихідні сигнали - рівню логічного 0. Для отримання активних вихідних рівнів, рівних 1, до виходів дешифратора підключено вісім інверторів C0 - C7;

- логічний генератор слова XWG1 ($f_2 = 500$ кГц) із записаними логічними словами в його комірки пам'яті, які еквівалентні десятковим числам від 0 до 7 (див. рис. 7.7, зліва);

- логічний аналізатор **XLA1**, на екран якого виводяться часові діаграми як трьох вхідних (**A, B, C**), так и восьми (**Y0, Y1 - Y7**) вихідних сигналів при покроковому режимі Step генератора **XWG1**;

- джерело VCC, напруга 5 В з виходу якого подано на інвертор **NOT**. Логічний 0 з інвертора подається на керуючий вхід дешифратора DC: при $\overline{G} = 0$ дешифратор знаходиться в активному стані.

Запустити програму моделювання дешифратора. Клацаючи мишею на кнопці **Step** генератора **XWG1**, послідовно подавати на вхід дешифратора логічні слова. Переконатися, що при подачі на вхід дешифратора кожної нової двійкової кодової комбінації засвічується тільки один пробник, який "розпізнає"

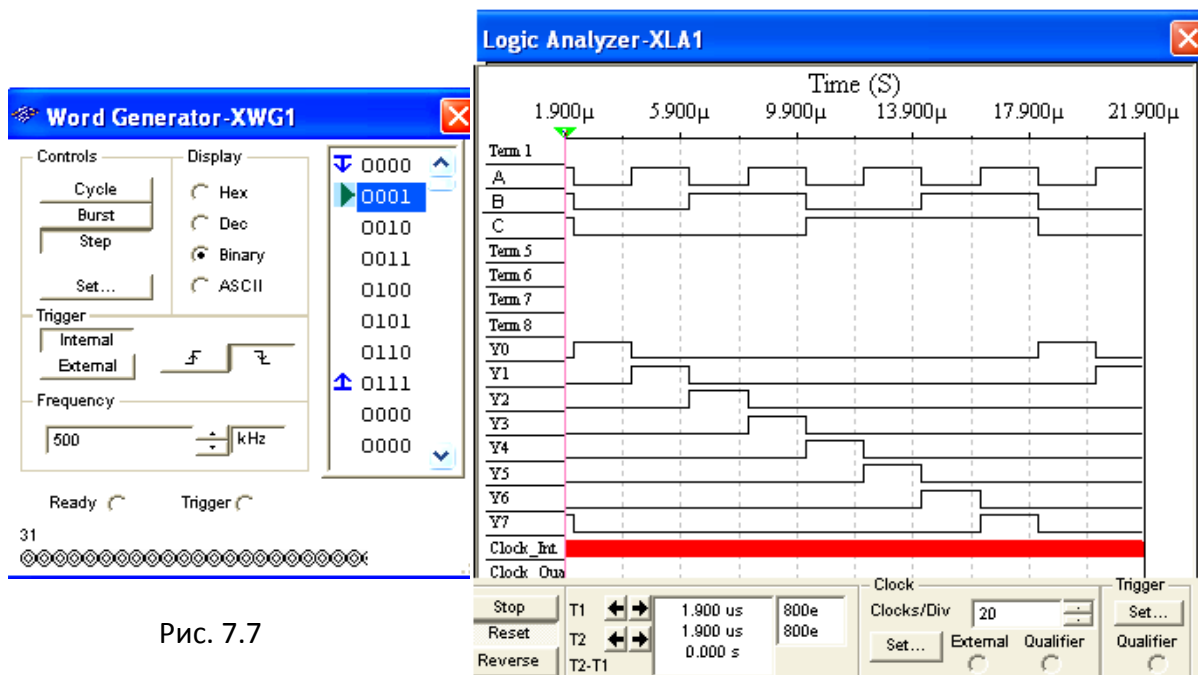


Рис. 7.7

свій вхідний код.

Скопіювати часові діаграми вхідних і вихідних сигналів дешифратора на сторінку звіту. За результатами моделювання скласти і заповнити таблицю перемикань (функцій на виходах дешифратора **DC** 3x8).

Завдання 2. Зібрати на робочому полі середовища Multisim Electronics Workbench схему для дослідження шифратора CD (рис. 7.8) і встановити в діалогових вікнах компонентів їх параметри або режими роботи.

Скопіювати схему (рис. 7.8) на сторінку звіту.

Інтегральний шифратор **CD** 8x3 (з 8 в 3) має 8 входів **D0, D1- D7**, підключених до виходів **Y0, Y1- Y7** дешифратора **DC**, і три інверсних виходи **A0, A1, A2**, до яких через інвертори **C0, C1, C2** підключені логічні пробники **X0,**

X1, X2 і семисегментний індикатор **Ind**. Вміст комірок пам'яті генератора слова **XWG1**: 000, 001, ..., 111 (див. Рис. 7.7, зліва).

Запустити програму моделювання шифратора. Клацаючи мишею на кнопці **Step** генератора **XWG1**, послідовно подавати на вхід дешифратора логічні слова. Переконайтеся, що при подачі з виходу **DC** на вхід шифратора **CD** 8-розрядної послідовності, в якій тільки одна позиція зайнята одиницею, а решта - нулями, на виході шифратора формуються 3-розрядні двійкові коди **A0A1A2**, де **A0 = A**, **A1 = B** і **A2 = C**, що відповідають двійковим кодовим комбінаціям на вході дешифратора **DC**.

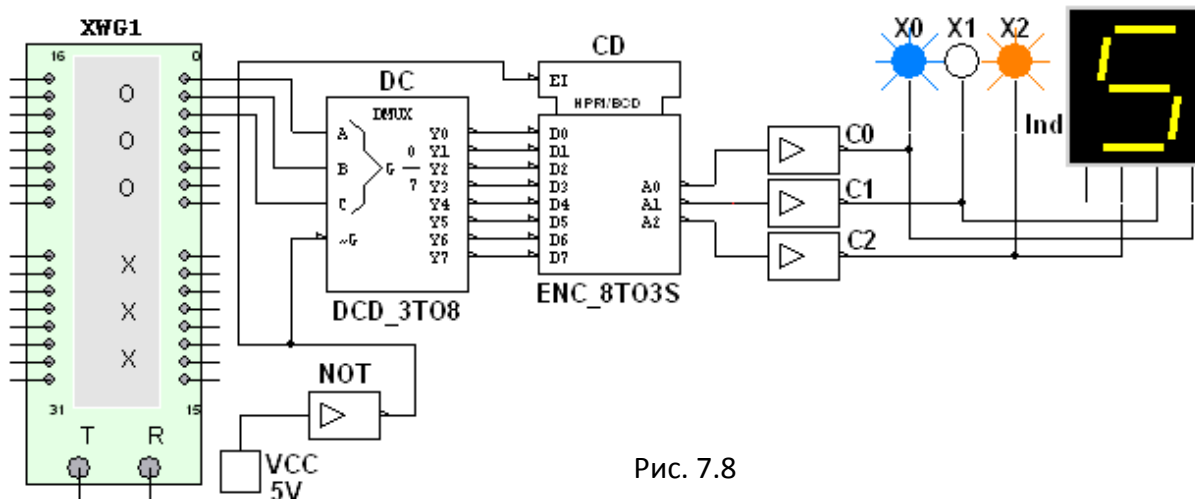


Рис. 7.8

За результатами моделювання (по спалахам логічних пробників **X0, X1, X2** і показами індикатора **Ind**) **скласти** і заповнити таблицю перемикань на виході шифратора **CD** 8x3.

Перетворити схему дешифратора **DC** 3x8 і шифратора **CD** 8x3 (див. Рис. 7.8) в схему **DC** 2x4 і шифратора **CD** 4x2, від'єднавши провід **C**, підключений до дешифратора, і провід **A2** з виходу шифратора, і **скласти** таблиці перемикань дешифратора 2x4 і шифратора 4x2.

Завдання 3. Зібрати на робочому полі середовища Multisim Electronics Workbench схему для дослідження демультимплексор **DMS** (рис. 7.9) і встановити в діалогових вікнах компонентів їх параметри або режими роботи.

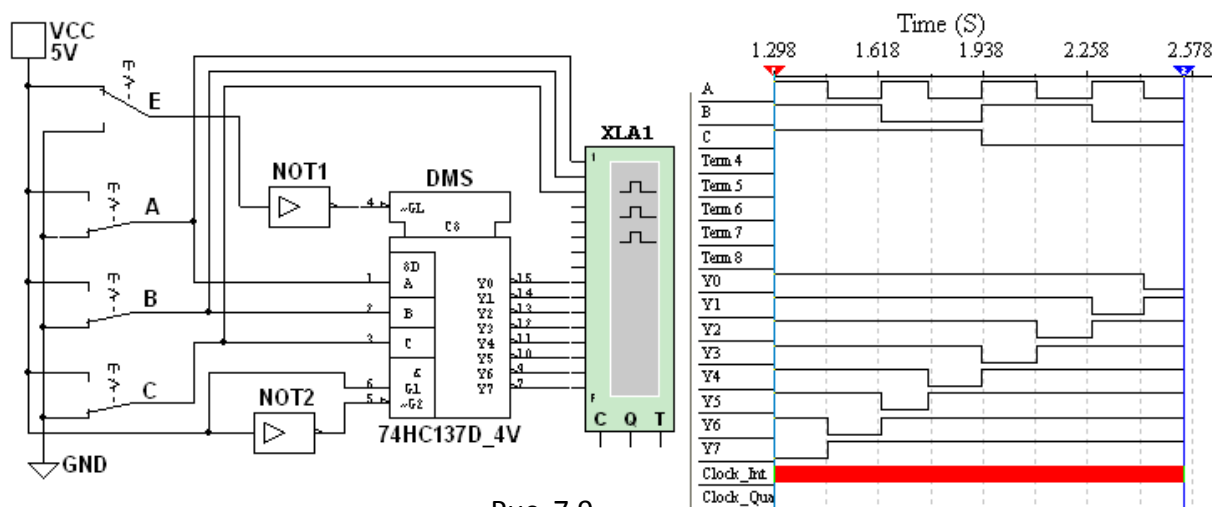


Рис. 7.9

Демультимплексор **DMS** 1x8 (із 1 в 8) (рис. 7.9) має один інформаційний вхід (з активними високим **G1** і низьким **G2** рівнями), три адресних **A**, **B**, **C** входу, дозвільний **GL** вхід з активним низьким рівнем і вісім **Y0**, **Y1-Y7** інверсних виходів, з'єднаних з входами логічного аналізатора **XLA1**. На вхід аналізатора також подаються сигнали з адресних входів **A**, **B**, **C**. За допомогою ключів **A**, **B** і **C** можна сформувати вісім трьохрозрядних довійкових адресних слів. При послідовній подачі зформованих ключами адресних слів від 111 до 000 на екран аналізатора **XLA1** при моделюванні виводяться 8-розрядні кодові послідовності з одним активним (низьким) рівнем.

Для забезпечення повільного переміщення променів на екрані аналізатора **XLA1** встановити частоту його таймера $f_a = 500$ Гц і число імпульсів, що припадають на одну поділку, $\text{Clocs} / \text{div} = 80$.

Задати код ключів 111 і клацнути мишею на кнопці **Run/Stop**. Криві адресних і вихідних логічних сигналів повільно розгортаються в часі на екрані аналізатора.

Зупинити (клацанням миші на кнопці **Stop**) процес моделювання при наближенні променів аналізатора до лінії розмітки екрану.

Повторювати перераховані вище операції для спадаючих рахункових комбінацій адресних сигналів (з 110 до 000) до тих пір, поки не буде записаний процес моделювання при адресному слові 000 (див. Рис. 7.9, праворуч).

Переконатися, що для кожної комбінації адресних сигналів демультимплексор формує логічний 0 на одному з восьми виходів, номер якого

відповідає певному кодовим словом на вході, тобто, демультиплексор подібний до комутатора, за допомогою якого потік цифрової інформації розділяється на 8 вихідних потоків.

Скопіювати схему (рис. 7.9) і часові діаграми вхідних і вихідних сигналів на сторінку звіту.

Якщо адресні входи **A**, **B** і **C** прийняти в якості інформаційних входів, а вхід **G1** (**G2**) в якості дозвольного входу, то мультиплексор перетвориться в дешифратор.

Завдання 4. Зібрати на робочому полі середовища Multisim Electronics Workbench схему для дослідження демультиплексора **DMS 1x16** (з 1 в 16) (рис. 7.10) і встановити в діалогових вікнах компонентів їх параметри або режими роботи. **Скопіювати** схему (рис. 7.10) в звіт.

З метою автоматизації процесу моделювання до входу демультиплексор **DMS** підключити логічний генератор **XWG1** з записаними в його комірки пам'яті адресними кодами від 0000 до 1111, а для візуалізації сигналів на виходах включити 16 логічних пробників **X1, X2-X16** і логічний аналізатор **XLA2**.

Запустити програму моделювання демультиплексора **DMS 1x16**. Послідовно подавати (клацаючи мишею на кнопки **Step** генератора **XWG1**) на вхід демультиплексора логічні слова, починаючи з комбінації 0000 адресного сигналу і закінчуючи комбінацією 1111, і спостерігати за змінами вихідних сигналів по показам індикаторів і в вікні аналізатора **XLA2**.

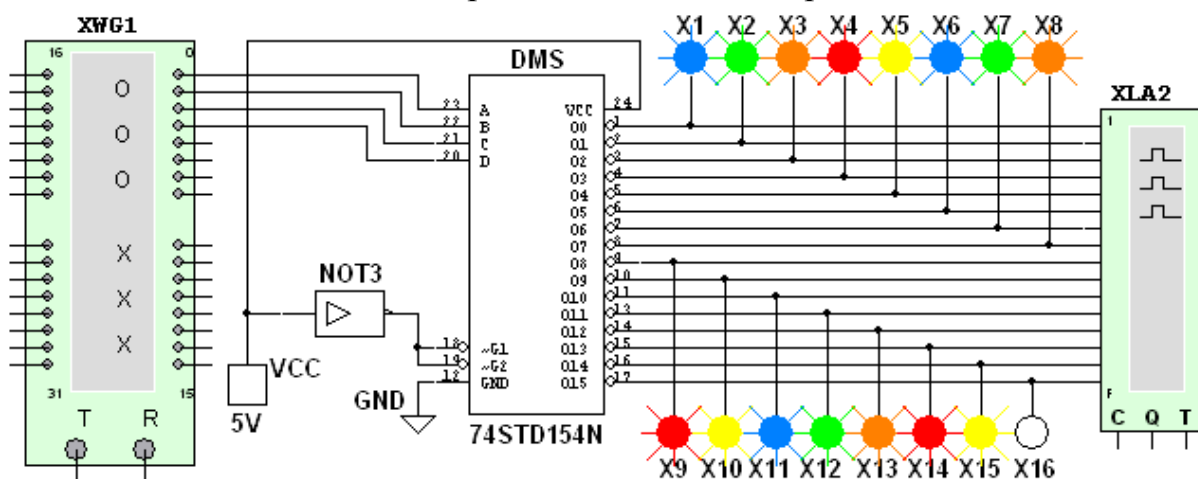


Рис. 7.10

У моделі демультимплексор що досліджується відповідний активний вихід має низький логічний рівень (рис. 7.11), тому пробник на цьому виході не світиться. Так, при подачі останньої кодової комбінації 1111 на вхід демультимплексора не светится пробник **X16**, так як активним є вихід **15** (див. рис. 7.10).

Скопіювати на сторінку звіту часові діаграми вихідних сигналів демультимплексора **DMS 1x16**.

Примітка. Демультимплексори як окремі пристрої промисловістю не випускаються, оскільки режим мультиплексування може бути реалізований як окремий випадок в інших пристроях - в дешифратор.

Завдання 5. Зібрати на робочому полі середовища Multisim Electronics Workbench схему (рис. 7.12) для дослідження мультиплексора **MS 8x1** (з 8 в 1) і

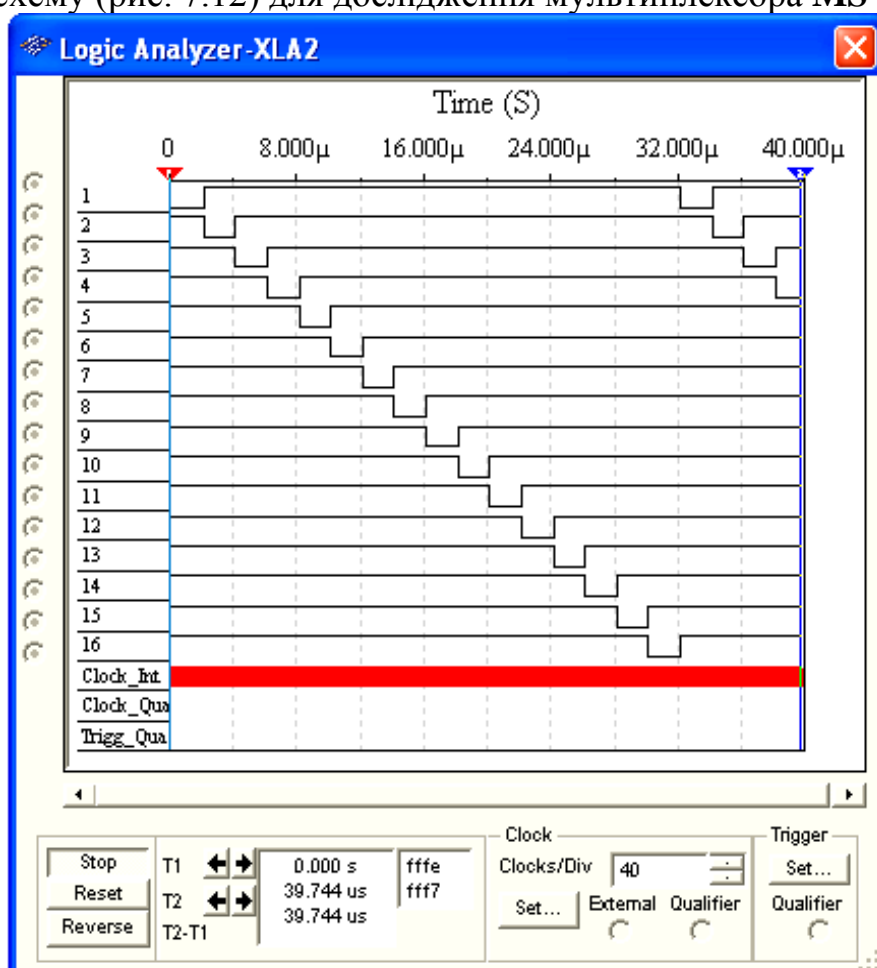


Рис. 7.11

встановити в діалогових вікнах компонентів їх параметри або режими

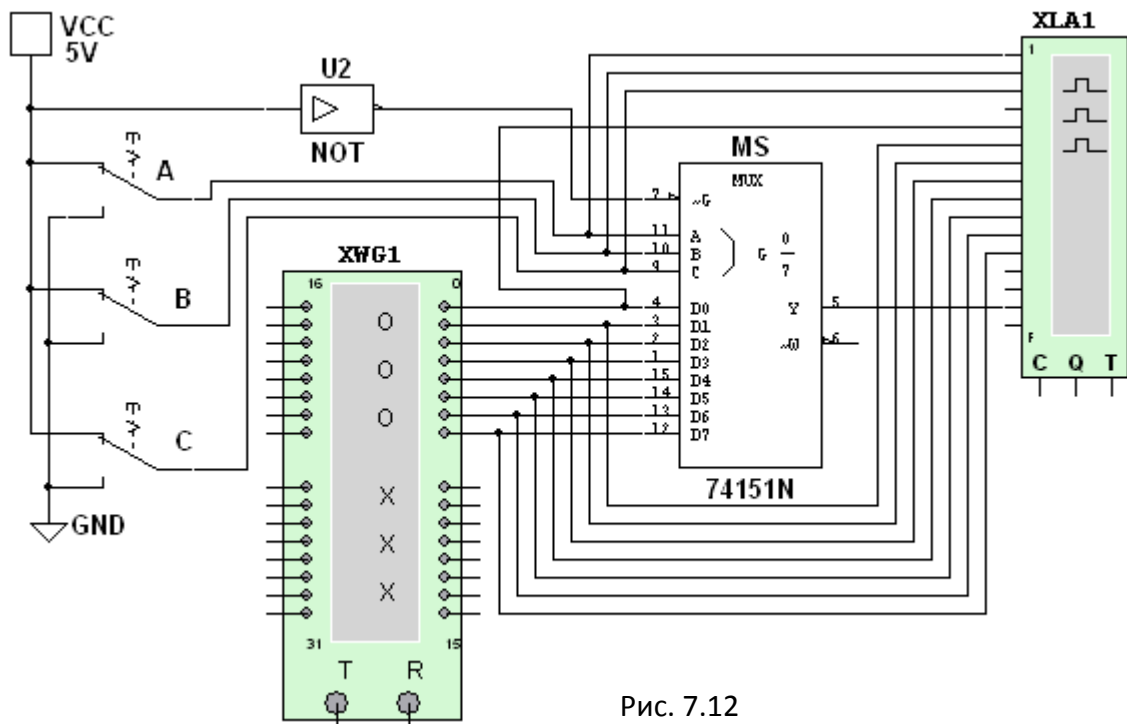


Рис. 7.12

роботи. Скопіювати схему (рис. 7.12) на сторінку звіту.

Мультиплексор **MS** з дозвільним входом **G** здійснює передачу сигналу з кожного інформаційного входу **D0, D1-D7**, заданійного 3-х розрядним кодом **ABC** - адресою обраного входу, на єдиний вихід **Y**. Розрядність (3) керуючого сигналу визначає кількість входів ($2^3 = 8$), з яких мультиплексор може приймати інформацію. Якщо припустити, що до входів **D0, D1-D7** мультиплексора **MS** приєднано 8 джерел цифрових сигналів - генераторів послідовних двійкових слів, то байти від будь-якого з них можна передавати на вихід **Y**.

Для ілюстрації роботи мультиплексора **MS** запишемо в комірки пам'яті генератора **XWG1** довільні 8-розрядні кодові слова (рис. 7.13, ліворуч), а за допомогою ключів **A, B, C** сформуємо керуючий сигнал 111. Послідовно клацаючи мишею на кнопки **Step** генератора **XWG1** і при **G = 1**, що надходять на вхід **D7** мультиплексора байти (сигнал 01001110) з 8-го розряду (на рис. 7.13, ліворуч 8-й розряд показаний стрілкою) логічних слів генератора **XWG1** передаються на вихід **Y** і на вхід аналізатора (див. рис. 7.13, праворуч).

Якщо ключ **A** встановити в нижнє положення (сформувавши, тим самим, адресний код 011), то з входу **D3** на вихід **Y** мультиплексора будуть надходити байти 4-го розряду логічних слів, записаних в комірці пам'яті генератора **XWG1**, і так далі.

Записати в перші вісім комірок пам'яті генератора **XWG1** довільні 8-розрядні кодові слова, задати частоту $f_T = 500$ кГц і режим **Step** його роботи (див. Рис. 7.13, зліва).

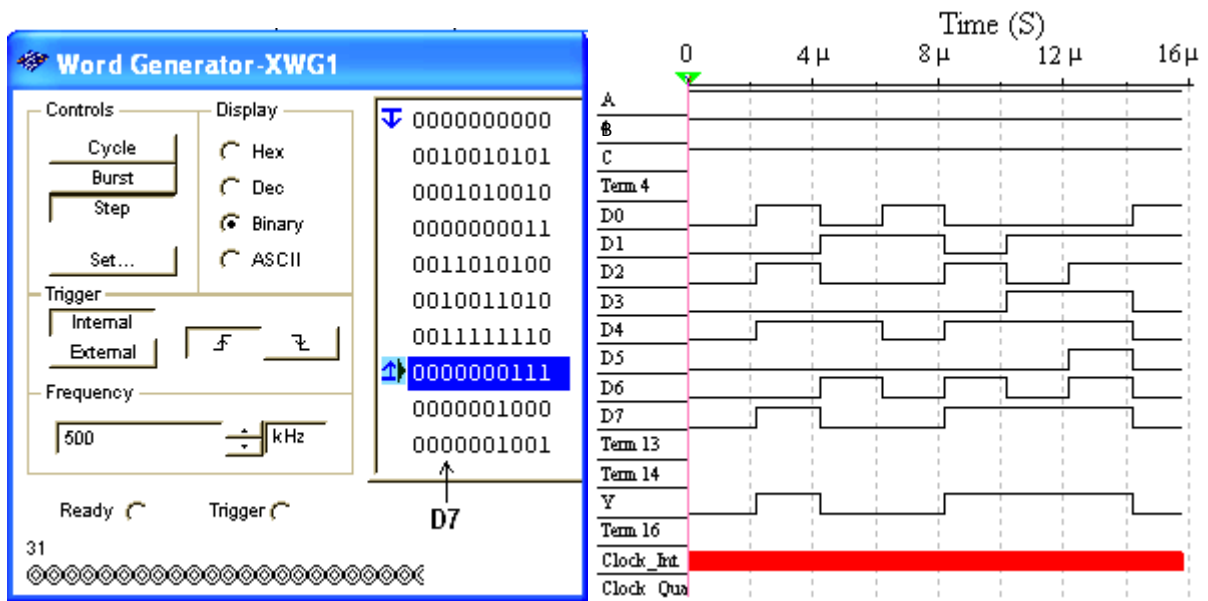


Рис. 7.13

Встановити частоту $f_a = 20$ МГц таймера логічного аналізатора **XLA1** і кількість імпульсів таймера **Clock/div = 20**, що припадають на одну поділку.

Встановити за допомогою ключів **A**, **B** і **C** адресний код (самостійно або за вказівкою викладача), наприклад 100_2 (4_{10}) і і запустити програму моделювання мультиплексора. Отримати і скопіювати часові діаграми вхідних сигналів **D0**, **D1-D7** і вихідного сигналу **Y** мультиплексора на сторінку звіту.

Примітка. Таблиці перемикань на виходах для розглянутих бібліотечних перетворювачів кодів можна викликати натисканням клавіші допомоги F1 після виділення на схемі відповідного перетворювача.

ЗМІСТ ЗВІТУ

1. Найменування і мета роботи.
2. Перелік приладів, використаних в експериментах, з їх короткими характеристиками.
3. Зображення електричних схем для дослідження цифро-аналогового перетворювача.
4. Копії часових діаграм і таблиці перемикань, що відображають роботу досліджуваних перетворювачів кодів.
6. Висновки по роботі.

РЕКОМЕНДОВАНІ ЛІТЕРАТУРНІ ДЖЕРЕЛА ТА ДОДАТКИ

ЛІТЕРАТУРНІ ДЖЕРЕЛА

1. Колонтаєвський Ю.П., Сосков А.Г. Електроніка і мікросхемотехніка: Підручник. 2-е вид. / За ред. А.Г. Соскова. -К.: Каравела, 2009.-416 с.
2. Міліх В.І., Шавьолкін О.О. Електротехніка, електроніка та мікропроцесорна техніка: Підручник. За ред. В.І.Міліх. 2-е вид. – К.: Каравела, 2012. -688 с.
3. Колесников В.Г. (глав. ред.) - Электроника. Энциклопедический словарь, - М.: Сов. Энциклопедия, 1991. – 688 с.
4. Multisim 7 современная система компьютерного моделирования и анализа схем электронных устройств: Marc E. Herniter, 2011. -487 с.

ДОДАТКИ

QR-посилання на файли Multisim Electronics Workbench, що необхідні для виконання лабораторних робіт:



ЗМІСТ

Передмова	3
Лабораторна робота №1 ЛОГІЧНІ ЕЛЕМЕНТИ І СХЕМИ	4
Лабораторна робота №2 ТРИГЕРИ	13
Лабораторна робота №3 ЛІЧИЛЬНИКИ	22
Лабораторна робота №4 РЕГІСТРИ	32
Лабораторна робота №5 АНАЛОГО-ЦИФРОВИЙ ПЕРЕТВОРЮВАЧ	40
Лабораторна робота №6 ЦИФРО-АНАЛОГОВИЙ ПЕРЕТВОРЮВАЧ	50
Лабораторна робота №7 МУЛЬТИПЛЕКСОРИ ТА ДЕШИФРОТОРИ	56
Рекомендовані літературні джерела та додатки	69

Формат 60x84/16. Ум. друк. арк. 3,95.
Тираж 100 прим. Зам. №1027

Видавництво «Аксиома».
вул. Симона Петлюри, 30а, м. Кам'янець-Подільський, 32300.
Відділ реалізації: тел. (067) 3812943; sales@aksioma.org.ua.
Загальні питання: тел./факс: (03849) 39006; info@aksioma.org.ua.
<http://aksioma.org.ua>
Свідоцтво суб'єкта видавничої справи ДК №1808 від 26.05.2004 р.