Міністерство освіти і науки України
Інженерний навчально-науковий інститут
ім. Ю. М. Потебні Запорізького національного університету

Л. Л. Верьовкін,
М. В. Світанько

ФУНКЦІОНАЛЬНІ ВУЗЛИ МІКРОПРОЦЕСОРНИХ СИСТЕМ

Методичні рекомендації до практичних занять
для здобувачів ступеня вищої освіти бакалавра спеціальності
G5 «Електроніка, електронні комунікації, приладобудування та радіотехніка»
освітньо-професійної програми
«Мікро- та наносистемна техніка»

Затверджено
вченою радою ЗНУ
Протокол № від

Запоріжжя
2026

УДК 621.38(076)
В377

Верьовкін Л. Л., Світанько М. В. Функціональні вузли мікропроцесорних систем : методичні рекомендації до практичних занять для здобувачів ступеня вищої освіти бакалавра спеціальності G5 «Електроніка, електронні комунікації, приладобудування та радіотехніка» освітньо-професійної програми «Мікро- та наносистемна техніка». Запоріжжя : Запорізький національний університет, 2026. 51 с.

У виданні відповідно до силабуса дисципліни «Функціональні вузли мікропроцесорних систем» подано зміст практичних занять. Кожне практичне заняття містить короткі теоретичні відомості, варіанти індивідуальних завдань для самостійної роботи, а також контрольні запитання для перевірки знань.

Охарактеризовано методики розрахунку, аналізу та синтезу основних функціональних вузлів, які використовуються у мікропроцесорній та мікросистемній техніці. Особливу увагу приділено принципам інтеграції мікропроцесорів із сенсорами, актуаторами та іншими компонентами на мікро- та нанорівнях. Методичні рекомендації містять перелік використаної та рекомендованої літератури, інформаційні ресурси дисципліни «Функціональні вузли мікропроцесорних систем».

Для здобувачів ступеня вищої освіти бакалавра спеціальності G5 «Електроніка, електронні комунікації, приладобудування та радіотехніка» освітньо-професійної програми «Мікро- та наносистемна техніка».

Рецензент
В. Л. Коваленко, доктор технічних наук, професор кафедри електричної інженерії та кіберфізичних систем ІННІ ім. Ю. М. Потебні ЗНУ.

Відповідальний за випуск
Т. В. Критська, доктор технічних наук, професор, завідувачка кафедри електроніки, мікроелектронних систем та програмного забезпечення ІННІ ім. Ю. М. Потебні ЗНУ ІННІ ім. Ю. М. Потебні ЗНУ.

Зміст

Вступ ……... 4
Практичне заняття №1 Дослідження логічних операцій повного переліку функцій двох аргументів ……... 6
Практичне заняття №2 Логічний синтез цифрових автоматів за заданим алгоритмом ……. 8
Практичне заняття №3 Дослідження операцій АЛП і відповідних їм логічних та логіко-арифметичних функцій ……. 13
Практичне заняття №4 Використання дешифраторів для виконання логічних операцій …….. 17
Практичне заняття №5 Мультиплексори даних мікропроцесорних систем. 23
Практичне заняття №6 Збільшення розрядності демультиплексорів ……... 27
Практичне заняття №7 Логічний синтез пристрою двійкового множення на суматорах та логічних елементах …… 31
Практичне заняття №8 Розробка кодуючого пристрою на основі сканованої матриці клавіш …… 33
Практичне заняття №9 Використання постійних запам’ятовувальних пристроїв ……. 37
Практичне заняття №10 Проектування цифрового послідовністного пристрою на регістрі зсуву ……... 39
Практичне заняття №11 Пристрої послідовністного типу призначені для спрацювання в заданих кодах ……. 43
Практичне заняття №12 Пристрої послідовністного типу призначені для генерації сигналів ……... 47
Література, інформаційні джерела ……... 49

Вступ

Метою викладання дисципліни «Функціональні вузли мікропроцесорних систем» є формування повного уявлення про принципи функціонування та розрахунки складових вузлів мікроконтролерів, побудови мікропроцесорних систем та їх використання в електронних приладах.

Основними **завданнями** вивчення дисципліни «Функціональні вузли мікропроцесорних систем» є: отримання знань складових частин мікропроцесорних систем; набуття уміння практично використовувати сучасні програмні засоби для аналізу функціонування вузлів мікропроцесорних систем.

У результаті вивчення навчальної дисципліни студент повинен

знати:

– загальні принципи побудови мікропроцесорних систем;

– основні пристрої обробки інформації мікроконтролерів;

– запам'ятовувальні пристрої мікропроцесорних систем;

– пристрої послідовністного та комбінаційного типу.

вміти:

– розв’язувати спеціалізовані задачі мікропроцесорних систем;

– розраховувати схемотехніку оперативних пристроїв комбінаційного типу;

– розраховувати схемотехніку оперативних пристроїв;

– виконувати аналіз схем з метою їх удосконалення.

Згідно з вимогами освітньо-професійної програми студенти повинні досягти таких **компетентностей**:

– здатність застосовувати знання у практичних ситуаціях;

– навички використання інформаційних і комунікаційних технологій;

– здатність працювати автономно та в команді;

– здатність використовувати знання і розуміння наукових фактів, концепцій, теорій, принципів і методів для проектування та застосування мікро- та наносистемної техніки;

– здатність застосовувати існуючі та розробляти нові методи, методики, технології та процедури для вирішення інженерних завдань мікро- та наносистемної техніки;

– здатність використовувати математичні принципи і методи для проектування та застосування мікро- та наносистемної техніки;

– здатність застосовувати відповідні наукові та інженерні методи, сучасні інформаційні технології і комп'ютерне програмне забезпечення, комп'ютерні мережі, бази даних та Інтернет-ресурси для розв’язання професійних задач в галузі мікро- та наносистемної техніки;

– здатність застосовувати знання з моделювання функціональних вузлів мікроелектроніки та конструювання приладів на їх основі.

Методичні рекомендації містять практичні заняття з курсу «Функціональні вузли мікропроцесорних систем», відповідно до кожної теми, Практичні роботи виконуються на комп’ютері з відповідним програмним забезпеченням. Для підготовки до лекцій, практичних занять та самостійного

засвоєння матеріалу студенти можуть використати теоретичні відомості з курсу.

Виконання кожної практичної роботи складається з двох етапів.

1. Підготовка до практичного заняття, вивчення теоретичного матеріалу, виконання усіх розрахунків, складання плану практичного заняття.

2. Складання на комп'ютері робочої схеми із розрахованих і підібраних елементів, дослідження її відповідності до складеного плану, оформлення теоретичних і практичних результатів, аналіз результатів, оформлення звіту з практичної роботи.

У процесі підготовки до практичного заняття студент повинен чітко усвідомити собі кінцеву мету практичної роботи, форму, характер зміни вхідних і вихідних параметрів вузлів схеми, виконати необхідні розрахунки.

Звіт, підготовлений до захисту практичного заняття, повинен містити назву і мету роботи, план досліджень. розрахунки, досліджувані схеми і часові діаграми, які пояснюють їхню роботу, Оформлення звітів виконується відповідно до вимог ЄСКД до текстових документів.

Порядок, виконання розрахунків та досліджень у аудиторії:

1. Студент допускається до виконання чергової практичної роботи при наявності підготовленого, відповідно до приведених вище вимог, звіту.

2. Звіт попередньої роботи підписується викладачем після його подання на наступному практичному занятті після занесення до нього усіх розрахунків і аналізу.

3. Звіти усіх робіт зберігаються у студента до виконання останньої роботи у загальному журналі з відповідною кількістю сторінок.

4. Після проведення розрахунків і дозволу виконувати дослідження, студент на комп'ютері збирає схему і самостійно налагоджує її. У випадку, якщо схема не працює, або працює неправильно, він має знайти причину несправності та усунути її.

5. Налагодивши схему і перевіривши відповідність її вихідних сигналів розрахунковим даним, студент запрошує викладача для огляду і перевірки.

6. Після цього виконуються намічені дослідження та розрахунки. По закінченню результати надаються на перевірку викладачу.

7. До наступної практичної роботи остаточно оформлюється звіт, який надається викладачу.

Перед виконанням кожної практичної роботи викладач опитує студентів як по змісту самої роботи, так і за методикою її виконання.

Практичне заняття №1

Дослідження логічних операцій повного переліку функцій двох аргументів

Метою практичного заняття є ознайомлення з повним переліком операцій функцій двох аргументів.

План практичного заняття.

1. Вивчити логічні операції булевої алгебри.
2. Засвоїти методику визначення параметрів повного набору функцій двох аргументів.
3. Дослідити практично формування значень функцій повного набору двох аргументів.

Ключові терміни та поняття: логічні операції, операнди, заборона, імплікація, рівнозначність, «Виключне АБО», еквівалентність.

Методичні вказівки до вивчення питань плану.

Цифрові пристрої виконують арифметичні і логічні операції, при цьому використовується два класи змінних: числа і логічні змінні [1].

Числа несуть інформацію про кількісні характеристики системи; над ними здійснюються арифметичні дії.

Логічні змінні визначають стан системи або приналежність її до певного класу станів (комутація каналів, управління роботою ЕОМ за програмою і т. п.). Розділ математичної логіки, який вивчає зв'язки між логічними змінними, що мають лише два значення, називається алгеброю логіки. Алгебра логіки розроблена англійським математиком Джорджем Булем і часто називається булевою алгеброю. Алгебра логіки є теоретичною базою для побудови систем цифрової обробки інформації. Спочатку на основі законів алгебри логіки розробляється логічне рівняння пристрою, який дозволяє з'єднати логічні елементи так, щоб схема виконувала задану логічну функцію [1].

У загальному випадку логічні вирази є функціями логічних змінних A, B, C, кожна з яких може мати значення 0 або 1. Якщо є k логічних змінних, то вони утворюють 2^k можливих логічних наборів з 0 і 1. При k = 2: AB = 00, 01, 10, 11. Для кожного набору змінних логічна функція F може набувати значення 0 або 1. Тому для k змінних можна утворити $n_k = 2^{2^k}$ різних логічних функцій.

Алгебра логіки базується на декількох аксіомах, з яких виводять основні закони для перетворень з логічними змінними [1].

Використовуючи закони асоціативності, будь-яку логічну функцію багатьох змінних (k > 2) можна представити у вигляді комбінації функцій двох змінних. Повний набір $2^{2^2} = 16$ логічних функцій двох змінних приведений в таблиці 1.1. Кожна функція позначає одну з 16 можливих логічних операцій над двома змінними A, B і має власну назву і умовне позначення [1].

Таблиця 1.1 – Повний набір функцій двох аргументів

A 0 0 1 1 / B 0 1 0 1	Представлення через операції «І», «АБО», «НЕ»	Найменування функції
F_0 0 0 0 0	$F_0 = 0$	Константа 0
F_1 0 0 0 1	$F_1 = AB$	Кон'юнкція
F_2 0 0 1 0	$F_2 = A\overline{B}$	Заборона
F_3 0 0 1 1	$F_3 = A$	Тотожність
F_4 0 1 0 0	$F_4 = \overline{A}B$	Заборона
F_5 0 1 0 1	$F_5 = B$	Тотожність
F_6 0 1 1 0	$F_6 = A\overline{B} + \overline{A}B$	Виняткове АБО
F_7 0 1 1 1	$F_7 = A+B$	Диз'юнкція
F_8 1 0 0 0	$F_8 = \overline{A + B}$	АБО–НЕ, стрілка Пірса
F_9 1 0 0 1	$F_9 = AB + \overline{A}\overline{B}$	Еквівалентність
F_{10} 1 0 1 0	$F_{10} = \overline{B}$	Інверсія
F_{11} 1 0 1 1	$F_{11} = A + \overline{B}$	Імплікація від A до B
F_{12} 1 1 0 0	$F_{12} = \overline{A}$	Інверсія
F_{13} 1 1 0 1	$F_{13} = \overline{A} + B$	Імплікація від B до A
F_{14} 1 1 1 0	$F_{14} = \overline{AB}$	І-НЕ, штрих Шеффера
F_{15} 1 1 1 1	$F_{15} = 1$	Константа 1

Питання для самоконтролю.

1. Арифметичні та логічні операції алгебри логіки.
2. Функціональні залежності між логічними змінними.
3. Повний набір функцій однієї логічної змінної.
4. Аксіоми і закони булевої алгебри.
5. Повний набір функцій двох аргументів.

Практичні завдання, задачі.

1. Зібрати експериментальний стенд в програмному застосуванні Electronics Workbench (рис. 1.1).

2. Для отримання функцій F6 та F6 виконати перетворення згідно з рівнянням

$$F6 = A\overline{B} + \overline{A}B = A\overline{B} + \overline{\overline{A}B} = A\overline{B} + \overline{A + \overline{B}}$$

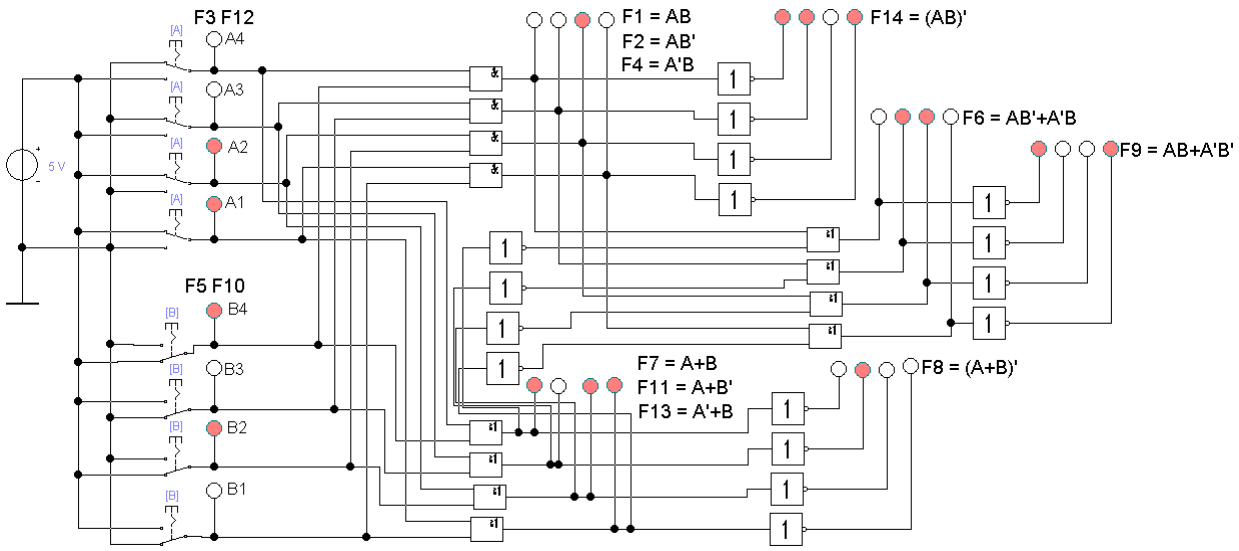


Рисунок 1.1 – Експериментальний стенд для дослідження повного набору функцій двох аргументів

3. Зафіксувати алгебраїчно отримання параметрів функцій згідно з операціями над аргументами A і B.

4. Зробити розгорнутий висновок за результатами досліджень.

Для отримання максимальної оцінки рішення пунктів завдання має бути виконане у повному обсязі, з відповідними поясненнями, оформлене стилістично і технічно грамотно.

Практичне заняття №2

Логічний синтез цифрових автоматів за заданим алгоритмом

Метою практичного заняття є вивчення методів синтезу складних практичних цифрових схем на підставі словесного опису їх роботи.

План практичного заняття.

1. Засвоїти методику виконання логічною синтезу складних практичних цифрових схем.
2. Засвоїти методику синтезу цифровий автоматів за заданим алгоритмом.
3. Провести аналіз функціонування цифрового автомату.

Ключові терміни та поняття: словесне завдання, таблиця істинності, рівняння функціонування, функціональні блоки, цифровий автомат.

Методичні вказівки до вивчення питань плану.

Дискретні пристрої як складові частини входять в різні системи управління, які використовуються у всіх областях техніки [2].

В даний час складність дискретних пристроїв настільки велика, що для успішного їх синтезу доцільно ділити пристрій на частини (функціональні блоки). Розбиття на функціональні блоки ведеться з тим розрахунком, аби

функції які реалізуються кожним блоком були нескладними, однозначно визначеними і по можливості легко контрольованими, що спрощує побудову всього пристрою в цілому, а також полегшує його експлуатацію (заміну блоків, виявлення несправностей, контроль функціонування і т. п.).

Необхідність розбиття на блоки витікає також з того факту, що в дискретних пристроях самого різного призначення деякі блоки часто повторюються і в 50 – 90% випадків це дозволяє зібрати довільний пристрій з декількох десятків так званих типових блоків із стандартними функціями.

Найбільш функціональними блоками комбінаційного типу, які часто повторюються в складних дискретних пристроях, є: перетворювачі кодів, комутатори, суматори, компаратори і різного роду контрольні схеми.

Кожен з цих блоків може бути реалізований в різних варіантах відповідно до різних завдань (описами роботи). Опис всіх вживаних схемних рішень неможливо виконати із-за великого числа варіантів, які відрізняються виглядом перетворюваної інформації, характером її представлення і використаною для реалізації елементною базою. Тому на практиці доцільно мати найбільш часто використовувані функціональні блоки, а їх модифікації отримувати шляхом вживання спеціальних методів синтезу.

Методика синтезу типових блоків єдина і складається з наступних основних етапів [3]:

– словесний опис роботи блоку;
– побудова таблиці істинності;
– обчислення вихідних функцій;
– мінімізація вихідних функцій;
– реалізація блоку в заданому елементному базисі.

Провести синтез цифрового автомата на логічних елементах, призначеного для аналізу істинності або хибності різних висловлювань.

Питання для самоконтролю.

1. Методика побудови цифрових комбінаційних блоків.
2. Основні закони і тотожності булевої алгебри.
3. Принцип спрощення аналітичних рівнянь.

Практичні завдання, задачі.

Словесне описання. Цифровий автомат імітує логічну гру «Перевізник». Завдання; перевізник повинен перевезти з одного берега річки на другий вовка, козу і капусту. Обмеженням являється те, що в човні можуть одночасно знаходитись тільки перевізник, або перевізник і одна з одиниць перевозу. Несумісні варіанти одночасного знаходження: вовк і коза; коза і капуста.

1. Скласти варіанти алгоритмів вирішення завдання.

Призначимо кодування фігурантів подій вирішення алгоритмів завдання: A – перевізник; B – вовк; C – коза; D – капуста. Завдання вирішується покроково (по тактах). Помилка в одному із тактів наводить до невірного результату.

2. Алгоритм 1. Перевізник перевозить козу (F1) → перевізник повертається один (F2) → перевізник перевозить капусту (F3) → перевізник повертається с козою (F4) → перевізник перевозить вовка (F5) → перевізник повертається один (F6) → перевізник перевозить козу (F7).

Представити Алгоритм 1 таблично і алгебраїчно (рис. 2.1).

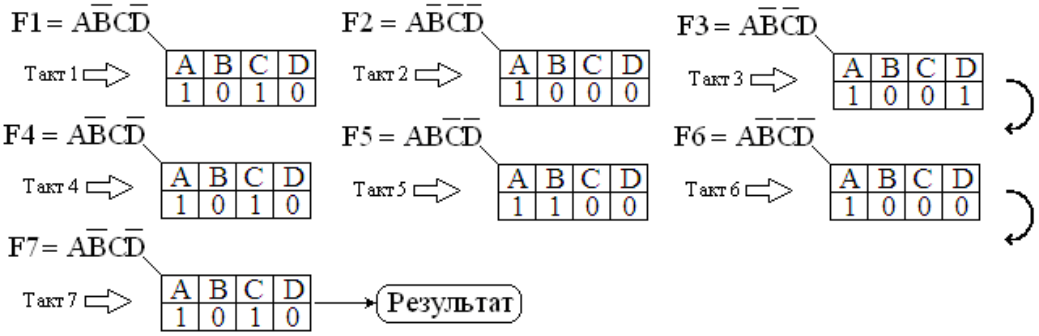


Рисунок 2.1 – Виконання завдання Алгоритму 1 по тактах

3. Алгоритм 2. Перевізник перевозить козу (F1) → перевізник повертається один (F2) → перевізник перевозить вовка (F3) → перевізник повертається с козою (F4) → перевізник перевозить капусту (F5) → перевізник повертається один (F6) → перевізник перевозить козу (F7).

Представити Алгоритм 2 таблично і алгебраїчно (рис. 2.2).

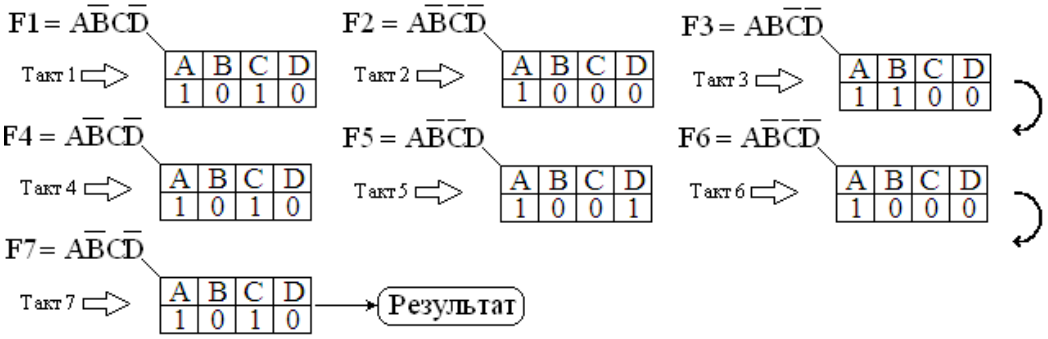


Рисунок 2.2 – Виконання завдання Алгоритму 2 по тактах

4. Провести аналіз отриманих алгоритмів.

1) По тактах 1, 4, 7 однакові функції $F1 = F4 = F7 = A\bar{B}C\bar{D}$;

По тактах 2, 6 однакові функції $F2 = F6 = A\bar{B}\bar{C}\bar{D}$.

2) При покроковому виконанні завдання попередні функції, які співпадають, повинні бути заблоковані.

3) Організацію покрокового виконання завдання здійснити за допомогою D-тригерів (рис. 2.3).

4) Алгоритми мають вірне вирішення завдання, але відрізняються операціями на третьому і п’ятому тактах. Ситуацію виправляє використання елемента АБО (рис. 2.4).

$$F3 = \bar{A}BC\bar{D} + \bar{A}\bar{B}CD;$$
$$F5 = \bar{A}\bar{B}CD + \bar{A}BC\bar{D}$$

5. Побудувати структурну схему комбінаційного автомата (рис. 2.5).

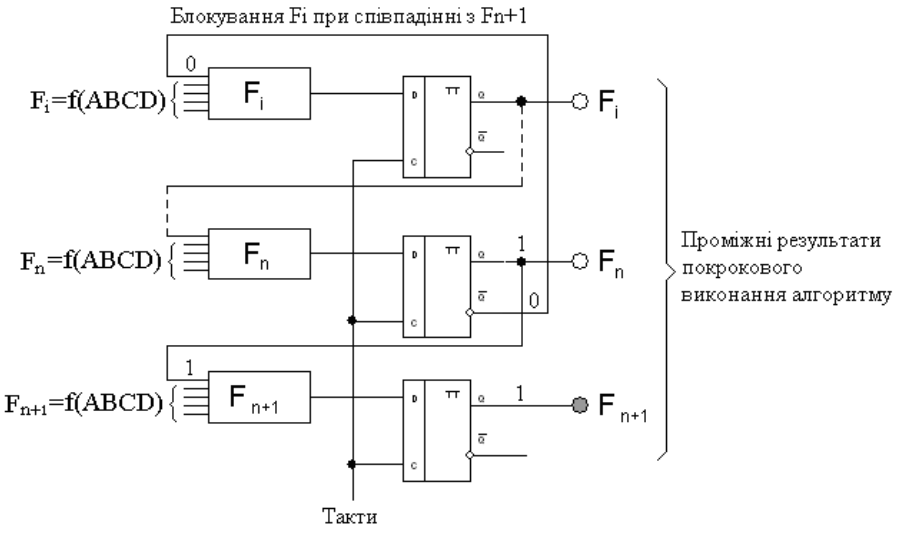


Рисунок 2.3 – Реалізація покрокового виконання завдання

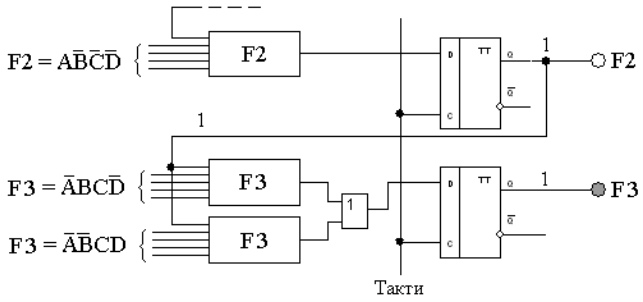


Рисунок 2.4 – Реалізація двох варіантів вирішення кроку алгоритмів на третьому і п’ятому тактах

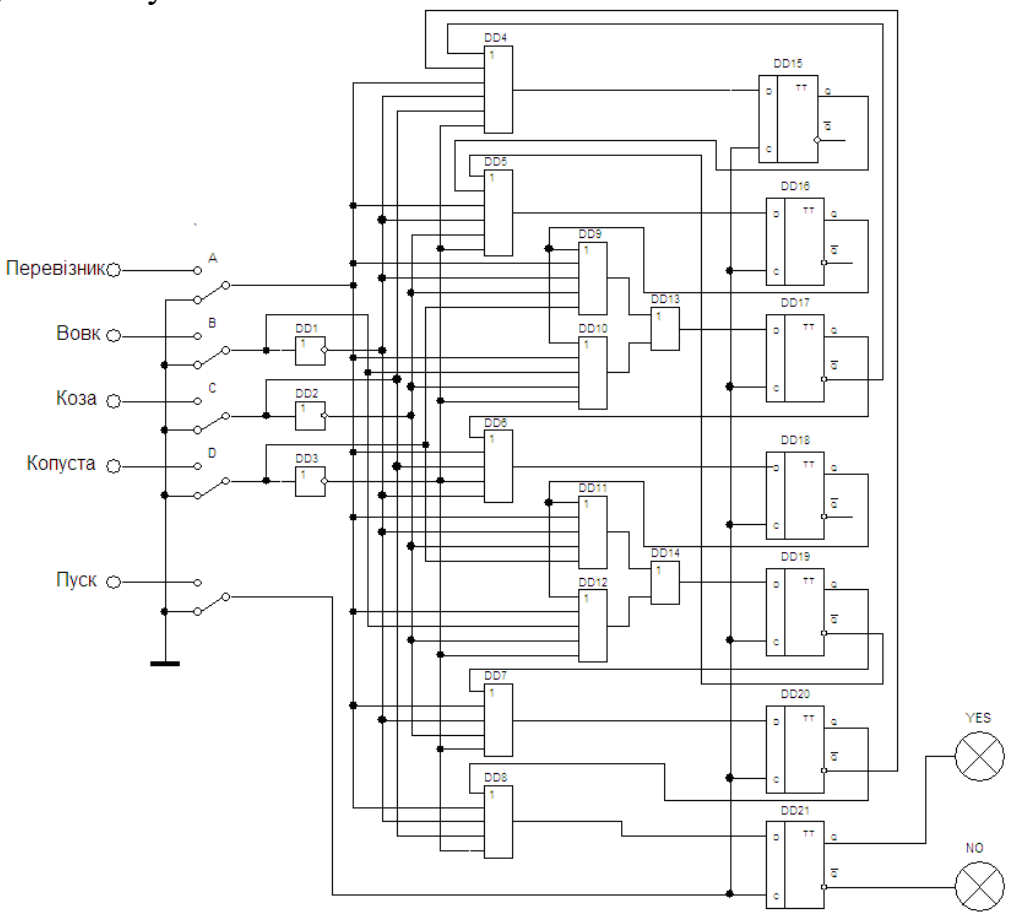


Рисунок 2.5 - Структурна схему комбінаційного автомата

6. Виконати перевірку функціонування схеми комбінаційного автомата в програмному забезпеченні Electronics Workbench (табл. 2.1).

Таблиця 2.1 - Таблиця функціонування комбінаційного автомата для Алгоритму 1

Dec	A	B	C	D	F
10	1	0	1	0	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
12	1	1	0	0	1
8	1	0	0	0	1
10	1	0	1	0	1

Побудувати схему для перевірки правильності функціонування за принципом генератора чисел (рис. 2.6) і отримати діаграму функціонування (рис. 2.7).

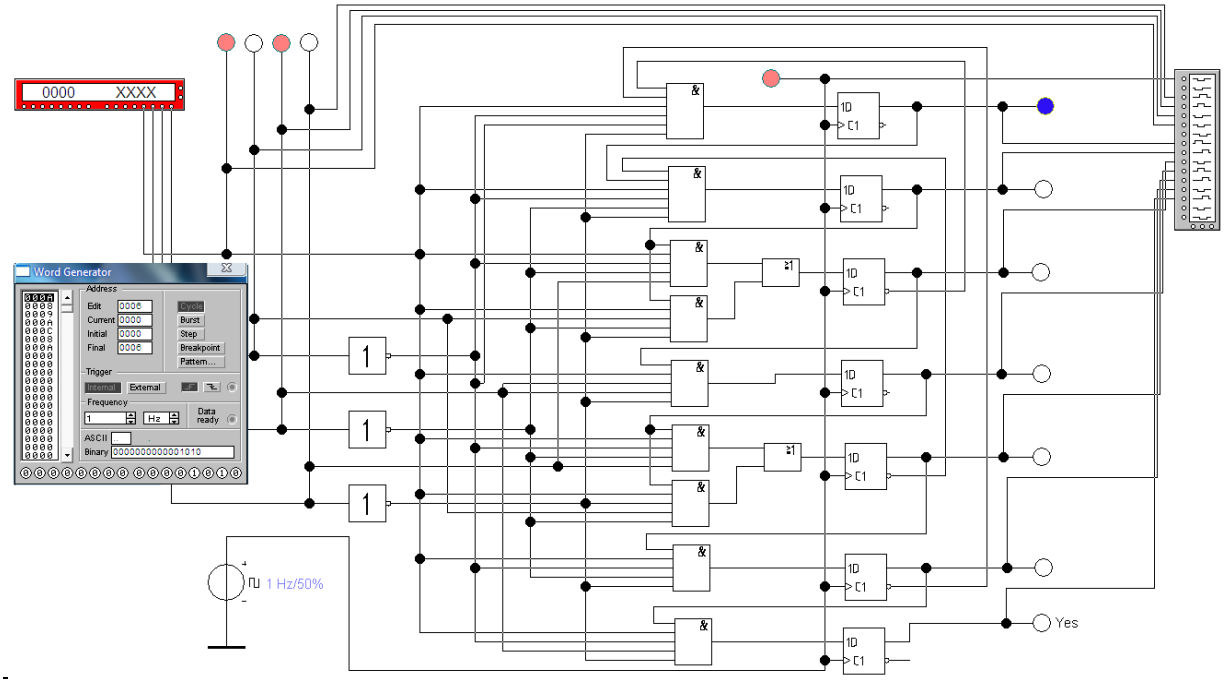


Рисунок 2.6 – Схема комбінаційного автомата за принципом генератора чисел

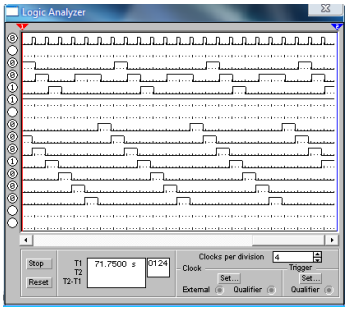


Рисунок 2.7 – Діаграма функціонування комбінаційного автомата згідно з Алгоритмом 1

Побудувати схему комбінаційного автомата з логічним принципом отримання правильного результату згідно з словесним описанням (рис. 2.8).

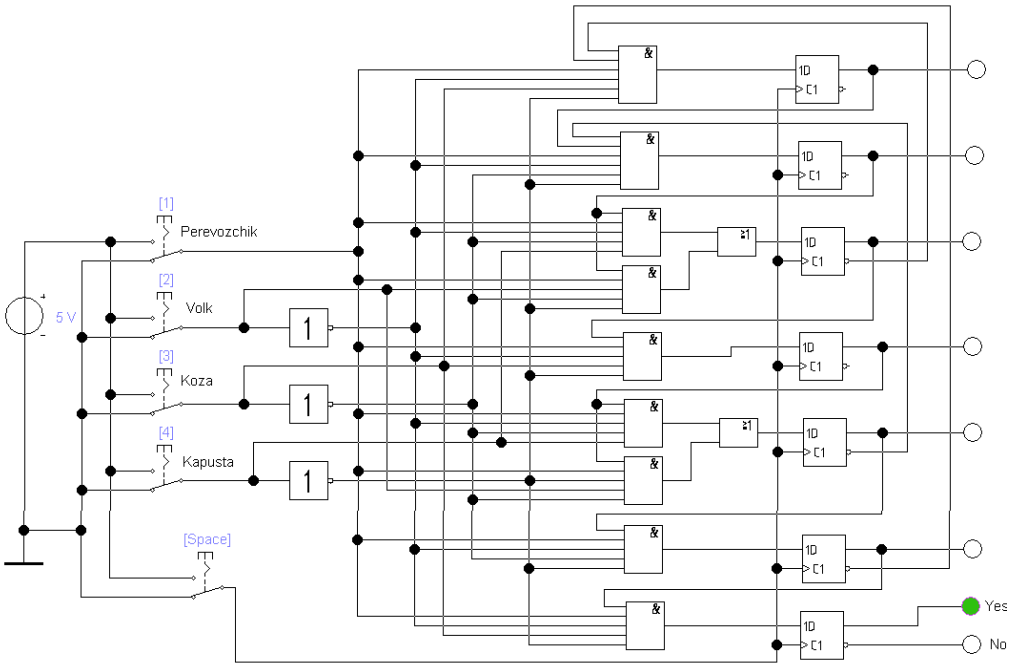


Рисунок 2.8 – Комбінаційний автомат, який імітує логічну гру «Перевізник»

Для отримання максимальної оцінки рішення пунктів завдання повинно бути виконано у повному обсязі з відповідними поясненнями і оформлено стилістично і технічно грамотно.

Практичне заняття №3

Дослідження операцій АЛП і відповідних їм логічних та логіко-арифметичних функцій

Метою практичного заняття є ознайомлення з операціями, які виконують арифметичні логічні пристрої мікроконтролерів та відповідних ним логічних та логіко-арифметичних функцій.

План практичного заняття.

1. Засвоїти методику виконання логічних операцій між операндами згідно з управляючими командами.
2. Засвоїти методику виконання арифметичних операцій між операндами згідно з управляючими командами.
3. Засвоїти методику виконання логіко-арифметичних операцій між операндами згідно з управляючими командами.

Ключові терміни та поняття: логічні операції, арифметичні операції, операнди, прискорене перенесення з розряду в розряд, набір операцій АЛП і відповідних їм допоміжних функцій.

Методичні вказівки до вивчення питань плану.

Арифметичнім логічним пристроєм (АЛП) називається функціональний блок, який виконує заданий набір арифметичних і логічних операцій над двома багаторозрядними операндами А, В. АЛП є основним блоком операційних пристроїв більшості сучасних цифрових систем [2].

Набір операцій, які виконує АЛП, визначається залежно від передбачуваної області його використання. Зазвичай АЛП виконують повний набір логічних функцій двох змінних або його частину, яка складається з найчастіше використовуваних функцій: кон'юнкція, диз'юнкція, інверсія, Виняткове АБО та ін. До складу арифметичних операцій обов'язково входять складання і віднімання [3].

Мікросхема К555ИП3 (рис. 3.1) призначена для дій з двома чотирьохрозрядними двійковими кодами: A = A3A2A1A0 і B = B3B2B1B0. Конкретний вигляд операції, яка виконується мікросхемою, задається п’ятирозрядним кодом на виходах MS3S2S1S0.

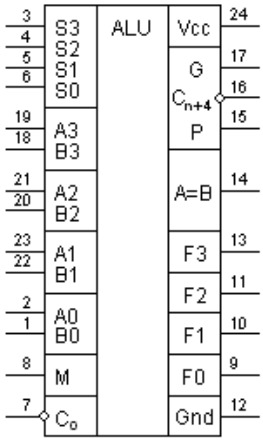


Рисунок 3.1 – Арифметичний логічний пристрій (мікросхема К155ИП3)

АЛП призначений для дій з двома чотирьохрозрядними двійковими кодами: A = A3A2A1A0 і B = B3B2B1B0. Конкретний вигляд операції, які виконується мікросхемою, задається п’ятирозрядним кодом на виходах MS3S2S1S0. Всього цей АЛП здатний виконати $2^5 = 32$ операції: 16 логічних і 16 арифметичних і логіко-арифметичних. Операції складання і віднімання проводяться з прискореним перенесенням з розряду в розряд [3].

У таблиці 2.2 приведений типовий набір з 16 логічних ($F_л$) і 16 арифметичних операцій (F_a) над операндами А і В, який виконується мікросхемами АЛП, які серійно випускаються [2].

На виходах F0F1F2F3 формуються результати логічних перетворень і арифметичних дій. На виході перенесення C_n+4 утворюється сигнал старшого (п'ятого) розряду при виконанні арифметичних операцій. Додаткові виходи – утворення прискореного перенесення G і поширення прискореного перенесення P – використовуються лише при організації багаторозрядних АЛП в разі їх поєднання з блоком прискореного перенесення. При виконанні арифметичних операції враховується значення перенесення С яке поступає в молодший розряд АЛП.

Таблиця 3.1 - Набір операцій АЛП і відповідних їм допоміжних функцій

Вибір функції				M = 1 логічні операції	M = 0 арифметичні операції	
S_3	S_2	S_1	S_0		C = 0	C = 1
0	0	0	0	$\overline{A}$	A	A+1
0	0	0	1	$\overline{A \vee B}$	$A \vee B$	$(A \vee B)+1$
0	0	1	0	$\overline{A} \wedge B$	$A \vee \overline{B}$	$(A \vee \overline{B})+1$
0	0	1	1	0	−1 (доповнення до 2)	0
0	1	0	0	$\overline{A \wedge B}$	$A+(A \vee \overline{B})$	$A+(A \vee \overline{B})+1$
0	1	0	1	$\overline{B}$	$(A \vee B)+(A \wedge \overline{B})$	$(A \vee B)+(A \wedge \overline{B})+1$
0	1	1	0	$A \oplus B$	$A+\overline{B}$	$A+\overline{B}+1$
0	1	1	1	$A \wedge \overline{B}$	$(A \wedge \overline{B})+1$	$(A \wedge \overline{B})+1+1$
1	0	0	0	$\overline{A} \vee B$	$A+(A \wedge B)$	$A+(A \wedge B)+1$
1	0	0	1	$\overline{A \oplus B}$	A+B	A+B+1
1	0	1	0	B	$(A \vee \overline{B})+(A \wedge B)$	$(A \vee \overline{B})+(A \wedge B)+1$
1	0	1	1	$A \wedge B$	$(A \wedge B)+1$	$(A \wedge B)+1+1$
1	1	0	0	1	A+A	A+A+1
1	1	0	1	$A \vee \overline{B}$	$(A \vee B)+A$	$(A \vee B)+A+1$
1	1	1	0	$A \vee B$	$(A \vee \overline{B})+A$	$(A \vee \overline{B})+A+1$
1	1	1	1	A	$\overline{A}+1$	$\overline{A}+1+1$

Слова A і B, які підлягають обробці, можуть бути представлені в позитивній або негативній логіці. Таблиці істинності для кожного варіанту логіки різні. Для позитивної логіки дійсна таблиця 3.2. Старший розряд коду вибору операції (вхід M) визначає характер дій, які виконуються АЛП. Коли на цьому вході сигнал високого рівня, АЛП виконує логічні операції порозрядно над кожною парою біт слів A і B, а перенесення між розрядами забороняються.

Останні управляючі сигнали (S3S2S1S0) визначають вибір однієї з 16 можливих операцій.

Якщо АЛП виконує логико-арифметичну операцію, логічна функція реалізується порозрядно, а арифметична – з перенесенням. Наприклад, вхідному коду MS3S2S1S0 = 01000 відповідає операція A^B плюс A, де AB - логічне множення двох слів. Якщо A = 0101 і B = 1011, то перша логічна операція множення дає A^B = 0001, друга арифметична операція сума дає 0001 плюс 0101 = 0110.

Питання для самоконтролю.

1. Пам'ять мікроконтролера.
2. Компаратори і схеми контролю.
3. Суматори.
4. Арифметичний логічний пристрій.
5. Набір операцій АЛП і відповідних їм допоміжних функцій.
6. Використання АЛП у якості компаратора.

Практичні завдання, задачі.

1. Вивчити призначення арифметичного логічного пристрою та набір арифметичних і логічних операцій, які він виконує над двома багаторозрядними операндами.

2. Обрати згідно з варіантом функцію (табл. 3.1).

3. Побудувати експериментальну схему дослідження обраної функції в програмному забезпеченні Electronics Workbench (Приклад функції F8 приведено на рисунку 3.2).

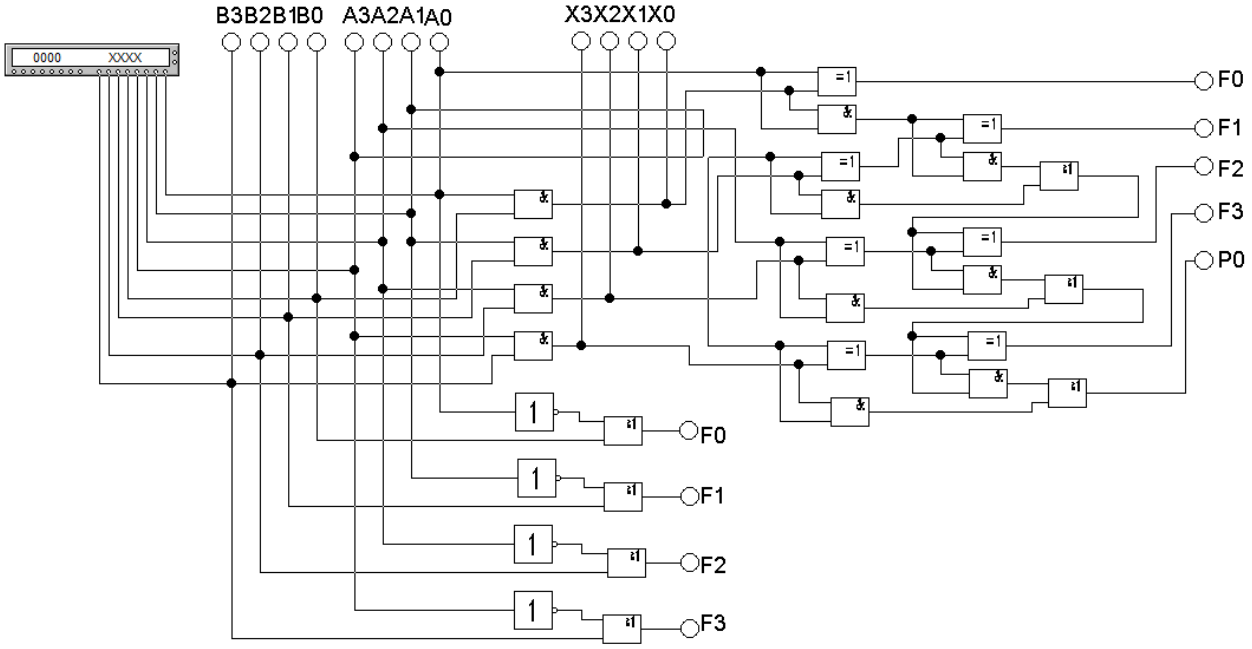


Рисунок 3.1 - Схема дослідження функції F8 в програмному забезпеченні Electronics Workbench

4. Задатись двома багаторозрядними операндами.

5 Провести логічні операції з операндами.

6. Провести дослідження логічних операцій за допомогою індикаторних пристроїв.

7. Провести логіко-арифметичні операції з операндами.

6. Провести дослідження логіко-арифметичних операцій за допомогою індикаторних пристроїв.

7. Зафіксувати алгоритм прискореного перенесення з розряду в розряд.

8. Зробити розгорнутий висновок за результатами досліджень.

Для отримання максимальної оцінки рішення пунктів завдання має бути виконане у повному обсязі, з відповідними поясненнями, оформлене стилістично і технічно грамотно.

Практичне заняття №4

Використання дешифраторів для виконання логічних операцій

Метою практичного заняття є вивчення методів синтезу цифрових дешифраторів мікропроцесорних систем.

План практичного заняття.

1. Засвоїти принципи використання дешифраторів в мікропроцесорній техніці.
2. Виконати розрахункові завдання з використанням засобів алгебри логіки.
3. Побудувати структурні та принципові схеми, аналітичні схеми в програмному забезпеченні Electronics Workbench, згідно з завданням, та провести дослідження їх функціонування за допомогою індикаторних пристроїв та діаграм.

Ключові терміни та поняття: дешифратор, таблиця істинності, адреса, комбінаційний пристрій, логічна функція.

Методичні вказівки до вивчення питань плану.

Дешифратор (декодер, Decoder) - цифровий пристрій, який має n входів і m виходів і який перетворює вхідний код в сигнал на одній окремій вихідній лінії. Дешифратор називають повним, якщо він має кількість виходів m, пов'язаних з кількістю розрядів п вхідного двійкового числа співвідношенням $m = 2^n$. На кожному з його виходів з'являється логічна одиниця лише в тому випадку, якщо на вхід приходить відповідна комбінація вхідних сигналів [4].

Функція мікросхем дешифраторів, як випливає з назви, полягає в тому, аби перетворювати вхідний двійковий код в номер вихідного сигналу, кількість яких відповідає кількості станів двійкового коду, тобто 2^n, де n – кількість розрядів двійкового коду (кількість інформаційних входів дешифратора).

Простий двохрозрядний дешифратор, виконаний на стандартних логічних елементах І і НЕ, показаний на рисунку 4.1. Таблиця істинності двохрозрядного дешифратора представлена в таблиці. 4.1

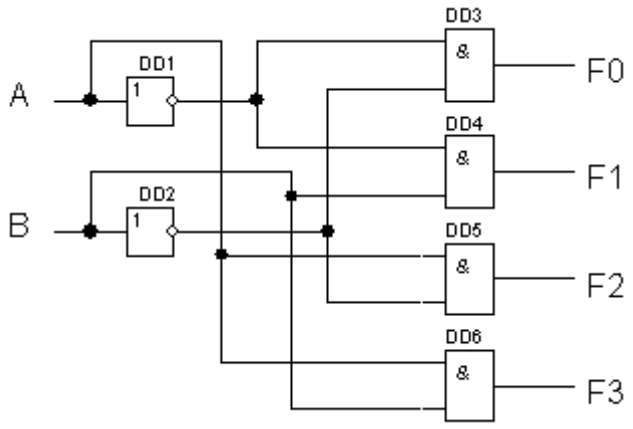


Рисунок 4.1 - Простий двохрозрядний дешифратор

Таблиця 4.1 - Таблиця істинності двохрозрядного дешифратора

Номер вхідного сигналу	Вхід		Вихід			
	A	B	F0	F1	F2	F3
0	0	0	1	0	0	0
1	0	1	0	1	0	0
2	1	0	0	0	1	0
3	1	1	0	0	0	1

У стандартних серіях мікросхем існують дешифратори на 4, 8 або 16 виходів, відповідно вони мають 2, 3 або 4 входи. Ще різниці між мікросхемами містять в собі схеми управління і типи виходів мікросхем (звичайно 2С вихід або вихід з загальним колектором [5].

Входи дешифраторів зазвичай позначають цифрами, яким відповідає вага розряду двійкового числа (1, 2, 4, 8), а виходи також позначають цифрами (1, 2, 3 і т. д.).

Мікросхеми стандартних дешифраторів, наприклад, К555ИД14, К555ИД7, К555ИД3, (рис. 4.2), мають інформаційні входи 1, 2, 4, 8, входи дозволу С1, С2, С3, об'єднані по функції І, а також виходи від 0 до 15. Відмінність між цими мікросхемами полягає в кількості входів і виходів.

Мікросхеми дешифратори використовують для: дешифрування вхідних код, селекції код, мультиплексування ліній.

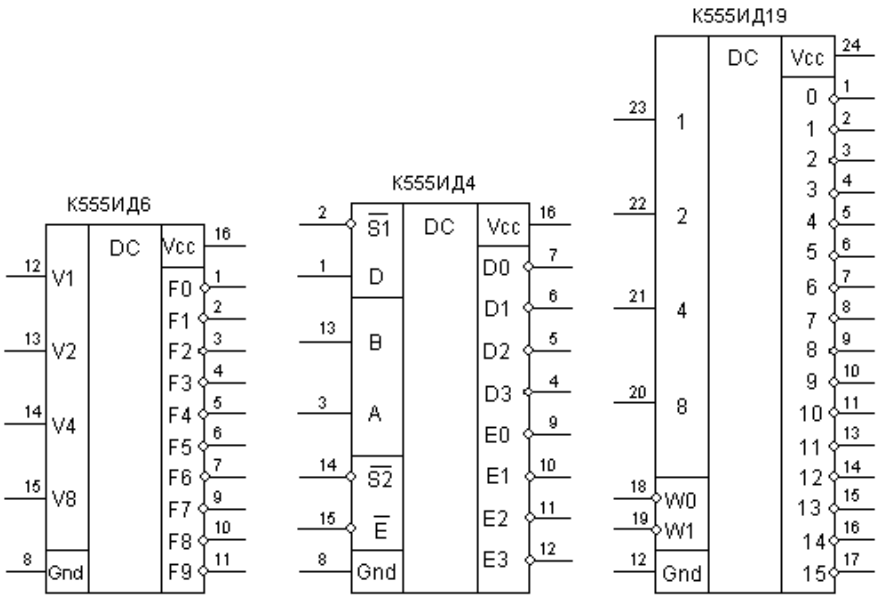


Рисунок 4.2 - Мікросхеми стандартних дешифраторів

Особливістю більшості серійних мікросхем є їх багатофункціональність. Приведена на рисунку 4.3 мікросхема DD1 дешифратора при A=0 виконуватиме функцію перетворювача трьохрозрядного двійкового коду у вісімковий.

Значення A = і при цьому буде заборонено. Якщо на вхід A подавати послідовний двійковий код, а входи B, C, D використовувати як адресні, то одержимо демультиплексор “1 в 8”.

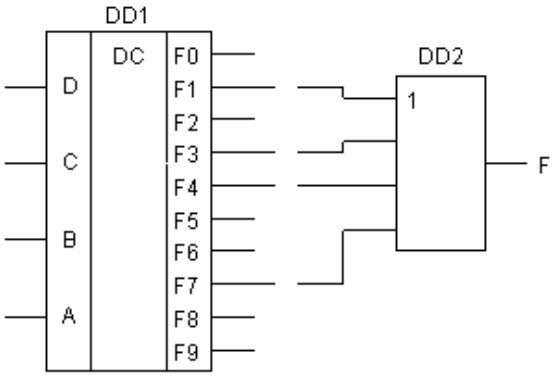


Рисунок 4.3 – Перетворювач трьохрозрядного двійкового коду

Питання для самоконтролю.

1. Мікропроцесорні функціональні вузли комбінаційного типу.
2. Дешифратори мікропроцесорних систем.
3. Подова логічних схем з використанням дешифраторів.
4. Використання дешифраторів.

Практичні завдання, задачі.

Приклад виконання завдання.

Розробити перетворювач чотирьохрозрядного двійкового коду в код “2 з 5”.

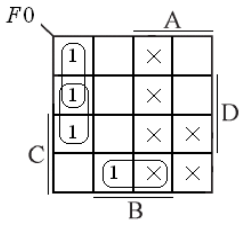
Код “2 з 5” використовується для безпомилкової передачі цифрової інформації. В цьому п’ятирозрядному коді кожна допустима комбінація містить дві одиниці та три нулі і, оскільки таких комбінацій десять, то кожна з них відповідає одній десятковій цифрі (табл. 4.2).

Таблиця 4.2 – Таблиця функціонування перетворювача коду

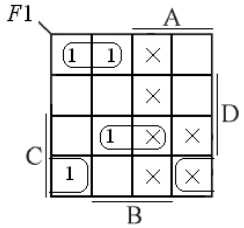
Десяткова цифра	Чотирьохрозрядний двійковий код				Код "2 з 5"				
	A	B	C	D	F4	F3	F2	F1	F0
0	0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	0	1	0	1
2	0	0	1	0	0	0	1	1	0
3	0	0	1	1	0	1	0	0	1
4	0	1	0	0	0	1	0	1	0
5	0	1	0	1	0	1	1	0	0
6	0	1	1	0	1	0	0	0	1
7	0	1	1	1	1	0	0	1	0
8	1	0	0	0	1	0	1	0	0
9	1	0	0	1	1	1	0	0	0

Мінімізуючи функції F4, F3, F2, F1, F0, отримаємо вирази для вихідних сигналів, які легко реалізовуються у базисі комбінаційної логіки або з допомогою дешифратора.

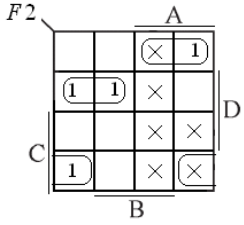
$$F0 = \Sigma(0,1,3,6) = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}D + BC\overline{D} = \overline{A}\overline{B}(\overline{C} + D) + BC\overline{D}$$



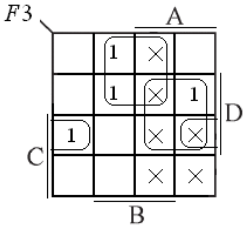
$$F1 = \Sigma(0,2,4,7) = \overline{A}C\overline{D} + BCD + \overline{B}C\overline{D} = C(BD + \overline{B}\overline{D}) + \overline{A}C\overline{D} = C(\overline{B \oplus D}) + \overline{A}C\overline{D}$$



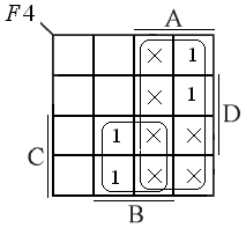
$$F2 = \Sigma(1,2,5,8) = A\overline{C}\overline{D} + \overline{A}C\overline{D} + \overline{B}C\overline{D} = \overline{C}(A\overline{D} + \overline{A}D) + \overline{B}C\overline{D} = \overline{C}(A \oplus D) + \overline{B}C\overline{D}$$



$$F3 = \Sigma(3,4,5,9) = B\overline{C} + AD + \overline{B}CD = B\overline{C} + D(A + \overline{B}C)$$



$$F4 = \Sigma(6,7,8,9) = A + BC$$



Схему перетворювача чотирьохрозрядного двійкового коду в код “2 з 5” можна побудувати на елементах звичайної логіки, використовуючи різні механізми спрощення та оптимізації (рис. 4.4).

Скористуємось методикою побудови схеми чотирьохрозрядного двійкового коду в код “2 з 5” на дешифраторі К555ИД6 та логічних елементах І, для комутації на відповідні виходи F4, F3, F2, F1, F0, згідно з таблицею 4.2, виходів дешифратора (рис. 4.5).

Реалізація схеми на логічних елементах (рис. 4.4).

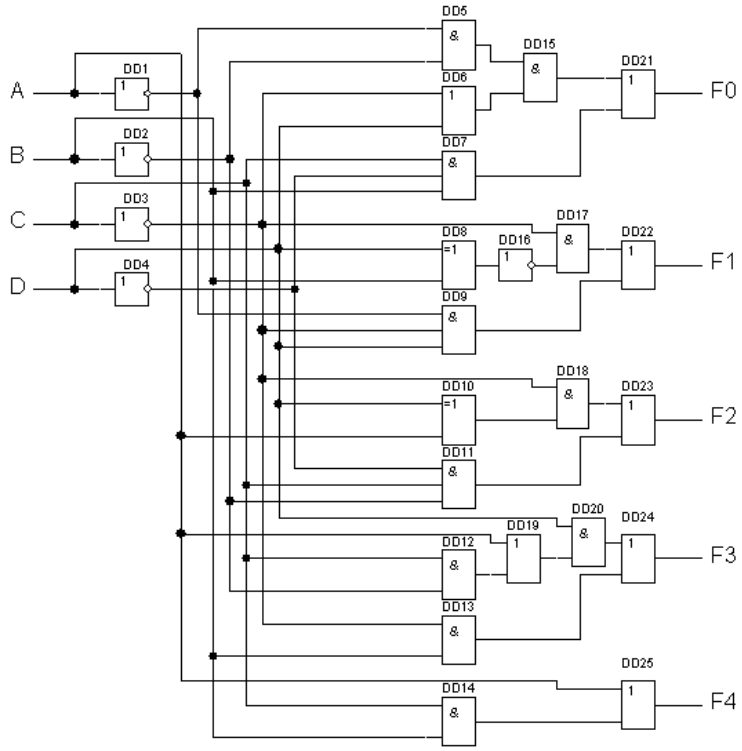


Рисунок 4.4 - Реалізація схеми перетворювача чотирьохрозрядного двійкового коду в код “2 з 5”

Реалізація схеми перетворювача на дешифраторі (рис. 4.5).

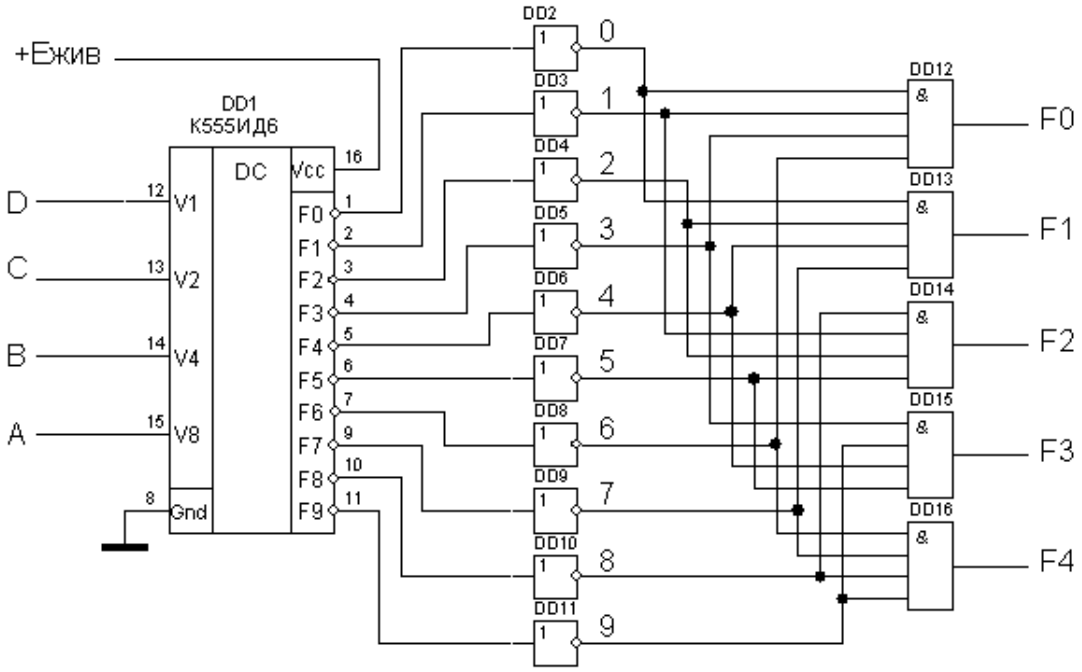


Рисунок 4.5 - Реалізація схеми перетворювач чотирьохрозрядного двійкового коду в код “2 з 5”

Проведемо аналіз функціонування схеми перетворювача в програмному забезпеченні EWB (рис.4.6).

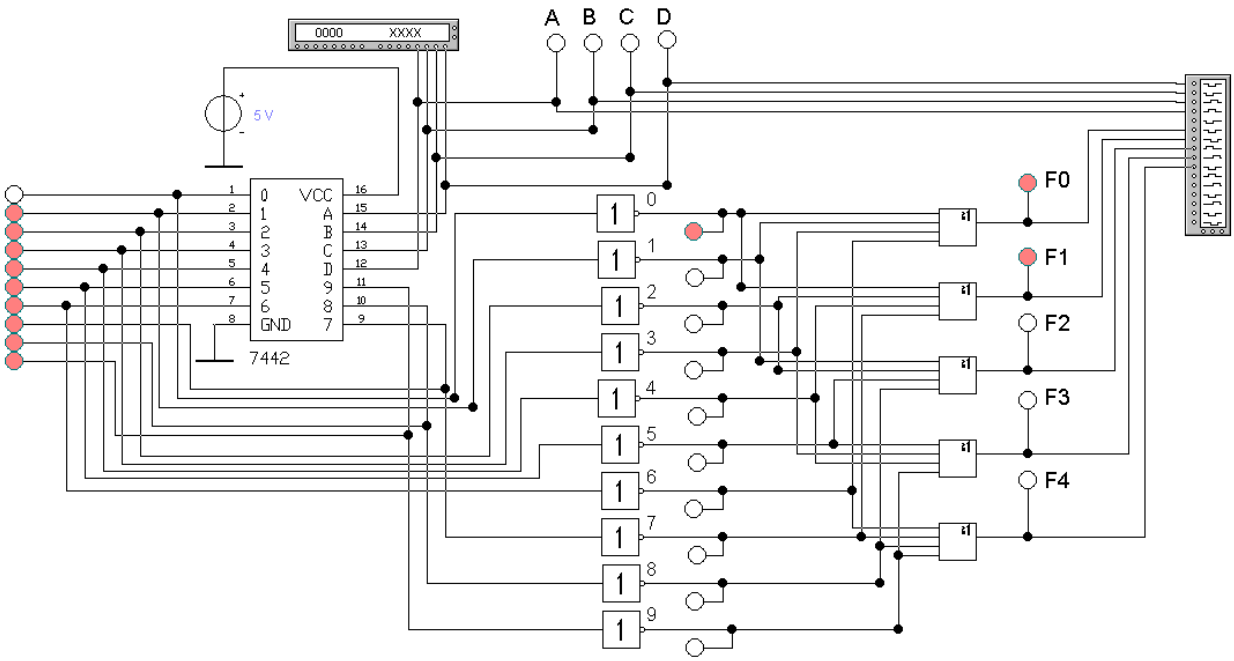


Рисунок 4.6 – перевірка функціонування схеми перетворювач чотирьохрозрядного двійкового коду в код “2 з 5”

Діаграма функціонування (рис. 4.7) відповідає таблиці функціонування перетворювач чотирьохрозрядного двійкового коду в код “2 з 5” (табл..4.2).

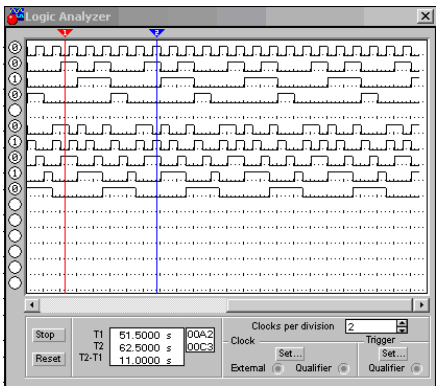


Рисунок 4.7 - Аналіз функціонування перетворювача коду на дешифраторі

Завдання.

Пороговою логічною схемою називається схема, в якій певна мінімальна кількість вхідних змінних повинна мати стан 1, аби на виході з'явилася логічна 1. Потрібно розрахувати схему з п'ятьма вхідними змінними. На виході має бути 1 лише тоді, коли, щонайменше, на чотирьох входах присутня 1.

Схему побудувати з використанням дешифратора.

Для отримання максимальної оцінки рішення пунктів завдання має бути виконане у повному обсязі, з відповідними поясненнями, оформлене стилістично і технічно грамотно.

Практичне заняття №5

Мультиплексори даних мікропроцесорних систем

Метою практичного заняття є вивчення методів синтезу цифрових комутаторів мікропроцесорних систем.

План практичного заняття.

1. Засвоїти принципи використання мультиплексорів в мікропроцесорній техніці.
2. Виконати розрахункові завдання з використанням засобів алгебри логіки.
3. Побудувати структурні та принципові схеми, аналітичні схеми в програмному забезпеченні Electronics Workbench, згідно з завданням, та провести дослідження їх функціонування за допомогою індикаторних пристроїв та діаграм.

Ключові терміни та поняття: мультиплексор, таблиця істинності, адреса, комбінаційний пристрій, логічна функція.

Методичні вказівки до вивчення питань плану.

У цифрових пристроях часто виникає завдання передачі цифрової інформації від джерел до одного приймача. Для цього на вході каналу встановлюється мультиплексор (MS), який згідно коду адреси підключає до виходу одне з джерел інформації. Мультиплексор призначений для почергової передачі на один вихід один з декількох вхідних сигналів, тобто для їх мультиплексування. Можна сказати інакше - мультиплексори призначені для комутації одного з декількох джерел сигналів до однієї вихідної шини. Кількість мультиплексуємих входів називається кількістю каналів мультиплексора, а кількість виходів - числом розрядів мультиплексора [1].

Робота мультиплексора описуватиметься наступною логічною функцією:

$$Q = V(\overline{A}\,\overline{B}D0 + \overline{A}\,\overline{B}D1 + \overline{A}BD2 + A\overline{B}D3) \qquad (5.1)$$

Оскільки мультиплексор є логічним пристроєм, то на його базі можуть реалізовуватись різні логічні функції. Як приклад, скористаємось формулою (5.1). Припустимо, що необхідно реалізувати логічну функцію:

$$Q = AB + \overline{A}\,\overline{B} \qquad (5.2)$$

Порівнюючи формули (5.1) і (5.2), бачимо, що, присвоївши адресним сигналам AB значення відповідно формули (5.2) і забезпечивши виконання умов $D1 = D2 = 0$, $D0 = D3 = 1$, $V = 0$, одержимо схему пристрою (рис. 5.1), який реалізує функцію (5.2).

Нескладні перетворення логічних функцій дають можливість реалізовувати на мультиплексорах функції з кількістю змінних, яка перевершує кількість адресних входів.

Виконується наступна послідовність перетворень:

- у мінімізованій диз'юнктивній нормальній формі функції виділяються змінні, які мають найвищий ранг, тобто повторюються в найбільшій кількості диз'юнкцій. Наприклад, вибирається m змінних;

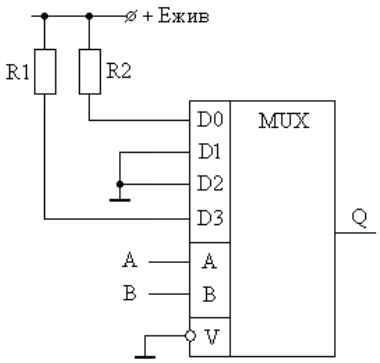


Рисунок 5.1 – Схемна реалізація функції

- виконується перетворення функції так, щоб забезпечити присутність виділених змінних в усіх диз'юнкціях. Для цього кожна диз'юнкція домножається на $1 = A + \overline{A}$;

- проводиться перегрупування змінних логічної функції таким чином, щоб виділені змінні були винесені за дужки у відповідних групах диз'юнкцій.

У результаті виконаних перетворень отримується формула вихідної функції, яка реалізується на мультиплексорі "M = 2^m в", якщо на його адресні входи подати виділені змінні, а на інформаційних входах забезпечується виконання тих логічних виразів, які залишились в дужках. Якщо вирази в дужках є функціями декількох змінних, то їх реалізація може бути забезпечена або за допомогою іншого мультиплексора, або на основі типових логічних елементів [4].

Питання для самоконтролю.

1. Загальне поняття селекторів даних.
2. Принципи побудови мультиплексорів.
3. Реалізація логічних схем на мультиплексорах.

Практичні завдання, задачі.

Приклад завдання 1. Функцію п'яти змінних реалізувати з використанням мультиплексорів 4 в 1.

$$Q = AB\overline{D}\overline{E} + \overline{A}\overline{D}E + A\overline{C}D + C\overline{D} + \overline{B}C$$

У використовуваному мультиплексорі наявні два адресні входи - отже, вибираємо дві змінні з найбільшими рангами. Ранги змінних приведені в таблиці 5.1.

Таблиця 5.1 – Ранги змінних в рівнянні функції

A	B	C	D	E
3	2	3	4	2

Вибираємо змінні C, D.

Виконуємо перетворення функції.

$$Q = AB\overline{D}\overline{E} \cdot (C + \overline{C}) + \overline{A}\overline{D}E \cdot (C + \overline{C}) + A\overline{C}D + C\overline{D} + \overline{B}C \cdot (D + \overline{D}) =$$

$$= ABC\overline{D}\overline{E} + AB\overline{C}\overline{D}\overline{E} + \overline{A}C\overline{D}E + \overline{A}C\overline{D}\overline{E} + A\overline{C}D + C\overline{D} + \overline{B}CD + \overline{B}C\overline{D} =$$

$$= \overline{C}\overline{D}(AB\overline{E} + \overline{A}E) + \overline{C}D(A) + CD(\overline{B}) + C\overline{D}(AB\overline{E} + \overline{A}E + 1 + \overline{B}) =$$

$$= \overline{C}\overline{D}(AB\overline{E} + \overline{A}E) + \overline{C}D(A) + CD(\overline{B}) + C\overline{D}(1)$$

Логічна сума в перших дужках також реалізується за допомогою мультиплексора 4 в 1, якщо на адресні входи його подати змінні A та E.

Принципова схема пристрою, який реалізує потрібну логічну функцію, приведена на рисунку 5.2. Але така схема є лише прикладом і на практиці не піддається реалізації, оскільки в серіях мікросхем відсутні мультиплексори 4 в 1, а спарений мультиплексор К555КП2 (зарубіжні аналоги 74153PC, SN74153N) має об’єднані адресні входи.

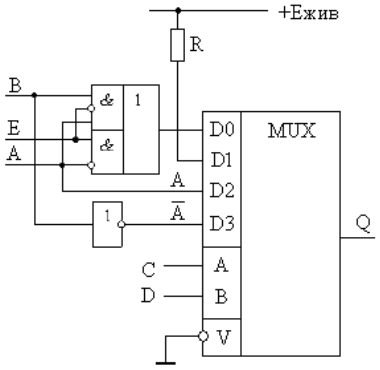


Рисунок 5.2 - Принципова схема заданого пристрою

Практично задача може бути розв’язана при використанні мультиплексора 8 в 1.

Вибравши логічні змінні A, C, D як змінні з найвищими рангами (табл. 5.1), функцію перетворимо до наступного вигляду

$$Q = AB\overline{D}\overline{E} \cdot (C + \overline{C}) + \overline{A}D\overline{E} \cdot (C + \overline{C}) + A\overline{C}D + C\overline{D} + \overline{B}C \cdot (D + \overline{D}) =$$

$$= ABC\overline{D}\overline{E} + AB\overline{C}\overline{D}\overline{E} + \overline{A}C\overline{D}\overline{E} + \overline{A}C\overline{D}E + A\overline{C}D + C\overline{D} + \overline{B}CD + \overline{B}C\overline{D} =$$

$$= \overline{A}\overline{C}\overline{D} \cdot (E) + A\overline{C}\overline{D} \cdot (B\overline{E}) + \overline{A}C\overline{D} \cdot (1) + AC\overline{D} \cdot (1) + \overline{A}\overline{C}D \cdot (0) +$$

$$+ A\overline{C}D \cdot (1) + \overline{A}CD \cdot (\overline{B}) + ACD \cdot (\overline{B})$$

У таблиці 5.2 зведені значення необхідних сигналів, які подаються на відповідні інформаційні входи.

Таблиця 5.2 – Сигнали на інформаційних входах

№	A	B	C	Di
0	0	0	0	E
1	0	0	1	0
2	0	1	0	1
3	0	1	1	$\overline{B}$
4	1	0	0	$B\overline{E}$
5	1	0	1	1
6	1	1	0	1
7	1	1	1	$\overline{B}$

На рисунку 5.3 приведена схема, яка реалізує задану функцію.

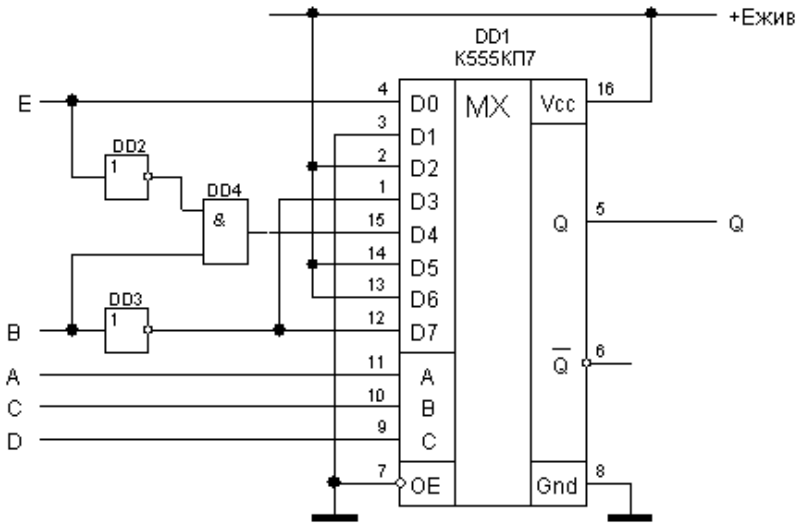


Рисунок 5.3 – Реалізація заданої функції на мікропроцесорі 8 в 1.

Схему на реальних елементах можна зібрати на мультиплексорі К555КП7 (його зарубіжні аналоги - мікросхеми 54151). Необхідні допоміжні логічні функції реалізуються з використанням одного корпусу мікросхеми К155ЛА3 (зарубіжні аналоги - 7400).

Перевірка функціонування розробленої схеми в програмному забезпеченні EWB (рис. 5.4).

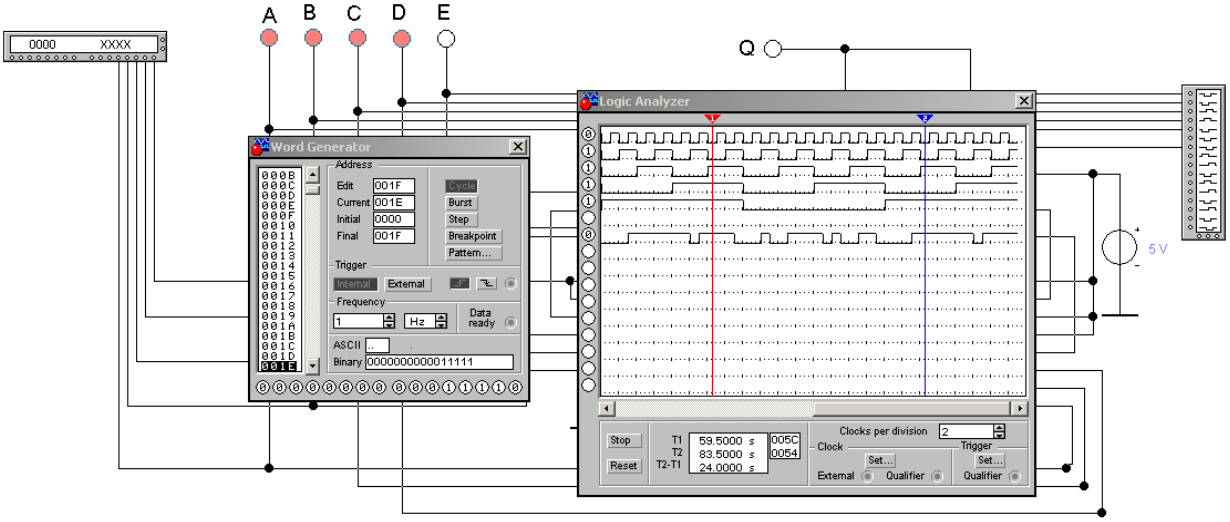


Рисунок 5.4 - Перевірка функціонування розробленої схеми

Приклад завдання 2. Використовуючи мультиплексор (8 в 1) К555КП7, реалізувати логічну функцію:

$$F = \sum (0, 2, 3, 5, 7)$$

Оскільки логічна функція має три змінні, то вони повинні бути приєднані до адресних входів ABC (рис. 5.5). Щоб дозволити роботу мультиплексора, вхід V повинен бути заземленим. Входи даних D0, D2, D3, D5, D7 необхідно приєднати до шини живлення, а решту входів заземлити.

В результаті схема пристрою може бути реалізована на мультиплексорі К555КП7 (74151).

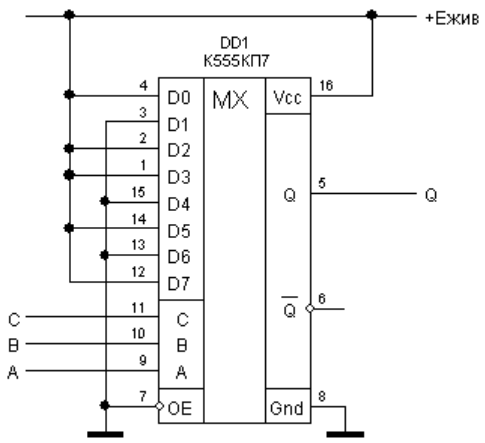


Рисунок 5.5 – Схема пристрою

Варіанти завдання 1:

Функцію п’яти змінних реалізувати з використанням мультиплексорів 4 в 1.

1. $Q = A\overline{B}D\overline{E} + \overline{A}B\overline{D}E + A\overline{C}D + \overline{A}C\overline{D} + \overline{B}\overline{C} + CDE$
2. $Q = B\overline{D}E + AB\overline{D}\overline{E} + \overline{A}\overline{D}\overline{E} + A\overline{C}DE + \overline{B}C\overline{D} + \overline{B}C$

Варіанти завдання 2:

Використовуючи мультиплексор (16 в 1) К155ИД3, реалізувати логічну функцію:

1. $F = \sum (0, 2, 4, 5, 7, 11, 13)$
2. $F = \sum (1, 3, 5, 9, 12, 14, 15)$

Практичне заняття №6

Збільшення розрядності демультиплексорів

Метою практичного заняття є вивчення методів синтезу цифрових комутаторів мікропроцесорних систем.

План практичного заняття.

1. Засвоїти принципи використання демультиплексорів в мікропроцесорній техніці.
2. Виконати розрахункові завдання з використанням засобів алгебри логіки.
3. Побудувати структурні та принципові схеми, аналітичні схеми в програмному забезпеченні Electronics Workbench, згідно з завданням, та провести дослідження їх функціонування за допомогою індикаторних пристроїв та діаграм.

Ключові терміни та поняття: демультиплексор, таблиця істинності, адреса, комбінаційний пристрій, логічна функція.

Методичні вказівки до вивчення питань плану.

Демультиплексори - це цифрові комбінаційні пристрої, функціональне призначення яких протилежне функціональному призначенню мультиплексорів. У них сигнали з одного інформаційного входу D розподіляються на 2^m виходів F_i , які комутуються m адресними входами, тобто фактично вони є генераторами мінтермів. Завдяки цій властивості легко створювати канонічні суми, тобто реалізовувати за допомогою логічних операцій АБО логічні функції вхідних змінних [1].

Якщо прийняти кількість адресних входів ш = 2 (рис. 6.1), то роботу пристрою описуватимуть булеві рівняння (6.1).

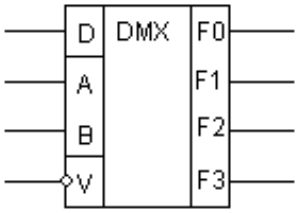


Рисунок 6.1 – Схема демультиплексора

$$\begin{aligned} F0 &= \overline{V}D\overline{A}\,\overline{B}; \\ F1 &= \overline{V}D\overline{A}B; \\ F2 &= \overline{V}DA\overline{B}; \\ F3 &= \overline{V}DAB. \end{aligned} \qquad (6.1)$$

Таблична форма запису роботи демультиплексора, згідно з рівняннями (6.1), представлена в таблиці 6.1.

Рисунок 6.1 – Таблиця істинності демультиплексора

V	A	B	D	F0	F1	F2	F3
0	0	0	1/0	1/0	0	0	0
0	0	1	1/0	0	1/0	0	0
0	1	0	1/0	0	0	1/0	0
0	1	1	1/0	0	0	0	1/0
1	0	0	×	$\overline{V}$	0	0	0
1	0	1	×	0	$\overline{V}$	0	0
1	1	0	×	0	0	$\overline{V}$	0
1	1	1	×	0	0	0	$\overline{V}$

У відповідності до схеми (рис. 6.1) активними станами виходів F0F1F2F3 є логічні “1”.

У мікросхемах ТТЛ виходи часто мають інверсні значення, і активні стани характеризуються низьким рівнем сигналу.

Збільшення розрядності дешифраторів та демультиплексорів

На рисунку 6.2 показано з'єднання показано з'єднання двох дешифраторів "3 в 8" для здобуття одного дешифратора "4 в 16", або двох демультиплексорів "1 в 8" для здобуття одного демультиплексора "1 в 16" [6].

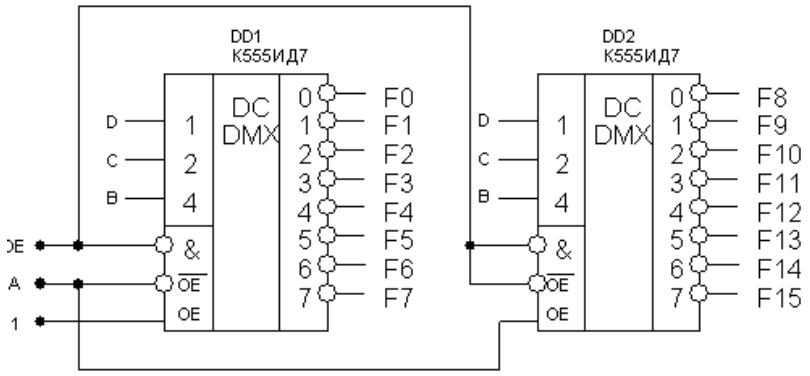


Рисунок 6.2 – Збільшення розрядності дешифратора та демультиплексора

При пасивному значенні OE = 1 на всіх виходах F_i буде "1", незалежно від значень вхідних сигналів. Якщо OE = 0 (активний рівень), то який з дешифраторів працює, залежить лише від сигналу A (табл. 6.2).

Таблиця 6.2 – Алгоритм збільшення розрядності дешифратора та демультиплексора

DEC № вих.	A	BCD	F0F1F2F3F4F5F6F7	F8F9F10F11F12F13F14F15
			F0F1F2F3F4F5F6F7	F8F9F10F11F12F13F14F15
0	0	0 0 0	0 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
1		0 0 1	1 0 1 1 1 1 1 1	1 1 1 1 1 1 1 1
2		0 1 0	1 1 0 1 1 1 1 1	1 1 1 1 1 1 1 1
3		0 1 1	1 1 1 0 1 1 1 1	1 1 1 1 1 1 1 1
4		1 0 0	1 1 1 1 0 1 1 1	1 1 1 1 1 1 1 1
5		1 0 1	1 1 1 1 1 0 1 1	1 1 1 1 1 1 1 1
6		1 1 0	1 1 1 1 1 1 0 1	1 1 1 1 1 1 1 1
7		1 1 1	1 1 1 1 1 1 1 0	1 1 1 1 1 1 1 1
8	1	0 0 0	1 1 1 1 1 1 1 1	0 1 1 1 1 1 1 1
9		0 0 1	1 1 1 1 1 1 1 1	1 0 1 1 1 1 1 1
10		0 1 0	1 1 1 1 1 1 1 1	1 1 0 1 1 1 1 1
11		0 1 1	1 1 1 1 1 1 1 1	1 1 1 0 1 1 1 1
12		1 0 0	1 1 1 1 1 1 1 1	1 1 1 1 0 1 1 1
13		1 0 1	1 1 1 1 1 1 1 1	1 1 1 1 1 0 1 1
14		1 1 0	1 1 1 1 1 1 1 1	1 1 1 1 1 1 0 1
15		1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 0

Якщо A = 0, то на всіх вирішуючих входах дешифратора DD1 будуть активні рівні, а на прямому вході дозволу OE дешифратора DD2 сигнал A, рівний нулю, переведе всі виходи в стан 1, тобто працездатним буде

дешифратор DD1. Коли A = 1, навпаки працездатним стає дешифратор DD2, тому що для прямого входу дешифратора цей сигнал дозволяє його виходи. Комбінації сигналів A і BCD утворюють послідовність двійкового коду 0000 ... 0111 (0 ... 7) для дешифратора DD1 і послідовність 1000 ... 1111 (8 ... 15) для дешифратора DD2. Тому нумерація виходів F_i, отриманого дешифратора "4 в 16" послідовна від 0 до 15. На рисунку 6.3 приведено умовне позначення, отриманого дешифратора-демультиплексора (у довідниках вони так часто і називаються і розміщуються в одному розділі) [1].

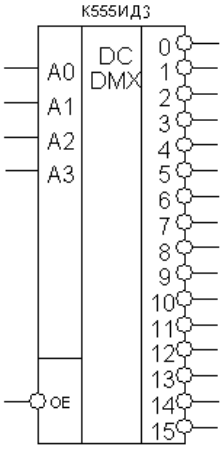


Рисунок 6.3 – Мікросхема дешифратора-демультиплексора

Питання для самоконтролю.

1. Методика побудови цифрових комбінаційних блоків.
2. Основні закони і тотожності булевої алгебри.
3. Принцип спрощення аналітичних рівнянь.

Практичні завдання, задачі.

1. Побудувати в програмному забезпеченні EWB схему демультиплексора «1 в 16» на мікросхемах дешифратора-демультиплексора 74138 (К555ИД7) згідно з рисунком 6.2.

2. Підключити джерело живлення, запрограмувати Word Generator, підключити індикатори до контрольних вузлів схеми

3. Провести аналіз функціонування схеми за допомогою індикаторних пристроїв та Logic Analyzer.

4. Зняти діаграму функціонування демультиплексора «1 в 16».

5. Побудувати таблицю функціонування демультиплексора «1 в 16».

6 Зробити розгорнутий висновок.

Для отримання максимальної оцінки рішення пунктів завдання має бути виконане у повному обсязі, з відповідними поясненнями, оформлене стилістично і технічно грамотно.

Практичне заняття №7

Логічний синтез пристрою двійкового множення
на суматорах та логічних елементах

Метою практичного заняття є вивчення методів синтезу практичних цифрових схем арифметичних логічних пристроїв.

План практичного заняття.

1. Засвоїти принципи побудови схем арифметичних логічних пристроїв.
2. Виконати розрахункові завдання з використанням засобів алгебри логіки.
3. Побудувати структурні та принципові схеми, аналітичні схеми в програмному забезпеченні Electronics Workbench, згідно з завданням, та провести дослідження їх функціонування за допомогою індикаторних пристроїв та діаграм.

Ключові терміни та поняття: суматор, таблиця істинності, рівняння функціонування, сигнал перенесення, множення.

Методичні вказівки до вивчення питань плану.

Двійковими суматорами називають комбінаційні цифрові пристрої, які виконують операції додавання і віднімання двійкових чисел, які подаються на вхід суматора [3].

Сума двох двійкових чисел з числом розрядів N може мати число розрядів (N + 1). Наприклад, при підсумовуванні чисел 13 (1101) і 6 (0110) виходить число 19 (10011). Тому кількість виходів суматора на одиницю більше кількості розрядів вхідного коду. Цей додатковий (старший) розряд називається виходом перенесення

Найпростішою схемою є напівсуматор, який виконує операцію знаходження суми двох однорозрядних слів A і B (A_1 A_2 ... A_i; B_1 B_2 ... B_i - розряди чисел, які складаються) [4].

Роботу напівсуматора описує таблиця 7.1, з якої знаходимо логічні рівняння для суми і для перенесення.

Таблиця 7.1 – Алгоритм роботи напівсуматора

A	B	S	P
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = \overline{A}B + A\overline{B};$$
$$P = AB.$$

При знаходженні суми багаторозрядних слів напівсуматор може використовуватись лише в молодшому розряді.

Для знаходження суми старших розрядів схема повинна мати три входи – два інформаційних і третій – вхід перенесення з молодших розрядів. Схема, яка відповідає цим вимогам, називається повним суматором.

Множення виконується так само, як і для десяткової системи. Розглянемо приклад множення на основі двійкової системи для чисел:

$$10(A_{10}) \times 11(A_{10}) = 110(A_{10})$$

$$\begin{array}{r} 1010 \\ \times\ 1011 \\ \hline 1010 \\ 1010 \\ 0000 \\ 1010 \\ \hline 1101110 \end{array}$$

Питання для самоконтролю.

1. Класифікація суматорів.
2. Принцип арифметичного складання двійкових чисел.
2. Принцип арифметичного віднімання двійкових чисел.
2. Принцип арифметичного множення двійкових чисел.

Практичні завдання, задачі.

Побудувати пристрій двійкового множення на суматорах та логічних елементах І, який дозволяє множити чотирьохрозрядне число А на трьохрозрядне число В.

Після множення трьохрозрядного і чотирьохрозрядного кодів отримується семирозрядний код. Результат операції утворюється за рахунок паралельного множення множеного на кожен розряд множника і складання проміжних добутків з зсувом на один розряд. При підрахуванні добутку виконуються наступні дії (рис. 7.1).

			A4 ×	A3	A2	A1
				B3	B2	B1
			A4B1	A3B1	A2B1	A1B1
		A4B2	A3B2	A2B2	A1B2	
	A4B3	A3B3	A2B3	A1B3		
S7	S6	S5	S4	S3	S2	S1

Рисунок 7.1 – Алгоритм арифметичного множення двійкових кодів

S1 = A1B1; S2 = A1B2 + A2B1; S3 = A1B3 + A2B2 + A3B1; S4 = A2B3 + A3B2 + A4B1; S5 = A3B3 + A4B2; S6 = A4B3; S7 – біт перенесення із попереднього розряду; знак “+” – знак арифметичного додавання.

Однорозрядне множення забезпечується елементами І. Суми можуть бути реалізовані суматорами К155ИМ3. Для реалізації цієї операції на рисунку 7.2 приведена принципова схема.

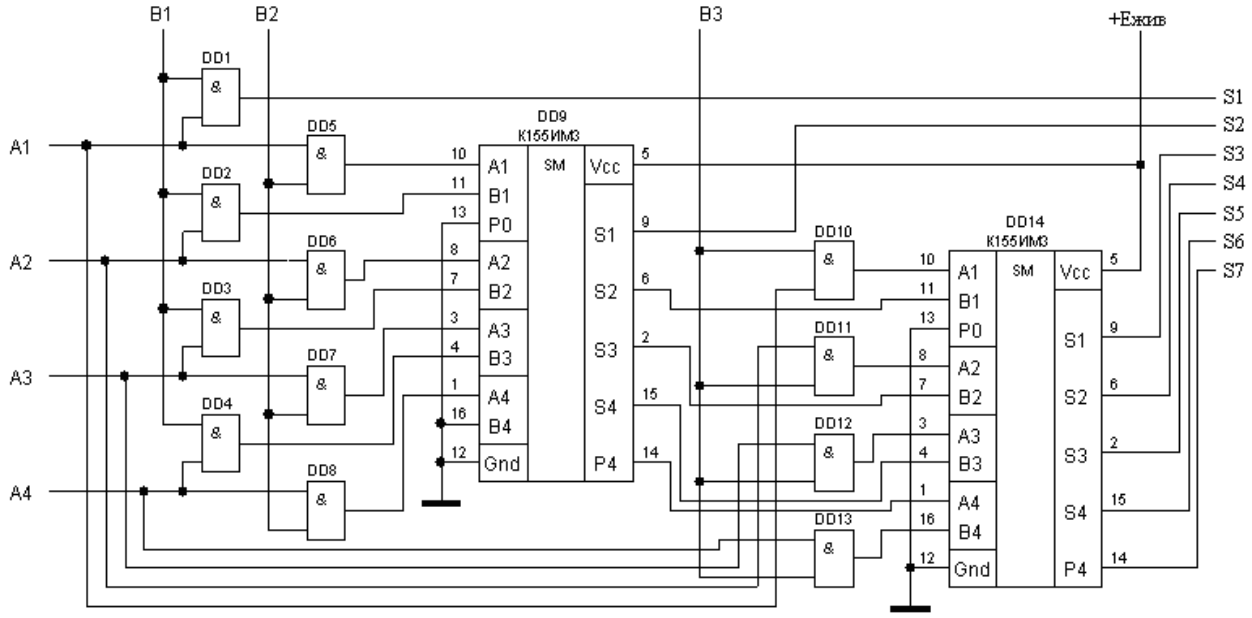


Рисунок 7.2 – Пристрій двійкового множення на суматорах та логічних елементах І

Дослідити схему в програмному застосуванні EWB, виконати приклади множення с проміжною індикацією, отримати діаграми функціонування.

Практичне заняття №8

Розробка кодуючого пристрою на основі сканованої матриці клавіш

Метою практичного заняття є вивчення методів синтезу практичних цифрових схем введення кодованої інформації.

План практичного заняття.

1. Засвоїти принципи побудови схем пристроїв введення інформації.
2. Засвоїти принципи побудови матричної клавіатури.
3. Особливість кодуючого пристрою на основі сканованої матриці

Ключові терміни та поняття: клавіатура, матриця, дешифратор, мультиплексор, кодування.

Методичні вказівки до вивчення питань плану.

Кодуючий пристрій на основі сканованої матриці формує до 256 восьмирозрядних двійкових кодів від клавіатури введення, яка має відповідне число перемикачів [3].

Розглянемо організацію некодуючої клавіатури розмірністю 5×3. Горизонтальні лінії X0…X4 (рядки) являються вихідними, а вертикальні Y0…Y2 (стовпці) являються вхідними (рис. 8.1).

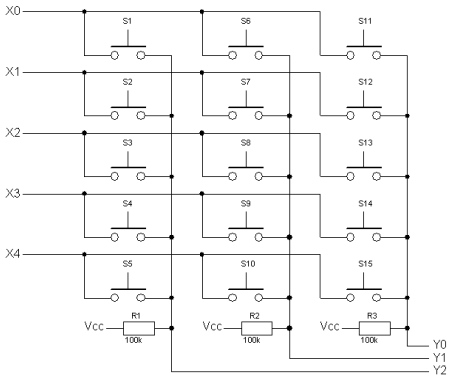


Рисунок 8.1 - Структура матричної клавіатури

У неактивному перебуванні на виходах X0… X4 присутній рівень логічної 1, а входи Y0…Y2 зазвичай підтягнуті до напруги живлення резисторами. Для цієї мети, як правило, використовуються вбудовані резистори, які є в деяких портах мікроконтролерів.

При скануванні на горизонтальних лініях матриці формується сигнал логічного нуля. На останніх виходах зберігається рівень логічної 1. Для кожної фази нуля виконується прочитування і аналіз коду Y0…Y2. При замиканні якої-небудь з клавіш матриці, програмно встановлюється код даної клавіші.

Особливість кодуючого пристрою на основі сканованої матриці – матричне, в 4 рядки на 4 стовпця, включення перемикачів в блоці клавіатури і відповідна організація блоку сканування цієї матриці [6].

Блок сканування утворюють дешифратор і мультиплексом (рис. 8.2). Їх адресні входи підключені до відповідних розрядних виходів лічильника. Сканування забезпечується послідовним опитуванням чотирьох рядків матриці вихідними сигналами з дешифратора і пошуком мультиплексором клавіші натискання у всіх чотирьох місцях для кожного з цих рядків.

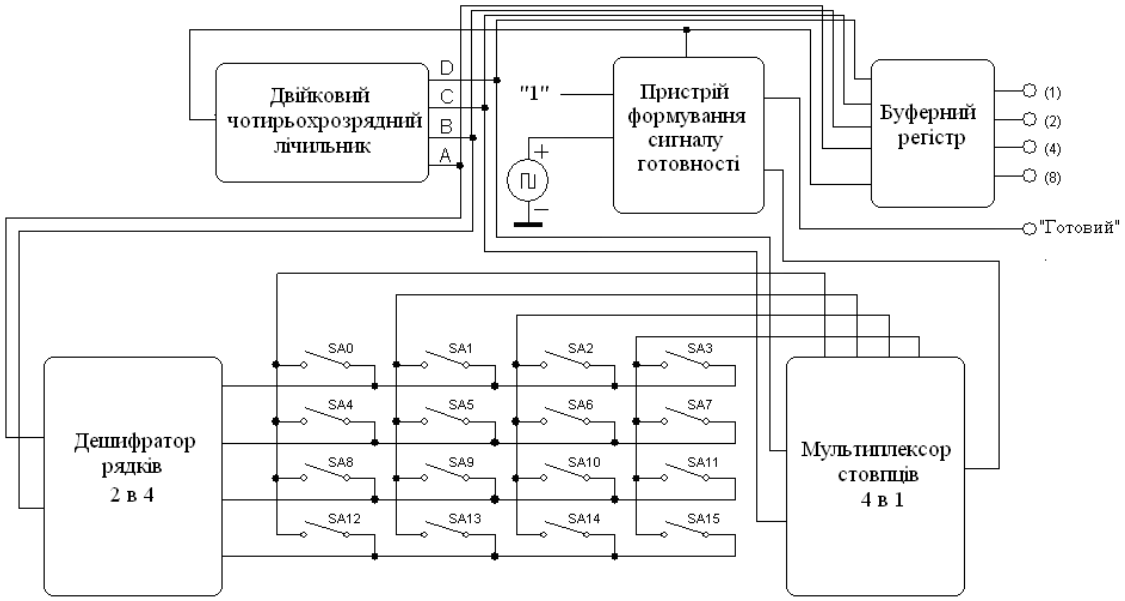


Рисунок 8.2 – Функціональна схема кодуючого пристрою на основі сканованої матриці

Після того, як шляхом сканування визначена натиснута клавіша, виробляється сигнал «Готовий». При цьому лічильник обнуляється і знову відбувається сканування до натиснутої клавіші.

Пристрій формування сигналу готовності складається з генератора імпульсів, комбінаційної схеми на логічних елементах І та І-НЕ и тригера (рис. 8.3).

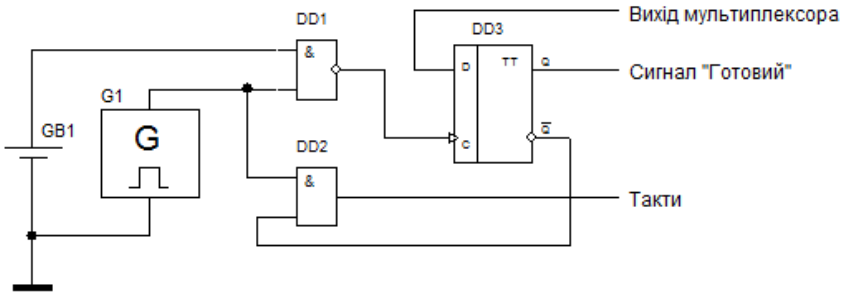
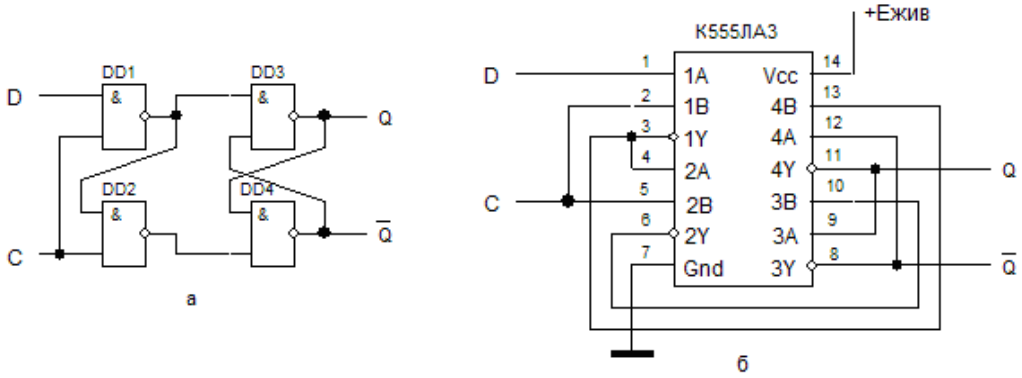


Рисунок 8.3 - Пристрій формування сигналу результату сканування матриці перемикачів

Вихідній код перебирається лічильником до тих пір, поки не буде натиснута необхідна клавіша, при цьому на виході мультиплексора логічна 1, яка записується в тригер DD3, і формує сигнал «Готовій», який відображається індикатором. Відповідно на інверсному виході тригера DD3 – логічний 0. Він поступає на один з входів елементу І (DD2), до іншого входу якого підключений генератор прямокутних імпульсів. Такий стан блокує лічильник, та буферні регістри. Лічильник зупиняє рахунок і на виході схеми фіксується код натиснутої клавіші. При відпусканні клавіші цикл сканування відновлюється за допомогою елемента І-НЕ (DD1). D-тригер, з метою економії елементних ресурсів, можна виконати по схемі представленій на рисунку 8.4.



а – функціональна схема;

б – схема D-тригера на мікросхемі К555ЛА3

Рисунок 8.4 – D-тригер

Схемна модель клавіатури на основі сканованої матриці представлена на рисунку 8.5.

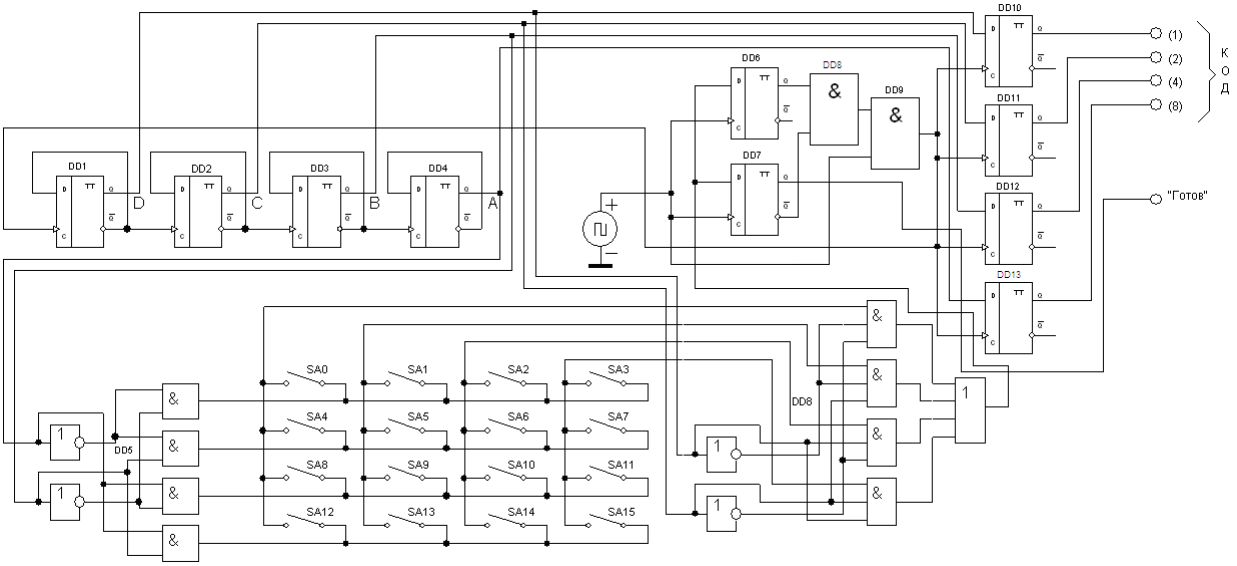


Рисунок 8.5 - Схемна модель клавіатури на основі сканованої матриці

Питання для самоконтролю.

1. Пристрої введення інформації.
2. Клавіатура комбінаційного типу.
3. Клавіатура сканованого типу.
4. Клавіатура на основі сканованої матриці клавіш.
5. Пристрій кодування інформації, який працює за принципом пецептрона.

Практичні завдання, задачі.

1. Виконати розрахункові завдання з використанням засобів алгебри логіки.

1) Розрахувати схему лічильника.
2) Розрахувати схему дешифратора.
3) Розрахувати схему мультиплексора.
4) Побудувати схему пристрою формування сигналу готовності.

2. Розрахувати, побудувати і дослідити функціонування схеми блоку кодування інформації на основі сканованої. Провести дослідження схем функціонування за допомогою індикаторних пристроїв і діаграм.

5. Побудувати схему буферного регістра.

6. Побудувати схему кодуючого пристрою на основі сканованої матриці клавіш на реальній елементній базі.

Для отримання максимальної оцінки рішення пунктів завдання має бути виконане у повному обсязі, з відповідними поясненнями, оформлене стилістично і технічно грамотно.

Практичне заняття №9

Використання постійних запам’ятовувальних пристроїв

Метою практичного заняття є вивчення постійного запам’ятовувального пристрою в пристрої введення цифрової інформації.

План практичного заняття.

1. Засвоїти принципи побудови та використання постійних запам’ятовувальних пристроїв.
2. Засвоїти принципи побудови та використання оперативних запам’ятовувальних пристроїв.
3. Засвоїти принципи побудови та використання оперативних запам’ятовувальних пристроїв.

Ключові терміни та поняття: дешифратор, шифратор, регістр, індикатор, клавіатура.

Методичні вказівки до вивчення питань плану.

Регістри з паралельним прийомом і видачою інформації служать для зберігання інформації і називаються регістрами пам'яті або зберігання [1].

Призначення пам'яті - зберігати двійкову інформацію невеликого об'єму протягом деякого проміжку часу.

Ці регістри є набором тригерів, кожен з яких зберігає один розряд двійкового числа.

У паралельних регістрах кожен з тригерів має свій незалежний інформаційний вхід (D) і свій незалежний інформаційний вихід. Тактові входи (C) всіх тригерів сполучені між собою [3].

В результаті паралельний регістр є багаторозрядним, багатовходовим тригером.

Введення (запис) і виведення (зчитування) інформації в регістр пам'яті виконується одночасно у всіх розрядах паралельним кодом.

Паралельні регістри, у свою чергу, діляться на дві групи:

- регістри, які спрацьовують по фронту управляючого сигналу C (або тактовані регістри);

регістри, які спрацьовують по рівню управляючого сигналу C (або стробовані регістри).

Найчастіше в цифрових схемах використовуються регістри, керовані фронтом (тобто тактовані), проте і стробовані регістри мають своє коло завдань, в яких їх ніщо не може замінити [5].

В разі, якщо тригер має настановні входи, запис даних в регістр здійснюється по настановних входах, то введення забезпечується подачею інформаційних сигналів і управляючого сигналу на установочні входи (R, $\overline{R}$, S, $\overline{S}$) в прямому або зворотному кодах.

Якщо тригер синхронний і не має установочних входів, то він представляє набір тригерів з незалежними інформаційними входами і загальним тактовим входом [1].

Як регістри подібного роду можуть бути використані без додаткових елементів багато типів синхронних тригерів. У таких регістрах інформація подається на інформаційні входи (вхід в тригері D) і запис здійснюється подачею тактового командного імпульсу.

З приходом чергового тактового імпульсу відбувається оновлення записаної інформації.

Якщо тригер асинхронний, то інформація і управляючий сигнал через комбінаторну логіку подаються на інформаційні входи (вхід в тригері D).

Питання для самоконтролю.

1. Методика побудови цифрових комбінаційних блоків.
2. Основні закони і тотожності булевої алгебри.
3. Принцип спрощення аналітичних рівнянь.

Практичні завдання, задачі.

В цифровій схемотехніці для тимчасового зберігання послідовності цифрових даних, які переміщуються, застосовуються послідовністні цифрові пристрої - регістри.

Дослідити схему використання регістра з паралельним введенням і паралельним виведенням інформації для запам’ятовування номера натиснутої клавіші при введенні інформації і фіксації цього номера на індикаторі (рис. 9.1).

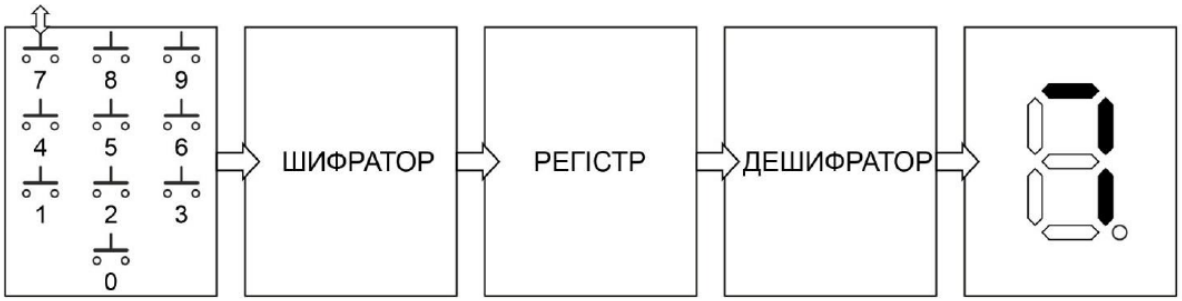


Рисунок 9.1 – Функціональна схема пристрою введення інформації

1. Розрахувати шифратор переведення десяткового коду натиснутої клавіші від 0 до 9 у двійковий чотирьохрозрядний код.
2. Розрахувати дешифратор переведення двійкового чотирьохрозрядного коду натиснутої клавіші від 0 до 9 у код семисегментного індикатора.
3. Регістр пам’яті побудувати на D-тригерах.
4. Провести аналіз функціонування схеми в програмному забезпеченні Electronics Workbench.
5. Побудувати схему електричну принципову пристрою введення інформації на реальній елементній базі.

Для отримання максимальної оцінки рішення пунктів завдання має бути виконане у повному обсязі, з відповідними поясненнями, оформлене стилістично і технічно грамотно.

Практичне заняття №10

Проектування цифрового послідовністного пристрою на регістрі зсуву

Метою практичного заняття є вивчення методів синтезу генераторів кодових послідовностей на регістрах і логічних елементах.

План практичного заняття.

1. Послідовні регістри (регістри зсуву).
2. Послідовний регістр на мікросхемах К155ТМ2.
3. Генератор кодових послідовностей на регістрі зсуву.

Ключові терміни та поняття: регістр, розрядність, перемикальний граф, діаграма термів, рівняння функціонування.

Методичні вказівки до вивчення питань плану.

У регістрі з послідовним введенням виконується послідовний зсув інформації, яка поступає на вхід, на один розряд управо в кожному такті синхросигналів (рис. 10.1) [1].

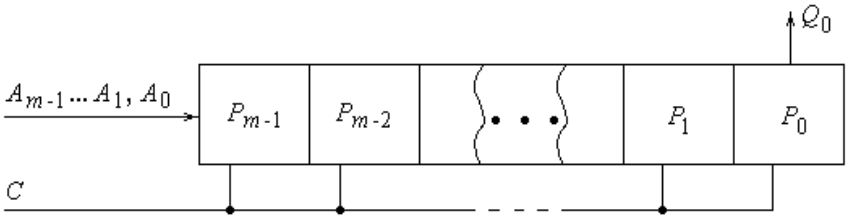


Рисунок 10.1 – Структурна схема регістра зсуву

Після вступу n-синхроімпульсів весь регістр виявляється заповненим розрядами числа А, і перший розряд числа (А0) з'являється на виході Q0 регістра. Протягом подальших n-синхроімпульсів виконується послідовний порозрядний вивід з регістра записаного числа, після чого регістр виявляється повністю очищеним.

Паралельний регістр зсуву являється, як правило, універсальним і може виконувати всі доступні для регістрів мікрооперації. Для цього розрядні схеми, які входять в його склад, сполучені між собою [4].

Регістри зсуву є колом послідовно включених D – тригерів або RS-тригерів і JK-тригерів, включених в режим D-тригера. Поява імпульсу на тактовому вході регістра зсуву викликає переміщення записаної в ньому інформації на один розряд управо або вліво. Як і інші регістри, регістри зсуву використовуються для запису, зберігання і видачі інформації, але основним їх призначенням є перетворення послідовного коду в паралельний або паралельного в послідовний. Схема чотирьохрозрядного регістра зсуву приведена на рисунку 10.2 [1].

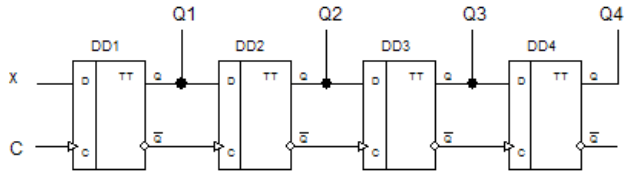


Рисунок 10.2 – Чотирьохрозрядний регістр зсуву

Завдяки тому, що вихід попереднього розряду сполучений з входом D подальшого, кожен тактовий імпульс встановлює подальший тригер в стан, в якому до цього знаходився попередній. Так здійснюється зсуву інформації управо.

С - вхід синхронізації. По імпульсу на цьому вході виконується запис даних в перший тригер. У свою чергу за цей такт попередня інформація першого тригера переписується (зсувається) в другій, з другого - в третій і так далі Таким чином здійснюється зсув інформації на один розряд. Вихідні дані регістра з'являються на виходах Q_i.

Вхід D першого тригера служить для прийому в регістр вхідної інформації X у вигляді послідовного коду. З кожним тактовим імпульсом на цей вхід повинен подаватися код нового розряду вхідної інформації (рис. 10.3).

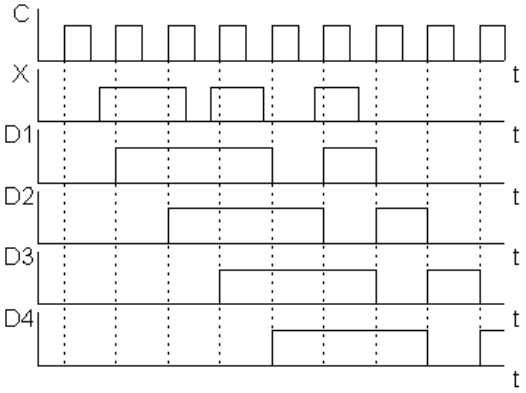


Рисунок 10.3 – Діаграма функціонування чотирьохрозрядного регістра зсуву

Першим тактовим імпульсом перша одиниця старшого розряду числа X записується в перший тригер. Другим тактовим імпульсом в перший тригер записується значення наступного розряду, а в другий тригер записується одиниця, яка була перед приходом другого тактового імпульсу на виході першого тригера.

Таким чином, кожним тактовим імпульсом в регістрі відбувається зсуву числа на один розряд. Чотирьохрозрядне число буде повністю записано в регістр після четвертого тактового імпульсу. При цьому на виході регістра можна прорахувати значення розряду, який був записаний першим. Для прочитування значень в послідовному коді потрібно подати ще чотири тактові імпульси. У двійковій системі числення при зрушенні числа на один розряд у бік старших розрядів відбувається збільшення числа в два рази. При зрушенні числа у бік молодших розрядів число записується в регістр і зменшується в два рази. Таким чином зруховий регістр можна використовувати для множення або ділення числа на 2^n, де n – кількість зрушень рівна кількості тактових імпульсів.

Питання для самоконтролю.

1. Як будуються послідовні регістри зсуву?
2. Які режими роботи регістрів?
3. Методика розрахунку генераторів кодових послідовностей на регістрі зсуву.

Практичні завдання, задачі.

Проєктування генератора кодових послідовностей на регістрі зсуву для реалізації заданої кодової циклічної послідовності **N=(010111)**.

Спочатку необхідно визначити розрядність регістра: 010111010111

$$\underset{1\,2\,1\,3}{\underline{01}0111010111}$$

$n = 2$ $\quad 1 \to 2 \to 1 \to 3$

$$\underset{2\,5\,3\,7\,6\,5\,2}{\underline{010}1110\underline{1}0111}$$

$n = 3$ $\quad 2 \to 5 \to 3 \to 7 \to 6 \to 5 \to 2$

$$\underset{5\,11\,7\,14\,13\,10\,5\,11}{\underline{0101}1101\underline{0}111}$$

$n = 4$ $\quad \underline{5 \to 11 \to 7 \to 14 \to 13 \to 10} \to 5 \to 11$

При n = 4 отримали замкнений цикл роботи пристрою, тобто основний перемикальний граф. Премикальний граф представлений на рисунку 10.4.

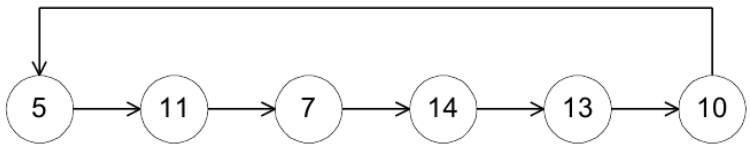


Рисунок 10.4 - Основний перемикальний граф роботи пристрою

В основному робочому циклі наявні 6 станів, тобто M = 6. У цьому випадку кількість надлишкових (не використовуваних) станів складає

$$M_н = 2^n - M = 2^4 - 6 = 16 - 6 = 10.$$

Для перемикального графу (рис. 10.4) складемо діаграму термів, яку заповнюємо наступним чином.

1. Якщо перехід з деякого стану відбувається до парного стану, то в клітинку на діаграмі термів, номер якої відповідає стану, з якого відбувається перехід, записують 0.

2. Якщо перехід з деякого стану відбувається до непарного стану, то в клітинку на діаграмі термів, номер якої відповідає стану, з якого відбувається перехід, записують 1.

3. До клітинок, номер яких відсутній на основному перемикальному графі, записують × (або 0, або 1).

Отримана для даного випадку діаграма термів наведена на рисунку 10.5.

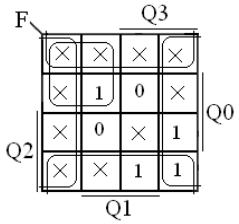


Рисунок 10.5 – Діаграма термів регістра зсуву

На діаграмі необхідно об’єднати одиничні клітинки з якомога більшою кількістю клітин, позначених символом × (при цьому клітини з номерами 0 або 2^n - 1 (в нашому випадку 15) не можуть входити до об’єднань.

Отримана при цьому функція зсуву буде мати такий вигляд:

$$F = \overline{Q1} + \overline{Q0} + \overline{Q3}\,\overline{Q2} = \overline{\overline{Q1} + \overline{Q0} + \overline{Q3}\,\overline{Q2}} = \overline{\overline{Q1}} \cdot \overline{\overline{Q0}} \cdot \overline{\overline{Q3}\,\overline{Q2}} = \overline{Q1 \cdot Q0 \cdot \overline{Q3}\,\overline{Q2}}$$

Реалізована за функцією зсуву схема генератора кодових послідовностей представлена на рисунку 10.6.

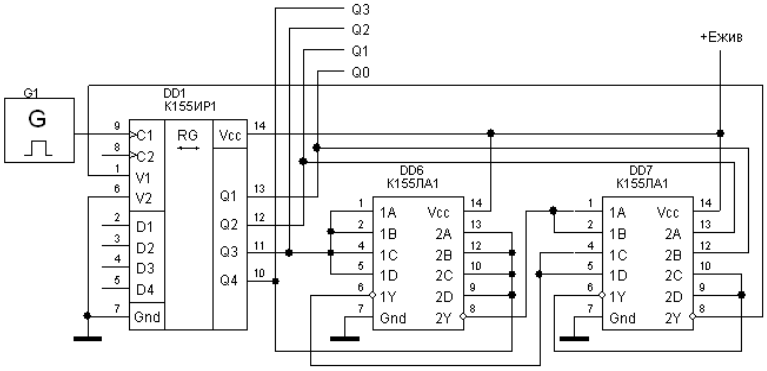


Рисунок 10.6 – Схема генератора кодових послідовностей

Часові діаграми роботи пристрою приведені на рисунку 10.7.

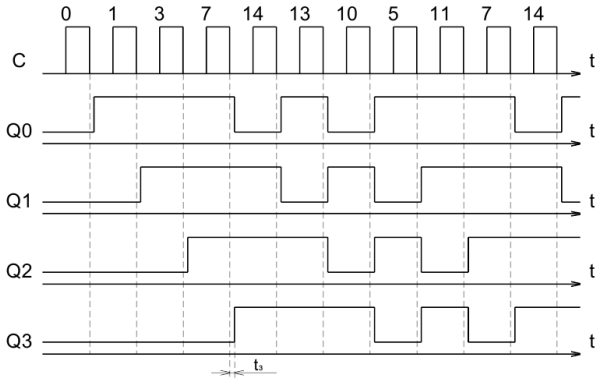


Рисунок 10.7 – Часові діаграми перемикання генератора кодових послідовностей

Варіанти завдань.

1. Реалізувати розраховану схему генератора кодових послідовностей в програмному забезпеченні Electronics Workbench.
2. Регістр зсуву реалізувати на мікросхемах 7474 (К555ТМ2).
3. Зняти діаграму функціонування генератора кодової послідовності на регістрі і логічних елементах за допомогою Logic Analyzer.
4. Зробити аналітичний висновок.

Варіант 1
N = 100111
Варіант 2
N = 101011
Варіант 3
N = 001111

Практичне заняття №11

Пристрої послідовністного типу призначені для спрацювання в заданих кодах

Метою практичного заняття є вивчення методів синтезу послідовністних схем для підрахунку цифрової інформації, представленої в заданих кодах.

План практичного заняття.

1. Лічильник прямого рахунку.
2. Лічильник зворотного рахунку.
3. Лічильники, які працюють у двійковому коді з заданою вагою розрядів.

Ключові терміни та поняття: тригер, таблиця істинності, рівняння функціонування, лічильник, цифровий код.

Методичні вказівки до вивчення питань плану.

Лічильником називається пристрій послідовністного типу, призначений для підрахунку числа імпульсів, які поступають на його вхід, і фіксації цього числа у вигляді коду, який зберігається в тригерах [1].

Лічильники - це цифрові автомати, внутрішні стани яких визначаються лише кількістю сигналів "1", які прийшли на вхід. Сигнали "0" не змінюють їх внутрішні стани [3].

Тригер Т-типу є простим лічильником, який рахує до двох. Лічильник, утворений колом з m тригерів, зможе підраховувати в двійковому коді 2^m вхідних імпульсів. Кожен з тригерів в цьому колі називають розрядом лічильника. Для установки початкового стану лічильника (скидання в нуль) зазвичай передбачається вхід скидання [4].

Основна характеристика лічильника – модуль рахунку, або ємність лічильника $К_{рах.}$. Це кількість вхідних сигналів, які повертають лічильник у вихідний стан. Лічильник, який не має додаткових зв'язків, має модуль рахунку $К_{рах.} = 2^n$. Лічильники, які мають модуль рахунку 2^n, називаються двійковими. Якщо $К_{рах.} \neq 2^n$, то лічильник називається недвійковим. Лічильники відрізняються один від одного кодом, в якому вони працюють. Код завжди буває двійковим, але може мати різні ваги розрядів, наприклад вага 8–4–2–1 або 5–2–1–1 і тому подібне. Одним з недвійкових є двійково-десятковий лічильник, в якому значення кожного розряду десяткового числа кодується двійковим кодом.

За призначенням лічильники можуть бути підсумовуючими, віднімаючими і реверсивними. Підсумовуючі лічильники виконують складання числа імпульсів, які поступають на вхід, з тим числом, яке зберігалося в ньому. Віднімаючі лічильники виконують віднімання числа імпульсів, які поступають, з початкового числа, записаного в ньому заздалегідь. Реверсивні лічильники можуть виконувати як додавання, так і віднімання імпульсів, які поступають на вхід, залежно від управляючих сигналів, які змінюють режим роботи лічильника [5].

За способом організації внутрішніх зв'язків лічильники можуть бути: з послідовним перенесенням, з паралельним перенесенням, з комбінованим перенесенням, кільцеві. Послідовний характер роботи є причиною двох недоліків послідовного лічильника: менша швидкість рахунку в порівнянні з паралельними лічильниками і можливість появи помилкових сигналів на виході схеми.

Допустима швидкість рахунку в лічильниках обох типів визначається максимальною швидкістю перемикання одного тригера. Визначаючи максимальну швидкість рахунку послідовного лічильника, слід враховувати найбільш несприятливий випадок зміни стану всіх m-тригерів. Сумарну тривалість перехідного процесу можна визначити як суму часів запізнювання окремих елементів, які складають тригери, і часів спрацьовування всіх тригерів. Наприклад, в лічильнику, який веде рахунок в чотирьохрозрядному двійковому коді 8–4–2–1, при переході від числа 0111 до 1000 на виході з'явиться наступна послідовність сигналів: 0111 → 0110 → 0100 → 0000 → 1000. Це означає, що при переході із стану 7 в стан 8 на виходах лічильника на короткий час з'являться стани 6; 4; 0. Ці додаткові стани можуть викликати неправильну роботу інших пристроїв.

Питання для самоконтролю.

1. Двійкові підсумовуючі лічильники з послідовним перенесенням.
2. Двійкові підсумовуючі лічильники з паралельним перенесенням.
3. Недвійкові лічильники з паралельним перенесенням.

Практичні завдання, задачі.

Потрібно синтезувати синхронний лічильник, який працює в коді 5–2–1–1. Відповідно до заданого коду заповнюємо ліву частину таблиці 11.1 функціонування лічильника (стовпці Q4, Q3, Q2, Q1).

Таблиця 11.1 – Таблиця функціонування лічильника, що працює в коді 5–2–1–1

n	$Q4^n$	$Q3^n$	$Q2^n$	$Q1^n$	$Q4^{n+1}$	$Q3^{n+1}$	$Q2^{n+1}$	$Q1^{n+1}$
0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	1
2	0	0	1	1	0	1	0	1
3	0	1	0	1	0	1	1	1
4	0	1	1	1	1	0	0	0
5	1	0	0	0	1	0	0	1
6	1	0	0	1	1	0	1	1
7	1	0	1	1	1	1	0	1
8	1	1	0	1	1	1	1	1
9	1	1	1	1	0	0	0	0

У таблиці n – номер стану лічильника. Він змінюється на наступний стан при подачі кожного рахункового імпульсу. Q_1^{n+1}, Q_2^{n+1}, Q_3^{n+1}, Q_4^{n+1} – логічні змінні на виході чотирьох тригерів. Перший тригер з виходом Q_1^n відповідає першому молодшому розряду. Складемо карти Карно для функцій переходів тригерів кожного розряду (рис. 11.1).

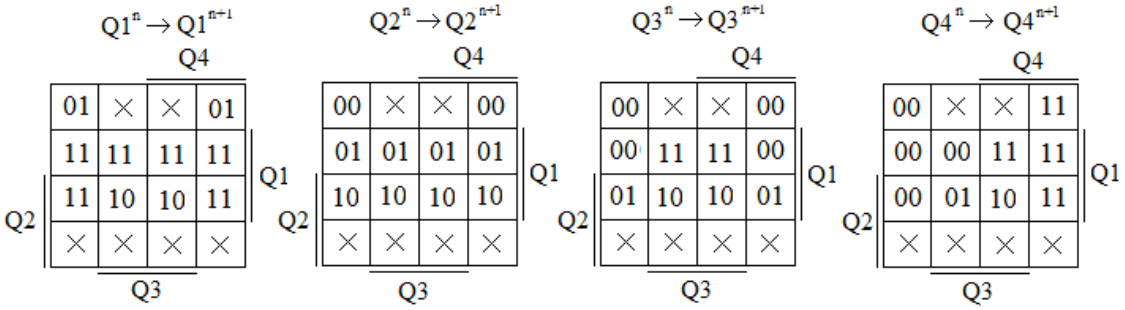


Рисунок 11.1 – Карти функцій переходів тригерів лічильника

Використовуючи словник переходів JK – тригера (рис. 11.2), для кожного входу тригера складемо карти Карно (рис. 11.3).

Перехід	J	K
0 → 0	0	×
0 → 1	1	×
1 → 0	×	1
1 → 1	×	0

Рисунок 11.2 – Словник переходів JK-тригера

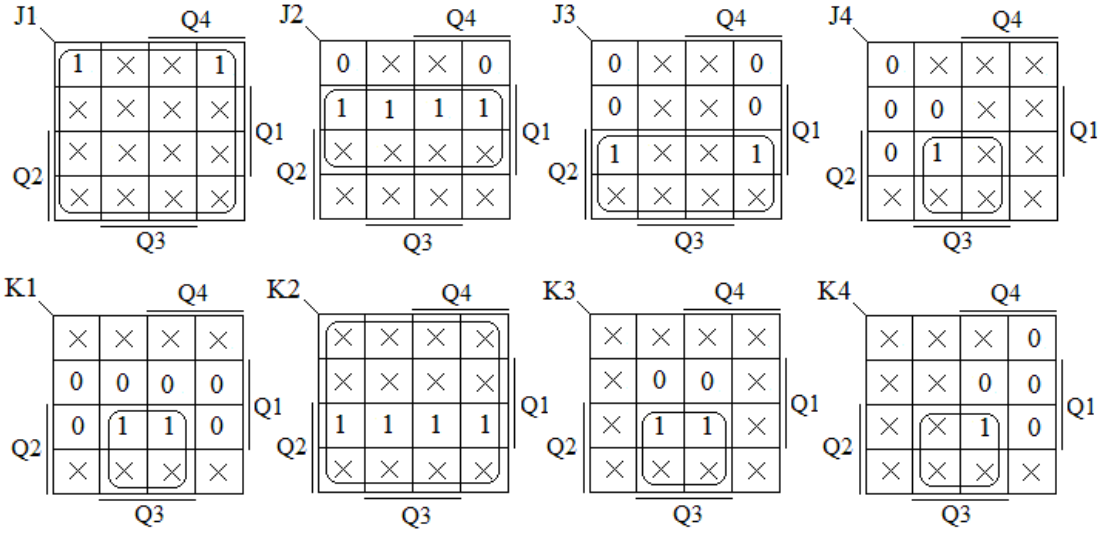


Рисунок 11.3 – Карти Карно функцій входів тригерів лічильника

Проведемо мінімізацію логічних функцій входів в картах Карно з метою отримання їх аналітичних виразів.

У результаті отримано наступні функції входів тригерів лічильника:

$$J1 = 1; \quad J2 = Q1; \quad J3 = Q2; \quad J4 = Q2Q3$$

$$K1 = Q2Q3; \quad K2 = 1; \quad K3 = Q2Q3; \quad K4 = Q2Q3$$

Схема синтезованого лічильника представлена на рисунку 11.4.

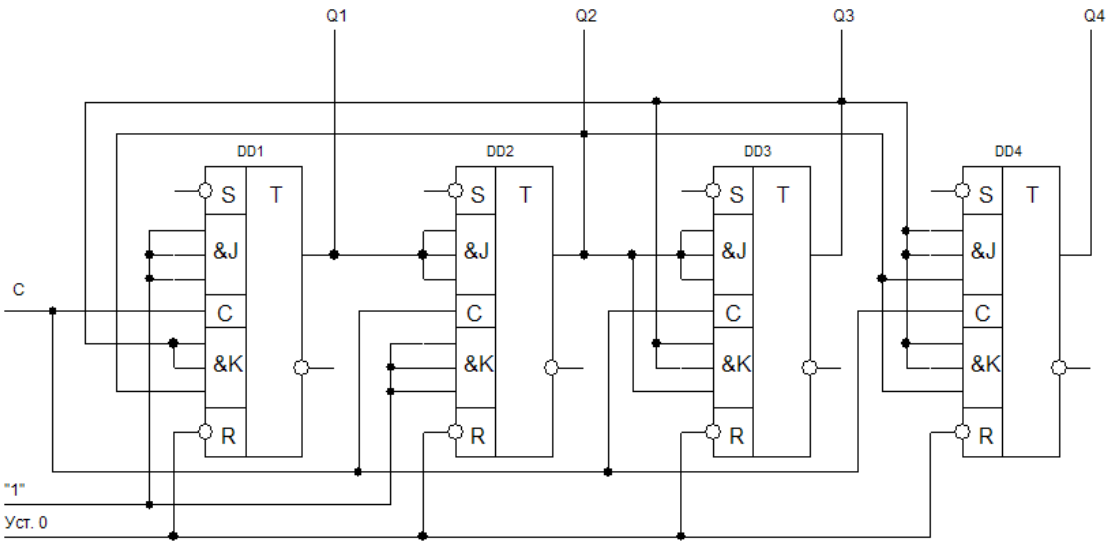


Рисунок 11.4 – Синхронний лічильник, який працює в коді 5–2–1–1

Варіанти завдань.

Варіант 1

Розрахувати і побудувати лічильник з $К_{ліч.}$ = 6, який працює в коді Грея (на JK-тригерах).

Варіант 2

Розрахувати і побудувати лічильник з $К_{ліч.}$ = 8, який працює в додатковому коді (на JK-тригерах).

Варіант 3

Розрахувати і побудувати лічильник, який працює в коді Айкена (2-4-2-1) (на JK-тригерах).

Варіант 4

Розрахувати і побудувати з лічильник з $К_{ліч.}$ = 10, який працює в коді з надлишком 3 (на JK-тригерах).

Варіант 5

Розрахувати і побудувати десятковий лічильник (на JK-тригерах).

Варіант 6

Розрахувати і побудувати лічильник з $К_{ліч.}$ = 8, який працює в коді Грея (на D-тригерах).

Варіант 7

Розрахувати і побудувати лічильник з $К_{ліч.}$ = 10, який працює в додатковому коді (на D -тригерах).

Варіант 8

Розрахувати і побудувати лічильник, який працює в коді Айкена (2-4-2-1) (на D -тригерах).

Практичне заняття №12

Пристрої послідовністного типу призначені для генерації сигналів

Метою практичного заняття є вивчення методів синтезу генераторі числових послідовностей на послідовних лічильниках.

План практичного заняття.

1. Засвоїти методику синтезу генераторів числових послідовностей на послідовних лічильниках

2. Засвоїти методику синтезу генераторів числових послідовностей на регістрах зсуву.

Ключові терміни та поняття: таблиця станів, таблиця істинності, рівняння функціонування, функціональні блоки, числові послідовності.

Методичні вказівки до вивчення питань плану.

Послідовністні пристрої - генератори чисел, які називають також розподільниками сигналів, оскільки послідовність двійкових чисел на їх виходах використовується для управління роботою інших цифрових вузлів [1].

Число станів генератора називається довжиною послідовності чисел L_n, яка дорівнює кількості тактів, після якої послідовність чисел на виході генератора повторюється.

По своїй структурі генератори чисел близькі або до лічильників, або до регістрів [4].

Будь-який лічильник можна вважати генератором послідовності чисел, яка має $L_n = K_{рах.}$

Як правило, необхідне число розрядів генератора дорівнює числу двійкових розрядів m в числах, які генеруються. Якщо $m > \log_2 L_n$, то для зменшення числа використовуваних тригерів, структура генераторів змінюється. В цьому випадку генератор доцільніше будувати у вигляді лічильника з модулем рахунку $K_{рах.} = L_n$ і підключеною до його виходів комбінаційною схемою, яка синтезує необхідні значення двійкових чисел послідовності [1].

Питання для самоконтролю.

1. Генератори числових послідовностей на регістрах зсуву.
2. Генератори числових послідовностей на лічильниках.

Практичні завдання, задачі.

Розглянемо генератор чисел N1-N2-N3-N4 (N1 – N4 – десяткові числа.

Оскільки $L_n = 4$, то за основу генератора можна взяти двохрозрядний лічильник $K_{рах.} = 4$, який генерує числа 0–1–2–3. Підключивши до виходів лічильника комбінаційну схему, яка виконує перетворення коду відповідно до таблиці функціонування, отримаємо структуру генератора, який створює задану послідовність чисел.

Завдання.

Провести синтез генератора числової послідовності 4-3-2-12.

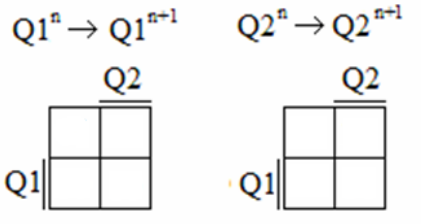
1. Для побудови генератора використати лічильник, побудований на двох JK – тригерах (використати схему віднімаючого лічильника).

2. Скласти таблицю переходів станів розрядів лічильника (табл. 12.1).

Таблиця 12.1 - Таблиця переходів станів розрядів лічильника

n	$Q2^n$	$Q1^n$	$Q2^{n+1}$	$Q1^{n+1}$
0				
1				
2				
3				

3. Скласти карти Карно функцій переходів тригерів генератора (рис. 12.1).



Перехід	J	K
0 → 0	0	×
0 → 1	1	×
1 → 0	×	1
1 → 1	×	0

Рисунок 12.1 - Карти Карно функцій переходів тригерів генератора

4. Скласти карти Карно функцій входів тригерів лічильника (рис. 12.2).

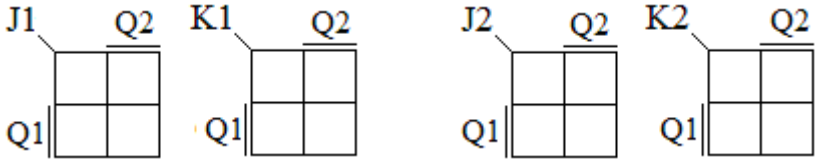


Рисунок 12.2 - Карти Карно функцій входів тригерів лічильника

5. Виконати мінімізацію логічних функцій входів в картах Карно з метою отримання їх аналітичних виразів. У результаті отримати наступні функції входів тригерів лічильника генератора.

J1 = ; K1 = ; J2 = ; K2 = .

6. Скласти логічну схему лічильника генератора чисел.

7. Провести аналіз функціонування лічильника генератора чисел.

8. Отримати діаграму функціонування.

9. Провести аналіз функціонування і зробити розгорнутий висновок.

Для отримання максимальної оцінки рішення пунктів завдання має бути виконане у повному обсязі, з відповідними поясненнями, оформлене стилістично і технічно грамотно.

Література, інформаційні джерела

Перелік використаної літератури

1. Цифрова схемотехніка : підручник / Л. Л. Верьовкін, М. В. Світанько, Є. М. Кісельов та ін. Запоріжжя : ЗДІА, 2016. 214 с.

2. Рябенький В. М., Жуйков В. Я., Гулий В. Д. Цифрова схемотехніка : навч. посіб. Львів : Новий Світ-2000, 2019. 736 с.

3. Комп'ютерна схемотехніка та архітектура комп'ютерів : навч. посіб. / О. В. Задерейко, Н. І. Логінова, О. Г. Трофименко та ін. Одеса : Фенікс, 2021. 163 с. URL: https://hdl.handle.net/11300/14473 (дата звернення: 01.04.2026).

4. Мірошник М. А., Клименко Л. А., Корольова Я. Ю. Технології та автоматизація проєктування цифрових пристроїв складних комп'ютерних систем на ПЛІС : навч. посіб. Харків : УкрДУЗТ, 2021. 220 с.

5. Схемотехніка: пристрої цифрової електроніки : підручник. Т. 1. / В. М. Рябенький, В. Я. Жуйков, Ю. С. Ямненко та ін. Київ : НТУУ КПІ, 2016. 399 с. URL:https://eds.kpi.ua/wpcontent/uploads/2015/05/CT_tom_1_електронне_виданн я.pdf (дата звернення: 01.04.2026).

6. Цирульник С. М., Лисенко Г. Л. Проектування мікропроцесорних систем : навч. посіб. Вінниця : ВНТУ, 2010. 201 с.

7. Михайлов С. Р. Мікропроцесорна техніка. Однокристальні мікроконтролери : навч. посіб. Київ : НТУУ КПІ, 2014. 123 с.

Перелік рекомендованої літератури

1. Бондаренко М. Ф., Білоус Н. В., Руткас А. Г. Комп'ютерна дискретна математика : підручник. Харків : Компанія СМІТ, 2004. 480 с.

2. Гавриленко С. Ю., Клименко А. М., Носков В. І. Логіка дискретних автоматів : навч.-метод. посіб. Харків : НТУ ХПІ, 2014. 129 с.

3. Гавриленко С. Ю., Клименко А. М., Любченко Н. Ю. Теорія цифрових автоматів та формальних мов : навч. посіб. Харків : НТУ ХПІ, 2010. 176 с.

4. Гавриленко С. Ю., Клименко А. М., Гоготов В. В. Основи комп'ютерної техніки. Харків : НТУ ХПІ, 2008. 272 с.

5. Матвієнко М. П. Комп'ютерна логіка. Київ : Ліра-К, 2012. 288 с.

6. Самофалов К. Г. Прикладна теорія цифрових автоматів. Київ : Вища школа, 1987. 344 с.

7. Czerwinski R., Kania D. Finite State Machine Logic Synthesis for Complex Programmable Logic Devices. Berlin : Springer, 2013. 172 p.

8. Roger Tokheim, Patrick Hoppe. Digital Electronics : Principles and Applications. 9th edition. New York, USA : McGraw Hill LCC, 2022. 576 p.

Перелік інформаційних джерел

1. 1. Наукова бібліотека Запорізького національного університету. URL: http:// https://library.znu.edu.ua (дата звернення: 01.04.2026)

2. Система електронного забезпечення навчання ЗНУ. URL: https://moodle.znu.edu.ua/course/view.php?id=8576 (дата звернення: 01.04.2026).

3. Електронний архів наукових та освітніх матеріалів КПІ ім. Ігоря Сікорського. Факультет електроніки. URL: https://ela.kpi.ua/handle/123456789/66 (дата звернення: 01.04.2026).

4. Електронний архів наукових та освітніх матеріалів КПІ ім. Ігоря Сікорського. Радіотехнічний факультет URL: https://ela.kpi.ua/handle/123456789/61(дата звернення: 01.04.2026).

5. Інституційний репозитарій Тернопільського національного технічного університету ім. Івана Пулюя URL: http://ehttps://elartu.tntu.edu.ua (дата звернення: 01.04.2026).

Навчально-методичне видання
(українською мовою)

Верьовкін Леонід Леонідович,
Світанько Микола Вікторович

ФУНКЦІОНАЛЬНІ ВУЗЛИ МІКРОПРОЦЕСОРНИХ СИСТЕМ

Методичні рекомендації до практичних занять
для здобувачів ступеня вищої освіти бакалавра спеціальності
G5 «Електроніка, електронні комунікації, приладобудування та радіотехніка»
освітньо-професійної програми
«Мікро- та наносистемна техніка»

Рецензент *В. Л. Коваленко*
Відповідальний за випуск *Т. В. Критська*
Коректор *Я. О. Коваленко*