

УДК 621.38 (075)

В377

Міністерство освіти і науки України
Інженерний інститут Запорізького національного університету

Л.Л. Верьовкін
М.В. Світанько

ЦИФРОВА СХЕМОТЕХНІКА

Методичні рекомендації до самостійної роботи
для здобувачів вищої освіти бакалавра
спеціальності «Мікро- та наносистемна техніка»
освітньо-професійної програми
«Мікро- та наносистемна техніка»

Міністерство освіти і науки України
Інженерний інститут Запорізького національного університету

Л.Л. Верьовкін, М.В. Світанько

ЦИФРОВА СХЕМОТЕХНІКА

Методичні рекомендації до самостійної роботи
для здобувачів вищої освіти бакалавра
спеціальності «Мікро- та наносистемна техніка»
освітньо-професійної програми
«Мікро- та наносистемна техніка»

Затверджено
вченою радою ЗНУ
Протокол №
від

Запоріжжя
2020

УДК 621.38 (075)

B377

Верьовкін Л.Л., Світанько М.В. Цифрова схемотехніка: Методичні рекомендації до самостійної роботи для здобувачів вищої освіти бакалавра спеціальності «Мікро- та наносистемна техніка» освітньо-професійної програми «Мікро- та наносистемна техніка». Запоріжжя : ЗНУ, 2020. 49 с.

Методичні рекомендації до самостійної роботи з дисципліни «Цифрова схемотехніка» розроблено для студентів денної та заочної форм навчання спеціальності «Мікро- та наносистемна техніка». Запропоновані завдання призначені для освоєння теоретичного матеріалу, опанування навичок вирішення практичних задач, вирішення контрольної роботи та курсового проекту.

Рецензент

Т.В. Критська, доктор технічних наук, професор кафедри електронних систем

Відповідальний за випуск

М.В. Світанько, кандидат фізико-математичних наук, доцент, завідувач кафедри мікроелектронних інформаційних систем.

Зміст

Вступ	4
Зміст самостійної роботи	5
1 Арифметичні і логічні основи цифрової схемотехніки	5
Питання для закріплення вивченого матеріалу та самоконтролю	14
Практичні завдання	14
2 Цифрові пристрої комбінаційного типу	15
Питання для закріплення вивченого матеріалу та самоконтролю	22
Практичні завдання	22
3 Цифрові пристрої послідовнісного типу	23
Питання для закріплення вивченого матеріалу та самоконтролю	32
Практичні завдання	32
4 Синтез цифрових програмних автоматів	33
Питання для закріплення вивченого матеріалу та самоконтролю	37
Практичні завдання	37
5 Запам'ятовуючі пристрої	38
Питання для закріплення вивченого матеріалу та самоконтролю	44
Практичні завдання	44
6 Питання для повторення та актуалізації знань	45
ЛІТЕРАТУРА	47

Вступ

Метою викладання дисципліни «Цифрова схемотехніка» є надання студентам повної уяви про елементну базу, принципи функціонування, характеристики і схемотехніку сучасних нано- та мікроелектронних приладів і мікросхем.

Основними **завданнями** вивчення дисципліни «Цифрова схемотехніка» є: ознайомлення студентів з принципами використання алгебри логіки при побудові електронних схем, розрахунком та застосуванням приладів комбінаційного та послідовнісного типів; знання особливостей застосування логічних цифрових програмних автоматів; вміння практично застосовувати елементи пам'яті різних функціональних ознак.

У результаті вивчення навчальної дисципліни студент повинен

знати:

- логічні та арифметичні основи цифрової схемотехніки;
- елементну базу статичних цифрових мікросхем;
- принципи побудови пристроїв комбінаційного типу;
- принципи побудови пристроїв послідовнісного типу;
- організацію схемотехніки запам'ятовуючих пристроїв.

вміти:

- виконувати логічний синтез цифрових пристроїв;
- розраховувати схемотехніку пристроїв комбінаційного типу;
- розраховувати схемотехніку пристроїв послідовнісного типу;
- виконувати аналіз схем з метою їх спрощення та удосконалення.

Мікро- та наносистемна техніка є базовою галуззю народного господарства з широким використанням засобів автоматизації на основі сучасної техніки.

У сучасних системах автоматизації і управління знаходять широке вживання засоби обробки даних, основу яких складають цифрові елементи і пристрої. Цифрова електроніка за короткий історичний період пройшла дорогу від простих пристроїв на дискретних елементах до пристроїв і цілих систем, що реалізуються на понад великих інтегральних схемах. Розвиток елементної бази змінив і підходи до розрахунку і проектування цифрових пристроїв і систем, які базуються на використанні функціонально і конструктивно закінчених елементів і пристроїв у вигляді цифрових інтегральних схем.

Цифрові пристрої приймають, обробляють і формують дискретно змінну інформацію, на відміну від аналогових пристроїв, що працюють з безперервно змінною інформацією.

Для опису роботи цифрових пристроїв використовують самі різні моделі, що відрізняються одна від одної складністю, що відображають з необхідною точністю характеристики приладу в широкому діапазоні напруги, струмів і температур, що мають відповідність між електричною напругою і фізичними процесами, що протікають в приладі. В основному ці моделі використовуються при комп'ютерних розрахунках цифрових схем. В даний час існують комп'ютерні програми, які не лише розраховують готові схеми, але здатні і проектувати нові схеми по формалізованим описам функцій, які даний пристрій повинен виконувати. Але навіть незалежно від того, що це достатньо зручно,

жодна програма ніколи не може порівнятися з людиною. По-справжньому ефективні, мінімізовані по апаратним засобам і красиві схеми може розробляти лише людина, яка завжди підходить до проектування творчо і використовує оригінальні ідеї.

Розробник цифрових пристроїв мікро- та наносистемної техніки теж використовує своєрідні моделі, або різні рівні представлення цифрових схем. Але, на відміну від комп'ютера, людина може гнучко вибирати потрібну модель. Логічний аналіз дозволяє зрозуміти, де досить простої моделі, а де потрібна складніша. Тобто людина ніколи не робитиме зайвої, надлишкової роботи і, отже, не вноситиме додаткових помилок, властивих будь-який, навіть найскладнішій, моделі.

Методичні рекомендації містять індивідуальні контрольні завдання з курсу «Цифрова схемотехніка» та приклади розв'язання завдань кожної теми. Для підготовки до лекцій, практичних занять та самостійного засвоєння матеріалу студенти можуть використати теоретичні відомості з курсу та приклади вирішення задач.

Зміст самостійної роботи

1 Арифметичні і логічні основи цифрової схемотехніки

Метою вивчення теми є засвоєння арифметичних і логічних операцій у двійковій системі счислення, теоретичних основ алгебри логіки.

Ключові терміни та поняття: арифметичні операції, логічні операції, логічна функція, мінімізація, мінтерм, макстерм, структурна формула, універсальний базис.

План самостійного опрацювання теми.

1. Засвоїти методи представлення чисел у двійковій системі счислення.
2. Засвоїти арифметичні та логічні операції булевої алгебри.
3. Засвоїти форми представлення логічних функцій.
4. Засвоїти основи синтезу цифрових пристроїв

Методичні вказівки до вивчення питань та виконання завдань.

Цифрові пристрої виконують арифметичні і логічні операції, при цьому використовується два класи змінних: числа і логічні змінні. Числа несуть інформацію про кількісні характеристики системи; над ними виробляються арифметичні дії. Логічні змінні визначають стан системи або приналежність її до певного класу станів (комутація каналів, управління роботою ЕОМ за програмою і т. п.).

Для формального опису логічної сторони процесів в цифрових пристроях використовується алгебра логіки. Алгебра логіки має справу з логічними змінними, які можуть набувати лише два значення. У цифрових пристроях обробки інформації використовується двійкова система счислення з основою 2 (A_2), в якій використовується два елементи позначення: 0 і 1.

$$A_{10} = \sum_{i=0}^{n-1} a_i \cdot (10^i) = A_2 = \sum_{j=0}^{n-1} a_j \cdot (2^j); a_i = 0, 1, 2, \dots, 9; a_j = 0, 1. \quad (1.1)$$

Вага розрядів справа наліво від молодших розрядів до старших збільшується в 2 рази, тобто має таку послідовність: 8421. У загальному вигляді ця послідовність має вигляд:

$$\dots 2^5 2^4 2^3 2^2 2^1 2^0 2^{-1} 2^{-2} 2^{-3} \dots$$

$$(\dots 32 \ 16 \ 8 \ 4 \ 2 \ 1 \ 0,5 \ 0,25 \ 0,125 \dots)$$

Для представлення багаторозрядних чисел використовують різноманітні коди (таблиця 1.1).

Таблиця 1.1 – Коди чисел від 0 до 15

Десяткове число	Коди				
	Двійковий	Восьмирічний	Шістнадцяти- річний	Двійково- десятковий	Код Грея
0	0000	0	0	0000	0000
1	0001	1	1	0001	0001
2	0010	2	2	0010	0011
3	0011	3	3	0011	0010
4	0100	4	4	0100	0110
5	0101	5	5	0101	0111
6	0110	6	6	0110	0101
7	0111	7	7	0111	0100
8	1000	10	8	1000	1100
9	1001	11	9	1001	1101
10	1010	12	A	00010000	1111
11	1011	13	B	00010001	1110
12	1100	14	C	00010010	1010
13	1101	15	D	00010011	1011
14	1110	16	E	00010100	1001
15	1111	17	F	00010101	1000

Цілочисельне складання двох чисел A і B виконується в двійковій системі точно так, як і в десятковій системі – по розрядах. Як і там, в кожному розряді мають бути підсумовані обидві двійкової цифри a_n і b_n і перенесення з попереднього розряду P_{n-1} . При складанні виникають нова сума S_n і нове перенесення P_n (таблиця 1.2). Для того, щоб мати малі витрати на апаратурну частину (hardware) комп'ютера, були зроблені зусилля за відомістю до одного алгоритму віднімання і складання.

Таблиця 1.2 – Складання в двійковій системі числення

A_n	B_n	P_{n-1}	S_n	P_n
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Цього можна добитися, якщо застосовувати двійкові цифри у їх доповнюючій формі. Розрізняють одиничне доповнення (зворотний код, порозрядне доповнення) і подвійне доповнення (точне доповнення).

Таким чином, цілочисельне віднімання двох чисел A і B виконується у двійковій системі точно так, як і в десятковій системі – по розрядам. При цьому до від'ємного додається від'ємник, представлений у точному додатковому коді.

Логічні вирази є функціями логічних (перемикаючих) 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 $k = 1$: $A = 0$ і $A = 1$; при $k = 2$: $AB = 00, 01, 10, 11$ і так далі. Для кожного набору змінних логічна функція F може набувати значення 0 або 1. Тому для k змінних можна утворити $l_k = 2^k$ різних логічних функцій. Таким чином, при $k = 2$ можна отримати $l_2 = 16$ функцій і далі при збільшенні k число l_k зростає надзвичайно швидко: $l_3 = 256$, $l_4 = 65536$ і так далі.

Функціональні залежності між логічними змінними можуть бути описані логічними формулами або таблицями істинності. У загальному вигляді логічна формула функції двох змінних записується у вигляді $F = f(A, B)$; де A, B вхідні змінні.

Основні операції булевої алгебри можливо представити алгебраїчно, таблично та схемно.

Інверсія (інші назви: заперечення, доповнення). Функція НІ - це функція одного аргументу. Вона дорівнює 1, коли її аргумент дорівнює 0, і навпаки.

Рівняння функції: $F = \bar{A}$.

Схему, яка забезпечує виконання такої функції, називають інвертором або схемою НІ (рисунок 1.1).

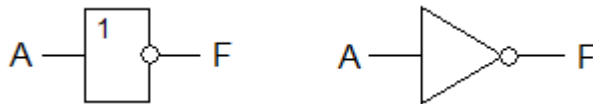


Рисунок 1.1 - Позначення схеми інвертора

Таблиця істинності інвертора

A	$F = \bar{A}$
0	1
1	0

Функція І (інші назви: кон'юнкція, логічне множення, AND) - це функція двох або більшого числа аргументів.

Рівняння функції: $F = AB$; $F = A \cdot B$; $F = A \& B$; $F = A \wedge B$

На рисунку 1.4 показані умовні зображення кон'юнктора.

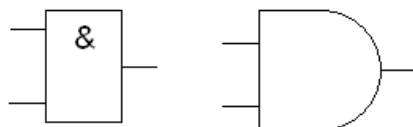


Рисунок 1.4 - Кон'юнктор

Таблиця істинності кон'юктора

AB	$F = A \cdot B$
00	0
01	0
10	0
11	1

Функція АБО (інші назви: диз'юнкція, логічне складання, OR) - це функція двох або більшого числа аргументів.

Рівняння функції:

$$F = A + B; \quad F = A \vee B$$

Таблиця істинності диз'юктора

AB	$F = A + B$
00	0
01	1
10	1
11	1

На рисунку 1.5 показані умовні зображення диз'юктора.

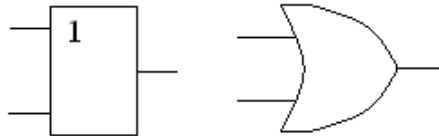


Рисунок 1.5 - Диз'юнктор

Функція І-НІ (NAND) здійснює інвертування отриманого результату операції І.

Рівняння функції:

$$F = \overline{AB}; \quad F = \overline{A \cdot B}; \quad F = \overline{A \& B}; \quad F = \overline{A \wedge B}$$

Таблиця істинності елементу І-НІ

AB	$F = \overline{A \cdot B}$
00	1
01	1
10	1
11	0

На рисунку 1.6 показані умовні зображення елементу І-НІ.

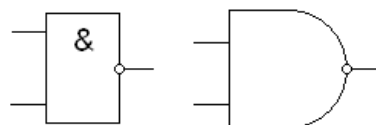


Рисунок 1.6 – Елемент І-НІ

Функція АБО-НІ (NOR) здійснює інвертування отриманого результату операції АБО.

Рівняння функції:

$$F = \overline{A + B}; F = \overline{A \vee B}$$

Читається «F є інверсія A або B».

Функція АБО-НІ дорівнює 1 тоді і лише тоді, коли всі її аргументи дорівнюють 0.

Таблиця істинності елементу АБО-НІ

AB	$F = \overline{A + B}$
00	1
01	0
10	0
11	0

На рисунку 1.7 показані умовні зображення елементу АБО-НІ.

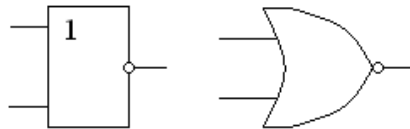


Рисунок 1.7 – Елемент АБО-НІ

Функція «Виключаюче АБО» (XOR). Функція «Виключаюче АБО» набуває значення 0, якщо змінні в наборі рівні між собою і набуває значення 1, якщо змінні в наборі різні між собою.

Рівняння функції:

$$F = A \oplus B; F = \overline{A}B + A\overline{B}$$

Таблиця істинності функції «Виключаюче АБО»

AB	$F = A \oplus B$
00	0
01	1
10	1
11	0

На рисунку 1.8 показані умовні зображення елементу «Виключаюче АБО».

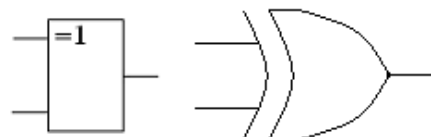


Рисунок 1.8 – Елемент «Виключаюче АБО»

Функція «Виключаюче АБО» - НІ (XNOR). Функція еквівалентність дорівнює 1, якщо змінні рівні між собою, функція дорівнює 0, якщо змінні різні між собою.

Рівняння функції:

$$F = \overline{A \oplus B}; F = AB + \overline{AB}$$

Таблиця істинності функції «Виключаюче АБО» - НІ

AB	$F = \overline{A \oplus B}$
00	1
01	0
10	0
11	1

На рисунку 1.9 показані умовні зображення елемента «Виключаюче АБО».

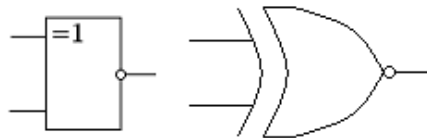


Рисунок 1.9 – Елемент «Виключаюче АБО» - НІ

Правила старшинства логічних операцій.

Заперечення – логічна дія першого рівня.

Кон'юнкція – логічна дія другого рівня.

Диз'юнкція – логічна дія третього рівня.

Якщо в логічному вираженні зустрічаються дії різних рівнів, то спочатку виконуються дії першого рівня, потім другого і лише після цього третьому рівню. Всяке відхилення від цього порядку має бути позначене дужками.

Логічні функції можуть мати різні форми представлення: словесну, табличну, алгебраїчну, графічну. Функцію задану у вигляді словесного опису можна представити у вигляді таблиці істинності. Таблиця істинності містить усі можливі набори значень логічних змінних і значення функції, відповідні кожному з наборів.

Аби здійснити перехід від табличного представлення до алгебраїчного, кожному набору змінних ставиться у відповідність мінтерм (конституента одиниці) - кон'юнкція всіх змінних, які входять в прямому вигляді, якщо значення даної змінної в наборі дорівнює 1, або в інверсному вигляді, якщо значення змінної дорівнює 0. Для k змінних складаються $q = 2^k$ мінтермів: m_0, m_1, \dots, m_{q-1} . У загальному випадку алгебраїчне вираження будь-якої логічної функції можна представити в наступній формі:

$$F = \sum_{i=0}^{q-1} f_i m_i = f_0 m_0 + f_1 m_1 + f_2 m_2 + f_3 m_3,$$

де f_i, m_i - значення функції (0 або 1) і мінтерм, відповідні i -му набору змінних. Таке представлення функції називається її досконалою диз'юнктивною нормальною формою (ДДНФ).

Інша алгебраїчна форма представлення функції виходить при використанні макстермів. Макстерм (конституента 0) - диз'юнкція всіх змінних, які входять в прямому вигляді, якщо значення даної змінної дорівнює 1, або в інве-

рсному вигляді, якщо, значення змінної дорівнює 0. Число макстермів для функції k змінних дорівнює $q = 2^k$. Алгебраїчне вираження функції можна представити в наступній формі:

$$F = \prod_{i=0}^{n-1} (f_i + M_i) = (f_0 + M_0) \cdot (f_1 + M_1) \cdot (f_2 + M_2) \cdot (f_3 + M_3),$$

де f_i, M_i - значення функції і макстерм, відповідні i -му набору змінних. Таке представлення функції називається її досконалою кон'юнктивною нормальною формою (ДКНФ).

Основи синтезу цифрових пристроїв.

1. Послідовність операцій при синтезі цифрових пристроїв комбінаційного типу:

- складання таблиці істинності комбінаційного цифрового пристрою згідно його визначення, призначення, словесного опису принципу роботи;
- складання логічної формули згідно таблиці істинності;
- спрощення логічної формули;
- аналіз отриманої формули з метою побудови різних варіантів і знаходження найкращого з них по тих або інших критеріях;
- складання функціональної схеми комбінаційного цифрового пристрою з елементів І, АБО, НІ.

2. Аналітичний запис логічної формули комбінаційного цифрового пристрою.

Запис у формі ДДНФ.

У ДДНФ логічна формула є логічною сумою декількох логічних множень, в кожне з яких входять всі незалежні змінні з інверсією або без неї.

Формула виходить в два етапи.

а) Записується логічна сума множень, в кожне з яких входять всі незалежні змінні. Кількість доданків дорівнює числу наборів таблиці істинності, на яких логічна функція рівна «1».

б) Ставиться знак інверсії над тими незалежними змінними, які рівні «0» в даному наборі.

Запис у формі ДКНФ.

У ДКНФ формула є логічним множенням декількох логічних сум, в кожну з яких входять всі незалежні змінні з інверсією або без неї.

Формула виходить в два етапи.

а) Записується логічне множення всіх співмножників. Кількість співмножників дорівнює числу наборів таблиці істинності, на яких логічна функція рівна «0».

б) ставиться знак інверсії над тими незалежними змінними, які рівні «1» в даному наборі.

Структурні формули у вигляді ДДНФ і ДКНФ еквівалентні і за допомогою законів алгебри логіки можуть бути перетворені одна в іншу.

Однозначна залежність складності логічної формули і функціональної схеми логічного пристрою наводять до виводу про необхідність мінімізації структурної формули логічного пристрою. Мінімізація здійснюється з використанням основних співвідношень, законів і теорем алгебри логіки. При цьому

отримуються мінімальні диз'юнктивні (МДНФ) або кон'юнктивні (МКНФ) нормальні форми.

Суть розрахункового методу мінімізації полягає в послідовному застосуванні до деякої формули законів і правил тотожних перетворень алгебри логіки. При цьому широко використовують наступні прийоми: збільшення одного або декількох членів, що входять в ДДНФ, оскільки відповідно до закону ідемпотентності: $A + A + A + A = A$; виділення членів, що містять множник $A + \bar{A} = 1$ (закон виключення третього); використання правила склеювання та ін. Алгебраїчна формула, що виходить в результаті мінімізації, називається тупиковою. Функція може мати декілька тупикових форм.

При відносно невеликому числі змінних ($k \leq 6$) зручним є графічне представлення логічних функцій у вигляді карт мінтермів. Найбільш розповсюдженою їх формою є карти Карно. На рисунку 1.10 показані карти Карно для функцій $k = 2, 3, 4, 5$ змінних.

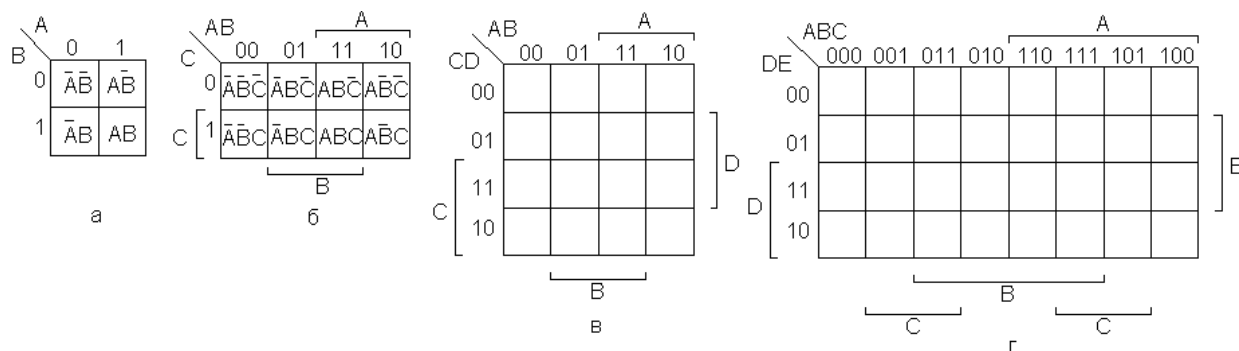


Рисунок 1.10 – Карты Карно для функций двух (а), трёх (б), четырёх (в) и пяти (г) переменных

Карта Карно містить $q = 2^k$ кліток, причому кожній клітці відповідає один з q мінтермів. Якщо потрібно представити на карті Карно логічну функцію, задану у вигляді ДДНФ, то в клітках карти, відповідних мінтермів, що входять в ДДНФ, ставляться 1. Останні клітки залишаються незаповненими або заповнюються 0. Процес мінімізації полягає у формуванні правильних прямокутників, що містять по 2^k вічок, де k – ціле число. У прямокутники об'єднуються сусідні вічка, які відповідають сусіднім елементарним множинам (тобто відрізняються лише в одному розряді).

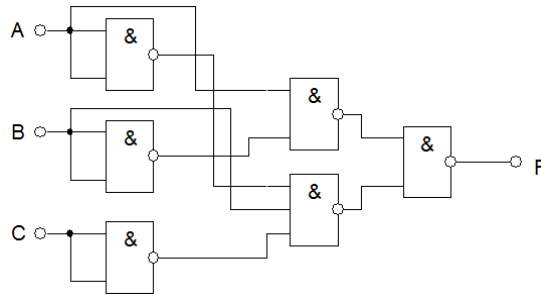
Правила групування мінтермів.

1. Грукуються дві клітки, які стоять поряд, в стовпці, або ряду.
2. Грукуються клітки, які є повними квадратами з 4, 16 кліток.
3. Грукуються клітки, які є повними горизонтальними рядами, або вертикальними стовпцями.
4. Грукуються клітки, які представляють два поруч розташованих стовпця, або рядка.
5. Клітка може входити в декілька об'єднань.
6. Грукуються клітки розташовані симетрично відносно вертикальної, або горизонтальної осі карти Карно для п'яти або шести змінних.

Для спрощення схемотехніки цифрових пристроїв використовують запис структурних формул в універсальних базисах.

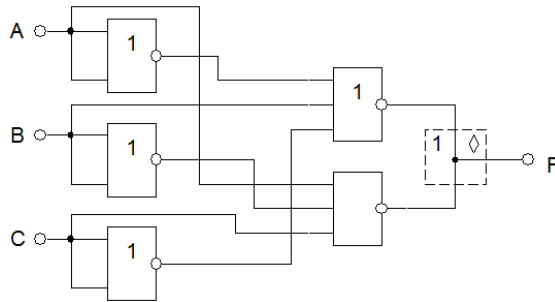
При реалізації на елементах **I-НІ** необхідно виконати подвійну інверсію отриманої МДНФ функції і перетворити по теоремі Де Моргана інверсію диз'юнкції в кон'юнкцію інверсій.

Приклад: $F = \overline{A}\overline{B} + \overline{A}\overline{B}\overline{C} = \overline{\overline{\overline{\overline{A}\overline{B} + \overline{A}\overline{B}\overline{C}}}} = \overline{\overline{\overline{A}\overline{B}} \cdot \overline{\overline{A}\overline{B}\overline{C}}}$



При реалізації на елементах **АБО-НІ** – «Монтажне АБО» необхідно виконати подвійну інверсію кожного кон'юнктивного члена отриманої МДНФ функції і перетворити по теоремі Де Моргана інверсії кон'юнкцій в диз'юнкції інверсій.

Приклад: $F = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}\overline{C} = \overline{\overline{\overline{\overline{A}\overline{B}\overline{C}}}} + \overline{\overline{\overline{\overline{A}\overline{B}\overline{C}}}} = \overline{\overline{A+B+C}} + \overline{\overline{A+B+C}}$



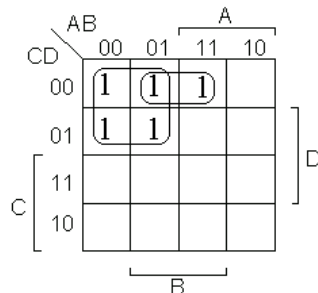
При реалізації на елементах **НІ** – «Монтажне І» необхідно виконати мінімізацію ДКНФ функції, призвести подвійну інверсію кожного диз'юнктивного члена отриманої МКНФ функції і перетворити по теоремі Де Моргана інверсії диз'юнкцій в кон'юнкції інверсій.

Приклад:

$$F = \overline{A+B+C} \overline{A+B+C+D} \overline{B+C+D}$$

$$\overline{F} = \overline{\overline{A+B+C} \overline{A+B+C+D} \overline{B+C+D}} = \overline{\overline{A+B+C}} \overline{\overline{A+B+C+D}} \overline{\overline{B+C+D}} = \overline{A+B+C} \overline{A+B+C+D} \overline{B+C+D}$$

$$= \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D + B \cdot \overline{C} \cdot \overline{D}$$

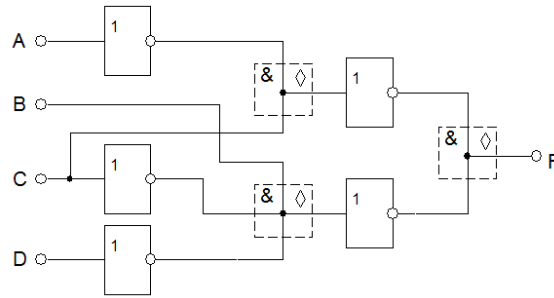


Після мінімізації: $\overline{F} = \overline{A} \cdot C + B \cdot \overline{C} \cdot \overline{D}$.

$$F = \overline{\overline{A \cdot C + B \cdot \overline{C} \cdot \overline{D}}} = \overline{\overline{A \cdot C} \cdot \overline{B \cdot \overline{C} \cdot \overline{D}}} = \overline{\overline{A + \overline{C}} \cdot \overline{B + C + D}}.$$

Приведення до базису НІ – «Монтажне І»:

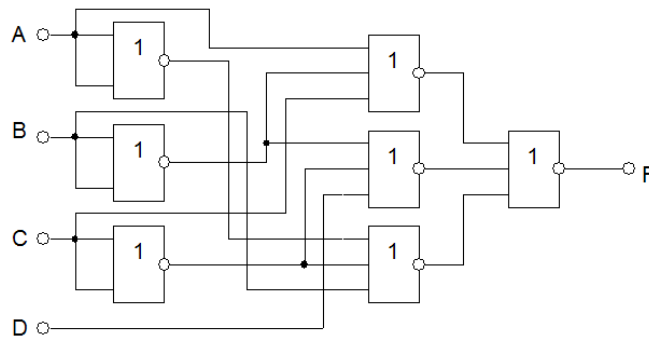
$$F = \overline{\overline{A + \overline{C}} \cdot \overline{B + C + D}} = \overline{\overline{A + \overline{C}}} \cdot \overline{\overline{B + C + D}} = \overline{\overline{A \cdot C}} \cdot \overline{\overline{B \cdot \overline{C} \cdot \overline{D}}}$$



При реалізації на елементах **АБО-НІ** необхідно виконати подвійну інверсію отриманої МКНФ функції і перетворити по теоремі Де Моргана інверсію кон'юнкції в диз'юнкцію інверсій.

Приклад:

$$F = (A + \overline{B} + C)(\overline{B} + \overline{C} + D)(\overline{A} + B + \overline{C}) = \overline{\overline{(A + \overline{B} + C)(\overline{B} + \overline{C} + D)(\overline{A} + B + \overline{C})}} = \\ = \overline{(A + \overline{B} + C) + (\overline{B} + \overline{C} + D) + (\overline{A} + B + \overline{C})}$$



Питання для закріплення вивченого матеріалу та самоконтролю.

1. Арифметичні операції цифрової схемотехніки.
2. Логічні операції цифрової схемотехніки.
3. Форми представлення логічних функцій.
4. Мінімізація логічних функцій.
5. Представлення структурних формул в універсальних базисах.

Практичні завдання.

1. Мінімізувати функцію і розробити структурну логічну схему на елементах І-НІ:

$$F = \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + B \cdot \overline{C} \cdot D + A \cdot \overline{B} \cdot \overline{C} \cdot D + \overline{A} \cdot B \cdot \overline{D} + A \cdot \overline{B} \cdot C \cdot D + \overline{A} \cdot \overline{B} \cdot D$$

2. Мінімізувати функцію і розробити структурну логічну схему на елементах АБО-НІ – «Монтажне АБО»:

$$F = A \cdot \overline{B} \cdot \overline{C} \cdot D + A \cdot B \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot \overline{C} \cdot D$$

3. Мінімізувати функцію і розробити структурну логічну схему на елементах НІ – «Монтажне І»

$$F = (A + \overline{B} + C)(A + B + C + D)(\overline{B} + C + \overline{D})(A + \overline{B} + \overline{C} + D)(\overline{A} + B + D)(A + B + \overline{C} + \overline{D})(\overline{A} + \overline{B} + \overline{D})$$

4. Мінімізувати функцію і розробити структурну логічну схему на елементах АБО-НІ:

$$F = (A + \bar{B} + C)(A + B + \bar{C} + D)(\bar{B} + C + \bar{D})(\bar{A} + \bar{B} + \bar{C} + D)(\bar{A} + B + D)(A + B + \bar{C} + \bar{D})$$

5. Мінімізувати функцію і розробити структурну логічну схему на елементах АБО-НІ – «Монтажне АБО»:

$$F = \bar{A} \cdot \bar{B} \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} \cdot D + B \cdot \bar{C} \cdot D + \bar{A} \cdot \bar{B} \cdot C \cdot D + A \cdot \bar{B} \cdot \bar{D} + A \cdot \bar{B} \cdot C \cdot D + \bar{A} \cdot \bar{B} \cdot \bar{D}$$

6. Мінімізувати функцію і розробити структурну логічну схему на елементах І-НІ:

$$F = A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} \cdot D + \bar{A} \cdot B \cdot C + \bar{A} \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot \bar{C}$$

2 Цифрові пристрої комбінаційного типу

Метою вивчення теми є засвоєння принципів побудови пристроїв комбінаційного типу.

Ключові терміни та поняття: суматор, дешифратор, мультиплексор, демультимплексор, шифратор, перетворювачі кодів.

План самостійного опрацювання теми.

1. Засвоїти принципи побудови цифрових суматорів.
2. Засвоїти арифметичні та логічні операції булевої алгебри.
3. Засвоїти форми представлення логічних функцій.
4. Засвоїти основи синтезу цифрових пристроїв

Методичні вказівки до вивчення питань та виконання завдань.

До комбінаційного типу відносяться пристрої вихід яких у момент часу однозначно визначається станом входу в той же момент часу $y^{(t)} = f(x^{(t)})$.

Принцип побудови інтегрального суматора.

Суматори призначені для арифметичного складання двох чисел. З принципу складання багаторозрядних двійкових чисел виходить, що в кожному n-розряді знаходиться сума S_n трьох чисел по модулю два: доданків A_n , B_n і перенесення що поступило з молодшого розряду P_{n-1} і формується сигнал перенесення в старший розряд P_n . У цифровій обчислювальній техніці використовуються однорозрядні схеми, які підсумовують, з двома і трьома входами, причому перші називаються напівсуматорами а другі – повними однорозрядними суматорами. Напівсуматори можуть використовуватися лише для підсумовування молодших розрядів чисел. Повні однорозрядні суматори мають додатковий третій вхід, на який подається перенесення з попереднього розряду при підсумовуванні багаторозрядних чисел. У таблиці 2.1 приведено алгоритм функціонування напівсуматора, на підставі якого складена його структурна формула у вигляді ДДНФ.

Таблиця 2.1 – Таблиця істинності напівсуматора

A	B	S	P_{i+1}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = \overline{A}B + A\overline{B}; P_{i+1} = AB.$$

Отримаємо однорозрядний напівсуматор (рисунк 2.1).

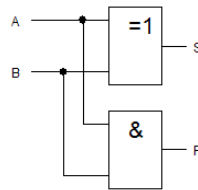


Рисунок 2.1 – Схема напівсуматора на елементі “Виключаюче АБО”

Схему повного однорозрядного суматора можна отримати на основі двох схем напівсуматорів і схеми «АБО», як показано на рисунку 2.2.

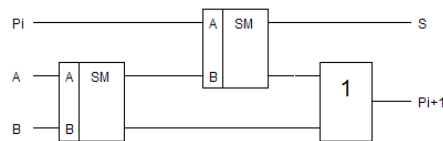


Рисунок 2.2 – Схема повного однорозрядного суматора

Принципи побудови інтегрального дешифратора.

Цей комбінаційний пристрій дозволяє визначити код вхідного числа по одиночному сигналу на одному з виходів.

Лінійна структура дешифратора.

З таблиці станів (таблиця 2.3) виходить, що на кожному вході дешифратора повинен стояти інвертор.

Таблиця 2.3 – Таблиця станів лінійного дешифратора

Входи		Виходи			
A	B	Y0	Y1	Y2	Y3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Рівняння функціонування дешифратора по кожному виходу визначається певною комбінацією вхідних сигналів.

Наприклад: на Y3 є логічна 1, якщо на вході комбінація AB.

Відповідно до рівнянь функціонування будується схема лінійного дешифратора (рисунк 2.3).

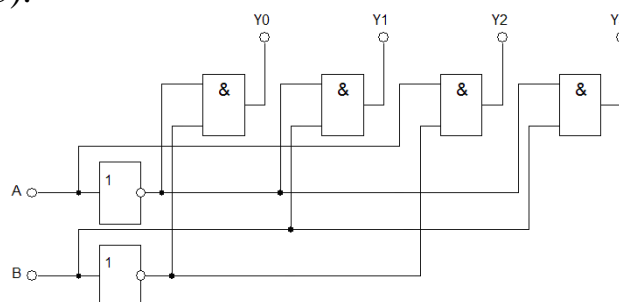
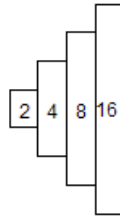


Рисунок 2.3 – Схема лінійного дешифратора

Пірамідальна структура дешифратора.

Число рівнів пірамідального дешифратора $n-1$, де n число входів. Число виходів рівня 2^N , де N номер рівня.



Пірамідальна структура дешифратора (рисунок 2.4) компактніше лінійної.

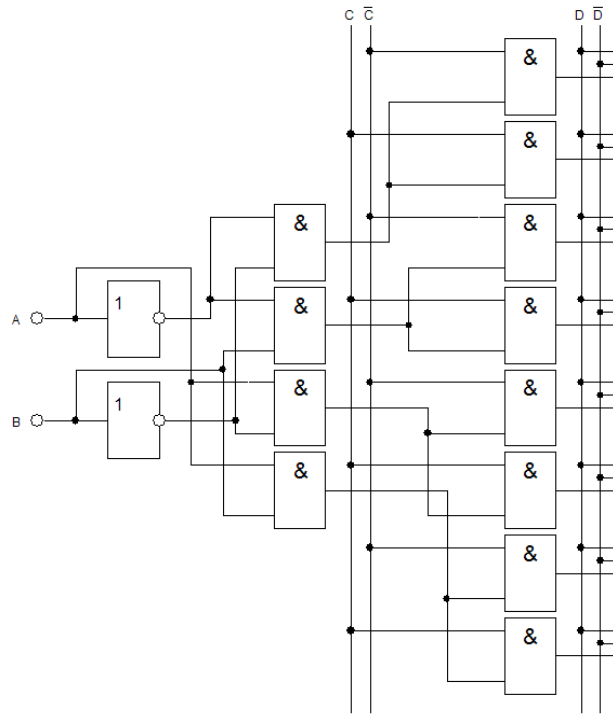


Рисунок 2.4 – Пірамідальна структура дешифратора

Принцип побудови інтегрального мультиплексора

Мультиплексор – функціональний вузол, який має n - адресних входів, $N=2^n$ інформаційних входів, один вихід і здійснює керовану комутацію інформації, що поступає по N вхідним лініям, на одну вихідну лінію. Комутація певної вхідної лінії відбувається відповідно до двійкового адресного коду (таблиця 2.4).

Рівняння функціонування мультиплексора:

$$Q = \bar{A}\bar{B}X_0 + \bar{A}BX_1 + A\bar{B}X_2 + ABX_3.$$

Схема мультиплексора 4 в 1 представлена на рисунку 2.5.

Таблиця 2.4 – Алгоритм функціонування мультиплексора

Адрес		X0	X1	X2	X3	Q
A	B					
0	0	1	0	0	0	X0
0	1	0	1	0	0	X1
1	0	0	0	1	0	X2
1	1	0	0	0	1	X3

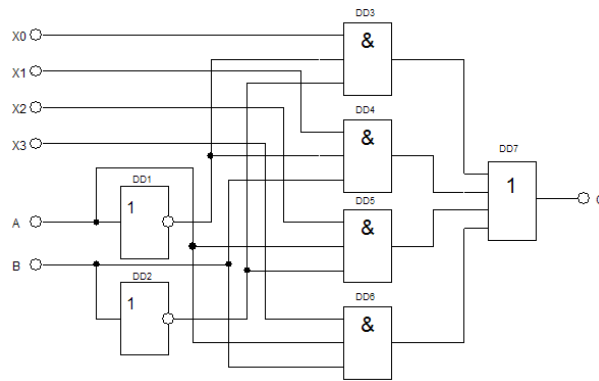


Рисунок 2.5 – Схема мультиплексора 4 в 1

Принцип побудови інтегрального демультимплексора.

Демультимплексор – це функціональний вузол, що здійснює керовану комутацію інформації, яка поступає по одному входу, на N виходів. Таким чином, демультимплексор реалізує операцію, протилежну тій, яку виконує мультиплексор. У загальному випадку число вихідних ліній N визначається кількістю адресних входів n і рівне $N=2^n$. Для випадку $n = 2$ функціонування демультимплексора здійснюється відповідно до таблиці істинності (таблиця 2.5).

Таблиця 2.5 – Таблиця істинності демультимплексора 1 в 4

Адрес		Виходи			
A	B	Y0	Y1	Y2	Y3
0	0	X	0	0	0
0	1	0	X	0	0
1	0	0	0	X	0
1	1	0	0	0	X

З таблиці істинності записуємо характеристичні рівняння демультимплексора:

$$Y0 = X\bar{A}\bar{B}; \quad Y1 = X\bar{A}B; \quad Y2 = XA\bar{B}; \quad Y3 = XAB.$$

Відповідна цим рівнянням функціональна схема демультимплексора приведена на рисунку 2.6. Вона має в своєму складі два інвертори і чотири елементи «І».

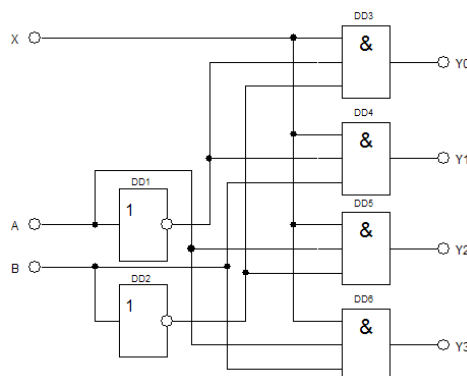


Рисунок 2.6 – Функціональна схема демультимплексора

Принцип побудови інтегрального шифратора

Шифратор (кодер) – це функціональний вузол, призначений для перетворення управляючих сигналів (команд), що поступають на його входи, в n , – розрядний двійковий код. Зокрема, такими сигналами або командами можуть бути десяткові числа, наприклад, номер команди, який за допомогою шифратора перетвориться в двійковий код.

Як приклад розробимо схему трьохрозрядного шифратора. Спочатку слід побудувати таблицю код (таблицю істинності), в якій код номера сигналу представимо двійковим кодом (таблиця 2.6).

Рівняння функціонування шифратора:

$$F1 = X1 + X3 + X5 + X7;$$

$$F2 = X2 + X3 + X6 + X7;$$

$$F3 = X4 + X5 + X6 + X7.$$

Таблиця 2.6 – Таблиця істинності трьохрозрядного шифратора

X_i	F3	F2	F1
X0	0	0	0
X1	0	0	1
X2	0	1	0
X3	0	1	1
X4	1	0	0
X5	1	0	1
X6	1	1	0
X7	1	1	1

У загальному випадку, при використанні двійкової коди, можна закодувати 2^n вхідних сигналів. У розглянутій вище схемі вихідний код «000» присутній на виході при подачі сигналу на вхід X0 і в разі, якщо вхідний сигнал взагалі не подається ні на один з входів. Для однозначної ідентифікації сигналу X0 в інтегральних схемах формується ще один вихідний сигнал – ознака подачі вхідного сигналу, який використовується і для інших цілей.

Схема, реалізована на елементах АБО, приведена на рисунку 2.7.

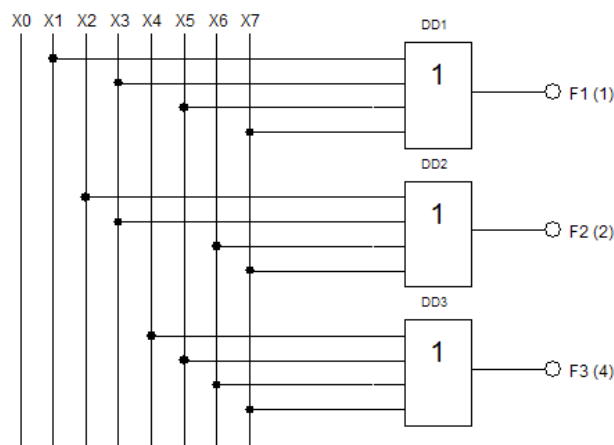


Рисунок 2.7 – Функціональна схема трьохрозрядного шифратора

Принцип побудови перетворювачів кодів.

Операція зміни коду числа називається його перекодуванням. Інтегральні мікросхеми, що виконують ці операції, звуться перетворювачами кодів. Перетворювачі кодів бувають прості і складні. До простих відносяться перетворювачі, які виконують стандартні операції зміни кодів чисел. Складні перетворювачі кодів виконують нестандартні перетворення кодів і їх схеми доводиться розробляти кожен раз за допомогою алгебри логіки. По своїй структурі перетворювачі кодів є дешифраторами, лише вони перетворюють двійковий код в сигнали не лише на одному, але і на декількох виходах. Вважатимемо, що перетворювачі кодів мають n входів і k виходів. Співвідношення між n и k можуть бути будь-якими: $n = k$, $n < k$ і $n > k$. При перетворенні кодів чисел з ними можуть виконуватися різні додаткові операції, наприклад, множення на вагові коефіцієнти.

Перетворювачі код (ПК) можуть бути ваговими і неваговими. Вагові ПК перетворюють інформацію з однієї системи числення в іншу. Основне призначення невагових - перетворення інформації для її подальшого відображення.

Як приклад розглянемо перетворювач двійкового коду в зворотній код.

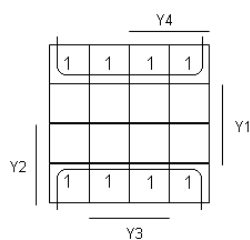
Для перетворення двійкового коду в зворотній код необхідно побудувати таблицю функціонування перетворювача (таблиця 2.7).

Таблиця 2.7 - Таблиця функціонування перетворювача двійкового коду в зворотній код

Двійковий код				Зворотній код			
Y4	Y3	Y2	Y1	A4	A3	A2	A1
0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	0
0	0	1	0	1	1	0	1
0	0	1	1	1	1	0	0
0	1	0	0	1	0	1	1
0	1	0	1	1	0	1	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	0	0
1	0	0	0	0	1	1	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	0	1
1	0	1	1	0	1	0	0
1	1	0	0	0	0	1	1
1	1	0	1	0	0	1	0
1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	0

Складемо рівняння функціонування пристрою і спростимо їх за допомогою карт Карно.

$$A1 = \overline{Y4}\overline{Y3}\overline{Y2}\overline{Y1} + \overline{Y4}\overline{Y3}Y2\overline{Y1} + \overline{Y4}Y3\overline{Y2}\overline{Y1} + \overline{Y4}Y3Y2\overline{Y1} + Y4\overline{Y3}\overline{Y2}\overline{Y1} + Y4\overline{Y3}\overline{Y2}Y1 + Y4\overline{Y3}Y2\overline{Y1} + Y4\overline{Y3}Y2Y1$$

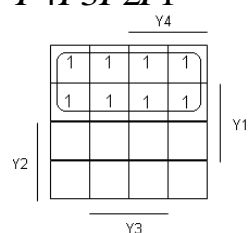


Після спрощення отримуємо:

$$A1 = \overline{Y1}$$

$$A2 = \overline{Y4}\overline{Y3}\overline{Y2}\overline{Y1} + \overline{Y4}\overline{Y3}\overline{Y2}Y1 + \overline{Y4}\overline{Y3}Y2\overline{Y1} + \overline{Y4}\overline{Y3}Y2Y1 + Y4\overline{Y3}\overline{Y2}\overline{Y1} +$$

$$+ Y4\overline{Y3}\overline{Y2}Y1 + Y4\overline{Y3}Y2\overline{Y1} + Y4\overline{Y3}Y2Y1$$

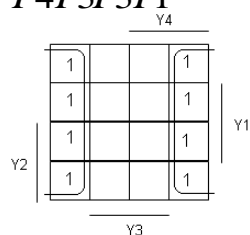


Після спрощення отримуємо:

$$A2 = \overline{Y2}$$

$$A3 = \overline{Y4}\overline{Y3}\overline{Y2}\overline{Y1} + \overline{Y4}\overline{Y3}\overline{Y2}Y1 + \overline{Y4}\overline{Y3}Y2\overline{Y1} + \overline{Y4}\overline{Y3}Y2Y1 + Y4\overline{Y3}\overline{Y2}\overline{Y1} +$$

$$+ Y4\overline{Y3}\overline{Y2}Y1 + Y4\overline{Y3}Y2\overline{Y1} + Y4\overline{Y3}Y2Y1$$

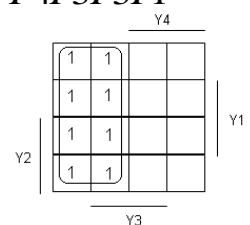


Після спрощення отримуємо:

$$A3 = \overline{Y3}$$

$$A3 = \overline{Y4}\overline{Y3}\overline{Y2}\overline{Y1} + \overline{Y4}\overline{Y3}\overline{Y2}Y1 + \overline{Y4}\overline{Y3}Y2\overline{Y1} + \overline{Y4}\overline{Y3}Y2Y1 + Y4\overline{Y3}\overline{Y2}\overline{Y1} +$$

$$+ Y4\overline{Y3}\overline{Y2}Y1 + Y4\overline{Y3}Y2\overline{Y1} + Y4\overline{Y3}Y2Y1$$



Після спрощення отримуємо:

$$A4 = \overline{Y4}$$

Згідно з отриманими рівняннями функціональна схема перетворювача двійкового коду в зворотний код представлена на рисунку 2.8.

Схема електрична принципова перетворювача двійкового коду в зворотний код зібрана на мікросхемі K155ЛН1 (рисунок 2.9).

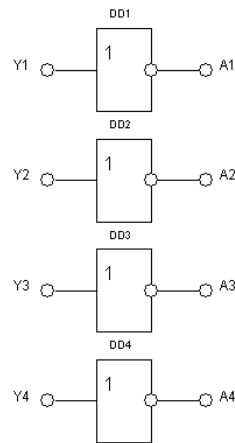


Рисунок 2.8 - Функціональна схема перетворювача двійкового коду в зворотній код

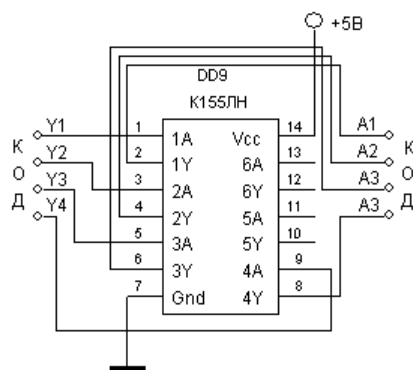


Рисунок 2.9 - Схема електрична принципова перетворювача двійкового коду в зворотній код

Питання для закріплення вивченого матеріалу та самоконтролю.

1. Принцип побудови лінійного дешифратора.
2. Принцип побудови пірамідального дешифратора.
3. Принцип побудови матричного дешифратора.
4. Принцип побудови демультиплексора.
5. Принцип побудови мультиплексора.
6. Принцип побудови шифратора.
7. Принцип побудови перетворювача коду.

Практичні завдання

1. Розробити логічну схему чотирьох розрядного суматора. Представити діаграму функціонування.
2. Розробити схему зведення в квадрат десяткових чисел від 0 до 9, з представленням інформації у двійковому коді.
3. Розробити логічну структуру пірамідального дешифратора з 4 в 16. Представити діаграму функціонування.
4. Розробити логічну структуру лінійного дешифратора з 4 в 16. Представити діаграму функціонування.
5. Розробити перетворювач чотирьох розрядного двійкового коду чисел в додатковий.

6. Розробити логічну схему мультіплексора «16 в 1». Представити діаграму функціонування.

7. Розробити схему перетворювача десяткового коду чисел від 0 до 15 в двійково-десятковий.

8. Розробити логічну схему демультіплексора «з 1 в 16». Представити діаграму функціонування.

9. Розробити структурну схему перетворювача чотирьохрозрядного двійкового коду чисел в код Грея.

10. Розробити структурну схему чотирьохрозрядного суматора – віднімача. Представити діаграму функціонування.

11. Розробити схему зведення в квадрат чотирьохрозрядного двійкового коду чисел 0 ... 9 з виведенням інформації на семисегментний індикатор десятикових цифр.

12. Розробити схему перетворювача чотирьохрозрядного коду 8-4-2-1 в двійково – десятковий.

13. Розробити логічну схему дешифратора двійкового коду в семисегментний код індикатора. Представити діаграму функціонування.

14. Розробити схему порівняння двохрозрядних двійкових чисел X_2X_1 і Y_2Y_1 для випадку $X_2X_1 > Y_2Y_1$. Представити діаграму функціонування.

15. Розробити схему порівняння двохрозрядних двійкових чисел X_2X_1 і Y_2Y_1 для випадку $X_2X_1 < Y_2Y_1$. Представити діаграму функціонування.

3 Цифрові пристрої послідовнісного типу

Метою вивчення теми є засвоєння принципів побудови пристроїв послідовнісного типу.

Ключові терміни та поняття: тригер, регістр, лічильник, генератор числових послідовностей, діаграма.

План самостійного опрацювання теми.

1. Засвоїти принципи побудови інтегральних тригерів.

2. Засвоїти принципи побудови інтегральних регістрів.

3 Засвоїти принципи побудови інтегральних лічильників.

4. Засвоїти принципи побудови інтегральних генераторів числових послідовностей.

Методичні вказівки до вивчення питань та виконання завдань.

Цифрові пристрої послідовнісного типові або цифрові автомати з пам'яттю – це електронні цифрові пристрої, логічні значення на виходах яких визначаються не лише сукупністю логічних сигналів на входах в даний момент часу, але і станом внутрішніх елементів пам'яті за результатами його попередньої роботи. Запам'ятовування попередніх станів виконується за допомогою тригерів і регістрів пам'яті.

Тригер – це пристрій послідовнісного типу з двома стійкими станами, призначений для запису і зберігання інформації. Під дією вхідних сигналів тригер може перемикається з одного стійкого стану в інше. При цьому напруга на

його виході стрибкоподібно міняється. За наявності тактованого входу С тригер називають синхронним, а при його відсутності – асинхронним.

RS-тригери. У простому RS-тригері інформаційні сигнали подаються безпосередньо на входи S і R елементу пам'яті (рисунок 3.1).

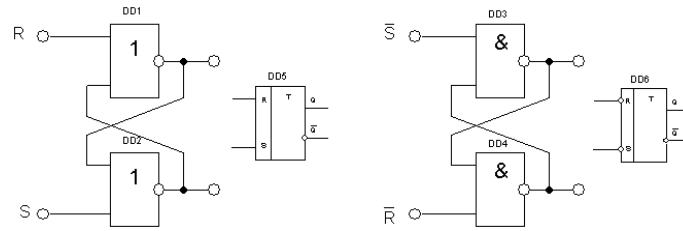


Рисунок 3.1 – RS-тригери

Повна таблиця функціонування (таблиця істинності) приведена на рисунку 3.2 (а), в якій попередній стан тригера Q^n до подачі входних сигналів є одним з входних сигналів. Вихідний стан тригера після подачі входних сигналів позначений символом Q^{n+1} . Таблиця переходів тригера приведена на рисунку 3.2 (б).

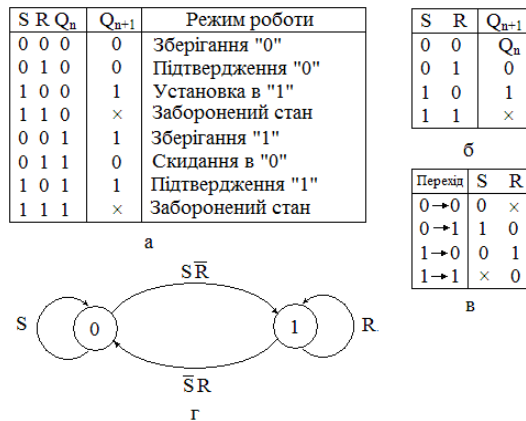


Рисунок 3.2 – Таблиця істинності (а), таблиця перемикачів (б), словник переходів (в), діаграма станів (г) RS-тригера

Рівняння функціонування RS-тригера: $Q^{n+1} = S + \bar{R}Q^n$; $\bar{Q}^{n+1} = R + \bar{S}\bar{Q}^n$.

RST-тригери. Для роботи в схемах з синхронізацією режиму розроблені синхронні RS-тригери. Синхронні RS-тригери мають три входи: S, R і C (рисунок 3.3).

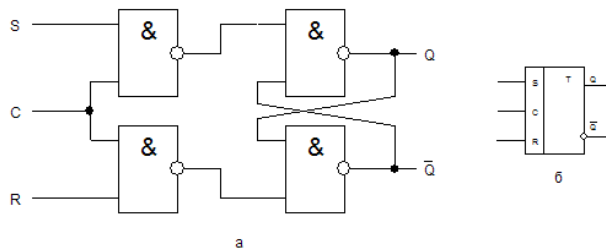


Рисунок 3.3 – Синхронний RS-тригер (а); УГП RS-тригера (б)

Рівняння функціонування RST-тригера: $Q^{n+1} = C(S + \bar{R}Q^n)$

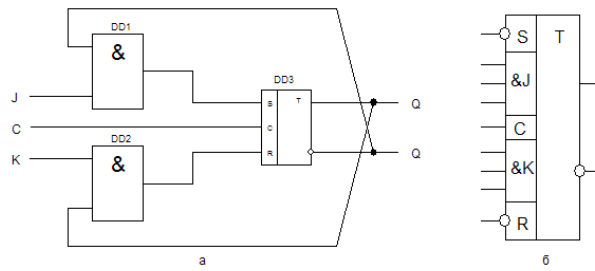


Рисунок 3.7 – JK -тригер: а) структурна схема; б) УГП JK-тригера

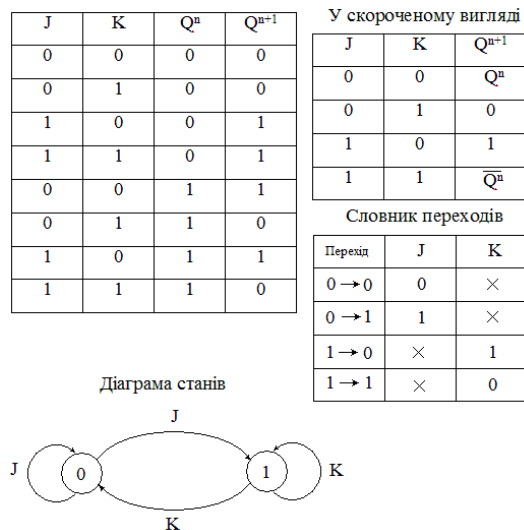


Рисунок 3.8 – Таблиця, словник і діаграма переходів JK-тригера

Регістри – це функціональні вузли на основі тригерів, призначені для прийому, короткочасного зберігання (на один або декілька циклів роботи даного пристрою), передачі і перетворення багаторозрядної цифрової інформації.

Паралельні регістри – це пристрої, призначені для запису, зберігання і видачі інформації, представленої у вигляді двійкових кодів. Для зберігання кожного двійкового розряду в регістрі використовується одна тригерна комірка. Для запам'ятовування багаторозрядних слів необхідне число тригерів об'єднують разом і розглядають як єдиний функціональний вузол-регістр (рисунок 3.9).

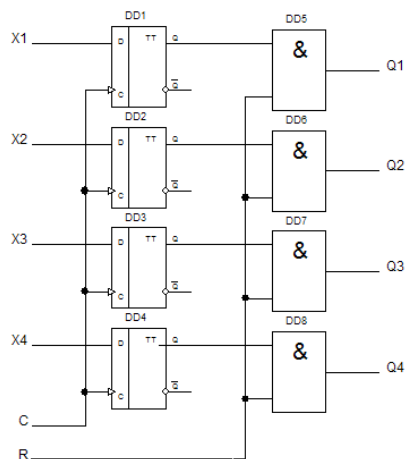


Рисунок 3.9 – Паралельний регістр на тактованих D-тригерах

Послідовний регістр будується шляхом з'єднання виходу кожного тригера з входом наступного. Для запису і прочитування одночасно на входи синхронізації всіх тригерів подаються тактові імпульси С (рисунок 3.10).

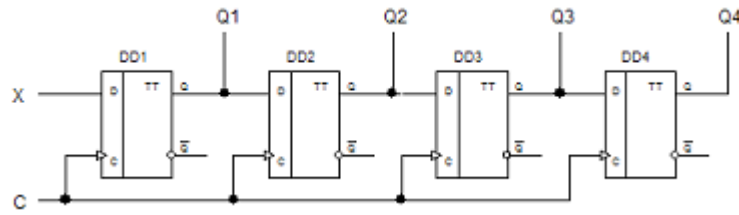


Рисунок 3.10 – Послідовний регістр на тактованих D-тригерах

Лічильником називається пристрій, призначений для підрахунку числа імпульсів, що поступають на його вхід, і фіксації цього числа у вигляді коди, що зберігається в тригерах.

Схема двійкового лічильника з послідовним перенесенням на JK-тригерах, що працює в коді 8421, показана на рисунку 3.11. Тут JK-тригери перетворені в Т-тригери шляхом подачі «1» на вхід J і K.

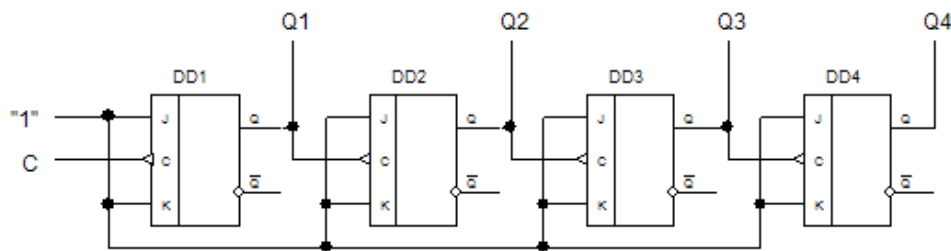


Рисунок 3.11 – Схема двійкового лічильника з послідовним перенесенням

Оскільки тригери мають інверсний динамічний вхід, то кожен подальший тригер перемикатиметься при скиданні в «0» попереднього тригера (рисунок 3.12).

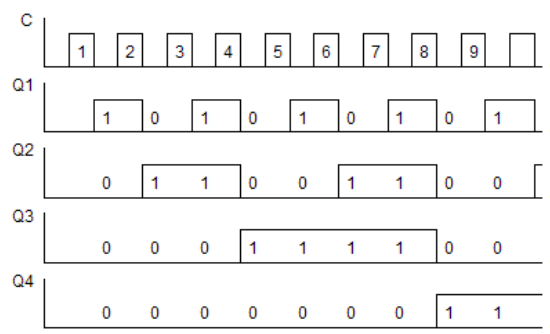


Рисунок 3.12 – Діаграма функціонування двійкового лічильника

Окрім основної функції – рахунки імпульсів, лічильник забезпечує ділення частоти дотримання імпульсів. Якщо частоту дотримання імпульсів, що поступають на вхід тригера Т1 позначити f , то на виході тригера Т1 сигнал змінюється з частотою $f/2$, на виході тригера Т2 з частотою $f/4$ і так далі.

Порядок розрахунку схеми двійкового лічильника з паралельним перенесенням.

1) Будуємо таблицю станів лічильника (таблицю 3.1).

Таблиця 3.1 - Таблиця станів чотирьох розрядного лічильника

C	Q_4^n	Q_3^n	Q_2^n	Q_1^n	Q_4^{n+1}	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}
1	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	0
1	0	0	1	0	0	0	1	1
1	0	0	1	1	0	1	0	0
1	0	1	0	0	0	1	0	1
1	0	1	0	1	0	1	1	0
1	0	1	1	0	0	1	1	1
1	0	1	1	1	1	0	0	0
1	1	0	0	0	1	0	0	1
1	1	0	0	1	1	0	1	0
1	1	0	1	0	1	0	1	1
1	1	0	1	1	1	1	0	0
1	1	1	0	0	1	1	0	1
1	1	1	0	1	1	1	1	0
1	1	1	1	0	1	1	1	1
1	1	1	1	1	0	0	0	0

2) Складемо карти Карно для функцій переходів тригерів кожного розряду (рисунок 3.13). Карта переходів будується по таблиці станів і відображує перехід тригера $Q_i^n \rightarrow Q_i^{n+1}$ у кожному такті залежно від станів останніх тригерів в такті n.

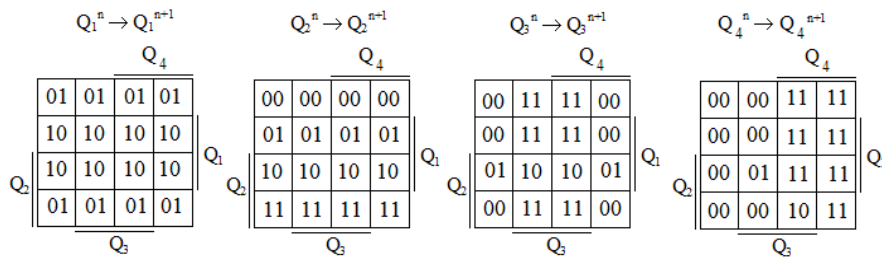


Рисунок 3.13 – Карти Карно для функцій переходів тригерів кожного розряду

3) Використовуючи словник переходів Т-тригера, для кожного входу тригера складаються карти Карно, в клітках яких проставляються сигнали, необхідні для забезпечення переходів тригерів, вказаних в однойменних клітках карт функцій переходів (рисунок 3.14).

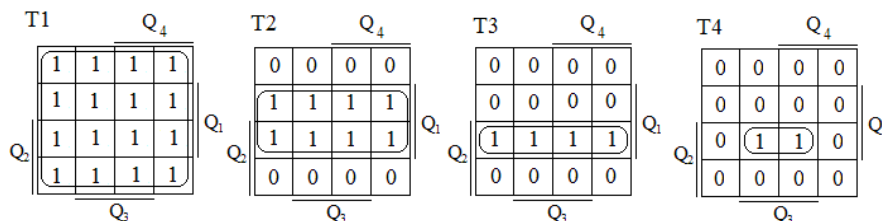


Рисунок 3.14 – Карти Карно для входів тригерів

Рівняння для входів тригерів:

$$T_1 = C \cdot 1 = C; \quad T_2 = C \cdot Q_1; \quad T_3 = C \cdot Q_2 Q_1; \quad T_4 = C \cdot Q_3 Q_2 Q_1$$

4) Будуємо схему лічильника з паралельним перенесенням (рисунок 3.15).

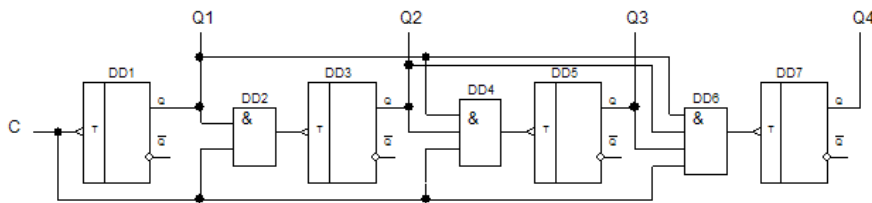


Рисунок 3.15 – Двійковий лічильник з паралельним перенесенням

Регістр зрушення можна перетворити на кільцевий лічильник, якщо вихід останнього тригера з'єднати з входом D першого (рисунок 3.16).

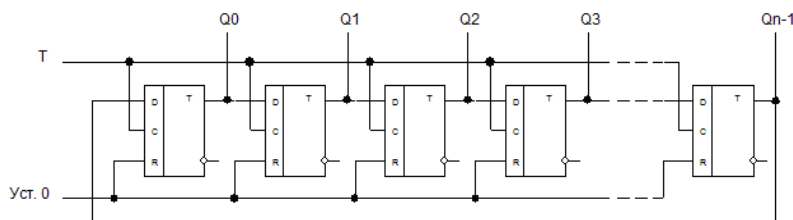


Рисунок 3.16 – Кільцевий лічильник з регістра зрушення

Перед початком рахунку імпульсом початкової установки в нульовий розряд лічильника Q_0 записується логічна 1, в останні розряди логічні 0. З початком рахунку кожен з рахункових імпульсів T , що приходять, перезаписує 1 в наступний тригер і число імпульсів, що поступили, визначається по номеру виходу, на якому є 1. Останній $n-1$ імпульс переведе в одиничний стан останній тригер, а n – імпульс перенесе цей стан на вхід нульового тригера і рахівниць почнеться спочатку. Таким чином, можна побудувати кільцевий лічильник з довільним коефіцієнтом рахунку.

Генератори числових послідовностей називають також розподільниками сигналів, оскільки послідовність двійкових чисел на їх виходах використовується для управління роботою інших цифрових вузлів. 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.

Розглянемо генератор чисел 4 – 3 – 2 – 12. Оскільки $L_n = 4$, то за основу генератора можна взяти двохрозрядний лічильник $K_{\text{рах.}} = 4$, 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.

Для побудови генератора використовуємо лічильник, побудований на двох JK-тригерах. Використовуємо схему віднімаючого лічильника, отже, подамо сигнал перенесення на тригер старшого розряду не з прямого виходу попереднього тригера, а з інверсного. Складемо таблицю переходів станів розрядів лічильника (таблиця 3.2).

Складемо карти Карно для функцій переходів тригерів кожного розряду (рисунок 3.17).

Таблиця 3.2 – Таблиця станів розрядів лічильника генератора

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

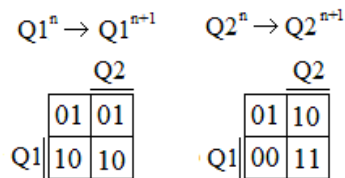


Рисунок 3.17 – Карти функцій переходів тригерів генератора

Використовуючи словник переходів JK-тригера (рисунок 3.8) для кожного входу тригера, складемо карти Карно, в клітинках яких проставимо сигнали, необхідні для забезпечення переходів тригерів, вказаних в однойменних клітинках карт функцій переходів (рис. 3.18).

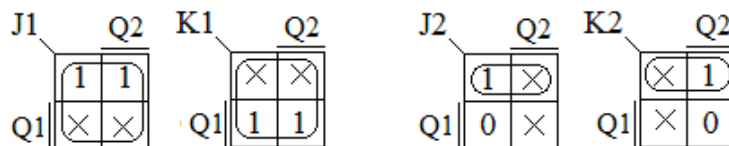


Рисунок 3.18 – Карти Карно функцій входів тригерів лічильника

Виконаємо мінімізацію логічних функцій входів в картах Карно з метою здобуття їх аналітичних виразів. У результаті отримуються наступні функції входів тригерів лічильника генератора:

$$J1 = 1; K1 = 1;$$

$$J2 = \overline{Q1}; K2 = \overline{Q1};$$

Складемо логічну схему лічильника генератора чисел (рисунок 3.19).

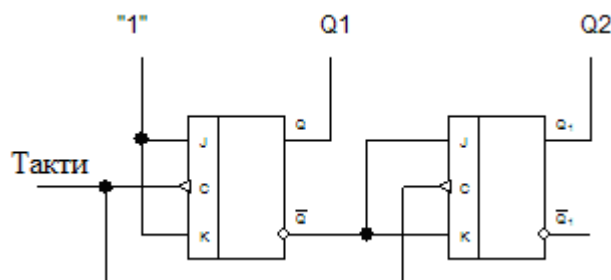


Рисунок 3.19 – Лічильник генератора чисел

Перевіримо алгоритм спрацьовування тригерів лічильника по тактах відповідно до рівняння JK-тригера: $Q^{n+1} = J\overline{Q}^n + \overline{K}Q^n$

Вихідний стан виходів тригерів:

$$Q1^n = 0; Q2^n = 0; \overline{Q1}^n = 1; \overline{Q2}^n = 1.$$

Спрацьовування тригерів по тактах:

- ① $Q1^{n+1} = 1 \cdot 1 + 0 \cdot 0 = 1$; $Q2^{n+1} = 1 \cdot 1 + 0 \cdot 0 = 1$;
- ② $Q1^{n+1} = 1 \cdot 0 + 0 \cdot 1 = 0$; $Q2^{n+1} = 1 \cdot 0 + 0 \cdot 1 = 0$;
- ③ $Q1^{n+1} = 1 \cdot 1 + 0 \cdot 0 = 1$; $Q2^{n+1} = 0 \cdot 1 + 1 \cdot 0 = 0$;
- ④ $Q1^{n+1} = 1 \cdot 0 + 0 \cdot 1 = 0$; $Q2^{n+1} = 1 \cdot 1 + 0 \cdot 0 = 1$.

Це відповідає таблиці станів розрядів лічильника генератора чисел.

Розрахуємо комбінаційну схему, яка виконує перетворення коду відповідно до таблиці функціонування (таблиця 3.3).

Таблиця 3.3 – Алгоритм функціонування комбінаційної схеми генератора чисел

Число	Q2	Q1	F4	F3	F2	F1
4	1	1	0	1	0	0
3	0	0	0	0	1	1
2	0	1	0	0	1	0
12	1	0	1	1	0	0

У результаті отримуємо наступні функції виходів генератора числової послідовності 4 – 3 – 2 – 12:

$$F1 = \overline{Q2} \cdot \overline{Q1};$$

$$F2 = \overline{Q2} \cdot \overline{Q1} + \overline{Q2} \cdot Q1 = \overline{Q2} \cdot (\overline{Q1} + Q1) = \overline{Q2} \cdot 1 = \overline{Q2};$$

$$F3 = Q2 \cdot Q1 + Q2 \cdot \overline{Q1} = Q2 \cdot (Q1 + \overline{Q1}) = Q2 \cdot 1 = Q2;$$

$$F4 = Q2 \cdot \overline{Q1}.$$

З метою скорочення кількості елементів схеми проінвертуємо вихідні сигнали F1 і F2:

$$F1 = \overline{\overline{Q2} \cdot \overline{Q1}} = \overline{\overline{Q2} + \overline{Q1}};$$

$$F2 = \overline{Q2};$$

$$F3 = Q2;$$

$$F4 = \overline{Q2} \cdot Q1 = \overline{\overline{\overline{Q2} + Q1}}.$$

Схема генератора числової послідовності представлена на рисунку 3.20.

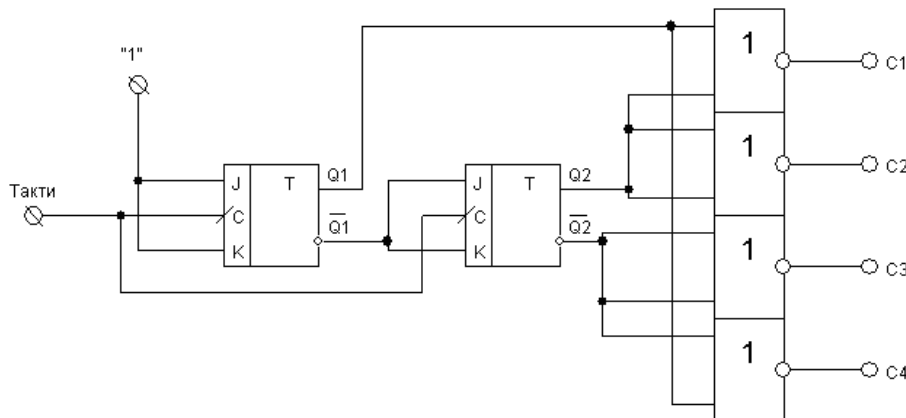


Рисунок 3.20 – Генератор числової послідовності 4 – 3 – 2 – 12 на основі лічильника з вихідною комбінаційною схемою

Часова діаграма роботи генератора (рисунок 3.21) показує, що періодична зміна вихідних комбінацій відбувається згідно з заданою числовою послідовністю. Розробка такого генератора за звичайною методикою проектування недвійкових лічильників зажадала б в два рази більше лічильників і додаткової логіки. Синтез структури генераторів послідовностей на регістрах зсуву, в першу чергу, полягає в знаходженні вигляду функцій входів. Відповідно до схеми регістра зсуву (рисунок 4.25) на вхід X подається періодична послідовність інформаційних «0» і «1».

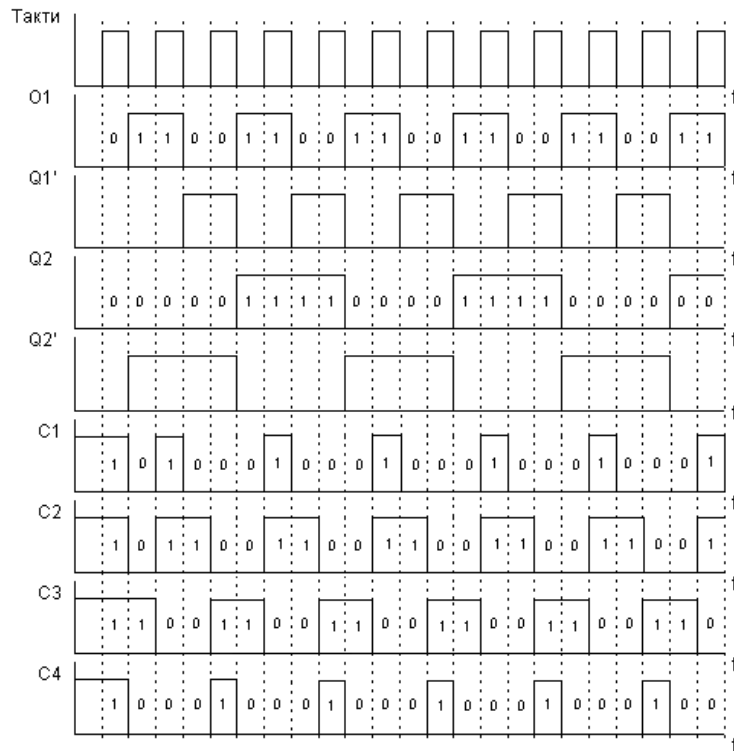


Рисунок 3.21 – Часова діаграма роботи генератора

Питання для закріплення вивченого матеріалу та самоконтролю.

1. Тригери. Таблиці станів, рівняння функціонування, словники переходів.
2. Принцип побудови регістрів з паралельним прийомом і паралельною видачею інформації.
3. Принцип побудови регістрів з послідовним прийомом і паралельною видачею інформації.
4. Принцип побудови двійкового лічильника з послідовним перенесенням інформації.
5. Принцип побудови двійкового лічильника з паралельним перенесенням інформації.
6. Принцип побудови кільцевого лічильника.
7. Принцип побудови генератора числової послідовності.

Практичні завдання

1. Розробити схему послідовно - паралельного чотирьох розрядного регістра на D-тригерах.

2. Розробити схему послідовного лічильника з модулем рахунку $K_{\text{рах}} = 8$.. Представити діаграму функціонування.
3. Розробити схему кільцевого лічильника на D-тригерах з модулем рахунку $K_{\text{рах}} = 8$. Представити діаграму функціонування.
4. Розробити структуру двійкового лічильника на T-тригерах з $K_{\text{рах}} = 16$.. Представити діаграму функціонування.
5. Розробити схему 12 – розрядного регістра пам'яті на D – тригерах. Представити діаграму функціонування.
6. Розробити схему чотирьох розрядного двійкового лічильника паралельним перенесенням інформації на D-тригерах. Представити діаграму функціонування.
7. Розробити схему паралельного лічильника з модулем рахунку $K_{\text{рах}} = 10$ на JK-тригерах. Представити діаграму функціонування.
8. Розробити схему генератора числової послідовності за заданими параметрами. Представити діаграму функціонування.

4 Синтез цифрових програмних автоматів

Метою вивчення теми є засвоєння принципів розрахунку, аналізу і синтезу цифрових автоматів.

Ключові терміни та поняття: комбінаційна схема, алфавіт станів, алфавіт вхідних значень, алфавіт вихідних значень, функція переходів, автомат Мура, автомат Мілі, С- автомат,

План самостійного опрацювання теми.

1. Засвоїти способи опису і завдання цифрових автоматів.
2. Засвоїти синтез частково визначених цифрових автоматів.
3. Засвоїти графічний спосіб завдання цифрових автоматів.
4. Засвоїти абстрактний синтез цифрових автоматів.

Методичні вказівки до вивчення питань та виконання завдань.

Цифровий (дискретний) автомат (ЦА) – пристрій, який здійснює прийом, зберігання і перетворення дискретної інформації по деякому алгоритму. Прикладами цифрових автоматів можуть служити живі організми, процесори, побутова техніка, калькулятори – це реальні пристрої, а також абстрактні, наприклад, моделі алгоритмів. Цифрові автомати можуть бути комбінаційного і послідовнісного типу.

Комбінаційною схемою прийнято називати пристрій з n входами і m виходами, в якого сукупність вихідних сигналів в даний момент часу повністю визначається сукупністю вхідних сигналів, що діють в даний момент часу і не залежить від вхідних сигналів, що діяли в попередні моменти часу.

Під синтезом комбінаційного цифрового автомату мається на увазі побудова логічної схеми в заданому базисі логічних елементів.

Синтез комбінаційних цифрових автоматів включає складання формалізованого завдання, перетворення логічної функції з метою оптимізації, з врахування наявної елементної бази і побудова принципової схеми. Як вихідні дані може виступати описове завдання, логічна функція або таблиця істинності. Пе-

ретворення і мінімізацію здійснюють за допомогою теорем і положень алгебри логіки, карт Карно і логічної схеми пристрою.

У деяких випадках при завданні автомата не визначені усі можливі переходи з окремих станів для конкретних вхідних сигналів. У таких випадках автомат називається частково визначеним. Ознакою часткової визначеності автомата є наявність незаповнених кліток в таблиці переходів і виходів.

Цифрові автомати визначаються тим, що значення виходів залежить не лише від вхідних значень, але і від поточного стану пристрою. Тобто вводиться поняття – стан. Для того, щоб зберігати дані про стан, в якому знаходиться пристрій в цифровому автоматі використовуються елементи, які запам'ятовують інформацію – тригери.

Автоматом з пам'яттю називають автомат, що описується функціями переходів і виходів, оператор якого є оператором з пам'яттю. Вихідні слова автомата з пам'яттю залежать не лише від вхідних слів, але і від послідовності їх вступу.

Автомат з пам'яттю має безліч внутрішніх станів, в які він переходить під впливом слів вхідного алфавіту. Наявність безлічі внутрішніх станів додає автомату здатність запам'ятовування вхідної інформації, що поступила на вхід автомата у минулому.

Математичною моделлю ЦА (а в загальному випадку будь-якого дискретного пристрою) є так званий абстрактний автомат, визначений як 6-компонентний об'єкт: $S = (Q^n, x, y, f, \lambda, Q_i)$.

1) $Q^n = \{Q_0, Q_1, \dots, Q_i\}$ – алфавіт станів – безліч станів, в яких може знаходитися проєктований цифровий автомат. Кількість станів грає важливу роль при реалізації ЦА. Чим більше станів, тим більше потрібний елементів (тригерів), що запам'ятовують, для побудови ЦА.

2) $x = \{x_1, x_2, \dots, x_m\}$ – алфавіт вхідних значень – безліч значень, які можуть поступати на вхід ЦА. Наприклад, якщо у автомата дворозрядний двійковий вхід, то елементами алфавіту можуть бути 00, 01, 10 і 11.

3) $y = \{y_1, y_2, \dots, y_n\}$ – алфавіт вихідних значень – безліч значень, які можуть бути встановлені на виході ЦА.

4) f – функція переходів $Q^{n+1} = f(Q^n(t), x(t))$. Функція переходів визначає, в який стан Q^{n+1} перейде автомат під впливом вхідного сигналу $x(t)$, якщо у нинішній момент часу автомат знаходиться в стані $Q^n(t)$.

5) λ – функція виходів $y(t) = \lambda(Q^n(t), x(t))$. Функція виходів визначає яке вихідне значення $y(t)$ буде встановлено на виході автомата залежно від вхідного значення $x(t)$ і поточного стану $Q^n(t)$.

6) $Q_i \in Q^n$ – початковий стан автомата – стан в який встановлюється ЦА після подачі живлення або після скидання.

На практиці найбільшого поширення набули два класи автоматів: автомати Мілі (Mealy) і Мура (Moore).

Закон функціонування автомата Мілі задається рівняннями:

$$Q^{n+1} = f(Q^n(t), x(t)); y(t) = f(Q^n(t), x(t)), \text{ де } t = 0, 1, 2, \dots$$

Закон функціонування автомата Мура задається рівняннями:

$$Q^{n+1} = f(Q^n(t), x(t)); y(t) = f(Q^n(t)), \text{ де } t = 0, 1, 2, \dots$$

Q^{n+1} – подальший стан автомата, Q^n – початковий стан автомата, $x(t)$ – вхідний сигнал, $y(t)$ – вихідний сигнал.

З порівняння законів функціонування видно, що, на відміну від автомата Мілі, вихідний сигнал в автоматі Мура залежить лише від поточного стану автомата і в явному вигляді не залежить від вхідного сигналу. Для повного завдання автомата Мілі або Мура додатково до законів функціонування, необхідно вказати початковий стан і визначити внутрішні, вхідні і вихідні множини попарно різних символів.

При табличному способі завдання автомата Мілі і Мура описуються за допомогою двох таблиць. Одна з них таблиця переходів, друга таблиця виходів.

При табличному способі завдання автомата Мілі таблиця переходів задає функцію $Q^{n+1} = f(Q^n(t), x(t))$, таблиця виходів – функцію $y(t) = f(Q^n(t), x(t))$.

Таблиця переходів відображує функцію переходів. Рядкам таблиці відповідають вхідні значення, які можуть поступати на входи ЦА, тобто в таблиці стільки рядків, скільки елементів у вхідному алфавіті. Стовпцям таблиці відповідають стани автомата, тобто стовпців стільки, скільки станів в автомата. На пересіченні i -стовпця і m -рядка в елементі таблиці вказується стан в яке перейде ЦА під впливом вхідного сигналу x_m (якому відповідає m -рядок) із стану Q_i (якому відповідає i -стовпець). Таблиця переходів приведена на рисунку 4.1.

	Вхідні стани ЦА				
Вхідний сигнал	Q0	Q1	Q2	⋮	Q _i
x0	Q2	Q _i	Q2	⋮	Q _i
...	...				
x _m	Q3	Q1	Q _i	⋮	Q2

Рисунок 4.1 – Таблиця переходів ЦА

Таблиця переходів має однаковий вигляд як для автомата Мура, так і для автомата Мілі. Для часткових автоматів Мілі і Мура в розглянутих таблицях на місці не певних станів і вихідних сигналів ставиться риска. У таких автоматах вихідний сигнал на якому-небудь переході завжди невизначений, якщо невизначеним є стан переходу.

Таблиця виходів для автомата Мілі має такий же вигляд як і таблиця переходів, лише на пересіченні i -стовпця і m -рядка в елементі таблиці вказується вихідне значення, яке сформує ЦА під впливом вхідного сигналу x_m (якому відповідає m -рядок) в стані Q_i (якому відповідає i -стовпець). Таблиця виходів автомата Мілі приведена на рисунку 4.2.

	Вхідні стани ЦА				
Вхідний сигнал	Q0	Q1	Q2	⋮	Q _m
x0	y0	y3	y _m	⋮	y4
...	...				
x _m	y2	y1	y5	⋮	y _m

Рисунок 4.2 – Таблиця виходів автомата Мілі

Таблиця виходів для автомата Мура складається з одного рядка. Стівпцям таблиці відповідають стани автомата, тобто в таблиці стільки стовпців, скільки станів в автомата. Приклад таблиці наведений на рисунку 4.3.

	Вхідні стани ЦА				
Вхідний сигнал	Q0	Q1	Q2	⋮	Q _m
x _m	y0	y3	y _m	⋮	y4

Рисунок 4.3 – Таблиця виходів автомата Мура

При графічному способі автомат задається у вигляді орієнтованого графа, вершини якого відповідають станам, а дуги – переходам між ними. Дуга, направлена з вершини Q_m, задає перехід в автоматі із стану Q_m в стан Q_s. На початку цієї дуги записується вхідний сигнал x_i, що викликає даний перехід Q_s = f(Q_m, x_i). Для графа автомата Мілі вихідний сигнал y_n, що формується на переході, записується в кінці дуги, а для автомата Мура – поряд з вершиною Q_m, відміченою станом Q_m в якому він формується. Якщо перехід в автоматі із стану Q_m в стан Q_s виробляється під дією декількох вхідних сигналів, то дузі графа, направлений з Q_m в Q_s, приписуються всі ці вхідні і відповідні вихідні сигнали. Граф автомата Мура представлений на рисунку 4.4, а автомата Мілі на рисунку 4.5.

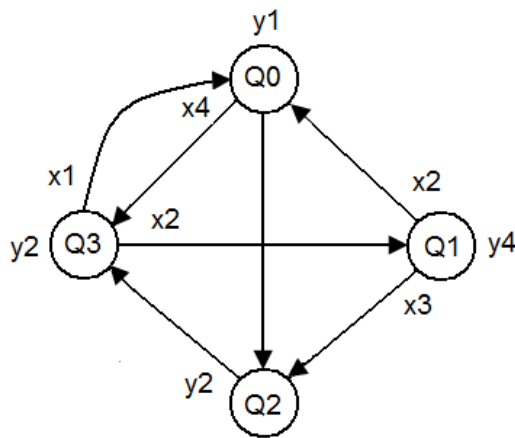


Рисунок 4.4 – Графічне представлення автомата Мура

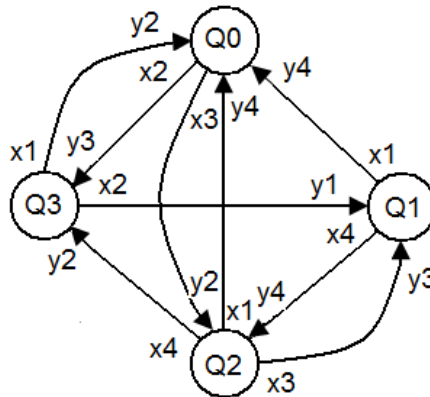


Рисунок 4.5 – Графічне представлення автомата Мілі

Автомат складається з набору m елементарних автоматів (тригерів T_1, T_2, \dots, T_m), комбінація станів яких в кожен момент часу визначає внутрішній стан в цілому всього автомата. Під впливом вхідних сигналів автомат повинен переходити з одного стану в інший. Для зміни стану автомата необхідно перемикнути один або декілька тригерів, що визначають стан автомата. Перемикання тригерів здійснюється подачею сигналів q_i на відповідні входи. Оскільки новий стан автомата визначається тим, яким був його попередній стан і який набір вхідних сигналів, то і сигнали q_i є функціями вихідних сигналів тригерів (Q_1, Q_2, \dots, Q_m) і вхідних сигналів автомата (x_1, x_2, \dots, x_m).

Для формування сигналів q_i управління тригерами використовується комбінаційний пристрій. Структура цього пристрою визначає функцію переходів автомата. Функція виходів реалізується іншим комбінаційним пристроєм, що формує вихідні сигнали автомата (y_1, y_2, \dots, y_k).

Таким чином, для синтезу автомата необхідно виконати наступне:

а) визначити, яка комбінація станів тригерів відповідатиме кожному з внутрішніх станів автомата, тобто провести кодування внутрішніх станів автомата;

б) синтезувати комбінаційний пристрій формування сигналів q_i , управління тригерами використовуючи таблицю переходів;

в) синтезувати комбінаційний пристрій, що формує вихідні сигнали y_i автомата, використовуючи таблицю виходів.

Питання для закріплення вивченого матеріалу та самоконтролю.

1. Принцип синтезу комбінаційних цифрових автоматів.
2. Принцип синтезу частково визначених цифрових автоматів.
3. Графічний спосіб завдання цифрових автоматів.
4. Абстрактний синтез цифрових автоматів Мілі.
5. Абстрактний синтез цифрових автоматів Мура.

Практичні завдання

1. Синтезувати комбінаційний цифровий автомат у базисі Шефера, який заданий рівнянням

$$F = A \cdot \bar{B} \cdot \bar{C} \cdot D + \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C \cdot D$$

2. Синтезувати комбінаційний цифровий автомат у базисі Пірса, який заданий рівнянням

$$F = (A + \bar{B} + C + \bar{D})(\bar{A} + B + C + D)(\bar{A} + \bar{B} + C + \bar{D})(\bar{B} + \bar{C} + \bar{D})(A + B + C + D)(A + \bar{B} + D)$$

3. Провести синтез автомата Мілі, функціонування якого описується заданими таблицями переходів і виходів на RS-тригерах.

Таблиця переходів автомата Мілі

Вхідний сигнал	Q0	Q1	Q2	Q3
0	Q0	Q3	Q2	Q1
1	Q3	Q1	Q2	Q0

Таблиця виходів автомата Мілі

Вхідний сигнал	Q0	Q1	Q2	Q3
0	1	1	0	1
1	0	0	1	1

4. Провести синтез автомата Мілі, функціонування якого описується заданими таблицями переходів і виходів на JK-тригерах.

Таблиця переходів автомата Мілі

Вхідний сигнал	Q0	Q1	Q2	Q3
0	Q1	Q2	Q3	Q1
1	Q2	Q3	Q1	Q0

Таблиця виходів автомата Мілі

Вхідний сигнал	Q0	Q1	Q2	Q3
0	1	0	1	0
1	1	0	1	1

5 Запам'ятовувальні пристрої

Метою вивчення теми є засвоєння принципів побудови елементів оперативної і постійної пам'яті цифрової схемотехніки.

Ключові терміни та поняття: комірка пам'яті, матрична організація, запис інформації, зчитування інформації, швидкодія, інформативна ємність, питома потужність, накопичувач, схема управління.

План самостійного опрацювання теми.

1. Засвоїти розділення запам'ятовувальних пристроїв по функціональних ознаках.

2. Засвоїти організацію статичних та динамічних комірок пам'яті.

3. Засвоїти принцип функціонування ОЗП.

4. Засвоїти принцип функціонування ПЗП.

Методичні вказівки до вивчення питань та виконання завдань.

Пристрої пам'яті (Memory Devices), чи запам'ятовувальні пристрої (ЗП), належать до одних з найважливіших функціональних пристроїв сучасних цифрових електронних систем. Вони слугують для фіксування, зберігання та обміну певного об'єму інформації. Якщо раніше ЗП застосовувались виключно для ЕОМ, то тепер їх використовують як складові частини пристроїв радіолокації, телебачення, вимірювальної та побутової техніки, мобільних платформ тощо.

В основі роботи запам'ятовуючих пристроїв лежать різноманітні фізичні ефекти в напівпровідниках, оптичних середовищах, магнітних матеріалах тощо. Ефекти в напівпровідникових матеріалах використовуються в пристроях оперативної пам'яті та енергонезалежної флеш-пам'яті, ефекти в оптичних середовищах – у DVD дисках, а ефекти в магнітних матеріалах – у твердих дисках (вінчестерах).

Оперативні ЗП (ОЗП, RAM - Random Access Memory – пам'ять з довільною вибіркою) виконують операцію звернення (запису і зчитування) довільної двійкової інформації майже за однаковий час. ОЗП призначені для тимчасового зберігання поточної інформації та швидкого звертання до запам'ятовувачів. Розрізняють енергозалежні ОЗП, які зберігають інформацію лише за наявності напруги живлення, та енергонезалежні.

Постійні ЗП (ПЗП, ROM – Real Only Memory – пам'ять тільки для читання) призначені для тривалого зберігання та швидкого зчитування постійно записаної раніше інформації, зміст якої під час експлуатації ЗП не змінюється, а за відсутності живлення не руйнується. Такий тип пам'яті позначають - nonvolatile memory. Розрізняють перепрограмовані ПЗП, які передбачають процес запису. Цей процес реалізують за допомогою програматорів.

Запам'ятовувальні пристрої цифрової схемотехніки можна розрізнити по функціональних ознаках.

1) надоперативні запам'ятовувальні пристрої (НОЗП) – здійснюють зчитування і запис інформації (регістрова пам'ять).

2) оперативні запам'ятовувальні пристрої (ОЗП) – здійснюють зчитування і запис інформації.

3) маскові постійні запам'ятовувальні пристрої (МПЗП) – здійснюють лише зчитування інформації (їх програмують до або під час виготовлення за допомогою спеціальних фотошаблонів-масок).

4) програмовані постійні запам'ятовувальні пристрої (ППЗП) – здійснюють лише зчитування інформації постійно і запис одноразово (програмуються одноразово за допомогою спеціального пристрою – програматора).

5) репрограмовані або перепрограмовані постійні запам'ятовувальні пристрої (РПЗП) - здійснюють зчитування інформації постійно і запис при необхідності (їх можна програмувати повторно з можливістю багаторазового стирання та записування нової інформації (програматором) різними способами – електричними імпульсами, ультрафіолетовими чи оптичними променями).

Найважливіші параметри запам'ятовувальних пристроїв (ЗП).

1) інформаційна ємність – максимально можливий об'єм інформації, що зберігається. Параметр, що характеризує міру інтеграції елементів на кристалі. Виражається в бітах або словах в байтах. Біт зберігається запам'ятовувальним елементом (ЗЕ), а слово – коміркою пам'яті (КП), тобто групою ЗЕ, до якої можливо лише одночасне звернення.

2) питома потужність – загальна потужність, яка споживається в режимі зберігання, віднесена до одного біта інформації.

3) швидкодія – час запису, зчитування і тривалості циклів запису/зчитування.

Час запису – інтервал після появи сигналу запису і встановленням комірки пам'яті в стан, який задається вхідним словом.

Час зчитування – інтервал між моментами появи сигналу читання і слова на виході ЗП.

Цикли запису і зчитування – це час між двома послідовностями запису або зчитування. Тривалість циклів може перевищувати часи запису і зчитування, оскільки після цих операцій потрібен час для відновлення початкового стану ЗП.

4) питома вартість одного біта інформації – загальна вартість кристала, поділена на інформаційну ємність.

Всяке ОЗП складається з двох основних частин: накопичувача і схем управління.

Накопичувач – це основна частина ОЗП, де зберігаються дані.

Схеми управління призначені для введення і виведення цих даних. У них входять дешифратори, підсилювачі, регістри, різного роду ключі, комутатори і інші схеми загального призначення.

Накопичувач складається з комірок пам'яті, кожна з яких зберігає один біт інформації (0 або 1). Основу КП складають бістабільні комірки, основною властивістю яких є наявність двох стійких станів $Q = 0$ або $Q = 1$. На рисунку 5.1 показана типова матрична організація ОЗП, де КП розташовані у вузлах ґрат, утворених адресними шинами x і y .

Кількість комірок 16 (4 горизонтальних шини на 4 вертикальних шини). Кожна комірка пов'язана з однією вертикальною і однією горизонтальною шиною. Тому, якщо подається напруга на одну вертикальну і одну горизонтальну шини, то до периферії підключається певна КП. У цю комірку можна записати, або зчитати з неї 1 біт інформації. Запис і зчитування виконується за допомогою розрядних шин РШ1, РШ0, які підключені до усіх КП.

Комірки пам'яті, які використовуються у накопичувачах всілякі.

Комірка на однотипних МОН транзисторах з р-каналом (рисунок 5.2) має класичну структуру RS-тригера з управляючими ключами VT5, VT6.

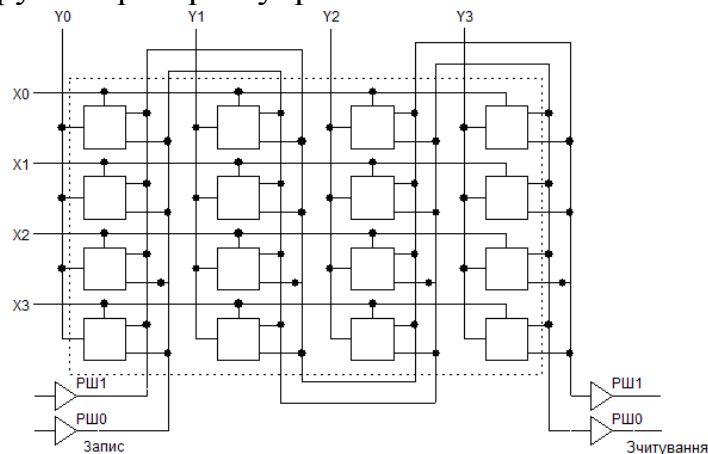


Рисунок 5.1 - Матрична організація ОЗП

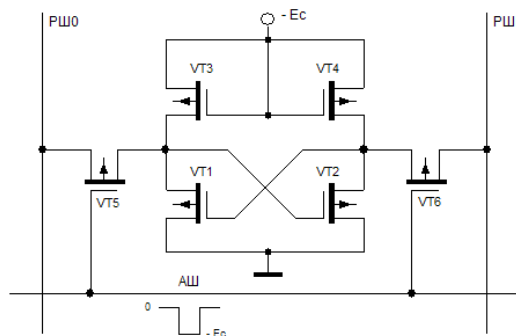


Рисунок 5.2 – Статичний елемент пам'яті на однотипних МОН транзисторах

Ці ключі нормально замкнуті і КП відключена від розрядних шин. Під час вступу негативного імпульсу $-E_c$ на адресну шину ключі VT5, VT6 відкриваються і підключають комірку до розрядних шин. При цьому на розрядні шини поступають рівні Q і \bar{Q} , записані у комірці. У режимі запису на адресну шину

також подається імпульс $-E_c$, а на розрядні шини подаються необхідні (взаємно протилежні) рівні, які переводять комірку у відповідний стан. Таким чином імпульс на адресній шині в обох режимах грає роль тактового імпульсу.

У комірці динамічного типу (рисунк 5.3) зберігання біта інформації здійснюється ємкостями $C1$, $C2$ (звичайно це паразитні ємкості МОН транзисторів).

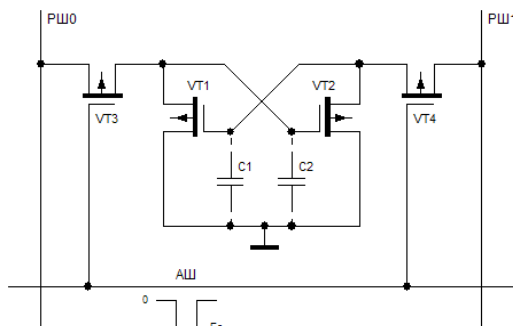


Рисунок 5.3 – Динамічний елемент пам'яті на однотипних МОН транзисторах

Методика запису і зчитування та ж, що і у комірці статичного типа. Нехай при запису на розрядні шини PШ1, PШ0 подані відповідні рівні $-E_c$ і 0. Рівень $-E_c$ через ключ VT4 поступить на затвор транзистора VT1 і цей транзистор буде відкритий. На затвор транзистора VT2 поступить рівень 0 і цей транзистор буде замкнутий. Напруга на ємкостях матиме значення $U_{c1} = -E_c$, $U_{c2} = 0$. Якщо залишковий струм замкнутого транзистора VT2 досить малий, то ємкість $C1$ розряджатиметься вельми повільно і, отже, напруга $-E_c$ і 0 зберігатимуться на виходах вічка (на стоках) тривалий час.

Динамічні комірки пам'яті, значно економічніше статичних, оскільки у них відсутнє джерело живлення. Отже, в режимі зберігання вони не споживають потужність.

МОН транзисторні комірки економічніше і компактніше біполярних (рисунк 5.4) але поступаються їм по швидкодії.

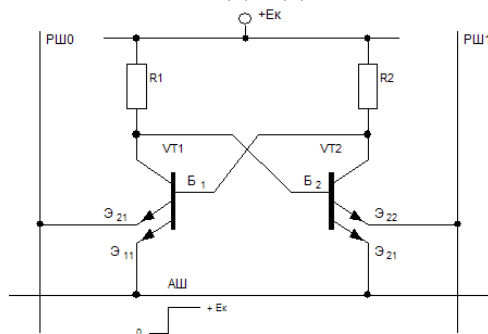


Рисунок 5.4 – Статичний елемент пам'яті на біполярних транзисторах

Нехай в режимі зберігання транзистор VT2 замкнутий, а VT1 відкритий і насичений. Тоді $U_{b1} = U^*$ і $U_{b2} = 0$. Якщо на розрядних шинах заданий невеликий (0,1...0,2 В) «черговий» потенціал, то емітерний перехід E_{21} буде практично замкнутий і весь струм протікатиме через E_{11} . У транзисторі VT2 будуть замкнуті обидва емітерних переходи.

При зчитуванні, коли на адресну шину подається позитивна напруга E_k емітер E_{11} закривається і струм транзистора $VT1$ протікає в шину $RШ0$ через емітер E_{21} , який залишився під низьким потенціалом. Шина $RШ1$ залишається знеструмленою.

При запису одночасно з адресним імпульсом подається напруга на ту розрядну шину, яка пов'язана з транзистором, який підлягає замиканню. Якщо подати $+E_k$ на розрядну шину $RШ1$, то транзистор $VT2$ залишиться замкнутим і стан КП не зміниться. Якщо ж подати $+E_k$ на шину $RШ0$, то виявляться замкнутими обидва емітера транзистора $VT1$. Тоді струм потече через базу транзистора $VT2$ в емітер E_{22} , який знаходиться під низьким потенціалом шини $RШ1$. При цьому транзистор $VT2$ відкриється і стан КП зміниться на протилежний.

Запис інформації в ПЗП виконується або один раз і назавжди, або є спеціальною, рідко використовуваною операцією. Тому ПЗП допускають лише прочитування занесеної до них інформації. У постійному запам'ятовуючому пристрої, за кожною n -розрядною адресою записано одне заздалегідь встановлене m -розрядне слово. Таким чином, ПЗП є комбінаційною схемою, що перетворює код адреси в код слова.

Запам'ятовуюча матриця ПЗП представляє собою систему взаємно перпендикулярних шин, в пересіченнях яких або стоїть (логічна «1»), або відсутній (логічний «0») елемент, що зв'язує між собою відповідні горизонтальну і вертикальну шини. Вибірка слів виконується так само, як і в ОЗП, за допомогою дешифратора.

Для реалізації простого ПЗП достатньо використати дешифратор та відповідну кількість діодів для побудови матриці-накопичувача.

Спочатку елементи поєднання розташовані у всіх вузлах матриці і в такому однорідному вигляді матрицю поставляють замовникові (рисунок 5.5).

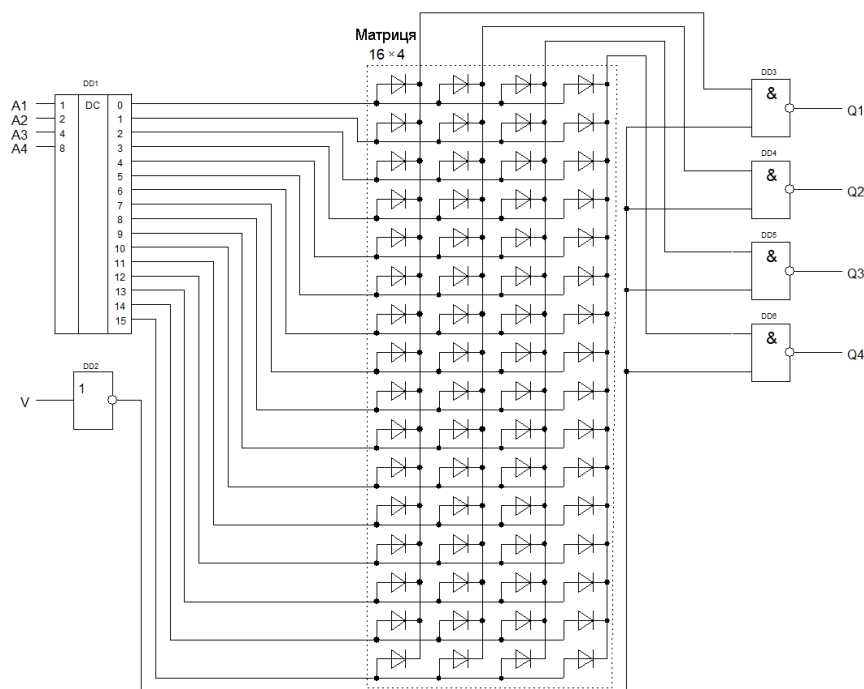


Рисунок 5.5 – Функціональна схема ПЗП

Кожен замовник сам записує в ПЗП потрібні йому коди. Для цього він (за допомогою спеціальних пристроїв) перепалює виводи-перемички тих діодів, які знаходяться в місцях розташування логічних «0» (рис. 5.7 а). Для того, щоб перегорали саме виводи діодів, а не прилеглі до них ділянки адресної і розрядної шин, виводи робляться більш високоомними и більш легкоплавкими, ніж шини. Недоліком діодних ПЗП є те, що необхідний струм в розрядних шинах повинен забезпечуватися дешифратором, який передає цей струм через адресну шину. Для того, щоб полегшити роботу дешифратора, замінюють діоди транзисторами (5.7 б).

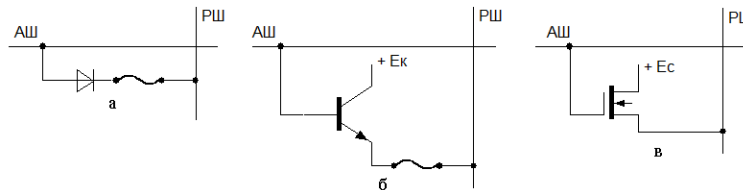


Рисунок 5.7 – Елементи пам'яті на діоді (а), біполярному транзисторі (б) і МДН-транзисторі (в)

Вихідні транзистори підсилювачів можуть бути з відкритим колектором або з третім станом. Тоді при опорному сигналі $V = 1$ мікросхема відключається від вихідної шини, що дозволяє нарощувати пам'ять простим об'єднанням виходів мікросхем ПЗП.

При використанні МДН-транзисторів (5.7 в) запис інформації здійснюється металізацією затворів на етапі останньої фотолітографії. Металізація затворів виконується лише в тих транзисторах, які повинні передавати «1» на розрядну шину. В останніх транзисторах затвори не будуть приєднані до адресних шин і ці транзистори не діятимуть.

Сучасним і найпоширенішим типом репрограмованих ПЗП є енергонезалежна напівпровідникова пам'ять типу EEPROM (Electrically Erasable Programmable Read-Only Memory). Одним з різновидів EEPROM є флеш-пам'ять - Flash Memory, особливістю якої є поблочне конфігурування комірок пам'яті та спосіб стирання інформації.

Базовим елементом EEPROM є спеціально оптимізована структура польового МДН транзистора з двошаровим діелектриком (рисунок 5.8 а). Транзистори з такою двошаровою структурою називаються МНОН (метал-нітрид-окисел-напівпровідник) транзистори.

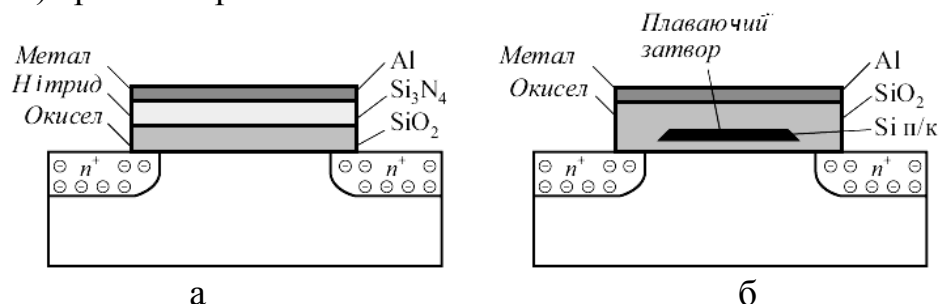


Рисунок 5.8 – Структура напівпровідникової пам'яті: а) МНОН – транзистор; б) МДН транзистор з плаваючим затвором (флеш-пам'ять)

У флеш-пам'яті використовується структура з плаваючим затвором (Floating Gate), яка дає змогу вирівняти вздовж каналу потенціал, зумовлений тунельованими в підзатворний діелектрик електронами (рисунок 5.8 б).

Операція програмування флеш-пам'яті, що реалізується зарядом плаваючого затвора, проводиться лавинною інжекцією електронів зі стокової області каналу МДН транзистора (рисунок 5.9). Якщо заряд плаваючого затвора одинбітного МДН транзистора менший ніж 500 електронів, то це означає, що комірки зберігають логічну одиницю, а якщо заряд перевищує 30000 електронів, то – логічний нуль. Заряд комірки викликає зміну порогової напруги транзистора, і під час операції зчитування вимірюється величина цієї порогової напруги, а за нею визначають кількість заряду на плаваючому затворі.

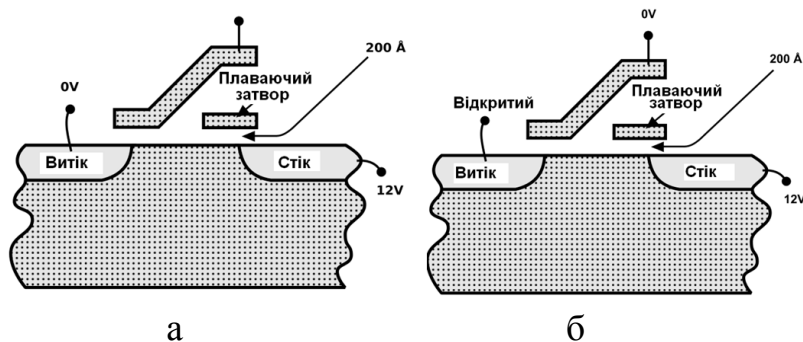


Рисунок 5.9 – Програмування (а) та стирання (б) флеш-пам'яті

Найбільш часто використовують два типи флеш-пам'яті, в яких запам'ятовувальні транзистори підключені до розрядних шин паралельно або послідовно.

Питання для закріплення вивченого матеріалу та самоконтролю.

1. Пояснити розрізнення елементів пам'яті по функціональних ознаках.
2. Пояснити принцип функціонування ОЗП.
3. Пояснити функціонування статичних елементів пам'яті.
4. Пояснити функціонування динамічних елементів пам'яті.
5. Пояснити принцип функціонування ПЗП.
6. Пояснити принцип функціонування репрограмуємих елементів пам'яті.
7. Пояснити принцип функціонування флеш-пам'яті.

Практичні завдання

1. Побудувати віртуальну установку у програмному забезпеченні Electronics Workbench для дослідження ПЗП.

2. Побудувати віртуальну установку у програмному забезпеченні Electronics Workbench для дослідження ОЗП.

3. Побудувати схему ПЗП з діодною організацією матриці 4×16. Записана інформація:

0000, 0001, 0010, 0100, 1000, 1100, 1110, 1111,
1110, 1100, 1000, 0000, 0110, 1001, 0110, 0000.

4. Побудувати схему цифрового автомату з проміжною пам'яттю покрокового виконання заданого алгоритму функціонування.

6 Питання для повторення та актуалізації знань

1. Представлення чисел в цифровій схемотехніці
2. Коди чисел в цифровій схемотехніці
3. Арифметичні операції в двійковій системі числення
4. Основні логічні операції булевої алгебри
5. Аксиоми і закони булевої алгебри
6. Форми представлення логічних функцій
7. Основи синтезу цифрових пристроїв
8. Розрахунковий метод мінімізації логічних функцій
9. Метод мінімізуючих карт Карно
10. Запис структурних формул в універсальних базисах
11. Система параметрів цифрових мікросхем
12. Логічні інвертори на біполярному та МОН транзисторі
13. Елементи діодно-транзисторної логіки
14. Елементи транзисторно-транзисторної логіки
15. Елементи емітерно-зв'язаної логіки.
16. Елементи інтегральної інжекційної логіки
17. Логічні елементи на МОН-транзисторах та комплементарних ключах
18. Принцип побудови інтегрального суматора
19. Схема порівняння кодів
20. Схема контролю парності (непарності)
21. Мажоритарні елементи
22. Принципи побудови інтегрального дешифратора
23. Принцип побудови інтегрального мультиплексора.
24. Принцип побудови інтегрального демультимплексора.
25. Принципи побудови інтегрального шифратора
35. Бістабільна комірка.
36. RS-тригер в інтегральному виконанні.
37. RST-тригер в інтегральному виконанні.
38. T-тригер в інтегральному виконанні.
39. JK-тригер в інтегральному виконанні.
40. D-тригер в інтегральному виконанні.
41. Принцип побудови інтегральних регістрів.
42. Паралельний регістр на тактуємих D-тригерах.
43. Послідовний регістр на тактуємих D-тригерах.
44. Принцип функціонування універсального регістра К 155 ИР1.
45. Двійковий лічильник з послідовним переносом.
46. Сумуючий двійковий лічильник з паралельним переносом.
47. Кільцевий лічильник з регістра зсуву.
48. Принцип функціонування лічильника К 155 ИЕ5.
49. Двійковий декадний лічильник.
50. Двійковий лічильник з вагою розрядів 5-2-1-1.
51. Генератори цифрових послідовностей.
52. Цифровий програмний автомат Мура.

- 53. Цифровий програмний автомат Мілі.
- 54. Оперативні запам'ятовувальні пристрої.
- 55. Постійні запам'ятовувальні пристрої.
- 56. Постійні запам'ятовувальні пристрої, що перепрограмовуються.

ЛІТЕРАТУРА

Рекомендована

1. Верьовкін Л.Л., Світанько М.В., Кісельов Є.М., Хрипко С.Л. Цифрова схемотехніка : підручник. Запоріжжя: ЗДІА, 2016. 214 с.
2. Гершунский Б.С. Основы электроники и микроэлектроники. К. : Вища шк., 1987. 422 с.
3. Бойко В.І., Гуржій А.М., Жуйков В.Я. Основы схемотехніки електронних систем: підручник. К. : Вища шк., 2004. 527 с
4. Гельжинський І.І., Голяка Р.Л., Готра З.Ю., Марусенкова Т.А. Мікросхемотехніка: підручник. Львів : Ліга-Прес, 2015. 492 с.
5. Дудикевич В.Б., Кеньо Г.В., Петрович І.В. Електроніка та мікросхемотехніка. Частина І: Електроніка: навчальний посібник. Львів : Видавництво Львівської політехніки, 2010. 204 с.
6. Лукашук Л.О. Схемотехніка логічних та послідовнісних схем: навч. посіб. Л. : Видавництво Нац. університету «Львівська політехніка», 2004. 116 с.
7. Верьовкін Л.Л., Світанько М.В., Кісельов Є.М. Цифрова схемотехніка. Методичні вказівки до виконання курсового проекту на тему: «Синтез цифрового логічного автомата». Запоріжжя : ЗДІА, 2016. 34 с.

Використана

1. Бойко В. І., Багрій В. В. Цифрова схемотехніка. К. : ІЗМН, 2001. 228 с.
2. Алексенко А.Г., Шагурин И.И. Микросхемотехника. М. : Радио и связь, 1990. 496 с.
3. Тарабрин Б.В. Интегральные микросхемы: Справочник. М. : Энергоатомиздат, 1985. 528 с.
4. Якубовский С.В. Цифровые и аналоговые интегральные схемы: Справочник. М. : Радио и связь, 1989. 496 с.