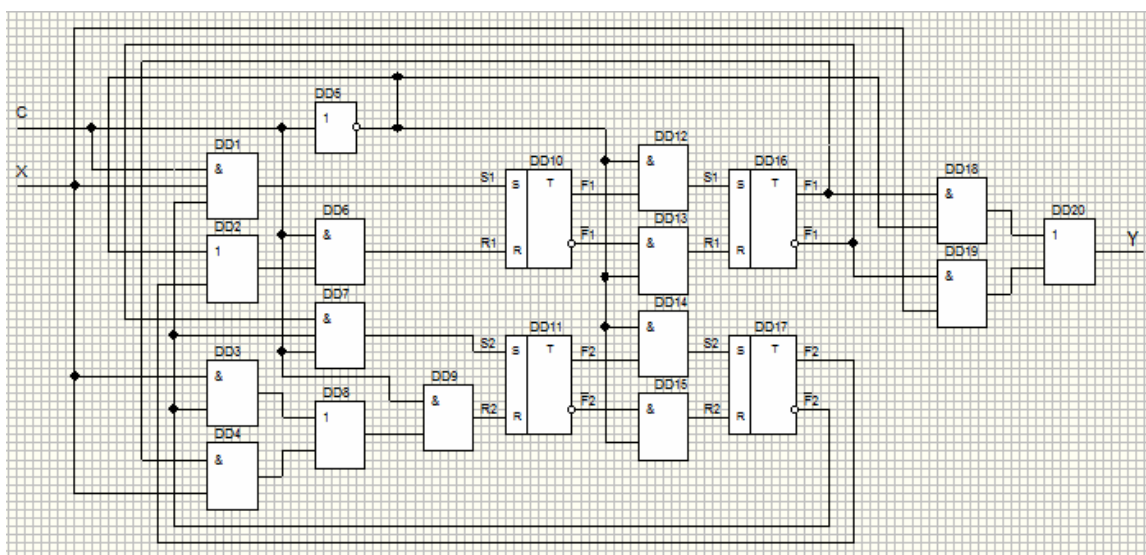




Л.Л. Верьовкін
М.В. Світанько
Є.М. Кісельов
С.Л. Хрипко

ЦИФРОВА СХЕМОТЕХНІКА

Підручник для студентів
технічних вузів і коледжів



Запоріжжя
2016

Міністерство освіти і науки України
Запорізька державна інженерна академія

Затверджено до друку
рішенням Вченої ради ЗДІА
протокол № 8 від 30.06.2016

ЦИФРОВА СХЕМОТЕХНІКА

Підручник для студентів
технічних вузів і коледжів

Рекомендовано до видання на засіданні кафедри МЕІС,
протокол №21 від 16.05.16

Запоріжжя
2016

Цифрова схемотехніка. Підручник для студентів технічних вузів і коледжів / Укл.: Л.Л. Верьовкін, М.В. Світанько, Є.М. Кісельов, С.Л. Хрипко. Запоріжжя. Видавництво ЗДІА. 2016. 214 с.

Підручник містить теоретичний матеріал по основним питанням аналізу, конструювання та розробки елементів цифрових електронних схем та систем, містить приклади розв'язання задач, контрольні питання та завдання для самостійної роботи студентів. Підручник може бути рекомендований також для осіб, які займаються розробкою та конструюванням приладів цифрової електроніки.

Укладачі:

Л.Л. Верьовкін – кандидат технічних наук, доцент
М.В. Світанько – кандидат фізико-математичних наук, доцент
Є. М. Кісельов – кандидат технічних наук, доцент
С. Л. Хрипко – кандидат фізико-математичних наук, доцент

Відповідальний за випуск:
Завдувач кафедри мікроелектронних
інформаційних систем
проф. Оселедчик Ю.С.

Рецензенти:

Пазюк М.Ю. – доктор технічних наук, професор ЗДІА
Кідалов В.В. – доктор фізико-математичних наук, професор БДПУ
Єрохов В.Ю. – доктор фізико-математичних наук, професор НУУ
«Львівська політехніка»

Зміст

Вступ	6
Розділ 1 Арифметичні та логічні основи цифрової схемотехніки	8
1.1 Представлення чисел в цифровій схемотехніці	8
1.2 Арифметичні операції в двійковій системі числення	12
1.3 Представлення інформації фізичними сигналами	15
1.4 Логічні основи цифрової схемотехніки	16
1.4.1 Основні операції булевої алгебри	18
1.4.2 Аксиоми і закони булевої алгебри	25
1.4.3 Форми представлення логічних функцій	29
1.5 Основи синтезу цифрових пристроїв	31
1.6 Мінімізація логічних функцій	35
1.6.1 Розрахунковий метод мінімізації	36
1.6.2 Метод мінімізуючих карт Карно	37
1.7 Запис структурних формул в універсальних базисах	40
Розділ 2 Елементна база статичних цифрових мікросхем	43
2.1 Система параметрів цифрових мікросхем	43
2.2 Логічний інвертор на біполярному транзисторі	49
2.2 Логічний інвертор на МОН транзисторах	51
2.3 Транзисторна логіка з безпосередніми зв'язками (ТЛБЗ)	54
2.4 Елементи діодно-транзисторної логіки (ДТЛ)	55
2.5 Елементи транзисторно-транзисторної логіки (ТТЛ)	58
2.6 Елементи емітерно-зв'язаної логіки (ЕЗЛ)	63
2.7 Елементи інтегральної інжекційної логіки (I^2L)	66
2.8 Логічні елементи на МОН-транзисторах	68
2.8.1 Логічні елементи на ключах з динамічним навантаженням	69
2.8.2 Логічні елементи на комплементарних ключах	70
2.9 Схемне моделювання логічних елементів	72
2.9.1 Моделювання схем на МОН та КМОН транзисторах	73

2.9.2 Моделювання схем на біполярних транзисторах	78
Розділ 3 Цифрові пристрої комбінаційного типу	90
3.1 Принцип побудови інтегрального суматора	90
3.1.1 Багаторозрядні суматори	94
3.2 Схема порівняння кодів	97
3.3 Схема контролю парності (непарності)	100
3.4 Мажоритарні елементи	102
3.5 Цифрова схема включення і виключення з декількох місць	107
3.6 Порогова логічна схема	108
3.7 Принципи побудови інтегрального дешифратора	110
3.8 Принцип побудови інтегрального демультіплексора	116
3.9 Принцип побудови інтегрального мультиплексора	119
3.10 Принцип побудови інтегрального шифратора	122
3.11 Перетворювачі кодів	131
Розділ 4 Цифрові пристрої послідовнісного типу	135
4.1 Бістабільна комірка	135
4.2 Тригери	136
4.2.1 RS – тригери	137
4.2.2 Синхронні RS – тригери	140
4.2.3 RS – тригери S, R і E – типів	141
4.2.2 D – тригери (тригери затримки)	142
4.2.3 Тригер T – типа (рахунковий тригер)	144
4.2.4 JK – тригери	148
4.3 Цифрові регістри	150
4.3.1 Паралельні регістри (регістри пам'яті)	151
4.3.2 Послідовні регістри (регістри зсуву)	152
4.3.3 Реверсивні регістри зсуву	155
4.3.4 Паралельно – послідовний регістр	157
4.3.5 Послідовно – паралельний регістр	158

4.4 Цифрові лічильники	158
4.4.1 Двійковий асинхронний підсумовуючий лічильник з послідовним перенесенням	159
4.4.2 Двійкові лічильники з паралельним перенесенням	162
4.4.3 Недвійкові лічильники з паралельним перенесенням	167
4.4.4 Двійково-десяткові лічильники	169
4.4.5 Лічильники, що працюють у двійковому коді з заданою вагою розрядів	171
4.4.6 Кільцеві лічильники	173
4.4.7 Лічильники Джонсона	175
4.5 Генератори числових послідовностей	177
Розділ 5 Запам'ятовуючі пристрої цифрової схемотехніки	185
5.1 Оперативні запам'ятовуючі пристрої	186
5.2 Постійні запам'ятовуючі пристрої	192
5.3 Репрограмуємі постійні запам'ятовуючі пристрої	194
Розділ 6 Контрольна робота «Синтез функціональних цифрових пристроїв»	196
Список літератури	210

Вступ

Основна мета дисципліни «Цифрова схемотехніка» – повне уявлення про елементну базу, принципи функціонування, характеристики і схемотехніку сучасних цифрових мікроелектронних приладів і мікросхем.

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

У сучасних системах автоматизації і управління знаходять широке використання засоби обробки даних, основу яких складають цифрові елементи і пристрої. Цифрова електроніка за короткий історичний період пройшла шлях 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. В процесі розвитку мікросхемотехніки виділилося декілька типів елементів, що мають досить хороші значення параметрів і зручних для реалізації в інтегральному виконанні.

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, які 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 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 також 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 достатньо простої моделі, а де потрібна складніша. Тобто людина ніколи не робитиме зайвої, надлишкової роботи і, отже, не вноситиме додаткових помилок, властивих будь-якій, навіть найскладнішій, моделі.

У підручнику авторами зроблена спроба компактного і доступного викладення матеріалу з максимальним наближенням до лекційної форми.

У кінці наводиться список використаної при підготовці даного посібника літератури, а також літератури, що рекомендується для поглибленого вивчення окремих питань.

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 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.).

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 (ІСТИНА і НЕПРАВДА, TRUE і FALSE, ТАК і НІ, 1 і 0). Найбільш поширено останнє позначення. При цьому 1 і 0 не можна трактувати як числа, над ними не можна виробляти арифметичні дії.

Логічні змінні добре описують стани таких об'єктів, як реле, тумблери, кнопки, тобто об'єктів, які можуть знаходитися в двох чітко помітних станах: включено – вимкнено. До таких об'єктів відносяться і напівпровідникові логічні елементи, на виході яких може бути лише один з двох чітко помітних рівнів напруги. Частіше вищий, або просто ВИСОКИЙ (HIGH) рівень береться за логічну одиницю, а нижчий, або просто НИЗЬКИЙ (LOW) – за логічний нуль.

1.1 Представлення чисел в цифровій схемотехніці

В даний час в буденному житті для кодування числової інформації використовується десяткова система числення з підставою 10 (A_{10}), в якій використовується 10 елементів позначення: числа 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. У першому (молодшому) розряді вказується число одиниць, в другому – десятків, в третьому – сотень і т. д. Іншими словами, в кожному наступному розряді вага розрядного коефіцієнта збільшується в 10 разів.

У цифрових пристроях обробки інформації використовується двійкова система числення з підставою 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 з права наліво від молодших розрядів до старших збільшується в 2 рази, тобто має таку послідовність: 8421. У загальному вигляді ця послідовність має вигляд:

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

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

і використовується для перекладу двійкового числа в десяткове. Наприклад, двійкове число 101111 еквівалентно десятковому числу 47:

$$2^5 \cdot 1 + 2^4 \cdot 0 + 2^3 \cdot 1 + 2^2 \cdot 1 + 2^1 \cdot 1 + 2^0 \cdot 1 = 47$$

У цифрових пристроях використовуються спеціальні терміни для позначення різних за об'ємом одиниць інформації: біт, байт, кілобайт, мегабайт і так далі

Біт або двійковий розряд визначає значення одного якого-небудь знаку в двійковому числі. Наприклад, двійкове число 101 має три біта або три розряди. Крайній справа розряд, з найменшою вагою, називається молодшим, а крайній зліва, з найбільшою вагою, – старшим.

Байт визначає 8-розрядну одиницю інформації, 1байт = 2^3 біт, наприклад, 10110011 або 01010111 і так далі

$$1\text{кбайт} = 2^{10} \text{біт}, 1\text{Мбайт} = 2^{10} \text{кбайт} = 2^{20} \text{біт}$$

Для представлення багаторозрядних чисел в двійковій системі числення потрібне велике число двійкових розрядів. Запис полегшується, якщо використовувати шістнадцятиричну систему числення.

Підставою шістнадцятиричної системи числення є число $16 = 2^4$. В ній використовується 16 елементів позначення: числа від 0 до 9 і букви А, В, С, D, Е, F. Для перекладу двійкового числа в шістнадцятиричне, досить двійкове чис-

ло розділити на чотирьох – бітові групи: цілу частину з права наліво, дріб – зліва направо від коми. Крайні групи можуть бути неповними.

Кожна двійкова група представляється відповідним шістнадцятиричним символом (табл. 1.1). Наприклад, двійкове число 0101 1100 0011 1001 в шістнадцятиричній системі виражається числом 5С39.

Користувачеві найбільш зручна десяткова система числення. Тому багато цифрових пристроїв, працюючи з двійковими числами, здійснюють прийом і видачу десяткових чисел. При цьому застосовується двійково – десятковий код.

Двійково – десятковий код утворюється заміною кожної десяткової цифри числа чотирьохрозрядним двійковим представленням цієї цифри в двійковому коді (табл. 1.1).

Таблиця 1.1 – Коди чисел від 0 до 15

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

Наприклад, число 15 представляється як 0001 0101 BCD (Binary Coded Decimal). При цьому в кожному байті розташовуються дві десяткові цифри. Ві-

дмітимо, що двійково-десятковий код при такому перетворенні не є двійковим числом, еквівалентним десятковому числу.

Вісімковий код застосовується аналогічно шістнадцятиричному, тільки об'єднуються лише по 3 розряди двійкового числа. Для вісімкового коду застосовуються цифри десяткового коду від 0 до 7, він позначається індексом 8.

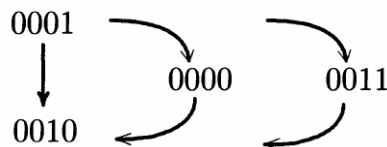
Наприклад:

$$\frac{011|101|001|010}{3|5|1|2}$$

Отже, справедливо $011101001010 (A_2) = 3512 (A_8)$.

Часто в цифровій техніці для числового коду потрібна схема кодування, відповідно до якої при переході від одного числа до наступного змінювалася б лише одна цифра. Виконання цієї умови необхідне, коли внаслідок технічних неточностей момент перемикавання не може бути точно витриманий. При перемиканні двох цифр через це можуть виникнути неправильні комутаційні операції.

Як приклад подібної помилки розглянемо перемикавання від 1 (A_{10}) до 2 (A_{10}) в двійковому коді:



При цьому перемиканні змінюються біти 0 і 1. При одночасному перемиканні безпосередньо досягається нове число. Якщо спочатку змінюється біт 0, то з'являється число 0000, і лише коли змінюється біт 1, отримуємо правильне число 0010. Якщо ж спочатку змінюється біт 1 і потім змінюється біт 0, то в проміжку виникає число 0011. Коди Грея дозволяють уникнути цієї дуже серйозної помилки за рахунок того, що при переході від одного кодового слова до наступного змінюється лише один розряд. У таблиці 1.2 представлений чотирирозрядний код Грея. На додаток до вищесказаного, вказаний код має властивість, яка полягає в його циклічності, оскільки також і при переході від вищого числа 15 (A_{10}) до нижчого числа 0 (A_{10}) змінюється лише один розряд.

Таблиця 1.2 – Код 8-4-2-1 і чотирьохрозрядний код Грея

Десяткове число	Коди	
	Код 8-4-2-1	Код Грея
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Якщо необхідно підвести до десяткових цифр деякі відмітки, то для цього підійде код, в якому окремим десятковим цифрам додані двійково-кодовані кодові слова. Оскільки окремі розряди мають ваги 8, 4, 2 і 1, даний код називають кодом типу 8-4-2-1 (табл. 1.2).

Існує також можливість побудувати двійково-десятковий код 2-4-2-1 (Aiken-Code). До інших двійково-десяткових кодів відносяться код з лишком три (3-Excess-Code) і двійково-десятковий код Грея (BCD-Gray-Code).

1.2 Арифметичні операції в двійковій системі числення

Цілочисельне складання двох чисел A і B генерується в двійковій системі точно так, як і в десятковій системі – по розрядах. Як і там, в кожному розряді мають бути підсумовані обидві двійкові цифри a_n , b_n і перенесення з попереднього розряду P_{n-1} . При складанні виникають (табл. 1.3) нова сума S_n і нове перенесення P_n .

Таблиця 1.3 – 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

У цій таблиці виділеною лінією розділені вхідні і вихідні величини. Наприклад:

$$\begin{array}{cccccccc}
 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & \text{перенесення} \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
 + & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\
 \hline
 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1
 \end{array}$$

Для того, щоб мати малі витрати на апаратну частину (hard ware) комп'ютера, були зроблені зусилля по зведенню до одного алгоритму віднімання і складання. Цього можна досягти, якщо застосовувати двійкові цифри в їх доповнюючій формі. Розрізняють одиничне доповнення (зворотний код, порозрядне доповнення) і подвійне доповнення (точне доповнення). Порозрядне доповнення формується шляхом заміни всіх нулів на одиниці і назад. Отже, порозрядним доповненням (one's complement) 0001 є 1110. Нижче порозрядне доповнення двійкового числа A позначено \bar{A} . Вочевидь, що при представленні n -бітового слова маємо:

$$\bar{A} + A = 2^n - 1. \quad (1.2)$$

Наприклад, при представленні 8-бітового слова маємо:

$$10110011 + 01001100 = 11111111 = 2^8 - 1.$$

Можна перетворити рівняння (1.2), аби отримати формулу для розрахунку порозрядного доповнення:

$$\bar{A} = 2^n - 1 - A. \quad (1.3)$$

Точне доповнення (two's complement) A_d утворюється з порозрядного доповнення \bar{A} шляхом додавання 1:

$$A_d = \bar{A} + 1. \quad (1.4)$$

Отже, з урахуванням (1.2) буде справедливо:

$$A_d = 2^n - 1 - A + 1 = 2^n - A. \quad (1.5)$$

Видно, що в даному представленні міститься «- A», завдяки чому воно зручне для проведення віднімання. Врахуємо також, що 2^n в двійковому представленні маємо $n + 1$ розрядів. Нижче приклад точного доповнення для 10101100:

$$A_d = \bar{A} + 1 = 01010011 + 1 = 01010100.$$

Таким чином, цілочисельне віднімання двох чисел A і B генерується в двійковій системі точно так, як і в десятковій системі – по розрядам. При цьому до зменшуваного додається від'ємник, представлений в точному додатковому коді. Наприклад десяткова операція віднімання $46 - 36 = 10$, у вигляді двійкової арифметичної операції виглядатиме таким чином:

$$A_{10} = 46 \quad A_2 = 101110$$

$$B_{10} = 36 \quad B_2 = 100100 \quad \bar{B}_2 = 011011 \quad B_{2_d} = 011100$$

$$\begin{array}{r} 101110 \\ + 011100 \\ \hline \textcircled{1} 001010 \end{array}$$

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

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

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

Для ділення можна використовувати той же самий алгоритм, що і в десятичній системі. Продемонструємо це на прикладі рівняння:

$$10 (A_{10}) : 2 (A_{10}) = 5 (A_{10})$$

$$\begin{array}{r} \underline{1010} \overline{)0010} \\ \underline{10} \quad 101 \\ \quad \underline{010} \\ \quad \quad \underline{10} \\ \quad \quad \quad 0 \end{array}$$

1.3 Представлення інформації фізичними сигналами

Фізичними аналогами логічних змінних «0» і «1» служать сигнали, здатні приймати два добре помітних стани, наприклад, потенціал низького і високого рівнів, розімкнений і замкнутий стан контакту реле і тому подібне.

У схемах цифрових пристроїв змінні і відповідні ним сигнали змінюються не безперервно, а лише в дискретні моменти, що позначаються цілими ненегативними числами: $0, 1, 2, \dots, i$. Часовий інтервал між двома сусідніми моментами дискретного часу називається тактом. Зазвичай цифрові пристрої містять спеціальний блок, що виробляє синхронізуючі сигнали, що відзначають моменти дискретного часу (кордони тактів). У сучасних цифрових пристроях застосовується потенційний спосіб представлення інформації. Потенційний сигнал зберігає постійний рівень протягом такту, а його значення в перехідні моменти не є визначеним. Слово інформації може бути представлене послідовним або паралельним кодом. При послідовному коді кожен часовий такт призначений для відображення одного розряду коду слова (рис. 1.1). В цьому випадку всі розряди слова фіксуються по черзі одним і тим же елементом і проходять через одну лінію передачі інформації.

При паралельному коді всі розряди двійкового слова представляються в одному часовому такті, фіксуються окремими елементами і проходять через окремі лінії, кожна з яких служить для представлення і передачі лише одного розряду слова.

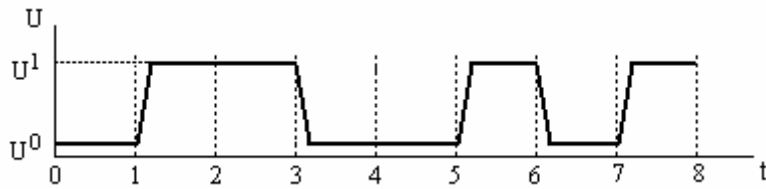


Рисунок 1.1 – Представлення цифрової інформації сигналами потенційного типу (послідовний код)

Код слова розгортається не в часі, а в просторі, оскільки значення всіх розрядів слова передаються по декількох лініях одночасно (рис. 1.2).

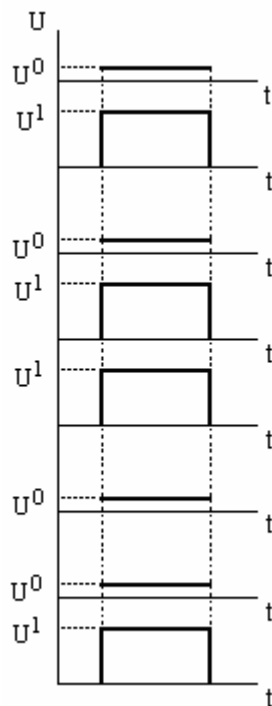


Рисунок 1.2 – Представлення інформації паралельним кодом

1.4 Логічні основи цифрової схемотехніки

Розділ математичної логіки, що вивчає зв'язки між логічними змінними, що мають лише два значення, називається алгеброю логіки. Алгебра логіки розроблена англійським математиком Дж. 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.

У загальному випадку логічні вирази є функціями логічних змінних А, В, С, 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: А = 0 и А = 1; при k = 2: АВ = 00, 01, 10, 11 і т. д. Для кожного набору змінних логічна функція F може набувати значення 0 або 1. Тому для k змінних можна утворити $n_k = 2^k$ різних логічних функцій. Таким чином, при k = 2 можна отримати $n_2 = 16$ функцій і далі при збільшенні k число n_k зростає надзвичайно швидко: $n_3 = 256$, $n_4 = 65536$ і т. д.

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 А,В вхідні змінні.

У таблиці істинності відображаються всі можливі поєднання (комбінації) вхідних змінних і відповідні ним значення функції F, що виходять в результаті виконання якої-небудь логічної операції. При одній логічній змінній повний набір складається з чотирьох функцій, які приведені в таблиці 1.4.

Таблиця 1.4 – Повний набір функцій однієї логічної змінної

А	0 1	Вираз функції	Найменування функції
F ₀	0 0	F ₀ = 0	константа 0
F ₁	0 1	F ₁ = А	тотожність
F ₂	1 0	F ₂ = \overline{A}	інверсія
F ₃	1 1	F ₃ = 1	константа 1

1.4.1 Основні операції булевої алгебри

Інверсія (заперечення, доповнення) є однією з основних логічних функцій, використовуваних в пристроях цифрової обробки інформації.

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 (рис. 1.3).

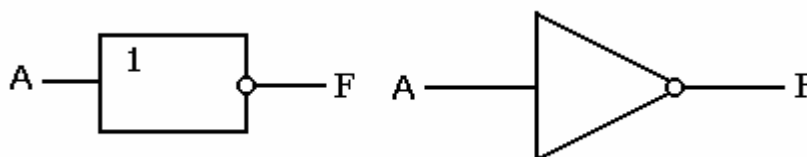


Рисунок 1.3 - Позначення схеми інвертора

Таблиця істинності інвертора

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

Чотири можливі логічні функції представлені в таблиці 1.4. Функції $F_3 = 0$ і $F_4 = 1$ здійснюють постійні, які не залежать від входу, $F_2 = A$. Тому істотною для алгебри логіки є лише однорозрядна двійкова функція $F_1 = \bar{A}$.

Операцію інверсії можна виконати чисто арифметично: $\bar{0} = 1 - 0$, $\bar{1} = 1 - 1$ і алгебраїчно: $\bar{\bar{A}} = A - 1$. З цих виразів випливає, що інверсія A, тобто \bar{A} доповнює A до 1. Звідси і виникла ще одна назва цієї операції – доповнення. Звідси ж можна зробити висновок, що подвійна інверсія призводить до вихідного аргументу, тобто:

$$\bar{\bar{A}} = 1 - \bar{A} = 1 - (1 - A) = A.$$

Це є закон подвійного заперечення.

У релейно-контактній техніці функцію ІІ реалізує нормально замкнутий контакт, тобто такий контакт реле, який замкнутий, поки в обмотці немає струмового сигналу А, і розмикається при подачі струму А (рис. 1.4).

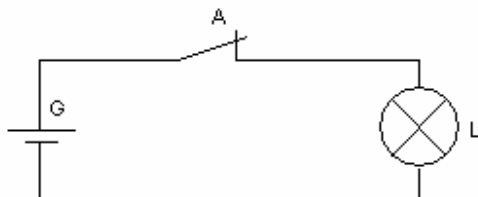


Рисунок 1.4 – Схема інвертора в релейно-контактній техніці

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

Рівняння функції:

$$F = AB; F = A \cdot B; F = A \& B; F = A \wedge B.$$

Читається «F є А і В». Функція І дорівнює 1 тоді і лише тоді, коли всі її аргументи дорівнюють 1.

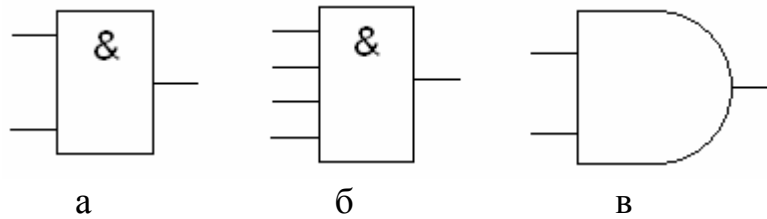
Схему, яка забезпечує виконання такої функції, називають кон'юнктором або схемою І. Таблиця істинності кон'юнктора:

AB	F = A · B
00	0
01	0
10	0
11	1

Елемент І часто використовують для управління потоком інформації. При цьому на один його вхід поступають логічні сигнали, що несуть деяку інформацію, а на іншій – сигнал керування: пропускати – 1, не пропускати – 0. Елемент І, який використовується таким чином, називають вентиль (gate).

Таблиця істинності переконливо показує тотожність операцій звичайного і логічного множень. Тому, як знак логічного множення, можливе використання знаку звичайного множення у вигляді крапки.

Функцію І можна побудувати для будь-якого числа аргументів. На рисунку 1.5 показані умовні зображення двох- і чотирьохходового кон'юнкторів.



а) умовне зображення двохвходового кон'юнктора 2І (AND2); б) умовне зображення чотирьохвходового кон'юнктора 4І (AND4); в) умовне зображення двохвходового кон'юнктора в американській символіці

Рисунок 1.5 – Кон'юнктор

У релейно-контактній техніці функція І реалізується послідовним включенням нормально розімкнених контактів (рис. 1.6).

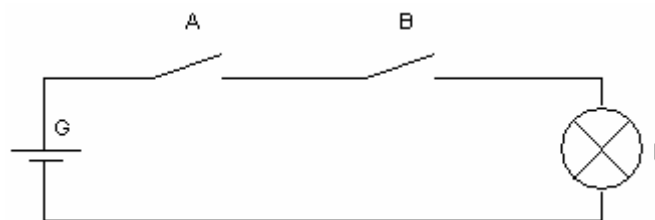


Рисунок 1.6 – Схема кон'юнктора в релейно-контактній техніці

Струм в колі тече, коли обидва контакти замкнуті, тобто знаходяться в одиничному стані.

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

Рівняння функції:

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

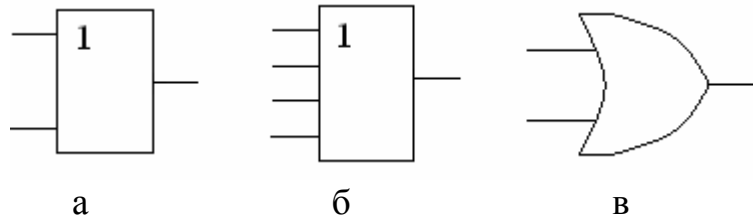
Читається «F є A або B». Функція АБО дорівнює 0 тоді і лише тоді, коли всі її аргументи дорівнюють 0.

Схему, яка забезпечує виконання такої функції, називають диз'юнктором або схемою АБО. Таблиця істинності диз'юнктора:

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

Використовувати знак «плюс» можна в тих випадках, коли диз'юнкцію не можна сплутати з арифметичним підсумовуванням і складанням по модулю 2.

Функцію АБО можна побудувати для будь-якого числа аргументів. На рисунку 1.7 показані умовні зображення двох- і чотирьохходового диз'юнкторів.



а) умовне зображення двохходового диз'юнктора 2АБО (OR2); б) умовне зображення чотирьохходового диз'юнктора 4АБО (OR4); в) умовне зображення двохходового диз'юнктора в американській символіці

Рисунок 1.7 – Диз'юнктор

У релейно-контактних схемах функція АБО реалізується паралельним включенням контактів (рис. 1.8).

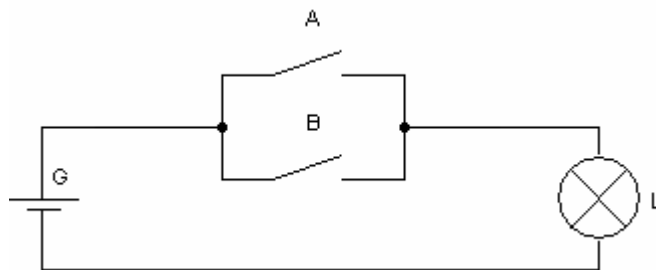


Рисунок 1.8 – Схема диз'юнктора в релейно-контактній техніці

Струм в колі не тече, коли обидва контакти розімкнені, тобто знаходяться в нульовому стані.

Функція І-НІ (NAND) здійснює інвертування отриманого результату операції І. Вона складається з схеми І і інвертора на її виході.

Рівняння функції:

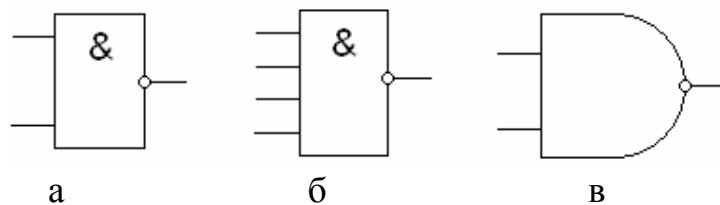
$$F = \overline{AB}; F = \overline{A \cdot B}; F = \overline{A \& B}; F = \overline{A \wedge B}.$$

Читається «F є інверсія A і B». Функція І-НІ дорівнює 0 тоді і лише тоді, коли всі її аргументи дорівнюють 1.

Схему, яка забезпечує виконання такої функції, називають кон'юнктором з інверсією або схемою І-НІ. Таблиця істинності елемента І-НІ:

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

Функцію І-НІ можна побудувати для будь-якого числа аргументів. На рисунку 1.9 показані умовні зображення двох- і чотирьохвходового елементів І-НІ.



а) умовне зображення двохвходового кон'юнктура 2І-НІ (NAND2); б) умовне зображення чотирьохвходового кон'юнктура 4І-НІ (NAND4); в) умовне зображення двохвходового елемента І-НІ в американській символіці

Рисунок 1.9 – Елементи І-НІ

Функція АБО-НІ (NOR) здійснює інвертування отриманого результату операції АБО. Вона складається з схеми АБО і з'єднаного з її виходом інвертора. Рівняння функції:

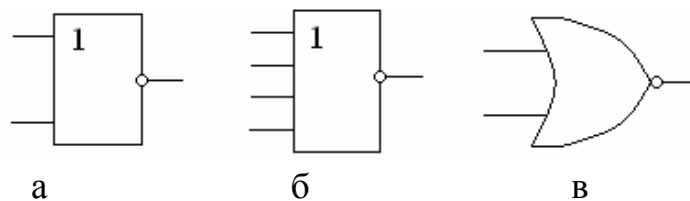
$$F = \overline{A + B} ; F = \overline{A \vee B} .$$

Читається «F є інверсія А або В». Функція АБО-НІ дорівнює 1 тоді і лише тоді, коли всі її аргументи дорівнюють 0.

Схему, яка забезпечує виконання такої функції, називають диз'юнктором з інверсією або схемою АБО-НІ. Таблиця істинності елемента АБО-НІ:

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

Функцію АБО-НІ можна побудувати для будь-якого числа аргументів. На рисунку 1.10 показані умовні зображення двох- і чотирьохвходового елементів АБО-НІ.



а) умовне зображення двохвходового диз'юнктура 2АБО-НІ (NOR2); б) умовне зображення чотирьохвходового диз'юнктура 4АБО-НІ (NOR4); в) умовне зображення двохвходового елемента АБО-НІ в американській символіці

Рисунок 1.10 – Елементи АБО-НІ

Функція «Виключаюче АБО» (XOR). Складання по модулю 2.

Рівняння функції:

$$F = A \oplus B; F = A\bar{B} + \bar{A}B.$$

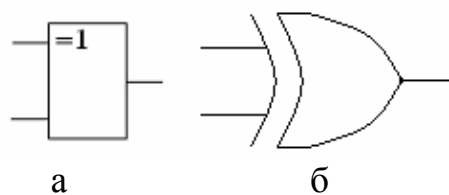
Читається «F є А «Виключаюче АБО» В».

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

Таблиця істинності функції «Виключаюче АБО»:

АВ	F = A ⊕ B
00	0
01	1
10	1
11	0

Ця операція аналогічна операції арифметичного підсумовування, але, як і інші логічні операції, без утворення перенесення. Тому вона має іншу назву складання по модулю 2 і позначення \oplus , схоже з позначенням арифметичного підсумовування. На рисунку 1.11 показані умовні графічні зображення елементів «Виключаюче АБО». Напис на позначенні елемента «Виключаюче АБО» «=1» (рис. 1.11, а) якраз і означає, що виділяється ситуація, коли на входах одна і лише одна одиниця.



а) умовне зображення двухвходового елемента «Виключаюче АБО»; б) умовне зображення двухвходового елемента «Виключаюче АБО» в американській символіці

Рисунок 1.11 – Елементи «Виключаюче АБО»

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

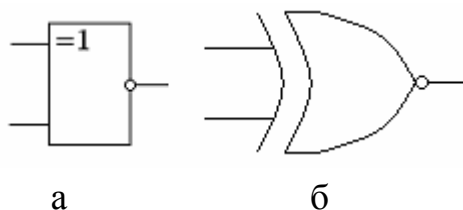
Рівняння функції:

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

Таблиця істинності функції «Виключаюче АБО» - НІ

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

Ця операція використовується при побудові схем в яких на виході завжди має бути логічна 1, якщо на входи подані однакові логічні сигнали – або обидва 0, або обидва 1. На рисунку 1.12 показані умовні графічні зображення елементів. Функція «Виключаюче АБО» - НІ.



а) умовне зображення двухвходового елемента «Виключаюче АБО» - НІ; б) умовне зображення двухвходового елемента «Виключаюче АБО» - НІ в американській символіці

Рисунок 1.12 – Елементи «Виключаюче АБО» - НІ

Правила старшинства логічних операцій.

Інверсія – логічна дія першого рівня.

Кон'юнкція – логічна дія другого рівня.

Диз'юнкція – логічна дія третього рівня.

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

1.4.2 Аксиоми і закони булевої алгебри

Як і інші розділи математики, булева алгебра ґрунтується на ряді постулатів, зокрема на тих, які були висунуті Хантінгтоном в 1904 р.

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

$$0 + 0 = 0 \quad (1.6) \qquad 0 \cdot 0 = 0 \quad (1.10) \qquad \bar{0} = 1 \quad (1.14)$$

$$1 + 0 = 1 \quad (1.7) \qquad 0 \cdot 1 = 0 \quad (1.11) \qquad \bar{1} = 0 \quad (1.15)$$

$$0 + 1 = 1 \quad (1.8) \qquad 1 \cdot 0 = 0 \quad (1.12)$$

$$1 + 1 = 1 \quad (1.9) \qquad 1 \cdot 1 = 1 \quad (1.13)$$

Кожна аксіома (тотожність) представлена в двох видах, що витікає з принципу дуальності логічних операцій, згідно якому операції кон'юнкції і диз'юнкції допускають взаємну заміну, якщо одночасно поміняти 1 на 0, 0 на 1, знак (+) на (\cdot), а знак (\cdot) на (+).

Приведені аксіоми справедливі також для булевих змінних А, В і С. Важливі властивості цих логічних змінних ілюструються наступними попарно об'єднаними законами:

13. Закон тавтології (ідемпотентності):

$$\left\{ \begin{array}{l} A \cdot A = A \quad (1.16) \\ A + A = A \quad (1.17) \end{array} \right.$$

Кон'юнкція однакових співмножників рівносильна одному з них.

Диз'юнкція однакових доданків рівносильна одному з них.

2. Закон нульової множини:

$$\begin{cases} A \cdot 0 = 0 & (1.18) \\ A + 0 = A & (1.19) \end{cases}$$

3. Закон універсальної множини:

$$\begin{cases} A \cdot 1 = A & (1.20) \\ A + 1 = 1 & (1.21) \end{cases}$$

4. Закон додатковості:

$$\begin{cases} A \cdot \bar{A} = 0 & (1.22) \\ A + \bar{A} = 1 & (1.23) \end{cases}$$

(1.22) – закон протиріччя. Жодне припущення не може бути істинним одночасно зі своїм запереченням.

(1.23) – закон виключення третього. Для кожного вислову є лише дві можливості: цей вислів або істинний, або помилковий, третього не дано.

5. Закон подвійної інверсії (подвійного заперечення):

$$\overline{\overline{A}} = A \quad (1.24)$$

Заперечувати заперечення якого-небудь висловлювання те ж саме, що і затверджувати цей вислів (невірно що $2 \times 2 \neq 4$).

6. Закон обертання:

$$\text{якщо } A = B, \text{ то } \bar{A} = \bar{B} \quad (1.25)$$

7. Закон комутативності (переміщення):

$$\begin{cases} A \cdot B = B \cdot A & (1.26) \\ A + B = B + A & (1.27) \end{cases}$$

8. Закон асоціативності (сполучний):

$$\begin{cases} A \cdot B \cdot C = A \cdot (B \cdot C) & (1.28) \\ A + B + C = A + (B + C) & (1.29) \end{cases}$$

9. Закон дистрибутивності (розподільний):

$$\begin{cases} A \cdot (B + C) = (A \cdot B) + (A \cdot C) & (1.30) \\ A + (B \cdot C) = (A + B) \cdot (A + C) & (1.31) \end{cases}$$

10. Закон дуальності (теорема Де Моргана):

$$\left\{ \begin{array}{l} \overline{A \cdot B} = \overline{A} + \overline{B} \quad (1.32) \\ \overline{A + B} = \overline{A} \cdot \overline{B} \quad (1.33) \end{array} \right.$$

Інверсія добутку дорівнює сумі інверсій, інверсія суми дорівнює добутку інверсій.

11. Закон склеювання:

$$\left\{ \begin{array}{l} (A + B) \cdot (A + \overline{B}) = A \quad (1.34) \\ (A \cdot B) + (A \cdot \overline{B}) = A \quad (1.35) \end{array} \right.$$

12. Закон поглинання:

$$\left\{ \begin{array}{l} A \cdot (A + B) = A \quad (1.36) \\ A + (A \cdot B) = A \quad (1.37) \end{array} \right.$$

13. Закон тотожності:

$$A \equiv A \quad (1.38)$$

Думка, втілена в деякому висловлюванні залишається незмінною впродовж всього міркування, в якому цей вислів фігурує (Аристотель).

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 1.5. Кожна функція позначає одну з 16 можливих логічних операцій над двома змінними A, B і має власну назву і умовне позначення. Наприклад, при виконанні операції “Виключаюче АБО” формується сигнал нерівності двох змінних: $F_6 = 1$ при $A \neq B$; $F_6 = 0$ при $A = B$. При виконанні операції “Рівнозначність” формується сигнал рівності змінних: $F_9 = 1$ при $A = B$; $F_9 = 0$ при $A \neq B$. Для складніших функцій: заборона, імплікація, рівнозначність, “Виключаюче АБО”, функції Пірса і Шеффера - приведені їх вирази за допомогою елементарних операцій інверсії, диз’юнкції, кон’юнкції. Вкажемо деякі співвідношення між функціями двох змінних, що використовуються при перетвореннях логічних виразів:

$$F_2 = A \rightarrow B = A\overline{B} = \overline{F_{13}} = \overline{A \rightarrow B} = \overline{\overline{A} + B}; \quad (1.39)$$

$$F_4 = B \rightarrow A = \overline{AB} = \overline{F_{11}} = \overline{B \rightarrow A} = \overline{A + \overline{B}}; \quad (1.40)$$

$$F_6 = A \oplus B = A\overline{B} + \overline{A}B = \overline{F_9} = \overline{B \oplus A} = \overline{AB + \overline{AB}}; \quad (1.41)$$

Справедливість цих співвідношень зазначено з таблиці 1.5, а також легко доводиться за допомогою законів.

Використання аксіом (тотожностей) і законів дозволяє здійснювати спрощення логічних функцій, тобто знаходити для них вирази, що мають найбільш просту форму.

Таблиця 1.5 - Повний набір функцій двох аргументів

A	0 0 1 1	Представлення через операції «І», «АБО», «НІ»	Найменування функції
B	0 1 0 1		
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{AB}	Заборона
F ₅	0 1 0 1	F ₅ = B	Тотожність
F ₆	0 1 1 0	F ₆ = A \overline{B} + $\overline{A}B$	Виключаюче 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–НІ, стрілка Пірса
F ₉	1 0 0 1	F ₉ = AB + \overline{AB}	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 А до В
F ₁₂	1 1 0 0	F ₁₂ = \overline{A}	Інверсія
F ₁₃	1 1 0 1	F ₁₃ = \overline{A} + B	Імплікація від В до А
F ₁₄	1 1 1 0	F ₁₄ = \overline{AB}	І-НІ, штрих Шеффера
F ₁₅	1 1 1 1	F ₁₅ = 1	Константа 1

Закон склеювання:

$$A \cdot B + A \cdot \overline{B} = A \cdot (B + \overline{B}) = A \cdot 1 = A;$$

$$(A + B) \cdot (A + \bar{B}) = A + (B \cdot \bar{B}) = A + 0 = A.$$

Закон поглинання:

$$A \cdot (A + B) = A \cdot A + A \cdot B = A + A \cdot B = A \cdot 1 + AB = A \cdot (1 + B) = A \cdot 1 = A;$$

$$A + (A \cdot B) = A \cdot 1 + A \cdot B = A \cdot (1 + B) = A \cdot 1 = A.$$

Закон дистрибутивності:

$$\begin{aligned} (A + B) \cdot (A + C) &= A \cdot A + A \cdot C + A \cdot B + B \cdot C = A + A \cdot C + A \cdot B + B \cdot C = \\ &= A \cdot 1 + AC + AB + BC = A \cdot (1 + C) + AC + BC = A \cdot 1 + AC + BC = \\ &= A \cdot (1 + C) + BC = A \cdot 1 + BC = A + BC. \end{aligned}$$

Теорема де Моргана корисна при виконанні подвійних перетворень булевих виразів.

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

Спрощення логічних функцій:

$$(A + C)(A + \bar{C})(\bar{B} + C) = (A + C\bar{C})(\bar{B} + C) = (A + 0)(\bar{B} + C) = A \cdot (\bar{B} + C).$$

Схеми початкової і спрощеної функцій представлені на рисунку 1.13.

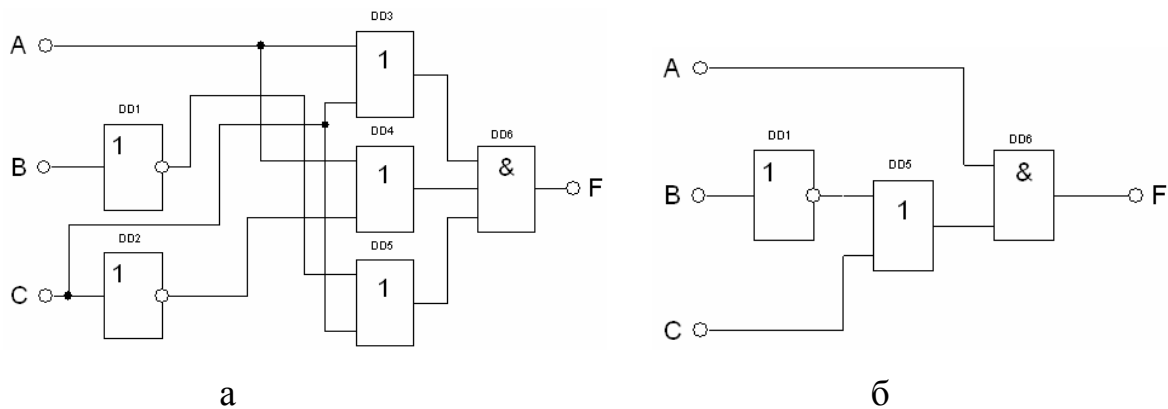


Рисунок 1.13 – Логічна схема реалізації виразу: а - початкова; б – спрощена

1.4.3 Форми представлення логічних функцій

Логічні функції можуть мати різні форми представлення: словесне, табличне, алгебраїчне, графічне. Наприклад, функцію $F_6 = f(A, B)$, задану у вигляді словесного опису: $F_6 = 1$, коли значення змінних $A \neq B$ і $F_6 = 0$, коли $A = B$,

можна представити у вигляді таблиці істинності (табл. 1.6), або в алгебраїчній формі $F_6 = A\bar{B} + \bar{A}B$ (табл. 1.5).

Таблиця істинності містить усі 2^k можливі набори значень логічних змінних і значення функції, які відповідають кожному з наборів.

Таблиця 1.6 - Таблиця істинності функції рівнозначності F_6

A	B	F_6
0	0	0
0	1	1
1	0	1
1	1	0

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 додаються $q = 2^k$ мінтермів: m_0, m_1, \dots, m_{q-1} . Усі мінтерми двох змінних представлені в таблиці 1.7. Значення функції F , які відповідають, згідно таблиці істинності, даному i -му набору змінних, позначимо f_i (табл. 1.7). Як видно з таблиць 1.6, 1.7, представлення алгебраїчної функції F_6 є сумою мінтермів, які відповідають наборам змінних, для яких $f_i = 1$.

Таблиця 1.7 - Мінтерми, макстерми і значення функції F_6

A	B	Мінтерми	Макстерми	Значення функції F_6
0	0	$m_0 = \bar{A}\bar{B}$	$M_0 = A + B$	$f_0 = 0$
0	1	$m_1 = \bar{A}B$	$M_1 = A + \bar{B}$	$f_1 = 1$
1	0	$m_2 = A\bar{B}$	$M_2 = \bar{A} + B$	$f_2 = 1$
1	1	$m_3 = AB$	$M_3 = \bar{A} + \bar{B}$	$f_3 = 0$

У загальному випадку алгебраїчний вираз будь-якої логічної функції можна представити в наступній формі:

$$F_6 = \sum_{i=0}^{n-1} f_i m_i = f_0 m_0 + f_1 m_1 + f_2 m_2 + f_3 m_3 = 0 \cdot (\overline{A}\overline{B}) + 1 \cdot (\overline{A}B) + 1 \cdot (A\overline{B}) + 0 \cdot (AB) = \overline{A}B + A\overline{B},$$

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

Інша форма алгебраїчного представлення функції здійснюється при використанні макстермів. Макстерм (конституента 0) – диз'юнкція всіх змінних, які входять в прямому вигляді, якщо значення даної змінної дорівнює 0, або в інверсному вигляді, якщо, значення змінної дорівнює 1 (таблиця. 1.7). Число макстермів, як і мінтермів, для функції k змінних дорівнює $q = 2^k$. Алгебраїчний вираз функції представлено у вигляді добутку:

$$F_6 = \prod_{i=0}^{n-1} (f_i + M_i) = (f_0 + M_0) \cdot (f_1 + M_1) \cdot (f_2 + M_2) \cdot (f_3 + M_3) = (0 + A + B) \cdot (1 + A + \overline{B}) \cdot (1 + \overline{A} + B) \cdot (0 + \overline{A} + \overline{B}) = (A + B) \cdot (\overline{A} + \overline{B}),$$

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

Часто ДДНФ і ДКНФ не є найбільш простими виразами заданих функцій.

1.5 Основи синтезу цифрових пристроїв

1. Послідовність операцій синтезу 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 добутків, в кожен з яких входять всі незалежні змінні із інверсією або без неї.

Формула здійснюється в два етапи.

а) записується логічна сума добутків, в кожен з яких входять всі незалежні змінні. Кількість доданків дорівнює числу наборів таблиці істинності, на яких логічна функція дорівнює «1».

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

Запис у формі ДКНФ.

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

Як і у попередньому випадку, формула здійснюється в два етапи.

а) записується логічний добуток всіх співмножників. Кількість співмножників дорівнює числу наборів таблиці істинності, на яких логічна функція дорівнює «0».

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

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

Приклад: синтезувати мажоритарний логічний елемент на три входи.

Мажоритарним називається логічний елемент, вихідний стан якого збігається з більшістю вхідних сигналів.

На підставі словесного опису мажоритарного елементу складається його таблиця істинності (табл. 1.8).

Таблиця 1.8 – Таблиця функціонування мажоритарного елементу

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

На основі таблиці істинності записується ДДНФ або ДКНФ функції, а потім складається функціональна схема елементу.

$$\text{ДДНФ: } F = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC.$$

$$\text{ДКНФ: } F = (A + B + C) \cdot (A + \bar{B} + \bar{C}) \cdot (\bar{A} + B + C).$$

Функціональна схема елементу, складена на основі функції ДДНФ мажоритарного елементу, приведена на рисунку 1.14. Схема складається з 8 елементів, що мають загальну кількість входів 19.

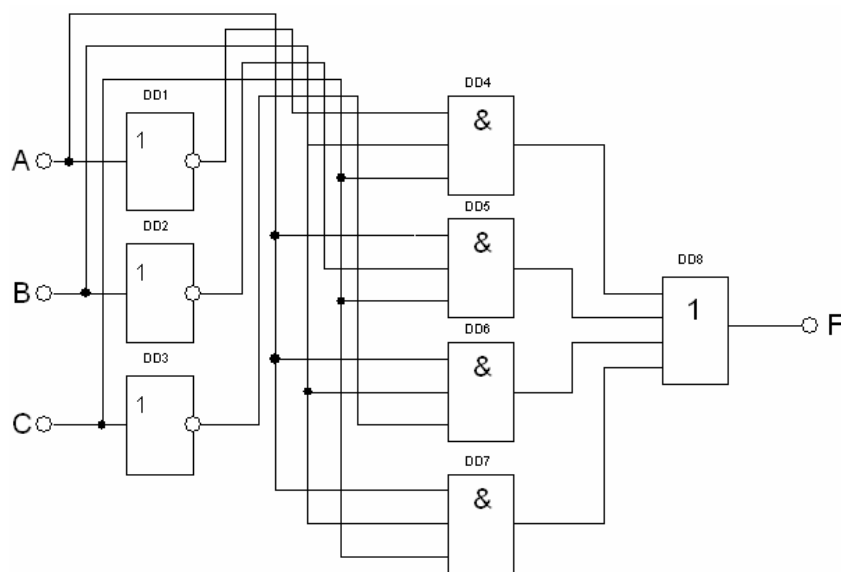


Рисунок 1.14 – Функціональна схема ДДНФ мажоритарного елементу

Кількість входів характеризує складність схеми і називається «Число по Квайну». Схема, яка складена на основі функції ДКНФ, також матиме 19 входів (рис. 1.15).

3. Поняття базису.

Для реалізації логічних пристроїв, призначених для обробки цифрових сигналів в загальному випадку, необхідно мати елементи, які виконують операції І, АБО, НІ. Такий набір елементів називається функціонально повною системою логічних елементів, або логічним базисом. Це означає, що з комбінації логічних елементів І, АБО, НІ, узятих в достатній кількості, можна побудувати скільки завгодно складний цифровий пристрій. Базис з елементів: І, АБО, НІ називається основним.

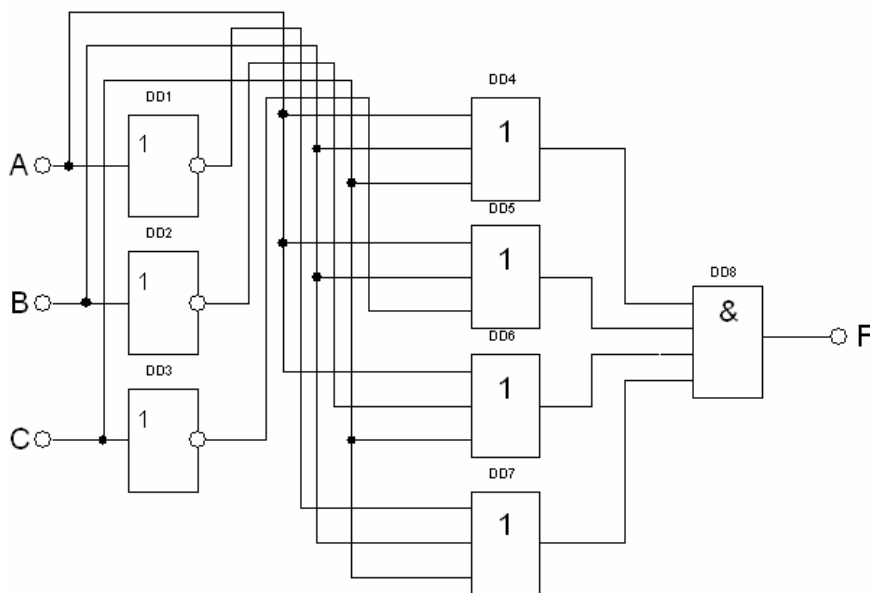


Рисунок 1.15 – Функціональна схема ДКНФ мажоритарного елемента

Число необхідних елементів в такій системі можна зменшити, виключивши з неї або елемент АБО, або елемент І. Наприклад, відповідно до закону подвійного заперечення і за теоремою де Моргана, маємо:

$$A + B = \overline{\overline{A + B}} = \overline{\overline{A} \cdot \overline{B}}; \quad A \cdot B = \overline{\overline{A \cdot B}} = \overline{\overline{A} + \overline{B}}$$

Елемент НІ можна реалізувати на елементах І-НІ, АБО-НІ, об'єднавши всі входи. Звідси випливає, що операцію логічного АБО можна реалізувати на еле-

ментах І-НІ, а операцію логічного І можна реалізувати на елементах АБО-НІ (рис. 1.16).

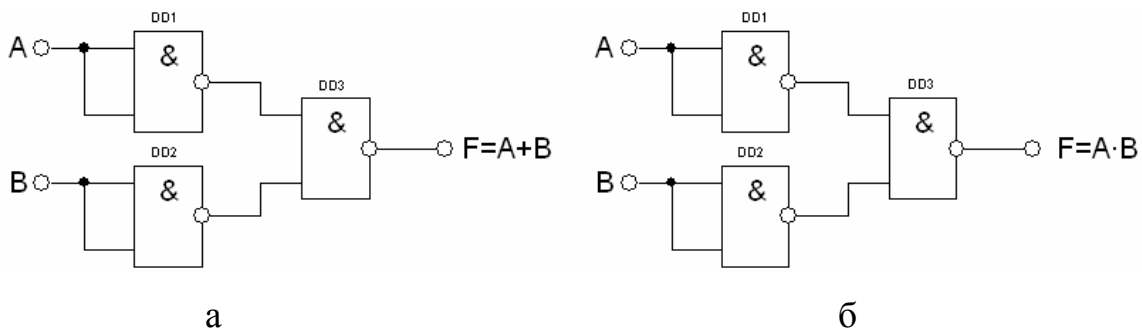


Рисунок 1.16 – Реалізація операцій АБО (а) і І (б) в єдиному елементному базисі

Таким чином, можна побудувати функціональну схему комбінаційного цифрового пристрою в єдиному елементному базисі.

У загальному випадку логічна функція F може залежати від декількох змінних $X_1, X_2, X_3, \dots, X_n$. Говорять, що функція F визначена, якщо відомі її значення для всіх можливих наборів змінних. Функція F не визначена, коли деякі поєднання змінних за умовою завдання неможливі (байдужі набори). Значення функції, які відповідають байдужим наборам можна довизначити, приписавши їм значення «1» або «0» з міркувань зручності реалізації.

1.6 Мінімізація логічних функцій

Структурні формули, отримані у вигляді ДДНФ і ДКНФ, у більшості випадків не сприяють побудові найбільш простих функціональних схем логічних пристроїв. Задану функцію можливо представити у більш компактному вигляді, щоб при її реалізації використовувалась менша кількість елементів. Процес знаходження виразів, які виконують ту ж саму логічну функцію, що і попередньо задана, але для реалізації яких потребується менша кількість елементів, і складає принцип методу мінімізації.

Мінімізація здійснюється з використанням основних співвідношень, законів і теорем алгебри логіки.

1.6.1 Розрахунковий метод мінімізації

Суть методу 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 (закон виключення третього); використання правила склеювання та ін. Алгебраїчна формула, яка отримується в результаті мінімізації, називається тупиковою. Функція може мати декілька тупикових форм.

Приклад: мінімізувати функцію ДДНФ мажоритарного елементу і реалізувати його схему на елементах основного базису.

$$F = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

Склеюючи перші три мінтерми з четвертим, до якого застосовано закон ідемпотентності, отримуємо мінімізовану функцію мажоритарного елементу.

$$F = \bar{A}BC + \underline{A\bar{B}C} + \underline{AB\bar{C}} + ABC + \underline{ABC} + \underline{ABC} = (\bar{A}BC + ABC) + (A\bar{B}C + ABC) + (AB\bar{C} + ABC) = BC \cdot (\bar{A} + A) + AC \cdot (\bar{B} + B) + AB \cdot (\bar{C} + C) = BC \cdot 1 + AC \cdot 1 + AB \cdot 1 = BC + AC + AB.$$

Мінімізована функціональна схема мажоритарного елементу представлена на рисунку 1.17. У мінімізованій схемі число по Квайну зменшилося до 9.

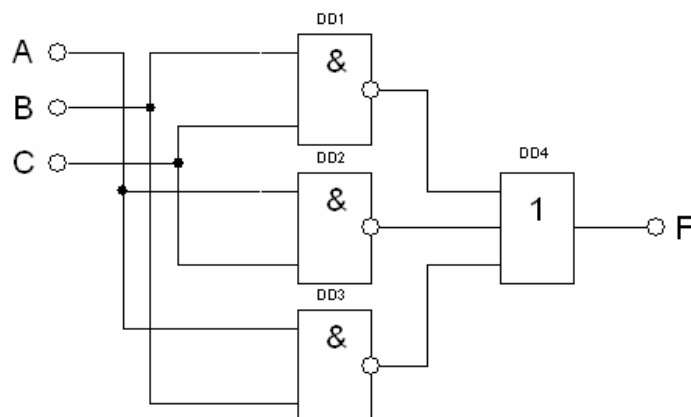


Рисунок 1.17 - Мінімізована функціональна схема мажоритарного елементу

1.6.2 Метод мінімізуючих карт Карно

При відносно невеликому числі змінних ($k \leq 6$) вельми зручним і наочним є графічне представлення логічних функцій у вигляді так званих карт мінтермів. Найбільш розповсюдженою їх формою є карти Карно. На рисунку 1.18 показані карти Карно для функцій $k = 2, 3, 4, 5$ і 6 змінних.

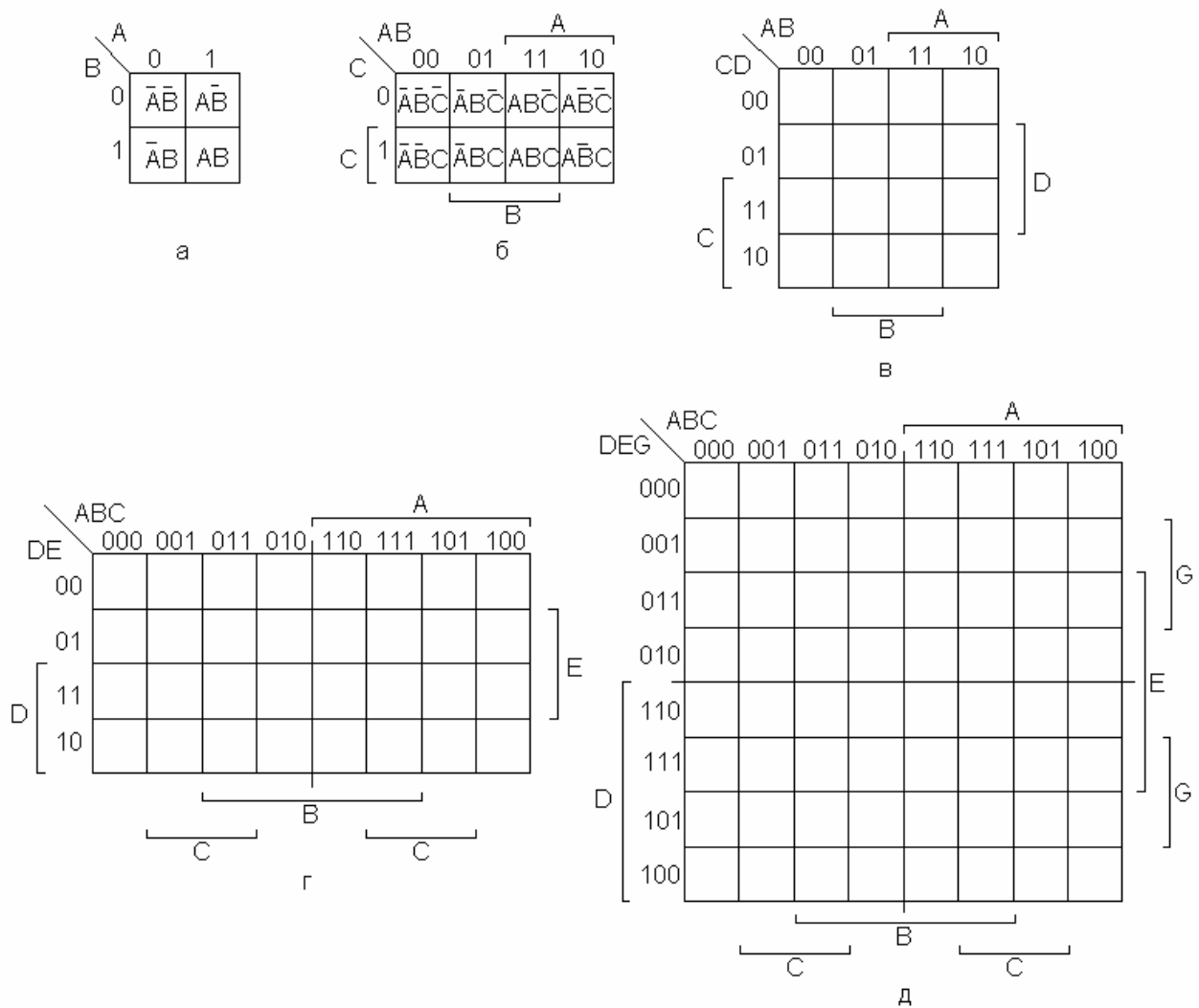


Рисунок 1.18 – Карты Карно для функцій двох (а), трьох (б), чотирьох (в) п'яти (г) і шести (д) змінних

Карта Карно містить $n = 2^k$ клітинок, причому кожній клітинці відповідає один з n мінтермів. Для ілюстрації цього на рисунку 1.18, а, б у клітинках карт Карно записані відповідні їм мінтерми.

Координати рядів і стовпців слідує не в природному порядку зростання двійкових кодів, а в порядку 00, 01, 11, 10. Це код Грея. Зміна порядку до-

тримання наборів зроблена для того, щоб сусідні набори (що відрізняються між собою лише цифрою одного розряду) були сусідніми в геометричному сенсі.

Якщо потрібно представити на карті Карно логічну функцію, задану у вигляді ДДНФ, то в клітинках карти, відповідних мінтермам, які входять в ДДНФ, проставляються 1. Останні клітинки залишаються незаповненими або заповнюються 0. Приклади графічного представлення функцій, заданих у вигляді ДДНФ, показані на рисунку 1.19 а, б.

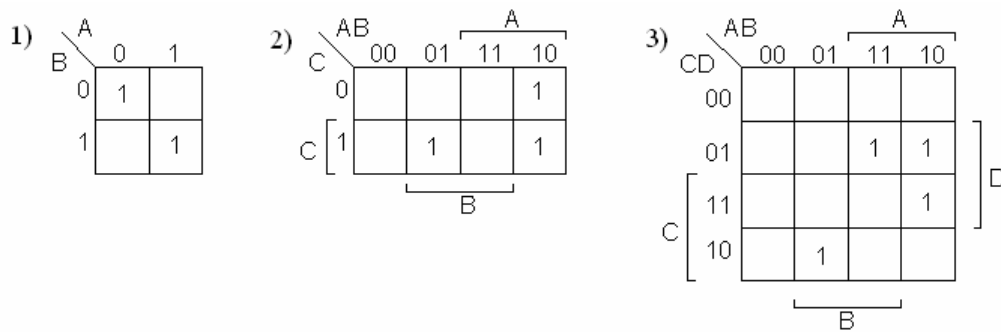


Рисунок 1.19 – Приклади графічного представлення логічних функцій за допомогою карт Карно

Між представленням функції в табличній (таблиця істинності), алгебраїчній (у вигляді ДДНФ) і графічній (на карті Карно) формах є однозначна відповідність.

1) $F = \bar{A}\bar{B} + AB.$

2) $F = A\bar{B}C + A\bar{B}\bar{C} + \bar{A}BC.$

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

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 добуткам (тобто відрізняються лише в одному розряді).

Для логічних функцій з числом змінних $k > 6$ карти Карно стають громіздкими (число клітинок $n > 64$) і не зручними для практичного використання.

Правила групування мінтермів.

1. Групуються дві клітинки, які стоять поряд, в стовпці, або рядку.
2. Групуються клітинки, які є повними квадратами з 4, 16, 64 клітинок.
3. Групуються клітинки, які є повними горизонтальними рядами, або вертикальними стовпцями.
4. Групуються клітинки, які представляють собою два поруч розташованих стовпця, або рядка.
5. Клітинка може входити в декілька об'єднань.
6. Групуються клітинки, які розташовані симетрично відносно вертикальної або горизонтальної вісі карти Карно для п'яти або шести змінних.

Не дивлячись на те, що карти Карно зображаються на площині, сусідство квадратів встановлюється на поверхні тора. Верхній і нижній кордони карти начебто склеюються, утворюючи поверхню циліндра. При склеюванні бічних кордонів виходить поверхня тора.

Приклад: мінімізувати функцію чотирьох змінних, задану таблицею істинності (табл. 1.9).

Таблиця 1.9 - Таблиця істинності заданої функції чотирьох змінних

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

ДДНФ функції:

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

1 2 3 4 5 6 7

Складемо карту Карно (рис. 1.19).

На карті Карно сформовано квадрат і три прямокутники. Квадрат об'єднує 2, 3, 5, 6 мінтерми (доданки), а прямокутники попарно 1, 3 доданки, 2, 4 доданки і 6, 7 доданки.

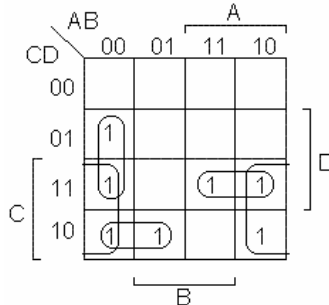


Рисунок 1.19 - Карта Карно заданої функції чотирьох змінних

Мінтерми, об'єднані в квадрат, відрізняються в двох розрядах. Мінтерми, об'єднані в прямокутники, відрізняються лише в одному розряді. Їх незмінна частина, при мінімізації розрахунковим методом виноситься за дужки, і є мінімізованим значенням функції:

$$F = \overline{B}C + \overline{A}CD + \overline{A}BD + ACD.$$

Скоротити роботу по мінімізації інколи можна за рахунок роботи не з самою заданою функцією, а з її інверсією. Якщо число одиниць в таблиці істинності перевищує половину числа комбінацій аргументів, то ДДНФ для інверсії функції міститиме менше кон'юнкцій, чим ДДНФ прямої функції. При апаратній реалізації до виходу схеми потрібно підключити інвертор.

1.7 Запис структурних формул в універсальних базисах

Запис в базисі І-НІ здійснюється в два етапи:

а) логічна формула, мінімізована в основному базисі, представляється у мінімальній диз'юнктивній нормальній формі (МДНФ).

б) над правою частиною отриманої формули ставиться два знаки інверсії і за допомогою теореми де Моргана здійснюється перехід в базис І-НІ.

Приклад: записати в базисі І-НІ мінімізовану функцію мажоритарного логічного елементу:

$$F = BC + AC + AB = \overline{\overline{BC + AC + AB}} = \overline{\overline{BC} \cdot \overline{AC} \cdot \overline{AB}}$$

Запис в базисі АБО-НІ також здійснюється в два етапи:

а) логічна функція, мінімізована в основному базисі, представляється в мінімальній кон'юнктивній нормальній формі МКНФ;

в) над правою частиною отриманої формули проставляються два знаки інверсії, і за допомогою теореми де Моргана здійснюється перехід в базис АБО-НІ.

Приклад:

$$\begin{aligned} F &= (A + \overline{C} + B) \cdot (\overline{B} + C + D) \cdot (A + \overline{B} + C + \overline{D}) = \\ &= \overline{\overline{(A + \overline{C} + B) \cdot (\overline{B} + C + D) \cdot (A + \overline{B} + C + \overline{D})}} = \\ &= \overline{(A + \overline{C} + B) + (\overline{B} + C + D) + (A + \overline{B} + C + \overline{D})} \end{aligned}$$

Запис в базисі І-АБО-НІ здійснюється також в два етапи:

а) логічна формула для інверсного значення функції мінімізується в основному базисі і представляється у формі МДНФ;

б) для переходу до базису І-АБО-НІ над обома частинами формули проставляється один знак інверсії, і за допомогою теореми де Моргана здійснюється перехід в базис І-АБО-НІ.

Приклад:

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

$$\begin{aligned} \overline{F} &= \overline{(\overline{A} + \overline{B} + C)(C + \overline{D})(\overline{A} + \overline{B} + \overline{C} + D)} = \overline{(\overline{A} + \overline{B} + C) + (C + \overline{D}) + (\overline{A} + \overline{B} + \overline{C} + D)} = \\ &= ABC + \overline{C}D + ABC\overline{D} \end{aligned}$$

Мінімізуємо функцію за допомогою карти Карно:

		AB		A	
		00	01	11	10
C	CD	00		1	
	01	1	1	1	1
	11				
	10			1	
				D	
				B	

$$\overline{F} = ABC + \overline{C}D + ABC\overline{D}$$

Остаточний результат представимо в базисі І-АБО-НІ:

$$F = \overline{ABC} + \overline{CD} + \overline{ABD}$$

Контрольні запитання до першого розділу

1. Представити задане десяткове число в двійковому кодї.
2. Представити набїр функцій заданої кількості аргументів.
3. Пояснити операції інверсії і логїчного складання.
4. Пояснити операції інверсії і логїчного множення.
5. Як організована елементна база статичних цифрових мікросхем?
6. Пояснити операцію «Виключаюче АБО».
7. Перелїчити аксіоми булевої алгебри.
8. Перелїчити теореми булевої алгебри.
9. Довести закон поглинання.
10. Виконати спрощення заданої функції методом карт Карно.
11. Привести задану функцію до базису І - НІ.
12. Привести задану функцію до базису АБО - НІ.
13. Привести задану функцію до базису І – АБО - НІ.
14. Привести задану функцію до базису НІ - Монтажне І
15. Привести задану функцію до базису НІ – Монтажне АБО.
16. Побудувати структурну схему заданої функції.

2.1 Система параметрів цифрових мікросхем

До параметрів, що характеризують логічні і схемотехнічні можливості логічних елементів (ЛЕ) мікросхем і великих інтегральних схем, відносяться функціональні параметри:

- логічна функція, яка реалізується;
- здатність навантаження n , що характеризує можливість підключення певного числа ідентичних ЛЕ;
- коефіцієнт об'єднання по входу m (m_I – для реалізації логічної функції І; $m_{АБО}$ – для реалізації логічної функції АБО);
- середня затримка передачі сигналу $\tau_{\text{сер}}$ (напівсума часу затримок передачі сигналів 1 і 0 з входу ЛЕ на його вихід);
- гранична робоча частота f_p (частота перемикання тригера, складеного з ЛЕ, що розглядаються);
- споживана потужність.

По вигляду логічної функції, що реалізовується, ЛЕ умовно можуть бути розбиті на два класи. До першого класу відносяться функціональні елементи одноступінчатої логіки. Це прості ЛЕ, що реалізують функції І, АБО, НІ, І-НІ, АБО-НІ. До другого класу відносяться функціональні елементи двоступінчатої логіки, що реалізують складніші функції: І-АБО, АБО-І, НІ-І-АБО, І-АБО-НІ, І-АБО-І та ін.

Навантажена здатність ЛЕ n визначає число входів ідентичних елементів, які можуть бути підключені до виходу. При цьому забезпечується неспотворена передача двійкових символів 0 і 1 в цифровому пристрої по колу з довільного числа послідовно включених елементів при найгірших поєднаннях дестабілізуючих чинників. Дестабілізуючими чинниками можуть бути зміна напруг живлення, розкид параметрів компонентів, зміна температури і тому подібне.

Часто навантажена здатність n називається коефіцієнтом розгалуження елементів по виходу і виражається цілим позитивним числом ($n = 4, 5, 7, 10$ і т. д.). Чим вище навантажена здатність ЛЕ, тим ширше їх логічні можливості і тим менше витрати при побудові цифрового пристрою. Проте, збільшення параметра n можливо до певних меж, оскільки при цьому погіршуються інші параметри ЛЕ: знижується швидкодія, погіршується завадостійкість і збільшується споживна потужність. З цієї причини до складу однієї серії ІМС часто входять ЛЕ з різною навантаженою здатністю: основні ЛЕ з $n = 4 \dots 10$ і буферні елементи – так звані підсилювачі потужності з $n = 20 \dots 50$. Це дозволяє гнучкіше проектувати цифрові пристрої, досягаючи оптимальних показників по споживаній потужності і числу ЛЕ.

Буферні елементи, як правило, виконуються із складним транзисторним виходом, що забезпечує комутацію великих струмів навантаження. Залежно від електричної схеми базові функціональні елементи в основному працюють в режимі, коли входи навантажень споживають струм з виходу елемента, або коли вихід елемента забирає струм від навантажених входів подальших елементів. Існують ЛЕ, які в одному логічному стані віддають струм в навантаження, а в іншому отримують його від навантаження.

Функціональні елементи інтегральних МДН-мікросхем, що мають високий вхідний опір ($K_{вх} > 10^{12}$ Ом), у статичному режимі практично не віддають струм в навантаження і не забирають його з навантаження. Їх навантажена здатність висока ($n > 10$), і її збільшення обмежується лише погіршенням динамічних параметрів ІМС за рахунок зростання постійних часу заряду і розряду паразитної ємності навантаження C_n при великих n , так як $C = C_{вх}n + C_{вих}$, де $C_{вх}$ – ємність входу одного ЛЕ; $C_{вих}$ – ємність виходу ЛЕ, включаючи лінію зв'язку.

Залежно від частотного діапазону роботи логічних МДН-мікросхем їх навантажена здатність може змінюватися в широких межах ($n = 10 \dots 100$).

Коефіцієнт об'єднання по входу m характеризує максимальне число логічних входів функціонального елемента. Із збільшенням параметра m розши-

рюються логічні можливості мікросхеми за рахунок виконання функцій з великим числом аргументів на одному типовому елементі І-НІ, АБО-НІ і т. п. Проте, при збільшенні числа входів, як правило, погіршуються інші параметри функціонального елемента, такі як швидкодія, завадостійкість і навантажена здатність.

Статичні параметри:

- вхідні і вихідні напруги логічного 0 і 1 ($U_{\text{вих}}^0, U_{\text{вих}}^1, U_{\text{вх}}^0, U_{\text{вх}}^1$);
- вхідні і вихідна порогові напруги логічного 0 і 1 ($U_{\text{вих. пор}}^0, U_{\text{вих. пор}}^1, U_{\text{вх. пор}}^0, U_{\text{вх. пор}}^1$);
- вхідні і вихідні струми логічного 0 і 1 ($I_{\text{вих}}^0, I_{\text{вих}}^1, I_{\text{вх}}^0, I_{\text{вх}}^1$);
- струми споживання в стані логічного 0 і 1 ($I_{\text{спож}}^0, I_{\text{спож}}^1$);
- потужність, що споживається ЛЕ від джерел живлення

$$P_{\text{спож}} = \sum_{i=1}^n U_i I_i,$$

де U_i – напруга і-го джерела живлення; I_i – струм у відповідному колі живлення.

Завадостійкість логічних елементів. Основною статичною характеристикою ЛЕ є передавальна характеристика $U_{\text{вих}} = f(U_{\text{вх}})$: залежність потенціалу на виході від потенціалу на одному з входів, при постійних значеннях потенціалу (U^0 або U^1) на останніх входах. За типом передавальної характеристики елементи діляться на: 1) інвертуючі - на виході яких утворюється інверсія вхідних сигналів; 2) неінвертуючі.

Передавальна характеристика (рис. 2.1) має три явно виражені ділянки.

I – стан $U_{\text{вих}} = U^0$;

II – стан $U_{\text{вих}} = U^1$;

III – проміжний стан.

Значення потенціалу $U_{\text{вх}}$, яке відповідає кордонам ділянок, називають порогоми перемикачів $U_{\text{п}}^0$ і $U_{\text{п}}^1$. Область між порогоми – зоною невизначеності. При послідовному з'єднанні декількох логічних елементів їх загальна передавальна характеристика матиме різкіше розділення ділянок I і II.

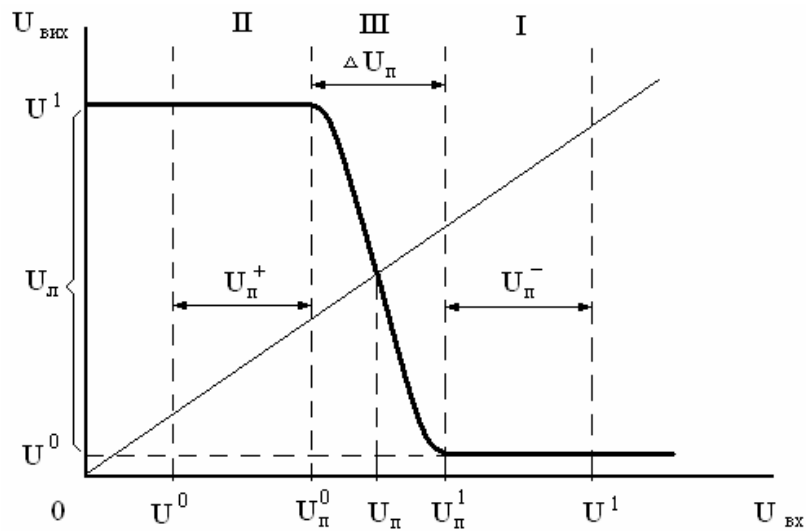


Рисунок 2.1 - Передавальна характеристика інвертуючого ЛЕ

Якщо на входах елементів послідовного кола встановлені логічні рівні U^0 і U^1 , то під час вступу позитивної перешкоди величиною $U_{\Pi}^+ > U_{\Pi}^0 - U^0$ і негативної перешкоди $U_{\Pi}^- > U^1 - U_{\Pi}^1$ відбувається перемикання, яке не передбачене нормальним логічним функціонуванням.

Окрім корисних (управляючих) сигналів на логічні елементи завжди впливають сигнали паразитні, зумовлені зовнішніми електромагнітними перешкодами (наведеннями), або внутрішніми процесами (зв'язками через загальне джерело живлення). Тому, робочі сигнали повинні перевищувати рівень перешкод, а до малих паразитних сигналів логічні елементи мають бути по можливості нечутливі. Здатність логічних елементів протистояти паразитним сигналам називають завадостійкістю. Максимально допустима величина потенційної перешкоди, що не викликає збоїв в цифровій схемі визначається виразом:

$$\begin{aligned}
 U_{\Pi}^+ &= U_{\Pi}^0 - U^0; & U_{\Pi}^- &= U^1 - U_{\Pi}^1 \\
 U_{\Pi}^+ + U_{\Pi}^- &= (U_{\Pi}^0 - U^0) + (U^1 - U_{\Pi}^1) = \\
 &= U_{\Pi}^0 - U^0 + U^1 - U_{\Pi}^1 = \\
 &= (U^1 - U^0) - (U_{\Pi}^1 - U_{\Pi}^0) = \\
 &= U_{\Pi} - \Delta U_{\Pi},
 \end{aligned}$$

де $U_{\Pi} = U^1 - U^0$ – логічний перепад; $\Delta U_{\Pi} = U_{\Pi}^1 - U_{\Pi}^0$ – ширина зони невідзначеності.

Таким чином, для підвищення завадостійкості слід збільшувати $U_{л}$ і зменшувати $\Delta U_{п}$.

Швидкодія ЛЕ при перемиканні визначається електричною схемою, технологією виготовлення і характером навантаження.

До основних динамічних параметрів логічного елементу відносяться: t_{ϕ}^{01} – фронт формування рівня логічної 1, t_{ϕ}^{10} – фронт формування рівня логічного 0, $t_{затр}^{10}$ – затримка перемикання із стану 1 в стан 0, $t_{затр}^{01}$ – затримка перемикання із стану 0 в стан 1, t_i – тривалість імпульсу, f_p – робоча частота. Визначення цих параметрів забезпечується при порівнянні сигналів на вході і виході ЛЕ, тобто при розгляді процесу передачі інформації через ЛЕ. На рисунку 2.2 приведені характеристики сигналів на вході і виході інвертора і показані рівні відліку (0,1 і 0,9 від логічного перепаду $U_{л}$), відносно яких визначаються динамічні параметри ЛЕ.

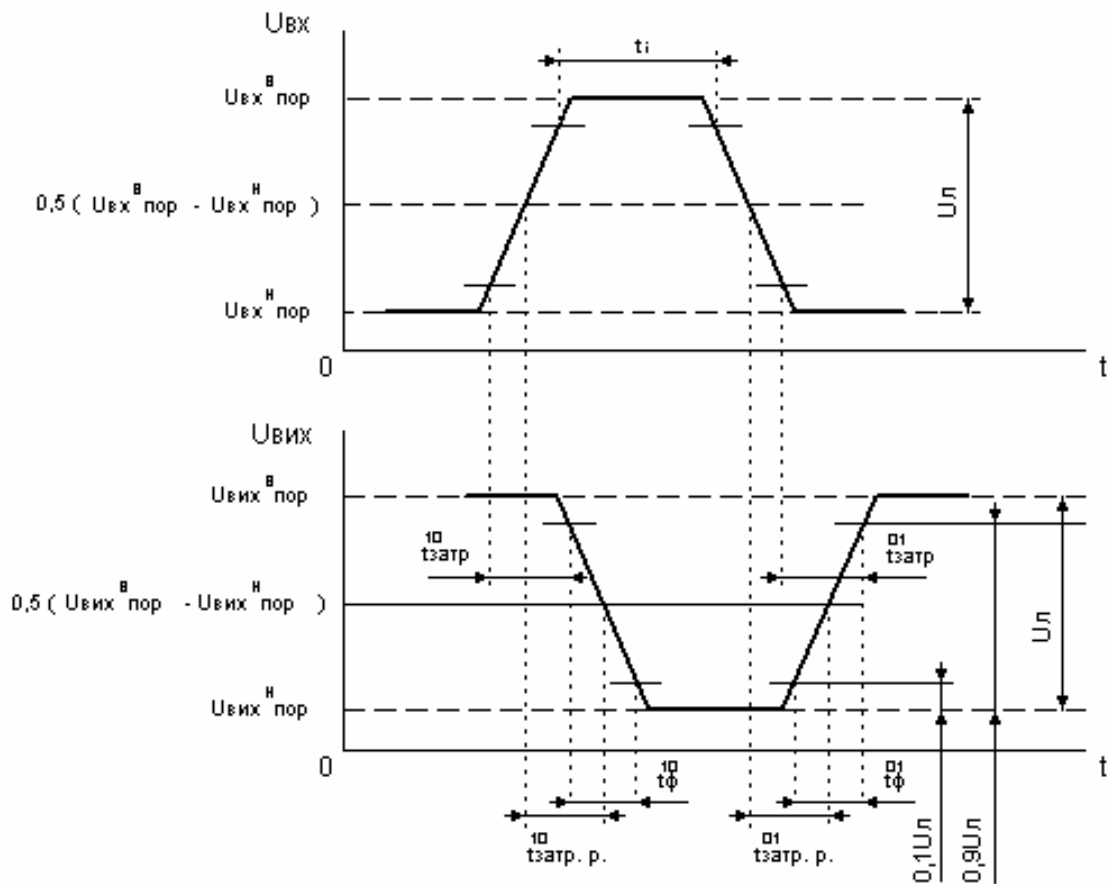


Рисунок 2.2 – Динамічні параметри логічного елементу

Рівнями відліку динамічних параметрів ЛЕ є (розглядається позитивна логіка) максимальні рівень логічного 0 ($U_{\text{вх пор}}^{\text{Н}}$, $U_{\text{вих пор}}^{\text{Н}}$) і мінімальний рівень логічної 1 ($U_{\text{вх пор}}^{\text{В}}$, $U_{\text{вих пор}}^{\text{В}}$). Затримка переключення $t_{\text{затр}}^{10}$ визначається як часовий інтервал між рівнем 1 фронту наростання вхідного імпульсу (позитивний імпульс) і рівнем 0 фронту спаду вихідного імпульсу (негативний імпульс). Затримка переключення $t_{\text{затр}}^{01}$ визначається як часовий інтервал між рівнем 0 фронту спаду вхідного імпульсу і рівнем 1 фронту наростання вихідного імпульсу. Фронти імпульсу, що визначаються між рівнями 1 і 0 спаду імпульсу, позначаються $t_{\text{ф}}^{10}$, між рівнями 0 і 1 наростання імпульсу – $t_{\text{ф}}^{01}$.

Затримки розповсюдження ($t_{\text{затр р}}^{10}$, $t_{\text{затр р}}^{01}$) вимірюються, як правило, по рівню $0,5(U_{\text{вих пор}}^{\text{В}} - U_{\text{вих пор}}^{\text{Н}})$.

Середня затримка $t_{\text{затр. р. сер}}$ логічного елемента визначається як напівсума затримок $t_{\text{затр р}}^{10}$ і $t_{\text{затр р}}^{01}$ та служить усередненим параметром швидкодії, що використовується при розрахунку часових характеристик багатоеlementних послідовно включених логічних мікросхем.

$$t_{\text{затр. р. сер.}} = (t_{\text{затр р}}^{10} + t_{\text{затр р}}^{01}) / 2$$

Параметр $t_{\text{затр. р. сер.}}$ наводиться в технічних умовах або в інструкції по використанню інтегральних мікросхем. Для спрощення процесу розрахунку часових характеристик складних логічних кіл часто вважають сигнали прямокутними, тобто $t_{\text{ф}}^{10} = t_{\text{ф}}^{01} = 0$.

Інтегральними параметрами відображають рівень розвитку технології, схемотехніки і якості цифрових ІС. Основними інтегральними параметрами ІС є енергія перемикавання P_t і рівень інтеграції N .

Енергія перемикавання: $P_t = P_{\text{спож. сер.}} \times t_{\text{затр. р. сер.}}$

По цьому параметру в даний час здійснюють оцінку рівня розвитку цифрової мікроелектроніки і порівняння різних типів ІС.

Міра інтеграції N логічних ІС визначається числом простих еквівалентних ЛЕ (зазвичай двоохводових вентилів) на кристалі. Функціонально склад-

ність ІС пристроїв, які є запам'ятовуваними і мають розмірну структуру, можна оцінювати числом біт пам'яті на кристалі.

2.2 Логічний інвертор на біполярному транзисторі

Логіка роботи логічного елемента НЕ (інвертора) представлена в таблиці істинності, на основі якої отримують вираження для вихідної булевої функції $F = \bar{X}$. Схема елемента НІ, його умовне графічне зображення і часові діаграми роботи показані на рисунку 2.3.

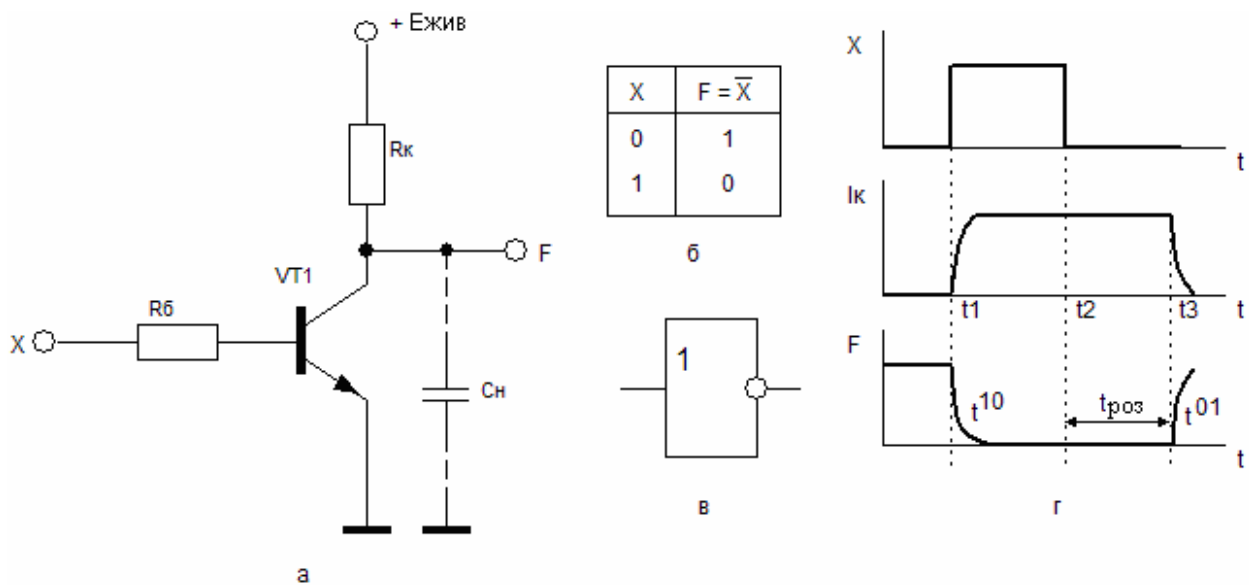


Рисунок 2.3 – Елемент НІ

Схема елемента НІ включає: VT1 – кремнієвий транзистор n-p-n типу; резистори в колі колекторного навантаження R_k і бази R_b ; і $E_{жив}$ – джерело живлення. Таку схему часто називають транзисторним ключем.

Транзистор VT1 може знаходитися в трьох основних режимах: відсічення колекторного струму (закритий стан), насичення (відкритий стан) і активної роботи (посилення).

У режимі відсічення колекторний і базовий переходи закриті (на вхід поданий низький рівень напруги U^0), у колі колектора протікає дуже малий зворотний струм колекторного переходу $I_{к0} < 1 \text{ мкА}$ і на колекторі транзистора встановлюється високий рівень напруги, який визначається виразом

$$U_{\text{вих}} = E_{\text{жив}} - I_{\text{к}} R_{\text{к}}$$

У режимі насичення (на вхід поданий високий рівень напруги U^1) на кремнієвому переході база-емітер пряме падіння напруги дорівнює $U^* = 0,7 \text{ В}$, через колектор протікає максимально можливий струм, який називається струмом насичення колектора $I_{\text{кн}}$. Значення цього струму знаходять із співвідношення

$$I_{\text{кн}} = \frac{(E_{\text{жив}} - U_{\text{кн}})}{R_{\text{к}}},$$

де $U_{\text{кн}} = U^0 \leq 0,4 \text{ В}$ напруга на колекторі насиченого транзистора. Наприклад, для типових значень $E_{\text{жив}} = 5 \text{ В}$, $R_{\text{к}} = 1 \text{ кОм}$ отримуємо для струму $I_{\text{кн}} \approx 5 \text{ мА}$. Для здобуття цього струму потрібний мінімальний базовий струм насичення $I_{\text{бн}}$, значення якого розраховують із співвідношення

$$I_{\text{бн}} = \frac{I_{\text{кн}}}{\beta},$$

де β – коефіцієнт передачі базового струму в транзисторі, включеному за схемою із загальним емітером. Для забезпечення надійного насичення транзистора значення базового струму насичення розраховують за допомогою виразу

$$I_{\text{б}} = S \cdot I_{\text{бн}},$$

де $S = 3 \dots 10$ – міра насичення. Наприклад, якщо $\beta = 50$, то $I_{\text{бн}} = 0,1 \text{ мА}$ і прямий струм бази, що включає транзистор, $I_{\text{б}} = 0,5 \text{ мА}$ для $S = 5$.

Значення опору резистора $R_{\text{б}}$ отримують з умови забезпечення необхідного значення включаючого струму $I_{\text{б}}$:

$$R_{\text{б}} = \frac{U^1 - U^*}{I_{\text{б}}}.$$

В режимі насичення в базі транзистора накопичується надлишковий заряд, значення якого пропорційно до міри насичення. При подачі низького рівня вхідного сигналу транзистор закривається. Проте колекторний струм залишається постійним в інтервалі часу $t_2 - t_3$, який називається часом розсмоктування $t_{\text{роз}}$ надлишкового заряду в базі (рис. 2.3, в). Після закінчення розсмоктування колекторний струм спадає і формується фронт вихідного сигналу t_{01} . Таким чи-

ном, наявність насичення викликає затримку виключення інвертора, що є недоліком даної схеми.

Зменшення тривалості перехідних процесів забезпечується схемою елементу НІ (рис. 2.4, а), у якій замість резистора R_b включені два послідовно включених діода VD1 і VD2.

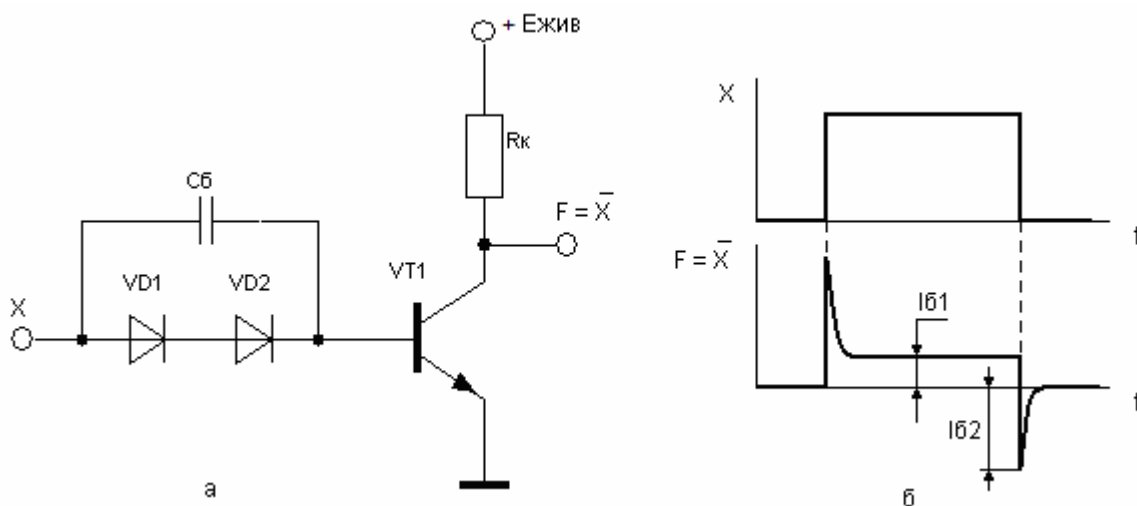


Рисунок 2.4 - Елемент НІ з діодами в колі бази: а – схема; б – часові діаграми базових струмів

Діоди VD1 і VD2 зміщують пороговий рівень вхідної відкриваючої напруги у більшу сторону на значення $2U^*$. Ємність C_b називається прискорюючою (форсуючою), оскільки вона при включенні швидко заряджається, збільшуючи при цьому прямий струм бази $I_{b1} > I_{bn}$, а при виключенні швидко розряджається, створюючи зворотний, вимикаючий струм з амплітудою I_{b2} . Цей струм прискорює розсмоктування надлишкового заряду в базі транзистора.

2.2 Логічний інвертор на МОН транзисторах

МОН-транзистори мають структуру: метал-діелектрик-напівпровідник і в загальному випадку називаються МДН-транзисторами (рис. 2.5). Металевий електрод, на який поступає керуюча напруга, називається затвором (З) а два інших електрода витоком (В) і стоком (С). Від витоку до стоку протікає робочий струм. Для р-каналу полярність стоку негативна, а для n-каналу — позити-

вна. Основна пластина напівпровідника називається підкладкою (П). Канал - це приповерхневий провідний шар між витоком і стоком, в якому величина струму визначається за допомогою електричного поля.

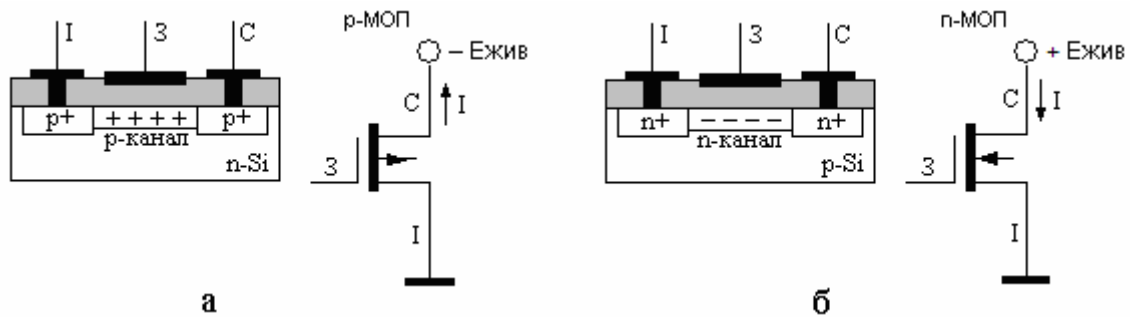


Рисунок 2.5 – Топологія і умовне графічне позначення МОН-транзисторів: а, б – р-канальний; в, г – n-канальний

При нульовому значенні напруги керування, канал відсутній і струм не протікає. Канал, який утворюється під дією зовнішньої керуючої напруги, називається індукованим. Напруга, при якій утворюється канал, називається пороговою U_0 . Канал з початковою додатковою концентрацією зарядів називається вбудованим.

Швидкодія n-МОН транзисторів в 5-8 разів вище за швидкодію р-МОН транзисторів, оскільки рухливість електронів істотно більше дірок. У МОН-схемах повністю виключені резистори, їх роль виконують МОН-транзистори.

Схеми логічних елементів НІ на МОН-транзисторах показані на рисунку 2.6.

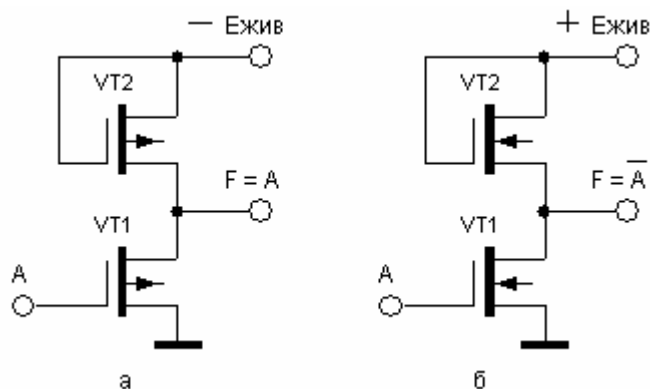


Рисунок 2.6 – Схеми елементів НІ на МОН-транзисторах: а – з р-каналами; б – з n-каналами

У схемі елементу НІ на р-МОН транзисторах у якості навантаження застосовують транзистор VT1, стік якого підключається до негативного джерела живлення - мінус $E_{жив}$ (рис. 2.6, а). Напряга негативної полярності вхідною змінною А поступає на затвор вхідного транзистора VT2. У цій схемі застосовують транзистори з індукованими каналами.

У схемі елементу НІ на n-МОН транзисторах у якості навантаження використовують транзистор VT1 із вбудованим каналом, який підключається до позитивного джерела живлення - плюс $E_{жив}$. Позитивна напруга вхідної змінної А поступає на затвор вхідного транзистора VT2 з індукованим каналом (рис. 2.6, б). Транзистори навантажень включені за схемою двополюсника.

Якщо вхідна напруга $U^0 < U_0$, то транзистор VT2 закритий, а VT1 – відкритий і на виході встановлюється рівень напруги, близький до значення $E_{жив}$. Якщо вхідна напруга $U^1 > U_0$ те обидва транзистори відкриті і вихідна напруга знімається з ділянки, утвореного опором каналів

$$U^0 = \frac{(E_{жив} \cdot R_в)}{(R_в + R_н)}$$

де $R_в$ и $R_н$ – опори каналів вхідного і навантаження транзисторів. Порогові напруги для р-МОН $U^0 = - (5...7)$ В, а для n-МОН $U^0 = + (1,5...2)$ В.

У комплементарній МОН-структурі (логіка КМОН) використовуються одночасно р- і n-канальні транзистори. Елемент НІ в схемотехніці КМОН побудований на двох транзисторах з індукованими каналами: навантажений VT2 з каналом р-типу і вхідному VT2 з каналом n-типу (рис. 2. 7).

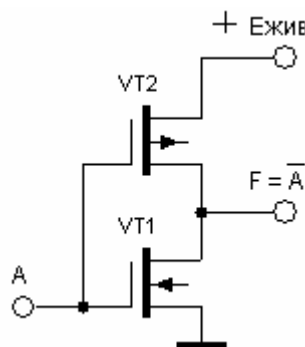


Рисунок 2.7 – Комплементарний МОН інвертор

Витік транзистора VT1 підключений до джерела позитивного живлення $E_{жив}$, напруга вхідної змінної A поступає на затвори обох транзисторів; вихідна напруга знімається з об'єднаних стоків.

При вхідному рівні U^1 транзистор VT2 відкритий, а VT1 – закритий, оскільки між його затвором і витоком є нульова напруга. На виході встановлюється рівень U^0 і струм в колі не протікає. При вхідному рівні U^0 транзистор VT2 закритий, а VT1 – відкритий, тому що між його затвором і витоком є напруга $E_{жив}$. На виході – рівень U^1 і струм в колі не протікає.

2.3 Транзисторна логіка з безпосередніми зв'язками (ТЛБЗ)

У схемі елементу ТЛБЗ опір навантаження включений в коло сполучених колекторів двох транзисторів (рис. 2.8, а). Вхідні сигнали A і B подаються на бази цих транзисторів. Якщо A і B одночасно дорівнюють U^0 , то обидва транзистора закриті і на виході схеми буде високий потенціал U^1 . Якщо хоч би на один, або на обидва входи, подати високий потенціал U^1 , то один або обидва транзистори відкрито і на виході схеми буде низький потенціал U^0 . Таким чином, схема виконує операцію АБО-НІ.

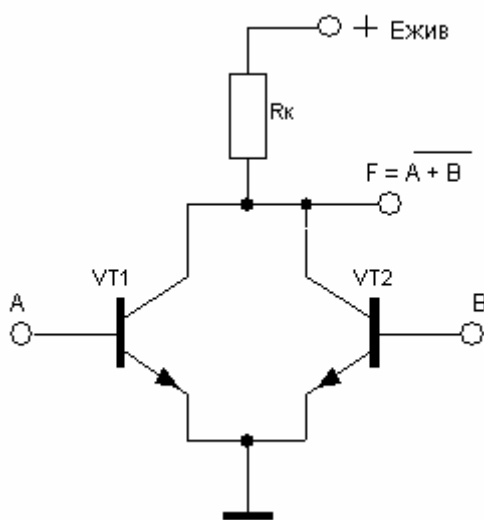


Рисунок 2.8 – Логічний елемент ТЛБЗ

Як видно, схема елементу ТЛБЗ проста, проте у неї є істотний недолік. Коли на виході елементу встановлений потенціал U^1 , на бази транзисторів на-

вантаження подається постійний потенціал U^1 . Із-за розкиду параметрів струми баз транзисторів можуть істотно розрізнятися. В результаті один з транзисторів може увійти до глибокого насичення, а інший – знаходитися в лінійному режимі. При цьому рівні U^1 істотно розрізнятимуться, що незмінно приведе до збоїв в роботі пристрою в цілому. Тому схема ЛЕ ТЛБЗ застосовується лише на транзисторах, керованих напругою.

2.4 Елементи діодно-транзисторної логіки (ДТЛ)

У цих елементах операції диз'юнкції і кон'юнкції реалізуються за допомогою діодних схем, а операцію заперечення виконує інвертор на основі транзисторного ключа. Логіка роботи двохвходового елемента АБО-НІ представлена в таблиці 2.4, на основі якої отримують вираз для булевої функції:

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

Таблиця 2.1 – Логіка роботи двохвходового елемента АБО-НІ

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

Використовуючи принцип суперпозиції, функцію F можна записати для довільного числа змінних:

$$F = \overline{A + B + C + \dots + n}$$

Схема ДТЛ – елемента, яка реалізує булеву операцію АБО-НІ для двох змінних і його умовне графічне позначення (УГП) показані на рисунку 2.9. Схему будують шляхом підключення виходу діодного елемента АБО до входу інвертора. Діоди зсуву VD3 і VD4 грають ту ж саму роль, що і в схемі, представленій на рисунку 2.4, а. Через резистор R1 протікає струм I_{R1} . Транзистор VT1

відкривається і формує низький рівень вихідної напруги U^0 лише у тому випадку, коли на одному або обох входах є високий рівень напруги U^1 .

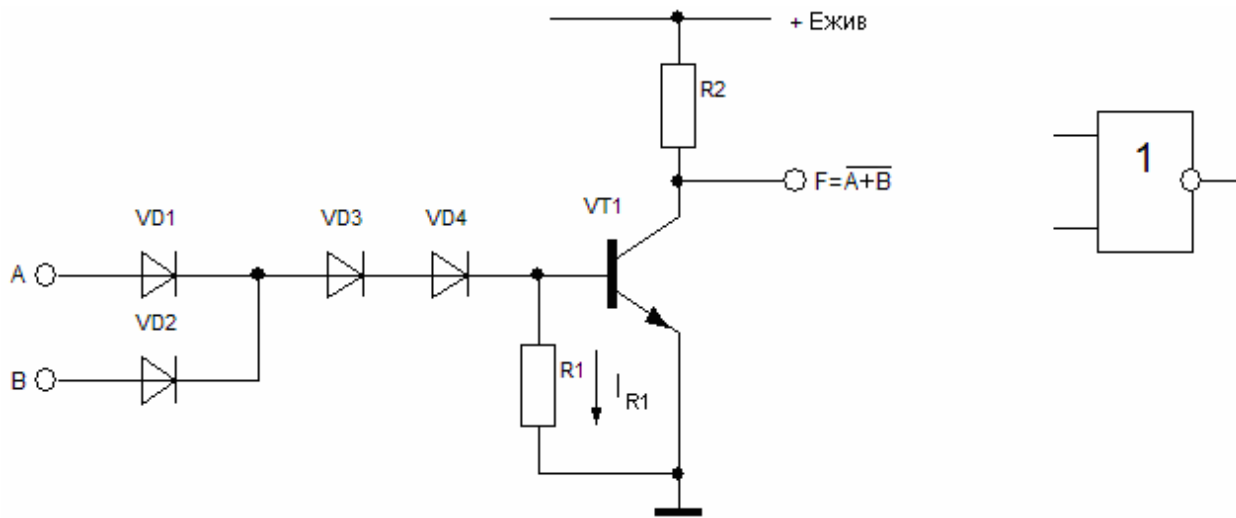


Рисунок 2.9 - Схема ДТЛ – елементу АБО-НІ

При цьому включаючий струм I_6 проходить від джерел вхідних сигналів. Часові діаграми роботи (ідеалізовані) елементу АБО-НІ показані на рисунку 2.10. Тривалість перехідних процесів практично визначається швидкодією транзистора.

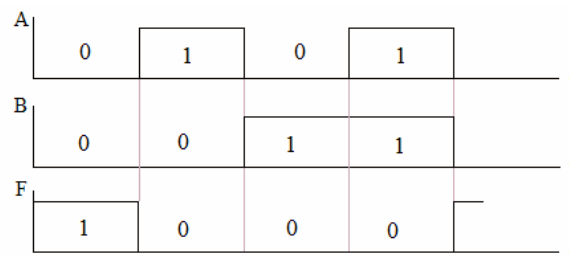


Рисунок 2.10 - Часова діаграма роботи елементу АБО-НІ

Логіка роботи двохвходового елементу І-НІ представлена в таблиці 2.2.

Таблиця 2.2 – Логіка роботи двохвходового елементу І-НІ

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

На основі логіки роботи отримують вираз для булевої функції:

$$F = \overline{AB}$$

Використовуючи принцип суперпозиції, функцію F можна записати для довільного числа змінних:

$$F = \overline{A \cdot B \cdot C \cdot \dots \cdot n}$$

Схема ДТЛ – елементу, що реалізує булеву операцію І-НІ для двох змінних A і B , і його умовне графічне позначення показані на рисунку 2.11.

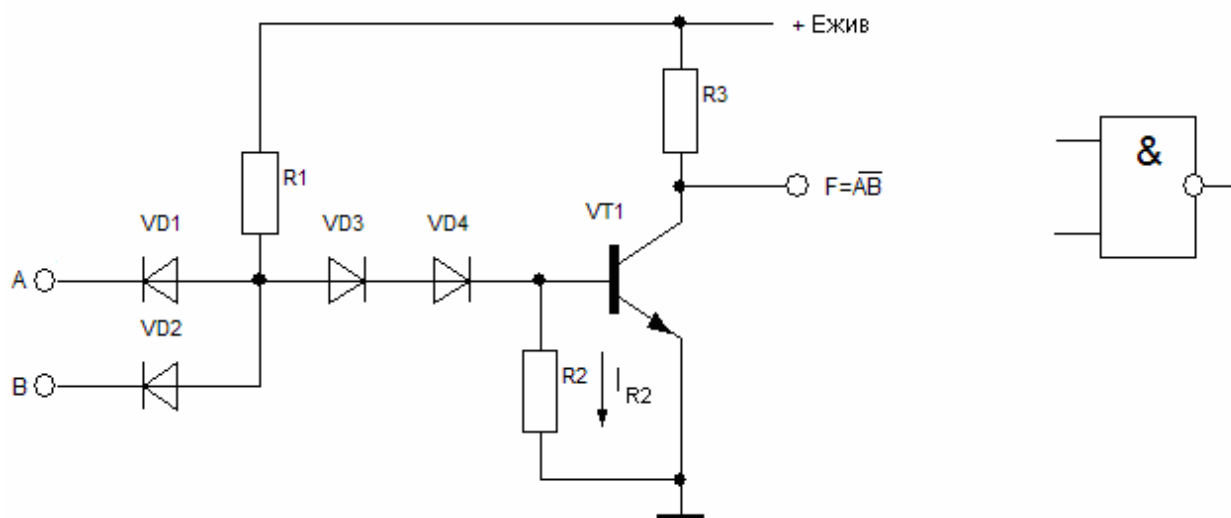


Рисунок 2.11 - Схема ДТЛ – елементу І-НІ

Транзистор $VT1$ відкривається і формує низький рівень вихідної напруги U^0 лише при наявності високих рівнів вхідних сигналів U^1 на обох входах одночасно. При цьому вмикаючий струм I_0 , протікає від джерела струму $U_{ж}$ по колу: резистор $R1$, діоди $VD3$, $VD4$ і база транзистора $VT1$. При подачі хоча б на один з входів (або на обидва) сигналу низького рівня весь струм від джерела живлення $U_{ж}$ замикається по колу: резистор $R1$, діод $VD1$ або $VD2$ (або обидва) на відповідне джерело вхідних сигналів. В результаті цього транзистор $VT1$ закривається і на виході встановлюється високий рівень напруги U^1 . Тривалість перемикання елементу практично визначається швидкодією транзистора. Часові діаграми роботи елементу І-НІ показані на рисунку 2.12.

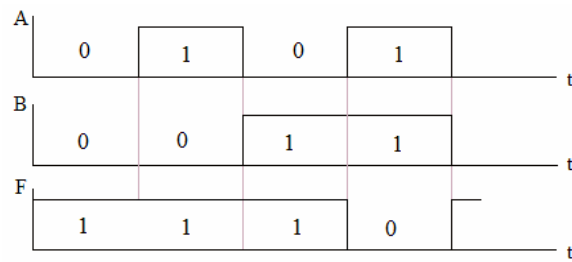


Рисунок 2.12 - Часові діаграми роботи елементу І-НІ

2.5 Елементи транзисторно-транзисторної логіки (ТТЛ)

Елементи транзисторно-транзисторної логіки складають базу мікросхем середньої і високої швидкодії. Розроблено і використовується декілька варіантів схем, що мають різні параметри.

До складу такого елементу входить багатоемітерний транзистор VT1 (рис. 2.13), що здійснює логічну операцію І і транзистор VT2, що реалізує операцію НІ.

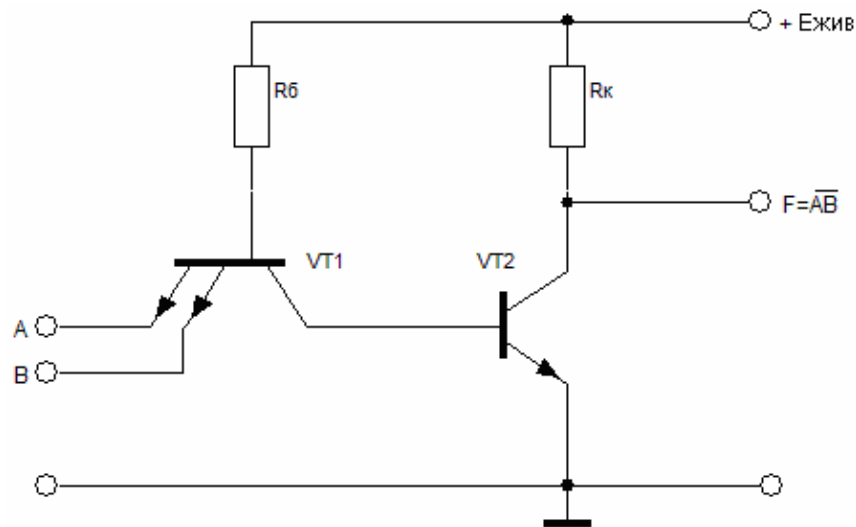


Рисунок 2.13 – Схема ТТЛ з простим інвертором

Багатоемітерний транзистор (БЕТ) є основою ТТЛ. За наявності на входах схеми, тобто емітерах БЕТ сигналу

$$U^0 = U_{\text{кена.}}$$

емітерні переходи зміщені в прямому напрямку і через VT1 протікає значний базовий струм

$$I_{\text{б1}} = (E - U_{\text{бнас}} - U_{\text{кнас}}) / R_{\text{б}}$$

достатній для того, щоб транзистор знаходився в режимі насичення. При цьому напруга колектор – емітер VT1 $U_{\text{кнас}} = 0,2$ В. Напруга на базі транзистора VT2 дорівнює

$$U^0 + U_{\text{кнас}} = 2U_{\text{кнас}} < U_{\text{бнас}}$$

і транзистор VT2 закритий. Напруга на виході схеми відповідає рівню U^1 . У такому стані схема знаходитиметься, поки хоч би на одному з входів сигнал дорівнює U^0 .

Якщо вхідну напругу підвищувати від рівня U^0 на всіх входах одночасно, або на одному з входів, за умови, що на останні входи поданий сигнал U^1 , то вхідна напруга на базі підвищується, і при

$$U_{\text{б}} = U_{\text{вх}} + U_{\text{кнас}} = U_{\text{бнас}}$$

і транзистор VT2 відкриється. В результаті, збільшиться струм бази VT2, який протікатиме від джерела живлення через резистор $R_{\text{б}}$ і колекторний перехід VT1, і транзистор VT2 перейде в режим насичення. Подальше підвищення $U_{\text{вх}}$ приведе до замикання емітерних переходів транзистора VT1, і в результаті він перейде в режим, при якому колекторний перехід буде зміщений в прямому напрямі, а емітерні – в зворотному (інверсний режим включення). Напруга на виході схеми:

$$U_{\text{вих}} = U_{\text{кнас}} = U^0 \text{ (транзистор VT2 в насиченні).}$$

Таким чином, розглянутий елемент здійснює логічну операцію І-НІ.

Проста схема елемента ТТЛ має ряд недоліків. При послідовному включенні таких елементів, коли до виходу елемента підключаються емітери інших таких же елементів, струм, який споживається від ЛЕ, збільшується, зменшується напруга високого рівня U^1 . Тому елемент володіє низькою здатністю навантаження. Це зумовлено наявністю великих емітерних струмів багатоемітерного транзистора в інверсному режимі, які споживаються від ЛЕ транзисторами – навантаження. Крім того, ця схема має малу завадостійкість по відношенню до рівня позитивної перешкоди:

$$U_{\text{пер}}^+ = U_{\text{бенас}} - U^0 = U_{\text{бенас}} - 2U_{\text{кенас}}$$

Для усунення вказаних недоліків використовують схеми ТТЛ із складним інвертором (рис. 2.14). Схема ТТЛ із складним інвертором також, як і схема з простим інвертором, здійснює логічну операцію І-НІ.

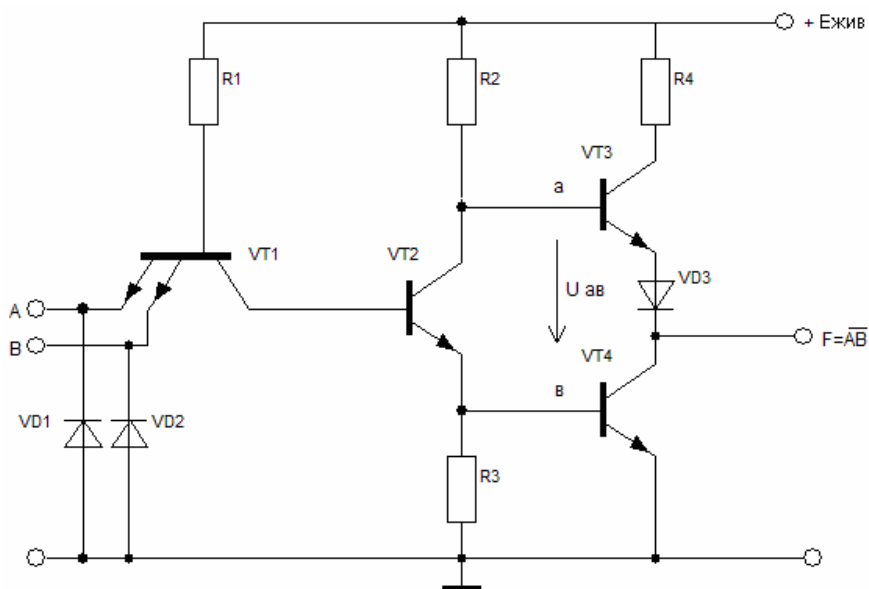


Рисунок 2.14 – Схема ТТЛ із складним інвертором

За наявності на входах напруги U^0 багатоемітерний транзистор VT1 знаходиться в режимі насичення, а транзистор VT2 закритий. Отже, закритий і транзистор VT4, оскільки струм через резистор R3 не протікає і напруга на базі VT4: $U_{\text{бе4}} = 0$. Транзистор VT3 відкритий, оскільки його база підключена до джерела живлення $E_{\text{жив}}$ через резистор R2. Опір резистора R4 невеликий, тому VT3 працює як емітерний повторювач. Через транзистор VT3 і відкритий діод VD3 протікає струм навантаження логічного елементу і вихідна напруга, яка відповідає рівню U^1 та дорівнює напрузі живлення за мінусом падіння напруги $U_{\text{бенас}}$ на VT3, падіння напруги на відкритому діоді $U_{\text{VD3}} = U_{\text{бенас}}$ і невелике падіння напруги на опорі R2 від струму бази VT3:

$$U^1 = E - 2U_{\text{бенас}} - R_2 I_{\text{б3}}; \quad U_{\text{п}} = 2U_{\text{бенас}}$$

Розглянутому режиму відповідає ділянка 1 передавальної характеристики логічного елементу ТТЛ (рис. 2.15).

При збільшенні напруги на всіх входах потенціал бази VT2 зростає і при $U_{вх} = U_{пор}^0$ транзистор VT2 відкривається, починає протікати колекторний струм $I_{к2}$ через резистори R2 і R3.

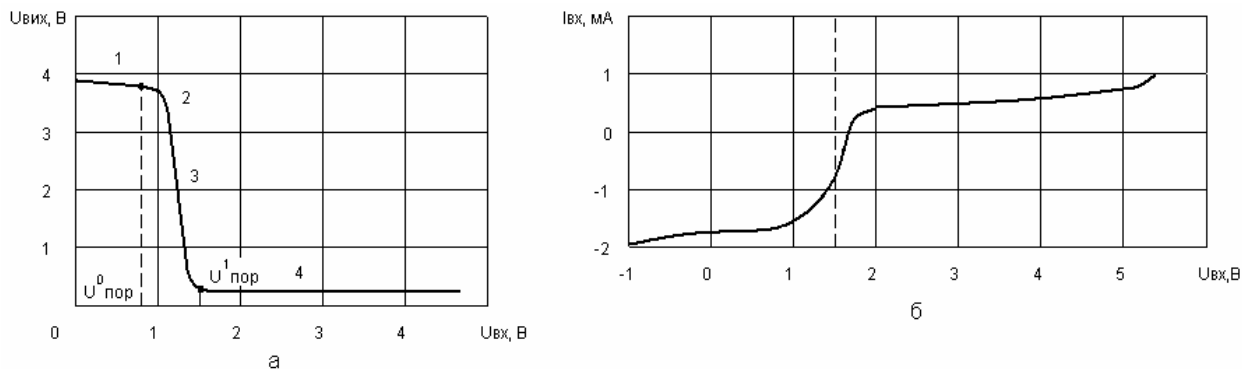


Рисунок 2.15 – Передавальна і вхідна характеристики базового логічного елемента ТТЛ

В результаті базовий струм VT3 зменшується, падіння напруги на ньому збільшується, і вихідна напруга знижується (ділянка 2). Поки на резисторі R3 падіння напруги $U_{R3} < U_{бенас}$ транзистор VT4 закритий. Коли

$$U_{вх} = U_{пор}^1 = 2U_{бенас} - U_{кенаc}$$

відкривається транзистор VT4. Подальше збільшення вхідної напруги призводить до насичення VT2 і VT4 і переходу VT1 в інверсний режим (ділянка 3). При цьому потенціал точки «а» (рис.2.14) дорівнює

$$U_a = U_{бенас} + U_{кенаc},$$

а точки «б»: $U_b = U_{кенаc}$, отже,

$$U_{аб} = U_a - U_b = U_{бенас}.$$

Для відмикання транзистора VT3 і діода VD3 потрібно $U_{аб} \geq 2U_{бенас}$. Оскільки ця умова не виконується, то VT3 і VD3 виявляються закритими і напруга на вході схеми дорівнює $U_{кенаc} = U^0$ (ділянка 4).

При перемиканні є проміжки часу, коли обидва транзистори VT3 і VT4 відкрито і виникають кидки струму. Для обмеження амплітуди цього струму в схему включають резистор з невеликим опором ($R4 = 100 \dots 160 \text{ Ом}$).

При негативній напрузі на емітерах БЕТ більше 2В розвивається тунельний пробій і вхідний струм різко збільшується. Для захисту ЛЕ від дії негативної перешкоди в схему введені діоди VD1, VD2, які обмежують її на рівні 0,5 ... 0,6 В.

З метою збільшення швидкодії елементів ТТЛ, в елементах ТТЛШ використовуються транзистори Шоттки, які являються поєднанням звичайного транзистора і діода Шоттки, включеного між базою і колектором транзистора. Оскільки падіння напруги на діоді Шоттки у відкритому стані менше, ніж на звичайному р-п-переході, то велика частина вхідного струму протікає через діод і лише його мала доля втікає в базу. Тому транзистор не входить в режим глибокого насичення.

Отже, накопичення носіїв в базі із-за їх інжекції через колекторний перехід практично не відбувається. У зв'язку з цим має місце збільшення швидкодії транзисторного ключа з бар'єром Шоттки в результаті зменшення часу наростання струму колектора при включенні і часі розсмоктування при виключенні.

Середній час затримки поширення сигналу елементів ТТЛ з діодами Шоттки (ТТЛШ) приблизно в два рази менше в порівнянні з аналогічними елементами ТТЛ. Недоліком ТТЛШ є менша в порівнянні з аналогічними елементами ТТЛ завадостійкість $U_{\text{пox}}^+$ із-за більшого значення U^0 і меншого $U_{\text{пор}}$.

Елементи ТТЛ з трьома вихідними станами – мають додатковий вхід V – вхід дозволу (рис. 2.16). При подачі на цей вхід напруги U^0 транзистор VT1 відкритий і насичений, а транзистори VT2 і VT3 закриті і тому не впливають на роботу логічного елемента. Залежно від комбінації сигналів на інформаційних входах на виході ЛЕ може бути сигнал з рівнем U^0 або U^1 . При подачі на вхід V напруги з рівнем U^1 транзистор VT1 закривається, а транзистори VT2 і VT3 відкриваються, напруга на базі транзистора VT6 зменшується до рівня:

$$U_{\text{бенас}} + U_{\text{VD1}},$$

транзистори VT5, VT6, VT7 закриваються і ЛЕ переходить в високоімпедансний (третій) стан, тобто відключається від навантаження.

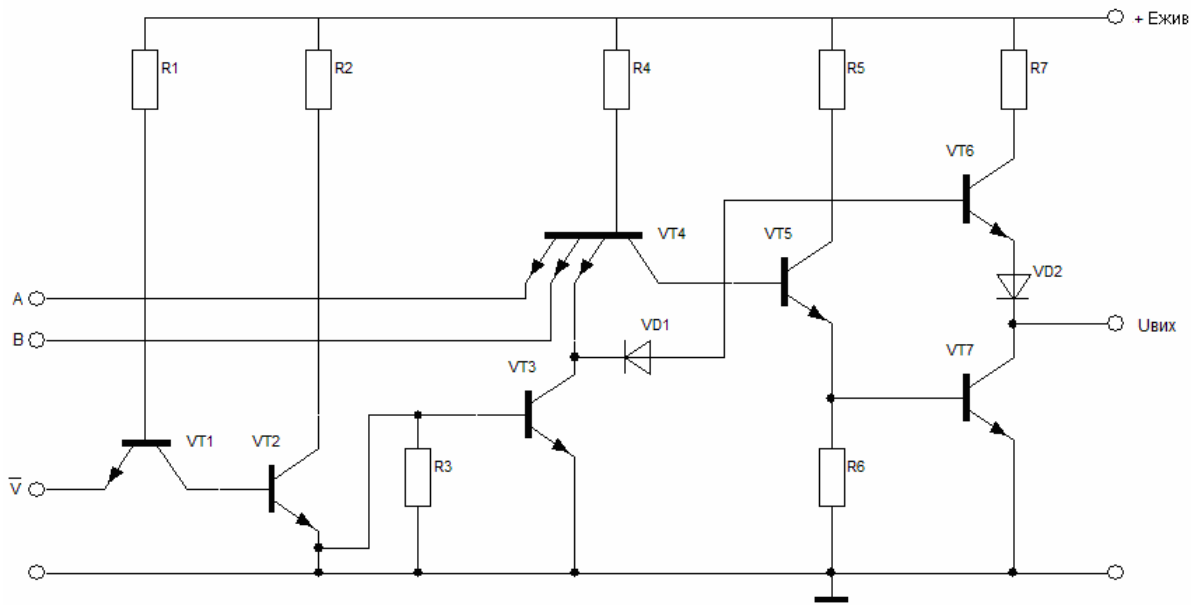


Рисунок 2.16 – Логічний елемент ТТЛ з трьома станами на виході

2.6 Елементи емітерно-зв'язаної логіки (ЕЗЛ)

Елементи емітерно-зв'язаної логіки являються основною елементною базою для мікросхем надвисокої швидкодії. Для зменшення затримок перемикання транзистори в елементах ЕЗЛ працюють в ненасиченому режимі і логічний перепад зменшений до $U_{л} < U^*$. Основою ЕЗЛ є швидкодіючий перемикач струму (рис. 2.17).

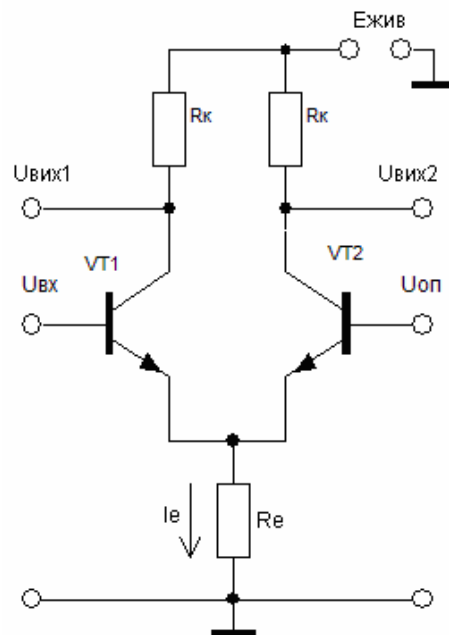


Рисунок 2.17 – Перемикач струму

Він складається з двох транзисторів, в колекторне коло яких включені резистори навантаження R_K , а в коло емітерів обох транзисторів - загальний резистор R_e , по величині значно більший R_K . На вхід одного з транзисторів подається вхідний сигнал $U_{вх}$, а на вхід іншого – опорна напруга $U_{оп}$.

Схема симетрична, тому у початковому стані ($U_{вх} = U_{оп}$) і через обидва транзистора протікають однакові струми. Через опір R_e протікає загальний струм I_e . При збільшенні $U_{вх}$ струм через транзистор VT1 збільшується, зростає падіння напруги на опорі R_e , транзистор VT2 призначується і струм через нього зменшується. При вхідній напрузі, що дорівнює $U_{вх} = U_1$, транзистор VT2 закривається і весь струм протікає через транзистор VT1. Параметри схеми і струм I_e вибираються так, щоб транзистор VT1 у відкритому стані працював в лінійному режимі на кордоні області насичення. При зменшенні $U_{вх}$ до рівня $U_{вх} = U^0$, навпаки, транзистор VT1 закритий, а транзистор VT2 знаходиться в лінійному режимі на кордоні з областю насичення.

У схемі ЕЗЛ (рис. 2.18) паралельно транзистору VT2 включається ще один або декілька транзисторів (залежно від коефіцієнта об'єднання по входу), які складають одне з плечей перемикача струму.

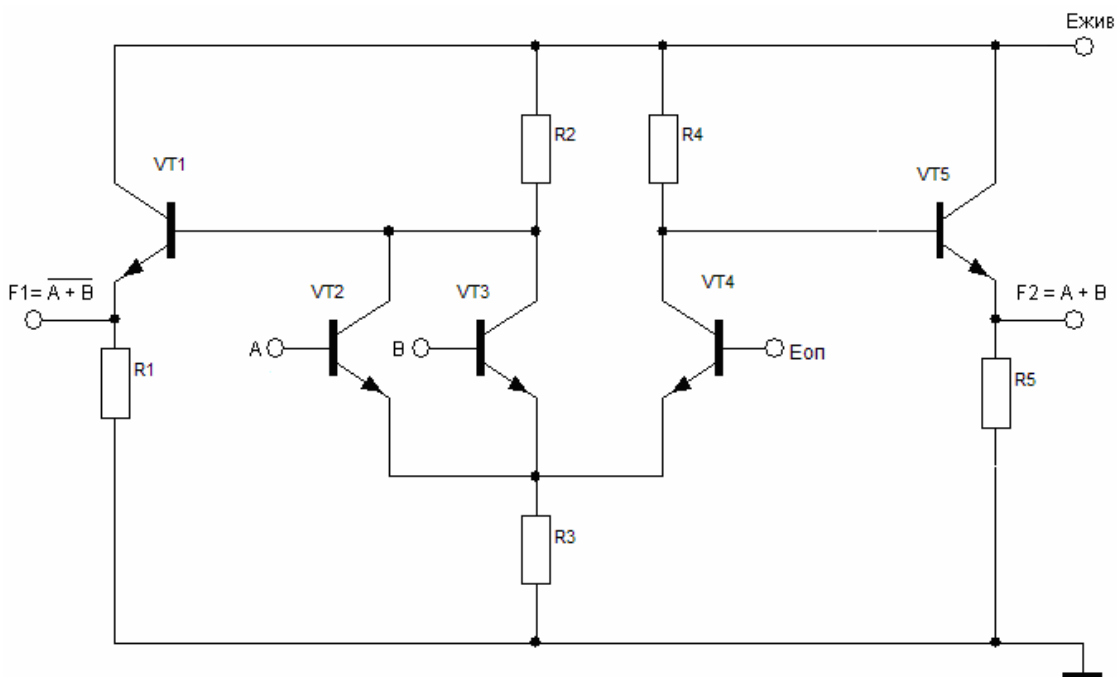


Рисунок 2.18 - Елемент АБО-НІ / АБО емітерно-зв'язаної логіки

До виходів ЛЕ для підвищення роздільної здатності навантаження підключено два емітерних повторювача VT1 і VT5. На виходах ЛЕ реалізуються функції $F1 = \overline{A + B}$ і $F2 = A + B$.

У вихідному стані транзистори VT2, VT3 замкнуті і струм від джерела проходить через транзистор VT4, який відкритий опорною напругою $E_{оп}$. Оскільки всі вхідні транзистори замкнуті, то на базу вихідного транзистора VT1 подається високий позитивний потенціал, рівний потенціалу колекторів транзисторів VT2, VT3. Транзистор VT1 при цьому відкритий. Через резистор R1 протікає великий струм, напруга на резисторі підвищується, що відповідає рівню логічної 1 на виході F1.

На колекторі транзистора VT4 низький потенціал, так як складова $I_{R4}R4$ в рівнянні

$$U_{кVT4} = E_{жив} - I_{R4}R4$$

має велике значення. Транзистор VT5 закритий. Через резистор R5 протікає залишковий струм закритого транзистора VT5, що відповідає рівню логічного 0 на виході F2.

У тих випадках коли відкритий хоч би один з транзисторів VT2, VT3, або обидва, різко збільшується струм I_{R2} , на колекторах транзисторів низький потенціал. Транзистор VT1 закритий. Через резистор R1 протікає залишковий струм закритого транзистора VT1, що відповідає рівню логічного 0 на виході F1.

Збільшується струм I_{R3} і падіння напруги на резисторі R3, яке підпирає емітер транзистора VT4. Транзистор VT4 закривається, на його колекторі високий потенціал. Транзистор VT5 при цьому відкритий. Через резистор R5 протікає великий струм, напруга на резисторі підвищується, що відповідає рівню логічної 1 на виході F2.

Значення логічного перепаду елементів ЕЗЛ:

$$U_{л} = U^* = 0,7В.$$

Опорна напруга $E_{оп}$ вибирається рівною напівсумі рівнів U^0 і U^1 , тобто

$$E_{\text{оп}} = E_{\text{жив}} - \frac{3}{2} U^*.$$

При цьому, рівні U^0 і U^1 розташовані симетрично відносно $E_{\text{оп}}$ на $\pm 1/2 U^*$.

На практиці в схемах ЕЗЛ заземляють не негативний, а позитивний полюс джерела живлення, так що всі робочі потенціали на рисунку 2.18 виявляються негативними. Зрозуміло, це не міняє принципу дії схеми і основних співвідношень, проте заземлення позитивної шини живлення істотно зменшує вплив перешкод, що по ній проходять, на величини рівнів U^0 і U^1 .

2.7 Елементи інтегральної інжекційної логіки (І²Л)

Елементи цього типу представляють собою фізично об'єднані горизонтальні р-п-р і вертикальні п-р-п транзистори (рис. 2.19).

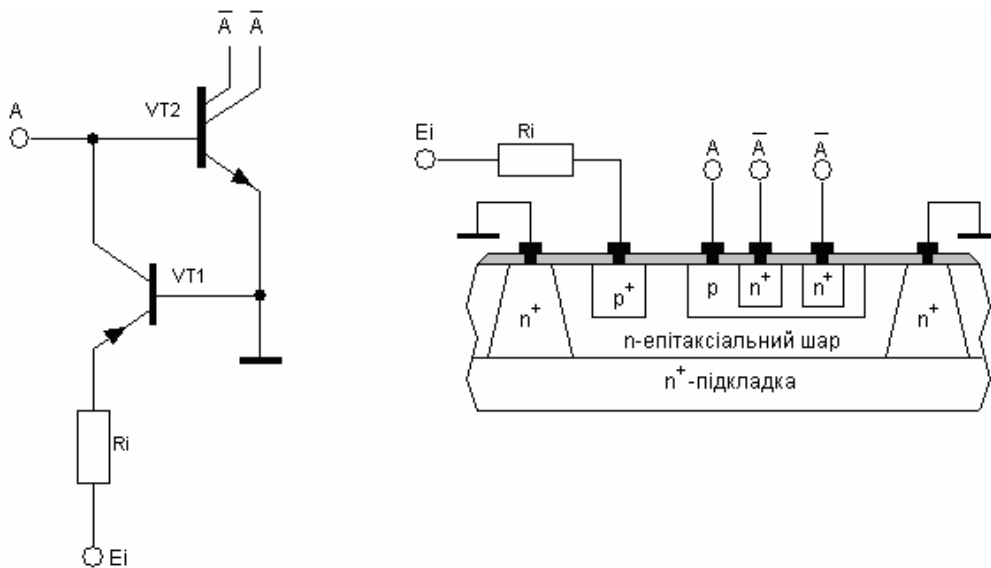


Рисунок 2.19 – Логічний елемент І²Л та його топологія

Емітерна область р-п-р транзистора, яка називається інжектором, підключається до позитивного джерела живлення. Загальна область п-типу служить базою р-п-р транзистора і емітером п-р-п транзистора і підключається до загального виводу. Колектор р-п-р транзистора і база п-р-п транзистора так само є єдиною областю напівпровідника р-типу.

У коло емітер – база інжектора подається напруга живлення $U_{\text{жив}}$. Мінімальна напруга джерела визначається падінням напруги на емітерному переході: $U_{\text{ке нас}} = 0,7\text{В}$. Для стабілізації струму емітера послідовно з джерелом включається резистор R_i і беруть напругу джерела живлення $U_{\text{жив}} = 1 \dots 1,2\text{В}$.

При цьому р-n-перехід емітер-база VT1 відкритий і відбувається дифузія дірок до колекторного переходу. По мірі руху до колектора частина дірок рекомбінує з електронами, але їх значна частина досягає колекторного переходу і, пройшовши через нього, потрапляє в р-базу інвертора VT2. Інжекція дірок в базу йде постійно, незалежно від вхідної дії.

Якщо напруга на базі VT2 $U_{\text{вх}} = U^0$ дірки, що потрапляють в р-базу інвертора, безперешкодно стікають до негативного полюса джерела живлення. У колі колектора транзистора VT2 струм не протікає і це еквівалентно розімкненому стану колекторного кола VT2. Такий стан вихідного кола відповідає напрузі U^1 .

При $U_{\text{вх}} = U^1$ дірки в р-базі інвертора накопичуються. Потенціал бази починає підвищуватися і відповідно знижується напруга на переходах VT2 до тих пір, поки ці переходи не відкриються. Тоді в колекторному колі транзистора VT2 протікатиме струм і різниця потенціалів між емітером і колектором інвертора VT2 буде близька до нуля, тобто цей транзистор є короткозамкнутою ділянкою кола, і цей стан відповідатиме рівню U^0 . Таким чином, елемент І²Л виконує роль ключа.

Від одного інжектора може житися декілька схем, тобто горизонтальний р-n-р VT1 може бути багатоклекторним (рис. 2.20).

Вертикальний n-p-n транзистор зазвичай має декілька колекторів, які являються логічними виходами елементу. При такій фізичній структурі не потрібна ізоляція між окремими елементами І²Л, оскільки вони мають загальну область. По бічних сторонах р-областей баз n-p-n транзисторів створюється розділове n-кільце яке виключає вплив паразитних горизонтальних р-n-р структур між елементами, а також сприяє збільшенню коефіцієнта посилення для n-p-n транзисторів.

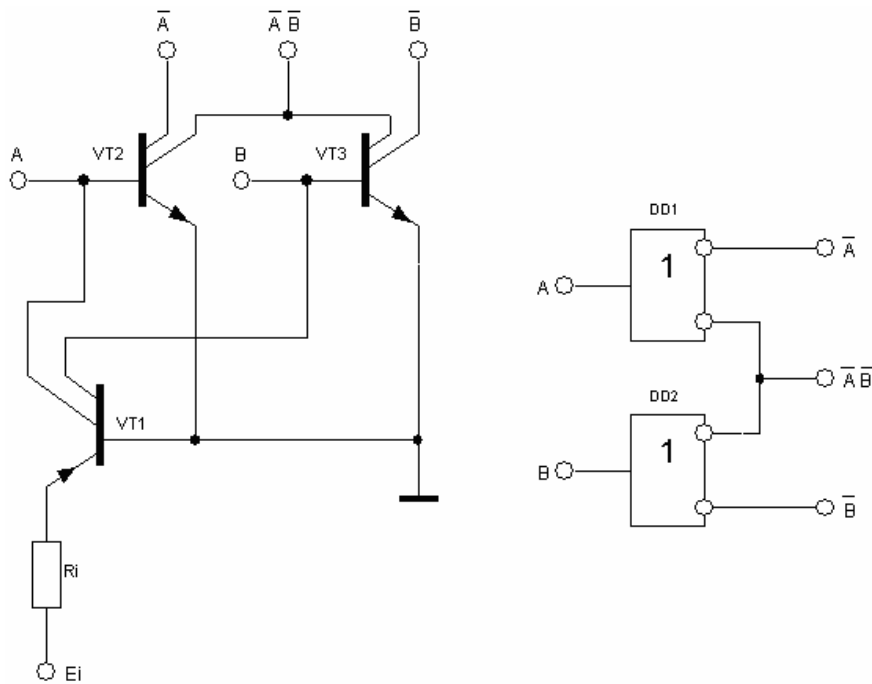


Рисунок 2.20 – Розширення логічних можливостей елементів І²Л

Типовий елемент І²Л є багатовихідний інвертор. Шляхом з'єднання виходів інверторів металевими провідниками реалізується функція І. На сполучених виходах елементів підтримується низький потенціал U^0 , якщо високий потенціал U^1 поданий хоча б на один з входів А або В.

2.8 Логічні елементи на МОН-транзисторах

В логічних елементах на МОН-транзисторах використовується два типи транзисторів: управляючі і навантаження. Управляючі – мають короткий, але досить широкий канал і тому мають високе значення крутизни та управляються малою напругою. Транзистори навантаження, навпаки, мають довший, але вузький канал, тому мають вищий вихідний опір і виконують роль великого активного опору.

Істотною перевагою логічних елементів на МОН-транзисторах перед логічними елементами на біполярних транзисторах є мала потужність, споживана вхідним колом. Проте, по швидкодії вони поступаються схемам на біполярних транзисторах. Це зумовлено тим, що у них є порівняно великі паразитні ємності C_{zi} і C_{ci} , на перезарядку яких витрачається певний час. Крім того, вихідний опір

у відкритого МОН-транзистора більший, ніж в біполярного, що збільшує час заряду конденсаторів навантаження і обмежує здатність навантаження ЛЕ.

2.8.1 Логічні елементи на ключах з динамічним навантаженням

Логічні елементи на ключах з динамічним навантаженням складаються з одного навантаження і декількох управляючих транзисторів. Якщо управляючі транзистори включені паралельно, то елемент здійснює логічну операцію АБО-НІ, а при послідовному з'єднанні – операцію І-НІ (рис. 2.21).

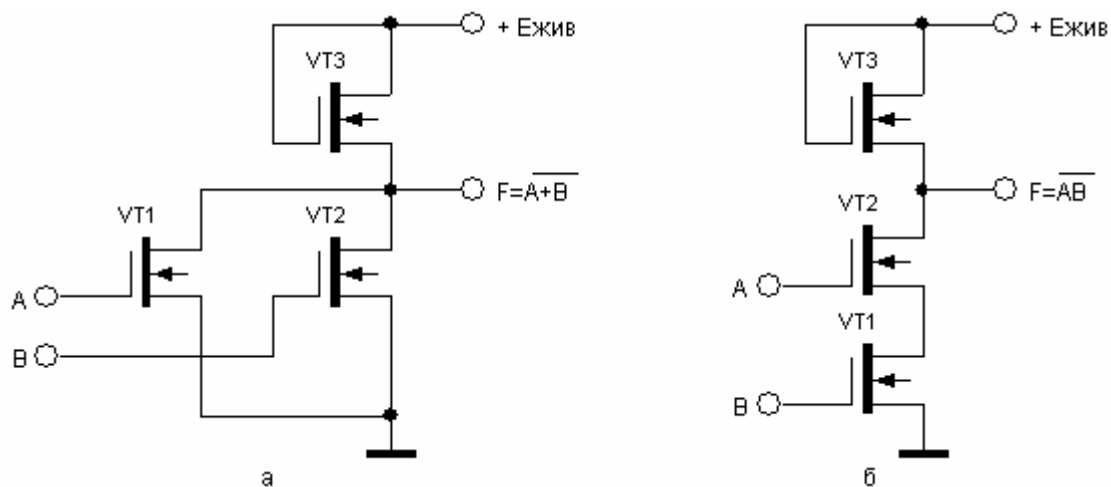


Рисунок 2.21 - Схеми елементів МОН ТЛ: а) АБО-НІ, б) І-НІ

За наявності на входах А і В напруги $U_{вх} = U^0 < U_{зи\ пор.}$ управляючі транзистори VT1 і VT2 закриті. При цьому напруга на виході відповідає рівню U^1 . Коли на одному або на обох входах елемента діє напруга $U_{вх} = U^1 > U_{зи\ пор.}$, то на виході маємо U^0 , що відповідає виконанню логічної операції АБО-НІ.

У схемі елемента І-НІ управляючі транзистори включені послідовно, тому рівень U^0 на виході схеми має місце лише при сигналах U^1 на обох входах.

Елементи МОН ТЛ мають високу завадостійкість, великий логічний перепад, малу споживану потужність і порівняно низьку швидкодію. Для елементів на низькопорогових МОН-транзисторах зазвичай $U_{жив} = 5 \dots 9\text{В}$, а на високпорогових - $U_{жив} = 12,6 \dots 27\text{В}$. Основні параметри МОН ТЛ: $P_{спож} = 0,4 \dots 5\text{ мВт}$, $t_{затр\ сер} = 20 \dots 200\text{ нс}$, $U^0 \leq 1\text{В}$, $U^0 \geq 7\text{В}$.

2.8.2 Логічні елементи на комплементарних ключах

Комплементарний ключ складається з двох МОН-транзисторів з каналами різного типу провідності, входи яких сполучені паралельно, а виходи послідовно (рис. 2.7). При напрузі на затворах більше порогової, для транзистора з каналом певного типу, відповідний транзистор відкритий, а інший закритий. При напрузі протилежної полярності відкритий і закритий транзистори міняються місцями. ЛЕ на комплементарних ключах (КМОН) мають ряд безперечних переваг. Вони успішно працюють при зміні в широких межах напруги джерела живлення (від 3 до 15 В), що недосяжно для ЛЕ, до складу яких входять резистори. У статичному режимі при великому опорі навантаження ЛЕ КМОН практично не споживають потужності. Для них також характерні: стабільність рівнів вихідного сигналу і мала його відмінність від напруги джерела живлення; високий вхідний і малий вихідний опори; легкість узгодження з мікросхемами інших технологій.

Схема ЛЕ КМОН, що виконує функцію АБО-НІ, приведена на рисунку 2.22, а.

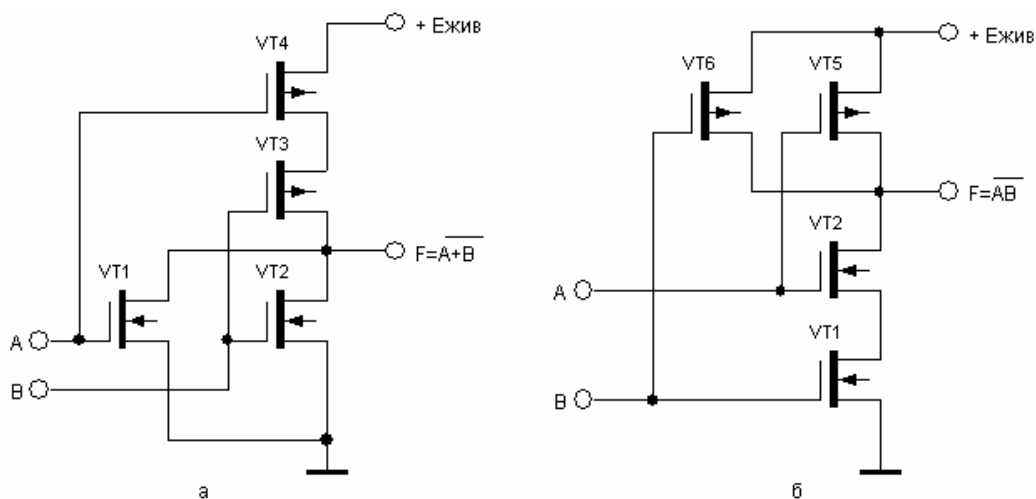


Рисунок 2.22 – Схеми логічних елементів КМОН ТЛ: а) АБО-НІ, б) І-НІ

Транзистори VT1 і VT3 мають канал р-типу і відкриті при напрузі на затворах, близьких до нуля. Транзистори VT2 і VT4 мають канал n-типу і відкриті при напрузі на затворах, більших порогового значення. Якщо на обох або на

одному з входів діє рівень U^0 , то на виході схеми буде сигнал U^1 , що відповідає виконанню логічної операції АБО-НІ.

Якщо групи ярусно і паралельно включених транзисторів поміняти місцями, то буде реалізовано елемент, що виконує функцію І-НІ (рис. 2.22, б). Він працює аналогічно попередньому. Транзистори VT1 і VT3 мають канал р-типу і відкриті при напрузі на затворах, близької до нуля. Транзистори VT2 і VT4 мають канал n-типу і відкриті при напрузі на затворах, більшої порогового значення. Якщо відкриті обидва ці транзистора, то на виході буде встановлено сигнал U^0 .

У ЛЕ КМОН дуже просто реалізуються елементи з трьома стійкими станами. Для цього послідовно з транзисторами інвертора включають два комплементарних транзистора VT1, VT4 (рис. 2.23), які керуються інверсними сигналами V і \bar{V} .

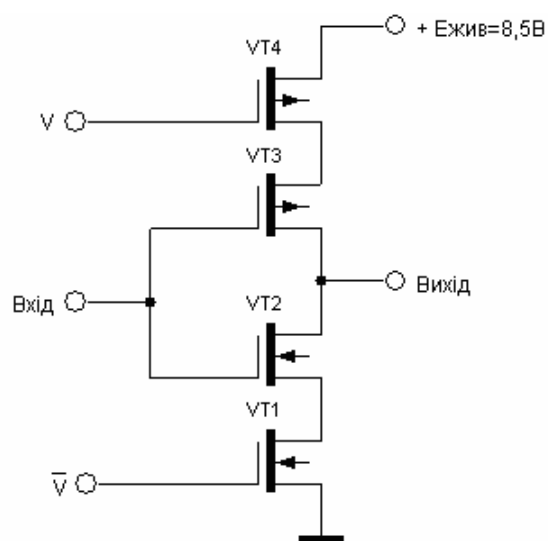


Рисунок 2.23 – Інвертор з трьома вихідними станами

Узгодження ЛЕ ТТЛ з ЛЕ КМОН можна виконати декількома способами:

1) жити ЛЕ КМОН малою напругою (+5 В), при якій сигнали ЛЕ ТТЛ перемикають транзистори ЛЕ КМОН.

2) використовувати ЛЕ ТТЛ з відкритим колектором, в коло виходу яких включений резистор, підключений до додаткового джерела напруги живлення (рис. 2.24).

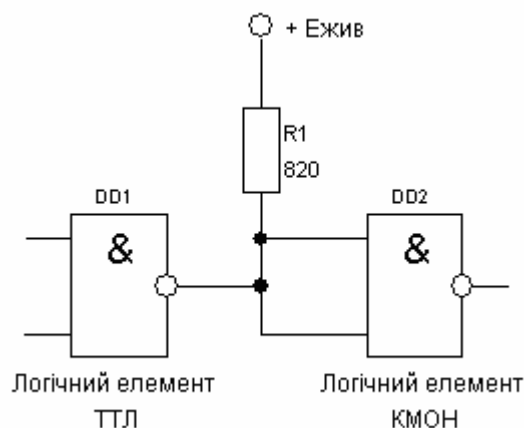


Рисунок 2.24 – Узгодження ЛЕ ТТЛ з ЛЕ КМОП

При зберіганні і монтажі слід уникати негативної дії статичної електрики. Тому при зберіганні виводи мікросхем електрично замикають між собою. Монтаж їх виконується при вимкненій напрузі живлення, причому, обов'язковим є використання браслетів, за допомогою яких тіло електромонтажників з'єднується із заземленням.

Логічні елементи КМОП широко застосовуються при побудові економічних цифрових пристроїв малої і середньої швидкодії.

2.9 Схемне моделювання логічних елементів

Існують різні схмотехнічні реалізації ЦІС залежно від використаної елементної бази (біполярні або польові транзистори) і технологічних особливостей їх реалізації. Мікроелектронні логічні елементи являються з'єднанням компонентів (транзисторів, резисторів, діодів і ін.), кожен з яких виконує певні функції перетворення електричного струму або потенціалу. В результаті цього виконується задана логічна функція, тобто відбувається певне перетворення логічних сигналів (0 і 1), що поступають на входи елемента у вигляді потенціалів U^0 і U^1 , у вихідні логічні сигнали, що також представляються потенціалами U^0 і U^1 . В процесі схмотехнічного моделювання з логічних елементів створюється схема з'єднання компонентів, яка забезпечує необхідне перетворення електричних сигналів і задовольняє вимогам, які пред'являються до її параметрів.

2.9.1 Моделювання схем на МОН та КМОН транзисторах

При послідовному включенні МОН та КМОН транзисторів виконується кон'юнкція з інверсією, при послідовному – диз'юнкція з інверсією (рис. 2.21 – 2.22). Моделювання цифрової схеми (комбінаційної схеми або мікроелемента великої інтегральної схеми), що виконує логічну функцію, виконується за наступною методикою:

1) за допомогою законів і тотожностей алгебри логіки або карт Карно виконується перетворення ДДНФ або ДКНФ функцій для здобуття їх мінімізованих виразів;

2) використовуючи різні варіанти логічних перетворень, отримують декілька різних мінімізованих виразів даної функції, для їх подальшого аналізу. Функція представляється у вигляді інверсно - диз'юнктивної нормальної форми, або у вигляді інверсно - кон'юнктивної нормальної форми (табл. 2.3);

Таблиця 2.3 – Різні форми представлення логічної функції

№	Форми представлення	Приклад
1а	Диз'юнктивна (МДНФ): (НІ)–І–АБО	$\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C + A\overline{B}\overline{C}\overline{D}$
1б	Інверсно - кон'юнктивна: (НІ)–І–НІ–І–НІ	$\overline{\overline{A}\overline{B}\overline{C}\overline{D}} \cdot \overline{\overline{A}\overline{B}C} \cdot \overline{\overline{A}\overline{B}\overline{C}\overline{D}}$
1в	Інверсно - кон'юнктивна: (НІ)–АБО–І–НІ	$\overline{(A + D) \cdot (\overline{A} + C) \cdot (C + \overline{D})}$
1г	Диз'юнктивно-інверсна: (НІ)–АБО–НІ–АБО	$\overline{(A + \overline{B}) + (\overline{C} + D) + (D + \overline{B})}$
2а	Кон'юнктивна (МКНФ): (НІ)–АБО–І	$\overline{(\overline{A} + D) \cdot (\overline{B} + \overline{C}) \cdot (A + C)}$
2б	Інверсно - диз'юнктивна: (НІ)–АБО–НІ–АБО–НІ	$\overline{\overline{(A + B) + (\overline{C} + D) + (\overline{D} + B)}}$
2в	Інверсно - диз'юнктивна: (НІ)–І–АБО–НІ	$\overline{\overline{A}\overline{B}C + \overline{A}\overline{B}\overline{C}\overline{D} + \overline{B}C}$
2г	Кон'юнктивно-інверсна: (НІ)–АБО–НІ–АБО	$\overline{(\overline{A}\overline{B}\overline{C}\overline{D}) \cdot (\overline{B}\overline{C}\overline{D}) \cdot (\overline{A}\overline{B}\overline{C}\overline{D})}$

3) виконується з'єднання управляючих МДН транзисторів відповідно до отриманих мінімізованих інверсно-диз'юнктивної нормальної форми або інвер-

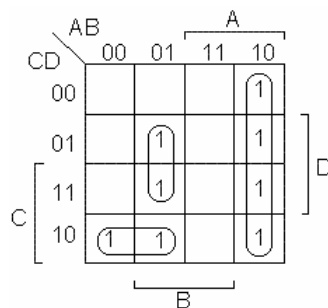
сно-кон'юнктивної нормальної форми функцій. При цьому кожній літері логічного виразу відповідає транзистор керування, на затвор якого надходить відповідна змінна. Для реалізації кон'юнктивних членів МДН транзистори керування з'єднуються послідовно, для реалізації диз'юнктивних членів – паралельно. Після з'єднання транзисторів керування підключаються транзистори навантаження і шини живлення. В результаті отримуються електричні схеми, які реалізують задану функцію;

4) виконується порівняльний аналіз отриманих схем. За результатами аналізу обирається схема, яка в найбільшій мірі задовольняє вимогам завдання на моделювання. При виборі схеми необхідно враховувати не лише її електричні параметри, але і її топологію, яка визначає необхідну площу кристала, довжину міжз'єднань, значення паразитних ємностей.

Приклад. Побудувати схемну модель на МДН-транзисторах, яка реалізує алгебраїчний вираз:

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

1) за допомогою карти Карно отримуємо мінімізований вираз функції;



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

2) за допомогою законів і тотожностей алгебри логіки отримуємо декілька різних мінімізованих виразів даної функції, для їх подальшого аналізу:

$$\begin{aligned} F &= \overline{A}BD + A\overline{B} + \overline{A}C\overline{D} = \overline{\overline{\overline{\overline{\overline{A}BD} + \overline{A\overline{B}} + \overline{A}C\overline{D}}}}} = \overline{\overline{\overline{\overline{A}BD} \cdot \overline{A\overline{B}} \cdot \overline{A}C\overline{D}}} = \\ &= \overline{(A + \overline{B} + \overline{D}) \cdot (\overline{A} + B) \cdot (A + \overline{C} + D)} = \overline{[A + (\overline{B} + \overline{D}) \cdot (\overline{C} + D)] \cdot (\overline{A} + B)} \end{aligned}$$

3) обираємо варіанти, які відповідають інвесно-кон'юнктивній формі і виконуємо моделювання кожної схеми (рис. 2.25 – 2.27);

4) виконується порівняльний аналіз отриманих схем.

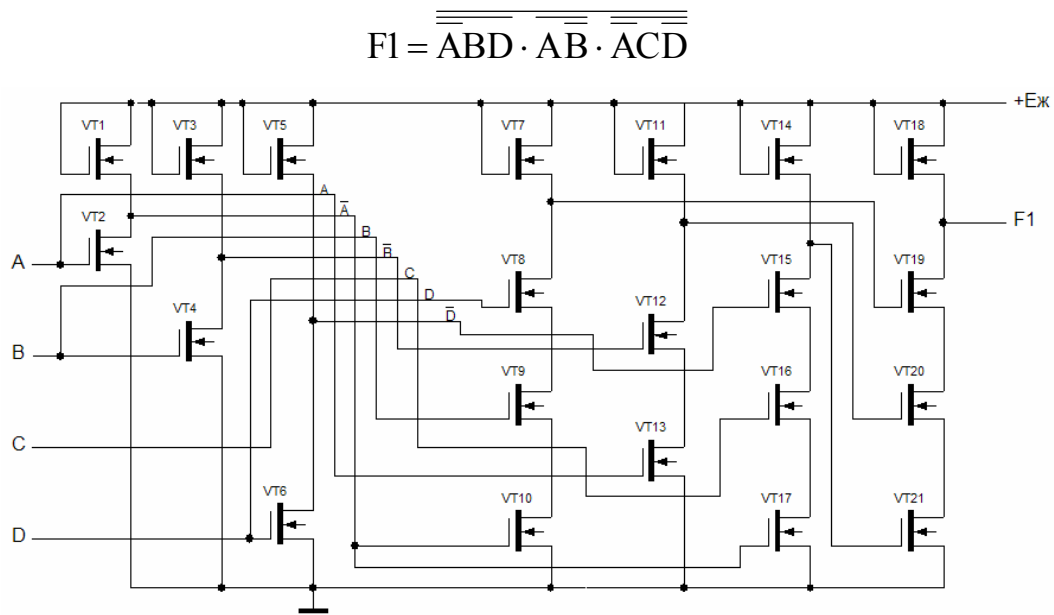


Рисунок 2.25 – Варіант схеми в інверсно-кон'юнктивній формі (НІ)–І–НІ–І–НІ

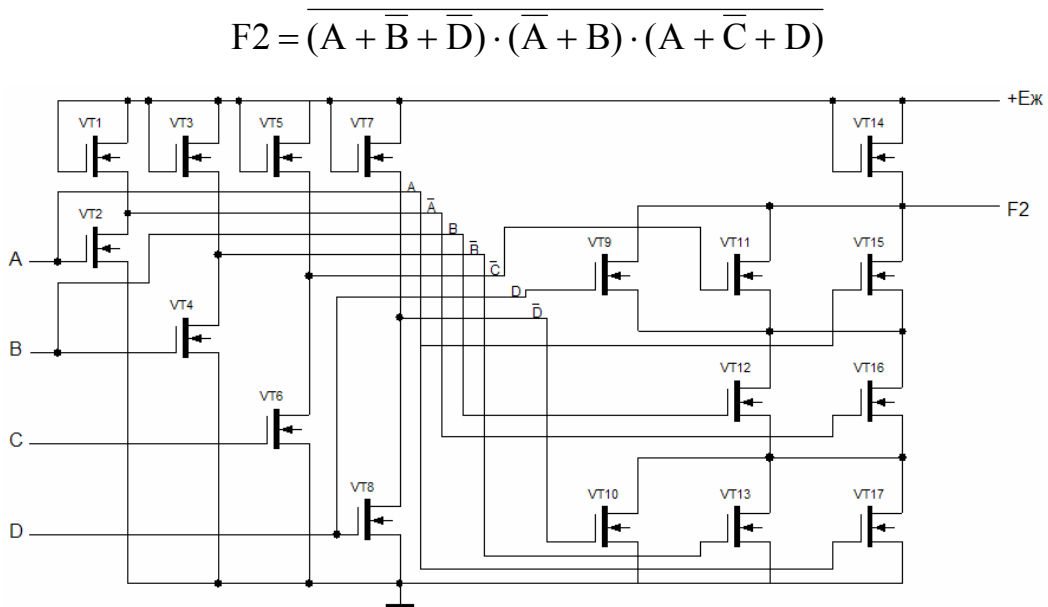


Рисунок 2.26 – Варіант схеми в інверсно-кон'юнктивній формі (НІ)–АБО–І–НІ

$$F3 = [A + (\overline{B + D}) \cdot (\overline{C + D})] \cdot (\overline{A + B})$$

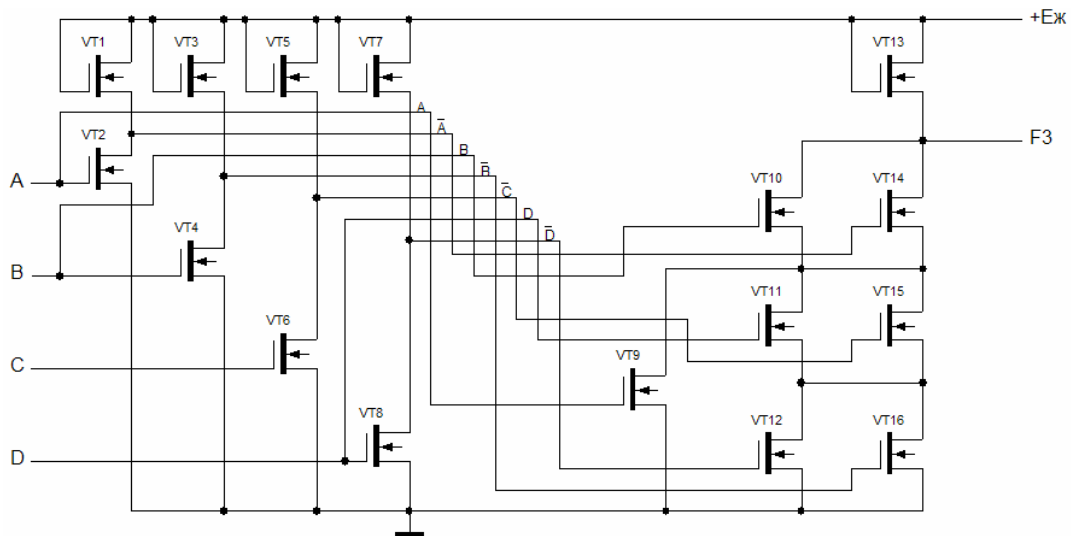


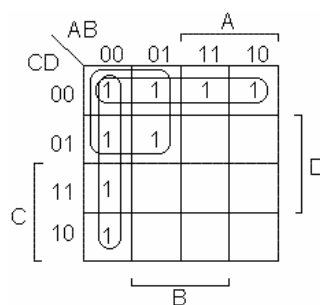
Рисунок 2.27 – Варіант схеми в інверсно-кон'юнктивній формі (НІ)–АБО–І–НІ з використанням закону дистрибутивності (1.31)

За результатами аналізу обирається схема, яка в найбільшій мірі задовольняє вимогам завдання на моделювання.

Приклад. Побудувати схемну модель на КМДН-транзисторах, яка реалізує алгебраїчний вираз:

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

1) за допомогою карти Карно отримуємо мінімізований вираз функції



2) за допомогою законів і тотожностей алгебри логіки отримуємо декілька різних мінімізованих виразів даної функції, для їх подальшого аналізу:

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

3) обираємо варіанти, які відповідають інверсно-кон'юнктивній формі і

виконуємо моделювання кожної схеми (рис. 2.28 – 2.30). У комплементарній МОН-структурі використовуються одночасно р – і – n - каналні транзистори. Логічні елементи в схемотехніці КМОП побудовані на двох типах транзисторів з індукованими каналами: навантажними з каналами р-типу і логічними з каналами n-типу.

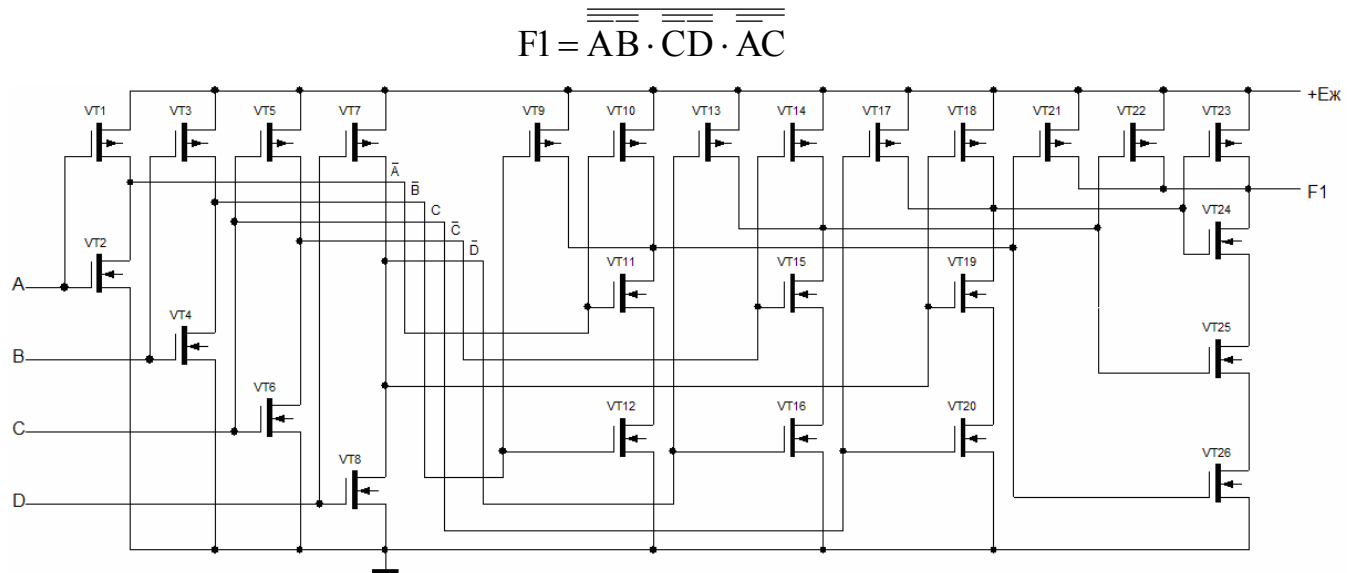


Рисунок 2.28 – Варіант схеми в інверсно-кон’юнктивній формі (НІ)–І–НІ–І–НІ

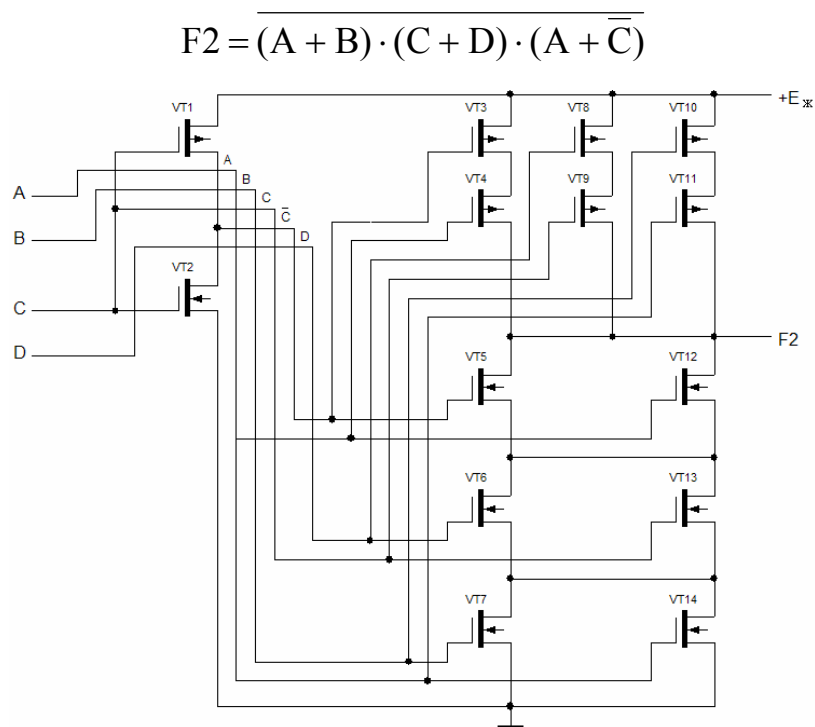


Рисунок 2.29 – Варіант схеми в інверсно-кон’юнктивній формі (НІ)–АБО–І–НІ

$$F3 = [A + (B \cdot \overline{C})] \cdot (C + D)$$

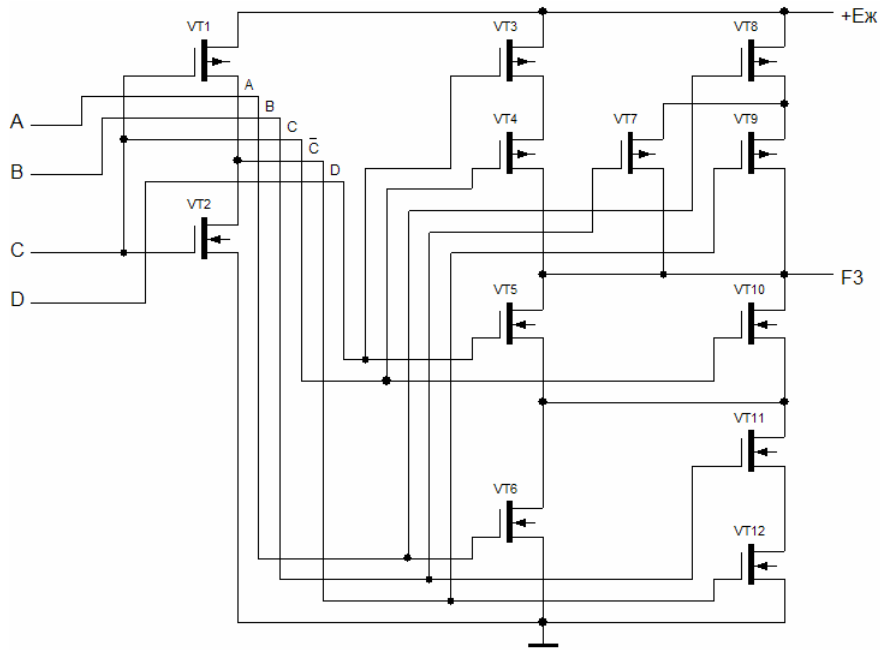


Рисунок 2.30 – Варіант схеми в інверсно-кон’юнктивній формі (НІ)–АБО–І–НІ з використанням закону дистрибутивності (1.31)

2.9.2 Моделювання схем на біполярних транзисторах

Моделювання схем ТТЛ, ДТЛ, ЕСЛ, І²Л виконується з використанням методу струмових графів. Метод заснований на узагальненому представленні електричної схеми у вигляді струмового графа вершинами якого служать потенціальні струмові функціональні елементи (ПСФЕ), які виконують операції над струмами та потенціалами, необхідними для реалізації логічних функцій. Типи ПСФЕ, їх зображення та призначення представлені в таблиці 2.4 [3].

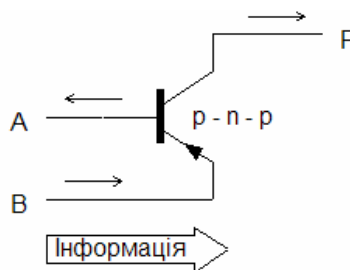
Знак ПСФЕ, який вказується у вигляді індексу $z = 0$, $z = 1$, при скороченому позначенні ДС0, СК1 та ін., задається залежно від напрямів протікання струму. Для РС і ОС знак визначається в залежності від напрямів протікання струму і передачі інформації від входів до виходів схеми: $z = 0$, якщо напрями збігаються, $z = 1$, якщо протилежні. Струм, який співпадає з напрямом передачі інформації вважатимемо позитивним, а який не співпадає – негативним.

Таблиця 2.4 – Основні типи і функції ПСФЕ та варіанти їх схемної реалізації

Тип ПСФЕ	Умовні позначення	Варіанти схемної реалізації	Логічні операції	Електричні Функції	
Активні (k = 1)	Джерело струму	ДС0		—	Завдання струму
		ДС1		—	
	Струмний ключ	СК0		Інверсія, заборона	Перемикання струму
		СК1		Інверсія, імплікація	
Пасивні (k = 0)	Розгалужувач струму	РС0		—	Розгалуження струму, підсилення або ослаблення струму
		РС1		—	
	Об'єднувач струму	ОС0		Диз'юнкція	Об'єднання струму, підсилення або ослаблення струму
		ОС1		Кон'юнкція	
Фіксатор потенціалу	ФП		—	Узгодження потенціалів	

Струмний ключ представляє собою інвертор. Розглянемо способи управління струмовими ключами.

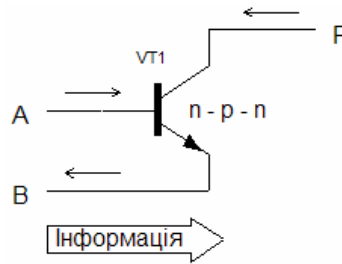
Транзистор р – n – р:



Емітерний струм співпадає з напрямом розповсюдження інформації, базовий – не співпадає.

Рівняння $F = \overline{AB}$ представляє собою функцію «заборона».

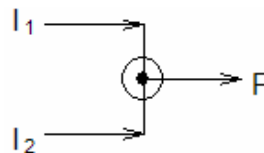
Транзистор n – p – n:



Емітерний струм не співпадає з напрямом розповсюдження інформації, базовий – співпадає.

Рівняння $\bar{F} = A\bar{B}$, $\bar{\bar{F}} = \bar{\bar{A}\bar{B}} = \bar{A} + \bar{B}$ представляє собою функцію «імплікація».

Різні схемні варіанти ОС0 виконують операцію диз'юнкції $F = A + B$, оскільки позитивний струм на виході тече за наявності струму хоч би в одній з вхідних гілок.



При включенні ОС1 аналогічна операція виконується із струмами протилежного напрямку, що відповідає інверсному значенню змінних $\bar{F} = \bar{A} + \bar{B}$, звідки отримуємо $\bar{\bar{F}} = \bar{\bar{A} + \bar{B}} = AB$. Таким чином реалізується операція кон'юнкції.

Розгалужувачі струму не виконують логічних операцій і відображають галуження електричного кола для підключення декількох подальших елементів або компонентів. ПСФЕ, які не виконують логічних операцій: джерела струму, які забезпечують струм живлення, необхідний для роботи схеми; фіксатори потенціалу, які служать для отримання необхідних потенціалів вузлів. Ці ПСФЕ необхідні для забезпечення нормального функціонування схеми.

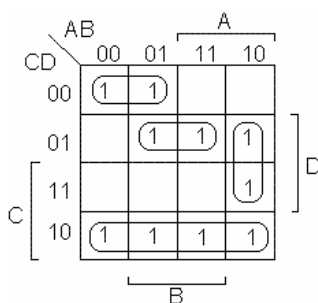
Моделювання схем на біполярних транзисторах виконується переходом від заданого логічного виразу до струмового графу, який його реалізує, і подальшою заміною ПСФЕ їх схемними варіантами. Згідно методу струмових графів синтез електричної схеми виконується наступним чином:

1) для заданої логічної функції знаходиться ряд мінімізованих форм представлення (табл. 2.3). Для кожного з отриманих мінімізованих виразів складається структурна схема з розділеними логічними операціями, а відповідно до неї - вихідний струмовий граф. При цьому включені СК реалізують операцію інверсії, заборони або імплікації. ОС реалізують диз'юнкцію або кон'юнкцію. РС реалізують розгалуження сигналу (розмноження логічних змінних). Вихідні графи містять ПСФЕ, необхідні для виконання логічних перетворень. Але їх недостатньо для електричного функціонування схеми.

Приклад. Побудувати схемну модель ТТЛ в базисі І-НІ на біполярних транзисторах, яка реалізує алгебраїчний вираз:

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

Мінімізуємо вираз за допомогою карти Карно:



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

Перетворимо вираз для побудови в базисі І-НІ:

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

Побудуємо структурну модель схеми з урахуванням розділу логічних операцій між елементами схеми (рис. 2.31).

Замінімо логічні вузли на ПСФЕ (рис. 2.32).

2) у струмові графи включаються ДС, необхідні для електричного функціонування схеми. При цьому спочатку перевіряється умова струмової сумісності сусідніх ПСФЕ, що забезпечує виконання закону Кірхгофа для струмів. Умова струмової сумісності виконується, якщо клас (к, z) і знак з'єднаних попереднього (к', z') і подальшого (к'', z'') ПСФЕ задовольняє логічному виразу «Виключаюче АБО»: $z' = z \oplus k'$

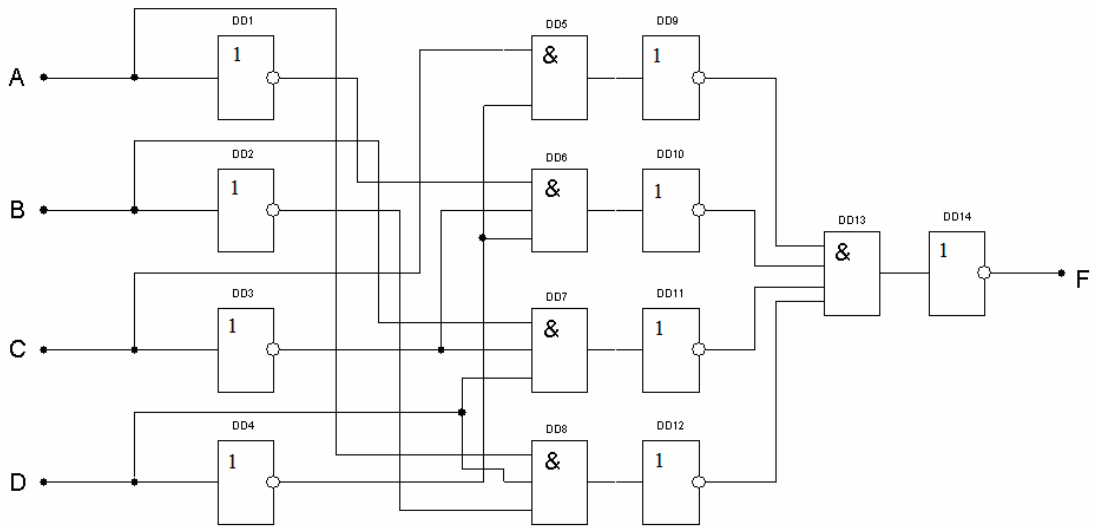


Рисунок 2.31 – Структурна модель схеми

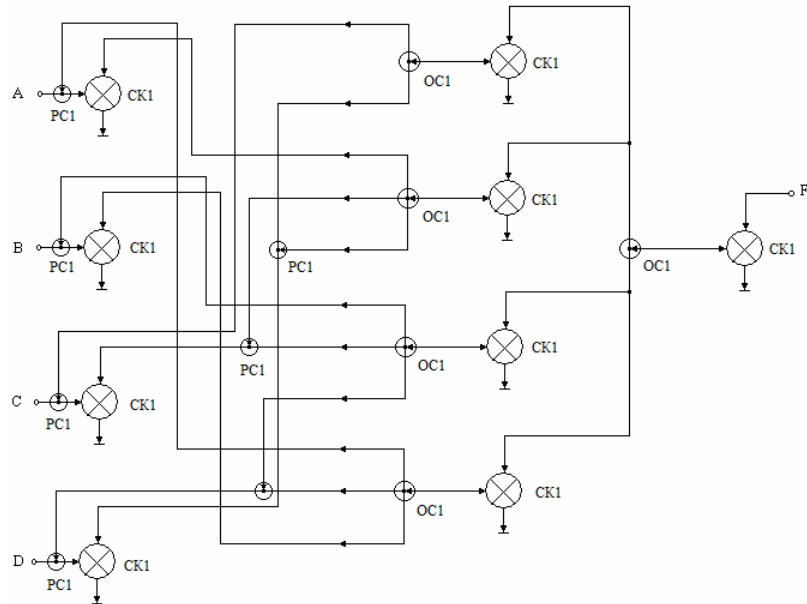


Рисунок 2.32 – Модель схеми на ПСФЕ

Якщо умова не задовольняється, то між ними включається елемент, який задовольняє умові «Виключаюче АБО» для попереднього і подальшого елементів.

Визначимо умови струмової сумісності між сусідніми елементами.

Попередній елемент:

OC1

$k = 0$

$z = 1$

Подальший елемент:

CK1

$k' = 1$

$z' = 1$

$$1 = 1 \oplus 1$$

Умова «Виключаюче АБО» не виконується. Отже, між даними елементами необхідно додатково поставити сумісний елемент. Поставимо ДС0 і перевіримо умови сумісності:

Попередній елемент:

ОС1

$$k = 0$$

$$z = 1$$

Подальший елемент:

ДС0

$$k' = 1$$

$$z' = 0$$

$$0 = 1 \oplus 1$$

Попередній елемент:

ДС0

$$k = 1$$

$$z = 0$$

Подальший елемент:

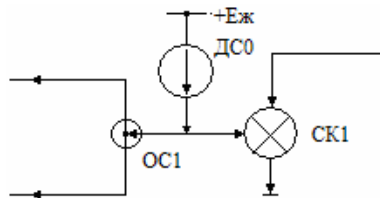
СК1

$$k' = 1$$

$$z' = 1$$

$$1 = 0 \oplus 1$$

Умова виконується. Решта даних кіл схемної моделі ідентична.



3) у отриманих струмових графах ПСФЕ замінюються їх можливими схемними реалізаціями відповідно до таблиці 2.4. В результаті для кожного графа виходить декілька варіантів електричних схем. Для цих схем перевіряється виконання потенціальних умов перемикання між сусідніми ключами:

$$U^0 \leq V_{\Pi} + \sum U_{\text{пад}}$$

$$U^1 \geq V_{\Pi} + \sum U_{\text{пад}},$$

де $V_{\Pi} = U_e + U^*$ - поріг перемикання подальшого СК; U^0, U^1 - потенціали на виході відкритого або закритого попереднього СК; $\sum U_{\text{пад}}$ - алгебраїчна сума падінь напруги на ПСФЕ, включених між даними ключами; U_e - потенціал емітера подальшого СК; $U^* = 0,7\text{В}$ - напруга перемикання подальшого СК.

Для виконання потенціальних умов перемикання при необхідності між сусідніми ключами включаються фіксатори потенціалу ФП.

Замінимо ПТФЕ їх схемною реалізацією (рис. 2.33).

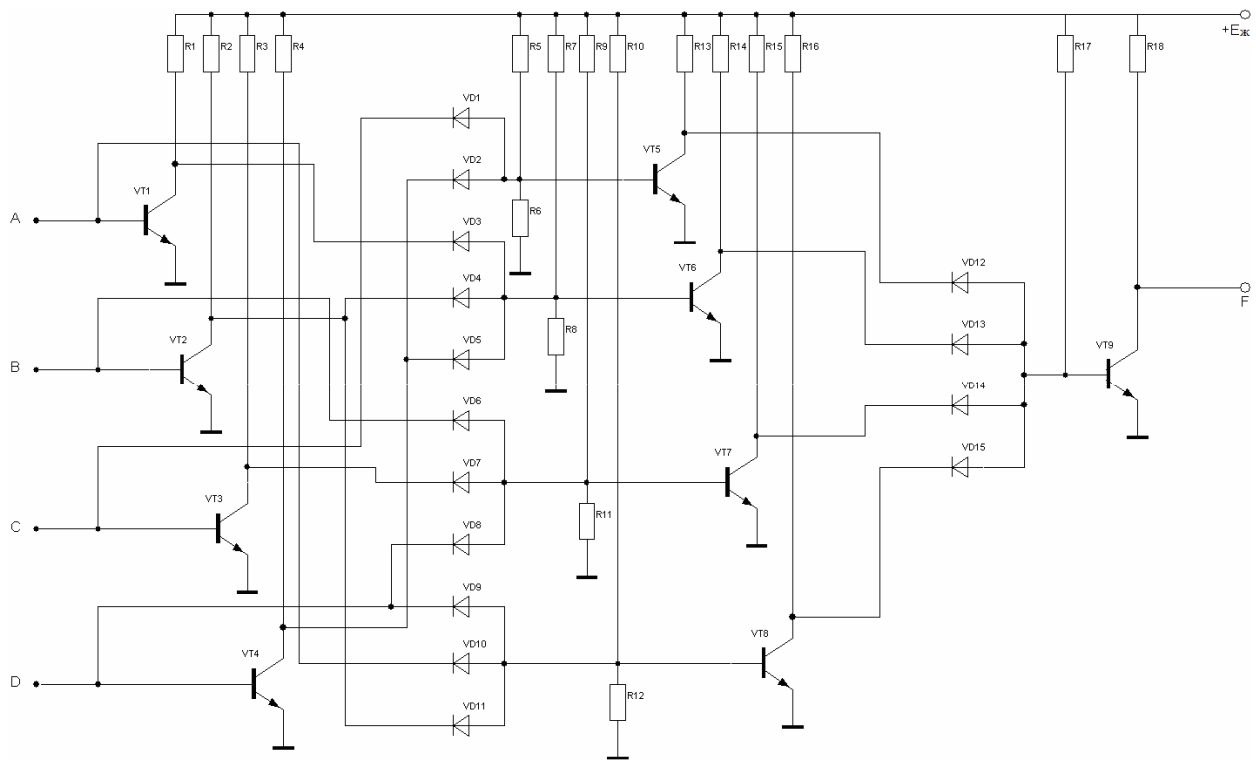


Рисунок 2.33 – Заміна ПСФЕ їх схемною реалізацією

Проведемо перевірку потенціальних умов між сусідніми ключами:

$$U^0 \leq V_{\pi} + \sum U_{\text{пад}} \leq U^* - U^* \leq 0,7 - 0,7 \leq 0$$

$$U^1 \geq V_{\pi} + \sum U_{\text{пад}} \geq U^* - U^* \geq 0,7 - 0,7 \geq 0$$

Для U^0 умови підходять, а для U^1 - ні. Отже, між даними ключами слід поставити фіксатор потенціалу (рис. 2.34).

Тоді:

$$U^0 \leq V_{\pi} + \sum U_{\text{пад}} \leq U^* + U^* - U^* \leq 0,7 + 0,7 - 0,7 \leq 0,7$$

$$U^1 \geq V_{\pi} + \sum U_{\text{пад}} \geq U^* + U^* - U^* \geq 0,7 + 0,7 - 0,7 \geq 0,7$$

Для виконання умови $U_{\text{вих.}} = E_{\text{жив}} - I_{\text{к}} R_{\text{к}}$ і забезпечення підвищення U^1 на базах СК в колекторні, базові і емітерні кола транзисторних СК включаються ФП, які забезпечують необхідні додаткові падіння напруги (рис. 2.35).

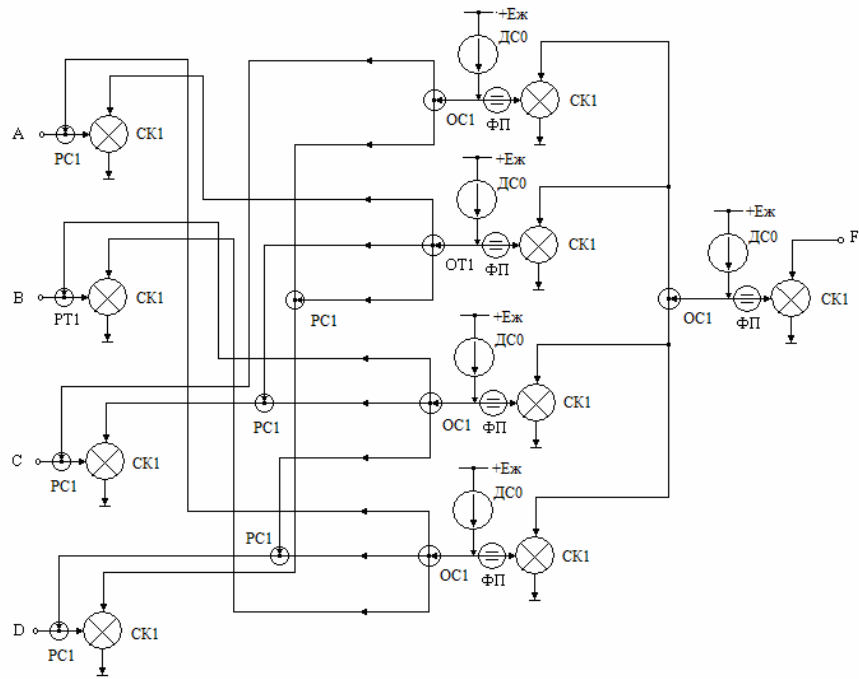


Рисунок 2.34 – Модель схеми на ПСФЕ з фіксаторами потенціалів

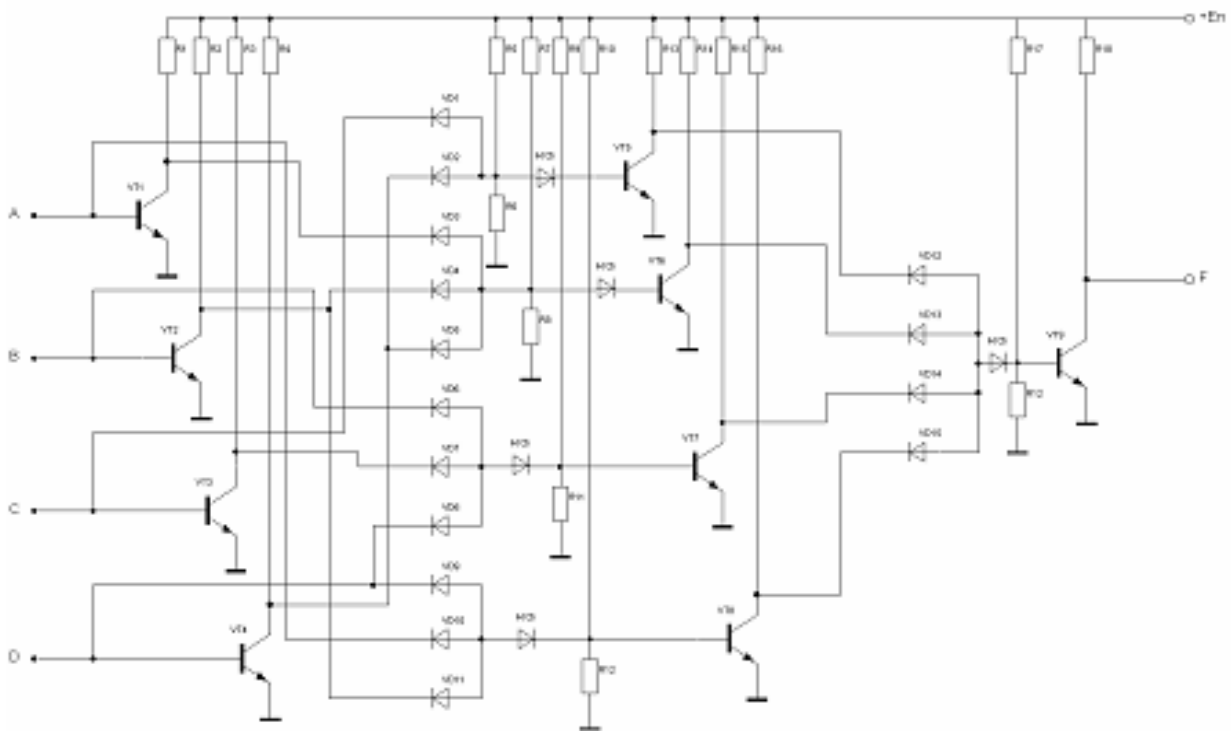


Рисунок 2.35 – Модель схеми з додатковими елементами

Для ключових транзисторів перевіряються також виконання засобів насичення:

$$S = B I_{\text{бн}} / I_{\text{кн}} \geq s_{\text{min}} > 1$$

$$s_{\min} = 1,2 \div 1,5$$

В разі необхідності в базове або колекторне кола включаються транзисторні ОТ або РТ, виконуючі посилення тока $I_{\text{бн}}$ або пониження тока $I_{\text{кн}}$.

4) проводиться аналіз можливостей фізичного поєднання компонентів схем в загальній області напівпровідника. За наявності таких можливостей виконується поєднання компонентів, в результаті яких можуть вийти нові варіанти схем з багатомітерними транзисторами (БЕТ), з інжекційним живленням та ін. Діодні з'єднання можна замінити БЕТ. Таким чином, отримуємо промодельований логічний пристрій (рис. 2.36).

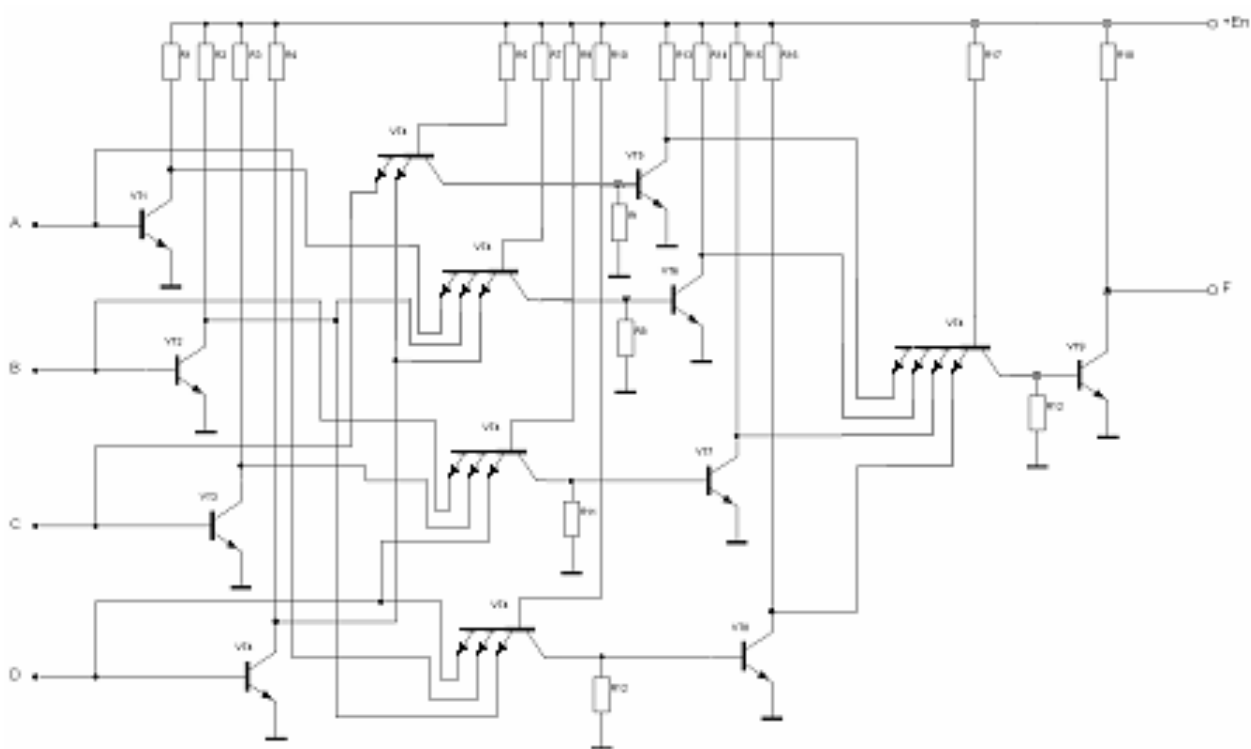


Рисунок 2.36 – Схемна модель ТТЛ в базисі І-НІ на біполярних транзисторах

5) виконується аналіз отриманої схеми за активною площею кристала, споживаною потужністю, затримкою перемикання.

Точний розрахунок електричних параметрів схем виконується на ЕОМ за допомогою сучасних програм аналізу. За результатами розрахунку проводиться остаточний вибір найкращої схеми.

Приклад. Побудувати схемну модель ЕЗЛ в базисі АБО - НІ на біполярних транзисторах, яка реалізує алгебраїчний вираз:

$$F1 = (A + B)(C + D) = \overline{\overline{(A + B)(C + D)}} = \overline{\overline{(A + B)} + \overline{\overline{(C + D)}}}$$

Базовий варіант схеми (рис. 2.18) складається з перемикача струму і вихідних емітерних повторювачів. Елемент має два виходи: інверсний, на якому реалізується функція $F_1 = \overline{(A + B)}$, та неінверсний, де реалізується $F_2 = A + B$. Таким чином, елемент виконує комбіновану функцію АБО – НІ / АБО.

При синтезі перемикач струму можна розглядати як різновид ТК, що має декілька інвертуючих виходів: колектори входних транзисторів VT2, VT3 і колектор опорного транзистора VT4. Об'єднуючи колектори перемикачів VT2 та VT3, отримуємо транзисторний об'єднувач ОС0 і реалізуємо операцію диз'юнкції. Таким чином, при розробці схем ЕЗЛ також зручно використовувати їх узагальнену форму у вигляді струмових графів. При цьому методика має ряд особливостей. Мінімізовану логічну функцію слід представити у вигляді МКНФ або МДНФ. Число перемикачів струму, потрібних для реалізації МДНФ функції, дорівнює числу кон'юнктивних членів в логічному виразі. Кожен кон'юнктивний член утворюється за допомогою об'єднання інвертуючих виходів одного з перемикачів струму на загальному ФП. Потім об'єднані виходи перемикачів підключаються до вихідного транзисторного ОС0, який реалізує диз'юнкцію кон'юнктивних членів відповідно до даного логічного вираження.

1) побудуємо структурну модель функції (рис. 2.37):

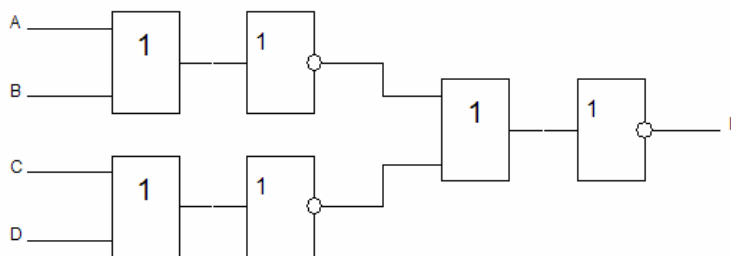


Рисунок 2.37 – Структурна модель функції

2) здійснимо заміну логічних вузлів на ПСФЕ (рис. 2.38):

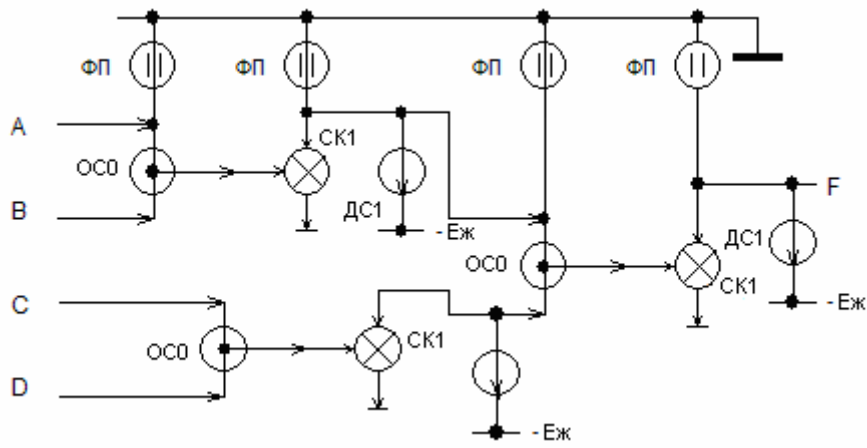


Рисунок 2.38 – Заміна логічних вузлів на ПСФЕ

3) проведемо заміну ПСФЕ їх схемною реалізацією (рис. 2.39):

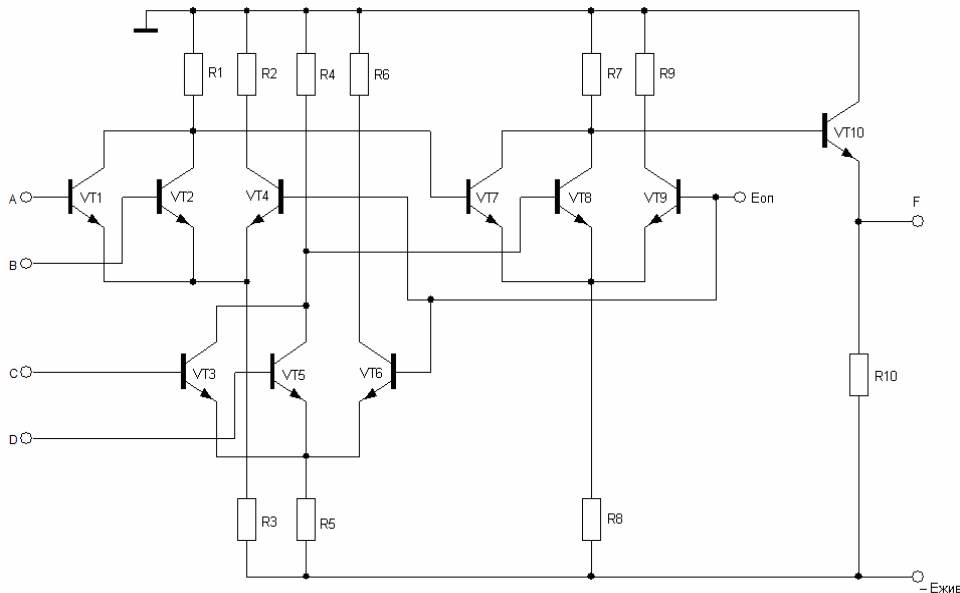


Рисунок 2.39 – Схемна модель ЕЗЛ в базисі АБО - НІ на біполярних транзисторах

Оскільки логічний перепад невеликий, то нестабільність напруги джерела живлення істотно впливає на завадостійкість ЕЗЛ. Для підвищення завадостійкості в схемах ЕЗЛ проводять заземлення не негативного полюсу джерела живлення, а позитивного. Це робиться для того, щоб значна частина напруги перешкоди падала на великих опорах R3, R5, R8 і лише мала її доля потрапляла на входи схеми.

Контрольні запитання до другого розділу

1. Обґрунтувати завадостійкість логічних елементів.
2. Пояснити функціонування елементів ДТЛ логіки.
3. Пояснити функціонування елементів ТТЛ логіки.
4. Пояснити функціонування елементів І²Л логіки.
5. Пояснити функціонування елементів ЕСЛ логіки.
6. Пояснити функціонування елементів МДН логіки.
7. Пояснити функціонування елементів КМДН логіки.
8. Виконати моделювання заданої схеми на МОН транзисторах.
9. Пояснити моделювання схем методом струмових графів.
10. Виконати моделювання заданої схеми на біполярних транзисторах.

До них відносяться пристрої, вихід яких в окремий момент часу однозначно визначається станом входу в той же момент часу

$$y^{(t)} = f(x^{(t)}).$$

3.1 Принцип побудови інтегрального суматора

Дискретна техніка оперує класом пристроїв, призначення яких полягає у виконанні арифметичних дій з двійковими числами. Особливість арифметичних пристроїв полягає в тому, що сигналам приписуються не логічні, а арифметичні значення 1 і 0 і дії над ними підлягають законам двійкової арифметики. Для опису їх роботи також зручно користуватися таблицями істинності. Найважливіша з арифметичних операцій – складання (підсумовування). Окрім прямого призначення вона використовується і при інших операціях: віднімання – це складання, в якому від'ємник перетворюється в зворотний, а потім в додатковий код, а множення і ділення – це послідовне складання і віднімання. До арифметичних пристроїв відносять також вузли, які виконують спеціальні операції: порівняння двох чисел і виявлення парності заданих чисел (визначення паритету).

Суматори призначені для арифметичного складання двох чисел. З принципу складання багаторозрядних двійкових чисел виходить, що в кожному i -розряді знаходиться сума S трьох чисел по модулю два: доданків A_i , B_i і перенесення, яке надійшло із молодшого розряду P_i , та формується сигнал перенесення в старший розряд P_{i+1} .

Суматори за принципом дії підрозділяються на комбінаційні і накопичувальні. Розрізняють суматори паралельної дії і послідовної дії.

Проаналізуємо таблицю істинності однорозрядного суматора (таблиця 4.1) і запишемо логічні вирази для вихідних величин.

$$S = \overline{A} \overline{B} P_i + \overline{A} B \overline{P}_i + \overline{A} B P_i + A B \overline{P}_i;$$

$$P_{i+1} = A B \overline{P}_i + \overline{A} B P_i + \overline{A} B P_i + A B P_i.$$

Таблиця 3.1 – Таблиця функціонування однорозрядного суматора

Вхід			Вихід	
Доданки		Перенесення	Сума	Перенесення
A	B	P_i	S	P_{i+1}
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

По цих функціях можна побудувати суматор на елементах І, АБО (рис. 3.1).

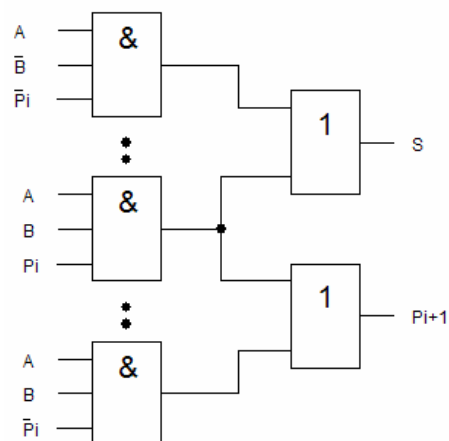


Рисунок 3.1 – Суматор на елементах І – АБО

У цифровій обчислювальній техніці використовуються однорозрядні схеми підсумовування 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 3.2 приведена таблиця істинності напівсуматора, на підставі якої складена його структурна формула у вигляді ДДНФ.

Таблиця 3.2 – 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

$$S = \bar{A}B + A\bar{B}$$

$$P_{i+1} = AB$$

Функціональна схема, складена на елементах основного базису, відповідно до цієї структурної формули, приведена на рисунку 3.3.

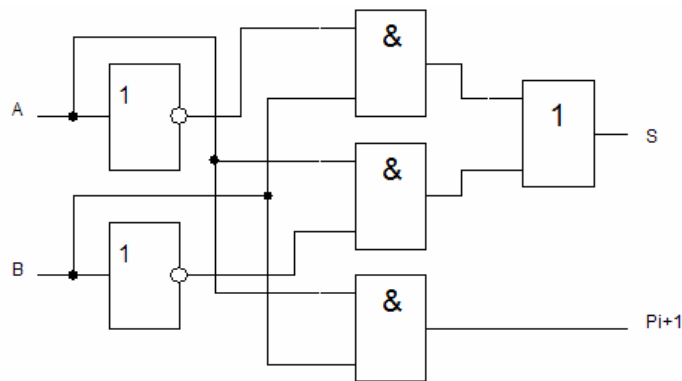


Рисунок 3.3 – Функціональна схема напівсуматора на елементах основного базису

Основними параметрами, які характеризують якісні показники логічних схем, є швидкодія і кількість елементів, що визначає складність схеми. Швидкодія визначається сумарним часом затримки сигналу при проходженні елементів схеми. У приведеній вище схемі швидкодія визначається затримкою в трьох логічних елементах. Окрім кількості елементів, складність схеми визначається кількістю входів елементів, по яких виконуються логічні операції (число по Квайну). Приведена вище схема містить 6 елементів і має 10 входів (число по Квайну дорівнює 10).

Недоліком схеми (рис. 3.3) є те, що на її входи необхідно подавати і прямі, і інверсні значення операндів. Застосовуючи закони алгебри логіки, схему можна перетворити, виключивши інверсії над окремими операндами (рис. 3.4):

$$S = \overline{A}B + A\overline{B} = \overline{\overline{A}B + A\overline{B}} = \overline{\overline{A}B} \cdot \overline{A\overline{B}} = (A + B)(\overline{A} + \overline{B}) = \overline{A\overline{A}} + \overline{A\overline{B}} + \overline{BA} + \overline{BB} =$$

$$= 0 + \overline{A\overline{B}} + \overline{BA} + 0 = \overline{A\overline{B}} + \overline{BA} = \overline{A\overline{B}} \cdot \overline{BA} = \overline{A\overline{B}} \cdot \overline{B}A = \overline{A\overline{B}} \cdot (A + B);$$

$$P_{i+1} = AB.$$

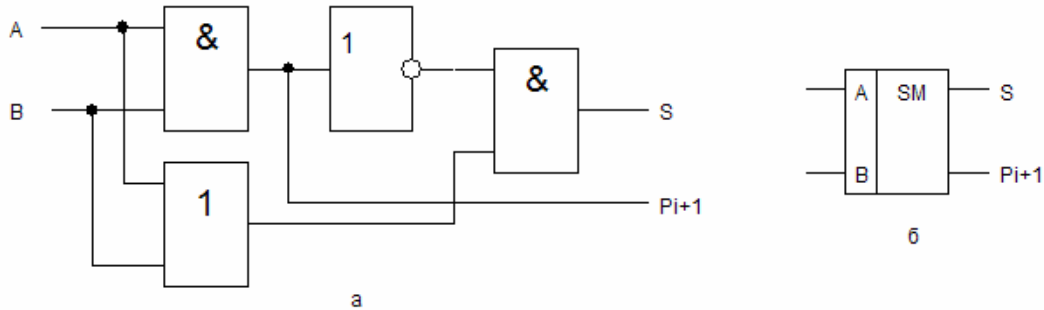


Рисунок 3.4 – Мінімізована схема (а) і умовне графічне позначення (б) напівсуматора на елементах основного базису

Мінімізована схема є більш швидкодіючою, оскільки замість 6, містить 4 елементи, а число по Квайну зменшилося з 10 до 7.

Логічний елемент “Виключаюче АБО” застосовується як суматор по модулю 2. На рисунку 3.5 приведений символ елемента без інверсії і його таблиця стану. Вихідний сигнал елемента відповідає логічному рівнянню:

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

Тут “ \oplus ” – символ складання по модулю 2. Нижній і верхній рядки таблиці відображають еквівалентність входних рівнів, тобто $A = B = 0$ (у верхньому рядку) і $A = B = 1$ (у нижньому). У випадку $A = B = 0$ вихідний сигнал $F = 0$ (це природний) тривіальний нуль. Коли $A = B = 1$ вихідний сигнал F також дорівнює нулю, хоча на двох входах A і B присутні одиниці.



Рисунок 3.5 – Елемент “Виключаюче АБО” і його таблиця функціонування

Якщо до елемента “Виключаюче АБО” додати елемент, який є формувачем одиниці старшого розряду (генератор перенесення, який створює вихід P), то отримаємо однорозрядний напівсуматор (рис. 3.6)

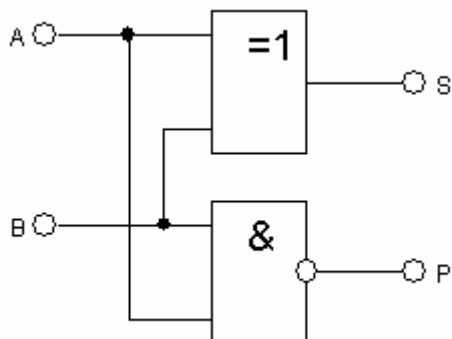


Рисунок 3.6 – Схема напівсуматора на елементі “Виключаюче АБО”

Схема дає при $A = B = 1$ результат $S = 0$ (це молодший розряд суми) і $P = 1$ (старший розряд, який тут має назву одиниці перенесення). У цьому випадку на обох виходах напівсуматора міститься двохрозрядне двійкове вихідне слово $A + B = 01 + 01 = 10_{(2)}$. Його десятковий еквівалент $1 + 1 = 2_{(10)}$.

Схему повного однорозрядного суматора можна отримати на основі двох схем напівсуматорів і схеми «АБО», як показано на рисунку 3.7.

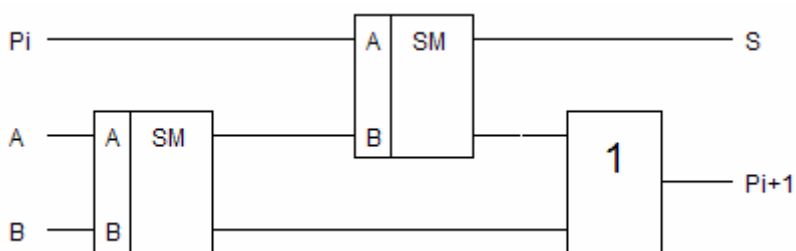


Рисунок 3.7 – Схема повного однорозрядного суматора

3.1.1 Багаторозрядні суматори

Методи підсумовування багаторозрядних чисел:

- послідовне підсумовування;
- паралельне підсумовування з послідовним перенесенням;
- паралельне підсумовування з паралельним перенесенням.

При послідовному підсумовуванні використовується один суматор, загальний для всіх розрядів (рис. 3.8).

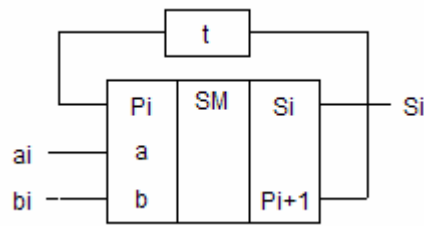


Рисунок 3.8 – Послідовне підсумовування багаторозрядних чисел

Операнди повинні вводитися в суматор через входи a_i і b_i синхронно, починаючи з молодших розрядів. Коло затримки забезпечує зберігання імпульсу перенесення P_{i+1} на час одного такту, тобто до приходу пари доданків наступного розряду, з якими він буде підсумований. Затримку виконує D-тригер. Результати підсумовування також прочитуються послідовно, починаючи з молодших розрядів. Для зберігання і введення операндів на входи суматора, а також для запису результату підсумовування, зазвичай, використовуються регістри зрушення. Перевага цього методу – малі апаратні витрати. Недолік – невисока швидкодія, оскільки одночасно підсумовується лише пара доданків.

Схема паралельного суматора з послідовним перенесенням приведена на рисунку 3.9.

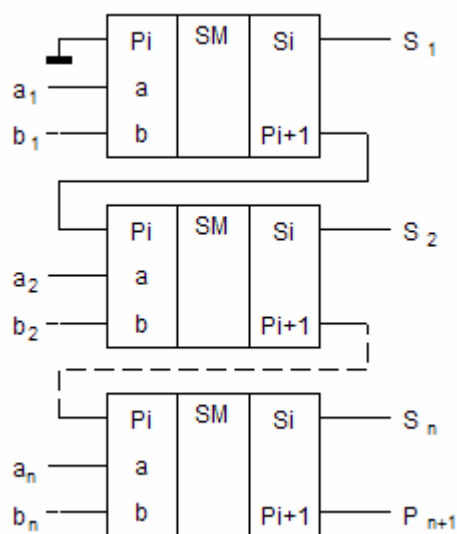


Рисунок 3.9 – Паралельне підсумовування багаторозрядних чисел з послідовним перенесенням

Кількість суматорів дорівнює числу розрядів чисел. Вихід перенесення P_{i+1} кожного суматора з'єднується з входом перенесення P_i наступного більш старшого розряду. На вході перенесення молодшого розряду встановлюється потенціал «0», оскільки сигнал перенесення сюди не поступає. Доданки a_i і b_i підсумовуються у всіх розрядах одночасно, а перенесення P_{i+1} поступає із закінченням операції складання в попередньому розряді. Швидкодія таких суматорів обмежена затримкою перенесення, оскільки формування перенесення на виході старшого розряду не може статися до тих пір, доки сигнал перенесення не пошириться по всьому колу суматорів. При побудові паралельних суматорів з паралельним перенесенням застосовуються спеціальні вузли – блоки прискореного перенесення. Принцип прискореного перенесення полягає в тому, що для кожного двійкового розряду додатково знаходяться два сигнали:

G – утворення перенесення і H – поширення перенесення.

$$G_i = a_i \cdot b_i$$

$$H_i = a_i + b_i$$

В разі $G_i = 1$, тобто $a_i = b_i = 1$, в даному i -розряді формується сигнал перенесення $P_i = 1$ в наступний вищий розряд незалежно від формування функцій суми в попередніх розрядах.

Якщо хоч би один з доданків a_i або b_i дорівнює «1», тобто $H_i = 1$, то перенесення в подальший розряд проводиться за наявності сигналу перенесення з попереднього розряду.

Якщо $H_i = H_{i-1} = 1$ і при цьому існує сигнал перенесення P_i з попереднього в i -розряд, то перенесення проводиться відразу в $i-2$ розряд.

У загальному випадку процес формування прискореного перенесення описується наступним рівнянням:

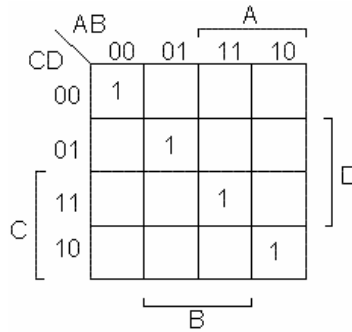
$$P_{i+1} = G_i + H_i \cdot G_{i+1} + H_i \cdot H_{i-1} \cdot G_{i-2} + \dots + H_i \cdot H_{i-1} \cdot \dots \cdot H_2 \cdot H_1 \cdot P_1$$

Блоки прискореного перенесення випускаються в інтегральному виконанні у вигляді окремих мікросхем, або безпосередньо з схемою суматора, або арифметико-логічного пристрою в одній мікросхемі.

3.2 Схеми порівняння кодів

Два кода X і Y вважаються рівними, якщо попарно рівні їх однойменні розряди. Функція $F(X = Y)$ дорівнює 1, якщо $x_i = y_i$ для всіх i , інакше її значення дорівнює нулю. Як приклад візьмемо два двохибітових числа $X = (A B)$ і $Y = (C D)$ (табл. 3.3).

Карта Карно для цих чисел:



Таблиця 3.2 – Таблиця функціонування схеми порівняння кодів

Входи				Вихід
Число X		Число Y		F (X = Y)
A	B	C	D	
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Рівняння функціонування пристрою порівняння кодів:

$$F(X = Y) = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}BC\overline{D} = \overline{B}\overline{D}(\overline{A}\overline{C} + AC) + B\overline{D}(\overline{A}\overline{C} + AC) = (\overline{A}\overline{C} + AC) \cdot (\overline{B}\overline{D} + B\overline{D}) = \overline{(A \oplus C)} \cdot \overline{(B \oplus D)} = \overline{(A \oplus C) + (B \oplus D)}$$

Перетворення в останніх двох рівняннях зроблені з урахуванням того, що $\overline{F_6(x,y)} = F_9(x,y)$, і навпаки (табл. 1.5). Одна з можливих схем реалізацій приведена на рисунку 3.10.

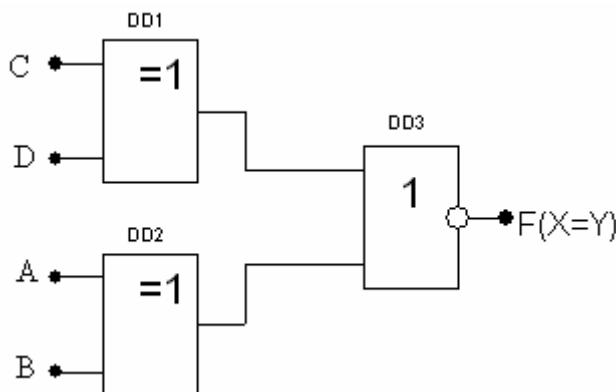


Рисунок 3.10 – Схема порівняння кодів

Практичні схеми доповнюються функціями "більше/менше" (табл. 3.3).

Таблиця 3.3 – Таблиця функціонування повного пристрою порівняння кодів

Входи				Виходи		
Число X		Число Y		F (X = Y)	F (X > Y)	F (X < Y)
A	B	C	D			
0	0	0	0	1	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	0	1	0
1	1	1	1	1	0	0

Рівняння функціонування:

$$F(X>Y) = \overline{A}BCD + \overline{A}BC\overline{D} + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} = \overline{A}C + \overline{B}CD + \overline{A}B\overline{D}$$

$$F(X<Y) = \overline{A}BCD + \overline{A}BC\overline{D} + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D} + \overline{A}BC\overline{D} + \overline{A}BC\overline{D} = \overline{A}C + \overline{B}CD + \overline{A}B\overline{D}$$

Спрощення за допомогою карт Карно:

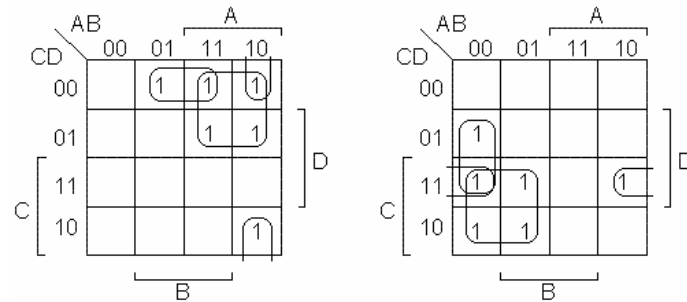


Схема пристрою порівняння кодів представлена на рисунку 3.11.

На рисунку 3.12 показано з'єднання мікросхем 555СП1 (схема порівняння двох чотирьохрозрядних чисел), для збільшення розрядності порівнюваних чисел до восьми.

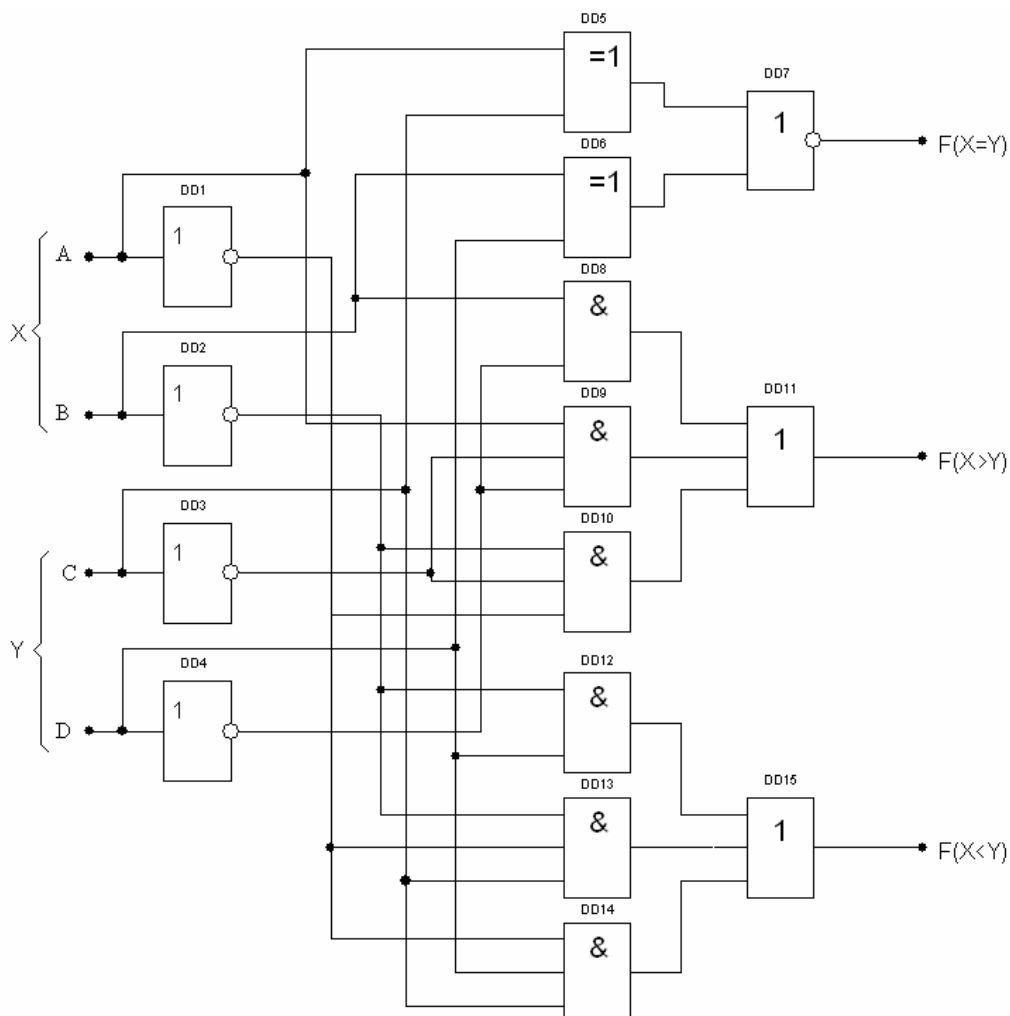


Рисунок 3.11 – Схема повного пристрою порівняння кодів

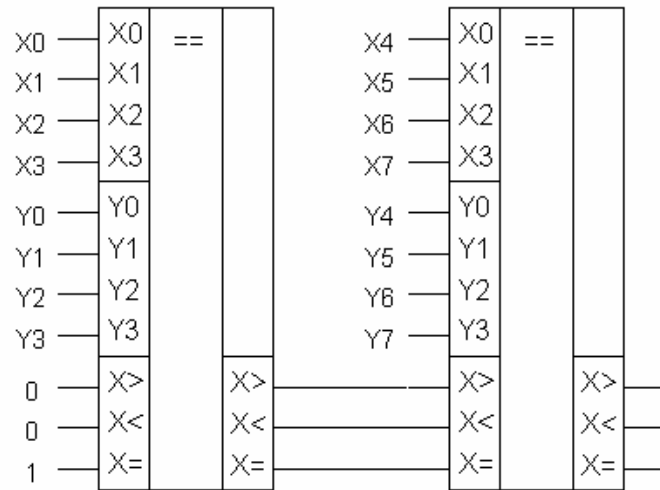


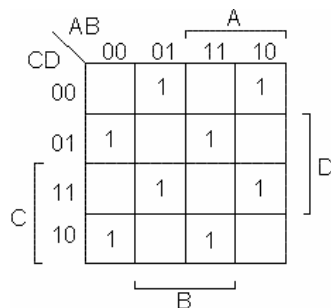
Рисунок 3.12 – Схема порівняння багаторозрядних чисел

Для правильного результату порівняння чисел $X=(X_7, X_6, \dots, X_0)$ і $Y=(Y_7, Y_6, \dots, Y_0)$ на вхід «X =» необхідно подати 1. Схема порівняння входить до складу АЛП мікропроцесора.

3.3 Схема контролю парності (непарності)

Схема застосовується для виявлення одиничних помилок, викликаних перешкодами в лінії зв'язку або в блоках пам'яті. Метод заснований на підрахунку числа одиниць в порції інформації, що передається в лінію або направляється в пам'ять на зберігання. Причому, якщо число одиниць парне - функція парності P (Parity) дорівнює нулю (табл. 3.4).

Карта Карно згідно з таблицею станів:



Рівняння функціонування схеми контролю парності:

$$\begin{aligned}
 P &= \overline{A}\overline{B}C\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BCD + A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D} + AB\overline{C}\overline{D} + ABC\overline{D} = \\
 &= \overline{A}\overline{B}(\overline{C}D + CD) + \overline{A}B(\overline{C}\overline{D} + C\overline{D}) + A\overline{B}(\overline{C}\overline{D} + CD) + AB(\overline{C}\overline{D} + CD) =
 \end{aligned}$$

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

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

Таблиця 3.4 - Таблиця станів схеми контролю парності для чотирьохрозрядного двійкового числа

Входи				Вихід
Число X		Число Y		P
A	B	C	D	
0	0	0	0	X
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Схемна реалізація і умовне позначення приведені на рисунку 3.13.

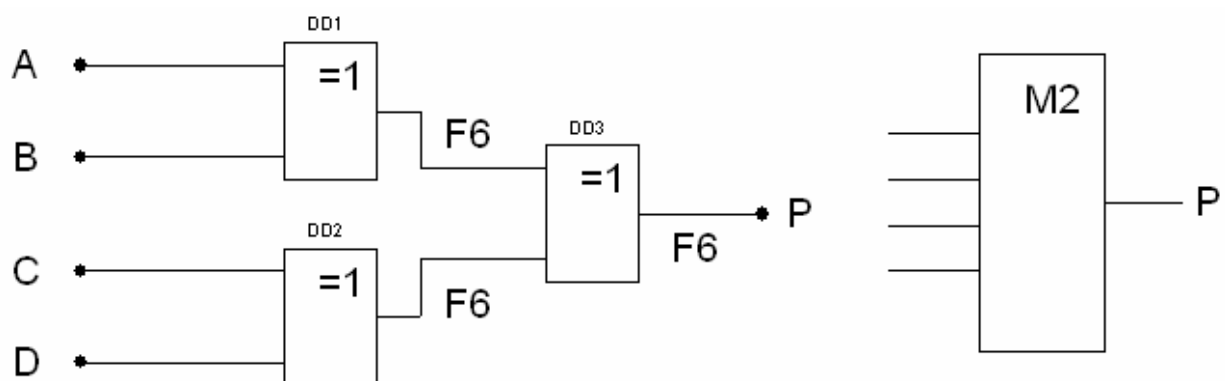


Рисунок 3.13 – Схема контролю парності

Символом M2 позначена операція - "сума по модулю два".

Розглянемо приклад.

Нехай по n -провідній лінії зв'язку (рис. 3.14) передається паралельний двійковий код $x(n-1), x(n-2), \dots, x_1, x_0$, а приймається код $x'(n-1), x'(n-2), \dots, x'_1, x'_0$.

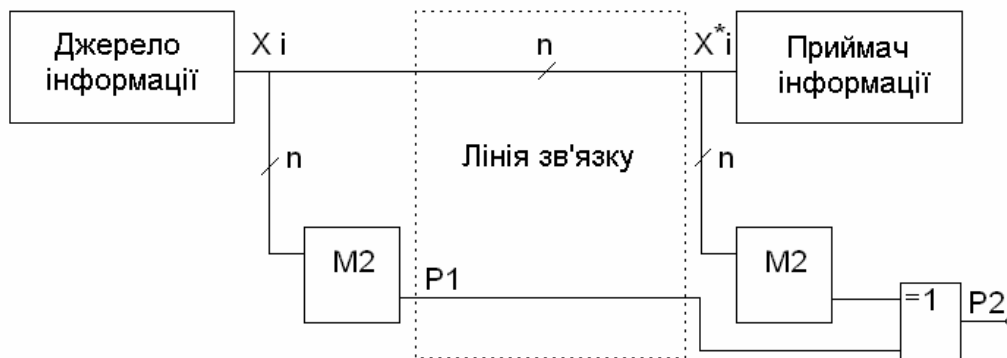


Рисунок 3.14 – Контроль парності в n -провідній лінії зв'язку

Тоді величина $P1 = x_0 \oplus x_1 \oplus \dots \oplus x(n-1)$. На приймальному кінці лінії зв'язку $P2 = x'_0 \oplus x'_1 \oplus \dots \oplus x'(n-1) \oplus P1$. Підставляючи в останню формулу вираз для $P1$ і групуючи змінні в однойменні пари, отримаємо: $P2 = (x_0 \oplus x'_0) \oplus (x_1 \oplus x'_1) \oplus (x_2 \oplus x'_2) \oplus \dots$ З останнього виразу виходить, що якщо передача пройшла без спотворень, то $x_i = x'_i$ та $x_i \oplus x'_i = 0$, а $P2 = 0$. При спотворенні одного i в загальному випадку непарного числа біт функція $P2 = 1$. Аналогічно протікає процес контролю і при послідовній передачі по одній лінії зв'язку n -біт інформаційних сигналів і одного біта парності.

3.4 Мажоритарні елементи

Мажоритарний закон це «Закон більшості». Вирішальний елемент зазвичай називають мажоритарним елементом. Мажоритарний елемент – це логічний пристрій з непарним числом входів $m = 2k + 1$ (де $k = 1, 2, 3, \dots$) і одним виходом. Робота мажоритарного елемента полягає в наступному: на входи елемента поступають двійкові сигнали від непарної кількості ідентичних елементів. Вихідний сигнал елемента набуває значення, рівного значенню, яке приймає більшість вхідних сигналів. Найчастіше використовують мажоритарні елементи, які працюють за законом «2 з 3» (табл. 3.5). У цих елементах значення вихідно-

го сигналу дорівнює значенню двох однакових вхідних сигналів. Крім того, відомі мажоритарні елементи, що працюють за законом «3 з 5», «4 з 7» і так далі.

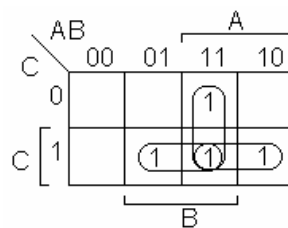
Таблиця 3.5 - Таблиця функціонування мажоритарного елемента

Входи			Вихід
A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Рівняння функціонування:

$$F = \overline{A}BC + A\overline{B}C + ABC\overline{C} + ABC$$

Спростимо вираз за допомогою карти Карно:



$$F = AB + AC + BC$$

Схема мажоритарного елемента, який працює за законом «2 з 3» і побудованого з логічних елементів І і АБО має вигляд, представлений на рисунку 3.15, а діаграма функціонування в динамічному режимі на рисунку 3.16

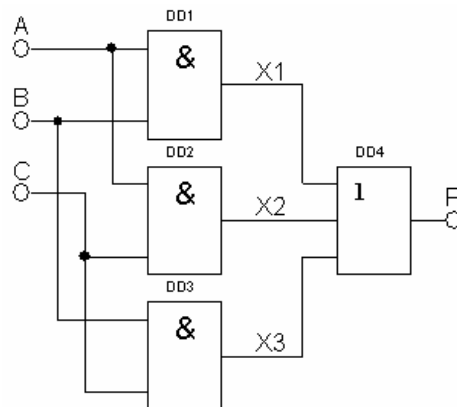


Рисунок 3.15 – Схема мажоритарного елемента

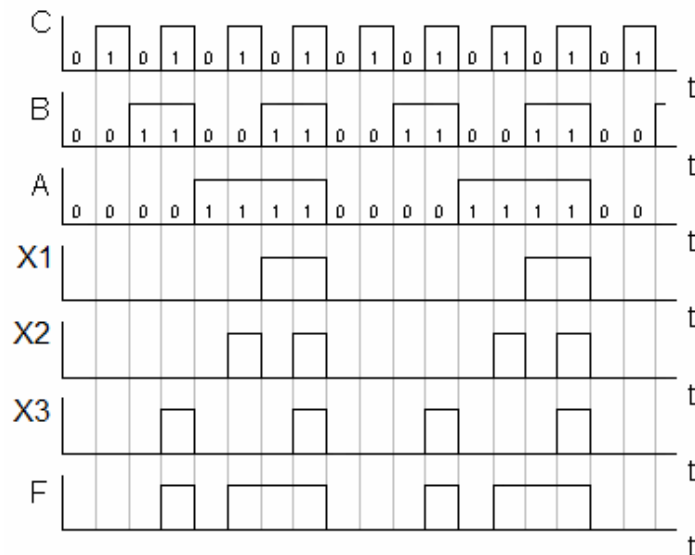


Рисунок 3.16 - Діаграма функціонування мажоритарного елемента в динамічному режимі

Мажоритарний елемент дозволяє забезпечити режим одночасного штатного функціонування основного і резервних елементів мікропроцесорних пристроїв і виключає використання спеціальних комутаційних вузлів, які усувають взаємний вплив основного і резервних елементів один на одне. При цьому відмова основного або резервного елементів не впливає на роботу справних елементів, що залишилися. В даний час знайшло значно широкое використання структурне резервування з мажоритарним елементом. Воно використовується для підвищення надійності цифрових електронних пристроїв і цифрових систем.

За способом включення резервних елементів функціональних пристроїв розрізняють три види резервування: постійне, заміщенням і ковзаюче.

При постійному резервуванні передбачають, що будь-який елемент, що відмовив, або вузол, не впливають на вихідні сигнали і тому їх прямого виявлення не виконується. Постійне резервування найбільш поширене в пристроях, які не відновлюються. Крім того, воно є єдино можливим в пристроях, де недопустима навіть короткочасна перерва в роботі. Мажоритарний елемент може бути виконаний у вигляді окремої мікросхеми або зібраний з декількох логіч-

них мікросхем. Принцип мажоритарного резервування пояснимо на прикладі за допомогою рисунка 3.17, який відображає частину деякої цифрової схеми.

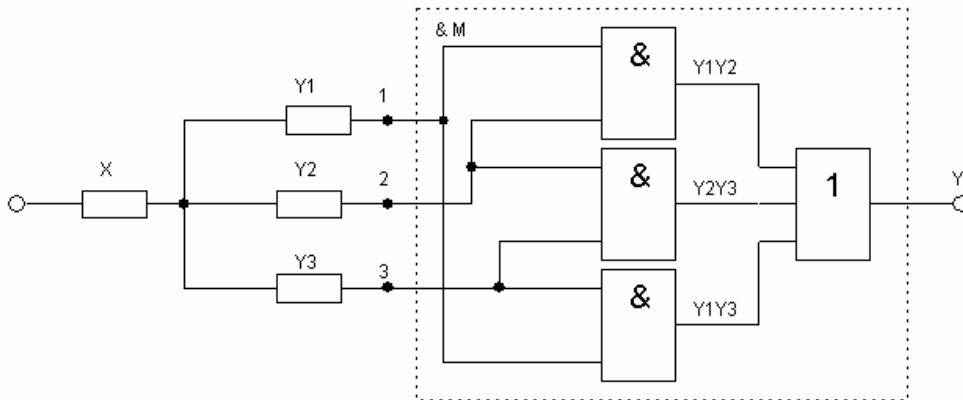


Рисунок 3.17 – Принцип мажоритарного резервування

З виходу цифрового пристрою X цифрові сигнали у вигляді послідовності символів «0» і «1» поступають на входи трьох одночасно працюючих однакових рівнонадійних пристроїв Y1, Y2, Y3, що створюють резервований вузол. Цифрові сигнали з виходу кожного з пристроїв Y1, Y2, Y3 поступають на відповідний вхід мажоритарного елемента (в даному випадку трьохвходового). Якщо кожен з пристроїв Y1, Y2, Y3 справний, то в даний момент часу на їх виходах буде один і той же двійковий символ (0 або 1), а значить і на входах мажоритарного елемента. Тоді і на виході мажоритарного елемента буде такий же двійковий сигнал. Якщо який-небудь з пристроїв Y1, Y2, Y3 відмовив, то лише на двох входах мажоритарного елемента в даний момент часу двійкові символи будуть однакові. На виході ж мажоритарного елемента буде двійковий символ, який співпадає з символом на виході двох справних пристроїв. Тобто, мажоритарний елемент виконує логічну операцію синхронізації рішення «по більшості» (операція «голосування»). Тепер стає зрозумілим, чому кількість входів в мажоритарному елементі має бути непарною і більше одиниці.

Постійне резервування вводиться або за допомогою блоку прийняття рішень (вирішувального блоку), або у вигляді однотипних елементів, або блоків, включених послідовно (паралельно), або, наприклад, згідно законів k-кратної логіки.

У якості вирішувального блоку можна використовувати мажоритарні елементи з постійними або змінними вагами, кодуючі-декодуючі пристрої і схеми з логічних елементів І, АБО, НІ.

Резервування заміщенням передбачає виявлення елемента або вузла, що відмовив, і підключення справного. Заміщення може відбуватися або автоматично, або вручну.

Резервування заміщенням має наступні переваги. Для багатьох схем при включенні резервного устаткування не потрібно додатково регулювати вихідні параметри, внаслідок того, що електричні режими в схемі не змінюються. Резервна апаратура до моменту включення в роботу знеструмлена, що підвищує загальну надійність системи за рахунок збереження ресурсу електронних пристроїв. Є можливість використання одного резервного елемента на декілька робочих.

Внаслідок складності апаратури для автоматичного включення резерву, резервування заміщенням доцільно застосовувати до великих блоків і окремих функціональних частин РЕА.

При ковзаючому резервуванні будь-який резервний елемент може заміщати будь-який основний елемент. Для здійснення цього резервування необхідно мати пристрій, який автоматично знаходить несправний елемент і підключає замість нього резервний. Переваги такого резервування в тому, що при ідеальному автоматичному пристрої буде найбільший виграш в надійності, в порівнянні з іншими методами резервування. Проте здійснення ковзаючого резервування можливе лише при однотипності елементів.

При мажоритарному контролі використовують декілька пристроїв, які одночасно виконують одні й ті ж дії. Рішення про те, який сигнал має бути на виході, приймається методом “голосування”, тобто по більшості вихідних сигналів окремих пристроїв. Генерування загального вихідного сигналу здійснюється мажоритарним елементом (елементом голосування).

3.5 Цифрова схема включення і виключення з декількох місць

Вихідний стан схеми включення і виключення з декількох місць змінюється лише в разі, якщо змінюється стан одного з входів. Якщо обидва входи змінюють свій стан, то вихідний стан не змінюється. Схема має два входи А і В і один вихід Z (рис. 3.18).

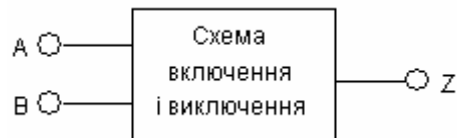


Рисунок 3.18 – Блок схема пристрою включення і виключення з декількох місць

Таблиця істинності схеми з двома входними змінними має 4 варіанта (табл. 3.6).

Таблиця 3.6 – Таблиця функціонування схеми включення і виключення з декількох місць

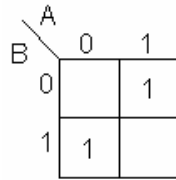
Варіанти	Входи		Вихід
	А	В	Z
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

Вихідний стан Z для першого варіанту може встановлюватися будь-яким чином. Вибране $Z = 0$. При переході від варіанту 1 до варіанту 2 змінна В змінює свій стан. Змінна А стан не змінює. Якщо лише один з входів змінює стан, то, згідно визначення, вихід Z повинен поміняти свій стан. Z має дорівнювати 1. При переході від варіанту 2 до варіанту 3 змінні А і В змінюють свої стани. Z не повинне змінитися. При переході від варіанту 3 до варіанту 4 змінна В змінює свій стан з 0 на 1. А залишається в стані 1. Таким чином, Z повинне поміняти стан з 1 на 0. Таблиця істинності могла б виглядати інакше, якби у варіанті 1 було обрано $Z = 1$.

Рівняння функціонування:

$$Z = (A\bar{B}) + (\bar{A}B).$$

Якщо нанести функцію на карту Карно, то видно, що подальше спрощення неможливе.



Побудуємо схему на елементах І-НІ:

$$Z = (A\bar{B}) + (\bar{A}B) = \overline{\overline{A\bar{B}} \cdot \overline{\bar{A}B}} = \overline{(\overline{A\bar{B}}) \cdot (\overline{\bar{A}B})}.$$

Схема, побудована згідно перетвореному рівнянню, показана на рисунку 3.19.

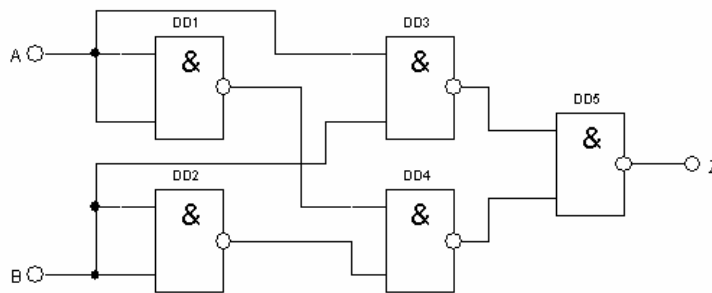


Рисунок 3.19 – Цифрова схема включення і виключення з декількох місць

3.6 Порогова логічна схема

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

Вхідні змінні A, B, C, D, E. Вихідна змінна – Z. Спочатку потрібно визначити таблицю істинності. При п'яти змінних величинах можливі 32 варіанти (табл. 3.7). Досконала диз'юнктивна нормальна форма складається з шести повних кон'юнкцій. ДДНФ спрощується за допомогою карти Карно (рис. 3.20).

Таблиця 3.7 - Таблиця істинності порогової логічної схеми

Входи					Вихід
A	B	C	D	E	Z
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	0
1	0	1	0	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	0
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

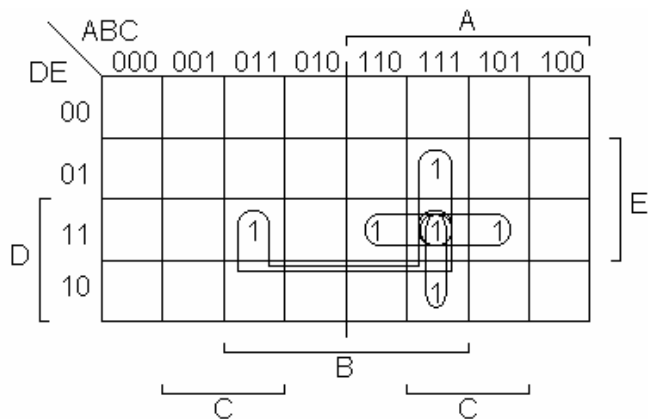


Рисунок 3.20 – Карта Карно порогової логічної схеми

$$F = \overline{A}BCDE + A\overline{B}CDE + ABC\overline{D}E + ABCD\overline{E} + ABCDE$$

Можна утворити 5 подвійних груп. Виходить наступна спрощена логічна функція:

$$F = BCDE + ACDE + ABDE + ABCE + ABCD.$$

У базисі І-НІ:

$$\begin{aligned} F &= \overline{\overline{BCDE + ACDE + ABDE + ABCE + ABCD}} = \\ &= \overline{\overline{BCDE} \cdot \overline{ACDE} \cdot \overline{ABDE} \cdot \overline{ABCE} \cdot \overline{ABCD}} \end{aligned}$$

Порогова логічна схема представлена на рисунку 3.21.

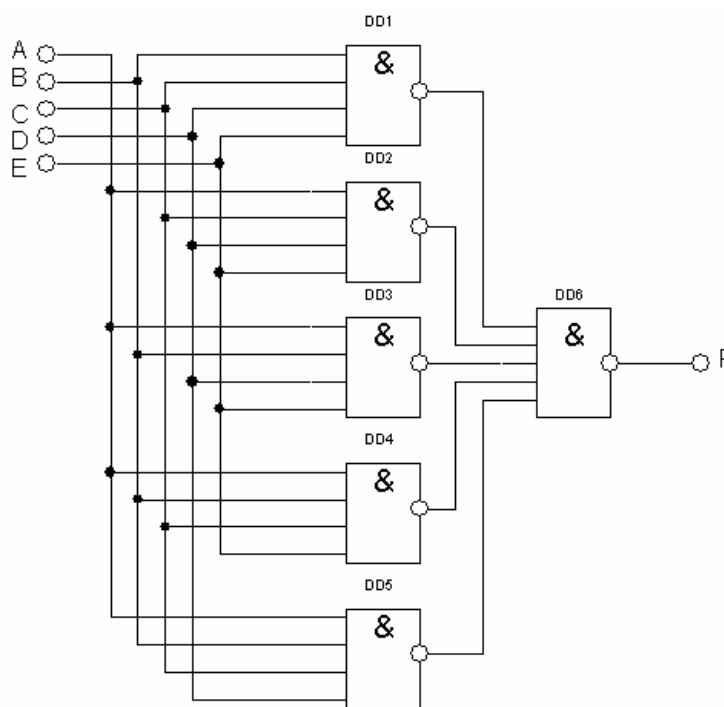


Рисунок 3.21 - Порогова логічна схема

3.7 Принципи побудови інтегрального дешифратора

Дешифратор (ДШ) – пристрій комбінаційного типу, який дозволяє визначити код вхідного числа по одиночному сигналу на одному з виходів.

Дешифратори широко застосовуються в пристроях управління, в системах цифрової індикації, для побудови розподільників імпульсів по різних колах і т. д. Дешифратор, в якого число виходів 2^n , де n - число входів, називається повним.

Лінійна структура дешифратора

З таблиці станів (табл. 3.3) випливає, що на кожному вході дешифратора повинен стояти інвертор. Рівняння функціонування дешифратора по кожному виході визначається певною комбінацією вхідних сигналів. Наприклад: на F3 логічна 1, якщо на вході комбінація $\overline{A}BC$.

Відповідно до рівнянь функціонування будується схема лінійного дешифратора (рис. 3.22).

Таблиця 3.8 – Таблиця станів лінійного дешифратора

Входи			Виходи							
A	B	C	F0	F1	F2	F3	F4	F5	F6	F7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

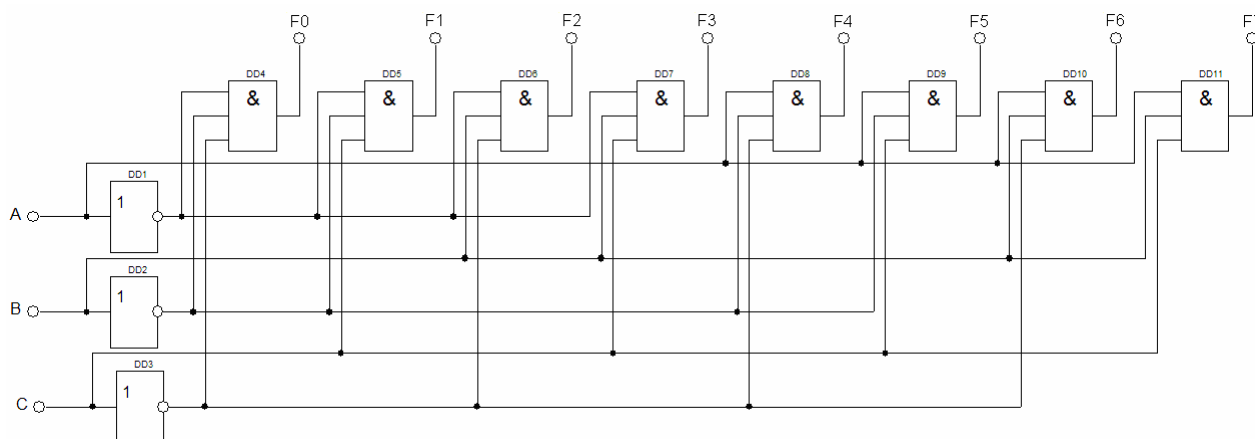


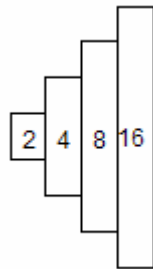
Рисунок 3.22 – Схема лінійного дешифратора

У ТТЛ дешифратори більш ніж на 4 ... 5 входів по здатності навантаження побудувати не можливо.

Кожному вхідному двійковому коду відповідає низький рівень лише на одному виході, а на всіх інших зберігається високий рівень.

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 (рис. 3.23) компактніше лінійної.

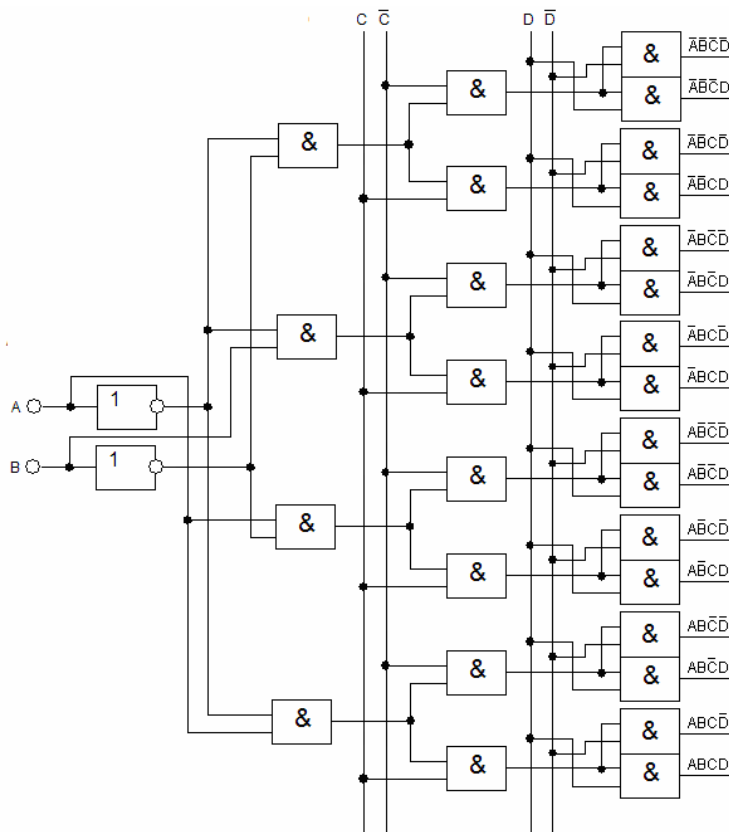


Рисунок 3.23 – Пірамідальна структура дешифратора

Матрична структура дешифратора.

На рисунку 3.24 показаний двохкасдний дешифратор «1 з 16», другий каскад якого зібраний за схемою координатного або матричного дешифратора. Розряди адреси розбиті на дві групи, кожна з яких незалежно від іншої розшифровується своїм дешифратором першого каскаду DD1 і DD2. При будь-якій комбінації значень вхідних змінних виявляються вибраними один рядок і один

стовпець сітки, у вузлах якої розташовані елементи 2І другого рівня. В результаті кожен адресний набір збуджує вихід єдиного відповідного йому елемента 2І. При використанні в другому рівні елементів 2І-НІ виходи дешифратора будуть інверсними. Їх можна зробити прямими, побудувавши координатну сітку на елементах 2АБО-НІ; тоді інверсними мають бути виходи дешифраторів першого каскаду. Розглянемо ДШ з $n = 2$ і $m = 4$, що називається також дешифратором "2 в 4" і доповнимо його входом дозволу виходів ОЕ (Output Enable).

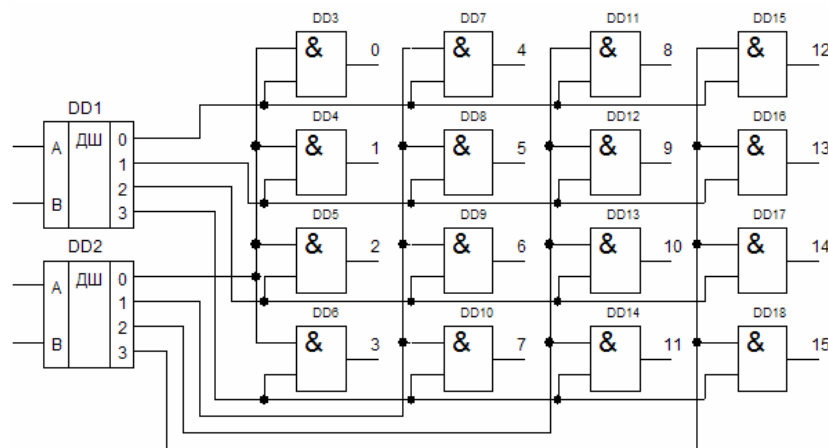


Рисунок 3.24 – Матрична структура дешифратора

Активним рівнем сигналу на прямих входах/виходах буде 1, а на інверсних - 0. За цим визначенням заповнимо таблицю істинності, де величина \times може набувати будь-яких значень (табл. 3.9).

Таблиця 3.9 – Таблиця істинності дешифратора "2 в 4"

DEC число	Входи				Виходи								Логічна функція	
	A	B	OE	\overline{OE}	F0	F1	F2	F3	$\overline{F0}$	$\overline{F1}$	$\overline{F2}$	$\overline{F3}$		
0	0	0			1	0	0	0	0	0	1	1	1	$F0 = OE \overline{A} \overline{B}$
1	0	1	1	0	0	1	0	0	1	0	1	1	1	$F1 = OE \overline{A} B$
2	1	0			0	0	1	0	1	1	0	1	1	$F2 = OE A \overline{B}$
3	1	1			0	0	0	1	1	1	1	0	1	$F3 = OE A B$
\times	\times	\times	0	1	0	0	0	0	0	1	1	1	1	$F_i = 0 \quad \overline{F_i} = 1$

Карта Карно для прямого виходу F0 і 3-х вхідних змінних матиме вигляд:

		A			
	AB	00	01	11	10
OE	0	0	0	0	0
		B			
OE	1	1	0	0	0

Логічна функція виходу F_0 матиме вигляд: $F_0 = OE \cdot \overline{A} \cdot \overline{B}$. Аналогічно отримано останні три рівняння. Перетворимо отримані для F_i рівняння за допомогою аксіоми подвійного заперечення до базису І-НІ:

$$F_0 = \overline{\overline{OE \cdot \overline{A} \cdot \overline{B}}}, F_1 = \overline{\overline{OE \cdot \overline{A} \cdot B}}, F_2 = \overline{\overline{OE \cdot A \cdot \overline{B}}}, F_3 = \overline{\overline{OE \cdot A \cdot B}}$$

Рішенню відповідає схема, представлена на рисунку 3.25.

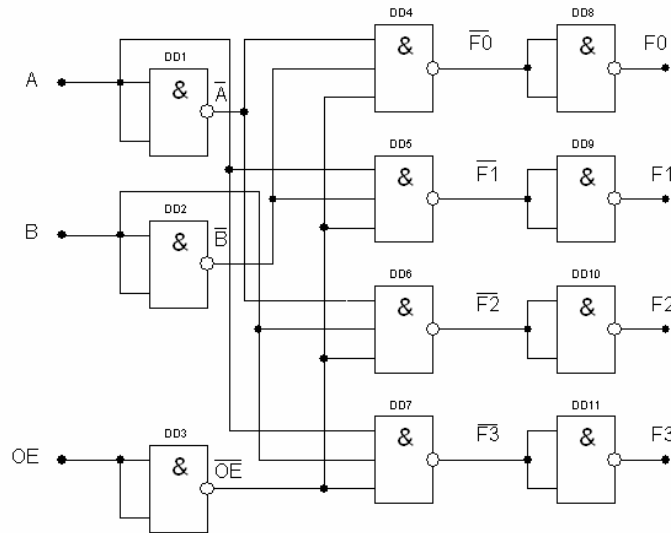


Рисунок 3.25 – Розрахована схема дешифратора

Можливі комбінації прямих і інверсних входів і виходів дешифраторів рисунку 3.26.

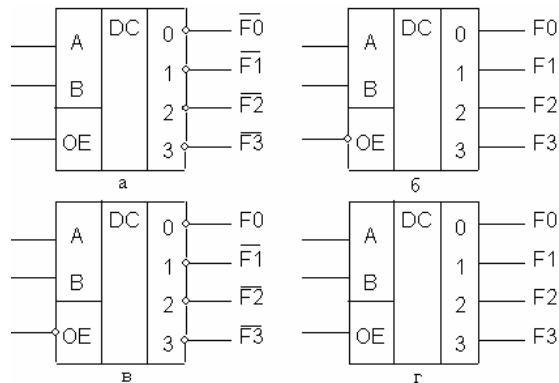


Рисунок 3.26 - Комбінації прямих і інверсних входів і виходів дешифраторів

Пояснити роботу ДШ можна за допомогою часових діаграм, наприклад для схеми (г) (рис. 3.27). Під час дії сигналу $\overline{OE} = 1$ (рис. 3.25) на нижніх входах елементів І-НІ (DD4...DD7) присутній $OE = 0$, і незалежно від значень А, В, вихідні значення $\overline{F_i} = 1$, а $F_i = 0$.

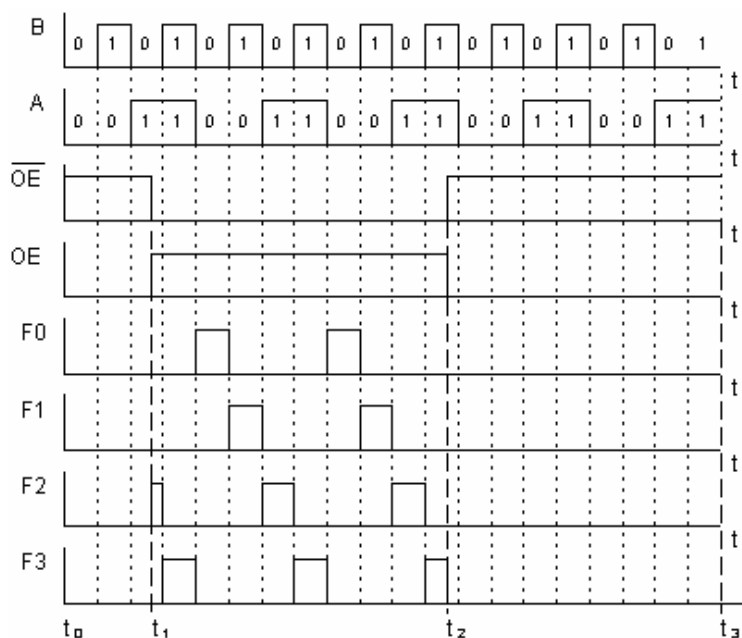


Рисунок 3.27 – Діаграма функціонування дешифратора з прямими входами, виходами і входом дозволу для виходу OE

У ці відрізки часу $t_0 \dots t_1$ і $t_2 \dots t_3$ виходи "заборонені", тобто на прямих виходах F_i встановлюється пасивний рівень "0", а на інверсних виходах пасивна "1". У інтервалі $t_1 \dots t_2$ сигнал $\overline{OE} = 0$ ($OE = 1$) і значення F_i залежать лише від змінних А, В. Якщо код на входах АВ = 10, що відповідає десятковій двійці, на входах другого елементу І-НІ зберуться три логічних "1". Сигнал $\overline{F_2} = 0$, а $F_2 = 1$, як це показано на діаграмі F2. Замість інвертора OE, може застосовуватися складніша схема, показана на рисунку 3.28. Тут $OE = 1$ у разі, коли $\overline{OE1} = \overline{OE2} = 0$ і $OE3 = 1$.

Дешифратори широко застосовуються в обчислювальній техніці у якості частини великих інтегральних схем, призначених для вибору одного з декількох зовнішніх пристроїв (ЗП) при обміні даними між ним і мікропроцесором. В

цьому випадку на входи a_i подаються сигнали, що називаються адресою ЗП, а входи називаються адресними.

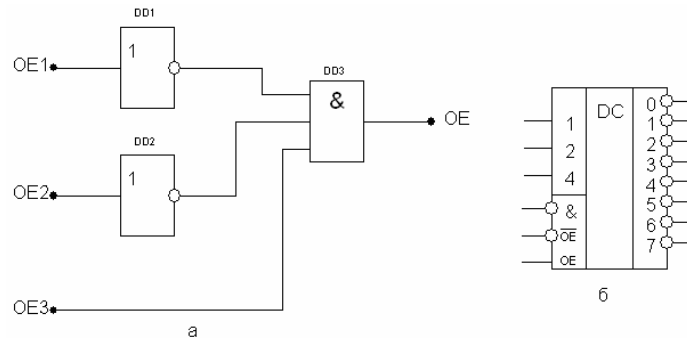


Рисунок 3.28 – Варіант схеми входу OE (а) та умовне графічне позначення дешифратора К555ИД7

3.8 Принцип побудови інтегрального демультиплексора

Демультиплексор – пристрій комбінаційного типу, що здійснює керовану комутацію інформації, який поступає по одному входу, на N виходів.

Узагальнена схема демультиплексора приведена на рисунку 3.29. У загальному випадку число вихідних ліній N визначається кількістю адресних входів n і дорівнює $N = 2^n$.

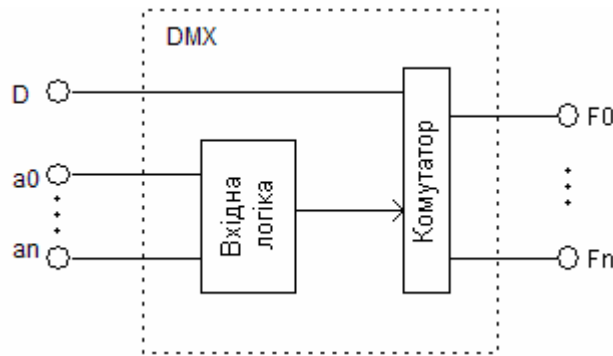


Рисунок 3.29 – Узагальнена схема демультиплексора

Для випадку $n = 2$ функціонування демультиплексора здійснюється відповідно до таблиці істинності (табл. 3.10).

Із таблиці істинності записуємо характеристичні рівняння демультиплексора:

$$F_0 = \overline{D}\overline{A}\overline{B}; \quad F_1 = \overline{D}A\overline{B}; \quad F_2 = D\overline{A}\overline{B}; \quad F_3 = DAB.$$

Таблиця 3.10 – Таблиця істинності демультимплектора “1 в 4”

Адреса		Виходи			
A	B	F0	F1	F2	F3
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

Функціональна схема демультимплектора, яка відповідає цим рівнянням, приведена на рисунку 3.30. Вона має в своєму складі два інвертори і чотири елементи «І».

Порівнюючи таблиці істинності і функціональні схеми демультимплектора і дешифратора, легко побачити схожість їх функцій.

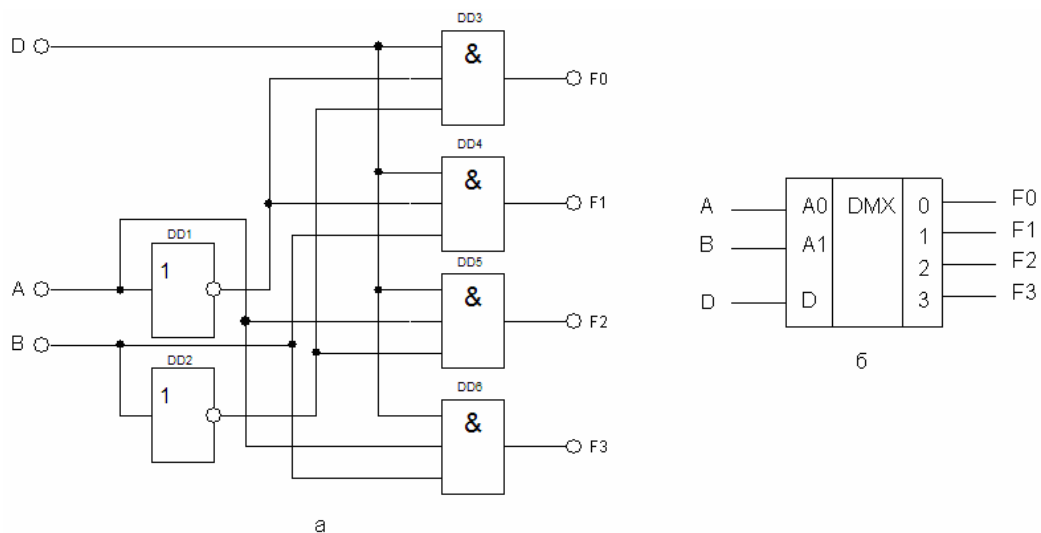


Рисунок 3.30 – Функціональна схема демультимплектора “1 в 4” (а) та його схемний аналог (б)

Якщо функція $D = 1$ постійно, то демультимплексор виконує функції дешифратора. З іншого боку, якщо в дешифраторі замість сигналу OE подається інформаційний сигнал D, то він виконує функції демультимплектора.

На рисунку 3.31 показано з'єднання двох дешифраторів "3 в 8" для отримання одного дешифратора "4 в 16", або двох демультимплекторів "1 в 8" для отримання одного "1 в 16".

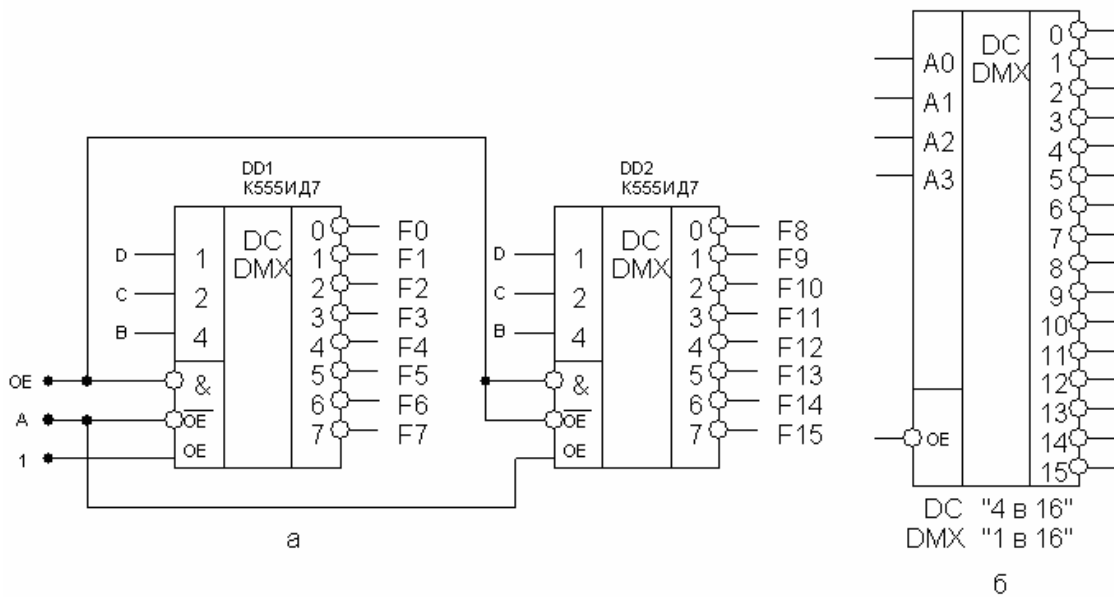


Рисунок 3.31 - Збільшення розрядності дешифратора до "4 в 16" та де-мультиплексора до "1 в 16", виконане на двох мікросхемах К555ИД7 (а), та умовне графічне позначення отриманої схеми

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 дешифра-торів працює, залежить лише від сигналу А (табл. 3.11).

Таблиця 3.11 – Таблиця функціонування схеми збільшення розрядів

DEC № виходу	Входи				Виходи																
	A	B	C	D	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	
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	1
2		0	1	0	1	1	0	1	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	1
4		1	0	0	1	1	1	1	0	1	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	1
6		1	1	0	1	1	1	1	1	1	0	1	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	1
8	1	0	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	
9		0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	
10		0	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	
11		0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	
12		1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	
13		1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	
14		1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	
15		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1

Якщо $A = 0$, то на всіх вирішуючих входах DD1 будуть активні рівні, а на прямому дозволяючому вході OE дешифратора DD2 сигнал A, рівний нулю, переведе всі виходи в стан 1, тобто працездатним буде 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 DD2. Тому нумерація виходів F_i дешифратора "4 в 16" є наскрізною від 0 до 15.

Враховуючи схожість функцій, що виконують мікросхеми дешифраторів і демультимплексорів, вони мають однакове умовне позначення – ИД і називаються «Дешифратор – демультимплексор».

3.9 Принцип побудови інтегрального мультиплексора

Мультиплексор – пристрій комбінаційного типу, який має n - адресних входів, $N = 2^n$ інформаційних входів, один вихід і здійснює керовану комутацію інформації, яка поступає по N вхідним лініям, на одну вихідну лінію. Комутація певної вхідної лінії відбувається відповідно до двійкового адресного коду. Якщо адресний код має n – розрядів, то можна здійснити $N = 2^n$ комбінацій адресних сигналів, кожна з яких забезпечить підключення однієї з N вхідних ліній до вихідної лінії. Такий мультиплексор називають « N в один». За наявності надлишкових комбінацій адресних сигналів можна спроектувати мультиплексор з будь-яким числом вхідних ліній $N \leq 2^n$

Розглянемо мультиплексор "4 в 1", що має 4 інформаційних входа і $\log_2 4 = 2$ адресних входа. Якщо є вхід дозволу виходу OE, то "0" на цьому вході повинен перевести вихід в пасивний стан. Алгоритм функціонування такого мультиплексора приведений в таблиці 3.12.

Величина \times може набувати будь-яких значень. Кількість вхідних змінних дорівнює 7, і таблиця істинності повинна мати 128 рядків.

Таблиця 3.12 – Таблиця істинності мультиплексора “4 в 1”

DEC число	Входи								Вихід Q	Логічна функція
	A	B	OE	\overline{OE}	D0	D1	D2	D3		
0	0	0			D0	X	X	X	D0	$Q = D0 \overline{A} \overline{B}$
1	0	1	1	0	X	D1	X	X	D1	$Q = D1 \overline{A} B$
2	1	0			X	X	D2	X	D2	$Q = D2 A \overline{B}$
3	1	1			X	X	X	D3	D3	$Q = D3 A B$
X	X	X			X	X	X	X	0	$Q = 0$

У таблиці в 4-х основних рядках упаковано 64 початкових рядки (з врахуванням значень D0 ... D3), і в останньому рядку - останні 64 рядки. Аналіз 0 рядка призводить до висновку, що $Q = D0$, якщо $A = 0$, $B = 0$ і $OE = 1$, незалежно від змінних D1 ... D3. Тому, для цього вхідного набору можна записати: $Q = D0 \cdot OE \cdot \overline{A} \cdot \overline{B}$. Аналогічно записуються рівняння для останніх трьох наборів змінних. Загальне рішення тоді матиме вигляд:

$$Q = OE \cdot (D0 \cdot \overline{A} \cdot \overline{B} + D1 \cdot \overline{A} \cdot B + D2 \cdot A \cdot \overline{B} + D3 \cdot A \cdot B)$$

Застосовуючи аксіоми подвійного заперечення і подвійності до правої частини рівняння, отримаємо:

$$Q = \overline{\overline{OE \cdot D0 \cdot \overline{A} \cdot \overline{B}} \cdot \overline{OE \cdot D1 \cdot \overline{A} \cdot B} \cdot \overline{OE \cdot D2 \cdot A \cdot \overline{B}} \cdot \overline{OE \cdot D3 \cdot A \cdot B}}$$

Рівнянню відповідає схема, приведена на рисунку 3.32.

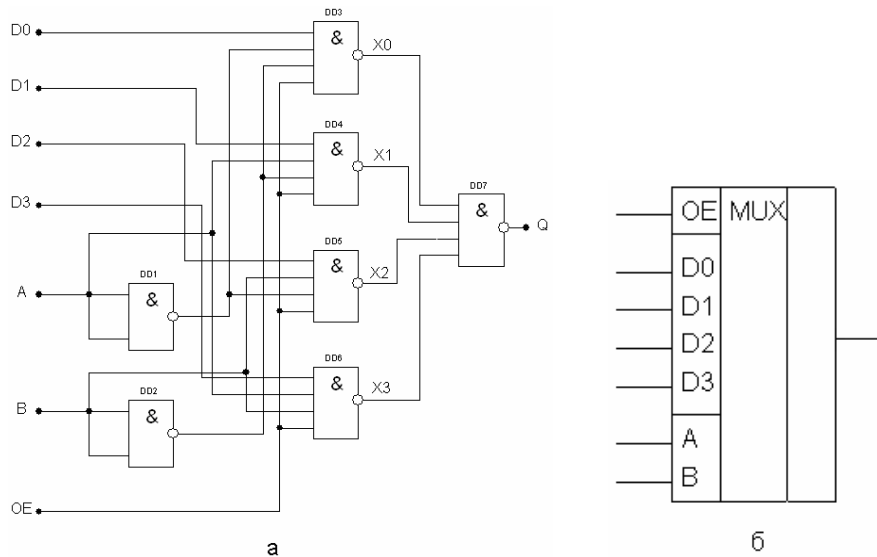


Рисунок 3.32 – Функціональна схема мультиплексора “4 в 1” (а) та його схемний аналог (б)

Діаграма функціонування мультиплексора представлена на рисунку 3.33.

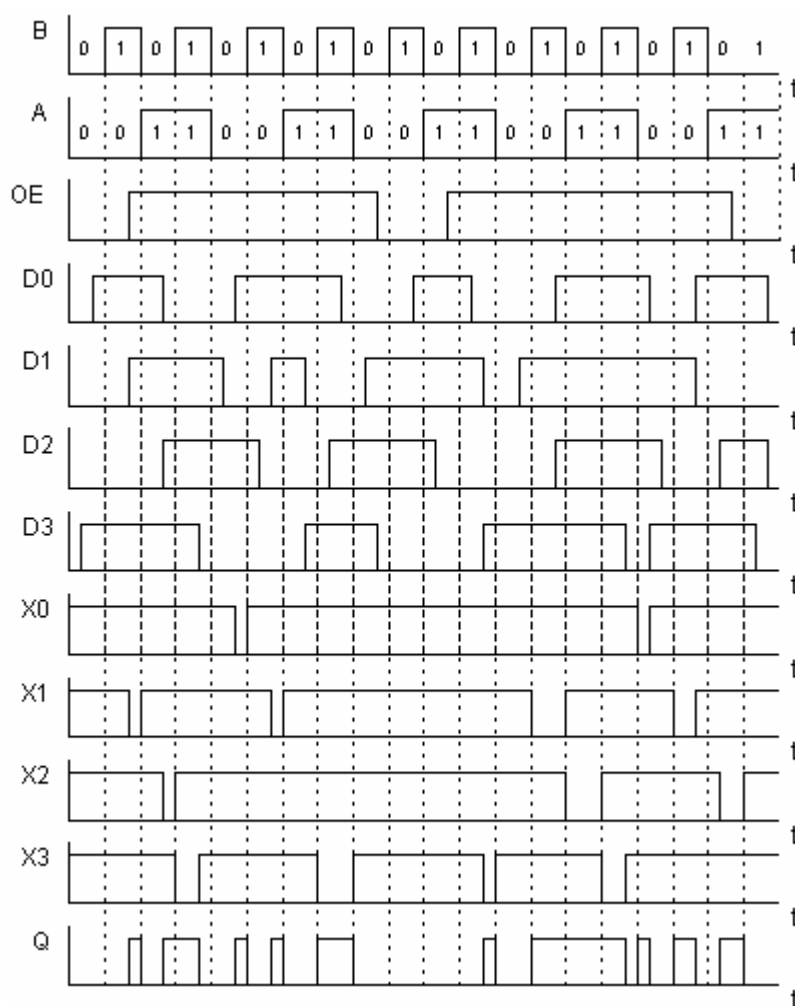


Рисунок 3.33 - Діаграма функціонування мультиплексора "4 в 1"

Мультиплексор може мати інверсний вихід, а також третій стан цього виходу, який позначається на схемі ромбом з поперечною рисою.

Мультиплексори знаходять широке використання в обчислювальній техніці, наприклад, частка виводів у мікропроцесорів "мультиплексована", тобто, до одного виходу підключається декілька внутрішніх джерел різних сигналів. Це можуть бути сигнали ліній шини даних і шини адреси, що передаються послідовно в часі, що дозволяє скоротити загальне число виводів мікропроцесора. За допомогою мультиплексора з "n" адресними входами можна реалізувати будь-яку логічну функцію з "n" змінними, подаючи на інформаційні входи мультиплексора значення D_i .

3.10 Принцип побудови інтегрального шифратора

Шифратор (кодер) – пристрій комбінаційного типу, 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. Якщо кількість входів настільки велика, що в шифраторі використовуються усі можливі комбінації сигналів на виході, то такий шифратор називається повним, якщо не всі, то неповним. Число входів і виходів в повному шифраторі зв'язане співвідношенням $n = 2^m$, де n – число входів, m – число виходів. Для перетворення коду кнопочового пульта в чотирьохрозрядне двійкове число досить використовувати лише 10 входів, тоді як повне число можливих входів дорівнює 16 ($n = 2^4 = 16$), тому шифратор "10 в 4" буде неповним.

Шифратор може бути неперіоритетним, якщо допускається подача лише одного активного сигналу і може бути періоритетним, якщо допускається подача одночасно декількох активних сигналів на входи. Неперіоритетний шифратор здійснює перетворення десяткового номера активного входу в двійковий еквівалент цього номера. Для неперіоритетного шифратора "10 в 4" алгоритм функціонування представлений в таблиці 3.13.

Таблиця 3.13 – Таблиця істинності чотирьохрозрядного шифратора

Входи	Виходи			
	F4	F3	F2	F1
X0	0	0	0	0
X1	0	0	0	1
X2	0	0	1	0
X3	0	0	1	1
X4	0	1	0	0
X5	0	1	0	1
X6	0	1	1	0
X7	0	1	1	1
X8	1	0	0	0
X9	1	0	0	1

Рівняння функціонування шифратора:

$$F1 = X1 + X3 + X5 + X7 + X9;$$

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

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

$$F4 = X8 + X9.$$

У загальному випадку, при використанні двійкового коду, можна закодувати 2^n вхідних сигналів. У розглянутій вище схемі вихідний код «0000» присутній на виході при подачі сигналу на вхід X_0 , а також в разі, якщо вхідний сигнал взагалі не подається ні на один з входів. Для однозначної ідентифікації сигналу X_0 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 3.34.

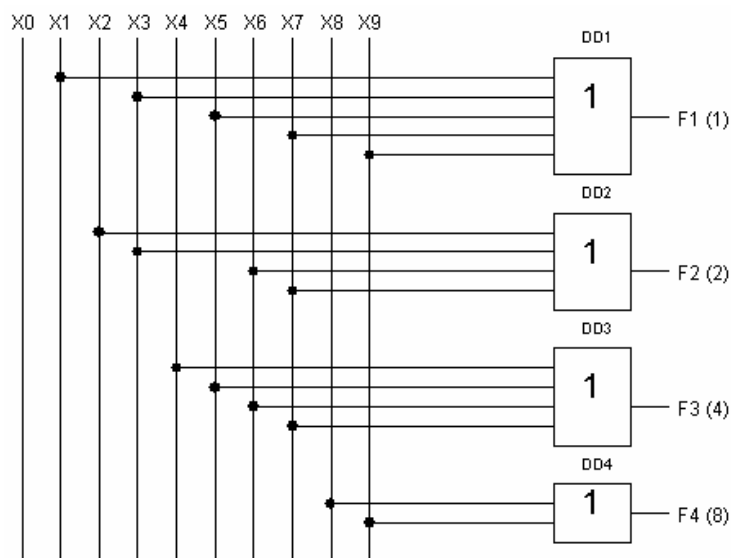


Рисунок 3.34 – Функціональна схема чотирьохрозрядного шифратора

У шифраторі з пріоритетом (пріоритетному шифраторі) здійснюється перетворення максимального десяткового номера активного входу в двійковий еквівалент цього номера. У таких шифраторах код двійкового числа відповідає найвищому номеру входу, на який поданий активний сигнал. Тобто, на пріоритетний шифратор допускається подавати сигнали на декілька входів, а він формує на виході код числа, відповідного старшому входу. Для такого шифратора

вхідні сигнали, які знаходяться знизу від одиничної діагоналі, за визначенням, не відомі ("×" може бути 0 або 1). Комбінація – усі нулі на входах не визначена.

Розглянемо як приклад шифратор з пріоритетом К555ІВ3 (рис. 3.35).

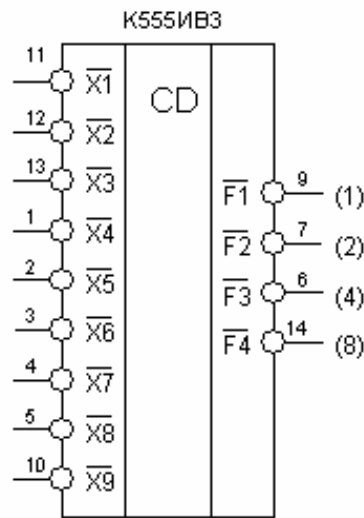


Рисунок 3.35 – Умовне графічне позначення мікросхеми К555ІВ3

Шифратор має 9 інверсних входів, які позначені через $\overline{X1}.. \overline{X9}$, та чотири інверсних виходи $\overline{F1}.. \overline{F4}$. Цифри біля виходів визначають значення активного рівня (нуля) у відповідному розряді двійкового числа. Вочевидь, що це неповний шифратор.

Якщо на всіх входах – логічна одиниця, то на всіх виходах також логічна одиниця, що відповідає числу 0_{10} в інверсному коді $(1111)_2$. Якщо хоч би на одному вході є логічний нуль, то стан вихідних сигналів визначається найбільшим номером входу, на якому є логічний нуль, і не залежить від сигналів на входах, що мають менший номер.

Наприклад (табл. 3.14), якщо на вході $\overline{X1}$ – логічний нуль, а на всіх останніх входах – логічна одиниця, то на виходах присутні наступні сигнали: $\overline{F1} = 0$, $\overline{F2} = 1$, $\overline{F3} = 1$, $\overline{F4} = 1$, що відповідає числу 1_{10} в інверсному коді $(1110)_2$. Якщо на вході $\overline{X9}$ логічний нуль, то незалежно від інших вхідних сигналів на виходах присутні наступні сигнали: $\overline{F1} = 0$, $\overline{F2} = 1$, $\overline{F3} = 1$, $\overline{F4} = 0$, що відповідає числу 9_{10} в інверсному коді $(0110)_2$.

Згідно з таблицею функціонування пріоритетного шифратора К555ІВ3 для виходів $\overline{F1}$, $\overline{F2}$, $\overline{F3}$, $\overline{F4}$:

$$\overline{F1} = \overline{X9} + \overline{X7X8X9} + \overline{X5X6X7X8X9} + \overline{X3X4X5X6X7X8X9} + \overline{X1X2X3X4X5X6X7X8X9};$$

Таблиця 3.14 - Таблиця істинності пріоритетного шифратора К555ІВ3

Входи									Виходи			
$\overline{X1}$	$\overline{X2}$	$\overline{X3}$	$\overline{X4}$	$\overline{X5}$	$\overline{X6}$	$\overline{X7}$	$\overline{X8}$	$\overline{X9}$	$\overline{F4}$	$\overline{F3}$	$\overline{F2}$	$\overline{F1}$
1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0
X	0	1	1	1	1	1	1	1	1	1	0	1
X	X	0	1	1	1	1	1	1	1	1	0	0
X	X	X	0	1	1	1	1	1	1	0	1	1
X	X	X	X	0	1	1	1	1	1	0	1	0
X	X	X	X	X	0	1	1	1	1	0	0	1
X	X	X	X	X	X	0	1	1	1	0	0	0
X	X	X	X	X	X	X	0	1	0	1	1	1
X	X	X	X	X	X	X	X	0	0	1	1	0

$$\overline{F2} = \overline{X7X8X9} + \overline{X6X7X8X9} + \overline{X3X4X5X6X7X8X9} + \overline{X2X3X4X5X6X7X8X9};$$

$$\overline{F3} = \overline{X7X8X9} + \overline{X6X7X8X9} + \overline{X5X6X7X8X9} + \overline{X4X5X6X7X8X9};$$

$$\overline{F4} = \overline{X9} + \overline{X8X9}.$$

Повторним застосуванням до кожної функції F_i співвідношення алгебри логіки:

$$A + F\overline{A} = (A + F)(A + \overline{A}) = (A + F) \cdot 1 = A + F,$$

можна спростити їх і отримати вирази, які визначають внутрішню структуру пріоритетного шифратора в його основній частині:

$$\overline{F1} = \overline{X9} + \overline{X7X8} + \overline{X5X6X8} + \overline{X3X4X6X8} + \overline{X1X2X4X6X8};$$

$$\overline{F2} = \overline{X7X8X9} + \overline{X6X8X9} + \overline{X3X4X5X8X9} + \overline{X2X4X5X8X9};$$

$$\overline{F3} = \overline{X7X8X9} + \overline{X6X8X9} + \overline{X5X8X9} + \overline{X4X8X9};$$

$$\overline{F4} = \overline{X9} + \overline{X8}.$$

Основна частина пріоритетного шифратора К555ІВ3 представлена на рисунку 3.36.

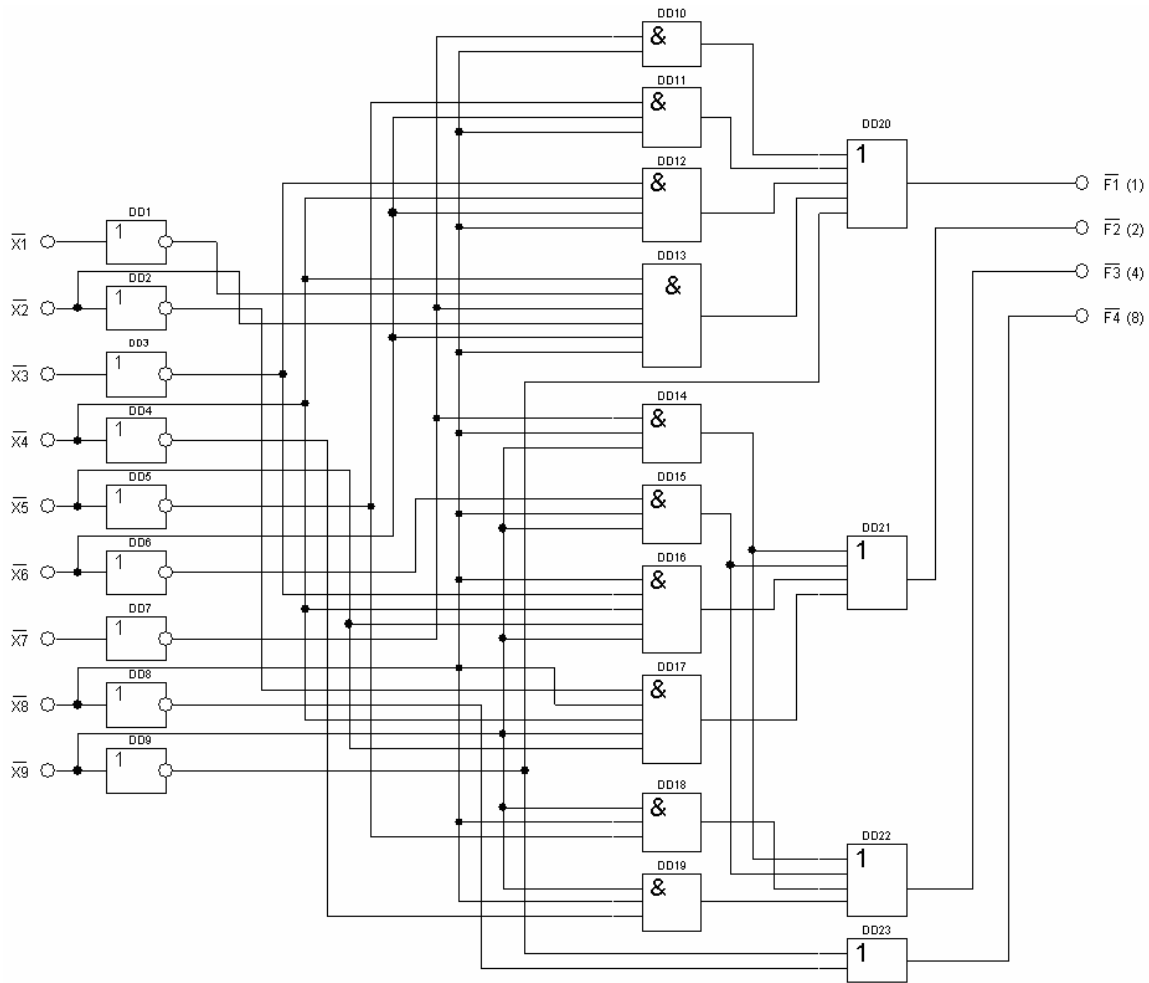


Рисунок 3.36 – Схема пріоритетного шифратора К555ІВ3

Для конструювання шифраторів з більшим числом входів, тобто нарощування розрядності шифратора, об'єднують мікросхеми шифраторів з додатковими виводами. Розглянемо варіант шифратора з додатковими виводами, де активним рівнем на входах і виходах, є логічна "1" (рис. 3.37).

Схема представляє собою пріоритетний шифратор "8 в 3", тобто має 8 прямих входів і 3 прямих виходи. Вона має вхід дозволу E1, вихід перенесення E0 і вихід CS, який визначає ознаку вхідного інформаційного сигналу. Якщо на всіх інформаційних входах логічний "0", то при подачі на вхід E1 логічної "1", на виходах F0 ... F3 і G будуть логічні "0", а на виході перенесення E0 – логічна "1".

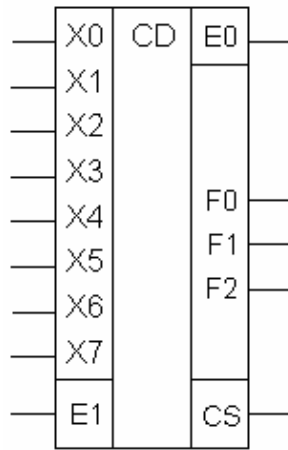


Рисунок 3.36 – Експериментальна схема пріоритетного шифратора "8 в 3" з додатковими виводами

Якщо активізувати один з інформаційних входів (подати на нього логічну "1"), то на виходах з'явиться код, який відповідає номеру активізованого входу, на вході CS – логічна "1", яка є ознакою подачі вхідного сигналу, а на виході E0 – логічний "0". Якщо ж схема не активізована, тобто на вхід дозволу E1 поданий логічний "0", то на всіх виходах схеми також буде логічний "0" незалежно від того, що буде подане на інформаційні входи (табл. 3.15).

Таблиця 3.15 – Таблиця функціонування пріоритетного шифратора "8 в 3" з додатковими виводами

Входи									Виходи				
E1	X0	X1	X2	X3	X4	X5	X6	X7	F2	F1	F0	CS	E0
1	1	0	0	0	0	0	0	0	0	0	0	1	0
1	X	1	0	0	0	0	0	0	0	0	1	1	0
1	X	X	1	0	0	0	0	0	0	1	0	1	0
1	X	X	X	1	0	0	0	0	0	1	1	1	0
1	X	X	X	X	1	0	0	0	1	0	0	1	0
1	X	X	X	X	X	1	0	0	1	0	1	1	0
1	X	X	X	X	X	X	1	0	1	1	1	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1
0	X	X	X	X	X	X	X	X	0	0	0	0	0

Згідно таблиці функціонування пріоритетного шифратора для виходів F2, F1, F0, CS, E0:

$$F2 = (X7 + X6\overline{X7} + X5\overline{X6X7} + X4\overline{X5X6X7})E1;$$

$$F1 = (X7 + X6\overline{X7} + X3\overline{X4X5X6X7} + X2\overline{X3X4X5X6X7})E1;$$

$$F0 = (X7 + X5\overline{X6X7} + X3\overline{X4X5X6X7} + X1\overline{X2X3X4X5X6X7})E1;$$

$$CS = (X7 + X6 + X5 + X4 + X3 + X2 + X1 + X0)E1;$$

$$E0 = \overline{X7 \cdot X6 \cdot X5 \cdot X4 \cdot X3 \cdot X2 \cdot X1 \cdot X0} \cdot E1.$$

Спростимо функції повторним застосуванням співвідношень алгебри логіки і отримаємо вирази, які визначають внутрішню структуру пріоритетного шифратора "8 в 3" з додатковими виводами в його основній частині:

$$F2 = (X7 + X6 + X5 + X4)E1;$$

$$F1 = (X7 + X6 + X3\overline{X4X5} + X2\overline{X4X5})E1;$$

$$F0 = (X7 + X5\overline{X6} + X3\overline{X4X6} + X1\overline{X2X4X6})E1.$$

Схема пріоритетного шифратора "8 в 3" з додатковими виводами згідно рівнянням функціонування представлена на рисунку 3.37.

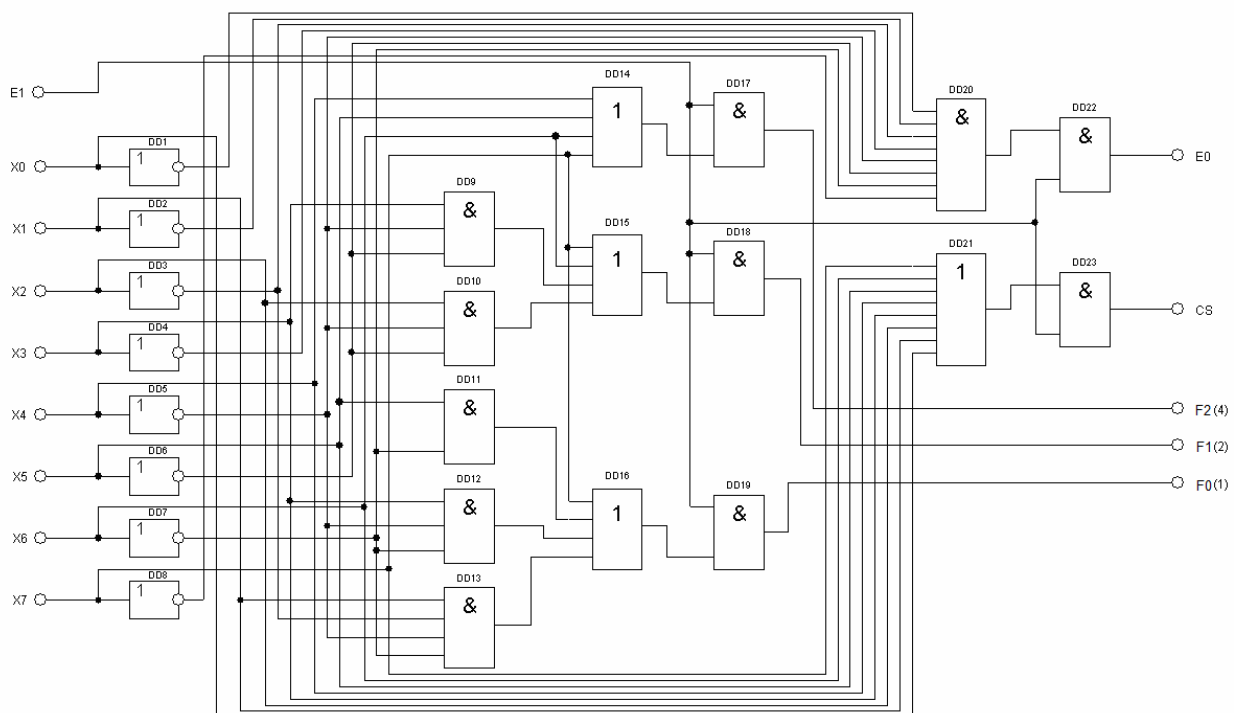


Рисунок 3.37 - Схема пріоритетного шифратора "8 в 3" з додатковими виводами

Мікросхема К555ИВ1 (74148) (рис. 3.38) представляє собою пріоритетний шифратор "8 в 3", тобто має 8 інверсних входів і 3 інверсних виходів. Окрім цього, вона має вхід дозволу Е1, вихід перенесення Е0 і вихід CS, що визначає ознаку вхідного інформаційного сигналу.

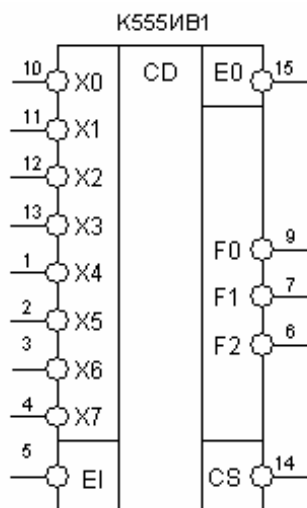


Рисунок 3.38 – Умовне графічне позначення мікросхеми К555ИВ1

Якщо на всіх інформаційних входах присутня логічна "1", то при подачі на вхід Е1 логічного "0", на виходах F0, F1, F2 і CS будуть також логічні "1", а на виході перенесення Е0 – логічний "0".

Якщо активізувати один з інформаційних входів (подати на нього логічний 0), то на виходах F0, F1, F2 з'явиться інверсний код, що відповідає номеру активізованого входу, на виході CS – логічний "0", що являється ознакою подачі вхідного сигналу, а на виході Е0 – логічна "1".

Якщо мікросхема не активізована, тобто на вхід дозволу Е1 подана логічна "1", то на всіх виходах мікросхеми також буде логічна "1" незалежно від того, що буде подане на інформаційні входи.

Дві мікросхеми К555ИВ1 можна з'єднати за схемою представленою на рисунку 3.39 для здобуття пріоритетного шифратора на 16 входів.

Якщо логічний "0" поданий на один з входів X0 ... X7, на виходах DD3 з'являться молодші розряди прямого вихідного коду, на виході CS DD1 – логічний "0", що визначає розряд F3 вихідного коду з ваговим коефіцієнтом (8), на

виході E0 – логічна “1”, що є ознакою подачі вхідного сигналу. Якщо логічний “0” подати на один з входів X8 ... X15, логічна “1” з виходу E0 DD2 заборонить роботу DD1, молодші розряди на виходах DD3 визначаються мікросхемою DD2, на виході F3 вихідного коду буде логічна “1”.

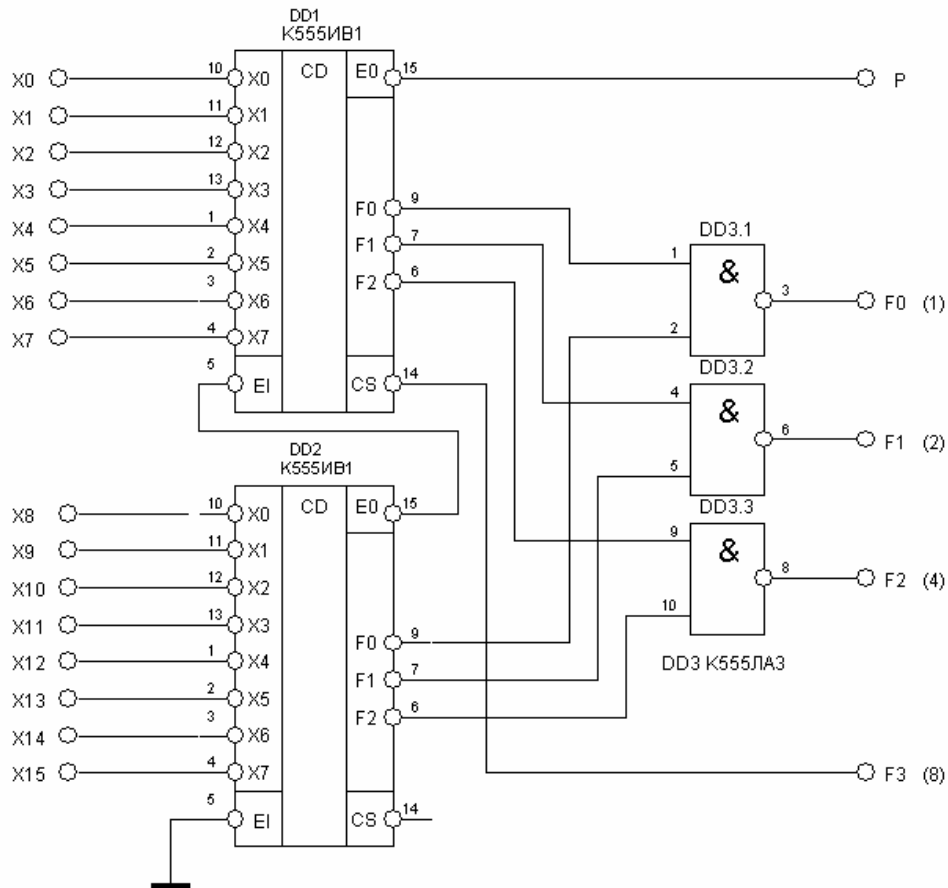


Рисунок 3.39 – Пріоритетний шифратор "16 в 4"

Таким чином, з виходів F0 ... F3 пристрою можна зняти прямий код, який відповідає номеру входу, на який поданий логічний “0”. У схемі (рис. 3.39) зберігається властивість пріоритетності шифраторів – при одночасній подачі логічного “0” на декілька входів, вихідний код завжди відповідає входу з найбільшим номером.

Шифратори застосовуються в контролерах переривань роботи мікропроцесора зовнішніми пристроями, в паралельному перетворювачі напруги в код і для кодування номера клавіші.

3.11 Перетворювачі кодів

Операція зміни коду числа називається його перекодуванням. Інтегральні мікросхеми, що виконують ці операції, називаються перетворювачами коду (ПК). Перетворювачі коду бувають прості і складні. До простих відносяться перетворювачі, які виконують стандартні операції зміни коду чисел, наприклад, перетворень двійкового коду в десятковий або зворотну операцію. Складні перетворювачі коду виконують нестандартні перетворення коду і їх схеми доводиться розробляти кожен раз за допомогою алгебри логіки.

По своїй структурі перетворювачі коду являються дешифраторами, тільки вони перетворюють двійковий код в сигнали не лише на одному, але і на декількох виходах.

Вважатимемо, що перетворювачі коду мають n входів і m виходів. Співвідношення між n і m можуть бути будь-якими: $n = m$, $n < m$ і $n > m$. При перетворенні кодів чисел з ними можуть виконуватися різні додаткові операції, наприклад, множення на вагові коефіцієнти.

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

Інтегральні мікросхеми перетворювачів коду випускаються лише для найбільш поширених операцій:

- перетворювачі двійково-десятькового коду в двійковий код;
- перетворювачі двійкового коду в двійково-десятьковий код;
- перетворювачі двійкового коду в код Грея;
- перетворювачі двійкового коду в код управління сегментними індикаторами;
- перетворювачі двійкового або двійково-десятькового коду в код управління шкальними або матричними індикаторами.

Як приклад, розглянемо перетворювач двійкового коду в код управління семисегментними світлодіодними індикаторами (рис. 3.40). На рисунку також

приведені зображення цифр. Такий перетворювач повинен мати чотири входи, оскільки для кодування десяткових цифр від 0 до 9 вистачає чотири двійкових розряди, і сім виходів, поодиноці на кожен сегмент.

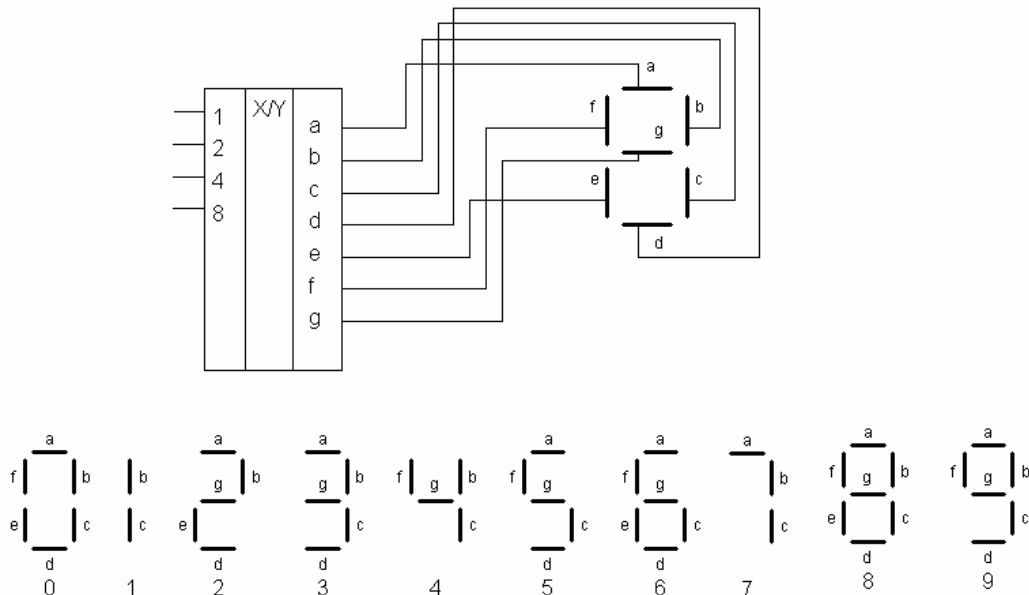
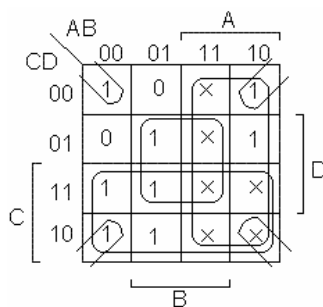


Рисунок 3.40 - Перетворювач двійкового коду в код управління семисегментними світлодіодними індикаторами

З таблиці істинності перетворювача коду (табл. 3.16) видно, що, наприклад, в цифрі 0 повинні світитися всі сегменти за винятком сегменту g, в цифрі 1 світяться лише два сегменти b і c і т. д. Вагові коефіцієнти двійкових розрядів дорівнюють $(8421)_2$. Нулі проставлені для тих цифр, в яких сегмент не світиться.

Вихідна функція для сегменту a:

$$F_a = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D}$$



Після спрощення отримаємо: $F_a = A + C + BD + \overline{B}\overline{D}$.

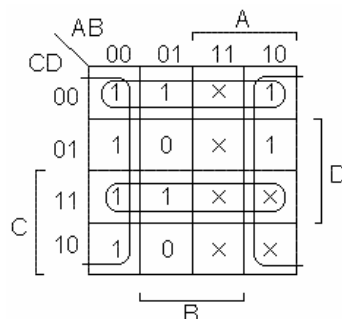
Таблиця 3.16 – Таблиця істинності перетворювача двійкового коду в код управління семисегментними світлодіодними індикаторами

DEC цифра	Входи				Виходи						
	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
10	×	×	×	×	×	×	×	×	×	×	×
11	×	×	×	×	×	×	×	×	×	×	×
12	×	×	×	×	×	×	×	×	×	×	×
13	×	×	×	×	×	×	×	×	×	×	×
14	×	×	×	×	×	×	×	×	×	×	×
15	×	×	×	×	×	×	×	×	×	×	×

У зв'язку з тим що індикатор може відобразити лише цифри від 0 до 9, в логічному наборі для чотирьох змінних - для цифр від 10 до 15 - стани вихідних сигналів є неактивованими. Тому, при мінімізації функцій виходів, їх можна довізначити (вважати що позначка × – логічна “1”), що спрощує схему перетворювача коду.

Вихідна функція для сегменту b:

$$F_b = \overline{A}BCD + \overline{A}BC\overline{D} + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D}$$



Після спрощення отримаємо: $F_b = \overline{B} + CD + \overline{C}\overline{D}$.

Аналогічно отримуємо функції для наступних виходів:

$$F_c = \overline{C} + B + D; F_d = A + C\overline{D} + \overline{B}C + \overline{B}\overline{D} + B\overline{C}\overline{D}; F_e = C\overline{D} + \overline{B}\overline{D};$$

$$Ff = A + B\bar{C} + B\bar{D} + \bar{C}\bar{D}; Fg = A + B\bar{C} + C\bar{D} + \bar{B}C.$$

Реалізація схеми на елементах елементарної логіки представлена на рисунку 3.41.

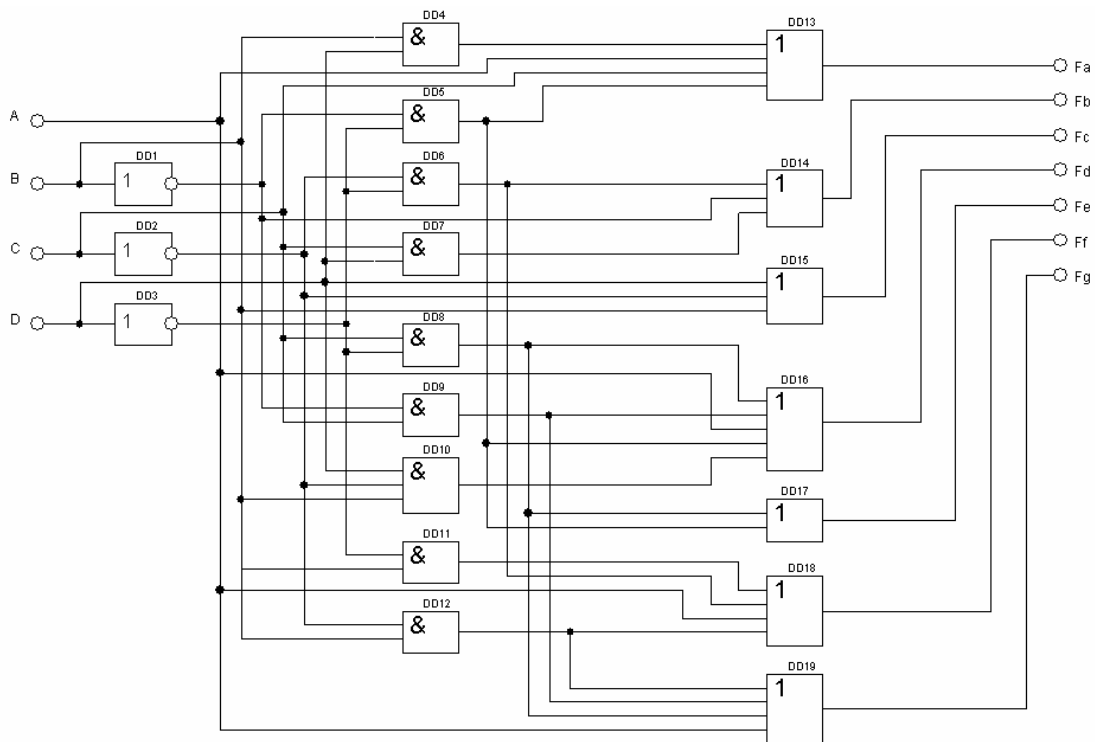


Рисунок 3.41 – Схема перетворювача двійкового коду в код управління семисегментними світлодіодними індикаторами

Контрольні запитання до третього розділу

1. Побудувати схему лінійного дешифратора.
2. Побудувати схему пірамідального дешифратора.
3. Побудувати схему розширення виходів дешифратора.
4. Побудувати схему шифратора з десяткового коду чисел в двійковий код.
5. Побудувати схему заданого перетворювача кодів.
6. Побудувати мультиплексор з 8 в 1.
7. Побудувати демультимплексор з 1 в 8.
8. Побудувати схему напівсуматора на елементарній логіці.
9. Побудувати схему напівсуматора на елементах «Виключаюче АБО».
10. Побудувати схему чотирьохрозрядного суматора.

ЦИФРОВІ ПРИСТРОЇ ПОСЛІДОВНІСНОГО ТИПУ

Цифрові пристрої послідовнісного типу 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. Функції переходу і виходів можуть задаватися у формі таблиць або за допомогою графів. При завданні у вигляді графів стани автомата представляють вершинами, а переходи із стану в стан – дугами. На дугах вказуються значення вхідних сигналів, що викликають відповідні переходи. Вихідні сигнали не визначаються вхідними в один і той же момент часу: $y^{(t)} \neq f(x^{(t)})$.

4.1 Бістабільна комірка

Розглянемо послідовне включення транзисторних ключів VT1...VT4 (рис. 3.1). Стан на виході транзистора VT3 такий же, як і на вході транзистора VT2.

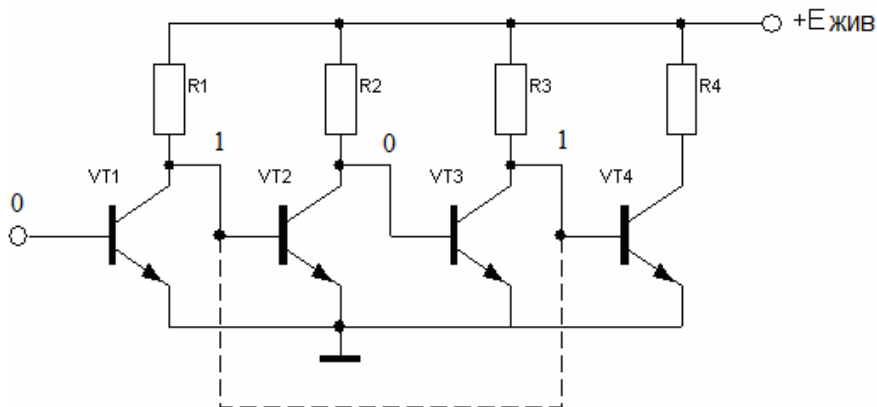


Рисунок 4.1 – Послідовне з'єднання транзисторів

Тому, ізолюючи дану пару, і з'єднавши вихід VT3 з входом VT2, ми не змінимо стан даної пари. Такі пристрої називають бістабільними комірками (БК) (рис. 4.2)

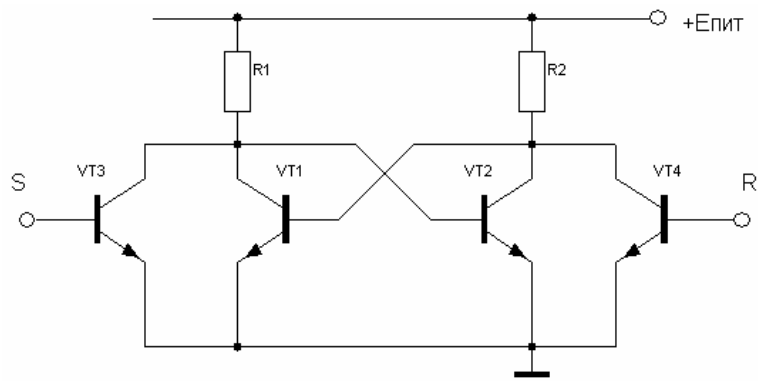


Рисунок 4.2 – Бістабільна коміррка

Стійкий стан БК характеризується тим, що один з ключів замкнутий, а другий відкритий і насичений. Інакше кажучи, бістабільній комірці властива електрична асиметрія.

Нехай транзистор VT1 замкнутий, а VT2 відкритий і насичений. Для того, щоб управляти коміркою необхідно підключити ще два ключі. Якщо на VT4 подати високий потенціал, стан на колекторі VT2 не зміниться ($U_{зал}$) – він відкритий. Якщо на VT3 подати високий потенціал – він відкриється, на колекторі VT1 з'явиться низький потенціал, а отже, і на базі VT2, який закриється. На базі VT1- високий потенціал і він відкриється. Отримаємо наступний стабільний стан.

4.2 Тригери

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 інший. При цьому напру- га на його виході стрибкоподібно змінюється.

За наявності тактового входу С тригер називають синхронним, а при його відсутності – асинхронним. Сигнали керування на асинхронний тригер впливають безпосередньо з початком своєї появи на входах, а в синхронних – лише з приходом сигналу на вхід С.

Тригери можуть мати статичні або динамічні входи. Як статичні, так і динамічні, входи можуть бути прямими або інверсними.

Входи називаються статичними, якщо вони мають безпосередній зв'язок з джерелом вхідних сигналів. Сигналом для управління статичним тригером з прямими статичними входами є рівень логічної «1», а для управління тригером з інверсними входами – рівень логічного «0».

Входи називаються динамічними, якщо вони сполучені з джерелом вхідних сигналів через розв'язуючі кола: магнітні, електронні або RC – кола. Вони реагують лише на перепади вхідних сигналів. Якщо спрацювання тригера відбувається при зміні вхідного сигналу від «0» до «1», то входи називаються прямими, а якщо при зміні сигналу від «1» до «0», то – інверсними.

Входи \bar{S} і \bar{R} називаються входами асинхронної установки тригера. Вони призначені для подачі пріоритетних сигналів установки тригера у вихідний стан («0» або «1») на початку циклу роботи незалежно від дії інформаційних сигналів, тобто в обхід схеми управління.

4.2.1 RS – тригери

Найбільшого використання знаходять тригери з роздільним запуском, які називаються RS – тригерами. Їх умовне графічне позначення приведені на рисунку 4.3.

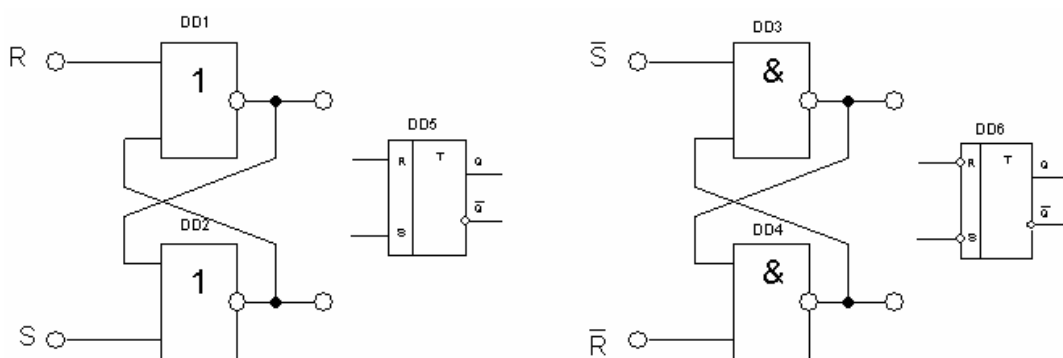


Рисунок 4.3 – RS – тригери та їх умовне графічне позначення (УГП)

У простому RS – тригері інформаційні сигнали подаються безпосередньо на входи S і R елемента пам'яті. Входи, на які подаються імпульси запуску, на-

зиваються установочними. Літерою S (Set – установка) позначають вхід, на який подається сигнал, що встановлює тригер в одиничний стан ($Q = 1, \bar{Q} = 0$). Літерою R (Reset – скидання) позначають вхід, на який подається сигнал скидання, що переводить тригер в стан «0» ($Q = 0, \bar{Q} = 1$). Літерою Q позначається прямий вихід, а \bar{Q} - інверсний.

RS – тригери застосовуються як самостійно, так і у складі інших складніших тригерів, а також входять до складу регістрів і лічильників.

Функціонування логічних пристроїв послідовнісного типу описується таблицями переходів, які відрізняються від таблиць істинності тим, що в них враховуються лише результативні переходи, коли зміна комбінації сигналів на вході призводить до зміни вихідного стану. Проте, таблиця переходів може бути зведена до таблиці істинності, якщо стан внутрішніх елементів пам'яті вважати вхідними сигналами.

Повна таблиця функціонування (таблиця істинності) приведена на рисунку 4.4 а. У таблиці 4.4а попередній стан тригера Q^n до подачі вхідних сигналів є одним з вхідних сигналів. Вихідний стан тригера після подачі вхідних сигналів позначений символом Q^{n+1} . Таблиця переходів тригера приведена на рисунку 4.4 б.

Таблиця істинності дозволяє застосувати методику синтезу логічних пристроїв комбінаційного типу для синтезу пристроїв послідовнісного типу, у тому числі і RS – тригерів.

Відповідно до таблиці істинності:

$$Q^{n+1} = \bar{S}\bar{R}\bar{Q}^n + \bar{S}RQ^n + SRQ^n$$

Для мінімізації структурної формули RS – тригера заповнимо карту Карно. Відповідно до теорії мінімізації невизначених логічних функцій, для визначення прямого значення функції Q^{n+1} невизначені значення карти Карно «×» (рис. 4.5 а) замінимо на «1» (рис. 4.5 б), а для визначення інверсного значення

\bar{Q}^{n+1} – замінимо на «0» (рис. 4.5 в). Для отримання функції \bar{Q}^{n+1} мінімізація виконується по нулях.

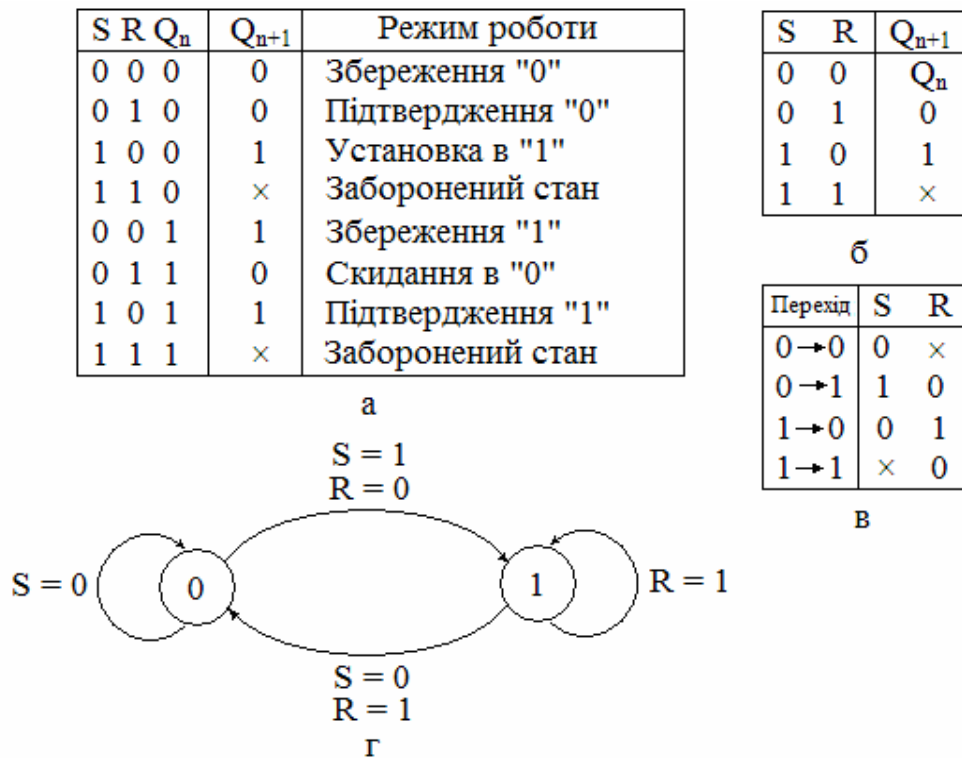


Рисунок 4.4 – Таблиця істинності (а), таблиця перемикач (б), словник переходів (в), діаграма станів (г) RS – тригера

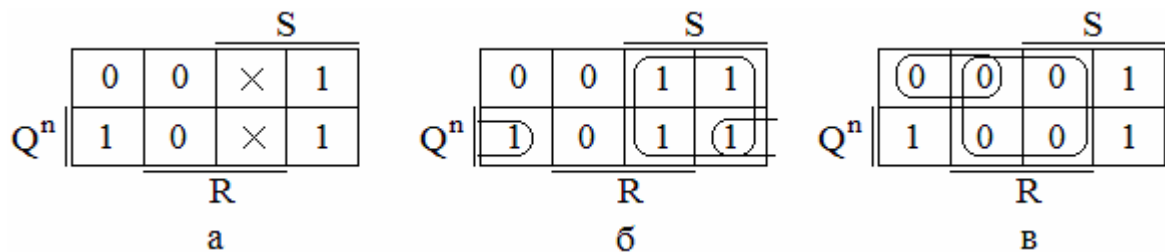


Рисунок 4.5 – Мінімізація структурної формули RS – тригера

Мінімізовані значення функцій Q^{n+1} і \bar{Q}^{n+1} на елементах основного базису мають вигляд: $Q^{n+1} = S + \bar{R}Q^n$; $\bar{Q}^{n+1} = R + \bar{S}\bar{Q}^n$.

При розробці цифрових схем, в які входять RS – тригери, необхідно враховувати наявність забороненого стану вхідних сигналів для RS – тригерів на елементах АБО-НІ $S = R = 1$, а для RS – тригерів на елементах І-НЕ $\bar{S} = \bar{R} = 0$. Умову нормального функціонування для обох схем RS – тригерів можна запи-

сати в наступному вигляді: $SR \neq 1$. Якщо в схемі, що розробляється, таке поєднання вхідних сигналів в принципі можливо, то цю ситуацію необхідно виключити шляхом включення у вхідне коло додаткових логічних елементів, або використовувати інші типи тригерів, що не мають забороненого стану.

Розглянуті RS – тригери є асинхронними, оскільки сигнали управління впливають на тригер безпосередньо з початком своєї появи на їх входах.

4.2.2 Синхронні RS – тригери

У пристроях сучасної цифрової техніки для виключення небезпечної конкуренції вхідних сигналів спрацьовування всіх вузлів і елементів в кожному такті повинне відбуватися строго одночасно. Для досягнення цієї мети застосовується жорстка синхронізація за допомогою спеціальних синхроімпульсів. Для роботи в схемах з синхронізацією режиму розроблені синхронні RS – тригери (RST– тригери). Особливістю синхронного тригера є те, що в схемі управління присутні інвертуючі елементи і відбувається зміна виконавчого значення сигналів управління, в порівнянні з асинхронними.

Синхронні RS – тригери мають три входи: S, R і C (рис. 4.6).

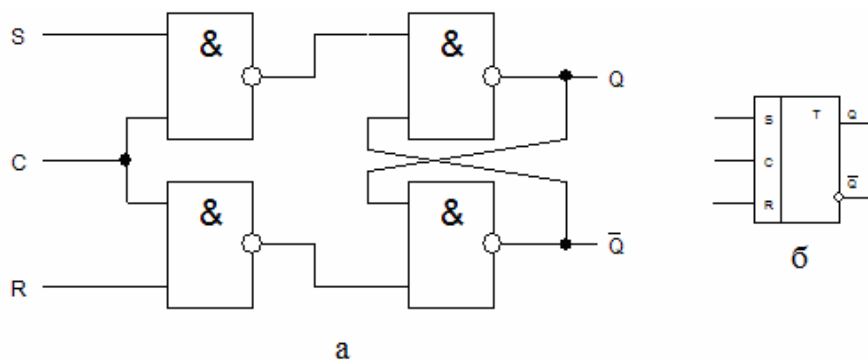


Рисунок 4.6 – Синхронний RS – тригер (а) та його УГП (б)

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

Використання синхронізації не усуває невизначений стан тригера, що виникає при одночасній подачі одиничних сигналів на всі три входи. Тому умовою нормального функціонування є наступна нерівність: $SRC \neq 1$.

Окрім трьох основних входів, синхронні RS – тригери забезпечуються ще і входами асинхронної установки стану тригера – \bar{S} і \bar{R} (рис. 4.7). По своїй дії на стан тригера входи \bar{S} і \bar{R} є головними і тому на УГП відділяються від останніх сигналів горизонтальною лінією.

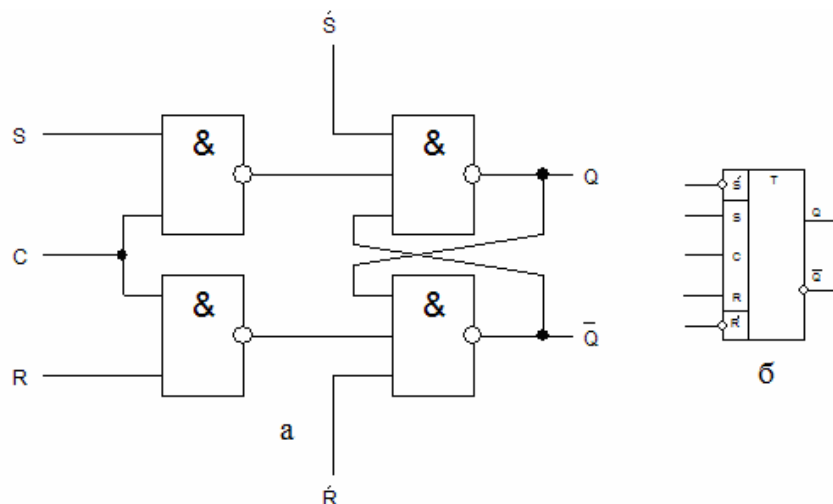


Рисунок 4.7 – RST– тригер з входами асинхронної установки

4.2.3 RS – тригери S, R і E – типів

На відміну від звичайних RS – тригерів, в тригерах S, R і E – типів комбінація сигналів $S = R = 1$ не є забороненою. При різнополярних сигналах алгоритм роботи тригерів S, R і E – типів такий же, як і в звичайних RS – тригерах, але при $S = R = 1$ тригер S – типу перемикається в «1», тригер R – типу в «0», а тригер E – типу не змінює свого стану (рис. 4.8).

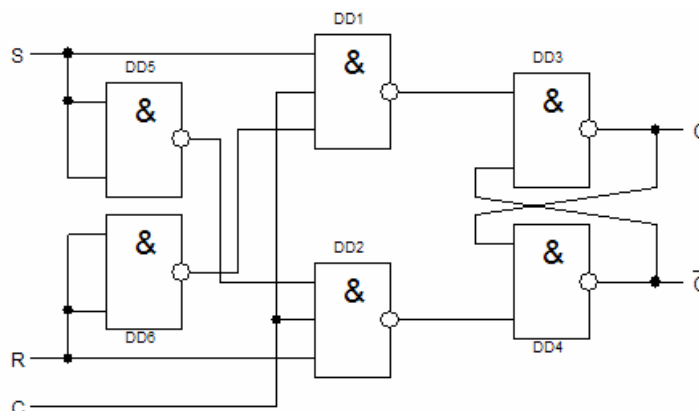


Рисунок 4.8 – RS – тригер E – типу

Схема працює як звичайний RS – тригер, але при подачі сигналів $S = R = 1$ вентилі DD5 і DD6 забезпечують закритий стан елементів DD1 і DD2, тому вихідний стан тригера Q залишається без зміни.

Якщо виключити зі схеми (рис. 4.8) елемент DD6, то при подачі на вхід сигналів $S = R = 1$ блокується лише елемент DD2, на виході якого встановлюється «1», а на виході DD1 формується «0». Ці сигнали встановлюють тригер в стан $Q = 1$, або підтверджують його, якщо до подачі сигналів $S = R = 1$ тригер знаходився в стані $Q = 1$. Такий тригер називається RS – тригером S – типу.

Якщо виключити зі схеми (рис. 4.8) елемент DD5, залишивши елемент DD6, то при подачі на вхід сигналів $S = R = 1$ блокується лише елемент DD1, тому тригер встановлюється в стан $Q = 0$ або підтверджує його, якщо до подачі сигналів $S = R = 1$ тригер знаходився в стані $Q = 0$. Такий тригер називається RS – тригером R – типу.

4.2.2 D – тригери (тригери затримки)

D – тригери – це електронні пристрої з двома стійкими вихідними станами і одним інформаційним входом D. Для D – тригера скорочена таблиця істинності, словник переходів і діаграма станів приведені на рисунку 4.9.

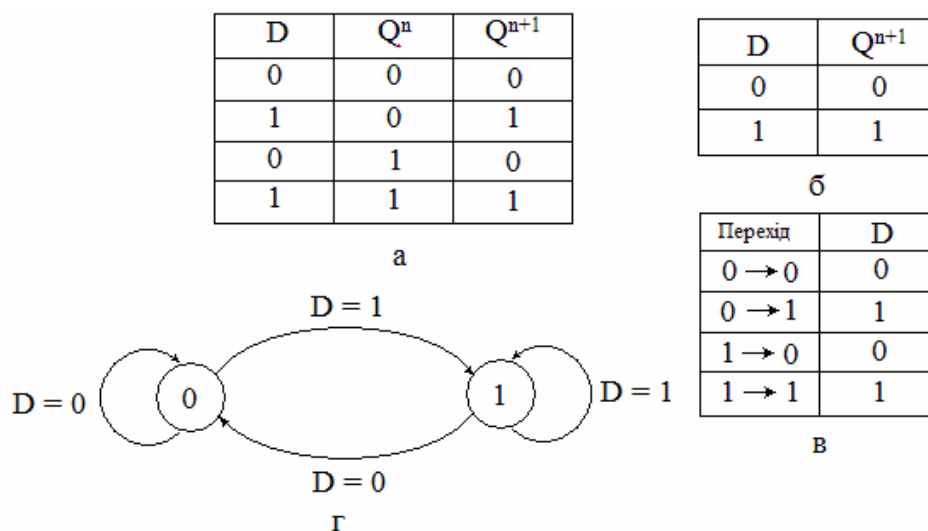


Рисунок 4.9 – Таблиця істинності (а), таблиця перемикань (б), словник переходів (в), діаграма станів (г) D-тригера

Характеристичне рівняння тригера: $Q^{n+1} = D_n$. Воно означає, що логічний сигнал Q^{n+1} повторює значення сигналу, встановлене на вході тригера в попередній момент часу.

Завдяки підключенню елемента DD1 на входи RS – тригера поступають різнополярні сигнали (рис. 4.10 а), тому заборонений стан входних сигналів виключений, але час затримки поширення сигналу елемента DD1 має бути менший, ніж у елементів DD2 і DD3.

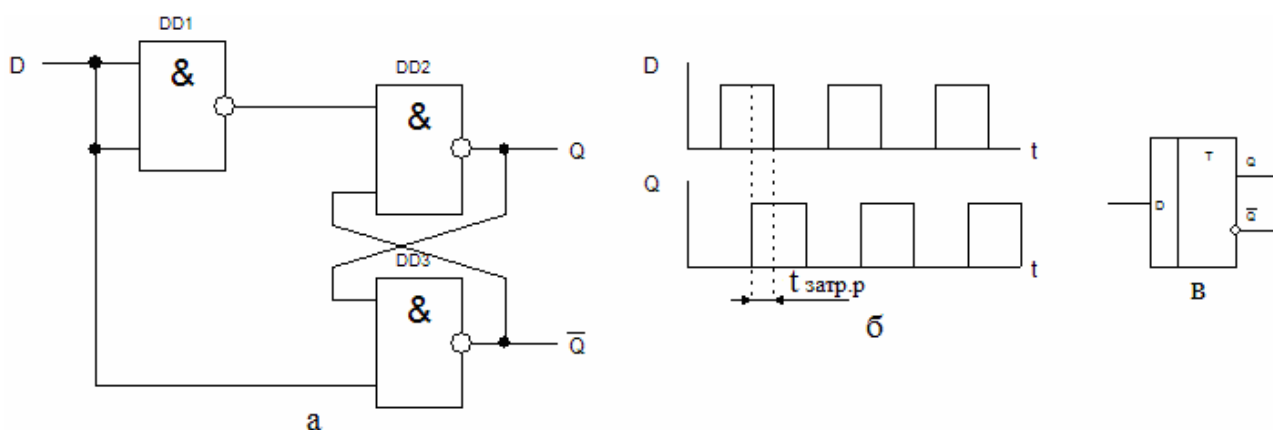


Рисунок 4.10 – Асинхронний D – тригер (а), затримка розповсюдження імпульсу (б) і умовне графічне позначення (в)

У приведеній вище схемі D – тригера внаслідок затримки поширення сигналів сигнал на виході Q з'являється з певною затримкою, як показано на рисунку 4.10 (б). Таким чином, в асинхронному D – тригері затримка визначається параметрами елементів схеми.

Тактований D – тригер (рис. 4.11) встановлюється в стан $Q^{n+1} = D^n$ лише з приходом тактового імпульсу $C = 1$, тому затримка тактового D – тригера визначається часом приходу тактового імпульсу.

У схемі D – тригера часто паралельно входу C виготовляється ще один вхід V, як показано на рисунку 4.12 (а). Такий тригер називається DV – тригером. При $V = 1$ DV – тригер працює як звичайний D – тригер, а при $V = 0$ немов би замикається і зберігає раніше записану інформацію.

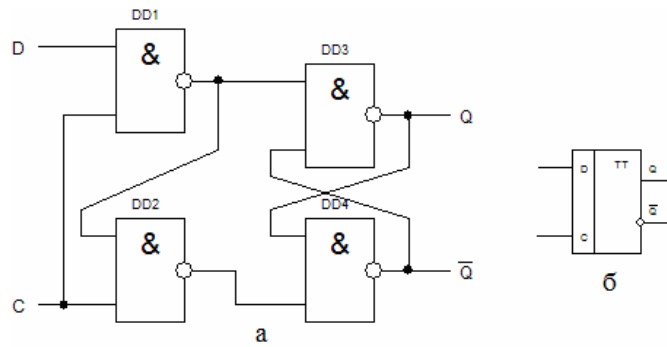


Рисунок 4.11 – Тактовий D – тригер (а) та його УГП (б)

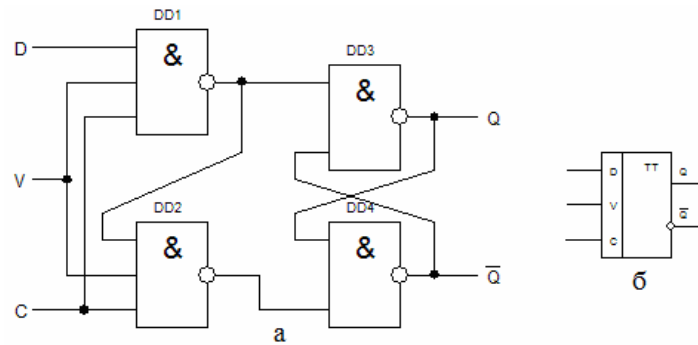


Рисунок 4.12 – Тактовий DV – тригер (а) та його УГП (б)

4.2.3 Тригер Т – типу (рахунковий тригер)

Т – тригер, або рахунковий тригер, має один рахунковий вхід Т і два виходи (рис. 4.13 а).

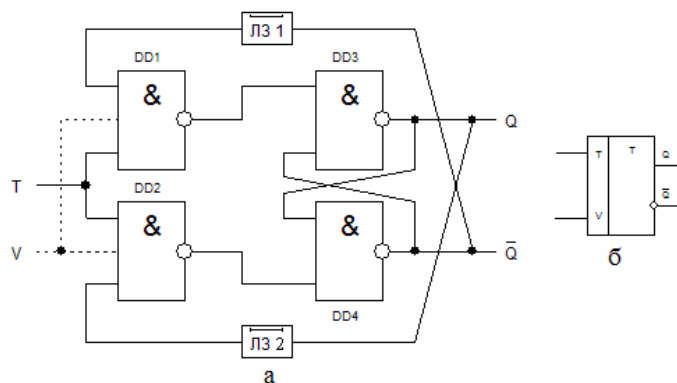


Рисунок 4.13 – Т – тригер: а) структурна схема, б) УГП TV – тригера

Повна таблиця функціонування Т – тригера приведена на рисунку 4.14, у якій попередній стан тригера Q^n – стан до подачі вхідних сигналів, Т вхідний

сигнал. Вихідний стан тригера після подачі вхідних сигналів позначений символом Q^{n+1} .

Відповідно до таблиці функціонування:

	Q^{n+1}	T	
		0	1
Q^n		1	0

Рівняння функціонування T – тригера:

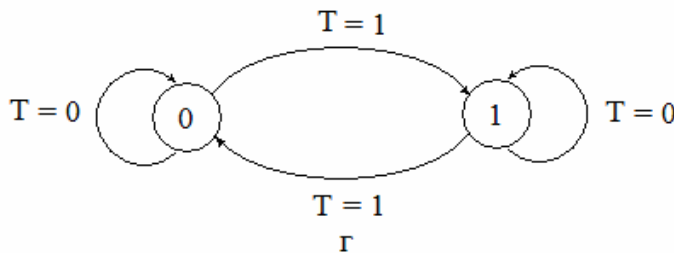
$$Q^{n+1} = T\bar{Q}^n + \bar{T}Q^n$$

T	Q^n	Q^{n+1}
0	0	0
1	0	1
0	1	1
1	1	0

а

T	Q^{n+1}
0	Q^n
1	\bar{Q}^n

б



г

Перехід	T
$0 \rightarrow 0$	0
$0 \rightarrow 1$	1
$1 \rightarrow 0$	1
$1 \rightarrow 1$	0

в

Рисунок 4.14 – Таблиця істинності (а), таблиця перемикань (б), словник переходів (в), діаграма станів (г) T – тригера

З рівняння випливає, що T – тригер кожного разу змінює свій стан на протилежний з приходом на рахунковий вхід T чергового тактуючого імпульсу тривалістю t_i . Цьому сприяє наявність перехресних зворотних зв'язків з виходів тригера на входи елементів $DD1$ і $DD2$. Для надійної роботи тригера, з метою збереження інформації про попередній стан тригера у момент його перемикання, в схему вводять елементи затримки, що мають час затримки $t_3 > t_i$.

Нехай у початковому стані $Q = 1$. Сигнал $T = 1$ відкриє елемент $DD2$, оскільки на другому вході $DD2$ є сигнал логічної одиниці («1») з виходу Q , а елемент $DD1$ буде закритий. Тригер переходить в стан $Q = 0$. Вентиль $DD2$ залишається відкритим протягом часу t_i , оскільки сигнал $Q = 1$ буде затриманий

лінією затримки (ЛЗ) 1 на якийсь час $t_3 > t_i$. В той же час, сигнал $\bar{Q} = 1$ не потрапить на вентиль DD1 із-за тимчасової затримки в ЛЗ 2. За відсутності елементів затримки можливе неодноразове перемикання тригера за умови, якщо тривалість імпульсу t_i значно перевищує час перемикання тригера.

Після закінчення дії тактуючого імпульсу, елементи DD1 і DD2 закриються, оскільки потенціал входу $T = 0$. Після чого на вхід елементу DD1 через ЛЗ 2 надійде сигнал $\bar{Q} = 1$. В результаті з приходом другого імпульсу $T = 1$ відкриється елемент DD1 і тригер перемкнеться в стан $Q = 1$ і так далі.

Окрім рахункового входу T – тригер може мати вхід дозволу V (рис. 4.13). Сигнал на цьому вході дозволяє (при $V = 1$) або забороняє (при $V = 0$) спрацьовування тригера від сигналів, що поступають на вхід T . T – тригери, що мають додатковий вхід V , називаються TV – тригерами. Наявність входу V дозволяє організувати рахунок в заданому часовому інтервалі, що істотно розширює функціональні можливості T – тригера.

Рахунковий T – тригер може бути побудований на основі D – тригера з'єднанням інверсного виходу \bar{Q} з входом D , як показано на рисунку 4.15. У такій схемі кожен перехід «1 → 0» на вході C буде призводити до переходу тригера в протилежний стан.

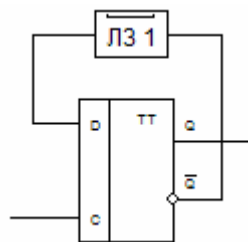


Рисунок 4.15 – T – тригер на основі D – тригера

Наприклад, якщо $Q^n = 1$, то $D^n = \bar{Q}^n = 0$, і тому черговий тактовий імпульс переведе тригер в новий стан, тобто встановить $Q^{n+1} = D^n = 0$. Для правильної роботи T – тригера тактовий імпульс має бути коротким, а наявність елементу затримки ЛЗ ослабляє вимогу до тривалості тактового імпульсу.

T – тригер може бути побудований на RST – тригерах (рис. 4.16).

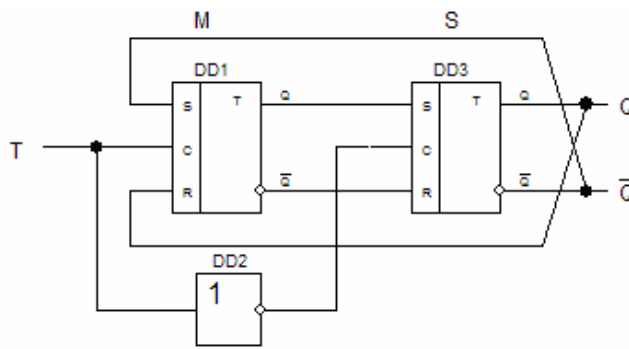


Рисунок 4.16 – Т – тригер на основі RST – тригерів

Схема, складається з двох послідовно включених синхронних RST – тригерів, перший з яких називається ведучим, або М – тригером (master), а другий – веденим, або S – тригером (slave). Завдяки загальному синхросигналу С вся схема функціонує як єдине ціле і називається двоступінчатим, або MS – тригером. Інформація, що задається рівнями на входах S і R, по фронту С – сигналу приймається в М – тригер, але протягом всього часу, поки С – сигнал дорівнює 1, не проходить в S – тригер, оскільки його робота блокована інверсією С – сигналу. Вони відкриваються лише при $C = 1$, тобто по спаду С - сигналу, і лише тоді S – тригер прийме стан М – тригера. Сказане ілюструє дуже важливу відмінність MS – тригера від DV – тригера: MS – тригер, зібраний за схемою (рис. 4.16), непрозорий по керівних R і S входах ні при $C = 0$, ні при $C = 1$. Кожен рівень його сам по собі прозорий, але включені рівні послідовно, і який-небудь один із них завжди виявляється замкнутим або синхросигналом, або його відсутністю. Таким чином, в цьому MS – тригері при $C = 1$ (і тим більше при $C = 0$) жодна зміна на вході управління не може сама по собі, без перемикання С – сигналу, потрапити на вихід. Тригер може змінити стан виходу лише по спаду С – сигналу. У зарубіжній літературі непрозорі тригери називають flip-flop на відміну від прозорих DV – тригерів, за якими закріпився термін latch.

Таким чином, з розгляду принципу роботи Т – тригера виходить, що при $T = 1$ спадаючий фронт сигналу на вході Т переводить тригер в протилежний стан (рис. 4.17).

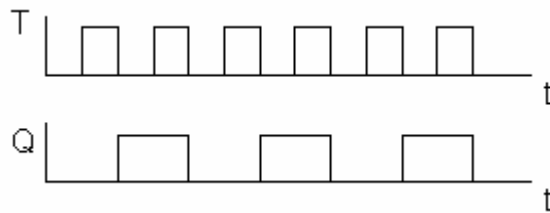


Рисунок 4.17 – Діаграма функціонування Т – тригера

Частота зміни потенціалу на виході Т – тригера в два рази менше частоти імпульсів на вході Т. Ця властивість Т – тригерів дозволяє будувати на їх основі двійкові лічильники. Тому ці тригери і називають рахунковими.

4.2.4 JK – тригери

JK – тригер – це схема з двома стійкими вихідними станами і двома входами J і K. Подібно RS – тригеру, в JK – тригері входи J і K це входи установки виходу Q тригера в стан 1 або 0. Проте, на відміну від RS – тригера, в JK – тригері наявність $J = K = 1$ наводить до переходу виходу Q тригера в протилежний стан (рис. 4.18).

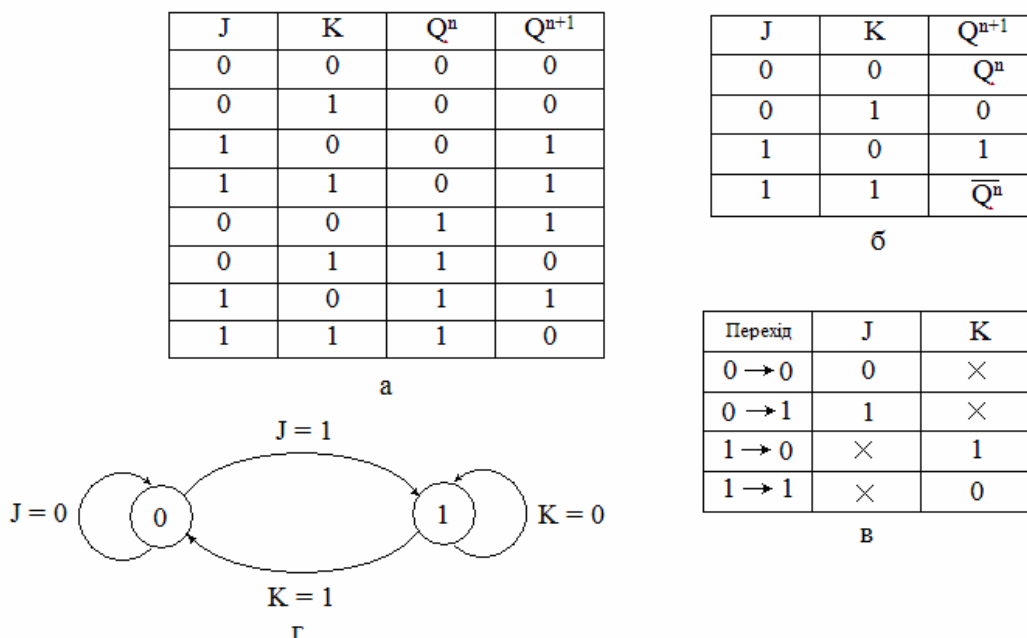


Рисунок 4.18 – Таблиця істинності (а), таблиця перемикань (б), словник переходів (в), діаграма станів (г) JK – тригера

Відповідно до таблиці функціонування:

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

Рівняння функціонування JK – тригера:

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

JK – тригери, які тактуються фронтом тактових імпульсів і не чутливі до тривалості тактових імпульсів, в основному, розробляються і застосовуються в інтегральному виконанні. JK – тригери, що тактуються фронтом, будуються за схемою представленою на рисунку 4.19.

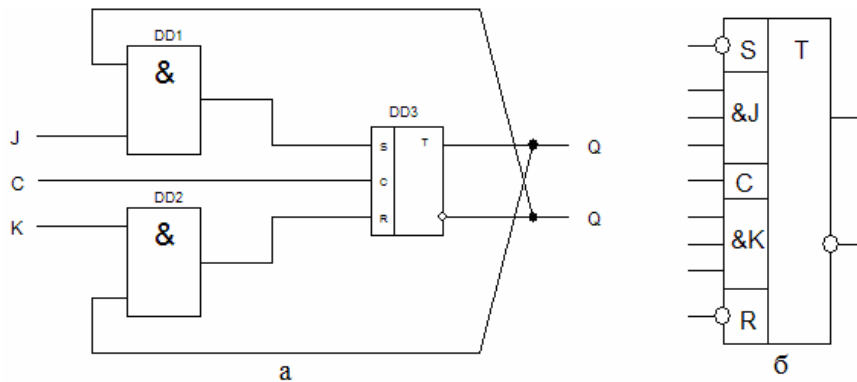


Рисунок 4.19 – JK – тригер: а) структурна схема, б) УГП JK – тригера

Тригер JK –типу називають універсальним тому, що на його основі за допомогою нескладних комутаційних перетворень можна отримати RS і T – тригери, а якщо між входами J і K включити інвертор, то вийде схема D – тригера (рис. 4.20).

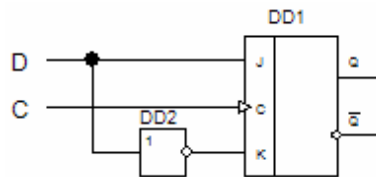


Рисунок 4.20 – D – тригер з JK – тригера

По схемі на вхід J поданий інформаційний сигнал D, а на вхід K йому подано інверсний сигнал \bar{D} . Згідно з рівнянням функціонування JK – тригера:

$$Q^{n+1} = J\bar{Q}^n + \bar{K}Q^n = D\bar{Q}^n + \bar{\bar{D}}Q^n = D \cdot (\bar{Q}^n + Q^n) = D \cdot 1 = D,$$

що відповідає рівнянню функціонування D – тригера.

Для отримання T – тригера досить об'єднати входи J і K і подавати на них загальний інформаційний сигнал T (рис. 4.21).

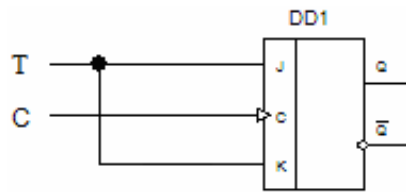


Рисунок 4.21 – T – тригер з JK – тригера

Згідно рівняння функціонування JK – тригера:

$$Q^{n+1} = J\bar{Q}^n + \bar{K}Q^n = T\bar{Q}^n + \bar{T}Q^n,$$

Тригер буде перемикатися в протилежний стан з приходом кожного тактового імпульсу C.

4.3 Цифрові регістри

Цифрові регістри – це пристрої, призначені для зберігання і перетворення багаторозрядних двійкових чисел. Тригери – запам'ятовуючі елементи регістрів. Їх кількість дорівнює розрядності чисел, що зберігаються. Окрім тригерів регістри містять також комбінаційні схеми, призначені для введення і виведення чисел, що зберігаються, перетворення їх кодів, зсування кодів на те або інше число розрядів. Інформація в регістрах зберігається, як правило, протягом деякої кількості тактів.

Залежно від способу прийому і видачі інформації регістри діляться на групи: з паралельним прийомом і видачею інформації (регістри пам'яті); з послідовним прийомом і видачею інформації; з послідовним прийомом і паралельною видачею інформації; з паралельним прийомом і послідовною видачею інформації і комбіновані (регістри зсуву).

У регістрах пам'яті число вводиться (виводиться) за один такт, а в регістрах зсуву – за n тактів, де n – розрядність чисел.

За способом введення-виведення регістри підрозділяються на однофазні і парафазні. В однофазних регістрах введення (і виведення) виробляється лише в прямому, або лише в зворотному коді. В парафазних регістрах можливе введення і виведення як в прямому, так і в зворотному кодах.

У паралельних регістрах можна виконувати порозрядні логічні операції з числом, що зберігається, і що знов вводиться. Вигляд логічних операцій залежить від типу тригерів, що складають регістр, і комбінації сигналів управління.

Регістри зсуву застосовуються для перетворення послідовного коду в паралельний (і навпаки), для перемноження і ділення багаторозрядних чисел і так далі.

4.3.1 Паралельні регістри (регістри пам'яті)

Паралельні регістри - це пристрої, призначені для запису, зберігання і надання інформації, представленої у вигляді двійкового коду. 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.

Структурна схема регістра цього типу представлена на рисунку 4.22. Запис інформації в паралельні регістри здійснюється паралельним кодом, тобто у всі розряди регістра одночасно. Їх функція зводиться лише до прийому, зберігання і передачі інформації. У зв'язку з цим паралельні регістри називають регістрами пам'яті.

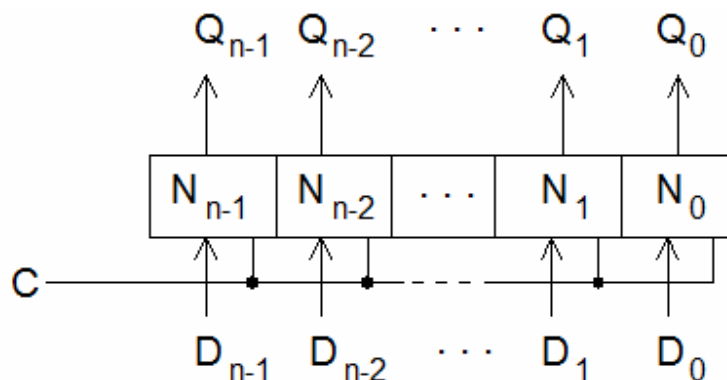


Рисунок 4.22 – Структурна схема паралельного регістра

Інформація (число) записується у всі розряди регістра, у всі D – тригери одночасно, тобто паралельним кодом. При подачі управляючого сигналу $C = 1$, інформація по входах $D1 – D4$ записується одночасно у відповідні розряди чотирьох D – тригерів (рис. 4.23).

При $C = R = 0$ інформація зберігається в регістрі пам'яті, а при $R = 1$ відбувається паралельне зчитування інформації.

Умовним зображенням регістра на рисунку 4.24 (а) користуються тоді, коли на схемі необхідно показати кожен вхід і вихід даних. Якщо ж тракт даних розглядається як єдине, збільшене поняття (шина даних), то користуються позначенням, показаним на рисунку 4.24 (б).

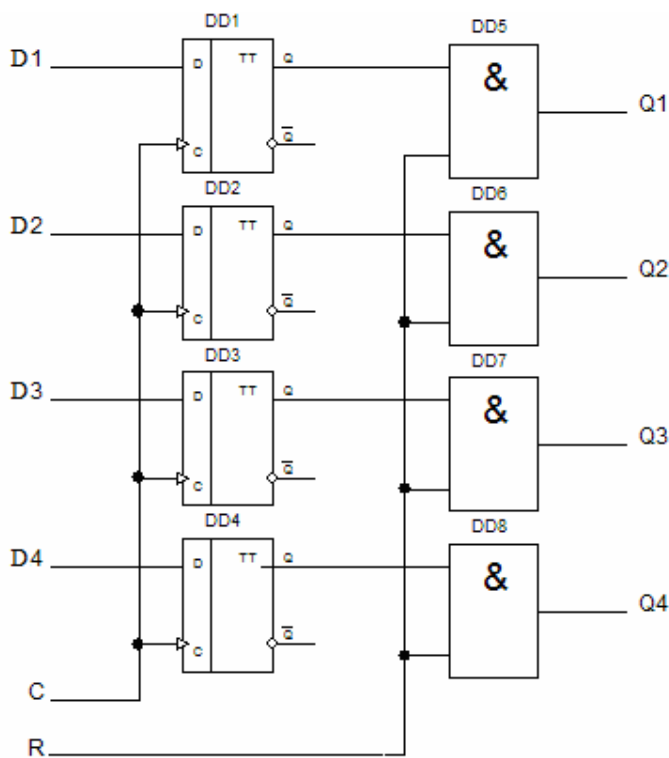


Рисунок 4.23 – Паралельний регістр на тактованих D – тригерах

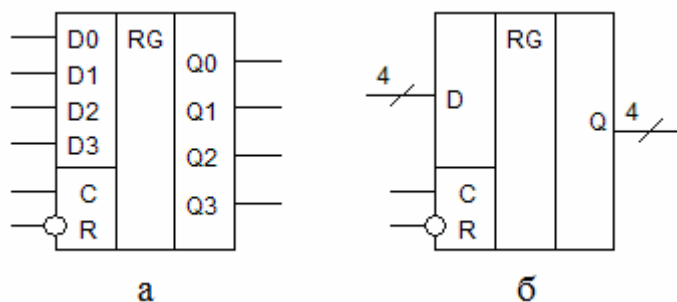


Рисунок 4.24 – Умовне позначення паралельного регістра

Регістри, що випускаються промисловістю, інколи об'єднують на кристалі мікросхеми з іншими вузлами, в парі з якими регістри часто використовуються в схемах цифрової апаратури.

4.3.2 Послідовні регістри (регістри зсуву)

У регістрі з послідовним введенням виконується послідовний зсув інформації, що надходить на вхід, на один розряд вправо в кожному такті синхросигналів (рис. 4.25).

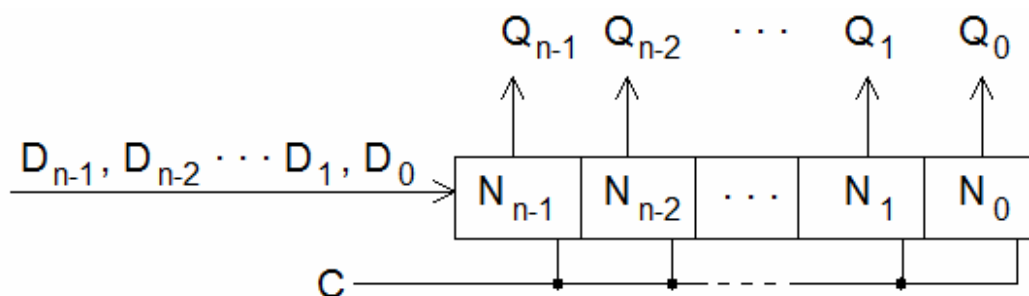


Рисунок 4.25 – Структурна схема регістра зсуву

Після проходження m синхроімпульсів весь регістр виявляється заповненим розрядами числа D , і перший розряд числа (D_0) з'являється на виході Q_0 регістра. Протягом подальших m синхроімпульсів виконується послідовний порозрядний вивід з регістра записаного числа, після чого регістр виявляється повністю очищеним.

Регістри зсуву є колом послідовно включених D – тригерів або RS і JK – тригерів, які включені в режим D – тригера. Поява імпульсу на тактовому вході регістра зсуву викликає переміщення записаної в ньому інформації на один розряд вправо 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 4-розрядного регістра зсуву приведена на рисунку 4.26. 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 зсув інформації вправо.

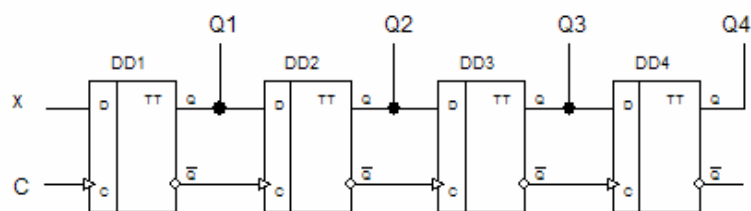


Рисунок 4.26 – Чотирьохрозрядний регістр зсуву

Вхід «D» першого тригера служить для прийому в регістр вхідної інформації у вигляді послідовного коду. З кожним тактовим імпульсом на цей вхід повинен подаватися код нового розряду вхідної інформації (рис. 4.27).

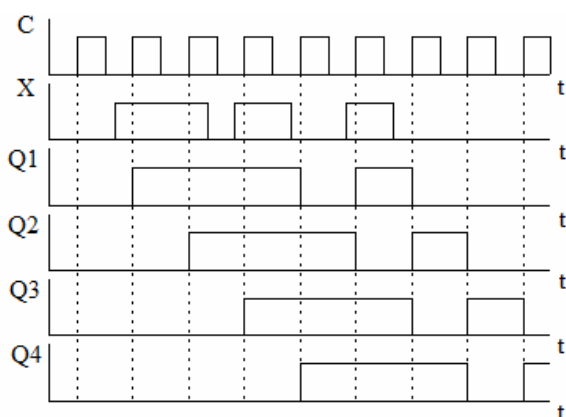


Рисунок 4.27 – Діаграма функціонування чотирьохрозрядного регістра зсуву

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 зсув числа на один розряд. Чотирьохрозрядне число буде повністю записано в регістр після четвертого тактового імпульсу. До приходу наступного тактового імпульсу це число зберігається в регістрі у вигляді паралельного коду на виводах Q1 ... Q4. Якщо необхідно отримати інформацію в послідовному коді, то її знімають з виходу Q4 в моменти приходу наступних трьох тактових імпульсів. Такий режим називається режимом послідовного зчитування. У двійковій системі числення при зсуві числа на один розряд у бік старших розрядів відбува-

ється збільшення числа в два рази. При зсуві числа у бік молодших розрядів число записується в регістр і зменшується в два рази. Таким чином, регістр зсуву можна використовувати для множення або ділення числа на 2^n , де n – кількість зсувів, що дорівнює кількості тактових імпульсів.

При побудові регістрів зсуву обов'язковим є використання тригерів, які синхронізуються фронтом. Інакше за час дії одного синхросигналу інформація в регістрі просувається більш ніж на один розряд, тобто нормальне функціонування регістра зсуву на один розряд за один такт порушується.

Дуже зручні універсальні регістри, що дозволяють виконувати як послідовний, так і паралельний запис і зчитування. Такі регістри можна використовувати як перетворювачі паралельного коду в послідовний і навпаки. Наприклад, мікросхема K555IP1 – чотирьохрозрядний універсальний регістр зсуву (рис. 4.28).

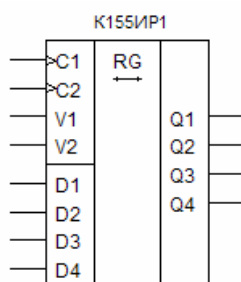


Рисунок 4.28 – Чотирьохрозрядний універсальний регістр зсуву

Регістр працює в режимі зсуву по тактових імпульсах, які поступають на вхід C1, якщо на вході V2 є напруга низького рівня. Вхід V1 служить для введення інформації в перший розряд регістра в цьому режимі. Якщо на вході V2 напруга високого рівня, то регістр виконує паралельний запис інформації з входів D1 ... D4, по імпульсах синхронізації, що поступають на вхід C2.

4.3.3 Реверсивні регістри зсуву

Реверсивні регістри зсуву забезпечують можливість зсуву інформації як вправо, так і вліво. Вони мають спеціальний вхід M управління напрямом зсуву (рис. 4. 29).

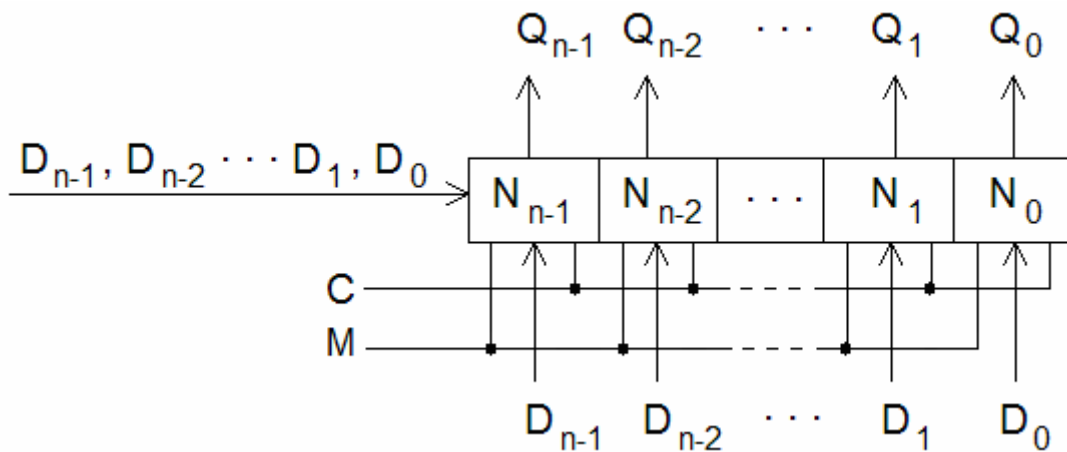


Рисунок 4.29 - Реверсивний регістр зсуву (універсальний)

Логічні елементи здатні передавати сигнали лише в одному напрямі з входу на вихід (зліва направо). Для зсуву інформації вліво, необхідно інформацію з виходу подальших тригерів по спеціально створених колах подавати на входи попередніх тригерів і записувати їх наступним тактовим сигналом. Це еквівалентно зсуву інформації вліво. Фрагмент функціональної схеми реверсивного регістра зсуву приведений на рисунку 4.30.

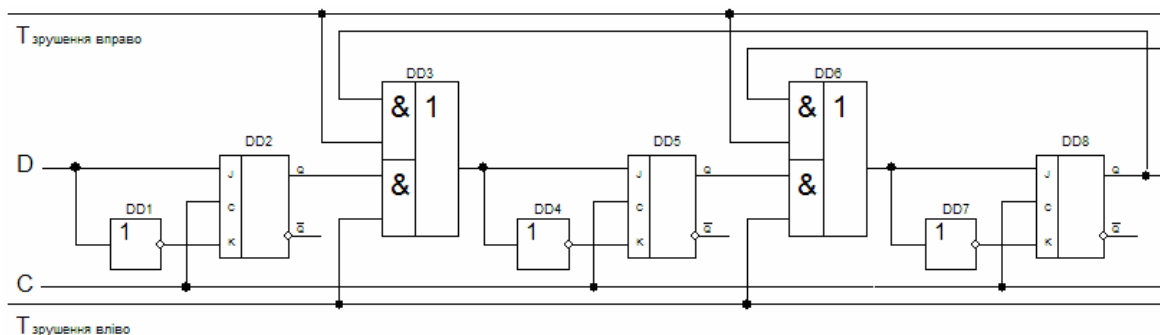


Рисунок 4.30 - Реверсивний регістр зсуву

Пристрої управління напрямом зсуву залежно від одиничного сигналу, що поступив або по шині $T_{\text{зсуву вправо}}$, або по шині $T_{\text{зсуву вліво}}$, повинні підключати входи кожного тригера регістра до виходів попереднього або до виходів подальшого тригерів.

Функціональні можливості регістра зсуву можна розширити, якщо його доповнити входами паралельного завантаження і виходами всіх розрядів регістра для паралельної видачі інформації. У регістрах зсуву між кожними парами

тригерів можуть бути вбудовані мультиплексори, що дозволяють подавати на їх входи зовнішні сигнали. Такі регістри називаються завантажуваними регістрами зсуву з паралельним доступом.

4.3.4 Паралельно – послідовний регістр

У паралельно – послідовному регістрі запис інформації відбувається в паралельному коді, а зчитування в послідовному. На рисунку 4.31 показаний приклад побудови такого регістра на JK – тригерах.

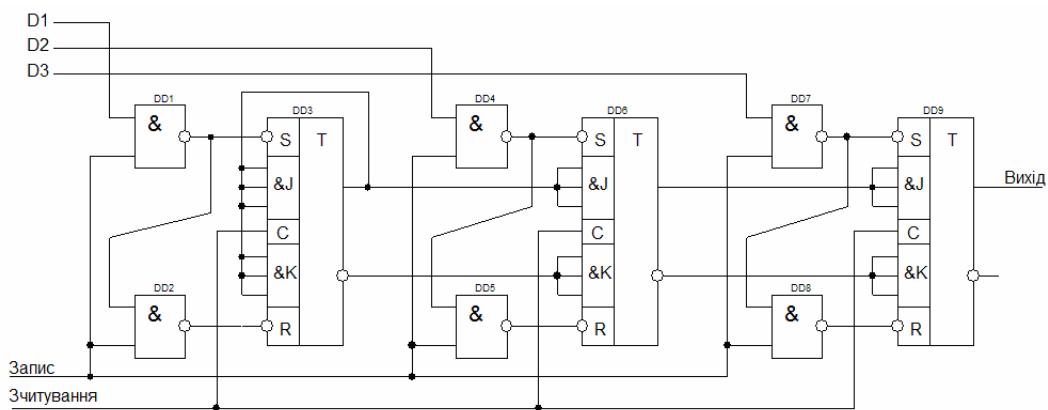


Рисунок 4.31 – Паралельно – послідовний регістр

У приведеній схемі JK – тригери двічі перетворені в D – тригери. Одне перетворення здійснене для організації запису з використанням допоміжних інверсних входів S і R і додаванням двох елементів І-НІ (рис. 4.11).

Запис числа в регістр в паралельному коді відбувається при подачі числа на входи D1, D2, D3 і сигналу «1» на вхід «Запис». Для зчитування інформації з регістра в послідовному коді JK – тригери другий раз перетворені в D – тригери, в яких виходи кожного тригера сполучені з входом D наступного, як це робиться в послідовному регістрі. У другому випадку JK – тригери перетворені в D – тригери у спосіб, представлений на рисунку 4.20. Але замість додаткового інвертора використовується інверсний вихід попереднього тригера. Виключення складає перший тригер, в якого входи J і K об'єднані і сполучені з основним входом цього ж тригера. Завдяки такій схемі, після подачі імпульсів зчитування перший тригер виявляється в стані логічного «0».

4.3.5 Послідовно – паралельний регістр

Для побудови послідовно – паралельного регістра достатньо в послідовному регістрі організувати паралельне зчитування, використовуючи додаткові елементи І – НІ, як це показано на рисунку 4.32.

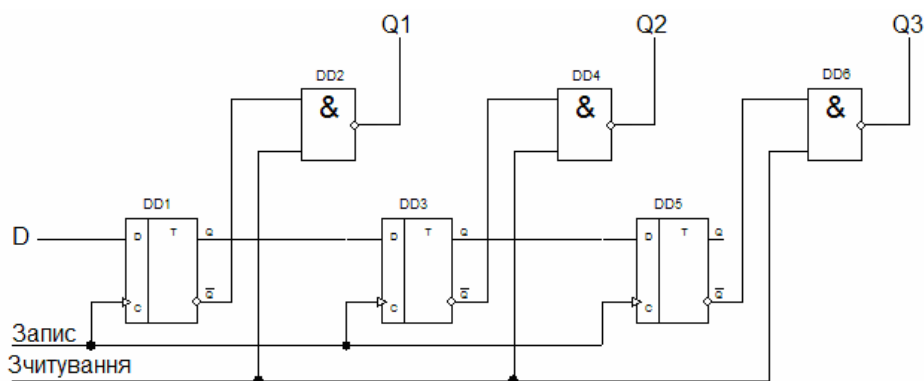


Рисунок 3.8 – Послідовно – паралельний регістр

При подачі сигналу «1» на вхід «Зчитування» значення розрядів числа D з інверсних виходів тригерів поступають на виходи Q1, Q2, Q3 через елементи І – НІ.

4.4 Цифрові лічильники

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

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

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.

Основна характеристика лічильника – модуль рахунку, або ємність лічильника $K_{\text{рах.}}$. Це кількість вхідних сигналів, які повертають лічильник у вихідний стан. Лічильник, що не має додаткових зв'язків, має модуль рахунку $K_{\text{рах.}} = 2^n$. Лічильники, що мають модуль рахунку 2^n , називаються двійковими. Якщо $K_{\text{рах.}} \neq 2^n$, 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 десятичного числа кодується двійковим кодом.

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

За способом організації внутрішніх зв'язків лічильники можуть бути: з послідовним перенесенням, з паралельним перенесенням, з комбінованим перенесенням, кільцеві.

Лічильники бувають синхронними, тобто, коли рахункові імпульси подаються на рахункові входи всіх тригерів, і асинхронними - коли сигнал на рахунковий вхід якого-небудь тригера подається з виходу одного з тригерів молодших розрядів.

4.4.1 Двійковий асинхронний підсумовуючий лічильник з послідовним перенесенням

Розглянемо роботу двійкового підсумовуючого лічильника з порядком рахунку $K_{\text{рах.}} = 8$. Для його побудови необхідно $m = \log_2 8 = 3$ тригера, що відповідає трьом розрядам двійкового числа (коду 421). У таблиці станів такого лі-

чильника (табл. 4.1) вхідний сигнал x^n позначимо через 1, Q^{n3} – старший розряд, Q^{n1} – молодший розряд.

Таблиця 4.1 – Таблиця станів лічильника

x^n	$Q3^n$	$Q2^n$	$Q1^n$	$Q3^{n+1}$	$Q2^{n+1}$	$Q1^{n+1}$
1	0	0	0	0	0	1
1	0	0	1	0	1	0
1	0	1	0	0	1	1
1	0	1	1	1	0	0
1	1	0	0	1	0	1
1	1	0	1	1	1	0
1	1	1	0	1	1	1
1	1	1	1	0	0	0

З аналізу таблиці видно:

- тригер молодшого розряду $Q1$ перемикається від кожного вхідного сигналу;
- другий розряд $Q2$ перемикається через два вхідні сигнали;
- третій розряд $Q3$ перемикається через чотири вхідні сигнали.

Таким чином, частота перемикання кожного наступного тригера зменшується вдвічі. Отже, лічильник можна побудувати як коло послідовно включених рахункових тригерів. Побудуємо такий лічильник на JK-тригерах, що працюють в рахунковому режимі (рис. 4.33). У схемі JK – тригери перетворені в T – тригери шляхом подачі «1» на вхід J і K.

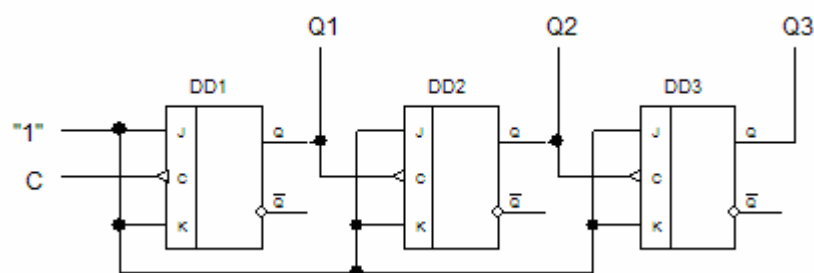


Рисунок 4.33 – Схема двійкового підсумовуючого лічильника з послідовним перенесенням

Оскільки тригери мають інверсний динамічний вхід, то кожен подальший тригер перемикатиметься при скиданні в «0» попереднього тригера (рис. 4.34).

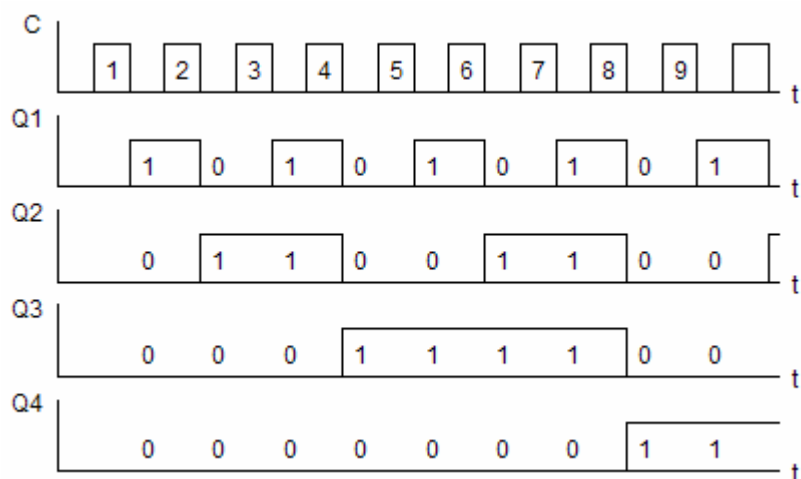


Рисунок 4.34 – Діаграма функціонування двійкового підсумовуючого лічильника з послідовним перенесенням

Окрім основної функції – рахунку імпульсів, лічильник забезпечує ділення частоти дотримання імпульсів. Якщо частоту дотримання імпульсів, що поступають на вхід тригера DD1 позначити f , то на виході тригера DD1 сигнал змінюється з частотою $f/2$, на виході тригера DD2 з частотою $f/4$ і так далі.

Двійковий лічильник з послідовним перенесенням може працювати як віднімаючий. Для цього необхідно сигнали на входи подальших розрядів подавати з інверсних виходів тригерів попередніх розрядів.

Важливим параметром лічильника є його максимальний час встановлення коду, тобто час, необхідний для встановлення коду після подачі рахункового імпульсу. Оскільки отриманий лічильник – асинхронний, то кожен його тригер спрацьовує із затримкою відносно вхідного сигналу. Тому, у міру просування сигналу від молодшого розряду до старшого, ця затримка підсумовується і може статися спотворення інформації у вигляді невідповідності числа імпульсів, що вже поступили в лічильник, і коду на його виходах. У схемах лічильників з послідовним перенесенням максимальний час встановлення $t_{вст.}$ визначається сумою часу затримки перемикання всіх тригерів. Час встановлення $t_{вст.}$ визначає швидкодію лічильника. У загальному випадку сумарна затримка пропорційна числу тригерів, що знижує швидкодію лічильника.

4.4.2 Двійкові лічильники з паралельним перенесенням

Для підвищення швидкодії лічильники виконуються синхронними з паралельним перенесенням (або паралельні). Їх особливість полягає в тому, що виходи всіх попередніх розрядів з'єднуються з входами тригера подальшого розряду, тому тривалість перехідного процесу визначається лише тривалістю перехідного процесу одного розряду і не залежить від кількості тригерів. Звідси випливає, що паралельні лічильники – синхронні.

Структура паралельного лічильника не настільки очевидна, як структура послідовного лічильника. Для її виявлення необхідна певна процедура синтезу. Як приклад, синтезуємо двійковий паралельний лічильник з $K_{\text{рах.}} = 16$.

Підсумовуючий лічильник. Процедура синтезу включає наступні операції.

1) визначається необхідна кількість розрядів m . В даному випадку:

$$m = \log_2 16 = 4.$$

2) будується таблиця станів лічильника (табл. 3.2).

Таблиця 4.2 – Таблиця функціонування двійкового підсумовуючого лічильника з паралельним перенесенням

C	$Q4^n$	$Q3^n$	$Q2^n$	$Q1^n$	$Q4^{n+1}$	$Q3^{n+1}$	$Q2^{n+1}$	$Q1^{n+1}$
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

3) складаються карти Карно для функцій переходів тригерів кожного розряду. Карта переходів будується по таблиці станів і відображає перехід тригера $Q_i^n \rightarrow Q_i^{n+1}$ у кожному такті, залежно від стану останніх тригерів в такті n (рис. 4.35).

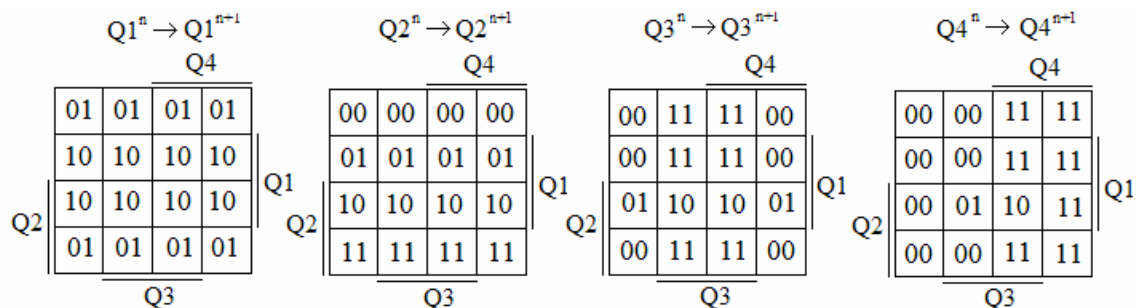


Рисунок 4.35 – Карти Карно для функцій переходів тригерів кожного розряду

Наприклад, першому рядку таблиці 4.2 $Q4 = Q3 = Q2 = Q1 = 0$ відповідає ліва верхня клітинка карт переходів. Оскільки під час вступу першої одиниці в лічильник Q1 він повинен перейти з нульового стану в одиничний, а Q2, Q3 і Q4 повинні зберегти стан нуля, у вказану клітинку карти переходів для Q1 слід поставити «01», а в картах для Q2, Q3 і Q4 поставити «00» і так далі.

4) вибирається тип тригера, наприклад, JK – тригер, для побудови лічильника. Використовуючи словник переходів JK – тригера (рис. 4.18), для кожного входу тригера складаються карти Карно, в клітинках яких проставляються сигнали, необхідні для забезпечення переходів тригерів, вказаних в однойменних клітинках карт функцій переходів (рис. 4.36).

Наприклад, для переходів «01» JK – тригера, згідно його словнику переходів, необхідно подати сигнал $J = 1$, а сигнал на вході K може бути будь-яким «×», тому у верхню ліву клітинку карти Карно для J1 проставляють одиницю, а для K1 – «×» і так далі.

5) проводиться мінімізація логічних функцій входів в картах Карно з метою здобуття їх аналітичних виразів, що показують зв'язки між входами і виходами всіх тригерів, що складають лічильник. В процесі мінімізації виробляється до визначення функцій там, де це доцільно, одиницями в клітинках «×».

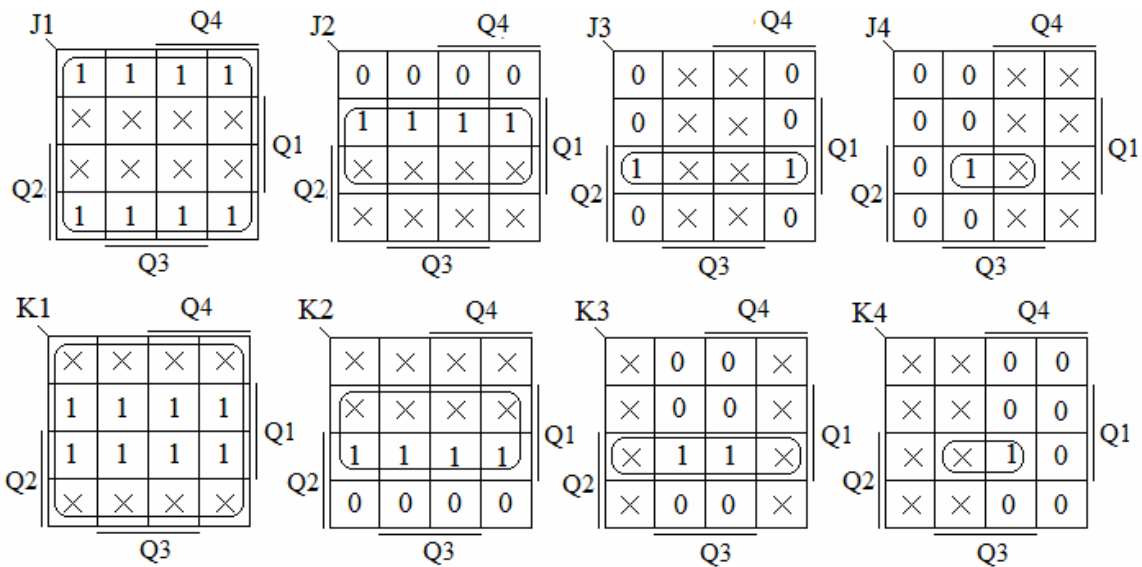


Рисунок 4.36 – Карти Карно для входів тригерів

У результаті, отримано наступні функції входів тригерів лічильника:

$$J1 = 1; K1 = 1; J2 = Q1; K2 = Q1; J3 = Q2 \cdot Q1; K3 = Q2 \cdot Q1;$$

$$J4 = Q3 \cdot Q2 \cdot Q1; K4 = Q3 \cdot Q2 \cdot Q1;$$

б) будується електрична схема лічильника, відповідно до реалізації функцій входів (рис. 4.37).

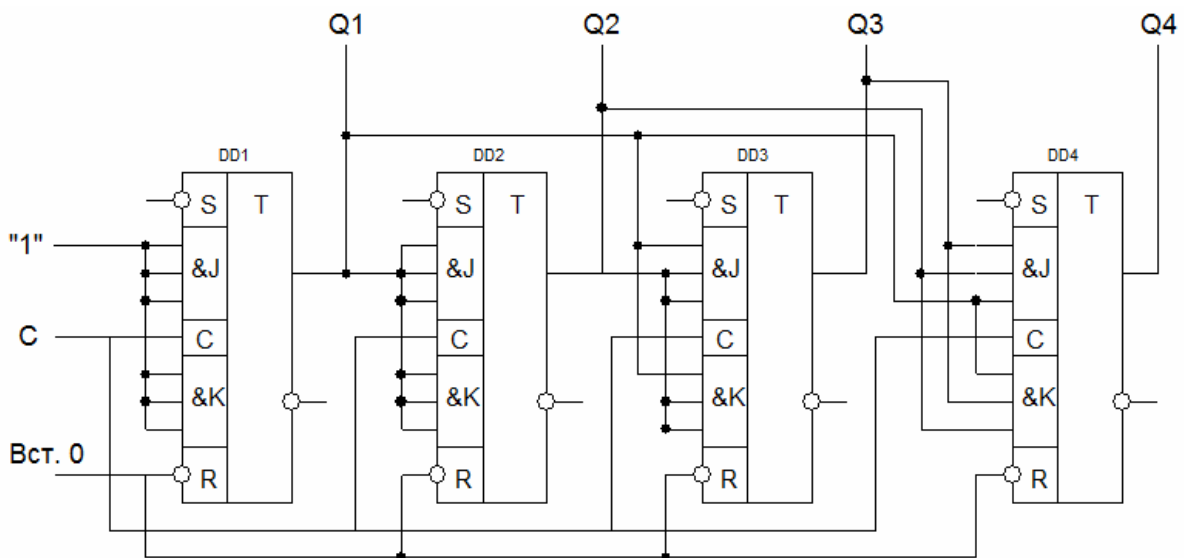


Рисунок 4.37 – Підсумовуючий двійковий лічильник з паралельним перенесенням

Віднімаючий лічильник. Як приклад синтезуємо двійковий паралельний лічильник з $K_{\text{рах.}} = 8$. Визначимо необхідну кількість розрядів m . В даному ви-

падку: $m = \log_2 8 = 3$. Лічильник працює відповідно до таблиці переходів (табл. 4.3), зворотній таблиці 4.1.

Таблиця 4.3 – Таблиця станів віднімаючого лічильника

x^n	$Q3^n$	$Q2^n$	$Q1^n$	$Q3^{n+1}$	$Q2^{n+1}$	$Q1^{n+1}$
1	0	0	0	1	1	1
1	0	0	1	0	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

Складемо карти Карно для функцій переходів тригерів кожного розряду (рис. 4.38).

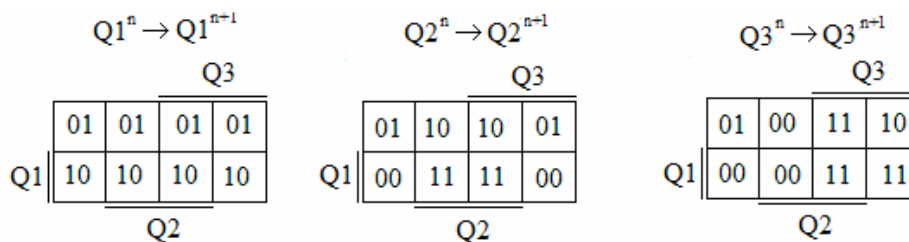


Рисунок 4.38 – Карти функцій переходів тригерів лічильника

Побудуємо схему на T – тригерах. Використовуючи словник переходів T – тригера (рис. 4.14), для кожного входу тригера складемо карти Карно (рис. 4.39).

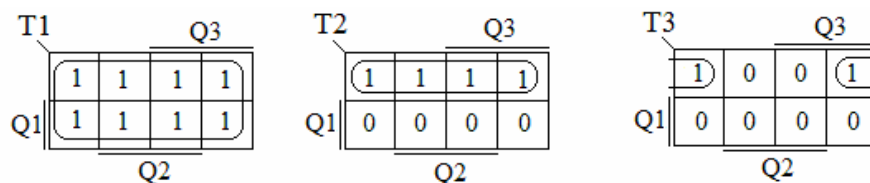


Рисунок 4.39 – Карти Карно функцій входів тригерів лічильників

Проведемо мінімізацію логічних функцій входів в картах Карно з метою отримання їх аналітичного представлення. У результаті, отримано наступні функції входів тригерів лічильника:

$$T1 = 1; \quad T2 = \overline{Q1}; \quad T3 = \overline{Q2} \cdot \overline{Q1};$$

Таким чином, віднімаючий лічильник відрізняється від підсумовуючого тим, що сигнали на входи Т подальших тригерів необхідно подавати з інверсних виходів тригерів попередніх розрядів (рис. 4.40).

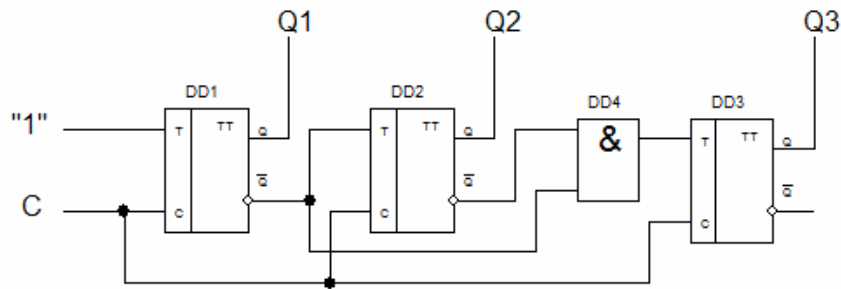


Рисунок 4.40 – Паралельний віднімаючий двійковий лічильник з $K_{\text{рах.}} = 8$

Реверсивний лічильник. Такий лічильник повинен, залежно від сигналів управління, забезпечувати або режим підсумовування, або режим віднімання вхідних сигналів. З порівняння функцій входів підсумовуючого і віднімаючого паралельних лічильників випливає, що самі функції мають один і той же вигляд. Лише в разі віднімаючого лічильника беруться інверсні значення змінних. Отже, реверсивний лічильник повинен містити схему управління, що забезпечує підключення або прямих, або інверсних виходів до входів подальших розрядів, залежно від сигналів управління напрямом рахунку М.

Функція входів для реверсивного лічильника на Т – тригерах (рис. 4.41) матиме вигляд:

$$T1 = 1; \quad T2 = M \cdot Q1 + \bar{M} \cdot \bar{Q1}; \quad T3 = (M \cdot Q1 \cdot Q2) + (\bar{M} \cdot \bar{Q2} \cdot \bar{Q1});$$

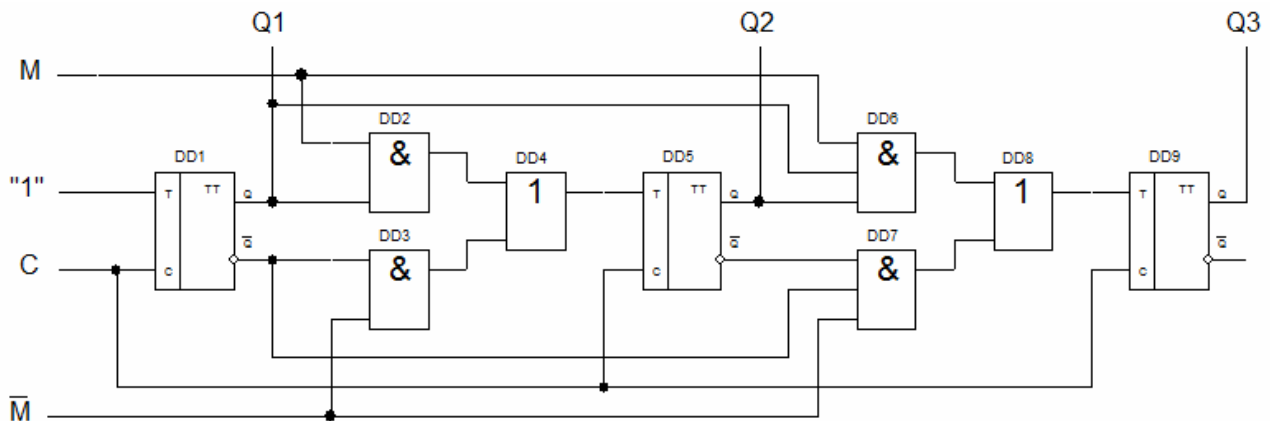


Рисунок 4.41 – Реверсивний паралельний двійковий лічильник з $K_{\text{рах.}} = 8$

4.4.3 Недвійкові лічильники з паралельним перенесенням

У двійкових лічильниках коефіцієнт рахунку $K_{\text{рах.}} = 2^n$ і може дорівнювати 2, 4, 8, 16, 32 і так далі. На практиці потрібні лічильники з коефіцієнтом рахунку не рівним 2^n , наприклад, 3, 6, 10, 12, 24 та ін. Вони виконуються на основі двійкових лічильників шляхом виключення в лічильники з $K_{\text{рах.}} = 2^n$ відповідного числа «надлишкових» станів N:

$$N = 2^n - K_{\text{рах.}}$$

Стани $(2^n - K_{\text{рах.}})$ виключаються усередині лічильника за допомогою зворотних зв'язків. Завдання синтезу таких лічильників зводиться до визначення вигляду необхідних зворотних зв'язків і мінімізації їх числа. Розглянемо приклад синтезу підсумовуючого лічильника з $K_{\text{рах.}} = 3$. Визначаємо необхідну кількість тригерів: $m = \log_2 3 = 1,58$

Округлюємо m до двох. Знаходимо число надлишкових станів:

$$2^2 - 3 = 1$$

З числа можливих станів лічильника виключимо, наприклад, стан $Q1 = Q2 = 1$.

Вибираємо тип тригерів (D – тригер). Для D – тригера словник переходів приведено на рисунку 4.9. Будуємо таблицю станів лічильника (табл. 4.4).

Таблиця 4.4 – Таблиця станів паралельного недвійкового лічильника

x^n	$Q2^n$	$Q1^n$	$Q2^{n+1}$	$Q1^{n+1}$
1	0	0	0	1
1	0	1	1	0
1	1	0	0	0
1	1	1	0	1

Складаємо карти функцій переходів тригерів лічильника (рис. 4.42).

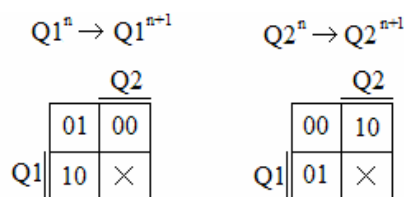


Рисунок 4.42 – Карти функцій переходів тригерів лічильника

Використовуючи словник переходів D – тригера і побудовані карти переходів тригерів лічильника, будемо карти функцій входів тригерів (рис. 4.43).

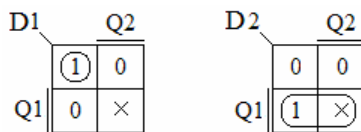


Рисунок 4.43 – Карти Карно функцій входів тригерів лічильника

В результаті отримані наступні функції входів тригерів лічильника:

$$D_1 = \bar{Q}_2 \bar{Q}_1; \quad D_2 = Q_1.$$

Будемо схему лічильника (рис. 4.44).

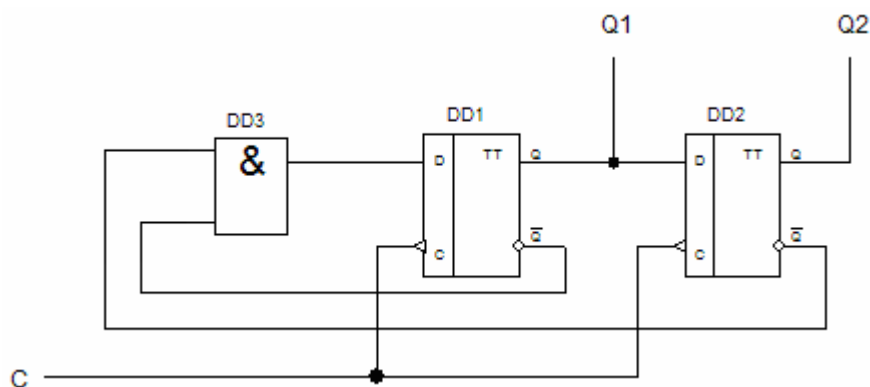


Рисунок 4.44 – Паралельний недвійковий лічильник з $K_{\text{рах.}} = 3$ на D – тригерах

Як видно зі схеми, виключення із станів лічильника двійкового числа 11 досягається подачею сигналів з інверсних виходів першого і другого розрядів на вхід першого розряду. Діаграма функціонування лічильника представлена на рисунку 4.45.

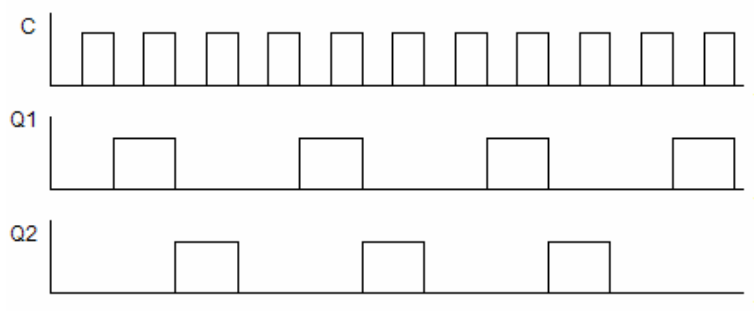


Рисунок 4.45 – Діаграма функціонування паралельного недвійкового лічильника з $K_{\text{рах.}} = 3$ на D – тригерах

4.4.4 Двійково-десяткові лічильники

Двійково-десяткові лічильники мають $K_{\text{рах.}} = 10$. Їх синтезують на основі чотирьохрозрядного лічильника, виключаючи $N = 2^m - K_{\text{рах.}} = 16 - 10 = 6$ надлишкових станів. Виключити можна будь-які 6 з 16 станів. У різних варіантах схем одному і тому ж десятковому числу можуть відповідати різні кодові комбінації, тобто різні варіанти лічильників працюють в різних двійково-десяткових кодах. Синтез двійково-десяткового лічильника можливо здійснити відповідно до таблиці станів (табл. 4.5).

Таблиця 4.5 – Таблиця станів двійково-десяткового лічильника

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	0
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	1	0	0
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	1	0
6	0	1	1	0	0	1	1	1
7	0	1	1	1	1	0	0	0
8	1	0	0	0	1	0	0	1
9	1	0	0	1	0	0	0	0

Складемо карти Карно для функцій переходів тригерів кожного розряду (рис. 4.46).

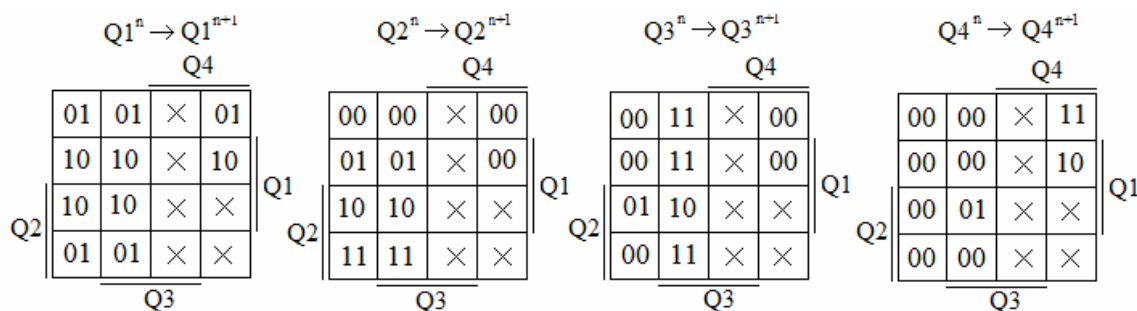


Рисунок 4.46 - Карти функцій переходів тригерів двійково-десяткового лічильника

Вибираємо JK – тригер, для побудови лічильника. Використовуючи слов-

ник переходів JK – тригера (рис. 4.18), для кожного входу тригера складаються карти Карно (рис. 4.47).

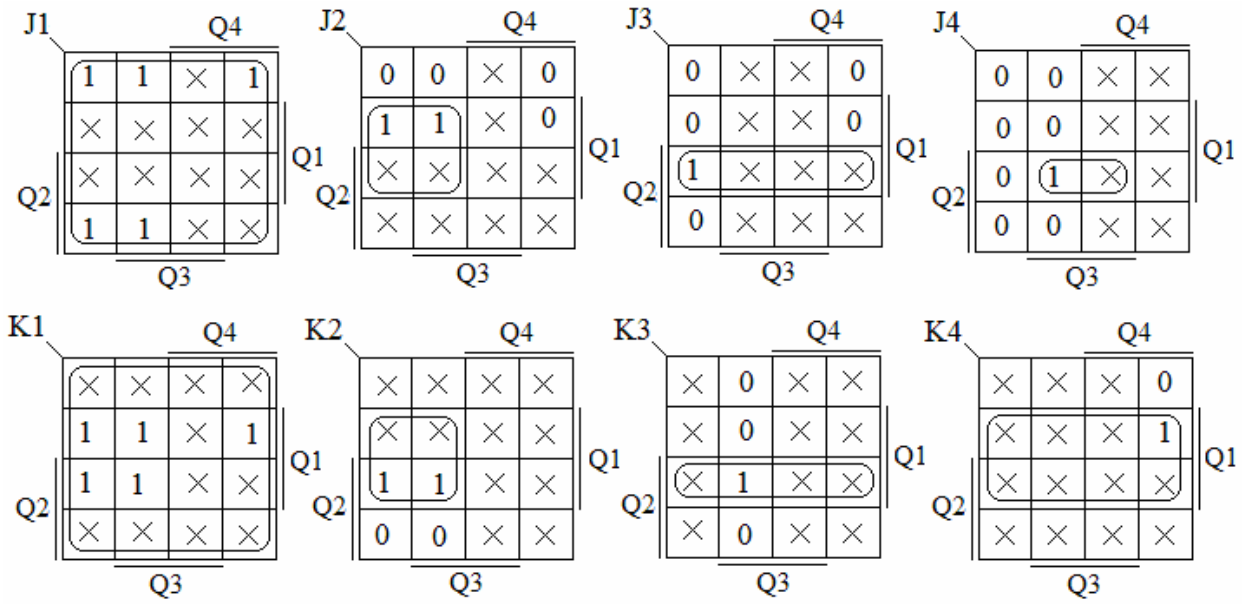


Рисунок 4.47- Карти Карно функцій входів тригерів лічильника

Функції входів лічильника мають вигляд:

$$J1 = 1; \quad J2 = \bar{Q}4Q1; \quad J3 = Q2Q1; \quad J4 = Q3Q2Q1$$

$$K1 = 1; \quad K2 = \bar{Q}4Q1; \quad K3 = Q2Q1; \quad K4 = Q1$$

Схема двійково-десятькового лічильника з $K_{\text{рах.}} = 10$ на JK – тригерах представлена на рисунку 4.48.

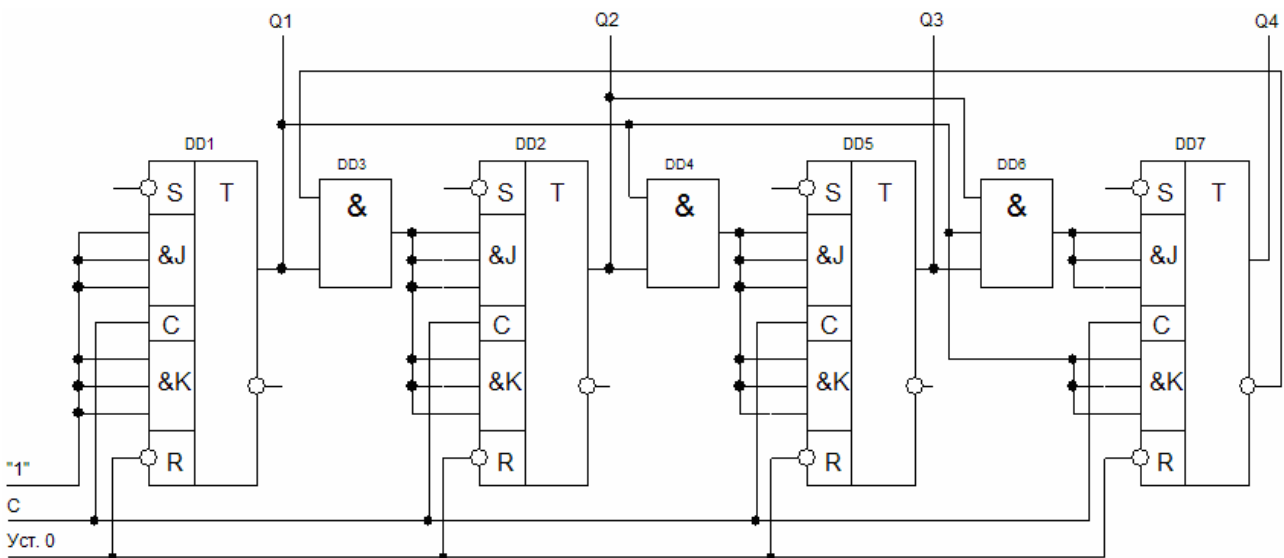


Рисунок 4.48 - Схема двійково-десятькового лічильника

4.4.5 Лічильники, що працюють у двійковому коді з заданою вагою розрядів

Нехай потрібно синтезувати синхронний лічильник, що працює в коді 5 – 2 – 1 – 1. Відповідно до заданого коду заповнюємо ліву частину таблиці 4.6 функціонування лічильника (стовпці Q_4, Q_3, Q_2, Q_1). 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. 4.49).

Таблиця 4.6 – Таблиця функціонування лічильника, що працює в коді 5 – 2 – 1 – 1

n	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}
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

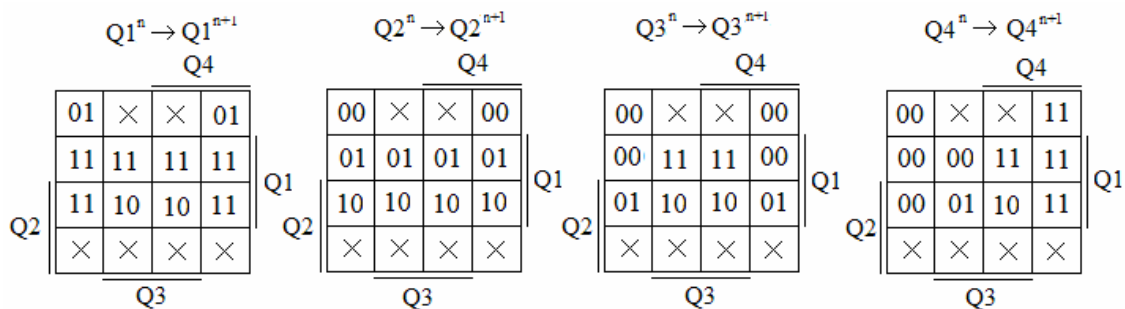


Рисунок 4.49 – Карти функцій переходів тригерів лічильника

Використовуючи словник переходів JK – тригера (рис. 4.18), для кожного входу тригера складемо карти Карно (рис. 4.50). Проведемо мінімізацію логічних функцій входів в картах Карно з метою отримання їх аналітичних виразів.

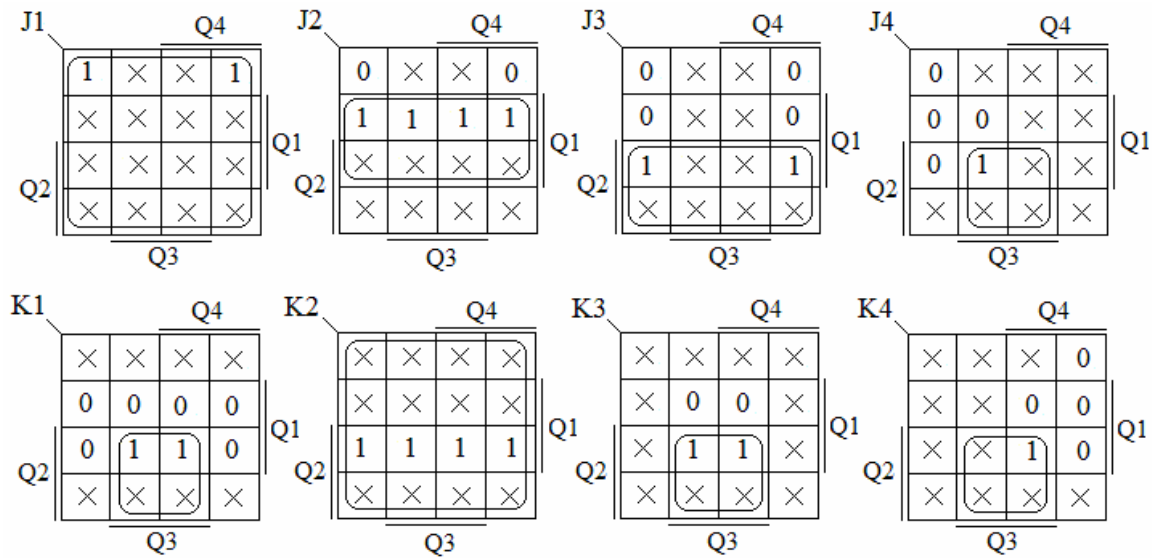


Рисунок 4.50 – Карти Карно функцій входів тригерів лічильника

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 4.51.

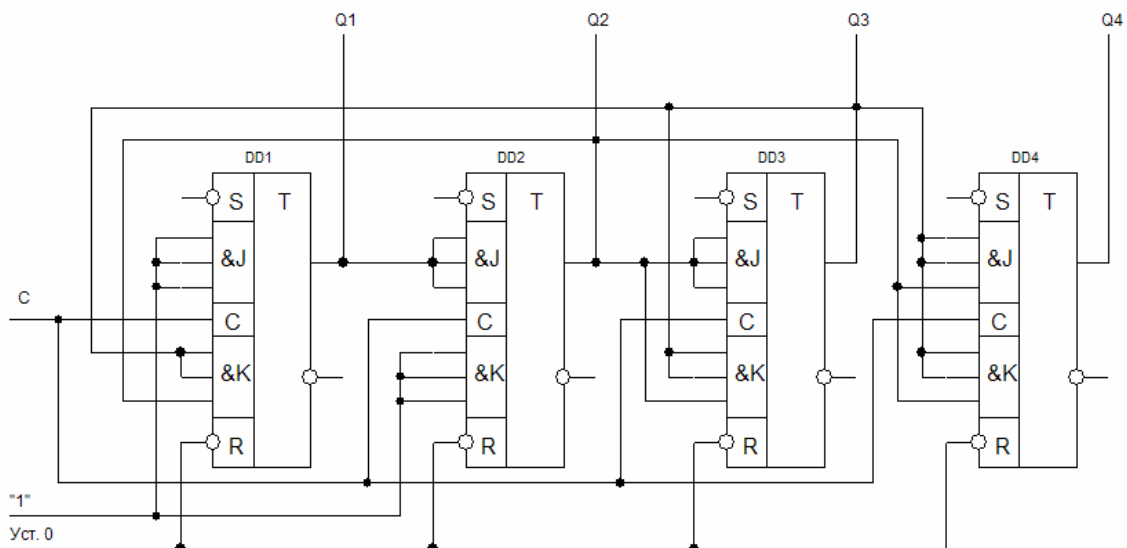


Рисунок 4.51 – Синхронний лічильник, який працює в коді 5 – 2 – 1 – 1

4.4.6 Кільцеві лічильники

Поширеним різновидом паралельних лічильників є кільцеві лічильники, виконані на базі регістрів зсуву. Найпростіша схема кільцевого лічильника отримується при з'єднанні виходу останнього тригера з входом першого (рис. 4.52).

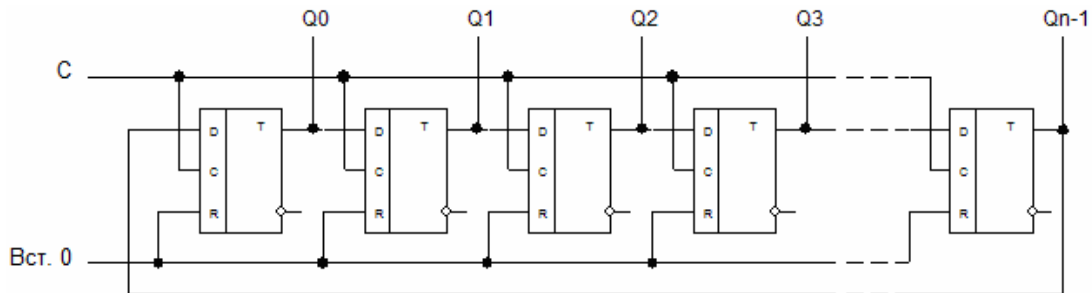


Рисунок 4.52 – Кільцевий лічильник з регістром зсуву

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 C , які надійшли, перезаписує 1 в наступний тригер. Число імпульсів, які надійшли, 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 рахунок почнеться спочатку (табл. 4.7). Таким чином, можна побудувати кільцевий лічильник з довільним коефіцієнтом рахунку.

Таблиця 4.7 – Таблиця функціонування кільцевого лічильника

n	Q1	Q2	Q3	Q4
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1

Модуль рахунку такого кільцевого лічильника має те саме значення, що і регістр зсуву, тобто $K_{рах.} = n$. Для збільшення модуля рахунку можна або збільшувати кількість тригерів в кільці, або включати лічильники послідовно. Так, наприклад, лічильник на 10 імпульсів ($K_{рах.} = 10$) можна реалізувати послідов-

ним з'єднанням одного рахункового тригера і кільцевого лічильника з п'яти тригерів.

Основним недоліком кільцевих лічильників є їх низька заводозахищеність. Наприклад, якщо під дією завади зникне записана в лічильник одиниця, то всі тригери виявляться в нульовому стані і лічильник працювати не зможе. Для усунення подібних збоїв використовуються різні способи корекції стану лічильника. Схема лічильника з автоматичною корекцією стану приведена на рисунку 4.53. У цій схемі, незалежно від того, в якому стані після включення виявляться тригери, після чотирьох тактових імпульсів на вході С встановиться необхідний вихідний стан «1000».

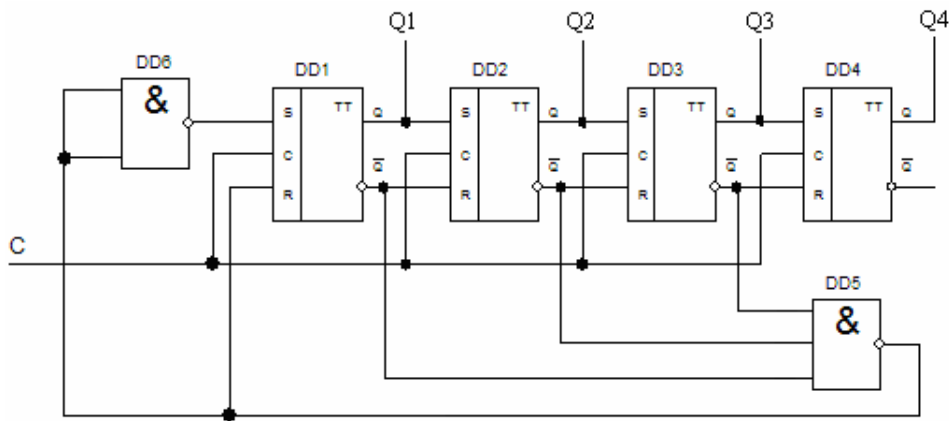


Рисунок 4.53 – Лічильник з автоматичною корекцією стану

Недоліком схем кільцевих лічильників є велике число тригерів, тому більш раціональною є схема з поділом частоти. Наприклад, мікросхема К155ИЕ5 (рис. 4.54) містить рахунковий тригер (вхід С1), дільник на вісім (вхід С2), утворений сполученими послідовно тригерами.

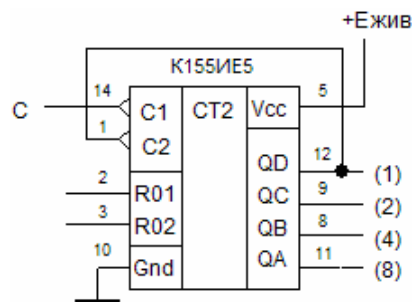


Рисунок 4.54 – Мікросхема К155ИЕ5

Тригери спрацьовують по зрізу вхідного імпульсу (переходу з 1 в 0). Якщо з'єднати послідовно усі 4 тригери, як показано на рисунку, вийде лічильник по модулю $2^n = 16$. Максимальне число, що зберігається в лічильнику при повному заповненні його одиницями, дорівнює: $N = 2^n - 1 = 15 (A_{10}) = 1111 (A_2)$. Такий лічильник працює з коефіцієнтом рахунку $K_{рах.}$, кратним цілому ступеню числа 2 і в ньому здійснюється циклічний перебір $K_{рах.} = 2^n$ стійких станів. Лічильник має виводи примусової установки в 0.

Часто необхідно мати лічильники з числом станів відмінним від 2^n . Наприклад, лічильник з $K_{рах.} = 10$ на мікросхемі К155ИЕ5 (рис. 4.55).

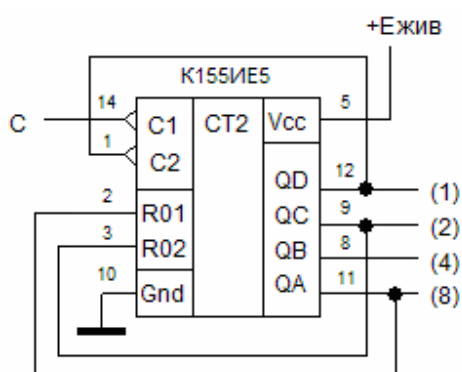


Рисунок 4.55 – Лічильник по модулю 10

4.4.7 Лічильники Джонсона

Різновидом кільцевих лічильників є лічильники Джонсона. У цих лічильниках вхід регістра сполучений не з прямим виходом Q, а з інверсним виходом \bar{Q} . В результаті, коли на вхід лічильника поступають тактові імпульси, то спочатку всі розряди лічильника заповнюються одиницями, а потім – нулями. Лічильник Джонсона має коефіцієнт перерахунку, удвічі більший числа складових його тригерів. Зокрема, якщо лічильник складається з трьох тригерів ($m = 3$), то він матиме шість стійких станів. Лічильники Джонсона використовуються в системах автоматики, наприклад, як розподільники імпульсів. Схема чотирьохрозрядного лічильника Джонсона приведена на рисунку 4.56, а стан його виходів приведений в таблиці 4.8.

Таблиця 4.8 – Стан виходів чотирьохрозрядного лічильника Джонсона

n	Q1	Q2	Q3	Q4	n	Q1	Q2	Q3	Q4
0	1	0	0	0	4	0	1	1	1
1	1	1	0	0	5	0	0	1	1
2	1	1	1	0	6	0	0	0	1
3	1	1	1	1	7	0	0	0	0

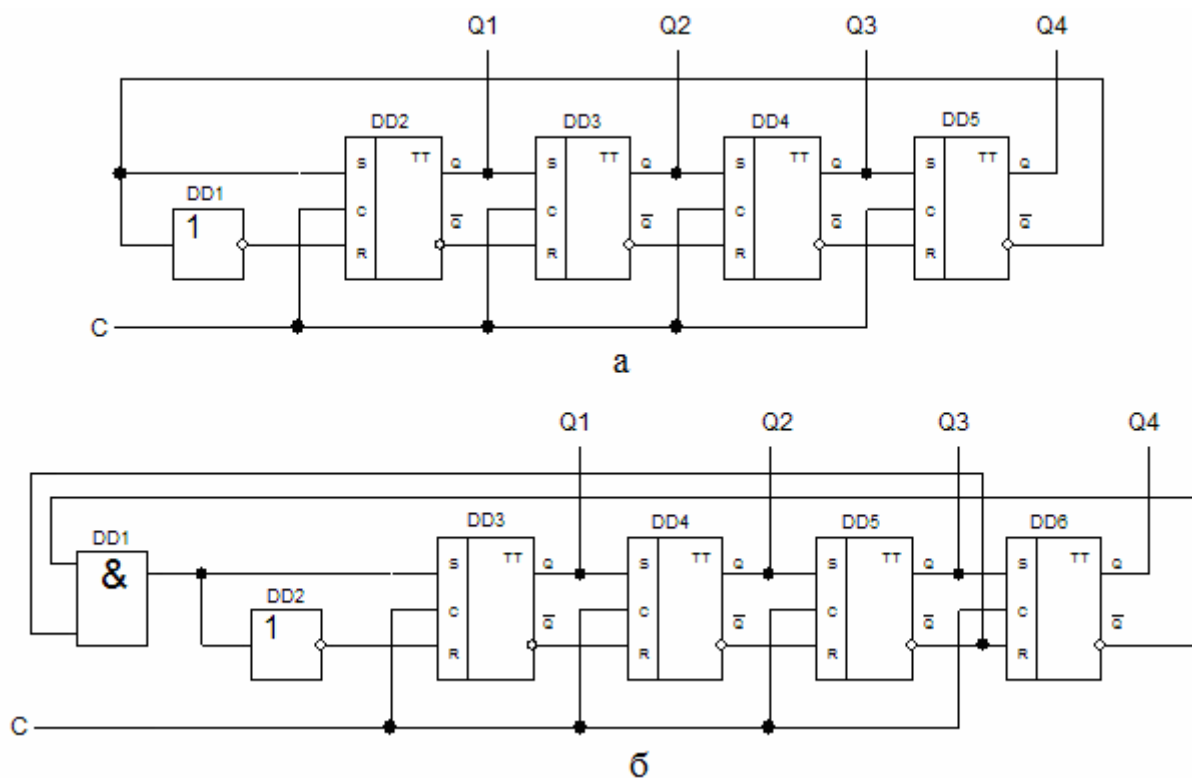


Рисунок 4.56 - Схема чотирьохрозрядного лічильника Джонсона з парним (а) і непарним (б) модулем рахунку

Як випливає з таблиці 4.8, модуль рахунку лічильника Джонсона в два рази більше модуля рахунку простого кільцевого лічильника. У лічильнику Джонсона, як і в інших кільцевих лічильниках, можуть бути збої, викликані завадами. Для корекції порушень, викликаних збоями, також використовуються способи, за допомогою яких виконується перехід з будь-якої забороненої комбінації в одну із дозволених.

Лічильники Джонсона широко використовуються в дільниках частоти імпульсів, генераторах випадкових чисел, в пристроях пам'яті та ін. На базі лічильника Джонсона можна легко реалізувати лічильники з будь-яким парним мо-

дулем рахунку. Якщо є необхідність мати непарне значення модуля рахунку, то можна на вхід першого розряду подавати сигнал $\bar{Q}_{n-1}\bar{Q}_n$, замість сигналу \bar{Q}_n , як це показано на рисунку 4.56 (б). При цьому з набору вихідних станів лічильника Джонсона виключається одна кодова комбінація, складена з нулів.

4.5 Генератори числових послідовностей

Послідовнісні пристрої - генератори чисел 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. Будь-який лічильник можна вважати генератором послідовності чисел, яка має $L_n = K_{\text{рах.}}$. 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 $K_{\text{рах.}} = L_n$ і підключеною до його виходів комбінаційною схемою, яка синтезує необхідні значення двійкових чисел послідовності. 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 (табл. 4.9).

Таблиця 4.9 – 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

Складемо карти Карно для функцій переходів тригерів кожного розряду (рис. 4.57).

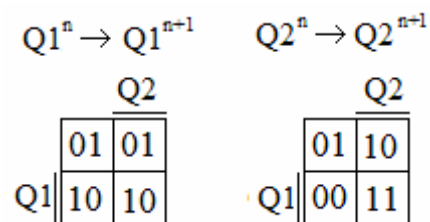


Рисунок 4.57 – Карти функцій переходів тригерів генератора

Використовуючи словник переходів JK – тригера 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. 4.58).

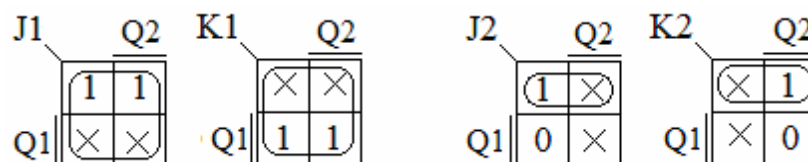


Рисунок 4.58 – Карти Карно функцій входів тригерів лічильника

Виконаємо мінімізацію логічних функцій входів в картах Карно з метою отримання 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 (рис. 4.59).

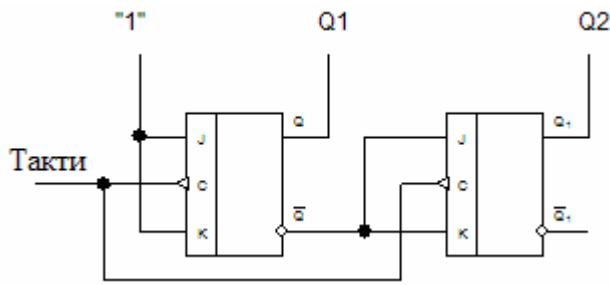


Рисунок 4.59 – Лічильник генератора чисел

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 (табл. 4.10).

Таблиця 4.10 – 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:

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

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

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

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

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

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

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

$$F3 = Q2;$$

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

Схема генератора числової послідовності представлена на рисунку 4.60.

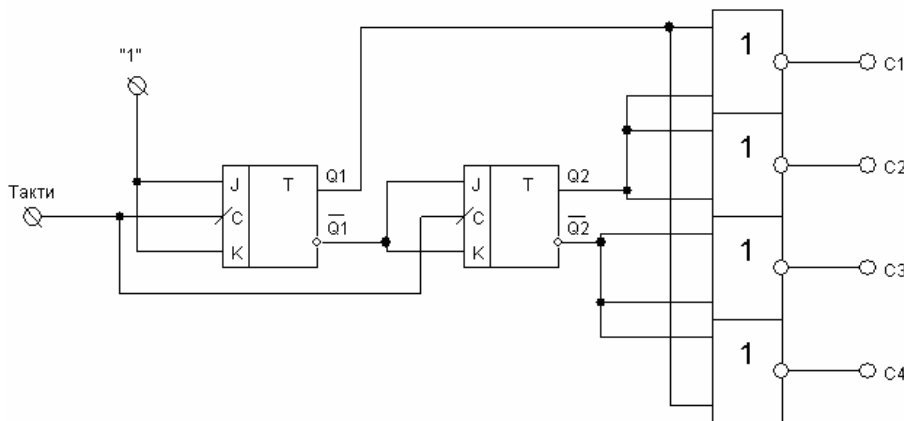


Рисунок 4.60 – Генератор числової послідовності 4 – 3 – 2 – 12 на основі лічильника з вихідною комбінаційною схемою

Часова діаграма роботи генератора (рис. 4.61) 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 (рис. 4.25) на вхід X подається періодична послідовність інформаційних «0» і «1».

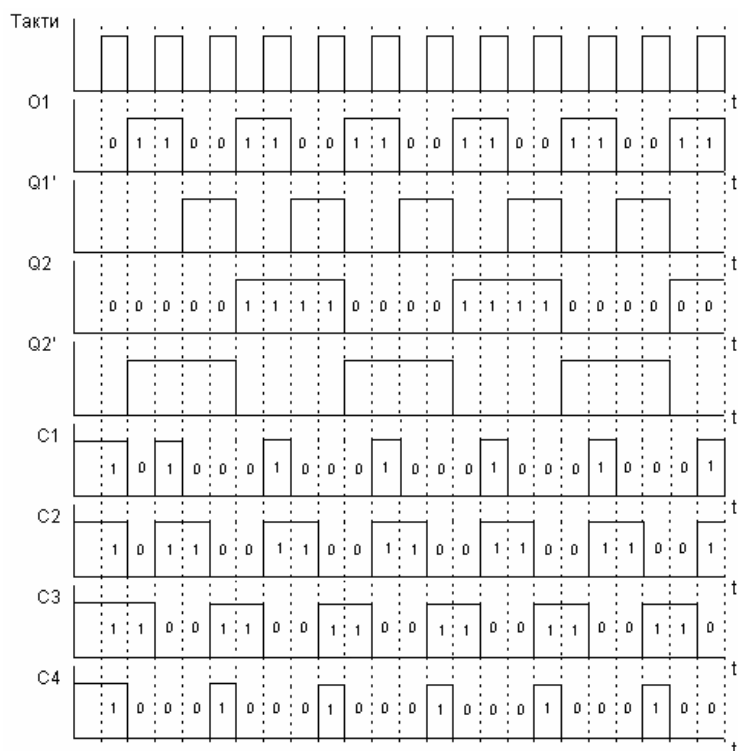


Рисунок 4.61 – Часова діаграма роботи генератора

Синтез структури генераторів послідовностей на регістрах зсуву в першу чергу полягає в знаходженні вигляду функцій входів. Відповідно до схеми регістра зсуву (рис. 4.25), на вхід x подається періодична послідовність інформаційних «0» і «1».

Наприклад. Синтезуємо генератор числової послідовності з семи цифр. Для цього буде потрібно трьохрозрядний регістр зсуву. На інформаційний вхід подається послідовність 0101100.0101100. Інформація з'явиться на всіх виходах регістра в паралельному коді після приходу третього тактового імпульсу. Далі, кожен тактовий імпульс зміщуватиме цю інформацію, і комбінації значень сигналів $Q_3Q_2Q_1$ періодично повторюватимуться відповідно до заданої послідовності (рис. 4.62). Таким чином, реалізується числова послідовність:

$$2 - 5 - 3 - 6 - 4 - 0 - 1$$

1. Кодування внутрішніх станів генератора числової послідовності наведено в таблиці переходів генератора (табл. 4.11).

2. Складаються карти Карно для функцій переходів тригерів кожного розряду (рис. 4.63).

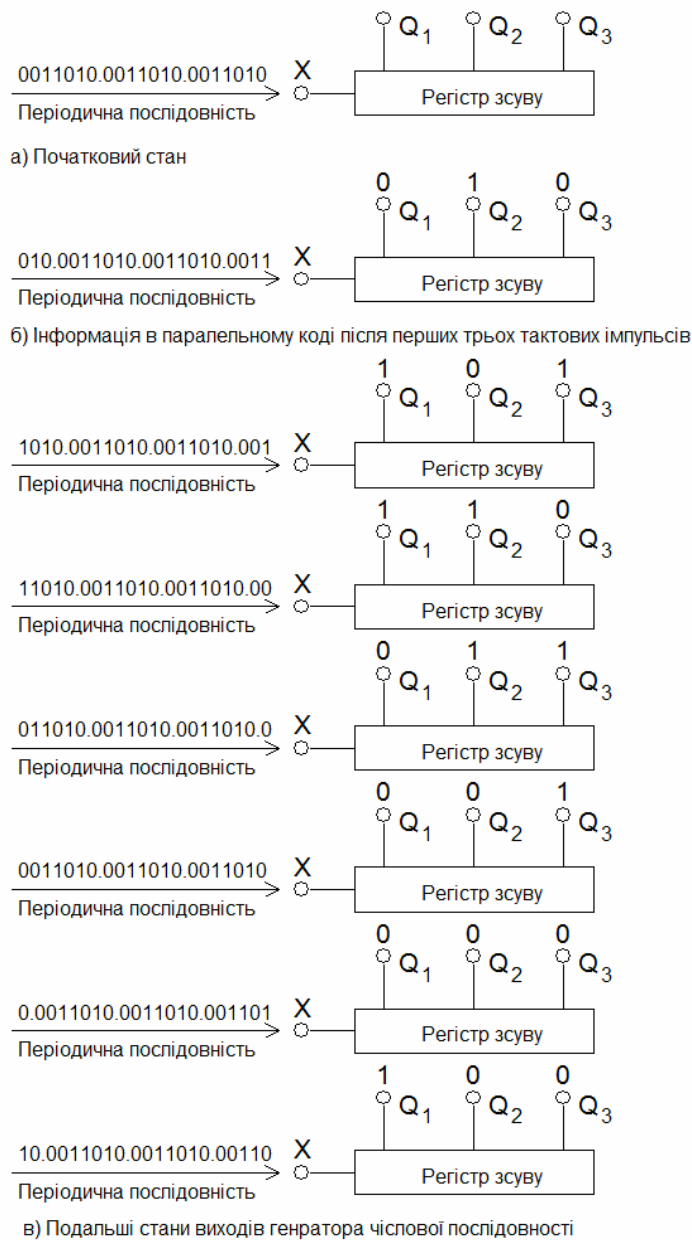


Рисунок 4.62 – Алгоритм функціонування генератора числової послідовності

Таблиця 4.11 – Таблиця станів генератора числової послідовності

x^n	Q_3^n	Q_2^n	Q_1^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}
2	0	1	0	1	0	1
5	1	0	1	0	1	1
3	0	1	1	1	1	0
6	1	1	0	1	0	0
4	1	0	0	0	0	0
0	0	0	0	0	0	1
1	0	0	1	0	1	0

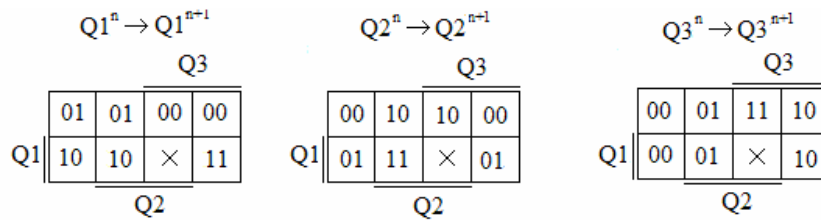


Рисунок 4.63 – Карти функцій переходів тригерів генератора

Зазвичай, в регістрах зсуву використовуються D – тригери.

3. Використовуючи словник переходів D – тригера, для кожного входу тригера складаються карти Карно, в клітинках яких проставляються сигнали, необхідні для забезпечення переходів тригерів, вказаних в однойменних клітинках карт функцій переходів (рис. 4.64).

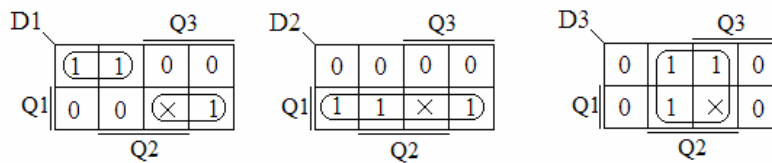


Рисунок 4.64 – Карти Карно функцій входів тригерів генератора

4. Проводиться мінімізація логічних функцій входів в картах Карно з метою отримання їх аналітичних виразів.

У результаті отримують наступні функції входів тригерів генератора:

$$D1 = \bar{Q}_3 \bar{Q}_1 + Q_3 Q_1; \quad D2 = Q_1; \quad D3 = Q_2.$$

5. Складання логічної схеми генератора виконується за рахунок побудови управляючої комбінаційної схеми, яка реалізує функцію входів, та її підключенням до входу першого розряду регістра зсуву (рис. 4.65).

Синтезувавши комбінаційну схему з перебудовуваною структурою, можна отримати ряд різних послідовностей від одного пристрою.

Генератори на основі регістрів утворюють лише циклічні послідовності чисел. Для реалізації будь-яких нециклічних послідовностей потрібне використання додаткових комбінаційних перетворювачів коду, які включаються на виході генератора. Порядок появи чисел в послідовності можна вважати випадковим, а повторення чисел відбувається через $2m - 1$ тактів. Тому такі схеми називаються генераторами псевдовипадкових послідовностей.

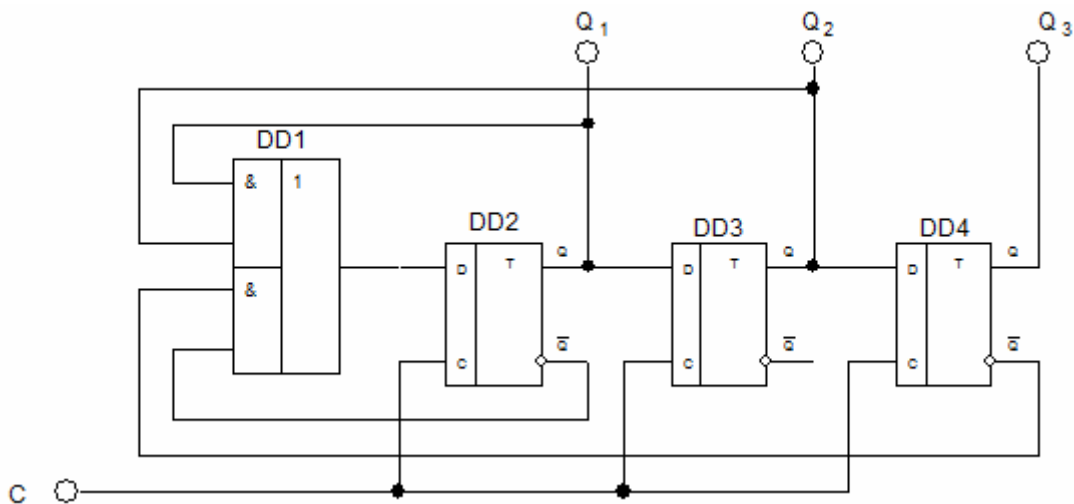


Рисунок 4.65 – Логічна схема генератора числової послідовності на основі регістра зсуву

При збільшенні m довжина псевдовипадкових послідовностей швидко зростає, тому генератори, що мають $m > 10$, широко використовуються в цифровій апаратурі для формування безлічі тестових сигналів, необхідних для контролю і діагностики несправностей.

Контрольні запитання до четвертого розділу

1. Обґрунтувати рівняння функціонування RS – тригера, RST – тригера, T – тригера, JK – тригера, D – тригера.
2. Побудувати паралельний регістр на тактових D – тригерах.
3. Побудувати послідовний регістр на тактових D – тригерах.
4. Побудувати двійковий лічильник з послідовним перенесенням.
5. Побудувати двійковий лічильник з паралельним перенесенням.
6. Побудувати недвійковий лічильник з паралельним перенесенням.
7. Побудувати двійково-десятковий лічильник.
8. Пояснити принцип функціонування кільцевих лічильників.
9. Пояснити принцип функціонування лічильників Джонсона.
10. Побудувати генератор заданої числової послідовності.

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

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

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

3) постійні запам'ятовуючі пристрої (ПЗП) – здійснюють лише зчитування інформації.

4) перепрограмовувані постійні запам'ятовуючі пристрої (ППЗП) – здійснюють лише зчитування інформації постійно і запис рідко.

Найважливіші параметри запам'ятовуючих 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'ятовуючим елементом (ЗЕ), а слово – запам'ятовуюча комірка (ЗК), 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 запам'ятовуючої комірки в стан, що 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 зчитуван-

ня, оскільки після цих операцій потрібен час для відновлення початкового стану ЗП.

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

5.1 Оперативні запам'ятовуючі пристрої

При створенні ОЗП малої ємності (наприклад НОЗП), застосовується організація пам'яті у вигляді окремих регістрів. Типовий приклад тригерного ОЗП – регістр з паралельним прийомом і паралельною видачею інформації. На рисунку 5.1 приведена схема 4-розрядного паралельного регістра, побудованого на RS – тригерах DD5 ... DD8.

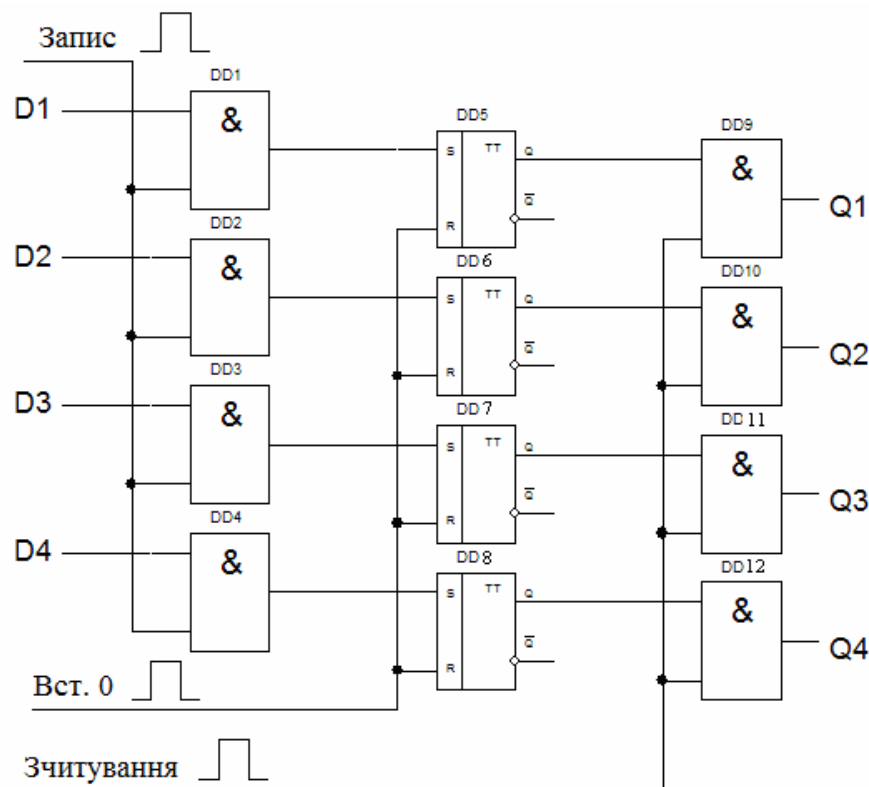


Рисунок 5.1 – Функціональна схема паралельного регістра на RS – тригерах

Елементи DD1... DD4 утворюють коло управління записом, а елементи DD9 ... DD12 – коло управління зчитуванням. Перед записом інформації всі тригери регістра встановлюють в стан «0» шляхом подачі імпульсу «1» на їх

R – входи. Записувана інформація подається на входи DD1 ... DD4. Для запису інформації подається імпульс «Запис», що відкриває вхідні елементи «I». Код вхідного числа записується в реєстр. Після закінчення імпульсу «Запис» елементи DD1 ... DD4 закриваються, а інформація, записана в реєстр, зберігається, не дивлячись на те, що вхідна інформація може змінюватися. Для зчитування інформації подають сигнал «1» на вхід «Зчитування». По цьому сигналу на вихідні шини реєстра на час дії сигналу передається код числа, записаний в реєстр. По закінченню операції зчитування вихідні ключі закриваються, а інформація, записана в реєстр, зберігається. Тобто, можливе багатократне зчитування інформації.

При збільшенні ємності ОЗП виникає проблема доступу до кожного елемента пам'яті при обмеженому числі виводів корпусу. Це завдання вирішується за допомогою адресної організації ЗП з використанням дешифратора коду адреси. Тому 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, реєстри, різного роду ключі, комутатори і інші схеми загального призначення.

Накопичувач складається з запам'ятовуючих комірок, кожна з яких зберігає один біт інформації. Основу ЗК складають бістабільні елементи, основною властивістю яких є наявність двох стійких станів $Q = 0$ або $Q = 1$. ЗК розташовані у вузлах ґрат, утворених адресними шинами. 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 ЗК. 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 всіх ЗК. Біполярні ЗК володіють значною швидкістю, а МДП – споживають значно менше енергії.

У накопичувачах використовуються будь-які ЗК.

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

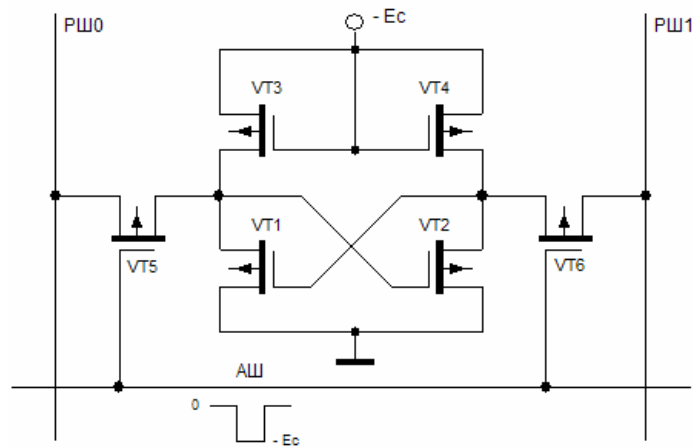


Рисунок 5.2 – Статичний елемент пам'яті на однотипних МОН транзисторах

Ці ключі нормально замкнуті і ЗК відключена від розрядних шин. Під час надходження негативного імпульсу « $-E_c$ » на адресну шину, ключі VT5, VT6 відмикаються і підключають комірку до розрядних шин. При цьому на розрядні шини поступають рівні Q і \bar{Q} , записані у комірку. У режимі запису на адресну шину також подається імпульс « $-E_c$ », а на розрядні шини подаються необхідні (взаємно протилежні) рівні, які наводять комірку у відповідний стан. Таким чином імпульс на адресній шині в обох режимах грає роль тактового імпульсу.

В комірці динамічного типу (рис. 5.3) зберігання біта інформації здійснюється ємностями $C1$, $C2$ (звичайно це паразитні ємності МОН транзисторів). Методика запису і зчитування та ж, що і у комірці статичного типу. Нехай при запису на розрядні шини РШ1, РШ0 подані відповідні рівні « $-E_c$ » і «0». Рівень « $-E_c$ » через ключ VT4 поступає на затвор транзистора VT1 і цей транзистор буде відкритий. На затвор транзистора VT2 поступить рівень «0» і цей транзистор буде замкнутий. Напруга на ємностях матиме значення $U_{c1} = -E_c$, $U_{c2} = 0$. Якщо залишковий струм замкнутого транзистора VT2 досить малий, то ємність $C1$ розряджатиметься вельми повільно і значення напруг « $-E_c$ » і «0» зберігатимуться на виходах комірки (на стоках) тривалий час.

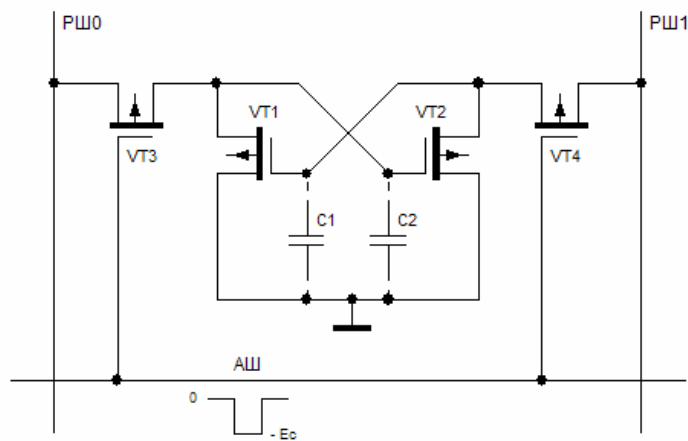


Рисунок 5.3 – Динамічний елемент пам'яті на однотипних МОН транзисторах

Динамічні ЗК більш економічні статичних, оскільки у них відсутнє джерело живлення. Отже, в режимі зберігання вони не споживають потужність.

МОН транзисторні комірки більш економічні і компактні біполярних (рис. 5.4), але поступаються їм по швидкодії.

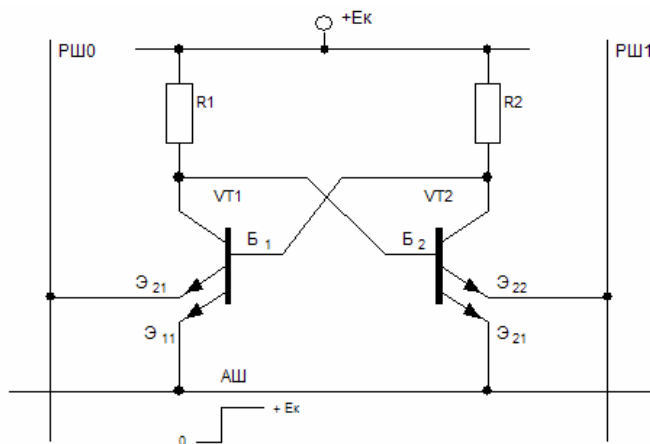


Рисунок 5.4 – Статична комірка пам'яті на біполярних транзисторах

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 емітерні переходи. При зчитуванні, коли на адресну шину подається позитивна напруга «+ Ек», емітер E_{11} закривається і струм транзистора

VT1 протікає в шину PШ0 через емітер E_{21} , який залишився під низьким потенціалом. Шина PШ1 залишається знеструмленою. Під час запису одночасно з адресним імпульсом подається напруга на ту розрядну шину, яка пов'язана з транзистором, що підлягає замиканню. Якщо подати «+ Ек» на розрядну шину PШ1, то транзистор VT2 залишиться замкнутим і стан ЗЯ не зміниться. Якщо ж подати «+ Ек» на шину PШ0, то виявляється, що обидва емітера транзистора VT1 будуть замкненими. Тоді струм потече через базу транзистора VT2 в емітер E_{22} , який знаходиться під низьким потенціалом шини PШ1. При цьому транзистор VT2 відкриється і стан ЗК зміниться на протилежний.

Розглянемо призначення і взаємодію складових частин ОЗП на 64 біта з адресною організацією вибірки 16 чотирьохрозрядних слів ($16 \text{ слів} \times 4 \text{ розряди} = 64 \text{ біт}$). Умовне зображення і функціональна схема такої мікросхеми приведені на рисунку 5.5. Масив пам'яті утворений 16 чотирьохрозрядними колами тригерів. При сигналі $V = 0$ одне з кіл, що відповідає виставленій адресі A1... A4, переходить в робочий стан і його сигнали поступають на входи елементів I (DD8 ... DD11). При сигналі $V = 1$ на всіх виходах дешифратора низькі рівні і всі тригери відключені від вихідних шин накопичувача. При $V = 0$ і $W = 0$ на вибране коло поступають інформаційні сигнали (входи D1... D4) і елементом DD1 виконується сигнал запису. У цьому режимі при зміні інформації на вході ОЗП відбувається перезапис інформації в даному слові масиву. При сигналах $V = 1$ і $W = 0$ вхідна інформація проходить безпосередньо на вихід мікросхеми, минувши масив тригерів (дешифратор не вибирає жодного з кіл). При $V = 1$ і $W = 1$ заборонена робота дешифратора, вузла, що генерує сигнал «Запис» і вхідних елементів I.

Таким чином, блок управління (десять елементів I) забезпечує роботу ОЗП в режимах: запис, зчитування, наскрізне перенесення, зберігання інформації. Вихідні логічні елементи I (DD8 ... DD11) виконані за відкритим колектором, що дозволяє сполучати разом виходи Q декількох мікросхем ОЗП. При цьому відбувається нарощування ємності ОЗП (дві мікросхеми - 32 слова, три -

48 і т. д.). Адресне управління A1 ... A4, інформаційні входи D1... D4 і виходи Q1 ... Q4 всіх мікросхем об'єднують в загальні шини, а вибір робочого масиву здійснюють додатковим дешифратором по входах V і W. Так виглядає структура мікросхеми K155PY2 (рис. 5.5 б).

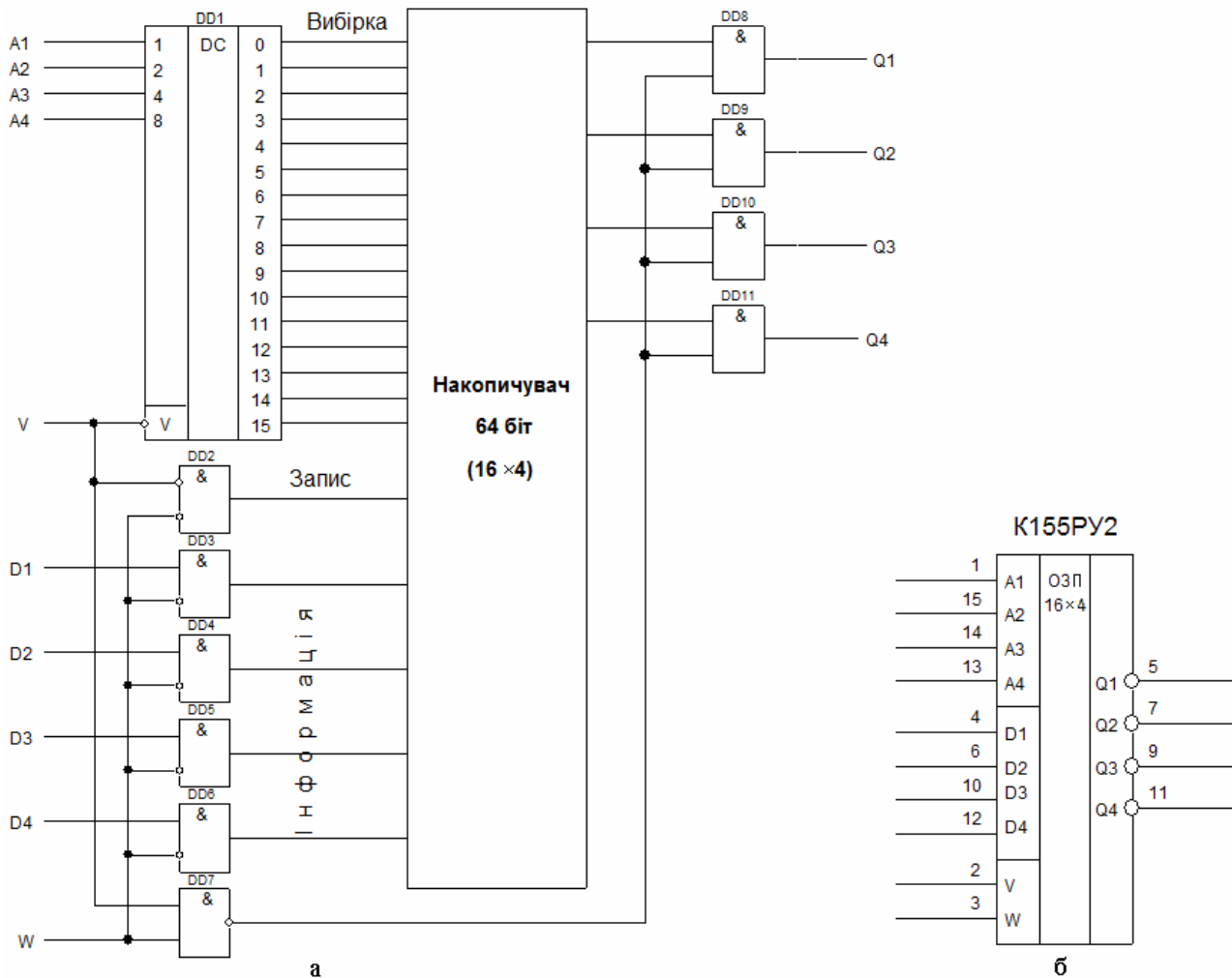


Рисунок 5.5 – Функціональна схема і умовне позначення ОЗП на 64 біт

ОЗП потребують для своєї побудови великої площі кристала, тому їх ємність відносно невелика. Статичні ОЗУ застосовуються для побудови мікроконтролерних схем із-за простоти побудови принципової схеми і можливості працювати на скільки завгодно низьких частотах, аж до постійного струму. Крім того, статичні ОЗП застосовуються для побудови КЕШ-пам'яті в універсальних комп'ютерах із-за високої швидкодії статичного ОЗП.

5.2 Постійні запам'ятовуючі пристрої

Запис інформації в ПЗП виконується або один раз і назавжди, або є спеціальною, рідко використовуваною операцією. Тому ПЗП допускають лише зчитування 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.

Спочатку елементи поєднання розташовані у всіх вузлах матриці і в такому однорідному вигляді матрицю поставляють замовникові (рис. 5.6). 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 і 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).

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.

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.

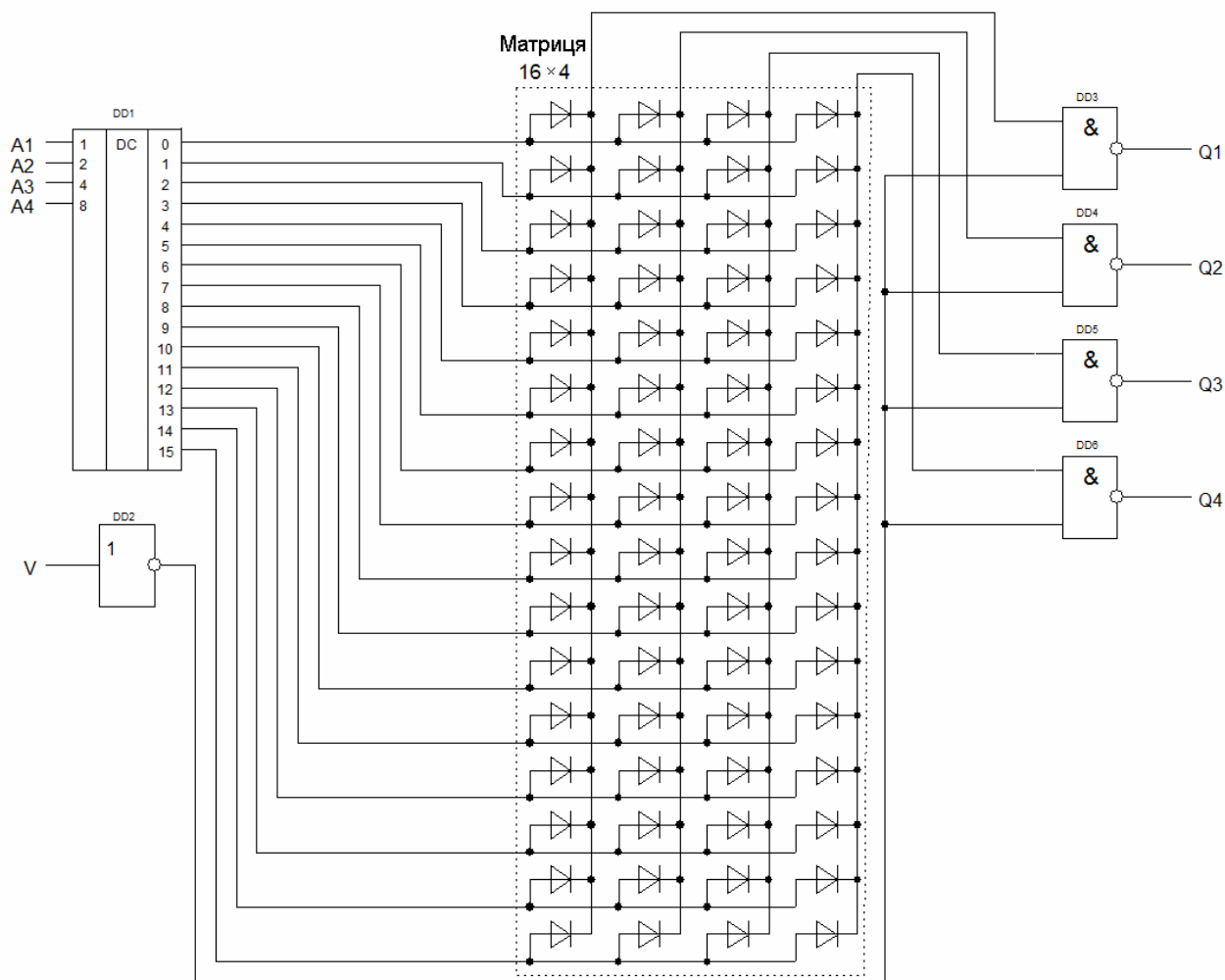


Рисунок 5.6 – Функціональна схема ПЗП

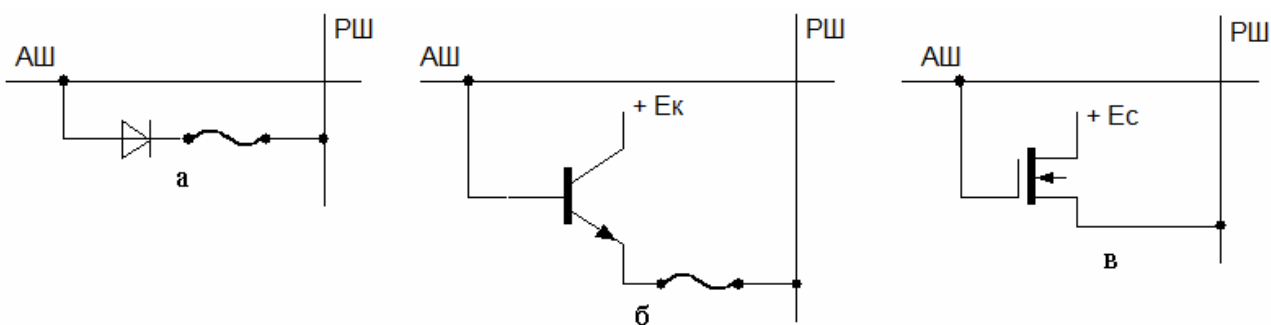


Рисунок 5.7 – Елементи пам'яті на діоді (а), біполярному транзисторі (б) і МДН-транзисторі (в)

5.3 Репрограмуємі постійні запам'ятовуючі пристрої

Репрограмуємі постійні запам'ятовуючі пристрої (РПЗП) дозволяють час від часу міняти записану інформацію і зберігати її невизначено довго при відключеному живленні.

Принцип функціонування РПЗП полягає в тому, аби зворотно міняти порогову напругу МОН-транзистора U_0 . Якщо зробити порогову напругу $|U_0| > U_{\text{АШ}}$, то транзистор не відкриватиметься адресними імпульсами (рис. 5.7 в). Останні транзистори, в яких $|U_0| < U_{\text{АШ}}$, функціонуватимуть нормально.

Способи управління пороговою напругою базуються на введенні додаткових зарядів в діелектрик.

Структури РПЗП-УФ використовують звичайні МОН-транзистори з одношаровим діелектриком в якому розташований плаваючий затвор (рис. 5.8 а). Прикладаючи до керуючого затвора чималу напругу викликають лавинний пробій в діелектрику, внаслідок чого у ньому накопичуються електрони. Відповідно змінюється порогова напруга. Заряд електронів зберігається протягом досить довгого часу. Стирання інформації здійснюється ультрафіолетовим опроміненням кристала, через спеціально виготовлені вікна. При цьому, електрони витісняються з плаваючого затвору у підкладку внаслідок посилення теплового руху за рахунок фотонного бомбардування від джерела ультрафіолетового випромінювання.

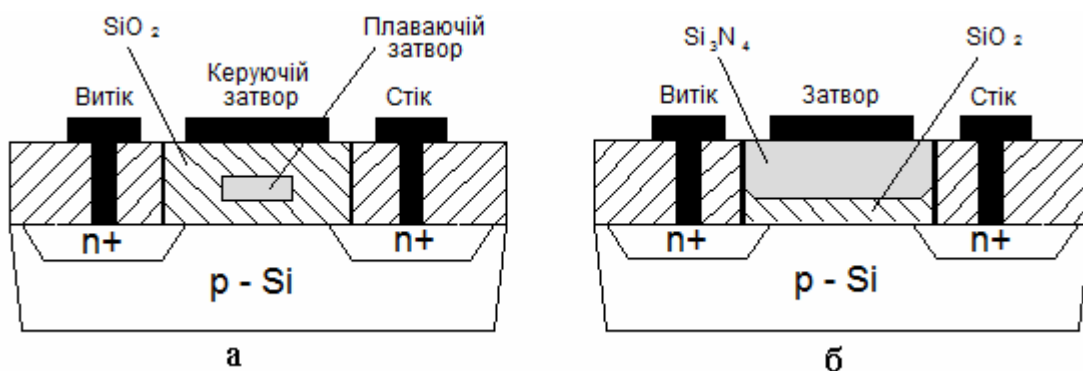


Рисунок 5.8 – Елементи пам'яті РПЗП: а – з плаваючим затвором і ультрафіолетовим затиранням; б – структури МНОН з електричним затиранням

У структурі РПЗП-ЕС використовують транзистори структури метал – нітрид – оксид – напівпровідник (МНОН) (рис. 5.8 б). В цих транзисторах введення і виведення зарядів здійснюється за допомогою коротких високовольтних імпульсів, що подаються на затвор. При записі інформації під дією сильного електричного поля між затвором і підкладкою електрони набувають достатньої енергії щоб пройти тонкий діелектричний шар на межі поділу двох діелектриків. Верхній шар Si_3N_4 має значну товщину, тому електрони подолати його не можуть. Накопичений на межі поділу двох діелектричних шарів заряд електронів знижує порогову напругу. При прикладенні імпульсу напруги протилежної полярності електрони витісняються в підкладку, що відповідає режиму стирання.

Flash-пам'ять (флеш- пам'ять) є представником класу репрограмуємих постійних ЗП з електричним затиранням. Проте затирання в ній здійснюється відразу у цілій області комірок: блоку або всієї мікросхеми. Це забезпечує швидший запис інформації (програмування ЗП). Для спрощення цієї операції в мікросхемі включаються спеціальні пристрої керування, що роблять запис інформації подібним до запису в звичайне ЗП, виконані на апаратному і програмному принципі управління.

Флеш- пам'ять будується на однострижисторних елементах пам'яті, що забезпечує щільність зберігання інформації навіть декілька вище, ніж в динамічній оперативній пам'яті. Найбільш часто використовують два типи флеш- пам'яті, в яких запам'ятовуючі транзистори підключені до розрядних шин паралельно або послідовно.

Контрольні запитання до четвертого розділу

1. Пояснити принцип функціонування ОЗП.
2. Пояснити функціонування статичних елементів пам'яті.
3. Пояснити функціонування динамічних елементів пам'яті.
4. Пояснити принцип функціонування ПЗП.

« СИНТЕЗ ФУНКЦІОНАЛЬНИХ ЦИФРОВИХ ПРИСТРОЇВ »

Для успішного виконання контрольної роботи студент повинен мати уявлення про основні функції і закони алгебри логіки, способи завдання і мінімізації функцій алгебри логіки, а також про методи синтезу комбінаційних дискретних пристроїв і кінцевих автоматів. Перш, ніж приступати до виконання контрольної роботи, студент повинен вивчити відповідні розділи підручника.

Мета контрольної роботи — закріпити знання, отримані студентом при самостійному вивченні дисципліни.

Контрольна робота містить завдання, що складається з двох задач. Записка пояснення повинна містити вихідні дані по варіанту, короткі пояснення до методики рішення кожної задачі з додатком необхідних креслень і таблиць.

Завдання на контрольну роботу**Задача №1**

Умови роботи комбінаційного пристрою, що має чотири входи (A, B, C, D) і один вихід F, задані таблицею істинності (табл. 6.1), де індекс при F відповідає номеру варіанта, який визнається останньою цифрою шифру студента.

Потрібно синтезувати функціональну логічну схему пристрою в базисі Шеффера I-NI (для парного номера варіанту) і в базисі Пірса АБО-NI (для непарного номера варіанту), застосовуючи методи мінімізації заданої логічної функції за допомогою алгебраїчних перетворень із використанням карт Карно.

Методичні вказівки до виконання задачі № 1

Для побудови функціональної логічної схеми необхідно сформулювати умови її роботи і записати їх у вигляді логічної функції (ЛФ).

Синтез логічної схеми включає декілька етапів:

1) завдання ЛФ у вигляді таблиці істинності, в якій для кожного набору значень вхідних змінних вказують значення функції (0 або 1).

Таблиця 6.1 – Варіанти таблиць істинності комбінаційних вузлів

№ набору	A	B	C	D	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9
0	0	0	0	0	1	0	0	0	1	0	0	0	0	1
1	0	0	0	1	×	×	1	0	0	0	1	1	0	0
2	0	0	1	0	0	1	1	1	×	0	0	0	×	0
3	0	0	1	1	×	0	0	1	0	×	×	0	1	1
4	0	1	0	0	1	×	×	0	1	0	0	1	0	0
5	0	1	0	1	0	1	0	0	0	1	0	1	1	×
6	0	1	1	0	0	0	1	1	0	1	0	×	1	0
7	0	1	1	1	0	0	1	1	×	0	0	0	1	1
8	0	0	0	0	1	1	0	0	1	1	0	1	0	0
9	1	0	0	1	1	0	×	1	0	1	1	0	0	×
10	1	0	1	0	1	1	0	0	0	×	0	0	1	1
11	1	0	1	1	1	1	1	×	0	0	1	×	×	0
12	1	1	0	0	0	0	1	0	1	1	×	0	0	1
13	1	1	0	1	0	0	0	0	1	1	1	0	1	0
14	1	1	1	0	0	1	1	×	1	0	1	1	0	1
15	1	1	1	1	0	0	0	0	1	1	0	1	0	1

Знаком «×» показаний невизначений стан функції.

2) перехід від таблиці істинності до структурної формули в базисі І, АБО, НІ.

3) мінімізація ЛФ, з урахуванням можливості довизначення функції.

4) вибір елементної бази і запис структурної формули мінімізованої ЛФ у вибраному базисі (І-НІ і АБО-НІ).

5) побудова функціональної логічної схеми комбінаційного пристрою, послідовність з'єднання елементів якої визначається послідовністю виконання логічних операцій в структурній формулі.

Правило складання запису структурної формули у вигляді ДДНФ полягає в тому, що для кожного рядка таблиці істинності, в якій значення функції дорівнює «1», записується кон'юнкція (добуток) всіх вхідних змінних, а потім виконується логічне складання елементарних добутків. Якщо в даному наборі значення вхідних змінних дорівнюють «1» вони представляються у прямому вигляді, якщо значення якої-небудь вхідної змінної в рядку таблиці істинності дорівнює «0», то така змінна записується у вигляді своєї інверсії.

Правило складання запису структурної формули у вигляді ДКНФ полягає в тому, що для кожного рядка таблиці істинності, в якій значення функції дорівнює «0», записується диз'юнкція (сума) всіх вхідних змінних, а потім виконується логічний добуток елементарних сум. Якщо в даному наборі значення вхідних змінних дорівнюють «0» вони представляються у прямому вигляді, якщо значення якої-небудь вхідної змінної в рядку таблиці істинності дорівнює «1», то така змінна записується у вигляді своєї інверсії.

Розглянемо приклад складання структурної формули для ЛФ, заданою таблицею істинності (табл. 6.2) стосовно трьох вхідних змінних (А, В, С).

Таблиця 6.2 – Таблиця істинності для ЛФ трьох змінних

№ набору	А	В	С	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	x
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Складемо структурну формулу ЛФ у вигляді ДДНФ для одиничних значень функції F, керуючись вищевикладеними принципами:

$$F_1 = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C + A \cdot B \cdot \bar{C} + A \cdot B \cdot C.$$

Складемо структурну формулу ЛФ у вигляді ДКНФ для нульових значень функції F:

$$F_2 = (A + B + C) \cdot (A + \bar{B} + C) \cdot (\bar{A} + B + \bar{C}).$$

Виконаємо мінімізацію функцій за допомогою карт Карно.

Карта Карно являється прямокутником, розбитим на клітки, число яких дорівнює 2^n , тобто загальному числу наборів значень n змінних ЛФ. Таким чином, кожній клітці відповідає один єдиний набір значень змінних, що є координатами, на пересіченні яких знаходиться дана клітка. У кожній клітці проставляється значення функції, яке вона приймає на наборі значень змінних, що є її координатами. Тоді, при представленні функції в ДДНФ кожній клітці, що містить одиничне значення функції (наприклад, з координатами $A = 0, B = 0$ і $C = 1$), відповідатиме лише одна кон'юнкція змінних - та, яка набуває одиничного значення на наборі значень координат даної клітки (у нашому прикладі, кон'юнкція трьох змінних матиме вигляд: $\bar{A} \cdot \bar{B} \cdot C$).

При представленні ЛФ в ДКНФ кожній клітці, що містить 0, відповідатиме лише одна диз'юнкція змінних - та, яка набуває нульового значення на наборі значень координат цієї клітки. Так наприклад, для клітки що містить нульове значення функції при значеннях координат $A = 0, B = 0$ і $C = 0$, диз'юнкція трьох змінних матиме вигляд: $A + B + C$.

На рисунку 6.1 представлена карта Карно для трьох змінних A, B і C , в клітках якої проставлені значення функції F , на наборах значень координат, заданих таблицею 6.2.

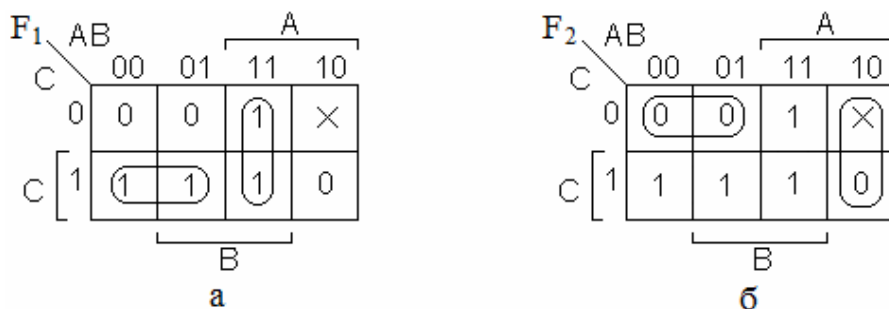


Рисунок 6.1 - Варіанти представлення карти Карно для функції F : а – для випадку запису F_1 в ДДНФ; б – для випадку запису F_2 в ДКНФ

У карті Карно координати сусідніх кліток повинні відрізнятися одна від одної значенням лише однієї змінної. Мінімізація ЛФ полягає в об'єднанні сусі-

дніх кліток, що містять «1» (для здобуття МДНФ) або «0» (для здобуття МКНФ), що дозволяє виключити одну змінну при об'єднанні двох кліток. Функція не визначена «X» в наборі змінних $A = 1, B = 0, C = 0$. Таким чином клітка $A \cdot \bar{B} \cdot \bar{C}$ при необхідності може прийняти значення «1» або «0» і функцію можливо довизначити. Цей прийом використаний при мінімізації функції F_2 .

Мінімізовані функції:

$$F_1 = \bar{A} \cdot C + A \cdot B; \quad F_2 = (\bar{A} + \bar{C}) \cdot (A + \bar{B})$$

Для запису отриманих в результаті мінімізації ЛФ в базисах І-НІ і АБО-НІ використовують закон подвійної інверсії і теорему Де Моргана (закон дуальності), згідно якому інверсія множення дорівнює сумі інверсій, а інверсія суми дорівнює множенню інверсій.

Представимо функцію F_1 в базисі І-НІ:

$$F_1 = \bar{A} \cdot C + A \cdot B = \overline{\overline{\bar{A} \cdot C + A \cdot B}} = \overline{\bar{A} \cdot C \cdot \bar{A} \cdot \bar{B}}$$

Представимо функцію F_2 в базисі АБО-НІ:

$$F_2 = (\bar{A} + \bar{C}) \cdot (A + \bar{B}) = \overline{\overline{(\bar{A} + \bar{C}) \cdot (A + \bar{B})}} = \overline{(\bar{A} + \bar{C}) + (A + \bar{B})}$$

Для технічної реалізації ЛФ використовується кількість логічних елементів типу І-НІ або АБО-НІ, що дорівнює числу інверсій в її алгебраїчному вираженні. На рисунку 6.2 представлена технічна реалізація функції F_1 на логічних елементах І-НІ, а на рисунку 6.3 – технічна реалізація функції F_2 на логічних елементах АБО-НІ.

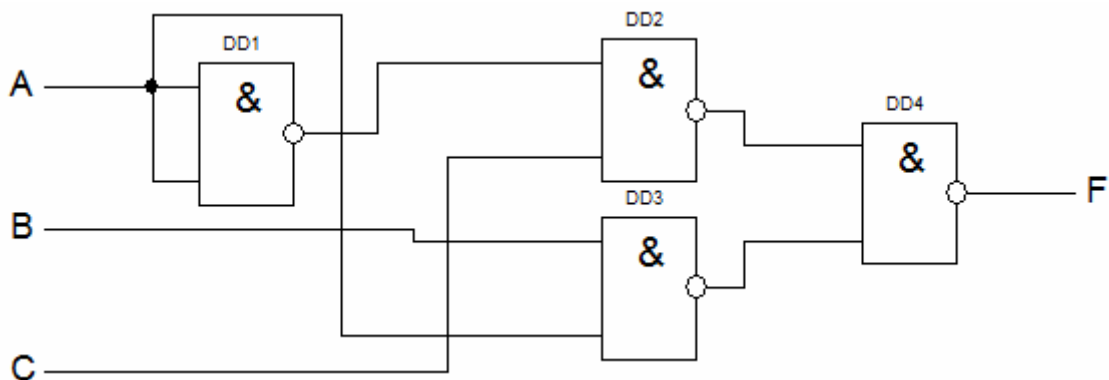


Рисунок 6.2 - Технічна реалізація функції F_1 на логічних елементах І-НІ

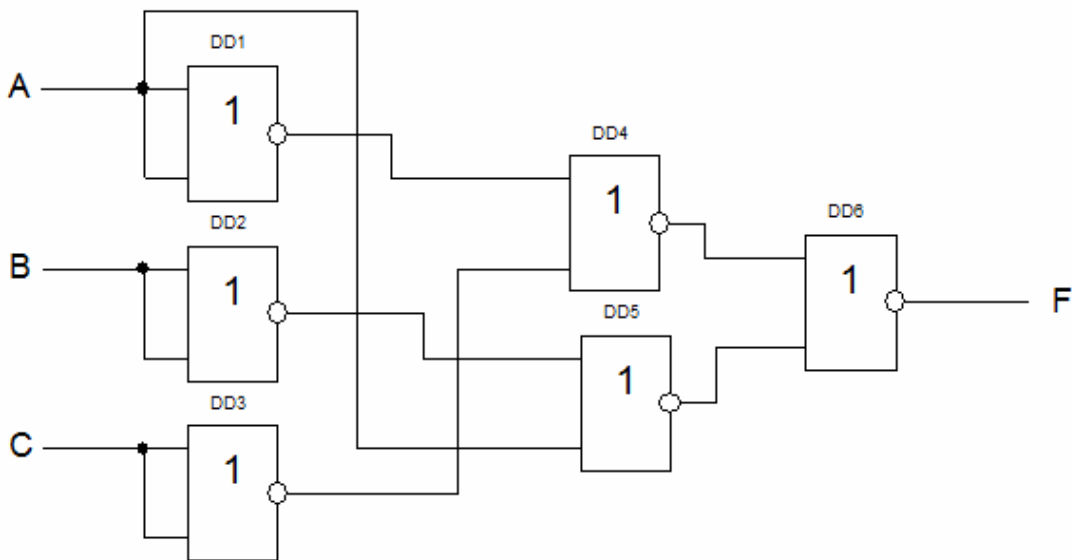


Рисунок 6.3 - Технічна реалізація функції F_2 на логічних елементах АБО-НІ

Задача №2

Провести синтез автомата Мілі, функціонування якого описується заданими таблицями переходів і виходів. Представити граф автомата, що синтезується. Задаючи довільну двійкову послідовність (вхідне слово), визначити відповідну двійкову вихідну послідовність (вихідне слово) автомата. Побудувати структурну схему синтезованого автомата в базисі І, АБО, НІ.

Варіанти таблиць переходів і виходів визначаються наступним чином. По останній цифрі номера варіанта з таблиці 6.3 визначається послідовність восьми станів (з чотирьох заданих: Q_0, Q_1, Q_2, Q_3). Ця послідовність рядками зліва направо і зверху вниз заноситься в таблицю переходів, що складається з двох рядків, верхній з яких визначає подальші стани автомата під впливом вхідного сигналу $x = 0$, а нижній під впливом вхідного сигналу $x = 1$.

Представити число з варіантів приведених в таблиці 6.4 в двійковій системі числення. Ця двійкова послідовність рядками зліва направо і зверху вниз заноситься в таблицю виходів, верхній рядок якої визначатиме вихідні сигнали автомата під впливом вхідного сигналу $x = 0$, а нижній під впливом вхідного сигналу $x = 1$.

Таблиця 6.3 - Варіанти таблиць переходів

№ набору	Послідовність станів автомата Мілі							
	Q0	Q3	Q2	Q1	Q3	Q1	Q2	Q0
0	Q0	Q3	Q2	Q1	Q3	Q1	Q2	Q0
1	Q2	Q1	Q3	Q0	Q3	Q1	Q0	Q2
2	Q1	Q2	Q0	Q3	Q3	Q2	Q0	Q1
3	Q3	Q0	Q1	Q2	Q0	Q1	Q3	Q2
4	Q3	Q0	Q1	Q2	Q0	Q3	Q2	Q1
5	Q0	Q3	Q2	Q2	Q3	Q2	Q1	Q0
6	Q0	Q1	Q3	Q1	Q3	Q2	Q0	Q2
7	Q2	Q3	Q0	Q2	Q1	Q2	Q0	Q3
8	Q2	Q0	Q1	Q3	Q0	Q2	Q3	Q1
9	Q2	Q0	Q2	Q0	Q3	Q1	Q0	Q2

Таблиця 6.4 - Варіанти завдань для заповнення таблиць виходів автомата

Варіант	1	2	3	4	5	6	7	8	9	10
Число	154	138	200	189	167	188	175	235	223	131
Варіант	11	12	13	14	15	16	17	18	19	20
Число	169	144	158	199	181	233	148	212	166	223

Розглянемо приклад визначення завдання для варіанту № 19. З приведеної таблиці 6.3 вибирається рядок під номером 9 (остання цифра варіанту №19). Число 166, вибране з таблиці 6.4 для варіанту №19, записується в двійковій системі числення як 10100110. Заповнюємо відповідні таблиці, як це показано на рисунку 6.4.

Таблиця переходів автомата Мілі

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

Таблиця виходів автомата Мілі

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

Рисунок 6.4 – Заповнення відповідних таблиць для варіанту №19

Методичні вказівки до виконання задачі № 2

Цифровий автомат – пристрій, що характеризується набором внутрішніх станів в які він попаде під впливом команд закладеної в нього програми. Перехід автомата з одного стану в інший здійснюється в певний момент часу.

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 автомат Мілі описується за допомогою двох таблиць. Одна з них (таблиця переходів) задає функцію $Q^{n+1} = f(Q^n(t), x(t))$, друга (таблиця виходів) – функцію $y(t) = f(Q^n(t), x(t))$ (рис. 6.4).

Таблиця переходів автомата Мілі

Вхідний сигнал	Q0	Q1	Q2	Q3
x1	Q2	Q0	Q2	Q0
x2	Q3	Q1	Q0	Q2

Таблиця виходів автомата Мілі

Вхідний сигнал	Q0	Q1	Q2	Q3
x1	y1	y2	y1	y2
x2	y2	y3	y4	y5

Рисунок 6.4 – Табличний спосіб завдання автомата Мілі

Кожному стовпцю з приведених таблиць поставлений у відповідність один стан з множини Q^n , кожному рядку – один вхідний сигнал з множини x_i .

На пересіченні стовпця Q^n і рядка x_i у таблиці переходів записується стан Q^{n+1} , у який повинен перейти автомат із стану Q^n під дією вхідного сигналу x_i , тобто $Q^{n+1} = f(Q^n, x_i)$. На пересіченні стовпця Q^n і рядка x_i у таблиці виходів записується вихідний сигнал y_i , що видається автоматом в стані Q^n під час приходу на вхід сигналу x_i , тобто $y_i = f(Q^n, x_i)$.

При графічному способі автомат задається у вигляді орієнтованого графа, вершини якого відповідають станам, а дуги - переходам між ними. Дуга, яка направлена з вершини Q^n , задає перехід в автоматі із стану Q^n в стан Q^{n+1} . На початку цієї дуги записується вхідний сигнал x_i , що викликає даний перехід $Q^{n+1} = f(Q^n, x_i)$. Для графа автомата Мілі вихідний сигнал y_i , що формується на переході, записується в кінці дуги. Якщо перехід в автоматі із стану Q^n в стан Q^{n+1} виконується під дією декількох вхідних сигналів, то дузі графа, направлений з Q^n в Q^{n+1} , приписуються всі ці вхідні і відповідні вихідні сигнали. Граф автомата Мілі заданий своїми таблицями переходів і виходів (рис. 6.4) представлений на рисунку 6.5.

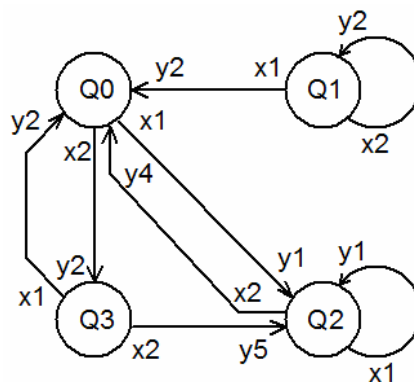


Рисунок 6.5 – Граф автомата Мілі заданий таблицями переходів і виходів

Для синтезу схем комбінаційного типу потрібний набір логічних елементів, що реалізують деяку повну систему логічних функцій (базис). Особливість схем послідовнісного типу полягає в тому, що вони володіють пам'яттю, для реалізації якої потрібні додаткові елементи. Такими елементами являються тригери.

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 переключити 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 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.

Розглянемо виконання цих етапів синтезу на прикладі реалізації автомата, закон функціонування якого заданий графом представленим на рисунку 6.6.

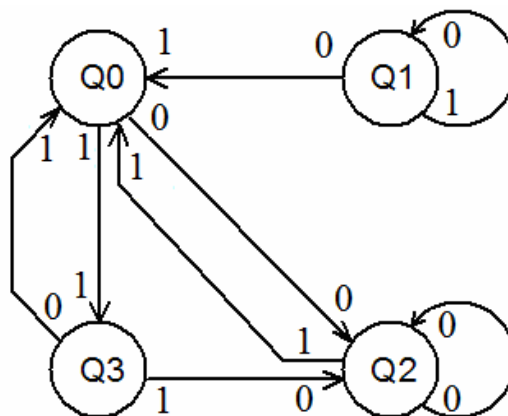


Рисунок 6.6 – Заданий граф функціонування автомата Мілі

Допустимо, для зберігання стану автомата використовується пам'ять, побудована на m тригерах, кожен з яких має два стани («0» або «1»). Отже, число різних комбінацій станів m тригерів дорівнює 2^m . Оскільки кожній з цих комбінацій може відповідати лише один певний стан автомата, що синтезується, то, вочевидь, вибір необхідного числа m тригерів повинен генеруватися з умови, що число станів автомата не перевищуватиме 2^m . Заданий за умовами завдання автомат має чотири стани, отже для його реалізації достатньо двох тригерів, тобто $m = 2$. Потім кожній комбінації станів тригерів потрібно поставити у відповідність певний стан автомата. Кодування станів автомата може здійснюватися різними способами. Для автомата, що синтезується, кодування його станів проведемо відповідно до таблиці 6.5.

Таблиця 6.5 – Кодування станів автомата Мілі

Стан автомату	Стан тригерів	
	F2	F1
Q0	0	0
Q1	0	1
Q2	1	0
Q3	1	1

Далі заповнимо таблицю 6.6 функціонування автомата, заданого графом, представленим на рисунку 6.6.

Таблиця 6.6 – Таблиця функціонування автомата Мілі

Вхідний сигнал x	Попередній стан		Сигнал стану		Сигнали керування тригерами				Вихідний сигнал y
	F2 ⁿ (t)	F1 ⁿ (t)	F2 ⁿ⁺¹ (t)	F1 ⁿ⁺¹ (t)	S2	R2	S1	R1	
0	0	0	1	0	1	0	0	×	0
0	0	1	0	0	0	×	0	1	1
0	1	0	1	0	×	0	0	×	0
0	1	1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	1	0	1
1	0	1	0	1	0	×	×	0	0
1	1	0	0	0	0	1	0	×	1
1	1	1	1	0	×	0	0	1	0

Розглянемо детальніше заповнення таблиці функціонування автомата Мілі. В перших трьох її стовпцях записуються всі можливі комбінації коду вхідного сигналу і стану автомата. Для заданого вхідного сигналу і стану автомата по графу знаходиться значення вихідного сигналу, яке записується в останньому стовпці таблиці, і наступний стан автомата, в який він переходить. Код цього стану заноситься в четвертий і п'ятий стовпці таблиці.

Стовпці з 6 по 9 відведені для запису сигналів управління тригерами. Управління тригерами здійснюється подачею сигналів на входи установки «0» (вхід R) і установки «1» (вхід S). Ці сигнали для кожного тригера визначаються порівнянням станів у момент часу $F^n(t)$ і в подальший момент часу $F^{n+1}(t)$. Наприклад, в першому рядку таблиці $F2^n(t) = 0, F2^{n+1}(t) = 1$. Це означає, що згідно словнику переходів RS – тригера (рис. 4.4 в) другий тригер переводиться із стану «0» в стан «1», для чого має бути поданий сигнал «1» на вхід S2 і «0» на вхід R2. У випадках коли логічний рівень сигналу управління нейтральний («0» або «1»), відповідні клітки таблиці залишаються порожніми або в них заноситься символ «x».

Для побудови комбінаційного пристрою, що формує сигнали управління тригерами, складемо для цих сигналів (S2, R2, S1, R1) таблиці істинності у формі карт Карно (рис. 6.7).

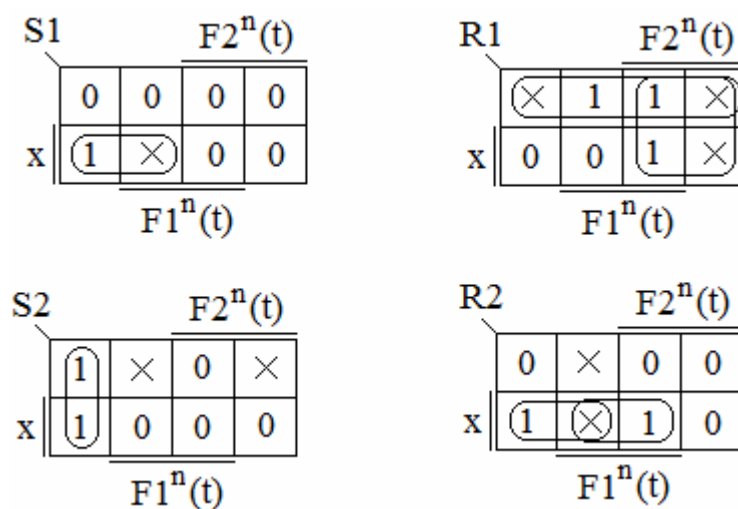


Рисунок 6.7 – Карти Карно для комбінаційного пристрою, що формує сигнали управління тригерами

Розглядаючи S2, R2, S1 і R1 як неповністю визначені логічні функції аргументів F2, F1 і x, запишемо МДНФ цих функцій:

$$S1 = \overline{F2} \cdot x; \quad R1 = F2 + \overline{x}; \quad S2 = \overline{F2} \cdot \overline{F1}; \quad R2 = \overline{F2} \cdot x + F1 \cdot x.$$

Для побудови комбінаційного пристрою, що формує вихідний сигнал автомата, будуємо таблицю істинності у формі карти Карно для y, представленої на рисунку 6.8.

		$F2^n(t)$			
		0	1	1	0
x	y	0	1	1	0
	x	1	0	0	1
		$F1^n(t)$			

Рисунок 6.8 - Карта Карно для комбінаційного пристрою, що формує сигнал виходу

МДНФ функції вихідного сигналу:

$$y = F1 \cdot \overline{x} + \overline{F1} \cdot x.$$

Використовуючи отримані логічні вирази і вибравши, як базис, логічні елементи І, АБО, НІ, побудуємо структурну схему автомата, який синтезується (рис. 6.9).

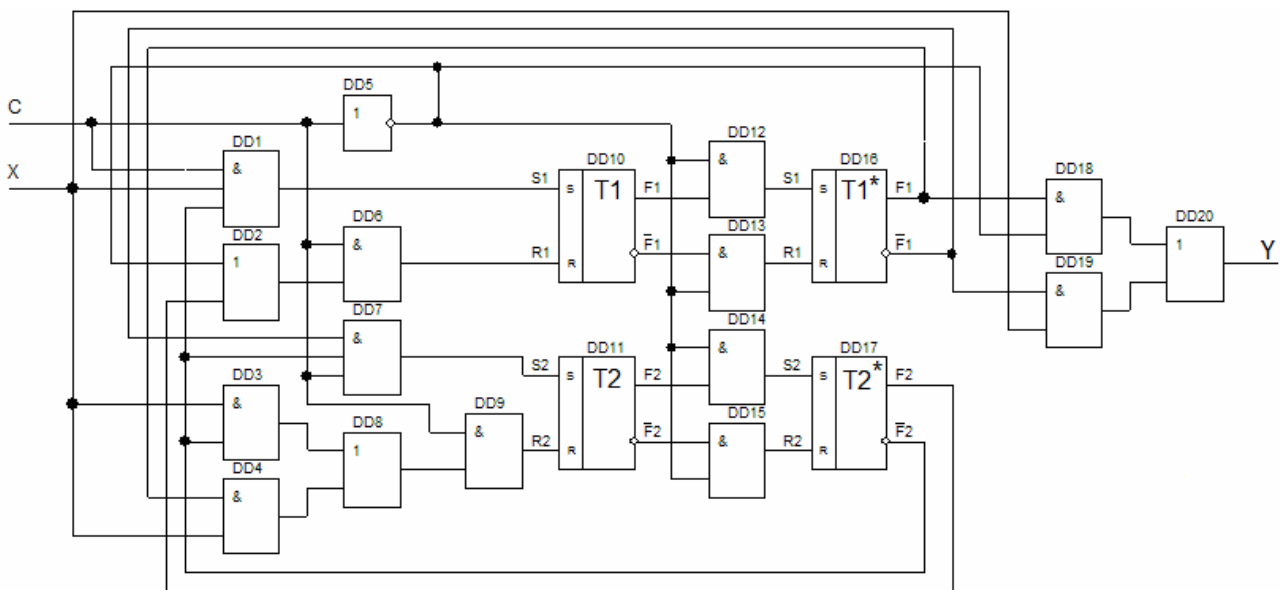


Рисунок 6.9 - Структурна схема синтезованого послідовнісного автомата

Мілі

Для забезпечення правильної роботи схеми автомата необхідно передбачити синхронізацію її функціонування в часі, тобто формування тактів (часових інтервалів) $1, 2, \dots, t, t+1, \dots$, під час дії яких здійснюються всі логічні залежності. На схемі (рис. 6.9) передбачений сигнал управління C , який в довільний момент часу t вирішує подачу управляючих сигналів з виходу комбінаційного пристрою на входи тригерів. При цьому, тригери $T1$ і $T2$ встановлюються в стани $F1^{t+1}$ і $F2^{t+1}$ відповідно до заданого таблицею переходів законом функціонування автомата. Проте, на час дії сигналу C (такту t) виходи (F і \bar{F}) вхідних тригерів $T1$ і $T2$ блоковані групою схем I , на вхід кожної з яких поступає сигнал, інверсний C . В цей же час група вихідних тригерів $F1^*$ і $F2^*$ зберігає свої стани $F1^t$ і $F2^t$, і лише їх вихідні сигнали поступають на входи обох комбінаційних схем і спільно з вхідним сигналом X визначають логіку роботи автомата. Після закінчення дії управляючого імпульсу C , протягом такту $(t+1)$, сигнали з виходів тригерів $T1$ і $T2$ через схеми I , на входи яких вже поступає вирішуючий інверсний сигнал C , подаються на відповідні входи тригерів $T1^*$ і $T2^*$, внаслідок чого останні переходять в стани $F1^{t+1}$ і $F2^{t+1}$. Завдяки наявності в схемі автомата додаткових тригерів $T1^*$ і $T2^*$, сигнали $F1^t$ і $F2^t$ на вході комбінаційного пристрою зберігаються незмінними протягом всього часу (такту t) дії управляючого імпульсу C . Це відповідає алгоритму роботи синхронного двоступінчатого (MS) RS – тригера. Основний рівень (M) служить для запису вхідного сигналу і попереднього запам'ятовування нового стану автомата, а допоміжний рівень (S) – для переходу автомата в новий стан і подальшого його зберігання.

Список літератури

1. Алексенко А.Г. Микросхемотехника [Текст] / А.Г. Алексенко, И.И. Шагурин. - М.: Радио и связь, 1982. – 416 с.
2. Папернов А.А. Логические основы цифровой вычислительной техники [Текст] / А.А. Папернов. – М.: Сов. Радио, 1972. – 345 с.
3. Основи схемотехніки електронних систем: Підручник [Текст] / [Бойко В.І., Гуржій А.М., Жуйков В. Я. та ін.]. – К.: Вища шк., 2004. – 527 с.: іл.
4. Бабич Н.П. Компьютерная схемотехника [Текст] / Н.П. Бабич, И.А. Жуков. – К: «МК-Пресс», 2004. – 576 с.
5. Преснухин Л.Н. Расчет элементов цифровых устройств [Текст] / Л.Н. Преснухин, Н.В. Воробьев, А.А. Шишкевич. – М.: Высш. Шк., 1991. – 526 с.
6. Прянишников В.А. Электроника: Курс лекций [Текст] / Прянишников В.А. – Санкт Петербург.: «КОРОНА принт», 1998. – 398 с.
7. Бойт К. Цифровая схемотехника [Текст] / Клаус Бойт. – Москва: Техносфера, 2007. – 472 с.
8. Савельев А.Я. Арифметические и логические основы цифровых автоматов [Текст] / А.Я. Савельев. – М.: Высш. шк., 1980. – 255 с.
9. Китаев Ю.В. Основы цифровой техники: Учебное пособие [Текст] / Ю.В. Китаев. – СПб: СПбГУ ИТМО, 2007. – 87 с.
10. Корис Р. Справочник инженера-схемотехника [Текст] / Р. Корис, Х. Шмидт-Вальтер. – Москва: Техносфера, 2008. – 608с.
11. Мальцева Л.А. Основы цифровой техники [Текст] / Л.А. Мальцева, Э.М. Фромберг, В.С Ямпольский. – М.: Радио и связь, 1987. – 128 с.
12. Угрюмов Е.П. Цифровая схемотехника [Текст] / Е.П. Угрюмов. – СПб.: БХВ – Санкт-Петербург, 2000. – 528 с: ил.
13. Чье ЕнУн Электроника. Цифровые элементы и устройства: Учебное пособие [Текст] / Чье ЕнУн. – Хабаровск: Изд-во Хабар. гос. техн. ун-та, 2002. – 97 с.

14. Титов В.С. Проектирование аналоговых и цифровых устройств: Учебное пособие [Текст] / В.С. Титов, В.И. Иванов, М.В. Бобырь. – М.: ИНФРА–М, 2014. – 143 с.
15. Миловзоров О.В. Основы электроники : Учебник для СПО [Текст] / О.В. Миловзоров, И. Г. Панков. – 5-е изд., перераб. и доп. – М. : Издательство Юрайт, 2016. – 407 с.
16. Новнков Ю.В. Введение в цифровую схемотехнику [Текст] / Ю.В. Новнков. – М.: Национальный открытый университет "ИНТУИТ", 2016 – 393 с.
17. Мышляева И.М. Цифровая схемотехника: Учебник [Текст] / И.М. Мышляева. – М.: Академия, 2005. – 400 с.: ил.
18. Скаржепа В.А. Электроника и микросхемотехника. Электронные устройства информационной автоматики: Учебник [Текст] / В.А. Скаржепа, А.И. Луценко. – Киев: Выща шк., 1989. – Ч.1. – 431 с.
19. Гусев В.Г., Гусев Ю. М. Электроника: Учеб. Пособие [Текст] / В.Г. Гусев, Ю.М. Гусев. – М.: Высш. шк., 1991. – 622 с.
20. Букреев И.Н. Микроэлектронные схемы цифровых устройств [Текст] / И.Н. Букреев, В.И. Горячев, Б.М. Мансуров. – М.: Радио и связь, 1990. – 416 с.
21. Пухальский Г.И. Проектирование дискретных устройств на интегральных микросхемах: Справочник [Текст] / Г.И. Пухальский, Т.Я. Новосельцева. – М.: Радио и связь, 1990. – 304 с.
22. Потемкин И.С. Функциональные узлы цифровой автоматики [Текст] / И.С. Потемкин. – М.: Энергоатомиздат, 1988. – 320 с.
23. Янсен И. Курс цифровой электроники: в 4-х т. Пер. с голланд. [Текст] / И. Янсен – М.: Мир, 1987. – 435 с.
24. Постников А.И. Основы теории цифровых автоматов: Учеб. пособие для вузов по спец. Вычислительные машины, комплексы, системы и сети [Текст] / А.И. Постников. – Красноярск: КГТУ, 2000. – 296 с.
25. Хоровиц П. Искусство схемотехники: В 3 т.: Пер. с англ. [Текст] / П. Хоровиц, У. Хилл – М.: Мир, 1993. – Т.2. – 367 с.

26. Токхейм Р. Основы цифровой электроники: Пер. с англ. [Текст] / Р. Токхейм. – М.: Мир, 1988. – 392 с.
27. Старос Ф.Г. Полупроводниковые интегральные запоминающие устройства [Текст] / Ф.Г. Старос, Л.П. Крайзмер. – Л.: Энергия, 1973. – 112 с.
28. Караханян Э.Р. Динамические интегральные схемы памяти с МДП структурой [Текст] / Э.Р. Караханян, В.А. Шилин. – М.: Радио и связь, 1984. – 136 с.
29. Гершунский Б.С. Основы электроники и микроэлектроники [Текст] / Б.С. Гершунский - К.: Вища шк., 1987. – 422 с.
30. Степаненко И.П. Основы микроэлектроники [Текст] / И.П. Степаненко. – М.: Сов. Радио, 1980. – 424 с.
31. Якубовский С.В. Аналоговые и цифровые интегральные микросхемы [Текст] / С.В. Якубовский, Н.А. Барканов, Б.П. Кудряшов. – М.: Радио и связь, 1985. – 365 с.

Підписано до друку __.__.2016 р.
Формат 60×84 1/32. Папір офсетний.
Умовн. друк. арк. 6,7. Наклад прим.
Замовлення № _____

Запорізька державна інженерна академія
Свідоцтво про внесення до Державного реєстру суб'єктів
видавничої справи ДК №2958 від 03.09.2007 р.

Віддруковано друкарнею
Запорізької державної інженерної академії
з оригінал-макету авторів

69006, м. Запоріжжя, пр. Соборний, 226, РВВ
ЗДІА,

Тел. 2238-240