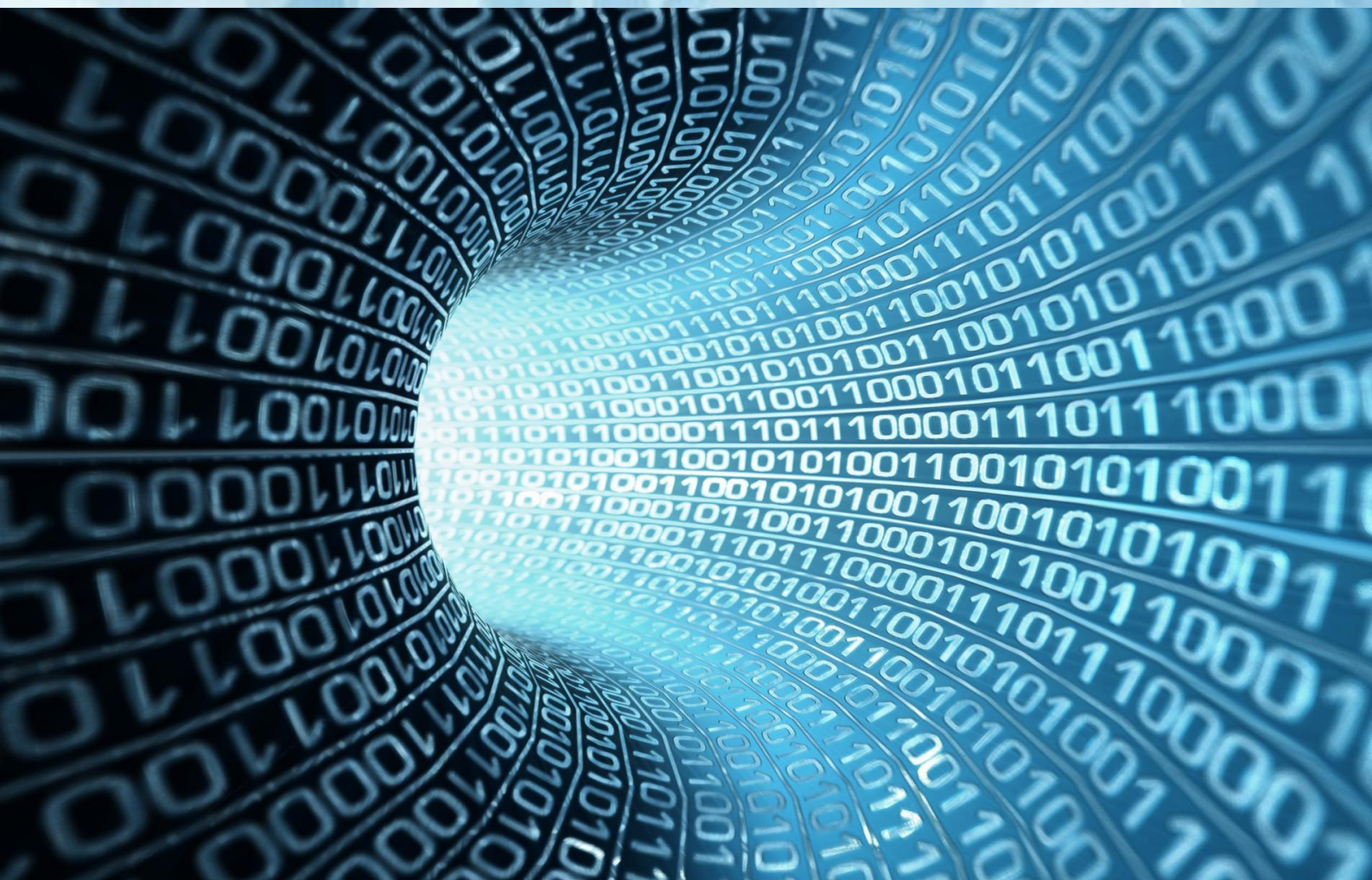


**С. Г. Кривогубченко
Ю. Ю. Іванов
Д. С. Кривогубченко
Я. А. Кулик
В. М. Папінов**

КОМП'ЮТЕРНА ЕЛЕКТРОНІКА



ЗМІСТ

ВСТУП	6
1 ОСНОВИ ЕЛЕКТРОНІКИ	7
1.1 Взаємозв'язок напруги і струму. Резистори	7
1.2 Конденсатори	9
1.3 Індуктивності	10
1.4 Мемристори	11
1.5 Напівпровідникові прилади	13
1.5.1 Діоди	14
1.5.2 Стабілітрони	16
1.5.3 Світлодіоди	16
1.5.4 Фотодіоди	17
1.5.5 Оптрони	18
1.5.6 Біполярний транзистор	18
1.5.7 Польові транзистори	22
1.6 Інтегральні схеми	24
2 ПРЕДСТАВЛЕННЯ ДАНИХ В КОМП'ЮТЕРІ	26
2.1 Системи числення	26
2.1.1 Основні визначення. Позиційні системи числення.	26
2.1.2 Переведення чисел із однієї позиційної системи числення в іншу	29
2.2 Текстові дані	31
2.3 Графічні дані	33
2.4 Звукові і відеодані	35
2.5 Числові дані	37
2.5.1 Формати цілих чисел	37
2.5.2 Формат дійсних чисел	42
3 ЛОГІЧНІ ОСНОВИ ОБРОБЛЕННЯ ДАНИХ	45
3.1 Поняття такту	45
3.2 Вентилі і комбінаційні схеми	46
3.2.1 Релейно-контактні вентилі	46
3.2.2 Напівпровідникові вентилі	47
3.2.3 Вентиль "НІ"	48
3.2.4 Вентилі "АБО НІ" та "І НІ"	48
3.2.5 Вентилі "І" та "АБО"	49
3.2.6 Синтез комбінаційних схем	50
3.2.7 Вентиль "Виключне АБО"	54
3.2.8 Багатовходові вентилі	54
3.2.9 Комбінаційна схема суматора	55
3.2.10 Комбінаційна схема зсуву	57
3.2.11 Компаратор	59
3.2.12 Перетворювачі кодів	60
3.2.13 Арифметико-логічний пристрій	70
3.3 Схеми пам'яті на базових вентилях	71
3.3.1 RS – тригери	72

3.3.2 D – тригери	75
3.3.3 Лічильні тригери (Т – тригери)	76
3.3.4 Універсальні тригери (JK - тригери)	77
3.4 Регістри	78
3.5 Лічильники	81
3.6 Напівпровідникові запам'ятовуючі пристрої	83
4 СХЕМОТЕХНІКА НАЙПРОСТІШИХ ЦИФРОВИХ ПРИСТРОЇВ	87
4.1 Формувачі та генератори імпульсів	87
4.1.1 Одновібратори	88
4.1.2 Одновібратори на основі ємнісного зв'язку	89
4.1.3 Типові схеми одновібраторів	90
4.2 Схеми виділення перепадів	93
4.2.1 Схеми виділення перепадів на основі диференційних кіл	93
4.2.2 Схеми виділення перепадів на основі диференційних кіл і логічних елементів	94
4.2.3 Схеми виділення перепадів з "0" в "1" та з "1" в "0" на основі інтегруючих кіл і логічних елементів	95
4.2.4 Схеми виділення перепадів, які використовують властивість затримки вхідних сигналів логічними елементами	95
4.3 Генератори прямокутних імпульсів	97
4.3.1 Мультивібратор	98
4.3.2 Генератори прямокутних імпульсів з одним RC колом	99
4.3.3 Генератори прямокутних імпульсів на основі тригерів	100
4.3.4 Генератори прямокутних імпульсів на основі інтегральних ключів	101
4.4 Схеми затримки імпульсів	102
5 ІНТЕГРАЛЬНИЙ ТАЙМЕР	105
5.1 Робота таймера в режимі одновібратора	105
5.2 Робота таймера в режимі генератора прямокутних імпульсів	106
6 ЗВОРОТНІЙ ЗВ'ЯЗОК ТА ОПЕРАЦІЙНІ ПІДСИЛЮВАЧІ	108
6.1 Інвертуючий підсилювач	109
6.2 Неінвертуючий підсилювач	110
6.3 Повторювач	110
6.4 Перетворювач струму в напругу	111
6.5 Диференційний підсилювач	111
6.6 Підсумовуючий підсилювач	112
6.7 Інтегратори	112
6.8 Диференціатори	113
6.9 Компаратори	113
6.10 Розрахунок ланцюгів, які містять ОП, за допомогою графів	113
7 ПЕРЕТВОРЮВАЧІ АНАЛОГОВИХ ТА ЦИФРОВИХ СИГНАЛІВ	117
ЛІТЕРАТУРА	118

ВСТУП

Електроніка взагалі – це наука (або мистецтво) про побудову різноманітних за функціями та цілями електронних вузлів і пристроїв. При цьому функції, реалізовані схемами, можуть накладати специфічні обмеження на шляху досягнення поставлених цілей. У цьому плані можна говорити про схемотехніку, наприклад, промислової автоматики, силової чи промислової електроніки, радіотехнічної схемотехніки. Безсумнівно, специфічною областю схемотехніки є і схемотехніка комп'ютерів. Англomовний термін «комп'ютер» (від computer – обчислювач) широке поширення одержав після 1985 р. Комп'ютер являє собою універсальний пристрій, який використовується для автоматизації процесів приймання, зберігання, оброблення і передавання інформації, які здійснюються заздалегідь розробленими людиною програмами. Тобто для виконання основних функцій комп'ютера передбачені спеціальні пристрої:

- пам'ять – група пристроїв, які забезпечують зберігання програм і даних;
- процесор (від process – оброблення) – один або декілька пристроїв, які забезпечують оброблення даних, що задаються програмою;
- пристрої введення / виведення – група пристроїв, які забезпечують обмін даними між користувачем і комп'ютером; між двома або більше комп'ютерами.

Різні пристрої комп'ютера з'єднують один з одним за допомогою стандартизованих та уніфікованих апаратних засобів – кабелів, з'єднувачів і т.д. При цьому пристрої обмінюються один з одним інформацією і керуючими сигналами, які також наводяться до деяких стандартних форм. Сукупність цих стандартних засобів і форм утворює конкретний **інтерфейс**.

Інтерфейсом називається сукупність уніфікованих стандартних угод, апаратних і програмних засобів, методів і правил взаємодії пристроїв та програм між собою або з користувачем.

Під час комп'ютерного оброблення інформації доводиться мати справу з даними різної природи, для зберігання яких застосовуються різні способи їх подання в двійковому алфавіті: {0, 1}, {true, false}, пара напруг {1 В, 5 В} і т.д. Двійковий розряд представляє собою найменшу кількість інформації, яка називається **бітом**. Основною одиницею виміру обсягу пам'яті є **байт** і ряд кратних одиниць, які утворюються за допомогою так званої двійкової тисячі - $2^{10} = 1024$. Перша – кілобайт (**Кбайт**), наступна – мегабайт (**Мбайт**): 1 Мбайт = 1024 Кбайт = 1048576 байт. Далі гігабайт (**Гбайт**, близько мільярда байт) і терабайт (**Тбайт**, близько трильйона байт)...

У навчальному посібнику розглядаються основи електроніки та схемотехнічні елементи, способи представлення даних в комп'ютері, вентиля та комбінаційні схеми, схеми пам'яті на базових вентилях, схемотехніка найпростіших цифрових пристроїв, інтегральний таймер, операційні підсилювачі та перетворювачі аналогових та цифрових сигналів.

1 ОСНОВИ ЕЛЕКТРОНІКИ

1.1 Взаємозв'язок напруги і струму. Резистори

Сутність електроніки полягає у взаємозв'язку напруги і струму. Напруга створюється шляхом впливу на електричні заряди в таких пристроях, як батареї (електрохімічні реакції), генератори (взаємодія магнітних сил), сонячні батареї (фотогальванічний ефект енергії фотонів) і т.п. Струм ми отримуємо, прикладаючи напругу між точками схеми. На схемах та в формулах **напруга** позначається літерою U , а **струм** – I і вимірюються відповідно ці величини у вольтах (**В**) та амперах (**А**). Загальноприйнятий напрямок струму у зовнішньому ланцюгу – від позитивного електроду джерела живлення до негативного. Проходячи по електричному ланцюзі, струм здійснює роботу, яка перетворюється на теплову, механічну та інші види енергії. Робота, здійснена струмом за одиницю часу, називається **потужністю** (P), вимірюється у ватах (**Вт**) і визначається як: $P = U \cdot I$.

Для характеристики здатності речовини проводити електричний струм були введені поняття **електричної провідності** (G) і величини зворотної провідності – електричного опору (R). **Електричний опір** як елемент електричного ланцюгу (рис. 1.1) називається **резистором**, вимірюється у омах (**Ом**) і визначається наступним чином: $R = U / I$. Це співвідношення називається законом **Ома**. Для утворення кратних одиниць вимірювання застосовуються наступні приставки:

тера	Т	10^{12}	гіга	Г	10^9	мега	М	10^6	кіло	к	10^3
піко	пк	10^{-12}	нано	н	10^{-9}	мікро	мк	10^{-6}	мілі	м	10^{-3}

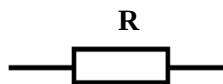


Рисунок 1.1 – Умовне позначення резистора

Елементи електричних ланцюгів можуть з'єднуватися як послідовно (рис. 1.2 а) так і паралельно (рис. 1.2 б).

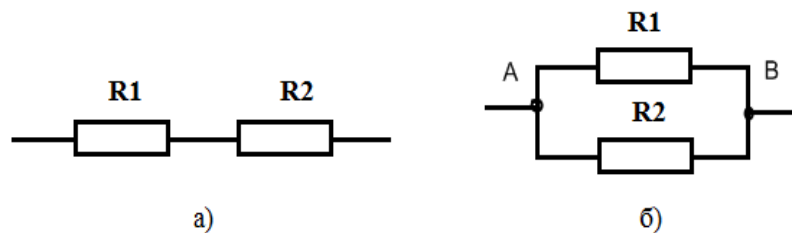


Рисунок 1.2 – З'єднання резисторів

Параметри послідовного кола (рис. 1.2, а) визначаються у відповідності до наступних законів:

- сила струму через будь-яку ділянку кола однакова, тобто $I = I_1 = I_2$;
- загальний опір кола рівний сумі опорів його окремих ділянок, тобто $R = R_1 + R_2$ і завжди отримуємо більший опір, ніж опір окремого резистора;
- напруга на ділянках кола прямо пропорційна опорам цих ділянок;
- напруга, прикладена до зовнішнього кола, дорівнює сумі напруг ділянок, тобто $U = U_1 + U_2$.

Аналогічно для паралельного кола характерні наступні закони:

- падіння напруг на всіх гілках рівні між собою, тобто $U_{AB} = U_1 = U_2$;
- сума падінь напруги в будь-якому замкнутому контурі дорівнює нулю. Це закон **Кірхгофа** для напруг.
- сума струмів, які виходять з точки розгалуження (вузла), дорівнює сумі струмів, які входять в неї (закон **Кірхгофа** для струмів), тобто $I = I_1 + I_2$;
- розподілення сили струму по гілках зворотно пропорційне опорам гілок;
- опір двох паралельно з'єднаних резисторів дорівнює $R = R_1 \cdot R_2 / (R_1 + R_2)$, причому завжди отримуємо менший опір, ніж опір окремого резистора [1].

Резистори мають широке застосування. На рис. 1.3 наведена схема подільника напруги, який використовується в схемах для отримання необхідної напруги: $U_{вих.} = I \cdot R_2 = U_{вх.} \cdot R_2 / (R_1 + R_2)$.

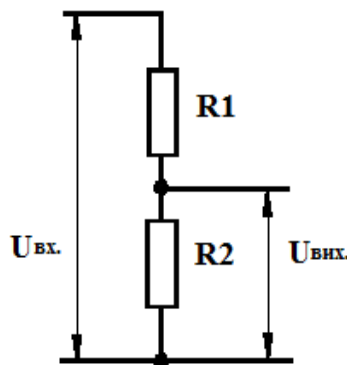


Рисунок 1.3 – Подільник напруги

Окрім вказаних раніше послідовного та паралельного з'єднання елементів, на практиці застосовуються складніші з'єднання, які мають, крім того, і декілька джерел живлення [2]. Один з варіантів розрахунку таких кіл описується законом Кірхгофа для напруг. Для ілюстрації розглянемо наступний приклад, в якому необхідно визначити напрям і величину струму, який проходить через резистор R_3 (рис. 1.4).

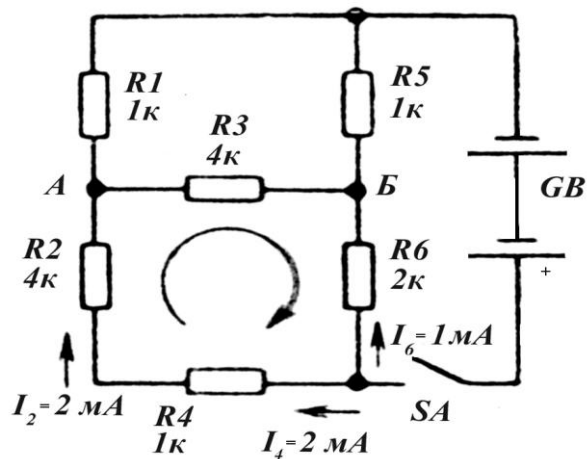


Рисунок 1.4 – Закон Кірхгофа для напруг

Для розрахунку вибираємо контур $R2R3R6R4$. За умовою нам відомі опори цих резисторів і величини струмів, які проходять через них. Оберемо напрям обходу контуру за годинниковою стрілкою і допустимо, що через резистор $R3$ струм тече від точки А до точки Б. Так як напрям обходу контуру співпадає з напрямом струмів через резистори $R2$, $R3$ і $R4$, то значення сили струму цих ділянок кола підставляються у формулу з додатнім знаком. Напрямок струму через резистор $R6$ не співпадає з обраним напрямом обходу контуру, відповідно, сила струму цієї ділянки кола підставляється в формулу з від'ємним знаком. Джерело живлення в контурі відсутнє, тому вважаємо, що $E = I_4 R_4 + I_2 R_2 + I_3 R_3 - I_6 R_6 = 0$. Звідки $I_3 = -2 \text{ mA}$.

Оскільки величина струму вийшла від'ємною, то можна зробити висновок, що в дійсності струм через резистор $R3$ протікає від точки Б до точки А.

1.2 Конденсатори

Як і резистори, конденсатори (рис. 1.5) широко використовуються в електронних пристроях. Електрична ємність (конденсатор) – елемент з властивістю накопичувати й утримувати електричні заряди. В ідеальному випадку залежність струму конденсатора від напруги на ньому описується виразом: $I = C \cdot (dU/dt)$.

Таким чином, струм конденсатора пропорційний не просто напрузі, а швидкості зміни напруги.

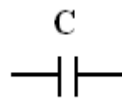


Рисунок 1.5 – Умовне позначення конденсатора

Коефіцієнт пропорційності C – це **ємність конденсатора**, що вимірюється у фарадах (Φ). Ємність в 1 Φ дуже велика, тому на практиці мають справи з мікрофарадами (**мкФ**), нанофарадами (**нФ**), пікофарадами (**пФ**). Ємність C визначає здатність конденсатора накопичувати **заряд**. Це видно з наступної властивості конденсатора: $Q = C \cdot U$.

Тобто конденсатор, що має ємність C фарад, до якого прикладена напруга U вольт, накопичує **заряд Q кулон**. Очевидно, що чим більше ємність конденсатора, тим більший заряд він може накопичити при однаковій напрузі.

При ввімкненні конденсатора в коло постійного струму в ньому відбуватиметься перехідний процес (рис. 1.6). Він полягає в тому, що напруга на конденсаторі збільшується поступово, за експоненціальним законом. Тривалість перехідного процесу (постійна часу), тобто час зарядження конденсатора до 95% або розрядження до 5% від максимальної величини напруження, можна визначити за формулою: $\tau_c = 3RC$.



Рисунок 1.6 – Перехідний процес при зарядженні–розрядженні конденсатора

В електроніці при розрахунку постійних часу τ_c кола часто за точку відліку приймають: при зарядці – 63%, при розрядці – 37% від величини максимального струму (напруги). В цьому випадку $\tau_c = RC$.

Електричні характеристики конденсаторів залежать від типу діелектрика між його обкладинками.

Як і інші елементи, конденсатори можуть з'єднуватися послідовно і паралельно. Ємність паралельно з'єднаних конденсаторів дорівнює сумі ємностей, тобто: $C = C_1 + C_2 + C_3 + \dots$ а для послідовного - маємо такий же вираз, що й для паралельного з'єднання резисторів: $1/C = 1/C_1 + 1/C_2 + 1/C_3 + \dots$

1.3 Індуктивності

В ідеальному індуктивному елементі (рис. 1.7) напруга на індуктивності прямо пропорційна швидкості зміни струму: $U = L \cdot (dI/dt)$, де L – коефіцієнт пропорційності між напругою і швидкістю зміни струму, який називається коефіцієнтом самоіндукції або індуктивністю. Індуктивність вимірюється в генрі (**Гн**).

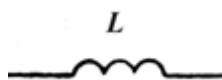


Рисунок 1.7 – Умовне позначення індуктивності

Індуктивний елемент має властивості, протилежні властивостям конденсатора. Так, постійна напруга, прикладена до індуктивності, викликає наростання струму, що протікає через нього (рис. 1.8). Час (тривалість) перехідного процесу: $\tau_L = 3L/C$.



Рисунок 1.8 – Перехідні процеси в колі постійного струму з індуктивністю

Котушки індуктивності, як правило, мають циліндричну або спіральну форму витків і виконуються як одношаровими, так і багатошаровими. Для збільшення значень індуктивності застосовують магнітопроводи. Матеріалом магнітопроводу найчастіше служить залізо або ферит.

Індуктивності, головним чином, використовуються у вибіркових колах, фільтрах електричних сигналів. Один з різновидів котушок індуктивностей є дроселі. Їх основне призначення – забезпечити великий опір для змінних струмів і малий опір для постійних або низькочастотних струмів.

1.4 Мемристори

Електричний ланцюг може описуватися чотирма фізичними величинами: в кожній точці (перерізу) – силою струму (I) і зарядом (Q), між двома точками (поверхнями) – напругою або різницею потенціалів (U) і магнітним потоком (Φ). Всі ці чотири величини попарно співвідносяться одна з одною, причому ці співвідношення представлені в фізичних елементах електросхеми (рис. 1.9). Так, резистор (опір) реалізує взаємозв'язок сили струму і напруги, конденсатор (ємність) – напруги і заряду, котушка індуктивності – магнітного потоку і сили струму. Ці три пасивні елементи – резистор, конденсатор і котушка індуктивності, вважаються базовими в електротехніці, оскільки електричну схему будь-якої складності теоретично можна звести до еквівалентної схеми, побудованої виключно з опорів, ємностей та індуктивностей. У 1971 році американський фізик Леон О. Чуа із Каліфорнійського університету в Берклі висунув гіпотезу, згідно якої повинен існувати четвертий базовий елемент електросхеми, який описував би взаємозв'язок магнітного потоку з зарядом.

Мемристор (англ. memristor, від *memori* – пам'ять, та *resistor* – опір) – це пасивний двополюсник з нелінійною вольт-амперною характеристикою, що має гістерезис "ефект пам'яті", тобто властивості елемента залежать від прикладеної раніше сили. В даному випадку опір мемристора залежить від пропущеного через нього заряду, що і дозволяє використовувати його в якості комірки пам'яті. Ця властивість була названа мемрезистивністю (M), значенням якої є відношення зміни магнітного потоку до зміни заряду.

Величина M залежить від кількості заряду, який пройшов через елемент, тобто від того, як довго через нього протікав електричний струм.

Принципова відмінність мемристора від більшості типів сучасної напівпровідникової пам'яті те, що він не зберігає свої властивості у вигляді заряду і повністю енергонезалежний. Тобто дані можуть зберігатися в мемристорі до тих пір, поки існують матеріали, з яких він виготовлений. Для порівняння: флеш–пам'ять починає втрачати записану інформацію вже після року зберігання без доступу до електричного струму.

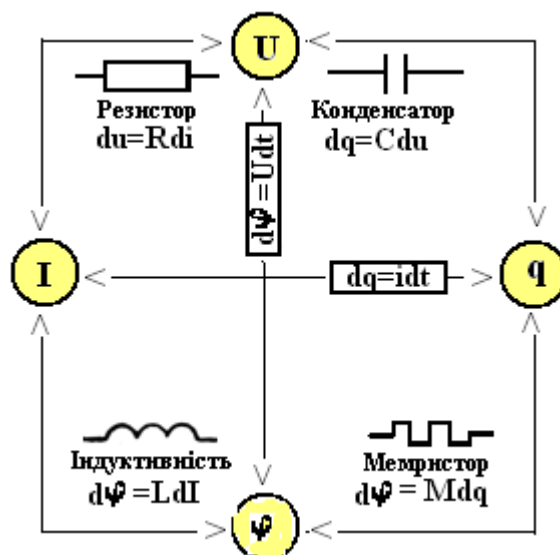


Рисунок 1.9 – Взаємозв'язок електричних величин в елементах електросхеми

Конструктивно мемристори значно простіше флеш–пам'яті: вони складаються з тонкої 50-нм плівки, яка складається з двох шарів – ізолюючого діоксиду титану і шару, збідненого киснем. Плівка розташована між двома платиновими 5-нм електродами. Під час подачі на електроди напруги змінюється кристалічна структура діоксиду титана: завдяки дифузії кисню його електричний опір збільшується на кілька порядків (в тисячі разів). При цьому після відключення струму зміни в комірці зберігаються. Зміна полярності поданого струму перемикає стан комірки, причому число таких перемикачів не обмежене. На практиці мемристор може приймати не тільки звичайні два положення – 0 або 1, а й будь-які інші значення в проміжку від 0 до 1 так, що такий перемикач здатний працювати і в цифровому (дискретно), і в аналоговому режимах.

Уже вивчені властивості мемристорів дозволяють говорити про те, що на їх основі можна створювати комп'ютери принципово нової архітектури, які по продуктивності значно перевищують напівпровідникові і здатні моделювати роботу людського мозку, в якому немає якогось єдиного центру збереження та оброблення інформації. Саме мемристорний комп'ютер стане проміжним ступенем на шляху до квантового комп'ютера [3].

1.5 Напівпровідникові прилади

Напівпровідниками є речовини, які займають проміжне місце по електропровідності між провідниками та діелектриками [4]. Особливістю напівпровідників є те, що під впливом зовнішньої дії або при введенні певних домішок напівпровідник різко змінює свою електропровідність. При докладанні електричної напруги до напівпровідника електрони почнуть рухатись в одному напрямі від одного атому до другого. Одночасно буде спостерігатись рух дірок у протилежному напрямі (рис. 1.10).

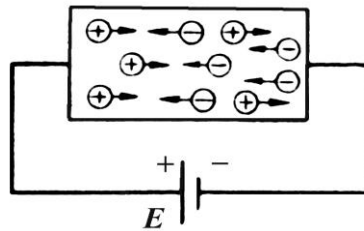


Рисунок 1.10 – Напівпровідник в електричному колі

Якщо напівпровідник має дві зони з різними типами провідності, то межа розділу цих зон має назву електронно – дірковим переходом або $p-n$ переходом (рис. 1.11, а). Характерною особливістю $p-n$ переходу є залежність його опору (провідності) від полярності докладеної напруги.

При підключенні позитивного полюса джерела напруги до зони з провідністю p -типу, а негативного – з провідністю n -типу дірки будуть відштовхуватись позитивним потенціалом джерела напруги, а електрони – негативним. Вони будуть рухатись назустріч одне одному до $p-n$ переходу, де відбудеться їх взаємопереміщення (рекомбінація) в області зони переходу. Пройшовши зону переходу, носії заряду попадають під вплив відповідних полюсів джерела напруги і притягуються ними. Таким чином, можна сказати, що через перехід протікає електричний струм. Величина цього струму буде залежати від величини докладеної напруги. В цьому випадку говорять, що перехід зміщений в прямому напрямку або відімкнутий.

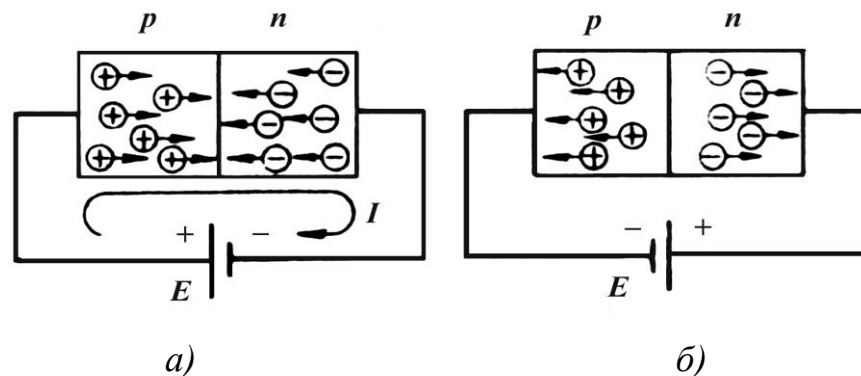


Рисунок 1.11 – $P-n$ перехід в електричному колі

Якщо замінити полярність увімкнення джерела напруги, то електрони будуть притягуватись до позитивного, а дірки – до негативного електродів джерела (рис. 1.11, б). В результаті цього $p-n$ перехід об'єднається з носіями заряду і струм через нього проходити не буде. В цьому випадку перехід зміщений у зворотному напрямку, або замкнений.

1.5.1 Діоди

Процеси, які відбуваються в провіднику з $p-n$ переходом, лежать у основі роботи різноманітних напівпровідникових пристроїв.

Зокрема, діоди (рис. 1.12) представляють собою двохелектродний напівпровідниковий прилад з одним $p-n$ переходом, один з виводів якого (анод) підключений до області напівпровідника з провідністю p -типу, а інший (катод) – до області з провідністю n -типу.

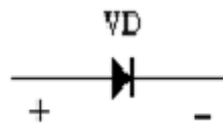


Рисунок 1.12 – Умовне позначення діода

При підключенні напруги позитивної полярності до аноду, а негативної – до катода перехід зміщується в прямому напрямку і діод відкривається. При зміні полярності діод закривається.

Залежність величини струму через діод від величини і полярності прикладеної напруги називається вольт-амперною характеристикою (ВАХ) діоду (рис. 1.13).

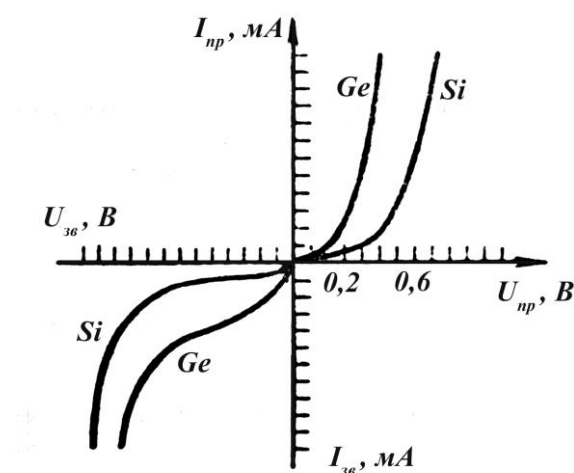


Рисунок 1.13 – Вольт-амперна характеристика напівпровідникового діоду

До основних електричних характеристик діодів відносяться: максимально допустимий прямий струм, максимально допустима зворотна

напруга, максимальний зворотній струм, напруга на відкритому діоді, максимальна потужність, яка розсіюється діодом.

Діоди знаходять широке застосування. На рис. 1.14 наведена схема випрямляча напруги та часові діаграми роботи.

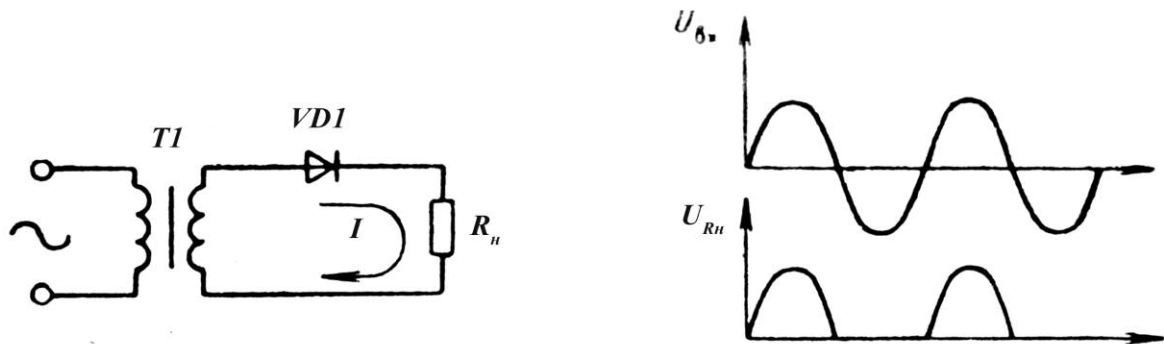


Рисунок 1.14 – Випрямляч напруги

Ппульсуюча постійна напруга, яка отримується на виході випрямляча, в більшості випадків негативно впливає на нормальну роботу електросхем.

Для зниження величини пульсації вихідної напруги на виході випрямляча встановлюють згладжувальні фільтри. Найпростішим фільтром є ємність, яка включається паралельно навантаженню випрямляча, що показано на рис. 1.15, а).

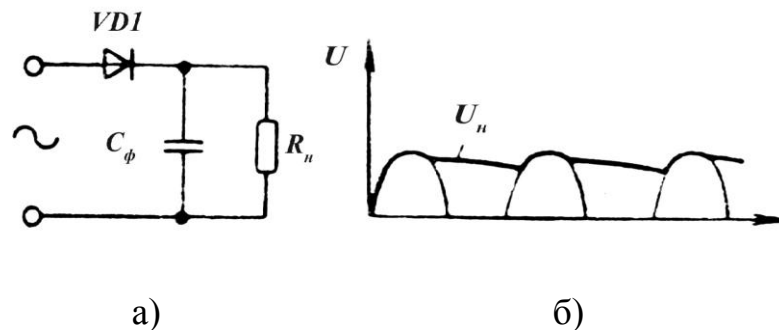


Рисунок 1.15 – Випрямляч напруги зі згладжувальним фільтром

Згладжуюча дія конденсатора полягає в наступному. Конденсатор заряджується весь час, поки напруга на виході випрямляча більша за напругу на навантаженні (рис. 1.15, б). Коли напруга на виході випрямляча починає зменшуватись, конденсатор розряджується, віддаючи накопичену енергію в навантаження. Якщо ємність конденсатора обрана доволі великою, то за час зменшення напруги на виході випрямляча, він не встигає достатньо сильно розрядитись і коливання напруги на навантаженні (пульсація напруги) буде відносно невеликою.

1.5.2 Стабілітрони

Напівпровідникові стабілітрони, які іноді називають опорними діодами, призначені для стабілізації напруги. При роботі стабілітрона використовується зворотна гілка вольт–амперної характеристики (стабілітрон підключають катодом до плюса, а діод катодом до мінуса). При підвищенні зворотної напруги відбувається пробій $p-n$ переходу, при якому зворотна гілка характеристики стабілітрона має крутий злам, обумовлений різким ростом струму. Цей злам відповідає напрузі стабілізації U_{cm} .

Стабілітрони використовуються для побудови стабілізаторів напруги. Найпростішим і найпоширенішим параметричним стабілізатором є стабілізатор, який використовує особливість вольт–амперної характеристики напівпровідникового стабілітрону. Він являє собою подільник напруги, який включає в себе баластний резистор R_b і стабілітрон $VD1$ (рис. 1.16).

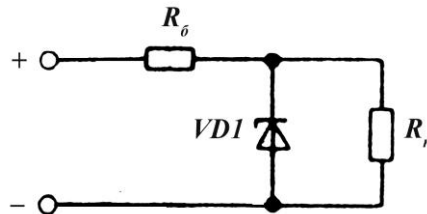


Рисунок 1.16 – Параметричний стабілізатор

При розрахунку параметрів елементів параметричного стабілізатора заздалегідь відомими величинами є: межі зміни напруги живлення; напруга стабілізації стабілітрона, яка дорівнює вихідній напрузі стабілізатора; мінімальний і максимальний струми стабілізації; динамічний опір стабілітрона.

Опір баластного резистора $R_b = (U_{ж} - U_{ст})/I_{рн}$, де $U_{ж}$ – номінальна величина напруги живлення, В; $U_{ст}$ – напруга стабілізації стабілітрона, В; $I_{рн}$ – номінальний робочий струм стабілітрона, мА.

1.5.3 Світлодіоди

Світлодіоди – це діоди, які випромінюють світло при протіканні через них прямого струму. Область спектрального випромінювання діодів має досить вузькі границі. Умовне графічне зображення світлодіодів і схема їх включення показані на рис.1.17.

Пряме падіння напруги на світлодіодах більше, ніж на звичайних діодах, і залежить від їх типу. Так для світлодіодів червоного світіння $U_d = 1,6 - 1,8$ В, для жовтогарячого і жовтого – $U_d = 2,0 - 2,2$ В. Яскравість світіння в широкому діапазоні пропорційна прямому струму світлодіода.

Світлодіоди використовуються як елементи індикації і виготовляються у вигляді окремих елементів, а також у вигляді семисегментних або матричних індикаторів.

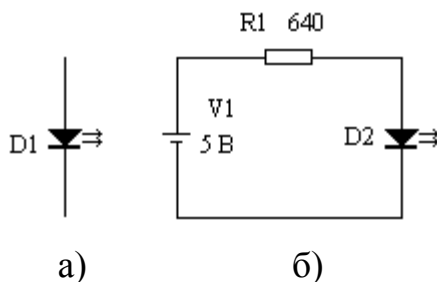


Рисунок 1.17 – Світлодіод

Як приклад, побудуємо індикатор напруги живлення 5 В (рис. 1.17,б). Використовуємо для цього світлодіод червоного світіння з $U_D = 1,8 \text{ В}$. Струм через світлодіод виберемо рівним 5 мА. Тоді величина резистора, що обмежує струм, буде дорівнювати $R1 = (U_{ж} - U_D) / I_D = (5 - 1,8) \text{ В} / 5 \text{ мА} = 640 \text{ Ом}$.

1.5.4 Фотодіоди

Фотодіод має структуру звичайного $p-n$ переходу. Зворотний струм фотодіода залежить від рівня освітленості. Фотодіоди містяться в металевому корпусі із прозорим вікном. Умовне графічне зображення фотодіода наведено на рис. 1.18. Фотодіод може включатися в схему із зовнішнім джерелом живлення (фотодіодний режим) і без нього (вентильний режим). У **вентильному режимі** фотодіод перетворює світлову енергію в електричну, генеруючи фото – е.р.с. під дією світла.

Фотодіоди у вентильному режимі придатні для одержання електричної енергії. Для цих цілей виготовляються спеціальні фотодіоди з великою площею $p-n$ переходу, які називаються сонячними елементами.



Рисунок 1.18– Умовне графічне зображення фотодіода

У **фотодіодному режимі** на фотодіод подається замикаюча напруга. При подачі на фотодіод замикаючої напруги фотострум практично не змінюється. При відсутності освітленості через фотодіод тече так званий темновий струм. Фотодіодний режим застосовують, коли потрібно одержати велику швидкість, тому що з ростом замикаючої напруги зменшується власна ємність $p-n$ переходу.

Фотодіоди широко застосовуються в різних датчиках, для побудови фоточутливих матриць, в оптронах.

1.5.5 Оптрони

Якщо в одному корпусі поєднати світлодіод і фоточутливий елемент, наприклад фотодіод, то можна одержати перетворення вхідного струму у вихідний з повним гальванічним розділом ланцюгів. Такі елементи називаються оптронами. Схемне зображення оптрона показано на рис. 1.19.

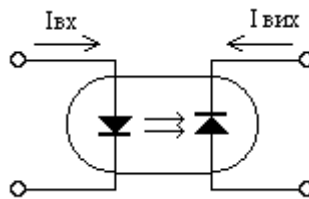


Рисунок 1.19 – Схемне зображення оптрона

Для одержання великого коефіцієнта корисної дії оптрони працюють в інфрачервоній області спектра. Найважливішим параметром оптрона є коефіцієнт передачі $\beta = I_{вих}/I_{вх}$, що у значній мірі визначається властивостями фоточутливого елемента. Оптрони застосовуються для передачі як цифрових, так і аналогових сигналів.

1.5.6 Біполярний транзистор

Транзистор – напівпровідниковий елемент із двома $p-n$ переходами і трьома виводами, що служать для підсилення та перемикання сигналу [5]. Розрізняють кремнієві та германієві транзистори. Вони бувають $n-p-n$ і $p-n-p$ типу. На рис. 1.20 показані їх умовні позначення та найпростіші моделі, що відображають структуру транзисторів.

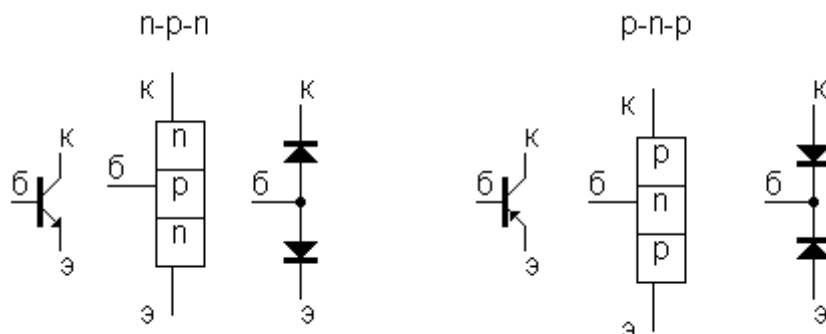


Рисунок 1.20 – Умовне графічне зображення транзисторів

Транзистор складається із двох протилежно включених $p-n$ переходів, які володіють одним загальним n - або p -шаром. Вивід транзистора,

пов'язаний з ним, називається базою (б). Два інших виводи називаються емітером (е) і колектором (к). Діодна модель транзистора пояснює структуру включення переходів транзистора. Хоча ця схема не характеризує повністю функції транзистора, вона дає можливість представити діючі в ньому прямі і зворотні напруги. Звичайно перехід б–е зміщений (включений) у прямому напрямку, а перехід б–к – у зворотному напрямку. В основу принципу дії транзистора покладений ефект впливу бази на проходження носіїв струму із області емітера в область колектора через область бази.

При підключенні джерела живлення (рис. 1.21) перехід база – емітер зміщується у прямому напрямку, що обумовлює протікання через нього струму бази. В транзисторі базова область має малу товщину, носії струму, попадаючи в неї, знаходяться під впливом відносно високої напруги колектора. Завдяки цьому вони сильно розганяються, запасавши енергію, що дозволяє їм подолати опір зміщеного колекторного переходу.

Основна частина носіїв з області емітера, пройшовши через область бази, переходить в область колектора і обумовлює величину колекторного струму. Якщо збільшити напругу на переході база – емітер, то він відкриється ще більше, а відповідно, збільшиться і число носіїв, які надходять з області емітера в область колектора і бази. Лише їх мала частина (біля 2%) рекомбінує в області бази, обумовлюючи базовий струм. Інакше кажучи, малий базовий струм керує більшим колекторним струмом.

Враховуючи, що напруга, яка докладається до бази, набагато менша напруги, яка докладається до колектору, а добуток сили струму на величину напруги є потужністю електричного струму, можна зробити висновок, що мала вхідна потужність керує більшою вихідною.

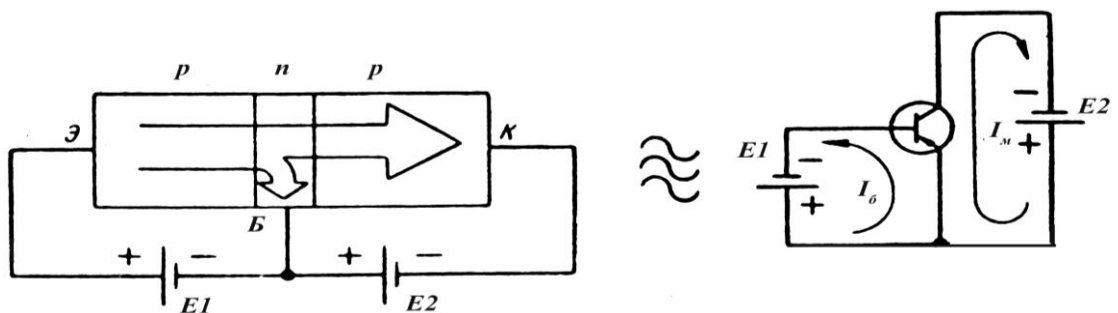


Рисунок 1.21 – До розподілення струму в транзисторі

Так як відношення зміни величини вихідного параметру до зміни величини вхідного параметра являє собою коефіцієнт підсилення, то можна сказати, що транзистор має коефіцієнт підсилення за потужністю, напругою або струмом.

Кожному з вказаних вище випадків відповідає своя схема увімкнення транзистора: із загальним емітером (ЗЕ), загальною базою (ЗБ), загальним колектором (ЗК).

Іншими важливими параметрами транзистора, які характеризують його електричні властивості, є вхідний та вихідний опір, величини яких використовуються при узгодженні каскадів підсилення.

На практиці найчастіше використовують схему увімкнення транзистору із 3Е (рис. 1.22, а). У цьому випадку вхідний сигнал подається між базою транзистору і загальною шиною, до якої підключений емітер транзистора. Вихідний сигнал знімається між колектором і загальною шиною. Ця схема забезпечує мале підсилення по струму та напрузі, але має відносно малий вхідний та відносно великий вихідний опір.

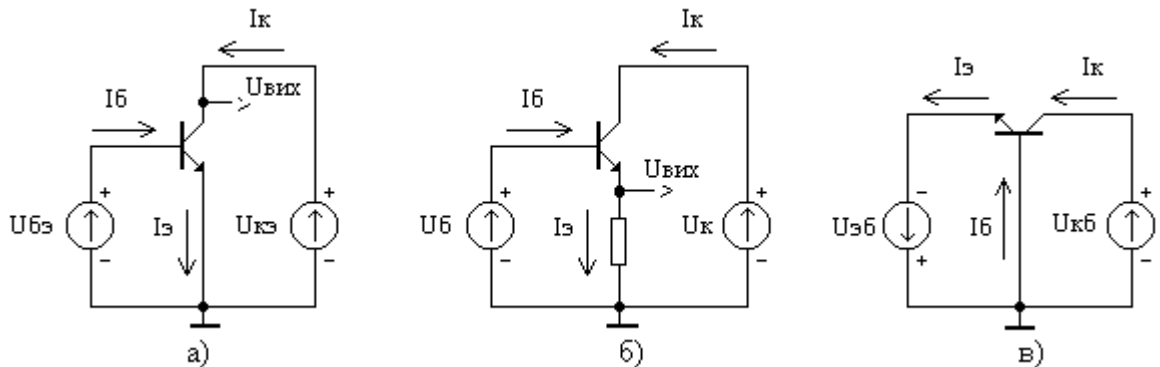


Рисунок 1.22 – Схеми включення $n-p-n$ транзистора

а) із загальним емітером; б) із загальним колектором; в) із загальною базою

В схемі із 3К (рис. 1.22, б) вхідний сигнал подається між базою транзистора та загальною шиною, до якої через опір навантаження підключений емітер транзистора, а знімається між емітером транзистора та загальною шиною.

Схема забезпечує великий коефіцієнт підсилення лише по струмові та має великий вхідний та малий вихідний опір. Цю схему називають емітерним повторювачем.

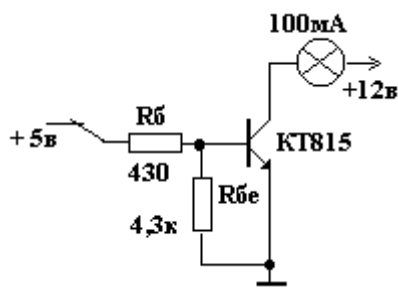
В схемі із 3Б вхідний сигнал подається між емітером транзистора та загальною шиною, до якої підключена база транзистора, а вихідний сигнал знімається між колектором та загальною шиною (рис. 1.22, в). Схема забезпечує підсилення лише по напрузі та має малий вхідний та великий вихідний опір.

Проаналізувавши схеми, показані на рис. 1.22 можна зробити висновок, що при будь-якому варіанті увімкнення транзистора керуючим переходом є перехід база – емітер, а керованим колом – коло, в яке увімкнені його емітер та колектор.

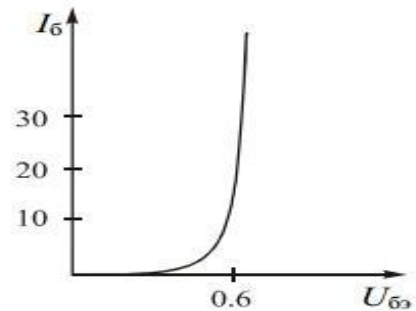
Схеми для $p-n-p$ транзистора будуть аналогічними. У цих схемах напрямки джерел напруг і струмів будуть протилежними, у порівнянні зі схемами для $n-p-n$ транзистора. Для будь-якої схеми включення транзистора, як для $n-p-n$ так і $p-n-p$ типу, справедливе співвідношення для струмів $I_e = I_б + I_к$. Підсилювальні властивості транзистора засновані на тому, що малими струмами бази можна управляти відносно великими струмами колектора. При цьому струм колектора $I_к$ є кратним базовому струму $I_б$. Відношення $\beta = I_к/I_б$ називається статичним коефіцієнтом

підсилення по струму. Коефіцієнт підсилення по струму транзистора набагато більше 1.

Окрім вказаних вище підсилювальних каскадів, транзистори широко використовуються в так званих ключових каскадах (рис. 1.23, а).



а)



б)

Рисунок 1.23 – Ключовий каскад

В них транзистор застосовують в якості електронного ключа, тобто елементу, який або пропускає електричний струм, або не пропускає його, комутуючи тим самим струм в навантаженні. Для цього обираються такі параметри базового кола, при яких базовий струм або відсутній (транзистор замкнутий, опір колектор – емітер дуже великий, струм через транзистор не проходить), або має таку величину, при якій транзистор відкритий, опір колектор – емітер малий, струм через транзистор має величину, яка обумовлюється лише напругою колекторного живлення та опору навантаження.

Основний сенс такого режиму в тому, що малий струм бази управляє великим струмом колектора, який у кілька десятків разів більше струму бази.

Практично для всіх сучасних транзисторів під час роботи в ключовому режимі коефіцієнт β не менше 10...20. Падіння напруги на переході база – емітер $U_{бэ} = 0,6 \text{ В}$. (див. вхідну характеристику транзистора на рис. 1.23, б). При таких даних струм в базі повинен бути $I_b = I_k / \beta = 100 / 10 = 10 \text{ (мА)}$. Напруга на базовому резисторі R_b становитиме (за відрахуванням напруги на переході база – емітер) $5 \text{ В} - U_{бэ} = 5 \text{ В} - 0,6 = 4,4 \text{ В}$. Згідно закону Ома: $R = U / I = 4,4 \text{ В} / 0,01 \text{ А} = 440 \text{ Ом}$. Зі стандартного ряду вибираємо резистор опором 430 Ом . Резистор $R_{бе}$ призначений, щоб надійно закрити транзистор в той момент, коли кнопка розімкнута. Справа в тому, що якщо база буде «висіти в повітрі», вплив усіляких перешкод на неї просто гарантовано. Зазвичай його приймають рівним десяти R_b . Згідно практичним даними його величина повинна бути 5...10 КОм.

Ключовий каскад можна підключати безпосередньо до виходів цифрових мікросхем та мікроконтролерів [6].

1.5.7 Польові транзистори

Польовими (уніполярними) транзисторами називаються напівпровідникові елементи, які на відміну від звичайних біполярних транзисторів управляються електричним полем, тобто практично без витрат потужності керуючого сигналу. Якщо в біполярних транзисторах носіями струму являються і електрони, і дірки, то в польових транзисторах – або електрони, або дірки. Існують дві великі групи польових транзисторів:

- польові транзистори з керуючим $p-n$ переходом (JFET – Junction Field Effect Transistor), у яких ізоляція каналу від джерела керуючої напруги забезпечується назад зміщеним $p-n$ переходом;

- польові транзистори з МОН (метал – оксид – напівпровідник) або МДН (метал – діелектрик – напівпровідник) структурою. Закордонне позначення MOSFET (або скорочено MOS). У цих транзисторів ізоляція каналу від керуючого електрода забезпечується за допомогою діелектрика (двоокису кремнію).

Всі типи транзисторів можуть бути як n -канальні, так і p -канальні. Класифікація та умовні графічні зображення транзисторів наведені на рис. 1.24.

Затвор Z (G – gate) – керуючий електрод. Він управляє величиною опору між **стоком** C (D – drain) і **витоком** B (S – source). Керуючою напругою є напруга $U_{зв}$. Більшість польових транзисторів є симетричними, тобто їх властивості не змінюються, якщо електроди C і B поміняти місцями.

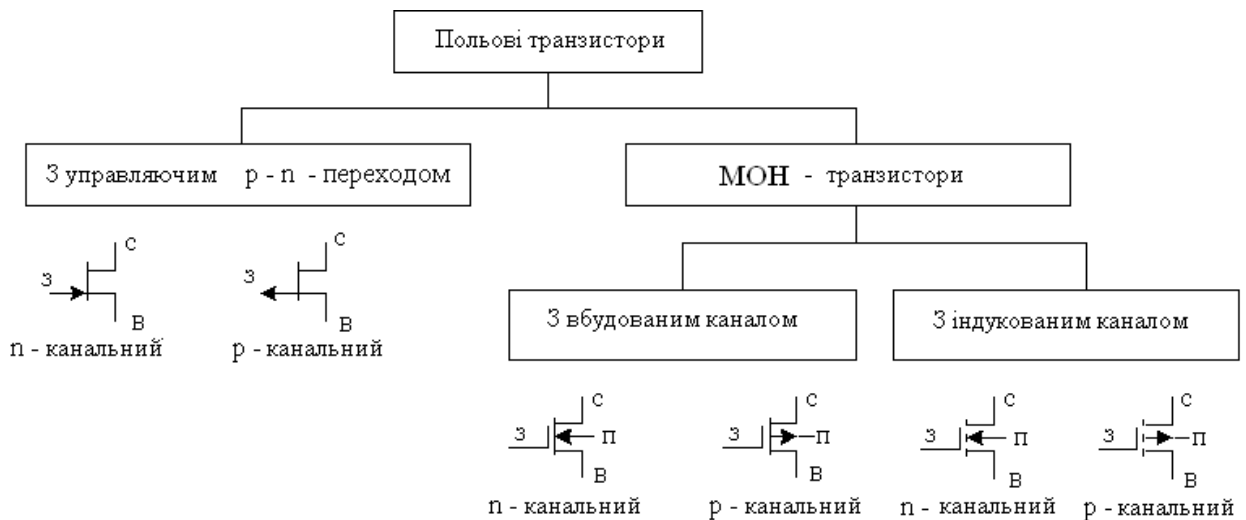


Рисунок 1.24 – Класифікація та умовні графічні зображення транзисторів

Польовий транзистор з керуючим $p-n$ переходом являє собою пластину з напівпровідникового матеріалу, що має електропровідність певного типу, від якого зроблені два виводи – стік-виток (рис. 1.25).

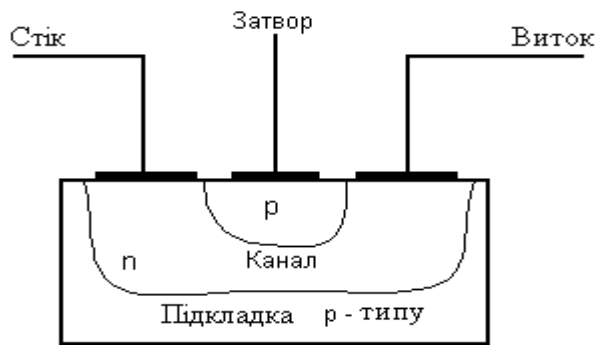


Рисунок 1.25 – Польовий транзистор з керуючим p - n переходом

Уздовж пластини виконано електричний перехід (p - n перехід чи бар'єр Шотткі), від якого зроблено третій вивід–затвор. При подачі на стік польового транзистора з каналом n -типу позитивної відносно джерела напруги, канал проводить електричний струм, величина якого залежить від величини докладеної напруги та опору каналу. Якщо при цьому подати на затвор негативну відносно джерела напругу, то канал «звужиться», його переріз зменшиться, опір збільшиться, а відповідно, зменшиться струм стоку.

На відміну від польових транзисторів з p - n переходом МОН–транзистори (рис. 1.26) області затвору не мають. В якості затвору використовується металева плівка, напилена на поверхню ізолюючого шару окису. Таким чином, затвор МОН – транзистору являє собою конденсатор, одна з обкладинок якого підключена до виводу затвору транзистора, а друга утворена «поверхнею» каналу.

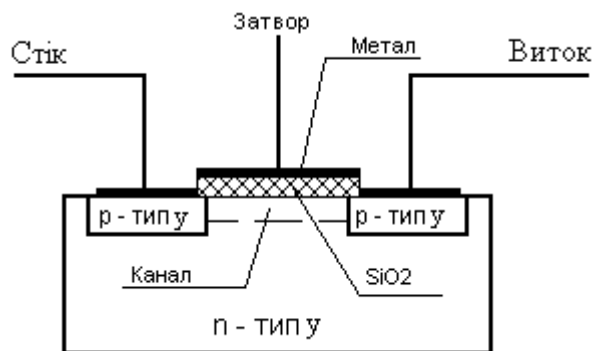


Рисунок 1.26 – МОН– транзистор

Основною перевагою польових транзисторів є надзвичайно великий вхідний опір. Зміщений в зворотному напрямку p - n перехід польового транзистору має опір, який вимірюється мегомами, в зв'язку з чим струм витоку канал – затвор має величину в декілька десятків наноампер.

Вихідний опір МОН–транзистора має ще більшу величину, так як входом транзистора в цьому випадку є конденсатор відносно невеликої ємності.

Як і біполярні транзистори, польові транзистори по електричному стану можуть знаходитись в трьох областях: відсічення (транзистор закритий, вихідні струми мінімальні, напруга на колекторі або стокові

максимальна), активної (вихідний струм транзистора і напруга на транзисторі залежать від величини сигналів керування) і насичення (транзистор повністю відкритий, вихідний струм максимальний, напруга на транзисторі мінімальна).

Польові транзистори мають три основні схеми увімкнення: із загальним витоком (ЗВ), загальним затвором (ЗЗ) і загальним стоком (ЗС). Характеристики цих схем аналогічні наведеним для біполярних транзисторів, але слід враховувати, що величини вхідних та вихідних опорів каскадів на польових транзисторах мінімум на порядок вищі, ніж у каскадів на біполярних транзисторах. Тому застосування польових транзисторів дає суттєвий вииграш в підсиленні потужності [4, 6].

1.6 Інтегральні схеми

Ключовою подією, що обумовила прихід інтеграції в електроніку, стала пропозиція американського інженера Дж. Кілбі з фірми «Texas Instruments» отримувати еквівалентні елементи для всієї схеми, такі як резистори, конденсатори, транзистори і діоди в монолітному кристалі чистого кремнію. Першу інтегральну напівпровідникову схему Кілбі створив в літку 1958 року. А вже в 1961 році фірма «Fairchild Semiconductor Corporation» випустила перші серійні мікросхеми. У стислі терміни в інтегральному виконанні були створені різні типи підсилювачів. У 1962 році фірма RCA розробила інтегральні мікросхеми пам'яті. Надзвичайно важливою подією на цьому шляху стало створення в 1971 році американською фірмою «Інтел» єдиної інтегральної схеми для виконання арифметичних і логічних операцій – мікропроцесора [7, 8].

Інтегральна мікросхема (ІС) – це мініатюрний електронний блок, що містить в загальному корпусі тисячі елементів (транзисторів, діодів, резисторів, конденсаторів) в кристалі. Призначення – генерування і оброблення сигналів, запис, збереження і відтворення інформації. ІС випускаються серіями. Серія – це група мікросхем, що мають єдине конструктивно – технологічне виконання і призначені для спільного застосування. При стиковці цифрових приладів, побудованих на мікросхемах різного типу, використовують перетворювачі (транслятори) рівнів логічних сигналів.

Найбільшого поширення набули ІС, у яких всі елементи і міжелементні з'єднання виконані в об'ємі і на поверхні напівпровідника. Їх називають напівпровідниковими. Для виготовлення напівпровідникових мікросхем використовують кремнієві монокристалічні пластини. Елементи мікросхеми формують в напівпровідниковій пластині методами, відомими як селективна дифузія, епітаксії та ін. Міжелементні з'єднання виконують напиленням вузьких провідних доріжок алюмінію.

Напівпровідникові ІС класифікують за функціональним призначенням (аналогові і цифрові), ступенем інтеграції (малі, середні, великі і надвеликі), функціональною гнучкістю (постійні і програмовані функції), частотою

(низькочастотні, високочастотні і надвисокочастотні), потужності розсіювання (потужні, малопотужні, мікропотужні, нановатні). Специфічною для напівпровідникових ІС є класифікація за типом базового елементу – транзистора. Їх поділяють на два підвиди: на біполярних транзисторах (**ТТЛ** – транзисторно – транзисторна логіка) і на уніполярних (польових) транзисторах (**КМОН** – логіка на комплементарних транзисторах зі структурою метал – оксид – напівпровідник). Вдосконалена транзисторно – транзисторна логіка – **ТТЛШ** використовує біполярні транзистори з діодом Шотткі, який включено між його базою і колектором, що не дозволяє транзистору увійти в режим насичення. В діодах Шотткі використовується перехід метал – напівпровідник, на відміну від звичайних діодів, де використовується $p-n$ перехід. За допомогою цих класів ІС можна задовольнити всі вимоги, які виникають при побудові цифрових схем та пристроїв, за виключенням надшвидкодіючих, де використовуються елементи з емітерними зв'язками – **ЕЗЛ** логіка.

Для кожного ряду мікросхем визначені допустимі значення високого та низького рівнів напруги (рис. 1.27). Такі широкі діапазони вибрані для того, щоб виробник мікросхем мав у своєму розпорядженні певний допуск, в межах якого параметри схеми можуть відхилятися за рахунок зміни температури, навантаження, напруги живлення, а також під дією шуму.

До основних параметрів ІС відносяться рівні вхідних і вихідних струмів та напруги, значення потужності споживання, середній час затримки сигналу, який визначає швидкодію схеми, навантажувальна здатність, завадостійкість, надійність та інші. Кількісно навантажувальна можливість оцінюється коефіцієнтом розгалуження по виходу, тобто кількістю одиночних навантажень, які можливо одночасно підключати до виходу мікросхеми. Найвища швидкодія у елементів ЕЗЛ та ТТЛШ, що дозволяє їм працювати з тактовою частотою 10-100 МГц (ЕЗЛ) та 5-10 МГц (ТТЛШ). Мінімальну потужність споживання мають елементи КМОН – 0,003-0,02 мВт / ел. Ці елементи мають саму низьку енергію переключення.

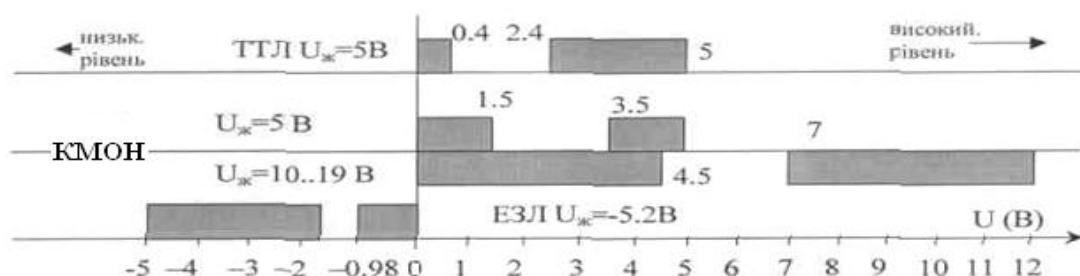


Рисунок 1.27 – Значення рівнів напруги

Умовні графічні позначення інтегральних мікросхем будують на основі прямокутника. Вони можуть вміщувати в собі основне та одно чи два додаткових поля в залежності від типу мікросхеми. Функціональне призначення елемента показують у верхній частині основного поля умовного графічного позначення. У додаткових полях подається інформація про функціональне призначення виводів.

2 ПРЕДСТАВЛЕННЯ ДАНИХ В КОМП'ЮТЕРІ

При комп'ютерній обробці інформації доводиться мати справу з текстовими, графічними, звуковими, числовими та іншими даними. Вся інформація незалежно від її характеру подається у чисельному вигляді [9-15].

2.1 Системи числення

2.1.1 Основні визначення. Позиційні системи числення.

Системою числення називається сукупність цифрових знаків і правил їх з'єднання для позначення (запису) чисел. Тобто, в загальному випадку це спеціальна мова, алфавітом якої є символи, що називаються цифрами, а синтаксисом – правила, що дозволяють однозначно сформулювати запис чисел.

Існує багато способів запису чисел цифровими знаками, тобто багато систем числення. В цифрових системах найбільше розповсюдження отримали однорідні позиційні системи числення, в яких будь-яке число може бути представлено поліномом :

$$A = a_{n-1}p^{n-1} + \dots + a_0p^0 + a_{-1}p^{-1} + \dots + a_{-m}p^{-m} = \sum_{i=-m}^{n-1} a_i p^i, \quad (2.1)$$

де p - основа системи числення, $p \in \{2, 3, \dots\}$;

a_i – цифра i -го розряду, $a_i = \overline{0, 1, 2, \dots, (p-1)}$;

n – кількість цифр (розрядів) цілої частини числа;

m – кількість цифр (розрядів) дробової частини числа.

Серед однорідних позиційних систем числення слід виділити:

– **двійкова система числення (бінарна, англ. binary):** $p = 2$; $a_i = 0, 1$.

Загальна формула для запису числа у цій системі числення матиме вигляд:

$$[A]_2 = a_{n-1} \cdot 2^{n-1} + \dots + a_0 \cdot 2^0 + a_{-1} \cdot 2^{-1} + \dots + a_{-m} \cdot 2^{-m}.$$

Ваги розрядів, рухаючись вліво від коми, дорівнюють відповідно 1, 2, 4, 8, 16, ..., а вправо від коми – 1/2, 1/4, 1/8 ...

Наприклад, двійкове число $110010,101_2$ відповідає такому представленню у десятковій системі числення:

$$110010,101_2 = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 50,625_{10}.$$

– **вісімкова система числення (октавна, англ. octal):**

$$p = 8; \quad a_i = 0, 1, 2, \dots 7.$$

Загальна формула для запису числа у цій системі числення матиме вигляд:

$$[A]_8 = a_{n-1} \cdot 8^{n-1} + \dots + a_0 \cdot 8^0 + a_{-1} \cdot 8^{-1} + \dots + a_{-m} \cdot 8^{-m}.$$

Ваги розрядів, рухаючись вліво від коми, дорівнюють відповідно 1, 8, 64, 256, ..., а вправо від коми – 1/8, 1/64, 1/256

Наприклад, вісімкове число 317,25₈ відповідає такому представленню у десятковій системі числення:

$$317,25_8 = 3 \cdot 8^2 + 1 \cdot 8^1 + 7 \cdot 8^0 + 2 \cdot 8^{-1} + 5 \cdot 8^{-2} = \left(207 \frac{21}{64} \right)_{10}.$$

– **десяткова система числення (децимальна, англ. decimal):**

$$p = 10; \quad a_i = 0, 1, 2, \dots 9.$$

Загальна формула для запису числа у цій системі числення буде така:

$$[A]_{10} = a_{n-1} \cdot 10^{n-1} + \dots + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + \dots + a_{-m} \cdot 10^{-m}.$$

Ваги розрядів, рухаючись вліво від коми, дорівнюють відповідно 1, 10, 100, ..., а вправо від коми – 1/10, 1/100,

– **шістнадцяткова система числення (гексагональна, англ. hexadecimal):**

$$p = 16; \quad a_i = 0, 1, 2, \dots 9, A, B, C, D, E, F.$$

Загальна формула для запису числа, при цьому, матиме вигляд:

$$[A]_{16} = a_{n-1} \cdot 16^{n-1} + \dots + a_0 \cdot 16^0 + a_{-1} \cdot 16^{-1} + \dots + a_{-m} \cdot 16^{-m}.$$

Ваги розрядів, рухаючись вліво від коми, дорівнюють відповідно 1, 16, 256, ..., а вправо від коми – 1/16, 1/256

Наприклад, шістнадцяткове число A7B,C8₁₆ відповідає такому представленню у десятковій системі числення:

$$A7B,C8_{16} = 10 \cdot 16^2 + 7 \cdot 16^1 + 11 \cdot 16^0 + 12 \cdot 16^{-1} + 8 \cdot 16^{-2} = \left(2683 \frac{25}{32} \right)_{10}.$$

– **двійково-десяткова система числення** (англ. **Binary Coded Decimal - BCD**) з вагами 8-4-2-1. У ній кожна цифра десятичного числа кодується двійковою тетрадою (чотирма двійковими розрядами).

В табл. 2.1 наведені представлення шістнадцяти перших десятичних цифр у вказаних системах числення.

Операції над десятичними цифрами виконуються за допомогою доповненої двійкової арифметики. Так, при додаванні двох чисел в коді прямого заміщення 8421 необхідно додати корегуючий елемент $6_{10} = 0110_2$ до кожної тетради, в якій в процесі додавання отримана цифра > 9 або виникло перенесення в наступну тетраду.

Таблиця 2.1

Системи числення				
$p = 10$	$p = 2$	$p = 8$	$p = 16$	8-4-2-1
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	0001 0000
11	1011	13	B	0001 0001
12	1100	14	C	0001 0010
13	1101	15	D	0001 0011
14	1110	16	E	0001 0100
15	1111	17	F	0001 0101

При відніманні чисел в коді 8421 корекція зводиться до віднімання $6_{10} = 0110_2$ з кожної тетради різниці, яка потребує займу. **Наприклад:**

$ \begin{array}{r} 38_{10} = 0011\ 1000 \\ + \\ 16_{10} = 0001\ 0110 \\ \hline 0100\ 1110 \\ \text{Корекція} + 0000\ 0110 \\ \hline 54_{10} = 0101\ 0100 \end{array} $	$ \begin{array}{r} \leftarrow 1 \\ 29_{10} = 0010\ 1001 \\ + \\ 58_{10} = 0101\ 1000 \\ \hline 1000\ 0001 \\ \text{Корекція} + 0000\ 0110 \\ \hline 87_{10} = 1000\ 0111 \end{array} $	$ \begin{array}{r} 1 \rightarrow \\ 63_{10} = 0100\ 0011 \\ - \\ 27_{10} = 0001\ 0111 \\ \hline 0011\ 1100 \\ \text{Корекція} - 0000\ 0110 \\ \hline 36_{10} = 0011\ 0110 \end{array} $
---	--	---

2.1.2 Переведення чисел із однієї позиційної системи числення в іншу

Щоб перевести ціле число з однієї позиційної системи числення в іншу, необхідно початкове число послідовно ділити на основу нової системи числення, записану в початковій системі числення, до отримання частки, яка дорівнює нулю. Число у новій системі числення записується із остач від ділення, починаючи з останньої.

Приклад. Перевести десяткове число $A_{10} = 345$ у двійкову, вісімкову і шістнадцяткову системи числення.

Виконаємо дії згідно з наведеним вище алгоритмом і отримаємо:

$$\begin{array}{r}
 345 \overline{) 2} \\
 \underline{344} \quad 172 \overline{) 2} \\
 \underline{172} \quad 86 \overline{) 2} \\
 \underline{86} \quad 43 \overline{) 2} \\
 \underline{42} \quad 21 \overline{) 2} \\
 \underline{20} \quad 10 \overline{) 2} \\
 \underline{10} \quad 5 \overline{) 2} \\
 \underline{4} \quad 2 \overline{) 2} \\
 \underline{2} \quad a_8 = 1 \\
 \underline{0} \quad a_7 = 0
 \end{array}$$

$$345_{10} = 101011001_2.$$

$$\begin{array}{r}
 345 \overline{) 8} \\
 \underline{344} \quad 43 \overline{) 8} \\
 \underline{40} \quad a_2 = 5 \\
 \underline{0} \quad a_1 = 3
 \end{array}$$

$$345_{10} = 531_8.$$

$$\begin{array}{r}
 345 \overline{) 16} \\
 \underline{336} \quad 21 \overline{) 16} \\
 \underline{16} \quad a_2 = 1 \\
 \underline{0} \quad a_1 = 5
 \end{array}$$

$$345_{10} = 159_{16}.$$

Щоб перевести правильний дріб із однієї позиційної системи числення в іншу, необхідно початкове число послідовно множити на основу нової системи числення, записану в початковій системі числення. Число у новій системі числення записується із цілих частин від множення, починаючи із першої. Процедурі необхідно продовжувати до отримання заданої точності.

Як видно з останнього виразу, на відміну від цілих чисел, точне переведення можливе не для всіх правильних дробів. Похибка при переведенні визначається одиницею молодшого розряду.

Приклад. Перевести правильний дріб $U_{10} = 0,357$ із десяткової системи числення у двійкову, вісімкову і шістнадцяткову системи числення.

Процедура переведення згідно із розглянутим алгоритмом представлена у табл. 2.2. Із врахуванням результатів табл. 2.2, отримаємо:

$$0,357_{10} = 0,0101101101_2 = 0,2666_8 = 0,544_{16}.$$

Для того щоб перевести число з початкової системи числення в нову систему, основа якої кратна основі початкової системи, досить кожну цифру числа, що переводиться, записати за допомогою m цифр у новій системі числення, якщо основа початкової системи числення більша від основи нової системи числення. У іншому разі кожні m цифр початкового числа необхідно записати за допомогою однієї цифри в новій системі числення,

починаючи для цілих чисел з молодшого розряду і зі старшого розряду для правильних дробів.

Таблиця 2.2

Основа системи числення					
$p = 2$		$p = 8$		$p = 16$	
Ціла частина	Дробова частина	Ціла частина	Дробова частина	Ціла частина	Дробова частина
0	357	0	357	0	357
	$\times 2$		$\times 8$		$\times 16$
0	714	2	856	5	712
	$\times 2$		$\times 8$		$\times 16$
1	428	6	848	4	272
	$\times 2$		$\times 8$		$\times 16$
0	856	6	784	4	352
	$\times 2$		$\times 8$		
1	712	6	272		
	$\times 2$				
1	424				
	$\times 2$				
0	848				
	$\times 2$				
1	696				
	$\times 2$				
1	392				
	$\times 2$				
0	784				
	$\times 2$				
1	568				

Наприклад, при переведенні вісімкового числа 316_8 у двійкову систему числення досить кожну цифру вісімкового числа записати у вигляді двійкової тріади, оскільки $8 = 2^3$:

$$316_8 = \underbrace{011}_3 \underbrace{001}_1 \underbrace{110}_6 = 011001110_2.$$

При переведенні, наприклад, двійкового числа в шіснадцаткову систему числення досить кожну тетраду початкового числа записати у вигляді шіснадцяткової цифри:

$$10010111110001011101_2 = \underbrace{1001}_9 \underbrace{0111}_7 \underbrace{1100}_C \underbrace{0101}_5 \underbrace{1101}_D = 97C5D_{16}.$$

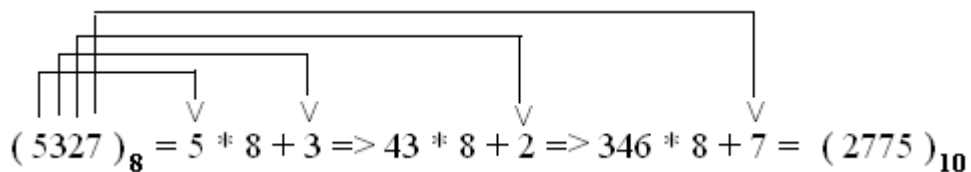
Для перетворення числа з будь-якої системи числення на десяткову достатньо обчислити значення відповідного поліному, підставивши до нього

десятькове значення розрядів і основ системи числення. Обчислення зручно виконувати за схемою Горнера, що основана на представленні поліному (2.1) у вигляді:

$$(A)_p = (((a_{n-1} * p + a_{n-2}) * p + a_{n-3}) * p + \dots + a_1) * p + a_0, \quad (2.2)$$

тобто цифра a_{n-1} старшого розряду числа, що переводиться, помножується на основу p початкової системи числення і результат додається до наступної цифри, потім процес повторюється до найменшого розряду.

Для $(A)_8 = 5327$ в десятковій системі числення, отримаємо:



$$(5327)_8 = 5 * 8 + 3 \Rightarrow 43 * 8 + 2 \Rightarrow 346 * 8 + 7 = (2775)_{10}$$

2.2 Текстові дані

При зберіганні в комп'ютері будь-який текст (документ, стаття, книга) розглядається як лінійна послідовність символів. Причому проміжок між окремими словами – пробіл, перехід на наступний рядок, перехід на наступну сторінку – також можуть розглядатися як спеціальні символи. Кожному символу з цієї послідовності ставиться у відповідність конкретний двійковий код, що складається рівно з восьми двійкових розрядів. Таким чином, код кожного символу тексту займає *рівно один байт* пам'яті.

Алфавіт, який використовується для представлення текстів на природній мові, повинен містити як мінімум 52 латинські букви (рядкові і прописні), десяткові цифри, розділові знаки, математичні знаки, спеціальні знаки і т. д. Всього приблизно 150 символів. Виходячи з теоретичних міркувань, при рівномірному алфавітному двійковому кодуванні для подання будь-якого знаку вихідного алфавіту потрібно не менше $\log_2 150 = 7,2$ бітів, тобто восьми двійкових цифр.

Списки всіх використовуваних при записі текстів символів і однозначно відповідних їм двійкових кодів, утворюють так звані **кодові таблиці**. У практиці програмування застосовуються різні кодові таблиці. Однією з найбільш часто використовуваних є кодова таблиця **ASCII** (American Standart Code for Information Interchange – американський стандартний код для обміну інформацією). У цій таблиці зафіксовані коди для 128 різних символів. Їх список і відповідні їм восьмирозрядні (тобто складаються з восьми двійкових цифр, розрядів) двійкові коди утворюють основну (базову) кодову таблицю ASCII. Але один байт може містити 256 різних восьмирозрядних кодів. Це означає, що в стандарті ASCII задіяна тільки половина можливостей 8-бітного кодування. Є різні розширення

основної кодової таблиці ASCII, в яких задаються коди ще для 128 символів, в тому числі і для символів різних національних алфавітів.

Фрагмент одного з розширень кодової таблиці ASCII, що включає кирилицю, наведено в табл. 2.3.

Таблиця 2.3

Символ	Двійковий код	Символ	Двійковий код	Символ	Двійковий код
А	10000000	Л	10001011	Ц	10010110
Б	10000001	М	10001100	Ч	10010111
В	10000010	Н	10001101	Ш	10011000
Г	10000011	О	10001110	Щ	10011001
Д	10000100	П	10001111	Ї	10011010
Е	10000101	Р	10010000	І	10011011
Ж	10000110	С	10010001	Ь	10011100
З	10000111	Т	10010010	Є	10011101
И	10001000	У	10010011	Ю	10011110
Й	10001001	Ф	10010100	Я	10011111
К	10001010	Х	10010101	’	00101100

Як приклад кодування отримаємо машинний код тексту, що складається з одного слова «**КОМП’ЮТЕР**». Цей текст складається з 9 символів, отже, для його зберігання потрібно 9 байтів пам'яті (табл. 2.4).

Таблиця 2.4 - Машинний код тексту «КОМП’ЮТЕР»

1	2	3	4	5	6	7	8	9
К	О	М	П	’	Ю	Т	Е	Р
10001010	10001110	10001100	10001111	00101100	10001110	10010010	10000101	10010000

Зверніть увагу на те, що в табл. 2.3 наведені коди великих літер. Малі букви мають інші коди. Наприклад, код літери «а» має вигляд 1010 0000₂, в той час як код літери «А» - 1000 0000₂. Не випадково, що розглядається слово записане саме в такому вигляді - машинний код слова «КОМП'ЮТЕР» відрізняється від машинного коду слова «комп'ютер».

Якщо врахувати всі можливі букви, які входять в національні алфавіти різних країн, всі можливі символи, що зустрічаються в математичних та інших спеціальних текстах, то їх загальна кількість виявиться значно більше 256 символів, кодування яких забезпечує один байт. Тому було розроблено кілька десятків різних кодових таблиць. При цьому в різних кодових таблицях один і той же код може відповідати різним символам. Враховуючи недостатність можливостей даних кодових таблиць, останнім часом все ширше використовується кодова таблиця з назвою **UNICODE** (UNiversal CODE - універсальний код), в якій для коду одного символу відводиться два байти, а не один, як у розглянутих раніше таблицях. За допомогою двох байтів, тобто 16 бітів, можна закодувати вже $2^{16} = 65\,536$ різних символів.

За деякими оцінками, в усьому світі в даний час використовується близько 200 000 різних символів. Такої кількості кодів цілком достатньо, щоб представити більшість із найбільш часто вживаних у текстах символів.

Якщо уважно проаналізувати дії, які доводиться виконувати над довільними текстами, можна помітити, що будь-які їхні перетворення зводяться до заміни одного символу тексту іншим, видалення символу з тексту і вставці нового символу в обране місце тексту. У таких випадках, коли описані вище дії з перетворення текстів необхідно виконати не над одним символом, а над деякою групою символів, відповідні операції можна виконати циклічно кілька разів. Якщо включити в алфавіт спеціальний «пустий» символ (не плутати з пробілом), який можна вважати порожнім текстом (тобто текстом, що не містить жодного символу), то виявиться, що будь-які перетворення текстів зводяться до однієї - єдиної операції: заміни одного символу алфавіту іншим. Саме тому, для визначення символу, який необхідно замінити чи видалити, або ж місця вставки нового символу, потрібно ще вміти порівнювати між собою будь-які два символи алфавіту на збіг чи розбіжність.

Отже, основні елементарні дії, які повинен «уміти» виконувати комп'ютер над текстовими даними, — це порівняння кодів двох символів текста і заміна одного коду іншим.

2.3 Графічні дані

Для обговорення загальних принципів кодування графічних даних розглянемо зображення на екрані монітора. Це зображення складається з деякої кількості горизонтальних ліній - рядків. А кожен рядок, в свою чергу, складається з елементарних найдрібніших одиниць зображення - точок, які прийнято називати **пікселями** (від pixel – PICTURE'S of ELEMENTS - елемент картинки). Весь масив елементарних одиниць зображення називають

растром (від лат. Rastrum - борона). Ступінь чіткості зображення залежить від кількості рядків на весь екран і кількості точок в рядку, які представляють роздільну здатність екрану, або просто **дозвіл**. Чим більше рядків і точок, тим чіткіше і краще зображення. Досить хорошим вважається, наприклад, дозвіл 800 x 600, тобто 800 точок на рядок і 600 рядків на екран. Рядки, з яких складається зображення, можна переглядати зверху вниз один за одним, як би склавши з них одну суцільну лінію. Після повного перегляду першого рядка проглядається другий, за нею третій, потім четвертий і т. д. до останнього рядка екрану. Цей процес дуже схожий на прийнятий у більшості країн світу спосіб читання текстів, коли рядки проглядаються один за одним зліва направо і зверху вниз. Такий спосіб роботи з зображенням називається рядковою розгорткою, або **скануванням** (від scan - побіжно переглядати, розгортати зображення). Так як кожен з рядків являє собою послідовність пікселів, то все зображення, витягнуте в лінію, також можна вважати лінійною послідовністю елементарних точок. У розглянутому випадку ця послідовність складається з $800 \times 600 = 480\,000$ пікселів.

Спочатку обговоримо принципи кодування монохромного зображення, тобто зображення, що складається з будь-яких двох контрастних кольорів – чорного і білого. Тоді кожен піксель зображення може мати або чорний, або білий колір. Поставивши у відповідність чорному кольору двійковий код 0, а білому – код 1 (або навпаки), ми зможемо закодувати в одному біті стан одного пікселя монохромного зображення. Так як 1 байт складається з 8 бітів, то на сходинку, що складається з 800 пікселів, буде потрібно 100 байтів пам'яті, а на все зображення – 60 000 байтів.

Однак отримане таким чином зображення буде надмірно контрастним. Реальне чорно-біле зображення складається не тільки з білого і чорного кольорів. У нього входить безліч різних проміжних відтінків – сірий, світло-сірий, темно-сірий і т. д. Якщо крім білого і чорного кольорів використовувати тільки дві додаткові градації, скажімо, світло-сірий і темно-сірий кольори, то щоб закодувати колірний стан одного пікселя, буде потрібно вже два біти: чорний колір – 00_2 , темно-сірий – 01_2 , світло-сірий – 10_2 , білий – 11_2 (рис. 2.1).

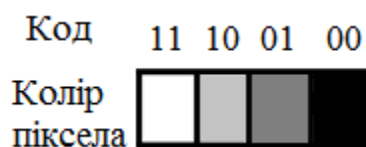


Рисунок 2.1 - Варіант кодування проміжних відтінків монохромного зображення

Досить реалістичним монохромне зображення вважається при кодуванні стану одного пікселя за допомогою восьми бітів, тобто одного байта, що дозволяє передавати 256 різних відтінків сірого кольору, від

повністю білого до повністю чорного. У цьому випадку для передачі всього растру з 800 x 600 пікселів буде потрібно вже не 60 000, а 480 000 байтів.

Кольорове зображення може формуватися різними способами. Один з них – метод RGB (Red, Green, Blue – червоний, зелений, синій), який опирається на те, що око людини сприймає всі кольори як суму трьох основних – червоного, зеленого і синього. Наприклад, бузковий колір – це сума червоного і синього, жовтий колір – сума червоного і зеленого і т. д. Знову спрощуючи ситуацію, будемо вважати, що для кодування кожного з кольорів досить одного біта, при цьому нуль в біті означає, що в сумарному кольорі даний основний відсутній, а одиниця – присутній. Отже, для кодування одного кольорового пікселя буде потрібно три біти – по одному на кожен колір. Хай перший біт відповідає червоному кольору, другий – зеленому, а третій – синьому. Тоді код 101_2 позначає бузковий колір (червоний є, зеленого немає, синій є), а код 110_2 – жовтий колір (червоний є, зелений є, синього немає). За такої схеми кодування кожен піксель може бути пофарбований в один з восьми можливих кольорів. Якщо кожен з кольорів кодувати за допомогою одного байта, як це прийнято для реалістичного монохромного зображення, то з'явиться можливість передавати по 256 відтінків кожного з основних кольорів. Всього в цьому випадку забезпечується передача $256 \times 256 \times 256 = 16\,777\,216$ різних кольорів, що досить близько до реальної чутливості людського ока. Таким чином, при даній схемі кодування кольору на зображення одного пікселя потрібно три байта, або 24 біта пам'яті. Цей спосіб представлення кольорової графіки прийнято називати режимом True Color (True color – точний колір), або **повнокольоровим режимом**. Відмітимо, що повноколірний режим вимагає дуже багато пам'яті. Так, для обговореного раніше растра 800 x 600 при використанні методу RGB потрібно 1 440 000 байт 1,37 Мбайт. З метою економії пам'яті розробляються різні режими і графічні формати.

Більшість дій, які доводиться виконувати над графічними даними в піксельному форматі, зводяться до заміни поточного кольору пікселя іншим кольором. Наприклад, стирання будь-якої ділянки зображення – це не що інше як заміна кольорів всіх пікселів обраної ділянки кольором фону малюнка. Отже, і для графічних даних елементарні дії, які повинен «уміти» виконувати комп'ютер, зводяться до порівняння двох двійкових кодів і заміні одного коду іншим.

2.4 Звукові і відеодані

Розвиток способів кодування звукової інформації, а також зображень в рухові – **анімацій і відеозаписів** – відбувався із запізненням відносно розвитку способів кодування розглянутих вище різновидів інформації. Прийнятні способи зберігання і відтворення за допомогою комп'ютера звукових і відеозаписів з'явилися лише в 1990-х роках. Ці способи роботи зі звуком і відео отримали назву **мультимедійних технологій**.

Звук являє собою достатньо складне, неперервне коливання повітря. Основний підхід до кодування звуку, який називається **перетворенням в цифрову форму, оцифровування або дискретизацією**, оснований на тому, що неперервний звуковий сигнал замінюється дискретним (тобто тим, який складається з відокремлених, розділених елементів) набором значень сигналу – **відліків сигналу** – в деякі послідовні моменти часу (рис. 2.2). Кожен відлік кодується в одному з розглянутих раніше числових форматів. Таким чином, кодування й обробка звукових даних фактично зводиться до кодування й обробки числових даних.

Нехай, будь-яка подія відбувається періодично через рівні проміжки часу T . Тоді кількість подій в одиницю часу $1/T$ називається частотою події. Частота вимірюється в герцах (Гц), при цьому 1 Гц – це одна подія в секунду. Часто використовуються кратні одиниці частоти кілогерци (1 кГц = 10^3 Гц), мегагерци (1 МГц = 10^6 Гц) і гігагерци (1 ГГц = 10^9 Гц).

У відповідності з цим визначенням, кількість відліків сигналу в одиницю часу називається **частотою дискретизації**. Так, на рис. 2.2 сигнал, який триває 2 с, замінюється 100 відліками (жирні крапки на графіку амплітуди сигналу). Отже, в даному випадку частота дискретизації дорівнює 50 Гц. При записі звуку в мультимедійних технологіях застосовують частоти 8, 11, 22 і 44 кГц. Частота дискретизації 44 кГц означає, що одна секунда неперервного звучання замінюється набором з сорока чотирьох тисяч окремих відліків сигналу. Чим вище частота дискретизації, тим краща якість оцифрованого звуку.

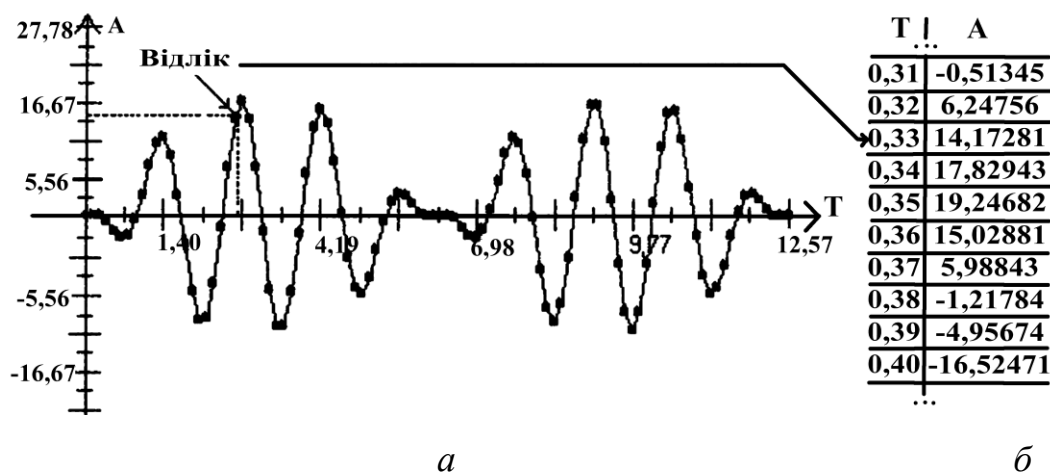


Рисунок 2.2 - Перехід від аналогової (неперервної) форми (а) до цифрової (дискретної) (б)

Кожен окремий відлік являє собою число, яке потім можна представити у вигляді деякого двійкового коду. Якість перетворення звуку в цифрову форму визначається не лише частотою дискретизації, а і кількістю бітів пам'яті, які відводяться на запис одного коду одного відліку. Цей параметр прийнято називати **розрядністю перетворення**. В теперішній час зазвичай

використовуються розрядності 8, 16, і 24 біт. На описаних принципах базується формат **WAV** (від WAVeform - audio – хвильова форма аудіо) кодування звуку. Отримувати запис звуку у цьому форматі можна від підключених до комп'ютера мікрофона, програвача, магнітофона, телевізора й інших пристроїв роботи зі звуком, які стандартно використовують. Однак формат WAV потребує дуже багато пам'яті. Так, при записі стереофонічного звуку з параметрами, які дають хорошу якість звучання, частотою дискретизації 44 кГц і розрядністю 16 біт на одну хвилину запису потрібно приблизно 10 Мбайт пам'яті.

Кодування відеоінформації – ще більш складна проблема, ніж кодування звукової інформації, так як потрібно потурбуватись не лише про дискретизацію неперервних рухів, але і про синхронізацію зображення зі звуковим супроводом. Для цього використовується формат, який називається **AVI** (Audio-Video Interleaved – аудіо і відео, які чергуються). Відмітимо, що основні мультимедійні формати AVI та WAV дуже вимогливі до пам'яті. Тому на практиці застосовуються різні способи компресії (тобто стиснення) звукових і відеокодів.

2.5 Числові дані

Для представлення числових даних у комп'ютерах використовуються два принципіально різних формати: формат з **фіксованою крапкою** (комою) і формат з **плаваючою крапкою** (комою).

Формат з фіксованою крапкою призначений для абсолютно точного представлення цілих чисел. У програмуванні ці числа відносяться до цілого типу, в той час як формат з плаваючою крапкою використовується для представлення тільки нецілих, наближених чисел. У програмуванні такі числа відносяться до дійсного типу. Нагадаємо, що дійсні числа виникають в задачах в результаті різних вимірів (наприклад, вимірювань ваги тіла або його довжини), які, як відомо, завжди виконуються з деякою похибкою, наближено.

2.5.1 Формати цілих чисел

Існують дві модифікації формату з фіксованою крапкою, які прийнято називати його беззнаковим і знаковим представленнями. Беззнакове представлення формату використовується для роботи з цілими додатніми числами, а існуюче в декількох варіантах знакове – для роботи як з додатніми, так і з від'ємними цілими числами.

У **беззнаковому представленні цілого числа** використовується прямий двійковий код, який являє собою запис цього числа в двійковій системі числення. При цьому всі розряди зайнятого числом поля містять його цілі цифри. Крапка, що відокремлює цілу частину числа від дробової, вважається розташованою, фіксованою праворуч від крайнього правого розряду. Отже, під дробову частину числа відводиться нульова кількість

розрядів, і в даному варіанті кодування можлива робота тільки з цілими числами. Постійне розташування, фіксація позиції крапки дала назву формату – з фіксованою крапкою.

В історії розвитку архітектури комп'ютерів використовувалися чотири різних **варіанти подання знакових чисел** :

1. система зі знаком ;
2. зворотний код, порозрядне доповнення або код з доповненням до одиниці;
3. додатковий, комплементарний код або код з доповненням до двох ;
4. система зі зміщенням.

В даний час перші дві системи практично вийшли з вжитку.

В системі кодування із знаком код знака числа прийнято розміщувати в самому лівому розряді поля, який у зв'язку з цим прийнято називати **знаковим бітом**. За традицією знак «плюс» кодується нулем, а знак "мінус" - одиницею. Наприклад, числа $+4_{10}$ і -4_{10} з прямим двійковим кодом модуля 100_2 . Тоді однобайтний код числа $+4_{10}$ і є $0000\ 0100_2$, або 04_{16} , а такий же код числа -4_{10} - це $1000\ 0100_2$, або 84_{16} . Ще приклад: число $+127_{10}$ має в цій системі код $01111\ 1111_2$ ($7F_{16}$), а число -127_{10} - код $1111\ 1111_2$ (FF_{16}). Подивимося тепер, до яких наслідків призведе прийняття пропонованого варіанта кодування знакових цілих чисел. По-перше, виявляється, що числу 0_{10} відповідають два різних коди: код $0000\ 0000_2$ ($+0_{10}$) і код $1000\ 0000_2$ (-0_{10}). Така неоднозначність вкрай небажана, так як необхідно або додатково враховувати її при різних перевірках апаратними засобами комп'ютера, або передбачати окремі перевірки в програмах. По-друге, і це найголовніше, виникає спеціальна арифметика з вдосконаленими незвичними правилами виконання самих звичайних дій. Так, за правилами звичайної арифметики, складання чисел $+4_{10}$ і -4_{10} дає в результаті 0_{10} . А тепер виконаємо додавання для отриманих раніше кодів цих чисел : $00000100_2 + 10000100_2 = 10001000_2$. Як бачимо, отриманий абсолютно несподіваний результат: замість очікуваного коду числа 0_{10} додавання дало код числа -8_{10} . Цей результат є наслідком невдалого вибору способу кодування. Його потрібно вибирати виходячи з *логіки використання коду*, а не з « очевидного » на перший погляд підходу . Основна вимога при виборі системи кодування чисел полягає в тому, що отриманий код повинен задовольняти правилам виконання додавання і віднімання в двійковій системі числення. У зв'язку з цим код кожного наступного *додатного* числа повинен виходити *додаванням* одиниці до коду поточного числа, а код кожного наступного *негативного* числа повинен виходити *відніманням* одиниці з коду поточного числа. Побудований таким чином код прийнято називати **додатковим**, з *доповненням до двох* або *комплементарним* (від complementary – доповнюючий).

Для простоти міркувань розглянемо побудову цього коду на прикладі чотирьох бітів. Код числа 0_{10} - це 0000_2 . Код наступного позитивного числа $+1_{10}$ отримуємо додаванням до нього одиниці: 0001_2 . Аналогічним чином

отримуємо код числа $+2_{10}$: 0010_2 . Подальший процес отримання кодів позитивних чисел зображений на рис. 2.3 , *зліва*. Відзначимо, що при записі коду в чотири біта цей процес можна продовжити тільки до числа $+7_{10}$, що має код 0111_2 . Перехід до наступного додатного числа $+8_{10}$ з кодом 1000_2 призведе до заміни знакового біта старшою цифрою коду модуля. Отже, числа більші $+7_{10}$, таким способом зобразити не можна.

Розглянемо тепер отримання кодів від'ємних чисел. Для отримання коду числа -1_{10} віднімемо з коду числа 0_{10} одиницю: $0000_2 - 1_2$. Це можливо тільки якщо домовитися про те, що необхідна для такого віднімання позика виконується з уявного додаткового розряду: $10000_2 - 1_2 = 1111_2$. Подальше отримання кодів негативних чисел не викликає ускладнень і показано на рис. 2.3, *справа*.

	$0 \rightarrow 0000_2$
$+7 \rightarrow 0111_2$	$-1 \rightarrow 1111_2$
$+6 \rightarrow 0110_2$	$-2 \rightarrow 1110_2$
$+5 \rightarrow 0101_2$	$-3 \rightarrow 1101_2$
$+4 \rightarrow 0100_2$	$-4 \rightarrow 1100_2$
$+3 \rightarrow 0011_2$	$-5 \rightarrow 1011_2$
$+2 \rightarrow 0010_2$	$-6 \rightarrow 1010_2$
$+1 \rightarrow 0001_2$	$-7 \rightarrow 1001_2$
$0 \rightarrow 0000_2$	$-8 \rightarrow 1000_2$

Рисунок. 2.3 - Отримання додаткових кодів

Спосіб кодування знакових чисел, заснований на використанні *додаткового* коду, усуває всі зазначені раніше недоліки застосування системи кодування зі знаком. По-перше, необхідні арифметичні властивості задовольняються автоматично за способом побудови коду. Наприклад, при додаванні кодів 0100_2 ($+4_{10}$) і 1100_2 (-4_{10}) виходить код 10000_2 , старша одиниця якого не поміщається у використовувані чотири розряди і відкидається. Таким чином, залишається код 0000_2 , який в точності відповідає потрібному результату.

Для отримання додаткового коду прямий код інвертується (звертається), тобто кожна цифра 0 коду замінюється цифрою 1, і навпаки, кожна цифра 1 коду замінюється цифрою 0. Звернемо увагу, що отриманий таким чином код називається **зворотним**, з *доповненням до одиниці* або з *порозрядним доповненням*. До зворотного коду добавляється одиниця .

Нехай, наприклад, маємо прямий код $100\ 1101_2$ і вибрано однобайтове поле. Тоді вихідним буде код у вигляді $0100\ 1101_2$, його зворотний код – $1011\ 0010_2$, а додатковий – $1011\ 00\ 11_2$.

Важливим моментом в отриманні правильного кінцевого результату є первісний запис коду в полі відповідної довжини. Візьмемо, наприклад, прямий код 100_2 . Його інвертування дає зворотний код 011_2 , а подальше додавання одиниці – додатковий 100_2 . Збіг прямого та додаткового кодів свідчить про помилку, так як в даному випадку однозначність кодування є одним з основних пред'явлених до коду вимог. Помилка полягає в тому, що

зроблена спроба працювати з невідповідним значенню числа кількістю бітів. Справді, прямий код 100_2 числа 4 займає $n = 3$ біта. Але, як впливає з вищевикладеного, обговорюваним способом цій кількості бітів можна представити коди чисел тільки з діапазону від -4 до +3. Для отримання правильного результату необхідно взяти хоча б 4 біта. Тоді вийде правильний результат: прямий код 0100_2 , зворотний – 1011_2 , а додатковий – 1100_2 . Отже, при виборі довжини поля для запису коду числа необхідно враховувати відповідності між довжиною поля і діапазоном представлених в ньому чисел.

На рис. 2.4 показані знакові подання для чисел +77 і -77 в полях різної довжини. У двійкових кодах вертикальною рисою відділений знаковий біт поля. Порівняння результатів показує, що при переході до полю більшої довжини вільні зліва біти поля заповнюються кодом знакового біта 0_2 (0_{16}) для додатніх і 1_2 (F_{16}) – для від’ємних чисел. Це правило називається *правилом розмноження знака*. Перехід до полів меншої довжини в тих випадках, коли він можливий, здійснюється відкиданням двох, чотирьох або шести цифр 0_{16} або F_{16} або ж відповідної кількості тетрад 0000_2 або 1111_2 .

Коди числа $+77_{10}$ в стандартних полях		
Поле 1 байт	0 100 1101 ₂ або	4D ₁₆
Поле 2 байта	0 000 0000 0100 1101 ₂ або	00 4D ₁₆
Поле 4 байта	0 000 0000 0000 0000 0000 0100 1101 ₂ або	00 00 00 4D ₁₆
Коди числа -77_{10} в стандартних полях		
Поле 1 байт	1 011 0011 ₂ або	B3 ₁₆
Поле 2 байта	1 111 1111 1011 0011 ₂ або	FF B3 ₁₆
Поле 4 байта	1 111 1111 1111 1111 1111 1011 0011 ₂ або	FF FF FF B3 ₁₆

Рисунок 2.4 – Розмноження знака при переході до полів більшої довжини

Щоб визначити значення числа, для якого заданий його машинний код у знаковому поданні формату з фіксованою крапкою, потрібно зробити наступне:

1. За знаковим бітом визначити знак числа. Для від’ємних чисел в двійковому кодуванні крайня зліва цифра дорівнює 1_2 , а в шістнадцятковій ця цифра повинна бути більше 7_{16} . Інакше поле містить прямий код позитивного числа.

2. Якщо число від’ємне, то в полі знаходиться додатковий код, від якого за формулою $P = M - D$ (де D – доповнення n -розрядного прямого коду модуля P цього числа до числа $M = 2^n$: $D = M - P$) слід перейти до прямого коду.

3. Перевести число з двійкової або шістнадцяткової системи числення в десяткову.

4. Приписати результату відповідний знак числа.

Нехай, наприклад, заданий код $005A_{16} = 0000\ 0000\ 0101\ 1010_2$ і відомо, що це знакове подання формату з фіксованою крапкою. За першою цифрою шістнадцятирічного коду 0_{16} або знакового біту 0_2 визначаємо, що в полі знаходиться код позитивного числа, переведення якого в десяткову систему числення дає $+90_{10}$. Ще приклад: нехай у тому ж форматі заданий код $FFE8_{16}$ або $1111\ 1111\ 1110\ 1000_2$. $7_{16} \leq F_{16}$ або рівний 1_2 знаковий біт є ознакою від'ємного числа. Перехід до прямого коду дає $10000_{16} - FFE8_{16} = 18_{16}$ або 24_{10} . Отже, розглянутий код $FFE8_{16}$ відповідає числу 24_{10} . Якщо ж код $FFE8_{16}$ розглядати як беззнакове подання числа, то він відповідає числу $65\ 512_{10}$.

Один і той же код в різних поданнях і, тим більше, різних форматах може відповідати різним числам.

Розглянемо систему **кодування зі зміщенням**, яка в англomовній літературі називається системою **excess 2^{n-1}** . Ця система кодування характеризується цілим позитивним числом – константою зміщення K , яке додається до будь-якого кодованого числа x з тим, щоб сума вихідного числа і констант зміщення $x + K$ потрапила в діапазон допустимих для поля обраної довжини N беззнакових чисел. В якості коду *вихідного числа* вибирається беззнаковий код суми.

Якщо взяти спеціальний випадок, коли константа $K=2^n$, де n - довжина поля в бітах, то виявиться, що система з таким зміщенням близька до додаткового коду.

Візьмемо, наприклад $n=8$, тоді $K=128$ і $\bar{Z}_N^{128} = \{ x \in \mathbb{Z} \mid -128 \leq x \leq 127 \}$. Як можна помітити, в даному випадку діапазони системи зі зміщенням і додаткового коду збігаються. Практично однакові і отримані при цьому коди чисел. Наприклад, в якості коду числа -4 в цій системі вибирається беззнаковий код числа 124 , що рівний $0111\ 1100_2$ або $7C_{16}$. Якщо порівняти отриманий результат з додатковим кодом $1111\ 1100_2$ того ж самого числа -4 , то легко побачити, що вони розрізняються тільки в знаковому біті. Іншими словами, код числа в системі зі зміщенням з константою $K=2^n$ співпадає з додатковим кодом, у якого інвертований знаковий біт.

Як ми побачимо надалі, в даний час система кодування зі зміщенням використовується як один з елементів подання чисел у форматі з плаваючою крапкою.

Необхідно зазначити, що пристрій, який здійснює арифметичні операції над числами в двійковій системі числення, має забезпечувати виконання дії додавання, віднімання та зсуву, так як множення і ділення зводяться саме до цих операцій. У зв'язку з тим, що віднімання завжди можна замінити додаванням з від'ємним числом, стає зрозуміло, що процесори комп'ютерів можуть «обійтися» і без операції віднімання.

Суму цілих двійкових чисел з урахуванням знаків можна звести до суми їх додаткових або зворотних кодів. **Наприклад:**

$$\begin{array}{r}
 (+9)_{10} = 0\ 1001 \quad \quad \quad 0\ 1001 \quad \quad \quad 0\ 1001 \\
 + \\
 (-5)_{10} = 1\ 0101 \xrightarrow{\text{зворотний код}} 1\ 1010 \quad \quad \quad 1\ 1010 \\
 \quad \quad \quad \quad \quad \quad + \quad 1 \quad \quad \quad 1\ 0\ 0011 \\
 \quad \quad \quad \text{додатковий код} \rightarrow 1\ 1011 \quad \quad \quad 1\ 1011 \quad \quad \quad \text{L} \rightarrow + 1 \\
 \quad \quad \quad \quad \quad \quad \quad \quad \quad 1\ 0\ 0100 \quad \quad \quad \quad \quad \quad 0\ 0100 \\
 \quad \quad \quad \quad \quad \quad \text{перенесення ігнорується} \uparrow
 \end{array}$$

Таким чином, для оброблення цілочисельних даних процесору досить «вміти» виконувати додавання і зсув для двійкових кодів [16-18].

2.5.2 Формат дійсних чисел

У багатьох розрахункових задачах використовуються величини, які є результатами багатьох різних вимірювань або отримані за допомогою різних математичних операцій над вимірюваними значеннями. Їхньою відмінною особливістю є принципово наближений характер, а значення ніколи не бувають в точності рівними цілому числу. Такого роду величини і їх значення прийнято називати дійсними.

Для кодування дійсних чисел у форматі з плаваючою крапкою використовується їх нормалізована форма $\pm m * p^{\pm q}$, $1 \leq m < 10$. Аналіз цієї форми показує, що у системі числення з заданою основою p код нормалізованого числа повинен містити:

1. код знака числа;
2. код нормалізованої мантиси $1 \leq m < 10$;
3. код знака порядку;
4. код порядку p .

Для спрощення розуміння матеріалу обговорення основних особливостей цього методу кодування проведемо із залученням уявного пристрою, що складається з восьми десяткових розрядів. У кожен розряд цього пристрою, який ми назовемо «навчальна комірка» можна записати будь-яку цифру десяткової системи числення, а не тільки 0 або 1. На рис. 2.5 показані два з декількох можливих варіантів розподілу розрядів навчальної комірки, яке забезпечує зберігання всіх елементів коду нормалізованого числа.

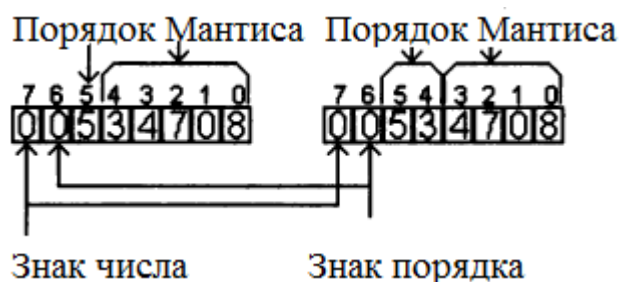


Рисунок 2.5 – Можливі розрядні сітки навчальної комірки

На рис. 2.6 наведені приклади отримання значення чисел для різних варіантів заповнення знакових шостого та сьомого розрядів навчальної комірки при одному і тому ж вмісті інших її розрядів.

7 6 5 4 3 2 1 0 0015347108	$+3,4708 \cdot 10^{+5}$
7 6 5 4 3 2 1 0 0115347108	$+3,4708 \cdot 10^{-5}$
7 6 5 4 3 2 1 0 1015347108	$-3,4708 \cdot 10^{+5}$
7 6 5 4 3 2 1 0 1115347108	$-3,4708 \cdot 10^{-5}$

Рисунок 2.6 – Приклади кодування чисел в навчальній комірці

Для представлення даних в реальних полях пам'яті комп'ютера, що складаються з бітів та байтів, а не з умовних десятичних розрядів в 1985 році комісією IEEE (Institute Electrical Electronics Engineers - Міжнародний інститут інженерів з електротехніки та електроніки) був створений стандарт IEEE 754, який передбачає, що двійкові коди чисел у форматі з плаваючою крапкою можуть займати поля довжиною 4, 8 і 10 байтів. Відповідно до загальних принципів кодування чисел в цьому форматі поля містять код знака числа, код порядку та його знака, а також код мантиси.

Код знака числа як завжди займає один, самий лівий розряд поля. Кодується знак також стандартним чином: 0 для позитивних чисел і 1 для негативних.

Раніше було з'ясовано, що у нормалізованих чисел порядки можуть бути як додатніми, так і від'ємними цілими числами. Отже, постає питання про кодування в деякій частині поля знакового цілого числа. Фактично, в прикладах, наведених на рис. 2.5, 2.6, для кодування порядку використана система кодування зі знаком, яка неприйнятна для практичної реалізації по з'ясованій раніше причини – в силу виникнення для цієї системи спеціальної арифметики. По ряду технічних причин зворотна і доповнювальна системи також виявилися незручними для кодування окремої групи бітів поля. Тому розробники стандарту запропонували використовувати для кодування порядку зі знаком системи зі зміщенням, константа якої дорівнює $K=2^n - 1$, де n - кількість розрядів поля, виділених під зберігання порядку і його знака. Якщо істинний порядок нормалізованого числа дорівнює P_e , то в зазначених бітах поля записується прямий двійковий (беззнаковий) код суми $P_m = P_e + K$, який прийнято називати машинним порядком вихідного дійсного числа. Кількість розрядів n , яка виділяється під машинний порядок, дорівнює 8, 11 і 15 для 4-, 8- і 10-байтових полів відповідно.

Щоб отримати машинний код числа у форматі з плаваючою крапкою, слід:

1. перевести модуль числа з десяткової системи числення в двійкову;
2. нормалізувати число;
3. сформулювати знаковий біт;
4. виділити істинний порядок P ;

5. отримати код машинного порядку $P = P + K$;
6. для чотирьох і восьмибітових полів отримати код мантиси M , відкидаючи біт цілої частини, для десятибайтового поля цей біт не відкидати;
7. записати код знака, код порядку P , і код мантиси M у відведені для них позиції розрядної сітки.

Знайдемо, наприклад, код числа $-15,375_{10}$ у форматі з плаваючою крапкою в чотирьохбайтовому полі. Переклад цього числа в двійкову систему числення дає в результаті $-1111,011_2$. Для нормалізації числа кому потрібно змістити на три позиції вліво. Щоб значення числа при цьому не змінилося, його слід помножити на основу системи числення в третій ступені. Таким чином, нормалізоване число має вигляд $1,111011 * 10_{10}^{+11}$. Зверніть увагу на те, що порядок $+3_{10}$ записаний як $+11_2$. Далі отримуємо код знака числа 1 і його справжній порядок $P_u = 3_{10} = 3_{16} = 11_2$. Код машинного порядку має вигляд $P_m = 11_2 + 111\ 1111_2 = 1000\ 0010_2$. Якщо у мантиси нормалізованого представлення числа $1,111011_2$ відкинути цілу частину (разом з комою), то вийде код мантиси $M = 111011_2$. Збираючи разом всі елементи коду і заповнюючи незначущі праві біти мантиси нулями, отримаємо шуканий код числа у вигляді $1|10000010|1110110000000000000000_2$. Для наочності в запису коду вертикальною рисою відокремлені один від одного біти знака, порядку і мантиси. Слід розуміти, що в реальних машинних кодах таких розділових знаків немає. Отже, код числа $-15,375_{10}$ у форматі з плаваючою крапкою в чотирьохбайтовому полі має вигляд $1100\ 0001\ 0111\ 0110\ 0000\ 0000\ 0000\ 0000_2$, або в компактному шістнадцятковому вигляді $C1\ 76\ 00\ 00_{16}$.

Логіка обчислень з даними, представленими у форматі з плаваючою крапкою, носить складний характер розрахунків, для якого необхідно, принаймні, вміти виділяти поодинокі біти і деякі групи бітів - ділянки полів, які відповідають кодам порядку і мантиси. Для чого мінімальний набір дій, якими повинен володіти процесор комп'ютера, повинен включати операції порівняння, додавання, диз'юнкції і кон'юнкції, а також пересилки, заперечення і зсуву одиночного коду на деяку кількість розрядів вправо і вліво. Саме цей набір дій спільно з операціями, які забезпечують організацію розгалужень, циклів, виклик підпрограм і т. д., утворює основу системи команд будь-якого процесора.

3 ЛОГІЧНІ ОСНОВИ ОБРОБКИ ДАНИХ

Після вияснення способів представлення даних в пам'яті комп'ютера необхідно вияснити як відбувається їх обробка. В даному розділі обговорюються деякі питання, які пов'язані з фізичними і логічними основами обробки *дискретних* даних [18-21].

3.1 Поняття такту

У виконанні будь-яких дій над даними беруть участь декілька пристроїв комп'ютера, дії яких повинні бути узгоджені і синхронізовані одне з одним. Цю роль в комп'ютері грає спеціальний пристрій – **тактовий генератор**, який через рівні проміжки часу виробляє імпульси синхронізації – **синхроімпульси**, які слугують орієнтирами у часі і використовуються для координації усіма пристроями (рис. 3.1). Тривалість імпульсів, які виробляються, також однакова.

Проміжок часу від початку одного імпульсу синхронізації до початку наступного імпульсу називається тактом. Такти мають рівні тривалості з високим ступенем точності .

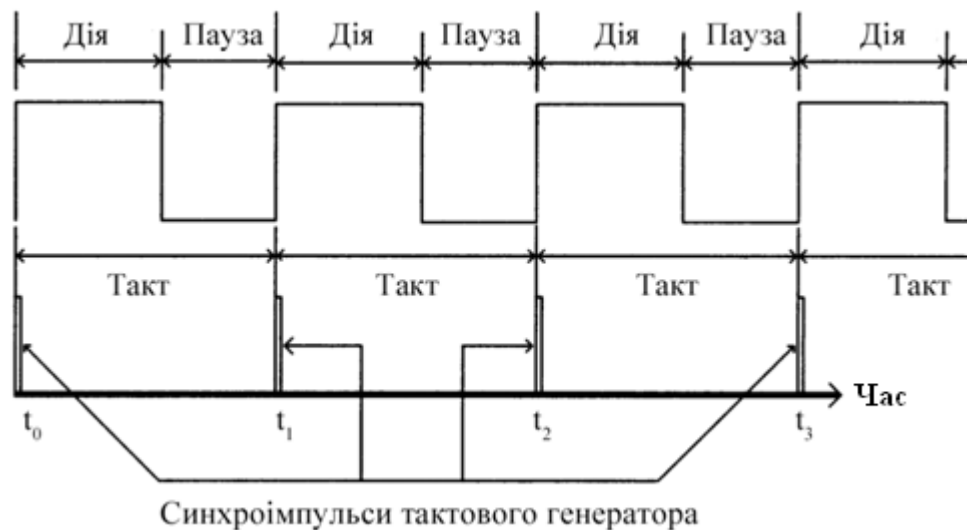


Рисунок 3.1 - До поняття такту

Виконання процесором будь-яких дій завжди відбувається під час деякої частини такту, потім настає пауза. Наявність такої паузи є принциповим фактором, обумовленим фізичними законами, які управляють роботою процесора. Обговорюючи виконання процесором визначеної в програмі послідовності дій, можна вважати, що будь-які дії, які тривають всередині такту, відбуваються *миттєво* в моменти часу t_0 , t_1 , t_2 ,... відповідні межах тактів. З поняттям такту пов'язана *тактова частота* – одна з найважливіших технічних характеристик різних пристроїв комп'ютера.

Тактова частота являє собою технічну характеристику окремих пристроїв комп'ютера, яка дорівнює кількості тактів, що керують роботою пристрою, в одиницю часу. Одиницею виміру тактової частоти є герц, який дорівнює одному такту в секунду.

Тактова частота є одним з головних факторів, які визначають швидкість обробки даних комп'ютером. Перші персональні комп'ютери робили на частотах 5–10 МГц. В теперішній час процесори комп'ютерів працюють з тактовими частотами від сотні мегагерц до сотень ГГц.

3.2 Вентилі і комбінаційні схеми

Основними базовими операціями, які обов'язково повинен "вміти" виконувати процесор комп'ютера над двійковими кодами даних, є логічні операції заперечення, диз'юнкції, кон'юнкції, арифметичного додавання, а також зсуву коду. Пристрої, які використовуються для реалізації цих й інших операцій, прийнято називати **вентиллями** (від нім. Ventil – клапан).

Вентилем називають фізичний пристрій, який реалізує одну з базових логічних операцій: заперечення, диз'юнкцію, кон'юнкцію, виключну диз'юнкцію і т. д. Вентилі, які входять до складу процесорів комп'ютера, називають також **логічними елементами**.

3.2.1 Релейно – контактні вентилі

У 1938 р. відомий спеціаліст у галузі теорії інформації Клод Шеннон запропонував використовувати для моделювання основних логічних операцій релейно-контактні електричні схеми. Цей підхід був використаний в електромеханічних релейних обчислювальних машинах Z-3 (Німеччина, Конрад Цузе, 1939), "Марк 2" (США, Говард Айкен, 1947), РВМ-1 (СРСР, М. І. Бессонов, 1951) і в цілій низці інших машин.

В якості прикладу розглянемо зображену на рис. 3.2 реалізацію логічних операцій кон'юнкції, диз'юнкції та заперечення за допомогою схем, які прийнято називати вентиллями "І", "АБО", "НІ" відповідно. Логічні операнди в цих схемах відповідають релейно-контактним перемикачам, які на рисунку позначені як p і q . При цьому логічне значення 0 і 1 моделюється відповідно розімкнутим і замкнутим станом контакту. Результат операції відображається включеною в мережу лампочкою. Якщо лампочка не горить, результат дорівнює 0; якщо лампочка горить, то вона відповідає результату, який дорівнює 1. Інакше кажучи, знаки двійкового алфавіту 0 і 1 моделюються *відсутністю* або *наявністю струму* в колі відповідно. Відмітимо, що перемикачі називаються релейними, оскільки управління ними зазвичай виконується за допомогою електромагнітних реле.

На рис. 3.2, *зліва* показані чотири стани вентиля "І", які відповідають різним рядкам таблиці істинності (табл. 3.1) цієї операції. Так, у стані *a* обидва контакти роз'єднані, тобто $p = 0$ і $q = 0$, і, відповідно, струм в колі не тече – лампочка не горить – $p \wedge q = 0$. Стан *б* і *в* моделюють інші рядки таблиці істинності операції кон'юнкції, в яких результат дорівнює нулю: $1 \wedge 0 = 0$ і $0 \wedge 1 = 0$ відповідно. Струм в колі тече і лампочка горить лише в стані *г*, тобто коли замкнуті обидва контакти, що відповідає рядку таблиці $1 \wedge 1 = 1$.

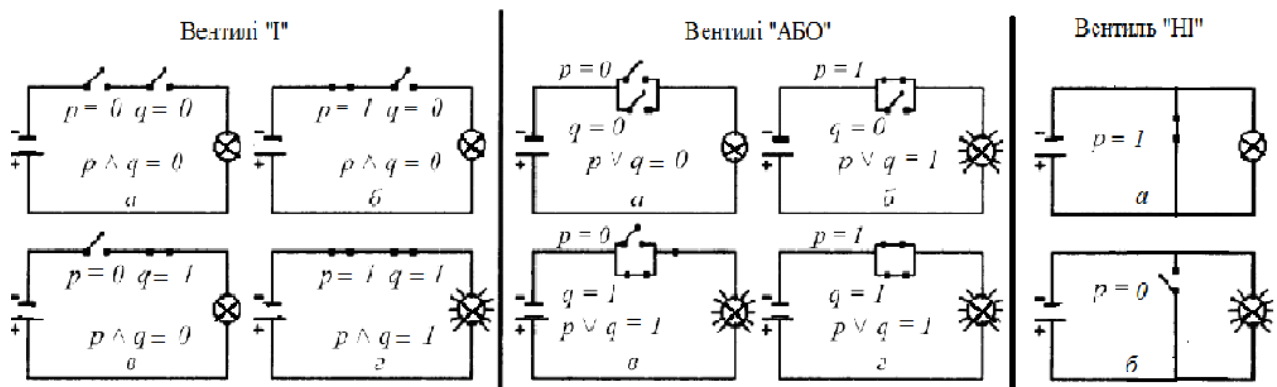


Рисунок 3.2- Вентилі "І" , "АБО" та "НІ" на базі релейно–контактної схеми

На тому ж малюнку посередині зображена схема вентиля "АБО", яка відповідає операції диз'юнкції. Лампочка не горить тільки в стані *a*, коли розімкнуті обидва контакти, тобто $0V0 = 0$. У всіх інших станах лампочка горить, і, відповідно, результат операції дорівнює 1, тобто моделюються рядки таблиці істинності $1V0 = 1$, $0V1 = 1$ і $1V1 = 1$. Релейна схема вентиля "НІ", який відповідає логічній операції заперечення, зображена на рисунку справа. Лампочка не горить в стані *a*, тобто коли контакт замкнутий.

3.2.2 Напівпровідникові вентилі

Основним недоліком описаного способу реалізації операції над даними є *наявність механічних переміщень* контактів релейної схеми, які потребують значних часових затрат. Час виконання операції визначається часом, протягом якого відбувається замикання або розмикання контакту. Навіть в самих сучасних релейних пристроях на це потрібні як мінімум соті долі секунди. Саме ця обставина привела до того, що вже в 1950–ті роки релейні обчислювальні машини були витіснені спочатку ламповими, а потім транзисторними комп'ютерами.

Всі сучасні вентилі реалізуються на основі напівпровідникових пристроїв – **транзисторів** або їх аналогів у інтегральних схемах, характерний час спрацювання яких наближається до частки наносекунди ($1 \text{ нс} = 10^{-9} \text{ с}$).

У вентильних системах на базі транзисторів двійковому знаку 0 відповідає низька напруга з рівнем від 0 до 1 В, а двійковому знаку 1 – висока напруга з рівнем від 2 до 5 В. Можуть застосовуватись й інші конкретні значення напруг, але в будь-якому випадку використовуються її *два чітко помітних рівня*. Подачу на базу транзистора низької напруги можна трактувати як надходження на вхід системи біта зі значенням 0, а подача на базу високої напруги відповідає надходженню на вхід системи біта зі значення 1. Аналогічно реєстрація низької напруги на колекторі транзистора може трактуватись як формування на виході значення 0, а реєстрація на ньому високої напруги – як формування на виході значення 1.

3.2.3 Вентиль "НІ"

Розглянемо поведінку схеми, зображеної на рис. 3.3, *а*, при різних значеннях вхідного біта p . Нехай на вхід схеми подано значення $p = 0$. Тоді транзистор замкнений, він веде себе в ланцюгу як додатковий резистор з опором набагато більшим, ніж опір резистора, через який транзистор підключений до живлення схеми. В силу того, що падіння напруги на частині ланцюга пропорційно опорів цієї частини, напруга в точці виходу буде мало відрізнятися від високої напруги джерела живлення. Інакше кажучи, на виході системи в цьому випадку формується значення 1.

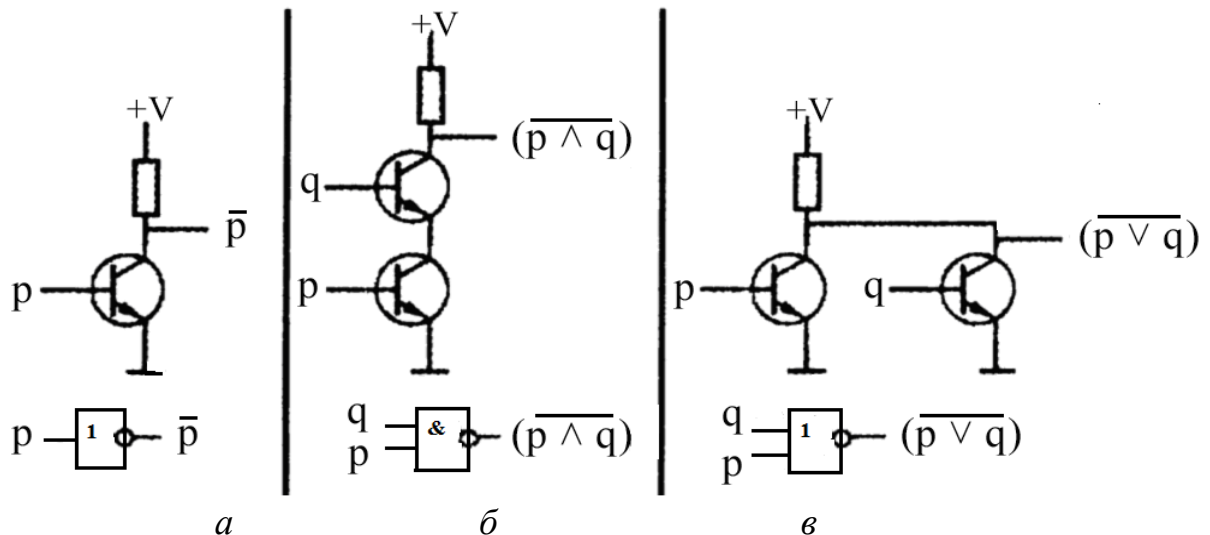


Рисунок 3.3- Вентилі та їх позначення: *а* – "НІ", *б* – "І НІ", *в* – "АБО НІ"

Нехай тепер на вхід надійшов біт $p = 1$. Тоді транзистор відкритий, він веде себе як провідник з надзвичайно малим опором, і всі падіння напруги відбуваються на резисторі. Вихід системи при цьому виявляється як би напругу з'єднаний з землею, тобто, напруга на ньому близька до нуля. Інакше кажучи, на виході схеми формується значення 0. Таким чином схема, яка обговорюється, представляє собою вентиль "НІ", який реалізує операцію заперечення: при надходженні на вхід значень p на виході схеми формується значення $\bar{p}$. Відмітимо, що час переключення вентиля з одного стану в інший суттєво залежить від фізичної реакції транзистора. Але в будь-якому випадку це переключення для електронних приладів відбувається дуже швидко, за час, який вимірюється мікро-, наносекундами або їх частками.

3.2.4 Вентилі " АБО НІ " та " І НІ "

Розглянемо логіку роботи схеми, зображеної на рис. 3.3, *б*. Вона складається з двох з'єднаних послідовно транзисторів. У цієї схеми два входи, позначені на рисунку буквами p і q , і один вихід. Якщо на входи надходять одиничні значення ($p=1$ і $q=1$), то обидва транзистора відкриті, частина ланцюга з ними має дуже малий опір і, відповідно, як і в вентилі "НІ", на виході формується значення 0. У всіх інших випадках хоча б один з транзисторів

виявляється закритим і частина ланцюга з транзисторами має високий опір, що призводить до формування на виході значення 1. Аналізуючи таблицю істинності роботи схеми (табл. 3.1, четвертий стовпець), приходимо до висновку, що вона описує вираз $(\overline{p \wedge q})$. Тому така схема називається вентилем "І НІ" і відома також під назвою "штрих Шеффера".

В схемі, зображеній на рис. 3.3, в, транзистори з'єднані паралельно. Відповідно, частина ланцюга з транзисторами має високий опір лише в тому випадку, коли обидва транзистори закриті одночасно. Тому якщо на обидва входи надходять нульові значення ($p=0$ і $q=0$), на виході формується значення 1. У всіх інших випадках хоча б один з транзисторів відкритий і, відповідно, вся частина ланцюга, яка містить їх паралельне з'єднання, має малий опір. Це значить, що на виході схеми формується значення 0. Аналізуючи таблицю істинності роботи цієї схеми (табл. 3.1, сьомий стовпець), приходимо до висновку, що вона описує вираз $(\overline{p \vee q})$. Тому така схема називається вентилем "АБО НІ" і відома також під назвою "стрілка Пірса".

Таблиця 3.1- Таблиці істинності базових вентилів

p	q	$p \wedge q$, "І"	$\overline{(p \wedge q)}$, "І НІ" штрих Шеффера	$p \vee q$, "АБО"	$p \underline{\vee} q$, "Виключне АБО"	$\overline{(p \vee q)}$, "АБО НІ", стрілка Пірса
1	2	3	4	5	6	7
0	0	0	1	0	0	1
0	1	0	1	1	1	0
1	0	0	1	1	1	0
1	1	1	0	1	0	0

Вентилі "НІ", "І НІ" і "АБО НІ", які використовуються для побудови інших вентилів і довільних схем, вважаються базовими, а схеми, які отримуються за допомогою різноманітних комбінацій базових вентилів, прийнято називати **цифровими логічними схемами**. Важливим частковим випадком цифрових схем є **комбінаційні схеми**, в яких значення, які отримуються на виходах схеми, залежать лише від значень, які поступають на її входи. Так схеми класифікуються також як **схеми без пам'яті**.

Умовні позначки трьох розглянутих раніше вентилів наведені в нижній частині рис. 3.3 під відповідними їм схемами.

3.2.5 Вентилі "І" й "АБО"

Теоретично для задання будь-якої логічної функції можна обійтись лише однією операцією – стрілкою Пірса або штрихом Шеффера. Таким чином, вентилі "І НІ" та "АБО НІ" можуть розглядатись як універсальні (**функціонально повні**), із яких можна скласти схему, що відповідає будь-якому логічному виразу. Але отримані при цьому логічні вирази і відповідні їм

цифрові схеми виявляються надзвичайно громіздкими і малозрозумілими. В той самий час відомо, що логічні функції зручно задавати, використовуючи три основні логічні операції: заперечення, диз'юнкцію і кон'юнкцію. У зв'язку з цим доцільно використовувати в комбінаційних схемах вентиля, які відповідають цим операціям.

Спосіб побудови вентилів для операцій кон'юнкції і диз'юнкції впливає з очевидних співвідношень $\overline{p \wedge q} \equiv p \vee \overline{q}$ і $\overline{p \vee q} \equiv p \wedge \overline{q}$. Отже, з'єднавши виходи вентилів "І НІ" і "АБО НІ" зі входом вентиля "НІ", отримаємо зручні для побудови будь-яких цифрових схем вентиля "І" та "АБО" операцій кон'юнкції та диз'юнкції відповідно. Схеми цих вентилів та їх позначення наведені на рис. 3.4. Вентилі "І" й "АБО" також відносять до базових.

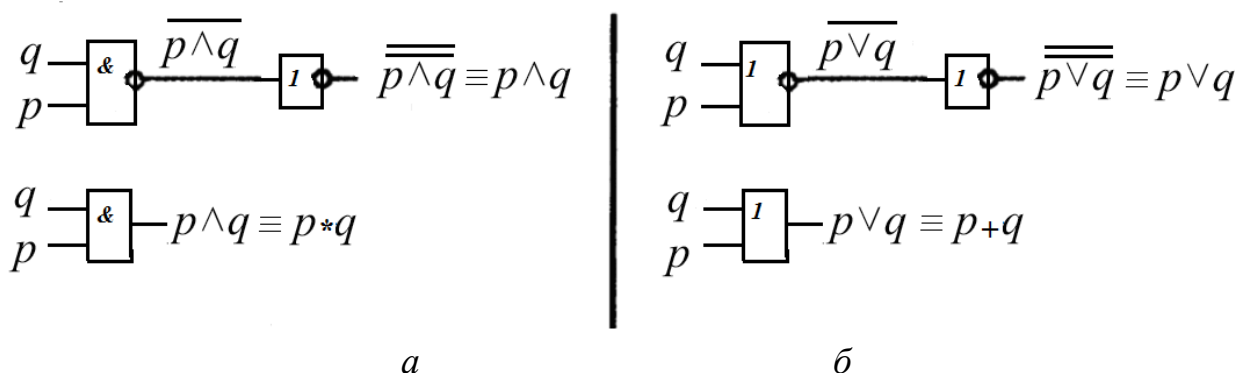


Рисунок 3.4 - Вентилі та їх умовні позначення: *a* – "І", *б* – "АБО"

Відмітимо, що для реалізації вентиля "НІ" достатньо одного транзистора, для вентилів "І НІ" та "АБО НІ" потрібно по два транзистора, а для вентилів "І" й "АБО" необхідно вже по три транзистора на кожну схему.

3.2.6 Синтез комбінаційних схем

Побудова комбінаційної схеми пристрою, що забезпечує виконання потрібних операцій, можлива при наявності *логічної функції*, яка описує логіку її роботи. В загальному випадку логічна функція може бути точно задана за допомогою таблиці істинності. Опираючись на таблицю істинності, можна сформулювати логічну функцію у вигляді так званої диз'юнктивної нормальної форми (ДНФ) згідно алгоритму:

- для кожного рядку таблиці істинності складають *кон'юнкти* (елементарні кон'юнкції), які повинні містити всі аргументи таблиці істинності;
- обираються кон'юнкти всіх рядків таблиці, в яких результат дорівнює 1.
- з усіх обраних кон'юнктів складається диз'юнкція.

Після чого проводиться її мінімізація і перетворення до вигляду, зручного до реалізації на логічних елементах заданого типу.

Для мінімізації логічної функції можна використати **основні закони алгебри логіки**:

- переміщувальний закон: $a + b = b + a$; $a b = b a$;
- сполучний закон: $(a + b) + c = a + (b + c)$; $(a b) c = a (b c)$;

- розподільний закон: $a(b+c) = ab+ac$; $a+bc = (a+b)(a+c)$;
- закон поглинання: $a+ab = a(1+b) = a$; $a(a+b) = a+ab = a$;
- закон склеювання: $ab+ab = a$; $(a+b)(a+b) = a$;
- закон заперечення: $\overline{a+b} = \overline{a} \overline{b}$; $\overline{a} \overline{b} = \overline{a+b}$;
- $\overline{\overline{a+b}} = a+b$; $\overline{\overline{a} \overline{b}} = a+b$;

Закон заперечення часто називають **правилом де Моргана**. Він справедливий для будь-якого числа змінних.

Цей шлях універсальний, але кінцевий результат залежить від кваліфікації фахівця, що проводить мінімізацію. Для функцій, що мають не більше п'яти-шести аргументів, зручно проводити мінімізацію за допомогою **діаграм Вейча (карт Карно)**.

Попередньо потрібно представити функцію в ДНФ формі, заповнити прямокутну таблицю, ставлячи 1 в клітинки таблиці при рівності одиниці, відповідних кон'юнкцій початкової функції, і 0 в решту клітинок таблиці.

На рис. 3.5 подані карти Карно для функцій декількох змінних.

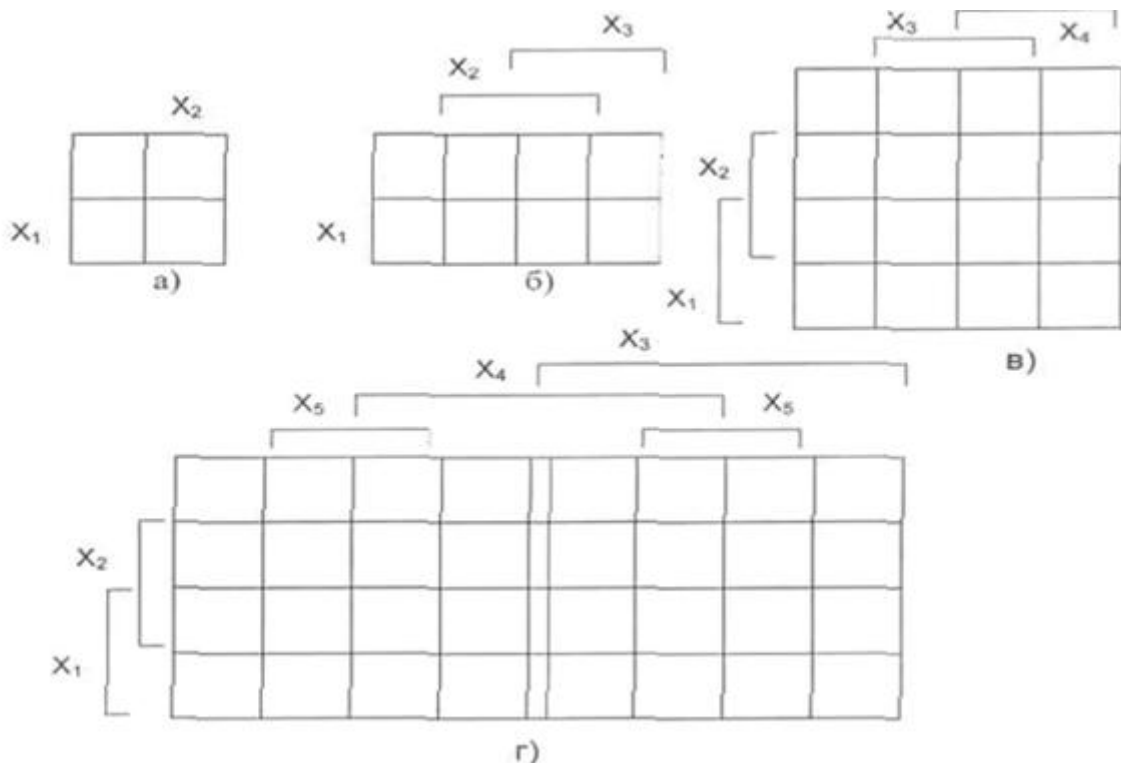


Рисунок 3.5 - Карты Карно для а) - двох, б) - трьох, в) - чотирьох, г) - п'яти змінних

В заповненій таблиці обводять контурами всі одиниці, а потім записують мінімізовану функцію у вигляді суми логічних добутків, що описують ці функції. В середині контуру повинні знаходитись тільки клітинки, заповнені одиницями, число яких повинно бути цілою ступінню числа 2, а одні і ті ж клітинки можуть входити в декілька контурів.

Розглянемо більш докладно процес синтезу комбінаційних ланцюгів на прикладі: необхідно побудувати мажоритарний елемент (елемент голосування) на

три входи, тобто такий елемент, у якого сигнал на виході дорівнює одиниці тоді, коли більшість вхідних сигналів дорівнюють одиниці.

Складемо таблицю істинності:

N	X_1	X_2	X_3	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

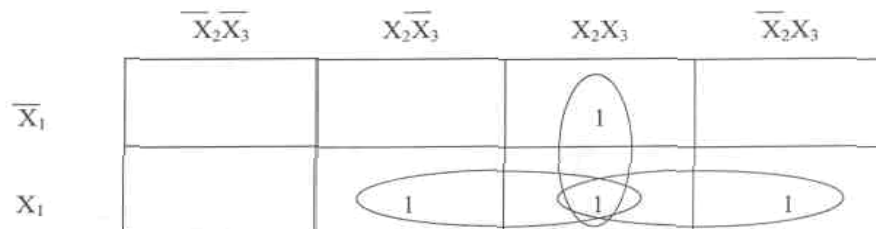
Користуючись таблицею, запишемо логічну функцію, для чого подамо її у вигляді суми логічних добутків, що відповідають тим стрічкам, для яких функція F дорівнює одиниці. Для мінімізації функції використаємо основні закони алгебри логіки:

$$\begin{aligned}
 F &= \overline{X_1}X_2X_3 \vee X_1\overline{X_2}X_3 \vee X_1X_2\overline{X_3} \vee X_1X_2X_3 = \\
 &= (\overline{X_1}X_2X_3 \vee X_1X_2X_3) \vee (X_1\overline{X_2}X_3 \vee X_1X_2X_3) \vee (X_1X_2\overline{X_3} \vee X_1X_2X_3) = \\
 &= X_2X_3(\overline{X_1} \vee X_1) \vee X_1X_3(\overline{X_2} \vee X_2) \vee X_1X_2(\overline{X_3} \vee X_3) = X_2X_3 \vee X_1X_3 \vee X_1X_2
 \end{aligned}$$

Як видно, отриманий кінцевий вираз явно простіше початкового.

В переглянutoму прикладі для мінімізації функції ми використали шлях алгебраїчних перетворень. Цей шлях універсальний, але кінцевий результат залежить від кваліфікації фахівця, що проводить мінімізацію. Для функцій, що мають не більше п'яти-шести аргументів, зручно проводити мінімізацію за допомогою діаграм Вейча.

Мінімізуємо за допомогою карти Карно функцію трьох змінних



В даному випадку всі одиниці в таблиці можна охопити трьома контурами. Виписавши позначення цих контурів, отримаємо мінімізовану функцію

$$F = X_1X_2 + X_1X_3 + X_2X_3.$$

У мінімізованому рівнянні немає змінних з інверсіями, а це означає, що елементи НІ для нашого пристрою непотрібні. Потрібні три логічні елементи І, які виконують три логічні операції множення: $X_1 * X_2$; $X_2 * X_3$; $X_1 * X_3$, а також два елемента АБО, які реалізують логічні суми в мінімізованому рівнянні. Замість двох елементів АБО на два входи можна використати також один на три входи. Принципову схему пристрою зображено на рисунку 3.6.

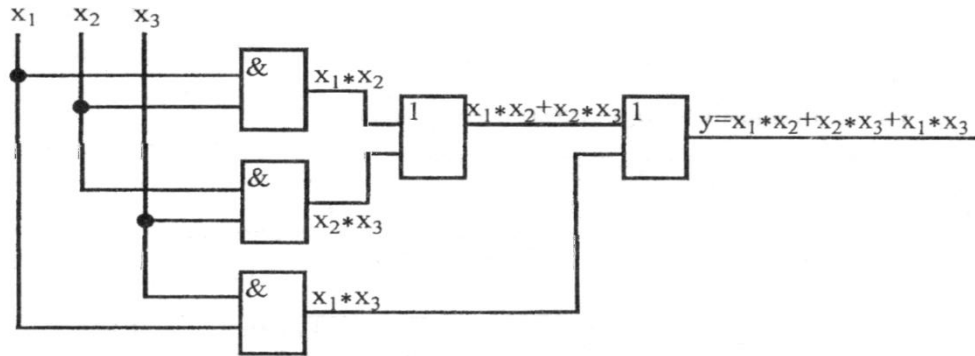


Рисунок 3.6 – Логічна схема мажоритарного пристрою

Промислові інтегральні мікросхеми будь-якого функціонального призначення будують в основному на базі логічних елементів І-НІ або АБО-НІ. Щоб реалізувати пристрій на елементах І-НІ логічне рівняння перетворимо, використавши закони де Моргана.

$$F = \overline{\overline{X_1 X_2} * \overline{X_1 X_3} * \overline{X_2 X_3}}$$

Принципову схему пристрою зображено на рисунку 3.7.

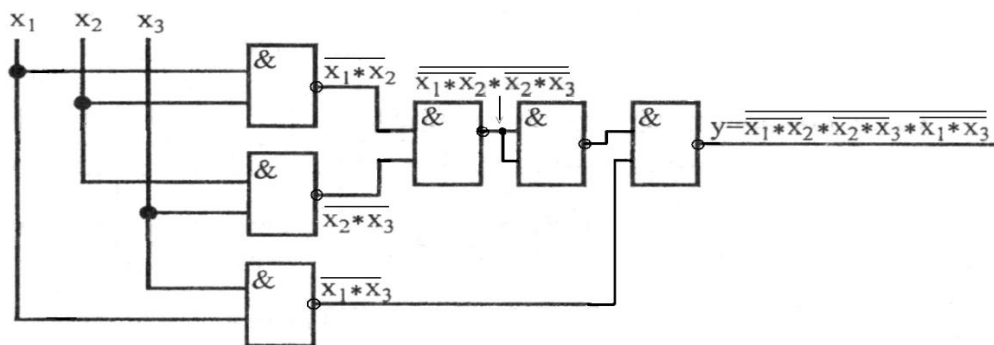


Рисунок 3.7 – Логічна схема мажоритарного пристрою на елементах І-НІ

Проілюструємо побудову ДНФ для логічної операції $p \vee q$ "Виключне АБО", яка визначається наступним чином: результат операції дорівнює одиниці, якщо операнди p і q мають різні значення, і дорівнює нулю, якщо мають однакові значення. Таблиця істинності цієї операції наведена в шостому стовпці табл. 3.1. Побудова по описаним раніше правилам кон'юнктив для всіх рядків таблиці дає наступний результат.

p	q	$p \vee q$	Елементарні кон'юнкції
0	0	0	$\overline{p} \wedge \overline{q}$
0	1	1	$\overline{p} \wedge q$
1	0	1	$p \wedge \overline{q}$
1	1	0	$p \wedge q$

Утворюючи диз'юнкцію з кон'юнктив другого і третього рядку таблиці, в яких результат дорівнює 1, приходимо до висновку про те, що вираз $\overline{p} \wedge q \vee p \wedge \overline{q}$ являє собою ДНФ операції $p \vee q$.

3.2.7 Вентиль "Виключне АБО"

Операція "Виключне АБО" виявляється корисною у багатьох ситуаціях побудови комбінаційних схем. Опираючись на ДНФ $(p \vee q) \equiv \overline{p} \wedge q \vee p \wedge \overline{q}$ і використовуючи базові вентиля "НІ", "І" й "АБО", досить легко побудувати відповідну схему. Для реалізації виразу виходи вентилів "НІ" потрібно з'єднати з одним з входів вентилів "І". Значення p і q подаються на вільні входи вентилів "І" а їх виходи – на входи вентиля "АБО" відповідно. Схема вентиля "Виключне АБО" і його умовне позначення наведені на рис. 3.8, а. Для реалізації цього вентиля потрібно 11 транзисторів.

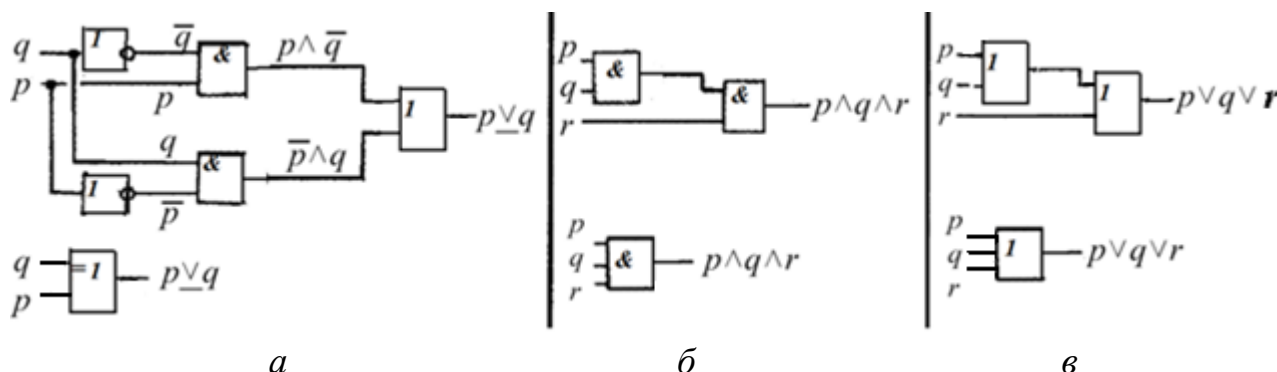


Рисунок 3.8- Вентиль "Виключне АБО" (а) і багатовходові вентиля (б, в)

Елемент "Виключне АБО" в разі двох входів поводить себе як елемент перевірки на непарність. Для перевірки на парність можна використати елемент "Виключне АБО-НЕ", для якого ДНФ $(p \vee q) \equiv \overline{p} \wedge \overline{q} \vee p \wedge q$. Але якщо входів з певним значенням більше двох, то елементи потребують певного налаштування для перевірки на парність і непарність.

3.2.8 Багатовходові вентиля

Стандартні вентиля "І", "АБО" є **двохходовими**, тобто вони містять два входи і один вихід. В загальному випадку вентиля "І" й "АБО" можуть бути **багатовходовими**, тобто містити більше двох входів. Відповідні схеми можна отримати, збільшуючи кількість послідовно або паралельно з'єднаних транзисторів, наведених на рис. 3.3. Багатовходові вентиля можна також

побудувати зі стандартних двовходових вентилів "І" й "АБО", послідовно з'єднуючи їх. На рис. 3.8, *б* і *в* зображені **трьохвходові** варіанти таких вентилів. Для реалізації n -входового вентиля, очевидно, потрібно $n+1$ транзисторів при використанні першого підходу та $3(n-1)$ – при використанні другого. Подібним чином можна реалізувати і багатовходові вентиля "І НІ" та "АБО НІ".

3.2.9 Комбінаційна схема суматора

Тепер розглянемо комбінаційні схеми, за допомогою яких може бути реалізовано арифметичне додавання. Аналіз алгоритму додавання двійкових кодів показує, що додавання молодших бітів і додавання всіх інших бітів, які додаються, відбувається по-різному. Відмінність обумовлена необхідністю враховувати біти переносів для всіх бітів, які додаються, окрім першого. Комбінаційна схема, яка реалізує додавання лише для двох молодших бітів доданків, називається **напівсуматором**, а схема, яка реалізує додавання для всіх інших бітів доданків, називається **суматором**, інколи використовуються також назви **повний суматор**.

Таблиця 3.2 - Таблиця істинності напівсуматора

a	b	$a+b$	p	Σ
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	10	1	0

Введемо наступне позначення. Нехай a і b – біти доданків, які беруть участь в операції, Σ – біт результату, а p – біт переносу в наступний розряд. Опираючись на правила додавання двійкових кодів, отримаємо, що робота напівсуматора може бути описана табл. 3.2. Видно, що для біта переносу справедливо співвідношення $p=a\wedge b$, а біт суми Σ отримується як результат операції "Виключне АБО", $\Sigma = a\vee b$. Схема напівсуматора повинна мати два входи, на які подаються біти, які додаються, a і b , і два виходи, на яких формуються біт суми і біт переносу. Ці міркування приводять до зображеної на рис. 3.9 схемі напівсуматора. Для її реалізації потрібно 14 транзисторів.

При додаванні кожної наступної пари бітів доданків необхідно враховувати біт переносу з попереднього розряду. Отже, ця операція залежить від трьох аргументів, а відповідна комбінаційна схема повинна мати три входи. В результаті додавання поточної пари бітів отримуємо біт поточного розряду суми і біт переносу в наступний розряд. Тому схема повинна мати два виходи. Нехай, як і раніше, a і b позначають біти доданків, а Σ – біт результату. Нехай далі p_{in} – біт переносу з попереднього розряду, а p_{out} – біт переносу в наступний розряд. Тоді додавання з врахуванням переносів можна описати в табл. 3.3.

Таблиця 3.3 – Таблиця істинності повного суматора

p_{in}	a	b	$a+b+p_{in}$	p_{out}	Кон'юнкти для p_{out}	Σ	Кон'юнкти для Σ
0	0	0	0	0	–	0	–
0	0	1	1	0	–	1	$\bar{p}_{in} \wedge \bar{a} \wedge b$
0	1	0	1	0	–	1	$\bar{p}_{in} \wedge a \wedge \bar{b}$
0	1	1	10	1	$\bar{p}_{in} \wedge a \wedge b$	0	–
1	0	0	1	0	–	1	$p_{in} \wedge \bar{a} \wedge \bar{b}$
1	0	1	10	1	$p_{in} \wedge \bar{a} \wedge b$	0	–
1	1	0	10	1	$p_{in} \wedge a \wedge \bar{b}$	0	–
1	1	1	11	1	$p_{in} \wedge a \wedge b$	1	$p_{in} \wedge a \wedge b$

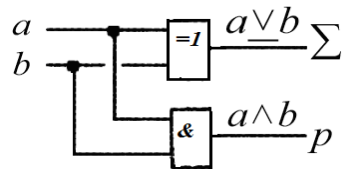


Рисунок 3.9 - Комбінаційна схема напівсуматора

Використаємо техніку побудови ДНФ для кожного з результуючих бітів операції. В табл. 3.3 наведені кон'юнкти лише для тих рядків, які беруть участь у побудові відповідної нормальної форми. Спочатку побудуємо вираз для біта Σ (дужки проставлені для спрощення сприйняття кон'юнктив, з яких створена ДНФ):

$$\Sigma = (\bar{p}_{in} \wedge \bar{a} \wedge b) \vee (\bar{p}_{in} \wedge a \wedge \bar{b}) \vee (p_{in} \wedge \bar{a} \wedge \bar{b}) \vee (p_{in} \wedge a \wedge b).$$

Виконуючи елементарні перетворення, отримаємо:

$$\Sigma = \bar{p}_{in} \wedge ((\bar{a} \wedge b) \vee (a \wedge \bar{b})) \vee p_{in} \wedge ((\bar{a} \wedge \bar{b}) \vee (a \wedge b)).$$

З врахуванням співвідношення $(p \vee q) \equiv \bar{p} \wedge q \vee p \wedge \bar{q}$ остаточно знаходимо зручний для реалізації у вигляді комбінаційної схеми вираз

$$\Sigma = (\bar{p}_{in} \wedge (a \vee b)) \vee (p_{in} \wedge \overline{(a \vee b)}) \equiv p_{in} \vee a \vee b.$$

Ця схема може бути побудована на базі двох послідовно з'єднаних вентилів "Виключне АБО". На входи першого вентиля слід подати біти, які додаються, а на входи другого вентиля – вихід з першого вентиля і біт переносу.

Побудуємо тепер ДНФ для біта переносу в наступний розряд:

$$P_{out} = (\bar{p}_{in} \wedge a \wedge b) \vee (p_{in} \wedge \bar{a} \wedge b) \vee (p_{in} \wedge a \wedge \bar{b}) \vee (p_{in} \wedge a \wedge b).$$

Групуємо перші і останні, а також другі й треті дужки і виносячи загальні множники, отримаємо зручний для реалізації у вигляді комбінаційної схеми вираз:

$$P_{out} = ((a \wedge b) \wedge (\bar{p}_{in} \wedge p_{in})) \vee p_{in} \wedge ((\bar{a} \wedge b) \vee (a \wedge \bar{b})) \equiv a \wedge b \wedge p_{in} \wedge (a \vee b).$$

Значення ab для останнього виразу можна отримати з виходу першого вентилі "Виключне АБО".

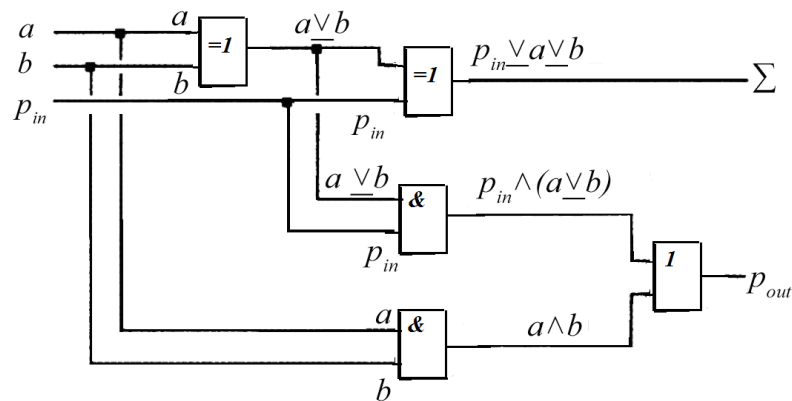


Рисунок 3.10 - Комбінаційна схема повного суматора

Комбінаційна схема повного суматора наведена на рис. 3.10. Відмітимо, що для її реалізації потрібен 31 транзистор.

3.2.10 Комбінаційна схема зсуву

Під час обговорення множення в двійковій системі числення виявилось, що ця операція, в принципі, зводиться до зсуву коду. Розрізняють декілька різновидів зсуву. Зсув коду вліво значить, що кожен його біт переміщується на сусідню зліва позицію, при цьому молодший (останній правий) розряд поля, який звільнився, заповнюється нулем, а останній лівий біт коду втрачається. Про такий біт говорять, що він виштовхується за розрядну сітку. Наприклад, зсув коду 00101101_2 вліво дає в результаті код 01011010_2 (рис. 3.11, а). Зсув вправо здійснюється у протилежному напрямку: кожен біт коду займає сусідній справа розряд, при цьому старший (останній лівий) розряд, який звільнився, заповнюється нулем, а молодший біт коду виштовхується за розрядну сітку, втрачається. Зсув того ж самого коду 00101101_2 вправо дає в результаті код 00010110_2 (рис. 3.11, б). Існують ще. так звані, *циклічні зсуви коду*, в яких біт, який виштовхується, не втрачається, а записується в розряд поля, який звільнився зліва або справа.

Уважний аналіз результатів звичайного зсуву двійкового коду показує, що зсув вліво є еквівалентом до множення на два, а зсув вправо – еквівалент до ділення на два без остачі. Так, в наведених раніше прикладах коду 00101101_2 , який розглядався як код в форматі з фіксованою точкою, якому відповідає число 45_{10} . Зсув цього коду вліво дає в результаті код 01011010_2 числа 90_{10} , а зсув вправо – код 00010110_2 числа 22_{10} . Для того, щоб отримати коректний результат у випадку множення, необхідно, щоб поле було достатньої для отриманого коду довжини, – точніше, щоб біти, які виштовхуються вліво, не були рівні одиниці.

У розглянутих ситуація зсув відбувався на одну позицію вправо або вліво. Має сенс розглянути також зсуви вправо і вліво на декілька позицій, що відповідає множенню або діленню на відповідну степінь двійки. Таким чином,

зсув коду може використовуватись не лише як допоміжна дія при реалізації загальної операції множення двійкових кодів, а і як самостійна операція множення або ділення націло на числа виду 2^n для цілих $n > 0$. Відмітимо, що операція зсуву виконується процесором комп'ютерів набагато швидше, ніж загальна операція множення або ділення.

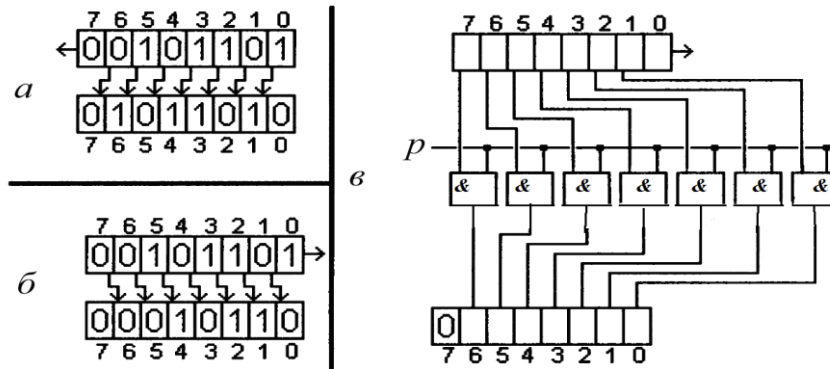


Рисунок 3.11- Зсув коду: *а* – вліво, *б* –вправо, *в* –спрощена схема зсуву вправо

На рис. 3.11, *в* представлена спрощена комбінаційна схема, яка реалізує зсув восьмого коду вправо на одну позицію. Кожному біту коду, який зсувається, окрім молодшого біта, який виштовхується, відповідає окремий вентиль "І". На рис. 3.11, *в* ці вентиля мають ті ж самі номери, що і відповідні їм біти коду. Кожен біт коду з'єднаний з входом відповідного йому вентиля, а на другий вхід кожного вентиля через єдину лінію надходить керуючий біт p . Якщо $p = 1$, то на виході кожного вентиля дублюється пов'язаний з ним біт коду. Ці виходи з'єднані з розрядами поля, які призначені для зберігання результату. Зверніть увагу: вихід кожного вентиля з'єднаний з розрядом поля, номер якого на одиницю менше, ніж номер вентиля, що, власне кажучи, і призводить до потрібного зсув коду. У крайньому зліва розряді, для якого немає відповідного вентиля, автоматично формується нуль.

Зсув коду вліво можна організувати за допомогою симетричної схеми; в ній виходи вентилів "І" з'єднані з розрядами полів, номери яких на одиницю більше, ніж номери відповідних вентилів. В більш загальних схемах обидва варіанта, які забезпечують зсуви вправо і вліво, з'єднуються в одну схему з одним єдиним керуючим бітом, який надходить в дві керуючі лінії, при чому в одну через вентиль "НІ". Таким чином, значення $p = 1$ запускає зсув в одну сторону, а значення $p = 0$ забезпечує зсув в іншу сторону. В такій схемі зсуву використовується $2n$ вентилів "І", де n – кількість розрядів, які зсуваються, і один вентиль "НІ". Відповідно, для її реалізації потрібно $7n + 1$ транзисторів.

3.2.11 Компаратор

Ще однією важливою операцією, яку, безумовно, повинен "вміти" виконувати процесор, є порівняння двох кодів на співпадіння. Для її реалізації

можна використовувати комбінаційну схему **компаратора**, восьмибітовий варіант якої зображений на рис. 3.12.

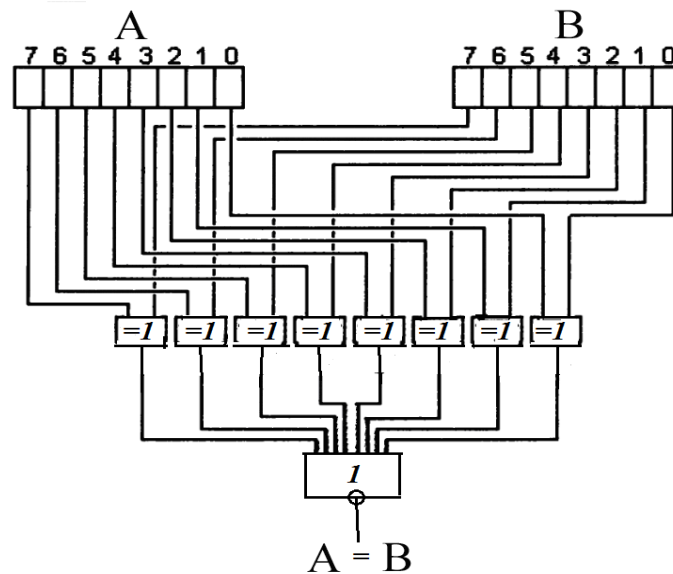


Рисунок 3.12 - Схема восьмибітового компаратора

Біти кодів для порівняння A і B, які мають один і той самий номер, приєднуються до входів одного і того ж самого вентиля "Виключне АБО". На рис. 3.12 вентилі зазначені номерами, які співпадають з номерами підключених до них бітів. Виходи всіх вентилів "Виключне АБО" приєднані до багатовходового вентиля "АБО НІ". Таким чином, якщо всі біти кодів A і B співпадають, то всі вентиля "Виключне АБО" сформуєть на своїх виходах значення 0, яке попадає потім на багатовходовий вентиль "АБО НІ". А якщо на його входах всі нулі, то на виході схеми сформується 1. А той самий час, якщо в кодах A і B не співпадає хоча б одна пара бітів, то відповідний вентиль "Виключне АБО" видасть на виході значення 1. Поява на входах вентиля "АБО НІ" хоча б одної одиниці призведе до формування на його виході значення 0. Таким чином, на виході компаратора завжди формується значення логічного виразу $A = B$. Відмітимо, що для реалізації компаратора, який здійснює порівняння двох n -бітових кодів, потрібно n вентилів "Виключне АБО" і один n -входовий вентиль "АБО НІ", відповідно потрібно $11n+2(n-1) = 13n-2$ транзисторів.

3.2.12 Перетворювачі кодів

Перетворювачі кодів призначені для перетворення чисел з однієї форми подання в іншу. Наприклад, при введенні інформації в ЕОМ необхідно перетворити десяткові числа в двійкові, а при виведенні інформації на індикатор чи друкуючий пристрій - двійкові або двійково-десяткові коди в коди управління знакогенератором, світлодіодами або рідиннокристалічними індикаторними панелями, механізмом друку. В якості прикладу побудуємо перетворювач коду

8421 в код семисегментного індикатора. Якщо сегменти позначені літерами, як показано на рис. 3.13, то табл. 3.4 встановлює відповідність між двійково-десятковим числом і потрібними для відображення десяткової цифри набором сегментів.

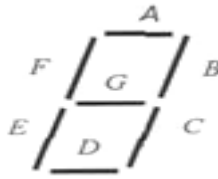


Рисунок 3.13 - Семисегментний індикатор

Таблиця 3.4- Таблиця відповідності

Десяткове число	Код 8421				Семисегментний код						
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>
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

Сегмент *A* визначається наборами коду 8421 наступним чином:

A =

ba \ dc	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	*	*	*	*
10	1	1	*	*

$$d + b + ac + \overline{ac} = \overline{ac} * \overline{ac} * \overline{b} * \overline{d}$$

Аналогічно отримаємо булеві вирази для інших сегментів:

$$\begin{aligned}
 B &= ab + \bar{a}\bar{b} + \bar{c} + \bar{d} = \overline{ab * \bar{a}\bar{b} * c * d}; \\
 C &= a + \bar{b} + c + d = \overline{a * b * \bar{c} * \bar{d}}; \\
 D &= \bar{a}\bar{b} + \bar{b}\bar{c} + \bar{a}\bar{c} + \bar{a}bc + \bar{d} = \overline{\bar{a}\bar{b} * \bar{b}\bar{c} * \bar{a}\bar{c} * \bar{a}bc * d}; \\
 E &= \bar{a}\bar{b} + \bar{a}\bar{c} + \bar{a}d = \overline{\bar{a}\bar{b} * \bar{a}\bar{c} * \bar{a}d}; \\
 F &= \bar{a}\bar{b} + \bar{a}\bar{c} + \bar{b}c + d = \overline{\bar{a}\bar{b} * \bar{a}\bar{c} * \bar{b}c * \bar{d}}; \\
 G &= \bar{a}\bar{b} + \bar{b}\bar{c} + \bar{b}c + d = \overline{\bar{a}\bar{b} * \bar{b}\bar{c} * \bar{b}c * \bar{d}}.
 \end{aligned}$$

Відповідно тотожним виразам на рис. 3.14 поданий один з можливих варіантів перетворювача на елементах І-НІ.

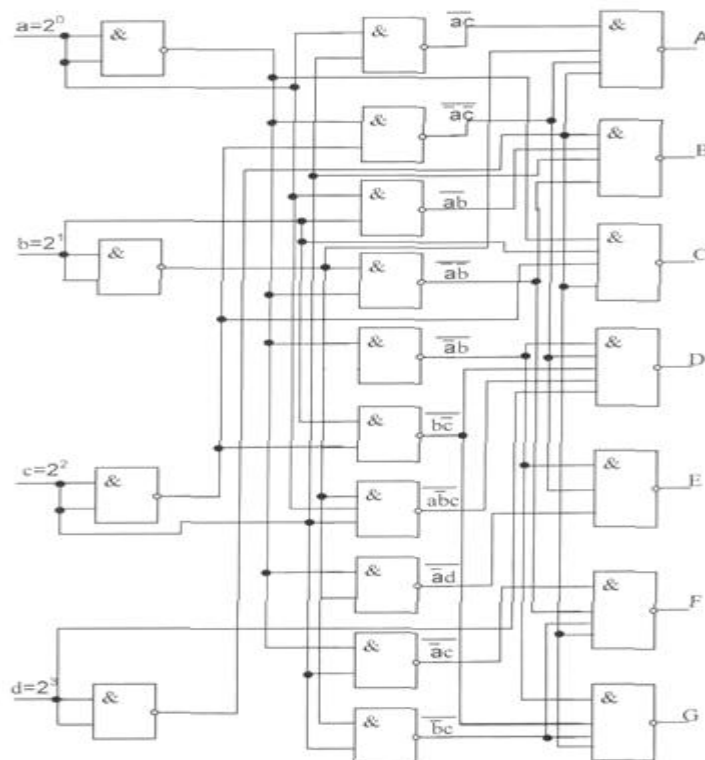


Рисунок 3.14 - Перетворювач коду 8421 в код семисегментного індикатора

Одним з випадків перетворювача кодів є **шифратор (кодер)** - пристрій, що забезпечує видачу певного коду в відповідь на збудження одного з входів. Зворотне перетворення двійкового коду в код «1 з n» виконують перетворювачі коду, що називаються **дешифраторами (декодерами)**.

Виконання над двійковими кодами таких операцій, як диз'юнкція, кон'юнкція, додавання і т. д., доцільно здійснювати за допомогою лише однієї комбінаційної схеми, яка забезпечує можливість не лише виконання, але і вибору потрібної операції. Вибір одного з декількох варіантів виконується за допомогою схеми, яка називається **декодером**.

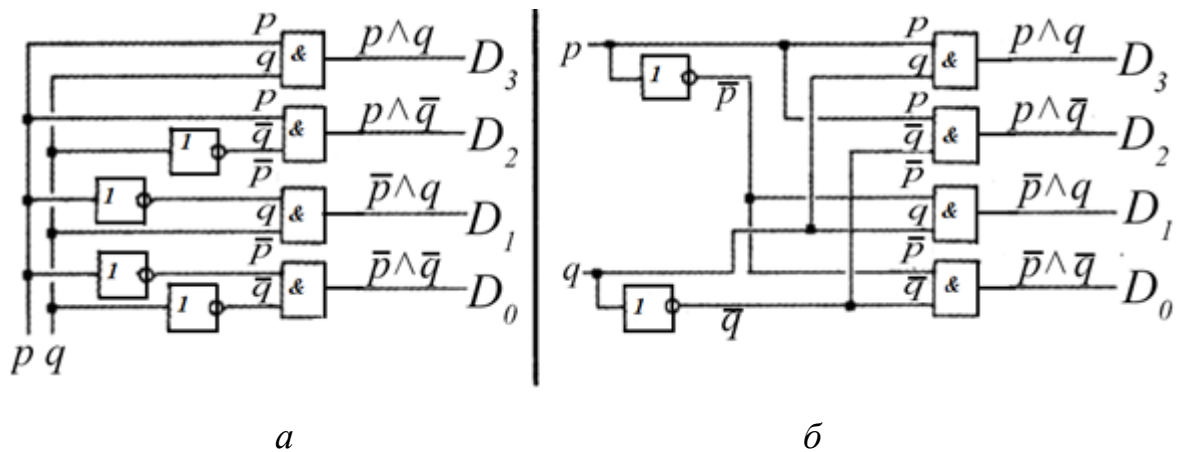


Рисунок 3.15 - Схема двовходового декодера

В загальному випадку в декодері за кожним з розглянутих варіантів закріплюється n -розрядний двійковий код, який надходить на вхідні лінії. Очевидно, що за його допомогою можна закодувати $2n$ різних варіанти. Кожному з них в схемі відповідає окремий n -входовий вентиль "І", входи якого з'єднуються з входами схеми або напряму, або через вентиль "НІ" (рис. 3.15, *a*). Таким чином, всі входи окремо взятого вентиля "І" в сукупності відповідають кон'юнкту, який залежить від n аргументів. Для будь-якого двійкового коду, який надійшов на вхід системи, лише для одного з вентилів "І" значення кон'юнкту дорівнює 1, і, відповідно, лише на його виході формується 1. На виходах всіх інших вентилів формується 0. Поява на одному з виходів схеми 1 трактується як вибір варіанту, який відповідає керуючому вхідному коду.

В якості прикладу розглянемо двохвходовий декодер, зображений на рис. 3.15, *a*. Цей декодер керується двохбітовим кодом, який забезпечує вибір одного з чотирьох варіантів. В схемі використані наступні позначення: p і q – входи, які приймають керуючий код, а D_0 – D_3 – вибрані виходи. Закріпимо за виходами схеми керуючі коди: $00_2 \rightarrow D_0$, $01_2 \rightarrow D_1$, $10_2 \rightarrow D_2$, $11_2 \rightarrow D_3$, – і складемо кон'юнкти, які визначають під'єднання до вентилів входи системи або їх заперечення. В результаті за кожним вентилям закріплюється один з чотирьох можливих кон'юнктив. На рис. 3.15 ці кон'юнкти наведені над відповідними їм виходами системи. При подачі на входи схеми будь-якого з двохбітових кодів лише один з вентилів сформує на своєму виході 1. Саме цей вихід вважається вибраним. Нехай, наприклад, $p=0$ і $q=1$. Тоді значення 1 має лише кон'юкт, якому відповідає вихід D_1 , а на всіх інших виходах формується 0. Таким чином, декодер за комбінаціями бітів 01_2 вибирає вихід D_1 .

Для створення n -входового декодера, зображеного на рис. 3.15, *a*, необхідно $2n$ вентилів "І" і в 2 рази менше вентилів "НІ", – відповідно, всього потрібно $3 \cdot 2^n + 2^{n-1} = 7 \cdot 2^{n-1}$ транзисторів. На рис. 3.15, *б* зображена еквівалентна схема тобто схема, яка описується такою ж самою таблицею істинності, що і схема на рис. 3.15, *a*. Видно, що для реалізації правої схеми потрібно менше вентилів "НІ", ніж для реалізації лівої. Це пов'язано з тим, що заперечення одного і того ж входу схеми знімаються з однієї і тієї ж лінії, а не формуються окремо для кожного вентиля "І". Для створення такої схеми необхідно всього n вентилів "НІ", отже, на схему потрібно лише $3 \cdot 2^n + n$ транзисторів. Наведений

приклад показує, що однієї і тієї ж цілі можна досягти за допомогою різних цифрових схем, які мають різні потреби до ресурсів, зокрема, до кількості потрібних для їх реалізацій вентилів.

Задача вибору одного з декількох вхідних значень і передачі його на вихід схеми вирішується схожим чином за допомогою пристрою, який називається **мультиплексором**. В загальному випадку мультиплексор містить 2^n основних вхідних ліній, n вхідних ліній керування і один вихід. Схема складається з 2^n багатовходових входів "І" і одного також багатовходового вентиля "АБО". В якості прикладу на рис. 3.16, а наведена схема чотирьохвходового мультиплексора, здатного вибрати один з чотирьох вхідних бітів і передати обраний біт на єдину вихідну лінію пристрою R . Для вибору потрібного варіанту в чотирьохвходовому мультиплексорі використовується дві лінії управління, p і q .

Схема працює наступним чином. Один з входів кожного вентиля "І" вважається основним. Він з'єднується з одним з входів системи, який вибирається. Кожен з n входів вентиля "І", які залишились, так само, як і в декодері, з'єднується з одною з керуючих ліній схеми або напряму, або через вентиль "НІ". Таким чином, ці входи вентиля "І" в сукупності відповідають кон'юнкту, який залежить від n аргументів. На рис. 3.16 кон'юнкти зображені над відповідною їм лінією основного входу вентиля. На керуючій лінії схеми подається n -розрядний двійковий код, на якому тільки один з кон'юнктив приймає єдине значення. На виході відповідного коду вентиля "І" дублюється біт його основної вхідної лінії, а на виходах всіх інших вентилів формується значення 0. Потім біт з вибраної вхідної лінії через вентиль "АБО" подається на вихід мультиплексора.

Ця схема може бути використана і в інших цілях. Зокрема, на базі мультиплексора можна створити схему, моделюючи будь-яку логічну функцію від n аргументів. Для цього потрібно побудувати таблицю істинності цієї функції і подати 1 на всі основні входи схеми, які відповідають кон'юнктам зі значенням 1, а на всі інші входи подати значення 0. Тоді при надходженні на керуючі входи схеми конкретних значень аргументів на її виході вийде значення логічної функції, яка моделюється, від цих аргументів.

Нехай, наприклад, логічна функція $R(p,q)$ задається таблицею істинності (табл. 3.5). В цій таблиці представлені також всі кон'юнкти і відповідні їм основні входи мультиплексора. З таблиці видно, що на входи D_0 і D_2 потрібно подати значення 1, а на входи D_1 і D_3 – значення 0 (рис. 3.16, б). Тоді при надходженні на лінії керування p і q значення аргументів на виході схеми сформується значення функції $R(p,q)$. Відмітимо, що на рис. 3.16, б спеціально не зображена внутрішня частина схеми мультиплексора, так як в даному випадку нас цікавлять лише її входи та виходи, її зовнішні лінії сполучення. Такий прийом досить часто використовується при аналізі складних схем, які складаються з декількох більш простих підсхем (рис. 3.21).

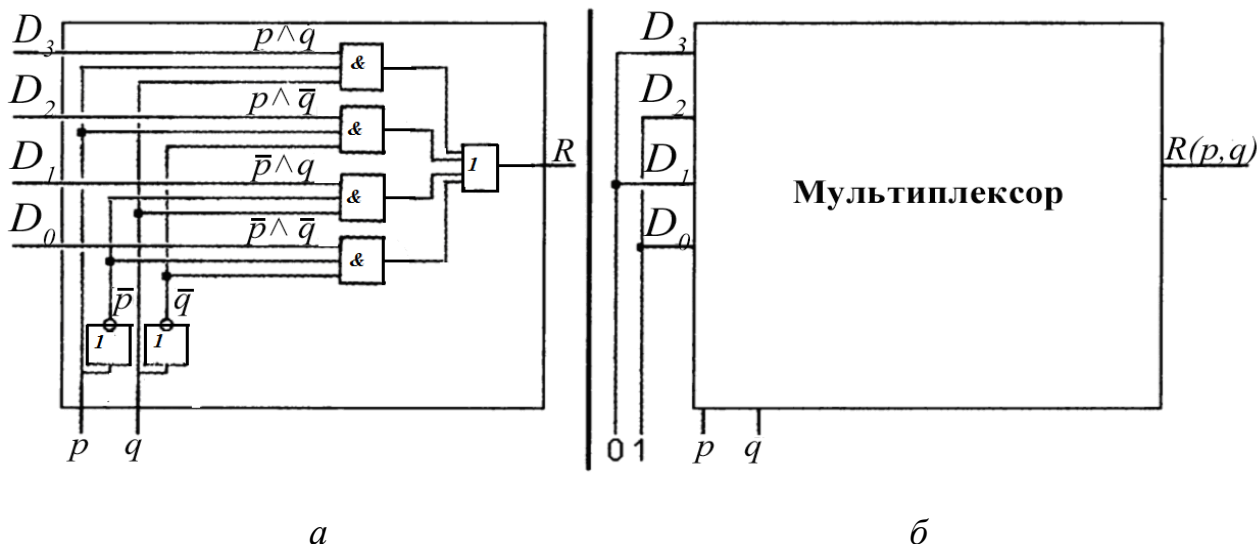


Рисунок 3.16 - Схеми: *a* – чотирьохвходового мультиплексора, *б* – яка реалізує логічну функцію $R(q, p)$

Таблиця 3.5 - Таблиця істинності функції

p	q	$R(p, q)$	Кон'юнкт	Вхід
0	0	1	$\bar{p} \wedge \bar{q}$	D_0
0	1	0	$\bar{p} \wedge q$	D_1
1	0	1	$p \wedge \bar{q}$	D_2
1	1	0	$p \wedge q$	D_3

Менш відоме в практиці застосування мультиплексорів [22] для побудови логічних схем у вигляді структур за результатами розкладення булевих функцій (БФ).

$$f(x_1, \dots, x_5) = \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 x_5 \vee \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 x_5 \vee \bar{x}_1 \bar{x}_2 x_3 x_4 \bar{x}_5 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \bar{x}_5 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \bar{x}_5 \vee \bar{x}_1 x_2 x_3 x_4 x_5 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \bar{x}_5$$

Для компактності представлення заданої булевої функції використовують десяткову форму запису з позначенням нею окремих термів (кон'юнкцій), представивши її у вигляді множини $\{R_k\}$:

$$f(X_1, \dots, X_5) = \{R_k\} = \sum(3, 5, 6, 8, 10, 15, 20)$$

З урахуванням специфіки роботи мультиплексорів і конструктивних особливостей їх реалізації з числом управляючих входів $g=2, 3, 4$ та інформаційних входів, рівним 2^g (4, 8, 16), розкладання заданої БФ можна вести за двома, трьома або чотирма змінними. Тоді при побудові логічної схеми на

мультиплексорах змінні повинні підключатись до управляючих входів а залишкові функції (ЗФ) розкладання - до інформаційних входів відповідного мультиплексора. Якщо утворені в результаті першого кроку ЗФ мають нетривіальний вид, то процедура розкладання кожної, що одержуємо на черговому кроці залишкові функції, повинна повторюватись до моменту перетворення їх у тривіальні, а саме:

$$\sum(0), \sum(1), \sum(0,1) \text{ } \emptyset \text{ (відсутня)}$$

При побудові логічної схеми на мультиплексорах, що реалізує задану БФ, можливі два випадки: а) $n \leq g$;

б) $n > g$.

В першому випадку БФ реалізується схемою, що складається з одного мультиплексора, в якому g змінних $(X_{n-g}, \dots, X_n)$ підключається до управляючих входів, а на інформаційні входи подаються константи 0 (якщо даний терм у функції відсутній) або 1 (якщо він присутній). В другому випадку процес побудови логічної схеми проводиться за результатами розкладання заданої БФ. Внаслідок першого кроку розкладання вихідної БФ по $f(X_1, \dots, X_n)$ по g змінним одержуємо сукупність ОФ, що залежать вже тільки від $n-g$ змінних. Наступні кроки розкладання зменшують кожний раз число змінних в ЗФ на g , аж до отримання в процесі розкладання ЗФ тривіального виду. Таким чином, число кроків розкладання БФ відповідає числу каскадів схеми на мультиплексорах з підключенням на управляючі входи мультиплексора тих змінних, по яких здійснювалось розкладання; на інформаційні входи мультиплексора останнього каскаду подаються окремі змінні x_i або $\bar{x}_i$, а також сигнали логічного 0 чи логічної 1, виходячи з виду отриманих ЗФ:

$$\sum(0) = \bar{x}_i; \sum(1) = x_i; \sum(0,1) = \bar{x}_i \cdot x_i; \emptyset = \log 0.$$

Згідно з приведеним вище алгоритмом здійснимо розкладання заданої БФ по двом, трьом та чотирьом змінним, зводячи результати розрахунків у таблиці.

Варіант розкладання БФ по двом змінним $\{X_4 X_5\}$ наведений у табл. 3.6.

Таблиця 3.6 - Результати розкладання БФ по двом змінним

$\{R_k\}$	3	5	6	8	10	15	20
$E\{R_k/4\}$	0	1	1	2	2	3	5
$F\{R_k/4\}$	3	1	2	0	2	3	0

Таким чином, на першому кроці розкладання БФ одержуємо наступні ЗФ:

$$Q_0 = \sum(2.5); \quad Q_1 = \sum(1); \quad Q_2 = \sum(1.2); \quad Q_3 = \sum(0.3);$$

Розкладання БФ продовжимо, так як не всі ЗФ мають тривіальний вид.

На другому кроці у вигляді вихідних даних розглядаються складові кожної з одержаних на першому кроці розкладання залишкові функції Q_i (табл.3.7).

Таблиця 3.7 - Результати розкладання БФ по двом змінним

Функція	Q_0		Q_1	Q_2		Q_3	
$\{Rk\}$	2	5	1	1	2	0	3
$E\{Rk/4\}$	0	1	0	0	0	0	0
$F\{Rk/4\}$	2	1	1	1	2	0	3

На другому кроці розкладання БФ маємо слідуєчи ЗФ:

$$\begin{aligned} \text{для } Q_0 : Q_0^1 &= \emptyset; & Q_1^1 &= \sum(1); & Q_2^1 &= \sum(0); & Q_3^1 &= \emptyset; \\ \text{для } Q_1 : Q_0^1 &= \emptyset; & Q_1^1 &= \sum(0); & Q_2^1 &= \emptyset; & Q_3^1 &= \emptyset; \\ \text{для } Q_2 : Q_0^1 &= \emptyset; & Q_1^1 &= \sum(0); & Q_2^1 &= \sum(0); & Q_3^1 &= \emptyset; \\ \text{для } Q_3 : Q_0^1 &= \sum(0); & Q_1^1 &= \emptyset; & Q_2^1 &= \emptyset; & Q_3^1 &= \sum(0); \end{aligned}$$

Оскільки ЗФ, отримані на другому кроці розкладання, є тривіальні, перейдемо до практичної реалізації двокаскадної схеми на мультиплексорах з $g=2$. Схемна реалізація БФ на мультиплексорах типу К1533КП2 наведена на рис. 3.17.

Варіант розкладання БФ по трьом змінним наведено в табл.3.8.

Таблиця 3.8 - Результати розкладання БФ по трьом змінним

$\{Rk\}$	3	5	6	8	10	15	20
$E\{Rk/8\}$	0	0	0	1	1	1	2
$F\{Rk/8\}$	3	5	6	0	2	7	4

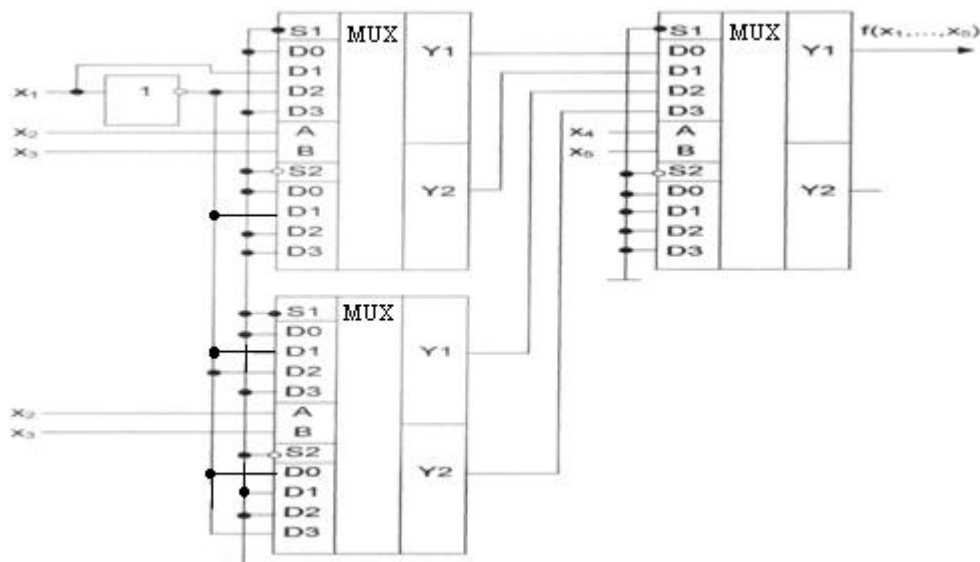


Рисунок 3.17 - Схемна реалізація БФ на мультиплексорах типу K1533КП2

Таким чином, після першого розкладання по трьом змінним отримано наступні ЗФ:

$$\begin{aligned} Q_0 &= \sum(1); & Q_1 &= \emptyset; & Q_2 &= \sum(1); & Q_3 &= \sum(0); \\ Q_4 &= \sum(2); & Q_5 &= \sum(0); & Q_6 &= \sum(0); & Q_7 &= \sum(1); \end{aligned}$$

Розкладання БФ продовжимо, так як не всі ЗФ мають тривіальний вид, результати наведені у табл.3.9.

Таблиця 3.9 - Результати розкладання БФ по одній змінній

Функція	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
$\{Rk\}$	1	$\emptyset$	1	0	2	0	0	1
$E\{Rk/2\}$	0	$\emptyset$	0	0	1	0	0	0
$F\{Rk/2\}$	1	$\emptyset$	1	0	0	0	0	1

На другому кроці розкладання БФ маємо слідуючи ЗФ:

$$\begin{aligned} \text{для } Q_0 : Q_0^1 &= \emptyset; & Q_1^1 &= \sum(0); & Q_2^1 &= Q_3^1 = Q_4^1 = Q_5^1 = Q_6^1 = Q_7^1 = \emptyset; \\ \text{для } Q_1 : Q_0^1 &= Q_1^1 = Q_2^1 = Q_3^1 = Q_4^1 = Q_5^1 = Q_6^1 = Q_7^1 = \emptyset; \\ \text{для } Q_2 : Q_0^1 &= \emptyset; & Q_1^1 &= \sum(0); & Q_2^1 &= Q_3^1 = Q_4^1 = Q_5^1 = Q_6^1 = Q_7^1 = \emptyset; \\ \text{для } Q_3 : Q_0^1 &= 0; & Q_1^1 &= Q_2^1 = Q_3^1 = Q_4^1 = Q_5^1 = Q_6^1 = Q_7^1 = \emptyset; \\ \text{для } Q_4 : Q_0^1 &= 1; & Q_1^1 &= Q_2^1 = Q_3^1 = Q_4^1 = Q_5^1 = Q_6^1 = Q_7^1 = \emptyset; \\ \text{для } Q_5 : Q_0^1 &= 0; & Q_1^1 &= Q_2^1 = Q_3^1 = Q_4^1 = Q_5^1 = Q_6^1 = Q_7^1 = \emptyset; \\ \text{для } Q_6 : Q_0^1 &= 0; & Q_1^1 &= Q_2^1 = Q_3^1 = Q_4^1 = Q_5^1 = Q_6^1 = Q_7^1 = \emptyset; \\ \text{для } Q_7 : Q_0^1 &= \emptyset; & Q_1^1 &= \sum(0); & Q_2^1 &= Q_3^1 = Q_4^1 = Q_5^1 = Q_6^1 = Q_7^1 = \emptyset; \end{aligned}$$

Схемна реалізація БФ наведена на рис.3.18

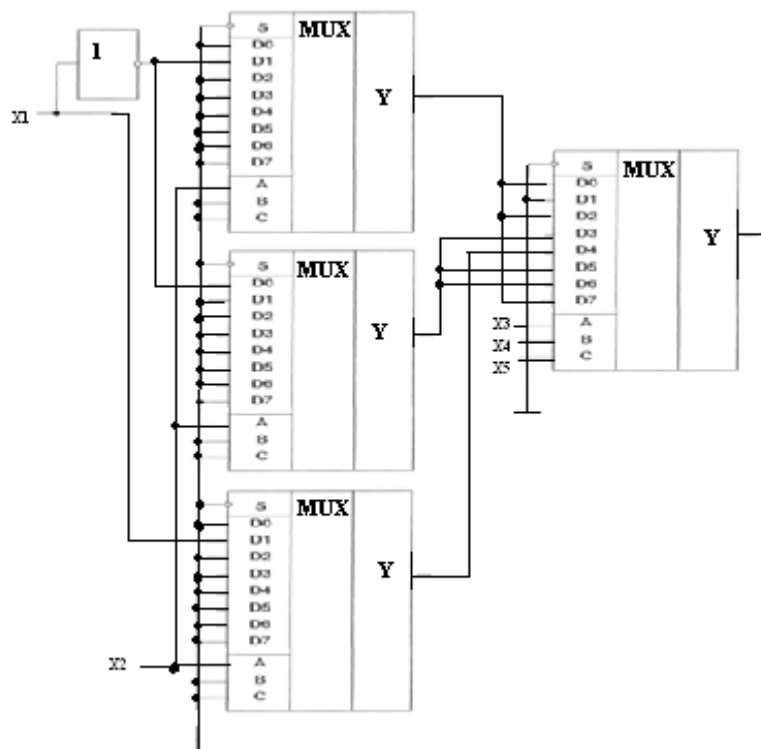


Рисунок 3.18 - Схемна реалізація БФ на мультиплексорах типу K1533КП7

Варіант розкладання БФ по чотирьом змінним $\{X_2, X_3, X_4, X_5\}$ наведено в табл.3.10.

Таблиця 3.10 - Результати розкладання БФ по чотирьом змінним

$\{Rk\}$	3	5	6	8	10	15	20
$E\{Rk/16\}$	0	0	0	0	0	0	1
$F\{Rk/16\}$	3	5	6	8	10	15	4

Таким чином, після першого кроку розкладання БФ по чотирьом змінним отримано наступні ЗФ:

$$Q_0=Q_1=Q_2=Q_7=Q_9=Q_{11}=Q_{12}=Q_{13}=Q_{14}=\emptyset;$$

$$Q_3=Q_5=Q_6=Q_8=Q_{10}=Q_{15}=0;$$

$$Q_4=1.$$

Оскільки всі ЗФ тривіальні, розкладання БФ закінчуємо і її можна реалізувати на одному мультиплексорі при $g=4$. Схемна реалізація заданої БФ на одному мультиплексорі типу K155КП1 наведена на рис. 3.19.

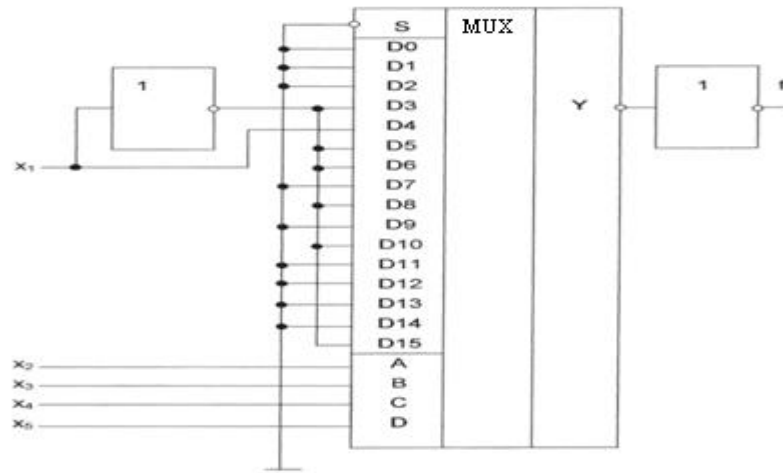


Рисунок 3.19 - Схемна реалізація БФ на одному мультиплексорі типу K155КП1

Для реалізації на мультиплексорах краще всього підходять БФ з кількістю змінних кон'юнкцій до 9.

Обговоримо ще одне важливе застосування схеми мультиплексора. Вона використовується для перетворення n бітів, які одночасно передаються різними лініями, в послідовність з n бітів, які передаються одне за одним по одній лінії (рис. 3.20). Таке перетворення доводиться виконувати, наприклад, при передачі даних від одного комп'ютера до іншого по лініям зв'язку в комп'ютерних мережах, оскільки всередині комп'ютера біти одного або декількох байтів зазвичай передаються між пристроями комп'ютера одночасно – паралельно, в той час як по зовнішнім лініям зв'язку дані, як правило, передають послідовно.

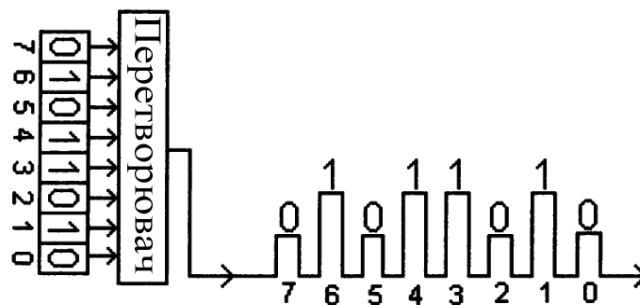


Рисунок 3.20 - Перетворення паралельного коду байта в послідовний

Одночасна передача по різним лініям декількох бітів називається **паралельною** передачею, а код, який передається в такий спосіб, прийнято називати паралельним кодом. Якщо біти коду передають послідовно, одне за одним по одній і тій ж лінії, то такий спосіб називається **послідовною** передачею, а код, який передається, називається послідовним кодом.

Для виконання такого перетворення потрібно під'єднати до основних входів мультиплексора всі лінії, по яким одночасно передаються біти. А на його керуючі входи подавати послідовність двійкових кодів, які здійснюють вибір основних входів ліній у бажаному порядку. Наприклад, подача на керуючі лінії чотирьохвходового мультиплексора коду 00_2 , тобто $p=0$ і $q=0$, призведе до

вибору основного входу D_0 і передачі біта, який знаходиться на ньому, на вихід схеми. Якщо трохи пізніше подати на входи керуючий код 01_2 , то на вихід попаде біт з основного входу D_1 . Наступна подача кодів 10_2 і 11_2 передасть на вихід біти спочатку з входу D_2 , а потім і з входу D_3 . Таким чином, код, який передається паралельно, виявиться перетвореним в код, який передається послідовно. Потрібно лише своєчасно фіксувати або ж передавати далі біти, які послідовно попадають на вихід схеми.

3.2.13 Арифметично – логічний пристрій

Частина процесора, яка виконує арифметичні, логічні та інші операції над даними, називається **арифметично–логічним пристроєм (АЛП)**.

Зазвичай АЛП, які забезпечують виконання дій над n -розрядними даними, складаються з n однакових схем, які виконують ці дії над двома бітами. Такі схеми називаються **однорозрядними АЛП**. На рис. 3.21 наведена спрощена схема однорозрядного АЛП, який може виконувати логічні операції заперечення, кон'юнкції, диз'юнкції і арифметичного додавання над двома бітами даних.

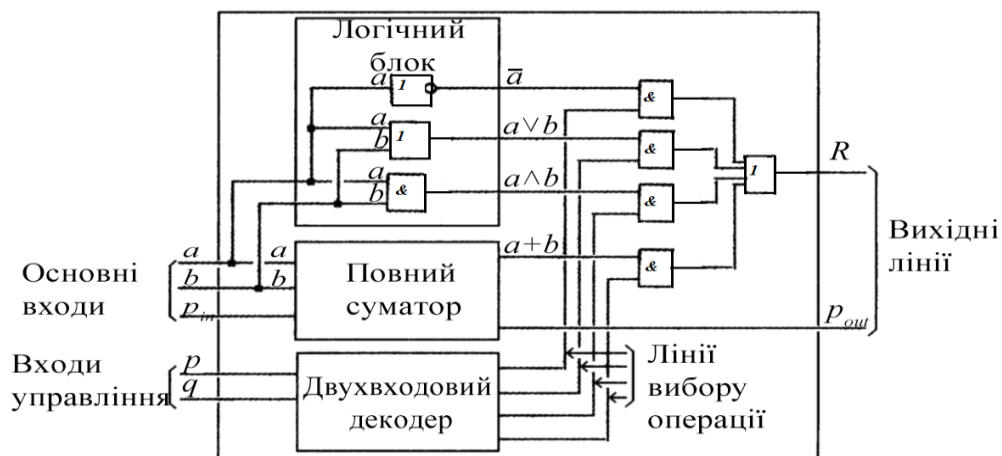


Рисунок 3.21- Спрощена схема однорозрядного АЛП

У схему по трьом основним вхідним лініям надходять два біти даних, a і b , а також біт переносу з попереднього розряду p_{in} . Крім того, в схему надходять два керуючих біти, p і q , значення яких визначають вибір бажаної операції. На виходах схема формує біт результату R і біт переносу в наступний розряд p_{out} . Однорозрядний АЛП містить блок виконання логічних операцій, який відповідає за операції заперечення, диз'юнкції і кон'юнкції, повний суматор, який відповідає за арифметичне додавання, і декодер, який організовує вибір потрібної операції.

На входи логічного блоку і повного суматора надходять біти даних a і b , а на вхід повного суматора ще і біт переносу p_{in} . Результати виконання операції одночасно формуються на трьох виходах вентилів "ІІ", "АБО", "І" логічного блоку і двох виходах суматора.

Біт переносу в наступний розряд p_{out} одразу попадає на вихід АЛП, а результуючі біти операції подаються спочатку в підсистему вибору потрібного результату. В цій підсистемі для кожної з чотирьох операцій, другий його вхід з'єднаний з вихідною лінією декодера. Декодер, отримавши на входах деяку комбінацію керуючих бітів p і q , формує значення 1 на виході, який відповідає цій комбінації. Таким чином, лише вентиль "І", який під'єднаний до цього виходу декодера, дублює на своєму виході результат обраної операції, інші вентиля формують на виході нульове значення. Щоб не організовувати декілька окремих вихідних ліній з АЛП, виходи всіх вентилів "І" з'єднані чотирьохвходовим вентилям "АБО", вихід якого є виходом R всієї схеми АЛП.

Простий підрахунок показує, що для реалізації описаного однорозрядного АЛП потрібно 67 транзисторів, а для аналогічного арифметико-логічного пристрою, який забезпечує дії над n -розрядними даними, 67 n -транзисторів. Наприклад, для шістнадцятибітного АЛП потрібно 1072 транзистора. Потрібно мати на увазі, що в навчальних цілях тут розглянутий значно спрощений варіант схеми, який відображає лише деякі принципи пристрою АЛП. Реальні АЛП процесорів можуть виконувати значно більшу кількість операцій, влаштовані більш складно і, звісно, вимагають набагато більшої кількості транзисторів.

3.3. Схеми пам'яті на базових вентилях

Схеми, які складаються з базових вентилів, застосовуються не лише для створення пристроїв, які виконують дії над даними. Вони використовуються також і для реалізації однієї з різновидів пам'яті в комп'ютері. Але схеми пам'яті не можуть бути віднесені до групи комбінаційних схем, оскільки результат, який отримуємо на їх виходах, залежить не лише від даних, які надійшли на вхід, а і від поточного стану схеми. Власне говорячи, ця залежність і забезпечує принципову можливість запам'ятовування даних [23].

Пристрій на базових вентилях, який можна використовувати для зберігання одного біта даних, називається **тригером**.

У загальному вигляді будь-який тригер можна розглядати як пристрій, що має кілька керуючих входів і два виходи. Виходи тригера позначають символами Q і $\bar{Q}$; їх називають відповідно прямим та інверсним. Напряга на інверсному виході завжди має значення, обернене логічному стану прямого.

На відповідні входи тригерів подають інформаційні і керуючі сигнали. **Інформаційними** називають сигнали, які треба запам'ятати. **Керуючі** – це ті сигнали, після подання яких інформація записується в тригер.

За способом запису інформації розрізняють нетактуючі (асинхронні) і тактуючі (синхронні) тригери. У нетактуючих тригерах перехід у новий стан спричинений змінами інформаційних вхідних сигналів. У тактуючих тригерах такий перехід відбувається тільки після подання спеціальних керуючих сигналів, які називають тактовими або синхронізуючими. Вхід тригера, на який надходять такі сигнали, називають тактовим і позначають літерою C .

За способом сприйняття тактових сигналів розрізняють тригери, які керуються рівнями і перепадами напруг. Керування рівнями означає, що при

одному рівні напруги на тактовому вході тригер перебуває в певному режимі (наприклад, зберігає раніше записану інформацію), а при другому – він переходить в інший режим (наприклад, змінює свій стан). Якщо тригер керується перепадами напруги, то такий дозвіл на запам'ятовування вхідної інформації тригером дається в момент перепаду тактового сигналу з одного рівня на другий. В інші моменти тригер не сприймає вхідні інформаційні сигнали. Тригери такого типу називають тригерами з динамічним керуванням.

Динамічний тактовий вхід може бути прямим або інверсним. У першому випадку тригер реагує на перепад тактового сигналу з нульового на одиничний, у другому – на перепад з одиничного значення на нульове.

Графічне зображення таких перепадів та умовні позначення динамічних тактових входів подано на рис. 3.22.

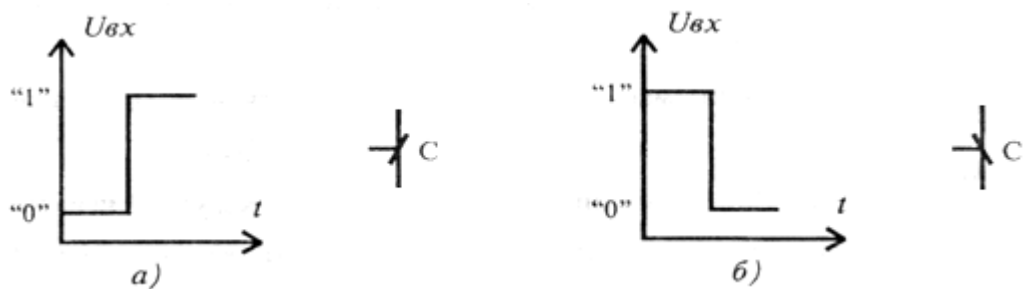


Рисунок 3.22 – Графічне зображення динамічних входів

Кількість і конкретне призначення входів тригера, а також форма керуючих сигналів залежать від типу тригера. Найбільш поширені такі основні типи: тригер з окремим запуском (*RS*-тригер), синхронний *RS*-тригер, з інформаційним входом (*D*-тригер), лічильний тригер (*T*-тригер), універсальний тригер (*YK*-тригер). Тактові тригери різних типів будують з керуванням за рівнем і за перепадом.

3.3.1 *RS* – тригери

RS-тригер має два керуючі входи: вхід встановлення в одиничний стан – *S*-вхід (від латинського *SET* – встановити) і вхід встановлення в нульовий стан – *R*-вхід (від латинського *RESET* – скинути). *RS*-тригер можна побудувати у двох варіантах, які показано на схемах (рис. 3.23).

Розглянемо роботу тригера, схему якого зображено на рисунку 3.22, а. Якщо схему тригера під'єднати до джерела, то на його прямому виході довільно встановиться який-небудь один з двох стійких станів. Нехай у початковий момент часу тригер перебуває в нульовому стані, а напруга на його керуючих входах відповідає за призначенням логічному «0». Якщо на *S*-вхід тригера надійде перепад напруг від значення, яке дорівнює «0», до значення, яке дорівнює «1», то тригер перейде в одиничний стан. При цьому кажуть, що в тригер записано «1».

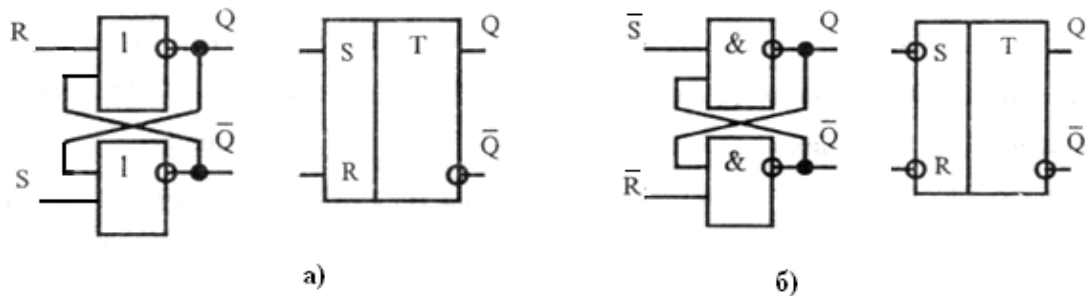


Рисунок 3.23 – Два варіанти побудови тригера

Якщо тригер перейшов в одиничний стан, то будь-які електричні сигнали, подані на S -вхід, не призведуть до зміни цього стану, якщо на R -вході, як і раніше, напруга відповідатиме логічному «0». При цьому кажуть, що тригер зберігає «1».

Щоб встановити RS -тригер у нульовий стан, якщо він перебуває в одиничному, треба подати перепад напруг з «0» в «1» на R -вхід і при цьому на S -вході підтримувати значення напруги, яке дорівнює логічному «0». У цьому разі в тригер буде записано «0».

Якщо тригер перебуває в нульовому стані, то всі наступні дії на R -вхід не змінять цього стану, якщо напруга на S -вході підтримується такою, що дорівнює логічному «0», то тригер зберігає «0».

Одночасне подання двох логічних одиниць на обидва керуючі входи заборонено, оскільки при цьому не визначено, в якому стані має бути тригер.

Робота другого RS -тригера (рисунок 3.23, б) відрізняється тільки тим, що керування досягається перепадами напруг з «1» в «0». Зазначимо також, що забороненим станом буде надходження на два входи одночасно нульових сигналів.

Роботу цифрових пристроїв дуже часто описують часовими діаграмами. **Часові діаграми** – це графіки напруг на всіх входах і виходах певного пристрою в різні моменти часу. Для зручності читання діаграм графіки зображають один під одним.

На рис. 3.24 подано часові діаграми роботи RS -тригера (рис. 3.22, а), які відповідають таким станам; вихідному, запису «1», зберігання «1», запису «0», зберігання «0».

Поряд із розглянутими RS -тригерами дуже поширені синхронні RS -тригери, в яких є додатковий тактуючий вхід C , що керується рівнем. Позначення таких тригерів подано на рис. 3.25.

Описані режими записування «1» і «0» використовуватимуться лише під час надходження на тактовий вхід C рівня напруги, яке відповідає значенню логічної «1». Наприклад, щоб записати в синхронний RS -тригер «1» (рис.3.25, а), треба на R -вході підтримувати напругу, яка дорівнює логічному «0», на S -вході встановити напругу, що дорівнює логічній «1» і подати «1» на C -вхід.

Щоб записати «1» в тригер, схему якого зображено на рис. 3.25, б, треба на R -вході підтримувати напругу, що дорівнює логічній «1», на S -вході встановити напругу, яка дорівнює логічному «0», і подати «1» на C -вхід.

На рисунку 3.26 зображено часові діаграми роботи синхронного RS -тригера, що відповідає позначенню, поданому на рисунку 3.25, а.

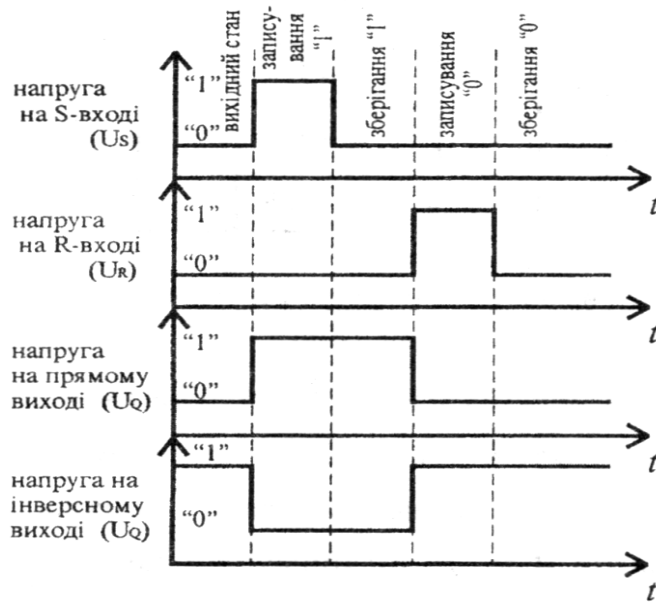


Рисунок 3.24 – Часові діаграми роботи *RS* – тригера

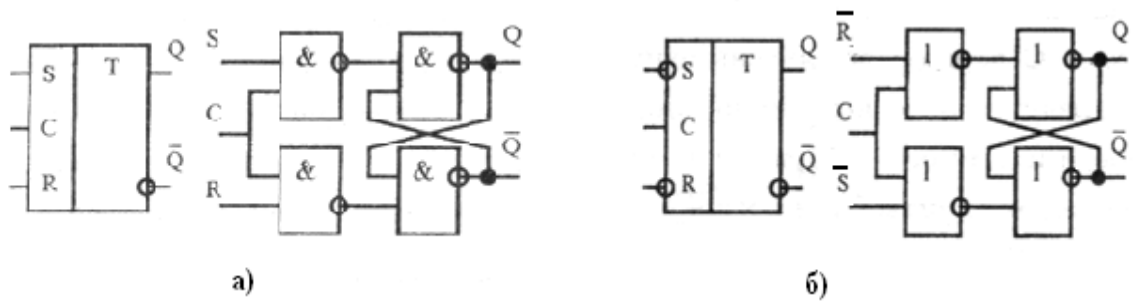


Рисунок 3.25 – Синхронні *RS* – тригери

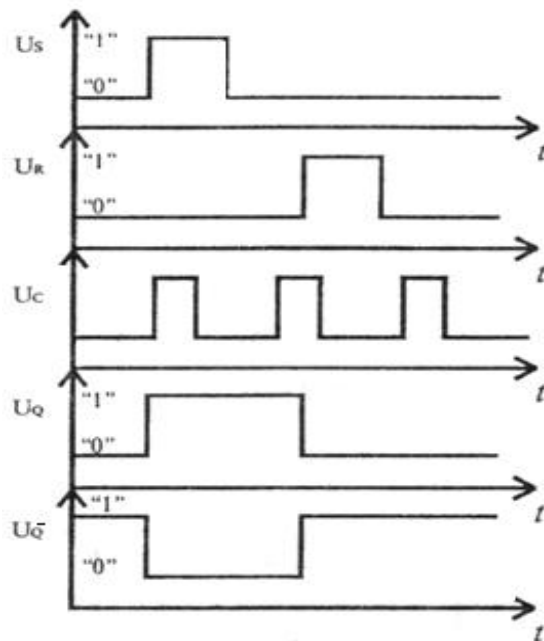


Рисунок 3.26 – Часові діаграми роботи синхронного *RS*–тригера

3.3.2 D–тригери

Тригер має два входи – інформаційний D і тактовий C . На рисунку 3.27 подано умовне позначення D – тригера з керуванням за рівнем напруги на тактуючому вході C .

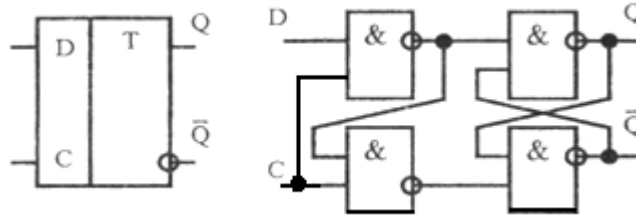


Рисунок 3.27 – D –тригер

Сигнал на виході Q інформаційного тригера повторює за значенням сигнал, поданий на інформаційний вхід D , якщо на тактовому вході C встановлено рівень напруги, що дорівнює «1». Якщо тактового сигналу не буде незалежно від значення напруги на D –вході, то тригер зберігає свій останній встановлений стан. Отже, якщо на D –вході напруга дорівнює за значенням логічній «1», то при надходженні «1» на C –вхід на виході Q запишеться «1», а якщо $D=0$, то і на виході Q напруга також дорівнюватиме за значенням логічному «0», якщо надійде «1» на тактовий вхід.

На рисунку 3.28 подано часові діаграми роботи D –тригера, який керується рівнем напруги на C –вході.

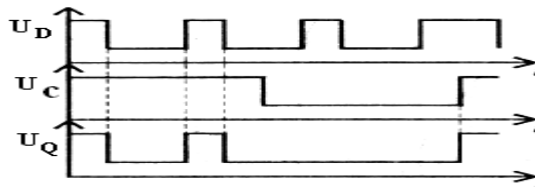


Рисунок 3.28 – Часові діаграми роботи D –тригера

Поряд з розглянутими поширені динамічні D –тригери, які керуються перепадом напруги на вході C . Крім входів D і C вони мають часто додаткові пріоритетні входи R і S , функції яких такі самі, як і в RS –тригері. На рис. 3.29 подано умовне позначення та часові діаграми роботи таких тригерів.

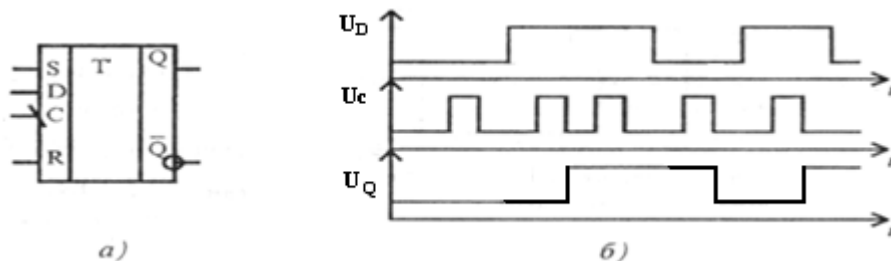


Рисунок 3.29 – Динамічний D –тригер

Тригер D -типу, якщо того вимагають обставини, можна використовувати як RS - , JK - , або як T -тригер. Це легко здійснити за допомогою додаткових зв'язків між входами і виходами або додаванням логічних елементів. На рис. 3.30, а показано, як D -тригер можна використовувати як асинхронний RS -тригер, але за умови, що мікросхема має установочні входи R і S , а імпульси на них надходять в різні моменти часу, на рис. 3.30, б показано, як слід включити два додаткові логічні елементи $DD1$ і $DD2$, аби перетворити D -тригер в синхронний $\bar{R}\bar{S}$ -тригер.

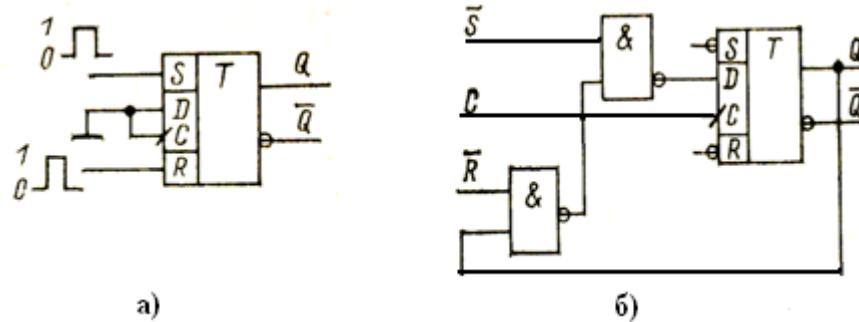


Рисунок 3.30 – Організація RS –тригера на основі D -тригера

3.3.3 Лічильні тригери (T -тригери)

Лічильні тригери широко застосовують для побудови електронних лічильників електричних імпульсів. Тому їх називають **лічильними**. Цей тип тригера має один керуючий (лічильний) вхід T . На рис. 3.31, а подано умовне позначення та часові діаграми роботи лічильного тригера.

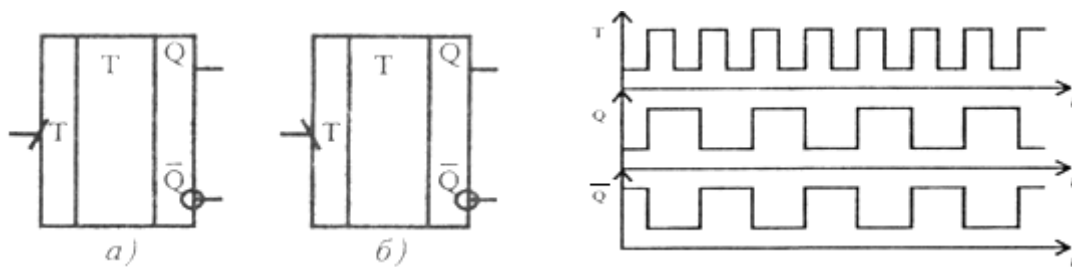


Рисунок 3.31 – Лічильні T -тригери

Лічильний тригер змінює свій стан на протилежний від кожного імпульсу, що надходить на вхід « T », а точніше, від перепаду напруги від «0» до «1» або від «1» до «0», залежно від різновидності тригера. Якщо T -тригер перебуває в нульовому стані, то він буде переведений в одиничний стан першим імпульсом, який надходить на його вхід. Другий імпульс знову поверне його в нульовий стан і т. д. У проміжках між імпульсами тригер зберігає свій останній стан.

Лічильний T -тригер має складну структуру. T -тригер можна побудувати, використавши динамічний D -тригер. Для цього треба інформаційний D -вхід тригера з'єднати з інверсним виходом $\bar{Q}$, а як лічильний вхід використати тактовий вхід C , як показано на рис. 3.32.

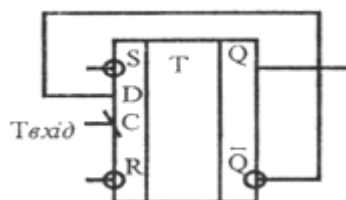


Рисунок 3.32 – Організація T -тригера на основі D -тригера.

Покажемо, що при такій будові тригера справді кожний імпульс, який приходить на вхід C (точніше перепад напруги з «0» в «1»), змінюватиме стан тригера на протилежний, що й відповідає лічильному режиму. Нехай початковий стан тригера $Q_1 = 1$, тоді $Q_1 = 0$. Значення напруги на вході D дорівнює напрузі на інверсному виході, оскільки вони з'єднані: $D = \bar{Q} = 0$. Якщо на вхід C надходить імпульс, то значення напруги на D -вході запишеться на прямий вихід, що впливає з принципу роботи D -тригера. Отже, тригер встановиться в нульовий стан, тобто змінить свій стан на протилежний.

3.3.4 Універсальні тригери (JK - тригери)

Універсальний тригер має два інформаційні входи J і K та один тактовий вхід C . Позначення JK -тригера та часові діаграми роботи подано на рис. 3.33.

R - і S -входи використовують для встановлення потрібного вихідного стану. Такий тригер можна використати як синхронний RS -тригер. При цьому вхід J відіграє роль входу S , а вхід K відповідає входу R .

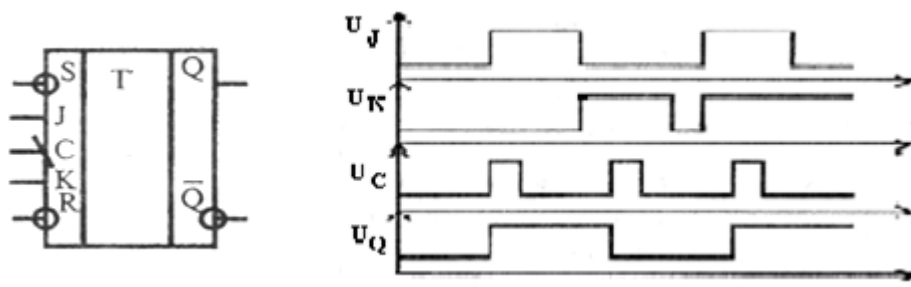


Рисунок 3.33 – Універсальний JK -тригер

Якщо на входи J і K подати напругу, яка відповідає за значенням логічній «1», то тригер почне працювати в лічильному режимі. Роль лічильного входу в цьому разі відіграє тактовий вхід C .

Дуже часто за наявності JK -тригера виникає необхідність використання його в іншій якості: як RS -, D - або T -тригер. Це легко забезпечити за рахунок додаткових зв'язків між входами і виходами або додаванням логічних елементів. На рис. 3. 34, а показано, як JK -тригер використовується в ролі RS -тригера, а на рис. 3.34, б показано спосіб використання JK -тригера як D -тригера. На рисунку 3.35, а, б – схеми T -тригера на основі JK -тригера.

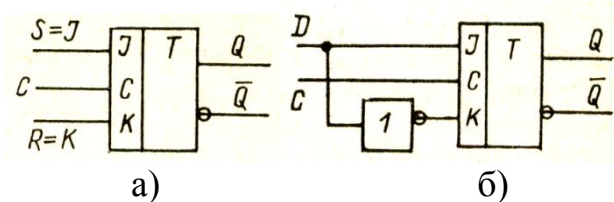


Рисунок 3.34 – JK –тригер в ролі RS (а) та D –тригера (б)

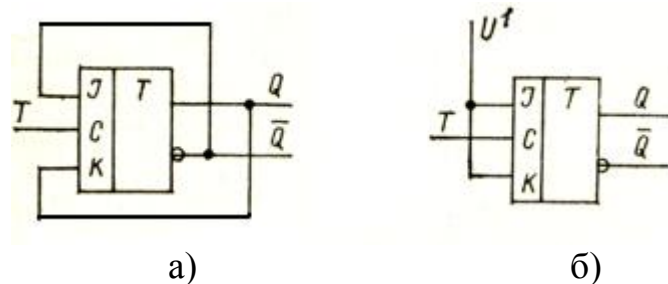


Рисунок 3.35 – T –тригер на основі JK –тригера

На рис. 3.36 показана організація JK –тригера на основі D –тригера

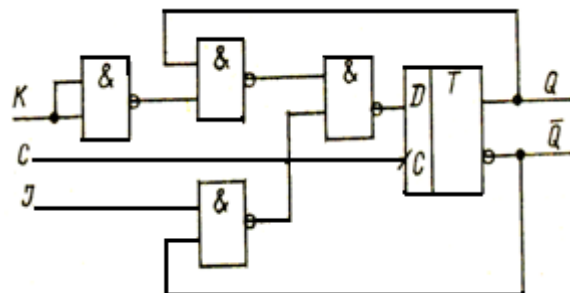


Рисунок 3.36 – Організація JK –тригера на основі D –тригера

3.4 Регістри

Регістри – це пристрої, призначені для приймання, зберігання, простих перетворень та передачі двійкових чисел. Під простими перетвореннями розуміють зсув чисел на задану кількість розрядів, а також перетворення послідовного двійкового коду в паралельний і паралельного в послідовний. Основне функціональне призначення регістрів - оперативна пам'ять для багаторозрядних двійкових чисел.

В залежності від способу прийому та передачі двійкової інформації розрізняють послідовні, паралельні та універсальні регістри.

В паралельних регістрах або регістрах пам'яті введення/виведення всіх розрядів числа проводиться одночасно за один такт. Для побудови n -розрядного регістра пам'яті потрібно n тригерів. Паралельні регістри служать основним функціональним елементом для побудови оперативних запам'ятовуючих пристроїв.

В послідовних регістрах введення/виведення інформації відбувається через один інформаційний вхід і один вихід зі зсувом числа. Тому послідовні регістри називають регістрами зсуву. За один такт інформація, що вводиться і виводиться зсувається на один розряд вправо або вліво. Регістри зсуву, що

реалізують по команді управління зсув інформації вправо або вліво, називають реверсивними.

Універсальні регістри мають можливості всіх типів регістрів і, крім того, забезпечують режим відключення входів та виходів (третій логічний стан) регістра від загальної інформаційної шини, перекомутацію входів та виходів регістра і тим самим перемикають функції приймання/передавання інформації в загальну інформаційну шину.

Технічні параметри регістрів визначаються параметрами їх основного функціонального вузла-тригера і розрядністю операнда. В якості прикладу розглянемо паралельний регістр (рис. 3.37 а) і регістр зсуву (рис. 3.37, б) на основі D -тригерів.

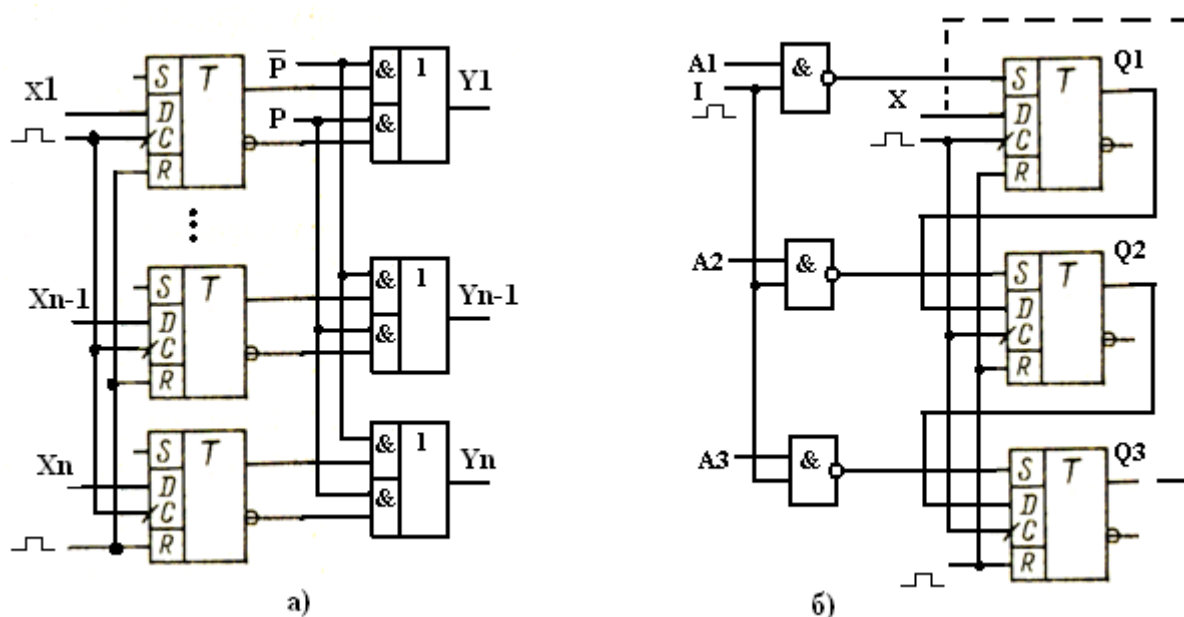


Рисунок 3.37 – Паралельний регістр (а) і регістр зсуву (б) на основі D -тригерів

У паралельному регістрі введення інформації відбувається на інтервалі синхронізації $C=I$. Вихідні логічні елементи формують вихідні сигнали відповідно виразу:

$$Y_i = \bar{P}Q_i + PQ_i, \quad (3.37)$$

тобто при $\bar{P}=1$ $Y_i=Q_i$ інформація з регістру видається в прямому коді, а при $P=1$ $Y_i=\bar{Q}_i$ – в зворотному.

Введення інформації в послідовний регістр може відбуватись послідовно із входу X з наступним зсувом інформації вправо на один розряд по кожному синхроімпульсу C або паралельно із входів A по сигналу $I=1$ асинхронного запису числа в попередньо очищений регістр ($R=1$). При закороченні виходів послідовного розряду з послідовним входом (штрихова лінія) реалізується кільцевий регістр зсуву, який зручно використовувати для побудови, наприклад, тактових генераторів цифрових пристроїв.

На рис. 3.38 зображена схема підключення універсального регістру K155IP13. Такий регістр може працювати в декількох режимах. Режим роботи регістру визначається кодом, поданим на керуючі входи регістру $M01, M02$.

Якщо на ці входи регістру буде подана комбінація символів "10" - регістр працює в режимі послідовного прийому інформації. При цьому на вхід *D* регістру послідовно подаються інформаційні символи, а на вхід *C* після кожного інформаційного символу подається тактовий імпульс. Запис кожного символу інформації здійснюється по перепаду з 0 в 1 тактового імпульсу. В такому режимі інформація направляється до виходу мікросхеми (зверху вниз, або зліва на право).

В режимі послідовного запису інформації код керуючого слова на входах *M01* і *M02* - "01". Інформація буде записуватися із зсувом від сьомого виходу до нульового (знизу вверху - справа наліво). Очистка регістру *K155IP13* здійснюється короткочасною подачею рівня логічного нуля на вхід *R*.

При паралельному запису інформації в регістр код керуючого слова - *M01* і *M02* - "11". Інформація по входах *D0-D7* записується в регістр при надходженні на вхід *C* перепаду напруги з 0 в 1. Часові діаграми роботи універсального регістру наведені на рис. 3.39.

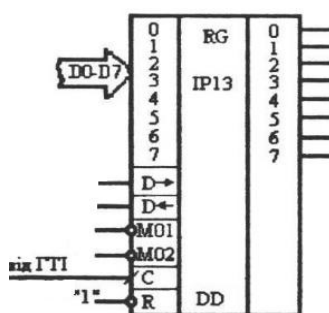


Рисунок 3.38 – Схема підключення універсального регістру

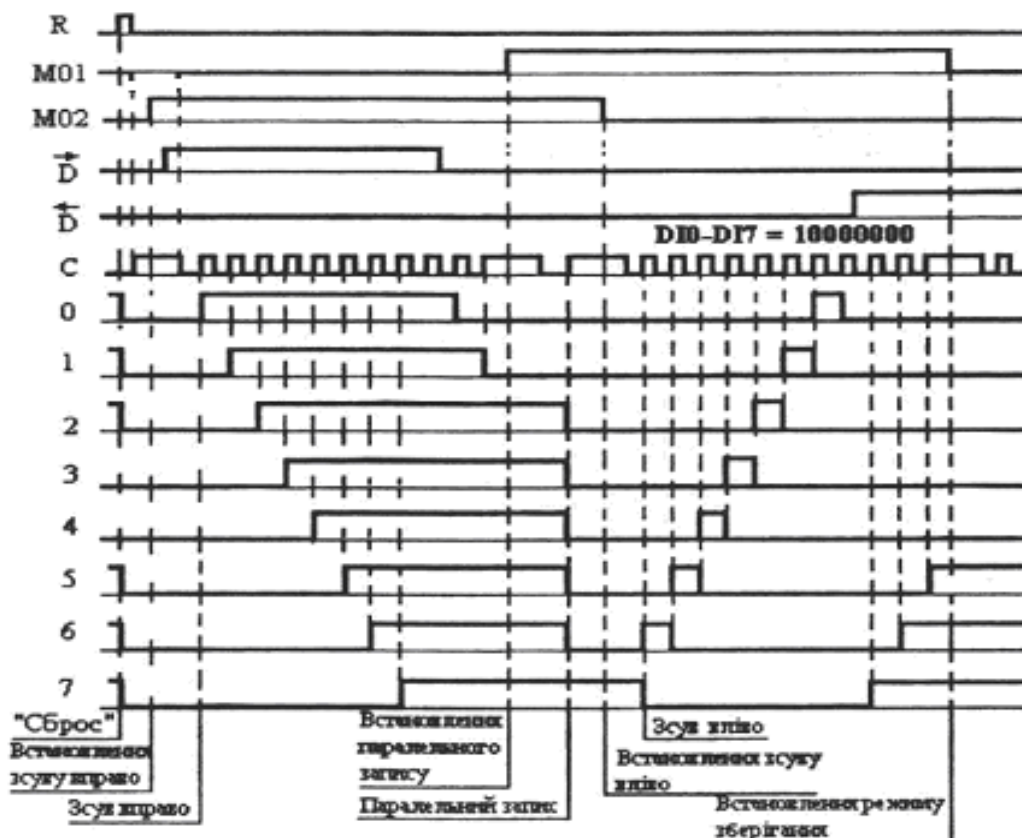


Рисунок 3.39 – Часові діаграми роботи універсального регістру

3.5 Лічильники

Лічильником називається цифровий пристрій, призначений для підрахунку та запам'ятовування кількості імпульсів, поданих в певному часовому інтервалі на його лічильний вхід. По характеру зміни стану лічильника вхідними імпульсами розрізняють на додавання, на віднімання та реверсивні лічильники. Кількість різних станів розрядів лічильника є його ознакою для класифікації, відповідно до якої лічильники називають двійковими, двійково-десятковими і т.п. Крім лічильного, вони можуть ще мати входи асинхронного або синхронного встановлення початкового стану.

В асинхронних лічильниках відсутня загальна для всіх розрядів синхронізація і перехід розрядів в новий стан відбувається послідовно розряд за розрядом (рис. 3.40).

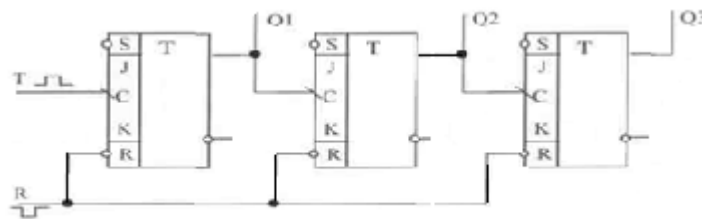


Рисунок 3.40 – Асинхронний лічильник

Головний недолік – низька швидкість, яка тим нижча, чим більше коефіцієнтів лічби. Одним із способів збільшення швидкодії асинхронних лічильників являється організація переходів між розрядами через додаткові логічні елементи (рис. 3.41).

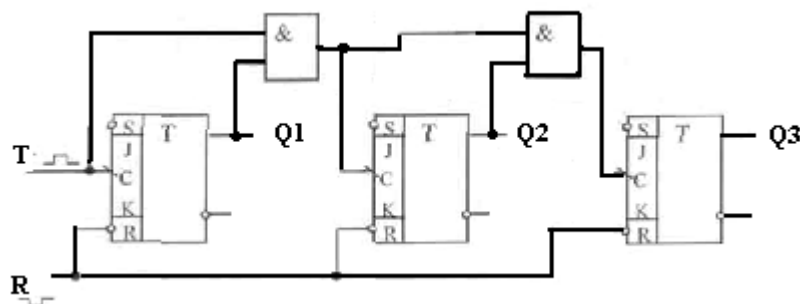


Рисунок 3.41 – Асинхронний лічильник з додатковими логічними елементами

Залежно від кількості розрядів N послідовні лічильники реалізують коефіцієнт лічби $K_{\text{ліч}}$, і їх можна використовувати в якості подільників частоти.

Часто при проектуванні цифрових пристроїв виникає необхідність в дільниках частоти, для яких $K_{\text{ліч}}$ – будь-яке ціле число. Якщо в арсеналі схемотехніка є найпростіші подільники частоти з коефіцієнтами рахунку $K_{\text{ліч}} = 2, 3, 4, 5, 6, 7, 8, 9, 10$, то велика кількість дільників частоти з $K_{\text{ліч}}$, що розкладається на найпростіші множники, будуються на їх основі. Необхідні

$K_{ліч}$ отримують введенням в лічильники зворотних зв'язків. Лічильники для $K_{ліч} = 2, 4, 8$ реалізуються без введення зворотних зв'язків включенням послідовно відповідно 1, 2 і 3 T -тригерів.

Загальний принцип побудови таких лічильників з непарним $K_{ліч}$ показаний на рис. 3.42. Якщо між вхідним і вихідним тригерами включений дільник частоти на будь-яке натуральне число n , то така схема забезпечує поділ частоти на

$$K_{ліч} = 2n + 1.$$

Наприклад, лічильник дільник частоти на 3, реалізований при $n = 1$, що відповідає прямому зв'язку між вхідним $DD1$ і вихідним $DD2$ тригерами. Для $K_{ліч} = 5$ відповідно $n = 2$, тому між вхідним і вихідним тригерами потрібно включити один додатковий тригер-дільник частоти на 2 і т.д.

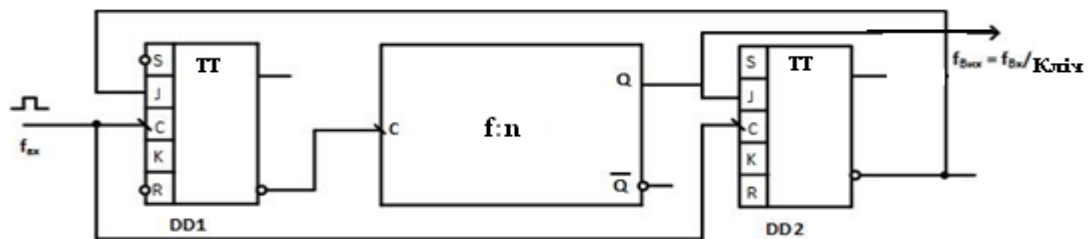


Рисунок 3.42 – Принцип побудови лічильників з непарним $K_{ліч}$

Схеми послідовних лічильників призначені в основному для використання в режимі поділу частоти без дешифрування їх станів, зміна яких не відповідає природному рахунку. Їх переваги – максимальна швидкодія, обмежена граничною частотою перемикавання вхідного тригера, при мінімальних витратах обладнання.

При побудові лічильників з дешифруванням станів необхідно забезпечити регулярність зміни станів згідно двійково-десятькового рахунку. Для їх реалізації необхідно визначити потрібну кількість тригерів N згідно співвідношенню

$$2^{N-1} \leq K_{ліч} \leq 2^N.$$

Крім послідовно включених N -тригерів такий лічильник повинен містити логічну схему, яка декодує стан $K_{ліч}$ і формує сигнал скидання всіх розрядів в стан «0». Приклад десятичного лічильника з регулярною зміною станів показаний на рис. 3.43. Кожен десятій імпульс переводить лічильник в стан 1010, на двох входах вентиля $DD5$ встановлюються рівні «1» і його вихідний сигнал $R = \overline{Q_2 Q_4} = 0$ скидає всі тригери в стан «0».

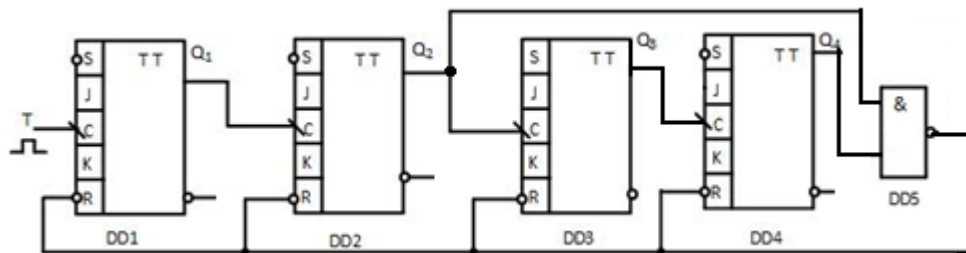


Рисунок 3.43 – Лічильник з дешифруванням станів $K_{\text{ліч}} = 10$

До синхронних лічильників відносяться лічильники, в яких перемикання розрядів відбувається одночасно, незалежно від віддаленості розряду від лічильного входу. Схема найпростішого синхронного лічильника на додавання наведена на рис. 3.44.

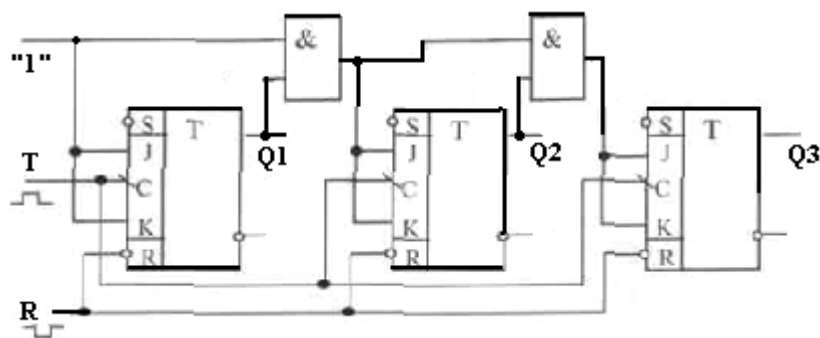


Рисунок 3.44 – Синхронний лічильник на додавання

Принцип побудови синхронного та асинхронного реверсивних лічильників аналогічний: прямий підрахунок реалізується при ввімкненні ланцюгів переносу з одиничних виходів тригерів молодших розрядів на входи старших розрядів, а зворотній рахунок (віднімання) – при вмиканні ланцюгів переносу між інверсними виходами тригерів молодших розрядів і входами старших розрядів.

3.6 Напівпровідникові запам'ятовуючі пристрої

Запам'ятовуючі пристрої (ЗП) служать для зберігання інформації та обміну нею з іншими частинами ЕОМ або мікропроцесорних систем. По функціональному призначенню ЗП розподіляються на зовнішні, буферні та внутрішні.

Зовнішні ЗП служать для зберігання великих об'ємів інформації та програмного забезпечення системи. В них використовуються ЗП на магнітних дисках.

Буферні ЗП призначені для проміжного зберігання даних при обміні між зовнішньою та внутрішньою пам'яттю.

Внутрішні ЗП по функціям діляться на оперативні та постійні.

Оперативні ЗП (ОЗП) виконують запис, зберігання та зчитування будь-якої інформації.

Постійні ЗП (ПЗП) виконують зберігання та видачу постійно записаної

інформації, склад якої не змінюється під час роботи системи. По способу занесення інформації ПЗП ділять на ПЗП, що програмуються на заводі-виробнику; програмовані ПЗП (ППЗП); програмовані однократно користувачем; перепрограмовані ПЗП (РПЗП); програмовані багатократно користувачем.

По способу зберігання інформації ОЗП ділять на **статичні та динамічні**. Запам'ятовуючі елементи статичних ЗП представляють собою бістабільні елементи і забезпечують зчитування інформації без її руйнування. В динамічних ЗП для зберігання інформації використовуються інерційні властивості реактивних елементів (конденсаторів), що потребує періодичної регенерації інформації. Класифікація ЗП наведена на рис. 3.45.



Рисунок 3.45 – Класифікація запам'ятовуючих пристроїв

Основними параметрами ЗП являються: інформаційна ємність (M) та швидкодія.

Інформаційна ємність характеризує кількість інформації, яка може зберігатися в запам'ятовуючому елементі, і визначається в бітах або кількості слів N із зазначенням їх розрядності n . Наприклад, 1×256 , 4×1024 , 8×1024 . Швидкодія характеризується часом вибірки та циклом запису. Час вибірки t_{ϕ} — інтервал часу між моментом подачі сигналу вибірки і появою інформації на виході мікросхеми ЗП. Цикл запису $t_{ц.з.}$ — мінімально допустимий час між моментом подачі сигналу вибірки при запису і моментом початку наступної операції зчитування (запису). Часові діаграми роботи ЗП представлені на рис.3.46, б.

По структурі напівпровідникові ЗП (рис.3.46, а) складаються з наступних типових вузлів: накопичувача НК; дешифратора стрічок та стовпчиків ДСХ, ДСУ; пристрою запису ПЗП; пристрою зчитування ПЗ; пристрою управління ПУ.

Накопичувач представляє собою матрицю запам'ятовуючих елементів, об'єднаних в стрічки та стовпчики через розв'язуючі ключові елементи, зв'язані дешифраторами.

В якості запам'ятовуючих елементів використовують діоди, біполярні транзистори МДН-структури, аморфні напівпровідники та ін.

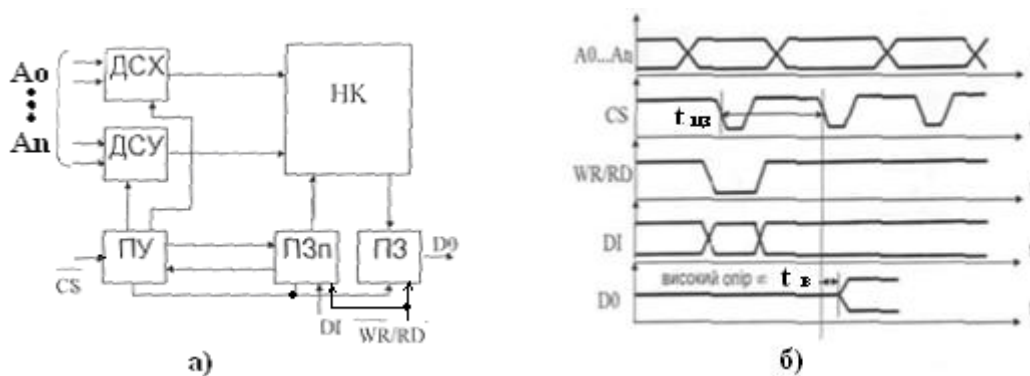


Рисунок 3.46 – Напівпровідникові запам'ятовуючі пристрої

На рис. 3.47 зображені схеми основних запам'ятовуючих елементів (ЗЕ), на основі яких розробляються ПЗП та ППЗП.

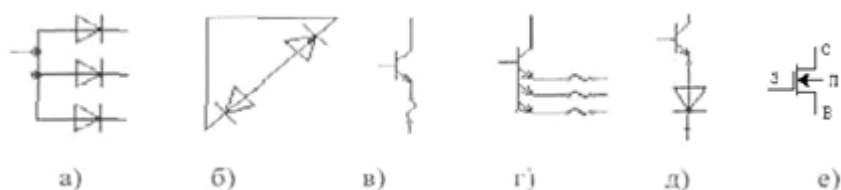


Рисунок 3.47 – Схеми основних запам'ятовуючих елементів

Для ПЗП з накопичуванням на ЗЕ (рис. 3.47, а, е) запис інформації відбувається двома способами: зміною конфігурації металізованої розводки (фотошаблон) або селективним покриттям контактних вікон під металізацію (лазерним променем). Для ППЗП ЗЕ (рис. 3.47, в, г) програмуються перепалюванням плавких перемичок, а ЗЕ (рис. 3.47, б, д) програмуються методом вибіркового руйнування діодів.

ППЗП відносяться до програмованих ІС двоступеневої структури, що складається з двох послідовних матриць. Матриця І жорстка, а матриця АБО програмована, а в випадку, коли обидві матриці програмовані, отримуємо **програмовану логічну матрицю**. Структура ПЛМ (рис. 3.48) складається з входних та вихідних буферних каскадів Бвх, Бвих і матриці елементів І та АБО (M_i , $M_{або}$). Вхідні буфери розвантажують вхідні ланцюги і перетворюють однофазні вхідні сигнали в парафазні.

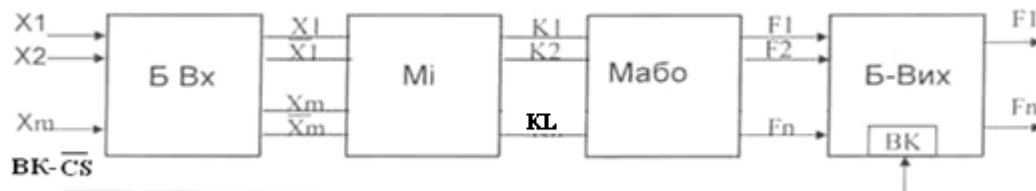


Рисунок 3.48 – Структура ПЛМ

Вихідні буфери забезпечують необхідну навантажувальну здатність ПЛМ і стробують її за допомогою входу вибірки кристала ВК, сигнал на якому або дозволяє роботу ПЛМ, або переводить виходи в стан «Вимкнено». Основними параметрами являється кількість входів m , кількість перехідних ланцюгів

(термів) L та кількість виходів N . Структура матриці І та АБО складається з горизонтальних та вертикальних шин, у вузлах перетинання яких знаходяться елементи зв'язку, які при програмуванні вводяться або знищуються (рис. 3.49, а). В якості елементів зв'язку можуть бути використані ЗЕ, наприклад діоди в матриці І (рис. 3.49, б) і транзистори в матриці АБО (рис. 3.49, в).

ПЛМ широко використовуються, як і ППЗП, для реалізації перемикаючих функцій F_n , перетворення кодів, так і в якості керуючої пам'яті ЕОМ з мікропроцесорним управлінням.

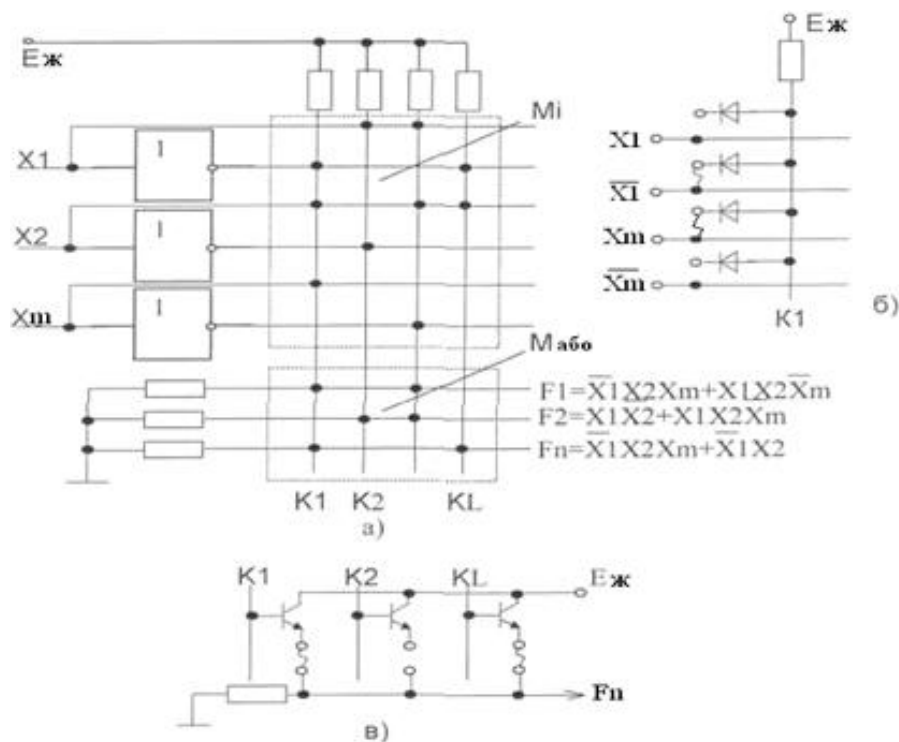


Рисунок 3.49 – Структура матриці І та АБО

4 СХЕМОТЕХНІКА НАЙПРОСТІШИХ ЦИФРОВИХ ПРИСТРОЇВ

4.1 Формувачі та генератори імпульсів

В системах автоматики робочі сигнали повинні мати певну амплітуду, тривалість, форму. Форма використаних сигналів може бути різною - прямокутна, синусоїдна, пилкоподібна, гострокінцева тощо [23, 24].

В цифровій техніці, яка є основою більшості пристроїв систем автоматики, найчастіше користуються сигналами, які являють собою поодинокі імпульси прямокутної форми, або послідовності таких імпульсів. Залежно від призначення, такі імпульси повинні мати точно визначені параметри (рис. 4.1). До параметрів імпульсів належать:

1. Амплітуда A_m прямокутного імпульсу, яка характеризується найбільшим значенням напруги або струму імпульсу;
 2. Тривалість τ прямокутного імпульсу;
 3. Період T прямокутного імпульсу (якщо діє послідовність імпульсів).
- Число імпульсів за одну секунду називають частотою повторення;

4. Коефіцієнт заповнення $K = \tau / T$ - відношення тривалості імпульсу до його періоду. Обернену величину називають скважністю:

$$C = 1/K = T/\tau. \quad (4.1)$$

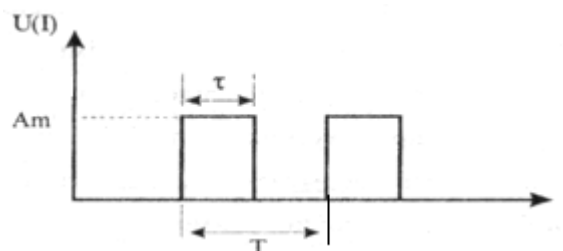


Рисунок 4.1 – Параметри імпульсів

За реальних умов отримати імпульс ідеальної прямокутної форми дуже важко. Імпульси, які виробляються в сучасних пристроях, близькі за формою до прямокутних (рис. 4.2). Для характеристики таких імпульсів вводять ще два параметри: тривалість фронту прямокутного імпульсу $\tau_{фн}$, що характеризує проміжок часу, за який амплітуда імпульсу збільшується від 0,1 A_m до 0,9 A_m ; тривалість зрізу прямокутного імпульсу $\tau_{зп}$ — проміжок часу, за який амплітуда імпульсу зменшується від 0,9 A_m до 0,1 A_m . При цьому під тривалістю прямокутного імпульсу τ звичайно розуміють інтервал часу, починаючи з якого амплітуда сигналу перевищує половину висоти A_m .

Імпульсні сигнали можна одержати двома способами: генеруванням і формуванням. Під час генерування прямокутних імпульсів енергія джерела живлення перетворюється в енергію послідовності прямокутних імпульсів.

При формуванні прямокутних імпульсів на вхід спеціальних пристроїв, які називають формувачами, подається вхідний сигнал - імпульси певної

форми (синусоїдні, прямокутні, або інших форм), а на їх виході утворюється прямокутний імпульс із заданими параметрами.

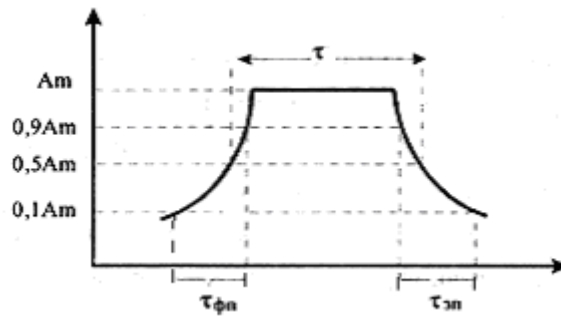


Рисунок 4.2 – Додаткові параметри імпульсів

Можливі і зворотні перетворення форми сигналу, при яких пристрої під дією прямокутних імпульсів формують синусоїдні коливання і т. ін.

Найчастіше в системах автоматики в якості формувачів застосовують одновібратори, схеми виділення перепадів з "0" в "1" і з "1" в "0".

4.1.1 Одновібратори

Одновібраторами називають пристрої, які під дією вхідного імпульсу формують вихідний імпульс певної тривалості. Вхідними звичайно є імпульси з електромеханічних або електронних ключів.

Для вияснення принципу роботи одновібратора розглянемо процеси, які протікають в пристроях з перехідною ємністю. Як відомо, логічні елементи мають так зване порогове (критичне) значення напруги на вході елемента, при якій він входить в активну область переключення. При збільшенні напруги на вході більше порогової величини, логічний елемент миттєво переходить у вихідний стан, який відповідає рівню логічної "1".

Якщо напруга на вході елемента нижче порогової, він переходить у вихідний стан, який відповідає рівню логічного "0" на його виході.

Для мікросхем ТТЛ-структури порогове значення напруги на вході складає $\approx 1,2 \text{ В.}$, для КМОН - $U_{жс} / 2$.

На рис. 4.3 зображена схема логічного елемента І-НЕ з перехідною ємністю (а) на вході і часові діаграми його роботи (б).

У початковому стані на виході елемента рівень логічної "1". Стрибок напруги на вході переводить елемент із стану логічної "1" в нульовий стан в зв'язку з тим, що при подачі напруги на опір Я виникає стрибок напруги, який відповідає рівню логічної "1". По мірі зарядження конденсатора С напруга на ньому збільшується, а на опорі R зменшується, і в той момент, коли напруга на опорі стане меншою критичної, схема повертається у вихідне становище.

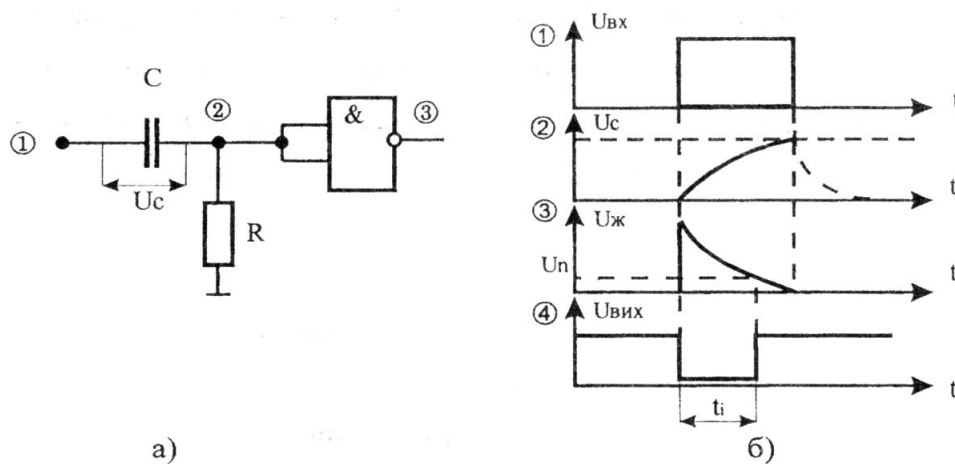


Рисунок 4.3 – Логічний елемент І-НЕ з перехідною ємністю

4.1.2 Одновібратори на основі ємнісного зв'язку

Робота одновібратора пояснюється на рис. 4.4. З часових діаграм видно, що у початковому стані на вході елемента DD1 – рівень логічної "1", на виході – рівень логічного "0". Вхід ключа 2 також має рівень логічного "0". (Для цього резистор R для мікросхем ТТЛ структури повинен мати величину 500 Ом – 1,8 КОм, а для мікросхем КМОН структури $> 20\text{кОм}$). При переході елемента DD1 в стан логічної "1" виникає струм заряду конденсатора C, на опорі R виникає стрибок напруги і в проміжку часу від t_1 до t_2 , коли напруга на опорі перевищує критичну, напруга на вході елемента DD2 буде мати стан логічного "0" і повернеться в початковий стан після розряду конденсатора C, або після переходу елемента DD1 в стан логічного "0".

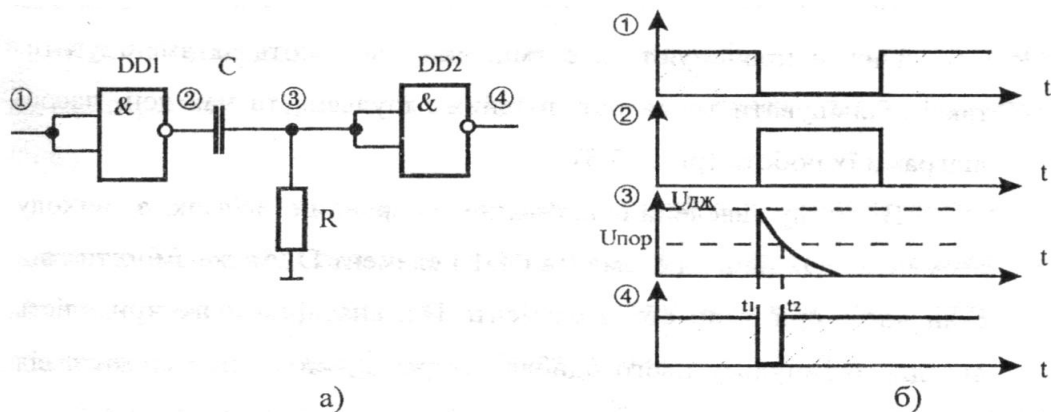


Рисунок 4.4 – Одновібратор на основі ємнісного зв'язку

Тривалість вихідного імпульсу в цій схемі $\tau_{вих} \approx RC$ за умови $\tau_{вих} > RC$ і $\tau_{вих} = t_{ex}$ якщо $\tau_{ex} < RC$.

4.1.3 Типові схеми одновібраторів

Нижче пропонуються схеми, які дозволяють як зменшувати, так і збільшувати тривалості вихідних імпульсів. В схему (рис.4.5) введений від'ємний обернений зв'язок і елемент DD1 може змінити свій стан лише при зміні стану елемента DD2, внаслідок чого тривалість вихідного імпульсу цього одновібратора $\tau_{вих} \approx RC$ і не залежить від тривалості вхідного імпульсу.

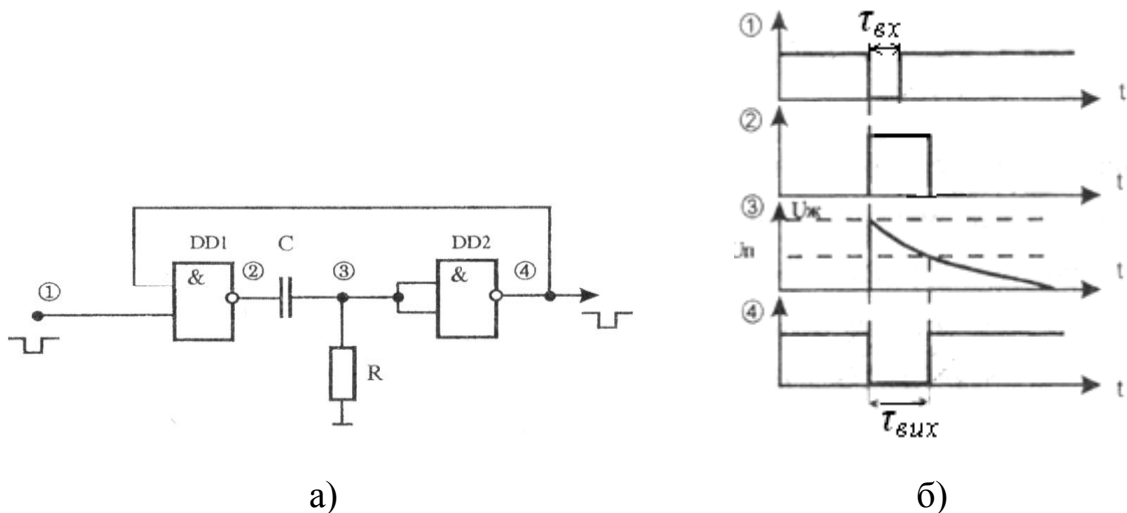


Рисунок 4.5 – Типовий одновібратор

Як видно з часових діаграм одновібратор запускається від'ємним імпульсом, на виході також одержується від'ємний імпульс, що має вигляд перепаду з "1" в "0" певної тривалості, яка залежить від параметрів R і C.

Для того, щоб одержати одновібратор, який запускається додатними імпульсами і на виході також одержати додатний імпульс, потрібно на вході і виході одновібратора поставити по інвертору. Але при цьому кількість логічних елементів в схемі збільшується вдвічі. Більш доцільним для одержання такого одновібратора є застосування логічних елементів АБО-НЕ (рис. 4.6).

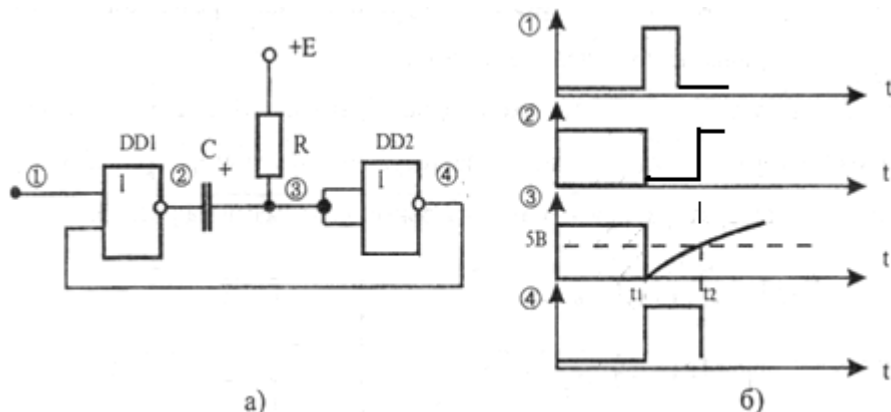


Рисунок 4.6 – Одновібратор з додатними вхідним та вихідним імпульсами

У початковому стані на вході 1 елемента $DD1$ рівень логічного "0". На вході елемента $DD2$ рівень логічної "1", що задається резистором R . Тому на виході логічного елемента $DD2$ рівень логічного "0". Слід відзначити, що напруга на ємності приблизно дорівнює нулю, оскільки з обох її сторін діє рівень логічної "1". При подачі на вхід 1 елемента $DD1$ пускового імпульсу елемент $DD1$ відкривається, напруга в точці 2 стає рівною нулю. Виникає зарядний струм по колу. $+E, R, C$, заземлення. Але спочатку напруга на ємності дорівнює нулю і з ростом її заряду зростає за експонентою. В проміжок часу від t_1 , до t_2 напруга на ємності (на вході 3 мікросхеми $DD2$) менша порогової, тому на виході $DD2$ (інвертор) буде рівень логічної одиниці. Після того, як напруга в точці 3 перевищить пороговий рівень і стане еквівалентною рівню логічної "1", елемент $DD2$ знову перейде в початковий стан, при якому на його виході встановиться рівень логічного "0". Тривалість імпульсів однобратора приблизно приймається рівною $\tau_{им} \approx 0,69RC$.

Це співвідношення використовується для визначення тривалості імпульсу більшості однобраторів, дія яких основана на процесі перезаряду ємності. Наближеність формули пов'язана з розкидом значень порогового рівня переключення мікросхем.

Існує велика кількість схем однобраторів. Цікавою є схема однобратора, побудованого на застосуванні RS - тригера (рис. 4.7, а), робота якого пояснюється часовими діаграмами (рис. 4.7, б).

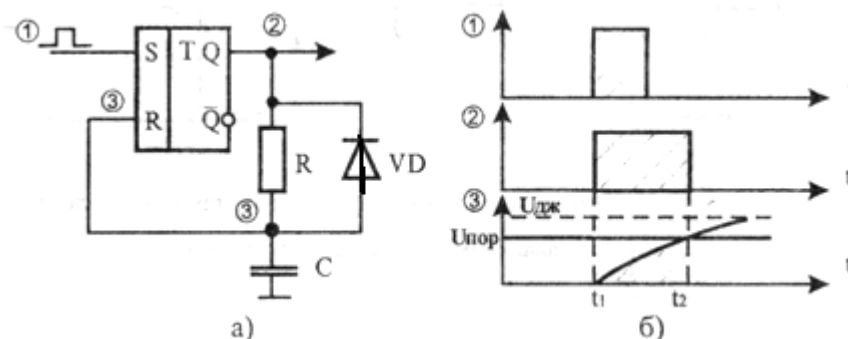


Рисунок 4.7– Однобратор на основі RS - тригера

В системах автоматики часто застосовують однобратори, в яких для одержання необхідної тривалості імпульсів використовують цифрові лічильники. Такі цифрові однобратори використовують в тих випадках, коли висувуються високі вимоги до стабільності часових інтервалів вихідних імпульсів однобраторів. В цьому випадку мінімальна тривалість імпульсу обмежується тільки швидкодією використовуваних елементів, а максимальна тривалість може бути довільною (на відміну від схем, які використовують RC -кола).

Принцип роботи цифрового однобратора (рис. 4.8) заснований на включенні тригера вхідним сигналом і скиданні його через часовий інтервал,

який визначається коефіцієнтом перерахунку лічильника (рис. 4.9). В початковому стані на прямому виході D -тригера рівень логічного "0", який закриває елемент $DD2$, і імпульси з постійно діючого генератора GN не проходять через елемент $DD2$ на лічильний вхід C лічильника $DD3$. При цьому на виході елемента $DD5$ - рівень логічного "0", який дає змогу тригеру $DD1$ зберігати нульовий стан. Після подачі на вхід одновібратора запускаючого імпульсу D -тригер переходить в одиничний стан, відкриває елемент $DD2$ і імпульси від генератора GN починають проходити на C -вхід лічильника.

В наведеній схемі тривалість імпульсу одновібратора дорівнює тривалості 10 періодів імпульсу генератора GN .

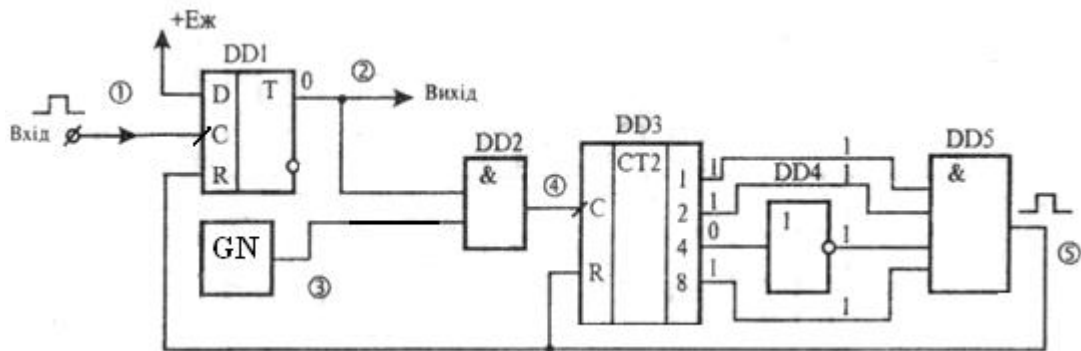


Рисунок 4.8 – Цифровий одновібратор

Після надходження на вхід C лічильника $DD3$ 11-го імпульсу на виході елемента $DD5$ виникає рівень логічної "1", який по входах R скидає тригер $DD1$ і лічильник $DD3$ в початковий стан.



Рисунок 4.9 – Часові діаграми цифрового одновібратора

Загальним недоліком цифрових одновібраторів є випадкова похибка, яка пов'язана з довільністю фази задаючого генератора в момент запуску схеми. Похибка зменшується з підвищенням частоти генератора і коефіцієнта перерахунку лічильника.

4.2 Схеми виділення перепадів

У системах автоматики досить часто застосовуються різновидності одновібраторів, які носять назву схеми перепадів імпульсів з "0" в "1" і з "1" в "0". В технічній літературі ці пристрої називають схемами виділення фронтів або "різницевиими перетворювачами" (РП).

Такі пристрої призначені для утворення вихідного сигналу, який несе інформацію про зміну значення вхідного сигналу. На виході схем виділення перепадів формуються короткочасні імпульсні сигнали у відповідь на відповідний перепад напруги.

Існують різні схеми виділення перепадів. Частина з них базується на основі застосування диференційних та інтегруючих RC -кіл, друга частина оснований на застосуванні деяких властивостей логічних елементів. Розглянемо принципи їх побудови.

4.2.1 Схеми виділення перепадів на основі диференційних кіл

Простими схемами виділення перепадів напруги є так звані диференційні кола, які дозволяють одержувати вихідні сигнали, амплітуда яких пропорційна крутизні вхідної напруги чи струму. Припустимо, що на вході кола, зображеного на рис. 4.10, діє імпульс напруги прямокутної форми з амплітудою U_{BX} . При цьому в момент перепаду напруги з "0" в "1" на виході RC -кола виникає гострокінцевий імпульс, який спадає за експоненційною кривою зі сталою часу RC . Цей імпульс має ту саму полярність, що і прямокутний вхідний імпульс. Після закінчення дії вхідного прямокутного імпульсу на виході кола з'явиться другий гострокінцевий імпульс, полярність якого буде протилежною полярності вхідного імпульсу.

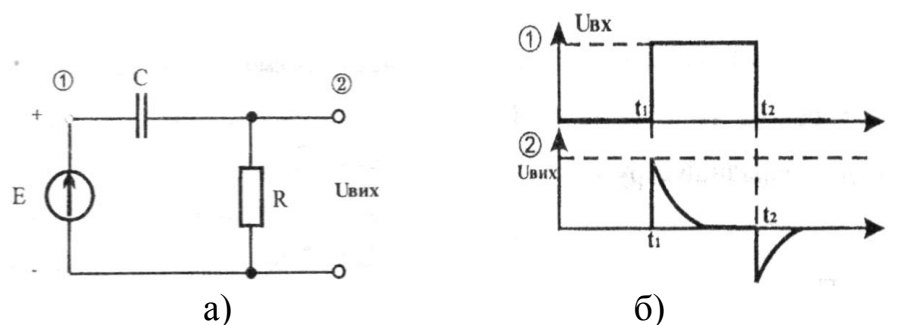


Рисунок 4.10 – Диференційне коло

Диференційні кола широко використовуються для одержання короткочасних імпульсів, які за часом відповідають перепаду з "0" в "1" або з "1" в "0".

Перепаду з "0" в "1" відповідає вихідний імпульс позитивної полярності, а перепаду з "1" в "0" - вихідний імпульс негативної полярності.

Більшість сімейств інтегральних мікросхем є схемами "додатної логіки" і від імпульсів від'ємної полярності не запускаються. Тому, для

одержання в схемі виділення перепаду з "1" в "0" імпульсу додатної полярності застосовують схему, зображену на рис. 4.11.

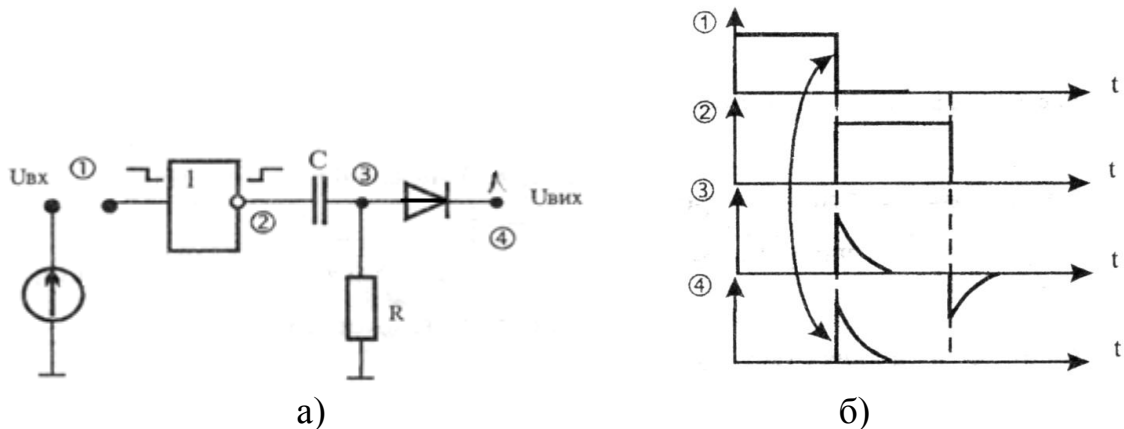


Рисунок 4.11 – Схема виділення перепаду з "1" в "0" та часові діаграми

4.2.2 Схеми виділення перепадів на основі диференційних кіл і логічних елементів

На рис. 4.12 та 4.13 наведені схеми виділення перепадів на основі диференційних кіл і логічних елементів.

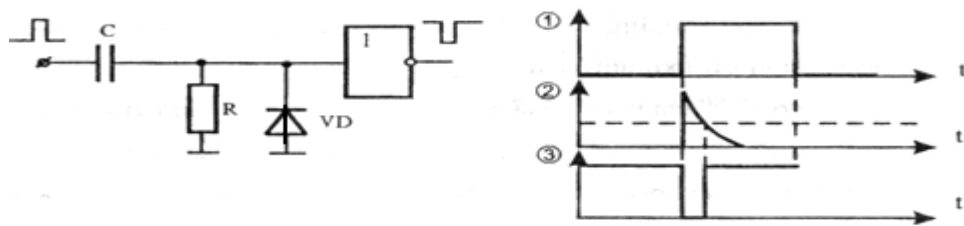


Рисунок 4.12- Схема виділення перепаду з "0" в "1" та часові діаграми

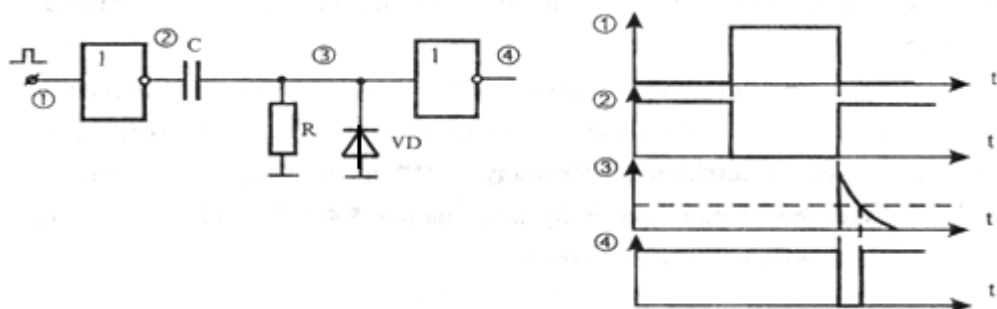


Рисунок 4.13 – Схема виділення перепаду з "1" в "0" та часові діаграми

4.2.3 Схеми виділення перепаду з "0" в "1" та з "1" в "0" на основі інтегруючих кіл і логічних елементів

Якщо не пред'являються високі вимоги до стабільності і тривалості вихідних імпульсів в схемах виділення перепадів з "0" в "1" застосовують схеми на основі інтегруючих RC -кіл. В такому випадку для розрахунку тривалості вихідних імпульсів використовують ті співвідношення, що і для однобраторів.

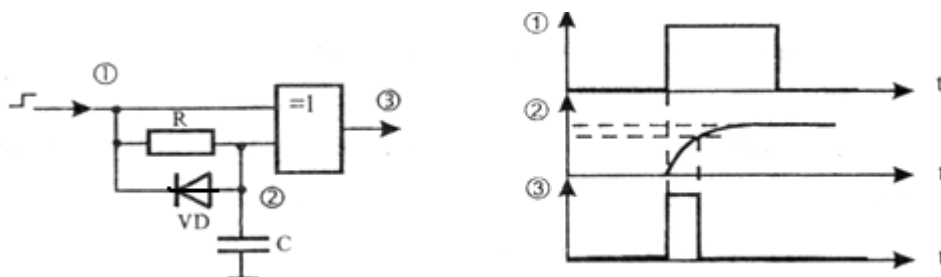


Рисунок 4.14 – Схема виділення перепаду з "0" в "1" та часові діаграми її роботи

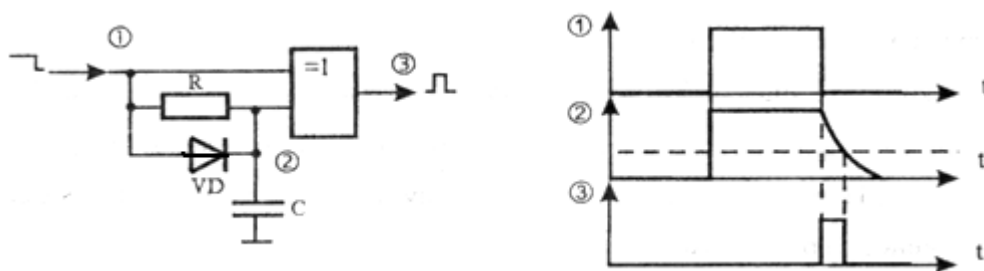


Рисунок 4.15 – Схема виділення перепаду з "1" в "0" та часові діаграми її роботи

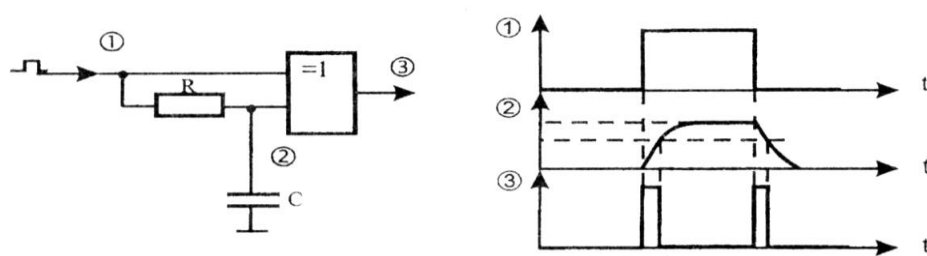


Рисунок 4.16 – Схема виділення перепаду з "0" в "1", з "1" в "0" та часові діаграми її роботи

4.2.4 Схеми виділення перепадів, які використовують властивість затримки вхідних сигналів логічними елементами

Ідея побудови таких схем базується на застосуванні спеціальних схем затримки вхідних сигналів. Найпростішими елементами затримки вхідних сигналів можуть служити логічні елементи ТТЛ або КМОН структур. Як відомо, логічний елемент переключається не одразу після подачі вхідного

сигналу, а через деякий проміжок часу, який обумовлений перехідними процесами всередині самого елемента. Так середня затримка розповсюдження сигналу для ТТЛ мікросхем складає 18-20 нс, а КМОН мікросхем близько 200 нс.

Принцип дії таких схем перепаду пояснимо на прикладі схеми, зображеної на рис. 4.17. Після подачі вхідного імпульсу в момент t_1 на вхід елемента DD1 на його виході виникне перепад напруги з "1" в "0" (точка 2) з затримкою $\tau_{затр}$ одного елемента DD1. З моменту часу t_1 на обох входах логічного елемента DD2 з'явиться логічна "1", але переключення елемента DD2 запізниться на час затримки $\tau_{затр}$, як зображено на часових діаграмах.

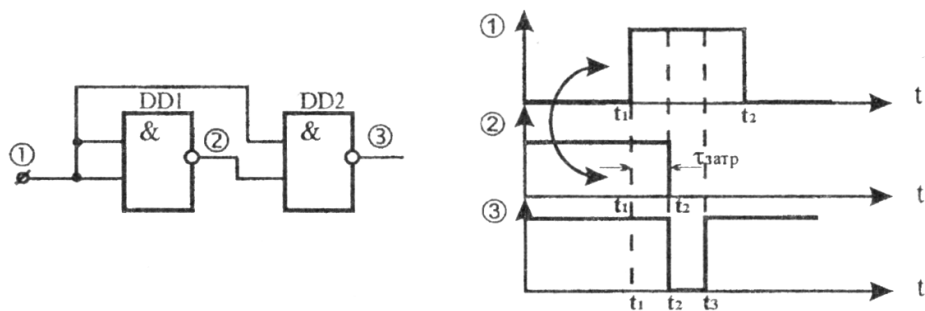


Рисунок 4.17 – Схема виділення перепаду з "0" в "1" та часові діаграми її роботи

Схема виділення перепадів з "1" в "0" і часові діаграми її роботи зображені на рис. 4.18.

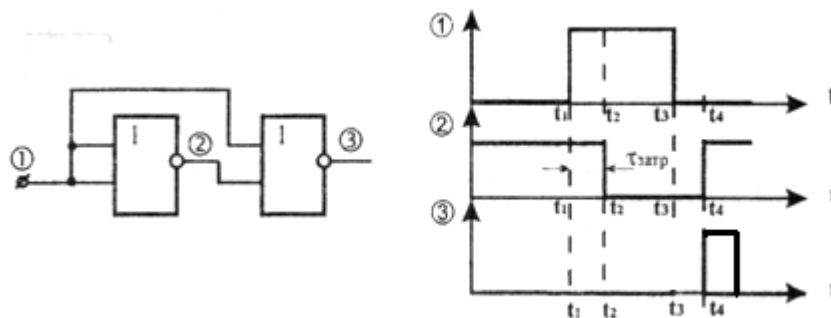


Рисунок 4.18 – Схема виділення перепаду з "1" в "0" та часові діаграми її роботи

При необхідності збільшити тривалість вихідного імпульсу між точками 1 і 2 схеми (рис. 4.18) необхідно включити непарну кількість інверторів.

В зв'язку з тим, що на вході схем можуть існувати два види перепадів і на виході також можуть бути сформовані імпульси 2-х видів, то можна побудувати чотири основні схеми виділення перепадів. Варіанти таких схем на логічних елементах представлені на рис. 4.19 – 4.22.

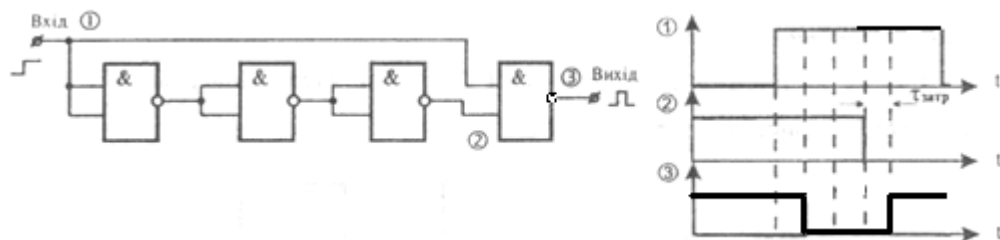


Рисунок 4.19 – Схема виділення перепаду з “0” в “1” з вихідним імпульсом від’ємної полярності та часові діаграми її роботи

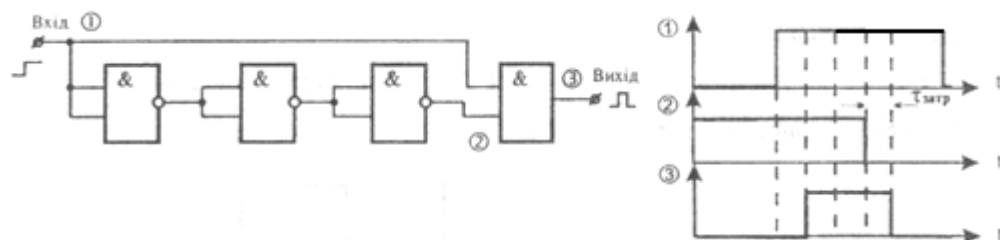


Рисунок 4.20 – Схема виділення перепаду з “0” в “1” з вихідним імпульсом додатної полярності та часові діаграми її роботи

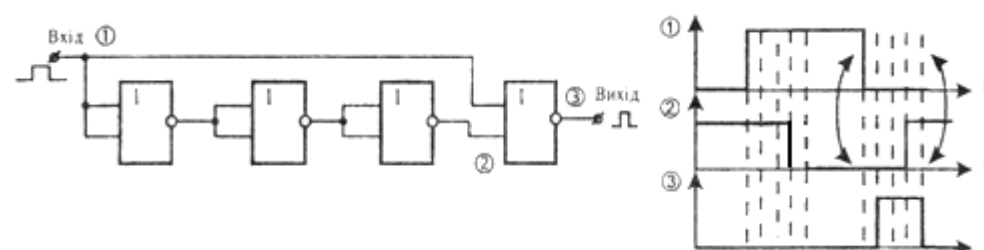


Рисунок 4.21 – Схема виділення перепаду з “1” в “0” з вихідним імпульсом додатної полярності та часові діаграми її роботи

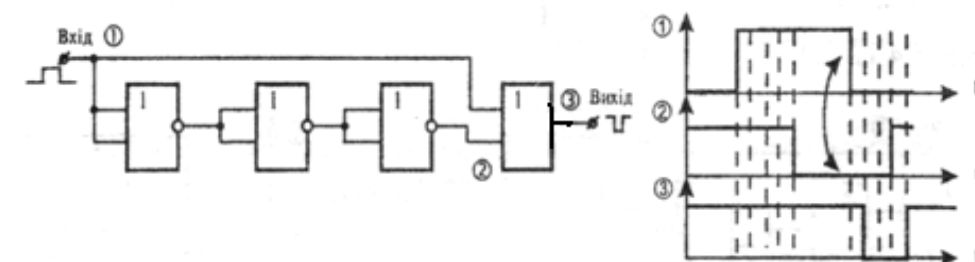


Рисунок 4.22 – Схема виділення перепаду з “1” в “0” з вихідним імпульсом від’ємної полярності та часові діаграми її роботи

4.3 Генератори прямокутних імпульсів

В техніці передачі інформації застосовується багато схем генераторів прямокутних імпульсів, які виконані на різній елементній базі. Найчастіше в сучасній техніці генератори прямокутних імпульсів виконують на інтегральних схемах – цифрових чи аналогових.

Робота більшості таких схем ґрунтується на періодичному зарядженні і розрядженні конденсаторів. Перемикають конденсатори із зарядження на розрядження так звані пускові пристрої. Такі пристрої перебувають у

закритому стані поки напруга їх на вході не перевищує певного рівня. Якщо напруга на їх вході перевищить цей рівень, пристрій переходить у відкритий стан. В якості пускового пристрою, при застосуванні цифрових інтегральних мікросхем, використовують логічні елементи – інвертори.

4.3.1 Мультивібратор

В якості генератора прямокутних імпульсів дуже часто використовують мультивібратори. За своєю структурою, принципом роботи і характером роботи кола, що задає час, схема генератора в основному подібна схемі такого мультивібратора на дискретних елементах.

Розглянемо принцип роботи мультивібратора, побудованого на логічних елементах (рис. 4.23). Припустимо, що інвертор $DD1$ закритий. При цьому на його виході встановлюється рівень логічної «1», конденсатор $C1$ заряджається вихідним струмом елемента $DD1$ по колу: вихід $DD1$ – ємність $C1$, резистор $R2$ - заземлення. На проміжку часу $t_1 - t_2$ доки напруга на резисторі $R2$ (вхід елемента $DD2$) перевищує пороговий рівень переключення елемента, він відкритий, на його виході логічний «0» і конденсатор $C2$ повністю розряджений через елемент $DD2$ і діод $VD1$.

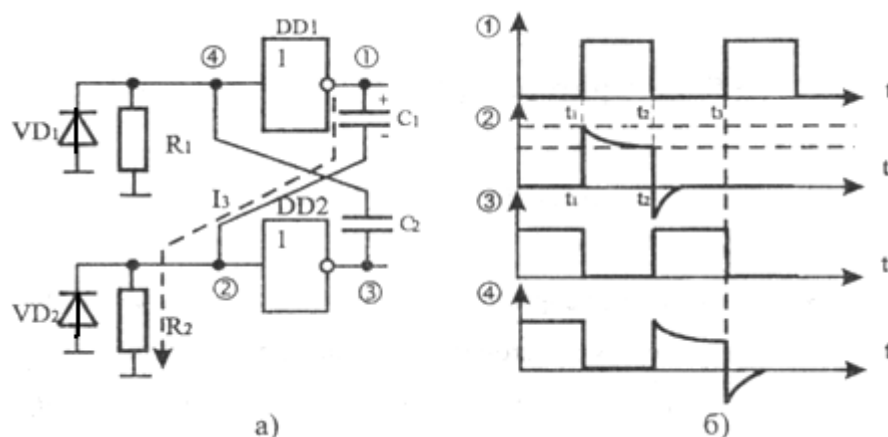


Рисунок 4.23 – Мультивібратор на логічних елементах

З ростом заряду конденсатора $C1$ його зарядний струм зменшується, зменшується додатна по відношенню до землі напруга на резисторі $R2$ (вхід елемента $DD2$) і в той момент часу, коли вона стане рівною пороговому значенню (момент часу t_2), елемент $DD2$ починає закриватись, виникає регенеративний процес зміни струмів і напруг, який завершується повним закриттям $DD2$ і відкриттям $DD1$. Тепер на вході елемента $DD2$ встановлюється рівень логічної "1" і почне заряджатися конденсатор $C2$ по колу: вихід $DD2$, $C2$, резистор $R1$, заземлення. Конденсатор $C1$ розряджається по колу: відкритий елемент $DD1$, заземлення, діод $VD2$, $C1$.

Таким чином схема безперервно генерує прямокутні імпульси. Якщо прийняти $R1=R2=R3$, а $C1=C2=C$, то частота коливань генератора визначається за формулою $f=1/1,4RC$.

В наведеній на рис. 4.23 схемі може виникнути такий стан, коли при підключенні джерела живлення напруга живлення буде наростати відносно уповільнено при малих струмах і падіння напруги ні на одному з резисторів $R1$ і $R2$ не перевищить порогового рівня переключення мікросхем $U_{пор}$. Тоді обидва елемента будуть закритими і автоколивань не буде. В цьому випадку необхідно подати запускаючий імпульс від стороннього джерела на вхід одного з елементів для переведу цього елемента у відкритий стан. Таким чином наведена схема працює в режимі жорсткого самозбудження.

М'яке самозбудження забезпечується доповненням схеми елементом "І" ($DD3$), як показано на рис. 4.24. Якщо обидва елементи в початковий момент будуть закритими, то за рахунок оберненого зворотного зв'язку елемент $DD1$ відкриється і в мультивібраторі виникнуть автоколивання.

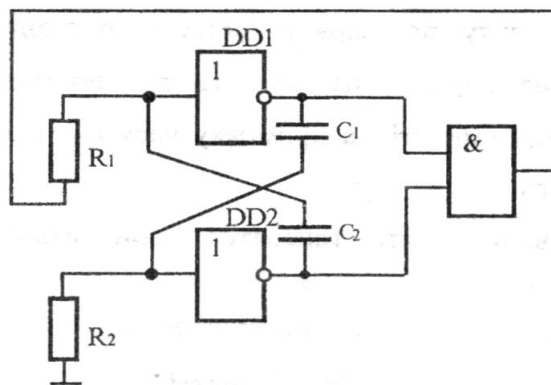


Рисунок 4.24 – Схема мультивібратора на логічних елементах, яка працює в режимі м'якого самозбудження

4.3.2 Генератори прямокутних імпульсів з одним RC колом

На практиці поширені схеми генераторів прямокутних імпульсів, в яких використовуються не два кола $R1C1$ і $R2C2$, що задають час, а тільки одне коло RC . Схема такого генератора наведена на рис. 4.25. Розглянемо принцип його роботи. Припустимо, що інвертор $DD1$ закритий, при цьому на його виході встановлюється рівень логічної "1". Конденсатор C починає заряджатись по колу: - вихід $DD1$ - резистор R - ємність C - відкритий інвертор $DD2$ - заземлення. Напруга на ємності збільшується за експоненціальним законом і прикладена до входу елемента $DD1$, резистор $R2$ під'єднаний додатною обкладинкою конденсатора C відносно землі. В момент t_2 ця напруга досягає порогового рівня переключення елемента $DD1$ і він переходить в нульовий стан. В цей момент часу напруга на ємності стає рівною $U_{дж} + U_{пор}$ бо конденсатор C не може одразу змінити свій стан. На проміжку часу від t_2 і t_3 починається перезаряд по колу: - вихід елемента $DD2$ - ємність C - резистор $R1$ - відкритий елемент $DD1$ - заземлення. Коли напруга на ємності досягає порогового рівня переключення елемента $DD1$, він закривається, на виході з'явиться стан логічної "1", і конденсатор C почне перезаряджатись. Але в момент t_3 до початку перезаряду ємності до входу

елемента $DD1$ прикладена від'ємна напруга на ємності, яка поступово зменшується внаслідок перезаряду ємності на проміжку часу $t_3 - t_4$. Далі процес повторюється. Проміжок (t_1-t_2) - час зарядження ємності $C1$ до $U_{пор}$; (t_2-t_3) - час переключення елементів $DD1$ і $DD2$ і перезаряд ємності $C1$ до $U_{пор}$; (t_3-t_4) - час переключення елементів $DD1$ і $DD2$ і перезаряд ємності $C1$ до $U_{пор}$.

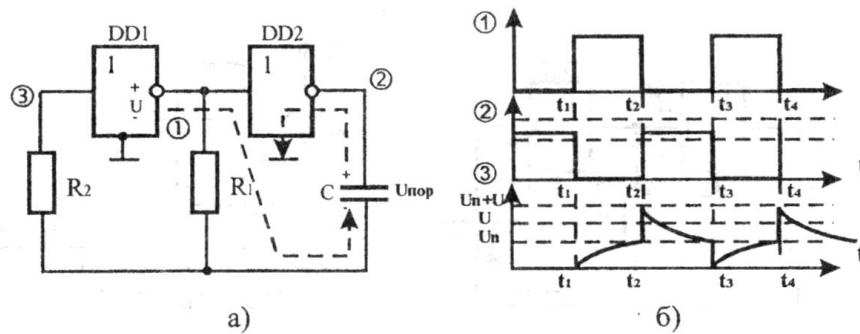


Рисунок 4.25 – Генератор з одним RC колом

4.3.3 Генератори прямокутних імпульсів на основі тригерів

Широке розповсюдження одержали генератори на основі тригерів Шмітта. Тригером Шмітта називається пристрій, який має різні рівні включення і виключення (пороги спрацьовування).

На рис. 4.26 наведене умовне позначення, передаточна характеристика (петля гістерезиса) і часові діаграми формування неінвертуючим тригером Шмітта вихідного сигналу $U_{вих}$ з крутими фронтами з повільно змінюваною вхідною напругою U_i .

Вихідна напруга тригера $U_{вих}$ стрибком змінюється з "0" на "1", якщо вихідна напруга досягне рівня $U_{пвкл}$, і стрибком змінюється з "1" на "0" при зміні вхідної напруги нижче $U_{пвик}$, тобто тригери Шмітта мають 2 сталих стани, але на відміну від звичайних тригерів мають один вхід, можуть управлятись як цифровими, так і аналоговими сигналами.

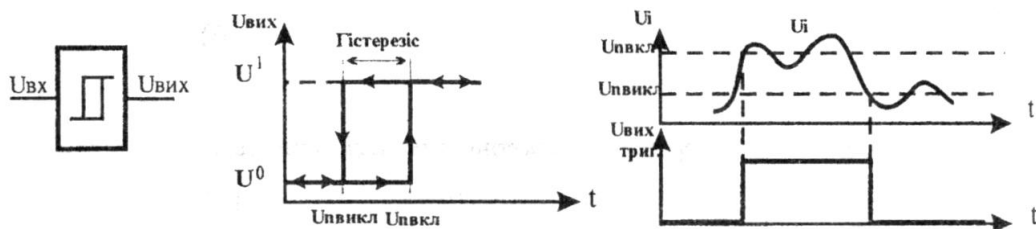


Рисунок 4.26 – Тригер Шмітта

Різниця рівнів $U = U_{пвкл} - U_{пвик}$ називається шириною петлі гістерезиса. Тригери Шмітта часто використовують для формування прямокутних імпульсів, а також для фільтрації.

Розглянемо роботу автогенератора прямокутних імпульсів на базі інвертуючого тригера Шмітта (рис. 4.27).

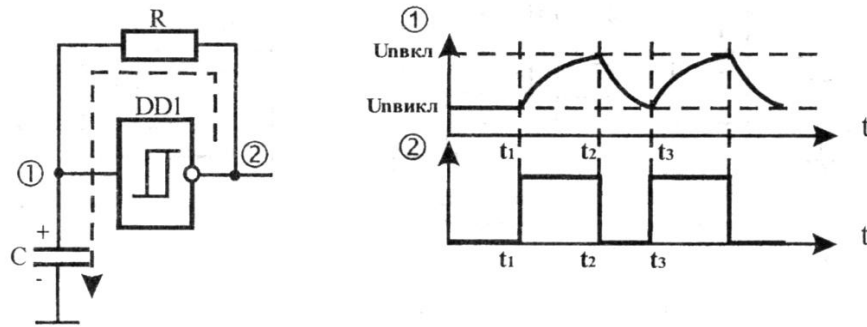


Рисунок 4.27– Автогенератор на базі тригера Шмітта

Припустимо, що в момент часу t_1 на виході тригера Шмітта рівень логічної "1". Конденсатор C заряджається по колу: вихід тригера, опір R , ємність C , заземлення. Напруга на виході конденсатора збільшується і коли вона досягне $U_{пор\ вкл}$, тригер включається (переходить в одиничний стан і зразу інвертується в нульовий). Конденсатор почне розряджатись по колу: верхня обкладинка конденсатора C , опір R відкритий елемент $DD1$, заземлення. Напруга на конденсаторі C зменшується до величини $U_{п\ вик}$, елемент $DD1$ знову переходить в стан логічної "1" і процес повторюється.

4.3.4 Генератори прямокутних імпульсів на основі інтегральних ключів

Схема генератора прямокутних імпульсів з підвищеною навантажувальною здатністю наведена на рис. 4.28.

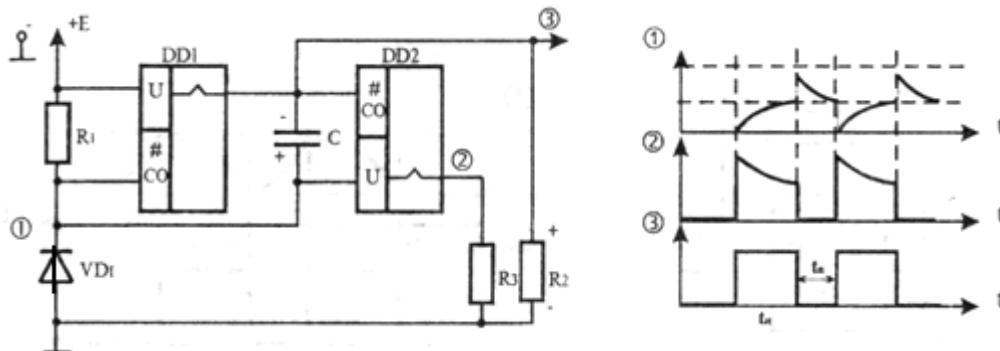


Рисунок 4.28 – Генератора з підвищеною навантажувальною здатністю

Схема працює наступним чином. Після включення напруги обидва ключі розімкнені. Конденсатор C розряджений, починає протікати зарядний ток від джерела по колу: $+E$, резистор $R1$, конденсатор C , резистор $R2$, заземлення. Резистори вибрані так ($R1 > R2$), щоб напруга на резисторі $R2$, яка управляє роботою ключа $DD2$, була меншою порогової і в подальшому зі зростанням заряду ємності C вона зменшується і стає рівною 0.

В той же час між точкою 1 і заземленням напруга, яка управляє ключом $DD1$, експоненціально збільшується і в деякий момент часу стає рівною

порогу включення ключа $DD1$. Після його включення через резистор $R2$ пройде струм по колу: $+E$, резистор $R2$. Напруга на резисторі збільшується і стає рівною порогу включення ключа $DD2$. Він включається зразу ж після замкнення обох ключів, верхня обкладка конденсатора буде під'єднана до $+$ джерела живлення. Заряд на ємності, який був накопичений, не може змінитись миттєво, тому напруга в точці 1 стрибком збільшується на величину $E_{дж} + U_{порDD1}$. Після цього напруга на ємності C починає зменшуватись з сталою часу, рівною $C/R1R3/R1+R3$, і намагається досягти рівня напруги, який задається подільником напруги $R1, R2$. В процесі перезарядження ємності напруга на C зменшиться до порога виключення ключа $DD1$. В результаті обидва ключі розімкнуться і знову почнеться процес заряду конденсатора по колу: $+E, R1, C, R2$ тощо.

Для захисту ключа $DD2$ від від'ємного кидка напруги в схему вводиться діод $VD2$. Тривалість паузи між імпульсами регулюється за допомогою резистора $R1$.

4.4 Схеми затримки імпульсів

У системах автоматики часто виникає потреба затримати вихідний імпульс на деякий час τ відносно вхідного. Пристрої, які затримують вихідний імпульс відносно вхідного на деякий час, називають елементами затримки.

Найпростішими елементами затримки вхідних сигналів можуть служити логічні елементи ТТЛ або КМОН структур. Як відомо, логічний елемент переключається під дією вхідних прямокутних імпульсів не одразу, а через деякий проміжок часу, який обумовлений перехідними процесами всередині самого елемента. Так середня затримка розповсюдження сигналу для ТТЛ мікросхем складає 18-20 нс, а для КМОН мікросхем близько 200 нс. Умовне позначення елемента затримки зображено на рисунку 4.29.

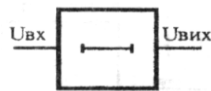


Рисунок 4.29 – Умовне позначення елемента затримки

Схема елемента затримки, яка не потребує застосування зовнішніх "навісних" елементів (резисторів, конденсаторів), та часові діаграми її роботи представлені на рисунку 4.30.

Час затримки в таких схемах залежить від кількості послідовно з'єднаних елементів і визначається за формулою:

$$\tau_{\Sigma} = n \cdot \tau_1, \quad (4.2)$$

де n - кількість елементів (інверторів), які використовуються в схемі;

τ_1 - час затримки одного елемента.

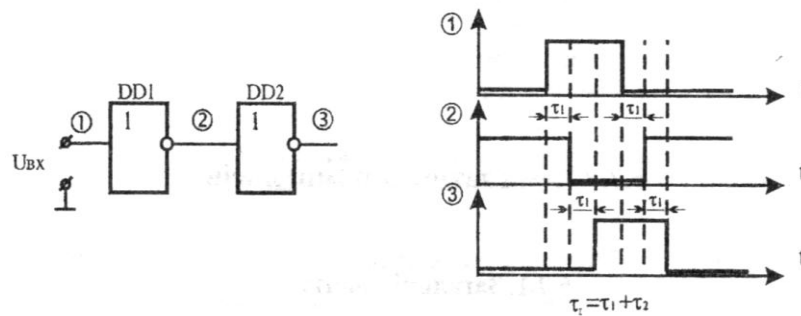


Рисунок 4.30 – Схема елемента затримки

Якщо необхідно збільшити час затримки на значний час, застосовують схему затримки, в якій використовують логічні елементи і RC -кола. Розповсюджена схема затримки, зображена на рис. 4.31.

В початковий момент часу конденсатор C розряджений, на вході і виході $DD1$ рівень логічного нуля.

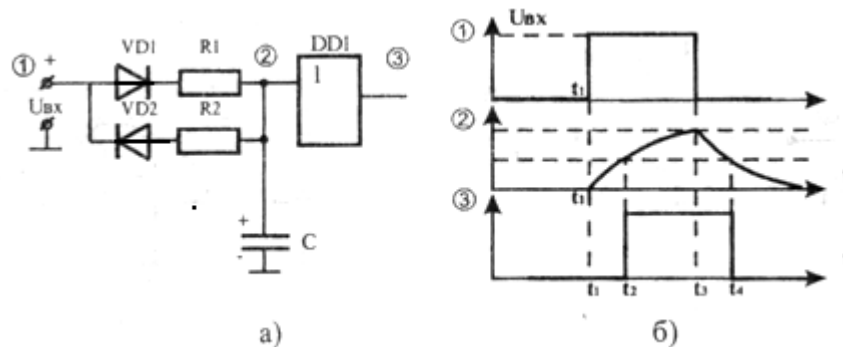


Рисунок 4.31 – Схема затримки сигналу на основі логічних елементів і RC -кіл

а) функціональна схема; б) часові діаграми роботи

При появі на вході схеми (момент t_1) вхідного імпульсу конденсатор C починає заряджатись по колу: $+U_{vx}$, діод $VD1$, опір R , конденсатор C , заземлення. Як тільки напруга на ємності C досягне порогового значення переключення елемента, він переходить в стан логічної "1". У момент t_3 після закінчення дії вхідного імпульсу конденсатор починає розряджатись по колу: верхня обкладка конденсатора C , опір $R2$, діод $VD2$, внутрішній опір джерела вхідного сигналу, заземлення, нижня обкладка конденсатора.

При цьому напруга на ємності зменшується і коли вона досягне порогового рівня напруги, елемент $DD1$ знову переходить в попередній стан. Як видно з часових діаграм, наведена схема дає можливість встановлювати затримку переключення логічного елемента: резистором $R1$ встановлюється затримка вхідного перепаду з "0" в "1", а резистором $R2$ - затримка вихідного перепаду з "1" в "0" при сталій ємності.

Максимальний час затримки фронту і спаду імпульсу не може перевищувати 80% від тривалості вхідного імпульсу.

Досить часто застосовують схему затримки фронту вхідного сигналу, яка зображена на рис. 4.32.

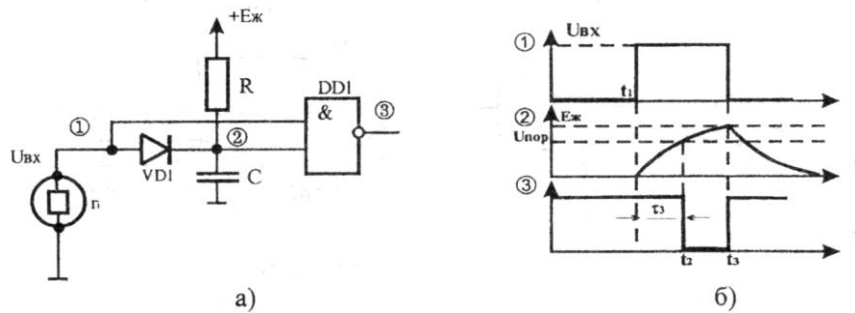


Рисунок 4.32 – Схема елемента затримки з інвертованим вихідним сигналом
а) функціональна схема; б) часові діаграми роботи

Схема затримки працює таким чином. В початковий момент конденсатор C розряджений, на вході елемента І-НЕ логічний "0", на виході логічна "1". При появі на вході елемента фронту вхідного імпульсу діод $VD1$ закривається і конденсатор C починає заряджатись по колу: $+E_{ж}$, опір R , ємність C , заземлення.

При цьому напруга на конденсаторі збільшується і коли вона досягне порогового рівня напруги переключення, елемент $DD1$ відкривається, на його виході з'явиться стан логічного "0". Цей стан буде зберігатись до спаду вхідного імпульсу (момент t_3 на часовій діаграмі). Після закінчення дії вхідного імпульсу вихідний стан логічного елемента перейде в попередній стан логічної "1". В цій схемі тривалість вихідного імпульсу не рівна тривалості вхідного, але час затримки вихідного імпульсу інвертованого відносно вхідного імпульсу: $\tau_{затр.} = 1,5 \tau$

Існують і більш складні цифрові схеми затримки, які застосовують генератори прямокутних імпульсів, зсувні регістри і спеціальні схеми управління.

5. ІНТЕГРАЛЬНИЙ ТАЙМЕР

Однією з найбільш широко застосовуваних простих аналого-цифрових інтегральних схем є інтегральний таймер типу КР1006ВИ1 (закордонні аналоги SE555, NE555) [1, 23].

Головне призначення - генерація імпульсів як малої, так і великої тривалості (від мікросекунд до 1 години). До появи цієї ІС інтервали схеми генерувалися за допомогою різноманітних релаксаційних схем та інтеграторів. Спеціалізована ІС дозволила стандартизувати і спростити ці дуже поширені вузли.

5.1 Робота таймера в режимі одновібратора

На рис. 5.1 зображений одновібратор на основі таймера КР1006ВИ1. До схеми залучені два зовнішні елементи R_x , C_x , що задають час. У схемі чотири головних вузли: два компаратора - верхній (ВК) і нижній (НК), тригер пам'яті (ТГ) і розрядний транзистор $VT14$.

На опорні входи компараторів подаються потенціали від подільника напруги (три однакових опори R_{3-5}). Нормальний стан $VT14$ - насичений (при цьому конденсатор C_x замкнений на землю). Це означає, що вихідний - початковий стан тригера нульовий, тобто на виході Q - рівень логічного нуля, а на виході $\bar{Q}$ - рівень логічної одиниці, який відчиняє транзистори $VT16$, $VT14$ (на виході 3 - нуль) і (на виході 7 - теж нуль).

Якщо на вхід 2 нижнього компаратора подати від'ємний імпульс запуску і потенціал цього входу стане менше напруги порога $+E_k/3$, компаратор, а за ним і тригер, переключиться, на виході Q виникає додатний, а на виході $\bar{Q}$ - від'ємний перепад (з 1 до 0). Цим стрибком транзистори $VT14$ і $VT16$ закриваються, обкладки C_x розмикаються, конденсатор починає заряджатися через резистор R_x від джерела живлення $+E_k$. Потенціал на конденсаторі підвищується. Коли він перевищить напругу $+E_k \cdot 2/3$ (поріг спрацьовування ВК), верхній компаратор переключиться і приведе тригер до вихідного стану. У цей момент $VT14$ відчиняється позитивним перепадом і C_x із великою швидкістю розряджається на землю через його колектор-емітер. На цьому цикл роботи таймера закінчується. Він триває після імпульсу запуску на протязі часу $T = 1,1R_xC_x$.

Для скидання таймера, тобто для встановлення на його виході низької напруги, незалежно від напруги на виходах 2 і 6, використовується вихід 4 (скидання). Якщо напруга на цьому виводі менше 0,4В (логічний «0»), напруга на виводі 3 дорівнює 0,1В. При напрузі більше 1В на виводі 4 коло скидання включене і не впливає на роботу таймера.

Допоміжний вивід 7 представляє собою відкритий колектор $VT14$. Цей вивід звичайно використовується для організації зворотного зв'язку з виходу на входи таймера.

Припустима зміна напруги на виводах 2, 4, 6 (стоп) і 7 (ланцюг розряду) лежить у межах 0-16,5В. Вивід 5 (контроль подільника) дозволяє додатково управляти роботою таймера, змінюючи граничні напруги компараторів при постійній напрузі живлення. Рекомендується шунтувати цей вивід конденсатором ємністю $\approx 0,01$ мкФ, щоб уникнути впливу зовнішніх перешкод і пульсацій напруги живлення на точність роботи таймера.

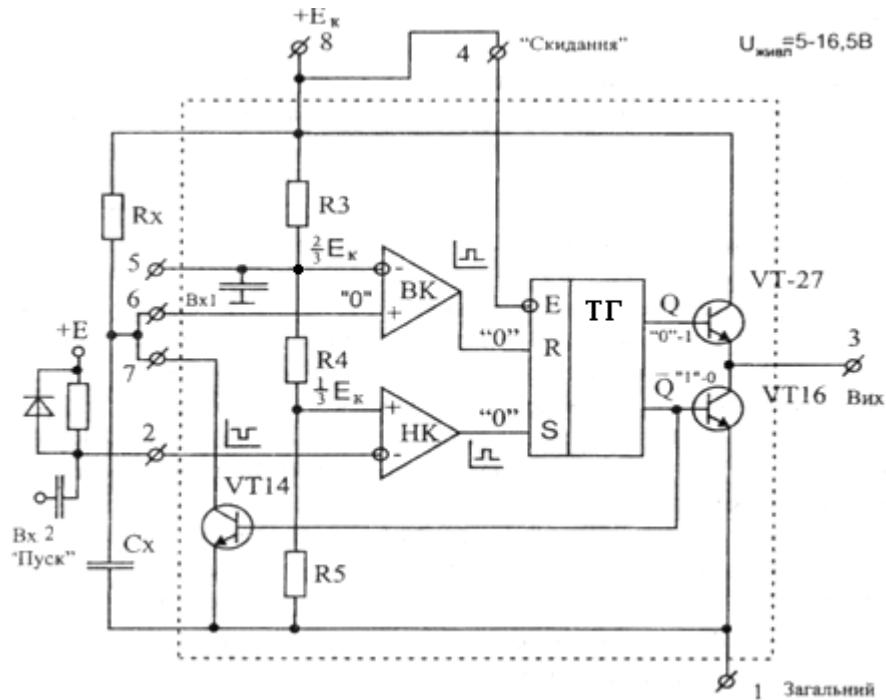


Рисунок 5.1 – Одновібратор на основі таймера КР1006ВИ1

Внутрішні компаратори таймера не рівносильні за своїм впливом на RC-тригер. Компаратор НК (нижній) має пріоритет при встановленні стану тригера. Компаратор ВК (верхній) може управляти тригером тільки тоді, коли на вхід 2 подана висока напруга.

5.2 Робота таймера в режимі генератора прямокутних імпульсів

Генератор прямокутних імпульсів на основі таймера КР1006ВИ1 представлений на рис. 5.2.

Припустимо, що в момент часу $t = 0$ починається стадія заряду конденсатора C , через резистори R_1 та R_t . Вихідний стан при цьому $Q = 1$, $\bar{Q} = 0$. На виході ВК буде при цьому 0, а на виході НК буде 1, що підтримує стан заряду конденсатора. Як тільки напруга на конденсаторі стане вище опорної на НК, але менше опорної ВК, напруга на виході НК перейде в 0. При цьому на обох виходах компараторів будуть нулі. Тригер залишиться в попередньому стані.

Як тільки конденсатор зарядиться до напруги, більшої U_{on} ВК, він перейде до стану 1 і переключить тригер в нульовий стан (рис. 5.3). Відкриються два транзистори VT14 та VT16, і конденсатор C , почне

розряджатися по колу: верхня обкладка – R_1 , емітер-колектор $VT14$ - заземлення.

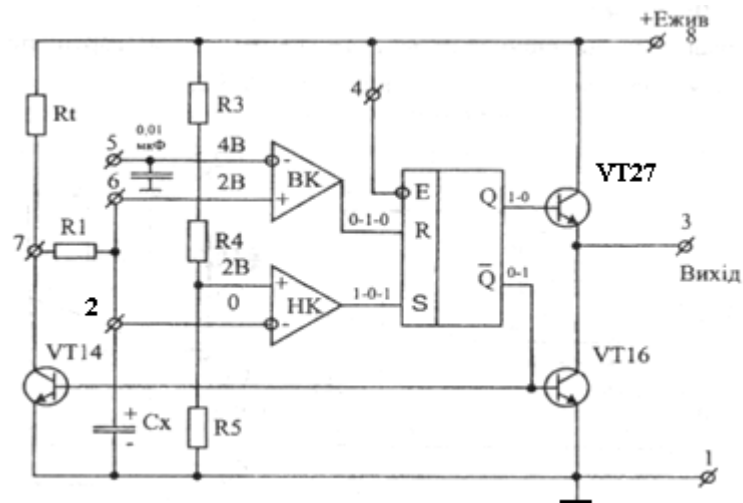


Рисунок 5.2 – Генератор на основі таймера КР1006ВИ1

Як тільки напруга на ємності стане менше U_{on} НК, тригер переключиться в одиничний стан і знову почнеться заряд конденсатора.

Період коливань $T = t_{зар} + t_{розр} \approx 0,7(R_1 + 2R_1)C$.

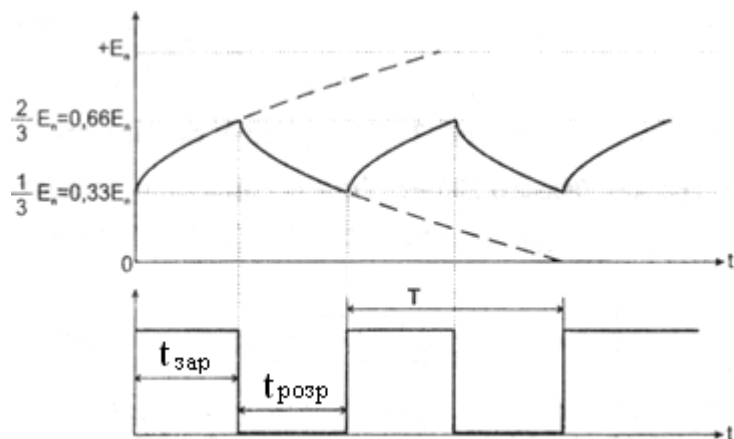


Рисунок 5.3 – Часові діаграми роботи автогенератора

6 ЗВОРОТНІЙ ЗВ'ЯЗОК ТА ОПЕРАЦІЙНІ ПІДСИЛЮВАЧІ

Поняття «зворотній зв'язок» (ЗЗ) відноситься до числа поширених, воно давно вийшло за рамки вузької області техніки і вживається зараз в широкому значенні. В системах управління зворотній зв'язок використовується для порівняння вихідного сигналу з заданим значенням і виконання відповідної корекції. В якості «системи» може виступати що завгодно, наприклад процес управління автомобілем, що рухається по дорозі - за вихідними даними (положенням машини та її швидкістю) слідує водій, який порівнює їх з очікуваними значеннями і відповідно корегує вхідні дані (за допомогою керма, перемикача швидкості, гальм).

В системі підсилювача вихідний сигнал повинен бути кратним вхідному, тому в підсилювачі зі зворотнім зв'язком вхідний сигнал порівнюється з певною частиною вихідного сигналу [1, 7, 25].

Розглянемо декілька прикладів використання зворотного зв'язку в операційних підсилювачах.

Операційний підсилювач (ОП) - це диференційний підсилювач постійного струму з високим коефіцієнтом підсилення і несиметричним виходом. Прообразом ОП може слугувати класичний диференційний підсилювач з двома входами і несиметричним виходом; правда, слід зазначити, що реальні операційні підсилювачі володіють значно більш високими коефіцієнтами підсилення (зазвичай порядку 10^5 - 10^6) і меншими вихідними імпедансами, а також допускають зміну вихідного сигналу майже в повному діапазоні напруги живлення (зазвичай використовують розщепленні джерела живлення ± 15 В). Умовне позначення ОП наведено на рис. 6.1.

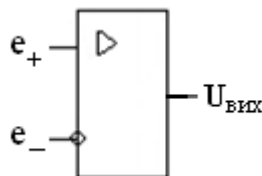


Рисунок 6.1 – Умовне позначення ОП

Один з входів ОП називається інвертуючим (-), який позначено кружком, а другий – неінвертуючим (+). Входи працюють наступним чином: вихідний сигнал змінюється в додатному напрямленні, коли потенціал на вході (+) стає більш додатним чим потенціал на вході (-) і навпаки.

Різницю напруг на входах ОП ($e+ - e-$) називають **диференційним** (різницеvim) вхідним сигналом ОП, а напівсуму цих сигналів $(e+ + e-)/2$ – **синфазним** вхідним сигналом. Диференційна вхідна напруга, при якій вихідна напруга ОП дорівнює нулю, називається **напругою зміщення**.

Поведінку операційного підсилювача, охопленого петлею зворотного зв'язку, визначають основні правила:

I. Вихід операційного підсилювача прагне до того, щоб різниця напруги між його входами дорівнювала нулю.

II. Входи операційного підсилювача струм не споживають.

Ці правила створюють достатню основу для розгляду схем на операційних підсилювачах.

6.1 Інвертуючий підсилювач

Проведемо аналіз схеми, представленої на рис. 6.2.

1. Потенціал точки В дорівнює потенціалу землі, отже, відповідно до правила I, потенціал точки А також дорівнює потенціалу землі.

2. Це означає, що: а) падіння напруги на резисторі R_2 дорівнює $U_{вих}$, б) падіння напруги на резисторі R_1 дорівнює $U_{вх}$.

3. Скориставшись правилом II, отримаємо $U_{вих}/R_2 = - U_{вх}/R_1$, або коефіцієнт підсилення по напрузі $K = U_{вих}/U_{вх} = - R_2/R_1$.

Найчастіше точку В краще заземлити через резистор.

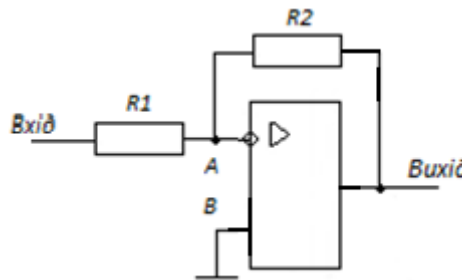


Рисунок 6.2 – Інвертуючий підсилювач

Отже, аналіз схеми на ОП виявився навіть занадто простим. Він, правда, не дозволяє судити про те, що насправді відбувається в схемі. Для того, щоб зрозуміти як працює зворотний зв'язок, представимо собі, що на вхід подано деякий рівень напруги = 1В.

Для конкретизації допустимо, що резистор R_1 має опір 10 кОм, а резистор R_2 - 100 кОм. Тепер уявімо собі, що напруга на виході вирішила вийти з-під контролю і стала рівною 0 В. Що станеться? Резистори R_1 і R_2 утворюють подільник напруги, за допомогою якого потенціал інвертуючого входу підтримується рівним 0,91В. Операційний підсилювач фіксує неузгодженість по входах, і напруга на його виході починає зменшуватися. Зміна триває до тих пір, поки вихідна напруга не досягне значення -10 В, в цей момент потенціали входів ОП стануть однаковими і рівними потенціалу землі.

Аналогічно якщо напруга на виході почне зменшуватися і далі та стане більш негативною, ніж -10 В, то потенціал на вході інвертується, стане нижче потенціалу землі і в результаті вихідна напруга почне рости.

Недолік цієї схеми полягає в тому, що вона володіє малим вхідним імпедансом, особливо для підсилювачів з великим коефіцієнтом підсилення по напрузі (при замкнутому колу ЗЗ), в яких резистор R_1 (як правило, буває

невеликим). Цей недолік усувається за допомогою використання схеми, представленої на рис. 6.3.

6.2 Неінвертуючий підсилювач

Розглянемо схему на рис. 6.3,а. Аналіз її також вкрай простий: $U_A = U_{\text{вх}}$. Напруга U_A знімається з дільника напруги: $U_A = U_{\text{вх}} * R_1 / (R_1 + R_2)$. Якщо $U_A = U_{\text{вх}}$, то коефіцієнт посилення $U_{\text{вих}}/U_{\text{вх}} = 1 + R_2/R_1$.

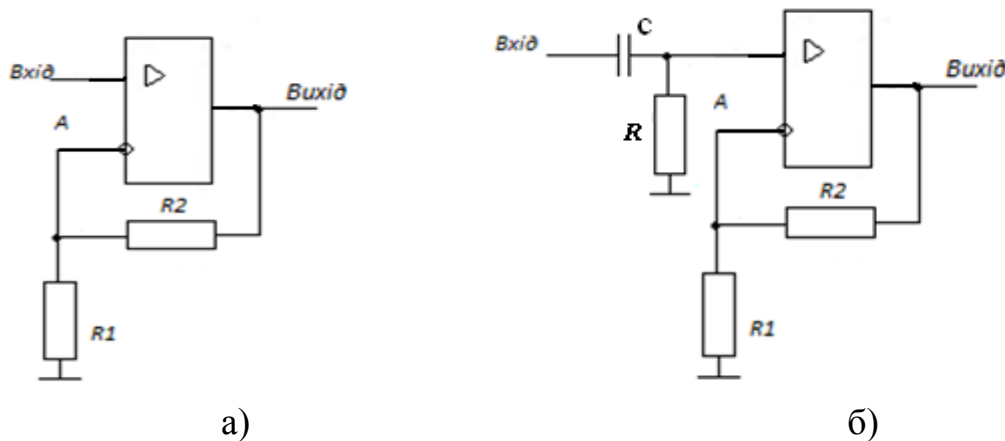


Рисунок 6.3 – Неінвертуючий підсилювач

Вхідний імпеданс цього підсилювача нескінченний - становить принаймні кілька сотень мегом, вихідний імпеданс, як і в попередньому випадку, дорівнює часткам ома. Як і у випадку з інвертуючим підсилювачем, поведінка схеми при зміні напруги на входах аналогічна.

Ця схема також є підсилювачем постійного струму. Якщо джерело сигналу і підсилювач пов'язані між собою по змінному струмові, то для вхідного струму (дуже невеликого за величиною) потрібно передбачити заземлення, як показано на рис. 6.3,б.

6.3 Повторювач

На рис. 6.4 представлений повторювач на основі операційного підсилювача.

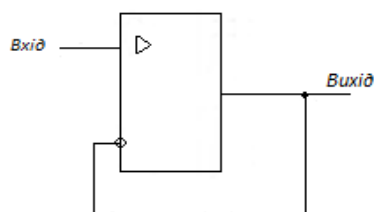


Рисунок 6.4 – Повторювач

Він являє собою не що інше, як неінвертуючий підсилювач, в якому опір резистора R_1 дорівнює нескінченності, а опір резистора R_2 - нулю (коефіцієнт підсилення дорівнює 1).

Підсилювач з одиничним коефіцієнтом посилення називають іноді буфером, так як він володіє ізолюючими властивостями (великим входним опором і малим вихідним).

6.4 Перетворювач струму в напругу

Найпростіший перетворювач струму в напругу - це всього-навсього резистор. Однак у нього є недолік, який полягає в тому, що для джерела входного сигналу входний опір такого перетворювача не дорівнює нулю; цей недолік може виявитися дуже серйозним, якщо пристрій, що забезпечує входний струм, має дуже малий вихідний робочий діапазон або не може забезпечити сталість струму при зміні вихідної напруги. Прикладом може служити діодний фотоеlement (фотодіод), або сонячна батарея (рис. 6.5,а).

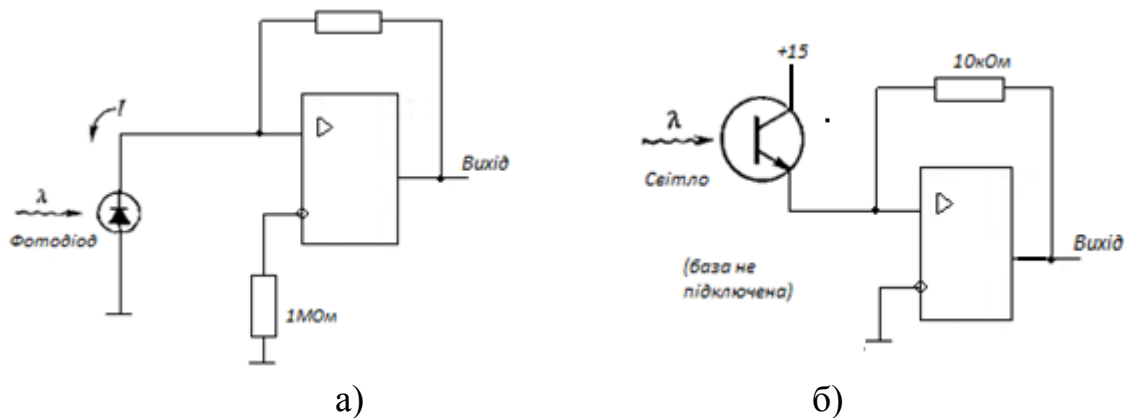


Рисунок 6.5 – Перетворювачі струму в напругу

У таку схему часто включають фотопомножувачі і фототранзистори (обидва елементи під дією світла починають споживати струм від позитивного джерела живлення) (рис. 6.5,б).

6.5 Диференційний підсилювач

На рис. 6.6 представлена схема диференційного підсилювача, коефіцієнт підсилення якого дорівнює R_1/R_2 .

Часто на практиці необхідно ввести в вихідний сигнал диференційного підсилювача постійну складову. Це можна реалізувати шляхом під'єднання джерела зміщення до інвертуючого входу ОП.

Якщо входний сигнал змінюється в межах $(-U_2 \text{ до } +U_2)$ та $E_{зм.} = -U_1$, то вихідний сигнал буде змінюватись від 0 до $\kappa \cdot U_2$, де коефіцієнт підсилення κ визначається відношенням резисторів R_2 до R_1 .

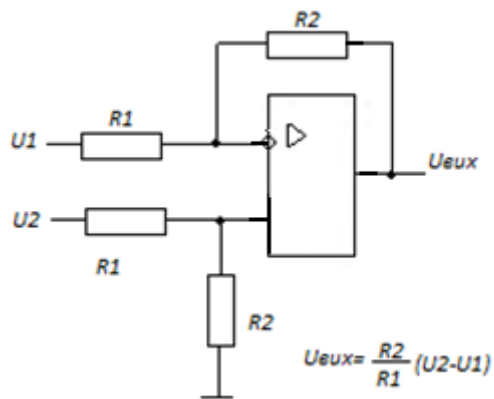


Рисунок 6.6 – Диференційний підсилювач

6.6 Підсумовуючий підсилювач

Схема, що показана на рис. 6.7, є одним з варіантів інвертуючого підсилювача. Точка «х» має потенційний нуль, тому вхідний струм дорівнює $U_1/R + U_2/R + U_3/R$, звідки $U_{вих} = - (U_1 + U_2 + U_3)$.

Зверніть увагу, що вхідні сигнали можуть бути як позитивними, так і негативними. Крім того, вхідні резистори не обов'язково повинні бути однаковими; якщо вони неоднакові, то отримаємо зважену суму. Наприклад, схема може мати чотири входи, на кожному з яких напруга дорівнює +1 В або 0 В; входи представляють двійкові значення: 1, 2, 4 і 8. Якщо використовувати резистори з опором 10, 5, 2,5 і 1,25 кОм, то вихідна напруга (у вольтах) буде пропорційна двійковому числу, яке задано на вході. Цю схему неважко розширити до декількох цифр. Описаний метод представлення чисел лежить в основі цифро-аналогового перетворення, правда, на вході перетворювача зазвичай використовують іншу схему (резистивну сітку R-2R).

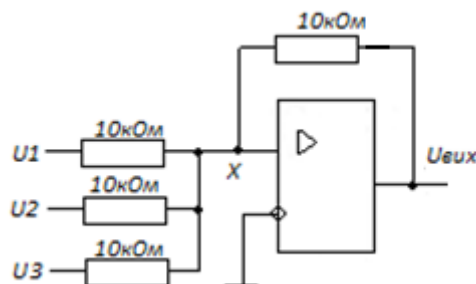


Рисунок 6.7 – Підсумовуючий підсилювач

6.7 Інтегратори

На основі операційних підсилювачів можна будувати майже ідеальні інтегратори, на які не поширюється обмеження $U_{вих} \ll U_{ох}$. На рис. 6.8 показана така схема. Вхідний струм $U_{вх}/R$ протікає через конденсатор C . У зв'язку з тим що інвертуючий вхід має потенційне заземлення, вихідна напруга визначається наступним чином:

$$U_{\text{вх}}/R = -C(dU_{\text{вих}}/dt) \text{ або } U_{\text{вих}} = -1/RC \int U_{\text{вх}} dt + \text{const.}$$

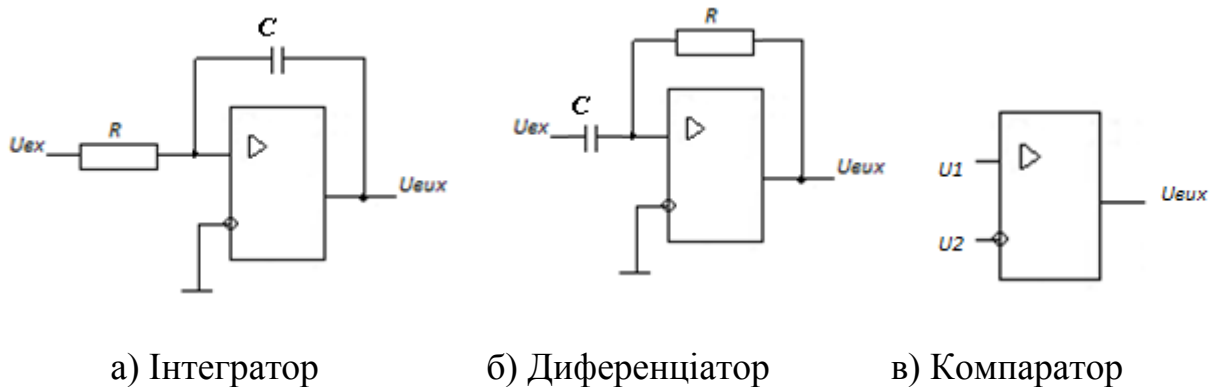


Рисунок. 6.8

Безумовно, вхідним сигналом може бути і струм. В цьому випадку резистор не потрібен.

6.8 Диференціатори

Диференціатори подібні інтеграторам, в них тільки міняються місцями резистор R і конденсатор C (рис. 6.8,б).

Інвертуючий вхід ОП заземлений, тому зміна вхідної напруги з деякою швидкістю викликає появу струму: $I = C(dU_{\text{вх}}/dt)$ і відповідно вихідної напруги: $U_{\text{вих}} = -RC(dU_{\text{вх}}/dt)$.

6.9 Компаратори

Найпростішим компаратором (перетворювачем неперервного сигналу в дискретний) є диференційний підсилювач (рис. 6.8,в). Напруга на виході компаратора може знаходитись на одному з двох фіксованих рівнів: на верхньому, якщо напруга на його неінвертуючому вході більше напруги на інвертуючому вході, і на нижньому – при протилежному співвідношенню цих напруг. Хоча в якості компаратора часто використовують звичайний операційний підсилювач, промисловість випускає спеціальні інтегральні схеми компараторів. Насамперед, компаратори перемикаються значно швидше, ніж ОП, а вихідна напруга компаратора змінюється в межах, що дозволяє безпосередньо здійснювати управління логічними інтегральними схемами.

6.10 Розрахунок ланцюгів, які містять ОП, за допомогою графів

В порівнянні з традиційними методами розрахунку електричних ланцюгів графи дозволяють більш наглядно продемонструвати взаємний вплив різноманітних змінних [7]. Слід відзначити, що існують різноманітні типи графів. Зокрема – узагальнений сигнальний граф, який можна будувати

виходячи безпосередньо з аналізованої схеми, складається з вузлів (вершин) і спрямованих гілок. Його особливістю є наявність зважених вузлів, які позначаються кружком з зазначенням невідомого всередині, та ваги вузлу. Вузли графу, які не мають входних гілок, називаються витоками (незваженими вузлами), позначаються крапками без зазначення ваги.

Часто при розрахунку електричних ланцюгів ОП вважають ідеальним. Граф такого ОП наведено на рис. 6.9.

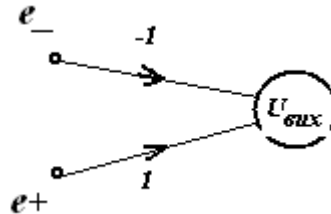


Рисунок 6.9 – Граф ідеального ОП

Приклад побудови графа. Необхідно визначити залежність $U_{вих}$ від $U_{вх}$ для ланцюга, схема якого наведена на рис. 6.10, за умови, що ОП можна вважати ідеальним. Граф даної схеми наведено на рис. 6.10, б. Необхідно відмітити схожість схеми і графа.

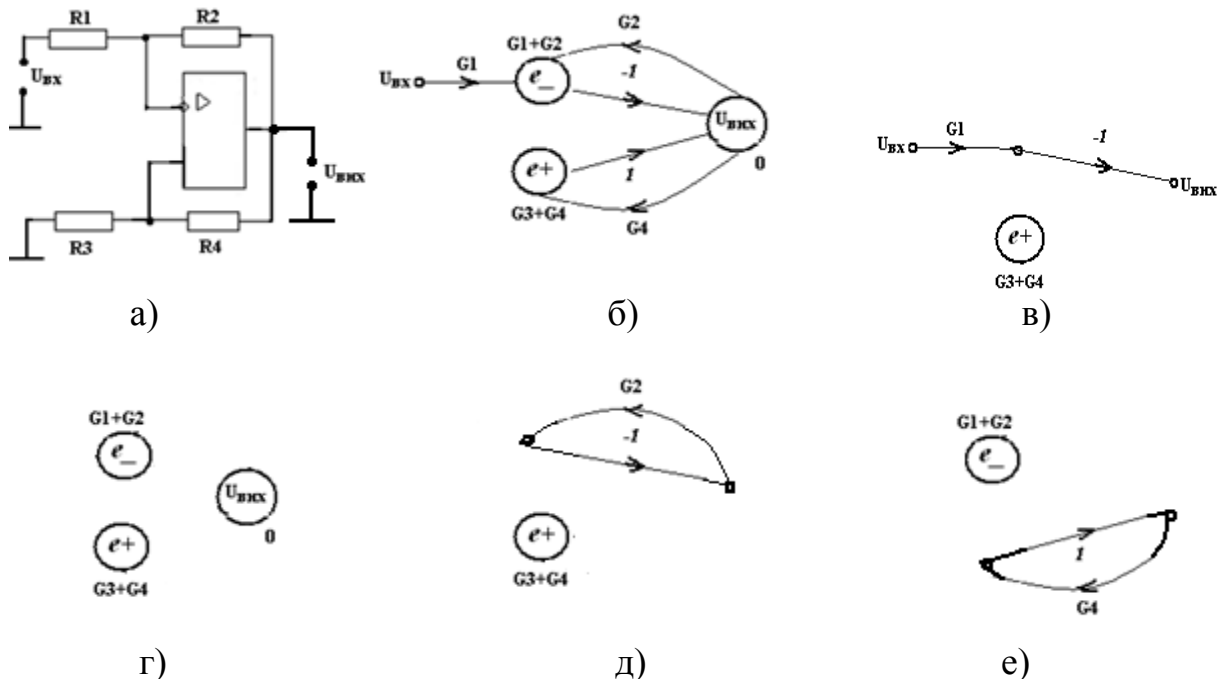


Рисунок 6.10 – Приклад розрахунку активного ланцюгу за допомогою графа

Розв'язання графа. Невідомий параметр $X_g = U_{вих}$, який відповідає одному із зважених вузлів графа, визначається у вигляді лінійної комбінації всіх n відомих параметрів a_k незважених вузлів (витоків):

$$X_g = \sum_{k=1}^n W_{kg} \cdot a_k. \quad (6.1)$$

Коефіцієнт W_{kg} визначається в свою чергу за формулою Мезона:

$$W_{kg} = \sum_{i=1}^m P_i \cdot \frac{\Delta_i}{\Delta}, \quad (6.2)$$

де P_i – коефіцієнт передачі i -го шляху від вузла a_k до вузлу X_g ;
 m – загальна кількість таких шляхів;
 Δ_i – визначник частини графа, який не торкається i -го шляху;
 Δ – визначник повного графа.

Шлях – це послідовність односпрямованих гілок. Шлях не може проходити двічі через один і той же вузол. Коефіцієнт передавання шляху дорівнює добутку коефіцієнтів передавання гілок, які входять до нього. Замкнений шлях називається **контуром**. Контур, в який входить лише одна гілка, носить назву **петлі**. Петлю, яка охоплює деякий вузол узагальненого графу, можна виключити, зменшивши вагу вузла на коефіцієнт її передавання.

Частина графу, яка не торкається i -го шляху, може бути отримана виключенням із графу всіх вузлів, через які проходить даний шлях, а також всіх гілок, які входять в ці вузли і виходять з них.

Визначник графа Δ дорівнює сумі визначників δ_j всіх елементарних графів, які входять в нього: $\Delta = \sum \delta_j$.

Елементарний граф – це сукупність контурів та зважених вузлів, які не торкаються один одного і, через які не проходять ці контури. Контури вважаються недоторканими, якщо вони не мають загальних вузлів.

Визначник елементарного графа δ_j обчислюється як добуток ваг, які входять до його вузлів, і взятих з протилежним знаком коефіцієнтів передачі контурів. Якщо при знаходженні визначника Δ_i , який відповідає i -му шляху P_i , з'ясується, що цей шлях проходить через всі зважені вузли графу, то приймається, що $\Delta_i = 1$.

Приклад рішення розглянемо стосовно до графу (рис. 6.10, б): знайдемо напругу $U_{вих}$. Граф в даному випадку містить тільки один незважений вузол $U_{вх}$. Тому сума в формулі (6.1) буде містити єдиний доданок $U_{вих} = W * U_{вх}$. При знаходженні коефіцієнта W враховуємо, що від вузла $U_{вх}$ до вузла $U_{вих}$ веде єдиний шлях і складається з двох гілок з коефіцієнтами передавання G_1 та -1 (рис. 6.10, в). Відповідно коефіцієнт передавання цього шляху визначається рівністю $P_1 = G_1 * (-1)$. Якщо виключити з вихідного графа (рис. 6.10, б) всі вузли, через які проходить цей шлях (**$U_{вх}$, e -, $U_{вих}$**), а також всі гілки, які входять в ці вузли або виходять з них, то залишиться лише один зважений вузол $e+$ (рис. 6.10, в). Таким чином, відповідний визначник даного шляху буде дорівнювати вазі цього вузла: $\Delta_1 = G_3 + G_4$.

Для знаходження визначника Δ повного вихідного графу (рис. 6.10, б) виділимо всі елементарні графи, які містяться в ньому. Ці елементарні графи

наведені на рис. 6.10, г, д, е. Перший з них (рис. 6.10, г) складається з трьох зважених вузлів, а його визначник дорівнює добутку ваг цих вузлів $\delta_1 = (G_1 + G_2) \cdot (G_3 + G_4) \cdot 0$. Другий і третій графи (6.10, д, е) містять по одному контуру та одному зваженому вузлу. Їх визначники можна знайти як добутки взятого з протилежним знаком коефіцієнта передачі контуру і ваги вузла: $\delta_2 = -(-) G_2 \cdot (G_3 + G_4)$; $\delta_3 = -1 \cdot G_4 \cdot (G_1 + G_2)$.

Скориставшись формулою Мезона (6.2) і враховуючи, що $G_k = 1/R_k$, остаточно отримаємо:

$$\frac{U_{\hat{a}\hat{o}}}{U_{\hat{a}\hat{o}}} = \frac{P_1 \cdot \Delta_1}{\delta_1 + \delta_2 + \delta_3} = \frac{-R_2/R_1}{1 - \left(\frac{R_2/R_1 + 1}{R_3/R_4 + 1} \right)}.$$

7. ПЕРЕТВОРЮВАЧІ АНАЛОГОВИХ ТА ЦИФРОВИХ СИГНАЛІВ

Крім чисто «цифрового» сполучення, часто потрібно перетворити аналоговий сигнал в число, пропорційне амплітуді сигналу і навпаки. Це відіграє важливу роль у тих випадках, коли комп'ютер або процесор реєструють або контролюють хід експерименту чи технологічного процесу, або всякий раз, коли цифрова техніка використовується для виконання традиційно аналогової роботи. Аналого-цифрове перетворення (АЦП) використовується в областях, де для забезпечення перешкодостійкої і шумозахищеної передачі, аналогова інформація перетворюється на проміжну цифрову форму (наприклад, «цифрова звукотехніка» або імпульсно-кодова модуляція). Це потрібно в різноманітних вимірювальних засобах (цифрових універсальних вимірювальних приладах, осцилографах з цифровою пам'яттю та інших), а також в пристроях генерації та обробки сигналів, таких, як цифрові синтезатори коливань і пристрої шифрування даних [8, 26-28].

ЦАП - пристрій, який створює на виході аналоговий сигнал (струм та напругу), пропорційний вхідному сигналу. При цьому значення вихідного сигналу залежить від опорної напруги, що визначає повну шкалу вихідного сигналу. Якщо замість опорної напруги використати будь-який аналоговий сигнал, то вихідний сигнал ЦАП буде пропорційний добутку вхідних цифрових та аналогових сигналів.

В АЦП цифровий код на виході визначається відношенням перетворюючого вхідного аналогового сигналу до опорного сигналу, що відповідає повній шкалі. Якщо опорний сигнал змінюється по будь-якому закону, то цифровий сигнал на виході АЦП буде пропорційний відношенню двох аналогових (вхідного та опорного) сигналів. В цьому випадку АЦП можна розглядати як вимірювач відношень або подільник напруги з цифровим виходом.

В залежності від області застосування на вході ЦАП або виході АЦП можуть бути або однополярні цифрові коди, або біполярні. До перших відносяться прямий двійковий та двійково-десятковий коди, до других - двійковий код зі зміщенням. В стандартному двійковому коді старший розряд має вагу $2^{-1} = 1/2$ від значення повної шкали (рис. 7.1). Розряд, що йде за старшим має вагу $2^{-2} = 1/4$ і так далі до молодшого розряду з вагою $1/2^m$, де m - число розрядів перетворювача.

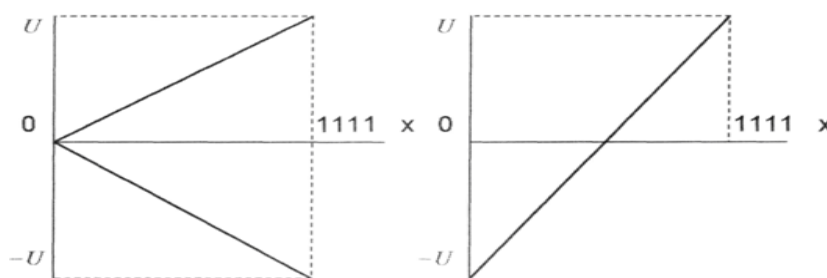


Рисунок 7.1 – Графіки відповідності цифрових кодів та аналогових напруг

Сума всіх розрядів визначає значення повної шкали перетворювача. Двійковий код зі зміщенням відрізняється від прямого тим, що нуль двійкового числа та нуль аналогової величини не співпадають, як в стандартному двійковому коді, а зміщені таким чином, що в старшому розряді встановлюється «0» для всіх від'ємних та «1» для всіх позитивних величин. Тому старший розряд визначає полярність цифрового коду.

З рис. 7.1 видно, що прямий код дає можливість використовувати в два рази більшу роздільність в порівнянні зі зміщенням.

Основними параметрами, що характеризують ЦАП та АЦП є:

Статичні характеристики: нелінійність, монотонність, коефіцієнт перетворення, абсолютна та відносна похибка, зміщення нуля, роздільна здатність.

Динамічні характеристики: час встановлення, час перетворення.

Роздільна здатність визначає число дискретних значень вихідного сигналу перетворювача, що складають його межі перетворення. Може виражатись або в процентах, або в долях до повної шкали, наприклад дванадцятирозрядний АЦП має роздільну здатність $1/4096$ або $0,0245\%$ повної шкали.

Нелінійність характеризується відхиленням значень реальної характеристики перетворювача від прямої. Перетворювач вважається лінійним, якщо його максимальна похибка лінійності не перевищує половини молодшого розряду перетворювача.

Монотонність перетворювача означає, що при кожному приросту вхідного сигналу відбувається приріст вихідного сигналу - перша похідна від неперервної функції вихід-вхід повинна бути меншою нуля.

Коефіцієнт перетворення - це відношення приросту вихідного сигналу до приросту вхідного сигналу, який визначає кут відхилення характеристики. Для ідеального ЦАП відхилення характеристики повинно бути таким, щоб при ввімкненні усіх розрядів вихідна напруга перетворювача була менше опорної напруги на величину молодшого розряду.

Ступінь відхилення реального значення коефіцієнту перетворення від розрахункового характеризується абсолютною похибкою, під якою розуміють різницю між номінальним значенням напруги повної шкали перетворювача та фактичним його значенням.

Похибка нуля (зміщення) для ЦАП - це вихідна напруга ЦАП з нульовим вхідним кодом, а для АЦП - середнє значення вхідної напруги АЦП, яке необхідне для отримання нульового коду на його виході.

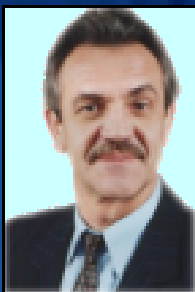
Час встановлення вихідного сигналу ЦАП є важливим динамічним параметром та визначається як інтервал часу, впродовж якого вихідний аналоговий сигнал ЦАП при зміні кодової комбінації на його цифрових входах досягає свого встановленого значення, яке не перевищує граничної похибки.

Розглянуті характеристики допомагають вибрати тип перетворювача для конкретного застосування в системах та пристроях, що розробляються. Практичну реалізацію багаторозрядних схем АЦП та ЦАП раціонально проводити на основі мікросхем, які містять основні блоки перетворювачів в одному корпусі.

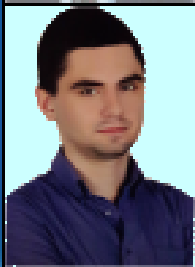
ЛІТЕРАТУРА

1. Хоровиц П. Искусство схемотехники / П. Хоровиц, У. Хил. – М.: Издательство БИНОМ, 2014. – 704 с.
2. Гусев В.Г. Электроника / В.Г. Гусев, Ю.М. Гусев. – М.: Высш.шк.1991. – 622с.
3. Журнал Компьютерра [Электронный ресурс] / Мемристор: недостающий элемент. – Режим доступа: <https://old.computerra.ru/vision/591537/>.
4. Титце У. Напівпровідникова схемотехніка / У. Титце, К. Шенк. Т1. – М. : Додэка-XXI, 2008. – 942с.
5. Степаненко И.П. Основы теории транзисторов и транзисторных схем / И.П. Степаненко. – М. : Энергия. 1977. – 607с.
6. Угрюмов Е.П. Цифровая схемотехника / Е.П. Угрюмов. – СПб.: Питер, 2010. – 809 с.
7. Гутников В.С. Интегральная электроника в измерительных устройствах / В.С. Гутников. – Л.: Энергоатомиздат, 1988. – 304 с.
8. Кривогубченко С.Г. Интегральна схемотехніка: навчальний посібник / С.Г. Кривогубченко. – Вінниця: ВДТУ, 1999. – 113 с.
9. Степанов А.Н. Архитектура вычислительных систем и компьютерных сетей / А.Н. Степанов. – СПб.: Питер, 2007 – 509 с.
10. Таненбаум Э. Архитектура компьютера / Э. Таненбаум. – СПб.: Питер, 2002. – 440 с.
11. Аладьев В.З. Основы информатики / В.З. Аладьев, Ю.Я. Хунт, М.Л. Шишаков. – М.: Филинь, 1999. – 544 с.
12. Андреева Е.В. Системы счисления и компьютерная арифметика / Е.В. Андреева, И.Н. Фалина. – М.: Лаборатория базовых знаний, 2000. – 245 с.
13. Архитектура компьютерных систем и сетей / Т.П. Барановская, В.И. Лойко, М.И. Семенов, А.И. Трубилин. – М.: Финансы и статистика, 2003. – 256 с.
14. Вернер М. Основы кодирования / М. Вернер. – М.: Техносфера, 2004. – 283 с.
15. Арифметичні основи проектування мікропроцесорних систем: навчальний посібник / Р.Н. Кветний, П.П. Повідайко, М.М. Компанець, В.В. Гармаш, Я.А. Кулик. – Вінниця: ВНТУ, 2017. – 111 с.
16. Кулик А.Я. Теорія інформації і кодування: навчальний посібник / А.Я. Кулик, С.Г. Кривогубченко. – Вінниця: ВНТУ, 2008. – 145 с.
17. Функціональні перетворювачі систем автоматики і управління: навчальний посібник / С.Г. Кривогубченко, А.Я. Кулик, М.М. Компанець, А.Ф. Хомчук. – Вінниця: ВНТУ, 2011. – 185 с.
18. Партала О.Н. Цифровая электроника / О.Н. Партала. – СПб.: Наука и техника, 2002. – 224 с.
19. Бабич Н.П. Основы цифровой схемотехники / Н.П. Бабич, И.А. Жуков. – М.: Издательство Додэка – XXI, 2007. – 224 с.

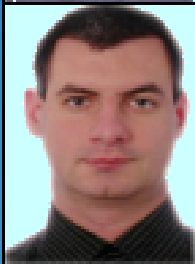
20. Угрюмов Е.П. Цифровая схемотехника / Е.П. Угрюмов. – СПб.: БХВ – Петербург, 2004. – 528 с.
21. Новиков Ю.В. Основы цифровой схемотехники. Базовые элементы и схемы. Методы проектирования / Ю.В.Новиков. – М.:Мир, 2001. – 379 с.
22. Арсеньев Ю.Н. Проектирование систем логического управления на микропроцессорных средствах. / Ю.Н. Арсеньев, В.М. Журавлев – М.: Высшая школа, 1991. – 319 с.
23. Зубчук В.И. Справочник по цифровой схемотехнике / В.И. Зубчук, В.П. Сигорский, А.Н. Шкуро. – К.: Техника, 1990. – 448 с.
24. Хаїмзон І.Я. Техніка передачі інформації. Функціональні вузли та схеми: навчальний посібник / І.Я. Хаїмзон. – Вінниця: ВДТУ, 2000. – 143 с.
25. Волович Г.И. Схемотехника аналоговых и аналого-цифровых электронных устройств / Г.И. Волович. – М.: Издательство Додэка – XXI, 2008. – 528 с.
26. Гусев В.Г. Электроника и микропроцессорная техника / В.Г. Гусев. – М.: Высшая школа, 2005. – 237 с.
27. Лаврентьев Б.Ф. Аналоговая и цифровая электроника / Б.Ф. Лаврентьев. – Йошкар-Ола: МарГТУ, 2000. – 155 с.
28. Никамин В.А. Аналого-цифровые и цифро-аналоговые преобразователи / В.А. Никамин. – М.: Техносфера, 2003. – 145 с.



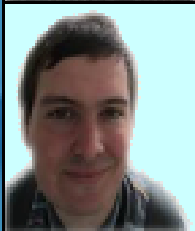
Кривоzubченко Сергій Григорович – канд. техн. наук,
доцент кафедри АІВТ ВНТУ.



Іванов Юрій Юрійович – канд. техн. наук, старший
викладач кафедри АІВТ ВНТУ.



Кривоzubченко Денис Сергійович – канд. техн. наук,
доцент.



Кулик Ярослав Анатолійович – канд. техн. наук,
старший викладач кафедри АІВТ ВНТУ.



Папінюв Володимир Миколайович – канд. техн. наук,
доцент кафедри АІВТ ВНТУ.

Колектив авторів включає провідних та молодих викладачів кафедри Автоматики та інформаційно-вимірювальної техніки Вінницького національного технічного університету, область наукових інтересів яких пов'язана з комп'ютерною електронікою, цифровими системами передавання даних, алгоритмами передавання інформації, стисненням мультимедійних даних, завадостійким кодуванням, вейвлет-аналізом, мережевими технологіями, математичними методами оптимізації тощо.

Загальний науковий та навчально-методичний доробок авторів складає 400 робіт, включаючи праці, що входять до фахових видань та світових науково-метричних баз даних Scopus, SPIE, Index Copernicus, IEEE Xplore, PII Science Index; патенти та свідоцтва на авторське право; монографії та посібники. Автори мають великий досвід роботи зі спеціалізованим програмним забезпеченням для моделювання різних систем.

