

Міністерство освіти і науки України
Інженерний навчально-науковий інститут
Запорізького національного університету

Верьовкін Л.Л.

ФУНКЦІОНАЛЬНІ ВУЗЛИ МІКРОПРОЦЕСОРНИХ СИСТЕМ

Методичні рекомендації до практичних занять
для здобувачів вищої освіти першого бакалаврського рівня
за спеціальністю 153 «Мікро- та наносистемна техніка» освітньо-професійної
програми «Мікро- та наносистемна техніка»

Міністерство освіти і науки України
Інженерний навчально-науковий інститут
Запорізького національного університету

Верьовкін Л.Л.

ФУНКЦІОНАЛЬНІ ВУЗЛИ МІКРОПРОЦЕСОРНИХ СИСТЕМ

Методичні рекомендації до практичних занять
для здобувачів вищої освіти першого бакалаврського рівня
за спеціальністю 153 «Мікро- та наносистемна техніка» освітньо-професійної
програми «Мікро- та наносистемна техніка»

Затверджено
Вченою
радою ЗНУ
Протокол № 11
від 23.06.2021

Запоріжжя
2021

УДК 621.38

В

Верьовкін Л.Л. Функціональні вузли мікропроцесорних систем. Методичні рекомендації до практичних занять для здобувачів вищої освіти першого бакалаврського рівня за спеціальністю 153 «Мікро- та наносистемна техніка» освітньо-професійної програми «Мікро- та наносистемна техніка». Запоріжжя : ЗНУ, 2021. 45 с.

Методичні рекомендації до практичних занять з дисципліни «Функціональні вузли мікропроцесорних систем» розроблено для студентів денної та заочної форм навчання інженерно-технічних спеціальностей та освітньо-професійних програм Інженерного навчально-наукового інституту. Запропоновані приклади розв'язання завдань по фізичним основам роботи, конструкції, технології виготовлення та основним параметрам оптоелектронних приладів та цифрових логічних систем.

Рецензент

Коваленко В.Л., доктор технічних наук, професор кафедри електротехніки та енергоефективності

Відповідальний за випуск

Т.В. Критська, доктор технічних наук, професор, завідувач кафедри мікроелектронних та електронних інформаційних систем.

Зміст

| | |
|--|----|
| Вступ | 4 |
| 1 Загальні принципи побудови мікропроцесорних систем | 6 |
| 1.1. Основні поняття і визначення | 6 |
| 1.2 Принципи побудови мікропроцесорних систем | 7 |
| 1.3 Архітектура мікропроцесорів | 7 |
| 1.4 Однокристальні мікропроцесори | 9 |
| 2 Комбінаційні функціональні вузли мікропроцесорної техніки | 10 |
| 2.1 Дешифратор | 10 |
| 2.2 Демультіплексор | 12 |
| 2.3 Мультиплексор | 13 |
| 2.4 Шифратор | 14 |
| 2.5 Перетворювачі кодів | 17 |
| 2.6 Схема порівняння чисел | 22 |
| 2.7 Схема контролю парності (непарності) | 23 |
| 2.8 Мажоритарні елементи | 24 |
| 2.9 Порогова логічна схема | 26 |
| <u>Контрольна робота №1</u> | 28 |
| 3 Послідовнісні функціональні вузли мікропроцесорної техніки | 33 |
| 3.1 Цифрові регістри. Пристрої накопичення інформації | 33 |
| 3.2 Цифрові лічильники | 34 |
| 3.3 Генератори числових послідовностей | 36 |
| 4 Арифметичні логічні пристрої | 36 |
| 4.1 Одноразрядні суматори | 38 |
| 4.2 Комбінаційні суматори арифметичних логічних пристроїв | 40 |
| <u>Контрольна робота №2</u> | 43 |
| Список рекомендованої літератури | 47 |

Вступ

Метою викладання навчальної дисципліни «Функціональні вузли мікропроцесорних систем» є формування повної уяви про принципи функціонування та розрахунки складових вузлів мікроконтролерів, будовання мікропроцесорних систем та їх використання в електронних приладах.

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

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

- здатність розв'язувати складні спеціалізовані задачі та практичні проблеми, що характеризуються комплексністю та невизначеністю умов, під час професійної діяльності у галузі мікро- та наносистемної техніки, або у процесі навчання, що передбачає застосування теорій та методів автоматизації та електроніки;

- здатність застосовувати знання у практичних ситуаціях;

- знання та розуміння предметної області та розуміння професійної діяльності;

- навички використання інформаційних і комунікаційних технологій;

- здатність до пошуку, оброблення та аналізу інформації з різних джерел;

- здатність оцінювати та забезпечувати якість виконуваних робіт;

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

- здатність виконувати аналіз предметної області та нормативної документації, необхідної для проектування та застосування приладів та пристроїв мікро- та наносистемної техніки.

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

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

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

Методичні вказівки містять індивідуальні контрольні завдання з курсу «Функціональні вузли мікропроцесорних систем» та приклади розв'язання завдань з кожної теми. Для підготовки до лекцій, практичних занять та самостійної проробки матеріалу студенти можуть використати приведені теоретичні відомості з курсу.

1. Загальні принципи побудови мікропроцесорних систем

1.1 Основні поняття і визначення

Мікропроцесор (МП) – це пристрій, який здійснює прийом, обробку і видачу інформації. Конструктивно МП містить одну або декілька інтегральних схем і виконує дії за програмою, записаною в пам'яті [1].

Мікропроцесорна система – обчислювальна, контрольно-вимірювальна або керуюча система, в якій основним пристроєм обробки інформації є МП. Мікропроцесорна система будується з набору мікропроцесорних ВІС.

Мультимікропроцесорна (або мультипроцесорна) система – система, яка утворюється об'єднанням деякої кількості універсальних або спеціалізованих МП, завдяки чому забезпечується паралельна обробка інформації і розподілене керування.

Мікропроцесорний комплект (МПК) [1] – сукупність інтегральних схем, сумісних за електричними, інформаційними та конструктивними параметрами і призначених для побудови електронно-обчислювальної апаратури та мікропроцесорних систем керування. Звичайно МПК містить: ВІС МП (один чи кілька корпусів інтегральних схем); ВІС оперативних запам'ятовувальних пристроїв (ОЗП); ВІС постійних запам'ятовувальних пристроїв (ПЗП); інтерфейси або контролери зовнішніх пристроїв; службові ВІС (тактовий генератор, регістри, шинні формувачі, контролери шин, арбітри шин).

Мікропроцесори та МПК класифікують за такими ознаками: призначенням; кількістю ВІС; способом керування; за типом архітектури; за типом системи команд [1].

Шина [1] – це інформаційний канал, який об'єднує всі функціональні блоки МПС і забезпечує обмін даними у вигляді двійкових чисел. Конструктивно шина являє собою n провідників та один спільний провідник (земля). Дані по шині передаються у вигляді слів, що є групою біт.

У паралельній шині n біт передаються по окремим лініям одночасно, у послідовній шині – по єдиній лінії послідовно у часі.

1.2 Принципи побудови мікропроцесорних систем

В основу побудови МПС систем покладено три принципи: 1) магістральності; 2) модульності; 3) мікропрограмного керування [1].

Принципи магістральності і модульності дозволяють нарощувати керуючі і обчислювальні можливості МП через під'єднання інших модулів.

Принцип мікропрограмного керування полягає у можливості здійснення елементарних операцій – мікрокоманд (зсуву, пересилки інформації, логічних операцій). Певною комбінацією мікрокоманд можна створити набір команд, який максимально відповідатиме призначенню системи, тобто створити технологічну мову. У секційних процесорах набір мікрокоманд можна змінити, використовуючи інші мікросхеми пам'яті мікрокоманд [1].

1.3 Архітектура мікропроцесорів

Поняття архітектури мікропроцесора визначає його складові частини, а також зв'язки та взаємодію між ними. Архітектура містить: 1) структурну схему самого МП; 2) програмну модель МП (описання функцій регістрів); 3) інформацію про організацію пам'яті (ємність пам'яті та способи її адресації); 4) опис організації процедур введення-виведення [1].

Існують два основних типи архітектури – фоннейманівська та гарвардська. Фоннейманівську архітектуру (рис. 1.1 а) запропонував 1945 року американський математик Джо фон Нейман.

Особливістю цієї архітектури є те, що програма і дані знаходяться у спільній пам'яті, доступ до якої здійснюється по одній шині даних і команд.

Гарвардську архітектуру вперше реалізовано у 1944 році в релейній обчислювальній машині Гарвардського університету (США).

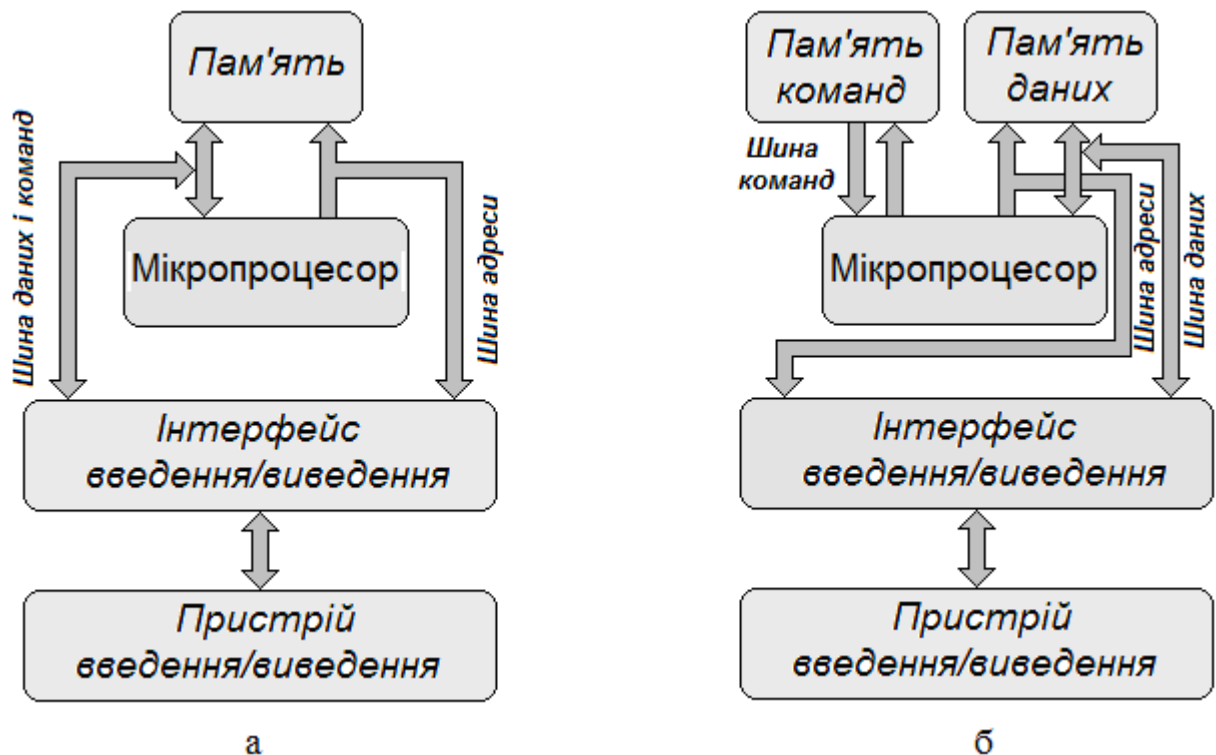


Рисунок 1.1 – Основні типи архітектури: а – фоннейманівська; б – гарвардська

Особливістю цієї архітектури є те, що пам'ять даних і пам'ять програм розділені і мають окремі шини даних і шини команд (рис. 1.1 б), що дозволяє підвищити швидкодію МП системи .

Структурні схеми обох архітектур містять: процесорний елемент, пам'ять, інтерфейси введення/виведення (ІВВ) і пристрої введення/виведення (ПВВ). Пам'ять і ІВВ для різних типів МП можуть бути як внутрішніми, тобто розміщуватися на тому ж кристалі, що і процесорний елемент, так і зовнішніми. Процесорний елемент містить регістри, арифметично-логічний пристрій (АЛП), пристрій керування і виконує функції обробки даних та керування процесами обміну інформацією. Пам'ять забезпечує зберігання кодів команд програми і даних. Інтерфейси призначені для зв'язку з ПВВ (наприклад, з клавіатурою, дисплеєм, печатними пристроями, датчиками інформації). Усі елементи структурної схеми з'єднані за допомогою шин.

Схема процесора містить пристрій керування, АЛП і регістри: адреси, даних, команд, а також стану, акумулятор, лічильник команд та вказівник стека.

1.4 Однокристальні мікропроцесори

Структурну схему узагальненого 8-розрядного однокристального МП показано на рисунку 1.2. Схема має єдину внутрішню 8-розрядну шину, по якій передаються дані, коди команд та адреси [1].

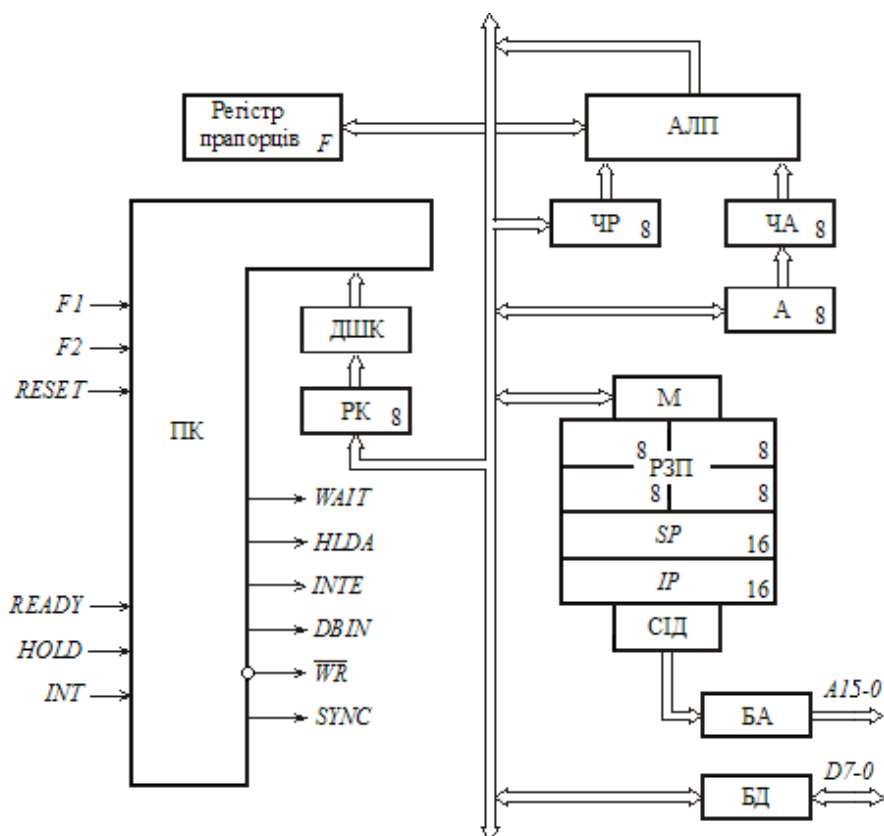


Рисунок 1.2 – Структурна схема 8-розрядного мікропроцесора

Структурна схема містить: пристрій керування ПК, дешифратор команд ДШК, регістр команд РК, арифметично-логічний пристрій АЛП, акумулятор А, часовий акумулятор ЧА, часовий регістр ЧР, регістр прапорців F, блок 8-розрядних регістрів загального призначення РЗП, мультиплексор, вказівник стека (Stack Pointer SP), вказівник команд (Instruction Pointer IP), буферний регістр адреси БА, буферний регістр даних БД, схему інкременту/декременту СІД.

2 Комбінаційні функціональні вузли мікропроцесорної техніки

2.1 Дешифратор

Дешифратор (ДШ) перетворює двійковий код на входах в активний сигнал на тому виході, номер якого дорівнює десятковому еквіваленту двійкового коду на входах [2]. У повному дешифраторі кількість виходів $m = 2^n$, де n - число входів. У неповному ДШ m менше ніж 2^n . За визначенням повний ДШ повинен генерувати 2^n вихідних логічних функцій (ЛФ), визначених на всіх наборах з n - вхідними змінними, тобто мінтермів. Розглянемо ДШ з $n = 2$ і $m = 4$, що зветься також дешифратором "2 в 4" і доповнимо його входом дозволу виходів OE (Output Enable). Активним рівнем сигналу на прямих входах/виходах буде 1, а на інверсних - 0. За цим визначенням заповнимо таблицю істинності (табл. 2.1), де величина (×) може набувати будь-яких значень.

Таблиця 2.1 – Таблиця функціонування дешифратора "2 в 4"

| DEC число | Входи | | | | Виходи | | | | ЛФ | |
|--------------|-------|---|----|------------------------|--------|----|----|----|---------|---|
| | A | B | OE | $\overline{\text{OE}}$ | F0 | F1 | F2 | F3 | | $\overline{\text{F0}}$ $\overline{\text{F1}}$ $\overline{\text{F2}}$ $\overline{\text{F3}}$ |
| 0 | 0 | 0 | | | 1 | 0 | 0 | 0 | 0 1 1 1 | $F_0 = \text{OE} \cdot \overline{A} \cdot \overline{B}$ |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 0 1 1 | $F_1 = \text{OE} \cdot \overline{A} \cdot B$ |
| 2 | 1 | 0 | | | 0 | 0 | 1 | 0 | 1 1 0 1 | $F_2 = \text{OE} \cdot A \cdot \overline{B}$ |
| 3 | 1 | 1 | | | 0 | 0 | 0 | 1 | 1 1 1 0 | $F_3 = \text{OE} \cdot A \cdot B$ |
| × | × | × | 0 | 1 | 0 | 0 | 0 | 0 | 1 1 1 1 | $F_i=0 \quad \overline{F}_i=1$ |

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

| | | | | | |
|----|--|----|---|---|---|
| | | F0 | | | |
| | | A | | | |
| OE | | 0 | 0 | 0 | 0 |
| | | 1 | 0 | 0 | 0 |
| | | B | | | |

Логічна функція виходу F0 матиме вигляд: $F_0 = \text{OE} \cdot \overline{A} \cdot \overline{B}$. Аналогічно отримано останні три рівняння. Перетворимо отримані для F_i рівняння за до-

помогою аксіоми подвійного заперечення до базису I-НІ: $F_0 = \overline{\overline{OE} \cdot \overline{A} \cdot \overline{B}}$. Рішенням відповідають схема на рисунку 2.1 і діаграма функціонування на рисунку 2.2.

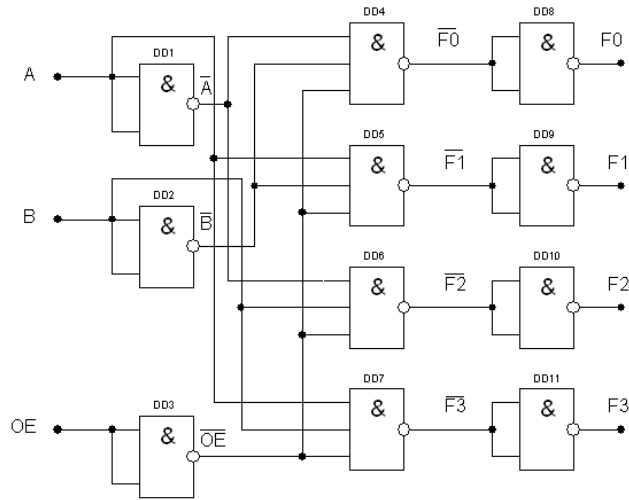


Рисунок 2.1 – Розрахована схема дешифратора

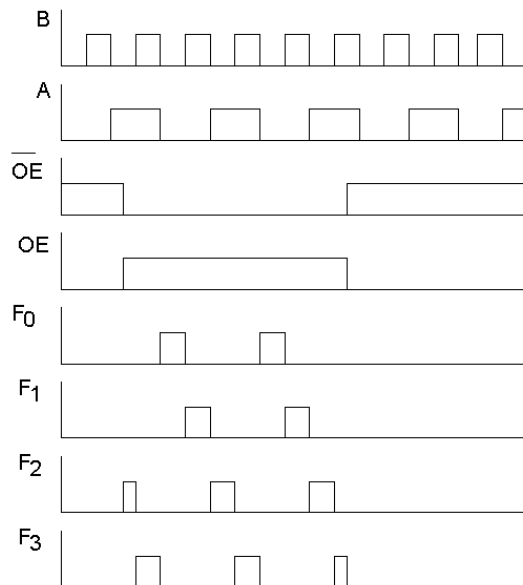


Рисунок 2.2 – Діаграма функціонування дешифратора

Можливі комбінації прямих і інверсних входів і виходів дешифраторів.

Дешифратори по структурі можуть бути лінійними, пірамідальними, матричними [2 – 4].

2.2 Демультимплексор

Демультимплексором (ДМ) називається пристрій, що передає сигнал з інформаційного входу на один з виходів, причому номер цього виходу дорівнює десятковому еквіваленту двійкового коду на адресних входах [2]. Як ДМ може використовуватися дешифратор, в якого замість сигналу OE подається інформаційний сигнал x . Наприклад, якщо на входи подати код $AB = 10_{(\text{BIN})} = 2_{(\text{DEC})}$, то сигнал x з'явиться на виході F_2 , а на останніх виходах $F_i = 0$. На рисунку 2.3. представлено схемний аналог ДМ "1 в 4".

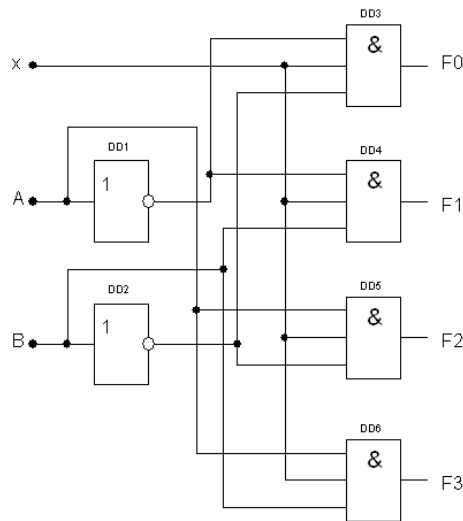


Рисунок 2.3 – Демультимплексор

2.3 Мультиплексор

Мультиплексор передає сигнал з одного з інформаційних входів x_i на єдиний вихід Q , причому номер цього входу дорівнює десятковому еквіваленту двійкового коду на адресних входах D_i [2 – 4]. Якщо є вхід дозволу виходу OE, то "0" на цьому вході повинен перевести вихід в пасивний стан (остання строчка таблиці 2.2). Розглянемо мультиплексор "4 в 1", що має 4 інформаційних входи і $\log_2 4 = 2$ адресних входи.

Аналіз 0 рядку, наводить до виводу, що $Q = X_0$, якщо $A = 0$ і $B = 0$ і $OE = 1$, незалежно від змінних $X_1 \dots X_3$. Тому для цього вхідного набору можна записати: $Q = X_0 \cdot OE \cdot \bar{A} \cdot \bar{B}$.

Таблиця 2.2 – Таблиця функціонування мультиплексора

| DEC число | Входи | | | | | | | | Вихід Q | ЛФ |
|--------------|-------|---|----|-----------------|----|----|----|----|------------|--|
| | A | B | OE | \overline{OE} | D0 | D1 | D2 | D3 | | |
| 0 | 0 | 0 | | | X0 | × | × | × | X0 | $Q = X0 \cdot OE \cdot A \cdot B$ |
| 1 | 0 | 1 | 1 | 0 | × | X1 | × | × | X1 | $Q = X1 \cdot OE \cdot A \cdot \overline{B}$ |
| 2 | 1 | 0 | | | × | × | X2 | × | X2 | $Q = X2 \cdot OE \cdot \overline{A} \cdot B$ |
| 3 | 1 | 1 | | | × | × | × | X3 | X3 | $Q = X3 \cdot OE \cdot A \cdot \overline{B}$ |
| × | × | × | 0 | 1 | × | × | × | × | 0 | $Q = 0$ |

Аналогічно записується в для останніх трьох наборів змінних. Загальне рішення тоді матиме вигляд:

$$Q = OE(X0\overline{A}\overline{B} + X1\overline{A}B + X2A\overline{B} + X3AB)$$

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

$$Q = \overline{\overline{OE X0\overline{A}\overline{B}} \cdot \overline{OE X1\overline{A}B} \cdot \overline{OE X2A\overline{B}} \cdot \overline{OE X3AB}}$$

Рішенню відповідає схема, приведена на рисунку 2.4.

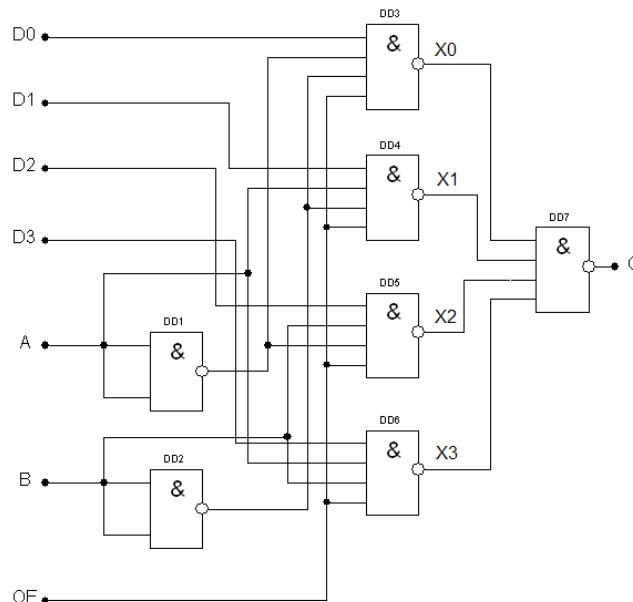


Рисунок 2.4 – Схема мультиплексора

Якщо на адресні входи подати комбінацію $AB = 11_{(BIN)} = 3_{(DEC)}$, то до виходу Q буде підключений вхід D3, за умови, якщо $OE = 1$. Діаграма функціонування мультиплексора представлена на рисунку 2.5.

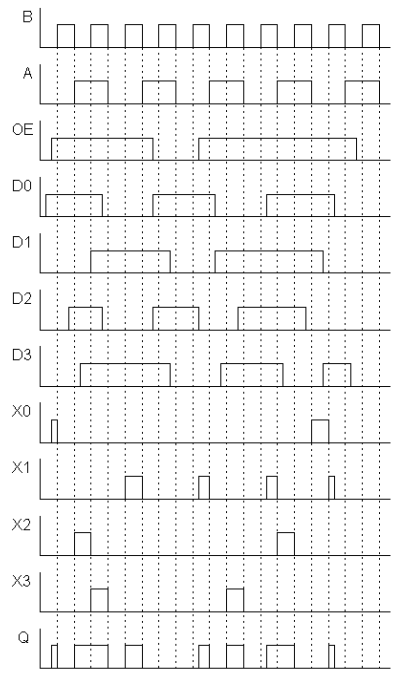


Рисунок 2.5 - Діаграма функціонування мультиплексора

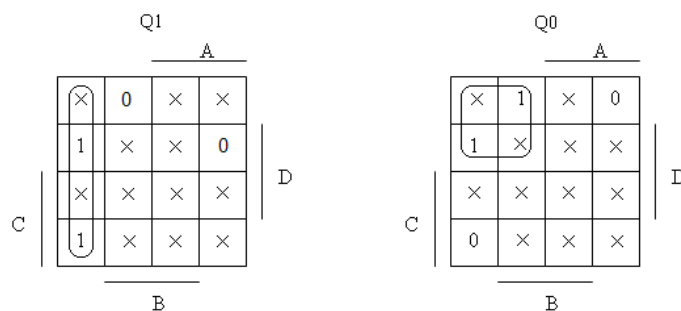
2.4 Шифратор

Шифратор – це комбінаційний пристрій, що перетворює десяткові числа в двійкову систему числення, причому кожному входу може бути поставлено в відповідь десяткове число, а набір вихідних логічних сигналів відповідає певному двійковому коду [2]. Шифратор (Ш) може бути неперіоритетним, якщо допускається подача лише одного активного сигналу і може бути пріоритетним, якщо допускається подача одночасно декількох активних сигналів на входи. Неперіоритетний Ш здійснює перетворення десяткового номера активного входу в двійковий еквівалент цього номера. Таблиця істинності неперіоритетного шифратора "4 в 2" представлена в таблиці 2.3.

Таблиця 2.3 – Таблиця істинності неперіоритетного шифратора "4 в 2"

| № входу | Входи | | | | Виходи | |
|---------|-------|---|---|---|--------|----|
| | A | B | C | D | Q1 | Q0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 1 | 1 | 1 |

Для синтезу схеми неперіоритетного Ш для кожного виходу складемо таблицю Карно. Комбінація 0000 на входах не визначена. Чотири вхідних змінних дають $2^4 = 16$ комбінацій з яких за визначенням задані в таблиці лише 4. Останні 12 невизначених (заборонених) комбінацій в таблицях Карно відзначимо символом (×). Оскільки поява цих комбінацій на входах не передбачена (за визначенням), то у відповідні клітки можна підставляти будь-які значення, у тому числі такі, які дозволяють якнайповніше мінімізувати ЛФ:



У обох випадках дві величини (×) довизначено до 1. З приведених таблиць знаходимо Q1 і Q0:

$$Q1 = \bar{A} \cdot \bar{B} = \bar{A} + \bar{B} \quad \text{и} \quad Q0 = \bar{A} \cdot \bar{C} = \bar{A} + \bar{C}$$

Рішенню відповідає схема, приведена на рисунку 2.6.

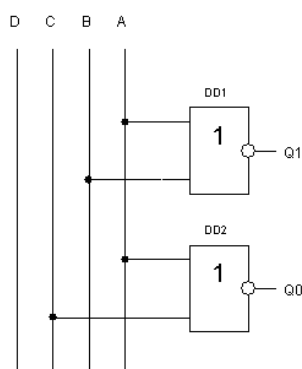


Рисунок 2.6 – Схема неперіоритетного шифратора "4 в 2"

У пріоритетному Ш виробляється перетворення максимального десятичного номера активного входу в двійковий еквівалент цього номера [2]. Для такого Ш вхідні сигнали, лежачі знизу від одиничної діагоналі, за визначенням не відомі ("×" може бути 0 або 1) (табл.2.4).

Таблиця 2.4 – Таблиця істинності пріоритетного шифратора "4 в 2"

| Входи | | | | | Виходи | | | |
|-------|---|---|---|---|--------|----|---|----|
| E1 | A | B | C | D | Q1 | Q0 | G | E0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | × | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | × | × | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | × | × | × | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | × | × | × | × | 0 | 0 | 0 | 0 |

Активним рівнем на входах і виходах, є логічна 1. Схема представляє собою пріоритетний шифратор 4×2 , тобто має 4 прямих входа і 2 прямих вихода. Вона має вхід дозволу E1, вихід перенесення E0 і вихід G, що визначає ознаку вхідного інформаційного сигналу. Комбінація 0000 на входах не визначена. Згідно таблиці функціонування пріоритетного шифратора для виходів A1, A0, G, E0:

$$Q1 = (D + C\bar{D})E1;$$

$$Q0 = (D + B\bar{C}\bar{D})E1;$$

$$G = (D + C\bar{D} + B\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D})E1;$$

$$E0 = (\bar{A}\bar{D}\bar{C}\bar{D})E1.$$

Повторним застосуванням до кожної функції Q_i ($i = 2, 1, 0$) співвідношення алгебри логіки:

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

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

$$Q1 = (D + C)E1;$$

$$Q0 = (D + B\bar{C})E1;$$

$$G = (D + C + B + A)E1;$$

$$E0 = (\bar{A}\bar{D}\bar{C}\bar{D})E1.$$

Схема пріоритетного шифратора згідно рівнянням функціонування представлена на рисунку 2.7.

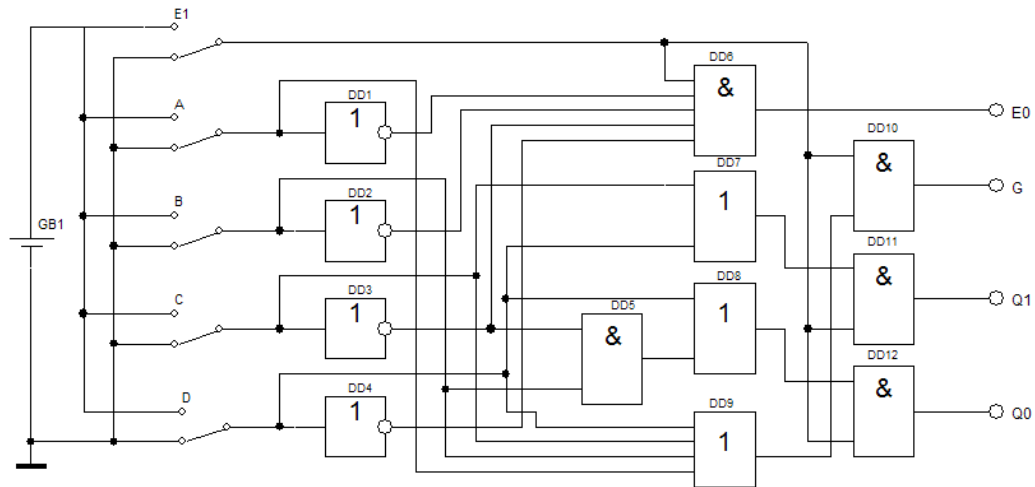


Рисунок 2.7 – Схема пріоритетного шифратора 4×2

2.5 Перетворювачі кодів

Перетворювачі кодів (ПК) бувають прості і складні. До простих відносяться перетворювачі, які виконують стандартні операції зміни кодів чисел, наприклад, перетворень двійкового коду в одинарний або зворотну операцію. Складні перетворювачі коду виконують нестандартні перетворення коду і їх схеми доводиться розробляти кожен раз за допомогою алгебри логіки [5].

По своїй структурі перетворювачі коду є дешифраторами, але вони перетворюють двійковий код в сигнали не лише на одному, але і на декількох виходах. Вважатимемо, що перетворювачі кодів мають n входів і k виходів. Співвідношення між n і k можуть бути будь-якими: $n = k$, $n < k$ і $n > k$. При перетворенні кодів чисел з ними можуть виконуватися різні додаткові операції, наприклад, множення на вагові коефіцієнти.

Перетворювачі кодів можуть бути ваговими і неваговими [5]. Вагові ПК перетворюють інформацію з однієї системи числення в іншу. Основне призначення невагових - перетворення інформації для її подальшого відображення.

Як приклад невагового розглянемо перетворювач двійково-десятькового коду в код для семисегментних світлодіодних індикаторів (рис. 2.8). На рисунку також приведені зображення цифр.

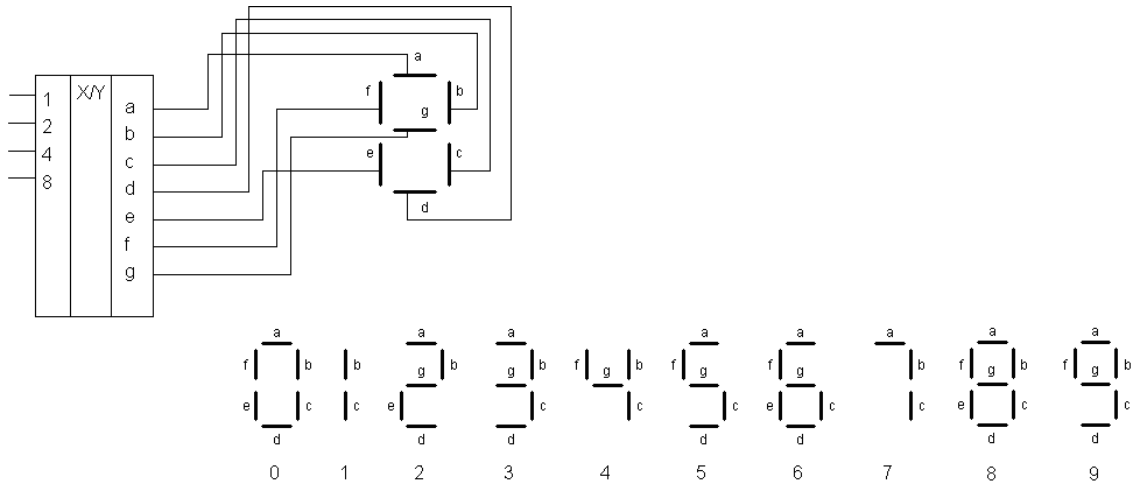


Рисунок 2.8 - Перетворювач двійково-десятькового коду в код для семи-сегментних світлодіодних індикаторів

Такий перетворювач повинен мати чотири входи, оскільки для кодування десятичних цифр від 0 до 9 досить чотири двійкових, і сім виходів, по одинці на кожен сегмент, представлена в таблиці 2.5 [5].

Таблиця 2.5 – Таблиця істинності перетворювача

| Десятична цифра | Входи | | | | Виходи | | | | | | |
|--------------------|--------|--------|--------|--------|--------|---|---|---|---|---|---|
| | 8 A | 4 B | 2 C | 1 D | a | b | c | d | e | f | g |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

Відповідно до таблиці істинності перетворювача, наприклад в цифрі 0 повинні світитися всі сегменти за винятком сегменту g. У цифрі 1 світяться лише два сегменти b та c і так далі. Вагові коефіцієнти b_i двійково-десяткових розрядів рівні 2^i (8,4,2,1). Нулі проставлені для тих цифр, в яких сегмент не світиться.

У загальному випадку для синтезу цього ПК потрібно скласти сім рівнянь. Знайдемо одне, для сегменту a, заповнивши спочатку для нього карту Карно:

Fa

| | | | | |
|---|---|---|---|---|
| | | A | | |
| | | 0 | x | |
| C | 1 | 0 | x | 1 |
| | 0 | 1 | x | 1 |
| | 1 | 1 | x | x |
| | 1 | 1 | x | x |
| | | B | | D |

Рівняння функціонування сегмента a:

$$F_a = \overline{A}C + A\overline{B}C + \overline{A}BD + \overline{B}CD$$

Як приклад вагового розглянемо перетворення двійкового коду в зворотній код [5]. Необхідно побудувати таблицю функціонування (табл. 2.6).

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

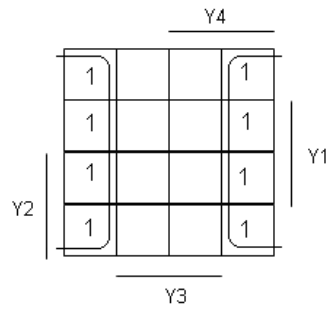
$$A1 = \overline{Y4}Y3\overline{Y2}Y1 + \overline{Y4}Y3Y2\overline{Y1} + \overline{Y4}Y3\overline{Y2}Y1 + \overline{Y4}Y3Y2Y1 + Y4\overline{Y3}\overline{Y2}Y1 + Y4\overline{Y3}Y2\overline{Y1} + Y4Y3\overline{Y2}Y1 + Y4Y3Y2Y1$$

| | | | | | | |
|----|---|----|---|---|---|----|
| | | Y4 | | | | |
| | | 1 | 1 | 1 | 1 | |
| Y2 | | | | | | Y1 |
| | | | | | | |
| | 1 | 1 | 1 | 1 | | |
| | | Y3 | | | | |

Після спрощення отримуємо:

$$A1 = \overline{Y1}$$

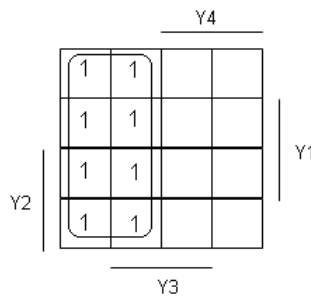
21



Після спрощення отримуємо:

$$A3 = \overline{Y3}$$

$$A3 = \overline{Y4}Y3\overline{Y2}Y1 + \overline{Y4}Y3Y2\overline{Y1} + \overline{Y4}Y3Y2Y1 + \overline{Y4}Y3\overline{Y2}Y1 + \overline{Y4}Y3Y2Y1 + \overline{Y4}Y3\overline{Y2}Y1 + \overline{Y4}Y3Y2Y1 + \overline{Y4}Y3Y2Y1$$



Після спрощення отримуємо:

$$A4 = \overline{Y4}$$

Згідно з отриманими рівняннями функціональна схема перетворювача двійкового коду в зворотний код представлена на рисунку 2.9.

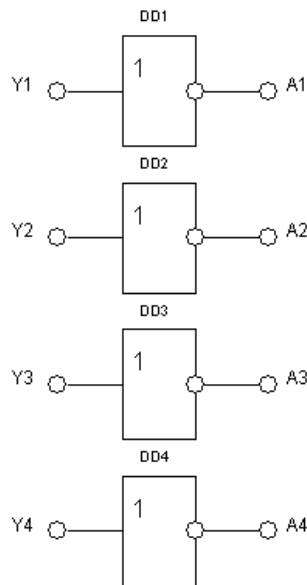


Рисунок 2.9 - Функціональна схема перетворювача двійкового коду в зворотній код

2.6 Схема порівняння чисел

Два кода X і Y вважаються рівними, якщо попарно рівні їх однойменні розряди [5]. Функція $F(X = Y)$ дорівнює 1, якщо $x_i = y_i$ для всіх i , інакше її значення дорівнює нулю. Як приклад (табл. 2.7) візьмемо два двобітових числа $X = (A B)$ і $Y = (C D)$.

Таблиця 2.7 – Таблиця станів кодів X і Y

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

Таблиця Карно для випадку $F(X = Y)$:

| | | | | | |
|---|---|---|---|---|---|
| | A | | | | |
| | 1 | | | | |
| | | 1 | | | |
| | | | 1 | | |
| | | | | 1 | |
| | B | | | | |
| C | | | | | D |

Рівняння функціонування пристрою порівняння:

$$\begin{aligned}
 F(X == Y) &= \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + A\overline{B}\overline{C}\overline{D} + ABCD = \\
 &= \overline{B}\overline{D}(\overline{A}\overline{C} + AC) + BD(\overline{A}\overline{C} + AC) = \\
 &= (\overline{A}\overline{C} + AC) \cdot (\overline{B}\overline{D} + BD) = \overline{(A \oplus C)} \cdot \overline{(B \oplus D)} = \overline{(A \oplus C) + (B \oplus D)}
 \end{aligned}$$

Перетворення в останній двох рівняннях зроблені з обліком, того що $\overline{\overline{F6(x, y)}} = F9(x, y)$ і навпаки. Одна з можливих схем реалізацій приведена на рисунку 2.10.

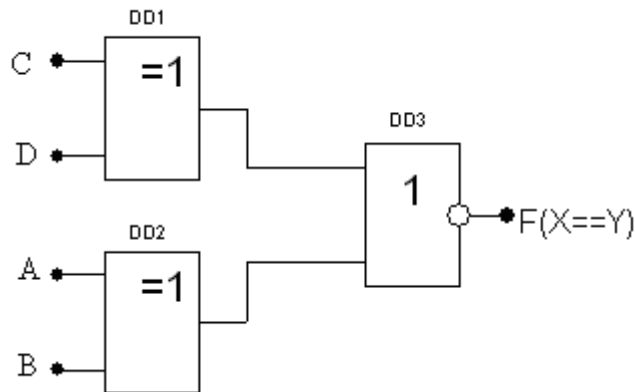


Рисунок 2.10 – Схема порівняння чисел

Практичні схеми доповнюються функціями "більше/менше" [5] (табл. 2.7), як наприклад в мікросхемі 555СП1, яка порівнює два чотирьохрозрядних числа.

Схема порівняння входить до складу АЛП мікропроцесора.

2.7 Схема контролю парності (непарності)

Схема застосовується для виявлення одиночних помилок, викликаних перешкодами в лінії зв'язку або в блоках пам'яті. Метод заснований на підрахунку числа одиниць, що передається в лінію або такій, що направляється в пам'ять на зберігання порції інформації, причому, якщо число одиниць парне – функція парності P (Parity) дорівнює нулю [5].

Розглянемо n-провідну лінію зв'язку (рис. 2.11) по якій передається паралельний двійковий код $x(n-1), x(n-2), \dots, x_1, x_0$, а приймається код $x'(n-1), x'(n-2), \dots, x'_1, x'_0$.

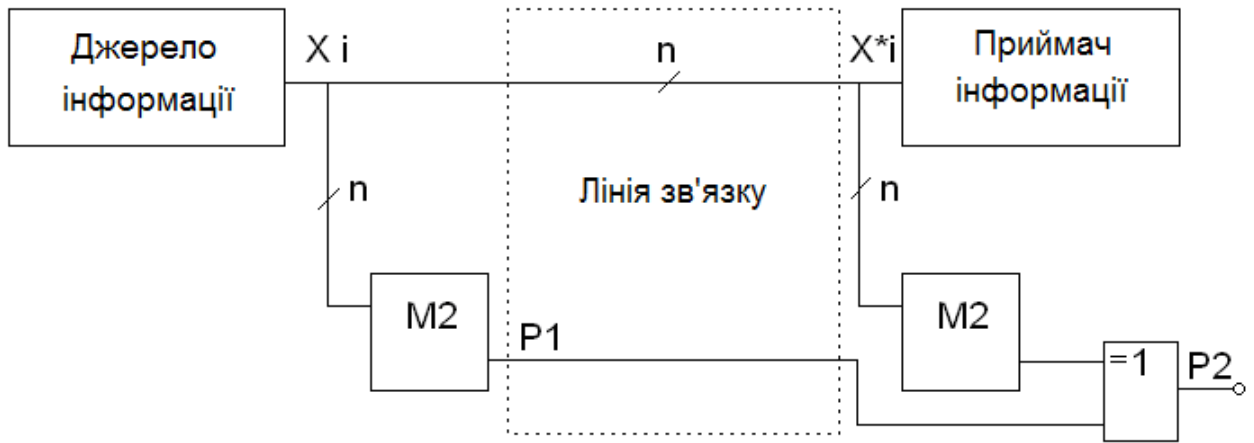


Рисунок 2.11 - Схема контролю парності

Тоді величина $P1 = x_0 \oplus x_1 \oplus \dots \oplus x_{(n-1)}$.

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

2.8 Мажоритарні елементи

Мажоритарний закон це «Закон більшості». Вирішальний елемент зазвичай називають мажоритарним елементом. Мажоритарний елемент — це логічний пристрій з непарним числом входів $m = 2k + 1$ ($k = 1, 2, 3$) і одним виходом. Робота мажоритарного елемента полягає в наступному: на входи елемента поступають двійкові сигнали від непарної кількості ідентичних елементів; вихідний сигнал елемента набуває значення, рівного значенню, яке приймає більшість вхідних сигналів. Найширше використовують мажоритарні елементи, що працюють за законом «2 з 3». У цих елементах значення вихідного сигналу дорівнює значенню двох однакових вхідних сигналів (табл. 2.8). Крім того, відомі мажоритарні елементи, що працюють за законом «3 з 5», «4 з 7» і так далі.

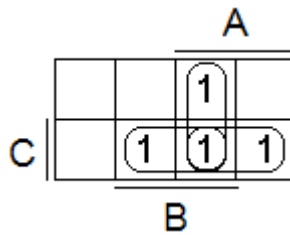
Таблиця 2.8 - Алгоритм функціонування мажоритарного елемента

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Рівняння функціонування:

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

Спростимо вираження за допомогою карти Карно:



$$F = AB + AC + BC$$

Схема мажоритарного елемента, який працює за законом «2 з 3» і побудовано з логічних елементів І та АБО має вигляд, представлений на рисунку 2.12.

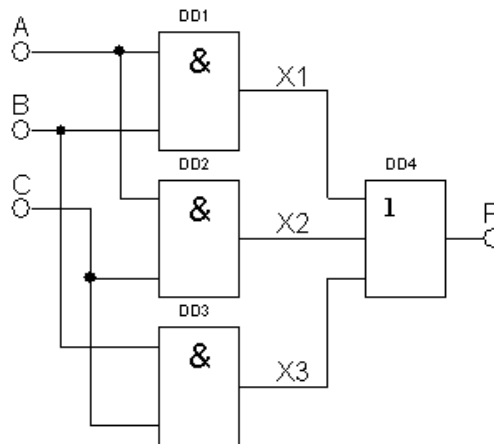


Рисунок 2.12 – Схема мажоритарного елемента

Мажоритарний елемент дозволяє забезпечити режим одночасного штатного функціонування основного і резервних елементів мікропроцесорного пристрою. При цьому відмова основного або резервного елементів не впливає на роботу справних елементів, що залишилися.

2.9 Порогова логічна схема

Пороговою логічною схемою називається схема, в якій певна мінімальна кількість вхідних змінних повинна мати стан 1, аби на виході з'явилася логічна 1.

Наприклад, потрібно розрахувати схему з чотирма вхідними змінними. На виході має бути 1 лише тоді, коли щонайменше на 3 входах присутня 1.

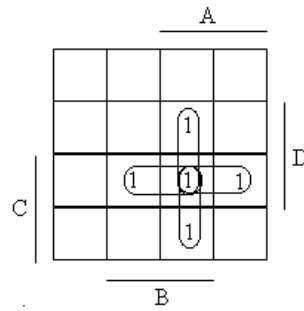
Вхідні змінні мають імена A, B, C, D. Вихідна змінна – Z. Спочатку потрібно визначити таблицю істинності. При чотирьох змінних величинах можливі 16 варіантів (табл. 2.9). Досконала диз'юнктивна нормальна форма (ДДНФ) складається з п'яти повних кон'юнкцій.

Таблиця 2.9 – Таблиця функціонування порогової логічної схеми

| Входи | | | | Вихід |
|-------|---|---|---|-------|
| A | B | C | D | Z |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

$$Z = \overline{A}BCD + A\overline{B}CD + AB\overline{C}D + ABC\overline{D} + ABCD$$

ДДНФ спрощується за допомогою діаграми Карно:



$$Z = BCD + ACD + ABD + ABC$$

У базисі Шеффера:

$$\begin{aligned} Z &= BCD + ACD + ABD + ABC = \overline{\overline{BCD} + \overline{ACD} + \overline{ABD} + \overline{ABC}} = \\ &= \overline{\overline{DCD} \cdot \overline{ACD} \cdot \overline{ABD} \cdot \overline{ABC}} \end{aligned}$$

Порогова логічна схема представлена на рисунку 2.13.

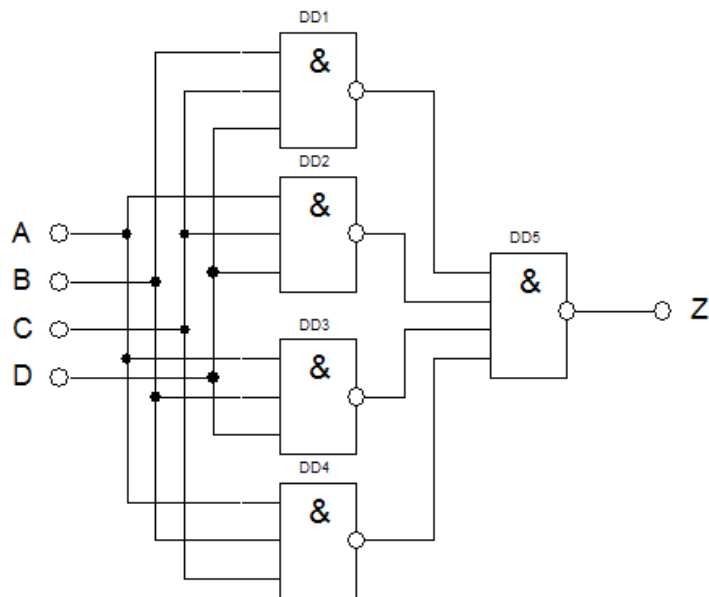


Рисунок 2.13 – Порогова логічна схема

Контрольна робота №1

Варіант 1

1. Сполучення мікроконтролерів з індикаторами.
2. Розробити логічну схему пристрою мажоритарного контролю шести-розрядних слів, які поступають на мажоруючий елемент з трьох ідентичних вузлів. Реалізацію схеми виконати на елементах базису Пірса.

Варіант 2

1. Робота мікроконтролера із зовнішньою пам'яттю даних.
2. Розробити порогову логічну схему з п'ятью вхідними змінними. На виході має бути 1 лише тоді, коли щонайменше на 4 входах присутня 1. Реалізацію схеми виконати на елементах базису Пірса.

Варіант 3

1. Мікроконтролери сімейства x51.
2. Розробити логічну схему пристрою мажоритарного контролю шести-розрядних слів, які поступають на мажоруючий елемент з трьох ідентичних вузлів. Реалізацію схеми виконати на елементах базису Шеффера.

Варіант 4

1. Класифікаційні ознаки МП і МПК.
2. Розробити логічну схему зведення в квадрат двійкових чисел від 0 до 9, з представленням інформації в додатковому коді, з виведенням інформації на семисегментні індикатори. Реалізацію схеми виконати на елементах базису Шеффера.

Варіант 5

1. Однокристальні AVR мікроконтролери.
2. Розробити логічну схему мажоритарного елемента що працює за законом «3 з 5» у базисі Шеффера.

Варіант 6

1. Однокристальні мікроконтролери з CISC архітектурою.

2. Розробити логічну схему пріоритетного шифратора 8×3 , з входом дозволу E1, виходом перенесення E0 і виходом G, що визначає ознаку вхідного інформаційного сигналу. Усі входи і виходи прямі.

Варіант 7

1. Модуль центрального процесора.
2. Розробити логічну схему перетворювача десяткового коду чисел від 0 до 9 в двійково-десятковий, з виведенням інформації на семи сегментні індикатори.

Варіант 8

1. Пристрої введення-виведення.
2. Розробити логічну схему порівняння двох двійкових двохранрядних чисел, з виведенням інформації про більше число на семисегментний індикатор. Реалізацію схеми виконати на елементах базису Пірса.

Варіант 9

1. Інтерфейси введення-виведення.
2. Розробити логічну схему зведення в квадрат десяткових чисел від 0 до 9, з представленням інформації в кодї Грея та виведенням інформації на семисегментні індикатори. Реалізацію схеми виконати на елементах базису Шеффера.

Варіант 10

1. Система переривань.
2. Розробити логічну схему порівняння двох двійкових двохранрядних чисел, з виведенням інформації про рівність чисел на семисегментний індикатор. Реалізацію схеми виконати на елементах базису Шеффера.

Варіант 11

1. Пристрій управління.
2. Розробити логічну схему порівняння двох двійкових двохранрядних чисел, з виведенням інформації про рівність чисел на семисегментний індикатор. Реалізацію схеми виконати на елементах базису Пірса.

Варіант 12

1. Арифметично логічний пристрій.
2. Розробити логічну схему пірамідального дешифратора 3 в 8 з прямими входами А, В, С, інверсним входом \overline{OE} , та інверсними виходами. Представити діаграму функціонування.

Варіант 13

1. Однокристальний 8-розрядний мікроконтролер.
2. Розробити логічну схему мультиплексора 8 в 1 з прямими інформаційними входами, інверсним входом \overline{OE} , та інверсним виходом Q. Представити діаграму функціонування.

Варіант 14

1. Програмна модель МП.
2. Розробити логічну схему шифратора десяткового коду клавішної клавіатури від 0 до 9 в двійковий еквівалент коду натиснутої клавіши та виведенням інформації на світлодіодний індикатор. Реалізацію схеми виконати на елементах базису Шеффера.

Варіант 15

1. Принципи магістральності, модульності та мікропрограмного управління.
2. Розробити логічну схему зведення в квадрат двійкових чисел від 0 до 9, з представленням інформації в додатковому коді, з виведенням інформації на семисегментні індикатори. Реалізацію схеми виконати на елементах базису Пірса.

Варіант 16

1. Однокристальні мікроконтролери з RISC архітектурою.
2. Розробити логічну схему контролю парності, для виявлення одиночних помилок при передачі чотирьохрозрядного коду і біта парності по паралельній лінії зв'язку.

Варіант 17

1. Організація шин.

2. Розробити логічну схему порівняння двох двійкових двохранрядних чисел, з виведенням інформації про менше число на семисегментний індикатор.

Варіант 18

1. Складові частини МПК.
2. Розробити логічну схему зведення в квадрат десяткових чисел від 0 до 9, з представленням інформації в кодї Грея та виведенням інформації на семисегментні індикатори. Реалізацію схеми виконати на елементах базису Пірса.

Варіант 19

1. Інтерфейс сполучення мікроконтролера з клавіатурою.
2. Розробити порогову логічну схему з п'ятью вхідними змінними. На виході має бути 1 лише тоді, коли щонайменше на 3 входах присутня 1. Реалізацію схеми виконати на елементах базису Шеффера.

Варіант 20

1. Однокристальний мікропроцесор.
2. Розробити логічну схему зведення в квадрат десяткових чисел від 0 до 9, з представленням інформації в кодї Грея та виведенням інформації на семисегментні індикатори. Реалізацію схеми виконати на елементах базису Шеффера.

Варіант 21

1. Мікропроцесорний комплект (МПК).
2. Розробити логічну схему дешифратора 3 в 8 з інверсними входами $\bar{A}, \bar{B}, \bar{C}, \bar{O\bar{E}}$, та прямими виходами. Представити діаграму функціонування.

Варіант 22

1. Регістри. Акумулятор.
2. Розробити логічну схему шифратора десяткового коду клавішної клавіатури від 0 до 9 в двійковий еквівалент коду натиснутої клавіши та виведенням інформації на світлодіодний індикатор. Реалізацію схеми виконати на елементах базису Пірса.

Варіант 23

1. Показчик команд. Показчик стека.

Розробити логічну схему зведення в квадрат десяткових чисел від 0 до 9, з представленням інформації в коді Грея та виведенням інформації на семисегментні індикатори. Реалізацію схеми виконати на елементах базису Пірса.

Варіант 24

1. Регістр команд.
2. Розробити логічну схему дешифратора 3 в 8 з інверсними входами $\bar{A}, \bar{B}, \bar{C}$, прямим входом OE та інверсними виходами. Представити діаграму функціонування.

Варіант 25

1. Регістр адреси і реєстри даних.
2. Розробити логічну схему порівняння двох двійкових двохрозрядних чисел, з виведенням інформації про більше число на семисегментний індикатор. Реалізацію схеми виконати на елементах базису Пірса.

3 Послідовні функціональні вузли мікропроцесорної техніки

3.1 Цифрові регістри. Пристрої накопичення інформації

Цифрові регістри - це пристрої, призначені для зберігання і перетворення багаторозрядних двійкових чисел. Елементами регістра, що запам'ятовують, є тригери, число яких дорівнює розрядності чисел, що зберігаються. Окрім тригерів регістри містять також комбінаційні схеми, призначені для введення і виведення чисел, що зберігаються, перетворення їх код, зрушення код на те або інше число розрядів. Інформація в регістрах зберігається, як правило, протягом деякої кількості тактів [2, 5].

Розрізняють паралельні регістри (регістри пам'яті), послідовні регістри (регістри зрушення), паралельно-послідовні регістри (наприклад, введення в паралельному коді, вивід в послідовному і навпаки).

Паралельні регістри - це пристрої, призначені для запису, зберігання і видачі інформації, представленої у вигляді двійкового коду. Для зберігання кожного двійкового розряду в регістрі використовується одне тригерне вічко. Для запам'ятовування багаторозрядних слів необхідне число тригерів об'єднують разом і розглядають як єдиний функціональний вузол-регістр. Структурна схема регістра цього типу представлена на рисунку 3.1.

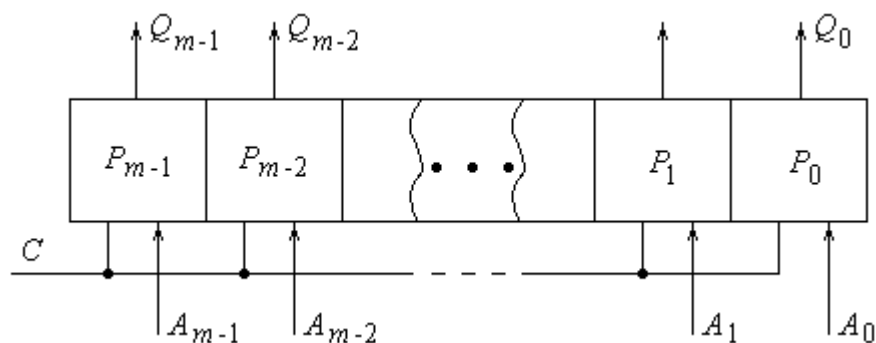


Рисунок 3.1 – Структурная схема параллельного регистра

Запис інформації в паралельні регістри здійснюється паралельним кодом, тобто у всі розряди регістра одночасно. Їх функція зводиться лише до

прийому, зберігання і передачі інформації. У зв'язку з цим паралельні регістри називають регістрами пам'яті. Як розряди регістра пам'яті використовуються D-тригери, що синхронізуються.

3.2 Цифрові лічильники

Лічильником називається пристрій, призначений для підрахунку числа вхідних сигналів і зберігання в певному двійковому коді цього числа [2, 5].

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

Основна характеристика лічильника – модуль рахунку, або ємкість лічильника $K_{\text{ліч}}$. Це кількість вхідних сигналів, що поступили, яка повертає лічильник у вихідний стан.

Процедура розрахунку і побудови лічильника включає наступні операції.

- 1) Визначається необхідна кількість розрядів m . В даному випадку

$$m = \log_2 K_{\text{ліч}}.$$

- 2) Будується таблиця станів лічильника.

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

4) Вибирається тип тригера для побудови лічильника. Таблиці переходів JK-тригера і D-тригера представлені на рисунку 3.2 відповідно до рівнянь функціонування.

Використовуючи словник переходів для кожного входу тригера складаються карти Карно, в клітках яких проставляються сигнали, необхідні для забезпечення переходів тригерів, вказаних в однойменних клітках карт функцій переходів.

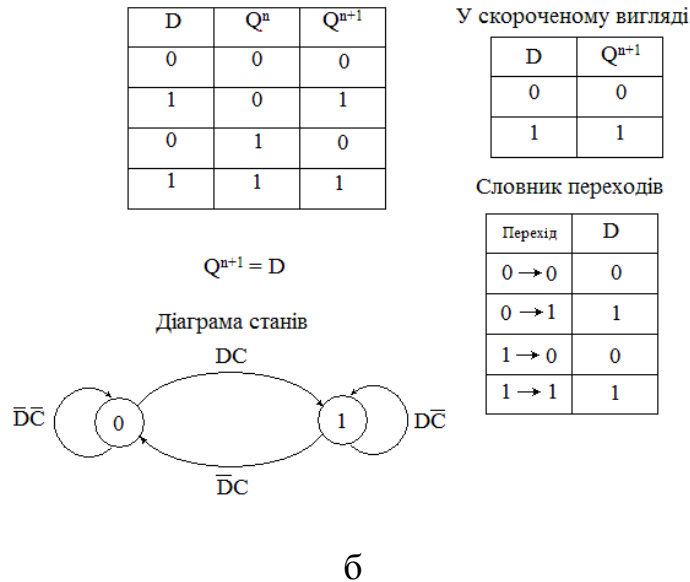
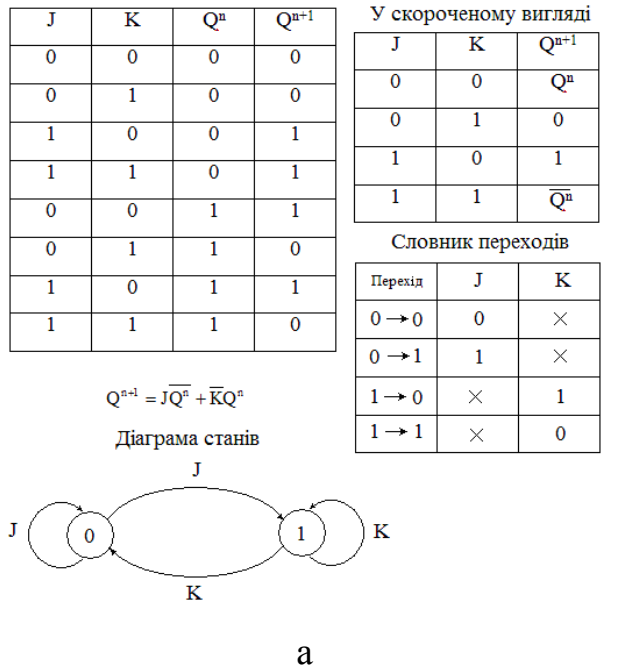


Рисунок 3.2 - Таблиці переходів JK-тригера (а) і D-тригера (б)

5) Проводиться мінімізація логічних функцій входів в картах Карно з метою здобуття їх аналітичних вистав, що показують зв'язки між входами і виходами всіх тригерів, складових лічильник. В процесі мінімізації виробляється до визначення функцій там, де це доцільно, одиницями в клітках (×). В результаті отримуються функції входів тригерів лічильника

б) Будується електрична схема лічильника, що реалізує функції входів.

3.3 Генератори числових послідовностей

Послідовнісні пристрою цього типа називають також розподільниками сигналів, оскільки послідовність двійкових чисел на їх виходах використовується для управління роботою інших цифрових вузлів. Число станів генератора називається довжиною послідовності чисел L_n , яка дорівнює кількості тактів, після якої послідовність чисел на виході генератора повторюється.

Синтез структури генераторів послідовностей на зрухових регістрах, в першу чергу, полягає в знаходженні вигляду функцій входів [2].

- 1) Складається таблиця переходів станів розрядів генератора.
- 2) Складаються карти Карно для функцій переходів тригерів кожного розряду. Зазвичай, в зрухових регістрах використовуються D-тригери.
- 3) Використовуючи словник переходів D-тригера, для кожного входу тригера складаються карти Карно, в клітках яких проставляються сигнали, необхідні для забезпечення переходів тригерів, вказаних в однойменних клітках карт функцій переходів.
- 4) Проводиться мінімізація логічних функцій входів в картах Карно з метою здобуття їх аналітичних виразів. В результаті отримуються функції входів тригерів генератора.
- 5) Складання логічної схеми генератора виконується побудовою комбінаційної схеми, що управляє, реалізовує функцію входів, і її підключенням до входу першого розряду зрухового регістра.

4 Арифметичні логічні пристрої

Арифметико-логічний пристрій (АЛП) функціонує на основі мікропрограмного управління. Кожна машинна операція розділяється на послідовність елементарних дій (передача слів, інверсія слів та ін.), що реалізуються в тактах. Елементарне функціональне обчислення, що виконується в одному машинному такті, називається мікрооперацією. Кожна мікрооперація ініцію-

ється відповідним сигналом, що управляє. Сукупність мікрооперацій, що виконуються в одному такті, називається мікрокомандою [2].

Будь-який цифровий обчислювач, у тому числі і АЛП, може бути представлений композицією операційного автомата (ОА) і управляючого автомата (УА). У операційному автоматі виконуються арифметико-логічні операції. Автомат, що управляє, забезпечує виконання операцій за допомогою послідовності сигналів, що управляють, яку він виробляє залежно від мікропрограми.

Арифметико-логічні пристрої (рис. 3.3) класифікують по наступних ознаках:

- способу обробки даних – паралельні, послідовні, паралельно-послідовні;
- системі числення – двійкові, вісімкові, десяткові, шостнадцятиричні, а також пристрої на основі спеціальних систем (залишкових класів, з штучним порядком ваги, чисел Фібоначчі) та ін.;
- формі представлення чисел – з плаваючою комою, з фіксованою комою, цілі двійкові і десяткові числа;
- часу виконання операцій – синхронні і асинхронні;
- способу виконання мікрооперацій – із закріпленими мікроопераціями, із загальними операціями;
- типу автомата, що управляє – з схемною або програмованою логікою;
- методу побудови – багатофункціональні і блокові.

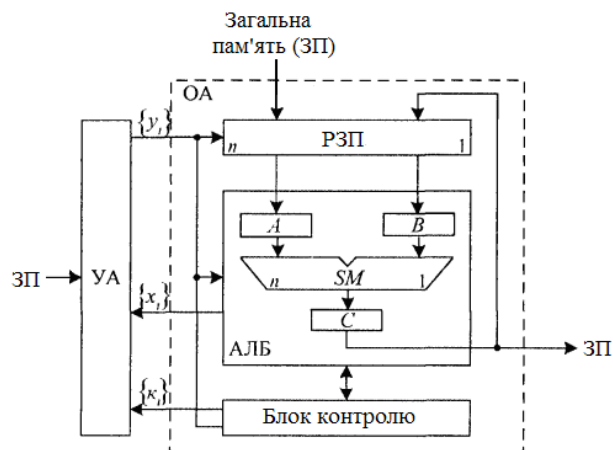


Рисунок 3.3 - Узагальнена структура АЛП

До складу ОА мікропроцесорів входять:

- арифметико-логічний блок (АЛБ);
- набір регістрів загального призначення (РЗП);
- блок контролю.

У АЛБ виділяють комбінаційний суматор SM , вхідні регістри A і B для прийому операндів і вихідний регістр C для запису результату. У АЛБ, є логічні схеми, що виробляють безліч $\{x_i\}$ сигналів логічних умов (ознак результату), наприклад, нульовий або негативний результат і ін. [2].

Регістри загального призначення використовують для прийому і зберігання операндів, проміжних і кінцевих результатів. Блок контролю забезпечує перевірку правильності виконання арифметико-логічних операцій, одночасною реалізацією тієї ж команди, дублюючою апаратурою і порівнянням результатів, або путем виконання дій над спеціальними кодами, отриманими від операндів при складанні по модулю два, три і ін.

При виявленні помилок і збоїв в роботі ОА блок контролю посилає в УА код помилок $\{k_i\}$.

У АЛП поступає код операції від центрального пристрою управління. Використання в АЛП пристроїв управління з схемною логікою прискорює виконання операцій. Вживання УА з програмованою логікою забезпечує гнучкість мікропрограмування, дозволяє змінювати склад мікропрограм при введенні нових команд. У сучасних АЛП можуть об'єднуватися обидва типа УА.

4.1 Однорозрядні суматори

У цифровій обчислювальній техніці використовуються однорозрядні схеми, що підсумовують, з двома і трьома входами, причому перші називаються напівсуматорами а другі – повними однорозрядними суматорами. Напівсуматори можуть використовуватися лише для підсумовування молодших розрядів чисел. Повні однорозрядні суматори мають додатковий третій вхід, на який подається перенесення з попереднього розряду при підсумовуванні багаторозрядних чисел [2].

На рисунку 3.4 а приведена таблиця істинності напівсуматора, на підставі якої складена його структурна формула у вигляді ДДНФ (рис. 3.4 б). Функціональна схема, складена на елементах основного базису відповідно до цієї структурної формули, приведена на рисунку 3.4 в.

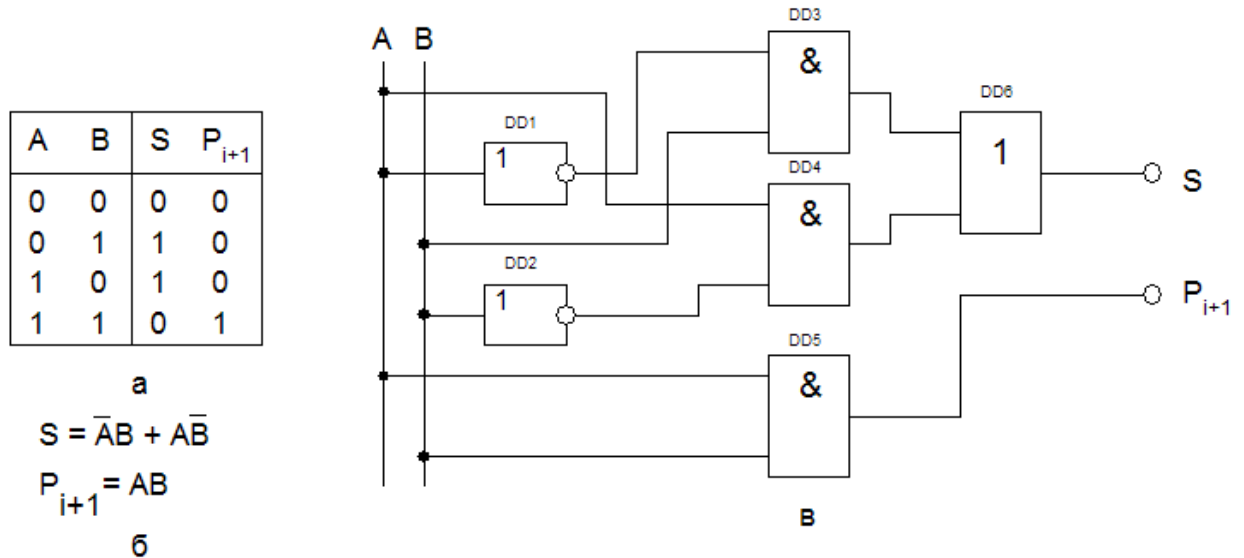


Рисунок 3.4 – Принцип побудови напівсуматора

Рівняння для S (рис. 3.4 б) представляє собою складення по модулю 2 та може бути реалізовано елементом «Виключаюче АБО» (рис. 3.5).

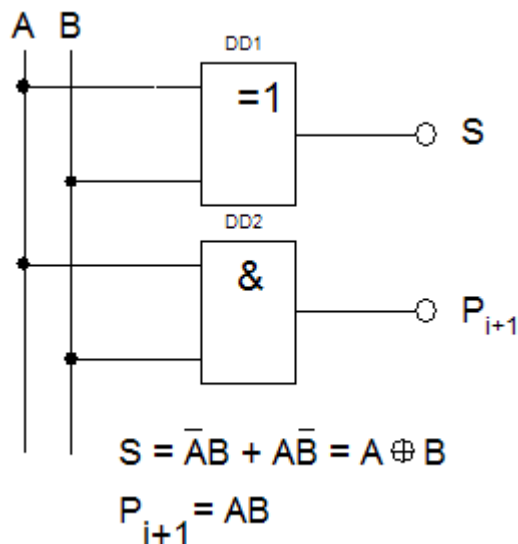


Рисунок 3.5 – Напівсуматор на елементі «Виключаюче АБО»

Якщо згідно таблиці істинності (рис. 3.4 а) скласти структурну формулу у вигляді ДКНФ:

$$S = (A + B)(\bar{A} + \bar{B});$$

$$P_{i+1} = (A + B)(A + \bar{B})(\bar{A} + B).$$

Застосуємо до рівняння для S теорему Де Моргана, а для P_{i+1} закон ідемпотентності:

$$S = (A + B)(\bar{A} + \bar{B}) = (A + B) \cdot \overline{AB};$$

$$\begin{aligned} P_{i+1} &= (A + B)(A + \bar{B})(\bar{A} + B) = \underline{(A + B)} \cdot \underline{(A + \bar{B})} \cdot \underline{(\bar{A} + B)} \cdot \underline{(A + B)} = \\ &= (A + B\bar{B})(B + A\bar{A}) = (A + 0)(B + 0) = AB. \end{aligned}$$

Схема на елементах елементарної логіки представлена на рисунку 3.6.

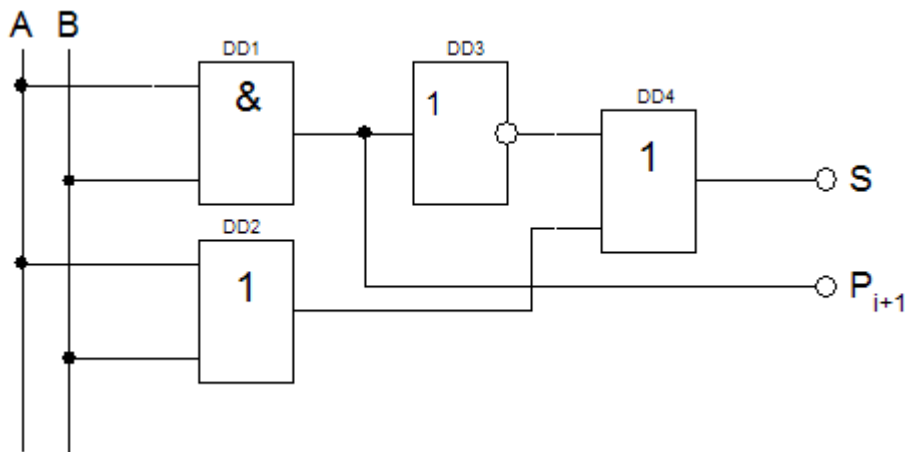


Рисунок 3.6 – Варіант напівсуматора на елементах елементарної логіки

4.2 Комбінаційні суматори арифметичних логічних пристроїв

Методи побудови багаторозрядних суматорів:

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

При послідовному підсумовуванні використовується один суматор, загальний для всіх розрядів (рис. 3.7). Операнди повинні вводитися в суматор через входи a_i і b_i синхронно, починаючи з молодших розрядів [2].

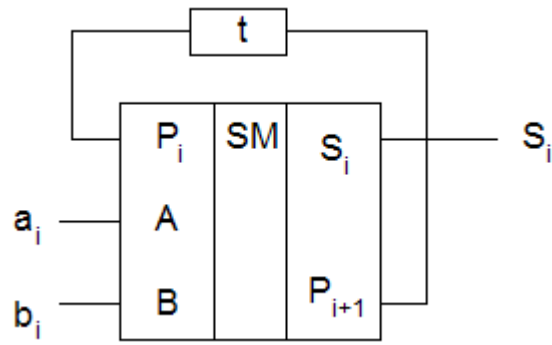


Рисунок 3.7 – Суматор з послідовним підсумовуванням

Коло затримки забезпечує зберігання імпульсу перенесення P_{i+1} на час одного такту, тобто до приходу пари доданків наступного розряду, з якими він буде підсумований. Затримку виконує D-тригер. Результати підсумовування також прочитуються послідовно, починаючи з молодших розрядів. Для зберігання і введення операндів на входи суматора, а також для запису результату підсумовування зазвичай використовуються регістри зсуву [2].

Схема паралельного суматора з послідовним перенесенням приведена на рисунку 3.8.

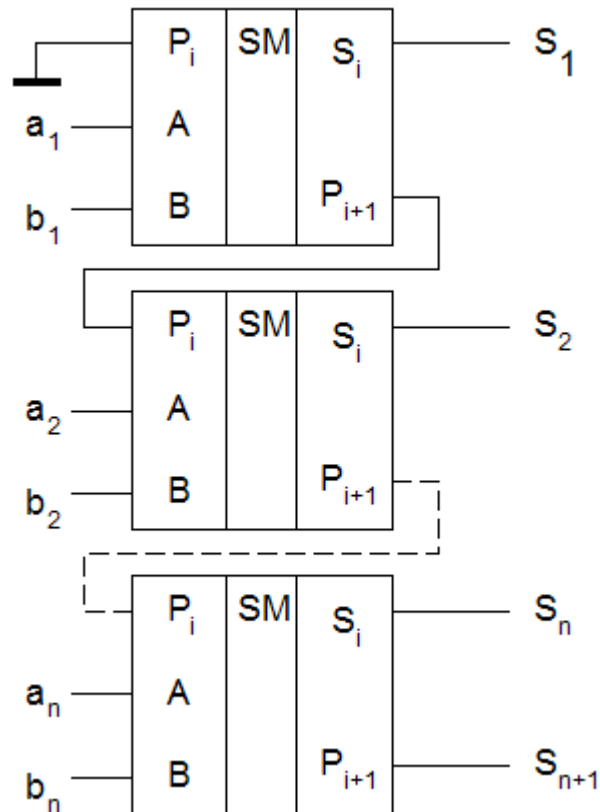


Рисунок 3.8 - Паралельний суматор з послідовним перенесенням

Кількість суматорів дорівнює числу розрядів чисел. Вихід перенесення кожного суматора P_{i+1} з'єднується з входом перенесення P_i наступного більш старшого розряду. На вході перенесення молодшого розряду встановлюється потенціал «0», оскільки сигнал перенесення сюди не поступає. Доданки a_i і b_i підсумовуються у всіх розрядах одночасно, а перенесення P_i поступає із закінченням операції складання в попередньому розряді.

При побудові паралельних суматорів з паралельним перенесенням застосовуються спеціальні вузли – блоки прискореного перенесення [2].

Принцип прискореного перенесення полягає в тому, що для кожного двійкового розряду додатково знаходяться два сигнали:

G - утворення перенесення і H - поширення перенесення.

$$G_i = a_i \cdot b_i$$

$$H_i = a_i + b_i$$

В случае $G_i = 1$, то есть $a_i = b_i = 1$, в данном i -разряде формируется сигнал переноса P_{i+1} в следующий высший разряд независимо от формирования функций суммы в предыдущих разрядах.

Если хотя бы одно из слагаемых a_i или b_i равно «1», то есть $H_i = 1$, то перенос в последующий разряд производится при наличии сигнала переноса из предыдущего разряда.

Если $H_i = H_{i-1} = 1$ и при этом существует сигнал переноса P_i из предыдущего в i -разряд, то перенос производится сразу в $i+2$ разряд.

В общем случае процесс формирования ускоренного переноса описывается следующим уравнением:

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

Блоки ускоренного переноса выпускаются в интегральном исполнении в виде отдельных микросхем или непосредственно со схемой сумматора или арифметико-логического устройства в одной микросхеме.

Контрольна робота №2

Варіант 1

1. Розрахувати і побудувати лічильник з $K_{\text{ліч.}} = 6$, що працює в коді Грея (на JK-триггерах).
2. Розробити логічну схему повного чотирьохрядного паралельного суматора з накопиченням інформації та виведенням інформації на світлодіодні індикатори.

Варіант 2

1. Розрахувати і побудувати лічильник з $K_{\text{ліч.}} = 8$, що працює в додатковому коді (на JK-триггерах).
2. Синтезувати структуру генератора послідовності 3-5-2-4-6-7-1-0 на зрухових регістрах.

Варіант 3

1. Розрахувати і побудувати лічильник, що працює в коді Айкена (2-4-2-1) (на JK-триггерах).
2. Синтезувати структуру генератора послідовності 0-3-5-4-1-2-6-7 на зрухових регістрах.

Варіант 4

1. Розрахувати і побудувати лічильник з $K_{\text{ліч.}} = 10$, що працює в коді з надлишком 3 (на JK-триггерах).
2. Розробити логічну схему паралельного чотирьохрозрядного суматора – віднімача, з виведенням інформації на світлодіодні індикатори.

Варіант 5

1. Розрахувати і побудувати десятковий лічильник (на JK-триггерах).
2. Синтезувати структуру генератора послідовності 7-6-5-4-3-2-1-0 на зрухових регістрах.

Варіант 6

1. Розрахувати і побудувати лічильник з $K_{\text{ліч.}} = 8$, що працює в коді Грея (на D-триггерах).

2. Синтезувати структуру генератора послідовності 0-1-2-3-4-5-6-7 на зрухових регістрах.

Варіант 7

1. Розрахувати і побудувати лічильник з $K_{\text{ліч.}} = 10$, що працює в додатковому коді (на D -триггерах).

2. Розробити логічну схему повного чотирьохрозрядного паралельного суматора – віднімача, з накопиченням інформації та виведенням інформації на світлодіодні індикатори.

Варіант 8

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

2. Синтезувати структуру генератора послідовності 7-5-3-1-0-2-4-6 на зрухових регістрах.

Варіант 9

1. Розрахувати і побудувати лічильник з $K_{\text{ліч.}} = 10$, що працює в коді з надлишком 3 (на D -триггерах).

2. Синтезувати структуру генератора послідовності 3-0-2-6-7-5-1-4 на зрухових регістрах.

Варіант 10

1. Розрахувати і побудувати десятковий лічильник (на D -триггерах).

2. Розробити логічну схему суматора з послідовним підсумовуванням та виведенням інформації на світлодіодні індикатори.

Варіант 11

1. Розрахувати і побудувати лічильник з $K_{\text{ліч.}} = 8$, що працює в коді з надлишком 4 (на JK-триггерах).

2. Синтезувати структуру генератора послідовності 7-2-3-1-5-4-6-0 на зрухових регістрах.

Варіант 12

1. Розрахувати і побудувати лічильник з $K_{\text{ліч.}} = 8$, що працює в коді з надлишком 4 (на D-триггерах).

2. Синтезувати структуру генератора послідовності 2-5-1-0-7-4-6-3 на зрухових регістрах.

Варіант 13

1. Розрахувати і побудувати 8 – розрядний паралельний регістр, інформація на який поступає у вигляді однофазних сигналів.

2. Синтезувати структуру генератора послідовності 0-2-4-6-7-5-3-1 на зрухових регістрах.

Варіант 14

1. Розрахувати і побудувати 8 – розрядний паралельний регістр, інформація на який поступає у вигляді парафазних сигналів.

3. Синтезувати структуру генератора послідовності 5-1-2-7-6-3-0-4 на зрухових регістрах.

Варіант 15

1. Розрахувати і побудувати чотирьох розрядний реверсивний регістр із зрушенням інформації вправо.

2. Розробити схему блоку порозрядних логічних операцій між двома операндами: I – НІ, АБО, АБО – НІ, «Виключаюче АБО», І, «Рівнозначність».

Варіант 16

1. Розрахувати і побудувати чотирьох розрядний реверсивний регістр із зрушенням інформації вліво.

2. Розробити схему блоку порозрядних логічних операцій між двома операндами: І, АБО, «Рівнозначність», «Виключаюче АБО», І – НІ, АБО – НІ.

Варіант 17

1. Розрахувати і побудувати лічильник з $K_{\text{ліч.}} = 10$, що працює в коді Джонсона (на D тригерах).

2. Синтезувати структуру генератора послідовності 3-5-2-6-7-8-0-9 на зрухових регістрах.

Варіант 18

1. Розрахувати і побудувати лічильник з $K_{\text{ліч.}} = 10$, що працює в коді Джонсона (на JK тригерах).

2. Синтезувати структуру генератора послідовності 4-0-5-2-3-6-7-9-8 на зрухових регістрах.

Варіант 19

1. Розрахувати і побудувати з $K_{\text{ліч.}} = 10$, що працює в коді «Два із п'яти» (на D -тригерах).

2. Синтезувати структуру генератора послідовності 0-3-2-5-7-0-2-6 на зрухових регістрах.

Варіант 20

1. Розрахувати і побудувати з $K_{\text{ліч.}} = 10$, що працює в коді «Два із п'яти» (на JK -тригерах).

2. Синтезувати структуру генератора послідовності 6-4-5-2-4-5-7-3 на зрухових регістрах.

Варіант 21

1. Розрахувати і побудувати лічильник з $K_{\text{ліч.}} = 6$, що працює в коді Грея (на D -тригерах).

2. Розробити схему блоку порозрядних логічних операцій між двома операндами: «Рівнозначність», АБО, АБО – НІ, «Виключаюче АБО», І, І – НІ.

Варіант 22

1. Розрахувати і побудувати лічильник з $K_{\text{ліч.}} = 8$, що працює в коді Грея (на JK -тригерах).

2. Розробити схему блоку порозрядних логічних операцій між двома операндами: НІ, І, «Виключаюче АБО», АБО, І – НІ, АБО – НІ.

Варіант 23

1. Розрахувати і побудувати лічильник з $K_{\text{ліч.}} = 10$, що працює в додатковому коді (на JK -тригерах).

2. Синтезувати структуру генератора послідовності 2-3-7-6-9-2-4-5 на зрухових регістрах.

Варіант 24

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

2. Розробити логічну схему повного чотирьохрозрядного послідовного суматора, з виведенням інформації на світлодіодні індикатори.

Варіант 25

1. Розрахувати і побудувати лічильник з $K_{\text{ліч.}} = 8$, що працює в коді з надміром 3 (на D -тригерах).

2. Розробити логічну схему суматора з паралельним підсумовуванням та послідовним перенесенням, з виведенням інформації на світлодіодні індикатори.

Список рекомендованої літератури

1. Основи схемотехніки електронних систем: підручник / [Бойко В.І., Гуржій А.М., Жуйков В. Я. та ін.]. – К.: Вища шк., 2004. – 527 с.: іл.

2. Бабич Н.П. Компьютерная схемотехника / Н.П. Бабич, И.А. Жуков. – К: «МК-Пресс», 2004. – 576 с.

3. Алексенко А.Г. Микросхемотехника / А.Г. Алексенко, И.И. Шагурин. - М.: Радио и связь, 1990. - 496 с.

4. Прянишников В.А. Электроника курс лекций / Прянишников В.А. – Санкт Петербург.: «КОРОНА принт». – 1998. – 401 с.

5. Бойт К. Цифровая схемотехника / Клаус Бойт. – Москва: Техносфера, 2007. – 472 с.

Начально-методичне видання
(українською мовою)

Верьовкін Леонід Леонідович

ФУНКЦІОНАЛЬНІ ВУЗЛИ МІКРОПРОЦЕСОРНИХ СИСТЕМ

Методичні рекомендації до практичних занять
для здобувачів вищої освіти першого бакалаврського рівня
за спеціальністю 153 «Мікро- та наносистемна техніка» освітньо-професійної
програми «Мікро- та наносистемна техніка»

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

Підп. до друку 23.06.2021. Формат 60×90×16.
Папір офсетний. Друк цифровий. Гарнітура Times.
Умовн. друк. арк. 1,3. Тираж 6 прим.. Зам №
Запорізький національний університет
69600, м. Запоріжжя, МСП – 41
вул. Жуковського, 66.
Свідотство про внесення суб'єкта видавничої справи
до Державного реєстру видавців, виготівників
і розповсюджувачів видавничої продукції
ДК № 5229 від 11.10.2016.