

6. ДЕШИФРАТОРИ

6.1. Загальна характеристика дешифраторів

Дешифратором називається функціональний вузол комп'ютера, призначений для перетворення кожної комбінації вхідного двійкового коду в керуючий сигнал лише на одному із своїх виходів. У загальному випадку дешифратор має n однофазних входів (іноді $2n$ парафазних) і $m=2^n$ виходів, де n – розрядність (довжина) коду, який дешифрується. Дешифратор з максимально можливим числом виходів $m=2^n$ називається повним. Функціонування повного дешифратора описується системою логічних виразів вигляду:

$$F_0 = \bar{X}_n \bar{X}_{n-1} \dots \bar{X}_2 \bar{X}_1;$$

$$F_1 = \bar{X}_n \bar{X}_{n-1} \dots \bar{X}_2 X_1;$$

$$\dots\dots\dots$$

$$F_{m-1} = X_n X_{n-1} \dots X_2 X_1,$$

де X_1, \dots, X_n – вхідні двійкові змінні; F_0, F_1, \dots, F_{m-1} – вихідні логічні функції, що являють собою мінтерми (конституенти 1) n змінних.

Індекс функції F_i визначає номер обраного виходу і відповідає десятковому еквіваленту вхідного коду. Вихід, на якому з'являється керуючий сигнал, називається активним. Якщо значення сигналу на активному виході відображається лог.1, то на решті пасивних виходів встановлюється лог.0. Двійковий код, який вміщує завжди тільки одну одиницю, а інші – нулі, називається унітарним. Тому дешифратор є перетворювачем вхідного позиційного коду в унітарний вихідний код.

У дешифраторах в інтегральному виконанні стан активного виходу часто відображається значенням лог.0, а на інших пасивних виходах встановлюється лог.1. Функціонування повного дешифратора з інверсними виходами представляється системою виду:

$$L_0 = X_n \vee X_{n-1} \vee \dots \vee X_2 \vee X_1;$$

$$L_1 = X_n \vee X_{n-1} \vee \dots \vee X_2 \vee \bar{X}_1;$$

$$\dots\dots\dots$$

$$L_{m-1} = \bar{X}_n \vee \bar{X}_{n-1} \vee \dots \vee \bar{X}_2 \vee \bar{X}_1,$$

де L_0, L_1, \dots, L_{m-1} – вихідні логічні функції, що є макстермами (конституенти 0) n змінних.

Індекс функції L_i визначає номер вибраного виходу і відповідає десятковому еквіваленту вхідного коду.

Між двома видами вихідних функцій існує простий зв'язок: $F_i = \bar{L}_i$.

Дешифратори класифікують за такими ознаками:

- способом структурної організації – одноступеневі (лінійні) і багатоступеневі, в тому числі пірамідальні та прямокутні (матричні);

- форматом вхідного коду – двійкові, двійково-десяткові;
- розрядністю коду, який дешифрується – 2, 3, ..., n ;
- формою подачі вхідного коду – з однофазними і парафазними входами;
- кількістю виходів – повні й неповні дешифратори;
- видом вхідних стробуючих сигналів – в прямому або інверсному значеннях;
- типом використовуваних логічних елементів – І, НЕ, ЧИ, НЕ І, НЕ ЧИ і т.д.

До основних характеристик дешифратора відносять: число ступенів (каскадів) дешифрації, кількість використаних логічних елементів або мікросхем, загальне число входів логічних елементів, час дешифрації і споживану потужність.

Умовні графічні позначення дешифраторів на електричних схемах показані на рис.6.1.

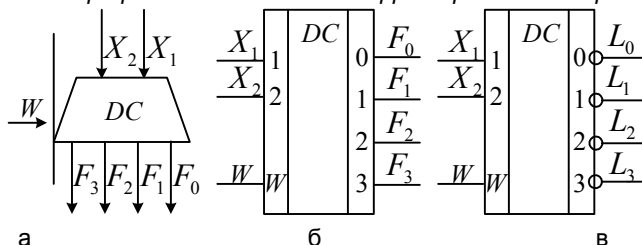


Рис. 6.1. Умовні графічні позначення дешифратора: а – на функціональних схемах; б, в – на принципіальних схемах

Логічна функція дешифратора позначається буквами DC (de-coder). Мітки лівого додаткового поля в умовному позначенні відображають десяткові ваги вхідних змінних, а мітки правого додаткового поля відповідають десятковим еквівалентам вхідних комбінацій двійкових змінних. У схему дешифраторів вбудовуються один або два стробуючих (дозволяючих) входи, наприклад, W (рис.6.1, б). За допомогою сигналу на вході W визначається момент спрацювання дешифратора; крім того, вхід W використовується для нарощування розрядності вхідного коду. На практиці повний дешифратор на n входів і m виходів для стислості називають дешифратором "з n в m " або " $n \rightarrow m$ ". Наприклад, дешифратор "з 3 у 8" – активізується одна з восьми вихідних ліній.

В комп'ютерах дешифратори використовують для виконання таких операцій:

- дешифрації коду операції, записаного в регістр команд процесора, що забезпечує вибір потрібної мікропрограми;
- перетворення коду адреси операнда в команді в керуючі сигнали вибору заданої комірки пам'яті в процесі записування або читання інформації;
- забезпечення візуалізації на зовнішніх пристроях;
- реалізації логічних операцій та побудови мультиплексорів і демультиплексорів.

Використання дешифраторів для дешифрації коду операції і адреси операнда, розташованих в регістрі команд процесора, показано на рис.6.2. Дешифрація коду операції в пристрої керування (ПК) визначає тип машинної команди. Дешифрація адреси операнда в оперативній пам'яті (ОП) забезпечує доступ до вказаної комірки пам'яті для записування або зчитування даних.

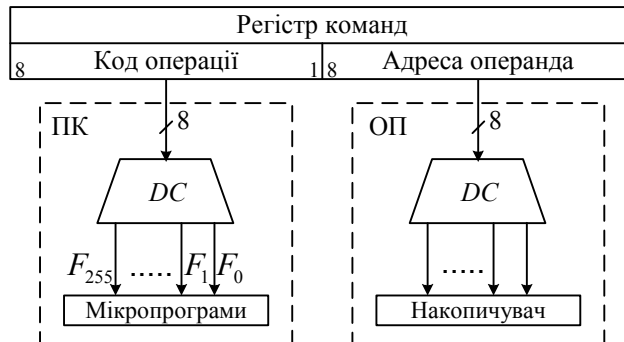


Рис. 6.2. Ілюстрація використання дешифраторів

6.2. Лінійні дешифратори на два входи і чотири виходи

У лінійному дешифраторі "з п в m" кожна вихідна функція F_i реалізується повністю окремим п-вхідним логічним елементом при використанні парафазного вхідного коду. Логіка роботи повних дешифраторів на два входи X_1, X_2 і чотири прямих виходи F_0, F_1, F_2, F_3 і чотири інверсних виходи L_0, L_1, L_2, L_3 наведена в табл.4.1 і 4.2 відповідно.

Таблиця 6.1

X_2	X_1	F_0	F_1	F_2	F_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Таблиця 6.2

X_2	X_1	L_0	L_1	L_2	L_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

За даними табл.6.1 отримують систему логічних функцій в ДДНФ:

$$F_0 = \overline{X_2} \overline{X_1}; F_1 = \overline{X_2} X_1; F_2 = X_2 \overline{X_1}; F_3 = X_2 X_1. \quad (6.1)$$

Для лінійного дешифратора зі стробуючим входом W система рівнянь (4.1) набуває вигляду:

$$F_0 = \overline{X_2} \overline{X_1} W; F_1 = \overline{X_2} X_1 W; F_2 = X_2 \overline{X_1} W; F_3 = X_2 X_1 W. \quad (6.2)$$

Схеми лінійних дешифраторів на основі рівнянь (6.1) і (6.2) показані на рис.6.3.

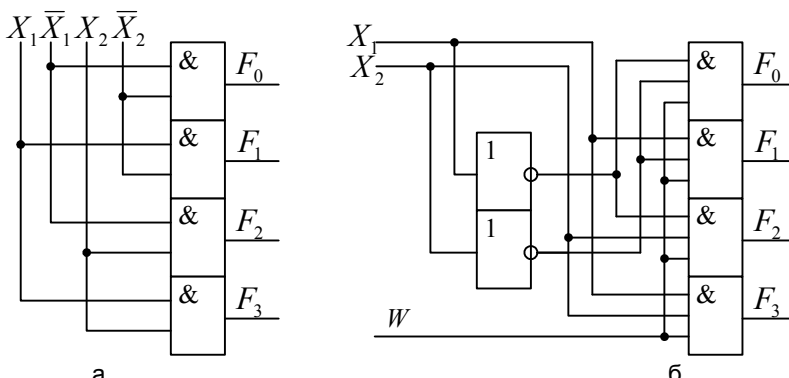


Рис. 6.3. Схеми лінійних дешифраторів на елементах І: а – з парафазними входами; б – з однофазними входами і стробуванням

У схемі, зображеній на рис. 6.3, б використовується однофазний вхідний код, оскільки інверсії змінних утворюються елементами НЕ. Якщо сигнал на стробуючому вході $W=0$, то робота дешифратора блокується – на всіх виходах установлюються логічні нулі незалежно від значень вхідних змінних. При $W=1$ дешифратор функціонує згідно з табл. 6.1.

За даними табл.6.2 записується система логічних функцій в ДКНФ:

$$L_0 = X_2 \vee X_1; L_1 = X_2 \vee \overline{X_1}; L_2 = \overline{X_2} \vee X_1; L_3 = \overline{X_2} \vee \overline{X_1}. \quad (6.3)$$

Схема лінійного дешифратора з парафазним вхідним кодом та інверсними виходами, побудована згідно з рівнянням (6.3) на елементах ЧИ, показана на рис. 6.4, а.

Для лінійного дешифратора із стробуючим W входом система керування (6.3) набуває вигляду:

$$\begin{aligned} L_0 &= X_2 \vee X_1 \vee W; & L_1 &= X_2 \vee \overline{X_1} \vee W; \\ L_2 &= \overline{X_2} \vee X_1 \vee W; & L_3 &= \overline{X_2} \vee \overline{X_1} \vee W. \end{aligned} \quad (6.4)$$

Схема лінійного дешифратора на основі рівнянь (6.4) показана на рис. 6.4, б.

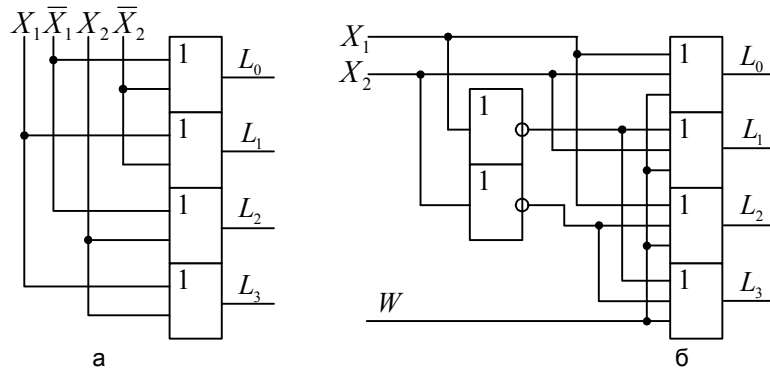


Рис. 6.4. Схема лінійних дешифраторів на елементах ЧИ: а – з парафазними входами; б – з однофазними входами і стробуванням

Тут використовується однофазний вхідний код, оскільки інверсії змінних утворюються елементами НЕ. Якщо сигнал на стробуючому вході $W=1$, то робота дешифратора блокується – на всіх виходах встановлюються лог. 1 незалежно від значень вхідних змінних. При $W=0$ дешифратор функціонує згідно з табл. 6.2.

6.3. Пірамідальні дешифратори

У пірамідальному дешифраторі число ступенів на одиницю менше розрядності вхідного коду, тобто $K=n-1$. В усіх ступенях використовуються тільки двохходові логічні елементи. На першому ступені використовуються лінійні дешифратори на два входи і чотири виходи. Число логічних елементів у кожному ступені дорівнює $M_i=2^{i+1}$, де $i=1, 2, \dots, k$. Це означає, що кожен подальший ступінь має в два рази більше елементів, ніж попередній. Вихід елемента i -го ступеня підключається до входів тільки двох елементів $(i+1)$ -го ступеня.

Пірамідальна структура для реалізації повного дешифратора "з 3 в 8" описується системою мінтермів виду:

$$F_0 = \overline{X_3} \overline{X_2} \overline{X_1}; \quad F_1 = \overline{X_3} \overline{X_2} X_1; \quad \dots \quad F_7 = X_3 X_2 X_1.$$

Схема пірамідального дешифратора з парафазним вхідним кодом на три входи і вісім виходів показана на рис. 6.5.

На першому ступені дешифруються змінні X_2 і X_1 , на другому ступені добавляється розряд X_3 . При більшому числі розрядів дешифрованого коду, наприклад, $n > 10$, дешифратор в $n/4$ економічніше лінійного.

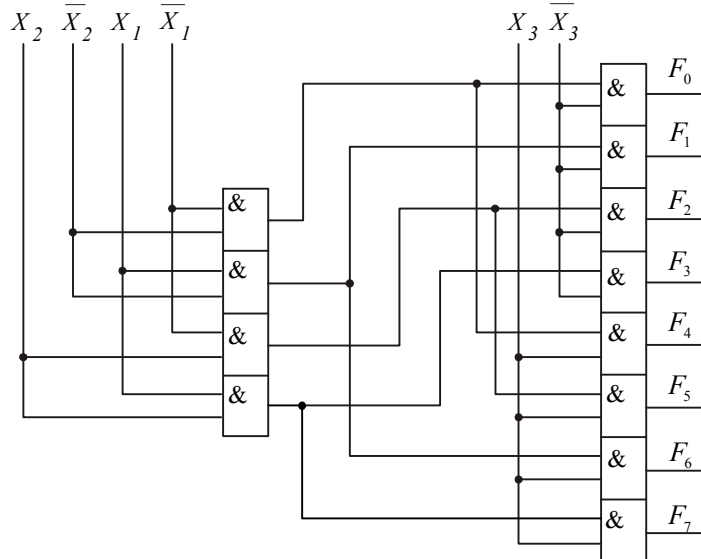


Рис. 6.5. Схема пірамідального дешифратора на три входи і вісім виходів

Основним недоліком пірамідального дешифратора є велике число ступенів, що суттєво збільшує час дешифрації коду.

6.4. Прямокутні дешифратори

Прямокутний дешифратор будується за двоступеневою схемою. При цьому вхідний код розбивається на дві групи по $n/2$ розрядів при парному n ; при непарній розрядності групи вміщують нерівне число змінних. Дві групи змінних декодуються на першому ступені двома повними лінійними (можливо і пірамідальними) дешифраторами, а на другому ступені формуються вихідні функції.

Умовно вважають, що один з дешифраторів першого ступеня формує адреси рядків матриці, а другий – адреси стовпчиків матриці. На перетині ліній рядків і стовпчиків підключається $m=2^n$ двовходових схем збігу, які утворюють другий, вихідний ступінь дешифратора. При парному n матриця вентилів квадратна, при непарному n – прямокутна. Тому такі дешифратори називаються матричними або прямокутними.

Запишемо систему вихідних функцій повного дешифратора "з 4 в 16" у вигляді таких скорочених значень:

$$\begin{aligned} F_0 &= a_0 b_0; & F_4 &= a_1 b_0; & F_8 &= a_2 b_0; & F_{12} &= a_3 b_0; \\ F_1 &= a_0 b_1; & F_5 &= a_1 b_1; & F_9 &= a_2 b_1; & F_{13} &= a_3 b_1; \\ F_2 &= a_0 b_2; & F_6 &= a_1 b_2; & F_{10} &= a_2 b_2; & F_{14} &= a_3 b_2; \\ F_3 &= a_0 b_3; & F_7 &= a_1 b_3; & F_{11} &= a_2 b_3; & F_{15} &= a_3 b_3. \end{aligned} \quad (6.5)$$

де введені дворозрядні функції a_i і b_i , які реалізуються дешифраторами рядків і стовпчиків відповідно:

$$\begin{aligned} b_0 &= \bar{X}_2 \bar{X}_1; & b_1 &= \bar{X}_2 X_1; & b_2 &= X_2 \bar{X}_1; & b_3 &= X_2 X_1; \\ a_0 &= \bar{X}_4 \bar{X}_3; & a_1 &= \bar{X}_4 X_3; & a_2 &= X_4 \bar{X}_3; & a_3 &= X_4 X_3. \end{aligned} \quad (6.6)$$

Схема прямокутного дешифратора на основі рівнянь (6.5) і (6.6) показана на рис.6.6.

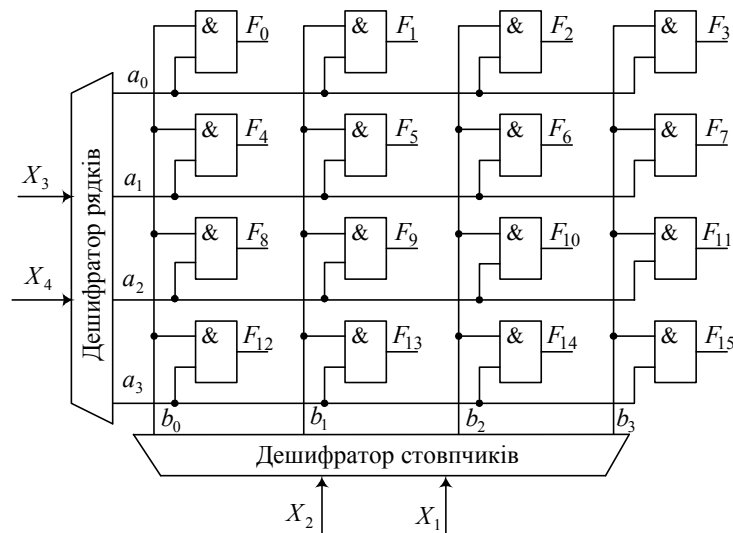


Рис. 6.6. Схема прямокутного дешифратора

При великому числі розрядів прямокутний дешифратор майже у $n/2$ рази економичніший лінійного і у два рази – пірамідального.

6.5. Багаступеневі дешифратори. Каскадування дешифраторів

Принцип побудови багаступеневих дешифраторів полягає у послідовному розбитті вхідного багаторозрядного коду до отримання у кожній групі двох - трьох розрядів. Як приклад на рис. 4.7 показано розбиття коду, який дешифрується для $n=10$ і $n=13$. Після цього багаступенева схема дешифратора зображується у вигляді з'єднання ряду лінійних схем.

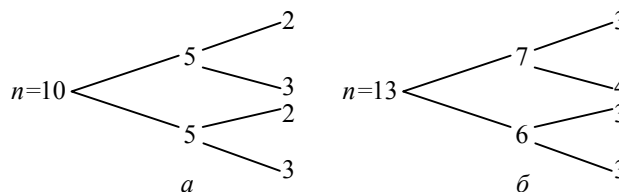
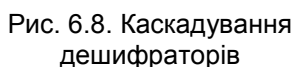


Рис. 6.7. Розбиття вхідного коду, який дешифрується на групи: а – при $n=10$; б – при $n=13$

Під каскадуванням (нарощуванням) розуміють спосіб з'єднання дешифраторів у вигляді мікросхем середнього ступеня інтеграції для одержання більшої розрядності вхідного коду. З'єднання двох трирозрядних дешифраторів типу К555ІДЗ для декодування чотирирозрядного коду показано на рис. 4.8.



a

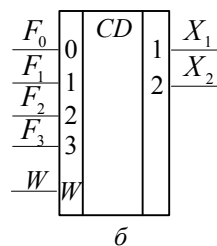


Рис. 7.1. Умовні графічні позначення шифратора: а – на функціональних схемах; б – на принципових схемах

У цифрових пристроях шифратори використовуються для таких операцій: перетворення унітарного вхідного коду у вихідний двійковий позиційний код; введення десяткових даних з клавіатури; показання старшої одиниці в слові; передачі інформації між різними пристроями при

обмеженому числі ліній зв'язку.

7.2. Пріоритетний шифратор клавіатури

Одне з основних застосувань шифратора – введення даних з клавіатури, наприклад, десяткових цифр. Натискання клавіші з десятковою цифрою 0, 1, ..., 9 мають приводити до передачі в цифровий пристрій двійково-десятьового коду цієї цифри. Для цього використовується неповний шифратор “з 10 в 4”.

Шифратори, які при одночасному натисканні декількох клавіш виробляють код тільки старшої цифри, називаються пріоритетними. Пріоритетні шифратори, які призначені для пошуку старшої (лівої) одиниці в слові та формування на виході двійкового номера шуканого розряду, називаються показчиками старшої одиниці. Їх застосовують у пристроях нормалізації чисел з плаваючою крапкою, в системах з пріоритетним обслуговуванням запитів на переривання роботи комп'ютера.

Логіка роботи пріоритетного шифратора на вісім входів наведена в табл.7.1, де прийняті такі позначення: $\overline{F_0}, \overline{F_1}, ..., \overline{F_7}$ – вхідні інверсні сигнали, записані в порядку зростання пріоритету: $\overline{F_0}$ – найнижчий, $\overline{F_7}$ – найвищий; $\overline{X_3}, \overline{X_2}, \overline{X_1}$ – вихідний інверсний позиційний код; \overline{W} – сигнал стробування; \overline{P} – функція, яка вказує на надходження вхідного сигналу; \overline{V} – функція, яка вказує на відсутність вхідних сигналів.

Таблиця 7.1

[illegible]

У табл. 7.1 значення вхідних змінних праворуч від діагоналі, утвореної цифрами 1, не повинні визначати вихідний код (вони позначені хрестиком). Це пояснюється тим, що сигнал з більшим пріоритетом блокує запити з меншими пріоритетами.

Із табл. 7.1 отримуємо вирази для вихідного коду шифратора $\overline{X}_3, \overline{X}_2, \overline{X}_1$ і функцій \overline{V} та \overline{P} , які відповідно визначають відсутність інформаційних сигналів на всіх виходах та наявність сигналу хоч би на одному вході. Для спрощення виразів використовуємо тотожність $F_i \vee \overline{F_i} F_k = F_i \vee F_k$ та закони де Моргана:

$$\overline{X}_3 = \overline{W} \vee W Y_1;$$

$$\overline{X}_2 = \overline{W} \vee W Y_1 \overline{F}_3 \overline{F}_2 \vee W \overline{F}_7 \overline{F}_6 F_5 \vee W \overline{F}_7 \overline{F}_6 F_4;$$

$$\overline{X}_1 = \overline{W} \vee W Y_1 \overline{F}_3 F_2 \vee W Y_1 \overline{F}_3 \overline{F}_1 \vee W \overline{F}_7 \overline{F}_6 \vee W F_7 F_5 F_4;$$

$$\overline{P} = \overline{W} \vee W Y_1 Y_2; \quad \overline{V} = W \vee \overline{Y_1 Y_2} W \vee \overline{Y_1} \vee \overline{Y_2};$$

$$Y_1 = \overline{F}_7 \overline{F}_6 \overline{F}_5 \overline{F}_4; \quad Y_2 = \overline{F}_3 \overline{F}_2 \overline{F}_1 \overline{F}_0.$$

На основі цих виразів побудована (рис. 7.2) схема пріоритетного шифратора "8 → 3".

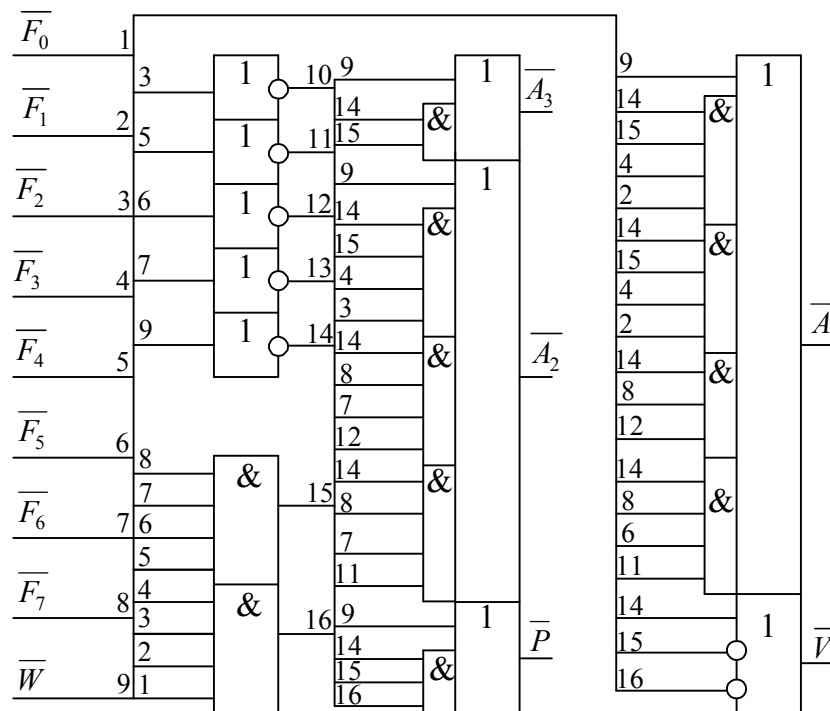


Рис. 7.2. Схема пріоритетного шифратора "8 → 3"

При $\overline{W} = 1$ робота схеми блокується і незалежно від сигналів на входах маємо на інверсних виходах: $\overline{X}_3 \overline{X}_2 \overline{X}_1 = 111$, $\overline{P} = 1$, $\overline{V} = 1$. Якщо, наприклад, $\overline{F}_6 = 0$ і $\overline{F}_2 = 0$, то схема формує на виходах код номера входу із старшим пріоритетом: $\overline{X}_3 \overline{X}_2 \overline{X}_1 = 001$ або в прямому коді $X_3 X_2 X_1 = 110_2 = 6_{10}$. Активний стан виходу відображається значеннями функцій $\overline{P} = 0$ і $\overline{V} = 1$, які передаються в процесор, а також використовуються при каскадуванні шифраторів. Схема, зображена на рис. 7.2, є аналогом шифратора К555ІВ1.

7.3. Каскадування шифраторів

Каскадування шифраторів використовується для збільшення розрядності вхідного слова. Схема каскадування двох восьмивходових шифраторів К555ІВ1 для пріоритетного обслуговування 16-розрядного слова $\bar{F}_{15} - \bar{F}_0$ показана на рис. 7.3.

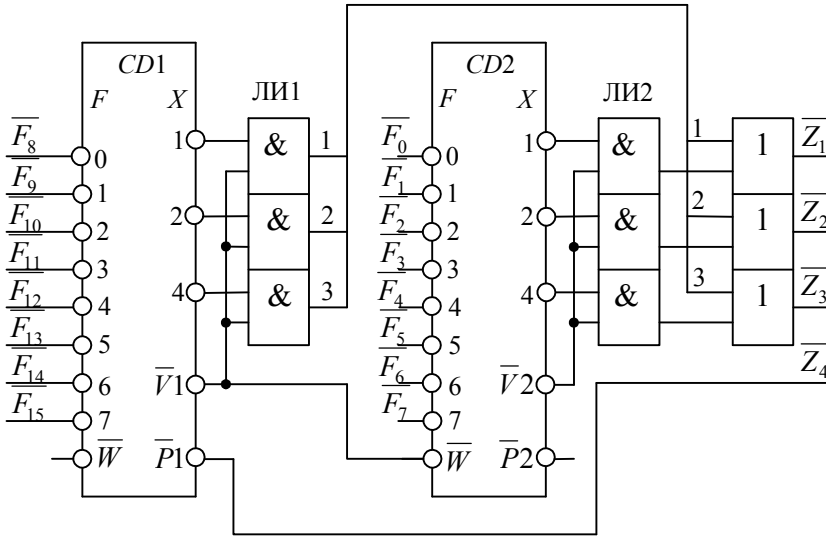


Рис. 7.3. Каскадування двох восьмивходових пріоритетних шифраторів

Розряди $\bar{F}_{15} - \bar{F}_8$ старшого байта вхідного слова поступають на перший шифратор CD1, а розряди $\bar{F}_7 - \bar{F}_0$ молодшого байта подаються на другий шифратор CD2. Вхід \bar{F}_{15} має найвищий пріоритет, а \bar{F}_0 – найнижчий. Інформаційні виходи обох шифраторів об'єднуються за допомогою логічних елементів ЧИ, утворюючи трирозрядний інверсний код $\bar{Z}_3\bar{Z}_2\bar{Z}_1$. Значення старшого розряду \bar{Z}_4 забезпечується безпосередньо сигналом \bar{P}_1 .

Інформація з виходів першого шифратора подається на входи елементів ЧИ за допомогою схем збігу ЛІ1 при $\bar{V}_1 = 1$ (мікросхема CD1 сприймає вхідні дані, а CD2 – блокована). Інформація з виходів другого шифратора подається на входи елементів ЧИ за допомогою схем збігу ЛІ2 при $\bar{V}_2 = 1$ (мікросхема CD2 сприймає вхідні дані, CD1 – блокується).

Схема працює так: коли на вході є активний сигнал із старшого байта вхідного слова, наприклад, $\bar{F}_{14} = 0$ ($\bar{P}_1 = 0$, $\bar{V}_1 = 1$), то працює шифратор CD1 і на виходах елементів ЧИ формується інверсний код $\bar{Z}_4\bar{Z}_3\bar{Z}_2\bar{Z}_1 = 0001$, що відповідає прямому значенню $Z_4Z_3Z_2Z_1 = 1110 = 14_{10}$. Якщо активний вхідний сигнал відноситься до молодшого байта слова, наприклад, $\bar{F}_6 = 0$, то працює шифратор CD2 ($\bar{P}_1 = 1$, $\bar{V}_1 = 0$, $\bar{V}_2 = 1$) і на інверсних виходах формується код $\bar{Z}_4\bar{Z}_3\bar{Z}_2\bar{Z}_1 = 1001$, що відповідає прямому числу $Z_4Z_3Z_2Z_1 = 0110 = 6_{10}$.

8.1. Загальна характеристика мультиплексорів

Мультиплексором називається функціональний вузол комп'ютера, призначений для почергової комутації (перемикання) інформації від одного з n входів на загальний вихід. Номер конкретної вхідної лінії, що підключається до виходу в кожний такт машинного часу, визначається адресним кодом A_0, A_1, \dots, A_{m-1} . Зв'язок між числом інформаційних n і адресних m входів визначається співвідношенням $n=2^m$. Таким чином, мультиплексор реалізує керовану передачу даних від кількох вхідних ліній в одну вихідну.

Умовне графічне позначення мультиплексорів показано на рис. 8.1. Функція мультиплексорів записується буквами *MUX* (multiplexor).

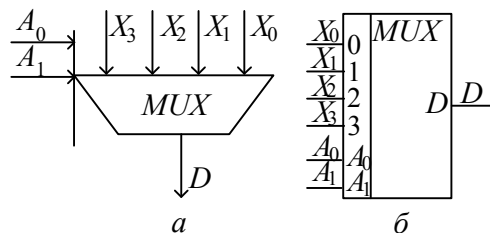
Мультиплексори застосовують для таких операцій: комутації як окремих ліній, так і груп ліній (шин);

перетворення паралельного коду в послідовний; реалізації логічних функцій; побудови схем порівняння, генераторів кодів.

Мультиплексор символічно часто позначають: " $n-1$ ".

Логіка роботи чотиривходового мультиплексора наведена в табл. 4.4, де A_0, A_1 – адресний код; F_0, F_1, F_2, F_3 –

Рис. 8.1. Умовне позначення мультиплексора: а – на функціональних схемах; б – на принципових схемах



виходи внутрішнього дешифратора; X_0, X_1, X_2, X_3 – вхідна інформація; D – загальний інформаційний вихід.

Таблиця 8.1

A_1	A_0	F_0	F_1	F_2	F_3	D
0	0	1	0	0	0	F_0X_0
0	1	0	1	0	0	F_1X_1
1	0	0	0	1	0	F_2X_2
1	1	0	0	0	1	F_3X_3

На основі табл. 8.1. вираз для вихідної функції D можна представити з використанням виходів F_0-F_3 внутрішнього дешифратора у вигляді:

$$D = F_0X_0 \vee F_1X_1 \vee F_2X_2 \vee F_3X_3, \quad (8.1)$$

або з мінтермами адресного коду:

$$D = \overline{A_1} \overline{A_0} X_0 \vee \overline{A_1} A_0 X_1 \vee A_1 \overline{A_0} X_2 \vee A_1 A_0 X_3. \quad (8.2)$$

Схеми мультиплексорів, відповідні рівнянням (8.1) і (8.2), показані на рис. 8.2.

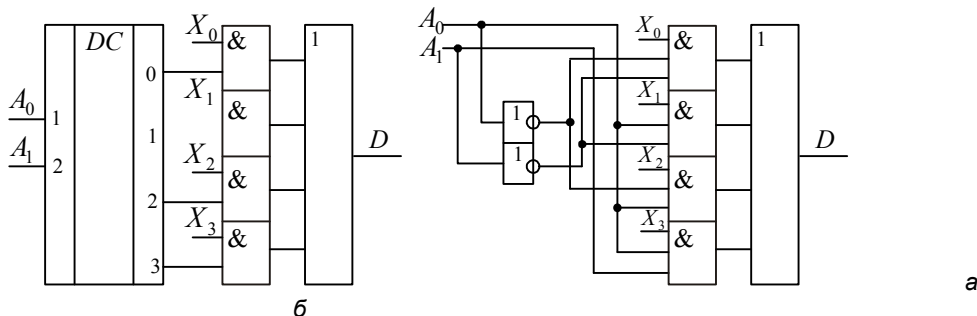


Рис. 8.2. Схеми мультиплексорів: а – з внутрішнім дешифратором; б – з адресними мінтермами

При побудові схеми мультиплексорів з внутрішнім лінійним дешифратором потрібні логічні елементи з меншим числом входів, проте при цьому збільшується час встановлення вихідного сигналу. При виключенні дешифратора швидкодія підвищується, однак потрібні схеми збігу з більшим числом входів.

8.2. Каскадування мультиплексорів

В інтегральному виконанні мультиплексори випускають на чотири, вісім або шістнадцять входів. Каскадування дозволяє реалізувати комутацію довільного числа вхідних ліній на базі серійних мікросхем мультиплексорів меншої розрядності.

Приклад побудови схеми мультиплексора на 16 входів на основі типових чотиривходових мультиплексорів показаний на рис. 8.3.

Молодші розряди адреси A_1, A_0 підключаються до адресних входів усіх мультиплексорів першого рівня, на виходах яких виробляються такі функції:

$$D'_0 = F_0X_0 \vee F_1X_1 \vee F_2X_2 \vee F_3X_3;$$

$$D'_1 = F_0 X_4 \vee F_1 X_5 \vee F_2 X_6 \vee F_3 X_7;$$

$$D'_2 = F_0 X_8 \vee F_1 X_9 \vee F_2 X_{10} \vee F_3 X_{11};$$

$$D'_3 = F_0 X_{12} \vee F_1 X_{13} \vee F_2 X_{14} \vee F_3 X_{15},$$

де $F_0 - F_3$ – виходи внутрішніх дешифраторів: $F_0 = \overline{A_1} \overline{A_0}$; $F_1 = \overline{A_1} A_0$; $F_2 = A_1 \overline{A_0}$; $F_3 = A_1 A_0$; $X_{15} - X_0$ – вхідні змінні.

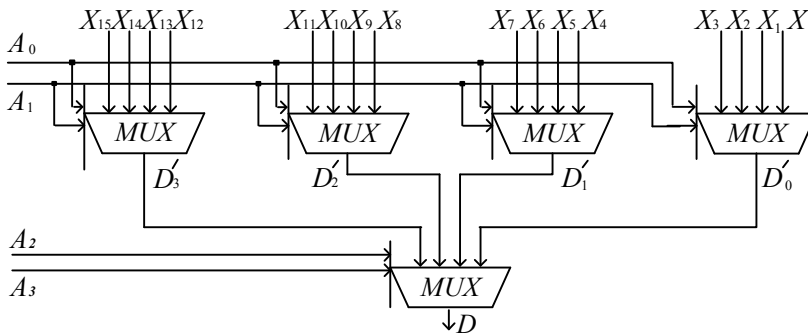


Рис. 8.3. Каскадування мультиплексорів

Старші розряди адреси A_3, A_2 подаються на адресні входи мультиплексора другого рівня, на виході якого формується остаточна функція

$$D = F'_0 D'_0 \vee F'_1 D'_1 \vee F'_2 D'_2 \vee F'_3 D'_3,$$

де внутрішні виходи дешифратора визначаються такими мінтермами:

$$F_0 = \overline{A_3} \overline{A_2}; F_1 = \overline{A_3} A_2; F_2 = A_3 \overline{A_2}; F_3 = A_3 A_2.$$

Нехай, наприклад, значення адреси $A_3 A_2 A_1 A_0 = 1011_2 = 11_{10}$. При цьому на вхід другого рівня комутується змінна D'_2 (оскільки $F'_2 = 1$), значення якої визначається з рівнянь першого рівня:

$$D'_2 = F_3 X_{11} = A_1 A_0 X_{11} = 1 X_{11} = X_{11}.$$

8.3 Реалізація логічних функцій мультиплексорами

За допомогою мультиплексорів реалізуються логічні функції з числом змінних m , що дорівнює розрядності адресного коду. Функція, що виконується, має бути представлена в ДДНФ. При цьому змінні поступають на адресні входи, а інформаційні входи використовуються як настроювальні – на них подаються константи нуля і одиниці реалізуються.

Вихідна функція триадресного мультиплексора на вісім входів описується рівнянням:

$$D(A) = \overline{A_2} \overline{A_1} \overline{A_0} X_0 \vee \overline{A_2} \overline{A_1} A_0 X_1 \vee \overline{A_2} A_1 \overline{A_0} X_2 \vee \overline{A_2} A_1 A_0 X_3 \vee \\ \vee A_2 \overline{A_1} \overline{A_0} X_4 \vee A_2 \overline{A_1} A_0 X_5 \vee A_2 A_1 \overline{A_0} X_6 \vee A_2 A_1 A_0 X_7.$$

Якщо потрібно отримати логічну функцію з десятковими еквівалентами мінтермів 1, 3, 5 і 7, то на парні входи X_0, X_2, X_4 і X_6 необхідно подати константу "0", а на непарні X_1, X_3, X_5 і X_7 – константу "1". У результаті отримаємо (рис. 8.4):

$$D(A) = \overline{A_2} \overline{A_1} A_0 \vee \overline{A_2} A_1 A_0 \vee A_2 \overline{A_1} A_0 \vee A_2 A_1 A_0.$$

За допомогою додаткових логічних перетворень можна реалізувати логічні функції з числом змінних $m+1$, тобто на одиницю більше розрядності адресного коду мультиплексора.

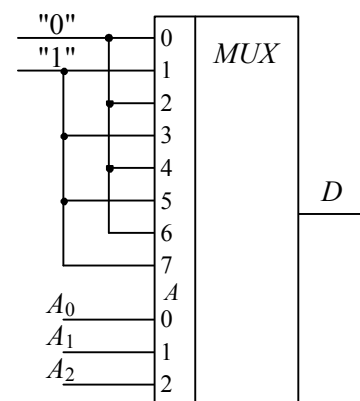
Можливі й інші схемотехнічні застосування мультиплексорів. Мультиплексор може виконувати функції перетворювача n -розрядного двійкового паралельного коду в послідовний.

Для цього потрібно подати паралельний код на інформаційні входи мультиплексора і потім змінювати код на адреси в необхідній послідовності.

8.4 Мультиплексування шин

Мультиплексування шин – це почергове перемикання шин (груп ліній) від кількох джерел інформації до одного приймача. Такі мікрооперації реалізуються схемами на основі мультиплексорів одиночних ліній. При виборі кількості й типу мультиплексорів враховують:

Рис. 8.4. Схема мультиплексора для реалізації логічної функції трьох змінних з десятковими еквівалентами мінтермів 1, 3, 5 і 7



- число комутуваних шин дорівнює 2^m , де m – довжина адресного коду;
- i -й номер входу всіх мультиплексорів служить для підключення розрядів певної однієї шини.

Схема мультиплексора чотирьох $X(n)$, $Y(n)$, $Z(n)$ і $S(n)$ шин показана на рис. 8.5. Для її побудови потрібно n двоадресних чотиривходових мультиплексорів, де n – довільна розрядність шин, що комутуються.

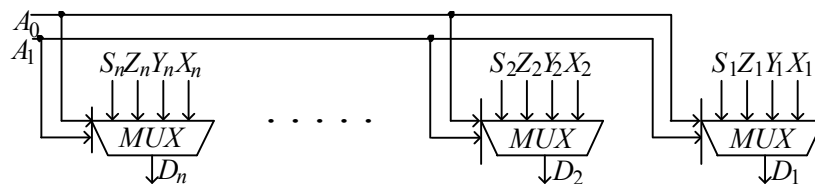


Рис. 8.5. Мультиплексор шин

8.5. Загальна характеристика демультиплексорів

Демультиплексором називається функціональний вузол комп'ютера, призначений для комутації (перемикання) сигналу з одного інформаційного входу D на один з n інформаційних виходів. Номер виходу, на який в кожний такт машинного часу передається значення вхідного сигналу, визначається адресним кодом $A_0, A_1, A_2, \dots, A_{m-1}$. Адресні входи m та інформаційні виходи n пов'язані співвідношенням $n=2^m$ або $m=\log_2 n$.

Демультиплексор виконує функцію, обернену функції мультиплексора. Стосовно мультиплексорів і демультиплексорів користуються також терміном "селектори" даних.

В умовних графічних позначеннях (рис. 8.6) функція демультиплексора позначається буквами DMX .

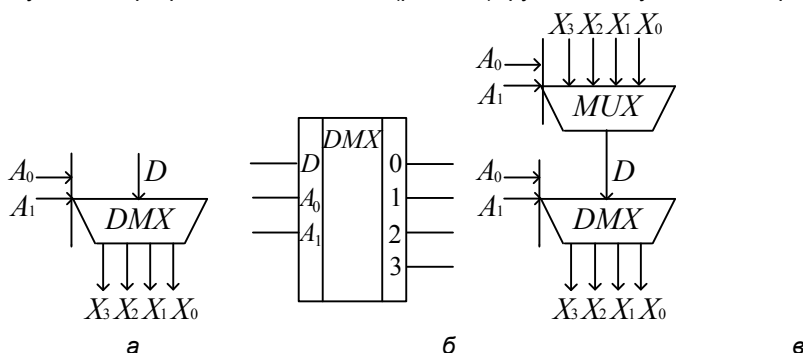


Рис. 8.6. Умовні графічні позначення демультиплексорів: а – на функціональних схемах; б – на принципових схемах; в – типове з'єднання з мультиплексором

Демультиплексори використовують для таких операцій:

- комутації як окремих ліній, так і багаторозрядних шин;
- перетворення послідовного коду в паралельний;
- реалізації логічних функцій та інших.

Демультиплексори часто позначають: " $1 \rightarrow n$ ".

Логіка роботи двоадресного демультиплексора на мові мікрооперацій наведена в табл. 8.2, де D – інформаційний вхід; F_0, F_1, F_2 і F_3 – виходи внутрішнього дешифратора адреси.

Таблиця 8.2

A_1	A_0	F_0	F_1	F_2	F_3	X_0	X_1	X_2	X_3
0	0	1	0	0	0	$F_0 D$	–	–	–
0	1	0	1	0	0	–	$F_1 D$	–	–
1	0	0	0	1	0	–	–	$F_2 D$	–
1	1	0	0	0	1	–	–	–	$F_3 D$

За даними табл. 4.5 запишемо систему рівнянь для інформаційних виходів:

$$\begin{aligned} X_0 &= F_0 D = \bar{A}_1 \bar{A}_0 D; & X_1 &= F_1 D = \bar{A}_1 A_0 D; \\ X_2 &= F_2 D = A_1 \bar{A}_0 D; & X_3 &= F_3 D = A_1 A_0 D; \end{aligned} \quad (8.3)$$

На основі рівнянь (8.3) побудовані схеми демультиплексорів із внутрішнім дешифратором (рис. 8.7, а) і з поєднанням адресних і вхідних змінних на тривходових елементах І (рис. 8.7, б).

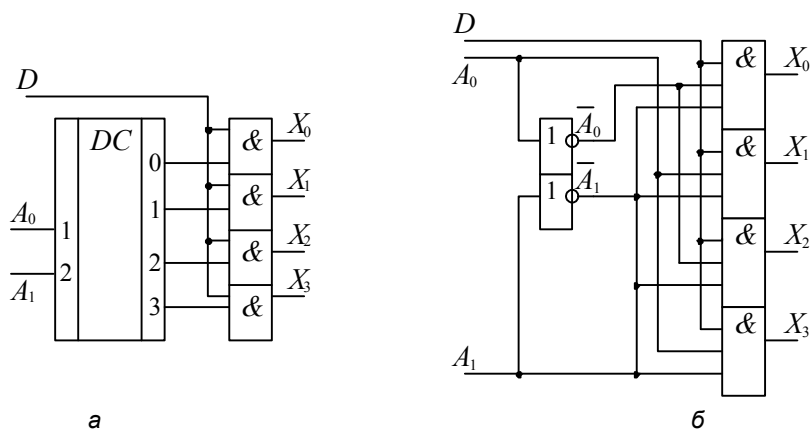


Рис. 8.7. Схема демультиплексорів: а – з внутрішнім дешифратором; б – з поєднанням адресних і вхідних змінних

Схема демультиплексора з поєднанням адресних і вхідних змінних забезпечує високу швидкодію, проте вимагає застосування логічних елементів з більшим числом входів.

8.6. Каскадування демультиплексорів

Каскадування дозволяє реалізувати комутацію одного вхідного сигналу на довільне число вихідних ліній на базі серійних мікросхем меншої розрядності. Нехай потрібно реалізувати демультиплексування вхідного сигналу на n вихідних ліній, що визначаються m -розрядним адресним кодом, на базі типових мікросхем меншої розрядності виду “1 → n ”.

Для цього потрібно використати $L = n/n_1$ типових демультиплексорів з числом адресних входів $m_1 = \log_2 n_1$ кожен. Число старших адресних розрядів, що дорівнює різниці $m - m_1$, використовується додатковим “ведучим” демультиплексором, який розташовується у першому рівні схеми каскадування. Ведучий демультиплексор визначає почергове увімкнення одного з L демультиплексорів мікросхем другого рівня. Каскадування демультиплексорів виду “1 → 4” для реалізації комутатора “1 → 16” показано на рис. 8.8.

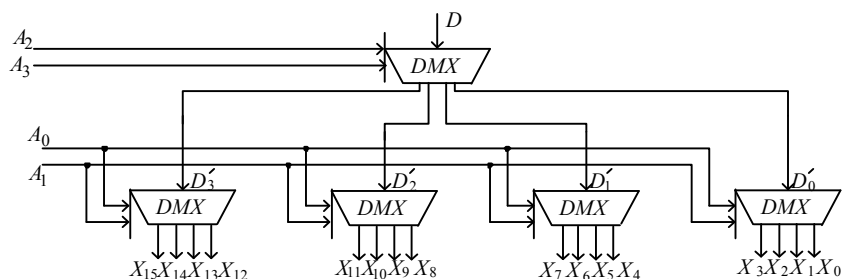


Рис. 8.8. Каскадування демультиплексорів

Нехай для схеми (рис. 8.8) адресний код $A_3A_2A_1A_0 = 1010$ і значення вхідного сигналу $D=1$. Тоді на виході ведучого демультиплексора $D'_2 = A_3 \bar{A}_2 D = 1$, а на інших виходах встановлюються нульові значення. Одиначне значення сигналу D'_2 передається на вихід X_{10} веденого демультиплексора згідно зі співвідношенням

$$X_{10} = A_1 \bar{A}_0 D_2 = 1.$$

Демультиплексори не випускають як самостійні вироби на інтегральних мікросхемах. Функцію демультиплексора звичайно реалізують на дешифраторах, що мають входи стробування (дешифратори-демультиплексори).

8.7. Демультиплексування шин

Під демультиплексуванням шин розуміється почергове перемикання груп ліній від одного джерела інформації до багатьох приймачів. Такі мікрооперації реалізуються звичайно на основі демультиплексорів одиночних ліній. При виборі кількості і типу демультиплексора враховують:

- число шин, які комутуються, дорівнює 2^m , де m – довжина адресного коду;
- кількість демультиплексорів, які використовуються, визначається розрядністю n шин, які демультиплексуються;
- адресні входи всіх мультиплексорів паралельно об'єднуються.

Схема мультиплексора вхідної шини $D(n)$ на чотири вхідні шини $X(n)$, $Y(n)$, $Z(n)$ і $S(n)$ показана на рис. 8.9.

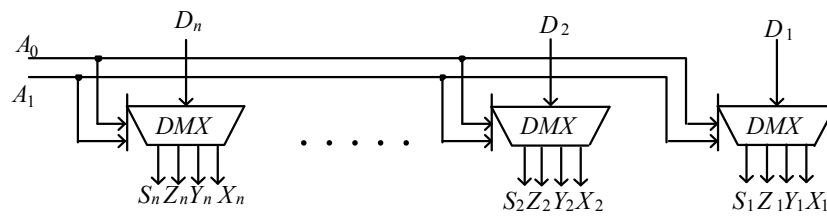


Рис. 8.9. Демультимплексор шин

Типове включення мультимплексорів і демультимплексорів для комутації вхідних і вихідних шин n -розрядних регістрів A , B , C і D показано на рис.8.10.

У АЛП така комутація забезпечує використання як першого операнда суматора вміст будь-якого регістра і запис результату операції в будь-який регістр, вказаний мікропрограмою команди, що виконується.

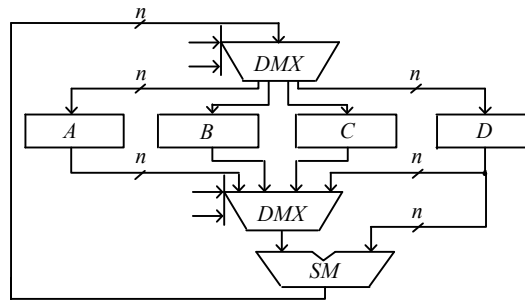


Рис. 8.10. Типова комутація вхідних і вихідних шин регістрів АЛП

10.1. Загальна характеристика суматорів

Суматором називається функціональний вузол комп'ютера, призначений для додавання двох n -розрядних слів (чисел). Операція віднімання замінюється додаванням слів в оберненому або доповняльному коді. Операції множення та ділення зводяться до реалізації багаторазового додавання та зсування. Тому суматор є важливою частиною арифметико-логічного пристрою. Функція суматора позначається буквами SM або Σ .

Суматор складається з окремих схем, які називаються однорозрядними суматорами; вони виконують усі дії з додавання значень однойменних розрядів двох чисел (операндів). Суматори класифікуються за такими ознаками:

- способом додавання – паралельні, послідовні та паралельно-послідовні;
- числом входів – напівсуматори, однорозрядні та багаторозрядні суматори;
- організацією зберігання результату додавання – комбінаційні, накопичувальні, комбіновані;
- організацією перенесення між розрядами – з послідовним, наскрізним, паралельним або комбінованим перенесеннями (з груповою структурою);
- системою числення – позиційні (двійкові, двійково-десяткові, трійкові) та непозиційні, наприклад, у системі залишкових класів;
- розрядністю (довжиною) операндів – 8-, 16-, 32-, 64-розрядні;
- способом представлення від'ємних чисел – в оберненому або доповняльному кодах, а також в їхніх модифікаціях;
- часом додавання – синхронні, асинхронні.

У паралельних n -розрядних суматорах значення всіх розрядів операндів поступають одночасно на відповідні входи однорозрядних підсумовуючих схем. У послідовних суматорах значення розрядів операндів та перенесення, що запам'ятовувалися в минулому такті, поступають послідовно в напрямку від молодших розрядів до старших на входи одного однорозрядного суматора. В паралельно-послідовних суматорах числа розбиваються на частини, наприклад, байти, розряди байтів поступають на входи восьмирозрядного суматора паралельно (одночасно), а самі байти – послідовно, в напрямку від молодших до старших байтів з урахуванням запам'ятованого перенесення.

У комбінаційних суматорах результат операції додавання запам'ятовується в регістрі результату. В накопичувальних суматорах процес додавання поєднується із зберіганням результату. Це пояснюється використанням T -тригерів як однорозрядних схем додавання.

Організація перенесення практично визначає час виконання операції додавання. Послідовні перенесення схемно створюються просто, але є повільнодіючими. Паралельні перенесення схемно організуються значно складніше, але дають високу швидкодію.

Розрядність суматорів знаходиться в широких границях: 4–16 – для мікро- та міні-комп'ютерів та 32–64 і більше – для універсальних машин.

Суматори з постійним інтервалом часу для додавання називаються синхронними. Суматори, в яких інтервал часу для додавання визначається моментом фактичного закінчення операції, називаються асинхронними. В асинхронних суматорах є спеціальні схеми, які визначають фактичний момент закінчення додавання і повідомляють про це в пристрій керування. На практиці переважно використовуються синхронні суматори. Суматори характеризуються такими параметрами:

швидкодією – часом виконання операції додавання t_{Σ} , який відраховується від початку подачі операндів до одержання результату; часто швидкодія характеризується кількістю додавання в секунду $F_{\Sigma}=1/t_{\Sigma}$, тут мають на увазі операції типу регістр–регістр (тобто числа зберігаються в регістрах АЛП);

апаратними витратами: вартість однорозрядної схеми додавання визначається загальним числом логічних входів використаних елементів; вартість багаторозрядного суматора визначається загальною кількістю використаних мікросхем;

споживану потужність суматора.

10.2. Однорозрядні суматори

Однорозрядним суматором називається логічна схема, яка виконує додавання значень i -х розрядів X_i та Y_i двійкових чисел з урахуванням перенесення Z_i з молодшого сусіднього розряду та виробляє на виходах функції результат S_i і перенесення P_i в старший сусідній розряд. На основі однорозрядних схем додавання на три входи та два виходи будуються багаторозрядні суматори будь-якого типу. Алгоритм роботи однорозрядного суматора відображається таблицею істинності (табл. 10.1).

На основі табл. 10.1 записується система логічних функцій для результату S_i та перенесення P_i у ДДНФ:

Таблиця 10.1

X_i	Y_i	Z_i	S_i	P_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S_i = \bar{X}_i \bar{Y}_i Z_i \vee \bar{X}_i Y_i \bar{Z}_i \vee X_i \bar{Y}_i \bar{Z}_i \vee X_i Y_i Z_i; \quad (10.1)$$

$$P_i = \bar{X}_i Y_i Z_i \vee X_i \bar{Y}_i Z_i \vee X_i Y_i \bar{Z}_i \vee X_i Y_i Z_i. \quad (10.2)$$

Мінімізація функцій (10.1) та (10.2) за допомогою карт Карно показана на рис. 10.1.

Як видно з карт Карно, функція результату S_i не мінімізується, а функція P_i мінімізується зі зниженням рангу кон'юнкції та використовує тільки прямі значення змінних:

$$P_i = X_i Y_i \vee X_i Z_i \vee Y_i Z_i = X_i Y_i \vee (X_i \vee Y_i) Z_i. \quad (10.3)$$

				$X_i Y_i$	Z_i
00	01	11	10	0	1
0	1	0	1		
1	0	1	0	0	1

a

				$X_i Y_i$	Z_i
00	01	11	10	0	1
0	0	1	0		
0	1	1	1	0	1

б

Рис. 10.1. Карты Карно для мінімізації функцій: *a* – S_i ; *б* – P_i

При проектуванні комбінаційних однорозрядних суматорів враховують такі чинники:

схема має характеризуватися регулярністю (подібністю) структури та мінімальною вартістю, тобто мати по можливості найменше число логічних входів всіх елементів;

з метою підвищення швидкодії багаторозрядного суматора потрібен мінімальний час одержання функції перенесення $t_n = k t_p$, де k – число послідовно увімкнених елементів від входів до виходів P_i або \bar{P}_i ; t_p – середня затримка розповсюдження сигналу одним логічним елементом в обраній серії інтегральних мікросхем; параметр k часто називають каскадністю (поверховістю) схем. Таким чином, для мінімізації часу одержання перенесення необхідно зменшити каскадність схеми та використати інтегральні мікросхеми з малим часом затримки розповсюдження сигналу;

для схем однорозрядних суматорів на основі рівнянь (10.1) і (10.2) необхідно виробляти як прямі P_i , так й інверсні \bar{P}_i значення функції перенесення. Така організація перенесень називається парафазною.

Для побудови схеми однорозрядного суматора на універсальних логічних елементах НЕ І рівняння (10.1) і (10.2) перетворюються на основі правил подвійної інверсії та де Моргана до такого вигляду:

$$S_i = \overline{X_i Y_i Z_i} \cdot \overline{X_i Y_i Z_i} \cdot \overline{X_i Y_i Z_i} \cdot \overline{X_i Y_i Z_i}; \quad P_i = \overline{X_i Y_i} \cdot \overline{X_i Z_i} \cdot \overline{Y_i Z_i}. \quad (10.4)$$

Схема однорозрядного суматора, побудована на елементах НЕ І відповідно до рівнянь (4.30), показана на рис. 10.2, а; її вартість, яка вимірюється числом логічних входів всіх елементів, становить 27, каскадність $k=3$.

Рівняння (10.1) та (10.2) можуть бути виражені через функцію «Виключальне ЧИ»:

$$S_i = (X_i \oplus Y_i) \bar{Z}_i \vee (\bar{X}_i \oplus \bar{Y}_i) Z_i = X_i \oplus Y \oplus Z_i; \quad (10.5)$$

$$P_i = X_i Y_i \vee (\bar{X}_i Y_i \vee X_i \bar{Y}_i) Z_i = X_i Y_i \vee (X_i \oplus Y_i) Z_i. \quad (10.6)$$

Схема однорозрядного суматора на елементах «виключальне ЧИ» згідно з рівняннями (10.5) і (10.6) показана на рис. 10.2, б; її вартість становить вісім входів і каскадність $k=2$.

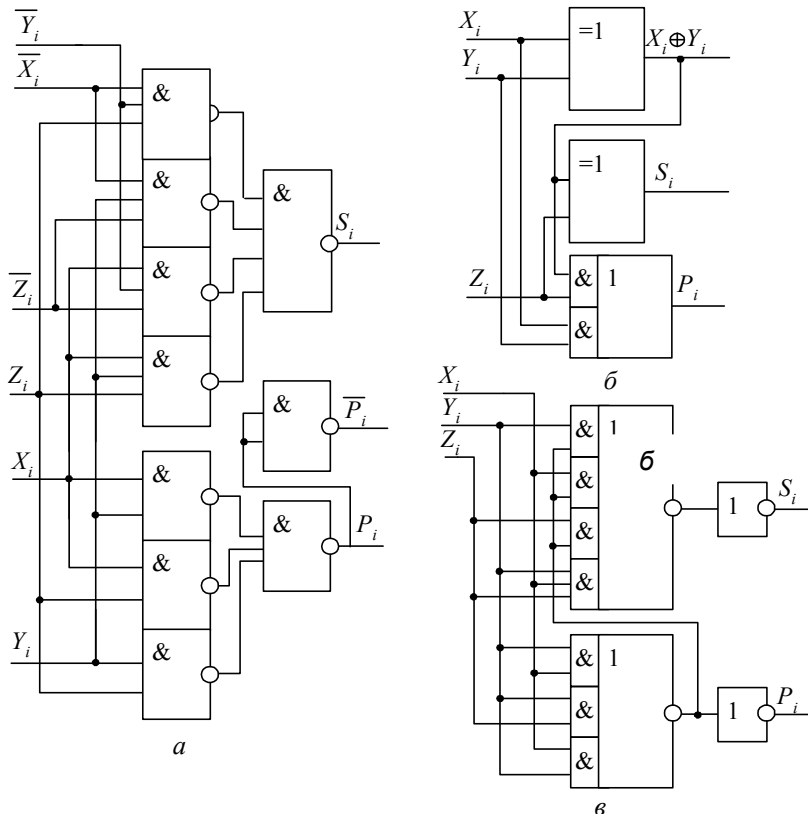


Рис. 10.2. Схеми однорозрядних суматорів: а – на елементах НЕ І; б – на елементах «виключальне ЧИ»; в – з використанням власного перенесення

Функції однорозрядного суматора – самоподвійні, тобто їхні інверсії утворюються інвертуванням значень аргументів без зміни місцезнаходження знаків диз'юнкції та кон'юнкції, наприклад, для перенесення з рівняння (10.2):

$$\bar{P}_i = X_i \bar{Y}_i \bar{Z}_i \vee \bar{X}_i Y_i \bar{Z}_i \vee \bar{X}_i \bar{Y}_i Z_i \vee X_i Y_i Z_i. \quad (10.7)$$

Помножуючи ліві та праві частини співвідношення (10.7) на макстерм $(X_i \vee Y_i \vee Z_i)$, одержують:

$$\bar{P}_i X_i \vee \bar{P}_i Y_i \vee \bar{P}_i Z_i = X_i \bar{Y}_i \bar{Z}_i \vee \bar{X}_i Y_i \bar{Z}_i \vee \bar{X}_i \bar{Y}_i Z_i. \quad (10.8)$$

Після підстановки лівої частини співвідношення (10.8) в праву частину виразу (10.1) одержують рівняння для функції S_i з використанням власного перенесення:

$$S_i = \bar{P}_i X_i \vee \bar{P}_i Y_i \vee \bar{P}_i Z_i \vee X_i Y_i Z_i. \quad (10.9)$$

Схема однорозрядного суматора відповідно до рівнянь (10.9) і (10.3) показана на рис. 10.2, а; її вартість дорівнює 17 входів, каскадність $k=2$. Важливою властивістю цієї схеми є використання тільки прямих значень вхідних змінних і однофазного ланцюга формування перенесення P_i в старший розряд.

Напівсуматором називається логічна схема, яка виконує додавання значень i -х розрядів X_i і Y_i двійкових чисел X і Y та реалізує на виході значення результату M_i і перенесення в старший сусідній розряд R_i :

$$M_i = \bar{X}_i Y_i \vee X_i \bar{Y}_i = X_i \oplus Y_i; \quad R_i = X_i Y_i. \quad (10.10)$$

Таким чином, напівсуматор виконує лише частину завдання підсумовування в i -му розряді, оскільки не враховує перенесення з сусіднього молодшого розряду. Схема напівсуматора, побудована на основі рівнянь (10.10), показана на рис. 10.3. З рівнянь (10.5) і (10.6) виходить, що схема однорозрядного суматора може бути побудована на основі двох напівсуматорів і додаткового логічного елемента ЧИ, як показано на рис. 10.3, в.

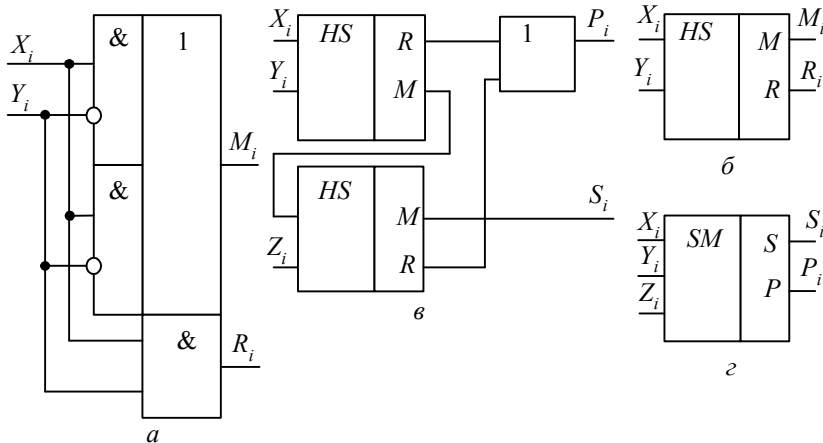


Рис. 10.3. Схеми підсумовування: а, б – напівсуматор і його умовне позначення; в, г – однорозрядний суматор і його умовне позначення

10.3. Послідовний багаторозрядний суматор

Послідовний двійковий багаторозрядний суматор містить: n -розрядні зсуваючі регістри операндів X і Y , регістр результату S , однорозрядний суматор SM і двоступеневий D -тригер для запам'ятовування перенесення. Усі регістри забезпечують одночасне зсування праворуч, у бік молодших розрядів (рис. 10.4).

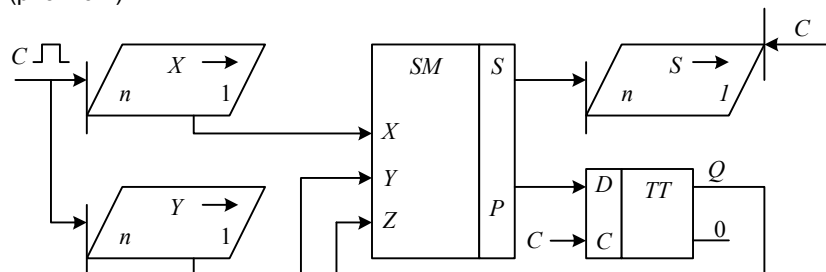


Рис. 10.4. Схема послідовного багаторозрядного суматора

У послідовному суматорі попарна подача значень розрядів X_i і Y_i починається з молодших розрядів. Утворюються значення суми S_i і перенесення P_i , які записуються відповідно в регістр результату та в тригер запам'ятовування перенесення на один такт T_c .

Послідовне додавання виконується за стільки тактів, скільки розрядів у числі. Тому час додавання t_Σ визначається співвідношенням: $t_\Sigma = nT_c$, де T_c – тривалість машинного такту.

Від'ємні числа рекомендується представляти в доповняльному коді.

Послідовний суматор потребує мінімальних апаратних витрат, однак тривалість операції додавання пропорційна розрядності операндів. Тому послідовний суматор можна використовувати у відносно

повільнодіючих цифрових пристроях.

10.4. Паралельні багаторозрядні суматори

Паралельний багаторозрядний суматор містить n однорозрядних схем додавання, наприклад, чотири, як показано на рис. 10.5.

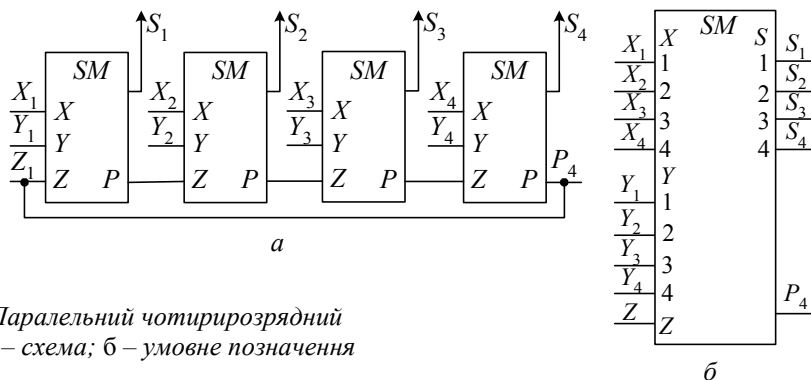


Рис. 10.5. Паралельний чотирирозрядний суматор: а – схема; б – умовне позначення

Значення всіх розрядів двох чисел X та Y поступають на входи відповідних однорозрядних суматорів паралельно (одночасно). В паралельних суматорах з послідовним перенесенням значення сигналу перенесення P_i передається від розряду до розряду послідовно в часі (асинхронно). При застосуванні оберненого коду перенесення з найстаршого розряду подається на вхід перенесення молодшого розряду по ланцюзі циклічного перенесення (рис. 10.5, а). При застосуванні доповняльного коду ланцюг циклічного перенесення розривається, а на вхід перенесення молодшого розряду подається логічний нуль.

У паралельних суматорах з послідовним перенесенням час додавання визначається співвідношенням:

$$t_s = (n-1) t_1 + t_s,$$

де t_1 – час формування перенесення в кожному розряді, t_s – час додавання в найстаршому розряді. У гіршому випадку можливий варіант, коли сигнал перенесення послідовно розповсюджується від першого до n -го розряду.

10.5. Мікросхеми ALU

Промисловість випускає мікросхеми із символом функції ALU для виконання 16 арифметичних та 16 порозрядних логічних мікрооперацій залежно від вхідних сигналів настройки. У серіях ТТЛШ 530, 531, 533, 555 та 1533 вони мають позначення ИПЗ; в серіях ЕЗЛ 100, 500 і 700 використовують позначення ИП179.

Мікросхема ALU в серіях ТТЛШ має (рис. 10.6):

- інформаційні входи для подання двох чотирирозрядних операндів X і Y ;
- входи налаштування E_3 – E_0 для задання номера однієї з мікрооперацій;
- вхід M для задання типу мікрооперації: $M=0$ – арифметичні, $M=1$ – логічні;
- вхід перенесення $\overline{C1}$, необхідний тільки при виконанні арифметичних мікрооперацій;
- виходи: результату мікрооперації S_4 – S_1 , послідовного перенесення L , генерації G , транзиту H , а також вихід з відкритим колектором від внутрішнього компаратора для вироблення ознаки рівності операндів $F_{A=B}$.

Перелік арифметичних і логічних операцій, які виконують ALU, наведений у табл. 10.2. При виконанні логічних операцій перенесення між розрядами не використовується. Арифметичні операції реалізуються з урахуванням перенесень і позик. В арифметичні операції включені фрагменти логічних дій. Наприклад, запис $(X \vee Y) + X\overline{Y}$ означає, що спочатку виконується операція інверсії (\overline{Y}), потім – логічного додавання $(X \vee Y)$ та логічного множення $(X \cdot \overline{Y})$, а потім одержані таким чином два числа додаються арифметично з урахуванням перенесень.

Таблиця 10.2

E_3	E_2	E_1	E_0	Логіка $M=1$	Арифметика $M=0$
0	0	0	0	\overline{X}	X
0	0	0	1	$\overline{X \vee Y}$	$X \vee Y$
0	0	1	0	\overline{XY}	$X \vee \overline{Y}$
0	0	1	1	0	-1
0	1	0	0	$\overline{X\overline{Y}}$	$X + X\overline{Y}$
0	1	0	1	\overline{Y}	$(X \vee Y) + X\overline{Y}$
0	1	1	0	$X \oplus Y$	$X - \overline{Y} - 1$
0	1	1	1	$X\overline{Y}$	$X\overline{Y} - 1$

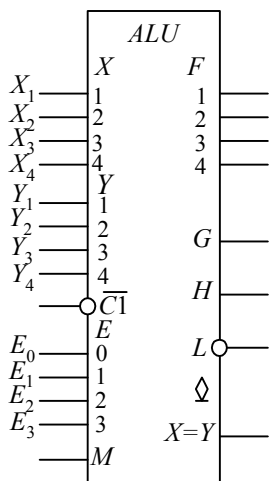


Рис. 10.6. Умовні позначення мікросхем: а – ALU 1533ИПЗ; б – CRU 1533ИП4

1	0	0	0	$\overline{X \vee Y}$	$X + XY$
---	---	---	---	-----------------------	----------