

Лекція 8

2. Обчислювальні пристрої

2.1. Суматори

Найпростішою схемою є *напівсуматор*, який виконує операцію знаходження суми двох однорозрядних слів.

Роботу напівсуматора описує Табл. 7.4, з якої знаходимо логічні рівняння для суми і для переносу:

$$S = a_0 \cdot \overline{b_0} + \overline{a_0} \cdot b_0 ;$$

$$P = a_0 \cdot b_0 .$$

Таблиця 7.4

a_0	b_0	S	P
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

При знаходженні суми багаторозрядних слів напівсуматор може використовуватись лише в молодшому розряді. Для знаходження суми старших розрядів схема повинна мати три входи – два інформаційні і третій – вхід переносу з молодших розрядів. Схема, що відповідає цим вимогам, називається *повним суматором*, а логіку її роботи описує Табл. 7.5.

Таблиця 7.5

N	a_i	b_i	p_{i-1}	S_i	P_{i+1}
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

З Табл. 7.5 знаходимо наступні рівняння:

$$S_i = \vee 1, 2, 4, 7 = p_{i-1} (\bar{a}_i \cdot \bar{b}_i + a_i \cdot b_i) + \bar{p}_{i-1} (a_i \cdot \bar{b}_i + \bar{a}_i \cdot b_i) ;$$

$$P_i = \vee 3, 5, 6, 7 = p_{i-1} (\bar{a}_i \cdot b_i + a_i \cdot \bar{b}_i + a_i \cdot b_i) + \bar{p}_{i-1} a_i \cdot b_i = a_i \cdot b_i + p_{i-1} (a_i \oplus b_i)$$

для побудови логічної схеми однорозрядного повного суматора.

Знаходження суми багаторозрядних слів може виконуватись як послідовно, так і паралельно. Для послідовного виконання операції використовується один повний суматор, загальний для всіх розрядів. Для виконання операції суми слова подаються на інформаційні розряди послідовно, починаючи з молодших розрядів (рис. 7.12).

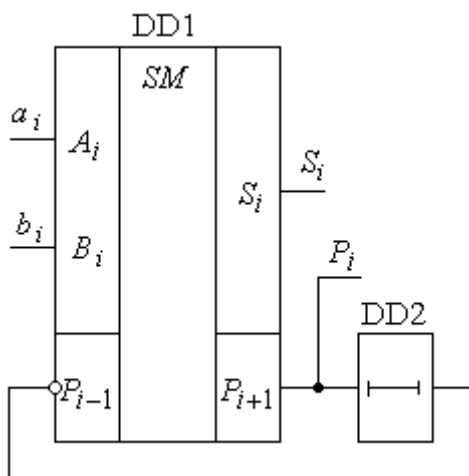


Рис. 7.12

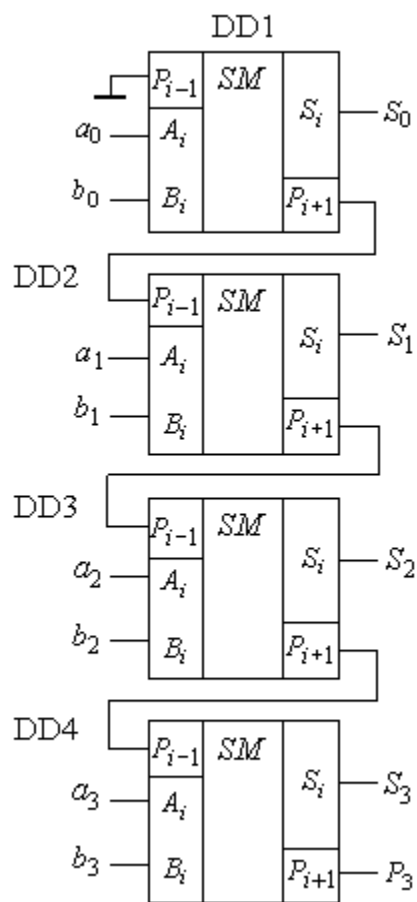


Рис. 7.13

Для врахування результату операції з попереднім розрядом вхід та вихід переносу об'єднується через елемент затримки DD2. Цей елемент повинен забезпечити затримку та зберігання сигналу P_{i+1} на один такт виконання операції. Синхронно з інтервалом в один такт повинні подаватись і вхідні дані a_i, b_i .

Приведена схема виконання операції суми сама по собі нескладна, але вимагає складного апаратного забезпечення. До того ж, послідовне порозрядне виконання операції суттєво знижує швидкодію.

На рис. 7.13 приведена схема організації чотирьохрозрядного паралельного суматора з послідовним переносом. Для кожного розряду в ній використовуються окремі однорозрядні повні суматори, які з'єднані між собою послідовно по виходах-входах переносу. Незважаючи на паралельну одночасну подачу всіх розрядів слів, операція в i -му розряді починає виконуватись тільки після того, як поступить сигнал переносу з $(i-1)$ -го розряду. Звідси витікає, що швидкодія суматора визначатиметься сумою затримок передачі сигналу переносу з молодшого розряду на вихід суматора старшого розряду.

За схемою, приведеною на рис. 7.13, виготовляються мікросхеми К1533ИМ3 (зарубіжні аналоги – 7483PC, CDB483E, FLH241, MC7483N, N7483N, SN7482AN, UCY7483N), умовне зображення якої приведено на рис. 7.14 (мікросхема DD2). Мікросхема може використовуватись для операцій як з додатними, так і з від'ємними числами. При виконанні операції додавання на вхід p_0 необхідно подати сигнал низького рівня.

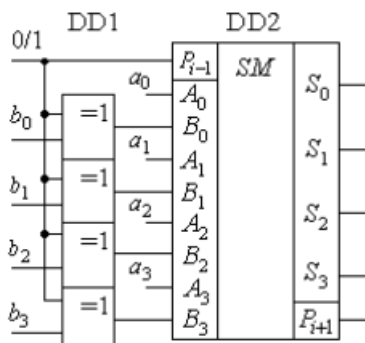


Рис. 7.14

Нарощування розрядності суматорів виконується на схемі, приведеній на рис. 7.13. Таблиця станів для суматорів за своєю сутністю є таблицею суми двох чисел (Див. Табл. 7.4 – Табл. 7.5). Вона має простий вигляд і зручна в користуванні для чисел невеликої розрядності. Суматор К1533ИМЗ, як і всі комбінаційні схеми, має широкі функціональні можливості. Перш за все, він може використовуватись для знаходження суми чисел з меншою розрядністю. Схема входів-виходів при цьому може бути різноманітною. Його можливо використовувати і для віднімання двох чисел. Ця операція виконується шляхом додавання зменшуваного і зменшуючого, представленого у доповнюючому коді. Для виконання цієї операції зменшуюче $b_3 b_2 b_1 b_0$ з допомогою інверторів перетворюється в зворотній код, а потім до нього по входу p_0 додається одиниця.

Приклад 7. Розглянути мікросхему К1533ИМЗ. Пояснити, як використовувати цей чотирьохрозрядний суматор для додавання:

- а) двох чотирьохрозрядних чисел;
- б) двох дворозрядних та двох однорозрядних чисел;
- в) двох восьмирозрядних (байтових) чисел.

Розв'язання. При додаванні двох чотирьохрозрядних слів необхідно вхід переносу P_0 заземлити (рис. 7.15). В дворозрядному суматорі входами є A_1, A_2 та B_1, B_2 , а виходами S_1, S_2, S_3 . Останній є виходом переносу. В однорозрядному суматорі використовуються входи

A_4, B_4 , а виходами є S_4 та P_4 . При додаванні двох однобайтових слів необхідно наростити два однотипних суматори. При цьому вихід P_4 суматора напівбайтів молодших розрядів з'єднується з входом P_0 суматора напівбайтів старших розрядів слів, що складаються.

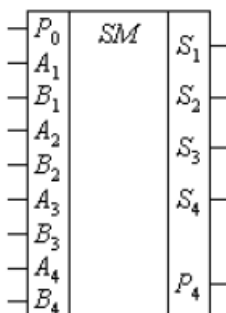


Рис. 7.15

Операції знаходження суми та різниці можна поєднати в одному пристрої, як показано на рис. 7.14. Для виконання операції додавання на вхід P_{i-1} переносу подається сигнал низького рівня. Відповідно на входах елементів **ВИКЛ. АБО** число $b_3 b_2 b_1 b_0$ буде представлено в прямому коді. При подачі на вхід P_{i-1} сигналу високого рівня, як відмічалось раніше, число $b_3 b_2 b_1 b_0$ буде представлено у доповнюючому коді.

Слід нагадати, що при виконанні операції віднімання результат на виході суматора залежить від співвідношення між числами, що беруть участь в операції. При

$$a_3 a_2 a_1 a_0 > b_3 b_2 b_1 b_0$$

результат одержується в прямому коді, а в розряді переносу P_{i+1} з'являється **1**.

Коли ж

$$a_3 a_2 a_1 a_0 < b_3 b_2 b_1 b_0,$$

результат отримується в зворотньому коді, а в розряді переносу P_{i+1} одержується **0**.

Приклад 8. Розробити пристрій двійкового множення на суматорах та логічних елементах І, що дозволяє множити чотирьохрозрядне число A на трьохрозрядне число B .

Розв'язання. Після множення трьохрозрядного і чотирьохрозрядного кодів отримується семирозрядний код. Результат операції утворюється за рахунок паралельного множення множеного на кожен розряд множника і складання проміжних добутків зі зсувом на один розряд. При підрахуванні добутку виконуються наступні дії

$$\begin{array}{r}
 \\
 \\
 \\
 \\
 \times \\
 \\
 \\
 \\
 \hline
 \\
 \\
 \\
 \\
 \\
 \\
 \\
 \hline
 M_7 \quad M_6 \quad M_5 \quad M_4 \quad M_3 \quad M_2 \quad M_1
 \end{array}$$

Тут $M_1 = A_1B_1$; $M_2 = A_1B_2 + A_2B_1$; $M_3 = A_1B_3 + A_2B_2 + A_3B_1$; $M_4 = A_2B_3 + A_3B_2 + A_4B_1$; $M_5 = A_3B_3 + A_4B_2$; $M_6 = A_4B_3$; M_7 – біт переносу із попереднього розряду; знак “+” – знак арифметичного додавання.

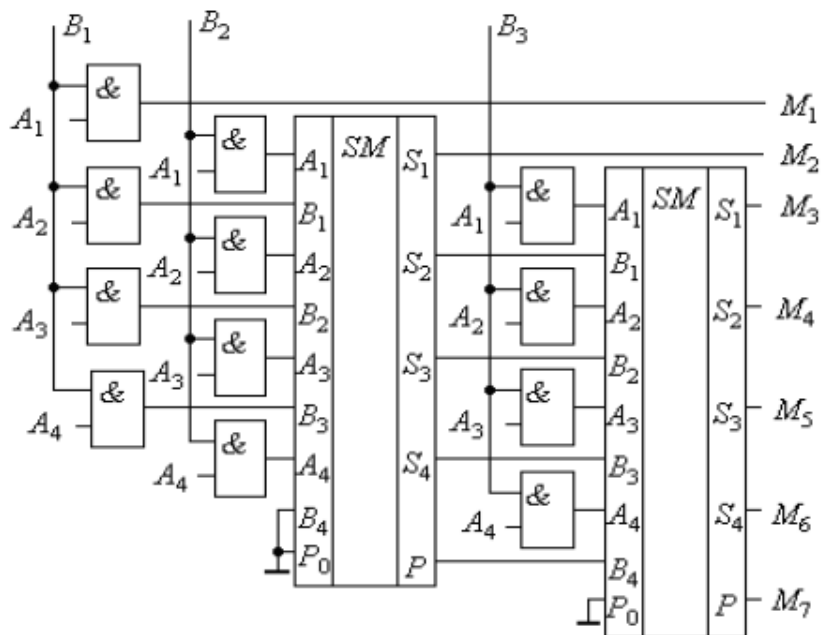


Рис. 7.16

Однорозрядне множення забезпечується елементами І. Суми можуть бути реалізовані суматорами К1533ИМЗ. Для реалізації цієї операції на рис. 7.16 приведена принципова схема.

Приклад 9. Розробити принципову схему суматора, який здійснює додавання однорозрядних десяткових чисел.

Розв'язання. Для додавання десяткових чисел необхідно використати їх двійково-десяткову форму представлення. Це дозволяє для виконання операції додавання використати чотириохрозрядний суматор двох чисел, наприклад, мікросхему К1533ИМЗ.

Відомо, що у тому випадку, коли в результаті додавання отримується результат $S > 9$, значення молодшого напівбайта є невірним. Для отримання вірного результату необхідно його відкоригувати. Корегування виконуємо шляхом додавання до результату числа $b_{10} = 0110_2$. Цю операцію в системі двійково-десяткового числення можна сформулювати наступним чином. При появі в результаті додавання двійкових кодів $1010_2 = 10_{10}$, $1011_2 = 11_{10}$, $1100_2 = 12_{10}$, $1101_2 = 13_{10}$, $1110_2 = 14_{10}$, $1111_2 = 15_{10}$ до отриманого результату необхідно додати число b_{10} . Якщо ці коди не з'являються, то додавання числа b_{10} не робити.

Зрозуміло, що розроблювана схема повинна мати дві мікросхеми К1533ИМЗ. Перша з них виконуватиме операцію додавання, а друга, за необхідності, виконувати корегування результату. Корегування результату виконується згідно з логічною функцією:

$$y = \vee 10, 11, 12, 13, 14, 15 .$$

Представляючи її в формі карти Карно й мінімізуючи, знаходимо:

$$y = x_2 x_3 + x_1 x_3 = x_3 \cdot (x_1 + x_2) .$$

При додаванні чисел, результат яких перевищує число 15_{10} , в сумі виникає перенос P в п'ятий – старший – розряд, який також необхідно враховувати при використанні десяткової корекції. Тому функція y керування десятковою корекцією набуде вигляду:

$$y = P + x_3 \cdot (x_1 + x_2) .$$

Отриманий результат дозволяє побудувати принципову схему суматора (рис. 7.17).

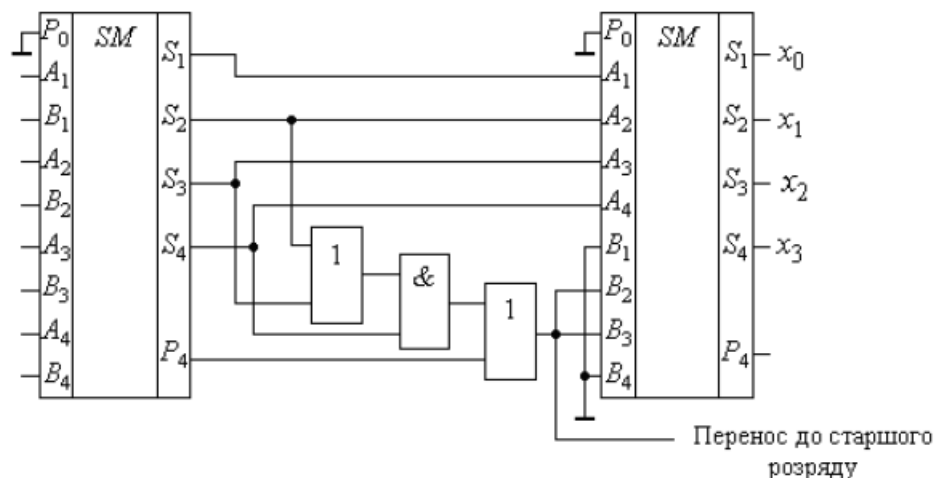


Рис. 7.17

У зв'язку з затримкою переносу, як відмічалось раніше, швидкодія суматора К1533ИМ3 обмежена. Для підвищення швидкодії в ряді серій мікросхем виготовляють спеціальні вузли, які називаються *пристроями прискореного переносу*. Принцип прискореного переносу полягає у тому, що для кожного двійкового розряду знаходять два допоміжних логічних сигнали: $g_i = a_i b_i$ (сигнал генерації переносу) та $h_i = a_i + b_i$ (сигнал розповсюдження).

Сигнал $g_i = 1$, коли складові i -го розряду такі, що перенос в старший розряд $P_{i+1} = 1$ незалежно від значення вхідного переносу P_{i-1} .

Сигнал h_i іноді називають *сигналом прозорості*, тому що відповідно до його визначення при $h_i = 1$ сигнал переносу з попереднього розряду p_i вільно пройде наступний розряд P_{i+1} . За допомогою g_i та h_i можна представити роботу вузла переносу одного розряду суматора $P_{i+1} = g_i + p_i h_i$.

Будуючи структуру зв'язків між переносами, починаючи зі старшого розряду до нульового, можна створити логічну схему, яка дасть можливість організувати паралельний суматор. Серед мікросхем ТТЛ паралельний перенос має суматор К1533ИМ6, який має затримку розповсюдження сигналу

t_3 від входів до виходу P_{i+1} приблизно у 2 рази меншу, ніж К1533ИМЗ. В мікросхемах КМОН суматори також виконуються з вузлом паралельного переносу.

Для прискорення переносу в суматорах з великою кількістю розрядів використовується принцип групового переносу. Суматор розділяється на групи, які мають невелику розрядність і входи-виходи переносу. Сутність групового переносу полягає в тому, що, допоміжно до вузла переносу, в групі будують вузол переносу між групами більш високого рівня, який формує сигнали групових переносів, котрі подаються на входи міні-суматорів. Паралельний перенос між групами суттєво прискорює роботу багаторозрядних суматорів.

2.2. Перемножувачі

Ідеологія перемноження двох бінарних слів полягає у використанні операцій додавання і зсуву проміжної суми.

Реальні чотирьохрозрядні перемножувачі використовують просту технологію, відповідно до якої для двох слів A і B , які необхідно перемножити, створюється таблиця істинності з вихідним словом C подвійної довжини. Кожен розряд c_i слова C є логічною функцією з логічними змінними слів A і B . Тому реалізація чотирьохрозрядного перемножувача є простою реалізацією восьми логічних функцій.

При необхідності реалізувати перемножувач двох восьмирозрядних слів кожне з них розбивається на групи, по 4 біти кожна, і з кожною з груп виконуються операції, як з однією змінною, за принципом знаходження проміжної суми з послідовним виконання операції зсуву.

На базі чотирьохрозрядних перемножувачів і з використанням такої структури будується перемножувач восьмирозрядних слів.

2.3. Компаратори

Компаратори – це пристрої, що виконують порівняння двох чисел.

Мікросхеми компараторів визначають не тільки рівність, але й нерівність двох чисел. Для цього мікросхема має три виходи: « $A > B$ », « $A < B$ » і « $A = B$ », на яких у залежності від співвідношення величин $A = a_3 a_2 a_1 a_0$, $B = b_3 b_2 b_1 b_0$ з'являється активний рівень сигналу.

Логічна функція, що реалізує алгоритм роботи компаратора для одного розряду:

$$y_i = a_i \cdot b_i + \overline{a_i} \cdot \overline{b_i} = \overline{a_i \oplus b_i}. \quad (7.2)$$

Для багаторозрядних чисел функція (7.2) повинна виконуватись для кожного розряду. Нерівність $A > B$ має місце тоді, коли в числі A одиниці дорівнює більш старший розряд, ніж у числі B . Для чотирьохрозрядного компаратора така нерівність може бути записана у вигляді:

$$(A > B) = a_3 \overline{b_3} + \overline{(a_3 \oplus b_3)} \cdot a_2 \overline{b_2} + \overline{(a_3 \oplus b_3)} \cdot \overline{(a_2 \oplus b_2)} \cdot a_1 \overline{b_1} + \\ + \overline{(a_3 \oplus b_3)} \cdot \overline{(a_2 \oplus b_2)} \cdot \overline{(a_1 \oplus b_1)} \cdot a_0 \overline{b_0}.$$

Побудова багаторозрядного компаратора можлива на базі суматора. Його реалізація можлива при виконанні операції віднімання.

Дійсно, в відповідності до законів арифметики при $A = B$ на виходах всіх розрядів суматора s_0, s_1, s_2, s_3 буде 0 при $P_{i+1} = 1$.

При $A > B$ значення 1 буде як на виході переносу $P_{i+1} = 1$, так і хоча б на одному виході s_i . Тому ознакою $A > B$ може бути функція

$$y_{>} = P_{i+1} \cdot \left(\sum s_i \right).$$

При $A < B$ результат наявності переносу $P_{i+1} = 0$.

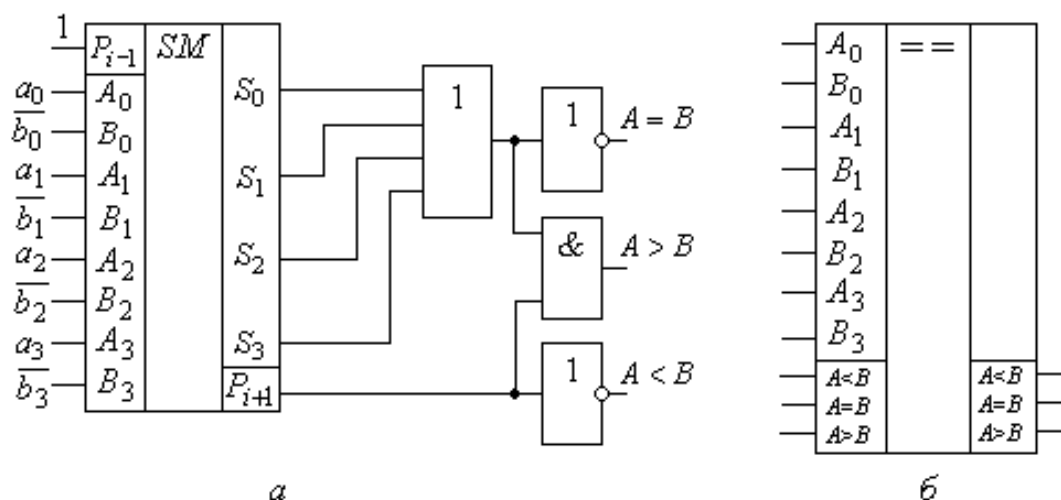


Рис. 7.18

Відповідно до сказаного, логічна схема компаратора на базі суматора К1533ИМЗ (зарубіжні аналоги – 7483РС, CDB483Е, МС7483N, N7483N, SN7483AN, UCY7483N) має вигляд, приведений на рис. 7.18, а.

Прикладом серійних мікросхем є компаратор К564ИП2 (зарубіжні аналоги – CD4585BD, CD4585BH, CD4585BK, CD4585BKF3) (КМОН) (рис 7.18, б); К555СП1 (зарубіжні аналоги – 74LS85N, DL083D, SN74LS85N, UCY74LS85N) (ТТЛ).

Мікросхеми мають розширення по входу (входи « $A > B$ », « $A < B$ » і « $A = B$ »), що дає можливість як послідовного нарощування, так і паралельного.

На рис. 7.19 приведена схема послідовного нарощування двох компараторів К561ИП2.

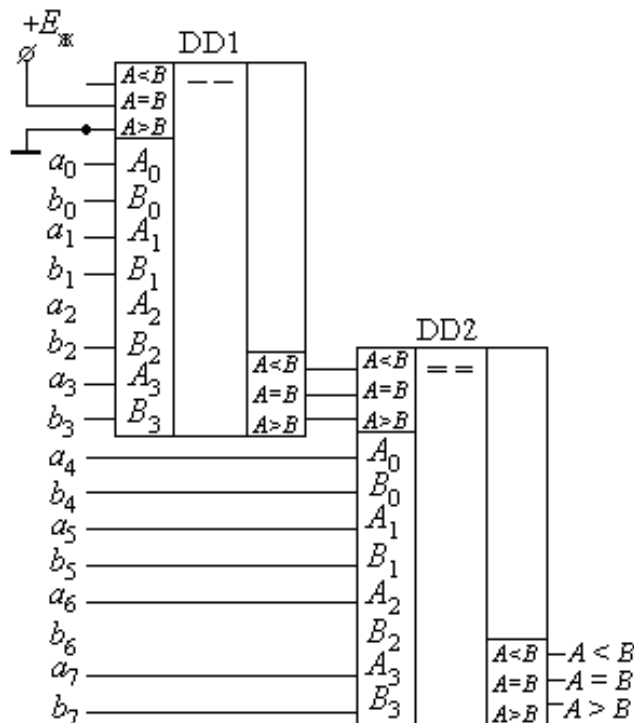


Рис. 7.19

Входи розширення першої мікросхеми необхідно з'єднати відповідно до рис. 7.19, а виходи з'єднуються безпосередньо з входами наступної мікросхеми. При такому способі нарощування компараторів затримки додаються, а стан виходів другого і послідовних компараторів залежить не тільки від побітних значень порівнюваних слів, а й від результату порівняння молодших розрядів цих слів. Тому значення виходів кожного з таких компараторів можуть бути описані рівняннями:

$$(A > B)_{\text{вих}} = (A > B)_{n-m} + (A = B)_{n-m} \cdot (A > B)_{\text{вх}} ;$$

$$(A = B)_{\text{вих}} = (A = B)_{n-m} \cdot (A = B)_{\text{вх}} ;$$

$$(A < B)_{\text{вих}} = (A < B)_{n-m} + (A = B)_{n-m} \cdot (A < B)_{\text{вх}} ;$$

де n, t – діапазон розрядів порівнюваних слів, які приєднані до компаратора.

При пірамідальному способі з'єднання тривалість затримок зменшується.

Слід зазначити, що при нарощуванні мікросхем компараторів необхідно уважно використовувати таблиці станів мікросхем, які відрізняються за характером сигналів на входах нарощування навіть для близьких мікросхем – наприклад, К561ИП2 (зарубіжні аналоги – CD4585BD, CD4585BE, CD4585BK) і К564ИП2.

Компаратори широко використовуються в інформаційних системах для виділення необхідного слова в потоці цифрової інформації, для відмітки часу в часових пристроях, для виконання умовних переходів в обчислювальних пристроях. У пристроях автоматики компаратори використовуються для контролю виходу величин за межі допусків і т.п.

2.4. Арифметично-логічні пристрої

Арифметично-логічні пристрої (АЛП, ALU – Arithmetic-Logic Unit) – це спеціалізовані мікросхеми, які виконують арифметичні та логічні операції у відповідності до двійкового коду, який подається на керуючі входи мікросхеми. Умовне позначення АЛП в електронних схемах приведене на рис. 7.20.

Мікросхеми АЛП, що належать до різних серій, – наприклад, ТТЛ (К1533ИП3 (зарубіжні аналоги – 74181PC, SN74181N)), КМОН (К564ИП3 (зарубіжні аналоги – 4581BDC, 4581BDM, 4581BFC, GD4581BC, GD4581BDC, HD14581B, MC14581BAL, MC14581BCL, MC14581BCPDS, SCL4581B)), – за своїми функціональними властивостями подібні. Вони призначені для роботи з двома чотирьохрозрядними двійковими словами. Конкретний вид операції задається п'ятирозрядним керуючим кодом, що подається на входи M_0 , M_1 , M_2 , M_3 , M_4 . Як результат, цей пристрій може виконувати 32 різноманітні команди. Вхід M_4 задає режим АЛП на виконання 16 логічних ($M_4 = 1$) або 16 арифметичних ($M_4 = 0$) операцій. Допоміжні входи **G** та **H** призначені для

використання АЛП в операціях над багаторозрядними словами при одночасному використанні спеціально для цієї мети призначеної мікросхеми К1533ИП4 (зарубіжні аналоги – DM74182N, FLH411, MC74182N, SN74182N, SN84182N, TL84182N, УСУ74182N) і К564ИП4 (4582BDC, 4582BDM, 4582BFC, GD4582BC, GD4582BDC, HD14582B, MC14582BAL, MC14582BCL, MC14582BCP, SCL4582B, SCL4582BC). АЛП призначений для роботи з сигналами як позитивної, так і негативної логіки, що відображається у таблиці станів мікросхеми (Табл. 7.6).

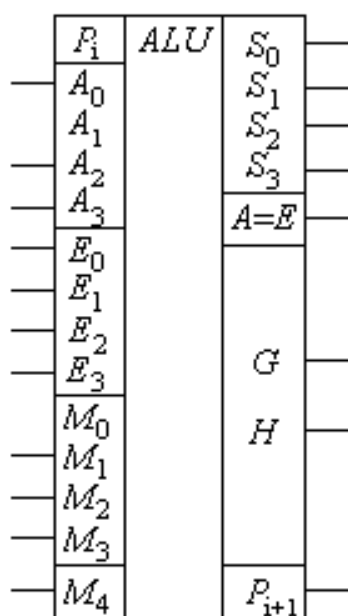


Рис. 7.20

У Табл. 7.6 L та H позначені відповідно низький і високий рівні сигналів, незалежно від типу логіки. Знаки (+) та (–) – знаки арифметичних операцій. Результати арифметичних дій виражені в доповнюючому коді. Так як між зворотнім та доповнюючими кодами існує однозначний зв’язок, то скрізь, де вказана арифметична дія “–1”, результат на виходах зчитується у зворотному коді.

Табл. 7.6

Входи вибору функції				Вхід-вихід (негативна логіка)		Вхід-вихід (позитивна логіка)	
M_3	M_2	M_1	M_0	Логічні функції $M_4 = H$	Арифметичні дії $M_4 = L;$ $P_i = L$	Логічні функції $M_4 = H$	Арифметичні дії $M_4 = L$ $P_i = H$
L	L	L	L	\bar{A}	$A - I$	\bar{A}	A
L	L	L	H	$\bar{A} \vee \bar{E}$	$AE - I$	$\bar{A}\bar{E}$	$A \vee E$
L	L	H	L	$\bar{A} \vee E$	$A\bar{E} - 1$	$\bar{A}E$	$A \vee \bar{E}$
L	L	H	H	I	$-I$	0	$-I$
L	H	L	L	$\bar{A}\bar{E}$	$A + (A \vee \bar{E})$	$\bar{A}E$	$A + A\bar{E}$
L	H	L	H	\bar{E}	$A + (A \vee \bar{E})$	\bar{E}	$(A \vee E) + A\bar{E}$
L	H	H	L	$\bar{A}(\cdot)\bar{E}$	$A - E - I$	$A(+)\bar{E}$	$A - E - I$
L	H	H	H	$A \vee \bar{E}$	$A \vee \bar{E}$	$A\bar{E}$	$A\bar{E} - 1$
H	L	L	L	$\bar{A}E$	$A + (A \vee E)$	$\bar{A} \vee E$	$A + AE$
H	L	L	H	$A(+)\bar{E}$	$A + E$	$\bar{A}(\cdot)\bar{E}$	$A + E$
H	L	H	L	E	$A\bar{E} + (A \vee E)$	E	$(A \vee \bar{E}) + AE$
H	L	H	H	$A \vee \bar{A}\bar{E}$	AE	AE	$AE - I$
H	H	L	L	0	$A + A$	I	$A + A$
H	H	L	H	$A\bar{E}$	$AE + A$	$A \vee \bar{E}$	$(A \vee E) + A$
H	H	H	L	AE	$A\bar{E} + A$	$A \vee \bar{E}$	$(A \vee \bar{E}) + A$
H	H	H	H	A	A	A	$A - I$

На відміну від суматорів, сигнал переносу на вході і виході є інверсним по відношенню до сигналів на входах A та E . Сигнали переносу діють лише при виконанні арифметичних операцій. При використанні логічних операцій АЛП всі зазначені в Табл. 7.6 функції виконує порозрядно. Функція переносу

діє тільки при виконанні арифметичних дій, але в результатах цих операцій символи AE означають операцію **I**.

Якщо АЛП використовується в якості компаратора (операція $A \vee \bar{E}$), то при $A = E$ на відповідному виході формується сигнал високого рівня. При $A \neq E$ відповідний сигнал формується на виході P_{i+1} .

Не вдаючись у подробиці роботи АЛП, слід зазначити, що в зв'язку з інтенсивним розвитком мікропроцесорної техніки розглянута схема АЛП у реальній схемотехніці майже не використовується. Її вивчення має скоріш методичне значення, адже дає можливість практично вивчати ідеологію побудови мікропрограмних автоматів і мікропроцесорів.

3. Пристрої контролю

3.1. Контроль парності

Незважаючи на високу надійність та перешкодостійкість цифрових систем передачі інформації, вірогідність виникнення похибок завжди існує. Тому всі високонадійні канали передачі інформації забезпечуються допоміжними схемами, які дають можливість впевнитись в відсутності похибок. У будь-якій структурі каналу передачі інформації з контролем похибок повинна бути надмірність каналу. Наприклад, якщо передається код $0101_2 = 5_{10}$ і на виході каналу з'являється $1101_2 = 13_{10}$, то в загальному плані похибку без спеціальних перевірок визначити неможливо. Але, якщо ми знаємо, що інформація передається у двійково-десятковому коді, то одержаний результат хибний. Тобто, наявність шести надлишкових станів дає можливість виявити деякі похибки.

Простий спосіб визначення похибок в словах, які передаються, базується на припущенні, що найбільша вірогідність збою можлива тільки в одному

біті, тобто при появі помилкової одиниці або нуля. Тому для визначення наявності такого збою використовують *контроль парності* або *контроль непарності* одиниць у переданому слові (*Parity Check*). В основі цього способу лежить операція знаходження суми за модулем **2** всіх двійкових розрядів контрольованого слова. При парній кількості одиниць вказана сума дорівнює **0**, а при непарній – **1**. Прикладом побудови такого пристрою контролю є схема, що приведена на рис. 7.21.

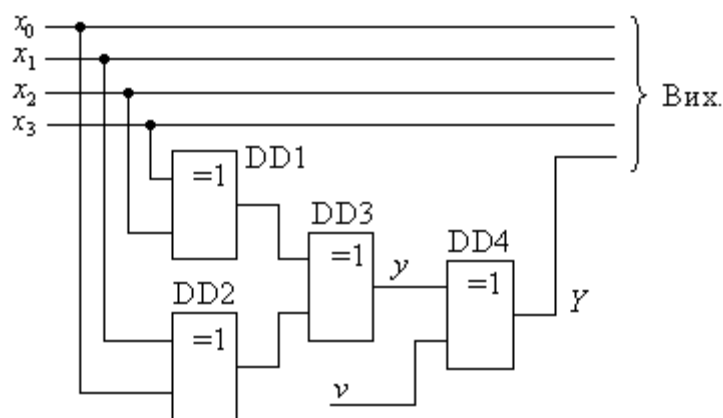


Рис. 7.21

Таблиця 7.7

x_3	x_2	x_1	x_0	$Y (v=0)$	$Y (v=1)$
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	0	1

Сигнал на виході Y визначається наступною логічною функцією:

$$Y = x_0 \oplus x_1 \oplus x_2 \oplus x_3 \oplus v = y \oplus v.$$

З аналізу роботи пристрою, схема якого приведена на рис. 7.21, витікає, що при парній кількості одиниць у слові $x_3 x_2 x_1 x_0$ значення виходу $y = 0$. При непарній – $y = 1$. Тому значення Y залежить від сигналу v , який задає режим контролю – контроль парності або контроль непарності.

При контролі парності задається $v = 0$. Якщо відбудеться контроль за непарною кількістю одиниць, то задається $v = 1$ (див. Табл. 7.7).

Вихід Y називається *контрольним бітом*. При прийомі інформації одержане слово знов перевіряється на парність або непарність. Якщо при контролі парності в приймальному контрольному пристрої з'явилась **1**, то це значить, що або в інформаційній шині, або в контрольному біті з'явилась похибка. Як бачимо, такий простий контроль не дає можливості виправити похибку, але він дає можливість визначитись з хибною інформацією, щоб потім або використати її, або відкоригувати.

У практиці контролю переважно використовують спосіб контролю по непарності. Пов'язано це з тим, що при появі слова з усіма нулями при контролі по парності цю ситуацію неможливо буде відрізнити від обриву лінії зв'язку.

Приведена схема не може визначити подвійної похибки і будь-якої кількості парних похибок. Пояснюється це лише малою надмірністю лінії зв'язку. Тому для більш глибокого контролю необхідно мати і більшу її надмірність. Оскільки цифрові лінії зв'язку дуже надійні і ймовірність появи двох незалежних похибок $p^2 = p_1^2$, тому навіть такий простий контроль суттєво піднімає надійність передачі інформації. Завдяки цьому цей вид контролю використовується практично в усіх каналах передачі цифрових даних, а також при роботі процесора з запам'ятовуючими пристроями, контролюючи тим самим і пристрої пам'яті.

Пристрої контролю парності/непарності двійкових слів випускаються у вигляді самостійних мікросхем у різних серіях (КМОН – К561СА1; ТТЛ – К1533ИП2, К531ИП5).

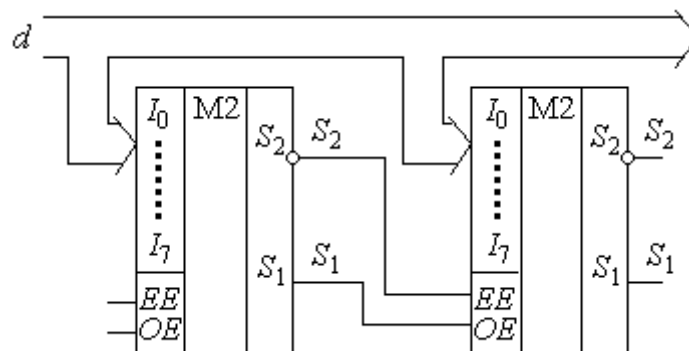


Рис. 7.22

Таблиця 7.8

Число одиниць на $I_0 \div I_7$	EE	OE	S_1	S_2
Парне	1	0	1	0
Непарне	1	0	0	1
Парне	0	1	0	1
Непарне	0	1	1	0
X	1	1	0	0
X	0	0	1	1

Мікросхема К1533ИП2 (рис. 7.22) (зарубіжними аналогами якої є мікросхеми 74180РС, СDB4180Е, SN74180N, УСУ74180N) має вісім інформаційних входів $I_0 \div I_7$ та два входи OE та EE , які дозволяють в одній схемі забезпечити різні режими роботи: керування полярністю вихідних сигналів, нарощування розрядності схем контролю, забезпечення контролю як по парності, так і по непарності одиниць.

Виходи S_1 , S_2 взаємно інверсні. Вихід S_1 є виходом непарності. Взаємозв'язок між значеннями вхідних та вихідних сигналів приводиться в Табл. 7.8.

Табл. 7.8 надає можливість визначитись з необхідними вхідними сигналами на входах **EE** і **OE** для забезпечення контролю по парності або непарності.

Наприклад, при вхідних сигналах, відповідних $EE = 1$; $OE = 0$, парність коду на входах $I_0 \div I_7$ відобразатиметься високим рівнем виході S_1 та низьким на S_2 . Якщо ж на входах $I_0 \div I_7$ код непарний, то на виходах S_1 , S_2 буде код, інверсний попередньому.

Слід звернути увагу на те, що при парності коду на входах $I_0 \div I_7$ маємо наступні логічні співвідношення між вхідними і вихідними сигналами:

$$S_1 = EE ; S_2 = OE ;$$

при непарності коду:

$$S_1 = \overline{EE} ; S_2 = \overline{OE} .$$

Нарощування довжини слова, що підлягає контролю, забезпечується послідовним з'єднанням мікросхем, причому виходи S_1 та S_2 попередньої схеми з'єднані з входами **EE** та **OE** наступної.

3.2. Контроль логічних перетворень

Розглянуті в попередніх параграфах засоби контролю правильності передачі кодів виконують контроль лише незмінних даних. Якщо коди піддаються логічним перетворенням, то засоби, контролюючі їх правильність,

стають дуже складними. Найпростіший спосіб контролю – це дублювання логічних перетворень та їх контроль (рис. 7.23).

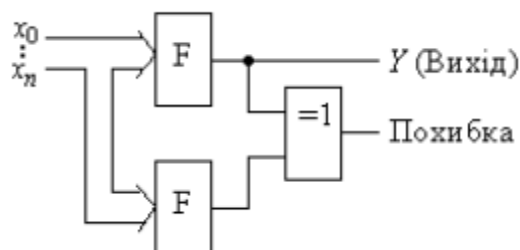


Рис. 7.23.

Відповідно до рисунку, логічна схема F дублюється аналогічною схемою. Це дає можливість забезпечувати безперервний контроль вихідного потоку інформації і у разі їх неспівпадання видавати сигнал помилки. При такій формі контролю слід враховувати той факт, що збої в роботі цифрової схемотехніки можуть з'являтися не тільки за рахунок дефектів схеми, але й при наявності завад по живленню. Тому при однакових цифрових схемах велика ймовірність появи однакових збоїв. Це значить, що дублюючі пристрої необхідно виготовляти на різних схемах. Один з таких варіантів побудови дублюючих схем – це використання взаємно протилежних логічних схем відповідно до теореми де Моргана.

У ряді випадків для передачі контрольованої інформації використовуються *парафазні коди*, коли дублюється канал передачі інформації, і по другому каналу завжди передається сигнал, інверсний сигналу першого каналу; якщо ці сигнали в кожний тактовий момент скласти за модулем 2 , то нульовий результат буде завжди говорити про наявність помилки в одному з каналів.

Комбінаційні схеми, що реалізують систему булевих функцій, на відміну від одновихідних схем, можливо контролювати, не використовуючи їх дублювання. Прикладом схеми з таким контролем приводиться на рис. 7.24.

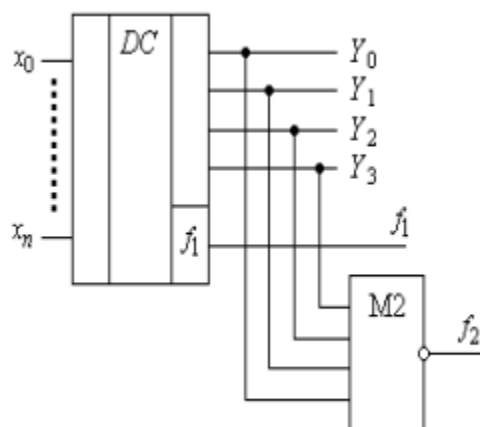


Рис. 7.24

Для цього при проектуванні кодового перетворювача необхідно закласти реалізацію ним допоміжної функції f_1 , яка задається як функція аргументів, значення якої співпадають із значеннями суми по модулю 2 решти функцій. Протифазна функція f_2 є схемою контролю парності всіх функціональних виходів перетворювача, окрім f_1 . Одиночна похибка в роботі як схеми перетворювача, так і схеми **M2** змінить значення парності або f_1 , або f_2 і буде зафіксована схемою контролю. Функція f_1 повинна синтезуватись з використанням вхідних змінних $x_0 \div x_n$. Якщо ж використати для її побудови вихідні значення $Y_0 \div Y_3$, то можливі похибки можуть змінити одночасно як f_1 , так і f_2 і не будуть виявлені. Вказану особливість необхідно враховувати при розробці схем контролю більш складних перетворювачів, в яких декілька різних виходів можуть мати загальні схеми і, відповідно, загальні збої, які потім не можуть бути виявлені схемами типу **M2**.

4. Буферні елементи

При побудові складних цифрових і мікропроцесорних систем створюється шинна форма обміну інформацією між окремими пристроями. *Шина* – це паралельна група провідників, по якій передаються інформаційні сигнали в обох напрямках і до якої може бути приєднана досить велика група пристроїв, що передають та приймають інформаційні сигнали. В загальному плані в залежності від характеру сигналів, що передаються і приймаються, шини класифікуються на *адресні, командні та шини даних*. По адресних шинах йдуть сигнали звернення до тих чи інших пристроїв, що приєднані до шин; по шині керування передаються керуючі сигнали, а по шині даних – безпосередньо дані.

Оскільки кожен з пристроїв повинен мати право доступу до шин, то для забезпечення необхідного порядку при роботі з шинами існує велика кількість мікросхем, які спеціально призначені для обслуговування різноманітних пристроїв при їх роботі з шинами. Такі мікросхеми здебільшого не виконують логічних функцій. Вони призначені для:

- формування цифрових сигналів для забезпечення необхідного струму, напруги, потужності;
- забезпечення тимчасової передачі шин від одних пристроїв до інших;
- розділення в часі інформації, що передається в обох напрямках;
- підключення (чи відключення) до шин необхідних пристроїв;
- виконання допоміжних функцій – таких, як тимчасове запам'ятовування інформації, що передається по шині, її інверсія і т. п.

Всі вказані функції забезпечуються спеціальними *буферними елементами*. Основою будь-якого буферного елемента є підсилювач. Оскільки по шинах інформація передається в обох напрямках, то підсилювач

повинен бути двонаправленим. Через те, що двонаправлений підсилювач буде збуджуватись, то він виготовляється у вигляді двох ідентичних, зустрічно-паралельно з'єднаних підсилювачів потужності, кожен з яких вмикається лише при передачі інформації від його входу до виходу. Для забезпечення такої комутації підсилювачів використовується спеціальна логіка. Кількість пар підсилювачів відповідає кількості провідників шини. Якщо, наприклад, шина даних має розрядність 8 біт, то мікросхема шинного підсилювача повинна мати 8 пар зустрічно-паралельно з'єднаних підсилювачів, керованих єдиним логічним вузлом.

Для забезпечення необхідної потужності використовуються різні схемотехнічні прийоми, що розглядались раніше.

У мікросхемах обслуговування широко використовуються логічні елементи з Z -станом. Переведення мікросхеми у Z -стан дає можливість тимчасово відділити шини від цифрових пристроїв, внаслідок чого ними можуть користуватися інші пристрої.

У технічній літературі буферні елементи часто називають "*шинними драйверами*". До них відносяться мікросхеми К1533АП3 (SN74ALS240), К1533АП4 (SN74ALS241), а також мікросхеми груп ЛН, ЛП, ХЛ.