

Лекція 29

Вузли обчислювальних пристроїв

Мета лекції: вивчення побудови основних обчислювальних пристроїв та збільшення їх розрядності.

План лекції:

- 29.1 Компаратори цифрових сигналів.
- 29.2 Суматори.
 - 29.2.1 Алгоритм двійкового додавання.
 - 29.2.2 Класифікація суматорів.
 - 29.2.3 Двійковий напівсуматор.
 - 29.2.4 Однорозрядний суматор.
 - 29.2.5 Багаторозрядний суматор паралельної дії.
- 29.3 Алгоритм віднімання двійкових чисел.
- 29.4 Структурна схема віднімання і додавання двох чисел.
- 29.5 Інтегральна схема арифметико-логічного пристрою.
- 29.6 Контрольно-навчальний тест до лекції 29.

29.1 Компаратори цифрових сигналів

Цифровим компаратором, називається комбінаційний логічний пристрій, призначений для порівняння чисел, представлених у вигляді двійкових кодів.

Число входів компаратора визначається розрядністю порівнюваних кодів. На виході компаратора зазвичай формується три сигнали:

$F_=$ — рівність кодів,

$F_>$ — якщо числовий еквівалент першого коду більше другого

$F_<$ — якщо числовий еквівалент першого коду менше другого.

Роботу компаратора при порівнянні двох однорозрядних кодів пояснює таблиця істинності (табл. 29.1).

Аналіз таблиці істинності показує, що при будь-якій комбінації вхідних сигналів на виході компаратора може бути сформований тільки один активний (одичний) логічний сигнал. Тому, при будь-якій розрядності вхідних кодів досить, використовуючи вхідні сигнали, сформувати тільки будь-які два з вихідних сигналів. Третій сигнал завжди може бути отриманий по двом відомим.

Система функцій алгебри логіки (ФАЛ), що відповідає наведеній таблиці істинності, має вигляд

$$\begin{aligned}
 F_&= \bar{x}_1 \bar{x}_0 + x_1 x_0 = \overline{x_1 \oplus x_0} = \overline{F_<} \overline{F_>}, \\
 F_&< = \bar{x}_1 x_0 = \overline{F_=} \overline{F_>}, \\
 F_&> = x_1 \bar{x}_0 = \overline{F_=} \overline{F_<}.
 \end{aligned}
 \tag{29.1}$$

Т а б л и ц я 29.1

Таблиця істинності компаратора однорозрядних кодів

x_1	x_0	$F_=$	$F_>$	$F_<$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

Аналізуючи наведені вирази з погляду зменшення апаратних витрат на реалізацію, відзначимо, що, використовуючи вхідні змінні, зручніше було б отримати значення $F_>$ і $F_<$, а $F_=$ — реалізувати як їх функцію. Проте з огляду на те, що вираз для визначення $F_=$ має в цифровій техніці велике самостійне значення, на нім слід зупинитися детальніше. Воно носить назву операції Виключаюче АБО — ІІ або інверсія від суми по модулю два. Приклад реалізації цієї операції з використанням елементів І, АБО і ІІ, а також її умовне графічне позначення наведені на рис. 29.1. На рис. 29.2 дана структурна логічна схема, що відповідає таблиці істинності цифрового компаратора.

На практиці часто доводиться стикатися із завданням порівняння багаторозрядних двійкових кодів. Записавши таблицю істинності, можна синтезувати логічну схему відповідного пристрою. Проте при збільшенні числа вхідних змінних зробити це зважаючи на громіздкість отримуваної таблиці вельми непросто. У даній ситуації зручно скористатися методами так званого блокового конструювання, або декомпозиції завдання. Суть такого підходу полягає в розбитті складного завдання на ряд простіших, рішення яких може бути виконане доступними засобами. Далі, з використанням отриманих результатів, проводиться рішення початкової задачі. Проілюструємо даний підхід на прикладі побудови цифрового компаратора багаторозрядних двійкових кодів. За основу приймемо схему компаратора однорозрядних двійкових слів. Очевидно, що результат порівняння 2-розрядних двійкових слів можна записати через результати порівняння однорозрядних слів. Відповідна система (ФАЛ) в цьому випадку матиме вигляд

$$F_{=} = F_{1=} F_{0=},$$

$$F_{>} = F_{1>} + F_{1=} F_{0>},$$

$$F_{<} = \overline{F_{=} + F_{>}}.$$

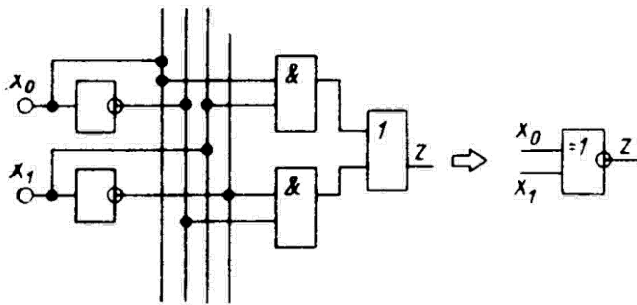


Рисунок 29.1 - Схемна реалізація операції Виключаюче АБО-НІ

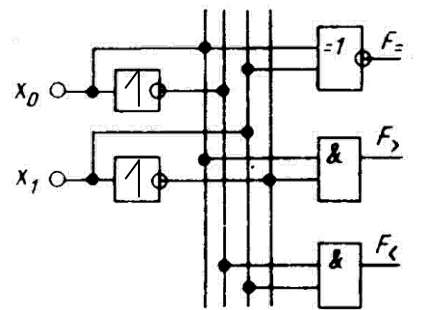


Рисунок 29.2 - Логічна схема компаратора

Аналогічно для 3-розрядних кодів отримаємо

$$F_{=} = F_{2=} F_{1=} F_{0=},$$

$$F_{>} = F_{2>} + F_{2=} F_{1>} + F_{2=} F_{1=} F_{0>},$$

$$F_{<} = \overline{F_{=} + F_{>}}.$$

У загальному випадку для n-розрядних двійкових кодів можна записати

$$F_{=} = F_{n-1=} F_{n-2=} \dots F_{0=},$$

$$F_{>} = F_{n-1>} + F_{n-1=} F_{n-2>} + \dots + F_{n-1=} F_{n-2=} \dots F_{1=} F_{0>},$$

$$F_{<} = \overline{F_{=} + F_{>}}.$$
(29.2)

Таким чином, з використанням цифрових компараторів, що мають обмежену розрядність вхідних слів, на підставі системи ФАЛ (29.2) завжди можна побудувати пристрій необхідної розрядності.

29.2 Суматори

Суматором називається комбінаційний логічний пристрій, призначений для виконання операції арифметичного додавання чисел, представлених у вигляді двійкових кодів.

Суматори є одним із основних вузлів арифметико-логічного пристрою. Термін суматор охоплює широкий спектр пристроїв, починаючи з простих логічних схем, до складних цифрових вузлів. Загальним для всіх цих пристроїв є арифметичне додавання чисел, представлених в двійковій формі. Розглянемо детальніше деякі конкретні. схемотехнічні рішення, призначені для реалізації поставленого завдання.

29.2.1 Алгоритм двійкового додавання

Спершу отримаємо ФАЛ, що описує операції арифметичного додавання двох однорозрядних двійкових кодів x_1 і x_0 . Алгоритм її виконання пояснюється таблицею істинності (табл. 29.2). У графі s наведено значення результату додавання (суми), а в графі p — отримане при цьому значення перенесення в старший розряд. Слід звернути увагу на відмінності результатів, що отримуються при арифметичному і логічному складанні. При логічному додаванні в останньому рядку стовпця s було б присутнє значення 1. Ця відмінність результатів даних операцій не дозволяє застосувати для арифметичного підсумовування елемент АБО, а вимагає розробки спеціалізованого пристрою.

Значення сигналу перенесення, рівного одиниці в останньому рядку таблиці 29.2 говорить про те, що результат, отриманий при виконанні операції арифметичного складання, в цьому випадку не може бути представлений двійковим кодом, розрядність якого рівна розрядності слів доданків.

Таблиця 29.2
Таблиця істинності складання двох однорозрядних двійкових кодів

x_1	x_0	s	p
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Таблиця 29.3
Таблиця істинності складання багатьох розрядів багатозрядних двійкових кодів

x_1	x_0	P_{-1}	s	p
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Використовуючи наведену таблицю, легко записати систему ФАЛ, що описують алгоритм операції арифметичного складання

$$s = \bar{x}_1 x_0 + x_1 \bar{x}_0, \quad (29.3)$$

$$p = x_1 x_0. \quad (29.4)$$

Функція, що описується виразом (29.3), дуже часто зустрічаються при розробці цифрових пристроїв. Її називають функцією Виключаюче АБО, або *сумою по модулю два*. Таким чином, для підсумовування двох двійкових однорозрядних кодів необхідно виконати логічну операцію Виключаюче АБО.

З метою спрощення вираз (29.3) зазвичай записують наступним чином:

$$s = x_1 \oplus x_0. \quad (29.5)$$

Операція $x_1 \oplus x_0$ називається операцією Виключаюче АБО—НІ. Використовуючи вираз (29.5), легко записати

$$\begin{aligned} \bar{s} &= \overline{x_1 \oplus x_0} = \overline{\bar{x}_1 x_0 + x_1 \bar{x}_0} = (\bar{x}_1 x_0)(x_1 \bar{x}_0) = (x_1 + \bar{x}_0)(\bar{x}_1 + x_0) = \\ &= x_1 \bar{x}_1 + x_1 x_0 + \bar{x}_1 \bar{x}_0 + \bar{x}_0 x_0 = \bar{x}_1 \bar{x}_0 + x_1 x_0 \end{aligned} \quad (29.6)$$

Відзначимо, що вираз (29.6) тотожний отриманому при розгляді цифрових компараторів співвідношенню для сигналу $F_ =$ рівності двох кодів (див. вираз 29.1).

Логічні елементи, що виконують операції Виключаюче АБО і Виключаюче АБО-НІ, завжди мають тільки два входи, т.е. операції завжди виконуються тільки над двома змінними.

Таблиця 29.2 застосовна тільки для складання однорозрядних двійкових кодів або молодших розрядів багаторозрядних слів. Таблиця складання старших розрядів багаторозрядних двійкових слів повинна бути доповнена змінною можливого перенесення з більш молодшого розряду (табл. 29.3). ФАЛ, що описують результати складань, в цьому випадку матимуть вигляд

$$s = (x_1 \oplus x_0) \bar{p}_1 + (\overline{x_1 \oplus x_0}) p_{-1}, \quad (29.7)$$

$$p = x_1 x_0 + (x_1 \oplus x_0) p_{-1}. \quad (29.8)$$

З виразу (29.7) маємо, що для отримання суми двох старших розрядів необхідно спочатку виконати операцію Виключаюче АБО над початковими доданками x_1 і x_0 і потім ще одну операцію Виключаюче АБО над результатом першої операції Виключаюче АБО і сигналом перенесення із попереднього розряду. Для отримання

сигналу перенесення також необхідно скористатися результатом операції Виключаюче АБО над доданками x_1 і x_0 .

29.2.2 Класифікація суматорів

Класифікація суматорів може бути виконана по різним ознакам. Розглянемо ті, що найчастіше зустрічаються з них.

За числом виходів розрізняють напівсуматори, однорозрядні суматори, багаторозрядні суматори.

Напівсуматором називається пристрій, що призначений для складання двох однорозрядних кодів, має два входи і два виходи і формує із сигналів вхідних доданків сигнали суми і перенесення в старший розряд.

Однорозрядним суматором називається пристрій, призначений для складання двох однорозрядних кодів, що має три входи і два виходи, і що формує з сигналів вхідних доданків і сигналу перенесення з молодших розрядів сигнали суми і перенесення в старший розряд.

Багаторозрядним суматором називається пристрій, призначений для складання двох багаторозрядних кодів, що формує на виході код суми і сигнал перенесення у випадку, якщо результат складання не може бути представлений кодом, розрядність якого співпадає з розрядністю кодів доданків.

У свою чергу, багаторозрядні суматори поділяються на *послідовні* і *паралельні*. У послідовних суматорах операція складання виконується послідовно розряд за розрядом, починаючи з молодшого. У паралельних всі розряди вхідних кодів підсумовуються одночасно.

Розрізняють *комбінаційні* суматори — пристрої, що не мають власної пам'яті, і *накопичуючі* суматори, забезпечені власною внутрішньою пам'яттю, в якій акумулюються результати виконаної операції. При цьому кожен черговий доданок додається до того значення, що вже знаходилось в пристрої.

За способом тактування розрізняють *синхронні* і *асинхронні* суматори. У *синхронних* суматорах час виконання операції арифметичного підсумовування двох кодів не залежить від виду самих кодів і завжди залишається постійним. У *асинхронних* суматорах час виконання операції залежить від виду доданків. Тому після закінчення виконання підсумовування необхідно формувати спеціальний сигнал завершення операції.

Залежно від використовуваної системи числення розрізняють двійкові, двійково-десяткові і інші типи суматорів.

29.2.3 Двійковий напівсуматор

Згідно визначенню, вихідні сигнали двійкового напівсуматора повинні відповідати системі ФАЛ (29.3) і (29.4). Для її технічної реалізації необхідні логічні

елементи І і Виключаюче АБО. Оскільки раніше елемент Виключаюче АБО не був описаний, розглянемо можливість його побудови на вже відомих елементах. Для цього перетворимо вираз (29.3) до базису І—НЕ

$$s = x_1 \oplus x_2 = \bar{x}_1 x_0 + x_1 \bar{x}_0.$$

Технічна реалізація отриманого виразу приведена на рис. 29.3. На цьому ж малюнку показано умовне позначення елемента Виключаючого АБО.

З використанням сказаного легко можна синтезувати логічну схему двійкового напівсуматора (рис. 29.4,а). Час підсумовування для схеми рис. 29.4 визначається

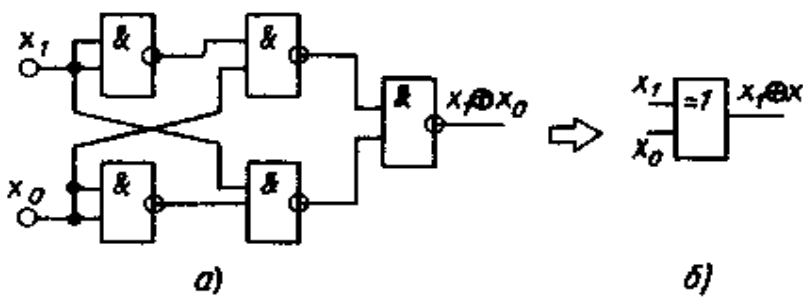


Рисунок 29.3 - Структурна схема реалізації операції Виключаючі АБО (а) та її умовне позначення (б)

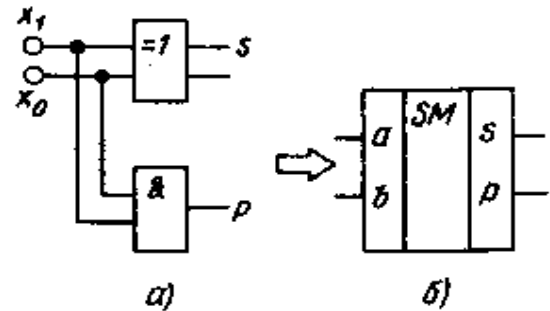


Рисунок 29.4 - Напівсуматор (а) та його умовне позначення (б)

часом виконання операції Виключаюче АБО

$$t_{n \Sigma} = 3t_{3p}, \tag{29.9}$$

де t_{3p} — час затримки розповсюдження для елемента І—НІ.

Якщо регістри, що зберігають доданки, забезпечені як прямими, так і інверсними виходами, вхідні інвертори з схеми рис. 29.3 можна виключити. Тоді загальний час складання зменшиться до $t_{n \Sigma} = 2t_{3p}$. Умовне графічне зображення двійкового напівсуматора показане на рис. 29.4,б.

29.2.4 Однорозрядний суматор

Функціонування однорозрядного суматора визначається системою ФАЛ (29.7), (29.8). Технічна реалізація даної ФАЛ може бути виконана на ЛЕ будь-якого типу. Розглянемо, наприклад, побудову однорозрядного суматора з використанням схем двійкових напівсуматорів (рис. 29.5, а). Очевидно, що для цієї мети необхідно два напівсуматори і елемент АБО.

Слід зазначити, що якщо синтезувати схему однорозрядного суматора безпосередньо за табл. 29.3, відносно елементарних ЛЕ, можна отримати простіше технічне рішення.

Час підсумовування в наведеній схемі також визначається часом виконання операції Виключаюче АБО

$$t_{0 \Sigma} = 2t_{n \Sigma} = 6t_{зр}. \quad (29.10)$$

Формування сигналу перенесення в старший розряд виконується швидше. Для цього необхідний час

$$t_{сп} = 5t_{зр}. \quad (29.11)$$

Умовне графічне позначення однорозрядного суматора наведено на рис. 29.5, б.

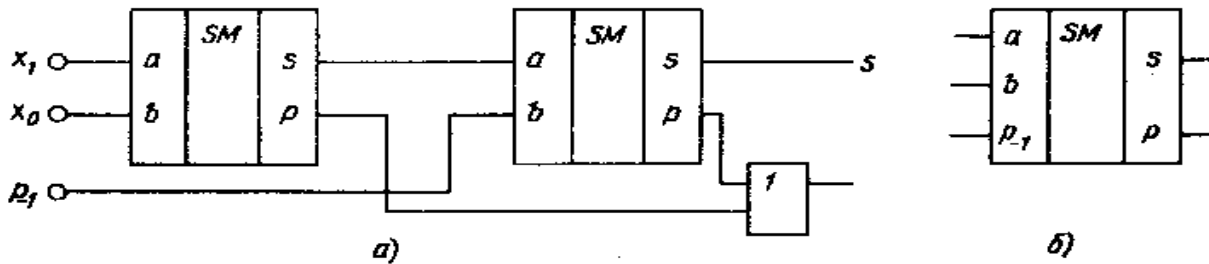


Рисунок 29.5 - Однорозрядний суматор (а) та його умовне позначення (б)

29.2.5 Багаторозрядний суматор паралельної дії

У цьому суматорі, згідно даному раніше визначенню, операції підсумовування повинні виконуватися одночасно для всіх розрядів початкових двійкових чисел. З цього виходить, що такий суматор повинен мати окремі апаратні засоби для виконання підсумовування в кожному розряді.

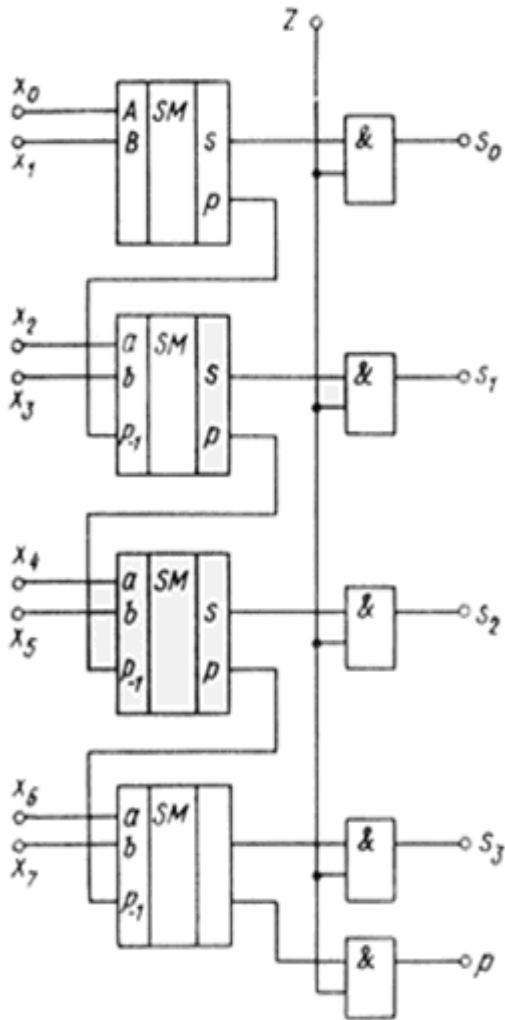


Рисунок. 29.6 - Структурна схема паралельного багаторозрядного суматора з послідовним переносом

Розглянемо наведену на рис. 29.6 типову структуру 4-розрядного суматора, виконаного з використанням трьох однорозрядних суматорів і одного напівсуматора. Розряди кодів доданків подаються на відповідні входи суматорів, виходи суми яких під'єднуються до перших входів ЛЕ І, використовуваних як вихідні ключі, на другі входи яких подається сигнал Z , що визначає момент прочитування результату. Вихід сигналу перенесення суматора нульового розряду подається на вхід перенесення суматора першого розряду і т.д.

Із сказаного виходить, що для отримання на виході сигналу, рівного реальній сумі вхідних кодів, необхідно, щоб сигнал перенесення послідовно сформувався на виходах суматорів всіх розрядів. Отже, не залежно від того, що для підсумовування в кожному розряді використовується окремий суматор, реальний час виконання операції в даній схемі визначається послідовним перенесенням сигналу P з розряду в розряд. Тому результат, який може бути знятий з виходу схеми через час, рівний часу підсумовування в одному розряді, не буде реальним значенням шуканої суми.

Для не допущення помилкового результату на виході схеми встановлені елементи І. Сигнал Z на входах цих елементів повинен з'являтися не раніше, чим після послідовної передачі сигналу перенесення по всіх розрядах суматорів.

Слід зазначити, що реально схеми багаторозрядних суматорів будуються тільки із застосуванням однорозрядних суматорів, що дозволяє, використовуючи їх послідовне включення, збільшити розрядність кодів доданків.

В цьому випадку, з урахуванням виразів (29.9) — (29.11) загальний час формування сигналу результату в даній схемі

$$t_{M\Sigma} = t_{zp} [6 + 2(n-1)], \quad (29.12)$$

де n — розрядність кодів доданків.

Реальний час підсумовування для 4-розрядного суматора даного типу рівний $12t_{zp}$, що в два рази більше часу підсумовування однорозрядного суматора. Вихідний сигнал перенесення, що іменується також сигналом переповнювання, формується за час $t_{МП} = 11t_{zp}$, тобто трохи швидше.

29.3 Алгоритм віднімання двійкових чисел

З математики відомо, що операцію віднімання двох чисел можна замінити операцією складання, якщо в якості від'ємника узяти число, знак якого протилежний початковому. Використання такого підходу дозволяє значно спростити технічну реалізацію арифметико-логічного пристрою АЛП, оскільки для виконання операції віднімання можна використовувати схеми суматорів. При цьому вельми актуальним є питання представлення від'ємних чисел у вигляді двійкових кодів.

На практиці для представлення від'ємних чисел найбільшого поширення набув метод, за яким для позначення знаку використовується старший розряд його двійкового коду. Так, якщо використовується 8-розрядний двійковий код, то сім його розрядів містять інформацію про величину (модулі) числа, а восьмий (старший) — про його знак. Звичайна присутність в цьому розряді нуля означає, що число позитивне, присутність одиниці — що число від'ємне. Отже, використовуючи 8-розрядний код, можна записати числа з десятковими еквівалентами від -127 до $+127$.

Проте присутність знакового розряду безпосередньо не дозволяє використовувати розглянуті раніше схеми суматорів для виконання операції віднімання чисел, представлених в прямому двійковому коді. Найчастіше для запису від'ємного числа використовується додатковий код.

Алгоритм отримання додаткового коду двійкового числа зводиться до наступного:

записують зворотний код початкового числа, для чого всі його розряди інвертують (замінюють доповненнями);

до отриманого після інвертування коду додають одиницю.

Надалі, при складанні отриманого таким чином коду від'ємника з кодом зменшуваного буде реалізована операція віднімання. При цьому знак результату визначається старшим розрядом отриманого коду. Якщо старший розряд рівний нулю, отримано додатне число, представлене в прямому коді. Якщо старший розряд рівний одиниці, отримано від'ємне число, представлене в додатковому коді. Проілюструємо сказане на прикладі.

Приклад 18.1. Використовуючи додатковий код, знайти різницю $31 - 12$.

Рішення. 1. Прямі двійкові коди заданих чисел

$$31_{10} = 00011111_2; \quad 12_{10} = 00001100_2.$$

2. Зворотний код від'ємника 11110011.

3. Додатковий код від'ємника $11110011 + 00000001 = 11110100$.

4. Складання

$$\begin{array}{r} 00011111 \\ + \\ \underline{11110100} \\ 100010011 \end{array}$$

Отриманий результат представлений 9-розрядним двійковим кодом. В цьому випадку самий старший дев'ятий розряд відкидають. Отримане двійкове число містить нуль в старшому розряді. Тому результат позитивний і представлений в прямому коді. Неважко бачити, що його десятковий еквівалент рівний 19.

Приклад 18.2. Використовуючи додатковий код, знайти різницю $12 - 31$.

Рішення. 1. Зворотний код від'ємника 11100000.

2. Додатковий код від'ємника $11100000 + 00000001 = 11100001$.

3. Складання

$$\begin{array}{r} 00001100 \\ + \\ \underline{11100001} \\ 111011001 \end{array}$$

Старший розряд отриманого результату рівний одиниці. Отже, отримано від'ємне число, записане в додатковому коді. Для отримання прямого коду можна скористатися наведеним вище алгоритмом перетворення прямого коду в додатковий. Проте існує простіше правило такого перетворення: додатковий код проглядають справа наліво, залишаючи без зміни всі значення лог. 0, що при цьому зустрілися. Першу лог. 1, що зустрілась, також пропускають, а далі всі цифри інверсують. Результатом такого перетворення буде прямий код числа.

Виконавши описані дії з отриманим кодом, отримаємо код 00010011, що з урахуванням знаку еквівалентно числу -19_{10} .

Слід зазначити, що аналогічно можна виконати і перетворення прямого коду в додатковий. Таким чином, використання при представленні негативних чисел додаткового коду дозволяє застосувати для виконання операції складання і віднімання єдину логічну схему.