

ЗАТВЕРДЖУЮ

Завідувач кафедри _____

В.І. Гостєв

(підпис, ініціали, прізвище)

"__" _____ 20__ року

Сторчак К.П.

(прізвище та ініціали автора)

ЛЕКЦІЯ

з навчальної дисципліни

КЕРУВАЛЬНІ КОМПЛЕКСИ ЕЛЕКТРОЗВ'ЯЗКУ

(назва навчальної дисципліни)

Тема 6: Алгоритми додавання в ЕКМ

(номер і назва теми)

Заняття 10: Алгоритми додавання в ЕКМ

(номер і назва заняття)

Навчальний час – 2 години.

Для студентів інституту (факультету):

Навчально-науковий інститут

Телекомунікацій та інформатизації

Факультет Інформаційних технологій

Навчальна та виховна мета:

Набуття практичних навичок в операціях додавання в цифрових процесорах

Обговорено та схвалено на засіданні кафедри

“__” _____ 20__ року Протокол №__

Зміст

Вступ.

- 1 Алгоритми операції додавання.
- 2 Алгоритм додавання операндів у додатковому коді.
- 3 Алгоритм додавання операндів в оберненому коді.

Л І Т Е Р А Т У Р А

1. Основы арифметики цифровых процессоров: учеб. пособие / Н.П. Вашкевич, Е.И. Калиниченко. - Пенза: Изд-во ПГУ, 2010. - 160 с.

Наочні посібники

(Інфокус, полілюкс, слайди, схеми, макети тощо).

Завдання на самостійну роботу

1. Структура послідовного арифметично-логічного пристрою (АЛП).
2. Структурна схема АЛП для операцій додавання і віднімання.

Вступ

Минулої лекції було розглянуто систему команд, що використовується в ЕКМ. Сьогодні розглянемо операції додавання, що виконуються в цифрових процесорах за допомогою систем команд.

1 Алгоритми операції додавання

Базовими арифметичними операціями в цифровому процесорі є: додавання (віднімання), множення, ділення, при виконанні яких використовуються порозрядні логічні операції, а також операції зміщень.

Існують два основні способи додавання в цифрових процесорах - послідовному і паралельному.

При послідовному способі додавання операндів, що знаходяться в регістрах процесора, виконується по тактах, починаючи з молодшого біта, і кількість тактів дорівнює розрядності процесора.

Розглянемо цей спосіб на прикладі додавання чисел, представлених в десятковій системі числення. Нехай потрібно отримати суму $S = K_1 + K_2$, де $K_1 = 265_{(10)}$; $K_2 = 346_{(10)}$.

Тоді таке додавання буде виконано за три такти:

Такти	1-е слагаемое	2-е слагаемое	Перенос из предыдущего разряда	Сумма	Перенос в следующий разряд
1	5	6	0	1	1
2	6	4	1	1	1
3	2	3	1	6	0

Додавання в одному розряді двійкової системи числення можна представити таблицею 1 (де X_i , Y_i - значення доданків в i -му розряді; C_{i-1} - перенесення з попереднього розряду; S_i - сума в i -му розряді; C_i - перенесення в наступний розряд).

Таблиця 1 - Додавання двох двійкових розрядів

X_i	Y_i	C_{i-1}	S_i	C_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

Розглянемо на прикладі послідовний спосіб додавання чисел X і Y в п'ятирозрядному цифровому процесорі (який код використовується, не має значення, оскільки числа додатні).

$$X=7_{(10)}=111_{(2)};$$

$$Y=2_{(10)}=10_{(2)}.$$

$$X= \begin{array}{|c|c|c|c|c|} \hline 0 & 0 & 1 & 1 & 1 \\ \hline \end{array}$$

$$Y= \begin{array}{|c|c|c|c|c|} \hline 0 & 0 & 0 & 1 & 0 \\ \hline \end{array}$$

Такти	X_i	Y_i	C_{i-1}	S_i	C_i
1	1	0	0	1	0
2	1	1	0	0	1
3	1	0	1	0	1
4	0	0	1	1	0
5	0	0	0	0	0

В результаті додавання отримана сума, що дорівнює:

$$S= \begin{array}{|c|c|c|c|c|} \hline 0 & 1 & 0 & 0 & 1 \\ \hline \end{array}$$

Перевіримо результат: $7_{(10)} + 2_{(10)} = 9_{(10)} = 1001_{(2)}$.

Далі в посібнику при вивченні алгоритмів арифметичних операцій в цифрових процесорах розглядатимуться і відповідні ним структури операційних автоматів (ОА). Операційний автомат - це частина цифрового пристрою, в якому виконуються дії над операндами, що знаходяться в регістрах. Спочатку розглянемо загальну структуру цифрового пристрою (ЦУ), складовою частиною якого є ОА (рисунок 1).

На рисунку прийняті наступні скорочення:

КОП - код операції, яку виконує ЦУ по сигналу "старт";

ОП - оперативна пам'ять;

ОА - операційний автомат;

УА - керуючий автомат, забезпечує виконання цифровим пристроєм алгоритму заданого КОП;

{Y} - керуючі сигнали, від УА, відповідні можливим мікроопераціям в ОА;

{X} - інформаційні сигнали (прапори) від ОА, що передаються в УА, для управління ходом алгоритму, значення яких визначається результатом виконаної мікрооперації.



Рисунок 1- Загальна структурна схема цифрового пристрою

До складу ОА входять регістри, тригери, комбінаційні схеми, пов'язані між собою і УА так, щоб можна було виконати певний алгоритм (алгоритми). УА і ОА разом утворюють простий цифровий процесор. Якщо цей цифровий процесор універсальний і призначений для виконання операцій додавання, віднімання, множення, ділення, то КОП визначає, яку саме операцію виконуватиме процесор при подачі сигналу "старт".

Структура операційного апарату для виконання алгоритму додавання послідовним способом приведена на рисунку 2.

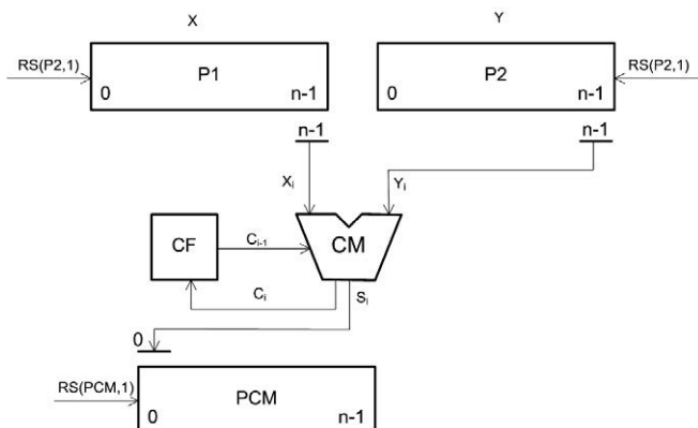


Рисунок 2 - Функціональна схема ОА для операції додавання послідовним способом

На цій функціональній схемі (і усіх наступних) прямокутниками позначені регістри (пристрої для зберігання інформації), трапецією - комбінаційні схеми (пристрої для перетворення інформації без її запам'ятовування). Призначення регістрів і комбінаційних схем на рисунку 2 таке:

- P1 - регістр для зберігання операнда X;
- P2 - регістр для зберігання операнда Y;
- PCM - регістр для зберігання суми;
- CM - однорозрядний суматор (працює відповідно до таблиці. 4.1);
- CF - прапор (тригер) перенесення;
- RS - сигнал зміщення управо регістра на один розряд;
- 0...(n - 1) - розрядність регістра (процесора).

При паралельному способі додавання операндів виконується за один такт роботи процесора, оскільки використовується багаторозрядний суматор (число розрядів суматора дорівнює розрядності процесора).

Схема ОА для алгоритму додавання паралельним способом представлена на рисунку 3.

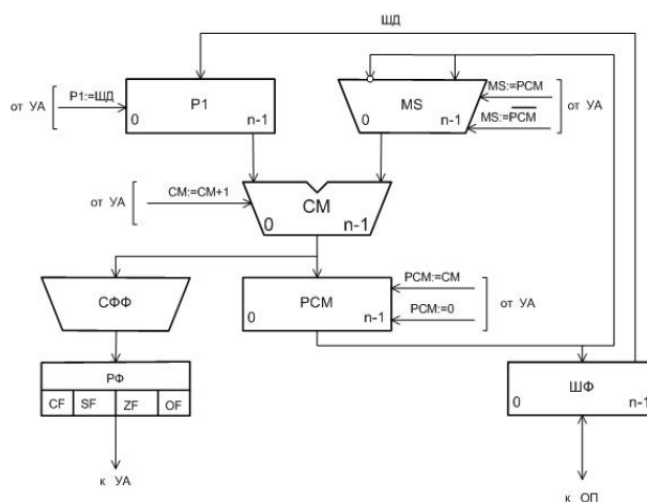


Рисунок 3 - Функціональна схема ОА для операції додавання паралельним способом

P1 - реєстр для зберігання операнда X;

PCM - реєстр для зберігання операнда Y, а потім для зберігання отриманої суми S;

0...(n - 1) - розрядність реєстра (процесора);

MS - мультиплексор (це схема, що має входи керування, декілька інформаційних входів і тільки один інформаційний вихід. Залежно від значення керуючих входів, в кожен момент часу тільки один інформаційний вхід підключається до виходу);

CM – n-розрядний суматор, який може працювати або в додатковому, або в оберненому коді;

СФФ - схема формування прапорів;

РФ - чотирьохбітовий реєстр прапорів (містить 4 біт);

CF - прапор перенесення (carry flag);

ZF - прапор рівності 0 результату операції (zero flag). Якщо результат операції дорівнює 0, то ZF = 1, інакше - 0;

SF - прапор знаку (sign flag). У нього записується значення знакового розряду результату;

OF - знак переповнювання (overflow flag). Якщо отриманий результат не поміщається в розрядній сітці процесора, OF=1, інакше - 0;

ШФ - шинний формувач дозволяє управляти шиною даних або на прийом, або на передачу.

Горизонтальна стрілка, що підходить до вузла, і напис над нею означають мікрооперацію, яка може бути виконана в цьому вузлі.

Оскільки в цьому конкретному ОА для введення операнда передбачена тільки одна шина даних, то для того, щоб скласти операнди X і Y, треба послідовно, один за одним, виконати два наступні алгоритми: «посилка X» і «додавання Y».

Алгоритми цих операцій представлені на рисунку 4 і 5.

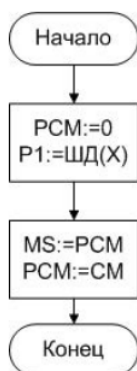


Рисунок 4 - Алгоритм операції «посилка X»

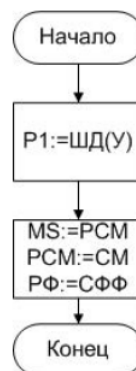


Рисунок 5 - Алгоритм операції «додавання Y»

2 Алгоритм додавання операндів у додатковому коді

Оскільки в цьому коді знаковий і числовий розряди розглядаються як єдине ціле, то суматор додаткового коду виконує над ними однакові дії.

Покажемо приклади додавання $S=X+Y$ в різних форматах для вісьмирозрядного процесора. Розглянемо додавання для формату цілих чисел з ФТ (фіксованою точкою).

Приклад 1. $X=53_{(8)}=101011_{(2)}$; $Y=35_{(8)}=11101_{(2)}$.

$[X]_2$		0	0	1	0	1	0	1	1
$[Y]_2$	+	0	0	0	1	1	1	0	1
$[S]_2$	=	0	1	0	0	1	0	0	0

SF=0; CF=0; ZF=0; OF=0.

Переведемо отриманий результат у вісімкову систему числення. Отримуємо $S=110_{(8)}$. Виконаємо перевірку додавання через вісімкову систему:

X		5	3
Y	+	3	5
S	=	1	10

Результат вірний.

Приклад 2. $X=53_{(8)}=101011_{(2)}$; $Y=-35_{(8)}=-11101_{(2)}$

$[X]_2$		0	0	1	0	1	0	1	1
$[Y]_2$	+	1	1	1	0	0	0	1	1
$[S]_2$	=	0	0	0	0	1	1	1	0

SF=0; CF=1; ZF=0; OF=0.

Переведемо отриманий результат у вісімкову систему числення. Отримуємо $S=16_{(8)}$. Виконаємо перевірку додавання через вісімкову систему:

X		5	3
Y	-	3	5
S	=	1	6

Результат вірний.

Приклад 3. $X=-53_{(8)}=-101011_{(2)}$; $Y=35_{(8)}=11101_{(2)}$.

$$\begin{array}{r}
 [X]_2 \\
 [Y]_2 + \\
 [S]_2 =
 \end{array}
 \begin{array}{|c|c|c|c|c|c|c|c|}
 \hline
 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
 \hline
 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
 \hline
 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\
 \hline
 \end{array}$$

SF=1; CF=0; ZF=0; OF=0.

Переведемо отриманий результат у вісімкову систему числення. Оскільки результат додавання має від'ємний знак, то спочатку треба перевести його з додаткового коду в двійкову систему числення. Отримуємо $S = -0001110_{(2)} = -1110_{(2)} = -16_{(8)}$. Виконаємо перевірку додавання через вісімкову систему:

X		5	3
	-		
Y	+	3	5
S	=	1	6
	-		

Результат вірний.

Розглянемо додавання для формату дробових чисел з ФТ.

Приклад 4. $X = -0,53_{(8)} = -0,101011_{(2)}$; $Y = 0,35_{(8)} = 0,11101_{(2)}$.

$$\begin{array}{r}
 [X]_2 \\
 [Y]_2 + \\
 [S]_2 =
 \end{array}
 \begin{array}{|c|c|c|c|c|c|c|c|}
 \hline
 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
 \hline
 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
 \hline
 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
 \hline
 \end{array}$$

SF=1; CF=0; ZF=0; OF=0.

Переведемо отриманий результат у вісімкову систему числення.

Оскільки результат додавання має від'ємний знак, то спочатку треба перевести його з додаткового коду в двійкову систему числення. Отримаємо $S = -0,0011100_{(2)} = -0,001110_{(2)} = -0,16_{(8)}$. Виконаємо перевірку додавання через вісімкову систему:

X	-	0,	5	3	
Y	+	0,	3	5	
S	=	-	0	1	6

Результат вірний.

Приклад 5. $X = 0,53_{(8)} = 0,101011_{(2)}$; $Y = 0,35_{(8)} = 0,11101_{(2)}$.

$$\begin{array}{r}
 [X]_2 \\
 [Y]_2 + \\
 [S]_2 =
 \end{array}
 \begin{array}{|c|c|c|c|c|c|c|c|}
 \hline
 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
 \hline
 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
 \hline
 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 \hline
 \end{array}$$

SF=1; CF=0; ZF=0; OF=1.

Результат додавання двох додатних чисел має від'ємний знак. Це тому, що сума не помістилася в 7 біт і зайняла ще один (знаковий) біт. Такий випадок називається "винятковим". Оскільки при виконанні арифметичних операцій можуть бути і інші "виняткові" випадки, наприклад ділення на 0, то кожен такий випадок дістав свою назву. Даний називається "переповнювання". У разі переповнювання розрядної сітки регістр РСМ містить неправильний результат додавання. Результатом додавання буде значення OF=1. Детальніше про переповнювання буде сказано далі.

З розглянутих прикладів видно, що алгоритм додавання дробових чисел у форматі з ФТ нічим не відрізняється від алгоритму додавання цілих чисел у форматі з ФТ.

Розглянемо додавання для формату змішаних чисел з ФТ в шістнадцятирозрядному процесорі (8 біт - ціла частина і знак, 8 біт - дробова).

Приклад 6. $X = -53,15_{(8)} = -101011,001101_{(2)}$; $Y = 35,11_{(8)} = 11101,001001_{(2)}$.

$[X]_2$	1	1	0	1	0	1	0	1	1	1	0	0	1	1	0	0
$[Y]_2$ +	0	0	0	1	1	1	0	1	0	0	1	0	0	1	0	0
$[S]_2$ =	1	1	1	1	0	0	1	0	1	1	1	1	0	0	0	0

SF=1; CF=0; ZF=0; OF=0.

Переведемо отриманий результат у вісімкову систему числення. Оскільки результат додавання має від'ємний знак, то спочатку треба перевести його з додаткового коду в двійкову систему числення. Отримуємо $S = -0001110,00010000_{(2)} = -1110,000100_{(2)} = -16,04_{(8)}$. Виконаємо перевірку додавання через вісімкову систему:

X	-	5	3	,	1	5	
Y	+	3	5	,	1	1	
S	=	-	1	6	,	0	4

Результат вірний.

Таким чином, видно, що додавання чисел в будь-якому форматі з ФТ відбувається по одному і тому ж алгоритму.

3 Алгоритм додавання операндів в оберненому коді

Відмінність алгоритму додавання в оберненому коді від алгоритму додавання в додатковому коді полягає в тому, що додавання виконується в два етапи. На першому етапі складаються операнди, а на другому до отриманої суми додається значення прапора CF, отриманого при складанні операндів.

Так само як і в додатковому коді, алгоритм додавання в оберненому коді однаковий для усіх форматів з ФТ. Розглянемо приклад додавання цілих чисел у форматі з ФТ на восьмирозрядному процесорі.

Приклад. $X = 53_{(8)} = 101011_{(2)}$; $Y = -35_{(8)} = -11101_{(2)}$.

1-й етап:

$[X]_3$	0	0	1	0	1	0	1	1
$[Y]_3$ +	1	1	1	0	0	0	1	0
$[S]_3$ =	0	0	0	0	1	1	0	1

CF=1;

2-й етап:

$[S]_3$	0	0	0	0	1	1	0	1
CF +								1
	0	0	0	0	1	1	1	0

SF=0; CF=0; ZF=0; OF=0.

Переведемо отриманий результат у вісімкову систему числення. Отримуємо $S = 0001110_{(2)} = 1110_{(2)} = 16_{(8)}$. Здійснимо перевірку додавання через вісімкову систему:

X	5	3	
Y	-	3	5
S	=	1	6

Результат вірний.