

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**  
**НАУКИ І ТЕХНОЛОГІЙ**

---

Кафедра «Електронні обчислювальні машини»

**АРХІТЕКТУРА КОМП'ЮТЕРІВ**

Методичні рекомендації для виконання практичних робіт

Електронний аналог  
друкованого видання

Дніпро  
2023

УДК 004.2  
А 87

Укладачі:  
*О.Й. Єгоров, В.В. Дзюба, П.В. Івін*

Рекомендовано МКФ «КТС» (протокол № 597 від 13.02.2023).

Зареєстровано НМВ УДУНТ (№ 597 від 13.02.2023)

Архітектура комп'ютерів [Текст] : методичні рекомендації для виконання практичних робіт / уклад. : О.Й. Єгоров, В.В. Дзюба, П.В. Івін; Укр. держ. ун-т науки і технологій. – Дніпро : УДУНТ, 2023. – 37 с..

Методичні рекомендації призначено студентам 3-го курсу спеціальності 123 «Комп'ютерна інженерія» для підготовки до виконання практичних робіт з дисципліни «Архітектура комп'ютерів».

Іл. 23. Табл. 7. Бібліогр.: 9 назв.

© Єгоров О.Й. та ін., укладання, 2023  
© Укр. держ. ун-т науки і технологій, 2023

## ЗМІСТ

Вступ.....	4
Практична робота № 1. Ознайомлення з мовою опису логічних схем JOLS-M. Виконання операцій на мові JOLS-M.....	5
Практична робота № 2. Виконання операції додавання чисел з фіксованою комою.....	10
Практична робота № 3. Дослідження пристрою множення чисел з фіксованою комою.....	15
Практична робота № 4. Дослідження логічних методів прискореного множення чисел (множення з аналізом декількох розрядів).....	21
Практична робота № 5. Вивчення способів ділення чисел з фіксованою комою.....	24
Практична робота № 6. Вивчення операції додавання двійкових чисел з плаваючою комою.....	30
Список літератури.....	36

## ВСТУП

Метою практичних робіт є набуття практичних навичок у використанні отриманих знань при розробці структур, алгоритмів та мікропрограм функціонування основних пристроїв арифметико-логічного пристрою (АЛП) центрального процесора, а також закріплення основних теоретичних положень курсу. В результаті виконання практичних робіт студенти повинні отримати чітке уявлення про взаємодію основних вузлів та блоків АЛП процесора у процесі виконання арифметичних операцій та навчитися використовувати апарат, методи та засоби проектування ЕОМ. Цій меті найкраще відповідає самостійне виконання студентами розробки структури, алгоритмічного опису, схем.

Виконання наведених практичних робіт сприяє:

- покращенню розуміння процесів виконання арифметичних операцій АЛП;
- отриманню практичних навичок розробки пристроїв, алгоритмів і мікропрограм функціонування АЛП;
- формуванню вміння документувати та презентувати результати розробок елементів АЛП через складання звітної документації та захист лабораторних робіт.

Програмні результати навчання, досягненню яких сприяє видання:

- знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж (ПРН1);
- вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей (ПРН6);
- вміти застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп'ютерних систем та мереж для вирішення технічних задач спеціальності (ПРН9).

## ПРАКТИЧНА РОБОТА № 1

### ОЗНАЙОМЛЕННЯ З МОВОЮ ОПИСУ ЛОГІЧНИХ СХЕМ JOLS-M. ВИКОНАННЯ ОПЕРАЦІЙ НА МОВІ JOLS-M

- Мета:**
1. Ознайомитися з синтаксисом мови JOLS-M.
  2. Вивчити особливості представлення змінних та виконання арифметичних та логічних операцій над числами.
  3. Отримати навички формування структури пристрою і написання мікропрограми на мові JOLS-M.

#### 1. Теоретичні відомості

Для створення мікропрограм на мові JOLS-M розроблено декілька інтегрованих середовищ розробки, одне з яких – JOLS-M, що дозволяє виконувати:

- опис структури елементів модельованого пристрою;
- введення мікропрограми, що моделює функціонування досліджуваного пристрою;
- виконання мікропрограми;
- покрокове відлагодження мікропрограми;
- видачу на друк або дисплей структури елементів, лістингу мікропрограми моделі і результатів моделювання.

Мікропрограма на JOLS-M є скінчене число рядків по одній мікрокоманді в кожному рядку. Ядро мови JOLS-M включає наступні мікрокоманди:

- INPUT
- OPERATION
- PRINT
- GO\_TO
- IF
- END
- REMAR

Перераховані мікрокоманди допускають використання міток.

#### Мікрокоманда INPUT

Мікрокоманда **INPUT** призначена для зміни в ході виконання мікропрограми вмісту будь-якого регістра або елемента пам'яті. При цьому в спеціальне вікно вводу виводиться поточний вміст вказаного регістра або елемента пам'яті в двійковому коді. Виконання мікропрограми припиняється для вводу нового значення або зміни старого. Після введення значення виконання мікропрограми продовжується.

```
{ mmm } INPUT [ операнд1 ],
```

де { mmm } – мітка (1...999);

**операнд1** – будь-який з перерахованих нижче варіантів:

- рХХ – реєстр з номером ХХ (0..39);
- пХХХ – комірка пам'яті з адресою ХХХ (0..254);
- р(рХХ) – реєстр з адресою, що зберігається в реєстрі ХХ;
- п(пХХХ) – комірка пам'яті з адресою, що зберігається в комірці з адресою ХХХ;
- р(пХХХ) – реєстр, адреса якого зберігається в комірці пам'яті;
- п(рХХ) – комірка пам'яті, адреса якої зберігається в реєстрі.

### Мікрокоманда OPERATION

Мікрокоманда **OPERATION** призначена для виконання різних дій над реєстрами, елементами пам'яті і константами, таких як: додавання, віднімання, логічне І, логічне АБО, присвоєння, додавання по модулю 2, циклічний зсув, логічний зсув, додавання з циклічним перенесенням, логічне НЕ.

За допомогою спеціального знаку (~) в операції братиме участь інверсне значення цього операнда, а сам він при цьому не зміниться. В операціях може брати участь безпосередньо реєстр або елемент пам'яті, окремий біт або група розрядів, константи в двійковій, десятковій або шістнадцятковій системі числення, а також реєстри або елементи пам'яті що адресуються за непрямою адресою (рос. – «косвенно»).

**{ mmm } OPERATION [операнд1] [функція] [операнд2 ]{~},**

де **{ mmm }** – мітка (1..999);

**операнд1** – будь-який з перерахованих нижче варіантів:

- рХХ;
- рХХ(NN);
- рХХ(NN:nn);
- пХХХ;
- пХХХ(NN);
- пХХ(NN:nn);
- р(рХХ);
- р(пХХХ);
- п(пХХХ);
- п(рХХ),

де ХХ – номер реєстра (0..39);

ХХХ – номер елемента пам'яті (0..254);

NN – номер старшого біта (включно);

nn – номер молодшого біта (включно);

**операнд2** – може приймати ті ж значення, що і операнд1, а також будь-яке з наведених нижче:

- dddd – десяткова константа ( $-2^{31}..2^{31}$ );
- \$hhh – шістнадцятирична константа (0..FFFFFFFFh);
- #bbb – (0.. 1111 1111 1111 1111 1111 1111 1111 1111);

**функція** – будь-яка з перерахованих нижче дій над операндами:

- = присвоєння;
- + додавання;
- - віднімання;
- / логічне додавання (АБО);
- & логічне множення (І);
- @ додавання по модулю 2;
- >> зсув вправо логічний;
- << зсув вліво логічний;
- >] зсув вправо циклічний;
- [< зсув вліво циклічний;
- ++ додавання з циклічним переносом;
- ~ ознака інверсії другого операнда.

### **Мікрокоманда PRINT**

Мікрокоманда **PRINT** дає можливість виводу значень регістрів, комірок пам'яті, а також текстових повідомлень в ході виконання мікропрограми.

а) **{mmm} PRINT [операнд1] .. [операнд7] ,**

де **{mmm}** – мітка (1..999);

**операнд1 .. операнд7** – будь-який з перерахованих нижче варіантів:

- рXX;
- пXXX;
- р(рXX);
- р(пXXX);
- п(пXXX);
- п(рXX),  
де XX – номери регістрів (0..39);  
XXX – номер елемента пам'яті (0..254);

б) **{mmm} ПЕЧАТЬ «будь-який текст»**

де **{mmm}** – мітка (1..999).

### **Мікрокоманда GO\_TO**

Мікрокоманда **GO\_TO** призначена для безумовної передачі управління мікрокоманди з міткою. При цьому мікропрограма продовжує виконуватися з рядка з вказаною міткою. Мітка повинна знаходитися в діапазоні значень 1..999. У мікрокоманди **GO\_TO** може бути вказано безпосередньо значення мітки, регістру або елемента пам'яті, що адресується побічно.

**{mmm} GO\_TO [ вираз ] ,**

де {mmm} – мітка (1..999);

**вираз** – будь-який з перерахованих нижче варіантів:

- ddd – мітка для переходу (1..999);
- рХХ;
- пХХХ;
- р(рХХ);
- р(пХХХ);
- п(пХХХ);
- п(рХХ),

де ХХ – номер регістра (0..39);

ХХХ – номер елемента пам'яті (0..254).

### Мікрокоманда IF

Мікрокоманда **IF** виконує операції відношення над операндами, такі як =, <>, <=, >= . Якщо відношення істинно, то наступною виконується команда, вказана за мікрокомандою **IF** в цьому ж рядку, інакше буде виконуватися наступний рядок.

{mmm} IF [опер1] [відношення] [опер2] [мікрокоманда] ,

де {mmm} – (1..999);

**опер1** – будь-який з перерахованих нижче варіантів:

- рХХ;
- рХХ(NN);
- рХХ(NN:nn);
- пХХХ;
- пХХХ(NN);
- пХХ(NN:nn);
- р(рХХ);
- р(пХХХ);
- п(пХХХ);
- п(рХХ),

де ХХ – номер регістра (0..39);

ХХХ – номер елемента пам'яті (0..254);

NN – номер старшого біта (включно);

nn – номер молодшого біта (включно) (**Увага !!! NN => nn**);

**опер2** – може приймати ті ж значення, що і **опер1**, а також будь-яке з приведених нижче:

- dddd – десяткова константа (-1\*2<sup>31</sup> .. 2\*2<sup>31</sup>);
- \$hhh – шістнадцятирична константа (0.. FFFFFFFFh);
- #bbb – двійкова константа (0 .. 1111 1111 1111 1111 1111 1111 1111);

**відношення** – будь-яке з перерахованих нижче мікрокоманд відносин:

- = дорівнює;
- < менше;

- > більше;
- <= менше або дорівнює;
- >= більше або дорівнює;
- < > нерівний;

**мікрокоманда** – будь-яка з перерахованих:

- INPUT;
- OPERATION;
- PRINT;
- GO\_TO;
- IF;
- END;
- REMAR.

**УВАГА!!!** Мікрокоманду **IF** не дозволяється використовувати як оператор.

### **Мікрокоманда END**

Мікрокоманда **END** припиняє виконання мікропрограми.

**{mmm} END,**

де **{mmm}** – (1..999).

### **REMAR**

Мікрокоманда ремарки (одинарна лапка `) призначена для вказівки в тексті мікропрограми коментарів, що різко підвищує наочність мікропрограми. При виконанні мікропрограми ігнорується.

### **2. Постанова задачі**

Розробити мікропрограму для демонстрації виконання мікрокоманд та операцій.

### **3. Порядок виконання роботи**

1. Завантажити JOLS-M.
2. Описати довільну структуру пристрою з використанням регістрів та комірок пам'яті.
3. Скласти мікропрограму, в якій передбачити:
  - введення інформації в декілька регістрів та комірок пам'яті;
  - передачу інформації з регістра в комірку пам'яті;
  - передачу інформації з комірки пам'яті в регістр;
  - виконання логічних операцій «I» та «АБО», додавання за модулем 2 над вмістом двох регістрів, вмістом регістра та комірки пам'яті;
  - логічний та циклічний зсув вліво та вправо вмісту регістрів на 1 та 4 розряди;

- логічний зсув вліво та вправо, при якому кількість зсувів вказано в регістрі;
- логічний зсув вліво та вправо на 1 та 4 розряди вмісту частини розрядів регістра (виконувати зсув приблизно половини розрядів регістрів);
- виконати передачу інформації з одного регістра в інший з перекосом на 2 розряди вліво та на 3 розряди вправо;

#### 4. Виконати моделювання на ЕОМ.

Результат кожної дії, в тому числі й вихідні числа слід вивести на друк.

Вихідні дані для виконання кожної операції слід підібрати так, щоб було видно характерні особливості виконання даної операції.

### 4. Зміст звіту

Звіт повинен містити короткі теоретичні відомості, роздруковану мікропрограму та результати моделювання.

### 5. Контрольні запитання та завдання

1. Чи існує можливість в мові JOLS-M виконувати дії над частиною вмісту регістру?

2. Як відбувається вирівнювання інформації при її передачі з регістра з меншою розрядністю в регістр з більшою розрядністю: по молодшим чи старшим розрядам?

3. Як передати інформацію з регістра в регістр не виконуючи мікрооперацію зсуву (передача з перекосом)?

4. Як поміняти місцями вміст двох регістрів не використовуючи проміжного?

5. Які основні операції використовуються в JOLS-M?

6. Чим відрізняється циклічний та логічний зсув?

7. Які мікрокоманди використовуються в мові JOLS-M?

## ПРАКТИЧНА РОБОТА № 2

### ВИКОНАННЯ ОПЕРАЦІЇ ДОДАВАННЯ ЧИСЕЛ З ФІКСОВАНОЮ КОМОЮ

**Мета:** 1. Ознайомитися з правилами та методами додавання чисел з фіксованою комою.

2. Отримати навички використання зворотного та додаткового кодів.

#### 1. Теоретичні відомості

**1.1. Зворотний та додатковий коди.** Для виконання операції додавання з різними знаками використовуються зворотний (ЗК) або додатковий коди (ДК). Старший розряд числа використовується як знаковий. Цифра 0 в знаковому розряді свідчить про додатний знак числа («+»), а цифра 1 – про від'ємний («-»).



Перетворення кодів та всі інші дії виконуються аналогічно звичайним ЗК та ДК. Наприклад, виконаємо додавання чисел 5 та 13 з використанням МДК та 4-розрядної сітки. Хід операції додавання наведено на рисунку 2.3.

```

00.0101 (5)
00.1101 (13)
-----
01.0010 (переповнення!)

```

Рисунок 2.3. Приклад операції додавання із виникненням переповнення

На рисунку 2.4 представлена структура пристрою для додавання двох n-розрядних чисел з фіксованою комою.

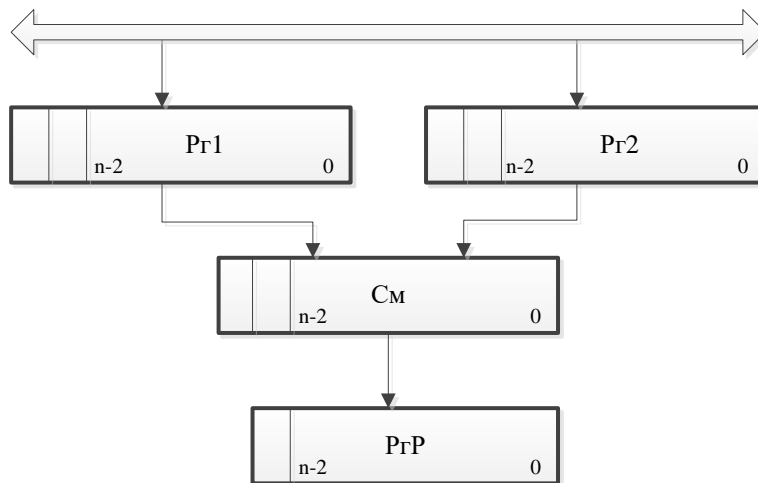


Рисунок 2.4. Структура пристрою для додавання двох n-розрядних чисел з фіксованою комою

Регістри Rr1 і Rr2 містять два вихідних операнда, старший біт яких є знаковим. На суматорі Cm відбувається складання чисел. У цьому слід врахувати, що суматор є накопичувальним. Результат виконання операції заноситься до регістру результату RrP.

На рисунку 2.5 відображено загальний алгоритм виконання операції додавання двох n-розрядних чисел з фіксованою комою.

## 2. Постановка задачі

Розробити структуру, алгоритм і мікропрограму пристрою для виконання операції додавання двох n-розрядних чисел з фіксованою комою згідно варіанту (табл. 2.1).

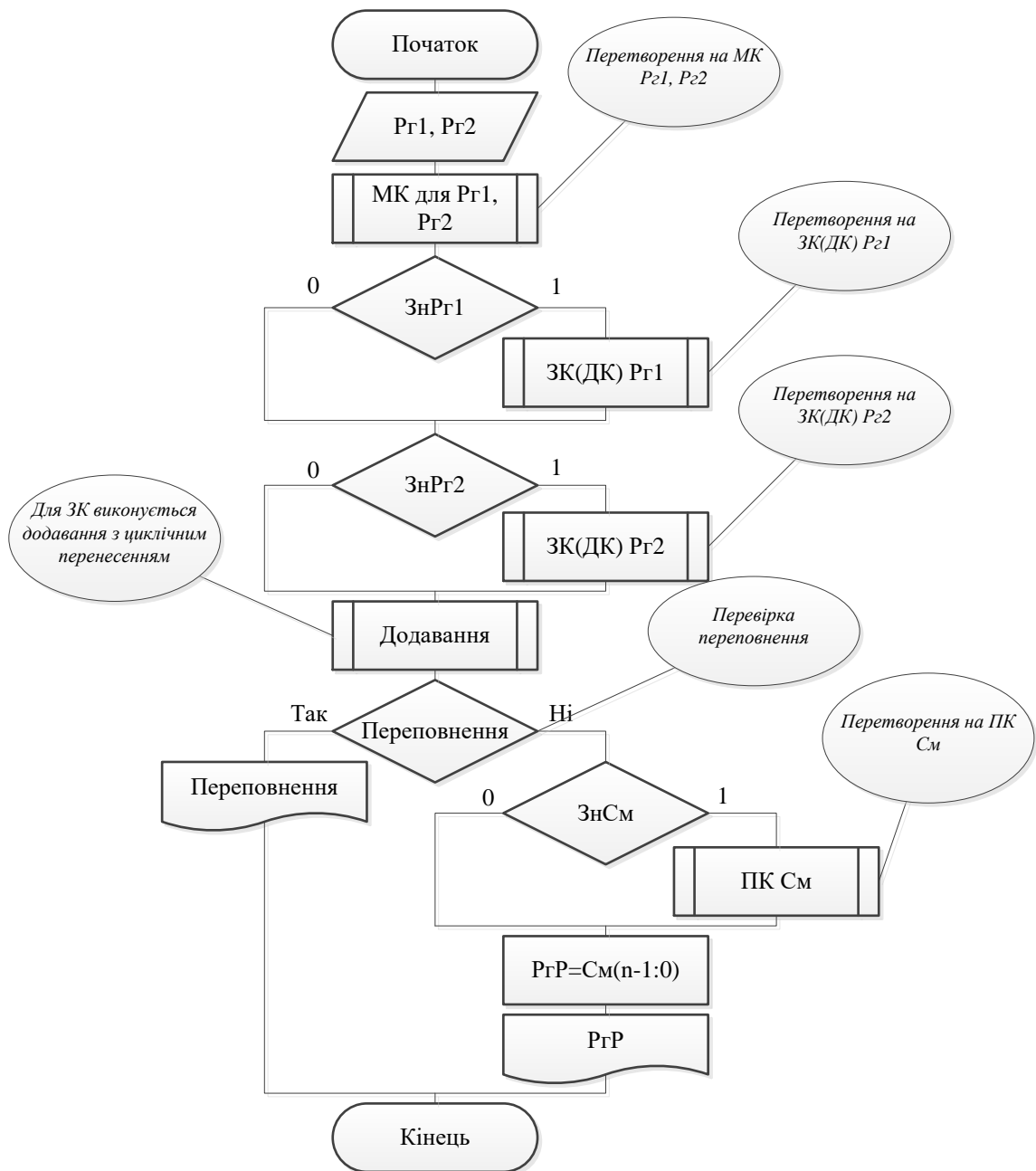


Рисунок 2.5. Загальний алгоритм додавання двох n-розрядних чисел з фіксованою комою

### 3. Порядок виконання роботи (згідно варіанту)

1. Скласти структуру пристрою додавання.
2. Розробити алгоритм додавання.
3. Завантажити JOLS-M.
4. Описати структуру пристрою додавання.
5. Скласти мікропрограму, в якій передбачити виконання операції додавання чисел з різними знаками, з імітацією застосування зворотного або додаткового модифікованого коду.

Таблиця 2.1. Варіанти завдань для виконання операції додавання

№ Варіанту	Розрядність	Код	№ Варіанту	Розрядність	Код
1	4	ЗК	14	10	ДК
2	4	ДК	15	11	ЗК
3	5	ЗК	16	11	ДК
4	5	ДК	17	12	ЗК
5	6	ЗК	18	12	ДК
6	6	ДК	19	13	ЗК
7	7	ЗК	20	13	ДК
8	7	ДК	21	14	ЗК
9	8	ЗК	22	14	ДК
10	8	ДК	23	15	ЗК
11	9	ЗК	24	15	ДК
12	9	ДК	25	16	ЗК
13	10	ЗК	26	16	ДК

**УВАГА!!!** Операнди слід обрати таким чином, щоб отримати один додатний, один від'ємний результат та переповнення.

Результат кожної дії, в тому числі й вихідні числа слід вивести на друк.

Вихідні дані для виконання кожної операції слід підібрати так, щоб було видно характерні особливості виконання даної операції.

#### 4. Зміст звіту

Звіт повинен містити короткі теоретичні відомості, розроблені структуру, алгоритм (блок-схема) і мікропрограму для виконання операції додавання двох операндів, а також результати моделювання для чисел з різними знаками.

#### 5. Контрольні запитання та завдання

1. Як утворюються зворотний та додатковий коди?
2. Чим відрізняється додавання в ЗК від додавання в ДК?
3. Для чого використовуються зворотний та додатковий коди?
4. Що таке переповнення розрядної сітки? Як визначається переповнення?
5. Чому нульове значення розряду переповнення від'ємного числа свідчить про переповнення?
6. Як визначається переповнення при відсутності в суматорі розряду переповнення?

## ПРАКТИЧНА РОБОТА № 3

### ДОСЛІДЖЕННЯ ПРИСТРОЮ МНОЖЕННЯ ЧИСЕЛ З ФІКСОВАНОЮ КОМОЮ

- Мета:** 1. Ознайомитися з існуючими способами множення чисел з фіксованою комою.  
2. Промоделювати роботу способів множення на ЕОМ згідно варіанту завдання.

#### 1. Теоретичні відомості

Частковий добуток – це добуток множеного на один або декілька розрядів множника. При аналізі одного розряду множника значення часткового добутку може дорівнювати нулю (коли розряд множника дорівнює «0»), або значенню множеного (коли розряд множника – «1»).

Для множення двійкових чисел можна застосовувати декілька способів.

**Початок множення з молодших розрядів множника та зсув суми часткових добутків (рисунок 3.1, спосіб №1).** В кожному такті множення аналізується молодший розряд множника. Якщо він дорівнює 1, то до вмісту суми часткових добутків додається значення множеного. Причому множене

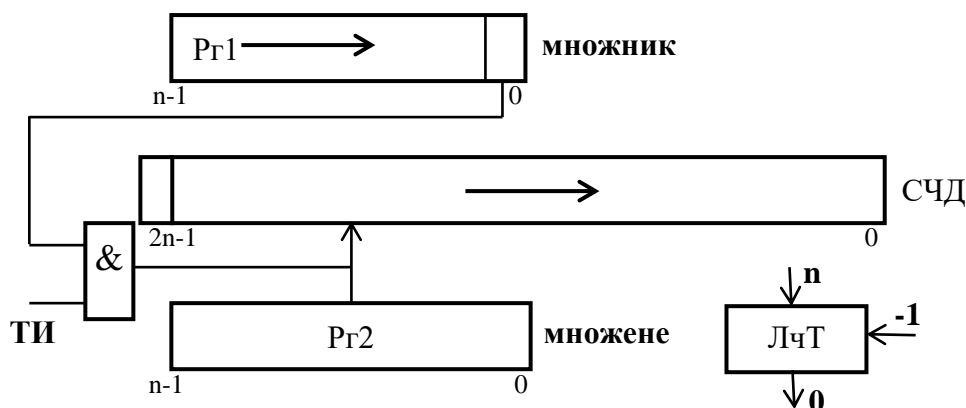


Рисунок 3.1. Структура пристрою множення з молодших розрядів множника та зсув суми часткових добутків

додається до старшої частини суми часткових добутків. Далі сума часткових добутків та множник зсуваються вправо на один розряд.

**УВАГА!!!** Для суматора потрібен додатковий розряд.

На рисунку 3.2 представлений приклад виконання операції множення даним способом.

<b>Множник:</b> $5_{10}=0101_2$		<b>Множене:</b> $13_{10}=1101_2$	
СЧД		Множник	
0	00000000	0101	п1
0	1101		
	<u>0 11010000</u>		
	0 01101000	0010	п1
	0 0000		
	<u>0 01101000</u>		
	0 00110100	0001	п1
	0 1101		
	<u>1 00000100</u>		
	0 10000010	0000	п1
	0 0000		
	<u>0 10000010</u>		
	0 01000001		

Рисунок 3.2. Приклад множення з молодших розрядів множника та зсув СЧД

**Початок множення з молодших розрядів множника та зсув множеного (рисунок 3.3, спосіб №2).** Перед початком множення розташовується у регістрі з подвійною розрядністю починаючи з молодших

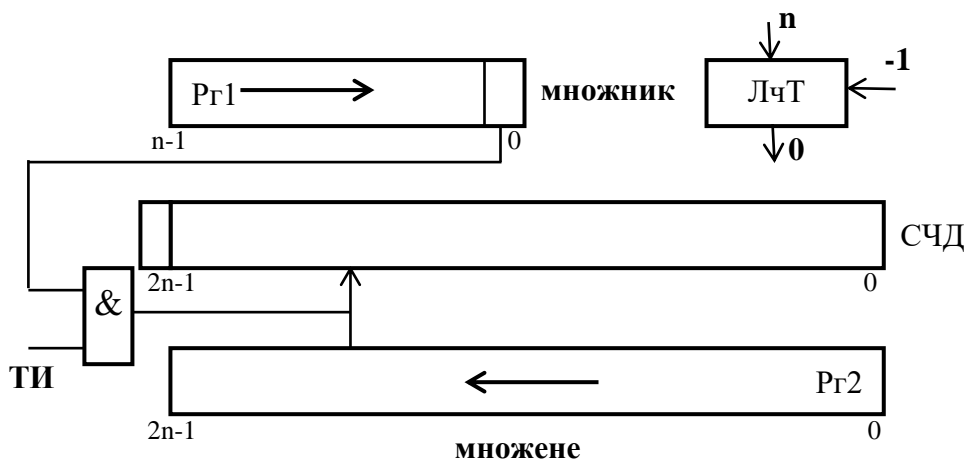


Рисунок 3.3. Структура пристрою множення з молодших розрядів множника та зсув множеного

розрядів. В кожному такті множення аналізується молодший розряд множника. Якщо він дорівнює 1, то до вмісту суми часткових добутоків

додається значення множеного. Далі множене зсувається вліво на один розряд, а множник – вправо на 1 розряд.

На рисунку 3.4 представлений приклад виконання операції множення даним способом.

СЧД	Множник	Множене
0 00000000	1001 П1	
0 00001011		00001011 Л1
0 00001011	0100 П1	
0 00000000		00010110 Л1
0 00001011	0010 П1	
0 00000000		00101100 Л1
0 00001011	0001 П1	
0 01011000		01011000 Л1
0 01100011		

Рисунок 3.4. Приклад множення з молодших розрядів множника та зсув множеного

Початок множення зі старших розрядів множника та зсув суми часткових добутоків (рисунок 3.5, спосіб №3). В кожному такті множення аналізується старший розряд множника. Якщо він дорівнює 1, то до вмісту суми часткових добутоків додається значення множеного. Причому множене додається до молодшої частини суми часткових добутоків. Далі сума часткових добутоків та множник зсуваються вліво на один розряд.

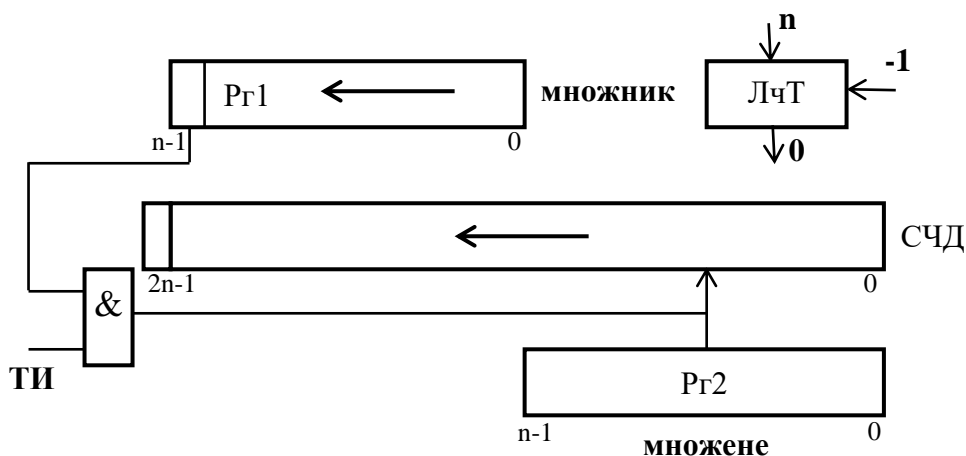


Рисунок 3.5. Структура пристрою множення зі старших розрядів множника та зсув суми часткових добутоків

На рисунку 3.6 представлений приклад виконання операції множення даним способом.

**УВАГА!!!** В даному способі останній зсув не виконується.

<b>Множник:</b> $5_{10}=0101_2$	
<b>Множене:</b> $13_{10}=1101_2$	
СЧД	Множник
0 00000000	0101 л1
0       0000	
0 00000000	л1
0 00000000	1010 л1
0       1101	
0 00001101	л1
0 00011010	0100 л1
0       0000	
1 00011010	л1
0 00110100	1000 л1
0       1101	
0 01000001	

Рисунок 3.6. Приклад множення зі старших розрядів множника та зсув СЧД

**Початок множення зі старших розрядів множника та зсув множеного (рисунок 3.7, спосіб №4).** Перед початком множення знаходиться у регістрі з подвійною розрядністю починаючи зі старших розрядів. Розпочинається кожен такт із зсуву множеного на 1 розряд вправо. Аналізується старший розряд множника. Якщо він дорівнює 1, то до вмісту суми часткових добутків додається значення множеного. Далі виконується зсув множника на 1 розряд вліво.

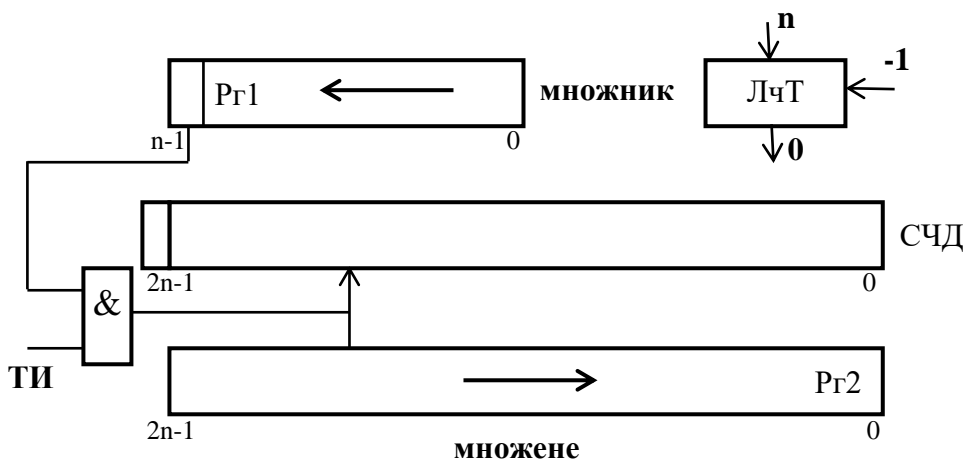


Рисунок 3.7. Структура пристрою множення зі старших розрядів множника та зсув множеного

На рисунку 3.8 представлений приклад виконання операції множення даним способом

**УВАГА!!!** В даному способі по перше виконується зсув, а потім додавання. Для суматора потрібен додатковий розряд.

СЧД	Множник	Множене
0 00000000	1001 Л1	
0 01011000		П1 01011000
0 01011000	0010 Л1	
0 00000000		П1 00101100
0 01011000	0100 Л1	
0 00000000		П1 00010110
0 01011000	1000 Л1	
0 00001011		П1 00001011
0 01100011		

Рисунок 3.8. Приклад множення зі старших розрядів множника та зсув множеного

Кількість тактів для кожного способу відповідає розрядності операндів. Час виконання операції множення можна оцінити за формулою:

$$T_{\text{множ}} = n \times (t_{\text{дод}} + t_{\text{зс}}), \quad (3.1)$$

де  $n$  – кількість розрядів;

$t_{\text{дод}}$  – час додавання;

$t_{\text{зс}}$  – час виконання зсувів.

Загальний алгоритм множення відображено на рисунку 3.9. Регістри Rг1 і Rг2 містять два вихідних операнда. На суматорі СЧД відбувається формування суми часткових добутків. Лічильник ЛчТ задає кількість ітерацій виконання операцій додавання і залежить від розрядності операндів. Результат виконання операції формується в СЧД.

## 2. Постановка задачі

Розробити структуру, алгоритм і мікропрограму пристрою для виконання операції множення двох  $n$ -розрядних чисел з фіксованою комою згідно варіанту (табл. 3.1). Знак операндів можна ігнорувати.

## 3. Порядок виконання роботи (згідно варіанту)

1. Скласти структуру пристрою множення.
2. Розробити алгоритм множення.
3. Завантажити JOLS-M.
4. Описати структуру пристрою множення.
5. Скласти мікропрограми реалізації заданого способу множення.

6. Промоделювати роботу пристрою множення на ЕОМ, виконуючи друк вмісту регістру множника та регістру сум часткових добутоків після кожного такту множення.

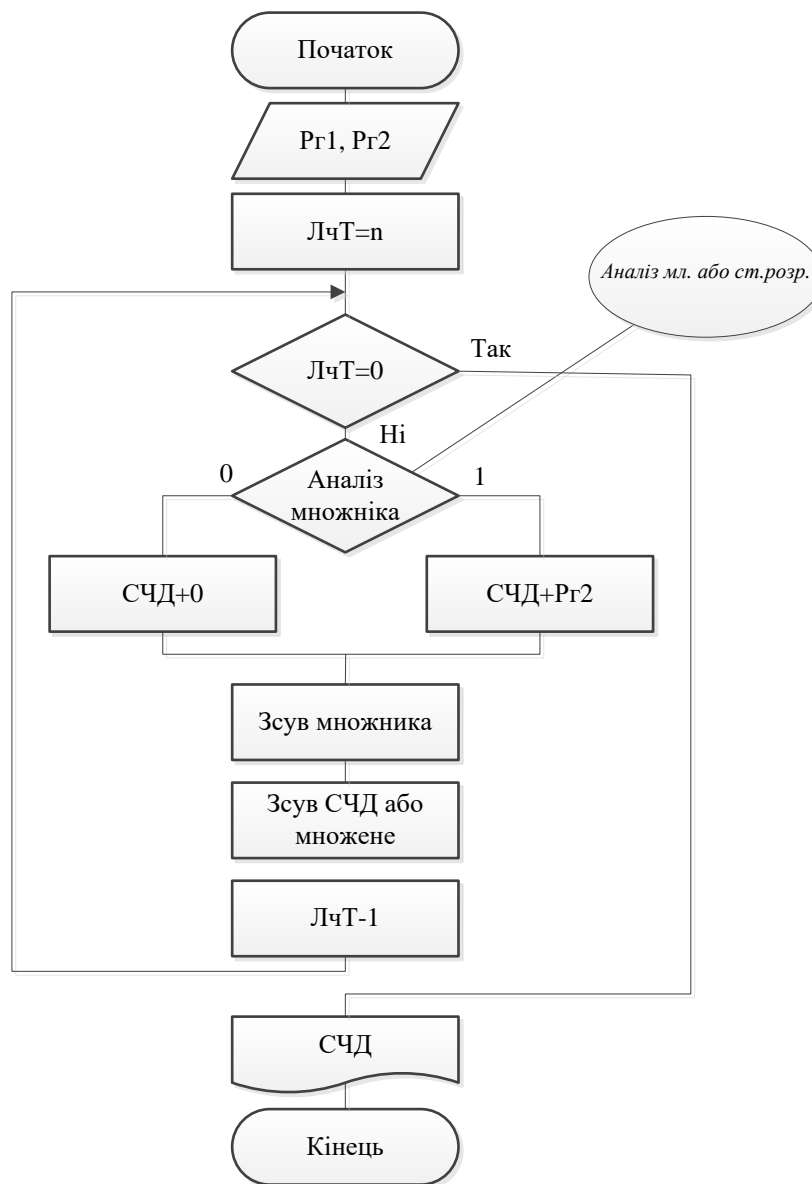


Рисунок 3.9. Загальний алгоритм множення двох n-розрядних чисел з фіксованою комою

#### 4. Зміст звіту

Звіт повинен містити короткі теоретичні відомості, розроблені структуру, алгоритм (блок-схема) і мікропрограму для виконання операції множення двох операндів, а також результати моделювання.

Таблиця 3.1. Варіанти завдань для виконання операції множення

№ варіанту	Розрядність співмножників	Спосіб множення	№ варіанту	Розрядність співмножників	Спосіб множення
1	4	1	14	7	2
2	4	2	15	7	3
3	4	3	16	7	4
4	4	4	17	8	1
5	5	1	18	8	2
6	5	2	19	8	3
7	5	3	20	8	4
8	5	4	21	9	1
9	6	1	22	9	2
10	6	2	23	9	3
11	6	3	24	9	4
12	6	4	25	10	1
13	7	1	26	10	2

Примітка: Способи множення позначені наступним чином: 1 – початок множення з молодших розрядів множника та зсув суми часткових добутоків; 2 – початок множення з молодших розрядів множника та зсув множеного; 3 – початок множення зі старших розрядів множника та зсув суми часткових добутоків; 4 – початок множення зі старших розрядів множника та зсув множеного.

### 5. Контрольні запитання та завдання

1. Що таке частковий добуток?
2. Чим відрізняються машинні способи множення один від одного?
3. Чому може дорівнювати черговий частковий добуток в двійковій системі числення?
4. Чому дорівнює розрядність добутку при n-розрядних співмножниках?
5. Яка мета округлення добутку?
6. Які мікрооперації виконуються в кожному такті множення?
7. Чому дорівнює час множення?
8. Які логічні способи множення Вам відомі? Назвіть час виконання множення при використанні цих способів.

### ПРАКТИЧНА РОБОТА № 4

#### ДОСЛІДЖЕННЯ ЛОГІЧНИХ МЕТОДІВ ПРИСКОРЕНОГО МНОЖЕННЯ ЧИСЕЛ (МНОЖЕННЯ З АНАЛІЗОМ ДЕКІЛЬКОХ РОЗРЯДІВ)

- Мета:**
1. Ознайомитися з існуючими способами прискореного множення чисел з фіксованою комою.
  2. Промоделювати роботу способів прискореного множення на ЕОМ згідно варіанту завдання.

## 1. Теоретичні відомості

Структурна схема пристрою для прискореного множення з аналізом декількох розрядів аналогічна пристрою для множення з порозрядним аналізом множника. Але існує декілька особливостей виконання даної операції. В кожному такті аналізується  $k$  розрядів множника, відповідним чином визначається частковий добуток, який додається до суми часткових добутоків. Зсув множника і суми часткових добутоків (або множеного – в залежності від способу) виконується на  $k$  розрядів.

В таблицях 4.1 і 4.2 наведено значення часткових добутоків при різних значеннях розрядів множника для способу з аналізом 2 розрядів. В таблиці 4.1 наведено значення для способу множення починаючи з молодших розрядів, в таблиці 4.2 – зі старших (*примітка*: А – множене).

Таблиця 4.1. Значення часткових добутоків для способу множення починаючи з молодших розрядів

Поточна пара розрядів	Ознака корекції в поточному такті	Частковий добуток	Ознака корекції в наступному такті
00	0	0	0
01		A	0
10		2A	0
11		-A	1
00	1	A	0
01		2A	0
10		-A	1
11		0	1

Таблиця 4.2. Значення часткових добутоків для способу множення починаючи зі старших розрядів

Поточна пара розрядів	Наступний розряд	Частковий добуток
00	0	0
01		A
10		-2A
11		-A
00	1	A
01		2A
10		-A
11		0

**Початок множення з молодших розрядів.** Біт ознаки корекції може формуватися автоматично. Для цього в регістрі суми часткових добутоків слід використовувати додатковий розряд (найстарший). Для множення із зсувом

суми часткових добутоків крім цього розряду слід використовувати ще 2 додаткових (для організації зсуву і т.п.). Слід зауважити, що якщо в останньому такті множення отримано розряд корекції, що дорівнює одиниці, то необхідно виконати ще один додатковий такт множення.

**Початок множення зі старших розрядів.** Множення починається з аналізу фіктивної старшої пари розрядів «00». Далі усі дії виконуються згідно табл. 4.2. При множенні із зсувом множеного в першому такті множене не зсувається. Закінчується множення операцією додавання.

Кількість тактів для операції множення обирається згідно розрядності співмножників і дорівнює  $n/2$  розряди (з округленням в більшу сторону).

Час виконання операції прискореного множення з аналізом двох розрядів можна оцінити за формулою:

$$T_{\text{множ}} = n/2 \times (t_{\text{дод}} + t_{\text{зс}}), \quad (4.1)$$

де  $n$  – кількість розрядів;

$t_{\text{дод}}$  – час додавання;

$t_{\text{зс}}$  – час виконання зсувів.

## 2. Постанова задачі

Розробити структуру, алгоритм і мікропрограму пристрою для виконання операції множення двох  $n$ -розрядних чисел з фіксованою комою з аналізом двох розрядів згідно варіанту (табл. 4.3). Знак операндів можна ігнорувати.

Таблиця 4.3. Варіанти завдань для виконання операції множення з аналізом двох розрядів

№ варіанту	Розрядність співмножників	Спосіб множення	№ варіанту	Розрядність співмножників	Спосіб множення
1	4	4	14	10	3
2	4	3	15	10	2
3	4	2	16	10	1
4	4	1	17	12	4
5	6	4	18	12	3
6	6	3	19	12	2
7	6	2	20	12	1
8	6	1	21	14	4
9	8	4	22	14	3
10	8	3	23	14	2
11	8	2	24	14	1
12	8	1	25	16	2
13	10	4	26	16	1

## 3. Порядок виконання роботи (згідно варіанту)

1. Скласти структуру пристрою множення з аналізом двох розрядів.
2. Розробити алгоритм множення з аналізом двох розрядів.

3. Завантажити JOLS-M.
4. Описати структуру пристрою множення з аналізом двох розрядів.
5. Скласти мікропрограми реалізації заданого способу множення.
6. Промоделювати роботу пристрою множення, виконуючи в кожному такті множення роздруківку чергових розрядів множника та вмісту регістра суми часткових добутоків.

#### **4. Зміст звіту**

Звіт повинен містити короткі теоретичні відомості, розроблені структуру, алгоритм (блок-схема) і мікропрограму для виконання операції множення двох операндів з аналізом двох розрядів, а також результати моделювання.

#### **5. Контрольні запитання та завдання**

1. Чому може дорівнювати частковий добуток при множенні на два розряди множника та початку множення з молодших розрядів множника? (зі старших розрядів)? При множенні на три розряди?
2. Чим визначається необхідність корекції чергової групи розрядів множника при початку множення з молодших розрядів множника? (зі старших розрядів)?
3. Чому при множенні на групу розрядів множника, починаючи з молодших розрядів, та зсувом суми часткових добутоків потрібні додаткові розряди переповнення в суматорі?
4. Чому дорівнює час множення при множенні з аналізом двох розрядів множника? Трьох розрядів множника?

### **ПРАКТИЧНА РОБОТА № 5**

#### **ВИВЧЕННЯ СПОСОБІВ ДІЛЕННЯ ЧИСЕЛ З ФІКСОВАНОЮ КОМОЮ**

- Мета:**
1. Ознайомитися з існуючими способами ділення чисел з фіксованою комою.
  2. Промоделювати роботу способів ділення на ЕОМ згідно варіанту завдання.

#### **1. Теоретичні відомості**

Загальна ідея операції ділення полягає в послідовному відніманні дільника від діленого. При цьому чергова цифра частки (результату) визначається інверсією знакового розряду залишку. В залежності від способу ділення може виконуватися відновлення залишку. Після цього виконується зсув діленого відносно дільника (вліво на 1 розряд), або дільника відносно діленого (вправо на 1 розряд).

Перед початком ділення виконується пробний такт для перевірки можливості подальшого виконання операції. Якщо на цьому етапі отримуємо додатний залишок, то подальше виконання неможливе, так як виникає

переповнення (внаслідок ділення більшого числа на менше отримуємо цілу частину, яку немає де зберігати).

Завершення операції ділення відбувається за лічильником тактів.

Для виконання операції ділення чисел використовуються два види пристроїв:

- пристрій зі зсувом залишків ліворуч (рисунок 5.1);
- пристрій зі зсувом дільника праворуч (рисунок 5.2).

На обох пристроях можливо виконувати ділення двома способами: з відновленням залишку та без токового.

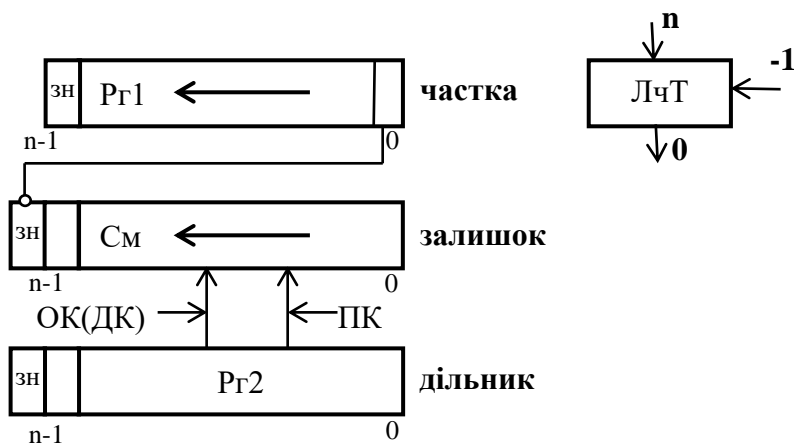


Рисунок 5.1. Структура пристрою ділення зі зсувом залишку

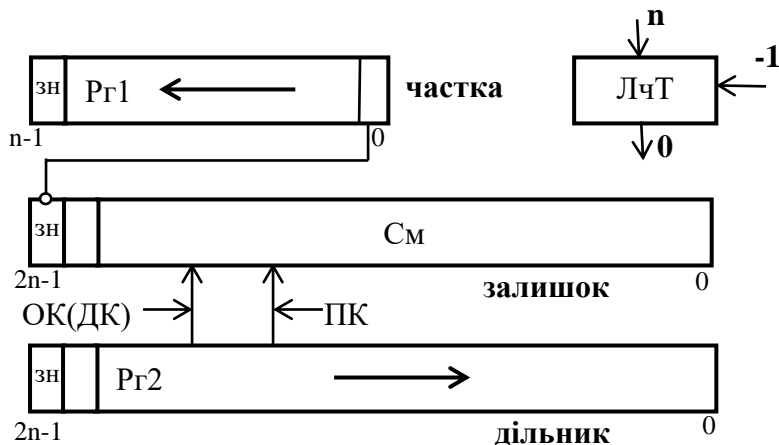


Рисунок 5.2. Структура пристрою ділення зі зсувом дільника

**Ділення з відновленням залишку.** Якщо після віднімання від діленого отримується від’ємний залишок, то його слід відновити шляхом додавання до залишку дільника.

Час виконання ділення з відновленням залишку можна оцінити за наступною формулою:

$$T_{\text{діл}} = n \times (1.5 \times t_{\text{дод}} + t_{\text{зс}}), \quad (5.1)$$

де  $n$  – кількість розрядів;

$t_{\text{дод}}$  – час додавання;

$t_{\text{зс}}$  – час виконання зсувів.

**Ділення без відновлення залишку.** Якщо після віднімання від діленого отримується від’ємний залишок, то в наступному такті до залишку буде додаватися значення дільника, якщо залишок додатний — в наступному такті від залишку буде відніматися дільник.

Час виконання ділення з відновленням залишку можна оцінити за наступною формулою:

$$T_{\text{діл}} = n \times (t_{\text{дод}} + t_{\text{зс}}), \quad (5.2)$$

де  $n$  – кількість розрядів;

$t_{\text{дод}}$  – час додавання;

$t_{\text{зс}}$  – час виконання зсувів.

На рисунках 5.3, 5.4, 5.5 та 5.6 представлені приклади виконання операції ділення різними способами (В – дільник).

Ділене: $9_{10} = 1001_2$		Дільник: $13_{10} = 1101_2 \ 0011_{\text{дк}}$	
Залишок		Частка	
0 0 1001			
<u>1 1 0011</u>	-В		
1 1 1100		0.	
<u>0 0 1101</u>	+В		
<u>0 0 1001</u>	Л1		
0 1 0010			
<u>1 1 0011</u>	-В		
<u>0 0 0101</u>	Л1	0.1	
0 0 1010			
<u>1 1 0011</u>	-В		
<u>1 1 1101</u>		0.10	
<u>0 0 1101</u>	+В		
<u>0 0 1010</u>	Л1		
0 1 0100			
<u>1 1 0011</u>	-В		
<u>0 0 0111</u>	Л1	0.101	
0 0 1110			
<u>1 1 0011</u>	-В		
<u>0 0 0001</u>		0.1011	

Рисунок 5.3. Приклад ділення зі зсувом залишку та з відновленням залишку

**Ділене:**  $9_{10}=1001_2$   
**Дільник:**  $13_{10}=1101_2$   $0011_{\text{дк}}$

Залишок		Частка
0 0 1001		
1 1 0011	-В	
<u>1 1 1100</u>	Л1	0.
1 1 1000		
0 0 1101	+В	
<u>0 0 0101</u>	Л1	0.1
0 0 1010		
1 1 0011	-В	
<u>1 1 1101</u>	Л1	0.10
1 1 1010		
0 0 1101	+В	
<u>0 0 0111</u>	Л1	0.101
0 0 1110		
1 1 0011	-В	
<u>0 0 0001</u>		0.1011

Рисунок 5.4. Приклад ділення зі зсувом залишку та без відновлення залишку

**Ділене:**  $7_{10}=0111_2$   
**Дільник:**  $12_{10}=1100_2$   $0100_{\text{дк}}$

Залишок		Частка	Дільник
0 0 01110000			
1 1 01000000	-В		$01000000_{\text{дк}}$ П1
<u>1 1 10110000</u>		0.	
0 0 11000000	+В		
<u>0 0 01110000</u>			
1 1 10100000	-В		$10100000_{\text{дк}}$ П1
<u>0 0 00010000</u>		0.1	
1 1 11010000	-В		$11010000_{\text{дк}}$ П1
<u>1 1 11100000</u>		0.10	
0 0 00110000	+В		
<u>0 0 00010000</u>			
1 1 11101000	-В		$11101000_{\text{дк}}$ П1
<u>1 1 11111000</u>		0.100	
0 0 00011000	+В		
<u>0 0 00010000</u>			
1 1 11110100	-В		$11101000_{\text{дк}}$
<u>0 0 00000100</u>		0.1001	

Рисунок 5.5. Приклад ділення зі зсувом дільника та з відновленням залишку

Залишок	Частка	Дільник
0 0 01110000		
<u>1 1 01000000</u>	-В	01000000 <sub>дк</sub> П1
1 1 10110000	0.	
<u>0 0 01100000</u>	+В	01100000 П1
0 0 00010000	0.1	
<u>1 1 11010000</u>	-В	11010000 <sub>дк</sub> П1
1 1 11100000	0.10	
<u>0 0 00011000</u>	+В	00011000 П1
1 1 11111000	0.100	
<u>0 0 00001100</u>	+В	00001100
0 0 00000100	0.1001	

Рисунок 5.6. Приклад ділення зі зсувом дільника та без відновлення залишку

Загальний алгоритм ділення чисел з відновленням залишку відображено на рисунку 5.7. Регістр Rg2 містить дільник, СМ містить ділене. На суматорі СМ відбувається залишок. Лічильник ЛчТ задає кількість ітерацій виконання операцій для формування остаточного значення частки. Результат виконання операції формується в Rg1.

## 2. Постановка задачі

Розробити структуру, алгоритм і мікропрограму пристрою для виконання операції ділення двох n-розрядних чисел з фіксованою комою згідно варіанту (табл. 5.1). Знак операндів можна ігнорувати.

## 3. Порядок виконання роботи (згідно варіанту)

1. Скласти структуру пристрою ділення.
2. Розробити алгоритм ділення.
3. Завантажити JOLS-M.
4. Описати структуру пристрою ділення.
5. Скласти мікропрограми реалізації заданого способу ділення.
6. Промодельовувати роботу пристрою ділення на ЕОМ, виконуючи друк вмісту регістру частки та суматору з залишками після кожного такту множення.

## 4. Зміст звіту

Звіт повинен містити короткі теоретичні відомості, розроблені структуру, алгоритм (блок-схема) і мікропрограму для виконання операції ділення двох операндів, а також результати моделювання.

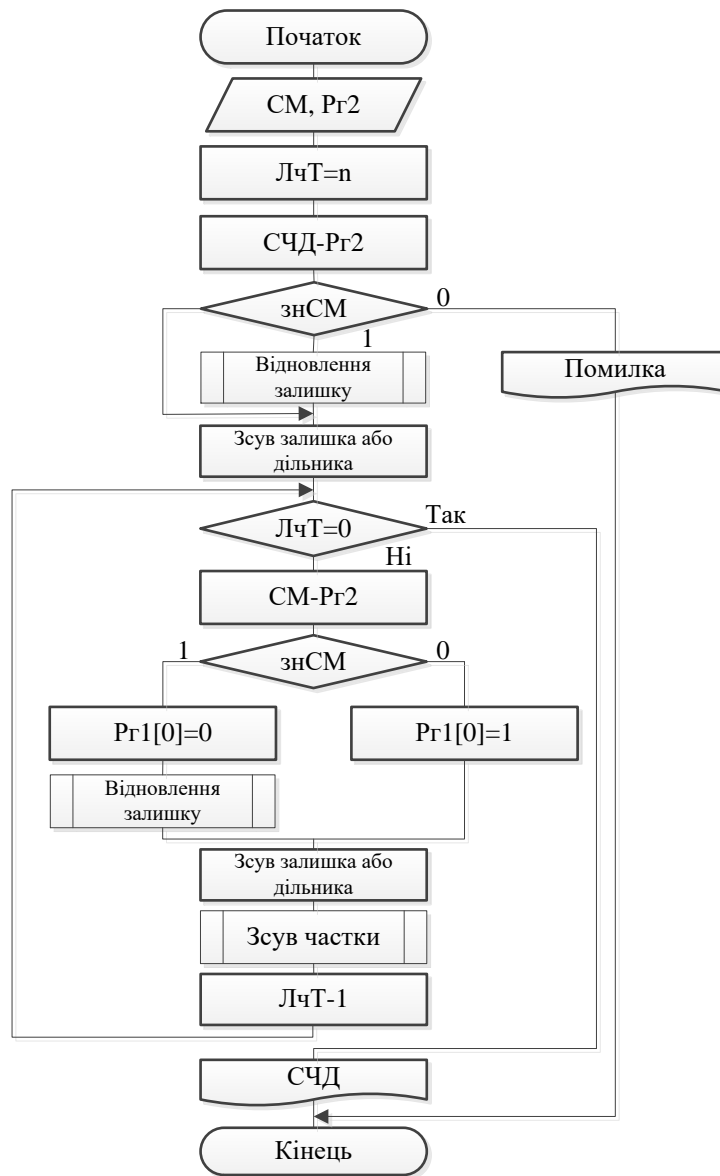


Рисунок 5.7. Загальний алгоритм ділення двох n-розрядних чисел з відновленням залишку

Таблиця 5.1. Варіанти завдань для виконання операції ділення

№ вар	Спосіб ділення	Відновлення залишку	Розрядність	№ вар	Спосіб ділення	Відновлення залишку	Розрядність
1	Зсув залишку	так	6	14	Зсув залишку	ні	9
2	Зсув залишку	ні	6	15	Зсув дільника	так	9
3	Зсув дільника	так	6	16	Зсув дільника	ні	9
4	Зсув дільника	ні	6	17	Зсув дільника	ні	5
5	Зсув залишку	так	7	18	Зсув дільника	так	5
6	Зсув залишку	ні	7	19	Зсув залишку	ні	5
7	Зсув дільника	так	7	20	Зсув залишку	так	5
8	Зсув дільника	ні	7	21	Зсув дільника	ні	10
9	Зсув залишку	так	8	22	Зсув дільника	так	10

Таблиця 5.1 (продовження)

10	Зсув залишку	ні	8	23	Зсув залишку	ні	10
11	Зсув дільника	так	8	24	Зсув залишку	так	10
12	Зсув дільника	ні	8	25	Зсув дільника	ні	11
13	Зсув залишку	так	9	26	Зсув залишку	так	11

#### 4. Зміст звіту

Звіт повинен містити розроблений алгоритм (блок-схема), роздруковану мікропрограму та результати моделювання, структурну схему пристрою операції ділення.

#### 5. Контрольні запитання та завдання

1. Які машинні способи ділення Вам відомі?
2. Як визначається наявність переповнення при діленні дробових чисел?
3. Як визначається кожна наступна цифра частки?
4. В яких випадках виконується відновлення залишку при діленні з відновленням залишків?
5. Доведіть можливість виконання ділення без відновлення залишку.
6. Як визначається знак частки?
7. Як виконується округлення при діленні?
8. Чому дорівнює час виконання операції ділення з відновленням залишку та без відновлення залишку?
9. В якій ситуації при діленні з відновленням залишку час ділення буде максимальним (мінімальним)?

### ПРАКТИЧНА РОБОТА № 6

#### ВИВЧЕННЯ ОПЕРАЦІЇ ДОДАВАННЯ ДВІЙКОВИХ ЧИСЕЛ З ПЛАВАЮЧОЮ КОМОЮ

- Мета:**
1. Ознайомитися зі способами представлення чисел з плаваючою комою.
  2. Вивчити порядок додавання чисел з плаваючою комою.
  3. Промоделювати роботу додавання чисел з плаваючою комою на ЕОМ згідно варіанту завдання.

#### 1. Теоретичні відомості

Числа з плаваючою комою представлені у вигляді мантиси і порядку:  $A = M_A \cdot 2^{P_A}$ ,  $B = M_B \cdot 2^{P_B}$ . При цьому розрядність мантиси дорівнює  $m$ , порядку  $n$ .

Для двійкових чисел операція додавання виконується за наступною послідовністю.

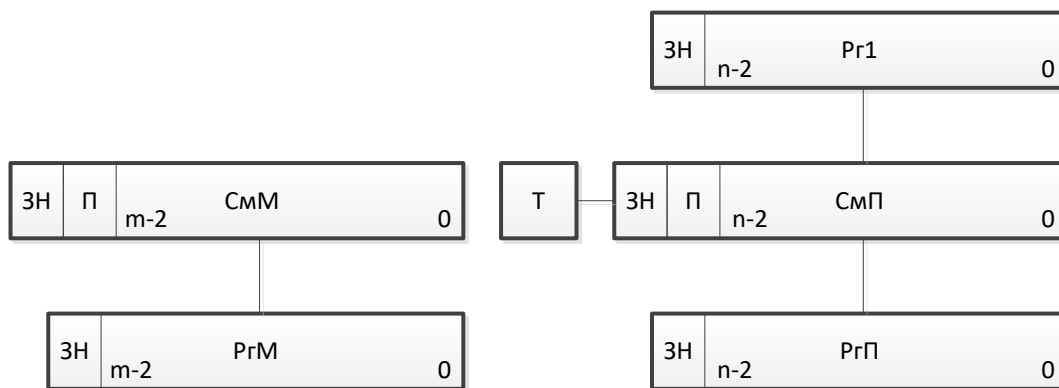
**1. Вирівнювання порядків.** Від порядку першого числа віднімається порядок другого. Мантиса числа з меншим порядком зсувається вправо на різницю порядків. Який з порядків менший визначається за знаком різниці порядків. На цьому етапі результат усієї операції можна визначити без

виконання самого додавання, коли різниця порядків перевищує розрядність мантиси. В такому випадку результатом буде число з більшим порядком.

**2. Додавання мантис.** Додавання мантис відбувається звичайним способом з урахуванням коду, в якому представлено доданки. При додаванні можливо переповнення.

**3. Нормалізація результату.** Після додавання мантис може виникнути переповнення. Його можна позбутися шляхом зсуву мантиси вправо на один розряд і збільшення порядку результату на 1. При цьому, якщо виникає переповнення при збільшенні порядку, то це буде істинне переповнення. Також після додавання мантис може знадобитися нормалізація результату. Нормалізація виконується зсувом мантиси вліво, і, відповідно, зменшенням порядку на величину зсуву. При цьому якщо відбувається від'ємне переповнення порядку, то виникає ситуація «зникнення порядку».

На рисунку 6.1 відображено структуру пристрою для додавання двох чисел з плаваючою комою. Суматори  $СмМ$  і  $СмП$  містять значення мантиси і порядку першого операнду, регістри  $РгМ$  і  $РгП$  другого. Тригер  $T$  використовуються для зберігання знаку різності порядків операндів.  $Рг1$  використовуються для тимчасового зберігання порядку першого операнду. Результат виконання операції додавання зберігається на суматорах  $СмМ$  і



СмП.

Рисунок 6.1. Структура припристрою для додавання двох чисел з плаваючою комою

На рисунку 6.2 відображено загальний алгоритм додавання двох чисел з плаваючою комою.

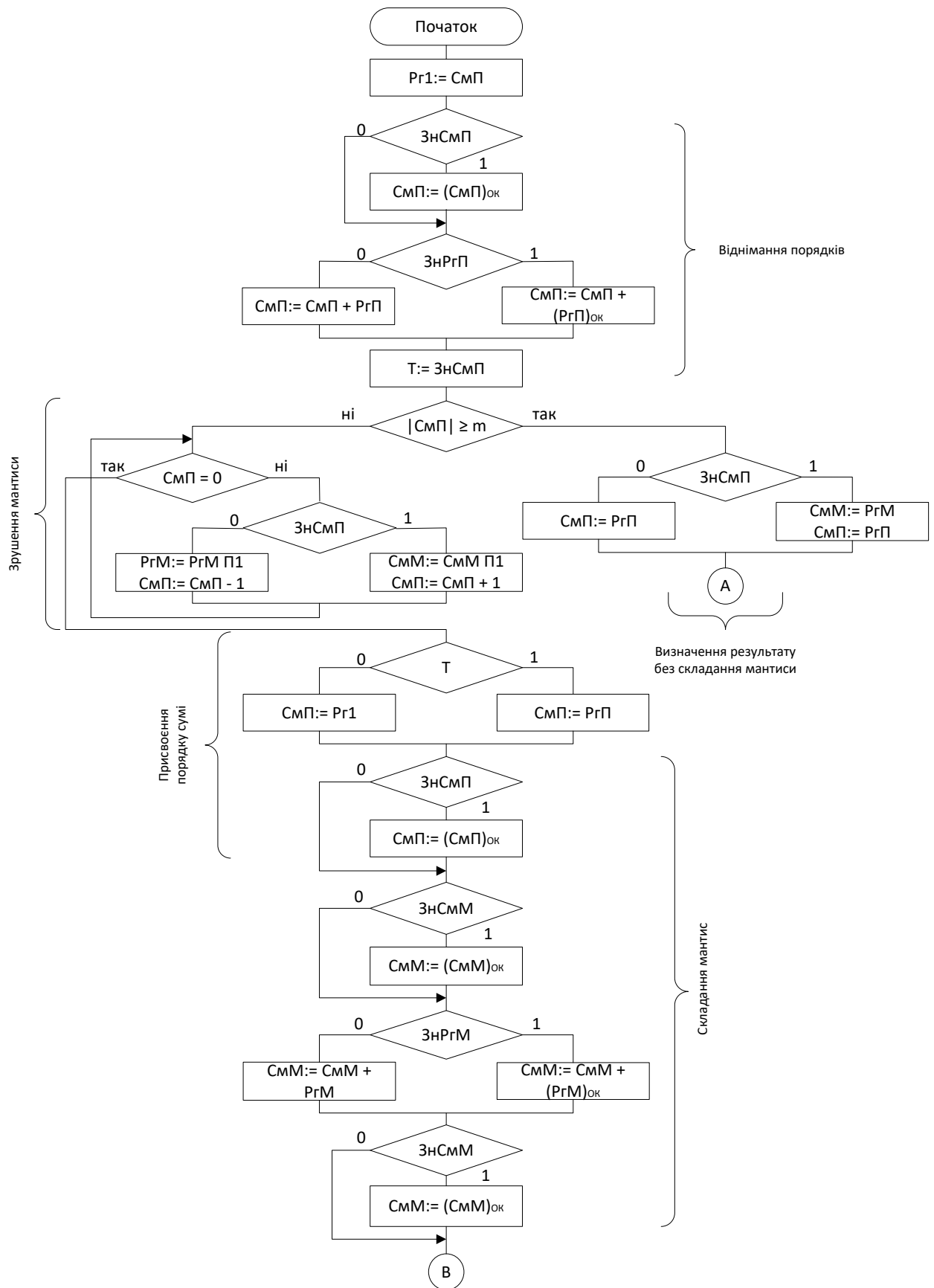


Рисунок 6.2. Загальний алгоритм додавання двох чисел з плаваючою КОМОЮ

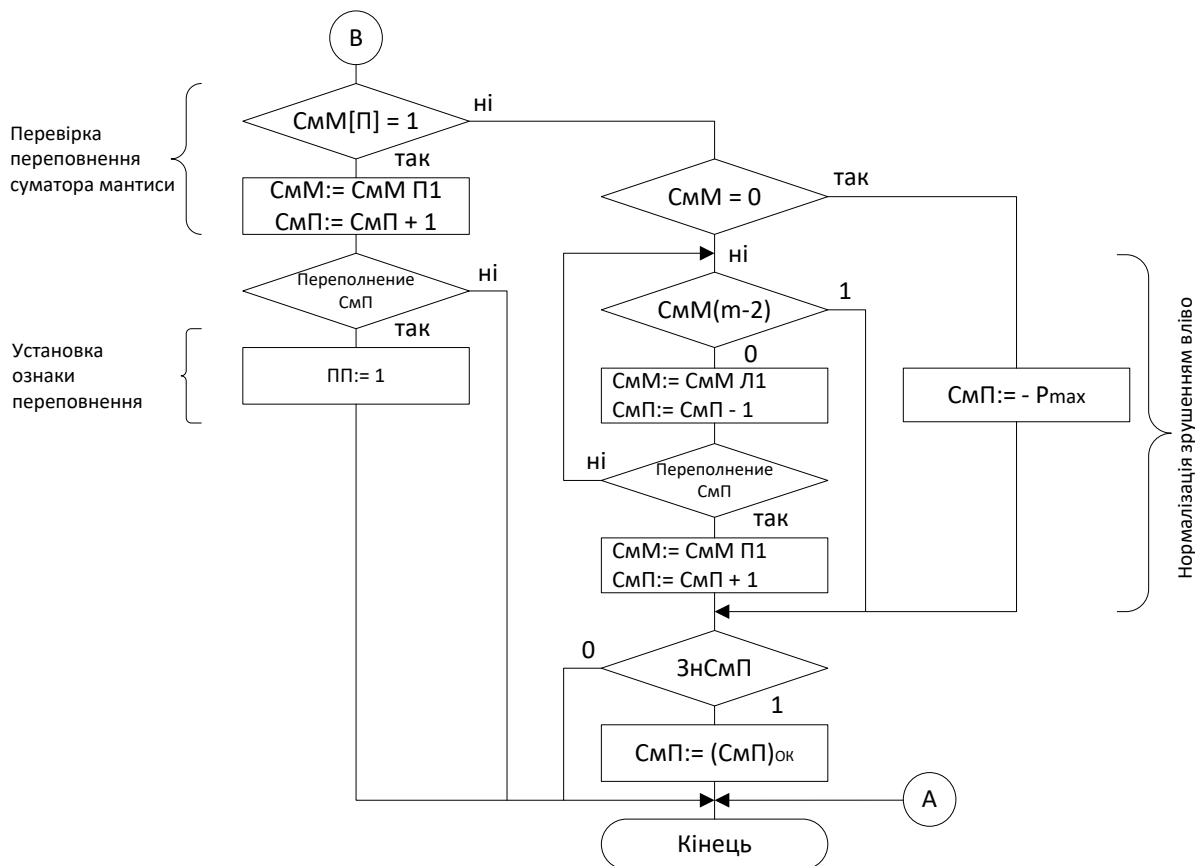


Рисунок 6.2. Загальний алгоритм додавання двох чисел з плаваючою комою (продовження)

## 2. Постановка задачі

Розробити структуру, алгоритм і мікропрограму пристрою для виконання операції додавання двох чисел з плаваючою комою згідно варіанту (табл. 6.1).

## 3. Порядок виконання роботи (згідно варіанту)

1. Скласти структуру пристрою додавання двох чисел з плаваючою комою.
2. Розробити алгоритм додавання двох чисел з плаваючою комою.
3. Завантажити JOLS-M.
4. Описати структуру пристрою додавання двох чисел з плаваючою комою.
5. Скласти мікропрограми реалізації заданого способу додавання двох чисел з плаваючою комою.
6. Виконати моделювання роботи АЛП на чотирьох прикладах:
  - 1)  $A > 0, B > 0, P_A > 0, P_B < 0$ ; 2)  $A > 0, B < 0, |A| > |B|, P_A > 0, P_B > 0$ ;
  - 3)  $A > 0, B < 0, |A| < |B|, P_A < 0, P_B > 0$ ; 4)  $A < 0, B < 0, P_A > 0, P_B > 0$ .

Числа обираються довільно, однак в кожному прикладі  $P_A \neq P_B$ . В мікропрограмі передбачити друк мантиси та порядку після кожного такту вирівнювання порядків та нормалізації.

Таблиця 6.1. Варіанти завдань для виконання операції додавання чисел з плаваючою комою

№ варіанту	Розрядність		Код, що застосовується
	мантиси	порядку	
1	10	5	ЗК
2	11	6	ДК
3	12	7	ЗК
4	13	8	ДК
5	14	5	ЗК
6	15	6	ДК
7	16	7	ЗК
8	17	8	ДК
9	18	5	ЗК
10	19	6	ДК
11	20	7	ЗК
12	10	8	ДК
13	11	5	ЗК
14	12	6	ДК
15	13	7	ЗК
16	14	8	ДК
17	15	5	ЗК
18	16	6	ДК
19	17	7	ЗК
20	18	8	ДК
21	19	5	ЗК
22	20	6	ДК
23	15	7	ЗК
24	16	5	ДК
25	17	6	ЗК

Примітка: ЗК – зворотний код; ДК – додатковий код..

#### 4. Зміст звіту

Звіт повинен містити короткі теоретичні відомості, розроблені структуру, алгоритм (блок-схема) і мікропрограму для виконання операції додавання чисел з плаваючою комою, а також результати моделювання

#### 5. Контрольні запитання та завдання

1. Яка основна мета досягається використанням представлення чисел з плаваючою комою?
2. Що характеризує розрядність порядку та розрядність мантиси?
3. Назвіть основні етапи додавання чисел з плаваючою комою.
4. Як відбувається порівняння порядків?
5. Порядок якого доданку є порядком результату?

6. В яких ситуаціях результат може бути отриманий без виконання додавання (тобто результат дорівнює одному з доданків)?
7. В якій ситуації може виникнути справжнє переповнення? Що є ознакою переповнення?
8. Які дії виконуються, якщо при додаванні мантис виникло переповнення?
9. Чому результат операції, як правило, нормалізується?
10. Коли може відбутися втрата порядку?
11. Які переваги має представлення чисел з плаваючою комою зі зміщеним порядком (характеристика)?
12. Як представляється результат з нульовою мантисою?

## СПИСОК ЛІТЕРАТУРИ

1. Харріс Д.М. Цифрова схемотехніка та архітектура комп'ютера [Текст] / Д.М.Харріс. – Морган Кауфман, 2013. – 1662 с.
2. Матвієнко М.П. Архітектура комп'ютерів: Навчальний посібник [Текст] / Матвієнко М.П., Розен В.П., Закладний О.М. – К.: Ліра-К, 2019. – 264 с.
3. Карачка А. Ф., Дудко О. І. Архітектура комп'ютерів: Навч. посіб. [Текст] / За ред. А. О. Саченка. – Тернопіль: Економічна думка, 2009. – 180 с.
4. Тарарака В.Д. Архітектура комп'ютерних систем: навчальний посібник [Текст] / В.Д.Тарарака – Житомир : ЖДТУ, 2018. – 383 с.
5. Антоненко О. В., Бардус І. О. Архітектура комп'ютера та конфігурування комп'ютерних систем (на основі фундаменталізованого підходу) : навч. посіб. [Текст] / О. В. Антоненко, І. О. Бардус – Бердянськ : БДПУ, 2018. – 292с.
6. Голотенко О.С. Архітектура комп'ютерних систем: конспект лекцій для студентів усіх форм навчання з курсу «Архітектура комп'ютерних систем» [Текст] / О.С. Голотенко– Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2016. – 120 с.
7. Linda Null, Julia Lobur. Essentials of Computer Organization and Architecture [Текст] / Jones & Bartlett Learning, 2018. – 744 с.
8. Paul D. Crutcher, Neeraj Kumar Singh, Peter Tiegs. Essential Computer Science [Текст] / Apress, 2021. – 290 с.
9. William Stallings. Computer Organization And Architecture [Текст] / PEARSON, 2015. – 864 с.
10. СДН «Лідер». Архітектура комп'ютерів. Процесори (<https://lider.diit.edu.ua/course/view.php?id=1792>).

Навчально-методичне видання

**Єгоров Олег Йосипович**  
**Дзюба Володимир Володимирович**  
**Івін Павло Вікторович**

## **АРХІТЕКТУРА КОМП'ЮТЕРІВ**

Методичні рекомендації до практичних робіт

Редактор А. В. Безверхня  
Комп'ютерна верстка В. В. Бердо

Формат 60x84 <sup>1</sup>/<sub>16</sub>. Ум. друк. арк. 2,15. Обл.-вид. арк. 2,17.  
Тираж пр. Зам. № 2

Український державний університет  
науки і технологій  
Свідоцтво суб'єкта видавничої справи ДК № 1315 від 31.03.2003

Адреса видавця та дільниці оперативної поліграфії:  
вул. Лазаряна, 2, Дніпро, 49010