МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Український державний університет науки і технологій

Кафедра «Електронні обчислювальні машини»

АРХІТЕКТУРА КОМП'ЮТЕРНИХ СИСТЕМ
(ЧАСТИНА 1)

Навчально-методичні рекомендації
для виконання практичних робіт

Електронне видання

ДНІПРО
2024

УДК 004.2
А 87

Упорядники:
А.А. Косолапов, О.Й. Єгоров, Л.С. Тимошенко

Електронне видання

Схвалено Групою забезпечення якості освітньої програми
125 «Кібербезпека»
Протокол № 2 від 12.01.2024

А87 Архітектура комп’ютерних систем : навчально-методичні рекомендації для виконання практичних робіт / упоряд.: А. А. Косолапов, О. Й. Єгоров, Л. С. Тимошенко ; Укр. держ. ун-т науки і технологій. – Електрон. вид. – Дніпро : УДУНТ, 2024. – Ч. 1. – 30 с.

Навчально-методичні рекомендації призначено студентам 3-го курсу спеціальності 125 «Кібербезпека» для підготовки до виконання практичних робіт з дисципліни «Архітектура комп’ютерних систем».

Навчально-методичні рекомендації містять основні теоретичні положення щодо засвоєння матеріалу, вказівки до виконання практичних робіт, вимоги до аналізу результатів та оформлення робіт.

Зміст

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 функціювання 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 і систем;

– вміти застосовувати знання для ідентифікації, формулювання і розв’язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей;

– вміти застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп’ютерних систем для вирішення технічних задач спеціальності.

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:

– VAR
– READ
– OPERATION
– PRINT
– GOTO
– IF
– END
– REMAR

Перераховані мікрокоманди допускають використання міток.

Мікрокоманда VAR

Мікрокоманда **VAR** призначена для опису використовуваних елементів пристрою (регістрів, суматорів тощо). Це декларативна мікрокоманда із наступним синтаксисом:

VAR елемент (розрядність), …

де **елемент** – назва елемента пристрою;
розрядність – розрядність елемента пристрою.

Мікрокоманда **VAR** повинна бути першою в мікропрограмі и виконувати опис всіх елементів пристрою.

Мікрокоманда READ

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

{ mmm } READ [операнд1],

де **{ mmm }** – мітка (1...999);

операнд1 – будь-який з перерахованих нижче варіантів:

– pXX – регістр з номером XX (0..39);

– пXXX – комірка пам’яті з адресою XXX (0..254);

– р(pXX) – регістр з адресою, що зберігається в регістрі XX;

– п(пXXX) – комірка пам’яті з адресою, що зберігається в комірці з адресою XXX;

– р(пXXX) – регістр, адреса якого зберігається в комірці пам’яті;

– п(pXX) – комірка пам’яті, адреса якої зберігається в регістрі.

Мікрокоманда OPERATION

Мікрокоманда **OPERATION** призначена для виконання різних дій над регістрами, 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 регістр або елемент пам'яті, окремий біт або група розрядів, константи в двійковій, десятковій або шістнадцятковій системі числення, а також регістри 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:

– pXX;

– pXX(NN);

– pXX(NN:nn);

– пXXX;

– пXXX(NN);

– пXX(NN:nn);

– р(рXX);
– р(пXXX);
– п(пXXX);
– п(рXX),
де XX – номер регістра (0..39);
XXX – номер елементу 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 елементу пам'яті (0..254);

б) **{mmm} ПЕЧАТЬ «будь-який текст»**

де **{mmm} –** мітка (1..999).

Мікрокоманда GOTO

Мікрокоманда **GOTO** 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 GOTO** може бути вказано безпосередньо значення мітки, регістру або елементу пам'яті, що адресується побічно.

{mmm} GOTO [вираз],

де **{mmm}** – мітка (1..999);

вираз – будь-який з перерахованих нижче варіантів:

– ddd – мітка для переходу (1..999);
– pXX;
– пXXX;
– р(pXX);
– р(пXXX);
– п(пXXX);
– п(pXX),

де XX – номер регістра (0..39);
XXX – номер елементу 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:

– pXX;
– pXX(NN);
– pXX(NN:nn);
– пXXX;
– пXXX(NN);
– пXX(NN:nn);
– р(pXX);
– р(пXXX);
– п(пXXX);

– п(pXX),
де XX – номер регістра (0..39);
XXX – номер елементу 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*2^31 .. 2*2^31);
– $hhh – шістнадцятирична константа (0.. FFFFFFFFh);
– #bbb – двійкова константа (0 .. 1111 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:

– READ;
– OPERATION;
– PRINT;
– GOTO;
– 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 «І» 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 («-»).

Якщо необхідно представити від'ємне число при використанні зворотного коду, то **у** знаковий розряд записується «1», а всі значущі розряди інвертуються. Якщо число необхідно представити в додатковому коді, то крім інверсії значущих розрядів треба додати до значущих розрядів «1». На рисунку 2.1 представлено число -5 в прямому, зворотному та додатковому кодах.

ПК: $-5_{10}=1.0101_2$ ЗК: $-5_{10}=1.1010_2$ ДК: $-5_{10}=1.1011_2$

Рисунок 2.1. Представлення числа -5 в ПК, ЗК і ДК

В ЗК, якщо після здійснення операції додавання чисел відбувається переповнення розрядної сітки (є перенос із старшого розряду), то слід виконати додавання до результату одиниці (додавання з циклічним перенесенням). Приклад виконання операції додавання з використанням зворотного коду зображено на рисунку 2.2.

ПК: $-5_{10}=1.0101_2$ $-4_{10}=1.0100_2$ (-5)+(-4)=?
ЗК: $-5_{10}=1.1010_2$ $-410=1.1011_2$

```
      1.1010
      1.1011
  1← 1.0101
           1
  ЗК: 1.0110
  ПК: 1.1001
```

Рисунок 2.2. Процес виконання операції додавання з використанням зворотного коду

1.2. Модифікований зворотний (МЗК) та модифікований додатковий (МДК) коди використовуються для визначення факту переповнення. Для цього використовується ще один додатковий розряд. Якщо число від'ємне, то додатковий та знаковий розряди дорівнюють 1 («11»), якщо число додатне – 0 («00»). При цьому, якщо після виконання операції додавання додатковий та знаковий розряди не дорівнюють один одному («10» або «01»), то виникло переповнення.

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.

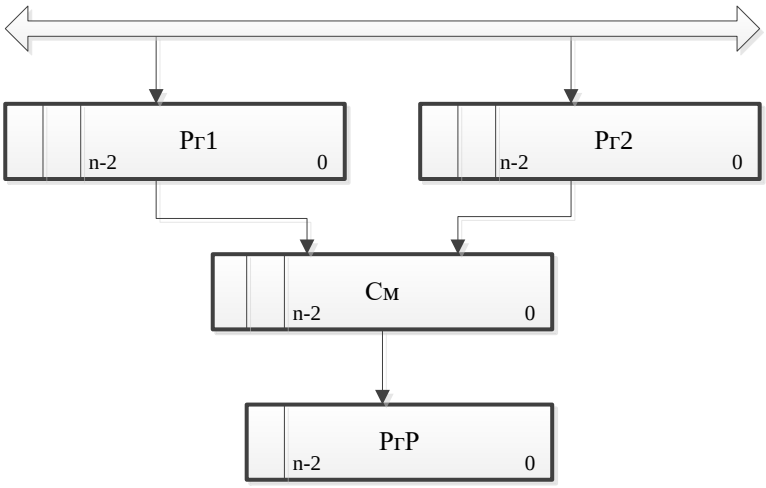

Рисунок 2.4. Структура пристрою для додавання двох n-розрядних чисел з фіксованою комою

Регістри Рг1 і Рг2 містять два вихідних операнда, старший біт яких є знаковим. На суматорі См відбувається складання чисел. У цьому слід врахувати, що суматор є накопичувальним. Результат виконання операції заноситься до регістру результату РгР.

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

2. Постанова задачі

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

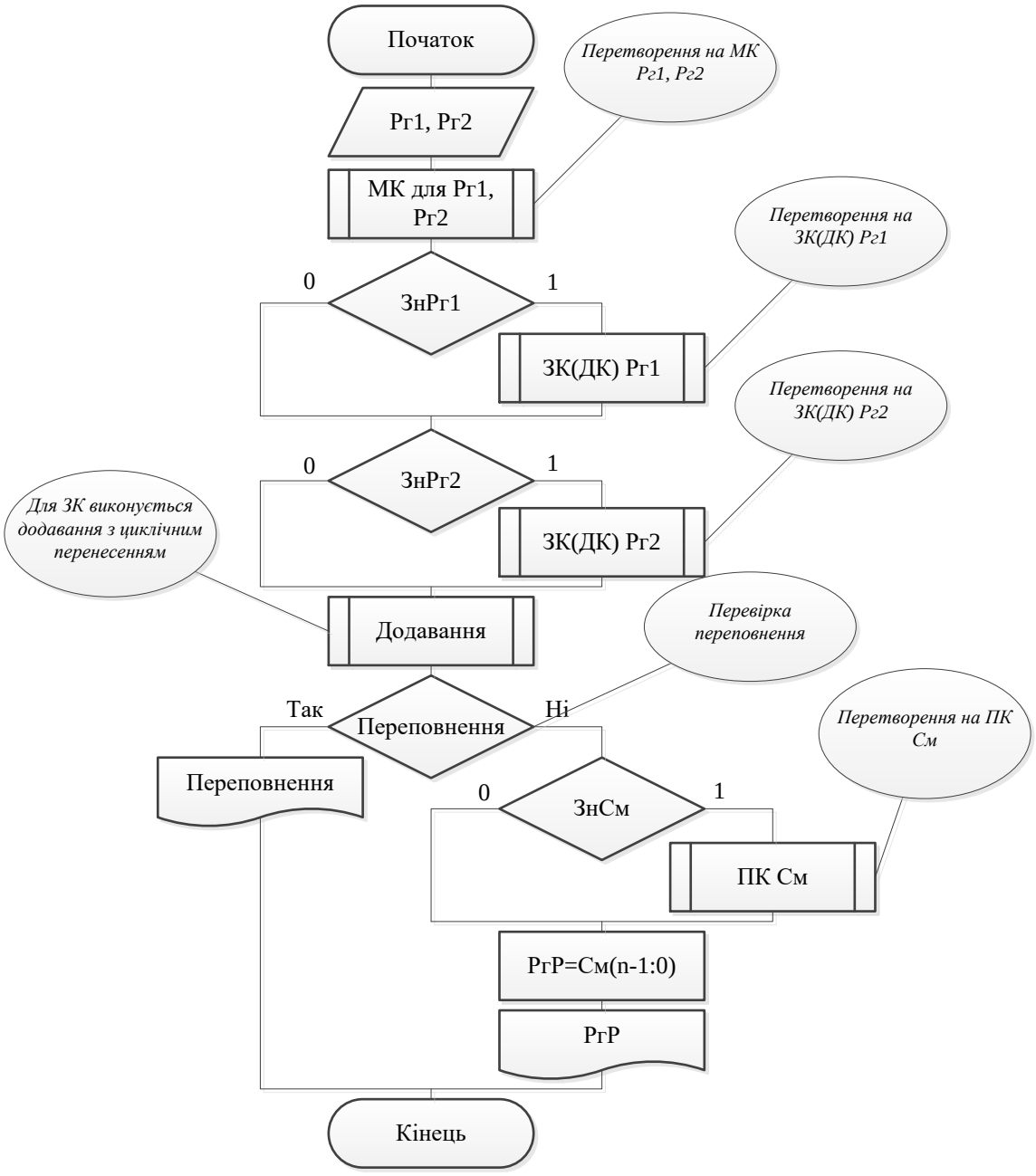

Рисунок 2.5. Загальний алгоритм додавання двох n-розрядних чисел з фіксованою комою

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

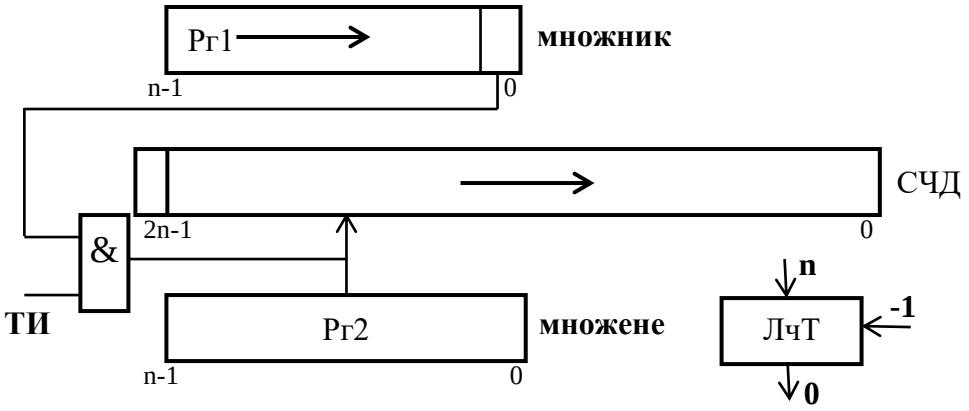

Рисунок 3.1. Структура пристрою множення з молодших розрядів множника та зсув суми часткових добутків

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.

Множник: $5_{10}=0101_2$
Множене: $13_{10}=1101_2$

```
СЧД                    Множник
0 00000000             0101 П1
0 1101
0 11010000 П1
0 01101000             0010 П1
0 0000
0 01101000 П1
0 00110100             0001 П1
0 1101
1 00000100 П1
0 10000010             0000 П1
0 0000
0 10000010 П1
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 множене розташовується у регістрі з подвійною розрядністю починаючи з молодших

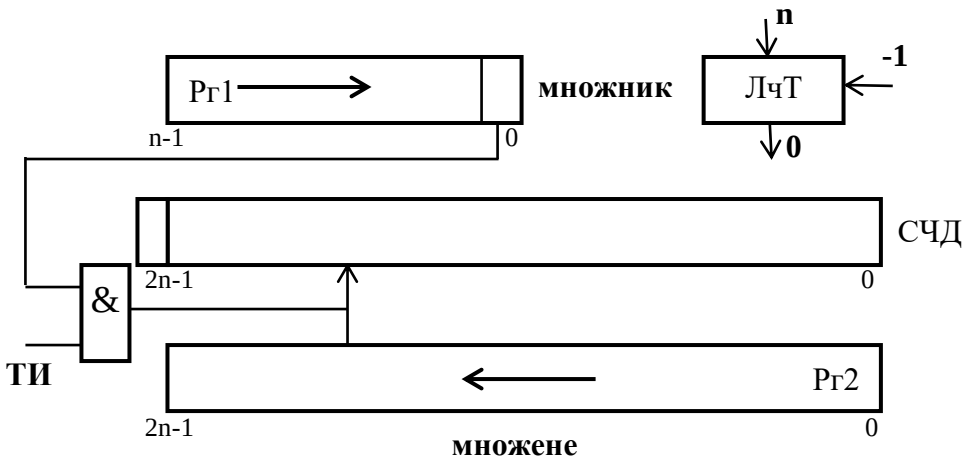

Рисунок 3.3. Структура пристрою множення з молодших розрядів множника та зсув множеного

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

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.

Множник: $9_{10}=1001_2$
Множене: $11_{10}=1011_2$

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

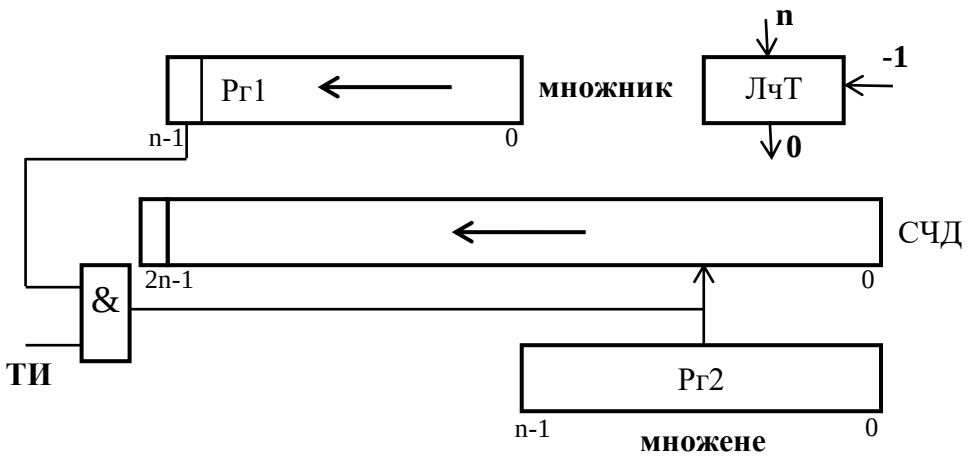

Рисунок 3.5. Структура пристрою множення зі старших розрядів множника та зсув суми часткових добутків

На рисунку 3.6 представлений приклад виконання операції множення даним способом.

Множник: $5_{10}=0101_2$
Множене: $13_{10}=1101_2$

```
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

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

Початок множення зі старших розрядів множника та зсув множеного (рисунок 3.7, спосіб №4). Перед початком множене 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.

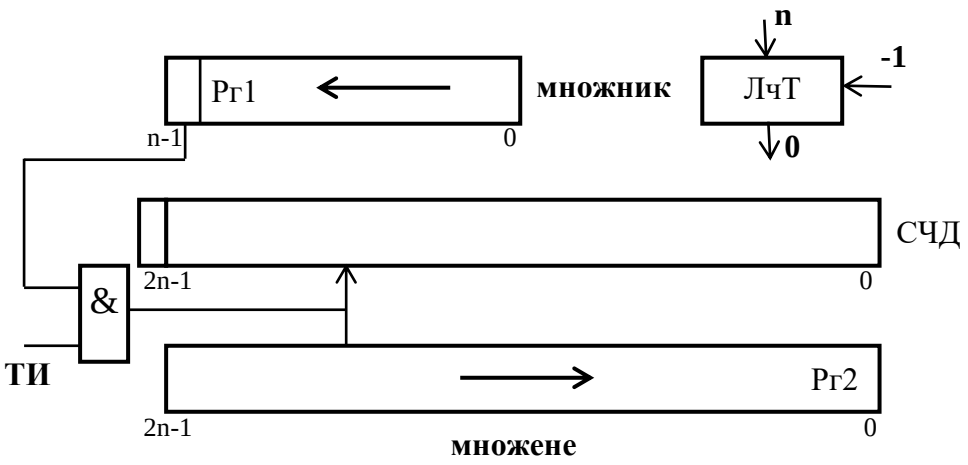

Рисунок 3.7. Структура пристрою множення зі старших розрядів множника та зсув множеного

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.*

Множник: $9_{10}=1001_2$
Множене: $11_{10}=1011_2$

```
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:

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

де *n* – кількість розрядів;

$t_{дод}$ – час додавання;

$t_{зс}$ – час виконання зсувів.

Загальний алгоритм множення відображено на рисунку 3.9. Регістри Рг1 і Рг2 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. Постанова 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.

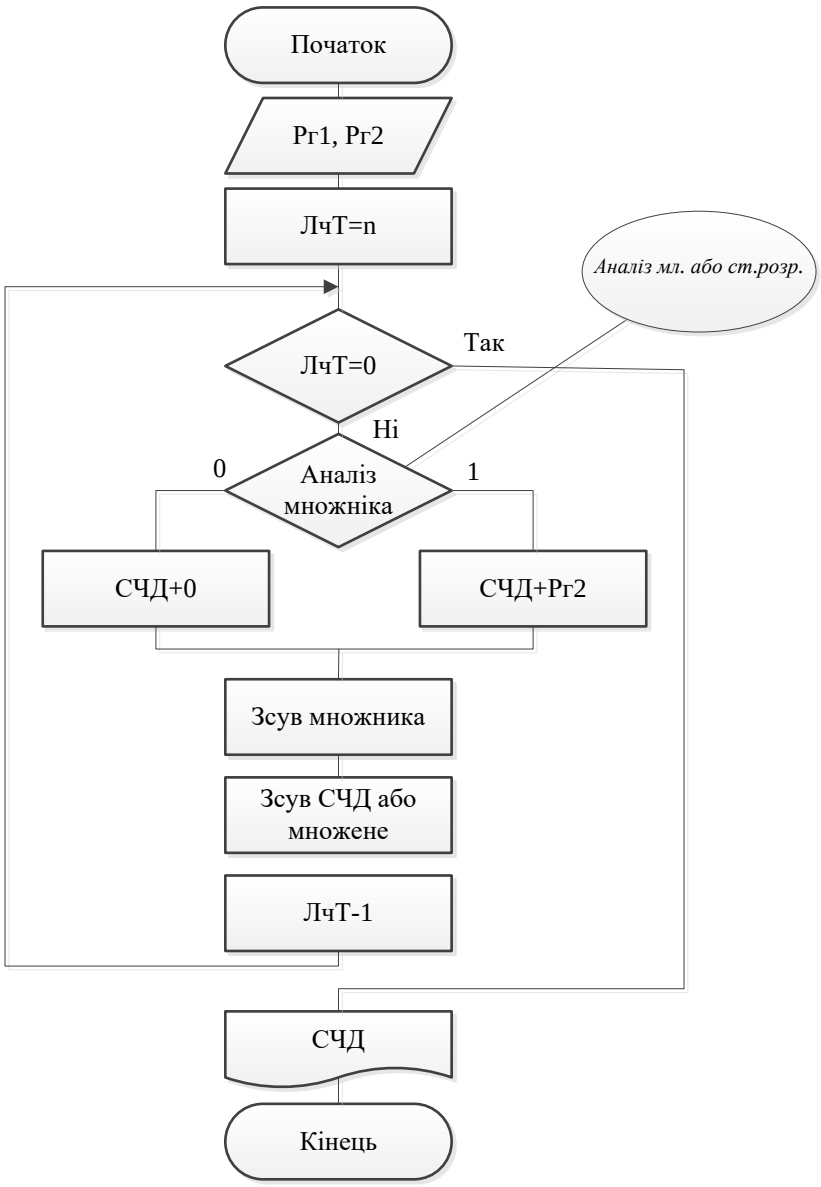

Рисунок 3.9. Загальний алгоритм множення двох n-розрядних чисел з фіксованою комою

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

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 добутків; 2 – початок множення з молодших розрядів множника та зсув множеного; 3 – початок множення зі старших розрядів множника та зсув суми часткових добутків; 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

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 4.1);
- пристрій зі зсувом дільника праворуч (рисунок 4.2).

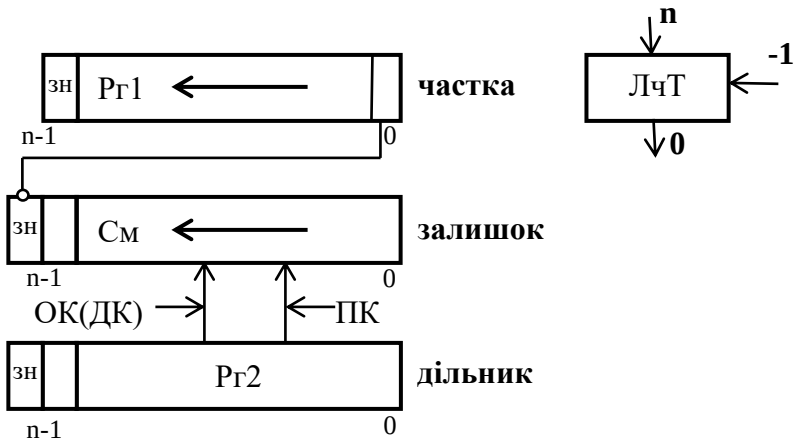

Рисунок 4.1. Структура пристрою ділення зі зсувом залишку

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

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:

$$T_{діл} = n\times(1.5\times t_{дод}+t_{зс}), \qquad (4.1)$$

де n – кількість розрядів;

$t_{дод}$ – час додавання;

$t_{зс}$ – час виконання зсувів.

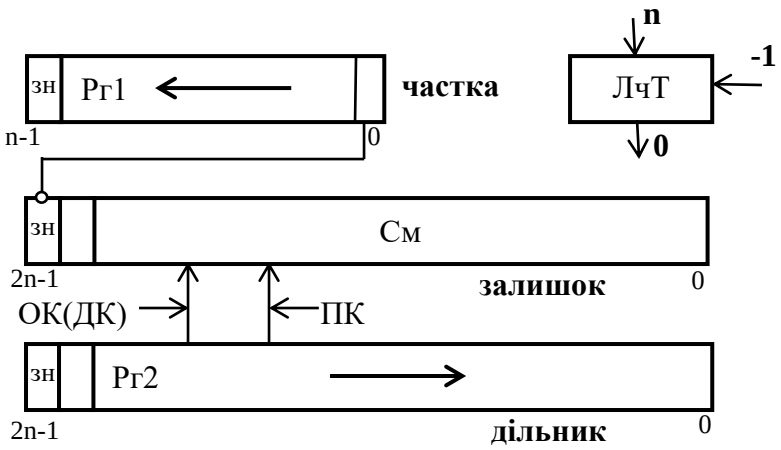

Рисунок 4.2. Структура пристрою ділення зі зсувом дільника

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.

Двійкова система числення дозволяє виконувати ділення без відновлення залишків. Розглянемо ділення із зсувом залишку вліво в кожному такті.

Припустимо, що залишок Ri, отриманий на і–му кроці ділення позитивний $R_i > 0$. Для переходу до отримання наступної (і+1)–й цифри частки отриманий позитивний залишок зсувається вліво на один розряд і з нього віднімається дільник В. Припустимо, що на цьому етапі отримано негативний залишок R_{i+1}:

$$R_{i+1}=2\cdot R_i - B < 0 \qquad (4.2)$$

При розподілі з відновленням залишку для продовження поділу необхідно повернутися до попереднього позитивного залишку (відновити $2\cdot R_i$), зрушити його вліво і зробити віднімання дільника для отримання наступної (і+2)–й цифри частки:

$$2\cdot Ri=Ri+1+B;\ Ri+2=2\cdot 2\cdot Ri-B \qquad (4.3)$$

Замінимо в останньому виразі $2\cdot R_i$ на $(R_{i+1}+B)$ отримаємо:

$$R_{i+2}=2\cdot(R_{i+1}+B)-B=2\cdot R_{i+1}+2\cdot B-B=2\cdot R_{i+1}+B \qquad (4.4)$$

Останній вираз визначає правило продовження операції ділення при отриманні від'ємного залишку. Якщо на черговому кроці отримано негативний залишок для продовження ділення необхідно зрушити негативний залишок вліво і на черговому кроці додати дільник до зрушеному негативного залишку. Тим самим поєднується відновлення позитивного залишку і віднімання дільника для отримання наступної цифри частки.

Для другого способу розподілу із зсувом дільника в кожному такті ділення також можливе ділення без відновлення залишку. Так само як і для попереднього способу припускаємо, що залишок R_i позитивний. Для

отримання чергового (i+1)–го залишку дільник, зрушений на 1 розрядів вправо, зсувається ще на один розряд і віднімається із залишку, утворюючи (i+1)–й залишок. Припустимо, що на цьому етапі отримано негативний залишок R_{i+1}:

$$R_{i+1}=R_i–1/2^{i+1}B <0 \quad (4.5)$$

При розподілі з відновленням залишку виконується відновлення позитивного залишку R_i:

$$R_i=R_{i+1}+1/2^{i+1}B \quad (4.6)$$

Далі дільник B зсувається ще на один розряд вправо і віднімається з відновленого залишку R_i:

$$R_{i+2}=R_i–1/(2\cdot2^{i+1})\ B \quad (4.7)$$

Замінивши R_i його значенням через $R_{i+1,}$ отримаємо

$$R_{i+2}=R_{i+1}+1/(2^{i+1})\ B–1/(2\cdot2^{i+1})\ B=R_{i+1}+1/(2\cdot2^{i+2})\ B \quad (4.8)$$

Тобто також як і для поділу першим способом, відновлення залишку не проводиться, а виконується зрушення дільника вправо і його додаток до негативного залишку.

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

$$T_{діл} = n\times(t_{дод}+t_{зс}), \quad (4.9)$$

де *n* – кількість розрядів;

$t_{дод}$ – час додавання;

$t_{зс}$ – час виконання зсувів.

На рисунках 4.3, 4.4, 4.5 та 4.6 представлені приклади виконання операції ділення різними способами (B – дільник).

Загальний алгоритм ділення чисел з відновленням залишку відображено на рисунку 4.7. Регістр Рг2 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 Рг1.

2. Постанова задачі

Розробити структуру, алгоритм і мікропрограму пристрою для виконання операції ділення двох n-розрядних чисел з фіксованою комою згідно варіанту (табл. 4.1). Знак операндів можна ігнорувати.

3. Порядок виконання роботи (згідно варіанту)

1. Скласти структуру пристрою ділення.
2. Розробити алгоритм ділення.
3. Завантажити JOLS-M.

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

Залишок		Частка
0 0 1001		
1 1 0011	-В	
1 1 1100		0.
0 0 1101	+В	
0 0 1001	Л1	
0 1 0010		
1 1 0011	-В	
0 0 0101	Л1	0.1
0 0 1010		
1 1 0011	-В	
1 1 1101		0.10
0 0 1101	+В	
0 0 1010	Л1	
0 1 0100		
1 1 0011	-В	
0 0 0111	Л1	0.101
0 0 1110		
1 1 0011	-В	
0 0 0001		0.1011

Рисунок 4.3. Приклад ділення зі зсувом залишку та з відновленням залишку

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

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

Рисунок 4.4. Приклад ділення зі зсувом залишку та без відновлення залишку

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

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

Рисунок 4.5. Приклад ділення зі зсувом дільника та з відновленням залишку

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

Залишок		Частка	Дільник
0 0 01110000			
1 1 01000000	-B		$01000000_{дк}$ П1
1 1 10110000		0.	
0 0 01100000	+B		01100000 П1
0 0 00010000		0.1	
1 1 11010000	-B		$11010000_{дк}$ П1
1 1 11100000		0.10	
0 0 00011000	+B		00011000 П1
1 1 11111000		0.100	
0 0 00001100	+B		00001100
0 0 00000100		0.1001	

Рисунок 4.6. Приклад ділення зі зсувом дільника та без відновлення залишку

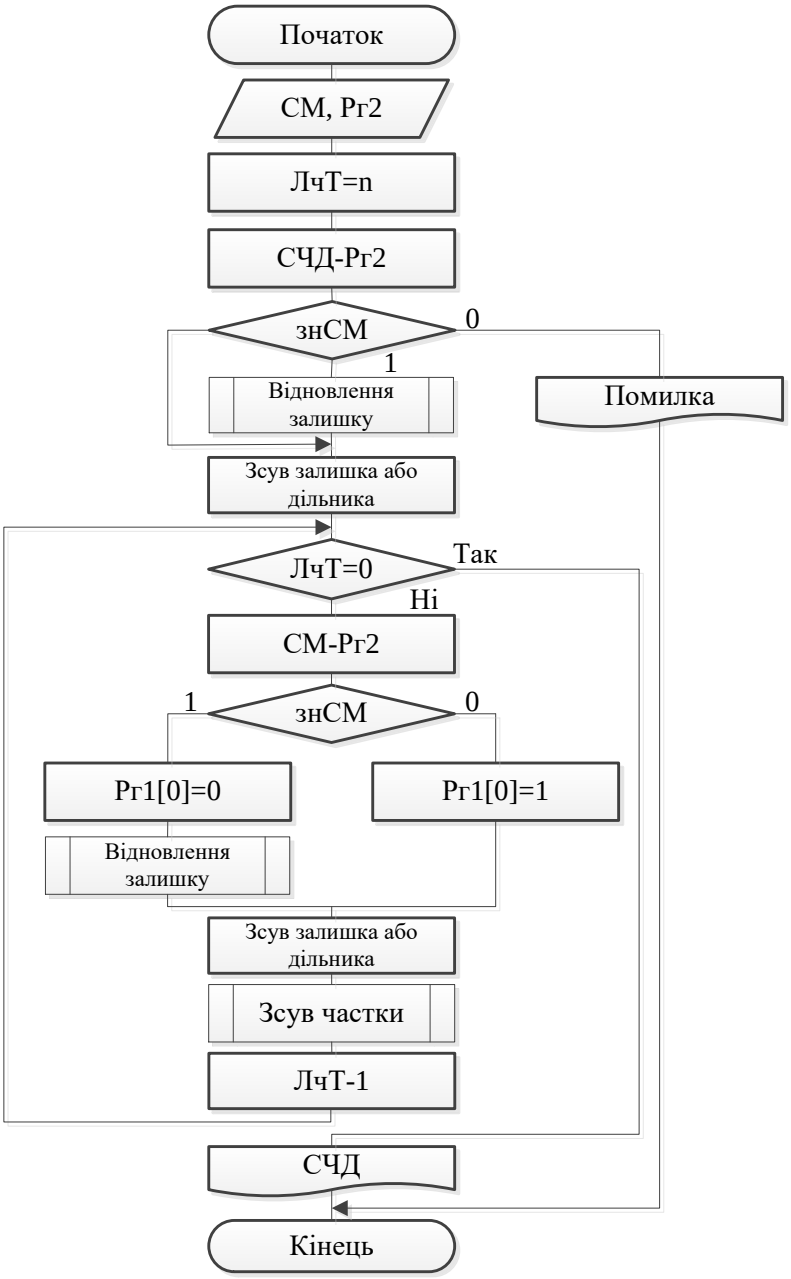

Рисунок 4.7. Загальний алгоритм ділення двох n-розрядних чисел з відновленням залишку

4. Описати структуру пристрою ділення.

5. Скласти мікропрограми реалізації заданого способу ділення.

6. Промоделювати роботу пристрою ділення на ЕОМ, виконуючи друк вмісту регістру частки та суматору з залишками після кожного такту ділення.

Таблиця 4.1. 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
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)?

СПИСОК ЛІТЕРАТУРИ

1. Harris D. M., Harris S. L. Digital Design and Computer Architecture. 2nd ed. Burlington : Morgan Kaufmann Publishers, 2013. 690 p.
2. Матвієнко М. П., Розен В. П., Закладний О. М. Архітектура комп’ютера : навч. посіб. Київ : Вид-во Ліра-К, 2016. 264 с.
3. Карачка А. Ф., Дудко О. І. Архітектура комп’ютерів : навч. посіб. / за ред. А. О. Саченка. Тернопіль : Економічна думка, 2009. 180 с.
4. Тарарака В. Д. Архітектура комп'ютерних систем : навч. посіб. 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. Бердянськ : БДПУ, 2018. 292 с.
6. Архітектура комп’ютерних систем : конспект лекцій для студентів усіх форм навчання з курсу «Архітектура комп’ютер. систем» / уклад. О. С. Голотенко. Тернопіль : Вид-во ТНТУ ім. Івана Пулюя, 2016. 120 с.
7. Null L., Lobur J. Essentials of Computer Organization and Architecture. Burlington : Jones & Bartlett Learning, 2018. 744 с.
8. Crutcher P. D., Singh N. K., Tiegs P. Essential Computer Science. New York City : Apress, 2021. 290 с.
9. Stallings W. Computer organization and architecture. Noida : Pearson India, 2016. 864 с.
10. СДН «Лідер». Архітектура комп’ютерних систем. *Український державний університет науки і технологій.* URL: https://lider.ust.edu.ua/course/view.php?id=862 (дата звернення: 09.01.2024).

Навчально-методичне видання

Косолапов Анатолій Аркадійович,
Єгоров Олег Йосипович,
Тимошенко Людмила Сергіївна

АРХІТЕКТУРА КОМП'ЮТЕРНИХ СИСТЕМ (ЧАСТИНА 1)

Навчально-методичні рекомендації до виконання практичних робіт

Електронне видання

Експертний висновок склав зав. кафедри ЕОМ, д-р техн. наук, проф. Жуковицький І. В.

Зареєстровано НМВ УДУНТ (№ 69 від 22.02.2024)

Формат 60x84 1/16. Ум. друк. арк. 1,74 . Обл.-вид. арк. 0,85.
Зам. № 11

Видавець: Український державний університет науки і технологій
вул. Лазаряна, 2, ауд. 2216, м. Дніпро, 49010.
Свідоцтво суб'єкта видавничої справи ДК № 7709 від 14.12.2022

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