ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
МІНІСТЕРСТВА ОСВІТИ І НАУКИ УКРАЇНИ
Кафедра комп'ютерних інтелектуальних систем та мереж

БОГОВИК Олександр Сергійович

ДИПЛОМНА РОБОТА МАГІСТРА

ДОСЛІДЖЕННЯ СИСТЕМИ ПАРАЛЕЛЬНОГО АРИФМЕТИЧНОГО ЗСУВУ

Спеціальність 123 – Комп`ютерна інженерія
Спеціалізація – Комп'ютерні системи та мережі

Керівник: Дрозд Олександр Валентинович,
доктор технічних наук, професор

Одеса – 2021

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ

Інститут комп'ютерних систем Кафедра КІСМ

Спеціальність Комп'ютерні інтелектуальні системи та мережі

ЗАТВЕРДЖУЮ:
Зав. кафедри КІСМ
_____
“ ” 2021 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТОВІ

Боговику Олександру Сергійовичу

1. Тема роботи “ Дослідження системи паралельного арифметичного зсуву ”
затверджена наказом по університету від “ ” 2021 р. № -
2. Термін здачі студентом закінченої роботи 05.12.2021
3. Вихідні дані до роботи у розділі “Завдання на проведення досліджень”
4. Зміст розрахунково-пояснювальної записки (перелік питань, що належить розробити)
1. 1 Огляд рішень з підвищення достовірності результатів, що обчислюються у сучасних арифметичних пристроях
2. Завдання на проведення досліджень
3. Аналіз задачі побудови моделі зсувача
4 Розробка програмної моделі арифметичного зсувача
5. Результати проведених досліджень
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
1) Розвиток персональних комп’ютерів
2) Асиметрія в роботі цифрових схем
3) Мета, об’єкт, предмет та задачі дослідження
4) Рівні схемного паралелізму в побудові паралельного арифметичного зсувача мантис
5) Алгоритм роботи паралельного арифметичного зсувача
6) Програмна модель для оцінки достовірності арифметичного зсувача мантис
7) Діаграми ймовірностей появи суттєвої помилки
8) Діаграми ймовірностей появи несуттєвої помилки
9) Лостовірність результатів зсуву 16-розрядних мантис.
10) Лостовірність результатів зсуву 32-розрядних мантис.

6. КОНСУЛЬТАНТИ ПО РОБОТІ, ІЗ ЗАЗНАЧЕННЯМ РОЗДІЛІВ РОБОТИ, ЩО СТОСУЮТЬСЯ ЇХ

		Підпис, дата	
		Завдання видав	Завдання прийняв
	.		

7. Дата видачі завдання 15.03.2021

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів магістерської роботи	Термін виконання етапів роботи	Примітка
1	Вивчення науково-технічної та патентної	*15.03 - 23.04*	
	літератури з питань побудови		
	арифметичних паралельних пристроїв		
2	Визначення задачі на дослідження	*24.04 - 10.05*	
	достовірності паралельного арифметичного		
	зсувача		
3	Дослідження вимог до програмної моделі	*11.05 - 30.06*	
	арифметичного зсувача мантис		
	щодо оцінки достовірності результатів		
4	Розробка алгоритмів та програмної	*01.09 - 17.10*	
	моделі паралельного арифметичного зсувача		
	щодо оцінки достовірності результатів		
5	Верифікація розробленої програмної моделі	*18.10 - 07.11*	
	на контрольних прикладах та		
	одержання результатів моделювання		
6	Обробка результатів моделювання	*08 .11 - 15.12*	
	паралельного арифметичного зсувача		
	мантис		

Студент Боговик О. С.

Керівник роботи Дрозд О.В.

№ рядка	Формат	Позначення	Найменування	Кільк.	№ екз.	Примітка
1			Документація загальна			
2			Знов розроблена			
3	А4	АМДР.АМ161.2020.01ПЗ	Пояснювальна записка	105		
4	А1	АМДР.АМ161.2020.01Д1	Розвиток персональних	1		
5			комп'ютерів			
6	А1	АМДР.АМ161.2020.01Д2	Асиметрія в роботі цифрових схем	1		
7	А1	АМДРАМ161.2020.01Д3	Контроль за модулем повного та	1		
8			скороченого множення мантис			
9	А1	АМДР.АМ161.2020.01Д4	Введення кратних відмов	1		
10			Рівні схемного паралелізму в			
11			побудові паралельного			
12			арифметичного зсувача мантис			
13	А1	АМДР.АМ161.2020.01Д5	Алгоритм роботи паралельного	1		
14			арифметичного зсувача			
15	А1	АМДР.АМ161.2020.01Д6	Програмна модель для оцінки	1		
16			достовірності арифметичного			
17			зсувача мантис			
18	А1	АМДР.АМ161.2020.01Д7	Діаграми ймовірностей появи	1		
19			суттєвої помилки			
20	А1	АМДР.АМ161.2020.01Д8	Діаграми ймовірностей появи	1		
21			несуттєвої помилки			
22	А1	АМДР.АМ161.2020.01Д9	Лостовірність результатів зсуву	1		
23			16-розрядних мантис			
24	А1	АМДР.АМ161.2020.01Д10	Лостовірність результатів зсуву	1		
25			32-розрядних мантис			
26						
27						
28						

					АММР.АМ161.0303		
Ізм.	*Лист*	*№ документа*	*Підпис*	*Дата*			
Розробив		*Боговик*			Дослідження системи паралельного арифметичного зсуву Відомість дипломної роботи	*Лит.*	*Лист* 1 / *Листів* 1
Перевірив		*Дрозд*					
Н. контр.						ОНПУ ІКС *КІСМ АМ161*	
Затверд.							

АНОТАЦІЯ

Боговик О. С. Дослідження системи паралельного арифметичного зсуву. – Магістерська кваліфікаційна робота. – Державний університет «Одеська політехніка», Одеса, 2021.

Об’єкт дослідження – паралельний арифметичний зсувач мантис одинарної точності.

Предмет дослідження – достовірність результатів, що обчислюються паралельним арифметичним зсувачем мантис в умовах дії типових несправностей на різних рівнях розпаралелювання схеми.

Метою магістерської роботи є дослідження паралельного арифметичного зсувача мантис для оцінки достовірності наближених результатів, що обчислюються в умовах дії типових несправностей.

Методи досліджень базуються на прикладній теорії цифрових автоматів, теорії ймовірностей, а також теорії алгоритмів та математичної логіки.

Магістерська робота розглядає проблему впливу несправностей на достовірність наближених результатів, що обчислюються в однотактних арифметичних пристроях. Проведено дослідження паралельного арифметичного зсувача мантис, що будується з різним схемним паралелізмом та працює під впливом одиночних константних несправностей. Розроблено алгоритми та програмну модель схем паралельного арифметичного зсувача мантис. Досліджено ймовірності виникнення суттєвих та несуттєвих помилок, а також залежність достовірності результату від паралелізму схеми. Моделювання паралельного арифметичного зсувача мантис показало підвищення достовірності результатів із зростанням паралелізму його схеми незважаючи на значне збільшення кількості точок прояву несправностей.

Ключові слова: паралельний арифметичний зсувач мантис, паралелізм схеми, достовірність наближених результатів.

АННОТАЦИЯ

Боговик А. С. Исследование системы паралельного арифметического сдвига. – Магистерская квалификационная работа. – Государственный университет «Одесская политехника, Одесса, 2021.

Объект исследования – параллельный арифметический сдвигатель мантисс одинарной точности.

Предмет исследования – достоверность результатов, вычисляемых параллельным арифметическим сдвигателем мантисс в условиях действия типовых неисправностей на разных уровнях распараллеливания схемы.

Целью магистерской работы являются исследования параллельного арифметического сдвигателя мантисс для оценки достоверности приближенных результатов, вычисляемых в условиях действия типовых неисправностей.

Методы исследований базируются на прикладной теории цифровых автоматов, теории вероятностей, а также теории алгоритмов и математической логики.

Магистерская работа рассматривает проблему влияния неисправностей на достоверность приближенных результатов, вычисляемых в однотактных арифметических устройствах. Проведено исследование параллельного арифметического сдвигателя мантисс, который строится с различным схемным параллелизмом и работает под действием одиночных константных неисправностей. Разработаны алгоритмы и программная модель параллельного арифметического сдвигателя мантисс. Исследованы вероятности появления существенных и несущественных ошибок, а также зависимость достоверности результатов от параллелизма его схемы. Моделирование параллельного арифметического сдвигателя мантисс показало повышение достоверности результатов с ростом параллелизма его схемы, несмотря на значительное увеличение количества точек проявления неисправностей.

Ключевые слова: параллельный арифметический сдвигатель мантисс, параллелизм схемы, достоверность приближенных результатов.

ABSTRACT

Bohovyk O. S. Research of the parallel arithmetic shift system. – The master qualifying work. – Odessa national polytechnic university, Odessa, 2021.

A research object is a parallel arithmetic shifter of mantissas of single exactness.

The article of research is authenticity of results, calculated the parallel arithmetic shifter of mantissas in the conditions of action of model disrepairs on the different levels of chart parallelism.

The purpose of master's degree work is researches of parallel arithmetic shifter of mantissas for the estimation of authenticity of close results, calculated in the conditions of action of model disrepairs.

The methods of researches are based on the applied theory of digital automata, theory of chances, and also theory of algorithms and mathematical logic.

Master work examines a problem of the influence of faults on reliability of the approximated results calculated in simultaneous arithmetic devices. The research of the parallel arithmetic shifter of mantissas, which is built with different circuit parallelism and works under action stuck-at faults, is carried out. Algorithms and program model of the parallel arithmetic shifter of mantissas are developed. Probabilities of essential and inessential errors and dependence of result reliability from parallelism of the circuit are investigated. Simulation of parallel arithmetic shifter of mantissas shows increase of result reliability with growing of the circuit parallelism in spite of significant increase of the fault points amount.

Key words: parallel arithmetic shifter of mantissas, circuit parallelism, reliability of approximated results.

ЗМІСТ

Вступ 5
1 Огляд рішень з підвищення достовірності результатів, що обчислюються у сучасних арифметичних пристроях 9
1.1 Вимоги до сучасних комп’ютерних систем та їх компонентів 9
1.2 Способи підвищення продуктивності комп’ютерних систем та їх компонентів 10
1.3 Способи підвищення достовірності комп’ютерних систем та їх компонентів 15
1.4 Організація обчислень у паралельному арифметичному зсувачі 20
1.5 Існуючі засоби моделювання 24
1.6 Висновки 25
2 Завдання на проведення досліджень 27
2.1 Найменування, область застосування й призначення досліджень . . . 27
2.2 Організація досліджень 28
2.3 Постановка задачі 29
3 Аналіз задачі побудови моделі зсувача 31
3.1 Рівні схемного паралелізму при побудові паралельного арифметичного зсувача мантис 31
3.2 Аналіз вимог до програмної моделі 35
3.3 План проведення досліджень моделі арифметичного зсувача. 39
3.4 Висновки 40

4 Розробка програмної моделі арифметичного зсувача 41
4.1 Алгоритм моделювання паралельного арифметичного зсувача. 41
4.2 Функціонування моделі паралельного арифметичного зсувача. 60
4.3 Висновки 63
5 Результати проведених досліджень 64
5.1 Програмна модель паралельного арифметичного зсувача мантис. . . . 64
5.2 Результати моделювання 69
5.3 Висновки 84
Висновок 85
Перелік літератури 87
Додаток А. Текст програмної моделі 89

ВСТУП

Розвиток комп’ютерних систем та компонентів відбувається у напрямку розпаралелювання обчислювального процесу та збільшення обсягів обробки наближених даних. Це сприяє підвищенню продуктивності комп’ютерних систем та розширенню діапазону оброблюваних даних, що поширює коло вирішуваних задач. Однак вимоги щодо розв’язання задачі складаються не тільки з обсягу обчислень, що потрібно виконати за обмежений час, тобто продуктивності, але й достовірності одержаних результатів. Тому стає важливим питання, як позначаються тенденції розвитку комп’ютерних систем та компонентів на достовірності обчислюваних результатів? Це питання загострюється у проблему, враховуючи широке проникнення комп’ютерних технологій у сфери критичного застосування, тобто в оборонну та космічну галузі, атомну енергетику, транспорт з підвищеними швидкостями, фінансову сферу та ін., що характеризуються надзвичайною відповідальністю до виникаючих в них ризиків. У цих обставинах має певний сенс проведення досліджень поведінки сучасних обчислювальних пристроїв за умови дії їх типових несправностей.

Враховуючи вимоги до високої продуктивності насамперед доцільно розглянути однотактні обчислювальні пристрої з матричним просторовим паралелізмом. Серед них найбільш поширеними стають обчислювальні пристрої для обробки наближених даних у форматах із плаваючою точкою, де безпосередньо наближені обчислення виконуються у частині операцій з мантисами. При тому кожна арифметична операція завершується нормалізацією результатів, а найбільш поширена операція додавання починається з операції денормалізації операндів, що виконується на арифметичних зсувачах мантис. До цього слід додати, що найбільш часто використовуються арифметичні зсувачі у базових форматах з плаваючою точкою з одинарною точністю, за якою результати зберігають розрядність

операндів. Таким чином, обрана тема, щодо розробки та дослідження паралельного арифметичного зсувача мантис з одинарною точністю в умовах дії типових несправностей цілком відповідає вимогам актуальності.

Мета і задачі дослідження**.** Метою магістерської роботи є дослідження паралельного арифметичного зсувача мантис для оцінки достовірності наближених результатів, що обчислюються в умовах дії типових несправностей.

Для досягнення мети в магістерській роботі вирішені такі задачі:

– проведено аналіз вимог до сучасних комп’ютерних систем та обчислювальних пристроїв, а також можливостей побудування паралельного арифметичного зсувача мантис з різним рівнем розпаралелювання схеми;

– визначено вимоги до програмної моделі паралельного арифметичного зсувача мантис для проведення досліджень його поведінки в умовах дії типових помилок на різних рівнях розпаралелювання його схемного рішення;

– розробка алгоритмів та програмної моделі паралельного арифметичного зсувача мантис в умовах дії типових помилок у варіантах різного рівня розпаралелювання його схемного рішення;

– верифікація розробленої програмної моделі паралельного арифметичного зсувача мантис на контрольних прикладах;

– порівняльне моделювання роботи паралельного арифметичного зсувача мантис під дією типових несправностей цифрової схеми на різних рівнях розпаралелювання його схемного рішення;;

– дослідження ймовірностей появи суттєвих та несуттєвих помилок, достовірності результатів, що обчислюються арифметичним зсувачем мантис в умовах дії типових несправностей, а також їх залежності від рівня розпаралелювання схеми.

Об’єкт дослідження – обчислювальний процес, що відбувається в паралельному арифметичному зсувачі мантис.

Предмет дослідження – достовірність результатів, що обчислюються паралельним арифметичним зсувачем мантис в умовах дії типових несправностей на різних рівнях розпаралелювання схеми.

Методи досліджень базуються на прикладній теорії цифрових автоматів, теорії ймовірностей, а також теорії алгоритмів та математичної логіки.

Елементи наукової новизни полягають у наступному:

– розроблена програмна модель паралельного арифметичного зсувача мантис одинарної точності у варіантах з різним рівнем розпаралелювання схеми;

– одержані результати моделювання, за якими визначено ймовірності появи суттєвих та несуттєвих помилок, що викликані типовими несправностями паралельного арифметичного зсувача мантис, достовірність наближених результатів, а також їх залежність від рівня розпаралелюваннія схеми.

Практичне значення одержаних результатів полягає у отриманні оцінки достовірності наближених результатів та її залежності від рівня розпаралелюваннія схеми в умовах дії типових несправностей, що дозволяє передбачити поведінку несправних пристроїв при прояві ризиків в галузях критичного застосування комп'ютерних систем відповідно до рівня розпаралелювання схем, а також вибирати рівень розпаралелювання схем для підвищення достовірності наближених результатів.

Крім того, одержані оцінки можуть бути використані для прогнозування зміни достовірності наближених результатів, що обчислюються у сучасних комп'ютерних системах, при подальшому розпаралелюванні обчислювальних схем.

Структура магістерської роботи. Магістерська робота містить вступ, п'ять розділів основної частини та висновки, а також перелік використаних науково-технічних літературних джерел та додаток.

Вступ спрямований на обґрунтування актуальності обраної теми, та містить основні положення магістерської роботи, включаючи мету дослідження й задачі, що розв'язуються для її досягнення, визначення об'єкту та предмету дослідження, методів, за якими воно здійснюється, а також елементи наукової новизни й практичне значення одержаних результатів. Завершується вступ викладенням структури магістерської роботи.

У першому розділі проводиться аналіз особливостей сучасних комп'ютерних систем та обчислювальних пристроїв, а також вимог до їх основних показників.

Розглядаються структури паралельного арифметичного зсувача мантис одинарної точності, що відрізняються різним рівнем розпаралелювання схем.

У другому розділі викладаються вихідні дані для проведення дослідження та визначаються вимоги, які витікають з мети та поставлених задач, обґрунтовується необхідність проведення дослідження та формулюються завдання на виконання його етапів у наступних розділах магістерської роботи.

У третьому розділі розробляються принципи, за якими має бути побудована програмна модель паралельного арифметичного зсувача мантис одинарної точності.

У четвертому розділі розробляються алгоритми, за якими будується програмна модель паралельного арифметичного зсувача мантис одинарної точності, після чого виконується верифікація розробленої програмної моделі на контрольних прикладах, доводиться достовірність її функціонування..

У п’ятому розділі планується проведення досліджень паралельного арифметичного зсувача мантис одинарної точності на розробленій програмній моделі. Виконується моделювання роботи паралельного арифметичного зсувача мантис під дією типових несправностей його цифрової схеми. За результатами проведеного моделювання визначаються ймовірності появи суттєвих та несуттєвих помилок, що викликаються типовими несправностями схеми паралельного арифметичного зсувача мантис. а також отримується оцінка достовірності обчислюваних результатів та визначаються її залежність від рівня розпаралелювання задіяних схем.

У висновках наводяться основні результати магістерської роботи, що доводять досягнення поставленої мети досліджень.

Перелік літератури містить дані за науково-технічні джерела, які були використані під час проведенні досліджень магістерської роботи з відповідними посиланнями. У додатку викладено вихідний текст програмної моделі паралельного арифметичного зсувача мантис одинарної точності, що функціонує під дією типових несправностей.

1 ОГЛЯД РІШЕНЬ З ПІДВИЩЕННЯ ДОСТОВІРНОСТІ РЕЗУЛЬТАТІВ, ЩО ОБЧИСЛЮЮТЬСЯ У СУЧАСНИХ АРИФМЕТИЧНИХ ПРИСТРОЯХ

1.1 Вимоги до сучасних комп’ютерних систем та їх компонентів

1.1.1 Розв’язання обчислювальної задачі складається з трьох складових:

– виконання певного обсягу обчислень;

– за обмежений час;

– з отриманням достовірних результатів.

Порушення хоча б однієї з цих умов робить задачу нерозв’язаною. Дійсно, частково вирішена задача – це інша задача. Результати, що одержані з порушенням терміну вирішення задачі можуть мати так ж цінність як прогноз погоди на наступний день, що обчислюється за тиждень. Очевидно, що недостовірні. Дві перші складові об’єднуються у такий показник як продуктивність, тобто кількість операцій (з яких складається обчислювальна задача), виконуваних за одиницю часу. Третя складова встановлює вимоги до достовірності результатів, що потребує забезпечення необхідної точності обчислень та надійності як протидії несправностям комп’ютерної системи, що знижують достовірність результатів обчислень.

Тому основними характеристиками комп’ютерної обробки є продуктивність засобів вирішення обчислювальної задачі та достовірність одержуваних результатів. Цим пояснюються високі вимоги до продуктивності комп’ютерних систем та їх компонентів, а також до достовірності обчислюваних результатів.

Увесь розвиток обчислювальної техніки є постійною боротьбою за підвищення продуктивності комп’ютерних систем та їх компонентів, а також достовірності результатів обчислень, що дає змогу розширювати коло вирішуваних задач.

1.2 Способи підвищення продуктивності комп’ютерних систем та їх компонентів

1.2.1 Основними способами підвищення продуктивності комп’ютерних систем та їх компонентів є вдосконалювання елементної бази та розвиток структурних рішень Граничне значення продуктивності є оцінкою рівня розвитку обчислювальної техніки.

Використання обчислювальної техніки обмежується можливостями вирішувати завдання в реальному масштабі часу, діапазон якого може складати для деяких завдань тижня, місяці та роки, а для інших – частки секунди. Протягом обмеженого часу необхідно виконати певний обсяг обчислень, тобто досягти відповідну продуктивність [1].

У вдосконалюванні елементної бази виділяються два аспекти, що спрямовані на підвищення її швидкодії та надійності.

Підвищення швидкодії елементної бази безпосередньо збільшує продуктивність комп’ютерних систем та пристроїв, знижуючи витрати часу на виконання заданого обсягу обчислень.

Надійність елементної бази безпосередньо впливає на продуктивність комп’ютерної системи, оскільки порушення функціонування приводять до зупинок, повторному рахунку, тобто до втрати часу та продуктивності.

Крім того, підвищення надійності елементної бази дозволяє збільшити складність використовуваних структурних рішень, що створює умови для їхнього вдосконалювання в напрямку подальшого розпаралелювання обчислень і в такий спосіб приводить до збільшення продуктивності комп’ютерних систем [2].

1.2.2 Межею розпаралелювання обчислень є складність структурного рішення, а межею складності – надійність комп’ютерної системи. Тому важливе місце у процесі розвитку обчислювальної техніки мають зайняти відмовостійкі комп’ютерної системи та компоненти, які забезпечують підвищення надійності структурними методами.

Серед відмовостійких рішень важливу роль відіграють комп’ютерної системи та компоненти з реконфігурацією. Основу таких систем складає однорідність елементів та регулярність зв’язків, що, у свою чергу збігається з шляхами структурного підвищення продуктивності, що полягають у нарощуванні однорідних структур. Таке збігання виділяє реконфігурацію як перспективний шлях розвитку комп’ютерних систем та компонент і звертає увагу на рівні забезпечення продуктивності обчислень як підставу для організації багаторівневої реконфігурації, тобто не тільки на рівні систем, а й на нижчих рівнях їх компонент [3].

1.2.3 Область застосування методів підвищення продуктивності обчислювальних систем охоплює всі рівні їх створення.

Найнижчий рівень полягає в технології виробництва елементної бази, куди потрібно віднести не тільки елементи, що реалізовують логіку обчислень, але також конструктиви, що обрамовують і зв'язують їх.

Для виконання обчислень найбільше поширення отримали великі (ВІС) та надвеликі (НВІС) інтегральні схеми, що програмуються на конкретні рішення [4]:

– базові матричні кристали (БМК);
– логічні матриці, що програмуються (ПЛМ);
– логічні інтегральні схеми, що програмуються (ПЛІС).

Історично склалася ремонтопридатна організація конструктивів: ВІС та НВІС встановлюються на плати – типові елементи заміни (ТЕЗ), які займають місце на полицях шаф. Ремонтопридатність конструктивів забезпечується можливостями зйому ТЕЗ'ів і доступу до їх елементів. За ремонтопридатність платиться висока ціна, що виражається у втраті надійності на роз'ємних з'єднаннях конструктивів і зниженні швидкодії в наслідок подовження ліній зв'язку та паразитних ємкостей з'єднань.

Швидкодія сучасних ВІС та НВІС сумірна із швидкістю передачі даних по з'єднуючих їх лініях зв'язку. Це пред'являє підвищені вимоги до проектування конструктивів (розміщенню елементів і трасуванню плат друкарського монтажу). Більш швидкодіючі елементи мають меншу щільність монтажу внаслідок

високочастотних перешкод і більшої потужності, що розсівається. Одним з підходів до зменшення суперечності між швидкодією, перешкодостійкістю і потужністю, що розсівається, є використання швидкодіючих елементів тільки у “вузьких місцях".

Наприклад, використання швидкодіючих елементів в ланцюгах перенесення багаторозрядного додавача істотно підвищує швидкодію пристрою загалом.

1.2.4 Наступний крок у напрямку підвищення продуктивності передбачає зменшення кількості логічних рівнів при реалізації комбінаційних схем. Будь-яка логічна функція може бути записана в два рівні логічних операцій за диз'юнктивною (ДНФ) або кон'юнктивною (КНФ) нормальними формами. Однак для складних пристроїв побудова дворівневої реалізації логічних функцій і можливості зниження кількості логічних рівнів обмежуються великими коефіцієнтами з'єднань елементів за входами і виходом [5].

Наступний рівень охоплює способи реалізації основних операцій, до яких відносяться додавання, множення, ділення, а для проблемно-орієнтованих обчислювальних систем також ряд інших операцій, що часто зустрічаються, наприклад, операції зведення в квадрат і вилучення квадратного кореня в обчислювачах для обробки комплексних чисел.

Основними способами підвищення швидкодії і продуктивності операційних пристроїв є матричне і конвеєрне розпаралелювання обчислень. Тому зараз поширення отримали однотактні матричні та порозрядні конвеєрні арифметичні пристрої.

Перспективним підходом до прискорення обчислень є також метод заготівлі результатів, що забезпечує в ряді застосувань кращу швидкодію при менших апаратних витратах у порівнянні з традиційними рішеннями [3].

Продуктивність обчислювальних систем може бути підвищена за рахунок реалізації апаратними або програмно-апаратними засобами вбудованих складних команд, відповідних функціям, що часто зустрічаються. До них можна віднести накопичення суми парних добутків, операції над матрицями чисел, включаючи

множення і обернення матриць, швидке перетворення Фур’є. Такий підхід дозволяє зменшити кількість команд у програмі і скорочує час її виконання, підвищуючи ефективність виконання основної частини команд.

Дослідження процесів виконання великої кількості програм показало найбільш часте використання невеликого набору простих команд. Це послужило основою для розвитку іншого підходу до організації обчислень. Він полягає у відборі і оптимізації виконання обмеженої кількості простих найбільш вживаних команд.

Даний підхід знайшов відображення в організації RISC (reduced instruction set computer) процесорів. Обчислення виконуються з використанням команд типу «регістр-регістр». Команди типу «регістр-пам'ять» застосовуються тільки для обміну даними між регістровою і оперативною пам'яттю. Це прискорює процес компіляції програми і формування завантажувального модуля, зменшує довжину програмного коду, а також скорочує час виконання програми за рахунок швидкого доступу до даних в регістровій пам'яті [6].

Використання обчислювальної техніки обмежується можливостями вирішувати завдання в реальному масштабі часу. Діапазон масштабів часу досить широкий. Для деяких завдань це тижня, місяці й роки, а для інших – частки секунди. За цей ресурс часу необхідно виконати певний обсяг обчислень, тобто досягти відповідну продуктивність.

1.2.5 Найбільші можливості розпаралелювання обчислень забезпечуються в обчислювальних системах з паралельною структурою загального виду [7].

Для керування порядком виконання команд застосовується механізм керування потоком даних.

Структура обчислень описується графом, вершини якого відповідають обчислювальним конструкціям або командам, а дуги – потокам даних.

Дуги спрямовані від вершин, у яких дані обчислюються як результати, до вершин, що використають їх у якості операндів. Крім вершин, що описують обчислення, граф має також вершини, що служать для керування транспортуванням даних.

Реалізація керування потоком даних зіштовхується з наступними основними проблемами:

1) організація циклів і виконання підпрограм;

2) подання й обробка складних структур даних.

Перша проблема вирішується двома групами способів:

1) способи, що засновані на статичному підході, коли цикли й програми розкриваються на етапі трансляції, так що в завантажувальному модулі кожна команда повинна виконуватися тільки один раз;

2) способи, що засновані на динамічному підході, коли операнди забезпечуються мітками й групуються в пари таким чином, щоб для використання різних реалізацій тіла циклу або підпрограми було можливим генерувати копії команд.

Перевага статичного підходу полягає в простоті керування (принцип керування потоком даних реалізується в чистому виді. Команди надходять на виконання із приходом операндів. Недоліком статичного підходу є довгий об'єктний модуль.

Недолік динамічного підходу полягає в складності збору даних, позначених однієї й тією же міткою, а перевагою є порівняно короткий об'єктний модуль і повніше реалізовувати паралелізм, наприклад, при виконанні рекурсивних процедур або циклів, що залежать від даних.

Друга проблема пов'язана з орієнтацією керування потоком даних на виконання скалярних операцій над неструктурованими даними. Тому обробка масивів і ієрархічних структур вимагає розширення концепції потокової обробки й визначення операцій над структурованими даними.

Одним з підходів до рішення другої проблеми є розгляд структурованого даного як елементарної одиниці з наступним включенням механізмів виділення, обробки й повернення для окремих елементів структур.

До факторів, що знижують продуктивність систем, керованих потоком даних, ставляться наступні:

1) Недостатній ступінь паралелізму в програмі. Якщо кількість команд, що допускають одночасне виконання, менше числа функціональних модулів, то частина цих модулів буде простоювати. Для ослаблення дії цього фактора можливо довантажувати систему додатковими завданнями.

1.3 Способи підвищення достовірності комп'ютерних систем та їх компонентів

1.3.1 До основних шляхів підвищення достовірності результатів обчислень є підвищення надійності комп'ютерних систем та компонентів, а також покращення їх діагностичного забезпечення.

Як і у випадку продуктивності підвищення надійності забезпечується за рахунок вдосконалення елементної бази та розвитку структурних рішень.

З арсеналу структурних рішень слід також виділити розробки відмовостійких комп'ютерних систем та компонентів [8].

Відмовостійкість комп'ютерних систем та компонент може забезпечуватися різними шляхами, до яких слід віднести насамперед такі:

- використання коригуючих кодів;
- компенсаційний підхід;
- мажоритарний принцип;
- багатоверсійні рішення.

1.3.1.1 Використання коригуючих кодів набуло найбільшого поширення для інформаційних каналів, за які в комп'ютерних системах, як правило виступають канали зв'язку та блоки пам'яті.

Прикладом коригуючих кодів є коди Хеммінга, що виправляють помилку в одному розряді та виявляють помилки в двох розрядах двійкового коду [9].

Контрольні розряди, що складають надмірну частину коду, беруться таким чином, щоб при декодуванні можна було б встановити не тільки факт наявності помилок у прийнятій комбінації, але й зазначити номер розряду, в якій трапилась

помилка. Локалізація помилки у викривленому слові досягається шляхом багаторазової перевірки прийнятої комбінації на парність певної множини її розрядів. Кількість перевірок дорівнює кількості контрольних розрядів. При кожній перевірці одержується двійковий контрольний сигнал.

Декодування слова, що було закодовано за методом по Хеммінгу, записано у блок пам’яті та потім прочитано з неї, призводить до обчислення контрольних сигналів за парністю, що виконується від певних розрядів слова.

При виявленні помилки контрольний сигнал приймає одиничне значення, а у противному випадку – нульове. Контрольні сигнали складають номер розряду слова, що був викривленим у блоці пам’яті. Номер викривленого розряду перетворюється на дешифраторі в унітарний код, який має одиничне значення на позиції викривленого розряду слова. Унітарний код додається порозрядно за модулем два до слова, що було прочитано з блоку пам’яті. Результатом додавання є зкореговане слово з виправленою помилкою, бо помилковий розряд було проінвертовано при додаванні за модулем два до одиниці.

До недоліків підходу щодо забезпечення відмовостійкості пристроїв пам’яті за використанням корегуючи кодів є висока складність апаратної реалізації, що потребує побудування кодерів та декодерів для визначення помилкових розрядів слова. У разі використання кодів Хеммінга деко дери містять складні дешифратори, які не тільки збільшують додаткові апаратурні витрати ресурсів, але й підвищують час, за яким відбувається виправлення помилкових даних.

1.3.1.2 Компенсаційний підхід. Відмова має за особливість постійний характер, що може бути використано для усунення дії такої несправності за допомогою компенсаційних методів. Такий підхід можна розглянути на прикладі пристрою цифрової затримки, який відноситься до запам’ятовуючих пристроїв, що називаються як FIFO (first input – first output) – «першим прийшов – першим вийшов», та використовується для затримки вхідної послідовності даних на задану кількість тактів [10].

При правильному функціонуванні пристрою цифрової затримки слова затримуються на *2k* тактів, двічі затримуючись на k тактів в одних і тих же n-

розрядних блоках затримки. При відмові пам’яті блоку затримки в одному розряді слово викривляється в цьому розряді.

Помилка виявляється схемою контролю, яка формує одиничне значення контрольного розряду за модулем два. Тоді слово інвертується елементами групи додавачів за модулем два, що відновлює викривлений розряд слова.

Повторне проходження слова супроводжується його записом у ті ж самі комірки пам’яті несправного блоку затримки з повторним викривленням того ж самого розряду та наступним його відновленням за тією ж схемою.

Таким чином, у процесі затримки слова на *2k* тактів викривлений розряд інвертується чотири рази – два рази несправністю та ще два рази додавачами за модулем два, а не викривлені розряди інвертуються тільки два рази додавачами за модулем два. Парна кількість інверсій відновлює правильне значення слова.

1.3.1.3 Мажоритарні системи. Одне з перших запропонованих рішень, що спрямовані на підвищення надійності комп’ютерних систем, полягає у використанні мажоритарної системи, тобто системи з мажоритарною структурою [11].

Мажоритарні системи належать до загальних рішень, що мають універсальних характер, та можуть бути використані у досить широкому колі задач та для різних класів систем та їх компонентів, включаючи запам’ятовуючі пристрої.

Характерною особливістю мажоритарної системи є наявність множини однакових каналів (більш за два однакові канали), кожний з яких вирішує поставлену обчислювальну задачу, а також мажоритарний орган МО, що забезпечує вибір правильного рішення з тієї множини рішень, що одержується множиною каналів.

Найбільш проста й тому найбільш поширена мажоритарна структура має мінімум, тобто три канали. Така система з мажоритарною структурою показана на рис. 1.4.

Мажоритарна система має вхід та вихід, між якими розташовуються канали та мажоритарний орган МО.

Канали мажоритарної системи підключаються паралельно один до одного та обчислюють результат задачі незалежно, але за однаковим рішенням.

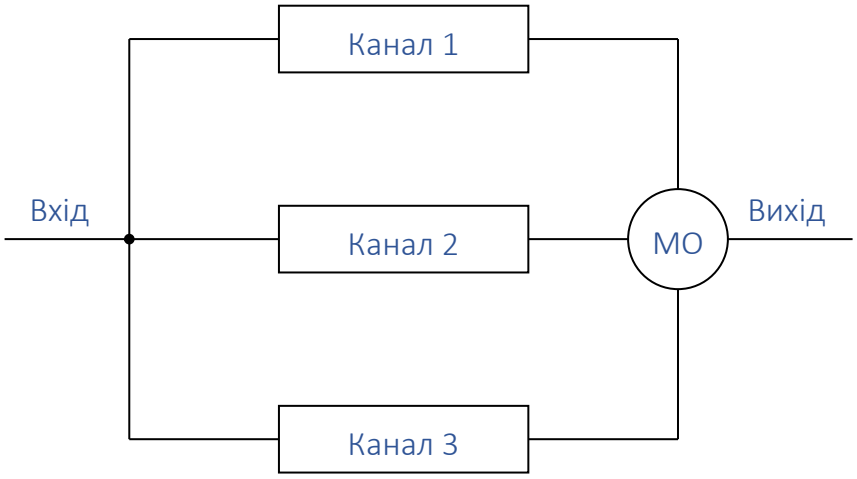

Рис. 1.5. Мажоритарна система з трьох каналів

Мажоритарний орган вибирає один з обчислених каналами результатів за певною ознакою. Як за правило, такою ознакою виступає результат голосування, який одержується при порівнянні результатів обчислень, що одержуються за окремими каналами.

Найбільш часто при голосуванні враховується більшість однакових результатів. Тому кількість каналів, як правило, вибирають непарною, що зменшує ймовірність паритетних варіантів голосування, тобто таких випадків, коли одержується декілька однакових більшостей голосів.

Для мажоритарної системи, що складається з трьох каналів, наявність відмови, яка дає помилку в результаті тільки одного з цих каналів, визначає два однакових правильних результатів та один неправильний результат обчислень.

Порівняння одержаних у каналах результатів, що відбувається у мажоритарному органі, визначає остаточний результат, вибираючи його з множини однакових, тобто правильних результатів.

Обчислення результатів в незалежних каналах забезпечує їх захист від дії несправності, що виникає за загальною причиною.

Недоліком мажоритарної системи є значна складність, яка обумовлена збільшенням втроє апаратних витрат у порівнянні з одним каналом, що є достатнім

для одержання результату розв’язання поставленої обчислювальної задачі. Крім того, є додаткові апаратурні витрати на побудування мажоритарного органу МО.

1.3.1.4 Багатоверсійні системи. Наступним за мажоритарними системами кроком є багатоверсійні комп’ютерні системи, які також вирішують задачу підвищення надійності обчислювальної техніки і перш за все спрямовані на усунення відмов, що спричиняються за загальною причиною [12].

Характерною рисою багатоверсійних систем є виконання каналів за різними версіями, що суттєво розширює множину загальних причин створення відмов, наперекір яким ці системи зберігають свою функціональність.

Однак порівняно до мажоритарних систем багатоверсійні комп’ютерні системи не тільки виключають загальну частину каналів, але й допускають в ній основну концентрацію апаратних ресурсів, що складає умови до суттєвого спрощення, тобто подолання основного недоліку мажоритарних систем.

Тому багатоверсійні комп’ютерні системи мають велику перспективу у їх використанні для підвищення надійності обчислювальної техніки, включаючи запам’ятовуючі пристрої.

Ще одним суттєвим фактором за використання багатоверсійних систем є їхня природність, яка полягає у розвитку живої природи за принципами версійності, тобто створення подібного за різними версіями.

Такий шлях є перевірений часом, що виступає найкращим доказом переваг на користь виконання розробок сучасних обчислювальних систем та їхніх компонентів на основі багатоверсійної архітектури.

1.3.2 Методи та засоби робочого діагностування вирішують задачу контролю достовірності результатів обчислень, що можуть бути викривлені несправностями цифрових схем пристроїв. Це дозволяє перерахувати недостовірні результати до завершення встановленого терміну, і таким чином вирішити обчислювальну задачу [13].

До найбільш поширених методів робочого діагностування арифметичних пристроїв належить контроль за числовим модулем.

1.4 Організація обчислень у паралельному арифметичному зсувачі

1.4.1 Паралельний арифметичний зсувач широко використовується в сучасних комп'ютерних системах. Найбільш часто паралельний арифметичний зсувач застосовується при виконанні операцій додавання чисел із плаваючою точкою [14].

На рис. 1.1 показана структурна схема додавача, що виконує операцію додавання у форматах з плаваючою точкою та використовує паралельні арифметичні зсувачі.

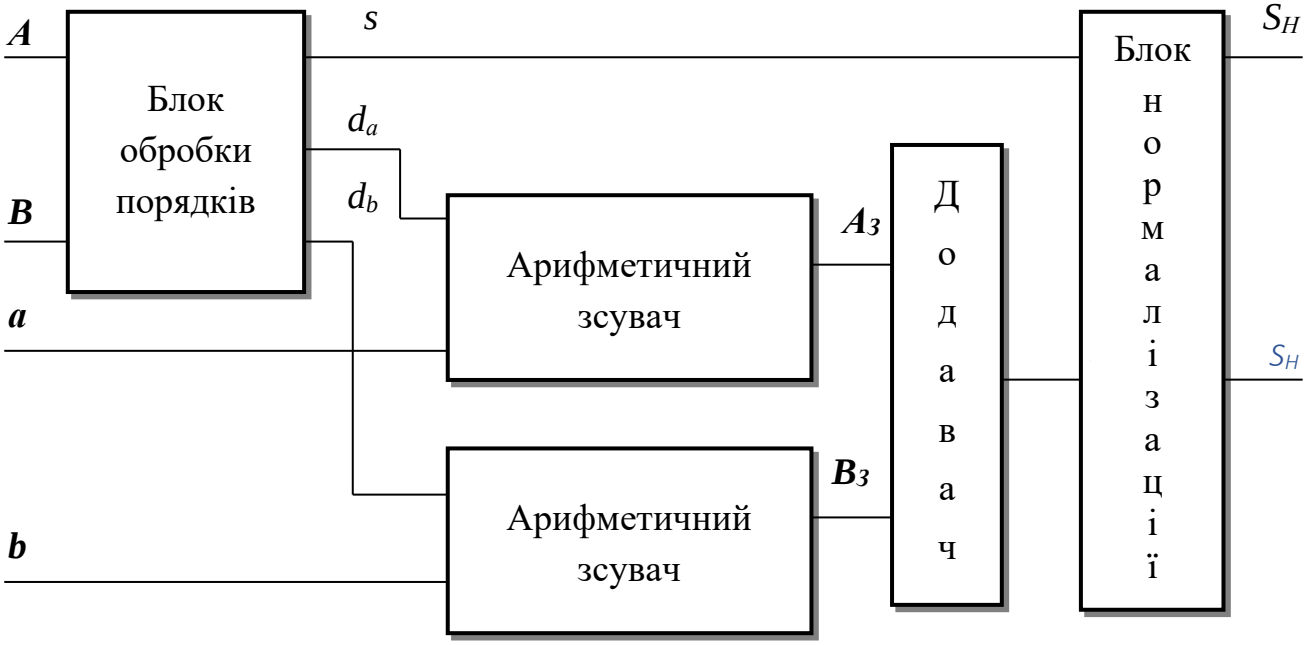

Рис. 1.1. Структурна схема додавача з плаваючою точкою

Додавач з плаваючою точкою містить блоки обробки порядків, два арифметичні зсувачі, додавач зсунутих мантис та блок нормалізації.

Блок обробки порядків приймає порядки *a b* доданків та обчислює порядок суми *s*, а також вирівнюючи різниці d_a, d_b.

Арифметичні зсувачі приймають мантиси доданків *A*, *B* та їх вирівнюючи різниці d_a, d_b і обчислюють зсунуті мантиси. $A_З$, $B_З$.

Додавач приймає зсунуті мантиси та обчислює їх суму *S*.

Блок нормалізації аналізує цю суму та у разі потреби нормалізує мантису до вигляду суми результату S_H відповідно змінюючи його порядок до вигляду s_H.

1.5.3 Паралельний арифметичний зсувач виконує операцію арифметичного зсуву мантиси праворуч, яка є окремим випадком операції множення і виконується із скороченням обчислень. Операція арифметичного зсуву складається з трьох дій, які на прикладі мантиси A містять:

– відкидання d_a молодших розрядів $A\{n\text{-}d_a\text{+}1 \div n\}$ мантиси;

– заповнення позицій, що вивільняються (ваги $2^{-1} \div 2^{-da}$), нулем для прямого коду та знаком $Зн$ для оберненого або додаткового кодів мантиси;

– зміна ваги розрядів $A\{1 \div n\text{-}d_a\}$ мантиси в 2^{da} разів з $2^{-1} \div 2^{-n+da}$ до $2^{-da-1} \div 2^{-n}$.

На рис. 1.2 показана матриця зсуву, що описує виконання арифметичного зсуву як скороченої операції.

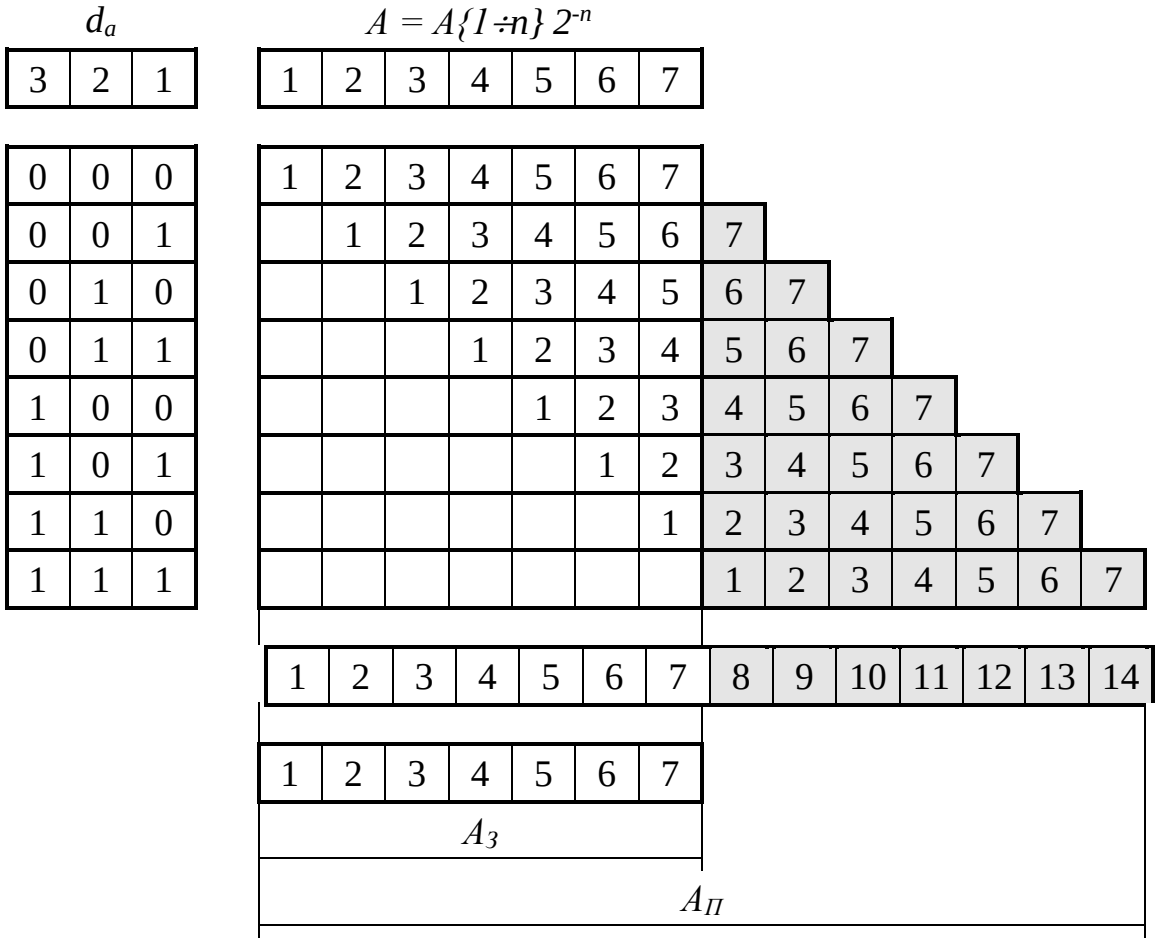

Рис. 1.2. Матриця зсуву

Рядки матриці містять варіанти положення розрядів мантиси *A* після її зсуву на $d_a = 0 \div n$ позицій ($n = 7$). Значення d_a подані в двійковій СЧ. Затемнені частини числа не обчислюються.

Операція арифметичного зсуву належить до множини операцій, виконуваних у форматах із плаваючою точкою і є варіантом зсуву зі скороченням обчислень. Арифметичний зсув приводить до втрати молодших розрядів операнда, причому кількість розрядів, що втрачаються, порівняна з кількістю розрядів, що залишаються, та довжиною операнда.

Виконання операції показано на рис. 1.3.

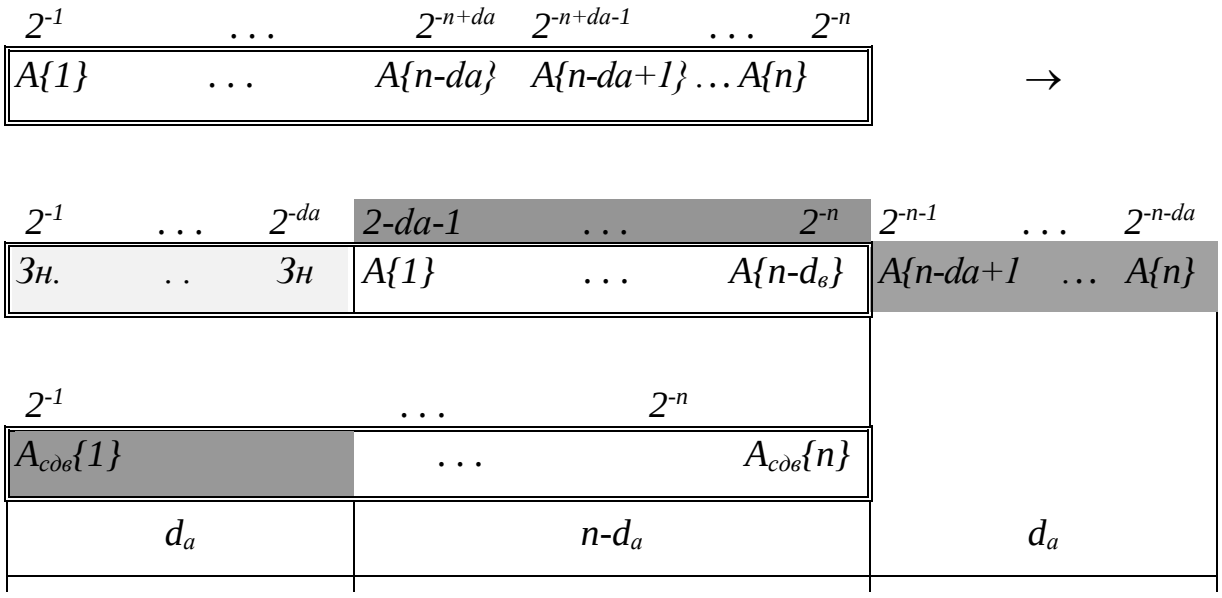

Рис. 1.3. Схема виконання арифметичного зсуву

Зміни, зв'язані зі зсувом мантиси, позначені затемненням відповідних полів:

- розрядів, що відкидаються, $A\{n\text{-}da+1 \div n\}$,
- знакових розрядів *Зн*;
- ваг $2\text{-}da\text{-}1 \div 2n$-розрядів $A\{1 \div n\text{-}da\}$.

Паралельний арифметичний зсувач виконує паралельну обробка даних та будується із застосуванням групи одночасно працюючих мультиплексорів. Кожний з них обчислює один конкретний розряд зсунутого числа.

На рис. 1.4 показаний паралельний арифметичний зсувач.

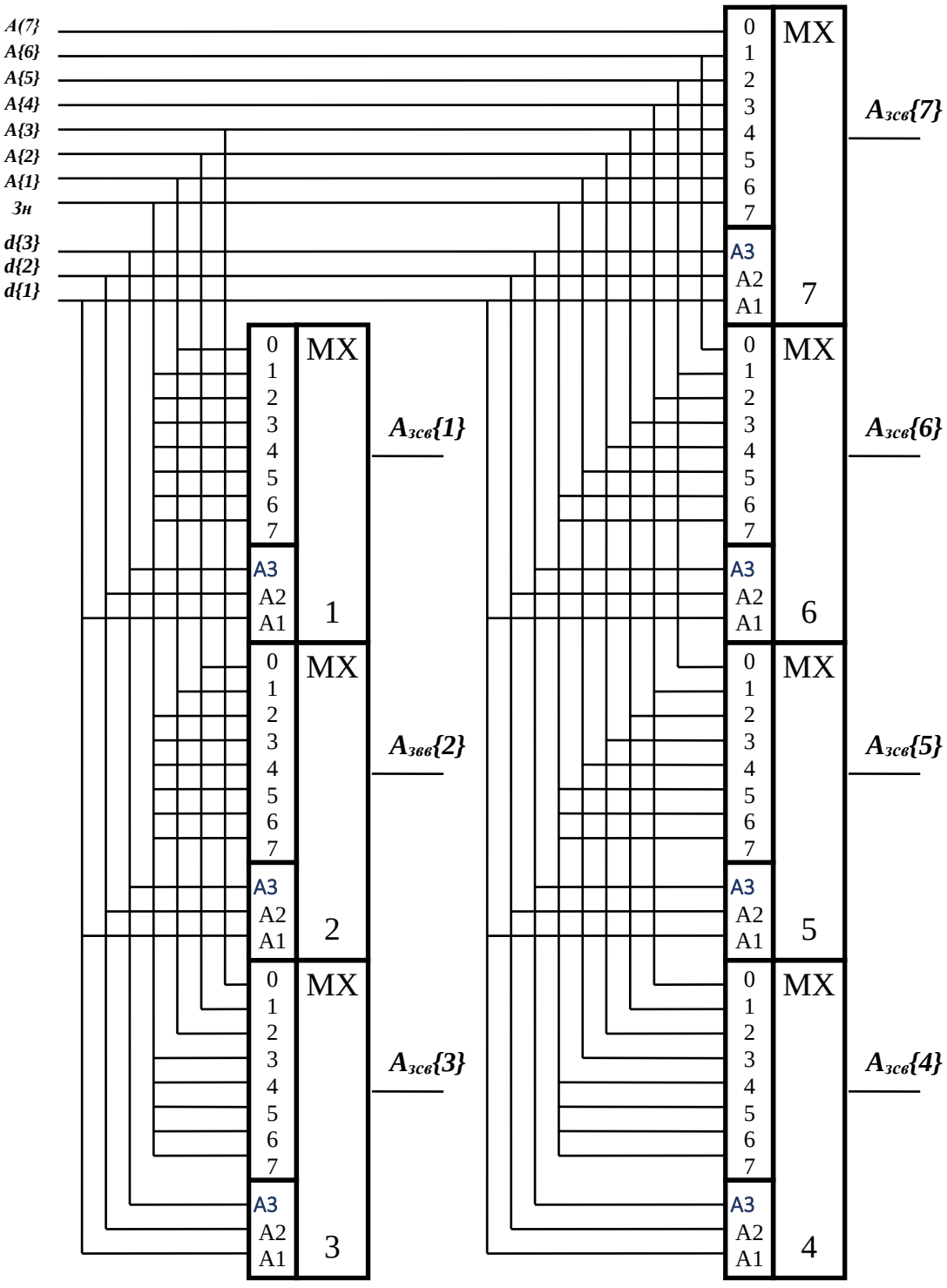

Рис. 1.4. Паралельний арифметичний зсувач

На адресні входи мультиплексора подається величина зсуву, а на інформаційні – розряди операнда.

Мультиплексор підключає на вихід інформаційний вхід, що відповідає величині зсуву. На цей вхід подається розряд операнда, що при даній величині зсуву стає закріпленим за мультиплексором розрядом результату.

Арифметичні паралельні зсувачі заповнюють вивільняють позиції нулями для прямого коду операнда та значеннями знакового розряду для оберненого й додаткового кодів.

Пристрій містить мультиплексори *MX 1 ÷ 7*. На їхні інформаційні входи надходять зі зміщенням в напрямку молодших розрядів розряди числа *A*, що витісняються знаковим розрядом. Адресні входи мультиплексорів підключаються до відповідних розрядів величини зсуву *d{1÷3}*. На виходах мультиплексорів обчислюються розряди зсунутого числа $A_{зсв}$. Найбільше поширення паралельний арифметичний зсувач знаходить у реалізації базової арифметичної операції – додаванні – на однотактному суматорі із плаваючої точкою.

1.5 Існуючі засоби моделювання

1.5.1 У нинішній час у галузі моделювання та проектування апаратних та програмних засобів обчислювальної техніки працює багато відомих фірм із світовим ім’ям, таких як Aldec, Altera Europe, Answer Systems, Aplas Solutions Corporation, Aptix Corporation, C Level Design, Cadence Design Systems, EDA Consortium, IKOS Systems, IMEC, InSilicon Corporation, Xpedion Design Systems та інші.

1.5.2 Розроблені автоматизовані системи проектування цифрової, аналогової та цифро-аналогової обчислювальної техніки забезпечують багаторівневе представлення об’єкту розробки і проектування від верхнього рівня – загальної структури, або просторого опису до самого низу – конкретної елементної бази за сучасними технологіями – ASIC, FPGA, SoC. Ці системи доповнені засобами моделювання, що дозволяють на рівні програмної моделі одержувати схемотехнічні показники витрат обладнання та часу об’єкту розробки. Системи підтримані засобами візуального програмування, що

забезпечують супроводження формального опису графічним зображенням та навпаки дозволяють за графічним зображенням схем одержувати їх формальний опис. Універсальний характер автоматизованих систем проектування допускає подання задачі на умовній елементній базі з подальшим викладенням рішення до конкретної реалізації на сучасних елементах.

1.5.3 Відкритість систем до їх збагачування стандартними рішеннями складає фундамент для піднесення процесу проектування на високі рівні представлення об’єкту розробки. До таких стандартних рішень відносяться вузли ОП, формалізовані за традицією мікросхем середнього ступеня інтеграції – багаторозрядні регістри, додавачі, збірки логічних елементів з єдиним управлінням, дешифратори, мультиплексори, лічильники, тощо, а також самі ОП – помножувачі, поділювані та інші. Розвиток цього напряму знайшов продовження в створенні бібліотек IP-корів, тобто рішень, що є інтелектуальною власністю, а їх бібіліотеки є накопиченням вирішених задач з розробки компонентів комп’ютерних систем.

Існуючі системи моделювання спрямовані на виявлення помилок проектування. Однак вони не розглядають вплив на достовірність результатів помилок, що викликаються несправностями цифрових схем. Таким чином, дослідження роботи пристроїв в умовах дії несправностей потребує розробки програмних моделей, що суттєво доповнюють відомі системи моделювання.

1.6 Висновки

1.6.1 Вимоги до сучасних комп’ютерних систем та їх компонентів висувають за основні показники продуктивність та достовірність результатів обчислень.

До основних способів підвищення продуктивності відноситься вдосконалення елементної бази у напрямку зменшення затримок елементів та підйому частоти їх роботи, а також структурні методи.

За структурними методами підвищення продуктивності одержуються рішення з розпаралеленням обчислень на системному та схемному рівнях.

Розпаралелення обчислень на системному рівні забезпечується виконанням окремих арифметичних пристроїв для виконання окремих арифметичних операцій.

Розпаралелення обчислень на схемному рівні виконується з використанням матричного паралелізму для побудови однотактних арифметичних пристроїв.

Крім того, схемний паралелізм може підвищуватися за рахунок зменшення кількості логічних рівнів при реалізації комбінаційних схем.

Основні способи підвищення достовірності також розподіляються за напрямками вдосконалення елементної бази та на структурні методи.

Структурні методи підвищення достовірності результатів насамперед спрямовані на підйом надійності засобів розв’язання обчислювальної задачі розробкою відмовостійких комп’ютерних систем і компонентів з використанням коригуючих кодів, компенсаційних методів, а також мажоритарних рішень і багатоверсійних технологій. Крім того, для підвищення достовірності результатів обчислень залучаються методи та засоби робочого діагностування, що дозволяють відсіяти недостовірні результати.

1.6.2 Слід відмітити, що практично зовсім не приділяється уваги до природних ресурсів підвищення достовірності результатів обчислень. Доцільно провести дослідження на виявлення таких природних ресурсів.

За об’єкт такого дослідження може бути розглянутий паралельний арифметичний зсувач мантис. Цей пристрій має поширене використання в домінуючому напрямку обробки наближених даних в форматах із плаваючою точкою. Він відноситься до однотактних і демонструє приклад розпаралелення обчислень на системному та схемному рівнях. Крім того, на паралельному арифметичному зсувачі мантис легко реалізовувати різну кількість логічних рівнів.

Такі особливості паралельного арифметичного зсувача мантис дозволяють дослідити на ньому залежність достовірності результату обчислень від рівня схемного паралелізму.

Слід відзначити, що відомі засоби моделювання цифрових схем не забезпечують проведення таких досліджень, що потребує виконання власної розробки програмної моделі арифметичного зсувача мантис.

2 ЗАВДАННЯ НА ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ

2.1 Найменування, область застосування й призначення досліджень

2.1.1 Тема магістерської роботи – Розробка та дослідження моделі паралельного арифметичного зсувача мантис в умовах дії несправностей присвячена дослідженню обчислювального процесу, що протікає під дією характерних несправностей у однотактному паралельному арифметичному зсувачі мантис.

2.1.2 Областю застосування розробки є сучасні комп'ютерні системи, що мають високу продуктивність, розширений діапазон подання чисел та підвищені вимоги до достовірності результатів обчислень.

Однотактний паралельний арифметичний зсувач мантис чисел із плаваючою точкою застосовується для обробки наближених даних у нормальній формі подання в якості ділянки або декількох ділянок конвеєра у векторних процесорах, а також у вигляді окремих функціональних пристроїв СуперЕОМ та у арифметичних блоках систем наддовгих команд і периферійних арифметичних процесорів.

2.1.3 Дані дослідження призначені для проведення аналізу обчислювального процесу за моделлю паралельного арифметичного зсувача нормалізованих двійкових мантис для оцінки достовірності результатів, що обчислюються під дією характерних несправностей,

Аналізується ймовірність появи помилки, викликаною характерною несправністю, а також їхній розподіл на суттєві й несуттєві помилки.

Вивчається статистика появи помилок, викликаних характерними несправностями однотактових ОП, що дозволяє оцінювати достовірність результатів, що обчислюються.

Розглядаються та порівнюються операції зсуву мантис у двійковому коді, що здійснюються в однотактових пристроях із різним ступенем схемного паралелізму.

2.2 Організація досліджень

2.2.1 Для проведення досліджень розробляється модель паралельного арифметичного зсувача мантис.

Передбачаються режими завдання мантиси операндів зсуву для визначеної розрядності величини зсуву, а також режими правильного функціонування пристрою і його роботи під дією несправності.

Операндами паралельного арифметичного зсувача є двійкові коди нормалізованої мантиси зсуву та величина зсуву, а результатом – двійковий код зсунутої мантиси.

Програмна модель забезпечує можливість порівнювати способи обчислення результату з різним ступенем схемного паралелізму та різним розподілом розрядів результату на вірні та невірні.

2.2.2 Найбільшу ефективність методи зсування демонструють при їхньому виконанні в однотактних пристроях. Такі пристрої набутили основного застосування в цей час і є перспективними на найближчі п'ять років. Особливістю однотактних пристроїв є клас характерних несправностей «одиночні константі несправності», які, як правило, спотворюють результат на вагу його будь-якого одного розряду. Основна частка оброблюваних чисел ставиться до наближених даних, представленим у форматах із плаваючою точкою. Наближене число містить старші вірні й молодші невірні розряди, тобто такі, що мають вагу нижче

рівня похибки. У цьому випадку помилки розділяються на суттєві й несуттєві. Суттєві помилки спотворюють вірні розряди результату, а несуттєві помилки мають величину нижче рівня похибки й не знижують достовірність результату, обумовлену його вірними розрядами.

Мантиси чисел із плаваючою точкою, як правило, обробляються з одинарною точністю, тобто округлений результат має ту ж розрядність, що і його операнди. Для операції зсуву мантиса завжди обробляється за скороченою операцією.

Це приводить до втрати, у середньому, кожної другої помилки, що викликається несправностями арифметичного пристрою. Такі помилки не роблять впливу на округлений результат і тому є несуттєвими.

Арифметичний зсув є окремим випадком операції множення мантис і, майже завжди виконується як скорочена операція, тобто з розрядністю мантиси до зсуву. Це відповідно зменшує кількість виникаючих у них несуттєвих помилок, знижуючи негативний ефект відбраковування достовірних результатів при їхньому виявленні.

Однак, скорочені операції не одержали належного поширення. Не розглянуто їхнє виконання в додатковому коді, що потребує проведення додаткових досліджень скороченого множення мантис, що виконується у додатковому коді у однотактному пристрої.

2.3 Постановка задачі

2.3.1 Ставиться задача на проведення досліджень по виконанню зсуву мантис у однотактному пристрої. Для цього необхідно розробити модель паралельного арифметичного зсувача мантис, що дозволяє вивчати його функціонування в справному стані пристрою та при виникненні його характерних несправностей. Операція виконується для обчислення результату з різним

ступенем схемного паралелізму та різним розподілом розрядів результату на вірні та невірні.

Характерні несправності арифметичного зсувача викликають суттєві та несуттєві помилки для достовірності округленого результату обчислень. Суттєві помилки спотворюють старші вірні розряди результату, а несуттєві помилки – молодші невірні.

2.3.2 Модель повинна задовольняти ряду вимог: бути наочної, зручної у використанні, функціональної й наближеної до реальності.

Наочність моделі потребує образотворчих мовних засобів, відображаючи структуру паралельного арифметичного зсувача і його елементів. Вихідні дані та результати моделювання повинні мати інформативну індикацію. Зручність у використанні повинна ґрунтуватися на простоті та варіантності завдання режимів роботи, вихідних даних та одержанні результатів моделювання.

Функціональність моделі потребує завдання будь-яких значень операнда мантиси та операнда зсуву та послідовностей довільних значень; виконання функцій паралельного арифметичного зсувача з урахуванням уведеної несправності та при її відсутності; оцінку помилок і статистики їх виникнення. Реальність моделі повинна забезпечуватися правильністю опису функцій, реалізованих зсувачем і наближенням моделі несправності до реальних дефектів однотактних пристроїв.

Використовуючи розроблену модель, необхідно оцінити ймовірності появи суттєвих та несуттєвих помилок та достовірність результатів, що обчислюються під дією типових несправностей в умовах обчислення результату з різним ступенем схемного паралелізму.

3 АНАЛІЗ ЗАДАЧІ ПОБУДОВИ МОДЕЛІ ЗСУВАЧА

3.1 Рівні схемного паралелізму при побудові паралельного арифметичного зсувача мантис

3.1.1 Одною з особливостей паралельного арифметичного зсувача мантис є можливості його побудови з різним рівнем схемного паралелізму, що складає умови для дослідження залежності достовірності результатів, що обчислюються під дією характерних несправностей,

На рис. 3.1 показана триярусна схема паралельного арифметичного зсувача мантис для розрядності $n = 7$. Кожний з ярусів схеми є також паралельним зсувачем мантис і будується на одноадресних мультиплексорах.

Перший ярус схеми містить одноадресні мультиплексори 1.1 – 1.7, приймає розряди $A\{1\} \div A\{7\}$ мантиси A операнда, знаковий розряд $A\{0\}$, а також молодший розряд $B\{1\}$ величини зсуву B та виконує зсув мантиси праворуч на нуль або одну позицію відповідно при нульовому або одиничному значенні розряду $B\{1\}$ величини зсуву.

Другий ярус схеми містить одноадресні мультиплексори 2.1 – 2.7, приймає мантису операнда, що вже зсунута на $B\{1\}$ позицій, знаковий розряд $A\{0\}$, а також другий розряд $B\{2\}$ величини зсуву B та виконує зсув мантиси праворуч на нуль або дві позиції відповідно при нульовому або одиничному значенні розряду $B\{2\}$ величини зсуву.

Третій ярус схеми містить одноадресні мультиплексори 3.1 – 3.7, приймає мантису операнда, що вже зсунута на $B\{1, 2\}$ позицій, а також третій розряд $B\{3\}$ величини зсуву B та виконує зсув мантиси праворуч на нуль або чотири позиції відповідно при нульовому або одиничному значенні розряду $B\{3\}$ величини зсуву. На вихід третього ярусу та схеми зсувача видаються розряди $A_З\{1\} \div A_З\{7\}$ зсунутої мантиси $A_З$.

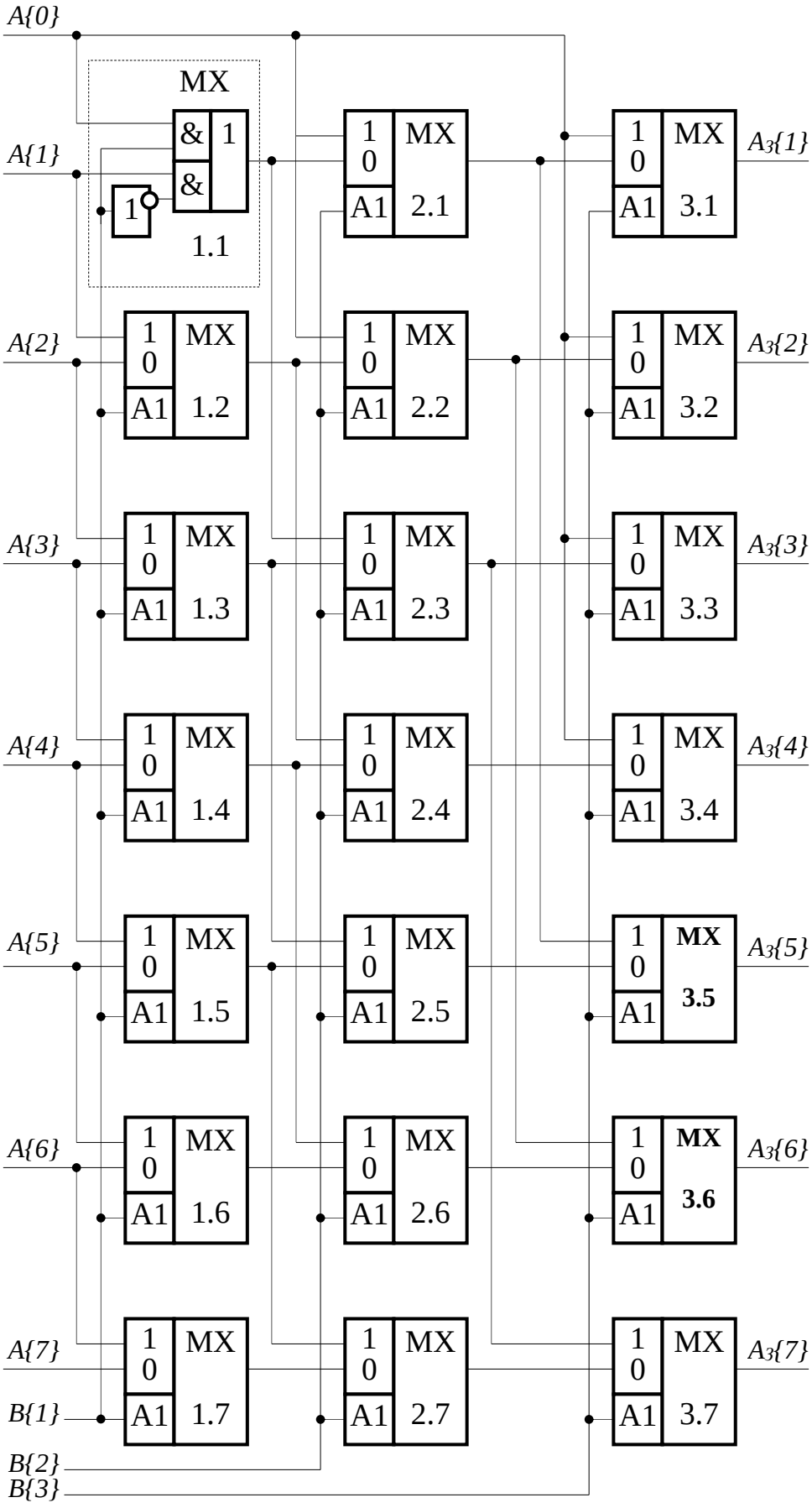

Рис. 3.1 – Триярусна схема паралельного арифметичного зсувача

Перший одноадресний мультитплексор 1.1 розкритий на рівні принципової схеми. Він містить інвертор та логічний елемент 2І-2АБО. Перші (верхні) входи елементів І є відповідно першим «1» та нульовим «0» інформаційними входами одноадресного мультиплексора, а другі входи підключені безпосередньо та з використанням інвертора до входу адреси А1. Вихід логічного елемента 2І-2АБО є виходом одноадресного мультитплексора. Одноадресний мультиплексор підключає на вихід нульовий або перший інформаційний вхід відповідно при нульовому або одиничному значенні на вході адреси А1.

Старші вивільнені позиції зсунутої мантиси заповнюються значеннями знакового розряду. Після першого ярусу значенням знакового розряду заповнюється $B\{1\}$ позицій, після другого ярусу – ще $2B\{2\}$ позицій, а після третього ярусу значенням знакового розряду заповнюється ще $4B\{3\}$ позицій, що загалом складає B позицій.

3.1.2 На рис. 3.2 показана двоярусна схема паралельного арифметичного зсувача мантис для розрядності $n = 7$.

Кожний з ярусів схеми є також паралельним зсувачем мантис і будується на двоадресних та одноадресних мультиплексорах.

Перший ярус схеми містить двоадресні мультиплексори 1.1 – 1.7, приймає розряди $A\{1\} \div A\{7\}$ мантиси A операнда й два молодших розряди $B\{1\}$ і $B\{2\}$ величини зсуву B та виконує зсув мантиси праворуч на $B\{1, 2\} = 0 \div 3$ позицій.

Другий ярус схеми містить одноадресні мультиплексори 2.1 – 2.7, приймає мантису операнда, що вже зсунута на $B\{1, 2\}$ позицій, а також третій розряд $B\{3\}$ величини зсуву B та виконує зсув мантиси праворуч на нуль або чотири позиції відповідно при нульовому або одиничному значенні розряду $B\{2\}$ величини зсуву.

На вихід другого ярусу та схеми паралельного арифметичного зсувача видаються розряди $A_З\{1\} \div A_З\{7\}$ зсунутої мантиси $A_З$.

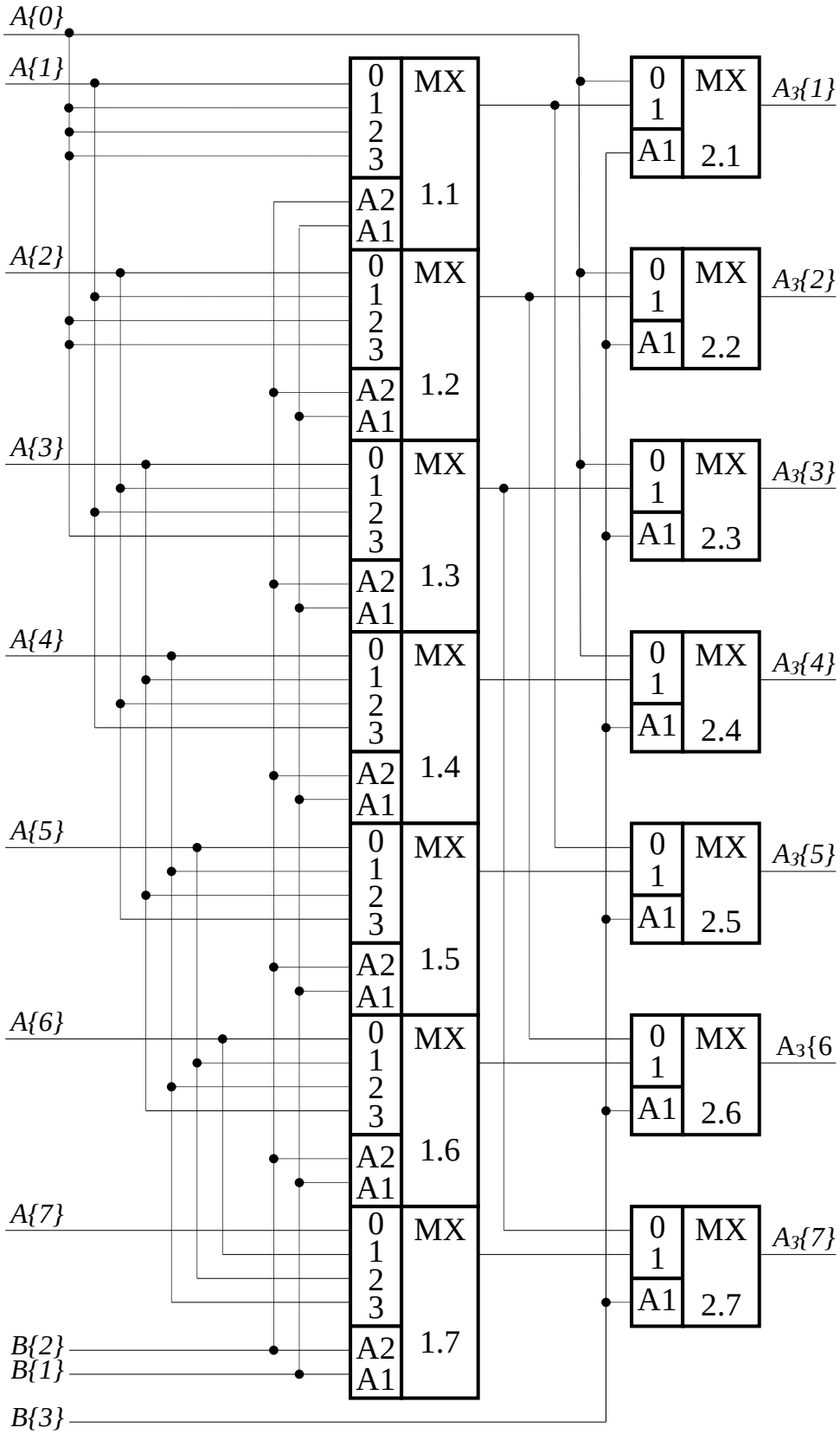

Рис. 3.2 – Двоярусна схема паралельного арифметичного зсувача

3.1.3 Схема паралельного арифметичного зсувача мантис може бути також одноярусною, як це показано на рис. 1.4.

Виконання паралельного арифметичного зсувача мантис з різною ярусністю схеми дозволяє реалізувати різні рівні схемного паралелізму.

Триярусна схема арифметичного зсувача мантис має найбільшу кількість логічних рівнів, одноярусна схема має найменшу кількість логічних рівнів, а двоярусна схема займає за схемним паралелізмом проміжне положення.

Такі реалізації паралельного арифметичного зсувача мантис дозволяють дослідити достовірність результатів, що обчислюються під дією несправностей, у залежності від рівня схемного паралелізму.

3.2 Аналіз вимог до програмної моделі

3.2.1 Наочну програмну модель паралельного арифметичного зсувача мантис у додатковому коді праворуч доцільно скласти, користуючись мовними засобами середовища Delphi.

Структурно, у моделі паралельного арифметичного зсувача мантис необхідно передбачити:

– демонстраційні елементи;

– елементи керування.

3.2.2 Демонстраційні елементи програмної моделі паралельного арифметичного зсувача мантис повинні містити:

– поле завдання значень двійкових розрядів мантиси операнда зсуву *A*;

– поле завдання десяткового значення мантиси операнда зсуву *A*;

– поле завдання значень двійкових розрядів величини зсуву *B*;

– поле завдання десяткового значення величини зсуву *B*;

– покажчик розподілу результату зсуву на вірні та невірні розряди;

– поле десяткових значень правильних результатів *C*;

– поле десяткових значень результатів *Се*, отриманих при впливі несправності;

– поля, що показують значення кодів величини зсуву для різних рівнів схемного паралелізму;

– поля, що показують значення кодів мантиси операнда після її зсуву праворуч на вказані значення кодів величини зсуву для різних рівнів схемного паралелізму;

– поля таблиці результатів моделювання, що вказують на рівні схемного паралелізму паралельного арифметичного зсувача мантис;

– поля таблиці результатів моделювання, що містять кількість контрольних точок паралельного арифметичного зсувача мантис для різних рівнів схемного паралелізму;

– поля таблиці результатів моделювання, що містять ймовірності появи помилки в результаті паралельного арифметичного зсувача мантис для різних рівнів схемного паралелізму;

– поля таблиці результатів моделювання, що містять ймовірності появи суттєвої помилки в результаті паралельного арифметичного зсувача мантис для різних рівнів схемного паралелізму;

– поля таблиці результатів моделювання, що містять ймовірності появи несуттєвої помилки в результаті паралельного арифметичного зсувача мантис для різних рівнів схемного паралелізму;

– поля таблиці результатів моделювання, що містять оцінки достовірності результатів паралельного арифметичного зсувача мантис для різних рівнів схемного паралелізму;

3.2.3 Елементи керування повинні включати у свій склад:

– меню завдання режимів визначення значень мантиси операнда зсуву А, величини зсуву В; справного або несправного стану арифметичного зсувача;

– панель вказівки розподілу результатів на вірні та невірні розряди;

– кнопки прийому даних у демонстраційні поля.

3.2.4 Функціонально, програмна модель арифметичного зсувача мантис у додатковому коді повинна забезпечувати:

– завдання розрядності паралельного арифметичного зсувача нормалізованих мантис;

– визначення режимів завдання операндів;

– завдання справного або несправного стану паралельного арифметичного зсувача мантис;

– завдання значень мантиси операнда зсуву та величини зсуву;

– завдання точки розподілу результатів, що обчислюються, на вірні та невірні розряди;

– виконання функцій арифметичного зсувача мантис у додатковому коді з урахуванням уведеної несправності та при її відсутності;

– обчислення та індикацію правильного значення мантиси зсувача та її значення при впливі введеної несправності;

– обчислення та індикацію рівнів схемного паралелізму арифметичного зсувача мантис;

– обчислення та індикацію кількості контрольних точок паралельного арифметичного зсувача мантис для різних рівнів схемного паралелізму;

– обчислення та індикацію ймовірності появи помилки в результаті паралельного арифметичного зсувача мантис для різних рівнів схемного паралелізму;

– обчислення та індикацію ймовірності появи суттєвої помилки в результаті паралельного арифметичного зсувача мантис для різних рівнів схемного паралелізму;

– обчислення та індикацію ймовірності появи несуттєвої помилки в результаті паралельного арифметичного зсувача мантис для різних рівнів схемного паралелізму;

– підрахунок та індикацію достовірності результату, що обчислюється в паралельному арифметичному зсувачі мантис для різних рівнів схемного паралелізму під впливом типових несправностей;

3.2.5 Режими завдання операндів повинні забезпечувати:

– завдання конкретного значення операнда;

– завдання довільного значення операнда;

– перебір всіх значень операнда;

– генерацію послідовності довільних значень операнда.

Конкретні значення операндів повинні задаватися наступними двома способами:

– у двійковому коді в полі розрядів мантиси операнда вказівкою значення кожного розряду або інверсією його попереднього значення шляхом щиглика миші по цьому розряду;

– у полі значення мантиси операнда безпосередньою вказівкою необхідного числа.

На початку операндам доцільно привласнити нульові значення.

3.2.6 За типову несправність паралельного арифметичного зсувача нормалізованих мантис досліджується одиночна константна несправність контрольних точок пристрою.

Контрольними точками паралельного арифметичного зсувача нормалізованих мантис призначаються всі входи та виходи логічних елементів цифрової схеми пристрою.

Одиночна константна несправність приймає в контрольних точках схеми паралельного арифметичного зсувача нормалізованих мантис нульові та одиничні постійні значення.

Нульова одиночна константна несправність контрольної точки передає нульове значення всім точкам схеми паралельного арифметичного зсувача нормалізованих мантис, що з’єднані з нею.

Одинична одиночна константна несправність контрольної точки є наслідком обриву цієї точки від виходу попереднього логічного елементу цифрової схеми паралельного арифметичного зсувача нормалізованих мантис, що формує у цій точці рівень сигналу, що перевищує порогове значення між нулем та одиницею.

3.3 План проведення досліджень моделі арифметичного зсувача

3.3.1 Необхідно переконатися в працездатності програмної моделі паралельного арифметичного зсувача мантис у додатковому коді шляхом перевірки на тестових прикладах

– режимів вихідних установок програмної моделі;
– виду демонстраційних елементів програмної моделі;
– функціонування програмної моделі в різних режимах;
– достовірності одержуваних результатів обчислень.

У вихідних установках програмної моделі паралельного арифметичного зсувача мантис варто перевірити

– правильність завдання режимів роботи програмної моделі;
– правильність завдання режимів одержання операнда мантиси
– правильність завдання режимів одержання величини зсуву;
– правильність завдання значень вихідних даних;
– правильність режимів завдання несправності;
– правильність розподілу результатів на вірні та невірні розряди;
– контроль завдання нормалізованих мантис паралельного зсувача при уведенні десяткового числа, а також уведенні або корекції двійкового коду;

Демонстраційні елементи програмної моделі паралельного зсувача мантис необхідно перевірити на:

– правильність відображення структур пристрою для різних рівнів схемного паралелізму

У функціонуванні програмної моделі паралельного арифметичного зсувача мантис необхідно перевірити:

– правильність обчислень, виконуваних у моделі матричного зсувача при відсутності несправностей
– на конкретних значеннях операнда мантиси та величини зсуву;
– при повному переборі значень операнда мантиси та величини зсуву;
– на випадковій послідовності значень мантиси та величини зсуву;

– для різних розрядностей арифметичного зсувача.

– для позитивних та негативних мантис;

– правильність обчислень у моделі пристрою при введеній несправності

– конкретного виду та у конкретному місці;

Достовірність одержуваних результатів перевіряється

– порівнянням правильних десяткових значень результатів з відповідними значеннями результатів, отриманих при впливі несправності;

– оцінкою ймовірностей появи суттєвих і несуттєвих помилок на послідовностях вхідних слів при завданні типових несправностей пристрою;

– візуальним спостереженням за роботою елементів керування й демонстраційних елементів моделі.

3.4 Висновки

Для вивчення поводження арифметичного зсувача мантис у додатковому коді під дією типових несправностей необхідно побудувати інформативну, функціональну й достовірну програмну модель пристрою.

Основне призначення програмної моделі складається в достовірній оцінці ймовірності появи помилок на послідовностях вхідних слів. Достовірність оцінки повинна забезпечуватися достовірністю програмної моделі арифметичного зсувача мантис та коректною постановкою експериментів.

Для одержання достовірної програмної моделі необхідно забезпечити багатобічну перевірку її функціонування. Перевірка вимагає введення режимів, дозволяють здійснити тестування й налагодження програмної моделі: виконання окремих операцій на заданих значеннях операнда мантиси. Важливу роль в організації перевірок грає демонстраційна складова моделі.

Коректна оцінка ймовірностей появи помилок вимагає організації експериментів на випадкових послідовностях вхідних слів до одержання сталих результатів моделювання.

4 РОЗРОБКА ПРОГРАМНОЇ МОДЕЛІ АРИФМЕТИЧНОГО ЗСУВАЧА

4.1 Алгоритм моделювання паралельного арифметичного зсувача

4.1.1 Моделювання виконується на заданому наборі мантиси операнда та величини зсуву для випадків зсуву при обраному місці й виді несправності типу «закоротка».

На рис. 4.1 показаний перший фрагмент схеми алгоритму моделювання паралельного арифметичного зсувача мантис.

Блок 2 вводить ознаки несправності *ff*, масивів *A, B* операндів, їх розмірів *n* та *m*.

Блок 3 здійснює перебирання рівнів розпаралелення від 1 до *m*.

Блок 4 здійснює обчислення 2^x по формулі x=*1 shl r*.

Блоки 5 – 8 виконують послідовне формування контрольних точок зсуву.

Блок 5 виконує обчислення по формулі *T[r]=4*r+1*, де *r* – рівень розпаралелення.

Блок 6 виконує обчислення по формулі *T[r]=T[r] +(r+2)*x*, де *r* – рівень розпаралелення, *T[r]* – результат обчислення блоку 5.

Блок 7 виконує обчислення по формулі *T[r] = T[r]+10*(m–r)*, де *r* – рівень розпаралелення, *m* – кількість рівнів розпаралелення, *T[r]* – результат обчислення блоку 6.

Блок 8 виконує обчислення по формулі *T[r] =T[r]*n, n* – кількість одноадресних мультиплексорів, *T[r]* – результат обчислення блоку 7.

Блок 9 здійснює випадкове завдання місця несправності схеми паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 10 здійснює випадкове завдання виду одиночної константної несправності цифрової схеми паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

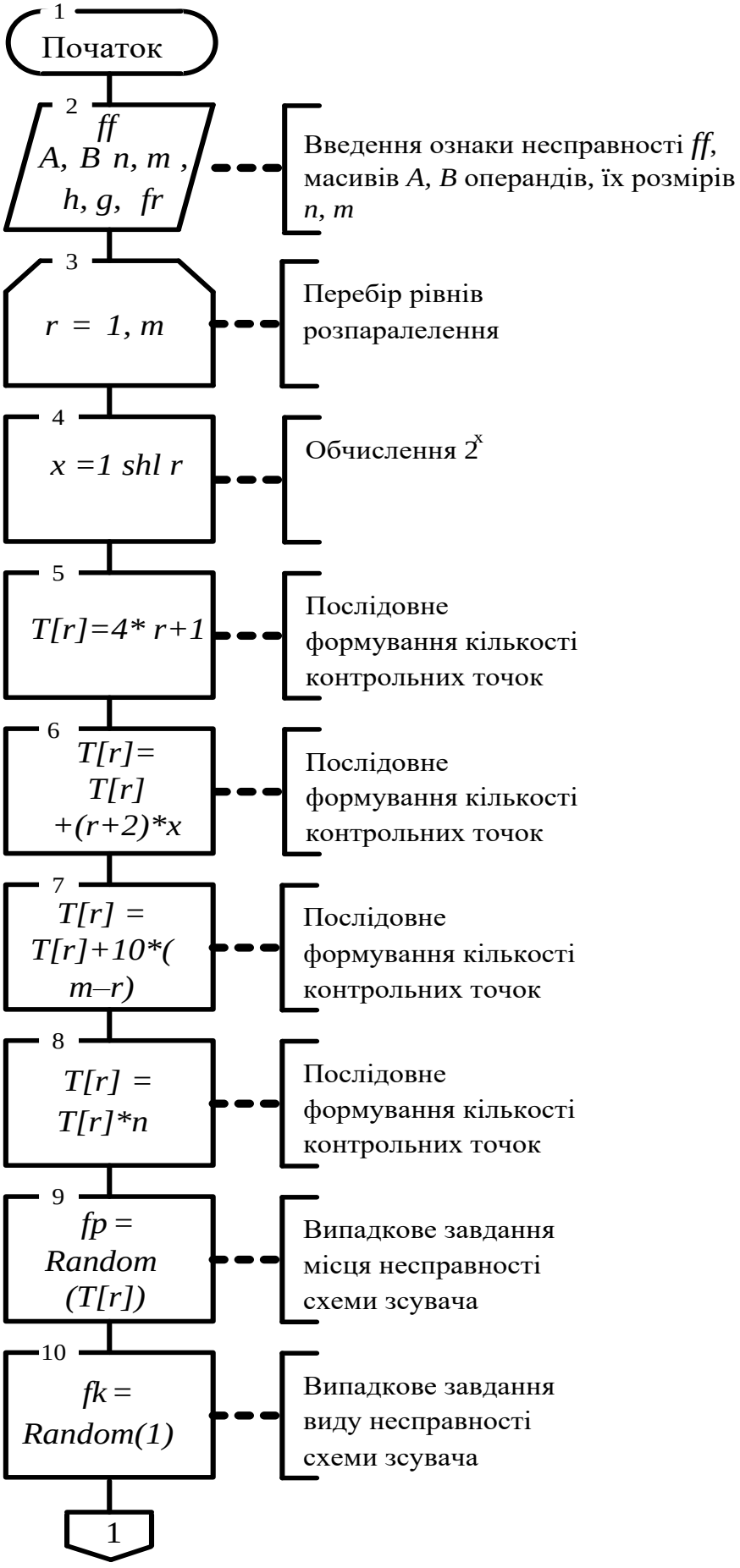

Рис. 4.1 – Перший фрагмент схеми алгоритму моделювання

4.1.2 Наступні блоки схеми алгоритму моделювання наведені в другому фрагменті, що показаний на рис. 4.2.

Блок 11 здійснює обчислення кількості адресних точок паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч за формулою *g1=4*r*n,* де *n* – кількість одноадресних мультиплексорів, *r* – рівень розпаралелення.

Блок 12 здійснює обчислення кількості точок паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч без виходів мультиплексорів першого ярусу за формулою *g2=g1+n*x*(r+2),* де *n* – кількість одноадресних мультиплексорів, *r* – рівень розпаралелення.

Блок 13 здійснює обчислення кількості контрольних точок ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч за формулою *g3=g2+n,* де *n* – кількість одноадресних мультиплексорів.

Блок 14 здійснює перевірку режиму несправності та її нульового значення. Якщо режим справний або значення *fk* відрізняється від 0, то здійснюється міжсторінковий перехід на наступний фрагмент (рис. 4.9).

Блок 15 здійснює перебирання номерів розрядів адреси першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч від 1 до *r.*

Блок 16 здійснює перебирання мультиплексорів першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч від 1 до *n.*

Блок 17 здійснює перевірку місця несправності.

Блок 18 здійснює введення несправності, що задає нульове значення в розряд адреси першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 19 завершує перебирання мультиплексорів першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 20 завершує перебирання розрядів адреси паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

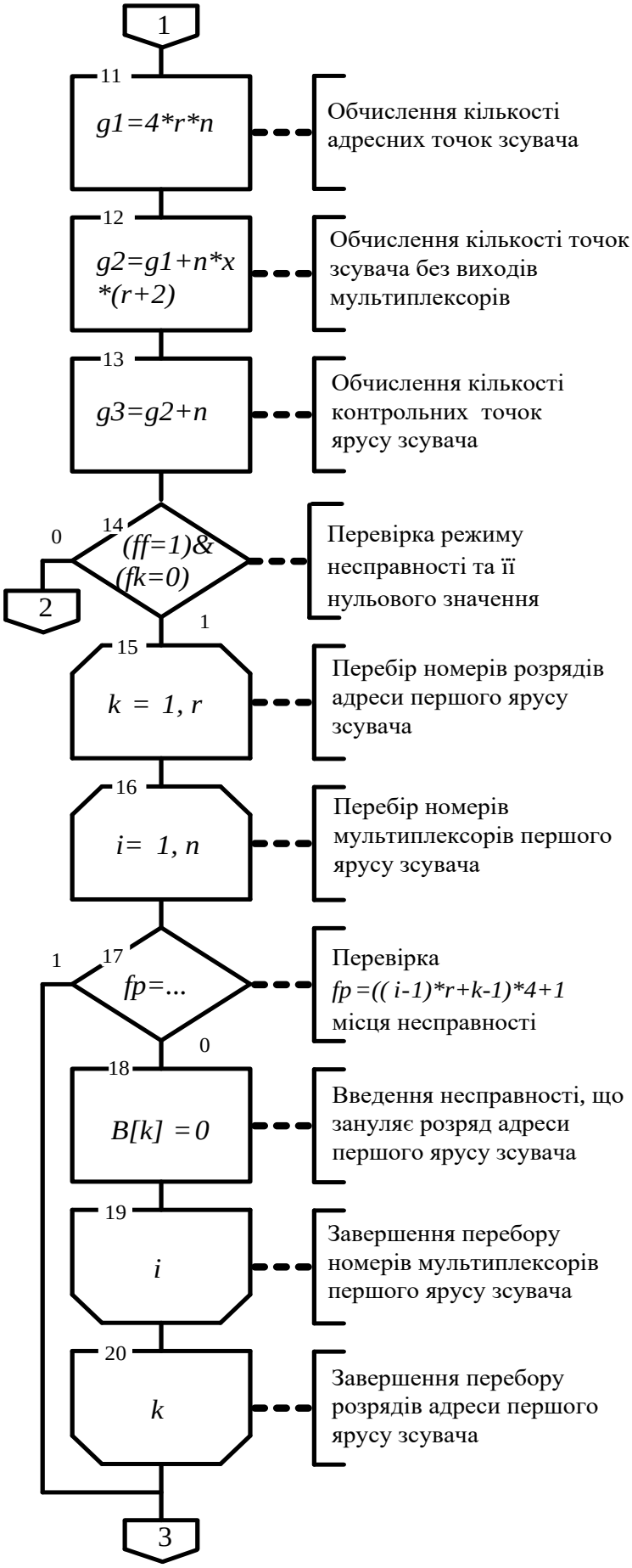

Рис. 4.2 – Другий фрагмент схеми алгоритму моделювання

4.1.3 Наступні блоки схеми алгоритму моделювання наведені в третьому фрагменті, що показаний на рис. 4.3.

Блок 21 здійснює перебирання номерів мультиплексорів першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч від 1 до *n*.

Блок 22 здійснює перебирання входів елементів I мультиплексорів першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 23 здійснює перевірку місця несправності.

Блок 24 здійснює перевірку несправності елементів I. Якщо $i>j$, то йдемо на блок 26.

Блок 25 задає нульове значення знаковому розряду s якщо виконується умова $i<j$ в умовному блоку 24.

Блок 26 здійснює введення несправності якщо не виконується умова $i<$j в умовному блоку 24. Тоді розряд $A\ [i–j+1]$ дорівнюємо 0 і повертаємось знову на блок 23.

Блок 27 завершує перебирання входів елементів I мультиплексорів першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 28 завершує перебирання номерів мультиплексорів першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 29 здійснює перебирання номерів розрядів адреси наступних ярусів одноадресних мультиплексорів паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 30 здійснює перебирання номерів одноадресних мультиплексорів поточного ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 31 здійснює обчислення базового номеру точки несправності на входах адреси мультиплексорів паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

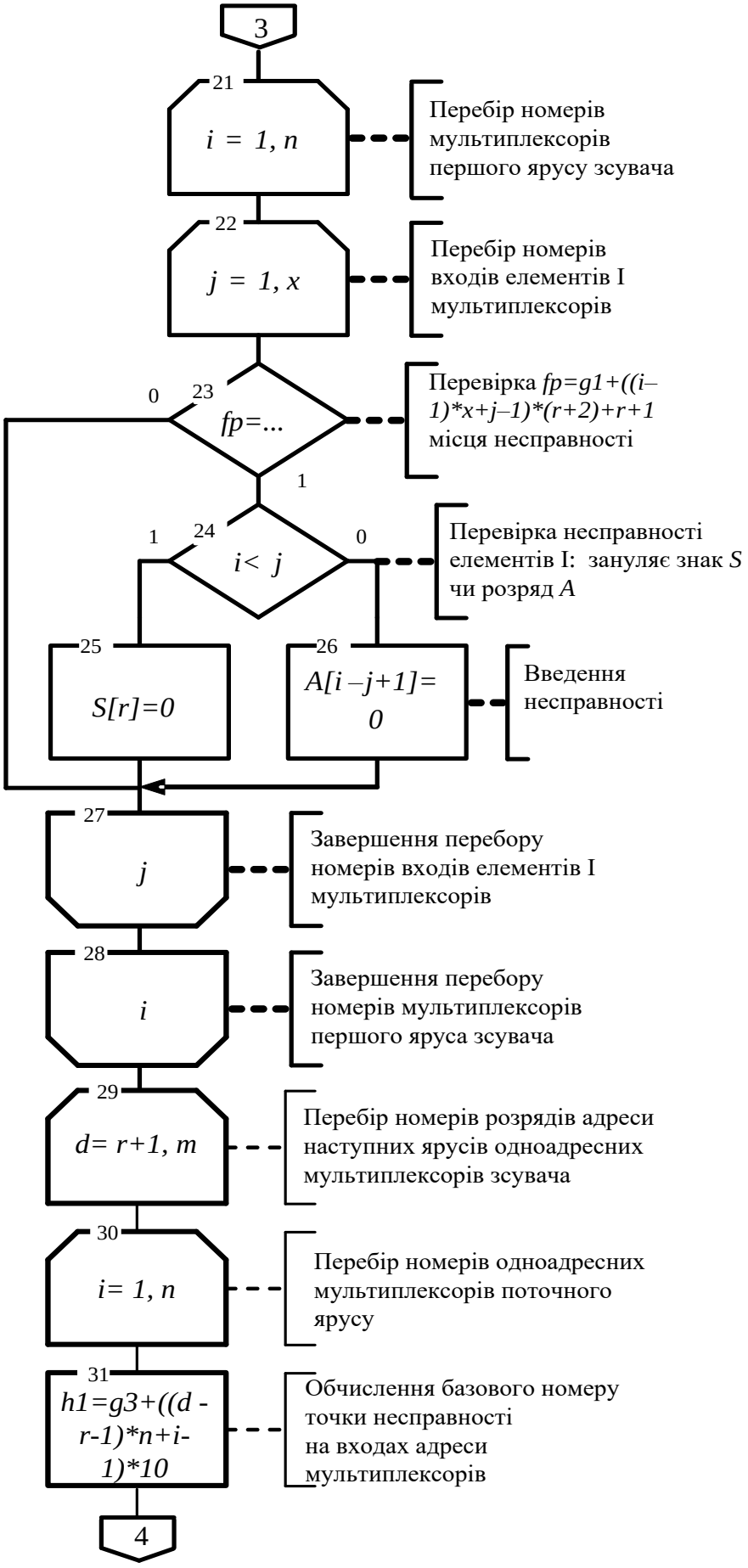

Рис. 4.3 – Третій фрагмент схеми алгоритму моделювання

4.1.4 Наступні блоки схеми алгоритму моделювання наведені в четвертому фрагменті, що показаний на рис. 4.4.

Блок 32 здійснює обчислення позиції, до якої вивільнена при зсуві частина операнда заповнюється знаковим розрядом, по формулі *h3=1 shl(d-1),* де *d* - номер розрядів адреси наступних ярусів одноадресних мультиплексорів паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч, shl – зсув операнда ліворуч.

Блок 33 здійснює перевірку місця несправності по формулі *fp =h1+1*, де *h1* - базовий номер точки несправності на входах адреси мультиплексорів.

Блок 34 вводить несправність, що задає нульове значення у розряді величини зсуву, якщо не виконується умова попереднього блоку 33 та здійснюється міжсторінковий перехід на наступний фрагмент (рис. 4.5).

Блок 35 здійснює перевірку місця несправності по формулі *fp =h1+6* при виконанні умови блоку 33, де *h1* – базовий номер точки несправності на входах адреси мультиплексорів.

Блок 36 вводить несправність, що задає нульове значення у розряді мантиси величини після зсуву на попередніх ярусах, при виконанні умови попереднього блоку 35.

Блок 37 здійснює перевірку умови *(fp=h1+7)and(i<=h3),* що визначає місце несправності, де *fp* – місце несправності, *h1* – базовий номер точки несправності на входах адреси мультиплексорів, *i* – номер одноадресних мультиплексорів поточного ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч, *h3* – номер позиції, до якої вивільнена при зсуві частина операнда заповнюється знаковим розрядом.

Блок 38 вводить несправність, що задає нульове значення у знаковому розряді мантиси при виконанні умови попереднього блоку 37.

Блок 39 завершує перебирання номерів мультиплексорів поточного ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 40 завершує перебирання номерів розрядів адреси наступних ярусів паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

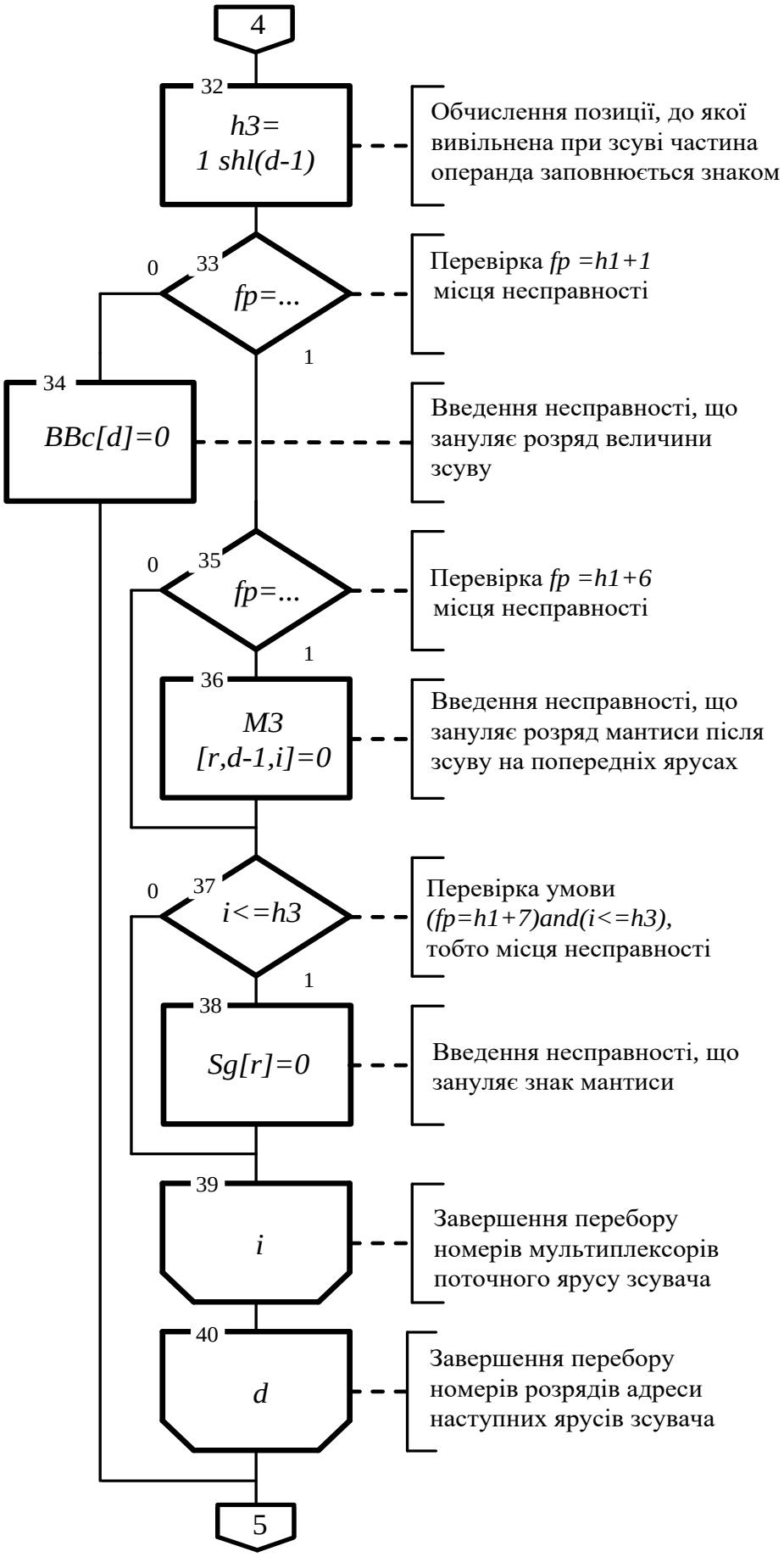

Рис. 4.4 – Четвертий фрагмент схеми алгоритму моделювання

4.1.5 Наступні блоки схеми алгоритму моделювання наведені в п’ятому фрагменті, що показаний на рис. 4.5.

Блок 41 здійснює перебирання номерів мультиплексорів першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 42 здійснює перебирання номерів розрядів адреси мультиплексорів першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч від *1* до *r*.

Блок 43 здійснює обчислення базового номеру точки несправності за формулою *h1=((i-1)*r+k-1)*4*, де *i* – номер мультиплексорів першого ярусу, *r* – рівень розпаралелення, k - номер розрядів адреси мультиплексорів першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 44 здійснює обчислення значення виходу першого інвертора на адресних входах мультиплексорів.

Блок 45 здійснює перевірку стану режиму „справно - несправно”. Якщо значення *ff=1,* то стан паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч знаходиться у режимі „несправно”. Інакше здійснюється міжсторінковий перехід на наступний фрагмент (рис. 4.6).

Блок 46 здійснює перевірку умови *fp= h1+1 and fk=1*, що визначає місце та вид несправності.

Блок 47 вводить несправність, що задає нульове значення знаковому розряду паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 48 здійснює перевірку місця несправності по формулі *fp =h1+2*.

Блок 49 вводить несправність, що встановлює розряд адреси першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч в значення *fk*.

Блок 50 здійснює перевірку умови *(fp=h1+3) and (fk=1)*, що визначає місце та вид несправності.

Блок 51 вводить несправність, що задає нульове значення знаковому розряду адреси першого ярусу паралельного арифметичного зсувача.

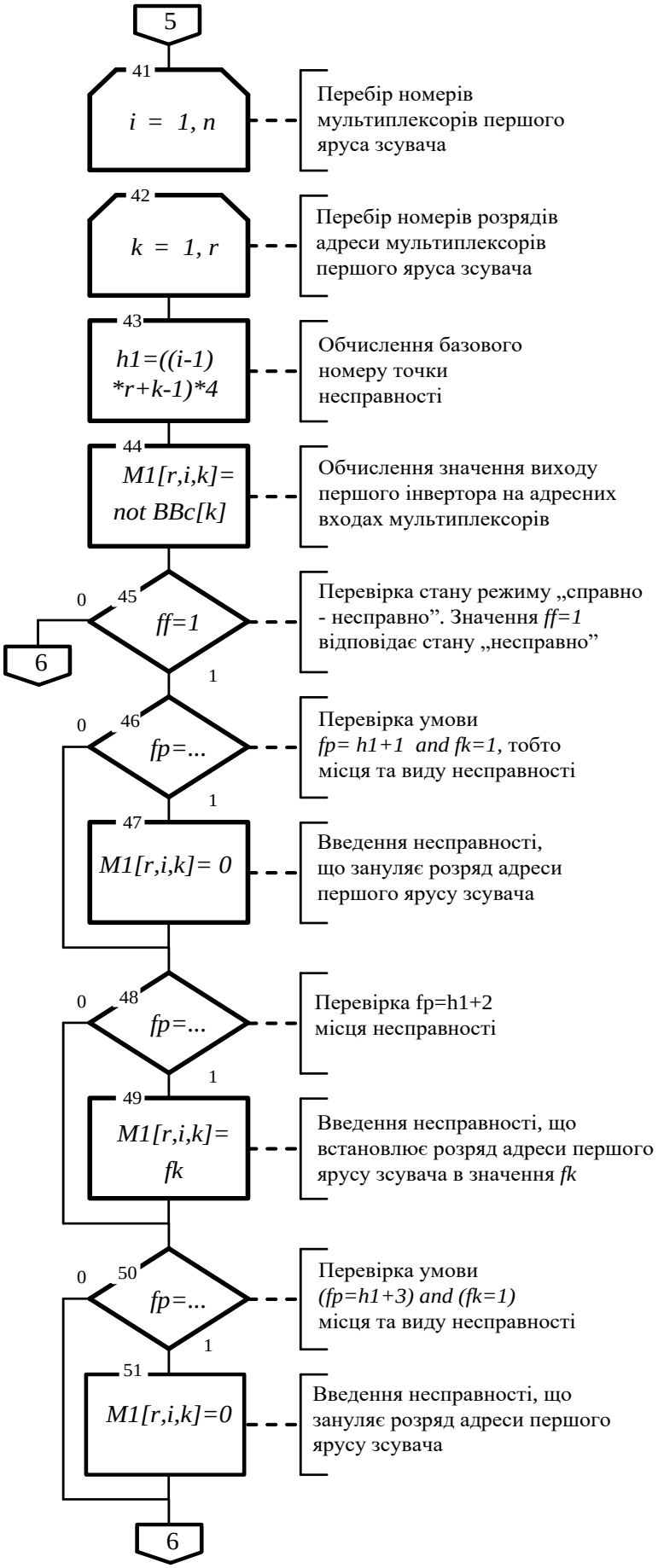

Рис. 4.5 – П'ятий фрагмент схеми алгоритму моделювання

4.1.6 Наступні блоки схеми алгоритму моделювання наведені в шостому фрагменті, що показаний на рис. 4.6.

Блок 52 здійснює перевірку умови *(ff=1) and fk=0*, що визначає місце та вид несправності. Якщо умова не виконується, то здійснюється перехід на блок 58.

Блок 53 здійснює перебирання номерів входів елементів І мультиплексорів першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч від *1* до *x*.

Блок 54 здійснює обчислення значення розряду адреси мультиплексора при несправності на вході елемента І.

Блок 55 здійснює перевірку умови *fp=g1+((i-1)*x+j-1)*(r+2) +k) and (BBd[k])=0*.

Блок 56 вводить несправність, що задає нульове значення знаковому розряду адреси першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч при виконанні умови попереднього блоку 55.

Блок 57 завершує перебирання номерів входів елементів І мультиплексорів.

Блок 58 здійснює обчислення значення виходу другого інвертора на адресних входах мультиплексорів першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 59 здійснює перевірку умови *ff=1 and fk=0*, що визначає,що стан паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч знаходиться у режимі „несправно” , та виду несправності. Якщо умова не виконується, то здійснюється міжсторінковий перехід на наступний фрагмент (рис. 4.7).

Блок 60 здійснює перебирання номерів входів елементів І мультиплексорів першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч від *1* до *x* при виконанні умови попереднього блоку 59.

Блок 61 здійснює обчислення розряду адреси мультплексора при несправності на вході елемента І.

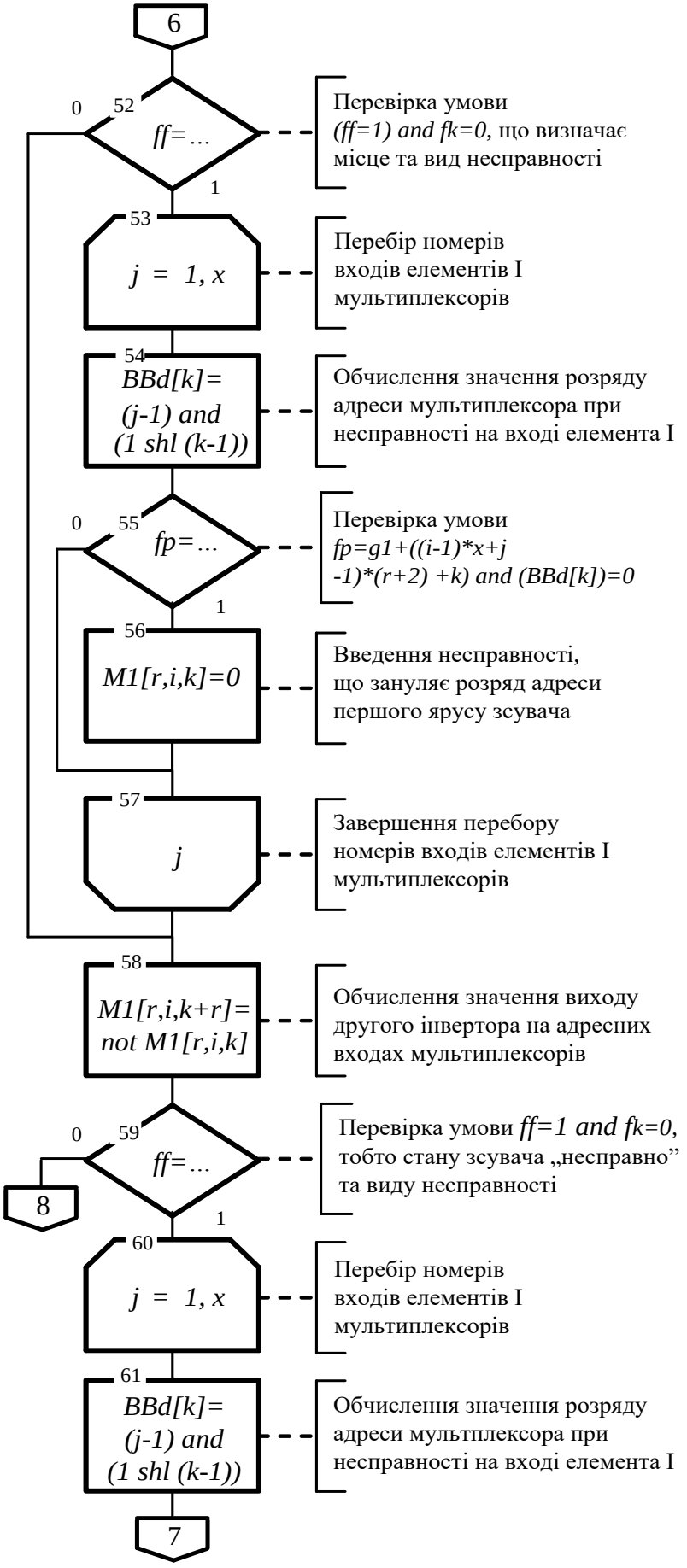

Рис. 4.6 – Шостий фрагмент схеми алгоритму моделювання

4.1.7 Наступні блоки схеми алгоритму моделювання наведені в сьомому фрагменті, що показаний на рис. 4.7.

Блок 62 здійснює перевірку умови *fp=g1+((i-1)*x+j-1)*(r+2)+k and BBd[k]=1*, де *i* – номер мультиплексорів першого ярусу, *r* – рівень розпаралелення, k - номер розрядів адреси мультиплексорів першого ярусу, g1 – кількість адресних точок паралельного арифметичного зсувача.

Блок 63 вводить несправність, що задає нульове значення знаковому розряду адреси першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч при виконанні умови попереднього блоку 62.

Блок 64 завершує перебирання номерів входів елементів І мультиплексорів.

Блок 65 здійснює перевірку умови *ff=1*, що визначає стан паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч „несправно”. Якщо умова не виконується, то здійснюється перехід на блок 71.

Блок 66 здійснює перевірку умови *ff=1, (fp=h1+3) and (fk=1)*, що визначає місце та вид несправності, де *h1* – базовий номер точки несправності на входах адреси мультиплексорів.

Блок 67 вводить несправність, що задає нульове значення знаковому розряду адреси першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч при виконанні умови попереднього блоку 66.

Блок 68 здійснює перевірку місця несправності по формулі *fp =h1+4*, де *h1* - базовий номер точки несправності на входах адреси мультиплексорів.

Блок 69 вводить несправність, що присвоює значення *fk* розряду адреси першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч при виконанні умови попереднього блоку 68.

Блок 70 завершує перебирання розрядів адреси мультиплексорів першого ярусу паралельного арифметичного зсувача нормалізованої мантиси.

Блок 71 здійснює завдання початкового нульового значення результату диз’юнкції виходів елементів І.

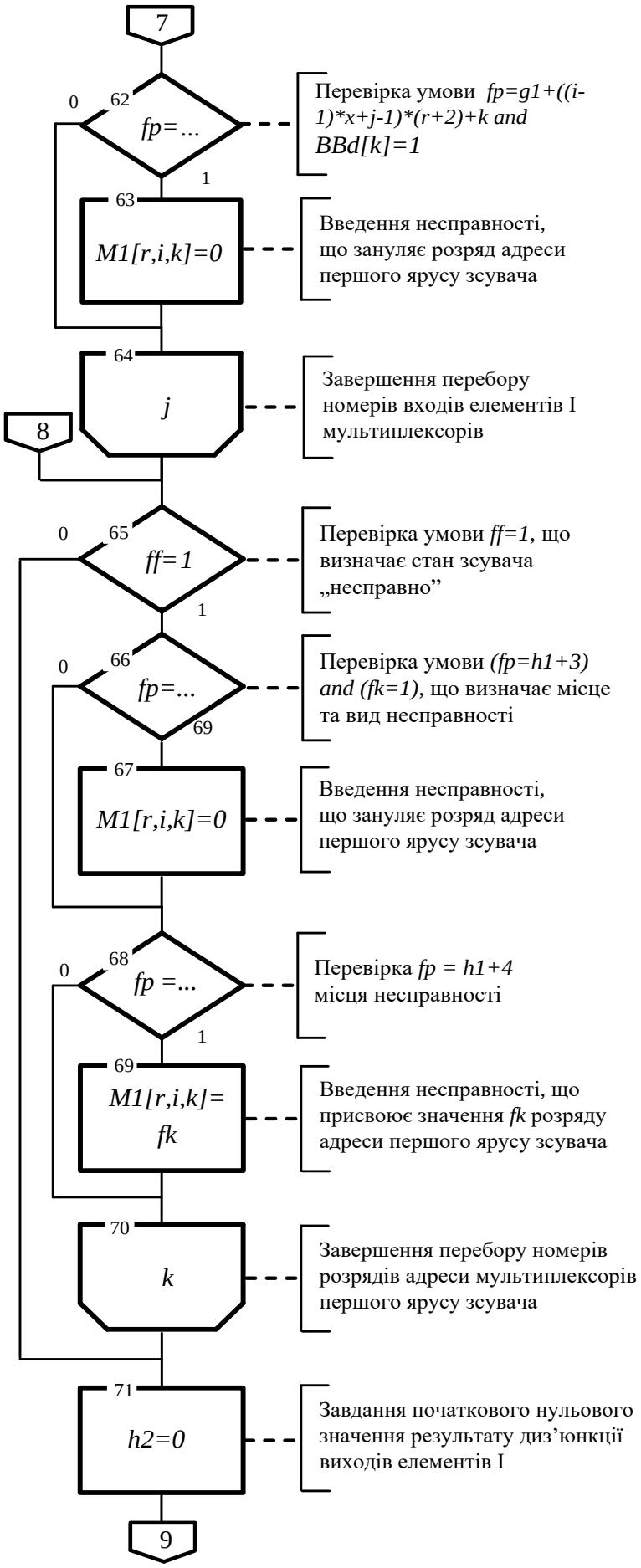

Рис. 4.7 – Сьомий фрагмент схеми алгоритму моделювання

4.1.8 Наступні блоки схеми алгоритму моделювання наведені в восьмому фрагменті, що показаний на рис. 4.8.

Блок 72 здійснює перебирання номерів входів елементів І мультиплексорів паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч від *1* до *x*.

Блок 73 здійснює перевірку позиції зсунутої мантиси операнда на заповнення її знаковим розрядом.

Блок 74 здійснює обчислення значення позиції якщо її місце займає розряд мантиси операнда.

Блок 75 здійснює обчислення значення позиції якщо її місце займає знак мантиси операнда.

Блок 76 здійснює присвоєння значення позиції розряду зсунутої мантиси операнда.

Блок 77 здійснює перевірку умови *(ff=1) and(fp=g1+((i-1)*x+j-1) *(r+2)+r+1)and(fk=1)*.

Блок 78 вводить несправність, що присвоює значення „1" виходу *j*-го елемента І *i*-го мультиплексора при виконанні умови попереднього блоку 77.

Блок 79 визначає номер елемента І *i*-го мультиплексора за формулою *h4=j-1*, де *j* – номер входу елементів І мультиплексорів.

Блок 80 здійснює перебирання номерів входів елементів І мультиплексорів першого ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч від *1* до *r*.

Блок 81 здійснює обчислення ознаки прямого або інверсного входу елемента І.

4.1.9 Наступні блоки схеми алгоритму моделювання наведені в дев’ятому фрагменті, що показаний на рис. 4.9.

Блок 82 здійснює підготовку обчислення ознаки прямого або інверсного наступного входу елемента І .

Блок 83 здійснює обчислення за формулою *M2[r,i,j,k]= M1[r,i,k] and not BBd[k] or M1[r,i,k+ r] and BBd[k])*.

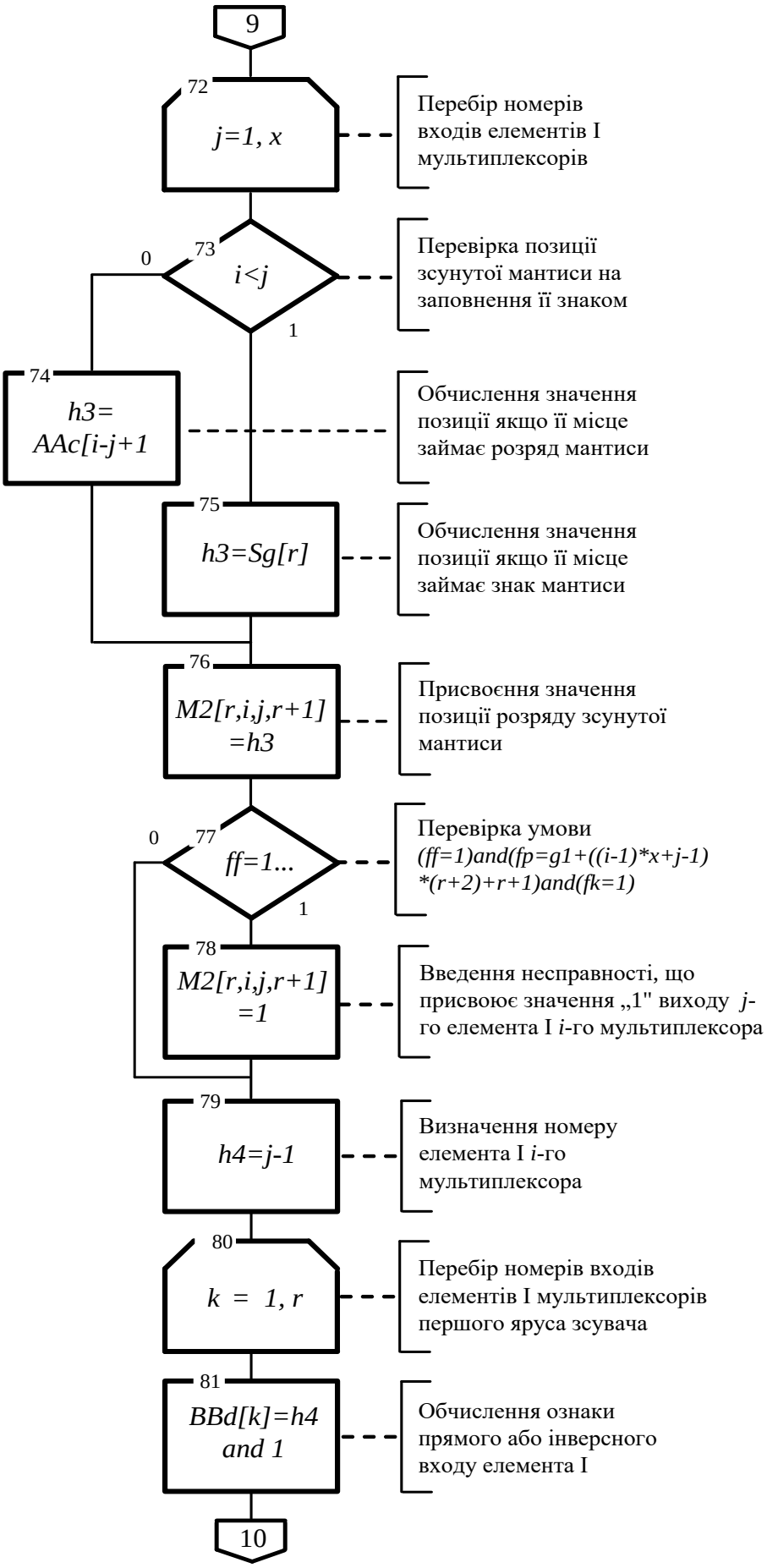

Рис. 4.8 – Восьмий фрагмент схеми алгоритму моделювання

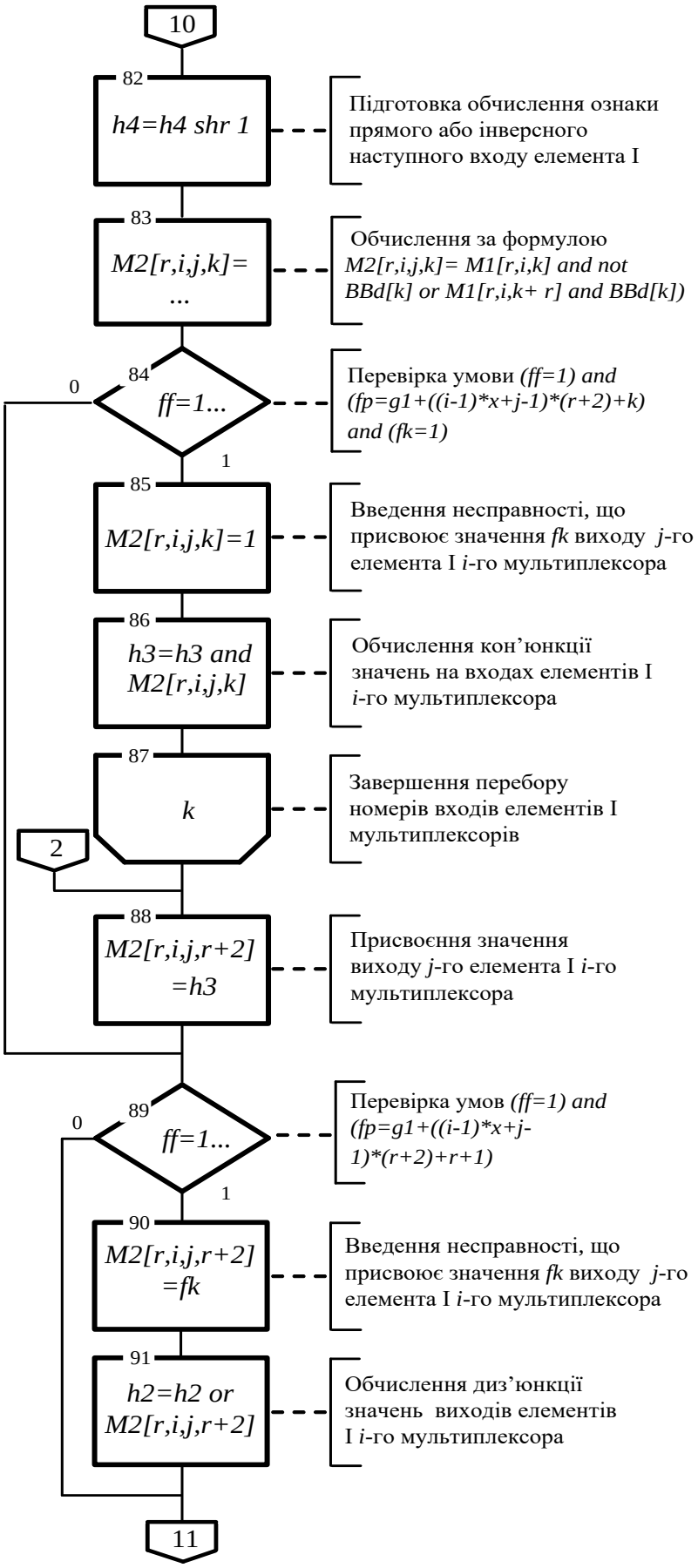

Рис. 4.9 – Дев'ятий фрагмент схеми алгоритму моделювання

Блок 84 здійснює перевірку умови *(fp=g1+((i-1)*x+j-1)*(r+2)+k) and (fk=1) and (ff=1).* Якщо умова не виконується, то здійснюється перехід на блок 89.

Блок 85 вводить несправність, що присвоює значення *fk* виходу *j*-го елемента І *i*-го мультиплексора при виконанні умови попереднього блоку 84.

Блок 86 обчислює кон'юнкції входів елементів І *i*-го мультиплексора.

Блок 87 завершує перебирання номерів входів елементів І мультиплексорів.

Блок 88 здійснює присвоєння значення виходу *j*-го елемента І *i*-го мультиплексора.

Блок 89 здійснює перевірку умови *(ff=1) and (fp=g1+((i-1)*x+j-1)*(r+2)+r+1).* Якщо умова не виконується, то здійснюється перехід на міжсторінковий перехід на наступний фрагмент (рис. 4.10).

Блок 90 вводить несправність, що присвоює значення значення *fk* виходу *j*-го елемента І *i*-го мультиплексора.

Блок 91 обчислює диз'юнкції виходів елементів І *i*-го мультиплексора.

4.1.10 Наступні блоки схеми алгоритму моделювання наведені в десятому фрагменті, що показаний на рис. 4.10.

Блок 92 завершує перебирання номерів входів елементів І мультиплексорів.

Блок 93 здійснює присвоєння значення виходу *i*-го мультиплексора.

Блок 94 здійснює перевірку умови *(ff=1) and (p=g2+i).*

Блок 95 вводить несправність, що встановлює значення *fk* виходу *i*-го мультиплексора при виконанні умови попереднього блоку 94.

Блок 96 здійснює перевірку умови *r=m* – чи оброблюється останній ярус паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 97 здійснює присвоєння значення виходу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 98 завершує перебирання номерів мультиплексорів поточного ярусу паралельного арифметичного зсувача нормалізованої мантиси операнда праворуч.

Блок 99 завершує перебирання рівнів розпаралелення.

Виведення масиву розрядів зсунутої мантиси операнда

Алгоритм закінчений.

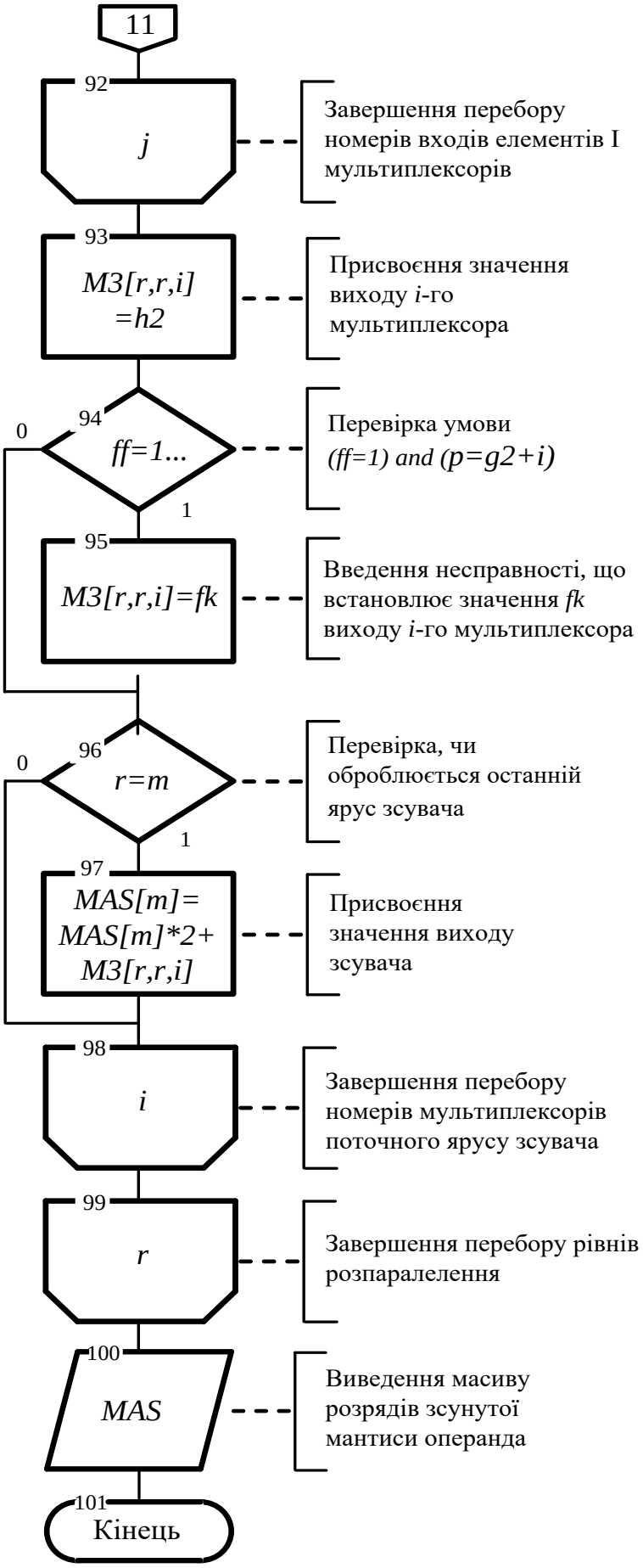

Рис. 4.10 – Десятий фрагмент схеми алгоритму моделювання

4.2 Функціонування моделі паралельного арифметичного зсувача

4.2.1 Програма запускається з довжиною адреси зсувача 4, та задамо значення операнда мантиси *A* та величини зсуву *B* у справному стані зсувача. Якщо лівий розряд мантиси у двійковому коді дорівнює нулю, то кількість розрядів ліворуч, яка дорівнює величини зсуву заповнюється одиницями. Якщо лівий розряд мантиси у двійковому коді дорівнює 1, то кількість розрядів ліворуч, яка дорівнює величини зсуву заповнюється нулями. Результат мантиси, яка залишилась після зсуву, у десятковій системі записаний у полі *C*. Результат у полі *Ce* дорівнює результату у полі *C*, так як зсувач знаходиться у справному стані. Достовірність результатів дорівнює 100%, так як немає помилок.

Результат роботи програми зображено на рис. 4.11.

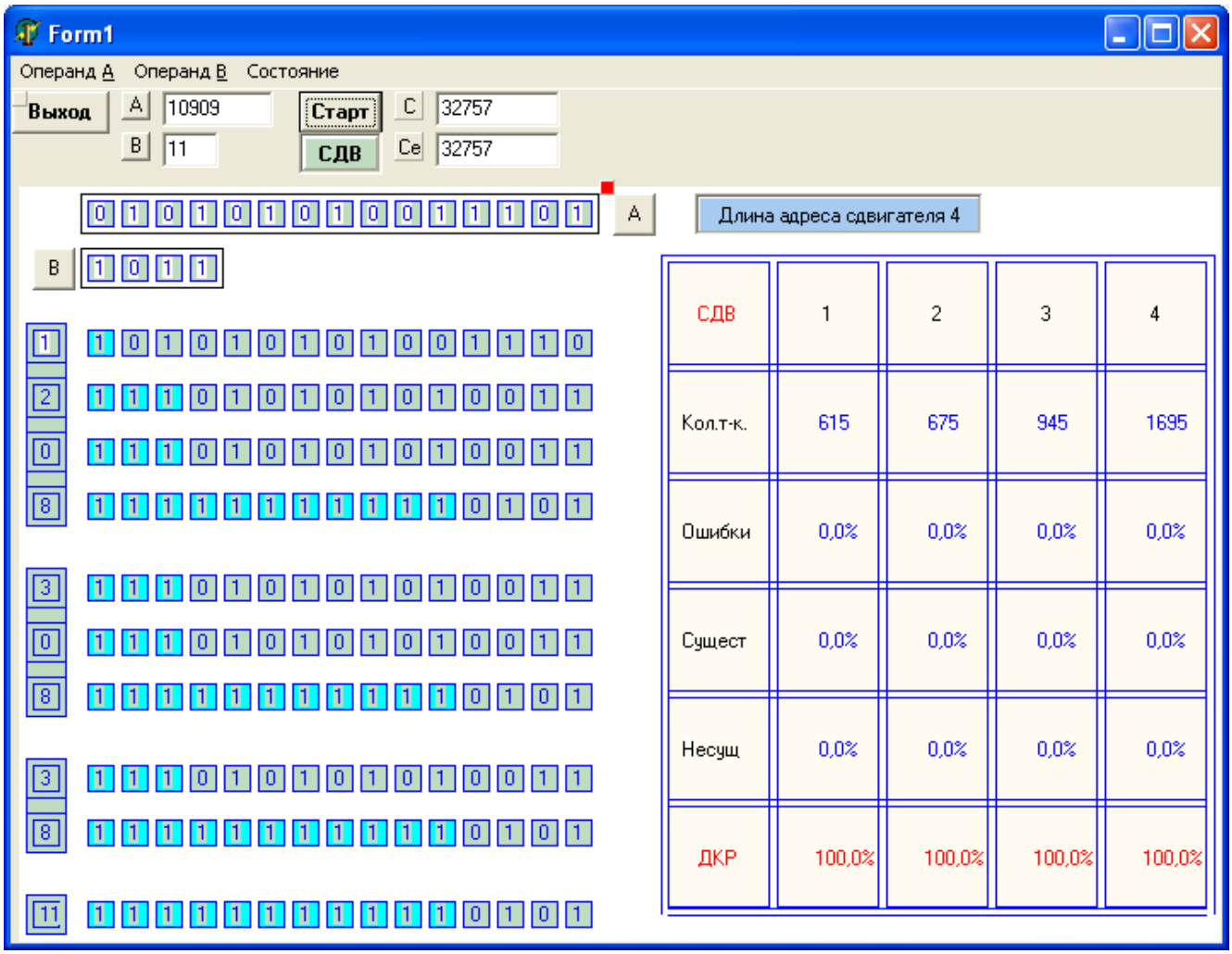

Рис. 4.11 – Результат роботи програми зсувача у справному стані

4.2.2 На рис 4.12 показано роботу програми при несправному стані зсувача. У четвертому ярусі першої схеми з-за несправності не виконується зсув на 8 позицій та виникає помилка. У даному полі видно, що достовірність у схемі чотирьохярусного зсувача дорівнює 0 з-за помилки (не виконання зсуву) та результат у десятковій системі Се відрізняється від правильного результату. Достовірність результатів у інших ярусах дорівнює 100%, так як у них немає помилок. Результат роботи програми зображено на рис. 4.12.

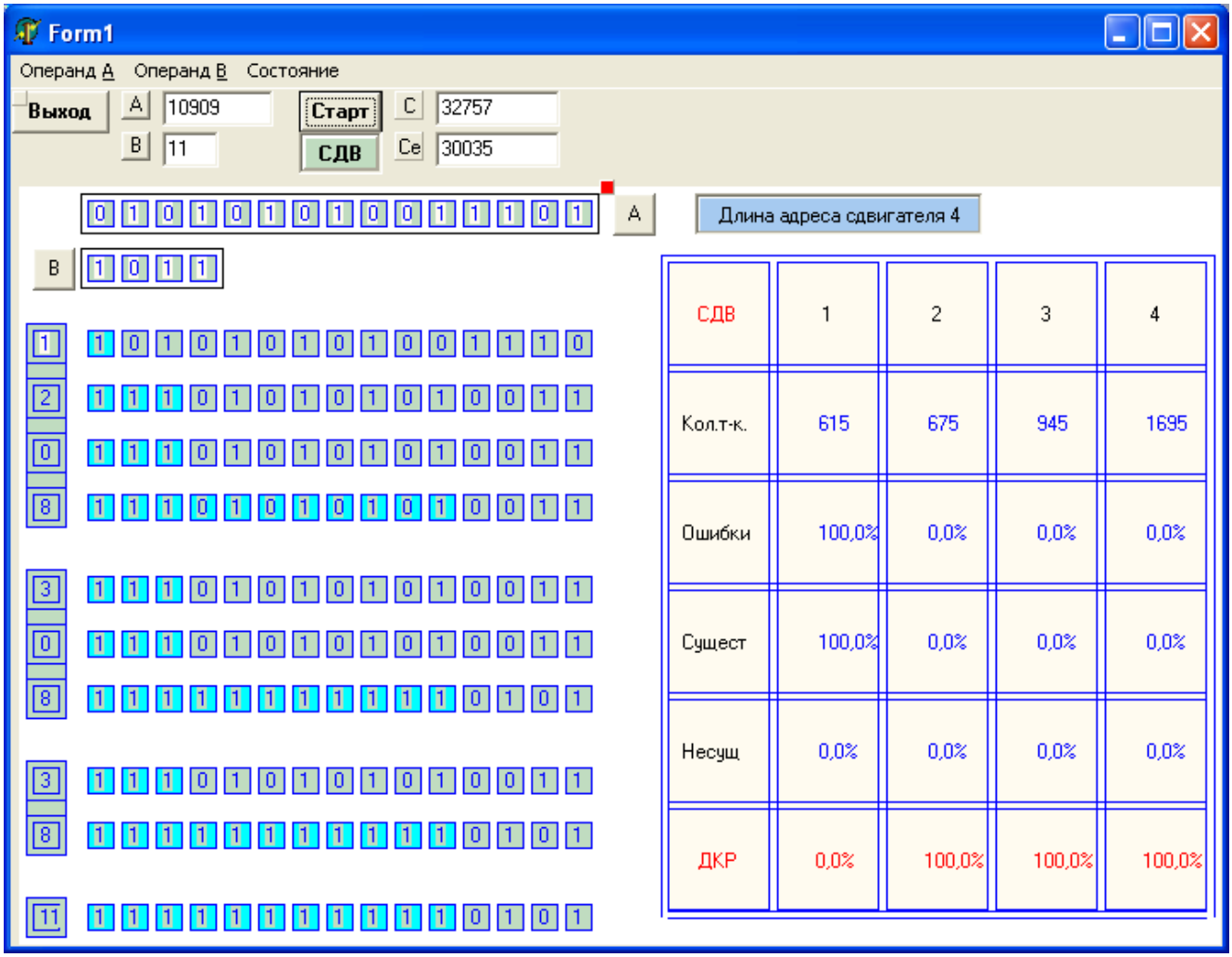

Рис. 4.12 – Результат роботи програми зсувача у несправному стані

4.2.3 Програмна модель дозволяє встановити розподіл результату зсуву на вірні та невірні розряди. На рис 4.13 показано переміщення показчику розподілу на 8 позицій ліворуч. Помилки у розрядах праворуч є несуттєвими та не впливають на достовірність. Також задається несправний стан пристрою.

Рисунок показує, що з-за несправності у першій та третій схемі при зсуві один з розрядів дорівнює 0 замість 1, але помилка у зсуві виникає праворуч від показчика і тому є несуттєвою та достовірність результатів дорівнює 100%.

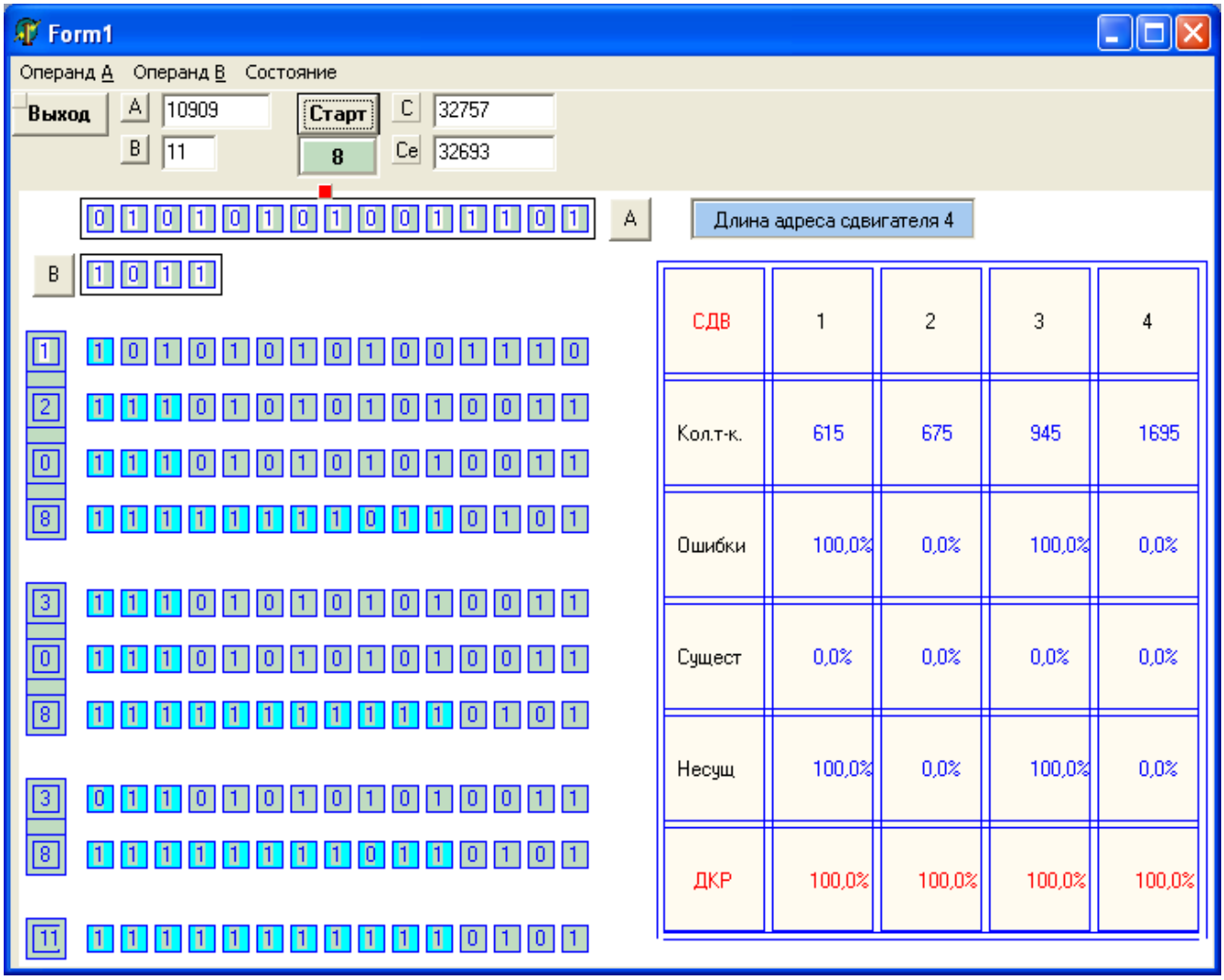

Рис. 4.13 – Результат роботи програми зсувача у несправному стані при положенні показчику розподілу розрядів на вірні та невірні на 8 позицій

На рис 4.14 показано виконання програми з повним перебором значень операнда мантиси *A* та величини зсуву *B* у несправному стані зсувача.

Найменшу ймовірність появи помилки та найбільшу достовірність результатів має схема з одноярусним зсувачем. Схема з чотирьохярусним зсувачем навпаки має найменшу достовірність результатів та найменший відсоток кількості помилок.

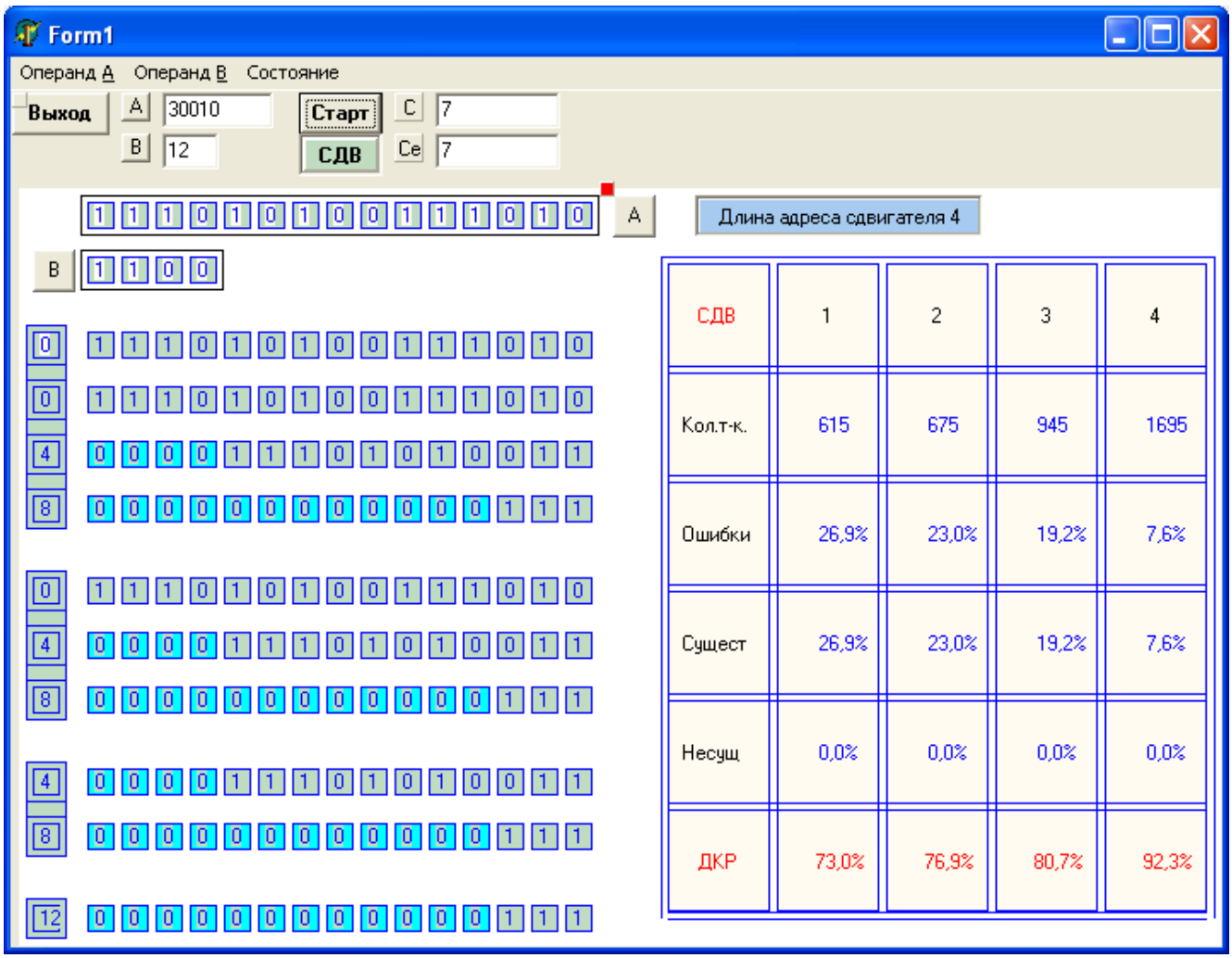

Рис. 4.14 – Результат роботи програми зсувача у несправному стані при повному переборі значень операнда мантиси *A* та величини зсуву *B*

4.3 Висновки

Алгоритм моделювання операції зсуву мантис у матричному пристрої імітує для заданого набору операнда мантиси та величини зсуву почергову роботу всіх осередків паралельного арифметичного зсувача мантис у додатковому коді.

Вихідними даними до алгоритму є мантиса операнд та величина зсуву у вигляді масивів двійкових розрядів, а також їх розмірів *n* та *m*. За цим даними з урахуванням уведеної несправності обчислюється достовірність результатів.

5 РЕЗУЛЬТАТИ ПРОВЕДЕНИХ ДОСЛІДЖЕНЬ

5.1 Програмна модель паралельного арифметичного зсувача мантис

5.1.1 Керівництво програміста

Призначення. Програмний продукт (ПП) призначений для моделювання поводження паралельного арифметичного зсувача мантис в додатковому коді в умовах дії характерних несправностей типу «замикання».

Склад. Програмний продукт написаний на Delphi, використовує одну форму та містить 29 процедур. Вихідні дані. Розрядність мантис n задається у вигляді значення константи. Інші дані визначаються в процесі роботи ПП.

Результати виконання програмного продукту. Результати моделювання зберігаються на екрані.

5.1.2 Керівництво користувача

5.1.2.1 Установка та запуск. Програмний продукт поставляється у вигляді завантажувального модуля для заданої розрядності паралельного арифметичного зсувача та запускається на виконання по команді ENTER.

5.1.2.2 Користувальницький інтерфейс.

Вихідними даними для проведення дослідження є такі.

– розрядність мантиси n та величини зсуву r, $n = 2^r$;

– варіанти схеми: від схеми з r ярусів до одноярусної схеми паралельного арифметичного зсувача;

– несправність – константна одиночна (нуль чи одиниця);

– поява несправності в точках схеми – рівноймовірна за місцем появи та рівнем сигналу;

– розподіл розрядів результату на вірні та невірні визначається у межах, що задаються;

– поява суттєвих та несуттєвих помилок визначається при рівній ймовірності появи будь-якого значення нормалізованої мантиси та величини зсуву;

– дослідження моделі проводиться до одержання стійких результатів.

При запуску програми з’являється панель, на якій розташовано кнопки «Вход» та «Выход». Натискання кнопки «Выход» призводить до завершення програми.

При натисканні кнопки «Вход» на екрані з’являється панель «Длина алреса сдвигателя (2-5)» та вікно під цією панеллю із значенням довжини адреси зсувача «4», що приймається за замовчуванням.

Користувач має можливість змінити довжину адреси зсувача в межах з 2 до 5. Крім того, з’являється панель «Дальше», при натисканні якої здійснюється перехід на перша основна панель – моделювання паралельного арифметичного зсувача.

На панелі моделювання паралельного арифметичного зсувача мантис розташовано:

– меню режимів завдання операндів і стану пристрою;

– поля для завдання операнда *A* та величини зсуву *B* у десятковій системі числення, що за замовчуванням приймають нульові значення, а також кнопки для прийому записаних значень;;

– поля для завдання операнда *A* та величини зсуву *B* двійковим кодом, а також кнопки для прийому записаних значень;

– структури паралельного арифметичного зсувача, що відповідає використаній розрядності пристрою;

– панель «Длина алреса сдвигателя» з вказаним десятковим значенням довжини адреси зсувача.

– кнопка «Старт» для початку моделювання роботи паралельного арифметичного зсувача;

– покажчик розподілу результату зсуву на вірні та невірні розряди.

Процес моделювання починається з установки режимів завдання операндів вибором з меню кожного співмножника одного з варіантів:

– значення;
– довільне значення;
– повний перебір значень;
– послідовність довільних значень.

За допомогою меню встановлюється також стан пристрою – «справно» або «несправно».

При виборі режиму «значення» операнду привласнюється значення завданням десяткового числа або двійкового коду.

При виборі режиму «несправно» встановлюються режими довільного вибору виду та місця несправності.

Допускається змінити положення покажчика розподілу результату на вірні та невірні розряди в межах *n* старших розрядів результату зсуву.

Далі натискається кнопка «Старт» та починається моделювання роботи паралельного арифметичного зсувача.

У випадку завдання одного конкретного або випадкового набору операнда та величини зсуву моделювання виконується з показом зсунутих значень мантиси у різних структурах паралельного арифметичного зсувача.

Зсунуті мантиси приймають значення, що обчислюються під впливом несправності, якщо був обраний режим «несправно».

Крім того, показуються десяткові значення результатів.

У режимах повного перебору значень операндів кнопка «Старт» міняє назву на на наступну назву «Фініш», при натисканні якої моделювання зупиняється також як при завершенні завдань у режимах повного перебору одного або двох операндів.

У режимах серії випадкових значень перебору значень користувач сам зупиняє процес моделювання, натискаючи на кнопку «Фініш», що із завершенням моделювання змінює назву на «Старт».

При роботі паралельного арифметичного зсувача на послідовностях вхідних наборів при їх повному переборі або на випадкових наборах результати моделювання виводяться у вигляді таблиці.

Таблиця результатів моделювання паралельного арифметичного зсувача містить стовпці за кількістю структур паралельного арифметичного зсувача та рядки, що показують наступне:

- кількість точок схем, де вводяться несправності;
- загальний відсоток помилок;
- відсотки суттєвих та несуттєвих помилок;
- обчислені значення достовірності результатів.

Мантиса операнда та мантиса результату зсуву має розрядність $n = 2^r$, де r – довжина величини зсуву. Мантиси є нормалізованими, тобто починаються зі значущої двійкової цифри, подані у додатковому коді та показуються $(n – 1)$– розрядним двійковим кодом без знакового розряду, що є інверсним до значення старшого розряду мантиси.

5.1.2.3 Обробка екстрених ситуацій.

Передбачено виведення повідомлення про помилки у разі порушення діапазону довжини адреси паралельного арифметичного зсувача нормалізованих мантис.

5.2 Результати моделювання

5.2.1 За результатами моделювання паралельного арифметичного зсувача мантис визначаються наступні дані:

K_i – кількість точок схем, де вводяться несправності для i-ї структури паралельного арифметичного зсувача мантис;

$P_{П\ i}$ – ймовірність появи помилки в результаті зсуву для i-ї структури паралельного арифметичного зсувача мантис;

$P_{ПС\ i}$ – ймовірність появи суттєвої помилки в результаті зсуву для *i*-ї структури паралельного арифметичного зсувача мантис;

$P_{ПН\ i}$ – ймовірність появи несуттєвої помилки в результаті зсуву для *i*-ї структури паралельного арифметичного зсувача мантис;

$ДР_i$ – достовірність результатів зсуву для *i*-ї структури паралельного арифметичного зсувача мантис;

Достовірність результатів, що обчислюються паралельним арифметичним зсувачем мантис у додатковому коді під дією його характерних несправностей, визначається, згідно (1.1), по формулі

$$ДР_i = 1 – P_{ПС\,i}.$$

5.2.2 Моделювання паралельного арифметичного зсувача мантис у додатковому коді проводилося при випадковому виборі несправної точки схеми до одержання стійких результатів, тобто результатів, що практично вже не мінялися у часі.

Результати моделювання паралельного арифметичного зсувача мантис у додатковому коді наведені в табл. 5.1 для випадку довжини величини зсуву $r = 4$, розрядності мантиси операнда $n = 16$ та коли всі n розрядів результату зсуву є вірними, тобто $n_T = n$.

Таблиця 5.1 – Ймовірності K_i, $P_{П\,i}$, $P_{ПС\,i}$, $P_{ПН\,i}$ та $ДР_i$ для $r = 4$ та $n_T = n$

Номери структур *N*	1	2	3	4
K_i, %	615	675	945	1695
$P_{П\,i}$, %	21,9	22,5	18,5	16,5
$P_{ПС\,i}$, %	21,9	22,5	18,5	16,5
$P_{ПН\,i}$, %	0	0	0	0
$ДР_i$, %	78	77,4	81,4	83,4

У табл. 5.2 наведені результати моделювання паралельного арифметичного зсувача мантис в додатковому коді для випадку $r = 4$, $n = 16$ та $n_T = n – 2$.

Таблиця 5.2 – Ймовірності K_i, $P_{П\,i}$, $P_{ПС\,i}$, $P_{ПН\,i}$ та $ДР_i$ для $r = 4$ та $n_T = n - 2$

Номери структур N	1	2	3	4
K_i, %	615	675	945	1695
$P_{П\,i}$, %	22,7	19,4	18,5	17,3
$P_{ПС\,i}$, %	18,9	17,1	16,5	15,4
$P_{ПН\,i}$, %	3,7	2,2	1,9	1,8
$ДР_i$, %	81	82,8	83,4	84,5

У табл. 5.3 наведені результати моделювання паралельного арифметичного зсувача мантис в додатковому коді для випадку $r = 4$, $n = 16$ та $n_T = n - 4$.

Таблиця 5.3 – Ймовірності K_i, $P_{П\,i}$, $P_{ПС\,i}$, $P_{ПН\,i}$ та $ДР_i$ для $r = 4$ та $n_T = n - 4$

Номери структур N	1	2	3	4
K_i, %	615	675	945	1695
$P_{П\,i}$, %	18,1	10,4	13,7	13,7
$P_{ПС\,i}$, %	13,7	9,2	12	11,6
$P_{ПН\,i}$, %	4,4	1,2	1,6	2
$ДР_i$, %	86,2	90,7	87,9	88,3

У табл. 5.4 наведені результати моделювання паралельного арифметичного зсувача мантис в додатковому коді для випадку $r = 4$, $n = 16$ та $n_T = n - 6$.

Таблиця 5.4 – Ймовірності K_i, $P_{П\,i}$, $P_{ПС\,i}$, $P_{ПН\,i}$ та $ДР_i$ для $r = 4$ та $n_T = n - 6$

Номери структур N	1	2	3	4
K_i, %	615	675	945	1695
$P_{П\,i}$, %	20,8	21	18,5	17,2
$P_{ПС\,i}$, %	9,4	12	9,3	11,1
$P_{ПН\,i}$, %	11,3	9	9,2	6,1
$ДР_i$, %	90,5	87,9	90,6	88,8

У табл. 5.5 наведені результати моделювання паралельного арифметичного зсувача мантис в додатковому коді для випадку $r = 4$, $n = 16$ та $n_T = n - 8$.

Таблиця 5.5 – Ймовірності K_i, $P_{П\,i}$, $P_{ПС\,i}$, $P_{ПН\,i}$ та $ДР_i$ для $r = 4$ та $n_T = n - 8$

Номери структур N	1	2	3	4
K_i, %	615	675	945	1695
$P_{П\,i}$, %	21,4	20,6	17,9	16,8
$P_{ПС\,i}$, %	10	9,7	8,7	10
$P_{ПН\,i}$, %	11,4	10,9	9,2	6,8
$ДР_i$, %	89,9	90,2	91,2	89,9

У табл. 5.6 наведені результати моделювання паралельного арифметичного зсувача мантис для випадку $r = 5$, $n = 32$ та $n_T = n$.

Таблиця 5.6 – Ймовірності K_i, $P_{П\,i}$, $P_{ПС\,i}$, $P_{ПН\,i}$ та $ДР_i$ для $r = 5$ та $n_T = n$

Номери структур N	1	2	3	4	5
K_i, %	1581	1705	2263	3813	7595
$P_{П\,i}$, %	20	22,7	18,9	16,4	13,9
$P_{ПС\,i}$, %	20	22,7	18,9	16,4	13,9
$P_{ПН\,i}$, %	0	0	0	0	0
$ДР_i$, %	79,9	77,2	81	83,5	86

У табл. 5.7 наведені результати моделювання паралельного арифметичного зсувача мантис в додатковому коді для випадку $r = 5$, $n = 32$ та $n_T = n - 2$.

Таблиця 5.7 – Ймовірності K_i, $P_{П\,i}$, $P_{ПС\,i}$, $P_{ПН\,i}$ та $ДР_i$ для $r = 5$ та $n_T = n - 2$

Номери структур N	1	2	3	4	5
K_i, %	1581	1705	2263	3813	7595
$P_{П\,i}$, %	22,1	19,1	18,5	13,9	14,5
$P_{ПС\,i}$, %	20,5	17,6	17,2	13	13,1
$P_{ПН\,i}$, %	1,5	1,5	1,3	0,9	1,3
$ДР_i$, %	79,4	82,3	82,7	86,9	86,8

У табл. 5.8 наведені результати моделювання паралельного арифметичного зсувача мантис в додатковому коді для випадку $r = 5$, $n = 32$ та $n_T = n - 4$.

Таблиця 5.8 – Ймовірності K_i, $P_{П\,i}$, $P_{ПС\,i}$, $P_{ПН\,i}$ та $ДР_i$ для $r = 5$ та $n_T = n - 4$

Номери структур N	1	2	3	4	5
K_i, %	1581	1705	2263	3813	7595
$P_{П\,i}$, %	23,8	20,3	17	15,9	16,5
$P_{ПС\,i}$, %	19,3	17,2	14	13,1	14,3
$P_{ПН\,i}$, %	4,4	3,1	3	2,7	2,1
$ДР_i$, %	80,6	82,3	82,7	86,9	86,8

У табл. 5.9 наведені результати моделювання паралельного арифметичного зсувача мантис в додатковому коді для випадку $r = 5$, $n = 32$ та $n_T = n - 6$.

Таблиця 5.9 – Ймовірності K_i, $P_{П\,i}$, $P_{ПС\,i}$, $P_{ПН\,i}$ та $ДР_i$ для $r = 5$ та $n_T = n - 6$

Номери структур N	1	2	3	4	5
K_i, %	1581	1705	2263	3813	7595
$P_{П\,i}$, %	19,6	19,2	17,3	14,1	13,1
$P_{ПС\,i}$, %	14,2	14,3	13,3	10,3	10,3
$P_{ПН\,i}$, %	5,3	4,8	4	3,7	2,7
$ДР_i$, %	85,7	85,6	86,6	89,6	89,6

У табл. 5.10 наведені результати моделювання паралельного арифметичного зсувача мантис в додатковому коді для випадку $r = 5$, $n = 32$ та $n_T = n - 8$.

Таблиця 5.10 – Ймовірності K_i, $P_{П\,i}$, $P_{ПС\,i}$, $P_{ПН\,i}$ та $ДР_i$ для $r = 5$ та $n_T = n - 8$

Номери структур N	1	2	3	4	5
K_i, %	1581	1705	2263	3813	7595
$P_{П\,i}$, %	20,6	20,5	18,2	15,3	14,3
$P_{ПС\,i}$, %	13,9	13,4	11,8	10,8	11
$P_{ПН\,i}$, %	6,7	7	6,3	4,4	3,2
$ДР_i$, %	86	86,5	88,1	89,1	88,9

5.2.3 Для обробки результатів моделювання, що зібрані у таблицях, будуються діаграми залежності ймовірностей від рівня схемного паралелізма зсувача мантис.

На рис. 5.1 показані діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n паралельного арифметичного зсувача мантис для $r = 4$ та кількості вірних розрядів $n_T = n$.

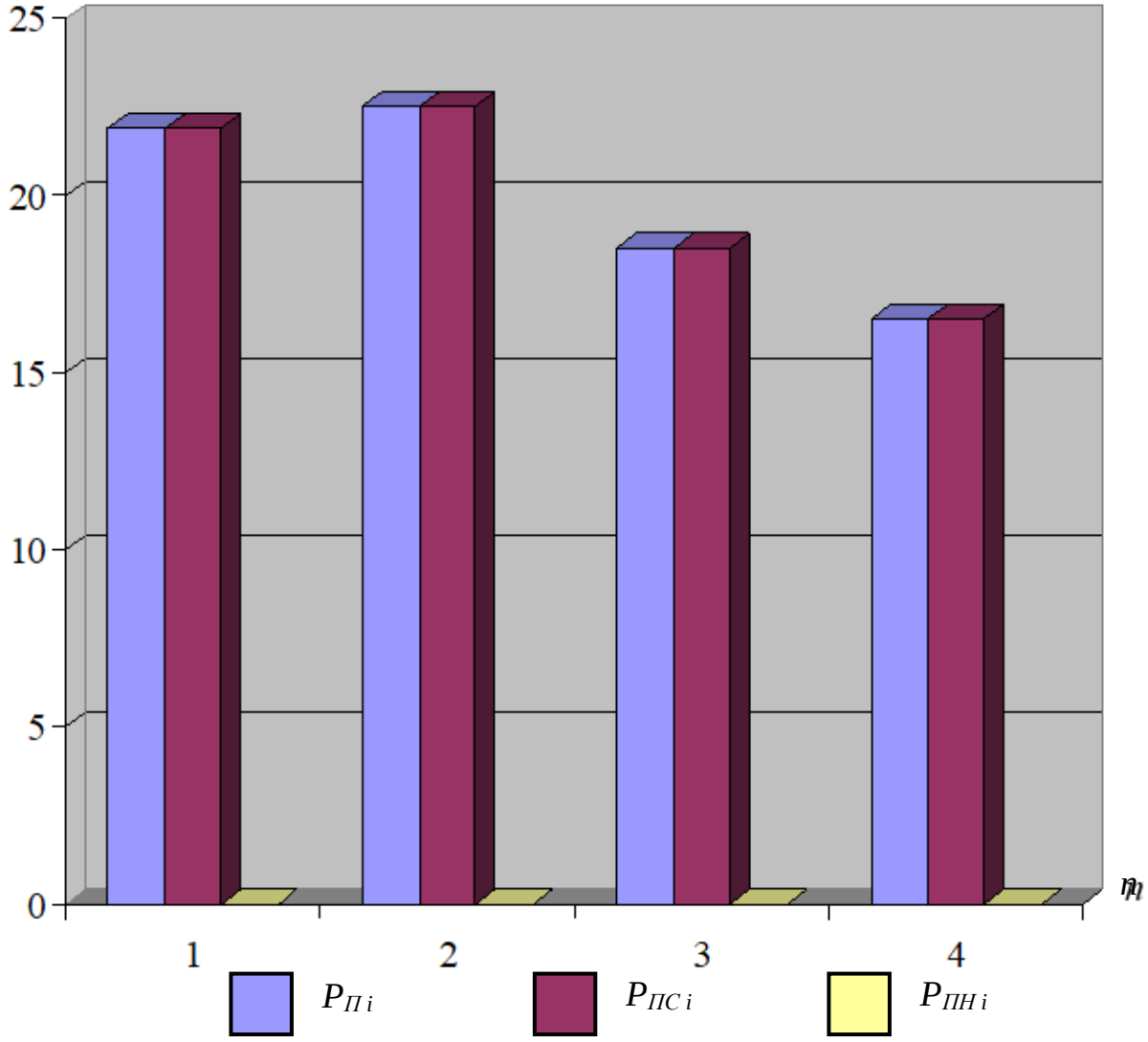

Рис. 5.1 – Діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n для $r = 4$ та $n_T = n$.

Побудовані діаграми показують, що ймовірність появи суттєвої помилки знижується із зростанням схемного паралелізму.

Несуттєві помилки відсутні, бо всі розряди результату вважаються вірними.

На рис. 5.2 показані діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n паралельного арифметичного зсувача мантис для $r = 4$ та кількості вірних розрядів $n_T = n - 2$.

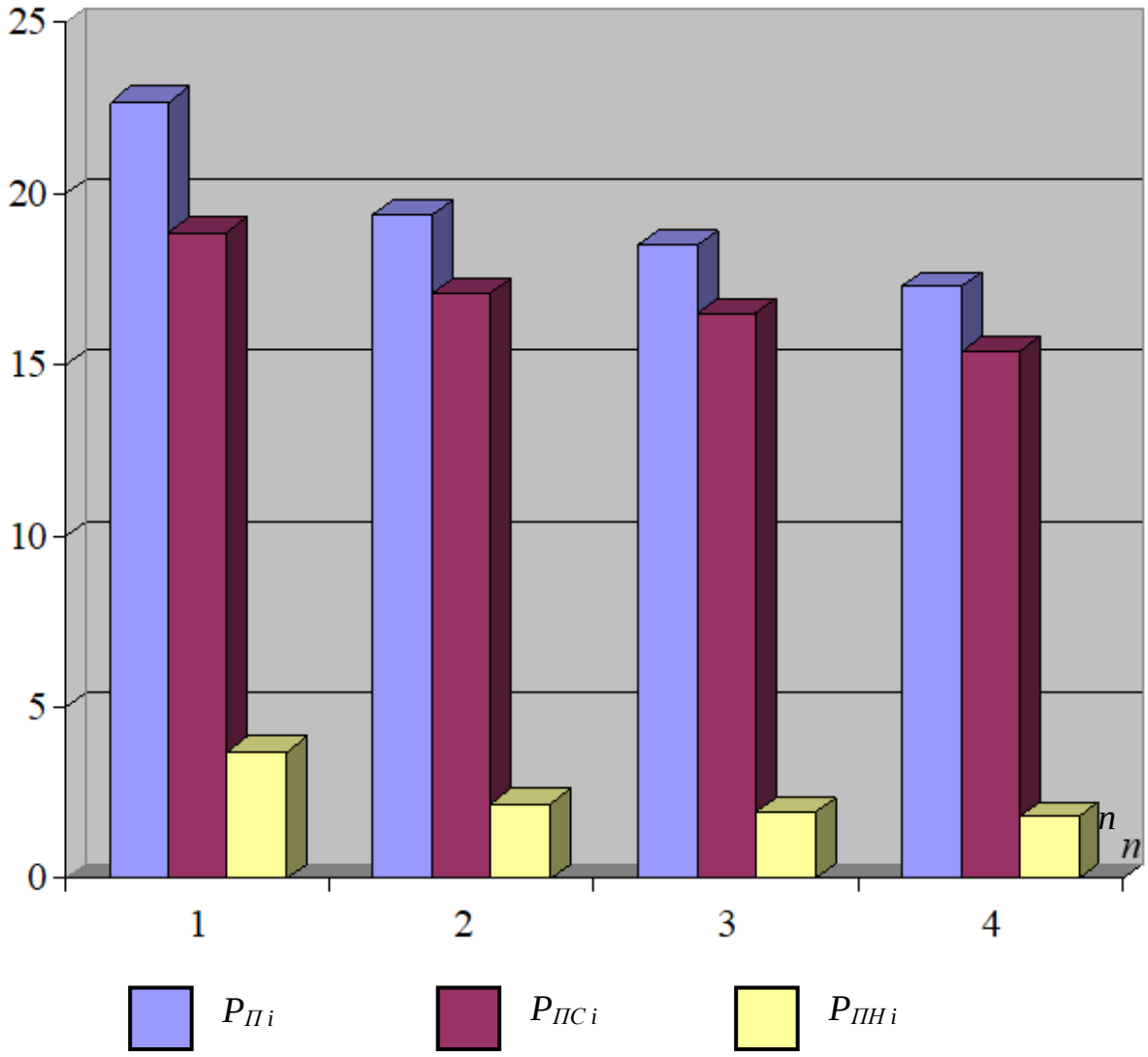

Рис. 5.2 – Діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n для $r = 4$ та $n_T = n - 2$.

Побудовані діаграми показують, що ймовірність появи суттєвої помилки знижується із зростанням схемного паралелізму.

Ймовірність несуттєвої помилки має значно менші розміри і також знижується із зростанням схемного паралелізму.

На рис. 5.3 показані діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n паралельного арифметичного зсувача мантис для $r = 4$ та кількості вірних розрядів $n_T = n - 4$.

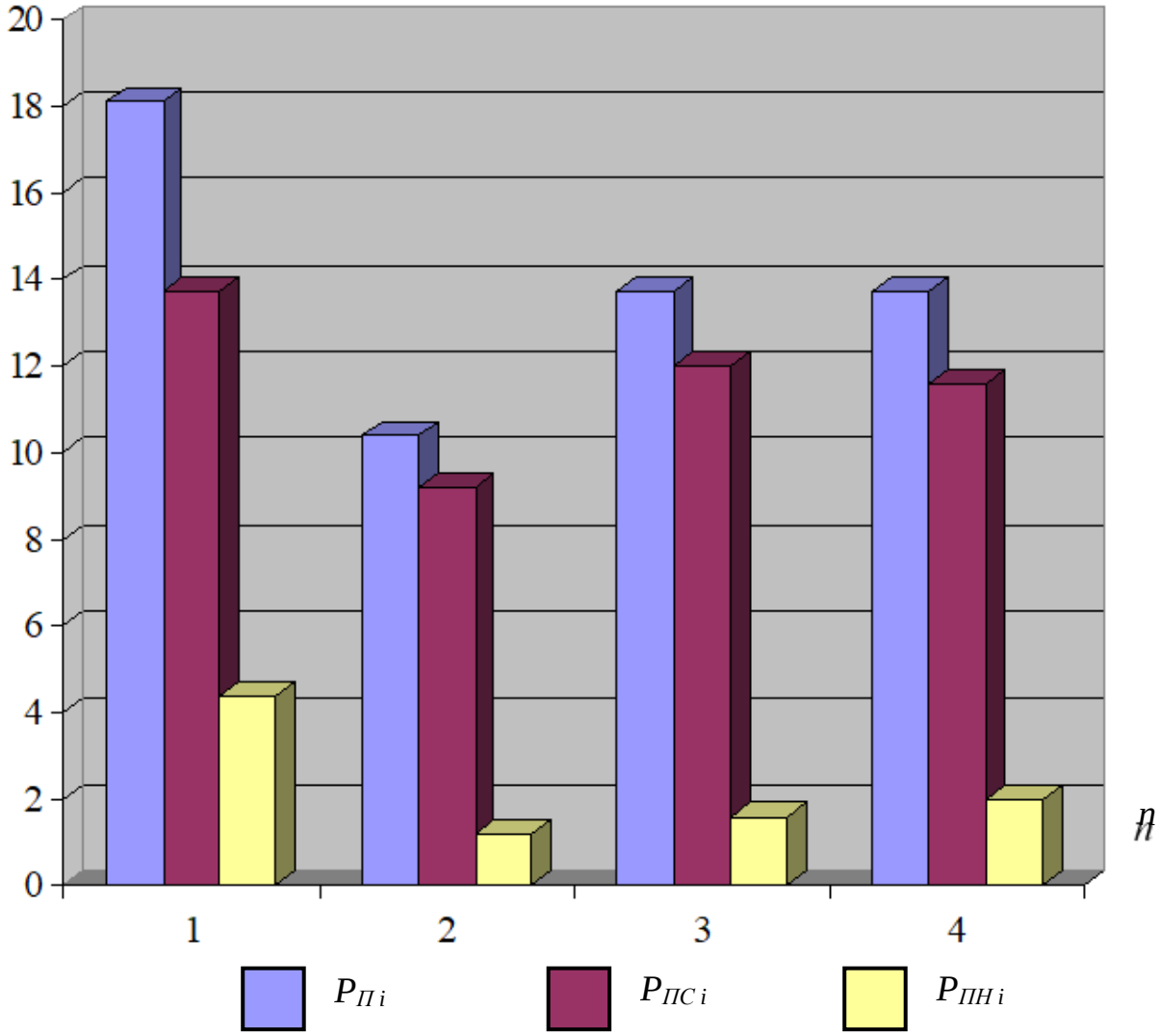

Рис. 5.3 – Діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах nдля $r = 4$ та $n_T = n - 4$.

Побудовані діаграми показують, що ймовірність появи суттєвої помилки знижується із зростанням схемного паралелізму.

Ймовірність несуттєвої помилки має значно менші розміри і також знижується із зростанням схемного паралелізму.

На рис. 5.4 показані діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n паралельного арифметичного зсувача мантис для $r = 4$ та кількості вірних розрядів $n_T = n - 6$.

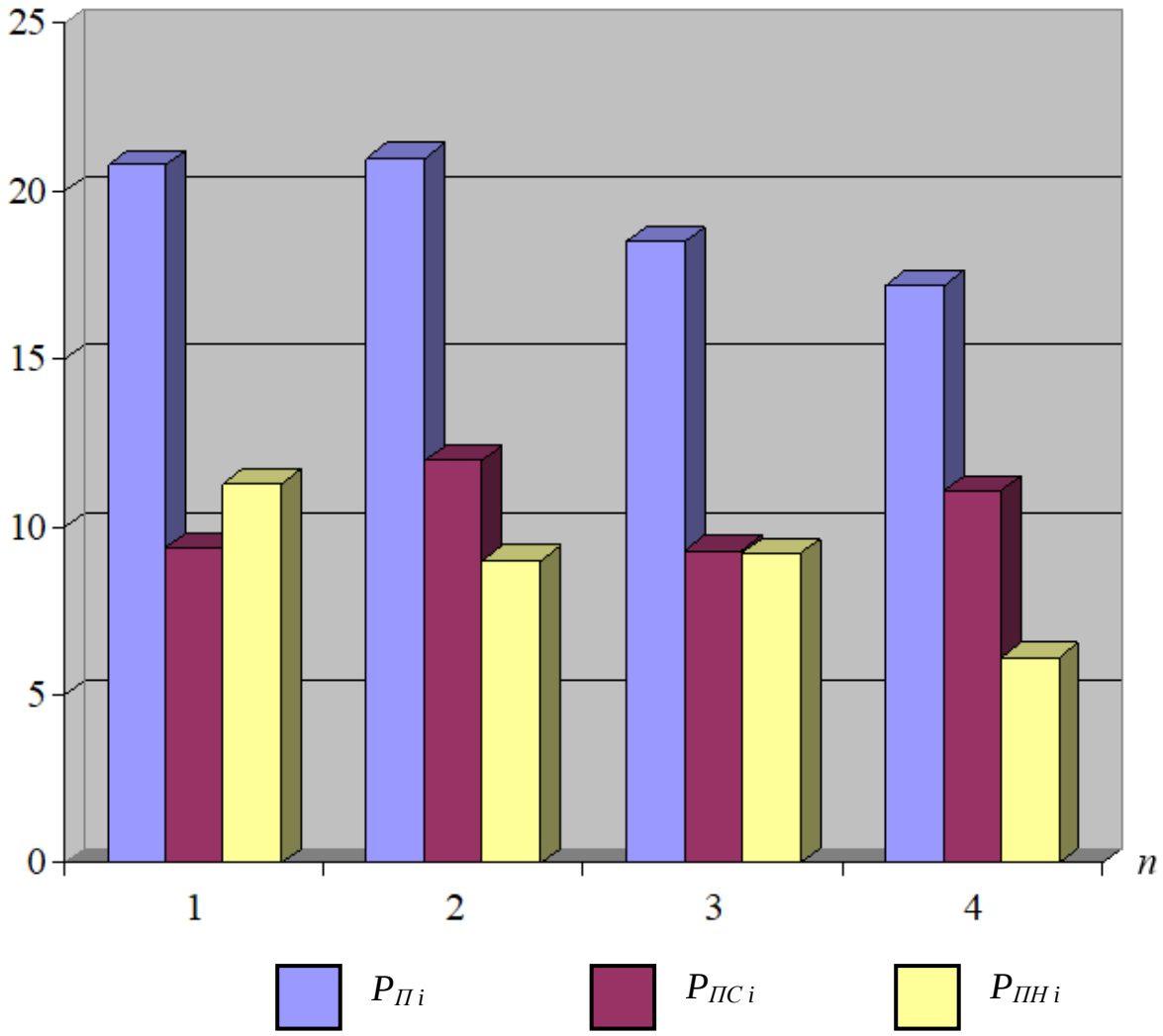

Рис. 5.4 – Діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n для $r = 4$ та $n_T = n - 6$.

Побудовані діаграми показують, що ймовірність появи суттєвої помилки знижується із зростанням схемного паралелізму.

Ймовірність несуттєвої помилки має майже такі ж розміри і також знижується із зростанням схемного паралелізму.

На рис. 5.5 показані діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n паралельного арифметичного зсувача мантис для $r = 4$ та кількості вірних розрядів $n_T = n - 8$.

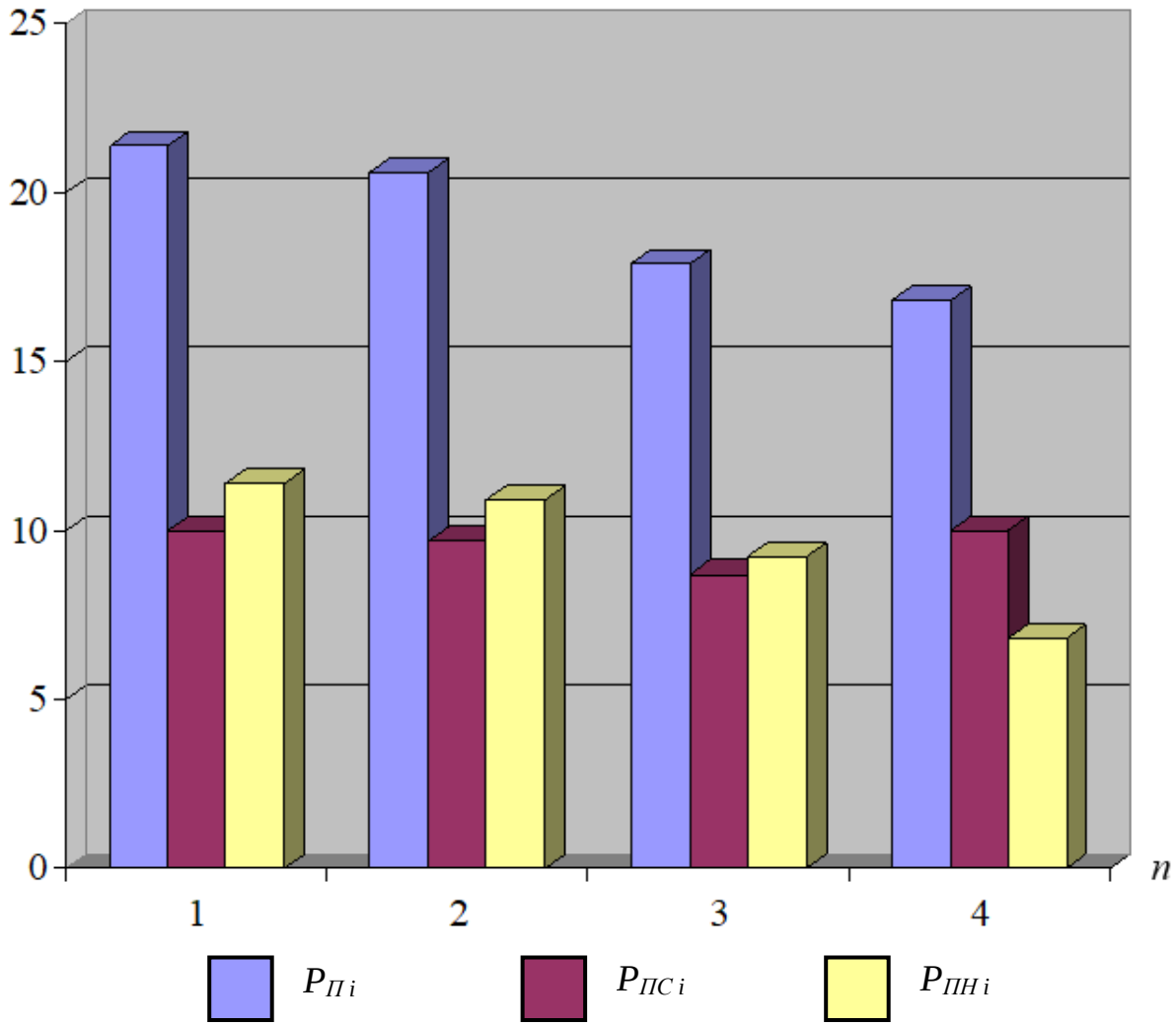

Рис. 5.5 – Діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n для $r = 4$ та $n_T = n - 8$.

Побудовані діаграми показують, що ймовірність появи суттєвої та несуттєвої помилки знижується із зростанням схемного паралелізму.

На рис. 5.6 показані діаграми достовірності результатів зсуву в різних структурах n паралельного арифметичного зсувача мантис для $r = 4$ і різної кількості вірних розрядів n_T.

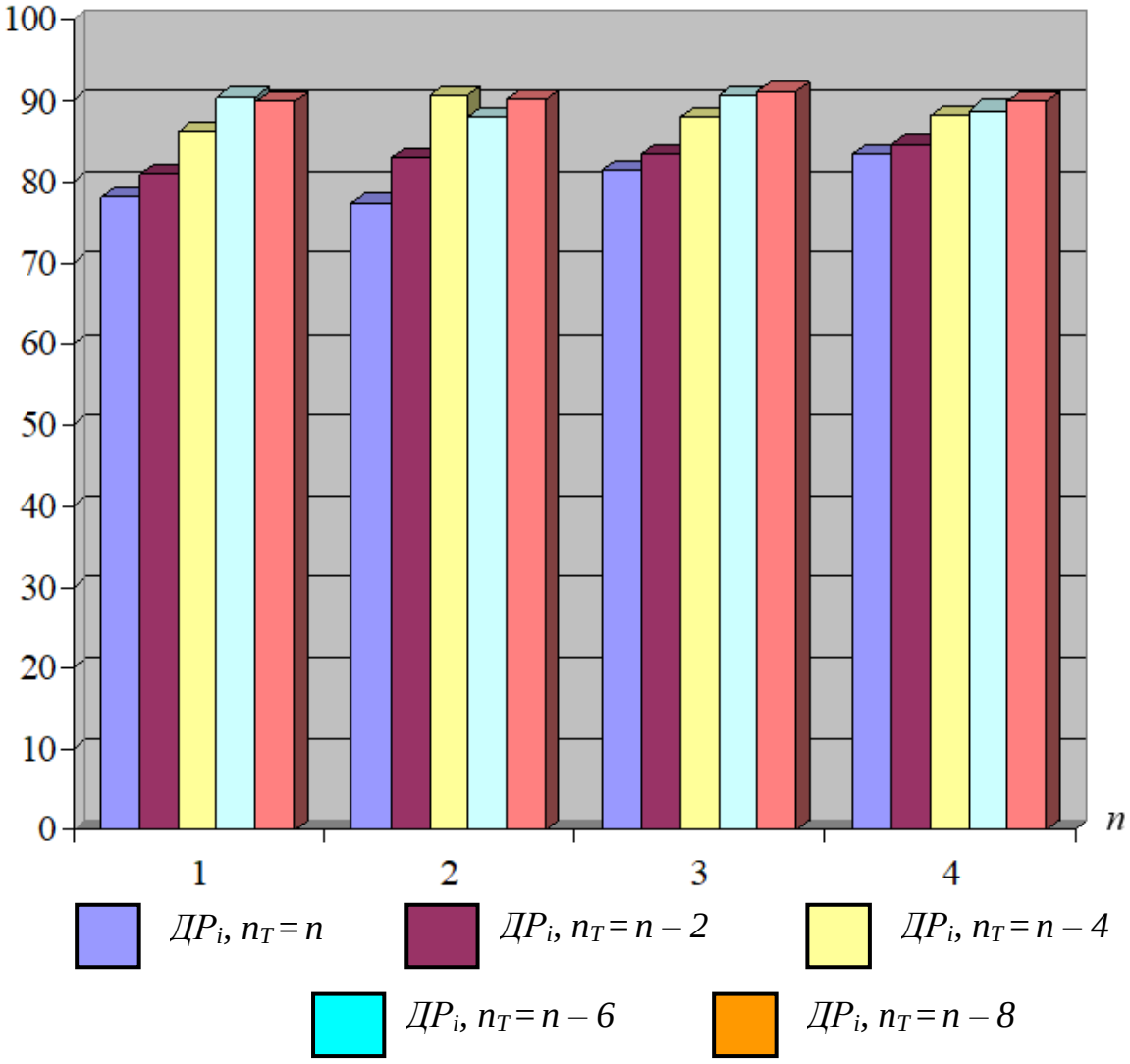

Рис. 5.6 – Діаграми достовірності результатів зсуву в різних структурах n для $r = 4$ та різної кількості вірних розрядів n_T.

Побудовані діаграми показують, що достовірність результатів зсуву мають тенденцію підвищення із зростанням схемного паралелізму та зниженням кількості необхідних вірних розрядів.

На рис. 5.7 показані діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n паралельного арифметичного зсувача мантис для $r = 5$ та кількості вірних розрядів $n_T = n$.

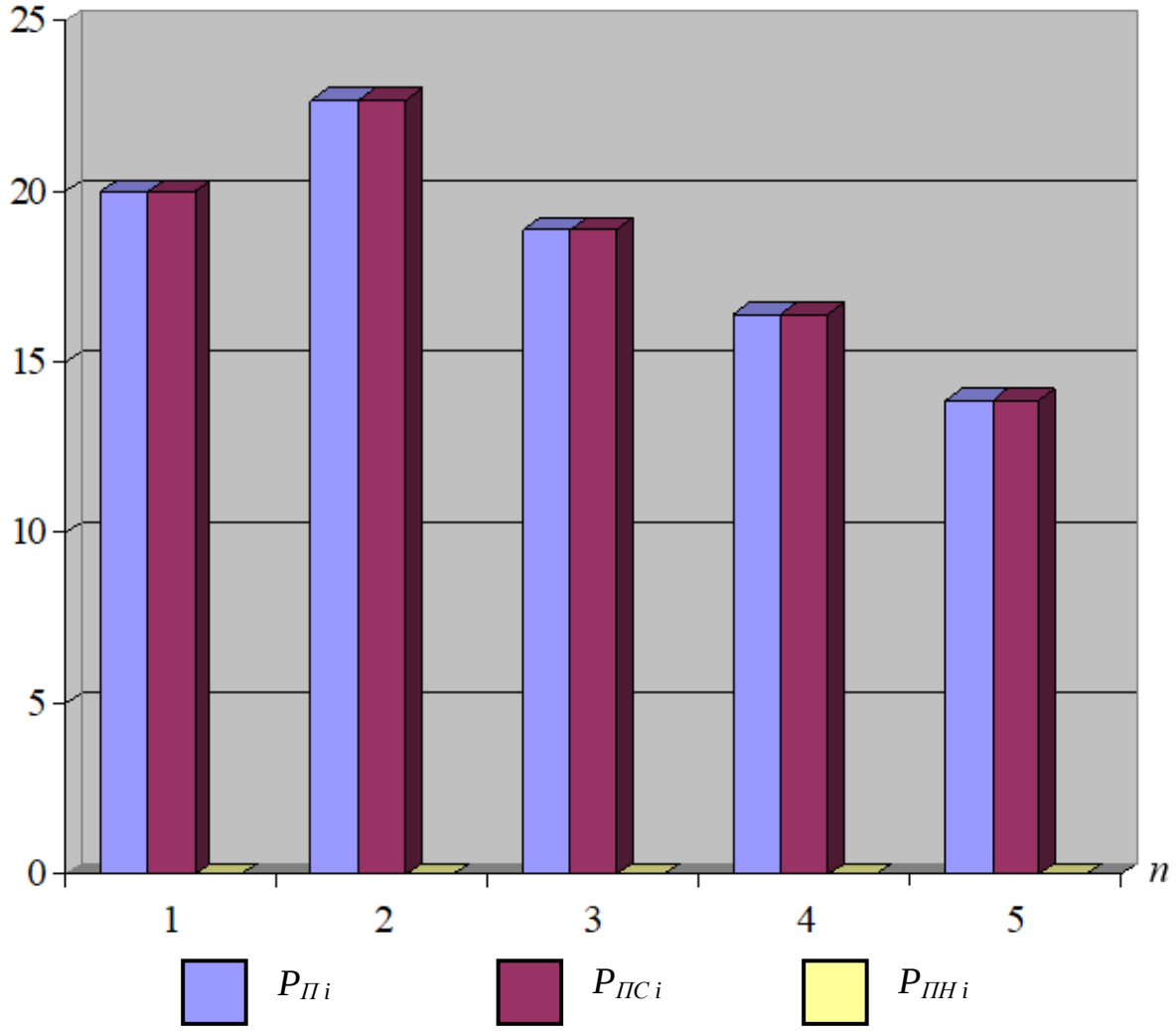

Рис. 5.7 – Діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n для $r = 5$ та $n_T = n$.

Побудовані діаграми показують, що ймовірність появи суттєвої помилки знижується із зростанням схемного паралелізму. Несуттєві помилки відсутні, бо всі розряди результату вважаються вірними..

На рис. 5.8 показані діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n паралельного арифметичного зсувача мантис для $r = 5$ та кількості вірних розрядів $n_T = n - 2$.

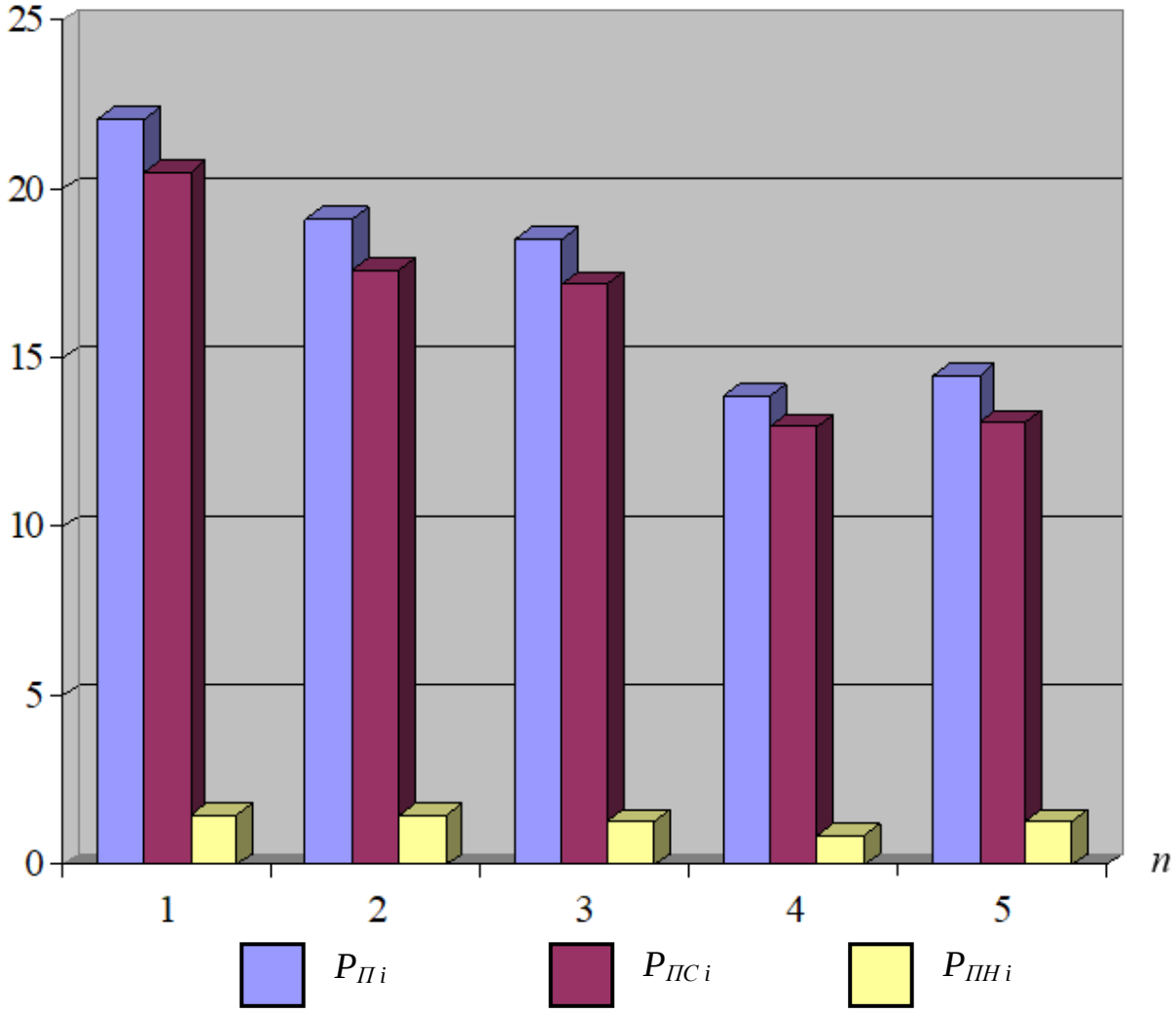

Рис. 5.8 – Діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n для $r = 5$ та $n_T = n - 2$.

Побудовані діаграми показують, що ймовірність появи суттєвої помилки знижується із зростанням схемного паралелізму.

Ймовірність несуттєвої помилки має значно менші розміри і також знижується із зростанням схемного паралелізму.

На рис. 5.9 показані діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n паралельного арифметичного зсувача нормалізованих мантис для $r = 5$ та кількості вірних розрядів $n_T = n - 4$.

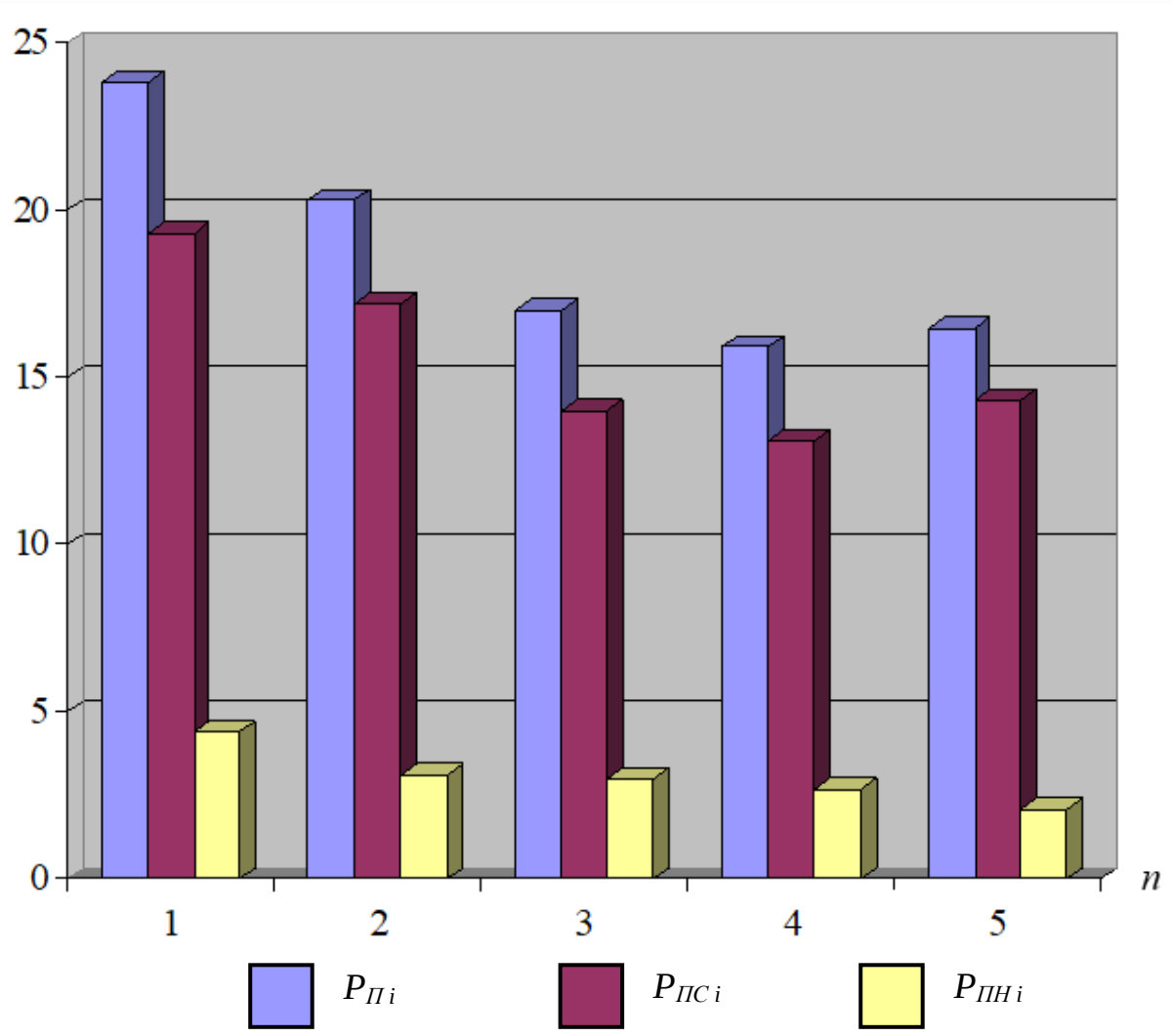

Рис. 5.9 – Діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n для $r = 5$ та $n_T = n - 4$.

Побудовані діаграми показують, що ймовірність появи суттєвої помилки знижується із зростанням схемного паралелізму.

Ймовірність несуттєвої помилки має значно менші розміри і також знижується із зростанням схемного паралелізму.

На рис. 5.10 показані діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n паралельного арифметичного зсувача нормалізованих мантис для $r = 5$ та кількості вірних розрядів $n_T = n - 6$.

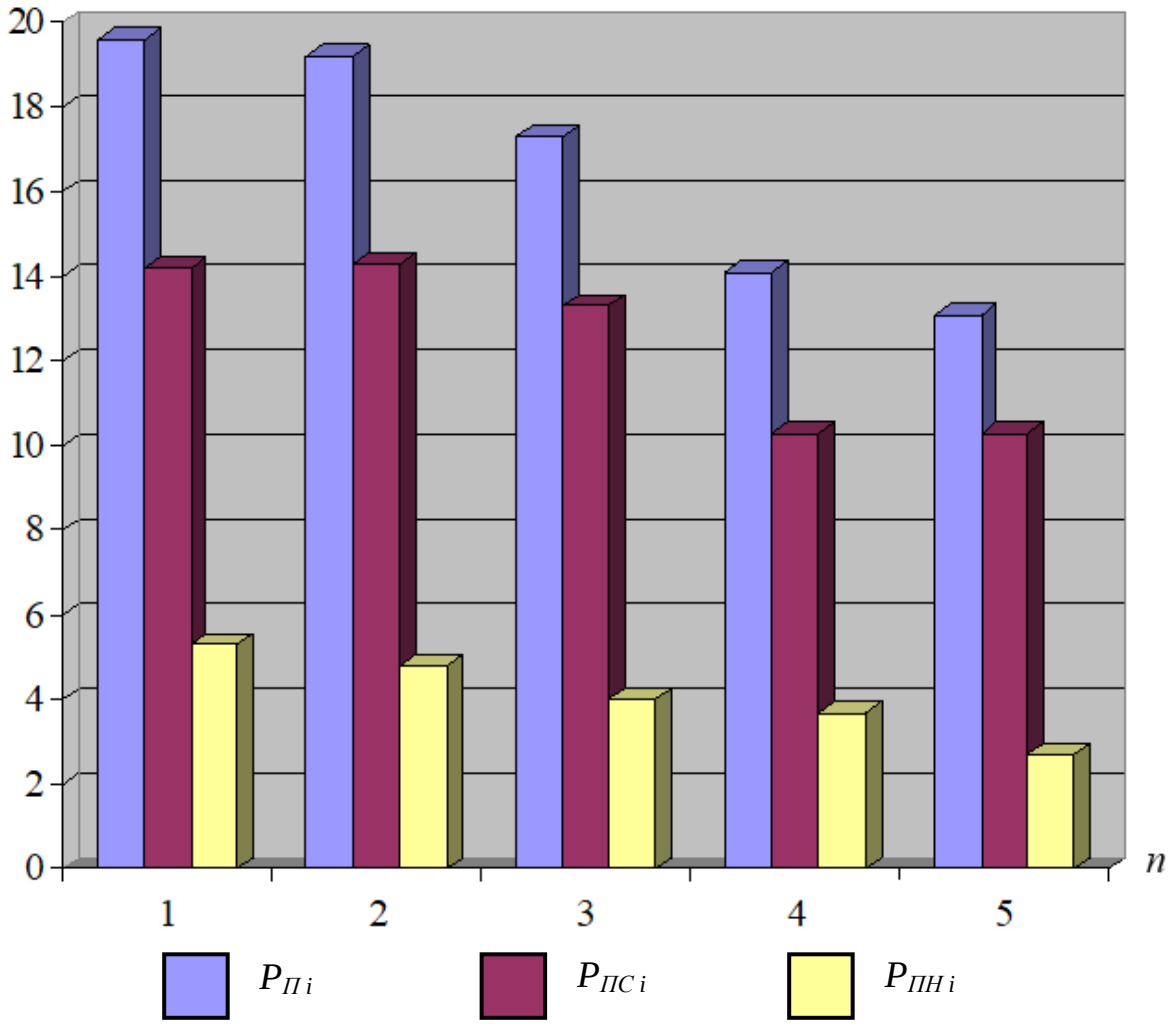

Рис. 5.10 – Діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах nдля $r = 5$ та $n_T = n - 6$.

Побудовані діаграми показують, що ймовірність появи суттєвої помилки знижується із зростанням схемного паралелізму.

Ймовірність несуттєвої помилки має значно менші розміри і також знижується із зростанням схемного паралелізму.

На рис. 5.11 показані діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n паралельного арифметичного зсувача мантис для $r = 5$ та кількості вірних розрядів $n_T = n - 8$.

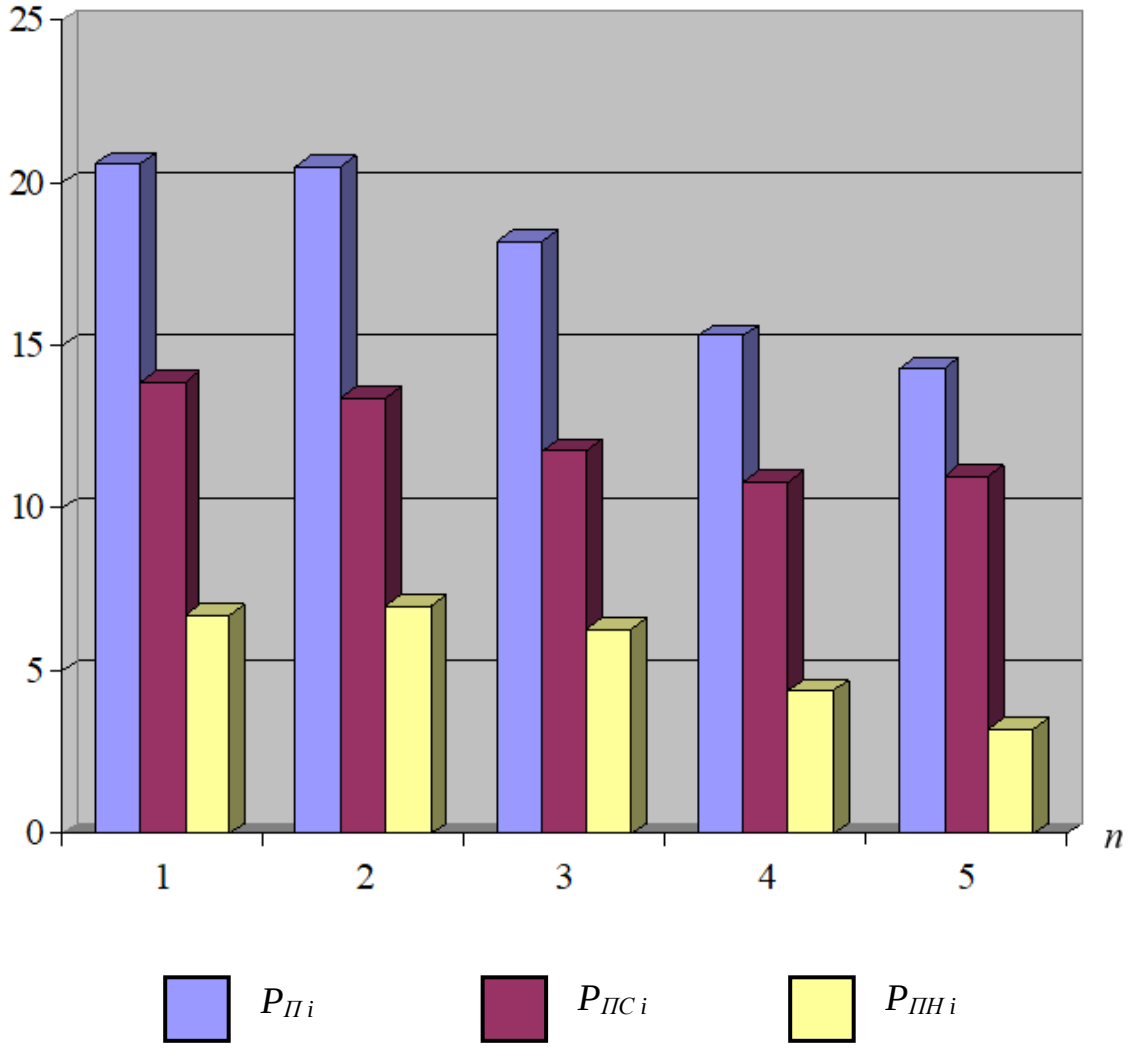

Рис. 5.11 – Діаграми ймовірностей появи помилки $P_{П\,i}$, суттєвої $P_{ПС\,i}$ та несуттєвої $P_{ПН\,i}$ помилки в різних структурах n для $r = 5$ та $n_T = n - 8$.

Побудовані діаграми показують, що ймовірність появи суттєвої помилки знижується із зростанням схемного паралелізму.

Ймовірність несуттєвої помилки має значно менші розміри і також знижується із зростанням схемного паралелізму.

На рис. 5.12 показані діаграми достовірності результатів зсуву в різних структурах n паралельного арифметичного зсувача мантис для $r = 5$ і різної кількості вірних розрядів n_T.

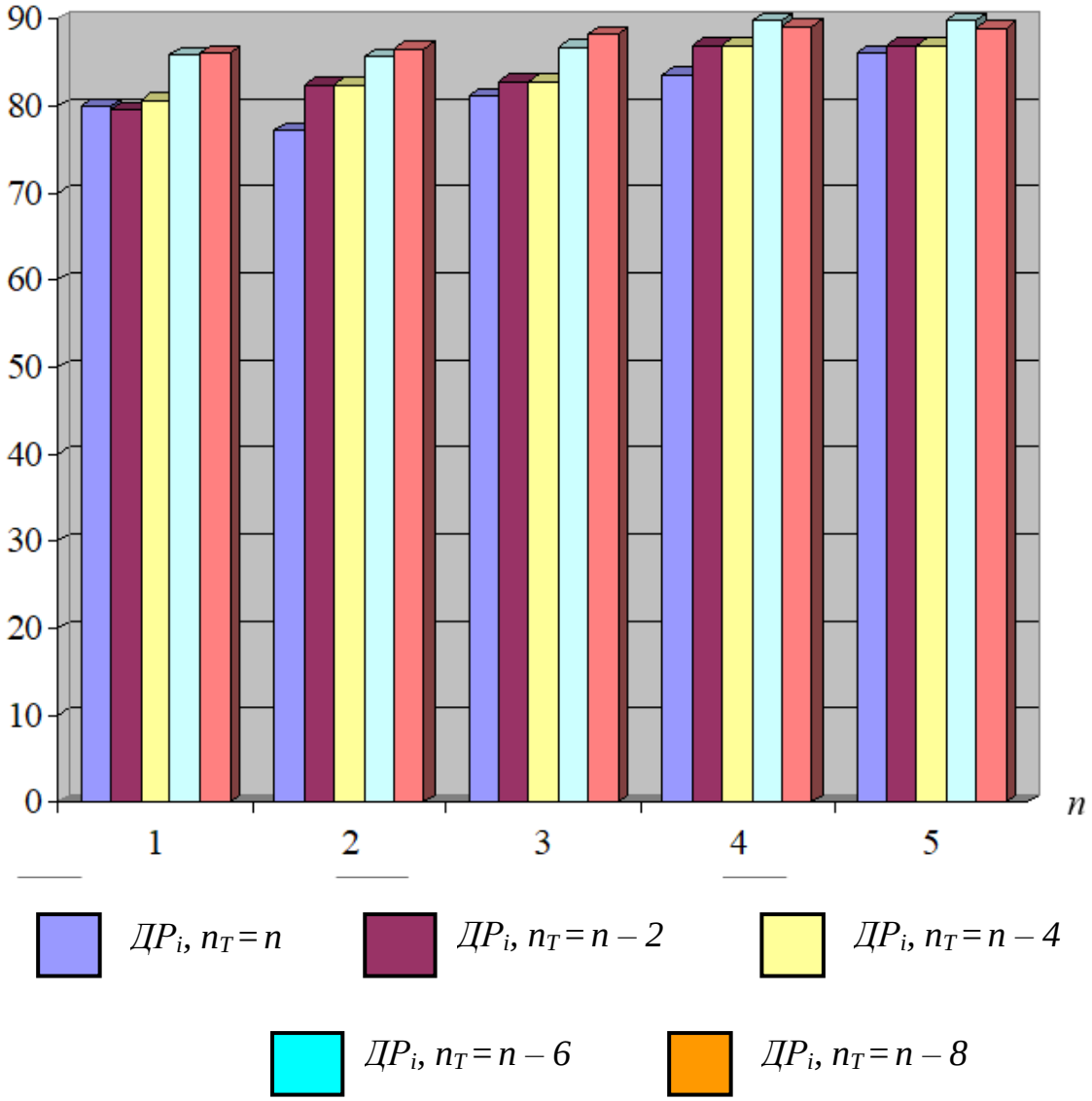

Рис. 5.12 – Діаграми достовірності результатів зсуву в різних структурах n для $r = 5$ та різної кількості вірних розрядів n_T.

Побудовані діаграми показують, що достовірність результатів зсуву, що обчислюються під впливом несправностей в паралельному арифметичному зсувачі нормалізованих двійкових мантис мають ще більш виражену тенденцію підвищення із зростанням схемного паралелізму пристрою та зниженням кількості необхідних вірних розрядів результату.

5.3 Висновки

Програмна модель паралельного арифметичного зсувача нормалізованих мантис написана у середовищі Delphi, використовує одну форму та складається з 29 процедур.

За вихідні дані задається розрядність мантис, а інші дані визначаються в процесі роботи програмної моделі. Результати моделювання паралельного арифметичного зсувача нормалізованих мантис зберігаються на екрані.

На панелі моделювання паралельного арифметичного зсувача мантис розташовано демонстраційні елементи та елементи керування.

Проведені дослідження однотактного паралельного арифметичного зсувача мантис виконувалися на програмній моделі, що створювала різні рівні схемного паралелізму пристрою, а також дозволяла встановлювати розподіл розрядів результату зсуву на вірні та невірні.

Результати моделювання показали, що ймовірність появи суттєвої помилки знижується із зростанням схемного паралелізму однотактного арифметичного зсувача мантис.

Ймовірність несуттєвої помилки підвищується зі збільшенням кількості невірних розрядів та знижується із зростанням схемного паралелізму арифметичного зсувача мантис.

Достовірність результатів арифметичного зсуву нормалізованих мантис має тенденцію підвищення зі зростанням схемного паралелізму однотактного пристрою та зі зниженням встановленої кількості необхідних вірних розрядів результату.

ВИСНОВОК

Огляд рішень з підвищення достовірності результатів, які обчислюються у сучасних арифметичних пристроях, показав, що ця проблема розв’язується разом з підвищенням продуктивності комп’ютерних систем та їх компонентів. Ці напрямки є основними тенденціями розвитку обчислювальної техніки і потребують значних витрат ресурсів.

В цих умовах стає актуальним дослідження залежності достовірності обчислювальних результатів від рівня розпаралелення схемних рішень, що відбувається для підвищення продуктивності комп’ютерних систем.

Такі дослідження доцільно провести на прикладі паралельного арифметичного зсувача мантис, що може бути побудований з різним рівнем схемного паралелізму. Тому за об’єкт досліджень вибрано обчислювальний процес, що відбувається в паралельному арифметичному зсувачі мантис, а за предмет досліджень – достовірність результатів, що обчислюються паралельним арифметичним зсувачем мантис в умовах дії типових несправностей на різних рівнях розпаралелювання схеми.

В роботі була поставлена мета дослідження паралельного арифметичного зсувача мантис для оцінки достовірності наближених результатів, що обчислюються в умовах дії типових несправностей,

Для проведення дослідження були розроблені алгоритми та побудована за ними програмна модель паралельного арифметичного зсувача нормалізованих мантис, що подаються в додатковому коді.

Програмна модель відповідає вимогам наочності, що досягнуто мовними засобами середовища Delphi.

Крім того, програмна модель відповідає вимогам функціональності, реалізуючи цілу низку основних та допоміжних режимів.

Програмна модель відповідає також вимогам достовірності, оскільки була перевірена на багатьох контрольних прикладах.

Результати моделювання були одержані на довільних послідовностях вхідних слів, що складаються з мантиси операнда та величини зсуву. Встановлювалося довільне місце одиночної несправності серед контрольних точок схем паралельного арифметичного зсувача мантис, а також довільне значення константної несправності.

Моделювання паралельного арифметичного зсувача мантис виконувалося одночасно для декількох схем зсувача, що мають різні рівні схемного паралелізму.

Дослідження проводилися для різної розрядності паралельного арифметичного зсувача мантис (з 8-ми до 15-ти), а також різного розподілу результатів на вірні та невірні розряди.

За моделюванням паралельного арифметичного зсувача мантис визначалися ймовірності появи суттєвих та несуттєвих помилок, а також достовірність результатів зсуву.

Результати моделювання однотактного арифметичного зсувача мантис показали, що ймовірність появи суттєвої помилки знижується із зростанням схемного паралелізму пристрою.

Ймовірність несуттєвої помилки підвищується зі збільшенням кількості невірних розрядів та знижується із зростанням схемного паралелізму арифметичного зсувача мантис.

Основний висновок визначає, що достовірність результатів арифметичного зсуву нормалізованих мантис має тенденцію підвищення зі зростанням схемного паралелізму пристрою та зі зниженням встановленої кількості необхідних вірних розрядів результату.

ПЕРЕЛІК ЛІТЕРАТУРИ

1. Вычислительные системы для решения задач оперативно-организационного управления / А. Г. Додонов, В. В. Хаджинов, И. И. Волосков. – Киев: Наукова думка, 1988. – 216 с.

2. Локазюк В.М., Савченко Ю.Г. Надійність, контроль, діагностика і модернізація ПК: Посібник - К.: Видавничий центр «Академія», 2004. - 376.

3. Проектирование и тестирование цифровых систем на кристаллах / В.И. Хаханов, Е.И. Литвинова, О.А. Гузь.– Харьков: ХНУРЭ. – 2009. – 484 с.

4. Проектирование и верификация цифровых систем на кристалах.Verilog & System Verilog / В.И. Хаханов, И.В. Хаханова, Е.И. Литвинова, О.А. Гузь. – Харьков: ХНУРЭ. – 2010. – 528 с.

5. Хоровиц П., Хилл У. Искусство схемотехники. – М.: Мир, – 1993. – 704 с.

6. Таненбаум Э. Архитектура компьютера. – СПб: Питер, 2002. – 704 с.

7. СуперЭВМ. Аппаратная и программная организация. / Под ред. С. Фернбаха. – М.: Радио и связь, 1991. – 320 с.

8. Основи цифрових систем / І.П.Барбаш, М.П.Благодарний, В.Я. Жихарев та ін. – Підручник. – Харків: Нац. Аерокосмічний ун-т, 2002. – 672 с.

9. Самофалов К. Г., Романкевич А. М., Валуйский В. Н., Каневский Ю. С., Пиневич М. М. Прикладная теория цифровых автоматов. – Киев: Вища школа, 1987. – 375 с.

10. Арифметика, принципы организации, диагностика и формальное проектирование вычислительных структур и устройств / В. П. Тарасенко, Н. В. Черкасский, Ю. С. Каневский и др. – Киев: Вища школа, 1989. – 343 с.

11. Справочник по цифровой вычислительной технике. / Б. Н. Малиновский, В. Я. Александров, В. П. Боюн и др. Киев: Техника, 1974. – 512 с.

12. Харченко В.С., Жихарев В.Я., Илюшко В.М., Нечипорук Н.В. Многоверсионные системы, технологии. – Х.: Нац. аэрокосмический ун-т «Харьковский авиационный ин-т», 2003. – 486 с.

13. Дрозд А. В., Антощук С.Г., Русинский А., Колахи Реза Джавад. Повышение достоверности контроля результатов в обработке приближенных данных. // Радіоелектронні і комп'ютерні системи. – 2008. – № 7 (34). – С. 30 – 34.

14. Каган Б. М. Электронные вычислительные машины и системы: Учеб. Пособие для вузов. – М.: Энергоатомиздат, 1991. – 592 с.

15. Дрозд О.В., Боговик О.С, Горячко М.С., Довгань Я.О., Мовсесян А.М. Проблема достовірності результатів в арифметичних компонентах систем критичного застосування // Матеріали IX Міжнародної науково-практичної конференції «Інформаційні управляючі системи і технології» (ІУСТОДЕСА-2021). – Одеса, 2021. – С. 109 – 112ггґ.

ДОДАТОК А

ТЕКСТ ПРОГРАМНОЇ МОДЕЛІ

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Menus;

type
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    Image1: TImage;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    Button6: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Panel3: TPanel;
    Edit3: TEdit;
    Edit4: TEdit;
    Panel4: TPanel;
    Button7: TButton;
    MainMenu1: TMainMenu;
    A1: TMenuItem;
    A2: TMenuItem;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    B1: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    Edit9: TEdit;
    Panel12: TPanel;
    Panel21: TPanel;
    Image2: TImage;
    Panel22: TPanel;
    Panel34: TPanel;
    Button15: TButton;
    Panel39: TPanel;
    Panel13: TPanel;
    Panel14: TPanel;
    Panel15: TPanel;
```

```
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure OnMauseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button7Click(Sender: TObject);
    procedure A2Click(Sender: TObject);
    procedure N1Click(Sender: TObject);
    procedure N2Click(Sender: TObject);
    procedure N3Click(Sender: TObject);
    procedure N4Click(Sender: TObject);
    procedure N5Click(Sender: TObject);
    procedure N6Click(Sender: TObject);
    procedure N7Click(Sender: TObject);
    procedure N9Click(Sender: TObject);
    procedure N10Click(Sender: TObject);
    procedure Button12Click(Sender: TObject);
    procedure FormDragOver(Sender, Source: TObject; X, Y: Integer;
      State: TDragState; var Accept: Boolean);
    procedure FormDragDrop(Sender, Source: TObject; X, Y: Integer);
    procedure Panel22MouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure Panel21Click(Sender: TObject);
    procedure Panel12Click(Sender: TObject);
    procedure Button15Click(Sender: TObject);
    procedure Panel35Click(Sender: TObject);
    procedure Panel13Click(Sender: TObject);
    procedure Panel14Click(Sender: TObject);
    procedure Panel15Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

const
  nh=31; mh=5;
var
  Form1: TForm1;
  AAb:array[1..nh]of integer;
  AAc:array[1..nh]of integer;
  ASb:array[1..nh]of integer;
  BBb:array[1..mh]of integer;
  BBc:array[1..mh]of integer;
  BBd:array[1..mh]of integer;
  M1:array[1..mh,1..nh,1..2*mh]of integer;
  M2:array[1..mh,1..nh,1..nh+1,1..mh+2]of integer;
  M3:array[1..mh,1..mh,1..nh]of integer;
```

```
  M41:array[1..mh,1..mh-1,1..nh]of integer;
  M42:array[1..mh,1..mh-1,1..nh]of integer;
  M43:array[1..mh,1..mh-1,1..nh]of integer;
  M44:array[1..mh,1..mh-1,1..nh]of integer;
  M45:array[1..mh,1..mh-1,1..nh]of integer;
  M46:array[1..mh,1..mh-1,1..nh]of integer;
  T:array[1..mh]of integer;
  Sg:array[1..mh]of integer;
  BD:array[1..mh,1..mh]of integer;
  ME1:array[1..mh]of integer;
  ME2:array[1..mh]of integer;
  MAS:array[1..mh]of integer;

  PPl:array[1..nh]of integer;
  PPh:array[1..nh]of integer;
  PPb:array[1..2*nh]of integer;
  PPe:array[1..2*nh]of integer;
  TTb:array[1..2*nh]of integer;
  TTe:array[1..2*nh]of integer;


  n,m,aa,bb,ac:longint; //integer;
  s,za,zb,ca,cb,cc,ff,fp,fk,fr,df1,df2,nfe,nfr,pp,pe:integer;
  mo,ka,kb,kp,e2,e4,e6,e8,ey:integer;
  sh,sl,ep,et,ce,ee,es,dp,dt:integer;
  er1,er2,e1,ex,v1,v2,vv,ev1,ev2,nhn:integer;

implementation

{$R *.dfm}

procedure TForm1.Button2Click(Sender: TObject);
begin
 if CloseQuery then Close;
end;

procedure TForm1.Button1Click(Sender: TObject);
  var i,j,h1:integer;
begin
  repeat
  Button2.Top:=0;  Button2.Left:=0;
  A1.Visible:=True;
  B1.Visible:=True;
  N8.Visible:=True;
  Application.ProcessMessages;
  m:=StrToInt(Edit9.Text);
  if (m<2)or(m>5) then
  Panel34.Caption:=
  ' Неверная длина адреса '+IntToStr(m);
  until (m>1)and(m<6);
  Panel34.Caption:=
  ' Длина адреса сдвигателя '+IntToStr(m);
```

```
n:=(1 shl m)-1;
Panel34.Left:=20*(n+5); Panel34.Top:=60;
Button1.Visible:=False;
Edit9.Visible:=False;
Button1.Visible:=false;
Panel12.Visible:=True;
Image1.Left:=4;              Image1.Top:=56;
Image1.Width:=20*n+64*m+148;  Image1.Height:=m*(m+2)*16+60;
Image2.Left:=8;              Image2.Top:=56;
Image2.Width:=32*(2*n+1);     Image2.Height:=32*(n+3);
if m*(m+2)*16+60<432 then Image1.Height:=432;
if 32*(n+6)<352 then Image2.Height:=352;
with Image1.Canvas do
begin
Rectangle(36,4,20*n+40,28);
Rectangle(36,36,20*m+40,60);
Button3.Top:=60;        Button3.Left:=20*n+52;
Button4.Top:=92;        Button4.Left:=12;
Button3.Visible:=True;  Button4.Visible:=True;
Button5.Visible:=True;  Button6.Visible:=True;
Button7.Visible:=True;  Button15.Visible:=False;
Panel3.Visible:=True;   Panel4.Visible:=True;
Panel12.Visible:=True;  Panel22.Visible:=True;
Panel22.Top:=52;        Panel22.Left:=20*n+44;
Edit1.Visible:=True;    Edit2.Visible:=True;
Edit3.Visible:=True;    Edit4.Visible:=True;

for i:=1 to m do
begin
 h1:=20*i+20;
 BBb[i]:=0;
 Pen.Color:=clBlue;
 Brush.Color:=clMoneyGreen;
   Rectangle(h1,40,h1+16,56);
   TextOut(h1+4,41,'0');
end;
for i:=1 to n do
begin
 h1:=20*i+20;
 AAb[i]:=0;
 Pen.Color:=clBlue;
 Brush.Color:=clMoneyGreen;
   Rectangle(h1,8,h1+16,24);
   TextOut(h1+4,9,'0');
end;
h1:=52;
for i:=1 to m do
begin
 Rectangle(4,h1+28,28,h1+32*(m+1-i)+16);
 for j:=1 to m-i+1 do
 begin
   h1:=h1+32;
```

```
     Rectangle(4,h1-4,28,h1+20);
     Rectangle(8,h1,24,h1+16);
     TextOut(12,h1+1,'0');
    end;
    h1:=h1+16;
   end;
  end;

end;

procedure TForm1.Button5Click(Sender: TObject);
var h1,ih:integer;
begin
 aa:=StrToInt(Edit1.Text);
 h1:=aa;
 for ih:=1 to n do
  begin
   AAb[ih]:=h1 and 1;
   h1:=h1 shr 1;
   Image1.Canvas.TextOut(20*(n+2-ih)+4,9,
                IntToStr(AAb[ih]));
  end;
  s:=1-AAb[n];
end;

procedure TForm1.Button6Click(Sender: TObject);
var h1,h2,h3,ih,jh:integer;
begin
 bb:=StrToInt(Edit2.Text);
 h1:=bb;
 for ih:=1 to m do
 begin
  BBb[ih]:=h1 and 1;
  h1:=h1 shr 1;
  Image1.Canvas.TextOut(20*(m+2-ih)+4,41,
            IntToStr(BBb[ih]));
 end;

  h2:=1;  h3:=64;
  for ih:=1 to m do
  begin
   h1:=bb;
   h2:=h2 shl 1;
   BD[ih,1]:=h1 and (h2-1);
   h1:=h1 shr ih;
   Image1.Canvas.TextOut(12,h3+21,
                IntToStr(BD[ih,1]));
   h3:=h3+32;
   for jh:=2 to m+1-ih do
   begin
    BD[ih,jh]:=(h1 and 1) shl (jh+ih-2);
    h1:=h1 shr 1;
```

```
    Image1.Canvas.TextOut(12,h3+21,
                IntToStr(BD[ih,jh]));
    h3:=h3+32;
   end;
   h3:=h3+16;
  end;
end;

procedure TForm1.OnMauseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
  var ii,jj,xx,yy,h1,h2:integer;
begin
  ii:=(X)div 20;
  jj:=(Y)div 20;
  xx:=ii*20+8;
  yy:=jj*20+8;
  if (X-xx<17)and(Y-yy<17)and(yy=8)and
       (ii<2+n)and(ii>=2) then
  begin //інвертування біт числа A
   h1:=n+2-ii;
   AAb[h1]:=1-AAb[h1];
   Image1.Canvas.TextOut(20*ii+4,9,
                IntToStr(AAb[h1]));
  end;
  s:=1-AAb[n];
  if (X-xx<17)and(Y-yy<17)and(yy=48)and
       (ii<=2+m)and(ii>=2) then
  begin  // інвертування біт числа B
   h1:=m+2-ii;
   BBb[h1]:=1-BBb[h1];
   Image1.Canvas.TextOut(20*ii+4,41,
                IntToStr(BBb[h1]));
  end;

end;

procedure TForm1.Button3Click(Sender: TObject);
// Запис числа A в двійкову сітку
 var ih:integer;
begin
 aa:=0;
 for ih:=1 to n do
  aa:=aa+AAb[ih]shl(ih-1);
  Edit1.Text:=IntToStr(aa);
end;

procedure TForm1.Button4Click(Sender: TObject);
//Запис числа B в двійкову сітку
 var i,ih,jh,h1,h2,h3,h4,h5:integer;
begin
 bb:=0;
 for ih:=1 to m do
```

```
  bb:=bb+BBb[ih]shl(ih-1);
  Edit2.Text:=IntToStr(bb);

  h2:=1;  h3:=64;
  for ih:=1 to m do
  begin
   h4:=0;
   h1:=bb;
   h2:=h2 shl 1;
   BD[ih,1]:=h1 and (h2-1);
   h1:=h1 shr ih;
   Image1.Canvas.TextOut(12,h3+21,
                IntToStr(BD[ih,1]));
   h4:=BD[ih,1];
   for i:=1 to n do
   begin
    h5:=20*i+20;
    if i<=h4  then
     Image1.Canvas.Brush.Color:=clAqua
     else Image1.Canvas.Brush.Color:=clMoneyGreen;
     Image1.Canvas.Rectangle(h5,h3+20,h5+16,h3+36);
   end;
   Image1.Canvas.Brush.Color:=clMoneyGreen;
   h3:=h3+32;
   for jh:=2 to m+1-ih do
   begin
    BD[ih,jh]:=(h1 and 1) shl (jh+ih-2);
    h1:=h1 shr 1;
    Image1.Canvas.TextOut(12,h3+21,
                IntToStr(BD[ih,jh]));
    h4:=h4+BD[ih,jh];
    for i:=1 to n do
    begin
     h5:=20*i+20;
     if i<=h4  then
      Image1.Canvas.Brush.Color:=clAqua
      else Image1.Canvas.Brush.Color:=clMoneyGreen;
      Image1.Canvas.Rectangle(h5,h3+20,h5+16,h3+36);
    end;
    Image1.Canvas.Brush.Color:=clMoneyGreen;
    h3:=h3+32;
   end;
   h3:=h3+16;
  end;
end;

procedure TForm1.Button7Click(Sender: TObject);
// Старт
  var h1,h2,h3,h4,h5,h6,h7,g1,g2,g3,g4,g5,h,i,j,r,d,k,x,ih,jh:integer;
begin
 if Button7.Caption='Старт' then
 begin
```

```
if ((za>1)or(zb>1))then
begin
 Button2.Visible:=False;
 Button7.Caption:='Финиш';
end;
Randomize;
h6:=1 shl m;
h7:=1 shl n;
for r:=1 to m do
begin
 ME1[r]:=0;
 ME2[r]:=0;
end;
cc:=0;
if za<2 then ca:=1 else ca:=0;
if za=2 then aa:=0;
repeat  //перебір по A
 if za and 1=1 then aa:=Random(h7);
 h1:=aa;
 Edit1.Text:=IntToStr(aa);
 for ih:=1 to n do
 begin
  AAb[ih]:=h1 and 1;
  h1:=h1 shr 1;
  Image1.Canvas.TextOut(20*(n+2-ih)+4,9,
     IntToStr(AAb[ih]));
 end;
 s:=1-AAb[n];
 if zb<2 then cb:=1 else cb:=0;
 if zb=2 then bb:=0;
 repeat  //їåðåáîð ïî B
  if zb and 1=1 then bb:=Random(h6);
  h1:=bb;
  Edit2.Text:=IntToStr(bb);
  for ih:=1 to m do
  begin
   BBb[ih]:=h1 and 1;
   h1:=h1 shr 1;
   Image1.Canvas.TextOut(20*(m+2-ih)+4,41,
           IntToStr(BBb[ih]));
  end;
  h2:=1;
  h3:=64;
  for ih:=1 to m do
  begin
   h4:=0;
   h1:=bb;
   h2:=h2 shl 1;
   BD[ih,1]:=h1 and (h2-1);
   h1:=h1 shr ih;
   Image1.Canvas.TextOut(12,h3+21,
     IntToStr(BD[ih,1]));
```

```
 h4:=BD[ih,1];
 for i:=1 to n do
 begin
  h5:=20*i+20;
  if i<=h4  then Image1.Canvas.Brush.Color:=clAqua
   else Image1.Canvas.Brush.Color:=clMoneyGreen;
  Image1.Canvas.Rectangle(h5,h3+20,h5+16,h3+36);
 end;
 Image1.Canvas.Brush.Color:=clMoneyGreen;
 h3:=h3+32;
 for jh:=2 to m+1-ih do
 begin
  BD[ih,jh]:=(h1 and 1) shl (jh+ih-2);
  h1:=h1 shr 1;
  Image1.Canvas.TextOut(12,h3+21,
    IntToStr(BD[ih,jh]));
  h4:=h4+BD[ih,jh];
  for i:=1 to n do
  begin
   h5:=20*i+20;
   if i<=h4 then Image1.Canvas.Brush.Color:=clAqua
    else Image1.Canvas.Brush.Color:=clMoneyGreen;
  Image1.Canvas.Rectangle(h5,h3+20,h5+16,h3+36);
  end;
  Image1.Canvas.Brush.Color:=clMoneyGreen;
  h3:=h3+32;
 end;
 h3:=h3+16;
end;

cc:=cc+1;
h5:=84;
for r:=1 to m do
begin
 MAS[r]:=0;
 Sg[r]:=s;    // Копіювання знака A
end;
for r:=1 to m do     // Перебір рівнів розпаралелення
begin
 x:=1 shl r;         //2 ст r
 T[r]:=((4*r+(r+2)*x+1)+10*(m-r))*n;   // Кількість точок схеми
 fp:=Random(T[r]);   // Вибір місця несправности
 fk:=Random(1);      // Вибір виду несправности
 for h1:=1 to m do   // Копіювання масиву розрядів B
  BBc[h1]:=BBb[h1];
 for h1:=1 to n do   // Копіювання масиву розрядів A
  AAc[h1]:=AAb[n+1-h1];
 g1:=4*r*n;
 g2:=g1+n*x*(r+2);
 g3:=g2+n;                    //  g3:=g2+(2*x+1)*n;
 if (ff=1)and(fk=0) then                // Zero
 begin
```

```
 for k:=1 to r do   //Перебір розрядів адреса
  for i:=1 to n do    //Перебір мультиплексорів першого рівня
   if fp=((i-1)*r+k-1)*4+1 then
   begin
    BBc[k]:=0;
    break;
   end;
  for i:=1 to n do    // Перебір мультиплексорів першого рівня
   for j:=1 to x do   // Перебір AND
    if fp=g1+((i-1)*x+j-1)*(r+2)+r+1 then     // r+1 point in AND
     if i<j then Sg[r]:=0 else AAc[i-j+1]:=0;
  for d:=r+1 to m do                    //Zero d
   for i:=1 to n do
   begin
    h1:=g3+((d-r-1)*n+i-1)*10;
    h3:=1 shl(d-1);
    if fp=h1+1 then
    begin
     BBc[d]:=0;
     break;
    end;
    if fp=h1+6 then M3[r,d-1,i]:=0;
    if (fp=h1+7)and(i<=h3) then Sg[r]:=0 ;
   end;
end;

for i:=1 to n do    //Перебір мультиплексорів першого рівня
begin
 for k:=1 to r do   //Перебір розрядів адреса
 begin
  h1:=((i-1)*r+k-1)*4;
  M1[r,i,k]:= not BBc[k];
  if ff=1 then
  begin
   if (fp=h1+1)and(fk=1) then M1[r,i,k]:=0;         // point 1 in NOT
   if fp=h1+2 then M1[r,i,k]:=fk;                   // point 2 in NOT
   if (fp=h1+3)and(fk=0) then M1[r,i,k]:=0;         // point 3 in NOT
  end;

  if (ff=1)and(fk=0) then
   for j:=1 to x do                                 // Перебір AND
   begin
    BBd[k]:=(j-1) and (1 shl (k-1));
    if (fp=g1+((i-1)*x+j-1)*(r+2)+k)and(BBd[k]=0)
         then M1[r,i,k]:=0;                    // k=1..r point in AND
   end;
  M1[r,i,k+r]:= not M1[r,i,k];
  if (ff=1)and(fk=0) then
   for j:=1 to x do                                 // Перебір AND
   begin
    BBd[k]:=(j-1) and (1 shl (k-1));
    if (fp=g1+((i-1)*x+j-1)*(r+2)+k)and(BBd[k]=1)
```

```
          then M1[r,i,k+r]:=0;                                  // k=1..r point in AND
   end;
  if ff=1 then
  begin
   if (fp=h1+3)and(fk=1) then M1[r,i,k+r]:=0;                   // point 3 in NOT
   if fp=h1+4 then M1[r,i,k+r]:=fk;                     // point 4 in NOT
  end;
 end;

 h2:=0;
 for j:=1 to x do                                               // Перебір ел. I мультиплексора
 begin
  if i<j then h3:=Sg[r] else h3:=AAc[i-j+1];                    // Інф. Вх.
  M2[r,i,j,r+1]:=h3;
  if (ff=1)and(fp=g1+((i-1)*x+j-1)*(r+2)+r+1)and(fk=1)
     then M2[r,i,j,r+1]:=1;                                     // r+1 point in AND
  h4:=j-1;
  for k:=1 to r do
  begin
   BBd[k]:=h4 and 1;
   h4:=h4 shr 1;
   M2[r,i,j,k]:=M1[r,i,k] and not BBd[k] or
          M1[r,i,k+r] and BBd[k];
   if (ff=1)and(fp=g1+((i-1)*x+j-1)*(r+2)+k)and(fk=1)
          then M2[r,i,j,k]:=1;                                  // k=1..r point in AND
   h3:=h3 and M2[r,i,j,k];
  end;
  M2[r,i,j,r+2]:=h3;
  if (ff=1)and(fp=g1+((i-1)*x+j-1)*(r+2)+r+1)
          then M2[r,i,j,r+2]:=fk;                               // r+2 point in AND
  h2:=h2 or M2[r,i,j,r+2];
 end;
 M3[r,r,i]:=h2;
 if (ff=1)and(fp=g2+i) then M3[r,r,i]:=fk;
 Image1.Canvas.TextOut(20*i+24,h5+1,
             IntToStr(M3[r,r,i]));
 if r=m then
  MAS[m]:=MAS[m]*2+M3[r,r,i];
end;
h5:=h5+32;
for d:=r+1 to m do
begin
 h3:=1 shl(d-1);
 for i:=1 to n do
 begin
  h1:=g3+((d-r-1)*n+i-1)*10;
  M41[r,d,i]:= not BBc[d];
  if ff=1 then
  begin
   if (fp=h1+1)and(fk=1) then M41[r,d,i]:=0;                            // point 1 in MX
   if fp=h1+2 then M41[r,d,i]:=fk;                              // point 2 in MX
   if ((fp=h1+3)or(fp=h1+5))and(fk=0) then M41[r,d,i]:=0;       // point 3 in NOT
```

```
   end;
   M42[r,d,i]:= not M41[r,d,i];
   if ff=1 then
   begin
    if (fp=h1+3)and(fk=1) then  M42[r,d,i]:=0;                    // point 3 in MX
    if fp=h1+4 then M42[r,d,i]:=fk;                               // point 4 in MX
   end;
   M43[r,d,i]:=M3[r,d-1,i];
   if (ff=1)and(fk=0)and((fp=h1+6)or((fp=h1+7-10*h3)and(i+h3<=n)))
          then M3[r,d-1,i]:=0;                                    // point 6,7 in MX
   if i<=h3 then M44[r,d,i]:=Sg[d]
    else M44[r,d,i]:=M3[r,d-1,i-h3];
   if (ff=1)and(fp=h1+7)and(fk=1)
          then M44[r,d,i]:=1;                                     // point 7 in MX
   M45[r,d,i]:=M41[r,d,i] and M43[r,d,i];
   M46[r,d,i]:=M42[r,d,i] and M44[r,d,i];
   if (ff=1)then
   begin
    if fp=h1+8 then M45[r,d,i]:=fk;                               // point 8 in MX
    if fp=h1+9 then M46[r,d,i]:=fk;                               // point 9 in MX
   end;
   M3[r,d,i]:=M45[r,d,i] or M46[r,d,i];
   Image1.Canvas.TextOut(20*i+24,h5+1,
          IntToStr(M3[r,d,i]));
   if (ff=1)and(fp=h1+10) then M3[r,d,i]:=fk;                     // point 8 in MX
   if (r<m)and(d=m) then
    MAS[r]:=MAS[r]*2+M3[r,d,i];
  end;
  h5:=h5+32;
 end;
 h5:=h5+16;
end;
ac:=0;
for i:=1 to n do
begin
 if i>n-bb then ASb[i]:=s else ASb[i]:=AAb[i+bb];
 ac:=ac+ASb[i]shl(i-1);
end;
Edit3.Text:=IntToStr(ac);
Edit4.Text:=IntToStr(MAS[1]);
for r:=1 to m do
begin
 if ac<>MAS[r] then ME1[r]:=ME1[r]+1;
 {if ac<>MAS[r] then
  repeat
    Application.ProcessMessages;
   until Panel13.Color=clRed;  }
 if abs(ac-MAS[r])>=(1shl(es))then ME2[r]:=ME2[r]+1;
end;

with Image1.Canvas do
begin
```

```
Pen.Color:=clBlue;
Brush.Color:=clCream;
Font.Color:=clBlack;
g4:=20*(n+4); g5:=44;
MoveTo(g4-4,g5-4);
LineTo(g4+64*(m+1),g5-4);
MoveTo(g4-4,g5-4);
LineTo(g4-4,g5+384);
MoveTo(g4+64*(m+1),g5-4);
LineTo(g4+64*(m+1),g5+384);
for i:=1 to m+1 do
begin
 Rectangle(g4,g5,g4+60,g5+380);
 g4:=g4+64;
end;
g4:=20*(n+4); g5:=44;
for i:=1 to 6 do
begin
 g5:=g5+60;
 MoveTo(g4,g5);
 LineTo(g4+64*(m+1)-4,g5);
 g5:=g5+4;
 MoveTo(g4,g5);
 LineTo(g4+64*(m+1)-4,g5);
end;
g4:=20*(n+4); g5:=44;
Font.Color:=clRed;
TextOut(g4+17,g5+24,'СДВ');
Font.Color:=clBlack;
TextOut(g4+8,g5+88,'к-во точек');
TextOut(g4+8,g5+152,'Ошибка');
TextOut(g4+8,g5+216,'Сущест');
TextOut(g4+8,g5+280,'Несущ');
Font.Color:=clRed;
TextOut(g4+17,g5+344,'ДКР');
Font.Color:=clBlack;
for i:=1 to m do
begin
 TextOut(g4+90,g5+24,IntToStr(i));
 g4:=g4+64;
end;
g4:=20*(n+4); g5:=44;
for i:=1 to m do
begin
 g4:=g4+64;
 Font.Color:=clBlue;
 TextOut(g4+24,g5+88,IntToStr(T[i]));
 h1:=(ME1[i]*2000 div cc)div 2;
 TextOut(g4+24,g5+152,
   IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
 h1:=(ME2[i]*2000 div cc)div 2;
 TextOut(g4+24,g5+216,
```

```
      IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
     h1:=((ME1[i]-ME2[i])*2000 div cc)div 2;
     TextOut(g4+24,g5+280,
      IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
     Font.Color:=clRed;
     h1:=((cc-ME2[i])*2000 div cc)div 2;
     TextOut(g4+22,g5+344,
      IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
     Font.Color:=clBlue;
    end;
   end;

   Application.ProcessMessages;
   if (za=3)and(zb>1) then cb:=1;
   if zb=2 then
   begin
    bb:=bb+1;
    if bb=h6 then
    begin
     cb:=1;
     bb:=0;
    end;
   end;
  until cb=1;
  if za=2 then
  begin
   aa:=aa+1;
   if aa=h7 then begin ca:=1;  aa:=0; end;
  end;
 until ca=1;
end;
Button2.Visible:=True;
Button7.Caption:='Старт';
ca:=1; cb:=1;
end;

procedure TForm1.A2Click(Sender: TObject);
begin
 za:=0;

end;

procedure TForm1.N1Click(Sender: TObject);
begin
 za:=1;
end;

procedure TForm1.N2Click(Sender: TObject);
begin
 za:=2;
end;
```

```
procedure TForm1.N3Click(Sender: TObject);
begin
 za:=3;
end;

procedure TForm1.N4Click(Sender: TObject);
begin
 zb:=0;
end;

procedure TForm1.N5Click(Sender: TObject);
begin
 zb:=1;
end;

procedure TForm1.N6Click(Sender: TObject);
begin
 zb:=2;
end;

procedure TForm1.N7Click(Sender: TObject);
begin
 zb:=3;
end;

procedure TForm1.N9Click(Sender: TObject);
begin
 ff:=0;
end;

procedure TForm1.N10Click(Sender: TObject);
begin
 ff:=1;
end;

procedure TForm1.Button12Click(Sender: TObject);
begin
 nfe:=1;
end;

procedure TForm1.FormDragOver(Sender, Source: TObject; X, Y: Integer;
  State: TDragState; var Accept: Boolean);
begin
  Accept:=Source is TPanel;
end;

procedure TForm1.FormDragDrop(Sender, Source: TObject; X, Y: Integer);
begin
    X:=X div 20;
    if X>n+2 then X:=n+2;
    if X<2 then X:=2;
    (Source as TPanel).Top:=52;
```

```
    (Source as TPanel).Left:=20*X;
   es:=n+2-X;
   Panel12.Caption:=IntToStr(es);
end;

procedure TForm1.Panel22MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  if button=mbLeft then
    TPanel(Sender).BeginDrag(false);
end;

procedure TForm1.Panel21Click(Sender: TObject);
begin
 Panel21.Tag:=1;
 Panel21.Visible:=false;
end;

procedure TForm1.Panel12Click(Sender: TObject);
var i,j,h1,xm,ym:integer;
begin
 mo:=1-mo;
 if mo=0 then
 begin
  Image1.Visible:=True;
  Image2.Visible:=False;
  Panel13.Visible:=False;
  Panel14.Visible:=False;
  Panel15.Visible:=False;
  Panel12.Caption:='ÓÌÍ';
  Panel12.Color:=clMoneyGreen;
  Button3.Top:=60;       Button3.Left:=32*n-20;
  Button3.Width:=25;     Button3.Height:=25;
  Button4.Top:=60;       Button4.Left:=64*n+12;
  Button4.Width:=25;     Button4.Height:=25;
  Panel22.Top:=116;  Panel22.Left:=20*(n-es)+44;
 end
 else
 begin
  Panel22.Top:=56;  Panel22.Left:=16*(3*n-es)+16;
  Image1.Visible:=False;
  Image2.Visible:=True;
  if Image2.Width<540 then Image2.Width:=540;
  Panel12.Caption:='ÊÎÍ';
  Panel12.Color:=clAqua;
  Panel13.Visible:=True;
  Panel14.Visible:=True;
  Panel15.Visible:=True;
  with Image2.Canvas do
  begin
   Pen.Color:=clBlue;
   Brush.Color:=clCream;
```

```
Font.Color:=clBlack;
Rectangle(32*n+10,6,48*n+14,26);
Rectangle(32*n+10,30,48*n+14,50);
Rectangle(32*n+10,78,64*n+14,98);
Rectangle(32*n+10,102,64*n+14,122);
Button3.Top:=62;       Button3.Left:=32*n-6;
Button3.Width:=21;     Button3.Height:=21;
Button4.Top:=86;       Button4.Left:=32*n-6;
Button4.Width:=21;     Button4.Height:=21;
for i:=1 to n do
begin
 h1:=48*n-16*i+12;
 Pen.Color:=clBlue;
 Brush.Color:=clCream;
 Rectangle(h1,8,h1+16,24);
 TextOut(h1+4,9,IntToStr(AAb[i]));
 Rectangle(h1,32,h1+16,48);
 TextOut(h1+4,33,IntToStr(BBb[i]));
 Rectangle(h1+16*n,80,h1+16+16*n,96);
 TextOut(h1+4+16*n,81,IntToStr(PPb[i]));
 Rectangle(h1,80,h1+16,96);
 TextOut(h1+4,81,IntToStr(PPb[i+n]));
 Rectangle(h1+16*n,104,h1+16+16*n,120);
 TextOut(h1+4+16*n,105,IntToStr(PPe[i]));
 Rectangle(h1,104,h1+16,120);
 TextOut(h1+4,105,IntToStr(PPe[i+n]));
end;
xm:=260;ym:=160;
  Pen.Color:=clBlue;
  Brush.Color:=clCream;
  Rectangle(xm+30,ym,xm+90,ym+80);
  MoveTo(xm+30,ym+40);
  LineTo(xm+60,ym+40);
  MoveTo(xm+60,ym);
  LineTo(xm+60,ym+80);
  MoveTo(xm,ym+20);
  LineTo(xm+30,ym+20);
  MoveTo(xm,ym+60);
  LineTo(xm+30,ym+60);
  MoveTo(xm+90,ym+40);
  LineTo(xm+210,ym+40);
  if nhn=0 then
  begin
  TextOut(xm+42,ym+12,'1');
  TextOut(xm+42,ym+52,'1');
  TextOut(xm+72,ym+12,'&');
  end
  else
  begin
  TextOut(xm+42,ym+12,'&');
  TextOut(xm+42,ym+52,'&');
  TextOut(xm+72,ym+12,'1');
```

```
  end;
  Rectangle(xm+120,ym+80,xm+180,ym+160);
  TextOut(xm+132,ym+92,'1');
  TextOut(xm+132,ym+132,'2');
  TextOut(xm+158,ym+92,'ÁC');
  MoveTo(xm+120,ym+120);
  LineTo(xm+150,ym+120);
  MoveTo(xm+150,ym+80);
  LineTo(xm+150,ym+160);
  MoveTo(xm,ym+100);
  LineTo(xm+120,ym+100);
  MoveTo(xm,ym+140);
  LineTo(xm+120,ym+140);
  MoveTo(xm+180,ym+120);
  LineTo(xm+210,ym+120);
  Brush.Color:=clWindow;
  TextOut(xm,ym+4,'A');
  TextOut(xm,ym+24,IntToStr(aa));
  TextOut(xm,ym+44,'B');
  TextOut(xm,ym+64,IntToStr(bb));
  TextOut(xm,ym+114,'V');
  TextOut(xm+10,ym+114,IntToStr(pp));
  TextOut(xm+70,ym+84,'V1');
  TextOut(xm+70,ym+104,IntToStr(v1));
  TextOut(xm+70,ym+124,'V2');
  TextOut(xm+70,ym+144,IntToStr(v2));
      TextOut(xm+200,ym+24,'E1');
  TextOut(xm+200,ym+44,IntToStr(er1));
      TextOut(xm+200,ym+104,'E2');
  TextOut(xm+200,ym+124,IntToStr(er2));
Pen.Color:=clTeal;
Rectangle(4,210,187,254);
Rectangle(4,256,187,328);
MoveTo(50,232);
LineTo(187,232);
MoveTo(4,280);
LineTo(187,280);
MoveTo(4,304);
LineTo(187,304);
MoveTo(142,232);
LineTo(142,328);
MoveTo(97,232);
LineTo(97,327);
MoveTo(52,210);
LineTo(52,327);
MoveTo(50,210);
LineTo(50,327);
Font.Color:=clRed;
TextOut(17,215,'СДВ');
Font.Color:=clBlack;
TextOut(97,215,'Кол т-к');
TextOut(57,236,'Ошибки');
```

```
  TextOut(102,236,'Существ');
  TextOut(147,236,'Несущ');
  TextOut(10,261,'ДКР');
  TextOut(10,285,' ');
  TextOut(10,309,'Старт');
 if cc>0 then
 begin
 Font.Color:=clRed;
 h1:=((ce+2*ex-e1-ep)*2000 div ce+1)div 2;
TextOut(17,236,
  IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
 Font.Color:=clBlue;
 h1:=(ex*20000 div ce+1)div 2;
TextOut(62,261,
  IntToStr(h1 div 100)+','+IntToStr(h1 mod 100)+'%');
 h1:=((ep-ex)*2000 div ce+1)div 2;
TextOut(62,285,
  IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
 h1:=(ep*2000 div ce+1)div 2;
TextOut(62,309,
  IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
 h1:=((e1-ex)*2000 div ce+1)div 2;
TextOut(107,261,
  IntToStr(h1 div 100)+','+IntToStr(h1 mod 100)+'%');
 h1:=((ce-e1-ep+ex)*2000 div ce+1)div 2;
TextOut(107,285,
  IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
 h1:=((ce-ep)*2000 div ce+1)div 2;
TextOut(107,309,
  IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
 h1:=(e1*2000 div ce+1)div 2;
TextOut(152,261,
  IntToStr(h1 div 100)+','+IntToStr(h1 mod 100)+'%');
 h1:=((ce-e1)*2000 div ce+1)div 2;
TextOut(152,285,
  IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
 h1:=(ce*2000 div cc+1)div 2;
TextOut(152,309,
  IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
  end;
 end;
end;
end;

procedure TForm1.Button15Click(Sender: TObject);
begin
 Panel34.Visible:=True;
 Panel34.Caption:=
 'Кількість рівнів розпаралелення (2-5)';
 Edit9.Visible:=True;
 Button15.Visible:=False;
 Button1.Visible:=True;
```

```
end;

procedure TForm1.Panel35Click(Sender: TObject);
begin
 nfe:=0;
end;

procedure TForm1.Panel13Click(Sender: TObject);
begin
vv:=1-vv;
if vv=0 then
 begin
  Panel13.Color:=clYellow;
  Panel13.Font.Color:=clBlack;
 end
 else
 begin
  Panel13.Color:=clRed;
  Panel13.Font.Color:=clWhite;
 end
end;

procedure TForm1.Panel14Click(Sender: TObject);
begin
ev1:=1-ev1;
Panel14.Caption:=IntToStr(ev1);
if ev1=0 then Panel14.Color:=clSkyBlue
else Panel14.Color:=clLime;
end;

procedure TForm1.Panel15Click(Sender: TObject);
begin
ev2:=1-ev2;
Panel15.Caption:=IntToStr(ev2);
if ev2=0 then Panel15.Color:=clSkyBlue
else Panel15.Color:=clLime;
end;

end.
```