

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

МОВСЕСЯН Агарон Міхитарович

ДИПЛОМНА РОБОТА МАГІСТРА

ДОСЛІДЖЕННЯ ЛОГАРИФМІЧНОГО КОНТРОЛЮ ПОМНОЖУВАЧА ПІД
ДІЄЮ КРАТНИХ НЕСПРАВНОСТЕЙ

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

Мовсесяну Агарону Міхитаровичу

1. Тема роботи “Дослідження логарифмічного контролю помножувача під дією кратних несправностей”

затверджена наказом по університету від “ ” 2021 р. № -в

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. Огляд рішень з обробки та контролю наближених даних

2. Завдання на проведення досліджень

3. Розробка та дослідження програмної моделі логарифмічного контролю матричного помножувача

4. Результати дослідження логарифмічного контролю матричного помножувача

5. Охорона праці та безпека в надзвичайних ситуаціях

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1) Визначення та обчислення логарифмічного контрольного коду числа

2) Логарифмічний контроль операції множення

3) Виявлення помилок в логарифмічному контролі добутку

4) Суттєві та несуттєві помилки в контролі добутку

5) Виконання контрольних обчислень

6) Моделювання логарифмічного контролю

7) Виявлення помилок в логарифмічному контролі помножувача

8) Пропуск помилок в логарифмічному контролі помножувача

9) Дослідження достовірності логарифмічного контролю результатів при всіх вірних розрядах

10) Достовірність логарифмічного контролю помножувача при зменшенні кількості вірних розрядах

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	<i>Вивчення науково-технічної та патентної літератури з питань контролю арифметичних операцій у матричних пристроях</i>	<i>015.03 - 16.04</i>	
2	<i>Визначення задачі на дослідження логарифмічного контролю матричного помножувача під дією кратних несправностей</i>	<i>17.04 - 30.06</i>	
3	<i>Дослідження вимог до програмної моделі логарифмічного контролю матричного помножувача</i>	<i>2.09 – 10.10</i> <i>11.10 - 27.10</i>	
4	<i>Розробка алгоритмів та програмної моделі логарифмічного контролю матричного помножувача</i>		
5	<i>Верифікація розробленої моделі на контрольних прикладах та одержання результатів моделювання</i>	<i>28.10 - 25.11</i>	
6	<i>Обробка і оцінка результатів моделювання логарифмічного контролю матричного помножувача під дією кратних несправностей</i>	<i>26.11 - 15.12</i>	

Студент _____ Мовсесян А.М.

Керівник роботи _____ Дрозд О.В.

№ рядка	Формат	Позначення	Найменування	Кільк.	№ екз.	Примітка
1			Документація загальна			
2			<u>Знов розроблена</u>			
3	A4	АМДР.АМ161.2222.01ПЗ	Пояснювальна записка	95		
4	A1	АМДР.АМ161.2222.01Д1	Визначення та обчислення	1		
5			логарифмічного контрольного			
6			коду числа			
7	A1	АМДР.АМ161.2222.01Д2	Логарифмічний контроль	1		
8			операції множення			
9	A1	АМДРЗАМ161.2222.01Д3	Виявлення помилок в	1		
10			логарифмічному контролі добутку			
11	A1	АМДР.АМ161.2222.01Д4	Суттєві та несуттєві помилки	1		
12			в контролі добутку			
13	A1	АМДР.АМ161.2222.01Д5	Виконання контрольних	1		
14			обчислень			
15	A1	АМДР.АМ161.2222.01Д6	Моделювання логарифмічного			
16			контролю			
17	A1	АМДР.АМ161.2222.01Д7	Виявлення помилок в	1		
18			логарифмічному контролі			
19			помножувача			
20	A1	АМДР.АМ161.2222.01Д8	Пропуск помилок в	1		
21			логарифмічному контролі			
22			помножувача			
23	A1	АМДР.АМ161.2222.01Д9	Достовірність логарифмічного	1		
24			контролю помножувача при всіх			
25			вірних розрядах			
26		АМДР.АМ161.2222.01Д10	Достовірність логарифмічного			
27			контролю помножувача при			
28			зменшенні кількості вірних розрядах			

АММР.АМ161.2222

Ізм.	Лист	№ документа	Підпис	Дата				
Розробив		Мовсесян			Дослідження логарифмічного контролю помножувача під дією кратних несправностей Відомість дипломної роботи	Лист.	Лист	Листів
Перевірив		Дрозд					1	1
Н. контр.					ОНПУ ІКС КІСМ АМ161			
Затверд.								

АНОТАЦІЯ

Мовсесян Агарон Міхитарович. Дослідження логарифмічного контролю помножувача під дією кратних несправностей. – Магістерська кваліфікаційна робота. – Одеса, 2021: 95 с., 23 рис., 19 табл., 1 додаток, 15 посилань

Об'єкт дослідження – процес логарифмічного контролю, що відбувається при обробці наближених даних у помножувачі під дією кратних несправностей.

Предмет дослідження – достовірність методу логарифмічного контролю наближених результатів у помножувачі під дією кратних несправностей.

Мета роботи – оцінка достовірності методу логарифмічного контролю наближених даних для робочого діагностування матричного помножувача з фіксованою точкою в умовах дії кратних несправностей.

Методи досліджень базуються на теорії робочого діагностування арифметичних пристроїв, прикладній теорії цифрових автоматів, теорії ймовірностей, а також елементах теорії алгоритмів та математичної логіки.

Магістерська робота присвячена проблемі робочого діагностування арифметичних пристроїв для обробки наближених даних. Сучасні методи робочого діагностування, що розроблені для точних даних, мають низьку достовірність контролю наближених результатів, ініціюючи розробку та аналіз нових методів контролю. Проведено дослідження методу логарифмічного контролю щодо його ефективності у виявленні кратних несправностей та оцінці достовірності наближених результатів для матричного помножувача. Розроблені й верифіковані алгоритми та програмна модель логарифмічного контролю матричного помножувача під дією кратних несправностей. Досліджено ймовірності появи, виявлення й пропуску суттєвих і несуттєвих помилок, а також достовірність логарифмічного контролю наближених результатів порівняно до контролю за модулем.

РОБОЧЕ ДІАГНОСТУВАННЯ, МАТРИЧНИЙ ПОМНОЖУВАЧ, ОБРОБКА НАБЛИЖЕНИХ ДАНИХ, ДОСТОВІРНІСТЬ, ЛОГАРИФМІЧНИЙ КОНТРОЛЬ, КРАТНІ НЕСПРАВНОСТІ.

ABSTRACT

Movsesian Aharon Mikhitarovych. Investigation of the logarithmic checking of the multiplier under conditions of multiple faults. – The master qualifying work. – Odessa, 2021: 95 p., 23 fig., 19 tables, 1 application, 15 references.

An object of research is process of the logarithm checking executed during processing the approximate data in multiplier under the influence of multiple faults.

A subject of research is reliability of logarithm checking the approximate results in multiplier under the influence of multiple faults.

A purpose of the work is evaluating the reliability of logarithm checking method of the approximate results for on-line testing of iterative array fixed-point multiplier in the conditions of action of multiple faults.

The methods of researches are based on on-line testing theory of arithmetic devices, applied theory of digital machines, probabilities theory and elements of algorithm and mathematical logic theory.

Master work is devoted to a problem of on-line testing the arithmetic devices for processing the approximate data. Modern methods of on-line testing developed for the exact data have low reliability of the checking the approximate results initiating the new methods development. Research of the logarithm checking concerning its efficiency in multiply faults detection and an evaluation of results reliability for fixed-point iterative array multiplier is carried out. Algorithms and program model of the logarithm checking the iterative array multiplier in condition of multiply faults are developed and tested up. Probabilities of occurrence, detection and skipping both the essential and inessential errors and reliability of logarithm checking the approximated results in comparison with residue checking are investigated.

ON-LINE TESTING, ITERATIVE ARRAY MULTIPLIER, APPROXIMATE DATA PROCESSING, RELIABILITY, LOGARITHM CHECKING, MULTIPLY FAULTS.

ЗМІСТ

Вступ	5
1 Огляд рішень з обробки та контролю наближених даних	10
1.1 Робоче діагностування обчислювальних пристроїв	10
1.2 Логарифмічний контроль матричного помножувача	11
1.3 Достовірність контролю результатів наближених обчислень	17
1.4 Висновки	21
2 Завдання на проведення досліджень	22
2.1 Найменування та область застосування	22
2.2 Призначення досліджень	22
2.3 Організація досліджень	23
2.3 Визначення задачі	25
3 Розробка та дослідження програмної моделі логарифмічного контролю матричного помножувача під дією кратних несправностей	27
3.1 Структура програмної моделі логарифмічного контролю	27
3.2 Функціонування програмної моделі логарифмічного контролю	28
3.3 План проведення досліджень логарифмічного контролю матричного помножувача під дією кратних несправностей.	29
3.4 Основні алгоритми програмної моделі логарифмічного контролю матричного помножувача	32
3.5 Висновки	38
4 Результати дослідження логарифмічного контролю матричного помножувача під дією кратних несправностей	40
4.1 Використання програмної моделі логарифмічного контролю для проведення досліджень	40

4.2 Дослідження ймовірностей виявлення та пропуску відмов логарифмічним контролем	46
4.3 Висновки	74
Висновок	75
Перелік джерел посилань	77
Додаток А. Текст програмної моделі логарифмічного контролю матричного помножувача	79

ВСТУП

Однією з найважливіших характеристик обчислювальних пристроїв (ОП) є достовірність результатів обчислень. Ця характеристика визначає ступінь відповідності обчислених результатів дійсним значенням. За рівнем достовірності результатів, що обчислюються в комп'ютерних системах, судять про можливість використання обчислених результатів та умовах застосування обчислювальної техніки в цілому [1].

Надійність сучасних ОП перебуває на високому рівні. Зі збільшенням надійності елементної бази створюються умови для ускладнення технічних рішень, що використовуються для розпаралелювання обчислень і підвищення продуктивності комп'ютерних систем та їх компонентів. Тому рівень надійності складних систем зберігається на рівні, що вимагає постійної оцінки достовірності результатів обчислень.

Достовірність результатів обчислень оцінюється з використанням методів та засобів робочого діагностування, які мають різну ефективність при виконанні в комп'ютерних системах точних та наближених обчислень [2, 3].

Робоче діагностування ОП, що виконують точні обчислення, вивчені досить повно. Обробка наближених даних істотно міняє умови робочого діагностування, вимагаючи додаткового вивчення поведінки ОП при його кратних несправностях і можливостей оцінки достовірності результатів методами робочого діагностування.

При обробці наближених даних несправності знижують вплив на достовірність результатів, викликаючи як істотні, так і несуттєві для неї помилки. Методи робочого діагностування демонструють нова якість відбракувати достовірні результати при виявленні несуттєвих помилок [4].

Операція множення присутнє в нормальній формі подання наближеного числа й тому є ключовою арифметичною операцією для вивчення особливостей діагностування ОП при обробці наближених даних. Тому важливо розглянути

виконання й робоче діагностування сучасних матричних помножувачів, що виконують цю операцію[5, 6].

Традиційні методи робочого діагностування були розроблені для випадку обробки точних даних і стосовно до наближених обчислень недостатньо ефективні [7, 8].

Тому доцільно оцінити ефективність інших методів робочого діагностування, серед яких виділяється логарифмічний контроль.

Метод логарифмічного контролю характеризується можливістю оцінювати величину результату, а також величину помилки, що викликається несправністю цифрової схеми.

Метод логарифмічного контролю використовує природну інформаційну надмірність форматів з фіксованою точкою, що дозволяє просто реалізувати для нього засоби робочого діагностування [9, 10].

Метою магістерської роботи є оцінка достовірності методу логарифмічного контролю наближених даних для робочого діагностування матричного помножувача з фіксованою точкою.

Для досягнення поставленої мети вирішуються наступні задачі:

- вивчається метод логарифмічного контролю операції множення наближених даних у форматах з фіксованою точкою;
- вивчається програмна модель матричного помножувача з фіксованою точкою;
- розробляються алгоритми та програмна модель засобів логарифмічного контролю помножувача наближених даних у форматах з фіксованою точкою;
- перевіряється правильність розробленої програмної моделі логарифмічного контролю;
- на програмній моделі досліджується виявляюча здатність методу логарифмічного контролю стосовно істотних і несуттєвих помилок, що викликаються кратними несправностями матричного помножувача;

- результати моделювання порівнюються з оцінками числового контролю за модулем три, що має найбільшу поширеність при вирішенні задач робочого діагностування арифметичних операцій в обчислювальних пристроях.

Об'єктом дослідження є процес логарифмічного контролю, що відбувається при обробці наближених даних у матричному помножувачі з фіксованою точкою.

Предметом дослідження є достовірність методу логарифмічного контролю наближених результатів, що обчислюються у матричному помножувачі з фіксованою точкою.

До елементів наукової новизни належить наступне:

- створення програмної моделі засобів логарифмічного контролю наближених результатів у матричному помножувачі з фіксованою точкою;
- одержання оцінок ймовірності виявлення та пропуску суттєвих і несуттєвих помилок, що викликаються кратними несправностями матричного помножувача.
- порівняльна оцінка достовірності методу логарифмічного контролю наближених результатів та визнаного основним для арифметичних операцій числового контролю за модулем три в матричному помножувачі з фіксованою точкою.

Практична значимість виконаних досліджень методу логарифмічного контролю матричного помножувача полягає у використанні оцінок його достовірності для підвищення ефективності застосування методу в області комп'ютерної обробки наближених даних.

Магістерська робота містить вступ, чотири основні розділи, висновок, додаток, що містить текст програмної моделі логарифмічного контролю помножувача з фіксованою точкою, та перелік використаних посилань на літературні джерела.

У вступі викладено основні положення магістерської роботи – актуальність теми, мета дослідження, задачі, що розв’язуються для її досягнення, об’єкт та предмет дослідження, методи, що були задіяні при його проведенні а також елементи наукової новизни й практичне значення одержаних результатів. Завершується вступ викладенням структури магістерської роботи.

У першому розділі розглядаються основні положення щодо виконання арифметичних операцій на матричних обчислювальних пристроях, особливості комп’ютерної обробки наближених даних, засади робочого діагностування, найбільш поширені методи робочого діагностування. Крім того викладені основи логарифмічного контролю та особливості його сучасного використання, а також проблеми що пов’язані з логарифмічним контролем обчислювальних пристроїв та потребують свого вирішення. Найбільша увага приділяється достовірності одержуваних результатів та її контролю методами та засобами робочого діагностування в умовах дії кратних несправностей.

У другому розділі викладено вихідні дані та вимоги для проведення дослідження логарифмічного контролю наближених результатів у матричному помножувачі з фіксованою точкою. Обґрунтовується необхідності проведення такого дослідження відповідно до нових вимог, що встановлюються при виконанні наближених обчислень у сучасних комп’ютерних системах. Конкретизуються завдання щодо розробки програмної моделі логарифмічного контролю матричного помножувача в умовах дії кратних несправностей.

Третій розділ містить аналіз вимог до програмної моделі логарифмічного контролю матричного помножувача з фіксованою точкою. Крім того, подається опис основних з розроблених алгоритмів, щодо виконання контрольних обчислень в програмній моделі логарифмічного контролю матричного помножувача в умовах дії кратних несправностей.

За алгоритмами створюється програмна модель логарифмічного контролю матричного помножувача з фіксованою точкою та виконується її верифікація на контрольних прикладах.

Програмна модель логарифмічного контролю матричного помножувача з фіксованою точкою будується, використовуючи мовні засоби DELFI, і містить демонстраційні та керуючі елементи, які відповідно показують вихідні данні та результати моделювання і забезпечують їх функціонування у передбачених режимах.

Верифікація програмної моделі виконана шляхом перевірки на тестових прикладах її початкових режимів та режимів функціонування, спостерігаючи демонстраційні елементи та використовуючи керуючі елементи, а також достовірності отриманих результатів моделювання логарифмічного контролю матричного помножувача в умовах дії кратних несправностей.

У четвертому розділі викладено план дослідження логарифмічного контролю щодо оцінки його достовірності для наближених результатів, що обчислюються в матричному помножувачі з плаваючою точкою в умовах дії кратних несправностей, а також результати моделювання та їх обробку до остаточних результатів дослідження.

План проведення дослідження логарифмічного контролю матричного помножувача мантис щодо визначення його достовірності для обробки наближених даних включає етап підготовки моделювання встановленням необхідних початкових умов, саме моделювання та обчислення ймовірностей виявлення та пропуску відмов за суттєвими та несуттєвими помилками, що викликані кратними несправностями матричного помножувача з фіксованою точкою, а також достовірності логарифмічного контролю наближених результатів.

1 ОГЛЯД РІШЕНЬ З ОБРОБКИ ТА КОНТРОЛЮ НАБЛИЖЕНИХ ДАНИХ

1.1 Робоче діагностування обчислювальних пристроїв

Робоче діагностування ОП спрямоване на виконання контролю достовірності результатів обчислень в процесі виконання основних операцій на робочих вхідних послідовностях. За методами та засобами робочого діагностування забезпечується контроль результатів шляхом виявлення помилок, що викликаються несправностями цифрових схем у процесі обчислень [1, 2].

На ефективність методів робочого діагностування суттєвий вплив створює тип даних, що обробляються в ОП.

Оброблювані дані відносяться до точних або наближених даних. Точні дані – то є нумеровані числа, тобто цілі за своєю природою. Це порядкові числівники, що використовуються для нумерації елементів множин. Інші дані є результатами вимірів або їхньої обробки та належать до наближених чисел, що становлять більшість даних для комп'ютерних обчислень [11].

Точне дане складається з вірних розрядів, перекручування яких несправністю завжди приводить до суттєвої помилки, що знижує достовірність результату.

Наближене дане складається зі старших вірних і молодших невірних розрядів. Вага невірних розрядів є нижчою за рівень абсолютної похибки, й тому помилки в невірних розрядах є несуттєвими для достовірності наближених результатів.

Ефективність методів робочого діагностування ОП визначається їхньою здатністю не пропускати суттєві помилки та ігнорувати несуттєві помилки.

До традиційних методів робочого діагностування відноситься контроль за модулем, а також контроль за паритетом [12]. Ці методи є ефективними для

випадку обробки точних даних, де всі помилки, що викликаються несправностями цифрової схеми є завжди суттєвими.

Традиційні методи робочого діагностування мають високу ймовірність виявлення помилки, однак не розрізняють суттєві та несуттєві помилки [13].

Для контролю точних даних перша властивість дає певні переваги для досягнення високої достовірності контролю результатів, а друга не має ніякого сенсу та значення.

Для наближених даних несуттєві помилки викликаються значно частіше суттєвих. Тому з високою ймовірністю виявляються в основному несуттєві помилки, що приводить до відбраковування достовірних результатів і низкою ефективності традиційних методів.

Логарифмічний контроль перевіряє дані, порівнюючи числове данне з його логарифмічною оцінкою [8].

Логарифмічний контроль тим складніше, чим вище вимоги до ймовірності виявлення помилок, оскільки тим точніше та складніше обчислюються логарифмічні оцінки чисел. Тому для точних даних цей метод не зміг скласти конкуренцію традиційним методам робочого діагностування та не знайшов практичного застосування.

Для наближених даних знижуються вимоги до ймовірності виявлення помилок, що дозволяє суттєво спростити логарифмічний контроль. Крім того, логарифмічний контроль дозволяє оцінювати величину помилки й у такий спосіб розрізняти суттєві та несуттєві помилки, що є перевагою методу перед традиційними методами робочого діагностування стосовно до обробки наближених даних. Тому логарифмічний контроль є перспективним для робочого діагностування ОП, що виконують наближені обчислення.

1.2 Логарифмічний контроль матричного помножувача

1.2.1 У методі логарифмічного контролю ціле число зв'язується з його контрольним кодом логарифмічною залежністю.

Контрольний код KA числа A с фіксованою точкою визначається рівним кількості розрядів значущої частини цього числа [9, 10]:

$$KA = \text{Int}(\log_2 |A|) + 1 \text{ для } A \neq 0;$$

$$KA = 0 \text{ для } A = 0.$$

Для $A \neq 0$ виконується

$$2^{KA-1} \leq |A| < 2^{KA}.$$

Дійсно, контрольному коду KA відповідає не одне число $|A|$, а діапазон чисел $2^{KA-1} \div 2^{KA} - 1$ з розрядністю KA .

Найменше значення діапазону чисел досягається при

$$\text{Int}(\log_2 |A|) = \log_2 |A|$$

для контрольного коду $KA = \log_2 |A| + 1$ числа $|A| = 2^{KA-1}$.

Верхньою границею діапазону чисел для контрольного коду KA служить найменше значення наступного діапазону чисел

$$2^{KA} \div 2^{KA+1} - 1,$$

відповідному контрольному коду $KA + 1$, тобто $|A| = 2^{KA}$.

Наприклад, для числа $A \{1 \div n\} = 1050$, що показане в табл. 1.1 у двійковій системі числення для випадку розрядності $n = 15$, його значуща частина виділена затемненням. Старший розряд значущої частини числа має номер 10.

Число A лежить у діапазоні

$$A = 2^{10} \div 2^{11} - 1.$$

Це визначає контрольний код $KA = 10$.

Таблиця 1.1 – Число A та номери його розрядів

Число A	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0
Номери розрядів	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

1.2.2 Контрольний код KA числа $A\{1 \div n\}$ обчислюється шляхом підрахунку кількості біт його значущої частини.

Обчислення здійснюються за прямим кодом числа, для якого старша незначуща частина містить нулі та передує одиниці старшого розряду молодшої значущої частини.

Для матричних ОП обчислення контрольного коду числа містить дві операції:

- заповнення значущої частини прямого коду числа одиницями;
- підрахунок кількості одиниць в отриманому коді.

Вихідними даними для виконання обчислень є код числа $A\{1 \div n\}$ та його розрядність n .

Перша операція визначає число B , що містить одиничні та нульові значення відповідно в значущій та незначущій частинах прямого коду числа A .

Старший, n -й розряд числа B визначається по формулі

$$B\{n\} = A\{n\}.$$

Молодші розряди числа B обчислюються ітераційно по формулі

$$B\{i\} = A\{i\} \vee B\{i+1\},$$

де $i = n - 1, n - 2, \dots, 2, 1$.

Розряд $B\{1\}$ коду $B\{1 \div n\}$ є ознакою нульового значення числа A та його контрольного коду, приймаючи нульове значення при $A = 0$ та одиничне значення у протилежному випадку.

Код $B\{1 \div n\}$ з заповненою одиницями значущою частиною показаний для розглянутого прикладу у табл. 1.2.

Таблиця 1.2 – Код B та номери його розрядів

Код B	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
Номери розрядів	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Код $B\{1 \div n\}$ є впорядкованим по одиницях, тобто його одиничні розряди розташовані підряд спочатку коду.

Друга операція – підрахунок кількості одиниць одержаного упорядкованого коду $B\{1 \div n\}$ – виконується ітераційно шляхом ділення кількості його одиниць на дільник

$$2^{i-1},$$

де $i = r, r-1, \dots, 2,$
 $r = E(\log_2 n) + 1.$

Код B є діленим для $i = r.$

$$R_r\{1 \div 2^r - 1\} = B\{1 \div n\}.$$

Частка від ділення кількості одиниць діленого на дільник 2^{i-1} визначається по формулі

$$q_i = R_i\{2^{i-1}\}.$$

За остачу вибирається частина коду діленого відповідно $R_i\{1 \div 2^{i-1} - 1\}$ або $R_i\{2^{i-1} + 1 \div 2^i\}$ при $q_i = 0$ та $q_i = 1.$

Одержана остача $R_r\{1 \div 2^i - 1\}$ служить далі діленим при виконанні наступної ітерації.

Операція ділення кількості одиниць в i -й ітерації описується за наступною формулою:

$$R_{i-1}\{1 \div 2^{i-1} - 1\} + q_i 2^{i-1} = R_i\{1 \div 2^i - 1\}.$$

Величини кодів діленого, частки та остачі визначаються кількістю їхніх одиниць.

Контрольний код KA числа $A\{1 \div n\}$ складається з обчислених розрядів частки:

$$KA\{q_1, q_2, \dots, q_{r-1}, q_r\}.$$

Для розглянутого прикладу $r = 4.$

У табл. 1.3 наведені значення остачі та частки операції ділення кількості одиниць для всіх номерів ітерацій $i.$

Таблиця 1.3 – Підрахунок кількості одиниць упорядкованого коду

i	$R_i \{1 \div 2^i - 1\}$														q_i		
4	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	
3	0	0	0	0	0	1	1								0		
2					0	1	1								1		
1					0												0

За результатами операції описаного алгоритму ділення кількості одиниць упорядкованого коду на основу системи числення визначається контрольний код $KA = 1010_2$ числа A .

Затемненням показано виділювані з поточних значень діленого $R_i\{1 \div 2^i - 1\}$ поточні значення остачі $R_{i-1}\{1 \div 2^{i-1} - 1\}$.

Центральним розрядом коду $R_i\{1 \div 2^i - 1\}$ є розряд частки q_i .

Контрольний код KA числа $A\{1 \div n\}$ може бути обчислений програмно переглядом прямого коду у напрямку від старших розрядів до молодшого, шукаючи перший одиничний розряд, що є старшим розрядом значущої частини.

У процесі перегляду прямого коду виконується підрахунок нульових розрядів незначущої частини прямого коду числа $A\{1 \div n\}$.

Доповнення отриманої кількості нульових розрядів до розрядності n числа $A\{1 \div n\}$ визначає його контрольний код KA для виконання логарифмічного контролю.

1.2.3 Контрольні співвідношення для перевірки арифметичних операцій виражають контрольні коди результатів через контрольні коди операндів.

Для множення чисел $A > 0$ та $B > 0$ справедливо

$$KP = KP^* \text{ або } KP = KP^* - 1,$$

де $P = A \cdot B$;

$$KP^* = KA + KB.$$

Це витікає з розгляду мінімального та максимального значень добутку при знаходженні співмножників у межах визначення контрольних кодів, тобто для діапазонів співмножників

$$A = 2^{KA-1} \div 2^{KA} - 1,$$

$$B = 2^{KB-1} \div 2^{KB} - 1$$

добуток перебуває в межах

$$2^{KA-1} \cdot 2^{KB-1} \leq P < 2^{KA} \cdot 2^{KB}.$$

За максимальним значенням добутку $P < 2^{KA+KB}$ визначається його контрольний код

$$KP = KA + KB,$$

відповідний випадку 1.

За мінімальним значенням добутку $2^{KA+KB-2}$ визначається його контрольний код

$$KP = KA + KB - 1,$$

відповідний випадку 2.

З урахуванням нульових значень співмножників контрольне співвідношення уточнюється наступною формулою:

$$KP^* = KA Z_B + KB Z_A, \quad (1.1)$$

де Z_A та Z_B – ознаки нуля співмножників A та B , що приймають значення нуля при нульових числах та одиничне значення в протилежному випадку.

Перевірка контрольного співвідношення в матричному помножувачі здійснюється в такий спосіб.

Вихідними даними є контрольні коди KA , KB та KP , що обчислюються у формувачах контрольних кодів, а також їхня розрядність r .

У контрольному блоці по контрольних кодах KA та KB , згідно з (1.1), обчислюється контрольний код KP^* .

У блоці порівняння визначається (використовуючи обернений код) різниця контрольних кодів добутку за наступною формулою:

$$C\{c_l \div c_r\} = -KP + KP^*.$$

При відповідності контрольних кодів KP^* та KP одне одному, обчислена різниця складається з усіх нулів або усіх одиниць, тобто з однакових за значенням розрядів.

Різниця $C\{c_1 \div c_r\}$ приводиться до коду з інверсним повторенням

$$D\{x_1 \div x_{r-1}, z_1 \div z_{r-1}\},$$

де $x_i = c_i,$

$$z_i = \neg c_r,$$

$$i = 1 \div r - 1,$$

Код з інверсним повторенням стискується до двохрозрядного коду контролю $E\{e_1, e_2\}$ на відомих елементах стиску.

1.3 Достовірність контролю результатів наближених обчислень

Достовірність D контролю результату оцінюється за умови виявлення суттєвої помилки або пропуску несуттєвої помилки з відбраковуванням достовірного результату по наступній формулі [4]:

$$D = P_{H\Phi}(t) (P_{BC} + P_{PH}),$$

де $P_{H\Phi}(t)$ – імовірність неправильного функціонування пристрою, що привело до помилки за період часу t ;

P_{BC} – ймовірність виявлення суттєвої помилки;

P_{PH} – ймовірність пропуску несуттєвої помилки.

Для матричних помножувачів періодом часу t виконання однієї обчислювальної операції є один такт.

Достовірність D_H контролю результату, обчисленого з помилкою під дією несправності, визначається за умови, що $P_{H\Phi}(t) = 1$.

$$D_H = P_{BC} + P_{PH}.$$

Ймовірність пропуску суттєвої помилки P_{PP} й імовірність відбраковування достовірного результату P_{OT} визначаються по наступних формулах:

$$P_{BC} = P_B P_C;$$

$$P_{\text{ПН}} = (1 - P_B) (1 - P_C),$$

де P_B – ймовірність виявлення помилки методом контролю;

P_C – ймовірність суттєвої помилки, тобто того, що помилка є суттєвою.

Ймовірність P_B може бути оцінена відношенням кількості вхідних слів, що виявляють помилку, до загальної кількості вхідних слів.

Ймовірність P_C визначається з урахуванням ряду факторів, які скорочують частку суттєвих помилок. До таких факторів відносяться наступні [13,14].

- відкидання невірних молодших розрядів результатів, що обчислюються;
- збільшення розрядності мантис за рахунок невірних розрядів;
- вирівнювання масштабуючих коефіцієнтів чисел з фіксованою точкою, що приводить до виключення молодших розрядів числа з меншим масштабуючим коефіцієнтом, які в попередніх обчислювальних операціях визначалися в складі вірних розрядів.

Помилка є суттєвою, якщо вона знижує достовірність результату, тобто спотворює вірні розряди, та несуттєвою у протилежному випадку.

До суттєвих помилок відносяться:

- помилка залишається після відкидання розрядів результату, що обчислюється;
- помилка, що перебуває в складі вірних розрядів результату;
- помилка, що зберігається в обчислювальних операціях при вирівнюванні масштабуючих коефіцієнтів.

Тоді ймовірність появи суттєвої помилки (тобто ймовірність того, що помилка, що з'явилася, є суттєвою) як складної події, що складає із трьох незалежних подій, оцінюється по формулі

$$P_C = K_1 K_2 K_3, \quad (1.2)$$

де K_1 – ймовірність появи помилки в розрядах результату, що залишаються;

K_2 – ймовірність появи помилки у вірних розрядах результату;

K_3 – ймовірність збереження помилки при вирівнюванні масштабуючих коефіцієнтів.

Ймовірність K_1 визначається при аналізі двох протилежних подій: поява помилки в розрядах результату, що залишаються або відкидаються. Ймовірності цих подій K_1 та $\bar{K}_1 = 1 - K_1$ визначаються відсотками помилок що залишаються та відкидаються разом з розрядами результату, відповідно.

Для матричних ОП ймовірність K_1 оцінюється (як далі і ймовірності K_2 та K_3), виходячи з припущення про рівномірний розподіл несправностей по апаратурі, за наступною формулою:

$$K_1 = O_{ЗАЛ} / O_{ОБЧ}, \quad K_1 \leq 1, \quad (1.3)$$

де $O_{ЗАЛ}$ – кількість устаткування, несправності, якого викликають арифметичні помилки в розрядах результату, що залишаються;

$O_{ОБЧ}$ – кількість устаткування, несправності якого викликають арифметичні помилки в усіх розрядах результату, що обчислюються.

Ймовірність K_2 визначається при аналізі двох протилежних подій: поява помилки у вірних або невірних розрядах результату. Ймовірності цих подій K_2 та $\bar{K}_2 = 1 - K_2$ визначаються відсотком помилок у вірних і невірних розрядах, відповідно.

Для матричних ОП ймовірність K_2 оцінюється з урахуванням рівномірного розподілу несправностей по апаратурі за наступною формулою:

$$K_2 = O_{ТВ} / O_{ОСТ}, \quad K_2 \leq 1,$$

де $O_{ТВ}$ – кількість устаткування, несправності якого викликають арифметичні помилки у вірних розрядах результату.

Наприклад, закладені у форматах із плаваючою точкою персональних ЕОМ можливості по збільшенню розрядності мантис в 2,8 разів визначають, зокрема, ймовірність $K_2 \leq 0,36$.

Ймовірність K_3 визначається при аналізі двох протилежних подій: збереження або втрата помилки при вирівнюванні масштабуючих коефіцієнтів. Ймовірності K_3 та $\bar{K}_3 = 1 - K_3$ цих подій визначаються відсотками помилок, що зберігаються та губляться внаслідок надлишкової точності, відповідно Тому при відсутності цієї операції $K_3 = 1$, а в протилежному випадку $K_3 < 1$.

Ймовірність K_3 можна оцінити, з огляду на те, що при зсуві одного з двох чисел на d позицій відкидається d з $2n$ розрядів операндів. Отже, із всіх проміжних результатів, що передують зсунутому даному, у середньому виключається $0,5 d/n$ вірних розрядів, а також усувається вплив відповідної частини помилок, що викликаються несправностями вузлів, які передують зсувачу при вирівнюванні масштабуючих коефіцієнтів.

Тоді ймовірність K_3 визначається по формулі:

$$K_3 = 1 - 0,5 \delta O_C d / n,$$

де $\delta O_C = O_C / O_0$;

O_C – кількість устаткування вузлів ОП, що передують даному зсувачу чисел;

O_0 – загальна кількість устаткування ОП.

Вихідні дані, що є безпосередньо результатами проведених вимірів, можуть мати однакові масштабуючі коефіцієнти, що визначає величину зсуву $d = 0$ та ймовірність $K_3 = 1$.

Кожна арифметична операція, що потребує вирівнювання масштабуючих коефіцієнтів може суттєво збільшити величину зсуву d .

При рівноймовірних значеннях величини зсуву, її середнє значення становить половину розрядності числа, тобто $d = n/2$, а формула для ймовірності приймає наступний вигляд $K_3 = 1 - 0,25 \delta O_C$.

Для декількох операцій вирівнювання масштабуючих коефіцієнтів, виконуваних незалежно в даному ОП, ймовірність K_3 визначається як добуток ймовірностей, знайдених для кожної із цих операцій.

Описані оцінки ймовірностей, що визначають достовірність логарифмічного контролю наближених результатів, не враховують характер несправностей матричних пристроїв.

Для уточнення ймовірності виявлення помилки P_B та появи суттєвої помилки P_C необхідно виконати моделювання обчислювальних процесів, що протікають в логарифмічному контролі під впливом кратних несправностей матричних пристроїв.

1.4 Висновки

Достовірність результатів обчислень є важливою характеристикою сучасних ОП й зокрема матричних помножувачів, які відіграють важливу роль в обробці наближених даних у форматах з фіксованою точкою.

Достовірність результатів обчислень оцінюється з використанням методів робочого діагностування.

Основним методом робочого діагностування арифметичних пристроїв є контроль за модулем. Однак цей метод недостатньо ефективний при обробці наближених даних, оскільки не розрізняє суттєві та несуттєві помилки.

Логарифмічний контроль оцінює величину помилки, виявляючи суттєві помилки в старших розрядах чисел з більшою ймовірністю в порівнянні з несуттєвими помилками, що виникають у молодших розрядах наближеного результату.

Одним з основних показників методу контролю є достовірність контролю результатів, що обчислюються. Для оцінки цього показника необхідно досліджувати ймовірності виявлення та пропуску суттєвих і несуттєвих помилок при впливі кратних несправностей ОП, що діагностується.

Сучасні системи моделювання не забезпечують гнучкого завдання несправностей цифрових схем та одержання ймовірностей виявлення та пропуску відмов за суттєвими та несуттєвими помилками при повному або випадковому переборі вхідних слів.

Вирішення цієї задачі для матричного помножувача з фіксованою точкою вимагає розробки спеціальної програмної моделі логарифмічного контролю та її дослідження в умовах дії кратних несправностей пристрою на довільних вхідних послідовностях даних.

2 ЗАВДАННЯ НА ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ

2.1 Найменування та область застосування

2.1.1 Магістерська робота виконується за темою «Дослідження логарифмічного контролю помножувача під дією кратних несправностей», яка присвячена дослідженню методу логарифмічного контролю стосовно до робочого діагностування матричного помножувача з фіксованою точкою, що виконує обробку наближених даних в умовах дії кратних несправностей матричних пристроїв.

2.1.2 Областю застосування розробки є методи та засоби робочого діагностування сучасних обчислювальних систем, до яких пред'являються високі вимоги з продуктивності, діапазону подання чисел, а також достовірності обчислюваних результатів.

Матричний помножувач з фіксованою точкою широко застосовується для обробки наближених даних в природній формі подання. Пристрій матричного помножувача з фіксованою точкою використовується в арифметичних блоках матричних процесорів, на ділянках конвеєрів векторних процесорів, а також у вигляді окремих функціональних пристроїв СуперЕОМ та систем наддовгих команд.

Предметом досліджень є достовірність логарифмічного контролю результатів наближеної обробки даних, що обчислюються в матричному помножувачі з фіксованою точкою під дією кратних несправностей.

2.2 Призначення досліджень

2.2.1 Дані дослідження призначені для вивчення логарифмічного контролю результатів обробки наближених даних в матричному помножувачі з фіксованою точкою, що має місце в умовах дії кратних несправностей матричних пристроїв.

Аналізується ймовірність появи помилки, що викликається кратною несправністю матричного помножувача, а також їхній розподіл на пропущені та виявлені методом логарифмічного контролю, позитивні та негативні, суттєві та несуттєві помилки.

За характерну несправність матричного помножувача слід розглянути одиночні «замикання» між двома точками в межах окремих операційних елементів пристрою.

2.2.2 Вивчається ймовірність виявлення та пропуску відмов за суттєвими та несуттєвими помилками, які викликані кратними несправностями матричного пристрою, що дозволяє оцінювати достовірність логарифмічного контролю результатів, що обчислюються.

2.3 Організація досліджень

2.3.1 Для проведення досліджень розробляється програмна модель логарифмічного контролю матричного помножувача з фіксованою точкою, що працює в режимах правильного функціонування пристрою, а також в умовах дії кратних несправностей.

Операндами матричного помножувача є n -розрядні двійкові числа, а результатом – $2n$ -розрядний добуток, що містить старші вірні та молодші невірні розряди. Кількість невірних розрядів не може бути меншою за довжину суттєвої частини найбільшого з операндів.

Структурно, програмна модель матричного помножувача з повною разрядністю повинна включати наступні складові:

- демонстраційні елементи;
- елементи керування.

2.3.1 Демонстраційні елементи програмної моделі повинні включати до свого складу наступне:

- операнди операції множення, тобто співмножники, а також правильний результат та результат, що обчислюється під дією кратної несправності, з виділенням їх значущих частин;
- структурну схему логарифмічного контролю з конкретними значеннями контрольних кодів операндів та результатів;
- результати моделювання логарифмічного контролю матричного помножувача з фіксованою точкою, що включають значення ймовірностей виявлення та пропуску позитивних та негативних, суттєвих та несуттєвих помилок.

2.3.2 До елементів керування слід віднести переміщений покажчик розподілу між вірними та невірними розрядами обчисленого в програмній моделі результату.

Розв'язання даного завдання включає:

- одержання вихідних даних від програмної моделі матричного помножувача з фіксованою точкою відповідно до встановлених режимів роботи;
- розробку алгоритмів реалізації функцій логарифмічного контролю матричного помножувача з фіксованою точкою з формування контрольних кодів операндів та результатів;
- розробку алгоритмів виконання функцій логарифмічного контролю з перевірки контрольного співвідношення, що зв'язує контрольні коди операндів та результату;
- розробку алгоритмів для оцінки ймовірностей виявлення та пропуску відмов за позитивними та негативними, суттєвими та несуттєвими помилками, що викликані кратними несправностями матричного помножувача з фіксованою точкою.

2.4 Визначення задачі

2.4.1 Ставиться задача на проведення досліджень методу логарифмічного контролю для робочого діагностування матричного помножувача з фіксованою точкою, що виконує обробку наближених жених під дією кратних несправностей.

Для цього необхідно виконати розробку алгоритмів побудови програмної моделі логарифмічного контролю матричного помножувача, що дозволяє вивчати можливості методу логарифмічного контролю при виникненні кратних несправностей матричних пристроїв. Дослідженню підлягають ймовірності виявлення суттєвих і несуттєвих для достовірності результату помилок, тобто таких помилок, які відповідно знижують і не знижують задану кількість старших вірних розрядів наближеного результату.

2.4.2 До програмної моделі пред'являються вимоги зручного використання, наочності, функціональності та достовірності, тобто наближеності до реальності.

Зручність у використанні повинна ґрунтуватися на простоті та варіантності завдання режимів роботи, вихідних даних, а також в одержанні результатів моделювання логарифмічного контролю матричного помножувача з фіксованою точкою.

Наочність програмної моделі варто забезпечити образотворчими мовними засобами, відображаючи структуру логарифмічного контролю, а також інформативною індикацією вихідних даних та одержуваних результатів моделювання.

Функціонально, модель матричного помножувача з фіксованою точкою повинна забезпечувати:

- виконання контрольних обчислень методу логарифмічного контролю матричного помножувача;
- показ структурної схеми логарифмічного контролю матричного помножувача;
- індикацію значень контрольних кодів на виходах блоків структурної схеми логарифмічного контролю матричного помножувача;

- установку порогового значення, що визначає розподіл наближеного результату множення на вірні та невірні розряди.
- індикацію виявлення та пропуску позитивних та негативних суттєвих та несуттєвих помилок наближеного результату, що обчислюється в матричному помножувачі з фіксованою точкою.

Реальність програмної моделі повинна забезпечуватися правильністю опису функцій, реалізованих логарифмічним контролем в матричному помножувачі та моделлю самого помножувача, включаючи використовувану в ньому модель кратної несправності.

2.4.3 З використанням розробленої програмної моделі повинні бути виконані діагностичні експерименти, які спрямовуються на оцінку ймовірностей пропуску та виявлення позитивних та негативних суттєвих та несуттєвих помилок, що викликаються кратними несправностями матричного помножувача з плаваючою точкою.

За результатами проведеного програмного моделювання повинна бути визначена достовірність методу логарифмічного контролю наближених результатів, що обчислюються в матричному помножувачі з фіксованою точкою.

Крім того, повинна бути виконана порівняльна оцінка достовірності логарифмічного контролю наближених результатів з достовірністю контролю за модулем наближених результатів, що обчислюються в матричному помножувачі з фіксованою точкою під дією кратних несправностей.

3 РОЗРОБКА ТА ДОСЛІДЖЕННЯ ПРОГРАМНОЇ МОДЕЛІ ЛОГАРИФМІЧНОГО КОНТРОЛЮ МАТРИЧНОГО ПОМНОЖУВАЧА ПІД ДІЄЮ КРАТНИХ НЕСПРАВНОСТЕЙ

3.1 Структура програмної моделі логарифмічного контролю

3.1.1 Програмна модель логарифмічного контролю матричного помножувача розроблена на основі мовних засобів програмного середовища DELFI.

Структура програмної моделі логарифмічного контролю матричного помножувача з фіксованою точкою побудована з використанням двох типів складових:

- демонстраційних елементів;
- елементів керування.

Демонстраційні елементи програмної моделі забезпечують відображення наступного:

- розрядності співмножників матричного помножувача з фіксованою точкою;
- режимів роботи програмної моделі;
- завдання конкретного типу та місця несправності «замикання» в операційних елементах матричного помножувача з фіксованою точкою або довільного їх характеру;
- співмножників з фіксованою точкою, що приймають участь у виконанні операції множення;
- результатів виконання операції множення з виділенням їх значущих частин: правильного повного добутку та повного добутку, що обчислюється під дією кратної несправності пристрою;
- структурної схеми логарифмічного контролю з обчисленими значеннями контрольних кодів операндів та результатів;
-

- результатів моделювання логарифмічного контролю матричного помножувача з фіксованою точкою, що включають значення ймовірностей виявлення та пропуску відмов за позитивними та негативними, суттєвими та несуттєвими помилками, що викликані кратними несправностями пристрою;

Елементи керування програмної моделі забезпечують завдання режимів роботи програмної моделі та перегляд роботи окремих блоків схеми логарифмічного контролю й тому до їхнього складу залучено наступні:

- панелі, що зображують блоки структурної схеми логарифмічного контролю матричного помножувача з фіксованою точкою;
- панель, що виконує розподіл результату, що обчислюється, на вірні та невірні розряди.

3.2 Функціонування моделі логарифмічного контролю

Модель логарифмічного контролю матричного помножувача з фіксованою точкою забезпечує:

- завдання режимів роботи програмної моделі з визначення операндів, а також справного або несправного стану матричного помножувача з фіксованою точкою;
- завдання операндів шляхом визначення окремих конкретних або випадкових значень для налагодження програмної моделі, а також послідовностей значень з повним перебором або довільного характеру для проведення досліджень логарифмічного контролю матричного помножувача з фіксованою точкою;
- завдання місця несправності типу «закоротка» між двома точками вибраного або довільного операційного елемента матричного помножувача з фіксованою точкою;
- завдання виду несправності типу «закоротка» між двома заданими або довільними точками вибраного або довільного операційного елемента матричного помножувача з фіксованою точкою;

- формування контрольних кодів операндів та результату, що обчислюють їх логарифмічну оцінку;
- перевірку контрольних співвідношень між контрольними кодами операндів та результату для логарифмічного контролю матричного помножувача з фіксованою точкою;
- перерозподіл вірних та невірних розрядів результату, що обчислюється в матричному помножувачі з фіксованою точкою під дією кратних несправностей пристрою;
- показ структурної схеми логарифмічного контролю матричного помножувача з фіксованою точкою;
- індикацію значень контрольних кодів на виходах операційних елементів структурної схеми логарифмічного контролю матричного помножувача з фіксованою точкою;
- індикацію виявлення суттєвих та несуттєвих помилок, що викликаються в результаті під дією кратних несправностей матричного помножувача з фіксованою точкою;
- підрахунок та індикацію кількості виявлених та невиявлених логарифмічним контролем позитивних та негативних помилок;
- підрахунок та індикацію відсотку виявлених та невиявлених логарифмічним контролем позитивних та негативних помилок;

3.3 План проведення досліджень логарифмічного контролю матричного помножувача під дією кратних несправностей

3.3.1 Необхідно переконатися в правильному функціонуванні моделі логарифмічного контролю матричного помножувача шляхом її перевірки на тестових прикладах, досліджуючи

- режими початкової установки програмної моделі;
- зовнішній вигляд демонстраційних елементів програмної моделі;
- роботу програмної моделі в її різних режимах;

- достовірність результатів моделювання логарифмічного контролю матричного помножувача з фіксованою точкою.

У початковій установці програмної моделі логарифмічного контролю матричного помножувача з фіксованою точкою варто перевірити наступне:

- правильність завдання режимів роботи програмної моделі логарифмічного контролю;
- правильність режимів визначення конкретних та довільних значень співмножників та їх послідовностей;
- правильність значень вихідних даних;
- правильність режимів завдання несправності матричного помножувача з фіксованою точкою;
- правильність місця та виду завдання конкретних та довільних значень несправності пристрою;
- правильність розподілу результатів обчислюваних результатів на вірні та невірні розряди;
- правильність контролю завдання співмножників при уведенні десяткового числа в частині можливого перевищення верхньої границі припустимого діапазону.

Демонстраційні елементи програмної моделі необхідно верифкувати, відслідковуючи наступне:

- правильність відображення структури матричного помножувача для випадків різної розрядності пристрою;
- правильність формування матриці операційних елементів з урахуванням різної розрядності пристрою;
- структури операційних елементів для першого та наступного рядків матриці пристрою;
- можливість правильного показу всіх припустимих видів несправності й виключення варіантів закороткі однойменних висновків, а також рівнів логічного нуля й одиниці.

У функціонуванні програмної моделі логарифмічного контролю необхідно перевірити:

- правильність обчислень, виконуваних у відсутності несправності на конкретних значеннях співмножників;
- правильність виконання обчислень при повному переборі значень співмножників;
- правильність виконання обчислень на послідовності вхідних даних, що задає випадково значення співмножників;
- для різних значень розрядності матричного помножувача з фіксованою точкою.
- правильність обчислень, виконуваних у моделі при уведеній несправності конкретного виду й у конкретному місці;
- правильність обчислень, що виконуються на заданих значеннях співмножників.

Достовірність одержуваних результатів логарифмічного контролю перевіряється

- порівнянням операндів та результатів з їхніми логарифмічними оцінками, тобто контрольними кодами, на окремих вхідних словах у режимі справного пристрою матричного помножувача;
- порівнянням результатів з їхніми контрольними кодами на окремих вхідних словах при виявленні помилок у режимі несправного пристрою матричного помножувача;
- вибірковою перевіркою роботи окремих операційних елементів матриці помножувача при відсутності та появи помилки під дією певної несправності;
- оцінкою величини помилки залежно від місця та виду несправності, що її викликає;
- оцінкою відсотків виявлення та пропуску суттєвих і несуттєвих помилок на послідовностях вхідних слів при завданні несправностей для генерації тільки суттєвих або несуттєвих помилок;

- візуальною перевіркою правильності функціонування демонстраційних елементів програмної моделі логарифмічного контролю та її елементів керування у різних режимах.

3.4 Основні алгоритми програмної моделі логарифмічного контролю матричного помножувача

3.4.1 Для побудови програмної моделі логарифмічного контролю помножувача розроблена низка алгоритмів обчислення контрольних кодів даних, які заносяться у відповідні поля форми, описуваної язовими засобами DELFI.

До них відносяться: контрольні коди множеного A та множника B , контрольні коди повного результату, що отримані по результату та обчислені по операндах, а також дані про помилки: величині, кількості помилок – виявленій і невиявленій, суттєвій і несуттєвих, а також їхній частці в загальній кількості результатів, що обчислюють.

3.4.2 На рис. 3.1 показана схема алгоритму формування контрольного коду операнда – множеного або множника – на прикладі числа A .

Алгоритм працює в такий спосіб.

Блок 2 виконує уведення операнда A у вигляді числа aa .

Блок 3 задає початкове значення контрольного коду ka операнда A .

Блок 4 виконує усікання значущої частини операнда на ka біт.

Блок 5 перевіряє усічену значущу частину операнда на позитивне значення, тобто що ця частина при усіканні не звернулася в нуль.

Блок 6 здійснює підрахунок кількості розрядів значущої частини операнда, збільшуючи значення контрольного коду на одиницю.

Блок 7 задає початкове значення ознаки нуля, що відповідає нульовому операнду.

Блок 8 виконує порівняння операнда з нулем. При позитивному значенні операнда блок 9 установлює одиничне значення ознаки нуля.

Блок 10 виводить контрольний код операнда та його ознаку нуля.

Алгоритм закінчено.

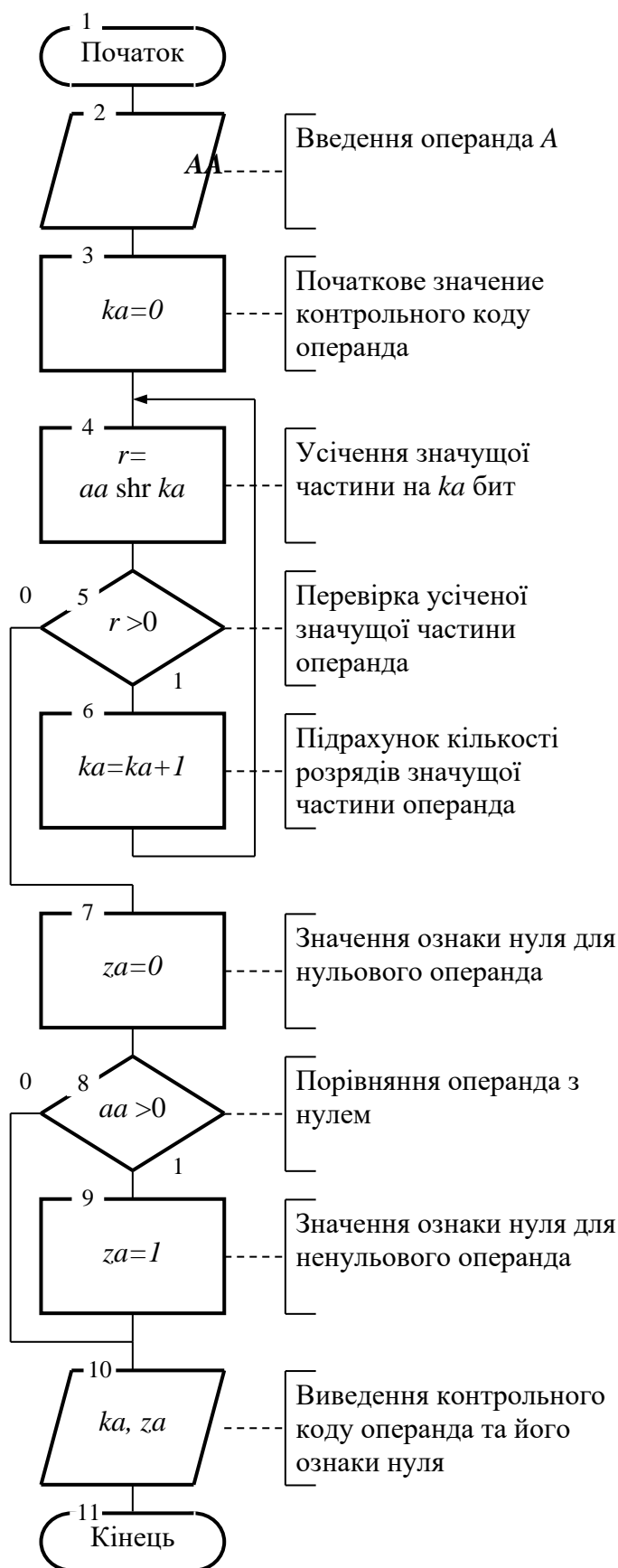


Рисунок 3.1 – Схема алгоритму формування контрольного коду операнда

3.4.3 На рис. 3.2 показана схема алгоритму перевірки контрольного співвідношення, що зв'язує контрольні коди операндів та результату при виконанні логарифмічного контролю в матричному помножувачі з фіксованою точкою.

Алгоритм перевірки контрольного співвідношення за логарифмами працює в такий спосіб.

Блок 2 вводить вихідні дані до алгоритму, до яких відносяться контрольні коди та ознаки нуля операндів – множеного та множника, а також кількість виконаних операцій множення й кількість виявлених при цьому помилок результату.

Блок 3 обчислює контрольний код результату по контрольних кодах операндів та ознакам їхнього нуля відповідно до описаного вище контрольного співвідношення.

Блок 4 задає передбачуване значення ознаки помилки для результату операції множення.

Блок 5 перевіряє відповідність обчислених контрольних кодів результату, порівнюючи їх.

Блок 6 перевіряє збігання обчислених кодів результату з урахуванням їх можливого відхилення на одиницю.

Блок 7 установлює нульове значення ознаки помилки при відповідності порівнюваних контрольних кодів результату.

Блок 8 при порушенні хоча б однієї з двох умов відповідності порівнюваних за логарифмічними оцінками контрольних кодів результату видає повідомлення про помилку.

Блок 9 збільшує на одиницю кількість виявлених логарифмічним контролем помилок результату.

Блок 10 збільшує на одиницю кількість операцій множення, що виконуються в матричному помножувачі з фіксованою точкою над співмножниками під дією кратних несправностей, що вводяться в несправному стані пристрою.

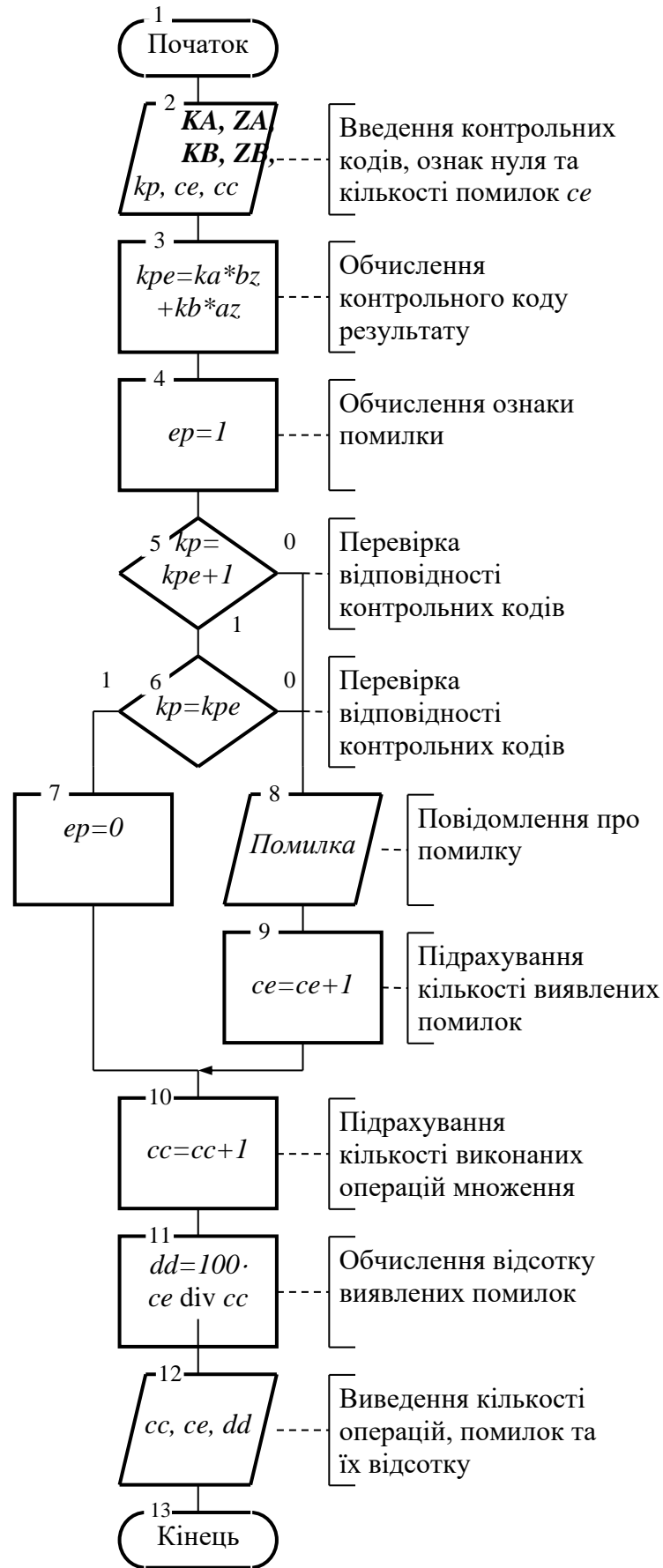


Рисунок 3.2 – Схема алгоритму перевірки контрольного співвідношення

Блок 11 обчислює частку виявлених помилок як відношення кількості виявлених помилок до кількості операцій множення, що виконуються в матричному помножувачі з фіксованою точкою.

Блок 12 виводить кількість виконаних операцій, а також кількість та відсоток помилок, що виявлені логарифмічним контролем в матричному помножувачі з фіксованою точкою.

Алгоритм закінчено.

3.4.4 На рис. 3.3 показана схема алгоритму оцінки ймовірностей виявлення суттєвих помилок округленого результату в матричному помножувачі з фіксованою точкою.

Алгоритм працює в такий спосіб.

Блок 2 вводить вихідні дані до алгоритму: правильне pp та обчислене під дією несправності pe результати множення, ознаку помилки ep , кількість виконаних операцій cs , кількість виявлених de та невиявлених ne суттєвих помилок, а також перевищення h кількості невірних розрядів у порівнянні з розрядністю операндів.

Блок 3 визначає величину помилки, порівнюючи правильний результат з результатом, поліченим під дією несправності матричного помножувача з фіксованою точкою.

Блок 4 обчислює припустиму похибку обчислення результату, використовуючи задане перевищення h кількості невірних розрядів у порівнянні з розрядністю операндов.

Блок 5 перевіряє умову, за якою помилка визначається суттєвою для достовірності результатів, що обчислюються в матричному помножувачі з фіксованою точкою під дією його кратних несправностей.

Блок 6 перевіряє факт виявлення помилки логарифмічним контролем по ознаці помилки.

Блок 7 збільшує на одиницю кількість виявлених суттєвих помилок при виконанні умови в блоці 5 та невиконанні умови в блоці 6.

Блок 8 збільшує на одиницю кількість невиявлених суттєвих помилок при виконанні обох умов у блоках 5 та 6.

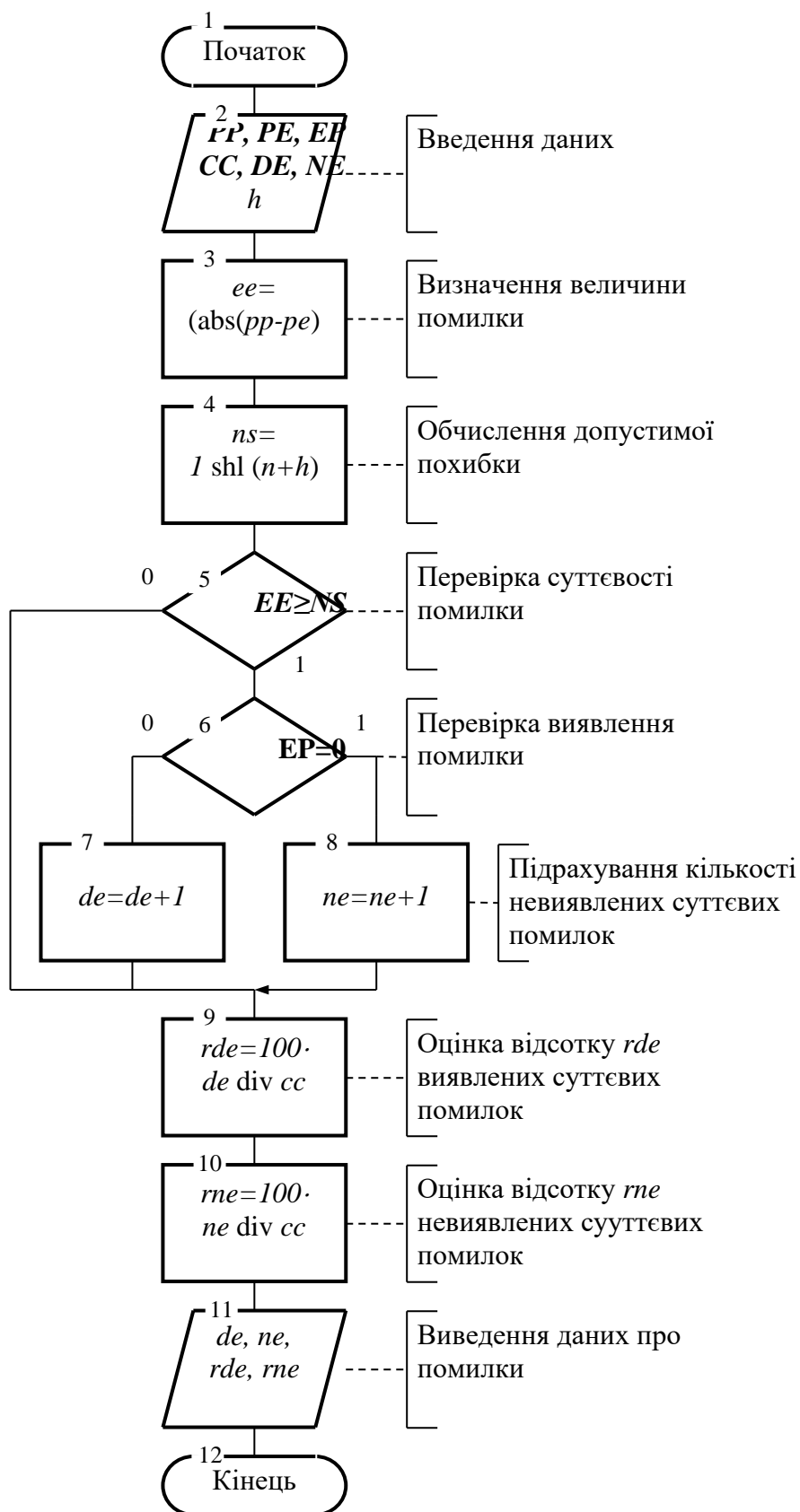


Рисунок 3.3 – Схема алгоритму оцінки відсотку виявлення суттєвих помилок округленого результату

Блок 9 оцінює відсоток *rde* суттєвих помилок, що виявляються логарифмічним контролем в обчислюваному результаті матричного помножувача з фіксованою точкою.

Блок 10 оцінює відсоток *rne* суттєвих помилок, що не виявляються логарифмічним контролем в обчислюваному результаті матричного помножувача з фіксованою точкою.

Блок 11 виводить дані про помилки: кількість та відсоток виявлених та невиявлених суттєвих помилок результату.

Алгоритм закінчено.

3.5 Висновки

Логарифмічний контроль матричного помножувача з фіксованою точкою може бути вивчений в умовах дії кратних несправностей шляхом побудови його програмної моделі, що відповідає вимогам інформативності, функціональності та достовірності.

Програмна модель логарифмічного контролю призначена для одержання достовірної оцінки значень ймовірності виявлення та пропуску відмов за позитивними й негативними, суттєвими та несуттєвими помилками на робочих послідовностях вхідних слів під дією кратних несправностей матричного помножувача з фіксованою точкою.

Достовірність оцінки повинна забезпечуватися достовірністю програмної моделі матричного помножувача з фіксованою точкою та наближенням до реальності моделі кратних несправностей пристрою, а також коректною постановкою діагностичних експериментів.

Достовірність програмної моделі логарифмічного контролю матричного помножувача з фіксованою точкою може бути забезпечена при багатобічній перевірці її функціонування, що вимагає введення режимів, які дозволяють здійснити тестування й налагодження моделі.

До режимів налагодження програмної моделі віднесено логарифмічний контроль окремих операцій на заданих значеннях співмножників для конкретного виду й місця несправності, в умовах справного й несправного функціонування матричного помножувача.

Демонстраційні елементи програмної моделі роблять перевірки її достовірності наочними та оперативними.

Достовірна оцінка ймовірностей виявлення та пропуску помилок повинна бути забезпечена дослідженням логарифмічного контролю матричного помножувача з фіксованою точкою на повній множині вхідних слів або у випадку великої розрядності пристрою на їхній випадковій послідовності до одержання стійких результатів.

До основних алгоритмів програмної моделі логарифмічного контролю відносяться алгоритми формування контрольного коду операнда, перевірки контрольного співвідношення операції множення та оцінки ймовірностей виявлення суттєвих помилок.

Алгоритм формування контрольного коду операнда – множеного або множника, показаний на прикладі числа A , забезпечує підрахунок кількості біт його значущої частини.

Алгоритм перевірки контрольного співвідношення, що зв'язує контрольні коди операндів та результату, визначає контрольний код результату по контрольних кодах операндів й ознакам їхнього нуля, а також порівнює його з контрольним кодом, отриманим безпосередньо за результатом, що отримується при моделюванні пристрою.

Алгоритм оцінки ймовірностей виявлення суттєвих помилок округленого результату в матричному помножувачі з фіксованою точкою виконується з урахуванням розподілу розрядів на вірні й невірні.

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ЛОГАРИФМІЧНОГО КОНТРОЛЮ МАТРИЧНОГО ПОМНОЖУВАЧА ПІД ДІЄЮ КРАТНИХ НЕСПРАВНОСТЕЙ

4.1 Програмна модель логарифмічного контролю

4.1.1 Керівництво програміста.

Призначення. Програмний продукт (ПП) призначений для моделювання процесу логарифмічного контролю обчислень у матричному помножувачі з фіксованою точкою в умовах дії його кратних несправностей типу «замикання».

Склад. Програмний продукт написаний на Delphi, використовує одну форму та містить 17 процедур.

Вихідні дані. Розрядність операндів n задається у вигляді значення константи. Інші дані визначаються в процесі роботи ПП.

Результати виконання програмного продукту. Результати моделювання зберігаються на екрані.

4.1.2 Керівництво користувача

4.1.2.1 Установка та запуск. Програмний продукт поставляється у вигляді завантажувального модуля для заданої розрядності матричного помножувача й запускається на виконання по команді ENTER.

4.1.2.2 Користувальницький інтерфейс.

При натисканні кнопки «вхід» на екрані з'являється панель опису об'єкта логарифмічного контролю – матричного помножувача з фіксованою точкою, що включає:

- меню режимів завдання операндів та стану пристрою;
- поля та кнопки для завдання й прийому операндів десятковим числом і двійковим кодом;
- структуру матричного помножувача з фіксованою точкою заданої розрядності;

- поля та кнопки завдання місця та виду несправності з можливістю їхнього визначення випадковим образом;
- панель зміни кількості вірних розрядів результату.

На рис. 4.1 показана панель опису матричного помножувача для розрядності операндів $n = 10$.

The screenshot shows a software interface for configuring a matrix multiplier. The window is titled "Form1" and has three tabs: "Operand A", "Operand B", and "State". The "State" tab is active, showing a "Word size 8" label, an "Exit" button, a "Start" button, and a "Multiplier" button. Two input fields for "A" (98) and "B" (97) are present, along with a green "1 - 2" indicator. The main area contains a grid of 10x10 cells for matrix elements, with a "Faulty element" section and "Random" buttons for configuration. At the bottom, there are two rows of 10 bits each, labeled "Vm" and "V".

Рисунок 4.1 – Панель представлення матричного помножувача

При щиглику миші по панелі «Логарифмічний контроль» графічне зображення матричного помножувача змінюється на панель логарифмічного контролю, що містить наступний набір елементів:

- поля операндів, для яких виділена кольорами значуща частина числа;
- поле результату множення, для якого виділена кольорами реальна й обчислена по операндах значуща частина числа;

- схему логарифмічного контролю із вказівкою значень операндів, результатів та їхніх контрольних кодів, а також коду контролю обчислюваного результату;
- поле невірних розрядів результату, що обчислюється матричним помножувачем з фіксованою точкою;
- таблицю результатів моделювання із вказівкою ймовірностей виявлення та пропуску суттєвих та несуттєвих помилок, а також ймовірностей появи суттєвих та несуттєвих помилок та виявлення та пропуску помилок.

На рис. 4.2 показана панель моделювання логарифмічного контролю матричного помножувача з фіксованою точкою для розрядності операндів $n = 11$.

The screenshot shows a software window titled 'Form1' with the following components:

- Operands:** Operand A = 184, Operand B = 139.
- Buttons:** 'Finish', 'Log check', and 'Log check'.
- Binary Representations:**
 - A: 1 1 0 1 1 1 0 0
 - B: 0 0 1 0 1 0 1 0
 - Vm: 0 0 1 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0
 - V: 0 0 1 0 0 1 0 0 0 1 0 1 1 0 0 0 0
- Error Statistics Table:**

	ERROR		Total
	Essential	Inessential	
R.a = 54,2%			
R.e = 7,5%			
Detection	3,0%	0,1%	3,1%
0 - 1	2,0%	0,0%	2,0%
1 - 0	1,0%	0,1%	1,1%
Skipping	19,1%	19,7%	38,8%
0 - 1	8,7%	8,7%	17,4%
1 - 0	10,4%	11,0%	21,4%
Total	22,1%	19,8%	41,9%
0 - 1	10,7%	8,7%	19,4%
1 - 0	11,4%	11,1%	22,5%
- Logic Diagram:** Shows inputs A (220), B (421), and V (93044) passing through fault blocks F_{KA}, F_{KB}, and F_{KV} to a control block CB. The output of CB is KV* (14), which is then processed by block BC to produce K_V (0).

Рисунок 4.2 – Панель моделювання логарифмічного контролю

Процес моделювання починається з організації основних обчислень на панелі матричного помножувача.

Установлюються режими завдання операндів з використанням для кожного з них меню, з якого вибирається конкретне або довільне значення, а також повний перебір або послідовність довільних значень.

Стан матричного помножувача з фіксованою точкою встановлюється за допомогою відповідного меню в режим справної або несправної (за одною відмовою) роботи.

У режимі конкретного значення операнд може бути заданий десятковим числом або двійковим кодом з використанням також можливостей інвертування окремих розрядів.

У режимі несправної роботи матричного помножувача задається вид та місце несправності типу «замикання» або встановлюються режими їхнього довільного вибору.

Поле вірних розрядів може бути збільшене на значення, що перебуває в межах розрядності операнда переміщенням лівої границі поля.

Обчислення в програмній моделі логарифмічного контролю починаються з натискання кнопки «Старт».

У режимах завдання одного набору операндів можливо простежити обчислення результату розкриваючи по черзі низку операційних елементів матричного помножувача, а також переглядаючи значення їхніх внутрішніх точок.

За результатами моделювання заповнюються поля правильного добутку та добутку, обчисленого під впливом несправності, у випадку вибору режиму несправності. Крім того, представляються десяткові значення результатів і величини помилок.

У процесі перебору значень виводяться також кількість помилок й їхня частка в загальній кількості виконуваних операцій, а у вигляді таблиці – частка суттєвих і несуттєвих помилок.

Щиглик миші по панелі «Логарифмічний контроль» міняє її назву на «Помножувач» і встановлює графічне зображення, що відповідає виконуваному процесу контролю, фіксуючи поточний стан операндів, що обчислюють результатів, схеми контролю й таблиці результатів моделювання.

Щиглик миші по перейменованій панелі «Помножувач» повертає вихідне графічне зображення, що відповідає поданню матричного помножувача, а також вихідна назва панелі. Перехід до демонстрації основних обчислень і логарифмічного контролю може здійснюватися багаторазово для супроводу процесу моделювання й вивчення їхніх остаточних результатів. Перебір значень операндів може бути зупинений повторним натисканням кнопки «Старт», що також приводить до її перейменування на назву «Фініш».

На рис. 4.3 показана панель моделювання логарифмічного контролю в момент виявлення помилки виду $0 \rightarrow 1$.

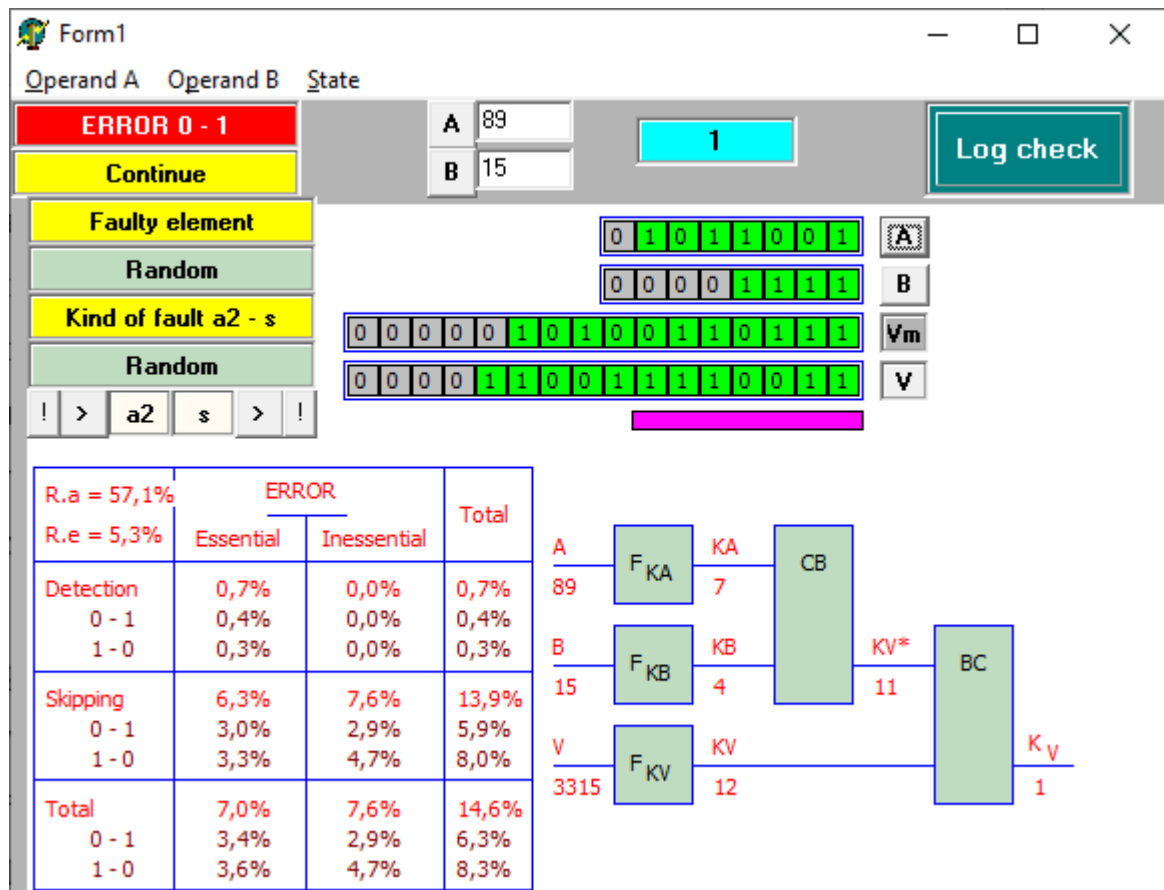


Рисунок 4.3 – Виявлення помилки виду $0 \rightarrow 1$.

У розряді 21 обчисленого результату відбулася помилка виду $0 \rightarrow 1$. Ця помилка виявлена, оскільки контрольний код $KP = 21$ помилкового результату Pe не відповідає контрольному коду $KP^* = 19$ правильного результату P , обчисленого як сума контрольних кодів $KA = 11$ та $KB = 8$ операндів.

На рис. 4.4 показана панель моделювання логарифмічного контролю в момент виявлення помилки виду $1 \rightarrow 0$.

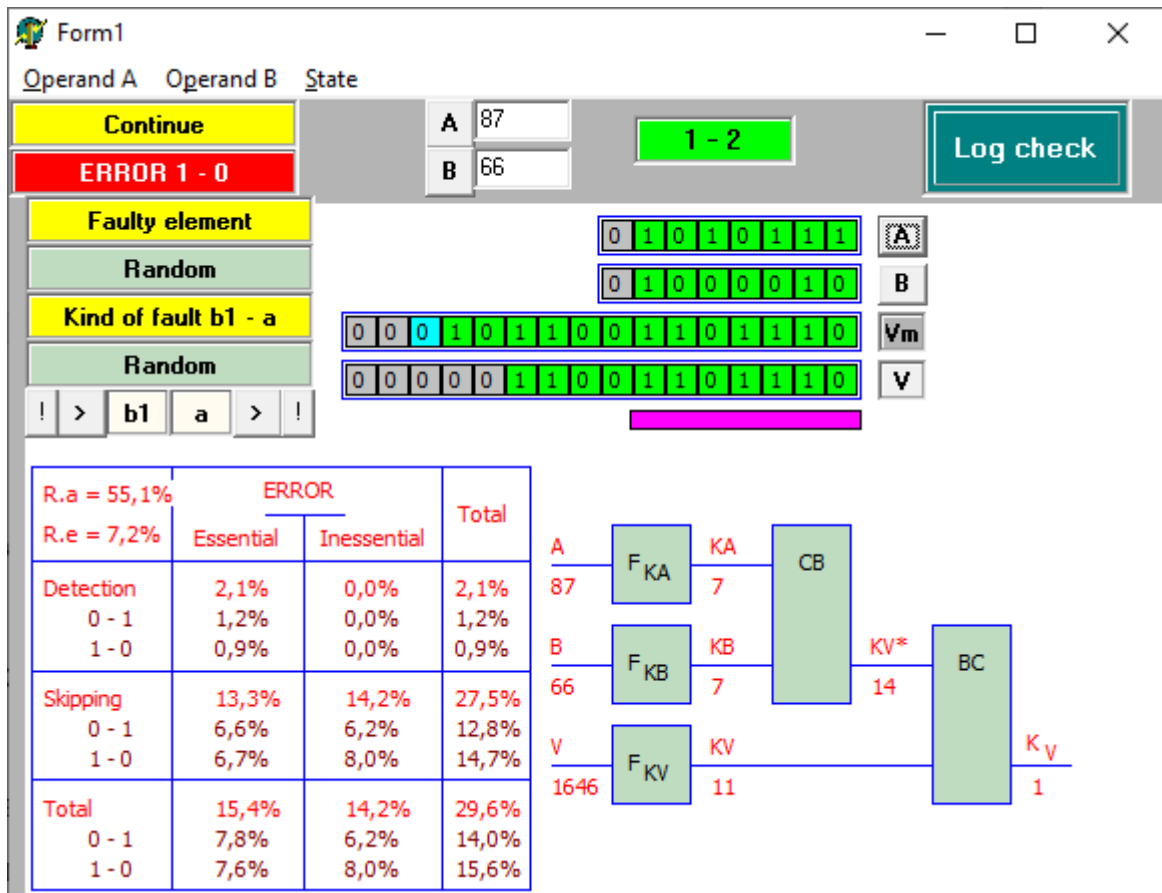


Рисунок 4.4 – Виявлення помилки виду $1 \rightarrow 0$.

У розряді 15 обчисленого результату відбулася помилка виду $1 \rightarrow 0$. Ця помилка виявлена, оскільки контрольний код $KP = 14$ помилкового результату Pe не відповідає контрольному коду $KP^* = 16$ правильного результату P , обчисленого як сума контрольних кодів $KA = 6$ й $KB = 10$ операндів.

4.1.2.3 Обробка екстрених ситуацій. Передбачено виведення повідомлень про помилки:

- порушення верхньої границі діапазону операндов, заданих цілим десятковим числом;
- уведення неправильного номера несправного осередку матричного помножувача.

Діапазон операндів обмежений зверху числом $2^n - 1$.

Номери операційних елементів матриці матричного помножувача складені з номерів рядка та стовпця, які змінюються в діапазонах від 1 до n і від 2 до n , відповідно.

У відзначених випадках екстрених ситуацій необхідно скорегувати відповідні дані й повторити їхнє уведення.

4.2 Дослідження ймовірностей виявлення та пропуску помилок логарифмічним контролем

4.2.1 За результатами моделювання матричного помножувача визначаються наступні дані:

P_{BC} – імовірність виявлення суттєвої помилки в результаті;

$P_{BC\ 0 \rightarrow 1}$ – імовірність виявлення суттєвої помилки виду $0 \rightarrow 1$;

$P_{BC\ 1 \rightarrow 0}$ – імовірність виявлення суттєвої помилки виду $1 \rightarrow 0$;

P_{BH} – імовірність виявлення несуттєвої помилки в результаті;

$P_{BH\ 0 \rightarrow 1}$ – імовірність виявлення несуттєвої помилки виду $0 \rightarrow 1$;

$P_{BH\ 1 \rightarrow 0}$ – імовірність виявлення несуттєвої помилки виду $1 \rightarrow 0$;

P_{PC} – імовірність пропуску суттєвої помилки результату;

$P_{PC\ 0 \rightarrow 1}$ – імовірність пропуску суттєвої помилки виду $0 \rightarrow 1$;

$P_{PC\ 1 \rightarrow 0}$ – імовірність пропуску суттєвої помилки виду $1 \rightarrow 0$;

$P_{ПН}$ – імовірність пропуску несуттєвої помилки результату;
 $P_{ПН 0 \rightarrow 1}$ – імовірність пропуску несуттєвої помилки виду $0 \rightarrow 1$;
 $P_{ПН 1 \rightarrow 0}$ – імовірність пропуску несуттєвої помилки виду $1 \rightarrow 0$;
 P_B – імовірність виявлення помилки в результаті;
 $P_{B 0 \rightarrow 1}$ – імовірність виявлення помилки виду $0 \rightarrow 1$;
 $P_{B 1 \rightarrow 0}$ – імовірність виявлення помилки виду $1 \rightarrow 0$;
 $P_{П}$ – імовірність пропуску помилки результату;
 $P_{П 0 \rightarrow 1}$ – імовірність пропуску помилки виду $0 \rightarrow 1$;
 $P_{П 1 \rightarrow 0}$ – імовірність пропуску помилки виду $1 \rightarrow 0$;
 $P_{ППС}$ – імовірність появи суттєвої помилки в результаті;
 $P_{ППС 0 \rightarrow 1}$ – імовірність появи суттєвої помилки виду $0 \rightarrow 1$;
 $P_{ППС 1 \rightarrow 0}$ – імовірність появи суттєвої помилки виду $1 \rightarrow 0$;
 $P_{ППН}$ – імовірність появи несуттєвої помилки в результаті;
 $P_{ППН 0 \rightarrow 1}$ – імовірність появи несуттєвої помилки виду $0 \rightarrow 1$;
 $P_{ППН 1 \rightarrow 0}$ – імовірність появи несуттєвої помилки виду $1 \rightarrow 0$;
 $P_{ПП}$ – імовірність появи помилки в результаті;
 $P_{ПП 0 \rightarrow 1}$ – імовірність появи помилки виду $0 \rightarrow 1$;
 $P_{ПП 1 \rightarrow 0}$ – імовірність появи помилки виду $1 \rightarrow 0$.

Отримані дані дозволяють оцінити достовірність $D_{ЛОГ}$ логарифмічного контролю результатів обробки наближених даних в матричному помножувачі по наступній формулі:

$$D_{ЛОГ} = P_{ВС} + P_{ПН}.$$

Ефективність логарифмічного контролю в частині підвищення достовірності контролю наближених результатів може бути оцінена в порівнянні із загальноприйнятим числовим контролем за модулем, користуючись наступним показником:

$$\delta D = D_{ЛОГ} / D_{МОД},$$

де $D_{МОД}$ – достовірність контролю за модулем три результатів обробки наближених даних.

4.2.2 Моделювання процесу виконання логарифмічного контролю помножувача з фіксованою точкою виконувалося при наступних умовах:

- стан пристрою встановлюється в значення «несправно»;
- випадковий вибір несправного операційного елемента пристрою;
- довільне завдання виду «замикання» двох точок у межах одного операційного елемента структури матричного помножувача з фіксованою точкою [15];
- повний перебір всіх вхідних слів, а також робота на випадкових вхідних послідовностях.

У табл. 4.1 показано розташування основних та узагальнюючих результатів моделювання логарифмічного контролю матричного помножувача з фіксованою точкою, одержуваних з введенням кратних несправностей пристрою в процесі проведення діагностичного експерименту.

Таблиця 4.1 – Розташування результатів моделювання при їхньому виведенні в процесі діагностичного експерименту

ДІЯ	ПОМИЛКИ		УСЬОГО
	СУТТЄВІ (ВІДМОВИ)	НЕСУТТЄВІ	
ВИЯВЛЕННЯ	R_{BC}	R_{BH}	R_B
	$R_{BC\ 0 \rightarrow 1}$	$R_{BH\ 0 \rightarrow 1}$	$R_{B\ 0 \rightarrow 1}$
	$R_{BC\ 1 \rightarrow 0}$	$R_{BH\ 1 \rightarrow 0}$	$R_{B\ 1 \rightarrow 0}$
ПРОПУСК	R_{PC}	R_{PH}	R_P
	$R_{PC\ 0 \rightarrow 1}$	$R_{PH\ 0 \rightarrow 1}$	$R_{P\ 0 \rightarrow 1}$
	$R_{PC\ 1 \rightarrow 0}$	$R_{PH\ 1 \rightarrow 0}$	$R_{P\ 1 \rightarrow 0}$
УСЬОГО	R_{PPC}	R_{PPH}	R_{PP}
	$R_{PPC\ 0 \rightarrow 1}$	$R_{PPH\ 0 \rightarrow 1}$	$R_{PP\ 0 \rightarrow 1}$
	$R_{PPC\ 1 \rightarrow 0}$	$R_{PPH\ 1 \rightarrow 0}$	$R_{PP\ 1 \rightarrow 0}$

Основні результати моделювання логарифмічного контролю наведені в табл. 4.2 для базового випадку, де кількість n_T вірних розрядів визначається довжиною $L_{MIN} = \min(KA, KB)$ найбільш короткої зі значущих частин операндів.

Таблиця 4.2 – Імовірності P_{BC} , $P_{BC\ 0 \rightarrow 1}$, $P_{BC\ 1 \rightarrow 0}$, P_{BH} , $P_{BH\ 0 \rightarrow 1}$, $P_{BH\ 1 \rightarrow 0}$, P_{PC} , $P_{PC\ 0 \rightarrow 1}$, $P_{PC\ 1 \rightarrow 0}$, P_{PH} , $P_{PH\ 0 \rightarrow 1}$, $P_{PH\ 1 \rightarrow 0}$, P_{PPC} , $P_{PPC\ 0 \rightarrow 1}$, $P_{PPC\ 1 \rightarrow 0}$, P_{PPH} , $P_{PPH\ 0 \rightarrow 1}$ та $P_{PPH\ 1 \rightarrow 0}$ для $n_T = L_{MIN}$

№ п/п	Результати моделювання, %	n							
		8	9	10	11	12	13	14	15
1	P_{BC}	1,4	1,1	0,9	0,8	0,6	0,5	0,5	0,4
2	$P_{BC\ 0 \rightarrow 1}$	0,9	0,7	0,6	0,5	0,4	0,4	0,3	0,3
3	$P_{BC\ 1 \rightarrow 0}$	0,5	0,4	0,3	0,3	0,2	0,1	0,2	0,1
4	P_{BH}	0,1	0,0	0,0	0,0	0,0	0	0,0	0,0
5	$P_{BH\ 0 \rightarrow 1}$	0,0	0,	0,0	0,0	0,0	0	0,0	0,0
6	$P_{BH\ 1 \rightarrow 0}$	0,1	0,0	0,0	0,0	0,0	0	0,0	0,0
7	P_{PC}	9	9,3	9,6	9,6	9,7	9,8	9,9	9,9
8	$P_{PC\ 0 \rightarrow 1}$	4,5	4,5	4,6	4,5	4,5	4,5	4,5	4,5
9	$P_{PC\ 1 \rightarrow 0}$	4,5	4,8	5,0	5,1	5,2	5,3	5,4	5,4
10	P_{PH}	10,5	10,7	10,8	10,9	11,0	11,1	11,2	11,2
11	$P_{PH\ 0 \rightarrow 1}$	5,0	5,0	5,0	5,0	5,0	5,0	5,0	5,0
12	$P_{PH\ 1 \rightarrow 0}$	5,5	5,7	5,8	5,9	6,0	6,1	6,2	6,2
13	P_{PPC}	10,4	10,4	10,5	10,4	10,3	10,3	10,4	10,3
14	$P_{PPC\ 0 \rightarrow 1}$	5,4	5,2	5,2	5,0	4,9	4,9	4,8	4,8
15	$P_{PPC\ 1 \rightarrow 0}$	5,0	5,2	5,3	5,4	5,4	5,4	5,6	5,5
16	P_{PPH}	10,6	10,7	10,8	10,9	11,0	11,1	11,2	11,2
17	$P_{PPH\ 0 \rightarrow 1}$	5,0	5,0	5,0	5,0	5,0	5,0	5,0	5,0
18	$P_{PPH\ 1 \rightarrow 0}$	5,6	5,7	5,8	5,9	6,0	5,1	5,2	6,2

У табл. 4.4 наведені основні результати моделювання логарифмічного контролю матричного помножувача з фіксованою точкою для випадку зменшення кількості вірних розрядів на 2 у порівнянні з базовим варіантом.

Таблиця 4.4 – Імовірності P_{BC} , $P_{BC\ 0 \rightarrow 1}$, $P_{BC\ 1 \rightarrow 0}$, P_{BH} , $P_{BH\ 0 \rightarrow 1}$, $P_{BH\ 1 \rightarrow 0}$, P_{PC} , $P_{PC\ 0 \rightarrow 1}$, $P_{PC\ 1 \rightarrow 0}$, P_{PH} , $P_{PH\ 0 \rightarrow 1}$, $P_{PH\ 1 \rightarrow 0}$, P_{PPC} , $P_{PPC\ 0 \rightarrow 1}$, $P_{PPC\ 1 \rightarrow 0}$, P_{PPH} , $P_{PPH\ 0 \rightarrow 1}$ та $P_{PPH\ 1 \rightarrow 0}$ для $n_T = L_{MIN} - 2$

№ п/п	Результати моделювання, %	n							
		8	9	10	11	12	13	14	15
1	P_{BC}	1,3	0,9	0,9	0,8	0,7	0,5	0,5	0,4
2	$P_{BC\ 0 \rightarrow 1}$	0,9	0,6	0,6	0,5	0,54	0,4	0,3	0,3
3	$P_{BC\ 1 \rightarrow 0}$	0,4	0,3	0,3	0,3	0,3	0,1	0,2	0,1
4	P_{BH}	0,1	0	0	0	0	0	0	0
5	$P_{BH\ 0 \rightarrow 1}$	0,1	0	0	0	0	0	0	0
6	$P_{BH\ 1 \rightarrow 0}$	0,0	0	0	0	0	0	0	0
7	P_{PC}	4,5	5,7	5,7	6,2	6,7	6,9	7,1	7,4
8	$P_{PC\ 0 \rightarrow 1}$	2,3	2,8	2,8	3,0	3,1	3,2	3,3	3,4
9	$P_{PC\ 1 \rightarrow 0}$	2,2	2,9	2,9	3,2	3,6	3,7	3,8	4,0
10	P_{PH}	15,0	15,0	14,6	14,4	14,2	14,1	14	13,9
11	$P_{PH\ 0 \rightarrow 1}$	7,1	7,0	6,8	6,6	6,5	6,4	6,3	6,2
12	$P_{PH\ 1 \rightarrow 0}$	7,9	8,0	7,8	7,8	7,7	7,7	7,7	7,7
13	P_{PPC}	5,8	6,8	6,6	7,0	7,2	7,4	7,6	7,8
14	$P_{PPC\ 0 \rightarrow 1}$	3,2	3,4	3,4	3,5	3,5	3,6	3,6	3,7
15	$P_{PPC\ 1 \rightarrow 0}$	2,6	3,4	3,2	3,5	3,7	3,8	4,0	4,1
16	P_{PPH}	15,1	15,1	14,6	14,4	14,2	14,1	14,0	13,9
17	$P_{PPH\ 0 \rightarrow 1}$	7,2	7,3	6,8	6,6	6,5	6,4	6,3	6,2
18	$P_{PPH\ 1 \rightarrow 0}$	7,9	7,8	7,8	7,8	7,7	7,7	7,7	7,7

У табл. 4.5 наведені узагальнюючі ймовірності виявлення, пропуску та появи помилок P_B , $P_{B0 \rightarrow 1}$, $P_{B1 \rightarrow 0}$, P_{Π} , $P_{\Pi 0 \rightarrow 1}$, $P_{\Pi 1 \rightarrow 0}$, $P_{\Pi\Pi}$, $P_{\Pi\Pi 0 \rightarrow 1}$ та $P_{\Pi\Pi 1 \rightarrow 0}$, значення достовірності $D_{\text{ЛОГ}}$ та $D_{\text{МОД}}$ логарифмічного контролю та контролю результатів за числовим модулем три, а також показник ефективності δD для випадку кількості вірних розрядів $n_T = L_{\text{MIN}} - 2$.

Таблиця 4.5 – Імовірності P_B , $P_{B0 \rightarrow 1}$, $P_{B1 \rightarrow 0}$, P_{Π} , $P_{\Pi 0 \rightarrow 1}$, $P_{\Pi 1 \rightarrow 0}$, $P_{\Pi\Pi}$, $P_{\Pi\Pi 0 \rightarrow 1}$, $P_{\Pi\Pi 1 \rightarrow 0}$, $D_{\text{ЛОГ}}$, $D_{\text{МОД}}$ та δD для $n_T = L_{\text{MIN}} - 2$

№ п/п	Результати моделювання, %	n							
		8	9	10	11	12	13	14	15
1	P_B	1,4	0,9	0,9	0,8	0,7	0,5	0,5	0,4
2	$P_{B0 \rightarrow 1}$	1,0	0,6	0,6	0,5	0,4	0,4	0,3	0,3
3	$P_{B1 \rightarrow 0}$	0,4	0,3	0,3	0,3	0,3	0,1	0,2	0,1
4	P_{Π}	19,5	20,3	20,3	20,6	20,9	21	21,1	21,3
5	$P_{\Pi 0 \rightarrow 1}$	9,4	9,3	9,6	9,6	9,6	9,6	9,6	9,6
6	$P_{\Pi 1 \rightarrow 0}$	10,1	11,0	10,7	11,0	11,3	11,4	11,5	11,7
7	$P_{\Pi\Pi}$	20,9	21	21,2	21,4	21,4	21,5	21,6	21,7
8	$P_{\Pi\Pi 0 \rightarrow 1}$	10,4	10,3	10,2	10,1	10,0	10,0	9,9	9,9
9	$P_{\Pi\Pi 1 \rightarrow 0}$	10,5	10,7	11,0	11,3	11,4	11,5	11,7	11,8
10	$D_{\text{ЛОГ}}$	95,4	94,3	94,3	93,8	93,3	93,1	92,9	92,6
11	$D_{\text{МОД}}$	78,2	78,0	78,1	78,1	78,0	78,1	78,0	77,9
12	δD	1,22	1,21	1,21	1,20	1,20	1,19	1,19	1,19

У табл. 4.6 наведені основні результати моделювання логарифмічного контролю матричного помножувача з фіксованою точкою під дією кратних несправностей пристрою із зменшенням кількості вірних розрядів результату на 4 у порівнянні з базовим випадком.

Таблиця 4.6 – Імовірності P_{BC} , $P_{BC\ 0 \rightarrow 1}$, $P_{BC\ 1 \rightarrow 0}$, P_{BH} , $P_{BH\ 0 \rightarrow 1}$, $P_{BH\ 1 \rightarrow 0}$, P_{PC} , $P_{PC\ 0 \rightarrow 1}$, $P_{PC\ 1 \rightarrow 0}$, P_{PH} , $P_{PH\ 0 \rightarrow 1}$, $P_{PH\ 1 \rightarrow 0}$, P_{PPC} , $P_{PPC\ 0 \rightarrow 1}$, $P_{PPC\ 1 \rightarrow 0}$, P_{PPH} , $P_{PPH\ 0 \rightarrow 1}$ та $P_{PPH\ 1 \rightarrow 0}$ для $n_T = L_{MIN} - 4$

№ п/п	Результати моделювання, %	n							
		8	9	10	11	12	13	14	15
1	P_{BC}	1,1	1,0	0,9	0,7	0,6	0,5	0,5	0,4
2	$P_{BC\ 0 \rightarrow 1}$	0,7	0,7	0,6	0,5	0,4	0,2	0,3	0,3
3	$P_{BC\ 1 \rightarrow 0}$	0,4	0,3	0,3	0,2	0,2	0,3	0,2	0,1
4	P_{BH}	0,4	0,2	0,1	0	0	0	0	0
5	$P_{BH\ 0 \rightarrow 1}$	0,3	0,1	0,1	0	0	0	0	0
6	$P_{BH\ 1 \rightarrow 0}$	0,1	0,1	0,0	0	0	0	0	0
7	P_{PC}	1,4	2,1	2,8	3,4	3,9	4,4	4,7	5,1
8	$P_{PC\ 0 \rightarrow 1}$	0,8	1,1	1,4	1,7	1,9	2,1	2,2	2,4
9	$P_{PC\ 1 \rightarrow 0}$	0,6	1,0	1,4	1,7	2,0	2,3	2,5	2,7
10	P_{PH}	18,1	17,8	17,5	17,2	16,9	16,4	16,4	16,1
11	$P_{PH\ 0 \rightarrow 1}$	8,6	8,4	8,1	7,9	7,7	7,4	7,3	7,2
12	$P_{PH\ 1 \rightarrow 0}$	9,5	9,4	9,4	9,3	9,2	9,0	9,1	8,9
13	P_{PPC}	2,5	3,1	3,7	4,1	4,5	4,9	5,2	5,5
14	$P_{PPC\ 0 \rightarrow 1}$	1,5	1,8	2,0	2,2	2,3	2,4	2,5	2,7
15	$P_{PPC\ 1 \rightarrow 0}$	1,0	1,3	1,7	1,9	2,2	2,5	2,7	2,8
16	P_{PPH}	18,5	18	17,6	17,2	16,9	16,4	16,4	16,1
17	$P_{PPH\ 0 \rightarrow 1}$	8,9	8,5	8,2	7,9	7,7	7,4	7,3	7,2
18	$P_{PPH\ 1 \rightarrow 0}$	9,6	9,8	9,4	9,35	9,2	9,0	9,1	8,9

У табл. 4.7 наведені узагальнюючі ймовірності виявлення, пропуску та появи помилок P_B , $P_{B\ 0 \rightarrow 1}$, $P_{B\ 1 \rightarrow 0}$, P_P , $P_{P\ 0 \rightarrow 1}$, $P_{P\ 1 \rightarrow 0}$, P_{PP} , $P_{PP\ 0 \rightarrow 1}$ та $P_{PP\ 1 \rightarrow 0}$ для випадку кількості вірних розрядів $n_T = L_{MIN} - 4$.

Таблиця 4.7 – Імовірності P_B , $P_{B0 \rightarrow 1}$, $P_{B1 \rightarrow 0}$, P_{Π} , $P_{\Pi 0 \rightarrow 1}$, $P_{\Pi 1 \rightarrow 0}$, $P_{\Pi\Pi}$, $P_{\Pi\Pi 0 \rightarrow 1}$, $P_{\Pi\Pi 1 \rightarrow 0}$, $D_{\text{ЛОГ}}$, $D_{\text{МОД}}$ та δD для $n_T = L_{\text{MIN}} - 4$

№ п/п	Результати моделювання, %	n							
		8	9	10	11	12	13	14	15
1	P_B	1,5	1,2	1,0	0,7	0,6	0,5	0,5	0,4
2	$P_{B0 \rightarrow 1}$	1,0	0,8	0,7	0,5	0,4	0,3	0,3	0,3
3	$P_{B1 \rightarrow 0}$	0,5	0,4	0,3	0,2	0,2	0,2	0,2	0,1
4	P_{Π}	19,5	19,9	20,3	20,6	20,8	20,8	21,1	21,2
5	$P_{\Pi 0 \rightarrow 1}$	9,4	9,5	9,5	9,6	9,6	9,5	9,5	9,6
6	$P_{\Pi 1 \rightarrow 0}$	10,1	10,4	10,8	11,0	11,2	11,3	11,6	11,6
7	$P_{\Pi\Pi}$	21,0	21,1	21,3	21,3	21,4	21,3	21,6	21,6
8	$P_{\Pi\Pi 0 \rightarrow 1}$	10,4	10,3	10,2	10,1	10,0	9,8	9,8	9,9
9	$P_{\Pi\Pi 1 \rightarrow 0}$	10,6	10,8	11,1	11,2	11,4	11,5	11,8	11,7
10	$D_{\text{ЛОГ}}$	1,8	2,3	2,9	3,4	3,9	4,4	4,7	5,1
11	$D_{\text{МОД}}$	21,7	22,0	21,7	22,1	21,9	22,0	22,1	22,0
12	δD	12,10	9,56	7,48	6,50	7,44	0,50	4,70	4,31

Крім того, наведені значення достовірності $D_{\text{ЛОГ}}$ та $D_{\text{МОД}}$ логарифмічного контролю та контролю за числовим модулем три наближених результатів, що обчислюються в матричному помножувачі з фіксованою точкою під впливом кратних несправностей пристрою, а також показник ефективності δD для випадку кількості вірних розрядів $n_T = L_{\text{MIN}} - 4$.

У табл. 4.8 наведені основні результати моделювання логарифмічного контролю матричного помножувача з фіксованою точкою, що обчислюються на довільних послідовностях вхідних даних під впливом кратних несправностей пристрою при зменшенні кількості вірних розрядів наближеного результату на b у порівнянні з базовим випадком $n_T = L_{\text{MIN}}$.

Таблиця 4.8 – Імовірності P_{BC} , $P_{BC\ 0 \rightarrow 1}$, $P_{BC\ 1 \rightarrow 0}$, P_{BH} , $P_{BH\ 0 \rightarrow 1}$, $P_{BH\ 1 \rightarrow 0}$, P_{PC} , $P_{PC\ 0 \rightarrow 1}$, $P_{PC\ 1 \rightarrow 0}$, P_{PH} , $P_{PH\ 0 \rightarrow 1}$, $P_{PH\ 1 \rightarrow 0}$, P_{PPC} , $P_{PPC\ 0 \rightarrow 1}$, $P_{PPC\ 1 \rightarrow 0}$, P_{PPH} , $P_{PPH\ 0 \rightarrow 1}$ та $P_{PPH\ 1 \rightarrow 0}$ для $n_T = L_{MIN} - 6$

№ п/п	Результати моделювання, %	n							
		8	9	10	11	12	13	14	15
1	P_{BC}	0,5	0,6	0,7	0,6	0,6	0,5	0,4	0,4
2	$P_{BC\ 0 \rightarrow 1}$	0,3	0,4	0,4	0,4	0,4	0,3	0,3	0,3
3	$P_{BC\ 1 \rightarrow 0}$	0,2	0,2	0,3	0,2	0,2	0,2	0,1	0,1
4	P_{BH}	1,0	0,6	0,3	0,1	0,1	0	0	0
5	$P_{BH\ 0 \rightarrow 1}$	0,6	0,3	0,2	0,1	0,0	0	0	0
6	$P_{BH\ 1 \rightarrow 0}$	0,4	0,3	0,1	0,0	0,1	0	0	0
7	P_{PC}	0,1	0,5	0,9	1,4	1,9	2,4	2,8	3,2
8	$P_{PC\ 0 \rightarrow 1}$	0,1	0,3	0,5	0,7	1,0	1,2	1,4	1,5
9	$P_{PC\ 1 \rightarrow 0}$	0	0,2	0,4	0,7	0,9	1,2	1,4	1,7
10	P_{PH}	0,7	19,5	19,4	19,2	18,8	18,6	18,3	18
11	$P_{PH\ 0 \rightarrow 1}$	0,4	9,2	9,0	8,8	8,6	8,4	8,2	8
12	$P_{PH\ 1 \rightarrow 0}$	0,3	10,3	10,4	10,4	10,2	10,2	10,1	10
13	P_{PPC}	0,6	1,1	1,6	2,0	2,5	2,9	3,2	3,6
14	$P_{PPC\ 0 \rightarrow 1}$	0,4	0,7	0,9	1,1	1,4	1,5	1,7	1,8
15	$P_{PPC\ 1 \rightarrow 0}$	0,2	0,4	0,7	0,9	1,1	1,4	1,5	1,8
16	P_{PPH}	0,9	20,1	19,7	19,3	18,9	18,6	18,3	18
17	$P_{PPH\ 0 \rightarrow 1}$	0,2	9,5	9,2	8,9	8,6	8,4	8,2	8
18	$P_{PPH\ 1 \rightarrow 0}$	0,7	10,6	10,5	10,4	10,3	10,2	10,1	10

У табл. 4.9 наведені узагальнюючі ймовірності виявлення, пропуску та появи помилок P_B , $P_{B\ 0 \rightarrow 1}$, $P_{B\ 1 \rightarrow 0}$, P_P , $P_{P\ 0 \rightarrow 1}$, $P_{P\ 1 \rightarrow 0}$, P_{PP} , $P_{PP\ 0 \rightarrow 1}$ та $P_{PP\ 1 \rightarrow 0}$. для випадку кількості вірних розрядів $n_T = L_{MIN} - 6$.

Таблиця 4.9 – Імовірності P_B , $P_{B0 \rightarrow 1}$, $P_{B1 \rightarrow 0}$, P_{II} , $P_{II0 \rightarrow 1}$, $P_{II1 \rightarrow 0}$, P_{III} , $P_{III0 \rightarrow 1}$, $P_{III1 \rightarrow 0}$, $D_{\text{ЛОГ}}$, $D_{\text{МОД}}$ та δD для $n_T = L_{\text{MIN}} - 6$

№ п/п	Результати моделювання, %	n							
		8	9	10	11	12	13	14	15
1	P_B	1,5	1,2	1,0	0,7	0,7	0,5	0,4	0,4
2	$P_{B0 \rightarrow 1}$	0,9	0,7	0,6	0,5	0,4	0,3	0,3	0,3
3	$P_{B1 \rightarrow 0}$	0,6	0,5	0,4	0,2	0,3	0,2	0,1	0,1
4	P_{II}	0,6	20	20,3	20,6	20,7	21,0	21,1	21,2
5	$P_{II0 \rightarrow 1}$	0,3	9,5	9,5	9,5	9,6	9,6	9,6	9,5
6	$P_{II1 \rightarrow 0}$	0,3	10,5	10,8	11,1	11,1	11,4	11,5	11,7
7	P_{III}	1,5	21,2	21,3	21,3	21,4	21,5	21,5	21,6
8	$P_{III0 \rightarrow 1}$	0,6	10,2	10,1	10,1	10,0	9,9	9,9	9,8
9	$P_{III1 \rightarrow 0}$	0,9	11,0	11,2	11,3	11,4	11,6	11,6	11,8
10	$D_{\text{ЛОГ}}$	1,1	1,1	1,2	1,5	2,0	2,4	2,8	3,2
11	$D_{\text{МОД}}$	21,7	21,8	21,8	21,9	21,9	21,8	21,8	18,4
12	δD	19,7	19,8	18,7	14,6	11,0	9,08	7,79	5,75

Крім того, наведені значення достовірності $D_{\text{ЛОГ}}$ та $D_{\text{МОД}}$ логарифмічного контролю та контролю за числовим модулем три наближених результатів, що обчислюються під дією кратних несправностей типу «закоротка» в матричному помножувачі а також показник ефективності δD для випадку кількості вірних розрядів $n_T = L_{\text{MIN}} - 6$.

Одержані результати моделювання логарифмічного контролю матричного помножувача з фіксованою точкою найбільш наглядно подаються з використанням діаграм залежності ймовірностей та достовірності від розрядності співмножників пристрою.

Остаточні результати моделювання визначають оцінки достовірності $D_{ЛОГ1}$, $D_{ЛОГ2}$, $D_{ЛОГ3}$, $D_{МОД1}$, $D_{МОД2}$, $D_{МОД3}$ граничного контролю щодо повного та усіченого результатів множення мантис при кількості вірних розрядів $n_T = n$ та $n_T = n - 4$ для трьох значень кратності несправності: 1, 2 та 3 наведено в табл. 4.10 та 4.11, відповідно.

Таблиця 4.10 – Достовірність $D_{ЛОГ.n.1}$, $D_{ЛОГ.n.2}$, $D_{ЛОГ.n.3}$, $D_{МОД.n.1}$, $D_{МОД.n.2}$, $D_{МОД.n.3}$ для кратних несправностей та $n_T = n$

№ п/п	Результати моделювання	n							
		8	9	10	11	12	13	14	15
1	$D_{ЛОГ.n.1}$, %	56,8	56,0	55,2	54,6	54,0	54,1	54,2	54,2
2	$D_{ЛОГ.n.2}$, %	49,3	49,2	49,1	48,7	48,4	48,4	48,3	48,3
3	$D_{ЛОГ.n.3}$, %	54,8	54,2	53,6	53,2	52,9	53,1	53,2	53,2
4	$D_{МОД.n.1}$, %	51,8	51,6	51,4	50,6	49,9	49,5	49,1	49,1
5	$D_{МОД.n.2}$, %	53,3	52,7	52,1	51,2	50,3	50,3	50,3	50,3
6	$D_{МОД.n.3}$, %	53,6	53,2	52,9	52,6	52,4	52,4	52,3	52,3

Таблиця 4.11 – Достовірність $D_{ЛОГ.n-4.1}$, $D_{ЛОГ.n-4.2}$, $D_{ЛОГ.n-4.3}$, $D_{МОД.n-4.1}$, $D_{МОД.n-4.2}$, $D_{МОД.n-4.3}$ для кратних несправностей та $n_T = n - 4$.

№ п/п	Результати моделювання	n							
		8	9	10	11	12	13	14	15
1	$D_{ЛОГ.n-4.1}$, %	90,4	88,4	87,1	84,0	81,7	79,8	78,0	76,8
2	$D_{ЛОГ.n-4.2}$, %	12,5	14,5	16,4	18,6	21,0	22,9	24,3	23,6
3	$D_{ЛОГ.n-4.3}$, %	90,3	87,6	85,5	82,8	80,6	78,9	77,2	75,5
4	$D_{МОД.n-4.1}$, %	12,3	15,6	18,2	20,5	22,7	24,6	25,1	26,2
5	$D_{МОД.n-4.2}$, %	90,1	87,8	85,2	82,4	79,1	77,5	76,0	75,0
6	$D_{МОД.n-4.3}$, %	12,9	16,0	18,7	21,6	24,4	25,5	26,5	27,3

4.2.3. На рис. 4.5 показані діаграми залежності ймовірностей P_{BC} , $P_{BC 0 \rightarrow 1}$ та $P_{BC 1 \rightarrow 0}$ виявлення суттєвої помилки логарифмічним контролем від розрядності n матричного помножувача з фіксованою точкою для кількості вірних розрядів $n_T = n$.

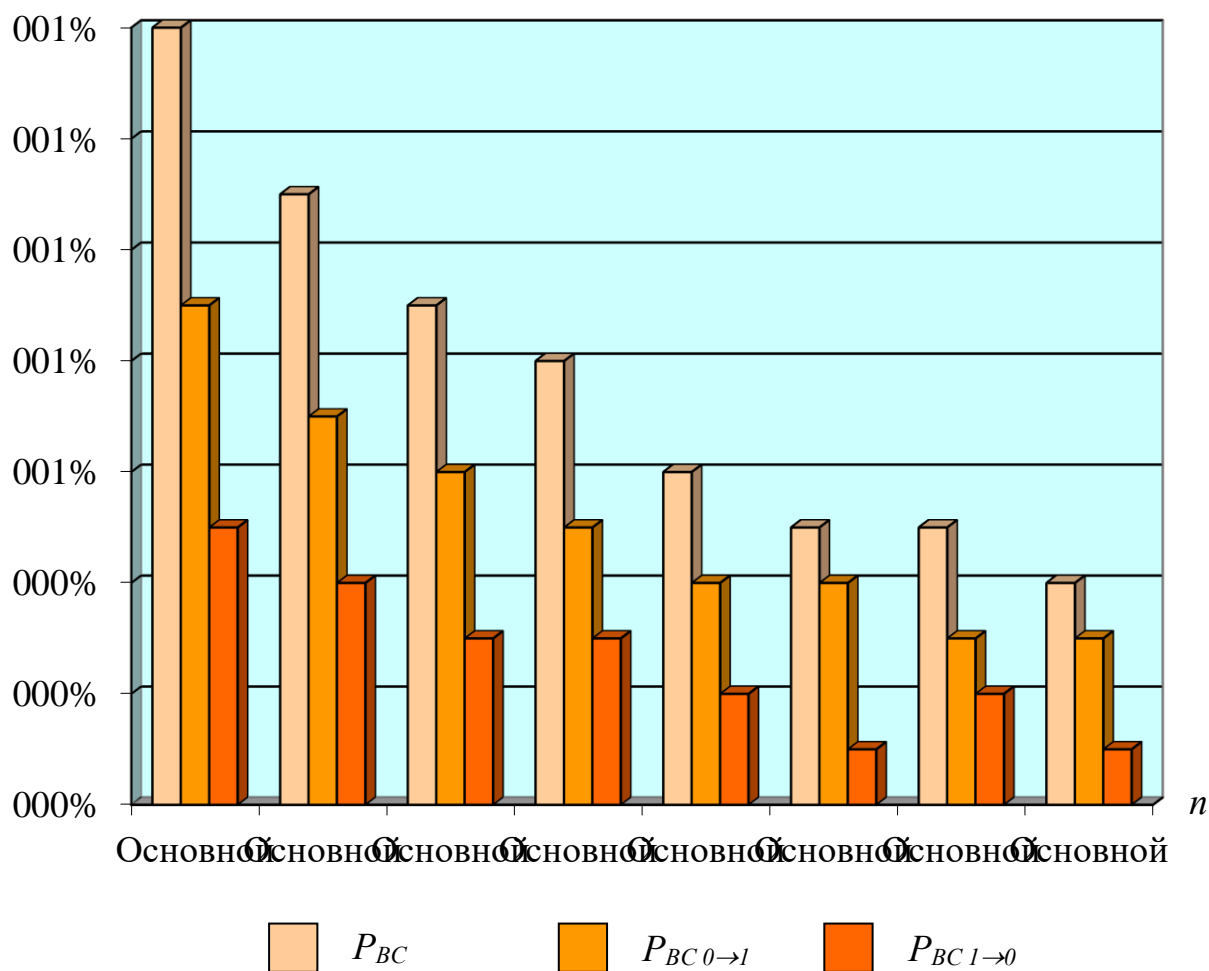


Рисунок 4.5 – Діаграми для ймовірностей P_{BC} , $P_{BC 0 \rightarrow 1}$ та $P_{BC 1 \rightarrow 0}$ виявлення суттєвих помилок

Побудовані діаграми показують, що ймовірності виявлення суттєвих помилок логарифмічним контролем зменшуються із зростанням розрядності операндів. Позитивна суттєва помилка виявляється в матричному помножувачі з фіксованою точкою з більшою ймовірністю в порівнянні з негативною суттєвою помилкою.

На рис. 4.6 показані діаграми зміни ймовірностей $P_{ПС}$, $P_{ПС 0 \rightarrow 1}$ та $P_{ПС 1 \rightarrow 0}$ пропуску суттєвих помилок логарифмічним контролем для ряду значень розрядності n матричних помножувачів з фіксованою точкою та кількості вірних розрядів $n_T = n$.

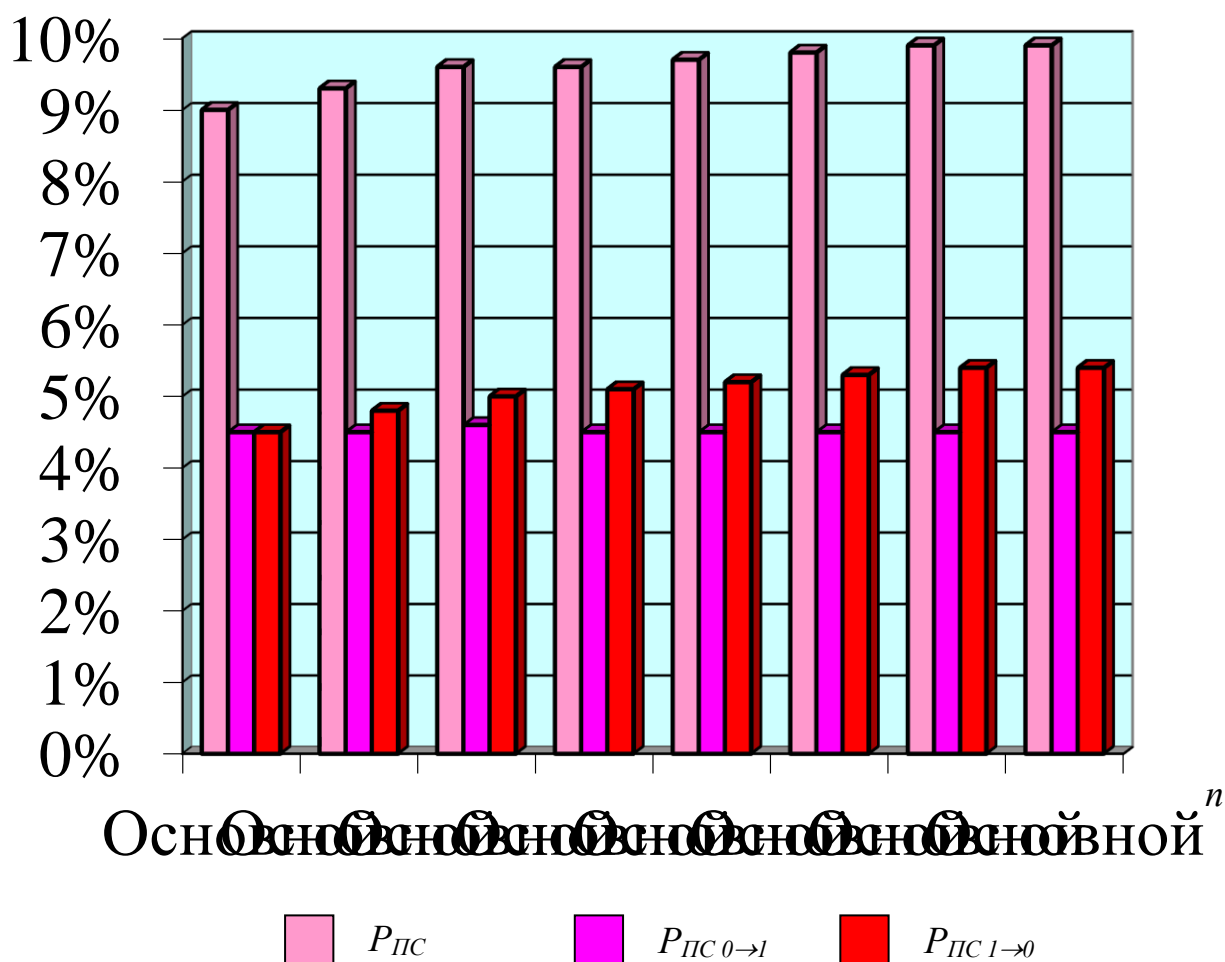


Рисунок 4.6 – Діаграми ймовірностей $P_{ПС}$, $P_{ПС 0 \rightarrow 1}$ та $P_{ПС 1 \rightarrow 0}$ пропуску суттєвих помилок

Побудовані діаграми показують, що ймовірності пропуску суттєвих помилок та ймовірності пропуску суттєвих негативних помилок збільшуються з ростом розрядності операндів. Позитивні суттєві помилки пропускаються логарифмічним контролем з однаковою ймовірністю, меншою в порівнянні з негативною помилкою.

На рис. 4.7 показані діаграми зміни ймовірностей $P_{ПН}$, $P_{ПН 0 \rightarrow 1}$ та $P_{ПН 1 \rightarrow 0}$ пропуску несуттєвої помилки логарифмічним контролем для ряду значень розрядності n матричних помножувачів з фіксованою точкою і кількості вірних розрядів $n_T = n$.

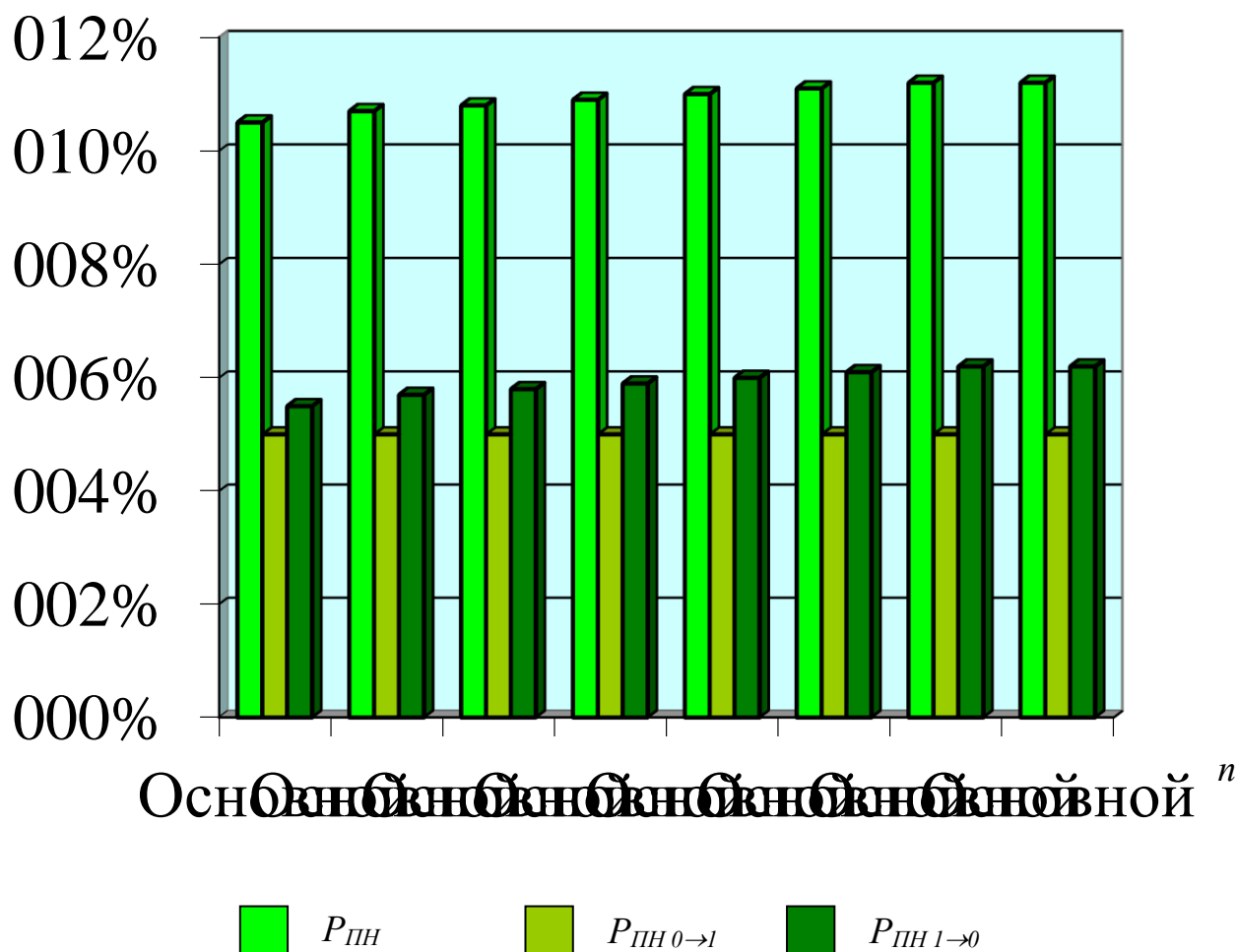


Рисунок 4.7 – Діаграми ймовірностей $P_{ПН}$, $P_{ПН 0 \rightarrow 1}$ та $P_{ПН 1 \rightarrow 0}$ пропуску несуттєвих помилок

Побудовані діаграми для оцінки логарифмічного контролю показують, що ймовірності пропуску несуттєвих помилок повторюють вид діаграм отриманих для ймовірності пропуску суттєвих помилок, але приймають значення в середньому на 10% більші.

На рис. 4.8 показані діаграми зміни ймовірностей P_B , $P_{B0 \rightarrow 1}$ та $P_{B1 \rightarrow 0}$ виявлення помилок логарифмічним контролем для ряду значень розрядності n матричних помножувачів з фіксованою точкою, а також кількості вірних розрядів $n_T = n$.

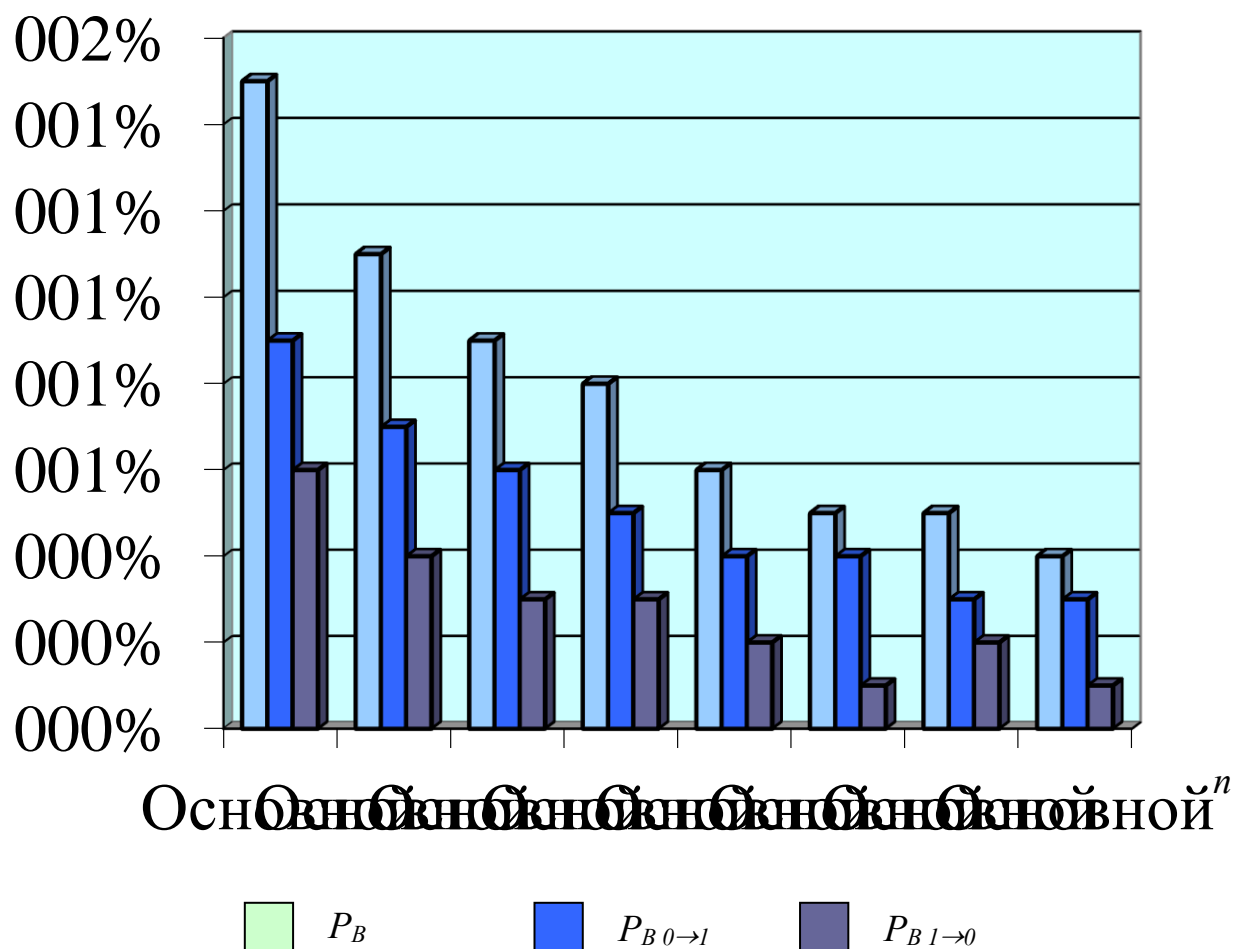


Рисунок 4.8 – Діаграми ймовірностей P_B , $P_{B0 \rightarrow 1}$ та $P_{B1 \rightarrow 0}$ виявлення помилок

Побудовані діаграми для оцінки логарифмічного контролю матричного помножувача з фіксованою точкою показують, що ймовірності виявлення помилок зменшуються із зростанням розрядності операндів. Позитивна помилка наближеного результату виявляється з більшою ймовірністю в порівнянні з негативною помилкою.

На рис. 4.9 показані діаграми зміни ймовірностей P_{Π} , $P_{\Pi 0 \rightarrow 1}$ та $P_{\Pi 1 \rightarrow 0}$ пропуску помилок логарифмічним контролем матричного помножувача з фіксованою точкою для ряду значень розрядності n матричних помножувачів і кількості вірних розрядів $n_T = n$.

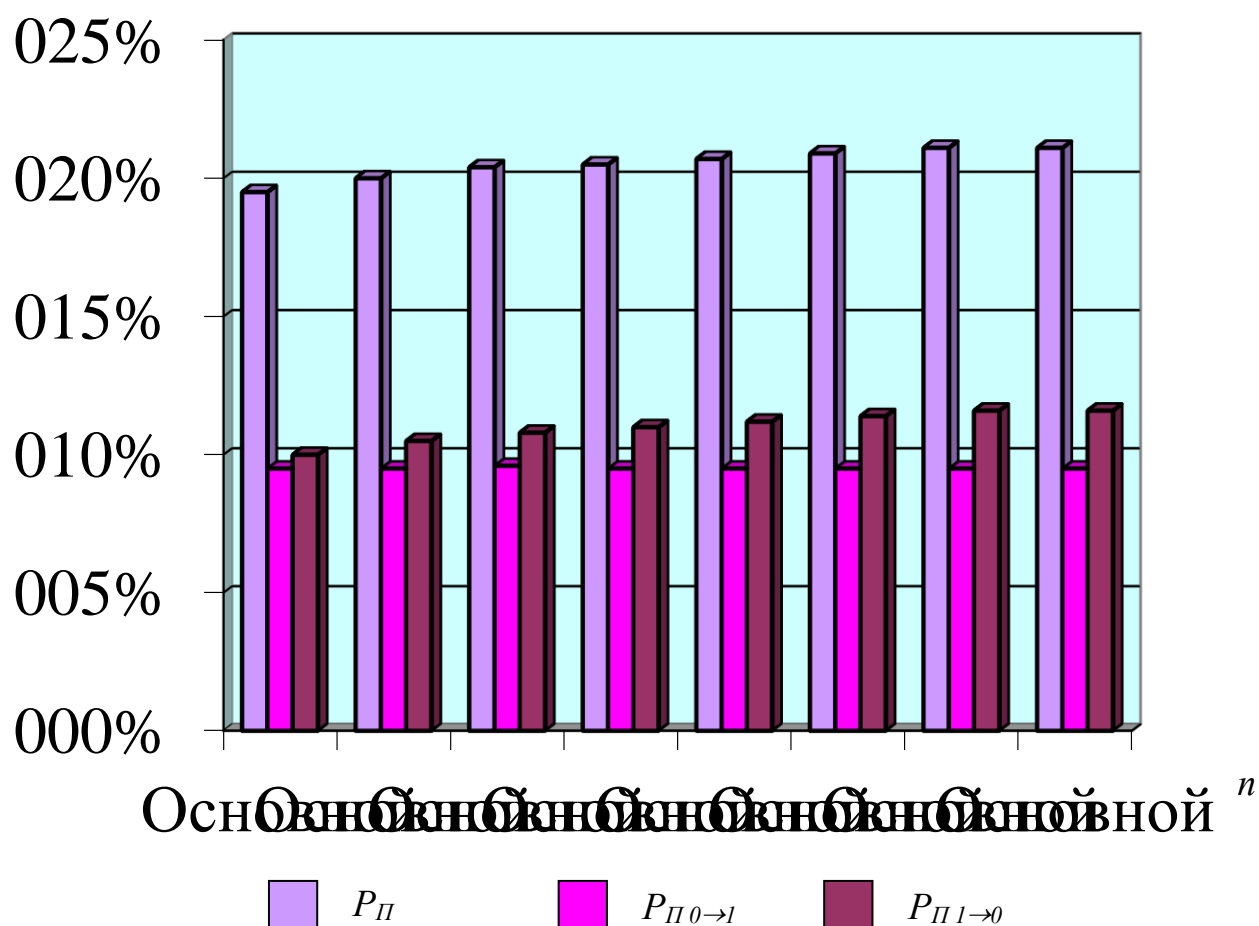


Рисунок 4.9 – Діаграми ймовірностей P_{Π} , $P_{\Pi 0 \rightarrow 1}$ та $P_{\Pi 1 \rightarrow 0}$ пропуску помилок

Побудовані діаграми для оцінки логарифмічного контролю матричного помножувача з фіксованою точкою показують, що ймовірності пропуску помилок і негативних помилок збільшуються з ростом розрядності операндів. Позитивні помилки виявляється з однаковою ймовірністю, меншої в порівнянні з негативною помилкою.

На рис. 4.10 показані діаграми зміни достовірності $D_{ЛОГ}$ та $D_{МОД}$ логарифмічного контролю та контролю за модулем для ряду значень розрядності n матричних помножувачів з фіксованою точкою та кількості вірних розрядів $n_T = n$.

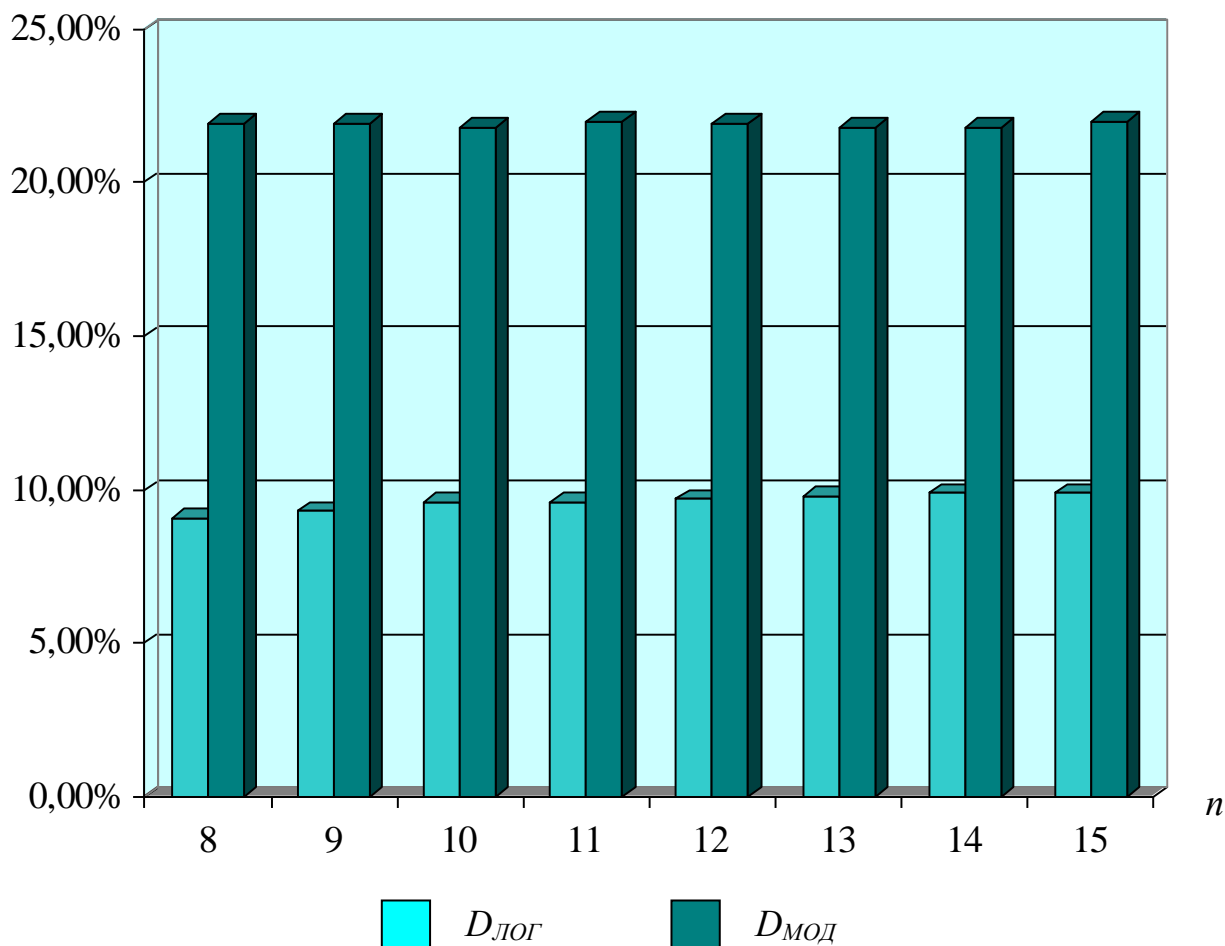


Рисунок 4.10 – Діаграми оцінки достовірності $D_{ЛОГ}$ та $D_{МОД}$ логарифмічного контролю та контролю по модулі

Побудовані діаграми для оцінки логарифмічного контролю матричного помножувача з фіксованою точкою показують, що достовірність $D_{ЛОГ}$ логарифмічного контролю значно менше достовірності $D_{МОД}$ контролю за модулем. Достовірність $D_{ЛОГ}$ логарифмічного контролю підвищується разом з розрядністю співмножників, а достовірність $D_{МОД}$ контролю за модулем практично не змінюється.

На рис. 4.11 показана діаграма зміни показника ефективності δD щодо підвищення достовірності логарифмічного контролю в порівнянні з контролем за модулем для ряду значень розрядності n матричних помножувачів та кількості вірних розрядів $n_T = n$.

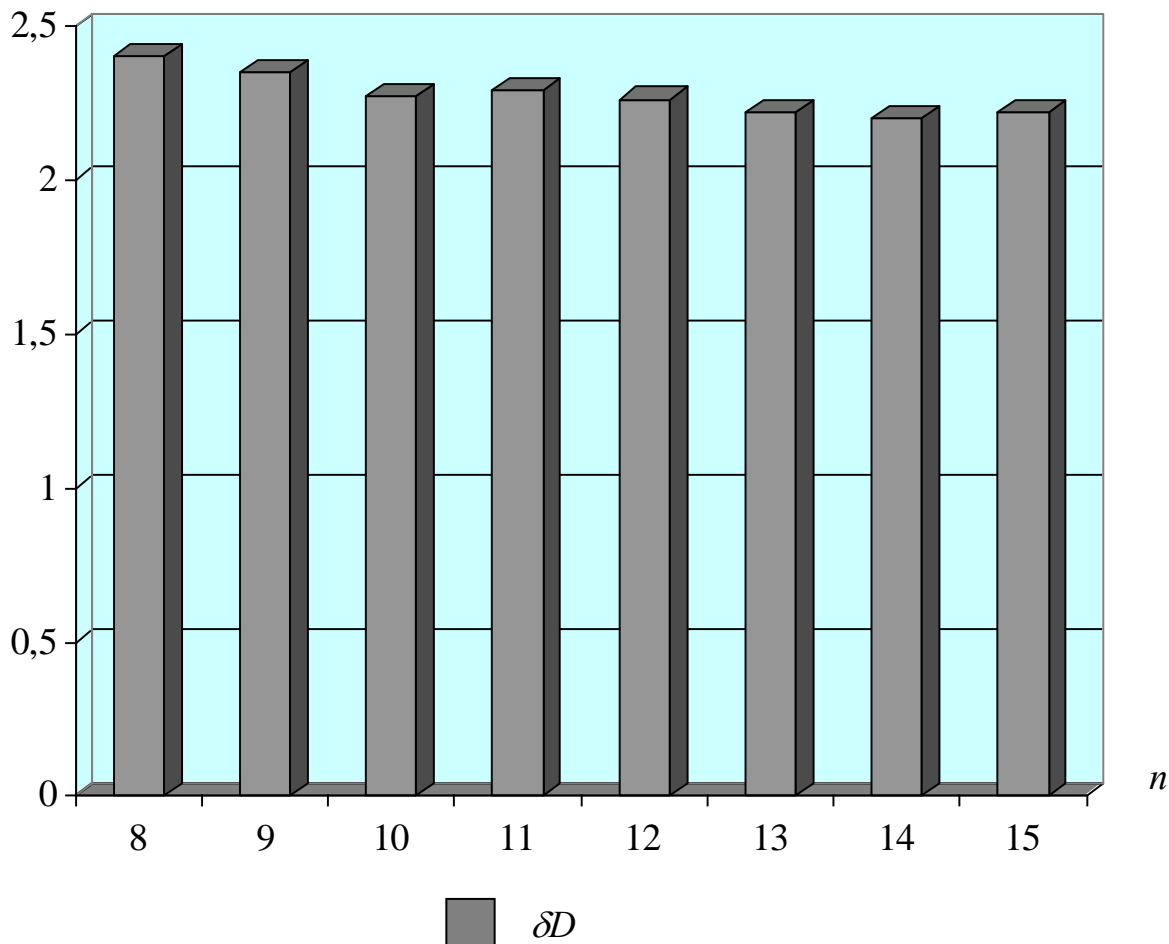


Рисунок 4.11 – Діаграми показника δD ефективності логарифмічного контролю порівняно до контролю за модулем

Побудована діаграма для оцінки логарифмічного контролю матричного помножувача з фіксованою точкою показує більш високу достовірність логарифмічного контролю результатів і незначне зниження показника δD з ростом розрядності операндів.

На рис. 4.12 показані діаграми зміни ймовірностей P_{BC} , $P_{BC 0 \rightarrow 1}$ та $P_{BC 1 \rightarrow 0}$ виявлення суттєвої помилки логарифмічним контролем для ряду значень розрядності n матричних помножувачів з фіксованою точкою та кількості вірних розрядів $n_T = n - 2$.

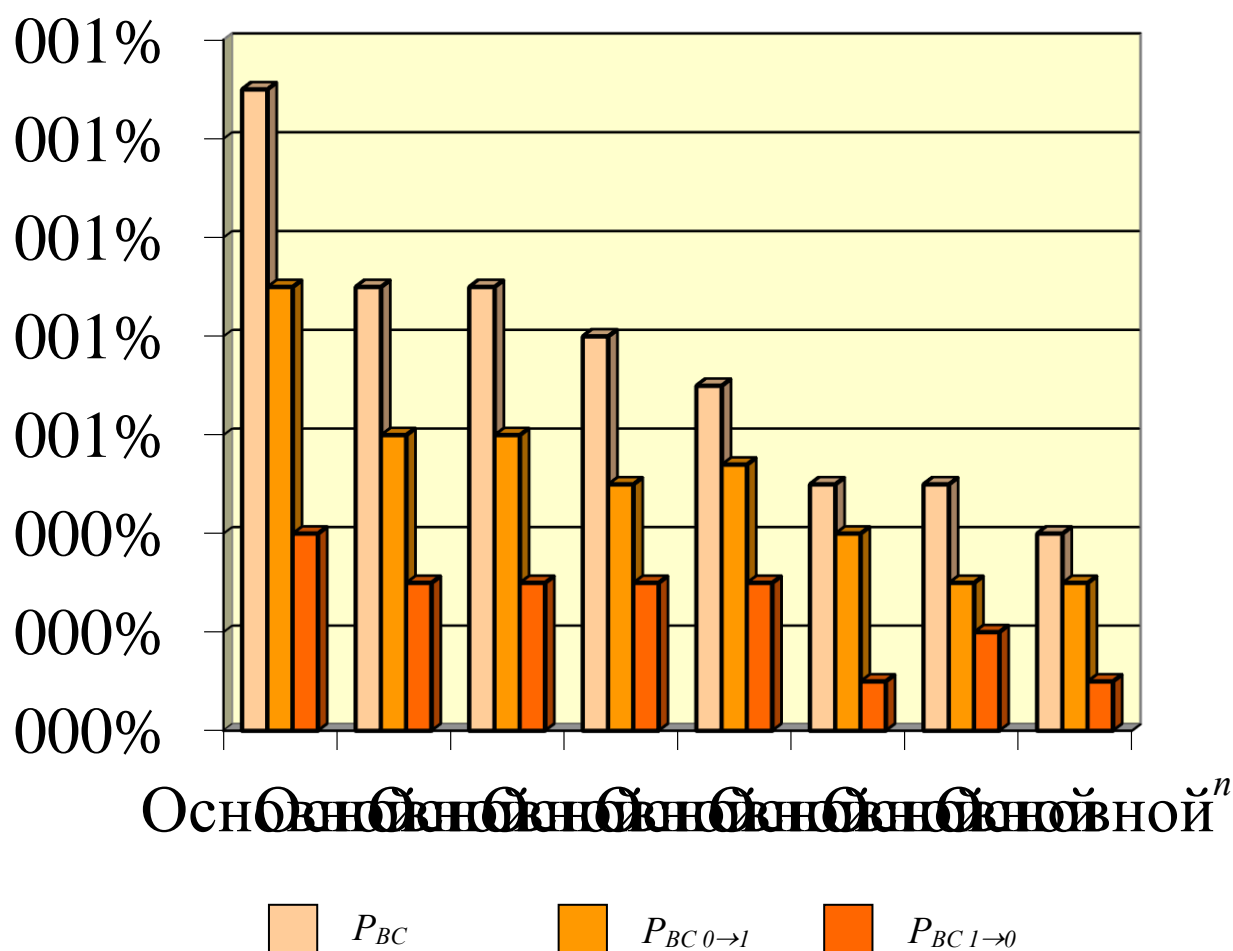


Рисунок 4.12 – Діаграми ймовірностей P_{BC} , $P_{BC 0 \rightarrow 1}$ та $P_{BC 1 \rightarrow 0}$ виявлення суттєвих помилок

Побудовані діаграми для оцінки логарифмічного контролю матричного помножувача з фіксованою точкою показують, що ймовірності виявлення суттєвих помилок зменшуються з ростом розрядності операндів. Позитивна помилка виявляється з більшою ймовірністю в порівнянні з негативною помилкою.

На рис. 4.13 показані діаграми зміни ймовірностей $P_{ПС}$, $P_{ПС 0 \rightarrow 1}$ та $P_{ПС 1 \rightarrow 0}$ пропуску суттєвої помилки логарифмічним контролем для ряду значень розрядності n матричних помножувачів з фіксованою точкою та кількості вірних розрядів $n_T = n - 2$.

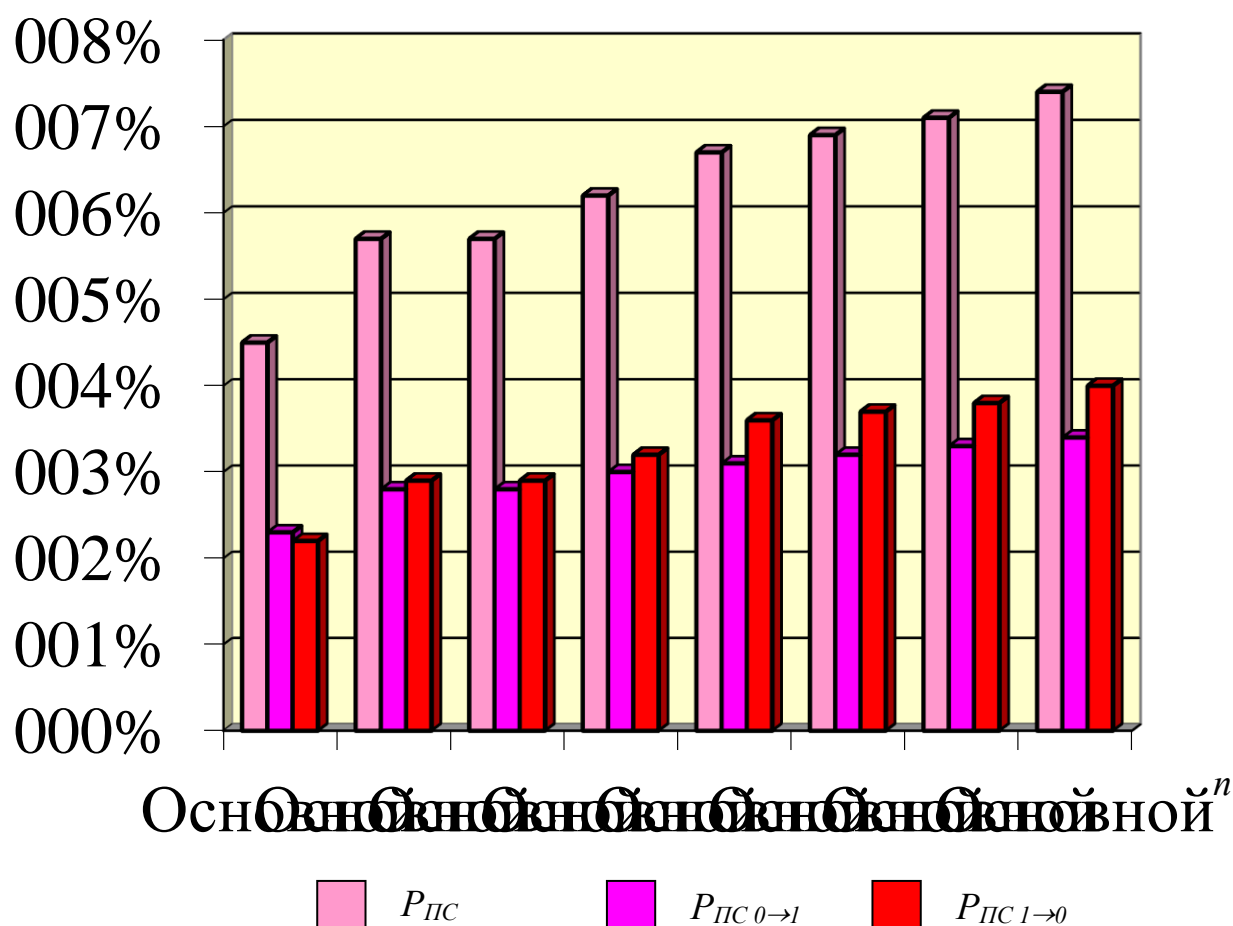


Рисунок 4.13 – Діаграми ймовірностей $P_{ПС}$, $P_{ПС 0 \rightarrow 1}$ та $P_{ПС 1 \rightarrow 0}$ пропуску суттєвих помилок

Побудовані діаграми для оцінки логарифмічного контролю матричного помножувача з фіксованою точкою показують, що ймовірності пропуску суттєвих помилок збільшуються з ростом розрядності операндів. Позитивні суттєві помилки пропускаються з незначно меншою ймовірністю в порівнянні з негативною помилкою.

На рис. 4.14 показані діаграми зміни ймовірностей $P_{ПН}$, $P_{ПН 0 \rightarrow 1}$ та $P_{ПН 1 \rightarrow 0}$ пропуску несуттєвої помилки логарифмічним контролем для ряду значень розрядності n матричних помножувачів з фіксованою точкою та кількості вірних розрядів $n_T = n - 2$.

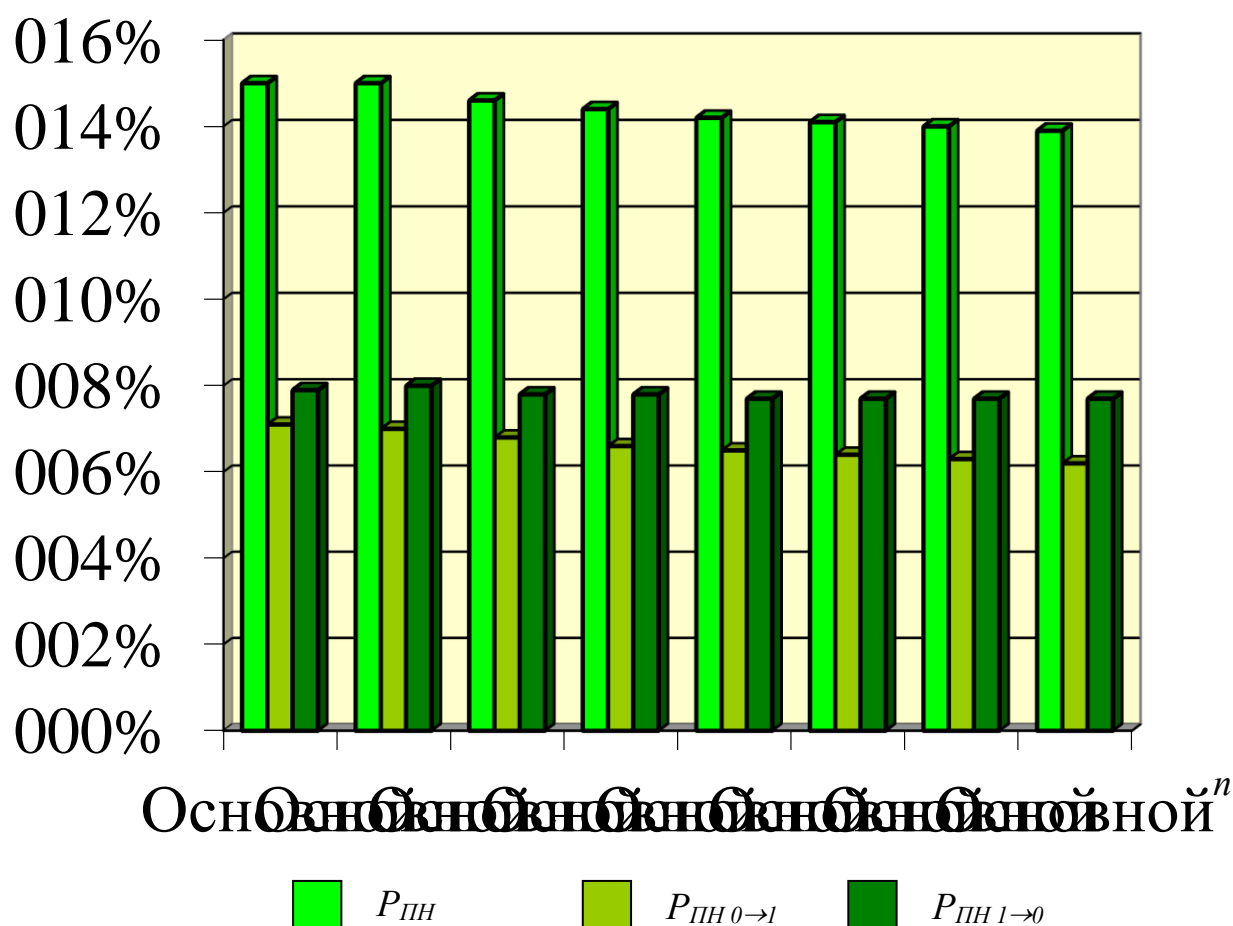


Рисунок 4.14 – Діаграми ймовірностей $P_{ПН}$, $P_{ПН 0 \rightarrow 1}$ та $P_{ПН 1 \rightarrow 0}$ пропуску несуттєвих помилок

Побудовані діаграми дають оцінку логарифмічному контролю матричного помножувача з фіксованою точкою, показуючи, що одержані при моделюванні ймовірності пропуску несуттєвих помилок незначно знижуються із зростанням розрядності операндів.

На рис. 4.15 показані діаграми зміни ймовірностей P_B , $P_{B0 \rightarrow 1}$ та $P_{B1 \rightarrow 0}$ виявлення помилок логарифмічним контролем для ряду значень розрядності n матричних помножувачів з фіксованою точкою, а також кількості вірних розрядів $n_T = n - 2$.

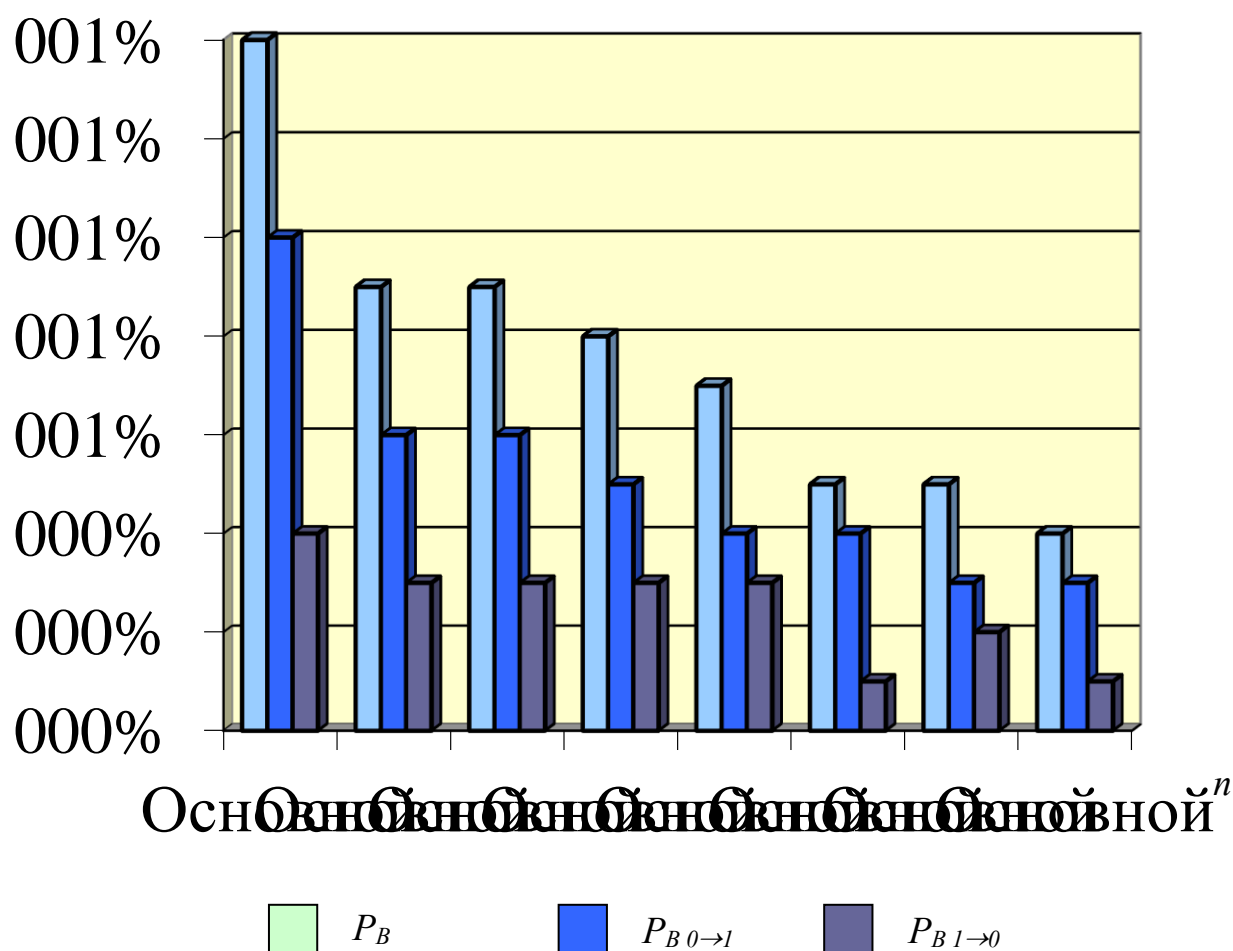


Рисунок 4.15 – Діаграми ймовірностей P_B , $P_{B0 \rightarrow 1}$ та $P_{B1 \rightarrow 0}$ виявлення помилок

Побудовані діаграми для оцінки логарифмічного контролю матричного помножувача з фіксованою точкою показують, що ймовірності виявлення помилок зменшуються з ростом розрядності операндів. Позитивна помилка виявляється з більшою ймовірністю в порівнянні з негативною помилкою.

На рис. 4.16 показані діаграми зміни ймовірностей P_{Π} , $P_{\Pi 0 \rightarrow 1}$ та $P_{\Pi 1 \rightarrow 0}$ пропуску помилок логарифмічним контролем для ряду значень розрядності n матричних помножувачів з фіксованою точкою, а також кількості вірних розрядів $n_T = n - 2$.

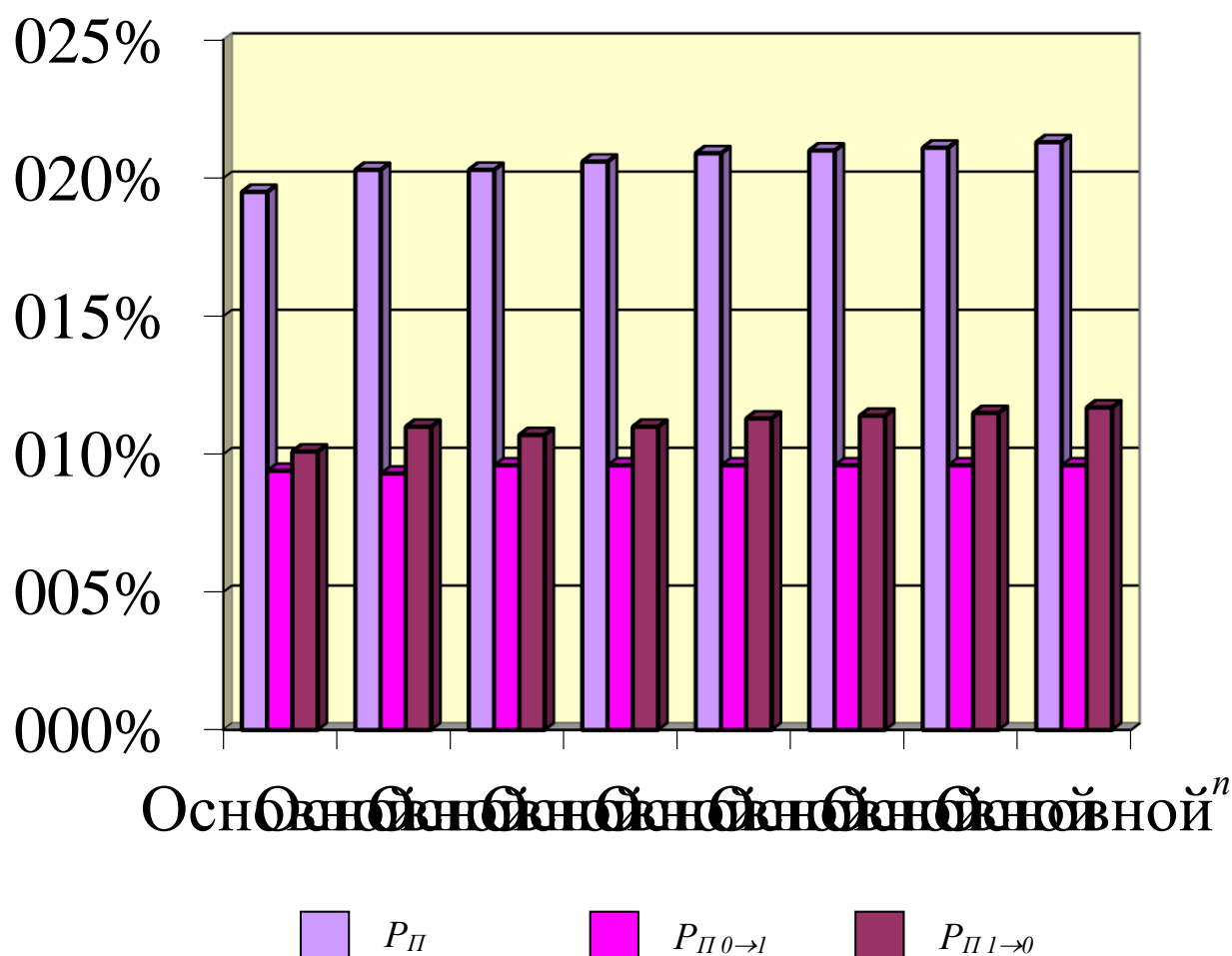


Рисунок 4.16 – Діаграми ймовірностей P_{Π} , $P_{\Pi 0 \rightarrow 1}$ та $P_{\Pi 1 \rightarrow 0}$ пропуску помилок

Побудовані діаграми для оцінки логарифмічного контролю матричного помножувача з фіксованою точкою показують, що ймовірності пропуску помилок і ймовірності пропуску негативних помилок збільшуються з ростом розрядності операндів. Позитивні помилки виявляється з однаковою ймовірністю, меншої в порівнянні з негативною помилкою.

На рис. 4.17 показані діаграми зміни достовірності $D_{\text{ЛОГ}}$ та $D_{\text{МОД}}$ логарифмічного контролю та контролю за числовим модулем для ряду значень розрядності n матричних помножувачів з фіксованою точкою та кількості вірних розрядів $n_T = n - 2$.

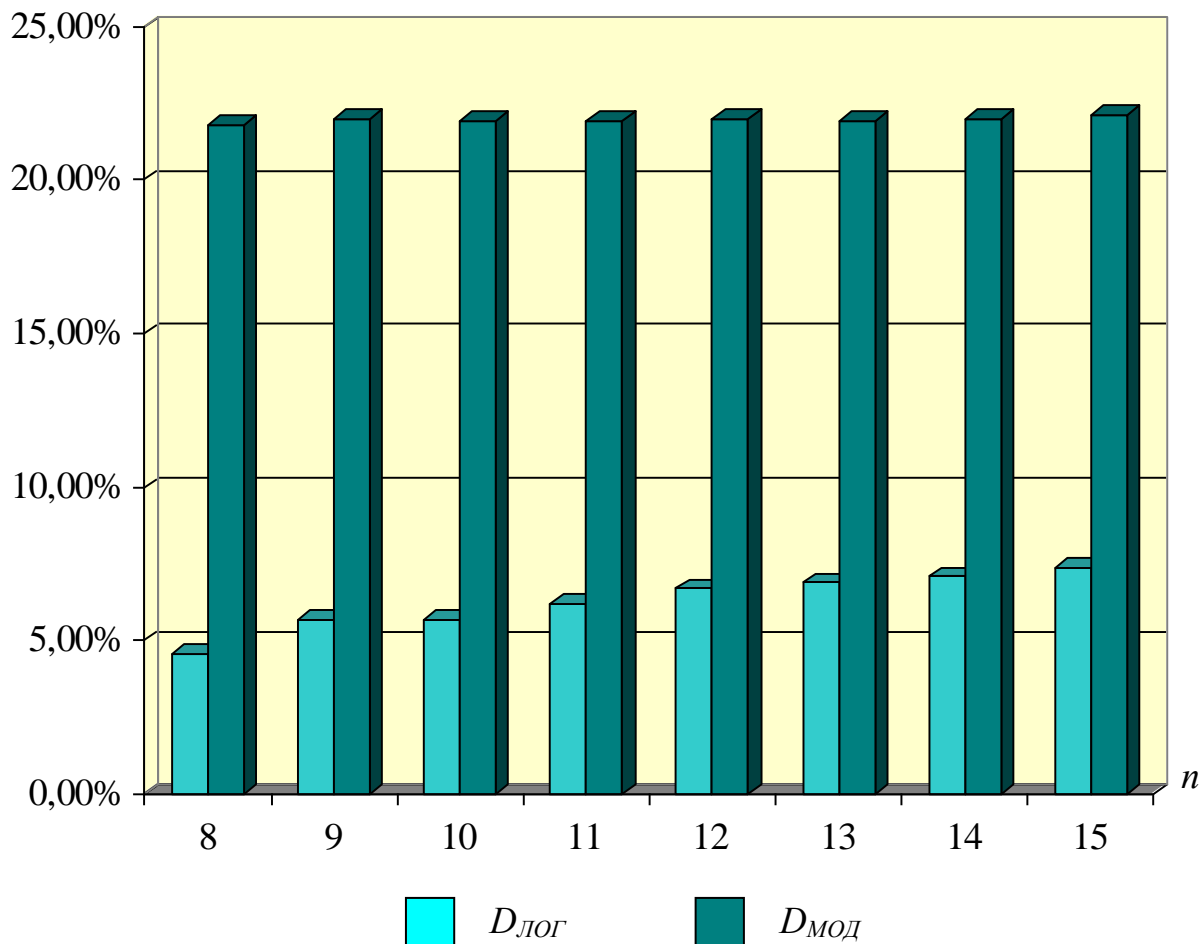


Рисунок 4.17 – Діаграми достовірності $D_{\text{ЛОГ}}$ та $D_{\text{МОД}}$ логарифмічного контролю та контролю за модулем

Побудовані діаграми для оцінки логарифмічного контролю матричного помножувача з фіксованою точкою показують, що достовірність $D_{\text{ЛОГ}}$ логарифмічного контролю значно менше за достовірність $D_{\text{МОД}}$ контролю за числовим модулем та зі зменшенням кількості вірних розрядів різниця між ними росте.

На рис. 4.18 показана діаграма зміни показника підвищення δD достовірності логарифмічного контролю в порівнянні з контролем за числовим модулем для ряду значень розрядності n матричних помножувачів та кількості вірних розрядів $n_T = n - 2$.

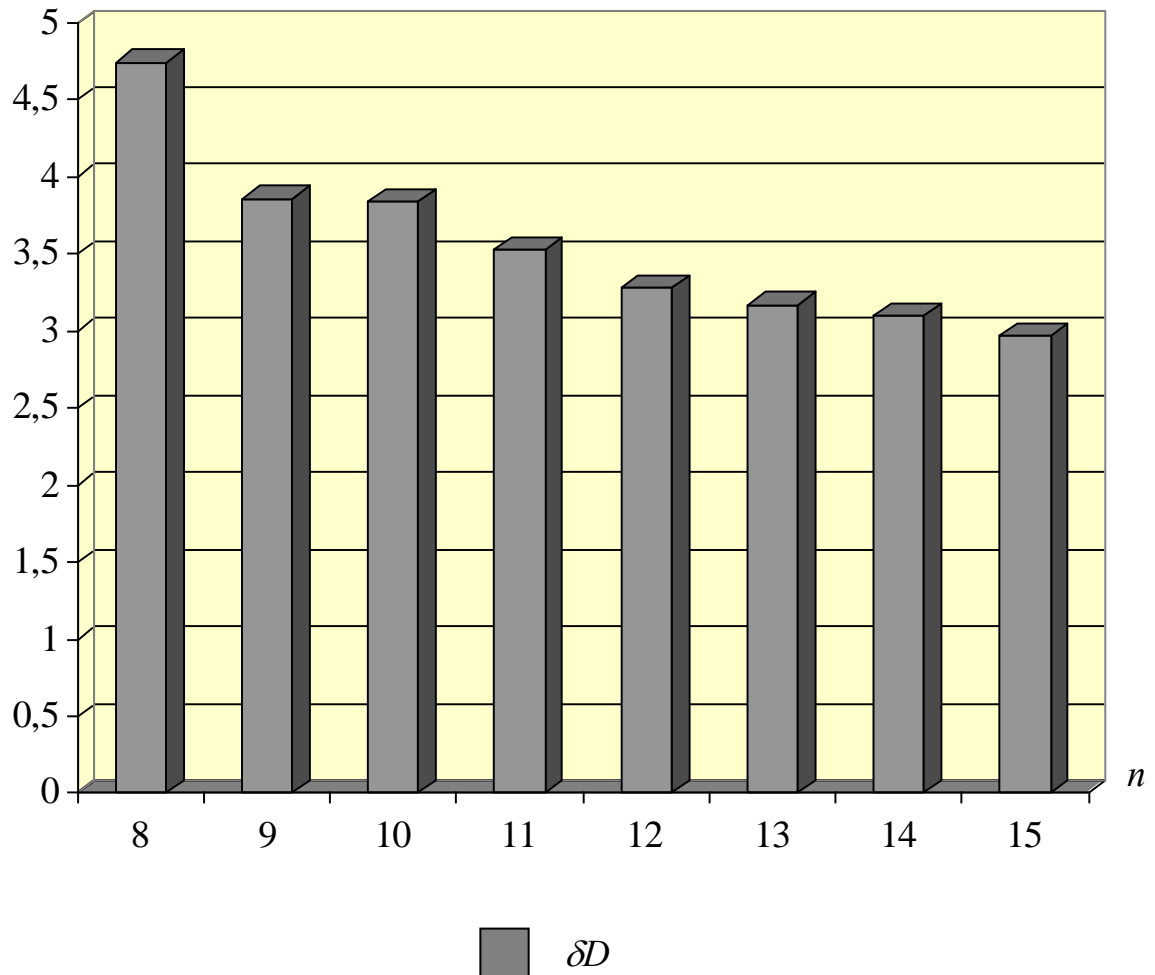


Рисунок 4.18 – Діаграми показника ефективності δD щодо достовірності логарифмічного контролю та контролю за модулем

Побудована діаграма для оцінки логарифмічного контролю матричного помножувача з фіксованою точкою показує більше високу достовірність логарифмічного контролю результатів і зниження показника δD з ростом розрядності операндів.

На рис. 4.19 показані діаграми зміни достовірності $D_{ЛОГ.n}$ та $D_{МОД.n}$ логарифмічного контролю та контролю за числовим модулем при кратних несправностях 1, 2, 3 для ряду значень розрядності n матричних помножувачів з фіксованою точкою та кількості вірних розрядів $n_T = n$.

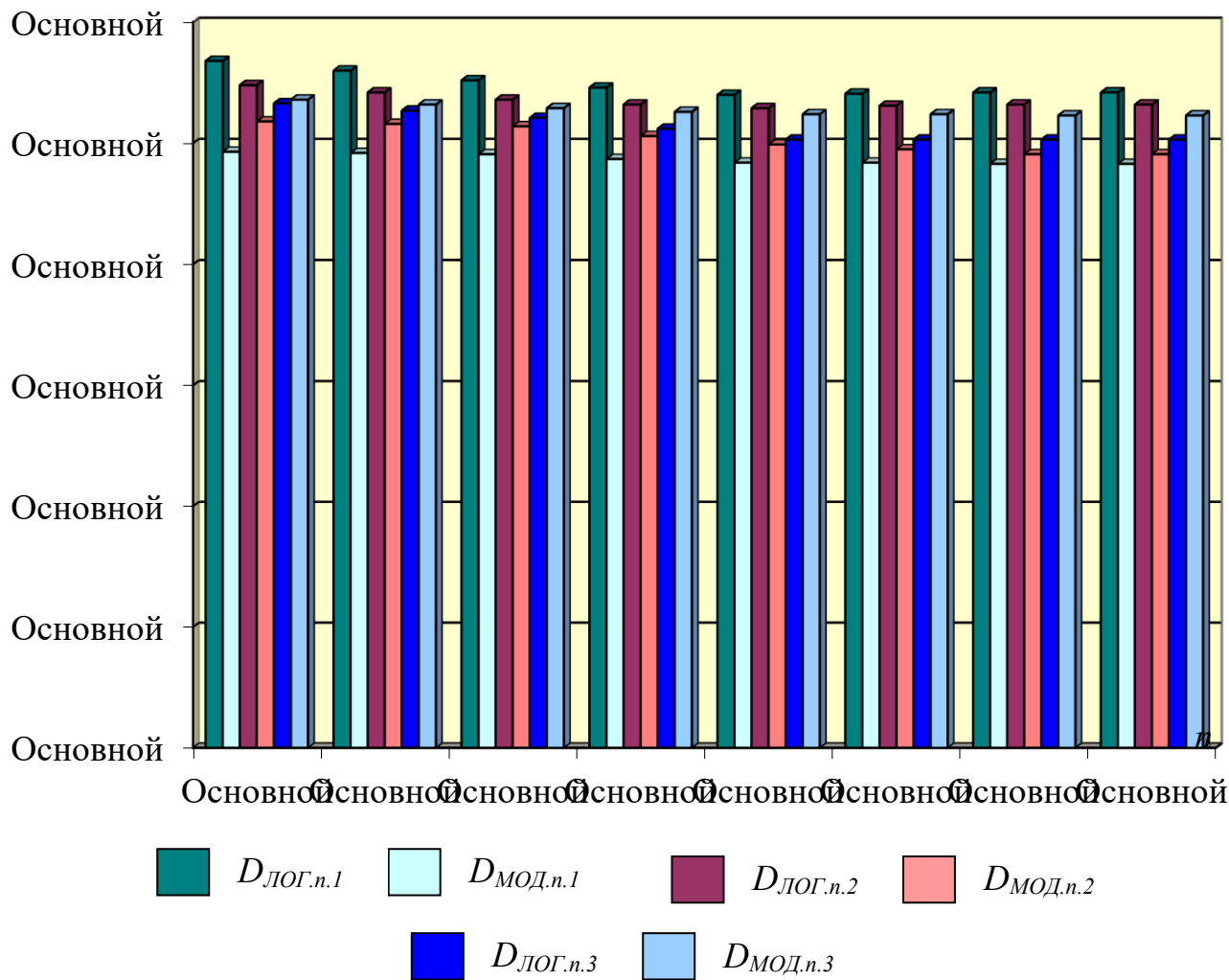


Рисунок 4.19 – Діаграми достовірності $D_{ЛОГ.n}$ та $D_{МОД.n}$ логарифмічного контролю та контролю за модулем під дією кратних несправностей

Побудовані діаграми для оцінки логарифмічного контролю матричного помножувача з фіксованою точкою показують, що достовірність $D_{ЛОГ.n}$ логарифмічного контролю та достовірність $D_{МОД.n}$ контролю за числовим модулем мало відрізняються та зі збільшенням кількості розрядів трохи зменшуються.

На рис. 4.20 показані діаграми зміни достовірності $D_{\text{ЛОГ.}n-4}$ та $D_{\text{МОД.}n-4}$ логарифмічного контролю та контролю за числовим модулем при кратних несправностях 1, 2, 3 для ряду значень розрядності n матричних помножувачів з фіксованою точкою та кількості вірних розрядів $n_T = n - 4$.

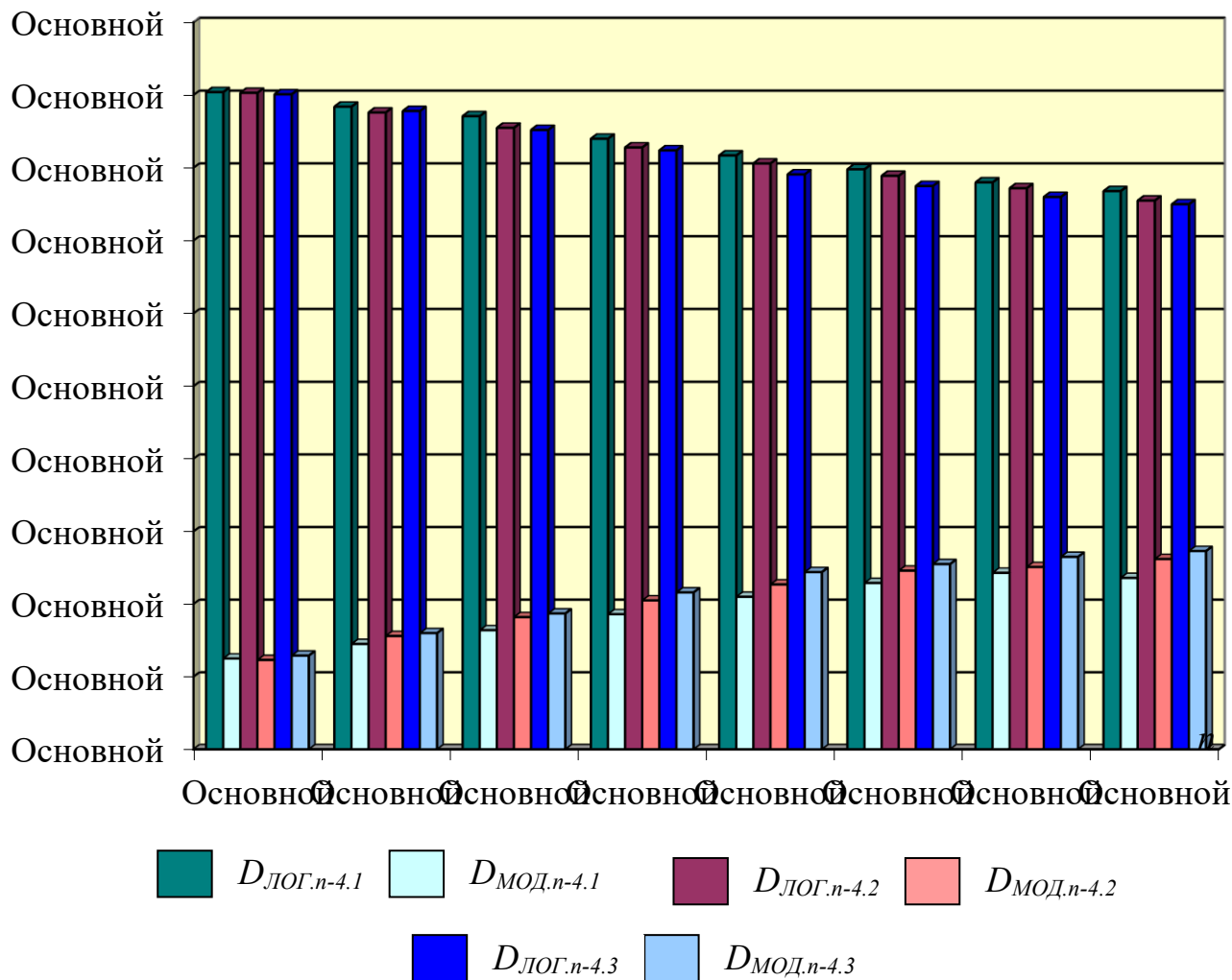


Рисунок 4.20 – Діаграми достовірності $D_{\text{ЛОГ.}n-4}$ та $D_{\text{МОД.}n-4}$ логарифмічного контролю та контролю за модулем під дією кратних несправностей

Побудовані діаграми для оцінки логарифмічного контролю матричного помножувача з фіксованою точкою показують, що достовірність $D_{\text{ЛОГ.}n-4}$ логарифмічного контролю суттєво перевищує достовірність $D_{\text{МОД.}n-4}$ контролю за числовим модулем. Їх значення відповідно зменшуються та збільшуються разом із зростанням кількості розрядів та кратності несправності.

4.3 Висновки

4.3.1 Проведені дослідження показали наступні особливості появи та виявлення логарифмічним контролем помилок різного виду:

- помилки виду $0 \rightarrow 1$ виникають в 1 – 8 разів рідше помилок виду $1 \rightarrow 0$;
- суттєві помилки виду $0 \rightarrow 1$ й $0 \rightarrow 1$ виникають у середньому з однаковою ймовірністю, а несуттєві помилки виду $0 \rightarrow 1$ виникають рідше несуттєвих помилок виду $1 \rightarrow 0$;
- помилки виду $0 \rightarrow 1$ виявляються в 2 – 3 рази частіше помилок виду $1 \rightarrow 0$;
- помилки виду $0 \rightarrow 1$ пропускаються на 3 – 6 рідше помилок виду $1 \rightarrow 0$;
- суттєві помилки виду $0 \rightarrow 1$ виявляються в 1,5 – 3 рази частіше суттєвих помилок виду $1 \rightarrow 0$, а несуттєві помилки виду $0 \rightarrow 1$ виявляються в середньому з тією же ймовірністю, що й несуттєві помилки виду $1 \rightarrow 0$;
- суттєві помилки виявляються з багаторазово більшою ймовірністю в порівнянні з несуттєвими помилками;
- імовірності виявлення суттєвих помилок зменшуються з ростом розрядності операндов;
- позитивні суттєві помилки виявляються з більшою ймовірністю в порівнянні з негативною помилкою;
- імовірності пропуску суттєвих помилок, суттєвих негативних помилок і несуттєвих помилок збільшуються з ростом розрядності операндів;
- позитивні суттєві помилки пропускаються з однаковою ймовірністю, меншою в порівнянні з негативною помилкою.
- імовірності виявлення несуттєвих помилок практично дорівнюють нулю.

4.3.2 Достовірність $D_{\text{ЛОГ},n-4}$ логарифмічного контролю суттєво зростає у порівнянні з достовірністю $D_{\text{МОД},n-4}$ контролю за числовим модулем при зменшенні кількості вірних розрядів. Ці достовірності відповідно зменшуються та збільшуються разом із зростанням кількості розрядів та кратності несправності.

ВИСНОВОК

Достовірність результатів обчислень є важливою характеристикою матричних пристроїв з фіксованою точкою, які належать до поширених та відіграють у сучасних комп'ютерних системах важливу роль в обробці наближених даних.

Достовірність наближених результатів обчислень оцінюється з використанням методів та засобів робочого діагностування, серед яких виділяється логарифмічний контроль, що оцінює величину помилки, що викликається несправностями цифрової схеми арифметичних матричних пристроїв.

Одним з основних показників методу робочого діагностування цифрових схем арифметичних пристроїв є достовірність контролю результатів, що обчислюються.

Для оцінки цього показника необхідно досліджувати ймовірності виявлення та пропуску суттєвих та несуттєвих помилок при впливі кратних несправностей діагностуємого ОП.

Сучасні системи моделювання не забезпечують гнучкого завдання несправностей схем та одержання статистики виявлення та пропуску суттєвих і несуттєвих помилок.

Серед матричних арифметичних пристроїв особливе місце посідає матричний помножувач, оскільки операція множення присутня у самому поданні наближеного даного в нормальній формі.

Вирішення цієї задачі для матричного помножувача зажадав розробки спеціальної програмної моделі логарифмічного контролю та її дослідження в умовах дії кратних несправностей.

Розроблено алгоритми формування контрольного коду операнда, перевірки контрольного співвідношення, а також оцінки ймовірностей виявлення суттєвих помилок.

Побудована за розробленими алгоритмами програмна модель логарифмічного контролю матричного помножувача з фіксованою точкою забезпечила одержання достовірної оцінки ймовірностей виявлення та пропуску суттєвих та несуттєвих, позитивних та негативних помилок на робочих послідовностях вхідних слів.

Проведені дослідження показали наступні закономірності:

- позитивні суттєві помилки виявляються з більшою ймовірністю в порівнянні з негативною помилкою;
- ймовірності виявлення суттєвих помилок логарифмічним контролем зменшуються із зростанням розрядності операндів матричного помножувача;
- ймовірності пропуску суттєвих помилок, суттєвих негативних помилок і несуттєвих помилок збільшуються з ростом розрядності операндів в матричному помножувачі;
- позитивні суттєві помилки пропускаються логарифмічним контролем з однаковою ймовірністю, що є меншої в порівнянні з негативною помилкою;
- показник ефективності, що визначається відношенням достовірності логарифмічного контролю та контролю за модулем матричного помножувача, вказує на високу достовірність логарифмічного контролю наближених результатів.

При зменшенні кількості вірних розрядів достовірність $D_{\text{ЛОГ},n-4}$ логарифмічного контролю суттєво зростає у порівнянні з достовірністю $D_{\text{МОД},n-4}$ контролю за числовим модулем. Ці достовірності відповідно зменшуються та збільшуються разом із зростанням кількості розрядів та кратності несправності.

ПЕРЕЛІК ДжЕРЕЛ ПОСИЛАНЬ

1. Щербаков Н. С. Достоверность работы цифровых устройств. – М.: Машиностроение, 1989. – 224 с.
2. Пархоменко П. П., Согомоян Е. С. и др. Основы технической диагностики. – М.: Энергия, 1981. – 320 с.
3. ДСТУ 2389-94. Технічне діагностування та контроль технічного стану. Терміни та визначення.
4. Рабочее диагностирование безопасных информационно-управляющих систем / А.В. Дрозд, В.С. Харченко, С.Г. Антощук, Ю.В. Дрозд, М.А. Дрозд, Ю.Ю. Сулима / Под ред. Дрозда А.В., Харченко В.С. – Х.: Нац. аэрокосмический ун-т им. Н.Е. Жуковского «ХАИ», 2012. – 614 с.
5. 13. Проектирование и диагностирование важных для безопасности встроенных систем. Практикум / А.В. Дрозд, В.С. Харченко, С. А. Нестеренко, С.Г. Антощук, М.А. Дрозд / Под ред. Дрозда А.В., Харченко В.С. – Х.: Нац. аэрокосмический ун-т им. Н.Е. Жуковского «ХАИ», 2013. – 111 с.
6. Локазюк В. М. Надійність, контроль, діагностика і модернізація ПК: Навчальний посібник для вузів / В. М. Локазюк, Ю. Г. Савченко / За ред. В. М. Локазюка. – Київ: “Академія”, 2004. - 376с.
7. Селлерс Ф. Методы обнаружения ошибок в работе ЭЦВМ. – М.: Мир, 1972. – 310 с.
8. Локазюк В. М., Поморова О. В., Домінов А. О. Інтелектуальне діагностування мікропроцесорних пристроїв та систем: Навч. посібник для вузів. – Хмельницький – Київ: “Такі справи”, 2001. – 286 с.
9. Дрозд А. В. Использование логарифмического контроля для обнаружения отказов арифметических устройств // Вісн. НТУУ «КПІ». Інф., упр. та обчисл. техніка. – К., 1998. – Вип. 31. – С. 224 – 231.

10. Дрозд А.В., Аль-Аззех Р. Логарифмический контроль приближенных вычислений в форматах с фиксированной точкой // Тр. Одес. политехн. ун-та. – Одесса, 2004. – Вып. 1 (21). – С. 86 – 92.
11. ANSI/IEEE Std 754-1985. IEEE Standard for Binary Floating-Point Arithmetic. IEEE, New York, USA, 1985. – 18 с.
12. Самофалов К. Г., Романкевич А. М., Валуйский В. Н., Каневский Ю. С., Пиневиц М. М. Прикладная теория цифровых автоматов. – Киев: Вища школа, 1987. – 375 с.
13. Журавлев Ю. П., Котелюк Л. А., Циклинский Н. И. Надежность и контроль ЭВМ. – М.: Советское радио, 1978. – 416 с.
14. Drozd A. On-line testing of computing circuits at approximate data processing // Радіоелектроніка та інформатика. 2003. № 3. – С. 113 – 116.
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 ІХ Міжнародної науково-практичної конференції «Інформаційні управляючі системи і технології» (ІУСТОДЕСА-2021). – Одеса, 2021. – С. 109 – 112.

ДОДАТОК А

ТЕКСТ ПРОГРАМНОЇ МОДЕЛІ ЛОГАРИФМІЧНОГО КОНТРОЛЮ
МАТРИЧНОГО ПОМНОЖУВАЧА

```
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;
```

```
Panel1: TPanel;
```

```
Button5: TButton;
```

```
Button6: TButton;
```

```
Edit1: TEdit;
```

```
Edit2: TEdit;
```

```
Panel2: TPanel;
```

```
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;
```

```
Panel5: TPanel;
```

```
Panel6: TPanel;
```

```
Panel7: TPanel;
```

```
Button8: TButton;
```

```
Button9: TButton;
```

```
Button10: TButton;
```

```
Button11: TButton;
```

```
Panel8: TPanel;
```

```
Panel9: TPanel;
```

```
Panel10: TPanel;
```

```
Edit5: TEdit;
```

```
Edit6: TEdit;
```

```
Panel11: TPanel;
```

```
Edit7: TEdit;
```

```
N8: TMenuItem;
```

```
N9: TMenuItem;
```

```
N10: TMenuItem;
```

```
Edit8: TEdit;
```

```
Edit9: TEdit;
```

```
Button12: TButton;
```

```

Button13: TButton;
Button14: TButton;
Image2: TImage;
Panel12: TPanel;
Panel13: TPanel;
Panel14: 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 Button8Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure Button11Click(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 Button13Click(Sender: TObject);
procedure Button12Click(Sender: TObject);
procedure Button14Click(Sender: TObject);
procedure FormDragDrop(Sender, Source: TObject; X, Y: Integer);
procedure FormDragOver(Sender, Source: TObject; X, Y: Integer;
  State: TDragState; var Accept: Boolean);
procedure Panel12MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Panel14Click(Sender: TObject);
procedure Panel13Click(Sender: TObject);

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

const
  n=10; nf=11;
var
  Form1: TForm1;
  Ma:array[1..n,1..n-1]of integer;
  Mb:array[1..n,1..n-1]of integer;
  Mc:array[1..n,1..n-1]of integer;
  Ma1:array[1..n,1..n-1]of integer;
  Mb1:array[1..n,1..n-1]of integer;
  Ma2:array[1..n-1]of integer;
  Mb2:array[1..n-1]of integer;
  Ms:array[1..n,1..n-1]of integer;
  Mp:array[1..n,1..n-1]of integer;
  AAb:array[1..n]of integer;
  BBb:array[1..n]of integer;
  PPb:array[1..2*n]of integer;

```

```

PPE:array[1..2*n]of integer;
MF:array[1..nf]of string;

```

```

aa,bb,pp,pe,ca,cb,cc,ce,mo,xm,ym:integer;
za,zb,f1,f2,ff,fr,df1,df2,nfe,nfr,hh:integer;
ka,az,kb,bz,kp,kpe,kpp,ep,pp1,pe1,ll,es:integer;
de,ne,dn,nn,rde,rne,rdn,rnn :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
  MF[1]:='a'; MF[2]:='b'; MF[3]:='c';
  MF[4]:='s'; MF[5]:='p'; MF[6]:='a1';
  MF[7]:='b1'; MF[8]:='a2'; MF[9]:='b2';
  MF[10]:='0'; MF[11]:='1';
  f1:=1; f2:=2; df1:=0; df2:=0; fr:=21; mo:=0; es:=0;
  hh:=32*n-6; nfr:=0;
  Image1.Left:=8;      Image1.Top:=76;
  Image1.Width:=32*(2*n+1); Image1.Height:=32*(n+3);
  Image1.Canvas.Rectangle(32*n+4,4,64*n-4,28);
  Image1.Canvas.Rectangle(64*n+4,36,64*n+28,32*n+28);
  Image1.Canvas.Rectangle(4,32*n+36,64*n-4,32*n+60);
  Image1.Canvas.Rectangle(4,32*n+68,64*n-4,32*n+92);
  Button3.Top:=56;    Button3.Left:=32*n-20;
  Button4.Top:=56;    Button4.Left:=64*n+12;
  Panel1.Top:=32*n+88; Panel1.Left:=64*n+12;
  Panel2.Top:=32*n+120; Panel2.Left:=64*n+12;
  Panel12.Top:=32*n+152; Panel12.Left:=32*n+10;
  Panel12.Width:=4;
  Button1.Visible:=False;
  Button3.Visible:=True; Button4.Visible:=True;
  Button5.Visible:=True; Button6.Visible:=True;
  Button7.Visible:=True; Button8.Visible:=True;
  Button9.Visible:=True; Button10.Visible:=True;
  Button11.Visible:=True; Button12.Visible:=True;
  Button13.Visible:=True; Button14.Visible:=True;
  Panel1.Visible:=True; Panel2.Visible:=True;
  Panel3.Visible:=True; Panel4.Visible:=True;
  Panel5.Visible:=True; Panel6.Visible:=True;
  Panel7.Visible:=True; Panel8.Visible:=True;
  Panel9.Visible:=True; Panel10.Visible:=True;
  Panel11.Visible:=True; Panel12.Visible:=True;
  Panel13.Visible:=True; Panel14.Visible:=True;
  Edit1.Visible:=True; Edit2.Visible:=True;
  Edit3.Visible:=True; Edit4.Visible:=True;
  Edit5.Visible:=True; Edit6.Visible:=True;
  Edit7.Visible:=True; Edit8.Visible:=True;
  Edit9.Visible:=True;

  for i:=1 to n do
    for j:=1 to n-1 do
      begin
        h1:=64*n-32*(i+j)+8;
        with Image1.Canvas do

```

```

begin
  Brush.Color:=clSilver;
  Rectangle(h1,32*i+8,h1+16,32*i+24);
  TextOut(h1+1,32*i+9,IntToStr(i*10+j+1));
  TextOut(h1+1,32*i+9,IntToStr(i)+IntToStr(j+1));
end;
end;
for i:=1 to n do
begin
  h1:=64*n-32*i+8;
  AAb[i]:=0;
  BBb[i]:=0;
  with Image1.Canvas do
  begin
    Pen.Color:=clBlue;
    Brush.Color:=clMoneyGreen;
    Rectangle(h1,8,h1+16,24);
    TextOut(h1+4,9,'0');
    Rectangle(64*n+8,32*i+8,64*n+24,32*i+24);
    TextOut(64*n+12,32*i+9,'0');
    Rectangle(h1,32*n+40,h1+16,32*n+56);
    TextOut(h1+4,32*n+41,'0');
    Rectangle(h1-32*n,32*n+40,h1+16-32*n,32*n+56);
    TextOut(h1+4-32*n,32*n+41,'0');
    Rectangle(h1,32*n+72,h1+16,32*n+88);
    TextOut(h1+4,32*n+73,'0');
    Rectangle(h1-32*n,32*n+72,h1+16-32*n,32*n+88);
    TextOut(h1+4-32*n,32*n+73,'0');
  end;
end;

end;

procedure TForm1.Button5Click(Sender: TObject);
var h1,ih:integer;
begin
  Edit1.Color:=clWindow;
  aa:=StrToInt(Edit1.Text);
  if aa>=(1 shl n) then
    Edit1.Color:=clRed
  else
  begin
    h1:=1;
    for ih:=1 to n do
    begin
      AAb[ih]:=(aa and h1)shr(ih-1);
      h1:=h1 shl 1;
      Image1.Canvas.TextOut(64*n-32*ih+12,9,
        IntToStr(AAb[ih]));
    end;
  end;
end;

procedure TForm1.Button6Click(Sender: TObject);
var h1,ih:integer;
begin
  Edit2.Color:=clWindow;
  bb:=StrToInt(Edit2.Text);
  if bb>=(1 shl n) then
    Edit2.Color:=clRed
  else
  begin

```

```

h1:=1;
for ih:=1 to n do
begin
  BBb[ih]:=(bb and h1)shr(ih-1);
  h1:=h1 shl 1;
  Image1.Canvas.TextOut(64*n+12,32*ih+9,
    IntToStr(BBb[ih]));
end;
end;
end;

procedure TForm1.OnMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
  var ii,jj,xx,yy,h1:integer;
begin
  ii:=(X-8)div 32;
  jj:=(Y-8)div 32;
  xx:=ii*32+8;
  yy:=jj*32+8;

  if (X-xx<17)and(Y-yy<17)and(yy=8)and
    (ii<2*n)and(ii>=n) then
  begin //инвертирование бит числа A
    h1:=2*n-ii;
    AAb[h1]:=1-AAb[h1];
    Image1.Canvas.TextOut(64*n-32*h1+12,9,
      IntToStr(AAb[h1]));
  end;
  if (X-xx<17)and(Y-yy<17)and(xx=64*n+8)and
    (jj<=n)and(jj>0) then
  begin //инвертирование бит числа B
    BBb[jj]:=1-BBb[jj];
    Image1.Canvas.TextOut(64*n+12,32*jj+9,
      IntToStr(BBb[jj]));
  end;
end;

procedure TForm1.Button3Click(Sender: TObject);
  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);
  var ih:integer;
begin
  bb:=0;
  for ih:=1 to n do
    bb:=bb+BBb[ih]shl(ih-1);
  Edit2.Text:=IntToStr(bb);
end;

procedure TForm1.Button7Click(Sender: TObject);
  var h1,h2,h4,h5,ih,i,j:integer;
begin //Старт
  if Button7.Caption='Старт'then
  begin
  if ((za>1)or(zb>1))then Button7.Caption:='Финиш';
  Randomize; Edit8.Text:=IntToStr(fr);

```

```

h2:=1 shl n;
cc:=0; ce:=0;
de:=0; ne:=0; dn:=0; nn:=0;
Edit3.Text:='0';
Edit4.Text:='0';
ca:=1-(za shr 1);
if za=2 then aa:=0;
repeat //перебор значений числа A
cb:=1-(zb shr 1);
if za mod 2=1 then aa:=Random (h2);
Edit1.Text:=IntToStr(aa);
  h1:=1;
  for ih:=1 to n do
  begin
  AAb[ih]:=(aa and h1)shr(ih-1);
  h1:=h1 shl 1;
  Image1.Canvas.TextOut(64*n-32*ih+12,9,
    IntToStr(AAb[ih]));
  end;
ka:=0; while (aa shr ka >0) do ka:=ka+1;
if aa=0 then az:=0 else az:=1;
if zb=2 then bb:=0;
repeat //перебор значений числа B
if zb mod 2=1 then bb:=Random (h2);
Edit2.Text:=IntToStr(bb);
  h1:=1;
  for ih:=1 to n do
  begin
  BBb[ih]:=(bb and h1)shr(ih-1);
  h1:=h1 shl 1;
  Image1.Canvas.TextOut(64*n+12,32*ih+9,
    IntToStr(BBb[ih]));
  end;
pp:=aa*bb;
kpp:=0; while (pp shr kpp >0) do kpp:=kpp+1;
kb:=0; while (bb shr kb >0) do kb:=kb+1;
if bb=0 then bz:=0 else bz:=1;
kp:=ka*bz+kb*az;
ll:=ka; if kb>ka then ll:=kb;
Edit3.Text:=IntToStr(pp);
h1:=1;
for ih:=1 to 2*n do
begin
  PPb[ih]:=(pp and h1)shr(ih-1);
  h1:=h1 shl 1;
  Image1.Canvas.TextOut(64*n-32*ih+12,32*n+41,
    IntToStr(PPb[ih]));
end;
if nfe=0 then
begin //задание произвольного неисправного элемента
  h4:=Random(n)+1; h5:=Random(n-1)+1;
end
else
begin
  h4:=nfe div 10; h5:=(nfe mod 10)-1;
end;
Edit9.Text:=IntToStr(h4*10+h5+1);

if nfr=1 then
begin //задание произвольной неисправности
  repeat
  f1:=Random(11)+1;

```



```

f2:=Random(11)+1;
until not((f1=f2)or
((MF[f1]='0')and(MF[f2]='1'))or
((MF[f1]='1')and(MF[f2]='0'))or
((MF[f1]='0')and(MF[f2]='0'))or
((MF[f1]='1')and(MF[f2]='1')));
Panel6.Caption:=MF[f1];
Panel7.Caption:=MF[f2];
if f1<f2 then fr:=f2*10+f1 else fr:=f1*10+f2;
end;
for i:=1 to n do
for j:=1 to n-1 do
begin
if i=1 then
begin
Ma2[j]:=AAb[j+1];
Mb2[j]:=BBb[1];
if ff=1 then
begin
if fr=108 then Ma2[h5]:=0;
if fr=118 then Ma2[h5]:=1;
if fr=109 then Mb2[h5]:=0;
if fr=119 then Mb2[h5]:=1;
if fr=89 then
begin
Ma2[h5]:=Ma2[h5]*Mb2[h5]; Mb2[h5]:=Ma2[h5];
end;
end;
Ma[1,j]:=Ma2[j]*Mb2[j];
end
else
if j=n-1 then Ma[i,j]:=Mp[i-1,n-1]
else Ma[i,j]:=Ms[i-1,j+1];
Ma1[i,j]:=AAb[i];
Mb1[i,j]:=BBb[j+1];
if ff=1 then
begin
if fr=106 then Ma[h4,h5]:=0;
if fr=116 then Ma[h4,h5]:=1;
if fr=107 then Mb[h4,h5]:=0;
if fr=117 then Mb[h4,h5]:=1;
if fr=67 then
begin
Ma1[h4,h5]:=Ma1[h4,h5]*Mb1[h4,h5]; Mb1[h4,h5]:=Ma1[h4,h5];
end;
end;
Mc[i,j]:=Ma1[i,j]*Mb1[i,j];
if j>1 then Mb[i,j]:=Mp[i,j-1]
else Mb[i,j]:=0;
if ff=1 then
begin
if fr=101 then Ma[h4,h5]:=0;
if fr=111 then Ma[h4,h5]:=1;
if fr=102 then Mb[h4,h5]:=0;
if fr=112 then Mb[h4,h5]:=1;
if fr=103 then Mc[h4,h5]:=0;
if fr=113 then Mc[h4,h5]:=1;
if fr=21 then
begin
Ma[h4,h5]:=Ma[h4,h5]*Mb[h4,h5]; Mb[h4,h5]:=Ma[h4,h5];
end;
if fr=31 then

```

```

begin
  Ma[h4,h5]:=Ma[h4,h5]*Mc[h4,h5]; Mc[h4,h5]:=Ma[h4,h5];
end;
if fr=32 then
begin
  Mb[h4,h5]:=Mb[h4,h5]*Mc[h4,h5]; Mb[h4,h5]:=Mc[h4,h5];
end;
if fr=61 then
begin
  Ma[h4,h5]:=Ma[h4,h5]*Ma1[h4,h5]; Ma1[h4,h5]:=Ma[h4,h5];
end;
if fr=62 then
begin
  Mb[h4,h5]:=Mb[h4,h5]*Ma1[h4,h5]; Ma1[h4,h5]:=Mb[h4,h5];
end;
if fr=63 then
begin
  Mc[h4,h5]:=Mc[h4,h5]*Ma1[h4,h5]; Ma1[h4,h5]:=Mc[h4,h5];
end;
if fr=71 then
begin
  Ma[h4,h5]:=Ma[h4,h5]*Mb1[h4,h5]; Mb1[h4,h5]:=Ma[h4,h5];
end;
if fr=72 then
begin
  Mb[h4,h5]:=Mb[h4,h5]*Mb1[h4,h5]; Mb1[h4,h5]:=Mb[h4,h5];
end;
if fr=73 then
begin
  Mc[h4,h5]:=Mc[h4,h5]*Mb1[h4,h5]; Mb1[h4,h5]:=Mc[h4,h5];
end;
if (fr=81)and(i=1) then
begin
  Ma[h4,h5]:=Ma[h4,h5]*Ma2[h5]; Ma2[h5]:=Ma[h4,h5];
end;
if (fr=82)and(i=1) then
begin
  Mb[h4,h5]:=Mb[h4,h5]*Ma2[h5]; Ma2[h5]:=Mb[h4,h5];
end;
if (fr=83)and(i=1) then
begin
  Mc[h4,h5]:=Mc[h4,h5]*Ma2[h5]; Ma2[h5]:=Mc[h4,h5];
end;
if (fr=91)and(i=1) then
begin
  Ma[h4,h5]:=Ma[h4,h5]*Mb2[h5]; Mb2[h5]:=Ma[h4,h5];
end;
if (fr=92)and(i=1) then
begin
  Mb[h4,h5]:=Mb[h4,h5]*Mb2[h5]; Mb2[h5]:=Mb[h4,h5];
end;
if (fr=93)and(i=1) then
begin
  Mc[h4,h5]:=Mc[h4,h5]*Mb2[h5]; Mb2[h5]:=Mc[h4,h5];
end;
end;
Ms[i,j]:=Ma[i,j] xor Mb[i,j] xor Mc[i,j];
Mp[i,j]:=Ma[i,j]*Mb[i,j] or Ma[i,j]*Mc[i,j] or Mb[i,j]*Mc[i,j];
if ff=1 then
begin
if fr=104 then Ms[h4,h5]:=0;
if fr=114 then Ms[h4,h5]:=1;

```

```

if fr=105 then Ms[h4,h5]:=0;
if fr=115 then Ms[h4,h5]:=1;
if fr=41 then
begin
  Ms[h4,h5]:=Ms[h4,h5]*Ma[h4,h5]; Ma[h4,h5]:=Ms[h4,h5];
end;
if fr=42 then
begin
  Ms[h4,h5]:=Ms[h4,h5]*Mb[h4,h5]; Mb[h4,h5]:=Ms[h4,h5];
end;
if fr=43 then
begin
  Ms[h4,h5]:=Ms[h4,h5]*Mc[h4,h5]; Mc[h4,h5]:=Ms[h4,h5];
end;
if fr=51 then
begin
  Mp[h4,h5]:=Mp[h4,h5]*Ma[h4,h5]; Ma[h4,h5]:=Mp[h4,h5];
end;
if fr=52 then
begin
  Mp[h4,h5]:=Mp[h4,h5]*Mb[h4,h5]; Mb[h4,h5]:=Mp[h4,h5];
end;
if fr=53 then
begin
  Mp[h4,h5]:=Mp[h4,h5]*Mc[h4,h5]; Mc[h4,h5]:=Mp[h4,h5];
end;
if fr=54 then
begin
  Mp[h4,h5]:=Mp[h4,h5]*Ms[h4,h5]; Ms[h4,h5]:=Mp[h4,h5];
end;
if fr=64 then
begin
  Ms[h4,h5]:=Ms[h4,h5]*Ma1[h4,h5]; Ma1[h4,h5]:=Ms[h4,h5];
end;
if fr=74 then
begin
  Ms[h4,h5]:=Ms[h4,h5]*Mb1[h4,h5]; Mb1[h4,h5]:=Ms[h4,h5];
end;
if (fr=84)and(i=1) then
begin
  Ms[h4,h5]:=Ms[h4,h5]*Ma2[h5]; Ma2[h5]:=Ms[h4,h5];
end;
if (fr=94)and(i=1) then
begin
  Ms[h4,h5]:=Ms[h4,h5]*Mb2[h5]; Mb2[h5]:=Ms[h4,h5];
end;
if fr=65 then
begin
  Mp[h4,h5]:=Mp[h4,h5]*Ma1[h4,h5]; Ma1[h4,h5]:=Mp[h4,h5];
end;
if fr=75 then
begin
  Mp[h4,h5]:=Mp[h4,h5]*Mb1[h4,h5]; Mb1[h4,h5]:=Mp[h4,h5];
end;
if (fr=85)and(i=1) then
begin
  Mp[h4,h5]:=Mp[h4,h5]*Ma2[h5]; Ma2[h5]:=Mp[h4,h5];
end;
if (fr=95)and(i=1) then
begin
  Mp[h4,h5]:=Mp[h4,h5]*Mb2[h5]; Mb2[h5]:=Mp[h4,h5];
end;

```

```

    end;
    end;
    PPe[1]:=AAb[1]*BBb[1];
    for ih:=1 to n-1 do
    begin
        PPe[ih+1]:=Ms[ih,1];
        PPe[n+ih]:=Ms[n,ih];
    end;
    PPe[2*n]:=Mp[n,n-1];
    pe:=0;
    for ih:=1 to 2*n do
    begin
        pe:=pe+PPe[ih]shl(ih-1);
        Image1.Canvas.TextOut(64*n-32*ih+12,32*n+73,
            IntToStr(PPe[ih]));
    end;
    pe1:=0;
    for ih:=n+1 to 2*n do
    pe1:=pe1+PPe[ih]shl(ih-1);
    pp1:=0;
    for ih:=n+1 to 2*n do
        pp1:=pp1+PPb[ih]shl(ih-1);
    kpe:=0; while (pe shr kpe >0) do kpe:=kpe+1;
    if (kp=kpe)or(kp=kpe+1) then ep:=0 else ep:=1;
    Edit4.Text:=IntToStr(pe);
    Edit5.Text:=IntToStr(pp-pe);
    cc:=cc+1;
    if pp<>pe then ce:=ce+1;
    Edit6.Text:=IntToStr(ce);
    Edit7.Text:=IntToStr((ce*100)div cc);
    if (abs(pp-pe)>=(1shl(ll+es)))and(ep=0) then ne:=ne+1;
    rne:=(ne*2000 div cc+1)div 2;
    if (abs(pp-pe)>=(1shl(ll+es)))and(ep=1) then de:=de+1;
    rde:=(de*2000 div cc+1)div 2;
    if (pp<>pe)and(abs(pp-pe)<(1shl(ll+es)))and(ep=0) then nn:=nn+1;
    rnn:=(nn*2000 div cc+1)div 2;
    if (pp<>pe)and(abs(pp-pe)<(1shl(ll+es)))and(ep=1) then dn:=dn+1;
    rdn:=(dn*2000 div cc+1)div 2;
    if (ep=1)and(Panel13.Color=clAqua) then
    begin
        Panel13.Color:=clRed;
        repeat
            Application.ProcessMessages;
        until Panel13.Color=clAqua;
    end;
    Application.ProcessMessages;
    if (za=3)and(zb>1) then cb:=1;
    if zb=2 then
    begin
        bb:=bb+1;
        if bb=h2 then cb:=1;
    end;
    until cb=1;
    if za=2 then
    begin
        aa:=aa+1;
        if aa=h2 then ca:=1;
    end;
    until ca=1;
    end
    else
    begin

```

```

Button7.Caption:='Старт';
ca:=1; cb:=1;
end;
end;

```

```

procedure TForm1.Button8Click(Sender: TObject);
begin
repeat
if df1=0 then
if f1<nf then f1:=f1+1
else f1:=1
else
if f1>1 then f1:=f1-1
else f1:=nf;
until not((f1=f2)or
((MF[f1]='0')and(MF[f2]='1'))or
((MF[f1]='1')and(MF[f2]='0')));
Panel6.Caption:=MF[f1];
fr:=f1*10+f2;
if f1<f2 then fr:=f2*10+f1;
end;

```

```

procedure TForm1.Button9Click(Sender: TObject);
begin
repeat
if df2=0 then
if f2<nf then f2:=f2+1
else f2:=1
else
if f2>1 then f2:=f2-1
else f2:=nf;
until not((f1=f2)or
((MF[f1]='0')and(MF[f2]='1'))or
((MF[f1]='1')and(MF[f2]='0')));
Panel7.Caption:=MF[f2];
fr:=f1*10+f2;
if f1<f2 then fr:=f2*10+f1;
end;

```

```

procedure TForm1.Button10Click(Sender: TObject);
begin
df1:=1-df1;
if df1=0 then Button8.Caption:='>'
else Button8.Caption:='<';
end;

```

```

procedure TForm1.Button11Click(Sender: TObject);
begin
df2:=1-df2;
if df2=0 then Button9.Caption:='>'
else Button9.Caption:='<';
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.Button13Click(Sender: TObject);
begin
    nfe:=0; Edit9.Text:=' ';
end;

procedure TForm1.Button12Click(Sender: TObject);
begin
    nfe:=StrToInt(Edit9.Text);
    if (nfe<=n*11)and(nfe>=12) then
        Edit9.Color:=clWindow
    else Edit9.Color:=clRed;
end;

procedure TForm1.Button14Click(Sender: TObject);
var i,j,h1:integer;
begin
    mo:=1-mo;
    if mo=0 then
        begin
```

```

Image1.Visible:=True;
Image2.Visible:=False;
Button14.Caption:='Умножитель';
Button3.Top:=56;   Button3.Left:=32*n-20;
Button3.Width:=25; Button3.Height:=25;
Button4.Top:=56;   Button4.Left:=64*n+12;
Button4.Width:=25; Button4.Height:=25;
Panel1.Top:=32*n+88; Panel1.Left:=64*n+12;
Panel1.Width:=25;  Panel1.Height:=25;
Panel2.Top:=32*n+120; Panel2.Left:=64*n+12;
Panel2.Width:=25;  Panel2.Height:=25;
end
else
begin
Image1.Visible:=False;
Image2.Visible:=True;
Button14.Caption:='Логарифм κ';
Image2.Left:=8;   Image2.Top:=52;
Image2.Width:=32*(2*n+1); Image2.Height:=32*(n+3);
Image2.Canvas.Rectangle(48*n-6,6,64*n-2,26);
Image2.Canvas.Rectangle(48*n-6,30,64*n-2,50);
Image2.Canvas.Rectangle(32*n-6,54,64*n-2,74);
Image2.Canvas.Rectangle(32*n-6,78,64*n-2,98);

Image2.Canvas.Brush.Color:=clWhite;
Image2.Canvas.Pen.Color:=clWhite;
Image2.Canvas.Rectangle(hh,103,64*n-2,113);
Image2.Canvas.Brush.Color:=clSkyBlue;
Image2.Canvas.Pen.Color:=clBlack;
hh:=16*(4*n-es-l)-6;
Image2.Canvas.Rectangle(hh,103,64*n-2,113);

Button3.Top:=58;   Button3.Left:=64*n+14;
Button3.Width:=21; Button3.Height:=21;
Button4.Top:=82;   Button4.Left:=64*n+14;
Button4.Width:=21; Button4.Height:=21;
Panel1.Top:=106;   Panel1.Left:=64*n+14;
Panel1.Width:=21;  Panel1.Height:=21;
Panel2.Top:=130;   Panel2.Left:=64*n+14;
Panel2.Width:=21;  Panel2.Height:=21;
for i:=1 to n do
begin
h1:=64*n-16*i-4;
with Image2.Canvas do
begin
if i<=ka then Brush.Color:=clLime
else Brush.Color:=clSilver;
Rectangle(h1,8,h1+16,24);
TextOut(h1+4,9,IntToStr(AAb[i]));
if i<=kb then Brush.Color:=clLime
else Brush.Color:=clSilver;
Rectangle(h1,32,h1+16,48);
TextOut(h1+4,33,IntToStr(BBb[i]));
if i<=kpp then Brush.Color:=clLime
else Brush.Color:=clSilver;
if (kp<>kpp)and(i=kp) then Brush.Color:=clAqua;
Rectangle(h1,56,h1+16,72);
TextOut(h1+4,57,IntToStr(PPb[i]));
if i<=kp-n then Brush.Color:=clLime
else Brush.Color:=clSilver;
if (kp<>kpp)and(i=kp-n) then Brush.Color:=clAqua;
Rectangle(h1-16*n,56,h1+16-16*n,72);

```

```

TextOut(h1+4-16*n,57,IntToStr(PPb[i+n]));
if i<=kpe then Brush.Color:=clLime
else Brush.Color:=clSilver;
Rectangle(h1,80,h1+16,96);
TextOut(h1+4,81,IntToStr(PPe[i]));
if i<=kpe-n then Brush.Color:=clLime
else Brush.Color:=clSilver;
Rectangle(h1-16*n,80,h1+16-16*n,96);
TextOut(h1+4-16*n,81,IntToStr(PPe[i+n]));
end;
end;
end;
xm:=260;ym:=170;
with Image2.Canvas do
begin
Pen.Color:=clBlue;
Brush.Color:=clMoneyGreen;
Rectangle(xm+30,ym,xm+70,ym+40);
MoveTo(xm,ym+20);
LineTo(xm+30,ym+20);
TextOut(xm+36,ym+12,'Φ-KA');
Rectangle(xm+30,ym+50,xm+70,ym+90);
TextOut(xm+36,ym+62,'Φ-KB');
MoveTo(xm,ym+70);
LineTo(xm+30,ym+70);
Rectangle(xm+30,ym+100,xm+70,ym+140);
TextOut(xm+36,ym+112,'Φ-KP');
MoveTo(xm,ym+120);
LineTo(xm+30,ym+120);
Rectangle(xm+110,ym,xm+150,ym+90);
TextOut(xm+124,ym+12,'KB');
MoveTo(xm+70,ym+20);
LineTo(xm+110,ym+20);
MoveTo(xm+70,ym+70);
LineTo(xm+110,ym+70);
MoveTo(xm+70,ym+120);
LineTo(xm+190,ym+120);
Rectangle(xm+190,ym+50,xm+230,ym+140);
TextOut(xm+204,ym+62,'BC');
MoveTo(xm+150,ym+70);
LineTo(xm+190,ym+70);
MoveTo(xm+230,ym+120);
LineTo(xm+260,ym+120);
end;
with Image2.Canvas do
begin
Font.Color:=clWhite;
Pen.Color:=clWhite;
Brush.Color:=clWhite;
TextOut(xm,ym+24,'00000');
TextOut(xm,ym+74,'00000');
TextOut(xm,ym+124,'00000');
TextOut(xm+80,ym+24,'00');
TextOut(xm+80,ym+74,'00');
TextOut(xm+80,ym+124,'00');
TextOut(xm+160,ym+74,'00');
Font.Color:=clRed;
Pen.Color:=clBlue;
Brush.Color:=clWindow;
TextOut(xm,ym+4,'A');
TextOut(xm,ym+24,IntToStr(aa));
TextOut(xm,ym+54,'B');

```



```

TextOut(xm,ym+74,IntToStr(bb));
TextOut(xm,ym+104,'P');
TextOut(xm,ym+124,IntToStr(pe));
TextOut(xm+80,ym+4,'KA');
TextOut(xm+80,ym+24,IntToStr(ka));
TextOut(xm+80,ym+54,'KB');
TextOut(xm+80,ym+74,IntToStr(kb));
TextOut(xm+80,ym+104,'KP');
TextOut(xm+80,ym+124,IntToStr(kpe));
TextOut(xm+160,ym+54,'KP*');
TextOut(xm+160,ym+74,IntToStr(kp));
TextOut(xm+240,ym+104,'E');
TextOut(xm+240,ym+124,IntToStr(ep));
TextOut(xm-250,ym+20,'ОШИБКИ');
TextOut(xm-180,ym+20,'существенные');
TextOut(xm-100,ym+20,'несущественные');
TextOut(xm-250,ym+40,'Обнаружено');
TextOut(xm-250,ym+60,'Необнаружено');
TextOut(xm-150,ym+40,IntToStr(rde div 10)+','+
  IntToStr(rde mod 10)+ '%');
TextOut(xm-150,ym+60,IntToStr(rne div 10)+','+
  IntToStr(rne mod 10)+ '%');
TextOut(xm-70,ym+40,IntToStr(rdn div 10)+','+
  IntToStr(rdn mod 10)+ '%');
TextOut(xm-70,ym+60,IntToStr(rnn div 10)+','+
  IntToStr(rnn mod 10)+ '%');
end;
end;

Procedure TForm1.FormDragDrop(Sender, Source: TObject; X, Y: Integer);
begin
  X:=X div 32;
  //(Source as TPanel).Top:=32*(n+6)+16;
  if X>n then X:=n;
  (Source as TPanel).Left:=X*32+4;
  es:=n-X;
  (Source as TPanel).Width:=32*es+6;
  Panel13.Caption:=IntToStr(es);
end;

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

procedure TForm1.Panel12MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  if button=mbLeft then
    TPanel(Sender).BeginDrag(false);
end;

procedure TForm1.Panel14Click(Sender: TObject);
begin
  nfr:=1-nfr;
  if nfr=0 then
    begin
      Panel6.Caption:=MF[f1];
      Panel7.Caption:=MF[f2];
      Panel14.Caption:='заданная';
    end;
end;

```

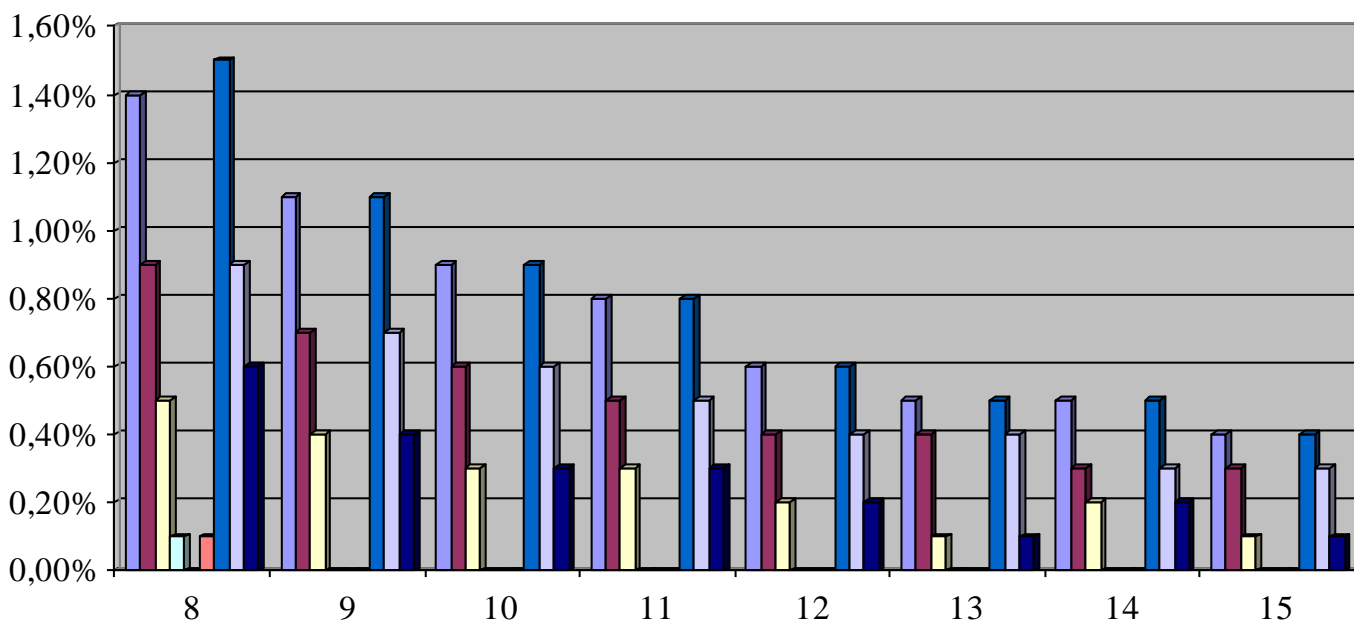
```
end
else
begin
  Panel6.Caption:='*';
  Panel7.Caption:='*';
  Panel14.Caption:='произвольная';
end
end;

procedure TForm1.Panel13Click(Sender: TObject);
begin
  if Panel13.Color=clAqua then
    Panel13.Color:=clBtnFace
  else
    if Panel13.Color=clBtnFace then
      Panel13.Color:=clAqua;
    if Panel13.Color=clRed then
      Panel13.Color:=clAqua;
    end;
  end;

end.
```

ДОСЛІДЖЕННЯ ЛОГАРИФМІЧНОГО КОНТРОЛЮ З ВИЯВЛЕННЯ НЕСПРАВНОСТЕЙ

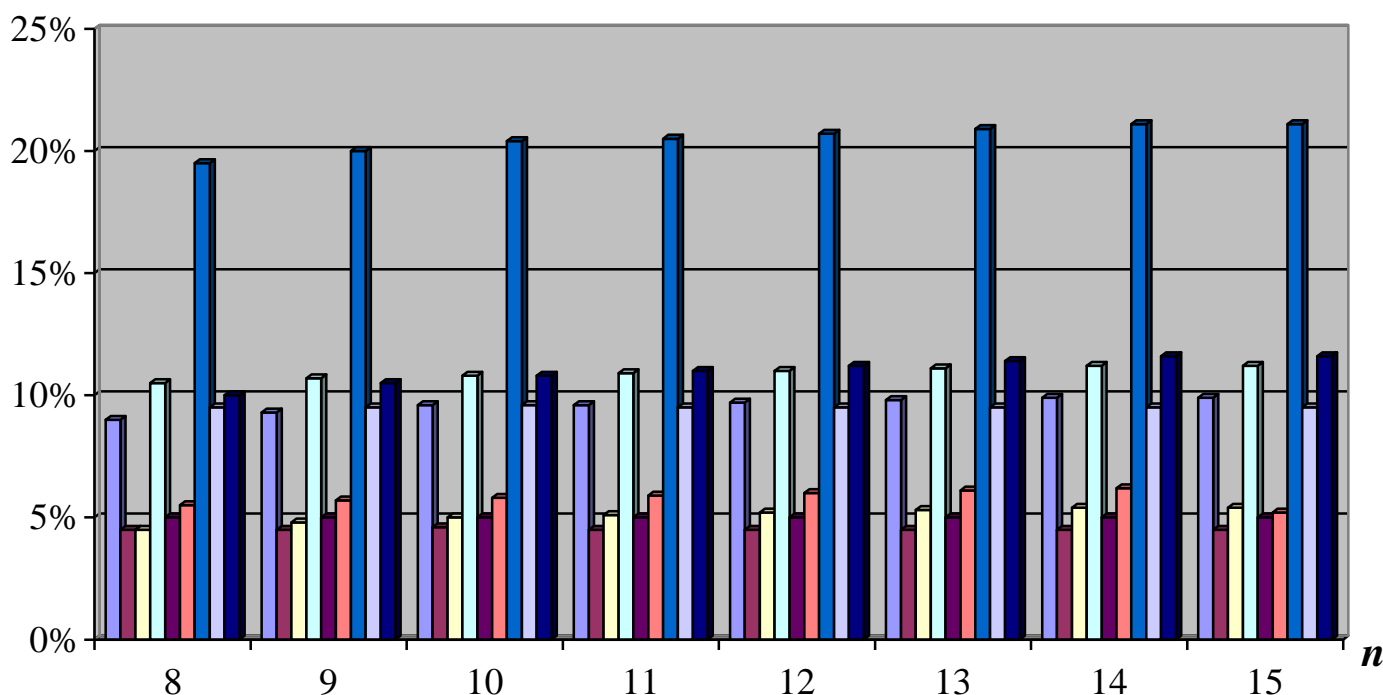
Результати моделювання	<i>n</i>							
	8	9	10	11	12	13	14	15
P_{BC}	1,4%	1,1%	0,9%	0,8%	0,6%	0,5%	0,5%	0,4%
$P_{BC\ 0 \rightarrow 1}$	0,9%	0,7%	0,6%	0,5%	0,4%	0,4%	0,3%	0,3%
$P_{BC\ 1 \rightarrow 0}$	0,5%	0,4%	0,3%	0,3%	0,2%	0,1%	0,2%	0,1%
P_{BH}	0,1%	0%	0%	0%	0%	0%	0%	0%
$P_{BH\ 0 \rightarrow 1}$	0%	0%	0%	0%	0%	0%	0%	0%
$P_{BH\ 1 \rightarrow 0}$	0,1%	0%	0%	0%	0%	0%	0%	0%
P_B	1,5%	1,1%	0,9%	0,8%	0,6%	0,5%	0,5%	0,4%
$P_{B\ 0 \rightarrow 1}$	0,9%	0,7%	0,6%	0,5%	0,4%	0,4%	0,3%	0,3%
$P_{B\ 1 \rightarrow 0}$	0,6%	0,4%	0,3%	0,3%	0,2%	0,1%	0,2%	0,1%



ЙМОВІРНОСТІ ВИЯВЛЕННЯ	ПОМИЛКИ		ВСЬОГО
	СУТТЄВІ (ВІДМОВИ)	НЕСУТТЄВІ	
ПОМИЛКИ	■ P_{BC}	■ P_{BH}	■ P_B
ПОЗИТИВНІ	■ $P_{BC\ 0 \rightarrow 1}$	■ $P_{BH\ 0 \rightarrow 1}$	■ $P_{B\ 0 \rightarrow 1}$
НЕГАТИВНІ	■ $P_{BC\ 1 \rightarrow 0}$	■ $P_{BH\ 1 \rightarrow 0}$	■ $P_{B\ 1 \rightarrow 0}$

ДОСЛІДЖЕННЯ ЛОГАРИФМІЧНОГО КОНТРОЛЮ З ПРОПУСКУ НЕСПРАВНОСТЕЙ

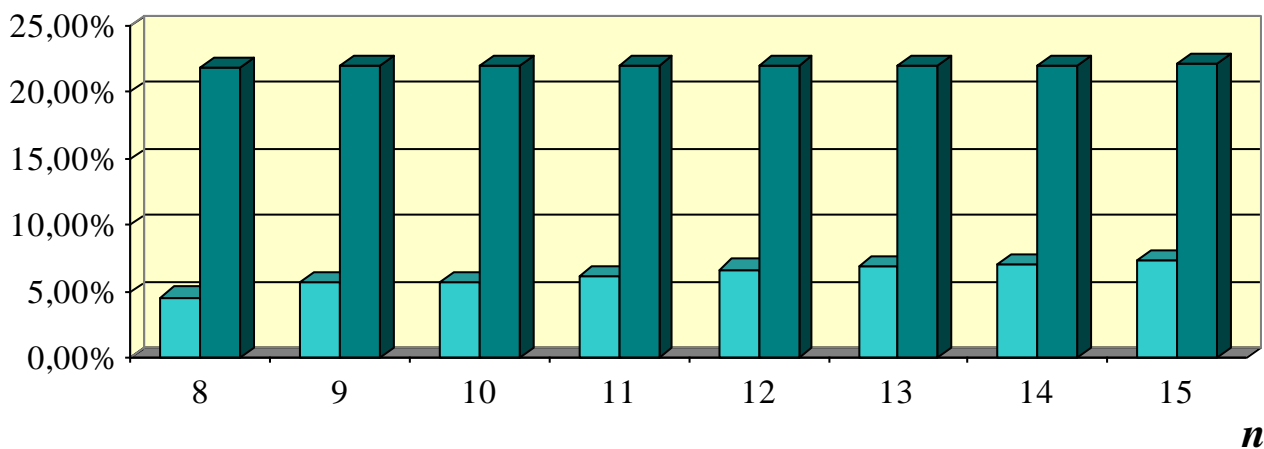
Результати моделювання	<i>n</i>							
	8	9	10	11	12	13	14	15
$R_{ПС}$	1,4%	1,1%	0,9%	0,8%	0,6%	0,5%	0,5%	0,4%
$R_{ПС\ 0 \rightarrow 1}$	0,9%	0,7%	0,6%	0,5%	0,4%	0,4%	0,3%	0,3%
$R_{ПС\ 1 \rightarrow 0}$	0,5%	0,4%	0,3%	0,3%	0,2%	0,1%	0,2%	0,1%
$R_{ПН}$	0,1%	0%	0%	0%	0%	0%	0%	0%
$R_{ПН\ 0 \rightarrow 1}$	0%	0%	0%	0%	0%	0%	0%	0%
$R_{ПН\ 1 \rightarrow 0}$	0,1%	0%	0%	0%	0%	0%	0%	0%
$R_{П}$	1,5%	1,1%	0,9%	0,8%	0,6%	0,5%	0,5%	0,4%
$R_{П\ 0 \rightarrow 1}$	0,9%	0,7%	0,6%	0,5%	0,4%	0,4%	0,3%	0,3%
$R_{П\ 1 \rightarrow 0}$	0,6%	0,4%	0,3%	0,3%	0,2%	0,1%	0,2%	0,1%



ЙМОВІРНОСТІ ПРОПУСКУ	ПОМИЛКИ		ВСЬОГО
	СУТТЄВІ	НЕСУТТЄВІ	
ПОМИЛКИ, ПОЗИТИВНІ, НЕГАТИВНІ	■ $R_{ПС}$	■ $R_{ПН}$	■ $R_{П}$
	■ $R_{ПС\ 0 \rightarrow 1}$	■ $R_{ПН\ 0 \rightarrow 1}$	■ $R_{П\ 0 \rightarrow 1}$
	■ $R_{ПС\ 1 \rightarrow 0}$	■ $R_{ПН\ 1 \rightarrow 0}$	■ $R_{П\ 1 \rightarrow 0}$

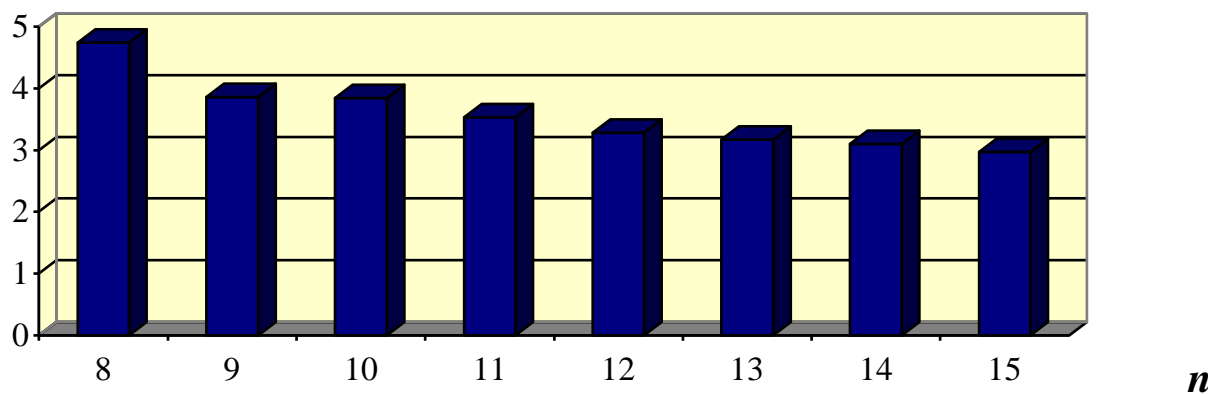
ДОСЛІДЖЕННЯ ДОСТОВІРНОСТІ ЛОГАРИФМІЧНОГО КОНТРОЛЮ НАБЛИЖЕНИХ РЕЗУЛЬТАТІВ

Достовірність контролю, %	<i>n</i>							
	8	9	10	11	12	13	14	15
<i>D</i> _{лог}	95,4	94,7	94,3	93,8	93,3	93,1	92,9	92,6
<i>D</i> _{мод}	78,2	78,1	78,1	78,1	78,0	78,0	78,0	78,0



$D_{\text{лог}} = P_{\text{вс}} + P_{\text{пн}}$
 $D_{\text{мод}}$
*D*_{лог} та *D*_{мод} – стовірність логарифмічного контролю та контролю за модулем

Показник ефективності	<i>n</i>							
	8	9	10	11	12	13	14	15
δD	4,74	3,86	3,84	3,53	3,28	3,17	3,1	2,97



$\delta D = D_{\text{лог}} / D_{\text{мод}}$

