

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

ДОВГАНЬ Ярослав Олександрович

ДИПЛОМНА РОБОТА МАГІСТРА

ДОСЛІДЖЕННЯ СИСТЕМИ ГРАНИЧНОГО КОНТРОЛЮ ПОМНОЖУВАЧА ІЗ
ЗРОСТАННЯМ КІЛЬКОСТІ ВІДМОВ

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

Довганя Ярослава Олександровича

1. Тема роботи “Дослідження системи граничного контролю помножувача із зростанням кількості відмов”

затверджена наказом по університету від “ ” № .

2. Термін здачі студентом закінченої роботи 03.12.2021

3. Вихідні дані до роботи у розділі “Визначення мети та задач дослідження”

4. Зміст розрахунково-пояснювальної записки (перелік питань, що належить розробити) _____

1. Робоче діагностування обчислювальних пристроїв для обробки наближених даних

2. Визначення мети та задач дослідження

3. Аналіз задачі з дослідження граничного контролю матричного помножувача

4. Виконання граничного контролю матричного помножувача мантис

5. Результати проведених досліджень

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

1) Виконання повної операції у матричному помножувачі мантис.

2) Виконання скороченої операції у матричному помножувачі мантис.

3) Особливості операцій, що підлягають контролю в помножувачі.

4) Визначення та обчислення границь та коду контролю.

5) Виконання граничного контролю.

6) Модель граничного контролю помножувача мантис.

7) Виявлення суттєвих та несуттєвих помилок,

8) Достовірність граничного контролю та контролю за модулем

8) Ефективність граничного контролю у порівнянні з контролем за модулем

10) Достовірність граничного контролю в умовах дії кратних відмов.

№ рядка	Формат	Позначення	Найменування	Кільк.	№ екз.	Примітка			
1			Документація загальна						
2			<u>Знов розроблена</u>						
3	A4	АМДР.АМ161.0303.01ПЗ	Пояснювальна записка	121					
4	A1	АМДР.АМ161.0303.01Д1	Виконання повної операції	1					
5			у матричному помножувачі мантис						
6	A1	АМДР.АМ161.0303.01Д2	Виконання скороченої операції	1					
7			у матричному помножувачі мантис						
8	A1	АМДРАМ161.0303.01Д3	Особливості операцій, що	1					
9			підлягають контролю в						
10			помножувачі						
11	A1	АМДР.АМ161.0303.01Д4	Визначення та обчислення	1					
12			границь та коду контролю						
13	A1	АМДР.АМ161.0303.01Д5	Виконання граничного контролю	1					
14	A1	АМДР.АМ161.0303.01Д6	Модель граничного контролю	1					
15			помножувача мантис						
16	A1	АМДР.АМ161.0303.01Д7	Виявлення суттєвих та несуттєвих	1					
17			помілок						
18	A1	АМДР.АМ161.0303.01Д9	Достовірність граничного	1					
19			контролю та контролю за модулем						
20	A1	АМДР.АМ161.0303.01Д10	Ефективність граничного контролю	1					
21			у порівнянні з контролем за						
22			модулем						
23	A1	АМДР.АМ161.0303.01Д8	Достовірність граничного	1					
24			контролю в умовах дії кратних						
25			відмов						
26									
27									
28									
АМДР.АМ161.0303									
Ізм.	Лист	№ документа	Підпис	Дата	Дослідження системи граничного контролю помножувача із зростанням кількості відмов Відомість дипломної роботи				
Розробив	Довгань						Лист	Лист	Листів
Перевірів	Дрозд								1
Н. контр.							ОНПУ ІКС		
Затверд.							КІСМ АМ-161		

АНОТАЦІЯ

Довгань Я. О. Дослідження системи граничного контролю помножувача із зростанням кількості відмов – Магістерська кваліфікаційна робота. Одеса, 2021.: 121 с., 23 рис., 12 табл., 1 додаток, 18 джерел.

Об'єкт дослідження – процес граничного контролю матричного помножувача мантис дією його кратних несправностей.

Предмет дослідження – достовірність граничного контролю матричного помножувача мантис одинарної точності під дією його кратних несправностей.

Мета роботи – дослідження граничного контролю для визначення достовірності результатів, що обчислюються у матричному помножувачі мантис під дією його кратних несправностей.

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

Магістерська робота розглядає проблему контролю арифметичних пристроїв для обробки приблизних даних. Традиційні методи контролю демонструють низьку ефективність в оцінці достовірності приблизних результатів. Стає доцільним розвиток нетрадиційних методів контролю.

Розроблено алгоритми та програмну модель схеми граничного контролю матричного помножувача мантис. Досліджено ймовірності появи, виявлення й пропуску суттєвих і несуттєвих помилок. Проведено дослідження достовірності результатів, що обчислюються у матричному помножувачі мантис під дією кратних відмов.

КОНТРОЛЬ АРИФМЕТИЧНИХ ПРИСТРОЇВ, ОБРОБКА ПРИБЛИЗНИХ ДАНИХ, МАТРИЧНИЙ ПОМНОЖУВАЧ МАНТИС, ГРАНИЧНИЙ КОНТРОЛЬ, КРАТНІ ВІДМОВИ

ABSTRACT

Dovgan Y. O. Research of the boundary checking system of multiplier with increasing number of failures. The master qualifying work. – Odessa, 2021: 121 p., 23 fig., 12 table, 1 annex, 18 references.

The object of the study is the process of boundary checking of a single-precision iterative array multiplier of mantissas under the action of its multiple faults.

The subject of the study is reliability of the boundary checking of the iterative array multiplier of mantissas of single accuracy under the action of its multiple faults.

The purpose of the work is to study the boundary checking to determine the trustworthiness of the results calculated in the iterative array multiplier of mantissas under the influence of its multiple faults.

Methods of researches are based on elements of the theory of diagnostics of computers, probability theory, the theory of algorithms and mathematical logic and also the applied theory of digital automatic machines.

Master work addressed the problem of checking arithmetic devices for processing approximate data. Traditional checking methods show poor effectiveness in assessing the trustworthiness of approximate results. The development of non-traditional checking methods was becoming desirable.

Algorithms and software model of boundary checking of iterative array multiplier of mantissas have been developed. Probabilities of emergence, detection and the admission of essential and inessential errors and reliability of results checking are investigated. The reliability of the results calculated in the iterative array multiplier of mantissas under the influence of multiple faults was investigated.

CHECKING OF ARITHMETIC DEVICES, PROCESSING OF APPROXIMATE DATA, ITERATIVE ARRAY MULTIPLIER OF MANTISSAS, BOUNDARY CHECKING, MULTIPLE FAILURES.

ЗМІСТ

Вступ	5
1 Робоче діагностування обчислювальних пристроїв для обробки наближених даних	9
1.1 Основи робочого діагностування арифметичних пристроїв	9
1.2 Граничний контроль	14
1.3 Використання нерівностей в граничному контролі помножувача мантис	16
1.4 Достовірність контролю результатів обробки мантис	20
1.5 Висновки	23
2 Визначення мети та задач дослідження	24
2.1 Найменування, область застосування та призначення досліджень	24
2.2 Необхідність проведення досліджень	25
2.3 Організація досліджень	27
2.4 Постановка задачі	28
3 Аналіз задачі з дослідження граничного контролю матричного помножувача	31
3.1 Визначення границь результату для контролю матричного помножувача мантис	31
3.2 Виконання контрольних обчислень	33
3.3 Структура програмної моделі граничного контролю	35
3.4 Функціонування програмної моделі граничного контролю	37
3.5 План проведення досліджень граничного контролю матричного помножувача мантис із зростанням кількості відмов	37
3.6 Висновки	40
4 Виконання граничного контролю матричного помножувача мантис	41

4.1 Формування поточної границі для граничного контролю результату	41
4.2 Порівняння результату з поточною границею та формування коду граничного контролю	43
4.3 Обчислення ймовірності виявлення помилки результату в однокітному матричному помножувачі мантис із зростанням кількості відмов	45
4.4 Оцінка ймовірностей виявлення та пропуску суттєвих та несуттєвих помилок в умовах зростання кількості відмов	47
4.3 Висновки	49
5 Результати проведених досліджень	51
5.1 Програмна модель граничного контролю матричного помножувача мантис із зростанням кількості відмов	51
5.2 Дослідження ймовірностей виявлення та пропуску помилок граничним контролем в умовах зростання кількості відмов	54
5.3 Висновки	86
Висновок	87
Перелік посилань	89
Додаток А. Програмна модель граничного контролю	91

ВСТУП

Достовірність результатів обчислень відноситься до найважливіших показників ефективності комп'ютерної обробки даних в сучасних арифметичних пристроях. Для виявлення недостовірних результатів арифметичні пристрої доповнюються схемами вбудованого контролю, що розробляються за методами контролю. Ці методи відіграють значну роль в забезпеченні продуктивної роботи комп'ютерних систем відповідно до їх власної ефективності, тобто спроможності відрізнити достовірний результат від недостовірного. Таку спроможність традиційно трактували як здатність виявляти несправності пристроїв при їх першому прояві у вигляді помилки результату.

Методи контролю за парністю та модулем були розроблені для матричних арифметичних пристроїв з традиційно високою виявляючою здатністю. Вони демонструють високу ефективність при контролі точних обчислень, які ототожнюють недостовірний результат з помилковим, та стають неспроможними для оцінки достовірності приблизного результату.

Приблизні результати містять старші вірні та молодші невірні розряди. Помилки в вірних розрядах роблять результат недостовірним, а помилкові невірні розряди ніякого впливу на достовірність результату не мають. Для приблизних обчислень, що стають все більш поширеними, традиційні методи контролю витрачають свою високу виявляючу здатність на пошук найбільш ймовірних несуттєвих помилок. Це суперечить забезпеченню високої продуктивності комп'ютерних систем, оскільки призводить до марних втрат часу на перерахування помилкових, але достовірних результатів. Тому намітився відхід від традиційних методів контролю й пошук методів, що безпосередньо націлені на контроль приблизних результатів, тобто розрізнення в ньому суттєвих та несуттєвих помилок. Виявлення таких методів потребує додаткових досліджень.

Альтернативні можливості представляє граничний контроль, який порівнює результат з його верхньою та нижньою границями, що дозволяє розрізнити суттєві та несуттєві помилки, оцінюючи величину результату та його помилки. Однак

ефективність методу граничного контролю суттєво залежить від форми границь, і її оцінка робить актуальною тему досліджень, що присвячена моделюванню граничного контролю для матричного помножувача мантис.

Мета й задачі дослідження. Метою магістерської роботи є дослідження граничного контролю для визначення достовірності результатів, що обчислюються у матричному помножувачі мантис під дією його кратних несправностей.

Для досягнення цієї мети в магістерській роботі розв'язуються наступні задачі:

- досліджуються традиційні методи контролю за їх ефективністю у визначенні достовірності результатів приблизних обчислень;
- досліджуються методи побудування матричного помножувача мантис одинарної точності та його кратні несправності;
- досліджується метод граничного контролю щодо перевірки добутку мантис одинарної точності, визначаються нижні та верхні границі результату;
- розробляються алгоритми та програмна модель схеми граничного контролю матричного помножувача мантис одинарної точності для різних визначень границь результату;
- перевіряється правильність функціонування програмної моделі схеми граничного контролю матричного помножувача мантис одинарної точності;
- виконується моделювання роботи схеми граничного контролю матричного помножувача мантис одинарної точності;
- за результатами моделювання досліджуються ймовірності появи, виявлення та пропуску суттєвих та несуттєвих помилок, що викликані кратними несправностями матричного помножувача мантис, а також достовірність контролю приблизних результатів.

Об'єкт дослідження – процес граничного контролю матричного помножувача мантис під дією його кратних несправностей.

Предмет дослідження – достовірність граничного контролю матричного помножувача мантис одинарної точності під дією його кратних несправностей.

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

Наукова новизна полягає в таких положеннях:

- дістав подальший розвиток метод граничного контролю матричного помножувача мантис, який досліджено для умов роботи пристрою при кратних відмовах;
- розроблені алгоритми та програмна модель схеми граничного контролю матричного помножувача мантис одинарної точності, що функціонує в умовах дії його кратних відмов;
- за результатами моделювання оцінені ймовірності появи, виявлення та пропуску суттєвих та несуттєвих помилок, що викликані кратними відмовами матричного помножувача мантис, а також достовірності контролю приблизних результатів.

Практичне значення одержаних результатів полягає у визначенні оцінок появи та виявлення суттєвих і несуттєвих помилок, а також достовірності граничного контролю результатів, які обчислюються на матричному помножувачі мантис одинарної точності під дією його кратних відмов.

Такі оцінки дозволяють ефективно використовувати матричний помножувач мантис, знижуючи кількість достовірних результатів, що бракуються при виявленні несуттєвих помилок. Це сприяє підвищенню ефективності виконання приблизних обчислень в умовах критичного застосування комп'ютерної техніки.

Структура магістерської роботи. Магістерська робота складається з вступу, п'яти розділів та висновків, переліку літератури а також додатку, що містить текст програмної моделі.

У вступі викладено основні положення магістерської роботи – актуальність теми, мета дослідження, задачі, що розв'язуються для її досягнення, об'єкт, предмет та методи дослідження, а також наукова новизна та практичне значення одержаних результатів.

У першому розділі розглядаються основні положення контролю цифрових схем комп'ютерних вузлів та особливості виконання приблизних обчислень у арифметичних пристроях. Досліджуються традиційні методи контролю та можливості їхнього використання для обробки приблизних чисел у форматах з плаваючою точкою. Розглядається метод граничного контролю та його переваги при застосуванні у виконанні приблизних обчислень.

У другому розділі викладені вихідні дані та вимоги для проведення дослідження схеми граничного контролю матричного помножувача мантис одинарної точності. Обґрунтовується необхідності проведення такого дослідження відповідно до нових вимог, що встановлюються до схем контролю в обробці приблизних даних. Конкретизуються завдання щодо визначення границь результату, розробки програмної моделі схеми граничного контролю та її дослідження в умовах дії кратних несправностей матричного помножувача мантис.

Третій розділ містить подальший розвиток методу граничного контролю матричного помножувача мантис одинарної точності. Досліджується задача визначення нижніх та верхніх границь результату, а також можливості обчислення поточної границі.

У четвертому розділі викладено розроблені алгоритми та програмна модель схеми граничного контролю матричного помножувача мантис одинарної точності, а також верифікацію одержаної програмної моделі на контрольних прикладах.

У п'ятому розділі викладено план дослідження схеми граничного контролю матричного помножувача мантис одинарної точності, результати моделювання під дією кратних відмов та їх обробку до остаточних результатів дослідження.

У висновку наводяться основні результати роботи.

У додатку викладено текст програми.

1 РОБОЧЕ ДІАГНОСТУВАННЯ ОБЧИСЛЮВАЛЬНИХ ПРИБРОЇВ ДЛЯ ОБРОБКИ НАБЛИЖЕНИХ ДАНИХ

1.1 Основи робочого діагностування обчислювальних пристроїв

Виконання обчислень на цифрових схемах підтримується в процесі виконання основних операцій на фактичних даних робочим діагностуванням.

На цифрові схеми впливають несправності, які приводять до перекручувань результатів, що обчислюються, тобто до помилок.

Методи та засоби тестового та робочого діагностування використовуються для виявлення несправностей і помилок [1].

В паузах роботи ОП на спеціально підготовлених тестових наборах несправності виявляються найбільше ефективно за методами й засобами тестового діагностування. Виявлення несправностей націлене на оцінку технічного стану ОП, достовірності його функціонування [2].

Разом з тим, достовірне функціонування ОП в паузах його роботи не гарантує одержання достовірних результатів розв'язання обчислювальних задач в основному робочому режимі.

Помилки, викликувані несправностями цифрових схем у процесі обчислень, виявляються методами й засобами робочого діагностування, націленого на оцінку достовірності результатів, що обчислюються [3].

Виявлення помилок ґрунтується на використанні інформаційної надмірності, що множину дозволених вихідних слів доповнює множиною заборонених значень. До дозволених вихідних слів відносяться правильні значення результату, які можуть бути обчислені на різних вхідних робочих словах. Вхідні слова утворюються операндами обчислювальних операцій. До заборонених значень відносяться вихідні слова, що не належать множині дозволених слів. Засоби контролю розрізняють дозвалені й заборонені вихідні слова, виявляючи помилку при обчисленні забороненого значення результату [4].

Робоче діагностування базується на визначеннях самоперевіряємості цифрових схем [5].

Схема називається такою, що самоперевіряється, якщо вона є захищеною й самотестуємою у заданому класі несправностей.

До захищених цифрова схема належить, якщо для кожної несправності із заданого класу на вхідних робочих словах обчислюється або правильний результат, або заборонене слово.

Цифрова схема є самотестуємою у заданому класі несправностей, якщо для кожної такої несправності знайдеться хоча б одне вхідне робоче слово, на якому обчислюється заборонене значення.

Слід відмітити, що вхідні й вихідні слова, використані у визначеннях самоперевіряємості цифрових схем, є точними даними, до яких відносяться порядкові числівники, що нумерують елементи множин. Це визначило розвиток робочого діагностування в рамках обробки точних даних.

Протягом десятиліть були розроблені методи робочого діагностування, які демонструють високу ефективність стосовно ОП, що виконує операції над точними даними.

До таких методів відносяться контроль за числовим модулем та контроль по паритету.

Контроль за числовим модулем визнаний одним з основних методів робочого діагностування ОП. В основі методу лежить порівнянність чисел по модулю, що описана в теорії чисел [6].

За цією теорією, будь-яке ціле число може бути представлене у вигляді $A = Md + r$, де M , d і r – цілі числа; M – модуль; $d = \text{Int}(A/M)$ – найменше ціле дробу A/M ; r – залишок від ділення числа A на модуль M , $r = 0, 1, \dots, M-1$.

Запис $r = A \bmod M$ установлює відповідність між числом A и його залишком r по модулю M .

Цілі числа поділяються за модулем M на M класів еквівалентності. Кожний із цих класів містить числа з однаковим значенням залишку, тобто порівнянними по модулю M . У контролі по числовому модулю залишок r використовується як

контрольний код KA числа A . При цьому у випадку $r=0$ код KA має два подання нуля: $+0$ і -0 , які кодуються значеннями 0 і M , відповідно.

Помилки обчислень визначаються за умови віднесення перекрученого результату до іншого класу еквівалентності, чим правильний результат. Якщо правильний і неправильний результати потрапляють у той самий клас еквівалентності, то контроль не розрізняє ці результати, і помилка не виявляється.

Ймовірність пропуску довільної помилки оцінюється як $P_{\text{ПРОП}} = 1/M$, а ймовірність її виявлення визначається по наступній формулі [7] $P_{\text{ОБН}} = 1 - 1/M$.

Ймовірність виявлення конкретних видів помилок може суттєво відрізнятись від $P_{\text{ОБН}}$.

Несправності матричних ОП спотворюють результат на вагу будь-якого одного розряду. Такі помилки мають вигляд $\pm 2^Z$, де Z – номер розряду, і виявляються по будь-якому непарному модулю M , оскільки $\pm 2^Z \bmod M \neq 0$. Тому для матричних ОП досить використовувати контроль по найбільш просто виконуваному модулю $M = 3$.

Операції додавання $S = A + B$ і множення $V = A \cdot B$ перевіряються з використанням наступних контрольних співвідношень:

$$KA + KB = KS; \quad KA \cdot KB = KV,$$

де KA та KB – контрольні коди операндів,

$$KA = A \bmod M; \quad KB = B \bmod M;$$

KS та KV – контрольні коди результатів,

$$KS = S \bmod M; \quad KV = V \bmod M.$$

Правильність обчислення арифметичних виразів, складених з операцій додавання та множення, перевіряється по контрольних співвідношеннях, які утворюються з основних формул заміною операндів на їхні контрольні коди.

Прикладом є контроль операції цілочислового ділення, що зв'язує ділене A , дільник B , частку C та залишок D по формулі $A = B \cdot C + D$, з якої витікає контрольне співвідношення для перевірки ділення $KA = KB \cdot KC + KD$.

Описаний контроль за модулем називається числовим на відміну від цифрового контролю по модулю, для якого контрольний код числа визначається як залишок від ділення на модуль суми цифр цього числа.

Контроль арифметичних операцій за цифровим модулем може зажадати реалізації більш складних контрольних співвідношень, однак відрізняється простим формуванням контрольних кодів.

Модулі, що сполучають простоту контрольних співвідношень і способів обчислення контрольних кодів, оскільки одночасно є й числовими й цифровими модулями мають практичний інтерес.

До них належать модулі виду $M = N - 1$, де N – основа системи числення [8].

Контрольні коди є цифрами N -ї системи числення та мають природу оберненого коду із двома значеннями нуля: $+0$ та -0 , відповідно 0 та $N - 1$.

Контрольний код числа обчислюється як сума його цифр. Причому, якщо сума має більше одного розряду, то підраховується сума її цифр, що виконується до одержання однієї контрольної цифри. За цим правилом може обчислюватися й контрольний код дробового числа. Операція обчислення контрольної цифри називається згорткою числа [9].

Для перевірки двійкової арифметики використовуються системи числення з основами $N = 2^L$ та контроль по модулю $M = 2^L - 1$.

Найбільше поширення одержав контроль по модулю три, що працює в четверичній системі числення.

Контроль по паритету використовується в обчислювальних системах, насамперед, для перевірки правильності передачі даних по шинах.

Дослідження самоперевіряємості схем дали ряд розв'язань із використанням контролю по паритету для суматора, помножувача та поділювача.

Однак такий контроль вимагає обліку міжрозрядних перенесень, що у випадку матричних ОП приводить до квадратичних витрат устаткування на контроль і обмежує область практичного застосування [10].

Основною перевагою описаних методів контролю є висока ймовірність виявлення помилок, що забезпечує їхню безумовну ефективність для робочого діагностування ОП у випадку обробки точних даних.

Однак до точних даних відносяться тільки числа, що є цілими по своїй природі. Цими даними є номери елементів множин, називані порядковими числівниками.

Інші числа, що становлять основну частину оброблюваних даних, належать до наближених чисел, і частка їх у комп'ютерній обробці постійно росте.

Стосовно наближених даних традиційні методи робочого діагностування проявляють суттєвий недолік, відбраковуючи достовірні результати. Це відбувається, оскільки стосовно наближених даних помилки можуть бути суттєвими й несуттєвими для достовірності результатів, що обчислюються. Виявлення несуттєвих помилок веде до відбраковування достовірних результатів, необґрунтовано знижуючи продуктивність ОП при проведенні повторних обчислень.

Традиційні методи робочого діагностування не розрізняють суттєві та несуттєві помилки, демонструючи високу виявляючу здатність стосовно тих та інших помилок.

А оскільки несуттєві помилки відбуваються значно частіше, ніж суттєві, то в основному виявляються несуттєві помилки. Це приводить до відбраковування достовірних результатів [11].

Для подолання зазначеного недоліку необхідно виходити за рамки традиційних методів.

До перспективних методів робочого діагностування відноситься метод граничного контролю. Цей метод дозволяє оцінювати величину помилки й у такий спосіб розрізнати суттєві та несуттєві помилки.

Однак у цей час метод граничного контролю не одержав широкого застосування, вимагаючи проведення додаткових досліджень його ефективності.

1.2 Граничний контроль

Метод граничного контролю ґрунтується на визначенні верхньої та нижньої границь результату як функцій операндів контрольованої операції. Результат порівнюється з його границями, що дозволяє виявляти позитивні помилки, що лежать вище верхньої границі та негативні, що розташовуються нижче нижньої границі результату.

Числовий граничний контроль був запропонований для перевірки функцій одного аргументу в автономних обчислювачах функцій з метою одержання високої ймовірності виявлення помилок [12].

В цьому методі значення функції $F(X)$ перевіряються на входження в деякий діапазон $[F_{min}, F_{max}]$ при зміні аргументу в заданому діапазоні X , тобто виконується перевірка наступної умови: $F_{min} \leq F(X) \leq F_{max}$ для всієї області визначення аргументу X та зміни функції $F(X)$.

Для вирішення цієї задачі будується схема контролю, основу якої складає постійний запам'ятовуючий пристрій (ПЗП). В ньому зберігаються значення нижньої та верхньої границь функції для низки значень аргументу.

Контроль здійснюється порівнянням функції з її границями.

Було запропоновано реалізації методу числового граничного контролю за двома способами: спосіб «прямокутника»; спосіб «коридору».

За першим способом відношення F_{max} / F_{min} вибирається за значенням кусково-лінійної апроксимуючої функції із заданим допуском.

За другим способом значення F_{min}, F_{max} вибираються для окремих інтервалів X за крайніми значеннями функції на цих інтервалах.

Даний метод не враховує особливостей контрольованих функцій. Висока ймовірність виявлення помилки потребує ПЗП великого обсягу для зберігання даних для значної кількості значень аргументу.

Цим пояснюється вузька область застосування методу та недоцільність його розповсюдження в такому вигляді на двомісні операції.

До переваг методу можна віднести його орієнтація на приблизний характер обчислень, що проявляється в ігноруванні помилок у відкидаємих розрядах та придатність оцінювати величину помилки.

Граничний контроль з використанням нерівностей був також розроблений для виявлення відмов в арифметичних пристроях [13 – 15].

В качестве границ используются просто вычисляемые по операндам линейные функции. Такой учет особенностей арифметических операций обеспечил получение простых средств контроля.

Функції границь вибираються, виходячи з умови простоти їхнього обчислення та наближеності до значень результату для забезпечення високої виявляючої здатності.

Найбільш близькі до результату значення функцій границь забезпечують виявлення найменших по абсолютній величині помилок і всіх більших їхніх помилок.

Найбільш віддалені значення функцій границь дозволяють виявляти більше грубі помилки, що забезпечує оцінку величини помилки. Зі збільшенням значення помилки росте ймовірність її виявлення. Це дозволяє розрізняти і суттєві помилки, що виникають у старших розрядах результатів, від несуттєвих помилок, що мають меншу величину, оскільки вони відбуваються в молодших розрядах.

Для спрощення порівняння результату з його границями, ця процедура виконується послідовно: спочатку результат рівняється з однією границею, а потім з іншої. Це дозволяє сполучити обчислення функцій границь, формуючи їх по черзі на одному виході схеми контролю у вигляді поточної границі.

Для організації почергової перевірки результату необхідно задавати черговість порівняння границь, уводячи параметр керування, що чергує пряме й інверсне значення. При цьому результат порівняння також стане чергувати правдиве та помилкове значення.

У сукупності з параметром керування складається дворозрядний код контролю, розряди якого приймають однакові значення при виявленні помилок і взаємо-інверсні значення в протилежному випадку.

1.3. Використання нерівностей в граничному контролі помножувача мантис

Множення мантис найбільш часто виконується з одинарною точністю, що досягається обмеженням розрядності добутку, що обчислюється, до розрядності операндів.

Області визначення мантис описуються нерівностями $0,5 \leq A < 1$ та $0,5 \leq B < 1$.

Тоді для операції множення верхня та нижня границі результату можуть бути одержані з наступної формули [18].

$$(A - X)(B - Y)(-1)^Z \leq 0, \quad (3.1)$$

де X та Y – константи, що приймають значення з діапазону $0 \div 1$;

$Z = 0$ та $Z = 1$ при визначенні верхньої та нижньої границь, відповідно.

Після розкриття дужок формула (3.1) перетворюється до наступного вигляду

$$A B (-1)^Z \leq (Y A + X B - Y X),$$

за яким нижня V_H та верхня V_B границі результату $V = A B$ визначаються як

$$V_H = Y A + X B - Y X,$$

де $A \leq X$ та $B \leq Y$ або $A \geq X$ та $B \geq Y$;

$$V_B = Y A + X B - Y X,$$

де $A \leq X$ та $B \geq Y$ або $A \geq X$ та $B \leq Y$.

Лінійні залежності границь результату від операндів визначають простоту контрольних обчислень і, відповідно, засобів контролю для їх реалізації.

Нижня границя результату в [13] визначена при $X = 1$ та $Y = 1$, а також $Z = 1$.

3.1.2 Визначення границь результату за значеннями параметрів: $X = 0,75$ та $Y = 0,75$, встановлює нижню та верхню границю відповідно при $Z = 1$ та $Z = 0$.

Дійсно, для вибраних значень параметрів X та Y , а також $Z = 1$ формула (3.1) перетворюється до наступного вигляду:

$$(A - 0,75)(B - 0,75) \geq 0,$$

що після розкриття дужок визначає нижню границю результату

$$V_H = 0,75(A + B) - 9 \cdot 2^{-4},$$

де $A \leq 0,75$ та $B \leq 0,75$ або $A \geq 0,75$ та $B \geq 0,75$.

Для вибраних значень параметрів X та Y , а також $Z = 0$ формула (3.1) перетворюється до наступного вигляду:

$$(A - 0,75)(B - 0,75) \leq 0,$$

що після розкриття дужок визначає верхню границю результату

$$V_B = 0,75(A + B) - 9 \cdot 2^{-4},$$

де $A \leq 0,75$ та $B \geq 0,75$ або $A \geq 0,75$ та $B \leq 0,75$.

3.2.1 Виконання контрольних обчислень у граничному контролі складається з обчислення границь та їх порівняння з результатом.

Обчислення нижньої та верхньої границь результату виконується за одною формулою, що визначає поточну границю:

$$V_{HB} = 0,75(A + B) - 9 \cdot 2^{-4},$$

Обчислення такої формули потребує виконання низки операцій:

- операції додавання, що визначає суму мантис співмножників:

$$S_{A+B} = A + B;$$

- операції множення на константу $0,75$, що визначає добуток, використовуючи додавання наступним чином

$$V_{0,75(A+B)} = (S_{A+B} + 0,5 \cdot S_{A+B}) 0,5;$$

- операції віднімання константи $9 \cdot 2^{-4}$, що визначає різницю, використовуючи додавання у додатковому коді з константою $7 \cdot 2^{-4}$, як

$$V_{HB} = (V_{0,75(A+B)} + 7 \cdot 2^{n-4}) \& (2^n - 1), \quad (3.2)$$

де $\& (2^n - 1)$ – операція логічного множення на двійковий код, що дорівнює $2^n - 1$, для відкидання старшої одиниці суми у додатковому коді.

3.2.2 Для розпізнавання поточної границі як нижньої або верхньої необхідно визначити умови

- для співмножника A

$$D_A = 0 \text{ при } A \geq 0,75 \text{ та } D_A = 1 \text{ при } A < 0,75;$$

- для співмножника B

$$D_B = 0 \text{ при } B \geq 0,75 \text{ та } D_B = 1 \text{ при } B < 0,75;$$

- для нижньої границі

$$D_A = 0 \text{ та } D_B = 0 \text{ або } D_A = 1 \text{ та } D_B = 1;$$

– для верхньої границі $D_A = 0$ та $D_B = 1$ або $D_B = 0$ та $D_A = 1$.

Умови визначаються наступним чином:

$$D_A = C(A < 0,75);$$

$$D_B = C(B < 0,75)$$

та обчислюються за формулами

$$D_A = \text{Sign}(A - 0,75) = \neg A\{2\};$$

$$D_B = \text{Sign}(B - 0,75) = \neg B\{2\},$$

де $C(*)$ – результат операції порівняння $*$, що міститься в дужках;

$\text{Sign}(*)$ – код знаку різниці, що міститься в дужках.

Нижня та верхня границі визначаються відповідно за умовами $D_A = D_B$ та $D_A = \neg D_B$ або $A\{2\} = B\{2\}$ та $A\{2\} = \neg B\{2\}$.

Порівняння поточної границі з результатом визначає розряди коду контролю як наступне:

$$E_1 = C((V_{HB} + D) < V),$$

$$E_2 = C(D_A = D_B),$$

що обчислюється за формулами

$$E_1 = (V\{1 \div n\} + D + \neg V_{HB}) \text{ shr } n,$$

$$E_2 = A\{2\} \oplus B\{2\},$$

де $D = \neg(D_A \vee D_B)$ або $D = A\{2\} \wedge B\{2\}$ – розряд корекції, який при $V_{HB} = V$ встановлює дозволене значення коду контролю.

1.4 Достовірність контролю результатів обробки мантис

Мантиси чисел із плаваючою точкою належать до наближених даних, які становлять основну частину оброблюваних чисел. Наближений результат обчислень може бути перекручений під дією несправності як у частині вірних, так і в частині невірних розрядів. Помилки, викликувані несправностями у вірних розрядах, знижують достовірність результату й тому називаються суттєвими помилками. Помилки в невірних розрядах не роблять впливу на достовірність результату й тому називаються несуттєвими помилками. Методи контролю є недостовірними у випадку помилкової оцінки результату, тобто при оцінці достовірного результату як недостовірного й недостовірного результату як достовірного. Показник недостовірності D_H контролю результату визначається по наступній формулі [12]:

$$D_H = P_{ПС} + P_{ОН},$$

де $P_{ПС}$ – імовірність пропуску суттєвої помилки результату, при якому недостовірний;

$P_{ОН}$ – імовірність виявлення несуттєвої помилки результату, при якому достовірний результат буде прийнятий за недостовірний.

Імовірність пропуску суттєвої помилки $P_{ПС}$ і ймовірність виявлення несуттєвої помилки $P_{ОС}$ можуть бути визначені, використовуючи наступні формули [17]:

$$P_{ПС} = (1 - P_{ОБ}) K_T; \quad P_{ОС} = P_{ОБ} (1 - K_T),$$

де $P_{ОБ}$ – імовірність виявлення помилки методом контролю;

K_T – імовірність того факту, що помилка є суттєвою.

Імовірність $P_{ОБ}$ визначається відношенням кількості вхідних слів, на яких виявляються помилки, до загальної кількості вхідних слів.

Імовірність K_T визначається, беручи до уваги ряд факторів, що скорочують частку суттєвих помилок результату:

- усунення несуттєвих помилок при відкиданні молодших невірних розрядів результатів, що обчислюються;
- додавання несуттєвих помилок у молодших невірних розрядах, які подовжують мантиси в розширених форматах даних;
- виключення несуттєвих помилок у вірних розрядах результатів операцій, що передують зрушенню мантиси при вирівнюванні порядків із плаваючою точкою, що приводить до втрати цих вірних розрядів.

З огляду на незалежну дію зазначених факторів, імовірність того факту, що помилка, що з'явилася, є суттєвою, може бути оцінена по наступній формулі [17]:

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

- де K_1 – імовірність появи помилки в розрядах результату, що залишаються;
 K_2 – імовірність появи помилки у вірних розрядах результату;
 K_3 – імовірність збереження помилки при вирівнюванні порядків.

Імовірність K_1 знаходиться при аналізі двох протилежних подій: поява помилки в розрядах результату, що залишаються або відкидаються. Імовірності цих подій K_1 та $\bar{K}_1 = 1 - K_1$ визначаються частки помилок у розрядах, що залишаються та відкидаються. Для матричного помножувача ймовірності K_1 , K_2 та K_3 оцінюються, виходячи із припущення про рівномірний розподіл несправностей у цифровій схемі. Імовірність K_1 обчислюється по формулі

$$K_1 = O_{OCT} / O_{OБЧ}, \quad K_1 \leq 1, \quad (1.3)$$

- де O_{OCT} – кількість устаткування, несправності якого викликають арифметичні помилки в розрядах результату, що залишаються;
 $O_{OБЧ}$ – кількість устаткування, несправності якого викликають арифметичні помилки у всіх обчислюваних розрядах результату.

Ймовірність K_2 оцінюється з урахуванням рівномірного розподілу несправностей по формулі [17]

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

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

Розрядність мантис персональних ЕОМ може збільшуватися від 24 біт у форматі з одинарною точністю до 64 біт у подвійному розширеному форматі із плаваючою точкою, що визначає ймовірність $K_2 \leq 0,36$.

Ймовірність K_3 може бути оцінена по формулі [17]

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

де d – величина зсуву мантиси, $\delta O_C = O_C / O_O$; O_C – кількість устаткування вузлів ОП, що передують даному зсувачу мантис; O_O – загальна кількість устаткування ОП; .

При рівноймовірних значеннях величини зсуву, її середнє значення становить половину розрядності мантиси, тобто $d = n / 2$, а формула для ймовірності приймає наступний вид [17]:

$$K_3 = 1 - 0,25 \delta O_C.$$

Для декількох операцій вирівнювання порядків, виконуваних незалежно в даному ОП, ймовірність K_3 визначається як добуток ймовірностей, полічених для кожної із цих операцій. Описані оцінки є наближеними, оскільки не враховують вид типових несправностей матричних пристроїв. Для уточнення ймовірностей пропуску й виявлення суттєвих і несуттєвих помилок контролем по нерівностям, а також показників недостовірності необхідно виконати моделювання обчислювальних процесів, що протікають в умовах дії кратних несправностей.

1.5 Висновки

Основна частина чисел належить до множини наближених даних, і їхня частка в комп'ютерній обробці постійно росте. Ключовою операцією в обробці наближених даних є операція множення, бо присутня в самому поданні наближеного числа у форматі із плаваючою точкою. Сучасні високі вимоги до продуктивності ОП визначають виконання множення в матричних пристроях.

Множення з одинарною точністю може виконуватися у двох варіантах: – повному та скороченому. Скорочене множення майже вдвічі спрощує пристрій і підвищує його швидкодію.

Робоче діагностування матричних помножувачів традиційно виконується з використанням контролю по модулю як основного методу перевірки арифметичних операцій. Однак контроль по модулю суттєво ускладнюється при перевірці скороченого множення. Крім того, цей метод виявляє помилки, викликувані несправностями в невірних розрядах результату, що відкидають при округленні. Граничний контроль має ту перевагу перед контролем по модулю, що він простіший та не враховує розряди результату, що відкидаються.

Крім того, цей метод оцінює величину помилки, виявляючи помилки в старших розрядах з більшою ймовірністю в порівнянні з молодшими розрядами.

У старших вірних розрядах результату виникають суттєві помилки для достовірності наближеного результату, а в молодших невірних розрядах – несуттєві помилки.

Однак метод граничного контролю недостатньо вивчений, і сучасні системи моделювання не дозволяють його досліджувати в умовах дії кратних несправностей матричного помножувача.

Вирішення цієї задачі вимагає розробки спеціальної програми моделювання граничного контролю помножувача мантис, що виконує повну та скорочену операції під дією кратних несправностей.

2 ВИЗНАЧЕННЯ МЕТИ ТА ЗАДАЧ ДОСЛІДЖЕННЯ

2.1 Найменування, область застосування та призначення досліджень

2.1.1 Магістерська робота виконана по темі «Дослідження системи граничного контролю помножувача із зростанням кількості відмов» і присвячена дослідженню процесу граничного контролю роботи матричного помножувача мантис під дією його кратних несправностей при виконанні повної й скороченої операції.

2.1.2 Областю застосування розробки є засоби робочого діагностування компонентів сучасних комп'ютерних систем, які характеризуються високими показниками продуктивності, діапазону чисел та вимог до достовірності результатів, що обчислюються.

Об'єктом робочого діагностування є однокітний матричний помножувач мантис, що застосовується для обробки даних у форматах із плаваючою точкою в функціональних пристроях конвеєрних та матричних процесорів, системах наддовгих команд і СуперЕОМ. Граничний контроль по нерівностям використовується для оцінки достовірності результатів обробки приблизних даних.

2.1.3 Магістерська робота спрямована на виконання дослідження граничного контролю для визначення достовірності результатів, що обчислюються у матричному помножувачі мантис під дією його типових несправностей.

За об'єкт дослідження розглядається процес граничного контролю матричного помножувача мантис одинарної точності.

Предметом дослідження є достовірність граничного контролю матричного помножувача мантис одинарної точності під дією його типових несправностей.

2.1.4 Дані дослідження призначені для вивчення процесу граничного контролю результатів, що обчислюють у матричному помножувачі мантис під дією його кратних несправностей типу «закоротка».

Аналізується ймовірність пропуску й виявлення контролем по нерівностях суттєвих і несуттєвих помилок, викликаних кратними несправностями матричного устрою при виконанні повного й скороченого множення.

Оцінюється вірогідність граничного контролю наближених результатів, що обчислюють, для випадків повного й скороченого множення.

2.1.5 Для проведення досліджень розробляється програмна модель граничного контролю матричного помножувача мантис, виконаного у двох варіантах з обчисленням повного та усіченого добутку.

Для верифікації розробленої моделі граничного контролю передбачаються можливості завдання одиночних значень операндів, а також певного виду несправності і її місця в матричному устрої.

Для проведення основних досліджень граничного контролю використовуються режими довільного завдання місця й типу несправності «закоротка», що діє в межах окремо взятого осередку устрою.

Діагностичний експеримент виконується на послідовності випадкових значень операндів матричного помножувача, у якості яких виступають нормалізовані двійкові мантиси. Тривалість експериментів визначається часом одержання сталих значень результатів моделювання.

2.2 Необхідність проведення досліджень

2.2.1 Стрімкий розвиток комп'ютерних систем, що орієнтовані на виконання приблизних обчислень, спонукають відповідно вдосконалювати методи та засоби робочого діагностування.

Традиційні методи робочого діагностування склалися в умовах домінування теорії та практики побудування повністю самоперевіряємих цифрових схем в заданому класі несправностей, що охопило не тільки керуючі пристрої, але й арифметичні вузли та блоки сучасних комп'ютерних систем.

Однак теорія та практика побудування повністю самоперевіряємих цифро-

вих схем визначила умови захищеності та самотестування в заданому класі несправностей, що обмежує розвиток робочого діагностування в межах обробки точних даних, тобто цілих за своєю природою.

Така обмеженість робочого діагностування насамперед проявилася у неспроможності традиційних методів ефективно підтримувати обчислювальний процес у сучасних комп'ютерних системах.

2.2.2 Для виправлення такого стану необхідно розвивати методи робочого діагностування, що враховують особливості обробки приблизних даних.

Одна з таких особливостей полягає в різній суттєвості помилок, що викликаються несправностями цифрових схем арифметичних пристроїв щодо достовірності обчислюваних результатів.

Тому найбільший попит складається на методи робочого діагностування, що розрізняють суттєві та несуттєві помилки за її величиною.

До таких методів належить граничний контроль, який доцільно розвивати у напрямку підтримки арифметичних пристроїв для обробки мантис.

Враховуючи ключовий характер операції множення, що присутня у самому запису приблизного числа у форматах із плаваючою точкою, доцільно вдосконалювати граничний контроль насамперед для помножувачів мантис.

А приймаючи до уваги сучасні вимоги до продуктивності арифметичних пристроїв, варто розвивати граничний контроль для однокатних матричних помножувачів мантис, що реалізують цифрову схему з матричним паралелізмом.

Розробку та дослідження граничного контролю однокатного матричного помножувача мантис слід виконати моделюванням процесу діагностування в умовах дії найбільш ймовірних несправностей, за які для сучасних цифрових схем визнають замикання сусідніх розрядів двійкових кодів.

Для проведення моделювання необхідно розробити функціональну, наочну та достовірну програмну модель граничного контролю в однокатному матричному помножувачі мантис, що забезпечить дослідження цифрової схеми під дією кратних відмов на послідовностях робочих слів при прийнятих способах обробки мантис.

До таких способів належать обчислення округлених результатів з виконанням повної операції множення мантис або їх обробку за методом скороченого множення. Найважливішим питанням у комп'ютерній обробці числових даних залишається достовірність обчислюваних результатів, яка визначається збереженням певної кількості вірних розрядів, незважаючи на вплив несправностей цифрової схеми.

Робоче діагностування цифрових схем арифметичних пристроїв, що виконується з використанням граничного контролю, повинно забезпечувати оцінку результатів щодо їх достовірності в умовах зростання кількості відмов.

Сам граничний контроль недостатньо вивчений щодо робочого діагностування результатів двомісних арифметичних операцій.

До недосліджених питань варто віднести достовірність граничного контролю округлених результатів обробки приблизних даних в однокіткових матричних арифметичних пристроях.

Таким чином, необхідність проведення досліджень граничного контролю однокіткового матричного помножувача мантис з повним або скороченим виконанням операції під дією найбільш ймовірних кратних несправностей цифрової схеми продиктовано вимогами сучасних комп'ютерних систем до продуктивності, виконання приблизних обчислень та достовірності округлених результатів, що одержуються.

2.3 Організація досліджень

2.3.1 Для проведення досліджень розробляються алгоритми та програмна модель граничного контролю однокіткового матричного помножувача мантис у для варіантів повного та скороченого виконання операції.

Передбачається використання режимів конкретного завдання операндів, місця та виду несправності для верифікації розробленої моделі граничного контролю однокіткового матричного помножувача мантис.

Основні дослідження граничного контролю проводяться в умовах дії характерних кратних несправностей матричного пристрою.

На входи одноктного матричного помножувача подаються випадкові послідовності значень вхідних слів, що складені з операндів – двійкових нормалізованих мантис.

Діагностичний експеримент, який виконується над програмною моделлю граничного контролю одноктного матричного помножувача мантис, триває до одержання сталих значень результатів моделювання.

Вид несправності типу «замикання» між двома точками задається довільно в межах окремого осередку матриці пристрою. Несправний осередок також визначається випадковим образом.

2.4 Постановка задачі

Ставиться задача на проведення досліджень граничного контролю множення мантис у матричному пристрої із зростанням кількості відмов.

Для цього необхідно розробити модель граничного контролю повної та скороченої операції в одноктному матричному помножувача мантис, що дозволяє вивчати можливості методу при виникненні суттєвих і несуттєвих помилок.

В основу розробки програмної моделі покладено викладені в огляді робочого діагностування:

- дослідження традиційних методи контролю за їх ефективністю у визначенні достовірності результатів приблизних обчислень;
- дослідження методів побудування матричного помножувача мантис одинарної точності з введенням його типових несправностей;
- дослідження відомих методів робочого діагностування, включаючи метод граничного контролю щодо перевірки результатів обробки мантис за визначенням нижніх та верхніх границь результату в одноктному пристрої.

При розробці програмної моделі граничного контролю помножувача мантис ставляться такі задачі:

- розробляються алгоритми та за ними складається сама програмна модель схеми граничного контролю в одноктачному матричному помножувачі мантис для різних визначень границь результату;

- перевіряється правильність функціонування програмної моделі схеми граничного контролю матричного помножувача мантис одинарної точності у відповідності до вимог наочності, зручності у використанні, функціональності та достовірності – наближеності до реальності.

Наочність програмної моделі варто забезпечити ефективним використанням образотворчих засобів візуального програмування для подання структури та процесу граничного контролю матричного помножувача мантис, інформативною індикацією результатів моделювання.

Зручність у використанні програмної моделі граничного контролю одноктачного матричного помножувача мантис повинна ґрунтуватися на простоті й варіантності завдання режимів роботи, вихідних даних, а також в одержанні результатів моделювання.

Функціональність програмної моделі граничного контролю результатів, що обчислюються в одноктачному матричному помножувачі мантис, повинна забезпечувати:

- одержання вихідних даних від моделі матричного помножувача відповідно до її режимів роботи;

- виконання функцій граничного контролю мантис в одноктачному матричному помножувачі мантис для формування нижньої та верхньої границь результату;

- виконання функцій граничного контролю, що здійснюють порівняння обчисленого наближеного результату обробки мантис з його нижньої та верхньої границями;

- оцінку ймовірності виявлення та пропуску суттєвих і несуттєвих помилок, що викликаються в одноктачному матричному помножувачі мантис кратними ві-

дмовами за несправностями короткого замикання та діагностуються за методом граничного контролю.

Реальність розробленої програмної моделі повинна забезпечуватися правильністю опису її функцій, реалізованих граничним контролем результатів множення мантис у однокітному матричному пристрої, а також моделлю однокітного матричного помножувача мантис з урахуванням реалізованої в ньому моделі несправності замикання двох точок в межах схеми операційного елемента матриці.

На розробленій програмній моделі варто виконати діагностичні експерименти, які оцінюють роботу граничного контролю однокітного матричного помножувача мантис за ймовірністю виявлення та пропуску суттєвих і несуттєвих помилок, що викликані кратними відмовами за несправностями типу замикання.

За результатами проведеного моделювання необхідно дослідити достовірність граничного контролю наближених результатів, що обчислюються в однокітному матричному пристрої із зростанням кількості відмов для випадків повного та скороченого множення мантис при різній розрядності мантис співмножників та різній кількості вірних розрядів добутку.

3 АНАЛІЗ ЗАДАЧІ З ДОСЛІДЖЕННЯ ГРАНИЧНОГО КОНТРОЛЮ МАТРИЧНОГО ПОМНОЖУВАЧА

3.1 Визначення границь результату для контролю матричного помножувача мантис

3.1.1 Граничний контроль арифметичних пристроїв починається із визначення границь результату виконуваної операції.

Для визначення границь результату множення мантис вихідними умовами є особливості подання операндів, тобто співмножників, за які виступають двійкові нормалізовані мантиси.

Операція множення $C = A \cdot B$ визначена для нормалізованих двійкових операндів, тобто для мантис, що перебувають у межах:

$$0,5 \leq A < 1;$$

$$0,5 \leq B < 1.$$

Контрольні нерівності для перевірки правильності виконання операції множення визначаються з використанням верхньої та нижньої границь результату C [16].

З метою одержання простих контрольних обчислень границі результату будемо шукати як функції з найбільш простою лінійною залежністю від операндів A або B .

Визначимо верхню границю результату. Оскільки кожна з мантис A та B менше одиниці, той їхній добуток C менше кожної з них. Із цього витікає визначення верхньої границі C_B результату із простою кусочно-лінійною функцією: $C_B = \min(A, B)$ і контрольна нерівність, що обмежує результат Z_i зверху: $C < C_B$.

Строга нерівність результату і його верхньої границі витікає зі строгих нерівностей, що визначають верхні границі діапазонів подання мантис A та B .

3.1.2 Визначимо нижню границю результату, використовуючи наступну нерівність:

$$(1 - A) \cdot (1 - B) > 0.$$

Ця нерівність справедлива, оскільки кожний з двох співмножників $(1 - A)$ та $(1 - B)$ її лівої частини є позитивним числом. Такий висновок витікає з нерівностей, що визначають верхні границі зміни діапазонів подання нормалізованих мантис A та B .

Після розкриття дужок дана нерівність перетвориться до наступного виду: $A + B - 1 < A \cdot B$ або з урахуванням заміни добутку $A \cdot B$ на C :

$$A + B - 1 < C.$$

Отримана нерівність обмежує результат множення знизу та визначає його нижню границю:

$$C_H = A + B - 1,$$

$$C_H < C.$$

Нижня границя C_H результату C описується простою лінійною функцією від операндів:

3.1.3 Таким чином, добуток мантис чисел розташовується між двома функціями з лінійними залежностями від операндів A та B .

Правильність обчислення операції множення перевіряється її порівнянням з лінійними функціями верхньої та нижньої границі відповідно до контрольних нерівностей, які мають вигляд:

$$A + B - 1 < C < \min(A, B).$$

3.2 Виконання контрольних обчислень

3.2.1 Відповідно до розширених технічних вимог СК повинні бути що самоперевіряємими у класі константних одиночних несправностей, тобто необхідно забезпечити виявлення власних несправностей у процесі функціонування пристрою множення.

Для виконання даної вимоги необхідно формувати дворозрядний код контролю на незалежному встаткуванні.

Розряди цього коду контролю повинні приймати протилежні значення при правильній роботі пристрою множення і його СК, а однакові значення розрядів коду контролю повинні вказувати на неправильне функціонування пристрою множення і його СК.

Ця задача вирішується організацією перевірки результату операції множення шляхом його почергового порівняння з верхньою та нижньою границями, що може бути виконане на тому самому встаткуванні. Така організація перевірки вимагає використання додаткового входу з можливістю установки його значення D у нульове та одиничне значення.

Один з розрядів коду контролю ($E1$) визначається як результат порівняння добутку з величиною C_{BH} , що приймає значення верхньої та нижньої границь по черзі залежно від значення D на додатковому вході.

У визначених умовах результат порівняння міняє своє значення зі зміною значення D , що дозволяє сформувати з них код контролю. Отже, додатковий вхід визначає другий розряд $E2$ коди контролю:

Установлюється:

$$C_{BH} = C_B \text{ для } D = 0 ;$$

$$C_{BH} = C_H \text{ для } D = 1.$$

Тоді при $D = 0$ виконується порівняння $C < C_B$, правдиве при правильній роботі пристрою множення та його СК, і розряди коду контролю приймають протилежні значення: $E1 = 1, E2 = 0$.

При $D = 1$ виконується порівняння $C < C_H$, що при правильній роботі пристрою множення і його СК, тобто при $C_H < C$, установлює розряд $E1$ у нульове значення, а при $C_H = C$ – помилково в одиничне.

3.2.2 Для усунення помилкового значення необхідно скорегувати операцію порівняння у випадку $D = 1$. Це може бути виконане з використанням значення D шляхом збільшення нижньої границі C_H на одиничне значення D з вагою 2^{-n} молодшого розряду мантис – операндів. Таким чином, розряди коду контролю обчислюються по наступних формулах:

$$E1 = S(C < (C_{BH} + D \cdot 2^{-n})),$$

$$E2 = D,$$

де $S(*)$ – результат операції порівняння $*$, укладеної в дужках.

Визначимо формулу для обчислення величини C_{BH} . Для цього зрівняємо формули, що визначають верхню C_B та нижню C_H границі результату. Формула для обчислення нижньої границі C_H результату містить суму мантис-співмножників за винятком одиниці.

Оскільки мантиси двійкових чисел є нормалізованими даними, то кожна з них становить не менш $1/2$, а їхня сума дає одиничне значення розряду перенесення в цілу частину, що далі компенсується вирахуванням одиниці. Із цього витікає, що нижня границя C_H результату є дробовою частиною суми співмножників A та B .

Формула для обчислення верхньої границі C_B результату містить мінімальний зі співмножників A або B , тобто границя C_B також як і границя C_H розташовується в дробовій частині розрядної сітки.

Отже, для обчислення обох границь може використатися функція F виділення дробової частини розрядної сітки.

Під знаком цієї функції повинні перебувати співмножники A та B , що об'єднуються знаком суми для обчислення нижньої границі C_H , а також які доповню-

ються коефіцієнтами $D1$ і $D2$ для виключення найбільшого з них при обчисленні верхньої границі C_B .

3.2.3 Таким чином, формула для обчислення C_{BH} приймає вигляд:

$$C_{BH} = F(D1 \cdot A + D2 \cdot B).$$

Для визначення коефіцієнтів $D1$ і $D2$ необхідно порівняти співмножники A та B , тобто обчислити результат їхнього порівняння $D3$:

$$D3 = S(A < B).$$

Тоді коефіцієнти $D1$ та $D2$ повинні приймати пряме та інверсне значення змінної $D3$ при $D = 0$ і одиничні значення в протилежному випадку, що описується наступними формулами:

$$D1 = D \vee D3;$$

$$D2 = D \vee \neg D3.$$

Код контролю приймає однакові значення розрядів $E_1 = E_2$ при виявленні помилки: нульові значення (якщо результат є нижчим за нижню границю та одиничні значення у випадку результату, що перевищує верхню границю) та різні значення у протилежному випадку.

3.3 Структура програмної моделі граничного контролю

3.3.1 Розробку моделі граничного контролю матричного помножувача мантис доцільно виконати з використанням мовних засобів DELFI.

Структура моделі граничного контролю одноктного матричного помножувача мантис будується з використанням демонстраційних елементів та елементів керування.

Демонстраційні елементи повинні представити:

- вихідні числові дані – мантиси співмножників, а також їхні добутки, що обчислюються з повним та скороченим виконанням операції множення за математичними формулами та за процесом моделювання із внесенням несправностей пристрою;

- обчислювані нижні та верхні границі результату множення нормалізованих двійкових мантис;

- межу, що розподіляє розряди результатів множення мантис на вірні та невірні розряди;

- структурну схему граничного контролю одноктного матричного помножувача мантис;

- функціональну схему формування умов перевірки нижньої або верхньої границь;

- функціональну схему обчислення поточної границі;

- функціональну схему порівняння результату з поточною границею вузлів контрольного блоку, що описують процес формування коду контролю;

- таблицю результатів моделювання граничного

3.3.2 Елементи керування повинні забезпечити виконання режимів моделювання граничного контролю та перегляд роботи окремих блоків схеми граничного контролю, а також процеси виявлення помилок різного виду, і тому в їхній склад варто включити:

- панелі, що зображують блоки структурної схеми граничного контролю і розкривають їхній зміст по щиглику миші;

- панелі, що зображують вузли блоків граничного контролю і розкривають їхній зміст по щиглику миші;

- покажчик, що виконує розподіл результату, що обчислюється, на вірні та невірні розряди;

– панелі, що забезпечують зупинку при виявленні помилки заданого типу – що збільшує або зменшує результат, а також продовження моделювання після зупинки.

3.4 Функціонування програмної моделі граничного контролю

3.4.1 Програмна модель граничного контролю матричного помножувача повинна забезпечувати:

- виконання функцій граничного контролю множення мантис у матричному пристрої;
- показ структурної схеми граничного контролю;
- показ організації блоків граничного контролю;
- індикацію значень контрольних кодів на виходах елементів показуваних структур;
- установку граничного значення, що розрізняє вірні та невірні розряди результату.
- індикацію виявлення та пропуску помилок результату, що є суттєвими та несуттєвими для його достовірності;
- підрахунок та індикацію кількості виявлених та невиявлених помилок, що викривляють обчислений добуток мантис нижче за нижню та вище за верхню його границі;
- підрахунок та індикацію ймовірності виявлення та пропуску невиявлених помилок добутку мантис при повному та скороченому виконанні операції множення під дією несправностей пристрою.

3.5 План проведення досліджень граничного контролю матричного помножувача мантис із зростанням кількості відмов

3.5.1 На першому етапі необхідно переконатися в правильному функціонуванні створеної програмної моделі граничного контролю однокітного матрично-

го помножувача мантис шляхом її перевірки на тестових прикладах, досліджуючи таке:

- режими початкової установки програмної моделі граничного контролю одноктного матричного помножувача мантис;
- зовнішній вигляд демонстраційних елементів, що використовуються у розробленій програмній моделі;
- достовірність результатів моделювання, що одержані за розробленою програмною моделлю.

У початковій установці програмної моделі граничного контролю варто перевірити наступне:

- правильність завдання кількості відмов;
- режимів роботи програмної моделі;
- режимів одержання співмножників;
- значень вихідних даних;
- режимів завдання несправності;
- місця й види несправності;
- розділ результатів на вірні та невірні розряди;
- контроль визначення нормалізованих мантис співмножників
- при уведенні десяткового числа;
- при уведенні або корекції двійкового коду.

3.5.2 Демонстраційні елементи програмної моделі необхідно верифікувати, відслідковуючи

- правильність відображення структури помножувача в частині:
- матриці осередків з урахуванням різної розрядності пристрою;
- структури осередків для першого й наступного рядків матриці пристрою;
- можливість правильного показу всіх припустимих видів несправності та виключення варіантів закоротки однойменних виводів, а також рівнів логічного нуля та одиниці.

У функціонуванні програмної моделі граничного контролю одноктного матричного помножувача мантис необхідно перевірити:

- правильність обчислень, що виконуються у відсутності несправностей пристрою;
- правильність виконання обчислень на конкретних значеннях співмножників;
- правильність виконання обчислень при повному переборі значень співмножників;
- правильність виконання обчислень на послідовності значень співмножників, що задається випадково;
- правильність виконання обчислень для різних розрядностей матричного помножувача;
- правильність обчислень, що виконуються на програмній моделі при введеній несправності конкретного виду та в конкретному місці на заданих значеннях співмножників;
- правильність обчислень, що виконуються на програмній моделі при введеній несправності для повного та скороченого множення.

Достовірність результатів моделювання, що одержуються при виконання граничного контролю одноктного матричного помножувача мантис, перевіряється наступним чином:

- порівнянням результатів обчислень з їх нижніми та верхніми границями на окремих вхідних словах у режимі справного та несправного пристрою одноктного матричного помножувача мантис;
- вибірковою перевіркою правильності роботи окремих осередків матриці одноктного помножувача мантис при відсутності та появи помилки під дією певної несправності;
- оцінкою знаку та величини помилки залежно від місця та виду несправності, що її викликає;
- оцінкою статистики виявлення та пропуску суттєвих та несуттєвих помилок на послідовностях вхідних слів при завданні несправностей, що забезпечують генерацію тільки суттєвих або тільки несуттєвих помилок;

- оцінкою статистики виявлення та пропуску позитивних та негативних помилок на послідовностях вхідних слів при завданні несправностей пристрою, що забезпечують обчислення результату що перевищує верхню границю або є нижчим за його нижню границю;
- спостереженням за функціонуванням демонстраційних елементів програмної моделі та її елементів керування.

3.6 Висновки

Граничний контроль матричного помножувача мантис може бути вивчений в умовах дії кратних відмов шляхом побудови її моделі, що відповідає вимогам інформативності, функціональності та достовірності.

Модель граничного контролю створюється для одержання достовірної оцінки ймовірностей виявлення та пропуску суттєвих та несуттєвих позитивних та негативних помилок на послідовностях вхідних слів.

Достовірність оцінки повинна забезпечуватися достовірністю моделі матричного помножувача та кратних відмов, а також коректною постановкою експериментів. Достовірність моделі повинна забезпечуватися багатобічною перевіркою її функціонування шляхом введення режимів, що дозволяють виконати тестування й налагодження моделі.

Для цього необхідно використати режими граничного контролю окремих операцій на заданих значеннях співмножників для конкретного виду та місця несправності, в умовах справного та несправного функціонування матричного помножувача мантис. Демонстраційні елементи моделі роблять перевірки наочними та оперативними.

Достовірна оцінка ймовірностей виявлення та пропуску помилок повинна бути забезпечена дослідженням граничного контролю на повній множині вхідних слів або у випадку великої розрядності помножувача мантис на їхній випадковій послідовності до одержання стійких результатів.

4 АЛГОРИТМИ ВИКОНАННЯ ГРАНИЧНОГО КОНТРОЛЮ МАТРИЧНОГО ПОМНОЖУВАЧА МАНТИС

Для побудови програмної моделі граничного контролю матричного помножувача мантис необхідно розробити алгоритми формування поточної границі, порівняння результату з поточною границею та формування розрядів коду контролю, а також обчислення ймовірностей виявлення та пропуску суттєвої та несуттєвої помилки під дією кратних відмов.

4.1 Формування поточної границі для граничного контролю результату

4.1.1 На рис. 4.1 показана схема алгоритму формування поточної границі для граничного контролю результату операції множення в одноктактному матричному пристрої.

4.1.2 Алгоритм працює наступним чином.

Блок 2 вводить двійкові коди aa та bb мантис A та B співмножників та їхню розрядність n .

Блок 3 починає формувати поточну границю результату, обчислюючи, згідно з формулою 3.2, суму двійкових кодів aa та bb мантис A та B співмножників.

Блок 4 додає до обчисленої суми її код, що зсунутий на одну позицію праворуч, обчислюючи півтори суми двійкових кодів мантис A та B співмножників.

Блок 5 зсуває одержаний код збільшеної в півтора рази суми на одну позицію праворуч, обчислюючи $0,75$ від суми двійкових кодів мантис A та B співмножників.

Блок 6 додає до обчисленого коду двійкову константу, яка є еквівалентною до значення $7 \cdot 2^{n-4}$, що забезпечує вирахування з цього коду константи $9 \cdot 2^{n-4}$ в додатковому коді.

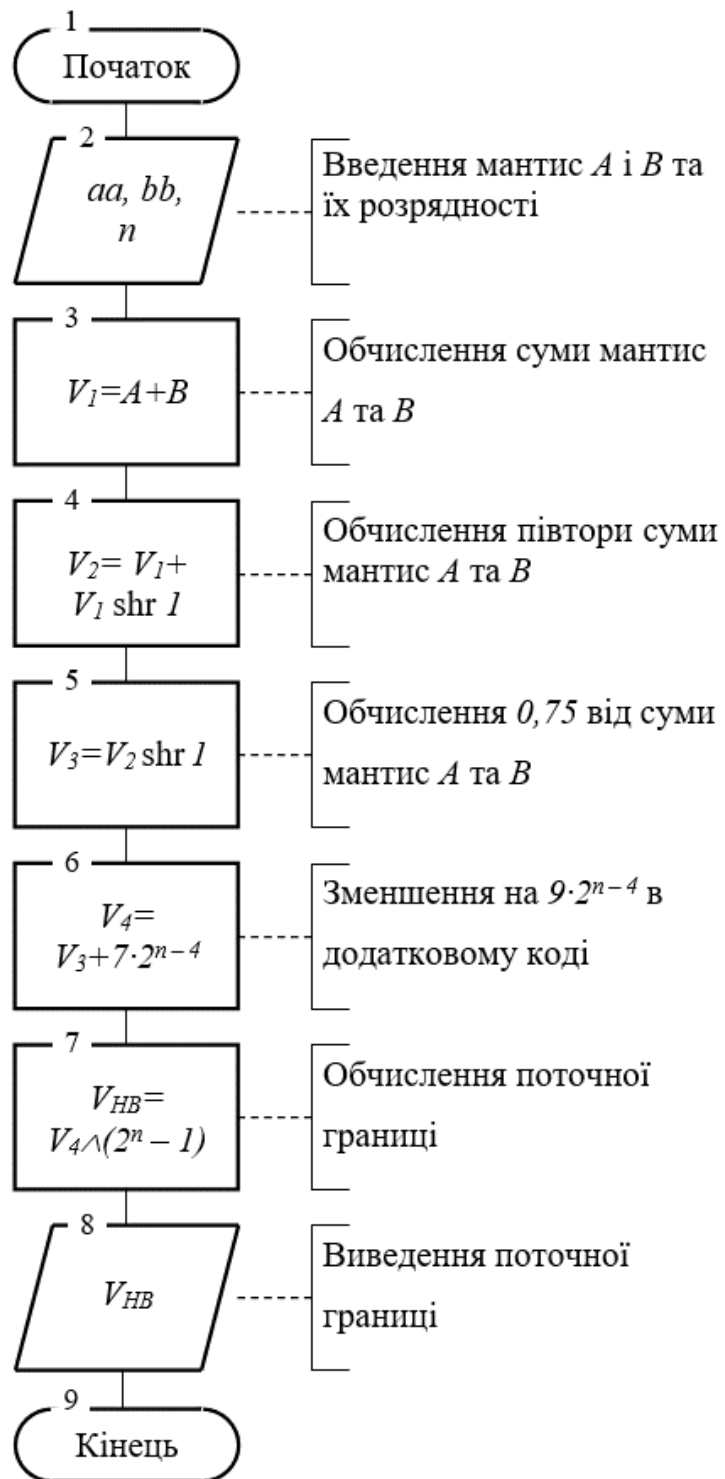


Рис. 4.1 – Схема алгоритму формування поточної границі результату множення мантис

Блок 7 виконує логічне множення розрядів одержаної в додатковому коді суми на розряди двійкової константи з n молодших одиниць, що дорівнює $2^n - 1$, завершаючи обчислення поточної границі результату за формулою (3.2).

Блок 8 виводить обчислену поточну границю V_{NB} результату множення мантис.

Алгоритм закінчено.

4.2 Порівняння результату з поточною границею та формування коду граничного контролю

4.2.1 На рис. 4.2 показана схема алгоритму порівняння результату з поточною границею та формування розрядів коду граничного контролю результату операції множення в одноктактному матричному пристрої.

4.2.2 Алгоритм працює наступним чином.

Блок 2 вводить двійкові коди мантис A та B співмножників, їхню розрядність n , поточну границю результату та сам результат – округлений добуток, що одержано в результаті моделювання роботи матричного помножувача мантис під дією типових несправностей.

Блок 3 обчислює розряд $D = A\{2\} \wedge B\{2\}$ корекції до порівняння результату з поточною границею, що забезпечує віднесення їх спільних точок до допустимих, тобто таких, які не викликають помилку. Обчислення виконується додаванням за модулем два других розрядів двійкових кодів мантис співмножників.

Блок 4 починає формувати перший розряд коду граничного контролю порівнянням результату з його поточною границею, додаючи до результату інверсію поточної границі.

Блок 5 уточнює код виконаного порівняння результату з поточною границею для значення нижньої границі, зміщуючи її у бік збільшення додаванням розряду корекції D .

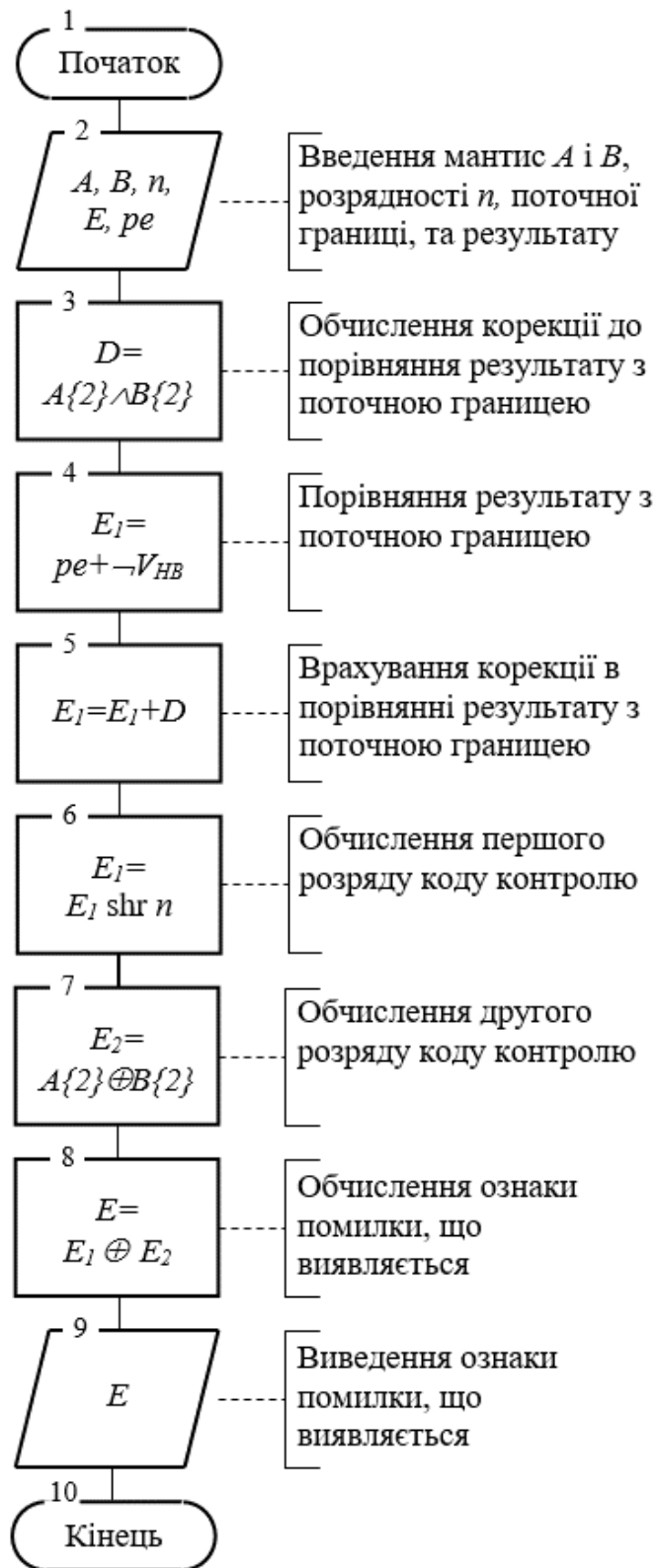


Рис. 4.2 – Схема алгоритму порівняння результату з поточною границею та формування розрядів коду граничного контролю

Блок 6 завершує формування першого розряду коду граничного контролю усунуванням молодших розрядів коду порівняння, зсуваючи його праворуч на n розрядів.

Блок 7 обчислює другий розряд коду граничного контролю матричного помножувача мантис, знаходячи суму по модулю два других розрядів двійкових кодів мантис співмножників.

Блок 8 обчислює ознаку помилки, що виявляється граничним контролем, порівнюючи розряди коду граничного контролю.

Блок 9 виводить ознаку помилки, що виявляється граничним контролем однотактного матричного помножувача мантис.

Алгоритм закінчено.

4.3 Обчислення ймовірності виявлення помилки результату в однотактному матричному помножувачі мантис із зростанням кількості відмов

4.3.1 На рис. 4.3 показана схема алгоритму обчислення ймовірності виявлення помилки результату граничним контролем в однотактному матричному помножувачі мантис.

4.3.2 Алгоритм працює наступним чином.

Блок 2 вводить кількість обчислених результатів множення мантис та кількість помилок, що виявлені граничним контролем, у поточному процесі моделювання роботи однотактного матричного помножувача мантис.

Блок 3 перевіряє факт виявленої помилки обчисленого результату множення мантис при моделюванні однотактного матричного помножувача мантис, порівнюючи ознаку виявленої помилки з одиничним значенням.

При позитивному результаті перевірки наступним виконується блок 4, а інакше блок 5.

Блок 4 видає повідомлення про виявлення помилки результату граничним контролем.

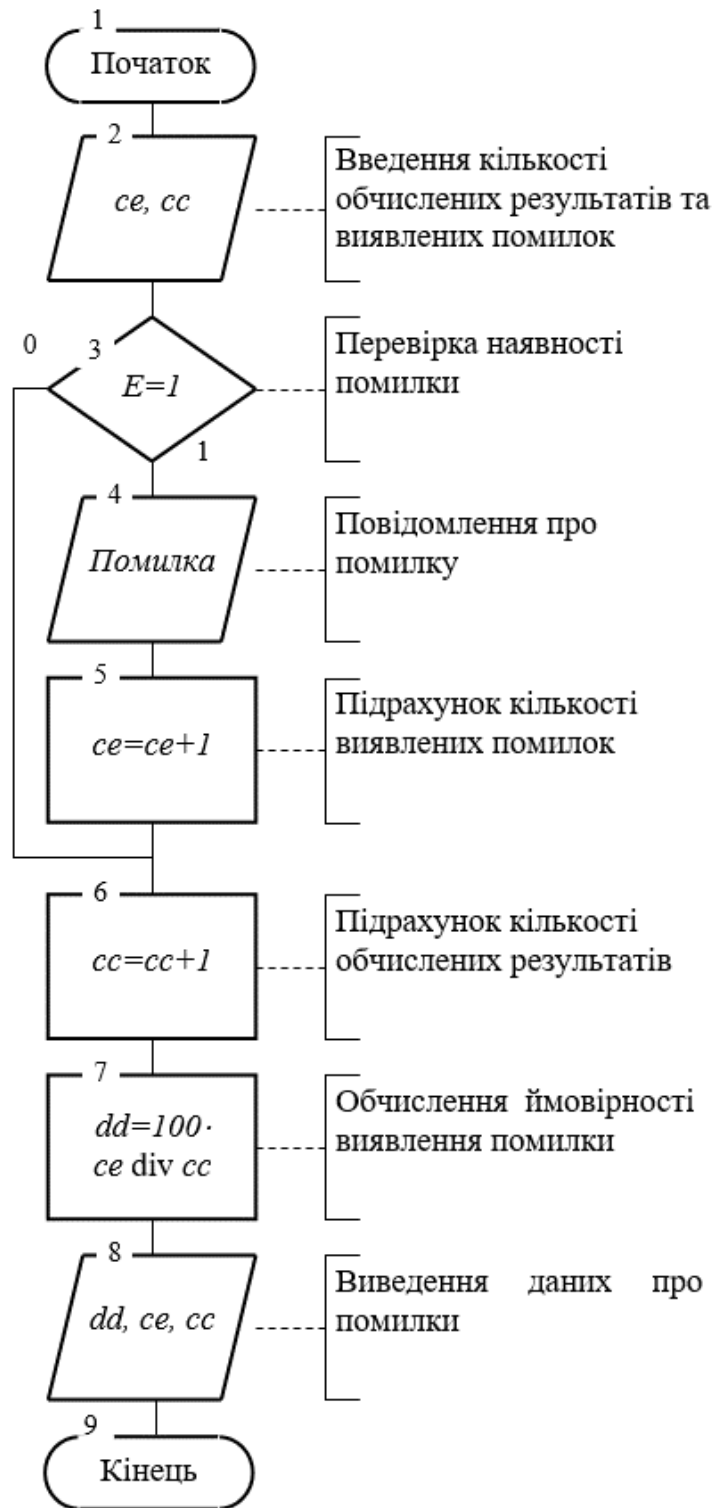


Рис. 4.3 – Схема алгоритму обчислення ймовірності виявлення помилки

Блок 5 збільшує на одиницю кількість помилок, що виявлено граничним контролем з початку процесу його моделювання.

Блок 6 збільшує на одиницю кількість обчислених результатів множення мантис за весь час моделювання.

Блок 7 визначає ймовірність виявлених помилки граничним контролем однотоктного матричного помножувача мантис.

Блок 8 виводить кількість помилок, що виявлено граничним контролем та кількість обчислених результатів множення мантис за весь час моделювання, а також ймовірність виявлених помилки граничним контролем однотоктного матричного помножувача мантис.

Алгоритм закінчено.

4.4 Оцінка ймовірностей виявлення та пропуску суттєвих та несуттєвих помилок в умовах зростання кількості відмов

4.4.1 На рис. 4.4 показана схема алгоритму оцінки ймовірностей виявлення та пропуску помилок результату в матричному помножувачі.

4.4.2 Алгоритм працює таким чином.

Блок 2 вводить вихідні дані до алгоритму: результати множення – правильне значення pp і значення, що обчислене під дією несправності pe , а також ознаку помилки E .

Крім того, вводиться перевищення h кількості невірних розрядів над розрядністю операндів, кількість виконаних операцій ss , а також кількість виявлених суттєвих de та пропущених несуттєвих nn помилок результату множення мантис.

Блок 3 визначає величину помилки, порівнюючи правильний результат множення мантис з результатом, що обчислюється на програмній моделі під впливом несправності.

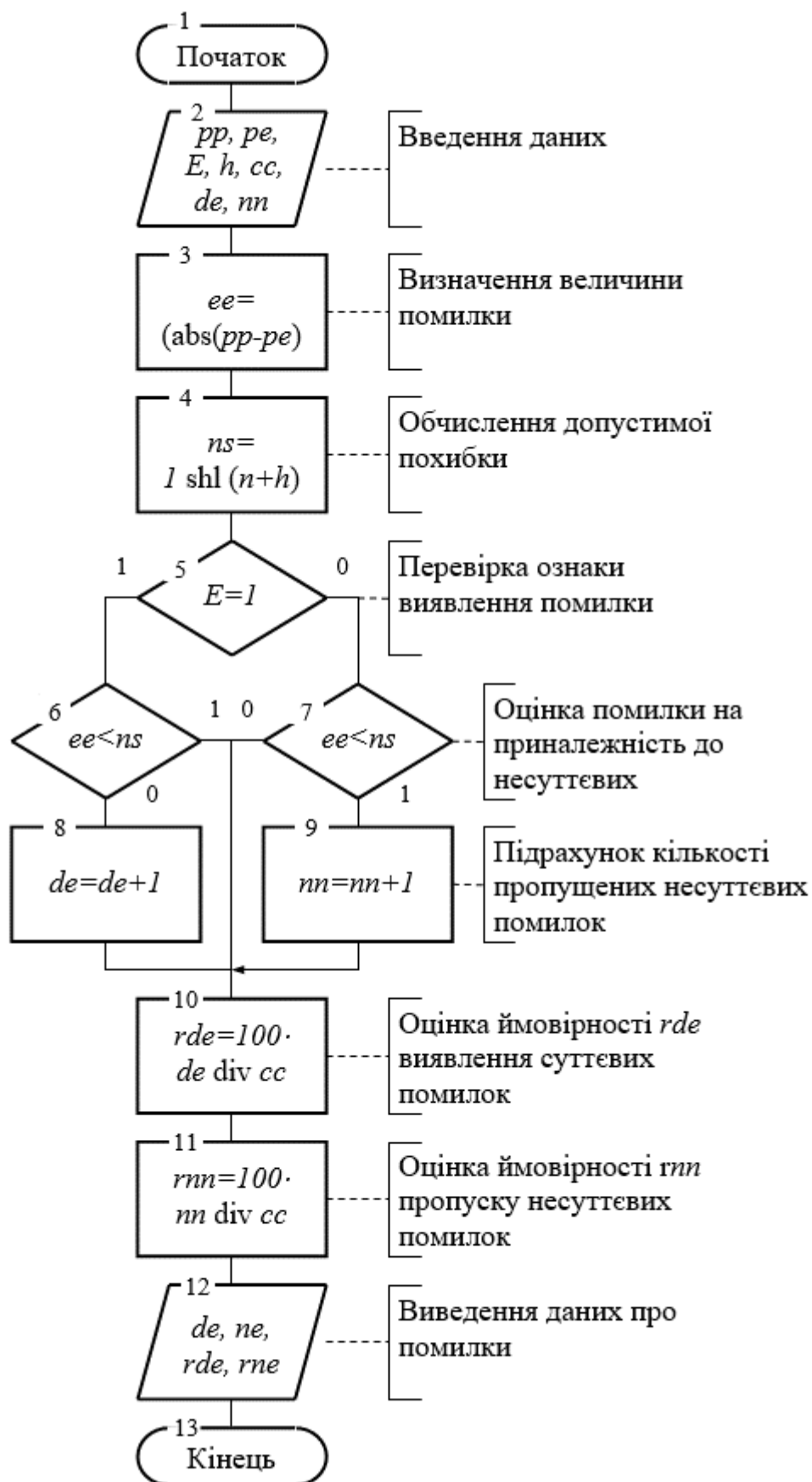


Рис. 4.4 – Схема алгоритму оцінки ймовірностей виявлення суттєвих та несуттєвих помилок

Блок 4 обчислює припустиму погрішність обчислення результату з урахуванням перевищення h кількості невірних розрядів над розрядністю операнда.

Блок 5 перевіряє факт виявлення помилки результату граничним контролем по ознаці помилки.

Блоки 6 та 7 перевіряють умову, за якою помилка результату ідентифікується як несуттєва.

При виконанні умови, що наведена в блоці 5, та невиконанні умови в блоці 6 блок 8 збільшує кількість виявлених граничним контролем суттєвих помилок результату на одиницю.

При невиконанні умови в блоці 5 і виконанні умови в блоці 7 блок 9 збільшує кількість пропущених граничним контролем несуттєвих помилок результату на одиницю.

Блок 10 визначає ймовірність rde виявленої суттєвої помилки а обчислених результатах одноктного помножувача мантис.

Блок 11 оцінює ймовірність rne пропуску несуттєвих помилки в обчислених результатах

Блок 12 виводить дані про помилки: кількість та ймовірність виявлення суттєвих та пропуску несуттєвих помилок.

Алгоритм закінчено.

4.5 Висновки

Для побудування програмної моделі граничного контролю одноктного помножувача мантис необхідно розробити алгоритми виконання контрольних обчислень.

До алгоритмів виконання контрольних обчислень належать такі:

- алгоритм формування поточної границі результату;
- порівняння результату з поточною границею та формування розрядів коду контролю;

- обчислення ймовірності виявлення помилки результату;
- обчислення ймовірностей виявлення та пропуску суттєвої та несуттєвої помилки.

Алгоритм формування поточної границі результату приймає двійкові коди мантис співмножників та їхню розрядність. За цими даними формується сума двійкових кодів мантис співмножників, що спочатку збільшується у півтора рази, а потім зменшується вдвічі та константу додаванням в додатковому коді у відповідності до формули (3.2). Завершується формування поточної границі результату обмеженням її розрядності для усунення перенесень за межі старших розрядів при додаванні у додатковому коді.

Алгоритм порівняння результату з поточною границею та формування розрядів коду контролю приймає результат, що обчислюється при моделюванні роботи матричного помножувача мантис під дією несправностей. Алгоритм здійснює порівняння результату з поточною границею формуючи перший розряд коду контролю. Другий розряд обчислюється за другими розрядами двійкових кодів мантис співмножників.

Алгоритм обчислення ймовірності виявлення помилки результату ідентифікує ознаку виявленої помилки та підраховує кількість виконаних операцій і кількість виявлених помилок, а також оцінює ймовірність виявлення помилки в умовах зростання кількості відмов.

Алгоритм оцінки ймовірностей виявлення та пропуску помилок результату виконує підрахунок кількості виявлених та пропущених суттєвих та несуттєвих помилок з урахуванням установленної границі між вірними та невірними розрядами результату. За отриманим даними оцінюється ймовірності виявлення та пропуску суттєвої та несуттєвої помилки результату у однокітному матричному помножувачі мантис із зростанням кількості відмов.

5 РЕЗУЛЬТАТИ ПРОВЕДЕНИХ ДОСЛІДЖЕНЬ

5.1 Програмна модель граничного контролю матричного помножувача мантис із зростанням кількості відмов

5.1.1 Керівництво програміста. Призначення. Програмний продукт (ПП) призначений для моделювання граничного контролю, що здійснюється в матричному помножувачі мантис. Обробка мантис виконується під дією кратних для матричного помножувача несправностей типу «закоротка».

Склад. Програмний продукт написаний на Delphi, використовує одну форму та містить 18 процедур.

Вихідні дані. Розрядність мантис n задається на початку роботи у діапазоні значень від 8 до 15. За замовчуванням прийнята розрядність 8. Інші дані визначаються в процесі роботи ПП.

Результати моделювання зберігаються на екрані.

5.1.2 Керівництво користувача

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

5.1.2.2 Користувальницький інтерфейс. При натисканні кнопки «Вхід» на екрані з'являється панель опису об'єкта контролю – матричного помножувача мантис, що включає:

- меню режимів завдання операндів і стану пристрою;
- поля й кнопки для завдання та прийому операндів десятковим числом і двійковим кодом;
- структуру матричного помножувача заданої розрядності;
- поля та кнопки завдання місця й виду несправності з можливістю їхнього визначення випадковим образом;
- панель зміни кількості вірних розрядів результату.

Перехід на огляд контрольних обчислень здійснюється шляхом натискання лівої кнопки мишки над панеллю «Контроль». При цьому графічне зображення матричного помножувача змінюється на схему граничного контролю.

Моделювання граничного контролю починається з організації виконання основних обчислень, починаючи з установки режимів завдання операндів. Для цього з меню кожного співмножника вибирається конкретне або довільне значення операнда, повний перебір значень або послідовність довільних значень. Крім того, встановлюється справний або несправний стан пристрою. За замовчуванням прийняте значення операнда й справний стан пристрою. Операнд одержує значення у вигляді десяткового числа або двійкового коду.

У режимі несправного стану пристрою задається вид несправності типу «закоротка» і номер несправного осередку, що може бути встановлено також довільним чином. За замовчуванням результат містить n вірних розрядів. Ця величина може бути зменшена зміною положення покажчика на розділ вірних і невірних розрядів. Після установки необхідних режимів натискається кнопка «Старт», що перейменовується на назву «Фініш» і використовується для завершення моделювання в режимах перебору операндів.

У випадку, коли операнди задаються одним значенням, моделювання виконується спочатку для варіанта обчислення повного добутку, а потім скороченого множення. У процесі моделювання заповнюються поля повного та усіченого добутків, обчислених коректно та під впливом несправності в режимі несправного пристрою. Крім того, вказуються значення результатів та величини помилок у десятковій системі числення.

З метою налагодження можна простежити процес обчислення результату по моделі матричного помножувача шляхом вивчення даних на входах, внутрішніх точках і виходах будь-якого осередку пристрою.

Для цього осередок вибирається щигликом миші, що забезпечує розкриття її структури із вказівкою в її точках значень сигналів. При виклику структури іншого осередку або повторному щиглику по даному осередку її структура згортається у вихідний вид.

На рис. 5.1 показана панель моделі матричного помножувача, що відображає можливості налагодження переглядом роботи його окремих осередків.

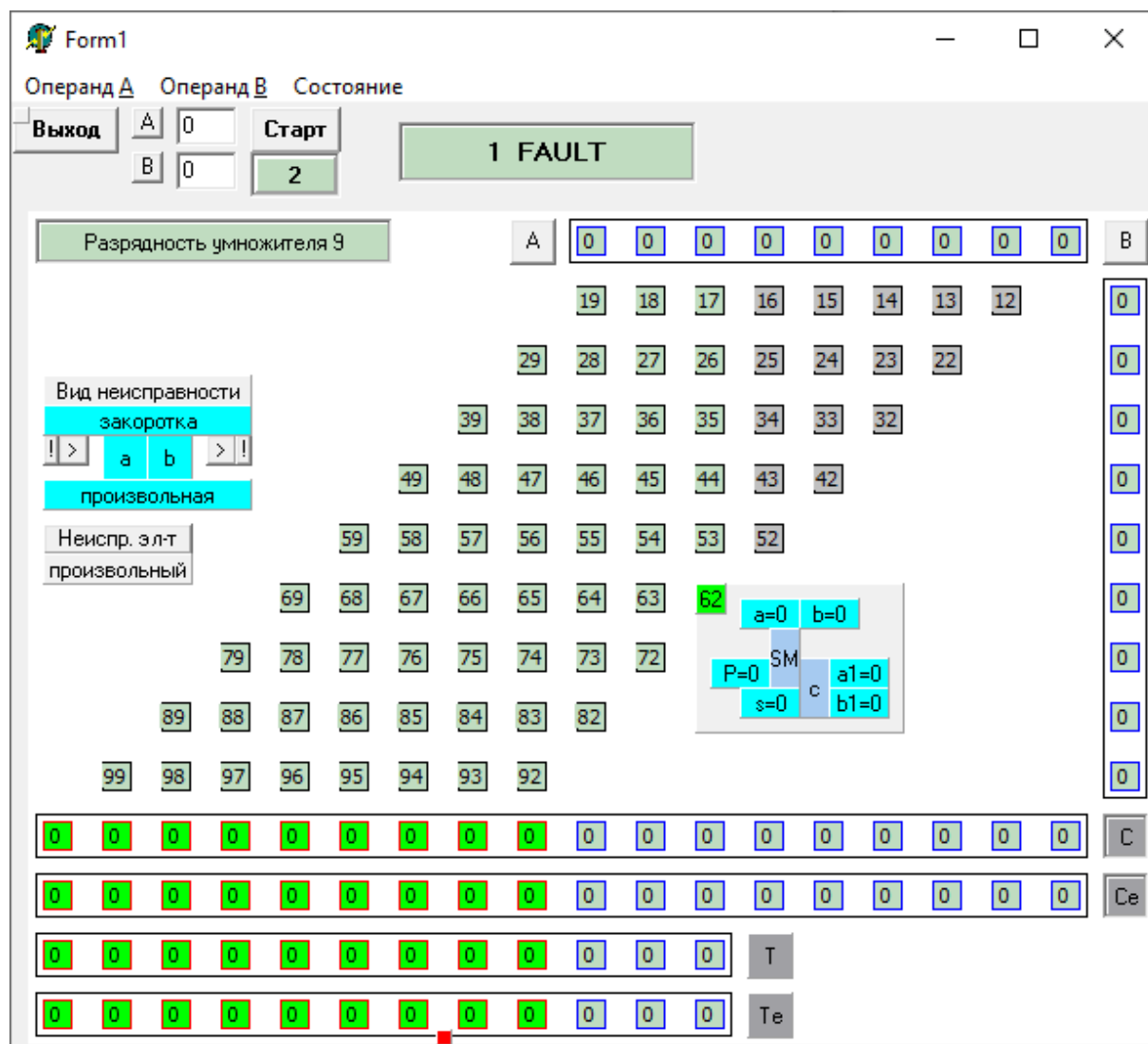


Рис. 5.1 – Панель моделі матричного помножувача, що відображає можливості налагодження переглядом роботи окремих осередків пристрою

У процесі перебору значень виводяться не тільки добутки, що обчислюються, та величини помилок, але також кількість помилок та їхня частка в загальній кількості виконуваних операцій, а також відсоток суттєвих і несуттєвих помилок для випадків виконання повної та скороченої операцій.

Для переходу до графічного зображення граничного контролю виконується щиглик миші по панелі множення «УМН». Це призводить до зміни назви панелі

на контрольну: «КОН» і фіксує поточний стан операндів, результатів, що обчислюються, даних, показуваних на схемі контролю та у таблиці результатів моделювання, а також доповнює демонстрацію повних та скорочених результатів обчислень їх нижньою та верхньою границями. Для повернення вихідної назви панелі та графічного зображення матричного помножувача виконується щиглик миші по перейменованій панелі «КОН».

Перехід на панель граничного контролю та повернення до вихідної панелі може бути виконаний багаторазово, дозволяючи відслідковувати процес моделювання обчислень й оцінювати його проміжні та остаточні результати.

5.1.2.3 В обробці екстрених ситуацій передбачено виведення повідомлень про помилки при порушенні діапазону операндів, що задаються цілим десятковим числом, введенні ненормалізованої мантиси множеного або множника та неправильного номеру несправного операційного елемента пристрою.

5.2 Дослідження ймовірностей виявлення та пропуски помилок граничним контролем в умовах зростання кількості відмов

5.2.1 За результатами моделювання граничного контролю матричного помножувача мантис визначаються наступні дані:

$P_{ОСП}$ – ймовірність виявлення суттєвої помилки в повному добутку;

$P_{ОСПH}$ та $P_{ОСПL}$ – ймовірності виявлення позитивної (H) та негативної (L) суттєвої помилки в повному добутку;

$P_{ОСУ}$ – ймовірність виявлення суттєвої помилки в усіченому добутку;

$P_{ОСУH}$ – ймовірність виявлення суттєвої помилки H в усіченому добутку;

$P_{ОСУL}$ – ймовірність виявлення суттєвої помилки L в усіченому добутку;

$P_{ОНП}$ – ймовірність виявлення несуттєвої помилки в повному добутку;

$P_{ОНПH}$ – ймовірність виявлення несуттєвої помилки H у повному добутку;

$P_{ОНПЛ}$ – ймовірність виявлення несуттєвої помилки L у повному добутку;

$P_{ОНУ}$ – ймовірність виявлення несуттєвої помилки в усіченому добутку;

R_{OHUH} – імовірність виявлення несуттєвої помилки H в усіченому добутку;
 R_{OHUL} – імовірність виявлення несуттєвої помилки L в усіченому добутку;
 $R_{ПСП}$ – імовірність пропуску суттєвої помилки в повному добутку;
 $R_{ПСПН}$ – імовірність пропуску суттєвої помилки H у повному добутку;
 $R_{ПСПЛ}$ – імовірність пропуску суттєвої помилки L у повному добутку;
 $R_{ПСУ}$ – імовірність пропуску суттєвої помилки в усіченому добутку;
 $R_{ПСУН}$ – імовірність пропуску суттєвої помилки H в усіченому добутку;
 $R_{ПСУL}$ – імовірність пропуску суттєвої помилки L в усіченому добутку;
 $R_{ПНП}$ – імовірність пропуску несуттєвої помилки в повному добутку;
 $R_{ПНПН}$ – імовірність пропуску несуттєвої помилки H у повному добутку;
 $R_{ПНПЛ}$ – імовірність пропуску несуттєвої помилки L у повному добутку;
 $R_{ПНУ}$ – імовірність пропуску несуттєвої помилки в усіченому добутку;
 $R_{ПНУН}$ – імовірність пропуску несуттєвої помилки H в усіченому добутку;
 $R_{ПНUL}$ – імовірність пропуску несуттєвої помилки L в усіченому добутку;
 $R_{ОП}$ – імовірність виявлення помилки в повному добутку;
 $R_{ОПН}$ – імовірність виявлення помилки H у повному добутку;
 $R_{ОПЛ}$ – імовірність виявлення помилки L у повному добутку;
 $R_{ОУ}$ – імовірність виявлення помилки в усіченому добутку;
 $R_{ОУН}$ – імовірність виявлення помилки H в усіченому добутку;
 $R_{ОUL}$ – імовірність виявлення помилки L в усіченому добутку;
 $R_{ПП}$ – імовірність пропуску помилки в повному добутку;
 $R_{ППН}$ – імовірність пропуску помилки H у повному добутку;
 $R_{ППL}$ – імовірність пропуску помилки L у повному добутку;
 $R_{ПУ}$ – імовірність пропуску помилки в усіченому добутку;
 $R_{ПУН}$ – імовірність пропуску помилки H в усіченому добутку;
 $R_{ПУL}$ – імовірність пропуску помилки L в усіченому добутку;
 $R_{ВСП}$ – імовірність появи суттєвої помилки в повному добутку;
 $R_{ВСПН}$ – імовірність появи суттєвої помилки H у повному добутку;
 $R_{ВСПЛ}$ – імовірність появи суттєвої помилки L у повному добутку;
 $R_{ВСУ}$ – імовірність появи суттєвої помилки в усіченому добутку;

P_{BCYH} – імовірність появи суттєвої помилки H в усіченому добутку;
 P_{BCYL} – імовірність появи суттєвої помилки в усіченому добутку;
 P_{BHII} – імовірність появи несуттєвої помилки в повному добутку;
 P_{BHIIH} – імовірність появи несуттєвої помилки H у повному добутку;
 $P_{BHII L}$ – імовірність появи несуттєвої помилки L у повному добутку;
 P_{BHV} – імовірність появи несуттєвої помилки в усіченому добутку;
 P_{BHVH} – імовірність появи несуттєвої помилки H в усіченому добутку;
 $P_{BHV L}$ – імовірність появи несуттєвої помилки L в усіченому добутку;
 P_{BII} – імовірність появи помилки в повному добутку;
 P_{BIIH} – імовірність появи помилки H у повному добутку;
 $P_{BII L}$ – імовірність появи помилки L у повному добутку;
 P_{BV} – імовірність появи помилки в усіченому добутку;
 P_{BVH} – імовірність появи помилки H в усіченому добутку;
 P_{BVL} – імовірність появи помилки L в усіченому добутку.

Отримані ймовірності виявлення та пропуску суттєвих та несуттєвих помилок оцінюють достовірність D граничного контролю наближених результатів в одно-тактному матричному помножувачі мантис для випадків повного та скороченого множення відповідно по наступних формулах:

$$D_{II} = (P_{OII} + P_{IIII}) / P_{BII};$$

$$D_{Y} = (P_{OCY} + P_{IIHY}) / P_{BV},$$

де D_{II} – значення достовірності D граничного контролю результатів при обчисленні повного добутку,

D_{Y} – значення достовірності D контролю результатів по нерівностях при обчисленні усіченого добутку.

Ефективність скорочення обчислень в матричному помножувачі мантис може бути оцінена по формулі $\delta D = D_{Y} / D_{II}$.

5.2.2 На рис. 5.2 показана панель моделі граничного контролю у процесі випадкового перебору вхідних слів при довільному завданні одної відмови.

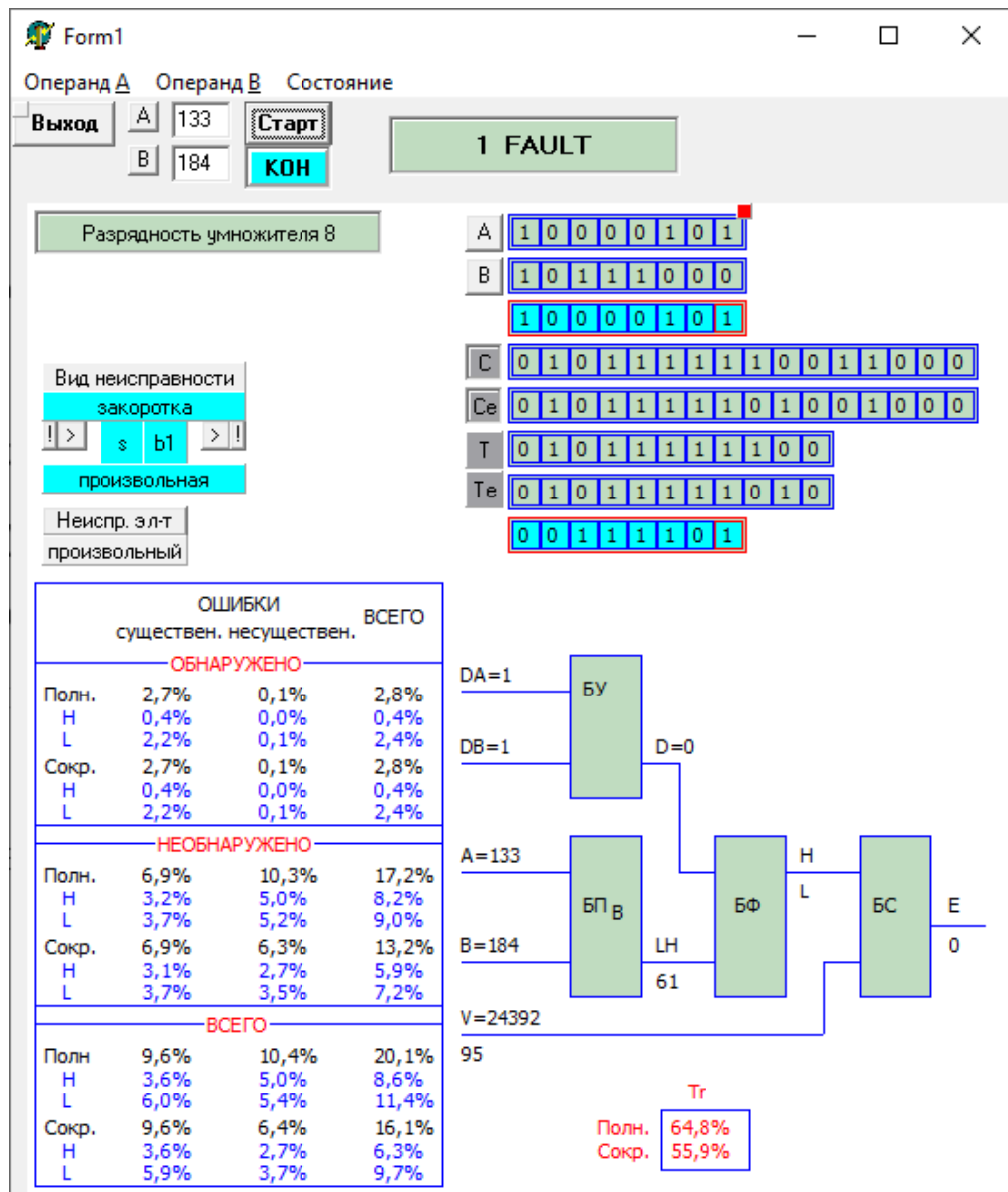


Рис. 5.2 – Панель моделі граничного контролю у процесі випадкового перебору входних слів

На панелі показані значення операндів, обчислених та правильних результатів, а також їх нижня та верхня границі. Крім того, зображено схему граничного контролю матричного помножувача мантис, вихід якої приймає значення відсутності помилки результату множення мантис, а також таблиця результатів моделювання, що містить значення ймовірностей появи, виявлення та пропуску позитивних та негативних суттєвих та несуттєвих помилок.

На рис. 5.3 показана панель моделі граничного контролю матричного помножувача в умовах дії двох відмов.

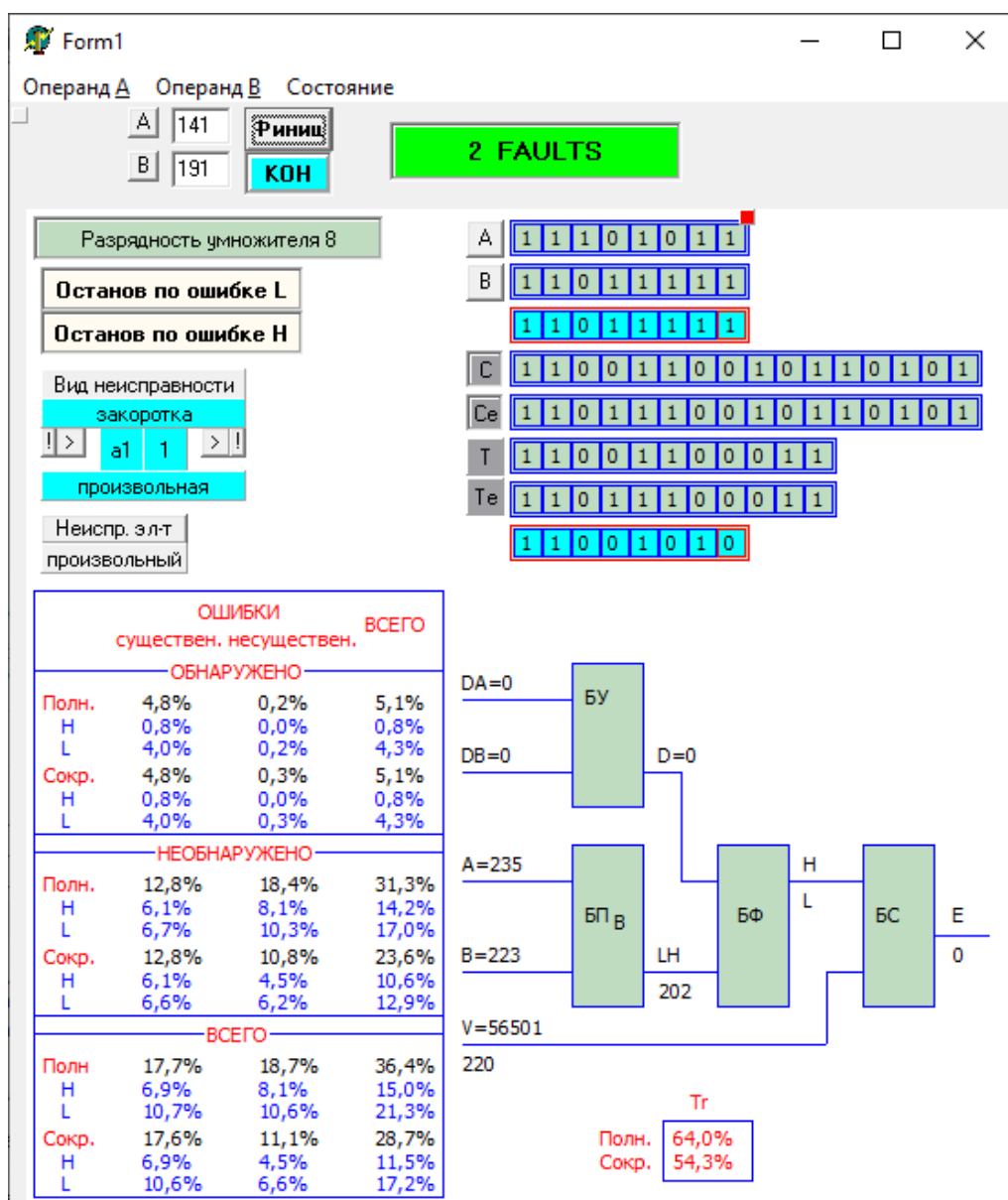


Рис. 5.3 – Панель моделі граничного контролю матричного помножувача в умовах дії двох відмов

Моделювання процесу граничного контролю матричного помножувача мантис проводилося при випадковому виборі несправного осередку пристрою й довільному завданні виду закоротки з повним перебором всіх вхідних слів, а також на випадкових вхідних послідовностях до одержання сталих значень результатів моделювання. Прораховувалися випадки виконання повного та скороченого множення. На панелі показаний режим зупинки по негативній помилці. Червоний індикатор вказує на виявлення помилки, що зменшує результат. Нижня границя перевищує обчислений результат.

На рис. 5.4 показана панель моделі контролю для випадку виявлення помилки в умовах дії трьох відмов.

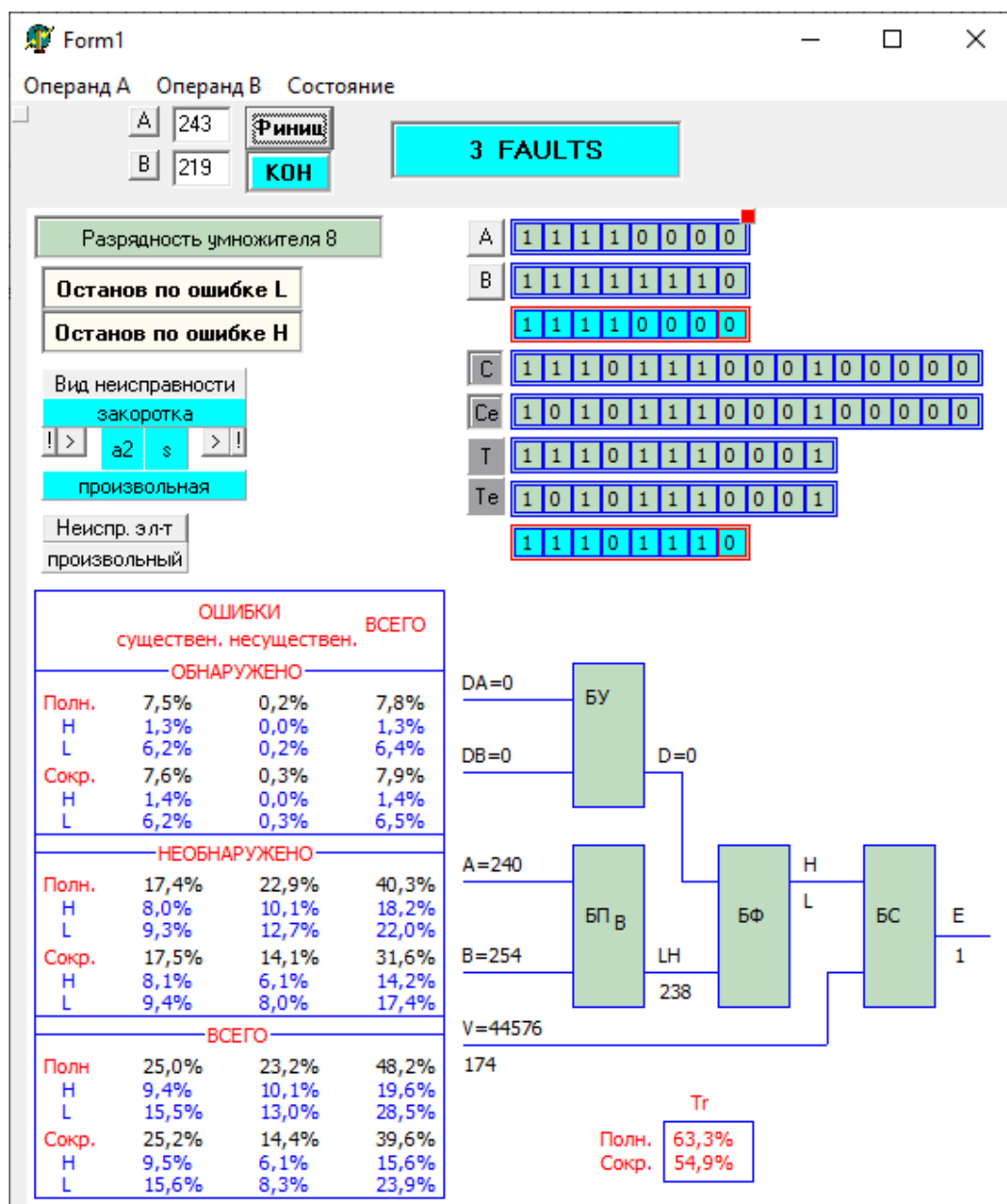


Рис. 5.4 – Панель моделі граничного контролю для випадку виявлення негативної арифметичної помилки в умовах дії трьох відмов

На панелі показано режим зупинки по позитивній помилці. Червоний індикатор указує на виявлення помилки, що збільшує результат. Верхня границя розташовується нижче обчисленого результату.

У табл. 5.1 показане розташування результатів моделювання, одержуваних у ході проведення діагностичного експерименту.

Таблиця 5.1 – Розташування результатів моделювання при їхньому виведенні в процесі проведення діагностичного експерименту

		ПОМИЛКИ		УСЬОГО
		суттєві	несуттєві	
Виявлено	Повне	$R_{ОСП}$	$R_{ОНП}$	$R_{ОП}$
	<i>H</i>	$R_{ОСПH}$	$R_{ОНПH}$	$R_{ОПH}$
	<i>L</i>	$R_{ОСПL}$	$R_{ОНПЛ}$	$R_{ОПЛ}$
	Скорочене	$R_{ОСУ}$	$R_{ОНУ}$	$R_{ОУ}$
	<i>H</i>	$R_{ОСУH}$	$R_{ОНУH}$	$R_{ОУH}$
	<i>L</i>	$R_{ОСУL}$	$R_{ОНУL}$	$R_{ОУL}$
Пропущено	Повне	$R_{ПСП}$	$R_{ПНП}$	$R_{ПП}$
	<i>H</i>	$R_{ПСПH}$	$R_{ПНПH}$	$R_{ППH}$
	<i>L</i>	$R_{ПСПL}$	$R_{ПНПЛ}$	$R_{ППL}$
	Скорочене	$R_{ПСУ}$	$R_{ПНУ}$	$R_{ПУ}$
	<i>H</i>	$R_{ПСУH}$	$R_{ПНУH}$	$R_{ПУH}$
	<i>L</i>	$R_{ПСУL}$	$R_{ПНУL}$	$R_{ПУL}$
УСЬОГО	Повне	$R_{ВСП}$	$R_{ВНП}$	$R_{ВП}$
	<i>H</i>	$R_{ВСПH}$	$R_{ВНПH}$	$R_{ВПH}$
	<i>L</i>	$R_{ВСПЛ}$	$R_{ВНПЛ}$	$R_{ВПЛ}$
	Скорочене	$R_{ВСУ}$	$R_{ВНУ}$	$R_{ВУ}$
	<i>H</i>	$R_{ВСУH}$	$R_{ВНУH}$	$R_{ВУH}$
	<i>L</i>	$R_{ВСУL}$	$R_{ВНУL}$	$R_{ВУL}$

Поля, позначені символами «*H*» і «*L*» указують на моделювання позитивних і негативних помилок, відповідно. Поля «Повне» і «Скорочене» вказують на виконання повного або скороченого множення, відповідно.

Основні й додаткові результати моделювання граничного контролю наведені в табл. 5.2 і 5.3 для випадку, коли всі n розрядів округленого добутку є вірними, тобто $n_T = n$.

Таблиця 5.2 – Імовірності $P_{ОСП}$, $P_{ОСПН}$, $P_{ОСПL}$, $P_{ОСУ}$, $P_{ОСУН}$, $P_{ОСУL}$, $P_{ОНП}$, $P_{ОНПН}$, $P_{ОНПЛ}$, $P_{ОНУ}$, $P_{ОНУН}$, $P_{ОНУL}$, $P_{ПСП}$, $P_{ПСПН}$, $P_{ПСПЛ}$, $P_{ПСУ}$, $P_{ПСУН}$, $P_{ПСУL}$, $P_{ПНП}$, $P_{ПНПН}$, $P_{ПНПЛ}$, $P_{ПНУ}$, $P_{ПНУН}$ і $P_{ПНУL}$ для $n_T = n$

№ п/п	Результати моделювання	n							
		8	9	10	11	12	13	14	15
1	$P_{ОСП}$, %	2,6	2,3	2,0	1,7	1,	1,3	1,0	0,8
2	$P_{ОСПН}$, %	1,2	1,0	0,8	0,7	0,6	0,6	0,5	0,4
3	$P_{ОСПL}$, %	1,4	1,4	1,2	1,0	0,8	0,7	0,5	0,5
4	$P_{ОСУ}$, %	2,7	2,3	2,0	1,7	1,4	1,3	1,0	0,9
5	$P_{ОСУН}$, %	1,2	0,9	0,8	0,7	0,6	0,6	0,5	0,4
6	$P_{ОСУL}$, %	1,5	1,4	1,2	1,0	0,8	0,7	0,5	0,5
7	$P_{ОНП}$, %	0,1	0	0	0	0	0	0	0
8	$P_{ОНПН}$, %	0	0	0	0	0	0	0	0
9	$P_{ОНПЛ}$, %	0,1	0	0	0	0	0	0	0
10	$P_{ОНУ}$, %	0,1	0	0	0	0	0	0	0
11	$P_{ОНУН}$, %	0	0	0	0	0	0	0	0
12	$P_{ОНУL}$, %	0,1	0	0	0	0	0	0	0
13	$P_{ПСП}$, %	7,6	7,7	8,3	8,3	9,0	8,9	9,2	9,3
14	$P_{ПСПН}$, %	3,4	3,3	3,5	3,6	3,7	3,4	3,9	3,8
15	$P_{ПСПЛ}$, %	4,2	4,4	4,8	4,7	5,3	5,5	5,3	5,5
16	$P_{ПСУ}$, %	7,6	7,7	8,2	8,2	9,0	8,9	9,3	9,3
17	$P_{ПСУН}$, %	3,4	3,3	3,5	3,6	3,7	3,4	4,0	3,8
18	$P_{ПСУL}$, %	4,2	4,4	4,7	4,6	5,3	5,5	5,3	5,5
19	$P_{ПНП}$, %	11,5	11,8	11,5	11,9	11,7	11,6	11,8	11,3
20	$P_{ПНПН}$, %	5,1	5,4	5,3	5,5	5,3	5,1	4,9	5,2
21	$P_{ПНПЛ}$, %	6,4	6,3	6,2	6,4	6,3	6,5	6,9	6,2
22	$P_{ПНУ}$, %	7	6,5	5,7	5,6	6,4	5,6	5,4	5,1
23	$P_{ПНУН}$, %	3,0	2,9	2,6	2,5	2,8	2,3	2,1	2,2
24	$P_{ПНУL}$, %	3,9	3,6	3,1	3,0	3,6	3,3	3,3	2,9

Таблиця 5.3 – Імовірності $P_{OP}, P_{OPH}, P_{OPL}, P_{OU}, P_{OUH}, P_{OUL}, P_{PP}, P_{PPH}, P_{PLL}, P_{PU}, P_{PUH}, P_{PUL}, P_{VSP}, P_{VSPH}, P_{VSL}, P_{VSV}, P_{VSVH}, P_{VSVL}, P_{VHP}, P_{VHPH}, P_{VHPL}, P_{VHV}, P_{VHVH}$ і P_{VHVL} для $n_T = n$

№ п/п	Результати моделювання	n							
		8	9	10	11	12	13	14	15
1	$P_{OP}, \%$	2,7	2,4	2,0	1,7	1,4	1,3	1,0	0,8
2	$P_{OPH}, \%$	1,2	1,0	0,8	0,7	0,6	0,6	0,5	0,4
3	$P_{OPL}, \%$	1,5	1,4	1,2	1,0	0,8	0,7	0,6	0,5
4	$P_{OU}, \%$	2,8	2,4	2,0	1,7	1,4	1,3	1,0	0,9
5	$P_{OUH}, \%$	1,2	1,0	0,8	0,7	0,6	0,6	0,5	0,4
6	$P_{OUL}, \%$	1,6	1,4	1,2	1,0	0,8	0,7	0,6	0,5
7	$P_{PP}, \%$	19,1	19,5	19,8	20,2	20,6	20,6	20,9	20,7
8	$P_{PPH}, \%$	8,5	8,8	8,8	9,1	9,0	8,5	8,8	8,9
9	$P_{PLL}, \%$	10,6	10,7	11	11,1	11,6	12	12,2	11,7
10	$P_{PU}, \%$	14,6	14,1	13,9	13,8	15,3	14,5	14,7	14,4
11	$P_{PUH}, \%$	6,4	6,2	6,1	6,1	6,5	5,7	6,1	6,0
12	$P_{PUL}, \%$	8,1	7,9	7,9	7,6	8,8	8,8	8,6	8,4
13	$P_{VSP}, \%$	10,2	10,1	10,3	10	10,4	10,2	10,2	10,2
14	$P_{VSPH}, \%$	4,3	4,3	4,3	4,3	4,3	4	4,4	4,1
15	$P_{VSL}, \%$	5,9	5,8	5,9	5,7	6,1	6,2	5,8	6,0
16	$P_{VSV}, \%$	10,3	10,0	10,3	9,9	10,4	10,2	10,3	10,2
17	$P_{VSVH}, \%$	4,6	4,3	4,3	4,3	4,3	4	4,4	4,2
18	$P_{VSVL}, \%$	5,7	5,7	5,9	5,6	6,1	6,2	5,9	6,0
19	$P_{VHP}, \%$	11,6	11,9	11,5	11,9	11,7	11,6	11,8	11,3
20	$P_{VHPH}, \%$	5,1	5,5	5,3	5,5	5,3	5,1	4,9	5,2
21	$P_{VHPL}, \%$	6,5	6,3	6,2	6,4	6,3	6,5	6,9	6,2
22	$P_{VHV}, \%$	7,1	6,5	5,7	5,6	6,4	5,6	5,4	5,1
23	$P_{VHVH}, \%$	3,1	2,9	2,6	2,5	2,8	2,3	2,1	2,2
24	$P_{VHVL}, \%$	4,0	3,6	3,1	3,1	3,6	3,3	3,3	2,9

Додаткові результати моделювання граничного контролю узагальнюють імовірності виявлення та пропуску суттєвих і несуттєвих помилок. Крім того, у табл. 5.4 дані ймовірності появи позитивних і негативних помилок, а також достовірність граничного контролю результатів для повного D_{II} і скороченого D_Y виконання операції. Для порівняння додатково зазначені відомі значення D_{MII} і D_{MY} достовірності контролю по модулю, а також полічені показники ефективності $\delta D_{II} = D_{II} / D_{MII}$ і $\delta D_Y = D_Y / D_{MY}$ для випадку $n_T = n$.

Таблиця 5.4 – Імовірності $P_{ВП}, P_{ВПН}, P_{ВПЛ}, P_{ВУ}, P_{ВУН}, P_{ВУЛ}, D_{II}, D_Y, D_{MII}, D_{MY}$ й показники $\delta D_{II}, \delta D_Y$ для $n_T = n$

№ п/п	Результати моделювання	n							
		8	9	10	11	12	13	14	15
1	$P_{ВП}, \%$	21,8	21,9	21,8	21,9	22,1	21,8	22,0	21,8
2	$P_{ВПН}, \%$	9,7	9,8	9,6	9,8	9,6	9,1	9,2	9,3
3	$P_{ВПЛ}, \%$	12,2	12,1	12,2	12,1	12,4	12,7	12,7	12,2
4	$P_{ВУ}, \%$	17,3	16,5	16,0	15,5	16,8	15,8	15,7	15,3
5	$P_{ВУН}, \%$	7,6	7,2	6,9	6,8	7,1	6,3	6,5	6,4
6	$P_{ВУЛ}, \%$	9,7	9,4	9,1	8,6	9,7	9,5	9,2	8,9
7	$D_{II}, \%$	64,2	63,0	61,9	60,7	59,6	58,8	58,1	57,4
8	$D_Y, \%$	54,0	50,7	47,7	46,7	45,7	44,2	42,7	41,5
9	$D_{MII}, \%$	46,6	46,6	46,6	46,6	46,6	46,6	46,6	46,6
10	$D_{MY}, \%$	46,6	46,6	46,6	46,6	46,6	46,6	46,6	46,6
11	δD_{II}	1,39	1,38	1,34	1,33	1,29	1,26	1,25	1,19
12	δD_C	1,20	1,14	1,03	1,01	1,00	0,95	0,87	0,84

Основні й додаткові результати моделювання граничного контролю наведені в табл. 5.5 і 5.6 для випадку $n_T = n - 2$.

Таблиця 5.5 – Імовірності $P_{ОСП}$, $P_{ОСПН}$, $P_{ОСПL}$, $P_{ОСУ}$, $P_{ОСУН}$, $P_{ОСУL}$, $P_{ОНП}$, $P_{ОНПН}$, $P_{ОНПЛ}$, $P_{ОНУ}$, $P_{ОНУН}$, $P_{ОНУL}$, $P_{ПСП}$, $P_{ПСПН}$, $P_{ПСПЛ}$, $P_{ПСУ}$, $P_{ПСУН}$, $P_{ПСУL}$, $P_{ПНП}$, $P_{ПНПН}$, $P_{ПНПЛ}$, $P_{ПНУ}$, $P_{ПНУН}$ і $P_{ПНУL}$ для $n_T = n - 2$

№ п/п	Результати моделювання	n							
		8	9	10	11	12	13	14	15
1	$P_{ОСП}$, %	1,9	2,1	1,8	1,7	1,4	1,2	1,1	1,0
2	$P_{ОСПН}$, %	0,8	0,9	0,8	0,7	0,6	0,5	0,5	0,4
3	$P_{ОСПL}$, %	1,1	1,1	0,9	1,0	0,8	0,7	0,6	0,6
4	$P_{ОСУ}$, %	1,9	2,1	1,8	1,7	1,4	1,2	1,1	1,0
5	$P_{ОСУН}$, %	0,8	0,9	0,8	0,7	0,6	0,5	0,5	0,4
6	$P_{ОСУL}$, %	1,1	1,1	1,0	1,0	0,8	0,7	0,6	0,6
7	$P_{ОНП}$, %	0,5	0,3	0,1	0,1	0	0	0	0
8	$P_{ОНПН}$, %	0,2	0,1	0	0	0	0	0	0
9	$P_{ОНПЛ}$, %	0,4	0,2	0,1	0	0	0	0	0
10	$P_{ОНУ}$, %	0,6	0,4	0,1	0,1	0	0	0	0
11	$P_{ОНУН}$, %	0,2	0,1	0	0	0	0	0	0
12	$P_{ОНУL}$, %	0,5	0,2	0,1	0,1	0	0	0	0
13	$P_{ПСП}$, %	3,4	4,2	4,4	5,1	5,5	6,1	6,3	6,7
14	$P_{ПСПН}$, %	1,3	1,9	1,8	2,1	2,3	2,4	2,7	2,5
15	$P_{ПСПЛ}$, %	2,0	2,3	2,5	3,0	3,2	3,7	3,6	4,3
16	$P_{ПСУ}$, %	3,4	4,2	4,4	5,1	5,5	6,1	6,4	6,8
17	$P_{ПСУН}$, %	1,3	1,8	1,8	2,1	2,3	2,4	2,7	2,5
18	$P_{ПСУL}$, %	2,0	2,3	2,6	3,0	3,2	3,7	3,7	4,2
19	$P_{ПНП}$, %	14,6	15,4	15,3	15,3	14,9	15	13,9	14,4
20	$P_{ПНПН}$, %	7,0	7,1	6,8	6,7	6,8	6,7	6,0	6,3
21	$P_{ПНПЛ}$, %	7,7	8,2	8,4	8,6	8,1	8,3	7,9	8
22	$P_{ПНУ}$, %	10,2	10,3	9,5	8,9	9,6	9,2	7,9	7,8
23	$P_{ПНУН}$, %	4,8	4,5	4,2	3,8	4,3	4	3,3	3,4
24	$P_{ПНУL}$, %	5,4	5,8	5,3	5,1	5,3	5,2	4,7	4,4

Таблиця 5.6 – Імовірності P_{OP} , P_{OPH} , P_{OPL} , P_{OU} , P_{OUH} , P_{OUL} , P_{PP} , P_{PPH} , P_{PLL} , P_{PU} , P_{PUH} , P_{PUL} , P_{VCP} , P_{VCPH} , P_{VCL} , P_{VCU} , P_{VCUH} , P_{VCL} , P_{VHP} , P_{VHPH} , P_{VHPL} , P_{VHU} , P_{VHUH} і P_{VHUL} для $n_T = n - 2$

№ п/п	Результати моделювання	n							
		8	9	10	11	12	13	14	15
1	P_{OP} , %	2,5	2,4	1,9	1,8	1,5	1,2	1,1	1,0
2	P_{OPH} , %	1,0	1,1	0,9	0,7	0,7	0,5	0,5	0,4
3	P_{OPL} , %	1,5	1,3	1,0	1,0	0,8	0,7	0,6	0,6
4	P_{OU} , %	2,6	2,4	2,0	1,8	1,5	1,2	1,2	1,0
5	P_{OUH} , %	1,0	1,1	0,9	0,7	0,7	0,5	0,5	0,4
6	P_{OUL} , %	1,6	1,4	1,1	1,0	0,8	0,7	0,6	0,6
7	P_{PP} , %	18	19,6	19,6	20,4	20,4	21	20,2	21,1
8	P_{PPH} , %	8,3	9	8,7	8,8	9,1	9,1	8,7	8,8
9	P_{PLL} , %	9,7	10,6	11	11,6	11,4	12	11,5	12,3
10	P_{PU} , %	13,6	14,4	13,95	13,9	15,2	15,3	14,3	14,6
11	P_{PUH} , %	6,1	6,3	6,0	5,9	6,6	6,4	6,0	5,9
12	P_{PUL} , %	7,5	8,1	7,9	8,0	8,6	8,9	8,3	8,6
13	P_{VCP} , %	5,3	6,3	6,2	6,7	7,0	7,3	7,4	7,7
14	P_{VCPH} , %	2,1	2,8	2,7	2,8	3,0	2,9	3,2	2,9
15	P_{VCL} , %	3,1	3,5	3,5	4,0	4,0	4,4	4,2	4,8
16	P_{VCU} , %	5,3	6,2	6,2	6,8	7,0	7,3	7,5	7,7
17	P_{VCUH} , %	2,1	2,8	2,7	2,8	3	2,9	3,2	2,9
18	P_{VCL} , %	3,2	3,5	3,5	4,0	4,0	4,4	4,3	4,8
19	P_{VHP} , %	15,2	15,7	15,4	15,4	14,9	15,0	14	14,4
20	P_{VHPH} , %	7,1	7,3	6,9	6,8	6,8	6,7	6	6,3
21	P_{VHPL} , %	8,0	8,4	8,5	8,6	8,2	8,3	7,9	8,1
22	P_{VHU} , %	10,9	10,6	9,6	9	9,7	9,2	7,9	7,8
23	P_{VHUH} , %	4,9	4,6	4,2	3,8	4,3	4	3,3	3,4
24	P_{VHUL} , %	5,9	6,0	5,4	5,1	5,4	5,2	4,7	4,4

Зменшення кількості вірних розрядів відповідає реальним умовам обробки наближених даних, коли використовуються розширені формати даних, що збільшують частку невірних розрядів результату. Крім того, до скорочення вірних розрядів приводить виконання операцій вирівнювання порядків.

У табл. 5.7 наведені ймовірності появи позитивних і негативних арифметичних помилок, достовірність контролю результатів D_{Π} , D_{γ} , $D_{МП}$ і $D_{МУ}$, а також полічені показники ефективності δD_{Π} і δD_{γ} для випадку $n_T = n - 2$.

Таблиця 5.7 – Імовірності $P_{ВП}$, $P_{ВПН}$, $P_{ВПЛ}$, $P_{ВУ}$, $P_{ВУН}$, $P_{ВУЛ}$, D_{Π} , D_{γ} , $D_{МП}$, $D_{МУ}$ й показники δD_{Π} , δD_{γ} для $n_T = n - 2$

№ п/п	Результати моделювання	n							
		8	9	10	11	12	13	14	15
1	$P_{ВП}$, %	20,4	22	21,6	22,2	21,9	22,3	21,4	22,1
2	$P_{ВПН}$, %	9,3	10	9,5	9,6	9,8	9,6	9,2	9,2
3	$P_{ВПЛ}$, %	11,2	11,9	12	12,6	12,6	12,2	12,2	12,9
4	$P_{ВУ}$, %	16,2	16,8	15,8	15,7	16,6	16,5	15,5	15,6
5	$P_{ВУН}$, %	7,1	7,4	6,9	6,7	7,2	6,9	6,5	6,3
6	$P_{ВУЛ}$, %	9,1	9,5	9,0	9,1	9,4	9,6	9,0	9,2
7	D_{Π} , %	81,3	79,9	78,1	77,6	74,4	74,0	72,0	70,3
8	D_{γ} , %	74,7	73,8	71,5	67,5	66,2	63,0	58,1	56,4
9	$D_{МП}$, %	24,1	28,2	28,6	30,5	31,8	33,2	34,1	35,0
10	$D_{МУ}$, %	32,7	36,9	39,2	43,3	44,6	44,2	48,4	49,4
11	δD_{Π}	3,37	2,83	2,73	2,54	2,34	2,23	2,11	2,01
12	δD_{γ}	2,28	2,00	1,82	1,56	1,48	1,43	1,20	1,13

Основні й додаткові результати моделювання граничного контролю наведені в табл. 5.8 і 5.9 для випадку $n_T = n - 4$.

Таблиця 5.8 – Імовірності $R_{ОСП}$, $R_{ОСПН}$, $R_{ОСПL}$, $R_{ОСУ}$, $R_{ОСУН}$, $R_{ОСУL}$, $R_{ОНП}$, $R_{ОНПН}$, $R_{ОНПЛ}$, $R_{ОНУ}$, $R_{ОНУН}$, $R_{ОНУL}$, $R_{ПСП}$, $R_{ПСПН}$, $R_{ПСПЛ}$, $R_{ПСУ}$, $R_{ПСУН}$, $R_{ПСУL}$, $R_{ПНП}$, $R_{ПНПН}$, $R_{ПНПЛ}$, $R_{ПНУ}$, $R_{ПНУН}$ і $R_{ПНУL}$ для $n_T = n - 4$

№ п/п	Результати моделювання	n							
		8	9	10	11	12	13	14	15
1	$R_{ОСП}$, %	1,1	1,5	1,4	1,3	1,2	1,1	1,0	0,9
2	$R_{ОСПН}$, %	0,5	0,5	0,7	0,6	0,5	0,5	0,3	0,4
3	$R_{ОСПL}$, %	0,6	0,8	0,8	0,9	0,9	0,7	0,7	0,5
4	$R_{ОСУ}$, %	1,1	1,5	1,4	1,3	1,2	1,1	1,0	0,9
5	$R_{ОСУН}$, %	0,5	0,5	0,7	0,6	0,5	0,5	0,3	0,4
6	$R_{ОСУL}$, %	0,6	0,8	0,8	0,9	0,9	0,7	0,7	0,5
7	$R_{ОНП}$, %	1,8	1,0	0,6	0,4	0,2	0,1	0,1	0
8	$R_{ОНПН}$, %	0,7	0,4	0,2	0,1	0,1	0	0	0
9	$R_{ОНПЛ}$, %	1,1	0,6	0,4	0,2	0,1	0,1	0	0
10	$R_{ОНУ}$, %	1,8	1,0	0,6	0,4	0,2	0,1	0,1	0
11	$R_{ОНУН}$, %	0,7	0,4	0,2	0,1	0,1	0	0	0
12	$R_{ОНУL}$, %	1,1	0,7	0,4	0,2	0,1	0,1	0	0
13	$R_{ПСП}$, %	1,2	1,6	2,0	2,9	3,1	3,6	4	4,6
14	$R_{ПСПН}$, %	0,5	0,7	0,8	1,2	1,2	1,5	1,5	1,8
15	$R_{ПСПЛ}$, %	0,7	0,9	1,2	1,7	1,8	2,1	2,5	2,7
16	$R_{ПСУ}$, %	1,2	1,6	2,0	2,9	3,1	3,6	4,0	4,6
17	$R_{ПСУН}$, %	0,5	0,7	0,8	1,2	1,2	1,5	1,5	1,8
18	$R_{ПСУL}$, %	0,7	0,9	1,1	1,7	1,8	2,1	2,5	2,7
19	$R_{ПНП}$, %	17,7	17,9	17,8	17,7	17	17	17,1	16,5
20	$R_{ПНПН}$, %	8,1	7,9	8	8	7,5	7,5	7,4	7,4
21	$R_{ПНПЛ}$, %	9,7	9,9	9,8	9,7	9,6	9,5	9,8	9,1
22	$R_{ПНУ}$, %	13,2	12,6	12,2	11,2	11,5	11,5	10,6	10,2
23	$R_{ПНУН}$, %	6,0	5,5	5,5	4,9	4,9	4,9	4,5	4,4
24	$R_{ПНУL}$, %	7,3	7,2	6,7	6,3	6,6	6,6	6,1	5,8

Таблиця 5.9 – Імовірності P_{OP} , P_{OPH} , P_{OPL} , P_{OU} , P_{OUH} , P_{OUL} , P_{OP} , P_{OPH} , P_{OPL} , P_{OU} , P_{OUH} , P_{OUL} , P_{VSP} , P_{VSPH} , $P_{VSP L}$, P_{VCSU} , P_{VCSUH} , $P_{VCSU L}$, P_{VHP} , P_{VHPH} , $P_{VHP L}$, P_{VHU} , P_{VHUH} і P_{VHUL} для $n_T = n - 4$

№ п/п	Результати моделювання	n							
		8	9	10	11	12	13	14	15
1	P_{OP} , %	2,9	2,3	2,1	1,9	1,6	1,2	1,0	1,0
2	P_{OPH} , %	1,2	0,9	0,9	0,7	0,6	0,5	0,4	0,4
3	P_{OPL} , %	1,7	1,4	1,2	1,1	1,0	0,7	0,7	0,5
4	P_{OU} , %	2,9	2,3	2,1	1,9	1,6	1,2	1,0	1,0
5	P_{OUH} , %	1,2	0,9	0,9	0,7	0,6	0,5	0,4	0,4
6	P_{OUL} , %	1,7	1,4	1,2	1,1	1,0	0,7	0,7	0,5
7	P_{OP} , %	18,9	19,4	19,8	20,6	20,1	20,6	21,2	21,2
8	P_{OPH} , %	8,6	8,6	8,8	9,2	8,7	9	8,9	9,2
9	P_{OPL} , %	10,3	10,8	10,9	11,4	11,4	11,6	12,3	11,9
10	P_{OU} , %	14,4	14,2	14,1	14,1	14,6	15,1	14,6	14,7
11	P_{OUH} , %	6,5	6,1	6,3	6,1	6,2	6,4	6	6,2
12	P_{OUL} , %	7,9	8,1	7,8	8	8,4	8,7	8,6	8,6
13	P_{VSP} , %	2,3	3,4	4,4	4,5	4,7	4,7	5	5,5
14	P_{VSPH} , %	1,0	1,5	1,8	1,7	1,8	1,9	1,8	2,2
15	$P_{VSP L}$, %	1,3	1,9	2,6	2,7	2,7	2,8	3,2	3,3
16	P_{VCSU} , %	2,3	3,4	4,4	4,5	4,6	4,7	5	5,5
17	P_{VCSUH} , %	1,0	1,5	1,8	1,7	1,8	1,9	1,8	2,2
18	$P_{VCSU L}$, %	1,3	1,9	2,6	2,7	2,7	2,8	3,2	3,3
19	P_{VHP} , %	19,5	18,4	18	17,2	17,1	17,1	17,2	16,6
20	P_{VHPH} , %	8,8	8,2	8,1	7,5	7,5	7,5	7,4	7,4
21	$P_{VHP L}$, %	10,7	10,2	9,9	9,7	9,6	9,6	9,8	9,2
22	P_{VHU} , %	15	12,8	11,6	11,7	11,7	11,6	10,7	10,2
23	P_{VHUH} , %	6,6	5,7	5,1	5	5	5	4,5	4,4
24	P_{VHUL} , %	8,4	7,1	6,6	6,7	6,7	6,7	6,1	5,8

У табл. 5.10 наведені ймовірності появи позитивних і негативних арифметичних помилок, недостовірність граничного контролю результатів та по модулю D_{II} , D_V та D_{MPI} , D_{MV} , відповідно при повному та скороченому виконанні операції множення, а також полічені показники δD_{II} та δD_V ефективності граничного контролю для випадку кількості вірних розрядів $n_T = n - 4$.

Таблиця 5.10 – Імовірності $P_{ВП}$, $P_{ВПН}$, $P_{ВПЛ}$, $P_{ВУ}$, $P_{ВУН}$, $P_{ВУЛ}$, D_{II} , D_V , D_{MPI} , D_{MV} та показники δD_{II} , δD_V для $n_T = n - 4$

№ п/п	Результати моделювання	n							
		8	9	10	11	12	13	14	15
1	$P_{ВП}$, %	21,8	21,7	21,8	22,4	21,7	21,8	22,2	22,1
2	$P_{ВПН}$, %	9,8	9,5	9,7	10	9,3	9,5	9,2	9,7
3	$P_{ВПЛ}$, %	12	12,2	12,1	12,5	12,4	12,3	13	12,7
4	$P_{ВУ}$, %	17,3	16,5	16,2	16	16,2	16,2	15,6	15,7
5	$P_{ВУН}$, %	7,6	7	7,2	6,9	6,8	6,9	6,4	6,6
6	$P_{ВУЛ}$, %	9,7	9,5	9	9,1	9,4	9,4	9,3	9,8
7	D_{II} , %	89,8	88,7	87,7	86,7	84,0	83,6	82,6	79,5
8	D_V , %	82,7	84,2	84,2	84,2	79,6	77,3	74,3	70,7
9	D_{MPI} , %	10,5	15,5	20,1	20,5	21,5	21,5	22,8	2,51
10	D_{MV} , %	13,3	20,6	27,2	28,1	29,0	29,0	32,1	35,0
11	δD_{II}	8,17	5,66	4,36	4,28	3,91	3,84	3,62	3,17
12	δD_V	6,21	4,09	3,10	2,99	2,75	2,67	2,31	2,02

Остаточні результати моделювання визначають оцінки достовірності D_{III} , D_{PI2} , D_{PI3} , D_{VI1} , D_{VI2} , D_{VI3} граничного контролю щодо повного та усіченого результатів множення мантис при кількості вірних розрядів $n_T = n$ та $n_T = n - 6$ для трьох значень кратності несправності: 1, 2 та 3 наведено в табл. 5.11 та 5.12, відповідно.

Таблиця 5.11 – Достовірність $D_{П1.n}$, $D_{П2.n}$, $D_{П3.n}$, $D_{У1.n}$, $D_{У2.n}$, $D_{У3.n}$ для кратних несправностей та $n_T = n$

№ п/п	Результати моделювання	n							
		8	9	10	11	12	13	14	15
1	$D_{П1.n}$, %	64,2	63,0	61,9	60,7	59,6	58,8	58,1	57,4
2	$D_{П2.n}$, %	62,7	61,0	59,3	58,1	56,9	56,1	55,4	55,7
3	$D_{П3.n}$, %	61,3	59,5	57,7	56,4	55,2	54,4	53,7	53,0
4	$D_{У1.n}$, %	54,0	50,7	47,7	46,7	45,7	44,2	42,7	41,5
5	$D_{У2.n}$, %	53,4	49,9	46,0	45,2	44,1	43,0	41,9	40,8
6	$D_{У3.n}$, %	53,0	49,3	45,3	44,3	43,2	41,6	40,0	38,6

Таблиця 5.12 – Достовірність $D_{П1.n-6}$, $D_{П2.n-6}$, $D_{П3.n-6}$, $D_{У1.n-6}$, $D_{У2.n-6}$, $D_{У3.n-6}$ для кратних несправностей та $n_T = n - 6$.

№ п/п	Результати моделювання	n							
		8	9	10	11	12	13	14	15
1	$D_{П1.n-6}$, %	89,7	92,6	94,2	93,3	92,3	90,7	89,1	88,0
2	$D_{П2.n-6}$, %	88,3	91,9	93,2	92,2	91,1	89,7	88,3	87,4
3	$D_{П3.n-6}$, %	87,3	91,1	92,6	91,5	90,4	88,9	87,4	86,8
4	$D_{У1.n-6}$, %	86,4	90,1	92,0	90,1	89,8	87,3	84,8	82,2
5	$D_{У2.n-6}$, %	85,3	89,5	91,2	90,4	89,7	87,0	84,0	81,7
6	$D_{У3.n-6}$, %	84,4	88,7	90,3	89,6	88,9	86,0	83,3	81,4

5.2.3. На рис. 5.5 показані діаграми зміни ймовірностей $P_{ОСП}$, $P_{ОСП H}$ і $P_{ОСП L}$ виявлення суттєвої помилки, а також ймовірності виявлення її позитивного та негативного значень в повному добутку для ряду значень розрядності n матричних помножувачів і кількості вірних розрядів $n_T = n$.

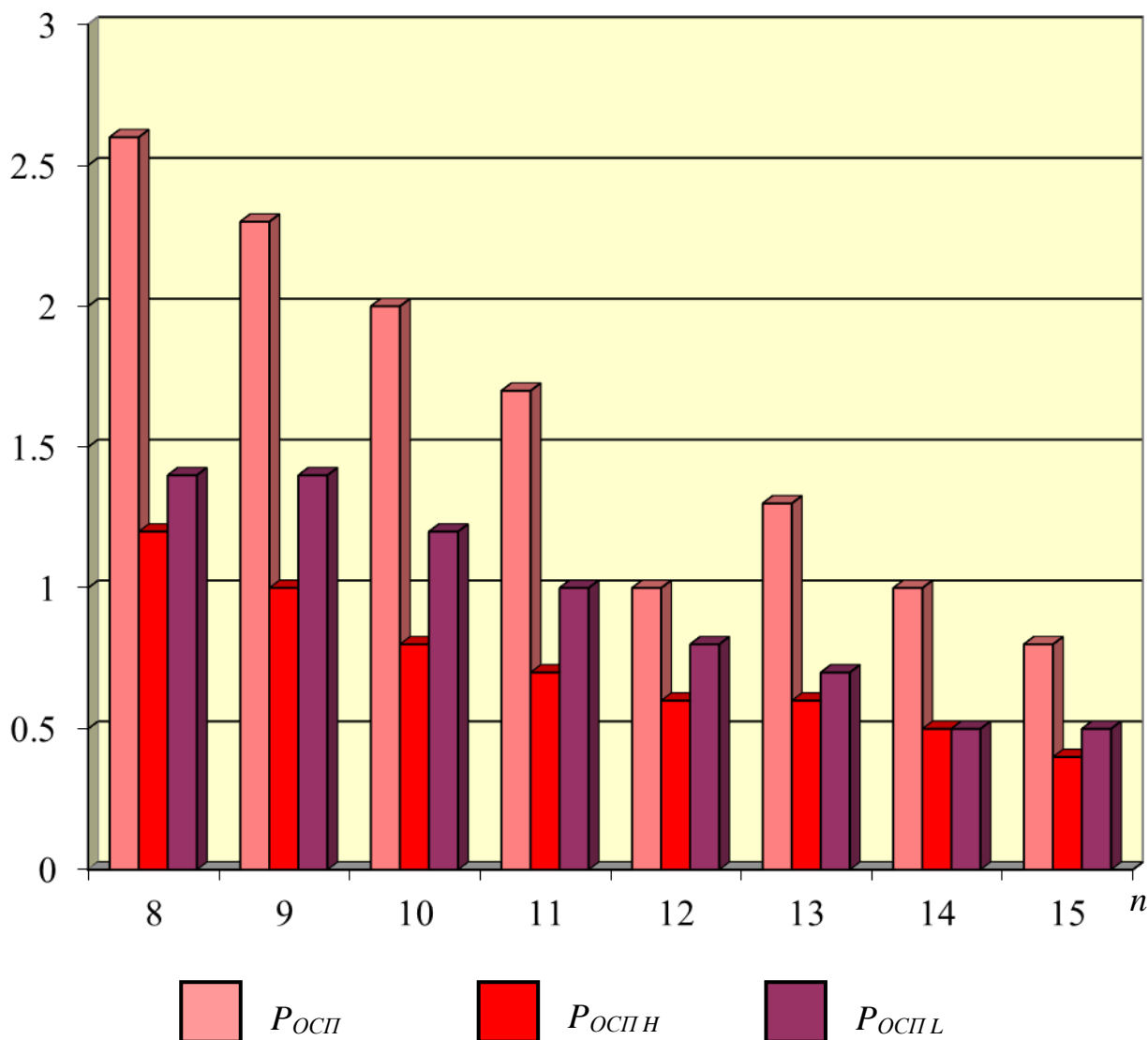


Рис. 5.5 – Діаграми ймовірностей P_{OCP} , P_{OCPH} і P_{OCPH} виявлення суттєвої помилки в повному добутку

Побудовані діаграми показують, що ймовірності виявлення суттєвої помилки в повному добутку зменшуються з ростом розрядності операндів. Позитивна помилка виявляється з меншою ймовірністю в порівнянні з негативною помилкою.

На рис. 5.6 показані діаграми зміни ймовірностей P_{OCY} , P_{OCYH} і P_{OCYL} виявлення суттєвої помилки в усіченому добутку для ряду значень розрядності n матричних помножувачів і кількості вірних розрядів $n_T = n$.

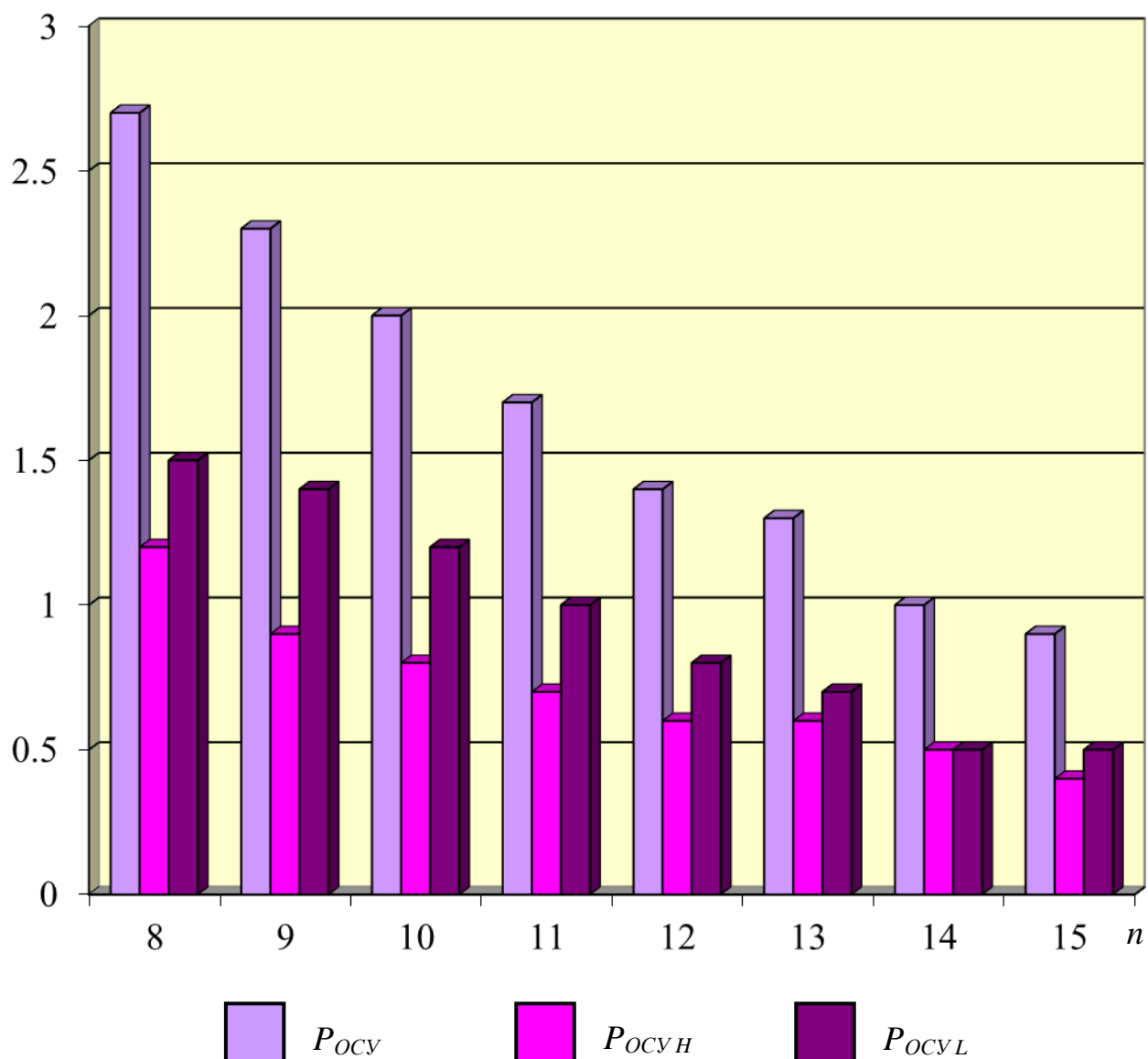


Рис. 5.6 – Діаграми ймовірностей P_{OCV} , P_{OCVH} і P_{OCVL} виявлення суттєвої помилки в усіченому добутку

Побудовані діаграми виявлення суттєвої помилки в усіченому добутку практично повторюють вид діаграм для ймовірностей виявлення суттєвої помилки в повному добутку, які показані на рис. 5.5.

На рис. 5.7 показані діаграми зміни ймовірностей $P_{ПСП}$, $P_{ПСПH}$ і $P_{ПСПЛ}$ пропуску суттєвої помилки в повному добутку для ряду значень розрядності n матричних помножувачів і кількості вірних розрядів $n_T = n$.

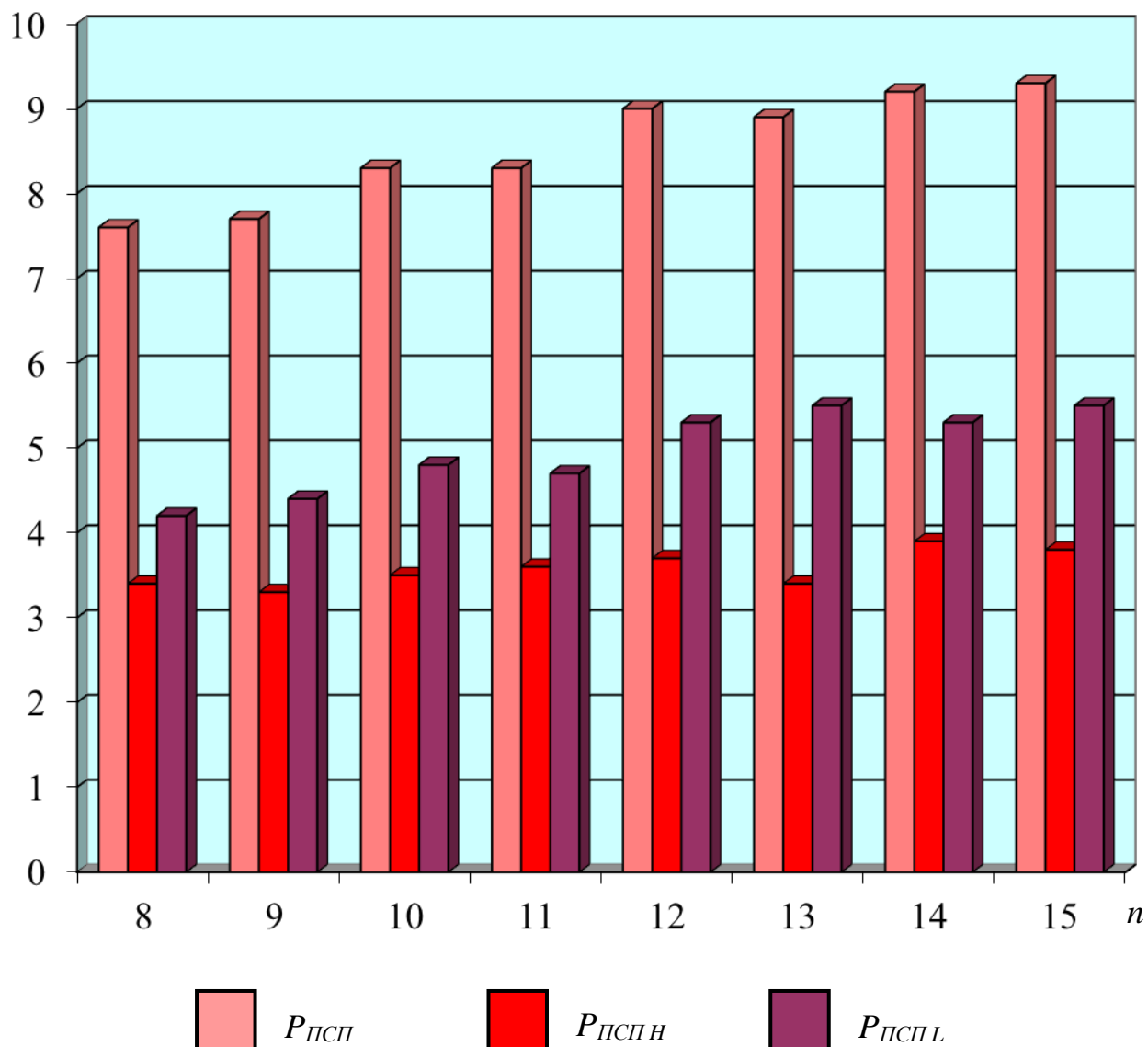


Рис. 5.7 – Діаграми ймовірностей P_{PSP} , P_{PSPH} і P_{PSPH} пропуску суттєвої помилки в повному добутку

Побудовані діаграми показують, що ймовірності пропуску суттєвої помилки в повному добутку збільшуються з ростом розрядності операндів. Позитивна помилка пропускається з небагато меншою ймовірністю в порівнянні з негативною помилкою.

На рис. 5.8 показані діаграми зміни ймовірностей P_{PSPU} , P_{PSPUH} і P_{PSPUH} пропуску суттєвої помилки в усіченому добутку для ряду значень розрядності n матричних помножувачів і кількості вірних розрядів $n_T = n$.

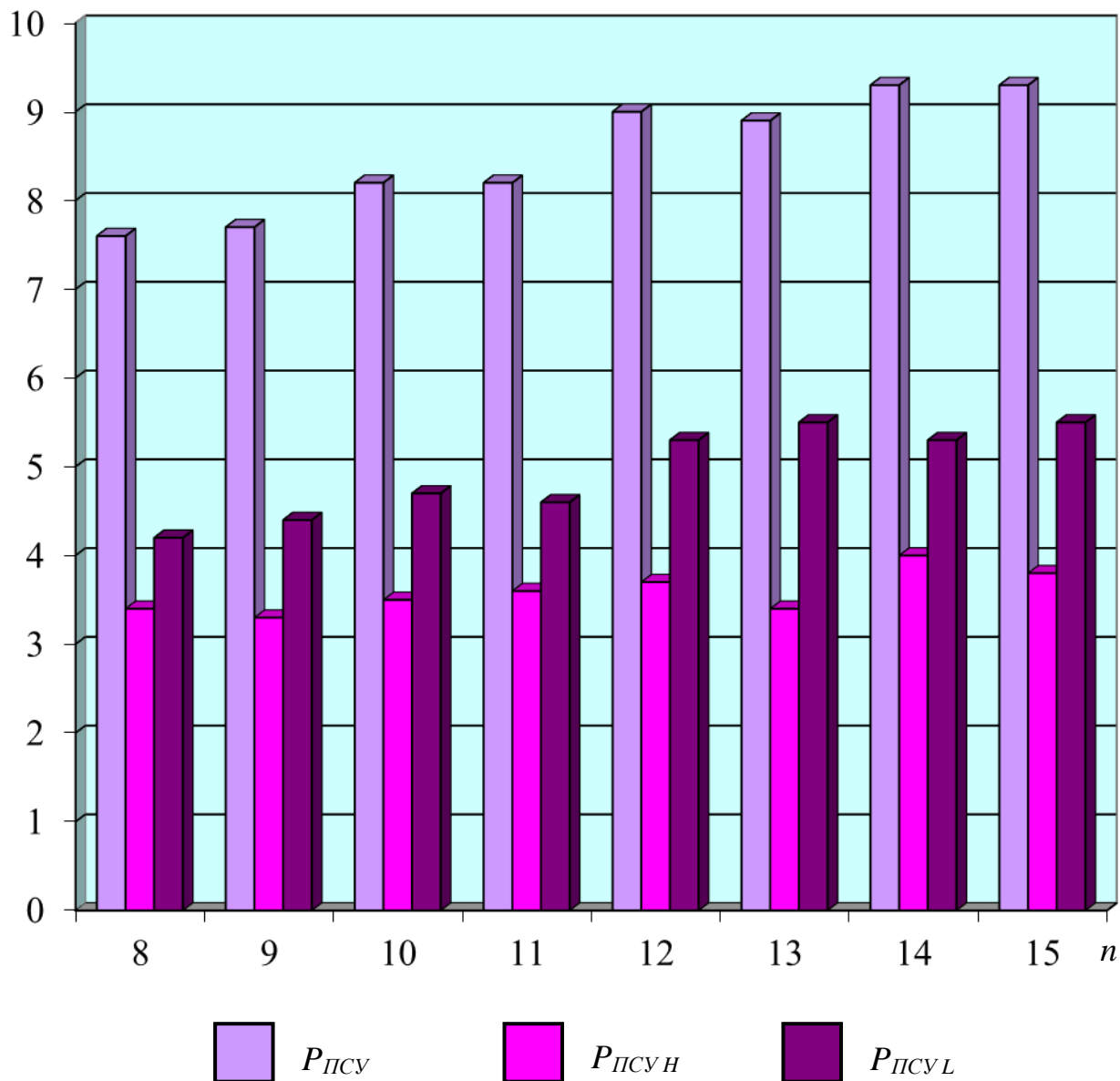


Рис. 5.8 – Діаграми ймовірностей P_{PCU} , P_{PCUH} і P_{PCU_L} пропуску суттєвої помилки в усіченому добутку

Побудовані діаграми пропуску суттєвої помилки в усіченому добутку практично повторюють вид діаграм для ймовірностей пропуску суттєвої помилки в повному добутку, які показані на рис. 5.9.

На рис. 5.9 показані діаграми зміни ймовірностей $P_{ПП}$, $P_{ППН}$ і $P_{ППЛ}$ пропуску несуттєвої помилки в повному добутку для ряду значень розрядності n матричних помножувачів і кількості вірних розрядів $n_T = n$.

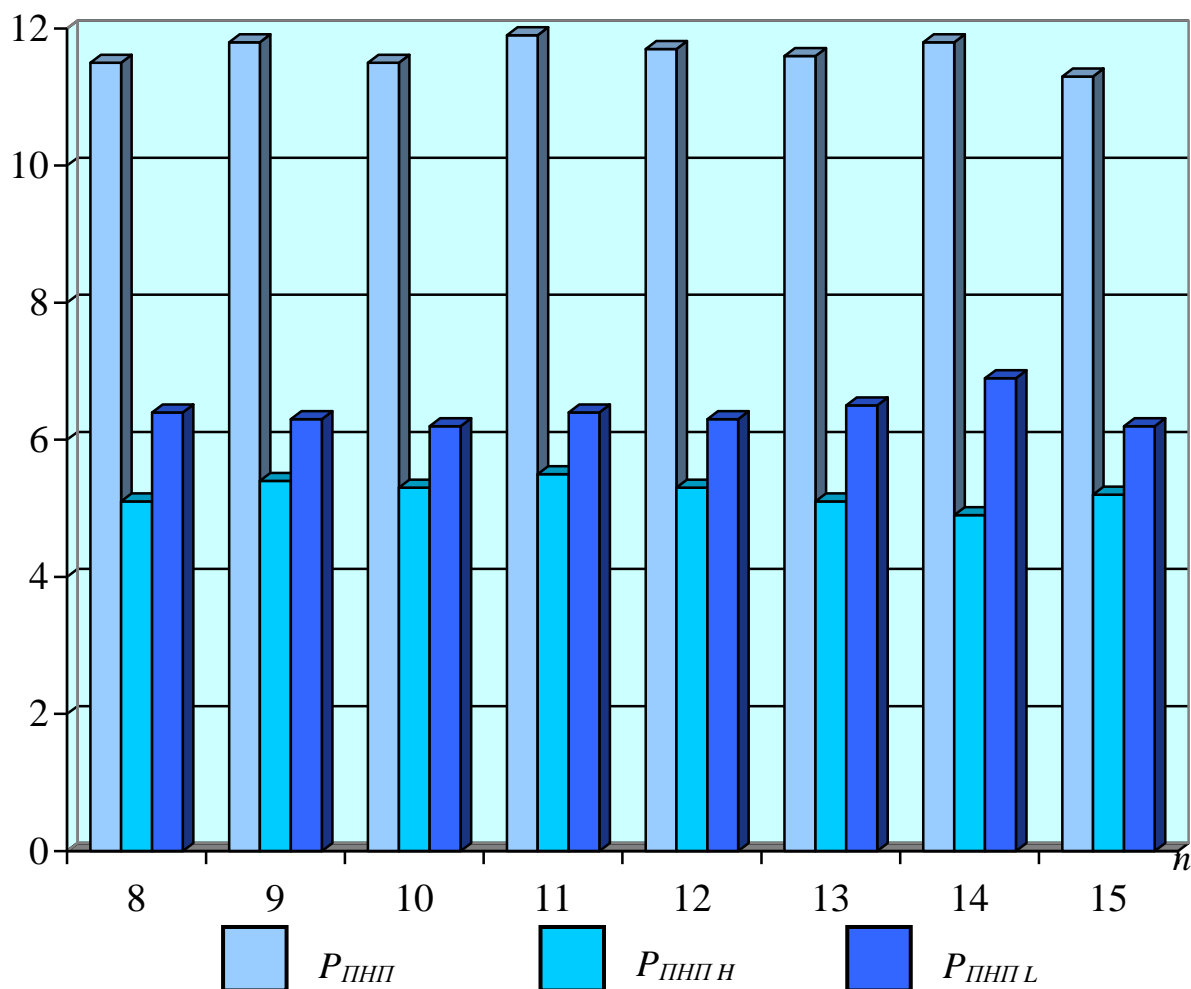


Рис. 5.9 – Діаграми ймовірностей P_{PNP} , P_{PNPn} і P_{PNP_L} пропуску несуттєвої помилки в повному добутку

Побудовані діаграми показують, що ймовірності пропуску несуттєвої помилки в повному добутку незначно зменшуються з ростом розрядності операндів.

Позитивна помилка пропускається з меншою ймовірністю, чим негативна помилка.

На рис. 5.10 показані діаграми зміни ймовірностей P_{PNV} , P_{PNVn} і P_{PNV_L} пропуску несуттєвої помилки в усіченому добутку для ряду значень розрядності n матричних помножувачів і кількості вірних розрядів $n_T = n$.

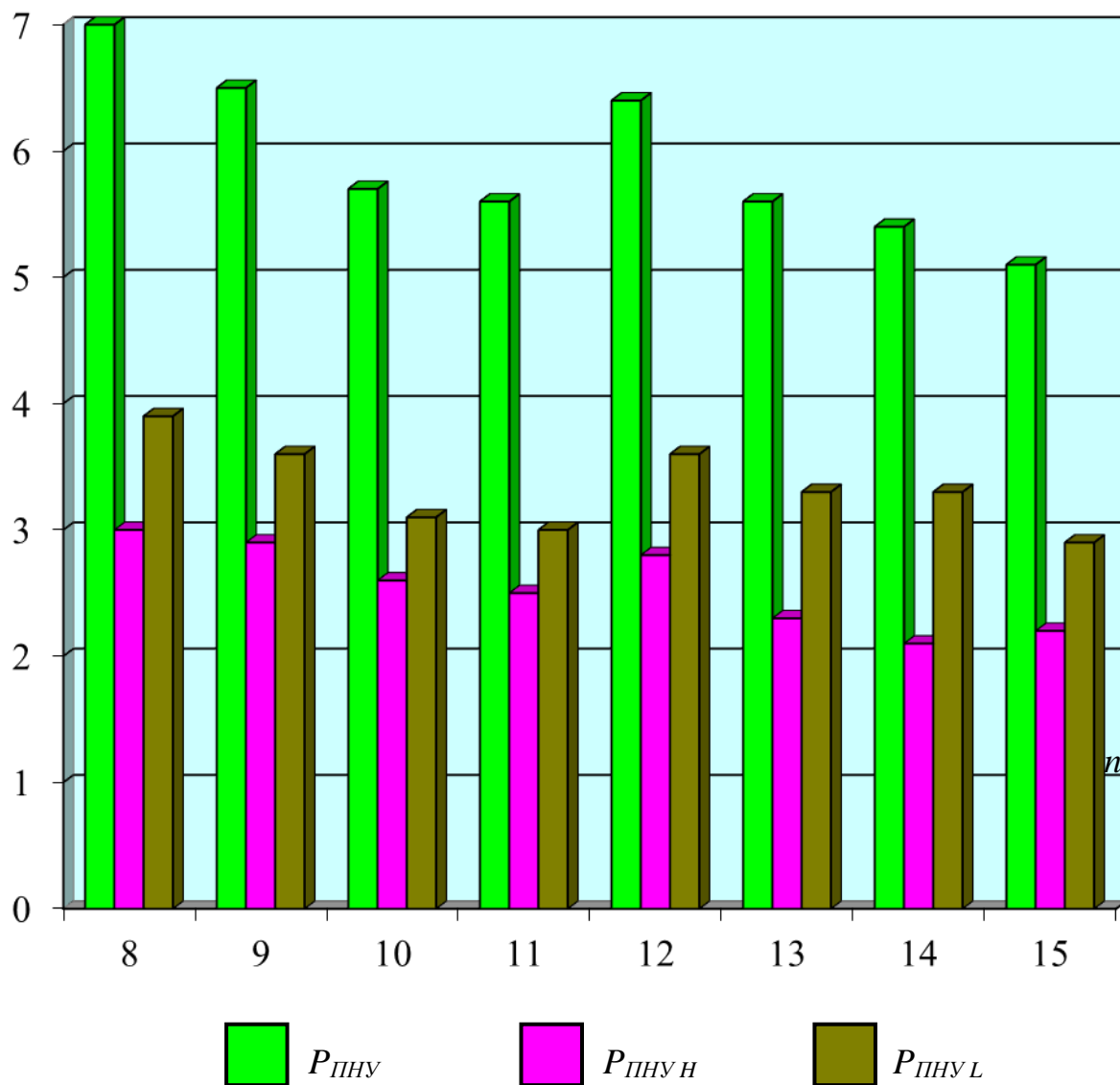


Рис. 5.10 – Діаграми ймовірностей P_{PNU} , P_{PNUH} і P_{PNUL} пропуску несуттєвої помилки в усіченому добутку

Побудовані діаграми показують, що ймовірності пропуску несуттєвої помилки в усіченому добутку зменшуються з ростом розрядності операндів. Позитивна помилка пропускається з меншою ймовірністю, чим негативна помилка.

На рис. 5.11 показані діаграми зміни ймовірностей P_{OP} , P_{OPH} і P_{OPL} виявлення помилки в повному добутку для ряду значень розрядності n матричних помножувачів і кількості вірних розрядів $n_T = n$.

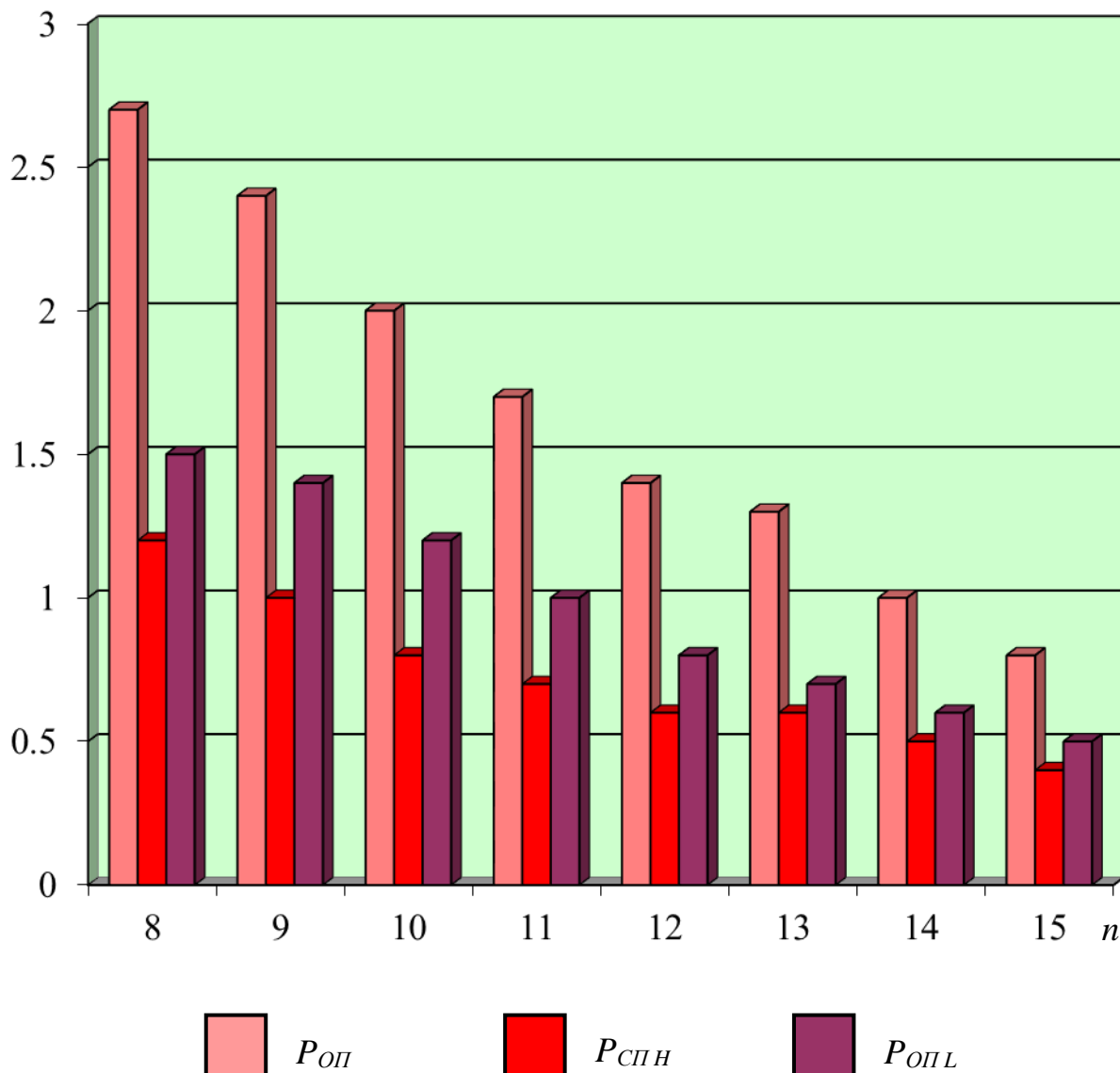


Рисунок 5.11 – Діаграми ймовірностей P_{OP} , $P_{СПН}$ і $P_{ОПЛ}$ виявлення помилки в повному добутку

Побудовані діаграми показують, що ймовірності виявлення помилки в повному добутку зменшуються із зростанням розрядності операндів. Позитивна помилка виявляється з меншою ймовірністю в порівнянні з негативною помилкою.

На рис. 5.12 показані діаграми зміни ймовірностей P_{OU} , P_{OUH} і P_{OUL} виявлення помилки в усіченому добутку для ряду значень розрядності n матричних помножувачів і кількості вірних розрядів $n_T = n$.

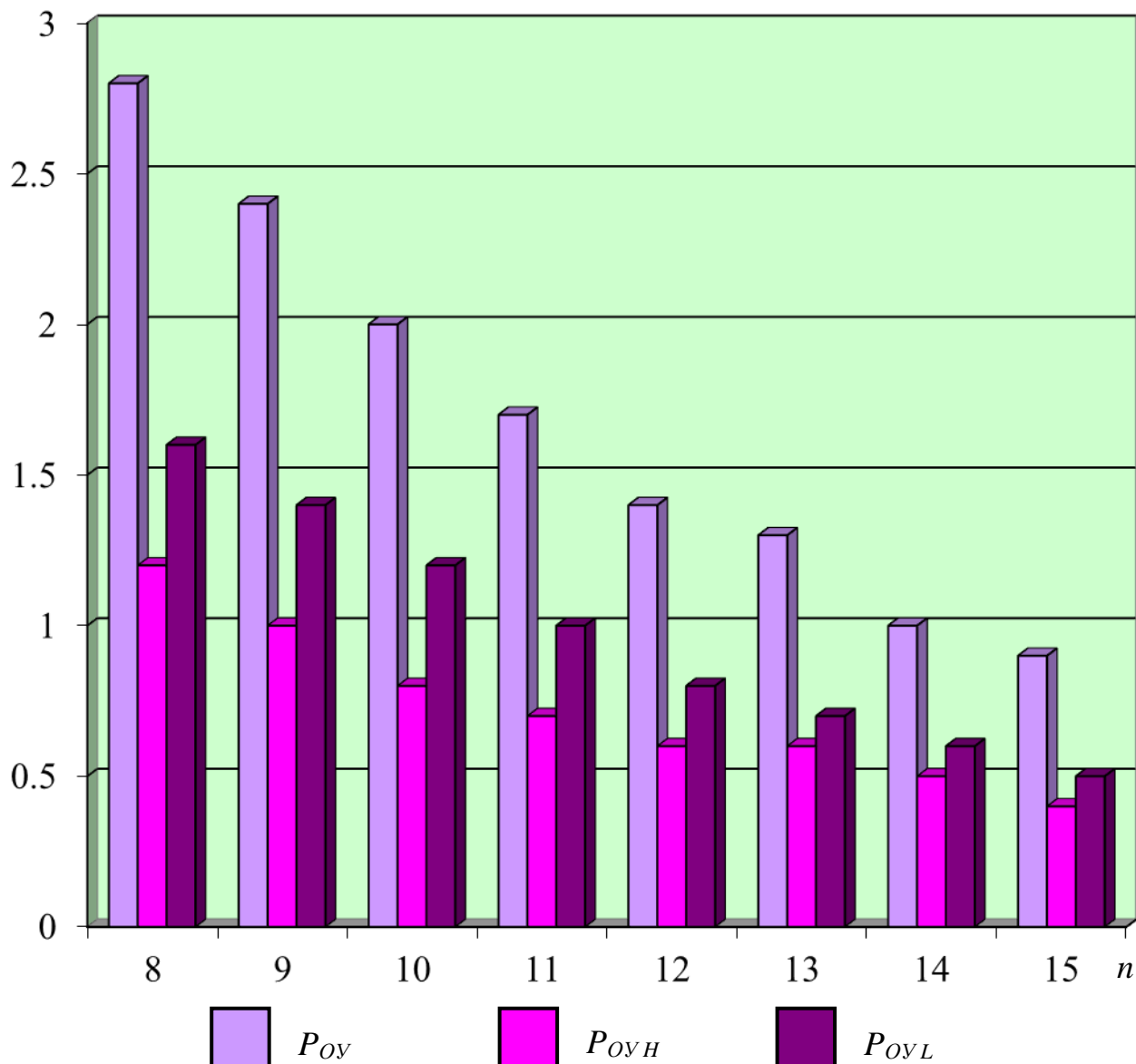


Рис. 5.12 – Діаграми ймовірностей P_{OY} , P_{OYH} і P_{OYL} виявлення помилки в усіченому добутку

Побудовані діаграми виявлення помилки в усіченому добутку практично повторюють вид діаграм для ймовірностей виявлення суттєвої помилки в повному добутку, які показані на рис. 5.13.

На рис. 5.13 показані діаграми зміни ймовірностей $P_{ПП}$, $P_{ППН}$ і $P_{ППЛ}$ пропуску помилки в повному добутку для ряду значень розрядності n матричних помножувачів і кількості вірних розрядів $n_T = n$.

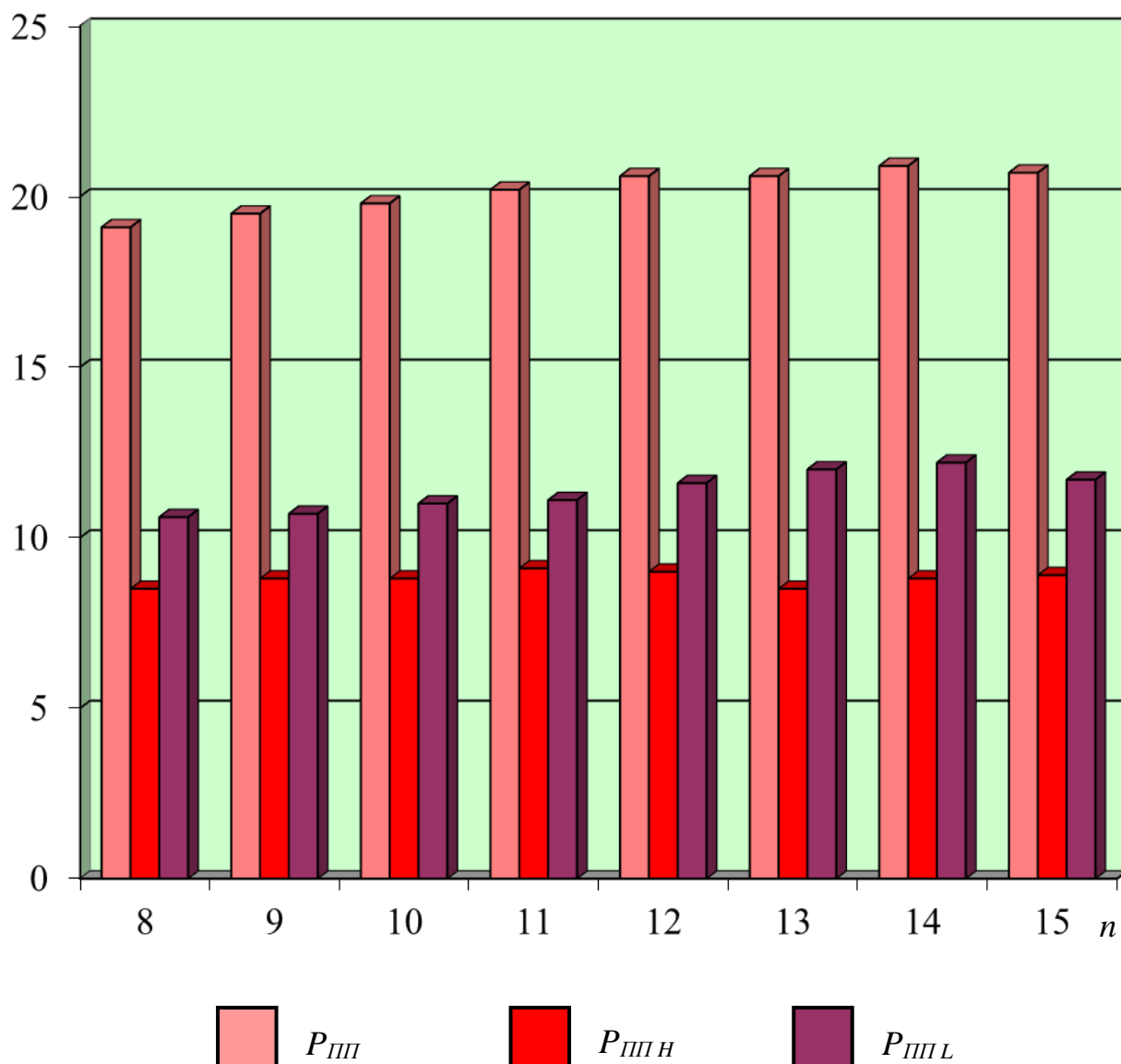


Рис. 5.13 – Діаграми ймовірностей P_{III} , P_{IIIH} і $P_{III L}$ пропуску помилки в повному добутку

Побудовані діаграми показують, що ймовірності пропуску помилки в повному добутку збільшуються із зростанням розрядності операндів. Позитивна помилка пропускається з небагато меншою ймовірністю в порівнянні з негативною помилкою.

На рис. 5.14 показані діаграми зміни ймовірностей P_{IIV} , P_{IIVH} і $P_{IIV L}$ пропуску помилки в усіченому добутку для ряду значень розрядності n матричних помножувачів і кількості вірних розрядів $n_T = n$.

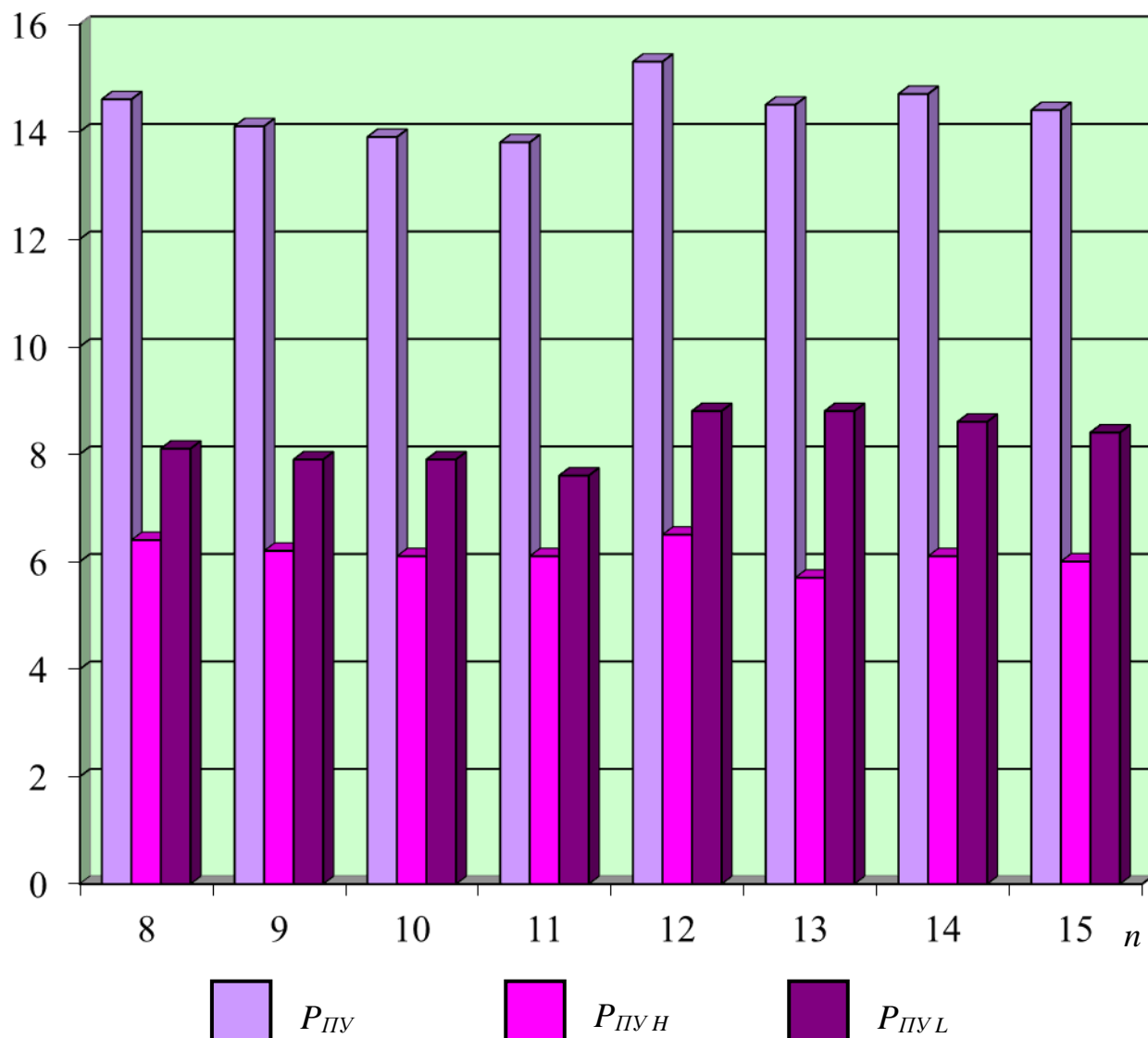


Рис. 5.14 – Діаграми ймовірностей $P_{ПВ}$, $P_{ПВН}$ і $P_{ПВЛ}$ пропуску помилки в усіченому добутку

Побудовані діаграми показують, що ймовірності пропуску помилки в усіченому добутку мають тенденцію збільшуватися з ростом розрядності операндів. Позитивна помилка пропускається з меншою ймовірністю в порівнянні з негативною помилкою.

На рис. 5.15 показані діаграми зміни показників $D_{П}$ і $D_{МП}$ недостовірності контролю результатів по нерівностям і по модулю в повному добутку для ряду значень розрядності n матричних помножувачів і кількості вірних розрядів $n_T = n - 4$.

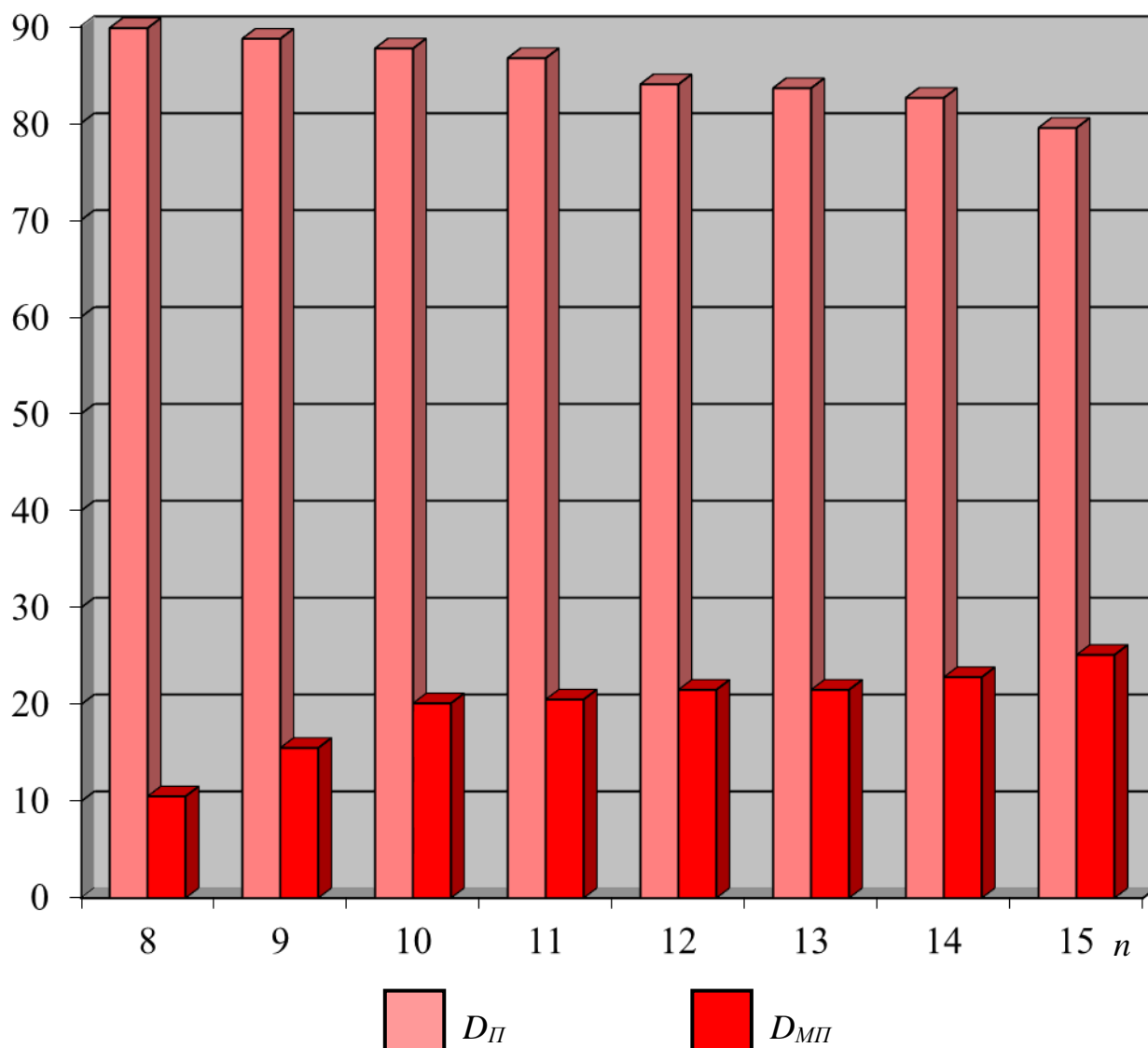


Рис. 5.15 – Діаграми показників D_{Π} і $D_{МП}$ достовірності контролю результатів у повному добутку

Побудовані діаграми показують, що для повного добутку граничний контроль має значно більшу достовірність контролю результатів у порівнянні з контролем по модулю.

На рис. 5.16 показані діаграми зміни достовірності D_{ν} та $D_{\mu\nu}$ граничного контролю результатів та контролю по модулю в усіченому добутку для ряду значень розрядності n однотактних матричних помножувачів і кількості вірних розрядів $n_T = n - 4$.

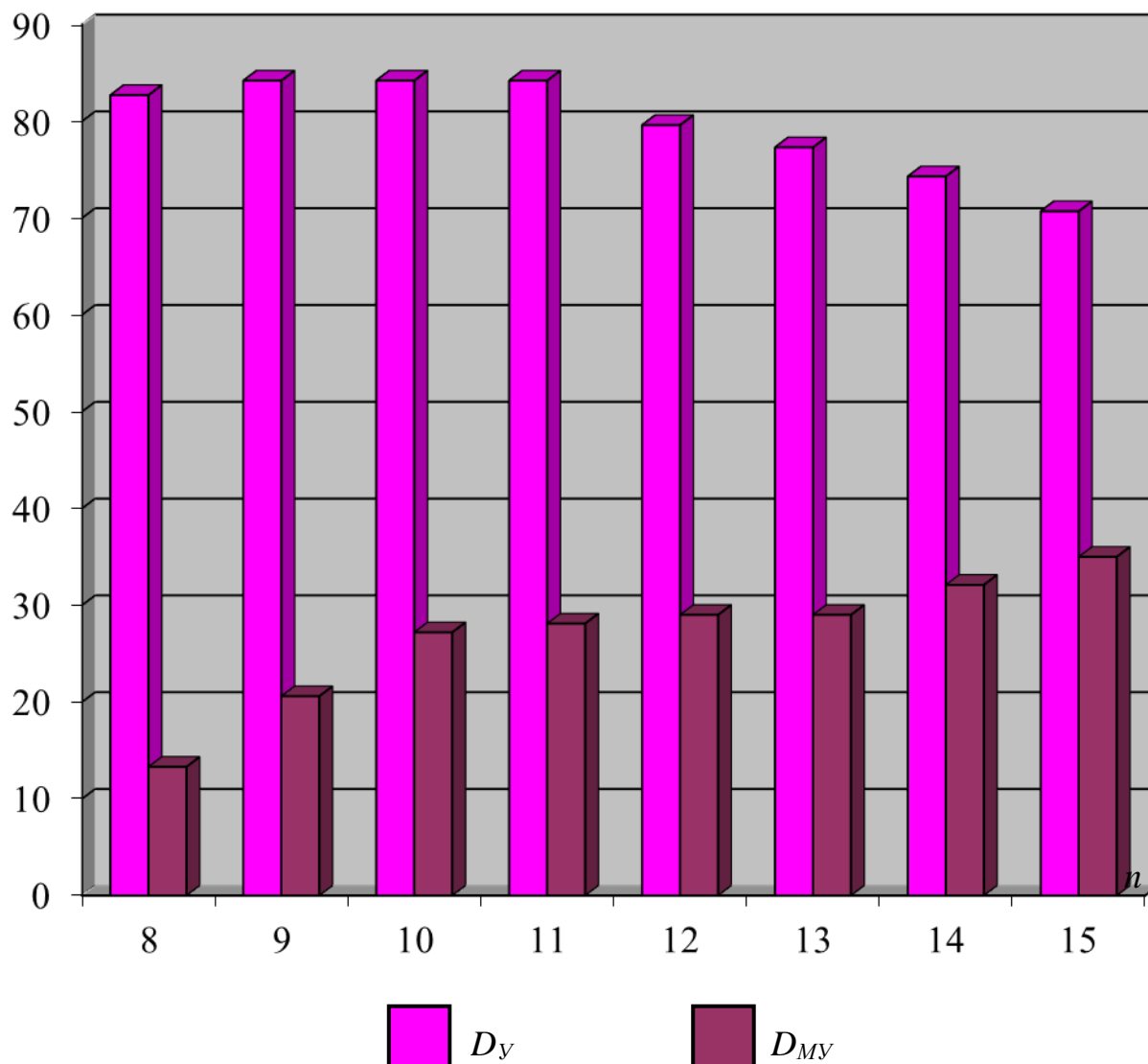


Рис. 5.16 – Діаграми показників D_V та D_{MV} достовірності контролю результатів в усіченому добутку

Побудовані діаграми показують, що для усіченого добутку граничний контроль має значно більшу достовірність контролю результатів у порівнянні з контролем по модулю.

На рис. 5.17 показані діаграми зміни показників δD_{II} і δD_V підвищення достовірності граничного контролю результатів у порівнянні з контролем по модулю для повної та скороченої операцій, ряду значень розрядності n і кількості вірних розрядів $n_T = n - 4$.

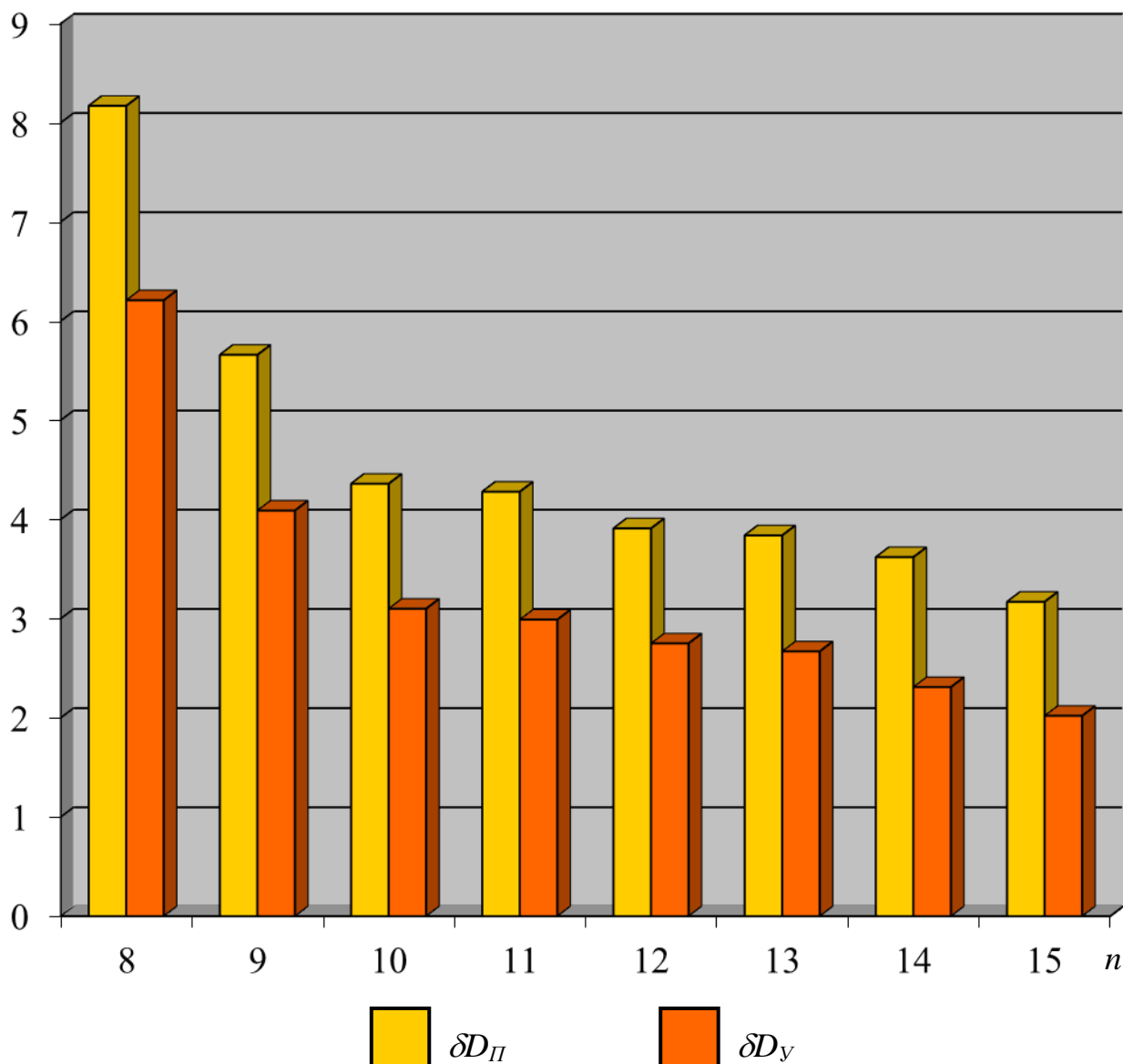


Рис. 5.17 – Діаграми показників підвищення δD_{π} і δD_{γ} достовірності контролю результатів

Побудовані діаграми показують, що показники підвищення достовірності граничного контролю наближених результатів зменшуються з ростом розрядності операндів.

Для повної операції показник достовірності граничного контролю наближених результатів знижується більшою мірою, чим при виконанні скороченої операції.

На рис. 5.18 показані діаграми зміни достовірності $D_{П1}$, $D_{П2}$, $D_{П3}$, $D_{У1}$, $D_{У2}$, $D_{У3}$ граничного контролю щодо повного та усіченого результатів множення мантис при кількості вірних розрядів $n_T = n$ для трьох значень кратності несправності: 1, 2 та 3.

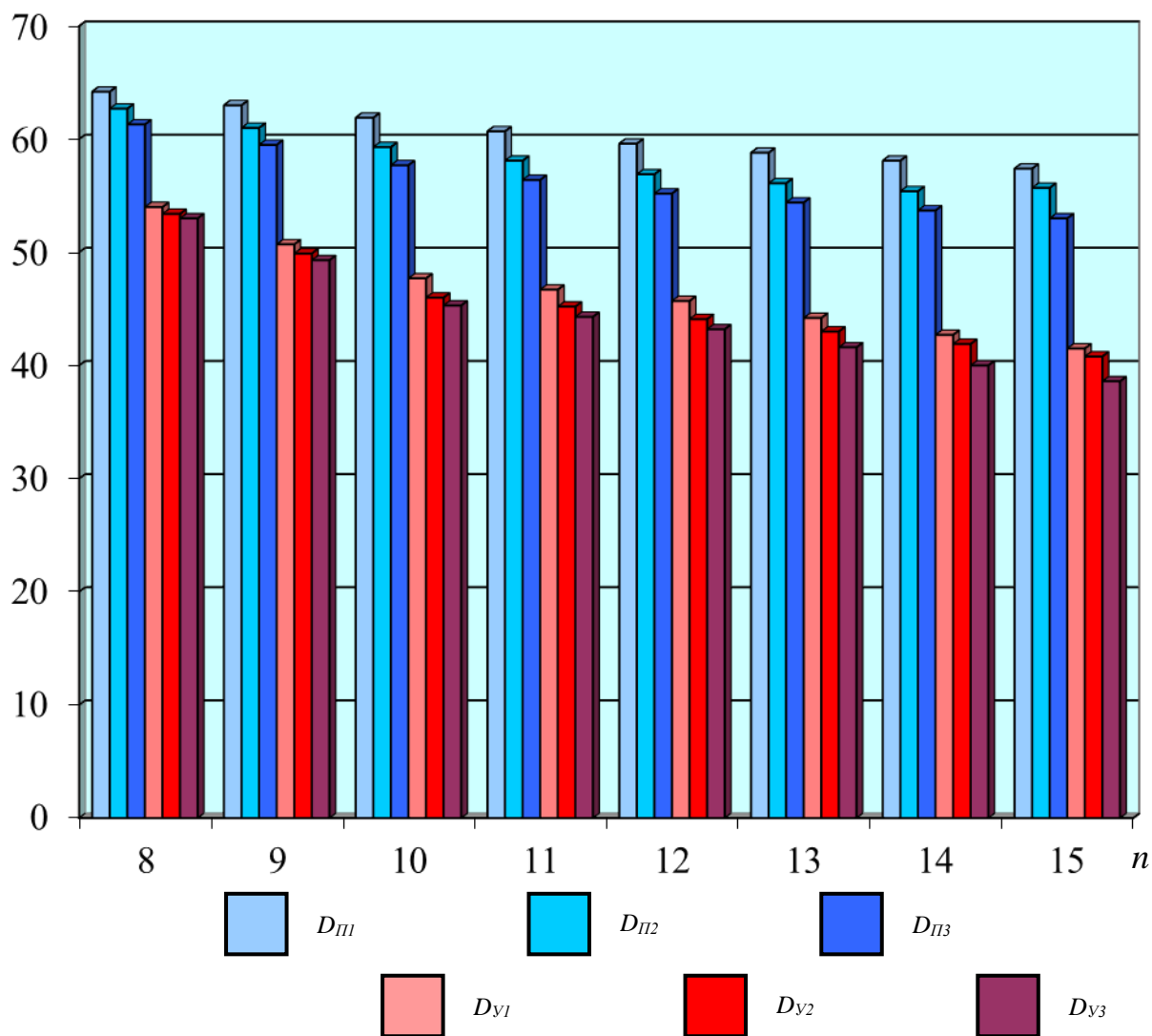


Рис. 5.18 – Діаграми достовірності $D_{П1}$, $D_{П2}$, $D_{П3}$, $D_{У1}$, $D_{У2}$, $D_{У3}$ граничного контролю при кратній несправності в повному добутку

Побудовані діаграми показують, що достовірність граничного контролю результату повної та скороченої операції зменшується з ростом розрядності операндів та кратності несправності. Достовірність граничного контролю усічених добутків суттєво менша за повні добутки.

На рис. 5.19 показані діаграми зміни достовірності $D_{П1}$, $D_{П2}$, $D_{П3}$, $D_{У1}$, $D_{У2}$, $D_{У3}$ граничного контролю щодо повного та усіченого результатів множення мантис при кількості вірних розрядів $n_T = n - 6$ для трьох значень кратності несправності: 1, 2 та 3.

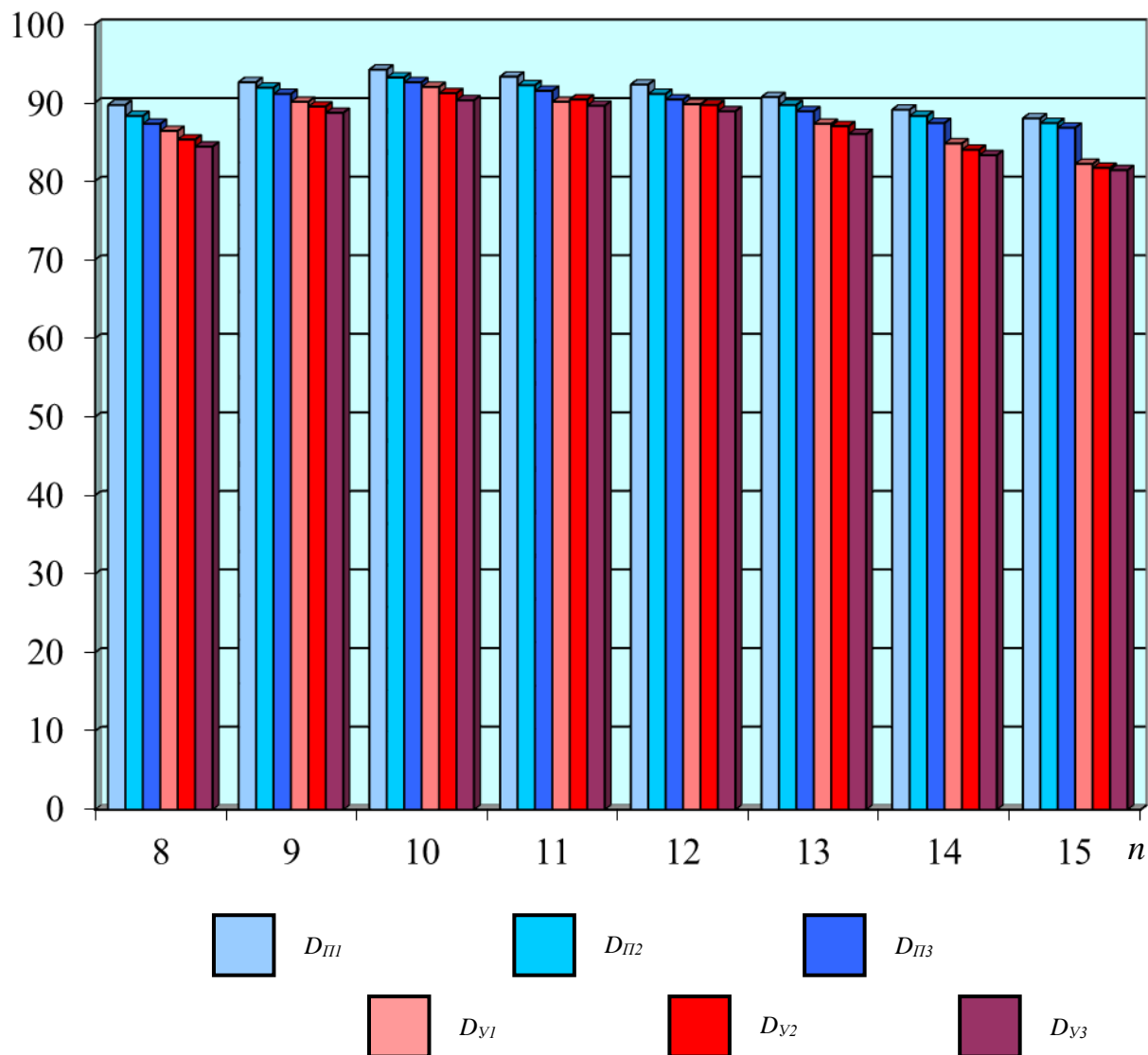


Рис. 5.19 – Діаграми достовірності $D_{П1}$, $D_{П2}$, $D_{П3}$, $D_{У1}$, $D_{У2}$, $D_{У3}$ граничного контролю при кратній несправності в повному добутку

Побудовані діаграми показують, що достовірність граничного контролю результату повної та скороченої операції зменшується з ростом розрядності операндів та кратності несправності. Достовірність граничного контролю усічених добутків суттєво менша за повні добутки.

5.3 ВИСНОВКИ

Проведені дослідження показали наступні закономірності появи, виявлення та пропуски суттєвих і несуттєвих, позитивних і негативних помилок при повному та зсіченому виконанні множення мантис у матричному пристрої:

- імовірності виявлення суттєвої помилки в повному і зсіченому добутку практично однакові і зменшуються з ростом розрядності операндів;

- позитивні суттєві помилки виявляються з меншою ймовірністю в порівнянні з негативними помилками;

- імовірності виявлення несуттєвої помилки в повному та зсіченому добутку практично збігаються й зменшуються з ростом розрядності операндів до нуля;

- позитивні несуттєві помилки виявляються з нульовою ймовірністю;

- імовірності пропуску суттєвої помилки в повному та зсіченому добутку практично однакові і збільшуються з ростом розрядності операндів;

- позитивні суттєві помилки пропускаються з небагато меншою ймовірністю в порівнянні з негативною помилкою.

- імовірності виявлення помилки в повному та зсіченому добутку практично збігаються й зменшуються із зростанням розрядності операндів;

- імовірності пропуску помилки в повному й усіченому добутку практично збігаються й збільшуються з ростом розрядності операндів

- для повного та усіченого добутку граничний контроль має значно більшу достовірність контролю результатів у порівнянні з контролем по модулю;

- показники підвищення достовірності граничного контролю результатів зменшуються з ростом розрядності операндів;

- для повної операції достовірність знижується більшою мірою, чим для скороченої операції.

- достовірність граничного контролю результату повної та скороченої операції зменшується з ростом розрядності операндів та кратності несправності.

ВИСНОВОК

В магістерській роботі була поставлена мета проведення дослідження граничного контролю для визначення достовірності результатів обробки мантис в матричному помножувачі під дією його кратних відмов. Таке дослідження має актуальну тему, оскільки присвячено питанням оцінки одної з основних характеристик сучасних обчислювальних пристроїв – достовірності контролю наближених результатів.

Сучасні системи моделювання не забезпечують завдання типових несправностей цифрових схем обчислювальних пристроїв та одержання оцінки ймовірності появи помилок у наближених результатах обчислень за повною та скороченою операцією. Тому для проведення дослідження було розроблено алгоритми та програмна модель граничного контролю однократного помножувача мантис для виконання повного та скороченого множення під дією кратних відмов пристрою.

В основу граничного контролю однократного помножувача мантис було покладено визначення простої за обчисленням (по мантисах співмножників) нижньої та верхньої границь результату.

При підвищенні величини результату за верхню границю або при зниженні за нижню границю метод визначає помилку за суттєву та відповідно позитивну та негативну. Помилки в межах границь результату визначаються несуттєвими, а сам результат достовірним. Насправді ознакою суттєвої та несуттєвої помилки є її приналежність відповідно вірним та невірним розрядам результату.

Дослідження було спрямоване на оцінку достовірності граничного контролю результатів, для чого необхідно було простежити збігання та різницю оцінки методу граничного контролю із справжнім становищем щодо достовірності результату.

Програмна модель граничного контролю однократного помножувача мантис пройшла досить повну перевірку на контрольних прикладах в різних режимах роботи пристрою.

Розроблена програмна модель граничного контролю забезпечила одержання оцінки щодо ймовірностей виявлення та пропуску суттєвих та несуттєвих позитивних і негативних помилок наближених результатів обчислень, а також достовірності контролю наближених результатів в умовах дії кратних відмов.

Проведені дослідження програмної моделі граничного контролю матричного помножувача мантис показали наступні закономірності:

- ймовірності виявлення суттєвої та несуттєвої помилки зменшуються із зростанням розрядності мантис;
- позитивні несуттєві помилки результатів обчислень добутоків у повній та скороченій операціях виявляються з нульовою ймовірністю;
- ймовірності пропуску суттєвої помилки наближених результатів збільшуються із зростанням розрядності мантис;
- позитивні суттєві помилки результатів обчислень виявляються та пропускаються з меншою ймовірністю у порівнянні з негативною помилкою.
- ймовірності виявлення та пропуску суттєвої і несуттєвої помилки наближених результатів практично збігаються між собою;
- для повного та усіченого добутку граничний контроль має значно більшу достовірність порівняно до найбільш поширеного для арифметичних операцій методу контролю по модулю;
- показники підвищення достовірності граничного контролю наближених результатів обробки мантис чисел з плаваючою точкою у матричних обчислювальних пристроях зменшуються в операціях повного та скороченого множення із зростанням розрядності мантис;
- для повної операції множення мантис, що виконується у матричному пристрої, достовірність контролю наближених результатів обчислень знижується більшою мірою, чим для скороченої операції множення.

Достовірність граничного контролю результату повної та скороченої операції зменшується з ростом розрядності операндів та кратності несправності.

ПЕРЕЛІК ПОСИЛАНЬ

1. Рабочее диагностирование безопасных информационно-управляющих систем / А.В. Дрозд, В.С. Харченко, С.Г. Антошук, 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.
2. Иыуду К. А. Надежность, контроль и диагностика вычислительных машин и систем. – М.: Высш. школа, 1989. – 216 с.
3. Каган Б. М. Электронные вычислительные машины и системы: Учеб. Пособие для вузов. – М.: Энергоатомиздат, 1991. – 592 с.
4. Селлерс Ф. Методы обнаружения ошибок в работе ЭЦВМ. – М.: Мир, 1972. – 310 с.
5. Савченко Ю. Г. Цифровые устройства, нечувствительные к неисправностям элементов. – М.: Советское радио, 1977. – 176 с.
6. Горяшко А. П. Синтез диагностируемых схем вычислительных устройств. – М.: Наука, 1987. – 288 с.
7. Щербаков Н. С. Достоверность работы цифровых устройств. – М.: Машиностроение, 1989. – 224 с.
8. Согомонян Е. С., Слабаков Е. В. Самопроверяемые устройства и отказоустойчивые системы. – М.: Радио и связь, 1989. – 208 с.
9. Журавлев Ю. П., Котелюк Л. А., Циклинский Н. И. Надежность и контроль ЭВМ. – М.: Советское радио, 1978. – 416 с.
10. Дрозд О. В. Контроль за модулем обчислювальних пристроїв. Навч. посібн. для студ. спеціальності 7.091501 – «Комп'ютерні та інтелектуальні системи та мережі» / Одеськ. нац. політехн. ун-т. – Одеса: АО Бахва, 2002. – 144 с.
11. Дрозд А. В. Особенности контроля приближенных вычислений // Тр. Одес. политехн. ун-та. – Одесса, 1999. – Вып. 1(7). – С. 209 – 211.
12. Моллов В. К. Структурно-функциональные методы оперативного контроля и диагностики цифровых устройств управляющих систем: Автореф. дис. . . канд. техн. наук: 05.13.13 / Киевск. политехн. ин-т – Киев, 1989. – 16 с.

13. Дрозд А. В. Контроль вычислительных устройств по неравенствам // Ученые записки Симферопольского гос. ун-та. – Винница-Симферополь, 1998. – Спецвып. – С. 237 – 240.

14. Дрозд А В. Использование неравенств для контроля вычислительных устройств / А.В. Дрозд, Е. В.Огинская, М.В. Лобачев // Тр. второй междунаро. научно-практической конференции «Современные информ. и электрон. технологии. – Одесса, 2001. – С. 80.

15. Дрозд А. В. Огинская Е. В. Метод контроля по неравенствам умножителя мантисс чисел / А.В. Дрозд, Е. В.Огинская // Технология и конструирование в электронной аппаратуре. – Одесса, 2004. – Вып. 1 (21). – С. 52 – 54.

16. Дрозд А. В., Огинская Е. В., Николенко И. Н. Контроль по неравенствам обработки приближенных данных / А.В. Дрозд, Е. В.Огинская, И.Н. Николенко // Холодильна техніка і технологія. – 2004. – № 2. – С. 100 – 101.

17. 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.

18. Дрозд О.В., Баланда А.В., Удуденко С.П., Мельничук Є.Ю. Особенности обработки и контроля мантисс в матричном делителе // Матеріали ІХ Міжнародної науково-практичної конференції «Інформаційні управляючі системи і технології» (ІУСТОДЕСА-2021). – Одеса, 2021. – С. 109 – 112.

ПРОГРАМНА МОДЕЛЬ ГРАНИЧНОГО КОНТРОЛЮ

```
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;
Edit9: TEdit;
Button12: TButton;
Panel12: TPanel;
Panel13: TPanel;
Panel14: TPanel;
Edit8: TEdit;
Edit10: TEdit;
Panel15: TPanel;
Panel16: TPanel;
Edit11: TEdit;
Edit12: TEdit;
Panel17: TPanel;
Edit13: TEdit;
Panel18: TPanel;
Panel19: TPanel;
Panel20: TPanel;
Panel21: TPanel;
Image2: TImage;
Panel22: TPanel;
Panel29: TPanel;
Panel30: TPanel;
Panel31: TPanel;
Panel24: TPanel;
Panel25: TPanel;
Panel26: TPanel;
Panel27: TPanel;
Panel28: TPanel;
Panel23: TPanel;
Panel32: TPanel;
Panel33: TPanel;
Panel34: TPanel;
Button15: TButton;
Panel35: TPanel;
Panel36: TPanel;
Panel37: TPanel;
Panel38: TPanel;
Panel39: TPanel;
procedure Button2Click(Sender: TObject);

```

procedure Button1Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure OnMouseDown(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 Button12Click(Sender: TObject);
procedure FormDragOver(Sender, Source: TObject; X, Y: Integer;
  State: TDragState; var Accept: Boolean);
procedure FormDragDrop(Sender, Source: TObject; X, Y: Integer);
procedure Panel22MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Panel23Click(Sender: TObject);
procedure Panel21Click(Sender: TObject);
procedure Panel12Click(Sender: TObject);
procedure Button15Click(Sender: TObject);
procedure Panel35Click(Sender: TObject);
procedure Panel36Click(Sender: TObject);
procedure Panel37Click(Sender: TObject);
procedure Panel38Click(Sender: TObject);

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

const
  nh=15; nf=11;
var
  Form1: TForm1;
  Ma:array[1..nh,1..nh-1]of integer;

```



```

Mb:array[1..nh,1..nh-1]of integer;
Mc:array[1..nh,1..nh-1]of integer;
Ma1:array[1..nh,1..nh-1]of integer;
Mb1:array[1..nh,1..nh-1]of integer;
Ma2:array[1..nh-1]of integer;
Mb2:array[1..nh-1]of integer;
Ms:array[1..nh,1..nh-1]of integer;
Mp:array[1..nh,1..nh-1]of integer;
AAb:array[1..nh]of integer;
BBb:array[1..nh]of integer;
PPl:array[1..nh]of integer;
PPh:array[1..nh]of integer;
PPb:array[1..2*nh]of integer;
PPe:array[1..2*nh]of integer;
TTb:array[1..2*nh]of integer;
TTe:array[1..2*nh]of integer;
MF:array[1..nf]of string;

```

```

n,aa,bb,pp,pe,tt,te,h4,h5:longint; //integer;
za,zb,f1,f2,ff,fr,df1,df2,nfe,nfr:integer;
mo,ka,kb,kp,e1,e2,ca,cb,k,pu,da:integer;
sh,sl,pl,ph,ep,et,cc,ce,ee,es,dp,dt:integer;
dehp,delp,dnhp,dnlp,nehp,nelp,nnhp,nnlp:integer;
deht,delt,dnht,dnlt,neht,nelt,nnht,nnlt:integer;

```

implementation

```
{ $R *.dfm }
```

```

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

```

```

procedure TForm1.Button1Click(Sender: TObject);
  var i,j,h1:integer;
begin
  repeat
    Button2.Top:=0; Button2.Left:=0;
    Application.ProcessMessages;
    n:=StrToInt(Edit9.Text);
    if (n<2)or(n>15) then
      Panel34.Caption:=
        'Íâïðààèèüíáÿ ðàçðÿáíñðü '+IntToStr(n);
    until (n>1)and(n<16);
    Panel34.Caption:=
      'Розрядність '+IntToStr(n);
    Panel34.Left:=12; Panel34.Top:=60;

```

```

Button1.Visible:=False;
Edit9.Visible:=False;
h4:=1; h5:=2; da:=0;
Button1.Visible:=false;
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;
k:=n; pu:=0; nfr:=0; es:=0;
while (1 shl k)*(k-1)+1>(1 shl n) do
  k:=k-1;
Panel12.Visible:=True;
Image1.Left:=8;      Image1.Top:=56;
Image1.Width:=32*(2*n+1); Image1.Height:=32*(n+5);
Image2.Left:=8;      Image2.Top:=56;
Image2.Width:=32*(2*n+1); Image2.Height:=32*(n+6);
if 32*(n+6)<550 then Image2.Height:=550;
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);
Image1.Canvas.Rectangle(4,32*n+100,32*(2*n-k)-4,32*n+124);
Image1.Canvas.Rectangle(4,32*n+132,32*(2*n-k)-4,32*n+156);
Button3.Top:=60;    Button3.Left:=32*n-20;
Button4.Top:=60;    Button4.Left:=64*n+12;
Panel1.Top:=32*n+92; Panel1.Left:=64*n+12;
Panel2.Top:=32*n+124; Panel2.Left:=64*n+12;
Panel19.Top:=32*n+156; Panel19.Left:=32*(2*n-k)+12;
Panel20.Top:=32*n+188; Panel20.Left:=32*(2*n-k)+12;
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;
Panel36.Visible:=True; Button15.Visible:=False;
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;
Panel15.Visible:=True; Panel16.Visible:=True;
Panel17.Visible:=True; Panel35.Visible:=True;
Panel19.Visible:=True; Panel20.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;
Edit10.Visible:=True; Edit11.Visible:=True;
Edit12.Visible:=True; Edit13.Visible:=True;
Panel22.Visible:=True;
Panel22.Top:=32*(n+6)+16; Panel22.Left:=32*n+4;

```

```

A1.Visible:=True;
B1.Visible:=True;
N8.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
          if i+j>k then Brush.Color:=clMoneyGreen
          else Brush.Color:=clSilver;
          Rectangle(h1,32*i+8,h1+16,32*i+24);
          TextOut(h1+1,32*i+9,IntToStr(i)+IntToStr(j+1));
        end;
      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:=clRed; Brush.Color:=clLime;
          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,32*n+72,h1+16-32*n,32*n+88);
          TextOut(h1+4-32*n,32*n+73,'0');
          Rectangle(h1-32*n,32*n+104,h1+16-32*n,32*n+120);
          TextOut(h1+4-32*n,32*n+105,'0');
          Rectangle(h1-32*n,32*n+136,h1+16-32*n,32*n+152);
          TextOut(h1+4-32*n,32*n+137,'0');
          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+72,h1+16,32*n+88);
          TextOut(h1+4,32*n+73,'0');
          if i>k then

```

```

begin
  Rectangle(h1,32*n+104,h1+16,32*n+120);
  TextOut(h1+4,32*n+105,'0');
  Rectangle(h1,32*n+136,h1+16,32*n+152);
  TextOut(h1+4,32*n+137,'0');
end;
end;
end;

end;

procedure TForm1.Button5Click(Sender: TObject);
var h1,ih:integer;
begin
  aa:=StrToInt(Edit1.Text);
  if (aa>=(1 shl n))or(aa<(1 shl (n-1))) then
  begin
    Panel21.Visible:=True;
    Panel21.Caption:='Число А в діапазоні '+
    IntToStr(1 shl (n-1))+ ' - '+IntToStr((1 shl n)-1);
  end
  else
  begin
    Panel21.Visible:=False;
    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
  bb:=StrToInt(Edit2.Text);
  if (bb>=(1 shl n))or(bb<(1 shl (n-1))) then
  begin
    Panel21.Visible:=True;
    Panel21.Caption:='Число В в діапазоні '+
    IntToStr(1 shl (n-1))+ ' - '+IntToStr((1 shl n)-1);
  end
  else
  begin
    Panel21.Visible:=False;

```

```

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,h2: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 // Å
    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 // Å
    BBb[jj]:=1-BBb[jj];
    Image1.Canvas.TextOut(64*n+12,32*jj+9,
      IntToStr(BBb[jj]));
  end;
  h1:=2*n-ii-jj;
  if Panel21.Caption=' ' then h2:=1
  else h2:=0;
  if (X-xx<17)and(Y-yy<17)and
    (jj>0)and(jj<=n)and(h1>0)and(h1<n)and
    (za<2)and(zb<2)and(jj+h1>k*h2)then
  begin
    Panel18.Left:=xx+8; Panel18.Top:=yy+56;
    Panel18.Visible:=True; Panel23.Visible:=True;
    Panel24.Visible:=True; Panel25.Visible:=True;
    Panel26.Visible:=True; Panel27.Visible:=True;
    Panel28.Visible:=True; Panel29.Visible:=True;
    Panel30.Visible:=True; Panel31.Visible:=True;
    Panel23.Caption:=IntToStr(jj)+IntToStr(h1+1);
  end;
end;

```

```

h4:=jj; h5:=h1+1;
Panel24.Caption:='P'+IntToStr(Mp[jj,h1]);
Panel25.Caption:='a'+IntToStr(Ma[jj,h1]);
Panel27.Caption:='s'+IntToStr(Ms[jj,h1]);
Panel30.Caption:='a1'+IntToStr(Ma1[jj,h1]);
Panel31.Caption:='b1'+IntToStr(Mb1[jj,h1]);
if jj>1 then
  Panel28.Caption:='b'+IntToStr(Mb[jj,h1])
else
begin
  Panel32.Caption:='a2'+IntToStr(Ma2[h1]);
  Panel33.Caption:='b2'+IntToStr(Mb2[h1]);
  Panel28.Caption:='b';
end;

if jj=1 then
begin
  Panel28.Width:=17; //Panel28.Top:=yy+64;
  Panel28.Height:=33; Panel28.Color:=clGradientActiveCaption;
  Panel32.Visible:=True; Panel33.Visible:=True;
end
else
begin
  Panel28.Height:=17;
  Panel28.Width:=33;
  Panel28.Color:=clAqua;
  Panel32.Visible:=False; Panel33.Visible:=False;
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);
if (aa>=(1 shl n))or(aa<(1 shl (n-1))) then
begin
  Panel21.Visible:=True;
  Panel21.Left:=264;
  Panel21.Top:=86;
  Panel21.Caption:='Мантиса А повинна бути нормалізованою';
end
else
begin
  Panel21.Visible:=False;
  Edit1.Text:=IntToStr(aa);

```

```
end;
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);
  if (bb>=(1 shl n))or(bb<(1 shl (n-1))) then
    begin
      Panel21.Visible:=True;
      Panel21.Left:=264;
      Panel21.Top:=86;
      Panel21.Caption:='Мантиса В повинна бути нормалізованою';
    end
  else
    begin
      Panel21.Visible:=False;
      Edit2.Text:=IntToStr(bb);
    end;
  end;
end;
```

```
procedure TForm1.Button7Click(Sender: TObject);
//Старт
  var h1,h2,h3,ih,i,j:integer;
begin
  if Button7.Caption='Старт' then
    begin
      if ((za>1)or(zb>1))then
        begin
          Button2.Visible:=False;
          Button7.Caption:='Фініш';
          Panel37.Visible:=True;
          Panel38.Visible:=True;
        end;
      Randomize;
      h2:=1 shl(n-1);
      h3:=0; cc:=0; ce:=0; ee:=0; ep:=0; et:=0;
      dehp:=0; dnhp:=0; deht:=0; dnht:=0;
      delp:=0; dnlp:=0; delt:=0; dnlt:=0;
      nehp:=0; nnhp:=0; neht:=0; nnht:=0;
      nelp:=0; nnlp:=0; nelt:=0; nnlt:=0; sh:=0; sl:=0;
      Edit3.Text:='0'; Edit4.Text:='0'; Edit6.Text:=' '; Edit7.Text:=' ';
      Edit8.Text:=' '; Edit10.Text:=' '; Edit11.Text:=' ';
      Edit12.Text:=' '; Edit13.Text:=' ';
      Panel21.Tag:=0;
      ca:=1-(za shr 1);
```

```

if Button7.Caption='Ôèìèø' then h3:=1;
Image1.Canvas.TextOut(64*(n-1)-48,32*(n-3)+16,'Помилки);
Image1.Canvas.TextOut(64*(n-2)-24,32*(n-2),'суттєві. ');
Image1.Canvas.TextOut(64*(n-1)-24,32*(n-2),'інсуттєві. ');
Image1.Canvas.TextOut(64*(n-3),32*(n-1),'Виявлено');
Image1.Canvas.TextOut(64*(n-3),32*n,'Пропущено. ');
if za=2 then aa:=h2;
repeat
cb:=1-(zb shr 1);
if za mod 2=1 then aa:=h2+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;
if zb=2 then bb:=h2;
repeat
if zb mod 2=1 then bb:=h2+Random(h2);
Edit2.Text:=IntToStr(bb);
pl:=aa+bb;
pl:=pl+(pl shl 1);
pl:=pl shr 2;
pl:=pl+7 shl (n-4);
pl:=pl and (1 shl n-1);
h1:=1;
for ih:=1 to n do
begin
BBb[ih]:=(bb and h1)shr(ih-1);
Image1.Canvas.TextOut(64*n+12,32*ih+9, IntToStr(BBb[ih]));
PPl[ih]:=(pl and h1)shr(ih-1);
h1:=h1 shl 1;
end;

da:=AAb[n-1] xor BBb[n-1];

pp:=aa*bb; tt:=pp
for i:=1 to k do
  for j:=1 to k-i+1 do
    tt:=tt-(AAb[i]*BBb[j])shl(i+j-2);
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;
h1:=1 shl k;
for ih:=k+1 to 2*n do
begin
TTb[ih]:= (tt and h1) shr (ih-1);
h1:=h1 shl 1;
Image1.Canvas.TextOut(64*n-32*ih+12,32*n+105, IntToStr(TTb[ih]));
end;
if nfe=0 then
begin
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
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;
repeat
for i:=1 to n do
for j:=1 to n-1 do
if i+j>k*pu then
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 Ma[h4,h5]:=0;
  if fr=117 then Ma[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 i+j=(k+1)*pu then 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 Mp[h4,h5]:=0;
if fr=115 then Mp[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;
if pu=0 then
begin
  cc:=cc+1;
  PPe[1]:=AAb[1]*BBb[1];
  for ih:=1 to n-1 do PPe[ih+1]:=Ms[ih,1];
  for ih:=1 to n-1 do PPe[n+ih]:=Ms[n,ih];
  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;
  Edit3.Text:=IntToStr(pp);
  Edit4.Text:=IntToStr(pe);
  Edit5.Text:=IntToStr(pp-pe);
  if pp<>pe then
  begin
    ce:=ce+1;
    Edit6.Text:=IntToStr(ce);
    Edit7.Text:=IntToStr((ce*100)div cc);
    //if ((pe shr n)>ph)or((pe shr n)<pl) then e1:=1 else e1:=0;
    if (((pe shr n)>pl)and(da=1))or(((pe shr n)<pl)and(da=0))
    then e1:=1 else e1:=0;
    if (abs(pp-pe)>=(1 shl(n+es)))then
    begin

```

```

ep:=ep+1
if pe>pp then
begin
dehp:=dehp+e1; nehp:=nehp+1-e1;
end
else
begin
delp:=delp+e1; nelp:=nelp+1-e1;
end;
end
else
if pe>pp then
begin
dnhp:=dnhp+e1; nnhp:=nnhp+1-e1;
end
else
begin
dnlp:=dnlp+e1; nnlp:=nnlp+1-e1;
end;
end;
end
else
begin
for ih:=k to n-1 do TTe[ih+1]:=Ms[ih,1];
for ih:=1 to n-1 do TTe[n+ih]:=Ms[n,ih];
TTe[2*n]:=Mp[n,n-1];
te:=0;
for ih:=k+1 to 2*n do
begin
te:=te+TTe[ih]shl(ih-1);
Image1.Canvas.TextOut(64*n-32*ih+12,32*n+137, IntToStr(TTe[ih]));
end;
Edit8.Text:=IntToStr(tt);
Edit10.Text:=IntToStr(te);
Edit11.Text:=IntToStr(tt-te);
if tt<>te then
begin
if (((te shr n)>pl)and(da=1))or(((te shr n)<pl)and(da=0))
then e2:=1 else e2:=0;
ee:=ee+1;
Edit12.Text:=IntToStr(ee);
Edit13.Text:=IntToStr((ee*100)div cc);
if (abs(tt-te)>=(1 shl(n+es)))then
begin
et:=et+1;
if te>tt then
begin
deht:=deht+e2; neht:=neht+1-e2;

```

```

end
else
begin
  delt:=delt+e2; nelt:=nelt+1-e2;
end;
end
else
if te>tt then
begin
  dnht:=dnht+e2; nnht:=nnht+1-e2;
end
else
begin
  dnlt:=dnlt+e2; nmlt:=nmlt+1-e2;
end;
end;
end;
if (za<2)and(zb<2) then
begin
  Panel21.Visible:=True;
  if pu=0 then
  begin
    Panel21.Caption:='Повна операція';
    Button2.Visible:=False;
    repeat
      Application.ProcessMessages;
    until Panel21.Tag=1;
  end
  else
    Panel21.Caption:='Скорочена операція';
    Button2.Visible:=True;
  end;
  pu:=1-pu;
  Panel18.Visible:=False;
  Panel23.Visible:=False;
  until pu=0;
  if (za=3)and(zb>1) then cb:=1;
  with Image1.Canvas do
  begin
    Font.Color:=clBlack;
    h1:=(ep*2000 div cc+1)div 2;
    TextOut(64*(n-2),32*(n-1),
      IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
    h1:=((ce-ep)*2000 div cc+1)div 2;
    TextOut(64*(n-1),32*(n-1),
      IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
    h1:=(et*2000 div cc+1)div 2;
    TextOut(64*(n-2),32*n,

```

```

    IntToStr(h1 div 10)+' '+IntToStr(h1 mod 10)+'%');
h1:=((ee-et)*2000 div cc+1)div 2;
TextOut(64*(n-1),32*n,
    IntToStr(h1 div 10)+' '+IntToStr(h1 mod 10)+'%');
end;
if e1=1 then
begin
if (sh=1)and(pe>pp) then
begin
    Button7.Visible:=False;
    Panel38.Color:=clRed;
    Panel38.Font.Color:=clWhite;
    Panel38.Caption:='H';
    Panel37.Color:=clCream;
    Panel37.Caption:=' ';
    repeat
        Application.ProcessMessages;
    until Panel38.Color<>clRed;
    end;
if (sl=1)and(pe<pp) then
begin
    Button7.Visible:=False;
    Panel37.Color:=clRed;
    Panel37.Font.Color:=clWhite;
    Panel37.Caption:='L';
    Panel38.Color:=clCream;
    Panel38.Caption:=' ';
    repeat
        Application.ProcessMessages;
    until Panel37.Color<>clRed;
    end;
end;
Application.ProcessMessages;
if zb=2 then
begin
    bb:=bb+1;
    if bb=2*h2 then cb:=1;
end;
until cb=1;
if za=2 then
begin
    aa:=aa+1;
    if aa=2*h2 then ca:=1;
end;
until ca=1;
end;
Button2.Visible:=True;
Button7.Caption:='Фініш';

```



```

Panel37.Visible:=False;
Panel38.Visible:=False;
ca:=1; cb:=1;
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.Button12Click(Sender: TObject);
begin
  nfe:=1;
  Panel35.Caption:=IntToStr(h4)+IntToStr(h5);
  Panel35.Color:=clLime;
end;
```

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

```
procedure TForm1.FormDragDrop(Sender, Source: TObject; X, Y: Integer);
begin
  if mo=0 then
    begin
      X:=X div 32;
      if X>n then X:=n;
      (Source as TPanel).Top:=32*(n+6)+16;
      (Source as TPanel).Left:=32*X+4;
    end
  else
    begin
      X:=X div 16-2*n-1;
      Panel12.Caption:=IntToStr(X);
      if X>n then X:=n;
      if X<0 then X:=0;
      Panel22.Top:=56;
      Panel22.Left:=32*n+16*X+16;
    end;
  es:=n-X;
  Panel12.Caption:=IntToStr(es);
```

```
//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;
```

```

    //Panel12.Caption:=IntToStr(es);
end;

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

procedure TForm1.Panel23Click(Sender: TObject);
begin
    Panel18.Visible:=false;
    Panel23.Visible:=false;
end;

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

procedure TForm1.Panel12Click(Sender: TObject);
var i,j,h1,xm,ym:integer;
begin
    mo:=1-mo;
    if mo=0 then
        begin
            Image1.Visible:=True;
            Image2.Visible:=False;
            Panel12.Caption:='ÓÍÍ';
            Panel12.Color:=clMoneyGreen;
            Button3.Top:=60;    Button3.Left:=32*n-20;
            Button3.Width:=25;    Button3.Height:=25;
            Button4.Top:=60;    Button4.Left:=64*n+12;
            Button4.Width:=25;    Button4.Height:=25;
            Panel1.Top:=32*n+92;    Panel1.Left:=64*n+12;
            Panel1.Width:=25;    Panel1.Height:=25;
            Panel2.Top:=32*n+124;    Panel2.Left:=64*n+12;
            Panel2.Width:=25;    Panel2.Height:=25;
            Panel19.Top:=32*n+156;    Panel19.Left:=32*(2*n-k)+12;
            Panel20.Top:=32*n+188;    Panel20.Left:=32*(2*n-k)+12;
            Panel22.Top:=32*(n+6)+16;    Panel22.Left:=32*(n-es)+4;
        end
    else
        begin
            Panel18.Visible:=False;
            Panel23.Visible:=False;

```

```

Panel22.Top:=56; Panel22.Left:=16*(3*n-es)+16;
Image1.Visible:=False;
Image2.Visible:=True;
if Image2.Width<540 then Image2.Width:=540;
Panel12.Caption:='БП';
Panel12.Color:=clAqua;

```

```

with Image2.Canvas do
begin
  Pen.Color:=clBlue;
  Brush.Color:=clMoneyGreen;
  Font.Color:=clBlack;
  Rectangle(32*n+10,6,48*n+14,26);
  Rectangle(32*n+10,30,48*n+14,50);
  Rectangle(32*n+10,78,64*n+14,98);
  Rectangle(32*n+10,102,64*n+14,122);
  Rectangle(32*n+10,126,16*(4*n-k)+14,146);
  Rectangle(32*n+10,150,16*(4*n-k)+14,170);
  Pen.Color:=clRed;
  Rectangle(32*n+10,54,48*n+14,74);
  Rectangle(32*n+10,174,48*n+14,194);
  Button3.Top:=62;   Button3.Left:=32*n-6;
  Button3.Width:=21; Button3.Height:=21;
  Button4.Top:=86;   Button4.Left:=32*n-6;
  Button4.Width:=21; Button4.Height:=21;
  Panel1.Top:=134;   Panel1.Left:=32*n-6;
  Panel1.Width:=21;  Panel1.Height:=21;
  Panel2.Top:=158;   Panel2.Left:=32*n-6;
  Panel2.Width:=21;  Panel2.Height:=21;
  Panel19.Top:=182;  Panel19.Left:=32*n-6;
  Panel19.Width:=21; Panel19.Height:=21;
  Panel20.Top:=204;  Panel20.Left:=32*n-6;
  Panel20.Width:=21; Panel20.Height:=21;
  for i:=1 to n do
  begin
    h1:=48*n-16*i+12;
    Brush.Color:=clAqua;
    if da=1 then
    begin
      Rectangle(h1,56,h1+16,72);
      TextOut(h1+4,57,IntToStr(PPI[i]));
    end
    else
    begin
      Rectangle(h1,176,h1+16,192);
      TextOut(h1+4,177,IntToStr(PPI[i]));
    end;
    Pen.Color:=clBlue;
  end;
end;

```

```

Brush.Color:=clMoneyGreen;
Rectangle(h1,8,h1+16,24);
TextOut(h1+4,9,IntToStr(AAb[i]));
Rectangle(h1,32,h1+16,48);
TextOut(h1+4,33,IntToStr(BBb[i]));
Rectangle(h1+16*n,80,h1+16+16*n,96);
TextOut(h1+4+16*n,81,IntToStr(PPb[i]));
Rectangle(h1,80,h1+16,96);
TextOut(h1+4,81,IntToStr(PPb[i+n]));
Rectangle(h1+16*n,104,h1+16+16*n,120);
TextOut(h1+4+16*n,105,IntToStr(PPe[i]));
Rectangle(h1,104,h1+16,120);
TextOut(h1+4,105,IntToStr(PPe[i+n]));
if i>k then
begin
  Rectangle(h1+16*n,128,h1+16+16*n,144);
  TextOut(h1+4+16*n,129,IntToStr(TTb[i]));
  Rectangle(h1+16*n,152,h1+16+16*n,168);
  TextOut(h1+4+16*n,153,IntToStr(TTe[i]));
end;
Rectangle(h1,128,h1+16,144);
TextOut(h1+4,129,IntToStr(TTb[i+n]));
Rectangle(h1,152,h1+16,168);
TextOut(h1+4,153,IntToStr(TTe[i+n]));

end;
xm:=250;ym:=216;
Pen.Color:=clBlue;
Brush.Color:=clMoneyGreen;
Rectangle(xm+50,ym+100,xm+90,ym+190);
MoveTo(xm-10,ym+120);
LineTo(xm+50,ym+120);
MoveTo(xm-10,ym+170);
LineTo(xm+50,ym+170);
TextOut(xm+58,ym+132,'ÄÏ');
TextOut(xm+74,ym+136,'Â');
Rectangle(xm+50,ym,xm+90,ym+80);
TextOut(xm+58,ym+12,'ÁÓ');
MoveTo(xm-10,ym+20);
LineTo(xm+50,ym+20);
MoveTo(xm-10,ym+60);
LineTo(xm+50,ym+60);
MoveTo(xm+90,ym+60);
LineTo(xm+110,ym+60);
MoveTo(xm+110,ym+60);
LineTo(xm+110,ym+120);
MoveTo(xm+110,ym+120);
LineTo(xm+130,ym+120);

```

```

Rectangle(xm+130,ym+100,xm+170,ym+190);
TextOut(xm+142,ym+132,'ÁÔ');
MoveTo(xm+90,ym+170);
LineTo(xm+130,ym+170);
Rectangle(xm+210,ym+100,xm+250,ym+190);
TextOut(xm+218,ym+132,'ÁÑ');
MoveTo(xm+170,ym+120);
LineTo(xm+210,ym+120);
MoveTo(xm+190,ym+170);
LineTo(xm+210,ym+170);
MoveTo(xm+190,ym+170);
LineTo(xm+190,ym+210);
MoveTo(xm-10,ym+210);
LineTo(xm+190,ym+210);
MoveTo(xm+250,ym+150);
LineTo(xm+280,ym+150);
Brush.Color:=clWindow;
TextOut(xm-10,ym+4,'DA='+IntToStr(1-AAb[n-1]));
TextOut(xm-10,ym+44,'DB='+IntToStr(1-BBb[n-1]));
TextOut(xm-10,ym+104,'A='+IntToStr(aa));
TextOut(xm-10,ym+154,'B='+IntToStr(bb));
TextOut(xm-10,ym+194,'V='+IntToStr(pe));
TextOut(xm-10,ym+214,IntToStr(pe shr n));
TextOut(xm+98,ym+44,'D='+IntToStr(da));
TextOut(xm+98,ym+154,'LH');
TextOut(xm+98,ym+174,IntToStr(pl));
TextOut(xm+178,ym+104,'H');
if da=0 then TextOut(xm+178,ym+124,'L');
if da=1 then TextOut(xm+260,ym+132,'E');
TextOut(xm+260,ym+154,IntToStr(e1));
Rectangle(4,210,230,545);
Rectangle(4,254,230,345);
Rectangle(4,354,230,445);
Rectangle(4,454,230,545);
TextOut(95,215,'ÏØÉÁËË');
TextOut(50,230,' ');
TextOut(112,230,'. ');
TextOut(10,265,'. ');
TextOut(10,305,'. ');
TextOut(10,365,'. ');
TextOut(10,405,'. ');
TextOut(10,465,' ');
TextOut(10,505,'. ');
TextOut(187,222,' ');
Font.Color:=clBlue;
TextOut(20,278,'H');
TextOut(20,290,'L');
TextOut(20,318,'H');

```

```

TextOut(20,330,'L');
TextOut(20,378,'H');
TextOut(20,390,'L');
TextOut(20,418,'H');
TextOut(20,430,'L');
TextOut(20,478,'H');
TextOut(20,490,'L');
TextOut(20,518,'H');
TextOut(20,530,'L');
if cc>0 then
begin
Font.Color:=clBlue;
h1:=(dehp*2000 div cc+1)div 2;
TextOut(64,278,
  IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=(delp*2000 div cc+1)div 2;
TextOut(64,290, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=(deht*2000 div cc+1)div 2;
TextOut(64,318, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=(delt*2000 div cc+1)div 2;
TextOut(64,330, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');

      h1:=(dnhp*2000 div cc+1)div 2;
TextOut(128,278, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=(dnlp*2000 div cc+1)div 2;
TextOut(128,290, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=(dnht*2000 div cc+1)div 2;
TextOut(128,318, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=(dnlt*2000 div cc+1)div 2;
TextOut(128,330, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
      h1:=(nehp*2000 div cc+1)div 2;
TextOut(64,378, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=(nelp*2000 div cc+1)div 2;
TextOut(64,390, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=(neht*2000 div cc+1)div 2;
TextOut(64,418, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=(nelt*2000 div cc+1)div 2;
TextOut(64,430, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=(nnhp*2000 div cc+1)div 2;
TextOut(128,378, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=(nnlp*2000 div cc+1)div 2;
TextOut(128,390, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=(nnht*2000 div cc+1)div 2;
TextOut(128,418, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=(nnlt*2000 div cc+1)div 2;
TextOut(128,430, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
      h1:=((dehp+nehp)*2000 div cc+1)div 2;
TextOut(64,478, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');

```



```

h1:=((delp+nelp)*2000 div cc+1)div 2;
TextOut(64,490, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((deht+neht)*2000 div cc+1)div 2;
TextOut(64,518, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((delt+nelt)*2000 div cc+1)div 2;
TextOut(64,530, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
    h1:=((dnhp+nnhp)*2000 div cc+1)div 2;
TextOut(128,478, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((dnlp+nnlp)*2000 div cc+1)div 2;
TextOut(128,490, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((dnht+nnht)*2000 div cc+1)div 2;
TextOut(128,518, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((dnlt+nnlt)*2000 div cc+1)div 2;
TextOut(128,530, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
    h1:=((dehp+dnhp)*2000 div cc+1)div 2;
TextOut(192,278, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((delp+dnlp)*2000 div cc+1)div 2;
TextOut(192,290, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((deht+dnht)*2000 div cc+1)div 2;
TextOut(192,318, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((delt+dnlt)*2000 div cc+1)div 2;
TextOut(192,330, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
    h1:=((nehp+nnhp)*2000 div cc+1)div 2;
TextOut(192,378, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((nelp+nnlp)*2000 div cc+1)div 2;
TextOut(192,390, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((neht+nnht)*2000 div cc+1)div 2;
TextOut(192,418, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((nelt+nnlt)*2000 div cc+1)div 2;
TextOut(192,430, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
    h1:=((nehp+nnhp+dehp+dnhp)*2000 div cc+1)div 2;
TextOut(192,478, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((nelp+nnlp+delp+dnlp)*2000 div cc+1)div 2;
TextOut(192,490, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((neht+nnht+deht+dnht)*2000 div cc+1)div 2;
TextOut(192,518, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((nelt+nnlt+delt+dnlt)*2000 div cc+1)div 2;
TextOut(192,530, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
Font.Color:=clBlack;
h1:=((dehp+delp)*2000 div cc+1)div 2;
TextOut(64,265, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((deht+delt)*2000 div cc+1)div 2;
TextOut(64,305, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((dnhp+dnlp)*2000 div cc+1)div 2;
TextOut(128,265, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((dnht+dnlt)*2000 div cc+1)div 2;
TextOut(128,305, IntToStr(h1 div 10)+','+IntToStr(h1 mod 10)+'%');
h1:=((nehp+nelp)*2000 div cc+1)div 2;

```

```

TextOut(64,365, IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
h1:=((neht+nelt)*2000 div cc+1)div 2;
TextOut(64,405, IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
h1:=((nnhp+nnlp)*2000 div cc+1)div 2;
TextOut(128,365, IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
h1:=((nnht+nnlt)*2000 div cc+1)div 2;
TextOut(128,405, IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
h1:=((ep*2000 div cc+1)div 2;
TextOut(64,465, IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
h1:=((et*2000 div cc+1)div 2;
TextOut(64,505, IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
h1:=((ce-ep)*2000 div cc+1)div 2;
TextOut(128,465, IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
h1:=((ee-et)*2000 div cc+1)div 2;
TextOut(128,505, IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
dp:=dehp+dnhp+delp+dnlp;
h1:=(dp*2000 div cc+1)div 2;
TextOut(192,265, IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
dt:=deht+dnht+delt+dnlt;
h1:=(dt*2000 div cc+1)div 2;
TextOut(192,305, IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
h1:=((ce-dp)*2000 div cc+1)div 2;
TextOut(192,365, IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
h1:=((ee-dt)*2000 div cc+1)div 2;
TextOut(192,405, IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
h1:=((ce)*2000 div cc+1)div 2;
TextOut(192,465, IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
h1:=((ee)*2000 div cc+1)div 2;
TextOut(192,505, IntToStr(h1 div 10)+'+',IntToStr(h1 mod 10)+'%');
end;
Font.Color:=clRed;
TextOut(80,248,' '); TextOut(73,348,' '); TextOut(100,448,' ');
end;
end;
end;

```

```

procedure TForm1.Button15Click(Sender: TObject);
begin
Panel34.Visible:=True; Panel34.Caption:= 'Розрядність (7 - 15)';
Edit9.Visible:=True;
Button15.Visible:=False; Button1.Visible:=True;
end;

```

```

procedure TForm1.Panel35Click(Sender: TObject);
begin
nfe:=0;
Panel35.Caption:=' '; Panel18.Visible:=False;
Panel23.Visible:=False; Panel35.Color:=clBtnFace;

```

end;

```

procedure TForm1.Panel36Click(Sender: TObject);
begin
  nfr:=1-nfr;
  if nfr=0 then
  begin
    Panel6.Caption:=MF[f1];      Panel7.Caption:=MF[f2];
    Panel36.Caption:=' ' ;      Panel6.Color:=clAqua;
    Panel7.Color:=clAqua;  Panel36.Color:=clAqua;
  end
  else
  begin
    Panel6.Caption:='*';      Panel7.Caption:='*';
    Panel36.Caption:=' ' ;  Panel6.Color:=clBtnFace;
    Panel7.Color:=clBtnFace;  Panel36.Color:=clBtnFace;
  end;
end;

```

end;

```

procedure TForm1.Panel37Click(Sender: TObject);
begin
  Button7.Visible:=True;
  Panel37.Font.Color:=clBlack; Panel38.Font.Color:=clBlack;
  if Panel37.Caption<>' ' then sl:=1-sl;
  if sl=0 then
  begin
    Panel37.Color:=clCream; Panel37.Caption:='L';
  end
  else
  begin
    Panel37.Color:=clLime; Panel37.Caption:='L';
  end;
  if sh=0 then
  begin
    Panel38.Color:=clCream; Panel38.Caption:= 'H';
  end
  else
  begin
    Panel38.Color:=clLime; Panel38.Caption:='H';
  end;
end;
end;

```

```

procedure TForm1.Panel38Click(Sender: TObject);

```

```

begin

```

```

  Button7.Visible:=True;

```

```

  Panel37.Font.Color:=clBlack; Panel38.Font.Color:=clBlack;

```

```

  If Panel38.Caption<>' ' then sh:=1-sh;

```

```

  if sh=0 then

```

```

  begin

```

```
Panel38.Color:=clCream; Panel38.Caption:='H';
end
else
begin
Panel38.Color:=clLime; Panel38.Caption:='H';
end;
if sl=0 then
begin
Panel37.Color:=clCream; Panel37.Caption:='L';
end
else
begin
Panel37.Color:=clLime; Panel37.Caption:='L';
end;
end;
end.
```